Fragile Watermarking With Error-Free Restoration Capability

Xinpeng Zhang and Shuozhong Wang

***Abstract*—This paper proposes a novel fragile watermarking scheme capable of perfectly recovering the original image from its tampered version. In the scheme, a tailor-made watermark consisting of reference-bits and check-bits is embedded into the host image using a lossless data hiding method. On the receiver side, by comparing the extracted and calculated check-bits, one can identify the tampered image-blocks. Then, the reliable reference-bits extracted from other blocks are used to exactly reconstruct the original image. Although content replacement may destroy a portion of the embedded watermark data, as long as the tampered area is not too extensive, the original image information can be restored without any error.**

***Index Terms*—Error-free restoration, fragile watermarking, lossless data hiding.**

I. Introduction

In recent years, digital watermarking has attracted considerable research interests, and various techniques have been developed. While robust watermarks can be used for ownership verification, fragile watermarks are intended for checking integrity and authenticity of digital contents [1], [2]. When a portion of the original content is replaced with fake information, it is desirable to be able to locate the modified areas. Some fragile watermarking schemes divide a host image into small blocks and embed the mark into each block [3]–[5]. The embedded data may be a hash of the principal content of each cover-block. If the image has been changed, the image content and the watermark corresponding to the tampered blocks cannot be matched so that the tampered blocks are detected. Although the attacker may select suitable blocks from many watermarked images to counterfeit an illegal image containing a fake complete watermark [6], a smart watermarking method described in [7] uses two pieces of identical index information to generate a fragile watermark for each block to achieve security against this type of attack. Block-wise fragile watermarking schemes can only identify tampered blocks, but not the tampered pixels. In other words, they cannot find the detailed pattern of the modification. To overcome this drawback, some pixel-wise fragile watermarking schemes have been proposed, in which the watermark information derived from gray values of host pixels is embedded into the host pixels themselves [8]–[11]. So, tampered pixels can be identified due to the absence of watermark information they carry. In these methods, however, since some information derived from new pixel values may coincide with the watermark, localization of the tampered pixels is not complete, and detection of the tampering pattern is inaccurate. To resolve this problem, a fragile watermarking scheme in [12] embeds a set of tailor-made authentication data into a host image and introduces a statistical mechanism for image authentication. By estimating the modification strength, two different distributions corresponding to tampered and original pixels can be used to exactly locate the tampered pixels.

The watermarks in the above mentioned schemes are designed to detect slight changes in host images. If the embedded watermark is sensitive only to malicious content modification, but not to normal signal processing such as low-pass filtering and compression coding, it is termed as semi-fragile watermark [13]–[16]. In many semi-fragile schemes, the watermark is derived from the local image content and embedded into the host image in a robust manner. This way, the absence of watermark reveals the position of tampering.

Moreover, some watermarking approaches that can reconstruct the original content in the tampered areas have been proposed. Two methods were proposed in [17]. With the first method, the primary discrete cosine transform (DCT) coefficients of every block sized 8×8 are quantized and represented as 64 or 128 bits, which are used to replace one or two least significant bits of another block. In the second method, a low color depth version of the original image generated by reducing gray levels is cyclic-shifted and embedded into the pixel differences. After the malicious modification on a watermarked image is localized, the quantized DCT coefficients and the low color depth data extracted from reserved regions can be exploited to recover the principal content of tampered areas. In [18], the embedded watermark signal is the exclusive-OR between a pseudo-random sequence and the polarity information of DCT coefficients. Similarly, the original content in the tampered areas can roughly be retrieved by iterative projections of the polarity information on a convex set. However, these methods can only recover the main information in the tampered areas, but not the exact original content. Actually, in certain applications such as military or medical imaging, even very small distortion in the recovered image is unacceptable. Therefore, it is important to develop improved fragile watermarking schemes with error-free restoration capability.

Manuscript received February 05, 2008; revised August 22, 2008. Current version published December 10, 2008. This work was supported in part by the National Natural Science Foundation of China under Grants 60872116, 60832010, and 60773079, in part by the High-Tech Research and Development Program of China under Grant 2007AA01Z477, and in part by the Shanghai Leading Academic Discipline Project under Grants T0102. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alex C. Kot.

X. Zhang and S. Wang are with the School of Communication and Information Engineering, Shanghai University, Shanghai 200072, China (e-mail: xzhang@shu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2008.2007334

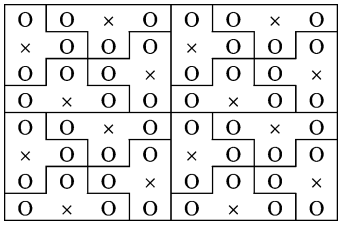


Fig. 1. Block made up of 16 pixel-patches, each of which contains one "unchangeable" pixel labeled "×", and three "changeable" pixels labeled "O".

As a special data-hiding technique, a number of lossless embedding methods can insert secret message into the host image in some invertible manner so that the original content can be perfectly restored after the hidden message is extracted. In [19], the host image is divided into blocks sized 4×4, 8×8, or 16×16, and gray values are mapped to a circle. After pseudo-randomly segmenting each block into two sub-regions, rotate histograms of the two subregions on this circle to embed one bit in each block. On the receiving side, the original block can be recovered from a marked image in an inverse process. Most other techniques make use of statistical redundancy of the host image by performing lossless compression in order to create a spare space to accommodate secret data. In the RS method [20], for example, a regular-singular status is defined for each group of pixels according to a flipping operation and a discrimination function. The entirety of RS status is then losslessly compressed to provide a space for data hiding. Alternatively, the least significant digits of pixel values in an L-ary system [21] or the least significant bits of quantized DCT coefficients in a JPEG image [22] can also be used to provide the required data space. In the difference expansion (DE) algorithm [23], differences between two adjacent pixels are doubled so that a new LSB plane without carrying any information of the original is generated. The additional message and a compressed location map, which is derived from the property of each pixel pair, are embedded into the generated LSB plane. Since compression rate of the location map is high, and since almost every pixel pair can carry one bit, the DE algorithm can embed a fairly large amount of secret data into a host image. When generalized integer transform [24] and histogram shifting technique [25] are combined with the DE algorithm, the performance is significantly improved.

The lossless data hiding technique can be integrated with fragile watermarking. When a digital signature of the host content is embedded in a lossless manner, a receiver can detect any modification to the marked medium if it has been tampered, otherwise the original host data can be retrieved without error. By using another framework of lossless fragile watermarking [26], the receiver can either locate the modified area from a tampered image or perfectly recover the original content from an authentic image. This means that the original content cannot be perfectly retrieved from a tampered image.

As mentioned above, previous fragile watermarking approaches can locate the tampered areas, and roughly reconstruct the main content. However, any distortion in the reconstructed content, no matter how small it is, is unacceptable to some applications, e.g., military or medical images. In other words, it is desired to exactly recover the original content from a tampered image. In this paper, we propose a novel fragile watermarking scheme with error-free restoration capability, in which a tailor-made watermark is derived from the original host image and embedded into the host using a lossless data-hiding technique. Although a malicious modification may destroy part of the embedded watermark-data, the tampered areas can be located if the malicious modification is not too extensive, and the watermark-data extracted from the reserved regions can be used to restore the host image without any error.

II. Watermarking With Error-Free Restoration Capability

In the proposed scheme, the watermark data to be hidden are made up of two parts: reference-bits, which are dependent on the original host image, and check-bits, which are determined by the host content and the reference-bits. A DE algorithm is employed to embed the reference-bits and check-bits into all blocks of the host image. In a transmission channel, an adversary may replace some content in a watermarked image with fake information. Although watermark data embedded in the tampered areas are destroyed, the watermarked content and the watermark data in other areas are unaffected. On the receiver side, after comparing the extracted check-bits with the calculated check-bits, one can identify the tampered blocks, and then extract the reliable reference-bits from the rest of the blocks to perfectly recover the original content in the image. Note that a necessary condition of error-free restoration is that the tampered area is not too extensive.

A. Watermark Embedding Procedure

1) Block/Patch Division: Before generation and insertion of watermark data, we first divide a host image into blocks and patches, and evaluate availability of the pixels for data embedding.

Denote the numbers of rows and columns in an original image as N_1 and N_2, and the total number of pixels as $N(N = N_1 \times N_2)$. Assuming that both N_1 and N_2 are multiples of eight, we first divide the original image into $N/64$ non-overlapped blocks sized 8×8, and denote the pixel-blocks as $\mathbf{G}_m$ $(m = 1, 2, \ldots, N/64)$ and the gray values of pixels in a block as $g_m(i,j)$ $(1 \leq i,j \leq 8)$. Each block is further divided into 16 T-shaped patches, each containing four pixels, in different orientations as shown in Fig. 1. The center pixel of a pixel-patch is called "unchangeable", and the other three "changeable". Thus, there are in total $N/4$ unchangeable and $3N/4$ changeable pixels. In each pixel-patch, all changeable pixels are neighbors of the unchangeable pixel. The unchangeable and changeable pixels are labeled "×" and "O" in Fig. 1, respectively. The pixels $g_m(1,3)$, $g_m(1,7)$, $g_m(2,1)$, $g_m(2,5)$, $g_m(3,4)$, $g_m(3,8)$, $g_m(4,2)$, $g_m(4,6)$, $g_m(5,3)$, $g_m(5,7)$, $g_m(6,1)$, $g_m(6,5)$, $g_m(7,4)$, $g_m(7,8)$, $g_m(8,2)$, and $g_m(8,6)$ are "unchangeable", while the others are "changeable". In the watermark embedding procedure, the original values of unchangeable pixels will be kept unchanged, and the differences between unchangeable and changeable pixels will be doubled by using difference expansion (DE) operations for watermark embedding. That means the values of changeable pixels may be altered.

TABLE I
DISTRIBUTION OF RATIO BETWEEN THE NUMBER OF UNUSABLE PIXELS AND THAT OF ALL CHANGEABLE PIXELS

Range	0~0.1%	0.1%~0.2%	0.2%~0.3%	0.3%~0.4%	0.4%~0.5%	0.5%~0.6%
Number of images	9	23	28	22	14	4

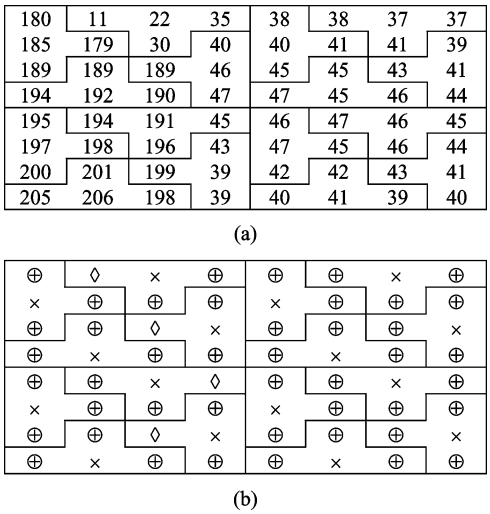


Fig. 2. (a) Example of pixel-block and (b) the unchangeable, usable, and unusable pixels labeled "×", "⊕" and "◇", respectively.

For each changeable pixel $g_m(i,j)$ $(1 \le m \le N/64, 1 \le i,j \le 8)$, denote the value of the unchangeable pixel belonging to a same patch as g_u. If $g_m(i,j) \ge g_u$, we check whether

$$g_u + [g_m(i,j) - g_u] \cdot 2 + 1 \ge 255. \quad (1)$$

If $g_m(i,j) < g_u$, we check whether

$$g_u + [g_m(i,j) - g_u] \cdot 2 \le 0. \quad (2)$$

When either (1) or (2) is true, we call the changeable pixel $g_m(i,j)$ "unusable", otherwise, call it "usable". Here, the term "usable" implies that a DE operation for watermark embedding does not cause any overflow or saturation at this pixel. The detailed DE operation will be described later. Fig. 2 shows an example of pixel-block, in which the unchangeable, usable and unusable pixels are labeled ×, ⊕ and ◇, respectively. In Fig. 2, the pixels $g_m(1,2)$, $g_m(3,3)$, $g_m(5,4)$ and $g_m(7,3)$ are unusable, and the other changeable pixels are usable. Since the values of adjacent pixels are close and the pixels with gray-levels near saturation are rare, most changeable pixels are usable, and the distortion due to DE operation would be low. For 100 images of landscapes and portraits, we calculate ratios between the numbers of unusable pixels and those of all changeable pixels. Table I gives distribution of the ratio. It can be seen that all the values are less than 0.6%.

2) Reference-Bit Generation: This step produces a set of reference-bits derived from the original image content. By representing the gray value of each pixel in 8 bits, the original image is equivalent to a total of $8 \cdot N$ bits. Then, we permute and divide the $8 \cdot N$ bits into $N/256$ bit-groups, each containing 2048 bits. The way of permutation is determined by a secret key. As such, the 2048 bits belonging to the same group are dispersed in the entire image. Denote the bits in a group as $b_t(1), b_t(2), \ldots, b_t(2048)$ $(t = 1, 2, \ldots, N/256)$, and, for each bit-group, calculate 128 reference-bits $r_t(1), r_t(2), \ldots, r_t(128)$,

$$\begin{bmatrix} r_t(1) \\ r_t(2) \\ \vdots \\ r_t(128) \end{bmatrix} = \mathbf{A}_t \cdot \begin{bmatrix} b_t(1) \\ b_t(2) \\ \vdots \\ b_t(2048) \end{bmatrix}, \quad t = 1, 2, \ldots, \frac{N}{256} \quad (3)$$

where $\mathbf{A}_t$ are pseudo-random binary matrixes sized 128×2048, and the arithmetic is modulo-2. The matrices $\mathbf{A}_t$ are also derived from the secret key. So, we have produced a total of $N/2$ reference-bits. Actually, if some of the original content is tampered, the reference-bits will be used to recover the corresponding original information.

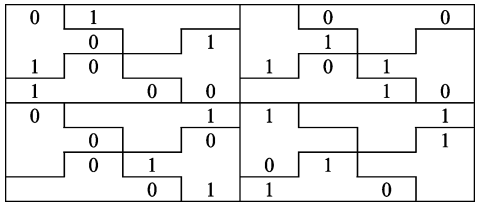


Fig. 3. 32 reference-bits at their mapped positions.

According to the secret key, pseudo-randomly select 32 changeable pixels from each pixel-block. The number of selected changeable pixels is $N/2$. Then, the $N/2$ reference-bits are also pseudo-randomly permuted and mapped to the selected changeable pixels in a one-to-one manner. That means the 128 reference-bits of a bit-group are also dispersed. For example, Fig. 3 shows 32 reference-bits at their mapped positions in a pixel-block. Then, the $N/2$ reference-bits will be embedded into their corresponding changeable pixels.

To ensure security, a number of operations in the watermarking scheme are dependent on the secret key, and the matrix $\mathbf{A}_t$ for different bit-groups and the selection of changeable pixels in different pixel-block should be mutually different. In fact, we may use a primary secret key, shared by the watermark hider and a receiver, to generate a pseudo-random sequence. For example, the sequence may be derived from a chaotic system with an initial condition determined by the secret key. Then, the generated sequence is divided into a series of pseudo-keys for directly controlling bit permutation, matrix generation, and pixel selection. In the following, we do not distinguish the primary key and the series of derived pseudo-keys.

3) Check-Bit Generation: In this step, we produce the check-bits used for tampering identification on the receiver side. For each pixel-block $\mathbf{G}_m$, we collect the values of 16 unchangeable pixels and all usable pixels, and the reference-bits corresponding to the usable pixels. Then, feed them into a hash function to produce 64 hash-bits $h_m(1), h_m(2), \ldots, h_m(64)$. For example, using the original block G_m in Fig. 2(a) and the mapped reference-bits in Fig. 3, the values of pixels

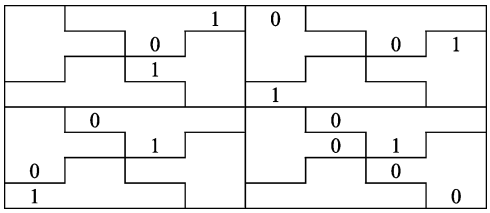


Fig. 4. Sixteen check-bits occupying the positions that are not occupied in Fig. 3.

176	0	22	49	36	39	37	37
185	174	38	35	40	43	45	38
193	187	255	46	51	45	46	41
197	192	188	48	50	45	48	47
193	197	191	0	46	48	46	45
197	199	202	47	47	43	47	48
203	196	255	39	37	44	45	41
205	206	190	40	40	41	37	39

Fig. 5. Watermarked version of the block in Fig. 2(a).

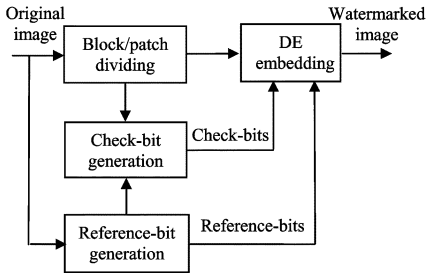


Fig. 6. Watermark embedding procedure.

except $g_m(1,2)$, $g_m(3,3)$, $g_m(5,4)$ and $g_m(7,3)$, plus the reference-bits except the four bits at the corresponding positions, i.e., (1,2), (3,3), (5,4) and (7,3), are used to compute the hash-bits. Here, the hash function must have the collision-resistant property: it is hard to find two different inputs corresponding to a same output or two outputs with a small Hamming distance. In this way, we yield a total of N hash-bits.

However, the amount of hash-bits is too large to be embedded. Thus, a folded version is produced, embedded and used for tampering localization. Pseudo-randomly permute and divide the hash-bits into $N/4$ subsets, each having four hash-bits, according to the secret key. Then, calculate modulo-2 sum of the four hash-bits in each subset, and call the $N/4$ sums the check-bits.

We map, in a one-to-one manner, the $N/4$ check-bits to the rest changeable pixels, which have not been selected to map the reference-bits. In a pixel-block, in summary, 32 changeable pixels are mapped to reference-bits, and 16 mapped to check-bits. For example, Fig. 4 shows 16 check-bits occupying the positions that are not occupied in Fig. 3.

4) DE Embedding: The watermark data consisting of the reference-bits and check-bits are embedded into their corresponding changeable pixels using the DE (difference expansion) method. For each usable pixel $g_m(i,j)$ $(1 \le m \le N/64, 1 \le i,j \le 8)$, denoting the value of the unchangeable pixel belonging to a same patch as g_u, calculate the new value of the usable pixel by using a DE operation

$$\tilde{g}_m(i,j) = g_u + [g_m(i,j) - g_u] \cdot 2 + w \tag{4}$$

where w is the reference-bit or check-bit mapping the usable pixel. Equation (4) implies that the difference between each usable pixel and its unchangeable pixel is doubled, and the least significant bit of the new difference-value is exploited to accommodate the corresponding watermark-bit. Obviously, new values of all usable pixels are within $[1, 254]$. On the other hand, the watermark-bits mapping to unusable pixels are ignored, and new values of unusable pixels are modified to saturation according to the following rule:

$$\tilde{g}_m(i,j) = \begin{cases} 0, & \text{if } g_m(i,j) < g_u \\ 255, & \text{if } g_m(i,j) \ge g_u \end{cases}. \tag{5}$$

In other words, the extreme white/black indicates dummy positions, and the watermark-bits mapping to the unusable pixels are not embedded, and therefore not used on the receiver side for tampering-localization/image-restoration. As mentioned above, the values of unchangeable pixels are preserved. Here, the more the difference between a usable pixel and its unchangeable pixel, the more modification would be introduced. So, distortion due to watermark embedding is related to the image content. Texture and edge areas are more distorted than smooth areas. Nonetheless, as gray value changes in busy areas are more tolerable to human visual system (HVS), visual qualities of watermarked images are generally acceptable. Assuming the original block is shown in Fig. 2(a) and the corresponding watermark-bits shown in Figs. 3–5 gives the watermarked block.

The entire procedure of watermark embedding is sketched in Fig. 6.

B. Image Restoration Procedure

Suppose that an adversary replaces some content in a watermarked image with fake information. We name the blocks in which all pixels are not changed or, in rare cases, only some saturated white/black changeable pixels are changed to saturated black/white, as "reserved blocks". Otherwise, the blocks are named "tampered blocks". In other words, if any of the following three cases occurs, the block is termed "tampered blocks": i) the unchangeable pixel is altered; ii) the unsaturated changeable pixel is altered; or iii) the saturated white/black of changeable pixel is changed to an unsaturated value. Denote the ratio between the number of tampered blocks and the number of all blocks, that is, the rate of tampering, as α. After obtaining an image on the receiver side, we first attempt to extract the watermark data, and identify the tampered blocks according to the check-bits, and then restore the original values of all pixels in the tampered blocks and the saturated pixels at changeable positions according to the reference-bits extracted from reserved blocks.

1) Watermark-Data Extraction: The received image is first divided into $N/64$ blocks and $N/4$ patches in the same manner as in the embedding process, and the pixels in the received image are denoted $g'_m(i,j)$ $(1 \le m \le N/64, 1 \le i,j \le 8)$. According to the received values, the changeable pixels are divided into two types: the saturated pixels with values 0 or 255, and

0			1	0	0		0
	0	0	1		1	0	1
1	0			1	0	1	
1		0	0	1		1	0
0	0			1	0		1
	0	1	0		0	1	1
0	0			0	1	0	
1		0	1	1		0	0

(a)

180		22	35	38	38	37	37
185	179	30	40	40	41	41	39
189	189		46	45	45	43	41
194	192	190	47	47	45	46	44
195	194	191		46	47	46	45
197	198	196	43	47	45	46	44
200	201		39	42	42	43	41
205	206	198	39	40	41	39	40

(b)

Fig. 7. (a) Extracted watermark-bits from Fig. 6 and (b) recovered gray-values.

the unsaturated pixels with values in $[1, 254]$. We can attempt to extract the embedded watermark-bit from each unsaturated changeable pixel

$$w = \mathrm{mod}\left[g'_m(i,j) - g'_u, 2\right] \tag{6}$$

and to recover the original gray-value

$$\hat{g}_m(i,j) = g'_u + \left\lfloor \frac{g'_m(i,j) - g'_u}{2} \right\rfloor \tag{7}$$

where g'_u is the received value of the unchangeable pixel belonging to a same patch and the operator "$\lfloor\rfloor$" takes the nearest integer toward negative infinity. For example, after receiving a pixel-block as in Fig. 5, one can extract 44 watermark-bits and recover the original gray-values except for four saturated changeable pixels. Fig. 7 shows the extracted watermark-bits and the recovered pixel-values. Note that, if the block has been tampered, the extracted watermark-bit and the recovered pixel-value may be incorrect.

2) Tampered-Block Identification: After obtaining the extracted watermark data and an interim recovered version, we compare the extracted check-bits with the calculated check-bits for each block. The state of consistency between them is a criterion for judging a block as "reserved" or "tampered".

For each pixel-block of received image, we collect the values of 16 unchangeable pixels, the recovered values of all unsaturated changeable pixels, and the extracted reference-bits corresponding to the unsaturated changeable pixels, and feed them into the same hash function to compute 64 hash-bits $h'_m(1), h'_m(2), \ldots, h'_m(64)$. We call them "calculated hash-bits". The hash should not change if the block is not tampered. Otherwise, probability of hash-bit flipping is 1/2. Note that, if the modification is only flipping of saturated white/black changeable pixels into saturated black/white, we still call this block "reserved", since the modification does not affect the watermark-bit extraction and hash-bit calculation. The original values in the tampered blocks and those of saturated pixels will be recovered after identifying the tampered blocks.

TABLE II
VALUES OF T WITH DIFFERENT N_{E}

N_{E}	64	63	62	61	60	59	58
$T(N_{\mathrm{E}})$	10	9	9	9	8	8	8

Collect N calculated hash-bits derived from the received image, divide them into $N/4$ subsets in a same manner according to the secret key, and calculate modulo-2 sums of the four hash-bits in each subset. These sums are called "calculated check-bits". Because the 64 hash-bits of each pixel-block are distributed into different subsets, each block corresponds to 64 calculated check-bits. On the other hand, since a check-bit is modulo-2 sum of four hash-bits, each check-bit corresponds to 4 pixel-blocks.

We compare the 64 calculated check-bits of each block with the corresponding ones extracted from the received image. Note that we cannot extract any bits from saturated changeable pixels. Therefore the number of extracted check-bits corresponding to a block may be less than 64. Denote the number of extracted check-bits as N_{E} $(N_{\mathrm{E}} \leq 64)$, the number of extracted check-bits not equal to their corresponding calculated ones as N_{F} $(N_{\mathrm{F}} \leq N_{\mathrm{E}})$, and the ratio between the number of saturated changeable pixels and the number of all changeable pixels as γ. So, the value of N_{E} follows a binomial distribution, and its probability distribution function is

$$P_{N_{\mathrm{E}}}(k) = \binom{64}{N_{\mathrm{E}}} \cdot (1-\gamma)^{N_{\mathrm{E}}} \cdot \gamma^{64-N_{\mathrm{E}}} \quad k = 0, 1, \ldots, 64. \tag{8}$$

Actually, after extracting the watermark data from the received image, the value of N_{E} for each block is fixed. If the block is tampered, the probability of a calculated check-bit being unequal to the corresponding extracted bit is also 1/2. That means the value of N_{F} follows another binomial distribution

$$P_{T,N_{\mathrm{F}}}(k) = \binom{N_{\mathrm{E}}}{k} \cdot 2^{-N_{\mathrm{E}}} \quad k = 0, 1, \ldots, N_{\mathrm{E}}. \tag{9}$$

For a given N_{E}, we find an integer T satisfying

$$\sum_{k=0}^{T} P_{T,N_{\mathrm{F}}}(k) < 10^{-9} \tag{10}$$

and

$$\sum_{k=0}^{T+1} P_{T,N_{\mathrm{F}}}(k) \geq 10^{-9}. \tag{11}$$

If N_{F} is greater than the threshold T, we judge the block as "tampered". Otherwise, it is "reserved". Thus, probability of a tampered block being falsely judged as "reserved" is less than a negligible 10^{-9}. Table II lists the values of T with different N_{E}.

Let us find probability of a reserved block being falsely judged as "tampered". Denoting the ratio between the number of tampered blocks and the number of all blocks as α, the calculated hash-bits are altered with probability $\alpha/2$. Because the calculated check-bit is a modulo-2 sum of 4 calculated hash-bits, it would be flipped when the number of altered

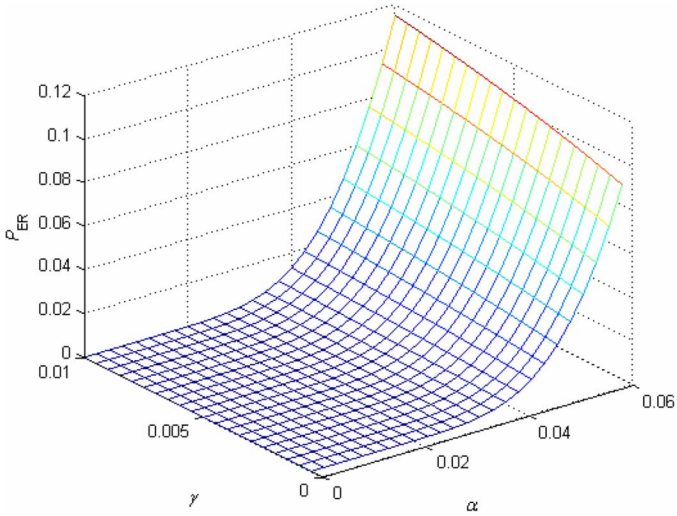


Fig. 8. Values of P_{ER} with different γ and α.

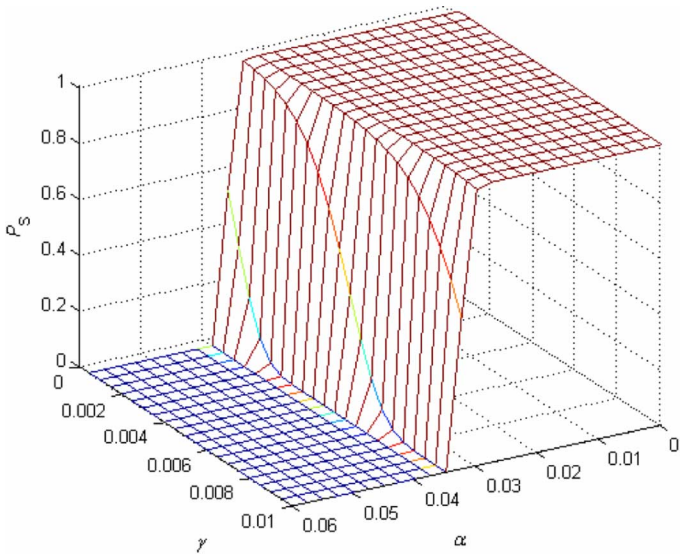


Fig. 9. Values of P_{S} with different γ and α when $N = 512^2$.

hash-bits is 1 or 3. For a reserved block, since its hash-bits are never altered, a calculated hash-bit of reserved block is flipped if all the three hash-bits or only one hash-bit in a same subset is altered. Thus, the calculated check-bits of a reserved block are flipped with probability

$$P_{\mathrm{EC}} = \left(\frac{\alpha}{2}\right)^3 + 3 \cdot \left(1 - \frac{\alpha}{2}\right)^2 \cdot \left(\frac{\alpha}{2}\right). \tag{12}$$

On the other hand, the extracted check-bits may be incorrect because of the malicious modification on the watermarked image. Denoting the probability of check-bit extraction error as P_{EE}, probability of the calculated check-bits differing from their corresponding extracted ones is

$$P_{\mathrm{E}} = P_{\mathrm{EC}} \cdot (1 - P_{\mathrm{EE}}) + P_{\mathrm{EE}} \cdot (1 - P_{\mathrm{EC}}). \tag{13}$$

Here, P_{EE} is within $[0, \alpha]$. For a reserved block, the value of N_{F} obeys the following binomial distribution:

$$P_{\mathrm{R},N_{\mathrm{F}}}^{N_{\mathrm{E}}}(k) = \binom{N_{\mathrm{E}}}{k} \cdot P_{\mathrm{E}}^{k} \cdot (1 - P_{\mathrm{E}})^{N_{\mathrm{E}}-k} \quad k = 0, 1, \ldots, N_{\mathrm{E}}. \tag{14}$$

So, probability of a reserved block being falsely judged as "tampered" is

$$P_{\mathrm{ER}} = \sum_{k_1=1}^{64} \left[P_{N_{\mathrm{E}}}(k_1) \cdot \sum_{k_2=T(k_1)+1}^{k_1} P_{\mathrm{R},N_{\mathrm{F}}}^{k_1}(k_2) \right]. \tag{15}$$

Fig. 8 shows the values of P_{ER} with different γ and α. Here, the threshold T is chosen according to Table II, and the value of P_{EE} is $\alpha/2$. Then, expectation of the rate of blocks being judged as "tampered" is $(\alpha + P_{\mathrm{ER}})$. Although a few of reserved blocks may be falsely judged as "tampered", we can re-find their original content in the next step when the area judged as "tampered" is not too extensive.

3) Image Restoration: In this step, we will restore the original gray values of all pixels in blocks judged as "tampered" and saturated changeable pixels in blocks judged as "reserved," while the original values of unsaturated changeable pixels in blocks judged as "reserved" have been recovered with (7). Here, the blocks judged as "tampered" contain the actual tempered blocks and some reserved blocks with false judgments, and the case that the tampered blocks are falsely judged as "reserved" is ignored because of the extremely low probability.

As mentioned above, $8 \cdot N$ bits are used to represent the original image and divided into $N/256$ bit-groups, each of which contains 2048 bits and is compressed to 128 reference-bits using (3). On the receiver side, the extracted reference-bits obtained only from the unsaturated changeable pixels in blocks judged as "reserved" are reliable. That means, for each bit-group, the number of reliable extracted reference-bits, denoting n_t, may be less than 128. So, (3) implies

$$\begin{bmatrix} r_t(s_1) \\ r_t(s_2) \\ \vdots \\ r_t(s_{n_t}) \end{bmatrix} = \mathbf{A}_t^{(\mathrm{R})} \cdot \begin{bmatrix} b_t(1) \\ b_t(2) \\ \vdots \\ b_t(2048) \end{bmatrix}, \quad t = 1, 2, \ldots, \frac{N}{256} \tag{16}$$

where the left side contains all reliable extracted reference-bits, and $\mathbf{A}_t^{(\mathrm{R})}$ is a matrix consisting of the rows in $\mathbf{A}_t$ corresponding to the reliable extracted reference-bits. Furthermore, the 2048 bits are made up of two types: the missing bits that are from the blocks judged as "tampered" or the saturated changeable pixels, and the recovered bits that are from other positions. Since the 2048 bits belonging to a same bit-group are dispersed over the entire image, the number of missing bits in each bit-group is small if the area to be restored is not too extensive. So, the reliable reference-bits can provide sufficient information to recover the original values of the missing bits. Denote a column vector

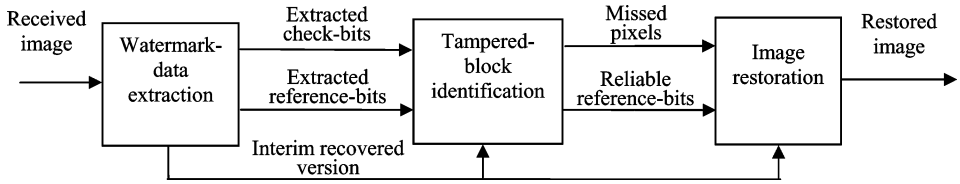


Fig. 10. Procedure of image restoration.

consisting of the missing bits as $B_{t,1}$, and a column vector consisting of the recovered bits as $B_{t,2}$. Equation (16) can be reformulated as

$$\begin{bmatrix} r_t(s_1) \\ r_t(s_2) \\ \vdots \\ r_t(s_{n_t}) \end{bmatrix} - \mathbf{A}_t^{(\mathrm{R},2)} \cdot B_{t,2} = \mathbf{A}_t^{(\mathrm{R},1)} \cdot B_{t,1}, t = 1, 2, \ldots, \frac{N}{256} \tag{17}$$

where $\mathbf{A}_t^{(\mathrm{R},1)}$ is a matrix consisting of the columns in $\mathbf{A}_t^{(\mathrm{R})}$ corresponding to the missing bits, and $\mathbf{A}_t^{(\mathrm{R},2)}$ is a matrix consisting of the columns in $\mathbf{A}_t^{(\mathrm{R})}$ corresponding to the recovered bits. In (17), the left side and the matrix $\mathbf{A}_t^{(\mathrm{R},1)}$ are known, and the purpose is to find $B_{t,1}$. Denoting the number of elements in $B_{t,1}$ as n_b, the size of $\mathbf{A}_t^{(\mathrm{R},1)}$ is $n_t \times n_\mathrm{b}$. We will solve the n_b unknowns according to the n_t equations in a binary system. Here, because both the reference-bits and the recovered bits in (17) are reliable, the original bits of $B_{t,1}$ must be a solution of (17). However, if the number of unknowns, n_b, is too large, or there are too many linearly dependent equations in (17), the solution may not be unique and, in this case, we cannot find the true solution, which is exactly the original bits, in the solution space. In summary, as long as (17) has a unique solution, we can obtain the original bits by using the Gaussian elimination method, that is, the restoration of original content will be successful.

Here is a discussion on the probability that (17) has a unique solution. The sufficient and necessary condition is that the rank of $\mathbf{A}_t^{(\mathrm{R},1)}$ equals n_b, meaning that the n_b columns of $\mathbf{A}_t^{(\mathrm{R},1)}$ are linearly independent. Consider a random binary matrix containing n_t rows and k columns, and denote probability of its columns being linearly dependent as $q(n_t, k)$. So, we have

$$q(n_t, 1) = \frac{1}{2^{n_t}} \tag{18}$$

$$q(n_t, k+1) = q(n_t, k) + [1 - q(n_t, k)] \cdot \frac{2^k}{2^{n_t}} \tag{19}$$

$$k = 1, 2, \ldots, n_t - 1$$

$$q(n_t, k) = 1, \text{ if } k > n_t. \tag{20}$$

Denote the sum of the number of changeable pixels in blocks judged as "tampered" and the number of saturated changeable pixels in blocks judged as "reserved" as S_1, the ratio between S_1 and the number of all changeable pixels as β_1. The number of changeable pixels in blocks with judgment "tampered" is $(\alpha + P_\mathrm{ER}) \cdot 3N/4$. If the saturated pixels are distributed over the entire image uniformly, the number of saturated changeable pixels in blocks with judgment "reserved" is $\gamma \cdot (1 - \alpha - P_\mathrm{ER}) \cdot 3N/4$. Thus

$$S_1 = [\alpha + P_\mathrm{ER} + \gamma \cdot (1 - \alpha - P_\mathrm{ER})] \cdot \frac{3N}{4}. \tag{21}$$

Since the number of all changeable pixels is $3N/4$

$$\beta_1 = \alpha + P_\mathrm{ER} + \gamma \cdot (1 - \alpha - P_\mathrm{ER}). \tag{22}$$

The probability distribution function of n_t follows a binomial distribution

$$P_{n_t}(u) = \binom{128}{u} \cdot (1 - \beta_1)^u \cdot \beta_1^{128-u}, \quad u = 0, 1, \ldots, 128. \tag{23}$$

Denote the sum of the number of pixels in blocks judged as "tampered" and the number of saturated changeable pixels in blocks judged as "reserved" as S_2, and the ratio between S_2 and the number of all pixels as β_2. The number of pixels in blocks judged as "tampered" is $(\alpha + P_\mathrm{ER}) \cdot N$, and the number of saturated changeable pixels in blocks judged as "reserved" is $\gamma \cdot (1 - \alpha - P_\mathrm{ER}) \cdot 3N/4$. Thus

$$S_2 = (\alpha + P_\mathrm{ER}) \cdot N + [\gamma \cdot (1 - \alpha - P_\mathrm{ER})] \cdot \frac{3N}{4} \tag{24}$$

and

$$\beta_2 = \alpha + P_\mathrm{ER} + \gamma \cdot (1 - \alpha - P_\mathrm{ER}) \cdot \frac{3}{4}. \tag{25}$$

The probability distribution function of n_b follows another binomial distribution:

$$P_{n_\mathrm{b}}(v) = \binom{2048}{v} \cdot \beta_2^v \cdot (1 - \beta_2)^{2048-v} \quad v = 0, 1, \ldots, 2048. \tag{26}$$

So, the probability of all columns in $\mathbf{A}_t^{(\mathrm{R},1)}$ being linearly independent is

$$P_\mathrm{LI} = \sum_{u=0}^{128} \sum_{v=0}^{2048} \{P_{n_t}(u) \cdot P_{n_\mathrm{b}}(v) \cdot [1 - q(u, v)]\}. \tag{27}$$

Since there are $N/256$ bit-groups in total, we can recover all the missing bits with probability

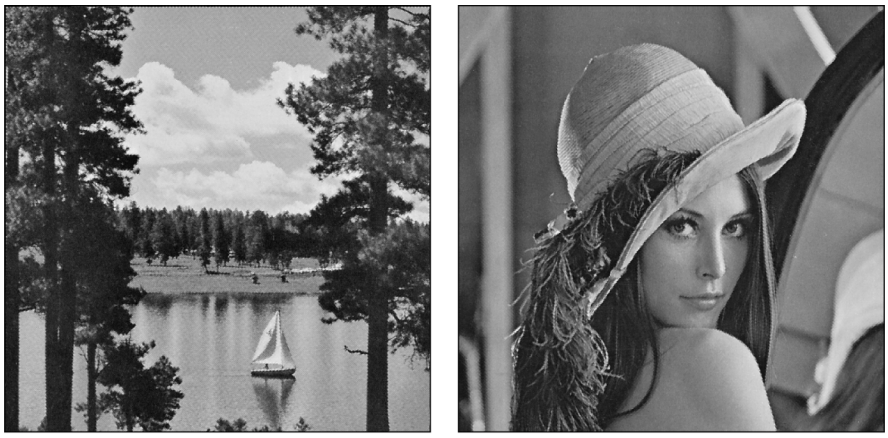
$$P_\mathrm{S} = P_\mathrm{LI}^{N/256}. \tag{28}$$

In summary, this probability is dependent on γ, α and N. When the tampering is not too severe, P_S is very close to 1. For example, Fig. 9 shows the values of P_S with different γ and α, where $N = 512^2$. It can be seen that the original image can be perfectly recovered when the rate of tampered blocks is no more than 0.032.

The procedure of image restoration is sketched in Fig. 10.

(a) (b)

Fig. 11. Two original images: (a) Lake and (b) Lena.



(a) (b)

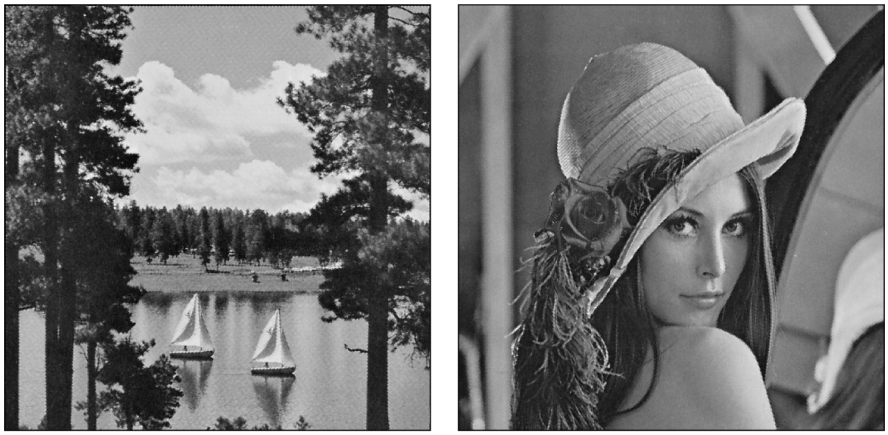
Fig. 12. Two watermarked images with PSNR 26.1 and 29.6 dB, respectively.

III. Experimental Results

Two test images Lake and Lena sized 512×512 were used as the host, shown in Fig. 11. Fig. 12 gives two watermarked versions, and PSNR values due to watermark embedding were 26.1 and 29.6 dB, respectively. Since Lake is busier than Lena, PSNR of watermarked Lake is lower than that of watermarked Lena. We replicated a boat and its shadow by changing 5.1×10^3 pixels and planted a flower on the girl's hat by changing 7.0×10^3 pixels to modify the watermarked images. The tampered images are shown in Fig. 13. In Fig. 13(a), the ratio between the numbers of saturated changeable pixels and all changeable pixels is $\gamma = 0.0041$ and the ratio between the number of tampered blocks and the number of all blocks is $\alpha = 0.026$. In Fig. 13(b), $\gamma = 0.0012$ and $\alpha = 0.037$. According to Fig. 8, the theoretical numbers of reserved blocks falsely judged as "tampered" are 0.61 and 13.92, similar to the actual numbers 0 and 10. Fig. 14 shows the positions of blocks with judgment "tampered" and saturated changeable pixels. In Fig. 14(b), the 10 isolated black blocks indicate the positions of falsely judged reserved blocks. By using the image restoration procedure, both the original images Lake and Lena can be perfectly recovered from the tampered versions. When embedding watermark into 100 host images using the proposed scheme, the average value of PSNR was 28.7 dB. Although the distortion is considerable, the receiver knowing the secret key can reconstruct the original content without error.

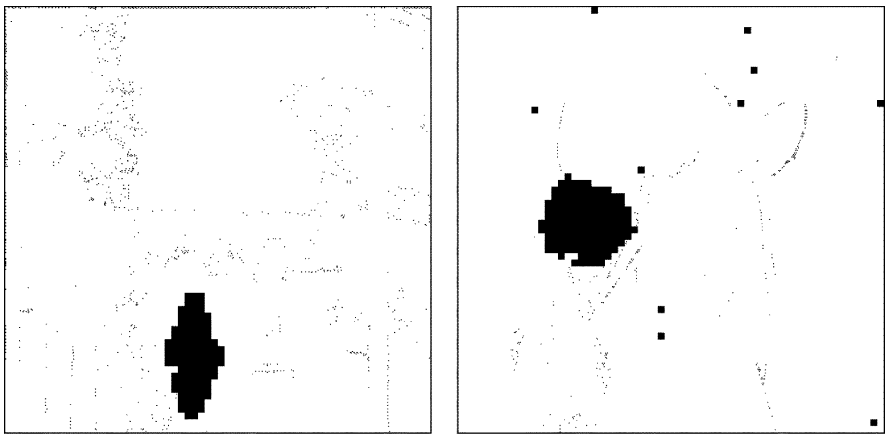
Fig. 15 gives an $\alpha - \gamma$ curve with $P_{\rm S} = 0.5$. Actually, the value of $P_{\rm S}$ changes rapidly in the neighborhood of the curve. Therefore the curve can be viewed as a boundary between two regions with and without error-free restoration capability. After using other test images with the same size as the host and tampering the watermarked images with different tampering rates, we attempted to recover the original images. Successful and unsuccessful restoration operations are respectively marked by "O" and "$\times$" in Fig. 15. In general, the original content of an image sized 512×512 can be perfectly recovered when the tampering rate is less than 3.2%. The experimental results were in agreement with the theoretical boundary.

Table III gives a comparison of several fragile watermarking schemes with restoration capability. By allowing more distortion in the watermarked image, the proposed scheme can recover the original content without error. In the previous methods, the main content in a region is embedded into another region of the image so that the restoration cannot be executed when some region and the region accommodating its original information are both tampered. In the proposed scheme, since both the bits



(a) (b)

Fig. 13. Two tampered images.



(a) (b)

Fig. 14. Positions of blocks with judgment "tampered" and saturated changeable pixels.

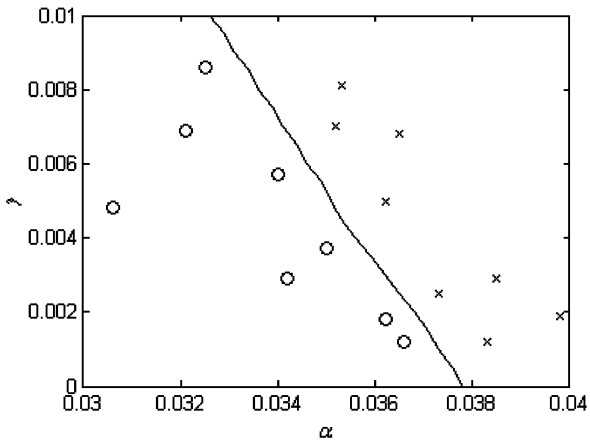


Fig. 15. Theoretical boundary of error-free restoration capability, and results of successful and unsuccessful restoration experiments.

in each bit-group and the corresponding reference-bits are dispersed over the entire image, the original image can be perfectly recovered if the tampering is not too severe.

IV. Conclusion and Discussion

This paper proposes a novel fragile watermarking scheme capable of recovering the original image without any error. In this scheme, the reference-bits determined by the host image and the check-bits derived from the hash of blocks are embedded into the entire image by using a lossless DE embedding technique. This way, the original content in most reserved area can be directly recovered through an inverse DE operation. By folding the hash-bits as the check-bits, the amount of data to be embedded for tampered-block localization is saved, and the tampered blocks can be identified by introducing a statistical mechanism. Furthermore, the reference-bits extracted from the reserved regions, as well as the original data recovered from the reserved regions, are exploited to retrieve the modified content. As long as the modified area is not too extensive, the original host image can be restored perfectly.

The proposed fragile watermarking scheme can also be used for color images in two different ways. In a component-wise manner, the red, green, and blue components of a color image are viewed as three single gray images, and the watermark embedding and image restoration procedures may be respectively

TABLE III
COMPARISON OF RESTORATION CAPABILITY AMONG FRAGILE WATERMARKING SCHEMES

Watermarking scheme	PSNR due to watermarking	PSNR in restored area	Condition of restoration
Method 1 in [17]	43.8 dB	21.5 dB	Regions storing the original information of tampered areas must be reserved.
Method 2 in [17]	33.1 dB	28.8 dB	
Method in [18]	36.7 dB	22.8 dB	
The proposed method	28.7 dB	$+\infty$	Tampering is not too severe.

executed in the three components. Alternatively, this can be realized in a block-wise manner. Each block containing three components is regarded as a unit for tampering localization. That means the hash-bits of 48 unchangeable pixels, usable pixels, and reference-bits corresponding to the usable pixels in each color block are used to obtain the check-bits, which will be embedded as well as the reference-bits.

Some issues deserve further investigation in the future. One is the relation between the watermark-induced distortion and the capability of image restoration. An upper bound of the tampered area that can be perfectly restored needs to be found on a theoretical basis. Schemes with less distortion while keeping error-free restoration capability are desired.

REFERENCES

[1] C. Vleeschouwer, J.-F. Delaigle, and B. Macq, "Invisibility and application functionalities in perceptual watermarking-an overview," *Proc. IEEE*, vol. 90, no. 1, pp. 64–77, Jan. 2002.

[2] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—a survey," *Proc. IEEE*, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.

[3] P. W. Wong and N. Memon, "Secret and public key image watermarking schemes for image authentication and ownership verification," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1593–1601, Oct. 2001.

[4] S. Suthaharan, "Fragile image watermarking using a gradient image for improved localization and security," *Pattern Recognit. Lett.*, vol. 25, pp. 1893–1903, 2004.

[5] H. Yang and A. C. Kot, "Binary image authentication with tampering localization by embedding cryptographic signature and block identifier," *IEEE Signal Process. Lett.*, vol. 13, pp. 741–744, 2006.

[6] M. Holliman and N. Memon, "Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 432–441, Mar. 2000.

[7] J. Fridrich, "Security of fragile authentication watermarks with localization," in *Proc. SPIE Security and Watermarking of Multimedia Contents IV*, San Jose, CA, Jan. 2002, vol. 4675, pp. 691–700.

[8] H. Lu, R. Shen, and F.-L. Chung, "Fragile watermarking scheme for image authentication," *Electron. Lett.*, vol. 39, no. 12, pp. 898–900, 2003.

[9] H. He, J. Zhang, and H.-M. Tai, "A wavelet-based fragile watermarking scheme for secure image authentication," in *Proc. 5th Int. Workshop on Digital Watermarking (IWDW 2006)*, 2006, vol. 4283, Lecture Notes in Computer Science, pp. 422–432.

[10] S.-H. Liu, H.-X. Yao, W. Gao, and Y.-L. Liu, "An image fragile watermark scheme based on chaotic image pattern and pixel-pairs," *Appl. Math. and Comput.*, vol. 185, no. 2, pp. 869–882, 2007.

[11] J. Wu, B. B. Zhu, S. Li, and F. Lin, "A secure image authentication algorithm with pixel-level tamper localization," in *Proc. Int. Conf. Image Processing*, 2004, pp. 1573–1576.

[12] X. Zhang and S. Wang, "Statistical fragile watermarking capable of locating individual tampered pixels," *IEEE Signal Process. Lett.*, vol. 14, no. 10, pp. 727–730, Oct. 2007.

[13] K. Maeno, Q. Sun, S.-F. Chang, and M. Suto, "New semi-fragile image authentication watermarking techniques using random bias and nonuniform quantization," *IEEE Trans. Multimedia*, vol. 8, no. 1, pp. 32–45, Feb. 2006.

[14] C. Fei, D. Kundur, and R. H. Kwong, "Analysis and design of secure watermark-based authentication systems," *IEEE Trans. Inform. Forensics and Security*, vol. 1, no. 1, pp. 43–55, 2006.

[15] Z.-M. Lu, D.-G. Xu, and S.-H. Sun, "Multipurpose image watermarking algorithm based on multistage vector quantization," *IEEE Trans. Image Process.*, vol. 14, no. 6, pp. 822–831, Jun. 2005.

[16] O. Altun, G. Sharma, M. U. Celik, and M. F. Bocko, "A set theoretic framework for watermarking and its application to semifragile tamper detection," *IEEE Trans. Inform. Forensics and Security*, vol. 1, no. 4, pp. 479–492, 2006.

[17] J. Fridrich and M. Goljan, "Images with self-correcting capabilities," in *Proc. IEEE Int. Conf. Image Processing*, 1999, pp. 792–796.

[18] X. Zhu, A. Hob, and P. Marziliano, "A new semi-fragile image watermarking with robust tampering restoration using irregular sampling," *Signal Process.: Image Commun.*, vol. 22, no. 5, pp. 515–528, 2007.

[19] C. Vleeschouwer, J.-F. Delaigle, and B. Macq, "Circular interpretation of bijective transformations in lossless watermarking for media asset management," *IEEE Trans. Multimedia*, vol. 5, no. 1, pp. 97–105, Feb. 2003.

[20] M. Goljan, J. Fridrich, and R. Du, "Distortion-Free data embedding," in *Proc. 4th Int. Workshop on Information Hiding*, 2001, vol. 2137, Lecture Notes in Computer Science, pp. 27–41.

[21] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Reversible data hiding," in *Proc. Int. Conf. Image Processing II*, 2002, pp. 157–160.

[22] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding for all image formats," in *Proc. SPIE Security and Watermarking of Multimedia Contents IV*, 2002, vol. 4675, pp. 572–583.

[23] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits, Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[24] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Trans. Image Process.*, vol. 13, no. 8, pp. 1147–1156, 2004.

[25] D. M. Thodi and J. J. Rodríguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, Mar. 2007.

[26] M. U. Celik, G. Sharma, and A. M. Tekalp, "Lossless watermarking for image authentication: A new framework and an implementation," *IEEE Trans. Image Process.*, vol. 15, no. 4, pp. 1042–1049, Apr. 2006.

Xinpeng Zhang received the B.S. degree in computation mathematics from Jilin University, Jilin, China, in 1995, and the M.E. and Ph.D. degrees in communication and information system from Shanghai University, Shanghai, China, in 2001 and 2004, respectively.

Since 2004, he has been with the faculty of the School of Communication and Information Engineering, Shanghai University, where he is currently a Professor. His research interests include information hiding, image processing, and digital forensics.

Shuozhong Wang received the B.S. degree in 1966 from Peking University, Peking, China, and the Ph.D. degree in 1982 from the University of Birmingham, Birmingham, U.K.

He is currently a Professor in the School of Communication and Information Engineering, Shanghai University, Shanghai, China. He was previously a Research Fellow with the Institute of Acoustics, Chinese Academy of Sciences, from January 1983 to October 1985 and joined Shanghai University of Technology in October 1985 as an Associate Professor. He was an Associate Scientist in the Department of EECS, University of Michigan, Ann Arbor, from March 1993 to August 1994, and a Research Fellow in the Department of Information Systems, City University of Hong Kong, during 1998 and 2002. His research interests include underwater acoustics, image processing, and multimedia security. He has published more than 150 papers in these areas. Many of his research projects are supported by the Natural Science Foundation of China.