Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition 38 (2005) 2519–2529
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

A hierarchical digital watermarking method for image tamper detection and recovery☆

Phen Lan Lin[a,*], Chung-Kai Hsieh[b], Po-Whei Huang[b]

[a]*Department of Computer Science and Information Engineering, Providence University, Shalu, Taichung 433, Taiwan*
[b]*Department of Computer Science, National Chung Hsing University, Taichung, Taiwan*

Received 2 December 2003; received in revised form 21 February 2005; accepted 21 February 2005

Abstract

In this paper, we present an efficient and effective digital watermarking method for image tamper detection and recovery. Our method is efficient as it only uses simple operations such as parity check and comparison between average intensities. It is effective because the detection is based on a hierarchical structure so that the accuracy of tamper localization can be ensured. That is, if a tampered block is not detected in level-1 inspection, it will be detected in level-2 or level-3 inspection with a probability of nearly 1. Our method is also very storage effective, as it only requires a secret key and a public chaotic mixing algorithm to recover a tampered image. The experimental results demonstrate that the precision of tamper detection and localization is 99.6% and 100% after level-2 and level-3 inspection, respectively. The tamper recovery rate is better than 93% for a less than half tampered image. As compared with the method in Celik et al. [IEEE Trans. Image Process. 11(6) (2002) 585], our method is not only as simple and as effective in tamper detection and localization, it also provides with the capability of tamper recovery by trading off the quality of the watermarked images about 5 dB.

Keywords: Image authentication; Tamper detection; Tamper localization; Digital watermarking

1. Introduction

Since the past decade, there has been a rapid growth in using digital multimedia data. The flourish of PC with the wideband Internet connections has made the distribution of multimedia data much easier and faster. On the other hand, the wide availability of powerful image processing tools has also made imperceptible image modifications possible. As a result, image authenticity becomes greatly threatened.

Classical image authentication mechanisms [1–4], sometimes called tamper-proofing mechanisms, attach a signature to the image and verify the received image by comparing its signature to the signature attached to it. The signature can be either an encrypted or a signed hash value of image contents or image characteristics (e.g., edges, histograms, moments, etc.) [5]. These mechanisms can detect if an image has been changed; however, they cannot locate where the image was changed. Besides, the attached signature requires additional bandwidth or storage that may not always be available. To solve these problems, many researchers have proposed watermarking for image authentication [6–17].

Based on the functionality, Vleeschouwer et al. [18] classified the image authentication watermarking techniques into three categories: (1) fragile watermarking, which detects any modification to the image [7,11–13,19]; (2) semi-fragile watermarking, which detects and localizes the modifications to the contents [6–8]; (3) content-based fragile watermarking, which detects only the significant changes in the image

☆ This research is supported by National Science Council of ROC under Grant NSC-92-2213-E-126-009.
* Corresponding author. Tel.: +886-4-632-8001x3409; fax: +886-4-632-4045.
E-mail address: lan@pu.edu.tw (P.L. Lin).

0031-3203/$30.00
doi:10.1016/j.patcog.2005.02.007

while permitting content-preserving processing such as coding and scanning [2,10]. In Ref. [20], the above categories (1) and (2) are combined as fragile watermarking, whereas category (3) is named as semi-fragile watermarking. Some examples of these schemes are presented in Refs. [21–24]. In this paper, we adopt the definitions of fragile and semi-fragile watermarking presented in Ref. [20].

Wong [12] proposed a public-key fragile watermarking scheme that embeds a digital signature of each block within the image into the least significant bits of the same block. However, Holliman and Memon [25] soon presented a vector quantization (VQ) counterfeiting attack that can construct a counterfeit image from a VQ codebook generated from a set of watermarked images. To solve the problem of VQ counterfeiting attack, several enhanced algorithms were proposed [7,25,27]. Nonetheless, they either fail to effectively address the problem or sacrifice tamper localization accuracy of the original methods [20]. Celik et al. then presented an algorithm based on Wong's scheme and demonstrated that their algorithm can thwart the VQ codebook attack while sustaining the localization property [20].

In this paper, we present an efficient and effective digital watermarking method for image tamper detection and recovery. Our method is efficient as it only uses simple operations such as parity check and comparison between average intensities. It is effective because the scheme inspects the image hierarchically with the inspection view increases along with the hierarchy so that the accuracy of tamper localization can be ensured. That is, if a tampered block is not detected in level-1 inspection, it will be detected in level-2 or level-3 inspection with a probability of nearly 1. Our method is also very storage effective, as it only requires a secret key and a public chaotic mixing algorithm to recover the tampered areas. The experimental results demonstrate that the precision of tamper detection and localization is close to 100% after level-2 inspection and the tamper recovery rate is better than 93% for a less than half tampered image. As compared with the method in Ref. [20], our method is not only as simple and as effective in tamper detection and localization, it also provides with the capability of tamper recovery by trading off the quality of the watermarked images about 5 dB.

The remainder of this paper is organized as follows. In Section 2, we briefly describe VQ counterfeiting attacks and Torus automorphism. Our proposed watermarking scheme is presented in Section 3. In Sections 4 and 5, we demonstrate the experimental results and analyze the performance of our scheme. Finally, the concluding remarks are given in Section 6.

2. Backgrounds

In this section, we briefly describe both vector quantization (VQ) counterfeiting attacks and Torus automorphism.

2.1. VQ counterfeiting attacks

Holliman and Memon [25] presented a counterfeiting attack on block-wise independent watermarking schemes. In such an attack, the attacker is able to create a counterfeit image by using a collage of watermarked blocks selected from a large database. Because block-wise independent schemes validate the watermark of each individual block and the watermark of each block comprises only the information of the block itself, the counterfeit image generated from the VQ codebook can easily pass the authentication test of block-wise independent schemes.

2.2. Torus automorphism

Torus automorphism is a kind of dynamic system. Briefly speaking, a dynamic system is a system whose states change with time t. When t is discrete, a dynamic system can be presented as an iteration of a function f, i.e., $S_{t+1} = f(S_t)$, $t \in Z = \{0, 1, 2, \ldots\}$, S_t, S_{t+1} are the states at time t and $t+1$, respectively. A two-dimensional Torus automorphism can be considered as a permutation function or a spatial transformation of a plane region. This transformation can be performed using a 2×2 matrix A with constant elements. More specifically, a state S_{t+1} or a point (x_{i+1}, y_{i+1}) can be transformed from another state S_t or another point (x_i, y_i) by

$$A = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}, \quad \begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = A \times \begin{pmatrix} x_i \\ y_i \end{pmatrix} \bmod N, \qquad (1)$$

where $a_i \in Z$, $|A|$ (the determinant of A) $=1$, and A has eigenvalues $\lambda_{1,2} \in R - \{-1, 0, 1\}$, R is the set of rational numbers. The detailed characteristics of A are described in Refs. [26,28]. A set of successive points $\{S_0, S_1, S_2, \ldots\}$, generated by Eq. (1), composes an orbit ϑ of the system and the initial point $S_0 = (x_0, y_0)$ classifies ϑ into two categories. When x_0 and/or y_0 are irrational, ϑ is infinite. When both x_0 and y_0 are rational, ϑ is chaotic and periodic at every R times, i.e., $S_R = S_0$. R is called the "recurrence time".

Voyatzis and Pitas [26] presented a one-parameter, two-dimensional, discrete Torus automorphism, shown in Eq. (2), for creating a unique and random mapping of the pixels within an image:

$$A = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix}, \quad \begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = A \times \begin{pmatrix} x_i \\ y_i \end{pmatrix} \bmod N, \qquad (2)$$

where $(x_i, y_i) \in [0, N-1] \times [0, N-1]$ and $k \in [0, N-1]$. The recurrence time R depends upon the parameters k, N, and the initial point (x_0, y_0). In most cases, R is equal to $N-1$ or $N+1$, when N is prime [26,28].

3. The proposed scheme

Our proposed scheme can perform both tamper detection and recovery for tampered images. While tamper detec-

Table 1
The mapping of blocks using Eq. (3) with $k = 37$, 6, and $N = 64$

k	X	1	2	3	4	5	6	7	8	33	34	35	36
	(x, y)	(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)	(4,0)	(4,1)	(4,2)	(4,3)
37	X'	38	11	48	21	58	31	4	41	6	43	16	53
	(x', y')	(4,5)	(1,2),	(5,7)	(2,4)	(7,1)	(3,6)	(0,3)	(5,0)	(0,5)	(5,2)	(1,7)	(6,4)
6	X'	7	13	19	25	31	37	43	49	7	13	19	25
	(x', y')	(0,6)	(1,4)	(2,2)	(3,0)	(3,6)	(4,4)	(5,2)	(6,0)	(0,6)	(1,4)	(2,2)	(3,0)

tion is achieved through a simple, block-based, three-level inspection of in-block information only, the recovery of a tampered block relies on its feature information hidden in another block that can be determined by a one-dimensional transformation similar to Torus automorphism, as described in Section 2. Our scheme can be best described through the following three stages: watermark embedding, tamper detection, and tampered image recovery.

3.1. Block-based watermark embedding

We describe the watermark-embedding procedure in this section. Each image is of size $M \times M$ pixels, where M is assumed to be a multiple of four and the number of gray-levels is 256.

3.1.1. Preparation

We need to prepare a block-mapping sequence $A \rightarrow B \rightarrow C \rightarrow D \rightarrow \cdots \rightarrow A$ for watermark embedding, where each symbol denotes an individual block. The intensity feature of block A will be embedded in block B, and the intensity feature of block B will be embedded in block C, etc.

Since the number of blocks in each dimension of most images can hardly be a prime number, we cannot obtain a one-to-one mapping among the blocks by applying Eq. (2), based on the analysis in Ref. [26]. Instead, we use a 1-D transformation as shown in Eq. (3) to obtain a one-to-one mapping sequence.

$$X' = [f(X) = (k \times X) \bmod N] + 1, \tag{3}$$

where X, $X'(\in [0, N-1])$ are the block number, k (a prime and $\in [0, N-1]$) is a secret key, and N ($\in Z - \{0\}$) is the total number of blocks in the image.

The generation algorithm of the block-mapping sequence is as follows.

3.1.2. Block mapping sequence generation algorithm

1. Divide the image into non-overlapping blocks of 4×4 pixels.
2. Assign a unique and consecutive integer $X \in \{0, 1, 2, \ldots, N-1\}$ to each block from left to right and top to bottom, where $N = (M/4) \times (M/4)$.
3. Randomly pick a prime number $k \in [1, N-1]$.
4. For each block number X, apply Eq. (3) to obtain X', the number of its mapping block.
5. Record all pairs of X and X' to form the block mapping sequence.

Note that the secret key k must be a prime in order to obtain a one-to-one mapping; otherwise, the period is less than N and a many-to-one mapping may occur. Table 1 lists some portions of the mapping sequence generated with $N = 64$, $k = 37$ and 6, respectively. In this table, X' starts to repeat at $X = 33$ when $k = 6$, which is not a prime.

3.1.3. Block watermark embedding

For each block B of 4×4 pixels, we further divide it into four sub-blocks of 2×2 pixels, as shown in Fig. 1. The watermark in each sub-block is a 3-tuple (v, p, r), where both v and p are 1-bit authentication watermark, and r is a 6-bit recovery watermark for the corresponding sub-block within block A mapped to B. The following algorithm describes how the 3-tuple watermark of each sub-block is generated and embedded.

3.1.4. Sub-block watermark generation and embedding algorithm

1. Set the two LSBs of each pixel within the block to zero and compute the average intensity of the block and each of its four sub-blocks, denoted by avg_B and avg_B_s, respectively.
2. Generate the authentication watermark v of each sub-block as in Eq. (4) below:

$$v = \begin{cases} 1 & \text{if } avg\_B_s \geqslant avg\_B, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

3. Generate the parity-check bit p of each sub-block as in Eq. (5) below:

$$p = \begin{cases} 1 & \text{if } num \text{ is odd,} \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

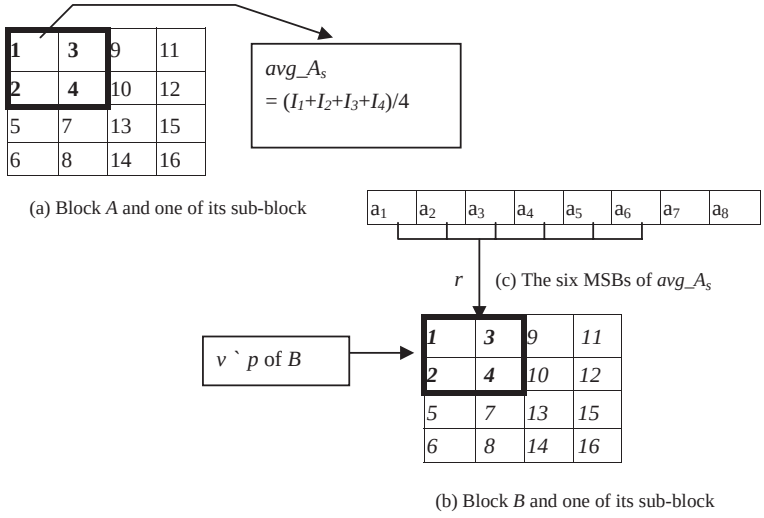

Fig. 1. Watermark generation procedure for the sub-block comprising pixels 1,2,3,4: (a) Block A and one of its sub-block; (b) block B and one of its sub-block; (c) the six MSBs of avg_A.

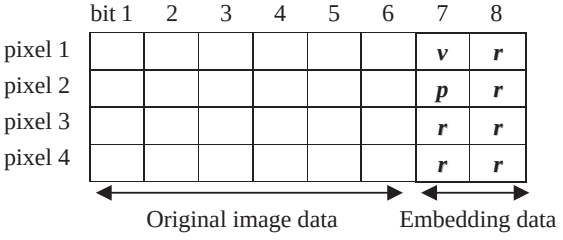

Fig. 2. The 8-bit Watermark (v, p, r) embedded in pixels 1,2,3,4.

where *num* is the total number of 1s in the six MSBs of avg_B_s.

4. From the mapping sequence generated in the preparation step, obtain block A whose recovery information will be stored in block B.
5. Compute the average intensity of each corresponding sub-block A_s within A, as shown in Fig. 1(a), and denote it avg_A_s.
6. Obtain the recovery intensity r of A_s by truncating the two LSBs in avg_A_s, as shown in Fig. 1(c).
7. Embed the 3-tuple-watermark (v, p, r), eight bits in all, onto the two LSBs of each pixel within B_s, as shown in Fig. 2.

The watermark generation procedure for the sub-block comprising pixels 1, 2, 3, and 4 is illustrated in Fig. 1. How the 8-bit (v, p, r) are embedded in these four pixels is shown in Fig. 2.

3.2. Block-based hierarchical tamper detection

The test image is first divided into non-overlapping blocks of 4×4 pixels, as in the watermark embedding process. For each block, denoted as B', we first set the two LSBs of each pixel in B' to zero and compute its average intensity, denoted as avg_B'. We then perform 3-level hierarchical detection. In level-1 detection, we check each 2×2 sub-block within one block. In level-2 detection, we treat a 4×4 block as one unit. Finally, we check the block by extending the inspection view to its 3×3 block-neighborhood in level-3 detection. Level-4 detection is for VQ-attack resiliency only. The procedure of our hierarchical tamper detection scheme is described in the following.

3.2.1. Hierarchical tamper detection algorithm

Level-1 *detection*. For each sub-block B'_s of 2×2 pixels within block B', perform the following steps:

1. Extract v and p from B'_s.
2. Set the two LSBs of each pixel within each B'_s to zero and compute the average intensity for each sub-block B'_s, denoted as $avg\_B'_s$.
3. Count the total number of 1s in $avg\_B'_s$ and denote it N'_s.
4. Set the parity-check bit p' of B'_s to 1 if N'_s is odd; otherwise, set it to 0.
5. Compare p' with p. If they are not equal, mark B'_s erroneous and complete the detection for B'_s.
6. Set the algebraic relation $v' = 1$ if $avg\_B'_s >= avg\_B'$; otherwise, set $v' = 0$.
7. Compare v' with v. If they are not equal, mark B'_s erroneous and complete the detection for B'_s; otherwise, mark it valid.

Level-2 *detection*. For each block of size 4×4 pixels, mark this block erroneous if any of its sub-block is marked erroneous; otherwise, mark it valid.

Error	*B*	Error
Error	Error	Error

Fig. 3. The 3×3 block-neighborhood of block *B*.

Level-3 *detection*. For each valid block of size 4×4 pixels, mark the block erroneous if there are five or more erroneous blocks in its 3×3 block-neighborhood, as shown in Fig. 3.

Level-4 *detection* (*only required for resisting against VQ attack*). For each valid block B' of size 4×4 pixels, perform the following steps:

1. Find the block number of block C using secret key k and Eq. (3). Note that block C is the one in which the intensity feature of block B' is embedded.
2. Locate block C.
3. If block C is marked erroneous, assume block B' is correct and complete the test for block B'.
4. If block C is valid, perform the following steps:
   (a) Obtain the 6-bit should-be intensity of each B'_s by firstly extracting the two LSBs from each pixel in the corresponding sub-block within block C then discarding the bits for v, p and padding two 0s to the end to make an 8-bit value, denoted as $avg\_B1'_s$.
   (b) Compute the average intensity of each B'_s with the two LSBs of each pixel set to zero beforehand and denote it $avg\_B'_s$.
   (c) Compare $avg\_B'_s$ with $avg\_B1'_s$ and mark B' erroneous if they are different.

3.3. Block-based tampered image recovery

After the detection stage, all the blocks are marked either valid or erroneous. We only need to recover the erroneous blocks and leave those valid blocks as they are. The restoration procedure for each erroneous block is described as follows.

For convenience of description, we call the erroneous block under recovery block B and the block embedded with its intensity block C.

3.3.1. Block recovery algorithm

1. Calculate the block number for block C using secret key k and Eq. (3).
2. Locate block C.
3. If block C is marked erroneous, skip the recovery for this block.
4. If block C is valid, obtain the 6-bit-intensity of each sub-block within block B by firstly extracting the two LSBs from each pixel in the corresponding sub-block within block C then discarding the bits v and p.
5. Pad the 6-bit-intensity with two 0s to the end and replace the intensity of each pixel within the sub-block with this new 8-bit intensity.
6. Repeat step 5 for all four sub-blocks within block B and mark block B valid.

4. Experimental results

We conduct three experiments to test the performance of our proposed scheme on both tamper detection and tampered image restoration.

4.1. Performance on images with slight to median degree of tampering

We use four images: Home, Car, Fingerprint, and Beach, as shown in Figs. 4(a)–7(a), for this experiment. All are of size 256×256 pixels. The experiments and their respective results are described in the following.

(a) The number on the doorplate in the watermarked Home has been modified, as shown in Fig. 4(b). The size of the modification is about 4×12. Fig. 4(c) shows the detected tamper area using level-1 detection, and Fig. 4(d) shows the recovered Home.

(b) The number on the license plate in the watermarked Car has been wiped out, as shown in Fig. 5(b). The size of the modification is about 68×20. Figs. 5(c) and (e) show the detected tamper area using hierarchical level-1 and -2 inspections, and Figs. 5(d) and (f) show the recovered Car, respectively.

(c) The watermarked Fingerprint, as shown in Fig. 6(a), has been slightly and randomly modified, as shown in

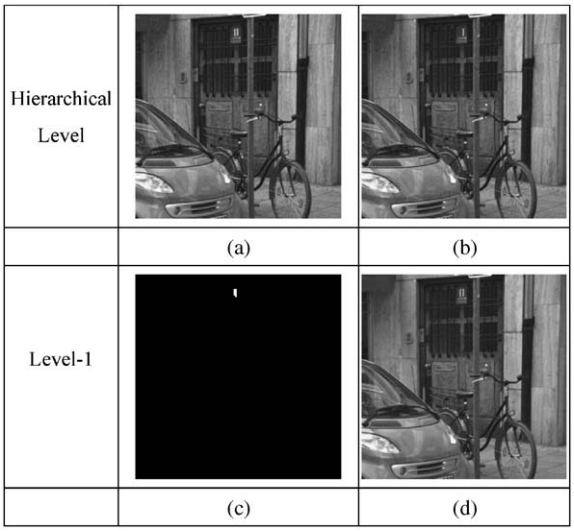

Fig. 4. (a) The watermarked Home; (b) tampered Home; (c) detected erroneous region; (d) recovered Home.

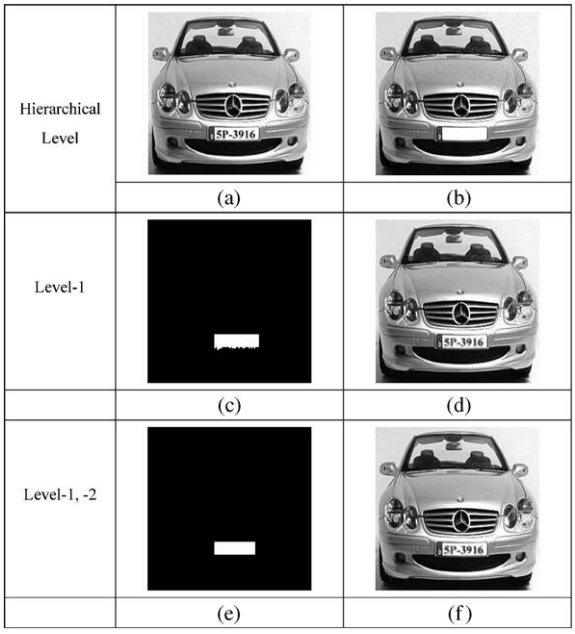

Fig. 5. (a) The watermarked Car; (b) tampered Car; (c,e) detected erroneous region; (d,f) recovered Car.

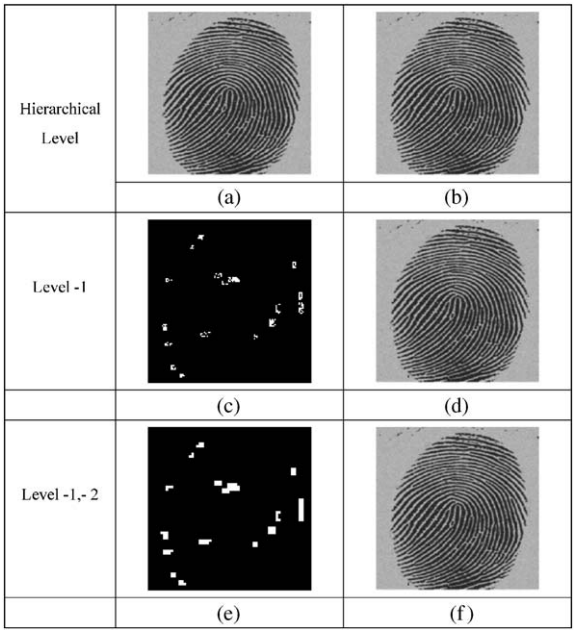

Fig. 6. (a) The watermarked Fingerprint; (b) tampered Fingerprint; (c,e) detected erroneous regions; (d,f) recovered Fingerprints.

Fig. 6(b). Comparing Figs. 6(a) with (b), we can hardly differentiate one from the other. Using our method, we can detect the modifications, as shown in Figs. 6(c) and (e) with only level-1 detection, or both level-1 and -2 detections, respectively. The size of the smallest modification is about

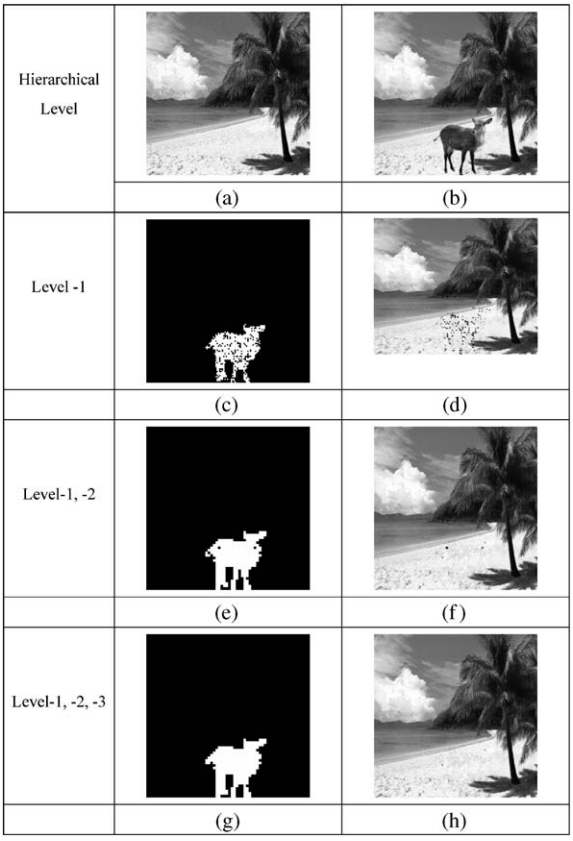

Fig. 7. The watermarked Beach; (b) tampered Beach; (c,e,g) detected erroneous regions; (d,f,h) recovered Beaches.

8×8 pixels. Figs. 6(d) and (f) show the recovered Fingerprints, respectively.

(d) The watermarked Beach, as shown in Fig. 7(a), has been inserted with a deer, as shown in Fig. 7(b). The modification is so natural that one can hardly suspect any changes that had been made to this picture. Using our method, we can detect the modification, as shown in Figs. 7(c), (e), and (g) with only level-1, both level-1 and -2, or all three levels of detection, respectively. The size of the modification is about 68×68 pixels. Figs. 7(d), (f), and (h) show the recovered Beaches, respectively.

4.2. Performance of tamper detection on 100% tampered images

We test the tamper detection capability for large tampered area with image Lena, as shown in Fig. 8(a). All blocks in image Lena are 100% altered by the four following methods.

M1: Covering the whole image with leaf patterns, as depicted in Fig. 8(b).

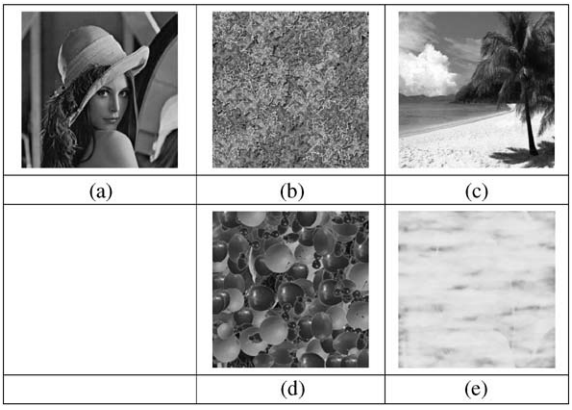

Fig. 8. (a) Unaltered image; (b–e) All blocks in (a) are 100% altered by method M1–M4, respectively.

Table 2
The miss detection rates after each level of inspection

Image	Home	Car	Fingerprint	Beach
Level-1	11.62%	10.58%	11.84%	11.56%
Level-2	0.35%	0.37%	0.32%	0.34%
Level-3	0%	0%	0%	0%

M2: Covering the whole image with another image, as depicted in Fig. 8(c).
M3: Covering the whole image with fruit patterns, as depicted in Fig. 8(d).
M4: Spreading plenty of mist to the whole image, as depicted in Fig. 8(e).

The test results are listed in Table 2 and summarized as follows:

(a) With only level-1 detection, the maximum rate of miss detection is less than 12%.

(b) With both level-1 and level-2 detections, the maximum miss detection rate drops significantly to 0.37%.

(c) With level-1, -2, and -3 detections, all tampered blocks are detected.

4.3. Performance of restoration on distribution of tampered blocks

In this experiment, we want to find out how well a tampered image can be recovered with respect to different sizes and distributions of tampering. The tampered blocks can be in a form of either single tampered chunk or spread tampered blocks. We altered image Lena 10–50% in total with both types of tampering distribution, as shown in Fig. 9.

The recovering rates of both types of tampering distribution with respect to the rate of tampering are depicted in Figs. 10 and 11, and can be summarized as follows.

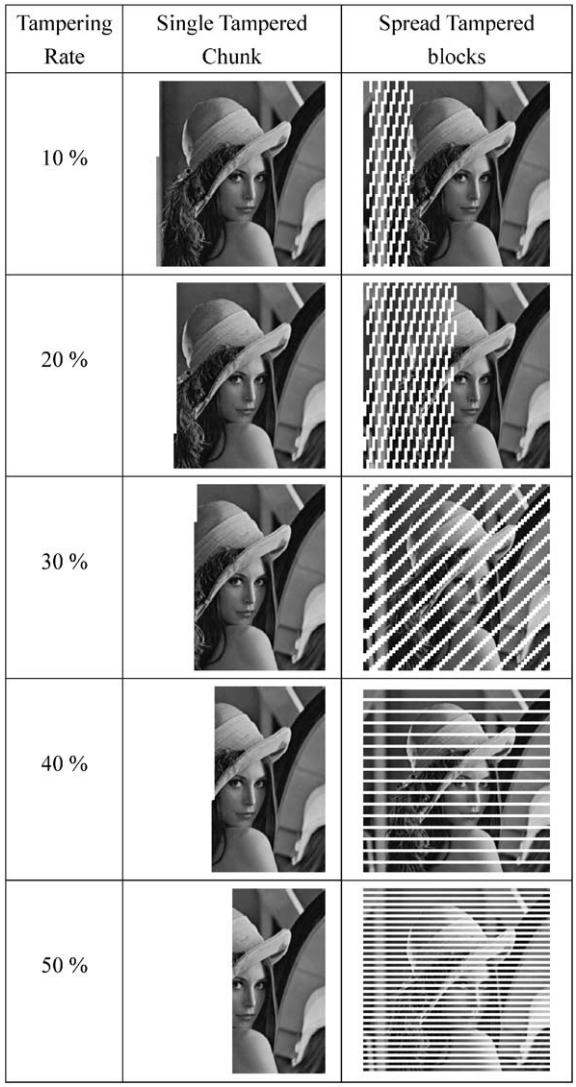

Fig. 9. 10–50% Tampered Lena with two types of tampering distribution: single-tampered-chunk and spread-tampered-blocks.

(a) The recovering rate for single-tampered images is better than 99% when the rate of tampering is less than 16%, is about 94% when the rate of tampering is 40%, and is about 93% when the rate of tampering increases to 50%.

(b) The recovering rate for spread-tampered images is better than 99% when the rate of tampering is less than 30%, is about 96% when the rate of tampering is 40%, and is about 94% when the rate of tampering increases to 50%.

(c) The recovering rate for spread-tampered images is almost twice as better as that for single-tampered images when the rate of tampering is less than 30%, is 30% better when the rate of tampering is between 30% and 50%, and is 20% better when the rate of tampering increases to 50%.

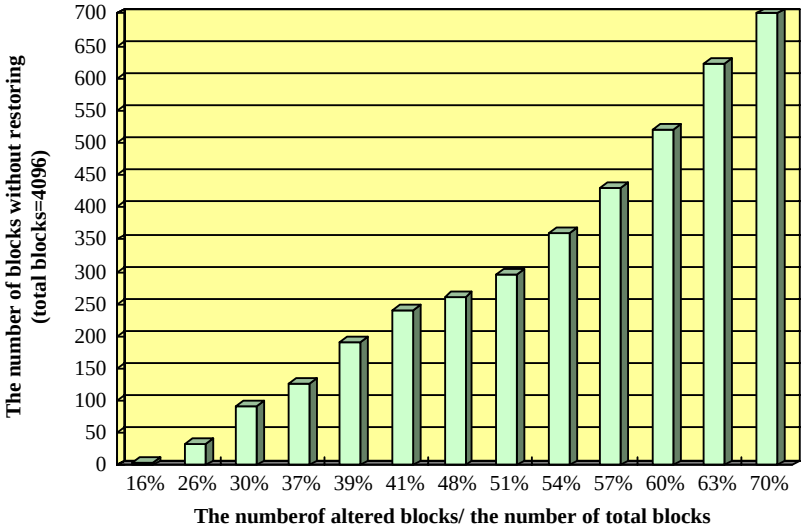

Fig. 10. The number of un-recovered blocks for single-tampered-chunk.

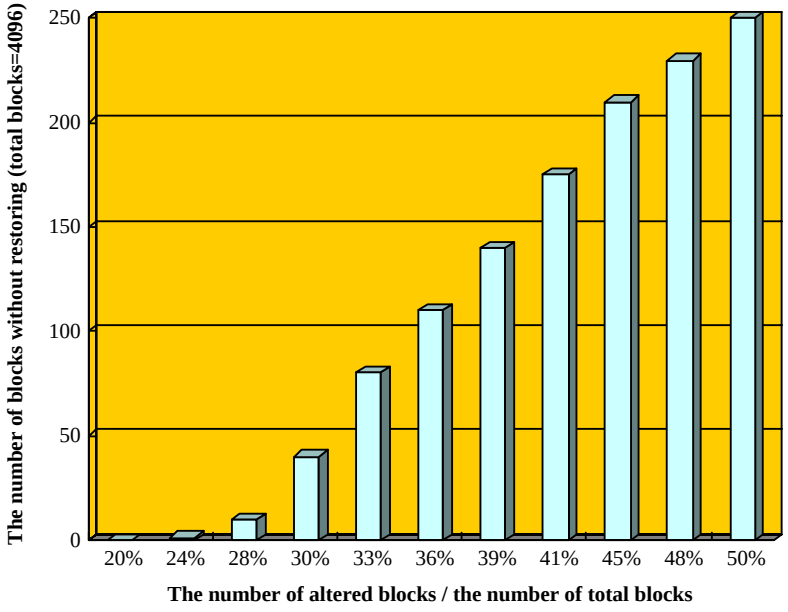

Fig. 11. The number of un-recovered blocks for spread-tampered-blocks.

5. Performance analyses

In this section, we analyze the performance of our method from aspects of the fidelity of watermarked images, the quality of recovered images, as well as the tamper detection rate.

5.1. Image quality analysis

We use *PSNR* defined in Eq. (6) as the indicator of the quality of both watermarked images and recovered images.

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE} \text{ dB}, \tag{6}$$

where $MSE = (1/m \times n) \sum_{i=0}^{i<n} \sum_{j=0}^{j<m} (c'_{ij} - c_{ij})^2$, m, n are the width and the length of the image, respectively, and c, c' are the pixel intensities of the two images under comparison.

(a) *PSNR of the watermarked image.* Since only two LSBs of each pixel are used for embedding the watermark, the

Table 3
The *PSNR*s of the watermarked images

Image	Home	Fingerprint	Car	Beach
PSNR (in dB)	44.36	44.36	44.39	44.37

*PSNR*s of the watermarked images are all better than 44 dB, as listed in Table 3. As compared with the scheme in Ref. [20] that only uses one LSB to embed the watermark, the quality of the images watermarked by our scheme is degraded about 5–6 dB.

(b) *PSNR of the recovered image.* The *PSNR* of a recovered image depends how well the tampered blocks are recovered. The recovery depends on how accurately those tampered blocks are detected. And, the tampering detection depends on both the size and the distribution of tampering. Table 4 lists the *PSNR* of five recovered images with various sizes and distributions of tampering. From this table, we find that among all the single tampered images, Home has the best *PSNR* of 48.48 dB, which is recovered from a tampered chunk of 4×12 pixels. Car-1 has the second best *PSNR* of 38.12 dB, which is recovered from a tampered chunk of size 24×16. Car-2 has *PSNR* of 32.52 dB, which is recovered from a tampered chunk of size 68×20. Beach has the worst *PSNR* of 30.85 dB, which is recovered from a tampered chunk of size 68×68. For spread tampering, Fingerprint has *PSNR* of 36.46 dB, which is recovered from a total tampered area of size 70×16. Comparing the results of Fingerprint and Car-2, we find that the *PSNR* of Fingerprint is much better than that of Car-2, even though the sizes of total tampering for both images are not much different.

5.2. Tamper detection analysis

We analyze the detection algorithm from the following aspects:

(a) *Level*-1 *detection*: If the type of error is parity error, then we are sure that the sub-block is indeed erroneous.

(b) *Level*-2 *detection*: If the type of error is intensity-relationship error, we cannot be sure whether the sub-block under inspection is erroneous or other sub-blocks within the same block are erroneous. However, some pixels within the block must be wrong. Thus, in case the tampered sub-block is not detected in level-1 inspection, the whole block will be marked erroneous after level-2 inspection.

The probability of failing both the parity check and the intensity-relation check simultaneously is at most $(1/2) \times (1/2) = 1/4$. So the probability of false detection for all four sub-blocks will be at most $(1/4)^4 \cong 0.39\%$, based on the production rule of statistics. Thus, almost all erroneous blocks can be detected after level-2 inspection.

(c) *Level*-3 *detection*: From (b), we know that the probability of failing to detect one tampered block after level-2 detection is at most 1/256. Thus, the miss detection rate after level-3 detection is the probability of failing to detect more than four tampered blocks within the 3×3-block-neighborhood, which is less than $C(9,4)(1/256)^4 \ll 1/2^{25} \cong 0$.

From the experimental results in Table 2, we find that the maximum missing rate after level-1, -2, and -3 detection is 12%, 0.37%, and 0%, respectively, which are quite consistent with the calculated probability derived for each level. Thus, we summarize the detection capability of our method as follows:

- Our method can detect any tampering of size 12×12 pixels or bigger without a single miss.
- Our method can detect any tampering of size 4×4 pixels with a missing rate less than 0.37%.
- Our method can detect any tampering of size 2×2 pixels with a missing rate less than 12%.

6. Conclusion

We have presented a hierarchical digital watermarking scheme for both image tamper detection and restoration in this paper. Our scheme is efficient in execution time since it only uses simple operations such as parity check and comparisons. It is effective in detection accuracy since it inspects the image hierarchically with the inspection view grows along with the hierarchy. It is also very storage effective since it only requires a secret key and a public chaotic mixing algorithm for both tamper detection and recovery. The experimental results demonstrate that the precision of tamper detection and localization is close to 100% after level-2 inspection and the tamper recovery rate is better than 93% for a less than half tampered image. As compared with the method in Ref. [20], our method is not only as simple and as effective in tamper detection and localization, it also provides with the capability of tamper recovery by trading off the quality of the watermarked images about 5 dB. Our

Table 4
The *PSNR*s of the recovered images relative to the watermarked images

Image	Home	Fingerprint	Car-1	Car-2	Beach
Size of tamper	4×12	70×16	24×16	68×20	68×68
PSNR (in dB)	48.48	36.46	38.12	32.52	30.85

future research includes improvements on both security and watermark-payload reduction while maintaining the capability of both tamper detection and recovery.

References

[1] G.I. Friedman, The trustworthy digital camera: restoring credibility to the photographic image, in: Proceedings of the IEEE International Conference on Image Processing, vol. II, Chicago, IL, USA, October 1998, pp. 409–413.

[2] J. Dittmann, A. Steinmetz, R. Steinmetz, Content-based digital signature for motion pictures authentication and content-fragile watermarking, in: Proceedings of the IEEE International Conference on Multimedia Computing Systems, 1999, pp. 209–213.

[3] C.Y. Lin, S.F. Chang, A robust image authentication method surviving JPEG lossy compression, Proc. SPIE 3312 (1998) 296–307.

[4] C.S. Lu, H.Y.M. Liao, C.J. Sze, Structural digital signature for image authentication: an incidental distortion resistant scheme, in: Proceedings of the Multimedia Security Workshop 8th ACM International Conference on Multimedia, Los Angeles, CA, 2000, pp. 115–118.

[5] W. Stallings, Network Security Essentials: Application and Standards, Prentice-Hall, Upper Saddle River, NJ, 2000.

[6] J. Fridrich, Image watermarking for tamper detection, in: Proceedings of the IEEE International Conference on Image Processing, vol. II, Chicago, IL, USA, October 1998, pp. 404–408.

[7] J. Fridrich, M. Goljan, A.C. Baldoza, New fragile authentication watermark for images, in: Proceedings of the IEEE International Conference on Image Processing, Vancouver, BC, Canada, September 2000, pp. 10–13.

[8] D. Kundur, D. Hatzinakos, Toward a telltale watermarking technique for tamper-proofing, in: Proceedings of the IEEE International Conference on Image Processing, vol. 2, 1998, pp. 409–413.

[9] E. Lin, E.J. Delp, A review of fragile image watermarks, in: Proceedings of the ACM Multimedia and Security Workshop, Orlando, FL, November 1999, pp. 35–40.

[10] M.P. Queluz, Content-based integrity protection of digital image, Proceedings of the SPIE, Security and Watermarking of Multimedia Contents II, vol. 3971, Bellingham, WA, 2000, pp. 85–93.

[11] M. Schneider, S.F. Chang, A robust content based digital signature for image authentication, in: Proceedings of the IEEE International Conference on Image Processing, vol. III, Lausanne, Switzerland, September 1996, pp. 227–230.

[12] P.W. Wong, A public key watermark for image verification and authentication, in: Proceedings of the IEEE International Conference on Image Processing, Chicago, IL, October 1998, pp. 455–459.

[13] M.M. Yeung, F. Mintizer, An invisible watermarking technique for image verification, in: Proceedings of the IEEE International Conference on Image Processing, vol. I, Santa Barbara, CA, October 1997, pp. 680–683.

[14] R.L. Lagendijk, G.C. Langelaar, I. Setyawan, Watermarking digital images and video data, IEEE Signal Process. Mag. 17 (2000) 20–46.

[15] F. Hartung, M. Kutter, Multimedia watermarking techniques, Proc. IEEE 87 (1999) 1079–1107.

[16] M.D. Swanson, M. Kobayashi, A.H. Tewfik, Multimedia data embedding and watermarking technologies, Proc. IEEE 86 (1988) 1064–1087.

[17] I.J. Cox, M.I. Miller, A.L. Mckellips, Watermarking as communications with side information, Proc. IEEE 87 (1999) 1127–1147.

[18] C.De. Vleeschouwer, J.F. Delaigle, B. Macq, Invisibility and application functionalities in perceptual watermarking—an overview, Proc. IEEE 90 (1) (2002) 64–77.

[19] C.W. Wu, D. Coppersmith, F.C. Mintizer, C.P. Tresser, M.M. Mueng, Fragile imperceptible digital watermark and privacy control, Proceedings of the SPIE, Security and Watermarking of Multimedia Contents, vol. 3657, January 1999, pp.79–84.

[20] M.U. Celik, G. Sharma, E. Saber, A.M. Tekalp, Hierarchical watermarking for secure image authentication with localization, IEEE Trans. Image Process. 11 (6) (2002) 585–594.

[21] D. Kundur, D. Hatzinakos, Digital watermarking for telltale tamper proofing and authentication, Proc. IEEE 87 (1999) 1167–1180.

[22] J. Eggers, B. Girod, Blind watermarking applied to image authentication, in: Proceedings of the IEEE ICASSP, Salt Lake City, UT, May 2001, pp. 1977–1980.

[23] S. Bhattacharjee, M. Kutter, Compression tolerant image authentication, in: Proceedings of the IEEE International Conference on Image Processing, Chicago, IL, October 1998, pp. 435–439.

[24] C.S. Lu, H.Y. Liao, Multipurpose watermarking for image authentication and protection, IEEE Trans. Image Process. 10 (2001) 1579–1592.

[25] M. Holliman, N. Memon, Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes, IEEE Trans. Image Process. 9 (2000) 432–441.

[26] G. Voyatzis, l. Pitas, Chaotic mixing of digital images and applications to watermarking, in: Proceedings of the European Conference on Multimedia Applications Services and Techniques (ECMAST'96), 1996, pp. 687–689.

[27] P.W. Wong, N. Memon, Secret and public key authentication schemes that resist vector quantization attack, Proc. SPIE 3971 (75) (2002) 417–427.

[28] G. Voyatzis, I. Pitas, Applications of toral automorphisms in image watermarking, in: Proceedings of the International Conference on Image Processing, vol. II, 1996, pp. 237–240.

About the Author—**PHEN LAN LIN** received the B.S. degree in Engineering Science from National Cheng-Kung University, Taiwan in 1973, the M.S. degree in Mathematics from Texas Tech University, Texas in 1978, and both the M.S. and Ph.D. degrees in Electrical Engineering from Southern Methodist University, Dallas, Texas in 1992 and 1994, respectively.

Dr. Lin was with Texas Instruments (TI) in Dallas, Texas as a member of technical staff, lead engineer, and project manager from 1978 to 1992. Her working experiences include control and computer vision software development, mobile robot navigation, as well as automatic IC inspection project management.
She has been a professor in the Department of Computer Science and Information Management, Providence University, Taiwan since 2001 and is a professor in the Department of Computer Science and Information Engineering. She has also been serving as the Director of Computer and Communication Center in the university since 2001.
Her current research interests are in the fields of multimedia security, network security, medical imaging, and visual inspection.

About the Author—CHUNG-KAI HSIEH received the B.S. degree in Computer Science and Information Management from Providence University, Taiwan in 2002 and the M.S. degree in Computer Science from Chung Hsing University, Taiwan in 2004.
His research interests are in the fields of multimedia and network security.

About the Author—PO-WHEI HUANG received the B.S. degree in Applied Mathematics from National Chung-Hsing University in 1973, the MS degree in mathematics from Texas Tech University in 1978, and the Ph.D. degree in Computer Science and Engineering from Southern Methodist University in 1989.
He was with Texas Instruments in Dallas as a member of technical staff, supervisor, lead engineer, section manager, and software development manager from 1978 to 1991. His working experiences include bubble memory testing and application, IC design automation, Lisp machine on a chip design, automatic program generation, production planning and scheduling, and wafer factory automation.
He was the department head and has been a professor in the Computer Science Department at National Chung-Hsing University. Since September 2002, he has also been serving as the vice president of National Huwei University of Science and Technology located in Yunlin County of Taiwan.
His current research interests are in the fields of multimedia database, medical imaging, visual inspection, pattern recognition, and artificial intelligence.