Hindawi Publishing Corporation
EURASIP Journal on Information Security
Volume 2010, Article ID 829516, 7 pages
doi:10.1155/2010/829516

Research Article

Logistic Map-Based Fragile Watermarking for Pixel Level Tamper Detection and Resistance

Shan Suthaharan

Department of Computer Science, University of North Carolina at Greensboro, Greensboro, NC 27402, USA

Correspondence should be addressed to Shan Suthaharan, ssuthaharan@uncg.edu

Received 21 April 2010; Revised 7 August 2010; Accepted 23 September 2010

Academic Editor: Miroslav Goljan

An efficient fragile image watermarking technique for pixel level tamper detection and resistance is proposed. It uses *five* most significant bits of the pixels to generate watermark bits and embeds them in the *three* least significant bits. The proposed technique uses a logistic map and takes advantage of its sensitivity property to a small change in the initial condition. At the same time, it incorporates the confusion/diffusion and hashing techniques used in many cryptographic systems to resist tampering at pixel level as well as at block level. This paper also presents two new approaches called nonaggressive and aggressive tamper detection algorithms. Simulations show that the proposed technique can provide more than 99.39% tamper detection capability with less than 2.31% false-positive detection and less than 0.61% false-negative detection responses.

1. Introduction

Fragile image watermarking has been proposed to authenticate digital images and detect tampering. In general, watermarks are generated from the most significant bits (MSB) and then embedded into the least significant bits (LSB) of the pixels while maintaining the image quality. The fragile image watermarking schemes also provide a procedure (or rules) to detect tampering at pixel level or block level. They assume the attacker's goal is not to change the watermark in the LSBs but to modify the MSBs so that the watermark generation algorithm produces the same watermarks. As image quality is also of paramount importance to fragile image watermarking schemes, more MSBs are used for watermark generation than the number of LSBs used for watermark embedding. It leads to many-to-one mapping between MSBs and LSBs and provides the attackers an opportunity to have more alternative MSBs for the same watermark, and this makes the watermarking fragile.

One of the original fragile image watermarking techniques was proposed in 1995 by Walton [1], and it is a 7MSB:1LSB technique (i.e., it uses 7 MSBs for watermark generation and 1 LSB for watermark embedding). It is a blockwise technique and it cannot detect pixel-level tampering. This drawback is called a localization problem and it was reported by Fridrich in 2002 [2]. Subsequently, fragile watermarking techniques have been developed to address localization problem [3–5]. Recently, Zhang and Wang proposed two related 5MSB:3LSB fragile watermarking techniques [6, 7]. The first method is a statistical technique which is capable of detecting pixel-level tampering if the tampered area is small. The second one improves the tamper detection capability for a larger area by incorporating a hybrid (blockwise and pixelwise) mechanism. However, the use of block information reduces its tamper resistance capability. At the same time, chaotic map-based fragile watermark has been proposed to improve both tamper resistance and detection capabilities. In [8], the difference between the image and a chaotic map is used along with pixel pairs for watermark generation and embedding. It is a 7MSB:1LSB technique and enhances tamper resistance more than tamper detection (in this scheme, the localization is restricted to pixel pairs). In [9], a blockwise 7MSB:1LSB fragile watermarking technique is proposed using two chaotic maps, one to select pixel locations for embedding and the other to generate watermarks. In this scheme, the localization is restricted to 2×2 pixel blocks. In [10], a composite chaotic iterative function is used along with a 7-MSB seed value to generate chaotic sequence. The chaotic sequence is used to choose a bit for watermark from the 7 MSBs. It is

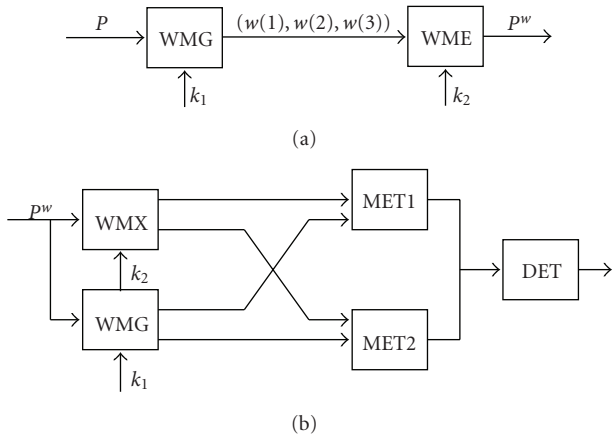

Figure 1: (a) Proposed fragile watermarking generation and embedding process and modules; (b) Proposed tamper detection process and modules.

also a 7MSB:1LSB technique and provides pixel-level tamper detection with the detection rate of only 50%.

This paper proposes a 5MSB:3LSB technique that uses logistic map and confusion/diffusion and hashing cryptographic techniques to improve tamper detection and resistance capabilities. It also uses a key and pseudorandom number generator to select secure pixels for watermark embedding. The confusion process is used to induce complexity in the relation map between the distribution of the watermark and the value of the user-defined key. Similarly, the diffusion process is used to dissipate the image property displayed in the 5MSBs of a pixel over a long-range statistics of the watermark using the logistic-map. The chaotic behavior of the logistic-map is also used as hashing mechanism to generate a set of 3-bits watermarks. This proposed technique is discussed in Section 2. This paper also proposes new concepts called nonaggressive (N_gTDM) and aggressive (A_gTDM) scenarios for efficient tamper detection and these scenarios are presented in Section 3 of this paper. Simulation results and findings are presented in Section 4. A conclusion is also presented in Acknowledgement.

2. Proposed Watermarking Scheme

Figure 1 presents a fragile watermarking scheme. Figure 1(a) shows modules, namely, watermarks generation (WMG) and watermarks embedding (WME). In this scheme, WMG generates three-bit watermark from the 5MSBs of each pixel using a logistic map and a key (k_1). Then, WME pseudorandomly embeds this watermark into the 3LSBs by selecting pixels using a key (k_2). Figure 1(b) shows four more modules (and WMG): watermark extraction (WMX), tamper detection using nonaggressiveness (MET1), tamper detection using aggressiveness (MET2), and final detection (DET). WMX extracts the watermarks in the LSBs of an input image using the key (k_2).

In parallel, WMG generates 3-bit watermark from the 5MSBs of each pixel using the same logistic map and the key k_1. The modules MET1 and MET2 use the watermarks (extracted and generated) to detect tampered pixels. MET1 uses a nonaggressive approach; hence, it is possible to have false-negative pixels in the detection results. A tampered pixel that is detected as not-tampered is called false-negative pixel. MET2 uses an aggressive approach and it artificially increases false-positive pixels in the detection results. A not-tampered pixel that is detected as tampered is called false-positive pixel. Combining these two detection results, the DET detects tampered pixels with very high accuracy.

2.1. Watermark Generation and Embedding Process. Suppose O is the original image and P is its exact copy image. The copy image P is used in the process of generating 3-bit watermarks and the 3LSB of the original image O is replaced with the watermark. Now suppose P contains $8N$ pixels and the gray value of its (i,j)th pixel is denoted by p_{ij}, where $i = 1 \cdots N_1$, $j = 1 \cdots N_2, N_1 \cdot N_2 = 8N$, and $p_{i,j} \in [0, 255]$. Let us denote the bits of the pixel p_{ij} by $(p_{ij}(8), p_{ij}(7), p_{ij}(6), p_{ij}(5), p_{ij}(4), p_{ij}(3), p_{ij}(2), p_{ij}(1))$; then the first 5 bits (left to right) are the most significant bits (MSB) and the last three bits are the least significant bits (LSB). Using the modules in Figure 1(a), the proposed watermark generation and embedding processes are explained step by step in this section.

Step 1. In homogeneous image regions, it is likely high that the majority of the pixels have the same intensity values and hence the majority of the fragile watermarks generated in that region will be the same too. To address this problem, the bits $p_{ij}(4)$ of the copy image P are pseudorandomly modified using the key k_1. Modifying this low-significant bit of the MSB will help to generate fragile watermarks that differ between homogeneous regions with respect to their intensity values. It is important to note that the homogeneous regions are not identified separately and treated differently in this step; instead the modification of the 4th bits of all the pixels in the image takes care of the stated security problem associated with the homogeneous regions.

Step 2. In this step, LSBs of the gray values $p_{ij}(i \cdot j = 1 \cdots 8N)$ are replaced by bits 0 so that the watermarks to be generated for each pixel depend only on the MSBs of the corresponding pixel. Let us denote the modified image as Q and its gray value as q_{ij}, where $q_{ij} = (p_{ij}(8), p_{ij}(7), p_{ij}(6), p_{ij}(5), p_{ij}(4), 0, 0, 0)$.

Step 3. A sequence of $M+2$ integers are generated for (i, j)th pixel using $q_{ij}/248$ as an initial value in the logistic map in (1) [11] until they pass a randomness test. In this equation, $x_{ij}(0) = q_{ij}/248$ and $k = 0 \cdots M + 1$. The nonlinearity parameter value of 4 is selected to achieve a maximum chaos in the map.

$$x_{ij}(k+1) = 4 \cdot x_{ij}(k) \cdot \Big(1 - x_{ij}(k)\Big). \tag{1}$$

Hence we have $8N(= N_1 \cdot N_2)$ integers, $x_{ij}(k)$, for each k and the integers $x_{ij}(k)$ $i = 1 \cdots N_1$, $j = 1 \cdots N_2$ pass the runs test used in [12] for all $k = M - 1, M, M + 1$. These chaotic unpredictable integers will be used to generate

3 images $c_{ij}(M-1)$, $c_{ij}(M)$, and $c_{ij}(M+1)$ according to the following round function:

$$c_{ij}(k) = \text{round}\Big(255 \cdot x_{ij}(k)\Big). \tag{2}$$

Although the runs test shows unpredictability around $M = 16$, a large value (i.e., $M = 81$) is arbitrarily selected to guarantee the unpredictability for many images. From the corresponding pixels in these three images the bits for the intermediate watermark are generated. This process provides a hashing mechanism to the proposed approach. The hashing can be done in many ways but in the proposed approach we select the 6th bit (and 3rd bit in Step 5) of the ijth pixel of the three images to map the corresponding 5MSBs that were used to generate these three images. Hence if the attacker wants to tamper the image he or she should maintain the pattern by modifying the image such that it gives the same bits at these bit positions for the three images. We selected 3rd and 6th bits because they form patterns based on the following: (i) they divide the bit positions equally, that is, 1 and 2; 4 and 5, and 7 and 8, (ii) the 3rd bit and below are used for watermarking, and (iii) the 6th bit and above are used for image understanding [13]. The intermediate watermark bits are denoted by $d_{ij}(1)$, $d_{ij}(2)$, and $d_{ij}(3)$.

Step 4. A key k_1 is used to pseudorandomly permute the pixels of image Q. The permutated image shows a noise-like structure. The noise-like image is divided into 8 partitions such that each partition has the same number of pixels. The LSBs of the pixels in partition 1 are replaced with bit (0,0,0); the LSBs of the pixels in partition 2 are replaced with bit (0,0,1), and so on. Using the same key, the permutation pixels are reassigned to their original positions. This process assigns the three bits of either (0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0), or (1,1,1) to the LSBs with equal probability. The pixels of this modified image R are denoted by r_{ij}.

Step 5. As per Step 3, integers are generated using $r_{ij}/255$ as an initial value in the following equation:

$$y_{ij}(k+1) = 4 \cdot y_{ij}(k) \cdot \Big(1 - y_{ij}(k)\Big), \tag{3}$$

where $y_{ij}(0) = r_{ij}/255$ and $k = 0 \cdots 81$. Using this logistic map sequence, their corresponding gray values are generated using the following round function as earlier:

$$e_{ij}(k) = \text{round}\Big(255 \cdot y_{ij}(k)\Big). \tag{4}$$

Three consecutive integers $e_{ij}(80)$, $e_{ij}(81)$, and $e_{ij}(82)$ are selected from this chaotic sequence. Using these three integers, their 3rd bits are selected to obtain 3 bits $f_{ij}(1)$, $f_{ij}(2)$, and $f_{ij}(3)$.

Step 6. In this step, the final watermark ($w_{ij}(1)$, $w_{ij}(2)$, $w_{ij}(3)$) is generated as a combination of the intermediate watermarks ($d_{ij}(1)$, $d_{ij}(2)$, $d_{ij}(3)$) and ($f_{ij}(1)$, $f_{ij}(2)$, $f_{ij}(3)$) as follows:

$$\begin{aligned} w_{ij}(1) &= d_{ij}(1) \oplus f_{ij}(1), \\ w_{ij}(2) &= d_{ij}(2) \oplus f_{ij}(2), \\ w_{ij}(3) &= d_{ij}(3) \oplus f_{ij}(3). \end{aligned} \tag{5}$$

The logical operator $\oplus$ represents the exclusive OR. Steps 1 through to 6 provide the watermark generation process in WMG module and they provide fragile watermark through confusion, diffusion, and hashing cryptographic techniques. (5) gives us 3 watermark bit planes $w(1)$, $w(2)$, and $w(3)$ where (i, j)th bit of $w(l)$ is $w_{ij}(l)$ and $l = 1, 2, 3$. We have $24N$ bits in all and these bits will be pseudorandomly mixed using the key k_2 and embedded into the 3 LSBs of the original image O. This will provide additional cryptographic strength via this extra confusion process. The watermarked image of O is denoted by P^w and its (i, j)th pixel is denoted by p^w_{ij}.

2.2. Complexity and Tamper Resistance. Tamper resistance measures the difficulty level of finding another pixel for a pixel within the same block, another homogeneous region for a homogeneous region in the same image, or another block for a block in the same image (or a different image) for replacement (tampering) by the attacker without altering the watermark while maintaining the quality of the image. Hence we quantify the tamper resistance as the probability (let us denote it by TRP in this paper) of failure of finding replacement pixels or blocks by random guess or systematic approach by an attacker. Thus, Step 1 is carried out to resist tampering in homogeneous regions; Step 2 is carried out to make the 3 LSB-bit watermark dependent on the 5MSB-bits and fragile; Step 3 is carried out to make the fragile watermarks independent of each other; Steps 4 and 5 are to eliminate the statistical property of the image present in the watermark; Step 6 is carried out to resist tampering in any part of the image. It means that the statistical properties, such as the frequency count of similar blocks or regions and local variance and mean of a block or an image region should not be noticeable in the watermark, because it will help the attacker to guess the pixel, image block, or image region for tampering.

2.3. Pixel-Level Tamper Detection. Figure 1(b) shows the process of pixel-level tamper detection. The module WMX accepts a watermarked image P^w and extracts the watermarks $w_{ij}(1)$, $w_{ij}(2)$, and $w_{ij}(3)$ from the 3LSBs using the same key k_2 and pseudorandom number generator used in watermark embedding. In parallel, the module WMG generates watermark bits $w'_{ij}(1)$, $w'_{ij}(2)$, $w'_{ij}(3)$ from the 5MSBs of P^w, as explained earlier, using the key k_1. These watermarks (extracted and generated) are used in MET1 and MET2 modules. They use N_gTDM and A_gTDM methods, respectively, to detect tampered pixels. The nonaggressive tamper detection approach N_gTDM is defined as follows:

$$W_{ij} = W_{ij}(1) \mid W_{ij}(2) \mid W_{ij}(3). \tag{6}$$

The logical operator "|" represents the logical OR. The operand bits $W_{ij}(1)$, $W_{ij}(2)$, and $W_{ij}(3)$ are defined by

$$W_{ij}(1) = w_{ij}(1) \oplus w'_{ij}(1),$$
$$W_{ij}(2) = w_{ij}(2) \oplus w'_{ij}(2), \quad (7)$$
$$W_{ij}(3) = w_{ij}(3) \oplus w'_{ij}(3).$$

If we denote $D_{ij} = (w_{ij}(1)w'_{ij}(1)w_{ij}(2)w'_{ij}(2)w_{ij}(3)w'_{ij}(3))$ then any of the D-values (000000), (000011), (001100), (110000), (111100), (110011), (001111), and (111111) indicates no-tampering of (i, j)th pixel. It gives $W_{ij}(l) = 0$ (where $l = 1, 2, 3$) and as a result we get the value of $W_{ij} = 0$, (i.e., $W_{ij} = 0$ indicates no-tampering of (i, j)th pixel). For other 56 D-values $W_{ij} = 1$ and it indicates tampering of (i, j)th pixel. Hence if a pixel is detected as tampered by this approach it is 100% accurate. However, if a pixel is detected as not tampered then it is possible that the pixel is tampered. This is called false-negative detection and it occurs because $2^5(= 32)$ MSBs are mapped to $2^3(= 8)$ watermarks, which is equivalent to mapping 4 MSB symbols to 1 LSB symbol and thus it facilitates tampering.

To handle false-positive tamper detection, the aggressive tamper detection A_gTDM is defined as follows:

$$W = W_{ij}(1) \mid W_{ij}(2) \mid W_{ij}(3) \mid W_{ij}(4) \mid W_{ij}(5), \quad (8)$$

where the operand bits $W_{ij}(t)$, $t = 1 \cdots 5$ are defined by

$$W_{ij}(1) = w_{ij}(1) \oplus w'_{ij}(1),$$
$$W_{ij}(2) = w_{ij}(2) \oplus w'_{ij}(2),$$
$$W_{ij}(3) = w_{ij}(3) \oplus w'_{ij}(3), \quad (9)$$
$$W_{ij}(4) = w_{ij}(1) \oplus w'_{ij}(2),$$
$$W_{ij}(5) = w_{ij}(1) \oplus w'_{ij}(3).$$

It is important to note that the matching relationships between $w_{ij}(1)$ and $w'_{ij}(2)$ in $W_{ij}(4)$, and $w_{ij}(1)$ and $w'_{ij}(3)$ in $W_{ij}(5)$ are independent of pixel tampering. These relationships are included in the AgTDM to artificially increase the false- positive tamper detection (in turn this will increase actual tampered pixel detection). Note that this is one of the novelties of the technique.

In this approach, the two D-values (000000) and (111111) are considered for the detection of not-tampered pixels and other 6 values are considered for the detection of tampered pixel in addition to remaining 56 values. This logic is called an aggressive logic and it is derived from the fact that it is difficult for an attacker to tamper a pixel that will give the D-values (000000) or (111111) than other six D-values. Due to its aggressiveness, this approach introduces false-positive tamper detection (i.e., it can detect not-tampered pixels as tampered pixels—Figure 2 illustrates this scenario). Now the goal is to combine tamper detection responses of N_gTDM and A_gTDM to make the final tamper detection response. This process is carried out by the module DET in Figure 1(b). The DET module uses the following decision rule for detection response.

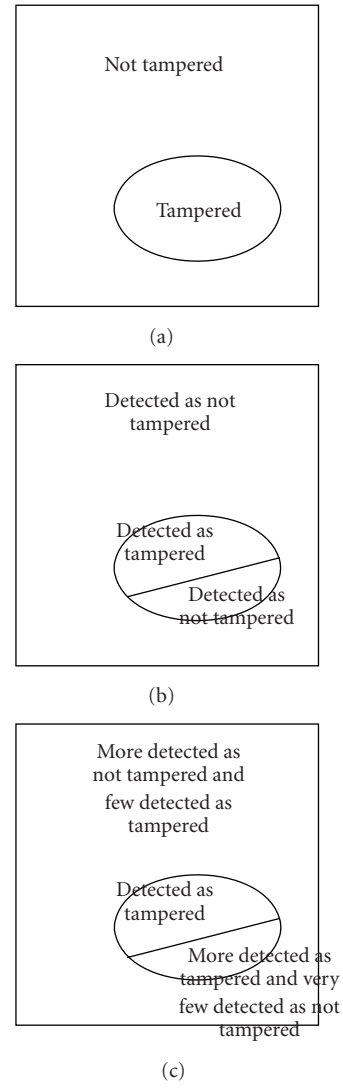

Figure 2: (a) An actual tampered scenario, (b) possible tamper detection response of N_gTDM, and (c) possible detection response of A_gTDM.

Note that if a pixel is detected as tampered by N_gTDM, then it is not possible for A_gTDM to detect it as a not-tampered pixel; hence, no rule is considered for this case.

2.4. Generalized Approach. The proposed 5MSB:3LSB technique can be easily generalized to nMSB:(8-n)LSB due to its simplicity. This generalization will lead to a trade-off between the image quality (i.e., PSNR value) and tamper resistance (i.e., TRP value). For example, if 4MSB:4LSB approach is selected we can certainly improve the tamper resistance capability (i.e., high TRP), but the quality of the image formed by the original 4MSB is not acceptable (low PSNR). Similarly, if the 6MSB:2LSB is selected the quality of the image can certainly be improved (high PSNR) but the tamper resistance capability will be very low (i.e.,

Input: Watermarked Image
Output: Tampered and Not-Tampered Pixels
(1) If a pixel is detected as tampered by N_gTDM then the pixel is certainly a tampered pixel;
(2) If a pixel is detected by both N_gTDM and A_gTDM as not tampered then the pixel is considered a not-tampered pixel;
(3) If a pixel is detected by N_gTDM as not tampered and detected by A_gTDM as tampered then
  (3.1) If this pixel belongs to a block (8×8 pixels) that consists of all pixels that are detected by N_gTDM as not tampered then the pixel is considered not tampered;
  (3.2) Else if the pixel has middle gray value using the 5-MSB bits (i.e., between 112 and 136 inclusive) then the pixel is high likely tampered otherwise the pixel is not tampered.

Algorithm 1: Pixel-Level Tamper Detection Rule.

Table 1: Comparison of proposed SS and ZW methods.

(a)

Image (1) Names	PSNR (2) SS method	PSNR (3) ZW method	Time (4) SS method	Time (5) ZW method
Biltmore	38.30	37.97	12.86	154.78
Lena	38.17	37.94	12.66	154.77
UNCG	37.74	37.87	12.64	154.38
Flower Grdn	37.72	37.90	12.66	155.13
House+Tree	37.65	37.83	12.64	153.14

(b)

Actually (6) Tampered	Detection (7) Rates		False (8) Negative		False (9) Positive	
	SS	ZW	SS	ZW	SS	ZW
4211	99.81%	69%	0.19%	0.31%	0.35%	2.18%
4184	99.90%	65%	0.10%	0.35%	2.31%	2.23%
4140	99.73%	65%	0.27%	0.35%	0.99%	2.20%
4098	99.39%	0.65	0.61%	0.35%	1.60%	2.29%
4186	99.95%	66%	0.05%	0.34%	1.06%	2.15%

low TRP). Also $2^6 (= 64)$ MSBs are mapped to $2^2 (= 4)$ watermarks causing significant increase in the false-negative pixels. Hence 5MSB:3LSB approach is preferable than other methods including 4MSB:4LSB and 6MSB:2LSB.

3. Simulation Results

Simulation results have been obtained using several images but the results of "Biltmore Estate" image are presented in Figures 3(a)–3(f). The results collated from several images are also presented in Table 1. The actual size of the images considered in this paper is 256×256 pixels. Figures 3(a) and 4(a) show the original "Biltmore Estate" and "UNCG" images and they are used to watermark using both the proposed (SS) and Zhang-Wang (ZW) [7] approaches. The Matlab code provided by *X*. Zhang and S. Wang has been used to obtain results of their approach. Figure 3(b) shows the tamper data and locations, and this tamper data alter 4211 pixels with $\alpha = 0.1631$ (where α is the ratio between the number of tampered blocks and the total number of blocks [7]).

Figure 3(a) is watermarked using the SS approach and then tampered according to the data in Figure 3(b). PSNR of the SS watermarked image is 38.30 dB. The difference between this tampered image and the original is shown in Figure 3(c). Similarly, ZW approach is used and its corresponding PSNR is 37.97 and the difference image is shown in Figure 3(d). The close PSNR values in Table 1 (columns 2 and 3) show that the watermark embedded by the SS and ZW approaches affect the quality of the images the same way, however, the other parameters such as the computational time, detection rate, false-positive, and false-negative show that the SS approach performs better than ZW approach. Also the high PSNR values and the Figures 3(c)

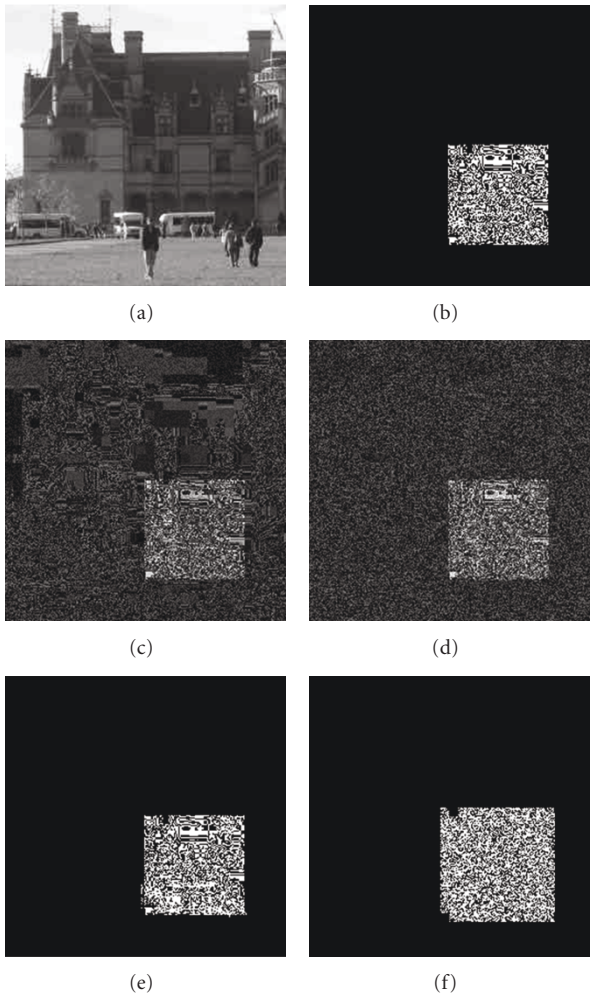

Figure 3: Simulation results. (a) Original "Biltmore Estate" image of size 256 × 256 pixels, (b) tamper data and their locations, (c) proposed watermark and tampered data-PSNR = 38.30 dB, (d) Zhang-Wang watermark and tampered data-PSNR = 37.97, (e) tampered pixels detected by A_gTDM, and (f) tampered pixels detected by ZW approach.

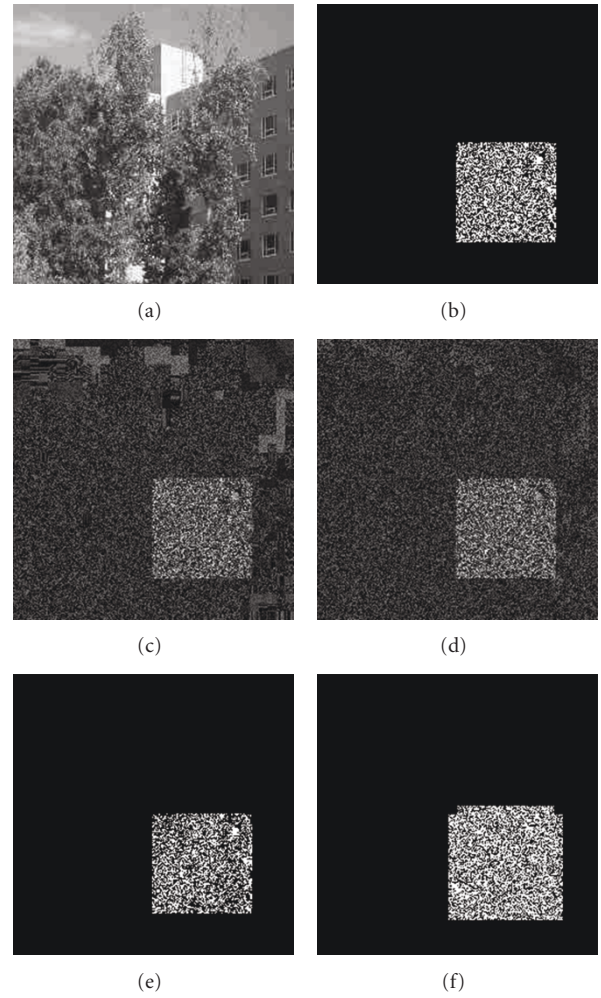

Figure 4: Simulation results. (a) Original "UNCG" image of size 256 × 256 pixels, (b) tamper data and their locations, (c) proposed watermark and tampered data–PSNR = 37.74 dB, (d) Zhang-Wang watermark and tampered data–PSNR = 37.87, (e) tampered pixels detected by A_gTDM, and (f) tampered pixels detected by ZW approach.

and 3(d) show that the proposed watermarks do not degrade the image properties in homogeneous regions compared to ZW approach.

In Figures 3(e) and 3(f), the watermarks extracted using the A_gTDM and ZW approach are presented, respectively. It can be seen from the patterns that the watermark extracted using A_gTDM is much closer to the actual watermark (see Figure 3(b)). Findings are as follows: A_gTDM detects 4420 pixels as tampered pixels where 3997 are certainly tampered pixels and 423 (which include 206 tampered pixels and 217 not-tampered pixels) are high likely tampered pixels. That is, 4203(= 3997 + 206) tampered pixels have been detected with 217 false-positive pixels and 8 false-negative pixels. In other words, A_gTDM is capable of detecting 99.8% (4203/4211) of tampered pixels with 0.35% (217/61325) false-positive detection and 0.19% (8/4211) false-negative detection. ZW approach detects 2912 tampered pixels with 1299 false-negative and 1341 false-positive pixels. The simulation also indicates that the proposed extraction takes about 12.86 seconds *cputime* (Matlab function) whereas ZW approach takes 154.78 seconds. The corresponding images of the "UNCG" image are displayed in Figures 4(b)–4(f). They demonstrate the similar results. The collated results, using several images, in Table 1 follow the same arguments and they support the same conclusions.

4. Conclusion

A fragile image watermarking scheme is proposed in this study. The proposed scheme is a 5MSB:3LSB technique and capable of detecting pixel-level tampering present in the 5 MSBs. This scheme has also introduced nonaggressive and aggressive tamper detection methods. The novelty of the

aggressiveness is that it artificially increases the false-positive tamper detection to increase the accuracy of the actual tampered pixel detection. Using these tamper detection, superior pixel-level tamper detection can be achieved with negligible false-negative and false-positive tamper detection responses.

Acknowledgments

The author of this paper wishes to thank X. Zhang and S. Wang for providing Matlab program of their approach in [7]. The author also thanks anonymous referees for their valuable comments and recommendations to enhance this paper.

References

[1] S. Walton, "Image authentication for a slippery new age," *Dr. Dobb's Journal*, vol. 20, no. 4, pp. 18–26, 1995.

[2] J. Fridrich, "Security of fragile authentication watermarks with localization," in *Security and Watermarking of Multimedia Contents IV*, vol. 4675 of *Proceedings of SPIE*, pp. 691–700, January 2002.

[3] Y. Li, H. Guo, and S. Jajodia, "Tamper detection and localization for categorical data using fragile watermarks," in *Proceedings of the 4th ACM Workshop on Digital Rights Management (DRM '04)*, pp. 73–82, October 2004.

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

[5] S. Suthaharan, "Fragile image watermarking using a gradient image for improved localization and security," *Pattern Recognition Letters*, vol. 25, no. 16, pp. 1893–1903, 2004.

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

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

[8] 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," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 869–882, 2007.

[9] P.-P. Liu, Z.-L. Zhu, H.-X. Wang, and T.-Y. Yan, "A novel image fragile watermarking algorithm based on chaotic map," in *Proceedings of the 1st International Congress on Image and Signal Processing (CISP '08)*, vol. 5, pp. 631–634, May 2008.

[10] S. Che, B. Ma, and Z. Che, "An adaptive and fragile image watermarking algorithm based on composite chaotic iterative dynamic system," in *Proceedings of the 4th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP '08)*, pp. 159–162, IEEE Computer Society, 2008.

[11] A. A. Tsonis, *Chaos: From Theory to Applications*, chapter 6, Plenum Press, New York, NY, USA, 1992.

[12] S. Suthaharan, "Chaos-based image encryption scheme using Galois field for fast and secure transmission," in *Real-Time Image Processing 2008*, vol. 6811 of *Proceedings of SPIE*, pp. 1–9, 2008, 681105.

[13] S. Suthaharan, "A perceptual quality metric for digital video coding," *Electronics Letters*, vol. 39, no. 5, pp. 431–432, 2003.