ELSEVIER

The Journal of Systems and Software 67 (2003) 131–140

The Journal of Systems and Software

www.elsevier.com/locate/jss

Detecting and restoring the tampered images based on iteration-free fractal compression

Ching-Te Wang [a,*], Tung-Shou Chen [b], Shao-Hau He [b]

[a] General Education Center, National Chin-Yi Institute of Technology, No. 35, Kabe 215, Sec. 1, Chung-Shan Road, Taiping, Taichung 411, Taiwan, ROC

[b] Department of Information Management, National Taichung Institute of Technology, Taichung 404, Taiwan, ROC

Received 1 October 2001; received in revised form 26 February 2002; accepted 16 April 2002

Abstract

In current research, fractal image compression scheme has been used to compress images. In this paper, we apply this technique to refine characteristic values of a specific image and embed the characteristic values into the least significant bits of pixels in the image. Once the picture is damaged by an intruder, the system can detect the tampered blocks and restore them using characteristic values without requiring the original image. From experimental results, we see that the proposed scheme can precisely achieve the detection of a damaged image and improve the performance of image restoration.

Keywords: Detecting and restoring a tampered image; Iteration-free fractal image coding; Least significant bit

1. Introduction

The Internet has become an important part of our world as computer and communication techniques have progressed rapidly. People use the Internet to communicate with each other via e-mail. Electronic commerce has also been developed for users to shop over the Internet. Currently, the World Wide Web is full of various information and knowledge, which allows users to browse and retrieve messages. Information such as digital images is created and easily transmitted over the Internet. Digital media with properties of portability and easy replication are suitable for communicating over the Internet. However, there are several associated problems that occur as a result of the convenience. For example, assume that a sender wants to transmit a digital image to a receiver, but an intruder intends to intercept the image, tamper it and spread it over the Internet. In this case, is the integrity of the image still preserved after it is transmitted to the receiver? Was the image forged by the intruder during transmission? If someone sees the tampered picture, can he tell whether or not the original meaning of the image has been changed? How does one authenticate the accuracy of the image? Once the intruder tampers the image, can we detect it and then restore it back to the original image? Obviously, these questions are based on problems that have occurred. Thus, in this paper we will focus on addressing the problems relating to the tamper proofing of digital images and authenticating their integrity.

In recent years, researchers have paid much attention to the area of digital watermarking systems, which are used to design techniques for copyright protection and tamper proofing of multimedia data. Traditionally, the robust technique for the copyright protection imperceptibly embeds a watermark into a digital image and protects the ownership of the image. In such methods, the protected image can resist general image processing such as lossy compression, sharpening and scaling, but these methods cannot authenticate the integrity of the image. Watermarking techniques can also be used to address the problem of tamper proofing. Applications for this purpose are to authenticate the integrity of an image and sensitively detect whether or not the image

* Corresponding author. Tel.: +886-4-23924505x2230; fax: +886-4-23920892.

E-mail addresses: ctwang@ncit.edu.tw, ctwang@chinyi.ncit.edu.tw (C.-T. Wang), tschen@ntit.edu.tw (T.-S. Chen), markho@seed.net.tw (S.-H. He).

0164-1212/03/$ - see front matter © 2002 Elsevier Inc. All rights reserved.
doi:10.1016/S0164-1212(02)00094-8

has been modified or forged during the transmission to its destination.

Currently, there is a trend toward using fragile watermarking techniques to handle the problems of tamper proofing and authentication by combining watermarking systems and signature schemes. In Lu and Liao (2000), they classified these methods into watermark-based (Kunder and Hatzinakos, 1999; Walton, 1995; Wu and Liu, 1998; Yeung and Mintzer, 1997), and signature-based (Bhattacharjee and Kutter, 1998; Lin and Chang, 2001; Lou and Liu, 2000; Lu and Liao, 2000; Schneider and Chang, 1996; Wong, 1998; Yang et al., 1999) systems, according to their design techniques. The watermark-based methods focus on imperceptibly embedding a watermark into a digital image and then extracting the signal from the marked image for verification. For example, Wu and Liu (Wu and Liu, 1998) proposed a watermark-based system for image authentication based on discrete cosine transformation (DCT for short). They modified the coefficients in the low-middle frequency domain and embedded a binary watermark into the image. Kundur and Hatzinakos (Kunder and Hatzinakos, 1999) also proposed an image authentication scheme based on discrete wavelet transformation. In their scheme, any modification will cause the extracted watermark to differ from the original, which indicates that a tampered signal has been found.

Alternatively, signature-based methods refine messages from an image as characteristic values and then derive a digital signature using cryptographic schemes. This signature can be attached to the protected image or be regarded as a watermark embedded into the image for verification. In Lu and Liao (2000), Lu and Liao first found an image structure in the frequency domain of discrete wavelet transformation. They used the inter-scale relationship as a characteristic to derive the structural digital signature (SDS for short). Thus, the scheme can detect the tampered regions depending on the number of pairs in SDS. In Lin and Chang (2001), Lin and Chang proposed an image authentication scheme that can detect the modified blocks and resist lossy compression. First, they find the relationship of coefficients in the DCT for the corresponding position in different blocks. Then, the relationship is encoded into characteristic codes and a signature is generated. Afterward, the modification can be detected from the malicious manipulation. Briefly, we found that some previously proposed schemes could only detect whether or not the image was tampered and some could partially detect where the blocks were modified or forged. However, none of the proposed schemes could return the damaged blocks to their original form.

To achieve these purposes, we propose a scheme to detect the tampered regions of an image and then restore the image using the iteration-free fractal compression method (Chang and Kuo, 2000). In the proposed scheme, an efficient domain pool can be generated from a mean image. By applying the affine transformation, the scheme assigns each range block to a similar domain block in the efficient pool. On the basis of Chang and Kuo's compression method, the proposed scheme refines the characteristic values from the image and hides them in the specific regions. Furthermore, by improving the iteration-free fractal coding, the system can clean the tampered blocks and return the image to its original form.

In Section 2, the iteration-free fractal compression and the block-averaging method are reviewed. In Section 3, we propose a detection and restoration system that contains the embedding and extraction techniques. The experimental results and some discussions about the system are described in Sections 4 and 5. Finally, we make some conclusions in Section 6.

2. Iteration-free fractal image coding

2.1. Encoding stage

In the iteration-free fractal compression scheme (Chang and Kuo, 2000; Fisher, 1995), an image F of size $I \times I$ is partitioned into non-overlapping blocks R_r of size $B \times B$, where R_r are range blocks and $0 \leqslant r \leqslant (I/B \times I/B) - 1$. Let $R_r(i,j)$ be the value of pixel (i,j) in R_r, where $0 \leqslant i,j \leqslant (B-1)$. The mean value m_r of all pixels in R_r is computed as follows:

$$m_r = \frac{1}{B \times B} \sum_{i=0}^{B-1} \sum_{j=0}^{B-1} R_r(i,j). \tag{1}$$

After computing the means of all range blocks in F, the scheme constructs a mean image M containing the means of $I/B \times I/B$ range blocks. Then, the mean image M is partitioned into overlapping domain blocks D_d, which are the same size as the range blocks. Assume that each domain block is shifted one pixel from the previous block and d is restricted within $0 \leqslant d \leqslant (I/B - B + 1) \times (I/B - B + 1)$. The set of all domain blocks is called the domain pool. Obviously, the neighboring domain blocks are similar to each other. The distance of each domain block is not only one certain pixel but may be a number of different pixels. The value T and the number N_D of domain blocks in the domain pool have the following relation (Chang and Kuo, 2000):

$$T = \left\lfloor \frac{I/B - B}{\sqrt{N_D} - 1} \right\rfloor, \quad T \geqslant 1. \tag{2}$$

Chang and Kuo employed the LBG (Linde et al., 1980; Nelson and Gally, 1999) algorithm or the block-averaging method to reduce the redundancies and enhance the equality among the domain blocks. After

using these methods, the domain blocks with a high similarity are reset to one block and thus an efficient domain pool is obtained (Chang and Kuo, 2000). In general, the block-averaging method has better performance than the LBG algorithm in obtaining the efficient domain pool because the block-averaging method is not required to iterate the computation. Therefore, we use the block-averaging method in the proposed scheme.

Also, Chang and Kuo computed the variance of each range block R_r. If the variance of R_r is smaller than a threshold value TH_V, all pixels in the range block are regarded as the same value and are identified as gray level by the human eye. Therefore, Chang and Kuo saved the mean of R_r as a fractal code. On the other hand, if the variance is greater than the threshold value TH_V, the domain blocks are transformed by the new affine transformation (Beaumont, 1990; Chang and Kuo, 2000). The new affine transformation is defined as follows:

$$R' = \iota\{\alpha \cdot D + \mu_R - \alpha \cdot \mu_D\} = \iota\{\alpha \cdot (D - \mu_D) + \mu_R\}, \quad (3)$$

where D is all pixel values in a domain block, μ_R is the mean of the correlative range block, μ_D is the mean of D, ι denotes one of eight kinds of rotative and reflective transformations that are also called isometries (Fisher, 1995; Jacquin, 1990), α is the contrast scaling, and $\mu_R - \alpha \cdot \mu_D$ is the luminance offset. Assume the transformed domain block D_d is the most similar to R_r, the characteristic value of this specific domain block is saved and used as the fractal code of the range block R_r. In order to distinguish between the encoding of range blocks, the scheme must save a header to indicate whether or not the variance of a range block is higher than the threshold. In addition, the position of the domain block P_D, which is the most similar to R_r, must be saved. The position of the transformed domain block in the mean image is considered as part of the fractal code. In general, the mean square error (MSE) can measure the distortion between the range block and the transformed blocks as follows:

$$\mathrm{MSE}(R, R') = \frac{1}{B^2} \sum_{i=0}^{B-1} \sum_{j=0}^{B-1} (R(i,j) - R'(i,j))^2, \quad (4)$$

where R is a range block in the original image and R' is the transformed domain block obtained by using the new affine transformation. The smaller value of MSE is the more similar of the range block R and the transformed block R'. In accordance with the previous statements, a fractal code is encoded by the transformation that contains the mean of the range block, isometry, contrast scaling, luminance offset, and corresponding position of the transformed domain block. The corresponding position of a transformed domain block is represented by the position of the upper-left-hand corner pixel in the specific image. After encoding all range blocks, the encoding phase of the iteration-free fractal compression is complete. A basic flow chart is shown in Fig. 1.

The fractal code of each range block is relative to its mean. If the variance of the range block is smaller than the threshold TH_V, the mean is used as the fractal code. On the other hand, if the variance is larger than the threshold value TH_V, the mean of the range block is used as a part of the fractal code that is coded by the affine transformation. In the decoding stage, the scheme uses the fractal codes to reconstruct the mean image, which is the same as the mean image of the original image. Consequently, the iteration-free scheme can reconstruct a high quality of decoded image.

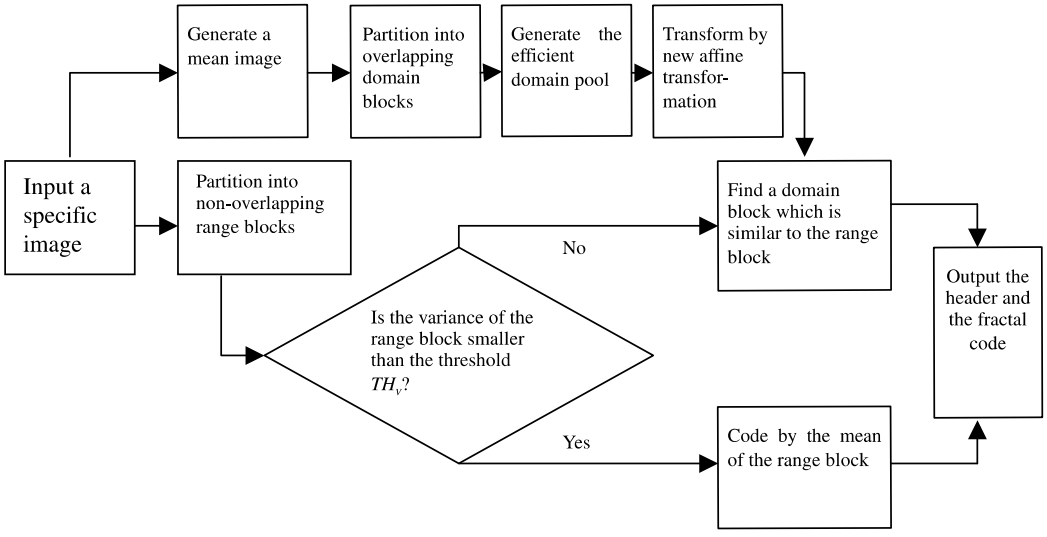

Fig. 1. Flow chart of the encoding stage for the iteration-free scheme.

2.2. Decoding stage

First, the scheme must reconstruct the mean image from the fractal codes in the decoding stage. Then, the reconstructed mean image is partitioned into overlapping $B \times B$ domain blocks, and the efficient domain pool is generated by the LBG algorithm or the block-averaging method (Chang and Kuo, 2000). The recorded header in the encoding stage is used to distinguish whether the range block is encoded by the mean or by the new affine transformation. If a range block is encoded by the mean, the block is reconstructed so that all pixel values are equal to the corresponding mean in the fractal codes. Otherwise, the scheme reconstructs a block, which is transformed by the new affine transformation. When all range blocks are decoded, the decoding image has been obtained. The flow chart of the decoding stage is shown in Fig. 2.

2.3. Block-averaging method

In general, a mean image is partitioned into overlapping domain blocks; the neighboring domain blocks are similar to each other. For the aforementioned reason, Chang and Kuo integrated the purposed block and its neighboring block into an efficient domain block. Assume that a pixel value in the top-left corner of the purposed block D_d is $D_d(i,j)$, where $0 \leqslant i,j \leqslant B-1$. Then, the scheme generates three $B \times B$ blocks whose top-left corners are $d_1(i,j)$, $d_2(i,j)$, and $d_3(i,j)$, respectively, as in Fig. 3, where $d_1(i,j)$ is equal to $D_d(i,j+1)$, $d_2(i,j)$ is equal to $D_d(i+1,j)$, $d_3(i,j)$ is equal to $D_d(i+1,j+1)$. The pixel value of the efficient domain block is defined as follows:

$$D'_d(i,j) = 1/4[D_d(i,j) + d_1(i,j) + d_2(i,j) + d_3(i,j)]. \quad (5)$$

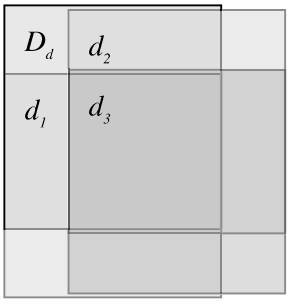

Fig. 3. The related position of four neighboring and partly overlapped image blocks.

3. System for detecting and restoring the tampered image

On the basis of iteration-free fractal coding, we propose a system to detect and restore the tampered image by extracting the characteristic values of a specific image. First, the system generates three copies of characteristic values and permutes them using a pseudorandom number generator. The copies of characteristic values are embedded in the least significant bit (LSB) of all pixels in the image. The specific image, which is processed by the aforementioned procedures, is called the protected image. Fig. 4 shows a flow chart of the embedding stage of the proposed system.

When a user requests that the system detects whether or not the protected image is tampered, the system can extract the data from the LSBs in the protected image and permute the extracted data using the pseudorandom number generator. Then, the data can be refined into a piece of correct characteristic values. From the correct values, the system can detect the protected image. If the image has been tampered by an intruder, the system outputs a detection image and a restoration image of the same size as the protected image. The detection image shows that the blocks in the protected image have been tampered by an intruder. The restoration image indi-

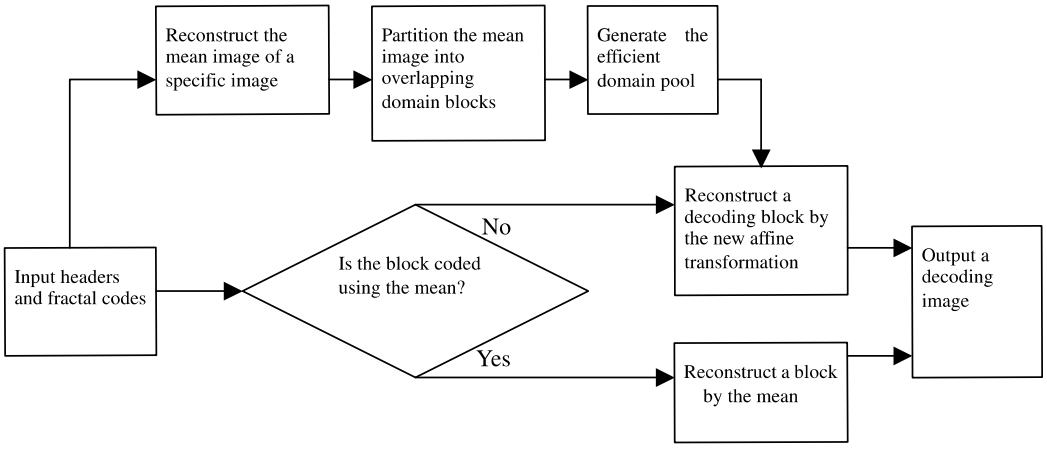

Fig. 2. Flow chart of the decoding stage for the iteration-free scheme.

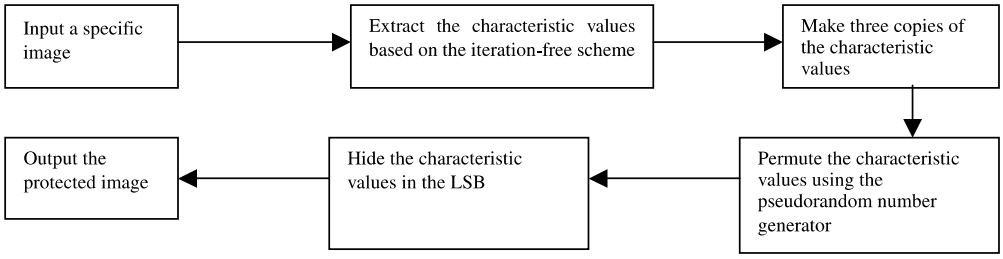

Fig. 4. Flow chart of the embedding stage for the proposed system.

cates that the tampered blocks have been restored by the characteristic values. A flow chart of the detection and restoration stage in the system is shown in Fig. 5.

3.1. Extraction and embedding stages

3.1.1. Extract characteristic values from a specific image

First, an $I \times I$ specific image F is inputted into the proposed system and the image F is partitioned into $I/B \times I/B$ range blocks R_r of size $B \times B$, where $0 \leqslant r \leqslant (I/B \times I/B) - 1$. $R_r(i,j)$ denotes the value of pixel (i,j) in R_r, and $0 \leqslant i,j \leqslant B-1$. Fig. 6(a) shows that an image is partitioned into range blocks. Then, the mean value m_r of the range block is computed by Eq. (1).

The system generates an $I/B \times I/B$ mean image whose pixel value corresponds to the average of each range block. Also, there are $(I/B - B + 1) \times (I/B - B + 1)$ domain blocks in the mean image M as in Fig. 6(b). However, the domain blocks are very similar to each other. The system employs the block-averaging method (Chang and Kuo, 2000) to reduce the redundancy and generates an efficient domain pool.

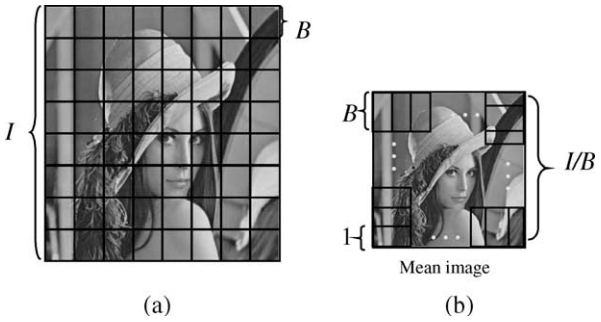

Fig. 6. Example of the partitioned image and the mean image.

Second, the variance of each range block R_r is evaluated. If the variance is smaller than the threshold TH_V, the mean of the range block R_r is used as the characteristic value of the block and the mean is expressed by 7 bits. On the other hand, if the variance is larger than the threshold TH_V; the domain block is rotated and reflected by the isometric transformation as in Fig. 7. Then, the blocks are compared with the range block and the most similar one is selected. The characteristic value

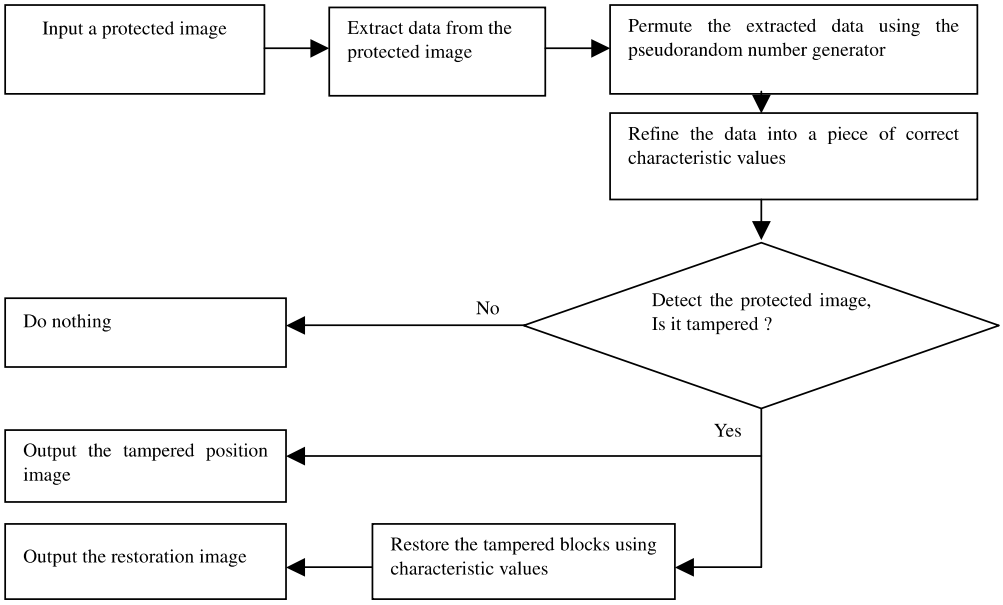

Fig. 5. Flow chart of the detection and restoration stages.

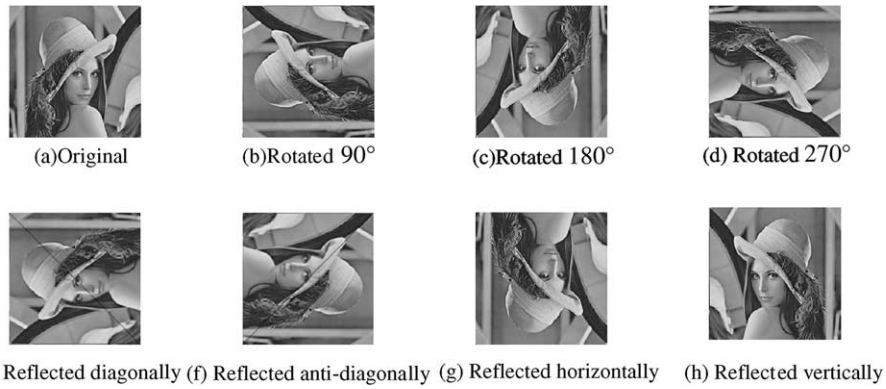

Fig. 7. Example of isometric transformation blocks.

of range block R_r is expressed by the position in the efficient domain blocks, the value of the isometric transformation and the mean of the range block. The value of the isometric transformation is represented by 3 bits and the mean of the range block is represented by 7 bits. To save the position of domain blocks, the number of bits Bits_{PD} is dependent upon the number of blocks in the efficient pool and is computed as follows:

$$\text{Bits}_{\text{PD}} = \lceil \log_2 N_{\text{D}} \rceil, \tag{6}$$

where N_{D} is the number of all of the domain blocks in the efficient domain pool. Once the characteristic values of all range blocks are found, the extraction procedure is complete.

After the aforementioned procedure, the extracted characteristic values consist of the means of all range blocks in the protected image, the isometries and the position of all range blocks for the high variance. Obviously, the number of characteristic values is not fixed, because the number of domain blocks in the efficient pool is variable and the threshold TH_{V} will affect the number of smooth range blocks, which are low variance. For example, if the number of domain blocks in the efficient pool is small, the number of bits used to record the position will also be decreased. Similarly, if the threshold TH_{V} is raised, the number of range blocks using the mean to represent the characteristic values will be increased, and the number of bits to store the position of the domain blocks and the isometries will be decreased by a significant amount.

For the convenience of extracting characteristic values, the header information $H = \{H_0, H_1, H_2, \ldots, H_{I\times I-1}\}$ of size $I \times I$ has to be added to the characteristic values, where H are binary values. The information can indicate whether or not the variance of range blocks is larger than the threshold TH_{V}.

3.1.2. Hiding characteristic values into the image

Each pixel in a gray digital image contains eight bits, so there are $2^8 = 256$ gray levels for a pixel. The heaviest bit in a pixel is called the most significant bit (MSB), because the MSB can determine the pixel value is increased or decreased by 128. On the other hand, the LSB can only determine whether the pixel value is increased or decreased by one. According to the previous statement, if we modify the LSB that slightly influences the pixel value, the view seen by the human eye is also affected slightly. Therefore, the characteristic values are embedded into the LSB in the proposed scheme and the protected image still retains its high quality.

Before embedding the characteristic values, the system has to make three pieces of characteristic values, which are used to correct the tampered regions if the blocks in the protected image are tampered by an intruder. In order to keep the characteristic values secret, the values have to be permuted by a pseudorandom number generator. Assume the bit values of the characterization are $C = \{C_i | i = 0, 1, 2, \ldots, I \times I - 1\}$. The system gives a seed to the pseudorandom number generator and generates a random sequence $L = \{L_0, L_1, L_2, \ldots, L_{I\times I-1}\}$ of size $I \times I$, where $L_i \in [0, I \times I - 1]$ and $L_i \neq L_j$, if $i \neq j$. The seed is stored and used in the detection and restoration. Here an array A of size $I \times I$ has to be set for storing the bit values of the characterization with the order of the sequence L. Thus, the permuted characteristic values can be found. Finally, the least significant bit $\text{LSB}_F = \{\text{LSB}_0, \text{LSB}_1, \text{LSB}_2, \ldots, \text{LSB}_{I\times I-1}\}$ of all pixels perform *AND* logical operation with zero, and all of the LSB_F bits will become to be zero. Consequently, a protected image F' can be obtained by executing *OR* logical operation using the values stored in array A and LSB_F, according to the order of the sequence L.

3.2. Detecting and restoring the tampered image

3.2.1. Extracting characteristic values

Let F' be the final result, i.e., the protected image. In the first step, the characteristic values have to be extracted from F' in detection and restoration of the tampered image stage. The seed is restored from the system to generate a random sequence $L = \{L_0, L_1,$

$L_2, \ldots, L_{I\times I-1}\}$, which is the same as the one in the embedding stage. Then, the index of array A is replaced by the sequence L, the values of $\mathrm{LSB}'_F = \{\mathrm{LSB}'_0, \mathrm{LSB}'_1, \mathrm{LSB}'_2, \ldots, \mathrm{LSB}'_{I\times I-1}\}$ are restored to array A and three pieces of characteristic values C' are obtained. Note that the characteristic values are used to generate the header information. The system first corrects the header errors. Let $H'_1 = \{H1'_0, H1'_1, H1'_2, \ldots, H1'_{I\times I-1}\}$, $H'_2 = \{H2'_0, H2'_1, H2'_2, \ldots, H2'_{I\times I-1}\}$, and $H'_3 = \{H3'_0, H3'_1, H3'_2, \ldots, H3'_{I\times I-1}\}$. The header bits with the same index in H'_1, H'_2, and H'_3 are compared with each other. If two or more are equal in these headers, we claim that the information is correct. When all headers are compared respectively, a piece of a correct header is obtained. The correct header is utilized to extract the positions and the isometries of domain blocks and the means of all range blocks in the three pieces of characteristic values. Finally, three copies of the range blocks are gathered to correct the errors, and a piece of characteristic values can be precisely integrated.

3.2.2. Detecting tampered blocks in an image

In the second step, the position of the tampered blocks in a protected image is computed during the detection stage. The system generates a detection image and a restoration image, and both sizes are equal to the protected image F'. The image F' is partitioned into non-overlapping range blocks of size $B \times B$, and the system computes the means of the range blocks. If the mean of the range blocks is equal to the corresponding mean of the characteristic values, this indicates that the range block of F' is not tampered by an intruder. Then, the range block is duplicated and outputted to the detection image and the restoration image in their corresponding positions. If the range blocks in the protected image are tampered by an intruder, the mean of the range block must be changed. Therefore, the mean of the tampered block is different from the corresponding mean in the characteristic values; the system deems that the range block is tampered by an intruder. When detecting a tampered range block, the system outputs a white block of size $B \times B$ to the detection image and labels the positions where is tampered by an intruder. At the same time, the system also creates a restoration image in which the range block is transformed by the characteristic values.

3.2.3. Restoring the tampered image

To restore the tampered blocks, the means of the range blocks in the characteristic values are combined to generate a mean image. Then the efficient domain pool is created from the mean image. Furthermore, the header information is provided to determine whether or not the tampered block has high variance. If the variance of the range block is low, it is restored by the mean. Otherwise, the tampered block is restored by its position in the domain pool, isometry, and the mean of the range block in the characteristic values.

4. Experimental results

In this paper, the system is designed by Java 1.2.1. on an Intel Pentium III 600E PC. Fig. 8(a) shows the original 512×512 LENA, which is partitioned into 8×8 range blocks and the PSNR of protected LENA is 51.15 dB. The mean image of LENA of size 64×64 is shown in Fig. 8(b). An efficient domain pool is generated from Fig. 8(b) and shown in Fig. 8(c), which has 225 8×8 efficient domain blocks. Fig. 8(d) is a tampered image with a PSNR of 28.39 dB. Fig. 8(e) is the detection image and Fig. 8(f) is the restoration image with a PRNR of 43.27 dB.

The experimental results of PEPPER are shown in Fig. 9. Fig. 9(a) shows a protected 512×512 PEPPER which is partitioned into 8×8 range blocks and its PSNR is 51.13 dB. Fig. 9(b) is the mean image of PEPPER of size 64×64. Fig. 9(c) is an efficient domain pool generated from Fig. 9(b) and there are 225 8×8 domain blocks in Fig. 9(c). Fig. 9(d) is a tampered image with a PSNR of 28.87 dB. Fig. 9(e) is the detection image and Fig. 9(f) is the restoration image with a PSNR of 38.67 dB.

5. Discussions

In this section, several discussions are presented.

(1) In the embedding stage, a problem is raised when a pixel value embeds one bit at most. The question becomes what is the relationship between the size of the range block and the amount of characteristic values? To solve this problem, we show how many characteristic values will be generated in this stage. Assume that the source image F of size $I \times I$ is partitioned into non-overlapping blocks R_r of size $B \times B$ where R_r are range blocks and $0 \leqslant r \leqslant (I/B) \times (I/B) - 1$ If the variance of R_r is smaller than the threshold $\mathrm{TH_V}$, the mean value of R_r is used as the characteristic value with 7 bits. If the variance of R_r is larger than the threshold $\mathrm{TH_V}$, the characteristic values contain the position in the efficient pool, the isometric values and the mean value. The number of bits used to express the position is evaluated as in Eq. (6). In general, the number of domain blocks in the efficient pool is less than 256, that is, $N_\mathrm{D} = 256$, $\mathrm{Bits_{PD}} = \lceil \log_2 N_\mathrm{D} \rceil = 8$ bits. The number of isometries is 3 bits. The mean value requires 7 bits to store it. In addition, the header information H_i of 1 bit size is added to the characteristic value for each range block. There are a total of 8 bits used to store the characteristic value if its variance is less than the threshold and 19 bits if its

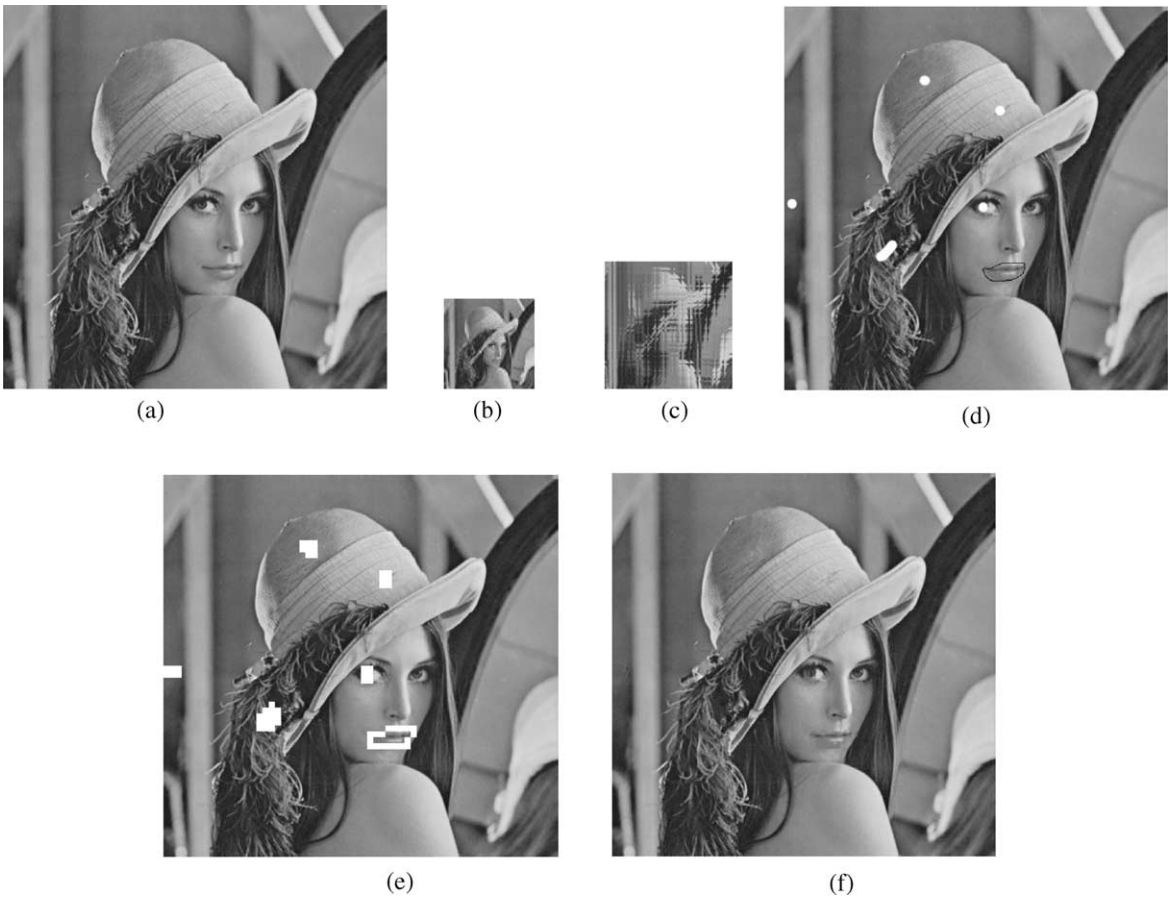

Fig. 8. Experimental results (1).

variance is larger than the threshold. Furthermore, the system makes three copies of the characteristic values in order to recover the tampered image. We assume without loss of generality that there is a 50% chance that the variance of the range block is larger than the threshold. Thus, the capacity of the image is sufficient to hide the characteristic values only if the following holds true:

$$3\left(\frac{1}{2}(8(I/B \times I/B)) + \frac{1}{2}(19(I/B \times I/B))\right) \leqslant I \times I. \quad (7)$$

We solve Eq. (7) and obtain the size B of each block that is larger than six pixels. That is, the image should be partitioned into range blocks of size $B \geqslant 7$ In our experiment, the block size is eight pixels.

(2) Two factors including the threshold value and the number of domain blocks in the efficient pool will influence the amount of characteristic values in the system. If the threshold value is raised, the number of blocks in which the variance is larger than the threshold, will be reduced. The number of characteristic values will also be reduced, since the number of bits needed to store the position and the isometries is not required. Similarly, the number of domain blocks in the efficient pool is raised, the number of bits needed to save the position will also be increased, and the number of characteristic values will also be increased. In general, there are 256 blocks in the efficient pool sufficient to match the range blocks.

(3) Recently, the image authentication technique (Lin and Chang, 2001) not only prevents image from tampered, but also allows acceptable lossy compression. However, their scheme could detect tampered regions, but could not recover the regions to original. In our method, the protected image can resist the lossless compression but the performance will be reduced in the lossy compression, since the embedding technique for hiding information in LSB is sensitive to the compression. To overcome the problem, we can properly modify the technique by using a transformation, such as discrete cosine transformation (Chen et al., 1999), which transforms the spatial domain of an image into a frequency domain. The information is hided in the low-middle frequency domain and the resistance of lossy compression can be improved. Thus, the proposed scheme not only detects the tampered regions and restores the image, but also resists the attacks of lossy compression.

(4) In this paper, we present a novel method different from the previous ones to solve the problem of image integrity. The proposed scheme uses the fractal compression method to refine the characteristic values and embed them into the least significant bits of the image.

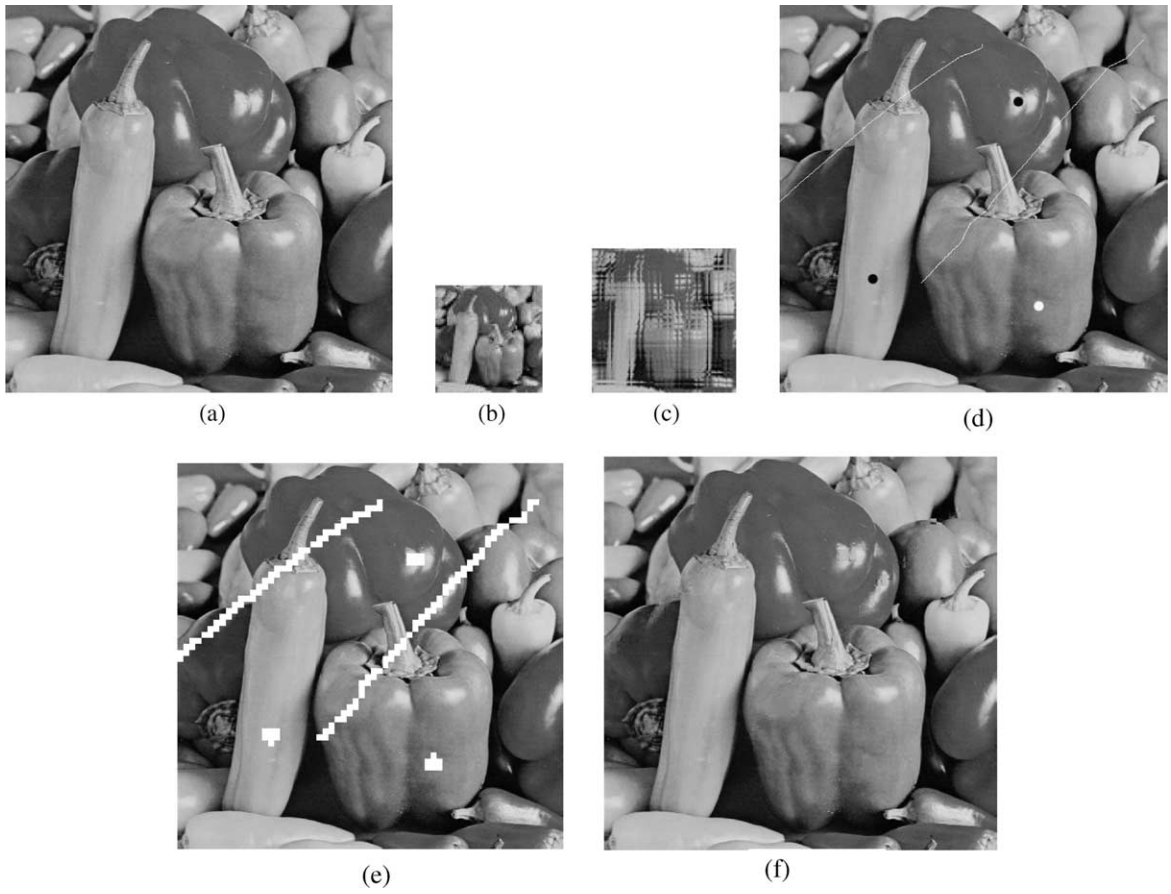

Fig. 9. Experimental results (2).

Furthermore, the system can extract the information from the protected image and permute it to find the characteristic values. The detection and restoration schemes are performed to determine and recover the tampered regions. Using this method, the detection of the tampered regions and recovery of an image can be efficiently achieved. Moreover, we can slightly modify the proposed scheme by inserting a signature scheme. That is, one copy of the characteristic value can be encrypted to a signature using the RSA scheme (Rivest et al., 1978), and then it can be embedded into the image. The receiver can verify the signature using the sender's public key and authenticate the author of the image. In this case, copyright protection, image integrity and image recovery are completely integrated into one system. Therefore, the new detecting and restoring the tampered images based on iteration-free fractal compression can be applied to the image transmission over the Internet.

6. Conclusions

We have proposed a detection and restoration system for tampered images based on iteration-free fractal compression. The system extracts characteristic values from a specific image and hides them into the protected image. The detection scheme can inspect whether or not the block is tampered. The restoration technique can restore the tampered blocks to their original form from characteristic values without needing the source image. The number of bits in the characteristic values is variable according to the number of domain blocks in the efficient pool and the threshold TH_V, which determines whether the range blocks are computed by the mean or by the characteristic values. The experimental results show that the tampered blocks in a protected image can be precisely detected, and the image is restored with high quality by the characteristic values.

References

Beaumont, J.M., 1990. Advances in block based fractal coding of still images. IEE Colloq. Application Fractal Techniques Image Processing, London, UK, pp. 3/1–3/5.

Bhattacharjee, S., Kutter, M., 1998. Compression tolerant image authentication. Proc. IEEE Int. Conf. Image Process. 1, 435–439.

Chang, H.T., Kuo, C.J., 2000. Iteration-free fractal image coding based on efficient domain pool design. IEEE Trans. Image Process. 9 (3), 329–339.

Chen, B., Latifi, S., Kanai, J., 1999. Edge enhancement of remote image data in the DCT domain. Image Vision Comput. 17 (12), 913–921.

Fisher, Y., 1995. Fractal Image Compression: Theory and Applications. Springer-Verlag, New York.

Jacquin, A.E., 1990. A novel fractal block-coding technique for digital images. IEEE Int. Conf. Acous. Speech Signal Process., 2225–2228.

Kunder, D., Hatzinakos, D., 1999. Digital watermarking for telltale tamper proofing and authentication. Proc. IEEE 87 (7), 1167–1180.

Lin, C.Y., Chang, S.F., 2001. A robust image authentication method distinguishing JPEG compression from malicious manipulation. IEEE Trans. Circuits Syst. Video Technol. 11 (2), 153–168.

Linde, Y., Buzo, A., Gray, R., 1980. An algorithm for vector quantizer design. IEEE Trans. Commun. 28, 84–95.

Lou, D.C., Liu, J.L., 2000. Fault resilient and compression tolerant digital signature for image authentication. IEEE Trans. Consumer Electron. 46, 31–39.

Lu, C.S., Liao, H.Y., 2000. Structural digital for image authentication: An incidental distortion resistant scheme. In: Proceedings on ACM multimedia 2000 workshop, California, USA, pp. 115–118.

Nelson, M., Gally, J.L., 1999. The Data Compression Book. M&T book, A Division of MIS! Press, Inc., New York, USA.

Rivest, R.L., Shamir, A., Addleman, L., 1978. A method for obtaining digital signatures and public-key cryptosystem. Comm. ACM 21 (2), 120–126.

Schneider, M., Chang, S.F., 1996. A robust content based digital signature for image authentication. Proc. IEEE Int. Conf. Image Process. 3, 227–230.

Walton, S., 1995. Image authentication for a slippery new age. Dr. Dobb's J. 20, 18–26.

Wong, P.W., 1998. A public key watermark for image verification and authentication. Proc. IEEE Int. Conf. Image Process. 1, 445–449.

Wu, M., Liu, B., 1998. Watermarking for image authentication. Proc. IEEE Int. Conf. Image Process. 2, 437–441.

Yang, C.R., Hwang, M.S., Tang, Y.L., 1999. Detecting the tampered pixels in images by length and breadth method. The Eighth National Conference on Science and Technology of National Defense, Tao-Yuan, ROC.

Yeung, M.M., Mintzer, F., 1997. An invisible watermarking technique for image verification. Proc. IEEE Int. Conf. Image Process. 2, 680–683.

Ching-Te Wang was born in Taichung, Taiwan, Republic of China, on October 28, 1956. He received his B.S. degree in mathematics from Tunghai University in 1980, his M.S. degree in applied mathematics from National Chung Hsing University in 1987, and Ph.D. degree in Computer Science and Information Engineering from National Chung Cheng University in 1999. Currently, he is an associate professor in General Education Center at National Chin-Yi Institute of Technology, Taichung, Taiwan. From 2001 till now, he has also been the Director of Computer Center of the college. His research interests include computer security, cryptography and computer algorithms.

Tung-Shou Chen was born in Taichung, Taiwan, Republic of China, on October 14, 1964. He received the B.S. and Ph.D. degrees from National Chiao Tung University in 1986 and 1992, respectively, both in Computer Science and Information Engineering. He served at the computer center, Chinese Army Infantry School, Taiwan, from 1992 to 1994. During the academic years 1994–97, he was on the faculty of the Department of Information Management at National Chin-Yi Institute of Technology, Taichung, Taiwan. From August 1998 to July 2000, he was the dean of Student Affairs and a professor of the Department of Computer Science and Information Management at Providence University. Since August 2000, he has been a professor of the Department of Information Management at National Taichung Institute Technology, Taichung, Taiwan. His current research interests include data structures, image cryptosystems, and image compression.

Shao-Hua Ho was born in Kaohsiung, Taiwan, Republic of China, on May 17, 1975. He received the B.S. degree in the Department of Information Management from National Taichung Institute of Technology in 2001. Now he is a computer software engineer in the Department of Information and System at Taiwan Power Company, Taipei, Taiwan. His current research interests include database and system design, and image compression.