Automatic Image Authentication and Recovery Using Fractal Code Embedding and Image Inpainting

Sung-Lin Tsai , Shuenn-Shyang Wang and Ya-Ling Lin

Department of Electrical Engineering
Tatung University

40 Chungshan N. Road, 3rd Sec.
Taipei, Taiwan, R.O.C.
Email: sswang@ttu.edu.tw

ABSTRACT

In this paper, we propose a scheme of automatic image authentication and recovery, in which the altered region of the image can be detected and recovered in a fully automatic fashion. To achieve high quality image recovery, we choose the region of importance (ROI) in the image and perform fractal encoding for the blocks of ROI. And then we insert the fractal codes of ROI and the watermark into the original image to obtain the watermarked image. If the watermarked image has been tampered, we can automatically localize the altered region, and apply fractal decoding and image inpainting for image recovery. Finally, experimental results are given to show the effectiveness of the proposed scheme.

1. INTRODUCTION

In recent years, Internet is an excellent distribution system for digital media because it is cheap and fast. It is known that digital media can be easily replicated and modified. Thus, digital watermarking has been developed to protect the copyright and integrity. A watermarking scheme can be classified as either robust watermarking or fragile watermarking. Robust watermarking is designed for the copyright and ownership verification. Fragile watermarking is used for image authentication and tamper detection, which can be used to identify any alteration of the images. A practical technique of fragile watermarking should have good capability to detect any malicious manipulations and to resist a wide spectrum of attacks including vector quantization attack, random alterations, collage attack, rotation and scaling. Many fragile watermarking methods have been proposed in the literature [1-5]. Wong [4] presents a public-key fragile watermarking that embeds a digital signature of the most significant bits (MSB) of the block into the least significant bits (LSB) of the same block. If the correct key is applied in the watermark extraction it shows the correct watermark indicating that the image is authentic and is not tampered. If the key is incorrect, the image is not watermarked, the image is cropped or the image is scaled, a random-noise like watermark will be extracted. Several enhanced methods based on Wong's scheme are proposed [5-7] to improve the property of tamper detection and localization or the ability of attack resistance. When the altered region is detected by fragile watermarking, instead of retransmitting the original image, one may desire to reconstruct the altered region using some techniques of image recovery. In [8], the method of self-embedding is presented to make it possible to recover portions of the image that have been cropped out, replaced or damaged without accessing the original image. Recently, a fragile scheme for image authentication and recovery is proposed [9] which recovers damaged

blocks by extracting the embedded average pixel value. In this paper, we intend to propose an alterative approach for image authentication and recovery based on the techniques of fractal code embedding and image inpainting.

In section 2, we briefly introduce the techniques of fragile watermarking, fractal image coding and image inpainting. In section 3, by combining these techniques we propose a scheme of automatic image authentication and recovery. Finally, some experiment results are shown in Section 4, and conclude it in section 5.

2. PERLIMINARIES

2.1. Wong's watermarking scheme
The watermark insertion and extraction procedures of Wong's watermarking scheme[7] are described in the following.

(A)Watermark Insertion
Consider a grayscale image X of size $M_X \times N_X$ pixels. By inserting an invisible watermark to each block of the image, one will get a watermarked image X^w. The watermark insertion procedure for each image block is shown as Fig.1. An original image X is partitioned into blocks X_r of size $(I \times J)$ pixels, and the rth block X_r in the image is denoted by

$$X_r = \left\{ X_{iI+a, jJ+b} : 0 \le a \le I-1; 0 \le b \le J-1 \right\} \quad (1)$$

For the binary watermark W of size $M_X \times N_X$ pixels that equals to the size of original image, the corresponding block W_r is denoted by

$$W_r = \left\{ W_{iI+a, jJ+b} : 0 \le a \le I-1; 0 \le b \le J-1 \right\} \quad (2)$$

Consider a cryptographic hash function

$$H(S) = (b_1, b_2, ..., b_n) \quad (3)$$

where S is input data of arbitrary length, b_i is binary output bits of the cryptographic hash function, and n is output size. An example of hash function is the well known MD5 with $n = 128$. For each block X_r, one forms the corresponding block $\tilde{X}_r$ where each element in $\tilde{X}_r$ equals to the corresponding element in X_r except that the least significant bit (LSB) is set to zero. And five parameters are input to the hash function

$$H\left(M_X, N_X, I_X, r, \tilde{X}_r\right) = \left(b_1^r, b_2^r, ..., b_n^r\right) \quad (4)$$

Then, the output array B_r is combined with W_r to form a new binary array F_r

$$F_r = W_r \oplus B_r \quad (5)$$

where $\oplus$ is the exclusive OR operation, and F_r is encrypted with a public key cryptographic system.

$$P_r = E_{K'}(F_r) \quad (6)$$

where $E(\cdot)$ is the encryption function of the public key system such as RSA and K' is the private key. Finally, the binary block of P_r is embedded into the least significant bit of the corresponding block $\tilde{X}_r$ to form a new block $\tilde{X}_r^w$ in the watermarked image.
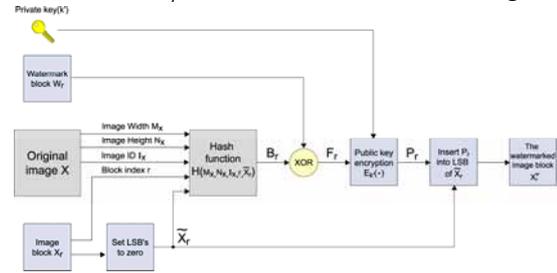
Fig. 1 Wong's watermark insertion procedure

(B)Watermark Extraction
The watermark extraction procedure for each image block is shown in Fig.2. The input watermarked image Y is partitioned

into blocks Y_r of size $(I \times J)$ pixels, and the r th block Y_r in the image is expressed as

$$Y_r = \left\{ Y_{iI+a, jJ+b} : 0 \leq a \leq I-1; \ 0 \leq b \leq J-1 \right\} \ (7)$$

We calculate the hash function to obtain $C_r = H(M_Y, N_Y, I_Y, r, \tilde{Y}_r)$. By extracting the least significant bits of Y_r to form Q_r and decrypting Q_r with the public key K that corresponds to the private key K' used in the insertion procedure, one obtains

$$G_r = D_K(Q_r) \qquad (8)$$

Finally, the output block is given by

$$Y_r^0 = G_r \oplus C_r \qquad (9)$$
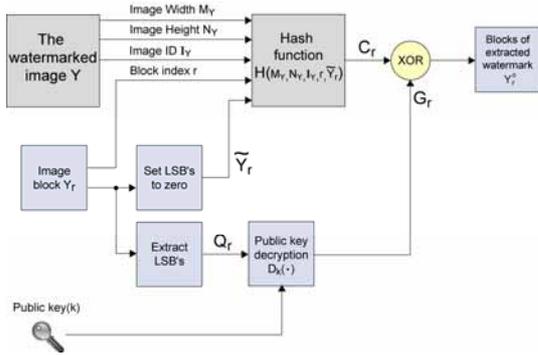
Fig. 2 Wong's watermark extraction procedure

2.2. Image Inpainting

Image inpainting is a technique which restores the disappeared or damaged region in an image. A number of image inpainting methods have been devised based on the extrapolation of neighboring pixels, recovery of edges and curvature-driven diffusions [10-12]. The image regions to be inpainted are manually selected and then color information is propagated inward from the region boundaries, i.e., the known image information is used to fill in the damaged region via diffusion. Image inpainting can produce a plausible rather than an exact reconstruction. Most inpainting methods work only for filling in small damaged region. Recently, an exemplar-based image inpainting[12] is proposed, which can restore large damaged region as well as small damaged region in a visually plausible way. This method will be applied to reconstruct the altered region that we localize by fragile watermarking instead of manually selection. Here, we briefly describe this method. Let Ω be the region to be inpainted and $\delta\Omega$ be its boundary. The inpainting is perform by smoothly propagating the information surrounding Ω in the direction of the isophotes entering $\delta\Omega$. In that algorithm, each pixel maintains a color value and a confidence value, and patches along the fill front are also given a temporary priority value which determines the order in which they are filled. Then the following three steps are iterated until all pixels have been filled:
1. Computing patch priorities.
2. Propagating texture and structure information.
3. Updating confidence values.

2.3. Fractal Image Coding

Fractal image coding was proposed by Bansley[13] based on Iterated Function System (IFS). Later, the practical fractal image coding based on Partitioned Iterated Function System (PIFS) was proposed by Jacquin [14-15]. Fractal image coding has the advantages of high compression ratio, highly reconstructed image quality, resolution independence in decoding and fast decoding process. The original image O of size $M \times M$ is divided into non-overlapping small blocks called range blocks $\{R\}$ of size $N \times N$ and overlapping large blocks called domain blocks $\{D\}$ of size $2N \times 2N$. $\overline{D}$ denotes the down sampled domain block of D and the size of $\overline{D}$ is equal to that of range blocks.

$$\overline{D}(i,j) = \{D(2i, 2j) + D(2i, 2j+1) + D(2i+1, 2j) + D(2i+1, 2j+1)\}/4 \ , i, j = 1.2, ..., N \quad (10)$$

For each range block, it is required to find the best matched domain block in terms of Mean Square Error (MSE)

$$E(R,D) = \| W \cdot \overline{D} - R \|^2 \qquad (11)$$

where W is an affine transformation. Let the isometric function t represents eight symmetries on the domain blocks, as shown in Fig.3. In the fractal image encoding, it needs to try the eight symmetries to find the minimum MSE

$$E = \| s \cdot \overline{D} + o - R \|^2$$
$$= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left[\left(s \times \overline{D}(i,j) + o \right) - R(i,j) \right]^2 \qquad (12)$$

where $R(i,j)$ is the gray level of the (i,j) pixel in the range block, $\overline{D}(i,j)$ is the gray level of the (i,j) pixel in the domain block, s is the contrast scaling factor and o is the luminance offset factor. Then s and o that minimize E are given by

$$s = \frac{N^2 \sum_{i=1}^{N} \sum_{j=1}^{N} R(i,j)\overline{D}(i,j) - \sum_{i=1}^{N} \sum_{j=1}^{N} R(i,j) \sum_{i=1}^{N} \sum_{j=1}^{N} \overline{D}(i,j)}{N^2 \sum_{i=1}^{N} \sum_{j=1}^{N} \overline{D}^2(i,j) - \left(\sum_{i=1}^{N} \sum_{j=1}^{N} \overline{D}(i,j) \right)^2} \qquad (13)$$

$$o = \frac{\left(\sum_{i=1}^{N} \sum_{j=1}^{N} R(i,j) - s \sum_{i=1}^{N} \sum_{j=1}^{N} \overline{D}(i,j) \right)}{N^2} \qquad (14)$$

Let the best matched domain block is labeled with index d. The main computational load of fractal image coding is to look for the proper fractal codes s, o, t and the index d of the best matched domain block. For the fractal decoding, the transformation with the fractal codes is applied iteratively on an arbitrary initial image to obtain the original image.

Fig. 3 The eight symmetries

3. THE PROPOSED SCHEME

This section proposes our scheme of automatic image authentication and recovery. First, we present the fractal encoding of ROI and watermark insertion. And then, we describe the watermark extraction and image recovery. The procedure of our proposed scheme is shown in Fig.4. We define the region of importance (ROI) as the important region to be protected. The ROI fractal codes are obtained by fractal encoding for ROI. Then the ROI fractal codes and the watermark are inserted into the original image to get the watermarked image. After receiving a watermarked image, one can extract the watermark to detect and localize the altered region. For the altered region, we can perform image recovery by fractal decoding or image inpainting.

(a) (b)
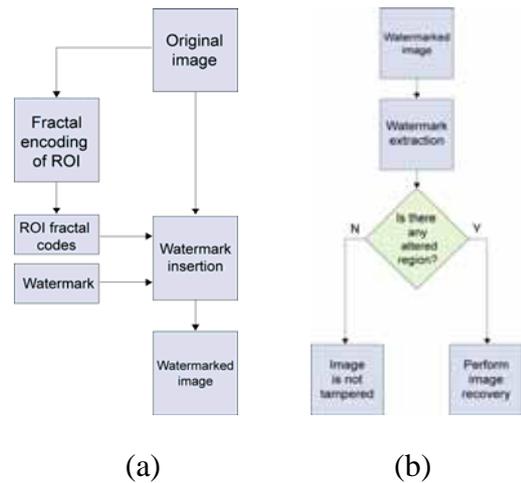
Fig. 4 (a) Fractal encoding of ROI and watermark insertion (b) Watermark extraction and image recovery

3.1 Fractal Encoding of ROI and Watermark Insertion

(A)Fractal Encoding of ROI

Because ROI is the important region to be protected, we encode it by fractal encoding in order to reconstruct the image with high quality. Fig.5 shows an example of Lena's ROI denoted by the white region. When we choose the ROI, we can find the corresponding range blocks of ROI, as shown in Fig.6.

Fig. 5 An example of Lena and its ROI
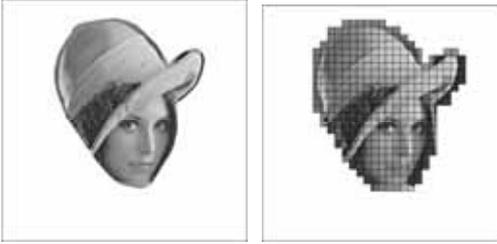
Fig. 6 Lena's ROI image and its corresponding range blocks of ROI

Here, we define two binary image data F_{so} and F_{td} whose size equal to the size of original image. Initially, the values of F_{so} and F_{td} are all set to zero. We partition F_{so} and F_{td} into blocks $(F_{so})_r$ and $(F_{td})_r$ of size $(I \times J)$ pixels, respectively. The corresponding block $(F_{so})_r$ and $(F_{td})_r$ is respectively denoted by

$$(F_{so})_r = \left\{ (F_{so})_{iI+a,jJ+b} : 0 \le a \le I-1; 0 \le b \le J-1 \right\} \quad (15)$$

$$(F_{td})_r = \left\{ (F_{td})_{iI+a,jJ+b} : 0 \le a \le I-1; \ 0 \le b \le J-1 \right\} \quad (16)$$

Then, we perform fractal encoding for all the range blocks of ROI, and put the fractal codes of s and o into the corresponding block of F_{so}, and put the fractal codes of t

and the index d of the best matched domain block into the corresponding blocks of F_{td}. In order to recover the block of ROI, when ROI is tampered or damaged, the fractal codes of the block of ROI should be embedded into other different block of the image. This can be simply achieved by a function of permutation. We divide F_{so} and F_{td} into four parts, as shown in Fig.7(a), and then perform a simple permutation, as shown in Fig.7(b). Note that other sophisticated permutation may be used for the permutation of F_{so} and F_{td}.

(a)   (b)
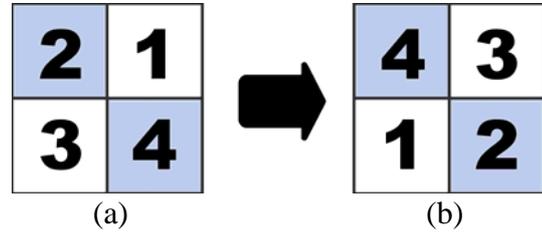
Fig. 7(a) original F_{so} and F_{td}
(b) permutated F_{so} and F_{td}

(B) Watermark Insertion

The overall procedure of watermark insertion is shown in Fig.8. After performing fractal encoding of ROI, we will insert the ROI fractal codes and the watermark into the original image. Consider a grayscale image O of $M_X \times N_X$ pixels, and let each k-bit pixel $O_{i,j}$ of the original image O be expressed as

$$O_{i,j} = (O_k, ..., O_2 O_1) \quad 0 \le i \le M-1, 0 \le j \le N-1 (17)$$

where O_1 is least-significant-bit(LSB), O_2 is LSB2, O_3 is LSB3, and O_k is MSB. For the original image O, we form the corresponding image $\tilde{O}$ where the image $\tilde{O}$ equals to the image O except that the least-significant-bit-two(LSB2) is set to zero. Then, we insert overall F_{so} into LSB2 plane and insert only $(F_{td})_r$ of ROI blocks into LSB3 plane to form another image X. When we get the image X, based on

Wong's scheme we insert watermark into the LSB of the image X to get the watermarked image X^w.
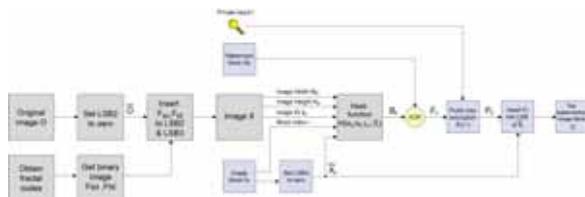
Fig. 8 Watermark insertion procedure

3.2. Watermark Extraction and Image Recovery

(A) Watermark Extraction

Fig.9 shows the watermark extraction procedure. We extract LSB2 plane and LSB3 plane of the watermarked image to get the binary data F_{so} and F_{td} respectively, and retrieve fractal codes of s, o, t and the domain block index d. We also extract the watermark from the watermarked image. After watermark extraction, we can determine whether the watermarked image has been altered or not. If the watermarked image has not been altered, we will extract the original watermark. For example, Fig.10(a) shows a watermarked image for the typical image "Lena" and Fig.10(b) shoes the extracted watermark which is the same as the original watermark. If the watermarked image has been altered, we will localize the altered region. Fig.10(c) shows a watermarked image with alteration and Fig.10(d) shows the extracted watermark which indicates the altered region marked by black dots.
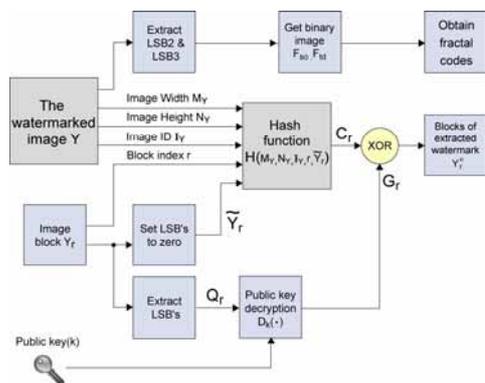
Fig. 9 Watermark extraction procedure

Fig. 10 (a) watermarked image without alteration (b) the extracted watermark for (a) (c) watermarked image with alteration (d) the extracted watermark for (c)

(B) Image Recovery

Before performing image recovery, we must automatically localize the altered region from the extracted watermark. Let a difference watermark be given by $M = W \oplus Y^0$ which is the difference between original watermark W and extracted watermark Y^0. Fig.11(a) shows the difference watermark for Fig.10(b) and (d). By simple processing the difference watermark shown in Fig.11(a), we can obtain the altered region shown in Fig.11(b).
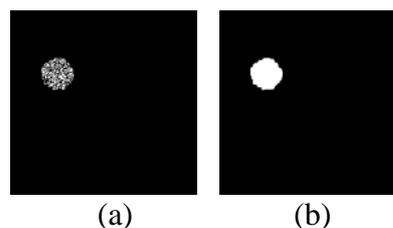
Fig. 11 (a) difference watermark (b) altered region

After we localize the altered region, we are ready to perform image recovery. Image recovery is done by fractal decoding or image inpainting, depending on whether there exists a fractal code for the block in altered region. The procedure of image recovery is shown in Fig.12. The block is which fractal codes exist is recovered by fractal decoding, and the block in which no

fractal codes exist is recovered by image inpainting.
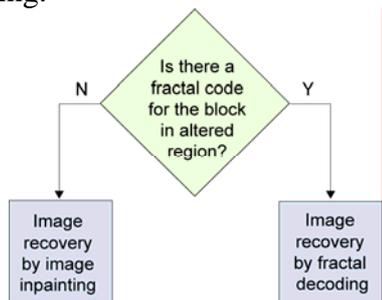
Fig. 2 Image recovery procedure

4. EXPERIMENT RESULTS

To validate the proposed scheme, some experiments have been made and the PSNR is measured for evaluation. First, the gray image "Lena" of size 256×256 is used, as shown in Fig.13(a). We select ROI as shown in Fig.13(b) and perform fractal encoding for all the range blocks of ROI. Here the size of range block is 4×4 and the fractal codes include 8-bit s, 8-bit o, 3-bit t and 12-bit d. We insert the ROI fractal codes and the watermark of Fig.13(c) to obtain the watermarked image shown in Fig.13(d). It is seen that the watermarked image is visually distinguishable from the original image. If we extract the watermark from the watermarked image in Fig.13(d) which has not been tampered, we can get the correct original watermark shown in Fig.13(e) which is the same as Fig.13(c). Fig.14(a) to Fig.14(c) show the images formed by some alteration in the watermarked image, and we can extract the watermark, as shown in Fig.15(a) to Fig.15(c),respectively. And Fig.15(d) to Fig.15(f) show the recovered images with high PSNR.

(a)   (b)   (c)

(d)   (e)
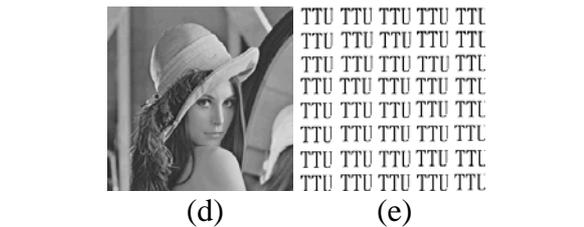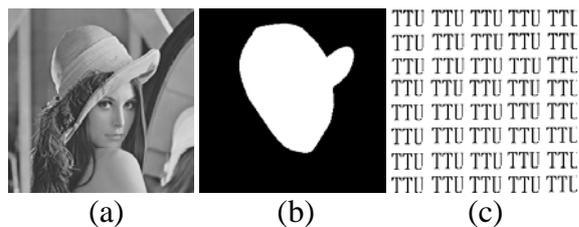
Fig. 3 (a) original image (b) ROI (c) watermark (d) watermarked image, PSNR=42.1102db (e) extracted watermark

(a)   (b)   (c)
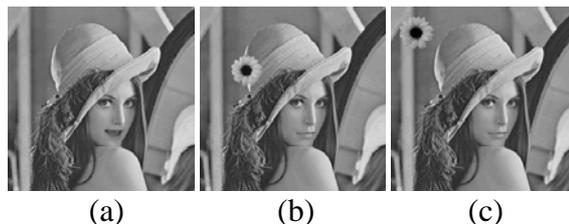
Fig. 4 (a) Lena's mouth in this image has been changed (b) the flower is pasted on the hat (c) the flower is pasted on background

(a)   (b)   (c)

(d)   (e)   (f)
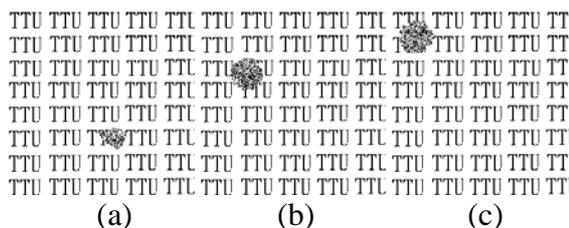
Fig. 5 (a) extracted watermark from Fig.14(a). (b) extracted watermark from Fig.14(b). (c) extracted watermark from Fig.14(c). (d) recovered image for Fig.14(a), PSNR=42.1411db. (e) recovered image for Fig.14(b), PSNR=41.7521db (f) recovered image for Fig.14(c), PSNR=41.1472db

Moreover, we select another gray image "Door" of size 256×256, choose some ROI regions and get the watermarked image, as shown in Fig.16(a)(b)(c), respectively. Now, consider the case we remove the shoes in the watermarked image of Fig.16(c) to form an altered image of Fig.17(a). If we merely use image inpainting to recover the destroyed region, we get the image of

Fig.17(b). Note that the shoes are disappeared. Since the shoes lies in the ROI region, we can use fractal coding and image inpainting to get the recovered image of Fig.17(c). Comparing Fig.17(b) and Fig.17(c), it is seen that our scheme using fractal coding and image inpainting coding could recover image more correctly.

(a)   (b)   (c)
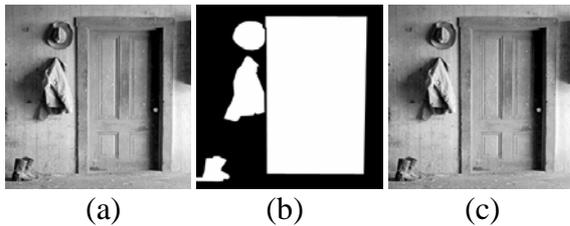
Fig. 6 (a) original image (b) ROI regions (c) watermarked image, PSNR=39.6798db

(a)   (b)   (c)
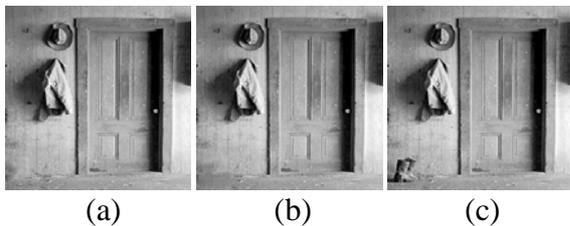
Fig. 17(a) altered image (b) recovered image obtained by only using image inpainting, PSNR=35.1308db (c) recovered image obtained by using our scheme, PSNR=38.5425db

5. CONCLUSION

In this paper, we propose a scheme of automatic image authentication and recovery, in which the altered region of the image can be detected and recovered in a fully automatic fashion. Fragile watermarking is used for image authentication to detect and localize the altered region of the image. To achieve high quality image recovery, we choose the region of importance (ROI) in the image and perform fractal encoding for the blocks of ROI. And then we insert the fractal codes of ROI and the watermark into the original image to obtain the watermarked image. By watermark extraction, we can determine whether the watermarked image has been tampered or not. If the watermarked image has been tampered, we can automatically localize the altered region, and apply fractal decoding and image inpainting for image recovery. In the altered region, the blocks in which fractal codes exist are recovered by fractal decoding and the remaining blocks are recovered by image inpainting. Finally, experimental results are given to show the effectiveness of the proposed scheme.

6. REFERENCES

[1] P.S.L.M. Barreto, H. Y. Kim and V. Rijmen, "Toward secure public-key blockwise fragile authentication watermarking," IEEE Proceedings- Vision, Image and Signal Processing, vol.149, no.2, pp.57-62, Apr. 2002.

[2] J. Fridrich, M. Goljan and A.C. Baldoza, "New fragile authentication watermark for images," Int. Conference on Image processing, vol.1, pp.446-449, 2000.

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

[4] P. W. Wong, "A Public Key Watermark for Image Verification and Authentication," in Proc. IEEE Int. Conf. Image Processing, vol.1, pp. 445-459, Chicago, USA, Oct, 1998.

[5] P. W. Wong, "A Watermark for Image Integrity and Ownership Verification," Proc. IS&T PIC Conf., (Portland OR), May 1998 (also available as Hewlett-Packard Labs. Tech. Rep. HPL-97-72, May 1997).

[6] M. M. Yeung and F. Mintzer, "An Invisible Watermarking Technique for Image Verification," Int. Conf. Image Processing, vol.1, pp. 680-683, 1997.

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

[8] J. Fridrich and M. Goljan "Images with self-correcting capabilities," Int. conference on Image Processing, vol.3, pp.792-796, 1999.

[9] P. L. Lin, P. W. Huang and A. W. Peng, "A fragile watermarking for image authentication with localization and recovery," Int. Symposium on Multimedia Software Engineering, pp.146-153, 2004.

[10] M. Ashikhmin, "Synthesizing natural textures," Proc. ACM Symp. Interactive3D Graphics, pp.217–226, Mar. 2001.

[11] A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," Proc. ACM Conf. Computer Graphics (SIGGRAPH), pp. 341–346, Aug. 2001.

[12] A. Criminisi, P. Perez and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," IEEE Trans. on Image Processing, vol.13, no.9, pp.1200–1212, Sept. 2004.

[13] M.F Barnsley, Fractal Everywhere, Academic Press, Inc., San Diego, 1988.

[14] A. E. Jacquin, "Fractal image coding: A Review," Processing of the IEEE, vol.81, no.10, pp.1451-1465, Oct. 1993.

[15] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," IEEE Transactions on Image Processing, vol.1, no.1, pp.18-30, Jan. 1992.