Self-embedding Fragile Watermarking Scheme Based on Bicubic Prediction

Fan Chen[†], Hongjie He[*], and Hongxia Wang[†]
[†] Sichuan Key Lab of Signal and Information Processing, Southwest Jiaotong University, Chengdu
[*] Information Securityand National Computing Grid Lab, Southwest Jiaotong University, Chengdu
E-mail: Fchen@swjtu.edu.cn Tel: +86-28-87601024

***Abstract*—To improve the quality of recovered images, a self-embedding fragile watermarking scheme is proposed based on the bicubic prediction. To take into account the PFA and the watermark payload, the 6-bit recovery data of a 2×2 block and the 8-bit key-based data of a 4×4 block are generated and inserted in the other 2×2 block and 4×4 block based on secret key, respectively. The validity of a 2×2 image block is determined by combining the recovery data with the key-based data. To improve the recovery quality, the recovery method based on bicubic prediction is designed to reconstruct the invalid blocks whose recovery watermark embedded in the other block is also destroyed. Simulation results demonstrate that the proposed scheme allows image recovery with an acceptable visual quality (PSNR ≈ 25 dB) up to 75% tampering.**

I. Introduction

Image fragile watermarking is designed to achieve digital image content authentication by imperceptibly embedding additional information into the host image [1, 2]. To further provide the tampering proofing, some fragile watermarking schemes inserted the compressed version of it to recover approximately the original content in the tampered regions. They are called as the self-embedding watermarking.

Self-embedding fragile watermarking techniques have received great attention in recent years. To achieve the self-recovery, the compressed version (recovery watermark) of a block was embedded into the other block in the host image [3]. However, this strategy makes it difficult to detect and localize the possible tampering. To resolve the tamper detection problem of self-embedding, Lin *et al*. [4] proposed that the validity of an image block was determined by the additional authentication data. Specifically, the watermark payload consisted of authentication data as well as recovery data. The authentication data of a block were embedded in the block itself and used to determine the validity of it. This method of tamper detection had been adopted in a number of self-embedding approaches [5], [6] and [7]. However, as pointed in [8], this detection method was vulnerable to the collage attack [9] due to the fact that the authentication watermark was block-wise independent. To address this problem, He *et al* [10] proposed a self-recovery fragile watermarking using block neighborhood tampering characterization. In He's method [10], the 6-bit recovery data and 2-bit key-based data of a 2×2 block were generated and embedded into the least significant bit (LSB) planes of the other block based on the secret key. The validity of a block was determined by comparing the number of inconsistent blocks in the 3×3 block-neighborhood of the block with that of its mapping block. The 3×3 block-neighborhood was also used to recover the tampered blocks whose feature hidden in another block is corrupted. He's scheme [10] could identify and recover the tampered 2×2 blocks even though the test image was maliciously modified by the collage attack. However, He's method [10] was still vulnerable to the improved constant-average attack. Moreover, the quality of the recovered image was generally poor if the large portions of an image were tampered. There are two possible reasons for this problem. Firstly, the probability of false acceptance (PFA) increased with the increase of the tampering ratios, and secondly the tampered block could not be reconstructed if all blocks in its 3×3 block-neighborhood were invalid.

In this work, we proposed an improved self-embedding watermarking scheme with superior recovery quality. To take into account the PFA and the watermark payload, the 6-bit recovery data of a 2×2 block and the 8-bit key-based data of a 4×4 block are generated and inserted in the other 2×2 block and 4×4 block based on secret key, respectively. The validity of a 2×2 block is determined by combining the recovery data with the key-based data. The recovery method based on bicubic prediction is designed to reconstruct the invalid blocks whose recovery-data hidden in another block is also corrupted. Experimental results confirm that recovery is possible with high probability and acceptable visual quality (PSNR ≈ 25dB) for up to 75% tampering.

II. Proposed Self-embedding Watermarking Scheme

This section describes in detail the proposed self-embedding fragile watermarking method. To improve the performance of tamper detection, the 8-bit key-based data of a 4×4 block are generated and embedded in the other 4×4 block based on secret key. The bicubic prediction is used to improve the recovery quality especially for a larger tampering ratios. The proposed algorithm is described through three stages: watermark embedding, tamper detection, and bicubic prediction recovery.

A. Watermark Embedding

This work is supported in part by the National Natural Science Foundation of China (60970122), by the Research Fund for the Doctoral Program of Higher Education (20090184120021), by the Fundamental Research Funds for the Central Universities (SWJTU09CX039, SWJTU10CX09), and by Beijing Key Laboratory of Advanced Information Science & Network Technology and Railway Key Laboratory of Information Science & Engineering. (No. XDXX1007).

To take into the watermark payload and the detection performance account, the 6-bit recovery data of a 2×2 block and the 8-bit key-based data of a 4×4 block are generated and inserted in the other 2×2 block and 4×4 block in the host image based on secret key, respectively. Suppose the size of a host image X is $4m×4n$ pixels, so the number of 2×2 blocks and 4×4 ones in the host image are the $N=4m×n$ and $M=m×n$, respectively. The embedding process consists of four steps.

Step 1: Mapping sequences. According to the secret key, two mapping sequences $\Psi = (\varphi_1, \dots, \varphi_N)$ of the integer interval [1, N] and $\Lambda = (\sigma_1, \dots, \sigma_M)$ of the integer interval [1, M] are obtained. The detailed procedure of generating mapping sequence refers to Ref. [10].

Step 2: Recovery-data embedding. The image X is partitioned into non-overlapping 2×2 blocks $X=(X_1,\dots,X_N)$. Each 2×2 block X_i can *be* expressed as,

$$X_i = \begin{bmatrix} x_{i1} & x_{i2} \\ x_{i3} & x_{i4} \end{bmatrix}, i = 1,2, \dots, N \tag{1}$$

For each block X_i, the 6-bit recovery data $W_i^R = (w_{i1}^R, \dots, w_{i6}^R)$ are obtained by encrypting the average intensity by truncating the two LSB planes of each pixel in block X_i, and hidden in the mapping block X_p, where $p= \varphi_i$. The block $X_p^{'} = (x_{p1}^{'}, \dots, x_{p4}^{'})$ is computed by,

$$x_{pj}^{'} = \begin{cases} 4\lfloor x_{pj}/4 \rfloor + 2w_{i(j+4)}^R + w_{ij}^R, & j = 1,2 \\ 4\lfloor x_{pj}/4 \rfloor + 2w_{ij}^R, & j = 3,4 \end{cases} \tag{2}$$

where $\lfloor a \rfloor$ denotes the largest integer less than or equal to a and *mod(,)* is the modulo operation. Performing this step for all blocks in X, we can obtained the image $X^{'} = (X_1^{'}, \dots, X_N^{'})$, in which the recovery-data has been embedded.

Step 3: key-based data generating. The image $X^{'}$ is partitioned into 4×4 blocks $X^{'} = (X_1^{'}, \dots, X_M^{'})$. Each 4×4 block $X_l^{'}$ is expressed as,

$$X_l^{'} = \begin{bmatrix} x_{l1}^{'} & \cdots & x_{i4}^{'} \\ \vdots & \ddots & \vdots \\ x_{l13}^{'} & \cdots & x_{l16}^{'} \end{bmatrix}, l = 1,2, \dots, M \tag{3}$$

For each 4×4 block $X_l^{'}$, the 8-bit key-based data $W_l^K = (w_{l1}^K, \dots, w_{l8}^K)$ are computed as,

$$w_{lj}^K = \mathrm{mod}((X_l^{'})_b * A_l, 2),\ j = 1, \dots, 8 \tag{4}$$

where $(X_l^{'})_b$ is binary code of the intensity by truncating the two LSB planes of each pixel in block $X_l^{'}$, A_l is key-generated random bit pattern, different for each block $X_l^{'}$, and the *-operator denotes the matrix multiplication. This implies that for any change of the content of $X_l^{'}$, each key bit is flipped with probability 1/2 [6].

Step 4: key-based data embedding. Setting $q= \sigma_l$, the key-based data $W_l^K = (w_{l1}^K, \dots, w_{l8}^K)$ of block $X_l^{'}$ are hidden in the partial pixels in the mapping block $X_q^{'}$. The watermarked block $Y_q=(y_{q1},..,y_{q16})$ is generated by,

$$y_{qj} = \begin{cases} 2\lfloor x_{qj}^{'}/2 \rfloor + w_{l(j-4)}^K, & j = 5,6,7,8 \\ 2\lfloor x_{qj}^{'}/2 \rfloor + w_{l(j-8)}^K, & j = 13,14,15,16 \\ x_{qj}^{'}, & otherwise \end{cases} \tag{5}$$

B. *Tamper Detection*

Suppose ***Z*** represent a tested image, which can be a distorted watermarked image or unaltered one. A binary sequence $T=(t_i|i=1,2,\dots,N)$ called the tamper detection mark (TDM) is used to represent the location of tampering, where N is the number of 2×2 blocks in the test image Z. The tamper detection procedure includes the following steps.

Step 1: Recovery data matching. According to the test image Z and the mapping sequence $\Psi = (\varphi_1, \dots, \varphi_N)$, the recovery-data match-matrix $D^R = (d_1^R, \dots, d_N^R)$ is calculated by,

$$d_i^R = \begin{cases} 0 & , \quad if\ W_i^R = E_p^R \\ 1 & , \quad \text{otherwise} \end{cases} \tag{6}$$

where W_i^R is the computed the recovery data of the 2×2 block Z_i, and E_p^R is the extracted watermark data from the corresponding mapping block Z_p (where $p = \varphi_i$). The recovery-data TDM $T^R = (t_i^R|i=1,2,\dots,N)$ is obtained by the block-neighborhood detection method proposed in [10]. That is,

$$t_i^R = \begin{cases} 1 & , \quad if\ (d_i^R = 1)\&(\Gamma_i \geq \Gamma_p) \\ 0 & , \quad otherwise \end{cases} \tag{7}$$

where $p= \varphi_i$, Γ_i and Γ_p denote the number of nonzero pixels that are adjacent to the i^{th} and p^{th} pixel in the D^R, respectively.

Step 2: Key-based data matching. Similarly, the key-based data TDM $T^A = (t_l^A|\ l=1,2,\dots,M)$ is obtained by the test image Z and the mapping sequence $\Lambda = (\sigma_1, \dots, \sigma_M)$. Note that $t_l^A = 1$ indicates that the 4×4 block Z_l is invalid; otherwise, it is valid. To mark the validity of each 2×2 block Z_i in the test image Z, the key-based data TDM $T^K = (t_i^K|i=1,2,\dots,N)$ is obtained. The value of t_i^K equals to that of t_l^A if the 2×2 block Z_i belongs to the 4×4 block Z_l.

Step 3 Tamper Detection: Setting $\Omega = (\omega_1, \dots, \omega_N)$, where $\omega_i = t_i^R + t_i^K$. The value of ω_i is an integer ranging from 0 to 2 since the value of t_i^B and t_i^T is 0 or 1. Let ξ_i denotes the sum of eight pixels that are adjacent to the pixel ω_i in the Ω. The TDM $T=(t_i|i=1,2,\dots,N)$ is obtained by

$$t_i = \begin{cases} 1 & , \quad if\ (\omega_i + \xi_i) > 4 \\ 0 & , \quad otherwise \end{cases} \tag{8}$$

C. *Bicubic Predictive Recovery*

After tamper detection, all 2×2 blocks in test image are marked as either valid or invalid. The proposed recovery procedure is only for the invalid blocks. The invalid blocks can be classified into two categories: data-destroyed and data-reserved invalid blocks. The former denotes the tampered block whose recovery data inserted in the corresponding mapping block is also destroyed, and the latter that tampered block whose recovery data is valid. For data-destroyed invalid blocks, the recovery method based on bicubic prediction is designed to reconstruct them.

(1) *Initialization*. According to the tested image Z, the TDM T and the mapping sequence $\Psi = (\varphi_1, \dots, \varphi_N)$, the recovered image $R = \{R_i | i = 1, \dots, N\}$ is initialized by,

$$R_i = \begin{cases} A_{ve}\left(E_p^R\right) + mod(Z_i, 4), & if\ (t_i = 1)\&\left(t_p = 0\right) \\ Z_i, & otherwise \end{cases} \tag{9}$$

where $A_{ve}\left(E_p^R\right)$ denotes the reconstructed average intensity by the extracted recovery data E_p^R from the associated block *Zp* (where $p= \varphi_i$) [10]. At the same time, the marked image $H = \{H_i | i = 1, \dots, N\}$ is obtained by the following expression,

$$h_{ik}=\begin{cases}2 \quad , & t_i=0\\ 1 \quad , & (t_i=1)\&(t_p=0)\\ 0 \quad , & (t_i=1)\&(t_p=1)\end{cases}, k=1,2,3,4 \qquad (10)$$

where p= φ_i. H_i=0 implies that the corresponding 2×2 block Z_i is not recovered.

Fig.1 illustrates an initialization procedure. Fig. 1(a) is a part of TDM *T*, in which two shaded blocks represent the data-destroyed invalid blocks. From (10), we can obtain the marked image *H* corresponding to Fig. 1(a), as shown in Fig. 1(b). The recovered image *R* obtained by (9) is shown in Fig. 1(c), in which the two shaded blocks are not recovered successfully.

(2) *Predictive mask*. The data-destroyed invalid block Z_i will be predicted by the valid pixels in the area spanned by the predictive mask. We select the predictive mask is 4×4 pixels since the block size is 2×2 pixels. For block Z_i, the pixels in the area spanned by the predictive mask is denoted as $S_i=(s_{in}|n=1,\ldots,16)$, shown in the gray region in Fig. 1 (c).

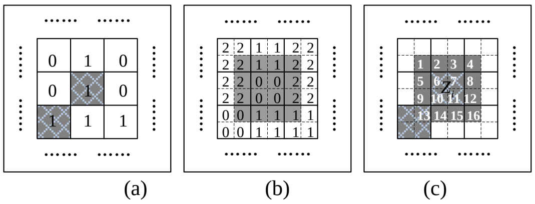


Fig.1 An instance of initialization and the predictive mask
(a) TDM *T*, (b) Marked image *H*, and (c) Recovered image *R*

(3) *Predictive coefficients*. Predictive coefficients include two parts. The adaptive weights $\Theta_i=(\vartheta_{in}|n=1,\ldots,16)$ derived from the mark matrix *H*, as shown in the gray region in Fig. 1(b). The fixed coefficients of a pixel in a 2×2 block are the inverse of the distance between a pixel in the predictive mask and the pixel. Four fixed-coefficient metrics corresponding to each pixel in a 2×2 block denote as $\Phi_k=\{\phi_{kn}|n=1,\ldots,16\}$ $(k=1,2,3,4)$ and are shown in Fig.2

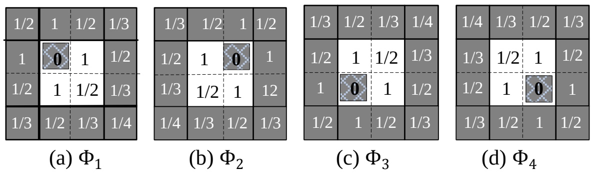


Fig. 2 Fixed coefficients corresponding to each pixel in a block of 2×2 pixels

(4) R*ecovery*. For a recovered blocks R_i, if the four pixels in the corresponding mark block H_i are zero, the pixels in the block R_i=(r_{i1},..., r_{i4}) is updated by,

$$r_{ik}=\left[\sum_{n=1}^{16}\left(\frac{\omega_{in}\phi_{kn}}{\sum_{j=1}^{16}\omega_{in}\phi_{kn}}\right)s_{in}\right], k=1,2,3,4 \qquad (11)$$

At the same time, the four pixels in H_i =(h_{i1},..., h_{i4}) is updated to 1. Each block in the recovered image *R* should performed this step.

(5) If $\exists\, H_i=0$ in the mark matrix *H* updated by *recovery* step, repeat *steps* 2~4 until the value of each pixel in *H* is not zero.

III. Experimental Results

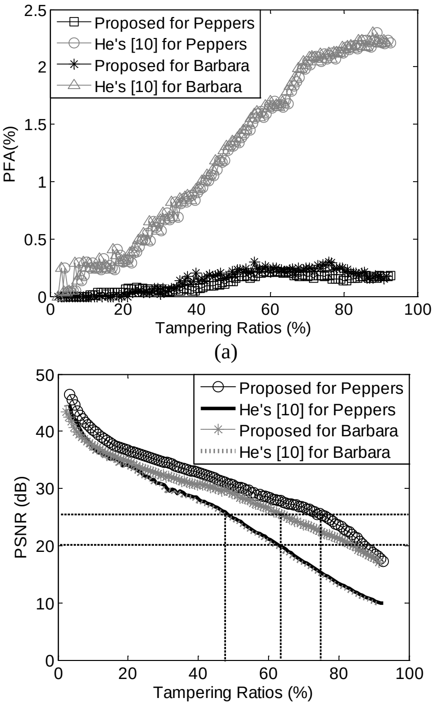
We conduct numerous experiments to demonstrate the effectiveness of the proposed scheme and compare with the method in [10] on the performance of tamper restoration. For quantitative evaluation, several measurements are introduced. (1) Restoration quality: PSNR between the recovered image and watermarked one, (2) tampering ratio $r_t=(N_T/N)\times 100\%$ and (3) The PFA $P_{fa}=(1-N_{td}/N_T)\times 100\%$, where *N* denotes the number of image blocks with size of 2×2 pixels in the tested image, N_T and N_{td} denote the number of actually tampered blocks and that of tampered blocks which are correctly detected, respectively.

Self-embedding fragile watermarking techniques enable the detection of tampering or replacement of a watermarked image. The distinction mainly lies in the tamper localization accuracy and the quality of recovered images [10]. The quality of a recovered image highly depends on the size of tampered regions, and the complexity of image content also affects the quality of the recovered image. Two watermarked images of size 512×512 pixels, a rough Barbara and a smooth Peppers, are used to demonstrate the performance of the self-embedding schemes in general tampering. For each test image, the 2×2 image blocks were randomly modified with different tamper ratios, and the tampered blocks were detected and recovered. Fig. 3 shows the performance comparison of experimental results under general tampering with different tampering ratios by the proposed and He's [10] schemes.

Fig. 3(a) reveals that the PFA of He's method [10] increases along with the increase of the tampering ratio. The PFA of He's method is more than 2% as the tampering ratio is up to 70%. On the contrary, the PFA of the proposed scheme is kept small (smaller than 0.5%) even if the tampering ratio is up to 90%. The low PFA will transform to a high quality image recovery. Moreover, Fig. 3(a) shows that PFAs for the Barbara and Peppers images by the same watermarking scheme are almost the same. This implies that the complexity of the image content does not have much impact on the performance of tamper detection.

Recovery quality by the proposed scheme is better than that by the He's method [10], as seen from Fig. 3(b). This may be due to the low PFA and the bicubic predictive recovery of the proposed scheme. The PSNRs of the proposed scheme are higher than those of He's scheme. The PSNRs of the proposed scheme are higher than 25 dB as long as the tampered ratio is no more than 65% of the Barbara or 75% for the Peppers. On the other hand, the recovery quality of Peppers image is better than that of Barbara one in the same tampering ratio for the proposed and He's schemes. This implies that the complexity of image content have impact on the performance of tamper recovery.

Fig. 4 shows the tampered images and their recovered ones. Fig. 4(a) and 4(b) are the tampered images with 15.25% and 77.24% tampering ratios, respectively. The recovered images of Fig. 3(a) by the proposed and He's schemes, shown in Fig. 3(c) and 3(d), have the PSNR of 37.91 dB and 36.67dB, respectively. As the tampering ratio is larger, the proposed scheme exhibits much better tamper recovery performance than He's scheme. For Fig. 3(b), PSNR of the recovered image by the proposed scheme is 24.37 dB, which is about 10 dB higher than that by He’s scheme. These results indicate that the tampered image can be recovered by the proposed scheme with an acceptable visual quality (25 dB) even the tamper ratio is up to 75% of the host image.



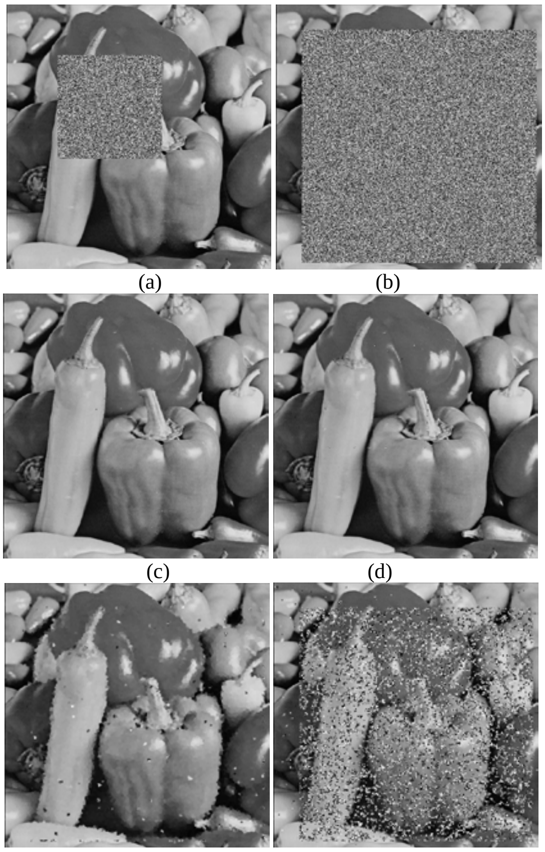
(a)

(b)

Fig. 3 Performance comparison under general tampering with different tampering ratios (a) PFA and (b) PSNR

IV. Conclusions

This work has proposed a improved self-embedding fragile watermarking scheme based on bicubic prediction. The 6-bit recovery data of a 2×2 block and the 8-bit key-based data of a 4×4 block are generated and randomly inserted in the other 2×2 block and 4×4 block, respectively. The validity of a 2×2 block is determined by combining the recovery data with the key-based data. The recovery method based on bicubic prediction is designed to reconstruct the data-destroyed invalid blocks. Experiment results have demonstrated the superiority of the proposed scheme in comparison to He's method. Future research includes extending this approach to resist mild distortion such as JPEG compression.



(a) (b)

(c) (d)

(e) (f)

Fig. 4 Tampered images and recovery ones (a) tampered image with 15.25% tampering ratio, (b) tampered image with 77.24% tampering ratio; The recovered images of (a) by, (c) Proposed: 37.91dB, and (d) He [10]: 36.67 dB; The recovered images of (b) by, (e) Proposed: 24.37 dB, and (f) He [10]: 14.13 dB

References

[1] P.W. Wong, N. Memon, "Secret and public key image watermarking schemes for image authentication and ownership verification," *IEEE Trans on Image Processing*, pp.1593-1601, Oct. 2001.

[2] Anthony T.S.Ho, Xunzhan Zhu, Jun Shen, and Pina Marziliano. "Fragile watermarking based on encoding of the zeros of the z-transform," *IEEE Trans. information Forensics and security*, vol. 3, pp.567-569, March 2008

[3] J. Fridrich and M. Goljan, "Protection of digital images using self-embedding," *Proc. Symp. Content Security and Data Hiding in Digital Media*, NJIT, NJ, May 1999.

[4] P.L. Lin, C.K. Hsieh, P.W. Huang, "A hierarchical digital watermarking method for image tamper detection and recovery, " *Pattern Recognition,* vol. 38, pp. 2519–2529 , 2005.

[5] C._W Yang, J._J Shen, "Recover the tampered image based on VQ indexing," *Signal Processing,* vol. 90 , pp. 331-343, 2010

[6] X.Zhang, S.Wang, Z.Qian, G. Feng, "Reference Sharing Mechanism for Watermark Self-Embedding," *IEEE Trans. On Image Processing,* vol. 20, pp. 485- 495, Feb. 2011.

[7] Z. Qian, G.Feng, X. Zhang and S. "Wang, Image self-embedding with high-quality restoration capability," *Digital Signal Processing,* vol. 21, pp. 278-286, 2011

[8] H.J He, J.S Zhang, and F Chen, "Adjacent-block based statistical detection method for self-embedding watermarking techniques," *Signal Process*. vol. 89,pp.1557-1566, 2009

[9] J. Fridrich, M. Goljan, and N. Memon, "Cryptanalysis of the Yeung-Mintzer fragile watermarking technique," *J. Electronic Imaging* , pp.262–274, Nov. 2002.

[10] H.J. He, J.S. Zhang, and H.-M Tai, "Self-recovery fragile watermarking using block-neighborhood tampering characterization," *11th Intl. Workshop, IH 2009, Darmstadt, Germany*, June 2009