A Flexible Scheme of Self Recovery for Digital Image Protection

Zhenxing Qian[1], Lili Zhao[2]

[1]School of Communication and Information Engineering, Shanghai University, Shanghai 200072, China

ABSTRACT

This paper proposes a self-embedding method with flexible restoration capability. We present a method of encoding DCT coefficients of each block into reference bits for self-embedding. In order to improve the restoration capability, we classify the selected coefficients into three types, and assign different bits for each type to generate reference bits. The generated reference bits are embedded into the original image, along with authentication bits derived from the content by hash function. On the receiver side, after identifying tampered blocks by authentication bits, reference bits are extracted from the reserved blocks to approximately reconstruct the principal content of the lost information. Three cases of flexible quality restoration are analyzed corresponding to different tampering rates. Even up to 75% of the protected image is tempered; the original content can still be recovered approximately.

Keywords: *Data Hiding, Self-Embedding, Content Restoration, Fragile Watermarking*

1. Introduction

As digital images are widely used today, many processing software emerge for common users, which make it easier to produce some unaware fakes on nature images [1], [2]. Because some parts of digital image may be replaced with fake information by an adversary, or suffer the information lost due to lossy channel condition, a number of intelligent methods capable of recovering the original content have been developed for the multimedia protection, such as methods in [3] and [4] for digital images, and methods in [5] and [6] for video contents. With these methods, the original content of the tampered parts can be identified and recovered using hidden data [7]. This kind of methods are referred to as self-embedding [8], which provides solutions for image protection, which can identify the tampered regions and roughly restore the original content.

Self-embedding was first proposed by Fridrich in [8], which embeds exactly 64 encoded bits from each 8×8 block into LSB of another block. When some contents are detected to be tampered, the hidden bits are extracted to reconstruct a rough image for reference. With similar ideas, many improved methods appear continuously [9-17]. In [12], an inpainting assisted self recovery method was proposed, which decreased reference bits and improved recovered quality. In [13], a reversible self-embedding method is proposed with watermark generated from entire content of the original image which can exactly restore the tampered regions without any errors. Nevertheless, limitation of this method is the tampered portion must not be larger than 3.2%. In [15], Zhang proposed a flexible self embedding scheme capable of recovering the original principal content to the extent of 54%. Most of these methods have good qualities of self recovery. However, restoration capabilities are not enough for some scenarios when large areas are substituted by fake content.

In this paper, we propose a feasible self-embedding method with flexible and large-area restoration capability. Tampered regions can accurately be detected and corresponding contents can approximately be recovered even we have only 25% blocks reserved. Recovery qualities of tampered regions are flexible according to different tampering rates. We classify tampering rates into three scales that are 0~50%, 51%~66.6%, and 66.7%~75%. When less tampering happens, we obtain the better recovery quality. Compared with the method in [15], the proposed method can recover more tampered areas, and provides a better recovery quality.

2. Self-Embedding Procedure

2.1 Data Embedding

We denote the size of image **M** as $N_1 \times N_2$, and the total number of pixels as N ($N = N_1 \times N_2$). Assuming both N_1 and N_2 are multiples of 8, divide the image into R non-overlapped 8×8 blocks, where $R = N/64$. Transform each block into frequency domain using DCT, and quantize the coefficients by quantization table with quality factor 50. Choose the first 21 coefficients of each block in zigzag order, as shown in Figure 1. Encode each coefficient into binary bits with a length assignment table which is defined in Figure 2.

During encoding, each one of the selected coefficients is represented as a binary sequence of E bits with Eq. (1) and (2),

$$b_k = \left\lfloor C_R / 2^{E-e} \right\rfloor \bmod 2, \quad e = 1, 2, ..., E \qquad (1)$$

$$C_R = \begin{cases} 0 & , \text{ if } C \le R_{\min} \\ \text{round}(C) - R_{\min} & , \text{ if } R_{\min} < C < R_{\max} \\ R_{\max} - R_{\min} & , \text{ if } C \ge R_{\max} \end{cases} \qquad (2)$$

Where C represents the selected DCT coefficient and E the corresponding value listed in Figure 2. $R_{\min}$ and $R_{\max}$ are the minimum and maximum values of the range corresponding to coefficient's position in Figure 2.

Classify the chosen 21 DCT coefficients to three types $\{T_1, T_2, T_3\}$, where $\{T_1 \mid C_1 \sim C_3\}$ corresponds to the low frequencies, $\{T_2 \mid C_4 \sim C_{10}\}$ the middle, and $\{T_3 \mid C_{11} \sim C_{21}\}$ the high. The classification manner is shown in Figure 2 ~ Figure4. Thus, assigned bits for all types are 19 bits, 31 bits and 33 bits. For each block $B^{(i)}$ ($1 \le i \le N$), we denote these encoded bits as $W_1^{(i)}$, $W_2^{(i)}$, and $W_3^{(i)}$, which will be used as reference bits for $B^{(i)}$.

Generate a group of embedding keys $K = \{k_u^v \mid k_1^1, k_2^1, k_3^1, k_1^2, k_2^2, k_1^3\}$,

$$k_u^v = \left\lfloor u \times N / (5 - v) \right\rfloor + k_0 \qquad (3)$$

Where k_0 is a predefined integer. With these keys, we generate the connection between block $B^{(i)}$ and remote blocks using a mapping function,

$$L_i \leftarrow f(i, K) : l_u^v = \text{mod}(i + k_u^v, N) \qquad (4)$$

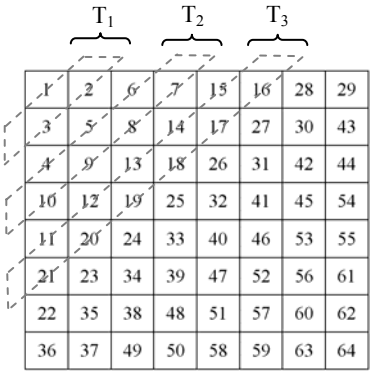


Figure 1 Zigzag order of DCT coefficients

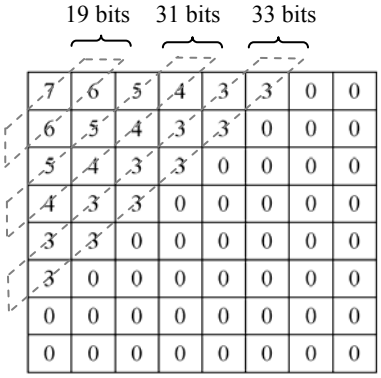


Figure 2 Length assignment for coefficients

7	6	5	4	0	0	0	0
6	5	4	0	0	0	0	0
5	4	0	0	0	0	0	0
4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 3 Length assignment table

7	6	0	0	0	0	0	0
6	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 4 Length assignment table

where $L_i = \{l_u^{v,(i)} \mid l_1^{1,(i)}, l_2^{1,(i)}, l_3^{1,(i)}, l_1^{2,(i)}, l_2^{2,(i)}, l_1^{3,(i)}\}$ denotes the positions of 6 remote blocks, and the symbol "←" the mapping relationship.

For each block, input 320 original bits of 5 MSB layers into a predefined hash function to generate 40 authentication bits, of which the hash function must have the ability that any change on the input will result in a different output. Denote the authentication bits of block $B^{(i)}$ as $H^{(i)}$.

$$H^{(i)} = hash(B^{(i)}) \qquad (5)$$

Then, the watermark to be embedded into $B^{(i)}$ is formed by 7 parts including 192 bits, denote as $S^{(i)}$,

$$S^{(i)} = W_1^{[l_1^{1,(i)}]} \mid W_1^{[l_2^{1,(i)}]} \| W_1^{[l_3^{1,(i)}]} \| W_2^{[l_1^{2,(i)}]} \|_2^{W[l_2^{2,(i)}]} \| W_3^{[l_1^{3,(i)}]} \| H^{(i)} \qquad (6)$$

Where "||" denotes the concatenation operation, Embed $S^{(i)}$ into three LSBs of $B^{(i)}$ by

$$\overline{B^{(i)}} = B^{(i)} - \mathrm{mod}(B^{(i)}, 8) + S^{(i)} \quad (7)$$

By repeating the embedding procedure until all the blocks are processed, we finally obtain the watermarked image.

This way, reference data corresponding to T_1 for each block is embedded into three remote blocks, T_2 into two remote blocks, and T_3 into one remote block. Usually, distortion caused by data hiding is imperceptible. Assuming that the original distribution of the LSB is uniform, about 50% pixels was modified. So, the PSNR is around 38 dB.

2.2 Content Restoration

Assuming that the size of received image **M'** maintains unchanged, we extract 192 hidden bits from each block. For example, the stream $S^{(i)}$ with192 bits extracted from block $B^{(i)}$ includes the reference bits and authentication bits. Separate authentication bits $H^{(i)}$ from $S^{(i)}$. Feed 320 bits of 5 MSBs of **M'** into the hash function defined in equation (5), which results in another 40 authentication bits $J^{(i)}$. Comparing the extracted authentication bits $H^{(i)}$ with the calculated $J^{(i)}$ by equation (8), block $B^{(i)}$ is judged as tampered if $AU^{(i)} \neq 0$, otherwise the block is reserved (not tampered), where $\oplus$ is the operation XOR.

$$AU^{(i)} = \sum_{u=1}^{40} H^{(i)}(u) \oplus J^{(i)}(u) \quad (8)$$

After identifying all reserved blocks, we extract hidden bits from these blocks. Use the secret key K to find the connections between each tampered block and the blocks containing its reference bits. Assuming three copies of reference bits $W_1^{(t)}$ for a tampered block $B^{(t)}$ were embedded in blocks $B^{(e)}$, $B^{(f)}$ and $B^{(g)}$; two copies of $W_2^{(t)}$ for $B^{(t)}$ in $B^{(j)}$ and $B^{(k)}$; and one copy of $W_3^{(t)}$ for $B^{(t)}$ in $B^{(p)}$, we can recover the tampered block $B^{(t)}$ using rules of restoration as follows.

Case 1: Condition: at least one of $\{B^{(e)}, B^{(f)}, B^{(g)}\}$ is reserved, at least one of $\{B^{(j)}, B^{(k)}\}$ reserved, and $B^{(p)}$ reserved.

Extract $\{W_1^{(t)}\ W_2^{(t)}\ W_3^{(t)}\}$ from the reserved blocks to find 83 bits for reconstructing the block $B^{(t)}$. According to the table in Figure 2, calculate the corresponding coefficient value using equation (9). After using inverse quantization and DCT operations, content of the tampered block $B^{(t)}$ is restored.

Case 2: Condition: at least one of $\{B^{(e)}, B^{(f)}, B^{(g)}\}$ is reserved, and at least one of $\{B^{(j)}, B^{(k)}\}$ reserved; while $B^{(p)}$ tampered.

Extract $\{W_1^{(t)}\ W_2^{(t)}\}$ from the reserved blocks to find 50 bits for reconstructing the block $B^{(t)}$. According to the table in Figure 3, calculate the corresponding coefficient value using equation (9). After using inverse quantization and DCT operations, content of the tampered block $B^{(t)}$ is approximately restored.

Case 3: Condition: at least one of $\{B^{(e)}, B^{(f)}, B^{(g)}\}$ is reserved; while $\{B^{(j)}, B^{(k)}, B^{(p)}\}$ tampered.

Extract $\{W_1^{(t)}\}$ from the reserved blocks to find 19 bits for reconstructing the block $B^{(t)}$. According to the table in Figure 4, calculate the corresponding coefficient value using equation (9). After using inverse quantization and DCT operations, content of the tampered block $B^{(t)}$ is approximately restored.

Otherwise: Content of the tampered block $B^{(t)}$ cannot be recovered.

$$C = \sum_{k=1}^{K} (b_k \cdot 2^{K-k}) + R_{\min} \quad (9)$$

In the scheme, if tampered rate is smaller than 1/2, image restoration belongs to the first case as the reserved part provides more reference data. When tampered rate is larger than 1/2 but smaller than 2/3, restoration for most blocks belongs to the second case. Moreover, case 3 is suitable for the situation when tampered rate within the range of 2/3 and 3/4. In fact, quality of restoration will decrease when tampering rate increases.

3. Experimental Results

We have implemented many experiments using the proposed method. Figure 5 provides a group of test images, where (a) is the original image "Portofino" sized 512×512, (b) the self-embedded image with PSNR equals 37.9 dB, (c) the tampered image with 13.7% areas tampered, (d) the identification map in which white parts represent the reserved blocks and black the tamped blocks.

A group of experimental results with different tampering rate are shown in Figure 6 ~ Figure 8. The original image is "Lake" of size 512×512. After self embedding, we obtain the watermarked image. In Figure 6, 22% of the watermarked image is identified to be tampered, in which most blocks belong to the first case of restoration; PSNR of the recovered image equals 38.5dB. Figure 7 shows the second case of restoration, where 62% of the watermarked image is tampered and PSNR of recovered image equals 31 dB. More blocks are tampered in Figure 8; with data extracted from the reserved blocks, 72% of areas are recovered with PSNR 25.6 dB. As described previously, quality of restoration is related to tampering rate. Better quality appears when the tampering is smaller.

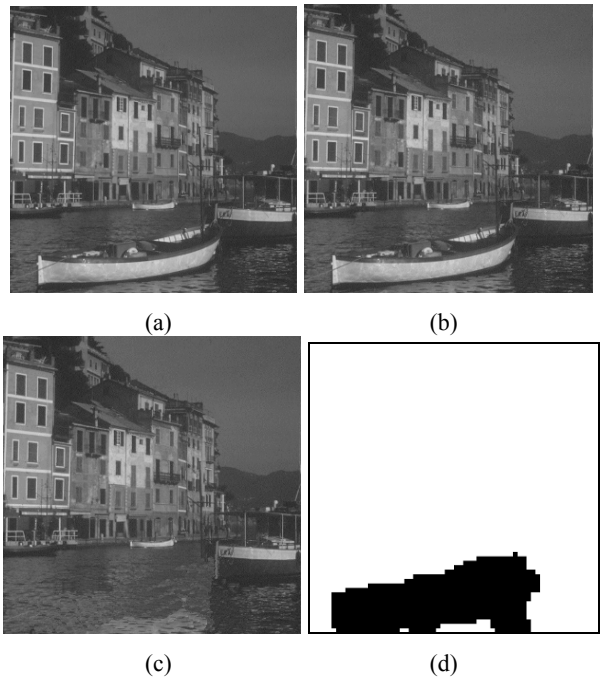
Figure 9 compares the proposed method with Zhang's method in [15]. The test image 'Lena' is used in both methods for comparison. PSNR values due to different

tampering rates show the proposed method provides better restoration quality.

Comparisons of restoration capabilities are shown in Table 1, in which tampering area up to 75% of total image can be recovered by using the proposed method, while Fridrich method in [1] tolerates only 50% of tampering, and Zhang's method in [8] 54% at most.

4. Conclusions

This paper proposes a self-embedding method with flexible restoration capability. We present a table for encoding DCT coefficients into reference bits. The coefficients are classified into three types, in which three copies of low frequency coefficients of a block are embedded into three remote blocks, two copies of middle frequency coefficients are embedded into two remote blocks, and one copy of middle frequency coefficients are embedded into another remote block. On the receiver side, one can recover the tampered blocks with extracted bits.
The proposed method is not complex which can be implemented easily. Distortion of an image after self-embedding is imperceptible to human vision. Three cases are analyzed for self recovery, and the quality of recovered contents is flexible, as we use different amount reference bits to recover the content corresponding to each tampering cases. As a result, the restoration quality gets better when the tampering rate turns smaller. Finally, we may recover as much as 75% of the whole image.



(a) (b)

(c) (d)

Figure 5 self embedding and tampering: (a) is the original image, (b) self embedded image, (c) tampered image, (d) map of identification

(a) tampered image (b) recovered image

Figure 6 Tampering and restoration in the first case, 22% tampered

(a) tampered image (b) recovered image

Figure 7 Tampering and restoration in the second case, 62% tampered

(a) Tampered Image (b) Recovered Image

Figure 8 Tampering and restoration in the third case, 72% tampered

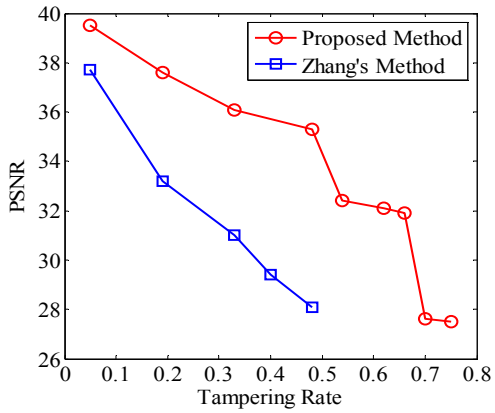


Figure 9 Tampering rate and recovery quality

Table 1 Comparison of Restoration Capabilities

Methods	Restoration Capability (Upper Bounds)
Method in [8]	50%
Method in [15]	54%
Proposed Method	75%

Acknowledgements

This work was supported by the Natural Science Foundation of China (Grant 61103181), the Natural Science Foundation of Shanghai (Grant 11ZR1413200), and the Innovation Program of Shanghai Municipal Education Commission (Grant 11YZ10).

REFRENCES:

[1]C. Vleeschouwer, J. F. Delaigle, and B. Macq, Invisibility and Application Functionalities in Perceptual Watermarking — an overview, *Proc. IEEE*, 90(1) (2002), 64–77.

[2]F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, Information Hiding—A Survey, *Proc. IEEE*, 87(7) (1999), 1062–1078.

[3]J. Fridrich and M. Goljan, Protection of Digital Images Using Self Embedding, in *Proc. Symp. on Content Security and Data Hiding in Digital Media*, (Newark, NJ, 1999).

[4]Z. Qian, G. Feng, X. Zhang, S. Wang, Image Self-Embedding with High-Quality Restoration Capability. *Digital Signal Processing*, 21(2) (2011), 278-286.

[5]S. Chen and H. Leung, A Temporal Approach for Improving Intra-frame Concealment Performance in H.264/AVC, *IEEE Trans. Circuits Syst. Video Technol.*, 19(3) (2009), 422–426.

[6]M. Yang and N. Bourbakis, An Efficient Packet Loss Recovery Methodology for Video Streaming Over IP Networks, *IEEE Trans. Broad.*, 55(2) (2009), 1, 190–210.

[7]C.-Y. Lin and S.-F. Chang, SARI: Self-Authentication-And-Recovery Image Watermarking System, *in Proc. 9th ACM Int. Conf. Multimedia*, (2001), 628–629.

[8]J. Fridrich and M. Goljan, Images with Self-correcting Capabilities, *in Proc. IEEE Int. Conf. Image Processing*, (1999), 792–796.

[9]I. Kostopoulos, S. A. M. Gilani and A. N. Skodras, Color Image Authentication Based on a Self-embedding Technique, in *Proc. 14th International Conference on Digital Signal Processing* (2002), 2, pp. 733–736.

[10]P. L. Lin, C. K. Hsieh and P. W. Huang, A Hierarchical Digital Watermarking Method for Image Tamper Detection and Recovery, *Pattern Recognition*, 38(12) (2005), 2519-2529.

[11]H. J. He, J. S. Zhang and F. Chen, Adjacent-block Based Statistical Detection Method for Self-embedding Watermarking Techniques, *Signal Processing*, 89(8) (2009), 1557-1566.

[12]Z. Qian and G. Feng, Inpainting Assisted Self Recovery with Decreased Embedding Data, *IEEE Signal Processing Letters*, 17(11) (2010), 929-932.

[13]X. Zhang and S. Wang, Fragile Watermarking with Error-Free Restoration Capability, *IEEE Transactions on Multimedia*, 10(8) (2008), 1490-1499.

[14]W. Xue, G. Zeng and M. Zhi, Self-Embedding Watermark for Image Restoration Based on Repeat Correcting Code, *Advanced Materials Research*, 219-220(2011), 66-70.

[15]X. Zhang, S. Wang, Z. Qian and G. Feng, Self-Embedding Watermark with Flexible Restoration Quality, *Multimedia Tools and Applications*, 54(2) (2011), 385-395.

[16]Z. Qian, X. Zhang, G. Feng, and Y. Ren, Color Filter Array Interpolation Based Self-recovery with Anti-cropping Capability. *International Journal of Multimedia Intelligence and Security*, 1(2) (2010), 191–203.

[17]C. Qin, C. C. Chang and P. Y. Chen, Self-embedding Fragile Watermarking with Restoration Capability Based on Adaptive Bit Allocation Mechanism, *Signal Processing*, 92 (2012), 1137-1150.

Zhenxing Qian received the B.S. degree in 2003 and the Ph.D. degree in 2007 from University of Science & Technology of China (USTC). Since 2009, he has been with the faculty of the School of Communication and Information Engineering, Shanghai University. He is now an associate professor. His research interests include data hiding, image processing, and digital forensics.

Lili Zhao received the B.S. degree from Shandong Polytechnic University in 2011. She is now a graduate student pursuing the master degree in the School of Communication and Information Engineering, Shanghai University. Her research interests include image processing and data hiding.