Self Embedding Watermarking Scheme Using Halftone Image

Hao LUO①, Student Member, Zhe-Ming LU②, Shu-Chuan CHU③④, Nonmembers, and Jeng-Shyang PAN⑤⑥⑦, Member

SUMMARY  Self embedding watermarking is a technique used for tamper detection, localization and recovery. This letter proposes a novel self embedding scheme, in which the halftone version of the host image is exploited as a watermark, instead of a JPEG-compressed version used in most existing methods. Our scheme employs a pixel-wise permuted and embedded mechanism and thus overcomes some common drawbacks of the previous methods. Experimental results demonstrate our technique is effective and practical.

key words: self embedding watermarking, digital halftoning, tamper detection, image recovery

1. Introduction

Self embedding watermarking is a technique that hides an image’s content into itself. It is often utilized for tamper detection, localization and repairing the tampered areas. Usually, these methods consist of (1) watermark generation, (2) watermark embedding and extraction, (3) tamper detection, localization and image recovery. In most existing self embedding watermarking methods, its JPEG-compressed version is adopted as the watermark. Digital halftoning is to transforming a continuous-tone image into a two-tone image [1], e.g. from an 8-bit gray level image to a binary image. When viewed from a distance, halftone images can resemble their original versions by the low-pass filtering of the human eyes. As an opposite process, inverse halftoning is to reconstruct a continuous-tone image from a halftone image such that the continuous-tone appears visually similar to the halftone.

This letter presents a novel self embedding technique. The key characteristics are as follows. (1) A halftone version of the host image is used as the watermark, instead of a JPEG-compressed image popular used. (2) Our technique employs a pixel-wise permuted and embedded mechanism so that some common limitations of the previous methods can be improved. (3) In tamper detection, block size can be arbitrarily adjusted. Hence small alterations can be more accurately detected and localized.

2. Previous Work

Fridrich et al. [2] introduced a self embedding scheme based on JPEG compression. First, the host image is partitioned into 8 × 8 blocks and each block is DCT transformed. Second, 11 lowest frequency coefficients are extracted and quantized using a JPEG quantization matrix. Next, the quantized coefficients of a block B are zig-zag ordered and encoded into a 64-bit binary sequence. The 64-bit binary sequence is hidden in the LSB (Least Significant Bit) of another distant block B', which selected by B + p̂, where p̂ is a vector with a randomly chosen direction. Given a suspect image, the LSB of B' are retrieved and used to authenticate content of B. This principle is illustrated in Fig. 1.

However, this scheme has some limitations. For example, if a large area of the image is tampered, thus the block B' is also likely to be changed. In this case content of B is lost and can not be recovered any longer. Another security flaw is occurred when an attacker modifies the 7 most significant bits of the block B while keeping its LSB intact. In this case the extracted watermark in B' is the same as the LSB of B, and this will lead to a wrong conclusion that content of B is intact. Besides Fridrich et al.’s scheme, some other self embedding methods [3], [4] are also based on quantization-based lossy compression. Therefore they also have insufficiencies similar as Fridrich et al.’s scheme. In next section, we aim to improve these insufficiencies from a new viewpoint.

Fig. 1  Principle of Fridrich et al.’s self embedding scheme.

① The author is with the Department of Automatic Test and Control, Harbin Institute of Technology, Harbin, China.
②③ The author is with Sun Yat-sen University, Guangzhou, China.
④⑤⑥⑦ The author is with the Department of Information Management, Cheng Shiu University, Kaohsiung County, Taiwan.
a) E-mail: luohao723@126.com
DOI: 10.1093/ietisy/e91-d.1.148

Copyright © 2008 The Institute of Electronics, Information and Communication Engineers
3. Proposed Scheme

Figure 2 is the block diagram of our method. As shown in Fig. 2(a), the watermark is obtained by halftoning the host image. Then it is permuted using a key $K$ and inserted in the LSB of the host image. In Fig. 2(b), the watermark is retrieved from the LSB of a suspect image and inverse permuted using $K$. Thus the embedded halftone image can be reconstructed and further inverse halftoned to a continuous-tone image $C_1$. Meanwhile, the suspect image is also preprocessed into $C_2$. Finally, $C_1$ is compared with $C_2$ for tamper detection and recovery.

3.1 Watermark Generation

As aforementioned, the watermark generation is a halftoning process of converting the host image into a halftone image. Popular halftoning methods can be divided into three categories: ordered dithering, error diffusion and iterative methods. Among these, error diffusion achieves a preferable tradeoff between good visual quality and reasonable computational complexity. In our case, considering halftoning and inverse halftoning processing smooth the host image to some extent, we adopt edge enhancement error diffusion halftoning [5]. Edge enhancement error diffusion is one of the variations of the conventional error diffusion, in which edge information of the input image can be well preserved.

The block diagram of the edge enhancement error diffusion is shown in Fig. 3. $x_{(i,j)}$ is the current processing pixel and $x'_{(i,j)}$ is the diffused error sum added up from the neighboring processed pixels. $b_{(i,j)}$ represents the binary output at position $(i,j)$. $u_{(i,j)}$ is the modified gray output and $e_{(i,j)}$ is the difference between the $u'_{(i,j)}$ and the $b_{(i,j)}$. The relationships of these variables are as follows:

- $u_{(i,j)} = x_{(i,j)} + x'_{(i,j)}$  \hspace{1cm} (1)
- $u'_{(i,j)} = u_{(i,j)} + LX_{(i,j)}$  \hspace{1cm} (2)
- $x'_{(i,j)} = \sum_{m=0}^{1} \sum_{n=-1}^{1} e_{(i+m,j+n)} \times k_{(m,n)}$  \hspace{1cm} (3)
- $e_{(i,j)} = u_{(i,j)} - b_{(i,j)}$  \hspace{1cm} (4)
- $b_{(i,j)} = \begin{cases} 0 & \text{if } u'_{(i,j)} < t \\ 1 & \text{if } u'_{(i,j)} \geq t \end{cases}$ \hspace{1cm} (5)

where $k_{(m,n)}$ in the Eq. (3) is the error diffusion kernel and the parameter $t$ in the Eq. (5) is a threshold value usually set as 0.5. In our context, the widely used Floyd-Steinberg kernel is adopted, as shown in Fig. 4.

In Eq. (2), $L$ is used to scale the current image pixel and the result is added to the thresholding input. As $L$ increases, the sharpness of the resulting halftone increases. In other words, smaller values of $L$ would cause blurring, and larger values would cause sharpening with respect to the original continuous-tone image. In order to obtain a good visual quality of the halftone image, Kite et al. [6] found a globally optimal value of $L$, i.e. 0.75, when Floyd-Steinberg kernel is employed in edge enhancement error diffusion.

The reasons and advantages of why the halftone version of the host image can be utilized as a watermark may lie in the following factors. (1) Important features (e.g. edges, textures) of the host image can be well preserved in a halftone. (2) Size of the watermark data is suitable for embedding invisibly. As an 8-bit gray level image is transformed to a 1-bit image, the host image can be regarded as “lossy compressed”. (3) The halftone image generation is simple and fast, for halftoning is originally applied in real-time printing. (4) The watermark can be reconstructed into a continuous-tone image with high quality. This makes tamper detection and recovery possible. For example, to 512×512 gray level Lena, the PSNR between the original image and the reconstructed continuous-tone image is 29.65 dB. The visual quality of the reconstructed image is moderate. If we select other high quality halftoning and inverse halftoning techniques, this PSNR value is likely to

![Fig. 2 Block diagram of the proposed scheme, (a) watermark generation and embedding, (b) watermark extraction, tamper detection, localization and image recovery.](image)

![Fig. 3 Block diagram of edge enhancement error diffusion halftoning.](image)

![Fig. 4 Floyd-Steinberg error diffusion kernel (X is the current pixel).](image)
increase to 31.3 dB using the method in [7]. But this improvement is achieved at the expense of higher computational complexity.

3.2 Watermark Embedding and Extraction

The watermark embedding is to permute pixels of the watermark $W$ according to a pseudo-random number generator with a key $K$. Thus $W_p$ is obtained, and used to replace LSB of the host image $I$.

Given a suspect image $I'$, the watermark extraction is to retrieve the LSB of $I'$ directly and inverse permute it using $K$. Thus the watermark image $W'$ is obtained.

3.3 Tamper Detection, Localization and Recovery

The extracted watermark $W'$ must be inverse halftoned to a continuous-tone image. Here we employ a simple lowpass filtering approach for inverse halftoning. A $5 \times 5$ Gaussian filter which simulates human visual system (HVS) characteristics is used, as shown in Eq. (6), to filter the watermark images $W'$ by convolution as Eq. (7).

$$F = \frac{1}{11.566} \begin{bmatrix} 0.1628 & 0.3215 & 0.4035 & 0.3215 & 0.1628 \\ 0.3215 & 0.6352 & 0.7970 & 0.6352 & 0.3215 \\ 0.4035 & 0.7970 & 1 & 0.7970 & 0.4035 \\ 0.3215 & 0.6352 & 0.7970 & 0.6352 & 0.3215 \\ 0.1628 & 0.3215 & 0.4035 & 0.3215 & 0.1628 \end{bmatrix}$$

$$\hat{W} = W' \otimes F$$

where $\otimes$ denotes the convolution operation. We utilize the reconstructed $\hat{W}$ for tamper detection, localization, and recovery. Steps are described as below.

Step 1. Preprocessing the suspect image $I'$, that is, halftoning $I'$ into a halftone image using edge enhancement error diffusion, and filtering this halftone again using the filter in Eq. (6) to obtain a continuous-tone image $I$.

Step 2. Partition $I$ and $\hat{W}$ into non-overlapping $n \times n$ blocks $B_{Il}$ and $BW_{Il}$ ($l = 1, 2, \ldots, m$) respectively.

Step 3. Compute the difference $D$ between block $B_{Il}$ and the corresponding block $BW_{Il}$ as

$$D = \frac{1}{n^2} \sum_{u=1}^{n} \sum_{v=1}^{n} (B_{Il}(u,v) - BW_{Il}(u,v))^2$$

where $(u,v)$ is the pixels coordinates.

Step 4. Quantize the difference $D$ into 0 or 1 according to a threshold value $T$ set in advance as

$$A = \begin{cases} 1 & D < T \\ 0 & D \geq T \end{cases}$$

where $A$ is the final decision of the block is authentic or unauthentic. Here we use 1 denotes authentic and 0 unauthentic. In other words, if the difference is smaller than $T$, we output the block is authentic, otherwise unauthentic. In this way, all blocks in $I'$ can be tagged as either authentic or unauthentic.

Step 5. Replace the unauthentic blocks of $I'$ using the blocks of $\hat{W}$ which localize at the same positions.

Actually, Step 1 is quite important and essential. Without it, if we directly compare $I'$ instead of $I$ with $W$, the difference $D$ is inefficient to discriminate the tampered blocks. The tampered areas are detected and localized by Step 3 and Step 4, and approximately recovered by Step 5.

4. Experimental Results and Discussions

Our test images are of the size $512 \times 512$ pixels. The halftoning adopted is edge enhancement error diffusion (with Floyd-Steinberg kernel, $L = 0.75$) and the inverse halftoning adopted is filtering using the Gaussian filter in Eq. (6). The first experiment is on cropping with results shown in Fig. 5, where a $40 \times 40$ square area in the watermarked image is cropped. The second experiment is on concealing an object using its neighbor background, with experimental results shown in Fig. 6. Note each pixel block is of the size $8 \times 8$ in the two experiments. Pixels values are normalized into the range of $[0, 1]$, and the threshold $T = 0.015$. The block difference $D$ magnitudes of these experiments are shown in Fig. 7, where the $x$-axis represents the block index scanning in the raster manner, and the $y$-axis is the difference values $D$. It is clear that in the tampered area, the $D$ values are significant larger than those in the unchanged area. Therefore the quantization operation can discriminate the unauthentic blocks.

Actually, the parameter $T$ is closely related with the block size. From Eq. (8), to a larger block, a larger difference $D$ is accumulated. In addition, as $T$ increases, the number of tampered blocks detected increases. That is, more blocks are marked as tampered blocks and then further be recovered.
Compared with available JPEG-compression based methods, our scheme improves the previous work in the following aspects. (1) Since content of a block is dispersed in the whole image instead of another distant block, it is impossible to destroy all its content even when a large area of the watermarked image is tampered. Therefore the tampered area also can be recovered to some extent. This high self recovery ability outperforms the previous block-wise-based methods. (2) The tampered positions can be more accurately localized by partitioning images into smaller blocks, e.g. 4 × 4 blocks, while unit of detected tampered area in JPEG compression-based methods is fixed as 8 × 8 blocks. This property is suitable for detecting and accurately localizing a small tampered area, meanwhile its neighbor content need not recovered. As an example shown in Fig. 8, a small tampered area is accurately localized and its original content is recovered. (3) In Fridrich et al.’s scheme, a wrong conclusion is made when an attacker modifies the 7 most significant bits of the block $B$ while keeping its LSB intact, for the extracted watermark in $B'$ is the same as the LSB of $B$. However, in our method, not only the LSB of $B$ but also its other bits plane information is all taken into account for content authentication. Hence this insufficiency is also improved.

Another experiment is performed to compare the properties of Fridrich et al.’s scheme and our method. Experimental results are shown in Fig. 9. The selected host image is the Lena shown in Fig. 5 (a). Figure 9 (a) and (d) are the watermarked images obtained by Fridrich et al.’s scheme and our method, respectively. In this experiment, we consider a special scenario. In Fig. 9 (a), only two blocks 8 × 8 $b_1$ and $b_2$ ($b_1$ lies in the upper-left while $b_2$ locates in the lower-right of the image) are changed to white blocks, as shown in Fig. 9 (b). Specifically, in watermark embedding the compressed version of $b_1$ is hidden in the LSB of $b_2$. Thus, as both $b_1$ and $b_2$ are changed, the original content of $b_1$ can not be recovered using the LSB of $b_2$, as shown in Fig. 9 (c),

![Fig. 6 Experimental results of replacing an object using its neighbor background, (a) the host image, (b) the watermark image, (c) the watermarked image, (d) the tampered image, (e) the localized tampered area, (f) the recovered image.](image1)

![Fig. 7 Blocks difference values in experimental (a) cropping, (b) replacing an object using its neighbor background.](image2)

![Fig. 8 Experimental results of 4 × 4 blocking, (a) the host image, (b) the watermark image, (c) the watermarked image, (d) the tampered image, (e) the localized tampered area, (f) the recovered image.](image3)

![Fig. 9 Comparison of Fridrich et al.'s scheme and our method, (a) the watermarked image obtained by Fridrich et al.'s scheme, (b) the tampered image of (a), (c) the recovered image of (b), (d) the watermarked image obtained by our method, (e) the tampered image of (d), (f) the recovered image of (e).](image4)
although the content of $b_2$ can be recovered using LSB of some other block. In our method, the positions of two tampered blocks as shown in Fig. 9(e) are exactly the same as those in Fig. 9(b). However, as the compressed version of $b_1$ is dispersed in all over the watermarked image, its original content still can be recovered by collecting the dispersed information, as shown in Fig. 9(f). From these experimental results, we can find that the security flaw of Fridrich et al.'s scheme is improved in our scheme.

Nevertheless, in our scheme, as the watermark hidden in the LSB, it is easily to be attacked by intentional alterations or common image operations, e.g. JPEG compression. When the LSB plane are replaced or discarded, our scheme fails. As most image transformations such as noise adding change the LSB values of the watermarked image, the hidden watermark is easily destroyed. Therefore our scheme is not robust enough to survive in these image transformations. To improve the watermark robustness, we can hide it in the second or higher LSB plane, although larger distortion introduced when replacing the higher LSB. From numerous experiments, we find that visual distortions resulted from up to the fourth LSB replacement can be accepted.

Actually, our scheme can be easily extended to color image self embedding. For example, to a 24-bit color image, first we decompose it into red, green, and blue channels. Second, each channel is regarded as a 8-bit image and transformed into a halftone version using edge enhancement error diffusion. Third, the halftone image is hidden in each channel's LSB bit-plane and thus the watermarked channel is obtained. At last, the three watermarked channels are composed into the final self embedded image.

5. Conclusions

A novel self embedding image watermarking technique is proposed in this letter. Its main characteristic is to employ a halftone version of the host image as a watermark. As the watermark is pixel-wise permuted and embedded, some limitations of the previous methods are improved. It is efficient to not only accurately detect and localize the tampered area, but also approximately recover its original content automatically.

References