Watermarking with Quality Control for Binary Images[1]

Chun-E Zhang, Zheng-Ding Qiu

Institute of Information Science, Beijing Jiaotong University, Beijing, China

Abstract

A fragile watermarking scheme with quality control for binary images is proposed in this paper. In this paper, a secret key and a weight matrix are used to protect the hidden data. For an $m \times n$ block, our scheme can embed as high as $\lfloor \log_2(mn+1) \rfloor$ bits data at cost of two pixels flipped. Pixel flippability guides the whole scheme to control the watermarked image quality. Shuffling is also adopted to improve the watermarking capacity and at the same time make the scheme more secure.

Keywords: watermarking, weight matrix, shuffling, pixel flippability

1. Introduction

As we know as so far, most watermarking schemes for digital images mainly give their efforts to images whose pixels can be represented by more than one bit such as 8 bit depth images. For such images plentiful textures and colors can be utilized to hide data or so-called watermarks.

Actually very little research had been done on watermarking for binary images while there are increasing applications concerned to binary images. Handwriting captured by electronic signing pads are digitally stored and used as credence for credit card payment and business online. Documents such as contracts and financial reports may also be stored as binary images in terms of FAX, etc. It is necessary to develop watermarking schemes to protect such binary images. Such situations demand fragile or semi-fragile watermarking. For a binary image watermarking means flipping, that is a white pixel is toggled to a black pixel or 1 is flipped into 0, and vice versa, which are easily detected and so the pixel flippability should be considered. References [1~5] went ahead in this area. Except Lu [5] most of these are in spatial transform and block based. Comparatively Lu's scheme is a robust scheme but it needs original images when extracting hidden data. Wu [2] referred a predefined flippability lookup table and flipped some most flippable pixels to embed one bit in one block. Wu's scheme took watermarked image quality into account by shuffling while the capability is limited for only one bit hidden for one block. Tseng [3] proposed a watermarking scheme with high capacity. Given a host block of size $m \times n$ [3] can watermark as many as $\lfloor \log_2(mn+1) \rfloor$ bits of data by flipping at most two bits in the block. Although [3] improved capacity greatly as a scheme for binary images, it didn't take much image quality into consideration, which will be studied further in this paper.

Thanks to Tseng's idea of high watermarking capacity this paper will make some improvements. Firstly the embedding process takes image quality into consideration. And secondly we will further improve the capacity and security of the proposed scheme by shuffling and quality control. This paper is arranged into four parts. The first is introduction. In the second section Tseng's scheme will be reviewed and analyzed. The third section proposes our scheme and some improvements. Experimental results and analysis will also be presented. Finally is the conclusion.

2. Review and Analysis

2.1. Review

Given two bitmaps as matrixes B_1 and B_2 of the same size, $B_1 \wedge B_2$ means the bitwise AND and $B_1 \oplus B_2$ means the bitwise XOR of the two matrixes. $B_1 \otimes B_2$ means the pair-wise multiplication between the two matrixes. There are some symbols should be defined.

- F_i: a host block bitmap of size $m \times n$, which is to be modified to embed data. Without loss of generality, we assume that one host image is partitioned into LF_i, where L is the blocks number in the host image..
- K: a secret key shared by the sender and the receiver. It is a randomly arranged matrix with elements {0, 1} of size $m \times n$.
- r: the number of bits to be embedded into each F_i. The value of r satisfies $2^r - 1 \le mn$.
- W: a secret weight matrix shared by the sender and the receiver. It is an integer matrix of size

[1] Supported by Innovation Fund for Small Technology Based Firms, China under Grant No.04C26213301189 and the Key Laboratory of Advanced Information Science and Network Technology of Beijing under Grant No. TDXX0509.

of $m \times n$ which is randomly arranged with elements $w = \{1,2,\cdots,2^r - 1\}$. And each element will appear at least once in *W*. Thus the number of choices for *W* is:

$$c_{2^{r-1}}^{mn} * (2^r - 1)! * (2^r - 1)^{mn-(2^r-1)}$$

For a suitable *m*/*n*/*r* this number will be large enough to forbid a brute-force attack.

- *B*: information consisting of *Lr* bits to be embedded in F_i.

Tseng scheme embeds *r* bits of data, $b_1 b_2 \ldots b_r$ (from the most significant bit to the least significant bit), into one block F_i by changing at most 2 pixels to make F_i to F_i'. After modification the following formula must be satisfied:

Let $S_i = F_i \oplus K$

$$s_i = SUM((F_i' \oplus K) \otimes W)$$

Then $$s_i (\mathrm{mod}\ 2^r) = (b_1 b_2 \ldots b_r)_2 \quad (1)$$

From the matrix S_i compute the following set for each $w = \{1,2,\cdots,2^{r-1}\}$:

$$P_w = \{(j,k) \mid ([W]_{j,k} = w \wedge [S_i]_{j,k} = 0) \vee ([W]_{j,k} = 2^r - w \wedge [S_i]_{j,k} = 1\} \quad (2)$$

The set P_w contains every matrix index pair (*j*, *k*) such that if we randomly complement one $[F_i]_{j,k}$, s_i will be increased by *w* and so **watermark is embedded**. More details are developed in [3]. Here undefined P_0 could be regarded as non-empty set although nothing is changed to achieve an increase of weight by 0, the same with multiples of 2^r.

2.2. Analysis of the Scheme

High capacity is the most overwhelming advantage of the scheme above. But through the embedding scheme the image quality after data embedding is not controlled. The pixels to be flipped are selected some randomly and the effect of the flipping to the image quality is not considered so that sometimes the modification to a region block is obtrusive.

In the next section we will discuss some measures to control the image and some improvements will be proposed.

3. Watermarking Scheme with Image Quality Control

With at most two pixels are flipped per block the image quality after data hiding should be considered. What we need to do is to make the flipping not so obtrusive or not so obvious.

There are two kinds of flipping which will influence the image quality after data hiding. The first, flipping occurring at a uniform black or white region influences the image quality obviously. The second, flipping which changes the object connectivity influences the image quality also, even influences optical character recognition (OCR) result. The first kind is a subset of the second kind. Before data hiding we should consider the flippability of every pixel. In our scheme pixels locating on edges of image objects and those locating where black and white pixels both distribute cross-uniformly are highly regarded as flippable pixels. But how we measure the flippability of one pixel? In this section we firstly introduce a measure for pixel flippability and secondly present a complete watermarking scheme.

3.1. Pixel Flippability

Definition 1: Distance Transform provides a measure for the separation of points in the image *I*. DT calculates the Euclidean distance between each pixel and the nearest pixel complement to it for binary images.

$$[DT(I)]_{i,j} = \min\{\sqrt{|i-x|^2 + |j-y|^2} \mid [I]_{i,j} \neq [I]_{x,y}\}$$

Definition 2: Euler number is a measure of the topology for an image. It is defined as the total number of objects in the image minus the number of holes in those objects. 4- or 8-connected neighborhoods are optional. Connectivity for an image is its topology, which remains invariant after translation, rotation, scaling and twist. Euler number is used to measure the connective topology for a binary image. In this paper 8-connected neighborhood is chosen.

We assign a Flippability Score, FS(*i*, *j*), to one pixel *I*(*i*, *j*) according to DT(*I*) down through steps below. The whole scheme is block based.

First all pixels are regarded as highly flippable and let its flippability score be DT(*I*(*i*, *j*)) by default. Then let F_i be the host block, extend the four boundaries of F_i repeatedly to get a new $(m+2) \times (n+2)$ block *M*. If flipping one pixel $[F_i]_{i,j}$ within the host block F_i results change of Euler number for *M*, set FS(*i*, *j*) to infinite, represented by Inf.

The filppability scores make the image quality under control after data hiding. Now every pixel has a flippability score. Score Inf. means flipping the corresponding pixel will cause serious image quality regression while one smaller score means the corresponding pixel is more flippable.

3.2. Uneven Flippability Distribution and Shuffling

In our scheme we can embed *r* bits data per block in theory. However the distribution of flippable pixels may vary dynamically from block to block. For some uniform blocks even one flipping could be obtrusive and can't be utilized to embed any data while blocks with plentiful details are with more flippable pixels and can endure more modifications. This uneven flippability distribution results to uneven embedding capacity for each block, which actually reduces total capacity greatly. In this paper we use shuffling to equalize the uneven flippability distribution, i.e., embedding capacity from region to region. Shuffling is controlled by a secret key and randomly distributes flippable pixels around the whole image *I*, which makes every block be possible to embed data and also enhances security since shuffling further reduce the chance weight matrix *W* be guessed out.

3.3. Watermarking Process

The following is our proposed embedding scheme.

E1. Calculate DT(*I*) of the original image and assign flippability score for every pixel.

E2. According to a secret key KEY shuffle forwards the original image to get image *F* after shuffling. Also shuffle forwards DT(*I*) to DT′ (I);

E3. Generate *K* and *W* according to KEY. Divide *F* into LF_i of size $m \times n$. Divide bitstream *B* to be multiples of *Lr* bits B_i.

E4. From the matrix S_i, compute for each $w=\{1, 2, \ldots, 2^r-1\}$ the set P_w, which contains every matrix index pair (j, k) such that if we complement $[F_i]_{j,k}$, s_i will be increased by *w*.

E5. Define a weight difference:

$$d' = (b_1 b_2 \ldots b_r)_2 - s_i \pmod{2^r} \quad (3)$$

If $d' = 0$, there is no need to change F_i else the following operation must be executed to transform F_i to F_i'. We assume that $P_w = P_{w'}$ if any $w \equiv w' \pmod{2^r}$.

a) Store all $h \in \{0,1,\cdots 2^r - 1\}$ such that $P_{hd} \neq \emptyset$ and $P_{-(h-1)d} \neq \emptyset$.

b) Store pairs of coordinates $(j, k) \in P_{hd}$ and $(p, q) \in P_{-(h-1)d}$ into SS_d, record the sum of their filppability scores FS;

c) Select one pair of coordinates in SS_d, the pixels on which have smallest sum of FS[*i*,*j*], complement the pixels;

d) Jump to the next block.

E6. Shuffle back F_i' to get watermarked binary image *I*′.

The watermark extraction process is a simple reverse process of watermark embedding.

4. Experimental Results and Analysis

In our experiments a secret KEY controls the random matrix *K*, random permutation of weight matrix *W* and the image shuffling. Tseng's scheme is implemented to compare with our new scheme in four aspects and also the experiments are on one Chinese document of size 270×270, one English document of size 270×270 and one cartoon image of size 200×200. Fig.1~Fig.3 show the visual results of our experiments.

From the figures, the two algorithms are in common that capacity of watermark is decided by *r*. The bigger *r* is the larger the capacity is and the smaller *r* is the better the image quality is. Fig.1~Fig.3 show the tendency. The comparisons are:

1). Block size and watermark capacity: In theory when equal block size is applied the algorithms are of the same watermarking capacity, no more than *Lr* bits. Experiments show that our scheme could reach the upper bound no matter what kind host image is. This is because Tseng must abandon uniform blocks while by shuffling we can embed data in every block. For rare uniform blocks Fig.1 and Fig.2 aren't outstanding in this aspect. In Fig.3 under equal size of blocks our scheme perform much better over Tseng's due to many uniform blocks in Mickey.

2). Image quality: It's difficult to impose equal quality on images after watermarking because the different locations of modified bits affect the image quality quite differently. Theoretically and experimentally say, under almost equal amount of embedded data, in our scheme the noise on the edges of objects and the larger size needed result the better image quality than in Tseng's. The advantage of our scheme in Fig.1~Fig.2 and Fig.3.h~Fig.3.i is overwhelming.

3). Level of security: The security of Tseng's scheme depends on the block size used. The larger the block size is, the larger the weight matrix choices will be, which increase the probability to avoid a brute-force attack. So under equal amount of embedded data in our scheme the larger size of blocks provides higher security. Different secret keys also can be used to generate *K* and *W* to further enhance the security. Another time the advantage of shuffling stands out. Shuffling further reduces the chance weight matrix be guessed out and increases the robustness to collusion attacks.

4). Host image: For different kinds of host images both scheme performs differently. Our scheme has equal embedding capacity per block to Tseng's while it can only embeds data in non-uniform blocks. So the capacity difference

between the two schemes increases with uniform blocks distribution of host images. The advantage of our scheme stands out when image Mickey is watermarked as Fig.3 shows. So in order to embed nearly the same amount of data into host images like Mickey, our scheme can apply larger block size and gain good image quality.

5. Conclusion

In this paper we proposed a fragile watermarking under quality control for binary images. In theory, our scheme has high capacity of $M = \lfloor \log_2(mn+1) \rfloor$ bits per block. We always choose r as half of M. Besides this we also utilize flippability score of every pixel to control image quality after watermarking. In addition shuffling is introduced to improve watermark capacity especially for images with many uniform blocks. Shuffling also reduces the probability of brute-force attack and improves the security of our scheme. Our scheme achieves good performance in aspect of image quality, capacity and security at balance.

6. References

[1] Jeanne Chen, Tung-Shou Chen, Meng-Wen Cheng, "A new data hiding method in binary image," *Proc. of Multimedia Software Engineering*, Fifth International Symposium on, 10-12 Dec. 2003, pp: 88 – 93.

[2] Wu M., Liu B., "Data Hiding in Binary Image for Authentication and Annotation," *Multimedia, IEEE Transactions* on Volume: 6, Issue: 4, Aug. 2004, pp: 528 – 538.

[3] Yu-Chee Tseng, Yu-Yuan Chen, Hsiang-Kuang Pan, "A secure data hiding scheme for binary images," *Communications, IEEE Transactions* on Volume: 50, Issue: 8, Aug. 2002, pp: 1227 – 1231.

[4] Kim, H.Y., Afif, A., "Secure authentication watermarking for binary images," *Computer Graphics and Image Processing*, 2003. SIBGRAPI 2003. XVI Brazilian Symposium on, 12-15 Oct. 2003, pp: 199 – 206.

[5] Haiping Lu, Xuxia Shi, Shi, Y.Q., Kot, A.C., Lihui Chen, "Watermark embedding in DC components of DCT for binary images," *Multimedia Signal Processing*, 2002 *IEEE Workshop* on 9-11 Dec. 2002, pp: 300 – 303.

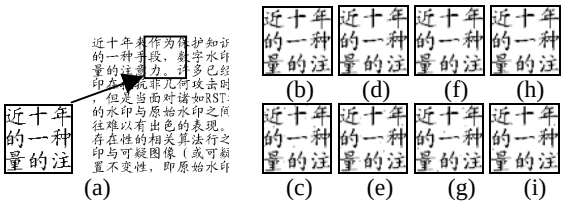

Fig.1: (a) Chinese document. (b)~(i) are watermarked images. (b) block size 12 × 12 by our scheme, r=3, length(B)=1452 bits. (c) block size 12 × 12 by Tseng's scheme, r=3, length(B)=1389 bits. (d) block size 24×24 by our scheme , r=4, length(B)=484 bits. (e) block size 24×24 by Tseng's scheme , r=4, length(B)=484 bits. (f) block size 24×24 by our scheme , r=7, length(B)=847 bits. (g) block size 24×24 by Tseng's scheme , r=7, length(B)=847 bits. (h) block size 36×36 by our scheme, r=7, length(B)=343 bits. (i) block size 36×36 by Tseng's scheme, r=7, length(B)=343 bits.

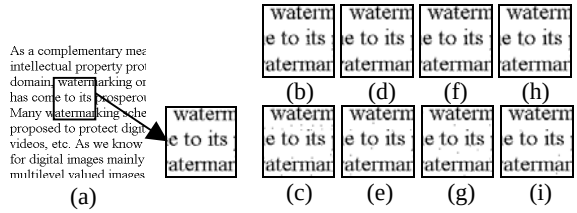

Fig.2: (a) English document: (b)~(i) are watermarked images. (b) block size 12×12 by our scheme, r=3, length(B)=1452 bits. (c) block size 12 × 12 by Tseng's scheme, r=3, length(B)=1158 bits. (d) block size 24×24 by our scheme, r=4, length(B)=484 bits. (e) block size 24×24 by Tseng's scheme, r=4, length(B)=484 bits. (f) block size 24×24 by our scheme, r=7, length(B)=847 bits. (g) block size 24×24 by Tseng's scheme, r=7, length(B)=847 bits. (h) block size 36× 36 by our scheme, r=7, length(B)=343 bits. (i) block size 36 ×36 by Tseng's scheme, r=7, length(B)=343 bits.

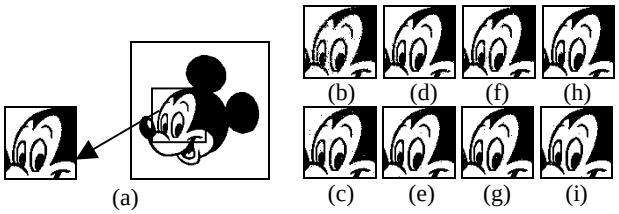

Fig.3. Embedding on Mickey (®Disney World): (a) Original Mickey image. (b)~(i) are watermarked images. (b) block size 12 × 12 by our scheme, r=3, length(B)=768 bits. (c) block size 12×12 by Tseng's scheme, r=3, length(B)=282 bits. (d) block size 24 × 24 by our scheme, r=4, length(B)=256 bits. (e) block size 24×24 by Tseng's scheme, r=4, length(B)=140 bits. (f) block size 24×24 by our scheme, r=7, length(B)=448 bits. (g) block size 24×24 by Tseng's scheme, r=7, length(B)=245 bits. (h) block size 36×36 by our scheme, r=7, length(B)=175 bits. (i) block size 36×36 by Tseng's scheme, r=7, length(B)=126 bits.