DATA HIDING FOR TEXT DOCUMENT IMAGE AUTHENTICATION BY CONNECTIVITY-PRESERVING

Huijuan Yang and Alex C. Kot

School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore 639798.
Email: ehjyang@ntu.edu.sg

ABSTRACT

In this paper, a novel blind data hiding method for text document images aims to preserve the connectivity in a local neighborhood is proposed. The "flippability" of a pixel is determined by imposing the three transition criterions in a 3×3 moving window which is centered at the pixel. The "embeddability" of a block is invariant in the watermark embedding process. While the "flipped" pixels can be located by imposing a constraint. The "uneven embeddability" of the host image is considered by embedding the watermark only in those "embeddable" blocks. The location is chosen in such a way that the visual quality of the watermarked image is guaranteed. Different types of blocks are employed and their abilities to increase the capacity are compared. A hard authenticator watermark is also generated to ensure the integrity and authenticity of the document.

1. INTRODUCTION

Authentication of digital documents has aroused great interest due to the wide application area nowadays, e.g., bank checks, legal documents, certificates, digital books and maps. Very often, digital documents are stored in binary image format. Since digital document is easy to copy and edit via the software tools, authentication and detection of tampering is of utmost concern.

In the past few years, a limited number of papers proposed new techniques for document watermarking and data hiding. Among these techniques, some result in noisy watermarked image due to the weak quality control, e.g., the key-weight matrix based method [1]. Some require a shuffling key in order to distribute the "flippable" pixels all over the image [2]. It may be difficult to find a proper shuffle key such that in each block of the shuffled image there is a suitable pixel to flip. Therefore, a larger block size, e.g., 12×12 is required.

In this paper, we propose a data hiding technique which is based on the connectivity-preserving in 3×3 neighborhood. The "uneven embeddability" of the host image is considered by embedding the watermark only in those "embeddable" blocks. A small block size, e.g., 4×4 is employed in order to achieve the larger capacity. The proposed scheme can be used for document authentication, e.g., eCertificate authentication.

2. PROPOSED METHOD

2.1. Flippability Decision

The flippability of a pixel depends on the transitions from the pixel to its eight neighbors in a 3×3 block. The 8 neighbors of the center pixel $p(i,j)$ are denoted as $N(p)$, and shown in Fig. 1.

$(i-1,j-1)$ w_6	$(i-1,j)$ w_7	$(i-1,j+1)$ w_8
$(i,j-1)$ w_5	(i,j) p	$(i,j+1)$ w_1
$(i+1,j-1)$ w_4	$(i+1,j)$ w_3	$(i+1,j+1)$ w_2

Fig. 1. Designations of pixels in 3×3 neighborhood.

Let's define "1" represents the black pixel and "0" represents the white pixel.

Definition 1. The number of uniform white and black transitions in a 3×3 block along the vertical and horizontal directions is named as "VH Transition", denoted as N_{VHW} and N_{VHB} and defined as

$$N_{VHW} = \sum_{i=1,3} \bar{p}.\bar{w}_i\bar{w}_{i+4} \;\; and \;\; N_{VHB} = \sum_{i=1,3} p.w_i.w_{i+4} \tag{1}$$

where, $\bar{w}$ implies logically "not w".

Definition 2. The number of transitions of the interior right angle in a 3×3 block is named as "IR Transition" and denoted as N_{IR}. It is defined as

$$N_{IR} = \sum_{i=1}^{4} \bar{p}.w_{2i}.\bar{w}_{2i-1}.\bar{w}_{2i+1} \tag{2}$$

where, $\bar{w}_{2i+1} = \bar{w}_1$, for $2i+1 > 8$.

Definition 3. The number of transitions from the center pixel to the sharp corners in a 3×3 block is named as "C Transition", denoted as N_C and defined as

$$N_C = \sum_{i=1}^{4} p.w_{2i}.w_{2i+1}.w_{2i+2}.w_{2i+3}.w_{2i+4} \tag{3}$$

where, $w_9 = w_1$, $w_{10} = w_2$, $w_{11} = w_3$ and $w_{12} = w_4$.

Definition 4. "Flippability Criterion", the center pixel in a 3×3 block is "flippable" if the number of VH transition, N_{VHW} and N_{VHB}, the number of interior right angle transition N_{IR} and the number of sharp

corner transition N_C remain the same before and after flipping the center pixel.

N_{VHW}, N_{VHB} and N_{IR} are calculated before and after flipping the center pixel. If the transition number doesn't change, it implies that flipping the pixel won't destroy the connectivity between pixels in the neighborhood and doesn't create extra clusters as well. These two conditions are collectively named as "Connectivity Preserving" criterion. While N_C is used to control not to flip pixels in sharp corners, as it is annoying to human observers. The qualified blocks which satisfy the "VH Transition", excluded by the "IR Transition" and "C Transition" are shown in Fig. 2 (a), (b) and (c) respectively. The pixels that meet the condition defined in (1) would have two white 4-neighbors, so, it is a boundary pixel. The condition defined in (2) is to ensure that flipping the center pixel doesn't create an isolated pixel (a pixel has eight white neighbors). Furthermore, by satisfying conditions defined in (1) and (2), at least one corner has three white pixels. This further ensures that flipping the center pixel won't destroy the local connectivity of the pattern.

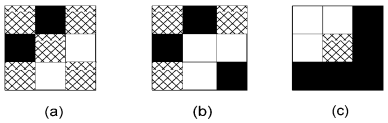

Fig. 2. An illustration of the three criterions used for the flippability decision excluding symmetric cases of rotation and complement. Pixel in grid represents "don't care" pixels.

2.2. Block Partition and Embeddability

Different types of blocks are employed. They are: fixed 3×3 block (FB), non-interlaced block (NIB) and interlaced block (IB), which are illustrated in Fig. 3. For the interlaced block, any two vertically or horizontally neighboring blocks share one common row or column.

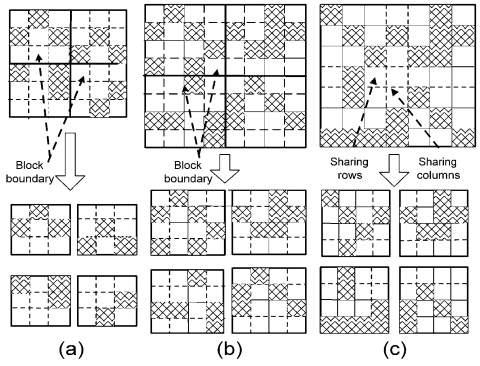

Fig. 3. An illustration of (a) fixed 3 ×3 block, the image (6×6) is partitioned into four 3×3 blocks. (b) non-interlaced block, the image (8×8) is partitioned into four non-interlaced 4×4 blocks and (c) interlaced block, the image (7×7) is partitioned into four interlaced 4×4 blocks.

The "embeddability" of a block depends on the "flippability" of determined pixels in the block, i.e., the center pixel of the block for the fixed 3×3 block, all pixels except the boundary pixels for the non-interlaced block and all pixels except those lie in the sharing rows and columns for the interlaced block. A moving window shown in Fig. 1 is employed to be centered at those determined pixels.

2.3. Capacities

Let's assume the probability that a pixel satisfies the three conditions is: p, then the probability of each block to be "embeddable" is $\frac{1}{9}p$ for a fixed 3×3 block; $\frac{(n-2)\times(n-2)}{n^2}p$ for a non-interlaced block with block size n × n; and $\frac{(n-2)\times(n-2)}{n^2}p$ for an interlaced block with block size n × n. The total block number is: $\lfloor W/3 \rfloor \times \lfloor H/3 \rfloor$ for a fixed 3×3 block; $\lfloor W/n \rfloor \times \lfloor H/n \rfloor$ for a non-interlaced block; and $\lfloor W/(n-1) \rfloor \times \lfloor H/(n-1) \rfloor$ for interlaced block, where, W, H are the width and height of the image, while $\lfloor x \rfloor$ is the floor function which gives the largest integer less than or equal to x.

It is obvious that the total block number has increased for the interlaced block compared with the non-interlaced block. More pixels can be flipped by using moving window to increase the probability of "embeddable" block for non-interlaced and interlaced block. A larger block size will definitely increase the probability that a block to be "embeddable". However, the total block number will be decreased.

2.4. Watermark Embedding and Extraction

The watermark embedding process is summarized as follows:

S1. Partition the image into equal size square blocks.

S2. Determine flippability of the determined pixels based on the "Flippability Criterion".

S3. Once a pixel is identified as "flippable", the block is marked as "embeddable".

S4. Proceed to the next block.

S5. Repeat steps S2 to S4 until all blocks are processed.

S6. Embed the watermark in the "embeddable" blocks by enforcing the odd-even feature of the number of black or white pixels in the block.

Lemma 1. The "embeddability" of a block is invariant in the watermark embedding process.

Proof. From the "Flippability Criterion", the "flippability" of a pixel is invariant in the embedding process. So, a "flippable" pixel is still "flippable" and an "embeddable" block remains "embeddable".

Let's divide the pixels in the kth "embeddable" block $\{P\}$ into two sets: determined pixels $D_k \in \{A\}$ and the non-determined pixels $U_k \in \{B\}$. Assume the first "flippable" pixel in the kth block is p_k, $f_{p_k} = 1$, since the "flippability" of a pixel is invariant, so, $f_{p_k}{}' = f_{p_k} = 1$. The "embeddability" of the block is: $S_k = f_{p_k} = 1$. Flip p_k will affect the flippability of its eight neighbors, f_{q_k}, $q_k \in \{N(p)\}$. However, since U_k won't be flipped, q_k may be located farthest at the boundary, i.e., $q_k \in \{B\}$, $q_k \in \{A\} \cup \{B\} = \{P\}$. Therefore, q_k is still in the same block, thus, flipping a pixel in one block doesn't affect the "flippability" of pixels in its neighboring blocks. The "embeddability"

of this block is: $S_k{}' = f_{p_k}{}' \vee f_{q_k}{}' \vee ... = 1$, if $q_k \in \{A\}$ and $f_{q_k}{}' = 1$. Otherwise, $S_k{}' = f_{p_k}{}' = 1$. Hence, the "embeddability" of the block is invariant. The watermark can be extracted blindly from the "embeddable" blocks by computing the odd-even feature of the number of black or white pixels.

3. THE AUTHENTICATION MECHANISM

The odd-even enforcement is employed for the watermark embedding, which is vulnerable to "parity attack", i.e., an adversary can carefully flip two pixels while keeping the odd-even feature of the block unchanged. So, we propose to adopt a hard authenticator watermark to tackle this problem.

3.1. Locate Flipped Pixels

In order to generate the hard authenticator watermark, the key issue is how to locate the flipped pixel given the watermarked image. For the fixed 3×3 block, the flipped location is always the center pixel of the block, therefore it is easy to locate the flipped pixel.

Lemma 2. For non-interlaced block, if flipping the current pixel does not change the "flippability" of its previous four neighbors in the same 3×3 window, the "flipped" pixel can be located.

Proof. Pixels in the 3×3 block (Fig. 1) are processed in row by row and column by column sequence, i.e., w_6, w_7, w_8, w_5, p, w_1, w_4, w_3 and w_2. Assume p is the first "flippable" pixel in the block, i.e., $f_{w_6} = f_{w_7} = f_{w_8} = f_{w_5} = 0$ and $f_p = 1$. Given the condition, i.e., flip pixel p won't change the "flippability" of its previous four neighbors, we get $f_{w_6}{}' = f_{w_7}{}' = f_{w_8}{}' = f_{w_5}{}' = 0$. Since the "flippability" of a pixel is invariant, so, $f'_p = f_p = 1$. During the watermark extraction, pixels in the block are processed in the same sequence. Hence, the "flipped" pixel p can be located. The boundary pixels are excluded from flipping renders the minimum distance between any two "determined" pixels in two neighboring blocks is 2. Therefore, changes in pixels in one block won't affect the "flippability" of pixels in its neighboring block.

While for the interlaced block, flip p may affect one of the transition numbers of its previous four neighbors $\{p_4\}$. If $\{p_4\}$ lie in the sharing row or column, they may again be the previous four neighbors of pixels, e.g., m, n in its previous block. These pixels, e.g., m, n will be processed prior to pixel p. So, it may change the "embeddability" of its previous block. Therefore, the flipped locations can not be located by setting the same constraint. In this case, we suggest apply shuffling to the original image or to the "embeddable" and "unembeddable" blocks to increase the system security.

3.2. The Authentication Process

Fixed 3×3 block and non-interlaced block are employed in the hard authenticator watermark embedding process, which is summarized below and shown in Fig. 4.

1. Find the "embeddable" locations based on the steps S1-S5 discussed in Section 2.4. Criterions for locating the flipped pixels are also imposed.

2. Similar to clear LSB for grayscale images [3], clear the "embeddable" location by setting it to a fixed value, e.g., "0" to generate the intermediate image Y_1.

3. Fed Y_1 into a hash function to generate the hash value, $H_o = Hash(Y_1)$.

4. Encrypt H_o by the private key K_s of the owner or issuer, e.g., RSA private key to generate the content signature of the document, $W_s = E_k(H_o, K_s)$.

5. XOR (Exclusive OR) or concatenate W_s with the payload watermark W_p to generate the authenticator watermark, e.g., $W_r = W_s \parallel W_p$.

6. Embed W_r in the "embeddable" blocks based on the odd-even feature of the block.

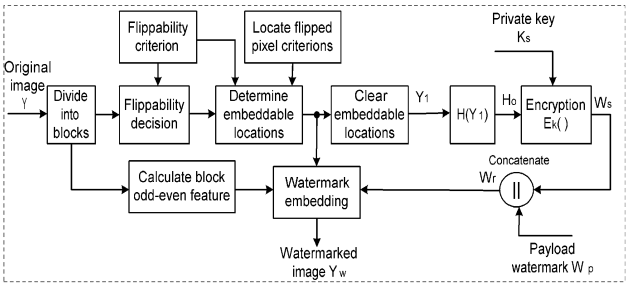

Fig. 4. Block diagram of hard authenticator watermark embedding process.

3.3. The Verification Process

The hard authenticator watermark verification process are summarized below and shown in Fig. 5.

1. The first three steps, i.e., find the "embeddable" locations, generate the intermediate image Y'_1 and generate hash of the watermarked image H_w are the same as steps 1-3 in the embedding process.

2. Extract the watermark based on the odd-even feature of the "embeddable" blocks, split it into two parts: the content signature W'_s and the payload W'_p.

3. Employ the public key K_p, e.g., RSA public key to decrypt W'_s, e.g., the first 1024 bits to obtain the hash value of the original image $H'_o = D_k(W'_s, K_p)$.

4. Compare W'_p with W_p and H_w with H'_o. If H'_o match H_w and W'_p is the same as W_p, the authenticity and integrity of the document can be ensured.

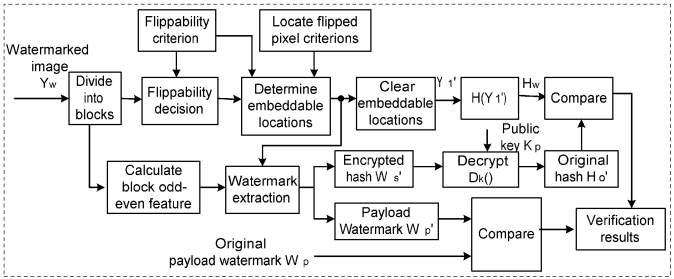

Fig. 5. Block diagram of hard authenticator watermark verification process.

4. EXPERIMENTAL RESULTS

A wide varieties of images, including cartoon images, English, Japanese, French, Chinese and handwritten text images are used to test the capacities of using different types of blocks. The results are shown in Table 1. It can be seen from the results, by employing the non-

Table 1. Capacity comparisons of different types of blocks.

File	*Size*	Capacity (bits)			
		FB 3×3	*NIB* 4×4	*IB* 3×3	*IB* 4×4
Fre	512×512	1795	2448	3383	4389
Gir	361×359	248	261	396	478
Chi	336×336	482	733	1052	1261
Typ	336×336	447	672	1006	1235
Han	336×336	313	454	741	972
Jap	336×336	526	822	1180	1488

interlaced block of size 4×4, the capacity increases compared with a fixed 3×3 block. By employing interlaced block of size 3×3, the capacity increases further. Experimentally, the use of interlaced block with size 4×4 gives the largest capacity.

Experiments are also conducted to verify the effectiveness of the proposed hard authenticator watermark. A logo image is used as the payload watermark to visually show the tamper occurred to the watermarked image. The results are shown in Fig. 6. It can be

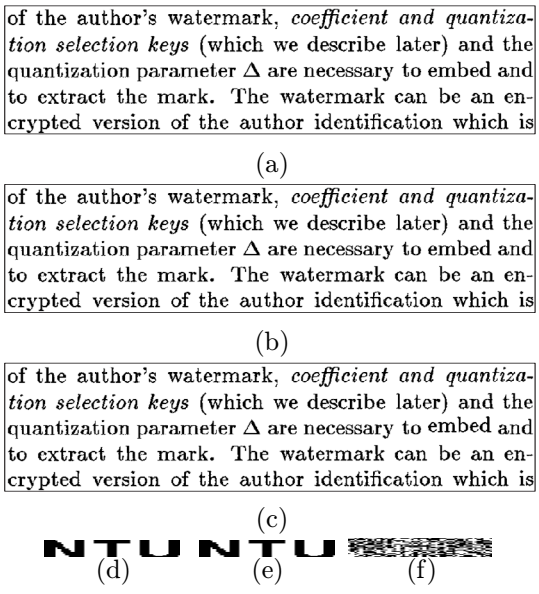

Fig. 6. Authentication results. (a) the original image of size 920×230, (b) hide 1056 bits by the proposed algorithm (fixed 3×3 block) and (c) the watermarked image which is tampered, "embed" in the 3rd line is shifted slightly. (d) the original logo image, (e) the reconstructed logo image (no tamper) and (f) the reconstructed logo image (tampered).

observed from the results that the proposed hard authenticator watermark is effective in detecting any tampering made to the watermarked document. The logo image can be reconstructed successfully when no tampering occurs. However, when tamper occurs, even the tamper is small, e.g., only one word is shifted slightly, the computed hash varies significantly.

Comparisons of the visual effects of the proposed method with methods proposed by Wu *et al.* [2] and Tseng *et al.* [1] are made. The block size of 12×12 is chosen for Wu's method to ensure that each block has a suitable pixel to flip. Same block size is chosen for Tseng's method. The results are shown in Fig. 7. It can be observed from the results that our proposed

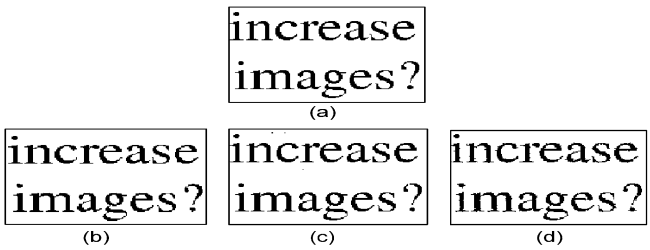

Fig. 7. Comparison results. (a) the original image (173 $\times$115), (b) hide 266 bits by the proposed algorithm (interlaced block of size 4×4), (c) hide 126 bits by Wu's method and (d) hide 756 bits by Tseng's method.

method achieve good visual results compared with Wu *et al.*'s method and Tseng *et al.*'s method. The watermarked image of Tseng's method looks noisy due to the randomness in choosing the embedding locations. The visual effects of our method and Tseng's method can be further improved if the same amount of bits is hidden because of the large capacity of the two methods.

5. CONCLUSIONS

In this paper, a novel blind data hiding scheme for binary images based on connectivity preserving of pixels in a local neighborhood is presented. A window of size 3×3 is employed to assess the "flippability" of a pixel in a block. Watermark is only embedded in those "embeddable" blocks based on the three transition criterions. The fixed 3×3 block, non-interlaced and interlaced block are employed and the capacity of using different types of blocks are compared. Experimentally, it is shown that the interlaced block with size 4×4 gives the largest capacity. A hard authenticator watermark is employed which is effective in detecting any tampering to the watermarked image.

6. REFERENCES

[1] Y. C. Tseng and H.-K. Pan, "Data Hiding in 2-Color Images," IEEE Transactions on Computers, Vol. 51, No. 7, pp. 873-878, July 2002.

[2] M. Wu, E. Tang and B. Liu, "Data Hiding in Digital Binary Image", IEEE International Conf. on Multimedia and Expo., 2000, pp. 393-396.

[3] P. W. Wong, Memon, N. D., "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.