

SECURE DATA HIDING IN BINARY DOCUMENT IMAGES FOR AUTHENTICATION

Haiping Lu, Alex C. Kot and Jun Cheng

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

ABSTRACT

In this paper, we present a data hiding algorithm for binary document images. This algorithm is based on the Distance-Reciprocal Distortion Measure [1] that is used to evaluate the amount of distortion caused by flipping a particular pixel in binary document images. The pixels that will cause less distortion after flipping are preferred candidates for flipping. We do the embedding by enforcing the odd-even features of non-uniform blocks and employ a 2-D shifting to provide security for tamper proofing and authentication. Experiments show that the watermark-embedded document image has good quality and tampering of content can be detected successfully.

1. INTRODUCTION

Today, digital media are getting more and more popular. Not only multi-level images, video and audio are in digital form, but binary document images are also digitized in many applications including legal documents, digital books, maps, and architectural and electronic drawings. Digital watermarking techniques have been proposed for ownership protection, copy control, annotation and authentication of digital media. When the purpose is for annotation or authentication, digital watermarking is often called data hiding and these two terms are used interchangeably by many authors. Most of image data hiding techniques in the literature are proposed for grayscale or color images, while data hiding for binary images are only addressed by a few authors [2–4].

In [4], Wu *et al.* propose a data hiding algorithm for digital binary images. A set of rules are used to calculate the flipping scores of pixels and shuffling is employed to handling the problem of uneven embedding capacity. Recently, Lu *et al.* [1] propose a Distance-Reciprocal Distortion Measure (DRDM) for binary document images. It has been shown that this measure has much better correlation with human visual perception than PSNR (peak signal-to-noise ratio) [5] when applied to binary document images.

In this paper, we propose a secure data hiding algorithm based on the DRDM measure for the purpose of authenti-

cation of digital documents in a binary image format. We combine a 2-D shifting technique with an odd-even embedding scheme and use the DRDM scheme to choose the appropriate pixels to flip. Experiments show that the algorithm has good imperceptibility and can be used for tamper proofing and authentication.

2. REVIEW OF DRDM

DRDM method [1] uses a weight matrix to measure the distortion. For an input binary document image $f(x, y)$ of size $M \times N$, suppose that the output binary document image after some processing is $g(x, y)$. A weight matrix \mathbf{W}_m of size $m \times m$ is calculated, $m = 2n + 1$, $n = 1, 2, 3, \dots$, where

$$\mathbf{W}_m(i, j) = \begin{cases} 0 & \text{for } i = i_C, j = j_C \\ \frac{1}{\sqrt{(i-i_C)^2 + (j-j_C)^2}} & \text{otherwise.} \end{cases} \quad (1)$$

for $1 \leq i, j \leq m$ and $i_C = j_C = (m + 1)/2$ is the center location of the matrix. This matrix is normalized to form the normalized weight matrix $\bar{\mathbf{W}}_m$.

$$\bar{\mathbf{W}}_m(i, j) = \frac{\mathbf{W}_m(i, j)}{\sum_{i=1}^m \sum_{j=1}^m \mathbf{W}_m(i, j)} \quad (2)$$

Supposing that there are P flipped pixels in $g(x, y)$, each pixel will have a distortion d_k , $k = 1, 2, 3, \dots, P$. For the k^{th} flipped (from black to white or from white to black) pixel at $(x, y)_k$ in the output image $g(x, y)$, the resulted distortion is calculated from a $m \times m$ block \mathbf{B}_k in $f(x, y)$ that is centered at $(x, y)_k$. The distortion measure d_k for this flipped pixel $g[(x, y)_k]$ is given by [1]

$$d_k = \sum_{i,j} [\mathbf{D}_k(i, j) \times \bar{\mathbf{W}}_m(i, j)] \quad (3)$$

where the elements of the difference matrix \mathbf{D}_k are given by

$$\mathbf{D}_k(i, j) = |\mathbf{B}_k(i, j) - g[(x, y)_k]| \quad (4)$$

For possibly flipped pixels near the corners or borders, where a $m \times m$ neighborhood may not exist, we choose

to expand the rest of $m \times m$ neighbors with the opposite value of $g[(x, y)_k]$ in data hiding since they are likely to be obvious isolated points against the background.

The distortion in $g(x, y)$ is then calculated as:

$$d = \frac{\sum_{k=1}^P d_k}{K} \quad (5)$$

where K is defined as the number of non-uniform (not all black or white pixels) 8×8 blocks in $f(x, y)$.

3. EFFECTS OF VARYING WEIGHT MATRIX SIZE

We studied the effects of varying weight matrix size m on the distribution of pixels (assuming flipped) with a smaller d in non-uniform 8×8 blocks. We use a text image of 512×512 obtained from one of the CCITT binary images [3] to demonstrate the effects. The original image is shown in Fig. 1.

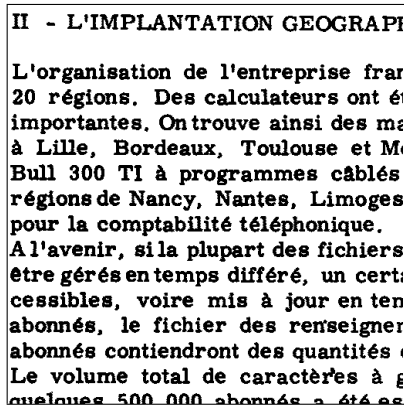


Figure 1: Original binary document image.

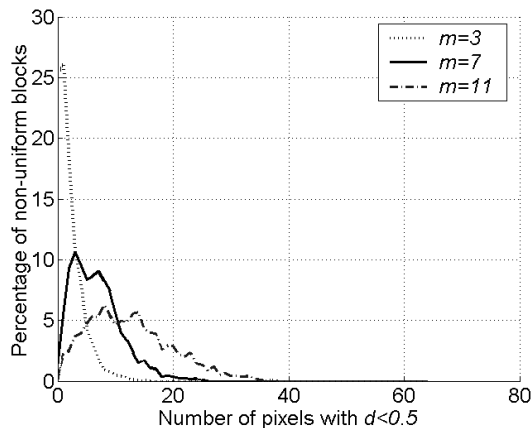


Figure 2: Distribution of pixels (assuming flipped) with $d < 0.5$ for the non-uniform 8×8 blocks shown in Fig. 1.

Fig. 2 shows the distribution of pixels with d less than 0.5 for the non-uniform 8×8 blocks shown in Fig. 1, with weight matrix size $m = 3, 7$ and 11. We observe that as m increases, the number of pixels with d less than 0.5 tends to distribute more evenly. Thus, the problem of uneven embedding capacity [4] becomes much less serious for a larger m . Not including the uniform (all black/white) blocks also helps to reduce this problem.

4. DATA HIDING ALGORITHM BASED ON DRDM

Fig. 3 shows the system flow of the proposed algorithm. We embed the watermark w into the original binary image $f(x, y)$ using a shift key k_s to obtain the output image $g(x, y)$. The image is of size $M \times N$.

4.1. 2-D Shifting

To embed the watermark, we first obtain $f_s(x, y)$, a shifted version of $f(x, y)$, using k_s . We do a circular left-shifting of N_s rows each time for all M rows first and then N_s columns each time for all N columns. The amount of shifting is determined by L_s bits from the key. Hence k_s is of length

$$L_{k_s} = L_s \cdot (\lceil M/N_s \rceil + \lceil N/N_s \rceil) \quad (6)$$

where $\lceil \cdot \rceil$ is the ceiling function, $1 \leq N_s \leq \min[M, N]$ and $L_s = \lceil \log_2 N_s \rceil$.

The 2-D shifting using k_s provides security for the simple odd-even embedding strategy [4] so that it is difficult to learn the watermark from the odd-even features of the marked image without the knowledge of the key, and an attacker can hardly modify the marked image without affecting the odd-even features in extraction [6]. Security level provided by this shifting increases with smaller N_s .

4.2. Watermark Embedding

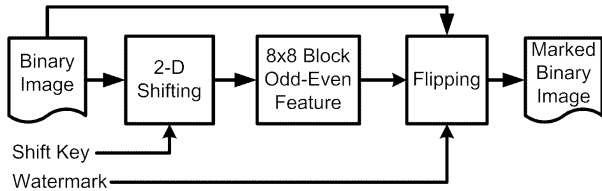
We embed the watermark by enforcing the odd-even features of the non-uniform 8×8 blocks in $f_s(x, y)$, with even number of black pixels in the block for embedding bit ‘0’ and odd for bit ‘1’. We skip the uniform blocks to preserve the quality of the image after embedding.

Flipping is necessary when the bit to be embedded does not match with the block’s odd-even feature. For example, when the number of black pixels in the block is odd (even) while the bit to be embedded is ‘0’ (‘1’). Wu *et al.* [4] use a set of rules or a precalculated look-up table to choose the pixels to flip. While this is not difficult for 3×3 patterns, it is quite complicated if we consider larger area of neighborhood. DRDM introduced in [1] provides a fast and accurate way to evaluate the distortion resulted from flipping a pixel, and it is able to consider an “arbitrarily” large neighborhood by choosing a large m . It also correlates well with human

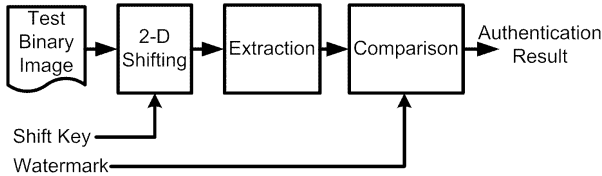
visual perception. Therefore, when flipping is necessary, we choose the pixel (assuming flipped) with the lowest DRDM value d in the block to flip.

We obtain the odd-even features from the shifted image $f_s(x, y)$, while for the blocks that need flipping, we calculate d in $f(x, y)$ so that the image after embedding can still preserve good quality. Flipping is done in both $f(x, y)$ and $f_s(x, y)$ so that there is no need to do reverse shifting after embedding. The shift key k_s is used to find the mapping from pixels in $f_s(x, y)$ to those in $f(x, y)$. However, it is possible that such flipping changes the current block in $f_s(x, y)$ from a non-uniform block to a uniform block. Whenever this happens, we record this change, do the flipping in $f(x, y)$ and $f_s(x, y)$, and postpone the embedding of the current bit to the next non-uniform block.

We assume that there are l_p blocks changed from non-uniform ones to uniform ones when we embed $w(p)$, the p^{th} bit of w , where $p = 1, 2, 3, \dots, L_w$ and L_w is the length of w . The bit $w(p)$ is embedded into the q^{th} non-uniform 8×8 block in $f_s(x, y)$ and $q = p + l_p$. After all necessary flipping in $f(x, y)$ to embed w , we have the marked binary image $g(x, y)$. The total number of blocks changed from non-uniform ones to uniform ones is L_r .



(a) Data embedding.



(b) Data extraction and authentication.

Figure 3: Data hiding algorithm for binary document image.

4.3. Embedding Capacity

We denote the number of non-uniform 8×8 blocks in $f_s(x, y)$ as K_s , and the number of 8×8 blocks in $f_s(x, y)$ with all except one pixels having the same pixel value as Q_s . Then, the guaranteed embedding capacity C_g is given by

$$C_g = K_s - Q_s \quad (7)$$

In practice, we can still embed $(Q_s - L_r)$ more bits, depending on the image and the watermark. From the discussion above, we have $0 \leq L_r \leq Q_s$.

4.4. Watermark Extraction

Watermark extraction is performed in a similar way as embedding. To extract the watermark \hat{w} from a test binary image $g(x, y)$, we do a 2-D circular left-shifting using k_s to obtain $g_s(x, y)$. The odd-even features in all non-uniform 8×8 blocks of $g_s(x, y)$ are extracted as \hat{w} . This extracted watermark \hat{w} is then compared with the original watermark w to give the authentication result, and $\hat{w} = w$ only when $g(x, y)$ is an intact marked image.

5. EXPERIMENTAL RESULTS

We use the image in Fig. 1 as the original image $f(x, y)$. For programming convenience, we choose $N_s = 8$ in our experiments. Thus, $L_s = 3$ and $L_{ks} = 384$. We use the weight matrix \bar{W}_7 ($m = 7$) in embedding and it is only calculated once. There are 2512 ($\approx 61\%$) non-uniform 8×8 blocks out of 4096 in $f(x, y)$. After the 2-D shifting using a randomly generated key of length 384, we obtain $f_s(x, y)$ with $K_s = 2539$ and $Q_s = 53$. Therefore, the guaranteed embedding capacity C_g is 2486 bits, and we generate a random binary sequence of length $L_w = 2486$ as w for embedding.

In the embedding, we have $L_r = 23$ ($\approx 0.9\%$) after necessary flipping. Thus we embed 30 more bits in the experiment and have 2516 bits of data embedded successfully. The image after embedding is shown in Fig. 4 and there are 1247 pixels flipped in the image.

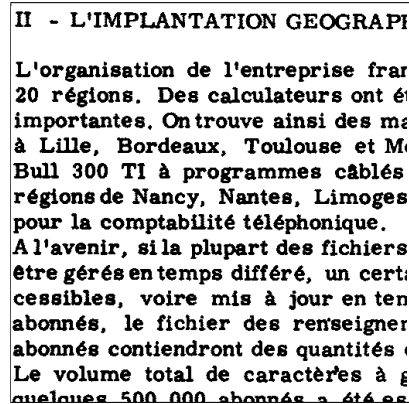


Figure 4: Binary document image after embedding.

From this figure, we can see that the image after embedding has good quality and flipping is hard to perceive. The quality measured in PSNR is 23.23dB and the DRDM measure $d = 0.182$ with $m = 5$. It has been shown in [1] that DRDM is a better distortion measure for binary document images than PSNR. From our experience, a d of value below 0.2 is generally considered of good quality.

To show how effective DRDM scheme is in choosing

most suitable pixels to flip, we cut the center 100×100 portion of the original image, as shown in Fig. 5(a), and the pixels flipped in this portion of the marked image are shown in Fig. 5(b) as black dots, with the original black pixels brightened.

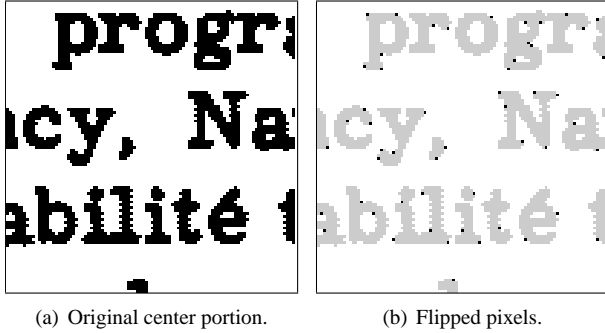


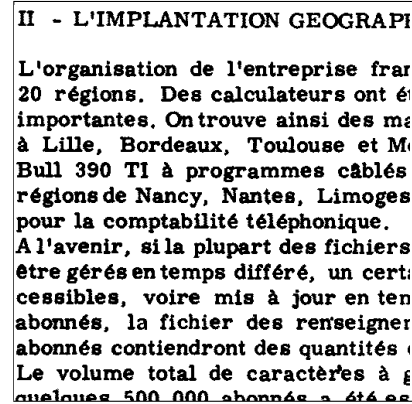
Figure 5: Flipped pixels for the center 100×100 portion.

We also simulated an active tampering to test the tamper proofing ability of our algorithm, which is important for authentication. We modify the content of the marked image and force the uniform/non-uniform property and odd-even features agreeing with the original marked image.

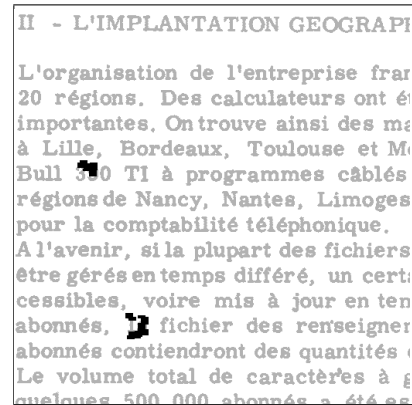
The image after such a simulated attack is shown in Fig. 6(a). We modify the number “300” (second word) at the fifth row (not including the top row of title) to “390”, and the second word “le” at the third last row (not including the bottom half row) to “la”. If the 2-D shifting is not used in the embedding, we will extract a watermark \hat{w} that is exactly the same as w , showing that the marked image is intact. The 2-D shifting makes such attack detectable as shown in Fig. 6(b) since the modification has changed the odd-even features of the shifted version of the marked image though such features are intact in the version before shifting. There are 10 bits error in the extraction, which implies that the modification has affected the odd-even features of at least 10 blocks.

6. CONCLUSION

A secure data hiding algorithm for binary document images is proposed in this paper. This algorithm is based on the Distance-Reciprocal Distortion Measure that provides an efficient way to select the pixels to flip in embedding. The distortion due to flipping is calculated online and able to take the effect of a large area of neighbor pixels into accounts. Data is embedded by enforcing the odd-even features of non-uniform blocks and the 2-D shifting is employed to provide security against tampering. Experiments show that the marked image has good quality and tampering can be detected in the extraction.



(a) Tampered image.



(b) Tampering detected.

Figure 6: Tamper proofing for the tampered image.

7. REFERENCES

- [1] H. Lu, J. Wang, A. C. Kot, and Y. Q. Shi, “An objective distortion measure for binary document images based on human visual perception,” in *Proc. Int. Conf. Pattern Recognition*, vol. 4, Quebec, Canada, Aug. 2002, pp. 239–242.
- [2] M. Chen, E. K. Wong, N. Memon, and S. Adams, “Recent developments in document image watermarking and data hiding,” in *Proc. SPIE Conf. 4518: Multimedia Systems and Applications IV*, Aug. 2001, pp. 166–176.
- [3] H. Lu, X. Shi, Y. Q. Shi, A. C. Kot, and L. Chen, “Watermark embedding in DC components of DCT for binary images,” in *Proc. Int. Workshop on Multimedia Signal Processing*, US Virgin Islands, Dec. 2002.
- [4] M. Wu, E. Tang, and B. Liu, “Data hiding in digital binary image,” in *Proc. IEEE Int. Conf. on Multimedia & Expo*, New York, NY, 2000, pp. 393–396.
- [5] Y. Q. Shi and H. Sun, *Image and Video Compression for Multimedia Engineering: Fundamental, Algorithm, and Standards*. Boca Raton, FL: CRC Press LLC, 1999.
- [6] D. Kundur and D. Hatzinakos, “Digital watermarking for tell-tale tamper-proofing and authentication,” *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1167–1180, July 1999.