Print-and-photocopy Resilient Text Watermarking based on Hadamard Transform

Xiaolong Li, Bin Yang and Wenfa Qi
Institute of Computer Science and Technology, Peking University, Beijing 100871
E-mail: {lixiaolong,yangbin,qiwenfa}@icst.pku.edu.cn

***Abstract*—In this paper, a novel watermarking algorithm is proposed for copy tracking of paper-based text documents. By modifying AC coefficients, the watermark is modulated and extracted in the transform domain of Hadamard transform, which is applied on the numbers of black pixels in each character of the binary text image. Meanwhile, by tuning DC coefficient, the distortion of each text character can be well controlled. The Hadamard transform is chosen based on the observation that their AC coefficients are less sensitive to print-and-photocopy. The novel algorithm can embed adequate data into office-like text document while keeping visual appearance, and resist print-and-photocopy attack as well. Practically, by this method, we can embed identity information into electronic text files. When obtaining a paper-based illegal copy, we can restore it into electronic format by scanner and extract the embedded data through watermark decoding, then identify the source of the copy. This can improve the management of confidential files. The validity of our method is demonstrated via extensive experiments.**

I. Introduction

The rapid development of information technology requests strongly the copyright protection and content identification for digital data. Digital watermarking, which progressively advances under this requirement, is becoming a hot topic of information security. This technique intends to embed useful information (the watermark) into digital covers in an imperceptible and robust way.

Nowadays, papers are still an important data carrier, and many paper-based text documents (e.g., confidential government files) are in fact more valuable than digital data and have a higher level of security. Hence, to improve the management of confidential files, encrypting paper-based text documents, just as what we do for electronic files, is of great importance.

Digital watermarking can provide a solution to this issue. By this technique, we can embed identity information (e.g., printer name, user-ID, printing time, etc.) into electronic text files. When obtaining a paper-based illegal copy, we can restore it into electronic format by scanner and extract the embedded data through watermark decoding, and then identify the source of the copy. Whereas, importantly, the watermarking algorithm should have the following features:

(1) The marked text document should look the same as the original one.
(2) The embedding capacity should be large enough to distinguish different identity information.
(3) Not only for the printed document, but also for its photocopies, one is able to extract the watermark.

Based on these considerations, we propose in this paper a novel watermarking algorithm for copy tracking of paper-based text documents. By modifying AC coefficients, the watermark is modulated and extracted in the transform domain of Hadamard transform, which is applied on the numbers of black pixels in each character of the binary text image. Meanwhile, by tuning DC coefficient, the distortion of each text character can be well controlled. Here, the Hadamard transform is chosen based on the observation that their AC coefficients are less sensitive to print-and-photocopy. Extensive experiments illustrate that, without reducing the visual quality, our method can embed more than 100 bits into an A4 sized ($21\mathrm{cm} \times 29.7\mathrm{cm}$) office-like text document, and can resist print-and-photocopy attack. In this way, the novel method satisfies the above requirements (1)-(3).

The rest of this paper is organized as follows. Some text watermarking algorithms are briefly reviewed in Section II. Then in Section III, the proposed method is introduced in detail. The experimental results, as well as discussions on the practical implementation of our method, are reported in Section IV. Finally, we conclude this work in last section.

II. Related works

Existing text watermarking algorithms can be roughly classified into five types: file-structure-based method, character-feature-based method, mesh-pattern-based method, natural-language-processing-based method and binary-image-watermarking-based method. Here we give a review.

File-structure-based method mainly considers embedding watermark by slightly adjusting the file structure according to the document characteristics, such as inter-line space, inter-word space and inter-character space [1]–[3]. For instance, in the line-shift encoding method of Brassil *et al.* [1], a line is moved up or down to embed one bit 0 or 1, and the line centroids are used as references for blind decoding. However, the embedding capacity of this type of methods is low, for instance, one can only embed about 40 bits into an A4 sized text document by the line-shift encoding method.

Character-feature-based method is similar to the above approach, in which character features (e.g., size, topological structure, luminance, stroke location, etc.) are manipulated to generate some distinguishable patterns. For instance, shifting up or down a stroke may get two patterns which correspond one bit 0 or 1 [4], [5]. However, these manipulations will deeply affect characters' appearance and can be easily recognized by human eyes.

Mesh-pattern-based watermark is a specific type of methods, in which micro-dots are inserted into the background of text document to embed watermark [6]–[8]. This type of methods can provide a high capacity, and can resist photocopy attack as well. But the inserted micro-dots will dramatically reduce the appearance and readability of text documents. Moreover, additional ink cost for printing micro-dots may limit the practicability of these methods.

Natural-language-processing-based method is another important type of text watermarking [9]–[12]. For these methods, watermark is embedded either by manipulating the semantic/syntactic structure of sentences without significantly altering their meaning, or by replacing words with their synonyms or variant forms. This type of methods has good imperceptibility and robustness. However, it is not applicable for text documents whose contents can not be changed, e.g., contracts or official documents.

Finally, we introduce binary-image-watermarking-based method, where text documents are viewed as black-and-white images. Current binary image watermarking algorithms are mainly block-based embedding [13]–[15]. In these methods, host image is first divided into blocks, and then data is embedded block by block. The representative binary image watermarking is Wu and Liu's method [13]. In [13], the authors illustrated that their algorithm can resist print attack, but a dash box is needed to determine the embedding region and to accurately restore the scanned image. Furthermore, in the experiment, the original image's resolution is 72 dpi (dot per inch) and the printed image is scanned at a high resolution of 600 dpi. This is equal to making a zoom-in on the original image and is more favorable to restore the scanned image. Notice that the resolution of a typical printer can not be lower than 300 dpi, so these restrictions may limit the practicability of Wu and Liu's method. Moreover, we remark that the block-based methods are usually fragile watermarking and can not be used directly for our purpose.

In summary, current text watermarking algorithms are hard to meet the requirements (1)-(3) described in Section I, hence difficult to be used for copy tracking.

III. The proposed method

A. Basic idea

Modeling print-scan (PS) and print-photocopy-scan (PPS) procedures is a difficult task, since these procedures heavily depend on the equipments. For instance, even for a specific printer, the PS noise is variable in different printing statuses, and the printed document may be darker or lighter due to different print density levels. So, it is not an ideal way to design print-and-photocopy resilient text watermarking by modeling PS/PPS procedures. We then prefer to find PS/PPS invariants and use them to embed watermark.

Notice that black-and-white printers can only receive rasterized binary image data, thus our method is designed for text document in binary image format. In brief, the binary text image is first divided into characters, and then the watermark is embedded by flipping boundary pixels (i.e., we change black boundary pixels to white, or vice versa) of each character. Here, a pixel is called boundary pixel, if nine pixels in its 3×3 neighborhood are not all black or all white.

Let us first see Fig. 1(a). It shows the number of black pixels (NBP) in each of 100 Chinese characters (font-size: 12pt, typeface: "Kai Ti"), before and after printing. This figure is obtained as follows. First, the text document is printed by two printers, Kyocera KM-5035 (an "all-in-one" printer-copier) and HP LaserJet 5100. Then, the printed paper-based documents are scanned by a scanner, HP ScanJet 4890, and restored into electronic files in gray-scale format. Finally, the gray-scale images are binarized to get binary images. Here, the resolution of printer and scanner is set to 600 dpi. We point out that the average NBP of the original text document is 1652, and this value is changed to 2009 (Kyocera KM-5035) and 1380 (HP LaserJet 5100) after PS. The variation of NBP is thus significant. However, for a specific printer, there is a notable fact that for all characters, their NBP are either increased or decreased . The simple observation gives us a clue to find PS/PPS invariants.

Referring to Fig. 1(b), it shows the difference of the number of black pixels (DNBP) of two adjacent characters, for the same document used in Fig. 1(a), before and after printing. In contrast to NBP, we see that DNBP is almost unchanged after PS. It illustrates that the "minus" operation between two adjacent characters will significantly eliminate the PS noise. In this way, the variation of DNBP is much smaller than that of NBP. Actually, the observation is also valid for photocopied documents (compare Figs. 1(c) with 1(d)). Here, in Figs. 1(c)-1(d), the text documents are first printed by the two printers and then photocopied by Kyocera KM-5035.

In conclusion, DNBP is much more stable than NBP, and this quantity can be approximately viewed as a PS/PPS invariant. Therefore, it is reasonable to explore DNBP to design print-and-photocopy resilient text watermarking.

B. Hadamard transform

The Hadamard transform will be employed in our method. We first define inductively the Hadamard matrix $\mathbf{H}_n$:

$$\mathbf{H}_0 = (1), \qquad \mathbf{H}_{n+1} = \begin{pmatrix} \mathbf{H}_n & \mathbf{H}_n \\ \mathbf{H}_n & -\mathbf{H}_n \end{pmatrix}.$$

These matrices are symmetric and satisfy

$$\mathbf{H}_n^2 = 2^n \mathbf{Id}_{2^n},$$

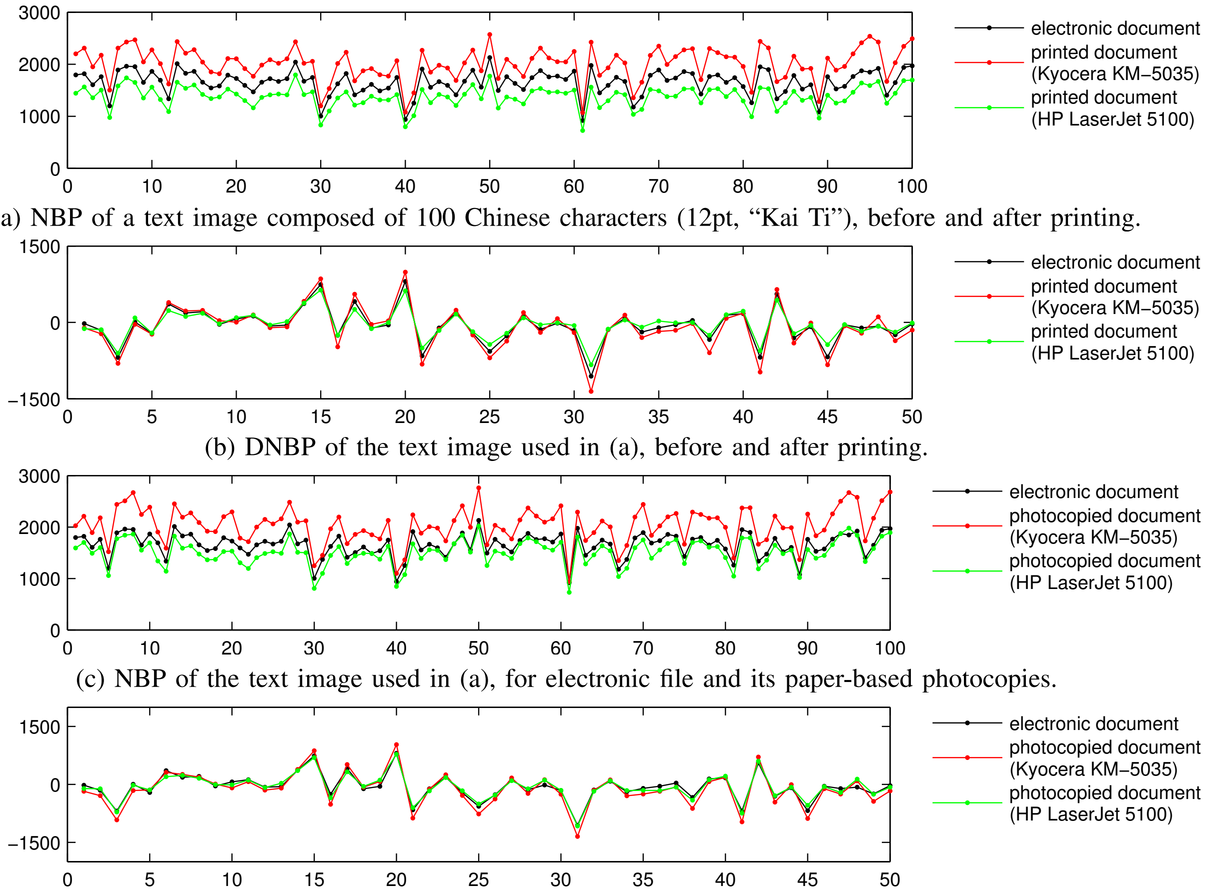
where $\mathbf{Id}_k$ is the k-by-k identity matrix. Thus we have

$$\mathbf{H}_n^{-1} = 2^{-n}\mathbf{H}_n.$$

We then define the Hadamard transform:

$$\mathcal{H}_n : \mathbb{R}^{2^n} \mapsto \mathbb{R}^{2^n}, \qquad \mathcal{H}_n(\mathbf{x}) = \mathbf{H}_n\mathbf{x}.$$

It is a linear transform with inverse $\mathcal{H}_n^{-1} = 2^{-n}\mathcal{H}_n$.



(a) NBP of a text image composed of 100 Chinese characters (12pt, "Kai Ti"), before and after printing.

(b) DNBP of the text image used in (a), before and after printing.

(c) NBP of the text image used in (a), for electronic file and its paper-based photocopies.

(d) DNBP of the text image used in (a), for electronic file and its paper-based photocopies.

Fig. 1. Comparison between NBP and DNBP.

Except the first one, each row vector of Hadamard matrix is composed of equal amounts of 1 and -1. Thus in the transform domain of Hadamard transform, every AC coefficient is a difference between sums of equal amounts of spatial domain coefficients. This property provides a way to extend the idea introduced in subsection III-A. Besides, utilizing Hadamard transform may increase the embedding capacity. Actually, $2^n - 1$ bits can be embedded into 2^n characters by taking $\mathcal{H}_n$, and thus each character may approximately carry one watermark bit when n is large enough.

C. Strategy for flipping boundary pixels

Binary image only has two luminance levels 0 (white) and 1 (black). Flipping even one single pixel will be noticeable, especially for non-boundary ones. In fact, only a part of boundary pixels in binary image can be flipped and used to embed watermark. We then discuss how to determine flippable boundary pixels in binary text image. The following discussion extends Wu and Liu's work in [13].

For a pixel x of a binary image I, we define its visual complexity as

$$V(x) = \sum_{i=1}^{8} |I(y_i) - I(y_{i+1})|,$$

y_1	y_8	y_7
y_2	x	y_6
y_3	y_4	y_5

Fig. 2. The 3×3 neighborhood of a pixel x.

where y_i are neighbors of x (see Fig. 2), $I(y_i) \in \{0, 1\}$ are pixel values, and we take $y_9 = y_1$ as convention. Obviously, $V(x) \in \{0, 2, 4, 6, 8\}$. Table I shows the distribution of visual complexity of boundary pixels. We see that for binary text image, there are few boundary pixels with visual complexity 0, 4, 6 or 8. The fact suggests that we may only consider boundary pixels with visual complexity 2. For convenience, such a pixel is noted as simple boundary pixel (SBP).

In [13], Wu and Liu devised a flippability score to each boundary pixel. The pixel with a larger score means that flipping it is less noticeable, and the one with a score lager than 0.1 is considered to be flippable. The scores of SBPs are presented in Fig. 3. We see that according to Wu and Liu's method, SBP with type 2W/32W/4W/51W/6W/71W/72W/11B/12B/2B/31B/4B/52B/6B

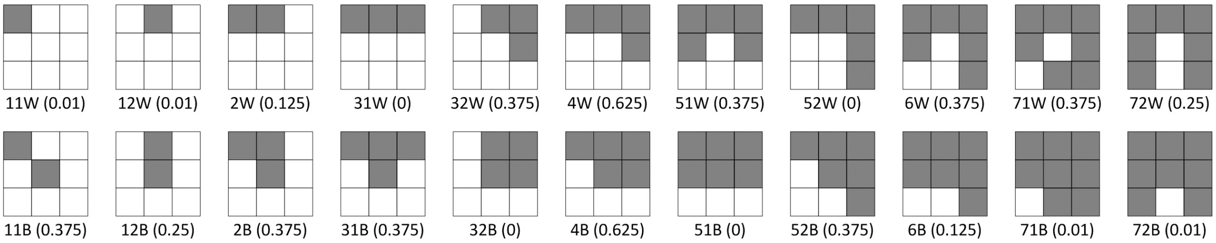
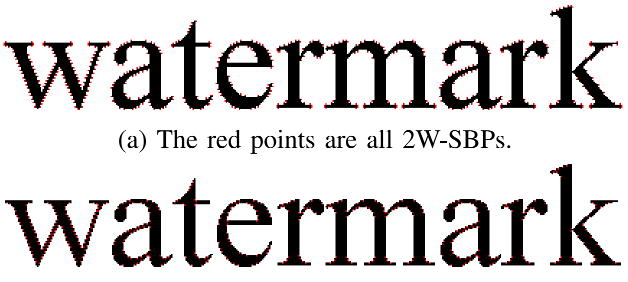


Fig. 3. Different patterns of SBP (the center one) and their flippability scores, excluding the patterns that differ by rotation or mirroring. Here, the letter W/B means that the SBP is white/black.

TABLE I
DISTRIBUTION OF VISUAL COMPLEXITY OF BOUNDARY PIXELS, FOR FOUR TEXT DOCUMENTS. HERE, EACH TEXT CONTAINS 1000 CHINESE CHARACTERS (12PT, "KAI TI" OR "SONG TI") OR ENGLISH WORDS (12PT, "TIME NEW ROMAN" OR "ARIAL").

$V(x)$	Chinese "Kai Ti"	Chinese "Song Ti"	English "Time New Roman"	English "Arial"
0	$< 0.01\%$	$< 0.01\%$	0.00%	0.00%
2	99.42%	99.34%	99.66%	99.77%
4	0.57%	0.65%	0.34%	0.23%
6	$< 0.01\%$	$< 0.01\%$	0.00%	0.00%
8	0.00%	0.00%	0.00%	0.00%



(a) The red points are all 2W-SBPs.

(b) The red points are all 6B-SBPs.

Fig. 4. Illustration of non-flippable SBPs.

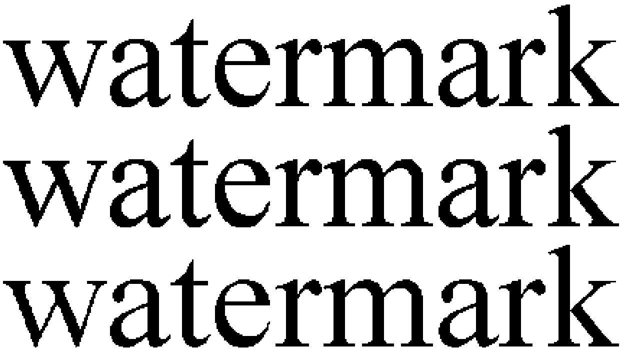


Fig. 5. Flipping results by using the proposed boundary pixel flipping strategy. From top to bottom: English word "watermark", the image generated by flipping half of flippable white SBPs, and the image generated by flipping half of flippable black SBPs, respectively.

is flippable. However, we argue that flipping SBP with type 2W/6B is in fact sensitive to human eyes. Referring to Fig. 4, all 2W- and 6B-SBPs are marked as red points. Flipping these pixels may destroy local smoothness and make image boundary scraggly. So, 2W- and 6B-SBP can not be flipped.

Based on the above discussion, we propose a strategy to flip SBP. For a character, if its black pixels need to be increased, we find all white SBPs with type 32W/4W/51W/6W/71W/72W, then flip them according to the number of white pixels that need to be flipped. More specifically, we first find all 4W-SBPs (i.e., white SBPs with the highest score) and sort them randomly, then flip these SBPs sequentially. If flipping all 4W-SBPs is not sufficient, we may find all 32W-SBPs and sort them randomly, then flip them sequentially, etc. Similarly, for a character, if its black pixels need to be decreased, we may find all black SBPs with type 11B/12B/2B/31B/4B/52B, then flip them accordingly. Moreover, to get better visual quality, if an SBP is flipped, the other SBPs in its 3×3 neighborhood will not be flipped. Fig. 5 shows flipping results of this strategy. From top to bottom, the three images show English word "watermark", the image generated by flipping half of flippable white SBPs, and the image generated by flipping half of flippable black SBPs, respectively. We see that the flipped images look almost the same as the original one, only a little thicker or thinner.

D. Watermark embedding

Step 1: The binary text image is first divided into characters. Then the characters are divided into groups such that each group contains 2^n characters, where $n > 0$ is a given integer.

Step 2: For a character group, let $\mathbf{x} = (x_0, x_1, ..., x_{2^n-1})$ be the NBP of its characters. Take the Hadamard transform of $\mathbf{x}$ to get $\mathbf{y} = (y_0, y_1, ..., y_{2^n-1}) = \mathcal{H}_n(\mathbf{x})$. Here, $y_0 = \sum_{i=0}^{2^n-1} x_i$ is the DC coefficient, and $y_1, ..., y_{2^n-1}$ are AC coefficients.

Step 3: For each $i \in \{1, ..., 2^n - 1\}$, through quantization, embed one bit $w_i \in \{0, 1\}$ into the AC coefficient y_i to get y'_i:

$$y'_i = \underset{y\in\{kQ:k\in\mathbb{Z},\, k\equiv w_i (\bmod 2)\}}{\arg\min} |y - y_i|, \tag{1}$$

where $Q > 0$ is a predefined quantization step size. In this way, y'_i is the even multiple of Q nearest to y_i if $w_i = 0$, while it is the odd multiple of Q nearest to y_i if $w_i = 1$.

Step 4: In this step, the DC coefficient y_0 is adjusted to y'_0, to minimize the embedding distortion:

$$y'_0 = \underset{y\in\mathbb{R}}{\arg\min} \|\mathcal{H}_n^{-1}(y, y'_1, ..., y'_{2^n-1}) - \mathbf{x}\|_{l^\infty}, \tag{2}$$

where $y'_1, ..., y'_{2^n-1}$ are quantified AC coefficients determined in *Step 3*, and $\| \cdot \|_{l^\infty}$ is the usual l^∞-norm. Notice that, in the proposed method, $\mathcal{H}_n^{-1}(y'_0, y'_1, ..., y'_{2^n-1})$ corresponds to the NBP of marked characters. Then, by Eq. (2), the maximum modification (i.e., the amount of flipped pixels) to each character is minimized. Moreover, with the adjusted DC coefficient y'_0 determined in this step, we can prove that

$$\|\mathcal{H}_n^{-1}(y'_0, y'_1, ..., y'_{2^n-1}) - \mathbf{x}\|_{l^\infty} \leq \frac{Q}{2}. \qquad (3)$$

This important property guarantees that the distortion of each character can be controlled by the quantization step size. The choice of y'_0 (i.e., the solution to Eq. (2)) and the proof of Eq. (3) will be given later in appendix.

Step 5: Let $(x'_0, x'_1, ..., x'_{2^n-1}) = \mathcal{H}_n^{-1}(y'_0, y'_1, ..., y'_{2^n-1})$. The marked characters are obtained as follows:

1) If $x'_i > x_i$, according to the strategy introduced in subsection III-C, select $\lfloor x'_i \rfloor - x_i$ white SBPs from the i-th character, then flip these pixels. Here, $\lfloor \cdot \rfloor$ is the floor function.
2) Otherwise, select $x_i - \lfloor x'_i \rfloor$ black SBPs from the i-th character, then flip them.

In consequence, the i-th marked character contains exactly $\lfloor x'_i \rfloor$ black pixels. Notice that, if there are not enough flippable SBPs, we may first flip the selected SBPs and then repeat selecting SBPs from the modified character. Generally, twice selection is sufficient to get adequate flippable SBPs. After this step, $2^n - 1$ bits are embedded into the character group.

Step 6: Repeat *Step 2-5* for each character group, and the marked binary text image is obtained.

E. Watermark extraction

we now describe the watermark extraction procedure for paper-based text documents.

Step 1: First, the text document is scanned to get a gray-scale image. Then by thresholding, the gray-scale image is converted to binary. Here, the threshold for binarization is determined as follows. We find the two peaks of the scanned image's histogram, which are representative values of text content and background. Then the threshold is taken as the average of the two peaks (see Fig. 6).

Step 2: As *Step 1* of watermark embedding, the binary text image is first divided into characters. Then the characters are divided into groups such that each group contains 2^n characters.

Step 3: For a character group, let $\mathbf{x}' = (x'_0, x'_1, ..., x'_{2^n-1})$ be the NBP of its characters. Take $(y'_0, y'_1, ..., y'_{2^n-1}) = \mathcal{H}_n(\mathbf{x}')$. Then, extract watermark as follows, for each $i \in \{1, ..., 2^n - 1\}$:

$$w_i = \begin{cases} 0, & \text{if } [y'_i/Q] \text{ is even,} \\ 1, & \text{if } [y'_i/Q] \text{ is odd,} \end{cases}$$

where Q is the quantization step size, and $[x]$ means the nearest integer to x.

Step 4: Repeat *Step 3* for each character group. The watermark is obtained by linking extracted data bits from each character group.

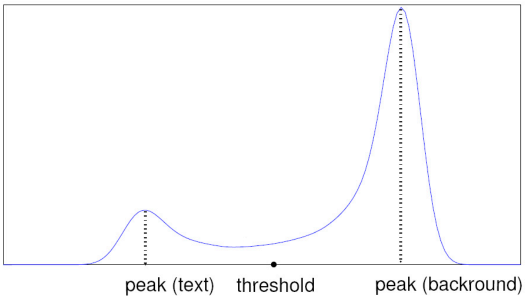


Fig. 6. Typical histogram of the scanned gray-scale image for paper-based text documents. The threshold for binarization is taken as the average of the two peaks.

F. Example of watermark embedding/extraction

Consider here a text image composed of four Chinese characters (12pt, "Kai Ti") showed in Fig. 7(a). We will embed three bits "101" into it by the proposed method with Hadamard transform $\mathcal{H}_2$ (i.e., $n = 2$) and quantization step size $Q = 600$. Notice that $\mathcal{H}_2$ maps $\mathbf{x} = (x_0, x_1, x_2, x_3)$ to $\mathbf{y} = (y_0, y_1, y_2, y_3)$ via

$$\begin{cases} y_0 = x_0 + x_1 + x_2 + x_3 \\ y_1 = x_0 - x_1 + x_2 - x_3 \\ y_2 = x_0 + x_1 - x_2 - x_3 \\ y_3 = x_0 - x_1 - x_2 + x_3 \end{cases}.$$

The Hadamard transform of the four characters' NBP, $\mathbf{x} = (2491, 1788, 1556, 1699)$, is $\mathbf{y} = (7534, 560, 1024, 846)$. To embed "101" into $\mathbf{x}$, the AC coefficients $y_1 = 560$, $y_2 = 1024$ and $y_3 = 846$ are first quantified to $y'_1 = 600$, $y'_2 = 1200$ and $y'_3 = 600$, respectively. We then adjust the DC coefficient. Regarding Eqs. (4)-(5), we see that $(\lambda_0, \lambda_1, \lambda_2, \lambda_3) = (7564, 7152, 7424, 7996)$ and $(y_{\max}, y_{\min}) = (7996, 7152)$. So, the adjusted DC coefficient is $y'_0 = (y_{\max} + y_{\min})/2 = 7574$. In this way, we get $\mathbf{x}' = \mathcal{H}_2^{-1}(y'_0, y'_1, y'_2, y'_3) = (2493.5, 1893.5, 1593.5, 1593.5)$ and $\lfloor \mathbf{x}' \rfloor - \mathbf{x} = (2, 105, 37, -106)$. Accordingly, we select respectively 2, 105 and 37 white SBPs from the first three characters and then flip them; for the last character, we select 106 black SBPs and then flip them. The marked text image is shown in Fig. 7(b), and an illustration of flipped pixels is presented in Fig. 7(e).

We now print the marked image by Kyocera KM-5035 and and scan the printed image by HP ScanJet 4890. The resolution of printer and scanner is set to 600 dpi. The scanned image is shown in Fig. 7(f). We then compute its histogram. Notice that the two peaks of this histogram are 39 and 255, the threshold for binarization is 147. After binarization, we get a binary image showed in Fig. 7(g). Counting the NBP of the four characters, we get $\mathbf{x}'' = (2747, 2073, 1725, 1752)$. Finally, by performing Hadamard transform, we get $\mathbf{y}'' = \mathcal{H}_2(\mathbf{x}'') = (8297, 647, 1343, 701)$, and thus the quantified AC coefficients are $([647/600], [1343/600], [701/600]) = (1, 2, 1)$. In this way, although the NBP are severely changed after PS, the embedded data "101" is correctly extracted.

Finally, the printed image is photocopied and scanned. The scanned image and its binarization are shown in Figs. 7(h)-7(i). We then compute the NBP and we get $(2841, 2044, 1757, 1813)$, whose Hadamard transform is $(8455, 741, 1315, 853)$. In this case, the quantified AC coefficients are $([741/600], [1315/600], [853/600]) = (1, 2, 1)$, and the watermark is also correctly extracted.

IV. Experimental results and discussions

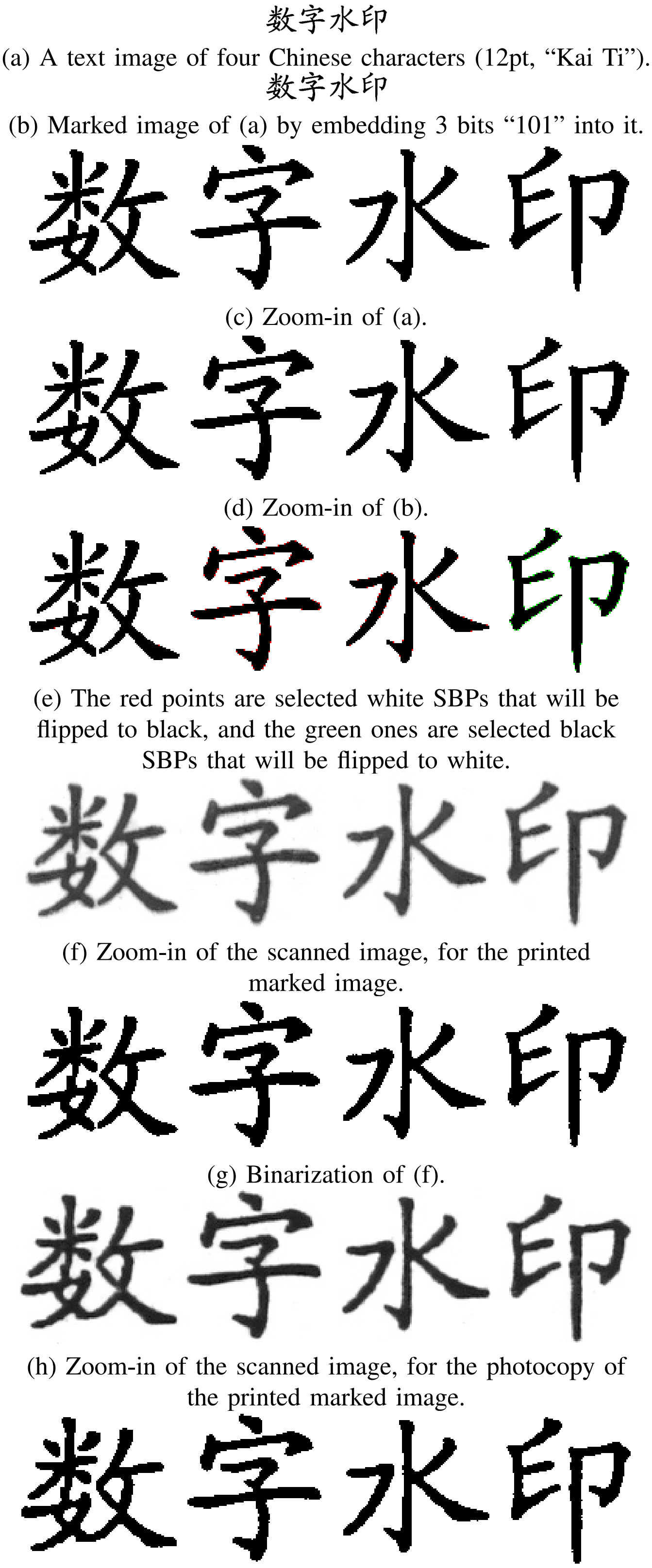
A. *Experiment with Chinese text document*

A Chinese text document containing 120 characters (12pt, "Kai Ti") is used in our experiment. The watermark is first embedded into the text image by the embedding algorithm described in subsection III-D with $n \in \{1, 2, 3\}$ and $Q = 600$. Notice that, the interpunctions in text image can be easily recognized by identifying their positions and NBP. Then the interpunctions can be ignored when dividing characters. Moreover, to enhance the robustness, the same watermark is repeatedly embedded five times, and majority voting is performed at watermark extraction. In this way, the embedding capacity is respectively 12 ($n = 1$), 18 ($n = 2$) and 21 ($n = 3$) bits. The original text image and marked ones are shown in Fig. 8. Clearly, the visual appearance of marked image becomes better with increasing n, and the quality degradation due to data embedding can not be easily perceived when $n > 1$.

The marked images are printed by five printers: Kyocera KM-5035, Canon iR5000, HP LaserJet 5100, Epson EPL-6200 and HP LaserJet 1200. Then the printed documents are photocopied several times by Kyocera KM-5035. Here, the $(i + 1)$-th photocopy means photocopied version of the i-th photocopy. After that, printed and photocopied documents are scanned by HP ScanJet 4890, to restore them as gray-scale images. Here, the resolution of printer and scanner is set to 600 dpi. Finally, watermark detection is performed on these images according to the extraction algorithm described in subsection III-E. The detection results are presented in Table II, where "success" means that all watermark bits are successfully extracted and "fail" means that at least one watermark bit is incorrectly extracted. From this table, we see that the proposed method can resist the print and 2nd-photocopy attack except the case $n = 3$ with HP LaserJet 1200, which is an ultra-potable printer and has been used six years. Moreover, the proposed method can get the best performance with $n = 2$. In this way, we argue that $n = 2$ is a promising choice, balancing the visual appearance, embedding capacity and robustness.

B. *Experiment with English text document*

In this experiment, we use an English text document containing 80 words (12pt, "Time New Roman"). The proposed method is implemented with $n = 2$ and $Q = 1500$, and a 12 bits watermark is repeatedly embedded five times. The original text image and the marked ones are shown in Fig. 9, and the watermark detection results are presented in Tab. III. We see that compared with Chinese, the proposed method performs



(a) A text image of four Chinese characters (12pt, "Kai Ti").

(b) Marked image of (a) by embedding 3 bits "101" into it.

(c) Zoom-in of (a).

(d) Zoom-in of (b).

(e) The red points are selected white SBPs that will be flipped to black, and the green ones are selected black SBPs that will be flipped to white.

(f) Zoom-in of the scanned image, for the printed marked image.

(g) Binarization of (f).

(h) Zoom-in of the scanned image, for the photocopy of the printed marked image.

(i) Binarization of (h).

Fig. 7. Example of the proposed watermarking method.

worse on English. It can resist the photocopy attack only once. The reason is that English words have variable length and contain very different amount of flippable SBPs. As a result, some short words such as "to","of", "by" and "as", do not have adequate flippable SBPs.

C. Discussions on practical implementation

We now discuss issues related to the practical implementation of the proposed method.

The character-dividing step is very important, since the watermark synchronization can not be guaranteed without a correct character-dividing. Indeed, it is rather difficult for scanned text image. According to our experience, most failed detections are due to the incorrect character-dividing. We just mention here that, by using OCR technique, one may improve the character-dividing accuracy. However, this improvement is limited, and performing OCR is time-consuming.

For the purpose of copy tracking, it is serious if one extracts a wrong watermark. Then, an error-detection code such as CRC-16 or CRC-32 should be used and embedded into text image as a part of watermark. If using an error-detection code, the threshold for binarizing is not important, since we may exhaustively search thresholds that can pass the redundancy check. In this way, the detection performance can be significantly improved. We now present some experimental results. By using CRC-32 and threshold-searching, the proposed method with $n = 2$ is tested for hundreds of A4 sized office-like text documents in different languages (Chinese, Korean, Japanese, English and French), typefaces and font-sizes (10pt-14pt). For most cases, the method can resist 2nd-photocopy attack (7th-photocopy, in the best case) for eastern languages, and 1st-photocopy attack (4th-photocopy, in the best case) for western languages. Here, the watermark is repeatedly embedded five times. The embedding capacity is approximately 100 bits (excluding the 32 bits CRC codes) for an eastern language text, and 40 bits for a western language text. So, the incorporation of error-detection code with threshold-searching is favorable for our method in terms of enhancing the practicability and improving the watermark detection performance.

Finally, we remark that the performance of the proposed method is heavily dependent on the status of printer/copier/scanner. Even for the same model of machine, the detection performance may be different according to the wear condition, ink quality, paper quality and many other uncontrollable factors. A more effective preprocessing stage may be helpful to this issue.

V. Conclusion

In this paper, a novel text watermarking algorithm is proposed. The method can embed adequate data into office-like text document without changing its visual appearance, and more importantly, it can resist print-and-photocopy attack. With these advantages, this method can be used for copy tracking of paper-based documents, which can improve the management of confidential files. However, the proposed method is ineffective for an incomplete document or a low resolution image such as the copy derived from cell phone or digital camera. All these issues will be investigated in future.

Appendix: Solution to Eq. (2) and proof of Eq. (3)

We first solve Eq. (2). Here, we use the notations introduced in subsection III-D. Assume

$$(z_0, z_1, ..., z_{2^n-1}) = \mathcal{H}_n^{-1}(y, y_1', ..., y_{2^n-1}'),$$

where y is the real number to be determined. Notice that the Hadamard matrix can be written as

$$\mathbf{H}_n = \begin{pmatrix} 1 & h_{0,1} & \cdots & h_{0,2^n-1} \\ 1 & h_{1,1} & \cdots & h_{1,2^n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & h_{2^n-1,1} & \cdots & h_{2^n-1,2^n-1} \end{pmatrix}.$$

Then, for each $i \in \{0, 1, ..., 2^n - 1\}$,

$$z_i = \frac{y + \sum_{j=1}^{2^n-1} h_{i,j} y_j'}{2^n}.$$

So,

$$z_i - x_i = \frac{y - \lambda_i}{2^n},$$

where

$$\lambda_i = y_0 - \sum_{j=1}^{2^n-1} h_{i,j}(y_j' - y_j) \tag{4}$$

is a constant. In this way, to minimize

$$\|\mathcal{H}_n^{-1}(y, y_1', ..., y_{2^n-1}') - \mathbf{x}\|_{l^\infty} = \max_{0 \le i \le 2^n-1} |z_i - x_i|,$$

one must have $y = (y_{\max} + y_{\min})/2$, where

$$y_{\max} = \max_{0 \le i \le 2^n-1} \lambda_i \quad \text{and} \quad y_{\min} = \min_{0 \le i \le 2^n-1} \lambda_i. \tag{5}$$

Consequently, the adjusted DC coefficient y_0' is $(y_{\max} + y_{\min})/2$.

We now prove Eq. (3). Clearly, by definitions of $y_{\max}$, $y_{\min}$ and y_0', for each $i \in \{0, 1, ..., 2^n - 1\}$, we have

$$|y_0' - \lambda_i| \le \frac{y_{\max} - y_{\min}}{2}. \tag{6}$$

In addition, by the definition of y_i' in Eq. (1), we know that $|y_i' - y_i| \le Q$. So, for $i_1 \ne i_2$,

$$|\lambda_{i_1} - \lambda_{i_2}| \le Q \sum_{j=1}^{2^n-1} |h_{i_1,j} - h_{i_1,j}| = 2^n Q. \tag{7}$$

This inequality uses a property of Hadamard matrix:

$$\sum_{j=1}^{2^n-1} |h_{i_1,j} - h_{i_1,j}| = 2^n.$$

Then, according to Eq. (7), we see that

$$y_{\max} - y_{\min} \le 2^n Q.$$

Incorporating this inequality with Eq. (6), we have

$$|x_i' - x_i| = \frac{|y_0' - \lambda_i|}{2^n} \le \frac{y_{\max} - y_{\min}}{2^{n+1}} \le \frac{Q}{2},$$

where $(x_0', x_1', ..., x_{2^n-1}') = \mathcal{H}_n^{-1}(y_0', y_1', ..., y_{2^n-1}')$. This completes the proof of Eq. (3).

TABLE II
WATERMARK DETECTION RESULTS FOR THE CHINESE TEXT DOCUMENT SHOWED IN FIG. 8.

	printer	printed document	1st-photocopy	2nd-photocopy	3rd-photocopy	4th-photocopy	5th-photocopy
$n = 1$	Kyocera KM-5035	success	success	success	success	fail	fail
	Canon iR5000	success	success	success	success	fail	fail
	HP LaserJet 5100	success	success	success	success	fail	fail
	Epson EPL-6200	success	success	success	success	fail	fail
	HP LaserJet 1200	success	success	success	fail	fail	fail
$n = 2$	Kyocera KM-5035	success	success	success	success	success	fail
	Canon iR5000	success	success	success	success	success	fail
	HP LaserJet 5100	success	success	success	success	fail	fail
	Epson EPL-6200	success	success	success	success	fail	fail
	HP LaserJet 1200	success	success	success	fail	fail	fail
$n = 3$	Kyocera KM-5035	success	success	success	fail	fail	fail
	Canon iR5000	success	success	success	success	fail	fail
	HP LaserJet 5100	success	success	success	fail	fail	fail
	Epson EPL-6200	success	success	success	fail	fail	fail
	HP LaserJet 1200	success	success	fail	fail	fail	fail

TABLE III
WATERMARK DETECTION RESULTS FOR THE ENGLISH TEXT DOCUMENT SHOWED IN FIG. 9.

printer	printed document	1st-photocopy	2nd-photocopy	3rd-photocopy	4th-photocopy
Kyocera KM-5035	success	success	success	success	fail
Canon iR5000	success	success	success	success	fail
HP LaserJet 5100	success	success	success	fail	fail
Epson EPL-6200	success	success	fail	fail	fail
HP LaserJet 1200	success	success	fail	fail	fail

REFERENCES

[1] J. T. Brassil, S. Low, and N. F. Maxemchuk, "Copyright protection for the electronic distribution of text documents," *Proc. of IEEE*, vol. 87, no. 7, pp. 1181–1196, July 1999.

[2] D. Huang and H. Yan, "Interword distance changes represented by sine waves for watermarking text images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 12, pp. 1237–1245, December 2001.

[3] H. Yang and A. C. Kot, "Text document authentication by integrating inter character and word spaces watermarking," in *Proc. the IEEE ICME*, 2004, pp. 955–958.

[4] T. Amano and D. Misaki, "A feature calibration method for watermarking of document images," in *Proc. of ICDAR*, 1999, pp. 91–94.

[5] D. Liu, "Hidden data communication method and the application thereof in text digital watermark technology," 2005, PCT International Patent: PCT/CN2005/001703.

[6] M. Suzaki and M. Suto, "A watermark embedding and extracting method for printed documents," *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, vol. 88, no. 7, pp. 43–51, July 2005.

[7] Y. Takahashi, T. Yamada, R. Ebisawa, Y. Fujii, and S. Tezuka, "Information embedding method for home printing of certifications," in *Proc. of ICACT*, vol. 3, 2008, pp. 2116–2120.

[8] N. B. Puhan, A. T. S. Ho, and F. Sattar, "High capacity data hiding in binary document images," in *Proc. of IWDW*, ser. LNCS, vol. 5703, 2009, pp. 149–161.

[9] M. J. Atallah, V. Raskin, C. F. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K. E. Triezenberg, "Natural language watermarking and tamperproofing," in *Proc. of IH*, ser. LNCS, vol. 2578, 2002, pp. 196–212.

[10] M. Topkara, C. M. Taskiran, and E. J. Delp, "Natural language watermarking," in *Security, Steganography, and Watermarking of Multimedia Contents VII*, ser. Proc. SPIE, vol. 5681, 2005, pp. 441–452.

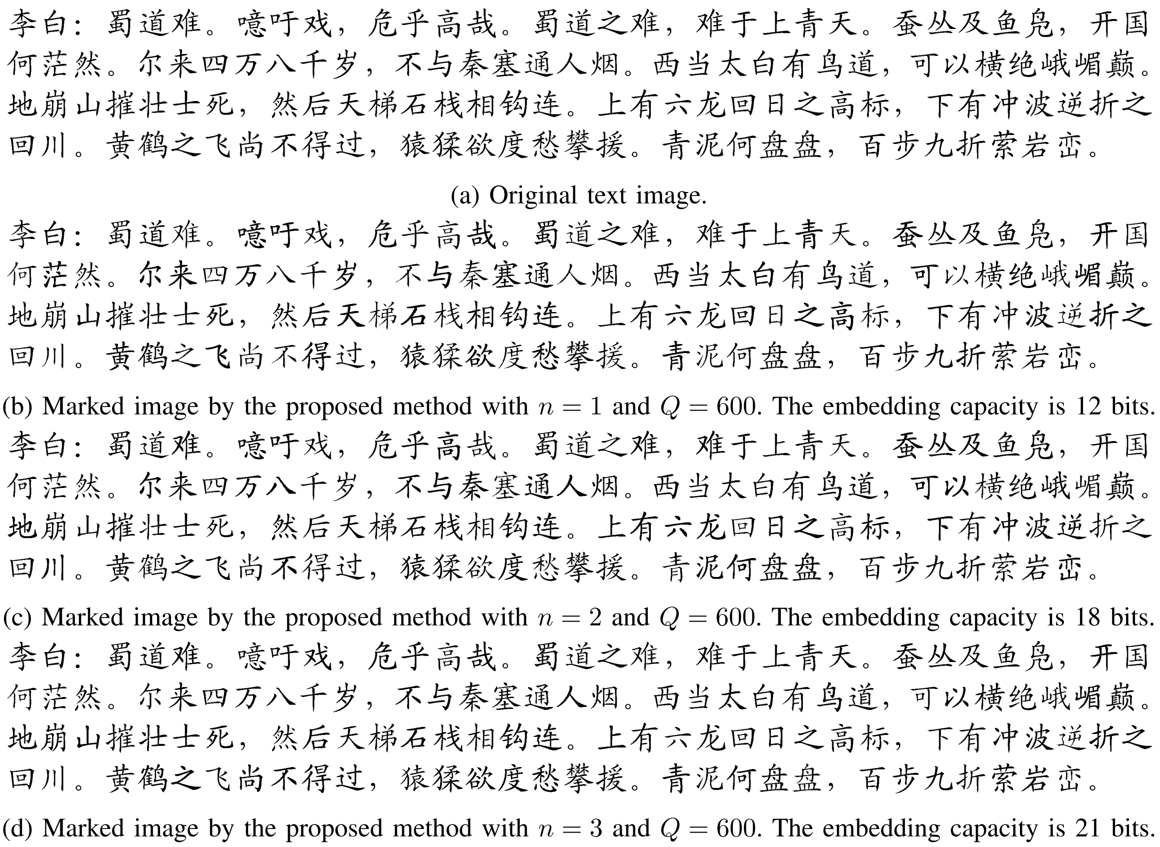
[11] Y. Liu, X. Sun, C. Gan, and H. Wang, "An efficient linguistic steganography for chinese text," in *Proc. the IEEE ICME*, 2007, pp. 2094–2097.

[12] H. M. Meral, B. Sankur, A. S. Ozsoy, T. Gungor, and E. Sevinc, "Natural language watermarking via morphosyntactic alterations," *Computer Speech & Language*, vol. 23, no. 1, pp. 107–125, January 2009.

[13] M. Wu and B. Liu, "Data hiding in binary image for authentication and annotation," *IEEE Trans. Multimedia*, vol. 6, no. 4, pp. 528–538, August 2004.

[14] H. Yang and A. C. Kot, "Pattern-based data hiding for binary image authentication by connectivity-preserving," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 475–486, April 2007.

[15] H. Yang, A. C. Kot, and S. Rahardja, "Orthogonal data embedding for binary images in morphological transform domain - A high capacity approach," *IEEE Trans. Multimedia*, vol. 10, no. 3, pp. 339–351, April 2008.



李白：蜀道难。噫吁戏，危乎高哉。蜀道之难，难于上青天。蚕丛及鱼凫，开国
何茫然。尔来四万八千岁，不与秦塞通人烟。西当太白有鸟道，可以横绝峨嵋巅。
地崩山摧壮士死，然后天梯石栈相钩连。上有六龙回日之高标，下有冲波逆折之
回川。黄鹤之飞尚不得过，猿猱欲度愁攀援。青泥何盘盘，百步九折萦岩峦。

(a) Original text image.

李白：蜀道难。噫吁戏，危乎高哉。蜀道之难，难于上青天。蚕丛及鱼凫，开国
何茫然。尔来四万八千岁，不与秦塞通人烟。西当太白有鸟道，可以横绝峨嵋巅。
地崩山摧壮士死，然后天梯石栈相钩连。上有六龙回日之高标，下有冲波逆折之
回川。黄鹤之飞尚不得过，猿猱欲度愁攀援。青泥何盘盘，百步九折萦岩峦。

(b) Marked image by the proposed method with $n = 1$ and $Q = 600$. The embedding capacity is 12 bits.

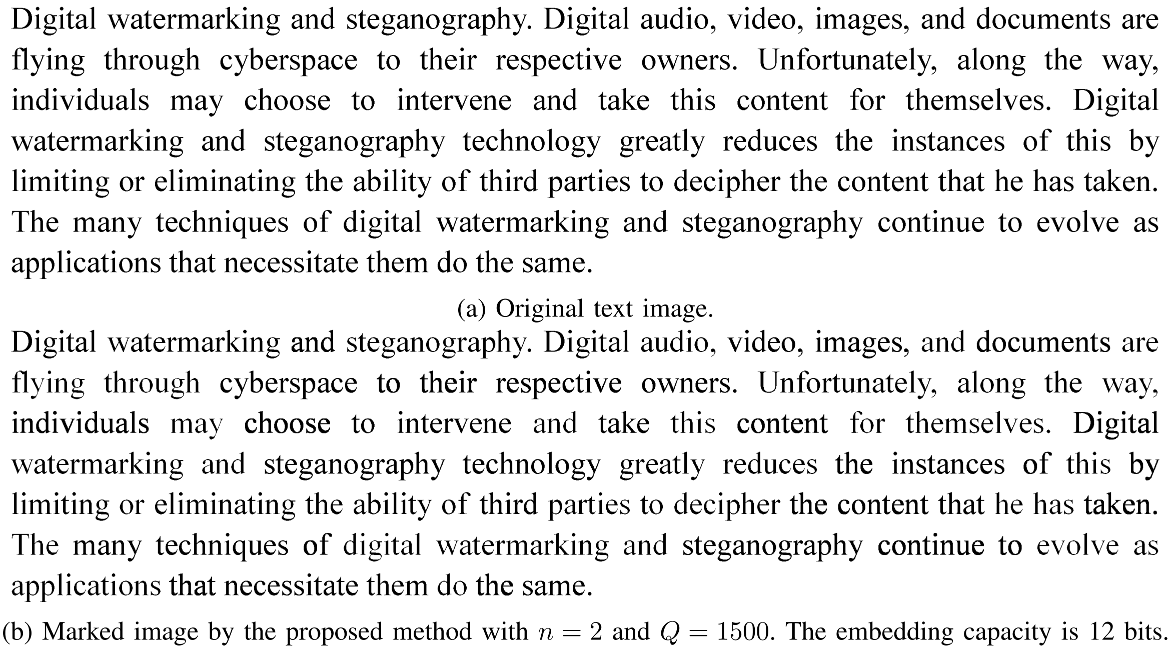
李白：蜀道难。噫吁戏，危乎高哉。蜀道之难，难于上青天。蚕丛及鱼凫，开国
何茫然。尔来四万八千岁，不与秦塞通人烟。西当太白有鸟道，可以横绝峨嵋巅。
地崩山摧壮士死，然后天梯石栈相钩连。上有六龙回日之高标，下有冲波逆折之
回川。黄鹤之飞尚不得过，猿猱欲度愁攀援。青泥何盘盘，百步九折萦岩峦。

(c) Marked image by the proposed method with $n = 2$ and $Q = 600$. The embedding capacity is 18 bits.

李白：蜀道难。噫吁戏，危乎高哉。蜀道之难，难于上青天。蚕丛及鱼凫，开国
何茫然。尔来四万八千岁，不与秦塞通人烟。西当太白有鸟道，可以横绝峨嵋巅。
地崩山摧壮士死，然后天梯石栈相钩连。上有六龙回日之高标，下有冲波逆折之
回川。黄鹤之飞尚不得过，猿猱欲度愁攀援。青泥何盘盘，百步九折萦岩峦。

(d) Marked image by the proposed method with $n = 3$ and $Q = 600$. The embedding capacity is 21 bits.

Fig. 8. Original text image and its marked versions, for a Chinese text document containing 120 characters (12pt, “Kai Ti”).



Digital watermarking and steganography. Digital audio, video, images, and documents are flying through cyberspace to their respective owners. Unfortunately, along the way, individuals may choose to intervene and take this content for themselves. Digital watermarking and steganography technology greatly reduces the instances of this by limiting or eliminating the ability of third parties to decipher the content that he has taken. The many techniques of digital watermarking and steganography continue to evolve as applications that necessitate them do the same.

(a) Original text image.

Digital watermarking and steganography. Digital audio, video, images, and documents are flying through cyberspace to their respective owners. Unfortunately, along the way, individuals may choose to intervene and take this content for themselves. Digital watermarking and steganography technology greatly reduces the instances of this by limiting or eliminating the ability of third parties to decipher the content that he has taken. The many techniques of digital watermarking and steganography continue to evolve as applications that necessitate them do the same.

(b) Marked image by the proposed method with $n = 2$ and $Q = 1500$. The embedding capacity is 12 bits.

Fig. 9. Original text image and its marked versions, for an English text document containing 80 words (12pt, “Time New Roman”).