Signal Processing 203 (2023) 108791

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Language universal font watermarking with multiple cross-media robustness

Xi Yang, Weiming Zhang*, Han Fang, Zehua Ma, Nenghai Yu

School of Information Science and Technology, University of Science and Technology of China, Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences, China

ARTICLE INFO

Article history:
Received 16 June 2021
Revised 30 March 2022
Accepted 20 September 2022
Available online 22 September 2022

Keywords:
Robust watermarking
Document protection
Multimedia forensics

ABSTRACT

The rapid development of digital devices and the increasing demand for copyright protection and leakage tracing of documents result in new demands for robust text watermarking. It becomes more important for a utility text watermarking scheme to have robustness in cross-media channels since the camera shooting operation to papers or screens has become very convenient. To achieve the corresponding cross-media robustness, this paper proposes a font-based text watermarking scheme, which is applicable to most commonly used languages while the previous works were usually designed for specific languages. To generate fonts with high efficiency, we propose a novel glyph centroid modification-based font generation algorithm, which can automatically create target-similar fonts. And we design an effective watermarking scheme that utilizes the relative centroid position (RCP) of glyphs to represent watermark signals. Compared with existing font-based text watermarking schemes that artificially design fonts, the proposed scheme ensures higher efficiency by generating the font codebook automatically. In addition, the proposed RCP-based watermarking scheme can achieve the robustness against lossy compression and several kinds of cross-media distortions. The watermark message can be extracted without the need to recognize the semantic information of glyphs by optical character recognition (OCR).

1. Introduction

Digital watermarking, as an effective means to protect copyright and trace the source of leakage, has already been well studied. There have been many excellent works in the fields of image [1–5], audio [6–9], and video [10,11] watermarking. But for text watermarking [12], it is still in its infancy. However, text documents are the most commonly used information carrier in daily life, so the source tracing and copyright protection of text documents are becoming an essential demand. Traditional watermarking schemes are more concerned with the robustness to digital editing since transmission mostly occurs in the digital domain. However, with the rapid development of digital devices, recording text documents has become increasingly convenient. By taking a photo or scanning through the scanner, the text content can be recorded in a high-quality and efficient way, which causes new risks of information leakage. As a result, the copyright of text documents should be carefully protected when such cross-media processes occur, which requires the watermarking scheme to guarantee robustness in cross-media processes.

Furthermore, most traditional text watermarking algorithms are developed for specific languages, and cannot be applied to other languages. However, with the fast pace of internationalization today, an increasing number of documents are mixed with different languages, which greatly influences the actual performance of such algorithms. Therefore, developing a language-independent text watermarking scheme is an urgent demand.

In summary, to meet the requirements today, a good text watermarking algorithm should satisfy three important properties: cross-media robustness, language universality, and high visual quality.

However, to the best of our knowledge, none of the existing schemes can fulfill these requirements at the same time. Traditional text watermarking schemes can be broadly divided into four categories: the format-based schemes, linguistic-based schemes, image-based schemes, and font-based schemes. For the first category, Maxemchuk and Low [13], Brassil et al. [14] first proposed an English language text watermarking algorithm by slightly shifting words or sentences horizontally or vertically in the original document. Since the original document is needed to detect the translation of words or sentences, this type of watermarking algorithm belongs to non-blind watermarking. Additionally, the payload of this method is limited, and as the amplitude of the translation is

* Corresponding author.
E-mail address: zhangwm@ustc.edu.cn (W. Zhang).

https://doi.org/10.1016/j.sigpro.2022.108791
0165-1684/

Table 1
The comparison of different text watermarking schemes from different aspects.

Scheme	Print-scan	Print-camera	Screen-camera	JPEG	Scaling	Low-resolution screenshot	Integrity	Language-universality
Format-based [13,14,23–25]	✓	×	×	×	×	✓	✓	✓
Linguistic-based [15,16,26]	✓	✓	✓	✓	✓	✓	×	×
Image-based [18–20,27,28]	✓	×	×	✓	✓	×	✓	×
Font-based [21,22]	✓	✓	×	✓	×	×	✓	×
Ours	✓	✓	✓	✓	✓	✓	✓	✓

tiny, the watermark can be easily destroyed in cross-media processes.

For the linguistic-based watermarking algorithms such as [15,16], the common method is using natural language processing (NLP) methods to replace certain words or sentences in the document to hide information. Modifications in the semantic dimension make these methods highly robust to most cross-media distortions. However, because these methods need to replace the text content and cannot retain the semantic information, they cannot be applied to scenarios where the integrity of the original sentences needs to be completely assured, such as commercial contracts, government documents, and academic papers. Therefore, these methods are more suitable for text steganography [17].

In addition, image-based watermarking algorithms such as [18–20] can also be used for document images, which disguise the watermark information as a background image with colors and patterns visible to the human eye and then superimpose it with the document image. However, such textures or under-paintings are also not allowed in many practical document application scenarios. Meanwhile, the watermark information will be destroyed in the region covered by text, and can only be retained in the line spacing region. That is, the larger the occupied text region and the closer the line spacing of the document, the lower the watermark capacity and extraction accuracy of such schemes will be. Therefore, is it possible to embed watermark information directly in the text region?

Recently, algorithms based on font modification have been proposed, which are more informative and robust than the above-mentioned methods. Qi et al. [21] proposed to embed watermark information by modifying the stroke position of glyphs to generate different deformations of the same glyph and encoding these deformations as templates. Then they used a template matching algorithm for extraction. However, since the glyphs will be down-sampled into dot-map images when displayed on a screen, the differences between the deformations will become fewer. This can disturb the template matching process. Therefore, it can only be used for paper-based documents that use vector format glyphs and is not adapted to screen-based documents. In addition, this method is more suitable for languages with complex glyph structures such as Chinese, as the changes to the strokes are less noticeable. However, when used for languages with simple structures such as English, the visual quality will be poor.

Xiao et al. [22] designed an English font codebook to represent the watermark information. On the extraction side, they trained a classification network for each letter to extract the information. However, such a method can only be successful when the font-size is larger than 200 px, which is not practical in common documents. Additionally, since the extraction needs to train a classification network for each English letter, it is not applicable in ideogram scenarios with a wide range of glyphs such as Chinese and Japanese.

Moreover, these two font-based text watermarking methods all require the participation of OCR to recognize the semantic information of the glyphs before extracting watermark signals. This means that their performance is closely related to the accuracy of OCR.

The advantages and disadvantages of the aforementioned text watermarking algorithms are summarized in Table 1.

To satisfy cross-media robustness, language universality, and high visual quality simultaneously, we propose a language universal font watermarking scheme with multiple cross-media robustness and effectively generate target-similar fonts for watermark embedding and extracting. Specifically, we first develop a font codebook generation algorithm based on glyph perturbation. By automatically modifying the glyphs in the target font to make their centroids shift, several target-similar fonts that can be used to represent watermark signals are generated. Then, we propose to use the relative centroid position (RCP) of each two neighboring glyphs in the text region to represent the watermark signal. Since we only modify the text content at the font dimension, the integrity of the original text at the semantic dimension can be completely preserved, which makes our method more applicable in common document usage scenarios compared with existing methods. In addition, by utilizing the stable and global feature RCP to represent watermark information, the strong robustness to cross-media distortions can be satisfied and there is no need for our method to recognize the semantic information of glyphs in watermark extraction.

The contributions of this paper can be summarized as follows:

- We propose to embed the watermark signal with an RCP comparison method, based on which, robustness against lossy compression, screen-camera shooting, print-scanning, and print-camera shooting processes can be well achieved.
- We propose a glyph centroid modification based automatic font codebook generation algorithm, which can effectively create target-similar fonts with high visual quality and language universality.
- There is no requirement for the participation of OCR to identify the glyphs when extracting the watermark, which greatly improves the extraction accuracy and efficiency of our scheme.
- Extensive experiments with different attacking conditions indicate the outstanding performance of the proposed scheme.

The rest of this paper is organized as follows. In Section 2, we analyze the cross-media distortions and the characteristics of the glyph centroid position. In Section 3 we propose the centroid modification based font codebook generation scheme. Section 4 and 5 illustrate the watermark embedding and the extraction process with the RCP comparison algorithm. The experimental results are shown and discussed in Section 6. Section 7 draws the conclusion.

2. Preliminaries

2.1. Cross-Media distortion analysis

To design a robust text watermarking scheme, we should first analyze the possible distortions in document transmission. In modern society, the text is mainly carried on printed papers and electronic screens. Therefore, we mainly consider distortions in print-

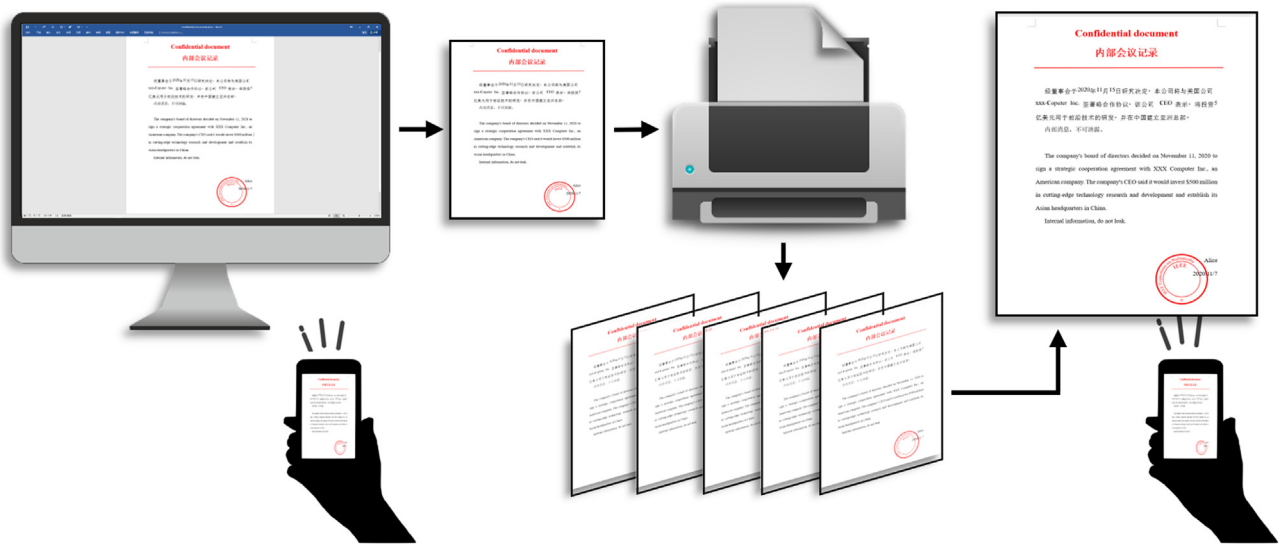


Fig. 1. Typical cross-media processes of text documents (e.g., screen-camera shooting, print-scanning, and print-camera shooting).

camera shooting, print-scanning, screenshot, and screen-camera shooting processes, as Fig. 1 shows.

In the case of print-camera shooting and print scanning, the distortions come from not only the printer or scanner, but also paper deformation (i.e., folded, curved, or crumpled paper). Furthermore, complex illumination conditions will also affect the quality of the captured document image [29].

For the screen-shooting distortions, Fang et al. [3] summarized them into four aspects: display distortion, lens distortion, sensor distortion, and processing distortion. Display distortion refers to format transformation, which is caused by the limitation of screen resolution. Since the screen will down-sample the vector graphics format glyphs into dot-map glyphs with different sizes, the actual glyph displayed on the screen will lose some information compared to its original vector graphics format. During the screen-shooting process, different shooting conditions will result in different lens distortions, light source distortions, and moirE pattern distortions.

To deal with cross-media distortions, previous image watermarking schemes [3,27,30,31] tried to design patterns that can survive these distortions. However, such methods will cause serious visual distortions in text documents since the colours and textures of the document images are quite simple. The key that such methods can be applied in image watermarking schemes is that the structure modification arising from the patterns is robust, so we should find a structure in text documents to represent the watermark to achieve robustness. Fortunately, we found that the glyph itself, as a highly encoded textured image, can be well retained in these cross-media processes. Therefore, the key is how to generate different fonts with high visual similarity to represent watermark signals.

2.2. Glyph centroid analysis

Regarding the glyph as a binary image, the modifications to it will result in the increase and decrease of the black pixels, which further influence the distribution of the pixels. Meanwhile, different distributions of the black pixels will result in different centroid locations. In other words, we can utilize the different centroid distributions of the glyphs to represent different fonts that will appear similar when seen by the human eye. The definition and analysis of the glyph centroid are as follows:

(a) The original size of 'Aa'. (b) The corresponding size after processing.

Fig. 2. The normalization before calculating the centroid.

2.2.1. The definition of glyph centroid

The centroid of an image is also called the center of mass of an image. For gray-scale images, the pixel value is a single number that represents the brightness of the pixel, which we use to represent the mass. The most typical pixel format is the byte image, where this number is stored as an 8-bit integer giving a range of possible values from 0 (black) to 255 (white). We regard a 2-D glyph image as a two-dimensional matrix $\boldsymbol{I}$. $\boldsymbol{I}(i, j)$ is the pixel value corresponding to point (i, j) of $\boldsymbol{I}$. As the glyph part of an image is usually black pixels, we define the mass of pixel (i, j) as

$$m(i, j) = 255 - I(i, j). \tag{1}$$

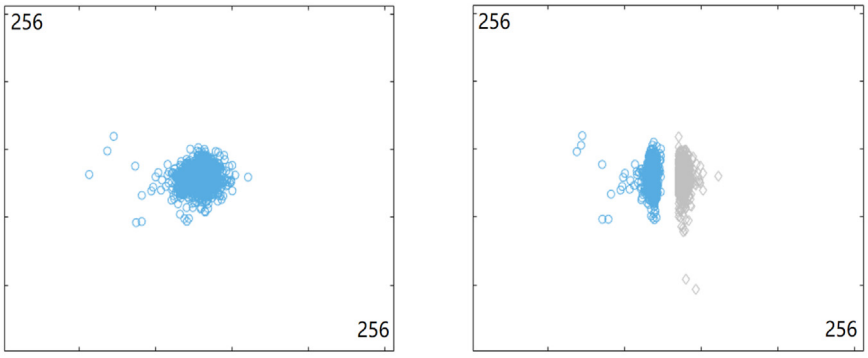
Then its centroid location is the point that can equally divide $\boldsymbol{I}$ both in $\boldsymbol{x}$-axis and $\boldsymbol{y}$-axis, which can be formulated by:

$$(x, y) = \left(\frac{\sum_{i=1}^{M} \sum_{j=1}^{N} m(i, j) \cdot i}{\sum_{i=1}^{M} \sum_{j=1}^{N} m(i, j)}, \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} m(i, j) \cdot j}{\sum_{i=1}^{M} \sum_{j=1}^{N} m(i, j)} \right), \tag{2}$$

where $m(i, j)$ is the mass corresponding to the pixel point (i, j). (x, y) is the centroid coordinate of the two-dimensional image, and M, N are the length and width of the image.

2.2.2. The property of the glyph centroid

In natural language, the size of different glyphs varies greatly, as shown in Fig. 2(a). To compare the relationship of centroids from different glyphs under the same coordinate system and make the centroids shift more with less glyph modification, we designed a unique character normalization algorithm in Section 5.2. In other words, we use the smallest circumscribed rectangle of the glyph to segment it and then resize the different circumscribed rectangles to the same scale. Fig. 2(b) shows the relative size of Fig. 2(a)



(a) The original centroids distribution.

(b) The centroids distribution of two fonts after fine-tuning.

Fig. 3. The distribution of centroids from 3000 glyphs including Chinese, English, and Arabic numerals.

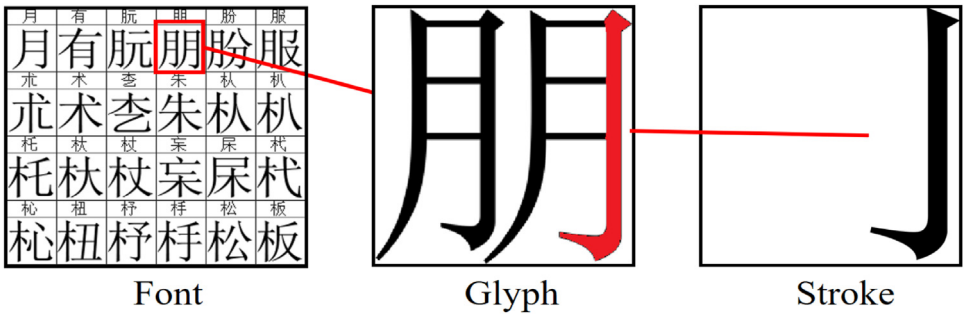


Fig. 4. Example of a font, a glyph and a stroke.

after normalization. Obviously, after normalization, the relation of different glyphs' centroids is easier to compare.

Using the above-mentioned normalization method, we analyzed the centroid distribution of Chinese, English, and Arabic numerals. We found that except for some very asymmetric glyphs (such as 'L', 'j'), the centroid coordinates of Chinese and English glyphs are commonly distributed near the center of the given coordinate system as shown in Fig. 3(a). In other words, we can take the centerline of the coordinate system as the dividing line, and make the centroid of a glyph shift to a predetermined direction by fine-tuning. Figure 3(b) shows the two centroid distributions after we fine-tuned the original glyphs horizontally. Then, we obtain two fonts with different centroid distributions, one of which has centroids all located on the left side of $x = 128$, and the other one is the opposite.

3. Target-Similar font codebook generation

Unlike common font generation tasks [32,33] that generate non-existent fonts with different styles from existing fonts, we slightly modify the glyphs of a given font to make their centroids shift and keep them similar to the original glyphs simultaneously. It should be noted that there is a distinction between the concepts of 'font', 'glyph', and 'stroke', where a font (e.g., Times New Roman, SimSun, etc.) is a collection of many different glyphs (e.g., 'A', '8', etc.) and a glyph is made up of strokes, as Fig. 4 shows.

Will subtle changes in glyphs affect people's reading quality? According to Rayner's study [34], the average time human eyes spend reading one word (glancing and gazing) is 225 ms, which makes it harder for human eyes to detect changes in a glyph's outline. In addition, according to Visual Psychology [35,36], the recognition of characters also requires the cognitive process of the brain. Character recognition is a process of matching the information from visual stimuli to memory information, and this process has a high degree of flexibility, making it different from images and videos that depend more on visual perception. Even if visual stimuli vary in size, orientation, and some small details, people can always identify these stimuli as different examples of familiar patterns. In the process of character semantic recognition, people have a good automatic adjustment function for the changes in glyphs. Therefore, subtle changes in glyphs will not affect the reading quality of the document. We also test the visual quality of the modified glyphs in Section 6.4.

Based on the above analysis and inspired by the traditional font design methods, we design the following two methods as the basic operations to modify the glyphs.

***Method* 1**: The increments or decrements of black pixels at the edge of a stroke can bring the centroid closer or farther away. We use δ_1 to represent this basic operation.

***Method* 2**: To move the centroid, the strokes of the glyph can be moved in the predetermined direction. We take this operation as δ_2.

With these basic operations, we can modify a glyph to move its centroid in the predetermined direction. Given the original font F_0, in the common file format (.ttf), which includes glyphs $F_0(G_k)$, $(k = 0, 1, \ldots, K)$ and each glyph is a vector graphic constructed from Bezier curves. We modify each glyph in the form of Bezier Curves to move its centroid in a predetermined direction and generate the new font F_i, $(i = 1, 2, \ldots, N)$ with new glyphs $F_i(G_k)$. Note that the glyphs in different generated fonts have identical semantics but different centroid locations. For example, when $N = 2$, the coordinate system is divided into two regions by $x = 128$, as shown in Fig. 3(b). We modify each glyph in F_0 to generate F_1 (i.e., blue circles) and F_2 (i.e., grey diamonds). The generation process can be

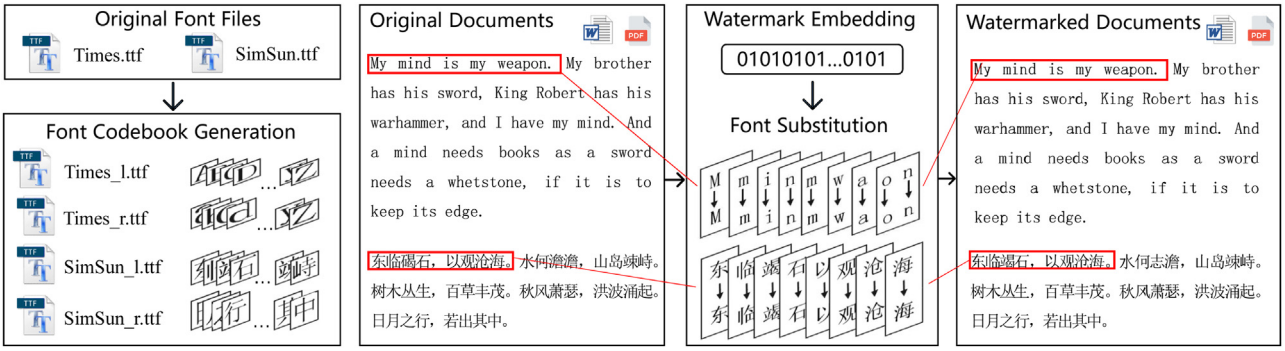


Fig. 5. The font codebook generation and watermark embedding process.

represented by

$$F_i(G_k) = \mathcal{M}(F_0(G_k), \Delta_{ik}), \tag{3}$$

where $\mathcal{M}(\cdot)$ represents the modification process to $F_0(G_k)$, and Δ_{ik} denotes the modification operations (i.e., combinations of δ_1 and δ_2) on $F_0(G_k)$ to generate $F_i(G_k)$.

So, how can we find the Δ_{ik} that ensures both the visual quality and robustness? Let C_{ik} represent the centroid of $F_i(G_k)$ and C_0 be the origin of the coordinate system. We formulate the modification process in Eq. (3) as a maximization problem and iteratively conduct the operation as follows:

$$\begin{aligned} &\arg\max_{\Delta_{ik}} \ NCC(F_0(G_k), F_i(G_k)) \\ &\text{subject to } D(C_{ik}, C_0) \ \geq \ \tfrac{r}{2}, \end{aligned} \tag{4}$$

where $NCC(\cdot)$ is the normalized cross-correlation function [37] that we use to evaluate the structural similarity of the generated font to the original font. It ranges from 0 (no similarity) to 1 (identical font). $D(C_{ik}, C_0)$ is the Euclidean distance of the centroid of $F_i(G_k)$ and the center of the coordinate system. r is the embedding strength. We choose $r = 8$ in our experiments to ensure robustness and we will discuss the impact of different values of r on robustness in Section 6.5. To facilitate understanding, we provide the pseudo-code of the font generation process when $N = 2$ in Algorithm 1. For practical applications, the corresponding settings can be changed according to the *direction* and N. In this way, different glyphs in different fonts will be deformed in the correct direction, as Fig. 3 shows.

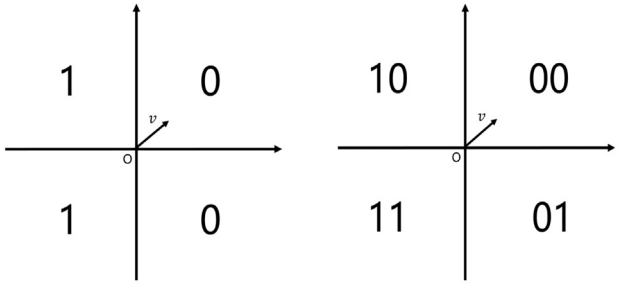
Considering that languages such as Chinese have a large number of glyphs, there may exist exceptions that some generated glyphs have poor quality. If necessary, we can fine-tune these exceptional glyphs to ensure that the generated fonts have better visual quality or refrain from using them to carry watermark information. In comparison, Qi et al. [21] and FontCode [22] artificially design Chinese fonts, which is time-consuming and heavily increases the manpower cost.

4. Watermark embedding

After generating the font codebook in advance, we can directly use it to embed the watermark information in real-time while a user is editing the document. Figure 5 shows the framework of the proposed watermark embedding process. Since we have modified the glyphs of the given target font F_0 to generate a font codebook $F = \{F_1, F_2, \ldots, F_N\}$, the corresponding glyphs can be selected to replace the original glyphs to embed the watermark information. The implementation details of these modules are illustrated as follows.

4.1. The definition of relative centroid position (RCP)

Since the distortions will cause pixel changes on glyphs, the centroid location of a single glyph may shift after cross-media transmissions. Considering this, we propose to utilize the relative centroid position (RCP) of two adjacent glyphs from different fonts to represent the watermark signal. Specifically, all glyph images are first normalized into the same coordinate system. Then, for every two adjacent glyphs (G_k, G_{k+1}), we can obtain their RCP vector $\boldsymbol{v}$ with the direction from the centroid of G_k to the centroid of G_{k+1}.



(a) The coordinate system partition when $N = 2$. (b) The coordinate system partition when $N = 4$.

Fig. 6. The relationship between the coordinate system partition and N.

Glyphs from different fonts will result in different directions of the RCP vector $\boldsymbol{v}$, so we can use the direction of $\boldsymbol{v}$ to represent watermark signals. Specifically, if we divide the whole coordinate system into two parts (i.e., left and right), each RCP vector will represent a 1-bit message. If we divide it into four parts (i.e., up-left, up-right, down-left, and down-right), each RCP vector represents a 2-bit message. We divide the coordinate system in a similar way to the font codebook generation in Section 3. Therefore, we further define N as the embedding rate to indicate the number of regions that can be used to express the watermark signal with one RCP vector.

For example, when $N = 2$ or 4, as shown in Fig. 6, the RCP vector $\boldsymbol{v}$ can carry 1 bit or 2 bits of information. To minimize the errors during cross-media transmissions, we choose the corresponding encoding strategy based on the idea of Gray code [38], which means the two adjacent regions are encoded with only one bit difference. For example, in Fig. 6(b), when $N = 4$, there is only one bit difference between two neighbouring regions (i.e., 00 - 10 - 11 - 01). In this way, we can encode the watermark signal with the direction of the RCP vector $\boldsymbol{v}$.

4.2. Message embedding

In Section 4.1, we have described how two glyphs can represent information by comparing their relative centroid positions. As we have already generated the font codebook F, we can modify the centroid location of a glyph in the original document by replacing

Algorithm 1 Target-Similar Font Codebook Generation.

Input: Original font $F_0 = \{F_0(G_0), F_0(G_1), \ldots, F_0(G_K)\}$, the predetermined *direction*, and the hyper-parameter r.

Output: Generated font with glyphs modified according to the predetermined direction.

$offset = 4$, $\lambda = 0.5$ ▷ Setting the offset of each stroke movement and the Bessel Curve modification strength.
for each glyph $F_0(G_k)$ in F_0 **do**
  Load the Bessel curve coordinates list corresponding to each stroke of the glyph into *Strokes*.
  Calculate its original centroid coordinates (x_0, y_0) via Eq. (2)
  while $|x_0 - 128| < \frac{r}{2}$ **do** ▷ Take $N = 2$ as an example. The coordinate system is divided into two regions.
    for each *stroke* in *Strokes* **do**
      for each Bessel curve point coordinates (x_i, y_i) in *stroke* **do**
        if *direction* ='left' **then**
          if *stroke* is not the leftmost or rightmost stroke of glyph $F_0(G_k)$ **then**
            $x_i = x_i - offset$ ▷ Move the stroke.
          else if (p_x, p_y) is not the leftmost or rightmost coordinates **then**
            $x_i = x_i - \frac{offset}{4}$ ▷ The leftmost or rightmost stroke are modified to a lesser extent and the leftmost and rightmost coordinates remain unchanged.
          Calculate the centroid location (s_x, s_y) of *stroke* via Eq. (2).
          $\Delta y = y_{i+1} - y_i$, $\Delta x = x_{i+1} - x_i$ ▷ The 'direction' to the next point.
          if $s_x < 128$ **then**
            $x_i = x_i - \lambda * sign(\Delta y), y_i = y_i + \lambda * sign(\Delta x)$ ▷ Increase the black pixels in the left area.
          if $s_x > 128$ **then**
            $x_i = x_i + \lambda * sign(\Delta y), y_i = y_i - \lambda * sign(\Delta x)$ ▷ Decrease the black pixels in the right area.
          Calculate the new glyph centroid coordinates (x_0, y_0) via Eq. (2).
          if $|x_0 - 128| \geq \frac{r}{2}$ **then**
            Break
        if *direction* ='right' **then**
          if *stroke* is not the leftmost or rightmost stroke of glyph $F_0(G_k)$ **then**
            $x_i = x_i + offset$ ▷ Move the stroke.
          else if (p_x, p_y) is not the leftmost or rightmost coordinates **then**
            $x_i = x_i + \frac{offset}{4}$
          Calculate the centroid location (s_x, s_y) of *stroke* via Eq. (2).
          $\Delta y = y_{i+1} - y_i$, $\Delta x = x_{i+1} - x_i$
          if $s_x < 128$ **then**
            $x_i = x_i + \lambda * sign(\Delta y), y_i = y_i - \lambda * sign(\Delta x)$ ▷ Decrease the black pixels in the left area.
          if $s_x > 128$ **then**
            $x_i = x_i - \lambda * sign(\Delta y), y_i = y_i + \lambda * sign(\Delta x)$ ▷ Increase the black pixels in the right area.
          Calculate the new glyph centroid coordinates (x_0, y_0) via Eq. (2).
          if $|x_0 - 128| \geq \frac{r}{2}$ **then**
            Break
return Generated glyphs modified according to the predetermined direction as a new font.

it with its corresponding glyph in the font codebook according to the watermark signal. Based on the generated font codebook and the RCP designed above, we propose to embed watermark information by choosing the glyphs with a specific RCP and using them to replace the original glyphs in the original documents, as shown in Fig. 7.

When setting $N = 2$ and $r = 8$, we can generate the font codebook $F = \{F_1, F_2\}$ according to Eq. (4). Given an input text $T = \{G_0, G_1, \ldots, G_n\}$, each group of adjacent glyphs G_k and G_{k+1} in T are modified according to the watermark signal w. Specifically, if $w = 0$, G_k and G_{k+1} can be replaced by:

$$G'_k = F_1(G_k), \ G'_{k+1} = F_2(G_{k+1}). \tag{5}$$

Let $\boldsymbol{v}$ represent the RCP vector of (G'_k, G'_{k+1}). As the glyphs in F_1 are all distributed on the left side of the $\boldsymbol{y}$-axis while the glyphs in F_2 on the right side, the angle θ between $\boldsymbol{v}$ and the $\boldsymbol{x}$-axis satisfies $\cos\theta > 0$.

Similarly, if $w = 1$, G_k and G_{k+1} can be replaced by:

$$G'_k = F_2(G_k), \ G'_{k+1} = F_1(G_{k+1}), \tag{6}$$

where $\cos\theta < 0$. From Eq. (4), the distance of (G'_k, G'_{k+1}) is greater than r. That is, when the RCP vector $\boldsymbol{v}$ points in the region corresponding to watermark signal w, we will choose them to replace the original two glyphs (G_k, G_{k+1}). Then, we embed the next watermark signal into (G_{k+2}, G_{k+3}). The embedding process is repeated until all the watermark information is embedded; then, we obtain the watermarked text $T' = \{G'_0, G'_1, \ldots, G'_n\}$.

5. Watermark extraction

The watermark extraction process is shown in Fig. 8. As a watermarked document spread in different media may suffer from various distortions, we should first conduct different document rectifications to recover the font information. Then, we apply the segmentation process to obtain the single normalized glyph and extract the watermark information. With the help of the RCP-based embedding, the extraction process does not need the OCR to recognize the semantic information of glyphs, which can significantly improve the extraction accuracy and efficiency.

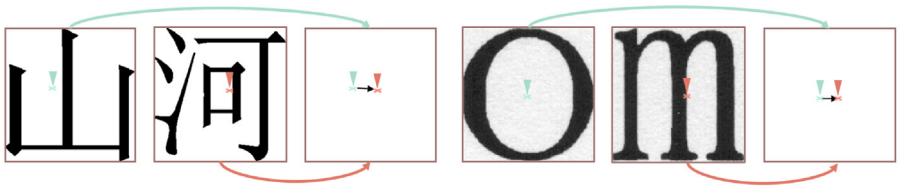
5.1. Document rectification

The distortions of print-scanning and print-shooting processes mainly come from the paper perspective and document deformation (i.e., folded, curved, or crumpled paper). We use the method proposed by Kil et al. [39] to recover the warped document image by using the natural features of line segments and text lines in the document image.

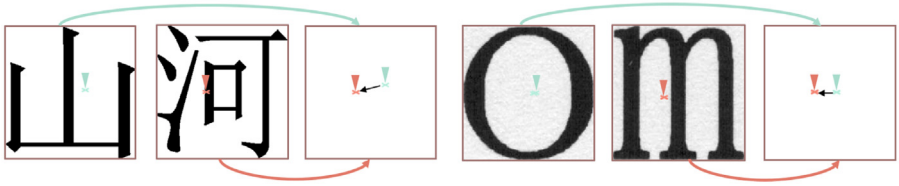
Digital-based documents usually suffer from image down-sampling and screen-camera shooting distortions. Image down-sampling occurs when the screen converts vectogram format glyphs to dot-map format images. This kind of distortion can invalidate the method in [21] because the tiny difference between their templates will be erased by image down-sampling. However, our RCP-based watermark can be well preserved due to its novel stability.

The screen-camera shooting process will introduce more complicated distortions, such as moirE interference and geometric distortion. To recover the document distorted by the screen-camera channel, we use the Gaussian low-pass filter to remove the moirE patterns and high-frequency distortions in the screen, as shown in Fig. 8. Specifically, the recovered document image $\widehat{I}$ can be calculated by

$$\widehat{I}(x, y) = G(x, y, \sigma) * \bar{I}(x, y) \tag{7}$$



(a) The RCP represents bit '0'.



(b) The RCP represents bit '1'.

Fig. 7. The RCP examples of embedding bit '0' and '1' when $N = 2$.

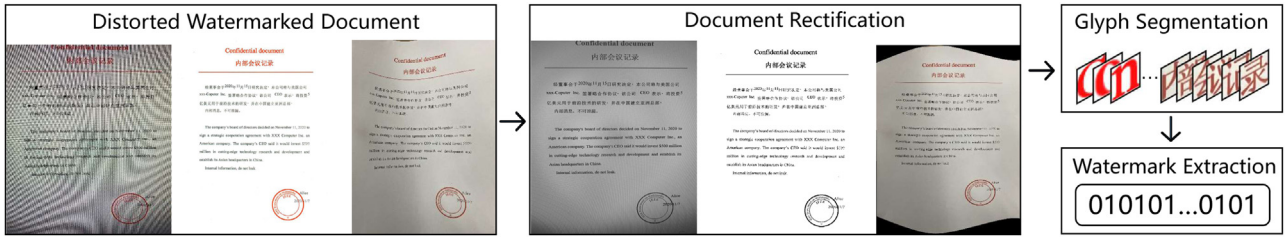


Fig. 8. The watermark extraction process.

and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \cdot e^{-(\frac{x^2+y^2}{2\sigma^2})}, \tag{8}$$

where (x, y) are the coordinates of the image. $\bar{I}$ indicates the document image taken by a camera. σ is the variance of the Gaussian filter. The value of σ determines the smoothness of the image. We choose $\sigma = 250$ in our experiments because it can well preserve the glyph information while weakening the moirE patterns. Then, we can use the method in [39] to rectify the image.

5.2. Glyph segmentation and normalization

To segment each glyph with its minimum circumscribed rectangle, we designed a new projection segmentation algorithm based on the traditional vertical projection character segmentation algorithm. As Fig. 9 shows, the document image is first horizontally projected, and each line of text is segmented with a size of (l, h). Then, each segmented line is vertically projected to obtain a single glyph image with a size of (l', h). Finally, the height of the minimum circumscribed rectangle h' is obtained by horizontally projecting the single glyph image, and then we resize the glyph image with size (l', h') to (256,256) to obtain the normalized glyph image.

5.3. Message extraction

Given the watermarked document T', we first use the methods in Section 5.1 to obtain the rectified watermarked text $\widehat{T} = \{\widehat{G}_0, \widehat{G}_1, \ldots, \widehat{G}_n\}$. Then the whole text is cut into glyph pieces by our normalization algorithm.

To cope with the embedding scheme in Section 4, we propose to extract the watermark signals by comparing the RCP of the adjacent glyphs. When $N = 2$, the process can be formulated as

$$w' = \begin{cases} 0, & if\ \cos\widehat{\theta} > 0 \\ 1, & if\ \cos\widehat{\theta} < 0 \end{cases}, \tag{9}$$

where w' is the extracted watermark signal from each two adjacent watermarked glyphs $(\widehat{G}_k, \widehat{G}_{k+1})$ and $\widehat{\theta}$ means the angle between their RCP vector $\widehat{\boldsymbol{v}}$ and the $\boldsymbol{x}$-axis.

6. Experiments

In this section, we compare our method with previous font-based text watermarking methods [21,22]. Note that because the other types of schemes like the format-based schemes, the linguistic-based schemes, and the image-based schemes do not modify the font to embed watermark information and their usability is also limited, we only reported a qualitative comparison for them in Table 1. Since we cannot obtain the font files and network models of [22] for re-implementation, we could only use the data from their paper for comparison in the print-camera shooting experiment and then align our experimental setup for comparison. The implementation details are described in Section 6.1. To prove that our method has strong robustness in different languages and media, we use documents whose content includes Chinese, English, Arabic numerals, and their mixture to conduct experiments in different scenarios. We also discuss the usability of our algorithm in other languages in ablation studies. In Section 6.2, we test the robustness of our algorithm in the digital domain, considering the distortions that documents may suffer from digital channels. After that, we perform experiments on cross-media scenarios,

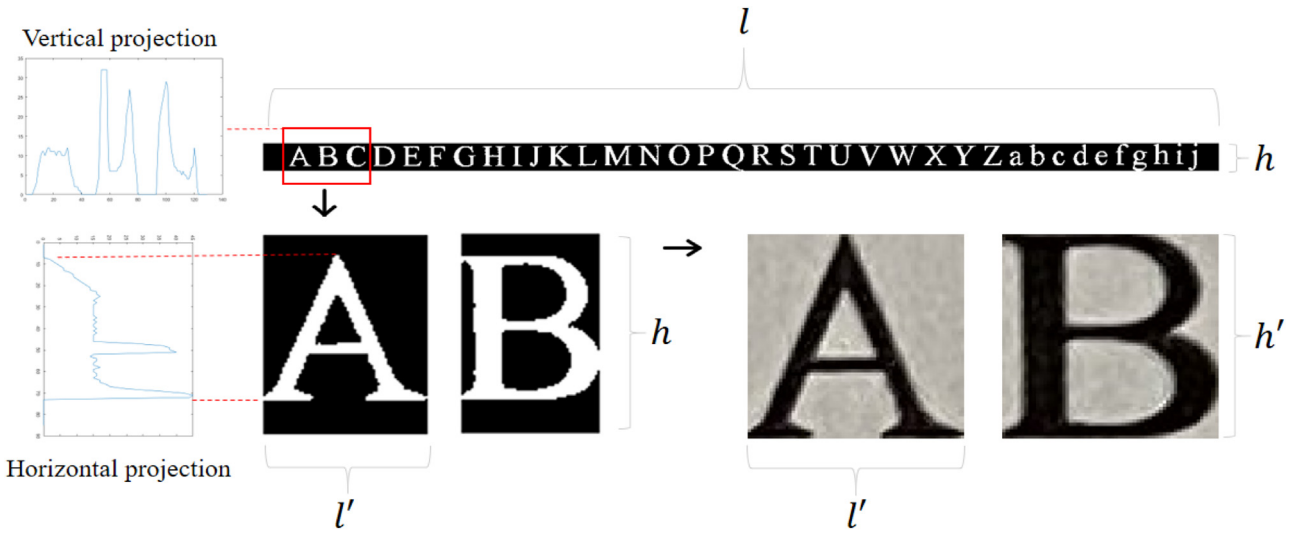


Fig. 9. The process of three time projections.

that is, screen-camera shooting, print-camera shooting, and print-scanning distortions are considered in Section 6.3. Section 6.4 evaluates the visual quality of our method based on subjective evaluation experiments. Finally, more analysis and ablation studies will be provided to justify our design.

6.1. Implementation details

For the screen-camera shooting experiments, the monitor used for the screen shooting experiments is 'Lenovo-P22i', the printer for the print shooting experiments is 'HP OfficeJet Pro 8720', and its scanning resolution is 600 dpi. For image capturing, the 'iPhone 11' is used by default. In our experiment for the robustness test, we choose $r = 8$ and $N = 2$ to generate the font codebook automatically. We will discuss the influence of the value of r on our algorithm in ablation experiments. We express the experimental results in terms of average bit accuracy to quantify the performance of our method. This is equivalent to the Bit Error Rate (BER), which is related by $Acc = 1 - BER$.

Languages can be divided into hieroglyphics, phonetic, and ideophonetic languages by their pronunciation and structure. Hieroglyphics are primitive human characters, such as Holy script, Maya, and Oracle. The phonetic and ideophonetic languages are represented by English and Chinese respectively, on which we will carry out our main experiments. Specifically, we choose the fragments from *Game of Thrones* for experiments of English with 8978 useful letters, a collection of poems with 47,961 useful characters from the Tang dynasty for Chinese, and a collection of financial statements with 65,536 useful glyphs including Chinese, English, and Arabic numerals for the multilingual language scenario. When $N = 2$, the payload is 0.5 bit/glyph. In addition, we also test the usability of our method in other languages in the ablation experiments.

For a fair comparison, we guarantee that the extent of glyph modification in the baseline method [21] and our method be as consistent as possible. Although their method is aimed at paper-based text documents, we still show their results on digital-based text documents as a baseline. It should be noted that the performance of the method [21] is strongly dependent on the OCR algorithm to recognize the semantic information of the glyphs. Only after OCR recognition is accurate, can it match the corresponding template to extract watermark information, which is hard to achieve in documents distorted by image down-sampling and cross-media distortions. Therefore, to conduct comparative experiments, we directly assume that the baseline method can completely and correctly identify the semantic information of characters in documents, which improves their actual performance. In contrast, there is no need for our method to recognize any semantic information of the glyphs, which greatly improves the robustness and efficiency in real use.

6.2. The robustness in the digital domain

In this section, a series of experiments are carried out to evaluate the robustness of our methods in the digital domain. Since a screen relies on an array of pixels to display an image, a glyph displayed on a screen is a dot-map image and its resolution is limited by the monitor resolution and fontsize we set. It should also be noted that when the fontsize is less than 8 pt, the human eyes can hardly recognize the glyph even with a 2K resolution display. An example of glyphs with different fontsizes from 5 to 20 pt on a printed paper, a screenshot and a photograph of a screen with a resolution of 1920×1080 is shown in Fig. 10. It can be seen that when the fontsize is small, Chinese characters will lose more glyph outline information than English characters because of their complex structure. So we first test our algorithm under the down-sampling distortion of the monitor with the screenshot document images in different fontsizes. The correspondence between fontsize(pt) f of a glyph and its resolution pixel p is $p \approx 4f/3$. Figure 11 shows the average extraction accuracy on Chinese, English, and multilingual documents under the down-sampling screenshots of the screen. It can be seen that our method can achieve strong resilience for low-resolution images with the extraction accuracy for all the documents with different languages more than 90% if the fontsize is larger than 10 pt.

Considering that the screenshot images of documents may be compressed again and propagated on social networks, we also test the robustness of our algorithm under different JPEG compression quality factor (QF) values. Specifically, we choose the multilingual language documents with fontsize from 10 to 14 pt to carry out the experiment. Table 2 shows the detailed results. The watermarked document images are compressed with JPEG compression QF from 10 to 100. We can see that the proposed scheme performs well under JPEG compression with all quality factors, owing to the stability of the glyph outline and RCP in the process of compression.

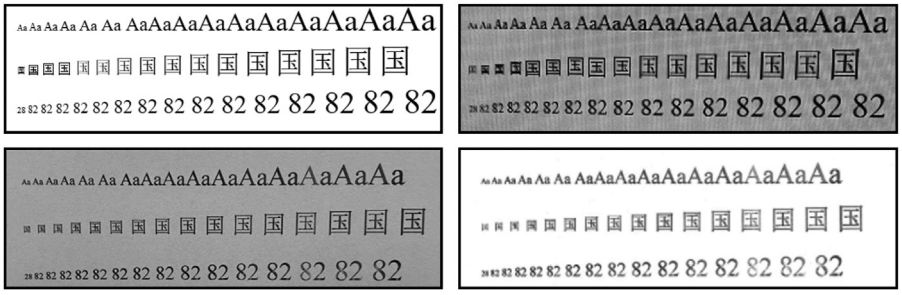
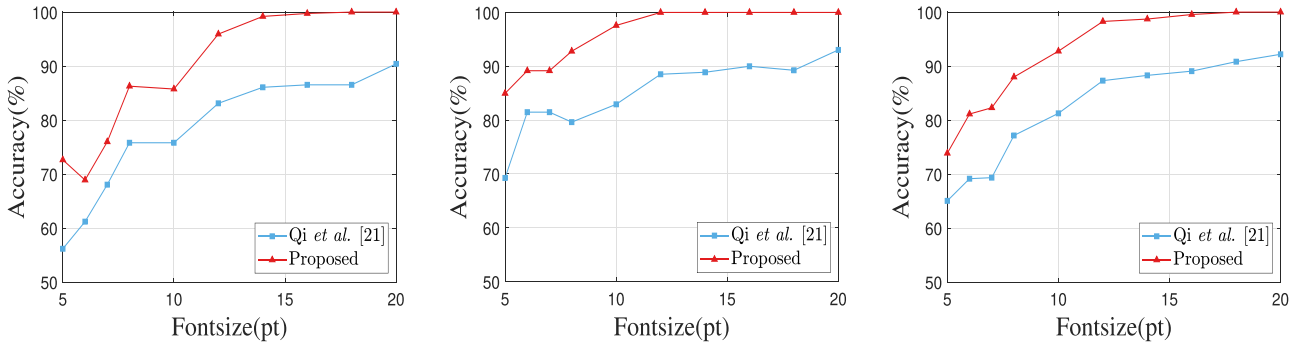


Fig. 10. The glyphs with font-size from 5 to 20 pt in four kinds of distortions (Top-left: screenshot, top-right: screen-camera shooting, bottom-left: print-camera shooting, bottom-right: print-scanning).



(a) Chinese documents. (b) English documents. (c) Multilingual documents.

Fig. 11. Extraction accuracy of the screenshots of documents with different languages and fontsizes.

Table 2
Average bit accuracy of the screenshots of watermarked multilingual document images under different JPEG Compression Quality Factors from 10 to 100.

Fontsize	QF	10	20	30	40	50	60	80	100
10 pt	Qi et al. [21]	78.71%	78.32%	80.08%	81.25%	79.10%	79.69%	79.49%	78.32%
	Proposed	**84.24%**	**86.69%**	**86.46%**	**87.51%**	**87.35%**	**86.05%**	**86.10%**	**86.66%**
12 pt	Qi et al. [21]	85.35%	83.95%	87.30%	86.13%	87.89%	88.09%	88.66%	87.44%
	Proposed	**84.76%**	**95.77%**	**97.60%**	**97.59%**	**97.77%**	**98.05%**	**98.21%**	**98.28%**
14 pt	Qi et al. [21]	87.50%	86.13%	87.58%	87.89%	88.63%	87.18%	87.79%	89.32%
	Proposed	**93.66%**	**95.75%**	**96.79%**	**97.17%**	**98.05%**	**97.80%**	**97.53%**	**97.69%**

6.3. The robustness in cross-media transmissions

As previously described, in addition to distortions in the digital domain, we further test the cross-media robustness of our algorithm from three aspects: screen-camera shooting, print-camera shooting, and print-scanning.

6.3.1. Robustness to screen-camera shooting

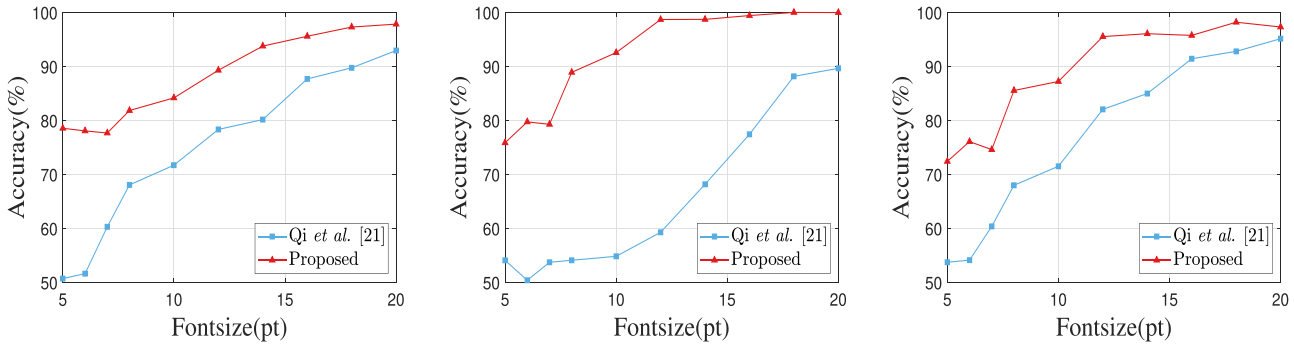
Figure 12 shows the test results of the screen-camera shooting process with the screen-camera distance of 20 cm. It can be seen that the documents watermarked by our algorithm can well resist the distortions introduced by the screen-camera shooting channel.

Then, we further test the robustness of our algorithm against the screen-camera shooting process with different shooting configurations, i.e., different shooting distances and angles. In the experiments for different shooting distances, we set the fontsize as 14 pt and choose the multilingual language documents to embed watermark information. Table 3 shows detailed results with different shooting distances. Compared with the baseline method [21], our method still maintains high accuracy at all different distances from 10 to 80 cm. It should also be noted that the decline in the accuracy rate of our method is not directly proportional to distance. By analyzing the images of different distances in Fig. 13, we surmise that this finding is because the longer distance weakened the moirĘ effect in the image, while the glyph outline information is still well preserved. Table 4 shows the accuracy rate for different screen-camera shooting angles. It can be seen that the shooting angles have little influence on our algorithm as the RCP of every two glyphs is stable during perspective transformation.

6.3.2. Robustness to print-camera shooting

Figure 14 shows the extraction accuracy of the print-camera shooting watermarked images. As the resolution of the printed glyphs is usually higher than that of screen displayed glyphs, the glyphs used on paper documents can be approximated as vectogram format glyphs. Therefore, the glyph outline in a paper document is better preserved than on an electronic screen under the same fontsize, which can also be seen from Fig. 10 left. For the print-camera shooting experiment of English text, we directly use the experimental data from the paper of Xiao et al. [22] for comparison because we cannot obtain their font files and network models for re-implementation. As previously mentioned, the application scenario of [22] used to be English posters with larger fontsizes than common documents, so their accuracy is lower than our method when the fontsize is less than 20 pt.

Then, we also further test the robustness of our algorithm against the print-camera shooting process with different shooting distances and angles. As shown in Tables 5 and 6, our algorithm



(a) Chinese documents. (b) English documents. (c) Multilingual documents.

Fig. 12. Extraction accuracy of screen-camera shooting documents with different languages and fontsizes.

Table 3
Average bit accuracy of watermarked multilingual document images with screen-camera distances from 10 to 80 cm.

Distance (cm)	10	20	30	40	50	60	70	80
Qi et al. [21]	91.02%	84.38%	82.26%	87.21%	89.12%	85.93%	81.63	81.25%
Proposed	**97.40%**	**96.82%**	**97.33%**	**95.01%**	**97.27%**	**96.29%**	**94.23%**	**94.14%**

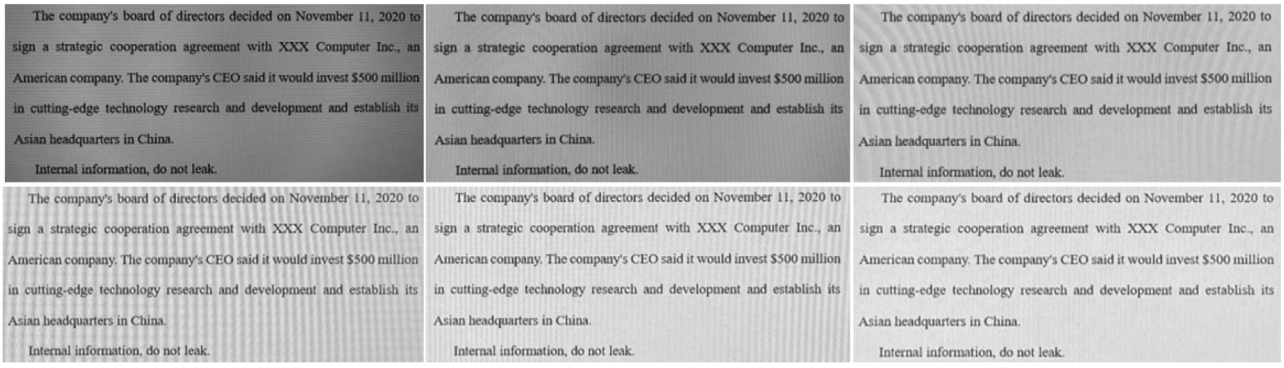
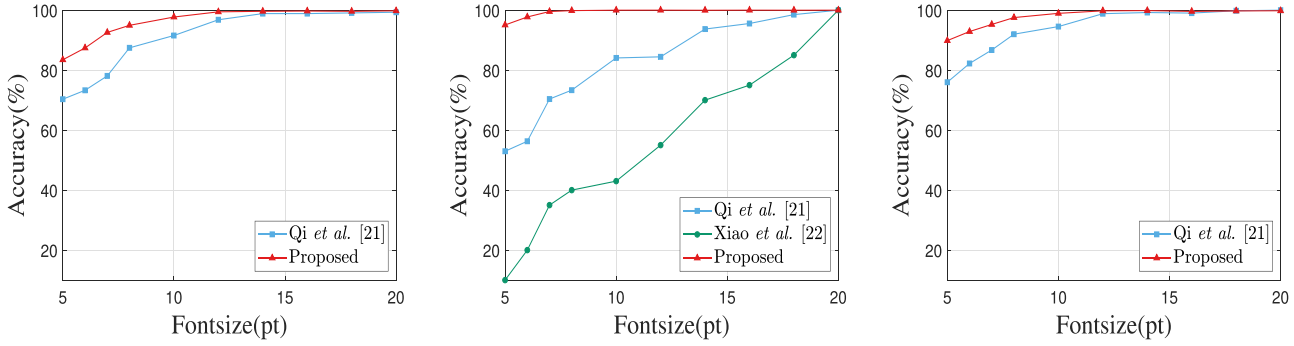


Fig. 13. The gray-scale screen-camera shooting photos with shooting distances from 10 cm (up-left) to 60 cm (down-right).

Table 4
Average bit accuracy of watermarked multilingual document images with different screen-camera angles from -40° to 40°.

Angles (°)	-40°	-30°	-20°	-10°	10°	20°	30°	40°
Qi et al. [21]	82.62%	84.96%	85.16%	89.65%	87.63%	84.31%	83.59%	83.20%
Proposed	**96.37%**	**96.08%**	**94.57%**	**96.17%**	**95.26%**	**96.68%**	**94.79%**	**97.75%**



(a) Chinese documents. (b) English documents. (c) Multilingual documents.

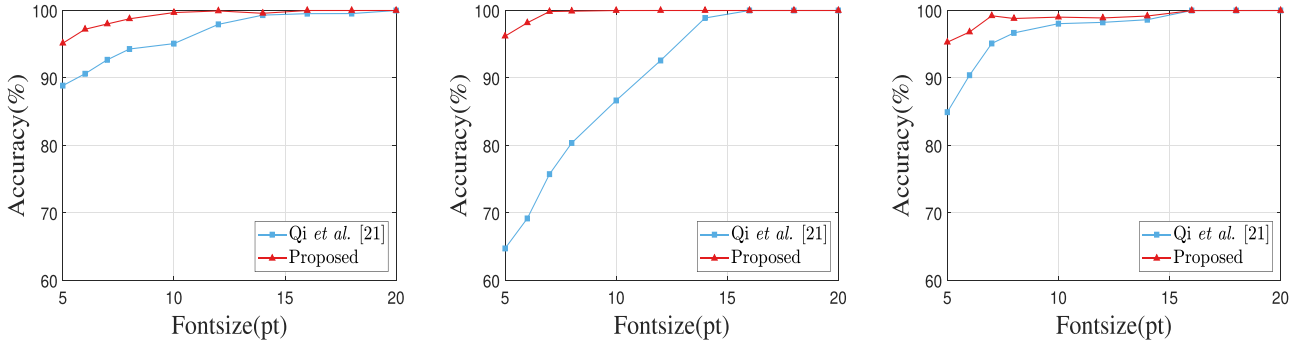
Fig. 14. Extraction accuracy of print-camera shooting documents with different languages and fontsizes.

Table 5
Average bit accuracy of watermarked multilingual document images with print-camera distances from 10 to 80 cm.

Distance (cm)	10	20	30	40	50	60	70	80
Qi et al. [21]	**99.22%**	98.89%	97.27%	95.31%	90.43%	82.61%	81.64%	65.27%
Proposed	98.64%	**99.98%**	**99.85%**	**99.41%**	**97.41%**	**93.72%**	**93.08%**	**90.79%**

Table 6
Average bit accuracy of watermarked multilingual document images with different print-camera angles from -40° to 40°.

Angles (°)	-40°	-30°	-20°	-10°	10°	20°	30°	40°
Qi et al. [21]	97.07%	97.46%	97.66%	99.22%	98.63%	97.62%	98.44%	96.68%
Proposed	**99.57%**	**99.69%**	**99.92%**	**99.94%**	**99.75%**	**99.68%**	**99.21%**	**99.58%**



(a) Chinese documents. (b) English documents. (c) Multilingual documents.

Fig. 15. Extraction accuracy of print-scanning documents with different languages and fontsizes.

Table 7
Average bit accuracy of watermarked multilingual document under different print-scanning times.

Fontsize	Print-scanning times	1	2	3	4
10 pt	Qi et al. [21]	98.63%	**95.82%**	86.52%	81.25%
	Proposed	**99.94%**	95.75%	**94.11%**	**92.53%**
12 pt	Qi et al. [21]	98.24%	97.85%	92.38%	89.77%
	Proposed	**99.41%**	**98.40%**	**96.68%**	**93.63%**
14 pt	Qi et al. [21]	98.55%	97.27%	97.07%	96.09%
	Proposed	**99.68%**	**99.30%**	**97.51%**	**97.26%**

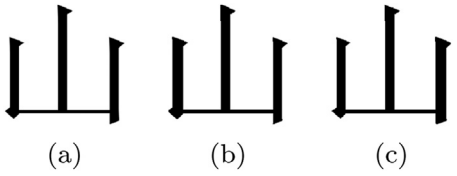


Fig. 16. The original glyph image (a), the generated glyph with the centroid moved to the left (b), and the generated glyph with the centroid moved to the right (c).

has great performance in different distances and angles, thanks to the novelty of RCP and the high resolution of printed documents.

6.3.3. Robustness to print-scanning distortions

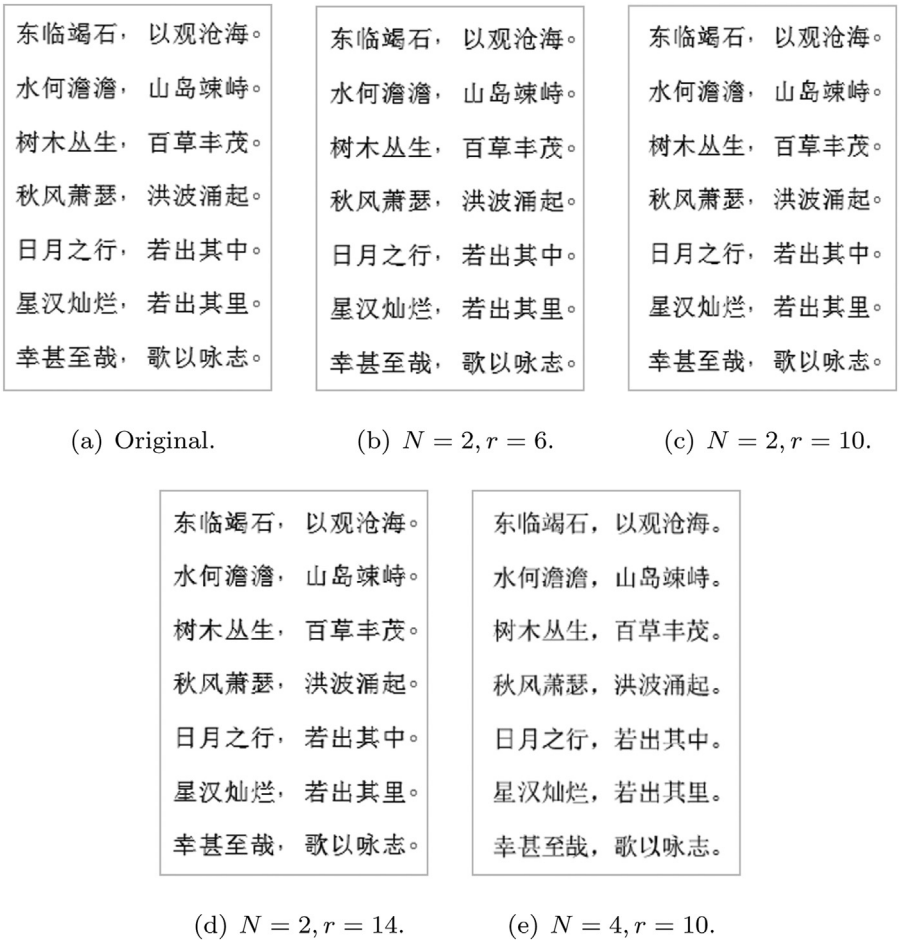
The robustness test for the print-scanning process is carried out with different fontsizes and scanning times. We scanned the printed watermarked documents with different fontsizes from 5 to 20 pt, and the extraction accuracy is shown in Fig. 15. As the RCP can remain stable from the error diffusion during scanning, the accuracy is mostly dependent on whether the watermarked document scanned is clean.

In real-life scenarios, a scanned document image may also be printed, and then the printed scanned-document may also be scanned, which means the print-scanning channel may be applied more than once on a document. Therefore, we test the robustness of our method with different print-scanning times while the fontsize is set from 10 to 14 pt. The detailed comparison results are shown in Table 7. We can see that the extraction accuracy is still higher than 90% even though the documents have passed through the print-scanning channel four times. Additionally, we noticed that the major distortion that occurred in this experiment was caused by the missing glyphs that were not completely scanned by the scanner. Therefore, a reasonable assessment is that as long as the font information in the scanned image exists, we can extract the watermark information successfully.

6.4. Perceptual evaluation

In the field of image watermarking, there are objective criteria for imperceptibility, such as the peak-signal-to-noise-ratio (PSNR) and structural similarity (SSIM). They are designed to compare two images at the pixel level but cannot translate the subjective feelings that emanate from the of human eye assessment. However, glyph modification occurs not only in the pixel dimension but also in the stroke dimension, which means that a large number of pixel values will be changed. As shown in Fig. 16, for normal readers, the generated glyphs can match the semantic images that form in the brain. They will not detect these subtle changes on a glyph without careful comparison with the original image. However, the values of PSNR and SSIM are not good, as shown in Table 8.

To evaluate the subjective visual quality of the proposed method, we invite 80 participants to rate the same documents watermarked by different parameters r and N according to the absolute evaluation scales in Table 9. We used three different fontsize document groups (i.e., 14, 16, and 18 pt), each group including five



(a) Original. (b) $N = 2, r = 6$. (c) $N = 2, r = 10$.

(d) $N = 2, r = 14$. (e) $N = 4, r = 10$.

Fig. 17. The example of screenshot documents used in perceptual evaluation.

Table 8
The values of PSNR and SSIM of (b), (c) with (a).

	(a)	(b)	(c)
PSNR (dB)		19.9573	17.8004
SSIM		0.8713	0.8912

kinds of documents: original documents and watermarked documents with different embedding parameters. One group of documents used in the experiment is shown in Fig. 17 and the font codebook used in this experiment is generated automatically without human interference to improve its visual quality. Since we have compared our extraction accuracy with Qi's method at the same visual quality, we only compare the visual quality of the documents watermarked by our method in different parameters with the original documents. The participants are informed in advance that the score of the original documents is 5. As shown in Table 9, when $N = 2$, $r \leq 10$, our method has a visual quality that is mostly similar to that of the original documents. When N increases, the glyphs will be modified more and the visual quality will be reduced, while r actually has no significant effect on the visual effect if it is less than 10.

It is worth mentioning that if we do not provide the original documents as a reference, participants can hardly detect the changes in glyphs in the usual reading time with $N = 2$, which we think is because the human brain has a good automatic adjustment function for the change of font in the process of glyph recognition [35].

6.5. Ablation experiments

In this section, we will provide more ablation experiments from various perspectives to justify the design of our method.

Table 9
The average score of the visual quality of our watermarked documents with different parameters. In the evaluation, 5 means 'Imperceptible', 4 means 'Perceptible, but not annoying', 3 means 'Slightly annoying', 2 means 'Annoying', and 1 means 'Very annoying'.

Document	Original	$N = 2, r = 6$	$N = 2, r = 10$	$N = 2, r = 14$	$N = 4, r = 10$
14 pt	**5**	4.87	4.65	3.99	3.32
16 pt	**5**	4.66	4.50	3.85	3.19
18 pt	**5**	4.53	4.29	3.77	2.82

Table 10
Average bit accuracy of screenshots of our watermarked document with different languages and fontsizes.

Fontsize (pt)	5	6	7	8	10	12	14	16	18	20
Japanese	68.94%	81.34%	82.69%	87.36%	94.04%	98.65%	98.12%	98.43%	99.78%	99.42%
Korean	93.56%	92.17%	91.74%	99.96%	99.47%	99.33%	99.97%	100%	100%	100%
Russian	83.66%	93.56%	92.86%	98.16%	99.14%	100%	100%	100%	100%	100%
Tibetan	79.76%	90.01%	89.37%	98.86%	99.80%	98.77%	100%	100%	100%	100%
Spanish	86.40%	93.14%	98.95%	96.18%	99.24%	99.32%	99.75%	99.93%	100%	100%
German	85.49%	96.67%	91.71%	96.63%	97.94%	100%	99.89%	98.77%	100%	100%

Table 11
The average bit accuracy with different combinations of screen-camera and print-camera devices.

Phone/Display device	AOC C24B1	Lenovo P22i	ViewSonic VA2261	HP OfficeJet Pro 8720	HP Color LaserJet Pro M180n	Aficio MP 9002
iPhone 11	96.87%	97.33%	96.84%	99.85%	99.29%	99.22%
Mi 9	96.99%	99.34%	97.01%	99.48%	100%	99.96%
Huawei Mate 40	98.28%	99.20%	98.20%	99.99%	100%	100%

Table 12
The average bit accuracy with different combinations of print-scan devices.

Printer/Scanner	HP N4000 SNW1	EPSON DS 1610	RICOH MP C6503SP
HP OfficeJet Pro 8720	99.68%	99.91%	99.87%
HP Color LaserJet Pro M180n	98.86%	99.73%	99.91%
Aficio MP 9002	99.88%	99.93%	99.97%

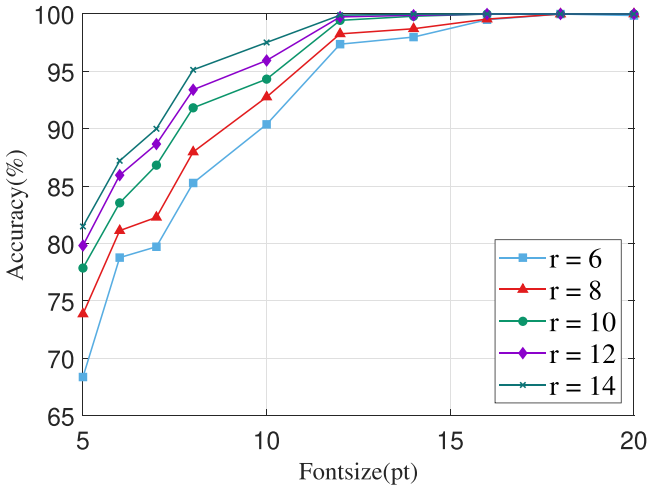


Fig. 18. The average bit accuracy of English documents with r from 6 to 14 px under screenshot distortions.

6.5.1. The impact of threshold r on robustness

To verify how the centroid distance threshold r affects the robustness of our method, we change r from 6 to 14 px under the 256×256 coordinate system and the average extraction accuracy of screenshot images with different r values is shown in Fig. 18. When the fontsize is less than 10 pt, the glyphs lose more information from their down-sampling process to dot-map images. We can solve this problem by increasing r. In the real scenario, $r = 8$ can provide enough robustness.

6.5.2. The usability in other languages

To prove the language universality of our method, we select other languages that are commonly used internationally, i.e. Japanese, Korean, Russian, Tibetan, Spanish, and German to test the extraction accuracy of their screenshot images with different fontsizes. As shown in Table 10, our method can perform well in different languages with normal fontsizes.

6.5.3. The adaptability to different devices

Adaptability to different devices is a key consideration for applicability. To evaluate it, we use the combinations of different devices to test the average accuracy of the watermarked documents. The devices we use include mobile phones ("iPhone 11", "Mi 9", and "Huawei Mate 40"), screens ("AOC-C24B1", "Lenovo-P22i", and "ViewSonic-VA2261"), and printers ("HP OfficeJet Pro 8720", "HP Color LaserJet Pro M180n", and "Aficio MP 9002"). Since we have carried out extensive experiments on different fontsizes, shooting angles, and shooting distances, here we set them as (12 pt, 0°, and 30 cm) to test the adaptability of our method to different devices. The results in Table 11 indicate that for all combinations of devices, the extraction accuracies of this method are all above 96%. We can assert that this method has excellent applicability to mobile phones, screens, and printers used in the tests.

Similarly, different print-scan device combinations have also been tested. The results are shown in Table 12, which indicates that the print-scan channels introduce very few distortions to the glyphs.

6.5.4. The robustness to superposed cross-media distortions

Although our watermarking scheme mainly focuses on common cross-media distortion scenarios (i.e., screenshot-compression, screen-camera shooting, print-camera shooting, and print-scanning), we further evaluate the robustness of our scheme under superposed cross-media distortions. In fact, screen-camera shooting is a kind of superimposed distortion (i.e., "screenshot→screen-camera channel distortion"). We further test the average bit accuracy under the "screenshot→reduce size→print→photograph" process. The experiments were conducted at different fontsizes (i.e., 10, 14, and 18 pt) and resize scales (i.e., 0.5, 0.7, and 0.9), and the print-camera distance was 20 cm. As shown in Table 13, our scheme still provides better robustness to this type of superimposed cross-media distortion compared with the baseline method. Indeed, there will be a decrease in extraction accuracy if more types of cross-media distortion are superimposed or stronger distortions are present, but the original content of the document will also be seriously corrupted.

Table 13
Average bit accuracy of watermarked multilingual document images under the superimposed "screenshot→reduce size→print→photograph" process.

Fontsize	Resize Scale	0.5	0.7	0.9
10 pt	Qi et al. [21]	55.66%	66.21%	76.37%
	Proposed	**78.17%**	**79.75%**	**84.45%**
14 pt	Qi et al. [21]	67.38%	74.41%	87.50%
	Proposed	**88.09%**	**91.30%**	**93.05%**
18 pt	Qi et al. [21]	79.10%	84.77%	90.63%
	Proposed	**89.73%**	**93.75%**	**94.38%**

7. Conclusion

Traditional font-based text watermarking algorithms only focus on specific usage scenarios, and the robustness of these methods mostly relies on the accuracy of the OCR method used, which highly limits their usability in the real world.

This paper proposes a language universal font watermarking scheme with cross-media robustness. With our RCP-based watermarking scheme, the algorithm does not need to recognize the semantic information of each glyph before extracting the watermark, and the computational complexity is also much lower than the methods using template matching and deep neural networks. At the embedding side, we designed a series of methods to automatically generate target-similar fonts with different centroid distributions from the original font, with which we can replace the glyphs in original documents according to the watermark signal. At the extraction side, we extract the watermark information by simply comparing the RCP of two adjacent glyphs.

The extraction accuracy is mostly dependent on whether we can segment the glyph outline precisely, which is very obvious when using a larger N. If we can segment the glyphs more accurately, the coordinate system can be divided into more regions to encode the watermark signal, which will greatly improve the capacity of our methods. Therefore, in the future, we will focus on how to segment the glyphs more accurately for the extraction side and try to utilize the super resolution algorithm to optimize the usability of the font-based text watermarking algorithm in small fontsize scenes. For the embedding side, our goal is to automatically generate fonts with higher visual semantic quality.

CRediT authorship contribution statement

Xi Yang: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Visualization, Writing – original draft.

Weiming Zhang: Methodology, Project administration, Resources, Writing – review & editing.

Han Fang: Formal analysis, Writing – review & editing.

Zehua Ma: Software, Visualization.

Nenghai Yu: Funding acquisition, Supervesion.

All the authors approved the final version of the manuscript.

Declaration of Competing Interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled, "Language Universal Font Watermarking with Multiple Cross-media Robustness".

Acknowledgment

This work was supported in part by the Natural Science Foundation of China under Grant 62072421, 62002334, 62121002 and U20B2047, Anhui Science Foundation of China under Grant 2008085QF296, the Exploration Fund Project of University of Science and Technology of China under Grant YD3480002001, and Fundamental Research Funds for the Central Universities under Grant WK5290000001.

References

[1] R. Hu, S. Xiang, Lossless robust image watermarking by using polar harmonic transform, Signal Process. 179 (2021) 107833.
[2] L. Zhang, D. Wei, Image watermarking based on matrix decomposition and gyrator transform in invariant integer wavelet domain, Signal Process. 169 (2020) 107421.
[3] H. Fang, W. Zhang, H. Zhou, H. Cui, N. Yu, Screen-shooting resilient watermarking, IEEE Trans. Inf. Forensics Secur. 14 (6) (2019) 1403–1418.
[4] M. Tancik, B. Mildenhall, R. Ng, StegaStamp: invisible hyperlinks in physical photographs, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, IEEE, Seattle, WA, USA, 2020, pp. 2114–2123. June 13–19, 2020
[5] Z. Yin, X. She, J. Tang, B. Luo, Reversible data hiding in encrypted images based on pixel prediction and multi-MSB planes rearrangement, Signal Process. 187 (2021) 108146.
[6] Z. Liu, F. Zhang, J. Wang, H. Wang, J. Huang, Authentication and recovery algorithm for speech signal based on digital watermarking, Signal Process. 123 (2016) 157–166.
[7] Z. Liu, Y. Huang, J. Huang, Patchwork-based audio watermarking robust against de-synchronization and recapturing attacks, IEEE Trans. Inf. Forensics Secur. 14 (5) (2019) 1171–1180.
[8] G. Hua, J. Huang, Y.Q. Shi, J. Goh, V.L. Thing, Twenty years of digital audio watermarking–a comprehensive review, Signal Process. 128 (2016) 222–242.
[9] W. Jiang, X. Huang, Y. Quan, Audio watermarking algorithm against synchronization attacks using global characteristics and adaptive frame division, Signal Process. 162 (2019) 153–160.
[10] A. Cedillo-Hernandez, M. Cedillo-Hernandez, M. Garcia-Vazquez, M. Nakano-Miyatake, H. Perez-Meana, A. Ramirez-Acosta, Transcoding resilient video watermarking scheme based on spatio-temporal HVS and DCT, Signal Process. 97 (2014) 40–54.
[11] M. Asikuzzaman, M.R. Pickering, An overview of digital video watermarking, IEEE Trans. Circuits Syst. Video Technol. 28 (9) (2018) 2131–2153.
[12] N.S. Kamaruddin, A. Kamsin, L.Y. Por, H. Rahman, A review of text watermarking: theory, methods, and applications, IEEE Access 6 (2018) 8011–8028.
[13] N.F. Maxemchuk, S.H. Low, Marking text documents, in: Proceedings 1997 International Conference on Image Processing, ICIP '97, Santa Barbara, California, USA, 1997, p. 13. October 26–29, 1997
[14] J.T. Brassil, S. Low, N.F. Maxemchuk, Copyright protection for the electronic distribution of text documents, Proc. IEEE 87 (7) (1999) 1181–1196.
[15] U. Topkara, M. Topkara, M.J. Atallah, The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions, in: Proceedings of the 8th Workshop on Multimedia & Security, MM&Sec 2006, pp. 164–174.
[16] M. Topkara, U. Topkara, M.J. Atallah, Words are not enough: sentence level natural language watermarking, in: Proceedings of the 4th ACM International Workshop on Contents Protection and Security, 2006, pp. 37–46.
[17] Z. Yang, S. Zhang, Y. Hu, Z. Hu, Y. Huang, VAE-Stega: Linguistic steganography based on variational auto-encoder, IEEE Trans. Inf. Forensics Secur. 16 (2021) 880–895.
[18] H. Fang, W. Zhang, Z. Ma, H. Zhou, S. Sun, H. Cui, N. Yu, A camera shooting resilient watermarking scheme for underpainting documents, IEEE Trans. Circuits Syst. Video Technol. 30 (11) (2020) 4075–4089.
[19] V.L. Cu, J. Burie, J. Ogier, C. Liu, A robust data hiding scheme using generated content for securing genuine documents, in: 2019 International Conference on Document Analysis and Recognition, ICDAR 2019, IEEE, 2019, pp. 787–792.
[20] P.V.K. Borges, J. Mayer, Text luminance modulation for hardcopy watermarking, Signal Process. 87 (7) (2007) 1754–1771.
[21] W. Qi, W. Guo, T. Zhang, Y. Liu, Z. Guo, Robust authentication for paper-based text documents based on text watermarking technology, Math. Bioences Eng. 16 (4) (2019) 2233–2249.
[22] C. Xiao, C. Zhang, C. Zheng, FontCode: embedding information in text documents using glyph perturbation, ACM Trans. Graph. 37 (2) (2018) 15:1–15:16.
[23] N.F. Maxemchuk, Electronic document distribution, AT&T Tech. J. 73 (5) (1994) 73–80.
[24] S.H. Low, N.F. Maxemchuk, A.M. Lapone, Document identification for copyright protection using centroid detection, IEEE Trans. Commun. 46 (3) (1998) 372–383.
[25] Y. Kim, K. Moon, I. Oh, A text watermarking algorithm based on word classification and inter-word space statistics, in: 7th International Conference on Document Analysis and Recognition (ICDAR 2003), IEEE Computer Society, 2003, pp. 775–779.
[26] H.M. Meral, B. Sankur, A.S. Özsoy, T. Güngör, E. Sevinç, Natural language watermarking via morphosyntactic alterations, Comput. Speech Lang. 23 (1) (2009) 107–125.
[27] H. Yang, A.C. Kot, Pattern-based data hiding for binary image authentication by connectivity-preserving, IEEE Trans. Multim. 9 (3) (2007) 475–486.

[28] Y. Kim, I. Oh, Watermarking text document images using edge direction histograms, Pattern Recognit. Lett. 25 (11) (2004) 1243–1251.
[29] X. Li, B. Zhang, J. Liao, P.V. Sander, Document rectification and illumination correction using a patch-based CNN, ACM Trans. Graph. 38 (6) (2019) 168:1–168:11.
[30] C. Chen, W. Huang, L. Zhang, W.H. Mow, Robust and unobtrusive display-to–camera communications via blue channel embedding, IEEE Trans. Image Process. 28 (1) (2019) 156–169.
[31] H. Cui, H. Bian, W. Zhang, N. Yu, UnseenCode: Invisible on-screen barcode with image-based extraction, in: 2019 IEEE Conference on Computer Communications, INFOCOM 2019, IEEE, 2019, pp. 1315–1323.
[32] Y. Jiang, Z. Lian, Y. Tang, J. Xiao, SCFont: structure-guided Chinese font generation via deep stacked networks, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, pp. 4015–4022.
[33] Y. Wang, Y. Gao, Z. Lian, Attribute2Font: creating fonts you want from attributes, ACM Trans. Graph. 39 (4) (2020) 69.
[34] K. Rayner, Eye movements in reading and information processing: 20 years of research, Psychol. Bull. 124 (3) (1998) 372.
[35] M.W. Eysenck, M.T. Keane, Cognitive Psychology: A Student's Handbook, Taylor & Francis, 2005.
[36] K. Rayner, A. Pollatsek, J. Ashby, C. Clifton Jr., Psychology of Reading, Psychology Press, 2012.
[37] J.-C. Yoo, T.H. Han, Fast normalized cross-correlation, Circuits Syst. Signal Process. 28 (6) (2009) 819.
[38] G. Frank, Pulse Code Communication, US Patent 2,632,058, 1953.
[39] T.H. Kil, W. Seo, H.I. Koo, N.I. Cho, Robust document image dewarping method using text-lines and line segments, in: 14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, IEEE, 2017, pp. 865–870.