A Camera Shooting Resilient Watermarking Scheme for Underpainting Documents
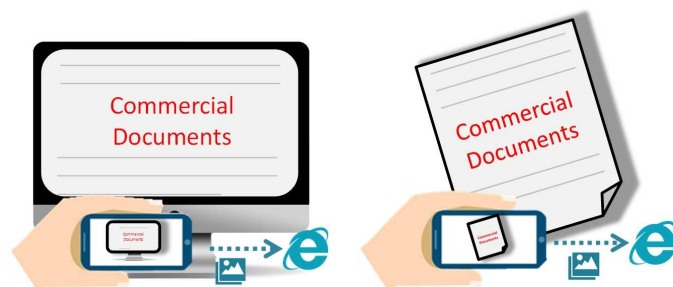
Han Fang, Weiming Zhang, Zehua Ma, Hang Zhou, Shan Sun, Hao Cui and Nenghai Yu

Abstract—This paper designs a novel underpainting based camera shooting resilient (CSR) document watermarking algorithm for dealing with the leak source tracking problem. By applying such algorithm, we can extract the authentication watermark information from the candid photographs. The watermarked underpainting contains three significant properties. 1). Inconspicuousness. The watermarked underpainting is inconspicuous and it will not easily be maliciously attacked. 2). Robustness. We propose DCT-based watermark embedding algorithm and distortion compensation based extracting algorithm, which make the watermark robust to camera shooting process. 3). Autocorrelation. We design the flip-based method to arrange the watermarked underpainting. So that a complete watermark region can be accurately located even if part of the document is recorded. Compared with previous watermarking algorithms, the proposed scheme guaranteed content independent embedding as well as the robustness to the camera shooting process. Besides, the proposed scheme satisfies the accuracy of extraction even when the captured document is incomplete.

Index Terms—Underpainting, document watermarking, camera shooting process, inconspicuousness, robustness, autocorrelation.

I. INTRODUCTION

With the acceleration of the commercialization process, commercial secrets leakage incidents occur frequently, which resulted to the leak source tracking (LST) of the commercial secrets receiving more and more attention [1]. The demands of LST are closely related to the development of multimedia technology. The development of multimedia technology such as audio, image and video technology will bring new challenges to LST methods. As an important means to realize LST, digital watermark algorithm has been paid much attention. And corresponding to different multimedia technologies, there are audio watermark schemes [2], image watermark schemes [3]–[5] and video watermark schemes [6], [7] being proposed. For the need to identify the image source in the cloud, there are also logo watermarking algorithms being proposed [8], [9]. But with the development of digital technology, the multimedia information transmission process undergones great changes, which has created new demands for LST. For traditional ways of stealing information, such as scanning and

All the authors are with CAS Key Laboratory of Electromagnetic Space Information, University of Science and Technology of China, Hefei, 230026, China. (e-mail: fanghan@mail.ustc.edu.cn, zhangwm@ustc.edu.cn, ynh@ustc.edu.cn) Corresponding author: Weiming Zhang.

(a) Sneak shooting the screen. (b) Sneak shooting the paper.

Fig. 1: The information leakage event caused by smart phones.

emailing business documents or copying electronic files, we can trace the leak source by using the traditional robust image watermarking schemes that used for image processing attacks [10], [11]. However, with the popularity of smart phones, photography has become the most simple and effective way of information transmission, which brings new challenges to the LST problem. As shown in Fig. 1, anyone who has access to a file (e.g. authorized employee) can simply leak information by taking a picture without leaving any records [12]. Besides, camera shooting process is not easy to be monitored or prevented from the outside, so designing a camera shooting resilient (CSR) document watermarking scheme is extremely important for solving this issue. CSR document watermarking scheme can act as a powerful guarantee for LST. We can embed information such as device ID or employee ID in the document file. When these files are sneak shot, we can extract the corresponding information from the photo. The leaked equipment or employee number will be located according to the information, so that we can narrow the scope of investigation and then achieve the accountability process. There are three important capabilities should be addressed in realizing such algorithm.

Inconspicuousness. The watermark needs to satisfy inconspicuousness, which contains two aspects, that is, the embedding of the watermark does not affect the normal reading and the watermark cannot be easily identified. The former requirement ensures that the normal use of the host file, and the latter requirement ensures that the watermark will not attract the attention of malicious attackers.

Robustness. Camera shooting process is an air channel information transfer process, which introduces a variety of distortions such as perspective distortion, illumination distortion,

(a) Embedding visible watermarking in background.

(b) Embedding 2-d barcode watermarking.
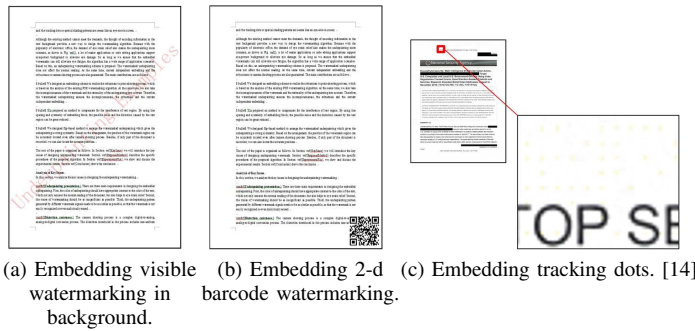
(c) Embedding tracking dots. [14]

Fig. 2: 3 typical solution in realizing LST.

and fall-off effect [13]. Therefore, ensuring that the watermark can survive from the distortions is the primary goal in realizing LST.

Content independence. Content independence means the watermark embedding is free from text content, which is especially important in LST events. Since in actual scenario, the watermarks (*e.g.* employee ID) are different but the documents are the same, so we cannot rely on the content of the document to embed the watermark. Besides, content-independent algorithms require less embedding time than content-related algorithms.

Traditional document watermarking algorithms [15], [16] can be generally divided into three categories: the semantic-based schemes [17]–[19], the structure-based schemes [20]–[22] and the image-based schemes [23]–[25], but neither of them work well for camera shooting process. The semantic-based schemes focus on using the semantic of text to perform the embedding process such as synonym replacement. However, this kind of scheme requires the modification of the source file, which cannot be applied since the word of the document is not able to be changed in official cases. Similarly, the structured-based method cannot be utilized too. Since the structured-based method alters the structure of the text such as spacing of words or punctuation to embed the watermark. The image-based approach treats the document as an image, and the embedding process is done by modifying pixels, but the modification of the spatial domain pixels is easily affected by camera shooting process.

In the camera shooting process, the image as well as the watermark undergoes a series of analog-to-digital (AD) and digital-to-analog (DA) conversion processes which will appear as suffering from a combination of strong attacks [26], and in previous years, the watermarking scheme designed for AD and DA process are extensively studied. Kang *et.al* [27] proposed a print scanning resilient watermarking scheme that embeds the watermark in Fourier-Mellin domain of the image. But it cannot realize content independence since different content will reflect to different Fourier coefficients. Pramila *et.al* [28] proposed a print-cam image watermarking scheme that can achieve the robustness to print shooting process. However, such method is not suitable for documents because it will cause significant visual distortion and the watermark pattern is easily identified. Xiao *et.al* [29] proposed a method of modifying

the shape of the character to embed the watermark. On the extraction side, they designed a corresponding convolution neural network to distinguish the subtle differences of the letter shape to extract the corresponding message. But this approach relies on the establishment of word libraries, but in many cases the software does not open the interface of the word library. David *et.al* [12] proposed a screen watermarking scheme. They embed the encoded pattern with different luminance which is imperceptible to human eyes on the computer screen. However, the algorithm cannot well balance the robustness and the inconspicuousness. Besides, they can only achieve screen shooting traceability but when the document are printed and captured, the algorithm cannot make it.

In our previous work [26], we proposed a screen shooting resilient image watermarking scheme, but compare to the previous work, the main improvements of the proposed scheme are: 1) In [26], watermark locating is based on I-SIFT keypoint, but compared with natural images, the texture and color of the document are more simple, which will result to the weak strength of I-SIFT kepoint. So the keypoint may be lost after camera shooting process when facing textual image, and such locating method cannot well adapt to document files. In this paper, to better synchronize the watermark region, we propose the flip-based method. 2) In document watermarking, embedding efficiency is very important. The method in [26] is content-related, which means that for every page of document, a complete embedding process is required, so it will greatly increases the embedding time. In order to improve the embedding efficiency, we update the embedding algorithm to achieve content-independent embedding in this paper, so that we only need to conduct one embedding process for a whole document. The method in [26] is content-related, which differs from the capability we searching for. So in this paper, we update the embedding algorithm to achieve content-independent embedding. 3) The first step in extraction side of [26] is rescaling the image to its original size, which means that we cannot extract the watermark when only part of the image is recorded. However, this paper improves the ability and achieves the traceability for incomplete documents.

In addition to academic research, the new demand brought by smart phones has attracted a lot of attention in the industry, and many companies have also launched solutions for LST events. There are three typical solutions in realizing LST. The first one, embedding visible watermarking in background. The watermark can be the text or the logo, as shown in Fig. 2a. Though such watermark can be recorded by camera, it will attract the attention of malicious attackers, besides, it is easy to be erased. The second solution is embedding 2-d barcode watermarking. As shown in Fig. 2b, the barcode that contains the device ID or the employee ID are set at the bottom or the head of the paper or the screen, when the barcode are captured, the message can be decoded according to the barcode. However, such barcode is easy to be erased or even tempered. The third method uses the background of the document to embed the watermark, such as tracking dots and special shading patterns [14], [30], as shown in Fig. 2c. The message is encoded in the tracking dots or the shape of the pattern. However, such method cannot well balance the

(a) The Original format in "Word".   (b) The format with underpainting in "Word".

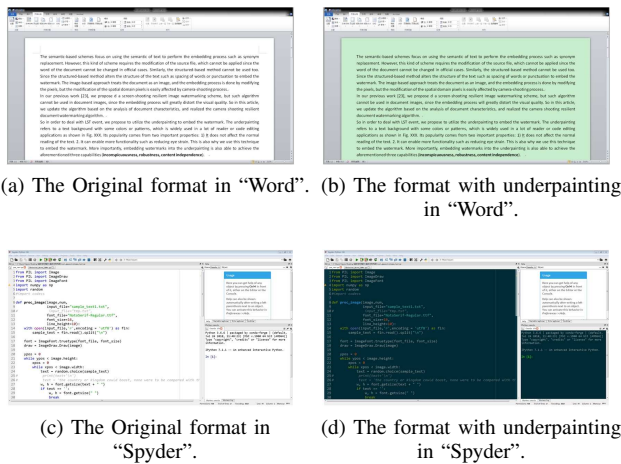(c) The Original format in "Spyder".   (d) The format with underpainting in "Spyder".

Fig. 3: The examples of underpaintings used in Microsoft Word 2010 [31] and Spyder [32].

invisibility and the robustness, especially in camera shooting scene.

In summary, the existing watermarking algorithms cannot be well adapted to camera shooting process. So in order to deal with LST event, we propose to utilize the underpainting to embed the watermark. The underpainting refers to a text background with some colors or patterns, which is widely used in a lot of reader or code editing applications as shown in Fig. 3. Its popularity comes from two important properties: **1)** It does not affect the normal reading of the text. **2)** It can enable more functionality such as reducing eye strain. This is also why we use this technique to embed the watermark. More importantly, embedding watermarks into the underpainting is also able to achieve the aforementioned three capabilities (inconspicuousness, robustness, content independence). The main contributions are as follows:

• We propose an embedding scheme to realize the robustness to camera shooting process, which is based on the analysis of the existing CSR watermarking algorithm. At the same time, we also take the inconspicuousness of the watermark and the rationality of the underpainting into account. Therefore, the features of watermarked underpainting ensures the inconspicuousness, the robustness and the content independent embedding.

• We propose a method to compensate for the interference of text region. By utilizing the line spacing region and symmetry of embedding block, the possible noise and the distortion caused by the text region can be greatly reduced.

• We designed flip-based method to give the underpainting a strong symmetry. Based on the flipping operation, the position of the watermark region can be accurately synchronized even after camera shooting process. Besides, if only part of the document is recorded, we can also synchronize the watermark regions.

The rest of the paper is organized as follows. In Section II we will introduce the key issues of designing underpainting watermark. Section III and Section IV describes the specific procedures of the proposed algorithm. In Section V, we show and discuss the experimental results. Section VI draws the conclusion.

II. ANALYSIS OF KEY ISSUES

In this section, we analyze the key issues in designing the underpainting watermarking.

Underpainting presentation. There are three main requirements in designing the watermarked underpainting. First, the color of underpainting should have appropriate contrast to the color of the text, which not only ensures the normal reading of the document, but also helps to relieving eye fatigue. Second, the watermark should be as invisiable as possible. Third, the underpainting pattern generated by different watermark signals needs to be as similar as possible, so that the watermark is not easily recognized or even maliciously erased.

Distortion resistance. The camera shooting process is a complex digital-to-analog, analog-to-digital conversion process. The distortion introduced in the process includes nonuniform illumination distortion, fall-off distortion, noise distortion, post-processing distortion, etc. [13], [33]. Watermarking algorithms need to be robust enough to resist all distortions caused by this process.

Automatic locating. Since in the actual scene, it is very likely that only a part of the document is captured, which means that we cannot determine the relative position of the watermark block through the document boundary. So the automatic locating of the watermark region is very important in achieving extractability in partial document, which is an important feature to put the algorithm into practical use.

Compensation for text regions. The existence of text regions will cause two main influences on the extraction process. First, the format of the text interferes with the locating of the watermark block. Since we need to repeatedly embed a lot of complete watermark information in one document, the watermarked underpainting also has a fixed arrangement. How to ensure the locating process not being affected by the text format is an important part of the algorithm. Second, the text region will influence the watermark information extracting. Since we embed the watermark in the underpainting without modifying the text region, the text region itself is an irreversible distortion. How to compensate for the distortion caused by the superposition of text region is another problem we need to pay attention to.

To obtain a more suitable color of the underpainting, we designed an optimization method. By solving the optimization problem, we will generate an underpainting with appropriate contrast to text color, while also retaining the original texture as much as possible.

For the purpose of inconspicuousness and robustness, based on the previous work [26], we update the DCT-based watermark embedding method to make it suitable for document embedding.

In order to automatically locate the watermark region, we proposed a flip-based method which gives the watermark a strong autocorrelation. Based on the autocorrelation, the watermark region can be accurately located after camera shooting process.
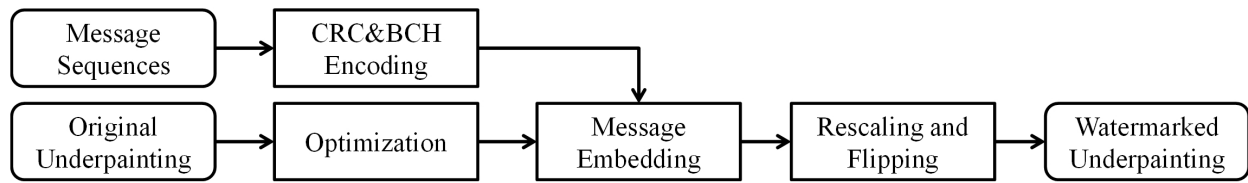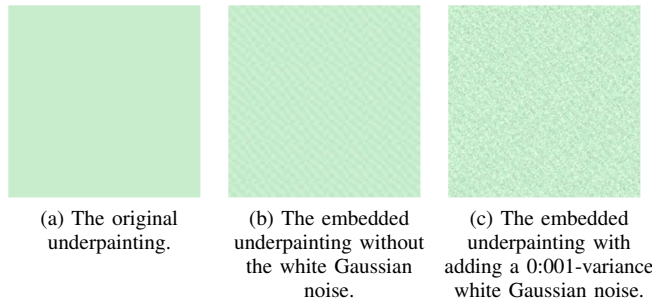
Fig. 4: The framework of the embedding process.

(a) The original underpainting.

(b) The embedded underpainting without the white Gaussian noise.

(c) The embedded underpainting with adding a 0:001-variance white Gaussian noise.

Fig. 5: The underpainting with and without white Gaussian noise.

(a) The spatial image after embedding bit '0' in [28].

(b) The spatial image after embedding bit '1' in [28].

(c) The spatial image after embedding bit '0' with r = 30.

(d) The spatial image after embedding bit '1' with r = 30.

Fig. 6: The embedded underpaintings with different message bit.

To cope with the embedding method, we proposed a compensation-based extracting scheme that can replace the most text regions by watermark regions. Therefore, in the overall extraction process, the watermark can be accurately extracted.

III. EMBEDDING PROCEDURE OF THE PROPOSED METHOD

A. Framework of embedding procedure

Fig. 4 shows the framework of the embedding process. For an underpainting, we need to modify the color to meet the requirements by solving an optimization problem if the original one is not suitable. Then, the encoded watermark sequences is embedded to the underpainting. After that, in order to reduce the influence of the text regions, we need to resize the unit block according to the format of the text. At the same time, the flip-based method is applied on the resized block to construct the row/column symmetry of the underpainting. Finally, the watermarked underpainting is generated by paving the flipped underpainting to the actual document size.

B. Optimization of underpainting

As mentioned before, the color of underpainting needs to have a certain contrast with the text color. But the contrast is not the larger the better, considering that if the color contrast between the underpainting and the text is too large, the watermark details in the underpainting may not be captured by the camera, so it cannot be too large. We summarized the color selection problem of the underpainting as an optimization problem. We use the luminance component (*i.e.* Y-channel image of YCbCr colorspace) of the underpainting as the constrained object. To ensure the normal reading, the luminance of the underpainting needs to have a lower bound, while to ensure the watermark detail to be captured, the luminance of the underpainting needs to have a higher bound. At the
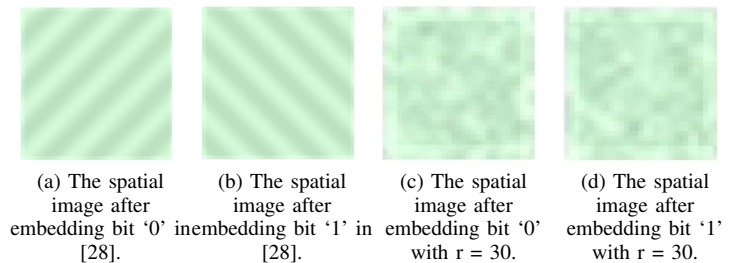
same time, we should guarantee that the original texture of the underpainting changes as little as possible. When the text is black, the whole optimization process can be formulated by

$$\min_{P_o'} \|P_o - P_o'\|_2^2$$
$$\text{s.t.} \quad th_1 \leq Y(P_o') \leq th_2 \tag{1}$$

where P_o denotes the original underpainting, P_o' denotes the optimized underpainting and $Y(P_o')$ represents the luminance component of P_o'. After iteration, we can optimize the underpainting into the color we need.

But only optimizing the color is not enough, since the embedding process will produce certain texture in the monochrome background, which may affect the visual quality, so before embedding, we add the white Gaussian noise with variance 0.001 into the underpainting, as a result, the texture generated by the embedding process will be invisible. As shown in Fig. 5, after adding Gaussian noise, the underpainting appears more random.

C. Message Embedding

For a binary watermark sequence, we first use the cyclic redundancy check (CRC) code [34] to encode the original sequence, then we use the BoseChaudhuriHocquenghem (BCH) code [35] to encode the sequence to generate a sequence with error correction and check capability, and the length of the final sequence is l. After encoding, we fill it into a binary matrix W of size $a \times b$ by column, as shown in Fig. 7a. Note that $a \times b$ should be no less than l, and in order to reduce redundancy bits, it cannot be greater than l too much. At the same time, a and b should satisfy the requirements we developed. In this paper we set a as two to three times of b, because when shooting angles are skewed, only the row part or the column part is well recorded, we must ensure the information still extractable under this situation. So if a is twice or three times as b, the whole watermark region can be

obtained even the shooting angle is skewed. So the selection of a and b can be formulated by

$$\min \quad a \times b - l$$
$$\text{s.t.} \quad 2 \times a \leq b \leq 3 \times a \tag{2}$$

Since the length of message sequence is 128, so an exhaustive search for Eq. (2) with $l = 128$ yields a different set of a and b, we begin with $b = 1$, $a = 2$ and end with $b = 8$, $a = 16$. So in this paper, a is set as 16, and b is set as 8. Then we divide the watermarked underpainting into 2 parts: Part 1 and Part 2 with the same size. Then we spliced Part 2 and Part 1 below the underpainting to generate the final underpainting unit $\mathbf{P}$, as shown in Fig. 7a.

After that, we apply the embedding procedure. There are 2 requirements we need to meet: **1)** The robustness against camera shooting process. **2)** The consistency of underpainting after embedding different watermarks. To realize the robustness against the camera shooting process, [36] and [28] proposed corresponding algorithms. A typical feature of the algorithms is that messages are represented by templates with different directional features, and the watermark is embedded by superimposing these templates into the image. The message is extracted by judging the direction of these templates. The effectiveness of such schemes demonstrates that the directional features of the template are robust to the camera shooting process, but we cannot directly apply this scheme in documents since the pattern is too obvious to be recognized or even erased. However, we find that the directionality of the spatial template is closely related to the relative magnitudes of a pair of DCT coefficients [26]. The relative magnitudes will not change before and after the camera shooting process. So we can embed the watermark by changing the relative magnitudes of different DCT coefficients. More importantly, by applying this embedding algorithm, the spatial blocks corresponding to bit-0 and bit-1 have high similarity. In a 16×16 pixels block, when embedding bit '0' and bit '1' by Eq. (3), we find that the spatial images generated by different bits are roughly the same as a noised block, we cannot easily distinguish the '0' block and the '1' block. By comparison, the block embedded by [36] is more obvious, as shown in Fig. 6. This means whatever the watermark is, the watermarked underpainting looks like the same. So embedding in this way not only ensures the consistency, but also satisfies the robustness to the camera shooting process.

In the proposed algorithm, 1-bit information is embedded in a block with size of 16×16 pixels. So for a optimized underpainting P'_o, we should resize and convert it into $a \times b$ blocks with size 16×16, where $a \times b$ represents the watermark size. For each block, we perform DCT on it and select the middle-frequency coefficients C_1 and C_2. Then according to the watermark bit w, we do the following operations to embed the watermark.

$$\begin{cases} C_1 = r, C_2 = -r & if \ w = 0 \\ C_1 = -r, C_2 = r & otherwise \end{cases}, \tag{3}$$
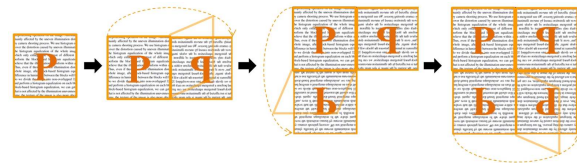
where r represents the embedding intensity. Then we perform IDCT to each block. The operation above is repeated until all the watermark bits are embedded.
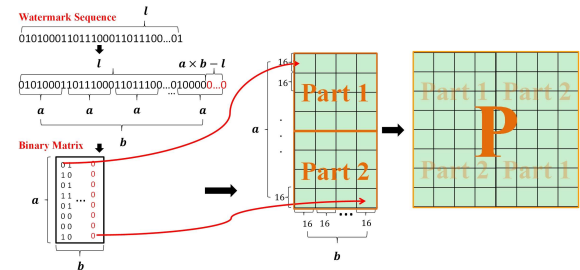
(a) The relationship between the watermark matrix and the underpainting.
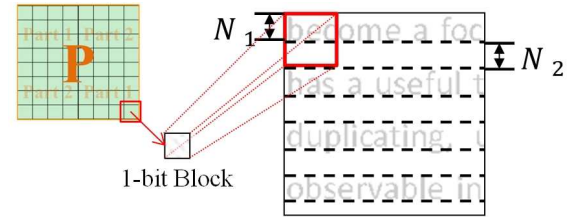
(b) The adjusted size of 1-bit block by fully using line spacing.

(c) The arrangement of the embedded underpainting unit.

Fig. 7: The settings of the underpainting when embedding the watermark.

D. Rescaling and flipping

After embedding the watermark, we need to rescale the underpainting according to the width of the text and the width of the line spacing. Since the underpainting is at the bottom layer, the text itself will cause irreversible distortion to the watermarked underpainting. Watermark information will no longer be remained in the region covered by the text, but the line spacing region can still retain the watermark information. In order to reduce the influence of text region, we need to make full use of the line spacing. As shown in Fig. 7b, the width of 1-bit watermarked block needs to be adjusted from 16 to $(N_1 + N_2)$, where N_1 denote the width of the text, and N_2 denote the width of line spacing. Thus, we can ensure that in one bit watermark unit, both text region and line spacing region are included. Furthermore, the block embedded by Eq. 3 will be symmetric along the diagonal, so in extraction side, most text regions in the watermark unit can be replaced by the line spacing regions in the symmetrical position. This operation greatly reduces the impact of text region on the extraction. The specific procedure will be introduced in Section IV-D.

When rescaling finished, we need to construct the properties of row symmetry and column symmetry on underpainting to facilitate the synchronization of watermarking regions. Since we will embed many complete watermarking units in a document, as long as we can synchronize to one complete watermarking unit, we can extract messages. The synchronization operation is based on the symmetry of the underpainting we constructed, by detecting the corresponding position of the
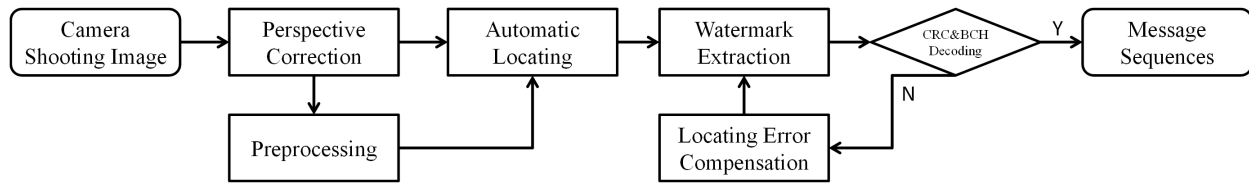
Fig. 8: The framework of the extracting process.

peak value about row/column symmetry, we can synchronize the region of a complete watermarking unit. The procedure can be regarded as a flip operation, as shown in Fig. 7c, which can be formulated by

$$
\begin{aligned}
P_f(x, y) &= P_f(2 \times N - x, y) \\
&= P_f(x, 2 \times N - y) \\
&= P_f(2 \times N - x, 2 \times N - y) \\
&= P(x, y)
\end{aligned}
\tag{4}
$$

where P_f denote the underpainting after flipping, $N = N_1 + N_2$, and (x, y) denote the coordinates of the underpainting. The final underpainting will be generated by paving P_f according to the actual document size. The embedding process is described as Algorithm 1.

Algorithm 1 Embedding Algorithm

Input: Original underpainting P_0, watermark sequence m, the width parameter N.

Output: Watermarked underpainting P_f .

1: Optimize P_0 with Eq. (1) to get the optimized underpainting P_0'.
2: Encode m with CRC and BCH code and generate the binary matrix W.
3: Embed W into P_0' with Eq. (3) to get embedded underpainting P.
4: Rescale P according to N and perform flipping operation with Eq. (4) to generate the underpainting P_f .
5: **return** Watermarked underpainting P_f .

IV. THE EXTRACTION PROCEDURE OF THE PROPOSED METHOD

A. Framework of extracting procedure

The extracting process can be described as Fig. 8. For a camera shooting image, the perspective transform is first performed to correct the lens distortions according to the content of the text. In the corrected image, we need to locate the watermark region according the symmetry of the watermarked underpainting. Then we extract the watermark from the located region.

B. Perspective distortion correction

Official documents or web texts generally have a fixed format, which means that we can circle a rectangular by the text alignment, as shown in Fig. 9. Then according to the vertex of the rectangular, we can correct the document image through perspective transformation. The correction process
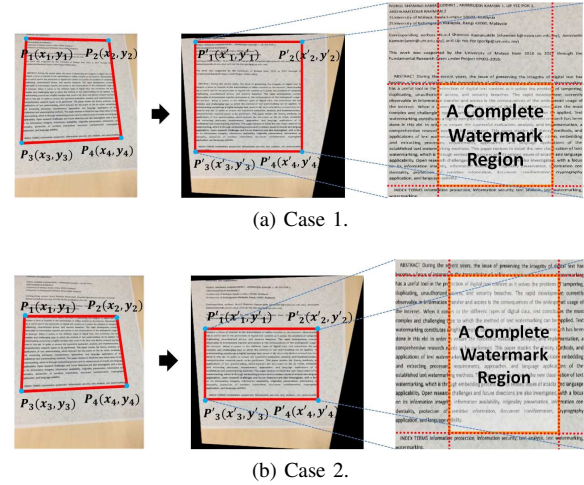

(a) Case 1.

(b) Case 2.

Fig. 9: The perspective correction and watermark region synchronization when selecting different P_1, P_2, P_3 and P_4.

is as follows: We first circle a rectangular in the distorted image and obtain the 4 vertices $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, $P_3(x_3, y_3)$ and $P_4(x_4, y_4)$, then we set the transformed coordinates corresponding to the 4 vertices as $P_1'(x_1', y_1')$, $P_2'(x_2', y_2')$, $P_3'(x_3', y_3')$ and $P_4'(x_4', y_4')$, in this paper, we set $P_1'(1, 1)$, $P_2'(1, 2048)$, $P_3'(2048, 1)$ and $P_4'(2048, 2048)$. Substituting the 8 coordinates into Eq. (5), the value of $a_1, b_1, c_1, a_0, b_0, a_2, b_2, c_2$ can be calculated.

$$
\begin{aligned}
x' &= \frac{a_1 x + b_1 y + c_1}{a_0 x + b_0 y + 1} \\
y' &= \frac{a_2 x + b_2 y + c_2}{a_0 x + b_0 y + 1}
\end{aligned}
\tag{5}
$$

Thus we can generate a mapping from the distorted image to the corrected image. It is worth noting that P_1, P_2, P_3 and P_4 are not fixed. As long as they are surrounded by a rectangular area, the selection of these 4 points meets our requirements. The extracted image I' can be obtained by cropping the area enclosed by P_1', P_2', P_3' and P_4'. But we cannot extract the watermark directly from the image I' because as shown in Fig. 9: 1) Compare to the original image, the size of I' is scaled. Because the size of the corrected image I' is determined by P_1', P_2', P_3' and P_4', which is set by ourselves, and it is not the same as that in the original image. We still don't know the specific location of the watermark region in the corrected image I', different selection of P_1, P_2, P_3 and P_4 will produce different I', and correspond to different watermark region. So after obtaining I' we need to synchronize the watermark region.
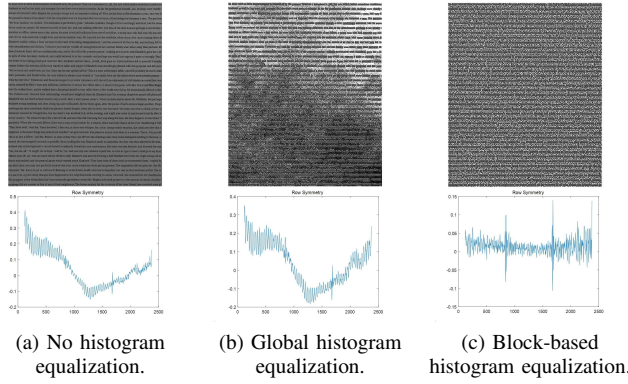
(a) No histogram equalization.　(b) Global histogram equalization.　(c) Block-based histogram equalization.

Fig. 10: The preprocessed image and its' corresponding row symmetry waveforms.

C. Preprocessing

In order to automatically synchronize the area of the complete watermark region, we need to perform preprocessing operation on the image. Automatic locating is mainly affected by the uneven illumination distortion in camera shooting process. We use histogram equalization to cover the distortion caused by uneven illumination. However, the histogram equalization of the whole image is useless, which only enlarges the contrast of different areas, so we perform the block-based histogram equalization, since we believe that the illumination is uniform within a small block. We divide the image into non-overlapped 32×32 blocks and perform a histogram equalization on each block. After the block-based histogram equalization, we can get an image I_p that is not affected by the illumination unevenness. At the same time, the texture of the image is also more obvious, which is beneficial to the locating operation, to better indicate the impact of block-based histogram, we conduct the corresponding experiment. After perspective correction, we perform the following three operations on the corrected image: 1). No histogram equalization, 2). Global histogram equalization, 3). Local histogram equalization, the result are shown in Fig. 10, after block-based histogram equalization, the texture of the underpainting can be greatly highlighted and the peak of row symmetry waveform can be easily filtered.

D. Automatic locating

The whole watermark unit is located based on the preprocessed image I_p. By detecting the symmetry of I_p about rows and columns, we can locate the watermark regions. The locating operation is mainly divided into three steps. a). Symmetry peak detection. b). Text format and noise compensation. c). Peak filtering.

For step a), we need to perform symmetry detection by row and column separately. Taking column detection as an example, as shown in Fig. 11, for the image to be detected I_p with size $m \times n$ (denote as $I^{m \times n}$), we carry out the symmetry detection operation for each column. So when detecting column $j(j \in [1, n])$, $I^{m \times n}$ will be divided into

the following three parts

$$
I^{m \times n} = \begin{cases} [J_1^{m \times d_j} J_2^{m \times d_j} J_3^{m \times (n - 2 \times d_j)}], & if \ j \leq (n+1)/2 \\ [J_3^{m \times (n - 2 \times d_j)} J_1^{m \times d_j} J_2^{m \times d_j}], & otherwise \end{cases},
$$

(6)

where

$$
d_j = \begin{cases} j, & if \ j \leq (n+1)/2 \\ n - j + 1, & otherwise \end{cases}.
$$

(7)

The symmetry of column j is measured by the similarity of the matrices $J_1^{m \times d_j}$ and $\Phi_{column}^{(J_2^{m \times d_j})}$ where Φ_{column} represents column flip operation, which is defined as

$$
\Phi_{column}^I(x, y) = I(x, n - y + 1).
$$

(8)

Then we zero-centered J_1 and $\Phi_{column}^{J_2}$ by Eq. (9).

$$
C(I) = \frac{I - E(I)}{\sqrt{D(I)}}
$$

(9)

where $E(I)$ denotes the mean value of the matrix I, $D(I)$ denotes the variance of the matrix I. And the similarity can be calculated by

$$
S_{column}(j) = \frac{\sum_{x=1}^{m} \sum_{y=1}^{d_j} C(J_1) C\left(\Phi_{column}^{J_2}\right)}{m \times d_j}
$$

(10)

Similarly, we can define the symmetry of row i as

$$
S_{row}(i) = \frac{\sum_{x=1}^{d_i} \sum_{y=1}^{n} C(I_1) C\left(\Phi_{row}^{I_2}\right)}{d_i \times n}
$$

(11)

where

$$
I^{m \times n} = \begin{cases} [I_1^{d_i \times n} I_2^{d_i \times n} I_3^{(m - 2 \times d_i) \times m}]^T, & if \ i \leq (m+1)/2 \\ [I_3^{(m - 2 \times d_i) \times m} I_1^{d_i \times n} I_2^{d_i \times n}]^T, & otherwise \end{cases},
$$

(12)

$$
d_i = \begin{cases} i, & if \ i \leq (m+1)/2 \\ m - i + 1, & otherwise \end{cases},
$$

(13)

$$
\Phi_2^I(x, y) = I(m - x + 1, y).
$$

(14)

When i and j are on the axis of symmetry, the value of S_{column} and S_{row} will exhibit higher values. It is worth noting
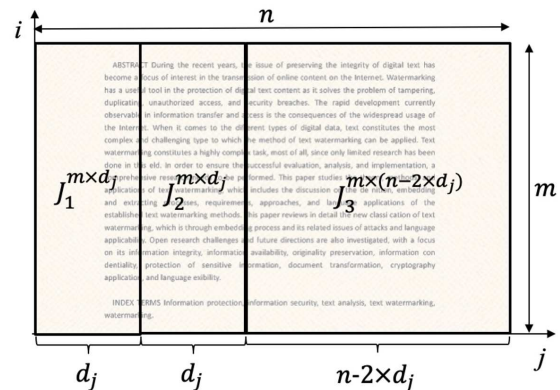

Fig. 11: The example on automatic locating.

(a) The original waveform of row symmetry.
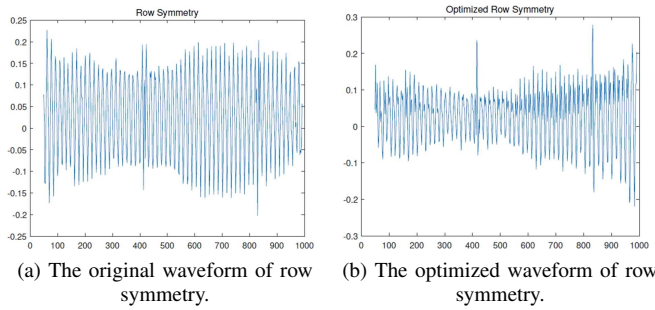
(b) The optimized waveform of row symmetry.

Fig. 12: The waveform before and after optimization.

that the step size of i and j does not need to be one pixel, in this paper, we choose 7 pixels as a step.

For step b), the text format itself as well as the noise caused by camera shooting process will affect the row symmetry measurement, which behaved in waveform S_{row} to be no peak value at any point, as shown in Fig. 12a. So we should reduce the influence of the text format and the channel noise by applying an optimization method. After calculating the symmetry signal (denoted as S_{row}), we need to use OCR [37] to detect and retain the text part of the camera shooting image to form the image I_t, which only contains the text region. And then we perform the automatic locating operation to I_t to get the symmetry signal of I_t (denoted as S_{row}^t). The relationship between S_{row} and S_{row}^t can be written as

$$S_{row} = S_{row}^t + S_{row}^u + N \tag{15}$$

where S_{row}^u denote the symmetry signal of the underpainting and N denote the the symmetry signal of the noise caused by camera shooting process. In the experiment, we use the additive white Gaussian noise (AWGN) N_g to simulate N. Note that there must be a certain deviation between the actual noise N and the simulated noise N_g, so we use coefficient k to make compensation. So S_{row}^u can be written as

$$S_{row}^u = S_{row} - S_{row}^t - k \times N_g \tag{16}$$

According to Eq. (11) - Eq. (14), we have

$$S_{row}^u(i) = \frac{\sum_{x=1}^{d_i} \sum_{y=1}^{n} C(I_1^u) C\left(\Phi_{row}^{I_i^u}\right)}{d_i \times n} \tag{17}$$

where I_1^u and I_2^u denote the underpainting without text. Only when i is on the symmetry axis, $C(I_1^u)$ and $C(\Phi_{row}^{I_2^u})$ are equal, otherwise, they are independent with each other. So when i is not on the symmetry axis, $S_{row}^u(i)$ satisfies the distribution with mean value of 0, since the mean value of $C(I_1^u)$ and $C(\Phi_{row}^{I_2^u})$ are all equal to 0 and they are independent. So the mean value of distribution about S_{row}^u satisfy

$$E(S_{row}^u) = \begin{cases} 1 & \text{On symmetry axis} \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where $E(S_{row}^u)$ denote the mean value of the distribution about S_{row}^u. So $E(S_{row}^u)$ should be approximately equal to 0 since the symmetry axis only accounts for a small percentage. That

is, $S_{row} - S_{row}^t - k \times N_g$ should approximate to 0. So the optimization equation can be written as

$$\min_k |E(S_{row} - S_{row}^t - k \times N_g|$$

where $E(S_{row} - S_{row}^t - k \times N_g)$ denote the mean value of $S_{row} - S_{row}^t - k \times N_g$. Once we determine the value of k, we can approximate the signal S_{row}^u. Fig. 12 indicates an example of the symmetry waveform before and after the optimization. It can be clearly seen that after optimizing, the peaks become sharper and easier to filter out.

For step c), we need to eliminate the interference caused by the noise on the detection procedure. As shown in Fig. 13, when the noise is strong, it will produce a lot of peaks with larger variance, which is not the peak we need to filter out. The value on the axis of symmetry behaves a very high peak with a relatively gentle jitter around the peak. So based on the peak and standard deviation of the waveform, we designed the filter method. Taking the column symmetry waveform as an example, we divide the generated entire waveform S_{column} into k parts, note as $S_c^1, S_c^2 \ldots S_c^k$ and perform peak filter operation in each part. For the i-th part S_c^i, we record the corresponding position L_c^i of the peak. Besides, we calculate the standard deviation σ_c^i of data in S_c^i except $S_c^i(L_c^i)$ (i.e. $S_c^i \backslash S_c^i(L_c^i)$) by Eq. (21). We define the score of $S_c^i(L_c^i)$ as V_c^i, which can be formulated by

$$L_c^i = \arg\max_j S_c^i(j) \tag{19}$$

$$\mu_c^i = \frac{\sum_{j=0}^1 |S_c^i(L_c^i + j) - S_c^i(L_c^i + j - 1)|}{2} \tag{20}$$

$$\sigma_c^i = \sqrt{\frac{1}{M-1} \sum_{j=1}^{M-1} (S_c^i(j)\backslash S_c^i(L_c^i) - E(S_c^i\backslash S_c^i(L_c^i)))^2} \tag{21}$$

$$V_c^i = \mu_c^i - \sigma_c^i \tag{22}$$

where M denote the number of points in S_c^i and $E(S_c^i\backslash S_c^i(L_c^i))$ denote the mean of $S_c^i\backslash S_c^i(L_c^i)$ and σ_c^i denote the standard deviation of $S_c^i\backslash S_c^i(L_c^i)$. So when k parts are filtered, we can get the coordinates of k peak points and their corresponding scores $V_c^1, V_c^2, \ldots, V_c^k$. We sort the k points in descending order of their scores and then take the first n_c points $L_c^{k_1}, L_c^{k_2}, \ldots, L_c^{k_{n_c}}$ as candidate points of column symmetry axis. Similarly, we can get n_r candidate points of row symmetry axis $L_r^{k_1}, L_r^{k_2}, \ldots, L_r^{k_{n_r}}$. Note that the value of nc and nr are determined by the number of whole watermark units. Then we choose the adjacent $\left\{L_c^{k_j}, L_c^{k_{j+1}}\right\}$ and adjacent $\left\{L_r^{k_j}, L_r^{k_{j+1}}\right\}$ to form 4 coordinates as the boundary points of complete watermark region. And the region B is cropped from the corrected image I' for message extraction.

E. Watermark Extracting

The watermark extraction operation will be operated in the cropped region B. But before extracting, we need to compensate the distortion of text region. As shown in Fig. 14, since we have ensured the embedding block containing
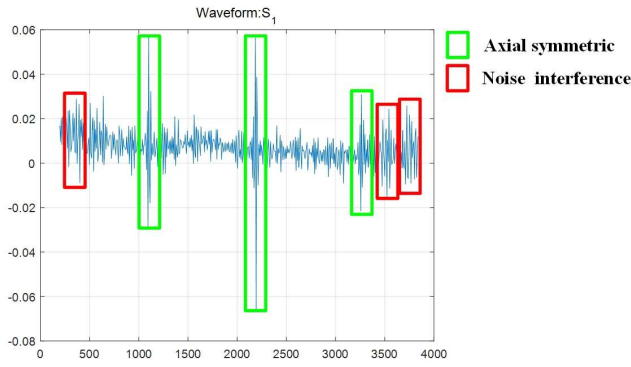
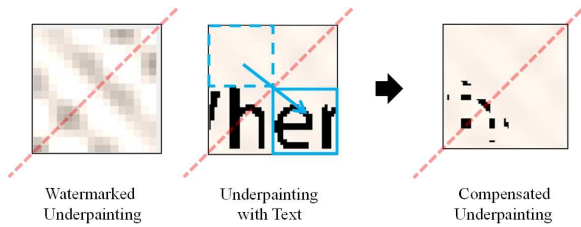Fig. 13: The waveform of S_{column}.

Fig. 14: The compensation process in text region.

both the line spacing and the text region and the watermarked underpainting is symmetric along the diagonal. So the text region can be replaced by the diagonal region of it to recover the watermark signal. Firstly, we binarized the image I_t to generate I_B with black text and white background. Then we take out the image block B0 corresponding to B in the binary image I_B. After that, we resize B and B' to $a \times b * (64 \times 64)$ pixels. Note that the camera shooting process will bring noise distortion to the image, so we cannot directly resize the block to $a \times b * (16 \times 16)$ pixels, which only enlarge the noise component. In order to reduce the noise component, we first resize the block to $a \times b * (64 \times 64)$ pixels and perform the compensate operation, and then resize them into original sizes by each block. For resized B and B', we divide it into non-overlapping $a \times b$ blocks with size (64×64) pixels. Denote one single block as D and D'. For each block D, we replaced the text region in D (the black part of D') by the diagonal symmetrical part of them in D, as Eq. 23 shown.

$$D(x,y) = \begin{cases} D(64-x, 64-y), & \text{if } D'(x,y) = 0 \\ & \& D'(64-x, 64-y) = 1 \\ E(D), & \text{otherwise} \end{cases}$$

(23)

Its worth noting that for the part where $D'(x,y) = 0$ and $D'(64-x, 64-y) = 0$, we compensate it for the mean value of this block. Then the extraction process is carried out on the compensated block. For the compensated block D, we resized it into 16×16 pixel and select its DCT coefficients C_1 and C_2 to do the following extraction

$$w = \begin{cases} 0 & if \ C_1 \geq C_2 \\ 1, & otherwise \end{cases}$$

(24)

The operation is repeated until l bits watermark are extracted.

F. Locating Error Compensation

After l bits watermark extracted, we perform BCH to decode it. Then the CRC is performed on the extracted watermark sequence. If no CRC error detected, the extraction operation is completed. Otherwise, we need to perform a small traversal of the locating point with range st_1 and st_2. Then we perform the perspective transform according to the new points to get the corrected block B and B' and carry out the extracting procedure until all point jitter is completed or no CRC error detected. If the extraction is still not successful, we need to select the next pair of points $\{L_1^{k_j+1}, L_1^{k_j+2}\}$ and $\{L_2^{k_j+1}, L_2^{k_j+2}\}$ for the overall extraction process.

Its worth noting that the extraction accuracy is sensitive to locating accuracy, so the traversal process is important to cover the locating distortion. The extracting process is described as Algorithm 2.

Algorithm 2 Extracting Algorithm

Input: Camera shooting photo I_{ph}.
Output: Watermarked sequence m.
1: Perform perspective correction according to the text alignment in I_{ph} to get the image to be extracted I'.
2: Conduct block-based histogram equalization on I' to get the preprocessed image I_p.
3: Synchronize the watermark region B according to Eq. (6) - Eq. (22).
4: Extract the watermark with I', B and Eq. (23) - Eq. (24).
5: Compensate the synchronization error and decode the sequence with CRC and BCH to get the extracted sequence m.
6: **return** Watermark sequence m.

V. Experimental Results and Analysis

In this section, we show and discuss the experimental results in visual quality and the robustness. In our experiment, the size of the underpainting we choose is 256×256 pixels and we choose $a = 16$, $b = 8$. The coefficients C_1 and C_2 we chose are $(4, 5)$ and $(5, 4)$. The error correction code (ECC) we choose is BCH(127,64) which can correct 10 bit errors while the length of CRC bits is 17 bits. So the actual message bits are 48 bits.

The monitor we used for screen shooting experiment in Section V-B is 'Lenovo-P22i', the printer we used for print shooting experiment in Section V-C is 'HP OfficeJet Pro 8720' and the mobile phone we used is 'iPhone 6s'. For each underpainting, we generated 10 different documents with the text that randomly selected from the book "Pride and Prejudice" [38], and the text format is: 16 pixel for text, 10 pixel for line spacing. The underpainting layout are shown as Fig. 15a, in the experimental format, we can detect 2 axes of symmetry in the column and 2 axes of symmetry in the row. The experimental system are shown as Fig. 15b and Fig. 15c. In Section V-A, we show and discuss the effects of different underpainting on visual quality. In Section V-B

and Section V-C, the robustness tests are performed on the proposed scheme via different shooting conditions. In Section V-D, the algorithm are applying to different devices to test the universality.

A. The visual quality with different underpainting setting

Fig. 16a shows the original document, Fig. 16b and Fig. 16c shows the embedded documents with embedded underpainting that used for camera shooting test. Fig. 16d indicates embedded documents with white underpainting for print shooting test. We can see that the embedding process does not cause serious visual distortion from the original documents to the embedded documents. But we can see in Fig. 16c, the watermarking trace in the underpainting of the light black is more noticeable than that in light green underpainting. So when facing darker background, the embedding intensity should be weaker.

To better quantify the visual quality of watermarked documents embedded with different approaches, in addition to calculating PSNR values, we also perform a mean opinion score (MOS) test. Specifically, 28 raters are asked to assign an integral score from 1 (bad quality) to 5 (excellent quality) to the watermarked documents. The raters rated 10 document images with each method ([12], [28], [39] and the proposed method). The document format used for comparison test is the same, and all the text is randomly selected in [38]. The experimental results of MOS tests are summarized in Table I.

TABLE I:
THE ASSESSMENT OF VISUAL QUALITY OF WATERMARKED DOCUMENTS WITH DIFFERENT SCHEMES.

Assessment	David et.al [12]	Pramila et.al [28]	Yang et.al [39]	Proposed green	black	white
PSNR(dB)	33.51	32.07	32.13	31.40	33.35	32.57
MOS	2.39	2.53	1.21	4.64	4	4.71

From Table I we can easily find that even under the same level of PSNR values, the MOS score of proposed method is much better than that of other schemes. Since we realize content independence embedding, we do not make any modifications to the text itself, while other algorithms will modify the text when embedding the watermark. So in terms of text clarity, our algorithm has more advantages, which is one of the reasons that our MOS score is better than other schemes.

after embedding the information, we divide the watermarked underpainting into 2 parts $Part\ 1$ and $Part\ 2$ with the same size. Then we spliced $Part\ 2$ and $Part\ 1$ below the underpainting to generate the final underpainting P.

B. The robustness tests in screen shooting process

We show the robustness performance of the proposed algorithm in the screen shooting process from multiple perspectives. The underpaintings' color we used are the light green (RGB-[199,237,204]) and the light black (RGB-[40,40,35]), as shown in Fig. 16b and Fig. 16c, which are commonly used. We compare the proposed scheme with the watermarking

(a) The layout of underpainting in the experiment.

(b) The system of screen-shooting process.

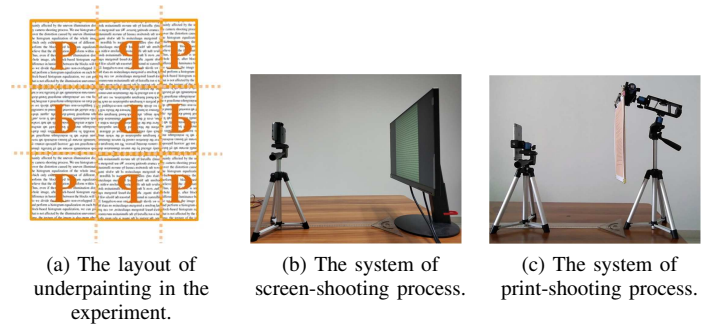(c) The system of print-shooting process.

Fig. 15: The experimental system of robustness tests.

scheme in [12], which is designed for screen shooting process, the watermarking scheme in [28], which is designed for print shooting process and the watermarking scheme in [39], which is designed for print scanning process. For fair comparison with other schemes, all the embedded documents' PSNR are set to the same level of 32-33dB, and the embedding documents are shown as Fig. 16. The average erroneous bits (EB) is utilized to measure the robustness of the algorithm in different shooting conditions.

TABLE II:
ERRONEOUS BITS OF THE EXTRACTED WATERMARKS WITH DIFFERENT SHOOTING DISTANCE.

Distance(cm) Erroneous Bits	David et.al [12]	Pramila et.al [28]	Yang et.al [39]	Proposed green	black
15	-	-	-	**5.1**	**9.9**
25	-	-	-	**9.8**	**9.8**
35	16.4	10.5	15.8	**9.9**	**8.2**
45	21.3	11.4	22.4	**9.0**	**9.1**
55	20.6	13.7	32.1	**7.6**	**9.5**
65	24.2	17.8	35.2	**9.2**	**9.6**
75	22.1	18.4	37.1	**9.9**	**9.9**
85	22.3	19.1	39.2	**9.2**	**9.3**
95	23.5	19.5	43.9	**13.3**	**15.5**

TABLE III:
THE EXAMPLE OF SCREEN-SHOOTING DOCUMENTS WITH DIFFERENT DISTANCE.

Distance	15cm	45cm	65cm	85cm	95cm
Recaptured					
Recovered					

1) The Impact of Distance on Robustness: Table II and Fig. 17a indicates the EB with different underpaintings at different recapture distances. Table III shows the examples of recaptured images in different distance and the corresponding recovered images. It's easy to see that the proposed algorithm has a better performance than the scheme in [12]. The EB are at least 7 bits
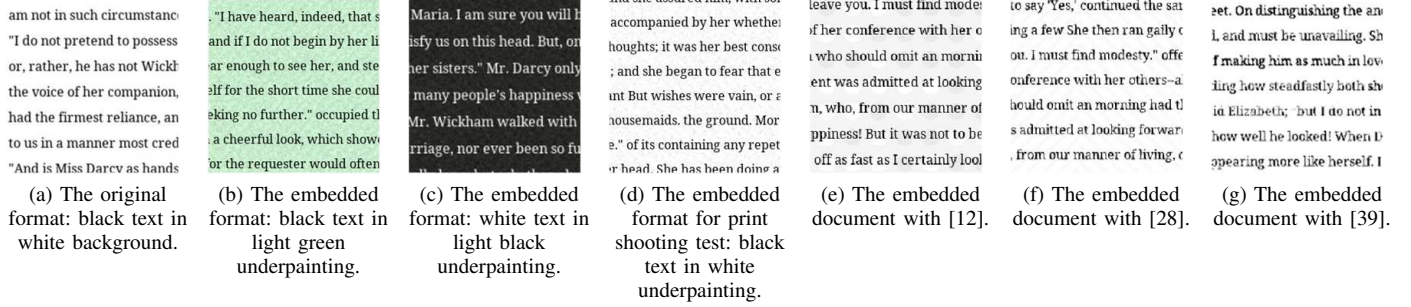
(a) The original format: black text in white background.

(b) The embedded format: black text in light green underpainting.

(c) The embedded format: white text in light black underpainting.

(d) The embedded format for print shooting test: black text in white underpainting.

(e) The embedded document with [12].

(f) The embedded document with [28].

(g) The embedded document with [39].

Fig. 16: The visual quality of the embedded underpainting with text.

(a) BER of different distance.

(b) BER of different horizontal perspective angles.

(c) BER of different vertical perspective angles.
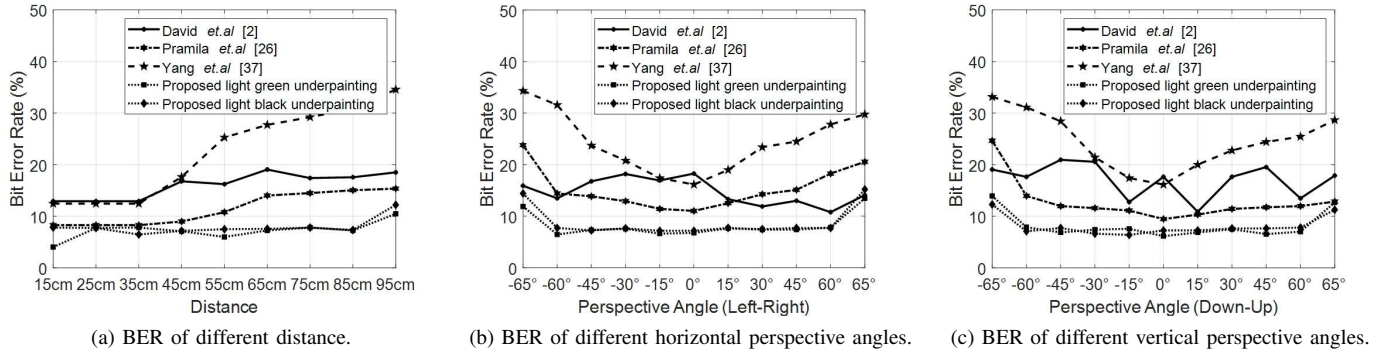
Fig. 17: BER of different shooting conditions.

lower in all distance. And in the case of shooting at 15cm and 25cm, the mobile phone cannot capture the whole screen, so the algorithm in [12] and [28] cannot extract effectively. From Fig. 17a we can see that both the light green underpainting and the light black underpainting can be well suited in the distance of 15cm-85cm, since the designed BCH code can correct 10-bit error. So the algorithm is robust to distance changing.

TABLE IV:
ERRONEOUS BITS OF THE EXTRACTED WATERMARKS WITH DIFFERENT HORIZONTAL PERSPECTIVE ANGLES.

Horizontal(°) Erroneous Bits	David et.al [12]	Pramila et.al [28]	Yang et.al [39]	Proposed green	black
Left 65	20.2	30.2	43.6	**15.1**	**18.3**
Left 60	17.1	18.3	40.1	**8.2**	**9.8**
Left 45	21.3	17.6	30.1	**9.3**	**9.2**
Left 30	23.1	16.4	26.4	**9.6**	**9.7**
Left 15	21.5	14.5	22.1	**8.4**	**9.1**
0	23.2	14.0	20.5	**8.6**	**9.1**
Right 15	16.9	15.9	24.1	**9.7**	**9.8**
Right 30	15.1	18.1	29.7	**9.4**	**9.5**
Right 45	16.5	19.2	31.1	**9.4**	**9.8**
Right 60	13.7	23.2	35.3	**9.9**	**9.8**
Right 65	17.9	26.1	37.8	**17.1**	**19.3**

2) The Impact of Horizontal Perspective Angle on Robustness: Table IV and Fig. 17b indicates the EB obtained with

TABLE V:
THE EXAMPLE OF SCREEN-SHOOTING DOCUMENTS WITH DIFFERENT HORIZONTAL PERSPECTIVE ANGLES.

Horizontal angle (°)	Left 65	Left 30	Left 15	Right 45	Right 60
Recaptured					
Recovered					

different schemes at the same shooting distance of 40cm but different shooting angles. Table V shows the examples of recaptured image with different horizontal perspective angle and the corresponding recovered images. The EB of the proposed are lower than the compared algorithm in the angles of Left 65° to Right 60°, and the message bits can be well recovered in the angle range. But when shooting angle is larger than 60°, the erroneous bits becomes too large to be corrected. So the suitable angle of the proposed algorithm is from Left 60° to Right 60°. It's worth noting that as the angle increases, the locating waveform becomes more unsatisfactory, and we should have a selection to extract the watermark in the well-recorded part of the image.

3) The Impact of Vertical Perspective Angle on Robustness: Table VI and Fig. 17c indicates the EB obtained with different schemes at the same shooting distance of 40cm but different vertical shooting angles. Table VII shows the examples of recaptured image with different vertical perspective angle and

TABLE VI:
ERRONEOUS BITS OF THE EXTRACTED WATERMARKS WITH
VERTICAL PERSPECTIVE ANLGE.

Vertical (°) Erroneous Bits	David et.al [12]	Pramila et.al [28]	Yang et.al [39]	Proposed green	Proposed black
Up 65	24.2	31.3	42.1	**17.7**	**15.6**
Up 60	22.4	17.7	39.5.1	**10.0**	**9.0**
Up 45	26.6	15.2	36.1	**8.7**	**9.8**
Up 30	26.1	14.7	27.2	**9.4**	**8.4**
Up 15	16.2	14.1	22.1	**9.6**	**8.1**
0	22.4	12.0	20.5	**7.8**	**9.2**
Down 15	13.8	13.1	25.4	**8.7**	**9.2**
Down 30	22.4	14.5	28.9	**9.5**	**9.7**
Down 45	24.8	14.9	31.0	**8.3**	**9.7**
Down 60	17.1	15.2	32.3	**8.9**	**9.9**
Down 65	22.7	16.3	36.4	**16.1**	**14.3**

TABLE VII:
THE EXAMPLE OF SCREEN-SHOOTING DOCUMENTS WITH
DIFFERENT VERTICAL PERSPECTIVE ANLGE.

TABLE VIII:
ERRONEOUS BITS OF THE EXTRACTED WATERMARKS WITH
DIFFERENT SHOOTING DISTANCE.

Distance(cm) Erroneous Bits	Pramila et.al [28]	Yang et.al [39]	Proposed white
15	-	-	**5.1**
25	10.1	15.8	**9.8**
35	13.5	20.3	**9.9**
45	14.4	24.5	**9.0**
55	13.7	29.1	**7.6**
65	17.2	33.4	**9.2**
75	19.5	36.8	**9.9**
85	22.3	40.2	**9.2**
95	34.1	39.1	**13.3**

TABLE IX:
THE EXAMPLE OF PRINT-SHOOTING DOCUMENTS WITH
DIFFERENT DISTANCE.

the corresponding recovered images. The EB of the proposed are lower than the compared algorithm in the angles of Up 65° to Down 65°, and the message bit can be recovered in the angle of Up 60° to Down 60°. Note that EB does not increase with shooting angle, this is because we set the error correction capability to 10 bits, so there is no difference when the error bit below 10 bits, in other words, we don't search for the lowest EB in each shooting condition, as long as it is lower than 10 bits.

C. The robustness tests in print-shooting process

We carry out the experiment from multiple perspectives to show the robustness of the proposed algorithm in the print shooting process. We use the white underpainting with black text for the experiment, as shown in Fig. 16d, which is more realistic since monochrome printer is more mainstream for document printing.

1) The Impact of Distance on Robustness: Table VIII indicates the EB with different shooting distances. Table IX shows the examples of recaptured images in different distance and the corresponding recovered images. It's easy to see that the proposed algorithm can be well suited in the distance of 15cm-85cm, since the designed BCH code can correct 10-bit error. So as long as the shooting distance is within 15cm-85cm, we can well recover the message bits. Besides, as we can see in Table IX, when shooting distance is beyond 85cm, the text

is also difficult to identify, so this suitable distance is in line with most application scenarios.

TABLE X:
ERRONEOUS BITS OF THE EXTRACTED WATERMARKS WITH
DIFFERENT HORIZONTAL PERSPECTIVE ANGLES.

Horizontal(°) Erroneous Bits	Pramila et.al [28]	Yang et.al [39]	Proposed white
Left 65	20.2	36.2	**16.3**
Left 60	17.5	39.1	**9.9**
Left 45	13.3	33.6	**9.7**
Left 30	12.1	32.1	**8.9**
Left 15	11.0	35.7	**7.6**
0	9.3	30.3	**7.6**
Right 15	10.9	31.2	**8.5**
Right 30	12.3	34.5	**9.8**
Right 45	14.7	36.5	**9.3**
Right 60	16.2	39.0	**9.2**
Right 65	18.5	41.1	**13.6**

2) The Impact of Horizontal Perspective Angle on Robustness: Table X shows the examples of recaptured image with different horizontal perspective angle and the corresponding recovered images. Table XI lists the EB obtained with different schemes at the same shooting distance of 40cm but different shooting angles. We can see that the suitable angle of the

TABLE XI:
THE EXAMPLE OF PRINT-SHOOTING DOCUMENTS WITH
DIFFERENT HORIZONTAL PERSPECTIVE ANGLES.

Horizontal angle (°)	Left 65	Left 45	Left 15	Right 45	Right 65
Recaptured					
Recovered					

proposed algorithm is from Left 60° to Right 60° since the EB is no more than 10 bits. Though the algorithm cannot work well in the shooting angle larger than 60°, but in actual, shooting a document in a large angle in quite rare.

TABLE XII:
ERRONEOUS BITS OF THE EXTRACTED WATERMARKS WITH
VERTICAL PERSPECTIVE ANLGE.

Vertical (°) Erroneous Bits	Pramila *et.al* [28]	Yang *et.al* [39]	Proposed white
Up 65	21.3	44.2	**14.1**
Up 60	18.1	40.3	**9.7**
Up 45	16.2	38.9	**9.4**
Up 30	14.1	37.9	**8.1**
Up 15	13.2	38.1	**8.0**
0	11.7	33.1	**8.2**
Down 15	11.2	34.3	**9.1**
Down 30	13.5	35.3	**8.9**
Down 45	13.8	34.9	**9.4**
Down 60	14.7	36.1	**10.0**
Down 65	16.4	39.3	**17.3**

TABLE XIII:
THE EXAMPLE OF PRINT-SHOOTING DOCUMENTS WITH
DIFFERENT VERTICAL PERSPECTIVE ANLGE.

Vertical angle (°)	Up 65	Up 30	Up 15	Down 45	Down 65
Recaptured					
Recovered					

3) The Impact of Vertical Perspective Angle on Robustness:
Table XII indicates the EB obtained with different schemes at the same shooting distance of 40cm but different vertical shooting angles. Table XIII shows the examples of recaptured image with different vertical perspective angle and the corresponding recovered images. The proposed algorithm can be applied in the angle from Up 60° to Down 60° since the EB is no more than 10 bits. So the proposed algorithm is robust

to most of the vertical shooting angles.

D. Universality tests for different devices

TABLE XIV:
ERRONEOUS BITS OF THE EXTRACTED WATERMARKS WITH
DIFFERENT DEVICES.

Mobile phones		iPhone 6s	Mi 6	Huawei G8	Meizu M Note6
Screens	ViewSonic VA2261	8.6	8.6	8.6	8.6
	Lenovo P22i	8.8	7.9	8.7	8.9
	Acer S230HL	8.4	8.9	8.1	8.6
	Acer S235HL	9.0	8.5	8.3	8.4
	ViewSonic VA 1916w	8.3	8.1	8.4	8.1
Printers	HP LaserJet Pro M403d	7.8	8.3	7.7	7.9
	HP OfficeJet Pro 8720	8.0	8.2	7.9	8.0

In order to tests the robustness of the algorithm for different devices, we conduct the algorithm with different mobile phones, different monitors and different printers. Similarly as the experiment in Section V-B and Section V-C, 10 documents with black text and light green underpainting were used for the screen shooting tests, and 10 documents with black text and white underpainting were used for the print shooting tests, while the shooting distance is 40cm.

Table XIV indicates the EB obtained with different equipments, we can see that for all tests of mobile phones, screens and printers, the extracted EB are within 10 bits, that is, this algorithm has good applicability to mobile phones, screens and printers used in the tests.

VI. CONCLUSION AND FUTURE WORK

An underpainting-based camera shooting resilient watermarking scheme is proposed in this paper. We innovatively proposed the idea of using the underpainting to embed watermark. In order to satisfy both the robustness of the camera shooting process and the inconspicuous of the watermark, we used the method of exchanging DCT coefficients to embed the watermark. Then, we designed the flip-based method to arrange the watermarked underpainting, which ensures the strong autocorrelation and symmetry of the underpainting. Thus we can achieve accurate locating even only part of the document is captured. Besides, a compensation method for camera shooting distortions are proposed to increase the extraction accuracy. The experimental results show that the proposed watermarking scheme achieves both the high robustness for camera shooting process and the high visual quality.

REFERENCES

[1] M.-J. Tsai, C.-L. Hsu, J.-S. Yin, and I. Yuadi, "Digital forensics for printed character source identification," in *Multimedia and Expo (ICME), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2953720, IEEE Transactions on Circuits and Systems for Video Technology

14

[2] Z. Liu and A. Inoue, "Audio watermarking techniques using sinusoidal patterns based on pseudorandom sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 801–812, 2003.

[3] G. Hua, L. Zhao, H. Zhang, G. Bi, and Y. Xiang, "Random matching pursuit for image watermarking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 3, pp. 625–639, 2018.

[4] X.-L. Liu, C.-C. Lin, and S.-M. Yuan, "Blind dual watermarking for color images authentication and copyright protection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 5, pp. 1047–1055, 2016.

[5] T. Zong, Y. Xiang, I. Natgunanathan, S. Guo, W. Zhou, and G. Beliakov, "Robust histogram shape-based method for image watermarking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 5, pp. 717–729, 2014.

[6] M. Asikuzzaman and M. R. Pickering, "An overview of digital video watermarking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2131–2153, 2017.

[7] P. W. Chan, M. R. Lyu, and R. T. Chin, "A novel scheme for hybrid digital video watermarking: approach, evaluation and experimentation," *IEEE transactions on circuits and systems for video technology*, vol. 15, no. 12, pp. 1638–1649, 2005.

[8] A. Draganić, M. Marić, I. Orović, and S. Stanković, "Identification of image source using serial-number-based watermarking under compressive sensing conditions," in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2017, pp. 1227–1232.

[9] I. Orović, P. Zogović, N. Žarić, and S. Stanković, "Speech signals protection via logo watermarking based on the time–frequency analysis," *Annals of telecommunications-annales des telecommunications*, vol. 63, no. 7-8, pp. 369–377, 2008.

[10] D. Zheng, J. Zhao, and A. El Saddik, "Rst-invariant digital image watermarking based on log-polar mapping and phase correlation," *IEEE transactions on circuits and systems for video technology*, vol. 13, no. 8, pp. 753–765, 2003.

[11] S. Xiang, H. J. Kim, and J. Huang, "Invariant image watermarking based on statistical features in the low-frequency domain," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 6, pp. 777–790, 2008.

[12] D. Gugelmann, D. Sommer, V. Lenders, M. Happe, and L. Vanbever, "Screen watermarking for data theft investigation and attribution," in *2018 10th International Conference on Cyber Conflict (CyCon)*. IEEE, 2018, pp. 391–408.

[13] L. Zhang, C. Chen, and W. H. Mow, "Accurate modeling and efficient estimation of the print-capture channel with application in barcoding," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 464–478, 2018.

[14] Accessed: Dec. 2018. [Online]. Available: https://theintercept.com/2017/06/05/top-secret-nsa-report-details-russian-hacking-effort-days-before-2016-election/.

[15] Z. Jalil and A. M. Mirza, "A review of digital watermarking techniques for text documents," in *2009 International Conference on Information and Multimedia Technology*. IEEE, 2009, pp. 230–234.

[16] N. S. Kamaruddin, A. Kamsin, L. Y. Por, and H. Rahman, "A review of text watermarking: Theory, methods, and applications," *IEEE Access*, vol. 6, pp. 8011–8028, 2018.

[17] X. Sun and A. J. Asiimwe, "Noun-verb based technique of text watermarking using recursive decent semantic net parsers," in *International Conference on Natural Computation*. Springer, 2005, pp. 968–971.

[18] M. Topkara, U. Topkara, and 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*. ACM, 2006, pp. 37–46.

[19] ——, "Information hiding through errors: a confusing approach," in *Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505. International Society for Optics and Photonics, 2007, p. 65050V.

[20] D. Huang and H. Yan, "Interword distance changes represented by sine waves for watermarking text images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 12, pp. 1237–1245, 2001.

[21] A. M. Alattar and O. M. Alattar, "Watermarking electronic text documents containing justified paragraphs and irregular line spacing," in *Security, Steganography, and Watermarking of Multimedia Contents VI*, vol. 5306. International Society for Optics and Photonics, 2004, pp. 685–695.

[22] Q.-C. Li and Z.-H. Dong, "Novel text watermarking algorithm based on chinese characters structure," in *2008 International Symposium on Computer Science and Computational Technology*, vol. 2. IEEE, 2008, pp. 348–351.

[23] V. Yazdani, M. A. Doostari, and H. Yazdani, "A new method to persian text watermarking using curvaceous letters," *J. Basic Appl. Sci. Res.*, vol. 3, no. 4, pp. 125–131, 2013.

[24] S. H. Low, N. F. Maxemchuk, and A. M. Lapone, "Document identification for copyright protection using centroid detection," *IEEE Transactions on Communications*, vol. 46, no. 3, pp. 372–383, 1998.

[25] P. V. K. Borges, J. Mayer, and E. Izquierdo, "Robust and transparent color modulation for text data hiding," *IEEE Transactions on Multimedia*, vol. 10, no. 8, pp. 1479–1489, 2008.

[26] H. Fang, W. Zhang, H. Zhou, H. Cui, and N. Yu, "Screen-shooting resilient watermarking," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 6, pp. 1403–1418, 2018.

[27] X. Kang, J. Huang, and W. Zeng, "Efficient general print-scanning resilient data hiding based on uniform log-polar mapping," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 1–12, 2010.

[28] A. Pramila, A. Keskinarkaus, and T. Seppänen, "Toward an interactive poster using digital watermarking and a mobile phone camera," *Signal, Image and Video Processing*, vol. 6, no. 2, pp. 211–222, 2012.

[29] C. Xiao, C. Zhang, and C. Zheng, "Fontcode: Embedding information in text documents using glyph perturbation," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 2, p. 15, 2018.

[30] T. Richter, S. Escher, D. Schönfeld, and T. Strufe, "Forensic analysis and anonymisation of printed documents," in *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2018, pp. 127–138.

[31] Accessed: April 2018. [Online]. Available: https://www.microsoftstore.com.cn.

[32] Accessed: April 2018. [Online]. Available: https://www.spyder.com.

[33] W.-g. Kim, S. H. Lee, and Y.-s. Seo, "Image fingerprinting scheme for print-and-capture model," in *Pacific-Rim Conference on Multimedia*. Springer, 2006, pp. 106–113.

[34] W. W. Peterson and D. T. Brown, "Cyclic codes for error detection," *Proceedings of the IRE*, vol. 49, no. 1, pp. 228–235, 1961.

[35] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and control*, vol. 3, no. 1, pp. 68–79, 1960.

[36] T. Nakamura, A. Katayama, M. Yamamuro, and N. Sonehara, "Fast watermark detection scheme for camera-equipped cellular phone," in *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*. ACM, 2004, pp. 101–108.

[37] H. Herbert, "The history of ocr, optical character recognition," *Manchester Center, VT: Recognition Technologies Users Association*, 1982.

[38] J. Austen, *Pride and prejudice*. Bantam Classics, 1981.

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

Han Fang received his B.S. degree in 2016 from Nanjing University of Aeronautics and Astronautics (NUAA). He is currently pursuing the Ph.D. degree in Information Security in University of Science and Technology of China (USTC). His research interests include image watermarking, information hiding and image processing.

Weiming Zhang received his M.S. degree and Ph.D. degree in 2002 and 2005 respectively from the Zhengzhou Information Science and Technology Institute, P.R. China. Currently, he is a professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include information hiding and multimedia security.

Shan Sun received her B.S. degree in 2018 from Anhui University (AHU). She is currently pursuing the M.S degree in Information Security in University of Science and Technology of China (USTC). Her research interests include text watermarking, information hiding and image processing.

Zehua Ma received his B.S. degrees in information security from the University of Science and Technology of China (USTC) in 2018. He is currently pursuing the M.S. degree in information security in USTC. His research interests include image watermarking, information hiding, and image processing.

Hao Cui received his dual B.S. degrees in Geophysics and Computer Science and Technology from the University of Science and Technology of China (USTC) in 2017. He is currently pursuing the M.S. degree in Electronics and Communication Engineering in USTC. His research interests include image watermarking and information hiding.

Hang Zhou received the B.S. degree in School of Communication and Information Engineering, Shanghai University, in 2015. He is currently pursuing the Ph.D. degree in Information Security in University of Science and Technology of China (USTC). His research interests include information hiding, image processing and computer graphics.

Nenghai Yu received his B.S. degree in 1987 from Nanjing University of Posts and Telecommunications, M.E. degree in 1992 from Tsinghua University and Ph.D. degree in 2004 from University of Science and Technology of China, where he is currently a professor. His research interests include multimedia security, multimedia information retrieval, video processing and information hiding.