Screen-Shooting Resilient Watermarking

Han Fang, Weiming Zhang, Hang Zhou, Hao Cui, and Nenghai Yu

***Abstract*— This paper proposes a novel screen-shooting resilient watermarking scheme, which means that if the watermarked image is displayed on the screen and the screen information is captured by the camera, we can still extract the watermark message from the captured photo. To realize such demands, we analyzed the special distortions caused by the screen-shooting process, including lens distortion, light source distortion, and moiré distortion. To resist the geometric deformation caused by lens distortion, we proposed an intensity-based scale-invariant feature transform (I-SIFT) algorithm which can accurately locate the embedding regions. As for the loss of image details caused by light source distortion and moiré distortion, we put forward a small-size template algorithm to repeatedly embed the watermark into different regions, so that at least one complete information region can survive from distortions. At the extraction side, we designed a cross-validation-based extraction algorithm to cope with repeated embedding. The validity and correctness of the extraction method are verified by hypothesis testing. Furthermore, to boost the extraction speed, we proposed a SIFT feature editing algorithm to enhance the intensity of the keypoints, based on which, the extraction accuracy and extraction speed can be greatly improved. The experimental results show that the proposed watermarking scheme achieves high robustness for screen-shooting process. Compared with the previous schemes, our algorithm provides significant improvement in robustness for screen-shooting process and extraction efficiency.**

***Index Terms*— Screen-shooting process, scale-invariant feature transform, SIFT feature editing, hypothesis testing, robust watermarking.**

I. Introduction

THIS paper proposed a novel screen-shooting resilient (SSR) watermarking scheme. The screen-shooting process, that is, using a camera to capture images displayed on the screen has become more common due to the development of smart phones. This makes it especially important to design a watermarking a watermarking algorithm that can extract information from screen-shooting images. There are two typical application scenarios of SSR watermarking scheme. **1). Leak tracking.** As shown in Fig. 1a, in a hardware-isolated environment, it is difficult to steal secret documents in the traditional way, such as using a USB flash drive or sending email. But with a digital camera, the commercial spy who usually is an authorized employee can steal the information by simply opening it on the screen and taking a picture without leaving any records. And this behavior is difficult to forbid from the outside. To eliminate this security problem, we can embed some identification information of the screen or the workstation. So once the secrets were leaked through the photo, we can use the SSR watermarking scheme to extract the message from the photo. Through the extracted message, we can locate the leaking device and narrow the scope of investigation. Thus we can achieve the accountability process for leaking behavior. **2). Information retrieving.** As shown in Fig. 1b, the SSR watermarking algorithm can serve as a channel to transmit information from the screen to camera. By applying SSR watermarking algorithm, the information such as webpage links, product introductions can be embedded into the image, and we can extracted them by simply taking a photo. In this way, we can not only ensure excellent visual quality, but also increase the dimension of information of an image.

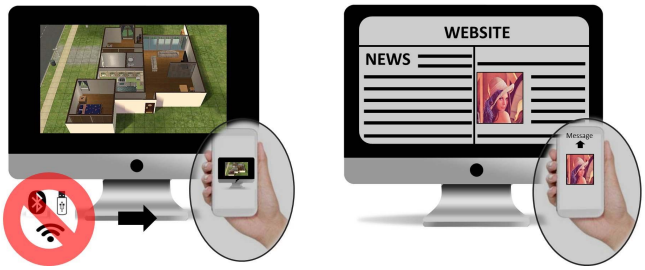


Fig. 1. Typical situations. (a) Recapture the secret documents in screen. (b) Information interaction between screen to camera.

Screen-shooting process is similar to traditional screen-camera communications. And there are mainly two types of traditional screen-camera communication algorithms. The first type uses code-image (*i.e.* 2-D barcode) to transmit information, which greatly affects the visual quality [1], [21]. The second type embeds data into temporal dimensions with high-frequency or low-amplitude inter-frame pixel change. Li *et al.* [44] proposed a screen-camera communication method based on encoding data into the pixel translucency change at the separate image layer, Nguyen *et al.* [45] propose to embed the message into the frequency changing of a video, while Iwata *et al.* [46] designed a video watermarking scheme based on changing the brightness of a single frame. This type of method requires multiple frames for extraction, which is not suitable for the scenario we concerned. Because in the scene of leak tracking, the commercial spy may only take

Manuscript received June 23, 2018; revised September 17, 2018 and October 13, 2018; accepted October 18, 2018. Date of publication October 29, 2018; date of current version February 13, 2019. This work was supported in part by the Natural Science Foundation of China under Grant U1636201 and Grant 61572452. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dinu Coltuc. *(Corresponding author: Weiming Zhang.)*

The authors are with the 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).

Digital Object Identifier 10.1109/TIFS.2018.2878541

a photo instead of record a video, so the method based on video cannot make it. In the scene of information retrieving, if the user is required to record a video to extract information, the user experience will be poor. However, by using the SSR watermarking algorithm, we can achieve the information transfer between screen and camera in a single photo. That is to say, we can get the corresponding message from a single photo instead of a particular code-image or a recorded video, which is very useful in both of the scenarios we concerned.

It is worth noting that the requirements of the above two scenarios are different. For the scene of leak tracking, we pay more attention to the transparency of the watermark and the accuracy of the extraction. This requires the modification of the original image should be as small as possible. But the extraction side is very flexible. For the scene of information interaction, we care more about the timeliness of extraction. In this case, whether extraction can be achieved real-time determines whether such scheme has a good user experience, while the requirements of visual quality on the embedded side are not so strict.

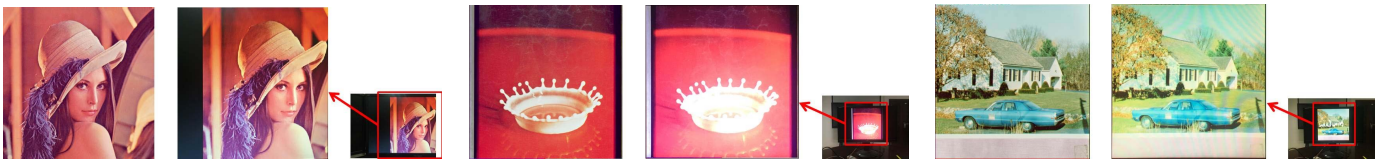
Traditional watermarking algorithms which are mostly robust to image processing attacks [2]–[5] do not work well for screen-shooting process. When shooting the image displayed on the screen, the image as well as the watermark undergoes a series of analog-to-digital and digital-to-analog conversion processes which will appear as suffering from a combination of strong attacks [6]. So in order to resist screen-shooting distortion, we need to analyze every distortion brought by the process. Screen-shooting process can be seen as a so-called cross-media information transfer process. And the cross-media information transfer process also includes print-scanning and print-camera process. In previous years, the print-scanning resilient (PSR) watermarking schemes and the print-camera resilient (PCR) watermarking schemes have been extensively studied. The process that first printing the image on a paper and then scanning though a scanner is called print-scanning process. Rotation, translation, scaling and cropping are the common distortions beyond the process. Besides, because of the difference of basic color system between printer and screen, the print-scanning process will bring color distortion as well [7]. PSR watermarking scheme can be broadly divided into three categories that are feature point based methods [8], [9], template based methods [10], [11], transform invariant domains based methods [12], [13]. Among these methods, the transform invariant domains based approach is most representative. Lin *et al.* [14] and Zheng *et al.* [15] proposed to embed a watermark into the Fourier-Mellin transform (log-polar mapping plus discrete fourier transform) domain. They proposed to perform inverse log-polar mapping (ILPM) to resist the rotation, scaling and translation (RST) distortions. However, Kang *et al.* [16] pointed out that the ILPM based methods would produce interpolation distortion and reduce the embedding areas. So they suggested the uniform log-polar mapping (ULPM) algorithm, based on which the embedding area is effectively increased. At the extraction side, spread spectrum operation will greatly reduce the false positive rate. However, this method embeds a watermark in the full image. Therefore, the robustness to crop attacks is weak.

Another cross-media process mentioned above is print-camera process which means taking a photo of the printed image. Print-camera process can be seen as an addition to the print-scanning process. The watermark must be robust to more distortions such as lens distortion, light distortion and so many other distortions [23]. So far, PCR watermarking schemes can be roughly divided into two categories. One is transform invariant domain based method which is developed from print-scanning watermarking scheme. For example, to resist lens distortion, Delgado-Guillen *et al.* [17] improved Kang's [16] method by applying a well-designed border outside the image, but the embedding and extracting algorithms are same as Kang's. The other PCR watermarking method is template-additive based method which is first proposed by Nakamura *et al.* [18]. They use a set of orthogonal templates to represent the $0/1$ message and add the corresponding templates to the cover image to embed the watermark. At the extraction side, they used template matching method to extract message. In the same year, they designed a border to help to correct the lens distortion [19]. Kim *et al.* [20] proposed a method to embed messages in the form of pseudo-random vectors. In order to read the message, the grid formed by the message is detected by the autocorrelation function and then the message is read by applying cross-correlation operation. Pramila *et al.* [22], [23] proposed a multi-scale template based method. They generated a periodic template and encoded the information by modulating the direction of the template. Based on which, the embedding capacity for an unit block is effectively expanded. At the extraction side, Pramila *et al.* [23] used the hough transform to detect the angle of template to extract the message. In recent years, Pramila *et al.* also optimized the image processing procedure for this method which effectively enhanced both the visual quality of the image and the robustness of the watermark [24], [25].

However, the experiments in Section V shows that these methods are not applicable in screen-shooting process. The screen-shooting process has its particularity, and the proposed algorithm is designed by analyzing the special distortions caused by the process. Our major contributions are:

- We suggested an intensity-based scale-invariant feature transform (I-SIFT) algorithm which can accurately locate the embedding regions without any information of the original image.
- We put forward a small-size template algorithm to repeatedly embed the watermark into different regions, so that at least one complete information region can survive from light source distortion and moirÂ´e distortion.
- We designed a cross-validation based blind extraction algorithm to work in combination with repeated embedding, based on which the extraction accuracy can be effectively guaranteed.
- We proposed a SIFT feature editing algorithm to enhance the intensity of the keypoints, by applying such algorithm, the robustness of the keypoint and the speed of the extraction process can be greatly improved.

The rest of the paper is organized as follows. In Section II, we analyze the distortion caused by screen-shooting process. Section III describes the specific procedures of the proposed



(a) (b) (c)

Fig. 2. The distortions caused by screen-camera process. (a) The lens distortion. (b) The light source distortion. (c) The moiré pattern distortion.

algorithm. In Section IV, we propose an optimized method to accelerate the speed of the extraction process. In Section V, we mainly discuss the choice of parameters in the algorithm and show the experimental results. Section VI draws the conclusion.

II. Screen-Shooting Distortion Analysis

Schaber *et al.* [27] established a screen-shooting model for video and analyzed the distortion caused by the screen-shooting process. Combining the model with the analysis of Kim *et al.* [20], we summarize the distortions of the screen-shooting process into four aspects, that are display distortion, lens distortion, sensor distortion, and processing distortion. Among them, there are the following three categories that we should pay more attention to.

A. Lens Distortion

Due to the diversity of shooting angles, RST distortion may occur after perspective transformation, as shown in Fig. 2a. When the image is not completely captured by the camera, some detail of the image may lose after the perspective correction, which requires the locating algorithm to be robust enough so that the embedded regions can be accurately located in the distorted image. Besides, we need to achieve blind extraction which means the original image or other prior information will not apply. So to sum up, we need a locating algorithm which requires no prior information to accurately locate the embedded region in the distorted image.

B. Light Source Distortion

As for the light source distortion, in addition to outside light, the screen is also a light source itself, the inhomogeneity of the screen light source will produce a lot of brightness distortion. As shown in Fig. 2b, the light at the bottom of the screen is stronger, which makes the lower part of the image brighter. So a full-map embedding method cannot apply. In order to solve this problem, we should repeatedly embed the message in multiple regions to ensure that at least one complete watermark sequence can survive from distortion.

C. Moiré Pattern Distortion

When the spatial frequency of the pixel in the camera sensor is close to that in the screen, the moiré pattern is generated as shown in Fig. 2c. The moiré pattern is irregular and will spread over the image [26], which will cause great distortions to the detail information. So we need to embed the message into a domain which is affected less by moiré pattern to make sure the influence caused by moiré pattern is as small as possible.

To resist screen-shooting process, these three types of distortion are the core issues we need to consider. For lens distortion, in addition to carry out perspective correction, we also need to accurately locate the embedding regions in the distorted image. Schaber *et al.* [27] proved that scale-invariant feature transform (SIFT) keypoints have the screen-shooting invariance. So we can use SIFT to locate the embedding regions. However, the traditional SIFT-based watermarking algorithms mainly have the following two disadvantages. **a)**. The descriptors are needed in locating process, which obviously cannot meet the requirements of blind extraction [40], [41]. **b)**. The descriptive information (*i.e.* main direction and scale) for each point needs to be calculated, which greatly increases the locating time [42], [43]. So we put forward the intensity based SIFT that does not require any prior information to locate the embedding regions. And at the same time, the locating process can be speeded up by applying such algorithm.

As for the loss of detail information caused by light source distortion and moiré distortion, we need to repeatedly embed the watermark information in the image. But the prerequisite for this operation is that a complete watermark embedding unit cannot be too large, so we designed a small-size template algorithm to embed watermark in small region. By applying such algorithm, the robustness in screen-shooting process and phone's compression process can be both satisfied.

Corresponding to the repeatedly embedding method, in order to obtain the watermark information correctly, we designed a cross-validation based extraction algorithm which can greatly reduce the false alarm rate and missed alarm rate.

Furthermore, in order to boost the extraction speed, we also proposed a SIFT intensity editing algorithm which can greatly enhance the intensity of keypoints so that the locating time at the extraction side can be greatly reduced.

III. Proposed Method

A. The Embedding Process

Fig. 3 shows the framework of the embedding process. For an image, if it is a color image, we convert it to YCbCr colorspace and then choose Y-channel image as the host image. Otherwise, we directly use the grayscale image as the host image. Then, we use the intensity based SIFT algorithm (I-SIFT) to locate the keypoints. According to the keypoints' intensity, we filter the feature regions. Finally, the watermark is embedded by small-size template method.

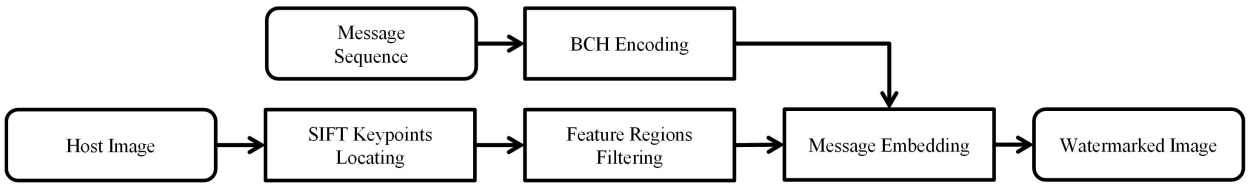


Fig. 3. The framework of the embedding process.

1) Intensity Based SIFT: The intensity based SIFT algorithm is evolved from SIFT. Lowe [28] proposed SIFT algorithm which is widely used for image matching and image retrieval. SIFT algorithm is mainly divided into two steps. **a)**. Locate the extreme points. **b)**. Generate the descriptor for each point. Step **a)** is described as follows. For a particular octave o, the input image I_o is down-sampled from the image I of the previous layer with the sampling scale ρ, as shown in Eq.(1).

$$I_o(x, y) = I(\lceil \rho x \rceil, \lceil \rho y \rceil); \quad x \in [1, M/\rho], \ y \in [1, N/\rho], \tag{1}$$

where M and N are the size of the image I. Then Gaussian filters with different scales are applied to I_o, as Eq.(2) and Eq.(3) indicates, to generate a series of Gaussian blurred images.

$$L(x, y, \sigma) = G(x, y, \sigma) * I_o(x, y) \tag{2}$$

and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \cdot \mathrm{e}^{-\frac{(x^2+y^2)}{2\sigma^2}}, \tag{3}$$

where (x, y) is the coordinates of the image. σ is the variance of Gaussian filter. The size of σ determines the smoothness of the image. In order to effectively detect the extreme keypoints in the scale space, Difference of Gaussians (DoG) domain is formed. For a particular octave o, the DoG image is defined by

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma). \tag{4}$$

We define $\mathbf{p} = (x, y, \sigma)$ to represent a specific point in DoG domain, so we can write $D(\mathbf{p}) \equiv D(x, y, \sigma)$. Each point in the DoG domain is compared with the 26 values around the $3 \times 3 \times 3$ cube centered at itself. If $D(\mathbf{p})$ is the maximum or the minimum, $\mathbf{p}$ is considered as an extreme point of the image. Extreme points will be further refined to remove low-contrast keypoints and unstable edge response points. At step **b)**, a 128-D descriptor is calculated for the remaining points. The specific procedure can be found in [28].

The descriptor of keypoints is the key to locate the corresponding points in the image. However, if we use the descriptor to locate the feature point, we need to obtain the 128-dimensional information of the keypoint in advance, which obviously cannot apply to blind extraction. In addition, it is also time-consuming when generating the descriptor. So in the proposed method, we put forward an intensity based SIFT algorithm which use the intensity of the keypoint instead of applying the descriptor to locate the keypoints. The intensity is defined by

$$In(\mathbf{p}) = |D(\mathbf{p})| \tag{5}$$

where $D(\mathbf{p})$ is defined by Eq.(4). So when keypoints are selected, we sort the intensity of keypoints in descending order and choose n keypoints with the largest intensity as the candidate points, we named these points as "top-n" keypoints. It's worth noting that the intensity as well as the exact location of the corresponding keypoint will change after the screen-shooting process, which will cause some points to fall out of "top-n". So to locate as many "top-n" keypoints of the original image as possible in the distorted image, we should perform an extra operation at the extraction side. The specific method will be introduced in Section III-B2.

2) Selecting Feature Regions: For a binary watermark sequence of length l, we first fill it into a binary matrix W of size $a \times b$ by column, as shown in Fig. 4. Note that $a \times b$ should be no less than l, so we need to choose the appropriate a and b to ensure a and b are as equal as possible and at the same time, $a \times b$ and l are as equal as possible, which can be formulated by

$$\min_{a,b} |a - b| + (a \times b - l) \tag{6}$$

The rest $a \times b - l$ bits in W are directly filled by 0, so the resulting W is closer to a phalanx. The reason can be summarized as since we need to embed multiple regions centered at keypoints, the closer the embedding region is to the square, the more the image space can be utilized. In the proposed algorithm, 1-bit information is embedded in a block with size of 8×8 pixels. So the size of an embedding region is $(a \times b) * (8 \times 8)$ pixels. Since the embedding regions centered at each keypoints (*i.e.* feature regions) should not overlap with each other, we need to filter the feature regions. The filter operation can be regarded as the following formulation which can be solved by greedy algorithm

$$\begin{aligned} &\max \Big(\sum_{i=1}^{n} In(\mathbf{p}_i)\Big) \\ &\text{s.t. } A(\mathbf{p}_i) \bigcap A(\mathbf{p}_j) = \varnothing \quad (i \neq j, \ i, j \in [1, n]), \end{aligned}$$

where $In(\mathbf{p}_i)$ denotes the intensity of the keyppoint $\mathbf{p}_i$ and $A(\mathbf{p}_i)$ denotes the embedding region centered at $\mathbf{p}_i$. We filter out n feature regions that satisfy the above condition. Then the watermark is embedded in the n regions. After that, the structural similarity index (SSIM) value is calculated between the embedded region and the original region. Among them, k regions with the highest SSIM value are selected to replace the original regions while the rest $n-k$ regions remain unchanged. In our scheme, n is set as 10 and the setting of k is discussed in Section V-A.

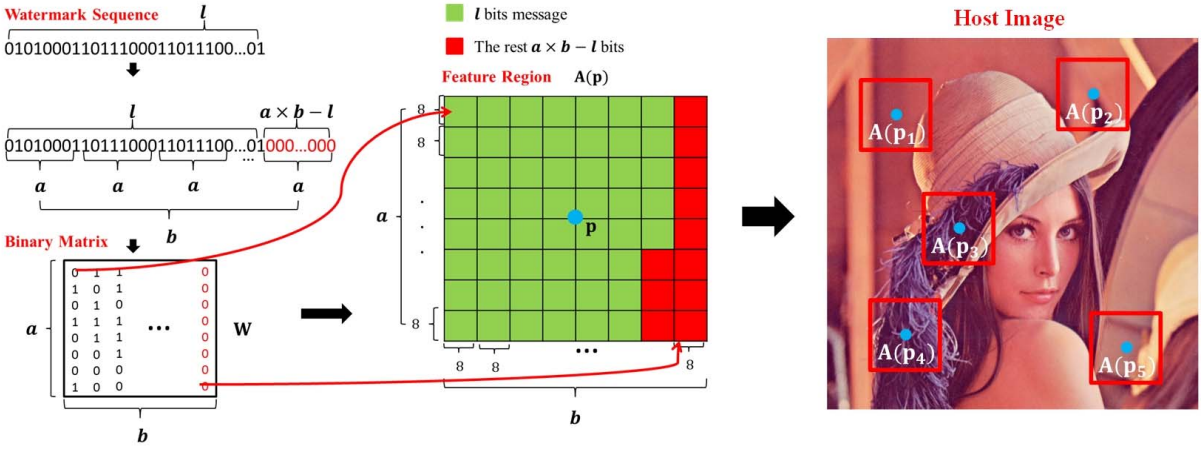


Fig. 4. The relation between watermark sequence and the feature regions.

3) Message Embedding: As mentioned before, since we want to embed a watermark message repeatedly in multiple regions, the size of each region cannot be too large. Previous PCR watermarking scheme has proved that the template-additive based method can resist the distortion caused by the photographing process. However, the key of this algorithm is that the template should be large enough so that its directional characteristics can survive from the distortion. Therefore, we cannot directly use the template-additive based method, but since this method can resist the photographing process, we need to deeply analyze the reason why it is robust to photographing process. So that we can design a small-sized template algorithm.

Fig. 5a shows the templates generated by Nakamura's method [18] and the embedded blocks with the message 0 and message 1. Fig. 5b and Fig. 5c shows the DCT coefficient matrix corresponding to the embedded blocks. Fig. 5d and Fig. 5e shows the DCT coefficient matrix corresponding to the blocks after screen-shooting process. The brighter part in Fig. 5b - Fig. 5e corresponds to a larger value of magnitude. We note that the sum of the DCT coefficient values at the positions (7,7) and (7,8) is R_1, the sum of the DCT coefficient values at the positions of (8,7) and (8,8) is R_2. It can be clearly seen that R_1 is larger than R_2 when the embedding message is 0 and R_1 is smaller than R_2 when the message is 1. So the direction feature of the template can be represented by the relative magnitude of a set of DCT coefficients. Besides, the relative magnitude does not change before and after the screen-shooting process. Therefore, the extraction side can be realized by comparing the value of two group of coefficients in DCT domain instead of the template matching operation in spatial domain, so does the embedding process. Based on the analysis about PCR watermarking scheme, we can ensure that the watermark signal can survive after the screen-shooting process by changing the relative magnitudes of 2 DCT coefficients according to the watermark bit. Besides, by selecting 2 coefficients instead of a set of coefficients, the size of the block to be embedded 1-bit message can be reduced to 8×8 pixels.

For a feature region B, we divide B into $a \times b$ blocks with size 8×8. For each block, we perform DCT to it and select the coefficients C_1 and C_2 of to do the following operation.

$$\begin{cases} C_1 = \max(C_1, C_2), C_2 = \min(C_1, C_2) & if\ w = 0 \\ C_1 = \min(C_1, C_2), C_2 = \max(C_1, C_2) & otherwise \end{cases} \tag{7}$$

As Eq.(7) indicates, we need to confirm that $C_1 \geq C_2$ when $w = 0$ and $C_1 < C_2$ when $w = 1$. Although the method of exchanging the value of DCT coefficients to embed the watermark has been proposed in previous work [36]–[38], it does not meet the requirements of screen-shooting. Since the mobile phone will take JPEG compression process after capturing, we need to ensure that these coefficients can still maintain the relationship after compression. So a redundancy parameter d ($d = |C_1 - C_2|$) is needed to make sure C_1 and C_2 satisfy Eq.(8).

$$\begin{cases} \lfloor \frac{C_1}{q_1} \rfloor \leq \lfloor \frac{C_2}{q_2} \rfloor, & if\ C_1 \leq C_2 \\ \lfloor \frac{C_1}{q_1} \rfloor > \lfloor \frac{C_2}{q_2} \rfloor, & otherwise \end{cases} \tag{8}$$

where q_1 and q_2 are the corresponding JPEG quantization steps of C_1 and C_2. So the embedding formula can be written by

$$\begin{cases} d \geq |q_2 - q_1| \cdot \frac{C_1 \cdot \max(q_1, q_2) + C_2 \cdot \min(q_1, q_2)}{2 q_1 q_2} + \frac{(q_1 + q_2) r}{2} \\ C_1 = \max(C_1, C_2) + \frac{d}{2}, C_2 = \min(C_1, C_2) - \frac{d}{2}. \\ \quad if\ w = 0 \\ C_1 = \min(C_1, C_2) - \frac{d}{2}, C_2 = \max(C_1, C_2) + \frac{d}{2}. \\ \quad otherwise \end{cases} \tag{9}$$

where r is the embedding strength and $r \geq 1$. In the proposed method, q_1 and q_2 are selected by the standard JPEG quality matrix with a quality of 50 [35] and r is set as 1. After modification, we perform IDCT to the block. The operation above is repeated until the k feature regions are embedded. The selection of C_1 and C_2 will be elaborated in Section V-A.

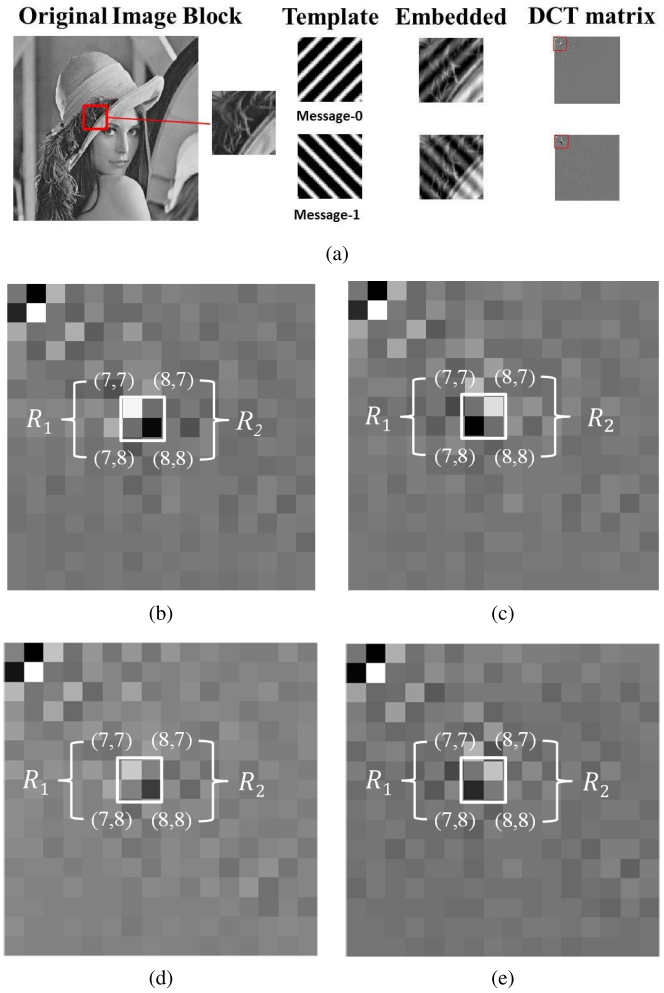


Fig. 5. The relationship between templates and DCT coefficients. (a) The template adding process. (b) The DCT matrix of the original block when embedding message 0. (c) The DCT matrix of the original block when embedding message 1. (d) The DCT matrix of the screen-shooting block when embedding message 0. (e) The DCT matrix of the screen-shooting block when embedding message 1.

The embedding process is described as Algorithm 1.

Algorithm 1 Embedding Algorithm

Input: Image I_{ori}, watermark W, intensity factor r.
Output: Watermarked image I'.

1: Encode the watermark sequences with BCH code and reshape it to a matrix.
2: Generate the Y-channel cover image I by I_{ori}.
3: Locate the keypoints $\mathbf{p}_i (1 \leq i \leq n)$ of I by applying I-SIFT.
4: Filter out feature regions $A(\mathbf{p}_j)(1 \leq j \leq k)$ to be embedded.
5: **for all** $A(\mathbf{p}_j)$ **do**
6: Modify the DCT coefficients by Eq.(9) to embed all the encoded watermark bits;
7: Apply IDCT to the embedded block;
8: **end for**
9: Generate the watermarked image I'.
10: **return** Watermarked image I'.

B. The Extracting Process

The extracting process can be described as Fig. 6. For a screen-shooting image, the perspective transform is first performed to correct the optical distortions. Then we carry out cropping and rescaling operation to obtain the image I' with the same size of the original image. After that, the I-SIFT algorithm is performed to locate the feature regions and watermarks will be extracted from each regions. Among the extracted watermarks, we will use the cross-validation based algorithm to filter out the correct watermark.

1) Optical Distortion Correction: Due to the different shooting conditions, the first thing we need to do is to correct the distortions. We utilize the perspective transform to correct the optical distortion. It is worth noting that since the correction process requires 4 vertices of the picture, the algorithm can only be effective when the four vertices of the picture are recorded, which is the limitation of the algorithm. For different application scenarios, we can use different strategies to get the four vertices. For the scene of leak tracking, the extraction process does not require real time. So the flexibility at the extraction side allows us to use the human eyes to locate the vertices of the image. As long as the image has a certain contrast with the background color, the human eyes can accurately locate the vertices of the image. But for the scene of information retrieving, the extraction process needs to be fast and automatic, which means we cannot introduce manual participation. However, we can add borders around the image to help locate since the image of this scene can be customized. So we can use the method in the article [19] to add a border and locate the image vertices. In either case, we can locate the 4 vertices $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, $P_3(x_3, y_3)$ and $P_4(x_4, y_4)$ of the embedded image in the screen-shooting image, as shown in Fig. 7. Then we set the transformed coordinates corresponding to these 4 points $P'_1(x'_1, y'_1)$, $P'_2(x'_2, y'_2)$, $P'_3(x'_3, y'_3)$ and $P'_4(x'_4, y'_4)$. Substituting the 8 coordinates into Eq.(10) we can get 8 equations, so the value of $a_1, b_1, c_1, a_0, b_0, a_2, b_2, c_2$ can be obtained by solving these equations.

$$x' = \frac{a_1 x + b_1 y + c_1}{a_0 x + b_0 y + 1}$$
$$y' = \frac{a_2 x + b_2 y + c_2}{a_0 x + b_0 y + 1} \tag{10}$$

After determining these parameters, we can form a mapping from the distorted image to the corrected image. Then the corrected image is cropped and rescaled to generating the image to be extracted.

2) Feature Region Locating: The locating process is slightly different from the embedding process. After performing I-SIFT to the Y-channel component of the image, we need to do some extra operations to avoid the errors on keypoints locating. The screen-shooting process will produce two main impacts to the keypoints: **a)**. The intensity change of the keypoints. **b)**. The offset of the point positions.

For impact **a)**, as the intensity order of the keypoints could have changed, we need to increase the number of extraction

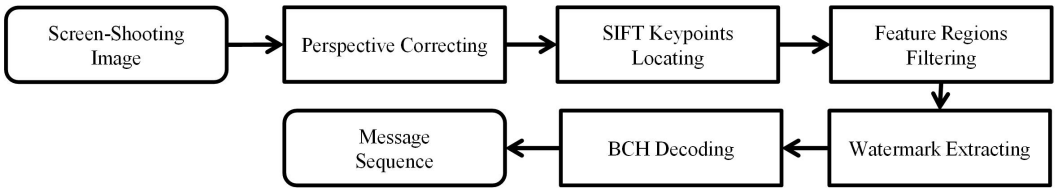


Fig. 6. The framework of extracting process.

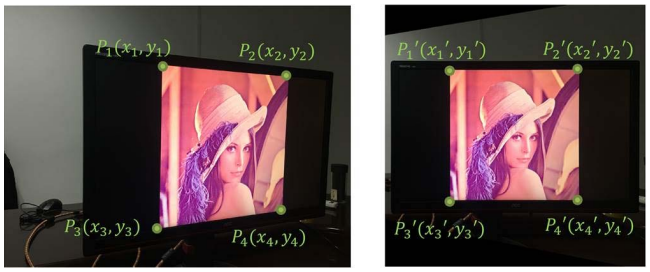


Fig. 7. The correction process.

points to ensure the most of the "top-n" keypoints we selected in the original image can be located in the distorted image. So we doubled the number of extracting keypoints as $2 \cdot k$.

For impact **b)**, as the exact position of the same keypoint will have some slight offset in the image with and without distortion, we need to make a neighborhood traversal to compensate the offset. When locating a keypoint, we perform a 3×3 pixels traversal of this point. The 9 neighborhood points centered at the keypoint are regarded as an extraction point group, then the 9 watermarks extracted from the group are treated as the watermark group of the keypoint. So in general, for an image, we need to extract $2 \cdot k \cdot 9$ complete watermark sequences.

3) Extracting Message: The extracting procedure is an inverse of the embedding procedure. For a feature region B, we divide B into $a \times b$ blocks with size 8×8. Then we perform DCT to each block and select the DCT coefficients C_1 and C_2 to make a comparison to extract the first l bits messages, as shown in Eq.(11).

$$w = \begin{cases} 0, & if\ C_1 \geq C_2 \\ 1, & otherwise \end{cases} \tag{11}$$

For an image, we will obtain $2 \cdot k$ watermark groups and $2 \cdot k \cdot 9$ watermarks. $\mathbf{W}_i (\mathbf{W}_i = [\mathbf{w}_{i1}, \mathbf{w}_{i2} \ldots \mathbf{w}_{i9}])$ denote the watermark group $i (1 \leq i \leq k)$. We select 2 watermark $\mathbf{w}_{i\alpha}, \mathbf{w}_{j\beta}$ from 2 different groups $\mathbf{W}_i, \mathbf{W}_j$ as a pair to make a comparison, then record the watermark pair $\mathbf{w}_f$ whose difference is less than th as shown in Eq.(12) and Eq.(13).

$$diff(\mathbf{w}_{i\alpha}, \mathbf{w}_{j\beta}) = \sum_{x=a}^{b} \sum_{j=a}^{b} [\mathbf{w}_{i\alpha}(x, y) \bigoplus \mathbf{w}_{j\beta}(x, y)] \tag{12}$$

$$\mathbf{w}_f = \begin{cases} \{\mathbf{w}_{i\alpha}, \mathbf{w}_{j\beta}\}, & if\ diff(\mathbf{w}_{i\alpha}, \mathbf{w}_{j\beta}) \leq th \\ \varnothing, & otherwise \end{cases} \tag{13}$$

where $\bigoplus$ is the XOR operation and (x, y) is the coordinates of the watermark matrix. Note that the watermark from the same group will not be compared with each other. Denote l as the number of watermark pairs we record for a image. So the final watermark set is written as

$$\begin{aligned} \overrightarrow{W} &= \{\mathbf{w}_{f1}, \mathbf{w}_{f2}, \ldots, \mathbf{w}_{fl}\} \\ &= \left\{\mathbf{w}_f^1, \mathbf{w}_f^2, \mathbf{w}_f^3, \mathbf{w}_f^4, \ldots, \mathbf{w}_f^{2l-1}, \mathbf{w}_f^{2l}\right\}, \end{aligned} \tag{14}$$

where $\{\mathbf{w}_f^{2l-1}, \mathbf{w}_f^{2l}\}$ is the two watermark matrixes of watermark pair $\mathbf{w}_{fl}$. Then the final watermark is extracted by

$$w(x, y) = \begin{cases} 1, & if\ \sum_{i=1}^{2l} \mathbf{w}_f^i(x, y) \geq l \\ 0, & otherwise \end{cases} \tag{15}$$

where (x, y) is the coordinates of the watermark matrix.

We believed that if the difference of the watermark pair is no larger than th, the watermark is extracted accurately, otherwise, it's a wrong extraction. The reason can be analyzed as follows. Excluding the effect of wrong locating, when the keypoints are located accurately, the extracted watermark would be particularly similar to the original watermark. Therefore, both of the two watermarks in the watermark pair will be approximately the same as the original watermark, so the two watermarks should also be similar with each other. But when the locating is not accurately, the extracted watermark bits should be relatively random, therefore, the similarity between the two watermarks should be very small. So when the similarity between the watermark pair is large, the probability of correctly extraction is high. In other words, we can roughly judge whether locating is accurate or not as well as the watermark is correct or not by the difference of the watermark pairs. The setting of th will be discussed in Section V-C.

The whole extracting process is described as Algorithm 2.

IV. Improvements for Fast Extraction

The above algorithm is relatively time consuming at the extraction side, so it is suitable for those scenarios where the extraction time requirement is not very strict, such as leak tracking. However, for the scene of information retrieval, it cannot meet the requirements well. Since information retrieving needs real-time extraction, we need to speed up our extraction speed. The main time-consuming part at the extraction side is that we need to increase the number of keypoints to avoid the disappearance of keypoints. So the core is that the keypoints are not robust enough to resist the screen-shooting distortions. Therefore, we proposed the SIFT intensity enhancement algorithm to improve the intensity of keypoint so that the number of extracting regions as well as the extraction time can be effectively reduced.

Algorithm 2 Extraction Algorithm

Input: Watermarked Image I'.
Output: Watermark sequences
1: Correct the perspective distortions and generate the image to be extracted I.
2: Locate the keypoints $\mathbf{p}_i (1 \leq i \leq n)$ of I by applying I-SIFT.
3: Filter out feature regions $A(\mathbf{p}_j)(1 \leq j \leq 2 \cdot k)$ to be extracted.
4: **for all** $A(\mathbf{p}_j)$ **do**
5: Extract the watermark group $\overrightarrow{W_j}$;
6: **end for**
7: Extract the watermark matrix w with Eqs.(11), (12), (13), (14), (15).
8: Reshape the watermark matrix w in to a sequence and decode it by BCH code.
9: **return** Watermark sequences.

A. The Modifications of SIFT Points

Li *et al.* [29] proposed a SIFT keypoint editing algorithm that can remove or add SIFT keypoints. But the intensity of the new keypoints cannot be controlled and there are some points that cannot be modified. So in our scheme, we make some adjustments on Li's algorithm [29] to generate the SIFT keypoint intensity editing algorithm. By applying such algorithm, we can enhance the intensity of the embedding points and weaken the intensity of the points with large intensity but not chosen to embed the message.

1) SIFT Keypoints Intensity Enhancement: For a fixed octave o, let $\mathbf{p}_o = (x_o, y_o, \sigma_{s_o})$ represent the index of the SIFT feature point and we denote $S_o = \{(x, y, \sigma_s) || x - x_o| \leq 1, |y - y_o| \leq 1, |s - s_o| \leq 1, x, y, s \in \mathbb{Z}\}$ as the group of the $3 \times 3 \times 3$ points centered at $\mathbf{p}_o$. In order to enhance the intensity of $\mathbf{p}_o$, we need to take different operations with respect to $\mathbf{p}_o$ being a maximum point or a minimum point. It's worth noting that the modification will bring distortions to the image, so in order to get the best visual quality, we treat the enhancement process as an optimization problem.

Let B_o denote an image patch of size $m \times m$ centered at (x_o, y_o) from the original image I, where in our scheme we set $m = 7$. I' denote the modified image, and the modified image patch is written as B'_o. The whole optimization problem is summarized as:

$$\min_{B'_o} \left\| B_o - B'_o \right\|_2^2$$
$$\text{s.t. } (C.1): \begin{cases} D_{I'}(\mathbf{p}_o) - D_I(\mathbf{p}_o) \geq \xi, & \mathbf{p}_o : maximum. \\ D_{I'}(\mathbf{p}_o) - D_I(\mathbf{p}_o) \leq -\xi, & \mathbf{p}_o : minimum. \end{cases}$$
$$(C.2): \text{No new feature points generated.}$$

where ξ is the intensity we set to be enhanced. Condition $(C.1)$ can be satisfied according to the above inequality. As for condition $(C.2)$, we guarantee it as follows. According to [28], for a fixed octave, there are 5 scales within it. As Fig. 8 indicates, we use $\{s | s \in [0, 1, 2, 3, 4]\}$ to denote the 5 scales. The keypoints only can be generated in the middle 3 scales.

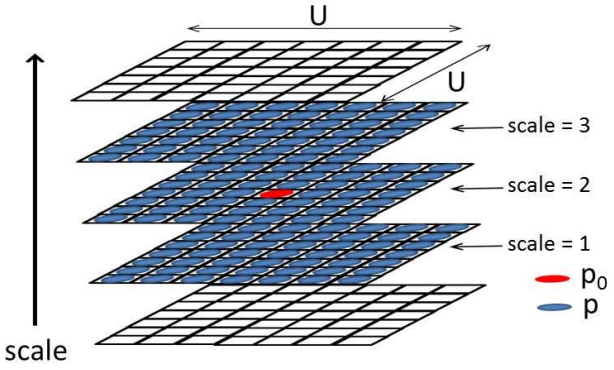


Fig. 8. The scales of keypoints.

Let T_o to be a cube with size $U \times U \times 3$ centered at $\mathbf{p}_o$ as Eq.(16) shows.

$$T_o = \left\{ (x, y, \sigma_s) \middle| |x - x_o| \leq \frac{U-1}{2}, |y - y_o| \leq \frac{U-1}{2}, 1 \leq s \leq 3 \right\} \backslash \{\mathbf{p}_o\}, \quad (16)$$

where U is set as 7 in our scheme. For each $\{\mathbf{p} | \mathbf{p} \in T_o\}$, we extract a $3 \times 3 \times 3$ cube $S_\mathbf{p}$ centered at $\mathbf{p}$ and compute

$$\mathbf{x}^\mathbf{p}_{min} = \underset{\mathbf{p} \in S_\mathbf{p} \backslash \{\mathbf{p}_o\}}{\text{argmin}} D_I(\mathbf{p})$$
$$\mathbf{x}^\mathbf{p}_{max} = \underset{\mathbf{p} \in S_\mathbf{p} \backslash \{\mathbf{p}_o\}}{\text{argmax}} D_I(\mathbf{p}), \quad (17)$$

to avoid generating a new, $\mathbf{p}$ should satisfy

$$D_{I'}(\mathbf{x}^\mathbf{p}_{min}) \leq D_{I'}(\mathbf{p}) \leq D_{I'}(\mathbf{x}^\mathbf{p}_{max}). \quad (18)$$

In summary, the optimization problem can be rewritten as

$$\min_{B'_o} \left\| B_o - B'_o \right\|_2^2$$
$$\text{s.t. } (C.1): \begin{cases} D_{I'}(\mathbf{p}_o) - D_I(\mathbf{p}_o) \geq \xi, & \mathbf{p}_o : maximum. \\ D_{I'}(\mathbf{p}_o) - D_I(\mathbf{p}_o) \leq -\xi, & \mathbf{p}_o : minimum. \end{cases}$$
$$(C.2): D_{I'}(\mathbf{x}^\mathbf{p}_{min}) \leq D_{I'}(\mathbf{p}) \leq D_{I'}(\mathbf{x}^\mathbf{p}_{max}), \quad \forall \mathbf{p} \in T_o$$

In our implementation, the function 'fmincon' provided by Matlab v.R2015b is used to solve the optimization problem.

2) SIFT Keypoints Intensity Weaken: This process is similar to the enhancement process, but the intensity of the target point should be weaken. So the optimization process can be written as

$$\min_{B'_o} \left\| B_o - B'_o \right\|_2^2$$
$$\text{s.t. } (C.1): \begin{cases} D_{I'}(\mathbf{p}_o) - D_I(\mathbf{p}_o) \leq \xi, & \mathbf{p}_o : maximum. \\ D_{I'}(\mathbf{p}_o) - D_I(\mathbf{p}_o) \geq -\xi, & \mathbf{p}_o : minimum. \end{cases}$$
$$(C.2): D_{I'}(\mathbf{x}^\mathbf{p}_{min}) \leq D_{I'}(\mathbf{p}) \leq D_{I'}(\mathbf{x}^\mathbf{p}_{max}), \ \forall \mathbf{p} \in T_o$$

B. Message Embedding and Extraction

The embedding procedure is generally the same as the previous one, but before message embedding, we need to modify the intensity of the chosen SIFT keypoints. Specifically, when k keypoints are selected, we need to choose j points with

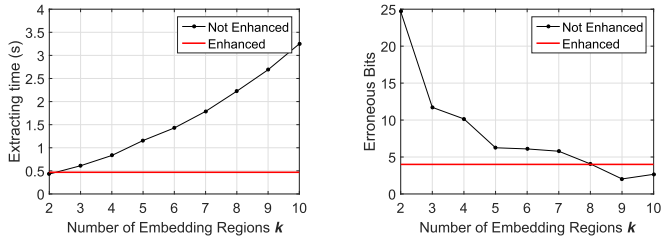


Fig. 9. The comparison of extraction performance. (a) Extracting time. (b) Erroneous bits.

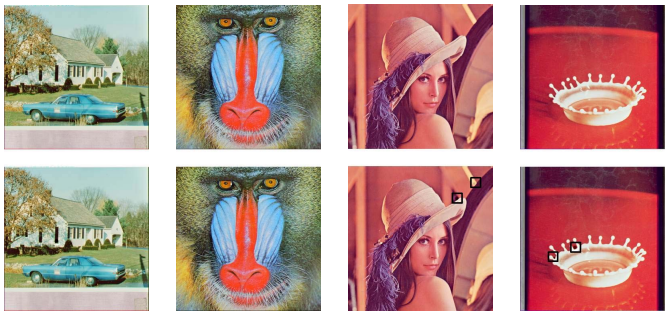


Fig. 10. The difference between original images and modified images. *Top row:* Host Images; *Bottom row:* Modifided Images. (a) Image 1. (b) Image 2. (c) Image 3. (d) Image 4.

the biggest intensity to enhance, where j is set as 2 in our experiment. Then the rest $n - j$ points should be weakened. The j feature regions we choose are used to embed the watermark. In the proposed method, the enhanced intensity ξ is set as 15. The extracting process is same as the previous method. But we just need to search for $2 \cdot j$ feature regions instead of $2 \cdot k$ regions.

C. The Analysis on Advantage and Disadvantage

1) The Advantages After Modification: Fig. 9a indicates the extracting time with different setting of embedding regions k between the modified image and the unmodified image. It's easy to see that when $k \geq 3$, the extracting speed of the modified image is faster than the unmodified one for 0.4s at least. Fig. 9b shows the extracting erroneous bits (EB) with different set of k. We can see only when k is larger than 5, the EB of the unmodified image is close to that of the modified image. So this scheme can improve the extraction speed and ensure the extraction accuracy with fewer blocks embedded.

2) The Disadvantages After Modification: Fig. 10 shows 4 different images in USC-SIPI image database [34] before and after modification. From the two images in Fig. 10a and Fig. 10b, we can see that the modification of the feature intensity has almost no effect on the image quality. But in Fig. 10c and Fig. 10d, the image quality declines after modification, which means we need to evaluate whether the image is suitable for the editing algorithm. In addition, the optimization problem of modifying the SIFT intensity is very time-consuming, which is also one of the disadvantages of this algorithm.

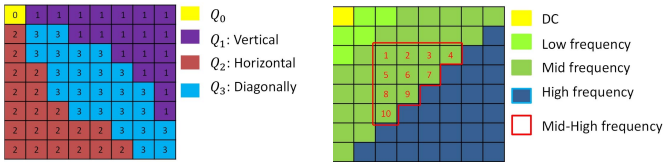
3) Conclusion: SIFT keypoint editing algorithm can effectively reduce the extraction time while ensuring the accuracy of the extraction. But the modification process costs a lot of time, which means the host image cannot be generated in the real time. And for some image, the algorithm may bring visual distortion. Therefore, for the scene of leak tracking, which requires good image quality and real-time embedding, the editing algorithm cannot fit. But for the scene of information retrieval, the algorithm works well. Because in this application, the embedded image can be generated offline. Besides, the increase in extraction speed brought by this algorithm is the most important part to realize information retrieval. In addition, although the visual quality of the image may be affected, it is still much better than that of 2-D barcode. So in general, this algorithm sacrifices the embedded time to reduce the extraction time, so it can be well adapted to scenes that require extraction speed but do not require high embedding speed and high visual quality.

V. Experimental Results and Analysis

The following experimental conclusions are based on the method without modifying the SIFT feature intensity. In our experiments, we choose $a = 8$, $b = 8$. The error correction code (ECC) we choose is BCH (64,39) which can correct 4 bit errors. So if erroneous bits is less than 4, we can successfully recover the correct watermark sequence. And the message bits we can encode is 39 bits. For the scene of leak tracking, 39 bits can support at most $2^{39} = 549755813888$ devices addressing, which is enough for a company or a workshop. The monitor we used in our experiment is 'AOC-G2770PF', and the mobile phone we used is 'iPhone 6s'. In Section V-A, we will show and discuss the experimental results of DCT middle-high coefficient pair selection. In Section V-B, the selection of embedding regions k will be discussed. In Section V-C, we will verify the correctness of our extraction method and show the experimental results of selecting the threshold th. In Section V-D, the proposed scheme is compared with some other state-of-the-art cross-media watermarking algorithms for screen-camera process.

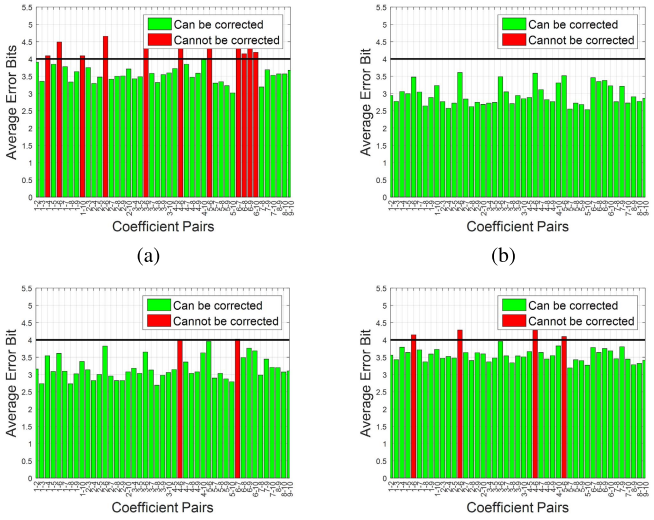
A. The Selection of DCT Middle-High Frequency Coefficient Pair

As analyzed in Section III-A3, we need to choose a pair of DCT coefficients for embedding. Liu *et al.* [26] experimentally proved that the DCT coefficients with mid-high frequency affected little by moiré pattern. That is to say, embedding watermark in DCT coefficients with mid-high frequency can effectively reduce the influence of moiré pattern. Besides, Zeng and Lei [47] proposed that the human eye has a lower sensitivity to the diagonal texture, and Lou *et al.* [48] roughly divide DCT coefficients into three parts, as shown in Fig. 11a, the area Q_1 and Q_2 mainly denote the texture in vertical and horizontal direction, while the area Q_3 mainly denotes the texture in diagonally direction. So in order to achieve invisibility, the coefficients should be in area Q_3. Combing the invisibility and the robustness, we select the 10 coefficients (The red part of Fig. 11b) in the middle-high frequency as



(a) (b)

Fig. 11. The DCT coefficients of 8×8 patch. (a) Texture direction. (b) Frequency division.



(a) (b) (c) (d)

Fig. 12. The average error bit corresponding to different coefficient pairs. (a) Error bits of database [30]. (b) Error bits of database [31]. (c) Error bits of database [32]. (d) Error bits of database [33].

well as in area Q_3 as the candidate coefficients to embed the watermark. Then we take out any two of these coefficients to form 45 coefficient pairs for choosing the most appropriate coefficient pair. The images we used are : 100 images in BOSSbase ver1.01 database [30], 100 images in BOWS-2-OrigEP3 database [31], 100 images in ImageNet [32] and 96 images in database [33]. For each image, we fix 2 feature regions to embed and extract the watermark. The parameter d mentioned in Eq.(9) is fixed at 24 to avoid redundant interference. After embedding, we use the screen-camera model in [27] to simulate the screen-camera process and extract the watermark from the processed image. And the average extraction error bits corresponding to different coefficient pairs are shown in Fig. 12. The x-axis of Fig. 12 represents the selection of different coefficient pairs, and the y-axis represents the average erroneous bits of the watermark extracted from the images. Since the error correction capability is set to 4 bits, we need to select the DCT coefficient pairs with the average erroneous bits less than 4. Then among them, the pair of coefficients with the best visual quality should be selected after embedding. The visual quality of the image is measured by multi-scale structural similarity (MS-SSIM) [39]. The top five coefficient pairs that made the highest visual quality with different database are shown in Table I.

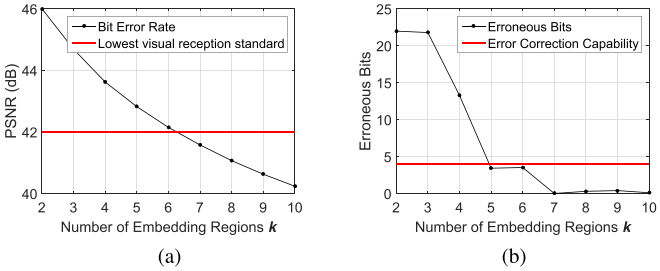
Obviously, $(4, 5)$ and $(5, 4)$ are the best choice. So we choose $(4, 5)$ and $(5, 4)$ as the embedding coefficient pair. And

TABLE I

THE FIVE PAIRS OF COEFFICIENTS FOR GREATEST VISUAL QUALITY IN DIFFERENT IMAGE DATABASE

MS-SSIM order		1	2	3	4	5
Database [30]	Value	99.9542	99.9536	99.9531	99.9529	99.9526
	C_1	**(4,5)**	(4,5)	(3,5)	(5,3)	(5,4)
	C_2	**(5,4)**	(6,3)	(4,5)	(5,4)	(6,3)
Database [31]	Value	99.9649	99.9642	99.9635	99.9634	99.9631
	C_1	**(4,5)**	(5,4)	(3,6)	(4,5)	(5,4)
	C_2	**(5,4)**	(6,3)	(5,4)	(6,3)	(5,4)
Database [32]	Value	99.9574	99.9568	99.9565	99.9564	99.9563
	C_1	(4,5)	**(4,5)**	(3,6)	(3,5)	(5,4)
	C_2	(6,3)	**(5,4)**	(4,5)	(4,5)	(6,3)
Database [33]	Value	99.9659	99.9648	99.9638	99.9635	99.9634
	C_1	(4,5)	**(4,5)**	(5,4)	(3,6)	(3,6)
	C_2	(6,3)	**(5,4)**	(6,3)	(4,5)	(5,4)

Fig. 13. The experimental sketch of the recapture process.



(a) (b)

Fig. 14. The influence of different embedding region numbers. (a) PSNR with different number of embedding regions. (b) Erroneous bits with different number of embedding regions.

the quantization step size of two coefficients q_1 and q_2 in JPEG compression table ($QF = 50$) are 51 and 56.

B. The Selection of the Embedding Region's Number k

The extraction method determines the number k of the embedding regions which should be no less than 2. So we performed a set of experiments on k from 2 to 10 in the color images of USC-SIPI image database [34]. Then the embedded image are displayed on the screen and we recapture them from the distance of $85cm$, as Fig. 13 demonstrated. The relationship between PSNR value and embedding number k is shown in the Fig. 14a and the relationship between the EB of extracted watermark and embedding number k is shown in the Fig. 14b.

TABLE II
THE IMAGE RATIO IN DIFFERENT EMBEDDING CONDITIONS

Embedding	Remaining Keypoints										
Keypoints	0	1	2	3	4	5	6	7	8	9	10
2	18.8%	56.2%	25.0%	-	-	-	-	-	-	-	-
3	12.5%	31.2%	43.8%	12.5%	-	-	-	-	-	-	-
4	0.0%	18.8%	25.0%	43.7%	12.5%	-	-	-	-	-	-
5	0.0%	0.0%	25.0%	37.4%	31.3%	6.3%	-	-	-	-	-
6	0.0%	0.0%	12.5%	43.8%	18.8%	12.4%	12.5%	-	-	-	-
7	0.0%	0.0%	12.5%	18.8%	18.8%	37.4%	0.0%	12.5%	-	-	-
8	0.0%	0.0%	6.3%	12.5%	12.4%	25.0%	25.0%	6.3%	12.5%	-	-
9	0.0%	0.0%	0.0%	12.5%	18.8%	18.8%	12.4%	25.0%	12.5%	0.0%	-
10	0.0%	0.0%	0.0%	0.0%	12.5%	18.8%	12.5%	25.0%	25.0%	0.0%	6.2%

Fig. 14a shows that when k is greater than 6, the PSNR value is less than 42dB. Since we think 42dB is the lowest visual reception standard, so k should be no larger than 6. Fig. 14b shows that as k increases, the BER is decreasing. But when $k = 5$, the EB is lower than 4 which means the extracted watermark can be recovered correctly. And the increase of k will not affect the correct extraction of the message. Besides, in order to measure the impact of the screen-shooting process on the embedding keypoints. We recorded the number of embedding keypoints remaining in the image after screen-shooting process. Table II indicates the relationship between the embedding keypoints and the remaining keypoints, when embedding regions are more than 5, there will be at least two complete blocks reserved in the screen-shooting images. Since we need at least 2 keypoints to extract in the proposed method, if two more keypoints remaining in the image, we believed the keypoints in this image are not affected after screen-shooting process. So combining the result of PSNR and EB, k is set as 5 in our experiment.

C. The Selection of the Threshold th

In our extraction method, we point out that when locating accurately, the difference of the watermark pairs will be small. So we recorded the watermark pair whose difference is less than th. So the threshold th is important to the extraction accuracy. To determine the most appropriate threshold th, we performed some experiments and use Neyman-Pearson theories to analyze the results. Since the extraction process requires that at least two keypoints should be accurately located, if the number of accurately located keypoints (ALK) is larger than 2, we believe that the image is accurately located, otherwise, we consider the image is inaccurately located. Let P_I denote the keypoints group of the original image I, $p_I^i \in P_I, i \in [1, k]$ and $P_{I'}$ denote the keypoints group of the extracted image I', $p_{I'}^j \in P_{I'}, j \in [1, 2k]$. So ALK is measured by

$$\begin{aligned} &\mathrm{ALK} = P_I \cap P_{I'} \\ &\text{s.t.} \begin{cases} |x_I^i - x_{I'}^j| \le 1 \\ |y_I^i - y_{I'}^j| \le 1 \end{cases} \end{aligned}$$

where (x_I^i, y_I^i) is the coordinates of the point p_I^i. Then we choose 1000 images which can be located accurately in BOSSbase ver1.01 database [30] to perform the experiment. The embedding coefficients are chosen as (4,5) and (5,4), the number of embedding regions k is set as 5, and the distance from the screen to the mobile phone is $85cm$. The minimum

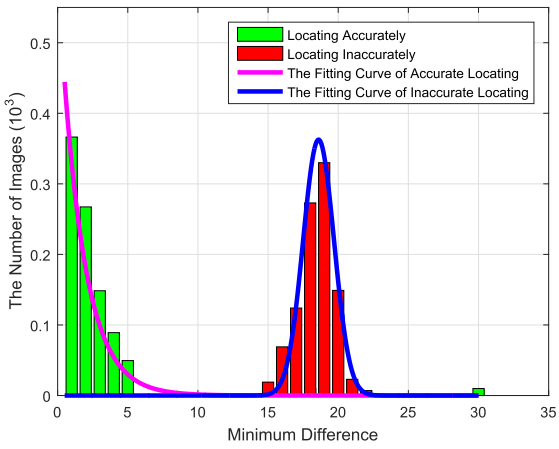


Fig. 15. The distributions of minimum difference.

difference (MD) of watermark pairs in each screen-shooting image is recorded in the extraction process. The MD is defined by

$$MD = \underset{\substack{w_{i\alpha} \in \mathbf{W}_i, w_{j\beta} \in \mathbf{W}_j \\ i,j \in [1,2\ldots k], i \ne j}}{\operatorname{argmin}} [diff(w_{i\alpha}, w_{j\beta})] \tag{19}$$

It is worth noting that for the 1000 images, we need to perform two rounds of extraction. In the first round, we used the $2k$ keypoints which are located accurately to represents the accurately located MD distribution. In the second round, $2k$ randomly selected keypoints are used to generate the inaccurately located MD distribution. Fig. 15 shows the accurately located and the inaccurately located MD distributions. When locating accurately, MD is approximated to exponential distribution. Otherwise, MD is approximately fit Gaussian distribution. So we can use MD to estimate whether the image is accurately located or not. And the estimating problem can be restated as a hypothesis testing problem

H_0: Locate accurately.

H_1: Locate inaccurately.

Here, H_0 denotes the accurately locating hypothesis and H_1 denotes the inaccurately locating hypothesis. And the likelihood function can be written as

$$f(x|H_0) = \lambda \cdot \mathrm{e}^{-\lambda x} \tag{20}$$

$$f(x|H_1) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{21}$$

In case of Neyman-Pearson (NP) approach in signal detection, the decision rule is defined as

$$\frac{f(x|H_1)}{f(x|H_0)} \underset{H_1}{\overset{H_0}{\gtrless}} \tau, \tag{22}$$

that is

$$x \underset{H_1}{\overset{H_0}{\gtrless}} g(\tau), \tag{23}$$

where τ is the decision threshold and $g(\tau)$ is a function of τ which is determined by Eqs.(20),(21),(22). Let $th = g(\tau)$, so the decision rule is equivalent to

$$x \underset{H_1}{\overset{H_0}{\gtrless}} th. \tag{24}$$

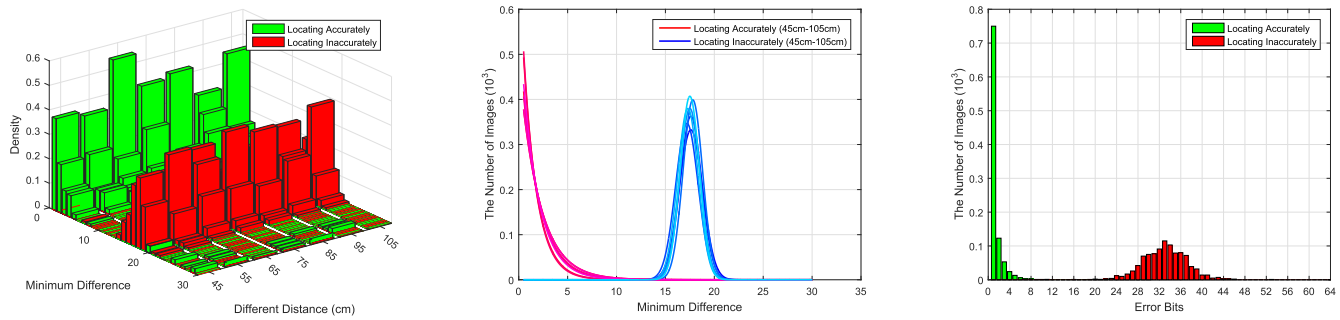


Fig. 16. Distributions of minimum difference and error bits. (a) Distributions of minimum difference with different distance. (b) Fitting curves of minimum difference with different distance. (c) The distributions of error bits.

TABLE III
THE THRESHOLD OF DIFFERENT DISTANCE

Distance	45cm	55cm	65cm	75cm	85cm	95cm	105cm
Threshold	5.57	5.23	5.13	5.85	5.74	5.76	5.68

Then according to the given probability of false-alarm (denoted by α_0), which is defined as

$$\alpha_0 = \int_{th}^{\infty} f(x|H_0)\,\mathrm{d}\,x, \tag{25}$$

we can calculate the parameter th. In this paper, α_0 is set as 10^{-2}. And according to the fitting curves, we obtain the parameter values as $\lambda = 0.87$, $\mu = 17.6$, $\sigma = 1.1$. So the threshold th can be calculated as 5.67 according to Eq.(25), which means when MD is no larger than 6, the image is considered as locating accurately, otherwise, it is locating inaccurately. And the detection probability, that is, the probability of locating inaccurately is calculated as 0.99 when MD is no larger than 6.

But considering that the threshold value may be affected by the screen-shooting conditions, we conducted a series of experiments at different distances to observe the change of threshold value. We set the distance is from $45cm$ to $105cm$ with the step of $10cm$. We captured 100 images in each distance and recorded the distributions of MD as before. The experimental results are shown in Fig. 16.

Fig. 16a shows the MD distributions with different shooting conditions, and Fig. 16b shows the fitting curve of MD distributions with different shooting conditions. As we can see, the difference of each MD distribution curves is very small. No matter the distance, the MD is always exponentially distributed when locating accurately, and the MD is always Gaussian distributed when locating inaccurately. So according to the calculation of NP hypothesis testing, we get the threshold value for each distance, as shown in Table III. To meet the requirements of false alarm in all conditions, we choose $th = 6$ as the threshold. In addition, we also calculated the distribution of the average EB of the extracted watermark in each image when locating accurately and inaccurately. Fig. 16c indicates the distributions of the average error bit of the extracted watermark in each image. Obviously, when the keypoints are located accurately, the error bit is mainly distributed in $[0, 4]$ which means it can be corrected by ECC. But when the image is located inaccurately, the error bit obey the Gaussian distribution with a mean of 32 which means the watermark sequence is almost impossible to be recovered successfully. Therefore, when locating is accurate, the extracted watermark can be regarded as a correct watermark with high probability, and when locating is inaccurate, the extracted watermark can be regarded as incorrect with high probability.

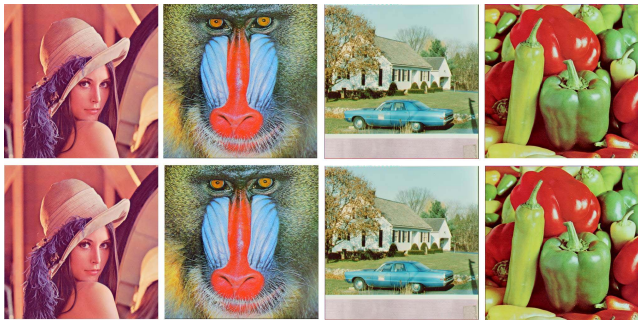


Fig. 17. *Top row:* Host Images; *Bottom row:* Watermarked Images.

D. Comparisons With Previous Methods

In this section, we show and discuss the comparative experimental results. The image database we used for comparison with other schemes under various of attacks is USC-SIPI image database [34].

Fig. 17 shows the results of 4 nature images and the corresponding watermarked images generated with the proposed method. We compare the proposed scheme with three state-of-the-art watermarking schemes (Kang *et al.* [16], Pramila *et al.* [23], Nakamura *et al.* [18]). Note that the method proposed by Kang *et al.* [16] are designed for print-scanning process, the methods proposed by Pramila *et al.* [23] and Nakamura *et al.* [18] are designed for print-camera process. Although these algorithms are applicable in the framework that they are designed for, the proposed approach works better for the problem area of screen-shooting process. The transverse comparisons of robust tests with different algorithms in screen-shooting process are illustrated by the following experiment. For fair comparison with other schemes, PSNR values of

TABLE IV

THE IMAGE EMBEDDED WITH DIFFERENT METHODS

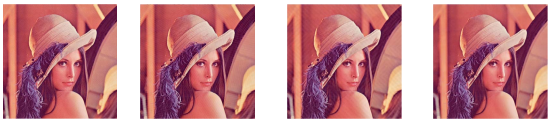
Methods	Kang *et.al* [16]	Pramila *et.al* [23]	Nakamura *et.al* [18]	Proposed
Image				
PSNR (dB)	42.3665	42.3256	42.0019	42.3003

TABLE V

THE EXAMPLE OF SCREEN-SHOOTING IMAGES WITH DIFFERENT SHOOTING DISTANCE

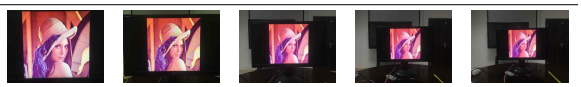
Distance	45cm	55cm	75cm	95cm	105cm
Recaptured					
Recovered					

TABLE VI

AVERAGE ERRONEOUS BITS OF THE EXTRACTED WATERMARKS WITH DIFFERENT SHOOTING DISTANCE

Distance(cm)	Kang *et.al* [16]	Pramila *et.al* [23]	Nakamura *et.al* [18]	Proposed	
	Erroneous bits	Erroneous bits	Erroneous bits	Erroneous bits	Message recovered?
45	14.94	32.00	10.50	**0.50**	Yes
55	26.50	30.19	9.44	**0.69**	Yes
65	24.37	28.62	17.81	**1.69**	Yes
75	23.81	28.94	8.75	**0.88**	Yes
85	19.88	28.38	18.44	**0.44**	Yes
95	16.25	26.56	19.06	**0.75**	Yes
105	29.38	31.00	19.50	**1.31**	Yes

the embedded images are set to the same level of 42.1 ± 0.03dB, for a more subjective visual assessment, the image ("*lena.tiff*") embedded with different methods are displayed in the Table IV.

1) The Impact of Distance on Robustness: Table V shows the examples of recaptured images in different distance and the corresponding recovered images. Table VI lists the average erroneous bits obtained with different schemes at different recapture distances and Fig. 18a indicates the average BER with different schemes. It is easy to see that our algorithm has a better performance than other methods in all the test distances. The EB are at least 15 bits lower than other algorithms. From Table V we can see that the phenomenon of moiré pattern occurs at close distance. When shooting out of $75cm$, there is almost no moiré pattern. But the moiré pattern has little effect on the algorithm of this paper, the EB of our algorithm are almost close to 1 bit, so the watermark is robust to distance changing and moiré pattern.

2) The Impact of Horizontal Perspective Angle on Robustness: Table VII shows the examples of recaptured images with different vertical perspective angle and the corresponding recovered images. Table VIII lists the average erroneous bits obtained with different schemes at the same shooting distance of $60cm$ but different shooting angles. Fig. 18b indicates the

TABLE VII

THE EXAMPLE OF SCREEN-SHOOTING IMAGES WITH DIFFERENT HORIZONTAL PERSPECTIVE ANLGE

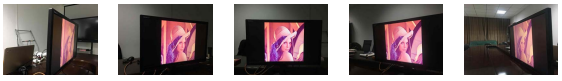
Angle	Left 65°	Left 30°	0°	Right 30°	Right 65°
Recaptured					
Recovered					

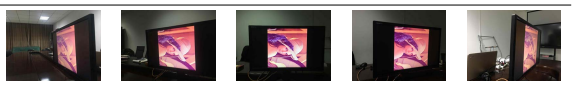
TABLE VIII

AVERAGE ERRONEOUS BITS OF THE EXTRACTED WATERMARKS WITH DIFFERENT HORIZONTAL ANGLES

Horizontal angle (°)	Kang *et.al* [16]	Pramila *et.al* [23]	Nakamura *et.al* [18]	Proposed	
	Erroneous bits	Erroneous bits	Erroneous bits	Erroneous bits	Message recovered?
Left 65	32.00	32.25	38.38	**18.94**	No
Left 60	22.00	31.56	12.56	**2.94**	Yes
Left 45	26.31	28.62	10.44	**0.69**	Yes
Left 30	17.81	27.50	8.44	**1.12**	Yes
Left 15	12.50	28.88	14.25	**2.19**	Yes
0	20.81	28.19	9.88	**1.38**	Yes
Right 15	16.44	24.94	8.88	**1.00**	Yes
Right 30	16.94	23.37	10.44	**3.81**	Yes
Right 45	12.75	18.31	10.13	**3.31**	Yes
Right 60	19.69	11.37	9.44	**3.88**	Yes
Right 65	**23.31**	24.81	36.69	24.81	No

TABLE IX

THE EXAMPLE OF SCREEN-SHOOTING IMAGES WITH DIFFERENT VERTICAL PERSPECTIVE ANLGE

Angle	Down 65°	Down 30°	0°	Up 30°	Up 65°
Recaptured					
Recovered					

average EB when shooting at different vertical perspective angles compared with different schemes. The watermark is robust to the angle range from $Left\_60°$ to $Right\_60°$ as shown in Fig. 18b. The EB of proposed algorithm are at least 7 bits lower compared with other schemes. And the EB is still within the acceptable range at most scenes, so the watermark is robust to the most of the vertical shooting angles.

3) The Impact of Vertical Perspective Angle on Robustness: Table IX shows the examples of recaptured images with different horizontal perspective angles and the corresponding recovered images. Table X lists the average erroneous bits obtained with different schemes at the same shooting distance of $60cm$ but different shooting angles. Fig. 18c indicates the average EB compared with different schemes. In Table VIII, $Up\_x°$ means the complementary angle between the shooting direction and the screen plane is above the image $x°$, similar as $Down\_x°$ means. From Fig. 18c, we conclude that the available shooting angle is from $Down\_45°$ to $Up\_60°$, watermarks are robust in these perspective angles. The EB

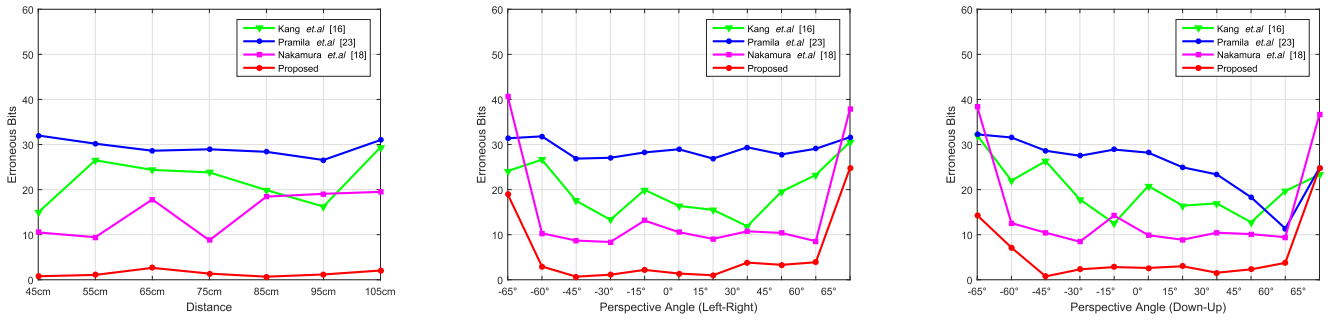


Fig. 18. Erroneous bits of different shooting conditions. (a) Erroneous bits of different distance. (b) Erroneous bits of different horizontal perspective angles. (c) Erroneous bits of different vertical perspective angles.

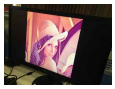
TABLE X

AVERAGE ERRONEOUS BITS OF THE EXTRACTED WATERMARKS WITH DIFFERENT VERTICAL ANGLES

Vertical angle (°)	Kang *et.al* [16]	Pramila *et.al* [23]	Nakamura *et.al* [18]	Proposed	
	Erroneous bits	Erroneous bits	Erroneous bits	Erroneous bits	Message recovered?
Down 65	24.12	31.38	40.69	**14.31**	No
Down 60	26.63	31.81	10.31	**7.13**	No
Down 45	17.50	26.87	8.69	**0.75**	Yes
Down 30	13.31	27.06	8.38	**2.31**	Yes
Down 15	19.94	28.25	13.13	**2.81**	Yes
0	16.38	28.94	10.56	**2.62**	Yes
Up 15	15.50	26.88	9.06	**3.00**	Yes
Up 30	11.88	29.38	10.75	**1.56**	Yes
Up 45	19.56	27.81	10.38	**2.31**	Yes
Up 60	23.25	29.06	8.56	**3.75**	Yes
Up 65	30.56	31.63	37.88	**24.69**	No

TABLE XI

THE EXAMPLES OF HANDHOLD SHOOTING

Handhold Situation	1	2	3	4
Recaptured				
Recovered				
Erroneous Bits	1	0	2	1

values of proposed scheme are below 5 bits which is at least 10 bits lower than other schemes. But when the shooting angle is greater than 60°, EB increase significantly. So the proposed algorithm is robust to most of the horizontal shooting angles.

4) The Impact of Handhold Shooting on Robustness: Table XI shows the examples of recaptured images in handhold shooting and the corresponding results of extraction. For the most scenes, the erroneous bits of the watermark are within the error-correcting capability, which makes the proposed algorithm also show good performance in practical applications.

E. Limitations

Although the algorithm works well for most images, it still has some limitations. One key limitation is that this scheme is dependent on scenes that flair well to SIFT feature extraction.

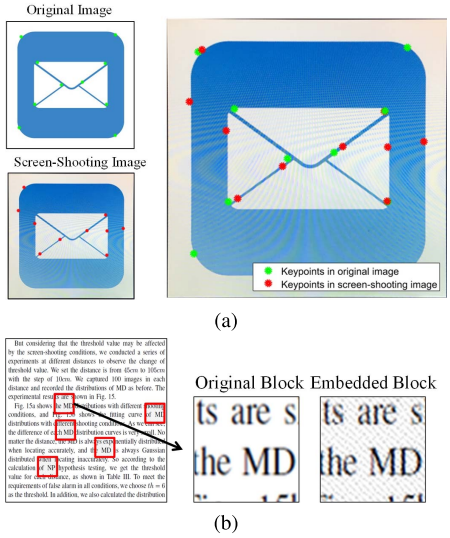


Fig. 19. The examples of failure cases. (a) The example of simple texture image. (b) The example of textual image.

So if the image has simple texture, the proposed algorithm does not work well on it. Because as shown in the Fig. 19a, the SIFT keypoints of the image with simple texture are not robust enough to keep the location of the keypoints unchanged before and after the screen-shooting process, so that the watermark region cannot be accurately located, which will badly affect the extraction process.

Another limitation is that the embedding algorithm will cause a lot visual distortions to binary images such as textual images. The embedding process in DCT domain will greatly affect the font of the text so as to affect the normal reading, as shown in Fig. 19b, so the algorithm is not suitable for documents.

VI. CONCLUSION AND FUTURE WORK

A robust watermarking scheme against screen-shooting process is proposed in this paper. We analyze the special distortions caused by the screen-shooting process. In order to resist the lens distortions, we propose an intensity based SIFT algorithm that can achieve accurate locating in distorted images. In addition, to avoid the loss of details caused by

light source distortion and moiré distortion, we proposed a small-size template method to embed the complete watermark information in the image repeatedly. To working in combination with the repeated embedding algorithm, at the extraction side, we propose an extraction scheme based on cross-validation, which can better guarantee the extraction accuracy. Furthermore, in order to achieve real-time extraction, we have also proposed an intensity enhancing scheme based on SIFT feature editing algorithm. The experimental results show that the proposed watermarking scheme achieves high robustness for screen-shooting process.

REFERENCES

[1] C. Chen, W. Huang, B. Zhou, C. Liu, and W. H. Mow, "PiCode: A new picture-embedding 2D barcode," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3444–3458, Aug. 2016.

[2] J. S. Seo and C. D. Yoo, "Image watermarking based on invariant regions of scale-space representation," *IEEE Trans. Signal Process.*, vol. 54, no. 4, pp. 1537–1549, Apr. 2006.

[3] M. A. Akhaee, S. M. E. Sahraeian, and C. Jin, "Blind image watermarking using a sample projection approach," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 883–893, Sep. 2011.

[4] M. Andalibi and D. M. Chandler, "Digital image watermarking via adaptive logo texturization," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5060–5073, Dec. 2015.

[5] M. Zareian and H. R. Tohidypour, "A novel gain invariant quantization-based watermarking approach," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 11, pp. 1804–1813, Nov. 2014.

[6] J. Fridrich, "Digital image forensics," *IEEE Signal Process. Mag.*, vol. 26, no. 2, pp. 26–37, Mar. 2009.

[7] Y. Xie, H. Tan, and K. Wang, "A novel color image hologram watermarking algorithm based on QDFT-DWT," in *Proc. Chin. Control Decis. Conf.*, May 2016, pp. 4349–4354.

[8] M. Alghoniemy and A. H. Tewfik, "Geometric invariance in image watermarking," *IEEE Trans. Image Process.*, vol. 13, no. 2, pp. 145–153, Feb. 2004.

[9] C.-S. Lu, S.-W. Sun, C.-Y. Hsu, and P.-C. Chang, "Media hash-dependent image watermarking resilient against both geometric attacks and estimation attacks based on false positive-oriented detection," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 668–685, Aug. 2006.

[10] S. Pereira and T. Pun, "Robust template matching for affine resistant image watermarks," *IEEE Trans. Image Process.*, vol. 9, no. 6, pp. 1123–1129, Jun. 2000.

[11] X. Kang, J. Huang, Y. Q. Shi, and Y. Lin, "A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 776–786, Aug. 2003.

[12] J. J. K. O'Ruanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking," *Signal Process.*, vol. 66, no. 3, pp. 303–317, 1998.

[13] D. He and Q. Sun, "A RST resilient object-based video watermarking scheme," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2004, pp. 737–740.

[14] C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y. M. Lui, "Rotation, scale, and translation resilient watermarking for images," *IEEE Trans. Image Process.*, vol. 10, no. 5, pp. 767–782, May 2001.

[15] D. Zheng, J. Zhao, and A. E. Saddik, "RST-invariant digital image watermarking based on log-polar mapping and phase correlation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 753–765, Aug. 2003.

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

[17] L. A. Delgado-Guillen, J. J. Garcia-Hernandez, and C. Torres-Huitzil, "Digital watermarking of color images utilizing mobile platforms," in *Proc. IEEE 56th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2013, pp. 1363–1366.

[18] T. Nakamura, A. Katayama, M. Yamamuro, and N. Sonehara, "Fast watermark detection scheme for camera-equipped cellular phone," in *Proc. 3rd Int. Conf. Mobile Ubiquitous Multimedia*, New York, NY, USA, Oct. 2004, pp. 101–108.

[19] A. Katayama, T. Nakamura, M. Yamamuro, and N. Sonehara, "New high-speed frame detection method: Side trace algorithm (STA) for i-appli on cellular phones to detect watermarks," in *Proc. 3rd Int. Conf. Mobile Ubiquitous Multimedia*, New York, NY, USA, Oct. 2004, pp. 109–116.

[20] W.-G. Kim, S. H. Lee, and Y.-S. Seo, "Image fingerprinting scheme for print-and-capture model," in *Proc. Pacific-Rim Conf. Multimedia*. Springer, 2006, pp. 106–113.

[21] C. Chen, B. Zhou, and W. H. Mow, "RA code: A robust and aesthetic code for resolution-constrained applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 11, pp. 3300–3312, Nov. 2018.

[22] A. Pramila, A. Keskinarkaus, and T. Seppänen, "Reading watermarks from printed binary images with a camera phone," in *Proc. IWDW*, Guildford, U.K., Aug. 2009, pp. 227–240.

[23] A. Pramila, A. Keskinarkaus, and T. Seppänen, "Toward an interactive poster using digital watermarking and a mobile phone camera," *Signal, Image Video Process.*, vol. 6, pp. 211–222, Jun. 2012.

[24] A. Pramila, A. Keskinarkaus, V. Takala, and T. Seppänen, "Extracting watermarks from printouts captured with wide angles using computational photography," *Multimedia Tools Appl.*, vol. 76, no. 15, pp. 16063–16084, Sep. 2016.

[25] A. Pramila, A. Keskinarkaus, and T. Seppänen, "Increasing the capturing angle in print-cam robust watermarking," *J. Syst. Softw.*, vol. 135, pp. 205–215, Jan. 2018.

[26] F. Liu, J. Yng, and H. Yue, "Moiré pattern removal from texture images via low-rank and sparse matrix decomposition," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2015, pp. 1–4.

[27] P. Schaber, S. Kopf, S. Wetzel, T. Ballast, C. Wesch, and W. Effelsberg, "CamMark: Analyzing, modeling, and simulating artifacts in camcorder copies," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 11, no. 2, pp. 42:1–42:23, Feb. 2015.

[28] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, Sep. 1999, pp. 1150–1157.

[29] Y. Li, J. Zhou, A. Cheng, X. Liu, and Y. Y. Tang, "SIFT keypoint removal and injection via convex relaxation," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 8, pp. 1722–1735, Aug. 2016.

[30] P. Bas, T. Filler, and T. Pevný, "'Break our steganographic system': The ins and outs of organizing BOSS," in *Proc. 13th Int. Workshop Inf. Hiding*. Springer, May 2011, pp. 59–70.

[31] P. Bas and T. Furon. *BOWS-2 (Break Our Watermarking System)*, Accessed: Jul. 2007. [Online]. Available: http://bows2.ec-lille.fr/

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, Jun. 2009, pp. 248–255

[33] *Related Images of the Experiments*. Accessed: Nov. 2018. [Online]. Available: http://decsai.ugr.es/cvg/dbimagenes/g512.php

[34] University of Southern California. *The USC-SIPI Image Database, Signal and Image Processing Institute*. Accessed: Jul. 2013. [Online]. Available: http://sipi.usc.edu/database

[35] C. Ken and P. Gent, "Image compression and the discrete cosine transform," College of the Redwoods, Tech. Rep., 1998. Accessed: Nov. 2018. [Online]. Available: https://www.math.cuhk.edu.hk/ lmlui/dct.pdf

[36] M. Moosazadeh and A. Andalib, "A new robust color digital image watermarking algorithm in DCT domain using genetic algorithm and coefficients exchange approach," in *Proc. IEEE 2nd Int. Conf. Web Res. (ICWR)*, Apr. 2016, pp. 19–24.

[37] K. Ramanjaneyulu, P. Pandarinath, and B. R. Reddy, "Robust and oblivious watermarking based on swapping of DCT coefficients," *Int. J. Appl. Innov. Eng. Manage.*, vol. 2, no. 7, pp. 445–452, Jul. 2013.

[38] M. Hamid and C. Wang, "A simple image-adaptive watermarking algorithm with blind extraction," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, May 2016, pp. 1–4.

[39] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th IEEE Asilomar Conf. Signals, Syst. Comput.*, Nov. 2003, pp. 1398–1402.

[40] T. Tayebe and M. E. Moghaddam, "A new visual cryptography based watermarking scheme using DWT and SIFT for multiple cover images," *Multimedia Tools Appl.*, vol. 75, no. 14, pp. 8527–8543, 2016.

[41] K. M. Singh, "A robust rotation resilient video watermarking scheme based on the SIFT," *Multimedia Tools Appl.*, vol. 77, no. 13, pp. 16419–16444, Jul. 2018.

[42] X.-J. Wang and W. Tan, "An improved geometrical attack robust digital watermarking algorithm based on SIFT," in *Proc. 6th Int. Asia Conf. Ind. Eng. Manage. Innov.*, 2016, pp. 209–217.

[43] K.-L. Hua, B.-R. Dai, K. Srinivasan, Y.-H. Hsu, and V. Sharma, "A hybrid NSCT domain image watermarking scheme," *EURASIP J. Image Video Process.*, vol. 2017, no. 1, p. 10, 2017.
[44] T. Li, C. An, X. Xiao, A. T. Campbell, and X. Zhou, "Real-time screen-camera communication behind any scene," in *Proc. 13th Annu. Int. Conf. Mobile Syst., Appl., Services*, 2015, pp. 197–211.
[45] V. Nguyen *et al.*, "High-rate flicker-free screen-camera communication with spatially adaptive embedding," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.
[46] M. Iwata, N. Mizushima, and K. Kise, "Practical watermarking method estimating watermarked region from recaptured videos on smartphone," *IEICE Trans. Inf. Syst.*, vol. E100-D, no. 1, pp. 24–32, 2017.
[47] W. Zeng and S. Lei, "Digital watermarking in a perceptually normalized domain," in *Proc. Conf. Rec. 33rd Asilomar Conf. Signals, Syst., Comput.*, vol. 2, Oct. 1999, pp. 1518–1522.
[48] O. Lou, S. Li, Z. Liu, and S. Tang, "A novel multi-bit watermarking algorithm based on HVS," in *Proc. 6th Int. Symp. Parallel Archit., Algorithms Program. (PAAP)*, Jul. 2014, pp. 278–281.

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

Weiming Zhang received the M.S. and Ph.D. degrees from the Zhengzhou Information Science and Technology Institute, China, in 2002 and 2005, respectively. He is currently 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.

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

Hao Cui received the dual B.S. degrees in geophysics and computer science and technology from the University of Science and Technology of China in 2017, where he is currently pursuing the M.S. degree in electronics and communication engineering. His research interests include image watermarking and information hiding.

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