Computers, Materials & Continua
DOI:10.32604/cmc.2022.023955
Article

Tech Science Press

Robust Watermarking of Screen-Photography Based on JND

Siyu Gu[1], Jin Han[1,*], Xingming Sun[1,2] and Yi Cao[1,3]

[1]Nanjing University of Information Science & Technology, Nanjing, 210044, China
[2]Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing, 201144, China
[3]Department of Electrical and Computer Engineering, University of Windsor, 401 Sunset Avenue, Windsor, ON, N9B 3P4, Canada
*Corresponding Author: Jin Han. Email: hjhaohj@126.com

Received: 28 September 2021; Accepted: 08 November 2021

Abstract: With the popularity of smartphones, it is often easy to maliciously leak important information by taking pictures of the phone. Robust watermarking that can resist screen photography can achieve the protection of information. Since the screen photo process can cause some irreversible distortion, the currently available screen photo watermarks do not consider the image content well and the visual quality is not very high. Therefore, this paper proposes a new screen-photography robust watermark. In terms of embedding region selection, the intensity-based Scale-invariant feature transform (SIFT) algorithm used for the construction of feature regions based on the density of feature points, which can make it more focused on the key content of the image; in terms of embedding strength, the Just noticeable difference (JND) model is applied to limit the intensity of the watermark embedding according to the luminance and texture of the picture to balance robustness and invisibility; after embedding watermark, the coefficients in the neighborhood are again adjusted with optimal constraints to improve the accuracy of watermark extraction. After experiments, it is shown that the method we proposed can improve the correct rate of watermark extraction, the quality of the visual aspect of the watermarked picture is also improved.

Keywords: Screen photography; robust watermarking; SIFT; JND

1 Introduction

Currently, as smartphones become more popular and widely used, people can easily record some important information they need by taking photos with their phones. However, the images stored by taking photos can also be easily copied, transmitted and restored, which will lead to leakage, malicious theft and illegal distribution of some important information, causing a series of security problems. As digital watermarking technology has been widely used in our work and life, and it enables copyright protection and leakage tracking. Therefore, in order to protect the image information in the screen from malicious leakage and to be able to track the source of leakage in time after the leakage, we can add a watermark to the picture. Once the picture is leaked, a watermark can be extracted from the picture and we can locate the source of the leak based on the watermark information to narrow down the scope of the investigation and thus protect important information.

The technology of digital watermarking is divided into two domains, namely spatial domain and transform domain. Embedding is done in the spatial domain by modifying pixel values and commonly used spatial domain techniques are Least Significant Bit(LSB) [1], additive based techniques etc. And in the transform domain, where pixel values are transformed into frequency coefficients and then is used to embed, commonly used techniques are Discrete Cosine Transform(DCT) [2,3], Discrete Wavelet Transform(DWT) [4], Discrete Fourier Transform(DFT) [5], there are some improvements on these basic techniques such as Redundant Discrete Wavelet Transformation(RDWT) [6], Quaternion Discrete Fourier transform(QDFT) [7], etc. Sometimes multiple frequency techniques are used in combination such as DCT-DWT [8,9], Dual Tree Complex Wavelet Transform- Discrete Cosine Transform(DTCWT-DCT) [10] etc.

The screen photography process, which is the process of capturing a screen with a mobile phone and generating an image, undergoes analog-to-digital and digital-to-analog conversion attacks [11] and also causes many distortions. In past years, many researchers have worked on print-scanning resilient watermarking schemes [12,13] as well as print-camera resilient watermarking schemes [14–16], but due to the special nature of screen photography, experiments have shown that the majority of watermarking schemes do not work for screen photography processes, so some new watermarking schemes are needed.

Screen-photography watermarking algorithms have been investigated by several researchers in recent years. Fang et al. [17] presented a screen-photography watermarking scheme that uses an intensity-based SIFT algorithm to determine embedding region, uses the small size of the template to embed the watermark over and over into different areas, and extracts watermarks using cross-validation. This method has high robustness to watermarks obtained from screen photography, but requires manually finding the four vertices of the image for correction; the scheme does not apply to images with simple textures due to the limitations of the SIFT algorithm; and the scheme causes large visual distortion for binary images. Chen et al. [18] proposed screenshot robust watermarking for satellite images, proposing to use DFT to embed to deal with the quality decline due to screenshots, and using the synchronous response index for estimating the appropriate scale level as well as the location of the synchronous watermark during extraction; the scheme is robust to common attacks and screenshot attacks. Chen et al. [19] presented a feature synchronization based watermarking scheme for screenshots, using Gaussian functions, improved Harris-Laplace detectors and accelerated robust feature orientation descriptors to construct embedding regions for watermark synchronization, and also using non-rotational embedding methods and preprocessing methods to modulate the DFT coefficients, which is robust to screenshot attacks and also to attacks with additional common desynchronization attacks on screenshots is effective. Chen et al. [20] proposed a new scheme combining encryption and screen protection. The scheme first embeds the watermark in the DFT domain and then generates the second watermark based on two-dimensional coding and inverse DFT. A segmentation encryption algorithm based on chaotic mapping is proposed to enhance the robustness of the encrypted watermark, and the watermark synchronization algorithm based on frame position is used in the watermark extraction process. The scheme is safe and reliable, and has high robustness.

Several of the above methods embed watermarks by performing modifications in the frequency domain, such as the DCT and DFT domains. In recent years, deep learning methods have also been shown to be able to undertake watermark embedding and extraction. Fujihashi et al. [21] proposed to embed watermarks into images using a deep Convolutional Neural Networks encoder and decoder for watermarking, which has higher throughput than DCT based neural network watermarking. Zhong et al. [22] proposed a watermarking system based on Deep Neural Networks(DNN) using an unsupervised structure and a new loss calculation method, which is capable of extracting watermarks in camera captured images as well, with good practicality and robustness. Jia et al. [23] proposed a new method for embedding hyperlinks into ordinary images that can detect watermarks by camera-equipped mobile devices. Incorporation of a distortion network which employs distinguishable 3D rendering operations among encoder and decoder can emulate distortion caused by the introduction of camera imaging and is somewhat robust to camera photography. Fang et al. [24] suggested a depth-template-based algorithm for watermarking, which designs a message embedding template and a localization template on the embedding side and uses a two-level DNN on the extraction side to achieve higher robustness against camera photography.

To increase the robustness of screen-photography watermarking and to strike a balance between robustness and invisibility. Therefore, we propose a screen-photography watermarking technique. Our contributions are the following:

(1) In regard to embedding area selection, the intensity-based SIFT algorithm used for the construction of feature regions according to the density of feature points, so that the feature region pays more attention to the key content of the image, and this method can accurately locate the embedded region during extraction;

(2) In terms of watermark embedding strength, the JND model is proposed to set the intensity of the watermark embedding, and the watermark embedding strength changes adaptively with image luminance and texture, to achieve the balance of robustness and invisibility;

(3) An optimization scheme of neighborhood coefficient size constraint is proposed to apply the same size constraint to other coefficient pairs in the neighborhood of modified DCT coefficients, so that the watermark information can still be extracted accurately when the feature points are shifted. Results of the experiment showed that after the above three methods can improve the correct rate of watermark extraction, and the quality of the visual aspect of the watermarked picture is also improved.

The organization of this paper is as follows. Section 2 is a related presentation that describes screen photography distortion, the SIFT algorithm, and the JND model. Section 3 describes the proposed method in this paper. The experimental results and comparative analysis are given in Section 4. Finally, Section 5 concludes the work of this paper.

2 Related Work

2.1 Screen Photography Distortion

Fang et al. [17] and Chen et al. [19] studied and summarized distortions that resulted from the screen capture process. The process of screen camera is classified into three sub-processes, which are screen display, shooting and camera imaging process. And a series of distortions are caused in these three sub-processes, Fang et al. divided the distortion in the screen capture process into four categories, which are display distortion, lens distortion, sensor distortion and processing distortion, and Chen et al. divided the distortion in the screen capture process into five categories, which are linear distortion, gamma adjustment, geometric distortion, noise attack and low-pass filter attack [25]. We believe that three of these distortions, lens distortion, light source distortion and Moiré streak distortion should be of concern.

2.2 SIFT Algorithm

The SIFT algorithm or scale invariant feature transformation is an algorithm in image processing, which is used to detect key points in the image and provide feature descriptors for the direction of key points [26]. The algorithm can adapt to changes in luminance with good stability and invariance. Schaber et al. [27] demonstrated that SIFT keypoints are screenshot invariant, but traditional SIFT algorithms do not allow blind extraction and have long localization times [28,29], so we also use the same intensity-based SIFT algorithm as Fang et al. for finding regions with embedded watermarks. The equation is defined as Eq. (1).

$$In(p) = |D(p)| \tag{1}$$

where p denotes a key point in the Difference of Gaussians (DoG) domain, and $p = (x, y, \sigma)$, the DoG image is described as Eq. (2).

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{2}$$

where (x, y) is a pixel location of the image and σ is the scale space factor. $L(x, y, \sigma)$ is defined as the original image $I(x, y)$ with a variable scale 2-dimensional Gaussian $G(x, y, \sigma)$ convolution operation. The equation is as follows:

$$L(x, y, \sigma) = G(x, y, \sigma) * I_o(x, y) \tag{3}$$

where $*$ denotes the convolution operation, (x, y) represents a pixel position of the image, and σ is the scale space factor.

2.3 JND Model

JND is the minimum perceptible difference, and in image processing, JND can be used to measure the sensitivity of the human eye to distortion in different regions of an image [30]. The JND model mainly considers the properties of luminance masking as well as texture masking of Human Visual System(HVS), and uses a nonlinear relationship to superimpose the effects of both. Qin et al. [31] suggested using the spatial JND model to determine the strength of the watermark embedding to obtain good visual performance. This paper uses JND to adjust the size of watermark embedding intensity, which can achieve an adaptive change of embedding intensity with image luminance and texture. The basic formula of JND is as follows.

$$JND_\theta^s(x, y) = T^l(x, y) + T_\theta^t(x, y) - C_\theta^{lt} * \min\{T^l(x, y), T_\theta^t(x, y)\} \tag{4}$$

Of these, the $T^l(x, y)$ and $T_\theta^t(x, y)$ are the visibility threshold of the luminance masking factor and the texture masking factor. C_θ^{lt} is the superposition constant, which reflects the degree of overlap of the two masks.

3 Proposed Method

3.1 Watermark Embedding Process

The watermark embedding method proposed is described below. First select the host image for embedding, if a color image is an input, convert the input to YCbCr color space, and then select the image of Y channel, otherwise, it will be directly used as the host image. Then select the feature region for embedding the watermark, find the feature points using intensity-based SIFT algorithm and construct the feature region according to the density of the feature points. The watermark sequence is encoded using Bose–Chaudhuri–Hocquenghem(BCH) codes and the encoded watermark sequence is used to embed. The watermark embedding is performed in the DCT domain. Finally, the coefficients in the neighboring domain are adjusted again with optimization constraints. The specific watermark embedding process is shown in Fig. 1.

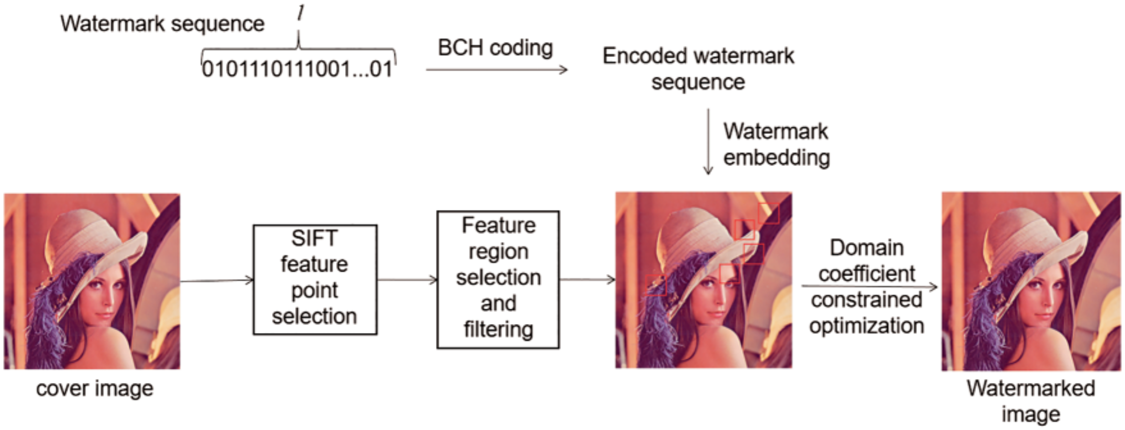

Watermark sequence
l
0101110111001...01
BCH coding
Encoded watermark sequence
Watermark embedding
cover image
SIFT feature point selection
Feature region selection and filtering
Domain coefficient constrained optimization
Watermarked image

Figure 1: Watermark embedding process

Convert the watermark sequence into a binary sequence and encode it using BCH codes. The obtained binary watermark sequence is put into a matrix W of size a ∗ b by column and W is the smallest square matrix possible. If the length of the binary watermark sequence is less than a ∗ b, then the other remaining positions in W are subjected to a complementary 0 operation to finally obtain the watermark matrix. Each bit of information is embedded in a block of size 8 × 8 pixels. With the candidate feature points as the center, the feature region is obtained based on the watermark matrix, and the size of the feature region is a ∗ b ∗ 8 ∗ 8.

The feature points are detected in the image using the intensity-based SIFT algorithm and sorted in descending order of feature point intensity, and the top n feature points with the largest feature points intensity are selected as candidate feature points. The feature region is filtered for the first time depending on the area that cannot exceed the borders of the image. Then the feature region with at least k candidate feature points is selected for the second time based on the density of feature points. Finally, the feature region is filtered based on the fact that each feature region cannot overlap with each other, and if there is an overlapping region, the one with low feature point intensity is removed. After finding all the feature points that satisfy the conditions, the top m feature points and the corresponding feature regions are the final choices in order of the intensity of the feature points. If the number of feature points is less than m, the embedding is done according to the actual feature points and feature regions. However, after extensive experiments, it is found that texture-rich images have at least five such regions. We found that the feature regions determined based on the feature point density can be more concentrated in key locations of the image, and such locations tend not to be cropped out. Taking the crown plot as an example. Fig. 2a shows the feature regions found by Fang et al.'s method, and Fig. 2b shows the feature regions found by the proposed method, from which we can see that the feature regions found by our method surround the key regions of the image.

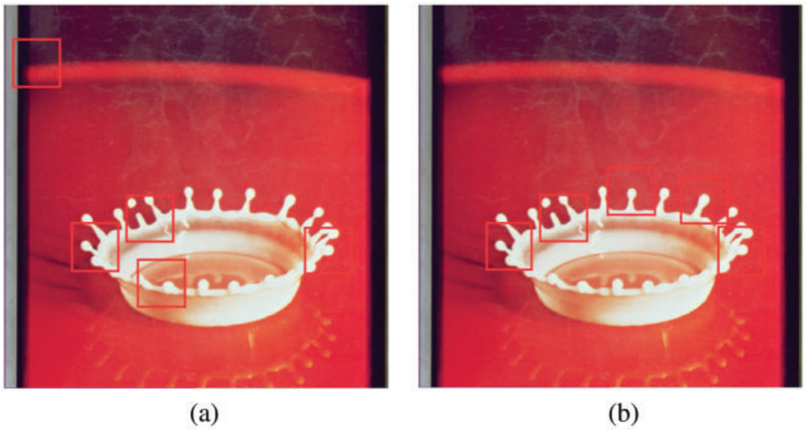


Figure 2: Selection results of feature regions

The embedding of the watermark is performed in the DCT domain. Fang et al. found that the relative magnitude of a pair of DCT coefficients at mid-frequency didn’t change after the screen-photography process, so embedding of the watermark, as well as blind extraction, can be achieved by comparing the magnitude of this pair of coefficients. The image is first transformed to DCT domain, and then embed watermark by comparing the coefficient values of two positions, the two coefficients are C1 and C2. If the information to be embedded is 0, C1 ≥ C2 is guaranteed; If the information to be embedded is 1, C1< C2 is guaranteed. The watermark strength in this paper is determined by the JND model to achieve the adaptive change of embedding strength with image luminance and texture, and the specific equation of watermark embedding is shown in Eq. (5).

$$\begin{cases} JND(x,y) = 2 * (2 * (T^{t}(x,y) + 3) + T^{l}(x,y)) \\ C1 = \max(C1, C2) + JND(x_{c1}, y_{c1}), C2 = \min(C1, C2) - JND(x_{c2}, y_{c2}) \quad \mathit{if}\ w = 0 \\ C1 = \min(C1, C2) - JND(x_{c1}, y_{c1}), C2 = \max(C1, C2) + JND(x_{c2}, y_{c2}) \quad \mathit{if}\ w = 1 \end{cases} \tag{5}$$

where (x, y) is a pixel position of the image, the $T^{l}(x, y)$ and $(T^{t}(x, y)$ are visibility threshold for the luminance masking factor and texture masking factor, the $JND(x, y)$ is JND value of a pixel location, C1 and C2 are the discrete cosine transform coefficient values of the two locations, and w is the watermark value.

Since the shooting process may lead to an offset of some feature points during extraction, Fang et al. use a neighborhood traversal method to compensate for the offset. However, when the offset occurs, the comparison may be performed with more than just the two coefficient values C1 and C2, and we take the coefficients at positions (4, 5) and (5, 4) as an example below. Fig. 3 plots the results of eight possible offsets, one coefficient pair in each ellipse, and ultimately it may be possible to determine whether the embedded information is 0 or 1 by comparing the magnitudes of the two coefficient values in these pairs. When the magnitude relationship between the values of the other coefficient pairs is exactly opposite to that of (4, 5) and (5, 4), and the number of opposite coefficient pairs is greater than the number of identical coefficient pairs, it is possible that the extracted information is the exact opposite of the embedded information, and thus the information cannot be extracted correctly.

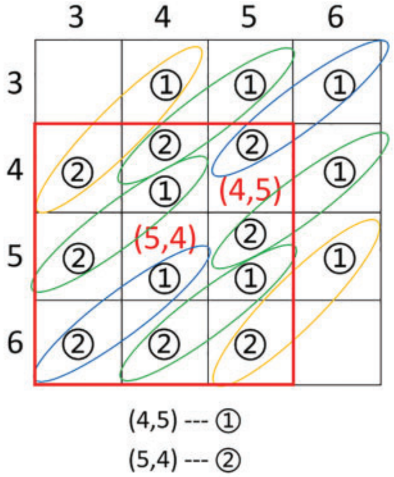


Figure 3: Possible offset results of feature points

Therefore, after embedding the watermark, we propose to apply size constraints to the discrete cosine transform coefficients in the $3 * 3$ neighborhood of (4, 5) and (5, 4) positions to make sure that watermark information is still extractable when feature points are offset. The formula we wish to end up with for the size relationship is shown in Eqs. (6) and (7).

$$\begin{cases} C(3,4) > C(4,3) \\ C(3,5) > C(4,4) > C(5,3) \\ C(3,6) > C(4,5) > C(5,4) > C(6,3) \qquad \textit{if } w = 0 \\ C(4,6) > C(5,5) > C(6,4) \\ C(5,6) > C(6,5) \end{cases} \tag{6}$$

$$\begin{cases} C(3,4) < C(4,3) \\ C(3,5) < C(4,4) < C(5,3) \\ C(3,6) < C(4,5) < C(5,4) < C(6,3) \qquad \textit{if } w = 1 \\ C(4,6) < C(5,5) < C(6,4) \\ C(5,6) < C(6,5) \end{cases} \tag{7}$$

When the above formula is not satisfied, we need to modify the adjustment for the coefficient values that do not satisfy the condition. For C(3, 4),C(4, 3) and C(5, 6),C(6, 5) the two pairs of coefficients are directly exchanged if the condition is not satisfied; for C(3, 5),C(4, 4),C(5, 3) and C(4, 6),C(5, 5),C(6, 4) the four pairs of coefficients , if they do not satisfy the condition, are sorted according to the size relationship and assigned to each coefficient in turn; for C(3, 6),C(4, 5),C(5, 4),C(6, 3) the three pairs of coefficients, if the condition is not satisfied, are modified according to Eq. (8).

$$\begin{cases} C(3,6) = C(4,5) + |C(3,6)|, C(6,3) = C(5,4) - |C(6,3)|, \textit{if } w = 0 \\ C(3,6) = C(4,5) - |C(3,6)|, C(6,3) = C(5,4) + |C(6,3)|, \textit{if } w = 1 \end{cases} \tag{8}$$

After modifying the coefficient value, perform the inverse DCT transformation to the spatial domain. Repeat the above operations sequentially until all feature regions are embedded in the watermark. Finally, add the previous Cr and Cb channels or directly get the image.

3.2 Watermark Extraction Process

The following describes the watermark extraction process, a detailed process can be seen in Fig. 4. Since mobile phone shooting will cause some image distortion, the photo taken by the mobile phone should be perspective corrected first, and the cropping and scaling operation should be performed to get the extracted image as large as the raw image. But here four vertices of the image need to be known. The SIFT algorithm based on intensity is used to select the feature points, select the first n feature points with the greatest intensity, and select the feature region as the embedding method. The feature regions are divided into a ∗ b blocks, a DCT transform is executed on each block, and the embedding information is determined as 0 or 1 by comparing the magnitude of the DCT coefficient values.

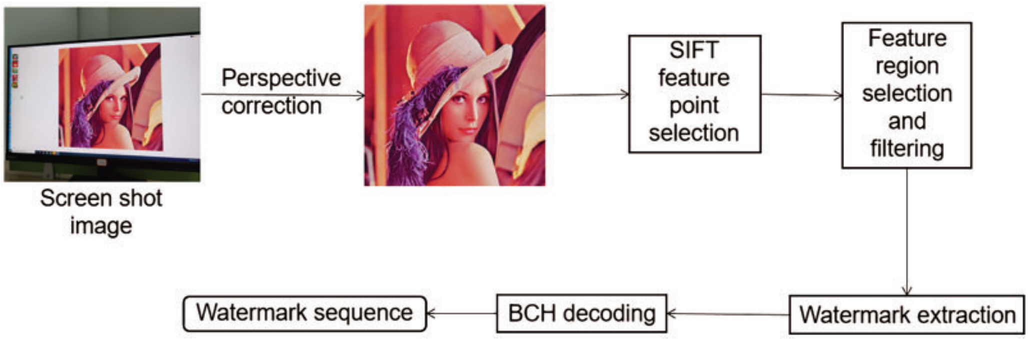


Figure 4: Watermark extraction process

Since the intensity of the feature points may change and their positions may be shifted after taking a picture, we perform offset compensation. With the feature point as the center, 9 neighboring feature points in their surrounding 3 ∗ 3 neighborhood are selected to form an extracted point group for offset compensation. The 9 watermarks extracted from this extracted point group are used as a watermark group. Watermarks in different watermark groups are compared, and if the difference between two watermarks is less than the threshold *th*, the two watermarks are recorded as a watermark pair w_f. Watermarks in the same group will not be compared. The formula for calculating the difference between two watermarks are shown in Eq. (9), and the formula for watermark pair w_f of the formula is shown in Eq. (10).

$$diff\left(w_{i\alpha}, w_{j\beta}\right) = \sum_{x=a}^{b} \sum_{j=a}^{b} \left[w_{i\alpha}(x,y) \oplus w_{j\beta}(x,y)\right] \tag{9}$$

$$w_f = \begin{cases} \{w_{i\alpha}, w_{j\beta}\}, & if\ diff\left(w_{i\alpha}, w_{j\beta}\right) \leq th \\ \emptyset, & otherwise \end{cases} \tag{10}$$

where $w_{i\alpha}$ is the αth watermark in the ith watermark group, $w_{j\beta}$ is the βth watermark in the jth watermark group, (x, y) are the coordinates of the watermark matrix, and $\oplus$ is the XOR operation.

Finally, a watermark pair w_f with a length of 2 l is obtained. Add all watermarks in w_f in sequence according to the coordinates. If the value at a certain position is greater than or equal to l, that is, half of the number of watermarks in w_f, it is considered that the watermark value at that position is 1, otherwise, it is 0. See Eq. (11) for details.

$$w(x,y) = \begin{cases} 1, & if \sum_{i=1}^{2l} w_f^i(x,y) \geq 1 \\ 0, & otherwise \end{cases} \tag{11}$$

Of these, $w_f^i(x,y)$ is the value of the ith watermark in w_f at position (x,y) and $2l$ is the number of watermarks in w_f.

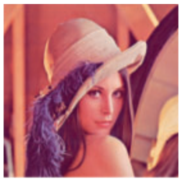
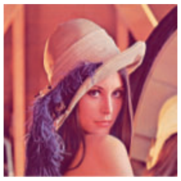
By adjusting the embedding region, the adaptive variation of embedding intensity and constraining the magnitude relationship of the coefficient in the neighborhood of the modified coefficient, we find that the probability of successful watermark extraction from the captured images increased and the quality of the watermarked images visually improved.

4 Experimental Results

Next, we will show and discuss the results of the comparison experiments with Fang et al. We carried out experiments under the MATLAB platform. The image database we used is the USC-SIPI image database [32]. Again, we choose a = 8, b = 8, and the error correction code chosen is BCH(64, 39), which corrects 4-bit errors. The n in the experiment is set to 50 and k is set to 3. The monitor we use is 'LEN T27q-20' and the phone used is 'Huawei p40 pro'. The following experiments compare the results in terms of PSNR values and different shooting processes including shooting distance, shooting angle, and handhold shooting, respectively.

Tab. 1 shows the comparison between Fang's method and the proposed method in PSNR value. From the table, it can be seen that most of the plots have some improvement in PSNR values, but there are also a small number of plots with slightly lower PSNR values.

Table 1: Images embedded by different methods and corresponding PSNR values

Imaging	Baboon	Crown	House	Lena	Peppers
Fang et al.					
PSNR (dB)	40.8648	42.5589	42.3142	42.3190	42.4843
Proposed					
PSNR (dB)	38.4291	44.3251	42.5541	43.8124	43.8554

(1) Effect of distance on robustness: Our shooting distance is 45–105 cm. Tab. 2 shows examples of the images obtained by shooting at different shooting distances and the images obtained after perspective correction. Tab. 3 lists the average watermark error bits extracted by Fang's and our method at different shooting distances. As can be seen from the table, the error bits of ours are lower than the method of Fang et al. at most distances, but they are also slightly higher or equal at some distances.

Table 2: Examples of screen shot images at different shooting distances

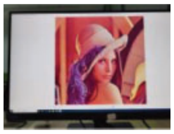
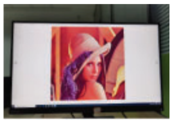
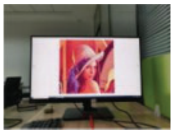
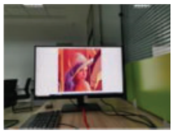
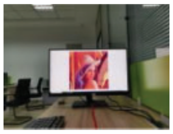
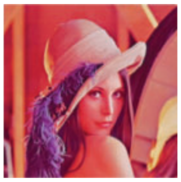
Length	45 cm	55 cm	75 cm	95 cm	105 cm
Shoot (a picture)					
Rehabilitate					

Table 3: Average error bits of watermark extracted under different shooting distances

Distance (cm)	Fang et al.	Proposed
45	1.50	0.44
55	2.15	1.50
65	1.69	0.75
75	1.50	1.50
85	0.75	1.12
95	1.37	0.69
105	0.75	0.55

(2) Effect of horizontal viewing angle on robustness: in doing experiments on the effect of horizontal viewing angle on robustness, we controlled the shooting distance to be 55 cm and the horizontal viewing angle was varied from 65° left to 65° right. Tab. 4 shows examples of the images obtained by shooting at different horizontal viewing angles and the images obtained after perspective correction. Tab. 5 shows the average extracted watermark error bits for the method of Fang et al. and the proposed method at different horizontal viewing angles. It can be seen from the table that the error bits of ours are lower than the method of Fang et al.

Table 4: Examples of screenshot images taken at different horizontal viewing angles

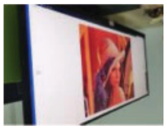
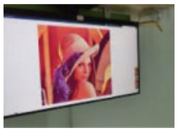
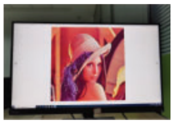
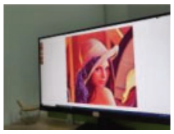
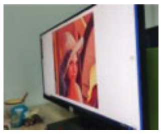
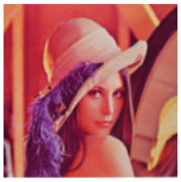
Perspectives	65° left	30° left	0°	30° right	65° right
Shoot (a picture)					
Rehabilitate					

Table 5: Average error bits of watermark extracted under different horizontal viewing angle

Horizontal angle (°)	Fang et al.	Proposed
Left 65	3.12	2.19
Left 60	1.50	0.85
Left 45	1.38	0.69
Left 30	2.34	0.75
Left 15	1.58	0.88
0	2.15	1.50
Right 15	3.00	0.94
Right 30	1.56	0.68
Right 45	0.65	0.51
Right 60	3.12	1.59
Right 65	4.50	2.94

(3) Effect of vertical viewing angle on robustness: in doing experiments on the effect of vertical viewing angle on robustness, we controlled the shooting distance to be 55 cm and the vertical viewing angle changed from 65° up to 65° down. Tab. 6 shows examples of the images obtained by shooting at different vertical viewing angles and the images obtained after perspective correction. Tab. 7 shows the average extracted watermark error bits for the method of Fang et al. and the proposed method at different vertical viewing angles. As can be seen from the table, the error bits of ours are less than or equal to the method of Fang et al.

Table 6: Examples of screenshot images taken at different vertical viewing angles

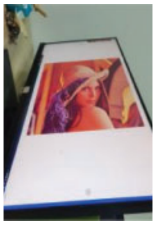
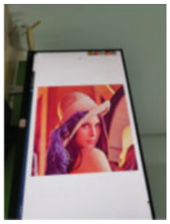
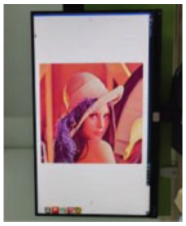
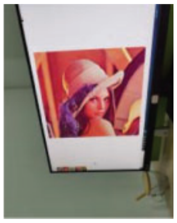
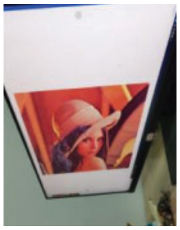
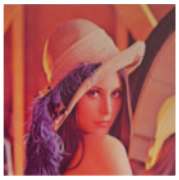
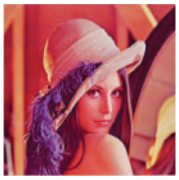
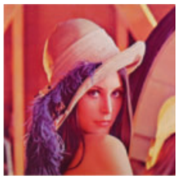
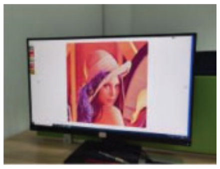
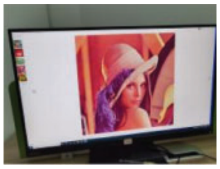
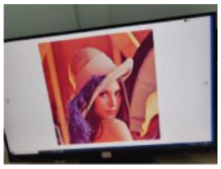
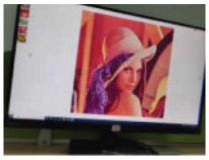
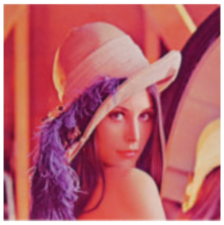
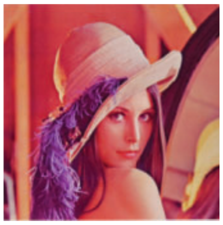
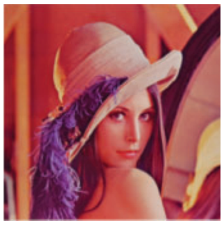
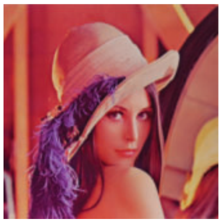
Perspectives	Down 65°	30° below	0°	Up 30°	Up to 65°
Shoot (a picture)					
Rehabilitate					

Table 7: Average error bits of watermark extracted under different vertical viewing angle

Vertical angle (°)	Fang et al.	Proposed
Up to 65	1.59	1.02
Go up to 60	1.50	1.50
Top 45	2.61	0.55
Upper 30	3.25	0.89
Up to 15	3.11	0.75
0	2.15	1.52
Next 15	1.50	1.14
Next 30	2.55	0.55
Lower 45	3.01	1.53
Lower 60	1.54	1.11
Lower 65	1.69	1.18

(4) Effect of handhold shooting on robustness: Examples of images obtained by shooting under handhold conditions are shown in Tab. 8, images obtained after perspective correction and the average error bits of the recovered images for watermark extraction. From the table, as we can see, the watermark information is correctly extracted even under handhold shooting conditions and the method is robust to handhold shooting.

Table 8: Examples of handhold shooting

Handhold shooting	1	2	3	4
Shoot (a picture)				
Rehabilitate				
False bit	0	0	1	0

The experimental comparison shows that our proposed method has some improvement in robustness and vision over the method of Fang et al. However, the scheme has some limitations like the method of Fang et al. Due to the limitations of the SIFT algorithm, if the texture of the image is too simple, the position of the selected feature points will change significantly, and thus the information cannot be extracted correctly. During extraction, it is still necessary to manually locate the four vertices of the image to be able to recover the extracted image. For some text images, the proposed embedding method leads to a very obvious visual distortion that can affect the normal reading of people.

5 Conclusion

For the purpose of protecting the image displayed on the screen, a robust watermarking solution based on image content for screen photography is proposed. In terms of embedded area selection, the intensity-based SIFT algorithm is used to construct feature regions based on the density of feature points, which can make the feature regions more focused on the key content of the image, more accurate in extraction and resistant to certain cropping attacks; the JND model is used to realize the adaptive change of embedding strength with image luminance and texture to achieve a balance between robustness and invisibility; after embedding, the coefficients in the domain are optimized again. In this way, when using statistical features to extract watermark, it can improve the number of correct coefficient size relationships, reduce the opposite situation of extraction results, and have higher accuracy. The results of the experiments indicate that the solution offers some improvements in terms of robustness and vision.

Funding Statement: This work is supported by the National Key R&D Program of China under grant 2018YFB1003205; by the National Natural Science Foundation of China under grant U1836208, U1836110; by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund; by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET) fund, China.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Y. El-mashad, A. M. Yassen, A. K. Alsammak and B. M. ElHalawany, "Local features-based watermarking for image security in social media," *Computers, Materials and Continua*, vol. 69, no. 3, pp. 3857–3870, 2021.

[2] A. Ustubioglu, G. Ulutas and M. Ulutas, "DCT based image watermarking method with dynamic gain," in *Int. Conf. on Telecommunications & Signal Processing IEEE*, Prague, Czech Republic, 2015.

[3] M. Moosazadeh and A. Andalib, "A new robust color digital image watermarking algorithm in DCT domain using genetic algorithm and coefficients exchange approach," in *2016 Second Int. Conf. on Web Research (ICWR) IEEE*, Tehran, Iran, 2016.

[4] A. Abrar, W. Abdul and S. Ghouzali, "Secure image authentication using watermarking and blockchain," *Intelligent Automation & Soft Computing*, vol. 28, no. 2, pp. 577–591, 2021.

[5] N. Jimson and K. Hemachandran, "DFT based coefficient exchange digital image watermarking," in *2018 Second Int. Conf. on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2018.

[6] T. D. Hien, Z. Nakao and Y. Chen, "Robust RDWT-ICA based information hiding," *Soft Computing*, vol. 10, no. 12, pp. 1135–1144, 2006.

[7] Y. Xie, H. H. Tan and K. L. Wang, "A novel color image hologram watermarking algorithm based on QDFT-DWT," in *2016 Chinese Control and Decision Conf. (CCDC)*, Yinchuan, China, 2016.

[8] F. Ernawan, D. Ariatmanto and A. Firdaus, "An improved image watermarking by modifying selected DWT-DCT coefficients," *IEEE Access*, vol. 9, pp. 45474–45485, 2021.

[9] A. S. Salama, M. A. Mokhtar, M. B. Tayel, E. Eldesouky and A. Ali, "A triple-channel encrypted hybrid fusion technique to improve security of medical images," *Computers, Materials and Continua*, vol. 68, no. 1, pp. 431–446, 2021.

[10] N. Bousnina, S. Ghouzali, M. Mikram, M. Lafkih, O. Nafea *et al.*, "Hybrid multimodal biometric template protection," *Intelligent Automation & Soft Computing*, vol. 27, no. 1, pp. 35–51, 2021.

[11] A. Pramila, A. Keskinarkaus and T. Seppanen, "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.

[12] C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller *et al.*, "Rotation, scale, and translation resilient watermarking for images," *IEEE Transactions on Image Processing a Publication of the IEEE Signal Processing Society*, vol. 10, no. 5, pp. 767–782, 2001.

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

[14] L. A. Delgado-Guillen, J. J. Garcia-Hernandez and C. Torres-Huitzil, "Digital watermarking of color images utilizing mobile platforms," in *IEEE Int. Midwest Symp. on Circuits & Systems IEEE*, Columbus, OH, USA, 2013.

[15] T. Nakamura, A. Katayama, M. Yamamuro and N. Sonehara, "Fast watermark detection scheme for camera-equipped cellular phone," in *Proc. of the 3rd Int. Conf. on Mobile and Ubiquitous Multimedia*, Maryland, USA, 2004.

[16] A. Pramila, A. Keskinarkaus and T. Seppaenen, "Increasing the capturing angle in print-cam robust watermarking," *Journal of Systems and Software*, vol. 135, pp. 205–215, 2018.

[17] H. Fang, W. M. 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, 2019.

[18] W. T. Chen, C. Q. Zhu, N. Ren, T. Seppanen and A. Keskinarkaus, "Screen-cam robust and blind watermarking for tile satellite images," *IEEE Access*, vol. 8, pp. 125274–125294, 2019.

[19] W. T. Chen, N. Ren, C. Q. Zhu, Q. F. Zhou and A. Keskinarkaus, "Screen-cam robust image watermarking with feature-based synchronization," *Applied Sciences*, vol. 10, no. 21, pp. 74–94, 2019.

[20] W. T. Chen, N. Ren, C. Q. Zhu, A. Keskinarkaus and Q. F. Zhou, "Joint image encryption and screen-cam robust two watermarking scheme," *Sensors*, vol. 21, no. 3, pp. 701, 2021.

[21] T. Fujihashi, T. Koike-Akino, T. Watanabe and P. V. Orlik, "DNN-Based simultaneous screen-to-camera and screen-to-eye communications," *IEEE GLOBECOM IEEE*, Waikoloa, HI, USA, pp. 1–6, 2019.

[22] X. Zhong and F. Y. Shih, "A robust image watermarking system based on deep neural networks," *arXiv*, 2019.

[23] J. Jia, Z. P. Gao, K. Chen, M. H. Hu, X. K. Min *et al.*, "RIHOOP: Robust invisible hyperlinks in offline and online photographs," *IEEE Transactions on Cybernetics*, pp. 1–13, 2020.

[24] H. Fang, D. D. Chen, Q. D. Huang, J. Zhang, Z. H. Ma *et al.*, "Deep template-based watermarking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 4, pp. 1436–1451, 2021.

[25] Z. R. Li, H. W. Tian, Y. H. Xiao, Y. Q. Tang and A. H. Wang, "An error-correcting code-based robust watermarking scheme for stereolithographic files," *Computer Systems Science and Engineering*, vol. 37, no. 2, pp. 247–263, 2021.

[26] Y. M. Li, J. T. Zhou, A. Cheng, X. M. Liu and Y. Y. Tang, "SIFT keypoint removal and injection via convex relaxation," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1722–1735, 2016.

[27] P. Schaber, S. Kopf, S. Wetzel, T. Ballast and W. Effelsberg, "Cammark: Analyzing, modeling, and simulating artifacts in camcorder copies," *Acm Transactions on Multimedia Computing Communications & Applications*, vol. 11, no. 2, pp. 1–23, 2015.

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

[29] X. J. Wang and W. Tan, "An improved geometrical attack robust digital watermarking algorithm based on SIFT," in *2014 Int. Conf. on Industrial Engineering and Information Technology*, Tianjin, China, 2014.

[30] W. B. Wan, J. Wang, J. Li, J. D. Sun, H. X. Zhang *et al.*, "Hybrid JND model-guided watermarking method for screen content images," *Multimedia Tools and Applications*, vol. 79, no. 7, pp. 4907–4930, 2020.

[31] L. W. Qin, X. L. Li and Y. Zhao, "A new JPEG image watermarking method exploiting spatial JND model," *18th IWDW*, Chengdu, China, pp. 161–170, 2019.

[32] University of Southern California. "The USC-SIPI image database, signal and image processing institute," Accessed: Jul. 2013. [Online] Available: http://sipi.usc.edu/database.