IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 21, 2020, accepted July 5, 2020, date of publication July 8, 2020, date of current version July 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3007689

Screen-Cam Robust and Blind Watermarking for Tile Satellite Images

WEITONG CHEN[1,2,3], CHANGQING ZHU[1,2,3], NA REN[1,2,3], TAPIO SEPPÄNEN[4], AND ANJA KESKINARKAUS[4]

[1]Key Laboratory of Virtual Geographic Environment, Ministry of Education, Nanjing Normal University, Nanjing 210023, China
[2]State Key Laboratory Cultivation Base of Geographical Environment Evolution, Nanjing 210023, China
[3]Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing 210023, China
[4]Center for Machine Vision and Signal Analysis, Physiological Signal Analysis Team, University of Oulu, 90570 Oulu, Finland

Corresponding author: Changqing Zhu (zcq88@263.net)

This work was supported in part by the National Natural Science Foundation of China under Grant 41971338, and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20191373.

ABSTRACT To achieve watermark synchronization from an image with perspective distortion is a challenge. This paper proposes a screen-cam robust and blind watermarking scheme for tile satellite images, which means we do not need any user interaction or additional information in watermark detection. To achieve this, at the watermark embedding side, we divide tiles into synchronization tiles and message tiles, and propose a discrete Fourier transform (DFT) based embedding method to cope with the quality degradation caused by screen-cam process. At the extraction side, based on the idea of template matching, we first use a synchronization response index to estimate an appropriate scale level and positions of embedded synchronization watermarks from the noise component, which is estimated by Wiener filter. Then, we propose verification, selection, and precise locations methods to ensure the effectiveness and accuracy of synchronization detection results. After that, we can extract the regions of message tiles. Finally, we extract watermark message based on the local max value from the DFT domain of the noise component. Experimental results demonstrate the validity of the proposed scheme against common attacks and screen-cam attack with a tripod as well as handhold shooting.

INDEX TERMS Screen-cam process, robust watermark, tile satellite images, blind detection, synchronization.

I. INTRODUCTION

Tile satellite images are multi-resolution and massive fixed-size square images widely used for their convenient browsing. With the explosive development of internet technology, tile satellite images face security issues of illegal download, dissemination, and use. The current solution is encryption [1]–[3] and access control [4]–[7] technology, which can keep the tile satellite images unreadable due to encryption or prevent them from being downloaded.

However, in recent years, with the popularity of smartphones, a new data leakage method has emerged which is using the mobile phone to directly capture the tile satellite images displayed on the screen. Obviously, traditional encryption and access control technologies cannot solve this problem. In this case, a technology that can trace leakage of the screen displayed content after it was captured by a camera has been investigated. Screen-cam robust watermarking [8] is a highly robust digital watermarking technology, which not only can resist common image processing attacks, but also has a good performance to resist screen-cam attacks. Therefore, how to design a screen-cam robust watermarking algorithm for tile satellite images is a meaningful issue.

In screen-cam process, a camera device is used to regenerate the content displayed on the screen into digital signals, which can be considered as a cross-media signal transmission process that contains digital-to-analog and analog-to-digital conversion. Most robust watermarking algorithms designed for cross-media signal transmission are mainly focused on print-scan and print-cam attacks. Print-scan and print-cam processes are using a scanner and a camera to convert a printed image into digital signals, respectively. Based on the

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmed Farouk.

embedding methods, the watermarking algorithms designed for print-scan and print-cam attacks can be divided into global watermarking methods and feature-based watermarking methods.

Global watermarking methods use the full image for watermark embedding. The global watermarking methods designed for print-scan and print-cam attacks mainly include pattern-based methods, Fourier domain-based methods, and multi-domain-based methods. Pattern-based methods encode watermark message with different designed patterns, for example, sinusoidal pattern [9], pseudorandom vector-based tiling pattern [10] and directed periodic pattern [11]–[14]. Fourier-domain based methods embed watermark messages in the magnitude spectrum of discrete Fourier transform (DFT) [15]–[18] or Fourier-Mellin transform domain [19], [20], which is the log-polar mapping (LPM) of the DFT magnitude spectrum. Multi-domain-based methods modulate the coefficients of the wavelet domain to embed watermark message combined with another domain-based method to achieve synchronization [21]–[23]. However, global watermarks are restricted to the desynchronization caused by stitching and cropping attack [8], [24].

Feature-based methods employ image features, which can be well preserved in the desynchronization attacks caused by translation and cropping, to achieve watermark synchronization. Keskinarkaus *et al.* [24] combined the pattern-based method with Harris detector-based synchronization method to achieve robustness to cope with print-scan and extra crop attacks.

In summary, the ideas of the watermarking algorithms designed for print-scan and print-cam attacks can be applied to screen-cam process due to the similarities of cross-media signal transmission. But the experiments in [8] show that the state-of-art methods of these algorithms are not applicable in screen-cam process due to the differences between screen-cam and print-scan/cam processes, for example, the quality degradation caused by moiré noise and illumination distortion.

Studies about screen-cam robust watermarking algorithms are yet few. Some researchers focused on the screen-cam communication of video watermark [25]–[27], which embed watermark message into temporal dimensions. Commonly, one frame of the video only carries a few bits. To embed multi-bits message and extract watermark message from one single screen-captured image, Fang *et al.* [8] and Chen *et al.* [28] proposed two different feature and frequency domain-based watermarking schemes, and Wang [29] improved the pattern-based method of [11] by image morphology methods to make it robust to screen-cam process. At the extraction side, these three methods all need to correct perspective distortion by manually selecting four points with additional information. Methods by [8] and [29] need to record the four corners of the original image in advance, which means the original size has to be known. Method by [28] at least needs to know the aspect ratio of the image or aspect ratio of the screen to help estimate the coordinates of selected four points.

According to the information required for watermark detection, watermarking methods can be divided into blind methods [30]–[32], which means we do not need any additional information or user interaction, and non-blind methods. Based on the summary above, blind watermark detection of a captured image with perspective distortion has not been comprehensively studied yet. All the methods against screen-cam developed thus far need additional information or user interaction to deal with perspective distortion. However, if the captured images do not contain the whole image or screen, it is not possible to select the four corners of the image or screen. Furthermore, due to the difference between tile satellite images and normal images, the tile satellite images displayed on the screen is a mosaic of tiles and there is also translation and cropping, it is difficult to estimate the original size or aspect ratio from captured image empirically for manual perspective correction. Therefore, how to achieve blind watermark detection of captured watermarked tile satellite images is a meaningful issue.

To solve this issue, a screen-cam robust and blind watermarking scheme for tile satellite images is proposed in this paper. The major contributions of this paper include: (1) We design a synchronization response index to estimate the positions of embedded synchronization watermarks from a captured image that has not been perspective corrected, so that blind detection can be achieved. (2) We propose a local max value and noise component-based extraction algorithm which can reduce the interference of the Fourier magnitude coefficients of the original image on the message extraction.

The rest of the paper is organized as follows. The distortions caused by screen-cam process are described in Section 2. The watermark embedding and watermark detection and extraction processes are presented in Section 3 and Section 4, respectively. Then the selection of parameters and experiment results are discussed in Section 5. Finally, Section 6 draws the conclusions.

II. PRELIMINARIES

A. DISTORTION ANALYSIS

Schaber *et al.* [33], Fang *et al.* [8] and Chen *et al.* [28] all analyzed the distortions caused by screen-cam attack, where various kinds of distortions are involved, including linear distortion, gamma tweaking, geometric distortion, noise attack, and low-pass filter attack. We can also divide these distortions into two categories: geometric distortion and quality degradation.

Geometric distortion includes the stitching, cropping and translation distortions caused by normal user operation, perspective distortion by shooting distance and angle, and lens distortion by the optical lens. To achieve watermark synchronization despite geometric distortion, we designed a synchronization watermark embedding method. At the extraction side, based on the idea of template matching, we can estimate the positions of embedded synchronization watermarks.

Quality degradation contains various distortions, of which there are two severe distortions. One is moiré noise, which commonly is significant at a close shooting distance. The other one is low-pass filter attack, which will cause blurring of the captured content as the shooting distance increases. To cope with the quality degradation, we employ error correction code to encode watermark message and embed it repeatedly to improve the robustness.

B. EMBEDDING OPERATION ANALYSIS

The screen-cam process will cause severe geometric desynchronization. Especially to achieve blind and automatic watermark detection, the designed message exaction method needs to be robust to slightly geometric distortion. Fortunately, due to the nature of DFT coefficients, when employing a large embedding region, the positions of the DFT coefficients only shift a little with slight perspective distortion and will not change with image translation [28]. It is a commonly used method to embed multiple bits of watermark message in the DFT domain of a large embedding region. Therefore, we choose the DFT domain as the watermark carrier.

The message embedding operations used in the DFT domain can be divided into selected coefficients-based methods [15], [34]–[38], log-polar mapping-based methods [19], [39]–[42], and further transform-based methods [43]. Log-polar mapping of the DFT coefficients rearranges the coefficients of the same frequency in straight lines, which can achieve better visualization and facilitate further detection operations. However, in essence, the log-polar mapping based method is the same as the selected coefficients-based methods. Further transform-based methods transform the DFT coefficients to another frequency domain for message embedding. Because of the severe distortion in screen-cam process, the further transform will make the vary of the coefficient values difficult to predict. The further transform-based method requires further research on how to apply it on screen-cam process. We design a message exaction method based on the statistical characteristics of the same frequency coefficients. Therefore, we embed the message into the selected coefficients of a selected frequency.

III. WATERMARK EMBEDDING

The watermark embedding process is shown in Fig.1. The tile satellite images are divided into synchronization tiles and message tiles. Tile satellite images are named and stored according to the indexing rules, where the index is commonly composed of the level, row, and column number, to achieve quick search. According to this, when loading the tile data, their indexes are used to determine what information the tiles should be embedded. A complete watermark message is embedded in 4 tiles, including one synchronization tile and three message tiles, as shown in Fig.2. In this way, any 2 × 2 tiles can form a complete watermark message. The embedding methods of synchronization watermark and watermark message are as follows.

A. EMBEDDING SYNCHRONIZATION WATERMARK

The synchronization tiles are used for blind detection of watermark and locating the watermark message. The synchronization watermarks are embedded in the DFT domain of the luminance spectrum of the tiles. We select the DFT coefficients at the radii ranging from 60 to 100 in increments of 5 and the angles ranging from 125 to 145 in increments of 5 in the second quadrant, which is totally 45 selected coefficients. The embedding method is defined as:

$$M_s(x, y) = k_1, \tag{1}$$

where $M_s(x, y)$ defines the watermarked magnitudes of synchronization tiles and k_1 is the embedding strength.

To reduce algorithm complexity, on the one hand, we directly modify the selected coefficients to a fixed value k_1. On the other hand, the coordinates of these coefficients are known and can be recorded in advance. Therefore, we can locate the coefficients quickly. Because of the rotational symmetry of the Fourier coefficients, the corresponding coefficients are also modified in the same way. An embedding result of synchronization watermark is shown in Fig.3 (a).

B. EMBEDDING WATERMARK MESSAGE

In our method, a whole watermark message is embedded in three tiles. We encode the watermark message with BCH code and embed it twice into the three tiles. The length of BCH code is $2^n - 1$. Considering the encoding length and

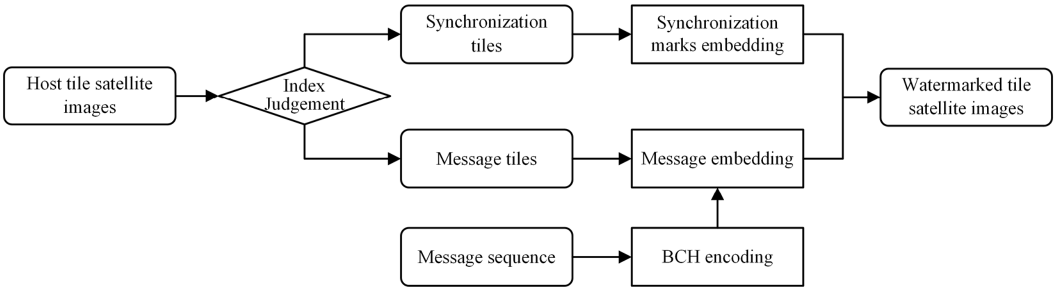


FIGURE 1. Framework of watermark embedding process.

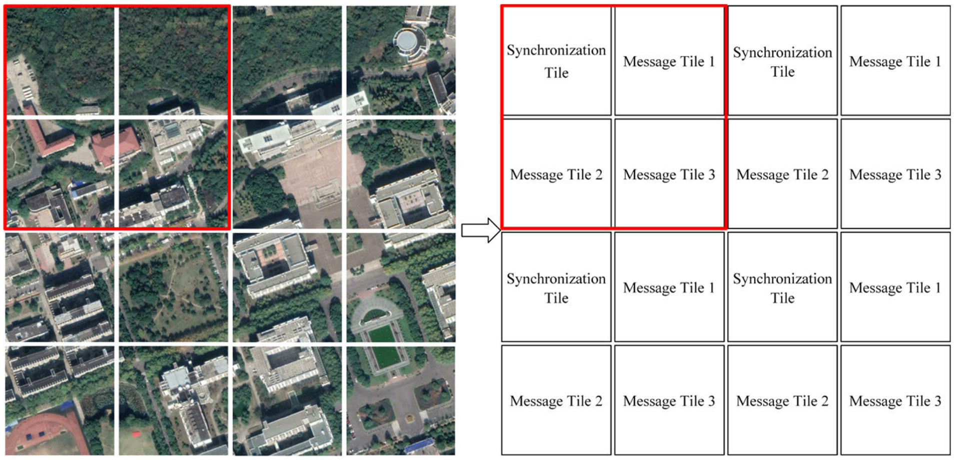
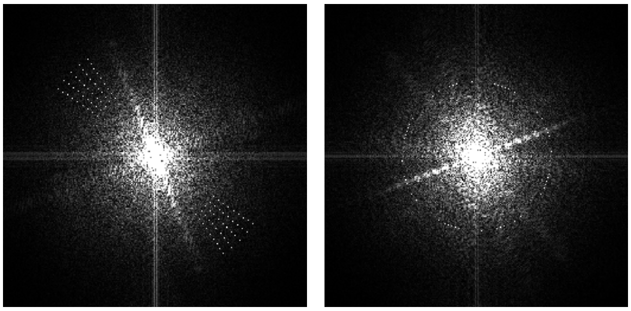


FIGURE 2. **Watermark embedding rules.**



(a) Synchronization watermark

(b) Watermark message

FIGURE 3. **Embedded watermark in the magnitude coefficients of DFT domain.**

watermark capacity, we divide the watermark message into two sub-messages. Then, we encode these two sub-messages into a 31-bit sequence and a 63-bit sequence respectively. For one complete watermark, the 31-bit sequence is embedded twice in Message Tile 1 and the 63-bit sequence is embedded once in each of Message Tile 2 and Message Tile 3.

The message sequence $W_k = \{w(j) \mid w(j) \in \{0, 1\}, j = 0, \ldots, l-1\}$ corresponding to each tile is embedded into the DFT magnitude coefficients of the luminance spectrum of the tile satellite images, where $k \in (1, 2, 3)$ defines the k-th of the three tiles and l is the length of each W_k. The message bits are evenly distributed at a distance of R from the center. The coordinates $W_k(x_i, y_i)$ of the message can be written by:

$$x_i = \frac{L_0}{2} + 1 + \text{floor}\left[R \cdot \cos\left(\frac{j}{l} \cdot \pi - \frac{1}{180} \cdot \pi\right)\right]$$
$$y_i = \frac{L_0}{2} + 1 + \text{floor}\left[R \cdot \sin\left(\frac{j}{l} \cdot \pi - \frac{1}{180} \cdot \pi\right)\right], \quad (2)$$

where L_0 is the side length of the tile, R is the embedding radius and j is the j-th element of W_k. Minus 1 degree is to avoid the frequently occurring high magnitude values at the lines $x = L_0/2 + 1$ and $y = L_0/2 + 1$ (see Fig.3).

According to [28], the low and medium-frequency magnitude coefficients with high values of DFT domain can be well preserved in screen-cam process. Therefore, we embed watermark by modulating the medium-frequency magnitudes of DFT domain to higher values. Furthermore, we can record the coordinates of the watermark embedding position in advance. When embedding watermark messages, we only need to determine the message bits are 1 or 0 and modify the corresponding magnitude coefficients to a fixed value. The embedding method is defined as:

$$M_k(x, y) = \begin{cases} k_2, & w(j) = 1 \\ \text{nochange}, & w(j) = 0, \end{cases} \quad (3)$$

where $M_k(x, y)$ defines the watermarked magnitudes of k-th tile and k_2 is the embedding strength. An embedded result of watermark message is shown in Fig.3 (b).

IV. WATERMARK DETECTION AND EXTRACTION

In this section, a blind and automatic watermark detection and extraction scheme is proposed. Fig.4 shows the framework of this scheme. In the captured photo, the original images have been resampled. Therefore we first perform rough detection at multiple scales to extract synchronization response points and determine the most appropriate scale for watermark detection and extraction. Second, the extracted synchronization points are verified whether they are from synchronization tiles by our designed method. Next, point sets are constructed, which are used for perspective correction and for locating the watermark message, of the verified points. Then, we find the precise locations of the synchronization response points of one point set in turn. After that, based on the corrected points, we perform the perspective correction and extract the message area. Finally, the message is extracted and decoded. Details are as follows.

A. ROUGH DETECTION AT MULTIPLE SCALES

This section presents a response index-based method for roughly detecting the centers of the watermark

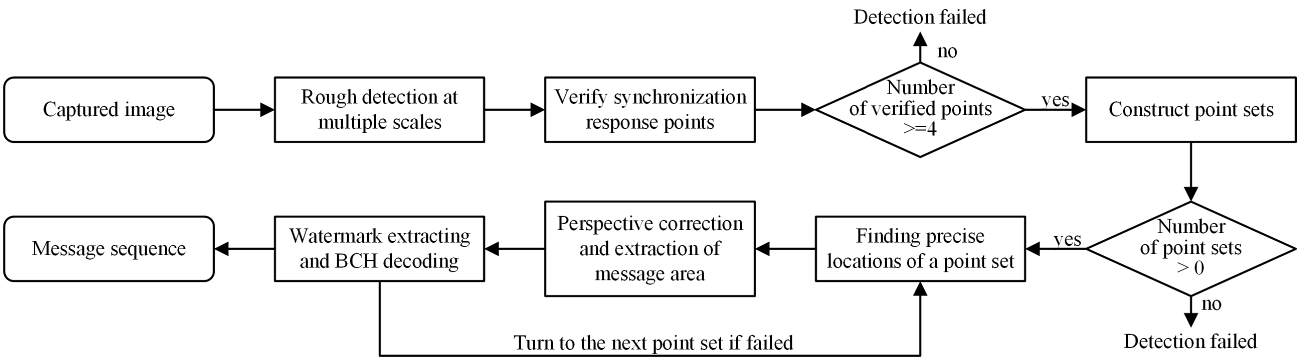

Captured image
Rough detection at multiple scales
Verify synchronization response points
Number of verified points >=4
no
Detection failed
yes
Construct point sets
Number of point sets > 0
no
Detection failed
yes
Finding precise locations of a point set
Perspective correction and extraction of message area
Watermark extracting and BCH decoding
Message sequence
Turn to the next point set if failed

FIGURE 4. Framework of watermark detection and extraction process.

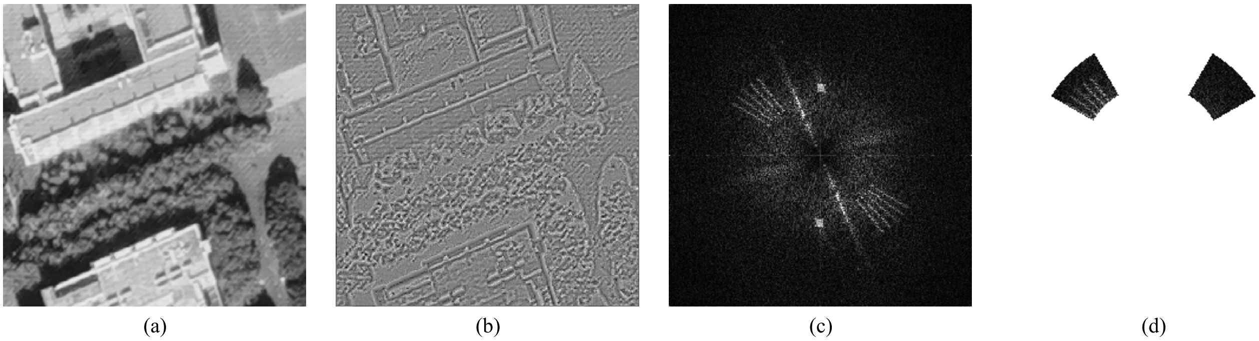


FIGURE 5. Rough detection of a window: (a) Luminance spectrum (b) Noise component (c) DFT magnitudes (d) Detection regions.

synchronization tiles at multiple scales and selecting the most suitable scale for subsequent watermark detection and extraction.

In the captured image, the original tile satellite images are scaled to varying degrees, which are determined by the shooting distance and angle. Therefore, for better detection, we need to find a suitable scale. It is not necessary to choose the scale closest to the original size, but the scale with the most significant detection results.

First, for each captured image, its luminance spectrum is scaled to five scales: 100%, 90%, 80%, 70%, and 60%. Watermark can be considered as a form of noise [44]. Therefore, we use the noise component, which is estimated by Wiener filter for detection. The noise component I_n is defined as:

$$I_n = I_c - h_w \cdot I_c, \tag{4}$$

where I_c is the luminance spectrum of the captured image and h_w defines a 3×3 spatial domain Wiener filter. Furthermore, the utilization of the I_n can also reduce the negative impact of the image itself on the detection results.

Then, at each scale, a window of 256×256 pixels is used to exhaustively detect I_n at intervals of 64 pixels. Because the synchronization watermark is embedded in a specific range of the DFT domain, detection is performed within the set range in the DFT domain of the noise component in each window. An example is shown in Fig.5. Fig.5 (a) is the corresponding luminance spectrum of the window content and Fig.5 (b) is the noise component. At each detection, the noise component is transferred into the DFT domain (Fig.5 (c)). Because of the perspective distortion, the position of the DFT magnitude coefficients will change. Therefore, considering the defined embedding range of the synchronization watermark and the perspective distortion, the detection range is set in two areas: a radius between 60 to 100, an angle between 30-60° and 120-150° respectively, as shown in Fig.5 (d).

According to the nature of the Fourier coefficients, the larger the area of the synchronization tiles contained in the window, the more obvious the noise, which means that the corresponding magnitude values are larger. Hence, a total of 45 maximum magnitude values in the two set regions are recorded as $V_l(i)$ and $V_r(i)$ according to whether their coordinates are in the second or first quadrant. The number of points in $V_l(i)$ and $V_r(i)$ are recorded as $N_{l,1}$ and $N_{r,1}$ respectively. The response index of a window can be written as:

$$R_{s,i} = \begin{cases} \text{sum}(V_l(i)) \cdot (N_{l,1} - N_{r,1}), & N_{l,1} > N_{r,1} \\ \text{sum}(V_r(i)) \cdot (N_{l,1} - N_{r,1}), & N_{l,1} < N_{r,1}, \end{cases} \tag{5}$$

where $R_{s,i}$ defines the response index of i-th window at scale s. Multiplying the number difference between $N_{l,1}$ and $N_{r,1}$ is to widen the difference between the cases with and without synchronization watermarks.

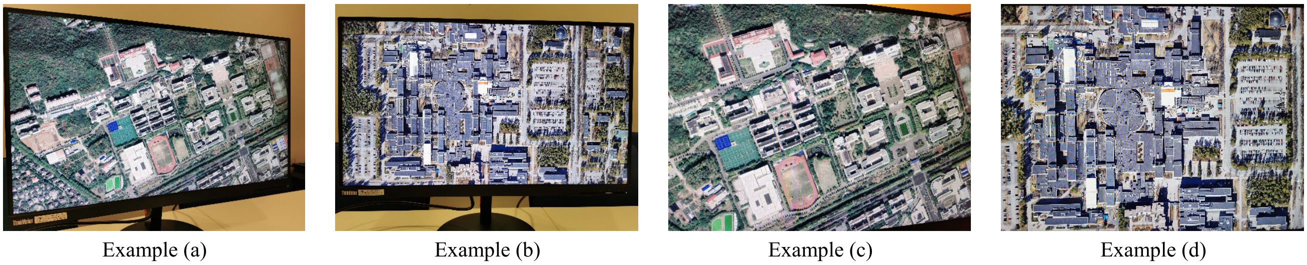

Example (a) Example (b) Example (c) Example (d)

FIGURE 6. Examples of captured images.

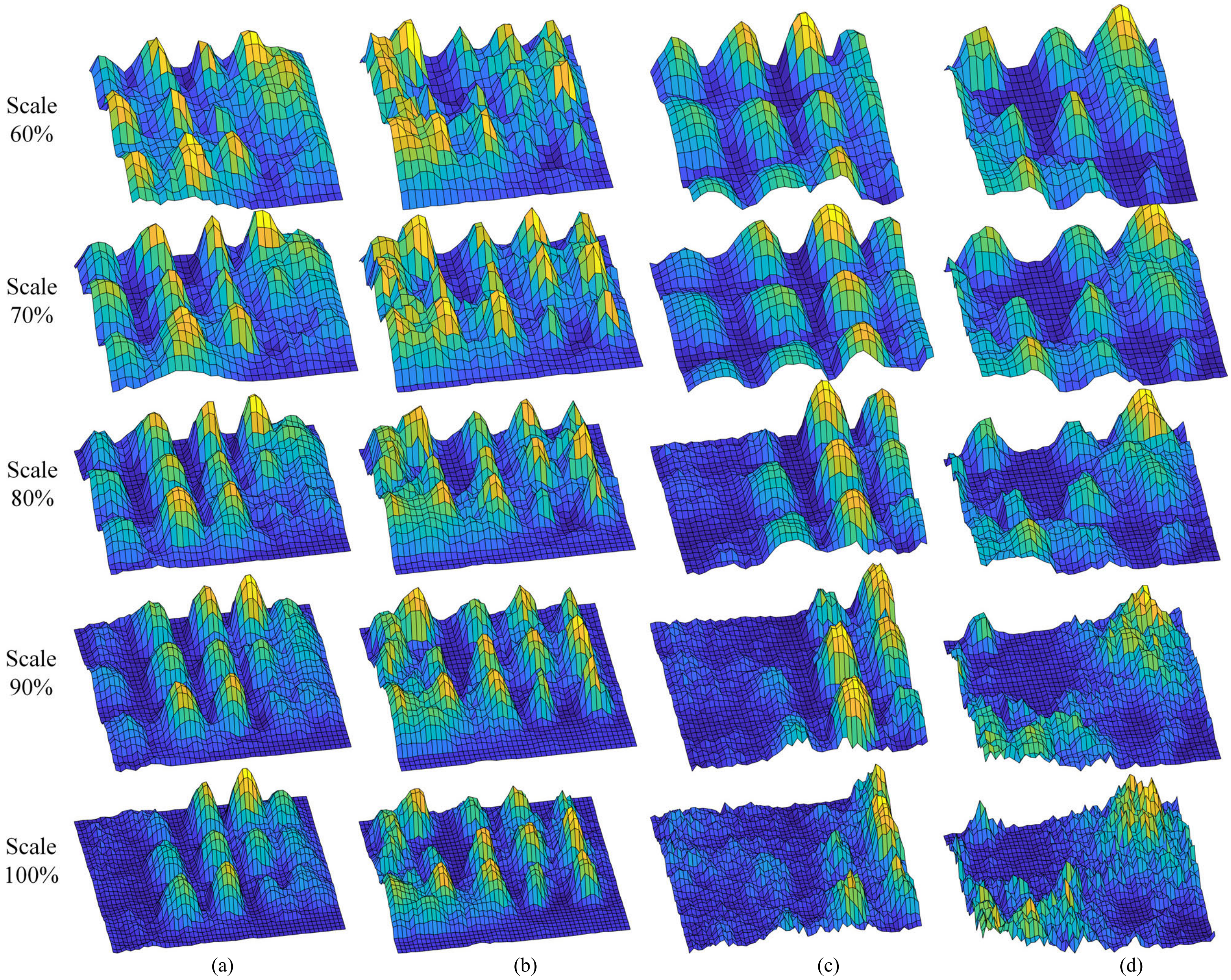


FIGURE 7. Rough detection response indexes at multiple scales.

We take four captured images as examples, which are shown in Fig.6. The $R_{s,i}$ of the four captured images on multiple scales are shown in Fig.7.

As shown in Fig.7 (a) and (c), when shooting with a big angle, the original tiles are scaled differently. Therefore, the most significant part of the response indexes varies between different scales. As shown in Fig.7 (b), when shooting relatively vertically, the response indexes commonly has a good performance at several scales. As for example (c) and (d), we take the photo at a close distance, so the synchronization watermarks can be detected at a larger scale of zoom out. In addition, when shooting a photo, the smartphone is commonly held horizontally or vertically in the hands. The orientations of the screens in the captured images are

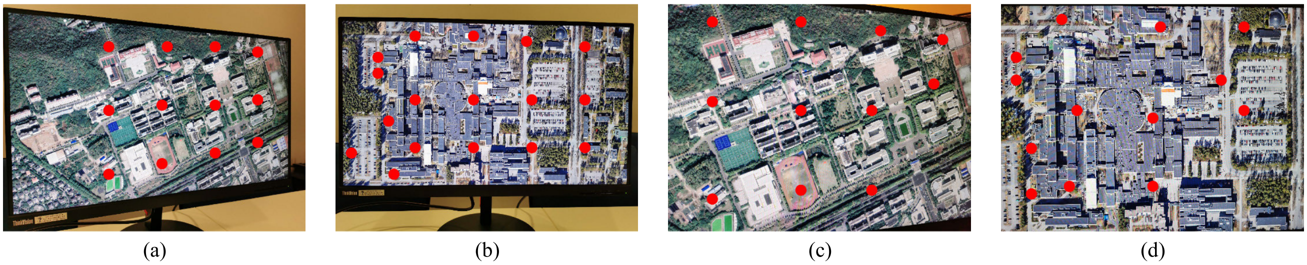


FIGURE 8. Rough detection results.

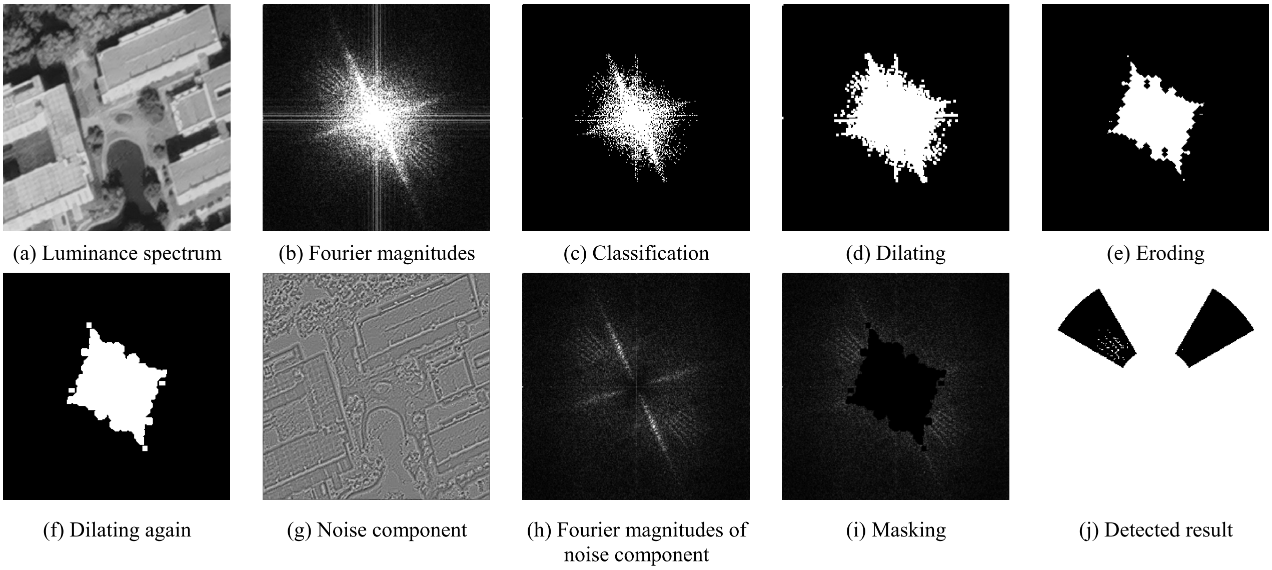

(a) Luminance spectrum (b) Fourier magnitudes (c) Classification (d) Dilating (e) Eroding (f) Dilating again (g) Noise component (h) Fourier magnitudes of noise component (i) Masking (j) Detected result

FIGURE 9. Verification process.

different. Therefore, when viewing these captured images, they may under a rotation by 90°. The positive and negative response indexes can be used to judge whether the image has undergone such a rotation.

Next, the most significant scale will be selected. If there are no synchronization watermarks in the captured image, the $R_{s,i}$ is only related to the image itself, which means the mean value of $R_{s,i}$ should theoretically approach zero. If there are synchronization watermarks in the captured image, the more significant the synchronization watermark detection, the higher the dispersion of $R_{s,i}$. Therefore, the response index of each scale R_s is defined as:

$$R_s = \sigma_s * s^2, \tag{6}$$

where σ_s is the standard deviation of $R_{s,i}$. The scale s with the maximum R_s is selected for subsequent watermark detection and extraction. The selected scales of the four examples are 100%, 100%, 60%, and 70%, respectively.

Finally, the surface peaks of $R_{s,i}$ which have a value greater than the threshold T_s are selected as the synchronization response points $P_{s,i}$. T_s is defined as:

$$T_s = \overline{R}_{s,i} + \sigma_s. \tag{7}$$

The rough detection results of the four examples are shown in Fig.8, the red dots indicating $P_{s,i}$.

B. VERIFICATION OF THE SYNCHRONIZATION POINTS

In this section, every point of $P_{s,i}$ is verified regarding whether they are close to the centers of the synchronization tiles. The main idea of the designed verification method is to judge the synchronization watermark information again while removing the impact of the image itself as much as possible. An example of the verification process is shown in Fig.9 and the details are as follows.

step.1. The luminance spectrum (Fig.9 (a)) of 256 × 256 area centered on the point from $P_{s,i}$ is transferred into the Fourier domain (Fig.9 (b)).

step.2. The Fourier magnitudes will be classified into two categories (Fig.9 (c)), which are the main components of the image and the rest. To achieve this, first, we delete the largest floor($\pi * (L_0/15)^2$) number of magnitude values and classify them directly into the main component class. Then, we classify the rest magnitudes based on Otsu's method [45], which is done by maximizing inter-class variance. If these high magnitude values are not deleted, we need to classify the

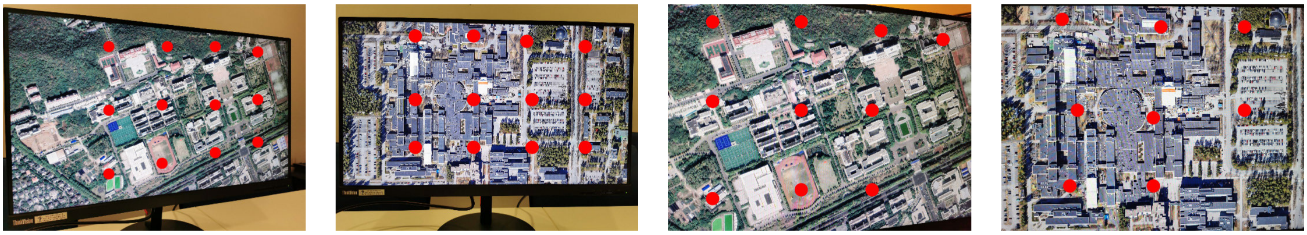


FIGURE 10. **Verified results.**

magnitudes into more classes to meet the needs, which will greatly increase the amount of calculations.

step.3. The mask is generated based on the main component class in this step. First, dilating the main component class with structuring element se_1 (Fig.9 (d)). Then, eroding twice with structuring element se_2 (Fig.9 (e)). se_1 and se_2 are defined as:

$$se_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}; \quad se_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{8}$$

Commonly, watermarks are independent of the images. Using se_2, we can better preserve the main component magnitudes when removing the watermark magnitudes from the class. Finally, after dilating twice with a structuring element se_1, a mask is generated (Fig.9 (f)), which covers the area of the main components of the image itself and makes the synchronization watermark more separable.

step.4. First, transfer the noise component (Fig.9 (g)) of the luminance spectrum to the Fourier domain (Fig.9 (h)). Then, mask the Fourier magnitude spectrum with the generated mask (Fig.9 (i)).

step.5. First, detect a total of 80 maximum magnitude values in the two regions with a radius between $L_0/6$ to $L_0/2$ and an angle between 30-60° and 120-150° respectively (Fig.9 (j)). Then, record the number of these maximum values in each of these two regions as $N_{l,2}$ and $N_{r,2}$, respectively. If $N_{l,2} - N_{r,2} > T_1$, we confirm this detected point is close to the center of a synchronization tile. Otherwise, we treat the point as a false detection and delete it.

step.6. Then, turn to the next point from $P_{s,i}$ and repeat step1-step5 until all the points are verified. Finally, we obtain the verified synchronization response points $P_{v,i}$. If $P_{v,i}$ is empty, then the watermark detection is failed.

After a perspective distortion, the positions of the watermarked magnitudes are varied. Commonly, they will not just shift to integer coordinates, which means it will cause variations in the adjacent magnitude values. Therefore, if the watermark is clear, it will cause more than 45 high-value magnitudes in the left detected region. Besides, we also need to consider the missing values due to masking error or unclear synchronization watermark. Therefore, the T1 is set to 40.

The verified results $P_{v,i}$ of the four examples are shown in Fig.10.

C. CONSTRUCTION OF THE SYNCHRONIZATION POINT SETS

In this section, all points that can be used for perspective correction in $P_{v,i}$ are selected and constructed into point sets. Then, the point sets based on their response indexes $R_{s,i}$ are sorted.

The perspective correction process can be written as [28]:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \text{where } H = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \tag{9}$$

where $[x', y', 1]^T$ and $[x, y, 1]^T$ define the homogeneous point coordinates of the corrected image and the captured image, respectively. H is a nonsingular 3 × 3 homogeneous matrix. It means we need at least four pairs of points to calculate H.

Because the side length L_0 is known, the distance between the centers of adjacent synchronization tiles is known. Therefore, we can use four synchronization response points which can represent the centers of four synchronization tiles that can form a square area for perspective correction, as shown in Fig.11. The construction method of all candidate point sets $S_j = \{P_1, P_2, P_3, P_4\}$ that can be used for perspective correction is as follows:

step.1. Select a point from $P_{v,i}$ in turn according to its $R_{s,i}$ from high to low. Assuming that this point is the point P_1 which is the top left corner of the four synchronization points used for perspective correction.

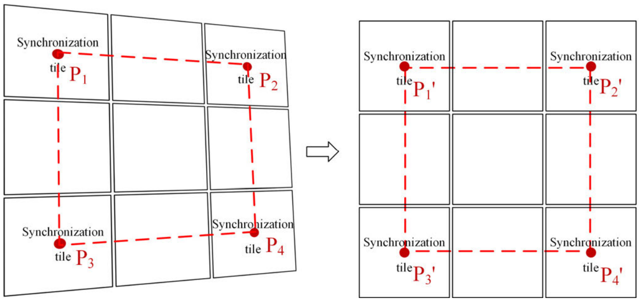


FIGURE 11. **Perspective correction with four points.**

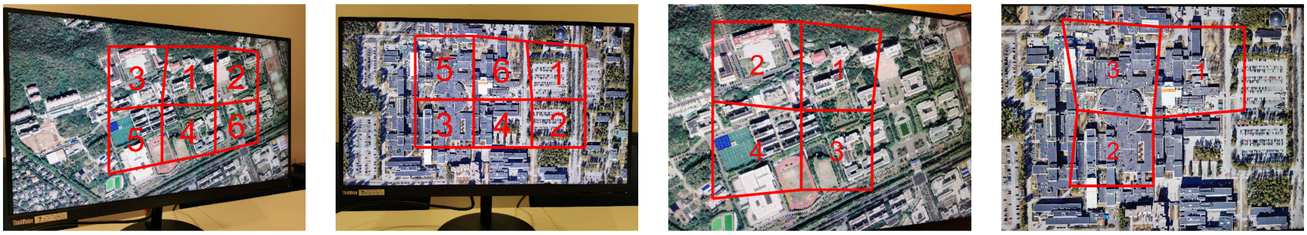
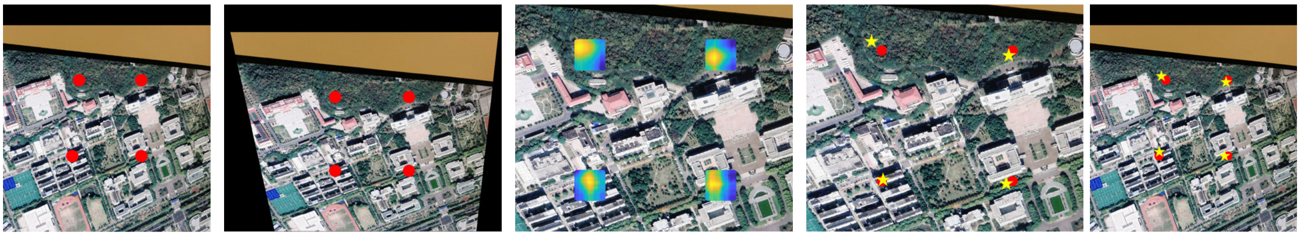


FIGURE 12. Quadrilaterals formed by constructed point sets.



(a) Local region (b) Perspective transfrom (c) Response index detection (d) Corrected result (e) Inverse perspective transform

FIGURE 13. Process of finding precise locations.

step.2. Calculate the magnitudes and directions of $\overrightarrow{P_1P_n(k)}$, where P_1 is the selected point and the $P_n(k)$ defines all the rest points from $P_{v,i}$. Find the $P_n(k)$ which has the minimum $\left\|\overrightarrow{P_1P_n\,(k)}\right\|$ and the angle between $\overrightarrow{P_1P_n(k)}$ and positive direction of the x-axis is less than 35 degrees as the top right point P_2. Also, find the $P_n(k)$ which has the minimum $\left\|\overrightarrow{P_1P_n\,(k)}\right\|$ and the angle between $\overrightarrow{P_1P_n(k)}$ and negative direction of the y-axis is less than 35 degrees as the bottom left point P_3. If there is no P_2 or P_3, go back to step 1 and continue.

step.3. Find the $P_n(k)$ for which the direction of $\overrightarrow{P_1P_n(k)}$ is between $\overrightarrow{P_1P_2}$ and $\overrightarrow{P_1P_3}$. Select the minimum $\left\|\overrightarrow{P_1P_n\,(k)}\right\|$ which is greater than $\min(\left\|\overrightarrow{P_1P_2}\right\|, \left\|\overrightarrow{P_1P_3}\right\|)$ and less than $(\left\|\overrightarrow{P_1P_2}\right\| + \left\|\overrightarrow{P_1P_3}\right\|)$. Record this point as the bottom right point P_4. If there is no P_4, back to step 1.

step.4. Record the point set $S_j = \{P_1, P_2, P_3, P_4\}$ and go back to step 1 until all points from $P_{v,i}$ are checked.

step.5. Sort S_j by maximum $R_{s,i}$ of the four points. If there are several S_j where the points with the maximum $R_{s,i}$ are the same points, then, sort these S_j by the second largest $R_{s,i}$. Otherwise, sort by the third largest $R_{s,i}$. If there is no S_j, then, the watermark detection is failed.

To some degree, the $R_{s,i}$ represents the significance of the watermark. Therefore, according to this construction method, the order of S_j theoretically represents the degree of significance of the watermark in the region of the point set. The quadrilaterals formed by S_j of the four examples are shown in Fig.12. The number shown is the order of the point set.

D. FINDING PRECISE LOCATIONS OF A POINT SET

S_j represents the four centers of four synchronization tiles. But the positions are not accurate yet. This section describes the method of finding the precise locations for S_j. The process of finding precise locations for the first S_j of example (a) is shown in Fig.13. In the experiment, we use I_n directly for calculation. Here, in Fig.13, we use the original image instead for a better demonstration.

First, in order to reduce the amount of calculations, we extract the required area (Fig.13 (a)).

Then, we perform perspective transformation on this area based on the S_j (Fig.13 (b)), which is to transfer this area to a size close to the original scale.

Next, a response index detection is performed in the 21×21 coordinates centered at each point of S_j with 5 pixels interval. An example of the detection at one coordinate is shown in Fig.14. We use the 256×256 block of the noise component centered at the coordinate (Fig.14 (a)) and transfer it into the Fourier domain (Fig.14 (b)). And then record the 45 maximum magnitude values (Fig.14 (c)) in the region with a radius between 58 and 100, and an angle between 123° and 147° in the second quadrant as $V_f\,(i)$. The precise location response index $R_{f,i}$ of each coordinate can be written as:

$$R_{f,i} = \text{sum}(V_f\,(i)). \tag{10}$$

The $R_{f,i}$ of the four points are shown in Fig.13(c) and the details are shown in Fig.15.

The number of recorded maximum magnitude values is critical for the correction result. As we explain in Section 4.2, if the synchronization watermark is clear, commonly the watermark will cause more than 45 high-value magnitudes. However, we need to consider the situation that the watermark information is not very clear, and to avoid the impact of the

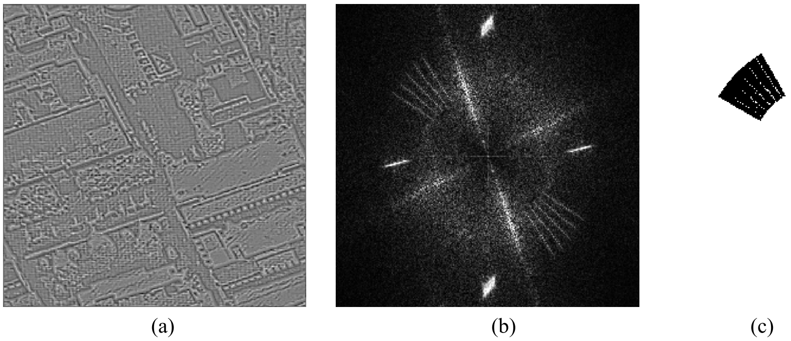


FIGURE 14. Detection process of one coordinate.

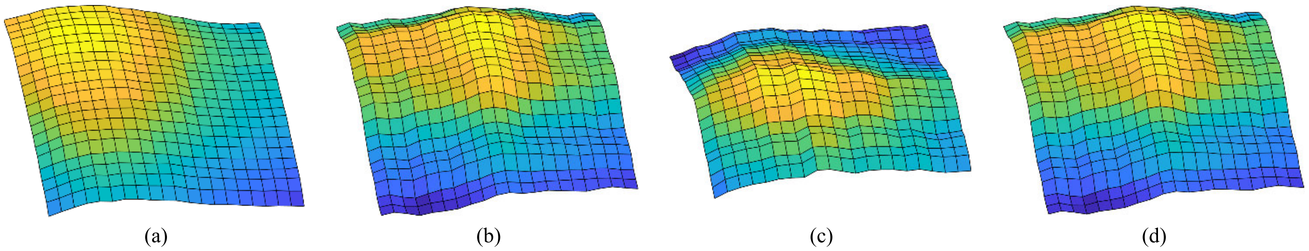
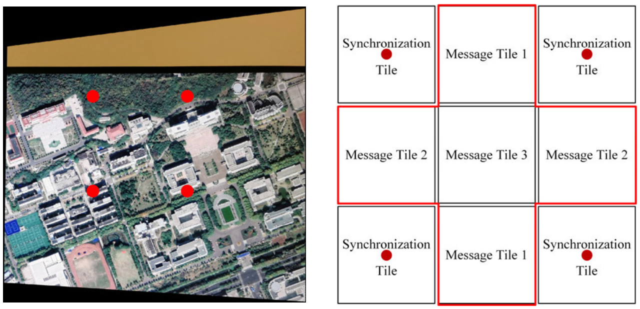


FIGURE 15. Precise location response indexes of upper left (a), upper right (b), bottom left (c) and bottom right (d) points of a point set.

image itself on the detection result. Therefore, the number cannot be set too large, we hence set it as 45.

The setting of detection region of the maximum magnitudes is also important. Although this area is perspective transformed, the transformation is not accurate. Therefore, the detection region is set a little larger than the embedding region.

Finally, as shown in Fig.13 (d), the positions of yellow pentagrams are coordinates with the maximum $R_{f,i}$. Then, we use these coordinates as the new coordinates of the corresponding points and inverse perspective transform these coordinates, as shown in Fig.13 (e). Finally, we record the coordinates of these points on the original image as the corrected point set S_j'.



(a) Perspective correction result by a point set

(b) Extraction method

FIGURE 16. Perspective correction and extraction.

E. PERSPECTIVE CORRECTION AND EXTRACTION OF MESSAGE AREA

In this section, we use S_j' to do the perspective correction and extract the message area. The perspective correction will be performed on the extracted area from the original captured image, such as the image of Fig.13 (a). The perspective correction result of Fig.13 (a) is shown in Fig.16 (a), where the original image is greatly scaled in the horizontal direction.

The four points in S_j' represent four centers of four synchronization tiles respectively. The extracted message area will be the five message tiles inside the red line, as shown in Fig.16 (b).

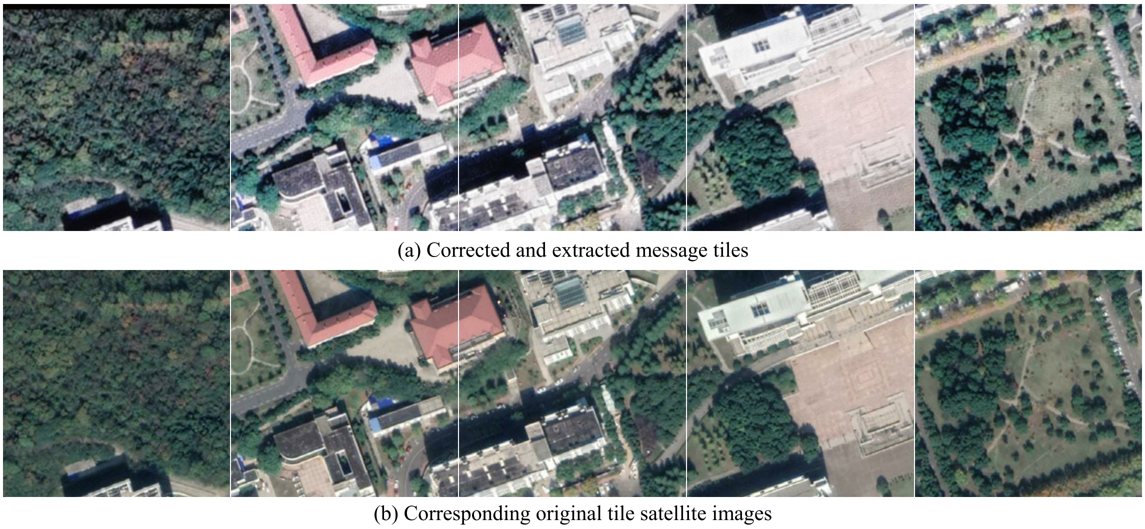
The comparison of the extracted five message tiles from Fig.16 (a) and the original tiles are shown in Fig.17, where the order is from top to bottom and from left to right. There is a black strip area at the top of the extracted tile on the far left in Fig.17 (a), which is the border of the monitor. Fortunately, this small error will not affect watermark detection.

F. MESSAGE EXTRACTION

This section presents the message extraction method and the process is shown in Fig.18. According to the embedding rules in Section 3.1, the extracted five message tiles represent two Message Tile 1, two Message Tile 2 and one Message Tile 3. Therefore, there are four embedded 31-bit message sequences and three embedded 63-bit message sequences in the extracted five message tiles.

For each corrected and extracted message tile, we use the noise component of the luminance spectrum and transfer it into the Fourier domain.

Because the perspective correction is not perfect, there is a slight shift of the original magnitude coefficients which will



(a) Corrected and extracted message tiles

(b) Corresponding original tile satellite images

FIGURE 17. **The extracted tiles and their original tiles.**

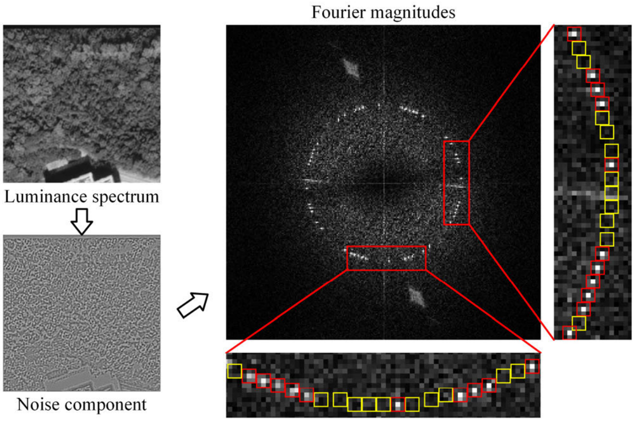


FIGURE 18. **Message extraction process.**

cause variations in the adjacent magnitude values. Therefore, we use the maximum value $V_m(j)$ within local 3×3 region centered at the embedded coordinates to determine the message bit $w'(j)$.

$$w'(j) = \begin{cases} 1, & \mathit{if}\ V_m(j) > T_w \\ 0, & \mathit{otherwise}, \end{cases} \tag{11}$$

$$T_w = \overline{M}_w + k_3\sigma_w, \tag{12}$$

where T_w is the set threshold. $\overline{M}_w$ and σ_w are the mean value and the standard deviation of all the magnitudes in the range of $\{R-2, R+2\}$, k_3 is a fixed value.

As shown in Fig.18, the red boxes and yellow boxes are the 3×3 areas of the positions where the embedded watermark bit is '1' and '0', respectively.

Based on the embedding rules, first, form $w'(j)$ into message sequences as $\overrightarrow{M}_{31}(i)$ and $\overrightarrow{M}_{63}(i)$ respectively. Then, decode $\overrightarrow{M}_{31}(i)$ and $\overrightarrow{M}_{63}(i)$ by BCH code and record them as $\overrightarrow{W}_{31}(i)$ and $\overrightarrow{W}_{63}(i)$ respectively. If two of $\overrightarrow{W}_{31}(i)$ are the same, the same decoded message sequence will be used as the detection and extraction result of the 31-bit message sequence and end the detection of this sequence. $\overrightarrow{W}_{63}(i)$ is processed similarly. When the detection results of these two messages are both obtained, which means there are two of $\overrightarrow{W}_{31}(i)$ and two of $\overrightarrow{W}_{63}(i)$ which are the same respectively, the watermark detection is successful. Otherwise, turn to the next S_j until watermark detection is successful or until all the S_j are detected.

V. EXPERIMENTAL RESULTS AND ANALYSIS

In order to ensure the success rate of watermark detection, we choose BCH(31,11) and BCH(63,24) to encode the watermark message. The 31-bit message sequence can correct 5 bits and the 63-bit message sequence can correct 7 bits. Therefore, the watermark message we can encode is 35 bits and the encoded message sequence is 94 bits.

Tile satellite images are used in real-time, so we test the embedding efficiency of the proposed embedding method. For 100 tile satellite images, it takes 2.142s to complete watermark embedding and storage with an i7-9700 CPU. Therefore, it can meet the real-time use. It is noteworthy the watermark message depends on the application, so the message bit allocation could also be different. In this paper, the embedded watermark message is set as a combination of a user ID and the current time. User ID is a 15-bit binary sequence, which means it can support 32768 IDs. And the time we embed is '01/12/2019 22:00:00'. The date is encoded using the difference in days between the current date and '01/01/2019' and encodes it as a 15-bit binary sequence, which means it can support until '19/09/2108'. The hour is directly converted to a 5-bit binary sequence.

The schematic diagram of eight host tile satellite image sets is shown in Fig.19, which includes different types of surface features, such as cities, rivers, coastal areas,

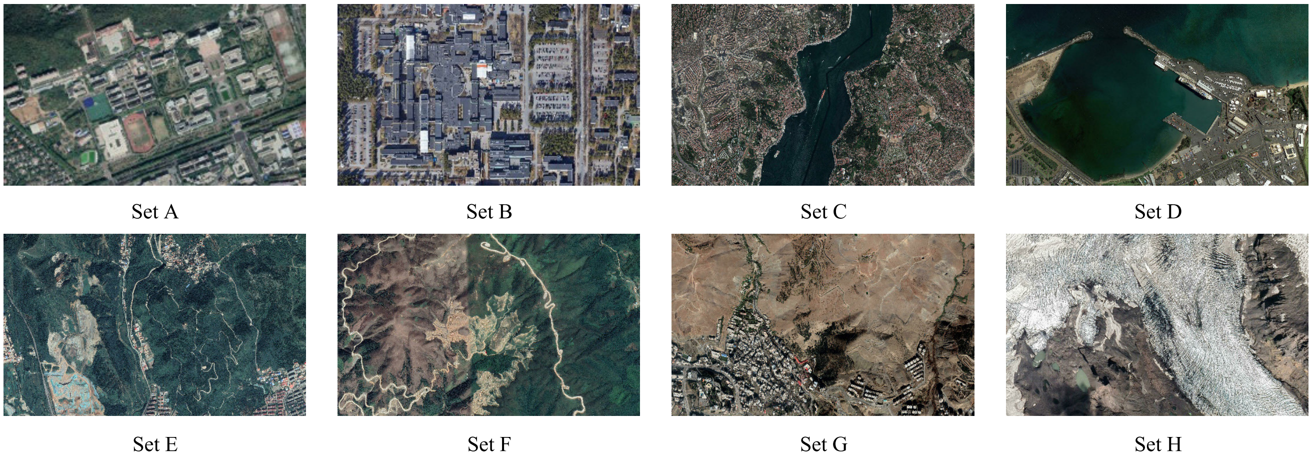

Set A
Set B
Set C
Set D
Set E
Set F
Set G
Set H

FIGURE 19. Schematic diagram of host tile satellite image sets.

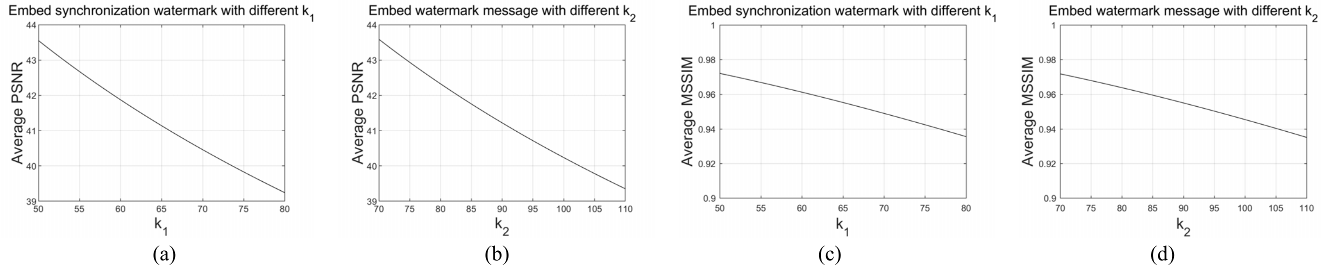


FIGURE 20. The influence of different embedding strength.

forests, plateaus, and mountains. These host data are obtained from Google Earth. The monitors we used are a 27-inch ‘ThinkVision P27q’ monitor with 2560 × 1440 pixels and a 24-inch ‘HP EliteDisplay E242’ monitor with 1920 × 1200 pixels. To mimic a real-world scenario, the monitors are not explicitly calibrated. A P30PRO with 40 MP pixels and an iPhone8 Plus with dual 12 MP pixels are used as the photography equipment.

In Section 5.1, the selection of embedding strength is discussed. In Section 5.2, we analyze the accuracy of watermark synchronization. In Section 5.3, the selection of threshold for message extraction is presented. In Section 5.4 and Section 5.5, we verify the robustness of the proposed method against common image processing attacks and screen-cam attacks, respectively.

A. SELECTION OF EMBEDDING STRENGTH

Embedding strength is used to balance the imperceptibility and robustness of the watermarking scheme. In our method, k_1 and k_2 define the embedding strength of synchronization tiles and message tiles. 1,000 tile satellite images are used for statistical experiments to select the appropriate embedding strength. The widely used peak signal-to-noise ratio (PSNR) and mean structural similarity index (MSSIM) [46] are used for image quality degradation evaluation. The average PSNR and MSSIM values of the 1,000 tiles after embedding synchronization watermarks with different k_1 and after embedding watermark message with different k_2 are shown in Fig.20.

To achieve good imperceptibility, the PSNR values are mostly controlled to be above 40 dB. Therefore, we choose $k_1 = 69$ and $k_2 = 96$ where the average PSNR values are 40.61 and 40.59, respectively. The PSNR and MSSIM value distribution over the whole test image set is shown in Fig.21 when using the selected embedding strength. The watermarked tile satellite image Set A and Set B are shown in Fig.22.

B. PERFORMANCE EVALUATION OF SYNCHRONIZATION POINTS VERIFICATION

The true positive rate and false positive rate of synchronization point verification will affect the efficiency and success rate of watermark detection. Therefore, we perform a static experiment with tile satellite image set A, 27-inch monitor, and P30PRO. The images are captured at different shooting conditions, where the shooting angles are 0°, 15°, 30°, and the shooting distances are from 30cm to 100cm at an interval of 10cm. For all 24 captured images, we calculate the $P_{s,i}$ and $P_{v,i}$. Then, we manually judge the $P_{s,i}$ are the true synchronization points or false synchronization points and compare them with the verification result $P_{v,i}$. The results of synchronization points verification are listed in Table 1.

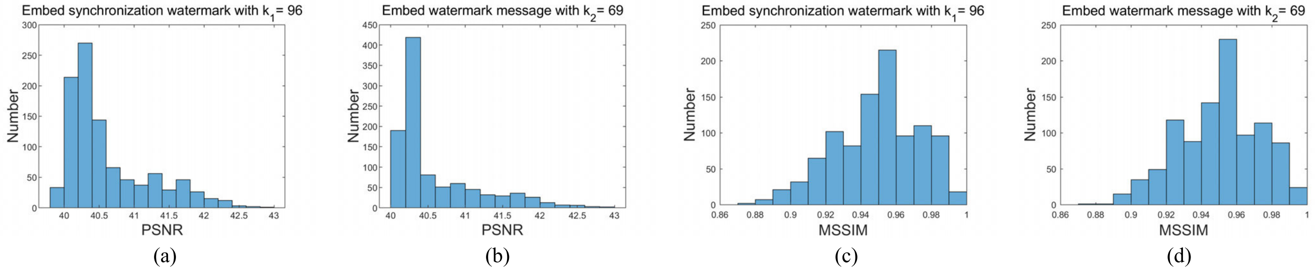


FIGURE 21. PSNR and MSSIM values with selected embedding strength.

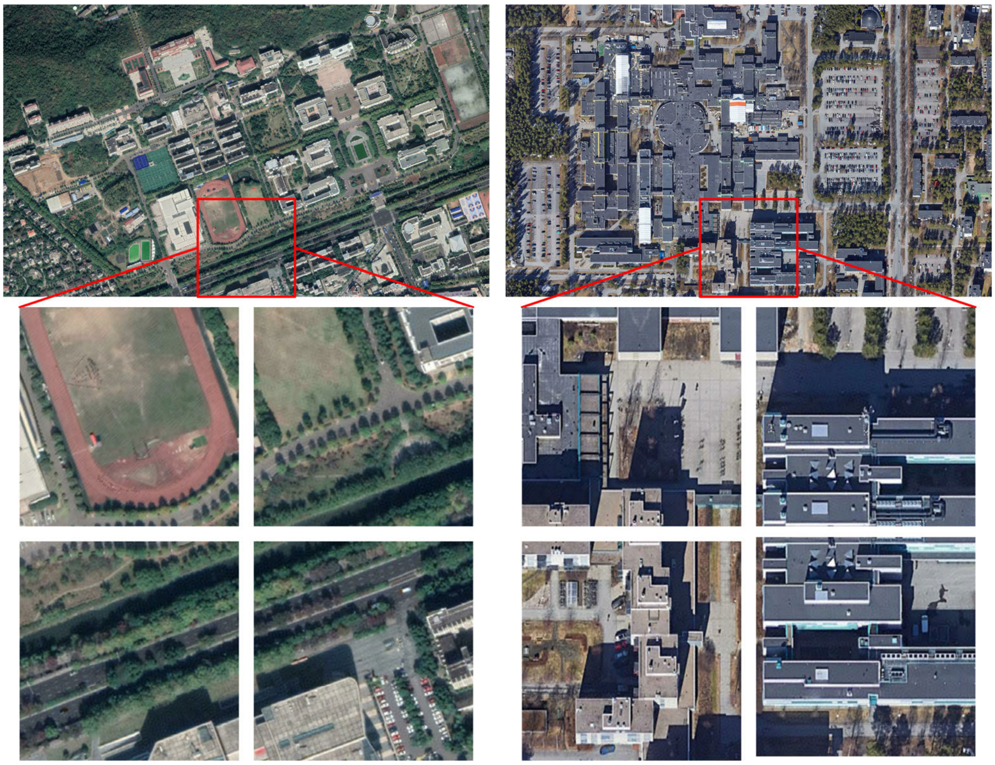


FIGURE 22. Watermarked tile satellite images of Set A and Set B.

TABLE 1. Performance of synchronization points verification.

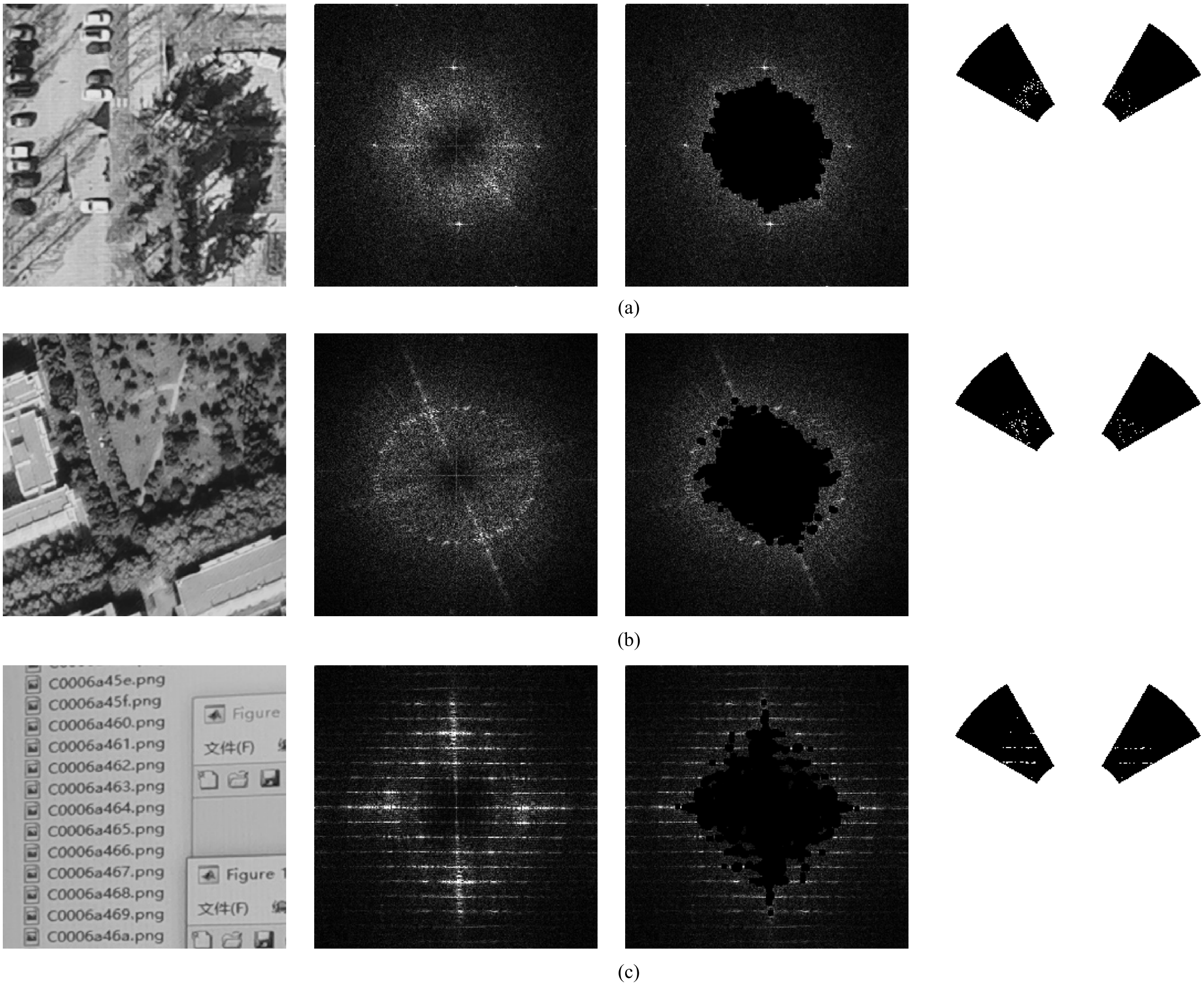
	True synchronization points	False synchronization points	Total
Verified to be true	249	4	253
Verified to be false	3	167	170
Total	252	171	423

The true positive rate is 98.81% and the false positive rate is 2.34%. Therefore, the verification method can be considered effective.

Three examples of verified false synchronization response points are shown in Fig.23, from left to right are the corresponding luminance spectrum, the Fourier magnitude of the noise component, the masked Fourier magnitude spectrum and the detected results, respectively. Fig.23 (a) is the example of a detected false synchronization point in Fig.8 (b), the false detection is mainly due to the texture of the luminance spectrum itself, which will cause some high magnitude values in the set detection region. Fig.23 (b) is the example of a detected false synchronization point in Fig.8 (c), it contains both the synchronization watermark and the watermark message. Because the synchronization watermark is not clear, which means it is not close to the center of the synchronization tile, the point is verified to be false. Fig.23 (c) is a case of false detection we encountered during the experiment, it can also be effectively verified.

C. ANALYSIS OF WATERMARK SYNCHRONIZATION ACCURACY

To verify the validity of watermark synchronization, we perform an experiment with tile satellite image set A, where the centers of the synchronization tiles are marked in red after the



(a)

(b)

(c)

FIGURE 23. **Examples of verified false synchronization response points.**

watermarks are embedded based on the scheme in Section 3. Then, we perform watermark detection of captured images with different shooting conditions until the S'_j of the first point set is found. After that, we manually select the positions of the red marker points and calculate the distances between S'_j and the corresponding marker points. The distance is the offset distance between the auto-corrected position and the real position of the center of a synchronization tile. We use this distance to quantitatively describe the accuracy of watermark synchronization.

The average offset distances of the first point set with different shooting conditions are shown in Fig.24. When the shooting angle is perpendicular to the screen, the average offset distance is around 10 pixels, and the corrected result can be considered to be good. For a shooting angle of 15° or 30°, the offset distances are less than 15 pixels when the shooting distance is within 90cm, which is still acceptable for watermark extraction.

The calculated $R_{f,i}$ of a point set, which represent the same synchronization tiles as the point set in Fig.13, from a captured image with a shooting distance of 60cm and a shooting angle of 15° is shown in Fig.25 as an example. The red dots on the underlying image are the mark we added on the original image. Dots with white edges in the center are the positions of $P_{v,i}$. Dots with magenta edges are the position that has the maximum $R_{f,i}$. As seen, the corrected points are

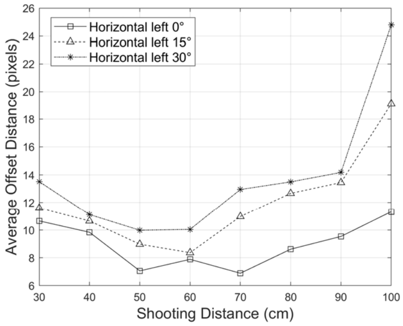


FIGURE 24. **Average offset distances of first point set with different shooting conditions.**

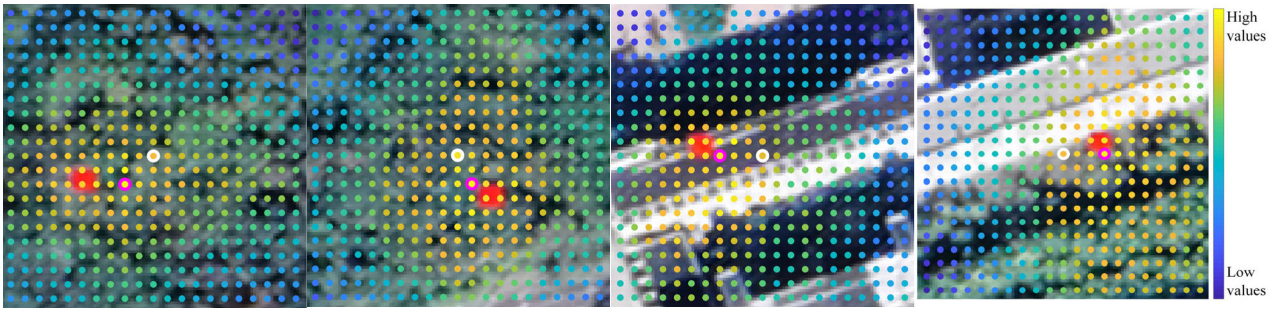


FIGURE 25. Corrected results of a point set from a marked image.

very close to the centers of synchronization tiles and this kind of offset is acceptable.

D. SELECTION OF THRESHOLD FOR MESSAGE EXTRACTION

According to Equation 12, k_3 defines the threshold for message extraction, which is used for judgment of message bits. For statistical experiments in this section, 72 well-captured images with shooting distances between 60 to 80cm and shooting angles between 0° to 30° are used. The images are manually perspective corrected in order to reduce the impact on the selection of k_3 due to the inaccurate perspective correction. Only the message tiles extracted by the first point set of their S_j are used, which are totally 360 extracted message tiles. Average erroneous bits of all these extracted tiles with different k_3 are shown in Fig.26. Therefore, k_3 is set to 1.5 in our experiment.

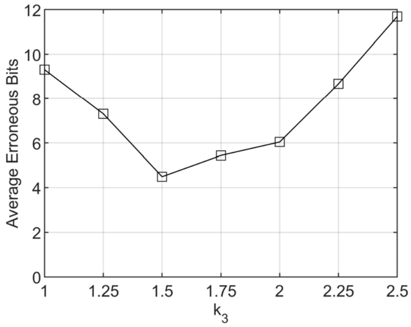


FIGURE 26. Average erroneous bits with different threshold.

E. ROBUSTNESS TO COMMON ATTACKS

In actual use, tile satellite images will encounter some common leakage attacks in addition to screen-cam attack. The most common leakage attack of tile satellite images is to re-stitch the tiles after illegal downloading the tile data or take a screenshot directly to obtain the content, which is a combination of stitching, cropping and translation attacks. Therefore, we use eight screenshot images of the eight host image sets to test the performance of the proposed algorithm to common attacks.

Bit Error Ratio (BER) is a commonly used metrics for robustness evaluation. Due to message repetition, we employ a Minimum Bit Error Ratio (MBER) to evaluate robustness, which is the detected minimum erroneous bits versus the encoded message sequences. Considering the fact that two sub-sequences are embedded, MBER is computed by

$$\mathrm{MBER} = \frac{\mathrm{Min}\,(\mathrm{E}_1) + \mathrm{Min}(\mathrm{E}_2)}{\mathrm{L}_1 + \mathrm{L}_2}, \tag{13}$$

where $\mathrm{Min}\,(\mathrm{E}_1)$ and $\mathrm{Min}(\mathrm{E}_2)$ define the minimum erroneous bits in all the extracted message sequences of sub-sequence 1 and sub-sequence 2, respectively. L_1 and L_2 define the length of sub-sequence 1 and sub-sequence 2, respectively.

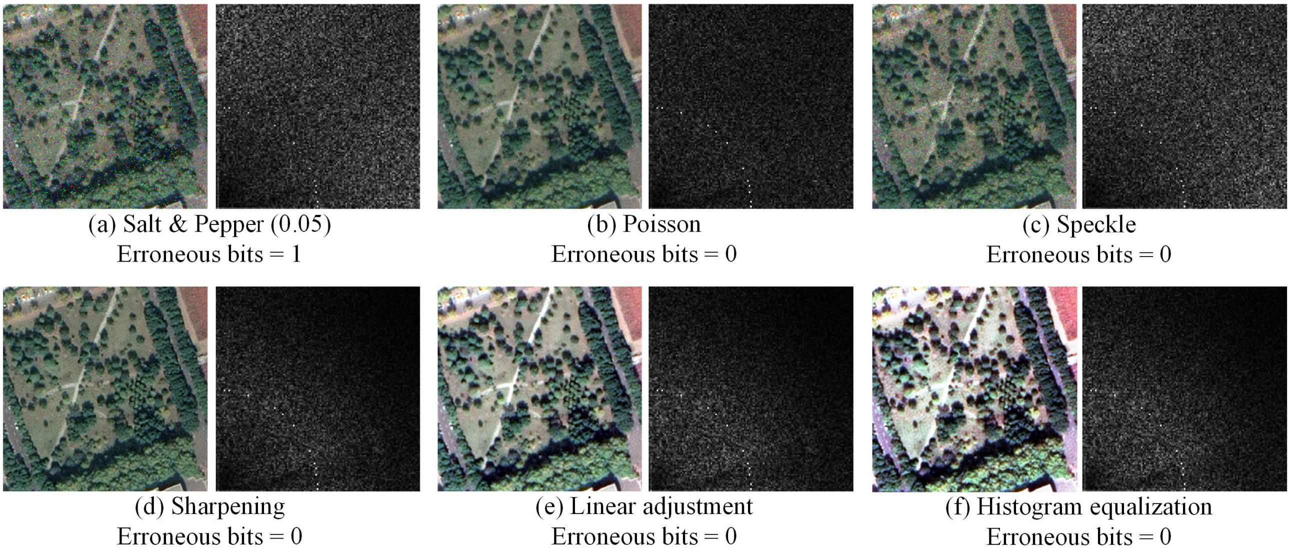
Table.2 lists the MBER of experiments with common attacks. The backslash symbols mean detection failed because no point set is constructed. The message watermark strength is greater than that of the synchronization watermark, so in the experiment of common attacks, there is no case where the message recovery fails when the point set is constructed. Since the tile satellite images have already undergone stitching, cropping and translation attacks, we perform other common image process attacks here.

The performance to JPEG is good, the message can still be effectively recovered at a JEPG compression quality of 20%. The robustness to zoom in is excellent, the message can be recovered perfectly under a scaling 250% attack. For zoom out, the algorithm is robust until a scaling 70% attack. With regard to rotation attack, because the synchronization watermark is embedded in the set region of DFT domain, the positions of watermarked coefficients will also rotate as the image rotates. We have defined a detection range, which is considered to be enough in practical applications.

The proposed algorithm extracts watermarks based on the high DFT magnitude coefficient values of the noise component. If extra noise is added, some values of DFT magnitude coefficients in the detection range may vary to higher values, which will cause misjudgment of watermark synchronization and error of message extraction. Hence, the proposed algorithm is weak to Gaussian Noise. Nevertheless, if we only consider the Gaussian noise attack and do not consider the desynchronization attacks, by extracting the watermark directly from the embedded positions with the DFT domain of the luminance spectrum rather than the noise component,

TABLE 2. MBER under different common attacks.

Attacks	MBER							
	Set A	Set B	Set C	Set D	Set E	Set F	Set G	Set H
JPEG 40%	0/94	0/94	0/94	0/94	0/94	0/94	0/94	0/94
JPEG 30%	0/94	1/94	0/94	2/94	0/94	0/94	0/94	0/94
JPEG 20%	5/94	6/94	6/94	7/94	3/94	2/94	5/94	6/94
JPEG 10%	\	\	\	\	\	\	\	\
Scaling 250%	0/94	0/94	0/94	0/94	0/94	0/94	0/94	0/94
Scaling 200%	0/94	0/94	0/94	0/94	0/94	0/94	0/94	0/94
Scaling 80%	0/94	0/94	0/94	0/94	0/94	0/94	0/94	0/94
Scaling 70%	0/94	3/94	2/94	0/94	0/94	1/94	0/94	1/94
Scaling 60%	\	\	\	\	\	\	\	\
Rotation 10°	0/94	0/94	0/94	0/94	0/94	0/94	0/94	0/94
Rotation 15°	0/94	0/94	0/94	0/94	0/94	0/94	0/94	0/94
Rotation 30°	\	\	\	\	\	\	\	\
Rotation 90°	0/94	0/94	0/94	0/94	0/94	0/94	0/94	0/94
Median filter 3×3	0/94	0/94	0/94	3/94	0/94	0/94	0/94	0/94
Gaussian Noise (0.05)	\	\	\	\	\	\	\	\
Salt & Pepper (0.05)	3/94	3/94	3/94	2/94	1/94	2/94	1/94	2/94
Poisson	0/94	1/94	1/94	0/94	0/94	1/94	0/94	0/94
Speckle	2/94	1/94	2/94	0/94	0/94	0/94	1/94	0/94
Sharpening	0/94	1/94	1/94	1/94	0/94	0/94	0/94	0/94
Linear adjustment	0/94	0/94	0/94	0/94	0/94	0/94	0/94	0/94
Histogram equalization	0/94	0/94	0/94	0/94	0/94	0/94	0/94	0/94



(a) Salt & Pepper (0.05)
Erroneous bits = 1

(b) Poisson
Erroneous bits = 0

(c) Speckle
Erroneous bits = 0

(d) Sharpening
Erroneous bits = 0

(e) Linear adjustment
Erroneous bits = 0

(f) Histogram equalization
Erroneous bits = 0

FIGURE 27. Extracted message tile with different common attacks and corresponding first quadrant of DFT magnitude spectrum.

watermark detection can still be successful. Fortunately, the proposed method is robust to other noise attacks of Salt & Pepper with 0.05 noise density, Poisson and Speckle. Examples of one extracted message tile with these attacks and the detected erroneous bits of this tile are shown in Fig.27.

In the use of satellite images, in order to obtain better visual effects, it is likely to have undergone an image enhancement process. Hence, we also test the effect of sharpening, linear adjustment and histogram equalization attacks. Examples are shown in Fig.27. The results show high performance against these image enhancement attacks.

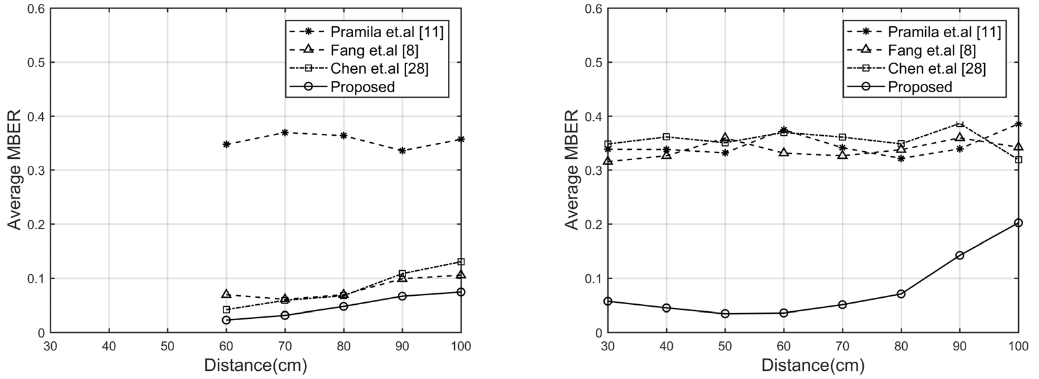
Thus, the proposed watermarking scheme has excellent robustness to re-stitch, screenshot attacks, and other common attacks.

F. ROBUSTNESS TO SCREEN-CAM ATTACK

In this section, the robustness of the proposed method against screen-cam attack with different shooting distances and angles is tested. First, we compare the proposed method with three state-of-art algorithms [8], [11] and [28]. Due to the different purpose of algorithm design, when using these three methods, we regard the tile satellite image set as a whole image, and adjust the size of watermark embedding regions accordingly. For fair comparison, the imperceptibility of the watermarked images is controlled at the same level by adjusting the embedding strength, as shown in Table.3. The transverse comparison experiment is set as: a shooting angle of 15° and a shooting distance between 30 and 100cm. Fig.28(a) indicates the average MBER of different methods

TABLE 3. Images generated by different methods.

Methods	Pramila *et.al* [11]	Fang *et al* [8]	Chen *et al* [28]	Proposed
Part of watermarked tile satellite image set				
PSNR (dB)	37.4459	40.3966	39.8715	40.0206
MSSIM	0.9551	0.9786	0.9753	0.9735



(a) Average MBER with manual correction

(b) Average MBER with blind detection

FIGURE 28. Comparison of different methods with a shooting angle of 15° and different shooting distances.

TABLE 4. Average MEBR with different shooting conditions.

Horizontal angle (left)	Shooting distance (cm)							
	30	40	50	60	70	80	90	100
0°	0.0433	0.0479	0.0220	0.0190	0.0312	0.0410	0.0464	0.0517
15°	0.0570	0.0448	0.0342	0.0357	0.0509	0.0707	0.1421	0.2021
30°	0.0790	0.0752	0.0638	0.0691	0.0699	0.1041	0.1778	0.2576

with manual correction. When shooting distance is between 30 and 50cm, the captured images do not contain the whole screen. Therefore, we cannot manually perform perspective correction based on the four corners of the screen. When shooting distance is between 60 and 100cm, our algorithm has a better performance than other methods. Fig.28(b) indicates the average MBER of different methods with blind detection. It is easy to see that the other methods do not work, and the proposed method still has a good performance when shooting distance is between 30 and 80cm. Although the three algorithms are applicable in the framework that they are designed for, only the proposed method works with blind detection, which is important in the real applications.

In theory, without considering external interference, the distortion caused by shooting from the horizontal left is similar to the distortion caused by shooting from the horizontal right. The distortion caused by shooting from a horizontal angle is similar to the distortion caused by shooting from a vertical angle with a 90-degree rotation of the image. Therefore, the experiments are designed as follows. We set the shooting angle from being perpendicular to the screen up to 30° of horizontal left at intervals of 15°. The shooting distance ranges from 30 cm to 100 cm at intervals of 10 cm. There are fourteen groups of experiments, where totally 336 captured images and their detection results are shown in Fig.29. There are two cases of failed detection. One of them is no point set is constructed, so we cannot perform message extraction. Another is erroneous synchronization, which means the extracted message will be wrong with a high MBER. In this case, we can go to the next point set or the detection is failed. The average MEBR with different shooting conditions are listed in Table.4, where the situations that no point set is constructed are not taken into consideration.

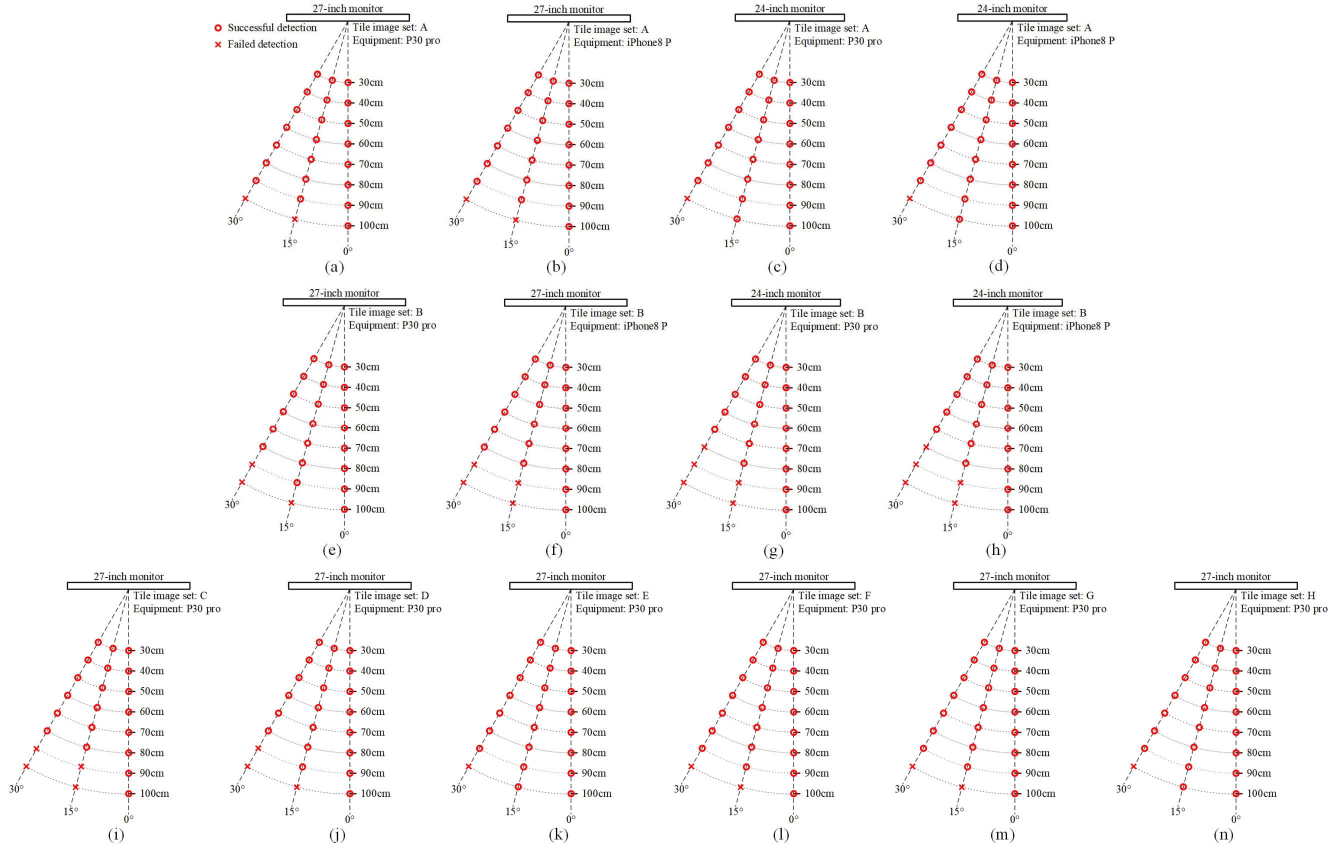
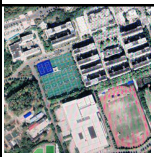


FIGURE 29. Watermark detection results against screen-cam attack.

TABLE 5. Example of Experiment Group (a).

Horizontal angle (left)	Shooting distance/cm							
	30	40	50	60	70	80	90	100
0°								
MBER	0.0426	0.0319	0.0213	0.0213	0.0319	0.0426	0.0426	0.0532
15°								
MBER	0.0532	0.0426	0.0319	0.0426	0.0426	0.0638	0.0957	0.1277
30°								
MBER	0.0851	0.0638	0.0745	0.0638	0.0638	0.0851	0.0851	0.1489

The MBER of Group (a) are listed in Table.5, where the underlined number means message recovery failed. The figures in Table.5 are the area of the 9 tiles in the range of the point set (similarly to Fig.16 (b)) where the MBER is detected.

The proposed method has good robustness to shooting distances and shooting angles. When shooting perpendicularly, all the detections are successful. When shooting at an angle of 15° or 30°, most detections of the captured images with a shooting distance below 80cm are successful. As shown

TABLE 6. False positives and false negatives of watermark detection.

	Watermarked	Simulated unwatermarked	Total
Detected	169	0	169
Not detected	23	192	215
Total	192	192	384

in Table.5, when the shooting distance is close, like 30cm, the moiré noise is obvious. Fortunately, as long as the magnitude coefficients which represent the noise in the Fourier domain are not at the region of the embedded watermark, it will not affect the detection. Therefore, all the detections at close shooting distances are successful. As the shooting distance increases, the image becomes more blurred, and the details of the synchronization watermark decrease, which affect the accuracy of perspective correction, especially with a shooting angle. Therefore, when shooting at an angle of 15° or 30°, the detections of some captured images with long shooting distances are failed.

In addition to camera quality, there are two other factors that affect the robustness of the algorithm. One is the ratio of screen size to screen resolution. The size of the image captured by the smartphone is fixed. Hence, at the same shooting distance, the ratio will affect the zoom of the data. Therefore, in theory, at the same shooting distance, the captured image of the 24-inch monitor should be clearer than those of the 27-inch display in our experiment. Therefore, the detection results of experiment Group (c) and (d) are better than those of Group (a) and (b). The other one is the image itself, i.e., some texture characteristics of the tiles that will affect the watermark detection. For example, the texture of some tiles of image set B causes some high magnitude values in the detection region of watermark, which may be mistaken for the embedded watermark. Therefore, the detection results of image set A are better than those of image set B. In addition, this adverse effect is also amplified by the quality of the equipment. The quality of the 24-inch monitor is worse than that of the 27-inch monitor. Therefore, the detection results of experiment Group (e) and (f) are better than those of Group (g) and (h).

In order to obtain a good quality of captured image, the content displayed on the screen commonly will be captured at a distance of around 50 to 70 cm and an angle of around 0° to 30°. Therefore, the proposed method can meet the practical needs.

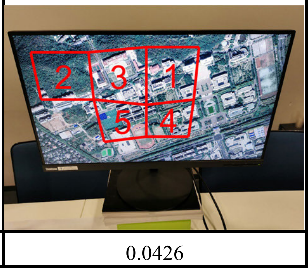
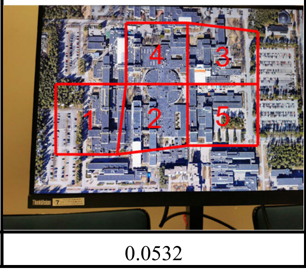
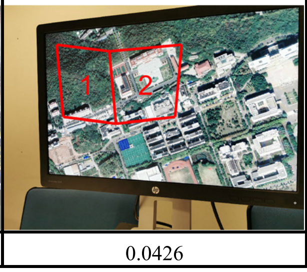
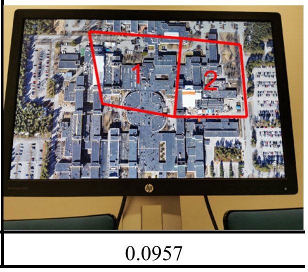
In our method, after the construction of point sets, if S_j exists, it can generally be considered that the watermark exists. According to the performance of synchronization points verification, unwatermarked images theoretically will have no S_j, which means our method can better prevent the occurrence of false positives. In order to check how many false positives may occur during the message extraction process, the 192 captured images are additionally detected at a radius where no watermark is embedded, which is to simulate the detection results of unwatermarked images but with wrong S_j. The detection results are shown in Table.6, and, as seen, there are still no false positives.

The captured images we used above are taken with a tripod. Because hand-hold shooting will cause camera shake, which can cause more blur, we also tested the performance at other tilt shooting angles with handhold shooting. The results are shown in Table.7, which has a good performance.

VI. CONCLUSIONS

This paper proposes a screen-cam robust and blind watermarking scheme for tile satellite images. To achieve blind watermark synchronization from a captured image with perspective distortion, we propose a DFT-based synchronization watermark embedding method. A response index is designed

TABLE 7. Examples of Handhold Shooting.

Captured image				
Detected point sets				
MBER	0.0426	0.0532	0.0426	0.0957

to estimate an appropriate scale level and the positions of embedded synchronization watermarks from the noise component at the detection side. To ensure the effectiveness of watermark synchronization, we propose a verification method to verify whether the detected positions are from the centers of embedded synchronization watermarks, a point set construction method to select all point sets that can be used for perspective correction, and a method to find precise locations to improve the accuracy of perspective correction. However, the perspective correction will never be perfect. In order to extract a watermark message effectively, a message extraction algorithm based on the maximum value in the local 3 × 3 area centered on the positions where a message bit is embedded and the DFT domain of the noise component is proposed. This scheme has proved to be robust to common attacks and screen-cam attack. The future work is to improve the efficiency of the automatic perspective correction method, so that it can also be applied to real-time copyright verification scenarios and increase the watermark capacity.

REFERENCES

[1] H. Liu, B. Zhao, and L. Huang, "A remote-sensing image encryption scheme using DNA bases probability and two-dimensional logistic map," *IEEE Access*, vol. 7, pp. 65450–65459, 2019.

[2] X. Zhang and X. Wang, "Remote-sensing image encryption algorithm using the advanced encryption standard," *Appl. Sci.*, vol. 8, no. 9, p. 1540, Sep. 2018.

[3] X. Zhang, G. Zhu, and S. Ma, "Remote-sensing image encryption in hybrid domains," *Opt. Commun.*, vol. 285, no. 7, pp. 1736–1743, Apr. 2012.

[4] G. Xu, H. Li, Y. Dai, K. Yang, and X. Lin, "Enabling efficient and geometric range query with access control over encrypted spatial data," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 4, pp. 870–885, Apr. 2019.

[5] S. Jha, S. Sural, V. Atluri, and J. Vaidya, "Specification and verification of separation of duty constraints in attribute-based access control," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 897–911, Apr. 2018.

[6] K. Xue, W. Chen, W. Li, J. Hong, and P. Hong, "Combining data owner-side and cloud-side access control for encrypted cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 8, pp. 2062–2074, Aug. 2018.

[7] A. Castiglione, A. De Santis, B. Masucci, F. Palmieri, A. Castiglione, and X. Huang, "Cryptographic hierarchical access control for dynamic structures," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 10, pp. 2349–2364, Oct. 2016.

[8] H. Fang, W. Zhang, H. Zhou, H. Cui, and N. Yu, "Screen-shooting resilient watermarking," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 6, pp. 1403–1418, Jun. 2019.

[9] 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 MUM*, 2004, pp. 101–108.

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

[11] 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, no. 2, pp. 211–222, Jun. 2012.

[12] A. Keskinarkaus, A. Pramila, and T. Seppänen, "Image watermarking with a directed periodic pattern to embed multibit messages resilient to print-scan and compound attacks," *J. Syst. Softw.*, vol. 83, no. 10, pp. 1715–1725, Oct. 2010.

[13] 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.

[14] 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, Aug. 2017.

[15] K. Gourrame, H. Douzi, R. Harba, R. Riad, F. Ros, M. Amar, and M. Elhajji, "A zero-bit Fourier image watermarking for print-cam process," *Multimedia Tools Appl.*, vol. 78, no. 2, pp. 2621–2638, Jan. 2019.

[16] R. Riad, R. Harba, H. Douzi, F. Ros, and M. Elhajji, "Robust Fourier watermarking for ID images on smart card plastic supports," *Adv. Electr. Comput. Eng.*, vol. 16, no. 4, pp. 23–30, 2016.

[17] R. Riad, F. Ros, R. Harba, H. Douzi, and M. El Hajji, "Pre-processing the cover image before embedding improves the watermark detection rate," in *Proc. 2nd World Conf. Complex Syst. (WCCS)*, Nov. 2014, pp. 705–709.

[18] A. Poljicak, L. Mandic, and D. Agic, "Discrete Fourier transform-based watermarking method with an optimal implementation radius," *J. Electron. Imag.*, vol. 20, no. 3, Jul. 2011, Art. no. 33008.

[19] 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.

[20] 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.

[21] T. Jassim, R. Abd-Alhameed, and H. Al-Ahmad, "A new robust and fragile watermarking scheme for images captured by mobile phone cameras," in *Proc. 1st Int. Conf. Commun., Signal Process., Their Appl. (ICCSPA)*, Feb. 2013, pp. 1–5.

[22] A. Pramila, A. Keskinarkaus, and T. Seppänen, "Multiple domain watermarking for print-scan and JPEG resilient data hiding," in *Proc. Int. Workshop Digit. Watermarking (IWDW)*, Dec. 2008, pp. 279–293.

[23] A. Keskinarkaus, A. Pramila, T. Seppänen, and J. Sauvola, "Wavelet domain print-scan and JPEG resilient data hiding method," in *Proc. Int. Workshop Digit. Watermarking (IWDW)*, Nov. 2006, pp. 82–95.

[24] A. Keskinarkaus, A. Pramila, and T. Seppänen, "Image watermarking with feature point based synchronization robust to print-scan attack," *J. Vis. Commun. Image Represent.*, vol. 23, no. 3, pp. 507–515, Apr. 2012.

[25] N.-T. Le, "Invisible watermarking optical camera communication and compatibility issues of IEEE 802.15.7r1 specification," *Opt. Commun.*, vol. 390, pp. 144–155, May 2017.

[26] M. Iwata, N. Mizushima, and K. Kise, "Practical watermarking method estimating watermarked region from recaptured videos on smartphone," *IEICE Trans. Inf. Syst.*, vol. 100, no. 1, pp. 24–32, 2017.

[27] M.-J. Lee, K.-S. Kim, H.-Y. Lee, T.-W. Oh, Y.-H. Suh, and H.-K. Lee, "Robust watermark detection against D–A/A-D conversion for digital cinema using local auto-correlation function," in *Proc. 15th IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 425–428.

[28] W. Chen, N. Ren, C. Zhu, and Q. Zhou, "Screen-cam robust image watermark with feature-based synchronization," to be published.

[29] X. Wang, *Research on Digital Watermarking Algorithm Oriented to Screen Capture Process for Remote Sensing Image*. Nanjing, China: Nanjing Normal Univ., 2018.

[30] M. Urvoy, D. Goudia, and F. Autrusseau, "Perceptual DFT watermarking with improved detection and robustness to geometrical distortions," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 7, pp. 1108–1119, Jul. 2014.

[31] L.-Y. Hsu and H.-T. Hu, "A reinforced blind color image watermarking scheme based on schur decomposition," *IEEE Access*, vol. 7, pp. 107438–107452, 2019.

[32] P. Niu, X. Shen, T. Wei, H. Yang, and X. Wang, "Blind image watermark decoder in UDTCWT domain using weibull mixtures-based vector HMT," *IEEE Access*, vol. 8, pp. 46624–46641, 2020.

[33] 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. 2s, pp. 1–23, Feb. 2015.

[34] W. Lu, H. Lu, and F.-L. Chung, "Feature based robust watermarking using image normalization," *Comput. Electr. Eng.*, vol. 36, no. 1, pp. 2–18, Jan. 2010.

[35] X. Wang, J. Wu, and P. Niu, "A new digital image watermarking algorithm resilient to desynchronization attacks," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 4, pp. 655–663, Dec. 2007.

[36] Q. Ying, J. Lin, Z. Qian, H. Xu, and X. Zhang, "Robust digital watermarking for color images in combined DFT and DT-CWT domains," *Math. Biosci. Eng.*, vol. 16, no. 5, pp. 4788–4801, 2019.

[37] G. Larbi, H. Roumaissa, and H. Sara, "Embedding watermark in the magnitude matrix of the DFT of image," in *Proc. Int. Conf. Comput. Pattern Recognit. ICCPR*, Jun. 2018, pp. 106–110.

[38] N. Jimson and K. Hemachandran, "DFT based coefficient exchange digital image watermarking," in *Proc. 2nd Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Jun. 2018, pp. 567–571.

[39] M. T. Mirza, Q. Ahmed, S. Munib, A. Khan, and R. K. Khalil, "A new hybrid domain based print-scan resilient image watermarking technique," in *Proc. 12th Int. Conf. Frontiers Inf. Technol.*, Dec. 2014, pp. 170–175.

[40] 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.

[41] L. A. Delgado-Guillen, J. J. Garcia-Hernandez, and C. Torres-Huitzil, "Validating the existence of watermarks on digital images using a mobile phone," in *Proc. Int. Conf. Internet Technol. Secured Trans.*, Dec. 2012, pp. 51–55.

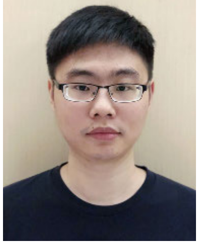
[42] J. Ouyang, G. Coatrieux, B. Chen, and H. Shu, "Color image watermarking based on quaternion Fourier transform and improved uniform log-polar mapping," *Comput. Electr. Eng.*, vol. 46, pp. 419–432, Aug. 2015.

[43] M. Hamidi, M. E. Haziti, H. Cherifi, and M. E. Hassouni, "Hybrid blind robust image watermarking technique based on DFT-DCT and arnold transform," *Multimedia Tools Appl.*, vol. 77, no. 20, pp. 27181–27214, Oct. 2018.

[44] S. Voloshynovskiy, S. Pereira, A. Herrigel, N. Baumgärtner, and T. Pun, "Generalized watermark attack based on watermark estimation and perceptual remodulation," *Proc. SPIE*, vol. 3971, pp. 358–370, May 2000.

[45] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

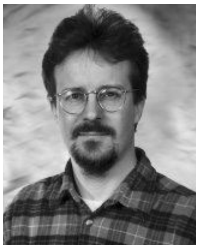
[46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.



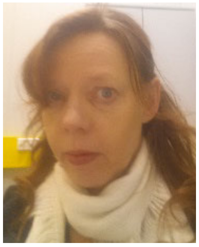
WEITONG CHEN received the B.S. degree in geography and the M.S. degree in marine geography from Nanjing Normal University, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree in cartography and geography information system with the School of Geography. His research interests include digital watermarking, geo-data security, and GIS.

CHANGQING ZHU received the B.S. degree in mathematics from PLA Information Engineering University, in 1982, the M.S. degree in mathematics from Zhengzhou University, in 1990, and the Ph.D. degree in cartography from PLA Information Engineering University, in 1997. He was a Professor with PLA Information Engineering University. He is currently a Professor with the School of Geography, Nanjing Normal University. His research interests include digital watermarking, data security, and GIS.

NA REN received the B.S. and M.S. degrees from Shaanxi Normal University, in 2004 and 2008, respectively, and the Ph.D. degree from Nanjing Normal University, in 2011. She is currently an Associate Professor with the School of Geography, Nanjing Normal University. Her research interests include digital watermarking, data security, and GIS.



TAPIO SEPPÄNEN is currently a Professor of biomedical engineering with the University of Oulu, Finland. His research interests include biomedical signal processing, multimedia signal processing, and pattern recognition.



ANJA KESKINARKAUS received the M.Sc. degree in information engineering and the Dr.Tech. degree from the University of Oulu, Finland, in 1998 and 2012, respectively. She currently works as a Postdoctoral Researcher with the Center for Machine Vision and Signal Analysis, University of Oulu. Her research interests include digital watermarking, multimedia signal processing and management, intelligent transport systems, biobanks, signal processing for biomedical signals, and biomedical engineering.