Multimedia Tools and Applications (2021) 80:29805–29826
https://doi.org/10.1007/s11042-021-11179-2

A dual-tamper-detection method for digital image authentication and content self-recovery

Tong Liu[1] · Xiaochen Yuan[1,2]

Received: 18 September 2020 / Revised: 12 January 2021 / Accepted: 24 June 2021 /
Published online: 13 July 2021

Abstract

This paper proposes an approach to protect image content against malicious tampering based on watermarking technology. The watermark is composed of two kinds of check bits which are used for tampered region localization, and one recovery bit which is used for image recovery and is embedded into the three-Least Significant Bit planes of the original image. The first check bit is generated by applying the proposed Parity Check Bit Labeled method to each pixel, and the other is generated by employing hashing algorithm to each block after image decomposition. The superposition result detected from the two check bits contributes to lowering the probability of false-negative errors. Moreover, we propose a post-processing method Adaptive Structural Element Calculation which improves the accuracy of tamper detection result further. Experimental results show that our algorithm has good performance in keeping high quality of recovered image, and meanwhile improving the accuracy of tamper detection result.

Keywords Dual-tamper-detection · Parity check bit labeled · Adaptive structural element calculation · Tamper detection and content self-recovery

1 Introduction

With the phenomenal development of digital communication technology, multimedia information goes hand in hand with people's social life. The multimedia technique riches our life but also brings

✉ Xiaochen Yuan
xcyuan@must.edu.mo

Tong Liu
1809853nii20001@student.must.edu.mo

[1] Faculty of Information Technology, Macau University of Science and Technology, Macau 853, China

[2] Guangdong-Hong Kong-Macao Joint Laboratory for Intelligent Micro-Nano Optoelectronic Technology, Foshan University, Foshan 528225, China

some troubles to us. On the one hand, it is general for multimedia information to be irresponsibly tampered with by digital editing software, which will bring plenty of misleading to the people who do not know the fact [1, 27, 29, 34, 36]. On the other hand, the author's copyright lacks protection, and as a result, works are easy to be copied and misappropriated. Thus, to protect their work and copyright, people used to append some specific marks declaring author's information. However, these marks will damage the work's content and integrity. Therefore, the steganography approach is considered to hide data into the self-carrier, such as audio, image, and video, which can guarantee the transparency of hidden information [9, 16, 18, 23, 25, 26, 30, 37].

Originally, steganography is mainly used for hiding secret information into messages defined as the cover objects [7, 12, 31]. The image is suitably chosen as the cover object because of its high degree of redundancy. An excellent image-based steganography system has some requirements, which are given as follow: (1) minimizing the perceptual difference between the stego and the cover image, (2) increasing the payload capacity of the cover images, (3) improving the security which can protect the scheme against various attacks [1]. However, it is really a challenge meeting the first two requirements at the same time. In other words, it is not easy to achieve a good balance between the transparency and high capacity of the system, since the high payload capacity will affect the quality of stego-images. Generally, the hidden information, which does not associate with the cover object, can be embedded into the frequency domain and spatial domain. However, for the purposes of image authentication, the secret information should be related to the cover images. Therefore, based on the steganography system, an invisible fragile watermarking technique is proposed, making hidden data undetectable with human eyes.

Unlike steganography, with the watermarking method, the confidential information hidden in the images is generated from the carrier itself. Since our final purposes are image authentication and protecting the integrity of content, rather than just hiding secret information. From this perspective, this fragile watermarking can be considered as a development of steganography. Actually, the fragile watermarking scheme could find out the tampered region of the forged image in an active way, which means that the information should be inserted into the cover image prior to transmission [15, 17, 22, 30]. Since the fragile watermark is sensitive to information variation, we can exactly utilize this characteristic for authentication. Additionally, this kind of active scheme can not only detect the tampered region but also recover it to the original form. Two types of bits are embedded for image authentication and content self-recovery as the watermark [6, 14, 19, 24]. Thereinto, check bit is utilized for image authentication, and recovery bit is used for image self-recovery. Also, there some passive [18, 31] methods that can authenticate images and do not need to embed data previously. However, comparing with the active methods, they are not sensitive enough and can not recover the forged contents.

Recently, some related schemes have been proposed for image integrity verification and detection of various forgeries. Elaskily et al. [13] proposed an automatic two-stage Copy-Move Forgery Detection (CMFD) methodology, which categorized the candidate image into forged or original. After the proposed matching stage and refinement stage, the target image was segmented into many objectives according to their similarity. The experimental result has shown that the best accuracy among the MICC-F220 dataset is 99.09%. However, this scheme only could detect which image is forged, but fail to locate the tampered region precisely. Moghaddasi et al. [21] presented an authentication scheme based on low-dimensional singular value decomposition of discrete cosine transform (DCT) coefficients. This scheme utilized a roughness measure algorithm to calculate the obtained single value and a support vector machine (SVM) algorithm to distinguish authenticated and spliced images. The proposed method has shown an average detection accuracy of 97.15%. But it was also unable to locate the specific spliced regions inside images. Agarwal et al. [2] created a passive method to detect copy-move attack on images. This method used a deep learning technique to extract features from segmented patches, which showed effective performance. However, this method could only achieve tamper detection, not further recover the forged region to the original form. Dhole et al. [9] proposed a fragile watermarking scheme, which could detect as well as recover the tampered image with its tampered region. They used XOR operation to generate authentication bit calculated by user key, block index, block content, image width, and height. And the recovery bits were generated by DCT method. In this work, the watermarked and recovered images for Lena were 35.14 and 38.45 dB, respectively. AlShehri et al. [3] proposed a scheme using binary rotation invariant and noise tolerant (BRINT) for tamper detection and extreme learning machine (ELM) for image recovery. This work was efficient in tamper detection, but its recovery ability was not better than that of the state-of-the-art approaches. Sarreshtedari et al. [30] presented a fragile watermarking method for image authentication and self-recovery. The watermark was composed of check bits and recovery bits. Thereinto, the hash data was considered for authentication, and the recovery bits were generated by Set Partitioning In Hierarchical Trees (SPIHT) algorithm. In this work, although some of recovery bits were lost caused by tampering, the scheme could still recover it. But its performance in terms of tamper detection was not good as recovery. Overall, although using passive methods [2, 13, 21] for image authentication does not require the image to be processed in advance, these schemes have no ability to recover the forged contents. Comparing with them, these active methods [3, 9, 30] are more sensitive for tamper detection, and can recover the images. Therefore, we also proposed an active scheme based on watermarking, which can both achieve tamper detection and self-recovery.

The hashing method is popular in generating the check bit in previous works due to its collision-resistant characteristic. However, because of the quantitative limitation of embedded check bit, some coincidences can make false-negative errors that directly affect the quality of reconstructed image. To lower the probability of false-negative errors, we propose a Dual-Tamper-Detection scheme for digital image authentication and furthermore for content self-recovery. The watermark is composed of one recovery bit generated by the SPIHT algorithm, and two check bits, which are the Hash-based check bit and the Parity Check Bit Labeled (PCBL)-based check bit, respectively. The hashing algorithm is applied to each block to generate the Hash-based check bit, and the PCBL algorithm is proposed to each pixel to produce the other check bit. To further improve tamper detection accuracy and meanwhile ensure the quality of content self-recovery, we propose an Adaptive Structural Element Calculation (ASEC) algorithm to calculate an appropriate structural element to adapt to different images.

The rest of this paper is organized as follows: firstly, Section 3 describes the generation of check bits and recovery bits respectively and gives the details of watermark embedding. Section 4 shows a specific description of our proposed Dual-Tamper-Detection scheme. Next, the experimental results and analysis are given in Section 5. Finally, the conclusions are summarized in Section 5.

2 Watermark generation and embedding

The watermark information is composed of check bits for tamper detection and recovery bits for image content recovery. Generally speaking, it is the tampered region that needs to be recovered, therefore, theoretically the more accurate the tampered region is localized, the better the region can be recovered. In this paper, to obtain a better recovery result, we propose the Dual-Tamper-Detection method which uses two categories of check bits to improve the accuracy of tamper detection, thus to bring a better reconstructed effect. In the proposed method, the check bits are generated in different ways. For the first check bit generation, hashing algorithm is chosen for block authentication due to its collision resistant feature. For the second check bit generation, we design the PCBL approach to increase the accuracy of tamper detection. According to the human visual system, a slight change of LSB will not make a huge difference on the content of images, therefore, based on this property, we propose to embed the watermark information into the LSB bit planes. By extracting the watermark information after transmission, the tampered region can be localized and furthermore recovered.

Given the original image $I_O$which is an 8-bit grayscale image, Fig. 1 gives the flowchart of watermark generation and embedding of the proposed scheme. Firstly, the m-bit Most Significant Bits (MSB) of I_O are extracted, denoting as I_{O_m}, thus each pixel of I_{O_m} is composed of the m-bit MSB in which the main information is concentrated. Then the proposed PCBL algorithm is applied to the extracted I_{O_m} to generate the PCBL-based check bits W_{CB_P} from pixel level, and the employed hashing algorithm is applied to the blocks of I_{O_m} after decomposition, to generate the Hash-based check bits W_{CB_H}. Meanwhile, the SPIHT encoding is applied to the original image I_O to generate the recovery bits W_{RB}. The generated PCBL-based check bits W_{CB_P}, Hash-based check bits, and recovery bits W_{RB} comprise the watermark accordingly. Finally, by reconstructing the watermark information with the extracted I_{O_m}, the corresponding watermarked image I_W is generated. Fig. 2 shows a demonstration of watermark generation and embedding. The procedures of our watermark embedding algorithm is as follows:

```
Algorithm I: Procedures of Watermark Embedding
Input: Original image ($I_O$)
Output: Watermarked image ($I_W$)
Step. 1 Extract the m-bit MSB image $I_{O\_m}$ containing the main information of the original image $I_O$.
Step. 2 Apply the proposed PCBL algorithm into each pixel of $I_{O\_m}$, and generate the PCBL-based check bits
    $W_{CB\_P}$.
Step. 3 Divide $I_{O\_m}$ into non-overlapped blocks of size $b_r \times b_c$, apply the MD5 algorithm into each block of $I_{O\_m}$,
    and generate the Hash-based check bits $W_{CB\_H}$.
Step. 4 Apply the SPIHT encoding algorithm into $I_O$, and generate the recovery bits $W_{RB}$.
Step. 5 Reconstruct the $I_{O\_m}$, $W_{CB\_H}$, $W_{CB\_P}$, $W_{RB}$ and generate the watermarked image $I_W$.
```

2.1 Generation of Hash-based Check Bits

As a part of the watermark component, check bits are generated for the purpose of tamper detection. To generate the Hash-based check bits W_{CB_H}, m-bit MSB image I_{O_m} is divided into non-overlapped blocks of size $b_r \times b_c$. Then considering the characteristics of MD5 Message-Digest Algorithm [35] that it turns diverse data into a fixed-size value, we propose to employ the MD5 as hashing function to generate check bits from the block pixels. The MD5 algorithm is a widely used hash function generating 128-bit hash values. MD5 processes variable-length messages into 128-bit (16-byte) fixed-length outputs. Regardless of the size of the input data, MD5 can output a fixed-length and unique hash value. This is a great advantage for us, and it helps to ensure the integrity of the information. When the image is tampered, the generated hash value of the tampered blocks will also be changed. In this way, we can locate the position of the tampered blocks. In the employed hashing algorithm, after applying the MD5 function

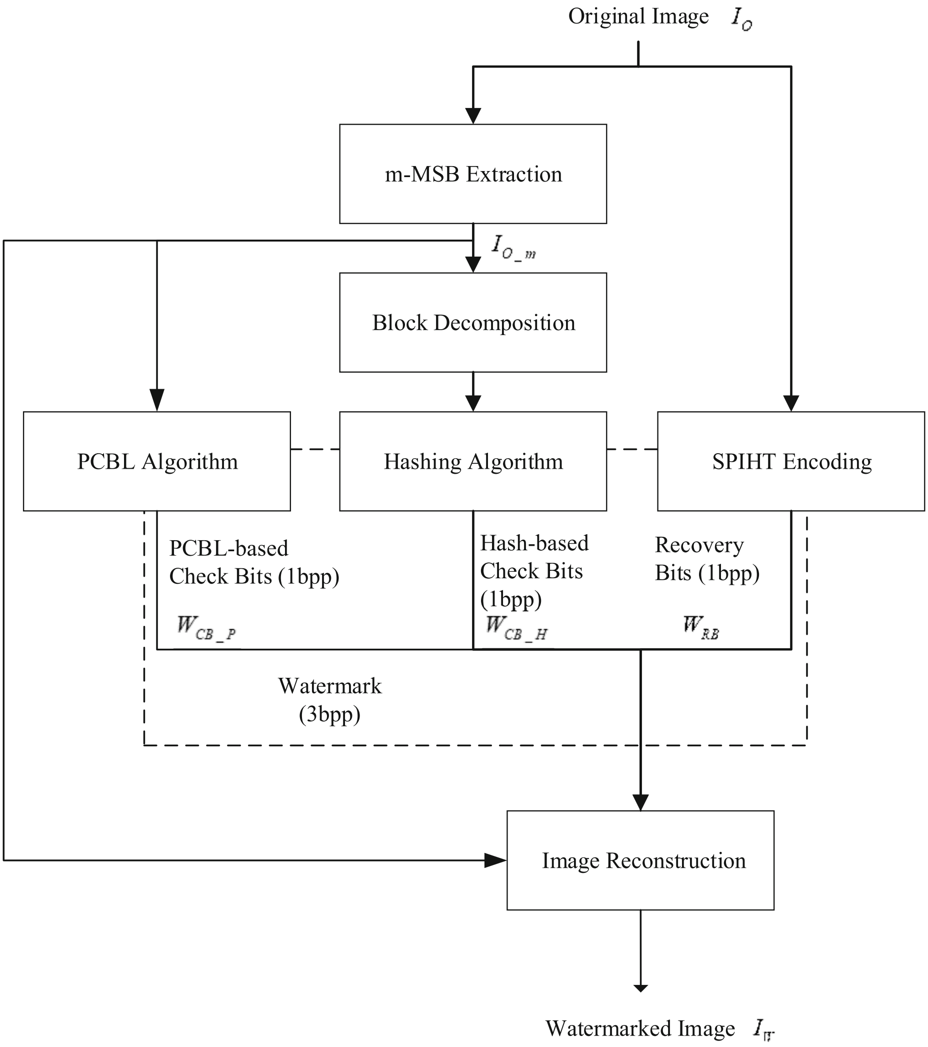

Fig. 1 Flowchart of watermark generation and embedding

into each block, the output is converted into its binary form, and by truncating the first $b_r \times b_c$ bits of which, the block check bits W_{CB_H} are generated accordingly. It can be seen that the size of the generated W_{CB_H} consists with the blocks. Therefore after transforming all the blocks, the size of the generated W_{CB_H} is $M \times N$ bits, given the size of the original image as $M \times N$ pixels.

2.2 Generation of PCBL-based check bits

In addition to the Hash-based check bit W_{CB_H}, we propose the PCBL algorithm to generate the second check bit, the PCBL-based check bit W_{CB_P}, to increase the accuracy of tamper detection. Being different from the Hash-based check bit produced from block level, the PCBL-based check bit is generated based on pixels.

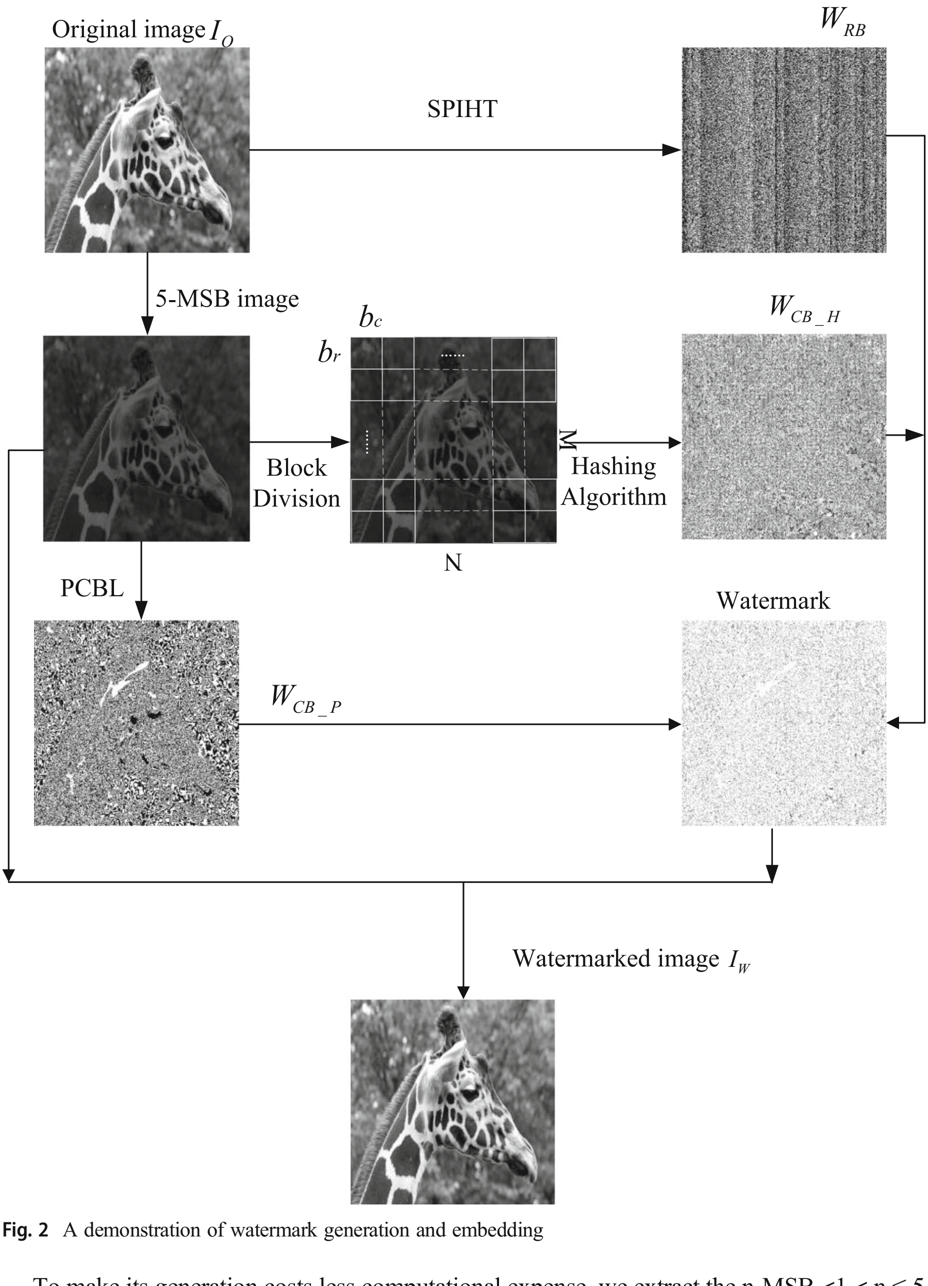

Fig. 2 A demonstration of watermark generation and embedding

To make its generation costs less computational expense, we extract the n-MSB <1 < n ≤ 5 > value of each pixel and judge whether it is odd or even. The PCBL-based check bit W_{CB_P} is generated successively by (1) as follow:

$$W_{CB\_P} = \begin{cases} 1, & \text{if } Pi \text{ is odd} \\ 0, & \text{else} \end{cases} \tag{1}$$

where P_i is n-MSB value of the i^{th} pixel in an image.

2.3 Generation of recovery bits

For the purpose of recovery bits generation, we propose to embed the compressed version of the original image I_O into its LSB bit plane. The SPIHT [28], which is widely used in digital signals compression, is employed to compress I_O. As an embedded-compression algorithm, SPIHT can truncate its output bit stream at the desired rate and come to a certain reconstruction of the original image. Since the quality of reconstruction depends on the output rate exploited, for better quality, the SPIHT sorts the rounded multi-resolution wavelet transform coefficients according to their magnitudes and transmits them based on significant bit order [30]. The same process will be used availably to the decoder inversely as well. These similarities can be found through wavelet transform spatial orientation trees as shown in Fig. 3. Therefore, we may exploit different compression rates to satisfy different goals. In this method, we set the output rate of SPIHT to be 1 bit per pixel (bpp), hence the output of SPIHT algorithm is truncated into $M \times N$ bits, which is the size of the generated recovery bits W_{RB}.

3 Proposed Dual-Tamper-Detection scheme

Under the premise of watermark embedding, Dual-Tamper-Detection scheme, which uses two categories of check bits to detect image from block and pixel level respectively, is proposed to locate the tampered region of a received image $I_{R\text{cvd}}$ and to recover it to its original form. This scheme is designed to improve the accuracy of tampering identification under the guarantee of the high quality of image reconstruction. Given the received watermarked image $I_{R\text{cvd}}$ which is an 8-bit grayscale image, Fig. 4 shows the flowchart of tampered region localization and image self-recovery, and the specific algorithm is as follows:

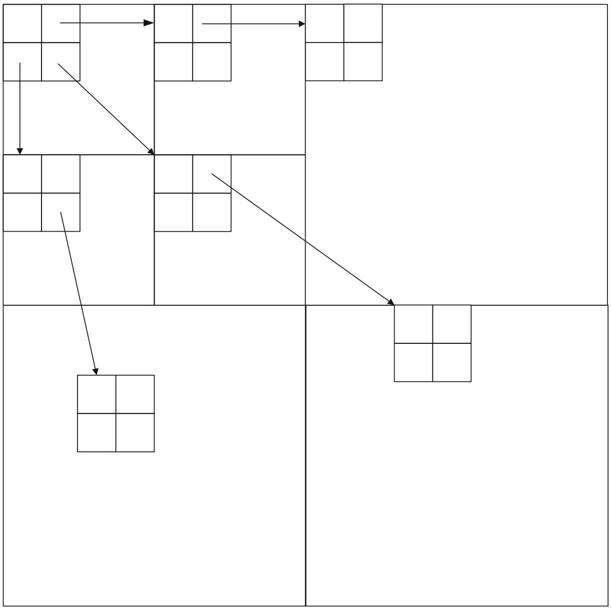

Fig. 3 Examples of root-leaves dependencies in the spatial-orientations of image pyramid decomposition

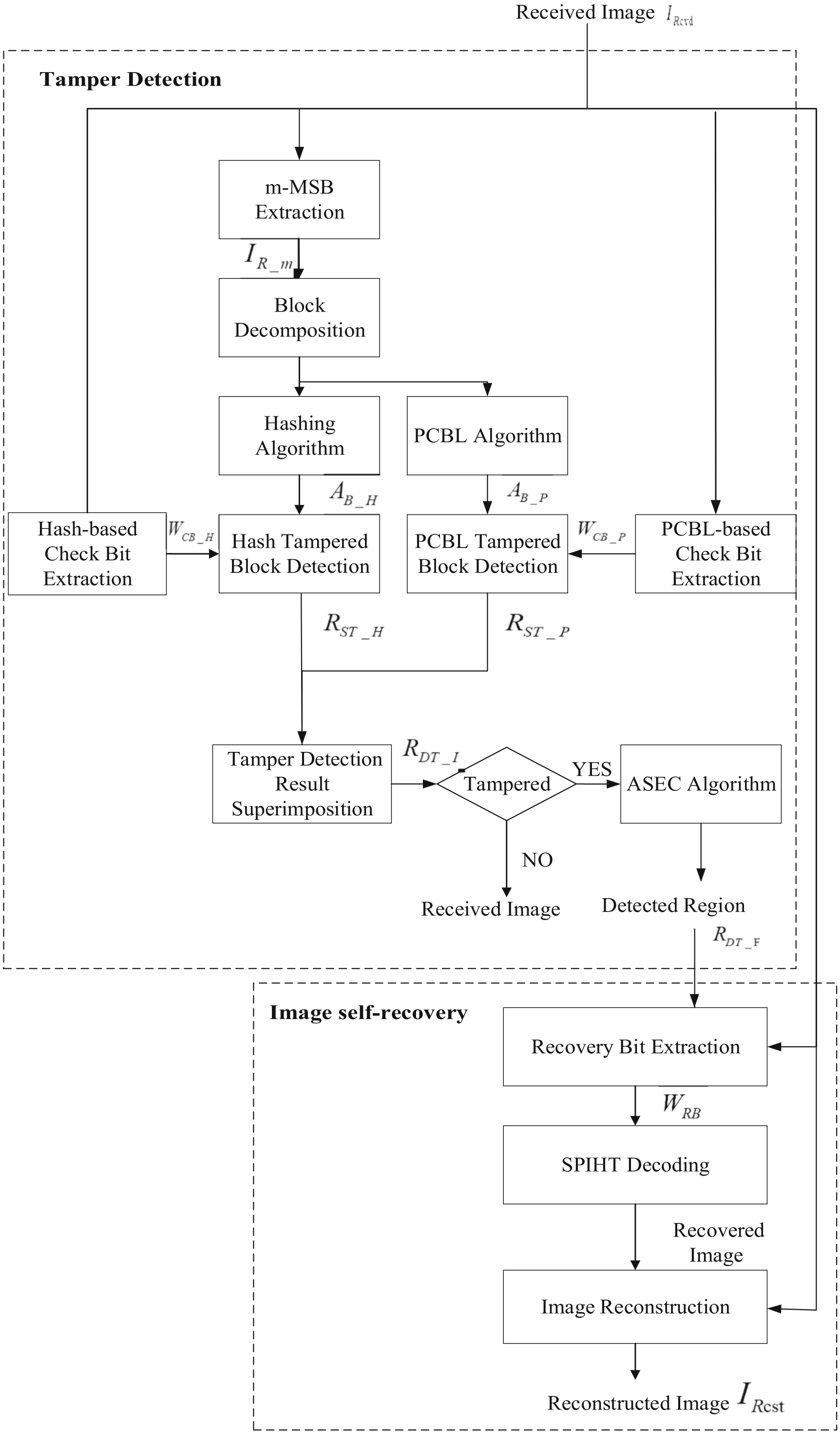

Fig. 4 Flowchart of tampered region localization and image self-recovery

Algorithm II: Procedures of Tampered Region Localization And Image Self-Recovery.
Input: Received image ($I_{R\text{cvd}}$)
Output: Reconstructed image ($I_{R\text{cst}}$)

Step. 1 Extract the m-MSB image I_{R_m}, the Hash-based Check Bit W_{CB_H}, the PCBL-based Check Bit W_{CB_P}, and the Recovery Bit W_{RB} from the received image $I_{R\text{cvd}}$.
Step. 2 Divide I_{R_m} into non-overlapped blocks of size $b_r \times b_c$, apply the MD5 algorithm into each block, and generate the Hash-based Authentication Bit A_{B_H}.
Step. 3 Apply the proposed PCBL algorithm into each pixel of I_{R_m}, and generate the PCBL-based Authentication Bit A_{B_P}.
Step. 4 Compare A_{B_H} with W_{CB_H} to obtain the hashing-suspected tampered region R_{ST_H}, and compare A_{B_P} with W_{CB_P} to obtain the PCBL-suspected tampered region R_{ST_P}.
Step. 5 Superimpose R_{ST_P} and R_{ST_H} together to get the initial detected region R_{DT_I}.
Step. 6 Apply the proposed ASEC algorithm to R_{DT_I}, to generate the final detected region R_{DT_F}.
Step. 7 Employ SPIHT decoding algorithm to W_{RB} to generate the recovered image.
Step. 8 Reconstruct the image by substituting the tampered region detected of R_{DT_F} with recovered image.

3.1 Tampered region localization

To detect whether the received image $I_{R\text{cvd}}$ has been tampered, authentication bits are calculated to compare with the check bits which have been hidden in the watermark. The process of calculating authentication bits is similar as the generation of check bits. Firstly, m-bit MSB image I_{R_m} is extracted from $I_{R\text{cvd}}$, then I_{R_m} is divided into non-overlapped blocks of size $b_r \times b_c$, which is the same as the blocks decomposed in the watermark generation procedure. Then PCBL-based Authentication Bit A_{B_P} is generated by applying the proposed PCBL algorithm to each pixel, and Hash-based Authentication Bit A_{B_H} is produced by employing hashing algorithm to each block. Moreover, check bits W_{CB_P} and W_{CB_H}, which are used to compare with A_{B_P} and A_{B_H} respectively, are extracted from LSB bit planes. Theoretically, if the received image $I_{R\text{cvd}}$ has not been tampered, newly generated authentication bits are same with the check bits extracted. However, in practice we cannot detect the whole tampered region by just comparing them, for the reason that W_{CB_H} and A_{B_H} are truncated hash data of each block, thus there exists contingency that W_{CB_H} and A_{B_H} just happen to be equal, which may cause false-negative errors. Therefore, in order to reduce the probability of false-negative error of blocks, we propose the Dual-Tamper-Detection scheme adding the PCBL algorithm detecting from pixel level. PCBL-suspected tampered region R_{ST_P} is labeled by comparing W_{CB_P} and A_{B_P} of each pixel, and hashing-suspected tampered region R_{ST_H} is labeled by comparing W_{CB_H} and A_{B_H} of each block. Finally, by superimposing the labeled-suspected results of R_{ST_P} and R_{ST_H} together, initial detected region R_{DT_I} is obtained.

To ensure the quality of reconstructed image, we propose the post-processing, which is used to further improve the accuracy of tamper detection result, and apply it intoR_{DT_I}. Considering in the procedure of post-processing, the size of tampered region has a big impact on selection of suitable structural element, we propose the ASEC algorithm which calculates the most appropriate structural element adaptively. The calculation of structural element SE is defined in (2) as follow:

$$SE = \lfloor \lg(a) \rfloor + 4 \tag{2}$$

Where a is the area percentage, which indicates the ratio of initial detected region R_{DT_I} to the whole image $I_{R\text{cvd}}$. The specific procedures of our ASEC algorithm are as follows:

```
Algorithm III: ASEC Algorithm
Input: Initial detected region ($R_{DT\_I}$)
Output: Final detected region ($R_{DT\_F}$)
Step. 1 Calculate the ratio of initial detected region $R_{DT\_I}$ to the whole image $I_{Rcvd}$, and denote it as area
    percentage $a$.
Step. 2 Fill the holes which are fully enclosed in $R_{DT\_I}$.
Step. 3 Calculate the adaptive structural element $SE$ by formula (2).
Step. 4 Eliminate the holes on the edge by expanding the graph obtained by Step.2.
Step. 5 Erode the graph obtained by Step.4.
```

In order to verify the availability of the proposed ASEC algorithm, we conduct the following experiments as shown in Fig. 5, where the first column shows the ground truth of tampered region in small, middle and large size respectively. The second to fourth columns show the detection results with three different structure elements selected, and the best result of each image are highlighted in bold. It can be easily seen that the most adaptive *SE* has positive correlation with the size of tampered region. The last column is the detected result with the *SE* calculated by the proposed ASEC algorithm. Fig. 5 demonstrates the *SE* calculated by ASEC algorithm can obtain a better performance of tampered region identification.

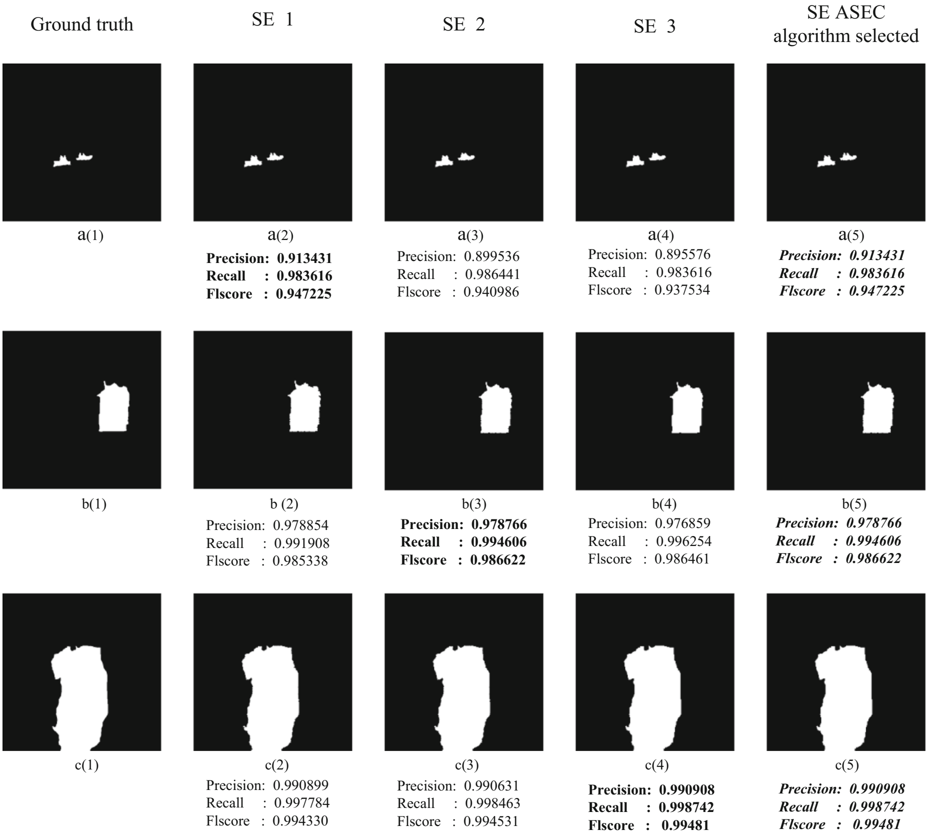

Fig. 5 The verification of proposed ASEC algorithm. **a(1)-c(1)** ground truth; **a(2)-c(2)** tamper detection results under SE = 1; **a(3)-c(3)** tamper detection results under SE = 2; **a(4)-c(4)** tamper detection result under SE = 3; and **a(5)-c(5)** tamper detection result under SE calculated by the proposed **ASEC** algorithm.

3.2 Content self-recovery

Upon the scenario that the received image $I_{R\mathrm{cvd}}$ has been identified as tampered, the content self-recovery is the next step. For the purpose of recovering image, SPIHT decoding algorithm is applied to the Recovery Bit W_{RB} which is extracted from LSB bit planes of $I_{R\mathrm{cvd}}$. As shown in Fig. 6, SPIHT is an effective compression algorithm that encodes an 8-bit grayscale image into only 1-bit recovery information per pixel. The third row presents the recovered images obtained after applying SPIHT decoding algorithm, which shows good capability of recovering the image content. Comparing with the original images, the Peak-Signal-to-Noise-Ratio (PSNR) of recovered images is at least 26.72 dB, which indicates the main information of image contents is recovered. The reconstructed images $I_{R\mathrm{cst}}$ are generated by only replacing the tampered region with the recovered one to further improve the quality of reconstructed image.

4 Experimental results and analysis

The experiments are performed on a Windows 10 PC with an Intel(R) CoreTM i7-4790 3.60GHz CPU and 8 GB RAM. To evaluate the performance of the proposed method, we randomly choose a set of images from the BOWS2 database [5]. To evaluate the performance in terms of tamper detection, we calculate the True Positive (TP) and True Negative (TN) by comparing the detected result with the dataset's ground truth. In addition, False Positive (FP) is the number of blocks which are negative marked incorrectly. On the contrary, False Negative

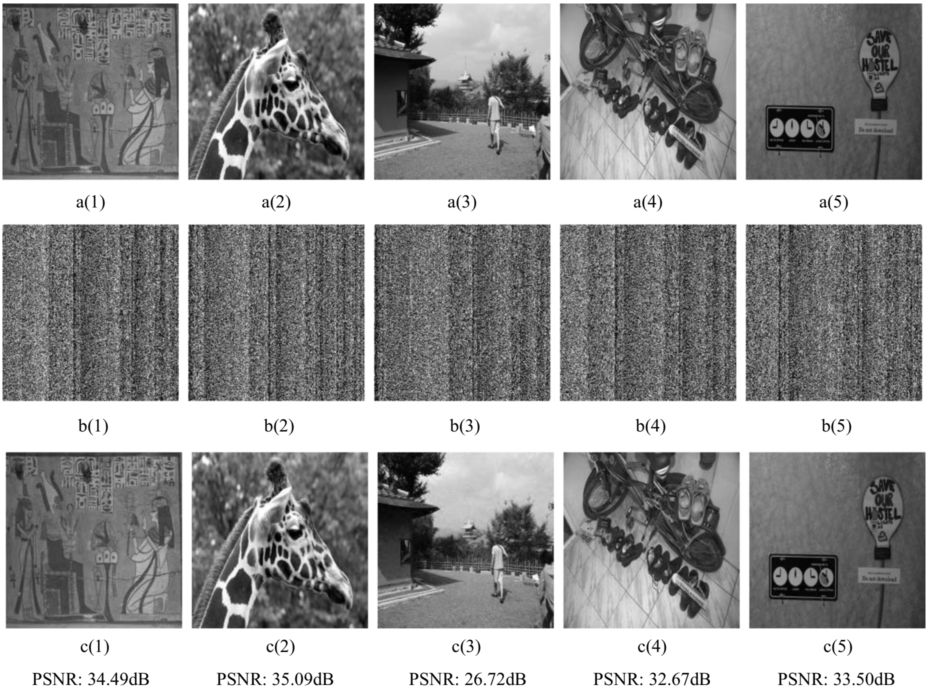

Fig. 6 Example of SPIHT encoding and decoding. a(1)-a(5) original images; b(1)-b(5) SPIHT-encoded images; and c(1)-c(5)) SPIHT-decoded images

(FN) is the number of positive blocks marked as negative. We use two sets of metrics to evaluate the performance of tamper detection. One set calculates *Precision*, *Sensitivity/Recall*, and *F1score*, which are defined in (3) (4) (5) respectively. The *Precision* represents the proportion of the real tampered pixels in those labeled as tampered. The index of *Recall*, also known as *Sensitivity*, measures the capability of identifying a forged pixel as tampered. *F1 score* is a comprehensive evaluation metric that combines *Precision* and *Recall.*

$$Precison = \frac{TP}{TP + FP} \tag{3}$$

$$Sensitivity/Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1score = \frac{2precision \times recall}{precision + recall} = \frac{2TP}{2TP + FP + FN} \tag{5}$$

The other set of evaluation metrics are *Sensitivity*, *Specificity* and *Accuracy* defined in (4), (6), and (7) respectively. The *Specificity* measures the capability of identifying an authentic pixel as authentic. The *Accuracy* presents the overall capability of tamper detection.

$$Specificity = \frac{TN}{TN + FP} \tag{6}$$

$$\text{Accuracy} = \frac{TN + TP}{TN + TP + FP + FN} \tag{7}$$

In addition to tamper detection, the recovery performance is evaluated by *PSNR,* which is used to evaluate the similarity of two images, and it is defined in (8).

$$PSNR = 10\log 10\frac{(MAX)^2}{MSE} \tag{8}$$

$$MSE = \frac{1}{a \times b}\sum_{a=0}^{a-1}\sum_{b=0}^{b-1}\left(f(x,y) - f'(x,y)\right)^2 \tag{9}$$

Where MSE means the Mean Square Error obtained by formula (9), and $a \times b$ is the size of image.

4.1 Parameters setting

To select the most adaptive parameter to conduct experiments, we need to clarify the influence of blocks size on the performance of tampered region localization and image recovery. Therefore, we design the following experiments, which divide the images into blocks of size 2 × 2, 4 × 4 and 8 × 8 respectively.

Figure 7 gives the *PSNR* value of the recovered image under different scenarios. The results indicate no apparent difference in the quality of reconstructed images with different block

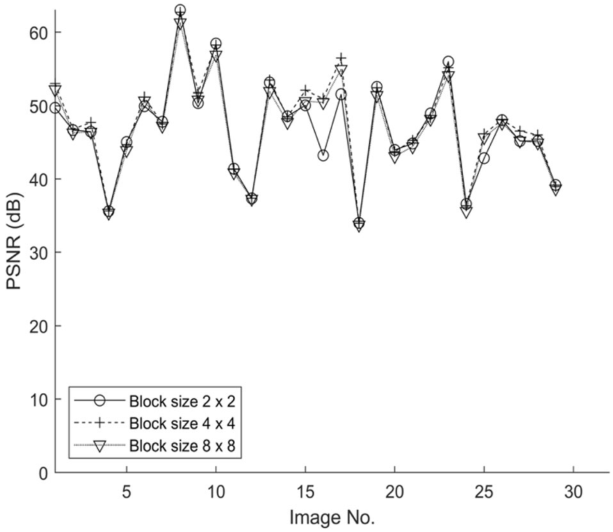

Fig. 7 Performances of content recovery in terms of *PSNR* under different block sizes

sizes. Meanwhile, Figs 8, 9 and 10 show tampering detection performance in terms of *Precision*, *Recall*, and *F1 score* respectively. According to the results, it is evident that the smaller block can achieve higher *Precision*. Therefore the *F1 score* (since *Recall* values remain unchanged basically) is improved, indicating more accurate tampering detection results. Although the small block enhances tampering detection accuracy, it will also cause more computational expenses in theory. Therefore, we calculate the corresponding computational expenses of the proposed scheme with different blocks, which of size 2 × 2, 4 × 4,and 8 × 8, and the average running times are measured as 29.81s, 17.90s, and 15.15s respectively. The results show that even with block size of 2 × 2, the computational expense is small. Therefore, to obtain a better performance in tampering detection, we choose the block of size 2 × 2 in the following experiments.

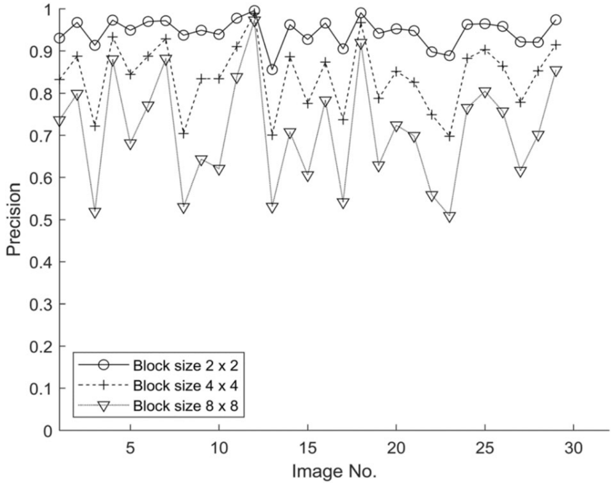

Fig. 8 Performances of tampering detection in terms of *Precision* under different block sizes

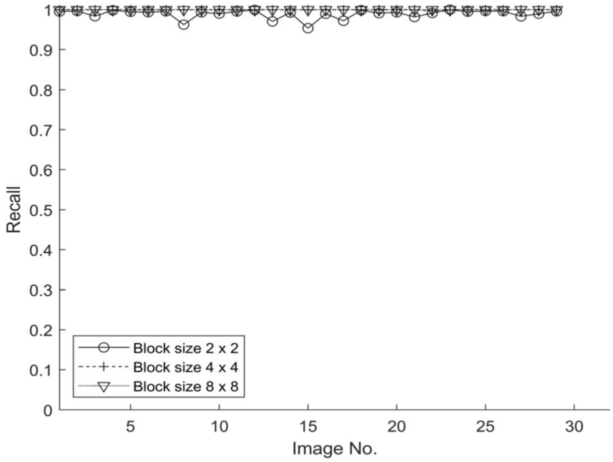

Fig. 9 Performances of tampering detection in terms of *Recall* under different block sizes

4.2 Performance of the proposed scheme

Figure 11 presents a demonstration of the proposed Dual-Tamper-Detection scheme. Five representative host images with various textures are selected from the database [5], and shown in the first row. The second row and third show the corresponding watermarked images and tampered images, respectively. It can be seen that there is no visual distortion in the watermarked images, indicating the 3-LSB scheme can be acceptable by human visual system. The fourth row and fifth row show the ground truth and the tamper detection results respectively. Finally, the recovered images are presented in the sixth row. The performance of tampering detection is measured with *Precision*, *Recall*, and *F1 score* respectively, and meanwhile the recovery performance is measured using *PSNR*. The corresponding results are respectively given in Table 1.

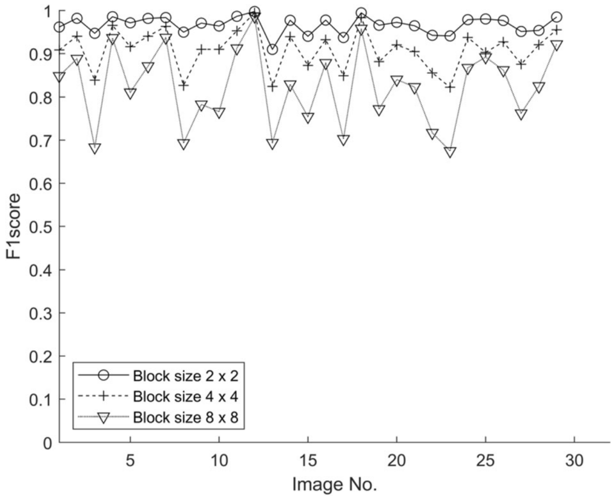

Fig. 10 Performances of tampering detection in terms of *F1 score* under different block sizes

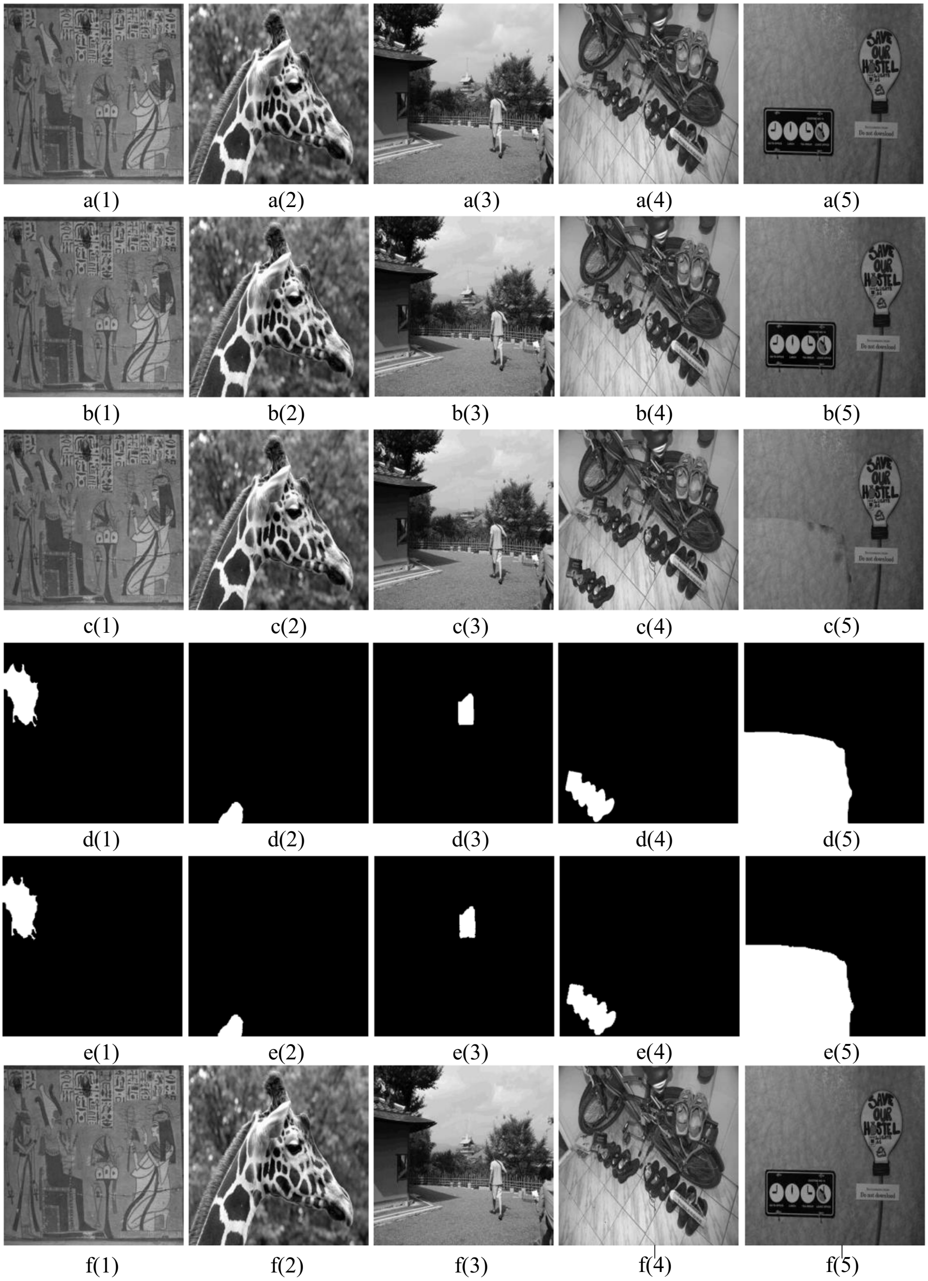

Fig. 11 Demonstration of the proposed Dual-Tamper-Detection scheme. **a(1)-a(5)** the original 8-bit grayscale images, **b(1)-b(5)** the watermarked images generated by our proposed method, **c(1)-c(5)** the tampered images, **d(1)-d(5)** ground truth, **e(1)-e(5)** tampered area detected by our propose method, and **f(1)-f(5)** the recovered images

The results in Fig. 11 and Table 1 illustrate that the proposed Dual-Tamper-Detection method is effective to either a small region of tampering (the second column) or a large region

Table 1 Objective measure of the results in Fig 11

	Metrics	Col. (1)	Col. (2)	Col. (3)	Col. (4)	Col. (5)
TAMPERING DETECTION	Precision (%)	97.2	97.0	97.5	96.4	99.6
	Recall (%)	99.5	99.3	98.5	99.6	99.9
	F1score (%)	98.3	98.1	98.4	98.0	99.7
IMAGE RECOVERY	PSNR (dB)	50.23	49.94	47.25	36.59	37.36

of tampering (the fifth column). In addition, our method also presents a good performance in tampered region identification whenever its edge is smooth (the third column) or curved (the fourth column). Furthermore, some fine details of the tampered region also can be identified by our method, as shown in the first column.

Figure 12 demonstrates the superiority of the proposed tamper detection method. In Fig. 12, the first row, a(1)~a(5), shows the ground truth; the second row, b(1)~b(5), shows the results labeled by hashing algorithm; the third row, c(1)~c(5) shows the results of proposed PCBL algorithm; the fourth row, d(1)~d(5), shows the results of superposition approach of hashing and PCBL; and the fifth row, e(1)~e(5) shows the final detected regions after applying the proposed ASEC algorithm, with which the higher detection accuracy can be achieved. It can be seen that there is a significant difference in the results obtained by different check bits. Especially in the third column, it indicates that the two labeled results are complementary in some cases. Overall, we can easily see that the result of superposition approach is more accurate than either the hashing method or the PCBL method.

In addition to the visual evaluation, we further objectively evaluate the detection performance using some metrics: *Precision*, *Recall*, and *F1 score*. The results corresponding to Fig. 12 are calculated respectively and shown in Table 2. In Table 2, the ‘b’ ~ ‘e’ respectively corresponds to the second to fifth row, and Col. (1) ~ Col. (5) corresponds to the first to fifth column in Fig. 12. It can be seen that there are no significant differences in the different methods in terms of *Precision*, while there is a noticeable improvement in *Recall*, thus producing the improved *F1 score*. The results indicate that the proposed scheme effectively reduces the probability of false-negative errors, therefore increasing the accuracy of tampering detection results.

4.3 Comparison with existing works

In this section, to demonstrate the superiority of our work, the performance of the proposed Dual-Tamper-Detection method is compared with some existing works. In Figs. 13~15, the BOWS2 database [5] is evaluated here to show the tamper detection performance in terms of *Precision*, *Recall*, and *F1 score*. Five methods are compared in these three figures: the Sarreshtedari’s method [30], the proposed PCBL method, the combination of work [30] and PCBL, the Sarreshtedari’s work [30] with proposed ASEC algorithm, and our Dual-Tamper-Detection method.

The work [30] generates the check bit with the hashing method, which is same as one of our check bits generation. To show the one-check-bit method' s detecting capability, the hashing method and proposed PCBL method are firstly compared here. However, it is kind of unfair to directly compare work [30] with the proposed Dual-Tamper-Detection method. Since the ASEC we used is actually a morphological operation, we also test the performance that applies the same ASEC operation on work [30]. From the experimental result we can see that our work still achieves better performance, which also indicates the efficiency of our two-check-bit

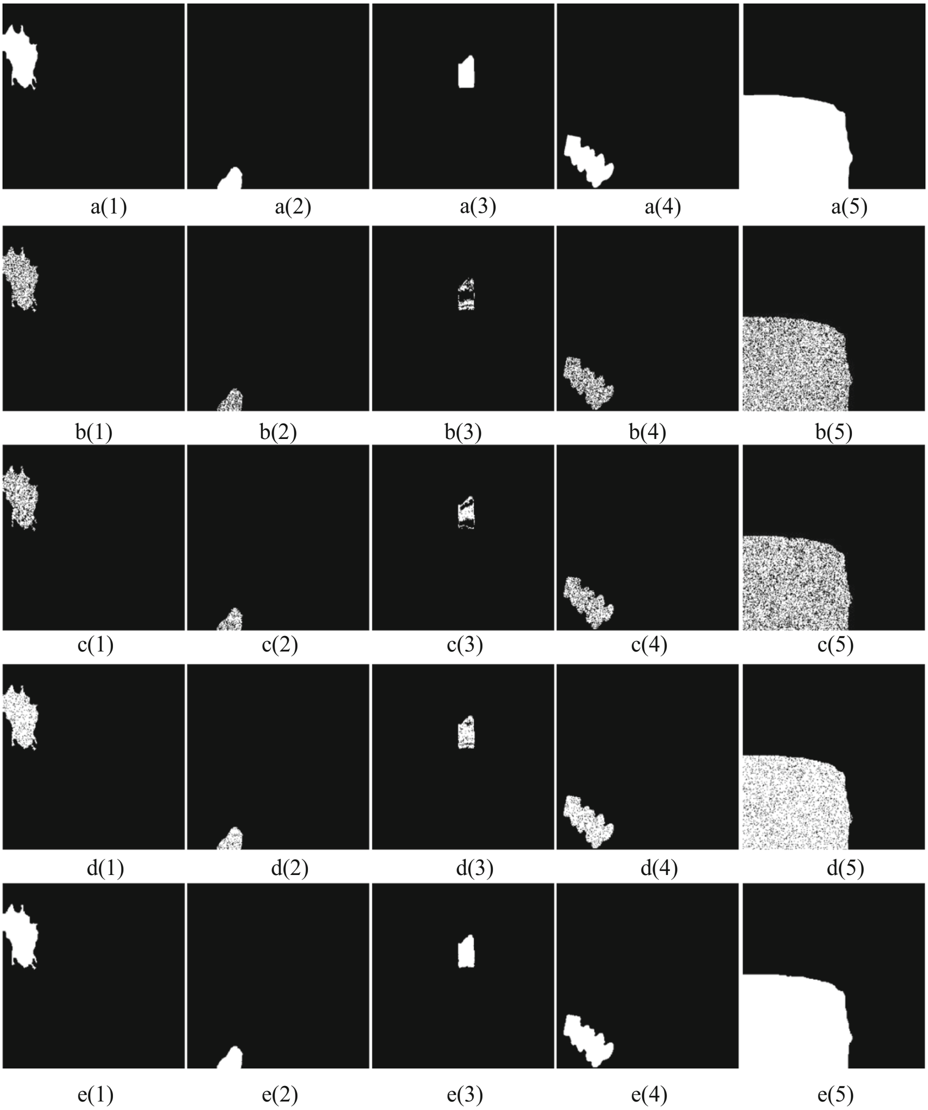

Fig. 12 Superiority of the proposed tamper detection method. **a(1)-a(5)** ground truth, **b(1)-b(5)** hashing-suspected tampered region, **c(1)-c(5)** PCBL-suspected tampered region, **d(1)-d(5)** superposition of initially detected tampered region, and **e(1)- e(5)** the final result of detected tampered region

method. Moreover, to illustrate the effectiveness of the proposed ASEC algorithm, the two-check-bit method that combines hashing and PCBL is compared with Dual-Tamper-Detection scheme in this section.

The results show that the proposed PCBL method achieves the highest precision values, as shown in Fig. 13; while the proposed Dual-Tamper-Detection method achieves the highest recall values, as shown in Fig. 14, which indicates the proposed two-check-bit-detection method performs better than any one-check-bit detection scheme and can reduce the probability of false-negative errors to a greater extent. By integrating the results of *Precision* and

Table 2 Objective measure of the results in Fig. 12

	Row\Column	Col. (1)	Col. (2)	Col. (3)	Col. (4)	Col. (5)
Precision (%)	b	97.3	96.7	96.4	96.4	99.6
	c	**97.9**	**98.1**	**98.1**	**97.9**	**99.7**
	d	97.2	97.1	97.4	96.8	99.6
	e	97.2	97.0	97.5	96.4	99.6
Recall (%)	b	63.1	55.6	36.1	58.1	64.0
	c	64.4	64.0	51.6	64.6	64.9
	d	87.3	82.2	79.4	85.5	86.7
	e	**99.5**	**99.3**	**98.5**	**99.6**	**99.9**
F1score (%)	b	76.6	70.6	52.5	72.5	77.9
	c	77.7	77.5	67.7	77.8	78.6
	d	92.0	89.0	87.5	90.8	92.7
	e	**98.3**	**98.1**	**98.0**	**98.0**	**99.7**

Recall, Fig.15 shows the results of F1 scores, and it clearly indicates that the proposed Dual-Tamper-Detection has better performance in tampering detection.

In addition to the comparisons given above, we also compare the proposed scheme with more existing methods [2–4, 7, 8, 10, 11, 32, 33]in terms of *Sensitivity*, *Specificity*, and *Accuracy* for measurement of tamper detection results, and in terms of *PSNR* for measurement of recovery results. The comparison results are shown in Table 3, where the best results are highlighted in bold and the results of our proposed Dual-Tamper-Detection scheme are highlighted in italic. It can be easily seen that the proposed scheme achieves the best *Accuracy*, which demonstrates that the proposed Dual-Tamper-Detection scheme has superior capability in tampering detection. As for the recovery performance, our work is second only to Chang's hierarchical work [7], which is good at recovering the small-scale tampering, although our work performs better than it in terms of tamper detection. The reason for this is that some recovery bits are actually lost caused by tampering in our scheme. However, the mean *PSNR* of recovered images can still achieve to 46.75dB, which is satisfied with human vision system.

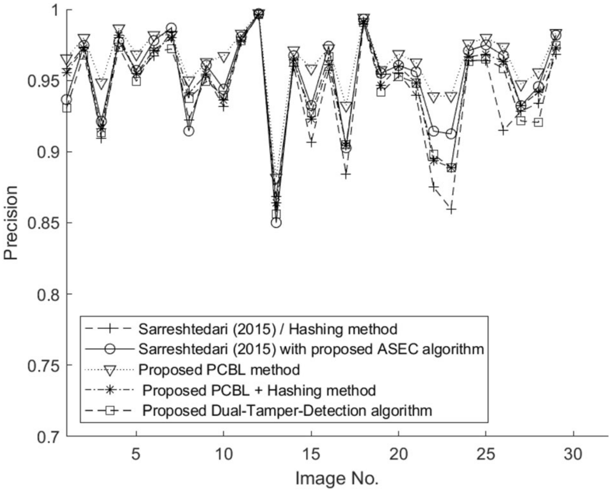

Fig. 13 Comparison with existing methods in terms of *Precision*

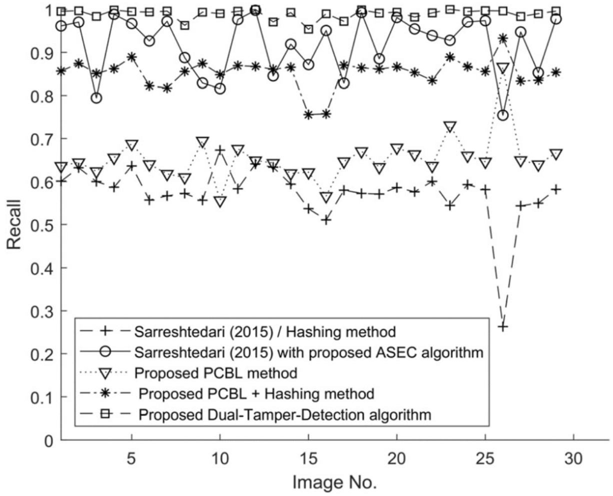

Fig. 14 Comparison with existing methods in terms of *Recall*

Overall, the proposed Dual-Tamper-Detection scheme shows good performance in improving tampered region identification accuracy while maintaining high-quality image reconstruction.

5 Conclusions

In this paper, a new scheme is proposed to improve the accuracy of tampered region identification while maintaining high-quality image reconstruction. Unlike the traditional one-check-bit detecting method [30], we propose the Dual-Tamper-Detection scheme using two check bits to reduce the probability of false-negative errors. One check bit is generated by

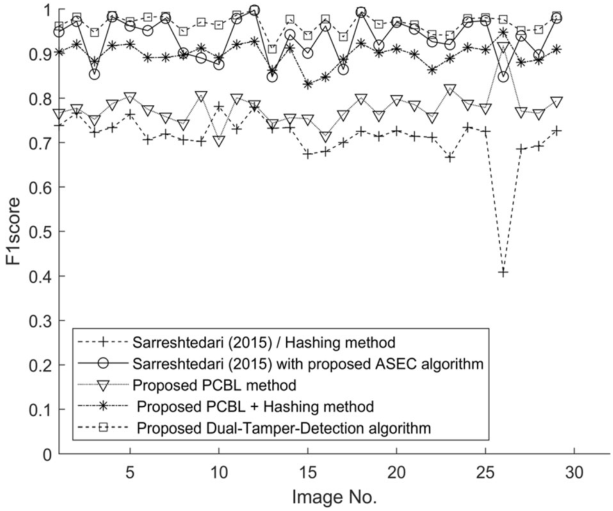

Fig. 15 Comparison with existing methods in terms of *F1 score*

Table 3 Performance comparison in terms of Sensitivity, Specificity, Accuracy and PSNR

Method	Tamper Detection Results		Recovery	
	Sensitivity (%)	Specificity (%)	Accuracy (%)	PSNR (dB)
Proposed	**98.94**	99.90	99.90	46.75
BRINT+ELM [3] (2020)	96.00	99.00	98.00	30.04
Deep Learning [2] (2019)	89.58	97.10	95.00	-
BFRE [11] (2019)	95.45	99.93	97.69	31.72
DCT [32](2017)	96.23	**99.97**	98.1	41.27
SVD [4] (2016)	96.28	**99.97**	98.13	44.63
Hierarchical [7] (2013)	95.99	99.95	97.97	**53.07**
F-transform [10] (2012)	94.31	99.91	97.11	-
FCM [8] (2009)	96.12	99.96	98.04	44.22
Chaos [33](1995)	96.04	99.96	98.01	-

*: - indicates the results were not discussed in that work.

applying the hashing algorithm to each block, and the other is produced by employing the proposed PCBL algorithm to each pixel. After that, we superpose the labeled results obtained by hashing algorithm and PCBL algorithm respectively. In order to further improve the tamper detection accuracy, we propose the ASEC algorithm to select the most adaptive structural element for different host images in an automatic way.

The experimental results show that the proposed Dual-Tamper-Detection scheme can lower the probability of false-negative errors, thus improving the accuracy of tampered region identification. The objective metrics *F1 score* and *Accuracy,* which are used to measure performance of tamper detection comprehensively, show that the proposed Dual-Tamper-Detection scheme performs better than the existing state-of-the-art works. On the other hand, the objective metric *PSNR* values are calculated to measure the recovery of tampered regions, and it is up to 46.75dB on average, which indicates the proposed method has a satisfying recovery capability. Moreover, as we know, the recovery bits are actually damaged caused by tampering, in future work, we plan to design a novel scheme to retrieve the lost recover bits, thus improving the recovery capability.

Acknowledgments This work was supported by the National Natural Science Foundation of China (Grant No. 61902448) and the Research Fund of Guangdong-Hong Kong-Macao Joint Laboratory for Intelligent Micro-Nano Optoelectronic Technology (Grant No. 2020B1212030010).

References

1. Abdulla AA (2015) "Exploiting similarities between secret and cover images for improved embedding efficiency and security in digital steganography," University of Buckingham
2. Agarwal R, Verma OP (2019) An efficient copy move forgery detection using deep learning feature extraction and matching algorithm. Multimedia Tools Applications:1–22
3. AlShehri L, Hussain M, Aboalsamh H, Wadood A (2020) Fragile watermarking for image authentication using BRINT and ELM. Multimedia Tools and Applications 79:29199–29223
4. Ansari IA, Pant M, Ahn CW (2016) SVD based fragile watermarking scheme for tamper localization and self-recovery. International Journal of Machine Learning and Cybernetics 7:1225–1239

5. Bas P, Furon T (2007) "BOWS-2, http://bows2.eclille.fr/index.php?mode=VIEW&tmpl=index1.," .
6. Cao F, An B, Wang J, Ye D, Wang H (2017) Hierarchical recovery for tampered images based on watermark self-embedding. Displays 46:52–60
7. Chang Y-F, Tai W-L (2013) A block-based watermarking scheme for image tamper detection and self-recovery. Opto-Electronics Review 21(2):182–190
8. Chen W-C, Wang M-S (2009) A fuzzy c-means clustering-based fragile watermarking scheme for image authentication. Expert Syst Appl 36(2):1300–1307
9. Dhole VS, Patil NN (2015) "Self embedding fragile watermarking for image tampering detection and image recovery using self recovery blocks," in 2015 International Conference on Computing Communication Control and Automation, pp. 752–757: IEEE
10. Di Martino F, Sessa S (2012) Fragile watermarking tamper detection with images compressed by fuzzy transform. Information Sciences 195:62–90
11. Di Martino F, Sessa S (2019) Fragile watermarking tamper detection via bilinear fuzzy relation equations. Journal of Ambient Intelligence and Humanized Computing 10:2041–2061
12. Douglas M, Bailey K, Leeney M, Curran K (2018) An overview of steganography techniques applied to the protection of biometric data. Multimedia Tools and Applications 77:17333–17373
13. Elaskily MA, Elnemr HA, Dessouky MM, Faragallah OS (2019) Two stages object recognition based copy-move forgery detection algorithm. Multimedia Tools and Applications 78:15353–15373
14. Eswaraiah R, Sreenivasa Reddy E (2015) Robust medical image watermarking technique for accurate detection of tampers inside region of interest and recovering original region of interest. IET Image Processing 9:615–625
15. Falkenstern KR, Reed AM, Holub V, Rodriguez TF (2019) "digital watermarking and data hiding with narrow-band absorption materials," ed: Google patents
16. Gao Y, Wang Ji, Shi Y-Q (2019) Dynamic multi-watermarking and detecting in DWT domain. Journal of Real-Time Image Processing 16:565–576
17. Gong D, Chen Y, Lu H, Li Z, Han Y (2018) Self-embedding Image Watermarking based on Combined Decision Using Pre-offset and Post-offset Blocks. Computers, Materials & Continua 57:243–260
18. Kaur N, Jindal N, Singh K (2020) A passive approach for the detection of splicing forgery in digital images. Multimedia Tools and Applications 79:32037–32063
19. Liu K-C (2012) Colour image watermarking for tamper proofing and pattern-based recovery. (in En), IET Image Processing 6(5):445–454
20. Mishra S, Markam K, (2018) "Analysis of active and passive mechanism for image forgery detection
21. Moghaddasi Z, Jalab HA, Noor RM (2019) Image splicing forgery detection based on low-dimensional singular value decomposition of discrete cosine transform coefficients. Neural Computing and Applications 31:7867–7877
22. Molina J, Ponomaryov V, Reyes R, Sadovnychiy S, Cruz C (2020) Watermarking framework for authentication and self-recovery of tampered colour images. IEEE Lat Am Trans 18(03):631–638
23. Nazari M, Sharif A, Mollaeefar M (2017) An improved method for digital image fragile watermarking based on chaotic maps. Multimedia Tools and Applications 76:16107–16123
24. Pal P, Jana B, Bhaumik J (2019) Robust watermarking scheme for tamper detection and authentication exploiting CA. (in En), IET Image Processing 13(12):2116–2129
25. Qin C, Ji P, Zhang X, Dong J, Wang J (2017) Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy. Signal Processing 138:280–293
26. Qin C, Qian Z, Feng G, Zhang X (2019) Special issue on real-time image watermarking and forensics in cloud computing. Journal of Real-Time Image Processing 16:559–563
27. Rajput V, Ansari IA (2020) Image tamper detection and self-recovery using multiple median watermarking. Multimedia Tools and Applications 79:35519–35535
28. Said A, Pearlman WA (1996) A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Transactions on circuits and systems for video technology 6(3):243–250
29. Sarkar D, Palit S, Som S, Dey KN (2020) Large scale image tamper detection and restoration. Multimedia Tools and Applications 79:17761–17791
30. Sarreshtedari S, Akhaee MA (2015) A source-channel coding approach to digital image protection and self-recovery. IEEE Trans Image Process 24(7):2266–2277
31. Sedighi V, Cogranne R, Fridrich J (2015) Content-adaptive steganography by minimizing statistical detectability. IEEE Transactions on Information Forensics and Security 11(2):221–234
32. Singh D, Singh SK (2017) DCT based efficient fragile watermarking scheme for image authentication and restoration. Multimed Tools Appl 76(1):953–977
33. Steve W (1995) Information authentication for a slippery new age. Dr Dobbs Journal:18–26
34. Su Z, Yao L, Mei J, Zhou L, Li W (2020) "Learning to hash for personalized image authentication," IEEE Transactions on Circuits and Systems for Video Technology

35. Tagliasacchi M, Valenzise G, Tubaro S (2009) Hash-based identification of sparse image tampering. IEEE Trans Image Process 18(11):2491–2504
36. Yang Q, Yu D, Zhang Z, Yao Y, Chen L, (2020) "Spatiotemporal trident networks: detection and localization of object removal tampering in video passive forensics," IEEE Transactions on Circuits and Systems for Video Technology, 1
37. Yao H, Wei H, Qin C, Tang Z (2020) A real-time reversible image authentication method using uniform embedding strategy. Journal of Real-Time Image Processing 17:41–54

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.