

Contents lists available at ScienceDirect



# Journal of King Saud University – Computer and Information Sciences

journal homepage: [www.sciencedirect.com](http://www.sciencedirect.com)

## AuSR1: Authentication and self-recovery using a new image inpainting technique with LSB shifting in fragile image watermarking

Afrig Aminuddin<sup>a</sup>, Ferda Ernawan<sup>b,\*</sup><sup>a</sup> Department of Informatic, Faculty of Computer Science, Universitas Amikom, Yogyakarta, Indonesia<sup>b</sup> Department of Computer Graphic and Multimedia, Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, Kuantan, Malaysia

## ARTICLE INFO

## Article history:

Received 18 October 2021

Revised 2 February 2022

Accepted 9 February 2022

Available online xxxx

## Keywords:

Blind fragile watermarking

Self-embedding

Image authentication

Self-recovery

Image inpainting

## ABSTRACT

With the rapid development of multimedia technology, editing and manipulating digital images have become more accessible than ever. This paper proposed color image authentication based on blind fragile image watermarking for tamper detection and self-recovery named AuSR1. The AuSR1 divides each channel of the cover image into non-overlapping blocks with the size of  $2 \times 2$  pixels. The authentication data is embedded into the original block location, while the recovery data is embedded into the distant location from the original location based on the block mapping algorithm. The watermark data is then embedded into the 2 LSB to achieve high quality of the recovered image under tampering attacks. In addition, the permutation algorithm is applied to ensure the security of the watermark data. The AuSR1 utilizes a three-layer authentication algorithm to achieve a high detection rate. The experimental results show that the scheme produced a PSNR value of 45.57 dB and an SSIM value of 0.9972 of the watermarked images. Furthermore, the AuSR1 detected the tampered area of the images with a high precision value of 0.9943. In addition, the recovered image achieved a PSNR value of 27.64 dB and an SSIM value of 0.9339 on a 50% tampering rate.

© 2022 The Author(s). Published by Elsevier B.V. on behalf of King Saud University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Introduction

Nowadays, the transmission of multimedia data has grown due to the development of internet technology. Multimedia data such as images are easily distributed over internet technology. Digital images may be vulnerable by unauthorized persons to modify or edit the data during transmission (Ray and Roy, 2020). Unauthorized persons can utilize image-editing software to tamper and alter the original image. For medical images, if any change or the small amount of modification on the medical images could affect the judgment of the medical doctor (Hemida et al., 2019). In another case, modifying image information for the crime evidence may lead to false judgment in court. Therefore, image authentication is required to ensure the authenticity and integrity of the

images. Researchers have proposed various authentication schemes to address this issue (Hemida et al., 2019; Belferdi et al., 2019; Hong et al., 2021; Huang et al., 2019; Su et al., 2021; Jafari Barani et al., 2019; Gul and Ozturk, 2019; Gul and Ozturk, 2021; Gul and Ozturk, 2020; Prasad and Pal, 2020). The authentication schemes work by detecting and localizing the tampered area of the images. The image authentication itself was classified into active and passive authentication, which differs by preliminary data on active authentication. Passive authentication relied on tamper detection on the image's features and properties (Jafari Barani et al., 2019). A scheme may work on certain types of attacks while it did not work on other types. In contrast, active authentication will work on all types of attacks as long as the scheme was adequately designed. Active authentication was further classified into two categories which are based on the hash function and digital watermarking techniques (Ouyang et al., 2020). On one hand, active authentication based on the hash function takes the image input to produce a hash value for authentication purposes. The hash value itself is then stored in a secure database, which is agreed upon by the sender and the recipient. The image data is then sent to the recipient using a public communication channel with a possible attack. On the other hand, active authentication based on digital image watermarking techniques works by

\* Corresponding author at: Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, Pekan, 26600 Kuantan, Malaysia.

E-mail address: [ferda@ump.edu.my](mailto:ferda@ump.edu.my) (F. Ernawan).

Peer review under responsibility of King Saud University.



Production and hosting by Elsevier

<https://doi.org/10.1016/j.jksuci.2022.02.009>

1319-1578/© 2022 The Author(s). Published by Elsevier B.V. on behalf of King Saud University.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

embedding the authentication data into the image itself instead of the database.

The digital image watermarking was classified into three groups: robust, semi-fragile and fragile watermarking. Robust image watermarking is primarily used for copyright protection, which relies on the integrity of the watermark logo. The watermark itself should survive under various attacks applied to the images, such as compression, filtering, cropping, rotation, etc. (Zhang and Wei, 2019; Kumar and Singh, 2021; Kang et al., 2020). Semi-fragile and fragile watermarking schemes can be easily destroyed so that the algorithm could locate the tampered area of the images. Semi-fragile and fragile watermarking were best suited for image authentication. Based on the ability, fragile watermarking can be classified into two categories (Hemida et al., 2019). The first category is the watermarking scheme which has the ability to authenticate and identify the tampered region or tampered location in the images. The second category is the watermarking scheme which has the ability to perform authentication and self-recovery. The watermarking process divides the cover image into non-overlapping blocks. Small block size on the non-overlapping blocks achieves high precision and accuracy in the tamper detection. If a pixel of each image block has been modified by attackers, then the remaining untampered pixels are marked as tampered, these pixels are considered as false positive detection. A small block image provides less false positive detection than a large image block. The false positive detection contributes to reduce the precision and accuracy of the tamper detection. In addition, a large block size also can provide pixelated effects on the recovered image. Therefore, most of fragile watermarking schemes used a small block size for image authentication and self-recovery.

The digital image watermarking embeds watermark data into the digital image, it can be performed in the spatial and frequency domains. In the spatial domain, the watermark data is embedded into the least significant bit (LSB) of the images. Embedding watermark into LSB performs less computation than frequency domain. In addition, embedding the watermark into LSB achieves high imperceptibility, the human eye cannot distinguish a modification in the LSB. For instance, the embedding watermark into LSB produced watermarked images with an average PSNR value of 51 dB. In comparison, an embedding watermark into two LSB produced 44 dB (Dadkhah et al., 2014; Fan and Wang, 2018; Tai and Liao, 2018; Molina-Garcia et al., 2020), while three LSB produced 37 dB (Tong et al., 2013; Singh and Singh, 2016). Even though embedding watermark in one LSB provides the high imperceptibility, it can only store a small amount of watermark data. For instance, embedding watermark into one LSB in a block size of  $2 \times 2$  pixels can store four bits. Typically, the recovery data require eight bits to represent the average value of each block image. The two LSB in a block size of  $2 \times 2$  pixels can provide eight bits for embedding watermark data, which consists of two authentication bits and six recovery bits. Therefore, embedding watermark into two LSB can provide self-recovery and high accuracy of tamper detection. In the frequency domain, the cover image is transformed into the frequency coefficient by using transform domains such as DCT, DWT, IWT, and SVD (Cox et al., 1997; Ouahabi, 2012; Hsu and Wu, 1998; Wang et al., 2002; Zermi et al., 2021; Soni et al., 2020; Hsu and Wu, 1999). Embedding the watermark data into the transform domain will preserve the watermark data under various attacks applied to the image. However, it has a limited capacity than embedding watermark in the spatial domain. The tamper localization accuracy is needed to find the tampered area for recovering the image.

The watermark data consists of authentication and recovery data. The authentication data is utilized to detect any modification on the image and the recovery data is used to recover the altered object in the image. The recovery data of each block is embedded

into the different block location based on block map. Thus, if a block is altered, then it will be replaced by using the recovery data. The recovery data may also be tampered by the attacker which called as tampered coincidence. This problem can be solved by using the image inpainting techniques (Molina-Garcia et al., 2020). The image inpainting techniques have been widely used to recover the corrupted paintings or photography and removal of undesirable objects (Qureshi et al., 2017). The image inpainting technique aims to fill in the corrupted images with realistic contents (Wang et al., 2021). The image inpainting techniques are commonly derived from geometric partial differential equations (PDEs), texture synthesis, deep generative model-based approach (Yu et al., 2018) and coherence among neighboring pixels (Bugeau et al., 2010). Image inpainting based on deep learning technique has some advantages of learning information from a large amount of data (Ouahabi and Taleb-Ahmed, 2021; Mimouna, 2020; Quan et al., May 2021). This technique is also widely used for image denoising (Ouahabi, 2013), image registration (Pluim and Fitzpatrick, 2003), ultrasound imaging (Ouahabi and Taleb-Ahmed, 2021), and object recognition (Khaldi et al., 2021; Adjabi et al., Jan. 2021; El Morabit et al., 2021; Adjabi et al., 2020).

This paper proposed a new image inpainting technique in fragile image watermarking. The proposed blind fragile image watermarking scheme has the ability to detect tamper location and recover self-image content. The proposed Authentication and Self-Recovery (AuSR1) scheme divides each channel of the cover image into non-overlapping blocks with the size of  $2 \times 2$  pixels. The utilization of the small block size is to achieve precise tamper detection and better recovery. The watermark data obtained from the cover image consist of authentication data and recovery data. The authentication data is embedded into the original block location, while the recovery data is embedded into the different locations by considering the block mapping algorithm. In addition, the block mapping is generated by using the LSB permutation algorithm with a secret key. The watermark data is then embedded into the 2 LSB to achieve high quality of the recovered image under various tampering attacks. In the tamper detection stage, the AuSR1 utilizes a three-layer authentication algorithm to achieve an optimal detection rate. The first layer identifies the tampered region by comparing extracted two authentication bits from the tampered image. The second layer authentication implements a convolutional tamper detection algorithm. The third layer authentication examines the result of second layer authentication with the number of RGB channels. Finally, the tampered image is recovered using the recovery algorithms. The AuSR1 scheme solves the tamper coincidence problem using the proposed image inpainting algorithm.

The rest of this paper is organized as follows. Section 2 presents state-of-the-art methods. Section 3 describes the proposed AuSR1 scheme, tamper detection, inpainting scheme, and recovery which are discussed in sub-sections. Section 4 presents the experimental results of the proposed AuSR1 scheme and the performance comparison with the state-of-the-art techniques. Finally, Section 5 concludes the proposed AuSR1.

## 2. Related works

Tong et al. (Tong et al., 2013) proposed a fragile watermarking scheme by utilizing a chaotic map permutation for the embedding process. This map ensures the random location of the recovery bits with the predetermined control parameters. The scheme embedded 12 bits into 3 LSB of  $2 \times 2$  non-overlapping blocks. The scheme produced an average PSNR value of 40 dB of the watermarked image. However, the scheme did not address the issue of tamper

coincidence. As a result, the recovered image did contain the trace of the tampered image due to the tamper coincidence. The average PSNR value of the recovered image was about 31 dB.

Dadkhah et al. (Dadkhah et al., 2014) proposed an active watermarking scheme based on Singular Value Decomposition (SVD) for tamper detection. The SVD provided a compact and sensitive detection to any content modification applied to the image. The authentication bits are embedded into non-overlapping blocks of  $4 \times 4$  pixels. At the same time, the recovery bits are embedded into sub-blocks of  $2 \times 2$  pixels based on the average value of those four pixels. The scheme utilized pseudorandom code for block mapping. The block mapping itself considers predetermined conditions to ensure the most distance location for mapping blocks. The conditions were selected based on the primary location of the original block. Thus, the located block on the upper part of the image will be mapped to the located block on the lower part and vice versa. The conditions also prevented any blocks from being mapped twice in pairs to prevent the tamper coincidence. However, when the large tamper occurred in the image area, the tamper coincidence problem was inevitable. The results show that the recovered image had a tamper coincidence issue when the watermarked image has tampered with a high tampering rate.

Singh et al. (Singh and Singh, 2016) proposed a self-embedding fragile watermarking scheme based on Discrete Cosine Transform (DCT). The scheme divided the image into non-overlapping blocks of  $2 \times 2$  pixels. The DCT was utilized to compute ten recovery bits of each block by considering the first and the second largest quantization matrix value. The scheme embedded ten recovery bits and two authentication bits into 3 LSB of each block. The mapping block utilized a random number for a one-to-one mapping sequence. Unlike the scheme proposed by Dadkhah et al. (Dadkhah et al., 2014), Singh et al. (Singh and Singh, 2016) did not consider the distance of the mapped block to the original block location. The scheme has a high probability of tamper coincidence problem. In addition, the embedding watermark in the 3 LSB produced low image quality with an average PSNR value of about 37 dB.

Fan et al. (Fan and Wang, 2018) proposed a fragile watermarking scheme based on Set Partitioning in Hierarchical Tree (SPIHT). The cover image is divided into four non-overlapping blocks. Each block was then divided into non-overlapping sub-blocks of  $2 \times 2$  pixels. Furthermore, each sub-block was compressed based on the SPIHT algorithm with the compression rate of 0.75 bpp (bit per pixel) for recovery bit's information. The recovery bits themselves were embedded into the adjacent block on the available four blocks. If one of the blocks is tampered with, the adjacent block recovered the tampered area of the image. Furthermore, the scheme also included 0.75 bpp parity bits and 0.5 bpp check bits for authentication purposes. In total, the watermark data 2.0 bpp was embedded into 2 LSB in the cover image. The scheme produced a watermarked image quality with an average PSNR value of 44 dB. However, when the tampering area is larger than a single block of the image, it may cause the tamper coincidence problem. Thus, the scheme has the potential to be improved especially in order to solve the tamper coincidence problem.

Tai et al. (Tai and Liao, 2018) proposed a fragile watermarking scheme that supports self-embedding and self-recovery. The scheme used a chaotic map to randomize the recovery bit location. First, the image is divided into non-overlapping blocks of  $4 \times 4$  pixels. Each block was embedded with 32-bits data, it consists of 4-bits authentication data and 28-bits recovery data. The watermark recovery was performed by using Integer Haar Wavelet. Furthermore, the scheme employed hierarchical tamper detection to achieve a high detection rate from tampering attacks. The scheme has three-layer detection algorithms. However, the scheme failed to achieve a 100% detection rate. The use of a  $4 \times 4$  block size could

lead to achieve a high false-positive rate. In tamper recovery, the scheme has implemented self-recovery with inpainting support. However, the inpainting algorithm only considers eight neighboring blocks that may not be available in large tampering areas due to the tamper coincidence problem.

Molina-Garcia et al. (Molina-Garcia et al., 2020) proposed fragile watermarking that supports image inpainting for self-recovery. The recovery data was generated from the luminance component of the cover image using a halftoning technique based on the error diffusion method which corresponds to 1.75 bpp. The authentication data was generated based on the non-overlapped blocks of  $4 \times 4$  pixels. Each block provided 4-bits authentication data, which correspond to 0.25 bpp. The total of embedded watermark data into the 2 LSB is about 2 bpp. The scheme produced a watermarked image with an average PSNR value of 44.63 dB. The scheme used a hierarchical tamper detection algorithm to authenticate the tampered image. This process has the potential to produce up to a 100% detection rate. However, the block size of  $4 \times 4$  pixels may lead to a high false-positive rate for generating the authentication data. If one of those  $4 \times 4$  pixels has been tampered with, the whole block was treated as tampered. Furthermore, the scheme has implemented an image inpainting algorithm to solve the tamper coincidence problem. However, when the tamper has occurred in the large regions, the scheme struggles to interpolate the tamper coincidence. The ineffectiveness of the scheme is the search area only on the neighboring pixels. The scheme showed a PSNR value of about 19.20 dB for the recovered image under 80% tampering rate. Furthermore, the image inpainting algorithm left some artifacts on the recovered image, leading to a low SSIM value on a high tampering rate.

Sreenivas et al. (Sreenivas and Kamakshi Prasad, 2016) proposed an improved self-recovery approach based on the block encoding method. The scheme divides the cover image into non-overlapping blocks of  $2 \times 2$  pixels. The authentication bit was generated using the chaotic maps, while the recovery bits were generated using block encoding with seven distinct schemes. The recovery bit was then embedded into a random block based on the chaotic maps. This technique presented a great performance in tamper detection. However, the quality of the recovered image still can be further improved. Cao et al. (Cao et al., Jan. 2017) proposed a self-embedding watermarking scheme with hierarchical recovery. The recovery bits are generated based on the selected embedding parameters, while the authentication bits are generated by using a hash function from the recovery bits. The watermark data is then permuted with a secret key to provide additional security. In the embedding process, the MSB of each pixel is kept unchanged, while the LSB is replaced with the watermark data. The scheme performed embedding watermark with various block sizes between 3 and 7, the scheme produced PSNR value of the watermarked image between 25.81 dB and 51.4 dB. The scheme has a good tamper detection due to the used of hash function for authentication bits. Haghighi et al. (Bolourian Haghighi et al., 2018) presented a dual watermarking scheme using the lifting wavelet transform and the half toning technique to generate the recovery bits. The cover image itself was divided into non-overlapping blocks of  $2 \times 2$  pixels. The authentication bits were obtained from the image digest. The watermark data was embedded into the cover image using the LSB rounding technique. Arnold Cat Map is used to determine the mapping blocks and to provide additional security. The scheme is able to authenticate and recover tampered images with large tampered regions. Qin et al. (Qin et al., 2016) presented a self-embedding based on reference-data interleaving and adaptive selection of embedding. The scheme utilized two types of embedding modes: overlapping and non-overlapping embedding. The scheme also implemented adaptive flexible number of MSB and LSB layers. MSB bits are

interleaved to generate reference bits, then are embedded into the LSB. However, the scheme has not been tested against a large tampering rate. The scheme has not investigated the tamper coincidence occurred on the large tampering rate.

It can be summarized that the existing watermark schemes for authentication and self-recovery still have not achieved a satisfactory level in terms of watermarked image quality, the precision of the tamper detection, and the quality of self-recovery under various altered image. The existing watermarking scheme achieved average PSNR value of about 44 dB for the watermarked image quality. The existing schemes still have high amount of tampered coincidence in the recovery process. A large number of tamper coincidences in the recovery process may decrease the quality of the recovered image. The existing schemes also have an adequate amount of false-negative detection to determine the tampered region. It can significantly contribute to the precision of the tamper detection. The proposed AuSR1 scheme has several contributions in image authentication and self-recovery. The main contribution of the proposed AuSR1 can be summarized as follows:

1. The proposed AuSR1 utilizes LSB shifting algorithm that can decrease the pixel intensity variation between the cover and watermarked images. The AuSR1 scheme improves 2% of the watermarked imperceptibility compared to the existing schemes.
2. The proposed AuSR1 scheme implements three-layers authentication. The first layer can achieve recall value of 0.75. The second layer further improves the recall value up to 0.99. Finally, the third layer authentication complements the tamper detection scheme in the RGB channels to achieve recall value equal to 1. The AuSR1 improves the precision by 3.8% compared to the existing methods towards regular attack.
3. The proposed AuSR1 employs a new image inpainting technique to solve the tamper coincidence problem. The AuSR1 searches the non-tamper coincidence pixel in a spiral outward direction until it has sufficient information to interpolate the tamper coincidence block. This technique improves 5.3% of the recovered image quality compared to the existing schemes.

### 3. Proposed method

The AuSR1 is divided into four stages: watermark embedding stages, pre-detection and pre-recovery stages, tamper detection stages, and tamper recovery stages as shown in Fig. 1.

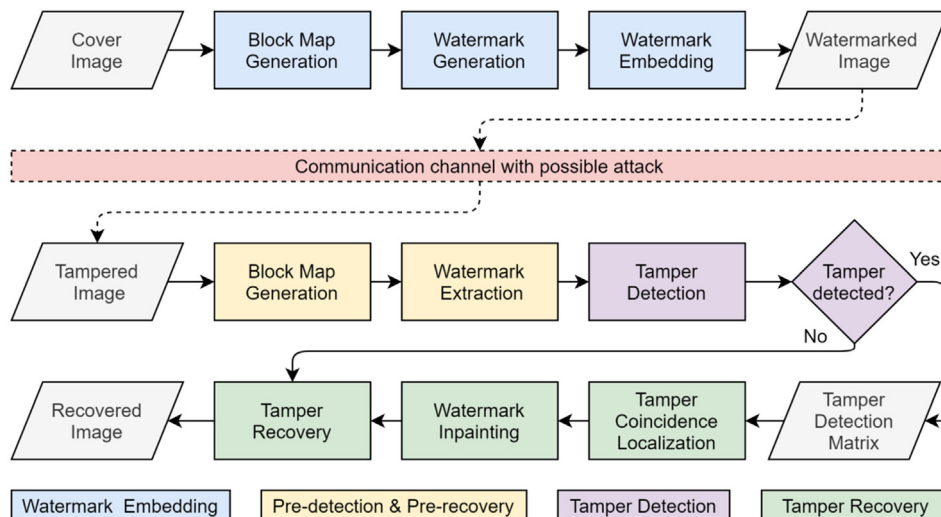


Fig. 1. The proposed AuSR1 scheme.

The watermark embedding process consists of three sub-stages denotes in blue: block map generation, watermark generation, and watermark embedding. The AuSR1 divides the cover image into non-overlapping blocks of  $2 \times 2$  pixels. The width and height of the cover image should be multiple of two. The last pixel of non-coverage blocks should not be performed with the embedding the watermark. The scheme embeds the watermark only in the covered block of  $2 \times 2$  pixels. The watermarked image may undergo possible attacks during the delivery in the communication channel. Once the recipient receives the watermarked image, the image will be fed into the pre-detection and pre-recovery stage denotes in yellow. Essentially this stage will produce matrices that are required for tamper detection and tamper recovery. This stage consists of block map generation and watermark extraction. The block map itself must be identical to the block map in the embedding stage. Therefore, the same key is utilized to generate a block map in both stages. In the watermark extraction process, it will extract two watermark data. The first watermark data is generated from 6 MSB and the second watermark data is extracted from 2 LSB.

The tamper detection stages consist of a three-layer authentication algorithm to achieve an optimal detection rate denotes in purple. The first layer identifies the tampered region by comparing two authentication matrices from previous watermark extraction. The second layer authentication implements a novel convolutional tamper detection algorithm. The third layer examines the result of second layer authentication with the number of RGB channels. The output of this stage is the tamper detection data which can be used for tamper recovery. The tamper recovery stage which is denoted in green color, it consists of three sub-stages: tamper coincidence localization, proposed watermark inpainting, and tamper recovery stages. The tamper coincidence localization generates a map that shows the location of tamper coincidence on the image. The watermark inpainting stage will solve the tamper coincidence problem, the detailed explanation is given in the watermark inpainting subsection. After all tamper coincidences are fixed, the tamper recovery stage will recover the image information.

#### 3.1. Block map generation

Block map is an integral part of the image authentication algorithm that supports tamper recovery. The block map is mainly used for mapping the recovery bits of each block to another location within the image based on the predefined location on the block

map. In addition, this scheme also utilizes the block map information as a key for various steps within the watermark embedding unit and the watermark extraction unit. The block map generation unit comprises four steps:

Step 1: Divide each channel of the cover image into a non-overlapping block of  $2 \times 2$  pixels.

Step 2: Create the key based on the maximum prime number with the following equation:

$$\begin{aligned} key_R &= \max(\text{prime}(K \times 1)) \\ key_G &= \max(\text{prime}(K \times 2)) \\ key_B &= \max(\text{prime}(K \times 3)) \end{aligned} \quad (1)$$

where  $key_R, key_G, key_B$  represent the key for each RGB channel,  $K$  represents the number of blocks for each channel,  $M, N$  denotes the width and height of the image.

Step 3: Arrange the index of  $i$ -th blocks of the cover image into a vector  $map_{x(i)} = \{1, 2, \dots, K\}$ . Each block index is started from 1 up to the number of blocks  $K$ .

Step 4: Permute the vector based on the chaotic map with the following equation:

$$map_{p(i)} = \text{permute}(\text{key}, map_{x(i)}) \quad (2)$$

where  $map_p$  represents the permuted block map,  $key$  represents the key for each channel.

The scheme produces three-block maps ( $map_{p(i)}$ ) which corresponds to each RGB channel. The block map  $map_p$  consists of the block index and mapped recovery location. The index refers to the location of the image block, and the value refers to the location for embedding the recovery bits.

### 3.2. Watermark generation

The watermark generation is a process to generate the watermark from the cover image. This watermark consists of the authentication bits and the recovery bits. The authentication bit is used for detecting the tampered area, while the recovery bits are used for recovering tampered area. The watermark generation comprises five steps:

Step 1: Divide each channel of the cover image into non-overlapping blocks of  $2 \times 2$  pixels. The small block size ensures precise tamper detection. Conversely, a larger block size will increase the false positive value in tamper detection.

Step 2: Calculate an average value of the selected image block. Six MSB of the average value is stored as the recovery bits  $r_i$  of the selected block.

Step 3: Retrieve the mapped recovery location of the selected block from the block map and convert the location into binary values as defined by:

$$map_{bin} = \text{dec2bin}(map_p) \quad (3)$$

where  $i$  is the index of  $i$ -th blocks of the cover image and  $map_p$  represents the mapped recovery location.

Step 4: Calculate two authentication bits  $a_1$  and  $a_2$  as defined by:

$$b = \text{dec2bin}(\text{mod}(m, 4)) \quad (4)$$

where  $m$  represents the bit number with the value of '1' from  $map_{bin}$  and six MSB of each pixel,  $a_1$  is the first LSB of  $b$  and  $a_2$  is the second LSB of  $b$ .

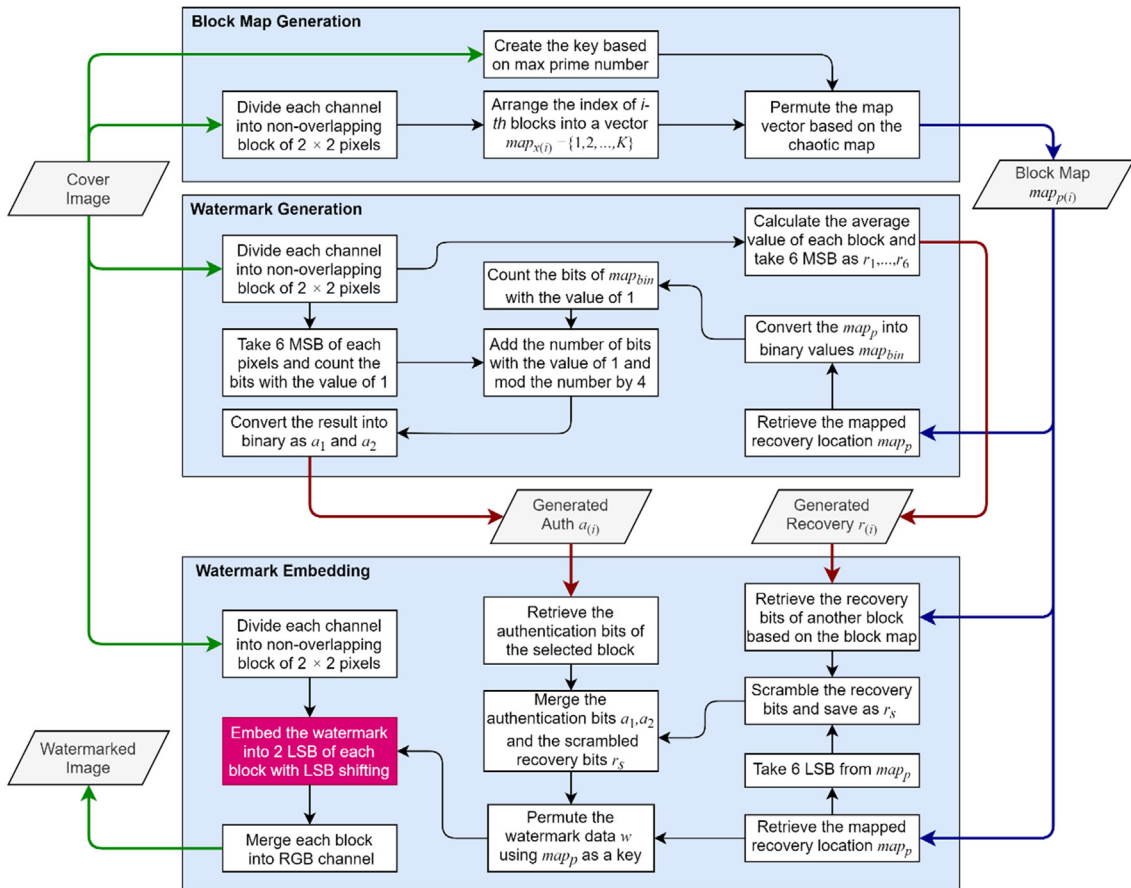


Fig. 2. The AuSR1 watermark embedding diagram.

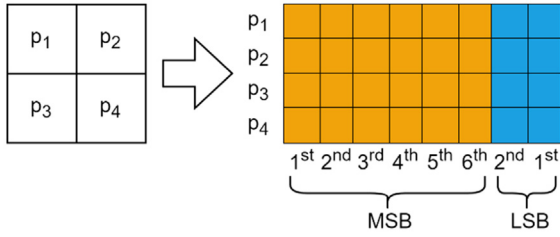


Fig. 3. The anatomy of each block of the image.

Step 5: Repeat Step 2 to Step 4 for all image blocks to obtain authentication data and the recovery data.

The watermark data  $w$  consists of two authentication bits  $a_1, a_2$  and six recovery bits  $r_1, r_2, r_3, r_4, r_5, r_6$  for each selected block. The watermark data will be embedded into the two LSB of the cover image.

### 3.3. Watermark embedding

Each channel of the image is divided into non-overlapping blocks with the size of  $2 \times 2$  pixels. These blocks are then feed into three stages: block map generation, watermark generation, and watermark embedding. A block map is generated to store the recovery bit information of a particular block into another block location of the image. Aside from that, the block map can also be utilized as the embedding key on a specific block location. The next step is to generate the watermark to be embedded. The watermark embedding diagram is visualized in Fig. 2.

The authentication data will be embedded into its block location, while the recovery data will be embedded into another location. Each image block has four pixels as illustrated in Fig. 3. Each pixel is represented in 8-bits information. The first six bits (6 MSB) are utilized to compute the authentication and recovery while the last two bits (2 LSB) are used for the embedding location.

The authentication bits are obtained from the parity of six MSB on each pixel and 32 bits value of the embedding key. The recovery data is obtained from 6 MSB of the average value of each block. In order to provide additional security, the watermark data are permuted before embedding the watermark. The watermark data will be obfuscated to the attacker. There are eight bits of watermark data to be embedded into the two LSB for each block of  $2 \times 2$  pixels. The scheme implements an LSB shifting algorithm to reduce the possibility of variance pixels of the watermarked image against the original pixel. LSB shifting algorithm can maintain the quality of the watermarked image. Finally, the watermark data is embedded into two LSB as visualized in Fig. 3.

This section explains the embedding process of the authentication data and recovery data into the cover image. The authentication bits will be embedded into the block location of the cover image, while the recovery bits will be embedded into the

corresponding block location based on the block map. The watermark embedding procedures comprises into six steps:

Step 1: Divide each channel of the cover image into non-overlapping blocks of  $2 \times 2$  pixels.

Step 2: Retrieve the authentication bits of the selected block  $a_1$  and  $a_2$ .

Step 3: Retrieve the recovery bits  $r_1, r_2, r_3, r_4, r_5, r_6$  of another block based on the block map.

Step 4: Retrieve the mapped recovery location  $map_p$  from the block map. Scramble the  $r_s$  value based on the following equation:

$$r_s = r \oplus \text{bitget}(map_p, 6) \quad (5)$$

where  $r$  represents six recovery bits,  $map_p$  is the mapped recovery location as a key,  $r_s$  denotes the scrambled recovery bits.

Step 5: Merge the authentication bits  $a_1, a_2$  and the scrambled recovery bits  $r_s$  and it is saved as  $w$ . Permute the watermark data  $w$  using  $map_{pr}$  as a key. This permutation will protect the watermark data from any intentional attack that focus on authentication bits. The authentication bit location becomes difficult to be detected by an unauthorized person.

Step 6: Embed the watermark  $w$  based on the permuted locations into the two LSB of the selected block as shown in Fig. 4.

The embedding watermark is performed by implementing LSB shifting algorithm as defined in Algorithm 1. The LSB shifting is applied to maintain the quality of the watermarked image. The algorithm works by reducing the probability pixel value obtain after embedding watermark data. The watermarked image pixel will be closer to the original cover image pixel.

#### Algorithm 1: LSB shifting algorithm

Input:  $p; w$

```

1         for  $i = 1$  to 4
2              $p_w(i) = p(i)$ 
3              $w_d(i) = \text{bin2dec}(w(i))$ 
4              $j = 0$ 
5              $Sign = 1$ 
6             while (bitand( $p(i) + j, 3$ )  $\sim = w_d(i)$ )
7                 if ( $Sign == 1$ )
8                      $j = \text{abs}(j) + 1$ 
9                 else
10                     $j = \text{abs}(j) - 1$ 
11                end if
12                 $Temp = p(i) + j$ 
13                if ( $0 \leq Temp \ \&\& \ Temp \leq 255$ )
14                     $p_w(i) = Temp$ 
15                end if
16                 $Sign = \sim Sign$ 
17            end while
18        end for

```

Output:  $p_w$

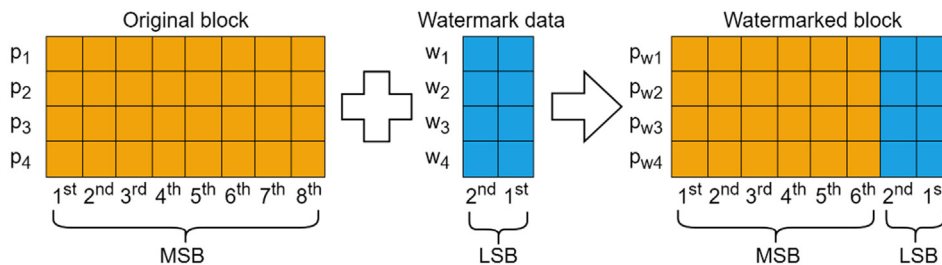


Fig. 4. Watermark embedding locations.

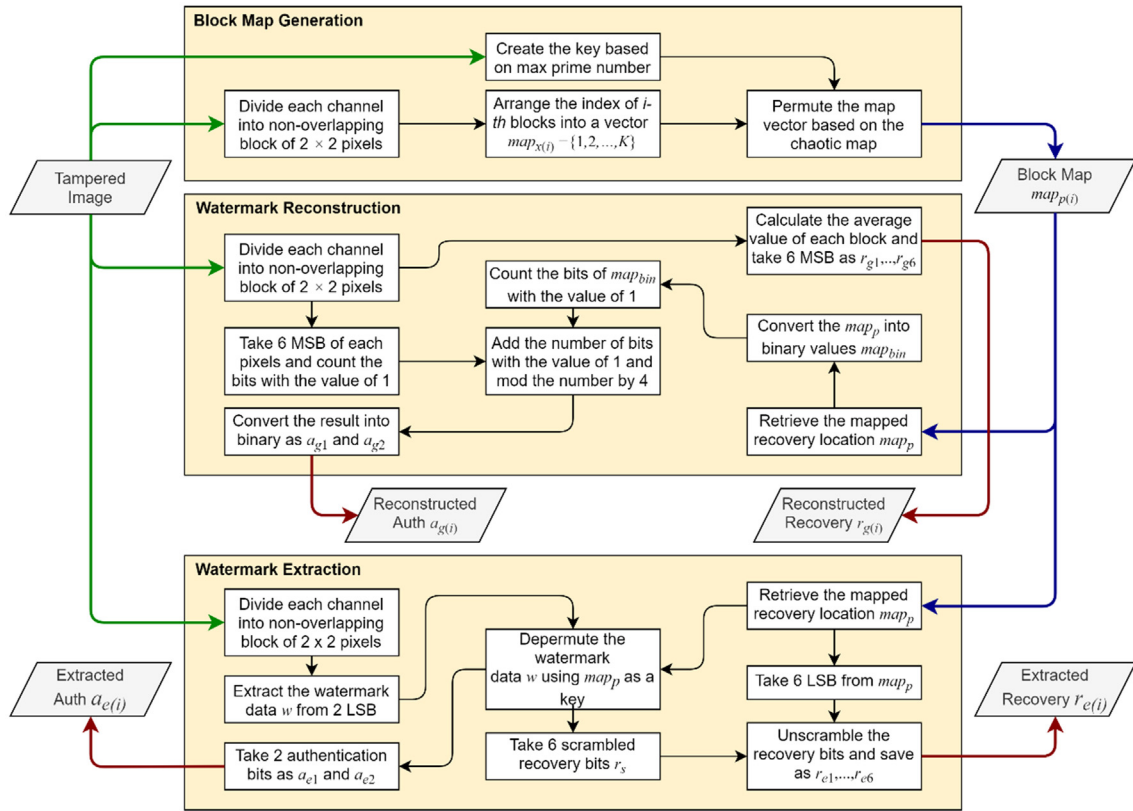


Fig. 5. The AuSR1 watermark extraction diagram.

For example, the original pixel has the binary form of 01,000,100 and the watermark bits have the binary form of 11. Traditionally, if the watermark embedding method replaces two LSB of the original pixel with the watermark data, the watermarked pixels become 01,000,111 in the binary form. The proposed LSB shifting algorithm can perform better than the traditional method. The LSB shifting can obtain the watermarked pixels of 01,000,011 in the binary form. The watermarked pixels obtained from LSB shifting are closer to the original pixel. This technique will also alter the sixth MSB of the original pixel. Thus, the sixth MSB does not consider bit authentication. The proposed embedding watermark scheme can maintain the watermarked image quality. In addition, the proposed scheme can detect tamper localization and recover the tampered image under various attacks in the communication channel.

### 3.4. Watermark extraction

Tamper detection and tamper recovery require three essential data: the block map, the extracted watermark from six MSB of the tampered image, and the extracted watermark from two LSB of the tampered image. The block diagram of the block map generation and the watermark extraction is visualized in Fig. 5.

The block map has an important role in the watermark embedding process. The block map is used to determine the locations of the recovery bits. The block map contains the block index and recovery location. In addition, the block map value is also used as the key for the authentication bits, recovery bits, and LSB permutation algorithm. Therefore, the block map must be the same for embedding watermark, tamper detection and recovery stage. Furthermore, the same image size is required in the block map generation stage. The watermark extraction stage comprises 11 steps as follows:

Step 1: Divide each channel of the tampered image into non-overlapping blocks of  $2 \times 2$  pixels.

Step 2: Calculate the average pixels for each selected block of the watermarked image. Take six MSB of the average pixels for each selected block. The six MSB are stored as the recovery bits of its selected block  $r_{g1}, r_{g2}, r_{g3}, r_{g4}, r_{g5}, r_{g6}$ .

Step 3: Retrieve recovery location  $map_{pr}$  from the block map and convert it into the binary values as defined by:

$$map_{bin} = dec2bin(map_p) \quad (6)$$

where  $i$  is the index of  $i$ -th blocks of the tampered image and  $map_p$  represents the mapped recovery location.

Step 4: Calculate two authentication bits  $a_{g1}$  and  $a_{g2}$  as defined by:

$$b = dec2bin(mod(m, 4)) \quad (7)$$

where  $m$  represents the bit number with the value of '1' from  $map_{bin}$  and six MSB of each pixel,  $a_{g1}$  is the first LSB of  $b$  and  $a_{g2}$  is the second LSB of  $b$ .

Step 5: Repeat Step 2 to Step 4 for all blocks to obtain authentication data  $a_{g(i)}$  and the recovery data  $r_{g(i)}$ .

Step 6: Extract the watermark data  $w$  from two LSB for each selected block of the watermarked image.

Step 7: Inverse permute the watermark data  $w$  obtained from two LSB using  $map_{pr}$  as a key.

Step 8: Retrieve two authentication bits  $a_{e1}$  and  $a_{e2}$  from the inverse permuted the watermark data  $w$ .

Step 9: Retrieve six recovery bits  $r_{s1}, r_{s2}, r_{s3}, r_{s4}, r_{s5}, r_{s6}$  from the watermark data  $w$  and unscramble the six recovery bits as defined by:

$$r_e = r_s \oplus msb(map_p, 6) \quad (8)$$

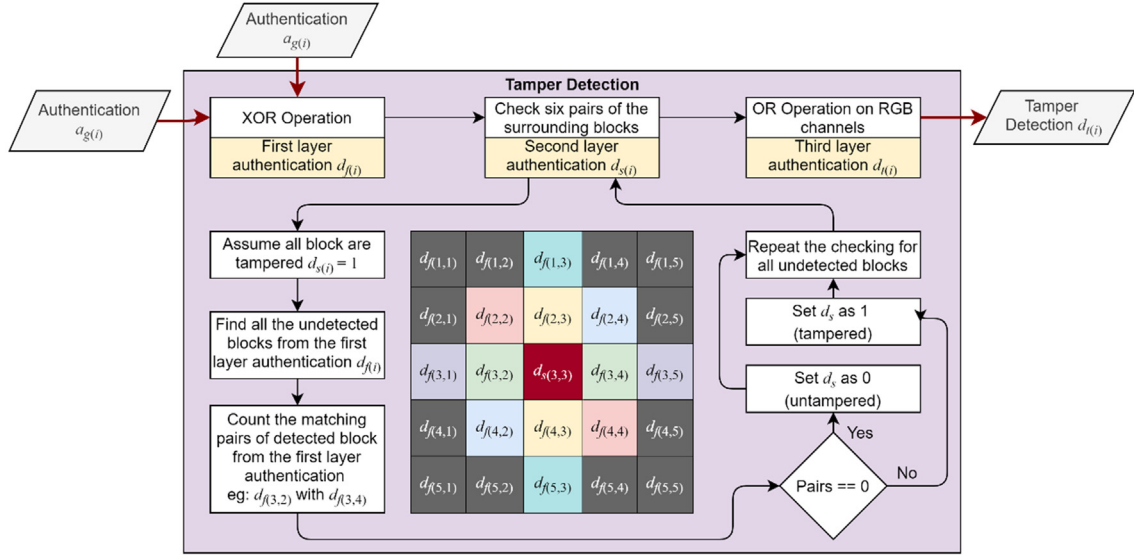


Fig. 6. The proposed tamper detection diagram in the AuSR1 scheme.

where  $r_s$  denotes the scrambled recovery bits,  $\oplus$  represents the XOR operation,  $map_p$  is the mapped recovery location,  $msb$  is a function to extract six LSB from  $map_p$ . This stage provides additional security for extracting the watermark, the attackers difficult to predict the recovery bits. This process aims to unscramble the data for recovery bit extraction.

Step 10: Repeat Step 6 to Step 9 for all blocks to obtain authentication data  $a_{e(i)}$  and the recovery data  $r_{e(i)}$ .

The reconstructed watermark consists of the authentication data  $a_{g(i)}$  and the recovery data  $r_{g(i)}$ . In addition, the watermark is extracted from two LSB of the tampered image, it consists of the authentication data  $a_{e(i)}$  and the recovery data  $r_{e(i)}$ . Those authentication data  $a_{g(i)}$  and  $a_{e(i)}$  will be used to authenticate the tampered image and tamper localization. Two recovery data  $r_{g(i)}$  and  $r_{e(i)}$  can be used to recover the tampered image.

### 3.5. Tamper detection

The proposed AuSR1 scheme has three-layer authentication check bits. The first-layer authentication compares the  $a_{g(i)}$  and the  $a_{e(i)}$ . The second-layer authentication checks the surrounding undetected block of the tampered image. Furthermore, the third layer authentication checks of the result second layer authentication for three RGB channels. The detail tamper detection process of the proposed AuSR1 scheme is depicted in Fig. 6, Fig. 7.

The tamper detection stage comprises of five steps:

Step 1: The first layer authentication is computed as defined by:

$$d_f = (a_{g1} \oplus a_{e1}) \vee (a_{g2} \oplus a_{e2}) \quad (9)$$

where  $a_{g1}$  and  $a_{g2}$  are the authentication data obtained from the six MSB of the tampered image,  $a_{e1}$  and  $a_{e2}$  is obtained from the two LSB of the tampered image. If the  $d_f$  value is equal to 1, it means the selected block has tampered. Otherwise, if the  $d_f$  value of 0, it indicates that the block does not tamper or the block has tampered but it is undetected. The first layer authentication bits have a probability of a 25% tampering area remaining undetected.

Step 2: The second layer authentication is computed by following Algorithm 2:

#### Algorithm 2: Second layer authentication algorithm

**Input:**  $d_f$

```

1  [M, N] = size(d_f)
2  for i = 1 to M
3      for j = 1 to N
4          p = 0
5          if (d_f(i, j) == 0)
6              p = p + (d_f(i, j - 1) ^ d_f(i, j + 1))
7              p = p + (d_f(i + 1, j - 1) ^ d_f(i - 1, j + 1))
8              p = p + (d_f(i + 1, j) ^ d_f(i - 1, j))
9              p = p + (d_f(i + 1, j + 1) ^ d_f(i - 1, j - 1))
10             p = p + (d_f(i, j - 2) ^ d_f(i, j + 2))
11             p = p + (d_f(i + 2, j) ^ d_f(i - 2, j))
12             d_s(i, j) = p > 0
13         else
14             d_s(i, j) = 1
15         end if
16     end for
17 end for
    
```

**Output:**  $d_s$

where  $d_f$  represents the result of the first layer authentication bit, and  $d_s$  represents the results of the second layer authentication bit. The second layer authentication checks the surrounding block with its pairs as shown in Fig. 6. Algorithm 2 checks the left and right of the block, checks the top and bottom of the block, check the diagonal pair of the blocks.

Step 3: The third layer authentication is performed as defined by:

$$d_t = d_sR \vee d_sG \vee d_sB \quad (10)$$

where  $d_sR$  is the  $d_s$  of the red channel,  $d_sG$  denotes the  $d_s$  of the green channel, and  $d_sB$  represents the  $d_s$  of the blue channel. If a



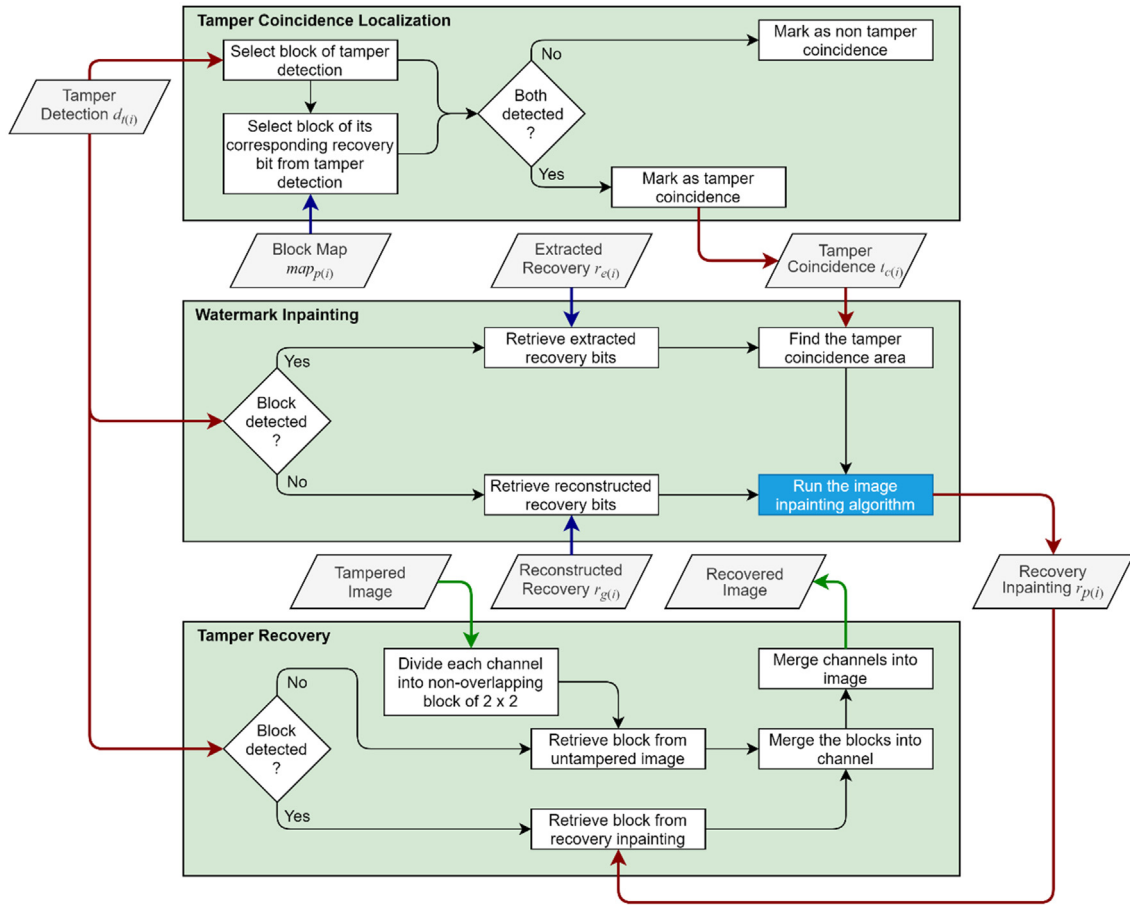


Fig. 7. The AuSR1 tamper recovery diagram.

tamper is detected on one of the image blocks of the RGB channel, then those blocks for all channels are considered tampered. The  $d_t$  value of '1' means that the block has tampered while the value of '0' means the block is untampered.

The tamper detection process produces  $d_{t(i)}$  and tamper detection images. In addition, this data will also be used for recovery. The AuSR1 scheme implements multi-layer authentication. The first layer authentication compares the extracted and the reconstructed authentication data from the tampered image. The first layer authentication produces a detection rate of about 75%. The second layer of authentication checks the surrounding block. If the surrounding block has been tampered, the selected block is considered a tampered block and vice versa. The second layer authentication can further increase the detection rate up to 99%. The third layer checks the combination result of the second layer authentication for RGB channels. If the result is equal to 1, the tampered image is detected and vice versa. The third layer authentication can achieve a detection rate of 100%.

### 3.6. The proposed watermark inpainting for recovered image

Tamper recovery consists of tamper coincidence localization, watermark inpainting, and tamper recovery as shown in Fig. 7. The proposed AuSR1 scheme presents a new image inpainting algorithm to improve the quality of the recovered image.

Tamper coincidence refers to a tampered block that recovery bits that reside on another block have tampered. The tamper coincidence problem is inevitable for the large tampering area of the images. Furthermore, the tamper coincidence problem significantly affects the effectiveness of the mapping block. Therefore,

block mapping should be adequately designed to prevent this tamper coincidence problem. Another way to solve the tamper coincidence problem is by using an image inpainting method. The image inpainting technique has been widely used to recover corrupted paintings or photographs. To find the tamper coincidence, the tamper coincidence localization should be computed to determine the locations. First, each block is mapped to the recovery block. If both blocks have been tampered with, then the blocks are marked as tamper coincidence. The tamper coincidence location is then stored into a matrix  $t_{c(i)}$ . The tamper coincidence matrix will be used in the watermark inpainting unit as one of its inputs.

The AuSR1 scheme proposes an image inpainting algorithm to solve the tamper coincidence problem. The proposed inpainting algorithm is discussed in the following steps:

Step 1: Prepare the recovery inpainting matrix  $r_{p(i)}$  from the extracted and reconstructed recovery data. First, check the tamper detection for each block. If the tamper is detected on the block, then retrieve the extracted recovery bits from  $r_{e(i)}$  matrix. If the tamper is not detected, then retrieve the reconstructed recovery bits from  $r_{g(i)}$  matrix. Noted that  $r_{e(i)}$  matrix contains tamper coincidence problem to be solved inside the tampered area. While  $r_{g(i)}$  helps to solve the tamper coincidence problem outside the tampered area.

Step 2: Find the tamper coincidence problem on the  $r_{p(i)}$  matrix based on the tamper coincidence matrix  $t_{c(i)}$ . Solve the tamper coincidence problem by using the proposed image inpainting technique as illustrated in Fig. 8.

where the (A) - (I) are non-tamper coincidences  $ntc$ ,  $tc$  represents a tamper coincidence problem that will be solved, and the empty boxes are another tamper coincidence that can be solved

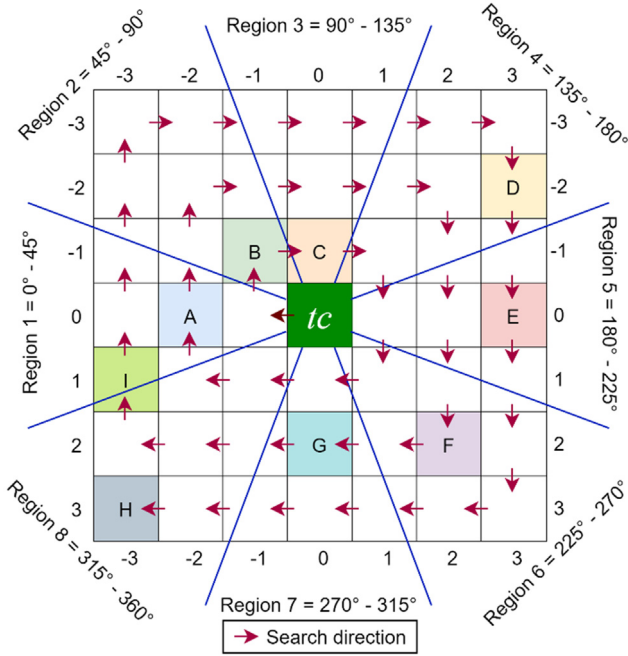


Fig. 8. The illustration of the proposed AuSR1 image inpainting.

in the next iteration. Fig. 8. shows the sample of nine non-tamper coincidences  $ntc$  from (A) to (I). The watermark inpainting algorithm will be implemented for all the tamper coincidences within the  $r_{p(i)}$  matrix. If the recovered image has more tamper coincidence, it consumes a large computational time to compute inpainting algorithm. According to Fig. 8, a simulation of the watermark inpainting is shown in Table 1.

where  $ntc_{x,y}$  represents the coordinates of each  $ntc$  and the proposed AuSR1 image inpainting considers eight regions. According to Table 1, each region of the matrix  $r_{p(i)}$  has its non-tamper coincidence  $ntc$  value from (A) to (I). The closest  $ntc$  against  $tc$  is selected if there are more than  $ntc$  in a region. Based on Table 1, the  $ntc$  of (I) is ignored and  $ntc$  of (A) is selected due to it is closer to  $tc$ .

Step 3: Find the  $ntc$  location from  $tc$  coordinates with outward spiral search direction.

Step 4: Compute the Euclidean distance between  $ntc$  and  $tc$  as defined by:

$$ed_i = \sqrt{(ntc_x - tc_x)^2 + (ntc_y - tc_y)^2} \quad (11)$$

where  $x$  and  $y$  are the corresponding coordinates of each  $ntc$  and  $tc$ .

Step 5: Normalize the  $ed_i$  value as defined by:

$$\alpha_i = \left(1 - \frac{ed_i}{\max(\{ed_1, \dots, ed_n\})}\right)^{ed_i} \quad (12)$$

where  $\alpha_i$  represents the weightage for surrounding non-tamper coincidence pixel,  $n = 8$  and  $ed_i$  represents the Euclidean distance of the non-tamper coincidence pixel.

Step 6: Compute the final  $tc$  value as defined by:

$$tc = \text{round}\left(\frac{\sum_{i=0}^7 (ntc_i \cdot \alpha_i)}{\sum_{i=0}^7 \alpha_i}\right) \quad (13)$$

where  $ntc_i$  represents the surrounding value of the non-tamper coincidence pixels,  $\alpha_i$  denotes the weightage for surrounding non-tamper coincidence pixel for each region.

Step 7: Repeat Step 3 to Step 6 to solve all the blocks with tamper coincidence problem.

Once the tamper coincidence problem within the matrix  $r_{p(i)}$  is solved, the tamper recovery process can be computed. First, each channel of the tampered image is divided into non-overlapping blocks of  $2 \times 2$  pixels. Check the tamper detection for each block based on the tamper detection data  $d_{t(i)}$ . Replace each tampered block with the recovery bits obtained from the recovery inpainting matrix  $r_{p(i)}$ . Next, merge all blocks into the recovered image and merge all RGB channels.

### 3.7. Evaluation

The performance of the proposed AuSR1 scheme is evaluated by imperceptibility measurement and the confusion matrix. Furthermore, the quality of the watermarked image and the recovered image are evaluated in terms of imperceptibility. The performance of the tamper detection algorithm is evaluated based on the confusion matrix.

#### 3.7.1. Imperceptibility measurement

The experiments conduct statistical measurements by comparing the original cover image and the watermarked image. The imperceptibility of the watermarked image is measured by using the peak-signal-to-noise-ratio (PSNR) and Structural SIMilarity (SSIM) index. PSNR is a quantitative analysis tool to measure the quality of the watermarked image compared to the cover image. SSIM is measured the image quality by concerning human visual characteristics such as structure, contrast, and brightness. PSNR can be defined by (Ferroukhi et al., 2019):

$$PSNR = 10 \log_{10} \frac{S^2}{\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (p(x,y) - q(x,y))^2} \quad (14)$$

where  $S$  represents the maximum pixel value,  $p(x,y)$  is the cover image,  $q(x,y)$  is the watermarked image, and  $x,y$  represent the coordinates. SSIM is used to measure the similarity between two

Table 1a

Simulation of the proposed AuSR1 image inpainting.

$ntc$	$ntc_{x,y}$	Pixel value	$ed_i$	$\alpha_i$	degree	region
A	-2, 0	120	2.0000	0.5714	22.5°	1
B	-1, -1	250	1.4142	0.6970	67.5°	2
C	0, -1	160	1.0000	0.7857	112.5°	3
D	3, -2	190	3.6056	0.2274	168.8°	4
E	3, 0	80	3.0000	0.3572	202.5°	5
F	2, 2	210	2.8284	0.3939	247.5°	6
G	0, 2	50	2.0000	0.5714	292.5°	7
H	-3, 3	10	4.2426	0.0909	337.5°	8
I*	-3, 1	90	3.1623	0.3224	4.1°	1
		max(ed)	4.2426			

\*I is ignored because  $ed_i$  value of I is greater than A.

images by using quality metric, it correlates with the quality perception of the human visual system (HVS). The SSIM is defined by (Ferroukhi et al., 2019):

$$SSIM(x, y) = [l(x, y)]^2 \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (15)$$

where  $l$  is a comparison function of luminance, and  $c$  is a comparison function of contrast, and  $s$  is a comparison function of the structure.  $l, c, s$  are defined by (Ferroukhi et al., 2019):

$$l(p, q) = \frac{2\mu_p\mu_q + C_1}{\mu_p^2 + \mu_q^2 + C_1} \quad (16)$$

$$c(p, q) = \frac{2\sigma_p\sigma_q + C_2}{\sigma_p^2 + \sigma_q^2 + C_2} \quad (17)$$

$$s(p, q) = \frac{\sigma_{pq} + C_3}{\sigma_p\sigma_q + C_3} \quad (18)$$

where  $p$  is the cover image,  $q$  is the watermarked image,  $C_1, C_2$ , and  $C_3$  are positive constants to avoid null denominators. The luminance function measures the closeness of two image's luminance. The contrast function calculates the similarity of two image's contrast. Next, the function measures the correlation coefficient between two images.

### 3.7.2. Confusion matrix

The tamper detection of the proposed AuSR1 scheme is evaluated by using the confusion matrix, which comprises true positive, false negative, false positive, and true negative. The confusion matrix is shown in Table 1.

According to Table 1, the true-positive is the number of correctly detected pixels within the actual tampered area. The false-negative represents the number of undetected pixels within the actual tampered area. The false-positive represents the number of incorrectly detected pixels within the untampered area. The true-negative means the number of undetected pixels within the untampered area. Furthermore, the tamper detection of the proposed AuSR1 scheme is evaluated by using true-positive rate (TPR), false-negative rate (FNR), false-positive rate (FPR), and true-negative rate (TNR) as defined by:

$$TPR = \frac{TP}{TP + FN} \quad (19)$$

$$FNR = \frac{FN}{TP + FN} \quad (20)$$

$$FPR = \frac{FP}{FP + TN} \quad (21)$$

$$TNR = \frac{TN}{FP + TN} \quad (22)$$

where TPR represents the ratio between the correctly detected area compared to the actual tampered area. The true-positive rate is also called recall or sensitivity. FNR represents the ratio between the undetected area and the actual tampered area. The higher the TPR value and lower the FNR value mean that the tamper detection algorithm correctly detects the tampered area of the image. The FPR represents the ratio between the incorrectly detected area and the untampered area. The TNR represents the ratio between undetected and untampered areas. The higher the FPR and the lower the TNR represent the tamper detection algorithm incorrectly detects the untampered area as the tampered area. In addition, the precision is also computed to measure the effectiveness of the tamper detection of the proposed AuSR1 scheme. The precision is defined by:

$$precision = \frac{TP}{TP + FP} \quad (23)$$

where *precision* represents the ratio between the true-positive compared to the true-positive and false-positive. *TP* is the true-positive, *FP* is the false positive. A high precision represents the superiority of the tamper detection algorithm in detecting the correct tampered area of the images.

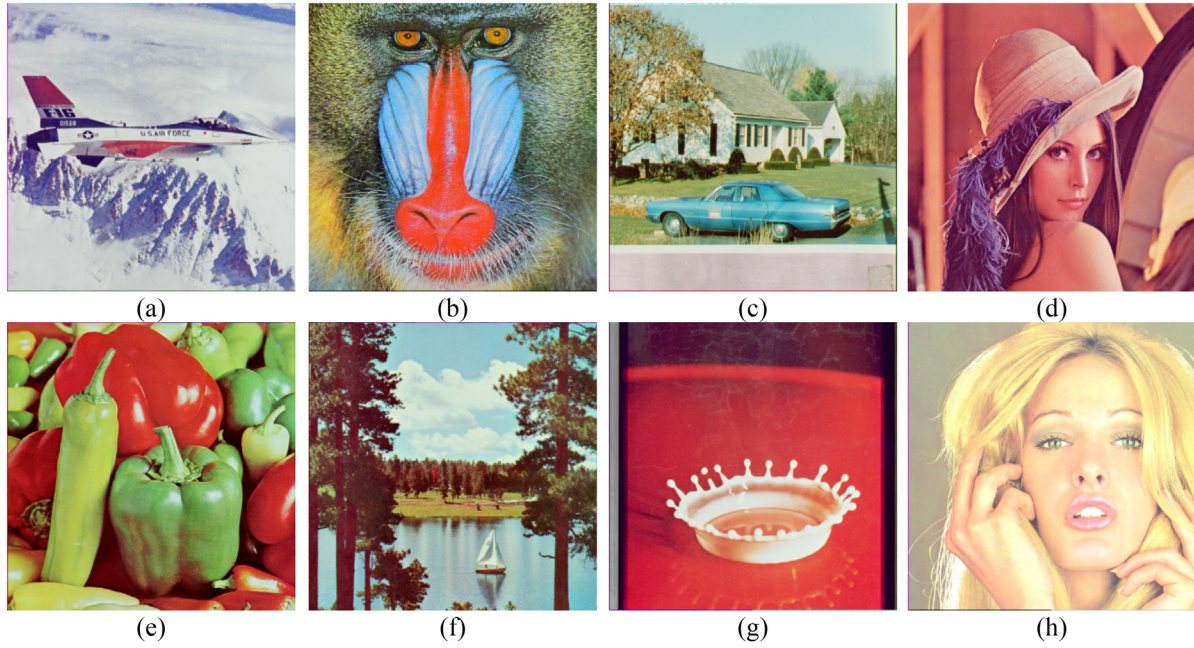
## 4. Experimental results

The AuSR1 scheme is evaluated using eight color images with the size of  $512 \times 512$  pixels. The images are namely "Airplane", "Baboon", "House", "Lena", "Peppers", "Sailboat", "Splash", and "Tiffany". The visual images are shown in Fig. 11. The experiments are performed on a computer with AMD Ryzen 7 5700U processor and 32 GB memory. The operating system installed is Windows 10 with MATLAB 2021a. The AuSR1 scheme is evaluated under regular alteration attacks in the center of the image. The collage attacks have been applied in the watermarked image with the various tamper attack sizes, including 10%, 20%, 30%, 40%, 50%, 60%, 70% and 80%. In addition, the watermarked images were also tested under irregular attacks by using Adobe Photoshop version 22.3.1.

Molina-Garcia et al. (Molina-Garcia et al., 2020) presented image tampering detection and self-recovery by using image inpainting. The authors replicated the scheme by Molina-Garcia et al. (Molina-Garcia et al., 2020) and improved on the bit-adjustment for embedding, hierarchical tamper detection, and image inpainting for recovery. The proposed scheme presented LSB shifting algorithm to decrease pixel variation between the cover image and watermarked image. The proposed scheme also presented three-layers tamper detection for improving the precision of the tamper detection. Scheme by Molina-Garcia (Molina-Garcia et al., 2020) interpolated the tamper coincidence problem by using an average of eight surrounding pixels of tamper coincidence. However, it still produced a granular effect in the recovered image. The scheme tested the image inpainting scheme by adding noises in the central region from 10% to 80%. This research presented a new image inpainting with a spiral outward direction to interpolate the tamper coincidence problem. Furthermore, the proposed scheme also tested with same noises in the central region from 10% to 80%. The experimental results in terms of imperceptibility of the watermarked image, precision of the tamper detection and the quality of the self-recovered image against various attacks are then compared to the existing schemes, including the scheme by Molina-Garcia (Molina-Garcia et al., 2020).

In fragile watermarking, the amount of embedding data into the cover image becomes crucial to obtain a high quality of the watermarked image. The AuSR1 scheme has embedded the watermark into two (2) LSB of the cover image. Schemes by Dadkhah (Dadkhah et al., 2014), Fan (Fan and Wang, 2018), Tai (Tai and Liao, 2018), and Molina-Garcia (Molina-Garcia et al., 2020) stated that two LSB were able to support image authentication and recovery in fragile watermarking. When two LSB were fully employed in the fragile watermarking, the quality of the watermarked image was about 44 dB. However, the proposed AuSR1 scheme using a new image inpainting technique with LSB shifting achieves the watermarked image quality of about 45 dB. The proposed scheme can improve 2.2% in terms of PSNR value from the existing schemes embedded into two LSB. The imperceptibility comparison of the proposed AuSR1 scheme with other existing schemes is listed in Tables 2 and 3.

According to Table 2, the proposed AuSR1 produces a higher PSNR value than other existing schemes, except for Tiffany image. Meanwhile, the quality of the watermarked image is slightly lower than a scheme by Molina-Garcia (Molina-Garcia et al., 2020).



**Fig. 11.** A set of test images (a) Airplane (b) Baboon (c) House (d) Lena (e) Peppers (f) Sailboat (g) Splash (h) Tiffany.

**Table 1b**  
Confusion matrix.

Tampered types	Detected	Undetected
Actual Tampered	True Positive (TP)	False Negative (FN)
Actual Untampered	False Positive (FP)	True Negative (TN)

Table 3 shows the proposed AuSR1 scheme superior to other watermarking schemes in terms of SSIM value. In this research, the watermarked images are tested under regular tamper attacks in the center of the image. The watermarked images have been tampered with under various Tampering Rates (TR). The watermarked images have tampered with tampering rates of 10% to 80%. The tamper detection performance of the AuSR1 scheme is evaluated by precision and recall. The comparison of tamper detection performance between the proposed AuSR1 scheme and the existing schemes is listed in Table 4.

The experimental results show that implementing three-level authentications can achieve high precision and recall values. The proposed AuSR1 scheme achieves slightly lower precision under a tampering rate of 40% due to the occurrence of false-positive detection. According to Table 4, it can be noticed that the proposed AuSR1 scheme produced a precision value of about 0.9938. The false-positive will occur when the image block was partially tam-

pered image. If a pixel in the image block has been tampered with, then its block considers a tampered area. In the image authentication scheme, the false-positive error can be recovered while the false negative error will never be recovered. The schemes by Dadkhah (Dadkhah et al., 2014), Tai (Tai and Liao, 2018), and Molina-Garcia (Molina-Garcia et al., 2020) used the block size of  $4 \times 4$  pixels. Large block size can increase the watermark capacity. However, this practice may lead to provide a high false-positive rate. In contrast, the proposed AuSR1 scheme utilizes a block size of  $2 \times 2$  pixels, it can achieve a higher precision value than the existing schemes. The experiments applied collage attacks at the center of the eight images. Each image undergoes various tampering rates from 10% up to 80%. Afterward, the tampered images are recovered using the proposed recovery algorithm. The experimental results show that the AuSR1 scheme yields superior recovered image quality compared to the existing schemes. The PSNR and SSIM values of the recovered image are listed in Tables 5 and 6.

According to Tables 5 and 6, the AuSR1 scheme produces high-quality of the recovered images under various tampering rates. The Splash image produced a highest PSNR value of 41.18 dB and for the House image obtained a lowest PSNR value of 35.67 dB under 10% tampering rate. It can be noticed that it showed significantly different results in the self-recovered image. The PSNR measurement utilized Mean Square Error (MSE), it has poor fidelity prediction and imperceptibility evaluation (Wang and Bovik,

**Table 2**  
The PSNR value comparison of the watermarked image between the proposed AuSR1 scheme and the other existing schemes.

Image	Tong (Tong et al., 2013)	Dadkhah (Dadkhah et al., 2014)	Singh (Singh and Singh, 2016)	Fan (Fan and Wang, 2018)	Tai (Tai and Liao, 2018)	Molina-Garcia (Molina-Garcia et al., 2020)	AuSR1
Airplane	37.88	44.12	37.88	44.11	44.12	44.69	<b>45.68</b>
Baboon	37.90	44.14	37.90	44.12	44.14	44.64	<b>45.70</b>
House	37.88	44.19	37.88	44.18	44.18	44.66	<b>45.69</b>
Lena	37.90	44.13	37.90	44.13	44.12	44.60	<b>45.71</b>
Peppers	37.79	44.06	37.79	44.06	44.06	44.54	<b>45.54</b>
Sailboat	37.90	44.12	37.90	44.10	44.11	44.61	<b>45.68</b>
Splash	37.84	44.08	37.84	44.08	44.09	44.47	<b>45.57</b>
Tiffany	37.44	43.85	37.44	43.84	43.85	44.87	<b>44.95</b>
Average	37.82	44.09	37.82	44.08	44.08	44.64	<b>45.57</b>

**Table 3**

The SSIM value comparison of the watermarked image between the proposed AuSR1 scheme and the other existing scheme.

Image	Tong (Tong et al., 2013)	Dadkhah (Dadkhah et al., 2014)	Singh (Singh and Singh, 2016)	Fan (Fan and Wang, 2018)	Tai (Tai and Liao, 2018)	Molina-Garcia (Molina-Garcia et al., 2020)	AuSR1
Airplane	0.9194	0.9782	0.9194	0.9781	0.9781	0.9812	<b>0.9889</b>
Baboon	0.9763	0.9941	0.9763	0.9941	0.9941	0.9947	<b>0.9990</b>
House	0.9319	0.9815	0.9319	0.9815	0.9815	0.9834	<b>0.9967</b>
Lena	0.9307	0.9820	0.9307	0.9820	0.9820	0.9840	<b>0.9993</b>
Peppers	0.9234	0.9791	0.9234	0.9791	0.9791	0.9816	<b>0.9991</b>
Sailboat	0.9494	0.9868	0.9493	0.9867	0.9868	0.9884	<b>0.9980</b>
Splash	0.8942	0.9695	0.8942	0.9695	0.9696	0.9737	<b>0.9983</b>
Tiffany	0.9246	0.9806	0.9246	0.9804	0.9805	0.9846	<b>0.9985</b>
Average	0.9312	0.9815	0.9312	0.9814	0.9815	0.9840	<b>0.9972</b>

**Table 4**

Comparison of the tamper detection compared to the existing scheme.

TR		Tong (Tong et al., 2013)	Dadkhah (Dadkhah et al., 2014)	Singh (Singh and Singh, 2016)	Fan (Fan and Wang, 2018)	Tai (Tai and Liao, 2018)	Molina-Garcia (Molina-Garcia et al., 2020)	AuSR1
20	Precision	0.9826	0.9658	0.9824	0.9025	0.9657	0.9336	<b>0.9978</b>
	Recall	0.7479	1.0000	0.7507	1.0000	0.9963	1.0000	<b>1.0000</b>
40	Precision	0.9936	0.9938	0.9937	0.9697	<b>0.9938</b>	0.9697	0.9918
	Recall	0.7489	1.0000	0.7501	1.0000	0.9966	1.0000	<b>1.0000</b>
60	Precision	0.9902	0.9801	0.9901	0.9801	0.9801	0.9608	<b>0.9925</b>
	Recall	0.7496	1.0000	0.7502	1.0000	0.9962	1.0000	<b>1.0000</b>
80	Precision	0.9956	0.9870	0.9956	0.9701	0.9870	0.9701	<b>1.0000</b>
	Recall	0.7505	1.0000	0.7487	1.0000	0.9962	1.0000	<b>1.0000</b>

**Table 5**

Comparison of the recovered image PSNR value under various tampering rate.

TR	Airplane	Baboon	House	Lenna	Peppers	Sailboat	Splash	Tiffany	Average
10	36.44	37.50	35.67	37.77	37.94	37.09	41.18	40.09	37.96
20	32.90	33.17	32.19	34.12	35.47	33.61	38.36	37.38	34.65
30	30.39	28.76	29.60	31.19	33.26	30.45	35.18	35.47	31.79
40	28.14	25.47	27.27	29.17	31.24	27.85	32.99	33.72	29.48
50	26.43	23.39	25.49	27.65	29.49	25.95	30.90	31.87	27.64
60	24.69	21.67	23.78	26.05	27.28	24.04	28.68	29.58	25.72
70	22.97	20.26	21.98	24.47	24.57	21.93	26.56	27.65	23.80
80	21.00	18.78	20.04	22.11	21.67	19.66	24.19	25.58	21.63

**Table 6**

Comparison of the recovered image SSIM value under various tampering rate.

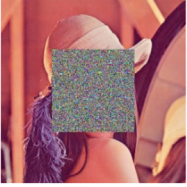
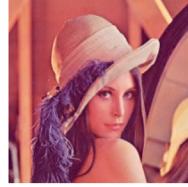
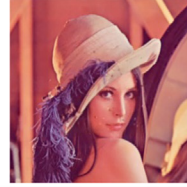

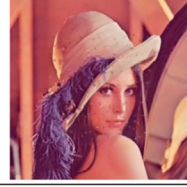

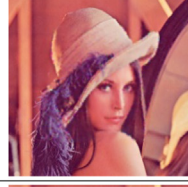
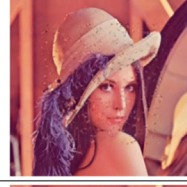
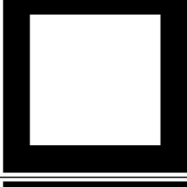
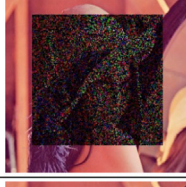
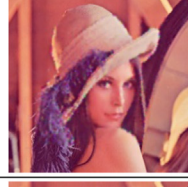
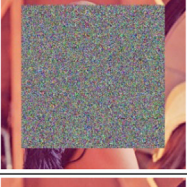

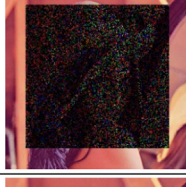
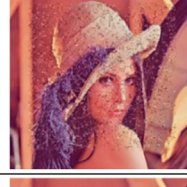
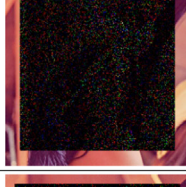
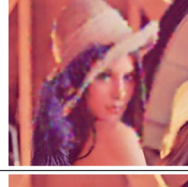
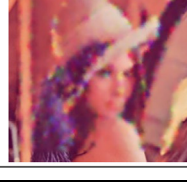
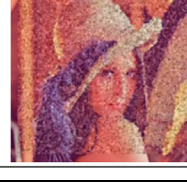
TR	Airplane	Baboon	House	Lenna	Peppers	Sailboat	Splash	Tiffany	Average
10	0.9794	0.9958	0.9879	0.9959	0.9960	0.9935	0.9973	0.9965	0.9928
20	0.9671	0.9894	0.9752	0.9909	0.9932	0.9857	0.9956	0.9942	0.9864
30	0.9496	0.9595	0.9584	0.9828	0.9890	0.9709	0.9920	0.9916	0.9742
40	0.9237	0.9047	0.9327	0.9731	0.9835	0.9502	0.9881	0.9883	0.9555
50	0.8937	0.8436	0.9033	0.9632	0.9764	0.9258	0.9825	0.9826	0.9339
60	0.8546	0.7730	0.8670	0.9511	0.9632	0.8931	0.9739	0.9711	0.9059
70	0.8067	0.7008	0.8175	0.9355	0.9402	0.8429	0.9624	0.9577	0.8705
80	0.7433	0.6227	0.7511	0.9072	0.9027	0.7662	0.9440	0.9377	0.8219

2009). PSNR is more sensitive against degradation that occurs in the image because of Gaussian noise addition (Horé and Ziou, 2013). In addition, the low texture on the watermarked image may produce low PSNR and SSIM values of the recovered image due to the significance distortion. In contrast, the watermarked image with high texture can be recovered with high PSNR and SSIM values. The center area of the Baboon image has the most complex texture compared to the available test images. Furthermore, the complexity of the image texture in the cover image also contribute to the effects of SSIM value of the recovered image. SSIM measurement is calculated based on loss correlation, luminance distortion and contrast distortion, it makes SSIM more correlated to the human visual system (Horé and Ziou, 2013). Therefore, the self-recovered of the Baboon image produced a high SSIM value

against tampering rate of 10% compared to the Airplane image. The visual comparison of the recovered image obtained from the proposed AuSR1 and the scheme by Molina-Garcia (Molina-Garcia et al., 2020) is shown in Table 7.

Table 7 shows regularly tampered in the center of the Lena image. Table 7 also shows the tampered image with collage attack and the tamper coincidence of the image. The scheme by Molina-Garcia (Molina-Garcia et al., 2020) shows that the scheme produces artifacts on the recovered image. The AuSR1 implemented an image inpainting algorithm to solve the tamper coincidence problem. The schemes by Tai et al. (Tai and Liao, 2018) and Molina-Garcia et al. (Molina-Garcia et al., 2020) also performed an image inpainting algorithm to solve tamper coincidence of the recovered image. However, the image inpainting algorithm in the

**Table 7**  
AuSR1 performance under regular attacks.

TR	Regular attack	Tamper detection	Tamper coincidence	The proposed AuSR1	Molina-Garcia [19]
10					
20					
30					
40					
50					
60					
70					
80					

scheme by Tai et al. (Tai and Liao, 2018) and Molina-Garcia et al. (Molina-Garcia et al., 2020) only consider eight surrounding blocks from the tamper coincidence. The experimental results show that tamper detection successfully marked the tampered area locations.

The untampered area is presented by black color and the tampered area is presented by white color in the second column of Table 7. In the recovery stage, the tampered image is recovered by using the recovery data which previously embedded into the two LSB. The

recovery data itself is embedded into another block based on the chaotic block map. A large tampering area may produce tamper coincidence. The tamper coincidence has occurred when the original block location and the recovery data location have been tampered with. To overcome this problem, the image inpainting algorithm is employed in the AuSR1 scheme. The experimental results show that the proposed image inpainting on AuSR1 produces better-recovered image quality than the existing schemes by Molina-Garcia et al. (Molina-Garcia et al., 2020). The comparison of the PSNR and SSIM values between the proposed AuSR1 scheme and other schemes is shown in Tables 8 and 9.

The schemes by Tong (Tong et al., 2013) and Singh (Singh and Singh, 2016) embedded the watermark bits into three LSB of the cover image. The schemes produced an average PSNR value of 37 dB. Even though the schemes modified three LSB to store a large amount of recovery data, the scheme will produce the low quality of the watermarked image. The scheme is not able to achieve the high quality of the recovered image. Hence, the modified two LSB in the fragile watermarking is the most reliable option for embedding the watermark. It can preserve the watermarked image quality and retain the quality of the recovered image under various tamper attacks. The modified one LSB in the fragile watermarking has limited space to store the recovery bits. In addition, schemes by Tai et al. (Tai and Liao, 2018) and Molina-Garcia et al. (Molina-Garcia et al., 2020) used an average value of the surrounding pixel with the size of  $3 \times 3$  pixels to overcome the tamper coincidence pixels. The surrounding pixel of each pixel may have tamper coincidence, therefore the image inpainting value may significantly obtain error distortion. In contrast, the AuSR1 scheme considers the distance between the tamper coincidence pixel and the neighboring pixels which are untampered coincidence pixels. The visual comparison of the PSNR and SSIM values between AuSR1 and the existing benchmarks is shown in Fig. 12.

Fig. 12. shows that AuSR1 produces high quality of the recovered images compared to the existing schemes. The AuSR1 scheme produces a slightly higher PSNR value than the scheme by Molina-Garcia et al. (Molina-Garcia et al., 2020). The AuSR1 scheme can

produce the PSNR value of 21.63 dB of the recovered image under a tampering rate of 80%. In contrast, the schemes by Tai (Tai and Liao, 2018), Fan (Fan and Wang, 2018), Singh (Singh and Singh, 2016), Dadkhah (Dadkhah et al., 2014), and Tong (Tong et al., 2013) rapidly degrade the PSNR value against a higher tampering rate. The AuSR1 scheme produces a superior SSIM value compared to the existing schemes. The other existing schemes produce lower SSIM values than the proposed scheme under a large tampering rate. The AuSR1 successfully maintains an SSIM value of 0.8219 under an 80% tampering rate, while the best existing method by Molina-Garcia et al. (Molina-Garcia et al., 2020) can only reach the SSIM value of 0.3958 on the same tampering rate.

In this research, the watermarked image is also tampered with irregularly tamper attacks using Adobe Photoshop 22.3.1. The watermarked image has tampered with copy-move, copy-paste, mosaic, removal, color change, background change, and face swab attack. The copy-move is an attack that copies the object and moves it to another location on the image. This attack is widely used to duplicate the interest object on an image. The copy-paste attack copies a part of the external image to be embedded into the target image. Next, the mosaic attack has been widely used to censor the specific part of the object image. For example, a crime victim's face is censored by using a mosaic attack. The removal attack is used to remove some parts of the image. A newer version of Adobe Photoshop has a content-aware tool that allows the user to remove a specific area of an image and replace it with new information based on the surrounding colors. Traditionally, the removed part of the image was filled with a background color. Next, the color change attack modifies the color of an object on the image. The background change attacks become popular in editing digital images. This attack can replace the background color with the other colors. Next, the face swab attack has the capability to replace the human face on the image with other human faces from the external image. This type of attack has been misused to incriminate someone by swabbing their face to the other face on the image. The visualization of the tamper detection and the recovered image is shown in Table 11.

**Table 8**  
Comparison of the recovered image PSNR value of the existing scheme.

TR	Tong (Tong et al., 2013)	Dadkhah (Dadkhah et al., 2014)	Singh (Singh and Singh, 2016)	Fan (Fan and Wang, 2018)	Tai (Tai and Liao, 2018)	Molina-Garcia (Molina-Garcia et al., 2020)	AuSR1
10	34.20	22.51	26.55	31.47	25.89	37.34	<b>37.96</b>
20	25.77	17.32	21.47	28.36	20.57	33.98	<b>34.65</b>
30	21.04	14.52	18.27	21.62	17.43	31.28	<b>31.79</b>
40	17.26	12.64	15.96	15.79	15.21	28.47	<b>29.48</b>
50	14.29	11.40	14.16	15.69	13.54	26.00	<b>27.64</b>
60	11.84	10.39	12.59	11.57	12.01	23.51	<b>25.72</b>
70	9.82	9.61	11.29	11.57	10.80	21.23	<b>23.80</b>
80	8.11	9.03	10.23	8.10	9.81	19.20	<b>21.63</b>

**Table 9**  
Comparison of the recovered image SSIM value of the existing scheme.

TR	Tong (Tong et al., 2013)	Dadkhah (Dadkhah et al., 2014)	Singh (Singh and Singh, 2016)	Fan (Fan and Wang, 2018)	Tai (Tai and Liao, 2018)	Molina-Garcia (Molina-Garcia et al., 2020)	AuSR1
10	0.9733	0.9131	0.9290	0.9731	0.9384	0.9714	<b>0.9928</b>
20	0.9171	0.7983	0.8310	0.9502	0.8443	0.9390	<b>0.9864</b>
30	0.8282	0.6855	0.7257	0.8875	0.7364	0.8977	<b>0.9742</b>
40	0.7150	0.5731	0.6215	0.7230	0.6226	0.8368	<b>0.9555</b>
50	0.5849	0.4704	0.5139	0.7202	0.5135	0.7571	<b>0.9339</b>
60	0.4520	0.3586	0.3984	0.4249	0.3899	0.6460	<b>0.9059</b>
70	0.3233	0.2506	0.2855	0.4249	0.2744	0.5157	<b>0.8705</b>
80	0.2042	0.1511	0.1799	0.0094	0.1655	0.3958	<b>0.8219</b>

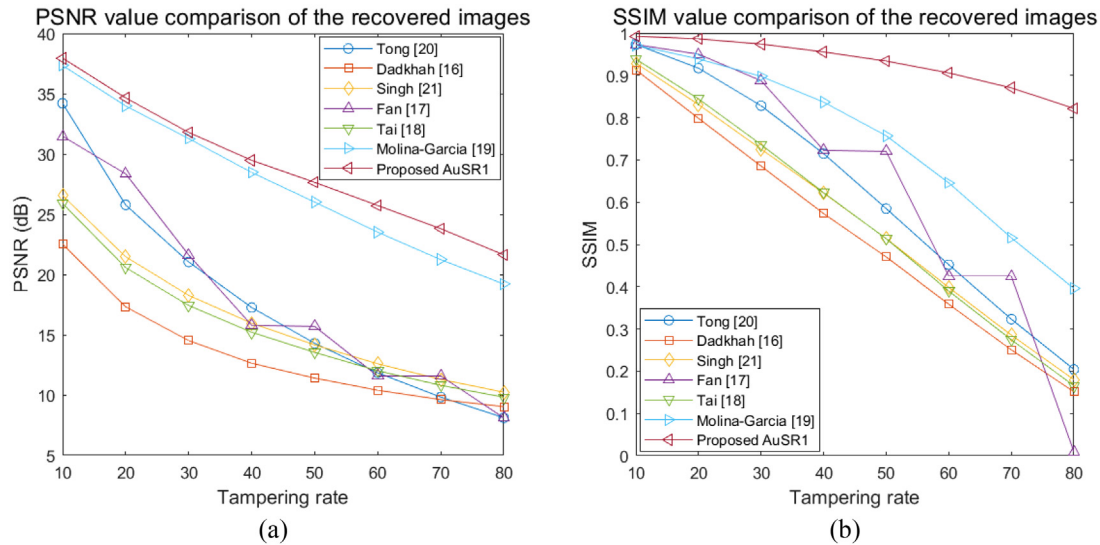


Fig. 12. The comparison of the (a) PSNR value (b) SSIM value of the recovered image under regular attacks.

The proposed AuSR1 scheme achieves a TPR value of 0.9996 under irregular attacks. The detection rate under irregular attacks is slightly lower than the detection rate under regular attacks due to the uniform shape of the irregular attacks. Therefore, the uniformity attack will cut through the block, then it can contribute to the false-positive detection. Overall, the AuSR1 scheme can recover the tampered image and provide a good quality of the recovered image. The proposed scheme produces an overall false-positive rate of 0.005 for both regular and irregular attacks. In the tamper recovery algorithm, the false-positive detection will be considered as tampered area, and the whole block will be recovered. In contrast, false-negative detection will lead to unrecoverable tampered areas. The experimental results obtained from the AuSR1 scheme show that the false-negative detection only occurred on the irregular attack with the overall value of 0.0004, it is lower than the false positive rate of 0.005. Furthermore, the AuSR1 successfully recover the tampered area with high-quality under irregular attacks. The Normalized Cross-correlation (NC) and Bit Error Rate (BER) of the recovered image under tampering rates of 10–40% are presented in Table 12.

The watermark correlation coefficient is computed between two watermark images. The first watermark image is obtained from the difference between the original and watermarked images. The second watermark image is obtained from the difference between the original and recovered images. The first and the second watermarks are computed by using NC and BER measurements to obtain the correlation coefficients. The experimental results showed that the AuSR1 achieved NC and BER values of about 0.966 and 0.0691 under 10% tampering rate. It means that the recovered image under various tamper attacks is closer towards the cover image. The computational time of the proposed scheme is computed based on the time taken to run the proposed algorithm. The computational time for the watermark embedding process is listed in Table 13.

According to Table 13, the watermark embedding requires an average time of about 4.89 s for generating block map, watermark data and embedding watermark. The watermark embedding process consumed largest computational time due to the scrambled watermark was performed on the 2 LSB. The scrambled LSB provides additional security of the watermarked image and the authentication bits are located on the scrambled locations for each block. The attackers may difficult to destroy the integrity of the images. The

computational time for the tamper detection process is shown in Table 14.

Table 14 shows the computational time for block map generation, watermark extraction and three-layers tamper detection under various tampering rates. Each image was tampered by using regular attacks, with the tampering rates from 10% to 80%. First, the block map should be generated from the watermarked image. Thus, the watermark is then extracted under various attacks. The first layer authentication performs XOR operation on two authentication matrices. The second layer authentication removes the mark on the untampered area based on the first layer authentication. The third layer authentication also performs OR operation on three RGB channels. Overall, the proposed scheme requires 2.59 s to detect the altered image. Next, the computational time for the tampered recovery is listed in Table 15.



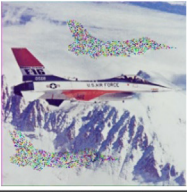

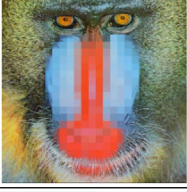
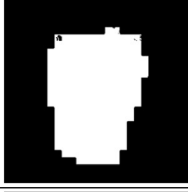
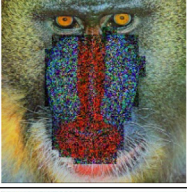
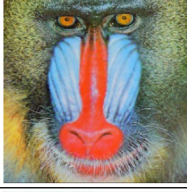


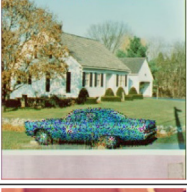

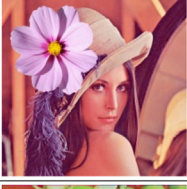
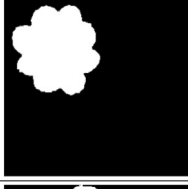
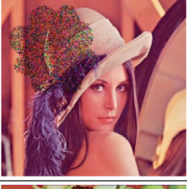
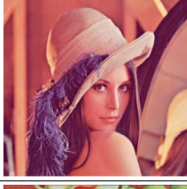
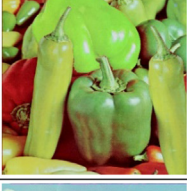










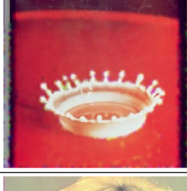
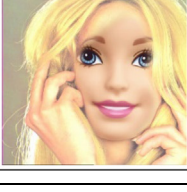

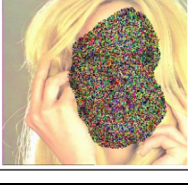
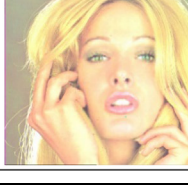
Based on the Table 15, the tampered recovery consumes a large computational time for recovering the altered image due to it involves the image inpainting technique. The image inpainting required a large computational time if the large altered was occurred in the image. To recovered image, it requires pre-detection, tamper detection, coincidence localization, inpainting process, then finally tamper recovery. The cumulative of the recovered image required 9.7 s under 10% tampering rates. The watermark inpainting process consumes a large computational time, the consequence a high tampering rate, it will provide a high tamper coincidence. The proposed scheme will find the available surrounding block to cover the tamper coincidence area. As a result, applying 80% tampering rate consumes 52.6 s to perform image inpainting.

## 5. Conclusion

This research has presented an AuSR1 scheme using blind fragile image watermarking for tamper detection and self-recovery. The AuSR1 utilizes a small block size of  $2 \times 2$  to achieve high precision of the tamper detection and quality of the recovered image. The watermark data is obtained from the cover image, while it consists of authentication and recovery bits. The watermark data has been scrambled to ensure embedding security. The authentication data is embedded into the current block location, while the recovery data is embedded into the distant location from the original location based on the block mapping. The watermark data is then



**Table 11**  
 Tamper detection and tamper recovery obtained from AuSR1 scheme under irregular attacks.

Tampered image	Tamper detection		Tamper coincidence	Recovered image	
		Tamper rate = 9.83% TPR = 0.9995 FPR = 0.0042			PSNR = 43.63 dB SSIM = 0.9868
		Tamper rate = 31.54% TPR = 0.9995 FPR = 0.0042			PSNR = 28.78 dB SSIM = 0.9664
		Tamper rate = 11.43% TPR = 0.9998 FPR = 0.0023			PSNR = 36.54 dB SSIM = 0.9911
		Tamper rate = 16.25% TPR = 0.9998 FPR = 0.0031			PSNR = 36.72 dB SSIM = 0.9953
		Tamper rate = 27.26% TPR = 0.9999 FPR = 0.0064			PSNR = 34.30 dB SSIM = 0.9934
		Tamper rate = 58.92% TPR = 0.9999 FPR = 0.0100			PSNR = 22.39 dB SSIM = 0.8539
		Tamper rate = 67.97% TPR = 0.9998 FPR = 0.0189			PSNR = 25.50 dB SSIM = 0.9543
		Tamper rate = 29.53% TPR = 0.9987 FPR = 0.0056			PSNR = 35.84 dB SSIM = 0.9921

**Table 12**  
The NC and BER value of the recovered image under tampering rates 10–40%.

Image	10		20		30		40	
	NC	BER	NC	BER	NC	BER	NC	BER
Airplane	0.9658	0.0690	0.9306	0.1389	0.8951	0.2087	0.8593	0.2796
Baboon	0.9650	0.0691	0.9291	0.1391	0.8937	0.2087	0.8578	0.2788
House	0.9648	0.0689	0.9291	0.1391	0.8936	0.2088	0.8576	0.2793
Lena	0.9652	0.0688	0.9301	0.1385	0.8943	0.2082	0.8588	0.2784
Pepper	0.9660	0.0688	0.9317	0.1387	0.8970	0.2089	0.8609	0.2797
Sailboat	0.9654	0.0690	0.9301	0.1392	0.8946	0.2090	0.8589	0.2794
Splash	0.9672	0.0692	0.9329	0.1395	0.8981	0.2089	0.8633	0.2792
Tiffany	0.9689	0.0696	0.9371	0.1408	0.9050	0.2115	0.8724	0.2825
Average	0.9660	0.0691	0.9313	0.1392	0.8964	0.2091	0.8611	0.2796

**Table 13**  
Watermark embedding time in seconds (s).

Image	Block map generation	Watermark generation	Watermark embedding	Cumulative of the watermark embedding
Airplane	0.0581	1.6919	3.0937	4.8437
Baboon	0.0123	1.6440	3.3594	5.0157
House	0.0506	1.6057	3.2188	4.8751
Lena	0.0212	1.6507	3.0937	4.7656
Pepper	0.0625	1.8751	3.0936	5.0312
Sailboat	0.0443	1.7369	3.2968	5.0780
Splash	0.0232	1.6018	3.0780	4.7030
Tiffany	0.0332	1.6074	3.2187	4.8593
Average	0.0382	1.6767	3.1816	4.8965

**Table 14**  
Tampered detection time under various tampering rates in seconds (s).

Tampering rate	Pre-detection			Tamper Detection			Cumulative of the tamper detection
	Block map generation	Watermark extraction	Cumulative of the pre-detection	1st layer	2nd layer	3rd layer	
10	0.1015	5.9258	6.0274	0.0011	2.5829	0.0078	2.5918
20	0.0273	5.9747	6.0020	0.0016	2.2559	0.0081	2.2656
30	0.0801	6.0664	6.1465	0.0019	2.0265	0.0067	2.0351
40	0.0742	5.9844	6.0586	0.0013	1.8347	0.0078	1.8438
50	0.0683	5.9102	5.9785	0.0018	1.5797	0.0083	1.5898
60	0.0781	5.9629	6.0410	0.0014	1.3003	0.0069	1.3086
70	0.0352	5.8887	5.9239	0.0012	1.0659	0.0072	1.0743
80	0.0254	5.7266	5.7520	0.0015	0.8441	0.0078	0.8534

**Table 15**  
Tampered recovery time under various tampering rates in seconds (s).

Tampering rate	Pre-detection	Tamper detection	Tamper recovery			Cumulative of the recovered image
			Coincidence localization	Watermark inpainting	Tamper recovery	
10	6.0274	2.5918	0.0014	0.1490	0.9804	9.7500
20	6.0020	2.2656	0.0039	0.6289	1.2930	10.1934
30	6.1465	2.0351	0.0098	1.5293	1.4044	11.1250
40	6.0586	1.8438	0.0215	3.0977	1.6524	12.6739
50	5.9785	1.5898	0.0156	5.5742	1.7970	14.9552
60	6.0410	1.3086	0.0137	10.6778	1.9318	19.9728
70	5.9239	1.0743	0.0176	21.8789	2.1741	31.0687
80	5.7520	0.8534	0.0156	52.6679	2.3674	61.6563

embedded into the two LSB by using LSB shifting algorithm to maintain the quality of the watermarked image. The experimental results show that the proposed AuSR1 scheme achieves a high PSNR value of about 45.57 dB and an SSIM value of 0.9972 for the watermarked image. It outperforms the existing schemes in terms of PSNR and SSIM values since the LSB shifting algorithm can minimise the pixel intensity variation between the cover and watermarked images. The watermarked images are tested under regular attacks with a tampering rate of up to 80% and irregular attacks. The AuSR1 presents three-layers authentication check bits

to achieve a high detection rate. The AuSR1 achieves a high recall value of 1 and a high precision value of 0.9943, which outperforms the existing schemes. The AuSR1 scheme also proposes a new image inpainting technique to overcome the tamper coincidence problem, it plays an important role for recovering the altered image. The proposed image inpainting technique achieves high quality of the recovered image with SSIM value of 0.9339 under a 50% tampering rate. It searches the non-tamper coincidence pixel in a spiral outward direction to interpolate the tamper coincidence problem with the weight factors.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This research was supported by Universiti Malaysia Pahang through the Fundamental Research Grant Scheme RDU190370.

## References

- Ray, A., Roy, S., 2020. Recent trends in image watermarking techniques for copyright protection: a survey. *Int. J. Multimed. Inf. Retr.* 9 (4), 249–270. <https://doi.org/10.1007/S13735-020-00197-9>.
- Hemida, O., Huo, Y., He, H., Chen, F., 2019. A restorable fragile watermarking scheme with superior localization for both natural and text images. *Multimed. Tools Appl.* 78 (9), 12373–12403. <https://doi.org/10.1007/S11042-018-6664-3>.
- Belferdi, W., Behloul, A., Noui, L., 2019. A Bayer pattern-based fragile watermarking scheme for color image tamper detection and restoration. *Multimed. Syst. Signal Process.* 30 (3), 1093–1112. <https://doi.org/10.1007/s11045-018-0597-x>.
- Hong, W., Chen, J., Chang, P.S., Wu, J., Chen, T.S., Lin, J., 2021. A color image authentication scheme with grayscale invariance. *IEEE Access* 9, 6522–6535. <https://doi.org/10.1109/ACCESS.2020.3047270>.
- Huang, R., Liu, H., Liao, X., Sun, S., 2019. A divide-and-conquer fragile self-embedding watermarking with adaptive payload. *Multimed. Tools Appl.* 78 (18), 26701–26727.
- Su, G.D., Chang, C.C., Chen, C.C., 2021. A hybrid-Sudoku based fragile watermarking scheme for image tampering detection. *Multimed. Tools Appl.* 80 (8), 12–13. <https://doi.org/10.1007/s11042-020-10451-1>.
- Jafari Barani, M., Yousefi Valandar, M., Ayubi, P., 2019. A new digital image tamper detection algorithm based on integer wavelet transform and secured by encrypted authentication sequence with 3D quantum map. *Optik* 187, 205–222.
- Gul, E., Ozturk, S., 2019. A novel hash function based fragile watermarking method for image integrity. *Multimed. Tools Appl.* 78 (13), 17701–17718. <https://doi.org/10.1007/S11042-018-7084-0>.
- Gul, E., Ozturk, S., 2021. A novel pixel-wise authentication-based self-embedding fragile watermarking method. *Multimed. Syst.* 27 (3), 531–545. <https://doi.org/10.1007/S00530-021-00751-3>.
- Gul, E., Ozturk, S., 2020. A novel triple recovery information embedding approach for self-embedded digital image watermarking. *Multimed. Tools Appl.* 79 (41–42), 31239–31264. <https://doi.org/10.1007/s11042-020-09548-4>.
- Prasad, S., Pal, A.K., 2020. A Secure Fragile Watermarking Scheme for Protecting Integrity of Digital Images. *Iran J. Sci. Technol. Trans Electr. Eng.* 44 (2), 703–727.
- Ouyang, J., Zhang, X., Wen, X., 2020. Robust Hashing Based on Quaternion Gyration Transform for Image Authentication. *IEEE Access* 8, 220585–220594. <https://doi.org/10.1109/ACCESS.2020.3043111>.
- Zhang, L., Wei, D., 2019. Dual DCT-DWT-SVD digital watermarking algorithm based on particle swarm optimization. *Multimed. Tools Appl.* 78 (19), 28003–28023. <https://doi.org/10.1007/S11042-019-07902-9>.
- Kumar, S., Singh, B.K., 2021. DWT based color image watermarking using maximum entropy. *Multimed. Tools Appl.* 80 (10), 15487–15510. <https://doi.org/10.1007/S11042-020-10322-9>.
- Kang, J., Hou, J.U., Ji, S., Lee, H.K., 2020. Robust Spherical Panorama Image Watermarking against Viewpoint Desynchronization. *IEEE Access* 8, 127477–127490. <https://doi.org/10.1109/ACCESS.2020.3006980>.
- Dadkhah, S., Abd Manaf, A., Hori, Y., Ella Hassanien, A., Sadeghi, S., 2014. An effective SVD-based image tampering detection and self-recovery using active watermarking. *Signal Process. Image Commun.* 29 (10), 1197–1210. <https://doi.org/10.1016/J.IMAGE.2014.09.001>.
- Fan, M.Q., Wang, H.X., 2018. An enhanced fragile watermarking scheme to digital image protection and self-recovery. *Signal Process. Image Commun.* 66, 19–29. <https://doi.org/10.1016/J.IMAGE.2018.04.003>.
- Tai, W.L., Liao, Z.J., 2018. Image self-recovery with watermark self-embedding. *Signal Process. Image Commun.* 65, 11–25. <https://doi.org/10.1016/J.IMAGE.2018.03.011>.
- Molina-García, J., García-Salgado, B.P., Ponomaryov, V., Reyes-Reyes, R., Sadovnichiy, S., Cruz-Ramos, C., 2020. An effective fragile watermarking scheme for color image tampering detection and self-recovery. *Signal Process. Image Commun.* 81, 115725–115725. <https://doi.org/10.1016/j.image.2019.115725>.
- Tong, X., Liu, Y., Zhang, M., Chen, Y., 2013. A novel chaos-based fragile watermarking for image tampering detection and self-recovery. *Signal Process. Image Commun.* 28 (3), 301–308. <https://doi.org/10.1016/J.IMAGE.2012.12.003>.
- Singh, D., Singh, S.K., 2016. Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability. *J. Vis. Commun. Image Represent.* 38, 775–789. <https://doi.org/10.1016/j.jvcir.2016.04.023>.
- Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T., 1997. Secure spread spectrum watermarking for multimedia. *IEEE Trans. Image Process.* 6 (12), 1673–1687. <https://doi.org/10.1109/83.650120>.
- A. Ouahabi, "Signal and Image Multiresolution Analysis." p. 308, 2012, Accessed: Jan. 01, 2022. [Online]. Available: <https://www.wiley.com/en-us/Signal+and+Image+Multiresolution+Analysis-p-9781118568668>.
- Hsu, C.T., Wu, J.L., 1998. Multiresolution watermarking for digital images. *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process* 45 (8), 1097–1101. <https://doi.org/10.1109/82.718818>.
- Wang, Y., Doherty, J.F., Van Dyck, R.E., 2002. A wavelet-based watermarking algorithm for ownership verification of digital images. *IEEE Trans. Image Process.* 11 (2), 77–88. <https://doi.org/10.1109/83.982816>.
- Zermi, N., Khaldi, A., Kafi, R., Kahlessenane, F., Euschi, S., 2021. A DWT-SVD based robust digital watermarking for medical image security. *Forensic Sci. Int.* 320, 110691. <https://doi.org/10.1016/J.FORSCIINT.2021.110691>.
- Soni, N., Saini, I., Singh, B., 2020. An integer wavelet transform and pixel value differencing based feature specific hybrid technique for 2D ECG steganography with high payload capacity. *Multimed. Tools Appl.* 80 (6), 8505–8540. <https://doi.org/10.1007/S11042-020-09856-9>.
- Hsu, C.T., Wu, J.L., 1999. Hidden digital watermarks in images. *IEEE Trans. Image Process.* 8 (1), 58–68. <https://doi.org/10.1109/83.736686>.
- Qureshi, M.A., Deriche, M., Beghdadi, A., Amin, A., 2017. A critical survey of state-of-the-art image inpainting quality assessment metrics. *J. Vis. Commun. Image Represent.* 49, 177–191. <https://doi.org/10.1016/J.JVCIR.2017.09.006>.
- Wang, N., Zhang, Y., Zhang, L., 2021. Dynamic selection network for image inpainting. *IEEE Trans. Image Process.* 30, 1784–1798. <https://doi.org/10.1109/TIP.2020.3048629>.
- Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S., 2018. Generative Image Inpainting with Contextual Attention. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 5505–5514. <https://doi.org/10.1109/CVPR.2018.00577>.
- Bugeau, A., Bertalmio, M., Caselles, V., Sapiro, G., 2010. A comprehensive framework for image inpainting. *IEEE Trans. Image Process.* 19 (10), 2634–2645. <https://doi.org/10.1109/TIP.2010.2049240>.
- Ouahabi, A., Taleb-Ahmed, A., 2021. Deep learning for real-time semantic segmentation: Application in ultrasound imaging. *Pattern Recognit. Lett.* 144, 27–34. <https://doi.org/10.1016/J.PATREC.2021.01.010>.
- Mimouna, A. et al., 2020. OLIMP: A Heterogeneous Multimodal Dataset for Advanced Environment Perception. *Electron* 9 (4), 560. <https://doi.org/10.3390/ELECTRONICS9040560>.
- Quan, Y., Teng, H., Chen, Y., Ji, H., May 2021. Watermarking Deep Neural Networks in Image Processing. *IEEE Trans. Neural Networks Learn. Syst.* 32 (5), 1852–1865. <https://doi.org/10.1109/TNNLS.2020.2991378>.
- Ouahabi, A. "A review of wavelet denoising in medical imaging," *2013 8th Int. Work. Syst. Signal Process. Their Appl. WoSSPA 2013*, pp. 19–26, 2013, 10.1109/WOSSPA.2013.6602330.
- Pluim, J.P.W., Fitzpatrick, J.M., 2003. Image registration. *IEEE Trans. Med. Imaging* 22 (11), 1341–1343. <https://doi.org/10.1109/TMI.2003.819272>.
- Khaldi, Y., Benzaoui, A., Ouahabi, A., Jacques, S., Taleb-Ahmed, A., 2021. Ear recognition based on deep unsupervised active learning. *IEEE Sens. J.* 21 (18), 20704–20713. <https://doi.org/10.1109/JSEN.2021.3100151>.
- Adjabi, I., Ouahabi, A., Benzaoui, A., Jacques, S., Jan. 2021. Multi-Block Color-Binarized Statistical Images for Single-Sample Face Recognition. *Sensors* 21 (3), 728. <https://doi.org/10.3390/S21030728>.
- El Morabit, S., Rivenq, A., Zighem, M.E.N., Hadid, A., Ouahabi, A., Taleb-Ahmed, A., 2021. Automatic pain estimation from facial expressions: a comparative analysis using off-the-shelf CNN architectures. *Electron.* 10 (16), 1926. <https://doi.org/10.3390/ELECTRONICS10161926>.
- Adjabi, I., Ouahabi, A., Benzaoui, A., Taleb-Ahmed, A., 2020. Past, present, and future of face recognition: a review. *Electron* 9 (8), 1188. <https://doi.org/10.3390/ELECTRONICS9081188>.
- Sreenivas, K., Kamakshi Prasad, V., 2016. "Improved block encoding method for an image self-recovery approach. 2016 Int. Conf. Inf. Commun. Embed. Syst. ICICES." <https://doi.org/10.1109/ICICES.2016.7518879>.
- Cao, F., An, B., Wang, J., Ye, D., Wang, H., Jan. 2017. Hierarchical recovery for tampered images based on watermark self-embedding. *Displays* 46, 52–60. <https://doi.org/10.1016/J.DISPLA.2017.01.001>.
- Bolourian Haghghi, B., Taherinia, A.H., Harati, A., 2018. TRLH: Fragile and blind dual watermarking for image tamper detection and self-recovery based on lifting wavelet transform and halftoning technique. *Journal of Visual Communication and Image Representation* 50, 49–64.
- Qin, C., Wang, H., Zhang, X., Sun, X., 2016. Self-embedding fragile watermarking based on reference-data interleaving and adaptive selection of embedding mode. *Inf. Sci. (Ny)* 373, 233–250. <https://doi.org/10.1016/J.INS.2016.09.001>.
- Ferroukhi, M., Ouahabi, A., Attari, M., Habchi, Y., Taleb-Ahmed, A., 2019. Medical Video Coding Based on 2nd-Generation Wavelets: Performance Evaluation. *Electron* 8 (1), 88. <https://doi.org/10.3390/ELECTRONICS8010088>.
- Wang, Z., Bovik, A.C., 2009. Mean squared error: lot it or leave it? A new look at signal fidelity measures. *IEEE Signal Process Mag* 26, 98–117. <https://doi.org/10.1109/MSP.2008.930649>.
- Horé, A., Ziou, D., 2013. Is there a relationship between peak-signal-to-noise ratio and structural similarity index measure? *IET Image Process* 7 (1), 12–24. <https://doi.org/10.1049/iet-ipr.2012.0489>.