Signal Processing: Image Communication 81 (2020) 115725

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

An effective fragile watermarking scheme for color image tampering detection and self-recovery☆

Javier Molina-Garcia [a], Beatriz P. Garcia-Salgado [a], Volodymyr Ponomaryov [a], Rogelio Reyes-Reyes [a,*], Sergiy Sadovnychiy [b], Clara Cruz-Ramos [a]

[a] Instituto Politecnico Nacional, Santa Ana 1000, ESIME Culhuacan, 04430, Mexico-City, Mexico
[b] Instituto Mexicano del Petroleo, Lazaro Cardenas 152, 07730, Mexico-City, Mexico

ARTICLE INFO

Keywords:
Fragile watermarking
Self-recovery
Image authentication
Image tampering detection
Tampering coincidence problem

ABSTRACT

In this paper, a fragile watermarking scheme for color-image authentication and self-recovery is proposed. Original image is divided into non-overlapping blocks, and for each i-th block, the watermarks used for recovery and authentication are generated, which are embedded into a different block according to an embedding sequence given by a permutation process. The designed scheme embeds the watermarks generated by each block within the 2-LSB, where a bit-adjustment phase is subsequently applied to increase the quality of the watermarked image. In order to increase the quality of the recovered image, we use in the post-processing stage the bilateral filter that efficiently suppresses noise preserving image edges. Additionally, in the tamper detection process high accuracy is achieved employing a hierarchical tamper detection algorithm. Finally, to solve tampering coincidence problem, three recovery watermarks are embedded in different positions to reconstruct a specific block, and a proposed inpainting algorithm is implemented to regenerate those regions affected by this problem. Simulation results demonstrate that the watermarked images appear to demonstrate higher quality, and the proposed novel scheme can reconstruct the alteration of extremely high rates (up to 80%), obtaining good quality for altered regions that are self-recovered with higher visual performance compared with a similar scheme from state of the-art methods.

1. Introduction

The use of digital files such as images has had multiple applications in numerous areas of research as well as in the lives of people, where these images are used mainly to store daily events. Nevertheless, with the rapid advancement in information technologies, some powerful tools have emerged to edit the visual content of images. This can be used for malicious purposes and can affect people and/or companies involved. To address this problem, multiple authentication and recovery techniques based on watermarking have been designed [1–16], where semi-fragile watermarking techniques [1–3] and fragile watermarking techniques [4–16] have been used. Semi-fragile watermarking methods [1–3] have the main feature of reconstructing tampered information using images in the JPEG compression format, allowing a certain degree of robustness against compression, but with a main disadvantage of a reduced reconstruction rate. Otherwise, methods based on fragile watermarking [4–16] demonstrate higher reconstruction rates using uncompressed images than semi-fragile based methods.

In order to carry out the reconstruction process, schemes presented in [4–16] divide the host image into small blocks, where for each block recovery watermarks are generated. Watermark representing the recovery content of a block, also called as digest image, is always embedded into another block for content recovery. Therefore, in this way, the tampered blocks of the image are reconstructed by means of the recovered watermark. Additionally, it is important to perform the tamper detection process, where some authentication schemes based on watermarking have been proposed [1–20]. These schemes are divided mainly in pixel-wise fragile watermarking [17–20] and block-wise fragile watermarking [5–16]. In pixel-wise based watermarking schemes, the authentication watermark is generated for each one of the pixels of an image and is embedded into the host pixels. On the other hand, in block-wise based watermarking schemes, the authentication watermark is generated using sub-blocks of the image. Finally, some self-recovery schemes should perform the authentication by means of the comparison using an objective metric between the image suspected of alteration and the image obtained when extracting the recovery watermark [2–4].

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.image.2019.115725.

* Corresponding author.
E-mail addresses: jmolinag@alumno.ipn.mx (J. Molina-Garcia), bgarcias1404@alumno.ipn.mx (B.P. Garcia-Salgado), vponomar@ipn.mx (V. Ponomaryov), rreyesre@ipn.mx (R. Reyes-Reyes), ssadovny@imp.mx (S. Sadovnychiy), ccruzra@ipn.mx (C. Cruz-Ramos).

https://doi.org/10.1016/j.image.2019.115725
Received 12 July 2019; Received in revised form 27 October 2019; Accepted 26 November 2019
Available online 29 November 2019

One of the main problems in the recovery methods for images is called *tampering coincidence* [5]. This occurs when both a certain block and the region containing its recovery watermark are tampered; therefore, it is impossible to recover the original content in the tampered block. In order to solve this problem, some authors embed two copies of the recovery watermark [4–7]. Consequently, a second chance to reconstruct the tampered region should be applied in case of the deletion of a specific region and its recovery watermark.

Aiming to minimize this vulnerability, in this paper a Block-based Fragile Watermarking scheme for Self-Recovery (denoted as BFW-SR) of tampered regions in color images is proposed, where the recovery of high tampering rates is carried out. In order to minimize negative effects of the tampering coincidence problem, the designed scheme embeds three copies of the recovery watermark into the RGB channels of the host image. Furthermore, three watermarks for tampering detection should be embedded, which are used to authenticate each RGB image channel. A hierarchical tamper detection algorithm is employed to achieve high tamper detection accuracy. Additionally, we propose to use an inpainting process that helps in regenerating the eliminated blocks by the tampering coincidence problem and a bilateral filtering in noise suppression in order to preserve edges and fine details in an image.

The generation of the recovery watermark is performed through two stages: the first stage takes advantage of the compression capacity offered by the YCbCr color space, which is used in the JPEG compression. Each chrominance channel is divided into blocks of 4×4 pixels by extracting 6 bits from each block, consequently, 0.375 bpp (*bits per pixel*) are obtained for each channel. In the second stage, the halftoning technique in form of the Error Diffusion algorithm [21] is used in the luminance channel, preserving fine details and obtaining 1 bpp. According to this, to reconstruct one RGB pixel of the image, 1.75 bpp are required. In order to obtain high quality in the protected image, this novel framework embeds the recovery and authentication watermarks into the 2-LSB of the host image.

In order to evaluate the quality of the results obtained, the PSNR (*Peak Signal-to-Noise Ratio*) and SSIM (*Structural Similarity Index Measure*) metrics are used. Additionally, we employ the recently proposed metric PSNR-HVS-M [22], which takes into account *Contrast Sensitivity Function* (CSF) and between-coefficient contrast masking of DCT basis functions. A point worth mentioning is that the PSNR-HVS-M metric demonstrates good correspondence with subjective perception via HVS *(Human Visual System*), for this reason, PSNR-HVS-M criterion was used for the parameters selection of the designed system.

The rest of the paper is organized as follows. Section 2 presents a review of related works. Section 3 describes the designed self-recovery method. Section 4 shows the analysis of the parameters' selection used. Section 5 presents the experimental results obtained with the proposed framework, and the performance comparisons with some state-of-the-art techniques. Finally, the study's conclusions are shown in Section 6.

2. Related works

The performance of the recovery schemes is commonly expressed through the allowed amount of modifications and the objective quality of both the watermarked image and the reconstructed image, which is typically given as PSNR and SSIM criteria. These performance aspects are connected, and the trade-off between them needs to be properly balanced. Recent research on recovery schemes of tampered images focuses to exploit this balance between quality and tampering rate, measuring the performance according to the method used during the recovery and authentication process. These methods can be divided into two categories: methods applied to color images [1,2] and methods applied to grayscale images [3–13], where the last ones can be used for color images applying the recovery and authentication process to each one of the RGB image channels.

In paper [1] a semi-fragile watermarking scheme for tampered images recovery is proposed. Here the image digest is generated by two watermarks. First, a binary matrix, which contains texture-blocks and smooth-blocks from a host image, is generated. Following, this matrix is applied to generate the second recovery watermark. The halftoning technique is employed for each one of the textured blocks, and the average value is applied for each smooth block from the host image. The embedding process is performed using the *Discrete Wavelet Transform* (DWT) and *Quantization Index Modulation* (QIM). Scheme proposed in [2] performs a method applied to color images, where the generation of image digest is executed in the YCbCr color space in the following fashion. Firstly, each channel is down-sampled, where a halftoning algorithm is applied in the luminance channel, and the *Discrete Cosine Transform* (DCT) compression method is applied in the chrominance channels. The embedding process is performed by DWT using the *Quantization Index Modulation — Dither Modulation* (QIM-DM) algorithm. The recovery watermark for the chrominance channels is embedded into the LH and the HL frequency sub-bands, and the halftoned image (image digest for luminance channel) is embedded into the LL subband. In [3], down-sampling and halftoning methods are performed to obtain the recovery watermark using two different embedding process that employ the QIM method: first with the *Integer Wavelet Transform* (IWT) and finally, with the DCT. Analyzing algorithms presented in [1–3], we can conclude that the halftoning algorithm used to generate the recovery watermark or image digest can obtain 1 bpp for each 8-bit channel. Additionally, such watermark method is a semi-fragile one, where only one version of the recovery watermark is embedded, and this significantly reduces the possibility in the reconstruction of high rate alterations; for example, the framework [2] can only reconstruct up to 10% alteration.

Methods based on fragile watermarks [4–16] tend to support a higher tampering rates than semi-fragile methods, obtaining acceptable quality in watermarked images. The principal embedding method employed in these schemes is based on the *Least Significant Bit* (LSB) approach, and it uses grayscale images. Techniques presented in papers [4–7] allow to embed two versions of the recovery watermark. Kiatpapan and Kondo [4] apply a down sampling with a factor of 0.25 on the host image to obtain the image digest, where a compression rate of 0.5 bpp is obtained for each one. In works [5,6], the recovery watermark is generated by a block-based algorithm, where MSBs are obtained for each block, these schemes perform a compression rate of 1.25 bpp. Additionally, the scheme designed in Ref. [5] applies an inpainting process to solve the tampering coincidence problem, and the algorithm in Ref. [6] implements a new chaotic map-based permutation process for embedding. Fan and Hong [7] perform an improvement of the block-based scheme proposed by Mohammad [23], which employs the SPIHT algorithm to generate the digest image divided in 32×32 blocks, obtaining a compression rate of 0.75 bpp. The main disadvantage of this scheme is that the tampering coincidence problem can significantly affect the recovered bits; therefore, the inverse process of the SPIHT algorithm could be affected, resulting in low quality during reconstruction.

Other schemes that use LSB-based embedding are developed in Refs. [8–13], which embed a single version of the digest image. Dadkhah in [8,9] performs the same digest image as in Refs. [5,6], where the principal difference is the tampering detection process, in which Ref. [8] performs an average-based method using sub-blocks, and [9] uses the SVD-based approach. Schemes presented in Refs. [10–12] embed the recovery watermark only once. The process of generating the digest image consists of dividing the image into 2×2 sub-blocks and applying the DCT to each block. Finally, the bits of the two most important coefficients of each block are extracted. These schemes can generate 2.5 bpp for each 8-bit depth image. Finally, Tai in [13] generates the image digest via IWT coefficients, where a compression rate of 1.75 bpp is obtained for each grayscale image.

Resuming the above presented revision, we can conclude that the main advantage of the schemes presented in [4–7] is that they avoid

the tampering coincidence problem by embedding two versions of the image digest as a watermark, where two different blocks could be used to recover a tampered block of the protected image. Additionally, methods proposed in [8–13] have only one chance to recover the tampered region. For this reason, our novel method focuses on embedding three copies of the recovery watermark while maintaining better quality in both, the watermarked image and reconstructed image, after the recovery process.

According to the aforementioned, the compression rate applied to a color image of the designed scheme and of the state-of-the-art methods is 1.75 bpp for the designed scheme, 1.5 bpp for [4], 3.75 bpp for [5,6,8,9], 2.25 bpp for [7], 7.5 bpp for [10–12], and 5.25 bpp for [13].

The main contribution in proposed BFW-SR can be summarized as follows:

1. **High quality of the watermarked image in comparison with other schemes.** In order to increase the quality of the watermarked image, a bit adjustment process is implemented during embedding that appears to demonstrate an increase in quality criteria.
2. **Proposed framework can obtain better robustness against the tampering coincidence problem than recent state-of-the-art schemes.** A compression method performed in the YCbCr color space generates a highly compressed digest that does not suffer degradation after decompression, which is embedded up to three times in the two LSB of the RGB image differing on state-of-the-art schemes that can embed up to two times the digest image generated.
3. **Tampering detection accuracy.** The tampering detection algorithm is performed in each RGB channel forming three tamper results. Here, a hierarchical tamper detection algorithm is applied, which can avoid false negatives during authentication and outperforms the tampering detection. This results in better manipulation detection.
4. **Better quality in the recovered image after recovery process comparing the proposed framework to other schemes.** Increased quality is performed using inpainting process for the regions affected by the tampering coincidence problem and employing bilateral filter that improves the quality of the extracted watermark. This filter is applied in the luminance channel where noise is suppressed while edges are preserved; additionally, the edge information from Y channel is used for the reconstruction of edges in chrominance channels, resulting in better quality of the recovered image.

3. Designed scheme

Designed BFW-SR is divided into two stages. The first stage consists of digest image generation, authentication watermark generation and watermarks embedding. Fig. 1 exposes the flowchart of the first stage. During the second stage, watermarks extraction, authentication and hierarchical tamper detection, recovery channels generation, post-processing and self-recovery are performed. Fig. 2 shows the flowchart of the second stage.

3.1. Digest image generation

In order to generate the watermarks used for recovery, the following steps are performed.

1. Original image of size $A \times B$ in the RGB color space is transformed to the YCbCr color space, and a compression method or digest image generation technique is applied. Firstly, a halftoning algorithm is used to the Y channel by the Error Diffusion method as it is described by Floyd and Steinberg [21]. This method is applied due to the compression rate offered: for each 8-bit image input a 1-bit image output is obtained giving a compression rate of *1* bpp. As it has been demonstrated in [24], this method provides better preservation of fine details compared with other halftoning schemes [25–29]. Fig. 3 shows the resulted halftoned image, which is called $W1$ of size $A \times B$.
2. The digest images corresponding to the chrominance channels (W_{Cb} and W_{Cr}) of size $1 \times 3AB/8$ are obtained by a block-based method, which is explained in the schemes [5,6,8,9]. This method offers to obtain different compression rates without significantly affecting the quality of the image. Each chrominance channel is divided into non-overlapped blocks of 4×4 pixels. Then, for the ith block the three LSBs are deleted. Afterwards, the average intensity value is obtained. Finally, the 6 MSBs are extracted. The compression rate for each chrominance channel is 0.375 bpp.

According to the mentioned above, the total number of bits generated for the recovery watermark is given by $1.75AB$, where 1.75 is obtained by adding the 1 bpp of the digest luminance image and 0.375 bpp for each digest chrominance images.

3.2. Authentication watermark generation

During the authentication watermark generation, the block-based method shown above is used for each RGB channel, where five MSB are extracted for each one of the non-overlapped blocks of 4×4 pixels. Finally, the XOR operation is applied to five bits using overlapped bits pairs, and four authentication bits are generated for each block (0.25 bpp). Generated watermarks for each RGB channel are called W_{AR}, W_{AG} and W_{AB}, respectively, which are of size $1 \times AB/4$.

The three LSB are deleted because the first two LSB planes are used to embed the recovery and authentication watermarks; and the third LSB plane is used to implement a bit adjustment process shown in the next section. Therefore, if these three LSBs are not eliminated, the authentication process could generate false detections. Fig. 4 shows the process of generating the digest image for the chrominance channels and the authentication bits for each RGB channel. It can be observed that, for this example, the authentication bits are 0110.

3.3. Watermarks embedding

The embedding process is done by modifying the 2-LSB of each RGB channel of the image to be protected. A total number of 1.75 bpp are generated for the recovery watermark, and 0.25 bpp per each one of the RGB channels of the image are computed for the authentication watermark. The process for embedding the watermarks is as follows.

1. Watermark $W1$ (halftoned luminance channel) is divided into non-overlapped blocks of 4×4, which are permuted by key_R.
2. Watermarks W_{Cb} and W_{Cr}, which are represented as a binary vector of size $3AB/8$, are divided into non-overlapped 6-bit blocks obtaining a total of $AB/16$ blocks for each vector, and are permuted by the key_R.
3. Watermark W_{AR} of size $AB/4$, is divided into non-overlapped 4-bit blocks, obtaining a total of $AB/16$ blocks.
4. The watermarks blocks of W_{Cb}, W_{Cr} and W_{AR} are concatenated as follows $W2_{R_i} = \left[W_{Cb_i}, W_{Cr_i}, W_{AR_i}\right]$, where $1 \le i \le AB/16$.
5. The R channel of the image to be protected is divided into non-overlapped blocks of 4×4 pixels, where the information of the ith block of $W1$ and $W2_R$ are embedded into the first and second LSB of the ith block of R, respectively.

This process is performed again permuting the blocks of $W1$, W_{Cb} and W_{Cr} by key_G. Then, $W2_G$ is generated using $\left[W_{Cb_i}, W_{Cr_i}, W_{AG_i}\right]$. Finally, the ith block of $W1$ and $W2_G$ are embedded respectively into the first and second LSB of the ith block of G channel. This process is

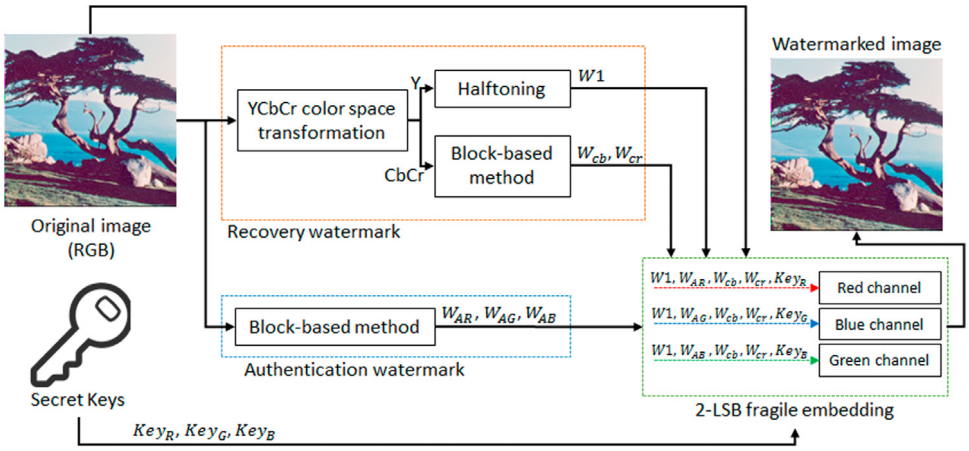

Fig. 1. Flowchart for extraction, tamper detection and recovery process.

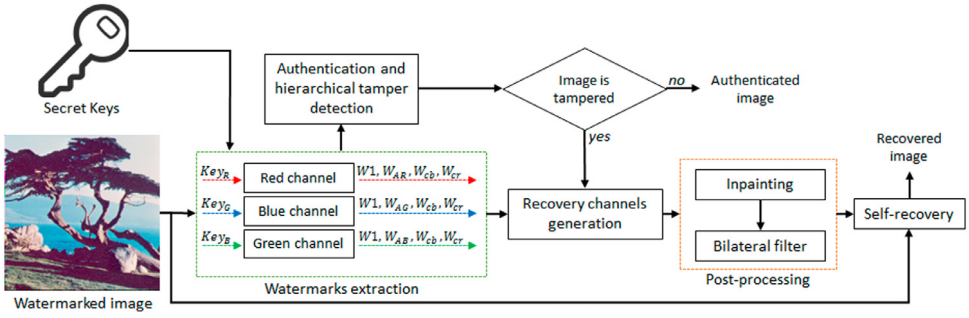

Fig. 2. Flowchart for recovery and authentication watermarks generation and embedding.

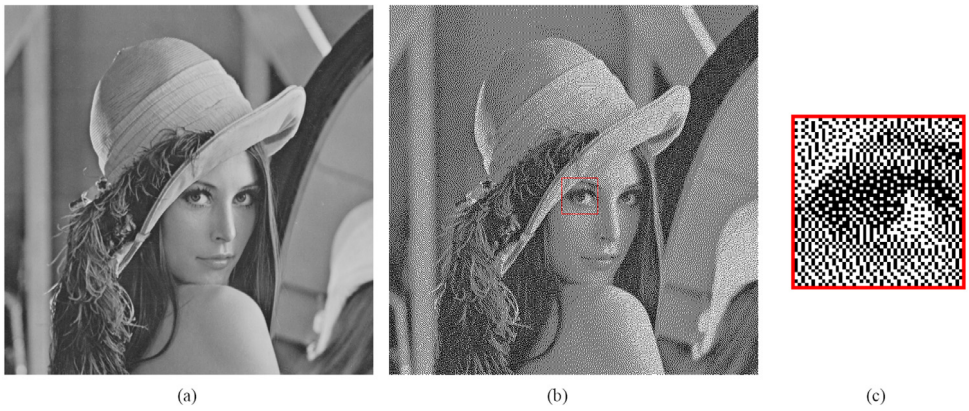

Fig. 3. *Halftoning* process. (a) Grayscale image, (b) Halftoned image, (c) Region of the halftoned image.

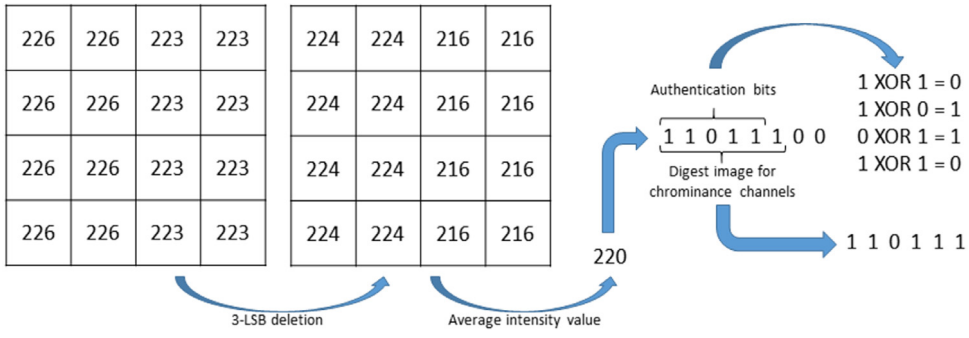

Fig. 4. Generation of authentication bits and digest image for chrominance channels.

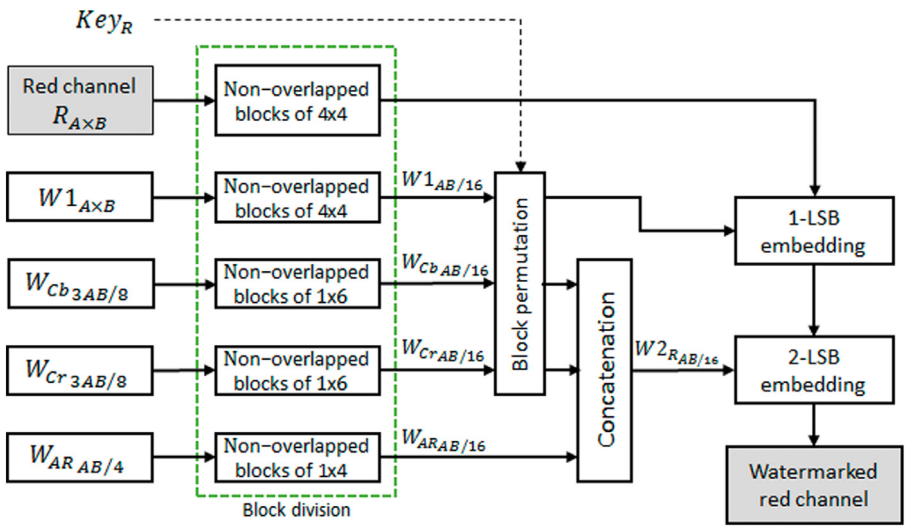

Fig. 5. Watermarks embedding using the red channel of host image.

repeated to embed the watermarks in the blue channel using key_B and W_{AB}. Fig. 5 shows the embedding process for the red channel.

After embedding the authentication and recovery watermarks, in order to minimize the difference between the original image and the watermarked image, the bit adjustment process is performed in each RGB channel of the watermarked image. With this adjustment, the quality of the watermarked image is improved. The following equation is applied to perform the bit adjustment of the watermarked image:

$$I_w(x) = \begin{cases} I(x) - 1, & \text{if } \upsilon \text{ is 3 and 3LSB}(I(x)) \text{ is 1} \\ I(x) + 1, & \text{if } \upsilon \text{ is } -3 \text{ and 3LSB}(I(x)) \text{ is 0,} \end{cases} \quad (1)$$

where υ is the difference between the watermarked pixel and the original pixel before embedding the watermark, $I(x)$ is the original pixel before embedding the watermark, and $I_w(x)$ is the watermarked pixel.

Take for instance an original pixel with the value 11000111_2 (199_{10}). After embedding the watermark, its value changes into 11000100_2 (196_{10}). Therefore, the difference is -3, and the third LSB of the original pixel is 1. Consequently, the value of the final pixel marked is $199_{10} + 1_{10} = 200_{10}$ (11001000_2). Note that the value of the watermark is preserved, and the difference between the original pixel and the watermarked pixel is now one.

3.4. Watermarks extraction

The following process is applied to each RGB channel of the watermarked image.

1. The watermarked-processed channel is divided into non-overlapped blocks of 4 × 4 pixels.
2. For each block, the extraction process 2-LSB is applied, and a total of $AB/16$ blocks are obtained.
3. Extracted blocks are inverse-permuted using the assigned key (key_R, key_G or key_B), where:
   - i Blocks extracted from the first LSB show the halftone-image watermark, which is shown in Fig. 3b.
   - ii Blocks extracted from the second LSB show the chrominance and authentication watermarks.
4. Watermarks corresponding to W_{Cb}, W_{Cr}, and the authentication watermark are obtained according to the concatenation process explained in Section 3.3. Each non-overlapped block of 4 × 4 contains 12 bits for Cb and Cr channels and four bits to authenticate the block.
   - i The binary to decimal conversion is applied using from bit one to six. the resulting value is assigned to an nth block of 4 × 4 of the chrominance channel W_{Cb}.
   - ii The binary to decimal conversion is applied using from bit seven to twelve, the resulting value is assigned to an nth block of 4 × 4 of the chrominance channel W_{Cr}.

Fig. 6 Presents the chrominance images for Lenna generated after the extraction process. At the end of this process, three copies of the YCbCr channels are obtained, and authentication bits are obtained for each block.

3.5. Authentication and hierarchical tamper detection

The following process is performed for each RGB channel during the authentication stage:

1. The method shown in Section 3.2 is applied to each channel of the suspected image of tampering.
2. The four bits generated for the ith block are compared to the authentication watermark extracted from the same block.
3. If any compared bit is not equal, then the whole block is considered manipulated (marked with bit 1), otherwise it is considered authentic (marked with bit 0).

At the end of this process, three binary authentication images are obtained, one for each RGB channel. Each authentication image is divided into non-overlapped blocks of 4 × 4, where a new image is generated according to the following equation:

$$A_{i,j} = AR_{i,j} + AG_{i,j} + AB_{i,j}, 1 \le i \le A/4, 1 \le j \le B/4, \quad (2)$$

where AR, AG and AB denote the authentication image generated from each RGB channel; A, B represent the total rows and columns of the image, and + represents OR operation. Finally, a hierarchical process is applied to improve the accuracy of detection of alterations using the following steps. The authentication image is down-sampled by a factor of 0.5 using the nearest-neighbor method.

1. The resulted image is divided into overlapped blocks of 3 × 3 pixels and the following process is applied to each block.
2. If the block in position 2,2 and at least two of its neighbor pixels are marked as tampered, then the completed block is marked as tampered with.
3. After the processing of all the blocks, an interpolation with a factor of two is applied to the image using the nearest neighbor method.

Fig. 6. Generated images for each a chrominance channel.

3.6. Recovery channels generation

During this stage, a single image of each YCbCr channel is generated from the extracted copies as watermark, and an image containing the blocks affected by the tampering coincidence problem is generated.

1. The authentication image generated in previous section is divided into non-overlapped blocks of 4 × 4 pixels.
2. The blocks are permuted using the user keys: key_R, key_G and key_B. Then, three binary images are obtained, and each binary image represents the extraction errors for each RGB channel of the watermarked image. For example, the binary image generated by key_R represents the erroneously extracted blocks of the watermarks from the red channel.
3. Taking three watermarks corresponding to the luminance channel extracted from each RGB channels, a single luminance image is obtained by analyzing the images generated by the keys: key_R, key_G and key_B. This step is explained in Algorithm 1.

Algorithm 1 Image generation

Input: Rows **R**; Columns **C**; Images to be processed **MatR**, **MatG** and **MatB**; Binary images **AutentR**, **AutentG**, **AutentB**

$MatOut = MatR$
For $i = 1$ **to** R **do**
  For $j = 1$ **to** C **do**
    If $AutentR_{i,j} == 1$**then**
      If $AutentG_{i,j} == 1$**then**
        If $AutentB_{i,j} == 0$
          $MatOut_{i,j} = MatB_{i,j}$
        End if
      Else
        $MatOut_{i,j} = MatG_{i,j}$
      End if
    End if
  End for
End for

Output: Single channel **MatOut**

Finally, this process is performed with the three copies of each one of the chrominance channels extracted from the RGB channels of the image. If the three binary images show an extraction error in the same ith block, then this region is affected by the tampering coincidence problem, that is, $TCP_{i,j} = AutentR_{i,j} * AutentG_{i,j} * AutentB_{i,j}, 1 \le i \le A, 1 \le j \le B$, where * represents AND operation, and A, B represent the total rows and columns of the image. Fig. 7 explains the processes shown in Sections 3.5 and 3.6.

3.7. Post-processing

After carrying out the process shown in Section 3.6, three images corresponding to each channel of the YCbCr color space of the image to be reconstructed are generated. Additionally, a binary image is computed, which shows the regions that cannot be recovered due to the tampering coincidence problem.

The image corresponding to the luminance channel is represented by halftoning, which is shown in Fig. 3(b). Gaussian filter is applied to this image [30,31]. This process applied to halftoned images generates grayscale images. In this manner, the luminance channel with depth of eight bits is obtained. In order to improve the quality of each YCbCr channel, an inpainting procedure is used (algorithm 2) to fill the regions affected by the tampering coincidence problem. In algorithm 2, *Io* is the single image channel (Y, Cb or Cr), and *TCP* corresponds to the binary image generated in Section 3.6.

Algorithm 2 Inpainting

Input: Rows **R**; Columns **C**; Image to be processed **Io**; Binary image **TCP**

$Iout = Io$
$TCPout = TCP$
For $i = 2$**to**$R - 1$**do**
  For $j = 2$**to**$C - 1$**do**
    $wTCP = TCP_{i-1:i+1,j-1:j+1}$
    If $wTCP_{2,2} == 1$**then**

$$tot = \sum_{i=1}^{3}\sum_{j=1}^{3}(1 - wTCP_{i,j})$$

      If $tot > 1$**then**
        $wIo = Io_{i-1:i+1,j-1:j+1}$
        $Imaux = wIo.*(1 - wTCP)$

$$Iout_{i,j} = \left(\sum_{i=1}^{3}\sum_{j=1}^{3} Imaux_{i,j}\right) \Big/ tot$$

        $TCPout_{i,j} = 0$
      End if
    End if
  End for
End for

Output: Image after inpainting process **Iout**, Binary image **TCPout**

The presented inpainting process fills the missing regions with the average value of neighbor pixels that have been extracted correctly; this algorithm is repeated while the *TCPout* output image contains non-zero values given the *Iout* and *TCPout* images as input parameters.

After implementing the inpainting algorithm, it is important to improve the quality of the image used for recovery. As previously observed, the generated chrominance images (Fig. 6) are affected by a pixelated effect, which decreases the quality of the image obtained in the RGB color space. Additionally, when applying the Gaussian filter during the inverse-halftoning phase, the generated image tends to be noisy due to the isolation of white or black dots. To avoid this, a bilateral filter proposed in [32–34] is employed in the form of algorithm SHIFTABLE-BF explained in [35]. The following steps are performed in bilateral filtering implementation:

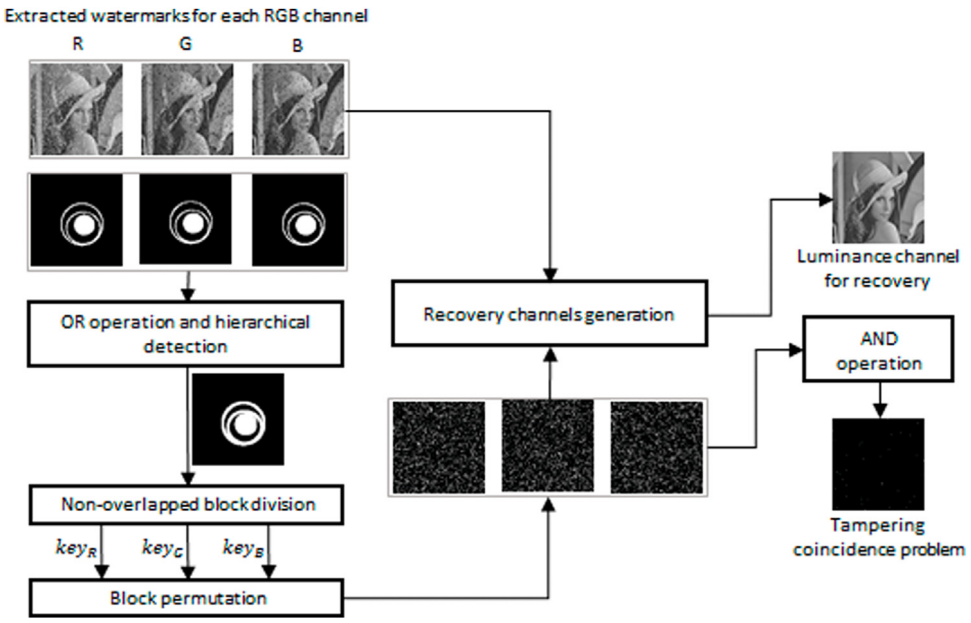

Fig. 7. Authentication and hierarchical tamper detection process and recovery channels generation.

1. SHIFTABLE-BF algorithm is applied to luminance channel generated after the Gaussian filtering implementation.
2. SHIFTABLE-BF algorithm is applied to each one of the chrominance channels. The range filter is used to the generated luminance channel in step 1, and the spatial filter is used in each chrominance channels, where this process eliminates the pixelated effect.
3. Generated images are transformed to the RGB color space.

3.8. Recovery

Taking the image in the RGB color space generated according to procedure explained in the previous section and taking the binary authentication image generated in Section 3.5. The tampered regions of the image suspected of alteration are recovered using the following equation: $I_{RECOVERED} = I_{TAMPERED} \cdot * \left(1 - I_{AUTENT}\right) + I_{RECOVERY} \cdot * I_{AUTENT}$, where $I_{TAMPERED}$ represents the image suspected of alteration, $I_{RECOVERY}$ is the recovery image generated in the previous section, I_{AUTENT} corresponds to the authentication image generated in Section 3.5, and $. *$ implies dot product.

4. Parameters selection

The designed scheme extracts three copies of each watermark, which are images used to recover the tampered regions of the suspicious image of tampering. In order to obtain the optimal parameters of each filter, several experiments have been carried out using different test images. Furthermore, different objective quality metrics were analyzed: PSNR, SSIM and PSNR-HVS-M. Parameters for each filter were selected taking into account the PSNR-HVS-M quality metric because, according to [22], this measure can obtain better performance compared to other objective metrics. The first filtering process uses Gaussian filter [30] that is applied to the halftone-luminance watermark, where a grayscale image is obtained. Fig. 8 shows the average results obtained for each quality metric using different values of σ and filter size for the Gaussian filter generation.

According to the results shown in Fig. 8, each quality metric shows a different behavior depending on σ value and filter size used. The proposed parameters for the Gaussian filter utilized in the designed framework are $\sigma = 1.1$ and a filter size of 1×5, resulting in criteria values 29.02 dB, 0.7401 and 27.99 dB for PSNR, SSIM, and PSNR-HVS-M, respectively. Subsequently, the bilateral filter is applied in the luminance-channel. As mentioned above, the bilateral filter uses two filters called spatial filter ($g_{\sigma_s}(x)$) and range filter ($g_{\sigma_r}(x)$). Therefore, the similar tests were performed using different values of σ_s and σ_r. Fig. 9 shows the average results when implementing the bilateral filter to the same set of test images.

The best results in the PSNR-HVS-M quality metric have been obtained utilizing values $\sigma_s = 2.5$ and $\sigma_r = 7.0$, where an average value of 28.55 dB is obtained. Additionally, the PSNR and SSIM values for $\sigma_r = 7.0$, are 29.78 dB and 0.8151, respectively. Table 1 shows the quality for the halftone images, for the images after applying the Gaussian filter, and for the images after bilateral filtering, comparing them with the original image before applying the halftone method.

Finally, the range filter is implemented to the previously-obtained luminance channel to detect edges and to control the implementation of the spatial filter, which is applied to each chrominance channel. To choose the optimal values of σ_s and σ_r, the averaged values of the objective quality metrics for different processed channels were calculated. Fig. 10 shows the average results when the bilateral filter is applied in the chrominance channel for the same set of test images used previously.

Since the implementation of the filter is employed to improve the edges of the chrominance images, the parameters to be used are different, as one can see in Fig. 10. The following values $\sigma_s = 2.0$ and $\sigma_r = 5.0$, demonstrate the optimal results in terms of the PSNR-HVS-M quality metric where an average value of 33.12 dB is obtained in terms of PSNR-HVS-M criterion. Additionally, the average quality values 35.33 dB and 0.8732 are obtained for the quality metrics PSNR and SSIM, respectively. Table 2 shows the average quality of the extracted chrominance images and the average quality of these images after applying the bilateral filter. Fig. 11 illustrates the chrominance images for Lenna generated after filtering, where it can be seen better performance in the images in comparison with those presented in Fig. 6; in particularly, better reconstruction of the edges can be observed analyzing Figs. 11c and d.

5. Experimental results and analysis

In the experiments carried out, eight 512×512 standard images "Lenna", "Sailboat", "Peppers", "Baboon", "House", "Splash",

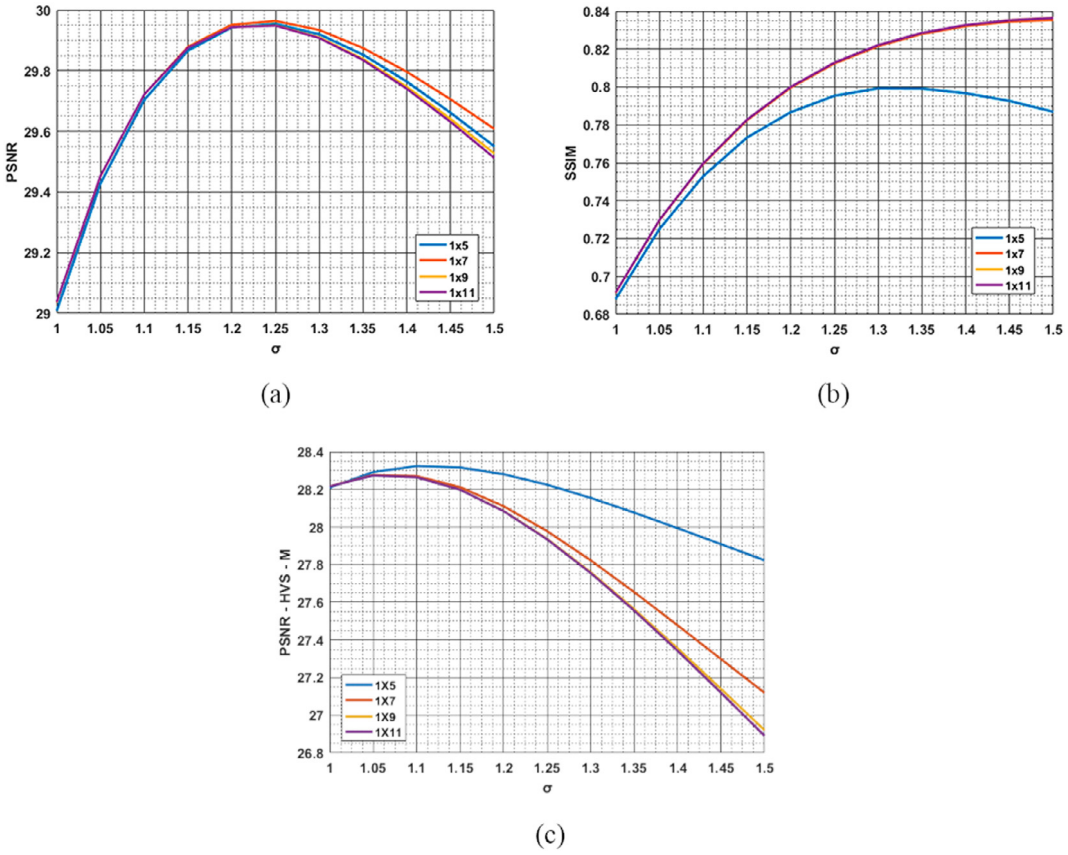

Fig. 8. Performance of the luminance image obtained after applying the Gaussian filter. (a) PSNR, (b) SSIM, (c) PSNR-HVS-M.

Table 1
PSNR (dB), SSIM and PSNR-HVS-M (dB) values for the halftoned images and filtered images.

		Lenna	Sailboat	Peppers	Baboon	House	Splash	Tiffany	F-16	Average
Halftoned images	PSNR	6.52	7.01	6.67	6.48	6.76	6.75	7.84	7.06	6.88
	SSIM	0.0241	0.0438	0.0255	0.0738	0.0355	0.0162	0.0211	0.0312	0.0339
	PSNR-HVS-M	16.95	17.23	17.06	17.06	17.01	16.87	17.02	16.87	17.00
Gaussian filter	PSNR	31.20	28.50	29.63	24.34	28.25	30.98	29.74	29.59	29.02
	SSIM	0.7710	0.7521	0.7394	0.6511	0.7711	0.7438	0.7122	0.7808	0.7401
	PSNR-HVS-M	30.00	27.64	28.08	25.75	27.02	28.95	28.58	27.96	27.99
Gaussian + Bilateral filter	PSNR	32.30	28.94	30.56	24.20	28.53	32.58	30.76	30.37	29.78
	SSIM	0.8521	0.8100	0.8293	0.6399	0.8302	0.8669	0.8137	0.8790	0.8151
	PSNR-HVS-M	31.02	28.00	28.82	25.29	27.13	30.09	29.50	28.57	28.55

Table 2
PSNR (dB), SSIM and PSNR-HVS-M (dB) values for the extracted chrominance images and filtered images.

		Lenna	Sailboat	Peppers	Baboon	House	Splash	Tiffany	F-16	Average
Extracted chrominance images	PSNR	38.58	33.00	33.48	31.16	34.30	35.77	34.77	36.41	34.68
	SSIM	0.9039	0.8044	0.8634	0.6757	0.8740	0.9385	0.8904	0.9413	0.8614
	PSNR-HVS-M	36.85	31.45	30.78	30.48	31.44	32.21	31.82	32.68	32.21
Filtered images	PSNR	39.24	33.52	34.50	31.49	34.63	36.86	35.19	37.21	35.33
	SSIM	0.9135	0.8187	0.8842	0.6794	0.8819	0.9552	0.8999	0.9532	0.8732
	PSNR-HVS-M	37.93	32.31	32.27	31.05	31.71	33.75	32.35	33.61	33.12

"Tiffany" and "F-16" shown in Fig. 12 have been used. The quality metrics used are the PSNR, SSIM, and the PSNR-HVS-M [22]. The optimal parameters chosen in the previous section are: in the filtering phase $\sigma = 1.1$ and a filter size of 1×5 for Gaussian filter generation applied to the halftoned image; in the bilateral filtering applied in luminance channel after the Gaussian filtering, values $\sigma_s = 2.5$ and $\sigma_r = 7.0$; in the bilateral filtering applied in the chrominance channels, values $\sigma_s = 2.0$ and $\sigma_r = 5.0$.

5.1. Quality of watermarked images

As mentioned above, the designed method performs the watermarks embedding using the 2-LSB algorithm. It performs a bit adjustment phase in order to minimize the difference between the watermarked image and the original image. Table 3 summaries objective criteria values for the watermarked images in cases with and without the bit adjustment applied where one can see improvement for each watermarked image when the bit adjustment process is applied.

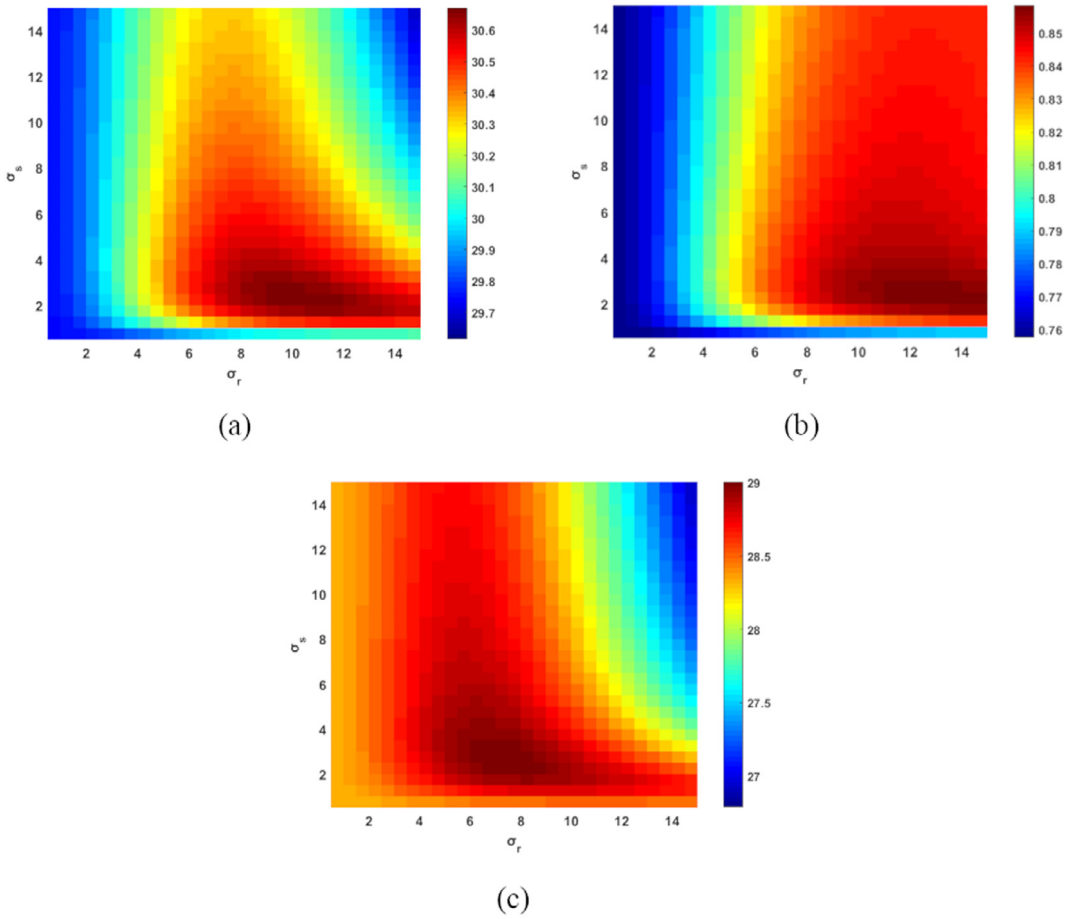

Fig. 9. Performance obtained in the luminance channel after applying the bilateral filter. (a) PSNR, (b) SSIM, (c) PSNR-HVS-M.

Table 3
Criteria values: PSNR (dB), SSIM and PSNR-HVS-M (dB) for watermarked images.

		Lenna	Sailboat	Peppers	Baboon	House	Splash	Tiffany	F-16	Average
Without bit adjustment	PSNR	44.13	44.13	44.10	44.15	44.09	44.09	44.27	44.00	44.12
	SSIM	0.9820	0.9870	0.9799	0.9941	0.9852	0.9712	0.9831	0.9790	0.9826
	PSNR-HVS-M	50.79	51.48	51.90	55.97	50.17	47.49	45.07	48.01	50.11
With bit adjustment	PSNR	44.60	44.61	44.54	44.64	44.66	44.47	44.87	44.69	44.63
	SSIM	0.9840	0.9884	0.9816	0.9947	0.9834	0.9737	0.9846	0.9812	0.9839
	PSNR-HVS-M	50.94	52.18	52.00	55.84	51.48	47.60	45.56	49.86	50.68

Tables 4–6 show the performance for the watermarked images in terms of PSNR, SSIM, and PSNR-HVS-M, respectively, comparing the results with state-of-the-art schemes. As it can be observed, the designed scheme appears to demonstrate higher performance in terms of PSNR and SSIM for all of the test images used. However, in terms of PSNR-HVS-M, the quality of the watermarked images remains superior in most of the images. In addition, the average performance for all tests is superior in comparison with all analyzed schemes, maintaining an average of 44.63 dB, 0.9839 and 50.68 dB for the PSNR, SSIM and PSNR-HVS-M metrics, respectively.

5.2. Tamper detection analysis

In order to test the manipulation detection capacity of the designed scheme, numerous modifications were introduced in the different images, using the inpainting method proposed in [36] to the Lenna, House and F-16 images. The tampering rates for each image are: 6.8% for Lenna, 12.1% for House, 13.1% for F-16 and 87.3% for Tiffany. Figs. 13–16 demonstrate the images generated during the tampering detection process for Lenna, House and F-16 test images, respectively. For all of these figures, subfigure (a) illustrates the original image without alteration, (b) shows the tampered image using [36], (c) displays the ground truth of the authentication image, (d) gives the authentication image generated from the red channel, (e) corresponds to the authentication image generated from the green channel, (f) presents the authentication image generated from the blue channel, and (g) exemplifies the authentication image generated after OR operation between images (c), (d) and (e). Finally, subfigure (h) represents the authentication image generated after hierarchical detection.

As one can summarize, the authentication images of each RGB channel, shown in subfigures (d), (e) and (f) from Figs. 13, 14, 15, and 16 present extraction errors, which are mostly eliminated using the OR operation. Later, the hierarchical authentication method is applied to the resulting images, where the majority of detection errors are eliminated, mainly, false negatives. To measure the ability to perform the correct detection of tampered regions, the *precision* and *recall* performance measures presented in [13] were used. The first one employs the true and false positive detections for the tampered area denoted as TP and FP, the second one also utilizes TP and the false negative detection FN. The precision criterion corresponds to the accuracy of

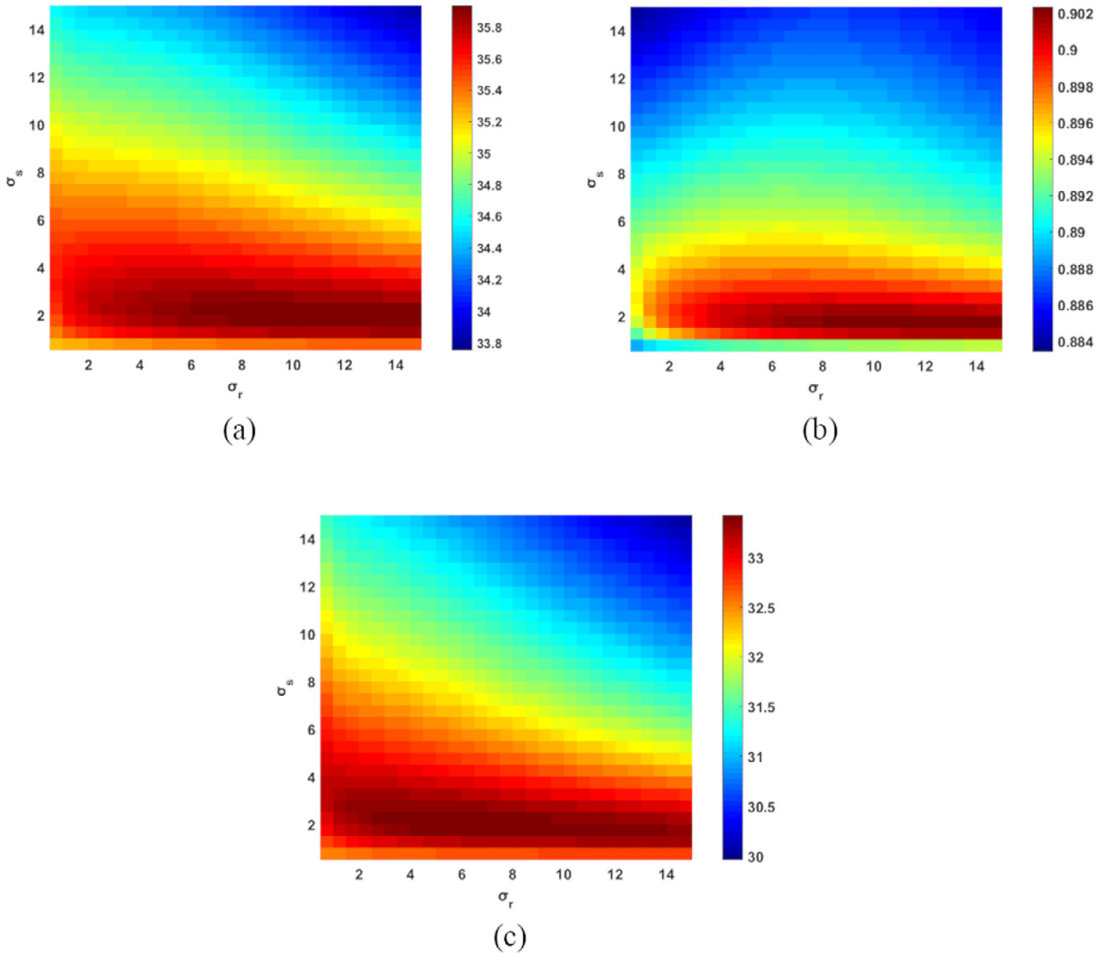

Fig. 10. Average performance obtained in the chrominance images after applying the bilateral filter. (a) PSNR, (b) SSIM, (c) PSNR-HVS-M.

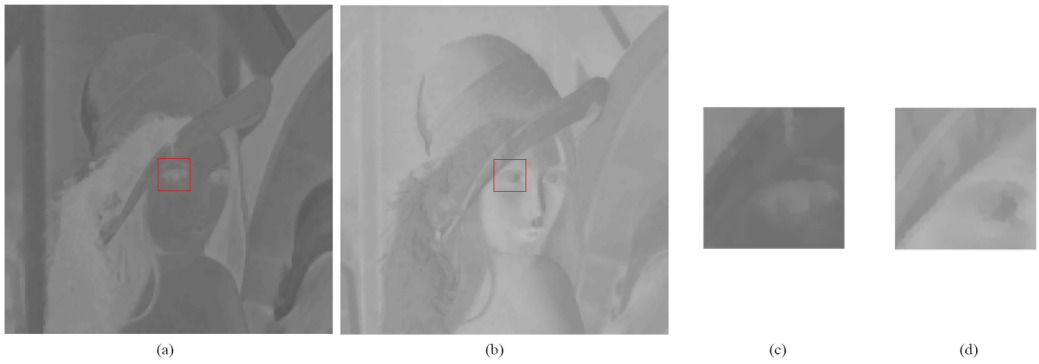

Fig. 11. Generated images for each a chrominance channel after filtering.

Table 4
PSNR (dB) values for the watermarked images.

	Lenna	Sailboat	Peppers	Baboon	House	Splash	Tiffany	F-16	Average
BFW-SR	**44.60**	**44.61**	**44.54**	**44.64**	**44.66**	**44.47**	**44.87**	**44.69**	**44.63**
Singh [10]	37.90	37.90	37.79	37.90	37.88	37.84	37.44	37.88	37.81
Dadkhah [9]	44.13	44.12	44.06	44.14	44.19	44.08	43.85	44.12	44.08
Tong [6]	37.90	37.90	37.79	37.90	37.88	37.84	37.44	37.88	37.81
Fan [7]	44.13	44.10	44.06	44.12	44.18	44.08	43.84	44.11	44.07
Tai [13]	44.12	44.11	44.06	44.14	44.18	44.09	43.85	44.12	44.08

the detection of a specific tampered region, and it is defined as follows: $Precision = TP/(TP + FP)$. The recall criterion is a measure of how many truly relevant results are given, and it is calculated as follows: $Recall = TP/(TP + FN)$.

Fig. 12. Set of test images used, (a) Lenna, (b) Sailboat, (c) Peppers, (d) Baboon, (e) House, (f) Splash, (g) Tiffany, (h) F-16.

Table 5
SSIM values for the watermarked images.

	Lenna	Sailboat	Peppers	Baboon	House	Splash	Tiffany	F-16	Average
Proposed	**0.9840**	**0.9884**	**0.9816**	**0.9947**	**0.9834**	**0.9737**	**0.9846**	**0.9812**	**0.9839**
Singh [10]	0.9307	0.9493	0.9234	0.9763	0.9319	0.8942	0.9246	0.9194	0.9312
Dadkhah [9]	0.9820	0.9868	0.9791	0.9941	0.9815	0.9695	0.9806	0.9782	0.9814
Tong [6]	0.9307	0.9494	0.9234	0.9763	0.9319	0.8942	0.9246	0.9194	0.9312
Fan [7]	0.9820	0.9867	0.9791	0.9941	0.9815	0.9695	0.9804	0.9781	0.9814
Tai [13]	0.9820	0.9868	0.9791	0.9941	0.9815	0.9696	0.9805	0.9781	0.9814

Table 6
PSNR-HVS-M (dB) values for the watermarked images.

	Lenna	Sailboat	Peppers	Baboon	House	Splash	Tiffany	F-16	Average
BFW-SR	**50.94**	52.18	**52.00**	**55.84**	**51.48**	47.60	**45.56**	49.86	**50.68**
Singh [10]	41.78	43.72	42.99	46.15	42.26	39.35	35.61	41.55	41.67
Dadkhah [9]	50.35	52.61	51.78	55.13	50.57	47.63	43.17	50.13	50.17
Tong [6]	41.90	43.76	42.99	46.15	42.26	39.35	35.61	41.55	41.69
Fan [7]	50.36	52.62	51.75	55.17	50.47	**47.72**	43.17	50.13	50.17
Tai [13]	50.40	**52.72**	51.69	55.17	50.58	47.67	43.15	**50.18**	50.19

In order to verify the efficiency of the proposed authentication method, some authentication and reconstruction tests were performed using OR method and hierarchical method. Table 7 shows the performance of the designed authentication system for two variants: the authentication method using OR operation and the authentication method using the OR operation with the hierarchical method. The second variant appears to demonstrate better *recall* measure rates, while the first one leads to better precision performance.

According to our experiments, the best results during the reconstruction process are given when the recall measure is close to one. Table 8 presents the results ensued when reconstructing the modified regions in the images used in Figs. 13–16. As mentioned above, the performance increases when the recall measure is close to one.

The tamper detection analysis for different tampering rates is shown in Table 9, where the average precision and recall measures for different state-of-the-art techniques are shown.

According to the results obtained, the methods [6,10] have got low recall. This can be explained in such way: they use only two bits to detect alterations in each block, consequently, there exists the probability that these bits can be extracted incorrectly, generating errors during the authentication. On the other hand, the scheme [13] employs four bits to authenticate each block of the image, resulting in better performance for recall and precision measures, decreasing errors during authentication. Finally, other frameworks presented in [7] and [9] use respectively 16 bits and 12 bits to authenticate each block of the image, demonstrating better performance in the recall measure with value of 1. It is worth mentioning that, for these methods, the amount of authentication bits is significantly high, which prevents embedding multiple copies of recovery watermark and, therefore, do not avoid the tampering coincidence problem. Nevertheless, the designed scheme uses only four bits of authentication for each block of the image, which allows embedding copies of the recovery watermark, and additionally, high values in recall and precision measures can be obtained after the tampering hierarchical detection.

5.3. Performance of inpainting process

In this stage, the performance of the designed inpainting method is evaluated to eliminate the regions affected by the tampering coincidence problem. During the tests carried out, tampered images shown in Figs. 13–16 were used, and the reconstruction process was implemented in two different ways. The first process applies the proposed inpainting algorithm, and the second one does not use this algorithm. Fig. 17 shows the reconstructed images for each implementation, where (a), (e), (i) and (m) show the original images; (b), (f), (j), and (n) represent the original regions: (c), (g), (k), and (o) show the recovered

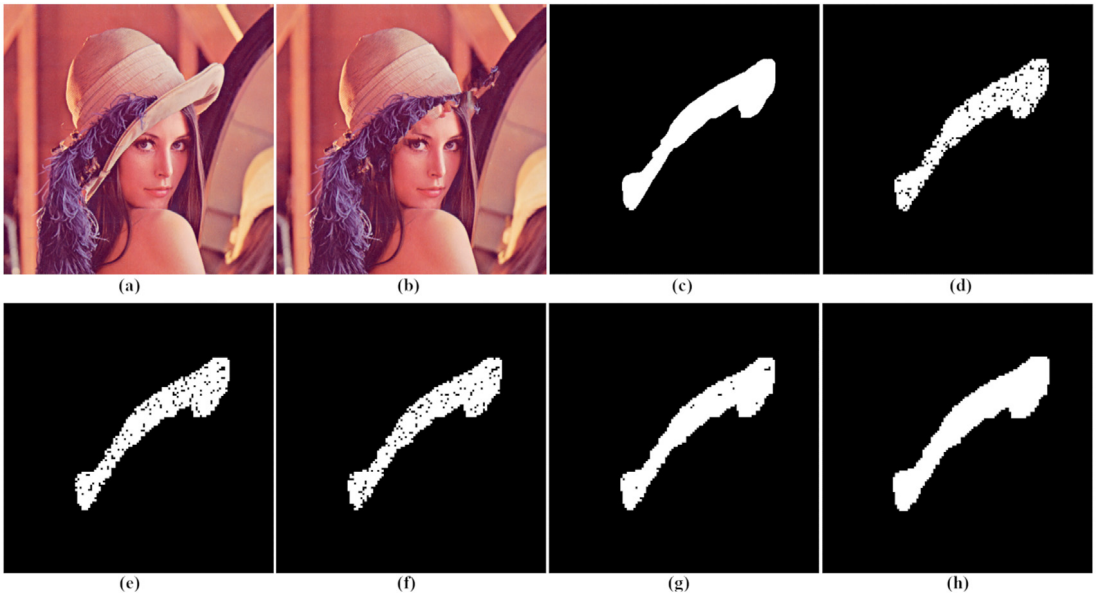

Fig. 13. Tampering detection. (a) Original image "Lenna". (b) Tampered image. (c) Authentication ground truth. (d) Authentication for red channel. (e) Authentication for green channel. (f) Authentication for blue channel. (g) OR operation. (h) Hierarchical authentication.

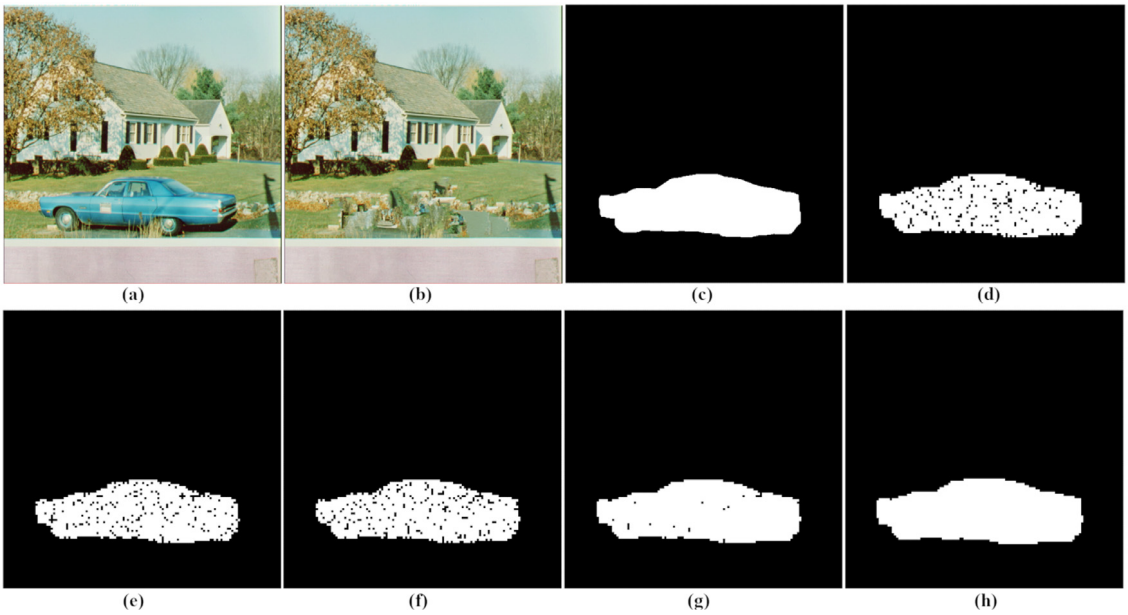

Fig. 14. Tampering detection. (a) Original image "House". (b) Tampered image. (c) Authentication ground truth. (d) Authentication for red channel. (e) Authentication for green channel. (f) Authentication for blue channel. (g) OR operation. (h) Hierarchical authentication.

Table 7
Tamper detection analysis.

	OR authentication		OR + Hierarchical authentication	
	Precision	Recall	Precision	Recall
Lenna	0.9338	0.9786	0.8356	0.9999
House	0.9604	0.9888	0.9060	1
F-16	0.9577	0.9800	0.8883	1
Tiffany	0.9935	0.9763	0.9839	0.9989

regions without the use of the inpainting process; and (d), (h), (l), and (p) show the recovered regions after the application of the inpainting process.

Because the alterations in each image imply less than 15% of the total image size in Fig. 17(c), (g) and (k), the tampering coincidence problem does not significantly affect the reconstructed images. However, for Tiffany image, due to the tampering rate that is higher than 80%, the tampering coincidence problem is more remarkable, and when the inpainting process is employed, the performance is better. Table 10 shows the quality of each reconstructed image, for the images used in Fig. 17. Here, it can be observed that the quality increases. In

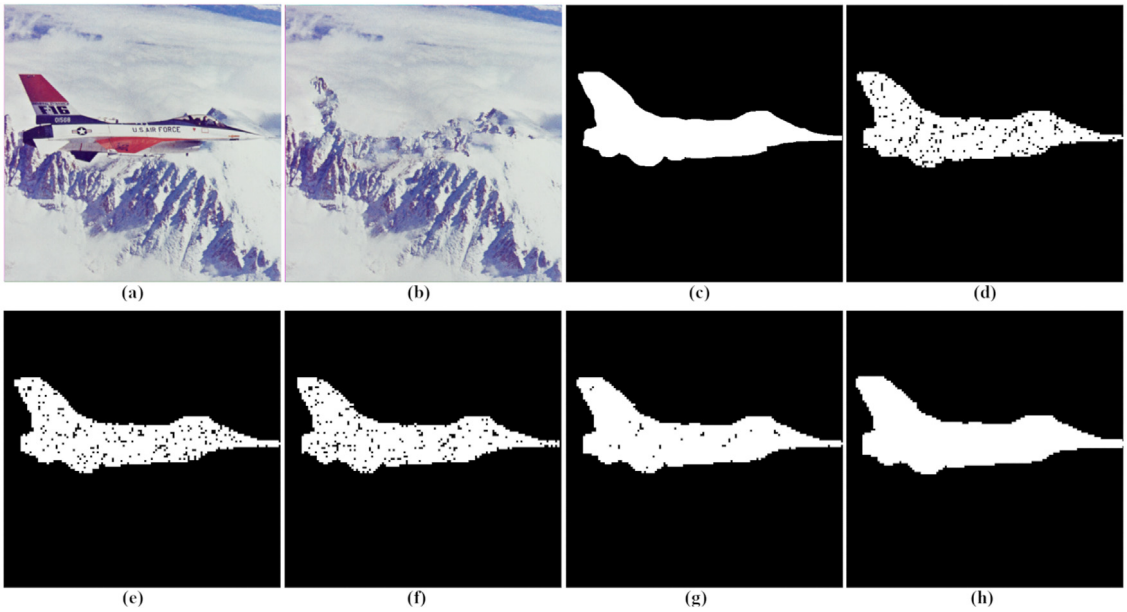

Fig. 15. Tampering detection. (a) Original image "F-16". (b) Tampered image. (c) Authentication ground truth. (d) Authentication for red channel. (e) Authentication for green channel. (f) Authentication for blue channel. (g) OR operation. (h) Hierarchical authentication.

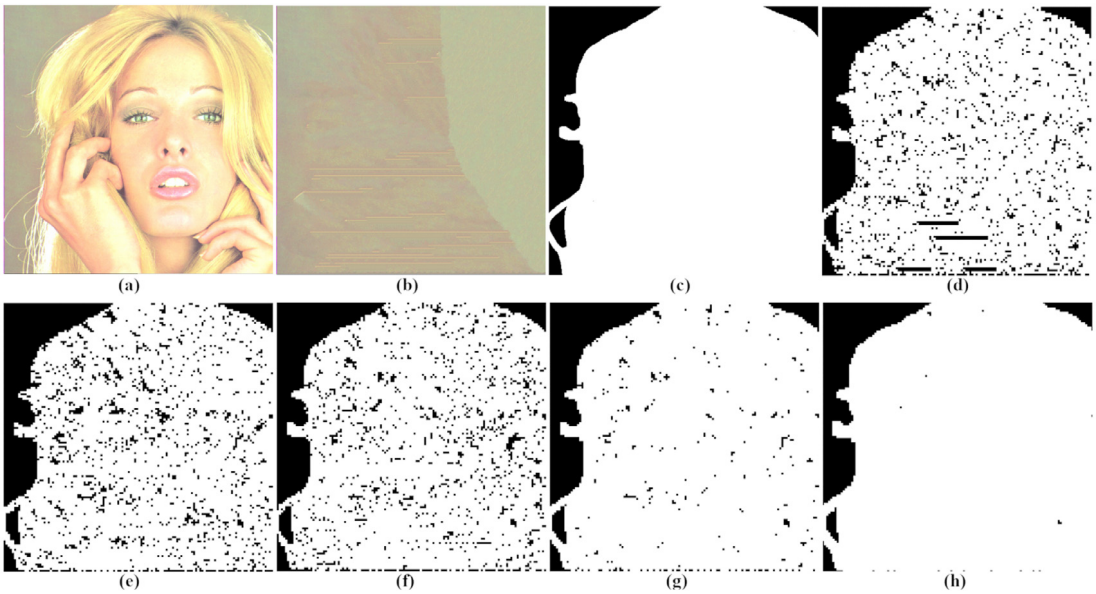

Fig. 16. Tampering detection. (a) Original image "Tiffany". (b) Tampered image. (c) Authentication ground truth. (d) Authentication for red channel. (e) Authentication for green channel. (f) Authentication for blue channel. (g) OR operation. (h) Hierarchical authentication.

Table 8
Performance in terms of PSNR (dB)/SSIM/PSNR-HVS-M (dB) for the reconstructed images using different both authentication schemes.

	OR authentication	OR + Hierarchical authentication
Lenna	36.38/0.9818/34.27	37.43/0.9811/37.10
House	33.73/0.9668/28.74	34.95/0.9678/30.70
F-16	32.29/0.9708/29.31	34.49/0.9729/33.04
Tiffany	18.89/0.4108/13.83	21.65/0.4711/17.65

addition, the quality of Tiffany reconstructed image rises significantly when high tampering rates are presented.

5.4. Self-recovery evaluation under different tampering rates

Taking into account the results presented in the previous section, the reconstruction process with the inpainting process was performed on the images with alterations from 10% to 80%. Fig. 18 demonstrates the different alterations made, where noise is added to the central region of each image. Fig. 19 exposes the reconstructed images for each image shown in Fig. 18. The recovery capacity of each image decreases accordingly to the tampering rate increasing in each image; however, the original content of the image is clearly appreciated.

Table 9
Tamper detection analysis comparison for the proposed BFW-SR and state-of-the-art schemes.

	Tampering rate							
	20%		40%		60%		80%	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
BFW-SR	0.9336	1	0.9697	1	0.9608	1	0.9701	1
Singh [10]	0.9824	0.7507	0.9937	0.7501	0.9901	0.7502	0.9956	0.7487
Dadkhah [9]	0.9658	1	0.9938	1	0.9801	1	0.9870	1
Tong [6]	0.9826	0.7479	0.9936	0.7489	0.9902	0.7496	0.9956	0.7505
Fan [7]	0.9025	1	0.9697	1	0.9801	1	0.9701	1
Tai [13]	0.9657	0.9963	0.9938	0.9966	0.9801	0.9962	0.9870	0.9962

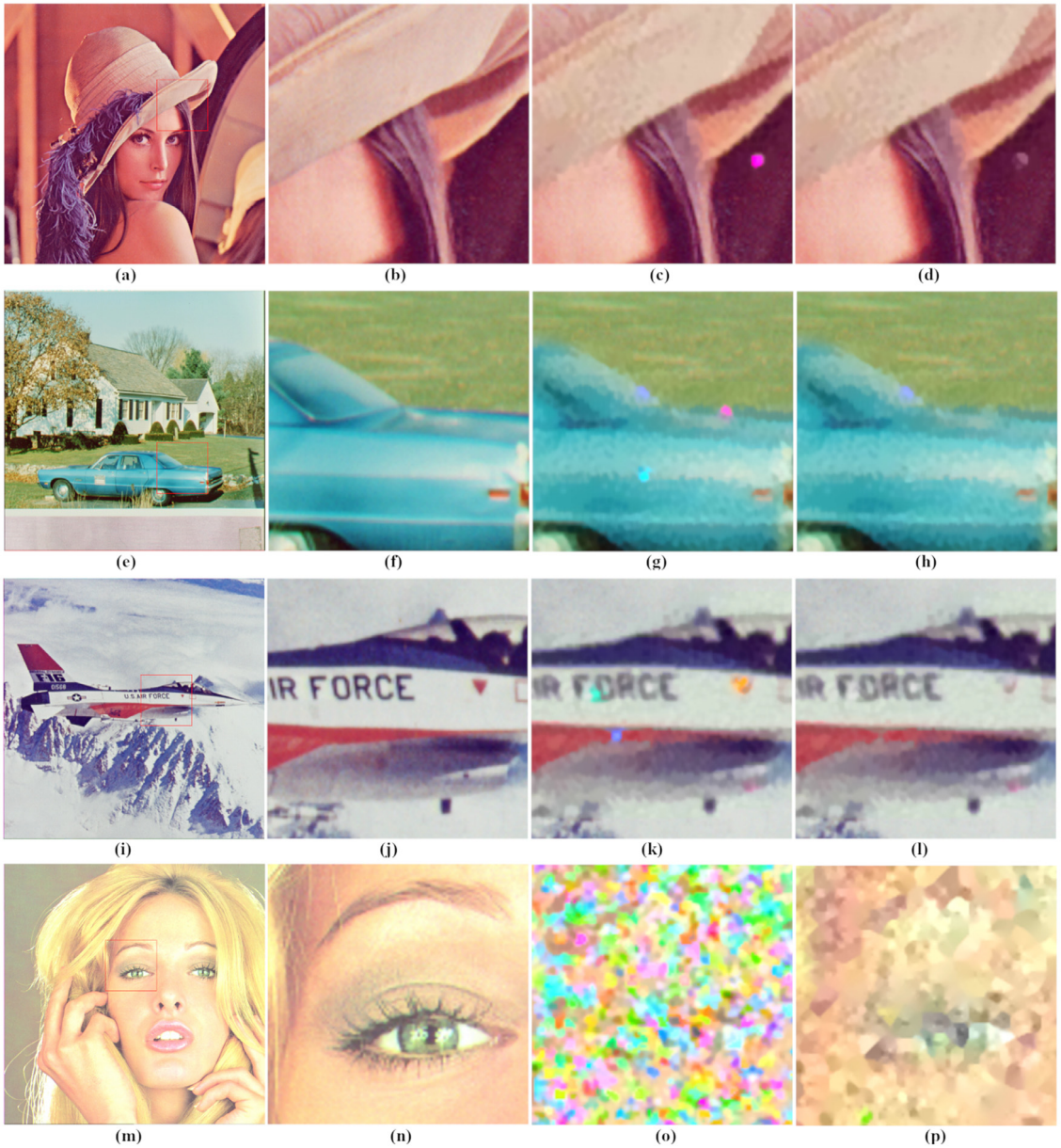

Fig. 17. Reconstructed images. (a), (e), (i) and (m) are the original image; (b), (f), (j) and (n) are the original region; (c), (g), (k) and (o) are the reconstructed regions without inpainting stage; and (d), (h), (l) and (p) are the reconstructed regions after inpainting stage.

Furthermore, it can be observed that the granulated effects increase in the appearance of the image regarding the rate of alteration made. This occurs due to the inpainting process, since only the regions affected by the tampering coincidence problem are filled with the intensity value of the neighboring pixels.

Finally, the previous process was implemented in each test image. Tables 11–13 show the values obtained in terms of PSNR (dB), SSIM and PSNR-HVS-M (dB), respectively. It can be appreciated that the objective criteria values do not decrease significantly in each alteration rate performed. Furthermore, as shown in Fig. 19, the images quality is acceptable when high tampering rates are given. For example, it

Table 10
Performance (PSNR (dB) / SSIM/PSNR-HVS-M (dB) values) for reconstructed images with and without the inpainting application.

	Without inpainting	With inpainting
Lenna	37.01/0.9810/35.51	37.43/0.9811/37.10
House	34.59/0.9671/30.23	34.95/0.9678/30.70
F-16	33.97/0.9723/31.33	34.49/0.9729/33.04
Tiffany	13.35/0.1942/7.61	21.65/0.4711/17.65

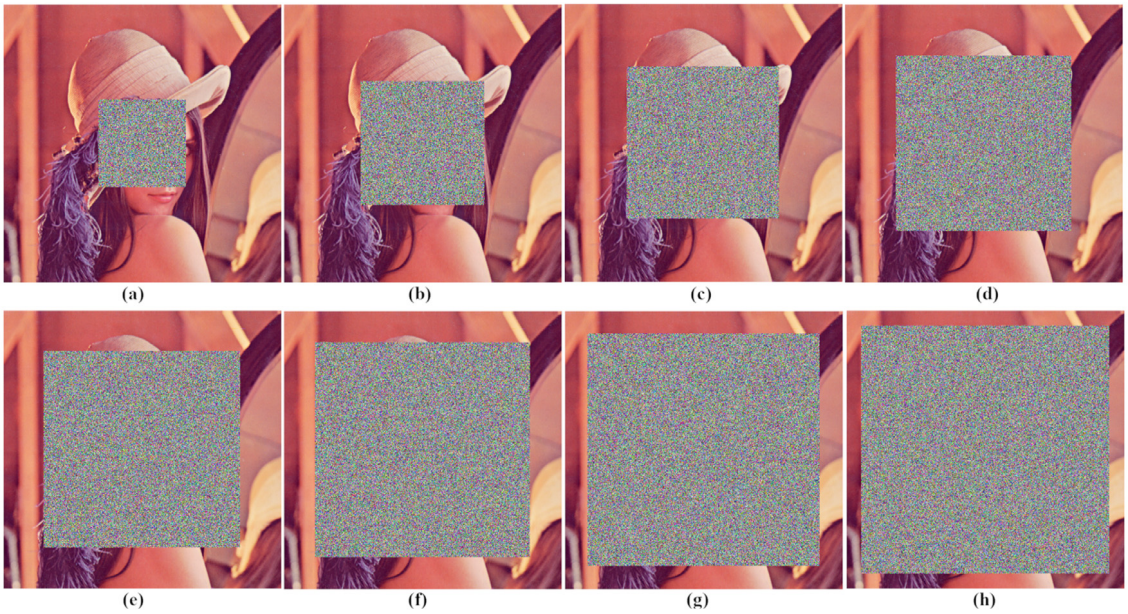

Fig. 18. Tampered images. (a) 10% tampering. (b) 20% tampering. (c) 30% tampering. (d) 40% tampering. (e) 50% tampering. (f) 60% tampering. (g) 70% tampering. (h) 80% tampering.

Fig. 19. Reconstructed images. (a) 10% tampering. (b) 20% tampering. (c) 30% tampering. (d) 40% tampering. (e) 50% tampering. (f) 60% tampering. (g) 70% tampering. (h) 80% tampering.

obtains respective averages of 19.20 dB, 0.3958 and 15.11 dB in terms of PSNR, SSIM and PSNR-HVS-M during the reconstruction of 80% alteration. It should be noted that other state-of-the-art schemes report reconstructing alterations of up to 50%, however, they do not consider the tampering coincidence problem. In comparison with them, the designed scheme takes into account this problem during the report of tests and results.

Table 11
PSNR (dB) values for the reconstructed images by the proposed BFW-SR scheme.

Tampering rate	Image								Average
	Lenna	Sailboat	Peppers	Baboon	House	Splash	Tiffany	F-16	
10	37.16	36.62	37.38	35.85	35.44	40.33	39.46	36.51	37.34
20	33.83	33.10	34.63	31.87	32.53	36.95	35.52	33.40	33.98
30	31.48	30.45	32.48	28.38	30.31	33.70	32.13	31.28	31.28
40	29.07	27.65	29.89	25.59	27.90	30.49	28.67	28.51	28.47
50	26.96	25.23	27.31	23.59	25.58	28.02	25.35	25.99	26.00
60	24.45	22.81	24.61	21.75	23.34	25.13	22.55	23.44	23.51
70	22.33	20.49	22.21	20.09	21.07	22.74	20.00	20.94	21.23
80	20.34	18.44	19.92	18.52	19.17	20.37	17.96	18.89	19.20

Table 12
SSIM values for the reconstructed images by the proposed BFW-SR scheme.

Tampering rate	Image								Average
	Lenna	Sailboat	Peppers	Baboon	House	Splash	Tiffany	F-16	
10	0.9718	0.9679	0.9667	0.9622	0.9730	0.9787	0.9735	0.9775	0.9714
20	0.9378	0.9358	0.9299	0.9202	0.9403	0.9531	0.9416	0.9536	0.9390
30	0.8991	0.8930	0.8867	0.8673	0.9040	0.9153	0.8958	0.9201	0.8977
40	0.8447	0.8301	0.8251	0.7959	0.8528	0.8583	0.8242	0.8632	0.8368
50	0.7700	0.7497	0.7491	0.7183	0.7817	0.7804	0.7250	0.7827	0.7571
60	0.6629	0.6402	0.6406	0.6166	0.6762	0.6589	0.6046	0.6681	0.6460
70	0.5368	0.5155	0.5134	0.4974	0.5415	0.5228	0.4691	0.5290	0.5157
80	0.4154	0.3951	0.3959	0.3676	0.4166	0.4070	0.3621	0.4067	0.3958

Table 13
PSNR-HVS-M (dB) values for the reconstructed images by the proposed BFW-SR scheme.

Tampering rate	Image								Average
	Lenna	Sailboat	Peppers	Baboon	House	Splash	Tiffany	F-16	
10	36.76	37.21	36.92	34.44	32.40	39.11	39.35	34.95	36.39
20	32.35	32.58	33.31	30.94	29.25	35.29	33.92	31.08	32.34
30	29.50	29.30	30.54	28.23	26.85	31.36	29.08	28.32	29.15
40	26.16	25.49	27.14	25.04	24.16	27.54	24.84	24.79	25.64
50	23.50	22.50	23.87	22.46	21.49	24.77	21.21	21.77	22.70
60	20.48	19.62	20.86	19.91	19.07	21.60	18.25	18.92	19.84
70	18.06	17.17	18.19	17.62	16.57	19.00	15.62	16.21	17.31
80	15.85	15.05	15.81	15.57	14.56	16.45	13.55	14.07	15.11

Table 14
PSNR (dB) values for the reconstructed images of previously proposed recovery schemes and novel BFW-SR.

	Tampering rate							
	10	20	30	40	50	60	70	80
BFW-SR	**37.34**	**33.98**	**31.28**	**28.47**	**26.00**	**23.51**	**21.23**	**19.20**
Singh [10]	26.55	21.47	18.27	15.96	14.16	12.59	11.29	10.23
Dadkhah [9]	22.51	17.32	14.52	12.64	11.40	10.39	9.61	9.03
Tong [6]	35.07	28.29	23.84	20.32	17.45	15.10	13.16	11.52
Fan [7]	31.47	28.36	21.62	15.79	15.69	11.57	11.57	8.10
Tai [13]	25.89	20.57	17.43	15.21	13.54	12.01	10.80	9.81

Table 15
SSIM values for the reconstructed images of previously proposed recovery schemes and novel BFW-SR.

	Tampering rate							
	10	20	30	40	50	60	70	80
BFW-SR	**0.9714**	**0.9390**	**0.8977**	**0.8368**	**0.7571**	**0.6460**	**0.5157**	**0.3958**
Singh [10]	0.9290	0.8310	0.7257	0.6215	0.5139	0.3984	0.2855	0.1799
Dadkhah [9]	0.9131	0.7983	0.6855	0.5731	0.4704	0.3586	0.2506	0.1511
Tong [6]	0.9733	0.9171	0.8282	0.7150	0.5849	0.4520	0.3233	0.2042
Fan [7]	0.9731	0.9502	0.8875	0.7230	0.7202	0.4249	0.4249	0.0094
Tai [13]	0.9384	0.8443	0.7364	0.6226	0.5135	0.3899	0.2744	0.1655

5.5. Comparison with different schemes

The last test consists of the comparison of the recovery capacity with the state-of-the-art schemes where the same attacks reported previously were carried out. Tables 14–16 show respectively the average values in terms of PSNR (dB), SSIM and PSNR-HVS-M (dB), obtained when carrying out the recovery process to the set of test images. One can observe that the designed scheme appears to demonstrate the best performance during the recovery process of all the tampering rates carried out. This is due to the fact that three copies of the recovery watermark are embedded. Moreover, the inpainting process and the additional hierarchical authentication employed to eliminate the regions affected by the tampering coincidence problem contributes to this performance. Subsequently, the schemes presented by Tong [6] and Fan [7] ensued the second and third best results in performance during recovery at low

Table 16
PSNR-HVS-M (dB) values for the reconstructed images of previously proposed recovery schemes and novel BFW-SR.

	Tampering rate							
	10	20	30	40	50	60	70	80
BFW-SR	**36.39**	**32.34**	**29.15**	**25.64**	**22.70**	**19.84**	**17.31**	**15.11**
Singh [10]	23.39	18.20	14.91	12.61	10.76	9.18	7.80	6.69
Dadkhah [9]	18.32	13.19	10.51	8.75	7.60	6.67	5.92	5.36
Tong [6]	34.20	25.77	21.04	17.26	14.29	11.84	9.82	8.11
Fan [7]	30.53	27.49	20.08	14.07	13.95	9.89	9.89	6.41
Tai [13]	23.18	17.77	14.74	12.65	11.13	9.76	8.68	7.82

Table 17
Main characteristics of previously proposed recovery schemes [6,7,9,10,13] and novel BFW-SR.

	Recovery bit rate (bpp)	Recovery watermark generation	Authentication bit rate (bpp)	Authentication watermark generation	Recovery watermarks embedding	Embedding scheme (8 bit deep)
Singh [10]	2.5	DCT	0.5	XOR	1	3-LSB
Dadkhah [9]	1.25	Block averaging	0.75	Block averaging	1	2-LSB
Tong [6]	1.25	Block averaging	0.5	Module 2 operation to recovery bits	2	3-LSB
Fan [7]	0.75	SPIHT	0.5	HASH	2	2-LSB
Tai [13]	1.75	IWT	0.25	IWT - XOR	1	2-LSB
BFW-SR	1.75	Halftoning, block averaging	0.25	XOR - block averaging	3	2-LSB

Table 18
Reconstructed images in case of Peppers for previously proposed recovery schemes [6,7,9,10,13] and novel BFW-SR.

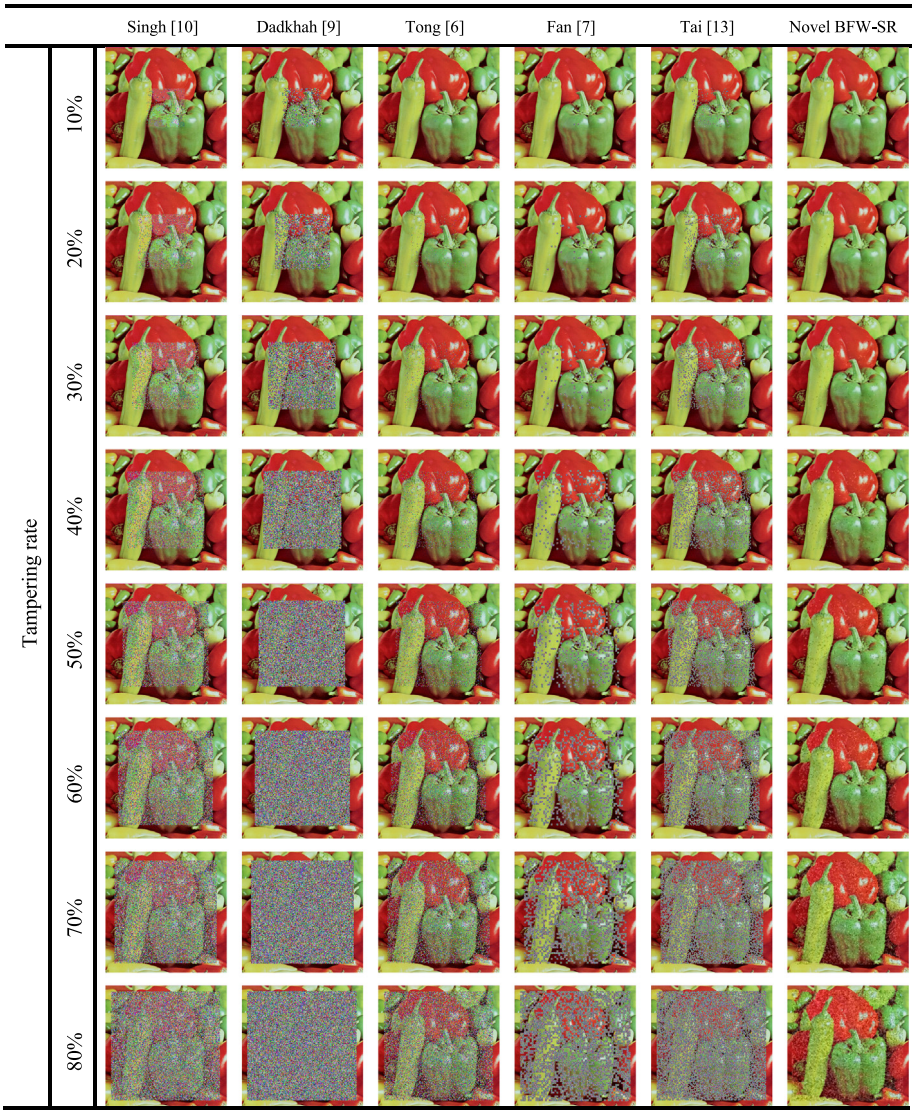

Table 19
Results for irregular attacks and multiple tampered areas for previously proposed recovery schemes [6,7] and novel BFW-SR.

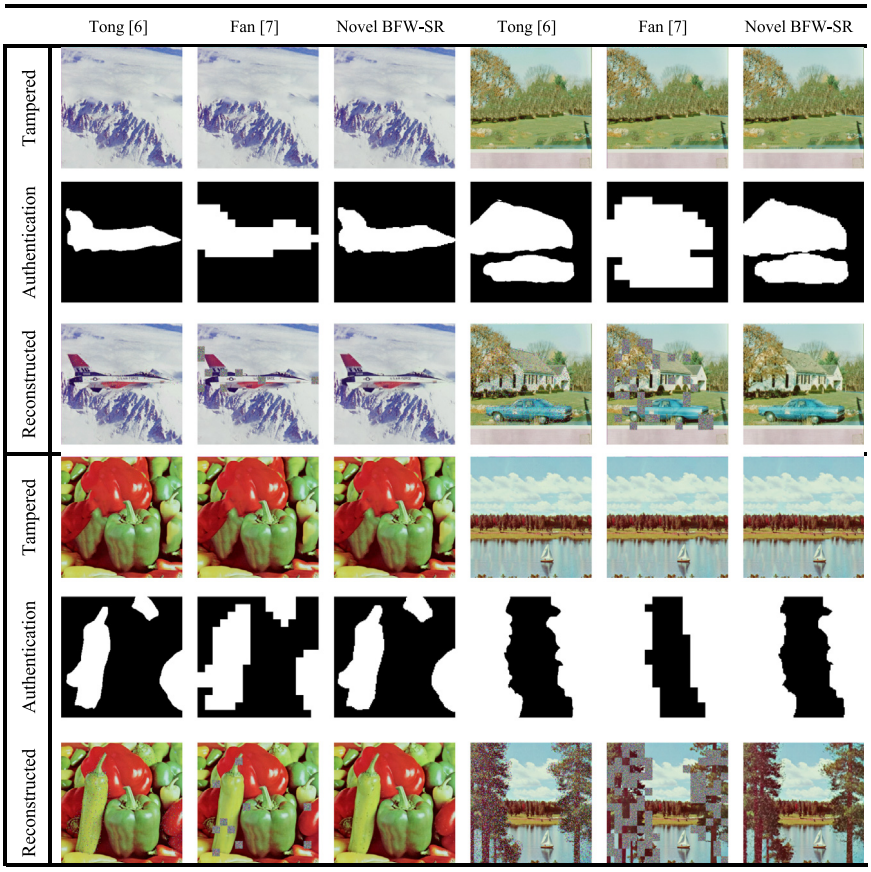

Table 20
PSNR (dB), SSIM and PSNR-HVS-M (dB) of the reconstructed images for the results shown in Table 19.

F-16			
	Tong [6]	Fan [7]	BFW-SR
PSNR (dB)	33.34	22.30	**33.82**
SSIM	**0.9699**	0.9231	0.9637
PSNR-HVS-M (dB)	32.12	20.35	**32.15**
House			
	Tong [6]	Fan [7]	BFW-SR
PSNR (dB)	20.65	16.67	**26.90**
SSIM	0.7605	0.7606	**0.8480**
PSNR-HVS-M (dB)	17.96	15.37	**23.43**
Peppers			
	Tong [6]	Fan [7]	BFW-SR
PSNR (dB)	25.80	24.18	**32.83**
SSIM	0.8628	0.8813	**0.9074**
PSNR-HVS-M (dB)	24.18	23.04	**30.25**
Sailboat			
	Tong [6]	Fan [7]	BFW-SR
PSNR (dB)	14.86	12.65	**22.22**
SSIM	0.4995	0.5413	**0.6654**
PSNR-HVS-M (dB)	13.31	13.72	**19.77**

tampering rates embedding two versions of the recovery watermark. Finally, the schemes proposed by Singh [10], Dadkhah [9], and Tai [13], which can embed just a single version of the recovery watermark, obtain low performance rates because they do not take into account the tampering coincidence problem. On the other hand, for tampering rates higher than 50%, the designed BFW-SR scheme demonstrates significantly better performance in comparison with other state-of-the-art methods. This is because, the inpainting process that is employed in our scheme can significantly improve the quality of the reconstructed images. It can be observed that the quality of the schemes proposed by Singh [10] and Tong [6] can obtain good performance, followed by methods proposed by Tai [13] and Dadkhah [9], and finally, the scheme of Fan [7].

Table 17 summarizes the comparison of the different algorithms implemented in the literature and the designed scheme for the compression rate generated of the recovery and authentication watermarks.

Table 18 shows the results for Peppers image for cases of usage the previously proposed recovery schemes [6,7,9,10,13] and novel BFW-SR, where one can observe that the quality of the designed scheme appears to be better than those reported in the literature. Finally, Table 19 exposes the comparison of the designed system and the schemes of Tong [6] and Fan [7] for the reconstruction of tampered of irregular regions, where in addition, the results of reconstruction for attacks in multiple regions are shown. The Tong [6] and Fan [7] schemes were used to make comparisons because these methods demonstrated better performance as it can be observed in Table 18. Table 20 exposes the performance results evaluated by different objective metrics for the reconstructed images shown in Table 19. Novel BFW-SR appears to demonstrate excellent performance compared to the state-of-the-art schemes, considering regular alteration attacks (Table 18), irregular and multiple alteration attacks (Tables 19 and 20).

6. Conclusions

In this paper, a fragile scheme for image authentication and recovery is proposed. Designed scheme performs the generation of the recovery watermark using the YCbCr color space, where three copies of the recovery watermark are embedded into the image in order to result in better objective quality of the recovered images against high tampering rates. In addition, a bit-adjustment process is applied during embedding to increase the quality of the watermarked images. It has been demonstrated that, for the designed scheme, better recall measure performance during authentication implies a rise of the recovery capacity that can be obtained, generating authentication watermarks for each one of the RGB channels. Furthermore, the processing based on OR operation and hierarchical authentication is implemented, improving the recall measure. Finally, an inpainting algorithm and bilateral filtering were employed to improve the quality of the results in the recovery process. Experimental results have justified that the designed scheme can detect different modifications, while maintaining high precision and recall using only four bits to authenticate each a block of 4×4 pixels. Besides, novel BFW-SR scheme can efficiently reconstruct different tampering rates (up to 80%), while maintaining superior visual and objective performance (in terms of PSNR, SSIM and PSNR-HVS-M values) after the recovery process.

Acknowledgments

Authors would like to thank to Instituto Politecnico Nacional, to Consejo Nacional de Ciencia y Tecnologia de Mexico (CONACyT), to Comision de Operacion y Fomento de Actividades Academicas (COFAA) del Instituto Politécnico Nacional, Mexico for their support during the development of this work.

References

[1] J. Molina-García, R. Reyes-Reyes, V. Ponomaryov, C. Cruz-Ramos, Watermarking algorithm for authentication and self-recovery of tampered images using dwt, in: IEEE International Kharkiv Symposium on Physics and Engineering of Microwaves, Millimeter and Submillimeter Waves, MSMW, 2016, pp. 1–4, http://dx.doi.org/10.1109/MSMW.2016.7538148.

[2] L. Rosales-Roldan, M. Cedillo-Hernandez, J. Chao, M. Nakano-Miyatake, H. Perez-Meana, Semi-fragile watermarking-based color image authentication with recovery capability, in: IEEE Int. Conf. on Telecommunications and Signal Processing, TSP, 2015, pp. 528–534, http://dx.doi.org/10.1109/TSP.2015.7296319.

[3] L. Rosales-Roldan, M. Cedillo-Hernandez, M. Nakano-Miyatake, H. Perez-Meana, B. Kurkoski, Watermarking-based image authentication with recovery capability using halftoning technique, Signal Process., Image Commun. 28 (1) (2013) 69–83, http://dx.doi.org/10.1016/j.image.2012.11.006.

[4] Kiatpapan Sawiya, Toshiaki Kondo, An image tamper detection and recovery method based on self-embedding dual watermarking, in: Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON, 2015, pp. 1–6, http://dx.doi.org/10.1109/ECTICon.2015.7206973.

[5] Tien-You Lee, Shinfeng D. Lin, Dual watermark for image tamper detection and recovery, Pattern Recognit. 41.11 (2008) 3497–3506, http://dx.doi.org/10.1016/j.patcog.2008.05.003.

[6] Xiaojun Tong, Yang Liu, Miao Zhang, Yue Chen, A novel chaos-based fragile watermarking for image tampering detection and self-recovery, Signal Process., Image Commun. 28 (3) (2013) 301–308, http://dx.doi.org/10.1016/j.image.2012.12.003.

[7] Ming Quan Fan, Hong Xia Wang, An enhanced fragile watermarking scheme to digital image protection and self-recovery, Signal Process. 66 (2018) 19–29, http://dx.doi.org/10.1016/j.image.2018.04.003.

[8] Sajjad Dadkhah, Azizah Abd Manaf, Somayeh Sadeghi, An efficient image self-recovery and tamper detection using fragile watermarking, in: International Conference Image Analysis and Recognition, Springer, 2014, pp. 504–513, http://dx.doi.org/10.1007/978-3-319-11758-4_55.

[9] S. Dadkhah, A.A. Manaf, Y. Hori, A.E. Hassanien, S. Sadeghi, An effective SVD-based image tampering detection and self-recovery using active watermarking, Signal Process. 29 (10) (2014) 1197–1210, http://dx.doi.org/10.1016/j.image.2014.09.001.

[10] Singh Durgesh, Sanjay K. Singh, Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability, J. Vis. Commun. Image Represent. 38 (2016) 775–789, http://dx.doi.org/10.1016/j.jvcir.2016.04.023.

[11] D. Singh, S.K. Singh, DCT based efficient fragile watermarking scheme for image authentication and restoration, Multimedia Tools Appl. 1104 (2015) 1–25, http://dx.doi.org/10.1007/s11042-015-3010-x.

[12] S. Shivani, D. Singh, S. Agarwal, DCT based approach for tampered image detection and recovery using block wise fragile watermarking scheme, in: Pattern Recognition and Image Analysis, Springer, 2013, pp. 640–647, http://dx.doi.org/10.1007/978-3-642-38628-2_76.

[13] W.L. Tai, Z.J. Liao, Image self-recovery with watermark self-embedding, Signal Process. 65 (2018) 11–25, http://dx.doi.org/10.1016/j.image.2018.03.011.

[14] P. Korus, A. Dziech, Efficient method for content reconstruction with self-embedding, IEEE Trans. Image Process. 22 (2013) 1134–1147, http://dx.doi.org/10.1109/TIP.2012.2227769.

[15] P. Korus, A. Dziech, Adaptive self-embedding scheme with controlled reconstruction performance, IEEE Trans. Inf. Forensics Secur. 9 (2014) 169–181, http://dx.doi.org/10.1109/TIFS.2013.2295154.

[16] X. Zhang, S. Wang, G. Feng, Fragile watermarking scheme with extensive content restoration capability, in: Digital Watermarking, Springer, 2009, pp. 268–278, http://dx.doi.org/10.1007/978-3-642-03688-0_24.

[17] D. Singh, S. Shivani, S. Agarwal, Self-embedding pixel wise fragile watermarking scheme for image authentication, in: Intelligent Interactive Technologies and Multimedia, Springer, 2013, pp. 112–122, http://dx.doi.org/10.1007/978-3-642-37463-0_10.

[18] Y.-Z. He, Z. Han, A fragile watermarking scheme with pixel-wise alteration localisation, in: 9th International Conference on Signal Processing, 2008, pp. 2201–2204, http://dx.doi.org/10.1109/ICOSP.2008.4697585.

[19] S. Shivani, A.K. Patel, S. Kamble, S. Agarwal, An effective pixel-wise fragile watermarking scheme based on ara bits, in: Proceedings of the 2011 International Conference on Communication, Computing & Security, ACM, 2011, pp. 221–226, http://dx.doi.org/10.1145/1947940.1947987.

[20] X. Zhang, S. Wang, Statistical fragile watermarking capable of locating individual tampered pixels, IEEE Signal Process. Lett. 14 (2007) 727–730, http://dx.doi.org/10.1109/LSP.2007.896436.

[21] Robert W. Floyd, Louis Steinberg, An adaptative algorithm for spatial greyscale, J. Soc. Inf. Disp. 17 (2) (1976) 75–77.

[22] N. Ponomarenko, F. Silvestri, K. Egiazarian, M. Carli, V. Lukin, On between-coefficient contrast masking of DCT basis functions, CD-ROM, in: Proc. of Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics, VPQM–07, 2007.

[23] Saeed S, A.A. Mohammad, A source-channel coding approach to digital image protection and self-recovery, IEEE Trans. Image Process. 24 (7) (2015) 2266–2277, http://dx.doi.org/10.1109/TIP.2015.2414878.

[24] Javier Molina-Garcia, Volodymyr Ponomaryov, Rogelio Reyes-Reyes, Clara Cruz-Ramos, Parallel halftoning technique using dot diffusion optimization, in: Real-Time Image and Video Processing 2017, Vol. 10223, International Society for Optics and Photonics, 2017, p. 102230K, http://dx.doi.org/10.1117/12.2262139.

[25] D.E. Knuth, Digital halftones by dot diffusion, ACM Trans. Graph. 6 (4) (1987) 245–273, http://dx.doi.org/10.1145/35039.35040.

[26] M. Mese, P.P. Vaidyanathan, Image halftoning using optimized dot diffusion, in: 9th European Signal Proc. Conf., EUSIPCO, 1998, pp. 1–4.

[27] M. Mese, P.P. Vaidyanathan, Optimized halftoning using dot diffusion and methods for inverse halftoning, IEEE Trans. Image Process. 9 (4) (2000) 691–709, http://dx.doi.org/10.1109/83.841944.

[28] J. Jarvis, C. Roberts, A new technique for displaying continuous tone images on a bilevel display, IEEE Trans. Commun. 24 (8) (1976) 891–898, http://dx.doi.org/10.1109/TCOM.1976.1093397.

[29] P. Stucki, MECCA: A Multiple-Error Correction Computation Algorithm for Bi-Level Image Hardcopy Reproduction, IBM Thomas J. Watson Research Division, 1981.

[30] N. Damera-Venkata, T.D. Kite, M. Venkataraman, B.L. Evans, Fast blind inverse halftoning, in: Proc. IEEE International Conference on Image Processing, vol. 2, 1998, pp. 64–68, http://dx.doi.org/10.1109/ICIP.1998.723318.

[31] Fernando Pelcastre-Jimenez, Mariko Nakano-Miyatake, Karina Toscano-Medina, Gabriel Sanchez-Perez, Hector Perez-Meana, An inverse halftoning algorithms based on neural networks and atomic functions, IEEE Lat. Am. Trans. 15 (3) (2017) 488–495, http://dx.doi.org/10.1109/TLA.2017.7867599.

[32] K.N. Chaudhury, D. Sage, M. Unser, Fast O(1) bilateral filtering using trigonometric range kernels, IEEE Trans. Image Process. 20 (11) (2011) 3376–3382, http://dx.doi.org/10.1109/TIP.2011.2159234.

[33] K.N. Chaudhury, Acceleration of the shiftable O(1) algorithm for bilateral filtering and non-local means, IEEE Trans. Image Process. 22 (4) (2013) 1291–1300, http://dx.doi.org/10.1109/TIP.2012.2222903.

[34] S. Ghosh, K.N. Chaudhury, On fast bilateral filtering using Fourier kernels, IEEE Signal Process. Lett. 23 (5) (2016) 570–573, http://dx.doi.org/10.1109/LSP.2016.2539982.

[35] K.N. Chaudhury, Fast bilateral filter, 2019, www.mathworks.com/matlabcentral/fileexchange/36657, MATLAB Central File Exchange (accessed 16 June 2019).

[36] A. Criminisi, P. Perez, K. Toyama, Region filling and object removal by exemplar-based image inpainting, IEEE Trans. Image Process. 13 (9) (2004) 1200–1212, http://dx.doi.org/10.1109/TIP.2004.833105.