entropy

MDPI

Article

A Fragile Image Watermarking Scheme in DWT Domain Using Chaotic Sequences and Error-Correcting Codes

Andy M. Ramos, José A. P. Artiles, Daniel P. B. Chaves and Cecilio Pimentel *

Department of Electronics and Systems, Federal University of Pernambuco, Recife 50740-550, Brazil;
andy.ramos@ufpe.br (A.M.R.); japalepg@gmail.com (J.A.P.A.); daniel.chaves@ufpe.br (D.P.B.C.)
* Correspondence: cecilio.pimentel@ufpe.br; Tel.: +55-81-98821-3442

Abstract: With the rapid development of digital signal processing tools, image contents can be easily manipulated or maliciously tampered with. Fragile watermarking has been largely used for content authentication purposes. This article presents a new proposal for image fragile watermarking algorithms for tamper detection and image recovery. The watermarked bits are obtained from the parity bits of an error-correcting code whose message is formed from a binary chaotic sequence (generated from a secret key known to all legitimate users) and from bits of the original image. Part of the codeword (the chaotic bits) is perfectly known to these users during the extraction phase, adding security and robustness to the watermarking method. The watermarked bits are inserted at specific sub-bands of the discrete wavelet transform of the original image and are used as authentication bits for the tamper detection process. The imperceptibility, detection, and recovery of this algorithm are tested for various common attacks over digital images. The proposed algorithm is analyzed for both grayscale and colored images. Comparison results reveal that the proposed technique performs better than some existing methods.

Keywords: authentication; chaotic maps; error-correcting codes; discrete wavelet transform; image watermarking; tamper detection; security

check for updates

Citation: Ramos, A.M.; Artiles, J.A.P.; Chaves, D.P.B.; Pimentel, C. A Fragile Image Watermarking Scheme in DWT Domain Using Chaotic Sequences and Error-Correcting Codes. *Entropy* **2023**, *25*, 508. https://doi.org/10.3390/e25030508

Academic Editor: Congxu Zhu

Received: 27 February 2023
Revised: 13 March 2023
Accepted: 14 March 2023
Published: 16 March 2023

1. Introduction

Digital watermarking is a technique of hiding information in multimedia data in such a way that the distortion due to watermarking is almost perceptually negligible [1]. Watermarking can serve a variety of purposes including copyright protection and data authentication. Image watermarking is the process of embedding binary information (called watermarked bits) into an original image, generating a watermarked image. In a self-embedding watermarking scheme, the watermarked bits are generated from the original image. The extraction process is called blind when it does not require knowledge of either the original image or the watermarked bits.

In general, image watermarking techniques can be categorized as robust, semi-fragile and fragile [2–4]. Robust watermarks are designed to survive image processing operations, such as scaling, cropping, filtering, and compression [5–9], and are usually used for copyright protection to declare ownership. Fragile watermarking is designed for detecting any modification of the watermarked image (tamper detection) and for recovering the tampered areas (image recovery) [2]. Semi-fragile schemes are designed for tamper detection and image recovery and are robust against some image processing operations. Their main disadvantage is a reduced recovering rate when compared to that achieved by fragile schemes. Fragile and semi-fragile watermarking schemes are mainly used for authentication purposes.

This work focuses on image fragile watermarking. The primary purpose is to invisibly embed a binary image (called watermark image) into an original image, creating a watermarked image, and then extract the embedded information from the watermarked image at the destination. The watermark is fragile in the sense that it is designed to be easily altered if any changes are made to the watermarked image, hence providing a means of detecting tampering or unauthorized alterations. Fragile watermarking can therefore be applied to copyright protection, tamper detection, and authentication.

In many applications, tamper detection and localization alone are insufficient. The fragile watermarking with self-recovery capability cannot only identify the tampered regions, but also recover the altered image's original content. At the destination, the authentication watermark is first extracted and applied to identify the authenticity of the received image. If the watermarked image is identified as tampered, the restoration technique is applied to the tampered parts.

In many image fragile watermarking schemes, the original image is divided into non-overlapping sub-blocks and the watermark embedded in each sub-block is composed of authentication bits and recovery bits [10–17]. The authentication bits are used for the purpose of tampering detection (the block is authenticated if the authentication bits are successfully retrieved). The tampered blocks are recovered by means of the recovered bits. The generation of the watermarked bits involves, in some cases, frequency-domain transforms, such as the discrete cosine transform (DCT) [15,18], and the discrete wavelet transform (DWT) [14,19–21].

The performance of an image watermarking scheme is analyzed with mutually exclusive parameters, including imperceptibility, capacity, and robustness against attacks. Trying to improve one of these parameters for a particular scheme usually deteriorates the others [1]. Several embedding schemes are based on the least significant bit (LSB) method [11–13,16,19,22], since it provides a good trade-off among these performance metrics.

Chaotic maps are commonly used to add security to image watermarking schemes [13,14,22–25]. These maps are characterized by their sensitivity to the initial conditions and pseudo-random behavior, despite being deterministic, resulting in noise-like signals [14,26,27]. Applications of these maps include scrambling the original image [13,14,22] and selecting sub-blocks to embed the watermark [14,22]. To support severe distortion imposed on the watermarked image, error-correction codes can also be applied [18,28,29].

In this work, we propose a new self-embedding fragile watermarking algorithm for image tamper localization and recovery using chaotic maps, DWT domain, and error-correcting codes. The bits embedded in the image are obtained from parity bits of an error-correcting code whose information sequence is formed by combining the watermarked bits with chaotic bits generated from a secret key. The distinguishing feature of the proposed extraction algorithm is that the error-correcting capability of the error-correction code is exclusively dedicated to recovering the watermarked bits, since part of the codeword (the chaotic bits) is known by the extraction algorithm, which provides high robustness to the proposed scheme. The DWT sub-bands are divided into non-overlapping 2×2 sub-blocks and two parity bits are embedded in each sub-block. These bits are used as authentication bits for the tamper detection process. A parameter controls the trade-off between imperceptibility and robustness. After locating the tampered area, in the process of recovering the damaged area, the parity bits and chaotic sequences are used to estimate the recovery bits. We investigate the trade-off between the imperceptibility of the watermarking embedding and the tampering detection/recovering capability of the proposed algorithm and comparison results reveal that it performs better than many existing fragile watermarking schemes.

The rest of this article is organized into five sections. Section 2 presents a brief review of the chaotic maps, the class of error-correcting codes considered in this work, and DWT. The proposed algorithm for grayscale images is detailed in Section 3. It is also discussed the watermark extraction, tamper detection, and the image recovery strategy. Some metrics commonly used for assessing the imperceptibility, detection, and recovery capability of a fragile watermarking algorithm are discussed in Section 4. Results with performance comparisons are presented in Section 5 for grayscale images, and in Section 6 for colored images. Conclusion remarks are outlined in Section 7.

Related Works

In this section, we briefly review several fragile watermarking schemes proposed in the literature.

Haghighi et al. [22] proposed a fragile blind watermarking scheme, based on lifting wavelet transform (LWT) and genetic algorithms. In this scheme, four digests are generated based on LWT and halftoning techniques. Each digest is separately scrambled using a chaotic map. The authentication bits for each 2 × 2 non-overlapping sub-block are calculated based on a relation of pixels. The watermarked bits are formed from a combination of digests and authentication bits and are embedded using the LSB technique. A genetic algorithm is employed to optimize the difference between the original and the watermarked values of each sub-block.

Barani et al. [23] proposed a digital image tamper detection algorithm based on the integer wavelet transform (IWT) and singular value decomposition (SVD). A SVD is performed in each 2 × 2 sub-block of the scrambled original image. The combination of the U matrix of the SVD of each sub-block and a sequence generated by a 3D quantum chaotic map forms an authentication sequence that is inserted into the IWT coefficients. A scheme that combines SVD and chaotic maps is proposed in the Ref. [25].

In the image fragile watermark scheme proposed in the Ref. [14], the original image is divided into 4 × 4 non-overlapping sub-blocks and the authentication and the recovery bits are both generated by using the DWT. The authentication bits are generated from the low-frequency sub-band of each sub-block, and the recovery bits are produced from high-frequency sub-bands. The chaotic Arnold’s cat map scrambles image sub-blocks in order to break their interdependence.

In the Ref. [30], Qin et al. proposed a self-embedding fragile watermarking scheme using vector quantization (VQ) and index sharing. The watermarked bits are composed of hash bits for tampering localization and reference bits for content recovery. The proposed scheme can locate tampered regions via VQ index reconstruction. Qin et al. [11] developed a self-embedding fragile watermarking based on reference data interleaving mechanism. This scheme utilizes the most significant bit (MSB) layers to generate the interleaved reference bits that are embedded into the LSBs. The scheme proposed in the Ref. [16] embeds the watermarked bits generated by a permutation process within the two LSB of each sub-block. A bit-adjustment phase is subsequently applied to increase the quality of the watermarked image. In the Ref. [31], the original image is divided into non-overlapping sub-blocks of 2 × 2 pixels, called small blocks, and each 4 × 4 small block is grouped as a large block. The watermarked bits containing authentication information and recovery information are embedded into the LSB.

In the Ref. [15], authentication data is generated for each 8 × 8 sub-block using the DCT. A block dependency is established using part of the authentication data of a distant block. Such sub-block dependency provides tamper detection and enables localization of tampered regions. A recovery technique based on unsupervised machine learning is proposed. The scheme presented in the Ref. [32] is also based on the DCT. Two authentication bits and ten recovery bits are generated from the five MSB of each sub-block. The authentication bits of each sub-block are embedded into the three LSB.

The algorithm proposed in the Ref. [12] consists of an overlapping block-wise mechanism for tampering detection and a pixel-wise mechanism for image recovery. Reference bits are derived from the mean value of each sub-block and are dispersedly hidden into 1 or 2 LSB according to two different embedding modes. Authentication bits are hidden into adaptive LSB layers of the central pixel for each block. After detecting tampered blocks and reconstructing mean-value bits, the original pixels are recovered using a pixel-wise approach with the help of different neighboring overlapping blocks. According to [17], two different types of detection processes, pixel-wise and block-wise processes, are suggested in order to locate and restore the tampered locations. The authentication data are created per pixel in the pixel-wise procedure while they are formed per block in the block-wise process. As a result, the block-wise method tunes the length of authentication data to the size of each block.

Peng et al. proposed in the Ref. [10] an algorithm based on reversible data hiding. The authentication and recovery bits are embedded into two identical original images. Secret information is embedded into one image while distortion information is embedded into the other one. In the Ref. [13], Sreenivas et al. proposed an image tamper localization scheme in which authentication bits of a 2 × 2 image sub-block are generated using chaotic maps. For each sub-block, two distinct sets of recovery bits are generated and embedded in the LSBs of two randomly chosen blocks. In the Ref. [33], a secret key based on pseudo-random binary sequences is used as a fragile watermark for tamper detection. The watermarked bits are embedded using a LSB process in a 9-base notation structure.

Li et al. [34] proposed an image tampering detection and a self-recovery method based on the Gauss–Jordan Elimination. A technique called Improved Check Bits Generation (ICBG) generates the check bits for tamper detection. The Morphological Processing-Based Enhancement (MPBE) is developed to improve the accuracy of tampering detection.

The scheme proposed in the Ref. [19] used two watermarks that combines spatial and transform domains to enhance the watermarking robustness, authentication and recovery performance. A robust watermarking is embedded into different DWT sub-bands, while a fragile one is embedded using the LSB approach. An image authentication system that combines DWT and convolutional neural networks (CNN) is proposed in the Ref. [20]. The watermark information is embedded into the DWT coefficients, and the CNN is employed to recover tampered areas. The combination of DCT and CNN is proposed in the Ref. [35].

Multiple median watermarking is a technique for image tamper region recognition and self-recovery proposed in the Ref. [36]. Four smaller versions of the cover image are hidden into the 4-LSB, which are determined by four pseudo-random codes. These copies can be used to identify the area of the altered image that has been tampered with.

2. Preliminaries

2.1. BCH Code

A well-known and powerful tool to enhance the robustness of a watermarking scheme is the use of error-correcting codes which permits to correct errors induced by a given attack [28,37,38]. This work employs the binary Bose, Chaudhuri, and Hocquenghem (BCH) code [39] over the Galois field GF(q) with codewords of length $n = q^m - 1$ (where m is an integer), k information bits, $n-k$ parity bits, and error correction capability t bits. It is denoted by BCH (n, k, t). This code is completely specified by its generator polynomial $g(x) = 1 + g_1 x + \cdots + g_{n-k-1} x^{n-k-1} + x^{n-k}$, where $g_i \in \mathrm{GF}(q)$. The degree of $g(x)$ is equal to the number of parity bits of the code. Let ξ be a primitive element of GF(2^4), and let $m_i(x)$ be the minimal polynomial of ξ^i in GF(2^4), the i-th power of ξ. The generator polynomial $g(x)$ is obtained from the least common multiple of the minimum polynomials $g(x) = \mathrm{LCM}(m_1(x), m_2(x), \ldots, m_{d_m-1}(x))$, where d_m is the code minimum distance and satisfies $d_m \geq 2t + 1$.

A polynomial representation $c(x)$ of a codeword $\mathbf{c} = (c_0, \cdots, c_{n-1})$ is of the form $c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$. The encoder operation can be expressed in the polynomial form $c(x) = g(x)\, u(x)$, where $u(x)$ is the information message to be encoded and the operations with polynomials follow the operations rules defined over the field. There are several techniques to decoding BCH codewords, it is worth mentioning the Berlekamp–Massey (BM) algorithm, an algebraic decoding algorithm for the BCH code. For the one-bit error correction code considered here, the generator polynomial is $g(x) = x^4 + x + 1$, for $q = 2$, $m = 4$, with $n = 15$, and $k = 11$ information bits, which has 4 parity bits and $t = 1$, represented as BCH(15,11,1). We also use a BCH code with $t = 2$, BCH(31,21,2), with 10 parity bits and generator polynomial $g(x) = x^{10} + x^9 + x^8 + x^6 + x^5 + x^3 + 1$. The use of both codes permits analyzing the impact of the code parameters on the watermarking algorithm.

In the proposed algorithm, the k information bits are split into 2 parts, one containing bits from the watermark image and the other a chaotic sequence generated from a secret key shared by legitimate users. The encoder operation finds the parity bits that are embedded into the original image, generating the watermarked image. The watermark extraction algorithm estimates the parity bits from a possibly modified watermarked image. A legitimate user generates the chaotic sequence and concatenates it with the retrieved parity bits. A decoding procedure estimates the watermarked bits forming the retrieved codeword (the concatenation of the estimated watermarking bits, chaotic sequence, and the estimated parity bits). It is worth highlighting that if the retrieved codeword contains errors (which occur if the errors introduced by an attack are beyond the error correction capability), these are spread out over all portions of the codeword and do not necessarily concentrate in the portion of the information sequence where the watermarked bits are located. However, the positions of the chaotic bits are known in advance, since this information is shared by all legitimate users. This allows for concentration of the code's correctness capability on the portion of the codeword that contains the relevant information, the parity and watermarked bits.

2.2. Chaotic Maps

The behavior of unidimensional chaotic maps is observed through a discrete time series $\{x_i\}_{i=0}^{\infty}$, and can be obtained by iterating a nonlinear and non-invertible function $f(x)$, over an initial condition x_0, as follows [40]

$$x_n = f(x_{n-1}), \quad n = 1, 2, 3, \ldots \tag{1}$$

The sequence $\{x_n\}_{n=0}^{\infty} = \{x_0, f(x_0), f(f(x_0)), \ldots\}$ is called an orbit of $f(x)$. The value of x_0 is obtained from the secret key and we choose to discard the first 100 samples of an orbit due to the transient behavior (the orbits of a good chaotic map diverge after few iteration and this discard operation is not mandatory). Examples of chaotic maps include the cubic map (MC) $f(x) = 4x^3 - 3x$, and the logistic map (ML) $f(x) = rx(1-x)$, where r is a control parameter. Chaotic maps are deeply sensitive to the initial condition, meaning that arbitrarily close initial points separate exponentially fast. Figure 1 shows two orbits of the logistic map with $r = 4$ generated by two initial conditions separated by 10^{-6}. These assume a distinct dynamical behavior after a few iterations.

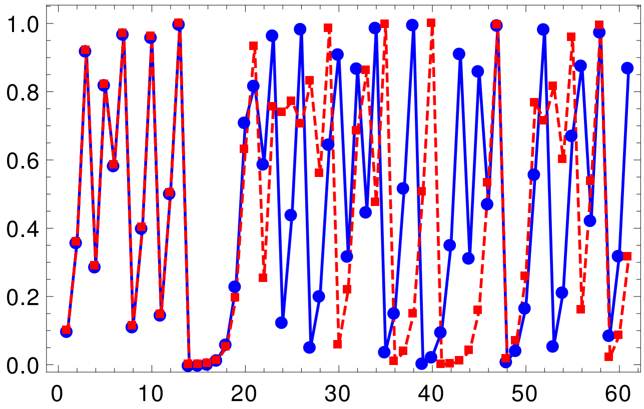


Figure 1. Two different orbits using (1) for the logistic map with $r = 4$. The initial conditions are $x_0 = 0.1$ and $x'_0 = 0.100001$.

The balanced binary sequence $\{z_n\}$, henceforth denoted by the chaotic binary sequence, is generated from $\{x_n\}$ from a partition of the map domain into two regions $\mathcal{R}_0$ and $\mathcal{R}_1$ satisfying $\Pr(x_n \in \mathcal{R}_0) = \Pr(x_n \in \mathcal{R}_1) = 1/2$, and such that, if $x_n \in \mathcal{R}_0$ then $z_n = 0$, or if $x_n \in \mathcal{R}_1$ then $z_n = 1$. There are several methods to discretize chaotic samples (see, for example, [41–43]) that may generate binary sequences with better random properties, but this topic is not explored in this work.

2.3. Discrete Wavelet Transform

The basis functions of the DWT are generated from a basic wavelet function, through translations and dilations. These functions allow reconstructing the original signal through the inverse discrete wavelet transform (IDWT). There are many types of wavelet functions, including Haar [44,45], Daubechies [46], Symlets [47], Coifflets [48,49]. Due to its low computing requirements, the Haar transformation has been used primarily for image processing and pattern recognition and is adopted in this work.

Mallat [50] proposed an algorithm based on a decomposition following a pyramid model, in which the image size decreases in each decomposition level. Figure 2 shows the first decomposition level applied to an image C_O of size $M \times N$, obtaining four output images $C_{LL_1}, C_{LH_1}, C_{HL_1}, C_{HH_1}$ of size $M/2 \times N/2$. At the end of each filtering operation, the output signal is down-sampled by two ($\downarrow 2$). The image C_{LL_1} is obtained from the convolution of two low-pass filters applied first to the rows and then to the columns of C_O. The first level of detail C_{LH_1} is obtained by applying a low-pass filter to the rows of C_O and then a high-pass filter to its columns. Similarly, C_{HL_1} and C_{HH_1} are obtained. Applying this procedure again having as input the image approximation C_{LL_1}, we obtain the second decomposition level of the image C_O, resulting in the approximations C_{LL_2} and the level of details C_{LH_2}, C_{HL_2}, C_{HH_2}, each one with size a quarter of the size of image C_O. Other decomposition levels are obtained using the same procedure.

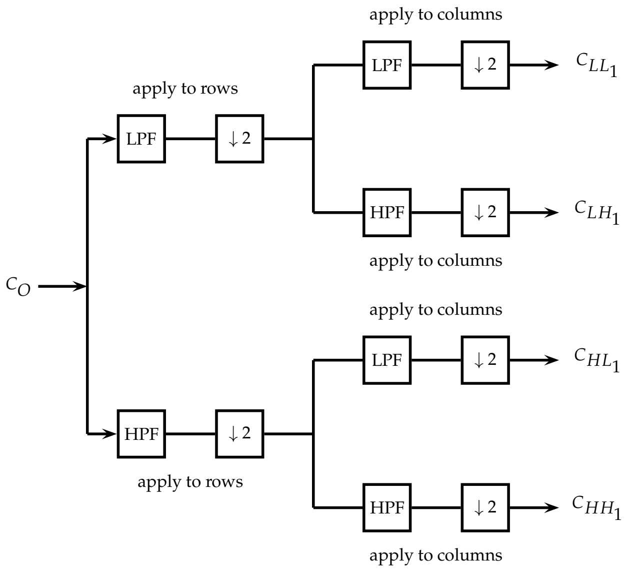


Figure 2. Wavelet decomposition scheme in two dimensions.

3. The Proposed Algorithm

A new fragile watermarking algorithm for images as well as a strategy for tamper detection and recovering of the tampered areas are proposed in this section. Initially, we consider grayscale images. colored images are considered in Section 6.

The embedding algorithm $E^{\cdot}$ has as input an 8-bit grayscale original image C_O of size $M \times N$ pixels and a key K which determines the initial condition x_0 of the chaotic sequence. The watermarked image C_W is described as

$$C_W = E(C_O, K). \tag{2}$$

The input to the blind extraction algorithm $E^{-}(\cdot)$ is the watermarked image possibly corrupted by attacks, namely C'_W, and a key K.

3.1. Watermark Embedding

Watermarked bits are embedded into the original image according to the following steps.

1. Generate a chaotic binary sequence SC_1 using the cubic chaotic map with the key K.
2. Apply the 2-level DWT decomposition to the original image C_O obtaining the sub-bands C_{LL_2}, C_{LH_2}, C_{HL_2}, C_{HH_2}. The sub-bands C_{LH_2} and C_{HL_2} (each one of size $M/4 \times N/4$ pixels) are divided into sub-blocks of size 2×2, where the watermarked bits are embedded. There are $\frac{MN}{64}$ sub-blocks in each sub-band. Each sub-block is composed of the coefficients:

c_{11}	c_{12}
c_{21}	c_{22}

3. Apply the 4-level DWT decomposition to the original image C_O. The image C_{LL_4} has size $M/16 \times N/16$ pixels. Convert each byte of this image to a binary sequence $\boldsymbol{\ell}_4$ of length $\frac{MN}{32}$ bits.
4. Construct the parity check sequence $\mathbf{p}$ of the BCH (15,11,1) code as follows. The 11 information bits are obtained by concatenating k_1 bits from $\boldsymbol{\ell}_4$ and k_2 from the chaotic map (SC_1 sequence), where $k_1 + k_2 = 11$. After encoding, a 15-bit codeword is obtained with 4 parity bits. After repeating this process for the entire $\boldsymbol{\ell}_4$, a parity sequence of size $\frac{MN}{8k_1}$ is obtained. This sequence is considered as an image and is scrambled with the Arnold cat map. After scrambled, this sequence is divided into sub-sequences of length 2 bits, $\mathbf{p} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{\frac{MN}{16k_1}}\}$, where $\mathbf{p}_i = p_{i1}, p_{i2}$. Each $\mathbf{p}_i$ is embedded into the sub-blocks of C_{LH_2} and C_{HL_2}.
5. In each 2×2 sub-block of C_{LH_2} and C_{HL_2} find the largest value (v_{max1}) and the second largest value (v_{max2}) of $c_{11}, c_{12}, c_{21}, c_{22}$. Let $\alpha_1 = v_{max1} - v_{max2}$. If $\alpha_1 \leq \alpha$, where α is a fixed positive parameter for all sub-blocks, then $v_{max1} \leftarrow v_{max1} + \alpha$, otherwise v_{max1} remains unchanged. The choice of α involves a trade-off between imperceptibility and robustness, as is discussed in the next sections. Each sub-sequence $\mathbf{p}_i$ is embedded in each sub-block of each sub-band according the following rules (consider that v_{max1} is in position (i_1, j_1) of the sub-block, $1 \leq i_1, j_1 \leq 2$):
   - If $\mathbf{p}_i = 00$, then replace $c_{i_1 j_1}$ by c_{11} and c_{11} by v_{max1}.
   - If $\mathbf{p}_i = 01$, then replace $c_{i_1 j_1}$ by c_{12} and c_{12} by v_{max1}.
   - If $\mathbf{p}_i obtain = 10$, then replace $c_{i_1 j_1}$ by c_{21} and c_{21} by v_{max1}.
   - If $\mathbf{p}_i = 11$, then replace $c_{i_1 j_1}$ by c_{22} and c_{22} by v_{max1}.
6. Apply the two-level IDWT and obtain the watermarked image C_W.

Since the number of sub-sequences $\mathbf{p}_i$ is $\frac{MN}{16k_1}$ and the total number of sub-blocks is $\frac{MN}{32}$, we have $k_1 = 2$, and consequently $k_2 = 9$. Figure 3 shows the block diagram of the proposed embedded algorithm, called Proposed 1.

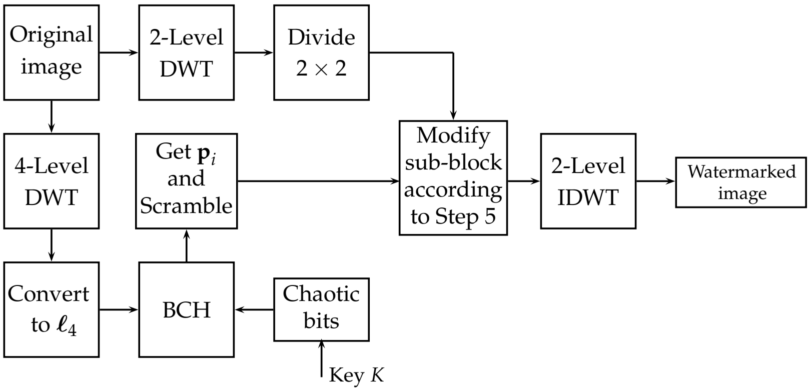


Figure 3. Block diagram of the proposed watermark embedding algorithm.

3.2. Watermark Extraction, Tamper Detection, and Image Recovery

The embedding of watermarked bits in C_O allows detecting modifications (tamper detection) and to recover the original image (image recovery).

3.2.1. Watermark Extraction

The extraction of parity sequence $\hat{\mathbf{p}}$ from C'_W (possibly modified watermarked image) and from K is based on the following steps.

- Generate the chaotic binary sequence SC_1 from the key K.
- Calculate the two-level DWT of C'_W obtaining the sub-bands C_{LH_2} and C_{HL_2}. Each sub-band is divided into sub-blocks of size 2 × 2.
- Find the highest value v'_{max} of each sub-block and its position. Decide the watermark information $\hat{\mathbf{p}}_i$ as:
    - If v'_{max} is in the position $(1,1)$, then $\hat{\mathbf{p}}_i = 00$.
    - If v'_{max} is in the position $(1,2)$, then $\hat{\mathbf{p}}_i = 01$.
    - If v'_{max} is in the position $(2,1)$, then $\hat{\mathbf{p}}_i = 10$.
    - If v'_{max} is in the position $(2,2)$, then $\hat{\mathbf{p}}_i = 11$.
- The estimated parity sequence is unscrambled with K_1 and is divided into 4-bit sub-sequences, $\hat{\mathbf{p}}_j = \hat{p}_{j1} \cdots \hat{p}_{j4}$, for $j = 1, \ldots, \frac{MN}{64}$.
- For each $\hat{\mathbf{p}}_j$, the extraction algorithm knows $k_2 = 9$ chaotic bits of an 11-bit information sequence. There are four possible parity sequences, depending on the remaining $k_1 = 2$ information bits. An estimate of these bits is obtained from the smallest Hamming distance between $\hat{\mathbf{p}}_j$ and these possible parity sequences. Then, concatenate the estimated k_1 bits, the k_2 the chaotic bits, and the four parity bits with the smallest Hamming distance to form a 15-bit word. This word is decoded using the BM algorithm, giving a new estimate of the k_1 bits of the sequence $\boldsymbol{\ell}_4$ and $\hat{\mathbf{p}}_j$.
- This procedure is repeated for each $j = 1, \ldots, \frac{MN}{64}$, obtaining two estimated sequences $\hat{\mathbf{p}}$ and $\hat{\boldsymbol{\ell}}_4$.

Figure 4 shows the block diagram of the proposed watermark extraction algorithm.

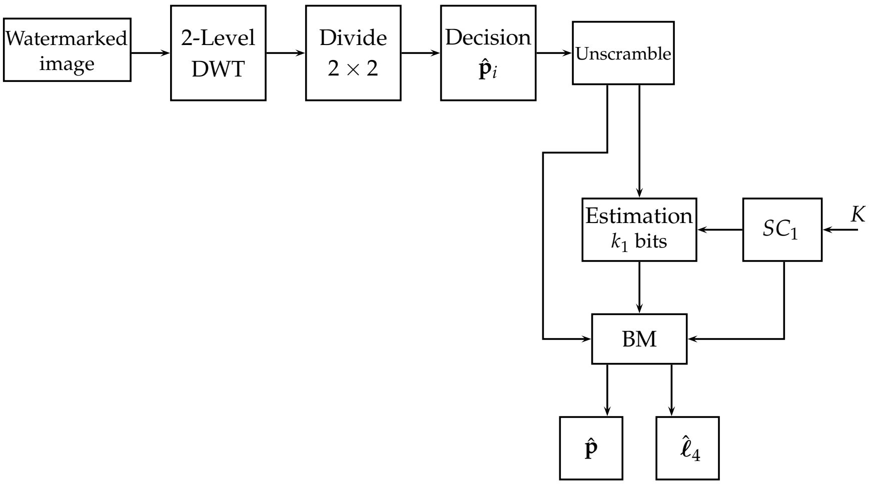


Figure 4. Block diagram of the proposed watermark extraction algorithm.

3.2.2. Tamper Detection

The image C'_W is used to replicate Steps 2–4 of the embedding algorithm, obtaining a new binary sequence $\tilde{\mathbf{p}}$ of length $\frac{MN}{8}$. In order to detect the tampered regions, a bitwise XOR operation is performed between the extracted watermark binary sequence $\hat{\mathbf{p}}$ and the binary sequence $\tilde{\mathbf{p}}$. The binary sequence resulting from this operation is organized in a binary image of size $M/4 \times N/4$ bits, which is called binary detection image. Figure 5 shows the block diagram of the proposed tamper detection algorithm.

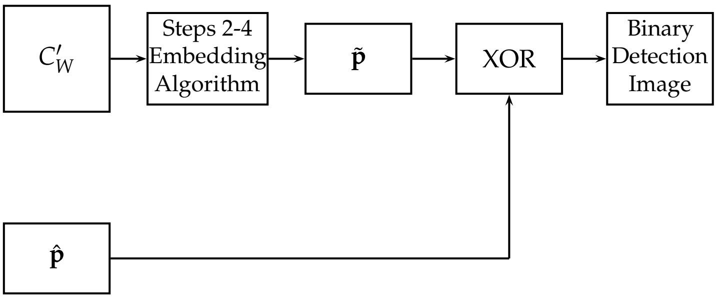


Figure 5. Block diagram of the proposed tamper detection algorithm.

3.2.3. Image Recovery

After detecting if there is any modification in the watermarked image C'_W, the next step is to recover the part of the image identified as tampered. In the recovering process, the first step is to calculate the details and sub-bands C_{HH_4}, C_{HL_4} and C_{LH_4} of the tampered image C'_W. The binary sequence $\hat{\ell}_4$ is converted to the image $\hat{C}_{LL_4}$ of size $M/16 \times N/16$ pixels. An intermediate image C_I is obtained from the 4-level IDWT of the image formed from $\hat{C}_{LL_4}$, C_{HH_4}, C_{HL_4}, and C_{LH_4}. The recovered image is constructed by replacing the pixels located at the detected tampered area of C'_W by the corresponding pixels of C_I. Figure 6 shows the block diagram of the proposed image recovery algorithm.

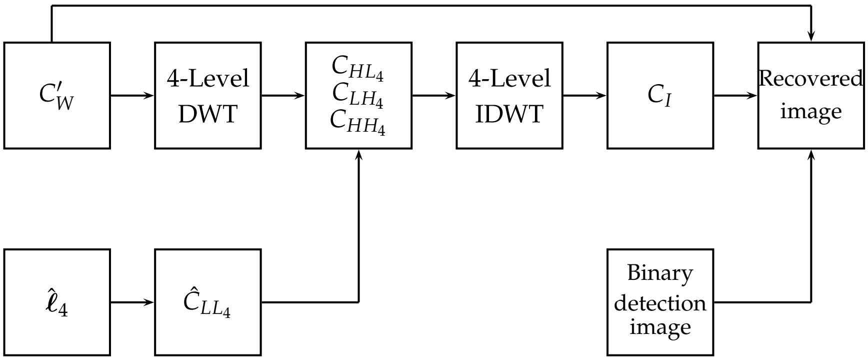


Figure 6. Block diagram of the proposed recovery algorithm.

4. Imperceptibility, Detection, and Recovery Metrics

This section describes commonly used metrics for assessing the imperceptibility and robustness of image watermarking schemes.

4.1. Imperceptibility Metrics

The peak signal-to-noise ratio (PSNR) is a measure of watermark imperceptibility, expressed in units of decibels (dB). For 8-bit grayscale images with pixel values from 0 to 255, the PSNR is defined as

$$\mathrm{PSNR} \;=\; 10\log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right)\,(\mathrm{dB}) \tag{3}$$

where the mean square error (MSE) for images of size $M \times N$ is

$$\mathrm{MSE} \;=\; \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}(C_O(i,j) - C_W(i,j))^2. \tag{4}$$

The recovered PSNR, PSNR^r, is calculated using (3) in which the MSE is obtained between the watermarked image and recovered image. The structural similarity index (SSIM) is another imperceptibility metric and is defined as

$$\text{SSIM} = \frac{(2\mu_O\mu_W+\gamma)(2\rho_{OW}+\beta)}{(\mu_O^2+\mu_W^2+\gamma)(\sigma_O^2\sigma_W^2+\beta)} \tag{5}$$

where μ_O and μ_W are the mean of the original and watermarked images, respectively, σ_O^2 and σ_W^2 are the variances of these images, ρ_{OW} is the covariance between C_O and C_W, γ and β are fixed constants, $\gamma = 2.55$ and $\beta = 7.65$.

4.2. Tampered Detection Metric

The performance of tamper detection is commonly measured in terms of the false positive rate (FPR) and false negative rate (FNR), defined as

$$\text{FPR} = \frac{\text{FP}}{\text{TN}+\text{FP}} \tag{6}$$

$$\text{FNR} = \frac{\text{FN}}{\text{FN}+\text{TP}} \tag{7}$$

where FP, FN, TP, TN are the false positive, false negative, true positive, and true negative, respectively. FP is the number of pixels that are non-tampered but are wrongly identified as tampered; FN is the number of pixels that are tampered but are incorrectly detected as non-tampered; TP is the number of pixels that are correctly identified as a tampered pixel, and TN is the number of pixels that are correctly identified as an untampered pixel. The lower FPR and FNR indicate a better performance of the tamper detection algorithm.

4.3. Watermark Image Attacks

Several attacks are performed on the watermarked image to check the behavior of the proposed algorithm, as described next.

- In the tamper attack, the pixels of a part of C_W are changed to zero [15].
- The first kind of collage attack (CA_1) tampers the C_W image by copying blocks of C_W and inserting them into arbitrary positions in the same watermarked image [14,15].
- The second kind of collage attack (CA_2) modifies C_W by combining portions of another watermarked image and preserving their relative spatial locations [14,15,17].
- In the normal tampering attack, some objects are added, deleted or modified on the watermarked image [22].
- The salt and pepper attack consists in adding this noise with density d to the C_W image [17].
- The constant-average attack (CAA) [14,15] is able to tamper a set of blocks with a constant average intensity and create a counterfeit image. The average value for each block in the tampered area is calculated, and then the 6 MSBs of each pixel, within the block, are replaced by the 6 MSBs of the calculated average value [15].

The performance analysis conducted in this chapter uses 141 original images from the USC-SIPI database (http://sipi.usc.edu/database, accessed on 10 March 2021). This database contains grayscale and colored images of distinct sizes. We resize and convert some images so that a new database contains 8-bit grayscale images of size 512 × 512 pixels. Figure 7 shows some examples of images used in this work.

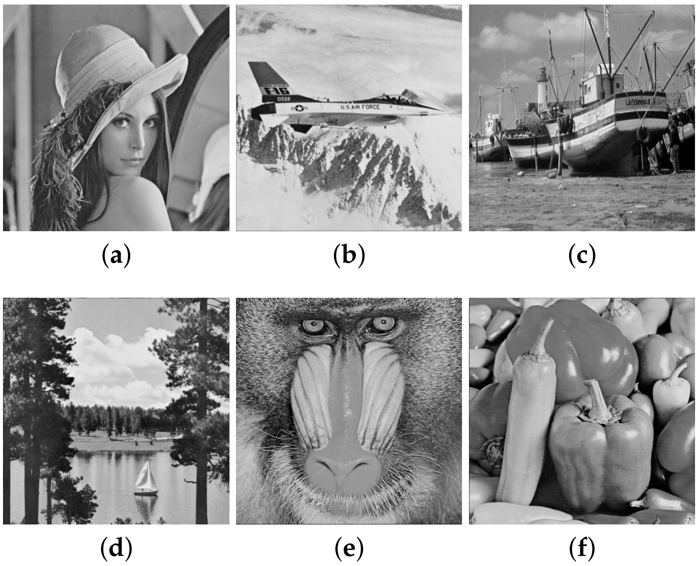


Figure 7. (**a**) Lena, (**b**) Airplane, (**c**) Boat, (**d**) Lake, (**e**) Baboon, (**f**) Pepper.

5. Results for Grayscale Images

The PSNR and SIMM are measures of image degradation caused by the watermark embedding, and the parameter α used in the embedded algorithm modifies the degradation of the original image. Table 1 shows the minimum and maximum values of PSNR and SIMM for several values of α for the 141 original images in the database. It is observed that increasing α (for $\alpha > 0$) slightly decreases the imperceptibility of the watermarked image. Table 2 shows similar results for FPR and FNR for several values of α for the 141 tampered images in the database with tampering rate 50% (the tampered Lena image with this tampering rate is illustrated in Figure 8e). We observe that these performance indicators slightly improve for $\alpha > 0$. Thus, this parameter provides a trade-off among these performance metrics. Hereafter, we fix the value of α to 0.01 in all simulations performed in this section.

Table 1. Minimum and maximum PSNR and SIMM for several values of α for the 141 images from the USC-SIPI database.

Metrics	$\alpha = 0$		$\alpha = 0.01$		$\alpha = 0.5$		$\alpha = 1$		$\alpha = 2$	
	Min	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
PSNR	∞	∞	37.46	51.04	37.37	50.98	37.34	50.92	37.32	50.80
SSIM	1	1	0.953	0.994	0.953	0.994	0.953	0.994	0.952	0.990

Table 2. Minimum and maximum FPR and FNR for several values of α for the 141 images with tampering rate 50%.

Metrics	$\alpha = 0$		$\alpha = 0.01$		$\alpha = 0.5$		$\alpha = 1$		$\alpha = 2$	
	Min	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
FPR	0.438	0.662	0.105	0.201	0.101	0.195	0.099	0.194	0.098	0.194
FNR	0.124	0.305	0	0.021	0	0.019	0	0.018	0	0.016

Table 3 shows PSNR comparisons between the algorithm Proposed 1 and several existing watermarking fragile methods. It can be seen that the proposed algorithm has better imperceptibility with PSNR higher than 47 dB for the images considered.

Table 3. PSNR comparison for several original images.

Scheme	PSNR					
	Lena	**Airplane**	**Boat**	**Lake**	**Pepper**	**Baboon**
Proposed 1	49.36	48.55	49.13	47.95	48.85	47.51
[12]	44.27	43.85	44.37	42.49	44.23	44.31
[14]	44.14	44.14	44.28	44.19	44.17	44.01
[17]	41.00	47.33	48.02	47.11	47.23	47.29
[15]	38.77	39.03	38.67	38.28	37.99	38.49
[22]	45.82	45.81	45.76	45.79	45.80	45.79
[23]	44.32	44.74	45.06	44.73	44.57	45.11
[20]	42.11	41.38	41.49	42.77	42.12	42.23

Figure 8 shows the tampered Lena images at various tampering rates, the corresponding binary detection images (the detected tampered region is marked in white color, whereas the non-tampered region is in black) and the recovered images. The quality of the recovered image is measured through the PSNR^r of the detected tampered region. Table 4 shows a comparison of PSNR^r versus tampering rates for several images, where it is seen that the algorithm Proposed 1 provides better recovery performance.

The time (in seconds) required to embed the parity bits into each 512 × 512 image shown in Figure 7 is in the range $[12.15, 14.68]$ (minimum and maximum values), while the range for the extraction of the parity bits is $[11.23, 13.46]$. The algorithms are implemented with the Matlab R2017b program on the Windows 10 Pro operating system running on a personal computer with 3.70 GHz Intel Xeon E5-1620 CPU and 64 GB RAM.

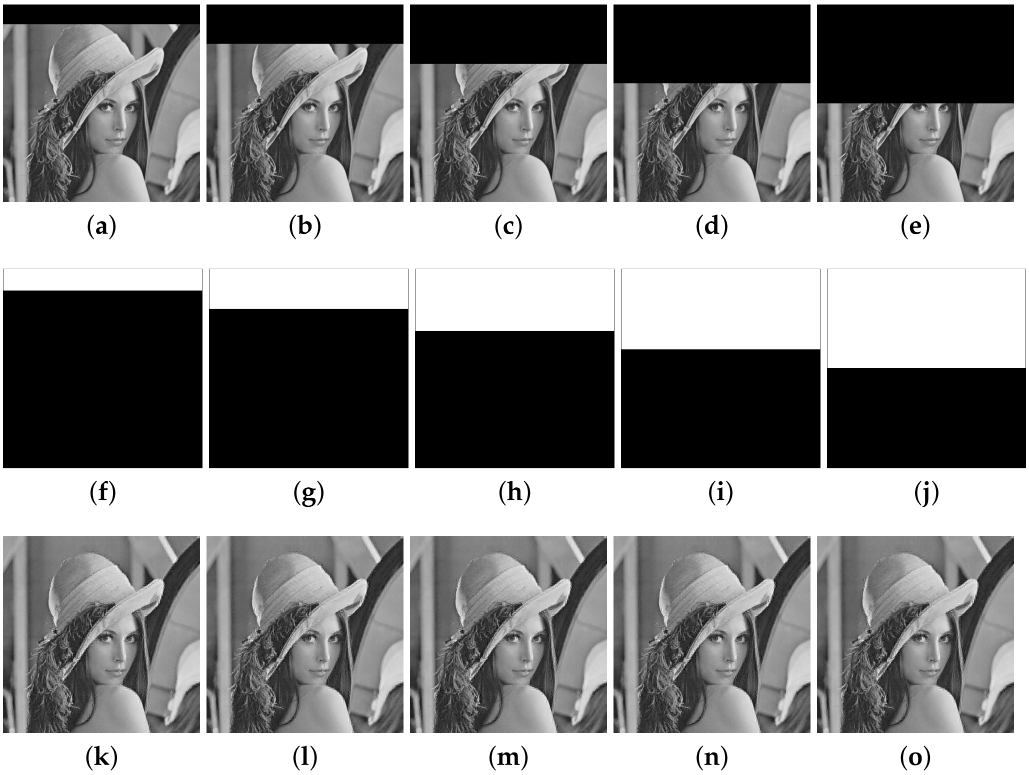


Figure 8. Tampered Lena images: (**a**) 10%, (**b**) 20%, (**c**) 30%, (**d**) 40%, (**e**) 50%. Binary detection images: (**f**) 10%, (**g**) 20%, (**h**) 30%, (**i**) 40%, (**j**) 50%. Recovered images: (**k**) 10%, (**l**) 20%, (**m**) 30%, (**n**) 40%, (**o**) 50%.

Table 4. PSNR^r versus tampered rate comparison for several original images.

Image	Scheme	Tampered Rate %				
		10	20	30	40	50
Lena	Proposed 1	51.35	48.78	47.18	45.52	43.12
	[22]	44.16	41.84	40.22	38.17	36.55
	[23]	40.52	37.60	35.89	31.92	29.32
	[17]	49.47	44.39	41.23	38.58	36.61
Baboon	Proposed 1	50.90	48.25	46.82	45.80	42.93
	[22]	41.18	42.58	41.03	38.45	34.82
	[23]	41.80	39.75	36.16	32.51	30.80
	[17]	38.69	35.55	33.95	32.93	32.13
Peppers	Proposed 1	51.08	48.80	46.88	45.07	43.05
	[22]	44.07	41.74	40.39	39.19	38.02
	[23]	41.35	39.60	35.97	33.05	31.68
	[17]	42.84	40.54	38.32	36.76	35.17
Airplane	Proposed 1	50.84	48.93	47.01	45.33	43.07
	[22]	41.99	40.24	38.57	36.99	35.95
	[23]	40.38	38.20	36.07	33.94	31.91
	[17]	46.59	44.54	42.83	40.32	36.79

The results for the CA_1 attack for the Airplane, Pepper, Lake and Countryside images are provided in Figure 9. In each row of this figure, the original image, the watermarked image with the PSNR value, the tampered image, the binary detection image with the FPR and FNR values, and the recovered image with the PSNR^r value are shown. The PSNR values for these four watermarked images are around 47 dB. The FPR and FNR are, respectively, 0.073 and 0.009 for Airplane, 0.117 and 0.008 for Pepper, 0.100 and 0.002 for Lake, 0.052 and 0.007 for Countryside, and which reveal good tampering detection performance. The Proposed 1 scheme can also achieve good image recovery results with PSNR^r around 41 dB for the Airplane, Pepper and Countryside images and around 47 dB for the Lake image. The CA_2 attack is considered in Figure 10 for the Baboon, Tree, Tank and Roof images. A portion of a watermarked image is copied in another watermarked image, preserving their relative spatial locations. In each row of this figure, two watermarked images with their PSNR values are shown, as well as the tampered image, the binary detection image with the values of FPR and FNR, and the recovered image with the PSNR^r value. The PSNR of the watermarked images are higher than 40 dB for these images. The FPR and FNR are respectively 0.099 and 0.007 for Baboon, 0.087 and 0.002 for Tree, 0.090 and 0.005 for Tank, 0.106 and 0.008 for Roof. The recovery results yield PSNR^r higher than 38 dB. The normal tampering attack is considered in Figure 11 in which some objects are added to the watermarked images (Lena, Elaine, Airport, and Aerial View).

The results for the CAA attack are presented in Figure 12 in which a distortion is created in a certain portion of the watermarked image. The obtained PSNR values are higher than 47 dB for the four images. The FPR and FNR are respectively 0.030, 0.007 for Boat, 0.025, 0.003 for Sailor, 0.026, 0.004 for Baboon, and 0.012, 0.001 for Zelda. The Salt and Pepper attack for the Lena image with $d = 0.3$ is considered in Figure 13. The FPR and FNR are 0.143 and 0.084, respectively.

Some attacks displayed in Figures 9, 10, 12 and 13 have also been considered in the literature. Table 5 compares the PSNR^r achieved by the algorithm Proposed 1 and by some existing methods. It is seen that the Proposed 1 technique provides, in some cases, better recovered performance for the considered attacks.

Table 5. PSNR^r achieved by the proposed algorithm and by some existing methods.

Figure	Image	PSNR^r		
		Proposed 1	Other Schemes	
Figure 9j	Pepper	40.98	[36]	33.59
Figure 9o	Lake	47.52	[20]	33.82
Figure 10e	Baboon	39.86	[20]	30.33
Figure 11o	Airport	47.42	[34]	46.03
Figure 12e	Boat	45.53	[14]	35.41
Figure 13e	Lena	33.78	[17]	40.68

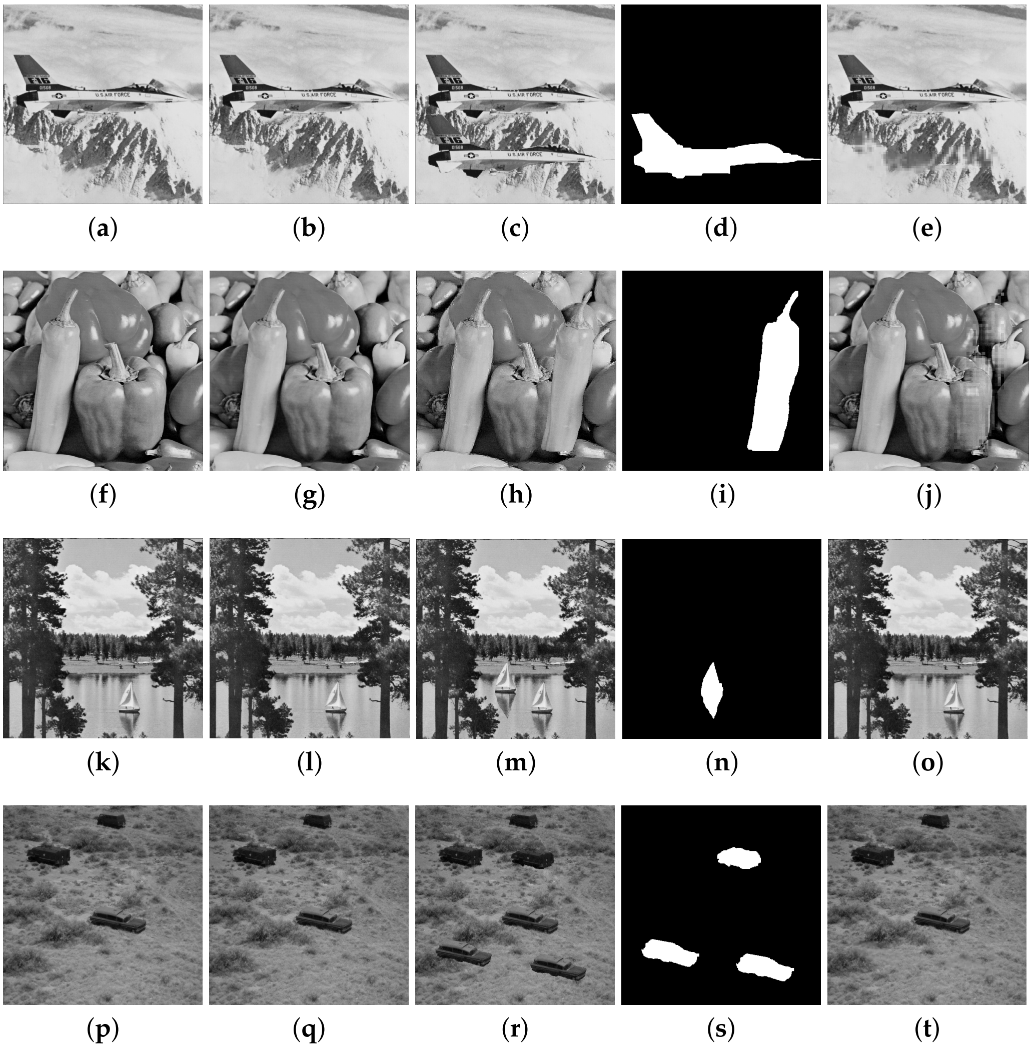


Figure 9. Tampering recovery for the CA_1 attack: (**a**) original Airplane image, (**b**) watermarked image (PSNR 48.55 dB), (**c**) tampered image (11%), (**d**) binary detection image (FPR = 0.073 and FNR = 0.009), (**e**) recovered image (PSNR^r = 41.02 dB). (**f**) original Pepper image, (**g**) watermarked image (PSNR = 48.85 dB), (**h**) tampered image (12%), (**i**) binary detection image (FPR = 0.117 and FNR = 0.008), (**j**) recovered image (PSNR^r = 40.98 dB). (**k**) original Lake image, (**l**) watermarked image (PSNR = 47.95 dB), (**m**) tampered image (2%), (**n**) binary detection image (FPR = 0.100 and FNR = 0.002), (**o**) recovered image (PSNR^r = 47.52 dB). (**p**) original Countryside image, (**q**) watermarked image (PSNR = 46.13 dB), (**r**) tampered image (6.4%), (**s**) binary detection image (FPR = 0.052 and FNR = 0.007), (**t**) recovered image (PSNR^r = 42.17 dB).

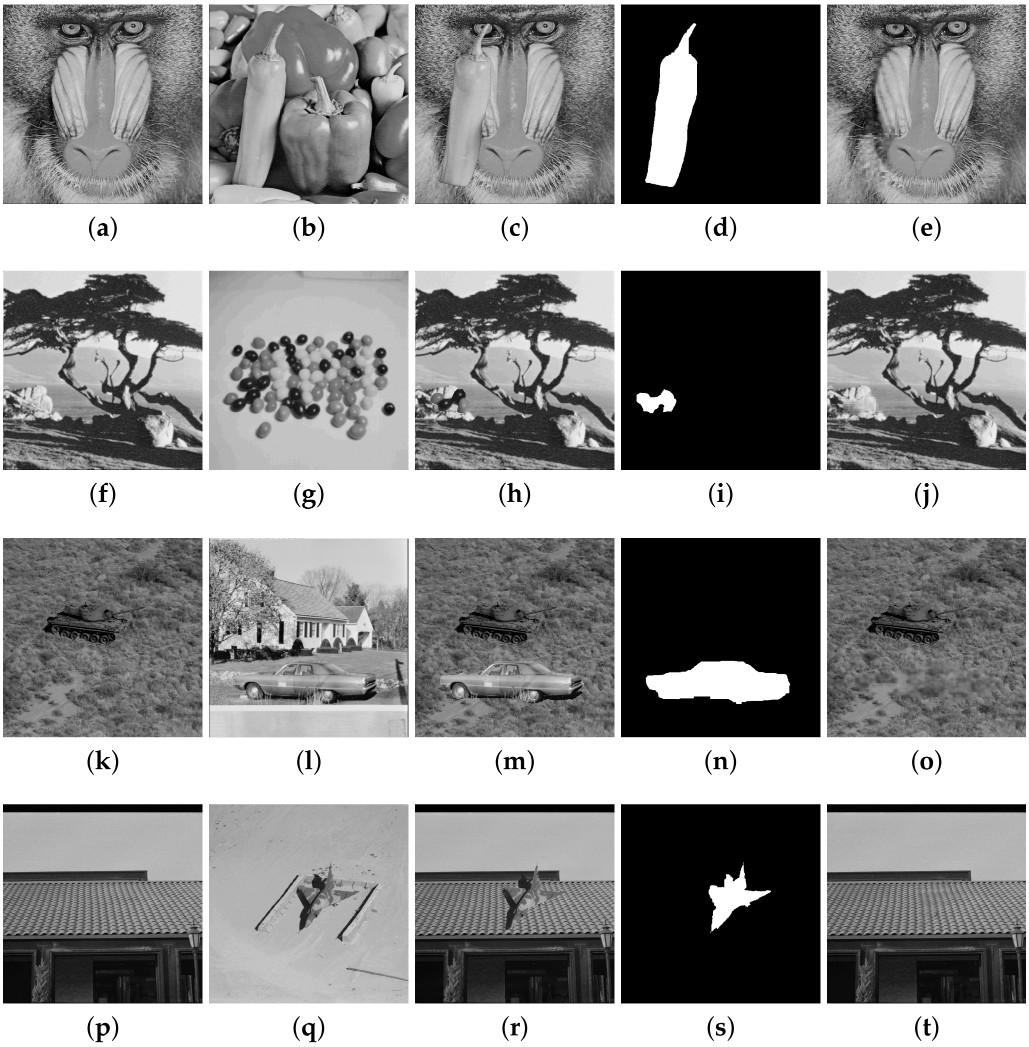


Figure 10. Tampering recovery for the CA_2 attack: (**a**) watermarked Baboon image (PSNR = 47.51 dB), (**b**) watermarked Pepper image (PSNR = 48.85 dB), (**c**) tampered image (12.2%), (**d**) binary detection image (FPR $=$ 0.099 and FNR $=$ 0.007), (**e**) recovered image (PSNR^r = 39.86 dB). (**f**) watermarked Tree image (PSNR = 41.15 dB), (**g**) watermarked Seeds image (PSNR = 40.23 dB), (**h**) tampered image (1.40%), (**i**) binary detection image (FPR = 0.087 and FNR = 0.002), (**j**) recovered image (PSNR^r = 46.32 dB). (**k**) watermarked Tank image (PSNR = 44.33 dB), (**l**) watermarked Car image (PSNR = 42.56 dB), (**m**) tampered image (10.70%), (**n**) binary detection image (FPR = 0.090 and FNR = 0.005), (**o**) recovered image (PSNR^r = 38.56 dB). (**p**) watermarked Roof image, (**q**) watermarked Airplane image (PSNR = 45.98 dB), (**r**) tampered image (4%), (**s**) binary detection image (FPR = 0.106 and FNR = 0.008), (**t**) recovered image (PSNR^r = 43.55 dB).

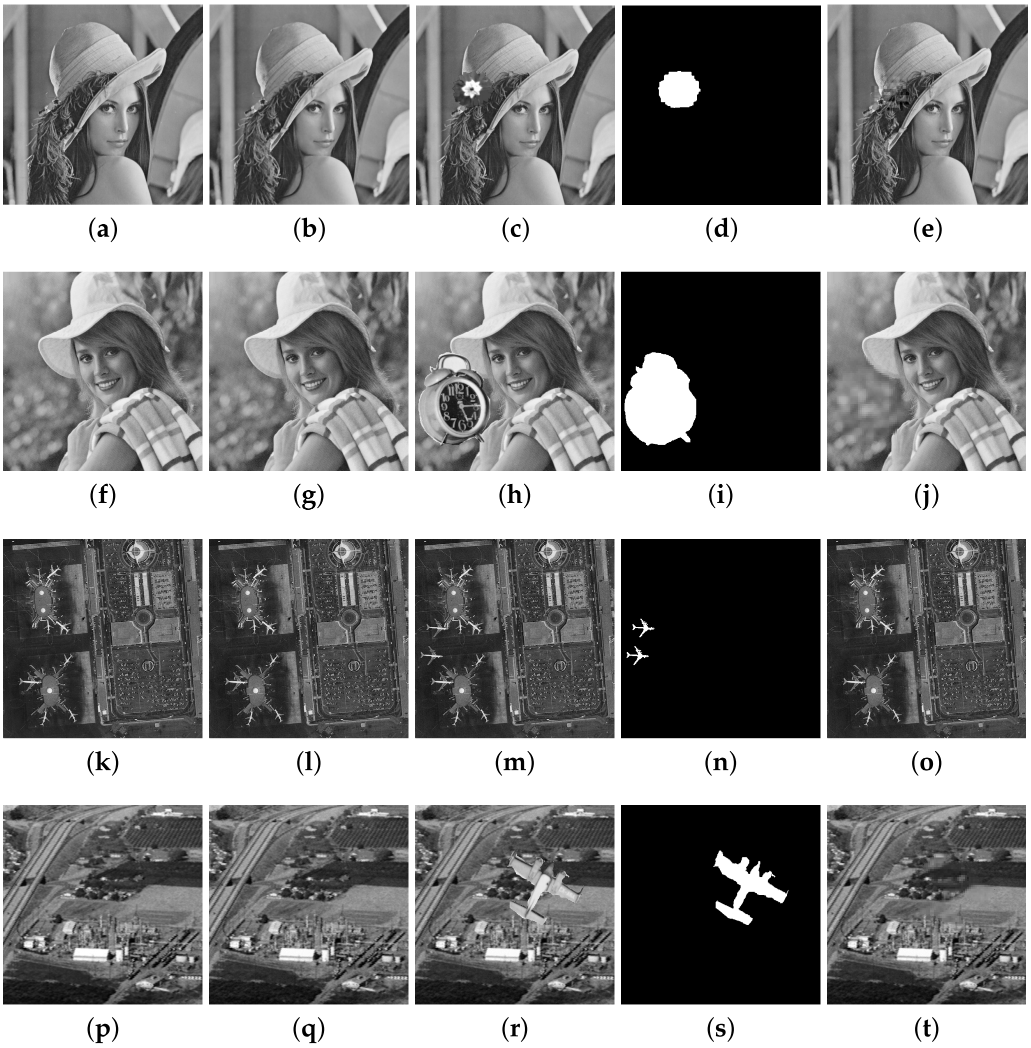
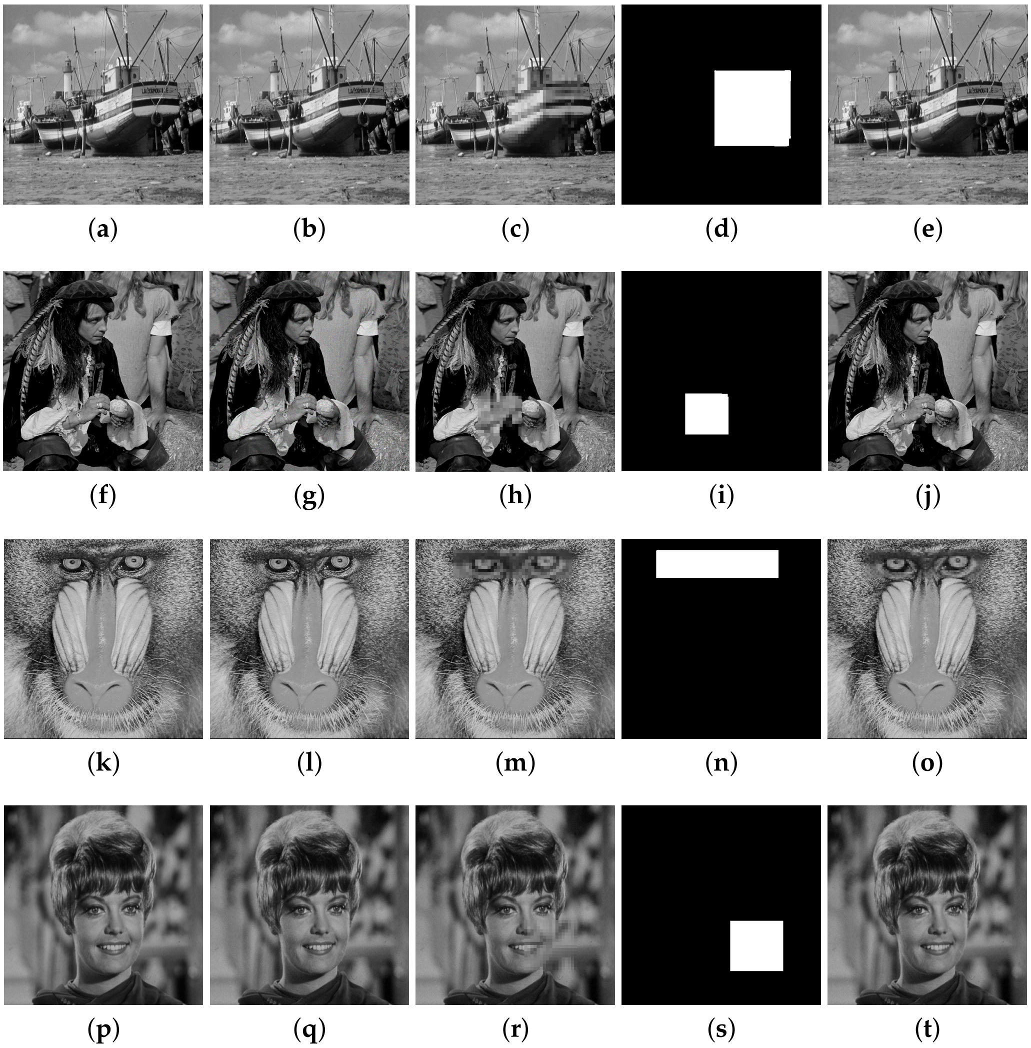


Figure 11. Tampering recovery for normal tampering attack: (**a**) original Lena image, (**b**) watermarked Lena image (PSNR = 49.36 dB), (**c**) tampered image (3%), (**d**) binary detection image (FPR = 0.035 and FNR = 0.003), (**e**) recovered image (PSNR^r = 47.25 dB). (**f**) original Elaine image, (**g**) watermarked Elaine image (PSNR = 43.36 dB), (**h**) tampered image (12.56%), (**i**) binary detection image (FPR = 0.103 and FNR = 0.008), (**j**) recovered image (PSNR^r = 40.28 dB). (**k**) original Airport image, (**l**) watermarked image (PSNR = 44.00 dB), (**m**) tampered image (2%), (**n**) binary detection image (FPR = 0.020 and FNR = 0.001), (**o**) recovered image (PSNR^r= 47.42 dB). (**p**) original Aerial View image, (**q**) watermarked image (PSNR = 44.10 dB), (**r**) tampered image (4.8%), (**s**) binary detection image (FPR = 0.135 and FNR = 0.007), (**t**) recovered image (PSNR = 46.12 dB).



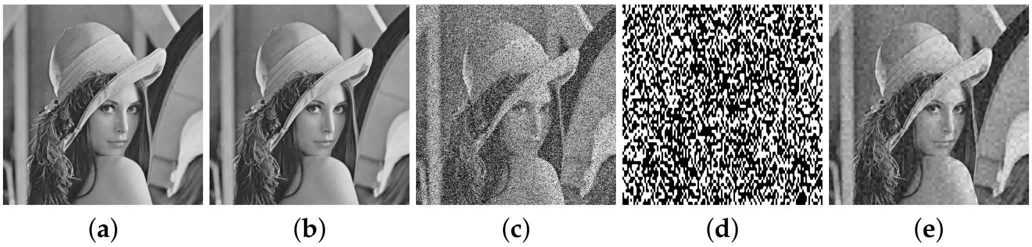
(**a**) (**b**) (**c**) (**d**) (**e**)

(**f**) (**g**) (**h**) (**i**) (**j**)

(**k**) (**l**) (**m**) (**n**) (**o**)

(**p**) (**q**) (**r**) (**s**) (**t**)

Figure 12. Tampering recovery for the CAA attack: (**a**) original Boat image, (**b**) watermarked image (PSNR = 49.13 dB), (**c**) tampered image (15%), (**d**) binary detection image (FPR = 0.030 and FNR = 0.007), (**e**) recovered image (PSNR^r = 45.53 dB). (**f**) Sailor original image, (**g**) watermarked image (PSNR = 48.27 dB), (**h**) tampered image (5%), (**i**) binary detection image (FPR = 0.025 and FNR = 0.003), (**j**) recovered image (PSNR^r = 44.92 dB). (**k**) original Baboon image, (**l**) watermarked image (PSNR = 47.51 dB), (**m**) tampered image (8.5%), (**n**) binary detection image (FPR = 0.026 and FNR = 0.004), (**o**) recovered image (PSNR^r = 44.31 dB). (**p**) original Zelda image, (**q**) watermarked image (PSNR = 47.04 dB), (**r**) tampered image (7%), (**s**) binary detection image (FPR = 0.012 and FNR = 0.001), (**t**) recovered image (PSNR^r = 45.18 dB).



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 13. Salt and Pepper attack for the Lena image (**a**) original image, (**b**) watermarked image (PSNR = 49.36 dB), (**c**) tampered image (30%), (**d**) binary detection image (FPR = 0.143 and FNR = 0.084), (**e**) recovered image (PSNR^r = 33.78 dB).

A BCH Code (31,21,2)

To analyze the impact of the BCH code in the proposed watermarking scheme, we consider the BCH (31,21,2). This code has a greater number of parity bits than the BCH (15,11,1) code, so it is necessary to take into account a greater number of sub-blocks where the parity bits are embedded. The modified algorithm uses the same steps as the previous one, modifying the code used and the level of the DWT, according to the following steps.

1. Repeat this step of the previous algorithm.
2. Apply the one-level DWT decomposition to the original image C_O. The sub-bands C_{LH_1} and C_{HL_1} (each one of size $M/2 \times N/2$ pixels) are divided into sub-blocks of size 2×2, where the watermarked bits are embedded. The total number of sub-blocks is $\frac{MN}{16}$.
3. Repeat this step of the previous algorithm.
4. Construct the sequence $\mathbf{p}$ from the BCH (31,21,2) code as follows. The 21 information bits are obtained by concatenating $k_1 = 2$ bits from ℓ_4 and $k_2 = 19$ from the chaotic map. After scrambling, we obtain $\mathbf{p} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{\frac{MN}{64}}\}$, where $\mathbf{p}_i = p_{i1}, p_{i2}$. Each $\mathbf{p}_i$ is embedded into some sub-blocks of C_{LH_1} and C_{HL_1}.
5. Repeat this step of the previous algorithm. Since there are four times more sub-blocks than subsequences $\mathbf{p}_i$, after inserting a given $\mathbf{p}_i$, the next three sub-blocks are not used by the embedded algorithm.
6. Apply the 1-level IDWT and obtain the watermarked image C_W.

Table 6 shows the PSNR comparison between the algorithm presented in the previous sections (Proposed 1) and the modified one (called Proposed 1-v1). A decrease in the PSNR value is observed due to the insertion at a higher level of the DWT decomposition (1-level). Table 7 presents a PSNR^r comparison for several tampering rates, observing a slight increase in the value of PSNR^r. Table 8 compares the PSNR^r for the attacks displayed in Figures 9–13. We observe that the modified algorithm presents a better recovery performance for these attacks. This is due to the code modification.

Table 6. PSNR comparison for several original images.

Scheme	PSNR					
	Lena	**Airplane**	**Boat**	**Lake**	**Pepper**	**Baboon**
Proposed 1	49.36	48.55	49.13	47.95	48.85	47.51
Proposed 1-v1	47.52	46.28	47.07	45.67	46.28	45.21

Table 7. PSNR^r versus tampered rate comparison for several original images.

Image	Scheme	Tampered Rate %				
		10	**20**	**30**	**40**	**50**
Lena	Proposed 1	51.35	48.78	47.18	45.52	43.12
	Proposed 1-v1	52.38	49.80	48.52	46.10	44.00
Baboon	Proposed 1	50.90	48.25	46.82	45.80	42.93
	Proposed 1-v1	51.00	48.42	47.01	48.92	43.01
Peppers	Proposed 1	51.08	48.80	46.88	45.07	43.05
	Proposed 1-v1	51.19	48.95	47.01	45.19	43.18
Airplane	Proposed 1	50.84	48.93	47.01	45.33	43.07
	Proposed 1-v1	50.96	49.07	47.13	45.49	43.16

Table 8. PSNR^r comparison for several attacks of proposed algorithms.

Figure	Image	PSNR^r	
		Proposed 1	Proposed 1-v1
Figure 9j	Peppers	40.98	43.16
Figure 9o	Lake	47.52	49.83
Figure 10e	Baboon	39.86	42.07
Figure 11o	Airport	47.42	49.28
Figure 12e	Boat	45.53	46.90
Figure 13e	Lena	33.78	35.21

6. Performance of the Proposed Algorithm for Colored Images

The performance of the proposed algorithms in colored images is analyzed in terms of imperceptibility, detection and recovery. We also present comparisons with literature results. The original colored image C_O is represented by three components R, G, and B, each one of size 512 × 512. A fragile watermarking algorithm in a grayscale image is applied in each component. We adopt the same performance metrics used for grayscale images. Table 9 presents an imperceptibility comparison between the proposed algorithms and some existing ones for several images. All proposed algorithms present better imperceptibility results, and the highest PSNR values are achieved by Proposed 1. Table 10 shows a comparison of PSNR^r versus several tampered rates. Some attacks presented in the previous chapter are presented in Figure 14 with recovered results for the algorithm Proposed 1. Behavior similar to that obtained with grayscale images is observed, obtaining FNR and FPR values close to zero (desired values) and PSNR^r values higher than 34 dB.

Table 9. PSNR comparison for several original colored images.

Scheme	PSNR					
	Lena	Airplane	House	Sailboat	Pepper	Baboon
Proposed 1	51.26	51.27	51.07	51.10	51.21	50.98
Proposed 1-v1	51.08	51.11	50.98	51.08	51.13	50.70
[22]	46.45	46.23	46.22	46.18	46.03	46.24
[16]	44.60	44.69	44.66	44.61	44.54	44.64
[19]	46.37	48.32	46.23	47.12	46.3	46.17

Table 10. PSNR^r versus tampered rate comparison for several color original images.

Image	Scheme	Tampered Rate %				
		10	20	30	40	50
Lena	Proposed 1	52.08	49.13	48.02	46.13	44.28
	Proposed 1-v1	52.31	49.82	48.53	46.68	45.04
	[22]	44.22	39.77	37.64	35.91	34.80
	[16]	37.16	33.83	31.48	29.07	26.96
	[33]	49.47	44.39	41.23	38.58	36.61
Baboon	Proposed 1	51.23	48.97	47.86	46.20	44.40
	Proposed 1-v1	51.76	49.28	48.61	47.55	45.88
	[22]	42.00	38.05	37.05	35.00	32.50
	[16]	35.85	31.87	28.38	25.59	23.59
	[33]	29.50	26.77	24.98	22.99	21.66
Peppers	Proposed 1	52.00	48.92	47.90	46.77	44.16
	Proposed 1v-1	53.60	49.71	48.96	47.90	45.73
	[22]	44.02	40.00	39.20	37.00	35.92
	[16]	37.38	34.63	32.48	29.89	27.31
	[33]	35.67	32.36	30.07	28.62	27.24
Airplane	Proposed 1	51.72	49.28	47.91	46.60	44.17
	Proposed 1-v1	52.66	50.93	49.08	47.95	45.78
	[22]	41.90	40.00	39.00	36.95	35.00
	[16]	36.51	33.40	31.28	28.51	25.99
	[33]	42.72	34.81	30.24	28.16	26.42

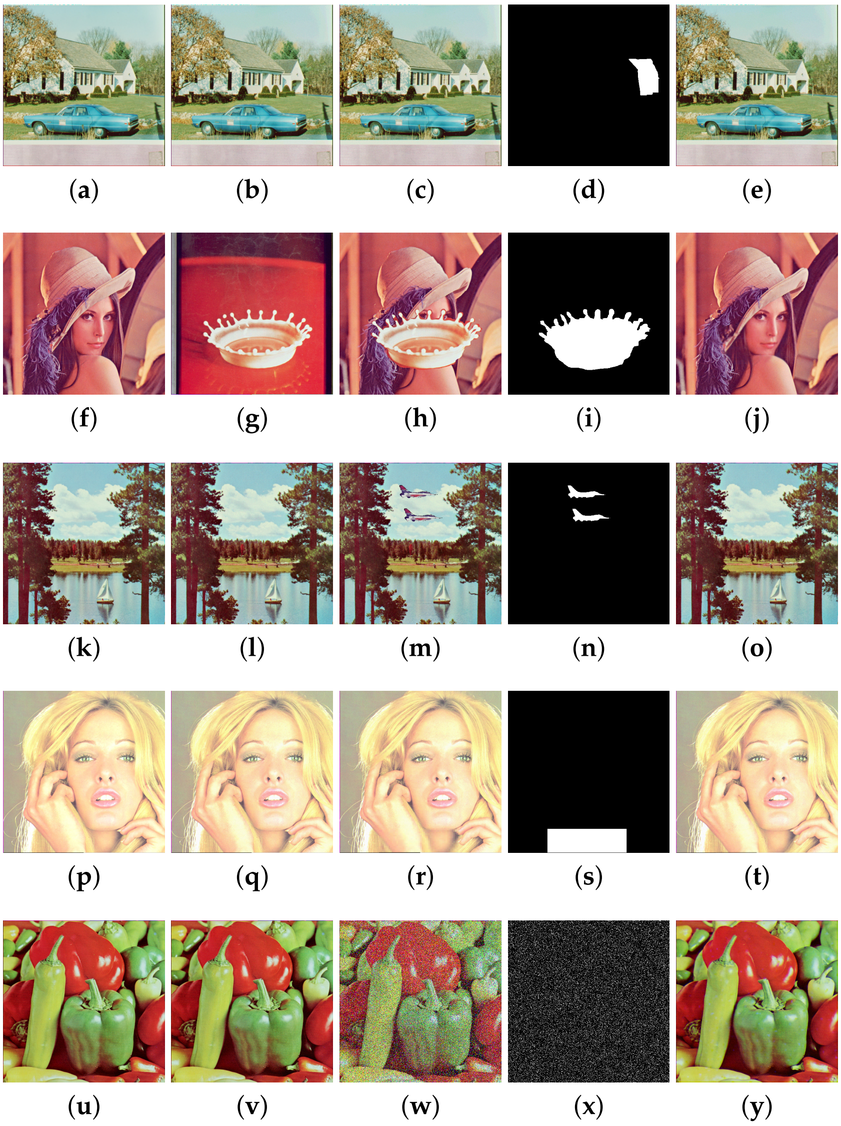


Figure 14. Different attacks on colored images (a-e) tampering recovery for the CA1 attack (**a**) original Car image, (**b**) watermarked Car image (PSNR = 47.35 dB), (**c**) tampered image (3%), (**d**) binary detection image (FPR = 0.069 and FNR = 0.008), (**e**) recovered image (PSNR^r = 51.61 dB), (**f–j**) tampering recovery for the CA_2 attack: (**f**) watermarked Lena image (PSNR = 47.26 dB), (**g**) watermarked Splash image (PSNR = 46.75 dB), (**h**) tampered image (16%), (**i**) binary detection image (FPR = 0.083 and FNR = 0.011), (**j**) recovered image (PSNR^r = 47.56 dB), (**k–o**) tampering recovery for normal tampering attack: (**k**) original Lake image, (**l**) watermarked Lake image (PSNR = 46.42 dB), (**m**) tampered image (1.5%), (**n**) binary detection image (FPR = 0.051 and FNR = 0.005), (**o**) recovered Tiffany image (PSNR^r = 49.34 dB), (**p–t**) tampering recovery for the CAA attack: (**p**) original Tiffany image, (**q**) watermarked image (PSNR = 47.25 dB), (**r**) tampered image (7%), (**s**) binary detection image (FPR = 0.023 and FNR = 0.004), (**t**) recovered image (PSNR^r = 51.04 dB), (**u–y**) Salt and Pepper attack (**u**) original Pepper image, (**v**) watermarked Pepper image PSNR = 48.01 dB, (**w**) tampered image (30%), (**x**) binary detection image (FPR = 0.103 and FNR = 0.062), (**y**) recovered image (PSNR^r = 34.08 dB).

7. Conclusions

This article presented a new self-embedding fragile watermarking algorithm for tamper detection and content recovery in images. The watermarked bits are the parity bits of a BCH code, in which its information sequence is composed of chaotic bits and bits obtained from the original image. The watermarked bits are embedded in the original image in the frequency domain using the DWT. The parameter α establishes a trade-off between imperceptibility and recovery. After investigating the trade-off between the imperceptibility, detection of tampered areas, and recovery capability of the algorithm, we compare its performance with that of some existing schemes. We conclude that the algorithm is competitive in terms of several metrics, such as PSNR, SIMM, FPR, FNR, and PSNR^r. The joint application of chaotic bits and BCH codes not only contributes to the recovery of the image information in the tampered areas, but also provides security, and the existence of a greater number of parity bits leads to higher recoverability. A natural continuation of this work is the incorporation of codes with unequal error protection, since part of the information bits is known at the extraction algorithm. Another topic for future research is to consider chaotic maps with high nonlinearities and constant chaos for a wide parameter range [51].

Author Contributions: Conceptualization, D.P.B.C. and C.P.; methodology, J.A.P.A. and A.M.R.; software, J.A.P.A. and A.M.R.; validation, J.A.P.A., A.M.R. and C.P.; formal analysis, D.P.B.C.; investigation, D.P.B.C. and C.P.; resources, A.M.R.; data curation, J.A.P.A. and A.M.R.; writing—original draft preparation, D.P.B.C. and C.P.; writing—review and editing, D.P.B.C. and C.P.; visualization, J.A.P.A. and A.M.R.; supervision, D.P.B.C. and C.P.; project administration, D.P.B.C. and C.P.; funding acquisition, D.P.B.C. and C.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under Grants 306662/2021-3 and 423008/2021-8, in part by the Foundation for Support of Science and Technology of the State of Pernambuco (FACEPE) under Grant APQ-0291-3.04/14, and in part by the Coordination for the Improvement of Higher Education Personnel (CAPES)-Finance Code 001, Brazil.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors are thankful to the reviewers for their constructive comments and useful suggestions for improving the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Naskar, R.; Chakraborty, R.S. *Reversible Digital Watermarking: Theory and Practices*; Morgan & Claypool Publishers: Williston, VT, USA, 2014.
2. Rakhmawati, L.; Wirawan, W.; Suwadi, S. A recent survey of self-embedding fragile watermarking scheme for image authentication with recovery capability. *EURASIP J. Image Video Process.* **2019**, *2019*, 61. [CrossRef]
3. Gul, E.; Ozturk, S. A novel pixel-wise authentication-based self-embedding fragile watermarking method. *Multimed. Syst.* **2021**, *27*, 531–545. [CrossRef]
4. Kosuru, D.; Swain, G.; Kumar, N.; Pradhan, A. Image tamper detection and correction using Merkle tree and remainder value differencing. *Optik* **2022**, *261*, 169–212.
5. Shih, F.Y. *Image Processing and Pattern Recognition: Fundamentals and Techniques*; John Wiley & Sons: Hoboken, NJ, USA, 2010.
6. Moosazadeh, M.; Ekbatanifard, G. A new DCT-based robust image watermarking method using teaching-learning-Based optimization. *J. Inf. Secur. Appl.* **2019**, *47*, 28–38. [CrossRef]
7. Ko, H.J.; Huang, C.T.; Horng, G.; Wang, S.J. Robust and blind image watermarking in DCT domain using inter-block coefficient correlation. *Inf. Sci.* **2020**, *517*, 128–147. [CrossRef]
8. Zhou, X.; Ma, Y.; Mohammed, M.A.; Damaševičius, R. A reversible watermarking system for medical colored images: Balancing capacity, imperceptibility, and robustness. *Electronics* **2021**, *10*, 1024. [CrossRef]
9. Gul, E.; Toprak, A.N. Contourlet and discrete cosine transform based quality guaranteed robust image watermarking method using artificial bee colony algorithm. *Expert Syst. Appl.* **2023**, *212*, 118730. [CrossRef]
10. Peng, Y.; Niu, X.; Fu, L.; Yin, Z. Image authentication scheme based on reversible fragile watermarking with two images. *J. Inf. Secur. Appl.* **2018**, *40*, 236–246. [CrossRef]
11. Qin, C.; Wang, H.; Zhang, X.; Sun, X. Self-embedding fragile watermarking based on reference-data interleaving and adaptive selection of embedding mode. *Inf. Sci.* **2016**, *373*, 233–250. [CrossRef]

12. Qin, C.; Ji, P.; Zhang, X.; Dong, J.; Wang, J. Fragile Image Watermarking With Pixel-wise Recovery Based on Overlapping Embedding Strategy. *Signal Process.* **2017**, *138*, 280–293. [CrossRef]
13. Sreenivas, K.; Kamakshiprasad, V. Improved image tamper localisation using chaotic maps and self-recovery. *J. Vis. Commun. Image Represent.* **2017**, *49*, 164–176. [CrossRef]
14. Tai, W.; Liao, Z. Image self-recovery with watermark self-embedding. *Signal Process. Image Commun.* **2018**, *65*, 11–25. [CrossRef]
15. Abdelhakim, A.; Saleh, H.; Abdelhakim, M. Fragile watermarking for image tamper detection and localization with effective recovery capability using K-means clustering. *Multimed. Tools Appl.* **2019**, *78*, 32523–32563. [CrossRef]
16. Molina, J.; Garcia, B.; Ponomaryov, V.; Reyes, R.; Sadovnychiy, S.; Cruz, C. An effective fragile watermarking scheme for colored image tampering detection and self-recovery. *Signal Process. Image Commun.* **2020**, *81*, 115725.
17. Lee, C.; Shen, J.; Chen, Z.; Agrawal, S. Self-Embedding authentication watermarking with effective tampered location detection and high-quality image recovery. *Sensors* **2019**, *19*, 2267. [CrossRef] [PubMed]
18. Sarreshtedari, S.; Akhaee, M.A.; Abbasfar, A. Source channel coding-based watermarking for self-embedding of JPEG images. *Signal Process. Image Commun.* **2018**, *62*, 106–116. [CrossRef]
19. Al-Otum, H.M.; Ellubani, A.A.A. Secure and effective colored image tampering detection and self restoration using a dual watermarking approach. *Optik* **2022**, *262*, 169280. [CrossRef]
20. Wu, H.C.; Fan, W.L.; Tsai, C.S.; Ying, J.J.C. An image authentication and recovery system based on discrete wavelet transform and convolutional neural networks. *Multimed. Tools Appl.* **2022**, *81*, 19351–19375. [CrossRef]
21. Klington, A.G.; Ramesh, K.; Kadry, S. Cost-Effective watermarking scheme for authentication of digital fundus images in healthcare data management. *Inf. Technol. Control* **2021**, *50*, 645–655. [CrossRef]
22. Bolourian Haghighi, B.; Taherinia, A.H.; Mohajerzadeh, A.H. TRLG: Fragile blind quad watermarking for image tamper detection and recovery by providing compact digests with optimized quality using LWT and GA. *Inf. Sci.* **2019**, *486*, 204–230. [CrossRef]
23. Jafari Barani, M.; Yousefi Valandar, M.; Ayubi, P. A new digital image tamper detection algorithm based on integer wavelet transform and secured by encrypted authentication sequence with 3D quantum map. *Optik* **2019**, *187*, 205–222. [CrossRef]
24. Azeroual, A.; Afdel, K. Real-time image tamper localization based on fragile watermarking and Faber-Schauder wavelet. *AEU-Int. J. Electron. Commun.* **2017**, *79*, 207–218. [CrossRef]
25. Raj, N.N.; Shreelekshmi, R. Fragile watermarking scheme for tamper localization in images using logistic map and singular value decomposition. *J. Vis. Commun. Image Represent.* **2022**, *85*, 103500.
26. Rawat, S.; Raman, B. A chaotic system based fragile watermarking scheme for image tamper detection. *AEU-Int. J. Electron. Commun.* **2011**, *65*, 840–847. [CrossRef]
27. Li, M.; Xiao, D.; Liu, H.; Bai, S. A recoverable chaos-based fragile watermarking with high PSNR preservation. *Secur. Commun. Netw.* **2016**, *9*, 237–238. [CrossRef]
28. Lefévre, P.; Carré, P.; Gaborit, P. Application of rank metric codes in digital image watermarking. *Signal Process. Image Commun.* **2019**, *74*, 119–128. [CrossRef]
29. Fan, M.; Wang, H. An enhanced fragile watermarking scheme to digital image protection and self-recovery. *Signal Process. Image Commun.* **2018**, *66*, 19–29. [CrossRef]
30. Qin, C.; Ji, P.; Wang, J.; Chang, C.C. Fragile image watermarking scheme based on VQ index sharing and self-embedding. *Multimed. Tools Appl.* **2017**, *76*, 226–228. [CrossRef]
31. Hsu, C.S.; Tu, S.F. Image tamper detection and recovery using adaptive embedding rules. *Measurement* **2016**, *88*, 287–296. [CrossRef]
32. Singh, D.; Singh, S.K. Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability. *J. Vis. Commun. Image Represent.* **2016**, *38*, 775–789. [CrossRef]
33. Sinhal, R.; Ansari, I.A.; Ahn, C.W. Blind image watermarking for localization and restoration of color images. *IEEE Access* **2020**, *8*, 57–69. [CrossRef]
34. Yuan, X.; Li, X.; Liu, T. Gauss Jordan elimination-based image tampering detection and self-recovery. *Signal Process. Image Commun.* **2021**, *90*, 11–60. [CrossRef]
35. Rezaei, M.; Taheri, H. Digital image self-recovery using CNN networks. *Optik* **2022**, *264*, 169–345. [CrossRef]
36. Rajput, V.; Ansari1, I. Image tamper detection and self-recovery using multiple median watermarking. *Multimed. Tools Appl.* **2020**, *79*, 35519–35535. [CrossRef]
37. Tan, Y.; Jiaohua, Q. A robust watermarking scheme in YCbCr color space based on channel coding. *IEEE Access* **2019**, *7*, 25026–25036. [CrossRef]
38. Lefévre, P.; Carré, P.; Fontaine, C.; Gaborit, P.; Huang, J. Efficient image tampering localization using semi-fragile watermarking and error control codes. *Signal Process.* **2022**, *190*, 108342. [CrossRef]
39. Lin, S.; Costello, D.J. *Error Control Coding*, 2nd ed.; Prentice Hall: Hoboken, NJ, USA, 2004.
40. Strogatz, S.H. *Nonlinear Dynamics and Chaos*, 1st ed.; Studies in Nonlinearity; Westview Press: Boulder, CO, USA, 2001.
41. Lan, R.; He, J.; Wang, S.; Liu, Y.; Luo, X. A parameter-selection-based chaotic system. *IEEE Trans. Circuits Syst. Ii Express Briefs* **2019**, *66*, 492–496. [CrossRef]
42. Zhou, Y.; Hua, Z.; Pun, C.M.; Philip Chen, C.L. Cascade chaotic system with applications. *IEEE Trans. Cybern.* **2015**, *45*, 2001–2012. [CrossRef]

43. Callegari, S.; Fabbri, M.; Beirami, A. Very low cost chaos-based entropy source for the retrofit or design augmentation of networked devices. *Analog. Integr. Circuits Signal Process.* **2016**, *87*, 155–167. [CrossRef]
44. Atawneh, S.; Almomani, A.; Al Bazar, H.; Sumari, P.; Gupta, B. Secure and imperceptible digital image steganographic algorithm based on diamond encoding in DWT domain. *Multimed. Tools Appl.* **2017**, *76*, 51–72. [CrossRef]
45. Gangadhar, Y.; Giridhar Akula, V.S.; Reddy, P.C. An evolutionary programming approach for securing medical images using watermarking scheme in invariant discrete wavelet transformation. *Biomed. Signal Process. Control* **2018**, *43*, 31–40. [CrossRef]
46. Farghaly, S.H.; Ismail, S.M. Floating-point discrete wavelet transform-based image compression on FPGA. *AEU-Int. J. Electron. Commun.* **2020**, *124*, 53–63. [CrossRef]
47. Al-Shayea, T.K.; Mavromoustakis, C.X.; Batalla, J.M.; Mastorakis, G. A hybridized methodology of different wavelet transformations targeting medical images in IoT infrastructure. *Measurement* **2019**, *148*, 106–813. [CrossRef]
48. El-Hoseny, H.M.; El Kareh, Z.Z.; Mohamed, W.A.; El Banby, G.M.; Mahmoud, K.R.; Faragallah, O.S.; El-Rabaie, S.; El-Madbouly, E.; Abd El-Samie, F.E. An optimal wavelet-based multi-modality medical image fusion approach based on modified central force optimization and histogram matching. *Multimed. Tools Appl.* **2019**, *78*, 73–97. [CrossRef]
49. Thakkar, F.N.; Srivastava, V.K. A fast watermarking algorithm with enhanced security using compressive sensing and principle components and its performance analysis against a set of standard attacks. *Multimed. Tools Appl.* **2017**, *76*, 191–219. [CrossRef]
50. Stephane, M. *A Wavelet Tour of Signal Processing*; Elsevier: Berlin/Heidelberg, Germany, 2009.
51. Hua, Z.; Zhou, B.; Zhou, Y. Sine chaotification model for enhancing chaos and its hardware implementation. *IEEE Trans. Ind. Electron.* **2019**, *66*, 1273–1284. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.