applied sciences

MDPI

Article

Evaluation of a Framework for Robust Image Reversible Watermarking

Jose Juan Garcia-Hernandez [1,*], Claudia Feregrino-Uribe [2], Alejandra Menendez-Ortiz [2] and Dan Williams Robledo-Cruz [1]

[1] Cinvestav Unidad Tamaulipas, Parque Cientifico y Tecnologico TECNOTAM, Km. 5.5 Carr. a Soto la Marina, Ciudad Victoria CP 87130, Mexico; drobledo@cinvestav.mx
[2] Coordinacion de Ciencias Computacionales, INAOE Luis Enrique Erro #1, Sta. Ma. Tonantzintla, Puebla CP 72840, Mexico; cferegrino@ccc.inaoep.mx (C.F.-U.); m.menendez@ccc.inaoep.mx (A.M.-O.)
* Correspondence: jjuan.garcia@cinvestav.mx

Abstract: In the literature, robust reversible watermarking schemes (RWSs) allow the extraction of watermarks after the images have suffered attacks; however, the modified images are compromised. On the other hand, self-recovery schemes will restore the compromised regions of the images, but no secret messages are inserted in these schemes. A framework for robust reversible watermarking with signal restoration capabilities was previously proposed in the literature. This study selects four fragile reversible watermarking techniques and two self-recovery schemes to design different framework configurations. These configurations are evaluated to test the framework's performance and determine the structure that yields better results in terms of perceptual transparency using a well-known image database as the signal input. It was found that fragile reversible watermarking schemes hold low perceptual distortion, while self-recovery schemes produce high perceptual distortion levels. The inherent characteristics of each algorithm determine, a priori, the behavior of the framework, which is approximated by a proposed equation.

Keywords: robust reversible watermarking; signal restoration; self-recovery watermarking

Citation: Garcia-Hernandez, J.J.; Feregrino-Uribe, C.; Menendez-Ortiz, A.; Robledo-Cruz, D.W. Evaluation of a Framework for Robust Image Reversible Watermarking. *Appl. Sci.* **2022**, *12*, 7242. https://doi.org/10.3390/app12147242

Academic Editors: David Megías, Minoru Kuribayashi and Wojciech Mazurczyk

Received: 14 June 2022
Accepted: 8 July 2022
Published: 18 July 2022

1. Introduction

Due to available fast Internet connections, massive access to multimedia material is common currently. However, this ease of connection produces some risks in the reliability of the accessed material. Although different strategies have been proposed, in particular, digital watermarking has been shown to be an appropriate option for the protection of multimedia content. Despite its protection capabilities, conventional digital watermarking produces distortion in the host signal. These distortions are not acceptable for critical applications such as medical or military imaging [1]. Reversible watermarking schemes (RWSs) emerge as an option to overcome the distortion induced by classical marking schemes. RWSs allow the host signal to be reconstructed to its original state after removing the inserted watermark.

On the other hand, most of the RWSs are not very robust to attacks. If a tagged signal is attacked, the recovery of the information and the reconstruction of the host signal will be very difficult. Recently, some robust reversible watermarking schemes have been proposed in the literature. Most of these schemes are deployed in the spatial domain and are robust to JPEG compression and the addition of Gaussian noise. Some examples of this kind of scheme are the ones presented by [2,3]. Furthermore, there are schemes that use the frequency domain for the insertion of the watermark. As is well known, the frequency domain is usually more robust to attacks than the spatial domain. There are some examples of such schemes in the literature, such as those proposed in [4–6]. Typically, these schemes are robust to attacks such as clipping, noise addition, scaling, and histogram

modification. Furthermore, RWSs for medical images have been recently proposed under a hybrid approach such as in [7].

An interesting watermarking application is known as signal authentication, where a fragile watermark is used to identify and localize tampering in the attacked signals. A representative fragile watermarking scheme for medical images is reported in [8]. Due to the need to restore the attacked signals, beyond identifying the attack, self-recovery schemes have emerged. In [9], the idea was proposed of embedding an image into itself to restore the tampered regions; this is the first self-recovering scheme proposed in the literature. The steps typically followed by self-recovering schemes are: obtain a compressed version of the host signal and insert this version into the host signal itself. Upon detection, using a fragile watermark, of an attacked region, it is possible to restore it using the compressed version originally inserted into the host signal. It has been observed in the literature that there are two types of results delivered by self-recovery schemes: approximate recovery and perfect recovery. The former refers to the ability to recover the attacked signal to levels of high similarity with the original one, while perfect recovery refers to the fact that the recovered signal will be exactly the same as the original signal.

Self-recovery schemes for audio, image, or video signals have been proposed in the literature in recent years. In general, the strategies deal with modifying the insertion domain or improving the compression process required for watermark insertion. Some relevant proposals that insert the watermark in the spatial domain are [10–13]; similarly, the use of the frequency domain as the insertion space was reported in [14–16]. The use of the two domains, spatial and temporal, has also been explored by utilizing video signal as the host in [17,18]. Reference [19] proposed a scheme for video signals in the DCT domain, and it resists MPEG attacks. However, self-recovering schemes are limited in their additional payload to the compressed version of the host signal that is used as a watermark. If additional information is inserted, the distortion caused can be prohibitive for real-world applications. In [20], a framework was first introduced; the idea behind this framework is the combination of a fragile reversible watermarking scheme that allows the insertion of a useful payload to the image and a self-recovery scheme to insert information that can aid in the recovery of tampered regions of the images. By combining these schemes, the framework achieves adequate transparency for the insertion of both control information and useful payload. At the same time, it maintains the capacity of restoring the areas tampered with content replacement. This approach is different from the framework proposed by Coltuc in [21], where the first stage is a robust watermarking algorithm followed by a reversible watermarking algorithm. Coltuc's algorithm has inspired other works recently, such as [22]; however, the perceptual distortion induced by that framework has been studied as a whole; thus, the contribution of each stage is little known.

In this manuscript, an evaluation of the framework proposed in [20] is presented. The main goal is to observe the perceptual distortion due to the framework and how it is related to distortion due to both reversible and self-recovery stages. The main contribution of this study is to determine the perceptual transparency of the framework from the knowledge of each of the two algorithms selected for a given configuration. No more experimentation of the whole framework will be needed for each new algorithm used to know its perceptual transparency.

The rest of the document is organized in the following way. In Section 2, the framework proposed in [20] among several reversible watermarking and self-recovery schemes is described. Numerical results and discussions are given in Section 3. Finally, Section 4 concludes this paper.

2. Evaluation of the Robust Framework for Robust Reversible Image Watermarking

The framework proposed by [20] was taken as a general outline to be used with different combinations of fragile reversible and self-recovery schemes to determine which combination obtains better robustness results and transparency. The idea behind this framework is detailed below.

2.1. Description of the Framework

In [20], a framework is proposed to construct a robust reversible watermarking scheme with signal restoration. Figure 1 shows the framework, which consists of two processes: encoding and decoding. Each process consists of a fragile stage and a self-recovery stage. Figure 2 shows a general fragile reversible watermarking scheme. Secret message M is hidden in a host signal X, resulting in a watermarked signal Y transmitted through a noise-free channel. The extraction process requires Y and control data to reconstruct the host signal X, and it also recovers a message M'.

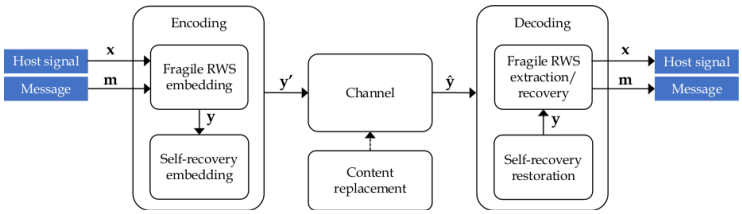

Figure 1. Framework for robust RWS with signal restoration.

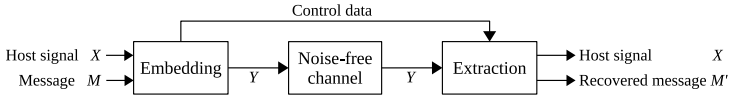

Figure 2. General fragile reversible watermarking scheme.

On the other hand, Figure 3 shows a general self-recovery scheme that can detect tampered regions and reconstruct them with the embedded control data.

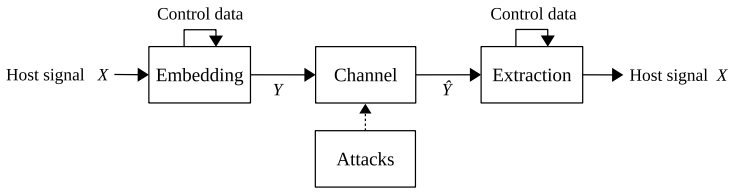

Figure 3. General self-recovery scheme.

In the encoding process, the fragile embedding stage inserts the secret message m into the host signal x, producing a watermarked signal y. Then, the self-recovery stage embeds control information, which allows the recovery of the signal after a content replacement attack. The result is the protected signal y'.

In the presence of a content replacement, attack signal y' becomes an attacked signal $\hat{y}$. The decoding process receives $\hat{y}$, then the restoration stage tries to revert the attack. The self-recovery restoration stage extracts the control information embedded in the encoding process and utilizes it along with the remaining non-attacked regions of the signal to restore the regions altered by the attack. If the attack is limited to a threshold, the watermarked signal y produced by the encoding process is obtained. From the watermarked signal y, the fragile RWS extraction and recovery stage can extract the secret message and recover the original samples from the host signal.

2.2. Selection of Schemes

The framework consists of two stages in each process that use the encoding and decoding algorithms of a fragile reversible watermarking and a self-recovery scheme. There are several fragile reversible watermarking schemes in the literature, but only three were selected to test the framework; the criteria used to choose them were their transparency and their embedding capacity; schemes with transparency over 40 dB and a payload over one bpp were selected. On the other hand, self-recovery schemes are not as typical in the literature; however, self-recovery schemes with perfect restoration capabilities were the

ones used for this methodology since that is a requirement for the framework to properly extract the watermarks in the fragile stage.

2.2.1. Fragile Reversible Watermarking Schemes

The scheme by [23] uses a histogram shifting technique, using the high correlation of neighboring pixels in each block of the image; the authors use a threshold K to insert the watermark bits; based on the difference values calculated for each block, the embedding strategy is divided into two categories: positives and negatives. The embedding strategy first divides the image in blocks of size 2×2, 4×4 or 8×8 pixels. Then, the difference values are calculated using the following equation:

$$\alpha = \frac{1}{n} \sum_{i=1}^{n} (a_i - b_i), \tag{1}$$

where n is the number of pairs of pixels, a_i are all the pixels marked with (+), and b_i are all the pixels marked with $(-)$. After the difference values have been calculated, a threshold $K+$ is selected using Equations (2) and (3) and a threshold $K-$ using Equations (4) and (5):

$$K_{ph} = \left\lceil \left(\frac{\alpha_{\text{maxP}} - \alpha_{\text{zero}} + 1}{\text{Partition level}} \right) \right\rceil, \tag{2}$$

$$K+ = K_{ph} + \text{mod}(K_{ph}, 2), \tag{3}$$

$$K_{Nh} = \left\lceil \left(\frac{\alpha_{\text{zero}} - \alpha_{\text{minN}} + 1}{\text{Partition level}} \right) \right\rceil, \tag{4}$$

$$K- = -1 \times (K_{Nh} + \text{mod}(K_{Nh}, 2)), \tag{5}$$

where α_{maxP} is the maximum difference value of the positive part, α_{zero} is 0 for the center of the coordinate, Partition level is the number of the partition, α_{minN} is the minimum difference value of the negative part, and $\lceil . \rceil$ is the ceiling function. The watermark bits are inserted based on the values of α, according to the following conditions:

- If $\alpha \geq 0$, there can be three cases:
  - **Case 1** $0 \leq \alpha \leq k$; if 1 is inserted, the histogram from that block is shifted a distance k to the right; if a 0 is inserted, the block is left intact.
  - **Case 2** $k \leq \alpha \leq 2k$; if a 1 is inserted, the histogram from that block is shifted a distance $2k$ to the right; if a 0 is inserted, the block is shifted a distance k to the right.
  - **Case 3** $2k \leq \alpha \leq 3k$; if a 1 is inserted, the histogram from that block is shifted a distance $3k$ to the right; if a 0 is inserted, the block is shifted a distance $2k$ to the right.
- If $\alpha < 0$, there can be three cases:
  - **Case 1** $-k \leq \alpha \leq 0$; if 1 is inserted, the histogram from that block is shifted a distance k to the left; if a 0 is inserted, the block is left intact.
  - **Case 2** $-2k \leq \alpha \leq -k$; if a 1 is inserted, the histogram from that block is shifted a distance $2k$ to the left; if a 0 is inserted, the block is shifted a distance k to the left.
  - **Case 3** $-3k \leq \alpha \leq -2k$; if a 1 is inserted, the histogram from that block is shifted a distance $3k$ to the left; if a 0 is inserted, the block is shifted a distance $2k$ to the left.

To extract the watermark bits, the image is divided into blocks in the same way as for embedding. Then, the difference values are calculated using Equation (1). After the difference values are obtained, the threshold K is obtained in the following way:

$$K = K_{pH} + \text{mod}(K_{ph}, 2), \tag{6}$$

where α_{maxP} is the maximum difference value of the positive part, α_{zero} is 0 for the center of the coordinate, Partition level is the number of the partition, and $\lceil . \rceil$ is the ceiling function. Finally, the watermark bit is extracted using the threshold K and the original image is reconstructed.

The scheme by [24] uses a histogram shifting technique as well; it divides the intensity range into non-overlapped segments and seeks the peak pixel, i.e., the pixel with a more significant occurrence in the histogram of each segment. Bit insertion is performed only in the peak pixels, except for each segment's first peak pixel, which is a reference for extraction. These schemes use a location map where all the peak pixels are marked. The embedding process first divides the intensity levels of the image into non-overlapped segments of equal size. Then, it generates a histogram for each block and identifies the intensity value of the peak pixel in each segment. The pixels from the original image are read, and then, k bits from the secret message M are embedded using a pixel substitution strategy on the peak pixels. Afterward, a location map is created, where a 1 is inserted if the pixel is a peak pixel or 0 otherwise. For the extraction process, the intensity values of the image are divided into non-overlapped segments of equal size in the same way as in the embedding process. The watermarked image and the location map are read. The peak pixels of each reference segment are obtained; then the watermark bits are extracted, and the current pixel is replaced with the reference peak pixel.

The scheme proposed in [25] expands the image size using an interpolation method that uses trigonometric functions; watermark bits are inserted in the interpolated pixels; this algorithm uses a location map. The extraction process recovers the watermark and eliminates the pixels generated by interpolation during embedding; therefore, the original image can be recovered. The embedding process first divides the host image into blocks of 2×2 pixels. Each block is transformed into blocks of 3×3 pixels through interpolation, using the following equations:

$$
\begin{aligned}
B_i(1,1) &= B_o(1,1) \\
B_i(3,3) &= B_o(2,2) \\
B_i(2,2) &= \sqrt{\frac{(B_i(1,2)^2) + (B_i(3,2))^2}{2}} \\
B_i(1,3) &= B_o(1,2) \\
B_i(1,2) &= \sqrt{\frac{(B_i(1,1))^2 + (B_i(1,3))^2}{2}} \\
B_i(2,3) &= \sqrt{\frac{(B_i(1,3))^2 + (B_i(3,3))^2}{2}} \\
B_i(3,1) &= B_o(2,1) \\
B_i(2,1) &= \sqrt{\frac{(B_i(1,1))^2 + (B_i(3,1))^2}{2}} \\
B_i(2,3) &= \sqrt{\frac{(B_i(3,1))^2 + (B_i(3,3))^2}{2}},
\end{aligned}
$$

where B_o is the original image block and B_i is the interpolated block. The new values are modified using the following trigonometric functions:

$$B_f(1,2) = B_i(1,2)\cos(\frac{B_i(1,1)+B_i(1,3)}{2})$$
$$B_f(2,2) = B_i(2,2)\cos(\frac{B_i(1,2)+B_i(3,2)}{2})$$
$$B_f(2,1) = B_i(2,1)\cos(\frac{B_i(1,1)+B_i(3,1)}{2})$$
$$B_f(2,3) = B_i(2,3)\cos(\frac{B_i(1,3)+B_i(3,3)}{2})$$
$$B_f(3,2) = B_i(3,2)\cos(\frac{B_i(3,1)+B_i(3,3)}{2}),$$

where B_f is the block modified by the trigonometric functions. Obtain 5 bits from the watermark M, and insert them in the interpolated pixels. The extraction process divides the image into blocks of size 3×3 pixels, called C_i. Then, calculate a new block using trigonometric functions, called C_i'. The watermark bits are extracted by subtracting the two blocks $C_i - C_i'$, and finally, the original image is obtained by eliminating the interpolated pixels.

2.2.2. Self-Recovery Schemes

The scheme by [26] is a self-recovery one, where reference bits and check bits are inserted into the images themselves to restore the original samples and identify the tampered regions. The embedding process uses a difference expansion (DE) strategy to insert both references and check bits in all the blocks of the host image; if the watermark inserted in one region of the image is altered, the rest of the watermark bits are not altered. The image can be recovered using the reference bits extracted from non-tampered regions. The embedding process divides the image into blocks of size 8×8, and then, each block is divided into 16 sub-blocks, where changeable and unchangeable pixels are assigned. For each changeable pixel, the inequality $g_m(i,j) \geq g_u$ is verified:

$$g_u + [g_m(i,j) - g_u] \cdot 2 + 1 \leq 255. \tag{7}$$

If $g_m(i,j) < g_u$, then:

$$g_u + [g_m(i,j) - g_u] \cdot 2 \leq 255, \tag{8}$$

where g_m represents the grayscale values of the changeable pixels for each block and g_u are the pixels designated as unchangeable of each block. When Equations (7) and (8) are fulfilled, the pixel $g_m(i,j)$ is deemed as unusable, otherwise as usable. Afterwards, the reference bits are generated, taking as the base the pixels from the original image. Then, the check bits are calculated in order to identify the tampered regions, using a 64 bit hash function. The embedding strategy used is DE using the following equation:

$$\tilde{g}_m = g_u + [g_m(i,j) - g_u] \cdot 2 + w, \tag{9}$$

where w is constructed with the reference and check bits to be embedded. For the extraction process, the image is divided into $N/64$ blocks and $N/4$ sub-blocks, in the same way as in the embedding process, where N is the total number of pixels in the image. The check and reference bits are extracted to identify the tampered and reserved blocks. Finally, the original grayscale values from all the pixels in the blocks identified as tampered are restored.

The scheme by [27] proposes a secure block mechanism, resilient to content replacement attacks. To locate the tampered blocks, it uses the unaltered pixels and the reference bits to estimate the original five MSB of the altered pixels, using an iterative and exhaustive restoration mechanism. The embedding process divides the image in a pseudo-random manner using a secret key k_1 in a subset of m pixels. The reference bits br_i are calculated, using Equations (10) and (11), and the br_i bits are embedded into the third LSB of the pixels in each subset.

$$br_i = H(\hat{x}_{i,1}, \cdots, \hat{x}_{i,m}) \tag{10}$$

$$\hat{x}_{i,j} = \lfloor \frac{x_{i,j}}{8} \rfloor \mod 16, j = 1, \cdots, m, \tag{11}$$

where $H(.)$ is a hash function, $\lfloor . \rfloor$ is the floor function, and $\hat{x} \in [0, 15]$ is the decimal value of the bits $b_4, \cdots, b_7$ of each pixel $x_{i,j}$. A second set of reference bits br_2 is calculated using Equations (10) and (12). The image is divided into subsets of m pixels, each using a second secret key k_2. A hash function with the five MSBs of each pixels for every subset is calculated, then br_2 is embedded into the pixels of each subset.

$$\hat{x}_{i,j} = \lfloor \frac{x_{i,j}}{8} \rfloor, j = 1, \cdots, m. \tag{12}$$

Then, the image is divided into non-overlapped blocks of size 8×8. Afterward, the authentication code ca for each block is calculated using Equation (13), which is inserted in the LSB of every pixel in each block.

$$ca_i = I \| n_1 \| \| n_2 \| p_i, \tag{13}$$

where I is an exclusive index associated with each image, p_i is the index of the block, and $\|n_1\|$ and $\|n_2\|$ is the bit concatenation of the image size. The extraction process divides the image in non-overlapped blocks of 8×8, as in the embedding process. Then, the authentication code is extracted from the LSB of each block, and the correct authentication code is assigned; the correct authentication code is obtained by majority voting. Afterward, the image is divided pseudo-randomly into subsets of m pixels, using a secret key k_1. The reference bits br_1 are restored from bit b_3, then the four MSB of every altered pixel of every block are calculated, by calculating the test codes $cp_i = H(\hat{y}_{i,1}, \cdots, \hat{y}_{i,m})$, where $\hat{y}_{i,j} = \lfloor \frac{x_{i,j}}{8} \rfloor$ mod 16. If $y_{i,j}$ is an altered pixel, then all the possible values of the four bits are exhaustively assigned to $\hat{y}_{i,j}$. The test codes and the reference bits are compared to identify the four MSB matches. Then, the four MSB are extended, adding to the beginning a bit 1 and a bit 0, generating two new values of five bits each, called restoration candidates. The image is divided into subsets of m pixels each, but now using the second secret key k_2, and only one restoration candidate is associated with the altered pixel. Finally, the original image is restored.

2.3. Perceptual Distortion Evaluation

The peak-signal-to-noise ratio (PSNR) measures the similarity of two signals, a reference signal, and a processed version of it, and this ratio is given in decibels [28,29] as some signals hold a very wide dynamic range. The PSNR is a metric complementary to distortion. The higher the PSNR, the lower the distortion is.

In image processing, a typically PSNR is measured in an 8 bit grayscale version. For an image f and its processed version g, both of size $M \times N$, the PSNR between f and g is computed by:

$$\text{PSNR(f, g)} = 10 \cdot \log_{10} \left(\frac{255^2}{MSE(f, g)} \right) \tag{14}$$

$$MSE(f, g) = \frac{1}{(M \cdot N)} \sum_{i=1}^{M} \sum_{j=1}^{N} (f_{ij} - g_{ij})^2 \tag{15}$$

As for the mean-squared error (MSE), the difference between the pixels f_{ij} and g_{ij} is considered an error that generates image quality loss. The lower the MSE, the higher the PSNR; therefore, the higher the PSNR(f, g) values, the higher the image quality is. For digital images, sometimes, a PSNR > 35 dB is considered as good quality [30].

The Watson metric, which quantifies the distortion of an image based on just noticeable differences [31], is another tool to evaluate the distortion of a processed image. This metric measures the errors for each DCT coefficient in each block by its corresponding sensitivity threshold. That threshold considers contrast sensitivity, luminance masking, and contrast masking.

In the Watson model, for each ij DCT coefficient, the relation between the luminance and frequency is considered, as follows:

$$t_{ijk} = t_{ij} \left(\frac{c_{00k}}{\bar{c}_{00}} \right)^{a_t} \tag{16}$$

where t_{ij} is the threshold for the smallest frequency coefficient that yields a visible signal, c_{00k} is the DC coefficient of block k, $\bar{c}_{00}$ is the DC coefficient corresponding to the mean luminance of the display (1024 for an 8 bit image), and a_t determines the degree of masking (set to 0.65) [32].

When the visibility of a pattern is reduced by the presence of another pattern in the image, this phenomenon is known as texture masking. Watson extends the results of luminance and frequency masking to include texture masking as follows:

$$m_{ijk} = \max[t_{ijk}, \mid c_{ijk} \mid^{w_{ij}} t_{ijk}^{1-w_{ij}}] \tag{17}$$

where m_{ijk} is the masked threshold and w_{ij} determines the degree of texture masking. Typically, $w_{00} = 0$ and $w_{ij} = 0.7$ for all other coefficients. The perceptual error in each frequency of each block is given by

$$d_{ijk} = \frac{e_{ijk}}{m_{ijk}} \tag{18}$$

where e_{ijk} is the quantization error.

To associate the errors in the model, the Minkowski metric is used as follows:

$$p_{ij} = \left(\sum_{k} \mid d_{ijk} \mid^{\beta_s} \right)^{\frac{1}{\beta_s}} \tag{19}$$

where different values of the exponent β_s implement different types or degrees of pooling. In practice, $\beta_s = 100$ is commonly used.

A typical threshold for the Watson metric is 0.4, since measures below this point guarantee visual imperceptibility [33].

The following section presents the experimental results obtained after implementing the framework using the fragile reversible watermarking schemes, and self-recovery schemes were selected.

3. Results and Discussions

In this section, framework evaluation results are presented. The evaluation is divided into three stages: first, the fragile reversible watermarking schemes are evaluated in terms of perceptual distortion versus payload; then, the maximum robustness of self-recovery schemes is measured; finally, the framework is evaluated for each possible configuration of reversible watermarking and self-recovery stages.

The six algorithms used in this study were implemented in the MatLab R2014 language, using a computer with an Intel Core i7-1255U processor, 16 GB RAM, 128 GB solid-state disk, and an Nvidia GeForce MX550 2 GB GDDR6 graphics card. Five algorithms were implemented in their sequential form following their description reported in the literature. The algorithm proposed by Bravo-Solorio et al. [27] was optimized using the MatLab parallel computing toolbox due to its very high time complexity. For each experiment, 2000 gray-level images from the BOWS 2 [34] data set were used; each image is 512 × 512 pixels in size.

3.1. Fragile Reversible Watermarking Schemes

Each fragile reversible watermarking scheme was applied to the data set using watermarks of 1000 bits to 10,000 bits with 1000 bits steps. The peak-signal-to-noise ratio (PSNR) and Watson metrics evaluated the algorithm's distortion for different watermark lengths.

Figure 4 shows the performance of the fragile reversible watermarking schemes using the PNSR metric as an evaluation tool. A comparison between fragile reversible watermarking techniques using the Watson metric as an evaluation tool is shown in Figure 5.

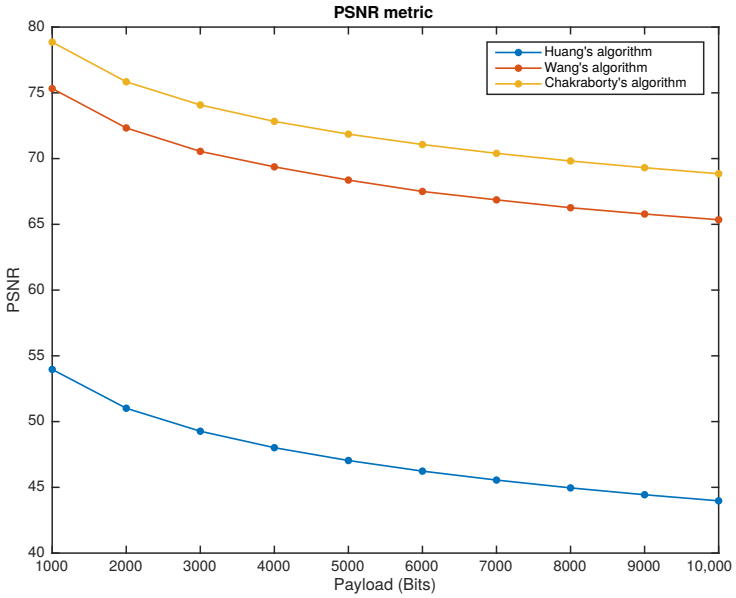

Figure 4. Fragile reversible watermarking schemes performance versus payload under PSNR metric.

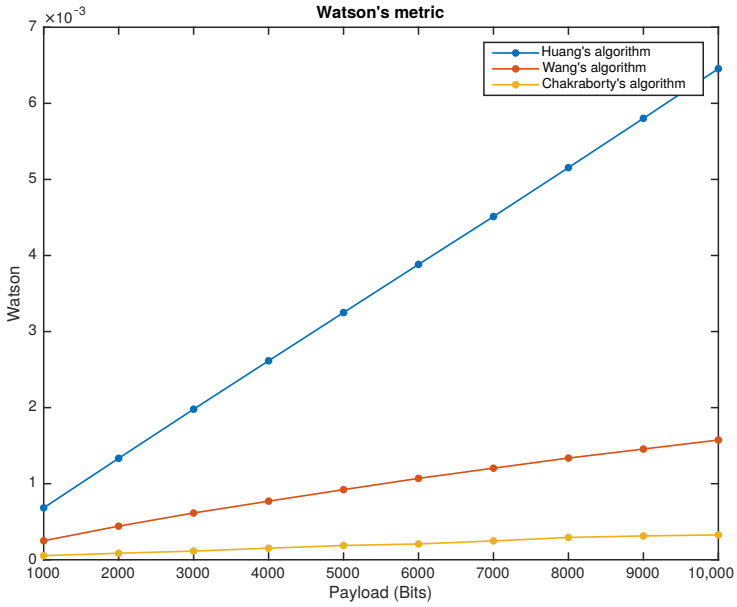

Figure 5. Fragile reversible watermarking schemes performance versus payload under the Watson metric.

From Figures 4 and 5, it is possible to observe that, although performance decreases when a higher payload is hidden, all the selected fragile reversible watermarking schemes hold high transparency for high payloads. The PSNR and Watson values for the highest payload applied are better than the minimum suggested in the literature.

3.2. Self-Recovery Schemes

Each self-recovery scheme was tested using the images from the data set to evaluate the perceptual distortion and robustness of each scheme; the perceptual distortion is measured using the PSNR and Watson metrics. To determine the robustness of the schemes, content

replacement attacks were applied to the watermarked images using different percentages to corroborate their robustness to a maximum percentage of attack, which is the same maximum as the ones reported by the authors. A content replacement attack takes a region of the image to attack and replaces it with another image with the same dimensions as the region being replaced. Although other common attacks exist, we focus on content replacement in this work since it is the attack that the two self-recovery schemes resist. Figure 6 shows an example of content replacement attacks using different percentages of substitution. Table 1 shows the mean value for the PSNR, and Watson's values were obtained from measuring the perceptual quality of the images watermarked with the self-recovery schemes; the maximum percentage of attack supported by each scheme is included in the last column. The maximum percentage of attack refers to the highest level of substitution where each scheme is capable of producing an image with perfect restoration, i.e., the scheme produces, as a result, an image containing the exact pixel values as the host image. Furthermore, the time complexity for processing one image is shown in Table 1.

Figure 6. Content replacement attack: (**a**) 3.2%, (**b**) 10%, (**c**) 15%, and (**d**) 20%.

Table 1. Self-recovery schemes' performance in perceptual, robustness, and time complexity terms.

Algorithm	PSNR dB	Watson	Max Attack	Time Complexity
Zhang and Wang [26]	29.57	0.135	3.2%	10 msec
Bravo-Solorio et al. [27]	37.90	0.067	20.0%	49.7 min

From Table 1, it is possible to observe that self-recovery strategies generate high distortion levels on the watermarked images; this is caused because a big amount of information from the image itself is embedded into the carrier image to guarantee self-recovery capacities. In terms of robustness, it can be observed that the scheme by Zhang and Wang [26] achieves robustness for a maximum attack of 3.2%, while the scheme by Bravo-Solorio et al. [27] achieves robustness for a maximum attack of 20%. It is worth mentioning that the scheme [27] can achieve robustness to higher percentages of attack at higher computational time cost; in this work, an optimized implementation of the scheme is used, which needs about 50 min to reconstruct an image tampered at 20%. Then, because of our limited computational resources, it was prohibitive to test [27] for percentages of tampering higher than 20%.

The workflow for the restoration of the self-recovery scheme is as follows: during encoding, the self-recovery stage receives an image watermarked with the fragile reversible scheme and produces a second watermarked image that contains information for self-recovery, then that second watermarked image is attacked with a percentage of content replacement. In the decoding phase, the self-recovery stage receives the attacked image and extracts self-recovery information that allows perfect restoration, given that the attacked region is smaller than the maximum percentage supported; after perfect restoration, the image is passed to the fragile reversible stage to extract the secret message and recover the host image. Figure 7 depicts an example of perfect restoration after a 10% attack.

(**a**) (**b**) (**c**)

Figure 7. Example of image perfect restoration after a 10% attack. (**a**) Watermarked image, (**b**) attacked image, and (**c**) restored image.

3.3. Framework

As described in Section 2, the framework consists of two main stages, a fragile reversible watermarking stage and a self-recovery stage. This section evaluates the framework through each possible combination of fragile reversible watermarking and self-recovery schemes. Table 2 shows the perceptual distortion performance for each scheme in detail.

Table 2. Perceptual distortion performance for fragile reversible watermarking (F1, F2, and F3) and self-recovery schemes (S1 and S2).

ID	Work	PSNR (dB)				Watson			
		μ	σ	min	max	μ	σ	min	max
F1	[23]	47.03	±2.84	39.68	56.40	0.1433	±0.0032	0.0007	0.0905
F2	[24]	65.35	±0.03	65.21	65.48	0.0016	±0.0015	0.0006	0.0271
F3	[25]	68.84	±0.04	68.72	69.00	0.0003	±0.0002	0.0002	0.0027
S1	[26]	29.57	±4.10	20.17	46.98	0.1350	±0.0675	0.0169	0.4146
S2	[27]	37.90	±0.12	36.78	38.71	0.0679	±0.0316	0.0422	0.7103

The framework was evaluated using six different configurations by combining the three selected fragile reversible watermarking and the two self-recovery schemes. Table 3 shows the results in terms of visual transparency for each configuration; the mean, standard deviation, minimum, and maximum values for PNSR and Watson metrics are presented. The robustness of the framework depends only on the self-recovery stage; therefore, no experimentation was carried out to observe the robustness of each configuration.

Table 3. Perceptual distortion performance for framework configuration. The best configuration in terms of average PSNR and Watson values is highlighted in bold.

Configurations	PSNR (dB)				Watson				Payload (bits)
	μ	σ	min	max	μ	σ	min	max	
F1-S1	29.02	±4.51	20.25	40.34	0.1433	±0.0830	0.0271	0.3953	5k
F1-S2	37.34	±0.33	36.34	37.90	0.0688	±0.0497	0.0497	0.4546	5k
F2-S1	29.56	±4.09	20.17	46.76	0.1355	±0.0675	0.0174	0.4145	10k
F2-S2	**37.90**	±0.13	36.53	38.74	**0.0678**	±0.0313	0.0419	0.7022	10k
F3-S1	32.39	±3.87	23.10	46.13	0.0985	±0.0459	0.0202	0.2977	10k
F3-S2	37.89	±0.12	36.90	38.90	0.0678	±0.0199	0.0420	0.3357	10k

As it is possible to observe from Table 2, fragile reversible watermarking schemes hold low perceptual distortion; thus, self-recovery schemes produce high perceptual distortion levels. This behavior is due to self-recovery schemes that need to embed considerable image information to have reconstruction capacities. Table 3 shows that the perceptual distortion

due to the framework is very close to the distortion due to the self-recovery scheme used in the configuration; in this table, the best configuration in terms of average PSNR and Watson values is highlighted in bold, which is the combination of the fragile reversible scheme [24] and the self-recovery [27]. However, additional experimentation was carried out to corroborate the framework's performance in terms of perceptual distortion. A fragile reversible watermarking scheme with high perceptual distortion was incorporated into the framework, and the perceptual distortion of the framework was measured. The fragile reversible watermarking scheme used is proposed by Coltuc and Tudoroiu [35]. Table 4 shows the performance of [35] in perceptual distortion terms when applied to the image data set. In Table 4, n is a payload controller; thus, $\log_2 n$ is the payload achieved per pixel. In [35], $n \in [2, 16]$; in this study, the minimum and maximum values are considered.

Table 4. Perceptual distortion performance of [35].

ID	Expansion n	PSNR (dB)				Watson			
		μ	σ	min	max	μ	σ	min	max
C1	2	26.51	±6.76	7.93	49.92	0.1673	±0.1657	0.0118	2.8645
C2	12	15.24	±3.37	6.25	27.83	0.7331	±0.3117	0.1402	4.3541

As it can be observed in Table 4, the scheme proposed in [35] holds high perceptual distortion; in fact, distortion due to this fragile reversible watermarking scheme is more elevated than self-recovery schemes.

Table 5 shows the performance of the framework for each possible configuration using the scheme in [35].

Table 5. Perceptual distortion performance for framework configuration using [35] as the fragile reversible watermarking scheme.

Configurations	PSNR (dB)				Watson			
	μ	σ	min	max	μ	σ	min	max
C1-S1	21.74	±5.11	7.89	40.70	0.3164	±0.2309	0.0351	3.3922
C1-S2	25.78	±5.88	7.92	37.65	0.1983	±0.1686	0.0539	2.9249
C2-S1	12.40	±2.59	6.15	22.66	1.0504	±0.3784	0.2753	4.9221
C2-S2	15.21	±3.34	6.55	27.44	0.7391	±0.3116	0.1518	4.3353

From Table 5, it is possible to observe that the framework performance is determined by the fragile reversible watermarking scheme instead of the self-recovery scheme, as in Table 3. Then, it is possible to claim that the expected perceptual distortion of the framework depends on the performance at each stage, regardless of its nature. Thus, the framework distortion could be modeled as follows:

$$D_{frame} \approx \max\{D_{fw}, D_{sr}\} \tag{20}$$

where D_{frame} is the distortion due to the framework, D_{fw} is due to the fragile reversible watermarking scheme, and D_{sr} is due to the self-recovery scheme. Alternatively, the performance of the framework can be expressed in terms of the PSNR as follows:

$$PSNR_{frame} \approx \min\{PSNR_{fw}, PSNR_{sr}\} \tag{21}$$

where $PSNR_{frame}$ is the PSNR value for the framework, $PSNR_{fw}$ is the PSNR value for the fragile reversible watermarking scheme, and $PSNR_{sr}$ is the PSNR value for the self-recovery scheme.

4. Conclusions

In this study, the framework proposed in [20] was evaluated using ten different configurations to determine its perceptual distortion and robustness performance behavior.

Using the appropriate reversible and self-recovery schemes within the framework makes it possible to achieve high robustness to content replacement attacks. At the same time, perceptual distortion is kept at practical levels. Moreover, an expression that approximated the framework performance in terms of perceptual distortion was proposed. The findings of this study will help the fast evaluation of reversible watermarking and self-recovery schemes to provide reversible watermarking robust to content replacement attack as the approximate perceptual distortion can be estimated by (20) and robustness is only due to the self-recovery stage.

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

Funding: This work was partially supported by the project 2017-01-7092 from CONACyT, Mexico.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: In this study, 2000 gray-level images from the BOWS 2 [34] data set were used. The BOWS 2 data set is freely available in http://bows2.ec-lille.fr (last access: 13 June 2022).

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Menendez-Ortiz, A.; Feregrino-Uribe, C.; Hasimoto-Beltran, R.; Garcia-Hernandez, J.J. A Survey on Reversible Watermarking for Multimedia Content: A Robustness Overview. *IEEE Access* **2019**, *7*, 132662–132681. [CrossRef]
2. An, L.; Gao, X.; Yuan, Y.; Tao, D.; Deng, C.; Ji, F. Content-adaptive reliable robust lossless data embedding. *Neurocomputing* **2012**, *79*, 1–11. [CrossRef]
3. An, L.; Gao, X.; Yuan, Y.; Tao, D. Robust lossless data hiding using clustering and statistical quantity histogram. *Neurocomputing* **2012**, *77*, 1–11. [CrossRef]
4. Yang, C.Y.; Lin, C.H.; Hu, W.C. Reversible Watermarking by Coefficient Adjustment Method. In Proceedings of the 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), Darmstadt, Germany, 15–17 October 2010; pp. 39–42. [CrossRef]
5. Tsai, H.H.; Tseng, H.C.; Lai, Y.S. Robust lossless image watermarking based on α-trimmed mean algorithm and support vector machine. *J. Syst. Softw.* **2010**, *83*, 1015–1028. Software Architecture and Mobility. [CrossRef]
6. An, L.; Gao, X.; Li, X.; Tao, D.; Deng, C.; Li, J. Robust reversible watermarking via clustering and enhanced pixel-wise masking. *IEEE Trans. Image Process.* **2012**, *21*, 3598–3611. [CrossRef] [PubMed]
7. Dai, Z.; Lian, C.; He, Z.; Jiang, H.; Wang, Y. A Novel Hybrid Reversible-Zero Watermarking Scheme to Protect Medical Image. *IEEE Access* **2022**, *10*, 58005–58016. [CrossRef]
8. Soualmi, A.; Alti, A.; Laouamer, L.; Benyoucef, M. A Blind Fragile Based Medical Image Authentication Using Schur Decomposition. In *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019)*; Hassanien, A.E., Azar, A.T., Gaber, T., Bhatnagar, R., Tolba, M.F., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 623–632.
9. Fridrich, J.; Goljan, M. Protection of digital images using self embedding. In *Symposium on Content Security and Data Hiding in Digital Media*; New Jersey Institute of Technology: Newark, NJ, USA, 1999.
10. He, H.; Zhang, J.; Chen, F. A self-recovery fragile watermarking scheme for image authentication with superior localization. *Sci. China Ser. F Inf. Sci.* **2008**, *51*, 1487–1507. [CrossRef]
11. He, H.J.; Zhang, J.S.; Tai, H.M. Self-recovery Fragile Watermarking Using Block-Neighborhood Tampering Characterization. In *Information Hiding*; Katzenbeisser, S., Sadeghi, A.R., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5806, pp. 132–145. [CrossRef]
12. Bravo-Solorio, S.; Li, C.T.; Nandi, A. Watermarking with low embedding distortion and self-propagating restoration capabilities. In Proceedings of the 19th IEEE International Conference on Image Processing (ICIP), Orlando, FL, USA, 30 September–3 October 2012 ; pp. 2197–2200. [CrossRef]
13. Aminuddin, A.; Ernawan, F. AuSR1: Authentication and self-recovery using a new image inpainting technique with LSB shifting in fragile image watermarking. *J. King Saud Univ.-Comput. Inf. Sci.* 2022, *in press*. [CrossRef]

14. Wu, H.C.; Chang, C.C. Detection and restoration of tampered JPEG compressed images. *J. Syst. Softw.* **2002**, *64*, 151–161. [CrossRef]
15. Zhang, X.; Qian, Z.; Ren, Y.; Feng, G. Watermarking With Flexible Self-Recovery Quality Based on Compressive Sensing and Compositive Reconstruction. *IEEE Trans. Inf. Forensics Secur.* **2011**, *6*, 1223–1232. [CrossRef]
16. Li, C.; Wang, Y.; Ma, B.; Zhang, Z. A novel self-recovery fragile watermarking scheme based on dual-redundant-ring structure. *Comput. Electr. Eng.* **2011**, *37*, 927–940. [CrossRef]
17. Hassan, A.M.; Al-Hamadi, A.; Hasan, Y.M.Y.; Wahab, M.A.A.; Michaelis, B. Secure Block-Based Video Authentication with Localization and Self-Recovery. *World Acad. Sci. Eng. Technol.* **2009**, *2009*, 69–74.
18. Shi, Y.; Qi, M.; Lu, Y.; Kong, J.; Li, D. Object based self-embedding watermarking for video authentication. In Proceedings of the International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE), Changchun, China, 16–18 December 2011; pp. 519–522. [CrossRef]
19. Mobasseri, B. A spatial digital video watermark that survives MPEG. In Proceedings of the International Conference on Information Technology: Coding and Computing, Las Vegas, NV, USA, 27–29 March 2000 ; pp. 68–73. [CrossRef]
20. Menendez-Ortiz, A.; Feregrino-Uribe, C.; Garcia-Hernandez, J.J. Reversible image watermarking scheme with perfect watermark and host restoration after a content replacement attack. In Proceedings of the The 2014 International Conference on Security and Management (SAM'14), Las Vegas, NV, USA, 7–9 April 2014; Volume 13, pp. 385–391.
21. Coltuc, D. Towards distortion-free robust image authentication. *J. Physics: Conf. Ser.* **2007**, *77*, 012005. [CrossRef]
22. Hu, R.; Xiang, S. Cover-Lossless Robust Image Watermarking Against Geometric Deformations. *IEEE Trans. Image Process.* **2021**, *30*, 318–331. [CrossRef] [PubMed]
23. Huang, L.C.; Tseng, L.Y.; Hwang, M.S. A Reversible Data Hiding Method by Histogram Shifting in High Quality Medical Images. *J. Syst. Softw.* **2013**, *86*, 716–727. [CrossRef]
24. Wang, Z.H.; Lee, C.F.; Chang, C.Y. Histogram-Shifting-Imitated Reversible Data Hiding. *J. Syst. Softw.* **2013**, *86*, 315–323. [CrossRef]
25. Chakraborty, S.; Maji, P.; Pal, A.; Biswas, D.; Dey, N. Reversible Color Image Watermarking Using Trigonometric Functions. In Proceedings of the International Conference on Electronic Systems, Signal Processing and Computing Technologies (ICESC), Nagpur, India, 9–11 January 2014; pp. 105–110.
26. Zhang, X.; Wang, S. Fragile Watermarking With Error-Free Restoration Capability. *IEEE Trans. Multimed.* **2008**, *10*, 1490–1499. [CrossRef]
27. Bravo-Solorio, S.; Li, C.T.; Nandi, A. Watermarking method with exact self-propagating restoration capabilities. In Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), Costa Adeje, Spain, 2–5 December 2012; pp. 217–222. [CrossRef]
28. Hore, A.; Ziou, D. Image Quality Metrics: PSNR vs. SSIM. In Proceedings of the 20th International Conference on Pattern Recognition (ICPR), Istanbul, Turkey, 23–26 August 2010; pp. 2366–2369.
29. Ismail Avcıbas, B.S. *Statistical Analysis of Image Quality Measures*; Technical Report; Department of Electrical and Electronic Engineering, Bogaziçi University: İstanbul, Turkey, 1999.
30. Garcia-Hernandez, J.J.; Gomez-Flores, W.; Rubio-Loyola, J. Analysis of the impact of digital watermarking on computer-aided diagnosis in medical imaging. *Comput. Biol. Med.* **2016**, *68*, 37–48. [CrossRef] [PubMed]
31. Watson, A.B. DCT quantization matrices visually optimized for individual images. In *Human Vision, Visual Processing and Digital Display IV*; SPIE: Bellingham, WA, USA 1993; Volume 1913, pp. 202–216.
32. Ahumada, A.J.; Peterson, H.A. Luminance-Model-Based DCT Quantization for Color Image Compression. In *Human Vision, Visual Processing and Digital Display III*; SPIE: Bellingham, WA, USA 1992; Volume 1666, pp. 365–374.
33. Rodriguez, T.F.; Cushman, D.A. Optimized Selection of Benchmark Test Parameters for Image Watermark Algorithms based on Taguchi Methods and Corresponding Influence on Design Decisions for Real-World. In *SPIE-IS&T Electronic Imaging*; SPIE: Bellingham, WA, USA, 2003; Volume 5020, pp. 215–228.
34. Laboratory, W.V. Break Our Watermarking Systems 2, 2007. Image Data Set. Available online: http://bows2.ec-lille.fr (accessed on 13 June 2022).
35. Coltuc, D.; Tudoroiu, A. Multibit versus Multilevel Embedding in High Capacity Difference Expansion Reversible Watermarking. In Proceedings of the 20th European Signal Processing Conference (EUSIPCO 2012), Bucharest, Romania, 27–31 August 2012; pp. 1791–1795.