FEDERAL UNIVERSITY OF PERNAMBUCO
CENTER FOR TECHNOLOGY AND GEOSCIENCES
DEPARTMENT OF ELECTRONICS AND SYSTEMS
GRADUATE PROGRAM IN ELECTRICAL ENGINEERING

ANDY MARTIN RAMOS

NEW FRAGILE IMAGE WATERMARKING SCHEMES USING CHAOTIC SEQUENCES

Recife

2021

ANDY MARTIN RAMOS

NEW FRAGILE IMAGE WATERMARKING SCHEMES USING CHAOTIC SEQUENCES

Dissertation presented to the Graduate Program in Electrical Engineering at the Federal University of Pernambuco as a partial requirement for obtaining a master's degree in Electrical Engineering.

Concentration Area: Communications.

Supervisor: Prof. Dr. Cecílio José Lins Pimentel

Co-supervisor: Prof. Dr. Daniel Pedro Bezerra Chaves

Recife

2021

Catalogação na fonte:
Bibliotecário Josias Machado, CRB-4 / 1690

R175n Ramos, Andy Martin.
New fragile image watermarking schemes using chaotic sequences. / Andy Martin Ramos. – 2021.
52 f.: il., figs., tabs., abrev. e sigl.

Orientador: Prof. Dr. Cecílio José Lins Pimentel.
Coorientador: Prof. Dr. Daniel Pedro Bezerra Chaves.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Elétrica, Recife, 2021.
Inclui referências.

1. Engenharia elétrica. 2. Marca d'agua frágil. 3. Mapas caóticos. 4. Códigos corretores de erro. 5. Transformada wavelet discret. 6. Detecção de modificações. I. Pimentel, Cecílio José Lins (orientador). II. Chaves, Daniel Pedro Bezerra (coorientador). III. Título.

UFPE

621.3 CDD (22. ed.) BCTG/2022-290

ANDY MARTIN RAMOS

NEW FRAGILE IMAGE WATERMARKING SCHEMES USING CHAOTIC SEQUENCES

Dissertation presented to the Graduate Program in Electrical Engineering at the Federal University of Pernambuco as a partial requirement for obtaining a master's degree in Electrical Engineering.

Concentration Area: Communications.

Approved in: 23 / 07 / 2021 .

Prof. Dr. Cecílio José Lins Pimentel (Supervisor:)
Federal University of Pernambuco

Dr. Carlos Eduardo Correia De Souza (External Examiner)
Federal University of Pernambuco

Prof.Dr. Daniel Pedro Bezerra Chaves (Internal Examiner)
Federal University of Pernambuco

Prof. Dr. Valdemar Cardoso Da Rocha Junior (Internal Examiner)
Federal University of Pernambuco

To God, for giving me strength and faith to overcome the idea of what seemed impossible to finish.
To my family for always being present despite the distance.
My advisors and friends at UFPE, because without them it would not have been possible to carry out this work.

ACKNOWLEDGEMENTS

First of all, I thank God for giving me the health and strength to finish this work. I would like to thank the professors Dr. Cecilio José Lins Pimentel and Dr. Daniel Pedro Bezerra Chaves for their guidance.

This work would not have been possible without their advice and commitment. I also thank my friend Dr. José Antonio Perez de Morales Artiles for his unconditional help throughout these two years of study. To my classmates Matheus Lobo and Sergio Silva for their support in times of difficulty.

To the DES lecturers with whom I formed the basis of my learning in different disciplines. To DES employees, especially Andréa Tenório, secretary of the Graduate Program in Electrical Engineering (PPGEE) for her competence.

Finally, the Coordination of the PPGEE and the Foundation for Scientific and Technological Support of the State of Pernambuco (FACEPE) for the financial support throughout the development of this dissertation.

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 purpose. This dissertation presents two new proposals 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 and from bits of the original image. In the first proposed algorithm, the watermarked bits are inserted in the frequency domain using the Discrete Wavelet Transform. The imperceptibility, detection, and recovery of this algorithm are tested for various attacks used in signal processing. In the second method, the watermarks bits are embedded using the least significant bit method. A comparison between the proposed algorithms shows that the former exhibits greater imperceptibility, while the latter exhibits better recover capability. The proposed algorithms are analyzed both for grayscale and color images. Comparison results reveal that the proposed technique performs better than some existing methods.

Keywords: Fragile watermarking; chaotic maps; error correcting codes; discrete wavelet transform; tamper detection.

RESUMO

Com o rápido desenvolvimento de ferramentas de processamento digital de sinais, o conteúdo de uma imagem pode ser facilmente manipulado ou adulterado de forma maliciosa. A marca d'água frágil tem sido amplamente usada para fins de autenticação de dados. Esta dissertação apresenta novas propostas de algoritmos de marcas d'água frágil em imagens para detecção de área manipulada e recuperação da imagem. Os bits com marca d'água são obtidos a partir dos bits de paridade de um código corretor de erro cuja mensagem é formada a partir de uma sequência caótica binária e de bits da imagem original. No primeiro algoritmo proposto, os bits com marca d'água são inseridos no domínio da frequência usando a Transformada Wavelet Discreta. A imperceptibilidade, detecção e recuperação alcançada por este algoritmo são testadas para vários ataques usados em processamento de sinais. No segundo método, os bits da marca d'água são incorporados usando o método do bit menos significativo. Uma comparação entre os algoritmos propostos mostra que o primeiro apresenta maior imperceptibilidade, enquanto o último apresenta melhor capacidade de recuperação. Os algoritmos propostos são analisados tanto para imagens em tons de cinza quanto imagens coloridas. Os resultados da comparação revelam que a técnica proposta tem um desempenho melhor do que alguns métodos existentes.

Palavras - chaves: Marca d'água frágil; mapas caóticos; códigos corretores de erro; transformada wavelet discret; detecção de modificações.

LIST OF FIGURES

Figure 1 – Two different orbits for $x_0 = 0.1$ and $x'_0 = 0.100001$. 20
Figure 2 – Wavelet decomposition scheme in two dimensions. 22
Figure 3 – (a) Original image, (b) first decomposition level, (c) second decomposition level, (d) third decomposition level. 23
Figure 4 – Block diagram of the proposed watermark embedding algorithm. 25
Figure 5 – Block diagram of the proposed watermark extraction algorithm. 27
Figure 6 – Block diagram of the proposed tamper detection algorithm. 27
Figure 7 – Block diagram of the proposed recovery algorithm. 28
Figure 8 – (a) Lena, (b) Airplane, (c) Boat, (d) Lake, (e) Baboon, (f) Pepper. 30
Figure 9 – 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%. 32
Figure 10 – Comparison of PSNR^r versus tampering rates for the Lena image. 32
Figure 11 – 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). 34

Figure 12 – 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). 35

Figure 13 – 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). 36

Figure 14 – 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). 37

Figure 15 – 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). 38
Figure 16 – Block diagram of the watermark embedding algorithm. 42
Figure 17 – Block diagram of the recovered image. 43
Figure 18 – Comparison of PSNR^r versus tampering rates for the Lena image. 44
Figure 19 – Different attacks on color 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 = 0.009 and FNR = 0), (e) recovered image (PSNR^r = 55.31 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 $= 0.013$ and FNR $= 0$), (j) recovered image (PSNR^r = 50.32 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 $= 0.005$ and FNR $= 0$), (o) recovered Tiffany image (PSNR^r = 56.12 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 $= 0.002$ and FNR $= 0$), (t) recovered image (PSNR^r = 55.15 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 $= 0.028$ and FNR $= 0.008$), (y) recovered image (PSNR^r = 35.92 dB). 47

Figure 20 – Different attacks on color images (a-e) tampering recovery for the CA1 attack (a) original Tree image, (b) watermarked Tree image (PSNR 46.32 dB), (c) tampered image (13%), (d) binary detection image (FPR =0.008 and FNR =0), (e) recovered image (PSNR^r = 54.28 dB), (f-j) tampering recovery for the CA_2 attack: (f) watermarked Female1 image (PSNR = 45.38 dB), (g) watermarked Female2 image (PSNR = 46.06 dB), (h) tampered image (6%), (i) binary detection image (FPR $= 0.004$ and FNR $= 0$), (j) recovered image (PSNR^r = 55.64 dB), (k-o) tampering recovery for normal tampering attack: (k) original Female3 image , (l) watermarked Female3 image (PSNR = 44.52 dB), (m) tampered image (7%), (n) binary detection image (FPR $= 0.005$ and FNR $= 0$), (o) recovered image (PSNR^r = 55.34 dB), (p-t) tampering recovery for the CAA attack: (p) original House image, (q) watermarked image (PSNR = 46.32 dB), (r) tampered image (20%), (s) binary detection image (FPR $= 0.005$ and FNR $= 0$), (t) recovered image (PSNR^r = 54.88 dB), (u-y) Salt and Pepper attack (u) original Seeds image, (v) watermarked Seeds image PSNR = dB, (w) tampered image (30%), (x) binary detection image (FPR = 0.037 and FNR = 0.011), (y) recovered image (PSNR^r = 36.15 dB). 48

LIST OF TABLES

Table 1 – Minimum and maximum PSNR and SIMM for several values of α for the 141 images from the USC-SIPI database. 31
Table 2 – Minimum and maximum FPR and FNR for several values of α for the 141 images with tampering rate 50%. 31
Table 3 – PSNR comparison for several original images. 31
Table 4 – PSNR^r versus tampered rate comparison for several original images. 33
Table 5 – PSNR^r achieved by the proposed algorithm and by some existing methods. . 34
Table 6 – PSNR comparison for several original images. 39
Table 7 – PSNR^r versus tampered rate comparison for several original images. 40
Table 8 – PSNR^r comparison for several attacks of proposed algorithms. 40
Table 9 – PSNR comparison for several original images. 43
Table 10 – FPR, FNR, and PSNR^r comparisons for several attacks. 44
Table 11 – PSNR comparison for several original color images. 45
Table 12 – PSNR^r versus tampered rate comparison for several color original images. . . 46
Table 13 – FPR, FNR, and PSNR^r comparisons for several attacks for color images. . . 46

CONTENTS

1	**INTRODUCTION**	**14**
1.1	LITERATURE REVIEW	15
1.2	OBJECTIVES AND CONTRIBUTIONS OF THE DISSERTATION	17
1.3	ORGANIZATION	18
2	**PRELIMINARIES**	**19**
2.1	CHAOTIC MAPS	19
2.2	ERROR-CORRECTING CODES	21
2.3	DISCRETE WAVELET TRANSFORM	21
3	**A FRAGILE IMAGE WATERMARKING SCHEME USING CHAOTIC SEQUENCES**	**24**
3.1	THE PROPOSED ALGORITHM	24
3.1.1	**Watermark Embedding**	**24**
3.1.2	**Watermark Extraction, Tamper Detection, and Image Recovery**	**26**
3.2	IMPERCEPTIBILITY, DETECTION AND RECOVERY METRICS	28
3.2.1	**Imperceptibility Metrics**	**28**
3.2.2	**Tampered Detection Metric**	**29**
3.2.3	**Watermark Image Attacks**	**29**
3.3	RESULTS	30
3.3.1	**A BCH Code (31,21,2)**	**39**
4	**A FRAGILE IMAGE WATERMARKING SCHEME IN THE LSB DOMAIN**	**41**
4.1	THE WATERMARKING ALGORITHM	41
4.1.1	**Watermark Embedding**	**41**
4.1.2	**Watermark Extraction, Tamper Detection, and Image Recovery**	**41**
4.2	RESULTS	43
4.3	PERFORMANCE OF THE PROPOSED ALGORITHMS FOR COLOR IMAGES	45
5	**CONCLUSIONS AND FUTURE WORK**	**49**
	BIBLIOGRAPHY	**50**

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 (NASKAR, 2014). Watermarking can serve a variety of purposes including copyright protection and data authentication. An image watermarking is the process of embedding binary information (called watermark bits) into an original image generating a watermarked image. In a self-embedding watermarking scheme, the watermark bits are generated from the original image. The extraction process is called blind when it does not require knowledge either of the original image or the watermark bits.

In general, image watermarking techniques can be categorized as robust, semi-fragile and fragile (RAKHMAWATI, 2019). Robust watermarks are designed to survive image processing operations, such as scaling, cropping, filtering, compression (SHIH., 2010; MOOSAZADEH, 2019; KO, 2020), 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) (RAKHMAWATI, 2019). 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.

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 (PENG, 2018; QIN, 2016; QIN C., 2017; SREENIVAS; KAMAKSHIPRASAD, 2017; TAI, 2018; ABDELHAKIM, 2019; MOLINA J., 2020; LEE C.F., 2019). 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 watermark bits involves, in some cases, frequency-domain transforms, such as the discrete cosine transform (DCT) (ABDELHAKIM, 2019; SARRESHTEDARI, 2018), and the Discrete Wavelet Transform (DWT) (TAI, 2018).

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 (NASKAR, 2014). Several embedding schemes are based on the least significant bit (LSB) method (QIN, 2016; QIN C., 2017; SREENIVAS; KAMAKSHIPRASAD, 2017; MOLINA J., 2020; HAGHIGHI, 2019), since it provides a good trade-off among these performance metrics.

Chaotic maps are commonly used to add security to image watermarking schemes (SREENIVAS; KAMAKSHIPRASAD, 2017; TAI, 2018; JAFARI, 2019; AZEROUAL, 2017; HAGHIGHI, 2019). These maps are characterized by their sensitivity to the initial conditions and pseudo-random behavior, despite being deterministic, resulting in noise-like signals (RAWAT, 2011; TAI, 2018; LI, 2016). Applications of these maps include scrambling the original image (TAI, 2018; SREENIVAS; KAMAKSHIPRASAD, 2017; HAGHIGHI, 2019) and selecting sub-blocks to embed the watermark (TAI, 2018; HAGHIGHI, 2019). To support severe distortion imposed on the watermarked image, error correction codes can also be applied (LEFEVRE, 2019; SARRESHTEDARI, 2018).

1.1 LITERATURE REVIEW

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

Haghighi et al. (HAGHIGHI, 2019) 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 technique. 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 watermark 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. (JAFARI, 2019) 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.

In the image fragile watermark scheme proposed in (TAI, 2018), 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 (QIN, 2017), Qin et al. proposed a self-embedding fragile watermarking scheme using vector quantization (VQ) and index sharing. The watermark bits are composed by 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.(QIN, 2016) 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 (MOLINA J., 2020) embeds the watermark 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 (HSU, 2016), the original image is divided into non-overlapping sub-blocks of 2×2 pixels, called small blocks, and each 4×4 small blocks is grouped as a large block. In the proposed scheme, the watermark bits containing authentication information and recovery information are embedded into the LSB.

In (ABDELHAKIM, 2019), an 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 (SINGH, 2016) 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 (QIN C., 2017) 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, a pixel-wise recovery is employed to recover the original pixels with the assist of different neighboring overlapping blocks.

Peng et al. proposed in (PENG, 2018) an algorithm based on reversible data hiding. The authentication and recovery bits are embedded into two identical original images. A secret information is embedded in one image while a distortion information is embedded in the other one. In (SREENIVAS; KAMAKSHIPRASAD, 2017), K. 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 (SINHAL; ANSARI; AHN, 2020), a secret key based on pseudo-random binary sequences is used as a fragile watermark for tamper detection. The watermark bits are embedded using a LSB process in 9-base notation structure.

Li et al.(YUAN, 2021) 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.

1.2 OBJECTIVES AND CONTRIBUTIONS OF THE DISSERTATION

This dissertation has three main objectives.

- Propose two self-embedding fragile watermarking algorithms for image tamper localization and recovery using chaotic maps, transform 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 watermark bits with chaotic bits generated from a secret key.
- Investigate the trade-off between the imperceptibility of the watermarking embedding and the tampering detection/recovering capability of each algorithm.
- Compare the performance of the proposed algorithms with that of the existing fragile watermarking methods.

The main contributions are:

- Fragile watermarking scheme 1: This scheme is based on the DWT, in which the 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. After locating the tampered area, in the process of recovering damaged area, the parity bits and chaotic sequences are used to estimate the recovery bits.
- Fragile watermarking scheme 2: In this scheme, the bits from a chaotic sequence are used as authentication bits and are inserted into the LSB of each pixel of the original image, while a sequence of parity bits, obtained from the original image and chaotic bits, is randomly inserted in the sixth and seventh position (second and third LSB) of some pixels of the original image. These are used in the recovery process to estimate the original image.

1.3 ORGANIZATION

This dissertation is organized into five chapters.

- Chapter 1 presents the motivation, objectives and contributions of this dissertation.
- Chapter 2 presents the mathematical tools used in this work (Chaotic maps, Error-correcting codes, and DWT).
- Chapter 3 presents a new fragile watermarking algorithm in the DWT domain. The performance of the proposed scheme and previously presented algorithms are compared under a variety of attacks.
- Chapter 4 presents a new fragile watermarking algorithm in the LSB domain to improve detection and recovery capability. A performance comparison is performed between the two proposed methods.
- Chapter 5 presents the conclusions and suggestions for future work.

2 PRELIMINARIES

This section briefly introduces basic concepts on chaotic maps, error correction codes, and DWT which are necessary for the development of the fragile watermark algorithm presented in the next chapter.

2.1 CHAOTIC MAPS

Dynamical systems are generally described by differential equations in the case of continuous systems, or by difference equations in the case of discrete systems. The variables that describe the state of the dynamical system are called state variables, which may or may not be associated with physical quantities. The dynamical evolution of the system is completely determined by the set of differential or difference equations which defines the system and an initial condition. The state vector is represented by a vector of dimension T that contains the values of all variables of the dynamical system at a given time or iteration. The Euclidean space in which the system is defined is called *phase space*.

The behavior of unidimensional chaotic maps is observed through a discrete time series $\{x_i\}_{i=0}^{\infty}$, can be obtained by iterating a nonlinear and non-invertible function $f(x)$, over an initial condition x_0, as follows (STROGATZ, 2001)

$$x_n = f(x_{n-1}), n = 1, 2, 3, ... \tag{2.1}$$

An *orbit* of x_0 under $f(x)$ is a set of points $\{x_0, f(x_0), f^2(x_0)$ ... $\}$, where f^k denotes the k-th composition of $f(x)$. Examples of chaotic maps include the cubic map (MC) $f(x) = 4x^3 - 3x$ (LAU, 2003), and the logistic map (ML) $f(x) = rx(1 - x)$ (LAU, 2003), where r is a control parameter.

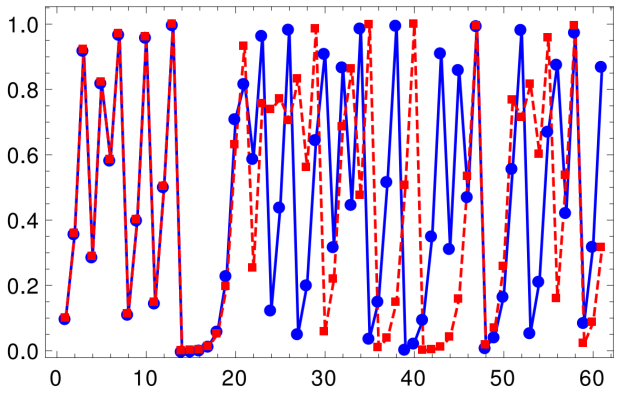
Chaotic systems are deeply sensitive to the initial condition of the system, meaning that infinitesimally close initial conditions generate low correlated sequences. A widely used metric to measure this sensitivity is the Lyapunov exponent, defined as

$$\lambda = \lim_{n\to\infty} \left[\frac{1}{n} \sum_{i=0}^{n-1} \ln(|f'(x[i])|) \right]. \tag{2.2}$$

where $f'(x)$ denotes the first derivative of $f(x)$. The Lyapunov exponent is the divergence rate between two sequences originated by infinitesimally close initial conditions. A positive Lyapunov exponent indicates that a chaotic map is sensitive to the initial condition. Figure 1 shows two orbits of a chaotic map generated by two initial conditions separated by 10^{-6}. These assume a distinct dynamical behavior after few iterations.

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

Figure 1 – Two different orbits for $x_0 = 0.1$ and $x_0' = 0.100001$.



Source: author (2021).

$\Pr(x_n \in \mathcal{R}_0) = \Pr(x_n \in \mathcal{R}_1) = 1/2$, 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 also chaotic systems in two dimensions. The iteration of the map from an initial condition (x_0, y_0) generates two-dimensional sequences $\{(x_0, y_0), (x_1, y_1), (x_2, y_2) \ldots\}$, where $(x_{n+1}, y_{n+1}) = f(x_n, y_n)$. For example, the two-dimensional Baker map is defined on the square
$f : [0, 1) \times [0, 1) \to [0, 1) \times [0, 1)$ by

$$f(x, y) = \begin{cases} (2x, \frac{y}{2}) & \text{for } 0 \leq x \leq 0.5; \\ (2x - 1, \frac{y+1}{2}) & \text{for } 0.5 < x \leq 1. \end{cases} \tag{2.3}$$

Another example of a two-dimensional map is the Arnold's cat map (ACM)
$f : [0, 1) \times [0, 1) \to [0, 1) \times [0, 1)$, given by:

$$f(x, y) = (2x + y, x + y). \tag{2.4}$$

The Discrete Arnold's Cat Map (DACM) is a generalization of the ACM for discrete sets, and it is defined as $f : Z_Q \times Z_Q \to Z_Q \times Z_Q$

$$f(x, y) = (2x + y, x + y) \pmod{Q} \tag{2.5}$$

where Z_Q is the integer ring modulus Q. The dynamical evolution generated by the DACM is represented in matrix form as

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = A \begin{bmatrix} x_n \\ y_n \end{bmatrix} \pmod{Q} \tag{2.6}$$

where

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \tag{2.7}$$

or

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = A^n \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \pmod{Q}. \tag{2.8}$$

2.2 ERROR-CORRECTING CODES

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 (LEFEVRE, 2019; LIN, 2004; TAN, 2019). In this dissertation we employ the binary Bose, Chaudhuri, and Hocquenghem (BCH) code over the Galois Field GF(q) with the following parameters: q is a prime number, $n = q^m - 1$ is the codeword length (where m is an integer), k is the number of information bits, and the error correcting capability of the code is t. 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 $g_i \in$ GF(q). The degree of $g(x)$ is equal to the number of parity check bits of the code. A polynomial representation $c(x)$ of a codeword $\mathbf{c} = (c_0, \ldots, c_{n-1})$ 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. In this work, we firstly use $q = 2$, $m = 4$, with $n = 15$, and $k = 11$ information bits, which has 4 parity bits and $t = 1$. Let α be a primitive element of GF(2^4), and let $m_i(x)$ be the minimal polynomial of α^i in GF(2^4). The generator polynomial $g(x)$ is obtained from the least common multiple of the minimum polynomials $g(x) =$ LCM$(m_1(x), m_2(x), \ldots, m_{d-1}(x))$, where d is the code minimum distance. For this one-bit error correction capability, the generator polynomial is $g(x) = x^4 + x + 1$. In order to analyze the impact of the code parameters on the watermarking algorithm, 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 Berlekamp-Massey (BM) algorithm is an algebraic decoding algorithm for the BCH code. The BM receives as input a received polynomial $r(x) = c(x) + e(x)$, where $e(x)$ is the error polynomial and the sum is over GF(q). The BM must find the minimum Hamming weight of the error vector **e** that could produce the received vector **r** (CLARK, 2013). For $i = 1, 2, \ldots, 2t$, the codeword is a multiple of the minimal polynomials $m_i(x)$. From the received word is obtained the syndrome (S_i), which is the remainder of the division between $r(x)$ and the minimal polynomial $m_i(x)$, and this depends on the error patterns only. After calculating $S_1, S_2, \ldots, S_{2t}$, the BM algorithm finds the error-locator polynomial $\sigma(x)$. The roots of $\sigma(x)$ determine the error-localization numbers α^i used to correct the error in $r(x)$ (LIN, 2004).

2.3 DISCRETE WAVELET TRANSFORM

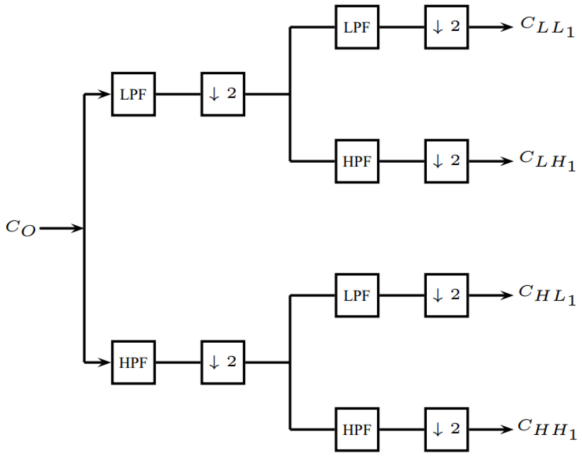
The DWT has been extensively used in image compression standards (SOWMYA, 2018). The basis functions of the DWT are generated from a basic wavelet function, through translations and dilations. These functions allow to reconstruct the original signal through the inverse discrete wavelet transform (IDWT). There are many types of wavelet functions, including Haar (ATAWNEH, 2017; GANGADHAR, 2018) (PANDEY, 2014; TAI, 2018), Daubechies (SOWMYA, 2018; FARGHALY, 2020), Symlets (AL-SHAYEA, 2019), Coifflets

(EL-HOSENY, 2019; THAKKAR, 2017). Due to its low computing requirements, the Haar transformation has been used primarily for image processing and pattern recognition and is adopted in this dissertation.

The wavelet transform depends on two functions, one in charge of scaling the wavelet function and another in charge of shifting the wavelet function. Using the scaling and shift variables, the wavelet transform allows a time-frequency analysis to be performed with a variable resolution. The scale function φ is in charge of analyzing the general behavior of the signal, while the wavelet function ψ is in charge of analyzing the behavior of the signal detail (STEPHANE, 2009).

Mallat (STEPHANE, 2009) proposed an algorithm based on a decomposition following a pyramid model, in which the image size decreases in each decomposition level. The implementation of this algorithm is carried out using filters and scaling functions. The low-pass filter (LFP) is associated with the scale function and allows analyzing the low-frequency components (this is considered the most important sub-band as it contains the main approximation of the original image), while the high-pass filter (HPF) is associated with the wavelet function and extracts the information regarding the high frequencies, that is, the details. 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. The parameters of each filter depend on the family of wavelet functions and the scaling used are always LFP and HPF.

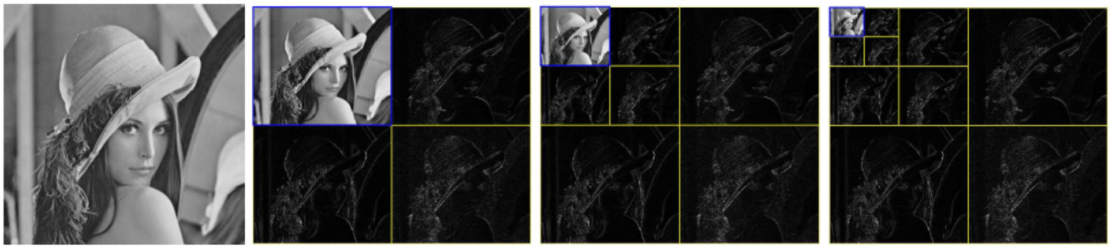
Figure 2 – Wavelet decomposition scheme in two dimensions.



Source: author, 2021.

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, as shown in Figure 3c. If we apply another decomposition level having as input the image C_{LL_2} we obtain the approximations C_{LL_3} and the levels of details C_{LH_3}, C_{HL_3}, C_{HH_3}, each one with size one eighth of the size of the image C_O, as shown in Figure 3d.

Figure 3 – (a) Original image, (b) first decomposition level, (c) second decomposition level, (d) third decomposition level.



Source: Matlab ToolBox, 2017b.

3 A FRAGILE IMAGE WATERMARKING SCHEME USING CHAOTIC SEQUENCES

A new fragile watermarking algorithm for images as well as a strategy for tamper detection and recovering of the tamperered areas are proposed in this chapter. The imperceptibility and recovery capacity of the proposed algorithm is compared to existing schemes in the literature.

3.1 THE PROPOSED ALGORITHM

The embedding algorithm E 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

$$C_W = E(C_O, K). \tag{3.1}$$

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

3.1.1 Watermark Embedding

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

1. Generate a chaotic binary sequence SC_1 using the cubic map with the key K.

2. Apply the 2-level 2D-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 watermark 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}

Source: author, 2021

3. Apply the 4-level 2D-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 ℓ_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}.

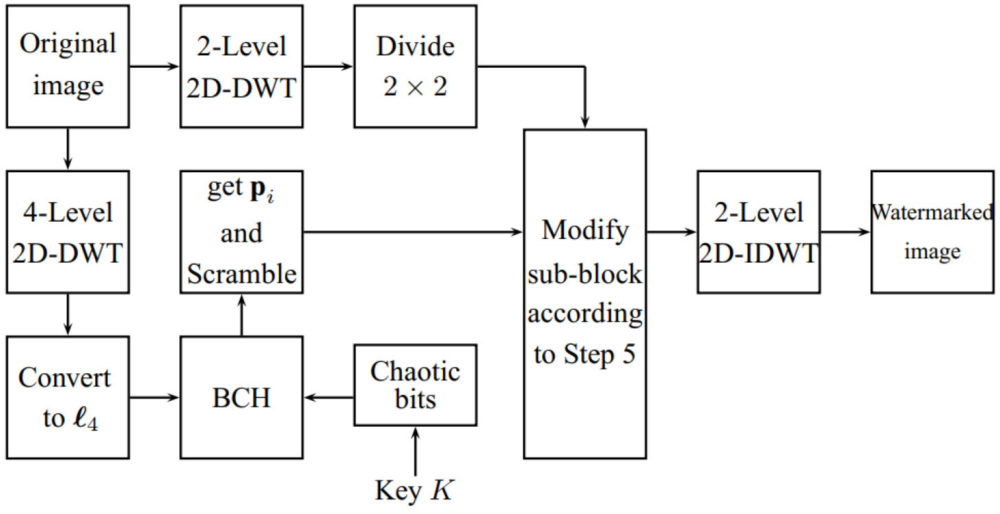
5. In each sub-block of C_{LH_2} and C_{HL_2} find the largest value (v_{max1}) and the second largest value (v_{max2}) of $c_{11}, c_{12}, c_{21}, c_{22}$. Let $\alpha_1 = v_{max1} - v_{max2}$. If $\alpha_1 \leq \alpha$, where α is a fixed positive parameter for all sub-blocks, then $v_{max1} \leftarrow v_{max1} + \alpha$, 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 $\mathbf{p}_i = 10$, 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 2-level 2D-IDWT and get 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 4 shows the block diagram of the proposed embedded algorithm, called Proposed 1.

Figure 4 – Block diagram of the proposed watermark embedding algorithm.



Source: author, 2021.

3.1.2 Watermark Extraction, Tamper Detection, and Image Recovery

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

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.

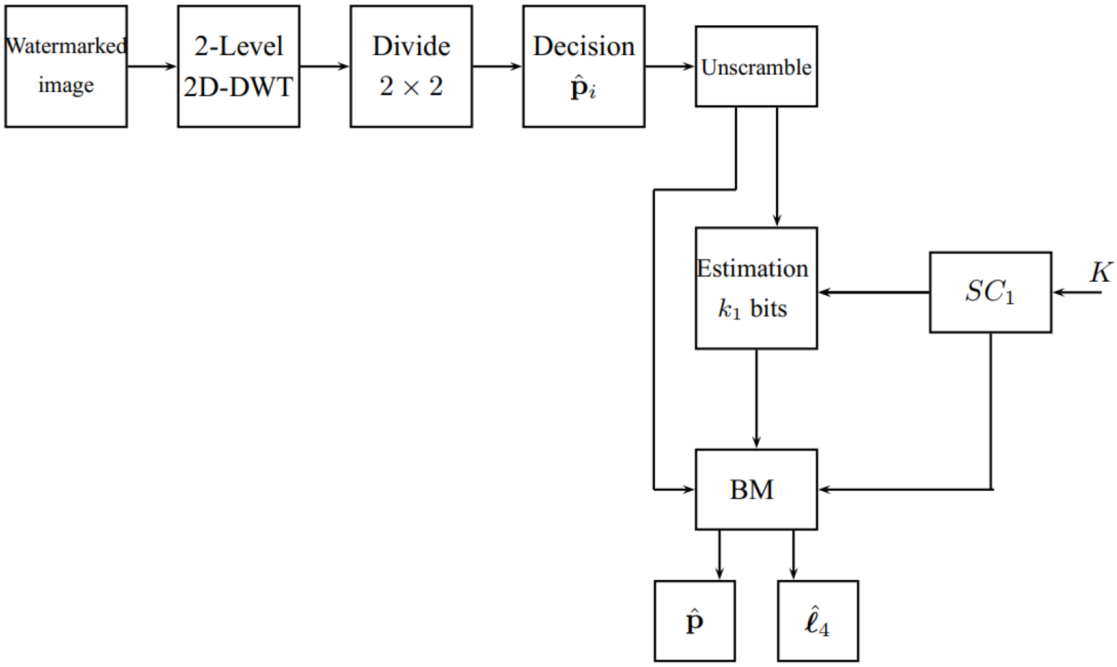
- Generate the chaotic binary sequence SC_1 from the key K.
- Calculate the 2-level 2D-DWT of C'_W obtaining the sub-bands C_{LH_2} and C_{HL_2}. Each sub-band is divided into sub-blocks of size 2×2.
- Find the highest value v'_{max} of each sub-block and its position. Decide the watermark information $\hat{\mathbf{p}}_i$ 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 4 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 5 shows the block diagram of the proposed watermark extraction algorithm.

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

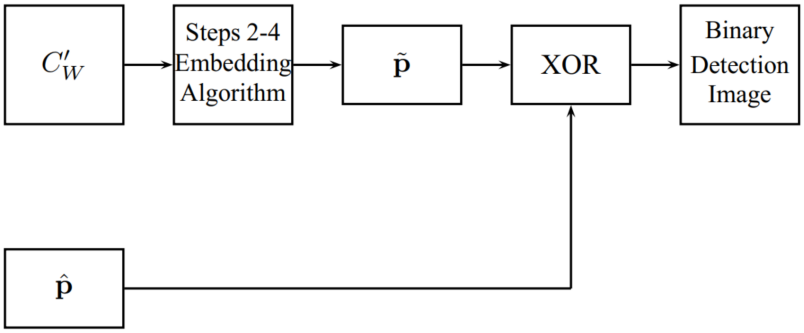
Figure 5 – Block diagram of the proposed watermark extraction algorithm.



Source: author, 2021.

size $M/4 \times N/4$ bits, which is called binary detection image. Figure 6 shows the block diagram of the proposed tamper detection algorithm.

Figure 6 – Block diagram of the proposed tamper detection algorithm.



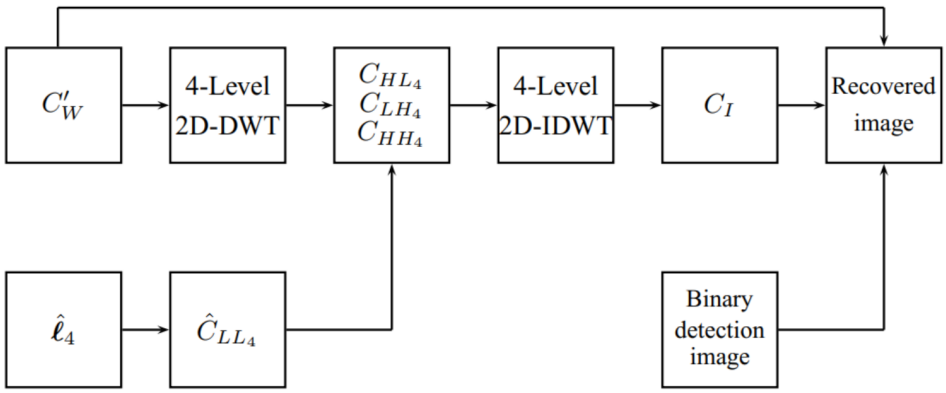
Source: author, 2021.

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 sub-bands C_{HH_4}, C_{HL_4} and C_{LH_4} of the tampered image C'_W. The binary sequence $\hat{\boldsymbol{\ell}}_4$ 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 2D-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 7 shows the block diagram of the proposed image recovery algorithm.

Figure 7 – Block diagram of the proposed recovery algorithm.



Source: author, 2021.

3.2 IMPERCEPTIBILITY, DETECTION AND RECOVERY METRICS

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

3.2.1 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 pixels values from 0 to 255, the PSNR is defined as

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2}{\text{MSE}}\right) \text{ (dB)} \tag{3.2}$$

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

$$\text{MSE} = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}(C_O(i,j) - C_W(i,j))^2. \tag{3.3}$$

The recovered PSNR, PSNR^r, is calculated using (3.2) 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

$$\text{SSIM} = \frac{(2\mu_O\mu_W + \gamma)(2\rho_{OW} + \beta)}{(\mu_O^2\mu_W^2 + \gamma)(\sigma_O^2\sigma_W^2 + \beta)} \tag{3.4}$$

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, α and β are fixed constants, $\gamma = 2.55$ and $\beta = 7.65$.

3.2.2 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

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{3.5}$$

$$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}} \tag{3.6}$$

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 tampered pixel, and TN is the number of pixels that are correctly identified as untampered pixel. The lower FPR and FNR indicates a better performance of the tamper detection algorithm.

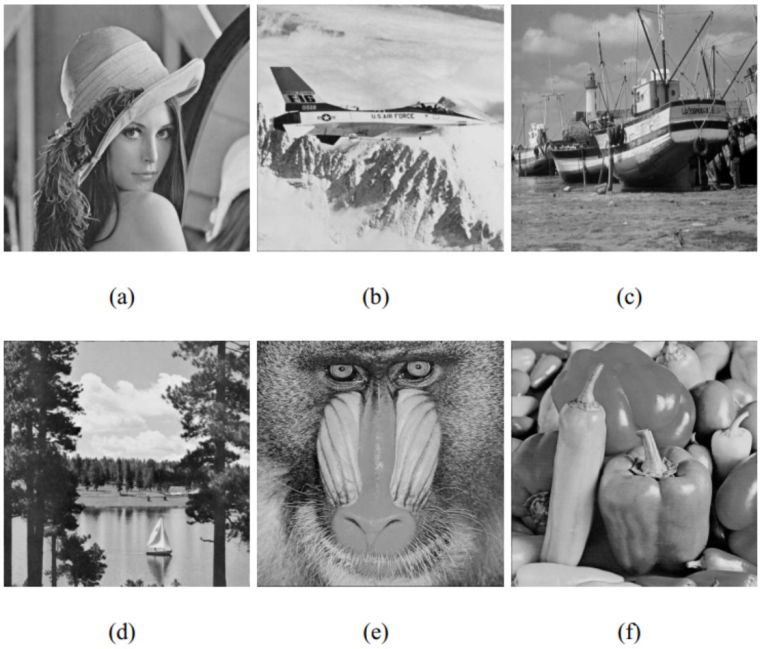
3.2.3 Watermark Image Attacks

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 (ABDELHAKIM, 2019).
- 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 (TAI, 2018; ABDELHAKIM, 2019).
- The second kind of collage attack (CA_2) modifies C_W by combining portions of another watermarked image and preserving their relative spatial locations (ABDELHAKIM, 2019; LEE C.F., 2019; TAI, 2018).
- In the normal tampering attack, some objects are added, deleted or modified on the watermarked image (HAGHIGHI, 2019).
- The salt and pepper attack consists in adding this noise with density d to the C_W image (LEE C.F., 2019).
- The constant-average attack (CAA) (ABDELHAKIM, 2019; TAI, 2018) 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(ABDELHAKIM, 2019).

The performance analysis conducted in this chapter uses 141 original images from the USC-SIPI database (http://sipi.usc.edu/database). This database contains grayscale and color 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 8 shows some examples of images used in this chapter.

Figure 8 – (a) Lena, (b) Airplane, (c) Boat, (d) Lake, (e) Baboon, (f) Pepper.



Source: author, 2021.

3.3 RESULTS

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. Next, we analyze the performance of the tamper detection algorithm for tampered images in this database. Table 2 shows the minimum and maximum values of 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 9(e)). We observe that these performance indicators remain almost unchanged for $\alpha > 0$. Hereafter, we fix the value of α to 0.01 in all simulations performed in this chapter.

Table 3 shows PSNR comparisons between the algorithm Proposed 1 and several existing watermarking fragile methods (QIN C., 2017; TAI, 2018; LEE C.F., 2019; ABDELHAKIM,

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	0.9532	0.9939	0.9530	0.9939	0.9527	0.9935	0.9521	0.9903

Source: author, 2021

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

Source: author, 2021

2019; HAGHIGHI, 2019; JAFARI, 2019). It can be seen from Table 3 that the algorithm Proposed 1 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
(QIN C., 2017)	44.27	43.85	44.37	42.49	44.23	44.31
(TAI, 2018)	44.14	44.14	44.28	44.19	44.17	44.01
(LEE C.F., 2019)	41.00	47.33	48.02	47.11	47.23	47.29
(ABDELHAKIM, 2019)	38.77	39.03	38.67	38.28	37.99	38.49
(HAGHIGHI, 2019)	45.82	45.81	45.76	45.79	45.80	45.79
(JAFARI, 2019)	44.32	44.74	45.06	44.73	44.57	45.11

Source: author, 2021

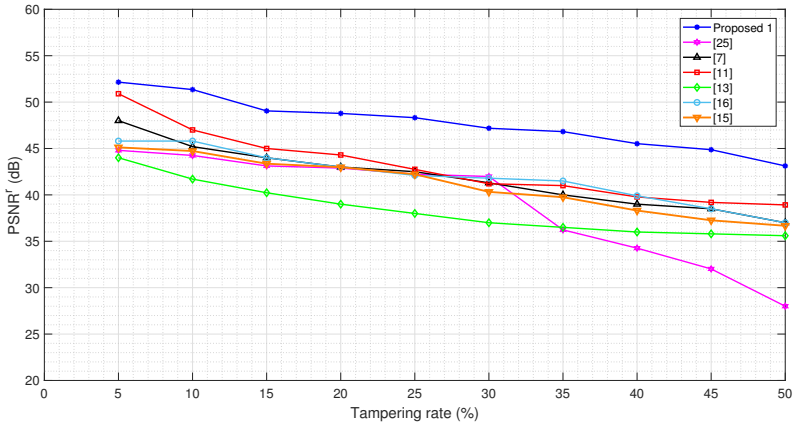
Figure 9 shows the tampered Lena images at various tampering rates, the corresponding binary detection images (the detected tampered region is marked in white color, whereas the non-tampered region is in black) and the recovered images. The quality of the recovered image is measured through the PSNR^r of the detected tampered region. The PSNR^r comparison under various tampering rates is illustrated in Figure 10 for the Lena image, where it is seen that the algorithm Proposed 1 provides better recovery performance. Table 4 shows a comparison of PSNR^r versus tampering rates for several images.

The results for the CA_1 attack for the Airplane, Pepper, Lake and Countryside images are provided in Figure 11. In each row of this figure, it is shown 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. The PSNR for these four

Figure 9 – 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%.

Source: author, 2021.

Figure 10 – Comparison of PSNR^r versus tampering rates for the Lena image.



Source: author, 2021.

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 tempering detection performance. The Proposed 1 scheme can also achieve good image recovery results with PSNR^r around 41 dB for the Airplane, Pepper and

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
	(HAGHIGHI, 2019)	44.16	41.84	40.22	38.17	36.55
	(JAFARI, 2019)	40.52	37.60	35.89	31.92	29.32
	(LEE C.F., 2019)	49.47	44.39	41.23	38.58	36.61
Baboon	Proposed 1	50.90	48.25	46.82	45.80	42.93
	(HAGHIGHI, 2019)	41.18	42.58	41.03	38.45	34.82
	(JAFARI, 2019)	41.80	39.75	36.16	32.51	30.80
	(LEE C.F., 2019)	38.69	35.55	33.95	32.93	32.13
Peppers	Proposed 1	51.08	48.80	46.88	45.07	43.05
	(HAGHIGHI, 2019)	44.07	41.74	40.39	39.19	38.02
	(JAFARI, 2019)	41.35	39.60	35.97	33.05	31.68
	(LEE C.F., 2019)	42.84	40.54	38.32	36.76	35.17
Airplane	Proposed 1	50.84	48.93	47.01	45.33	43.07
	(HAGHIGHI, 2019)	41.99	40.24	38.57	36.99	35.95
	(JAFARI, 2019)	40.38	38.20	36.07	33.94	31.91
	(LEE C.F., 2019)	46.59	44.54	42.83	40.32	36.79

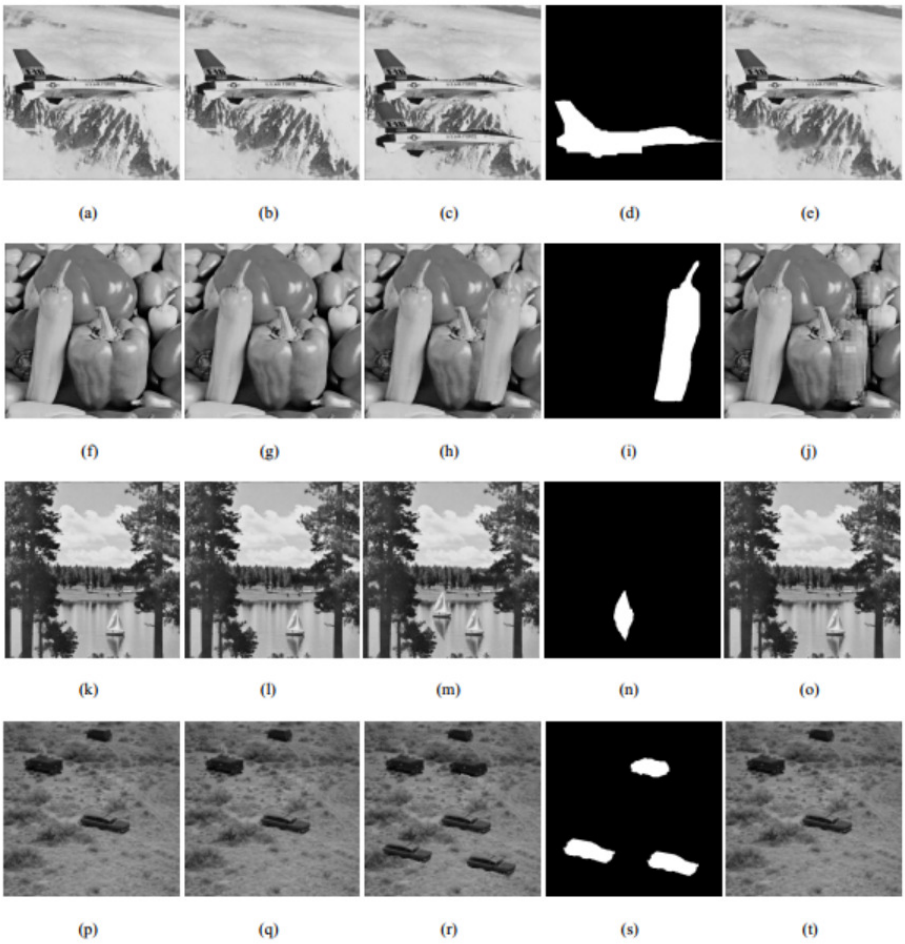
Source: author, 2021

Countryside images and around 47 dB for the Lake image.The CA_2 attack is considered in Figure 12 for 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, it is shown two watermarked images with their PSNR values, 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 13 in which some objects are added to the watermarked images (Lena, Elaine,Airport, and Aerial View).

The results for the CAA attack is presented in Figure 14 in which a distortion is created in 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 15. The FPR and FNR are 0.143 and 0.084, respectively.

Some attacks displayed in Figures 11, 12, 14 and 15 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.

Figure 11 – 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).



Source: author, 2021.

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

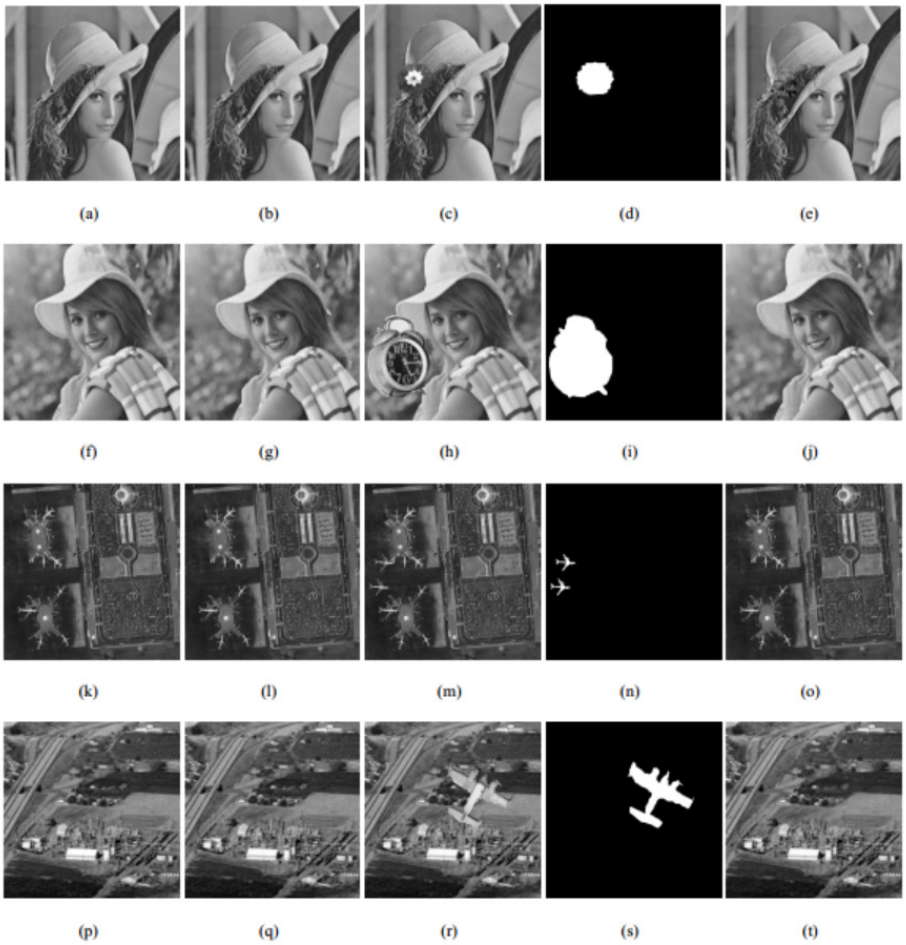
Figure	Image	PSNR^r		
		Proposed 1	Other Schemes	
11(j)	Pepper	40.98	(TAI, 2018)	37.10
11(o)	Lake	47.52	(YUAN, 2021)	46.03
12(e)	Baboon	39.86	(TAI, 2018)	31.35
13(o)	Airport	47.42	(YUAN, 2021)	46.03
14(e)	Boat	45.53	(TAI, 2018)	35.41
15(e)	Lena	33.78	(LEE C.F., 2019)	40.68

Source: author, 2021

Figure 12 – 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).

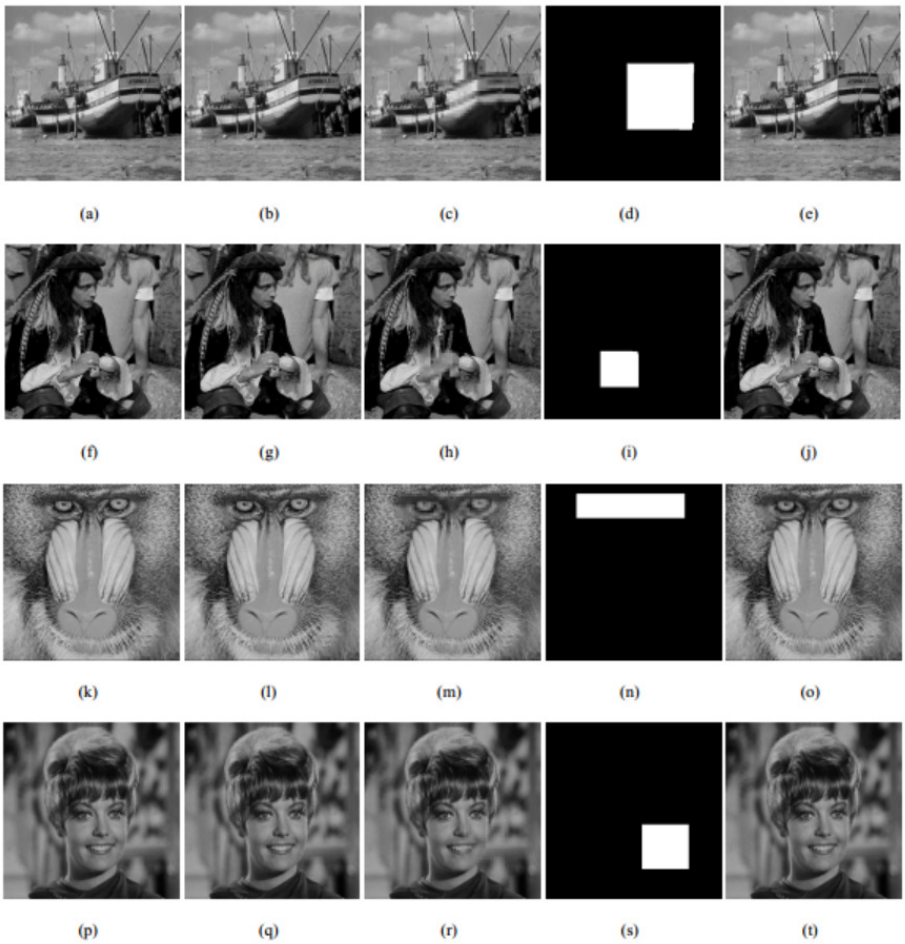
Source: author, 2021.

Figure 13 – 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).



Source: author, 2021.

Figure 14 – 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).



Source: author, 2021.

Figure 15 – 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) (b) (c) (d) (e)

Source: author, 2021.

3.3.1 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 1-level 2D-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 watermark 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 $\boldsymbol{\ell}_4$ and $k_2 = 19$ from the chaotic map. After scrambling, we get $\mathbf{p} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{\frac{MN}{64}}\}$, 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 4 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 2D-IDWT and get 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). It is observed a decrease in the PSNR value due to the insertion at a higher level of the DWT decomposition (1-level). Table 7 presents a similar 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 11, 12, 13, 14 and 15. 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

Source: author, 2021

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

Source: author, 2021

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

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

Source: author, 2021

4 A FRAGILE IMAGE WATERMARKING SCHEME IN THE LSB DOMAIN

Different algorithms use the LSB method due to better tampered detection (QIN C., 2017; QIN, 2017; LEE C.F., 2019; ABDELHAKIM, 2019; RAKHMAWATI, 2019). A modification of the previously proposed watermarking scheme is presented in this chapter. The watermark bits are embedded using in the LSB method.

4.1 THE WATERMARKING ALGORITHM

The new embedding algorithm has as input an 8-bit grayscale original image C_O of size $M \times N$ pixels and a key K. The watermark bits are obtained from C_O and a key K.

4.1.1 Watermark Embedding

The watermark bits are embedded in the original image according to the following steps.

- Generate a binary chaotic sequence SC_1 from the key K and divide this into two sub-sequences (SC_{1a}, SC_{1b}). SC_{1a} is used to detect modifications of the watermarked image and SC_{1b} is part of the information sequence of the BCH code.
- Apply the 2-level 2D-DWT decomposition to the original image C_O. The image C_{LL_2} has size $M/4 \times N/4$ pixels. Convert each byte of this image to a binary sequence and a new sequence $\boldsymbol{\ell}_2$ is formed from the 6 MSBs of each pixel, being of length $\frac{3MN}{8}$ bits.
- The parity sequence $\mathbf{p}$ of the BCH (31,21,2) code is generated as follows. The 21 information bits are obtained by concatenating $k_1 = 2$ bits from $\boldsymbol{\ell}_2$ and $k_2 = 19$ from the chaotic map (SC_{1b} sequence). After encoding, a 31-bit codeword is obtained with 10 parity bits. After repeating this process for the entire $\boldsymbol{\ell}_2$, a parity sequence of size $\frac{15MN}{8}$ bits is obtained. This sequence is scrambled and divided into sub-sequences of length 2 bits, $\mathbf{p} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{\frac{15MN}{16}}\}$, where $\mathbf{p}_i = p_{i1}, p_{i2}$.
- Each bit of SC_{1a} replaces the LSB of each pixel of C_O and each $\mathbf{p}_i$ replaces the sixth and seventh position (second and third LSB) of pixels of C_O generating the watermarked image C_W.

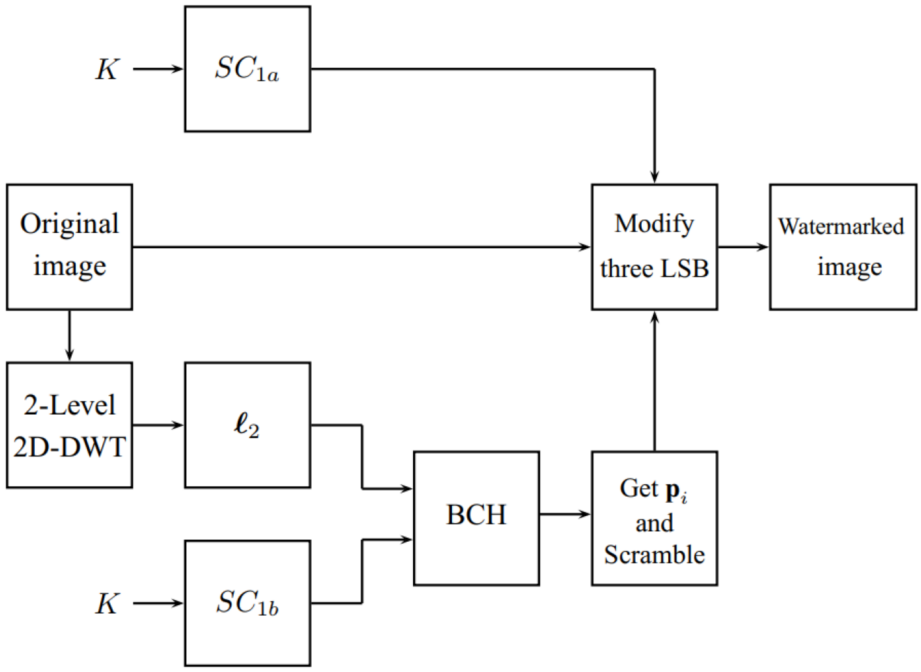
Figure 16 shows the block diagram of the proposed embedding algorithm (called Proposed 2).

4.1.2 Watermark Extraction, Tamper Detection, and Image Recovery

The watermark extraction follows the steps.

- Extract the sequence SC'_{1a} from the LSB of each pixel of C'_W.

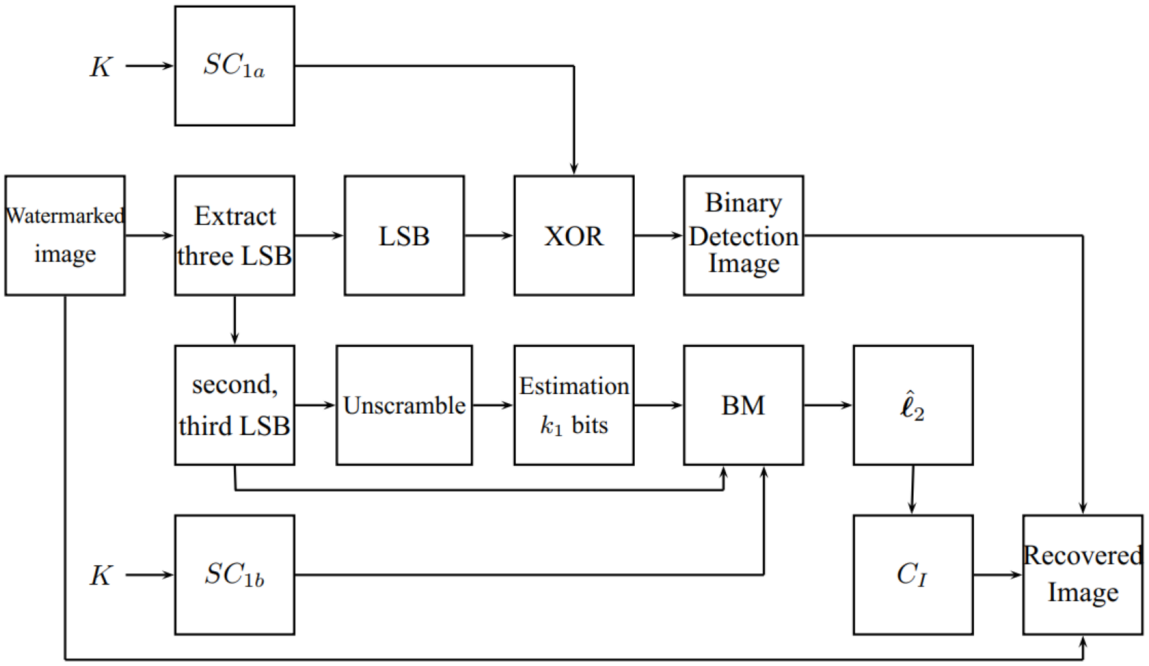
Figure 16 – Block diagram of the watermark embedding algorithm.



Source: author, 2021.

- Generate the binary chaotic sequences SC_{1a} and SC_{1b} from K.
- The binary detection image is obtained from the XOR operation between SC'_{1a} and SC_{1a}.
- Extract sequence $\mathbf{p}'$ from the second and the third LSB of pixels of C'_W. Unscramble this sequence and obtain an estimate of the parity sequence $\hat{\mathbf{p}}$.
- The sequence $\hat{\mathbf{p}}$ is divided into a sub-sequence of 10 bits each, obtaining $\hat{\mathbf{p}}_l = \{p_{l0}, p_{l1}, \ldots, p_{l9}\}$. From $\hat{\mathbf{p}}_l$ and SC_{1b}, there are 2^{k_1} possible parity sequences, depending on the remaining k_1 information bits. An estimate of these bits is obtained from the smallest Hamming distance between $\hat{\mathbf{p}}_l$ and these possible parity sequences. Then, concatenate the estimated k_1 bits, the k_2 chaotic bits, and the 10 parity bits with the smallest Hamming distance to form a 31-bit word. This word is decoded using the BM algorithm, giving a new estimate of the k_1 bits of the sequence $\boldsymbol{\ell}_2$.
- This procedure is repeated for each $l = 1, \ldots, (\frac{3MN}{16})$, obtaining the sequence $\hat{\boldsymbol{\ell}}_2$.
- Calculate the details sub-bands C_{HH_2}, C_{HL_2} and C_{LH_2} of the tampered image C'_W. The binary sequence $\hat{\boldsymbol{\ell}}_2$ is converted to the image $\hat{C}_{LL_2}$ of size $M/4 \times N/4$ pixels. An intermediate image C_I is obtained from the 2-level 2D-IDWT of the image formed from $\hat{C}_{LL_2}$, C_{HH_2}, C_{HL_2}, and C_{LH_2}. 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 17 – Block diagram of the recovered image.



Source: author, 2021.

4.2 RESULTS

We do a performance comparison between the algorithms presented in the previous chapter (Proposed 1, Proposed 1-v1) and in this chapter (Proposed 2).

Table 9 compares the PSNR for several images and Table 10 compares the FPR, FNR, and PSNR^r for the attacks carried out in Figures 11, 12, 13, 14 and 15. The PSNR^r comparison under various tampering rates is illustrated in Figure 18 where we observe that algorithm Proposed 2 provides better recovery performance than the Proposed 1 and Proposed 1-v1 at the expense of a decrease in imperceptibility.

Table 9 – PSNR comparison for several original images.

Algorithm	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
Proposed 2	41.92	41.08	42.07	40.51	41.66	39.95

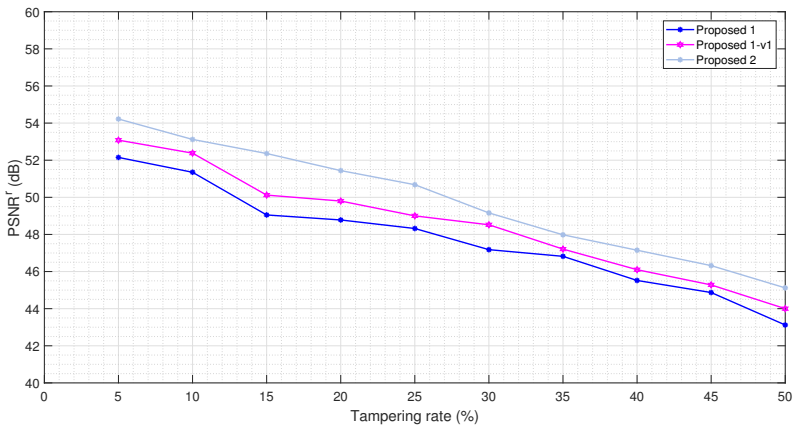
Source: author, 2021

Table 10 – FPR, FNR, and $PSNR^r$ comparisons for several attacks.

Figure		Proposed 1		Proposed 1-v1		Proposed 2		Proposed 1	Proposed 1-v1	Proposed 2
		FPR	FNR	FPR	FNR	FPR	FNR	$PSNR^r$	$PSNR^r$	$PSNR^r$
11	(a)	0.073	0.009	0.070	0.007	0.010	0	41.02	42.64	44.80
	(f)	0.117	0.008	0.112	0.009	0.028	0	40.98	43.16	44.43
	(k)	0.100	0.002	0.101	0.002	0.011	0	47.52	49.83	50.61
	(p)	0.052	0.007	0.042	0.006	0.016	0	42.17	43.78	46.33
12	(a)	0.099	0.007	0.087	0.006	0.012	0	39.86	42.07	44.35
	(f)	0.087	0.002	0.079	0.002	0.009	0	46.32	47.91	50.09
	(k)	0.090	0.005	0.079	0.005	0.011	0	38.56	40.55	43.24
	(p)	0.106	0.008	0.0902	0.006	0.024	0	43.55	44.18	45.89
13	(a)	0.035	0.003	0.024	0.004	0.008	0	47.25	49.06	51.80
	(f)	0.103	0.008	0.095	0.007	0.012	0	40.28	42.33	45.24
	(k)	0.020	0.001	0.036	0.006	0.032	0	47.42	49.28	50.12
	(p)	0.135	0.007	0.110	0.005	0.050	0	46.12	47.23	48.14
14	(a)	0.030	0.007	0.041	0.005	0.009	0	45.53	46.90	48.98
	(f)	0.025	0.003	0.018	0.003	0.008	0	44.92	47.03	48.22
	(k)	0.026	0.004	0.020	0.002	0.008	0	44.31	46.89	48.87
	(p)	0.012	0.001	0.008	0.001	0.006	0	45.18	46.92	48.10
15	(a)	0.143	0.084	0.135	0.082	0.020	0.005	33.78	35.21	39.06

Source: author, 2021

Figure 18 – Comparison of $PSNR^r$ versus tampering rates for the Lena image.



Source: author, 2021.

4.3 PERFORMANCE OF THE PROPOSED ALGORITHMS FOR COLOR IMAGES

The performance of the proposed algorithms in color images is analyzed in terms of imperceptibility, detection and recovery. We also present comparisons with literature results.

The original color image C_O is represented into three components R, G, and B, each one of size $M \times N$. A fragile watermarking algorithm in grayscale image is applied in each component. We adopt the same performance metrics used for grayscale images. Table 11 presents an imperceptibility comparison between the proposed algorithms and some exiting ones for several images. All proposed algorithms present better imperceptibility results, and the highest PSNR values are achieved by Proposed 1. Table 12 shows a similar comparison of PSNR^r versus several tampered rates. The algorithm Proposed 2 provides better results.

Some attacks presented in the previous chapter are presented in Figure 19 with recovered results for the algorithm Proposed 2. It is observed a behavior similar to that obtained with grayscale images, obtaining FNR and FPR values close to zero (desired values) and PSNR^r values higher than 40 dB. A similar analysis is performed in Figure 20 for color images of size 256×256 pixels. Table 13 shows a comparison between the FPR, FNR, and PSNR^r for the attacks carried out in Figure 19 and Figure 20.

Table 11 – PSNR comparison for several original color 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
Proposed 2	47.26	47.98	46.85	47.63	48.01	46.72
(HAGHIGHI, 2019)	46.45	46.23	46.22	46.18	46.03	46.24
(MOLINA J., 2020)	44.60	44.69	44.66	44.61	44.54	44.64
(SINHAL, 2020)	49.88	49.88	49.87	49.87	49.70	49.88

Source: author, 2021

Table 12 – 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
	Proposed 2	55.08	52.51	49.84	48.06	46.94
	(HAGHIGHI, 2019)	44.22	39.77	37.64	35.91	34.80
	(MOLINA J., 2020)	37.16	33.83	31.48	29.07	26.96
	(SINHAL, 2020)	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
	Proposed 2	54.48	51.73	49.25	47.98	46.88
	(HAGHIGHI, 2019)	42.00	38.05	37.05	35.00	32.50
	(MOLINA J., 2020)	35.85	31.87	28.38	25.59	23.59
	(SINHAL, 2020)	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
	Proposed 2	56.00	51.86	50.13	49.22	47.51
	(HAGHIGHI, 2019)	44.02	40.00	39.20	37.00	35.92
	(MOLINA J., 2020)	37.38	34.63	32.48	29.89	27.31
	(SINHAL, 2020)	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
	Proposed 2	55.12	52.77	49.93	48.81	47.60
	(HAGHIGHI, 2019)	41.90	40.00	39.00	36.95	35.00
	(MOLINA J., 2020)	36.51	33.40	31.28	28.51	25.99
	(SINHAL, 2020)	42.72	34.81	30.24	28.16	26.42

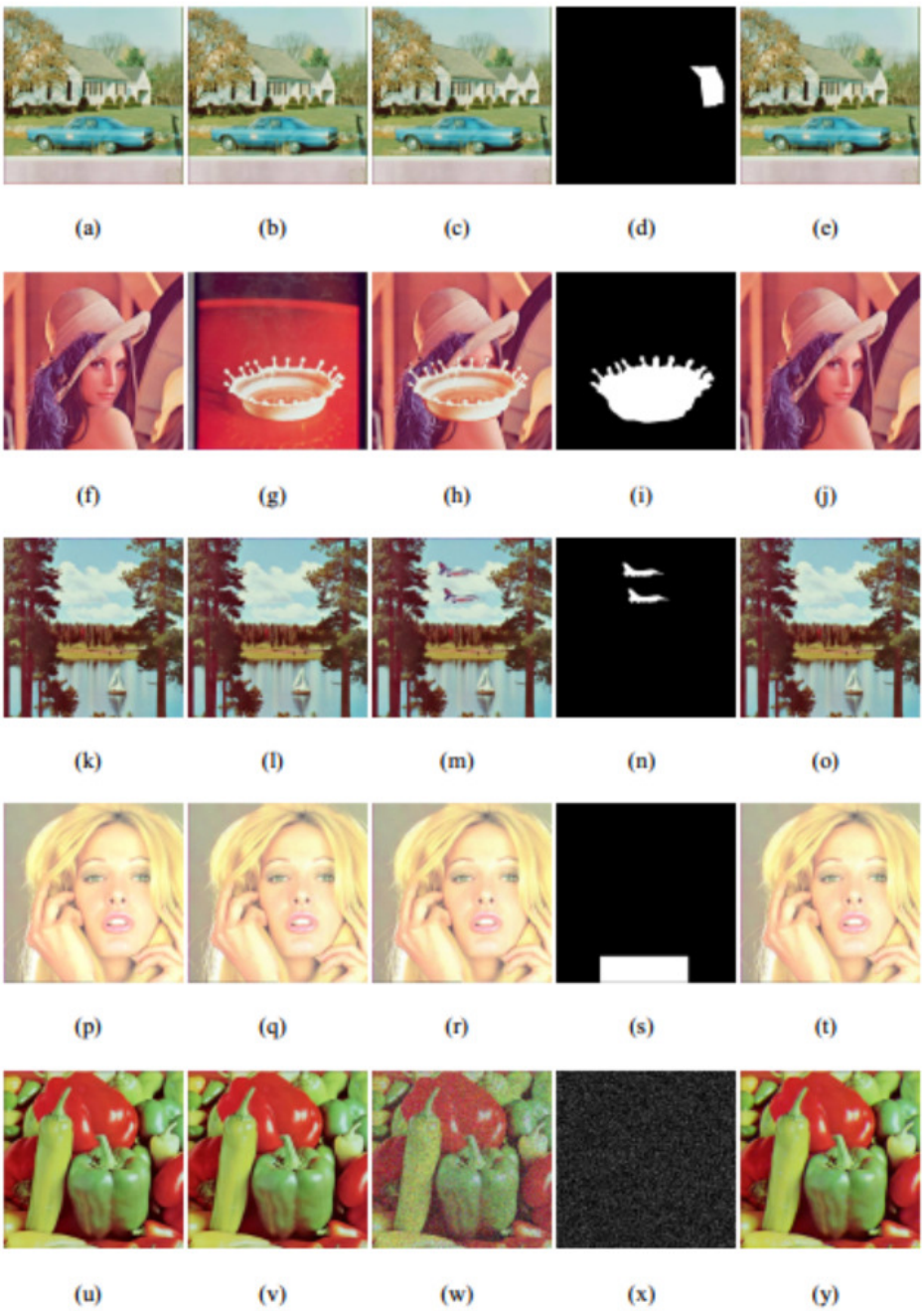
Source: author, 2021

Table 13 – FPR, FNR, and PSNR^r comparisons for several attacks for color images.

Figure		Proposed 1		Proposed 1-v1		Proposed 2		Proposed 1	Proposed 1-v1	Proposed 2
		FPR	FNR	FPR	FNR	FPR	FNR	PSNR^r	PSNR^r	PSNR^r
19	(a)	0.069	0.008	0.062	0.006	0.009	0	51.61	52.33	55.31
	(f)	0.083	0.011	0.079	0.011	0.013	0	47.56	49.02	50.32
	(k)	0.051	0.005	0.049	0.006	0.005	0	49.34	51.72	56.12
	(p)	0.023	0.004	0.021	0.003	0.002	0	51.04	52.68	55.15
	(u)	0.103	0.062	0.098	0.055	0.028	0.008	34.08	34.69	35.92
20	(a)	0.075	0.009	0.072	0.009	0.008	0	48.39	50.03	54.28
	(f)	0.066	0.008	0.060	0.009	0.004	0	51.53	52.88	55.64
	(k)	0.062	0.008	0.064	0.008	0.005	0	51.80	53.04	55.34
	(p)	0.025	0.005	0.023	0.005	0.005	0	51.63	52.81	54.88
	(u)	0.108	0.058	0.093	0.053	0.037	0.011	33.83	34.73	36.15

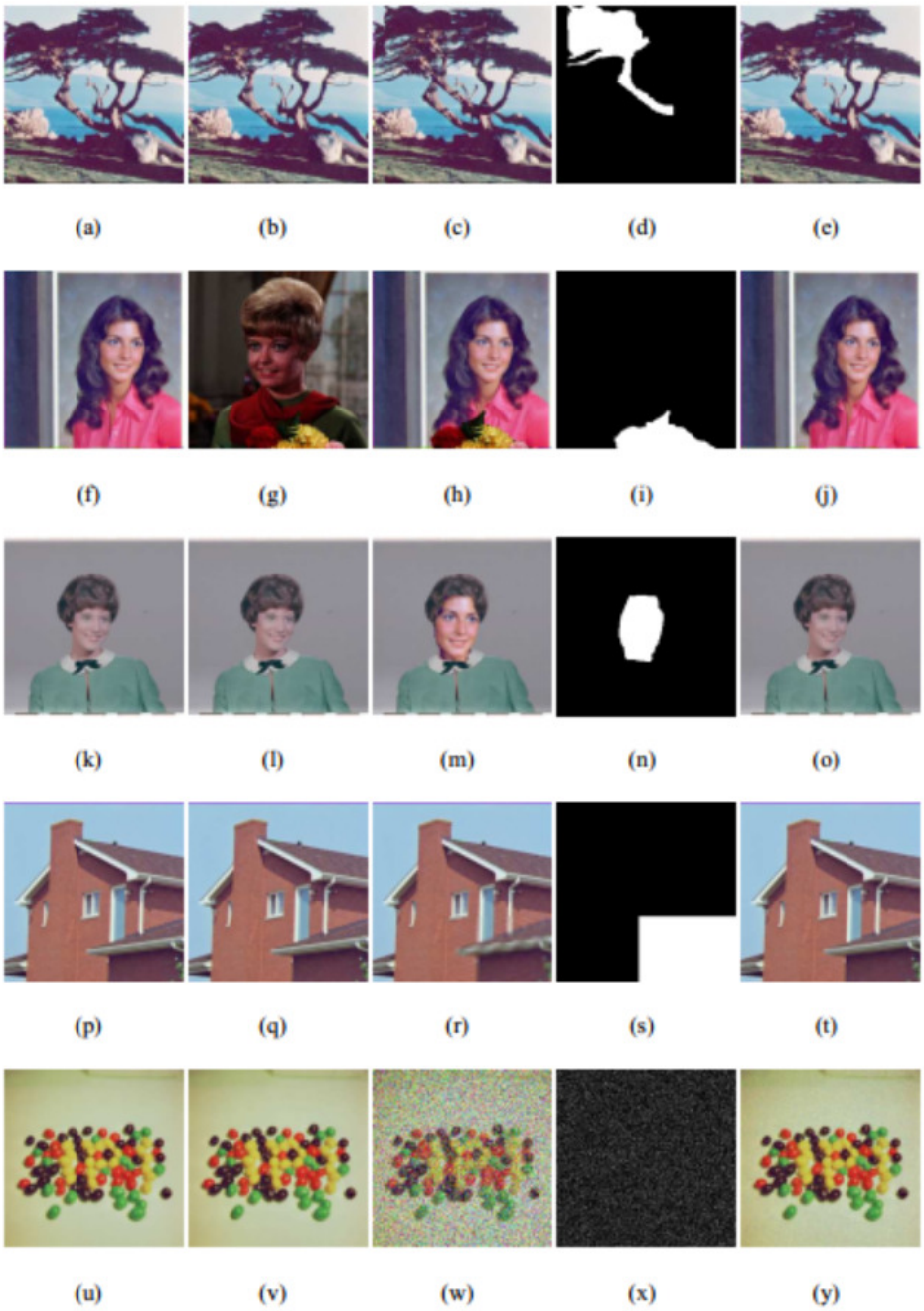
Source: author, 2021

Figure 19 – Different attacks on color 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 = 0.009 and FNR = 0), (e) recovered image (PSNR^r = 55.31 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 $= 0.013$ and FNR $= 0$), (j) recovered image (PSNR^r = 50.32 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 $= 0.005$ and FNR $= 0$), (o) recovered Tiffany image (PSNR^r = 56.12 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 $= 0.002$ and FNR $= 0$), (t) recovered image (PSNR^r = 55.15 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 $= 0.028$ and FNR $= 0.008$), (y) recovered image (PSNR^r = 35.92 dB).



Source: author, 2021.

Figure 20 – Different attacks on color images (a-e) tampering recovery for the CA1 attack (a) original Tree image, (b) watermarked Tree image (PSNR 46.32 dB), (c) tampered image (13%), (d) binary detection image (FPR =0.008 and FNR =0), (e) recovered image (PSNR^r = 54.28 dB), (f-j) tampering recovery for the CA_2 attack: (f) watermarked Female1 image (PSNR = 45.38 dB), (g) watermarked Female2 image (PSNR = 46.06 dB), (h) tampered image (6%), (i) binary detection image (FPR $= 0.004$ and FNR $= 0$), (j) recovered image (PSNR^r = 55.64 dB), (k-o) tampering recovery for normal tampering attack: (k) original Female3 image , (l) watermarked Female3 image (PSNR = 44.52 dB), (m) tampered image (7%), (n) binary detection image (FPR $= 0.005$ and FNR $= 0$), (o) recovered image (PSNR^r = 55.34 dB), (p-t) tampering recovery for the CAA attack: (p) original House image, (q) watermarked image (PSNR = 46.32 dB), (r) tampered image (20%), (s) binary detection image (FPR $= 0.005$ and FNR $= 0$), (t) recovered image (PSNR^r = 54.88 dB), (u-y) Salt and Pepper attack (u) original Seeds image, (v) watermarked Seeds image PSNR = dB, (w) tampered image (30%), (x) binary detection image (FPR = 0.037 and FNR = 0.011), (y) recovered image (PSNR^r = 36.15 dB).



Source: author, 2021.

5 CONCLUSIONS AND FUTURE WORK

This chapter summarizes the main contributions and results obtained in this dissertation and comments on possible future work. Two self-embedding fragile watermarking algorithms are proposed for tamper detection and content recovery in images. The watermark 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 first algorithm consists of two versions where the watermark bits are embedded in the original image in the frequency domain using the DWT. For both versions, 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 a higher recoverability.

We also present a watermarking algorithm using the LSB method, using the LSB for tampering detection and the second and third LSB for recovery. Comparing the two proposed algorithms, it can be seen that the latter presents higher PSNR^r (an improvement around 2 dB), but with lower PSNR (a loss around 1 dB). Finally, the proposed algorithms are applied in color images, using the same insertion, extraction, detection, and recovery process in each component (R, G, and B).

A natural continuation of this work is the incorporation of other classes of codes, such as codes with unequal error protection (UEP), since part of the information sequence is known at the decoder. A statistical study of the behavior of the chaotic sequence within the coded message must be carried out, as well as an understanding on how to use these bits to obtain better recovery. Another topic for future work is to propose watermarking schemes for Internet of Things (IoT) systems. Due to the integration of services, the IoT devices can be susceptible to attacks. It would be useful to develop watermarking algorithms using chaotic maps to solve this problem. Two important points of this system need to be addressed, data authentication and property authentication. Watermarking can be used to offer these characteristics, even though there are conflicting resources, since the data authentication is obtained with fragile watermarking and property authentication is obtained with robust watermarking. One possibility is to use the parity bits as fragile watermarking bits and the bits that are part of the coded message (not chaotic bits) as the robust watermarking bits.

BIBLIOGRAPHY

ABDELHAKIM, o. Fragile watermarking for image tamper detection and localization with effective recovery capability using k-means clustering. **Multimedia Tools and Applications**, v. 78, n. 22, p. 32523–32563, 2019.

AL-SHAYEA, o. A hybridized methodology of different wavelet transformations targeting medical images in IoT infrastructure. **Measurement**, v. 148, p. 106–813, 2019. ISSN 0263-2241.

ATAWNEH, o. Secure and imperceptible digital image steganographic algorithm based on diamond encoding in DWT domain. **Multimedia Tools and Applications**, v. 76, n. 18, p. 51–72, 2017. ISSN 1573-7721.

AZEROUAL, o. Real-time image tamper localization based on fragile watermarking and Faber-Schauder wavelet. **AEU - International Journal of Electronics and Communications**, v. 79, p. 207–218, 2017. ISSN 1434-8411.

CLARK, o. G. **Error-Correction Coding for Digital Communications**. [S.l.]: Springer Science & Business Media, 2013.

EL-HOSENY, o. An optimal wavelet-based multi-modality medical image fusion approach based on modified central force optimization and histogram matching. **Multimedia Tools and Applications**, v. 78, p. 73–97, 2019. ISSN 1573-7721.

FARGHALY, o. Floating-point discrete wavelet transform-based image compression on FPGA. **AEU - International Journal of Electronics and Communications**, v. 124, p. 53–63, 2020. ISSN 1434-8411.

GANGADHAR, o. An evolutionary programming approach for securing medical images using watermarking scheme in invariant discrete wavelet transformation. **Biomedical Signal Processing and Control**, v. 43, p. 31–40, 2018. ISSN 1746-8094.

HAGHIGHI, o. B. TRLG: Fragile blind quad watermarking for image tamper detection and recovery by providing compact digests with optimized quality using LWT and GA. **Information Sciences**, v. 486, p. 204–230, 2019. ISSN 0020-0255.

HSU, o. Image tamper detection and recovery using adaptive embedding rules. **Measurement**, v. 88, p. 287–296, 2016. ISSN 0263-2241.

JAFARI, o. A new digital image tamper detection algorithm based on integer wavelet transform and secured by encrypted authentication sequence with 3D quantum map. **Optik**, v. 187, p. 205–222, 2019. ISSN 0030-4026.

KO, o. Robust and blind image watermarking in DCT domain using inter-block coefficient correlation. **Information Sciences**, v. 517, p. 128–147, 2020. ISSN 0020-0255.

LAU, o. **Chaos-Based Digital Communication Systems: Operating Principles, Analysis Methods, and Performance Evaluation**. Berlin Heidelberg: Springer-Verlag, 2003. (Signals and Communication Technology). ISBN 978-3-540-00602-2.

LEE C.F., o. Self-Embedding authentication watermarking with effective tampered location detection and high-quality image recovery. **Sensors**, v. 19, n. 2267, p. 1–18, 2019.

LEFEVRE, o. Application of rank metric codes in digital image watermarking. **Signal Processing: Image Communication**, v. 74, p. 119–128, 2019. ISSN 0923-5965.

LI, o. A recoverable chaos-based fragile watermarking with high PSNR preservation. **Security and Communication Networks**, v. 9, n. 14, p. 237–238, 2016. ISSN 1939-0122.

LIN, o. **Error Control Coding**. [S.l.]: Pearson-Prentice Hall, 2004.

MOLINA J., o. An effective fragile watermarking scheme for color image tampering detection and self-recovery. **Signal Processing: Image Communication**, v. 81, p. 1–20, 2020.

MOOSAZADEH, o. A new DCT-based robust image watermarking method using teaching-learning-Based optimization. **Journal of Information Security and Applications**, v. 47, p. 28–38, 2019. ISSN 2214-2126.

NASKAR, o. **Reversible Digital Watermarking: Theory and Practices**. [S.l.]: Morgan & Claypool Publishers, 2014.

PANDEY, o. Rightful ownership through image adaptive DWT-SVD watermarking algorithm and perceptual tweaking. **Multimedia Tools and Applications**, v. 72, p. 723–748, 2014. ISSN 1573-7721.

PENG, Y. Image authentication scheme based on reversible fragile watermarking with two images. **Journal of Information Security and Applications**, v. 40, p. 236–246, 2018.

QIN C., o. Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy. **Signal Processing**, v. 138, p. 280–293, 2017.

QIN, o. Self-embedding fragile watermarking based on reference-data interleaving and adaptive selection of embedding mode. **Information Sciences**, v. 373, p. 233–250, 2016. ISSN 0020-0255.

QIN, o. Fragile image watermarking scheme based on VQ index sharing and self-embedding. **Multimedia Tools and Applications**, v. 76, n. 2, p. 226–228, 2017. ISSN 1573-7721.

RAKHMAWATI, o. A recent survey of self-embedding fragile watermarking scheme for image authentication with recovery capability. **EURASIP Journal on Image and Video Processing**, v. 2019, n. 1, p. 61, 2019. ISSN 1687-5281.

RAWAT, o. A chaotic system based fragile watermarking scheme for image tamper detection. **AEU - International Journal of Electronics and Communications**, v. 65, n. 10, p. 840–847, 2011.

SARRESHTEDARI, o. Source channel coding-based watermarking for self-embedding of JPEG images. **Signal Processing: Image Communication**, v. 62, p. 106–116, 2018. ISSN 0923-5965.

SHIH. **Image Processing and Pattern Recognition: Fundamentals and Techniques**. [S.l.]: John Wiley & Sons, 2010.

SINGH, o. Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability. **Journal of Visual Communication and Image Representation**, v. 38, p. 775–789, 2016. ISSN 1047-3203.

SINHAL, o. Blind image watermarking for localization and restoration of color images. **IEEE Access**, v. 8, p. 200157 – 200169, 2020.

SINHAL, R.; ANSARI, I. A.; AHN, C. W. Blind Image Watermarking for Localization and Restoration of Color Images. **IEEE Access**, v. 8, p. 57–69, 2020. ISSN 2169-3536.

SOWMYA, o. Discrete Wavelet Transform Based on Coextensive Distributive Computation on FPGA. **Materials Today: Proceedings**, v. 5, n. 4, Part 3, p. 60–66, 2018. ISSN 2214-7853.

SREENIVAS, K.; KAMAKSHIPRASAD, V. Improved image tamper localisation using chaotic maps and self-recovery. **Journal of Visual Communication and Image Representation**, v. 49, p. 164–176, 2017. ISSN 1047-3203.

STEPHANE, M. **A Wavelet Tour of Signal Processing**. [S.l.]: Elsevier, 2009. ISBN 978-0-12-374370-1.

STROGATZ, S. H. **Nonlinear Dynamics and Chaos**. 1. ed. [S.l.]: Westview Press, 2001. (Studies in Nonlinearity).

TAI, o. Image self-recovery with watermark self-embedding. **Signal Processing: Image Communication**, v. 65, p. 11–25, 2018.

TAN, o. A Robust Watermarking Scheme in YCbCr Color Space Based on Channel Coding. **IEEE Access**, v. 7, p. 25026 – 25036, 2019.

THAKKAR, o. A fast watermarking algorithm with enhanced security using compressive sensing and principle components and its performance analysis against a set of standard attacks. **Multimedia Tools and Applications**, v. 76, p. 191–219, 2017. ISSN 1573-7721.

YUAN, o. Gauss jordan elimination-based image tampering detection and self-recovery. **Signal Processing: Image Communication**, v. 90, p. 11–60, 2021. ISSN 0923-5965.