Efficient image tampering localization using semi-fragile watermarking and error control codes

Pascal Lefèvre, Philippe Carré, Caroline Fontaine, Philippe Gaborit, Jiwu Huang

Efficient Image Tampering Localization using Semi-Fragile Watermarking and Error Control Codes

Pascal Lefèvre[a,*], Philippe Carré[b], Caroline Fontaine[c], Philippe Gaborit[d], Jiwu Huang[a]

[a] *College of Information Engineering, Shenzhen University, Guangdong, Shenzhen, China*
[b] *XLIM Laboratory UMR CNRS 7252 University of Poitiers, France*
[c] *Specification and Verification Laboratory UMR CNRS 8643, Cachan, France*
[d] *XLIM Laboratory UMR CNRS 7252 University of Limoges, France*

Abstract

In this paper, we propose an image tampering localization algorithm using semi-fragile watermarking and Error-Locating codes in the DWT domain. By introducing different classes of codes, we show the benefit in terms of image tampering localization and complexity of using control code error localization as an authentication function. Indeed, we first experimentally show that error localization block codes is as precise as using classical error correcting codes (Reed-Solomon and BCH codes) to locate image tampering. However, their corresponding decoding algorithms complexity is at least quadratic which make them impractical for some real time applications. To solve this problem, we introduce a new class of codes called Error-Locating codes where error localization is reduced to a single syndrome computation performed with quasi-linear number of binary operations. We provide comparisons of image quality and tampering localization performances using error-detection, error-localization and error-correction approaches with different error control codes.

Keywords: Semi-fragile watermarking, error control codes, parity check matrix, tampering detection and localization, Error-Locating codes, real-time applications.

1. Introduction

Multimedia content authentication must keep up with the fast pace of technologies development. This research field has received a growing attention from

*Corresponding author. College of Information Engineering, Shenzhen University, Shenzhen, China.
Email addresses : plefevre@szu.edu.cn (Pascal Lefèvre), philippe.carre@univ-poitiers.fr (Philippe Carré), caroline.fontaine@lsv.fr (Caroline Fontaine), gaborit@unilim.fr (Philippe Gaborit), jwhuang@szu.edu.cn (Jiwu Huang)

the community to prevent and eliminate content abuses. In this article, we focus on the problem of digital image tampering detection. More precisely, we focus on image tampering localization rather than the problem of distinguishing maliciously from non-maliciously modified images [1].

Several solutions can be found in the literature, based on different research areas. For example, one can mention algorithms which are able to detect image forgery as inconsistencies in the natural properties of images (for instance see [2]). Other techniques propose to use *hashing* algorithms that are *robust* to some image processing: such a hashing technique is applied on an image and the obtained hashed value is then transmitted, as well as the image. The receiver uses the corresponding image hash value to locate the tampered regions in the received image. One of the most recent contributions in this field is [3]. A third approach is to embed a *(semi-)fragile* watermark in the image, so that this watermark will vanish if the image is tampered. Several techniques have been proposed in this direction [4, 5, 1]. In this paper, we focus on this approach, in the specific track of semi-fragile watermarking schemes using Error Correcting Codes (ECC).

Error-correcting codes represent a class of error control codes that is well known in data hiding. They have been used on the very early years of this research field to improve the robustness against image processing. Their objective is to protect the payload embedded in the host image. To satisfy image quality constraints, only a smaller subset of image coefficients are modified. For example, Baudry et al. discuss in [6] the coding strategies and robustness with BSC or AWGN watermarking channels. In [7], Lefèvre et al. completes the study of robust watermarking by applying rank metric codes in other error structures. However, the use of codes in semi-fragile watermarking slightly differs from robust watermarking. Indeed, the content to be protected is the host image and the payload can be randomly chosen from the image content as it is encoded as a breakable pattern in order to detect and locate an image tampering. Hence, every image pixel must contain some watermark information to protect the host image.

First contributions on fragile and semi-fragile watermarking schemes using ECC appeared around 2000. In [8], Lee et al. proposed to use Reed-Solomon parity check coefficients as a payload which is scrambled using a random sequence generated by their embedding key. The imperceptibility is ensured by an embedding in the lowest significant bits. Their scheme can locate and correct tampered areas. However, experiments are limited and the performance complexity is not addressed. Moreover, their scheme cannot resist common image processing such as JPEG compression. In 2002, Sun et al. [9] introduced a semi-fragile image authentication framework using error correcting codes. They also propose to embed associated parity check bits in DCT block-based invariant features to resist JPEG compression. Although Sun et al. discuss the use of Hamming codes, there are no experimental result on images.

He et al. [10] embeds their watermark using quantization on coefficients from a region-based shape descriptor called Angular Radial Transformation (ART) enabling robustness to several geometric distortions. ART coefficents and water-

mark payload are both encoded using a Hamming code which helps their scheme to survive common image processing. Their work is focused on the resistance to these attacks and do not provide a detailed study on the tampering detection and localization performances. Zhou et al. [11] proposed a block-based semi-fragile authentication algorithm in the DWT domain able to detect and locate tampered image regions. Their authentication mechanism relies on a signature extracted from the host image which is encoded using a small length BCH code. As in previously described works, their use of error correcting code helps to survive small distortions caused by common image processing such as compression and AWGN. Moreover, the experiments are limited to three images and do not include realistic tamperings. Chan et al. [12] details a method embedding BCH codewords into the least significant bits to locate and recover from the tamperings. Moreover, they use Torus automorphisms and bit rotations to scramble the watermark inside the image. In [13], Chan improved the previous work [12] by solving potential inacurate prediction of most significant bits.

In the meantime, Chang et al. showed in [14] a way to overcome the problems of bursts errors occuring in MSB pixels and Vector Quantization attack (described by Wong et Memon [15]). They described a fragile watermarking algorithm using a small length Hamming code and a chaotic map (similar to a random number generator function) able to locate tampered regions. Their scheme is not robust against image compression and other non malicious attacks.

The use of correcting codes in fragile and semi-fragile watermarking for tampering localization schemes has mostly been addressed between 2000 and 2010. Although there are more recent contributions in the literature on semi-fragile watermarking (they are described in [16]), we dedicate a particular attention on the description of papers related to the application of codes. The idea relies on the error correction ability by embedding parity check bits. However, involved correcting codes only are small length BCH and Reed-Solomon (RS) codes. Moreover, most of the experiments are limited (small image database, few image tampering situations, etc).

In this paper, we propose a semi-fragile algorithm based on quantization combined with error control codes in the DWT domain. We focus on their application in the image tampering localization problem. The main contributions in this paper are as follows:

1. We give a general view on the use of control codes for semi-fragile watermarking by introducing authentication variant functions SYN (error detection via syndrome), LOC (erroneous symbol localization) and COR (erroneous bit localization or correction).
2. We investigate the application of *Error-Locating* codes (EL codes) which is a novelty in the field of semi-fragile watermarking. These codes can locate erroneous subwords (i.e. tampered image regions) without correcting them. By using a particular parity check matrix, it is possible to locate errors with a lower number of binary operations compared to our baseline of codes (BCH and RS codes) where the number of binary operations is at least quadratic.

3. We propose a semi-fragile embedding with several authentication functions able to locate tampered image areas. First, we experimentally show that locating erroneous sub-blocks (LOC authentication variant) as a tampered region is as precise as locating erroneous bits (COR variant) but is faster. Secondly, we show that the proposed embedding outperforms one of the most recent DWT-based embedding from Qi et al. [1] described section 4.4.

The rest of the paper is organized as follows. We first error-locating codes for an application in semi-fragile watermarking in section 2. We introduce our semi-fragile watermarking method in section 3. Several experiment results are presented and analyzed in section 4.

2. Error-Locating codes

In our work, we define the term *error control codes* as a mathematical object with error processing properties. For example, an error correcting code is an error control code that corrects errors occured in a codeword transmitted over a noisy channel. The previous statement is also valid for error detecting codes where the presence of errors are detected only. In any case, we denote by *decoding algorithm* the algorithm to process errors and extract meaningful information from it. Since one key concept of this paper is to show the potential of error locating codes, we decide to borrow from [17] the idea that error control codes can be classified into three (non-mutually exclusive) classes for the sake of pedagogy.

In the data hiding field, the most well known is the *error-correcting* code class which aims at correcting every errors on an erroneous codeword. This class has the most powerful decoding abilities, but at a price of a greater complexity cost. Examples of codes are binary BCH and Reed-Solomon codes. They have been widely used in the literature of digital watermarking.

A second class, called *error-detecting* codes, is only able to detect errors. These codes have a lower cost but are sufficient in some applications such as feeback transmissions where the receiver can ask for the retransmission of an erroneous codeword. For instance, well known codes are Hamming codes. Error-detecting and error-correcting codes can decode up to some error threshold determined by their definition and parameters.

A third class is called *error-locating* (or EL) codes and stands between the two classes of codes previously described. In the literature, these block codes are claimed to be useful to optimize bandwidth usage in feedback communication systems. The idea relies on retransmitting erroneous blocks only instead of the whole codeword.

They are still unkown in data hiding and have the potential to improve digital watermarking tampering localization algorithms due to their error localization properties and low complexity. These codes were proposed in 1963 by Wolf et al. [17, 18, 19].

We note that there may be an inclusion relationship between these three classes. If a code can correct errors, it can locate and detect them as well. A

code may detect more errors than it can locate and it may locate more errors than it can correct. In other words, an error-correcting code can belong to the three previously described classes. By specifically considering a code into a particular category, one may be able to develop optimized strategies.

In the following of this paper, we transpose the concept described for the three error control code classes to semi-fragile watermarking and emphasize the effectiveness of error-localization for image tampering localization.

The idea of *encoding* is to add to the original message redundancy bits (or *parity check bits*) to obtain a *codeword*. Instead of sending a k-bit message $m = (m_1, \ldots, m_k) \in \mathbb{F}_2^k$ [1] over a noisy channel $\mathcal{N}$, a n-bit codeword $c = (c_1, \ldots, c_n) \in \mathbb{F}_2^n$ is sent. A function **Enc** encodes m into a codeword $c = \mathbf{Enc}(m)$ where **Enc** can be a generator matrix. Here, $k < n$.

The receiver obtains a word $y = c + e$ with e a binary vector representing potential errors for each component. Then, y is processed by a *decoding* function **Dec**.

In this paper, we are interested in erroneous subword localization as a mean to achieve a finer image tampering localization without having to use error-correcting codes which are known to have a high computational cost.

2.1. Application of two classical error-correcting code families

We first consider binary BCH codes which are well adapted to deal with random errors. The second family are the Reed-Solomon (RS) codes. They are *optimal* non-binary BCH codes (or MDS) meaning that they can correct up to $t = (n - k)/2$ errors. RS codes have optimal efficiency against *burst* errors. Although there are more efficient and more recent codes in the literature, the scope is limited to these two families of codes which are chosen as a baseline of codes representing the error structure they are efficient with.

In the context of tampering localization, they are interesting because errors may, most of the time, occur in compact shapes. For example, one can embed a 9-bit RS codeword symbol into a 3×3 block so that if it contains some errors, it can be marked as tampered.

However, there are some constraints in order to use these codes in a block-based embedding. The first disadvantage lies in the code length choice. If blocks have square shapes $\delta \times \delta$, the code length needs to be a square integer. In this paper, we consider binary BCH codes with length $n = 2^u - 1$. We have the restriction:

$$\delta = 2^{u/2}. \tag{1}$$

Meanwhile for a RS code $RS(n, k)$ over $\mathbb{F}_{2^s}$, we have:

$$\delta = \sqrt{ns}. \tag{2}$$

where n and s are much easier to find. RS codes are more flexible since they only impose an existence condition, $n < 2^s$, on the length n and the size of

[1] $\mathbb{F}_2$ is a finite field corresponding to the integer remainder classes of the Euclidean division by 2. Its two elements are represented by symbols 0 and 1.

the finite field $\mathbb{F}_{2^s}$. Another drawback is about the size of these parameters. Larger parameters imply a larger computational cost and can strongly decrease the decoding performances.

Several algorithms were proposed in the literature to decode RS codes. The Berlekamp-Welch algorithm with complexity $\mathcal{O}(n^2)$ is one of them. Another well known algorithm is the Berlekamp-Massey algorithm with the same complexity over $\mathbb{F}_{2^s}$. It uses the BCH cyclic structure of RS codes to lower the complexity. In practice, Berlekamp-Massey or Euclid's algorithm are the used for decoding.

But, eventhough the decoding performances are more than satisfactory for an application in fragile watermarking, it is possible to gain similar decoding performances while significantly lowering the computational cost by using a class of codes called *error-locating codes* (or EL codes). In the next subsection, we provide some details to understand and use these codes.

2.2. Application of Error-Locating codes

EL codes are chosen to represent a key concept in this paper. For an algorithm designed for image tampering localization, it is sufficient to use codes that locate errors instead of codes that correct errors. To achieve image tampering localization, the idea is to associate the spatial positions of coefficients with EL codeword bits. When an error is located in an EL codeword subblock, the corresponding spatial positions are marked as tampered. In the following, we propose to apply the work of Das et al. [20, 21, 22] in which bounds of code existence and product code constructions are studied.

An EL code $\mathcal{C}$ is a linear block code of length n' over $\mathbb{F}_q$. It consists of k information symbols and $r = n' - k$ parity symbols. A codeword $c \in \mathcal{C}$ is divided into m mutually exclusive sub-blocks, each sub-block containing $t = n/m$ digits. By definition, $\mathcal{C}$ can locate errors if and only if the number of corrupted sub-blocks is at most l, and if each sub-block is affected by e or less errors where $l \leq m$. The decoding outputs which blocks (or sub-words) contain errors.

In this work, we use binary digits, i.e. $q = 2$. EL codes are generally defined from their parity check matrix H.

These codes are interesting because of the parameter freedom. Using lower and upper EL codes bound theorems [22], one can determine the possible parameters of the form $t = t'^2$ and $m = m'^2$ so that codewords and block size are equal. Moreover, the decoding algorithm complexity can also be lowered from quadratic to quasi-linear by using a parity check matrix defined by blocks.

To do so, we are interested here in a modified version of the 13×16 parity check matrix of Example 3.5 proposed after Theorem 3.3 of [22], which is a parity check matrix for a $(16, 3)$ binary product code with $m = 4, t = 4, l = 2, e = 3$.

We have:

$$H_0 = \begin{pmatrix}
1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0
\end{pmatrix} \tag{3}$$

Notice that the subcode is a repetition code of length 4 and dimension 1, with a 3×4 parity check matrix H_r represented in red such as :

$$H_r = \begin{pmatrix} 1&0&0&1\\ 0&1&0&1\\ 0&0&1&1 \end{pmatrix} \tag{4}$$

This means that the parity matrix H_0 can be rewritten using H_r. More precisely, the first twelve rows represent the Kronecker product of the 4×4 identity matrix by H_r denoted by H_1. The last row of H_0 is denoted by L_{13}. For the sake of clarity, we give the expression of H_1:

$$H_1 = \begin{pmatrix}
1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1
\end{pmatrix} \tag{5}$$

The decoding is straighforward: once the syndrome $\sigma(y) = H_0 y$ of the received word y is computed, the positions of the tampered regions are given by non zero bits in $\sigma(y)$. Nevertheless, an error in the decoding process can happen when the subword e_r is equal to the repetition code codeword $(1, 1, 1, 1)$. Indeed, the syndrome of e_r is $H_r e_r^T = 0$ eventhough $e_r \neq 0$.

Thanks to row L_{13}, an error is detected if there is an odd number of erroneous subwords. Otherwise, the error remains undetected and is counted as a missed detection in the authentication algorithm. For sake of simplicity, we only use one parity row although it is possible to add more parity rows that will help detecting and localizing more errors.

The number of binary multiplications can be determined by looking at the syndrome $\sigma(y)$. In general, a matrix/vector multiplication is quadratic over $\mathbb{F}_2$ but H_0 is sparse with submatrix H_r in the diagonal. We have:

$$H_0 = \begin{pmatrix} H_r&0&0&0\\ 0&H_r&0&0\\ 0&0&H_r&0\\ 0&0&0&H_r\\ X&X&X&X \end{pmatrix} \tag{6}$$

so we have $\sigma(y) = (H_r + X)(U + V + W)$ where $X = (1, 0, 0, 0)$ and $y = (U, V, W)$ the word defined by block. Since H_r has size $t \times (t-1)$, there are $\mathcal{O}(t^2)$ operations. Moreover, in the case where the approximation approximation $t \simeq m$ holds, we obtain a number of operations that is quasi-linear in the binary length of the EL code and is given by $\mathcal{O}(t^2) = \mathcal{O}(mt)$. We refer to this as quasi-linear complexity. On the other hand, the number of operations to decode RS codes (at equal length and subwords size) is $\mathcal{O}(n^2 s^2)$ since a multiplication in $\mathbb{F}_{2^s}$ costs $\mathcal{O}(s^2)$ multiplications over $\mathbb{F}_2$. The binary complexity becomes $\mathcal{O}((ns)^2) = \mathcal{O}((mt)^2)$ which is higher compared to $\mathcal{O}(mt)$. Both codes are compared with the same block size and the same block length ($n = m, s = t$). More details about code parameters are discussed in section 2.3.

We will show later in the experiments that using the same construction as parity check matrix H_0 allows us to obtain more balanced tampering localization performances compared to the product repetition code parity check matrix obtained by removing L_{13} from H_0. For an EL code consisting of codewords of m subwords of length t, we generalize the parity check matrix construction (denoted as H_0 in our experiments) with a repetition code of length t and dimension $t-1$ with an $m \times m$ identity matrix. The last row of the corresponding parity check matrix H_0 is a row vector filled with the row vector $v = (1, 0, \ldots, 0) \in \mathbb{F}_2^t$ repeated m times. In the next section, we explain in detail the embedding strategy of codewords in a block-based embedding.

2.3. Embedding process of codewords

In order to use control codes in a block-based scheme, the correspondence between block size and codes parameters is presented in table 1. For block codes, we have (n, s) with n the number of symbols or subwords inside a codeword and s the binary size of a symbol or subword. In the case of block codes, we can choose a RS code $RS_{\mathbb{F}^{2^s}}(n, k)$ and an EL code of parameters (m, t) such that the code length is equal $n = m$ and $s = t$.

As for the image block shape, we chose to embed square blocks of size $\sqrt{ns} \times \sqrt{ns}$ for the sake of simplicity. Thus, one block represents the embedding area of a codeword with parameter (n, s). The condition on n and s is they need to be square integers.

Table 1: Code parameters and corresponding image block sizes. Since the content of the watermark has no use in the context of semi-fragile watermarking, we choose $k = 1$ to obtain the maximum number of symbol errors allowed.

n	s	$\sqrt{ns} \times \sqrt{ns}$	$\lfloor (n-1)/2 \rfloor$
4	4	4×4	1
16	4	8×8	7
4	16	8×8	1
16	9	12×12	7
9	16	12×12	4

In an image, on can embed codewords by choosing a specific order block selection. A common choice is the raster scan order: blocks are selected from left to right in a row and from the top row to the bottom row. For codeword symbols, the process is the same. Since s is a square integer, subwords are embedded in a $\sqrt{s} \times \sqrt{s}$ sub block also in the raster scan order.

Table 2: BCH code parameters with correponding block sizes.

block size	BCH parameters
16×16	[255, 9, 63]
8×8	[63, 7, 15]

In the case of binary codes such as BCH codes, it is possible to simulate the same type of embedding by embedding one codeword bit into one pixel and fill the missing pixels with zeros. This implies a BCH code must have a code length lower and as close as possible to the block size ns.

As a summary of this section, we introduced the required knowledge to use error control codes in semi-fragile watermarking. After a description of the decoding concept and the three classes of codes, we quickly described the properties of classical codes and introduced EL codes in the data hiding field for the first time as an application for tampering localization using watermarking. In the next section, we detail the proposed embedding method and the different authentication variants.

3. A new method to locate image tampering using control codes

Our scheme is fully blind which is a desirable watermark property in practical scenarios. In order to authenticate and localize tamperings, the receiver does not require any extra information, but only the secret key and public parameters. Hence, the security of the secret watermarking key generation only relies on the choice of a secret seed (for a Mersenne twister pseudo-random generator for example) and is independent from the host image and the watermark content (chosen by the user). Moreover, this secret seed can also be used to scramble the watermark (using Arnold's cat map, for instance).

As for the transform domain, we chose the DWT transform domain applied on the whole image. By choosing this transform, we benefit from several advantages. The transform to spatial domain correspondence for tampering map translation leads to satisfying results and we avoid performing a block based embedding which can produce visible artifacts at block boundary regions in the case of strong embedding distortions. Moreover, DWT transform brings interesting watermarking robustness properties against common image processing such as JPEG compression and filtering which is also true for the DCT transform. However, block DCT transform is well known in the literature of

signal and image processing for producing visible artifacts at block boundaries. In the context of JPEG compressed images, several solutions were proposed to address this problem such as [23, 24]. This problem also exists for images that are watermarked using a block DCT embedding such as in [25, 26].

Using error control codes, our scheme is able to create a tampering map $\mathcal{T}'$ which is an estimation of the binary map $\mathcal{T}$ representing the original tampered regions. Several wavelet families were considered such as Debauchies and Haar. As described later in section 3.4.3, the tampering map is obtained in the DWT domain and needs to be translated into the spatial domain. The wavelet family that leads to the best reconstruction is the Haar wavelet family.

Experimentally, the Haar wavelet family provides the best localization performances. Our method divides a chosen wavelet sub-band into non-overlapping blocks of size $\delta \times \delta$ (examples are given in table 1). δ is chosen in function of the authentication variant and the error control code described later in this section.

In the embedding strategy, we propose to embed both information and parity bits in order to enable flexibility in the code dimension k. Moreover, only codewords error localization properties are required so the value and the length of information bits do not matter, i.e., codewords can be randomly chosen.

However, the embedded watermark becomes more predictible. In order to prevent any unauthorized party to modify or delete the fragile mark, one can add a random noisy sequence to every codeword. The concatenation of these noisy vectors can be secretely generated using the watermarking private key k. Before we describe the embedding, extraction and authentication process, we propose to quickly describe the concept of digital watermarking based on quantization.

3.1. Digital Watermarking based on Quantization

In this work, we propose to use the well known Quantization Index Modulation (QIM) by Chen et al. [27], further studied in [28]. There are several improvements to the QIM method in the literature such as DC-QIM [27] and DM-QIM [29] but we choose the original QIM method for the sake of pedagogy since this paper focuses on the application of codes. The QIM method is well known for its robustness in watermarking applications and motivates us to choose it over the spread spectrum method [30]. We note that we choose the binary QIM in order to be able to compare binary BCH, RS and EL codes in our experiments. Nevertheless, one can still choose a multiple symbol QIM if a block code (such as a RS code). In the following, we describe the binary QIM method.

To embed one bit of information m in a host sample x of dimension L, we use a quantizer Q_m defined by:

$$y = Q_m(x, \Delta) = \left\lfloor \frac{x}{\Delta} \right\rfloor \Delta + (-1)^{m+1} \frac{\Delta}{4}$$

with Δ the quantization step and y the modified (or watermarked) host sample. On the receiver side, detection step processes the received vector z of dimension

L, and an estimation of the original message $\hat{m}$ is computed:

$$\hat{m} = \underset{m \in \{0,1\}}{\arg\min}\, dist(z, \Lambda_m)$$

with

$$dist(z, \Lambda) = \min_{y \in \Lambda} \|z - y\|_2$$

and

$$\begin{cases} \Lambda_0 = \Delta\mathbb{Z}^L - \frac{\Delta}{4} \\ \Lambda_1 = \Delta\mathbb{Z}^L + \frac{\Delta}{4} \end{cases}$$

A larger quantization step Δ implies a larger distortion on the vector, i.e. a lower image quality. Reciprocally, a smaller Δ allows one to embed a watermark while preserving a better iamge quality. In the later case, the watermark is less robust to image processing. In practice, one needs to find a tradeoff adapted to the watermarking real world application.

3.2. Watermark embedding

We denote by $\mathcal{I}$ the host image, and ℓ the wavelet decomposition level. We empirically determined that the best performance tradeoff is obtained with $\ell = 2$ and HH subband. The embedding algorithm is decomposed into the following steps:

1. Apply a level ℓ decomposition on host image $\mathcal{I}$
2. Extract sub band HH (of size $h_0 \times w_0$) and divide it into non overlapping blocks of size $\delta \times \delta$
3. For each block, generate a 2D random binary blocks of length $\delta \times \delta$ using a secret key k as a pseudo-random number seed
4. Generate a random binary codeword $c_{u,v}$ of size δ^2
5. For each HH sub band block $B_{u,v}$, quantize ($L = 1$) every coefficient where the binary sequence is $c_{u,v} + n_{u,v}$
6. Recompose the image with the modified coefficients to obtain the watermarked image $\mathcal{I}'$

Secret key k allows one to secure the watermark by scrambling the embedding sequence (random codewords). In order to extract the watermark, the receiver must know k. Block size $\delta \times \delta$ can be chosen using tables 1 and 2. Note that the embedding is done in the sub band HH only at level ℓ.

3.3. Watermark extraction

The receiver then processes the watermarked image $\mathcal{I}'$ that has potentially been modified. We have:

1. Apply a level ℓ decomposition on the received image $\mathcal{I}''$
2. Extract sub band HH (of size $h_0 \times w_0$) and divide it into non overlapping blocks of size $\delta \times \delta$

3. For each received block $B'_{u,v}$, extract the associated received codeword $y_{u,v}$ of size δ^2 using the secret key k using the quantization detector with step Δ.

The extraction algorithm is limited to the extraction of codewords in a binary form from each quantized DWT blocks.

3.4. Image authentication

The authentication process of image $\mathcal{I}''$ aims at estimating a tampering map $\mathcal{T}'$ that is very close to the original tampering map $\mathcal{T}$ which is the binary difference between the image before and after the tampering.

In this paper, we focus on image tampering localization. One will interpret the nature of the tampering map content after $\mathcal{T}'$ is computed. For example, the receiver can decide if the image has been modified by a malicious user or conclude that the image has been damaged by an innocuous image processing. A third and rather edge case is to conclude that the image has suffered from weak distortions, malicious or not and can thus be ignored.

3.4.1. Error detection via syndrome (SYN)

The class of codes known as *error-detection* consists in returning the number of errors in a codeword without locating and correcting them. Moreover, by simply using a classic error-correcting code, we can setup a simple algorithm to detect if a received codeword has been modified.

The syndrome computation of a word y is a faster operation compared to the next two authentication variants we describe later. In the case of an error-correcting code $\mathcal{C}$, it is enough in our context to detect the presence of at least one error no matter how many errors there are in a given codeword. The algorithm we implement for this authentication variant is called *error-detection via syndrome* denoted as SYN.

For a received word y, there is a codeword $c \in \mathcal{C}$ such that $y = c + e$. If there is an error, $e \neq 0$ which can be verified by the following equation:

$$c \in \mathcal{C} \iff \sigma(c) = Hc^T = 0 \tag{7}$$

with H the associated parity check matrix of $\mathcal{C}$.

In the authentication step, we compute the syndrome $\sigma_{u,v}$ of every received codewords $y_{u,v}$ that were extracted from their corresponding sub band blocks $B'_{u,v}$.

If $\sigma_{u,v} = 0$, the sub band tampering map $\mathcal{T}'_{HH}$ is marked with grayscale value 0 (for black as a non tampered region) at the location of block $B'_{u,v}$. Else, $\sigma_{u,v} \neq 0$ and $\mathcal{T}'_{HH}$ is marked with grayscale value 255 or 1 for white as a tampered region.

3.4.2. Erroneous symbol/subword localization (LOC)

We discuss two ideas about the second authentication variant. The first one is developing the SYN variant. The decoding algorithm of a codeword $c \in RS_{n,k}$ over $\mathbb{F}_q$ can be further developed. Instead of only detecting the existence of an error, one may want to know more by locating which symbols are erroneous.

We can obtain a much finer tampering localization algorithm by modifying the previous one described above. Instead of marking the whole block $B'_{u,v}$ as tampered, we mark as tampered the spatial location of the erroneous codeword symbols.

More precisely, we have $c = (c_1, \ldots, c_n), c_i \in \mathbb{F}_q$ with $q = 2^\alpha, \alpha \geq 2$. In block $B'_{u,v}$, the spatial locations of modified c_i's will be marked as tampered.

Hence, the tampered regions are described more presicely but at the cost of a higher complexity and a lower computational speed. However, the computational cost is in practice lower than trying to completely correcting all the errors since the remaining linear systems over $\mathbb{F}_q$ are not executed.

The second idea is about using EL codes. As we saw in the last section, these codes provide a solution to tampering localization at a lesser complexity (and a faster time computation) compared to conventional error-correcting codes such as RS and BCH codes. As we saw in the previous section, a well chosen parity check matrix of an EL codes allows one to simply read the syndrome to locate the errors.

3.4.3. Erroneous bit localization or correction (COR)

The third authentication variant is the most precise authentication method. By continuing our demonstration with the previous RS code, we can obtain the finest tampering localization performances by correcting erroneous symbols (but with a greater computation cost). Then, the tampering localization can be done bit by bit instead of symbol by symbol.

Independently from the authentication variant being used, the output is a sub band tampering map $\mathcal{T}'_{HH}$. However, the quality of $\mathcal{T}'_{HH}$ is not satisfying and can be further refined using a sliding window method such as in Qi et al.'s method. The result of this operation is a tampering map with better localization results defined. For each pixel of $\mathcal{T}'_{HH}$, we do the following:

1. Count the number t_α of tampering pixels in $\mathcal{T}'_{HH}$ in a $\nu \times \nu$ pixel window
2. If $t_\alpha > t_0$, with t_0 a threshold integer, mark the pixel at the center of the $\nu \times \nu$ window as tampered in the new map (image borders are padded with pixels considered as not tampered)

Sliding windows with $\nu = 5, 7$ and $t_\alpha > 3$ were tested but did not lead the best localization results. The best parameters we found are $\nu = 3$ and $t_\alpha = 3$.

4. Experiments

In all our experiments, we use an image dataset from [31] called *Realistic Tampering Dataset*. This dataset gathers image pairs (original and realistically

tampered) with different cameras. These images were collected in a natural environment and were tampered in such a way the human eye cannot tell if the image was tampered. Eventhough, by definition, the proposed method does not distinguish between realistic and non realistic tamperings, this image dataset provides more credibility to our work for a real world application. On the opposite, the majority of past contributions in this field only have tested their algorithms with a small number of images with simple tampering situations. Moreover, choosing this dataset also helps on defining more precisely the context or possibility the nature of errors produced by the tampering operations.

Our measures are computed using the 55 images of the Canon 60D camera. In our experiments, the tampering is applied on the watermarked image by adding the image difference between the pristine image as a host image and the tampered one.

As for the authentication variant parameters, we chose block sizes as square integers for the sake of simplicity. We choose (n, s) for convenience with n and s always being square integers. Let us remind that n is the number of symbols inside a codeword with s the binary size of a symbol. One also must make the difference between (n, s) and the block size inside the image which is $\sqrt{ns} \times \sqrt{ns}$. For example, $(n, s) = (16, 16)$ is represented as a 16×16 image block. Also, $(n, s) = (16, 4)$ is represented as a 8×8 image block.

In the next subsection, we introduce some metrics to evaluate the tampering localization performances of every authentication methods.

4.1. Metrics

In this context, the choice of metrics medium used to evaluate tampering localization performances are not standardized in the literature of watermarking. However, it is very common to compute the false alarm (FA) and missed detection (MD) rates. In addition, researchers in the image tampering localization research field are also used to use metrics such as F_1 and Precision. As a consequence, we propose to use these four metrics on the original tampering map $\mathcal{T}$ and the estimated tampering map $\mathcal{T}'$.

Let us note that F_1 score can be defined as the harmonic mean of precision and recall metrics such as:

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{8}$$

Moreover, precision metric is only considering true positives and false alarms and is defined such as:

$$\text{precision} = \frac{TP}{TP + FP} \tag{9}$$

With FP and TP the number of false positives and true positives respectively. For recall, it is the opposite over 1 of the corresponding missed detection rate:

$$\text{recall} = 1 - \text{missed detection rate} \tag{10}$$

with :

$$\text{missed detection rate} = \frac{FN}{FN + TP} \tag{11}$$

With FN the number of false negatives.

As a summary, F_1 score can be seen as an overall performance measure but doesn't allow one to have a closer observation at the underlying tradeoff in terms of false alarms and missed detections for example. On the other hand, we believe it is also essential to be able to visually evaluate performances in a efficient way. Hence, we introduce a color map we called as *confusion map* combining the information of $\mathcal{T}$ and $\mathcal{T}'$ in the same image using the corresponding 2×2 *confusion matrix* containing TP, FP, FN and TN.

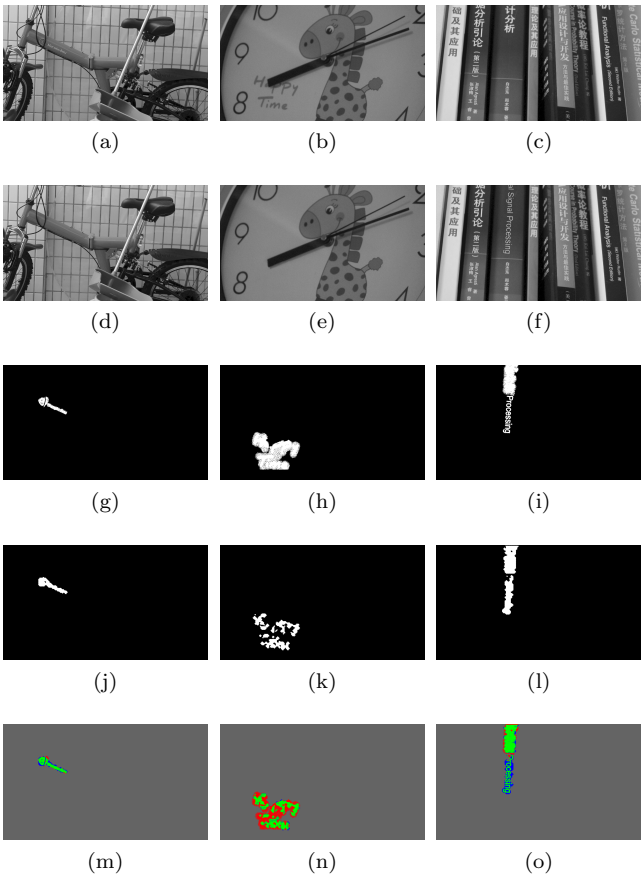


Figure 1: From top to bottom: examples of watermarked, tampered images, original tampering map $\mathcal{T}$, estimated tampering map $\mathcal{T}'$ and confusion maps. From left to right: DPP0012, DPP0022, DPP0027.

The associated colors defines the confusion map color coding. An example

of confusion map is given in figure 1. One can visualize the original tampering map $\mathcal{T}$ by only considering regions in green and red (TP and FN). For the estimated tampering map $\mathcal{T}'$, the corresponding colors are green and blue (TP and FP). Image regions in grey correspond to true negatives (TN).

We have now provided the necessary explanations for our experiments and propose in the next subsection to show our experimental results.

4.2. Image quality evaluation

Image quality can be controlled at the embedding step by adjusting the quantization step Δ in the DWT domain. In figure 2, we show several curves corresponding to different wavelet decomposition level in the HH sub band. They represent the evolution of watermarked image PSNR in function of Δ.

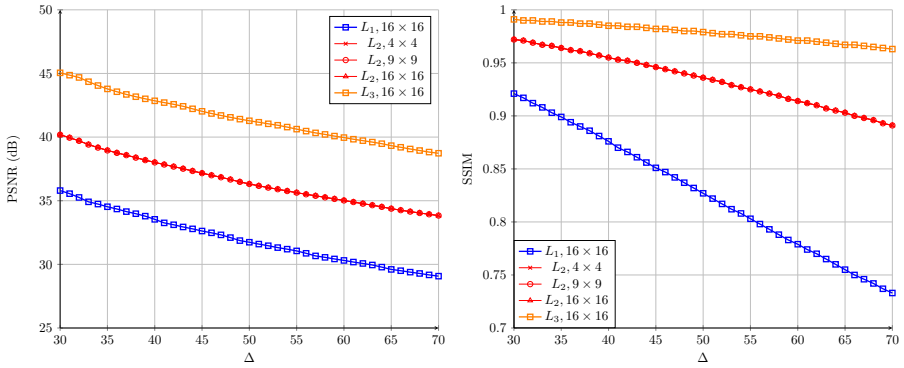


Figure 2: Evolution of PSNR and SSIM in function of Δ for several Haar wavelet decomposition levels (L_i for a decomposition level of i) in the HH sub band. For each measure, the standard deviation is less than 0.6 and 0.1 for PSNR and SSIM measures respectively. Each legend entry is of the form $L_i, n \times s$, with n the number of sub blocks and s the size of one sub block.

We can see that L_2 curves are overlaping in figure 2, PSNR measures are the same for every Δ for different block sizes because the whole sub band is quantized. Even if some rows and columns are ignored by the quantization due to the sub band size not being an integer multiple of the block size, the PSNR variation can be neglected.

However, PSNR values are clearly changing with the wavelet decomposition level. Between curves L_1 and L_2, the PSNR difference is around 4dB and we have around 5dB difference for curves L_2 and L_3. Hence, embedding in a higher wavelet decomposition level sub band allows the proposed method to have a better image quality and reciprocally. In practice, embedding with a higher decomposition level results in a more difficult tampering localization. We experimentally determined that using the Haar wavelet family produces better tampering localization maps compared to Debauchies wavelets.

For SSIM measures, we can see that they remain higher than 0.73 for every curves. A higher quantization step implies a lower PSNR as explained in the

paper. SSIM measures also decrease when Delta increases. It can be explained by the fact that the watermark can be considered as an "uniform-like" or "well spread" noise in all parts of the host image.

An ideal image quality which is visually acceptable can be achieved with PSNR = 40dB. For curve L_1, it is not possible to obtain such image quality. For curves L_2, we achieve this PSNR value with $\Delta = 30$ and SSIM = 0.975. In the same way, curves L_3 have corresponding values of $\Delta = 60$ and SSIM = 0.975. The image quality must be adjusted in order to optimize a tradeoff by also considering tampering localization and robustness performances. In the next two subsections, we detail the previously mentioned experiments.

4.3. Tampering localization performance evaluation

In this subsection, we evaluate the performances of our semi-fragile algorithm and its variants with different error control codes. First, we propose to illustrate the impact of the different authentication variants on the tampering localization performances. In section 3, we detailed three ways (SYN, LOC and COR) to implement an authentication algorithm. Each one of them has different tradeoff in terms of tampering localization and complexity. In table 3, those three authentication variants are evaluated using Reed-Solomon codes and different block sizes with F_1 score, precision, false alarm rate and missed detection rate.

Globally, we can see that F_1 is the best when the block size is smaller (4×4 compared to 9×9 and 16×16) for RS-SYN, RS-LOC and RS-COR respectively. For every block size, F_1 score of RS-LOC are generally closer to RS-COR than RS-SYN. This observation is clearer to see by looking at the precision measures. For example with block size $(9, 9)$, a precision of 0.754 for RS-LOC is closer to RS-COR precision (0.7797) than RS-SYN precision (0.5881) with standard deviations $\sigma \leq 0.17$. Moreover, the same observation can be done for FA rates.

For FA and MD rates, when the block size is increasing, rates are respectively increasing and decreasing except for RS-COR where MD rates are increasing. Hence, we have a natural tradeoff for RS-SYN and RS-LOC for these metrics. On one side, tampering distortions are fooling the quantization so the MD rate is naturally increasing. On the other side, since the block size is increasing, the probability to mark a untampered pixel as tampered is increasing (i.e., the FA rate is increasing) but it also helps the MD rate to "artificially" decrease (we can see the result as a "lucky guess"). This can be seen as a numerical optimization but deciding which of FA or MD rates is more important to lower is an important issue to be resolved by the scheme user.

As for the exception of RS-COR MD rates increasing, it can be explained by the nature of decoding involved in the authentication process. Even though the block size is $\sqrt{ns} \times \sqrt{ns}$, the decoding is done bit by bit independently of the block size. In table 4, the same behavior occurs for BCH-COR variant MD rates.

A final remark of the first part of this experiment, we see that RS-LOC performances are much closer to RS-COR than RS-SYN with the advantage of

Table 3: Tampering localization performances of variants RS-SYN, RS-LOC and RS-COR based on RS codes. $\Delta = 30$*, PSNR* $= 40.2dB$*.*

(n, s)	F1 score	Precision	FA rate	MD rate
RS-SYN				
$(4, 4)$	0.7945	0.7597	0.0093	0.1325
σ	0.1202	0.1264	0.0045	0.1652
$(9, 9)$	0.6994	0.5881	0.0237	0.0801
σ	0.1389	0.168	0.0103	0.1159
$(16, 16)$	0.5919	0.4566	0.0450	0.0621
σ	0.1612	0.1791	0.0209	0.1030
RS-LOC				
$(4, 4)$	0.7999	0.8227	0.0058	0.1868
σ	0.1392	0.1038	0.0031	0.1941
$(9, 9)$	0.7853	0.754	0.0098	0.1450
σ	0.1183	0.1232	0.0048	0.1667
$(16, 16)$	0.7448	0.6683	0.0162	0.1160
σ	0.1144	0.1383	0.0082	0.1495
RS-COR				
$(4, 4)$	0.7906	0.8249	0.0056	0.2047
σ	0.1464	0.1032	0.0030	0.2012
$(9, 9)$	0.7524	0.7797	0.0076	0.2319
σ	0.1559	0.1162	0.0041	0.2049
$(16, 16)$	0.7059	0.6879	0.0123	0.2439
σ	0.1467	0.1378	0.0075	0.1941

having a decreasing MD rate when the block size is increasing. This is why F_1 scores obtained with RS-LOC are a little better than RS-COR. Moreover, RS-LOC has a lower complexity because the RS decoding algorithm is limited to localizing erroneous symbols inside a codeword (block of ns pixels) as we saw in section 2. Its implementation is hence easier and faster than the complete RS error correction algorithm. In conclusion, RS-LOC is the best authentication variant by achieving better tampering localization performances and having a lower complexity.

As a second part of this experiment, we propose to show that it is possible to further decrease the complexity of the authentication process by using the previously introduced EL codes. The corresponding variant is denoted as EL-LOC and uses the parity check matrix H shown in section 2. In table 4, we show measures using the same metrics as in the previous experiment.

Compared to RS variants, EL-LOC globally obtains better results. For block size $(4, 4)$, EL-LOC F_1 score and precision are the highest and false alarm rates is

Table 4: Performances comparisons of variants EL-LOC, REP-LOC and BCH-COR with different parameters (n, s). $\Delta = 30$ and $PSNR = 40.2dB$.

(n, s)	F1 score	Prec.	FA rate	MD rate
EL-LOC				
$(4, 4)$	0.8021	0.8301	0.0054	0.1911
σ	0.1364	0.0986	0.0029	0.1902
$(9, 9)$	0.7568	0.6989	0.0128	0.1330
σ	0.1245	0.1492	0.0057	0.1518
$(16, 16)$	0.6897	0.5918	0.0231	0.1116
σ	0.1467	0.1739	0.0121	0.1505
REP-LOC				
$(4, 4)$	0.7935	0.8836	0.0031	0.2480
σ	0.1551	0.0741	0.0017	0.2043
$(9, 9)$	0.8079	0.8127	0.0063	0.1664
σ	0.1236	0.1088	0.0031	0.1746
$(16, 16)$	0.7902	0.7589	0.0093	0.1406
σ	0.1170	0.1267	0.0045	0.1633
BCH-COR				
$(16, 4)$	0.7019	0.85	0.0036	0.3529
σ	0.2166	0.1058	0.0022	0.2469
$(16, 16)$	0.6519	0.8732	0.0044	0.4182
σ	0.2271	0.0846	0.0052	0.2618

the lowest which is a convincing result even though its MD rate is slightly higher than RS-LOC MD rate. As the block size increases, these measures decrease and become lower than RS-LOC and RS-COR measures. Moreover, the false alarm rate is higher. For MD rates, EL-LOC and RS-LOC have similar measures. We easily conclude that the variant EL-LOC achieves nearly better average measures but still similar (because of the standard deviation) performances than RS-LOC with even lower complexity since the decoding algorithm doesn't involve any equation system to solve and only consists in reading the syndrome with parity check matrix H.

In addition to this result, figure 4 also shows performance results about product repetition codes with parity check matrix H_1 and BCH codes through variants REP-LOC and BCH-COR respectively. F_1 and precision measures of REP-LOC are higher than EL-LOC with lower false alarm rates but higher missed detection rates for a given block size.

For BCH-COR, we only propose measures for block size 16×16 and 8×8 ($(16, 16)$ and $(16, 4)$ respectively). The corresponding BCH code parameters can be consulted in table 2. Since the decoding is done bit by bit, i.e., pixel by

pixel inside one block, BCH-COR performances should be similar to RS-COR (table 3) but that is not the case. One explanation requires us to assume that errors from the tampering are random and hence BCH codes are optimal to handle this type of errors.

In this subsection, we evaluated the tampering localization performances of the proposed authentication variants. These variants are using an embedding based on the quantization of DWT coefficients. We showed that LOC variant equiped with an EL code obtained the best tradeoff in terms of tampering localization performances and complexity. If one requires the highest precision performances, a smaller block size is recommended.

4.4. Comparisons with a state of the art method

We compare our method performances with Qi et al's method [1]. To our knowledge, no recent work on semi-fragile algorithms using codes were found. One common point for comparison is the use of the DWT domain. We remind that the goal of this work is to present a proof of concepts for the application of codes in digital watermarking. This method comparison serves as a reference point to the reader.

Compared to our method, Qi et al.'s method is built differently and has a different goal. Their embedding strategy is based on SVD performed on 4×4 DWT coefficients. The authors describe a watermark contained a content-independent mark and a content-dependent mark. By manipulating the relationships between singular values in JPEG quantized sub-blocks, a block pattern that represents the content-dependent part of the mark is created. After an image tampering, relationships between singular values may break. Qi et al. propose to use five authentication measures and a three-level authentication process to localize tampered areas and classify the nature of the tampering. Three types of modifications are analyzed: intentional tampering, incidental modifications and mild to severe content-preserving modifications. We note that these metrics are independent from the watermarking scheme they proposed. By following their analysis on the five authentication measures, we can also apply them to our work.

In the work we propose, we call malicious image manipulation strong changes in the content such as content removal and addition. These changes preserve the image quality and conceal their existence through high quality hand-crafted forgeries. Then, we consider non malicious (or incidental) attacks as common image processing such as JPEG compression, filtering and noising. When the tampering map is extracted, only the maliciously tampered regions are meant to appear. In this context, the ideal goal is to allow one to accurately segment the modified areas of the image. To perform a fair comparison, we choose to compute the previously used metrics on Qi et al.'s tampering maps after using the same filtering strategy used in our scheme (see section 3.4.3), i.e., a new error map is computed by assigning a pixel as tampered if at least three of its eight neighbor pixels are tampered.

We first measured the image quality of QI-SVD and obtained a PSNR of 39.8dB which is similar to the proposed method PSNR measures previously showed in figure 2 for L_2 curves at $\Delta = 30$ or L_3 curves at $\Delta = 60$.

On the tampering localization performances , QI-SVD precision measures (showed in table 5) is the highest compared to every other measures in table 4 with a very low false alarm rate. However, QI-SVD's missed detection rate is very high (0.5425) meaning that only half of the tampered pixels are correctly detected. As for its F_1 score (0.5919), it is similar to the lowest F_1 measure (0.5919) obtained by RS-SYN $(16, 16)$. Moreover, QI-SVD's measures of standard deviation are the lowest which makes this method very stable in terms of performances.

In table 5, we compare the QI-SVD with EL-LOC performances. EL-LOC variant is a more balanced authentication method compared to QI-SVD at an equal block size of 4×4 with a higher F_1 score and precision and a lower missed detection rate.

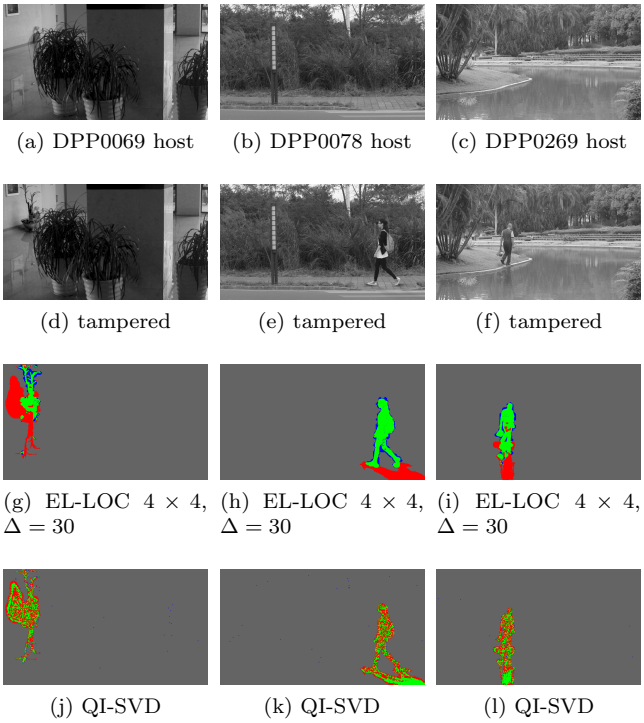
Table 5: Performances comparisons of variant EL-LOC $(4, 4)$*, and method QI-SVD.* $\Delta = 30$ *and PSNR* $= 40.2dB$*.*

(n, s)	F1 score	Prec.	FA rate	MD rate
EL-LOC				
$(4, 4)$	0.8021	0.8301	0.0054	0.1911
σ	0.1364	0.0986	0.0029	0.1902
QI-SVD				
-	0.6088	0.9177	0.0013	0.5425
σ	0.0394	0.0788	0.001	0.0352

In conclusion, we have proposed a semi-fragile embedding using quantization in the DWT domain using error-control codes. We first showed than LOC variant allows to obtain tampering localization peformances at least as good as COR variant with lower complexity leading to easier implementation and potentially faster computations. Moreover, we propose to use EL codes instead of RS codes to further decrease the computational complexity. This semi-fragile embedding using codes allow us to obtain better performances than a state of the art method [1].

However, one limitation of the proposed method is its inability to detect shadow modifications such as adding a light shadow or modifying a shadow. For example, a person or an object is added in the foreground of the host image. In order to be realistic with the image luminance, the corresponding person's shadow must be added. This type of modifications only introduces small grayscale distortions that are related to the image content. If the quantization step Δ is high enough, image regions corresponding to shadow will be part of

the missed detections represented by the red regions in figure 3. In the case of QI-SVD, we can see that this phenomenon does not happen. Instead, missed detections are mixed with the true positives in green in the entire tampered regions.



(a) DPP0069 host (b) DPP0078 host (c) DPP0269 host

(d) tampered (e) tampered (f) tampered

(g) EL-LOC 4×4, $\Delta = 30$ (h) EL-LOC 4×4, $\Delta = 30$ (i) EL-LOC 4×4, $\Delta = 30$

(j) QI-SVD (k) QI-SVD (l) QI-SVD

Figure 3: Tampering localization performances in the case of low grayscale content related modifications (eg: shadows). Results obtained with variant EL-LOC and method QI-SVD.

We propose now to study the tampering localization performances under high constraints such as JPEG compression and Additive White Gaussian Noise.

4.5. Robustness evaluation

It is realistic to expect a fragile embedding to be robust to some image processing and embedding a watermark in DWT domain can achieve such requirement. In this subsection, we evaluate the tampering localization performances against two of the most common unintentional image processing (JPEG compression and additive white gaussian noise abbreviated AWGN). This evaluation is not meant to be exhaustive but rather to illustrate a robustness evaluation of the tampering localization performances and focus on the analysis of EL codes. We must note that the experiment in this section does not evaluate the robustness of a watermark payload (typically evaluated by observing binary error

rates) but rather the tampering localization performances using EL codes under image processing attacks.

In this experiment, we embed our watermark in the entire diagonal subband HH at the third level resolution. Several quantization steps are used ($\Delta = 30, 45, 60$) and corresponding PSNR can be consulted in figure 2. The authentication variant being used is EL-LOC. Experimentally, we provide a large range of parameters for both attacks. Compared to EL-LOC, results are similar for other variants.

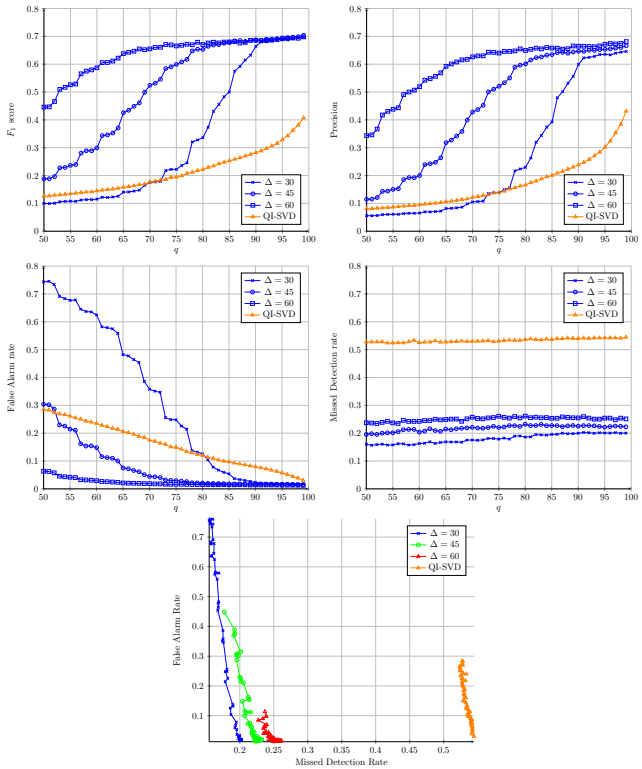


Figure 4: Tampering localization performances evolution in function of the JPEG quality factor q for variant EL-LOC at a level 3 wavelet decomposition.

For JPEG compression, the best robustness results were obtained with a block size of 4×4 in figure 4. For different values of Δ, curve values are increasing from nearly 0 to a maximum of 0.70 for F_1 score and precision, decreasing from 0.78 to 0.012 for false alarm rates and increasing from 0.15 to 0.25 for missed detection rates. Note that tampering maps produced in this experiments are visually satisfying when F_1 scores are close to 0.7. Values for which F_1 scores are reaching 0.69 are high quality factors only ($q = 90, 80, 75$ with $\Delta = 30, 45, 60$ respectively). We have the same remark for precision measures with $q = 95, 85, 75$. False alarm rates are also reaching 0.1 for $q = 90, 80, 65$

respectively. By increasing the quantization step Δ, performances are better for every metrics except for missed detection rates which slightly increase when Δ increases. Also, one could embed the watermark in a higher level of decomposition to obtain a better robustness. However, experiments on that matter showed that using a higher decomposition level gives better robustness to the quantization based method but localization results obtained from the DWT to spatial tampering map translation is worse.

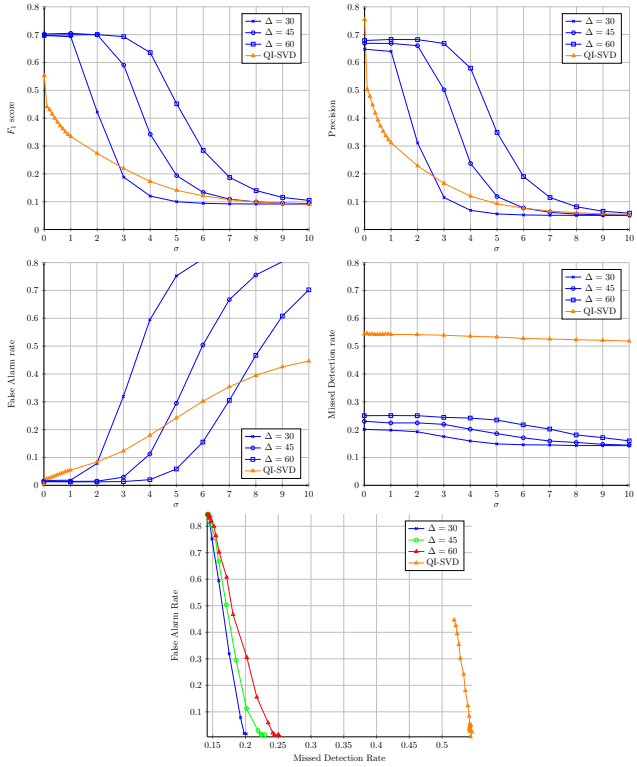


Figure 5: Tampering localization performances evolution in function of the AWGN standard deviation σ for variant EL-LOC at a level 3 wavelet decomposition.

As for the experiments on AWGN, we obtain values around 0.7 for F_1 score and precision for small values of σ (figure 5). For F_1 metric, measures can be maintained at 0.7 with respect to respective $\Delta = 30, 45, 60$ when $\sigma = 1, 2, 3$ and get below 0.6 for $\sigma = 1.5, 3, 4$. We can observe the same behavior for precision measures. For false alarm rates, values stay minimal for $\sigma = 1, 2, 3$ and become higher than 0.1 $\sigma = 2, 4, 5.5$. Missed detection rates are stable and vary between 0.15 and 0.25.

An explanation of the obtained performances is the ability to classify error types. In the dataset [31] we use, realistic tampering are characterized by compact modified regions and can be created by different type of tamperings.

More importantly, the modified grayscale values are, most of the time, unrelated to the original ones. Some examples are copy-pasting other regions of the image (copy-move), copy pasting contents from other images and partial content erasure. In this dataset, tampering are realistic, i.e., invisible to the naked eye. Therefore, by applying a filter-like operation which is the window sliding method described in section 3, it is possible to differentiate realistic tamperings and noisy error structure types induced by JPEG compression and AWGN.

In figure 6, we propose to visually appreciate the robustness of EL-LOC with $\Delta = 60$. We chose a quality factor $q = 85$ and a standard deviation $\sigma = 2$ for which performances are satisfying. For EL-LOC, some artifacts (false alarms in blue) produced by a compression for example sometimes appear in the confusion map.

Finally, we also added the tampering localization performances under JPEG compression and AWGN for method QI-SVD in figures 4 and 5. In the case of JPEG compression, we can see that, even at high quality factor $q \geq 90$, QI-SVD's performances are below 0.3 for F_1 score and precision whereas EL-LOC curves are at 0.7. Its false alarm rate is low but is much higher than EL-LOC curves and its missed detection rate is very high with a horizontal-like curve around 0.54. For the AWGN attack, the F_1 score and precision curves are also decreasing rapidly when σ increases from 0 to 1 whereas EL-LOC remain at 0.7 again. The false alarm rate is also increasing starting at 0 meanwhile EL-LOC curves start to increase after $\sigma \geq 1$. The missed detection curve is similar to the one obtained for JPEG compression.

It can be seen that the authentication method QI-SVD is not robust against JPEG compression and AWGN. We propose some confusion map examples in figure 6 to observe a result sample of QI-SVD localization performances under both attacks. One can see that these attacks produce much more false alarms (blue artefacts) in subfigures (m) to (r).

An extra experiment with low pass filtering as an attack has also been performed but the proposed variants are not robust. Other attacks such as median filtering and gaussian filtering can be studied as a work perspective. Another class of image processing is the class of geometrical attacks (rotation, scaling and translation). To obtain robustness to such distortions, the coefficient synchronization strategy needs to be adapted. For example, an image rotation will change the position of blocks which is not considered in our paper.

In conclusion, the proposed authentication variant EL-LOC is able to resist JPEG compression and AWGN, i.e. is able to provide satisfying tampering localization performances thanks to a quantization-based embedding on DWT coefficients. The other studied variants also achieve similar performances but with a higher computation cost.

Before ending this section, we propose a comment on a common property of semi-fragile watermarking called image recovery as part of our work perspective to extend the proposed semi-fragile watermarking framework. This property allows the receiver end to recover tampered regions of the images. Based on our work, error correcting codes (BCH and Reed-Solomon codes in our work) can allow the receiver end recovering tampered regions of the images but it is not

possible with error locating (EL) codes.

When errors occur in an EL codeword, the goal of the decoding algorithm is to determine which blocks of the codeword contain errors but these errors are not corrected. However, EL codes decoding abilities are sufficient for an application in image tampering localization only. In the paper, EL codes are used in the variants REP-LOC and EL-LOC. However, the use of COR variants (with BCH and RS error correcting codes) may be a good start in order to be able to recover tampered image regions. When an image is watermarked using BCH-COR or RS-COR, some tampered coefficients can be recovered while other coefficients cannot be recovered due to the quantization method redundancy. Eventhough the embedded bits are fully recovered, the original associated quantization value cannot always be correctly deduced if the pixel modification is large since several quantization values encode the same information bit (redundancy property). After correction of the embedded bits, it may be possible to design an algorithm which guesses the original quantization values based on the information given by pixel neighborhood values. Then, coefficients that have the highest probability of being original can be considered as recovered. By applying this algorithm several times on previously recovered coefficients, it is possible to obtain a highly probable reconstruction.

5. Conclusion

We have proposed a semi-fragile watermarking algorithm using quantization in the DWT domain. It is equipped with different authentication variants and control codes to solve the problem of image tampering localization. We also introduced a point of view on error control codes applied in digital watermarking for image authentication with the introduction of SYN, LOC and COR authentication variants. In particular, we introduced a class of codes called error-locating codes that are able, by choosing suitable parameters, to further reduce the computation cost of image tampering localization compared to our baseline of codes.

On the experimental part, we showed that error localization (LOC) is as precise as error correction (COR). In addition, we showed that EL codes could achieve the same localization performances as Reed-Solomon codes except it is faster (quasi-linear decoding complexity) and simpler to use compared to Reed-Solomon and BCH codes. With different block sizes, performance tradeoffs can be easily chosen to fit the desired application.

In the mean time, we analyzed the performances of our algorithms in terms of image quality, tampering localization performances and robustness. Results showed that the authentication variant EL-LOC proposes the best tradeoff performances among all the variants and method QISVD. Finally, the semi-fragile algorithm is also robust to JPEG compression and additive white gaussian noise within a limited range of attacks parameters.

An example of perspective is to work on parity check matrix H_0 and more generally on EL codes to improve the error-decoding performances. In addition,

we are convinced it is possible to adapt the quantization step in function of the pixel value intensity to detect shadows as part of the tampered regions.

References

[1] X. Qi, X. Xin, A singular-value-based semi-fragile watermarking scheme for image content authentication with tamper localization, Journal of Visual Communication and Image Representation 30 (2015) 312–327.

[2] L. Bondi, S. Lameri, D. Güera, P. Bestagini, E. J. Delp, S. Tubaro, Tampering detection and localization through clustering of camera-based cnn features, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2017, pp. 1855–1864.

[3] C. Yan, C. Pun, Multi-scale difference map fusion for tamper localization using binary ranking hashing, IEEE Transactions on Information Forensics and Security 12 (9) (2017) 2144–2158. doi:10.1109/TIFS.2017.2699942.

[4] H. Yang, X. Sun, Semi-fragile watermarking for image authentication and tamper detection using hvs model, in: 2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07), IEEE, 2007, pp. 1112–1117.

[5] H. Liu, X. Yao, J. Huang, Semi-fragile zernike moment-based image watermarking for authentication, EURASIP Journal on advances in signal processing 2010 (1) (2010) 341856.

[6] S. Baudry, J.-F. Delaigle, B. Sankur, B. Macq, H. Maitre, Analyses of error correction strategies for typical communication channels in watermarking, Signal Processing 81 (6) (2001) 1239–1250.

[7] P. Lefèvre, P. Carré, P. Gaborit, Application of rank metric codes in digital image watermarking, Signal Processing: Image Communication 74 (2019) 119–128.

[8] J. Lee, C. S. Won, A watermarking sequence using parities of error control coding for image authentication and correction, IEEE Transactions on Consumer Electronics 46 (2) (2000) 313–317.

[9] Q. Sun, S.-F. Chang, K. Maeno, M. Suto, A new semi-fragile image authentication framework combining ecc and pki infrastructures, in: 2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No. 02CH37353), Vol. 2, IEEE, 2002, pp. II–II.

[10] D. He, Q. Sun, Q. Tian, A semi-fragile object based video authentication system, in: Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS'03., Vol. 3, IEEE, 2003, pp. III–III.

[11] X. Zhou, X. Duan, D. Wang, A semifragile watermark scheme for image authentication, in: 10th International Multimedia Modelling Conference, 2004. Proceedings., IEEE, 2004, pp. 374–377.

[12] C.-S. Chan, C.-C. Chang, An efficient image authentication method based on hamming code, Pattern Recognition 40 (2) (2007) 681–690.

[13] C.-S. Chan, An image authentication method by applying hamming code on rearranged bits, Pattern Recognition Letters 32 (14) (2011) 1679–1690.

[14] C.-C. Chang, K.-N. Chen, C.-F. Lee, L.-J. Liu, A secure fragile watermarking scheme based on chaos-and-hamming code, Journal of Systems and Software 84 (9) (2011) 1462–1470.

[15] P. W. Wong, N. D. Memon, Secret and public key authentication watermarking schemes that resist vector quantization attack, in: Security and watermarking of multimedia contents II, Vol. 3971, International Society for Optics and Photonics, 2000, pp. 417–428.

[16] X. Yu, C. Wang, X. Zhou, Review on semi-fragile watermarking algorithms for content authentication of digital images, Future Internet 9 (4) (2017) 56.

[17] J. Wolf, B. Elspas, Error-locating codes–a new concept in error control, IEEE Transactions on Information Theory 9 (2) (1963) 113–117.

[18] J. K. Wolf, On an extended class of error-locating codes, Information and Control 8 (2) (1965) 163–169.

[19] J. Wolf, On codes derivable from the tensor product of check matrices, IEEE Transactions on Information Theory 11 (2) (1965) 281–284.

[20] P. Das, Codes correcting and simultaneously detecting solid burst errors, Computer Engineering and Applications Journal 2 (2).

[21] P. K. Das, L. K. Vashisht, Error locating codes by using blockwise-tensor product of blockwise detecting/correcting codes, Khayyam Journal of Mathematics 2 (1) (2016) 6–17.

[22] P. Kumar Das, Codes detecting, locating and correcting random errors occurring in multiple sub-blocks, Proyecciones (Antofagasta) 38 (1) (2019) 129–144.

[23] A. Al-Fohoum, A. M. Reza, Combined edge crispiness and statistical differencing for deblocking jpeg compressed images, IEEE Transactions on Image Processing 10 (9) (2001) 1288–1298.

[24] Y. Luo, R. K. Ward, Removing the blocking artifacts of block-based dct compressed images, IEEE transactions on Image Processing 12 (7) (2003) 838–842.

[25] N. C. Tungala, A. Noore, Elimination of visual artifacts in digital image watermarking, in: Proceedings of the 35th Southeastern Symposium on System Theory, 2003., IEEE, 2003, pp. 64–68.

[26] K. Veeraswamy, S. S. Kumar, Adaptive ac-coefficient prediction for image compression and blind watermarking., Journal of Multimedia 3 (1).

[27] B. Chen, G. W. Wornell, Quantization index modulation: A class of provably good methods for digital watermarking and information embedding, IEEE TRANS. ON INFORMATION THEORY 47 (4) (1999) 1423–1443.

[28] P. Moulin, R. Koetter, Data-hiding codes, Proceedings of the IEEE 93 (12) (2005) 2083–2126. doi:10.1109/JPROC.2005.859599.

[29] F. Pérez-González, C. Mosquera, M. Barni, A. Abrardo, Rational dither modulation: a high-rate data-hiding method invariant to gain attacks, IEEE Transactions on Signal Processing 53 (10) (2005) 3960–3975.

[30] I. J. Cox, J. Kilian, F. T. Leighton, T. Shamoon, Secure spread spectrum watermarking for multimedia, IEEE transactions on image processing 6 (12) (1997) 1673–1687.

[31] P. Korus, J. Huang, Multi-scale analysis strategies in prnu-based tampering localization, IEEE Trans. on Information Forensics & Security.

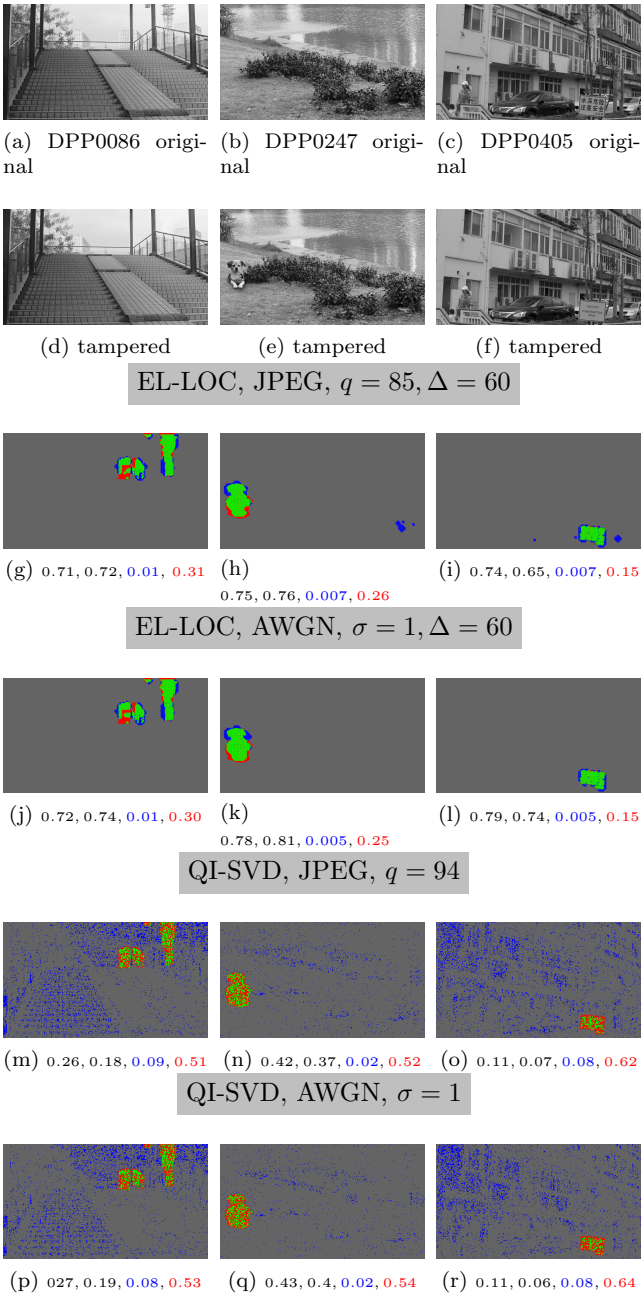


Figure 6: Confusion map examples from EL-LOC for images DPP0086, DPP0247, DPP0405 respectively from the left to the right. Each sub figure caption contain the following: (q, Δ, F_1, *Precision*, *FA rate*, *MD rate*).