EURASIP Journal on Information Security
a SpringerOpen Journal

RESEARCH Open Access

Error correcting codes for robust color wavelet watermarking

Wadood Abdul[1], Philippe Carré[1*] and Philippe Gaborit[2]

Abstract

This article details the conception, design, development and analysis of invisible, blind and robust color image watermarking algorithms based on the wavelet transform. Using error correcting codes, the watermarking algorithms are designed to be robust against intentional or unintentional attacks such as JPEG compression, additive white Gaussian noise, low pass filter and color attacks (hue, saturation and brightness modifications). Considering the watermarking channel characterized by these attacks, repetition, Hamming, Bose Chaudhuri Hocquenghem and Reed-Solomon codes are used in order to improve the robustness using different modes and appropriate decoding algorithms. The article compares the efficiency of different type of codes against different type of attacks. To the best of our knowledge this is the first time that the effect of error-correcting codes against different attacks are detailed in a watermarking context in such a precise way: describing and comparing the effect of different classes of codes against different type of attacks. This article clearly shows that list decoding of Reed-Solomon codes using the algorithm of Sudan exhibits good performance against hue and saturation attacks. The use of error correcting codes in a concatenation mode allows the non-binary block codes to show good performance against JPEG compression, noise and brightness attacks.

1 Introduction

Watermarking provides a possible solution to ensure and safeguard copyright and intellectual property rights for online multimedia content. The watermarking of color images raises the issues of robustness against intentional or unintentional attacks; the invisibility with respect to the human visual system (HVS); the maximum allowable information that can be inserted into the image and the security of the watermark. The watermarking algorithms must be designed in order to cater for these requirements. In this article, we will discuss and propose effective solutions for the issue of robustness related to color image watermarking.

Robustness to intentional attacks is a main issue for color image watermarking where the inserted watermark can be removed or manipulated to such an extent that the attribution of a watermark to a particular individual or image is difficult or impossible. The robustness of watermarking algorithms can also be affected by unintentional attacks which can result from a change in color space or common signal distortions.

The watermarking problem is considered analogous to the transmission of a signal over a noisy channel and the underlying characteristics of the channel are defined by the different attacks. Error correcting codes have been widely used to protect the signature (identification of a buyer/seller or transaction) of an image for watermarking applications. The robustness performance of our wavelet based watermarking algorithm (presented in Section 2), which uses the relation between wavelet color coefficients, is enhanced with the help of error correcting codes. The robustness improvement against attacks such as JPEG compression, additive white Gaussian noise, low pass filtering and color attacks (hue, saturation and brightness modifications) is demonstrated using different families and modes of error correcting codes. We explore and demonstrate the use and effectiveness of the concatenation of repetition codes, Hamming codes and BCH codes to enhance the robustness of the watermarking algorithm. Reed-Solomon codes are also used in a standalone manner using list decoding algorithm of Sudan [1] to correct errors resulting from attacks which can

*Correspondence: carre@sic.univ-poitiers.fr
[1]Laboratory SIC-XLIM, University of Poitiers, bat. SP2MI, av. Marie et Pierre Curie, 86960, Chasseneuil Cédex, France
Full list of author information is available at the end of the article

induce burst errors. Generally watermarking algorithms use bounded distance decoding algorithms for different error correcting codes along with the concatenation of these codes with each other [2-5]. We compare the performance of list decoding of Reed-Solomon codes with bounded distance decoding algorithms of repetition, Hamming, BCH, and the concatenation of these codes. Relatively recent developments [1,6] in the field of error correcting codes, for decoding Reed-Solomon codes, have made it possible to correct errors beyond the conventionally used bounded distance algorithms. As code rates tends towards 0, the list decoding of Reed-Solomon codes shows asymptotic improvement in performance over the bounded distance algorithms. As generally [5,7,8] the codes rates for watermarking schemes rates are very low, we can therefore employ this asymptotic improvement to our advantage.

These different error correcting codes exhibit different performance when the watermarked image is attacked. As the image and the attack have different characteristics we give the best error correcting code against the different types of attacks. We intend to find out the relationship between the different attack types and the protection provided by the error correcting codes. Our main focus in the robustness analysis is to provide suitable countermeasures against color attacks. Detailed analysis is carried out for the robustness issue against the attacks under consideration using the different modes and families of the error correcting codes.

Moreover, as watermarking algorithms do not usually consider color attacks and counter measures to protect the color images against such attacks have not been explored in earlier study. One of the objectives of this article is to study color attacks and propose adequate robustness measures.

In this study, our contribution is twofold. We first propose in Section 2, a wavelet based color image watermarking algorithm, with enhanced invisibility. The insertion is intended to keep the watermark invisible and the blind detection is performed without using the original (unwatermarked) image. In Section 3, we present the error correcting codes along with their use in the watermarking process. The last section studies the effectiveness of the codes against different types of attack.

2 Color image watermarking algorithm based on the wavelet transform

In this section, we want to describe the design of invisible color image watermarking schemes in terms of human perception of change and image quality degradation proposed by the authors. After, we also intend to improve the robustness performance of invisible color image watermarking algorithms so that such algorithms can resist intentional or unintentional attacks.

To cater for the requirements of invisibility and robustness, watermarking techniques employ the spatial and transform domains [9-13]. In general, the insertion of the watermark in the spatial domain has low complexity but also low robustness to attacks originating from the transform domains, such as JPEG compression, or for example median filtering. We could choose the band of frequencies in the multiresolution domain, thus giving us more control as to where to place the watermark. It is also important to note that these algorithms also differ in other aspects such as the way the watermark is prepared and inserted.

This article deals withe transform domain watermarking algorithm. Such algorithms employ discrete Fourier transform (DFT) [14-17], discrete cosine transform (DCT) [18-21], discrete wavelet transform (DWT) [22-27] and the contourlet transform (CT) [8] to insert the watermark with the best compromise between the invisibility, robustness and capacity criteria.

It is known that robustness against image distortion is enhanced if the watermark is placed in perceptually significant parts of the image. This contrasts with one of the requirements of an invisible watermarking algorithm, the embedded watermark should be invisible to the human visual system (HSV). Watermarking techniques have to be developed taking into account the masking properties of the HVS. Some characteristics of the HVS with respect to watermarking are highlighted in the literature [28,29]. These characteristics include frequency sensitivity, that is the difference sensitivity of the human eye to sine wave gratings at different frequencies; luminance sensitivity, that is the different sensitivity of the eye to a noise signal on a constant background, depending on the average value of the background luminance and on the level of the noise luminance; and contrast masking, which refers to the perception of a signal in presence of a masking stimulus, and which depends on the relative spatial frequency, location and orientation. For example, an approach [30] based on the Fourier transform insists that interoperability between the HVS model and the watermark embedding may not be optimal and the DCT and DWT domains do not allow the implementation of a suitable HVS model. Another important observation is that the CSF is not adapted to predict invisibility for complex signals such as natural images, essentially because the HVS is modeled by a single channel. Based on psychovisual experiments, they have derived a perceptual channel decomposition (PCD).

The algorithm presented during the course of this study is based on the wavelet transform. In the case of the DFT any change in the transform coefficients affects the entire image but in the case of the wavelet transform we have the additional spatial description of the image. Another great advantage is that we can adapt the watermark insertion according to the local image information. The DCT is

non adaptive to the image as the different levels of information could not be extracted and only the frequency information is present. Whereas transform domains such as the DWT map an image into the spatial-temporal domain. As a typical natural image is dominated by low frequency components, the energy concentration in corresponding coefficients could be efficiently exploited to insert the watermark. Those low frequencies represent the overall shapes and outlines of features in the image, and its luminance and contrast characteristics. High frequencies represent sharpness in the image, but contribute little spatial-frequency energy. The main advantage of the DWT and the CT is that we can choose the band of frequencies and the spatial and frequential combinations which are most suitable to carry the watermark. Our later discussion will be limited to the DWT.

The watermarking algorithms discussed, designed and implemented in this article belong to the class of blind algorithms [12,19,31-44] which means that (unlike the non-blind watermarking algorithms [45,46]) the originally image is not consulted at the time of the detection or decoding of the watermark.

One can mention mainly three insertion methods, each of them can be applied on pixels in the spatial domain, or on coefficients in any transform domain: LSB modification, Spread Spectrum and the Quantization Index Modulation. One such invisible watermarking algorithm uses LSB modifications of any color or grey-scale image [9]. The algorithm uses m-sequences due to their good auto-correlation properties and pseudo-random nature. The algorithm embeds the m-sequence on the LSB of the image. The watermark is decoded by comparing the LSB bit pattern with a stored counterpart. The Spread Spectrum techniques are well known in Communications for their low SNR operations. A message bit is "spread" using a pseudo-random unit vector. To decode, a scalar correlation is computed and the final decision is computed with a maximum likelihood decision rule. Lastly, the quantization index modulation is a generalization of LSB embedding. At each position, the quantizer Q_i is selected according to the message value $m = i$. To decode, the distances between the signal value and all the quantizer are computed and the smallest distance is selected.

The wavelet based color image watermarking algorithm presented in this Section is used to test the robustness improvement achieved by the incorporation of the different modes and families of error correcting codes. The signature is the information we want to embed into the image, it identifies uniquely the person who intends to watermark the image or a transaction. The watermark is the information that we actually embed into the image, it could be the same as the signature, or it could be processed. In our case, we pass the signature through an encoding procedure to make the watermark robust and invisible.

2.1 The watermark construction

The initial matrix or the signature is constructed by a random number generator according to the user specified parameters. After we apply the encoding scheme to construct the watermark which is then embedded into the image with certain limitations depending on image size and user defined parameters. The signature size is chosen to be 8×8 as it corresponds to a compromise between a sufficient size for a copyright application and a minimum robustness.

In the literature, we have many options to construct the final watermark from the initial matrix. For the purpose of illustration let us consider the very basic encoding scheme—the repetition codes. This signature or the initial matrix is repeated four times into an intermediate matrix which is again repeated according to the image capacity.

The human visual system is sensitive to changes in the lower frequencies as they are associated to the more significant characteristics of the image. The higher frequencies give the details of the image but changes in the higher frequencies could be easily eliminated by a low pass filter. Therefore the proposed algorithm uses middle frequencies for the insertion of the mark as both invisibility and robustness against low pass filter attacks is required in such an algorithm.

2.2 Wavelet decomposition

The wavelet decomposition is applied to each color component *R*, *G*, and *B*. The wavelet decomposition gives us the decomposition of the signal into different frequency bands. This decomposition is done by a filter bank in such a way that we split the low frequency band into small segments in order to separate all the components of the signal and we split the higher frequency bands into large segments as they contain less information. We embed the mark into middle frequencies as the higher frequencies could be simply eliminated by a low pass filter and the lower frequencies carry the overall form of the image and changing these lower frequencies may make the watermark visible.

2.3 Vector definition

The wavelet decomposition gives us the wavelet coefficients to the level/scale L associated to a middle frequency band. From these coefficients the vectors are defined

$$(\vec{V}_a[n,m])_{0<(n,m)<\frac{N}{2^L}}. \tag{1}$$

Such that,

$$\vec{V}_a[n,m] = \{d^a_{1,L}[n,m], d^a_{2,L}[n,m], d^a_{3,L}[n,m]\}. \quad (2)$$

With $a = \{R, G, B\}$, $[n, m]$ representing the coordinates, and $(d_{j,L})_{j=1,2,3}$ the sub-bands of the wavelet decomposition at the Lth level, as shown in Figure 1. The top right side of the Figure 1 (after wavelet decomposition) corresponds to the result of a low pass operation and the corresponding detail bands generated by the sub band filtering operation for the red color component. This process is repeated for each of the following wavelet decompositions until we reach the Lth level where we are interested to insert the watermark. Then the vectors are defined for each of the color component. The bottom part of Figure 1 shows the vector definition for the red color component at scale $L = 1$ and position $[n, m]$. Here $(d_{1,L})$ (resp., $(d_{2,L})$ and $(d_{3,L})$) corresponds to the horizontal (resp., vertical and diagonal) details of the image.

The maximum information (capacity) that can be embedded using the watermarking scheme is calculated using $\frac{D_x}{2^L} * \frac{D_y}{2^L}$, where D_x and D_y are the horizontal and vertical dimensions of the image and L is the level of the wavelet decomposition where we are interested to insert the watermark M.

2.4 Watermark insertion

In order to define the insertion process, we propose to adapt the QIM principle to vectorial case. For this, we will introduce a modification rule of color wavelet coefficients. As we have said the QIM uses a quantizer that is a function that maps a value to the nearest point belonging to a class of pre-defined discontinuous points. For non-adaptive QIM, the quantization step size is independent of the content. However, it is well known that the ability to perceive a change depends on the content. For example, the HVS is much less sensitive to changes in heavily textured regions and much more sensitive to changes in uniform regions. Moreover, the coefficient modifications and QIM process pose some challenges when applied generally to the color domain.

To account for this, we propose to use a method to automatically adapt the quantization step size at each sample. First, the step value is controlled by the wavelet coefficients that measure the spatial local activity. Second, the watermark insertion process is based upon moving one of the three color vectors (R, G, and B). A better candidate is defined in order to minimize the distortion at each insertion space.

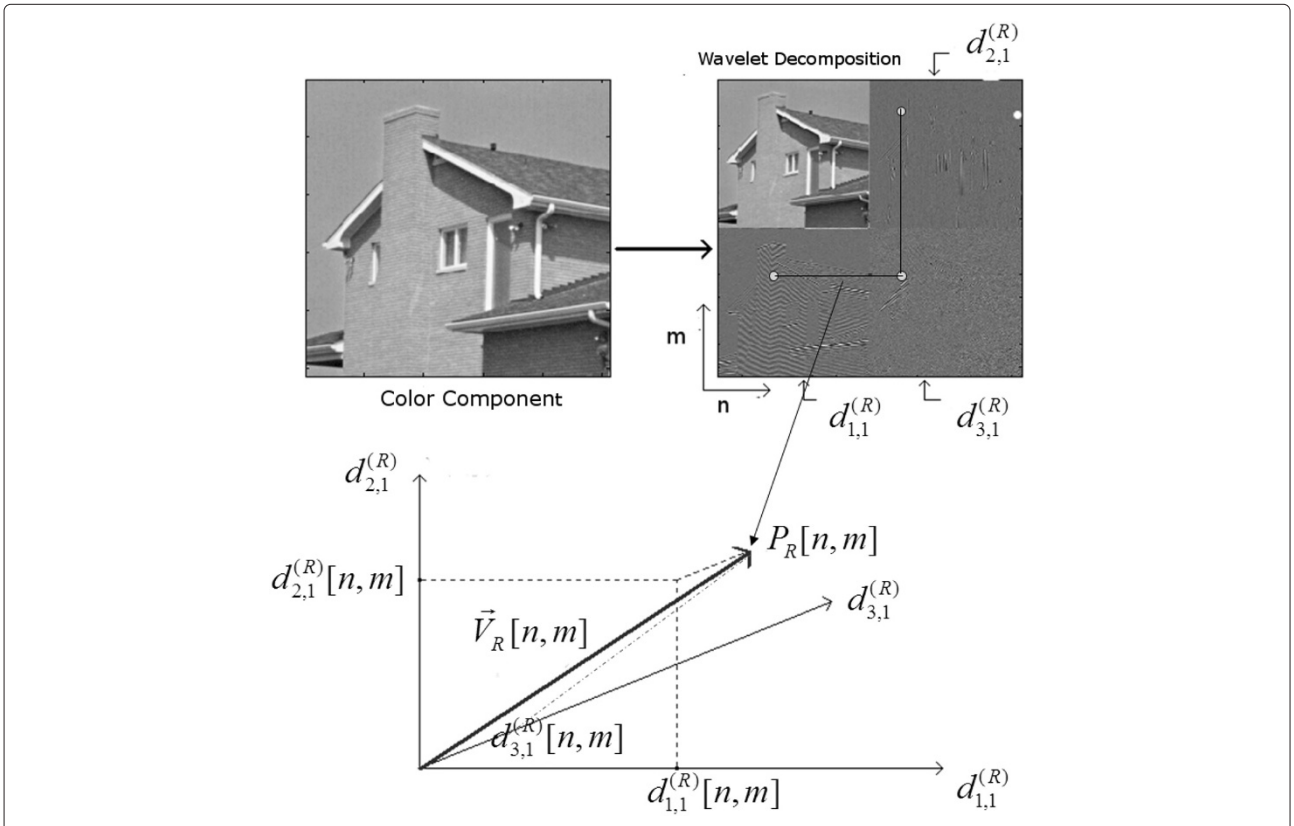

Figure 1 Vector definition. Definition of a vector $V_R[n, m]$ from the wavelet coefficients at scale $L = 1$ and for the red component at position $[n, m]$.

For each coordinate, we have to define one vector $\vec{V}_M$ that denotes the vector to be watermarked, and two reference vectors $\vec{V}_{vref1}$ and $\vec{V}_{vref2}$. $\vec{V}_M$, $\vec{V}_{vref1}$, and $\vec{V}_{vref2}$ are selected with respect of the correspondence to the following equations

$$\|P_{\text{ref1}} - P_{\text{ref2}}\|^2 = \max_{(a,b)\in\{R,G,B\}_{a\neq b}} \|P_a - P_b\|^2;\ P_M = P_c. \tag{3}$$

With $c \in \{R, G, B\}$, $c \neq a$ and $c \neq b$

Figure 2 shows that P_{ref1} and P_{ref2} are the most distant points from each pair of points and that is why P_R is chosen as P_M. Here P_x refers to the extreme point of the vector $\vec{V}_x$. P_M corresponds to $\vec{V}_M$ which is marked with the contents of the watermarking matrix M.

The watermarking convention is presented in Figure 3, where the watermarked vector $\vec{V}_{M,W}$, that corresponds to the original vector $\vec{V}_M$.

After watermarking, if $\vec{V}_M$ denotes $\vec{V}_R$, then:

- if $M[n,m] = 0$, then $\vec{V}_{R,W}[n,m]$ will be nearer to $\vec{V}_G[n,m]$ than $\vec{V}_B[n,m]$,
- else $\vec{V}_{R,W}[n,m]$ will be nearer to $\vec{V}_B[n,m]$ than $\vec{V}_G[n,m]$.

One of the most important possibilities lies on the ability of tuning the $P_{M,W}$ shift in order to limit the visual degradations on the image. Figure 4 shows the possible shifts of $P_{M,W}$. Two cases are considered, *Shift 1* and *Shift 2*. The limit of the two possible modifications is the median line between P_{ref1} and P_{ref2}. To be more robust, we define around this line a particular area (Figure 4) such that after the watermarking if $P_{M,W}$ is in this area, it has to be moved out by increasing the strength of the insertion process. The border of this area can be equivalent to $\pm 5\%$ of the distance between P_{ref1} and P_{ref2}.

With this approach, there exist two cases of *Shift*. In the first case, P_M is already nearest to P_{ref1} and the possible positions of $P_{M,W}$ after watermarking belongs to the segment $\vec{P}_M P_{\text{ref1}}$ (if P_M is out of the median area). In the second case, P_M is not already nearest to P_{ref1} and we create an intermediate point P_{int}, defined by:

- $\vec{P}_M P_{\text{int}}$ is parallel to $\vec{P}_{\text{ref1}} P_{\text{ref2}}$,
- P_{int} is located at the border of the median area (the distance between P_{int} and the median line must be equivalent to 5% of the distance between P_{ref1} and P_{ref2}).

Then, the possible positions of $P_{M,W}$ belongs to the segment $\vec{P}_{\text{int}} P_{\text{ref1}}$. For the *Case 1*, where P_M is the initial point of $P_{M,W}$, and for the *Case 2*, where P_{int} is the initial point of $P_{M,W}$, the watermark is defined by:

$$\vec{V}_{M,W}[n,m] = \vec{V}_{\text{refi}}[n,m] - (1 - F_a[n,m]).(\vec{V}_{\text{refi}} - \vec{V}_S[n,m]), \tag{4}$$

where $i = \{1,2\}$, $a = \{R,G,B\}$, $0 \leq F_a[n,m] \leq 1$, $S = \{M; int\}$ and F_a represents the weighted matrix for watermarking for each location $[n,m]$.

- If $F_a[n,m] = 0$, the force of insertion is minimum.
- If $F_a[n,m] = 1$, the force of insertion is maximum and $P_{M,W}$ is superposed on P_{refi}.

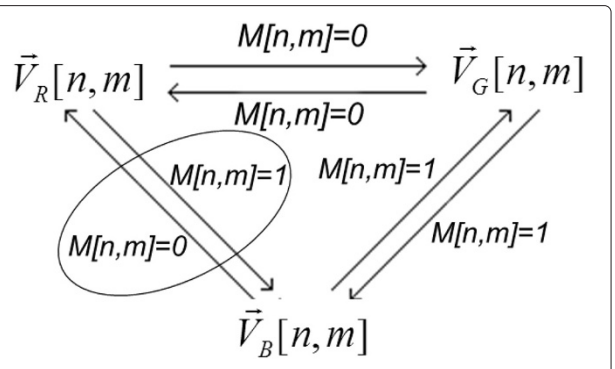

Figure 3 Watermarking convention. Example of watermarking convention.

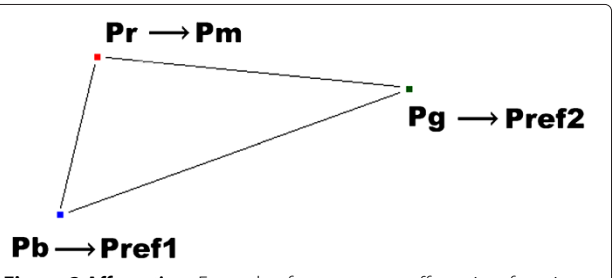

Figure 2 Affectation. Example of name vector affectation, function of their position from each other.

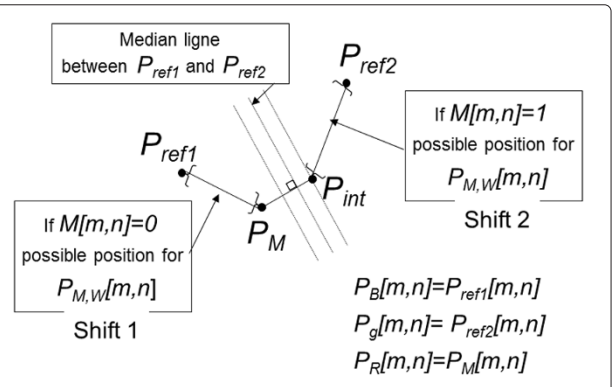

Figure 4 Modification. Possible positions of $P_{M,W}$.

In the case of maximum force of insertion $F_a = 1$, a conflict problem is highlighted, as shown on Figure 3 with a circle. It is observed that the bit value between P_R and P_B can be different: M can receive 0 or 1. This means that, in the detection step, the vector identification (V_R, V_R, and V_R to $\vec{V}_M$, $\vec{V}_{\text{ref1}}$, and $\vec{V}_{\text{ref2}}$) could be false. Thus, to avoid this configuration, F_a must be set inferior to 1.

The modification operation is applied on the whole host image in the wavelet domain. The last step in the watermark insertion process is the reconstruction of the image in to the spatial domain by inverse wavelet transform.

2.5 Watermark extraction

The first step of the extraction process consists also in a decomposition of the image with the same wavelet basis used in the insertion step. The watermark M_D is detected by measuring the largest distance between $||\vec{V}_{v\text{ref1}} - \vec{V}_M||$ and $||\vec{V}_{v\text{ref2}} - \vec{V}_M||$. Following the convention used in insertion, the watermark is thus reconstructed, bit by bit. The signature S_D is obtained by making an average and a binarization that corresponds to the coding method used for the creation of the mark M. In order to decide if S_D corresponds to S, a threshold is fixed to accept an extracted signature. This threshold is based on the acceptable level of bit error rate for the watermarking system.

Based on this we can define an acceptance threshold which decides as to whether to accept the detected watermark or to reject it. The different modes and families of error correcting codes, discussed in Section 3 help to lower this acceptance threshold, thus increasing the robustness of the watermarking scheme.

3 How to improve robustness with error correcting codes

Lots of research has been carried out to improve the robustness of a watermarked image where the use of error correcting codes to protect the signature is the most highlighted [2-5]. The watermarking problem is synonymous to the transmission of a signal over a noisy channel, where the image is considered to be the channel, the attacks are considered to be noise signals and the signature is considered to be the signal to be transmitted in the form of the watermark.

This section deals with the investigation of the performance capabilities of different error correcting schemes employed for a digital color image watermarking application based on the discrete wavelet transform. We worked on improving the robustness of the signature with the help of four families of error correcting codes. These four families of error correcting codes give different response when tested against different attacks on watermarked images. This is so because each of the attacks modifies the watermarked image in a diverse way and the properties exhibited by the error correcting codes are different against different error types (burst errors or random errors). To counter this problem we have employed repetition codes (presented at the first section), Hamming codes [47], Bose Chaudhuri Hocquenghem (BCH) codes [48] and Reed-Solomon codes [49].

In the literature different types of error correcting schemes for the watermarking problem are proposed. For example since 1998 Wang et al. use Hamming codes [50], Perreira et al. study the BCH codes for watermarking applications [51], some were hybrids between for example BCH and repetition codes [4]. Finally, some articles suggest using convolutional codes for watermarking [52,53]. Some compared different types of coding schemes, e.g., Reed Solomon, BCH and repetition codes [3].

What makes our study original is that we describe and compare the effect of different classes of codes against different type of real image attacks, we include different codes and the list decoding scheme in a color watermarking complete process. With this study, we propose to describe the errors introduced by different attacks and thus to illustrate the connection of a particular attack with a particular error correcting scheme in the context of our color wavelet algorithm. Since, there is a relationship between the contents of an image and the error nature attack, the result section analyzes the different results with empirical observation and provides intuitive explanations.

We adopted a rigorous testing process where we tested the robustness of different watermarked images with multiple signatures. We employed some standard attacks which include color attacks, filtering attacks, noise attacks and image compression attacks. The scheme had already been tested against some of these attacks with the use of repetition codes [5]. It proved to be robust against these attacks to a certain extent. We wished to explore the effectiveness of other error correcting codes against these attacks.

The different error correcting codes are tested using the wavelet based color image watermarking scheme presented in Section 2. Then using some possible attacks the robustness obtained using the different families and modes of error correcting codes is shown and the results are presented in Section 4.

3.1 Characteristics of the watermarking channel

Due to the requirement of watermark invisibility, the watermarks are weakly inserted in to the image. This makes the watermark signal prone to errors or attacks. The watermark channel is very noisy due to the different types of intentional or unintentional attacks. We consider the problem of watermark robustness against different errors or attacks analogous to the transmission of a signal over a noisy channel. To correctly transmit a signal over a noisy channel error correcting codes are used to protect the signal from the effects of the channel.

The characteristics of the watermarking channel depend upon the type of attacks experienced by the watermarked image. Like in the transmission of a signal over a noisy channel, error correcting codes are used to protect the signature in the form of a watermark so that the effects of the channel are reduced or minimized. The underlying characteristics of an image, e.g., the texture and color information also determine the effect an attack has on the watermarked image. The watermarking algorithm and the type and mode of error correcting codes also play an important role in defining the combined performance of robustness and invisibility.

The characteristics of the watermarking channel are primarily determined by the different attacks. We consider JPEG compression, additive white Gaussian noise, low pass filter, hue, saturation, and brightness as the underlying characteristics of the watermarking channel. Each of the different error correcting codes presented in the following Section exhibit different properties against these attacks.

The watermarking channel is characterized by very high error rates. To correct these errors we use different error correction schemes. Four families of error correcting schemes are used in our study to enhance the robustness of the watermark—repetition codes, Hamming codes, BCH codes and Reed-Solomon codes. We explore the use and effectiveness of the concatenation of these different families of error correcting codes to enhance the robustness of the watermarking scheme.

We employ a concatenation model where two of these error correcting codes are concatenated so that the two error correcting codes can facilitate one another. The outer error correcting codes are a second version of the repetition codes: the watermark is built up from some repetitions of the signature. The outer error help in reducing the error rates so that the inner error correcting codes (repetition, Hamming, or BCH) could then further reduce the errors so that the decision that the received watermark is valid or not could be taken.

Error correcting codes are expressed in the following article in the form of (n, k, d), where n is the length of the code, k is the dimension and d is the minimum Hamming distance between any pair of different codewords. The Hamming distance, H_d is based on the Hamming weight of a codeword c given by $H_w(c)$, the number of non zero elements in a vector. The Hamming distance H_d between two codewords is the number of elements in which they differ. The minimum Hamming distance d, between any two different codewords defines the error correcting capability of the particular error correcting code. An (n, k, d) error correcting code is capable of correcting t errors where $t < \frac{d}{2}$.

3.2 Concatenated error correcting codes

As we have said, the robustness of the watermarking scheme could be improved by concatenating these codes using signature repetition codes as outer codes and bit repetition, Hamming or BCH codes as inner codes. The outer coding is adaptive and is in accordance to the size of the image and user parameters, it is always repetition coding as shown in Figure 5.

At the receiver side an exact opposite procedure is applied to decode the signature from the watermark, i.e., we decode the watermark using repetition decoding first and then we decode the resulting information using repetition, Hamming or BCH decoding and we have the signature.

Such a concatenation mode has been selected because error correcting codes cannot display their potential unless the error rate induced by the channel is reduced below a critical value which brings about the possibility of first improving the channel error rate via repetition coding to an acceptable level, before any further decoding. The watermark channel may have to operate at very high bit error rates and codes such as BCH stop bringing in any advantage while the repetition codes continue with their modest protection. However concatenation of repetition and BCH codes is a way to improve the decoding performance when the error rates are high [3]. The BCH codes can correct up to $t = \lfloor (d-1)/2 \rfloor$ errors, all errors exceeding t may cause the decoder to decode erroneously. The repetition codes display better characteristics than BCH under high error rates. This could be seen in Section 4.2 (noise attack) where the repetition codes perform much better than the BCH (63,16,23) codes when the SNR < 2.

As mentioned in the introduction Reed-Solomon codes are used in a standalone mode to correct burst errors.

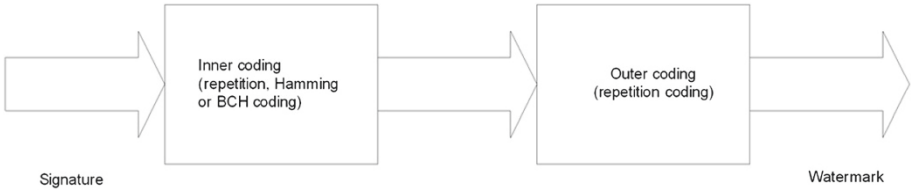

Figure 5 Concatenation model. Coding model of the concatenation codes.

The decoding of Reed-Solomon codes is carried out using list decoding algorithms [1,6]. The list decoding algorithms offer enhanced performance over bounded distance algorithms when the code rates are low.

3.3 Repetition codes

Repetition codes are used to construct the watermark from the signature and they are expressed in the form of (n, k, d). They are always used as $(n, 1, n)$ where each codeword is repeated n number of times. The repetition codes are used as inner codes in the construction of the watermark. They are also used as outer codes in all cases. The decoding of the repetition codes is always done using a mean operation on the received codeword to distinguish between a 0 or a 1.

3.4 Hamming codes

Hamming codes are linear block codes. For an integer $m > 1$, we have the following representation for the binary Hamming codes in the form $(n, k, d) = (2^m - 1, 2^m - 1 - m, m)$.

For $m = 3$, we have $(7, 4, 3)$ Hamming error correcting codes. These Hamming codes encode 4 bits of data into 7 bit blocks (a Hamming code word). The extra 3 bits are parity bits. Each of the 3 parity bits is parity for 3 of the 4 data bits, and no 2 parity bits are for the same 3 data bits. All of the parity bits are even parity. The $(7, 4, 3)$ Hamming error correcting code can correct 1 error in each of the Hamming codeword.

When we multiply the received codeword with the parity check matrix we get the corresponding parity ranging from 000 to 111. These three bits give us the error location. 000 indicating that there were no errors in transmission and the rest from 001 to 111 indicate the error location in our seven bit received codeword. Here we can correct one error according to $t = \lfloor (d-1)/2 \rfloor$ as the minimum Hamming distance between our code words is $7 - 4 = 3$, we have 1 as the number of correctable errors. Now we have the error location, we could simply flip the bit corresponding to the error location and the error will be corrected. Then we discard the parity bits from position one, two and four we have our received data words. Hamming Codes are perfect 1 error correcting codes. That is, any received word with at most one error will be decoded correctly and the code has the smallest possible size of any code that does this. The Hamming codes that we used could correct 1 error in each codeword. There was a need to test other types of codes which can correct more errors. We selected the BCH codes which are explained in the following section.

3.5 Bose Chaudhuri Hocquenghem (BCH) codes

BCH codes are cyclic block codes such that for any positive integers $m \geq 3$ and t with $t \leq 2^{m-1} - 1$, there is a BCH codes of length $n = 2^m - 1$ which is capable of correcting t error and has dimension $k = n - m * t$.

Let C be a linear block code over a finite field F of block length n. C is called a cyclic code, if for every codeword $c = (c_1, \ldots, c_n)$ from C, the word $(c_n, c_1, \ldots, c_{n-1})$ in F^n obtained by a cyclic right shift of components is also a codeword from C.

We have selected the BCH$(15, 7, 5)$ and the BCH$(63, 16, 23)$ error correcting codes for the purpose of our experimentation. The BCH$(63, 16, 23)$ is in line with our algorithm testing parameters since the size of our initial matrix is 8×8 bits.

3.6 Reed-Solomon codes

Reed-Solomon codes [49,54] are q-ary $[n, k, d]$ error correcting codes of length n, dimension k and Hamming minimum distance d equal to $n - k + 1$. These codes can decode in a unique way up to $\frac{n-k}{2}$ errors, and there exists the possibility to decode them beyond the classical bounded radius $\frac{n-k}{2}$. Usually these codes are considered over the Galois field $GF(p^m)$ (for p a prime) and have parameters $[p^m - 1, p^m - 1 - 2t, 2t + 1]$. In particular the case $p = 2$ is often considered for applications since in that case any symbol of the code can be described with m bits. It is also possible either by considering less coordinates in their definition, either by shortening them, to construct Reed-Solomon codes over $GF(p^m)$ with parameters $[p^m - 1 - s, p^m - 1 - 2t - s, 2t + 1]$, which can be decoded in the same way that non shortened Reed-Solomon codes.

Reed-Solomon codes are particularly useful against burst noise. This is illustrated in the following example.

Consider an $(n, k, d) = (40, 11, 30)$ Reed-Solomon code over $GF(2^6)$, where each symbol is made up of $m = 6$ bits as shown in Figure 6. As $d = 30$ indicates that this code can correct any $t = 14$ symbol error in a block

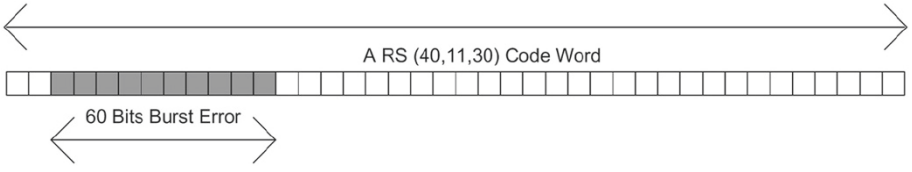

Figure 6 Reed-Solomon codes. Burst error performance of Reed-Solomon codes.

of 40. Consider the presence of a burst of noise lasting 60 bits which disturbs 10 symbols as highlighted in Figure 6. The Reed-Solomon $(40, 11, 30)$ error correcting codes can correct any 14 symbol errors using the bounded distance decoding algorithm without regard to the type of error induced by the attack . The code corrects by blocks of 6 bits and replaces the whole symbol by the correct one without regard to the number of bits corrupted in the symbol, i.e., it treats an error of 1 bit in the symbol in the same way as it treats an error of 6 bits of the symbol—replacing them with the correct 6 bit symbol. This gives the Reed-Solomon codes a tremendous burst noise advantage over binary codes. In this example, if the 60 bits noise disturbance can occur in a random fashion rather than as a contiguous burst, that could effect many more than 14 symbols which is beyond the capability of the code.

In the watermarking channel, the errors, characterized by the different attacks, occur in random or burst manner. Depending on the placement of the watermark in an image and the use of error correcting codes, the robustness of the signature can be increased against the attacks.

For Reed-Solomon codes the conventionally used, bounded distance decoding algorithms correct up to $t = \lfloor (n-k)/2 \rfloor$ symbol errors as shown in the above example. Using list decoding, Sudan [1] and later Guruswami-Sudan [6] showed that the error correcting capability of Reed Solomon could be improved to $t_S = n - \sqrt{2kn}$ and $t_{\mathrm{GS}} = n - \sqrt{nk}$ respectively.

3.7 List decoding of Reed-Solomon codes

It is well known that for a linear code $[n, k, d]_q$ over the field $GF(q)$, of length n, dimension k and distance d, it is possible to decode the code in a unique way up to a number of errors: $t = [(d-1)/2]$. Now what happens if the number of errors is greater than t? Clearly there will always be cases where a unique decoding will not occur. For instance if d is odd and a codeword c has weight d, any element x of weight $(d+1)/2$ (which support the set of non zero coordinates) is included in the support of c, will be at distance $(d+1)/2$ of two codewords: x and $(0, 0, \ldots, 0)$, which gives two possibilities for decoding. Meanwhile if one considers a random element of weight $(d+1)/2$ the probability that such a situation occurs is very unlikely. A closer look at probabilities leads to the fact that in fact even for larger t (but with t bounded by a certain bound, called the Johnson bound) the probability of a random element to be incorrectly decoded is in fact very small.

The idea of list decoding is that for $t > (d-1)/2$ a list decoding algorithm will output a list of codewords rather than a unique codeword. List decoding was introduced by Elias [55], but the first usable algorithm for a family of codes, the Reed Solomon codes, was proposed by Sudan in [1], later the method was improved by Guruswami and Sudan [6].

The list decoding method is a very powerful method but it is slower than classical algorithms which decode less errors. For usual context in coding theory the decoding speed is a very important factor since one wants to optimize communications speed, but there exist contexts in which the use of such a decoding is not as important since the use of the algorithm is only causal in the overall process. This is for instance the case in cryptography and in traitor tracing schemes [56] where list decoding algorithms are used when one wants to search a corrupted mark (which does not occur all the time).

The principle of the algorithm is a generalization of the classical Welch-Berlekamp algorithm, the algorithm works in two steps: first construct a particular bivariate polynomial $Q(x, y)$ over $GF(q)$ and then factorize it for finding special factors. These factors lead to a list of decoded codewords.

The first algorithm by Sudan permits (for $k/n < 1/3$) to decode up to $n - \sqrt{2kn}$ errors rather than $n/2$ for classical algorithms. This method is based on Lagrange interpolation.

The list decoding algorithm of Sudan [1,57] is detailed in the following steps

For a received codeword $r = (r_1, r_2, \ldots, r_n)$ and a natural number

$$t_S < n\frac{l}{l+1} - \frac{l}{2}(k-1) \tag{5}$$

and for

$$\frac{k}{n} < \frac{1}{l+1} + \frac{1}{n} \tag{6}$$

1. Solve the following system of linear equations

$$\sum_{j=0}^{l} \begin{bmatrix} r_1^j & \ldots & 0 & 0 \\ 0 & r_2^j & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \ldots & r_n^j \end{bmatrix} \begin{bmatrix} 1 & x_1 & \ldots & x_1^{l_j} \\ 1 & x_2 & \ldots & x_2^{l_j} \\ \vdots & \vdots & \ldots & \vdots \\ 1 & x_n & \ldots & x_n^{l_j} \end{bmatrix} \begin{bmatrix} Q_{j,0} \\ Q_{j,1} \\ Q_{j,2} \\ \vdots \\ Q_{j,l_j} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{7}$$

   where $l_j = n - 1 - t_G - j(k-1)$.
2. Put

$$Q_j(x) = \sum_{r=0}^{l_j} Q_{j,r} x^r$$

and

$$Q(x, y) = \sum_{j=0}^{l} Q_j(x) y^j.$$

3. Find all factors of $Q(x,y)$ of the form $(y-f(x))$ with degree $(f(x)) < k$.
4. A list of factors $f(x)$ that satisfy the following is obtained

$$H_d((f(x_1),f(x_2),\ldots,f(x_n)),(r_1,r_2,\ldots,r_n)) \leq t_G$$

5. Calculate $f(x)$ over the encoding elements to obtain the corrected codeword $(c_1,c_2,c_3,\ldots,c_n)$.

The second method of Guruswami and Sudan [6,57] permits to decode up to $n-\sqrt{kn}$ errors, but is trickier to use since it is based on Hermite bivariate interpolation and on the notion of Hasse derivative.

For a bivariate polynomial:

$$Q(x,y)=\sum_{a=0}^{\infty}\sum_{b=0}^{\infty} q_{a,b}x^a y^b,$$

the Hasse derivative for the point (a',b') is defined as:

$$Q^{[a',b']}(x,y)=\sum_{a\geq a',b\geq b'}\binom{a}{a'}\binom{b}{b'}q_{a,b}x^{a-a'}y^{b-b'}.$$

In practice the hard step of decoding is finding the polynomial $Q(x,y)$. It can be done in cubic complexity in an elementary (but slow) way by the inversion of a matrix, or also in quadratic complexity but with a more hard to implement method [58].

The Guruswami-Sudan list decoding algorithm detailed in [1,57] could be summarized in the following three steps

1. For a received word $(r_1,r_2,r_3,\ldots,r_n)$ and encoding elements $(x_1,x_2,x_3,\ldots,x_n)$ belonging to a Galois Field, solve for $Q_{a,b}$ the system of homogeneous linear equations

$$\sum_{a\geq h,b\geq u}\binom{a}{h}\binom{b}{u}Q_{a,b}x_i^{a-h}r_i^{b-u}=0, \tag{8}$$

where $h+u<s$, $i=1,2,\ldots,n$, and s is a natural number.
$Q_{a,b}=0$ if $l>a$ or $b>l_a$ where
$l_a=s(n-t_{\mathrm{GS}})-1-a(k-1)$ and l and s are the list size and multiplicity factor [6,57] for the Reed-Solomon code. Where

$$t_{\mathrm{GS}}=\frac{n(2l-s+1)}{2(l+1)}-\frac{l(k-1)}{2s} \tag{9}$$

and

$$\frac{k}{n}\leq\frac{1}{n}+\frac{s}{l+1} \tag{10}$$

2. Put $Q_j(x)=\sum_{u=0}^{l_j}Q_{j,u}x^u$ and consequently $Q(x,y)=\sum_{j=0}^{l}Q_j(x)y^j$.
3. Find all factors of $Q(x,y)$ of the form $(y-f(x))$ with $degree(f(x)) < k$,and then calculate $f(x)$ over the encoding elements to obtain the corrected codeword $(c_1,c_2,c_3,\ldots,c_n)$.

The performance of Guruswami-Sudan algorithm is better than the algorithm proposed by Sudan when the code rate $R=k/n$ is high. When the code rate is very low they have similar performance. The performance of both the list decoding algorithms shows clear improvement over the bounded distance (BD) algorithms when the code rate is low. We exploit this property of the list decoding algorithms to encode the signature in to the watermark. The improvement of performance is shown in Figure 7.

We select the Sudan's algorithm for the purpose of decoding as the code rate $R<1/3$ for the watermarking scheme presented in Section 2 [5] for a signature size of 64 bits and the actual performance gain by the Guruswami-Sudan over the Sudan algorithm is not significant. The parameters used to demonstrate the performance of the Reed-Solomon codes are RS$(40,11,30)$, RS$(127,9,119)$, and RS$(448,8,441)$ and the code rates for these three cases are 0.275, 0.071, and 0.018 respectively. Therefore it is useless to use the high complexity Guruswami-Sudan algorithm as for the code rates are very low for the given cases and Sudan algorithm has similar performance, specially for RS$(127,9,119)$ and RS$(448,8,441)$, as seen in Figure 7.

According to the properties of the different codes, we are now going to study the integration of these tools in our color watermarking process.

4 Tests and results

For the tests we have used 15 images of different types and sizes (256×256, 512×512, and 1024×1024). In this article, we focus on the largest images 1024×1024 shown in Figure 8. Each image is marked with 5 different signatures where the signature size is 64 bits. This made us independent of the signature and helped us to measure

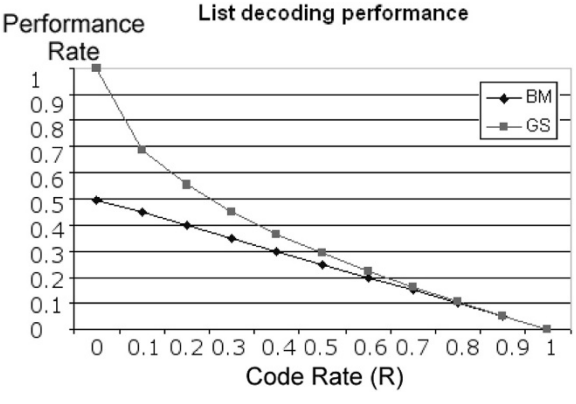

Figure 7 List decoding. List decoding performance comparison.

Figure 8 Test Images.

certain robustness. The attack types under study are JPEG compression, noise, low pass filter, hue, saturation and brightness attack. Then corresponding graphs show average performance of the error correcting codes, BER (bit error rate) against the attack type for the different image sizes. Bit error rate (BER) is used to measure the performance of the error correcting codes BER $= \frac{B_E}{B_T}$, where B_E is the number of erroneous bits received for each attacks and B_T is the total number of bits of the signature.

In the following figures, the graphs represent the parameter of the attack on the x-axis and on the y-axis the bit error rate is shown. The graphs show the average for each of the tested image size and error correcting codes against each of the attacks.

4.1 JPEG compression

Due to the lossy JPEG compression we lose the higher frequency components which results in blurring of the image. As the watermarking algorithm embeds the watermark in middle frequencies, the attack does not completely remove the watermark at low levels. Blurring and blocking effects can be noticed in attacked images with a higher compression level causing random errors in the watermark.

We tested our method against JPEG attack by starting off the compression from a quality level of 1% to a quality level of 96% with a step size of 5% as one could see in Figure 9a.

In general, we can say that our watermarking method is robust against JPEG compression where the compression level is up to 50%. If the compression level of the image is reduced beyond 50% then the image is quite degraded. This is so because the JPEG compression starts to compress middle frequencies where the watermark resides.

If the quality level is reduced beyond 50% then we observe loss of high frequency information, artifacts on subimage boundaries and localized stronger artifacts. The loss of high frequency information has no effect, but the localized artifacts have influence to some wavelet coefficient values. Due to the relationship between the contents of an image and the position of the artifacts, the effects of the errors induced by the JPEG compression on the wavelet coefficients are random in nature.

Given the random comportment of the errors induced by the JPEG compression, the BCH codes show the best performance in all cases of image size, with and without the extra protection of repetition codes, and the Reed-Solomon codes give the worst performance due to the random nature of the attack.

As we have said, there is also a relationship between the contents of an image and the attack applied. The dependability of this relationship and the type of error correcting codes used is also significant. The test images have different distribution of frequency content. The watermarking scheme uses these characteristics to insert the watermark. The robustness of these images using Reed-Solomon codes against JPEG compression is shown in Figure 9b.

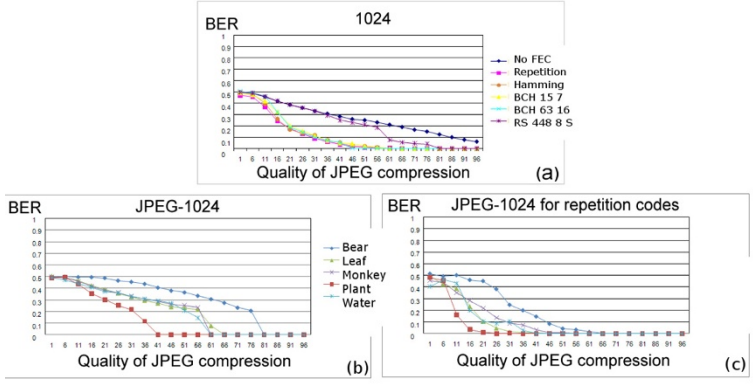

Figure 9 JPEG compression attack. (a) Comparison of coding schemes against JPEG compression for image size 1024 × 1024; **(b)** Comparison of different images for RS codes; **(c)** Comparison of different images for repetition codes.

The effect of the use of the different types of error correcting codes and the relationship between the type of image used for the JPEG compression attack could be observed while considering repetition codes and Reed-Solomon codes. Let us consider repetition codes for the same images. In the results shown in Figure 9b,c we can notice the difference in performance of the repetition codes and Reed-Solomon codes for the same images. It is observed that the repetition codes protect the same images at higher compression levels than the Reed-Solomon codes but with the same sensibility to the characteristics of the image.

Notice the difference in robustness of a compression factor of 40 for the image *Bear* and the image *Plant* shown in Figure 8. This difference in performance for the two images is due to their characteristics, the manner in which watermarking scheme exploits these characteristics to insert the watermark and how these locations are effected by the attack. The frequency contents of the image Bear are generally high, very sensitive to the compression scheme. Whereas the figure Plant has relatively lower frequency content and has important discontinuities. The watermarking scheme uses middle frequencies to insert the watermark and the JPEG compression attack will first remove any high frequencies in an image, this explains the observed results.

To conclude, the analysis of robustness against JPEG compression using the different families of error correcting codes shows that the image play a very important role for the overall robustness of the watermarking algorithm, and also shows that different error correcting codes have different performance against the different attacks for every type of image. We consider that Color watermarking scheme with BCH codes is usually robust to JPEG compression.

4.2 Additive white Gaussian noise

The type of noise that we introduce into the image is additive white Gaussian noise (AWGN). The insertion of the noise or the attack is completely random in nature and there are no bursts of noise. The x-axis in the Figure 10a shows the signal to noise ratio (SNR) and the y-axis shows the bit error rate. The AWGN is distributed uniformly through the image and due to the randomness of the attack the wavelet coefficients are effected uniformly. The difference in the performance of the Reed-Solomon codes and the others is evident as all other codes except the Reed-Solomon codes are more capable to correct the random errors.

4.3 Low pass filtering

The low pass filter gives a blurring effect to an image as it filters out high frequency components from the image. The watermark is robust against low pass filtering as it is not embedded into high frequencies. The error correcting codes perform equally well for the low pass filter attack, the bit error rate is 0 (Figure 10b) except for some cases of large filter dimensions (9×9) and no code protection. This is because the low pass filter starts filtering frequencies where the watermark is embedded but the error correcting codes provide enough robustness so that this error remains negligible. This robustness against low pass filter is a generic feature of transform domain watermarking schemes as usually transform domain watermarking schemes do not insert the mark in high frequencies.

One of the main objectives of this article is to highlight the effects of modification of the color components for watermarking schemes and provide effective robustness against color attacks. We will now discuss some of these color attacks and effective counter measures using error correcting codes.

4.4 Hue

The term hue describes the distinct characteristics of color that distinguishes red from yellow and yellow from blue. These hues are largely dependent on the dominant wavelength of light that is emitted or reflected from an object. Hue is the angle between the color vector associated with the pixel and a color vector taken anywhere on the plan orthogonal to grey axis and which sets the reference zero Hue angle. This reference Hue value is often taken to represent the red color vector, so we decided arbitrarily to associate the red color vector and gave it a

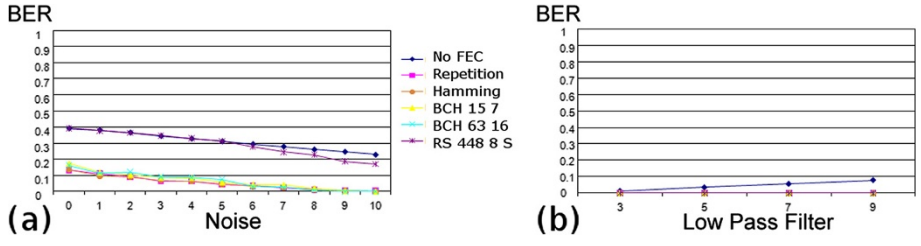

Figure 10 Noise and low pass filter attacks. Comparison of coding schemes against: **(a)** noise (AWGN) **(b)** low pass filter for image size 1024×1024.

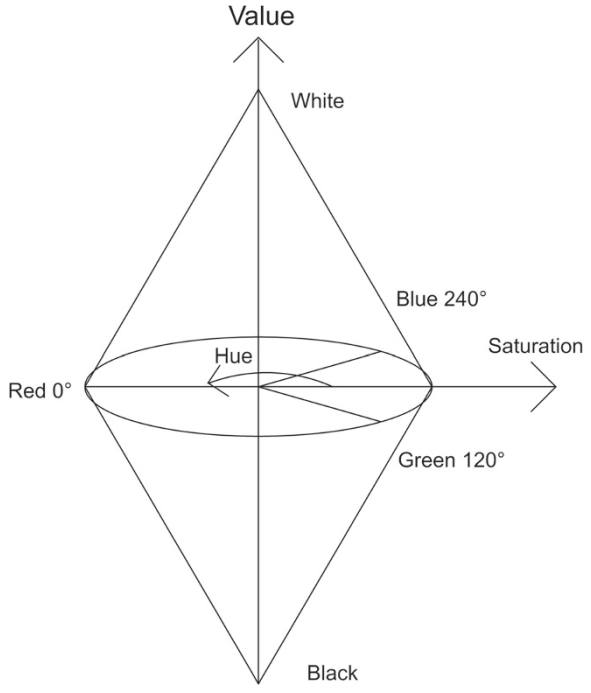

Figure 11 Hue, saturation, and value.

zero Hue value (Figure 11). Hue is the angle between this reference color vector and the color vector.

The scheme is not very resistant against changes in hue because they effect directly the color vectors that we use in the watermarking process.

In general, all the coding schemes except the Reed-Solomon codes give almost the same results as the error rates are beyond their error correcting capacity. To characterize the noise introduced by hue modification, we can say that changes in hue affect all coefficients representing a particular color similarly and this would have hue effect in blocks.

This is in accordance with the block design of Reed-Solomon codes which help protect the watermark to a certain limit. The progressive improvement of the performance of Reed-Solomon codes could be seen in Figure 12, where reasonable changes modulo 180° are correctable when using RS(448, 8, 441). The little resistance to the changes in hue is provided by list decoding of Reed-Solomon codes as changes in hue effect all the color components at the same time, consequently effecting the wavelet coefficients, where the watermark has been inserted.

Even Reed-Solomon codes are not able to resist the hue attack if the force of the attack is increased (the changes in hue are increased) as the blocks of Reed-Solomon codes and the blocking effect of changes in hue do not cater for exactly the same wavelet color vectors.

4.5 Saturation

Saturation is the measure of color intensity in an image, the Saturation is the distance between the color vector and the grey axis (Figure 11). The less saturated the image, the more washed-out it appears until finally, when saturation is at −100, the image becomes a monochrome or grayscale image and as the saturation increases the colors become more vivid until they no longer look real.

The negative saturation poses no problem to any of our error correction schemes except for a value of −100 but then the image is just a grayscale image. Now the image is desaturated and all the color planes have the same values and there is no difference between the different color vectors. Therefore the watermarking algorithm is not able to decode the watermark. When the image is not completely desaturated then the difference between the wavelet color coefficients is changed in relative proportion to the changes in saturation and the watermarking algorithm is able to decode the signature without the help of error correcting.

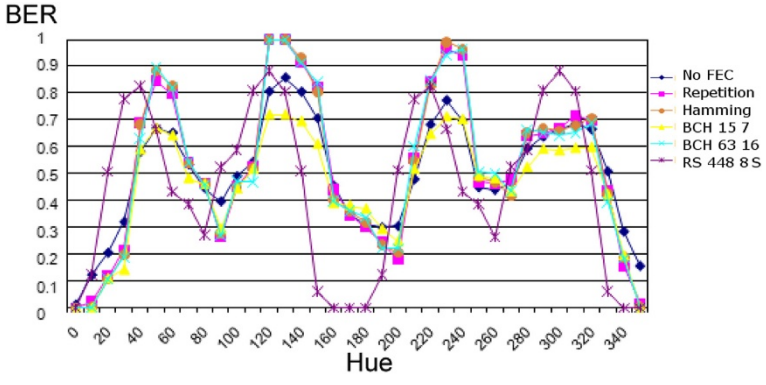

Figure 12 Comparison of coding schemes against hue for image size 1024 × 1024.

On the positive side, except the list decoding of Reed-Solomon codes, the coding schemes start giving a significant number of errors when the change in saturation exceeds a value of about 30.

As for the previous hue attack, the saturation attacks has effect in blocks. In fact, the saturation on the positive side might not effect the watermark if the change in saturation does not effect the color vectors strongly. But if the change in saturation effects the color components beyond a certain limit, then the error correcting codes are not able to provide robustness for these high levels. This is the case where the pixels values go out of bounds after processing. To take care of the R, G, and B values exceeding the bound, this problem is tackled by clipping the out of boundary values to the bounds. Clipping the values to the bounds create undesirable shift of hue and as in the previous section affect all coefficients representing a particular color similarly and this would have effect in blocks. This is in accordance with the block design of Reed-Solomon codes. The performance improvement using the Reed-Solomon codes is evident in Figure 13a, where the BER < 0.1. Furthermore if the image is saturated beyond a value of 60, it ceases to have a commercial value as natural images are not saturated to such values on the positive side.

The saturation attack, like the other attacks, has dependency on the contents of the image and the robustness measure applied to protect the watermark. The BER increase in Figure 13a for high saturation values is due to the image *Leaf* and image *Monkey* (Figure 8) as shown in Figure 13b and it is not attributed entirely to the incapability of list decoding of Reed-Solomon codes to decode the watermark for highly saturated images. For example, the original image Leaf is highly saturated and the modification of the saturation is quickly associated with R, G, and B values exceeding the bound.

4.6 Brightness

The brightness of a color measures the intensity of light per unit area of its source. We consider that brightness is the norm of the color's orthogonal projection vector on the grey axis (Figure 11). It is enough to say that brightness runs from very dim (dark) to very bright (dazzling).

Our scheme resists the modification in brightness in the negative side to the extent of -90, where the image is hardly invisible. On the positive side at a value after 40 none of the schemes could correct the errors but then the image is degraded to such an extent that it is no longer useful.

The modification of the Brightness distorts the saturation as a side effect and the effect of saturation change is more significant when the change in luminance is important. Contrary to the others attacks, it may be more difficult to fully characterize this transform. The attack impact will depend on a significant number of factors (the parameter of the modification, the saturation value, the intensity value). Furthermore, the modification of the brightness may be considered as a random degradation dependent on the characteristics of the image.

In general, all the coding schemes except the Reed-Solomon codes give almost the same results as the error rates are beyond their error correcting capacity and give better performance than RS code for large size images (Figure 14a). Moreover, the response of individual watermarked images is also dependent on the characteristics of the image as shown in Figure 14b. For example, using Reed-Solomon codes, the figure "Bear" has the same robustness as for other error correcting codes, this image is unsaturated. The rest of the images have different robustness.

The results and the special cases discussed in this section show that the characteristics of the image, the watermarking scheme, the use of error correcting codes and the effects of the attack all play a very important role determining the robustness for watermarking schemes. Each of these characteristics has to be considered separately and also the relationship between each of these issues is to be considered while determining the robustness performance of watermarking schemes.

On the whole, we can say that the strategy we propose based on wavelet domain and error correcting codes (with judicious choices) perform well for the different attacks.

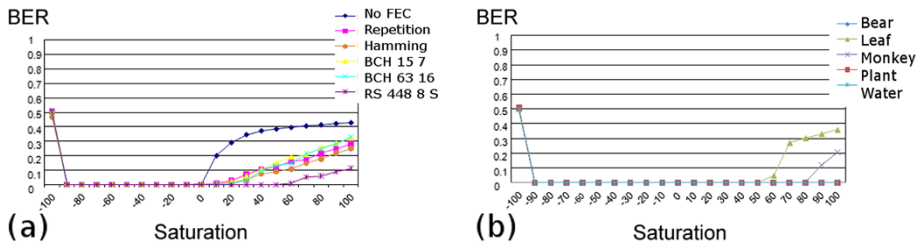

Figure 13 Saturation attack. (a) Comparison of coding schemes against saturation; **(b)** Comparison of different images against saturation for image size 1024×1024.

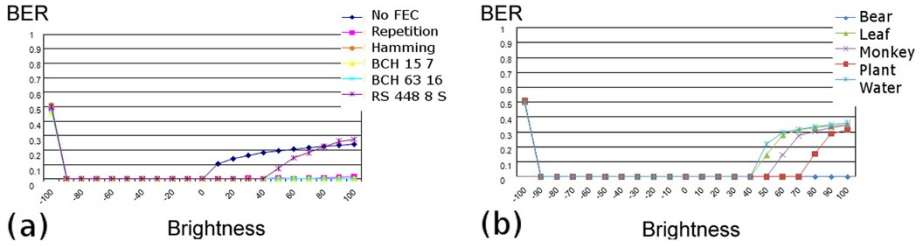

Figure 14 Brightness attack. (a) Comparison of coding schemes against brightness; **(b)** Comparison of different images against brightness for image size 1024 × 1024.

5 Conclusion

It is general knowledge that no digital image watermarking scheme is robust against all types of attacks. The purpose of this article is to be able to evaluate the performance of the error correcting codes with reference to each attack type.

The first contribution is to adapt the QIM principle to vectorial case with a watermark insertion process based upon moving one of the three color vectors (R, G, and B). This process, associated with the Wavelet decomposition is defined in order to minimize the color distortion .

The second contribution concerns the investigation of the performance capabilities of different error correcting schemes employed for a digital color image watermarking application based on the discrete wavelet transform. We have worked on improving the robustness of the signature with the help of four families of error correcting codes. We have described and compared the effect of different classes of codes against different type of real image attacks, we have included different codes and the list decoding scheme in a color watermarking complete process. For this, we have analyzed the errors introduced by different attacks and highlighted the connection of a particular attack with a particular error correcting scheme in the context of our color wavelet algorithm.

The results shown in this article confirm that the approach of using error correcting codes as a tool to enhance the robustness of watermarking schemes. During the course of this study, we observed that the use of different types of error correcting codes give different robustness to the inserted watermark. This robustness to the different types of attacks depends upon the underlying characteristics of the image.

It was observed that the four different families of error correcting codes exhibit different characteristics when different attacks are applied to different watermarked images. In general, the BCH$(63, 16, 23)$ outperform the other error correcting codes used to evaluate the bounded distance algorithms but when the error rates are very high then the repetition codes continue to give modest performance. This is of no great advantage as the image is quite degraded in such cases.

This property of the repetition codes helps in the concatenation model and they are always used as the outer error correction codes. It was also observed that the concatenation model used in the course of this study is very useful in terms of reducing the error rate so that the inner error correcting codes can correct the errors at a lower error rate. We conclude that in general the BCH$(63, 16, 23)$ (using with the repetition codes and the concatenation model) give us the best performance against attacks which induce errors in a random manner.

The list decoding of Reed-Solomon codes using Sudan's algorithm was used to further enhance the robustness of the watermarking scheme. We obtain very good results especially for codes with low rates. An important consequence of using the method is that when the code rate k/n tends to 0 the algorithm can decode asymptotically $n\left(1-\sqrt{\frac{2k}{n}}\right)$ errors which means that the proportion of potential errors tends to 1.

The list decoding of Reed-Solomon codes shows better performance for color attacks, specially hue and saturation, where the pixels values go out of bounds after processing and have effect in blocks. The performance against noise and JPEG compression is not good because of the complete and semi-random nature of the two attacks, respectively. The low pass filtering does not effect the watermark as the watermark is inserted in middle frequencies and all error correcting schemes encounter minimal errors even for the large filter size of 9×9, which significantly degrades the quality of an image.

To conclude, we will once again underline the fact that our discrete wavelet transform based color image watermarking scheme is very useful itself to protect the watermark. It uses middle frequencies and local image characteristics; this makes it robust to a certain extent against low pass filter attack and JPEG compression. The error correcting codes add further robustness to the cases where the scheme is not inherently robust, as shown during the course of this article. Our results show that some codes are better than others for different types of attacks. Each attack modifies the image in a different

way depending on the characteristics of the image. The error correcting codes can correct certain error types and depending upon their own construction the error correcting codes could be beneficial in the case of burst or random noise. Based on the above, it is inferred that for a precise scenario and thus associated with a given restricted set of attacks, we can choose the adapted error correcting codes with our wavelet transform based color image watermarking scheme.

In future study, we will study the invisibility issue of the watermarking algorithm. The use of the image characteristics have a very significant role to play in determining the robustness performance of a particular watermarking scheme and the method used to enhance the robustness or invisibility of the inserted watermark. The actual study is dedicated to the efficient insertion of the watermark inside an image depending upon the characteristics of the image, hence increasing the robustness and reducing the degradation of image quality due to watermark insertion.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Laboratory SIC-XLIM, University of Poitiers, bat. SP2MI, av. Marie et Pierre Curie, 86960, Chasseneuil Cédex, France. [2]MLaboratory DMI-XLIM, University of Limoges, 123, av. A. Thomas, 87060, Limoges, France.

Received: 20 January 2012 Accepted: 16 January 2013
Published: 21 February 2013

References

1. M Sudan, Decoding of Reed-Solomon codes beyond the error correction bound. J. Complex. **12**, 180–193 (1997)
2. J Darbon, B Sankur, H Maitre, Error correcting code performance for watermark protection. Secur. Watermark. Multimedia Contents 3 SPIE. **4314**, 663–672 (2001)
3. N Terzija, M Repges, K Luck, W Geisselhardt, in *Visualization, Imaging, and Image Processing*. Digital image watermarking using discrete wavelet transform: performance comparison of error correcting codes (Acta press, 2002)
4. S Baudry, J Delaigle, B Sankur, B Macq, H Maitre, Analyses of error correction strategies for typical communication channels in watermarking. Signal Process. **81**, 1239–1250 (2001)
5. A Parisis, P Carre, C Fernandez-Maloigne, N Laurent, Color image watermarking with adaptive strength of insertion. IEEE ICASSP. **3**, 85–88 (2004)
6. V Guruswami, M Sudan, Improved decoding of Reed-Solomon codes and algebraic geometry codes. IEEE Trans. Inf. Theory. **45**, 1755–1764 (1999)
7. W Abdul, P Carre, P Gaborit, in *Proceedings of SPIE Media Forensics and Security XII*, vol. 75410U. Human visual system-based color image steganography using the contourlet transform, (2010)
8. W Abdul, P Carre, H Saadane, P Gaborit, in *IEEE ICIP*. Watermarking using multiple visual channels for perceptual color spaces, (2010), pp. 2597–2600
9. R van Schyndel, A Tirkel, C Osborne, in *International Conference on Image Processing*, vol. 2. A digital watermark, Austin Texas, 1994), pp. 86–90
10. R Wolfgang, E Delp, in *International Conference on Images Processing*. A watermark for digital images, Lausanne, 1996), pp. 219–222
11. A Tirkel, G Rankin, R Van Schyndel, W Ho, N Mee, C Osborne, Electronic watermark. Digital Image Comput. Technol. Appl. (DICTA93), 666–673 (1993)
12. J Liu, S Chen, Fast two-layer image watermarking without referring to the original image and watermark. Image Vision Comput. **19**(14), 1083–1097 (2001)
13. M Queluz, Spatial watermark for image content authentication. J. Electron. Imag. **11**, 275 (2002)
14. A De Rosa, M Barni, F Bartolini, V Cappellini, A Piva, in *Information Hiding*, vol. 3657. Optimum decoding of non-additive full frame DFT watermarks, San Jose, 2000), pp. 159–171
15. E Ganic, SD Dexter, AM Eskicioglu, Eskicioglu, in *Proceedings of SPIE: Security, Steganography, and Watermarking of Multimedia Contents VII*, vol. 5681. Embedding multiple watermarks in the DFT, domain using low and high frequency bands, (2005), pp. 175–184
16. X Kang, J Huang, Y Shi, Y Lin, A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression. 13. **8**, 776–786 (2003)
17. V Solachidis, L Pitas, Circularly symmetric watermark embedding in 2-D DFT domain. IEEE Trans. Image Process. **10**(11), 1741–1753 (2001)
18. F Alturki, R Mersereau, in *Proceedings of the Acoustics, Speech, and Signal Processing, 2000 on IEEE International Conference*, vol. 04. An oblivious robust digital watermark technique for still images using DCT phase modulation, Washington, 2000), pp. 1975–1978
19. A Briassouli, P Tsakalides, A Stouraitis, *Hidden messages in heavy-tails: DCT-domain watermark detection using alpha-stable models*, vol. 7, (2005)
20. JR Hernndez, M Amado, F Prez-Gonzlez, DCT-domain watermarking techniques for still images: detector performance analysis and a new structure. IEEE Trans. Image Process. **9**(1), 55–68 (2000)
21. S Lin, C Chin, A robust DCT-based watermarking for copyright protection. IEEE Trans. Consumer Electron. **46**, 415–421 (2000)
22. Z Zhuancheng, Z Dianfu, Y Xiaoping, A robust image blind watermarking algorithm based on adaptive quantization step in DWT. J. Image Graph. **11**(6), 840–847 (2006)
23. S Agreste, G Andaloro, D Prestipino, L Puccio, An image adaptive, wavelet-based watermarking of digital images. J. Comput. Appl. Math. **210**(1–2), 13–21 (2007)
24. M Barni, F Bartolini, A Piva, Improved wavelet based watermarking through pixel-wise masking. IEEE Trans. Image Process. **10**, 783–791 (2001)
25. Z Fan, Z Hongbin, in *EUSIPCO. Conference*. Wavelet domain watermarking capacity analysis, (Vienna, 2004), pp. 1469–1472
26. J Wang, G Liu, Y Dai, J Sun, Z Wang, S Lian, Locally optimum detection for Barni's multiplicative watermarking in DWT domain. Signal Process. **88**, 117–130 (2008)
27. X Xia, C Boncelet, G Arce, Wavelet transform based watermark for digital images. Optics Exp. **3**(12), 497–511 (1998)
28. A Piva, M Barni, F Bartolini, in *Proc. of SPIE Mathematics of Data/Image Coding, Compression, and Encryption*, vol. 3456, ed. by Schmalz. Copyright protection of digital images by means of frequency domain watermarking, (San Diego, 1998), pp. 25–35
29. F Bartolini, M Barni, V Cappellini, A Piva, in *Proceedings of 5th IEEE International Conference on Image Processing ICIP*, vol. I. Mask building for perceptually hiding frequency embedded watermarks, (Chicago, 1998), pp. 450–454
30. F Autrusseau, PL Callet, A robust watermarking technique based on qunatization noise visibility thresholds. Signal Process. **87**(6), 1363–1383 (2007)
31. A Piva, M Barni, F Bartolini, V Cappellini, in *IEEE International Conference on Image Processing*. Dct-based watermark recovering without resorting to the uncorrupted original image, (Santa Barbara, 1997), pp. 520–523
32. S Pereira, T Pun, Robust template matching for affine resistant image watermarks. IEEE Trans. Image Process. **9**, 1123–1129 (2000)
33. DD Vleeschouwer, CD Vleeschouwer, B Macq, Watermarking algorithm based on a human visual model. Signal Process. **66**, 319–335 (1998)
34. A Piva, M Barni, F Bartolini, V Cappellini, in *IEEE International Conference on Image Processing*. DCT-based watermark recovering without resorting to the uncorrupted original image, (Santa Barbara, 1997), pp. 520–523
35. H Qiang, M Hong, *Blind watermark algorithms based on HVS in DCT domain*, vol. 3, (2005)
36. L Xudong, *Blocked DCT and quantization based blind image watermark algorithm*, vol. 21, (2006)
37. W Jin-wei, D Yue-wei, W Zhi-quan, *Blind watermark scheme replacing middle frequency coefficients in DCT domain*, (2005)
38. X Cong, A new blind watermark embedding detection scheme based on DCT, vol. 2, (2004)

39. L Chen, M Li, in *7th World Congress on Intelligent Control and Automation*. An effective blind watermark algorithm based on DCT, (Chongqing, 2008), pp. 6822–6825
40. X Li, *Blocked DCT quantization based blind image watermark algorithm*, vol. 32, (2006)
41. M Jun, S Jiang-hai, *A blind watermark embedding detection scheme based on DCT apply to core image*, vol. 4, (2006)
42. Q Yuan, H Yao, W Gao, S Joo, in *Proceedings 2002 IEEE International Conference on Multimedia and Expo, 2002. ICME'02*, vol. 2. Blind watermarking method based on DWT middle frequency pair, (Lausanne, 2002), pp. 473–476
43. L Ting, Y Weiyan, *A digital watermarking technique for color images based on DWT and HVS*, (2003)
44. Y Zhang, Blind watermark algorithm based on HVS and RBF neural network in DWT domain. WSEAS Trans. Comput. **8**, 174–183 (2009)
45. S Joo, Y Suh, J Shin, H Kikuchi, S Cho, A new robust watermark embedding into wavelet DC components. ETRI J. **24**(5), 401–404 (2002)
46. G El-Taweel, H Onsi, M Samy, M Darwish, Secure and non-blind watermarking scheme for color images based on DWT. GVIP Special Issue Watermarking. **5**, 1–5 (2007)
47. RW Hamming, Error detecting and error correcting codes. Bell Syst. Tech. J. **29**(2), 147–160 (1950)
48. R Bose, D Ray-Chaudhuri, On a class of error correcting binary group codes. Inf. Control. **3**, 68–79 (1960)
49. IS Reed, G Solomon, Polynomial codes over certain finite fields. SIAM J. Appl. Math. **8**, 300–304 (1960)
50. J Wang, G Wiederhold, in *Proceedings of SPIE*, vol. 3528. WaveMark: digital image watermarking using Daubechies' wavelets and error correcting coding, (Boston, 1998), pp. 432–439
51. S Pereira, T Pun, in *Information Hiding*, vol. 1768. Fast robust template matching for affine resistant image watermarks, (Dresden, 2000), pp. 199–210
52. B Verma, S Jain, D Agarwal, A Phadikar, A new color image watermarking scheme. Infocomp. J. Comput. Sci. **5**(2), 37–42 (2006)
53. M Schlauweg, D Prufrock, E Muller, in *Proceedings of the 9th international conference on Information hiding*, vol. 4567. Soft feature-based watermark decoding with insertion/deletion correction, (Saint Malo, 2007), pp. 237–251
54. B Sklar, *Digital Communications: Fundamentals and Applications*. (Prentice Hall, Upper Saddle River, 2001)
55. P Elias, *List decoding for noisy channels. Tech. rep., Technical Report 335*. (Research Laboratory of Electronics, MIT, 1957)
56. A Silverberg, J Staddon, J Walker, *Efficient traitor tracing algorithms using list decoding*, (2001). http://eprint.iacr.org/2001/016
57. J Justesen, T Høholdt, *A course in error-correcting codes*, (2000)
58. P Gaborit, O Ruatta, Efficient interpolation for algebraic list decoding. IEEE Int. Symp. Inf. Theory ISIT, 143–147 (2006)

doi:10.1186/1687-417X-2013-1
Cite this article as: Abdul *et al.*: **Error correcting codes for robust color wavelet watermarking.** *EURASIP Journal on Information Security* 2013 **2013**:1.