

# Approaching the capacity limit in image watermarking: a perspective on coding techniques for data hiding applications<sup>☆</sup>

Fernando Pérez-González<sup>a,\*</sup>, Juan R. Hernández<sup>b</sup>, Félix Balado<sup>a</sup>

<sup>a</sup>*Dept. Tecnologías de la Comunicaciones. ETSI Telecom., Universidad de Vigo, 36200 Vigo, Spain*

<sup>b</sup>*Lysis, S.A. Cotes de Montbenon, 1003 Lausanne, Switzerland*

Received 15 April 2000; received in revised form 31 October 2000

## Abstract

An overview of channel coding techniques for data hiding in still images is presented. Use of codes is helpful in reducing the bit error probability of the decoded hidden information, thus increasing the reliability of the system. First, the data hiding problem is statistically modeled for the spatial and DCT domains. Then, the benefits brought about by channel diversity are discussed and quantified. We show that it is possible to improve on this basic scheme by employing block, convolutional and orthogonal codes, again giving analytical results. It is shown that the use of superimposed pulses does not produce any benefit when applying them to data hiding. The possibility of using codes at the ‘sample level’ (that is, without repeating every codeword symbol) is introduced and its potential analyzed for both hard- and soft-decision decoding. Concatenated and turbo coding are also discussed as ways of approaching the hidden channel capacity limit for the sample level case. Finally, experimental results supporting our theoretical approach are presented for some cases of interest. © 2001 Elsevier Science B.V. All rights reserved.

*Keywords:* Digital communications; Digital watermarking; Capacity; Channel coding; Turbo codes

## 1. Introduction

The increasing growth of electronic commerce has fostered a huge research effort in techniques for preventing illegal uses of digital information, espe-

cially by means of cryptography and watermarking. Watermarking systems insert an imperceptible and secret signal, called watermark, into the object that it is intended to protect, which in our case will be a still image. Research in digital watermarking techniques has experienced two different (and somehow overlapped) phases: in the first years, a plethora of methods were proposed relying on psychovisual techniques but paying little attention to effective ways of detecting and/or extracting the watermark and lacking analytical methods for assessing their performance. In the second phase, recently started, watermarking has been likened to digital communications and from here, a set of

<sup>☆</sup>This work has been partially supported by Xunta de Galicia under project PGIDT99 PX132203B and by EC under project CERTIMARK, IST-1999-10987.

\* Corresponding author.

*E-mail addresses:* fperez@tsc.uvigo.es (F. Pérez-González), juanra@caramail.com (J.R. Hernández), fiz@tsc.uvigo.es (Félix Balado).

analysis and synthesis tools, generally resting on statistical bases, is emerging.

In this paper we concentrate on the so-called data hiding problem, that refers to embedding and retrieval of hidden information in a given image with the highest possible reliability. The creation of this hidden channel proves to be extremely important in many commercial applications where identifiers for the copyright owner, recipient, transaction dates, etc., need to be hidden. We will show how channel coding techniques known for digital communications can be effectively used for significantly improving the performance of data hiding systems and thus approach the capacity limit that Shannon's theorem predicts for this case. We will also briefly comment on the use of these techniques for the so-called detection problem in which one is interested in determining whether a certain image has been watermarked with a certain key. An in-depth discussion of this topic can be found in [14].

The use of channel coding in data hiding was first proposed in [12] for the spatial domain. Since then, new schemes have been published, some lacking a theoretical performance analysis, some with proposals than can be improved. Our objective is then to provide a perspective on the different existing possibilities, to analyze and compare them by introducing fair quality measures and to propose new promising schemes adapted from the area of deep-space communications. In the course of our development we will clearly identify the underlying assumptions for the theoretical analysis. Throughout the paper, we will use interchangeably the terms data extraction/decoding and data hiding/transmission.

Of course, it is not possible to include here an exhaustive description of all the watermarking algorithms that employ or admit coding, in particular, we have chosen here the DCT and spatial domains, but most of the results can be extended to other domains such as the discrete wavelet transform (DWT), the fast fourier transform (FFT), etc. The same can be said regarding attacks. We have not considered them here because coding does not help to mitigate those attacks, except for tamper-proofing applications [3]. However, the advantages of coding given here can be still used if watermarking is performed in a more 'robust' do-

main, as it occurs, for instance, with the reportedly good properties of the Fourier–Mellin transform against affine transformations [4]. Regarding distortions or unintentional attacks, design of robust decoders is possible provided that a (mild) statistical characterization of the distortion is available. In any case, the robustness issue constitutes an interesting subject of research that will be not pursued here.

In the following, variables in bold letters represent vectors whose elements will be referenced using the notation  $\mathbf{x} = (x[1], \dots, x[L])$ . An image  $\mathbf{x}$  can also be regarded to as a unidimensional sequence with  $L$  elements defined in the domain in which watermarking is performed. Thus, the elements of vector  $\mathbf{x}$  will be also considered as samples. Note that the particular ordering of these samples is arbitrary and will have no effect on the final theoretical results.

Suppose that we want to hide  $N$  bits of information and  $\mathbf{x}$  has  $L$  elements. Let  $b[i] \in \{\pm 1\}$ ,  $i = 1, \dots, N$  denote a sequence of antipodal symbols obtained by directly mapping the information bits following the rule  $0 \rightarrow -1$ ,  $1 \rightarrow +1$ . These symbols, for convenience arranged in a vector  $\mathbf{b}$ , are hidden in  $\mathbf{x}$  in a secret way depending on a key  $K$  that is known only to the copyright owner and to the authorized recipient. The information symbols are used to construct a watermark  $\mathbf{w}$  that is added to  $\mathbf{x}$  to produce a watermarked image that conceals the secret information.<sup>1</sup> At the recipient side, the objective will be to extract this information with the highest possible fidelity, so an adequate performance measure is the probability of bit error, that will be introduced later.

The model we are following for generation of the watermark is represented in Fig. 1.

First, a sequence  $\mathbf{s}$  is produced by a pseudorandom generator initialized to a state depending on the key  $K$ . This i.i.d. sequence is constrained to have zero mean and unit variance at each sample, that is,  $E\{s[i]\} = 0$ ,  $E\{s^2[i]\} = 1$  for all  $i = 1, \dots, L$ . In order to guarantee invisibility, the sequence  $\mathbf{s}$  is multiplied by a *perceptual mask*  $\alpha$  which results

<sup>1</sup> Non-additive watermarking schemes will not be considered in this paper.

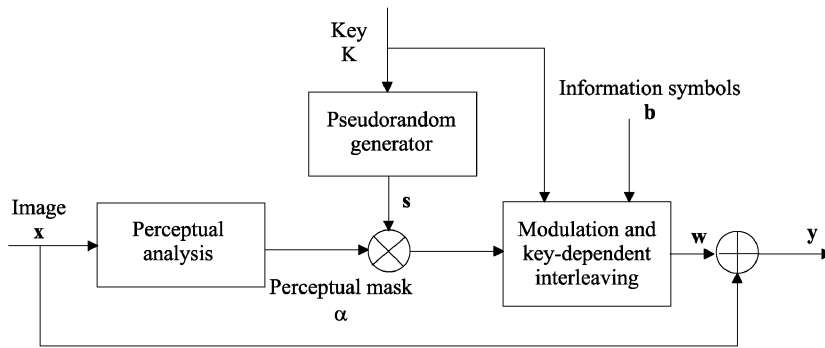


Fig. 1. Model for watermark insertion.

after analyzing the original image with a psychovisual model which takes into account the different sensitivity of the human visual system (HVS) to alterations in the different elements of  $\mathbf{x}$ . Then,  $\alpha^2[i]$  can be seen as the peak energy constraint that the watermark must satisfy at the  $i$ th sample for being imperceptible. After  $s$  and  $\alpha$  are multiplied, the resulting sequence modulates the information symbols  $\mathbf{b}$  in a way that will be detailed in subsequent sections and that may include channel coding. The product sequence is the watermark that is scrambled in a key-dependent manner (see Section 2) before adding it to  $\mathbf{x}$  to produce a watermarked image  $\mathbf{y}$ .

The perceptual mask is computed differently depending on the domain chosen for watermarking and on the specific properties of the HVS that are being taken into account. The details on how  $\alpha$  is evaluated in the spatial and DCT domains are out of the scope of the paper and can be found in [8,13]; we simply mention that the use of coding and the methodology for performance analysis proposed here can be readily extended to other watermarking schemes in these or other domains. This is an important advantage of providing a general framework.

More relevant for our study is the issue of the availability of statistical models characterizing the image  $\mathbf{x}$ . In the DCT domain, several authors have recently proposed to approximate the AC coefficients by independent random variables following a generalized Gaussian probability density function (pdf), given by the expression

$$f_x(x[i]) = A[i]e^{-|\beta[i]x[i]|^{\lambda}}, \quad (1)$$

where both  $A[i]$  and  $\beta[i]$  can be expressed as a function of  $\lambda$  and the standard deviation  $\sigma[i]$

$$\beta[i] = \frac{1}{\sigma[i]} \left( \frac{\Gamma(3/\lambda)}{\Gamma(1/\lambda)} \right)^{1/2}, \quad A[i] = \frac{\beta[i]\lambda}{2\Gamma(1/\lambda)}. \quad (2)$$

In (1) the standard deviation  $\sigma$  is allowed to vary from sample to sample to permit a more flexible modeling of the coefficients at different frequencies. The use of this statistical characterization was independently proposed in [1,11] to derive optimal extraction functions in the DCT domain and has proven to be an invaluable help in the design of hidden information decoders, especially when redundancy is introduced by means of channel coding. On the other hand, some domains do not admit such characterization, so the design of extraction functions cannot rest on statistical grounds and has to resort to a more ‘heuristic’ design. Such is the case of watermarking in the spatial domain, but also it will be so in other less studied domains until researchers come up with good statistical models. However, it should be stressed that this lack of models does not prevent us from improving the performance of data hiding systems with signal processing techniques. For instance, we have shown in [13] that it is possible to obtain a Wiener estimate of the watermark which produces significantly better results.

With all these considerations in mind and trying to provide an overview of the advantages and limitations of channel coding techniques for data hiding purposes, we have chosen three scenarios: (1) *Unfiltered spatial*, i.e., data hiding in the spatial

domain; (2) *Wiener filtered*, i.e., same as (1) but with a Wiener estimate of the watermark and (3) *DCT domain*.

It is appropriate to mention the recent proposal [3,16] of what could be termed as ‘deterministic methods’ as opposed to the probabilistic approach taken here. Deterministic data hiding takes advantage of perfect knowledge of the channel (image) at the transmitter side. With a known (although possibly key-dependent) decoding function, information is hidden in such a way that it is later decoded as desired while meeting the perceptual constraints. The design of the deterministic decoding function divides the perceptually reachable region into subregions that are decoded differently (typically, two subregions, corresponding to a single bit). Thus, in principle, with no distortions present these methods can achieve zero probability of error. Unpublished results reveal, however, a rapid degradation of performance as the energy of the distortion increases; on the other hand, the same distortions have much less impact on probabilistic methods, the reason being that distortions that do not alter the perceptual contents of the image barely alter its pdf, so the structures derived for the undistorted case are still valid [18].

The paper is organized as follows. In Section 2 we present the diversity method that is used for subsequent coding schemes. Thus, in Section 3 both hard- and soft-decision decoders are analyzed for block codes and in Section 4 convolutional codes are studied. These two sections form the basis for the analysis of orthogonal codes and superimposed pulses given in, respectively, Sections 5 and 6. In Section 7 the diversity assumption is dropped and coding at the sample level is introduced. Section 8 is devoted to briefly discuss the use of coding for the detection problem. Finally, Section 9 presents experimental results and comparisons.

## 2. Watermarking with diversity

A very simple way of hiding information would be to alter each sample of  $\mathbf{x}$  in an amount with magnitude  $\alpha[i]$  and sign depending on the hidden symbol, so that the perceptual constraint is met. In this way, the watermark would be obtained as

$w[i] = b[i]\alpha[i]s[i]$ ,  $i = 1, \dots, L$ , so  $L$  bits of information could be conveyed. Unfortunately, such a simple scheme would result useless because generally  $\alpha[i] \ll x[i]$ , and consequently there would be a large probability of error for each hidden bit.

It is possible then to increase the signal to noise ratio (SNR) that ‘sees’ each information bit by repeating it at different samples of  $\mathbf{x}$ . This idea closely resembles diversity techniques used in digital communications over fading channels [19]. Therefore, we will also use the term *diversity* to identify this approach. Obviously, if  $N$  bits are to be transmitted over  $L$  samples, each bit will be hidden at an average of  $L/N$  samples.<sup>2</sup> In order to maximize the uncertainty associated to each bit and make the system robust against certain types of attacks (e.g., cropping), the set of  $L$  available samples is partitioned into  $N$  subsets that we will denote by  $\{\mathcal{S}_i\}_{i=1}^N$  and which are non-overlapping, i.e.,  $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ ,  $\forall i \neq j$ . The particular partition that is used depends on the key  $K$  and is uniformly chosen from the set of all possible partitions, here denoted by  $\mathcal{T}$ . Again, the idea of spreading the information throughout the image has its counterpart in digital communications where it is known as *interleaving*.

The generated watermark  $\mathbf{w}$  (see Fig. 1) is then

$$w[j] = b[i]\alpha[j]s[j], \quad j \in \mathcal{S}_i, \quad (3)$$

where  $b[i]$ ,  $i = 1, \dots, N$  are the information symbols.

In this section we will consider the set  $\mathcal{B}$  of possible information words  $\mathbf{b}_l$ ,  $l = 1, \dots, 2^N$ , that are obtained by combining the  $N$  information symbols in every possible way. Clearly,  $\mathbf{b}_l = (b[1], \dots, b[N])$ , where  $l = 1, \dots, 2^N$ . Then, the data extraction problem can be posed as: find the information word  $\mathbf{b} \in \mathcal{B}$  that has been hidden in the image. If we want to extract the hidden information in an optimal way, we need to know the pdf of  $\mathbf{y}$  conditioned on a certain key  $K$  and on a certain information word. The optimal maximum

<sup>2</sup> From now on, we will neglect possible ‘border effects’ due to non-integer quotients.

likelihood (ML) decoder would decide  $\hat{\mathbf{b}} \in \mathcal{B}$ , such that

$$\hat{\mathbf{b}} = \arg \max_{l=1, \dots, 2^N} \{f_y(\mathbf{y}|\mathbf{b}_l, K)\}, \quad (4)$$

where  $f_y(\cdot)$  is the pdf of  $\mathbf{y}$ .

Alternatively, (4) can be solved by maximizing the log-likelihood ratio between transmitted code-words, i.e.,

$$\hat{\mathbf{b}} = \mathbf{b}_l \in \mathcal{B} : \log \frac{f_y(\mathbf{y}|\mathbf{b}_l, K)}{f_y(\mathbf{y}|\mathbf{b}_m, K)} > 0, \quad \forall m \neq l, \quad (5)$$

where  $\log(\cdot)$  is the natural logarithm function.

As it was discussed in the Introduction, sometimes the distribution of  $\mathbf{y}$  is known or can be accurately estimated with parametric methods. This is, in fact, the case of watermarking in the DCT domain. However, in the spatial domain statistical models are not available and one has to resort to an extraction function based on the cross-correlation between  $\mathbf{y}$  and  $\mathbf{w}$  as was discussed in [10] with some detail. We will also assume that the extracting device is able to exactly compute the perceptual mask  $\alpha$ . Although this is obviously not the case in practice, experimental results show that the error made when estimating the mask is very small. In any case, taking these errors into account constitutes an open line of research.

### 2.1. Equivalent channel parameters

In the DCT case we have discussed in [8] that (5) can be solved by computing a set of sufficient statistics  $\mathbf{r} = (r[1], \dots, r[N])$  that reduce the number of dimensions of the problem from  $L$  down to  $N$ . In this case,  $\mathbf{r}$  takes the form

$$r[i] = \sum_{j \in \mathcal{S}_i} \frac{|y[j] + \alpha[j]s[j]|^2 - |y[j] - \alpha[j]s[j]|^2}{\sigma[j]^2} \quad (6)$$

for all  $i \in \{1, \dots, N\}$ .

In the unfiltered spatial and Wiener scenarios the cross-correlation between the watermarked image and the watermark results in a set of statistics that is by no means sufficient but has proven to yield good results [10]. In the first case, the elements of

$\mathbf{r}$  are computed as follows:

$$r[i] = \sum_{j \in \mathcal{S}_i} y[j]\alpha[j]s[j] \quad (7)$$

for all  $i \in \{1, \dots, N\}$ . For the Wiener case,  $y[j]$  in (7) is replaced by an estimate of the watermark  $\hat{w}[j]$ . Note that when (6) is specialized to  $\lambda = 2$  and  $\sigma[j] = \sigma$  for all  $j$ , a scaled version of (7) results. Thus, the cross-correlation decoder can be considered as optimal in the ML sense if the image  $\mathbf{x}$  follows a Gaussian distribution.

Interestingly, for the three scenarios under analysis, the decoding problem is tantamount to the following bit-by-bit hard decisor:

$$\hat{b}[i] = \text{sgn}(r[i]), \quad i = 1, \dots, N. \quad (8)$$

Once the data extraction structures have been derived, it is possible to analyze their performance. For the three examined models we will compute the bit error probability ( $P_b$ ) assuming that the original image  $\mathbf{x}$  is fixed and the key  $K$  is the only random variable in the system. This choice is justifiable from the point of view that different images reveal quite different performance results; on the other hand, it is reasonable to average the results over the whole set of keys since the key election/assignment should be random<sup>3</sup> and different keys will produce a different number of errors. The election of the key  $K$  affects the watermarking system in two ways: first, in the generation of the sequence  $\mathbf{s}$  and second in the partition that produces the sets  $\mathcal{S}_i$ . Then, for the analysis below, it is useful to introduce two independent random variables:  $\mathbf{s}$ , with pdf  $f_s(\mathbf{s})$  that determines the sequence used, and  $T$ , with pmf (probability mass function)  $P_T(T)$ , that selects the partition used to generate the sets  $\mathcal{S}_i$ ,  $i = 1, \dots, N$ .

As we will see shortly, evaluation of  $P_b$  requires statistical knowledge of the pdf  $f_i(\mathbf{r}|\mathbf{x})$ . In particular, we will be interested in determining the marginal pdf's of  $r[i]$ ,  $i = 1, \dots, N$ , while keeping in mind that, for a fixed  $\mathbf{x}$ , each  $r[i]$  is a function of the

<sup>3</sup>We will not deal here with a possible key-expurgation scheme that discards 'bad' keys; this would make the subsequent analysis more cumbersome and in any case would be questionable from the point of view of maximizing the uncertainty in the watermarking system.

random variables  $\mathbf{s}$  and  $T$ . When the order of diversity is high, that is, when the number of samples per information bit  $L/N$  is large, the central limit theorem guarantees that  $f_r(\mathbf{r}|\mathbf{x})$  will be reasonably well approximated by a Gaussian pdf whose first and second order moments can be analytically calculated. This will be achieved in two steps: first, we will fix the partition  $T$  and determine the first and second order moments of  $f_r(\mathbf{r}|\mathbf{x}, T)$  considering that  $\mathbf{s}$  is the only random variable. In the second step,  $f_r(\mathbf{r}|\mathbf{x})$  is simply calculated as

$$f_r(\mathbf{r}|\mathbf{x}) = \sum_{T \in \mathcal{T}} f_r(\mathbf{r}|\mathbf{x}, T) P_T(T). \quad (9)$$

Let  $b[i]$  be the  $i$ th transmitted information symbol. Then, for a fixed partition  $T$ ,  $r[i]$  has a Gaussian conditional pdf with mean  $b[i]a[i]$  and variance  $\gamma^2[i]$ , with  $a[i]$  and  $\gamma^2[i]$  as given below. Note that the conditioning upon  $T$  and  $\mathbf{x}$  is not explicitly shown for the sake of clarity in the notation. Also note that, from the i.i.d. property in the sequence  $\mathbf{s}$  it follows that the random variables  $r[i]$  are mutually independent.

For the spatial (both unfiltered and Wiener filtered) and DCT cases, we have the following expressions for  $a[i]$  and  $\gamma[i]$  (see [8,13]):

$$a[i] = \sum_{j \in \mathcal{S}_i} q_0[j], \quad (10)$$

$$\gamma^2[i] = \sum_{j \in \mathcal{S}_i} q_v[j], \quad (11)$$

where in the unfiltered spatial case:

$$\begin{aligned} q_0[j] &= \alpha^2[j], \\ q_v[j] &= \alpha^2[j](x^2[j] + (E\{s^4\} - 1)\alpha^2[j]) \end{aligned} \quad (12)$$

and for the Wiener filtered spatial case:

$$q_0[j] = \frac{\alpha^2[j]}{\hat{\sigma}_x^2[j] + \alpha^2[j]} \left( \frac{P-1}{P} \right), \quad (13)$$

$$\begin{aligned} q_v[j] &= \frac{\alpha^6[j]}{(\hat{\sigma}_x^2[j] + \alpha^2[j])^2} \left[ \left( \frac{P-1}{P} \right)^2 \alpha^2[j] (E\{s^4\} - 1) \right. \\ &\quad \left. + (x[j] - \hat{m}_x[j])^2 + \frac{1}{P^2} \sum_{l \in \mathcal{E}_j} \alpha^2[l] \right], \end{aligned} \quad (14)$$

where  $\hat{\sigma}_x^2[j]$  and  $\hat{m}_x[j]$  are local two-dimensional estimates of, respectively, the variance and the

mean of  $\mathbf{x}$  and  $\mathcal{E}_j$  denotes a square-shaped neighborhood of  $P$  image samples around  $j$ , excluding  $j$  itself.

Finally, for the DCT case with a pseudorandom sequence  $\mathbf{s}$  that takes at each sample the values  $+1$  and  $-1$  with equal probability, we have

$$\begin{aligned} q_0[j] &= \frac{(\sigma[j])^{-\lambda}}{2} [(|x[j]| + 2\alpha[j])^\lambda \\ &\quad + ||x[j]| - 2\alpha[j]|^\lambda] - |x[j]|^\lambda, \end{aligned} \quad (15)$$

$$\begin{aligned} q_v[j] &= \frac{(\sigma[j])^{-2\lambda}}{4} [(|x[j]| + 2\alpha[j])^\lambda \\ &\quad - ||x[j]| - 2\alpha[j]|^\lambda]^2. \end{aligned} \quad (16)$$

In all three cases, it can be shown that, when averaged over all possible partitions  $T \in \mathcal{T}$ , the marginal pdf's of each  $r[i]$  will follow identical Gaussian distributions which are approximately independent and whose respective means and variances are:

$$a = \frac{1}{Z} \sum_{j=1}^L q_0[j], \quad (17)$$

$$\gamma^2 = \frac{1}{Z} \sum_{j=1}^L \left[ q_v[j] + \frac{Z-1}{Z} q_0^2[j] \right], \quad (18)$$

where  $q_0[j]$  and  $q_v[j]$ ,  $j = 1, \dots, L$  were defined in Eqs. (10)–(16) for the different scenarios under consideration and  $Z$  is the number of sets  $\mathcal{S}_i$  in which the available samples are partitioned. Note that for the moment  $Z = N$ , but we will deal later with other values of  $Z$  when coding is introduced.

We will also find useful to define a quantity called *per sample signal to noise ratio* (PSSNR) as

$$\text{PSSNR} \triangleq \frac{(\sum_{j=1}^L q_0[j])^2}{L \sum_{j=1}^L (q_v[j] + q_0^2[j])} \simeq \frac{a^2 Z}{\gamma^2 L}, \quad (19)$$

which is just the total available SNR divided by the number of available samples. Note that the PSSNR highly depends on the original image and also on the particular method and domain chosen for data hiding. By differentiating (19) with respect to  $\alpha[i]$  it is straightforward but tedious to show that for the unfiltered and Wiener filtered spatial cases the PSSNR increases with  $\alpha[i]$ . Consequently, for these cases, making the watermark peak energy higher

will yield better performance at the price of increasing the visibility of the watermark. Surprisingly, this is not always the case for DCT-domain watermarking. It can be shown that if  $\lambda \leq 1$  and  $\alpha[i] < |x[i]|$ , then the PSSNR may decrease with increasing  $\alpha[i]$ . Therefore, increasing the watermark peak energy in the DCT domain does not necessarily produce better results.

### 2.2. Computation of the bit error probability

Let  $P_b(\mathbf{b}_j)$  denote the bit error probability when word  $\mathbf{b}_j$  is transmitted. Then, for a given image  $\mathbf{x}$ , it is possible to derive an expression for  $P_b(\mathbf{b}_j)$  as follows. Assume that  $b_j[i] = +1$ , for all  $i = 1, \dots, N$ , then recalling that the decoder (ML in the DCT case) is equivalently given in terms of the  $r[i]$  by (8) and using the fact that the components of  $\mathbf{r}$  are jointly independent (and thus an error in symbol  $b_j[i]$  depends exclusively on  $r[i]$ ), it is possible to write

$$P_b(\mathbf{b}_j) = \frac{1}{N} E_T \left\{ \sum_{i=1}^N \int_{r[i] < 0} f(r[i] | T) dr[i] \right\} \\ = \frac{1}{N} \sum_{i=1}^N \int_{r[i] < 0} E_T[f(r[i] | T)] dr[i], \quad (20)$$

where conditioning of  $f(r[i] | T)$  upon  $\mathbf{x}$  is assumed but not explicitly shown.

We have already seen that  $E_T[f(r[i] | T)]$  follows a Gaussian distribution with mean  $a$  and variance  $\gamma^2$ , so we can finally write

$$P_b(\mathbf{b}_j) = Q(a/\gamma). \quad (21)$$

It is immediate to repeat the derivation above for any other information word  $\mathbf{b}_l \in \mathcal{B}$ , to arrive at the same expression as in (21). Then, the bit error probability is independent from the transmitted codeword and we can write

$$P_b = Q(a/\gamma) \simeq Q(\sqrt{\text{PSSNR } L/N}). \quad (22)$$

Note that this independence with the transmitted bit is not surprising since we are averaging over the set of all the possible set partitions. It is also interesting to remark that  $\text{PSSNR } L/N$  can be regarded as the average SNR per hidden information bit; thus, it plays an analogous role to the ratio  $E_b/N_0$  used in digital communications and in the same

way allows a fair comparison between different modulation and coding schemes. In fact, the widely used curves representing  $P_b$  versus  $E_b/N_0$  can also be adopted for data hiding purposes. However, care must be taken when employing these curves, since most of them are suited for passband modulations (ours could be regarded as a baseband case) so a 3 dB difference arises. In addition, Eq. (22) clearly shows how performance increases with the number of samples in  $\mathbf{x}$  (for the same PSSNR) and decreases with the number of hidden bits  $N$ .

### 3. Block coding

Once we have built a basic scheme for reliable information hiding, its performance can be improved by means of coding. Coding is an effective way of reducing the probability of bit error by creating interdependencies between the transmitted symbols at the price of an increased complexity.

We will first deal with block codes and then proceed to describe how convolutional codes can be used for data hiding purposes. Suppose that, instead of transmitting raw information symbols through the hidden channel, we use a  $(n, k)$  block code that maps  $k$  information symbols  $b[i]$ ,  $i = 1, \dots, k$ , into  $n$  binary antipodal channel symbols  $c[i]$ , with  $c[i] \in \{\pm 1\}$ ,  $i = 1, \dots, n$ . From the way it is constructed, it is clear that this code, that we will denote by  $\mathcal{C}$ , consists of a total of  $2^k$  codewords that will be denoted by  $\mathbf{c}_l$ ,  $l = 1, \dots, 2^k$ , each with  $n$  binary antipodal symbols. In order to use this code for data hiding, the set of  $N$  source information bits is divided into  $N/k$  blocks, and each block of size  $k$  bits mapped into  $n$  symbols that are hidden using a procedure for watermark insertion similar to that summarized in (3). Therefore, the watermark is generated in the following way:

$$w[i] = c^{(l)}[j] \alpha[i] s[i], \quad i \in \mathcal{S}_j^{(l)}, \quad j \in \{1, \dots, n\}, \quad (23)$$

where  $\mathbf{c}^{(l)}$ ,  $l \in \{1, \dots, N/k\}$  is the  $l$ th transmitted codeword and  $\mathcal{S}_j^{(l)}$ ,  $l \in \{1, \dots, N/k\}$  are the subsets of samples in which the  $l$ th transmitted codeword is inserted. By construction, it is easy to see that each codeword symbol will be replicated at  $L/N k/n$  different samples.

Regarding watermark extraction, two strategies are possible: hard- and soft-decision decoding, each admitting different implementations and simplifications. An important difference in the treatment of this and following sections with respect to Section 2 is that for the sake of clarity in the exposition (and with no impact in the final results) we will analyze here the transmission of a single codeword, corresponding to  $k$  information bits or  $n$  channel symbols. Thus, superscripts will be dropped from (23).

### 3.1. Hard-decision decoding

In this case, an independent threshold-based decision is taken for each symbol of the transmitted codeword, producing a received word. Then, the codeword with minimum Hamming distance to the received word is chosen. Note that this two-step decoding process is not optimal in the ML sense, but gives good results at a low computational cost, since efficient decoding algorithms are available for certain types of block codes (generally belonging to the class of linear codes).

When trying to assess the performance of hard decoding, one finds that the number of errors  $n(T,s)$  depends, for a fixed partition, in a complicated manner with the sequence  $s$  so this precludes obtaining an exact expression for  $P_b$ . However, useful approximations can be given for many cases of interest, particularly for perfect linear codes [28]. First, instead of the bit error probability, it is simpler to obtain the probability of block error, that is, the probability of incorrectly decoding a certain transmitted codeword  $c_j$ . This probability, here denoted by  $P_c(c_j)$ , can be used to bound the bit error probability  $P_b$  and can be written as

$$P_c(c_j) = \sum_{T \in \mathcal{T}} P_T(T) P_c(c_j|T). \tag{24}$$

In the case of perfect codes, it is possible to write an exact expression for  $P_c(c_j|T)$  in terms of the cross-over probability of the corresponding binary symmetric channel (BSC), that is, the probability of error when sending an arbitrary symbol belonging to the transmitted codeword.

For the remaining of this section (and for certain derivations in the sections to follow) we will make the assumption of i.i.d. parallel channels (IPC) that

states that the equivalent Gaussian channels seen by each of the transmitted antipodal symbols are independent and identically distributed. This implies that  $a[i]$  and  $\gamma^2[i]$  are independent of  $i$ , so we can write  $a(T)$  and  $\gamma^2(T)$  instead (recall that they are still conditioned on a fixed partition  $T$ ). This assumption is reasonable as long as  $L/N$  is large and the sets  $\mathcal{S}_i$  are typical, in the sense that each one gathers contributions from all over the image. In addition, this assumption has been verified experimentally.

Since for the three scenarios under analysis each codeword symbol sees a Gaussian channel, we can write

$$p(T) = Q(a(T)/\gamma(T)), \tag{25}$$

where  $p(T)$  is the cross-over probability in the BSC that results after adopting the IPC assumption for a given partition  $T$ . For the general non-perfect case, there are upper bounds to  $P_c(c_j|T)$  again available in terms of  $p(T)$ . These bounds have the form

$$P_c(c_j|T) \leq (2^k - 1) \sum_{m=t+1}^n \binom{d_{\min}}{m} p(T)^m (1 - p(T))^{n-m}, \tag{26}$$

where  $t \triangleq \lfloor d_{\min}/2 \rfloor$  is the maximum number of bit errors that the code is able to correct and  $d_{\min}$  is the minimum Hamming distance (minimum number of differing antipodal symbols) between any two codewords. For linear codes, we have the Bhattacharyya bound

$$P_c(c_j|T) \leq \sum_{m=2}^{2^k} [4p(T)(1 - p(T))]^{w_m/2}, \tag{27}$$

where  $w_m$  is the Hamming weight (number of antipodal symbols with the value  $+1$ ) of the  $m$ th codeword.

Note, however, that upper-bounding Eq. (24) is cumbersome because of the dependence of  $p(T)$  with  $T$ . Although a formal proof is not available at this moment, we have observed that for the cases of interest the variance of  $a(T)/\gamma(T)$  with  $T$  is small when compared to  $a/\gamma$ , so with this approximation it is straightforward to write

$$p(T) \simeq p = Q\left(\frac{a}{\gamma}\right) \simeq Q\left(\sqrt{\frac{\text{PSSNR } LR}{N}}\right), \tag{28}$$

where  $a$  and  $\sigma$  were defined in Eqs. (17) and (18) and  $R = k/n$  is the so-called code rate. Then, the value of  $p$  obtained from (28) can be plugged into Eqs. (26) or (27) to obtain upper bounds to the probability of block error. Once this probability is available, it is possible to write the bit error probability corresponding to the Bhattacharyya bound as

$$P_b \leq \frac{2^{k-1}}{2^k - 1} \sum_{m=2}^{2^k} [4p(1-p)]^{w_m/2}. \quad (29)$$

A simple yet accurate approximation to  $P_b$  that has been used in the computer simulations presented in Section 9 is [24]

$$P_b \approx \frac{1}{n} \sum_{m=t+1}^n m \binom{n}{m} p^m (1-p)^{n-m}. \quad (30)$$

For bounds suitable for specific types of codes, the reader is recommended to consult a monograph on channel coding, of which [17,28] are excellent examples.

### 3.2. Soft-decision decoding

In this case, a soft-decision decoder, implementing the ML decoder, should seek the codeword  $\hat{c} \in \mathcal{C}$ , that maximizes the probability

$$\hat{c} = \arg \max_{l=1, \dots, 2^k} \{f_y(\mathbf{y}|c_l, K)\}. \quad (31)$$

Alternatively, (31) can be solved by means of the log-likelihood ratio between transmitted codewords, in a similar way to (5). For the DCT case, it follows that the optimal ML decoder should find  $c_l$  such that

$$\sum_{i=1}^n \sum_{j \in S_i} \frac{(|y[j] + \alpha[j]s[j]c_l[i]|^2 - |y[j] + \alpha[j]s[j]c_m[i]|^2)}{\sigma^2[j]} > 0, \quad \forall l \neq m. \quad (32)$$

With the  $r[i]$  as defined in (6) it is possible to show that the ML decoder will decide  $\hat{c} \in \mathcal{C}$  such that

$$\hat{c} = \arg \max_{l=1, \dots, 2^k} \sum_{i=1}^n c_l[i]r[i]. \quad (33)$$

In the spatial domain (both unfiltered and Wiener filtered cases) the lack of statistics for  $\mathbf{y}$  that has been discussed previously precludes using (31).

Alternatively, the cross-correlating decoder that was used in Section 2 can be simply extended to coding and becomes identical to (33) where the  $r[i]$  have been defined in (7). Note that the decisor in (33) is equivalent to minimizing the Euclidean distance between  $c_l$  and  $\mathbf{r}$ . Also note that, although not ML for the unfiltered and Wiener filtered cases, this decisor is soft in the sense that no bit-by-bit hard decisions are taken.

Once we have shown the structure of the soft-decision decoders, we can evaluate their performance. The methodology mimics the approach taken before: for a fixed image  $\mathbf{x}$  and considering  $K$  as the only random variable in the system, we will compute the bit error probability  $P_b$ . Exact computation of  $P_b$  for the soft-decision decoder is extremely involved except for trivial cases. Instead, and as is customary in communications theory, we will obtain the so-called union bound in which the probability of error between two codewords is the fundamental ingredient. To this end, we want to calculate the probability that the decoder decides codeword  $c_2$  when  $c_1$  is sent assuming that these two are the only existing codewords. We will denote this probability by  $P(c_1 \rightarrow c_2)$ . Let

$$h(T, \mathbf{s}) = \sum_{i=1}^n (c_1[i] - c_2[i])r[i]. \quad (34)$$

Then, in the three cases under study it follows from (33) that the decoder will decide  $c_2$  iff  $h(T, \mathbf{s}) < 0$ . Thus, the block error probability for these two codewords can be obtained from the pdf of  $h(T, \mathbf{s})$ . It is straightforward to show that  $h(T, \mathbf{s})$  follows a Gaussian distribution with respective mean and variance:

$$\begin{aligned} E\{h\} &= \sum_{i=1}^n (c_1[i] - c_2[i])a, \\ \text{Var}\{h\} &= \sum_{i=1}^n (c_1[i] - c_2[i])^2 \gamma^2. \end{aligned} \quad (35)$$

Then, the probability of block error for two words is

$$P(c_1 \rightarrow c_2) = Q\left(\frac{d_{1,2}a}{2\gamma}\right), \quad (36)$$

where  $d_{1,2}$  is the Hamming distance between the two codewords;  $a$  and  $\gamma$  were defined in (17) and (18) and now  $Z = (Nn)/k$ .

For linear codes, the probability of block error is independent of the transmitted codeword. Then, assuming that the transmitted codeword is  $c_1$ , the probability of block error  $P_c$  can be bounded as

$$P_c \leq \sum_{l=2}^{2^k} P(c_1 \rightarrow c_l) \\ = \sum_{i=w_{\min}}^n A_i Q\left(\sqrt{\frac{iR \text{PSSNR} L}{N}}\right), \quad (37)$$

where  $R = k/n$  is the rate of the code and  $A_i$  is the so-called weight spectrum, which indicates the number of codewords with Hamming weight  $i$ .

Given the difficulty of soft decoding for block codes, except for some cases (see Section 5), suboptimal decoding algorithms have been proposed such as the one due to Chase [28].

#### 4. Convolutional coding

The main advantage of convolutional codes is their error correction power together with the availability of efficient algorithms that perform soft decoding. Convolutional codes are advantageous over block codes for similar rates. Use of convolutional codes in watermarking applications was first proposed in [20] and has been used in [9] for watermarking of video sequences due to their superior performance properties.

A description of convolutional codes is out of the scope of this paper. We refer the reader to [17,19,27] where a detailed exposition and properties are presented. Here we point out at the existence of the well-known Viterbi algorithm that makes ML decoding feasible for channels with i.i.d. noise.

Implementation of convolutional codes for watermarking applications follows the same lines than for block codes with soft-decision decoding. Given a rate  $R = k/n$  convolutional code, the  $N$  information bits are divided in groups of  $N/k$  symbols that are sequentially introduced in the convolutional encoder. The latter evolves through its state diagram and produces an output in groups of  $n$  antipodal symbols, thus resulting a total of  $M = (Nn)/k$  symbols  $c[i]$ ,  $i = 1, \dots, M$  that are

transmitted through the hidden channel exactly as was described in the previous section. Regarding soft decoding with Viterbi's algorithm, the required metrics can be easily obtained following the discussion for soft-decision decoding of block codes. For the three models we are considering the branch metrics are computed as the Euclidean distance between the vector of  $n$  statistics  $r[i]$  and the vector of  $n$  channel symbols  $c_j[i]$  generated by the convolutional encoder when it follows the  $j$ th state transition.

Most approaches for assessing the performance of convolutional codes resort to the so-called input-output weight enumerator function (IOWEF) [2], which has the form

$$T(W, H) = \sum_{w=d_{\text{free}}}^{\infty} \sum_{h=1}^{\infty} A(w, h) W^w H^h, \quad (38)$$

where  $d_{\text{free}}$  is the minimum Hamming distance between any two output sequences and  $A(w, h)$  is the number of codewords with Hamming weight  $w$  associated with an input word of Hamming weight  $h$ . The weight enumerator function (WEF)  $T(W)$  is obtained from (38) by setting  $H = 1$  and summing over  $h$ . The WEF has the form

$$T(W) = \sum_{w=d_{\text{free}}}^{\infty} A(w) W^w. \quad (39)$$

Both the WEF and IOWEF can be generated in systematic ways amenable for computer implementation, so they can be readily available for a given code.

In order to apply the union bound to the probability of block error in soft decoding of convolutional codes, it is necessary to compute the probability of block error for two-codewords  $P_2$ , which for linear codes will show up as a function of the Hamming distance, that is,  $P_2(w)$ . Then, the probability of block error can be bounded as

$$P_c < \sum_{w=d_{\text{free}}}^{\infty} A(w) P_2(w) = \sum_{w=d_{\text{free}}}^{\infty} A(w) Q\left(\frac{\sqrt{wa}}{\gamma}\right) \quad (40)$$

with  $a$  and  $\gamma$  as defined in (17) and (18) and now  $Z = (Nn)/k$ .

The bit error probability can be obtained in a similar fashion by using the IOWEF, thus

$$P_b < \frac{1}{k} \sum_{w=d_{\text{free}}}^{\infty} \sum_{i=1}^{\infty} iA(w,i)Q\left(\sqrt{\frac{wR \text{PSSNR} L}{N}}\right), \quad (41)$$

where the term  $1/k$  comes from the number of information bits in a codeword. It is also possible to obtain simpler bounds, such as the Bhattacharyya bound. We refer the reader to [27,28] for more details.

### 5. Orthogonal coding

The use of orthogonal codes is another way of improving the performance of the basic diversity method described in Section 2. Although, in principle, orthogonal signals could be transmitted with different schemes, here we will analyze the structure proposed by Kutter [15] that constructs a set of orthogonal codewords, that can be studied within the block coding context although some peculiarities arise. Different ways of generating orthogonal waveforms exist, but we will deal here with Walsh codes that will serve for illustrative purposes. A Walsh code can be regarded to as a  $(2^k, k)$  code in which all the  $n = 2^k$  codewords  $c_l, l = 1, \dots, n$  are orthogonal, in the sense that

$$\sum_{i=1}^n c_l[i]c_j[i] = n\delta_{j,l}, \quad \forall j, l \in \{1, \dots, n\}, \quad (42)$$

where  $\delta_{j,l}$  is the Kronecker delta, that takes the value 1 if  $j = l$  and is zero otherwise.

The codewords in a Walsh code are generated by a simple recursive procedure that is detailed in [26,28]. The use of Walsh codes for data hiding matches the description given for block codes. Note that for  $N$  information bits and  $L$  samples, each codeword symbol will be replicated at  $L/Nk/n$  samples. As with the block codes case, we will analyze the performance for a single transmitted codeword, that produces the same results as for the set of all transmitted codewords.

Although Walsh codes can be seen as block codes, simple soft-decision decoding algorithms are available [26], so here we will analyze only the performance of this type of decoder. A first

approach would be to follow the lines of Section 3 to find the union bound; however, the results so obtained are only accurate for high SNRs. Fortunately, the special structure of the code allows for a semi-analytical expression. Let  $r[i], i = 1, \dots, n$  be the set of statistics (obtained as in (7)) for the first transmitted codeword and assume without loss of generality that this word is the all-ones word, that is  $c_1[j] = +1, j = 1, \dots, n$ . Then, following (33), the soft-decision decoder will compute  $n$  cross-correlations of the form

$$h[j] = \sum_{i=1}^n c_j[i]r[i], \quad \forall j \in \{1, \dots, n\} \quad (43)$$

and decide that  $c_j$  for which  $h[j]$  is largest. Therefore, a correct decision will be made iff  $h[1] > h[j]$  for all  $j > 1$ . With the IPC assumption, for a fixed partition each  $h[j]$  is Gaussian distributed with respective mean and variance

$$E\{h[j]\} = 0, \quad \text{Var}\{h[j]\} = n\gamma^2(T), \quad 1 < j \leq n, \quad (44)$$

while

$$E\{h[1]\} = na(T), \quad \text{Var}\{h[1]\} = n\gamma^2(T), \quad (45)$$

the probability  $P(\text{correct}|T)$  that  $t[1] > t[j]$  for all  $j > 1$ , which in turn is the probability of correct decision, is

$$\begin{aligned} P(\text{correct}|T) &= \int_{-\infty}^{\infty} \int_{-\infty}^{h_1} \dots \int_{-\infty}^{h_1} \frac{e^{-(h_1 - na(T))^2/2n\gamma^2(T)}}{(2n\pi\gamma^2(T))^{1/2}} \\ &\times \left[ \prod_{j=2}^n \frac{e^{-h_j^2/2n\gamma^2(T)}}{(2n\pi\gamma^2(T))^{1/2}} dh_j \right] dh_1. \end{aligned} \quad (46)$$

Now, recalling that  $P(\text{correct}) = E_T\{P(\text{correct}|T)\}$  and noting that what we have in the integrand of (46) is the product of  $n$  independent random variables with the integral limits independent of  $T$ , it is possible to write

$$\begin{aligned} P(\text{correct}) &= \int_{-\infty}^{\infty} \frac{e^{-(h_1 - na)^2/2n\gamma^2}}{(2n\pi\gamma^2)^{1/2}} \\ &\times (1 - Q(h_1/\gamma\sqrt{n}))^{n-1} dh_1, \end{aligned} \quad (47)$$

where  $a$  and  $\gamma^2$  were given in (17) and (18) and  $Z = (Nn)/k$ . From here, the probability of block error is simply  $P_c = 1 - P(\text{correct})$ . Eq. (47) cannot

be evaluated in closed form, although it is quite simple to compute the integral numerically. Alternatively, there exist bounds and approximations [6] for both the low and the high SNR cases. In particular, the probability of block error can be bounded by [27]

$$P_c \leq (n-1)^\rho \exp\left(-\frac{k \text{PSSNR } L \rho}{2N(1+\rho)}\right), \quad (48)$$

where

$$\rho = \sqrt{\frac{\text{PSSNR } L}{N2 \log 2}} - 1 \quad \text{if } \frac{1}{4} \leq \frac{N2 \log 2}{\text{PSSNR } L} \leq 1 \quad (49)$$

and

$$\rho = 1 \quad \text{if } \frac{N2 \log 2}{\text{PSSNR } L} \leq \frac{1}{4}. \quad (50)$$

Regarding the bit error probability, it is straightforward to show [19] that

$$P_b = \frac{n}{2n-2} P_c. \quad (51)$$

It is interesting to remark that when  $\text{PSSNR}(L/N) < 2 \log 2 = 1.4$  dB the bit error probability cannot be made arbitrarily small or, in other words, the total SNR per information bit lies below the asymptotic value of Shannon's Capacity for Gaussian channels [21,25]. This asymptotic value is achieved for an infinite number of channel uses. Alternatively, for a given image with  $L$  available samples and a certain PSSNR, by applying the previous condition we can upper-bound the maximum amount of information bits that could be reliably conveyed. For the case of orthogonal pulses, it is also possible to show [27] that the capacity limit is reached asymptotically when  $n \rightarrow \infty$ ; however, since the number of available samples is limited to  $L$ , the codewords size  $n$  must satisfy  $n \leq L$  so this bound cannot be reached in practice. Furthermore, other channel coding schemes (e.g., convolutional codes) allow us to fall closer to the capacity limit than orthogonal codes. Note that, in any case, we are not claiming that the value given above is the actual capacity bound of a given image because, above all, the physical data hiding channel is not really Gaussian (see Section 7) and the use of diversity and/or interleaving also reduces the true capacity.

It is possible to improve the performance of orthogonal coding by using the so-called biorthogonal codes and simplex codes. For the former, each transmitted codeword is further modulated by a binary antipodal symbol, taking advantage of the fact that sign changing does not destroy orthogonality. Now, there are two classes of error events: errors between orthogonal codewords and errors in the symbol that modulates a certain codeword. Since orthogonal codewords are closer in Euclidean distance than antipodal symbols, it is possible to approximate the performance of biorthogonal codes by concentrating only on error events of the first kind. The analysis closely resembles what has been already presented for orthogonal codes, so we will not repeat it here. We just point out that the use of biorthogonal signals allows to double the number of signals with a performance similar to that of orthogonal codes. We refer the reader to [19,28] where full descriptions of biorthogonal signals are given.

Another slight improvement can be achieved by the use of simplex codes. These signals are not orthogonal but have cross-correlation coefficients that take the value  $-1/(n-1)$ , becoming *equicorrelated*. This allows a reduction in the ratio  $\text{PSSNR } L/N$  of  $10 \log_{10}(n/(n-1))$  dB for the same bit error probability [19]. Nevertheless, this gain approaches zero as  $n$  is increased.

Finally, we mention that some authors have proposed the use of 'spread-spectrum' waveforms for data hiding [4,22]. These modulations may be useful in overloaded multiple access environments, but nothing is gained when a single user is present, which is our case. Moreover, most of them (e.g., Gold or Kasami codes) have non-negligible cross-correlations that would seriously affect performance when compared to orthogonal codes. For this reason, we have decided not to dwell on them here. For a critic comparison between spread-spectrum and other modulations, see [23].

## 6. Modulations with superimposed orthogonal pulses

Next, we turn our attention to a class of modulations in which some codewords corresponding to

different information bits are transmitted superimposed over the same set of samples  $\mathcal{S}_i$ . One of such possibilities was proposed by Csurka et al. [4]. Here, we will adapt their method to the lines of our exposition, but the conclusions and results remain valid.

We will also focus only the unfiltered spatial and Wiener cases. The ML decoder in the DCT domain results prohibitively complex for this modulation since it is not possible to decouple the contributions of all the information bits that are transmitted over the same set of samples, except for the Gaussian (i.e.,  $\lambda = 2$ ) model.

Consider then the following expression for the watermark:

$$w[j] = \sum_{l=1}^k \frac{b_l[l]c_l[j]}{\sqrt{k}} \alpha[j]s[j], \quad j \in \mathcal{S}_i, \quad (52)$$

where the codewords  $c_l$ ,  $l = 1, \dots, k$ , are orthogonal (cf. Eq. (42)) with  $n$  binary antipodal symbols. As in Section 2, the  $b_l[l]$ ,  $l = 1, \dots, k$ , are binary antipodal information symbols. We can see that for this scheme each codeword modulates a different information bit, so if  $L$  is the total number of samples and  $N$  is the number of information bits, then each codeword spans  $n = (Lk)/N$  samples. It is interesting to see that the term  $\sum_{l=1}^k (b_l[l]c_l[j])/\sqrt{k}$ , for large  $k$ , follows a zero-mean unit-variance Gaussian distribution that is independent from  $s$ . This guarantees satisfaction of the perceptual constraint through  $\alpha$ . In this case, decoding is implemented for each transmitted bit by computing the statistic

$$r_i[l] = \sum_{j \in \mathcal{S}_i} y[j] \frac{c_l[j]}{\sqrt{k}} \alpha[j]s[j], \quad \forall l \in \{1, \dots, n\} \quad (53)$$

in the unfiltered spatial case, replacing  $y[j]$  by  $\hat{w}[j]$  in the Wiener scenario. As in previous sections, the estimate of each transmitted bit is simply

$$\hat{b}_i[l] = \text{sgn}(r_i[l]). \quad (54)$$

Let us analyze the performance of this scheme. To this end, we will concentrate on a single set, say  $\mathcal{S}_i$ . It is not difficult to see that, for a fixed assignment of samples to  $\mathcal{S}_i$ , i.e., for a fixed partition, the  $r_i[l]$  will be i.i.d. Gaussian random variables with mean

$b_i[l]a[i]$  and variance  $\gamma^2[i]$  where

$$a[i] = \sum_{j \in \mathcal{S}_i} \frac{q_0[j]}{k}, \quad (55)$$

$$\gamma^2[i] = \sum_{j \in \mathcal{S}_i} \frac{q_v[j]}{k} \quad (56)$$

and the expressions for  $q_0[j]$  and  $q_v[j]$  are identical to those given in (10)–(16) except that the term  $(E\{s^4\} - 1)$  is replaced now by  $(E\{s^4\} - 1/k)$ . Thus, after averaging over all possible partitions, formulas (17) and (18) are still applicable with  $Z = N/k$ , so the bit error probability can be computed.

In order to make a comparison between this scheme and the diversity method presented in Section 2, we first note that the means  $a$  of the equivalent Gaussian channels are identical for the two cases. Regarding the variances, let  $\gamma_1^2$  be the variance for diversity and let  $\gamma_2^2$  be the variance for the superimposed orthogonal case. Then, it is possible to show that

$$\gamma_2^2 - \gamma_1^2 = \frac{\sum_{j=1}^L \alpha^4[j]}{N} \left( \frac{1 + Nk - k^2 - N/k}{N} \right). \quad (57)$$

The two variances are identical for  $k = 1$  (as should be expected, since then the two schemes become the same) and for  $k = N$  (i.e., codewords with size equal to the number of available samples). In these two situations, both schemes provide the same bit error probability.<sup>4</sup> Otherwise, the diversity method reveals a superior performance for the same value of  $E\{s^4\}$ . On the other hand, note that if one wants to maximize the entropy for each sample of the watermark,  $s$  should follow a Gaussian distribution which gives  $E\{s^4\} = 3$ , but this same distribution is achieved (for large  $k$ ) in the case of superimposed orthogonal pulses with a binary antipodal  $s$  for which  $E\{s^4\} = 1$ .

Some final comments are due regarding this method. First, it is straightforward to combine it with the various forms of channel coding that we

<sup>4</sup>This is not surprising since spread-spectrum systems, that also superimpose the transmitted waveforms from different users, do not represent any increase in channel capacity with respect to time division multiple access, as has been discussed for instance in [23].

have presented with the possibility of soft-decision decoding in those cases where it is already feasible (e.g., Walsh or convolutional codes). Unfortunately, extension to data hiding in the DCT domain becomes hard to implement because orthogonality is helpful when the decoder is based on cross-correlations which is not the case in the DCT domain (see (6)). Alternatively, a sub-optimal cross-correlation-based receiver could be implemented but this would result in a significant performance degradation as we have shown in [8]. Finally, for the unfiltered spatial and Wiener cases the computational complexity of the superimposed orthogonal pulses scheme is  $k$  times higher than the corresponding to the diversity method. Considering the discussion above, we conclude that using superimposed orthogonal pulses is not advisable for data hiding purposes.

## 7. Coding at the sample level

All the data hiding schemes that we have considered so far relied in replicating each information bit in different samples of the image in order to gain enough SNR so that a low probability of error could be achieved. This was also the case for coding since each symbol at the output of the encoder was repeated at a number of samples. Diversity can be seen as a simple form of block coding (with only two codewords) for which soft-decision decoding is extremely simple (see Section 2). The good performance achieved by this almost trivial way of coding, also known as repetition coding, opens the door for more sophisticated types of codes that provide additional improvements over the basic scheme. In this section we investigate coding at the sample level, which differs from the approach taken in previous sections (and in most of the prevalent literature) in that each codeword symbol is now transmitted over a single sample. This is tantamount to say that the sets  $\mathcal{S}_i$  in which we partition the image are now one sample wide. Thus, a codeword with  $n$  symbols would require exactly  $n$  samples to be transmitted. The watermarking generation equation is then (23), but as before, we will drop the superscripts since we are interested in analyzing the performance for a single transmitted

codeword. Then

$$w[j] = c[i]\alpha[j]s[j], \quad j \in \mathcal{S}_i, \quad (58)$$

where  $c$  is the transmitted codeword.

We will see that most of the structures already developed can be extended to the sample level; unfortunately, an exact performance analysis turns out to be cumbersome, since now it becomes obvious that it is not possible to resort to the central limit theorem approximation. In this section we will concentrate only on the DCT domain with a Laplacian model, i.e.,  $\lambda = 1$ . The reason for this election is two-fold: first, the existence of a statistical model will allow to obtain analytical results and bounds for  $P_b$ . The lack of good models as in the spatial domain makes this analysis very difficult if not impossible, since averages over the ensemble of samples cannot be taken. Second, the Laplacian is, together with the Gaussian, the simplest in the family of generalized Gaussian models, and it produces relatively accurate results as we have discussed in [8]. It is possible to extend most of the results that will be presented here to other values of  $\lambda$  but sometimes no closed-form expressions exist and use of numerical methods is required.

Unlike the approach taken in Section 2, here we will rely on the statistical characterization of  $x[i]$  and consider the key  $K$  as deterministic. We will also assume that the sequences of both  $\alpha[i]$  and  $\sigma[i]$ ,  $i = 1, \dots, L$ , can be characterized by means of a stochastic process with joint pdf  $f_{\alpha, \sigma}(\alpha, \sigma)$ . Moreover, by virtue of the key-dependent interleaving used in the watermark insertion stage, we can consider both sequences  $\alpha[i]$  and  $\sigma[i]$  as i.i.d. With these assumptions it is clear that all key-dependent partitions will produce the same results.

We will consider first the case of hard-decision decoding and then the soft-decision case. We will later discuss the use of concatenated codes and finally give some hints on how *turbo-codes* can be employed to improve the performance of data hiding systems.

### 7.1. Hard-decision decoding

In this case, an independent decision is taken for each codeword symbol. If the pdf of  $x[j]$  is symmetric then the symbol-by-symbol ML decision

threshold will be set at the origin. Let us assume, without loss of generality, that the  $i$ th codeword symbol takes the value +1 and that  $s[j] = 1, j \in \mathcal{S}_i$ . Then, the probability of error  $p(i | \alpha[j], \sigma[j])$  for this symbol is

$$p(i | \alpha[j], \sigma[j]) = P\{x[j] + \alpha[j] < 0\}, \quad j \in \mathcal{S}_i. \quad (59)$$

The probability  $p(i | \alpha[j], \sigma[j])$  can be easily evaluated for the case in which  $x[j]$  follows a Laplacian distribution, yielding

$$p(i | \alpha[j], \sigma[j]) = \frac{\exp(-\sqrt{2}\alpha[j]/\sigma[j])}{2}. \quad (60)$$

When this probability is averaged over the sequence  $\alpha$  of perceptual masks and the sequence  $\sigma$  of variances, it becomes independent of the index of the transmitted symbol. Thus, the BSC cross-over probability  $p$  for this decoder can be written as

$$p = \int_{\alpha, \sigma} \frac{\exp(-\sqrt{2}\alpha/\sigma)}{2} f_{\alpha, \sigma}(\alpha, \sigma) d\alpha d\sigma. \quad (61)$$

Knowledge of the joint pdf of  $\alpha$  and  $\sigma$  is then necessary for solving (61). The marginal pdf of  $\alpha$  can be obtained from the way the perceptual mask is computed by taking advantage of a ‘continuous approximation’ on the magnitudes of the DCT coefficients which will not be pursued here due to the space limitation. Another way of solving (61) is by means of an ergodic assumption on  $\alpha$  and  $\sigma$  that allows to approximate  $p$  as

$$p = \frac{1}{L} \sum_{j=1}^L \frac{\exp(-\sqrt{2}\alpha[j]/\sigma[j])}{2}. \quad (62)$$

Once  $p$  is available, it can be substituted into (29) and (30) to give bounds and approximations to  $P_b$ . Note that since the value of  $p$  will be in general quite large, use of powerful codes will be essential in order to achieve good performance. An effective way of attaining this is through the use of concatenated codes, as will be discussed shortly.

### 7.2. Soft-decision decoding

The optimal ML detector for the Laplacian case would decide codeword  $\hat{c} \in \mathcal{C}$  such that (33) is

satisfied, where the sufficient statistics  $r[i]$  are computed as in (6), taking into account that now the sets  $\mathcal{S}_i$  consist of just one sample.

Regarding the bit error probability, in order to use the union bound, the probability of error between two codewords has to be computed. Assuming that  $c_1$  and  $c_2$  are the only existing codewords and that  $c_1$  is sent, we should determine the probability of error conditioned on a certain sequence of perceptual masks  $\alpha$  and variances  $\sigma$ , that is,  $P(c_1 \rightarrow c_2 | \alpha, \sigma)$  and where now  $x$  is the only random variable in the system. Collecting all these considerations, we can state that there will be a block error (i.e., the decoder will decide  $c_2$ ) iff

$$\sum_{i=1}^n (c_1[i] - c_2[i]) \times \frac{|y[j] + \alpha[j]s[j]| - |y[j] - \alpha[j]s[j]|}{\sigma[j]} \Big|_{j \in \mathcal{S}_i} < 0. \quad (63)$$

In Appendix A we derive a Chernoff upper bound to the probability of error between two codewords. For linear codes, once the set of two-codewords error probabilities between  $c_1$  and any other codeword are available, we can write

$$P_c \leq \sum_{i=2}^n P(c_1 \rightarrow c_i). \quad (64)$$

### 7.3. Concatenated coding

Sample-level coding for data-hiding applications faces the problem of the very low PSSNR that is usually encountered, which is a direct consequence of the imperceptibility constraint. Then, although we have shown the potential advantage of coding instead of simple repetition, it is also true that powerful codes would be required in order to achieve a small probability of error. Unfortunately, for moderately large values of  $d_{\text{free}}$  ( $d_{\text{min}}$ ) in the convolutional (block) code, decoding has a tremendous complexity. This difficulty also arises in deep-space communications where transmitted power is as well severely limited [29].

A popular solution to this problem is that of concatenated codes, proposed by Forney [5] and summarized in Fig. 2 for a typical data hiding

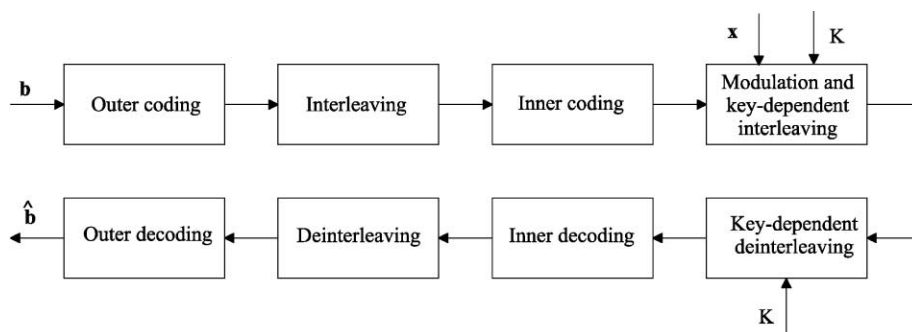


Fig. 2. Model for concatenated coding in data hiding.

application. Note that only one level of concatenation is shown, but the idea is easily generalized to any level. In our context, the inner code would be a binary block  $(n, k)$  or convolutional  $(k/n)$  code and the outer code would be a block code (typically, a Reed–Solomon) with an alphabet of  $q$  symbols (typically,  $q = 256$ ) so that the output of the latter is compatible with the input of the former. With concatenation it is possible to achieve a  $d_{\min}$  which is the product of the minimum distances of the two concatenated codes; on the other hand, decoding complexity is merely that of each individual code. An important element of many concatenated codes is the interleaver [24,28] which is a device that simply produces a deterministic permutation of the symbols at the output of the outer encoder. This permutation avoids that error bursts at the output of the inner decoder appear at the input of the outer decoder, making it easier to correct them. These error bursts are common at the output of convolutional decoders. Note that interleaver memory, a frequent concern when designing interleavers for communications applications is not so critical here due to the availability of the entire image.

Concerning performance analysis, this is quite straightforward given the results of Sections 3 and 4: first, the cross-over probability  $p$  of a decoding error (either with soft or hard decision) in the inner code is computed (or upper-bounded) and this value is substituted into (30) or similar to obtain the overall bit error probability  $P_b$ . Of course, this approach to performance assumes a perfect interleaver that makes the decisions of the inner decoder look to be independent. The performance  $P_b$  versus

PSSNR  $L/N$  curves for concatenated codes are very steep, meaning that a small increase in the PSSNR produces an enormous improvement in the overall  $P_b$ . Outer Reed–Solomon codes for data-hiding applications were first proposed by O’Ruanaidh and Pun [22] combined with an election for the inner code that resembles pulse position modulations (PPM).<sup>5</sup> In practice, complexity issues should be taken into account but concatenated codes used in digital communications will also perform well for data hiding purposes. As a final remark, note that the use of codes combined with diversity can be seen also as a form of concatenation in which the inner code is simply a repetition code; interestingly, this choice allows for soft decoding of the outer code.

#### 7.4. Turbo coding

Closely related to concatenated codes is the recently proposed idea of iterative decoding. The original term ‘turbo codes’ refers to parallel concatenation of two or more systematic convolutional encoders that are decoded iteratively, with the advantage of a complexity comparable to that of the basic (constituent) codes. The subject has evolved rapidly during the past years and now variations on the basic theme exist (e.g., serially concatenated

<sup>5</sup> Although not explicitly discussed in our paper, this inner modulation can be analyzed in a way much similar to our discussion for orthogonal codes with approximately the same results.

codes, use of block codes, etc.), all of them sharing the idea of iterative decoding. A fundamental element for this codes is an interleaver that in the case of parallel concatenated codes acts upon the information bits that enter the parallel encoders, and that usually has a very large size that precludes practical ML decoding. Fortunately, after some iterations, near-optimal performance is achieved, which implies astonishing results. As an example, for very large interleavers and a moderate number of iterations, values of PSSNR  $L/N$  smaller than 3 dB suffice to achieve values of  $P_b$  lower than  $10^{-5}$ . Regarding decoding algorithms, they are based on maximum a posteriori (MAP) probabilities and give ‘soft’ (sometimes called reliability) information together with hard decisions thus providing an effective way of exchanging information between decoders.

Implementation of turbo coding for data hiding becomes a very promising line of future research. Similarly to the previous section, interleavers can be as large as the size of the original image so typical constraints appearing in digital communications, such as latency time, do not apply here. However, it is important to take into account that in order to provide reliability information, statistical knowledge of the image is required. In the DCT domain the soft information would take a form similar to (63). For an excellent view on turbo codes, see [2,7].

## 8. A glance at the detection problem

In several applications, such as copyright protection, we are interested in determining whether a given image contains a watermark. This is what we call the *watermark detection* problem. This should not be confused with the decoding of embedded information that we have analyzed in previous sections, since now we are interested only in detecting the mere presence of a watermark in the image we are testing.

Therefore, the watermark detection problem can be expressed as a hypothesis test with two hypotheses, namely “the image contains a watermark” ( $H_1$ ) and “the image does not contain a watermark” ( $H_0$ ). The optimal detector corresponds to the

Neyman–Pearson rule, in which  $H_1$  is decided if

$$\frac{f_y(\mathbf{y}|H_1)}{f_y(\mathbf{y}|H_0)} > \eta, \quad (65)$$

where  $\eta$  is a decision threshold, otherwise  $H_0$  is decided. The pdf in the numerator corresponds to the statistical distribution of the image under test when it contains a valid watermark, whereas the pdf in the denominator corresponds to the statistical distribution when no watermark is present. Again, we have similar problems as those we found in Section 2 regarding the statistical characterization of images. Specifically, we do not have good statistical models for images when watermarks are embedded in the spatial domain. However, we can resort to the Gaussian approximate model derived in Section 2 if we assume that the heuristically justified correlator receiver is used. Hence, in this case in the optimal detection rule  $H_1$  is decided when

$$\frac{f_r(\mathbf{r}|H_1)}{f_r(\mathbf{r}|H_0)} > \eta, \quad (66)$$

where  $\eta$  is the decision threshold, otherwise  $H_0$  is decided. In the DCT domain we do have fairly good statistical models, so we can use them in Eq. (65). Note that in this detection test we are not interested in extracting any information that the watermark might carry, as it was the case in previous sections. Therefore, if the watermark can encode a binary message, the pmf of the set of possible messages should be considered when  $f_y(\mathbf{y}|H_1)$  is derived.

In some cases, especially when error protection coding is used (Sections 3–6), the decision rule in Eqs. (65) and (66) can be difficult to implement. However, suboptimal detectors can be used. For instance, a suboptimal decision can be made in two steps. First, an ML decoder obtains an estimate  $\hat{\mathbf{b}}$  of the message  $\mathbf{b}$  carried by the watermark. Then, a hypothesis test similar to that in Eq. (65) is applied, now changing hypothesis  $H_1$  to “the image contains a watermark carrying the message  $\hat{\mathbf{b}}$ ”.

Another suboptimal detection algorithm is the following. First, hard-decision estimates of the encoded message  $\mathbf{c}$  are computed. Then, a binary test similar to (65) is applied, now using the pmf’s of the

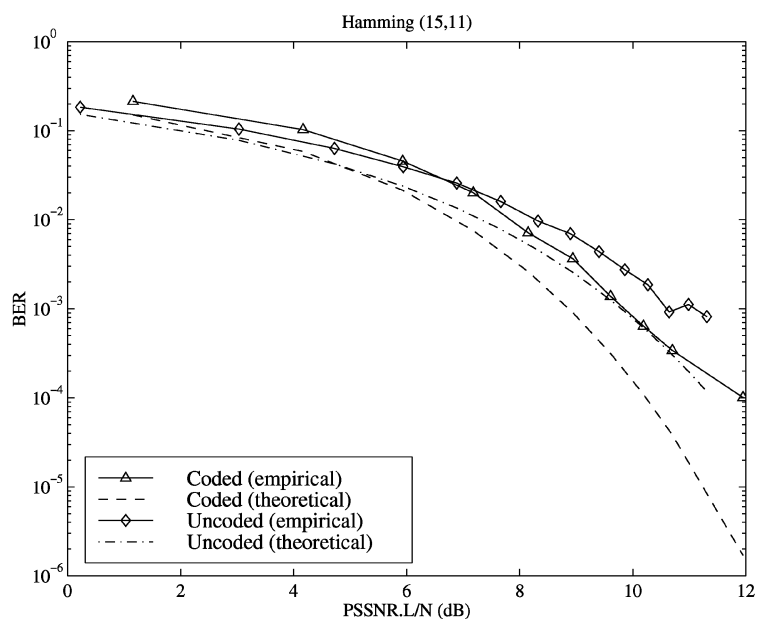


Fig. 3. Theoretical and empirical results for the Hamming (15,11).

hard-decisions  $\hat{c}$  conditioned to hypothesis  $H_1$  and  $H_0$ . This approach can be compared to hard-decision schemes used for information decoding. As a matter of fact, a similar performance degradation problem is experienced due to the hard-decision step.

From the discussion above, it is clear that the procedure proposed sometimes in the literature, consisting in using the watermark decoder to obtain an estimate of the message carried by the possibly existing watermark and verifying that it is a valid message (using for instance a checksum field), is in fact a suboptimal approach to solving the watermark detection problem expressed in Eq. (65). This combines the two suboptimal schemes presented, using both hard decisions and a suboptimal binary hypothesis test with only one message considered in hypothesis  $H_1$ . Further details and analytical results for the spatial domain are given in [14].

## 9. Experimental results

In order to illustrate the validity of our theoretical results we have watermarked the ‘Lena’

image ( $256 \times 256$  pixels) in the spatial domain with Wiener filtering in the extraction stage and different coding schemes have been compared. In all cases the empirical values have been obtained by averaging out the results for 100 randomly chosen keys and with an i.i.d sequence  $s$  that has a discrete pmf with four equiprobable levels  $\{-\sqrt{8/5}, -\sqrt{2/5}, \sqrt{2/5}, \sqrt{8/5}\}$ . This sequence is chosen for the purpose of illustrating how the proposed theoretical results are valid for any general distribution. In practice, other distributions will achieve better performance as has been shown in [13]. The curves here presented show the theoretical and empirical values of the bit error rate (BER) as a function of the parameter  $PSSNR L/N$  and include for comparison the BER for the uncoded case.

Fig. 3 shows the BER for a Hamming (15,11) code which has a very small error correcting capability of  $t = 1$ . Note that the uncoded and coded curves cross at a relatively high value of  $PSSNR L/N$ . The discrepancy between the theoretical results (obtained with (30) and the empirical ones is due to the errors in the estimates of  $a$  and  $\gamma$  in (17) and (18). Fig. 4 presents the results for

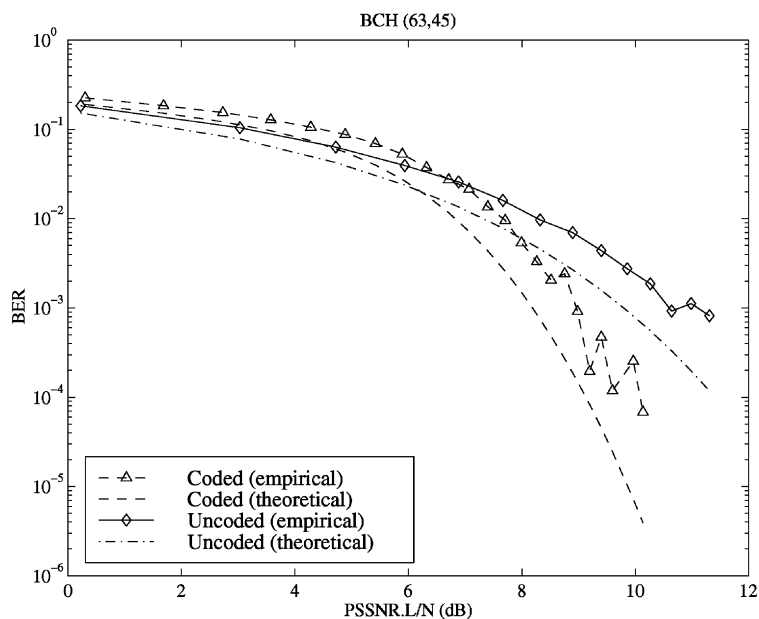


Fig. 4. Theoretical and empirical results for the BCH (63,45).

a BCH(63,45) code for which  $t = 3$ . Note the improvement in performance that is summarized in a lower PSSNR  $L/N$  crossing point. We have also performed experiments with other codes and in all cases the theoretical results closely match the empirical.

We also present in Figs. 5–7 results obtained for three different convolutional codes with respective rates  $1/2$ ,  $1/3$  and  $1/4$  when soft decision decoding is employed. These codes are designed by their generator polynomials expressed in octal form (see [19]) which are, respectively, (15, 17), (13, 15, 17) and (5, 7, 7, 7). The respective *constraint lengths* are 4, 4 and 3 (the definition in [19] is used) and the respective  $d_{\text{free}}$  parameters are 6, 10, 10. The theoretical results were obtained with the bound of (41) in which only the first term of the outer sum was considered. This provides good asymptotic results but offers a poor approximation for low values of PSSNR  $L/N$  due to the fact that more terms in the sum are non-negligible. With all these considerations and the discrepancies between the theoretical and empirical values of  $a$  and  $\sigma$  mentioned in the previous paragraph, truncated (41) becomes an

approximation and is no longer an upper bound. Asymptotically (for high SNR) the gain provided by the convolutional codes approaches  $10 \log_{10}(Rd_{\text{free}})$ , so the respective asymptotic gains (in dB) are 4.77, 5.22 and 3.97 which implies that the best results are achieved for the second code. Also shown in Figs. 5–7 are the results obtained for these convolutional codes with hard-decision decoding. Note how performance degrades when compared to soft decision, but it is still better (provided that a minimum SNR is achieved) than for the uncoded case.

Finally, Fig. 8 shows the results obtained with orthogonal codes in two cases:  $n = 8$  and 64. For these two cases, besides the empirical results, we show the theoretical performance evaluated by numerical integration of (47) and bound (48). Note again that errors in the estimation of the parameters  $a$  and  $\gamma$  convert (51) into just an approximation. For the Wiener filter case we have observed that increasing  $n$  does not necessarily lead to improved asymptotic results as theory would predict. This is due to the non-negligible correlations between transmitted words that become higher as the

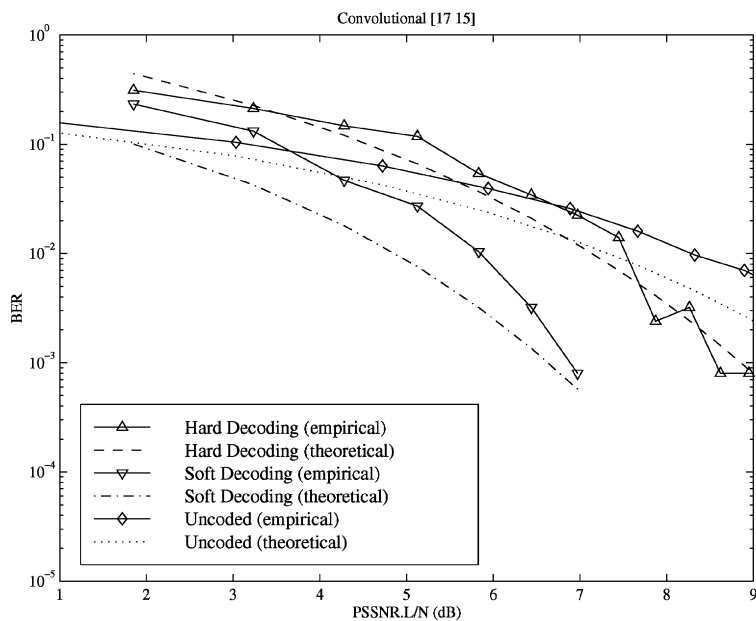


Fig. 5. Theoretical and empirical results for the rate 1/2.

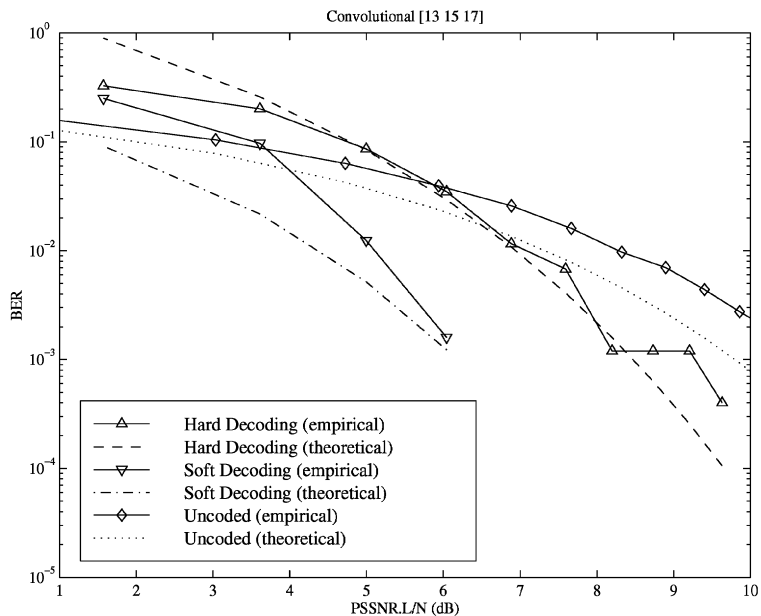


Fig. 6. Theoretical and empirical results for the rate 1/3.

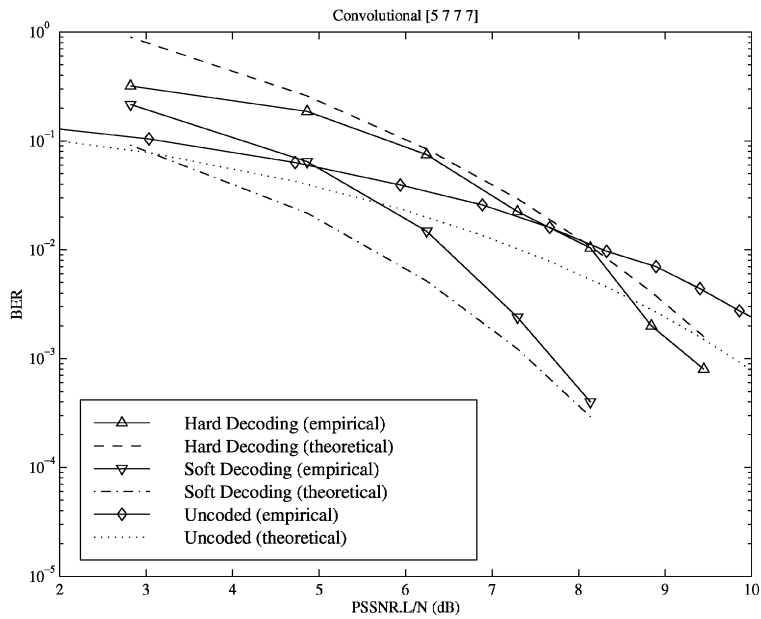


Fig. 7. Theoretical and empirical results for the rate 1/4.

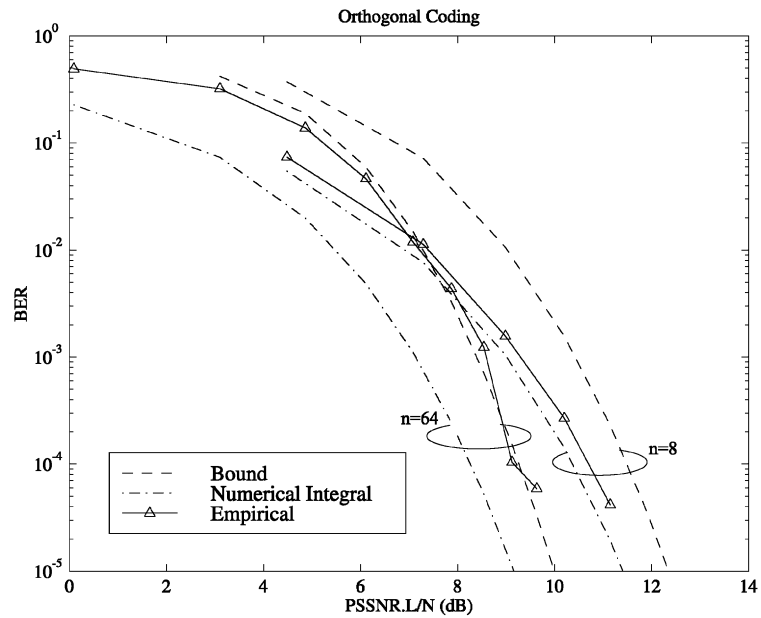


Fig. 8. Theoretical and empirical results for orthogonal codes with  $n = 8$  and  $n = 64$ .

number of samples per pulse increase. Comparing Figs. 6 and 8 we see that for practical sizes convolutional codes fall closer to the capacity limit.

## 10. Conclusions

In this paper we have given an overview of the advantages that channel coding brings about in data hiding applications, although the similarities with the detection problem have been also pointed out. Careful statistical modeling is the key for deriving decoding structures with optimal performance. However, in those cases where no such model is available, judicious choice of a heuristic decoding function also produces good results. In any case, we have chosen the bit error probability as the reference quality measure and have given theoretical results that are represented in a way that becomes independent of the image to be watermarked, whereas the operating point does depend on the particular image. Analysis of the different coding schemes reveals superior performance of convolutional codes for a reasonable complexity. Comparison with the uncoded case shows gains of about 5 dB for simple codes. Note that a coding gain of 3 dB allows doubling the number of hidden information bits for the same  $P_b$ .

We have also discussed the benefits of coding at the sample level, especially when concatenated or turbo codes are employed. However, much theoretical work remains to be done at this level, since it is no longer possible to resort to central limit considerations which are very helpful for the diversity case as it was shown in the paper. Moreover, compact results for the sample level case demand a model not only for the pdf of the image, but also for the parameters involved in the detector such as the sequence of perceptual masks. One possible approach might follow the treatment given to digital communications over Rayleigh channels, but adapted to the models at hand.

The work here presented can be extended to other domains; it would be particularly interesting to generalize it to the Fourier–Mellin transform domain [22] where affine transformation attacks can be compensated. Again, the statistical approach requires more knowledge

about the distributions of the coefficients in this domain.

Finally, as was mentioned in the Introduction, a deterministic approach leads to zero probability of error decoding schemes which unfortunately have little robustness against distortions and attacks. An attractive possibility would be to combine the two approaches (for instance, watermarking part of the samples with a deterministic method and the remaining with a probabilistic scheme) in order to collect the advantages of each. Of course, this asks for a rigorous study of the possible distortions and attacks and perhaps new procedures that might include ideas from the field of robust statistics.

## Appendix A. Chernoff bound for sample-level data hiding in the DCT domain

In this appendix we obtain a Chernoff bound for the two-codewords error probability in the DCT domain (Laplacian case) with coding at the sample level and soft decoding. We will assume that the samples of  $s[i]$  take the values  $\{\pm 1\}$ . First, we upper-bound the probability that inequality (63) holds when  $c_1$  is correct and  $\alpha$  and  $\sigma$  are fixed. This probability, denoted by  $P_2(\alpha, \sigma)$ , can be written as

$$P_2(\alpha, \sigma) = P\left\{\sum_{j \in \mathcal{J}} \frac{|x[j]| - |x[j] + 2\alpha[j]s[j]|}{\sigma[j]} > 0\right\}, \quad (\text{A.1})$$

where  $\mathcal{J}$  is the set of samples belonging to  $\mathcal{S}_i$  for which  $c_1[i] \neq c_2[i]$ . Without loss of generality, assume that  $s[j] = 1$  for all  $j \in \mathcal{J}$ , then the Chernoff bound to  $P_2$  can be written as

$$P_2(\alpha, \sigma) \leq \min_{z > 0} \prod_{j \in \mathcal{J}} E\left\{\exp\left(z \frac{|x[j]| - |x[j] + 2\alpha[j]|}{\sigma[j]}\right)\right\}. \quad (\text{A.2})$$

Each of the expectations in the product above takes the form

$$\frac{\sqrt{2}}{2\sigma[j]} \int_{-\infty}^{\infty} \exp\left(\frac{(z - \sqrt{2})|x[j]| - z|x[j] + 2\alpha[j]|}{\sigma[j]}\right) dx[j]. \quad (\text{A.3})$$

The integral above can be split into three pieces, which can be evaluated analytically and manipulated to yield

$$\frac{ze^{2(z-\sqrt{2})\alpha[j]/\sigma[j]} + (z-\sqrt{2})e^{-2z\alpha[j]/\sigma[j]}}{(2z-\sqrt{2})}. \quad (\text{A.4})$$

The minimum of (A.4) can be shown to be achieved at  $z = \sqrt{2}/2$  and is independent of  $j$ , so this value of  $z$  also minimizes (A.2) and the final Chernoff bound becomes

$$P_2(\alpha, \sigma) \leq \prod_{j \in \mathcal{J}} \exp\left(\frac{-\sqrt{2}\alpha[j]}{\sigma[j]}\right) \left(1 + \frac{\sqrt{2}\alpha[j]}{\sigma[j]}\right). \quad (\text{A.5})$$

The case of  $\alpha[j] = \alpha$  and  $\sigma[j] = \sigma$  for all  $j \in \mathcal{J}$  is particularly illustrative, since (A.5) becomes

$$P_2(\alpha, \sigma) \leq e^{-d_{1,2}\sqrt{2}\alpha/\sigma} \left(1 + \frac{\sqrt{2}\alpha}{\sigma}\right)^{d_{1,2}}, \quad (\text{A.6})$$

where  $d_{1,2}$  is the Hamming distance between  $c_1$  and  $c_2$ . Note that, as expected, the bound decreases with  $d_{1,2}$  and also with  $\alpha/\sigma$  which could be taken as a PSSNR for this case.

In any case, the randomness and independence in  $\alpha$  and  $\sigma$  can be taken into account to write

$$P(c_1 \rightarrow c_2) \leq \left[ \int_{\alpha, \sigma} e^{-\sqrt{2}\alpha/\sigma} \left(1 + \frac{\sqrt{2}\alpha}{\sigma}\right) f_{\alpha, \sigma}(\alpha, \sigma) d\sigma d\alpha \right]^{d_{1,2}}. \quad (\text{A.7})$$

The right hand side of (A.7) can be approximated as in (62).

## References

[1] M. Barni, F. Bartolini, V. Capellini, A. Piva, F. Rigacci, A MAP Identification Criterion for DCT-based watermarking, Proceedings of the European Signal Processing Conference, Vol. I, Rhodes, Greece, September 1998, pp. 17–20.  
 [2] S. Benedetto, D. Divsalar, G. Montorsi, F. Pollara, Analysis, design and iterative decoding of double serially concatenated codes with interleavers, IEEE J. Selected Areas Commun. 16 (2) (February 1998) 231–244.  
 [3] I.J. Cox, M.L. Miller, A.L. McKellips, Watermarking as communications with side information, Proc. IEEE 87 (7) (July 1999) 1127–1141.

[4] G. Csurka, F. Deguillaume, J.K.O. Ruanaidh, T. Pun, A Bayesian approach to affine transformation resistant image and video watermarking, Proceedings of Information Hiding Workshop, Dresden, Germany, Springer, October 1999.  
 [5] G.D. Forney, Concatenated Codes, MIT Press, Cambridge, MA, 1966.  
 [6] R.G. Gallager, Information Theory and Reliable Communication, Wiley, New York, 1968.  
 [7] J. Hagenauer, E. Offer, L. Papke, Iterative decoding of binary block and convolutional codes, IEEE Trans. Inform. Theory 42 (2) (March 1996) 429–445.  
 [8] J.R. Hernández, M. Amado, F.P. González, DCT-domain watermarking techniques for still images: detector performance analysis and a new structure, IEEE Trans. Image Process. 9 (1) (January 2000) 55–68.  
 [9] J.R. Hernández, J.-F. Delaigle, B. Macq, Improving data hiding by using convolutional codes and soft-decision decoding, Proceedings of IST/SPIE 12th Annual International Symposium, San Jose, California, USA, January 2000.  
 [10] J.R. Hernández, F. Pérez-González, Statistical analysis of watermarking schemes for copyright protection of images, Proc. IEEE 87 (7) (July 1999) 1142–1166.  
 [11] J.R. Hernández, F. Pérez-González, M. Amado, Improving DCT-domain watermarking extraction using generalized gaussian models, Proceedings of the COST #254 Workshop on Intelligent Communications and Multimedia Terminals, Ljubljana, Slovenia, November 1998, pp. 23–26.  
 [12] J.R. Hernández, F. Pérez-González, J.M. Rodríguez, The impact of channel coding on the performance of spatial watermarking for copyright protection, Proceedings of ICASSP'98, Vol. 5, Seattle, Washington, USA, May 1998, pp. 2973–2976.  
 [13] J.R. Hernández, F. Pérez-González, J.M. Rodríguez, G. Nieto, Performance analysis of a 2D-multipulse amplitude modulation scheme for data hiding and watermarking of still images, IEEE J. Selected Areas Commun. 16 (May 1998) 510–524.  
 [14] J.R. Hernández, J.M. Rodríguez, F. Pérez-González, Improving the performance of spatial watermarking of images using channel coding, Signal Processing 80 (7) (July 2000) 1261–1279.  
 [15] M. Kutter, Performance improvement of spread spectrum based image watermarking schemes through M-ary modulation, Proceedings of Information Hiding Workshop, Dresden, Germany, Springer, Berlin, October 1999.  
 [16] D. Kundur, D. Hatzinakos, Digital watermark for telltale tamperproofing and authentication, Proc. IEEE 87 (7) (July 1999) 1167–1180.  
 [17] S. Lin, D.J. Costello, Error control coding: Fundamentals and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1983.  
 [18] F. Pérez-González, F. Balado, Provably or probably robust digital watermarking?, in preparation.  
 [19] J.G. Proakis, Digital Communications, 2nd Edition, McGraw-Hill, New York, 1989.

- [20] J.M. Rodríguez, Channel coding for spatial image watermarking, Master's Thesis, University of Vigo, 1998 (in Spanish).
- [21] J.J.K.O. Ruanaidh, W.J. Dowling, F.M. Boland, Watermarking digital images for copyright protection, *IEE Proc. Vision Image Signal Process.* 143 (4) (August 1996) 250–256.
- [22] J.J.K.O. Ruanaidh, T. Pun, Rotation, scale and translation invariant spread spectrum digital image watermarking, *Signal Processing* 66 (3) (May 1998) 303–317.
- [23] H. Sari, F. Vanhaverbeke, M. Moeneclaey, Extending the capacity of multiple access channels, *IEEE Commun. Mag.* 38 (1) (January 2000) 74–82.
- [24] B. Sklar, *Digital Communications, Fundamentals and Applications*, Prentice-Hall International Editions, Englewood Cliffs, NJ, 1988.
- [25] J.R. Smith, B.O. Comiskey, Modulation and information hiding in images, *Proceedings of International Workshop on Information Hiding*, Cambridge, UK, Springer, May 1996, pp. 207–226.
- [26] A.J. Viterbi, *CDMA, Principles of Spread Spectrum Communication*, Addison-Wesley Publishing Company, Reading, MA, 1995.
- [27] A.J. Viterbi, Jim K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill, New York, 1979.
- [28] S.G. Wilson, *Digital Modulation and Coding*, Prentice-Hall, Upper Saddle River, NJ, 1996.
- [29] J. Yuen, M. Simon, W. Miller, C. Ryan, D. Divsalar, J. Morakis, Modulation and coding for satellite and space communications, *Proc. IEEE* 78 (8) (July 1990) 1250–1266.