ELSEVIER

Signal Processing 81 (2001) 1239–1250

SIGNAL PROCESSING

www.elsevier.nl/locate/sigpro

Analyses of error correction strategies for typical communication channels in watermarking

Severine Baudry[a,b,*], Jean-François Delaigle[c], Bülent Sankur[d], Benoît Macq[c], Henri Maître[a]

[a]*Ecole Nationale Supérieure des Télécommunications, 46 rue Barrault, 75013 Paris, France*
[b]*Thomson CSF Communications, 66 rue du Fosse Blanc, BP 156, 92231 Gennevilliers Cedex, France*
[c]*Laboratoire de Télécommunications et Télédétection, Université catholique de Louvain, Bâtiment Stévin-2 place du Levant, B-1348 Louvain-la-Neuve, Belgium*
[d]*Boğaziçi University, Department Electrical-Electronics Engineering, Bebek, Istanbul, Turkey*

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

Abstract

Most watermarking techniques rely on redundancy where signature bits are encoded in a sufficiently large amount of sites for robustness against distortions and attacks while the watermark energy is kept low enough to remain imperceptible. The goal of this paper is to explore some strategies for exploiting this redundancy using error correcting codes. In some watermarking techniques bits are extracted via hard-decisions leading to a binary symmetric channel model while in others the extraction is carried out with soft-decisions leading to a additive Gaussian channel model. First, we consider error correcting codes for very high error rates of the watermarking channel where the trade-offs of Bose–Chaudury–Hocquenheim (BCH) and repetition codes are investigated. We also present the performance and a fast implementation of soft-decoders. We give two potential realizations of soft-decoding, namely, Viterbi decoder for convolutional codes and a new algorithm for soft-BCH decoding.

Keywords: Watermarking; Error correcting codes; Soft-decoding

1. Introduction

Digital watermarking systems can be viewed as digital communication systems [4,10,12]. To guarantee the survivability of the watermark message, that is to say, reliable transmission of the watermark symbols through this particular communication channel, recourse is made to spreading of the watermark over the image. This can be done by means of Spread Spectrum [3,7,8,19] or via multi-site marking. In addition, since the signal to noise ratio under which this particular communication system operates is very low due to imperceptibility constraints it is natural to envision the use of error correcting codes (ECC). While the role of ECC is well understood in communication channels, some of the trade-offs for watermarking

* Corresponding author. Ecole Nationale Supérieure des Télécommunications, 46 rue Barrault, 75013 Paris, France.

E-mail addresses: severine.baudry@tcc.thomson-csf.com, baudry@tsi.enst.fr (S. Baudry), delaigle@tele.ucl.ac.be (J.F. Delaigle), sankur@boun.edu.tr (Bülent Sankur), macq@tele.ucl.ac.be (B. Macq), maitre@tsi.enst.fr (H. Maître).

0165-1684/01/$ - see front matter
PII: S0165-1684(01)00041-X

channels remain to be explored. A number of relevant studies [11,12,17] to this effect have been conducted.

In this paper we investigate the role of two error correcting schemes for protecting watermark messages in still and motion pictures. More specifically we would like to show under what conditions it is beneficial to employ ECC in watermarking, and discuss coding strategies for watermarking channels modeled as a binary symmetric channel (BSC) or as a Gaussian channel. In the first scheme we consider the concatenation of Bose-Chaudury-Hocquenheim (BCH) and repetition codes for very noisy conditions typical of watermarking channels. There is a trade-off between repetition and BCH code word lengths. In the second scheme the detector statistics form a Gaussian channel for which the advantages of convolutional codes are explored. Finally, we compare the performance of watermark detection and decoding under both hard-decision and soft-decision schemes.

In Section 2 we describe the coding trade-offs for BCH and repetition codes and indicate when repetition coding becomes the last resort for very unreliable channels. In Section 3 a new soft-decision scheme for block codes decision is introduced. The protection afforded by convolutional codes along with a Viterbi decoder are presented in Section 4. Results and concluding remarks are given in Section 5.

2. Concatenation of error correcting codes in BSC channels

A "watermarking channel" is modeled as a binary symmetric channel when the embedded information is demodulated with hard-decision to a binary code word. The average probability that a bit of the received sequence in this channel is in error is denoted by p_b (uncoded error probability). The 'raw' bit error probability can be improved with repetition coding, or with an algebraic code like BCH, or their concatenation, as shown in Fig. 1. The resulting bit-error and signature[1]-error probabilities can be computed as in Table 1. The superscripts bit and sig, respectively, denote the bit and message error probability, whereas the subscripts cod and rep, respectively, denote the probabilities with error correcting code and with repetition.

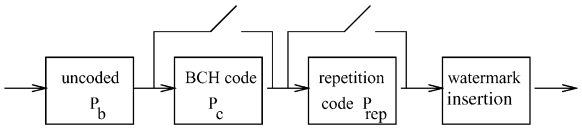

Fig. 1. Block diagram of a watermarking scheme with code protection.

These probabilities depend on the global rate of redundancy, R, which is the number of watermarking sites used, on the average, for each bit of the message,and on message length k The total number of bits available for coding is Rk. For the algebraic code, the performance also depends on the minimum distance of the code $d_{\min} = 2t + 1$. The formulae for the concatenated codes can be easily deduced by replacing p_b by the error probability improved with repetition p_{rep}. In each box in Fig. 1 corresponding to no coding, BCH coding or repetition coding, the corresponding signature error probability is denoted.

2.1. Performance with repetition codes

The error performance of a k-bit signature when an $(R,1)$ repetition code is used to upgrade the channel bit error probability, p_b to p_{rep} is shown in Fig. 2a. Note that the rate of repetition should be an odd number, at least for small R. In this figure $R = 1$ corresponds to the signature error probability in the original BSC channel, i.e., without any repetition.

2.2. Performance with BCH codes

The improvement brought in by an (n, k) error correcting code can be quantified as a function of redundancy, n/k, and the minimum distance. It is assumed that an $(n, k, d_{\min})$ code, like BCH codes, will correct all code words containing up to t errors and will fail for all others. The issue of when a

[1] In this paper, the signature is nothing else than the watermark message.

Table 1
Bit and signature error probability expressions

	Repetition	Coded
Bit error	$p_{\text{rep}}^{\text{bit}} = \sum_{i=R/2+1}^{R} \binom{R}{i} p_b^i (1-p_b)^{R-i}$	$p_{\text{cod}}^{\text{bit}} \cong \frac{k}{n} \sum_{i=t+1}^{n} i \binom{n}{i} p_b^i (1-p_b)^{n-i}$
Signature error	$P_{\text{rep}}^{\text{sig}} = 1-(1-p_{\text{rep}}^{\text{bit}})^k$	$P_{\text{cod}}^{\text{sig}} = \sum_{i=t+1}^{n} \binom{n}{i} p_b^i (1-p_b)^{n-i}$

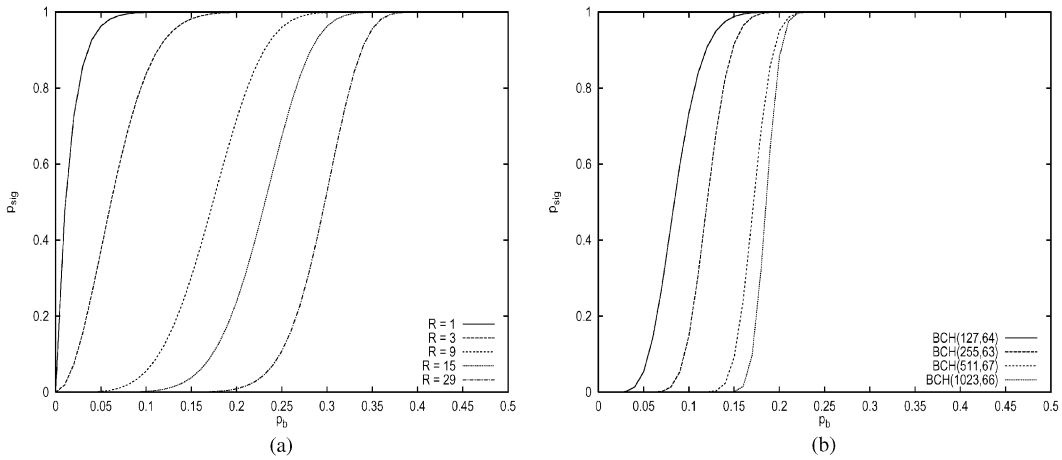

Fig. 2. Probability of incorrect 64-bit signature detection as a function of: (a) R repetition redundancy; (b) BCH code redundancy.

decoder fails to find a decoded word is addressed separately in Section 3 in the context of soft-decoding. In a BSC channel the number of errors will have approximately a Gaussian distribution with mean np_b and variance $np_b(1-p_b)$. For elevated values of p_b most of the mass of this distribution falls outside the correcting capability of the code, that is beyond the threshold t. One can observe indeed in Fig. 2b that BCH codes contribute significantly to error protection when the channel error probability is below approximately 0.1. For worse channels it starts failing causing even more errors than the uncoded case.

To explore the trade-off between repetition and BCH, we have determined the value of p_b beyond which a repetition code performs, if ever, better than an error-correcting code. This should throw some light on how the redundancy should be traded-off. As shown in Fig. 3 the ratio of $P_{\text{sig}}^{\text{rep}}/P_{\text{sig}}^{\text{code}}$ falls below 1 (hence repetition is to be preferred to BCH) after a certain threshold value of p_b around 0.15–0.20. The advantage of the repetition vis-a-vis BCH becomes even more prominent with increasing amount of redundancy. It must be pointed out, however, that BCH codes perform, as expected, orders of magnitude better than repetition codes below this threshold value.

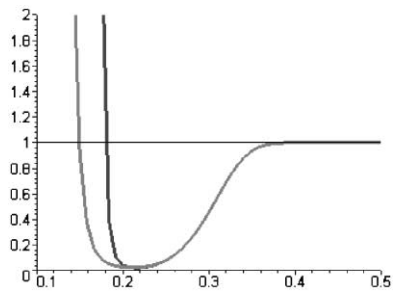

Fig. 3. The ratio of error probability of 64-bit signature with 31 repetitions to that of BCH(255, 9) (left); and BCH(1023, 36) (right). Repetition, BCH(255, 9) and BCH(1023, 36) result in 1984, 2040, 2046 code word lengths, respectively.

2.3. Performance with concatenation

The observation that the error correcting codes cannot display their potential unless the channel BER is reduced below a critical value brings about the possibility of first improving the channel BER via repetition coding to an acceptable level, before BCH decoding. There are in fact two possible concatenations:

- repetition as an inner code and BCH as an outer code,
- BCH as an inner code followed by repetition as an outer code.

The second alternative is obviously more viable as in the first strategy most BCH coders might fail with uncoded (no repetition) channel error rate.

As an example consider the exploitation of a redundancy factor of 31 for a watermark message of 16 bits. One can consider the following configurations: (a) Repeat the watermark message 31 times, that is use a (31,1) repetition code; (b) Use a (511,19) BCH code with no repetition; (c) Use a (31,6) BCH code followed by a repetition code of (5,1). In these configurations the total number of sites used amount to, respectively, 506, 511, 555. Notice in Fig. 4 that for high enough p_b the repetition code is still the best remedy. However in the interval $0.15 < p_b < 0.25$, the concatenation of repetition and BCH coding becomes superior. Notice that this is an operation range where conventional communication systems rarely operate.

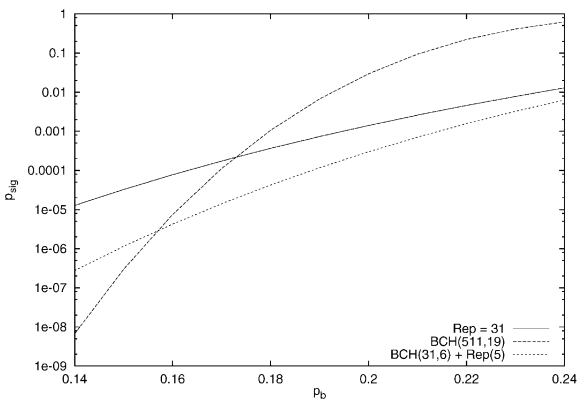

Fig. 4. Concatenated coding performance: Signature length = 16, redundancy factor = 31. Comparison of (solid) (31,1) repetition code, of (dashed) BCH(511,19) code, of (dotted) (31,6) BCH code followed by (5,1) repetition. The respective code word lengths are, 506, 511, 555 bits. Concatenation performs better between 0.16 and 0.22.

3. Soft-decoding of concatenated codes in BSC channels

In this section we study soft-decoding applied to block codes with concatenation. Block codes soft-decoding is rarely used due to the prohibitive complexity of state-of-the-art algorithms. This is why we have developed a new algorithm with reduced complexity.

3.1. The coding scheme

We use here the coding scheme presented in Section 2, and we run simulations for the specific case of watermarks encoded with BCH (127, 64, 21), then repeat the coded word R times. The coded bits are then embedded into the image using a binary substitutive modulation technique adapted from the Zhao–Koch method [1,13]. In this method, 8×8 blocks of the image are pseudo-randomly selected with a key. The DCT of each block is computed and two coefficients C_1 and C_2 in the mid-frequencies are selected. A watermark bit b_i is then embedded by imposing $|C_1| > |C_2|$ if $b_i = 0$ and $|C_2| > |C_1|$ if $b_i = 1$. If the modification caused by the watermark embedding is too strong and leads to psycho-visual artifacts, the block is rejected and another block is chosen.

3.2. Hard-decoding

We model the channel as a BSC with error probability p_b. A simple way to decode the information $\mathbf{r}$ at the output of the demodulator is to perform hard-decoding. First the repetition is decoded by majority logic as in Section 2: for a bit b_i repeated R times, the decoder will decide $b_i = 1$ if the number of demodulated 1's is greater than $R/2$. The retrieved binary word $\hat{\mathbf{c}}^{Maj}$ is then BCH-decoded by the Berlekamp–Massey algorithm.

Obviously, this method is not optimal, since the data at the output of the repetition-decoder are quantized, leading to an irreversible loss of

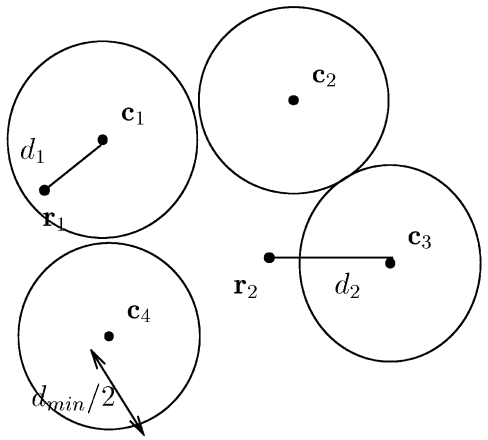

Fig. 5. Geometrical sketch of incomplete decoding. $\mathbf{r}_1$ is inside a code-word sphere and can be decoded as $\mathbf{c}_1$. $\mathbf{r}_2$ is outside the codeword spheres and thus cannot be decoded, even if it is closer to $\mathbf{c}_3$ than to every other word. For complete decoding, the decoding regions are the Voronoi regions associated with the code words.

information. Moreover, the Berlekamp–Massey algorithm itself is not optimal, since it is an incomplete algorithm. It can be shown that optimal decoding, which corresponds to maximizing the a-posteriori probability $P_c(\mathbf{c}/\mathbf{r})$ over all the code-words $\mathbf{c}$, is equivalent to finding the code-word $\mathbf{c}$ which is closest to $\mathbf{r}$ (the distance being the Euclidean distance for an AWGN channel,[2] or the Hamming distance for a BSC channel). Incomplete algorithms enable to find the closest code-word only if $d(\mathbf{c},\mathbf{r}) \leqslant t$ (see Fig. 5).[3] The probability of non-decoding P_{ND} is thus higher than with exhaustive decoding; however, the probability of erroneous decoding P_{ERR} is lower. Unfortunately, there is no known complete decoding algorithm with reasonable complexity for BCH codes [15]. Note that with incomplete decoding, detection and decoding can be performed jointly. When the decoding algorithm fails, the received word is far from any code-word, thus the data can be considered as non-watermarked. Note also that joint detection/decoding is more optimal than separate detection/decoding which implies quantization, thus loss of data.

Although there is no complete decoding algorithm for non-trivial codes, the watermark retrieval can be improved by using soft-decoding instead of hard-decoding in the following way: instead of dealing with binary symbols at the output of the repetition decoder, we will keep some information about the reliability of the decoded bits. This information will allow us to work with a set of "likely" binary words for BCH-decoding step, rather than with only one binary word as in hard-decoding.

3.3. Channel information

Instead of decoding the repetition code by majority logic, we now use the log-likelihood λ_i of the received symbols, defined by

$$\lambda_i = \ln\left(\frac{P(b_i = 1/z_i)}{P(b_i = 0/z_i)}\right),$$

where z_i is simply the count of "1" demodulations out of R repetitions.

We note that $|\lambda_i|$ is a measure of the relative reliability of the received symbols, and that $\mathrm{sign}(\lambda_i)$ indicates which binary symbol is the most likely. Assuming the equiprobability of 0 and 1, and using $P(z_i/b_i = 0) = \binom{R}{z_i} p_b^{z_i}(1 - p_b)^{R - z_i}$, we have

$$\lambda_i = (2z_i - R)\ln\left(\frac{p_b}{1 - p_b}\right).$$

Note that the log-likelihood is proportional to $2z_i - R$, thus we do not need to know the specific value of p_b. In the following we will take $\lambda_i = 2z_i - R$ for the sake of simplicity.

3.4. Chase soft-decoding algorithms

BCH hard-decoding sometimes fails if $\hat{\mathbf{c}}^{Maj}$ is outside the decoding spheres (Fig. 5). Nevertheless it is possible to reduce the number of failures by applying the Berlekamp–Massey algorithm not only on $\hat{\mathbf{c}}^{Maj}$ but also on words "close" to $\hat{\mathbf{c}}^{Maj}$ (i.e. "likely" binary words).

[2] AWGN: Additive White Gaussian Noise.

[3] For the Berlekamp–Massey algorithm, $t = \lfloor (d_{\min} - 1)/2 \rfloor$ and $d(.,.)$ is the Hamming distance.

More precisely, the Chase algorithm [2] consists of two steps: first we compute the log-likelihood λ_i for every symbol i and deduce from it a set E_T of "likely" binary-valued words $\hat{\mathbf{c}}$ (the likely binary words are obtained by flipping some unreliable bits of $\hat{\mathbf{c}}^{Maj}$). Finally, a hard-decoder (Berlekamp–Massey in our case) is applied to every binary word of E_T, and the decoded word closest to $\mathbf{r}$ is declared as the final decision.

Chase [2] proposed three versions of this algorithm, which differ in the choice of E_T. In algorithm 1, E_T is very large, so the decoding process is very complex but closely approaches complete decoding. On the contrary, algorithm 3 is very simple but less efficient. We based our study on algorithm 2 which achieves a good trade-off between complexity and performance. Let $\{b_{i_1} \dots b_{i_t}\}$ be the $t = d_{\min}/2$ least reliable symbols of $\hat{\mathbf{c}}^{Maj}$ (i.e. with the lowest $|\lambda_i|$'s). In algorithm 2 E_T is defined by

$$E_T = \{\hat{\mathbf{c}} \in \{0,1\}^n;\ \forall j \notin \{i_1, \dots i_t\} \hat{\mathbf{c}}_j = \hat{\mathbf{c}}_j^{Maj}\}.$$

In other words, we allow the less reliable bits of $\hat{\mathbf{c}}^{Maj}$ to be flipped. In our case, we have $d = 21$, thus there are 1024 words in E_T.

3.5. A new soft-decoding algorithm

One shortcoming of Chase algorithms is that they do not guarantee that E_T contains the most likely binary words $\hat{\mathbf{c}}$. For instance, in algorithm 2, the word obtained by flipping the $t+1$ least reliable bit of $\hat{\mathbf{c}}^{Maj}$ (which is not in E_T) may be closer to $\mathbf{r}$ than the word obtained by flipping all the t less reliable bits (which is in E_T). We thus propose an algorithm which enables to go through the most likely words: given a number m, we would like E_T to comprise the m words[4] $\hat{\mathbf{c}}^j$ closest to the received word $\mathbf{r}$:

$$E_T = \{\hat{\mathbf{c}}^j / 1 \leqslant j \leqslant m;\ d(\mathbf{r},\hat{\mathbf{c}}^j) \leqslant d(\mathbf{r},\hat{\mathbf{c}}^k)\ \forall k > m\}.$$

Here $d(.,.)$ is the Hamming distance since the demodulated symbols are binary. If the demodulator were soft, we would have used the Euclidean distance instead.

[4] For the sake of simplicity, we use $\hat{\mathbf{c}}$ instead of $C_R(\hat{\mathbf{c}})$ (the word obtained by applying the repetition-code to $\hat{\mathbf{c}}$).

Let $d_i = \min(z_i, R - z_i)$ and $\bar{d}_i = \max(z_i, R - z_i) = d_i + |\lambda_i|$. We have

$$d(\mathbf{r},\hat{\mathbf{c}}^{Maj}) = \sum_{i=1}^{n} d_i$$

and for any binary word $\hat{\mathbf{c}}$

$$d(\mathbf{r},\hat{\mathbf{c}}) = d(\mathbf{r},\hat{\mathbf{c}}^{Maj}) + \sum_{j=1}^{u} |\lambda_{i_j}|$$

with $\{i_1 \dots i_u\}$ being the indices of the bits of $\hat{\mathbf{c}}$ which differ from $\hat{\mathbf{c}}^{Maj}$.

Let $S_{\hat{\mathbf{c}}} = \sum_{j=1}^{u} |\lambda_{i_j}|$. To find the set of words which are closer to $\mathbf{r}$, we thus have to find the smallest $S_{\hat{\mathbf{c}}}$'s. Since an exhaustive search is far too complex, we will show that it is possible to simplify the search by using a partially ordered tree.

3.5.1. Construction of the tree

The tree is built the following way: each node corresponds to a binary word $\hat{\mathbf{c}}$ and its value is $d(\mathbf{r},\hat{\mathbf{c}})$. For each child node N_j of a node N_i, the corresponding word M_j differs from M_i by only one bit. At a given position, a bit cannot be flipped several times along a tree-path, so that every binary word appears in the whole tree only once; and, the values of the branches correspond to the $|\lambda_i|$'s. Moreover, to obtain a partial order between the nodes of the tree, we force the less reliable bits to be flipped before the more reliable ones.

These tedious rules are illustrated in Fig. 6a. We have shown the case of a 3-bits word, each bit being repeated 5 times. The received values are $\mathbf{z} = \{1,5,3\}$. The log-likelihood values are thus: $|\lambda| = \{3,5,1\}$, and the word is decoded by majority logic as $\hat{\mathbf{c}}^{Maj} = \{0,1,1\}$. On the tree we have represented $d(\mathbf{r},\hat{\mathbf{c}})$ inside the nodes and $\hat{\mathbf{c}}$ besides the nodes.

By construction, the nodes issued from the same father (we name them "brother nodes") are sorted by ascending value. It is also obvious that all the descendants of a father node N_i have higher value than N_i. We can also consider "distant relatives": if we consider "cousin nodes" (nodes that share the same grand-father), it can be shown that eldest brothers have smaller values than their younger cousin (a node N_j is younger than an other node N_i if both nodes are on the same level and N_j is on

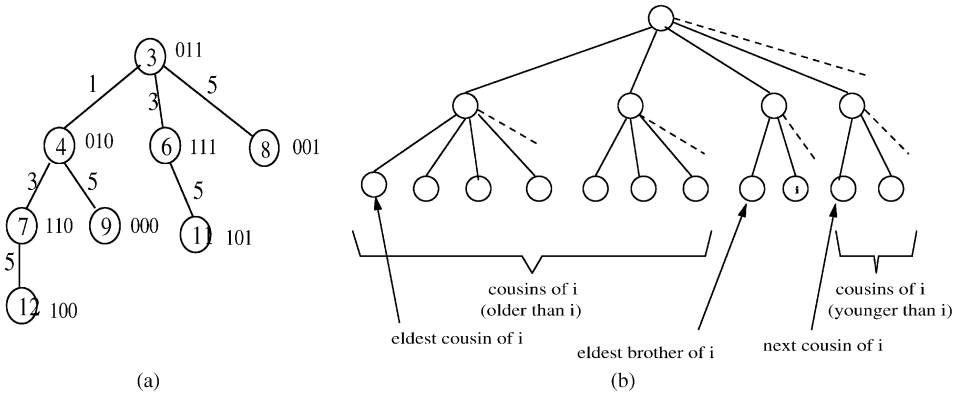

Fig. 6. (a) Example of a tree built from a 3-bit word and, (b) relationships between nodes.

N_i's right). The relationships between nodes are shown in Fig. 6b.

3.5.2. Exploration of the tree

We use two lists: L_1 is the list of the resultant nodes ordered in ascending order, L_2 is the queue containing the nodes susceptible to be placed in L_1. We would like to store as few nodes as possible in L_2 while guaranteeing that L_1 is correctly filled up. We propose the following algorithm:

(1) Store the root node in L_2.
(2) Extract the first node N_i from L_2, put it in L_1, and store in L_2:
   (a) N_i's younger brother whose value is the smallest.
   (b) If N_i is an eldest brother, store its next cousin in L_2.
   (c) If N_i is an eldest cousin, store its eldest child and its next cousin in L_2.
(3) Sort L_2 by ascending order and go to step 2.

It can easily be shown that this algorithm enables to find the m nodes of the tree with smallest values. We will see in Section 5 that this algorithm enables to reduce the complexity of soft-decoding.

4. Error coding for watermarks in Gaussian channel

4.1. Description of the watermarking method

Some watermark detection schemes lead to a Gaussian channel model [5,9]. For example Delaigle et al. [5] embed the watermark bits in 8×8 pixels blocks, in the spatial domain. The underlying principle is to modify the means of groups of pixels so that their mutual differences are equal to a given embedding level. These four groups of pixels are chosen in a judicious way both to control perceptual effects and to guarantee a certain level of secrecy. Perceptual considerations help to adapt the embedding level to the block content, discarding blocks that are not suitable for marking and preserving the mean luminance. The sign of the embedding level depends on the bit to embed.

The high redundancy due to the number of blocks suitable for marking is exploited as follows. These bits are expanded by replication to a length equal to the number of suitable blocks. This longer bitstream is mapped one to one to the blocks in a pseudo-random way. This randomization is realized by interleaving the indexes of the image blocks under a secret key.

The retrieval simply consists in recomputing the difference of the means of the groups after deinterleaving the blocks. Since the decision variable for each bit results from the sum of the above differences from several independent blocks, the resulting statistic can be shown to be Gaussian [9]. In other words the retrieved signal $\mathbf{r}$ is a Gaussian vector with mean $\mathbf{c}$ or equivalently one has an AWGN channel.

4.2. Convolutional codes and soft-decision decoding

Convolutional codes have been extensively used in digital communications systems because of their

good performance and the low computational complexity of the decoders specifically designed for these coding schemes [18]. In fact, convolutional codes can perform better than BCH block codes for similar code rates [14,18]. In addition, convolutional codes are more powerful when they are combined with soft-decision Viterbi decoding, which is in fact the optimum Maximum Likelihood decoding structure for the AWGN channel [14,16,18]. For convolutional codes also, the classical approach to the decoding problem is to use a *hard-decision decoder*. This kind of decoders consist of two steps: first, a binary decision is made for each of the outputs of the AWGN channel, just comparing them with a threshold; then, the resulting bits, also called hard information, are fed into a binary decoder.

However, this is not the optimum approach since there is loss of valuable information in the hard decision step. Better performance can be achieved if we try to design a decoder taking directly the real-valued outputs from the AWGN channel and providing as output the decoded bit sequence. The outputs of the AWGN are also called soft information, and this is the reason why this second strategy is called *soft-decision decoding*.

Soft-decision decoders employed in practice with block codes are computationally complex. Convolutional codes, on the other hand, allow low-cost implementations of soft-decision decoders employing the well-known Viterbi algorithm [14,18]. This is the reason why this kind of codes can be superior to good block codes such as BCH.

Considering the AWGN channel model that applies to the watermarking method we are studying, we can use the channel codes commonly used in communications. In addition, the real-valued outputs r_i, $i \in \{1, \ldots, n\}$ of the equivalent channel are indeed soft information which can be used by a soft-decision decoder. For this reason, convolutional codes combined with a soft-decision Viterbi decoder form a viable alternative to block codes.

In Section 5.2 we have compared the performance of convolutional codes and simple repetition, which is also referred to as uncoded. As an indication we have also computed the BER after hard-decoding of BCH block codes. The BCH code used has similar characteristics and it has already been tested in a previous paper [6].

An interesting effect it is necessary to take into account when using channel codes for error correction purposes is that in general for each coding scheme there is a SNR level below which the code does not introduce a gain in performance. In fact the BER for this interval of SNR values is worse than in the uncoded case. This fact might lead us to think that channel codes are of little interest. However, this is not true, since the BER for the uncoded case is already extremely high for SNR values below this crossing point.

5. Experimental results and conclusions

5.1. Soft-decoding of block codes

The Chase algorithm and the tree algorithm have been tested on video sequences. The 64-bit message was encoded with a (127,64) BCH, then the resulting code-word was repeated 51 times. The coded message was then embedded into the image using the watermarking method described in Section 3.1.

For experiments on moving pictures, we used 13 ITU-R BT.601 sequences[5] of various kinds (natural, synthetic or cartoon images) gathered in 11,000 frames. The composite video has been MPEG-2 encoded at 3, 3.5, 4, 5 and 6 Mbs. The search depth of the tree algorithm is $m = 1024$ to allow fair comparison with the Chase algorithm. Experimental results are shown in Figs. 7 and 8. We note that soft-decoding algorithms enable to improve the watermark retrieval, although it also increases the bit error rate. The BER is relatively high, but it can be reduced by controlling the global distance between the received word and the decoded word, and by exploiting the inter-image redundancy (the watermark is the same for pictures of the same sequence). Performances of the tree algorithm are approximately the same than performances of the Chase algorithm, but the complexity is significantly smaller (see Fig. 9).

[5] ITU-R BT.601: Standard television format = 576×720 pixels.

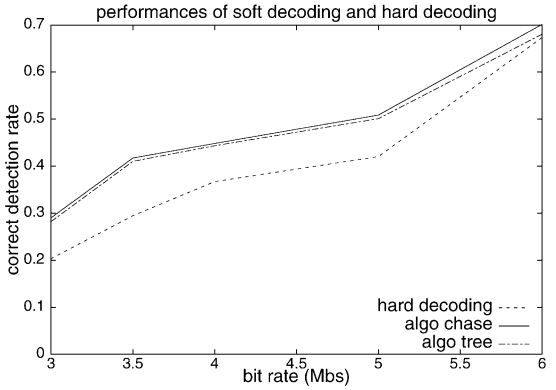

Fig. 7. Probability of correct decoding for soft and hard decoding: the Chase algorithm and the tree algorithm approximately achieve the same performances.

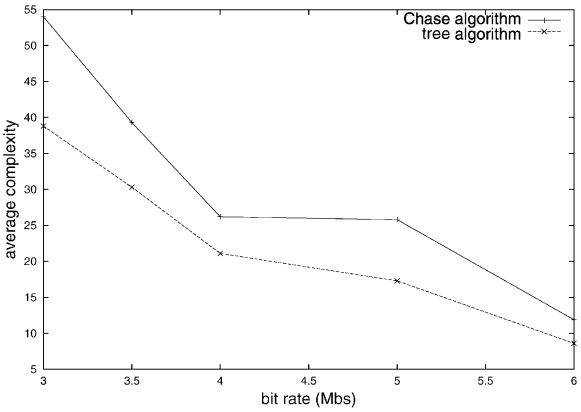

Fig. 9. Complexity of soft-decoding: average number of hard decodings performed per soft-decoding.

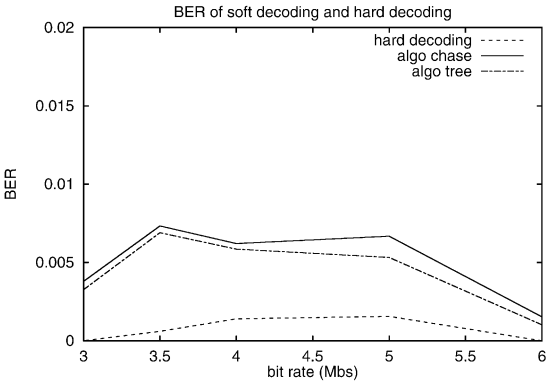

Fig. 8. BER of soft and hard-decoding: the BER is slightly better for the tree algorithm than for the Chase algorithm.

5.2. Soft-decoding of convolutional codes

In Fig. 11 we show plots of the BER measured empirically when the watermarked image is JPEG compressed with different quality levels. The tests were performed on the images shown in Fig. 10. We have represented one BER curve corresponding to the uncoded case, and two curves corresponding to a BCH (127,64) code and a rate $\frac{1}{2}$ convolutional code with constraint length $v = 8$. In all cases the message length is 64 bits and the empirical measures have been obtained by averaging out 50 different keys randomly taken. We can clearly see for all the test images how the BER curves cross each other, as discussed in Section 4.2. Note that convolutional coding performs better than BCH coding and the uncoded case only when the JPEG compression quality level is above a certain minimum value. However, note also that below this quality level the BER for the uncoded case and the BCH coding scheme are already very high (around 0.05) and that for quality levels above this minimum value the BER curve for the convolutional code decreases much faster.

5.3. Conclusions

We have studied the protection of watermarks by error correcting codes. In the case of hard-decision for the received watermark sequences, the equivalent watermarking signal can be considered as a stationary binary symmetric channel. The watermark channel may have to operate at very high bit error rates, that is $0.1 < \mathrm{BER} < 0.5$. Under such severe conditions 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 decoding performance in this critical range.

When using repetition coding in a binary symmetric channel, the bit decision variables possess

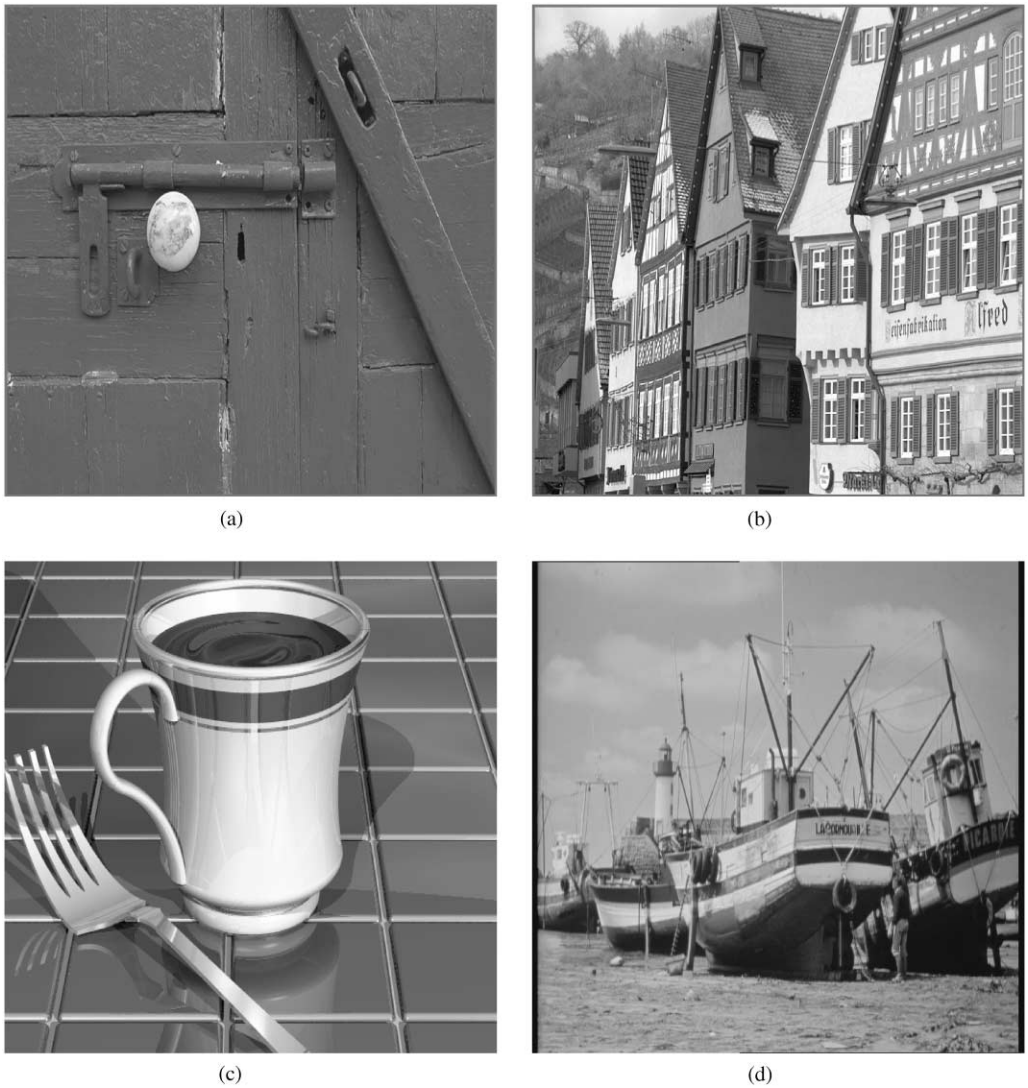

Fig. 10. Test images: (a) 02 (768 × 512), (b) 08 (768 × 512), (c) cup (640 × 480) and (d) boats (720 × 576).

binomial distributions. Using this information soft-decoding becomes possible. A fast soft-decoding algorithm for BCH codes has been developed and its effectiveness shown.

In the case the watermark extraction is based on the sum of several independent variables an AWGN channel can be assumed for which an efficient decoding is achieved using soft-Viterbi decoding of correlated sequences.

So, we have studied repetition versus BCH codes on the basis of analytical results, we have investigated the performance of soft BCH decoding and soft Viterbi decoding trough simulation results. These effectiveness of these error protection schemes has been analyzed under JPEG/MPEG compression attacks. The performance under a more extensive set of attacks, such as geometrical distortion, will be studied in the future.

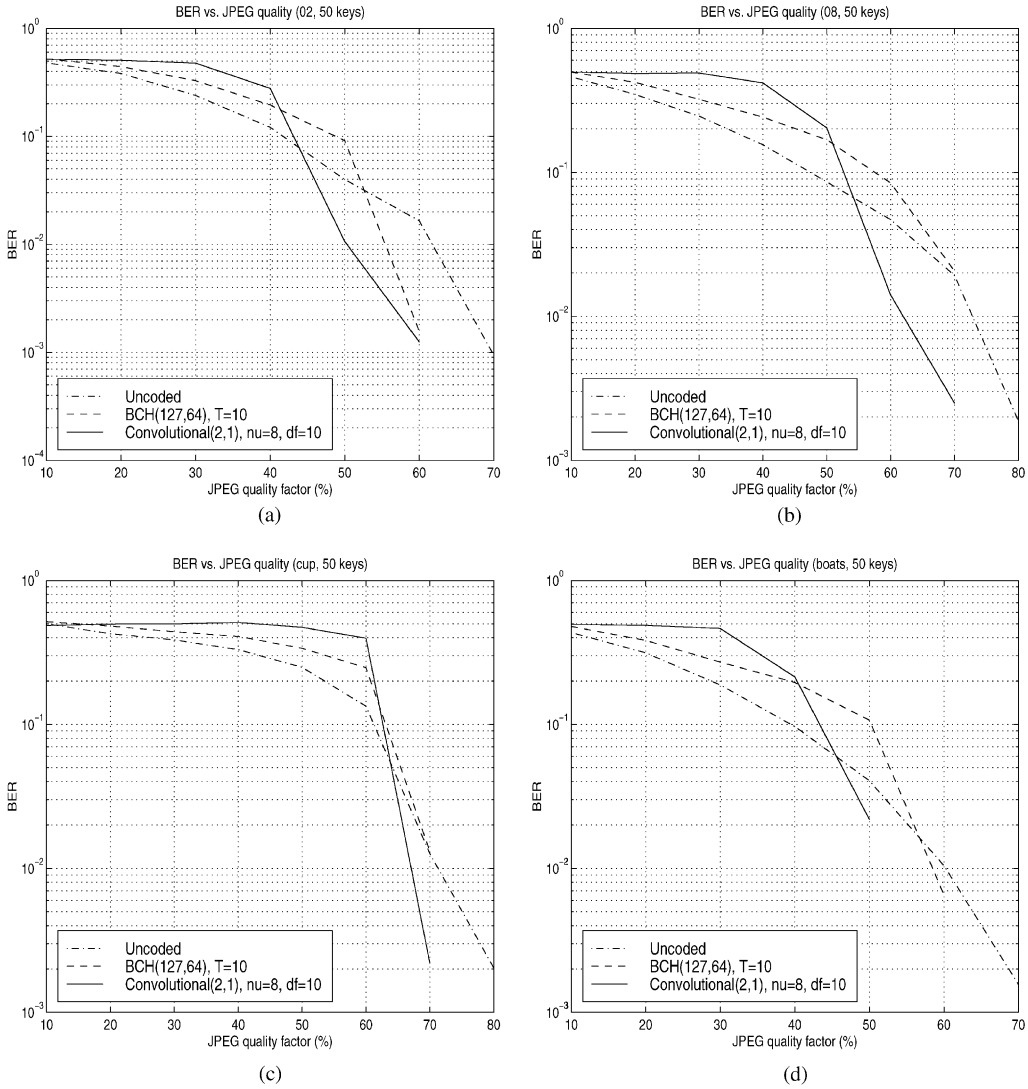

Fig. 11. Empirical BER for different values of JPEG quality factor, the test images: (a) 02, (b) 08, (c) cup and (d) boats.

References

[1] S. Burgett, E. Koch, J. Zhao, Copyright labelling of digitized image data, IEEE Commun. Magazine 32 (1998) 94–100.

[2] D. Chase, A class of algorithms for decoding block codes with channel measurement information, IEEE Trans. Informat. Theory 18 (1) (January 1972) 170–182.

[3] I.J. Cox, J. Kilian, T. Leighton, T. Shamoon, Secure spread spectrum watermarking for images, audio and video, in: IEEE-ICIP'96, Lausanne, Switzerland, Vol. III, October 1996, pp. 243–246.

[4] I.J. Cox, M.L. Miller, A.L. McKellips, Watermarking as communications with side information, Proc. IEEE (Special Issue on Identification and Protection of Multimedia Information) 87 (July 1999) 1127–1141.

[5] V. Darmstaedter, J.F. Delaigle, J.J. Quisquater, B. Macq, Low cost spatial watermarking, Comput. Graphics 22 (4) (July 1998).

[6] V. Darmstaedter, J.F. Delaigle, J.J. Quisquater, B. Macq, Low cost spatial watermarking, Comput. Graphics 22 (4) (1998) 417–424.

[7] J.F. Delaigle, C. De Vleeschouwer, B. Macq, Digital watermarking, in: R. von Renesse (Ed.), Conference 2659—Optical Security and Counterfeit Deterrence Techniques, San Jose, February 1996, SPIE Electronic Imaging: Science and Technology, pp. 99–110.

[8] J.F. Delaigle, C. De Vleeschouwer, B. Macq, Watermarking algorithm based on a human visual model, Signal Process. 66 (3) (May 1998) 319–335.

[9] J.R. Hernandez, J.F. Delaigle, B. Macq, Improving data hiding by using convolutional codes and soft-decision decoding, in: Ping Pah Wong, E.J. Delp (Eds.), Security and Watermarking of Multimedia Contents, SPIE, Vol. 3971, San-Jose, CA, USA, January 2000.

[10] J.R. Hernández, F. Pérez-González, Statistical analysis of watermarking schemes for copyright protection of images, Proc. IEEE (Special Issue on Identification and Protection of Multimedia Information) 87 (7) (July 1999) 1142–1166.

[11] J.R. Hernandez, F. Perez-Gonzalez, Statistical analysis of watermarking schemes for copyright protection of images, in: Ping Pah Wong, E.J. Delp (Eds.), Security and Watermarking of Multimedia Contents, SPIE, Vol. 3971, San-Jose, CA, USA, January 2000.

[12] J.R. Hernandez, F. Perez-Gonzalez, J.M. Rodriguez, The impact of channel coding on the performance of spatial watermarking for copyrights protection, in: IEEE ICASSP'98, Seattle, USA, Vol. 5, May 1998, pp. 2973–2976.

[13] E. Koch, J. Zhao, Towards robust and hidden image copyright labeling, in: IEEE (Ed.), Non Linear Signal Processing Workshop, Thessaloniki, Greece, October 1995, pp. 452–455.

[14] E.A. Lee, D.G. Messerschmitt, Digital Communication, Kluwer Academic Publishers, Dordrecht, 1988.

[15] F.J. MacWilliams, N.J.A. Sloane, The Theory of Error-Correcting Codes, Elsevier, Amsterdam, 1977.

[16] D.R. Martin, P.L. McAdam, Convolutional code performance with optimal jamming, in: Conf. Rec. Int. Conf. Commun., May 1980, pp. 4.3.1–4.3.7.

[17] D. Mukherjee, J.J. Chae, S.K. Mitra, A source and channel coding approach to data hiding, in: ICIP 98, October 1998.

[18] B. Sklar, Digital Communications, Fundamentals and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[19] R.B. Wolfgang, E.J. Delp, A watermark for digital images, in: IEEE-ICIP'96, Lausanne, Switzerland, Vol. III, 1996, pp. 219–222.