Enhanced Robustness in Image Watermarking using Block Turbo Codes

Christian Rey [a], Karine Amis [b], Jean-Luc Dugelay [a], Ramesh Pyndiah [b] and Annie Picart [b],

[a] Institut Eurécom, dept. of Multimedia Communications, Sophia Antipolis.
http://www.eurecom.fr/~image
{rey, jld}@eurecom.fr
[b] ENST Bretagne, Brest, France
http://www.enst-bretagne.fr/
{karine.amis, ramesh.pyndiah, annie.picart}@enst-bretagne.fr

ABSTRACT

In this paper, we investigate the improvement achieved by turbo coding for robustness in still image watermarking. We use an error correcting scheme based on the concatenation of a BCH product code and a repetition code. The product code is iteratively decoded using the Chase-Pyndiah algorithm (turbo decoded). For this study, we set the watermarking distortion to around 38 dB and we consider different payloads that can correspond to different applications and services. We compare different coding strategies (i.e. repetition code only and a concatenation of product and repetition codes) in terms of robustness to different photometric attacks, in particular additive noise or lossy compression. Typically, for a payload of 121 bits, the robustness gain for a given message error probability is significant: first error appears at a JPEG quality factor of about 25% with the new coding scheme instead of about 50% when using only repetition codes.

Keywords: Image, Watermarking, Error Correcting Codes, Product Codes, Turbo Codes, Robustness.

1. INTRODUCTION

The explosion of digital multimedia documents circulating worldwide through communication networks (e.g. WWW or WLAN) or commercial networks (e.g. CD or DVD) has created a strong demand for efficient watermarking techniques in order to protect these documents. Watermarking[1] includes a complex trade-off between three parameters: distortion, payload and robustness. Different watermarking solutions have already been proposed in the literature. One major challenge is to design a technology that offers an efficient robustness while keeping a fair distortion and payload. The range of possible attacks on images is very large. One possibility to overcome this problem could consist in specifying algorithms relatively to a given category of attacks (e.g. JPEG[2] or shifting and scaling[3]) or to a specific service (e.g. including a very low payload).

The watermarking scheme considered in this paper is based on self-similarities, and uses a blind mode of extraction (which is mandatory for most of practical applications). In its basic version[4] (i.e. without any advanced error correcting coding scheme), the considered approach already includes an efficient and generic robustness against most of both possible photometric and geometric attacks (including Stirmark[5]) but with some limitations. For example, if one considers JPEG compression in particular, our technology may have some difficulties when dealing with factor of quality lower than 50%.

The motivation of this work is then to globally increase performances of the watermarking scheme in terms of robustness against some specific photometric attacks (such as lossy compression) through the use of Error Correcting Codes (EEC), while keeping at the same level other parameters in terms of visibility and payload, and without reducing its performances against other attacks (i.e. without specialization).

EEC have been successfully used in digital communication systems and data storage applications in order to achieve reliable transmission on a noisy channel. However, the introduction of EEC in this application is not straightforward and requires a specific design to cope with the characteristics of the watermarking technique.

The paper is organized as follows: We first shortly described in Section 2 the main steps involved in the watermarking technology used here. One weakness of the considered approach is that the insertion probability (i.e. host pixels) is quite low (less than 1 out of 2). Powerful error correcting codes are required to recover the non-inserted bits and to improve robustness against some photometric attacks. In Section 3, we draw a short panorama of some previous works in the aforementioned topic and then gradually introduce the ECC framework investigated more particularly in this article. The Section 4 describes our proposed watermarking scheme with the optimised ECC, namely block turbo codes. In Section 5, we report a set of selected simulation results that clearly demonstrate that ECC can play a significant role for enhancing robustness in image watermarking. Finally, we conclude by some remarks and perspectives in Section 6.

2. WATERMARKING

2.1 Basic Recalls on Watermarking

Image watermarking is now a major domain within multimedia processing. Basically, digital watermarking gives the possibility to owners (or providers) to hide an invisible and robust message inside multimedia content, often for security purposes, in particular for owner or content authentication. There exists a complex trade-off between three parameters in digital watermarking: payload, watermarking distortion and robustness. Robustness means that the retriever is still able to recover the hidden message even if the watermarked content has been altered by additive noise, JPEG compression or geometrical attacks.

2.2 Eurécom's core technology

The simulation results reported in this paper have been performed using the Eurecom's still image watermarking technology[6] that can be summarized as follows. The considered approach is derived from fractal image coding theory, in particular the notion of self-similarity. The main idea is to use some invariance properties of fractal coding, such as invariance by local affine (geometric and photometric) transformations, to ensure watermark robustness. The extraction mode is supposed to be blind.

Signer. The watermark embedding process can be described by the following three steps: formatting and encryption of the message to hide, cover generation, merging of the watermark and the cover.

(i) Formatting and encryption of the watermark.

The message bits to be hidden are redundantly distributed: by over-sampling and duplication of the message to obtain a watermark of the size of the image. This redundancy is necessary for a good robustness. Finally, the watermark is globally encrypted using an XOR with a pseudo-random binary sequence generated using a secret key, yielding the encrypted watermark *W*. The XOR operation enables, on one hand, to secure the hidden message, and on the other hand, to remove repetitive patterns reducing in this way the psycho-visual impact of the watermark embedding.

(ii) Cover generation

First, a "fractal approximation" I_{approx} is computed from the original image I_{original}. The cover I_{cover} corresponds to the error image that is the signed difference between the original image and its fractal approximation.

$$I_{\text{cover}} = I_{\text{original}} - I_{\text{approx}} \tag{1}$$

(iii) Merging of the watermark and the cover

The last step of the embedding process consists in modulating the cover I_{cover} with *W*. The modulation consists of zeroing some of the cover pixels depending on their sign and the corresponding watermark bit to hide. In some other words, only one bit out of two in average really carries hidden information. For visibility reasons, pixels in I_{cover} superior to a given threshold will not host hidden data. Finally, the modulated cover $\hat{I}_{\text{cover}}$ is added to the fractal approximation I_{approx} to produce the watermarked image $I_{\text{watermarked}}$.

$$I_{\text{watermarked}} = I_{\text{approx}} + \hat{I}_{\text{cover}} \tag{2}$$

In average, one pixel out of two of the resulting picture is either equal to its original value or its approximated one.

Retriever. The watermark extraction algorithm is similar to the embedding algorithm (i.e. dual operations) and its complexity is very similar too. First a fractal approximation is calculated from the watermarked image, which generates a cover close to the original one. Finally the cover is decoded according to the modulation rules (e.g. a positive pixel is supposed to carry a one valued bit and a negative pixel a zero valued bit). The crucial point is that most geometric transformations on the watermarked image are also transferred to the cover: the mark is not lost but the noise has to be correctly positioned with respect to the cover before applying XOR. Therefore, some additional bits called 'resynchronization bits' are added to the useful message bits in order to allow a self and blind resynchronization of samples via two procedures: one for global geometric distortion (rotation and rescaling) based on FFT properties of periodic signals and Hough transform, one for local geometric distortion based on block-matching. Then the watermark can be decrypted and the message recovered.

2.3 Use of Error Correcting Codes in watermarking

Since early nineties, a wide range of algorithms and techniques for still image watermarking have been proposed, emerging from various communities. Most algorithms can be characterized according to the strategies used for (i) formatting the message to be hidden, (ii) the choice of a workspace to perform the hiding operation, and then (iii) the method of merging the message and the cover. In general, steps (i) and (ii) are performed in parallel. By default, ECC are mainly involved in step (i). Currently, most of watermarking techniques use, in order to guarantee a minimum of robustness, a redundancy based on a basic repetition of each bit to be hidden. For the technology herein used for our test, the formatting step performed on the payload *msg1* can be summarized as follows:

(i) *msg1* is over-sampled as illustrated in figure 1 in order to obtain a *msg2*;

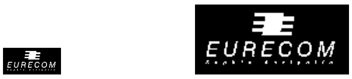

Figure 1: over-sampling of the binary sequence *msg1*

(ii) *msg2* is then duplicated till the size of the picture as illustrated in figure 2 in order to obtain *msg3*;

(iii) *msg3* is globally encrypted (i.e. XOR with a pseudo random binary sequence) as illustrated in figure 3 in order to finally obtain *msg4*.

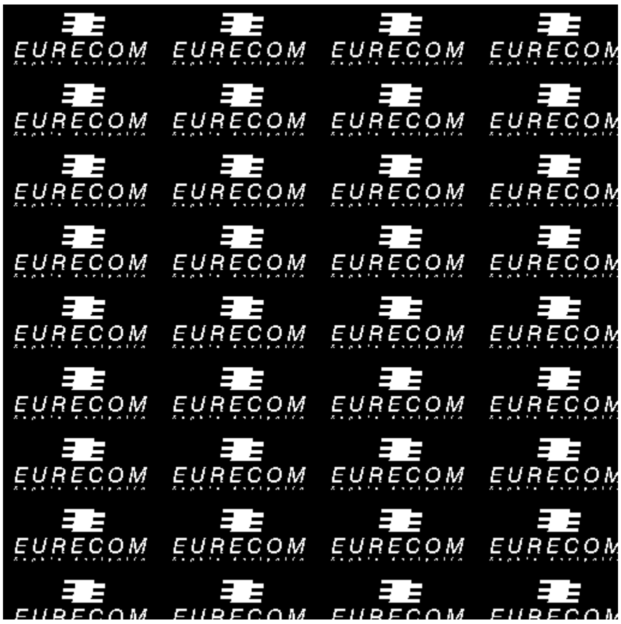

Figure 2: message duplicated

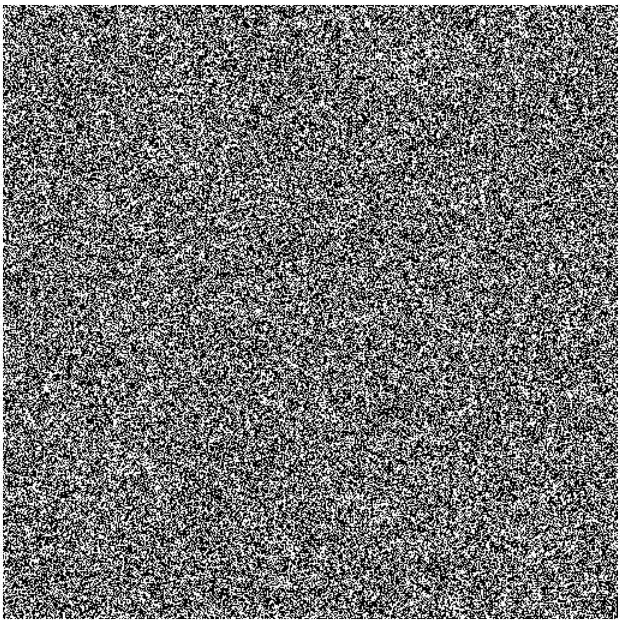

Figure 3: pseudo random binary sequence

In order to improve robustness against attacks including more or less complex impacts on luminance values[7], an error correcting code can be inserted prior to over-sampling and duplication of the message. It seems obvious that it would be more efficient to use correcting codes instead of a simple duplication of bits. Indeed, this approach appears natural if one compares the watermarking problem with the transmission of a signal over a noisy channel. This model considers the cover as the channel, the watermark as the message to send, and the different attacks as noise. Error-correcting codes are widely used in digital communications which should make them also useful for watermarking issues. Nevertheless, early studies in this field unfortunately report that the present problem is not trivial in practice (correcting codes have not originally been designed for such a context). Basically, the introduction of an error correcting code in the formatting of the watermark increases the number of bits to hide; consequently the number of repetitions of each bit of the watermark decreases in the same proportion.

3. ERROR CORRECTING CODES IN WATERMARKING

3.1 Previous works

Early watermarking techniques used repetition codes to protect the watermark against possible attacks intentional or not. Repetition codes are not efficient in terms of error correction since they exhibit a coding gain of zero dB. Therefore, in a more recent literature, one can find watermarking techniques using more powerful error correcting codes, such as convolutional codes[8], BCH codes[9,10,11] or concatenated codes based on a convolutional code followed by a Reed-Solomon code[12] or even Convolutional Turbo Codes[13,14,15]. Most of the tests were carried out in a JPEG attack context. For very low JPEG quality (under 20%), repetition codes seem the best solution. Note that for such low values, the image is no more exploitable. For the watermarking technique considered here, very powerful and efficient error correcting codes are required to optimize its robustness against some specific attacks (e.g. JPEG or noise) as this technique has been originally designed for a generic and default context of payload, visibility and robustness against attacks, malicious or not (with a special emphasis on geometrical ones). Product codes using iterative decoding techniques[16] are an attractive solution for this application as they achieve high error-correction capability with reasonable decoding complexity.

3.2 Product codes

Introduced by Elias in 1954, product codes are a serial concatenation of two or more block codes. Let us consider the product code $C_1 \otimes C_2$, resulting from the concatenation of $C_1(n_1,k_1,\delta_1)$ and $C_2(n_2,k_2,\delta_2)$, where n_i, k_i and δ_i respectively denote the code length, its dimension and its minimum Hamming distance. A product code word is obtained as follows. The $k_1 \times k_2$ matrix information bits are placed in a $k_1 \times k_2$ matrix. Each row (respectively column) of the matrix is first encoded by C_2 (respectively C_1). The resulting n_2 columns (respectively n_1 rows) are then encoded by C_1 (respectively C_2). An interesting property of product codes is that all rows are code words of C_2 and all columns code words of C_1. The product code parameters are the product of the corresponding elementary code parameters. Another interesting feature of such codes is their low decoding complexity. Indeed an elementary block code having the same minimum distance δ exhibits a decoding complexity in $2^{\lfloor \delta/2 \rfloor}$. The product code can be decoded by first decoding all rows and then decoding all columns (or vice versa) and its decoding complexity is thus given by $2^{\lfloor \xi/2 \rfloor}$, with $\xi = \max(\delta_1,\delta_2)$.

3.3 The Chase algorithm

Block codes can be algebraically decoded using Petersons's direct method or the Berlekamp algorithm. This hard decoding is not optimal in the case of soft channel outputs. In this case, the optimal algorithm consists in minimizing the Euclidean distance between the observation and the code word set. Chase proposed an algorithm that achieves near-optimum decoding. The Maximum Likelihood (ML) decoding looks for the most probable code word among all possible code words. Chase proposed to reduce the searching space to a sphere centered on the hard observation. From this hard observation, a test sequence is built by commuting the least reliable positions. Each test sequence is then algebraically decoded. Among the resulting code word sub-set, the one that minimizes the Euclidean distance with the observation is selected as the decision.

3.4 Soft product code decoding : the Chase-Pyndiah algorithm

The near-optimal soft-decoding of product codes was first proposed by Pyndiah[17] and is an extension of the Turbo iterative principle introduced by Berrou for parallel concatenation of two convolutional codes[18]. One decoding iteration consists of a decoding of all rows followed by a decoding of all columns or vice versa. Each elementary decoder uses the Chase algorithm and at each step, from the resulting code words sub-set, the most probable code word is selected and for each position, a competing code word is searched. These two code words are used to compute a metric, approximation of the Logarithm of the Likelihood Ratio (LLR) corresponding to the current position. The extrinsic information is extracted from this LLR value to feed the next decoder input.

4. PROPOSED FRAMEWORK

Let us consider a payload of $k_1 \times k_2$ binary elements. The payload is encoded by product code $C_1 \otimes C_2$. Then synchronization bits are introduced to fight against geometric attacks and the resulting matrix is called **S**. Each bit of **S** is locally repeated by a factor ($e_1 \times e_2$) resulting in matrix **P**. This over-sampling aims at increasing robustness against low-pass filtering. **P** is then replicated L times to fit the cover image size. Finally the binary elements of the resulting matrix **X** are summed modulo-2 to a pseudo-random binary sequence in order to obtain a matrix **W** of independent and identically distributed binary elements. This step is crucial for security in order to prevent malicious modifications of the watermark. The resulting formatted mark **W** is then inserted in the cover following the Eurécom method explained in Section 1.

At the receiver side, after the watermark extraction and the resynchronization procedure, the soft values corresponding to each coded bit are processed to build an $n_1 \times n_2$ observation matrix **R**. **R** is turbo-decoded according to the iterative process described in Section 2. The decoded message gives an estimation of the original payload.

A compromise must be found between payload size ($k_1 \times k_2$), code length ($n_1 \times n_2$), over-sampling factor ($e_1 \times e_2$) and replication factor L for a given image size ($N_1 \times N_2$) and average number of synchronization bits introduced per coded bit (α). For a given image size, over-sampling factor, code length and average number of synchronization bits introduced per coded bit, the replication factor is upper bounded by:

$$L \leq \left(\frac{N_1 \times N_2}{n_1 \times n_2 \times (1+\alpha) \times e_1 \times e_2} \right) \tag{3}$$

In our application, ($N_1 = N_2 = 512$; $e_1 = e_2 = 3$; $\alpha = 1$). For a code length ($n_1 \times \mathrm{n}_2 = 16 \times 16 = 256$), we obtain $L < 57$. For a code length ($n_1 \times \mathrm{n}_2 = 32 \times 32 = 1024$), we obtain L<15. Statistically, the probability that a bit of the watermark is inserted in the image is less or equal to 50% (see Section 2.2). Moreover, so as to preserve the original image quality, that is to say guarantee the watermark invisibility, a threshold is used to prevent insertions leading to large distortions. This will again reduce the insertion probability (25-50%). A sufficiently large replication factor L is therefore required to combat insertion loss as well as image cropping.

Furthermore, L is also lower bounded by unity and thus the code length is upper bounded by:

$$(n_1 \times n_2) \leq \left(\frac{N_1 \times N_2}{(1+\alpha) \times e_1 \times e_2} \right) \tag{4}$$

In our application ($n_1 \times n_2$) ≤ 120^2. This limits the range of codes (16, 32, 64) used in this application. For a given code length, we can use different code dimensions to adapt the payload size. Simulation results are presented in next Section.

5. PRELIMINARY RESULTS

In this section, first results performed on a database of 75 images of size 512×512 pixels are presented. All the results are expressed in terms of message error rate (MER). We define the MER as the ratio of the number of erroneous watermarks on the number of extracted watermarks. A watermark is erroneous so at least one of its bits is false. In figure 4 we show the experimentally measured MER versus the quality level of the JPEG compression applied to the watermarked images. The payload was set to 121 bits. In this figure, several curves have been plotted corresponding to different coding (and decoding) strategies (repetition code only, BCH(128,120), BCH(256,123) and product code BCH(16,11)^2). For any given JPEG quality, concatenation of repetition and product codes using soft-decoding algorithm appears to be the best solution. The watermark is recovered without any bit error down to 30% JPEG quality, while some errors are observed at 50% JPEG quality when using only repetitions. Concatenation of repetition and BCH codes also performs better than a simple repetition code, but it is less efficient than concatenation of repetition and product codes. Moreover, when the repetition codes are hard-decoded, the performances of the turbo decoder decrease significantly.

Figure 5 and 6 shows Message Error Rate performances against JPEG compression and additive gaussian noise for different payloads (i.e. 77, 121, 441 and 676 bits) and code lengths using concatenation of repetition and product codes. For constant code lengths (256 or 1024) we observe that there is a mild degradation as we increase the pay load. As we increase the pay load at constant code length, the error correction capability decreases and results in a lower coding gain. On the other hand, there is a much larger degradation when the code length is increased for codes with the same minimum distance ($(16,11)^2$ and $(32,26)^2$). Theoretically, as we increase code length for a given minimum distance the coding gain increases. However, this coding gain is not sufficient to compensate the loss due to the decrease in the replication factor L which is here divided by 4. The received signal energy per information bit is divided by 4 and the insertion probability of a given information bit varies as $(1\text{-}(1\text{-}p)^{-L})$ where p is the insertion probability per pixel. Thus as we increase code length in order to achieve higher pay loads, there is a significant penalty in terms of robustness.

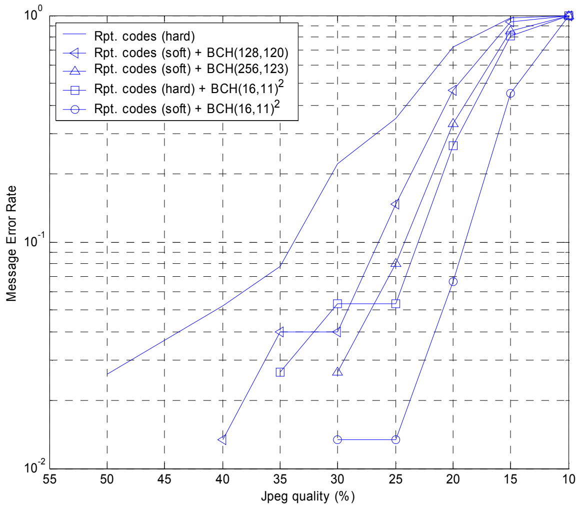

Figure 4: Message error rate versus JPEG quality for different coding (and decoding) schemes

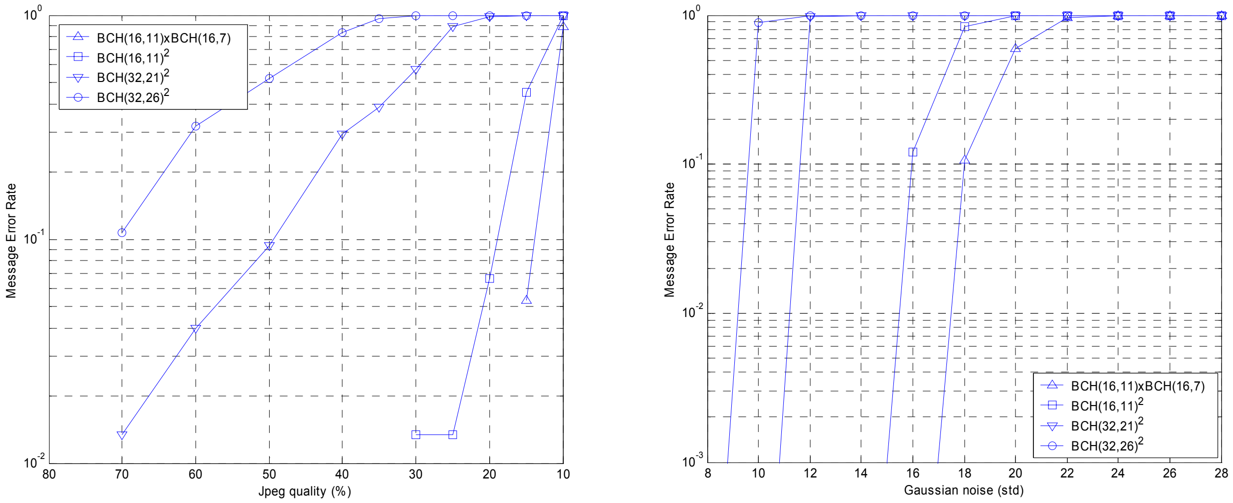

Figure 5: Message error rate versus JPEG quality (left) and gaussian noise (right), for different payloads and code lengths.

6. CONCLUDING REMARKS

In this paper, we have proposed an enhanced still image watermarking scheme using powerful error correction block turbo codes to improve its resistance to photometric attacks. Simulation results show that for a fixed distortion (by default fixed to 38dB) and a given payload (here 121), the message error rate is still equal to zero down to 30% of quality instead of 50% without product codes. For different tested payloads and attacks (JPEG or additive Gaussian noise), there is a significant global gain. Our results confirm previous ones reported by Kesal et al. that BTC coding scheme outperforms all other coding solutions for watermarking applications.

Our perspectives will focus on the study of a channel model in order to determine the theoretical capacity, and then to compare simulation results with theoretical performances. Also, we plan to investigate how obtained results on error correcting codes can be extended to other watermarking technologies. Finally, some possibly more powerful product codes and flexible BTC in terms of payloads and code rates will be investigated.

REFERENCES

1. S. Katzenbeisser and F. Petitcolas Eds., 'Handbook on Information hiding techniques for steganography and digital watermarking', Artech House, 2000.
2. E. Koch and J. Zhao, 'Towards robust and hidden image copyright labeling', IEEE wksp. on NSIP, pp. 456-459, 1995.
3. S. Tsekeridou, N. Nikolaidis, N. Sidiropoulos and I. Pitas, 'Copyright protection of Still Images using Self-similar Chaotic Watermarks', IEEE Int. Conf. on Image Processing,2000.
4. C. Rey and J.-L. Dugelay, 'Tech. demo.: Image Watermarking for Owner and Content Authentication', ACM Multimedia, Los Angeles, Nov. 2000.
5. J.-L. Dugelay and F. Petitcolas, 'Image watermarking: possible counterattacks against random geometric distortions', SPIE, Vol. 3971, Security and Watermarking of Multimedia Contents II, San Jose, Jan. 2000.
6. J.-L. Dugelay and C. Rey, 'Method of marking a multimedia document having improved robustness', Pending Patent EUP 99480753 (EURECOM 14 EP), May 2001.
7. S. Voloshynovskiy, S. Pereira, A. Herrigel, N. Baumgartner and T. Pun, 'Generalized watermarking attack based on watermark estimation and perceptual remodulation', SPIE, Security and Watermarking of Multimedia Contents II, pp. 24-48, San Jose, USA, Jan. 2000.
8. J.R. Hernández, J.-F. Delaigle and B. Macq, 'Improving Data Hiding by Using Convolutional Codes and Soft-Decision Decoding', SPIE, - Security and Watermarking of Multimedia Contents II, pp. 24-48, San Jose, USA, Jan. 2000.
9. A. Tefas and I. Pitas, 'Multi-bit image watermarking robust to geometric distortions', IEEE Int. Conf. on Image Processing, Vancouver, Canada, Sep. 2000.
10. J. Darbon, B. Sankur and H. Maître, 'Error Correcting Code Performance for Watermark Protection', SPIE, Security and Watermarking of Multimedia Contents III, San Jose, USA, Jan. 2000.
11. S. Baudry, J.-F. Delaigle, B. Sankur, B. Macq and H. Maître, 'Analyses of error correction strategies for typical communication channels in watermarking', EURASIP Signal Processing, Vol. 81, n° 6, pp. 1239-1250, June 2001.
12. T. Brandão, M.P. Queluz and A. Rodrigues, 'Improving Spread Spectrum Based Image Watermarking Through Non-binary Channel Coding', Proc. of 3 Conference on Telecommunications, Figueira da Fez, Apr. 2001.
13. M. Kesal, M.K. Mıhçak, R. Koetter and P. Moulin, 'Iteratively Decodable Codes for Watermarking Applications', Proc. of 2 Int. Symp. on Turbo Codes and Related Topics, Brest, France, Sep. 2000.
14. S. Pereira, S. Voloshynovskiy and T. Pun, 'Effective Channel Coding for DCT Watermarks', IEEE Int. Conf. on Image Processing, Vancouver, Canada, Sep. 2000.
15. F. Balado, F. Pérez-González and S. Scalise, 'Turbo Coding for Sample-level Watermarking in the DCT domain', Proceedings of IEEE Int. Conf. on Image Processing, pp. 1003-1006, Thessaloniki, Greece, Oct. 2001.
16. R. Pyndiah, 'Near optimum decoding of product codes: block turbo codes', *IEEE Trans. on Communications,* Vol. 46, n° 8, Aug. 1998.
17. R. Pyndiah, A. Glavieux, A. Picart and S. Jacq, 'Near-Optimum Decoding of Product Codes', Globecom'94, vol. 1/3, Nov. –Dec. 1994.
18. C. Berrou and A. Glavieux, 'Near Shannon limit Error Correcting Coding and Decoding: Turbo Codes', IEEE ICC, vol. 2/3, May 1993.