Pergamon

Comput. & Graphics, Vol. 22, No. 4, pp. 417–424, 1998

Printed in Great Britain
0097-8493/98 $19.00 + 0.00

PII: S0097-8493(98)00031-4

Data Security in Image Communications and Networking

LOW COST SPATIAL WATERMARKING

V. DARMSTAEDTER†, J.-F. DELAIGLE, J. J. QUISQUATER and B. MACQ

Laboratoire de Télécommunication et Télédétection, Université catholique de Louvain, Bâtiment Stévin, Place du Levant, 2, B-1348 Louvain-la-Neuve, Belgium

Abstract—This paper presents a block-based spatial watermarking method. The main interest of this method is that it attempts to realise a good compromise between robustness performance, quality of the embedding and computational cost. This method is based on elementary perceptual criterion and adapts the embedding with regards to the content of the blocks. The paper ends with results deduced from many perceptual tests. The robustness towards lossy compression is also deeply analysed. © 1998 Elsevier Science Ltd. All rights reserved

1. INTRODUCTION

From the content owner point of view it is not clear that classical legal copyright protections are sufficient. There is a great demand for new technologies specifically suited for digital media [1]. As a matter of fact the nature of digital media threatens its own viability. Replications of digital works are easy and identical to original contents. The plasticity of digital media is also a great menace. Fortunately, digitisation of audiovisual content offer in counterpart new possibilities for the development of copyright protection techniques. Watermarking is one of these. The principle of watermarking is the robust and secret embedding of copyright information in a content. This content may be as text [2, 3], or audio content [4], but most of the time watermarking is applied to still or moving pictures.

A large number of watermarking techniques using different approaches already exist. Most of them directly modify the luminance [5, 6] or the chrominance [7]. Such methods have the advantage of being very fast. It is also interesting to consider the image content as a broadband channel that can convey a narrow bandwidth information. Different techniques use this approach by means of the spread-spectrum theory to embed a copyright code [8]. Some authors apply the watermarking on some characteristics of the image, such as DCT coefficients [9], fractals coefficients, motion estimation vector, high resolution coefficients in case of multiresolution encoding or the phase of DFT coefficients [10]. Finally, other authors propose methods based on Human Visual System model to guarantee the invisibility of the embedding [11, 12].

Such very elaborated techniques have the advantage of realising a good trade-off between robustness and quality of the watermarked image. Nevertheless, their main drawback is their complexity and computational cost. This can be annoying for video sequences since real-time embedding is a major issue. The watermarking method presented in this paper has been developed in the scope of the European ACTS project TALISMAN (AC 019) [13]. The main purpose of this project is to make near real-time watermark embedding and real-time retrieval of watermarked copyright information. The goal is to provide facilities for monitoring broadcast networks in order to detect copyright violations. With this aim in view, real-time retrieval is crucial.

This paper presents a low computational cost watermarking technique for still pictures and for video sequences. The embedding process consists of embedding redundantly a copyright information (typically 64 bits) in a digital picture with the use of a secret key. The retrieval process allows the copyright information to be retrieved without the use of the original image. There are a few parameters in the algorithm, many perceptual tests have been achieved so as to determine the optimal solution. Many tests were also performed in order to analyze the robustness of the watermark, such as JPEG and MPEG compression etc.

2. EMBEDDING

2.1. *Introduction*

The presented embedding method is block-based. Before being embedded, the copyright information is converted into a bitstream. Each bit is spread over one block. In the current implementation, the block size is 8×8. The main purpose of this choice is that it corresponds to the size of JPEG blocks. So the effects of JPEG compression affect independently each embedded bit. Besides, this allows to

† Corresponding author. Tel.: +32 10 47 41 05, Fax: +32 10 47 20 89, e-mail: darm@tele.ucl.ac.be.

embed 4096 bits in a 512×512 image, which is far more than sufficient for the copyright information. This information is thus embedded redundantly, which increases the robustness.

The embedding of the bits is performed thanks to the following steps:

1. *Subdivision* of the image in 8×8 blocks;
2. *Classification* of the pixels of one block in *zones* of homogeneous luminance values;
3. *Subdivision* of each zone in *categories* defined by a specific grid;
4. *Embedding* of the bit through the relationship between categories mean values in each zone, following a given embedding rule.

2.2. *Classification in zones*

The aim is to classify the pixels of the block in groups of relatively homogeneous luminance. This classification takes the block features into account, which is interesting for invisibility and robustness. This will be discussed below.

Three types of contrasts can be distinguished:

1. *Hard contrast* (Fig. 1a) where two zones separated by a luminance step can be defined.
2. *Progressive contrast* (Fig. 1b) where two uniform zones are separated by a progressive luminance variation.
3. *Noise contrast* (Fig. 1c) with a luminance distributed like random noise.

The luminances of the block pixels are represented by an increasing function $F(x)$ where $F(0)$ is the lowest luminance value and $F(n^2)$ is the highest luminance value, n being the block size. The slope of $F(x)$, $S(x)$, determines the block contrast type. Let $S_{\max}$ be the maximum slope of F, located at the index $x = \alpha$. If $S_{\max}$ is lower than a given threshold T_1, the block has a noise contrast. If $S_{\max}$ is higher than T_1 the block has a progressive or hard contrast. In this case, let $\beta+$ and $\beta-$ be the indexes, closest to a, respectively higher and lower, and respecting the relationships:

$$S(\beta_+) - S(\alpha) > T2 \text{ and } S(\alpha) - S(\beta_-) > T2 \qquad (1)$$

where T_2 is another given threshold value.

If the contrast is hard, $\alpha = \beta+$ and $\alpha = \beta-$. If the contrast is progressive, the interval $[\beta-, \beta+]$ is the progressive contrast transition zone

The classification of the pixels $p(i,j)$ in two zones is defined by the following rules:

- for *progressive* and *hard contrasts*:

  if $p(i,j) < F(\beta-)$, $p(i,j)$ belongs to zone 1.
  if $p(i,j) > F(\beta+)$, $p(i,j)$ belongs to zone 2.
  Note that it is possible some pixels do not belong to any of the two zones if their luminance is too close to $F(\alpha)$.

- for *noise contrasts*, the pixels are separated in two groups having the same dimension:
- if $p(i,j) < F(n^2/2)$, $p(i,j)$ belongs to zone 1.
- if $p(i,j) > F(n^2/2)$, $p(i,j)$ belongs to zone 2.

2.3. *Subdivision of the zones in categories*

After having computed the zones, it could be envisaged to embed the bit by modifying some characteristics of the zones. Unfortunately, it appears that direct work on the zones is neither robust nor acceptable for the invisibility.

In order to find a room for embedding the bit, two categories (A and B) are created in each zone. The subdivision is determined by a grid defined before the embedding. Figure 2 shows examples of typical grids. Other subdivisions are used such as 2×1 or 1×2 grids, or more complex combinations. In the current implementation there are a dozen of

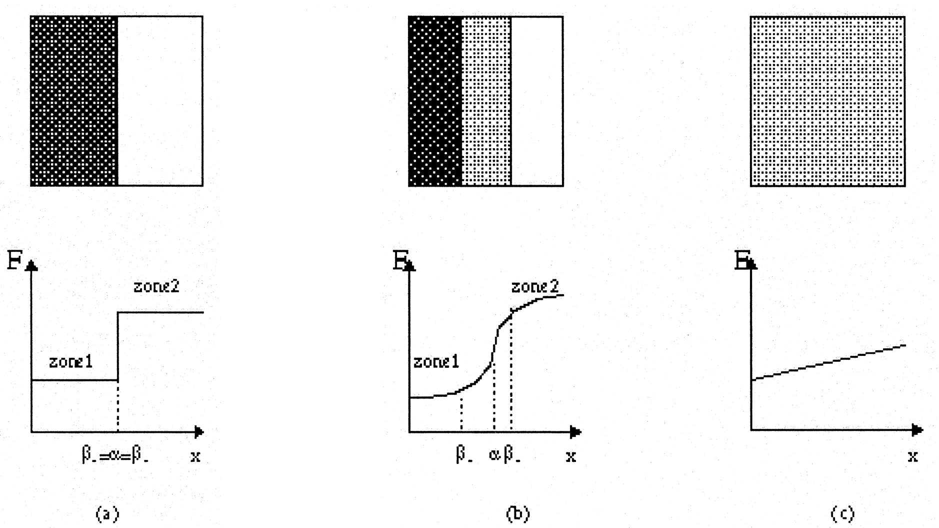

Fig. 1. Classification in zones: (a) hard contrast; (b) progressive contrast; (c) noise contrast

B	B	A	A	B	B	A	A
B	B	A	A	B	B	A	A
A	A	B	B	A	A	B	B
A	A	B	B	A	A	B	B
B	B	A	A	B	B	A	A
B	B	A	A	B	B	A	A
A	A	B	B	A	A	B	B
A	A	B	B	A	A	B	B

(a)

B	B	B	B	A	A	A	A
B	B	B	B	A	A	A	A
B	B	B	B	A	A	A	A
B	B	B	B	A	A	A	A
A	A	A	A	B	B	B	B
A	A	A	A	B	B	B	B
A	A	A	A	B	B	B	B
A	A	A	A	B	B	B	B

(b)

Fig. 2. Categories grids for one block: (a) grid 2×2; (b) grid 4×4

grids. The category of a pixel depends on two elements: the pixel spatial position in the block and the zone it belongs to.

It is also important to note that this grid has to remain **secret**. It is indeed easier to remove the watermark with the knowledge of this grid. In the current implementation, the grid changes for every block according to a secret key.

2.4. *Embedding rule*

From the first two steps, 4 different groups of pixels in the blocks are defined, depending on the *zone* (1 or 2) and the *category* (A or B). Six values can be computed from this subdivision: the 4 means m1A m1B, m2A and m2B of groups contain n1A, n1B, n2A, and n2B pixels, respectively; and the zones mean values m1 and m2. The means from the same zones are combined together. The bit is thus embedded twice through both zones. This increases the robustness and permits the bit to be embedded without altering the block excessively.

The embedding of a bit b in the block is performed through the relationship between categories luminance mean values. The embedding rule is given by:

$$\text{if } b = 0\text{: } m_{1B}^{*} - m_{1A}^{*} = l \tag{2}$$

$$m_{2B}^{*} - m_{2A}^{*} = l \tag{3}$$

$$\text{if } b = 1\text{: } m_{1A}^{*} - m_{1B}^{*} = l \tag{2'}$$

$$m_{2A}^{*} - m_{2B}^{*} = l \tag{3'}$$

where m_{1A}^{*}, m_{1B}^{*}, m_{2A}^{*} and m_{2B}^{*} are the mean values required by b and l is the *embedding level*, i.e. the required difference between the mean values.

In order to have an embedding as invisible as possible, the low frequencies (to which the eye is the most sensitive) must be conserved. The conservation of the mean value in each zone imposes:

$$\frac{n_{1A}.m_{1A}^{*} + n_{1B}.m_{1B}^{*}}{n_{1A} + n_{1B}} = m_1 \tag{4}$$

$$\frac{n_{2A}.m_{2A}^{*} + n_{2B}.m_{2B}^{*}}{n_{2A} + n_{2B}} = m_2 \tag{5}$$

Equations (1)–(3) to Equation (4) allow us to compute the values of m1A*, m1B*, m2A* and m2B*. In each zone the pixels' luminance values must be adapted to respect these means. We choose to impose the same luminance variation to all the pixels belonging to the same zone.

Let Δ_{1A}, ç$_{1B}$, Δ_{2A} and Δ_{2B} be the luminance variations. We have:

$$\Delta_{xy} = m_{xy}^{*} - m_{xy} \tag{6}$$

with x = 1,2 and y = A,B.

2.5. *Example*

We want to embed the bit b = 1 in the 8×8 block represented in Fig. 3a. With $T_1 = 5$ and $T_2 = 3$, the corresponding zones are in Fig. 3b. Dashes correspond to unclassified pixels, they will not be changed. The contrast is progressive. The pixel categories corresponding to a 2×2 grid size and to the zones in Fig. 3b are represented in Fig. 3c. Note that the grid is inverted in zone2.

The means computing gives: $m_{1A} = 66$, $m_{1B} = 59$, $m_{2A} = 148$ and $m_{2B} = 141$.

With an embedding level l = 4, the required means are: $m_{1A}^{*} = 60$, $m_{1B}^{*} = 64$, $m_{2A}^{*} = 142$ and $m_{2B}^{*} = 146$.

The required luminance variations are: $D_{1A} = -6$, $D_{1B} = 5$, $D_{2A} = -6$ and $D_{2B} = 5$.

The embedded block is shown in Fig. 3d.

3. INFORMATION EXTRACTION METHOD

Extraction of the information embedded in the image requires knowledge of the size of the block(s) and the grid used to determine the categories. Decoding is performed with the following steps:

1. *Subdivision* of the image in 8×8 blocks
2. *Classification* of the pixels of the block in zones.
3. *Subdivision* of each zone in categories.
4. *Comparison* of the means in order to determine the embedded bit value.

Steps 1 and 2 are the same as for the embedding algorithm. Step 3 is detailed below.

5	5	31	103	142	134	99	75
7	13	65	127	156	87	70	90
11	44	109	142	132	67	67	119
25	83	124	151	81	63	98	154
55	124	136	109	70	75	124	183
88	131	138	72	63	93	161	191
123	138	97	64	77	137	188	177
129	119	68	58	118	185	193	166

(a)

1	1	1	1	2	2	1	1
1	1	1	2	2	1	1	1
1	1	1	2	2	1	1	-
1	1	2	2	1	1	1	-
1	2	2	2	-	-	2	2
1	2	2	-	-	2	2	2
2	2	1	1	1	2	2	2
2	2	1	1	2	2	2	2

(b)

B	B	A	A	A	A	A	A
B	B	A	B	A	B	A	A
A	A	B	A	B	A	B	-
A	A	A	A	A	A	B	-
B	A	B	B	-	-	B	B
B	A	B	-	-	A	B	B
B	B	B	B	A	B	A	A
B	B	B	B	B	B	A	A

(c)

10	10	25	97	136	128	93	69
12	18	59	132	150	92	64	84
5	38	114	136	137	61	72	119
19	77	118	145	75	57	103	154
60	118	141	114	70	75	129	188
93	125	143	72	63	87	166	196
128	143	102	69	71	142	182	171
134	124	73	63	123	190	187	160

(d)

Fig. 3. Example of embedding: (a) original block; (b) zones; (c) categories; (d) marked block

Be Σ_1 and Σ_2 the values obtained by the comparison of the categories mean values:

$$\Sigma_1 = m_{1A} - m_{1B} \tag{7}$$

$$\Sigma_2 = m_{2A} - m_{2B} \tag{8}$$

The signs Σ_1 and Σ_2 allow decoding of the embedded bit. Moreover, the absolute values of Σ_1 and Σ_2 give a *confidence level* to this determination. Three cases can happen:

Case 1: $\Sigma_1 \cdot \Sigma_2 > 0$

- the embedded bit is 1 if $\Sigma_1 > 0$ and 0 if $\Sigma_1 < 0$.
- the confidence level is:

  very certain if $|\Sigma_S 1|$ and $|\Sigma_2| > 2$;
  very certain if $|\Sigma_1|$ and/or $|\Sigma_2| > 2.5$;
  uncertain if $|\Sigma_1|$ and $|\Sigma_2| < 0.7$;
  certain otherwise.

Case 2: $\Sigma_1 \cdot \Sigma_2 < 0$

- Be $\Sigma = (n_{1A} + n_{1B}) \cdot \Sigma_1 + (n_{2A} + n_{2B}) \cdot \Sigma_2$
- the embedded bit is 1 if $\Sigma > 0$ and 0 if $\Sigma < 0$.
- the confidence level is *uncertain*.

Case 3: $\Sigma_1 \cdot \Sigma_2$»0

- Be $\Sigma = \max(|\Sigma_1|, |\Sigma_2|)$.
- the embedded bit is 1 if $\Sigma > 0$ and 0 if $\Sigma < 0$ and the confidence level is *uncertain*. If $\Sigma = 0$ the bit is *undetermined.*

As explained in Section 2.1, each bit is embedded redundantly. The global confidence level of the bit is the integrated value of each occurrence of this bit in the image. The bit is determined afterwards, according to the global confidence level.

4. IMPLEMENTATION AND RESULTS

A few parameters are required to embed a binary code into a picture: the embedding level l, the categories grid size, the number of bits to be embedded, and the level of redundancy: if the code is short enough compared to the number of blocks, a repetition of the code with error correction codes (e.g. BCH codes) can be used. The correcting capacity of this code is also a parameter, this will be discussed in Section 4.2.

Two aspects determine the quality of the embedding: the **invisibility** of the code and its **robustness** to image processing and lossy compression.

The visibility increases with the **embedding level**. Typical values are $0 < l < 7$. If l is higher than 7 the code is really visible, but the higher l is, the better the robustness is. Figure 4 illustrates the influence of the level on the robustness. The Bit Error Rate (BER), percentage of not recovered bits, is drawn with regards to the JPEG quality factor. The choice of l depends on the expected compression level. In TALISMAN, an invisibility criterion has been added. This preliminary test determines if a block can be watermarked or not. This simple test is based on analysis of the variance of the block and will be presented in Section 4.1.

Visibility depends slightly on the **categories grid size**: with a 1×1 grid the transitions between categories are frequent and the embedded code is similar to high frequency noise, it is thus less visible than with a 4×4 grid since high frequency noise is less perceptible by the human eye. At the opposite, a bigger grid resists better to compression, the trade-off between invisibility and robustness is a 2×2 grid. Other patterns are also used such as 1×2 or 2×1 grids. The grid size is determined by a secret key for each block.

The use of **error correcting codes** allows to deeply reduce the decoding error rate without affecting the visibility. This will be presented in Section 4.2.1.

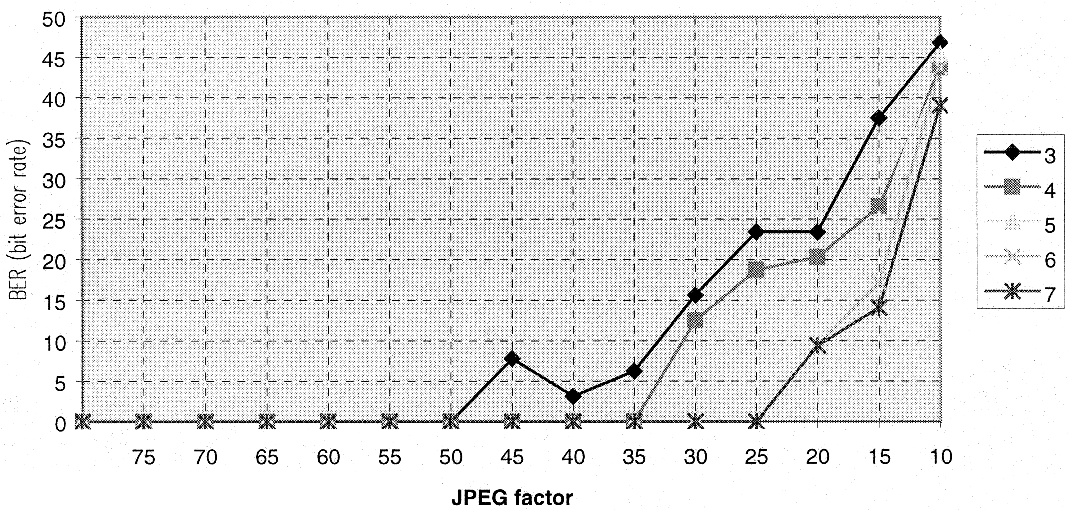

Fig. 4. BER for different embedding levels tested on Lena

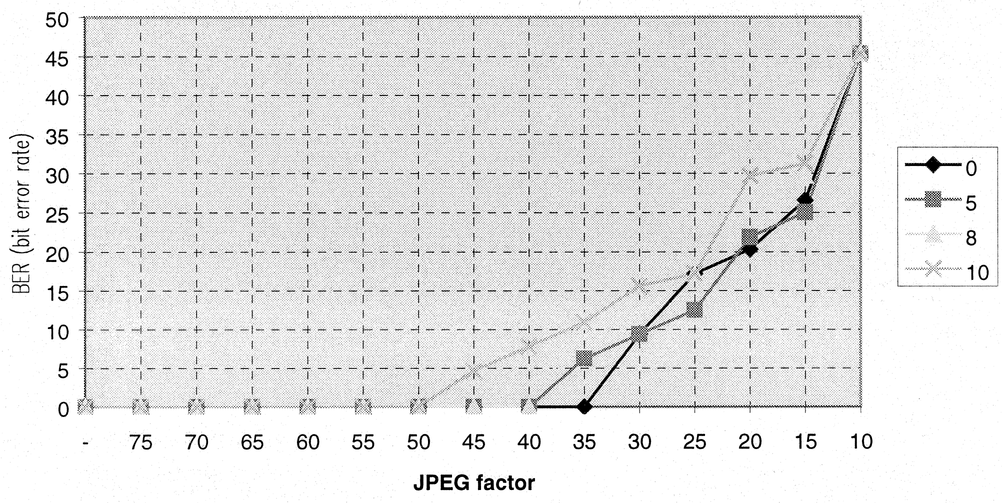

Fig. 5. Influence of the variance (level = 4)

4.1. *Invisibility improvements*

In order to be compliant with the strict invisibility criterion, an invisibility criterion has been added before the embedding. This criterion discards low activity blocks for which it is clear that the watermark will be visible.

- The variance of each block is calculated before the embedding. If this *variance* is inferior to a given *threshold* the block is left unchanged.
- Another criterion has been added inside the block. If the *variance of one zone* in a block is equal to zero (flat zone), the block is also left unchanged.

The quality of the embedding is highly dependent on the **variance threshold.** Fig. 5 shows the robustness towards JPEG compression for different variance thresholds. When this threshold is above 10 for 8-bit pictures (256 grey levels), the watermark is always invisible for reasonable levels, but the robustness decreases a lot because too many blocks are left unchanged. Figure 6 shows the influence of typical choices of parameters on Lena. In the current implementation, the choice that has been made is 8 for the variance threshold and 4 for the corresponding embedding level. Figure 7b represents the picture of Lena watermarked with these parameters. Figure 7d is represents the blocks that are discarded by the criterion for the same parameters. Figure 7a is the original picture of Lena and Fig. 7c is the result of the embedding at a too high level without the variance criterion (l = 20).

4.2. *Robustness improvements*

4.2.1. *BCH codes.* Even with the variance criterion, there is a huge number of embedded blocks. It is thus interesting to embed redundantly the binary copyright information, the length of which is typically 64 or 128 bits. These lengths correspond to future promising identification ISO standards, such as ISAN number [14] for audiovisual material and IMLP and SPIFF for still images [15].

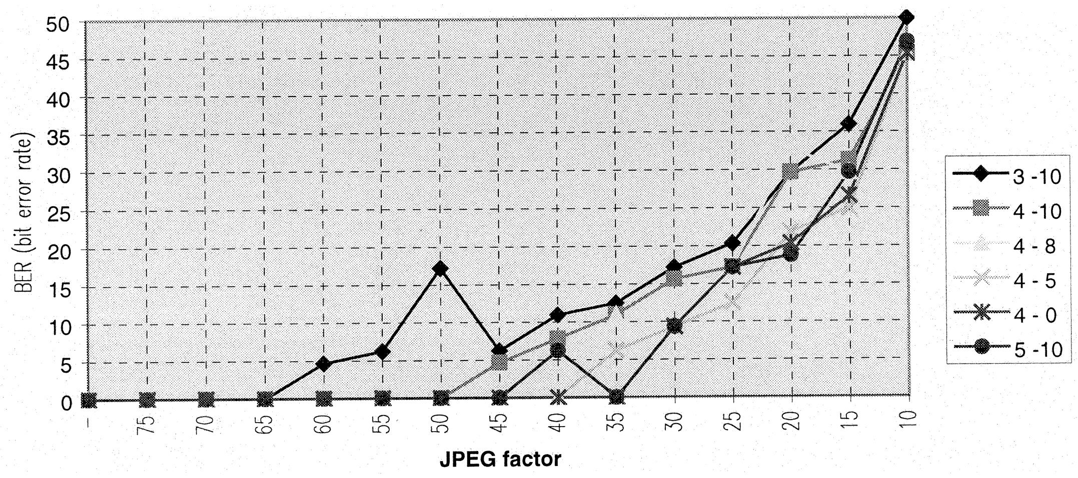

Fig. 6. Results for typical embedding levels and threshold values tested on Lena

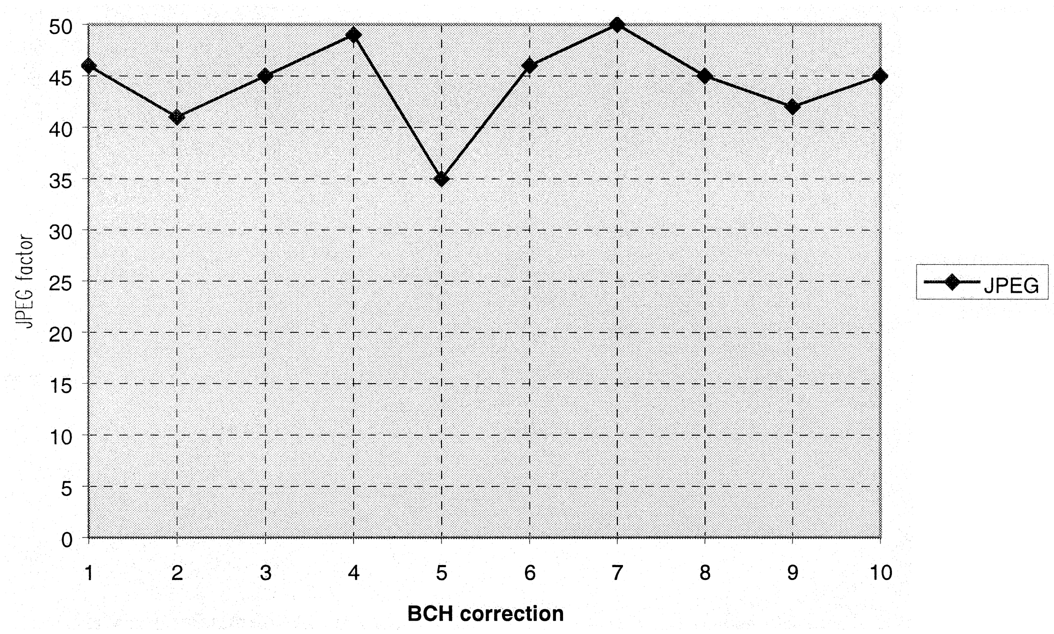

Fig. 7. (a) Lena original; (b) watermarked; (c) level = 20; (d) modified blocks

Nevertheless, a more clever use of this redundancy is to make use of error correcting codes, e.g., BCH codes [16]. BCH codes have adaptable correction capacities. The binary code is first encoded by the BCH, then the resulting bitstream is embedded redundantly in the picture. During the retrieval, the embedded bitstream is then decoded through the BCH and the original message can be recovered.

Figure 8 shows that the use of these codes is profitable. When the error correction capacity grows, the robustness increases, this is due to the correction capacity of the BCH. Nevertheless, when the correction capacity grows, the redundancy decreases. As a consequence, there is an optimum. In the current implementation, the choice of an error correction capacity of 5 has been made.

4.2.2. *Temporal redundancy*. In the case of video sequences, the same copyright information can be embedded in consecutive frames. Sometimes, the same code is embedded in the whole video. It is thus interesting to take this into account and to use the temporal redundancy for the retrieval. Tests have been performed on 720×576 color pictures compressed at 6 Mb/s with MPEG2. They have proved that after 1 or 2 sec the BER is approximately equal to zero if the embedded code is randomly distributed all over the picture.

4.3. *JPEG compression*

The robustness towards JPEG and MPEG compression is an essential feature for every watermarking algorithm. Figure 8 represents the Bit Error

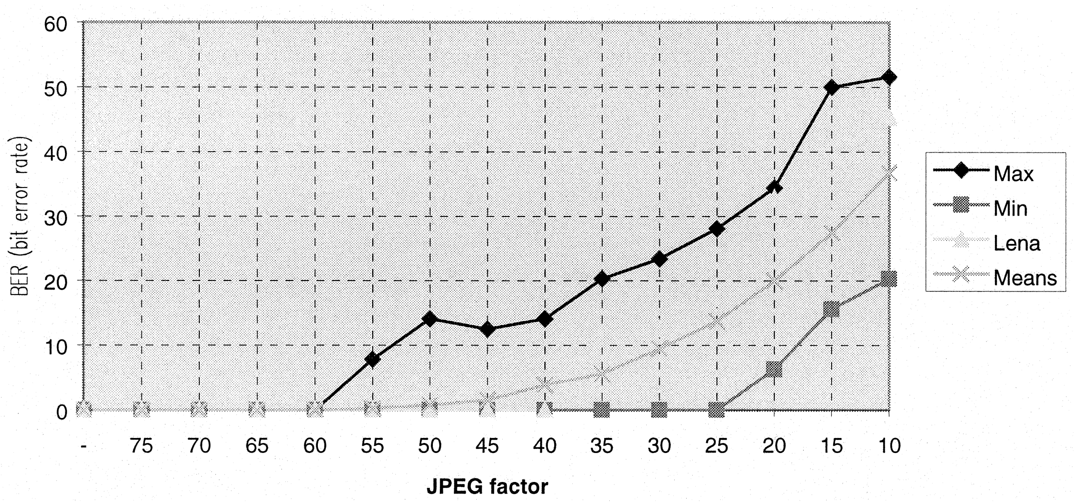

Fig. 8. Influence of the BCH

Fig. 9. Retrieval results for the database

Rate after retrieval with regard to the JPEG quality factor. A quality factor of 75% does not introduce artefacts. These results have been achieved on a database of 50 pictures representing a wide range of natural pictures (not synthetic ones). Four curves are drawn on Fig. 9: the maximum BER, the minimum, the average and the results for Lena.

4.4. *Complexity issues*

One of the most important features of the algorithm is that its complexity is very low. Very simple operations are performed on the image, while neither frequency transforms nor spatial filtering are used. The costliest operation is the sorting of the pixels for each block. Other costly operations for the block are the computing of variance and embedding, but their complexity is inferior to that of sorting. Moreover, since the operations are performed independently in each block, it is possible to achieve them in parallel.

5. CONCLUSION

In this paper a new approach for embedding a code into a picture has been presented. The technique is based on the image spatial decomposition in blocks and the classification of the pixels in homogeneous luminance zones. The code is embedded in the order relationship between mean values inside the zones. The influence of the different embedding parameters on the visibility and the resistance of the code has been studied. In particular, the resistance to JPEG compression is being studied in the TALISMAN project. An invisibility criterion was added that indicates when a block does not contain enough information to be watermarked. Tuning of the parameters for this criterion is currently studied.

REFERENCES

1. Kahin, B., The Strategic Environmant for Protecting Multimedia. *IMA Intellectual Property Proceedings*, Vol. 1, pp. 1–8, January 1994.
2. Matsui, K. and Tanaka, K., Video-Steganography: How to Embed a Signature in a Picture. *IMA Intellectual Property Proceedings*, 1994, **1**(1), 187–205.
3. Brassil, J. T., Maxemchuk, N. F., Low, S. and O'Gornam, L., Electronic Marking and Identification techniques to Discourage Document Copying. *Proceeding of IEEE INFOCOM '94*, Toronto, pp. 1278–1287, June 1994.
4. Bender, W., Gruhl, D. and Moromoto, N., Techniques for Data Hiding. *Proceedings of the SPIE*, San Jose, **2420**(40), February 1994.
5. Caronni, G., Assuring Ownership Rights for Digital Images. *Proceedings of reliable IT systems*, VIS 95 Germany, June 1995.
6. Delp, E. J. and Wolgang, R. F., A watermark for Digital Images. *Proceedings of the IEEE International Conference on Image Processing*, Lauzanne, pp. 219–222, September 1996.
7. Kutter, M., Jordan, F. and Bossen, F., Digital signature of color images using amplitude modulation. *Proceedings of the SPIE/IS and T conference on Storage and Retrieval for Image and Video Databases V*, San Jose, pp. 518–526, February 1997.
8. Cox, I. J., Kilian, J., Leighton, T. and Shamoon, T., Spread Spectrum Watermarking for Multimedia. *Proceedings of the SPIE, San Jose*, 1995, **2420,** 456–459.
9. Koch, E. and Zhao, J., Towards Robust and Hidden Image Copyright Labeling. *Proceedings of IEEE Workshop on non-linear Processing*, Thessaloniki, pp. 452–455, June 1995.
10. O'Rinuaidh, J. K., Dowling, W. J. and Boland, F. M., Phase Watermarking of images. *Proceedings of the IEEE International Conference on Image Processing*, Lauzanne, pp. 239–242, September 1996.
11. Swanson, M. D., Bin Zhu and Tewfik, A. H., Transparent Robust Image Watermarking. *Proceedings of the IEEE International Conference on Image Processing*, Lauzanne, pp. 211–214, September 1996.
12. Delaigle, J. F., De Vleeschouwer, C. and Macq, B., Digital Watermarking. *Proceedings of the SPIE/IS and T conference 2659 on Optical Security and Counterfeit Deterrence Techniques*, San Jose, pp. 99–110, February 1996.
13. TALISMAN Project ACTS 019: URL: http://www.tele.ucl.ac.be/TALISMAN/index.html.
14. ISAN (International Standard Audiovisual Number) project number 15706—ISAN Agency in Geneva.
15. ISO SC29/WG11 the Joint Photographic expert Group—(SPIFF, IMLP).
16. Puser, M., Introduction to Error Correcting Codes. Artech House, Boston-London, 1995, ISBN 0-89006-784-8.