DCT-Based Watermarking Method Using Color Components

Gregory Lo-Varco, William Puech, Michel Dumas

DCT-Based Watermarking Method Using Color Components

G. Lo-varco[1], *W. Puech*[2] *and M. Dumas*[1]
[1] *Laboratoire CEM2, UMR CNRS 5507, Université Montpellier II*
Pl. Gabriel Péri, 30021 Nîmes Cedex 1, France
[2] *Laboratoire LIRMM, UMR CNRS 5506, Université Montpellier II*
161, rue Ada, 34392 Montpellier Cedex 5, France
lovarco@cem2.univ-montp2.fr, puech@lirmm.fr, dumas@univ-montp2.fr

Abstract

In this article, we present an original DCT-based watermarking method on color images and integrating an error-correcting code (ECC) by generating polynom. This technique has been developed to be robust versus compression.

1. Introduction

With the growth of networked multimedia systems, the need of secure communication and data transfer go increasingly. Applications are numerous and various like satellite or medical imagery, or remote monitoring. During images transfer, data integrity is not really secure [4]. Watermarking can be an answer to such problems. For applications dealing with images, the watermarking objective is to embed an invisible message inside the image data. The length of the transmitted message can be relatively important, in fact, longer than the one necessary for identification. Insertion can be made in a different way according to the length of the message or desired robustness. In this paper, we present a DCT-based watermarking method [3, 1, 7], which use the color information [8, 2] and an addition of error correcting code (ECC).
In the section 2, we present the DCT-based watermarking method. Section 3, firstly, we present the message coding and encapsulating, then we describe the adaptation of watermarking method to JPEG algorithm. In the section 4, we apply the method to remote monitoring and we analyze the robustness to compression.

2. DCT-based watermarking method

An application of watermarking consists in inserting a long message in an image for transmission via the network [5]. We describe here a method of watermarking in the frequential field which must resist to compression JPEG. To increase the robustness, we propose a technique of frequential watermarking on the continuous component of DCT because it is used in many standardized applications such as JPEG and MPEG compression.

The proposed watermarking method is based on substitution, in a statistical criterion, of the first, second or third Least Significant Bits (LSB1, LSB2 or LSB3) of pixels blocks [8]. The choice of the LSB1, LSB2 or LSB3 depends on the quality factor of the JPEG compression. Blocks of N pixels, with $N = n^2$ and n the block side, allow to insert the information several times in the image. For example, in an image of 500×400 pixels, with $N = 64$ and a message size of 100 bits, the information will be inserted 31 times.
Thus the objective is to embed a message M made up of m bits b_k, $k \in [1..., m]$. From a block made up of N pixels $p_k(j)$ of the image, we calculate the DCT continuous component $F_k(0,0)$ of this block:

$$F_k(0,0) = \frac{1}{n}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} p_k(i,j) = \frac{1}{n}\sum_{j=0}^{N-1} p_k(j). \quad (1)$$

where $p_k(j)$ is the gray level of pixel i in the block k.
Then, after quantization, we have:

$$F'_k(0,0) = F_k(0,0)/q(0,0). \quad (2)$$

This real value $F'_k(0,0)$ is used in order to make our method more robust to quantization, then to compression. While calculating $R_{F'_k(0,0)}$, the division real rest of $F'_k(0,0)$ from 2, we have: $0 \leq R_{F'_k(0,0)} < 2.0$. The objective is to bring back the value $F'_k(0,0)$ to a stable value $F''_k(0,0)$ according to the value of the bit b_k to be inserted. This is made in order to resist to the maximum of variations due to the deformations of another coding. Let us calculate, d, the difference between $R_{F'_k(0,0)}$ and $b_k + 0.5$:

$$d = b_k + 0.5 - R_{F'_k(0,0)}. \quad (3)$$

The method of watermarking by block on the LSB of $F'_k(0,0)$ real value modify only N_d pixels of the block according to the following equation:

$$p'_k(j) = p_k(j) + sign(d), \quad (4)$$

with $N_{d_k} = [|d| \times N]$, the number of modified pixels of the block k made up of N pixels.

The bit b_k thus corresponds to LSB integer part of ${F''}_k(0,0)$ such as ${F''}_k(0,0)\%2 = b_k + 0.5$. We have then:

$$ {F''}_k(0,0) = \frac{1}{n}\Big(\sum_{j=0}^{N_{d_k}-1} p'_k(j) + \sum_{j=N_{d_k}}^{N-1} p_k(j) \Big) / q(0,0). \tag{5} $$

3. Color watermarking and ECC

3.1. Message coding and encapsulating

The suggested watermarking system comprises several stages which optimize the length of the message in order to increase the redundancy [6]. For remote monitoring application, the embedding message requires 18 characters. The camera number ID and its location, hour and date are coded on 72 bits by using a Huffman code. Bits are then gathered by blocks of 12. The ECC by generating polynom, Frame Check Control, is then added. It allows to detect the presence of errors. The complete message is then coded and encapsulated by blocks of 96 bits. The details of 96 bits obtention are described in Figure 1.

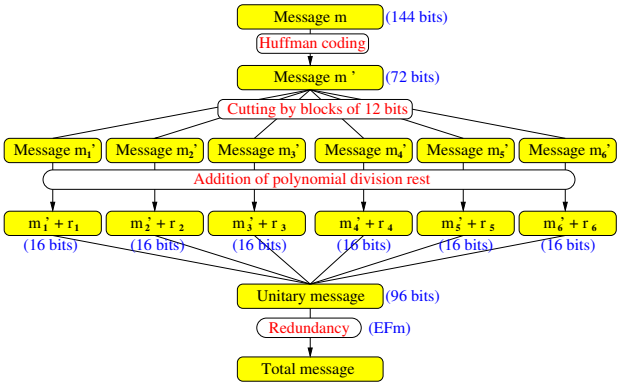


Figure 1: *Message coding.*

3.2. Adaptation of watermarking method to the JPEG algorithm

The color images are generally built with three components corresponding to three grey level images. The quantity of informations and the color image size are very important (approximately three times larger). Therefore for transfering quickly color images, a compression algorithm is usually applied. In this paper, we describe a watermarking method which is adapted to JPEG compression. So the different stages of watermarking method must be as near as possible to the stages of JPEG algorithm.
Firstly, the color image is decomposed in three components Red (R), Green (G) and Blue (B). This decomposition is illustrated Figures 2.b, c and d. Then, with a matrix of components changing, we obtain three others planes

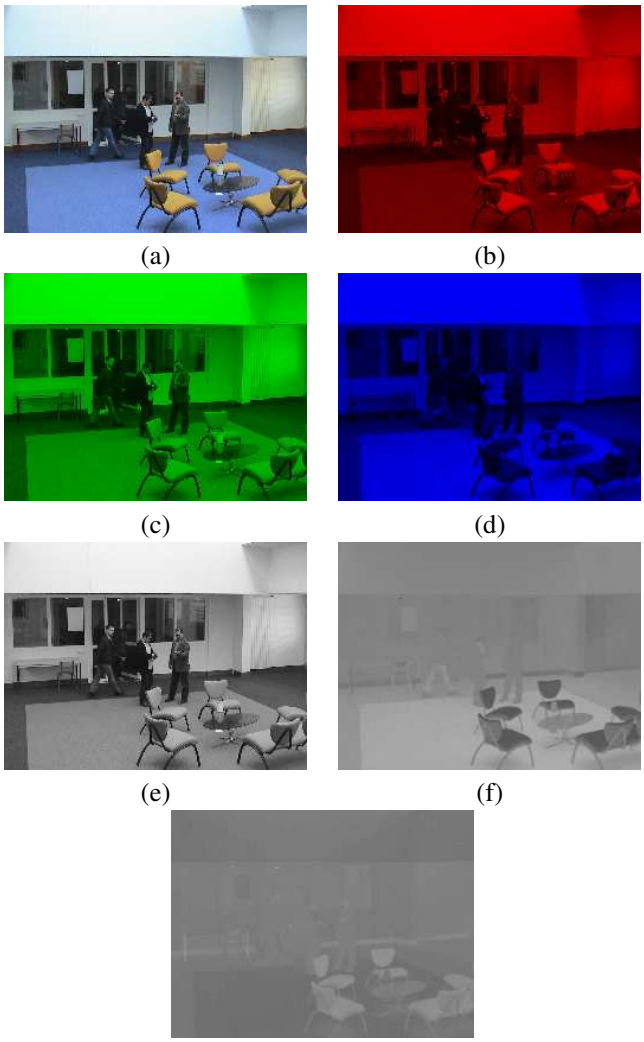


Figure 2: *a) Original image, b) Red component, c) Green component, d) Blue component, e) Y component, f) u component, g) v component.*

called Y, u and v, illustrated Figures 2.e, f and g. The Y component corresponds to the luminance information. In each plane, informations are embedded with the watermarking method, and we obtain three new planes Y', u' and v' including the message and the generating polynom. On the basis of these planes, we built three new watermarked components R', G' and B'. Finally, starting from these components, we recomposed the color watermarked image. All these watermarking method steps are described in the diagram illustrated Figure 3.

3.3. Message watermarking on Y component

The Y component contains the majority of color image informations. So it is the most sensitive component to the modifications. Therefore, the JPEG algorithm uses small

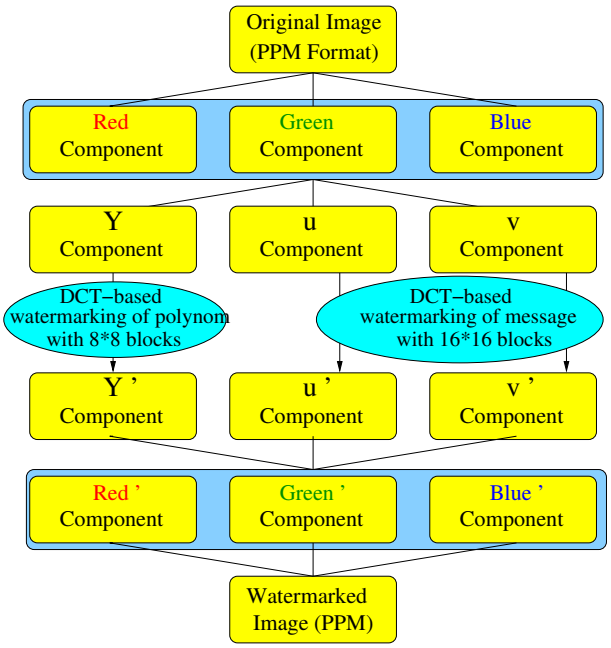


Figure 3: *Diagram of the watermarking method.*

8×8 blocks to compress the image. We use also this block size to embed the message. The message is coded with 96 bits. The obtention of 96 bits is described in Figure 1. These 96 bits are embedded as many times as possible according to the image size. With our watermarking method, one bit is embedded by pixel blocks and we can define the block size. To be adapted to JPEG compression, we choose the same size of the blocks used in this compression algorithm i.e. a 8×8 block. In the next section 3.4, we will see that concerning the u and v components the block size will be different. After the definition of blocks, we can determinate an embedding factor EFm which gives an idea of message redundancy according to the image size. This is one possible definition of this embedding factor:

$$EFm = \frac{\frac{Image\ size}{64}}{96} = \frac{Image\ size}{6144}. \qquad (6)$$

For example, in the original image (1024×768 pixels), illustrated Figure 2.a, the message is inserted 128 times.

3.4. Generating polynom watermarking on u and v components

The u and v components are used to insert the generating polynom, necessary to the ECC. The generating polynom is coded with only five bits. These five bits are embedded as many times as possible according to the image size. As for the Y component, one bit is embedded by pixel blocks but we use an other block size. To be adapted to JPEG compression, we choose 256 pixels as block size. Indeed, in this compression algorithm, the DCT coefficient is calculated on 8×8 blocks of oversampled u and v components i.e. the choice of 16×16 blocks. Since the new block size is defined, we obtain an embedding factor EFp which gives the polynom redundancy according to the image size:

$$EFp = \frac{\frac{Image\ size}{256}}{5} = \frac{Image\ size}{1280}. \qquad (7)$$

Because of polynom lenght, shorter than the message (approximately 15 times), the embedding factor EFp is much bigger. For the same image (1024×768 pixels), illustrated Figure 2.a, the generating polynom is embedded 614 times in each component.

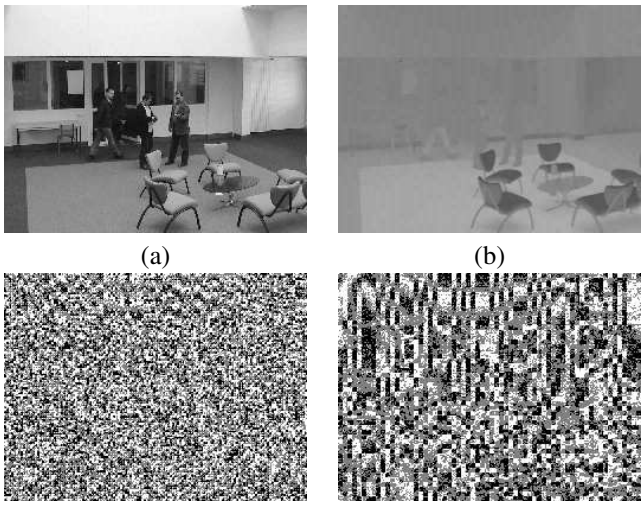
3.5. Extraction of message and uses of ECC

After having shown the watermarking method, we are interested in the extraction of the message and more specialy in using of color information. First we can describe the obtention of watermarked polynom: The u and v component are divided by sets of five neighbour blocks. Every set gives a polynom. So, at the end of image reading, EFp polynoms are obtained. Then, with a majority vote, the generating polynom is extracted.
Concerning the message, the method of extraction is more complex, because the block size and the set size changes. We get sets of sixteen neighbour blocks and we extract sixteen bits. The first twelve bits code the message and the four other bits are used for the ECC.
Now, we can describe the utility of ECC. The ECC allows us to detect errors due to transfer and compression. Its application is very easy: we get again the sets of sixteen neighbour blocks. These sixteen bits give a number. This number is also divided by the extracted polynom and we obtain a rest. If the rest is different of zero, an error is detected else the next sets of blocks is read. Then we can check, with the polynom, if this part of message is right or not. Finally, to read an unitary message, six sets of sixteen blocks are necessary. We remind that initially the unitary message lenght was 18 characters.

4. Results

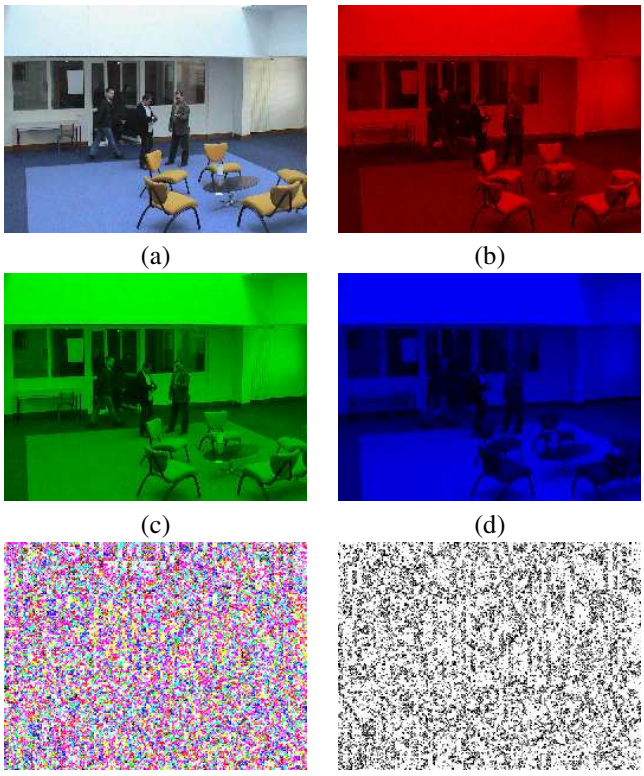
4.1. Remote monitoring applications

Figure 4.a presents the Y component which is watermarked by using 8×8 blocks. Figure 4.c shows the difference between the original Y component and the watermarked Y component. Concerning the other components, u and v, the block size is different. Figure 4.b presents the u watermarked plane with a block size of 16×16. We can see this size difference, Figure 4.d, where we show the difference between the original u component and the u watermarked component.
By reverse transformation of Y, u and v watermarked planes, we obtain three new planes: the watermarked red plane Figure 5.b, the watermarked green plane Figure 5.c and the watermarked blue plane Figure 5.d. The image Figure 5.f



(a) (b) (c) (d)

Figure 4: *a) Y watermarked component, b) u watermarked component, c) Difference between Y and Y watermarked component, d) Difference between u and u watermarked component.*



(a) (b) (c) (d) (e) (f)

Figure 5: *a) Watermarked image, b) Red watermarked component, c) Green watermarked component, d) Blue watermarked component, e) Difference between original and watermarked image, f) Difference between red and red watermarked component.*

shows the difference between the original red component and the watermarked red component. Finally we recombine these three planes to obtain the final watermarked image, Figure 5.a. To have an idea of watermarking impact on the image, Figure 5.e presents the difference between the original image and the watermarked image.

4.2. Analyze of compression robustness

	Generating polynom				
Quality Factor	bit 1 (right)	bit 2 (right)	bit 3 (right)	bit 4 (right)	bit 5 (right)
without	97	100	97	100	97
100%	97	100	97	100	97
90%	97	100	97	100	97
80%	97	100	97	100	97
70%	96	95	96	96	97
60%	91	83	94	85	91
50%	75	86	67	83	71

Table 1: **Extraction of generating polynom on the u component according to the quality factor of compression.**

During transfer, the watermarked image undergo compression. Figure 6.a illustrates a JPEG compression with a quality factor of 60%. Indeed, the difference between the original Y component Figure 2.f. and the compressed Y component, Figure 6.b, is illustrated Figure 6.e. Concerning the watermarking method, the modifications must remain undetectable for the Human Visual System (HVS). That explains the noise which is present in the whole image.
Because of this nature, compression modify the whole image but not the whole watermark. However, watermarked message is redundant, same information are embedded a great number of times. Consequently, only some unit messages are modified by compression and so the information can be recovered.
To evaluate the robustness of our method, we are interested in the amount of bits and characters which are different of whose embedded, according to the quality factor of compression. Table 1 illustrates the polynom extraction on the u component of the image, with a polynom of 5 bits repeated 614 times, that means 3072 bits, with QF of compression ranging between 100% and 50%. As the image is compressed, some bits are false but we can note that, for all five polynom bits, the percentage of right detected bit is higher than 67%. Then, by a bit to bit vote method, we are able to restore good value for generating polynom.
From this polynom, thanks to the ECC, we detect, on Y component, a degraded message because of image compression. Indeed, on Y plane, the original message **132HAL 13:47 08/01/2004** is inserted 128 times. Table 2 shows the message extraction and more specialy the percentage of right detected full messages according to the same quality factor of compression. Thanks to ECC, we detect errors on all the messages. So Table 2 allows us to evaluate the robustness of our system versus compression. Obtained

Quality Factor	Right Message (/128) 132HAL134708012004
without	128
100%	128
90%	128
80%	125
70%	87
60%	17

Table 2: **Extraction of right full message according to the quality factor of compression.**

results validate ECC use and bit to bit vote use. Thus, conversion of the watermarked image in JPEG format by preserving the watermark is possible with our method.

5. Conclusion

In this paper, we have presented a color DCT-based watermarking method resisting to disturbances that an image can undergo during its transfer. This robustness is due to ECC addition in the message and also to the use of the three color components. Moreover, we have shown how this method can also resist to such type of compression, often used for transfer. As new research orientations, we plan on one hand to use contents image to insert the message but, on the other hand, to adapt the choice of block size to the quality factor of compression. We hope also to increase the robustness against compression.

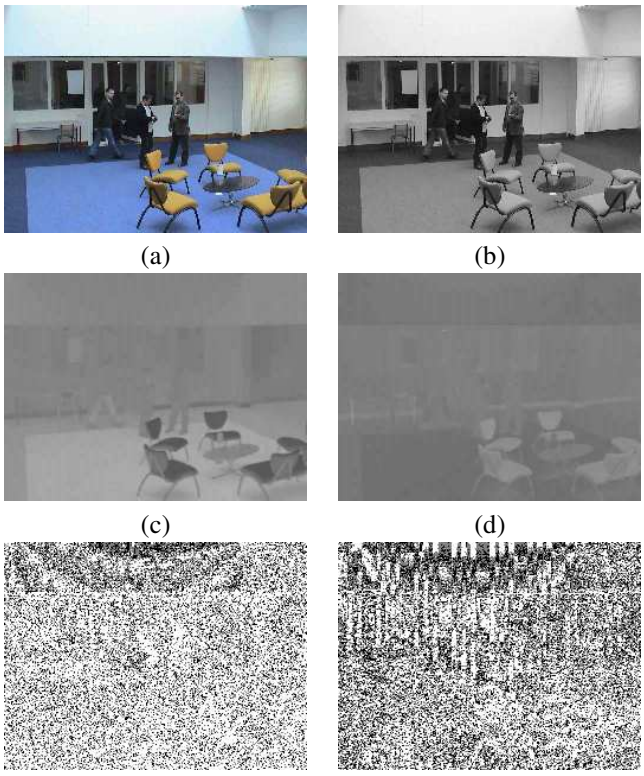


Figure 6: *a) Compressed and watermarked image with QF=60%, b) Compressed and watermarked Y component with QF=60%, c) Compressed and watermarked u component with QF=60%, d) Compressed and watermarked v component with QF=60%, e) Difference between Y and compressed and watermarked Y component with QF=60%, f) Difference between u and compressed and watermarked u component with QF=60%.*

References

[1] A. Bors and I. Pitas. Image watermarking using block site selection and DCT domain constraints. *Optics Express*, 3(12):512–522, 1997.

[2] G. Chareyron and A. Tremeau. Watermarking of color images based on a multi-layer process. In *CGIV'02, Poitiers, France*, pages 77–80, 2002.

[3] I. Cox, J. Killian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Trans. on Image Processing*, 6(12):1673–1687, 1997.

[4] F. Deguillaume, S. Voloshynovskiy, and T. Pun. Hybrid robust watermarking resistant against copy attack. In *EUSIPCO'02, Toulouse, France*, 2002.

[5] J. Delaigle, C. D. Vleeschouwer, and B. Macq. Watermarking algorithm based on a human visual model. *Special Issue on Watermarking, Signal Processing*, 66(3):319–336, 1998.

[6] A. Guyader, E. Fabre, and C. Guillemot. Robust decoding of vlc encoded markov sources. In *GRETSI'01, Toulouse, France*, 2001.

[7] E. Koch and J. Zhao. Towards robust and hidden image copyright labeling. In *IEEE Workshop on Nonlinear Signal and Image Processing, Neos Marmaras, Greece*, pages 452–455, 1995.

[8] W. Puech, P. Montesinos, and M. Dumas. Color Image Watermarking Robust to JPEG Compression. In *Proc.* 1^{st} *European Conference on Color in Graphics, Imaging and Vision (CGIV-02), Poitiers, France*, pages 81–85, 2002.

6. Biography

G. Lo-varco is French. He was born in September 1977. He is a PhD student at the Montpellier II university (France) since October 2002. His thesis' director, **Michel Dumas**, and his co-director, **William Puech**, are the two other authors of this paper. His thesis subject concerns the watermarking and more specially the image secure transfer using watermarking.

W. Puech was born in December 1967, in France. He received the diploma of Electrical Engineering from the University of Montpellier, France, in 1991 and the Ph.D. Degree in Signal-Image-Speech from the Polytechnic National Institute of Grenoble, France in 1997. He initialised its research activities in image processing and computer vision. He served as a Visiting Research Associate at the University of Thessaloniki, Greece. From 1997 to 2000, he had been an Assistant Professor at the University of Toulon, France, with research interests include methods of active contours applied to medical images sequences. Since 2000, he is Associate Professor at the University of

Montpellier, France. He works now in the robotic departement of the LIRMM (Laboratory of Informatic, Robotic and Microelectronic of Montpellier. His current interests are in the areas of security of digital image transfer (watermarking and cryptography) and edges detection applied to medical images and road security.

M. Dumas has obtained his PHD on July 9th 1987. His thesis' title was: "Sur l'analyse des propriétés physicochimiques et électroniques de la surface InP(100). Contribution aux propriétés de l'interface métal-InP(110)". He worked about the contribution of the interface state on the noise influence in Si MOS structure(capacitor and transistor). In 1996 he has created the STINIM research team (Systèmes de Traitement de l'Information Numérique et Ingénierie Multimédia, 9 members) and heads it since that date. In 1999 he has created the "Institut Universitaire Professionalisé" on Information and Communication Tools with the industrial creation option, he is Director of this institute.