SAFE ROIS OF COLOR IMAGES BY INDUCTIVE DATA HIDING

G. Lo-Varco[1] *and W. Puech*[2]

[1] CEM2 Laboratory, UMR CNRS 5507, University of Montpellier II
Pl. Gabriel Péri, 30021 Nîmes Cedex 1, France
[2] LIRMM Laboratory, UMR CNRS 5506, University of Montpellier II
161 rue Ada, 34392 Montpellier Cedex 5, France
email: lovarco@cem2.univ-montp2.fr, puech@lirmm.fr

ABSTRACT

In this paper a region-based color image watermarking algorithm is presented. The objective of the method is to embed a particular message in each region of interest (ROI) in the image. The embedding message has to be detected after image manipulations such as cropping, rotations and color JPEG compression. By using shape information, the embedding message is synchronized with each ROI on each color component Y, Cr and Cb. To embed information non-rounded DCT coefficients are watermarked. Experimental results demonstrate the performance of the algorithm against spatial and frequential attacks.

1. INTRODUCTION

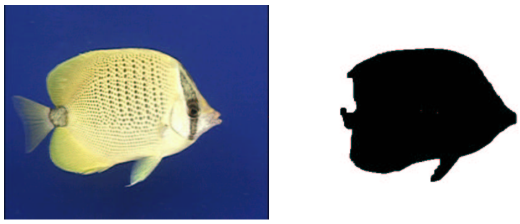
Several techniques have been proposed in the literature to embed information in digital images [1]. Watermarking applications include copyright protection, authentication, embedded and hidden information. Our color watermarking method is developed for these last applications. The length of embedded data can be relatively important, between 5 and 200 bits in each region of interest (ROI) according to their sizes. The current paper presents a technique for color watermarking of images, based on non-rounded DCT-coefficients [2]. This method belongs to a class of embedding schemes called quantization index modulation (QIM) [3]. The hidden data are synchronized with the ROIs. This method mainly protects against interaction with visual content, geometrical manipulations [4] and JPEG compression. The paper is organized as follows. In Section 2 we introduce the ROIs and the Principal Component Analysis (PCA) is presented. Section 3 describes our color DCT-based watermarking method. In Section 4, we present and analyze some concluding results.

2. ROIS EXTRACTION AND DESCRIPTORS

Section 2.1, we present the ROIs detection by segmentation. From these ROIs, a binary mask is obtained. In Section 2.2, each ROI is analyzed and defined by several descriptors. Section 2.3, we describe the synchronization between hidden data and image.

2.1 Content-mask obtention

Our objective is to use image regions, which may not completely correspond to an object, but are generated by a simple segmentation algorithm. First, the color image is converted to grey-level format. Then a region-growing algorithm is used. Each ROI consists of a subset of pixels forming a shaped area of the image. To illustrate this process, segmentation is applied on the original image "Fish", Figure 1.a, and we obtain the associated content-mask, illustrated Figure 1.b. After segmentation, the image is a set of convex areas. The ROI are reduced by a set of erosion and dilatation. We lost information about the ROI outlines but we increase the robustness. Then the region-labeling algorithm indicates all the ROIs. Each pixel obtains a label L_{ROI}, which depends on the intensity of pixel and its local neighborhood (4-connexity). After ROI extraction, an analysis is necessary to synchronize correctly the hiding data.



(a) (b)

Figure 1: a) The original image "Fish", b) The associated binary mask.

2.2 ROI features

In an attempt to detect the embedded data after geometrical modifications, we insert information in a frame of reference depending on each ROI shape. To this end, we define two features of ROI: the position indicator and the orientation and shape descriptor. To indicate the position of ROI in the image, we use the spatial centre G_{ROI}. The moments of first degree noted $_i(ROI)$ and $_j(ROI)$ precisely locate this singular point of ROI. This centroid G_{ROI} is the origin of a specific frame of reference. Then, to build this frame, we have to determinate two directions specific to the ROI. We employ the PCA in an attempt to obtain the principal directions of the ROI. To determinate them, the ROI moments of the second degree are calculated. We obtain the horizontal variance $V_i(ROI)$, the vertical variance $V_j(ROI)$, and the covariance $V_{ij}(ROI)$. This set of coefficients is represented by a 2×2 covariance matrix $C_0(ROI)$ noted:

$$C_0(ROI) = \begin{bmatrix} V_i(ROI) & V_{ij}(ROI) \\ V_{ij}(ROI) & V_j(ROI) \end{bmatrix}. \quad (1)$$

An analysis of this matrix gives two eigenvalues $_1(ROI)$ and $_2(ROI)$ and two associated eigenvectors $\vec{V}_1(ROI)$ and $\vec{V}_2(ROI)$ [5]. These vectors denote directions providing maximal variance of the ROI. In this context, these couples $\{\ ,\vec{V}\}$ represent the major and minor axes of the ROI. These two axes are illustrated in Figure 2. Associated with the centroid G_{ROI}, they form a frame of reference adapted to receive the hidden data. The hidden data are consequently oriented and synchronized according to each frame of reference.

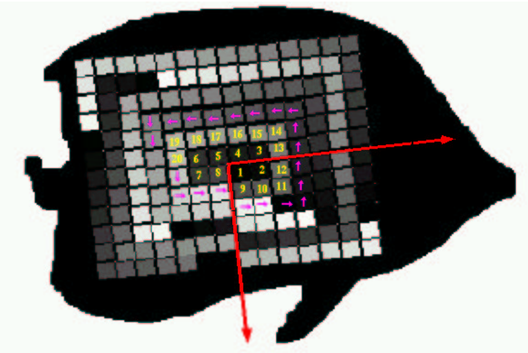


Figure 2: The appropriate frame of reference with the building order of blocks.

2.3 Synchronization of hidden data according to the ROIs

Our watermarking method implies several constraints on the unitary blocks of N pixels. Firstly, the block size is normalized in order to use a unique and fast detection process. Then, the block is generated according to the ROI shape. The eigenvectors are the two oriented sides of block. The block is also a set of N 2D variables $p_b(i,j)$ (pixels of block) with $0 \leq b < N$. The unitary block is specifically oriented and adapted to each ROI.
Then, to embed data, a building order is followed. We use one more time the couples $\{\ ,\vec{V}\}$ and the centroid G_{ROI} to determinate the insertion path, noted Ip_{ROI}. Ip_{ROI} is a tidy set of m starting pixels sp_k with $0 \leq k < m$ and m the number of embedded bits in the ROI. Each point sp_k is the origin of a unitary block. The coordinates (i_{sp_k}, j_{sp_k}) of these pixels sp_k are added to each $p_b(i,j)$. We also obtain a set of m tidy blocks built with N pixels. In other hand, Ip_{ROI} has an eccentric growth. The first pixels sp_k are near the centre of gravity G_{ROI} and the last pixels are near the ROI outline. The development of Ip_{ROI} is illustrated Figure 2. The embedding capacity depends at the same time on each ROI shape and size.

3. COLOR DCT-BASED WATERMARKING METHOD

To detect data after JPEG compression, we have created a method that is adapted to the main stages of JPEG algorithm [6]. The Section 3.1 shows how to use the color conversion to increase the redundancy. Then, in Section 3.2, we describe the function of DCT coefficients. Finally, in Section 3.3, we present the watermarking by induction.

3.1 Color conversion and redundancy

Color conversion is a part of the redundancy removal process in JPEG algorithm. JPEG handles colors as separate components. To resist against this first compression process, our method uses the same color conversion in order to increase the redundancy of messages. The image is also converted to YCrCb format. The messages, specific to each ROI, are embedded three times, one time by component. All the bits of the message are embedded on each component according to the synchronization described Section 2.3. Consequently, three watermarked channels Y', Cr' and Cb' are obtained. Then, an inverse transformation of color space is done. Three new watermarked components are obtained R', G' and B'. Finally a color watermarked image is built with these three watermarked channels.

3.2 The function of DCT coefficients

In JPEG compression algorithm, after the stage of color conversion, each channel is transformed from the spatial domain into the frequency domain. This process consists of dividing the luminance and chrominance information into square (typically 8×8) blocks and applying a two-dimensional Discrete Cosine Transform (DCT) to each block. Concerning our method, the used block are obtained after the PCA of the ROIs, presented Section 2.2. For each block k of the image, we obtain the follow DCT continuous component:

$$F_k(0,0) = \frac{1}{n}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} p_k(i,j) = \frac{1}{n}\sum_{x=0}^{N-1} p_k(x), \qquad (2)$$

where $p_k(x)$ is the intensity of pixel x in the block k, n the block side and N the pixels number of the block.
In the third stage of JPEG algorithm, each block of DCT coefficients is quantized. Each $F_k(0,0)$ coefficient is divided by $q(0,0)$ which is the first coefficient of quantization table. The resulting numerical value is rounded. The originality of our method is in this stage. We not round the resulting numerical value corresponding to the quantized DC component. We also have:

$$F'_k(0,0) = \frac{F_k(0,0)}{q(0,0)}, \qquad (3)$$

where $F'_k(0,0)$ is a floating value and not either an integer value. Contrary of JPEG algorithm, the floating part is kept then used during our watermarking process.

3.3 Inductive watermarking

Taking into account the constraint of robustness with regard to compression, in particular JPEG, several methods have been developed in the coefficients issued from the DCT transform [7]. Our method follows this JPEG-based approach. But the use of non-rounded DCT coefficients is new and original. The watermarking scheme which belongs to the QIM class [3] becomes inductive.
Firstly, we evaluate the nearest even integer which is inferior or equal to $F'_k(0,0)$. Then the difference between $F'_k(0,0)$ and this integer is noted $R_{F'_k(0,0)}$. We obtain:

$$R_{F'_k(0,0)} = F'_k(0,0) - \left\lfloor \frac{F'_k(0,0)}{2} \right\rfloor \times 2 \text{ and } R_{F'_k(0,0)} \in [0,2[\qquad (4)$$

where $\lfloor\ \rfloor$ represents the rounded value, the nearest inferior integer. $R_{F'_k(0,0)}$ is the value which is modified in order to embed data. Algorithm is normalized so as to detect the embedded bit b_k as follows:

$$b_k = \begin{cases} \mathbf{0} & \text{if } 0.0 \leq R_{F'_k(0,0)} < 1.0\ , \\ \mathbf{1} & \text{if } 1.0 \leq R_{F'_k(0,0)} < 2.0\ . \end{cases} \qquad (5)$$

Two values allow a maximum variance without error: the middles of intervals i.e. 0.5 and 1.5. Indeed with these values, b_k is correctly detected even if $R_{F'_k(0,0)}$ varies of 0.5. Consequently the watermarked value of $R_{F'_k(0,0)}$ is equal to:

$$R_{F'_{kw}(0,0)} = b_k + 0.5. \qquad (6)$$

where $R_{F'_{kw}(0,0)}$ is the stable value of the remainder $R_{F'_k(0,0)}$ specific to the embedded bit b_k.
The objective of our watermarking method is to modify pixels in order to have $R_{F'_k(0,0)} = R_{F'_{kw}(0,0)}$. To this end, we define d_k as the difference between these values. We also have:

$$d_k = R_{F'_{kw}(0,0)} - R_{F'_k(0,0)} = b_k + 0.5 - R_{F'_k(0,0)}. \qquad (7)$$

This difference is proportional to the number of modified pixels in the block k. Indeed a modification of pixel intensities transforms $F_k(0,0)$ and consequently $F'_k(0,0)$. Then this variation of $F'_k(0,0)$ modifies $R_{F'_k(0,0)}$. The pixels number to modify is N_{d_k}:

$$N_{d_k} = \lfloor n \times q(0,0) \times |d_k| \rfloor, \qquad (8)$$

where n is the side of the block k and q(0,0) the first coefficient of quantization table.
We underline that only a part of pixels of the block are modified. But if $N_{d_k} > n^2$ then $(N_{d_k} - n^2)$ pixels should be modified twice. The modified pixels are selected in order to reduce the variance in the block. In this way, the pixels modifications are invisible. Their intensities only change one or two grey level. The following equation describes this transformation:

$$p'_k(x) = p_k(x) + sign(d_k), \qquad (9)$$

where $p'_k(x)$ is the intensity of modified pixel x.
Finally, we obtain a watermarked block composed of modified and original pixels. To detect the embedded bit in the block k, we calculate the quantized and watermarked DCT Direct component $F'_{Wk}(0,0)$. Thus, we have:

$$F'_{kw}(0,0) = \frac{1}{n \times q(0,0)} \left(\sum_{x=0}^{N_{d_k}-1} p'_k(x) + \sum_{x=N_{d_k}}^{N-1} p_k(x) \right) \qquad (10)$$

Then we read the LSB of $F'_{kw}(0,0)$ which is equal to the embedded bit b_k.
Figure 3 gives the complete scheme of our watermarking method. It shows the PCA, the two symmetric color conversions and the DCT-based watermarking method.

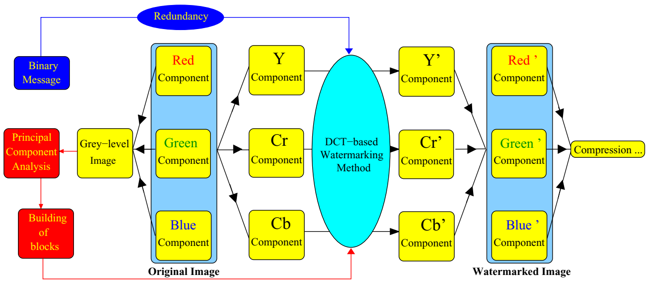


Figure 3: DCT-based watermarking method.

4. RESULTS

In this section we apply the proposed technique on image "Fish" (429×347), presented in Figure 1.a. In Section 4.1, we describe the insertion and detection schemes. Then our algorithm has been confronted with three treatments: image cropping in Section 4.2, rotation in Section 4.3, and JPEG color compression in Section 4.4.

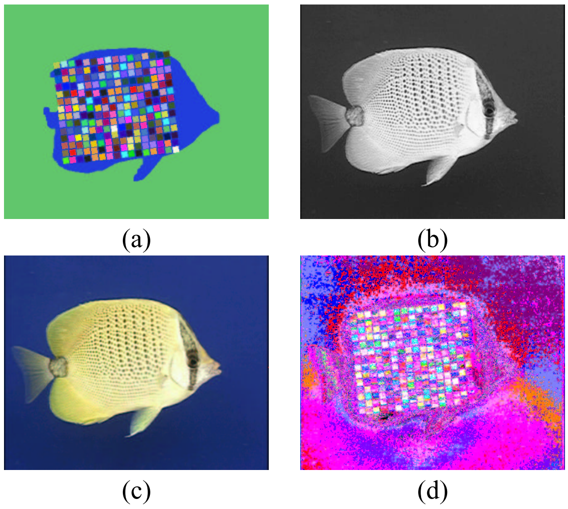


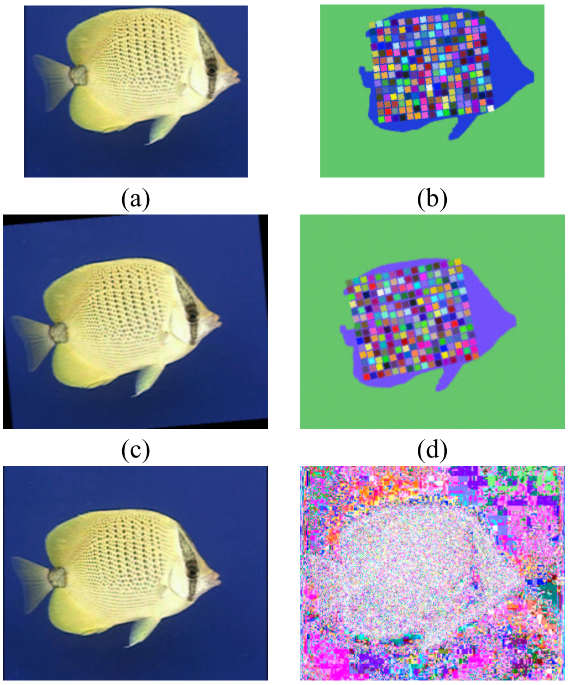
Figure 4: a) Insertion path and watermarked blocks, b) Y watermarked component, c) Color watermarked image, d) Difference between original and watermarked image.

4.1 Insertion and detection of data in a standard image

In the image "Fish" Figure 1.a, there is only one object on a clearly background. After segmentation, the object becomes a ROI where a message will be embedded. Then the PCA is applied to obtain the characteristics such as principal directions. These features are necessary to embed blocks of watermarked pixels. The Figure 4.a shows the ROI shape and the watermarked blocks built according to the insertion path Ip_{ROI}.
After this analyze, the watermarking method start. To increase the robustness, we have chosen to embed many times the messages. The redundancy is double. First the bits of message are repeated twice in the ROI. Then the color information is used to embed the information three times: one time by color component. The chosen color decomposition space is YCrCb because it is used in color JPEG algorithm. The Y watermarked component is presented Figure 4.b. Finally the watermarked color image is built with the three watermarked channels. The embedded data are invisible as the Figure 4.c shows it. But if we make the difference pixel by pixel between the original image Figure 1.a and the watermarked image Figure 4.c we visualize the embedded blocks in Figure 4.d. The PSNR between these two images is equal to 42.85 dB. The line 2 of Table 1 gives the results of detection in the ROI. Errors are explained by the color space transformation. After vote, all embedded bits are correctly detected.

Modification	% of right bits			
	On Y Component	On Cr Component	On Cb Component	after vote
Without	98.2	99.6	100	100
Cropping	89.8	96.0	99.6	100
Rotation	55.4	93.3	98.7	100
Compression (QF=80%)	56.3	88.0	85.3	100

Table 1: Results of bits detection according to modifications, on Y, Cr and Cb components then after vote.



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

Figure 5: a) Crop of watermarked image, b) Detection and visualization of watermarked blocks, c) 5 Degrees rotation of color watermarked image “Fish”, d) Color label image with watermarked blocks obtained after rotation, e) Compressed color watermarked image with QF = 80%, f) Difference between original and compressed color watermarked image with QF = 80%.

4.2 Robustness against image cropping

The Figure 5.a shows the results of cropping on the watermarked image ”Fish”, Figure 4.c. Only 68 % of image is recovered but the ROI stays complete. We obtain the detected blocks illustrated Figure 5.b. The results of detection after cropping image are given in Table 1. In the recovered ROI, some errors are detected. They are explained by the color space transformation. But after the vote, all bits are rightly detected. We can conclude that the synchronization resists to the cropping image if the ROI stays of course complete.

4.3 Robustness against Rotations

In this subsection, a 5 degree rotation is applied on watermarked image Figure 4.c. A rotated image is obtained Figure 5.c. After segmentation, we obtain the rotated ROI and we start the PCA. Then the detection path and the watermarked blocks are detected, Figure 5.d. We can observe that the ROI principal axes are just turned of 5 degrees. So the detection path stays usable. But if the principal axes change, the shape of watermarked blocks changes equally. It is explained by the discrete structure of images. It is one of reasons of detected errors in Table 1. With the double redundancy, the bits are watermarked 6 times, 2 times by channels. So a process of vote corrects errors. In the ROI, the message is correctly detected. Therefore our synchronization resists to rotations.

4.4 Detection results after Compression

In this subsection we applied the color JPEG algorithm on the watermarked image, Figure 4.c, to test the robustness against compression. The Figure 5.e shows the compressed watermarked image with a quality factor equal to 80 %. With our method, the detection is correct if the quality factor is superior to 75 %. Below this value, the noise is too significant. We observe in Table 1 that the messages are correctly detected thanks to the voting use. The Figure 5.f shows the difference between the compressed and watermarked image Figure 5.e and the original image Figure 1.a. The invisibility of watermarking is shown with the PSNR which is equal to 36.15 dB. The modifications made by our watermarking method are weak against compression modifications.

5. CONCLUSION AND PERSPECTIVES

In this paper, we have presented a color DCT-based watermarking method, which uses the content of images. To obtain the synchronization between the message and the image, an analysis is made and several ROIs are given. Then the 3 color components Y, Cr and Cb are used to embed three times the message. It is the first degree of redundancy. What is more, each bit is duplicated and embedded two times. It corresponds to the second degree of redundancy. As a result, the robustness is largely improved. Moreover, our watermarking method is inductive because we modify the non-rounded DCT coefficients. The watermarking is made by anticipating the quantization.
Our method is robust to a great variety of processing such as rotations, cropping or color JPEG compression. The embedded information stays invisible. The watermarking impact on the image is weak compared with others modifications like compression. To improve the quantity of embedded data, we intend to adapt the watermarked block shape to the ROI shape. As new research orientation, we would level-headed the size block according to the ROI size to be robust against zoom.

REFERENCES

[1] F. Hartung and M. Kutter, “Multimedia watermarking techniques,” in *IEEE Proceeding 87*, 1999., pp. 1079–1107.

[2] I.J. Cox, J. Killian, T. Leighton, and T. Shamoon, “Secure spread spectrum watermarking for multimedia,” *IEEE Trans. on Image Processing*, vol. 6, no. 12, pp. 1673–1687, 1997.

[3] B. Chen and G. Wornell, “An information-theoretic approach to the design of robust digital watermarking systems,” in *ICASSP’99, Phoenix, Arizona*, 1999.

[4] J.J.K. O Ruanaidh and T. Pun, “Rotation, scale and translation invariant spread spectrum digital image watermarking,” in *Signal Processing 66*, 1998., pp. 303–317.

[5] P. Bas and B. Macq, “A new video-object watermarking scheme robust to object manipulation,” in *Proc. of ICIP’01, Tessaloniki, Greece*, 2001, pp. ”526–529”.

[6] G. Wallace, “The JPEG still picture compression standard,” *Communication of the ACM*, vol. 34, no. 4, pp. 31–44, Apr. 1991.

[7] A. Bors I. and Pitas, “Image watermarking using dct domain constraints,” in *Proceedings ICIP, vol. 3, Lausanne, Switzerland*, 1999., pp. 231–234.