Content Based Watermarking for Securing Color Images

G. Lo-Varco and M. Dumas
CEM2 Laboratory, UMR CNRS 5507, University of Montpellier II, Cedex, FRANCE

W. Puech
LIRMM Laboratory, UMR CNRS 5506, University of Montpellier II, Cedex, FRANCE

In this article 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, rotation and color JPEG compression. As a basis for the watermarking, a segmentation algorithm is used. Based on the extracted regions, characteristic features are estimated 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 show the performance of the algorithm against spatial and frequential attacks.

Journal of Imaging Science and Technology 49: 464–473 (2005)

Introduction

Several techniques have been proposed in the literature to embed information in digital images.[1-4] These techniques provide various degrees of robustness. Watermarking applications include copyright protection, authentication, embedded and hidden information. Firstly, watermarking systems that are intended for copyright protection require a very high degree of robustness. Then, watermarking process for authentication belong to the fragile class of schemes. Slightest change in the image completely destroys the mark. Finally watermarking for embedding information requires resistance against moderate level of modification due to routine image processing such as compression or cropping. Our color watermarking method belongs to this last group of watermarking systems. Furthermore, in our application, the length of embedded data can relatively be important. In each region of interest (ROI), we embed a number of bits which fluctuates between 5 and 200 bits according to the ROI size.

The current article presents a technique for color watermarking of images,[5] based on non-rounded DCT-coefficients.[6–8] The hidden data are synchronized with the ROIs. This method mainly protects against interaction with visual content, geometrical manipulations[9–11] and JPEG compression.

Original manuscript received July 9, 2004

Corresponding Author: William Puech, William.puech@lirmm.fr

Presented in part at the Second International Conference on Color in Graphics, Imaging and Vision, 2004.

Supplemental Material—Figures 4–8 can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication.

ROIs Extraction and Descriptors

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

Obtaining the Content Mask

Segmentation Process

Our objective is to use image regions, which may not completely correspond to an object, but are generated by a simple segmentation algorithm.[12] The image content is described by all unconnected ROIs. These ROIs are required to have a minimum size to be considered for watermarking, otherwise they are merged with neighboring regions. We took the minimum size of a region to be 3000 pixels in total area so as to prevent the watermark relying on too small area for data hiding.

First, the color image is converted to gray level format. Then a region-growing algorithm is used. From each pixel, every adjacent pixel is added to the same region if their difference is minor. The strict definition of a minor difference is left to the discretion of the user, but usually we consider that two pixels are similar if the color difference is less than 20 gray levels (on a scale of 256 gray levels). 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", Fig. 1(a), and we obtain the associated content-mask, illustrated Fig. 1(b).

Work on the ROIs

After segmentation, the image is a set of convex areas. Each area has an outline made of boundary pixels. These pixels, by definition, would move to the ROI when there are geometrical modifications like rotations. Conse-

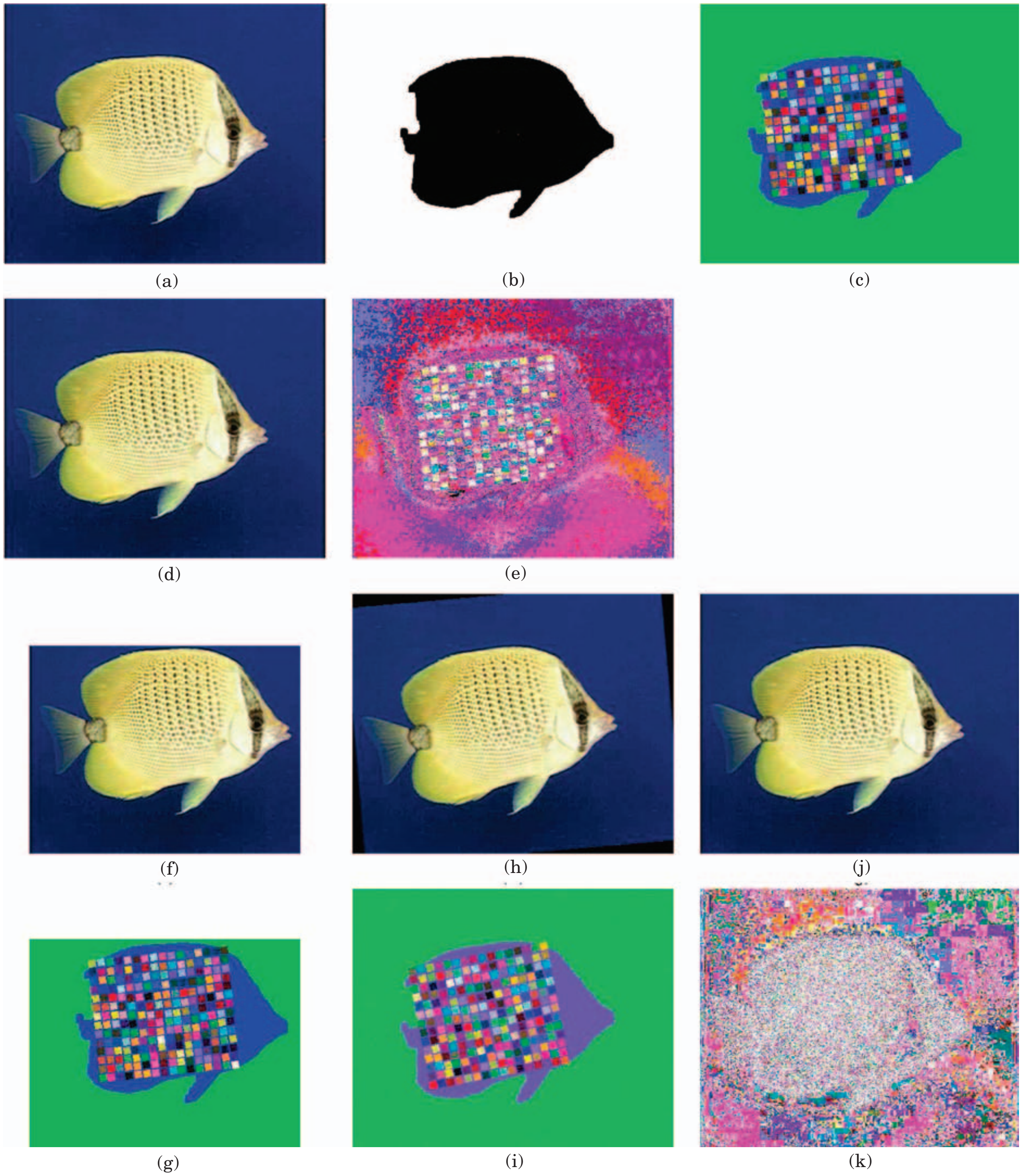


Figure 1. (a) The original image "Fish"; (b) The associated binary mask; (c) "Fish" color label image with blocks; (d) Color watermarked image; (e) Difference between original color image and color water-marked image; (f) Crop of watermarked image; (g) Detection and visualization of watermarked blocks; (h) 5-degree of rotation on color watermarked image "Fish"; (i) "Fish" color label image with watermarked blocks obtained after rotation; (j) Compressed color watermarked image with QF = 80%; (k) Difference between original and compressed color watermarked image with QF = 80%.

quently, to increase the robustness, these pixels are not used to define the ROI. Indeed, the ROI is reduced by a set of erosion and dilatation.[13] We lose information about the outline but we increase the robustness. If a deformation completely modifies the ROI outline, the erosion and dilatation decrease its effect, and we obtain the same simplified shape of ROI.

Robust Labeling of ROIs

After the reduction of ROIs, the image is a set of simplified and regular shapes. Each ROI corresponds to an area in the binary mask. Now the region-labeling algorithm indicates all the ROIs. This method is based on the Rosenfeld labeling operator.[14] Each pixel obtains a label, which depends on two characteristics: the intensity of pixel and its local neighborhood. This technique is applied here on 4-connexity. This type of parallel labeling methods is used because the image is read only one time. After an initialization, a table of equivalences between the labels is built. Finally, each pixel of ROI gets the specific ROI label L_{ROI}. After ROI extraction, an analysis is necessary to base correctly the hiding data.

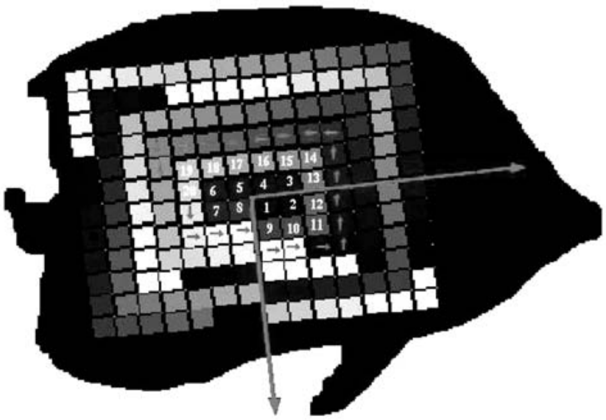


Figure 2. The appropriate frame of reference with the building order of blocks.

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 that, we define three features of ROI: the size descriptor, the position descriptor, and last the orientation and shape descriptor.

A Size Descriptor: The ROI Area

The ROI size $S(ROI)$ is the first calculated parameter. It is represented by the number of pixels which L_{ROI} have the same ROI label. Accordingly, a first classification of ROIs is possible. A quantization is made with a standard size of ROI S_s. We finally obtain a factor of size:

$$F_S(ROI) = \frac{S(ROI)}{S_s}. \tag{1}$$

S_s generally is equal to 10% of the original image size, i.e., 15,000 pixels with an image consisting of an array 400×500 of pixels like "Fish" image. In this way, the detection of hidden data should be possible after a scaling operation.

A Position Descriptor: The Spatial Center

To indicate the position of ROIs in the image, we use the spatial center G_{ROI}. The moments of first degree noted $\mu_i(ROI)$ and $\mu_j(ROI)$ precisely locate this singular point of ROI. The average i-coordinate of ROI pixels is $\mu_i(ROI)$. The average j-coordinate of ROI pixels is $\mu_j(ROI)$. To calculate them, we have:

$$\begin{cases} \mu_i(ROI) = \dfrac{1}{S(ROI)} \displaystyle\sum_{k=0}^{S(ROI)-1} i(k) \\ \mu_j(ROI) = \dfrac{1}{S(ROI)} \displaystyle\sum_{k=0}^{S(ROI)-1} j(k) \end{cases} \tag{2}$$

where $i(k)$ and $j(k)$ respectively are the vertical and horizontal coordinates of the pixel k. We finally obtain:

$$G_{ROI} = G[\mu_i(ROI), \mu_j(ROI)]. \tag{3}$$

This centroid G_{ROI} is the origin of a specific frame of reference. This frame will be used to embed the data.

An Orientation and Shape Descriptor: Principal Component Analysis

Our embedding scheme uses a frame of reference depending on the ROI shape. To build this frame, one point and two directions are necessary. The point G_{ROI} has been calculated in the previous section. Now we have to determinate these two specific directions of 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 $V_x(ROI)$, the vertical variance $V_y(ROI)$, and the covariance $V_{xy}(ROI)$ given by:

$$\begin{cases} V_x(ROI) = \dfrac{1}{S(ROI)} \displaystyle\sum_{k=0}^{S(ROI)-1} (i(k) - \mu_i(ROI))^2 \\ V_y(ROI) = \dfrac{1}{S(ROI)} \displaystyle\sum_{k=0}^{S(ROI)-1} (j(k) - \mu_j(ROI))^2 \\ V_{xy}(ROI) = \dfrac{1}{S(ROI)} \displaystyle\sum_{k=0}^{S(ROI)-1} [i(k) - \mu_i(ROI][j(k) - \mu_j(ROI)]. \end{cases} \tag{4}$$

This set of coefficients is represented by a 2×2 covariance matrix $C_0(ROI)$ noted:

$$C_0(ROI) = \begin{bmatrix} V_x(ROI) & V_{xy}(ROI) \\ V_{xy}(ROI) & V_y(ROI) \end{bmatrix}. \tag{5}$$

An analysis of this matrix gives two eigenvalues $\lambda_1(ROI)$ and $\lambda_2(ROI)$ and two associated eigenvectors $\vec{V}_1(ROI)$ and $\vec{V}_2(ROI)$.[15] These vectors denote directions providing maximal variance of the ROI. In this context, these couples $\{\lambda, \vec{V}\}$ represent the major and minor axes of the ROI. These two axes are illustrated in Fig. 2. Associated with the centroid $G(ROI)$, they form a frame of reference adapted to receive the hidden data. The hidden data consequently are oriented and synchronized according to each frame of reference.

Synchronization of Hidden Data According to the ROIs

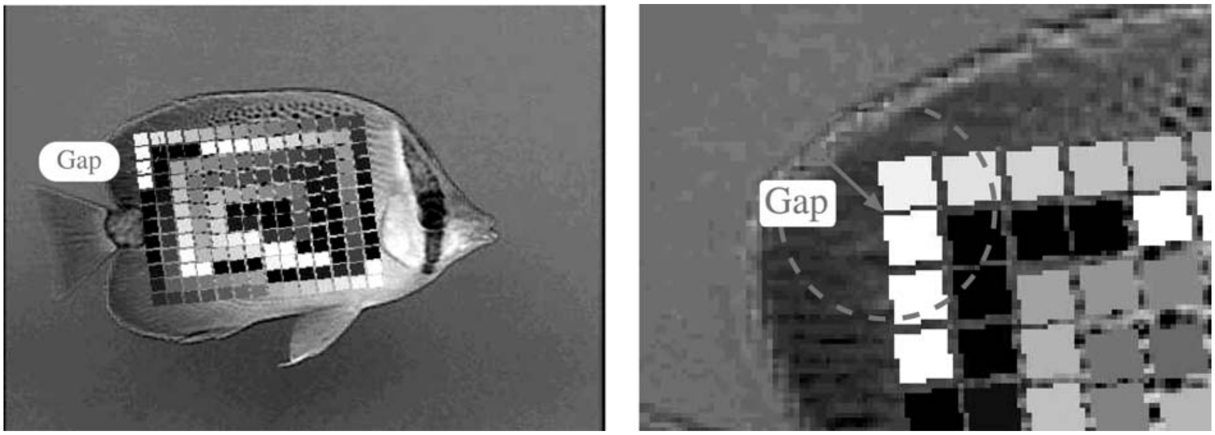
Our watermarking method uses unitary blocks of N pixels. These blocks are definitely differently according to the ROI shape and the ROI orientation. Moreover they are built in a specific order to keep the information integrity.

Specific Definition of ROI Unitary Blocks

Our watermarking method implies several constraints on the unitary blocks. 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 $\vec{V}_1(ROI)$ and $\vec{V}_2(ROI)$ are the two oriented sides of block. In this way, the block is built with all pixels $p_b(x,y)$ such that:

$$\begin{cases} 0 \le x < \left\|\vec{V}_1(ROI)\right\| \\ 0 \le y < \left\|\vec{V}_2(ROI)\right\| \end{cases}, \text{ and } \left\|\vec{V}_1(ROI)\right\| = \left\|\vec{V}_2(ROI)\right\| = \sqrt{N}, \tag{6}$$

where $\|\vec{u}\|$ is the norm of vector $\vec{u}$ and (x,y) the coordinates of pixels in the ROI reference frame.



(a) (b)

Figure 3. (a) Minimal gap between the outline and the blocks; and (b) enlarging of the outline.

The block is also a set of N 2D variables $p_b(x, y)$ (pixels of block) with $0 \leq b < N$. The unitary block is specifically oriented and adapted to each ROI.

To obtain the coordinates (i, j) of pixels p_b in the image, we change the frame of reference and we obtain:

$$\begin{cases} i = x + i_G = x + \mu_i(ROI) \\ j = y + j_G = y + \mu_j(ROI). \end{cases} \tag{7}$$

Building Order of Watermarking Block

To embed data, an order is followed. We use one more time the couples $\{\lambda, \vec{V}\}$ 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 (x_{sp_k}, y_{sp_k}) of these pixels sp_k are added to each $p_b(x, y)$. We also obtain a set of m tidy blocks built with N pixels, illustrated Fig. 1(c). In other hand, Ip_{ROI} has an eccentric growth. The first pixels sp_k are near the center of gravity G_{ROI} and the last pixels are near the ROI outline. A minimal gap is kept between the outline and the blocks. It corresponds to the minimal distance between the outline and any starting pixel sp_k. This gap is left to the discretion of the user, but for our tests we considered a gap equal to $\sqrt{N}/2$, i.e., the major part of block is in the ROI. The development of Ip_{ROI} is illustrated in Fig. 2. The gap is described Figs. 3(a) and 3(b).

In this section, we have described how to use the content of color image. The following section presents in details how to embed data by watermarking.

Color DCT-Based Watermarking Method

To detect the data after JPEG compression, we have created a method that is adapted to the main stages of the JPEG algorithm.[16] In this way, the embedded data is not eliminated during compression process. The section below, shows how to use the color conversion to increase the redundancy. Then we present the interest of DCT coefficients in JPEG algorithm and so in our method. Finally, we demonstrate the originality of our method, watermarking by induction.

Color Conversion and Redundancy

Color conversion is a part of the redundancy removal process in JPEG algorithm. JPEG handles colors as separate components. Therefore, it can be used to compress data from different color spaces such RGB, YCrCb, and CMYK. However, the best compression results are achieved if the color components are independents (no correlated) such as in YCrCb. In this color space, most of information is concentrated in the luminance (Y) and less in the chrominance (Cr and Cb). RGB color components can be converted via a linear transformation into YCrCb components as the equations below show:

$$\begin{cases} Y = (0.2989 \times R) + (0.5866 \times G) + (0.1145 \times B) \\ Cr = (0.7132 \times R) - (0.7132 \times Y) + 128 \\ Cb = (0.5647 \times B) - (0.5647 \times Y) + 128. \end{cases} \tag{8}$$

Another advantage of using the YCrCb color space comes from reducing the spatial resolution of the Cb and Cr chrominance components. Because chrominance does not need to be specified as frequently as luminance: every other Cr and Cb element can be discarded. As a consequence, a data reduction of 3:2 is obtained by transforming RGB (4:4:4) format into YCrCb (4:2:2) format. The conversion in color space is a first step toward image compressing.

To resist against this first compression process, our method uses color conversion in order to increase the redundancy of messages. The messages, specific to each ROI, are embedded three times, one time by each component. All the bits of the message are embedded on each component according to the synchronization described above. 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.

The Function of DCT Coefficients

In the 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. In our method, the blocks used are obtained after the PCA of the ROIs, as presented above.

The DCT converts each block of spatial information into an efficient frequency-space representation that is better suited for compression. Specifically, the transformation produces an array of coefficients for real valued basis functions that represent each block of data in frequency space. For each block k of the image, we obtain the follow DCT continuous component:

$$F_k(0,0) = \frac{1}{n}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} p_k(i,j) = \frac{1}{n}\sum_{x=0}^{N-1} p_k(x), \qquad (9)$$

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 the 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 result is rounded. We also have:

$$F_k^{'}(0,0) = \left[\frac{F_k(0,0)}{q(0,0)}\right], \qquad (10)$$

where [ ] represents the value rounded to the nearest integer.

The originality of our method is in this stage. We do not round the quantized DC component. We obtain:

$$F_k^{'}(0,0) = \frac{F_k(0,0)}{q(0,0)}, \qquad (11)$$

where $F_k^{'}(0,0)$ is a floating value and not necessarily an integer value. Contrary to the JPEG algorithm, the floating part is kept and then used during our watermarking process.

Inductive Watermarking

In order to take into account the constraint of robustness with regard to compression, in particular JPEG, several methods have been developed in the coefficients obtained from the DCT transform.[17] Our method follows this JPEG-based approach. But the use of non-rounded DCT coefficients is new and original. The watermarking scheme 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:

$$R_{F_k^{'}(0,0)} = F_k^{'}(0,0) - \left\lfloor \frac{F_k^{'}(0,0)}{2} \right\rfloor \times 2 \ \text{ and } \ 0.0 \le R_{F_k^{'}(0,0)} < 2.0, \qquad (12)$$

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

$$b_k = \begin{cases} \mathbf{0} & \text{if } 0.0 \le R_{F_k^{'}(0,0)} < 1.0, \\ \mathbf{1} & \text{if } 1.0 \le R_{F_k^{'}(0,0)} < 2.0. \end{cases} \qquad (13)$$

Two values allow a maximum variance without error: the mid-points 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 by 0.5. Consequently the watermarked value of $R_{F_k^{'}(0,0)}$ is equal to:

$$R_{F_k^{'}(0,0)}\ R_{F_k^{'}(0,0)} = b_k + 0.5, \qquad (14)$$

where $R_{F_k'(0,0)}$ 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 $R_{F_k'(0,0)}$. To this end, we define d_k as the difference between these values. We also have:

$$\begin{aligned} d_k &= R_{F_{kw}^{'}(0,0)} - R_{F_k^{'}(0,0)} \\ &= b_k + 0.5 - R_{F_k^{'}(0,0)}. \end{aligned} \qquad (15)$$

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

$$N_{d_k} = \lfloor n \times q(0,0) \times |d_k| \rfloor, \qquad (16)$$

where n is the side of the block k and $q(0,0)$ the first coefficient of quantization table.

We emphasis that only a part of pixels of the block is 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 gray level. The following equation describes this transformation:

$$p_k'(x) = p_k(x) + sign(d_k), \qquad (17)$$

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 $F'_{kw}(0,0)$. Thus, we have:

$$F'_{kw}(0,0) = \frac{1}{n \times q(0,0)}\left(\sum_{x=0}^{N_{d_k}-1} p'_k(x) + \sum_{x=N_{d_k}}^{N-1} p_k(x)\right). \qquad (18)$$

Then we read the LSB of $F'_{kw}(0,0)$. Figure 4 *(available in color as Supplemental Material which can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication)* gives the complete scheme of our watermarking method. It shows the PCA, the two symmetric color conversions and the DCT-based watermarking method.

Let us now provide an example of our watermarking method. Take the central area of the first block, which has been built in "Fish" image, Fig. 1(a). On the Y luminance component, it corresponds to the following 8×8 matrix:

$$M_W = \begin{bmatrix} 190 & 152 & 136 & 158 & 176 & 191 & 202 & 223 \\ 225 & 212 & 204 & 212 & 220 & 221 & 227 & 235 \\ 244 & 245 & 245 & 245 & 245 & 244 & 243 & 241 \\ 244 & 242 & 243 & 237 & 237 & 242 & 241 & 238 \\ 240 & 230 & 212 & 188 & 193 & 215 & 236 & 225 \\ 223 & 204 & 172 & 133 & 127 & 157 & 196 & 212 \\ 222 & 195 & 152 & 100 & 90 & 125 & 179 & 208 \\ 230 & 206 & 163 & 114 & 109 & 144 & 194 & 215 \end{bmatrix},$$

where each coefficient gives the grey-level of pixels. We calculate the DCT continuous component with Eq. (9) and we obtain $F_1(0,0) = 1608.625$. Then $F_1(0,0)$ is quantized by $q_Y(0,0) = 16$ which is the first coefficient in the lumi-

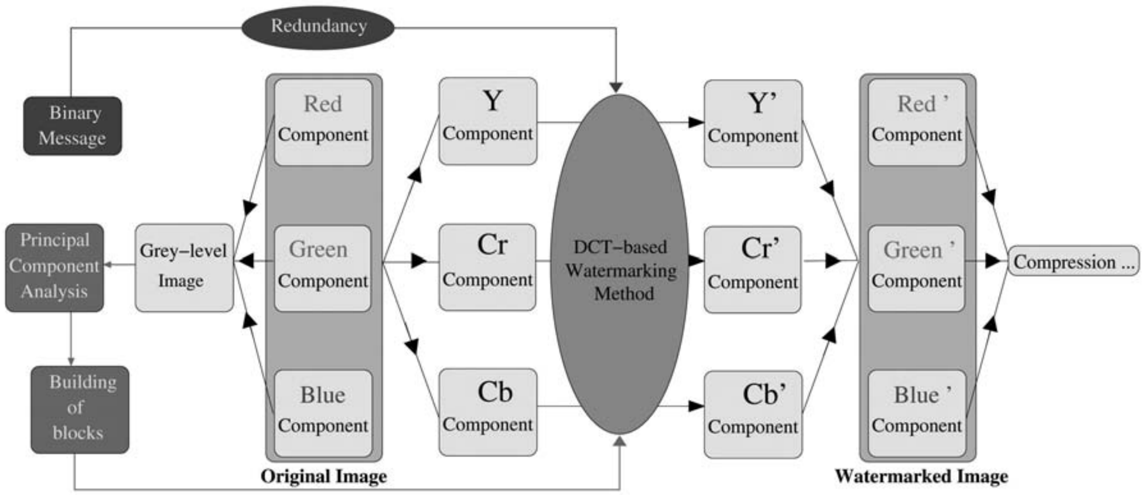


Figure 4. DCT-based watermarking method. *Supplemental Material—Figure 4 can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication.*

nance quantization table. So we have $F_1'(0,0) = 100.539$. According to the Eq. (12), the rest $R_{F_k'(0,0)}$ is equal to 0.539. The bit to be embedded can be a 0 or an 1. Let us carry out the insertion of a bit $b_1 = 0$. In this case $R_{F_k'(0,0)} = 0.5$, then from the Eq. (15) we have $d_1 = 0.5 - 0.539 = -0.039$. The value of d_1 gives the variation which should be applied to $F_1'(0,0)$ to obtain the desired stable remainder $R_{F_{kw}'(0,0)}$. We note $F_1{}'_w(0,0)$, the corresponding value of the DCT continuous coefficient. To obtain $F_1{}'_w(0,0)$, from Eq. (16) we have to modify N_{d1} pixels of block. These 5 pixels decrease by one gray level. They are selected in order to reduce the variance in the block. From Eq. (18) we get:

$$
\begin{aligned}
F_{1w}'(0,0) &= \frac{1}{8\times 16}\left(\sum_{x=0}^{5-1} p_k'(x) + \sum_{x=5}^{8^2-1} p_k(x)\right)\\
&= \frac{1}{8\times 16}\left(\sum_{x=0}^{4} (p_k'(x)-1) + \sum_{x=5}^{63} p_k(x)\right)\\
&= \frac{1}{8\times 16}\left(\sum_{x=0}^{63} p_k(x) + \sum_{x=0}^{4} (-1)\right)\\
&= \frac{F_1(0,0)}{16} - \frac{5}{128} = 100.5.
\end{aligned} \tag{19}
$$

Then we calculate $F_{1W}(0,0)$:

$$
\begin{aligned}
F_{1w}(0,0) &= q(0,0)\times F_{1w}'(0,0)\\
&= 16\times 100.5 = 1608.0.
\end{aligned} \tag{20}
$$

Finally the watermarked matrix M_W is:

$$
M_W = \begin{bmatrix}
190 & 152 & 136 & 158 & 176 & 191 & 202 & 223\\
225 & 212 & 204 & 212 & 220 & 221 & 227 & 235\\
244 & 245 & 245 & 245 & 245 & 244 & 243 & 241\\
244 & 242 & 243 & 237 & 237 & 242 & 241 & 238\\
240 & 230 & 212 & 188 & 193 & 215 & 236 & 225\\
223 & 204 & 172 & 133 & 127 & 157 & 196 & 212\\
222 & 195 & 152 & \mathbf{99} & \mathbf{89} & \mathbf{124} & 179 & 208\\
230 & 206 & 163 & \mathbf{113} & \mathbf{108} & 144 & 194 & 215
\end{bmatrix},
$$

where the values written in **bold** decreased by only one gray level.

To extract the bit from the value $F_{1W}(0,0)$, we calculate $F_{1w}'(0,0) = 1608/16 = 100.5$. Consequently we obtain $R_{F_{kw}'(0,0)} = 100.5\%\ 2 = 0.5$. The detected bit b_1 is 0. With this example, we illustrate the inductive aspect of our watermarking method.

Results

In this section we apply the proposed technique on two images with ROIs, “Fish” (429 × 347) Fig. 1(a) and “Objects” (1013 × 760) Fig. 5(a). Then several processing attacks are tested on these two images. Below, we describe the detection scheme with the image “Objects”. The detected data are detailed to be used as references in the check of robustness. Then our algorithm has been confronted with three treatments: image cropping, rotation and JPEG color compression.

Detection of Data in a Standard Image

In this section, we work on the original image “Objects”, which is presented in Fig. 5(a) *(available in color as Supplemental Material which can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication)* . There are 8 objects on a clear background. After segmentation, each object becomes a ROI where a message can be embedded. We regard all regions under 3000 pixels as being too small. Then the picture with the ROI shapes is analyzed to obtain some characteristics such as principal directions. These data are necessary to detect blocks of water-marked pixels. The detection path Dp_{ROI} uses equally these characteristics. Figure 5(b) shows the ROI shapes and the watermarked blocks built according to the Dp_{ROI}. The specific labels associated with each ROI are likewise shown.

After this analysis, the watermarking can start. To increase robustness, we have chosen to embed the messages many times. The redundancy is thereby doubled. First the bits of the message are repeated twice in each 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 Figs. 5(c) and 5(d) present respectively the Y and the Cr watermarked components. Finally the watermarked color image is built with the three watermarked channels. The embedded data are invisible as the Fig. 5(e) shows. But if we evalu-

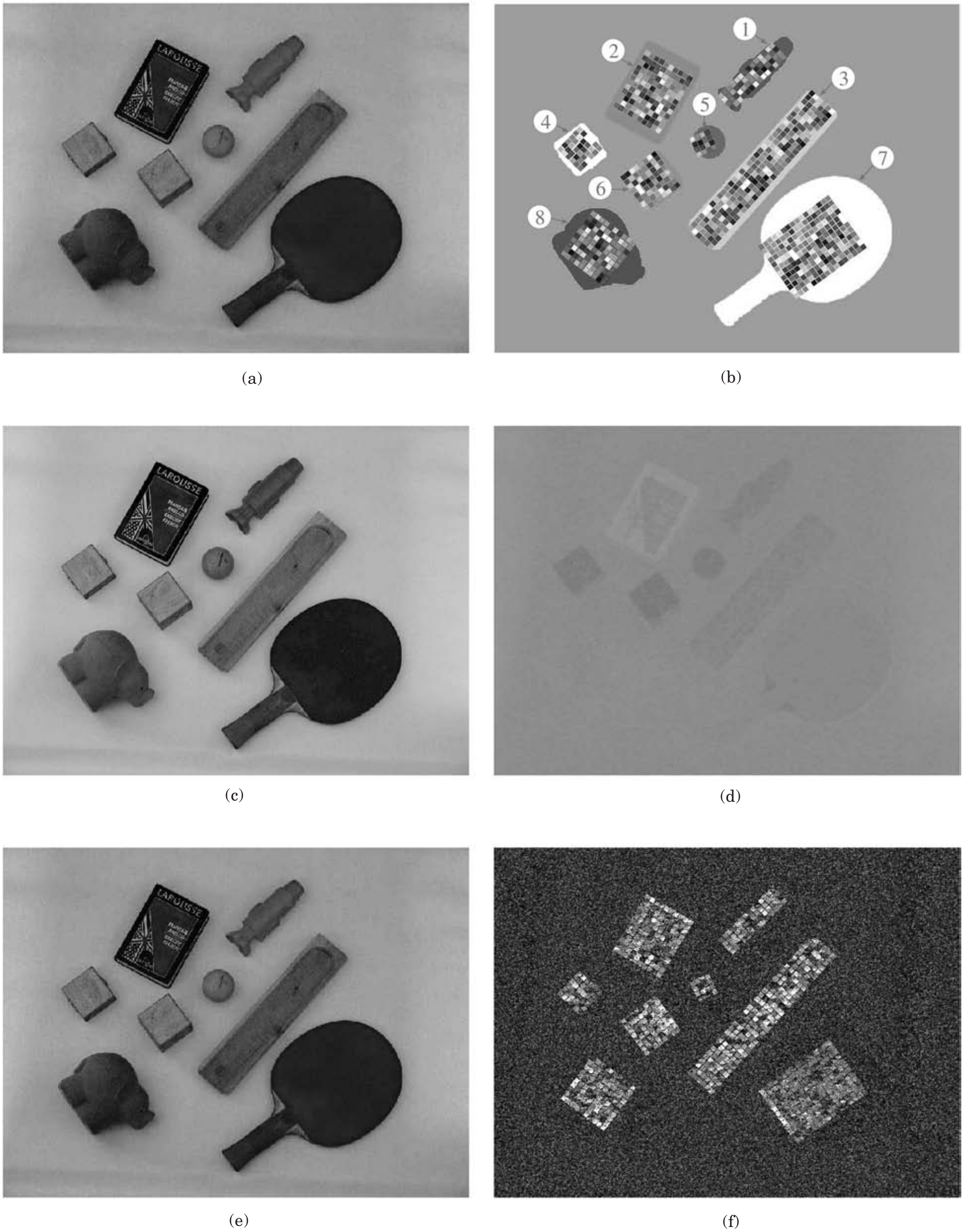
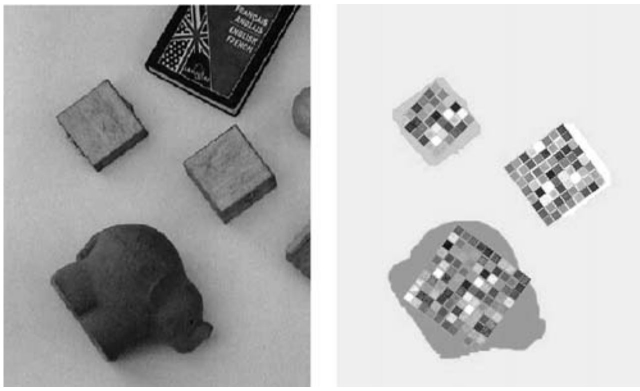


Figure 5. (a) Original image; (b) Detection path and watermarked blocks; (c) Y watermarked component; (d) Cr watermarked component; (e) Color watermarked image; (f) Difference between original and watermarked image. *Supplemental Material—Figure 5 can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication.*

ate the difference pixel by pixel between the original image Fig. 5(a) and the watermarked image Fig. 5(e) we visualize the embedded blocks shown in Fig. 5(f). The PSNR between these two images is equal to 50.52 dB.

Table I gives some information about the embedded data in each ROI: the message length, the number of embedded bits by component, the size of detected block, the ROI size and the embedding rate. A first analysis



(a) (b)

Figure 6. (a) Crop of watermarked image; and (b) Detection and visualization of watermarked blocks. *Supplemental Material—Figure 6 can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication.*

shows that the message length depends on the ROI size. Indeed if the ROI size increases, the number of pixel blocks increases equally. Consequently, more bits can be embedded. For example, in the first ROI, 56 bits are detected and in the third ROI, 3 times larger, the number of detected bits is multiplied by 4. Then we observe that the number of detected bits is twice the message length, which corresponds to the first degree of redundancy. For example, in the fifth ROI, the binary message is 00110101, and the detected bits are **00110101**00110101. Each bit is repeated. It is the result of the embedding, which is made in two different places in the ROI. Therefore the detection results are improved. On other hand, column 4 of Table I presents the block sizes used in each ROI. We observe that the size varies slightly, around 6%. As our method needs square blocks, the input data is the size of the block edge if it is aligned with the edges of the image. Then this size is modified according to the ROI orientation and the desired block size. Finally each region obtains a block according to this specific definition. By using the block size and the ROI size, we evaluate the embedding rate of ROIs:

$$E_r(ROI) = \frac{ROI\ block\ size \times Number\ of\ embedded\ bits\ in\ the\ ROI}{ROI\ size}. \tag{21}$$

This rate varies from 31.7% to 83.9% according to the ROI shape. It shows the disadvantage to use square blocks and the limit of the detection path Dp_{ROI}. Indeed there are still many pixels to watermark information.

To validate our method more precisely, another image is used to embed information. Figure 1(d) shows the color watermarked image. The length of watermarked information is 112 bits and the number of embedded bits is 224. To visualize the embedding data, the difference pixel by pixel between the original image, Fig. 1(a), and the watermarked image is shown Fig. 1(e).

Robustness Against Image Cropping

This subsection shows how our method resists a particular geometrical deformation: image cropping. The results of detection are presented and then analyzed. We thus evaluate the robustness of our method.

TABLE I. Embedded Messages and Embedding Rate for Each ROI

ROI	Message length (bits)	Number of embedded bits by component	block size (pixels)	ROI size (pixels)	Embedding rate
1	28	56	113	10581	59.8%
2	60	120	117	26441	53.1%
3	102	204	113	35261	65.4%
4	18	36	113	7833	51.2%
5	8	16	106	4138	40.9%
6	32	64	113	8620	83.9%
7	96	192	113	68474	31.7%
8	50	100	113	24940	45.3%

TABLE II. Results of Bits Detection After Image Cropping on Y, Cr and Cb Watermarked Components and Results after Voting

	% of right bits (right bits/embedded bits)			
ROI	On Y Component	On Cr component	On Cb component	After voting
1	—	—	—	—
2	—	—	—	—
3	—	—	—	—
4	88.9% (32/36)	94.4% (34/36)	91.7% (33/36)	100% (18/18)
5	—	v	—	—
6	75.0% (48/64)	93.7% (60/64)	93.7% (60/64)	100% (32/32)
7	—	—		
8	76.0% (76/100)	90.0% (90/100)	94.0% (94/100)	100% (50/50)

Figure 6(a) *(available in color as Supplemental Material which can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication)* shows the results of cropping on the watermarked image "Objects". Only 25% of image could be recovered. In this case, the number of ROIs decreases and only 4 ROIs are detected. The PCA is made and the detection paths are calculated. We obtain the detected blocks illustrated in Fig. 6(b). We can observe that only 3 ROIs appear on the picture. In fact, when an ROI touches the picture boundaries, we supposed that it is truncated and the detected message will naturally be wrong. So the detection does not start in this ROI.

The results of detection after cropping image are given in Table II. The ROIs, which have disappeared, are of course not detected and their messages are lost. In the recovered ROIs some errors are detected, they are explained by the color space transformation. But after the selection, all bits are rightly detected.

Figure 1(f) shows the results of cropping on the watermarked image "Fish". We obtain the detected blocks illustrated in Fig. 1(g). All bits are also rightly detected. We can conclude that the synchronization resists the image cropping. Our method is robust against this kind of geometrical modification.

Robustness Against Rotations

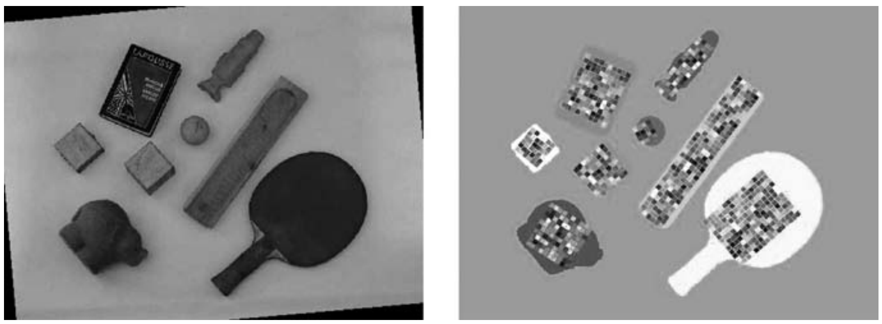
In this subsection, another geometrical modification is used to evaluate the robustness of our method. A 5 degree rotation is applied on two watermarked images, Figs. 5(e) and 1(d). Two rotated images are obtained Figs. 7(a) and 1(h) *(available in color as Supplemental Material which can be found in color on the IS&T website*

TABLE III. Results of Bits Detection After 5-degree of Rotation on Each Color Component Y, Cr and Cb.

	% of right bits (right bits/embedded bits)			
ROI	On Y Component	On Cr component	On Cb component	After voting
1	92.9% (52/56)	94.6% (53/56)	89.3% (50/56)	100% (28/28)
2	87.5% (105/120)	84.2% (101/120)	94.2% (113/120)	100% (60/60)
3	75.0% (153/204)	86.7% (177/204)	93.6% (191/204)	100% (102/102)
4	87.5% (14/16)	93.7% (15/16)	93.7% (15/16)	100% (8/8)
5	80.6% (29/36)	86.1% (31/36)	88.9% (32/36)	100% (18/18)
6	56.2% (36/64)	93.7% (60/64)	92.2% (59/64)	100% (32/32)
7	81.2% (156/192)	94.3% (181/192)	87.5% (168/192)	100% 96/96)
8	57.0% (57/100)	87.0% (87/100)	94.0% (94/100)	100% (50/50)

TABLE IV. Results of Bits Detection After Compression on Each Color Component Y, Cr and Cb.

	% of right bits (right bits/embedded bits)			
ROI	On Y Component	On Cr component	On Cb component	After voting
1	69.6% (39/56)	87.5% (49/56)	75.0% (42/56)	100% (28/28)
2	75.8% (91/120)	87.5% (105/120)	81.7% (983/120)	100% (60/60)
3	75.5% (154/204)	88.2% (180/204)	88.7% (181/204)	100% (102/102)
4	72.2% (26/36)	94.4% (34/36)	83.3% (30/36)	100% (18/18)
5	68.8% (11/16)	75.0% (12/16)	87.5% (14/16)	100% (8/8)
6	50.0% (32/64)	81.3% (52/64)	82.8% (53/64)	100% (32/32)
7	75.5% (145/192)	76.0% (146/192)	81.25% (156/192)	100% 96/96)
8	50.0% (50/100)	82.0% (82/100)	88.0% (88/100)	100% (50/50)



(a) (b)

Figure 7. (a) 5-degree of rotation on color watermarked image "Objects"; and (b) "Objects" color label image with watermarked blocks obtained after rotation. *Supplemental Material—Figure 7 can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication.*

(www.imaging.org) for a period of no less than two years from the date of publication). With this modification, we check the robustness of synchronization between image and hidden data. After segmentation, we obtain the rotated ROIs and we start the PCA. Then the detection paths and the watermarked blocks are detected, illustrated in Figs. 7(b) and 1(i). We can observe that the ROI principal axes are just rotated by 5 degrees. So the detection path remains usable. On the other hand, if the principal axes change, the shapes of watermarked blocks change equally, as is explained by the discrete structure of images. Moreover the discretization creates another problem: if the image is turned, several pixels that compose the boundaries of blocks can change and the corresponding direct DCT components can consequently be modified.

It is one of reason for the detected errors, which are illustrated in Table III. This Table gives the percentage of right bits in each ROI, on Y, Cr and Cb components. With the double redundancy, the bits are watermarked 6 times, i.e., 2 times in each channel. So a process of selection is started, and the results are presented in the last column of Table III. In each ROI, the message is correctly detected. Therefore, our synchronization resists rotation.

Detection Results After Compression

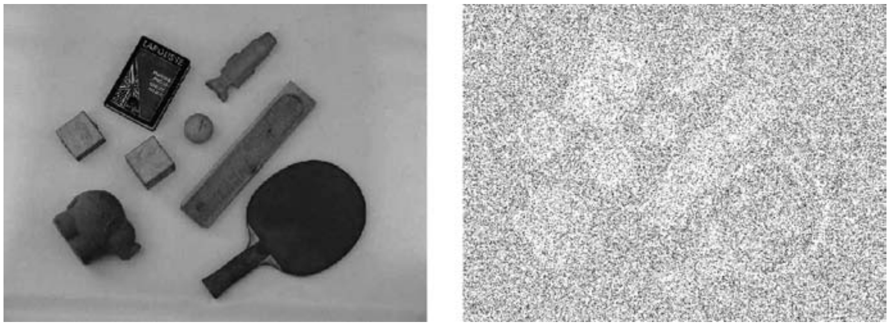
In this subsection we applied the color JPEG algorithm on the watermarked image to test the robustness against compression. Figure 8(a) *(available in color as Supplemental Material which can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication)* shows the compressed watermarked image with a quality factor equal to 80%. To decrease the image size, the JPEG compression algorithm modifies the color pixels. And if the quality factor is small, the modifications are important and the modified pixels are numerous. With our method, the detection is correct if the quality factor is superior to 75%. Below this value, the noise is too significant. Table IV gives the percentage of right detected bits in each ROI on three components. We observe that the messages are correctly detected, thanks to the voting use.

Figure 8(b) shows the difference between the compressed and watermarked image of Fig. 8(a) and the original image, Fig. 5(a). With this difference image, the invisibility of watermarking is illustrated, the PSNR is equal to 43.19 dB.

Figure 1(j) shows the compressed watermarked image "Fish" with a quality factor equal to 80%. Figure 1(k) shows the difference between the compressed and watermarked image Fig. 1(j) and the original image, Fig. 1(a). The message is correctly detected thanks to the selection strategy used. The modifications made by our watermarking method are weak compared to compression modifications.

Conclusion and Perspectives

In this article, we have presented a color DCT-based watermarking method, which exploits the content of images. To obtain the synchronization between the message and the image, an analysis is made and several ROIs are created. The content of the image is used to synchronize message and image. Then the three color components Y, Cr and Cb are used to embed the message three times. 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.



(a) (b)

Figure 8. (a) Compressed color watermarked image with QF = 80%; and (b) Difference between original and compressed color watermarked image with QF = 80%. *Supplemental Material—Figure 8 can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication.*

As a result, the robustness is greatly improved. Finally, our watermarking method is inductive because we modify non-rounded DCT coefficients. The watermarking is made in anticipation of quantization.

The different results illustrate the fact the robustness of our watermarking method depends on the image color segmentation. But the primary objectives were realized. Our method is robust to a variety of processing attacks such as rotation, cropping or color JPEG compression. The embedded information remains invisible. Moreover the watermarking impact on the image is weak compared with others modifications like compression.

All the results have been obtained with a watermarked block size around 11×11. If the block size block is smaller, the number of embedded messages increases but the robustness decreases. On the contrary if the block size increases, the robustness is improved but the hidden data become very small. To improve the quantity of embedded data, we intend to adapt the watermarked block shape to the ROI shape. All of the block would not square and the embedding rate increases. As a new research orientation, we would like to change the size of the block according to the ROI size in order to be robust against zoom.

References

1. P. Bas, J. M. Chassery and B. Macq, "Image watermarking: an evolution to content based approaches", *Pattern Recognition* **35,** 545–561 (2002).
2. F. Deguillaume, S. Voloshynovskiy and T. Pun, "Hybrid robust watermarking resistant against copy attack", in *EUSIPCO'02*, (EURASIP, Toulouse, France, 2002), p. 25.
3. F. Hartung and M. Kutter, "Multimedia watermarking techniques", *Proc. IEEE* **87,** 1079–1107 (1999).
4. G. Voyatzis and I. Pitas. "The use of watermarks in the protection of digital multimedia products", *Proc. IEEE 87*, 1197–1207 (1999).
5. G. Chareyron and A. Tremeau, "Watermarking of color images based on a multi-layer process", *Proc. CGIV'02*, (IS&T, Springfield, VA, 2002), pp. 77–80.
6. A.G. Bors and I. Pitas, "Image watermarking using block site selection and DCT domain con-straints", *Optics Express* **3**(12), 512–522 (1997).
7. I.J. Cox, J. Killian, T. Leighton, and T. Shamoon, Secure spread spectrum watermarking for multimedia", *IEEE Trans. Image Proc.* **6**(12), 1673–1687 (1997).
8. E. Koch and J. Zhao, "Towards robust and hidden image copyright labeling", in *IEEE Workshop on Nonlinear Signal and Image Processing*, (IEEE Press, Los Alamitos, CA, 1995), pp. 452–455.
9. M. Kutter, "Watermarking resisting to translation, rotation and scaling", *Proc. SPIE* **3528,** 423–431 (1998).
10. M. P. Queluz, "Content-based integrity protection of digital images", *Proc. SPIE* **3657**, 85–93 (1999).
11. J. J. K. O. Ruanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking", *Signal Processing* **66,** 303–317 (1998).
12. G. Healey, "Using color for geometry-insensitive segmentation", *J. Opt. Soc. Amer. A* **6**(6), 920–937 (1989).
13. J. Serra, *Image Analysis and Mathematical Morphology.* vol. 2. (Academic Press, London, 1988).
14. A. Rosenfeld and J. Pfaltz, "Distance functions in digital pictures", *Pattern Recognition* **1,** 33–61 (1968).
15. P. Bas and B. Macq, "A new video-object watermarking scheme robust to object manipulation", in *Proc. of ICIP'01*, (*IEEE*, Piscataway, NJ, 2001), pp. 526–529.
16. G. Wallace, "The JPEG still picture compression standard", *Comm. ACM* **34**(4), 31–44 (1991).
17. A. Bors and I. Pitas, "Image watermarking using dct domain constraints", in *Proc. ICIP*, vol. 3, (*IEEE* , Piscataway, NJ, 1996), pp. 231–234.