

Image watermarking based on color quantization process

Jean-Baptiste Thomas, Gael Chareyron and Alain Trémeau
Laboratoire LIGIV EA 3070 - Université Jean Monnet - Saint-Étienne, France

ABSTRACT

The purpose of this paper is to propose a color image watermarking scheme based on an image dependent color gamut sampling of the $L^*a^*b^*$ color space. The main motivation of this work is to control the reproduction of color images on different output devices in order to have the same color feeling, coupling intrinsic informations on the image gamut and output device calibration. This paper is focused firstly on the research of an optimal LUT (Look Up Table) which both circumscribes the color gamut of the studied image and samples the color distribution of this image. This LUT is next embedded in the image as a secret message. The principle of the watermarking scheme is to modify the pixel value of the host image without causing any change neither in image appearance nor on the shape of the image gamut.

Keywords: Image watermarking; Color quantization; Gamut Sampling.

1. INTRODUCTION

The idea of this paper is to focus on three steps of a color imaging system (see Fig. 1). Firstly the image is captured from an input device, e.g. a digital camera. Input data are by definition represented in a device dependent color space, e.g. RGB. The RGB data are device dependent, since another capturing device may produce other RGB values for the same scene. Therefore, a transform needs to be used to compensate for any colorimetric differences between original data and captured data, e.g. a gamma correction. Next, a second transform is used to watermark input data to protect captured image from illicit copy or to embed some characteristics data from input values into output values. Lastly, the image data are transformed and then reproduced on a display device, e.g. a CRT display. Therefore, a third transform needs to be used to compensate for any colorimetric differences between capturing and display device, e.g. a gamma correction. When using either another capturing device or another display device, a new color transform has to be developed. Such a problem may be solved by a color management process. The aim of color management is to control color images reproduction on different display devices. To reach this goal each device needs calibration to build transforms from a device-dependent color space to an independent one which offset and correct small color variations in order to have the same color feeling on any display device (see Fig. 2). Most of management methods are based on the use of parametric models or LUTs which are determined during the characterization step to calibrate output devices. Moreover, it's well known that each output device is limited by its color gamut. So they can only display a small amount of colors, in regards to the spectrum locus. In order to reduce the color differences between one device and another one, as well as to minimize the color differences between an image perceived by a Human Visual System and the same image reproduced on an output device, the use of a gamut mapping function is necessary. Such a function enables to map the color gamut of the input device (e.g. the capturing device) into the (often reduced) color gamut of the reproduction device¹.

The purpose of this paper is to build a color image watermarking scheme based on an image dependent color gamut sampling of the $L^*a^*b^*$ color space. The main motivation of this work is to control the reproduction of any color image on different devices to have the same color feeling on any output device. This paper focuses on the research of a specific (optimal) LUT which both circumscribes the color gamut of the studied image and samples the color distribution of this image. In order to optimize the model determination (i.e. to minimize the errors of interpolation), the LUT computation needs to be done in an perceptually uniform color space, such as CIELAB, because equal Euclidean distances in this color space represent equal perceived differences in appearance. Let us note that the use of CIELAB color space means that we have information on illuminant and primaries, e.g. that we know which lighting source and input device had been used to capture the picture and which output device is used to display the picture, that's why we are talking about intrinsic information.

The main contribution of this paper is to use a watermarking technique² which enables to embed into an input image the LUT computed specifically for this image. The principle is to modify the pixel value of the host image without causing

any change in appearance (i.e. to minimize changes in its colors distribution), and without causing any change on the shape of the color histogram. This LUT can be considered as a secret message that we embed in the input data. In a general way, the LUT extracted from the watermarked image is identical to those directly computed from the original image (i.e. before embedding the watermark). In other cases, the watermarked image may be altered by malicious attacks or simply by image processing (e.g. JPEG compression), then we can not retrieve from the watermarked image the LUT embedded. To strengthen the robustness of the embedding process the LUT is embedded several times into the image. The proposed scheme extends some image watermarking schemes based on vector quantization^{3, 4}. The most commonly used approach to provide image copyright protection is digital watermarking. Two categories of watermarking schemes can be used: spatial domain watermarking schemes and transform domain watermarking schemes. In spatial domain, the watermark is directly embedded into the host image. The main constraint of the spatial domain watermarking schemes is to minimize changes in appearance. The main advantage is that computational cost is lower than for the transform domain watermarking schemes. In previous work², we proposed to use the ΔE^*_{ab} color distance to minimize the amount of distortions between the original image and the watermarked image. This color distance was used to adjust the value of watermark pixels in function of the sensitivity of the visual human system.

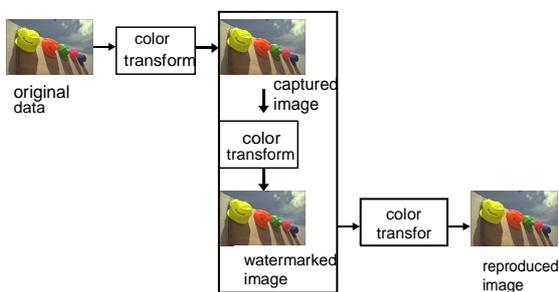


Fig. 1: The three steps of the color imaging system considered in this study.

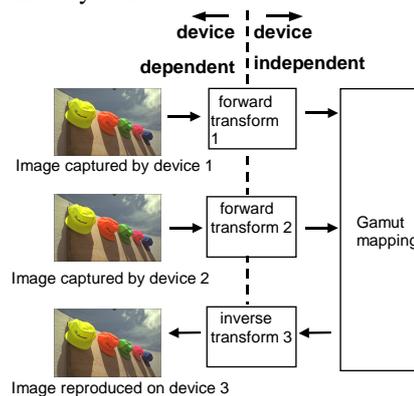


Fig. 2: Example of device dependent and independent color transforms¹.

In the following sections, we describe briefly the watermarking technique that we have developed. In section 2 we describe the method we developed to compute a LUT which both circumscribes the color gamut of the image studied and samples the color distribution of this image. This method is based on an image dependent color gamut sampling in $L^*a^*b^*$ color space. In section 3 we describe the watermarking method developed to embed into an image a LUT computed specifically for this image. Next we describe the extracting method developed to extract the LUT embedded in a watermarked image. Lastly, in section 4 we have a discussion on compromises we propose between precision, color histogram and color gamut preserving.

2. THE SAMPLING PROCESS USED

The color sampling technique that we propose is based on six steps. The aim is to select some representative colors (i.e. a color palette) specific to the image studied. This color palette consists of a limited number of representative colors, each having three dimensions for RGB-color images, which can be considered as a codeword in the codebook. The first step consists on a uniform sampling of $L^*a^*b^*$ color space based on a sphere packing technique (see Kepler's conjecture⁵) constraining each sphere's center to be at equal distance from its neighbors so that all neighbors of one center form a Johnson polyhedron⁶. So, each sample of the grid is surrounded by 12 equidistant samples. The second one consists on removing all samples which do not belong to the gamut of the $L^*a^*b^*$ color space. The third one consists on over-sampling the $L^*a^*b^*$ color space when the color distribution around a sample can not be correctly represented by only one sample. The main advantage of this process is that we can improve the discretization accuracy where it is necessary without increasing computational time too much. The fourth one consists on sub-sampling the $L^*a^*b^*$ color space when the color distribution around a sample is empty. The fifth one consists on the selection of the 2^n most representative color samples among those previously defined. The latest one consists on refinement of this set of colors in order to make it more representative of the color image gamut. An overview of this sampling scheme is given in Fig. 3.

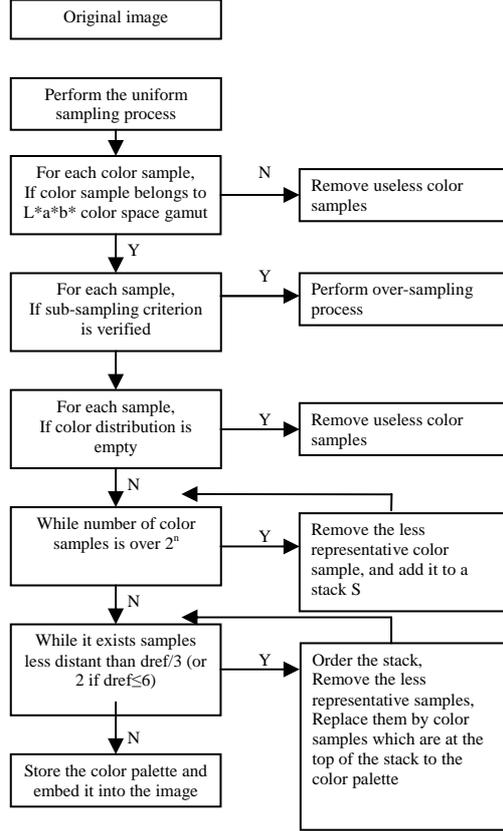


Fig. 3: The overview of the proposed sampling scheme

This sampling process is based on a split and merges strategy. The main criterion used in the proposed sampling scheme is the CIE ΔE^*_{ab} metric. To sample uniformly the $L^*a^*b^*$ color space we have used a hexagonal grid such as each sample of the grid is surrounded by 12 samples with a distant equal to d_{ref} (6 in a^*b^* plane, 3 in the upper L^* plane and 3 in the lower L^* plane). The proposed uniform sampling process is parameterized by the distance value d_{ref} between two samples. The smaller the d_{ref} value is the finer the sampling of $L^*a^*b^*$ color space is. By construction, the number of color samples increases inversely proportionally to the d_{ref} value (Fig. 4). In our process the number of samples computed has been restricted to the only samples belonging to the color gamut of the $L^*a^*b^*$ color space. Let us note that the smaller the d_{ref} distance is the higher the computing time is. Thus, with $d_{ref} = 9$ we have obtained $N = 1444$ samples in around 1 s, meanwhile with $d_{ref} = 6$ we obtained $N = 4835$ samples in around 4 s.

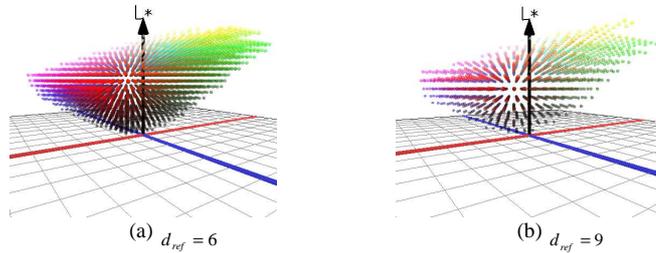


Fig. 4: Examples of uniform sampling of $L^*a^*b^*$ color space based on a hexagonal grid.

The sampling process thus defined enables to design a color palette overlapping entirely the color gamut of the L*a*b* color space. In order to reduce the size of this color palette to a given number of representative colors, we have developed a method based on color image quantization. Let be $\{\beta_j\}$ the set of color selected by the uniform sampling process. Considering that these colors are representative of the L*a*b* color space, we have computed for each color \mathcal{C}_i the closest representative color by $\arg \min_{j=0,\dots,K-1} \Delta E^*_{ab}(\mathcal{C}_i, \beta_j)$.

In the worst case, the maximal error that we can have (between the original color of a pixel and the closest color of a palette computed with our uniform sampling process) is by construction the isobarycentre of a regular pyramid defined by d_{ref} , e.g. $d_{ref}/\sqrt{2}$. One way to reduce the mean quantization error is to increase the number of samples around samples for which the mean error is too high rather than increasing globally the number of samples in decreasing uniformly the distance between them, as we do in over-sampling the studied sample neighborhood with $d_{ref}/3$. by using the Johnson polyhedron number 27. Initially each sample had 12 samples in its surrounding at a distance of d_{ref} , now it has 12 samples in its surrounding at a distance of $d_{ref}/3$. The maximum quantization error, after over-sampling, is around $d_{ref}/3$.

In order to reduce the average error of our uniform sampling process, we have fixed the following constraint. If $average_{\{\mathcal{C}_i/\beta_j\}} \Delta E^*_{ab}(\mathcal{C}_i, \beta_j) > 25\% d_{ref}/\sqrt{2}$ then sub-sample the hexagon centered on β_j . Where $\{\mathcal{C}_i/\beta_j\}$ represents the set of color of image I for which the closest color in color palette is the color β_j . This constraint determines which areas must to be sub-sampled or not according to the color image distribution is centered or not on palette's colors. Another interest of this over-sampling is that it enables us to reduce computational time and to break the regularity of the sampling scheme. Consequently to sample the L*a*b* color space with a precision of $d_{ref}=3$ for example, we have developed a process which firstly uniformly samples this color space with a precision of $d_{ref}=9$, next over-samples this color space with a precision of $d_{ref}/3=3$ and a sampling factor of 12 (for the worst case). Several studies had shown that under a precision of $d_{ref}=2$ the errors of quantization are imperceptible for observers and that with a precision of $d_{ref}=3$ the errors become perceptible but are not annoying². It seems therefore that a good compromise between precision and computing time can be obtained with $d_{ref}/3=3$ and a two steps process based on a uniform sampling followed by a sub-sampling process. In order to reduce the size of the color palette without increasing the quantization error, we have fixed the following constraint: If $card \{\mathcal{C}_i/\beta_j\}=0$ then remove β_j of color palette. This second constraint enables to remove all useless color samples of color palette.

Let $Density_1(P_j) = \frac{card \{\mathcal{C}_i/\beta_j\}}{\max_{k=0,\dots,K-1} card \{\mathcal{C}_i/\beta_k\}}$ be the probability of density of the sample β_j defined in function of the number of

colors of the original image distributed around each sample. Where: $card \{\mathcal{C}_i/\beta_j\}$ represents the number of colors of the original image closer of β_j than any other color sample, and $\max_{k=0,\dots,K-1} card \{\mathcal{C}_i/\beta_k\}$ represents the maximal number of colors of the original image linked to the same sample. The higher this density estimation is, the higher the studied color sample is representative of the original image in respect to this criterion. This density based on the tri-dimensional histogram, is the most commonly used criterion in quantization algorithm and influence gamut mapping algorithm¹.

Let $Density_2(P_j) = \frac{1 - average_{\{\mathcal{C}_i/\beta_j\}} \Delta E^*_{ab}(\mathcal{C}_i, \beta_j) / \Delta E^*_{max}\{\beta_j\}}{\max_{k=0,\dots,K-1} (1 - average_{\{\mathcal{C}_i/\beta_k\}} \Delta E^*_{ab}(\mathcal{C}_i, \beta_k) / \Delta E^*_{max}\{\beta_k\})}$ be the relative probability density of the sample β_j defined in

function of the density distribution of other samples. This density represents the relative dispersion of colors of the original image around the sample. Where : $1 - average_{\{\mathcal{C}_i/\beta_j\}} \Delta E^*_{ab}(\mathcal{C}_i, \beta_j) / \Delta E^*_{max}\{\beta_j\}$ represents the probability density of the sample β_j defined in function of the distance from β_j to colors; $\Delta E^*_{max}\{\beta_j\}$ represents the maximal distance between a color and a sample β_j ; and $\max_{k=0,\dots,K-1} (1 - average_{\{\mathcal{C}_i/\beta_k\}} \Delta E^*_{ab}(\mathcal{C}_i, \beta_k) / \Delta E^*_{max}\{\beta_k\})$ represents the maximal probability density of

samples. The higher this density estimation is, the higher the studied color sample is representative to the original image in regards to this criterion.

Let $Density_3(P_j) = \frac{1}{n} \sum_{i=0}^n SD(\mathcal{C}_i / \mathcal{P}_j)$ be the relative spatial density of sample \mathcal{P}_j . Considering that the color of a pixel

corresponding to a low frequency shift (homogeneous areas) is perceptually more important for an observer than those of a high frequency shift (as a "border" of a region in the image)⁷, we have defined this density criterion to weight the spatial frequency shifts of all colors linked to each sample. Considering that the probability that two consecutive pixels have the same color is low for a "true color image" (i.e. a 24 bit image), we have considered that all colors included in a sphere of radius 1 (in $L^*a^*b^*$ color space) have the same color. Where: $SD(\mathcal{C}_i) = \frac{1}{(2s+1)^2} \sum_{l=-s}^s \sum_{m=-s}^s f(p+l, q+m)$ represents the

spatial density of pixel of coordinates (p, q) and of color \mathcal{C}_i ; and $(2s+1, 2s+1)$ represents the size of the spatial mask used to analyse neighborhood pixels. In this study $s=3$, i.e. a mask $7*7$, f is a weighting function defined by $f(p+l, q+m) = \begin{cases} 1 & \text{if } \Delta E_{ab}^*(Y(p, q), Y(p+l, q+m)) \leq 1 \\ 0 & \text{else} \end{cases}$. The higher this density is, the higher the studied sample is representative to the original image in regards to this criterion.

We have combined these three density functions together to order samples from the most representative to the less representative. The idea is to reduce the number of samples to the most representative ones. To do that we have defined the following function : $Density(P_j) = Density_2(P_j) * Density_1(P_j) + (1 - Density_2(P_j)) * Density_3(P_j)$

Let us note that most of quantization algorithms used only the density 1, or for some of them the densities 1 and 3. In a general way these algorithms give more weight to density 1 than to density 3. In order to give more weight to density 1 than to density 3, we have used the third kind of combination. We have thus weighted the density 1 by the density 2 and weighted the density 3 by (1 - density 2). In that way, a sample is considered as representative if its occurrence's frequency is high (i.e. the density 1 is maximal) in the image and if the set of pixels it represents in the $L^*a^*b^*$ color space is compact (i.e. the density 2 is maximal). In such a case the density 3 is not determining. Inversely, if the occurrence's frequency of another sample is lower and that the set of pixels it represents in the $L^*a^*b^*$ color space is dispersed (i.e. heterogeneous), then the sample can be nevertheless considered as representative if it represents connected areas in the image (i.e. the density 3 is maximal). In such a case the density 3 plays a more important role. Moreover, This combination of density is image adaptive, so it avoid us to give some fixed and useless parameters within a linear combination.

Next, we have applied the following algorithm:

- While $N > 2^n$ remove the color sample $\mathcal{P}_k \in \{\mathcal{P}_j\}$ defined by: $\arg \min_{j=0, \dots, K-1} Density(P_j)$
- then $N = N - 1$;
- then redistribute image's colors which have lost their leader sample.

The aim is to reduce the size of color palette to a given number (e.g. $n = 8$) of representative colors selected among those previously defined by the sampling process. As example, see Fig. 5, after sampling with $d_{ref}/3=2$ ($d_{ref}/3=3$, respectively), we have obtained a color palette of 5717 colors (1759 colors) and a quantization error equal in average to 1.33 (1.88). Next, with a color palette reduced to 256 colors, we have obtained a quantization error equal to 2.23 (2.37). Let us note (compare Fig. 5 (b) and (d), next (c) and (e)) that the gamut of the color palette has been unfortunately reduced by selection of most representative colors in regards to the color gamut of all color samples. The problem is that nor the three criteria above defined; nor the constraint previously used to select the most representative colors; has been explicitly defined to preserve the shape and the size of the gamut of the palette. In order to compensate this drawback, we have introduced another criterion based on the Voronoi's partition. This criterion is used *a posteriori* to refine the color palette.

Let $Compacity(P_j) = \frac{V(S(\mathcal{P}_j, R_j)) / V(C(\mathcal{P}_j))}{\max_{k=0, \dots, K-1} (V(S(\mathcal{P}_k, R_k)) / V(C(\mathcal{P}_k)))}$ be the relative compactness of the sample \mathcal{P}_j . defined in function of the density distribution of colors of the original image represented by this sample. This compactness represents the

relative density of colors of the color palette around this sample. Where : $v(s(\mathcal{P}_j, R_j))$ represents the volume of the sphere $s(\mathcal{P}_j, R_j)$ centered on \mathcal{P}_j and of radius $R_j = \text{mean}_{\{\mathcal{P}_i/\mathcal{P}_j\}} \Delta E^*_{ab}(\mathcal{P}_i, \mathcal{P}_j)$, $v(c(\mathcal{P}_j))$ represents the volume of the Voronoi's cell $c(\mathcal{P}_j)$ centered on \mathcal{P}_j after partition of the L*a*b* color space from $\{\mathcal{P}_j\}_0$ and $\max_{k=0, \dots, K-1} (v(s(\mathcal{P}_k, R_k)) / v(c(\mathcal{P}_k)))$ represents the maximal compactness of color samples of $\{\mathcal{P}_j\}_0$. The higher this compactness estimation is, the less the weight of the studied color sample is important in the computation of the color gamut of the color palette of $\{\mathcal{P}_j\}_0$. In the assumption or two color samples of $\{\mathcal{P}_j\}_0$ are less distant than $d_{ref}/3$ (or less distant than 2 if $d_{ref} \leq 6$) then we can consider that one of these two samples may be withdrawn of the set $\{\mathcal{P}_j\}_0$ and replaced by another sample of $\{\mathcal{P}_j\}_0$ more relevant in regards to the color gamut of the image studied.

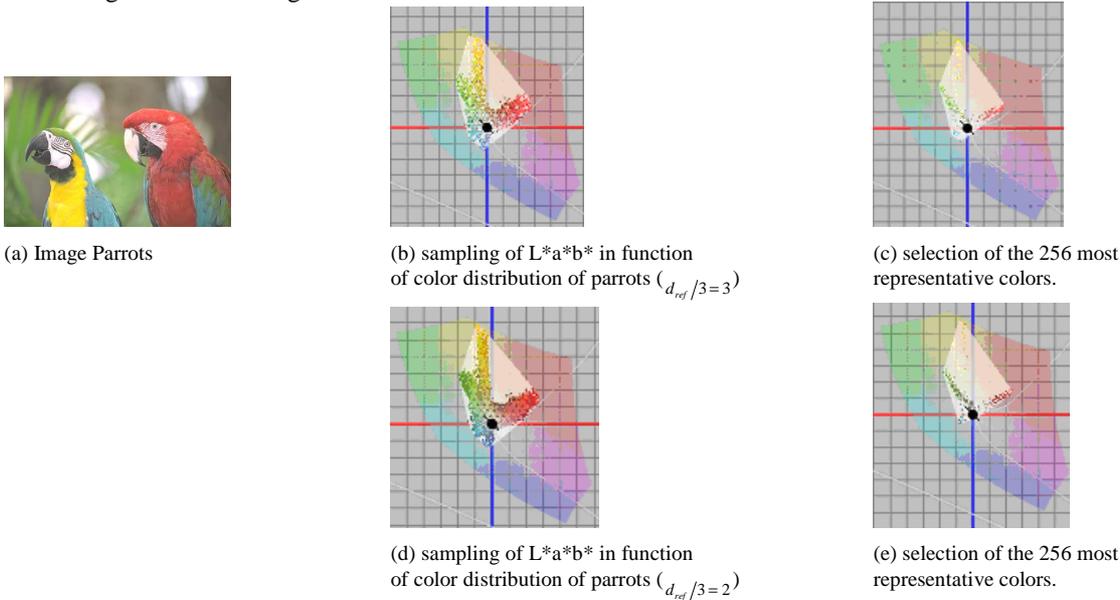


Fig. 5. Example of color palettes in L*a*b* color space.

Let $Eccentricity(\mathcal{P}_j) = 10 - \left\lfloor 10 \times \frac{d(\mathcal{P}_j, CH_j)}{d_{max}/2} \right\rfloor$ be the relative distance from the sample \mathcal{P}_j to the convex hull of the gamut of $\{\mathcal{P}_j\}$

discretized on 10 values. Where $d(\mathcal{P}_j, CH_j)$ represents the color distance of sample \mathcal{P}_j to the convex hull of the gamut of $\{\mathcal{P}_j\}$, $d_{max}(CH_j)$ represents the maximal diameter of the convex hull of the gamut of $\{\mathcal{P}_j\}$ and $\lfloor x \rfloor$ the integer value of x . The higher this eccentricity is, the higher the studied sample plays an important role in the computation of the shape and of the size of the color gamut of $\{\mathcal{P}_j\}$. Next, we have applied the following algorithm:

Step 1: order $\{\mathcal{P}_j\}$ in a pile according to the two following criteria: firstly from upper to lower eccentricity value, secondly from upper to lower density value in regards to $Density_2(\mathcal{P}_j) * Density_1(\mathcal{P}_j)$.

Step 2: if two samples \mathcal{P}_1 and \mathcal{P}_2 of $\{\mathcal{P}_j\}_0$ are less distant than $d_{ref}/3$ (or less distant than 2 if $d_{ref} \leq 6$) then: if $Compacity(\mathcal{P}_1) > Compacity(\mathcal{P}_2)$ then withdraw \mathcal{P}_2 of $\{\mathcal{P}_j\}_0$ and replace it by the sample at the top in the pile. Re-iterate 2).

With such a process we preserve better the shape and the size of the gamut of the palette in regards to the shape and the size of the gamut of the studied image. Another consequence: the maximal error decreases but with depends of the average error. Let be $\{\mathcal{P}_j\}_1$ the new color palette defined by this process. As example, let us consider the Fig. 5 (a)

sampled with $d_{ref} = 9$ and 256 samples, then we note that, after refinement of the color palette by the compactness, the gamut of the palette has increased of 12 points (60% / 48%) meanwhile the average error has increased from 2.37 to 3.22. Let us now consider a sampling with $d_{ref} = 6$, after refinement of the color palette by the compactness, the gamut of the palette has increased of 32 points (75% / 43%) meanwhile the average error has increased from 2.23 to 2.40 (see Fig. 6). These results are better than the previous ones in regards of precision but are worse in regards to computing time. That is, in regards to gamut conserving, results are better. Let us now consider a sampling with $d_{ref} = 9$ and 512 samples, after refinement of the color palette, the gamut of the palette has increased of 41 points (100% / 59%) meanwhile the average error has increased from 2.09 to 4.16. These results are not better than the first ones with $d_{ref} = 9$ in regards of precision and are worse in computing time but they are better in regards to color gamut. Let us lastly consider a sampling with $d_{ref} = 6$ and 512 samples, after refinement of the color palette, the gamut of the palette has increased of 1 point (75% / 74%) meanwhile the average error has increased from 1.78 to 2.26. These latter results are slightly better than the second ones in regards of precision and are worse in computing time; but are really better in regards to color gamut. Consequently, the increase of number of samples enables to better preserve the gamut of the image studied especially when d_{ref} is low. This increase is done with depend of computing time but has no influence on precision. Likewise, results that we have obtained have showed that the better value for d_{ref} is image dependent.

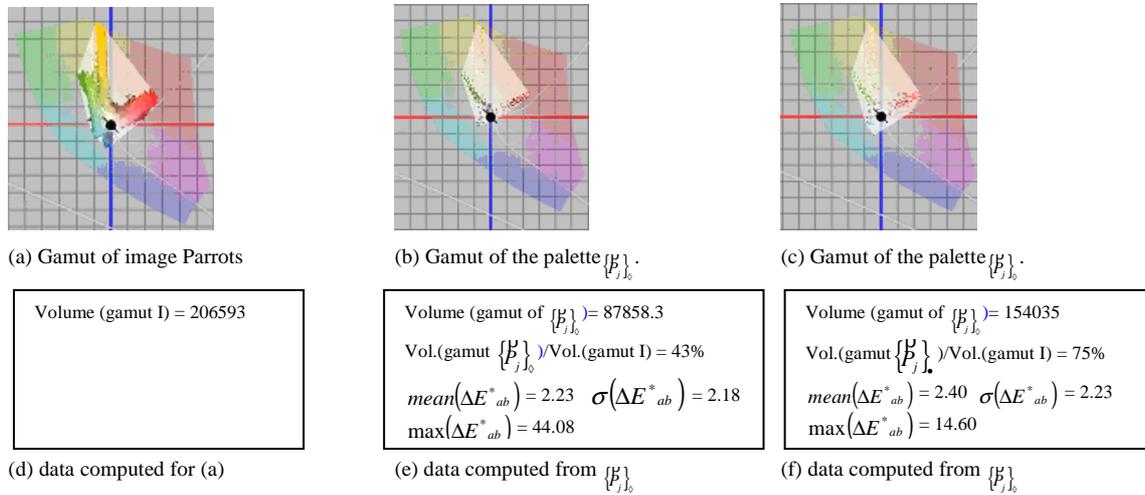


Fig. 6. Color palette in L*a*b* color space of image Parrots sampled with 256 color before and after refinement of color palette. $\{P_j\}_6$ has been computed with $d_{ref} = 6$. The volume of each color set has been computed in the L*a*b* color space from the convex hull corresponding to the set studied.

3. THE EMBEDDING SCHEME USED

The LUT that we have computed is now used as a secret message. In this section, we describe how to embed this LUT into an image and how to strengthen the robustness of the embedding process in embedding several times the same LUT into the image. An overview of the proposed watermark embedding scheme is given in Fig. 8 (a).

Let us suppose that each color of the original image is coded in RGB with 24 bits then to store a color palette of 2^n colors coded in RGB we need $(2^n \times 24)$ bits. If we want to store this color palette with sets of 2^a bits (e.g. $a = 8$ for an octet, or $a = 2$), then we need $b \times 2^a$ bits, where $b = (2^n \times 24) / 2^a$. Let us suppose now that we want to embed all pixels of an image with 2^a bits per pixel from the color palette under study, then we can embed t times this color palette, where $t = \frac{x \times y}{(2^n \times 24) / 2^a}$ and $x \times y$ is the size of the image. For example, in a color image of size 256×256 , we can embed about 10 times a color palette of 256 colors with 1 bit per pixel and about 42 times a color palette with 4 bits per pixel. With a color palette of 1024 colors we can embed at most 2 times such a color palette with 1 bit per pixel and about 10 times this

color palette with 4 bits per pixel. The more we embed bits per pixel, the more we can store several time the color palette in the image. Inversely, the more the size of the color palette is, the less we can store several time this LUT in the image.

For security considerations, the colors of the palette are ordered differently for each use of the LUT. A pseudorandom number generator (PRGN) is used to rearrange the colors of palette. Thus, the sequence of bits used to embed the host image differs for each use of the LUT. All pixels of the host image are embedded. For security considerations, a pseudorandom number generator (PRGN) is used to determine the positions of the pixels that are embedded by the watermark bits computed from the LUT previously defined.

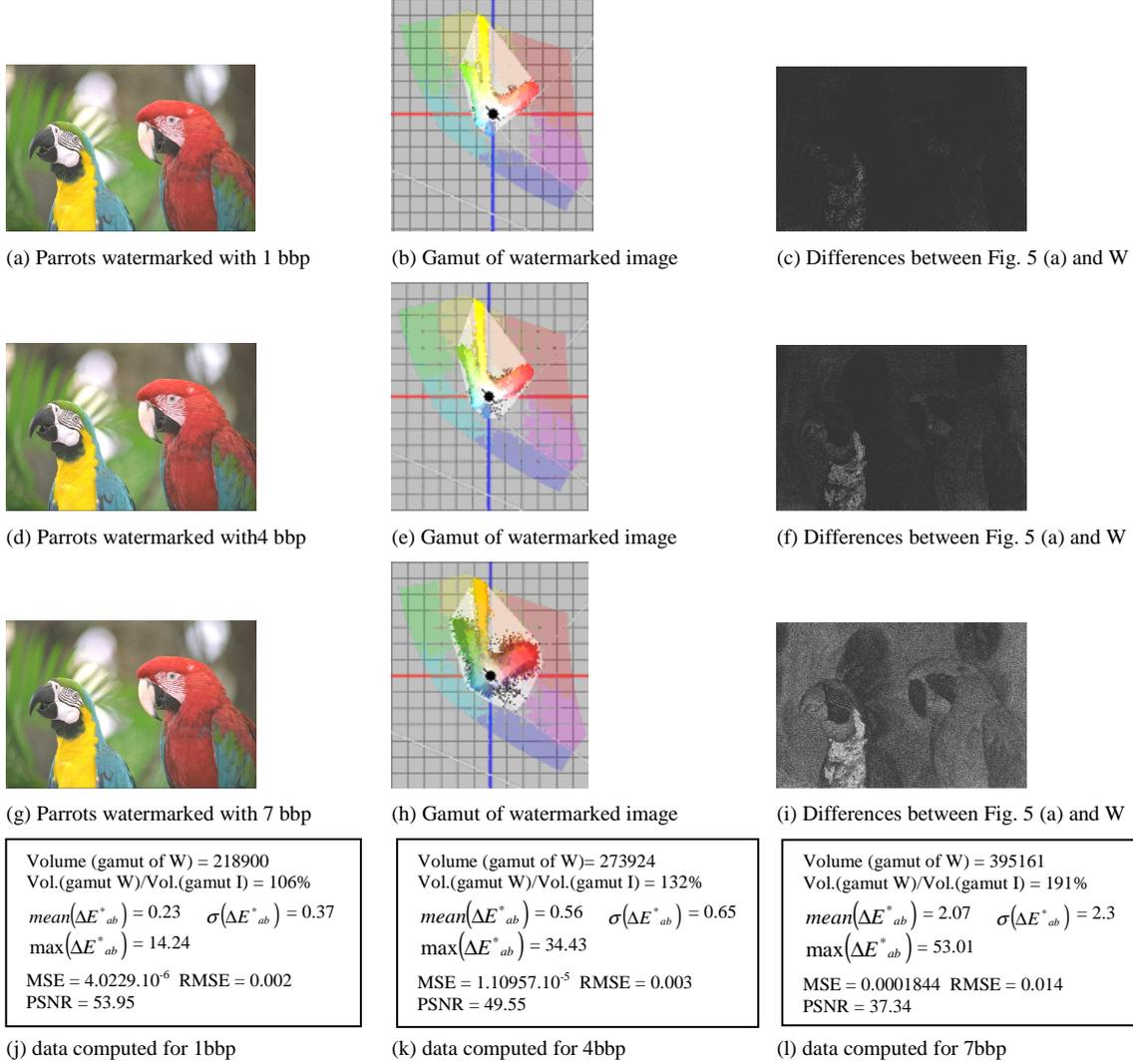


Fig. 7. Color palette in L*a*b* color space of image Parrots watermarked with 1 bbb, 4 bbb and 7 bbb respectively. $\{\beta_j\}_6$ has been computed for 256 color samples, with $d_{ref} = 9$, without refinement of color palette. Color differences between the original image (I) and the watermarked one (W) had been computed pixel by pixel according a logarithmic.

Let be p_j the position of the j^{th} pixel of the host image; \mathcal{C}_j the color of pixel p_j in the host image; a the size of the secret message m that we want to embed into the pixel of position p_j ($b \times 2^a$ represents the size of the LUT); \mathcal{W}_j the color of pixel p_j in the watermarked image. For each pixel p_j , it is possible to find a color \mathcal{W}_j such as: $[\Delta E_{ab}(\mathcal{C}_j, \mathcal{W}_j) \times c] \bmod 2^a = m$

(1) where $\Delta E_{ab}(\mathcal{C}_j, \mathcal{W}_j)$ represents the color distance between \mathcal{C}_j and \mathcal{W}_j , and $\lfloor \cdot \rfloor$ is the integer function and c is a scaling factor constant. The watermarking scheme proposed consists to watermark each pixel P_{j+1} in function of the color of the pixel P_j , such as each pixel of the host image is watermarked only one time. Let be W the watermarked image. The embedding process consists of: *Step 1*: find the color \mathcal{W}_j which satisfies the criterion (1) and which minimizes the distance $\Delta E_{ab}(\mathcal{C}_{j+1}, \mathcal{W}_j)$ in order to minimize the changes of color due to the watermarking scheme proposed. That is to say, find the color \mathcal{W}_j which satisfies: $\arg \min_{\mathcal{W}_j \in \lfloor \Delta E_{ab}(\mathcal{C}_j, \mathcal{W}_j) \rfloor \bmod 2^m} \Delta E_{ab}(\mathcal{C}_{j+1}, \mathcal{W}_j)$. *Step 2*: change the color \mathcal{C}_{j+1} of pixel P_{j+1} by \mathcal{W}_j in the host image.

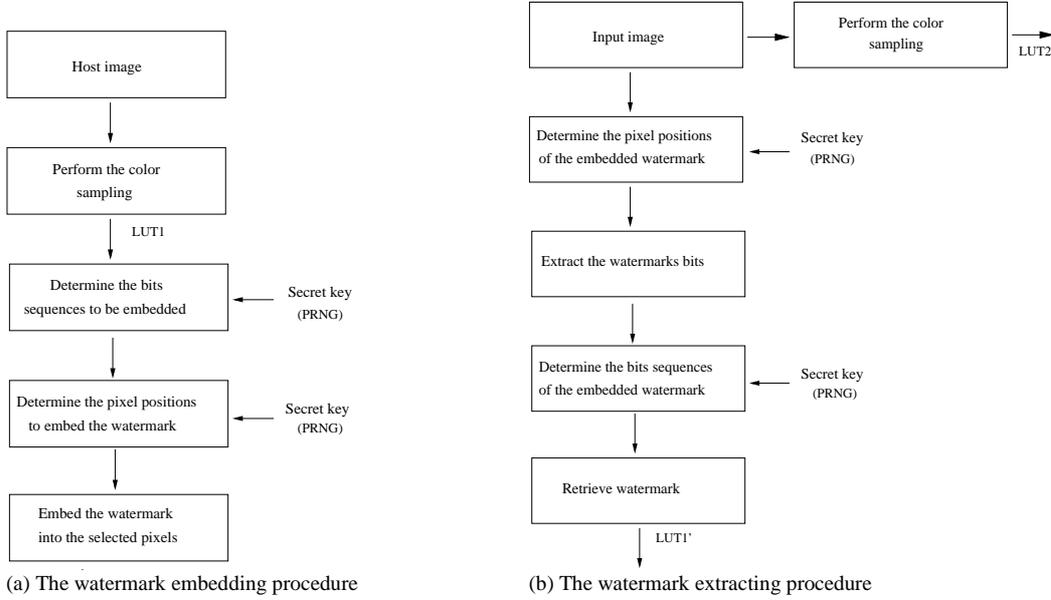


Fig. 8. The overview of the proposed watermarking scheme.

As we can see on Fig. 7, under 4 bbp the original and the watermarked images can be considered as perceptibly identical. Likewise, their histogram can be considered as similar. Two examples of color histogram are given in Fig. 9 and 10. Note that in order to compare color histograms we have used the *earth mover's distance*⁸. Inversely, the gamut of the watermarked image is by construction bigger than those of the original because the watermarking process has generated new colors which do not belong to the original image². Nevertheless, let us note that we can find again the gamut of the original image from the watermarked image in extracting the LUT of the original image embedded in the watermarked one. Above 6 bbp differences become more noticeable, especially in homogeneous areas.

The structure of the watermark extracting procedure is defined by the structure of the watermark embedding procedure. An overview of the proposed watermark extracting procedure is given in Fig. 8 (b). To extract the watermark, the user needs to know: the secret key used to determine the positions of pixels embedded, and the secret key used to determine the sequences of bits embedded. The watermark detection is blind, i.e., the original image is not needed. The extracting process consists of: *Step 1*: use the PRNG to determine the positions of pixels that contain the watermark bits. *Step 2*: extract the watermarks bits (i.e. the message m) embedded in each pixel from the criterion (1). *Step 3*: use the PRNG to determine the sequences of bits used to embed all copies of the color palette. *Step 4*: extract each copy of the color palette (the size of each sequence of embedding bits is equal to 2^n), from all these copies compute the color palette used to watermark the image.

In a general way, the LUT extracted from the watermarked image (LUT1') is identical to those of the original image (LUT1). In other cases, the watermarked image may be altered by malicious attacks or simply by image processing (e.g. JPEG compression), or by color management (e.g. gamut mapping), then we can not retrieve from the watermarked image the LUT embedded. Let be LUT2 the color palette of the watermarked image directly computed from the image as for the original image. As we can see on Fig. 9 and 10, the LUT extracted from the watermarked image is quite similar to

those of the original image. More exactly, the color histogram from the quantization of watermarked image with LUT2 is quite similar to the color histogram from the quantization of the original image with LUT1 even if the size of the gamut is bigger in the former case. Likewise, in a general way, the LUT of the watermarked image (LUT2) is quite similar to those computed after extracting the watermark (LUT1'). In case of image modification (e.g. contrast enhancement), then either only some copies of the color palette can be extracted from the image considered or none copy of the color palette can be extracted among all sequences of embedding bits. In such a case, we consider that the image studied is not in conformity with the original one in regards to the watermarking process applied.

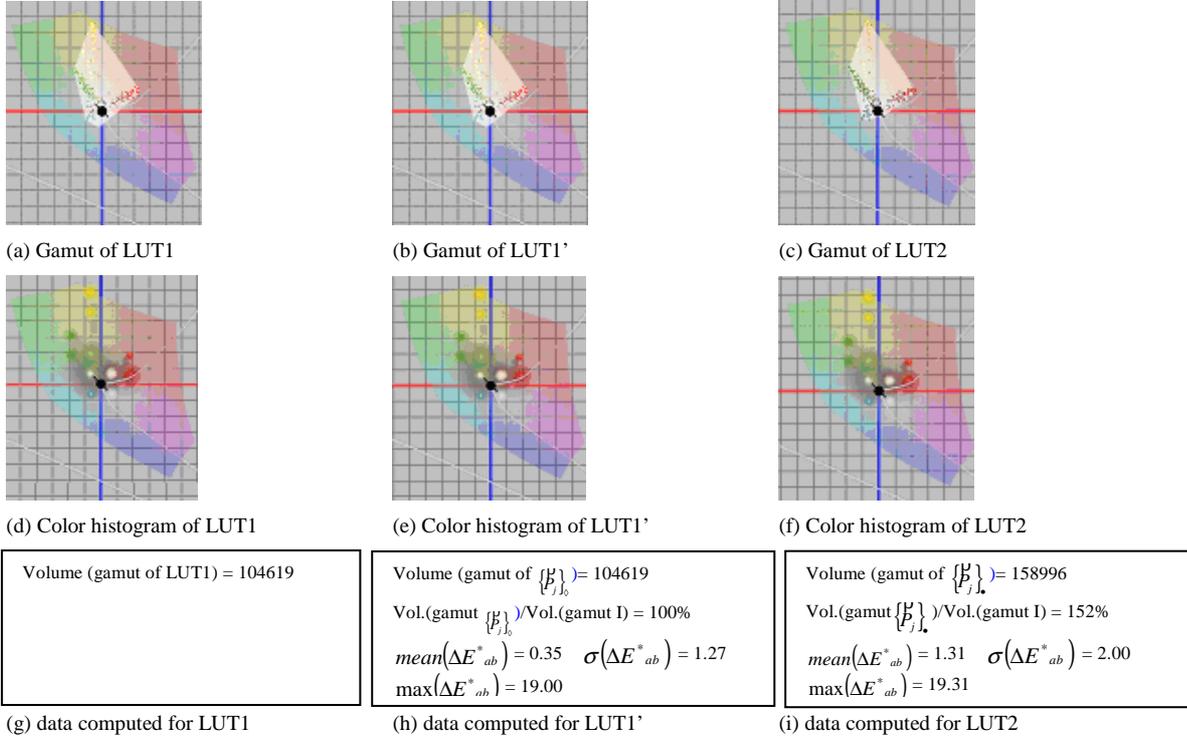


Fig. 9. Color palette in L*a*b* color space of image Parrots sampled with 256 colors and $d_{ref} = 9$ (LUT1), of the watermark extracted of image watermarked with 1 bpb (LUT1') and of watermarked image sampled with 256 colors and $d_{ref} = 9$ without extraction of the watermark (LUT2). For color histograms, we have associated at each color a sphere for which the size is proportional to its occurrence's frequency.

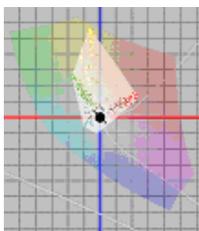
4. DISCUSSION AND CONCLUSIONS

The purpose of this paper was to build a color image watermarking scheme based on an image dependent color gamut sampling of the L*a*b* color space. This paper focuses on the research of a specific (optimal) LUT which both circumscribes the color gamut of the studied image and samples the color distribution of this image. This LUT will be used to find a model which will allow to estimate, with a strong reliability, the color distribution of the input image. In order to optimize the model determination, i.e. to minimize the errors of interpolation, the LUT computation was done in the CIELAB color space.

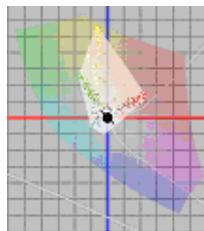
Several compromises had been considered in this study.

- The first compromise concerns the distance value. We have shown that in a general way $d_{ref}/3=3$ or $d_{ref}/3=2$ is a good compromise between precision and computing, and that the choice of d_{ref} value is also image dependent.

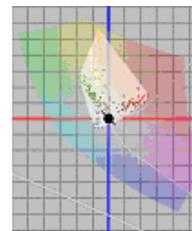
- The second compromise concerns how color samples are defined. Rather than using a uniform sampling, we have developed a two-steps process, based firstly on a uniform sampling next on an image dependent adaptive over-sampling, in order to break the regularity of the sampling scheme.
- The third compromise concerns the number of representative samples. We have shown that the increase of number of samples enables to better preserve the gamut of the image studied especially when d_{ref} is low. This increase is done with depend of computing time but has no influence on precision of sampling. Inversely, it has influence on precision of interpolation between samples. Let's specify that the computationnal time of the first selection of samples, i.e. before refinement, is nearly independent of the number of required samples (under the constraint that with d_{ref} value we have a sufficient number of samples). In other case, the number of required samples influences the computational time of the refinement. The more we need samples, the more computational time increases (moreover it is influenced by the d_{ref} value).
- The fourth compromise concerns the criteria used to characterize a representative color sample. Representative colors have been defined in function of three density parameters which have been combined together. We have considered that a sample was representative firstly if its occurrence's frequency is high in the image and if the set of pixels it represents in the L*a*b* color space is compact. Likewise, we have considered that a sample was representative, even if its occurrence's frequency is low and that the set of pixels it represents in the L*a*b* color space is dispersed (i.e. heterogeneous), when it represents connected areas in the image.
- The latest compromise concerns the color gamut preserving in regards to the color histogram preserving. Rather than preserving the gamut, we have preferred preserve the shape of the color histogram because this latter influences greater the result of gamut mapping processes. We have done the assumption that when two color samples are too closed we can consider that one of these two samples may be withdrawn and replaced by another sample more relevant in regards to the color gamut of the image studied. We have therefore defined two criteria. The first one based on Voronoi's partition characterizes the compactness of the color gamut. The second based on the convex hull computation enables to preserve the shape and the size of the gamut. We have shown that the use of these two criteria enable to reduce the maximal error but with depends of the average error. We have therefore weighted their influence in giving more weight to the three criteria described above.



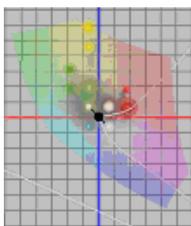
(a) Gamut of LUT1



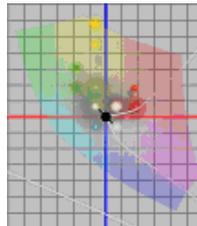
(b) Gamut of LUT1'



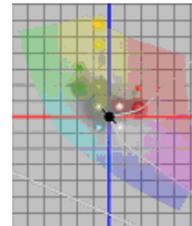
(c) Gamut of LUT2



(d) Color histogram of LUT1



(e) Color histogram of LUT1'



(f) Color histogram of LUT2

Volume (gamut of LUT1) = 104619

(g) data computed for LUT1

Volume (gamut of LUT1') = 104619
 Vol.(gamut LUT1')/Vol.(gamut LUT1)=100%
 $mean(\Delta E_{ab}^*) = 0.72$ $\sigma(\Delta E_{ab}^*) = 1.76$
 $max(\Delta E_{ab}^*) = 22.06$

(h) data computed for LUT1'

Volume (gamut of LUT2) = 158996
 Vol.(gamut LUT2)/Vol.(gamut LUT1)=85%
 $mean(\Delta E_{ab}^*) = 3.29$ $\sigma(\Delta E_{ab}^*) = 2.29$
 $max(\Delta E_{ab}^*) = 37.23$

(i) data computed for LUT2

Fig. 10. Color palette in L*a*b* color space of image Parrots sampled with 256 colors and $d_{ref} = 9$ (LUT1), of the watermark extracted of image watermarked with 4 bbp (LUT1') and of watermarked image sampled with 256 colors and $d_{ref} = 9$ without extraction of the watermark (LUT2). For color histograms, we have associated at each color a sphere for which the size is proportional to its occurrence's frequency.

The watermarking technique used consists to embed into the host image the LUT previously computed without causing any change in appearance and without causing any change on the shape of the color histogram. To strengthen the robustness of the embedding process the LUT is embedded several times into the image. In a general way, the LUT extracted from the watermarked image is identical to those directly computed from the original image (i.e. before embedding the watermark). In other cases, the watermarked image may be altered by malicious attacks or simply by image processing (e.g. JPEG compression), then we can not retrieve from the watermarked image the LUT embedded. This enables us to prove that the image has been modified. Likewise, in a general way, the LUT of the watermarked image (directly computed from the image as for the original image) is quite similar to those of the original image. More exactly, the color histogram of the latter LUT is quite similar to the former one even if the size of the gamut is bigger in the former case.

REFERENCES

1. J. Morovic and M.R. Luo, The Fundamentals of Gamut Mapping: A Survey, *Journal of Imaging Science and Technology*, Vol. 45, N° 3, (2001) 283–290.
2. G. Chareyron, D. Coltuc and A. Trémeau, Watermarking and authentication of color images based on segmentation of the xyY color space, *Journal of Imaging Science and Technology*, Sep. 2006.
3. A. Makur, S. Sethu Selvi, Variable dimension vector quantization based image watermarking, *Signal Processing*, Vol. 81, N°4 (2001) 889-893.
4. P. Tsai, Y.C. Hu and C. C. Chang, A color image watermarking scheme based on color quantization, *Signal Processing*, vol. 84 (2004) 95-106.
5. J. Kepler, monograph, *The Six-Cornered Snowflake*, 1611.
6. N.W. Johnson, *Convex Solids with Regular Faces*, *Canadian Journal of Mathematics*, 18 (1966) 169-200.
7. S. Voloshynovskiy, A. Herrigel, N. Baumgärtner, and T. Pun. A stochastic approach to content adaptive digital image watermarking. *Lecture Notes in Computer Science*, Vol. 1768 (1999) 212–236.
8. J. Morovic, J. Shaw and P.-L. Sun: A fast, non-iterative and exact histogram matching algorithm. *Pattern Recognition Letters*, Vol. 23 (2002) 127-135.
9. P.L. Sun and J. Morovic, 3D Histograms in Colour Image Reproduction. *Proceedings of SPIE*, Vol. 4663 (2002) 51-62.