Adaptive color quantization using the "baker's transformation"

Christophe Montagne[1], Sylvie Lelandais[1], André Smolarz[2],
Philippe Cornu[2], Mohamed Chaker Larabi[3], Christine Fernandez-Maloigne[3]

IBISC Laboratory
CNRS FRE 2873
University of Evry-Val d'Essonne
40 rue du Pelvoux
91020 Evry Cedex
cmontagne@iup.univ-evry.fr

ISTIT Laboratory
CNRS FRE 2732
University of Technology of Troyes
12 rue Marie Curie BP 2060
10010 Troyes Cedex
andre.smolarz@utt.fr

IRCOM-SIC
CNRS FRE 2731
University of Poitiers
BP 30179
86962 Futuroscope Cedex
chaker.larabi@sic.univ-poitiers.fr

Abstract: *In this article we propose an original technique to reduce the number of colors contained in an image. This method uses the "Baker's Transformation", which obtains a statistically suitable mixture of the pixels of the image. From this mixture, we can extract several samples, which present the same characteristics as the initial image. The concept we imagined is to consider these samples as potential pallets of colors. These pallets make it possible to do an adaptive quantization of the effective number of colors. We consider, and we put in competition, three methods to obtain a single pallet. Firstly, we present the "Baker's Transformation". Secondly, we present methods to have a single pallet. Third part is dedicated to results illustrating the good visual quality reached by the quantized images. Finally, we present a comparison between our method and three classical methods of quantization.*

Keywords : *Color image processing - Color quantization - Baker's Transformation - Quasi-Mixing Transformation - Method comparison.*

Introduction

The use of colored numerical images is very current for professional or ludic applications. Most of the time, these images are defined in a color space with three components as RGB cubic space (Red Green Blue - 24 bits per pixel) where the value of each channel is coded on one byte (1 byte = 8 bits $\Leftrightarrow 2^8 = 256$ available levels). In this case, to code an image, the number of available colors is higher than 16 million ($2^{24} = 16,777,216$). This large number implies that, generally, none of the image pixels has a color identical to the other pixels. Here, we deal with real images both outside and indoor. For an image of size 256×256, it is not surprising to find $2^{16} = 65,536$ different colors. The 3D color histogram of the figure 1 (see figure 2) illustrates this phenomenon: colors form a scattered "cloud", where each point corresponds to one or two pixels. The will to reproduce the human visual system explains this situation but it requires computer files to be larger than needed. An image of size 256×256, stored in a "bitmap" file (*.bmp), weighs 192,662 bytes, if the coding of the colors

is 24 bpp. Whereas, if this coding is 8 bpp, the size of the file decreases to $66,614$ bytes, which is three times less. Moreover, many image processing are difficult because of 24 bpp coding. We think for example of the use of a color histogram in image indexation: with a rate close to one color for one pixel, the histogram is almost flat. Computing times are also multiplied in the case of a 24 bpp image. Psychovisual research proves that the human eye does not perceive a great disparity of colors: it appears useless, therefore, to have too many colors to code an image.

Figure 1: Original image (256×256): "mandrill".

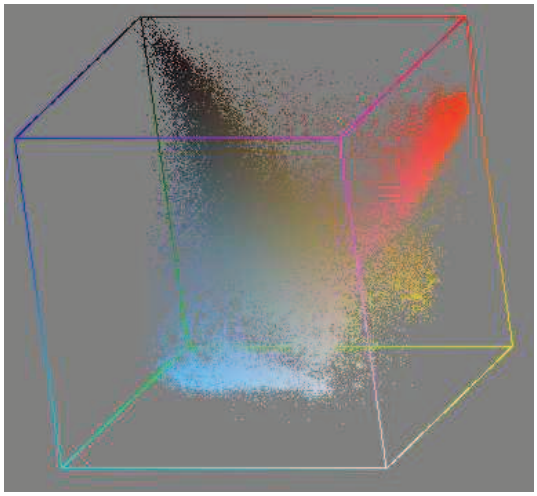


Figure 2: 3D histogram of the image in figure 1.

For these reasons, it seems useful and interesting to define methods to significantly reduce the number of the colors in an image, while preserving a constant visual quality. Color quantization techniques are one solution. The objective is to replace a color in the image by another selected in a preset pallet, which contains a limited number of colors. The new color will have to be the nearest possible to the original color according to a preset criterion [1] [2]. In the scientific literature, we find articles that treat color quantization, and which propose various adaptive methods to select the best pallet [3] [4] [5] [6]. The common objective of all of these methods is to preserve the initial appearance of the image. In this article, we propose a new adaptive method of color quantization, which is based on the use of the "Baker's Transformation" [7] [8]. In the first part of this document, we present the principle of this transformation, which is a result of ergodic processes analysis. In the second part, we explain how it is possible to use this transformation to quantize, in an adaptive way, the number of colors in the image. Several methods are possible and we present them with visual results and quantitative measurements. In the third part, we present results obtained from several images and we discuss the interest of the considered approaches. We present also results from the point of view of L*a*b* space. In the fourth part, we present comparison results between our method and three classical methods of

quantization. The comparison is twice with objective and subjective tools. Finally, we conclude by evoking possible continuations of this work.

1 The Baker's Transformation

It is possible to associate a texture to any image by means of a peculiar transformation of the pixels. As this transformation is a permutation, it is one-to-one and thus invertible. The computed texture is locally very homogeneous, even if the original image were not. Furthermore, as this transformation acts only on the position of the pixels, we can use this transformation with monochromic or with color images. In the case of color images (in RGB space for example), we just have to apply the permutation to each of the three channels.

1.1 Mixing dynamical systems

The transformation we use for our purpose is the so-called "Baker's Transformation" – or BT in the sequel. This function originates in the theory of mixing dynamical systems, theory which has been introduced by Gibbs in the study of trajectories in the phase space of a mechanical system. We give now some basic definitions. Let $M = [0,1] \times [0,1]$ be the unit square, and let μ be a measure defined on M such that $\mu(M) = 1$. Now let $f : M \to M$ be a one-to-one transformation, which is measure preserving, that is such that $\mu(f(A)) = \mu(A)$ for all measurable A. Following Arnold and Avez [9], we say that (M, μ, f) is an abstract dynamical system. When studying such systems, we are interested – among others – by the behavior of points of M or of subsets of M when we apply $f^n,\ n \in \mathbb{Z}$.
By definition (see [9]), an abstract dynamical system is said to be mixing if:

$$\lim_{n \to +\infty} \mu(f^n(A) \cap B) = \mu(A) \cdot \mu(B)\ \ \forall A, B \in M,\ A, B\ \textit{measurable}$$

These authors give a more "intuitive" definition (see [9] pp. 18-19): «Let M be a glass with 90% of Martini and 10% of gin, and let S be a spoon. Then mix the cocktail by turning the spoon in the glass (each turn of the spoon being an iteration of f). Physically, after a while, one can hope that any part of the cocktail contains 90% of Martini and 10% of gin.»
The interesting fact here is that any (measurable) part of M becomes (asymptotically) a small-scale model of M.
Let's now examine the finite discrete case, which is the one of digital images. In this case, a one-to-one

transformation is a permutation and is therefore periodic. All works with a finite number of steps (finite but even large). After we have mixed the cocktail, we can separate with a finite number of steps the (discrete) gin from the (discrete) martini. But in this case, unlike for the continuous case, we can't expect that all part of M is a small-scale model of M. Indeed, we know from [10] that in the discrete finite case it is not possible for an abstract dynamical system to be mixing.

1.2 Quasi-Mixing Transformations and the BT

Nevertheless, for some mixing transformations, when restricting its action to a finite space, despite the fact that the transformation is no longer mixing, it remains a mixing-like property. In this case, we say that the transformation is a Quasi-Mixing Transformation (or QMT for short - see [8] for details). The Baker's Transformation is such a transformation. In the continuous case, following [9], the BT is described as follows. *To compute f on the unit square M (i.e. $0 \leq x, y \leq 1$):*

- *M is contracted by a factor 2 in the y direction and expanded by a factor 2 in the x direction (see figure 3);*
- *cut vertically the rectangle in the middle and glue the right half on top of the left half to obtain a unit square (see figure 4).*

Figure 3: First step of the BT initial iteration (128×512).

Figure 4: Third step of the BT initial iteration (256×256).

When working with images, the first step is obtained by interlacing lines two by two - we thus get an image twice larger than the original one, with half height (the transformation is one-to-one). Then, we cut and paste the image in order to achieve the second step.

Thus, the only condition is that the image must have an even number of lines. As mentioned above, the BT is periodic but its period is a rather irregular function of the size of the image. Nevertheless, in the case of a $2^n \times 2^n$ image, we retrieve the original image after $4n$ iterations of the BT. For example, for a 256×256 image, i.e. for $n = 8$, the period is 32. On the other hand, for a 258×258

image, the period is 1.1698×10^{34}, and for a 300×300 image, it is 2.0492×10^{19}. That is the reason why we are mainly interested with $2^n \times 2^n$ images in the sequel.

Let's examine the case of a 256×256 image (that is with $n = 8$). We observe that for each iteration from 1 to 8, the pixels get more and more mixed, and the image appears to be the most mixed for iteration 8: we have a microscopic texture (see figure 5). This observation is justified in [11]. Then, for iterations from 9 to 16, the image is unmixed step by step so that we retrieve the original image inverted (bottom at the top). Then it works in the same way for iterations from 17 to 32, and we retrieve the original image in the right position.

In theory, if we apply the inverse BT on a sample extracted from the mixed image, then we obtain an image of small size which is very close to the original image (see figure 6). In our application, we are not interested by the inverse transformation. We are interested in obtaining a sample, which is a good representation of the original image. Hence, for non-$2^n \times 2^n$ images and for rectangular images, we approach the BT transformation by a homogeneous sub-sampling of the original image.

1.3 The local reconstruction property of the QMT

The property we are going to use in the sequel for the quantification of the image colors comes from mixing dynamical system properties (continuous case). When one examines the image of figure 5, one observes a kind of double periodicity (along the x and y axes). As mentioned above, it is not possible in the finite case to take any subset of the space and to find in it the right proportion of each element of the image. Nevertheless, this property is preserved if we select the part of the texture with respect to this double periodicity. More precisely, if we have a $2^n \times 2^n$ image, let's consider a partition of the image with $2^p \times 2^p$ ($p < n$) blocks. Thus, for the image of figure 5, we can have a partition in four blocks of 128×128 pixels each. If now we apply the inverse transformation TB^{-1} to the block shown on figure 5 with the right number of iterations – that is, using the fact that $128 = 2^7$ – we get the image shown in figure 6, which is a 128×128 version of the original one. If now we choose $p = 6$, we have a 16 block partition with 64×64 pixels each block. If now we apply the inverse transformation TB^{-1} with the right number of iterations – that is, using the fact that $64 = 2^6$, with 6 iterations, we retrieve a 64×64 version of the original image, and so on. If we don't follow this rule when extracting a block from the image, we generally don't get a faithful reconstruction.

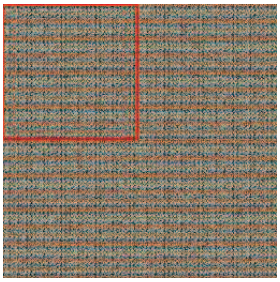


Figure 5: Suitably mixed image (after $N = 8$ iterations) (256×256) (with extracted window highlighted in red).

Figure 6: Window of size 128×128 extracted from figure 5 and result obtained after $p = 7$ BT^{-1}.

1.4 Fast implementation of the BT

The creation of the color pallet is based on the mixture provided by the BT. This transformation is iterative: it is a repetition of the same treatment. Consequently, it is a long time treatment. But, for a given image size, a pixel will be always affected at the same position in the mixed image. It is thus possible to create a "LUT" (look up table), which defines this position. By this way, we obtain directly the mixture from the original image. The "LUT" is in 3D and has the following dimensions: $2^N \times 2^N \times 2$. For each pixel of an image $2^N \times 2^N$, the "LUT" provides its new coordinates i and j. Moreover, a single "LUT" of given size is enough to treat all the images of same dimensions.

2 Color quantization

2.1 Definition of the pallets

As we have shown in the previous paragraph, a window of size $2^p \times 2^p$ preserves the characteristics of the original image: forms, textures and colors. On the basis of this observation, we consider this window and its pixels as a set of colors, which are representative of the complete color range of the image. According to the dimensions of the window, we dispose of a color pallet, which is adapted to the image and is obtained in a simple and fast way. An obvious difficulty of our proposal is the manner of choosing the window to be extracted. With an image of size $2^N \times 2^N$ pixels, it is possible to obtain $K = 2^{2(N-p)}$ pallets containing a maximum of 2^{2p} colors. For our experimental protocol, we limit the number of colors in the pallet to 256, 64, 16 or 4 colors. These pallets correspond to extracted windows of size $2^4 \times 2^4$, $2^3 \times 2^3$, $2^2 \times 2^2$, $2^1 \times 2^1$ (i.e. $p = 4$, 3, 2, 1). Table 1 shows, for an image 256×256, the number of available windows, from 256 to 16,384.

Table 1: Values for an image 256×256 ($2^8 \times 2^8$, $N = 8$).

Number of wished colors	**Dimensions of the windows**	**Available number of windows-pallets** $K = 2^{2(N-p)}$
16,384	$\mathbf{128 \times 128}$	**4**
4,096	$\mathbf{64 \times 64}$	**16**
1,024	$\mathbf{32 \times 32}$	**64**
256	$\mathbf{16 \times 16}$	**256**
64	$\mathbf{8 \times 8}$	**1,024**
16	$\mathbf{4 \times 4}$	**4,096**
4	$\mathbf{2 \times 2}$	**16,384**

2.2 Variability of the pallets

To consider one window, which is extracted from the mixed image, as a potential pallet is one thing, to consider all the extracted windows as equivalent and offering the same potential is another. The average intensity of the image "mandrill" is 123.29. Figures 7 to 10 show the distribution of the average intensities of the windows according to their dimensions. For each one of these histograms, the intensity corresponding to the average intensity of the original image is highlighted in blue. For each case, the distribution takes the shape of a bell-shaped curve (or Gaussian) whose base is widening as dimensions of the windows decrease. The variability of the pallets is more significant when the number of their colors is reduced. The reading of the standard deviation (table 2) of each distribution confirms this analysis. This variability has a consequence: not all the windows have the potential to become a pallet. These observations are presented from the image "mandrill" but they are also valid for other examples. In the case of windows 16×16, which have a reduced range of average intensities, the impact of the choice of an extreme window-pallet will be limited. For windows of smaller size, the extreme intensities are not comparable with the original intensity. It is necessary to develop a method of selection of a window-pallet, which will be the nearest possible to the central tendency. We have considered three approaches [12], which are presented in the continuation of this document.

Table 2: Standard deviation of average intensities of "mandrill" windows.

Number of colors	256	64	16	4
Standard deviation	2.16	3.58	10.95	21.38

2.3 Color quantization of the image and evaluation

Color quantization: Before describing the three approaches of selection, we present the adopted quantization process (see figure 11) and the manner to evaluate the quality of the process result.

We obtain the suitably mixed image from the initial image by the use of a "LUT". Because we know

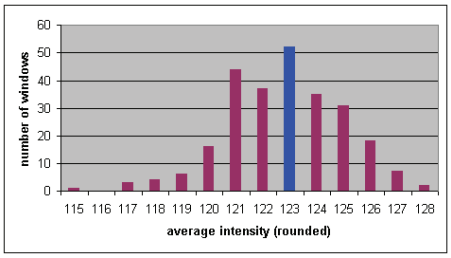


Figure 7: Histogram of average intensities (rounded) of 16×16 mandrill windows (256 colors).

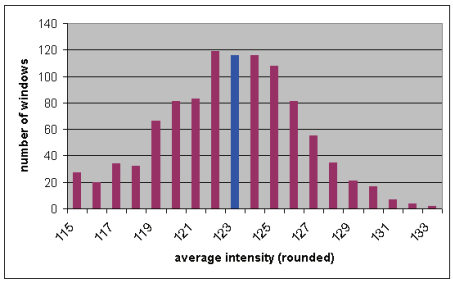


Figure 8: Histogram of average intensities (rounded) of 8×8 mandrill windows (64 colors).

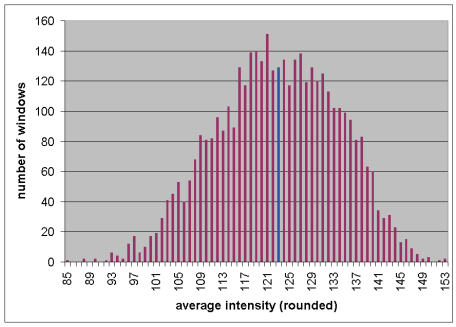


Figure 9: Histogram of average intensities (rounded) of 4×4 mandrill windows (16 colors).

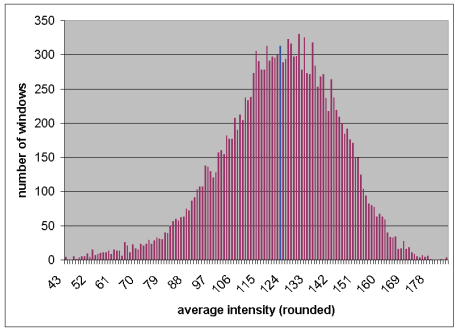


Figure 10: Histogram of average intensities (rounded) of 2×2 mandrill windows (4 colors).

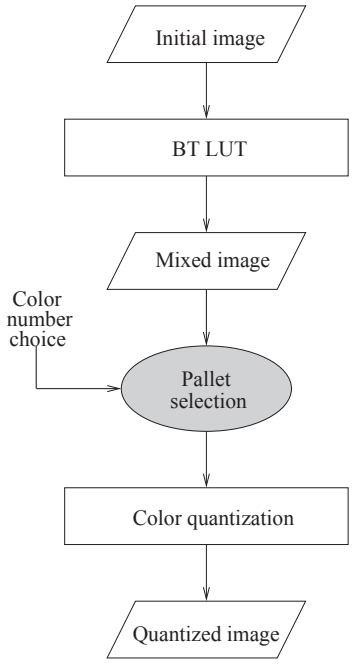


Figure 11: Principle of the adaptive color quantization.

the desired number of colors in the final image, we can determine the size of the extracted window, which is necessary to build the pallet. We choose a window: pixel colors of this window constitute the final pallet. Finally, we assign a color of the pallet to each pixel of the image. To do that, for each pixel of the original RGB image, we calculate an Euclidean distance between its initial color and the 2^{2p} colors of the pallet:

$$\Delta_{RGB}(i,j,k) = \sqrt{(R(i,j) - R(k))^2 + (G(i,j) - G(k))^2 + (B(i,j) - B(k))^2}$$

where (i,j) are the coordinates of the pixel, and k the index in the pallet. We assign to the pixel the color of the pallet giving the minimal distance.

Evaluation: In the following paragraphs, we detail the three suggested methods. For each, we expose the principle of the method, we present results obtained from "mandrill" (see figure 1) reduced to 256, 64, 16 and 4 colors, and we give a table, which shows two distances between the original image and the quantized image. We use these distances to evaluate the quality of the color quantization. The first distance is presented in the form of vector and is called *delta*. It is an average difference between two images, color axis by color axis (see equation 1). The second distance is an average Euclidean distance between two images (see equation 2).

$$delta_{RGB} = \begin{bmatrix} \delta_R \\ \delta_G \\ \delta_B \end{bmatrix} = \begin{bmatrix} \frac{\sum_{i=1}^{2^N}\sum_{j=1}^{2^N}|R_o(i,j)-R_q(i,j)|}{2^N \times 2^N} \\ \frac{\sum_{i=1}^{2^N}\sum_{j=1}^{2^N}|G_o(i,j)-G_q(i,j)|}{2^N \times 2^N} \\ \frac{\sum_{i=1}^{2^N}\sum_{j=1}^{2^N}|B_o(i,j)-B_q(i,j)|}{2^N \times 2^N} \end{bmatrix} \quad (1)$$

$$\Delta_{RGB} = \frac{\sum_{i=1}^{2^N}\sum_{j=1}^{2^N}\sqrt{(R_o(i,j) - R_q(i,j))^2 + (G_o(i,j) - G_q(i,j))^2 + (B_o(i,j) - B_q(i,j))^2}}{2^N \times 2^N} \quad (2)$$

X_o and X_q indicate the original axis and the quantized axis. i and j are the coordinates of the pixels. Initially, we compute these distances on RGB space because it is the original color space of the images. But, in scientific literature ([13] and [14] for example), it is more significant to use a color distance based on color spaces such as L*a*b*. So, to complete our study, we compute similar distances on L*C*h space, which corresponds to cylindrical coordinates of L*a*b* space (see equations 3 and 4). For simplicity, we use the sRGB (standard RGB space) to XYZ conversion matrix. This conversion is defined for D_{65} illuminant and it includes a gamma correction. The conversion from XYZ to

L*a*b* is standard but we replace a* (red-green opposition) and b* (blue-yellow opposition) by $C* = \sqrt{a*^2 + b*^2}$ (chroma) and $h = \arctan \frac{b*}{a*}$ (hue).

$$delta_{LCH} = \begin{bmatrix} \delta_L \\ \delta_C \\ \delta_H \end{bmatrix} = \begin{bmatrix} \frac{\sum_{i=1}^{2^N}\sum_{j=1}^{2^N}|\Delta L*(i,j)|}{2^N \times 2^N} \\ \frac{\sum_{i=1}^{2^N}\sum_{j=1}^{2^N}|\Delta C*(i,j)|}{2^N \times 2^N} \\ \frac{\sum_{i=1}^{2^N}\sum_{j=1}^{2^N}|\Delta H*(i,j)|}{2^N \times 2^N} \end{bmatrix} = \begin{bmatrix} \frac{\sum_{i=1}^{2^N}\sum_{j=1}^{2^N}|L_o^*(i,j) - L_q^*(i,j)|}{2^N \times 2^N} \\ \frac{\sum_{i=1}^{2^N}\sum_{j=1}^{2^N}|C_o^*(i,j) - C_q^*(i,j)|}{2^N \times 2^N} \\ \frac{\sum_{i=1}^{2^N}\sum_{j=1}^{2^N}\left|2\sqrt{C_o^*(i,j)C_q^*(i,j)}\sin\left(\frac{\Delta h_{ab}(i,j)}{2}\right)\right|}{2^N \times 2^N} \end{bmatrix} \quad (3)$$

$$\Delta_{LCH} = \frac{\sum_{i=1}^{2^N}\sum_{j=1}^{2^N}\Delta E_{ab}^*(i,j)}{2^N \times 2^N} \quad (4)$$

with $\Delta E_{ab}^*(i,j) = \sqrt{(\Delta L*(i,j))^2 + (\Delta C*(i,j))^2 + (\Delta H*(i,j))^2}$.

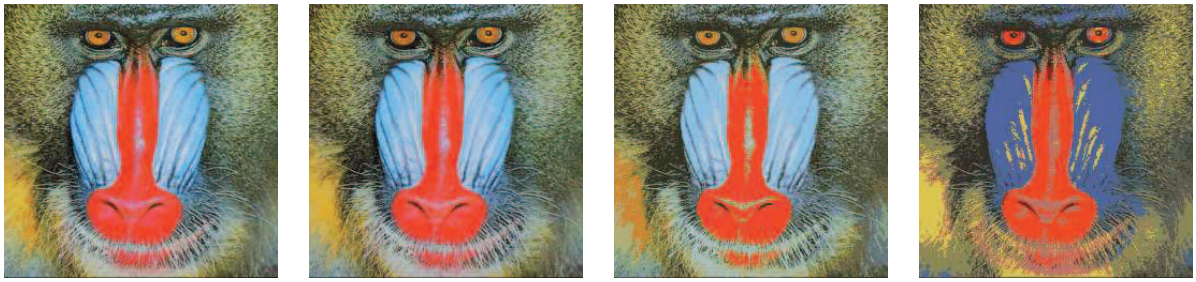
2.4 Techniques of pallet choice

2.4.1 "Best" and random methods

The ideal solution to choose the window-pallet would be to operate all the available quantizations, one by available window, and to compare them with the original image. The window-pallet giving the best result would be selected, but *a posteriori*. This choice is given by seeking the window-pallet, which leads to the minimal *delta* (in a basic way: $\min(\Delta C_1 + \Delta C_2 + \Delta C_3)$ if we consider that C_1, C_2 and C_3 have the same weight). For example, table 3 gives the *delta*, which is obtained from the best and the worst quantization of "mandrill" with 16 and 256 colors. We also give the quantized images corresponding to these values (see figures 12a to 13d). As we suggested in section 2.2, the difference is more perceptible with few colors like 16 colors. We also see that the use of L*a*b* *delta* is visually more relevant than the use of RGB *delta* (see figures 12d and 13d). Although simple to implement, this solution is bad in computing times because of the significant number of possible pallets. For example, the quantization of "mandrill" on 16 colors (4096 available pallets) takes over 11 minutes with a processor frequency of 1.1 GHz and a software written in C++ language.

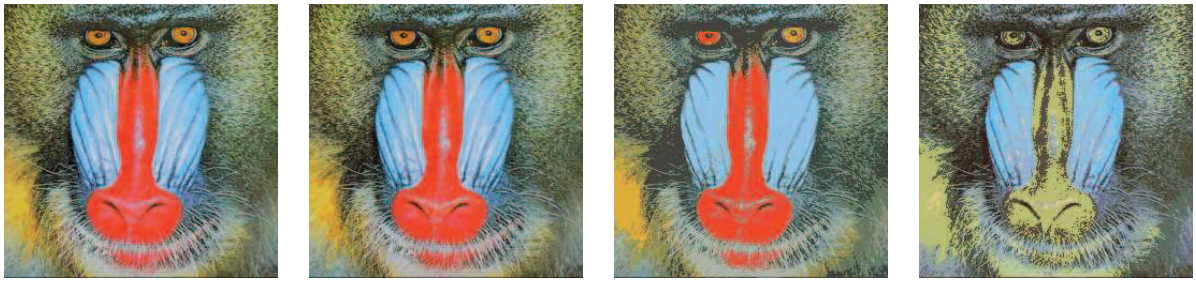
Table 3: Extreme *delta* from "16 colors" and "256 colors" quantization of "mandrill".

Extreme	Colors	δR	δG	δB	Δ_{RGB}	δL	δC	δH	Δ_{LCH}
Best	256	4.90	4.89	4.88	9.86	1.47	3.17	3.07	5.25
	16	12.12	11.15	13.80	24.66	3.97	7.28	6.13	11.87
Worst	256	5.34	5.10	5.37	10.57	1.48	3.44	3.48	5.76
	16	18.14	23.92	29.17	46.54	6.13	13.09	13.20	21.69



a) b) c) d)

Figure 12: Extreme quantizations of "mandrill" according to RGB *delta* : a) best with 256 colors; b) worst with 256 colors; c) best with 16 colors; d) worst with 16 colors.



a) b) c) d)

Figure 13: Extreme quantizations of "mandrill" according to LCH *delta* : a) best with 256 colors; b) worst with 256 colors; c) best with 16 colors; d) worst with 16 colors.

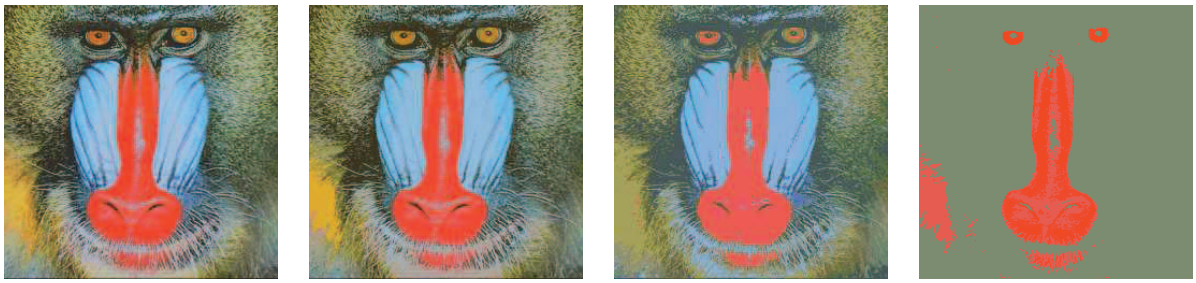
Another simple solution consists to choose by a random way, or by an arbitrary way, one of the K extracted windows, which offers the desired number of colors. Considering the remarks on the variability of the pallets made in section 2.2, we can easily understand that the random choice can lead to a suitable result but also to a "catastrophic" result: on the previous images, we can see important differences due to the used pallet. In the case of a pallet of 256 colors (window 16×16), the difference between the two extremes is weak as we can see on figures 12a, 12b, 13a and 13b. The advantage of a random choice is its low computing time. For example, one quantization of "mandrill" on 16 colors takes less 10 milliseconds in the conditions previously described.

To avoid a method, which gives an unguaranteed result, we imagined three approaches with the aim to obtain a repeatable result. Two of them compute a new window-pallet and one selects a window-pallet among those available. Usual statistic tools like median or average of a distribution inspire these methods.

2.4.2 Building of a color median window-pallet (method 1)

The first approach is to build a new window from the K available windows. For a given pixel (i, j), we seek, in the K windows, the median color at this same position. The color found is affected at the position (i, j) to the new window ($K+1^{\text{th}}$ window). This technique is repeated for all positions. By median color at the position (i, j), we consider the median value in the sorted list of the K mean intensities of pixel (i, j). Figure 14 gives visual results obtained from "mandrill". Table 4 gives numerical results of *delta*. Final colors are in the original image but are not from the same extracted window. The algorithm below details this approach.

For $i = 1$ to 2^p
  For $j = 1$ to 2^p
    For $k = 1$ to K
      $I_{\text{pixel}}(k) = \frac{R(i,j,k)+G(i,j,k)+B(i,j,k)}{3}$
    End
    To sort the $I_{\text{pixel}}(k)$ by increased order.
    $k_{\text{result}} = \{k | I_{\text{pixel}}(k) = I_{\text{pixel}}^{\text{sorted}}(\frac{K}{2})\}$
    $R_{\text{result}}(i, j) = R(i, j, k_{\text{result}})$
    $G_{\text{result}}(i, j) = G(i, j, k_{\text{result}})$
    $B_{\text{result}}(i, j) = B(i, j, k_{\text{result}})$
  End
End



a) b) c) d)

Figure 14: Quantized image by approach 1: a) 256 colors; b) 64 colors; c) 16 colors; d) 4 colors.

Table 4: Values of *delta* calculated from the four images of the figure 14.

Colors	δR	δG	δB	Δ_{RGB}	δL	δC	δH	Δ_{LCH}
256	5.59	5.59	5.91	11.45	1.71	3.47	3.39	5.83
64	8.81	8.61	9.71	18.08	2.71	5.45	5.21	9.02
16	16.54	17.93	16.34	32.81	5.96	8.40	6.35	13.94
4	48.81	38.12	25.85	71.81	13.36	17.14	12.57	28.53

2.4.3 Choice of the median window-pallet (method 2)

In the second approach, we use a criterion to find a single window from the K available windows. We have chosen to calculate the mean intensity of the windows and to retain the window giving the median intensity, as seen in the following algorithm. Figure 15 presents visual results from "mandrill". Table 5 gives numerical results of *delta*. It is possible to simplify this method by using a different criterion: the sum of the color levels on each channel. Logically, resulting image is between the two extreme quantized images (see subsection 2.4.1). We also tested this method by selecting minimal and maximal intensity but results are bad.

For $k = 1$ to K

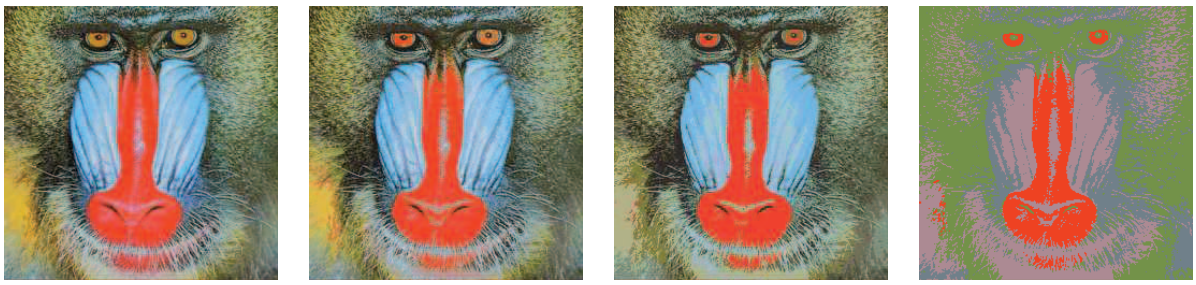
$$I_{\text{window}}(k) = \frac{\sum_{i=1}^{2^p} \sum_{j=1}^{2^p} \frac{R(i,j,k)+G(i,j,k)+B(i,j,k)}{3}}{2^p \times 2^p}$$

End

To sort the $I_{\text{window}}(k)$ by increased order.

$k_{\text{result}} = \{k | I_{\text{window}}(k) = I^{\text{sorted}}_{\text{window}}(\frac{K}{2})\}$

$\text{Window}_{\text{result}} = \text{Window}(k_{\text{result}})$



a) b) c) d)

Figure 15: Quantized image by approach 2: a) 256 colors; b) 64 colors; c) 16 colors; d) 4 colors.

Table 5: Values of *delta* calculated from the four images of the figure 15.

Colors	δR	δG	δB	Δ_{RGB}	δL	δC	δH	Δ_{LCH}
256	5.00	4.90	4.89	9.91	1.45	3.16	3.16	5.28
64	8.22	7.76	8.68	16.44	2.41	4.86	4.92	8.32
16	14.09	14.23	15.57	29.16	4.38	8.85	7.66	14.16
4	25.92	39.78	40.23	67.58	12.95	12.48	14.45	26.15

2.4.4 Building of a mean window-pallet (method 3)

The third approach consists, as for the first, to build a new window ($K + 1^{\text{th}}$ window) from the K initial windows. But, this time, the final color at the position (i, j) is the mean of the K colors at this position, as seen in the following algorithm. Visual results from "mandrill" of this method are presented on figure 16. Table 6 gives numerical results of *delta* obtained from images of figure 16. *A priori* all the colors in the final pallet are new. In the opposite way, this will be coincidence.

For $i = 1$ to 2^p

  For $j = 1$ to 2^p

$$R_{\text{result}}(i,j) = \frac{\sum_{k=1}^{k=K} R(i,j,k)}{K}$$
$$G_{\text{result}}(i,j) = \frac{\sum_{k=1}^{k=K} G(i,j,k)}{K}$$
$$B_{\text{result}}(i,j) = \frac{\sum_{k=1}^{k=K} B(i,j,k)}{K}$$

  End

End

a) b) c) d)

Figure 16: Quantized image by approach 3: a) 256 colors; b) 64 colors; c) 16 colors; d) 4 colors.

Table 6: Values of *delta* calculated from the four images of the figure 16.

Colors	δR	δG	δB	Δ_{RGB}	δL	δC	δH	Δ_{LCH}
256	6.21	6.28	6.80	12.83	1.89	4.03	3.63	6.51
64	11.40	10.31	11.47	21.86	3.27	6.49	5.11	10.14
16	22.60	21.85	27.14	44.90	6.10	13.19	8.27	19.89
4	34.16	34.76	43.65	70.88	11.36	15.73	12.34	27.26

3 Other results and observations

Visually and numerically, the previous results lead us to conclude the second approach is generally the best. This approach consists of seeking the median window-pallet from the K available windows. The interest of this method is confirmed by viewing the processing times. For example, quantization of "mandrill" on 16 colors takes 70 milliseconds for method 1 and 10 milliseconds for methods 2 and 3, in the conditions previously described. To confirm this conclusion, we present a second set of experimental results realized from the images in figure 17. Figure 18 shows quantized images to 256 colors by the second approach of the images in figure 17. If we compare these images, we conclude that the difference between the 2^8 color image and the 2^{24} color image is less obvious. Results from the two other methods are not presented because the differences are not significant in the case of the quantization to 256 colors. In tables 7 and 8, we present numerical results of *delta*, with the best values are in bold. We can see the second approach usually gives the best results according to RGB *delta* and L*a*b* *delta*. We note that δL indicates quasi-systematically approach 2 like the best: this approach preserves more the luminosity than the two others. Concerning approaches 1 and 3, we sometimes obtain better results, but it is generally for 64, 16 or 4 colors.

To illustrate the differences between the three suggested approaches, we present the results obtained for a quantization to 16 colors on "food" and "flowers". "Food" (figure 19) presents a large variety of colors, which occupy small surfaces. "Flowers" (figure 20), on the contrary, presents large regions of relatively homogeneous colors. On these figures, we note approaches 1 and 3 are characterized by a loss of average brightness, which gives an effect of "fog" on the image. This effect is confirmed by the values of δL. The initial colors appear less degraded by approach 3 than by approach 1. Approach 2 is also satisfactory visually because it preserves a good contrast between the colors. Finally, for images with 256 colors and obtained by the second method, the ratio does not exceed three percent between *delta* values and their maximum values.

4 Algorithms comparison

To finalize this work, we compare this quantization method with three classical color quantization methods : *median cut* [15] [16], *split & merge* [17], and *octree* [18] [19]. The comparative study is based on psychovisual tools (subjective criteria) and numerical calculations (objective criteria): M. C. Larabi provides us the means and the experience of his laboratory on the problematic of quantization [6] [20]. The comparison shows the interest of our adaptive colors quantization method by the use of

Figure 17: Original images used for calculations of table 7 (*food*, *flowers*, *girl*, *peppers* & *Lena*).

Figure 18: Quantized images from figure 17 to 256 colors by the method 3 (*food*, *flowers*, *girl*, *peppers* & *Lena*).

Table 7: Values of *delta* calculated from the images of the figure 17 for each approach ($\delta R, G, B \in [0, 255]$, $\delta L \in [0, 100]$, $\delta C \in [0, 134]$, $\delta H \in [0, 252]$).

Method	1	2	3	1	2	3	1	2	3
Image "FOOD" - Size 256 × 256 - Initial coding: 2^{24} colors									
Colors	δR			δG			δB		
256	10.14	**7.04**	17.44	12.72	**7.31**	21.70	12.23	**7.46**	19.04
64	21.50	**12.16**	34.67	28.27	**13.19**	32.17	25.08	**12.68**	39.27
16	43.87	**20.50**	49.66	40.24	**22.77**	47.79	43.46	**24.03**	48.30
4	**49.39**	51.26	58.22	52.47	**32.77**	53.00	54.96	**39.47**	63.57
Colors	δL			δC			δH		
256	3.79	**2.08**	6.11	6.05	**4.62**	11.08	5.39	**4.36**	7.66
64	9.17	**3.65**	10.69	9.70	**7.89**	19.93	9.42	**7.31**	12.88
16	15.24	**6.33**	16.20	12.47	**12.40**	24.01	17.86	**14.17**	14.90
4	19.69	**11.21**	18.55	**18.67**	19.49	25.94	25.14	27.78	**18.03**
Image "FLOWERS" - Size 256 × 256 - Initial coding: 2^{24} colors									
Colors	δR			δG			δB		
256	4.66	**4.65**	5.33	5.89	**4.87**	7.16	5.14	**4.80**	6.26
64	8.99	**8.48**	11.22	9.45	**8.46**	12.99	10.03	**7.71**	12.74
16	21.98	**15.77**	27.18	21.17	**18.95**	29.91	20.11	**13.85**	23.77
4	40.04	**32.97**	57.69	53.51	**26.56**	43.72	**27.62**	34.11	41.48
Colors	δL			δC			δH		
256	1.62	**1.35**	1.91	3.45	**3.13**	4.24	3.13	**2.89**	3.42
64	2.74	**2.40**	3.73	**5.19**	5.64	6.54	5.34	**4.36**	5.77
16	7.26	**5.49**	9.76	**8.88**	8.99	12.09	**7.26**	9.56	11.06
4	17.62	**10.26**	16.40	**12.57**	19.71	15.21	21.48	**14.65**	16.09
Image "GIRL" - Size 256 × 256 - Initial coding: 2^{24} colors									
Colors	δR			δG			δB		
256	4.11	**3.84**	4.48	3.71	**3.50**	3.96	3.64	**3.45**	4.14
64	6.98	**6.68**	8.26	5.85	**5.81**	6.19	**6.39**	6.40	6.53
16	17.24	**15.75**	15.92	11.76	12.59	**11.35**	**10.23**	10.65	10.70
4	22.74	**22.14**	25.91	21.09	25.22	**20.28**	20.19	20.49	**18.57**
Colors	δL			δC			δH		
256	1.23	**1.12**	1.27	2.39	**2.34**	2.60	2.53	**2.51**	2.82
64	2.04	**1.93**	2.17	**3.92**	3.95	4.31	3.80	**3.79**	3.89
16	4.78	**3.82**	4.48	7.06	7.47	**6.63**	5.94	6.54	**4.56**
4	**6.98**	8.16	7.19	**9.53**	10.72	10.04	**5.51**	8.04	6.31
Image "PEPPERS" - Size 256 × 256 - Initial coding: 2^{24} colors									
Colors	δR			δG			δB		
256	3.98	**3.91**	4.96	3.79	**3.66**	4.51	4.14	**3.81**	5.60
64	8.35	**6.36**	12.62	7.08	**6.33**	11.91	9.75	**6.58**	11.49
16	17.76	**11.29**	19.97	16.39	**9.42**	32.44	18.15	**11.16**	22.69
4	49.52	37.06	**31.33**	26.69	**21.74**	55.37	**19.98**	22.72	26.39
Colors	δL			δC			δH		
256	1.12	**1.08**	1.44	2.64	**2.39**	3.22	2.25	**2.09**	2.60
64	2.45	**1.86**	4.27	5.58	**3.99**	5.87	4.04	**3.43**	4.61
16	6.12	**3.17**	9.30	8.89	**6.85**	9.63	6.55	**5.52**	8.40
4	12.11	**9.35**	14.06	12.48	**12.39**	19.40	**9.91**	11.66	21.45
Image "LENA" - Size 256 × 256 - Initial coding: 2^{24} colors									
Colors	δR			δG			δB		
256	2.92	**2.62**	3.63	3.05	**2.75**	3.20	3.13	**2.86**	3.28
64	5.47	**4.65**	6.23	5.58	**4.92**	6.49	5.33	**5.07**	7.80
16	11.65	**9.12**	19.63	14.34	**8.18**	17.22	11.35	**7.95**	13.80
4	23.65	**17.81**	28.13	**20.82**	23.84	27.15	20.40	**18.73**	19.54
Colors	δL			δC			δH		
256	0.86	**0.74**	0.93	1.55	**1.51**	1.77	**1.86**	1.90	2.12
64	1.65	**1.37**	1.98	**2.51**	2.53	2.93	**3.04**	3.15	3.45
16	4.50	**2.75**	6.08	4.47	**3.78**	5.09	5.24	**4.44**	6.69
4	7.43	**6.64**	9.74	9.92	10.14	**6.63**	10.24	**7.69**	8.39

Table 8: Values of Δ_{RGB} and Δ_{LCH} calculated from the images of the figure 17 for each approach.

	Δ_{RGB} - $range \approx [0, 440]$			Δ_{LCH} - $range \approx [0, 260]$		
Method	1	2	3	1	2	3
Colors	Image "FOOD"					
256	23.34	**14.89**	37.64	10.36	**7.61**	16.77
64	48.18	**26.15**	68.00	18.59	**13.12**	29.29
16	81.70	**45.36**	92.25	30.34	**22.32**	37.04
4	102.82	**84.23**	109.59	**41.63**	41.93	42.30
Colors	Image "FLOWERS"					
256	10.63	**9.67**	12.58	5.59	**5.04**	6.59
64	18.81	**16.67**	24.05	9.16	**8.50**	11.07
16	40.18	**32.57**	51.38	**15.72**	16.49	21.79
4	78.54	**62.24**	89.44	33.96	**29.89**	32.48
Colors	Image "GIRL"					
256	7.85	**7.42**	8.36	4.18	**4.07**	4.53
64	12.94	**12.71**	13.83	6.65	**6.63**	7.00
16	25.98	25.84	**24.55**	11.93	12.25	**10.70**
4	**40.01**	43.39	41.21	**15.67**	17.83	16.51
Colors	Image "PEPPERS"					
256	8.16	**7.71**	10.17	4.14	**3.79**	4.92
64	17.42	**13.11**	23.55	8.21	**6.33**	9.62
16	35.05	**22.01**	47.59	14.28	**10.68**	18.08
4	63.88	**54.62**	73.67	22.17	**22.07**	34.78
Colors	Image "LENA"					
256	6.03	**5.51**	6.69	2.93	**2.85**	3.31
64	10.68	**9.66**	13.42	4.91	**4.83**	5.76
16	23.76	**16.52**	31.79	9.46	**7.33**	12.10
4	43.34	**39.43**	46.72	18.17	16.48	**16.41**

a) b) c)

Figure 19: Image "food" quantized to 16 colors by the three methods.

a) b) c)

Figure 20: Image "flowers" quantized to 16 colors by the three methods.

the "Baker's Transformation". Note that we use only the second method based on the selection of the median pallet because of our previous conclusions.

4.1 Classical color quantization methods

4.1.1 Median cut method

Heckbert has introduced the *median cut* algorithm in 1980 [15]. Its concept is based on using each color of the synthesized color map in order to represent an equal number of pixels in the original image. This consist in recursive subdivision of the RGB color space cube, containing the $n \times m$ colors (with n and m respectively the row and column numbers), in rectangular boxes of continuous smaller size. The basic idea is to obtain the same number of pixels in these boxes.

4.1.2 Split & Merge method

This method, introduced by Brun in [17], uses two steps. In the first step, we uniformly split the color space in N classes. Then we have to merge $N - N_c$ classes to obtain the wished number of colors. We look for the two classes. The merging of these two classes has to give a minimal error. This process is iterated until we obtain N_c classes.

4.1.3 Octree method

This method introduced by Gervautz and Purgathofer [19], as the *median cut* method, is based on the partition of the RGB space. The RGB cube is divided in eight sub-cubes of same size by iterative way. The deep of the tree is equal to $\log_2(N_c)$. Leaves of this tree are the colors from the image. Then we have to reduce the tree in order to obtain N_c leaves only. Nodes having no leaves or a small number of leaves are removed.

4.2 Tools for quality evaluation

The quality assessment is an unavoidable stage of color quantization algorithms. In fact, it is important to compute a distance criterion between original and reduced images. To do that, we use both objective and subjective assessment. Notice that using only objective criteria is not enough because the color is directly related to perception, which is very subjective and difficult to quantify.

Objective assessment. In some of works on color quantization, authors use only the "Mean Square

Error" (or MSE) [21] to measure the distortion given to the original image. MSE is defined by:

$$MSE(I, I_q) = \frac{1}{N} \sum_{(x,y) \in I} D^2(I(x,y); I_q(x,y)) \tag{5}$$

where I and I_q are respectively the image before and after the color quantization, N is the total number of pixels of the image, and D^2 indicates the quadratic error computation. We also propose to use local descriptors of quality [22]. We compute a value called D_Q, by the following equation:

$$D_Q = \frac{1}{2}\sqrt{D_{Lum}^2 + D_{Chr}^2 + D_{Emg}^2 + D_{Cor}^2}$$

where D_{Lum} is the local difference of intensity, D_{Chr} is the local difference of chromaticity, D_{Emg} is the local difference of color occurrence in a given neighborhood, and D_{Cor} is the local difference of color correlation between two regions.

Subjective assessment. Here, experiment is of "human in loop" type. It uses the capacities of human visual system (HVS) to evaluate quality of images. The human observers have a normal perception of the colors according to Ishihara test [23]. The experiment is realized in a standardized room according to the international norms ITU-R 500-10 [24]. Walls are in neutral gray: the light is not reflected. The illumination is artificial and controlled. As shown on figure 21, the observers were seated three feet from the calibrated CRT. The screen delivers sufficient light to stimulate the HVS. The experiment is an ordering test. The aim of this test is to make a classification (from the best to the worst) of an image series with regard to the original image. For the needs of this psychophysical experimentation, we retained, as device of study, a process that shows on the screen the original image and the four images reduced by the previous methods, as shown in figure 22. Then, the human observer is asked to indicate, by a mouse click, the image he sees most qualitatively distant from the original image. This image is then masked and the same question is then asked and this until the four images are masked in the presentation device. This technique allows having a classification of the images with regard to a reference one. Contrary to the classical techniques where the observer has to classify the images in increasing or lessening quality order, this technique presents the advantage to never put the observer in a complex choice position.

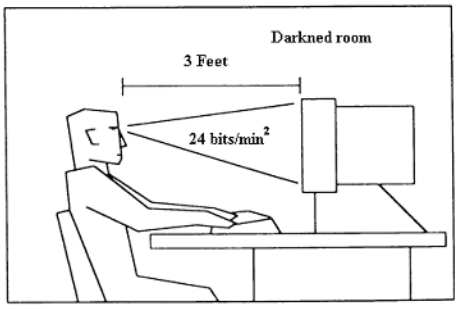


Figure 21: Psychophysical quality assessment protocol.

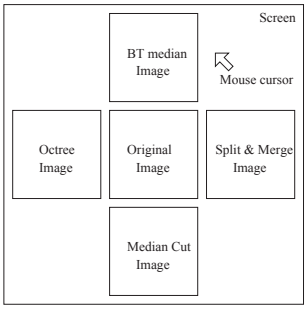


Figure 22: Ordering test principle on the screen.

4.3 Results

For our experimental needs, we used images classified in four categories: portraits, outdoor scenes, indoor scenes and computer-generated images. These categories were chosen because they provide a range of natural and artificial colors and textures. All of the images were with a precision of eight bits per channel and three channels per pixel. The methods are identified by "BTmed" for our method, "Oct" for *octree*, "S&M" for *split & merge*, and "Mcut" for *median cut*. Figure 23 presents the "Mean Square Error", which is computed on all database images and for the four numbers of colors. Smaller the error is, better is the result: we can see the good behavior of *Oct* and *BTmed* methods for 64 and 256 colors. Figure 24 shows the value of the "quality" descriptor D_q, which is also computed on all database images and for the four numbers of colors. This value is normalized between 0 and 1. The best quality corresponds to $D_q = 1$: best methods are the same than previously.

The subjective tests are performed with a panel of 20 observers, with different image processing backgrounds, which have been evaluated for visual acuity and normal vision of colors by using the Ishihara test. Subjective assessments are described by Mean Opinion Score (MOS). This value shows the observer behavior in front of the different visual tests. Table 9 presents average and standard deviation of MOS for all images and observers. Higher is the MOS value, best is the quantization method. Nevertheless, MOS value above 4 is considered as very good result. Only *Mcut* method does not satisfy this condition. Our method is not the best but it can be qualified of "good".

Table 9: MOS and standard deviation for all images.

Algorithm	MOS	Std. dev.
BTmed	4.6	2.1
Oct	5.1	1.9
S&M	5.3	2.3
Mcut	3.0	2.1

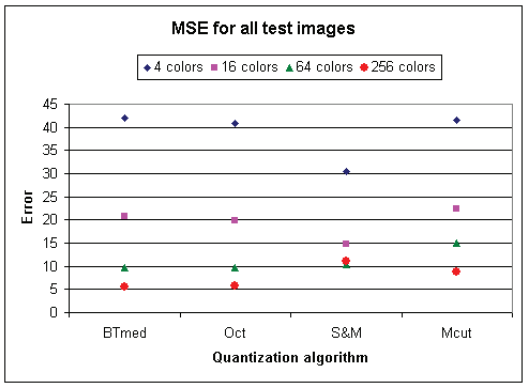


Figure 23: Quality descriptor for all images.

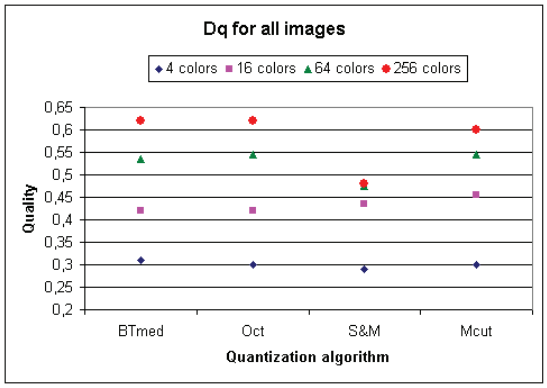


Figure 24: Mean quadratic error for all images.

5 Conclusion

In this article we present a technique to reduce the number of colors contained in an image. This method, based on the use of the "Baker's Transformation", is very effective and has the advantage of being completely adaptive. The user imposes only the desired number of colors in the final image. We explored three approaches to select the pallet giving the most satisfactory quantization. The approach based on the research of the median pallet satisfies this aim. This result is confirmed by analysis performed on L*a*b* space. To conclude about the interest of our method, we compared it with classical methods like *median cut*, *split & merge* or *octree*. Objective and subjective assessments show that our method is as good as the *octree* method.

Future works will be to improve our method by introducing a weight of color axes in *delta*, and to use our method in color quantization in video sequence. We are currently exploring another way, which consists of using this technique to build a color "invariant". This "invariant" would be used in a procedure of image indexing.

References

[1] M. T. Orchard and C. A. Bouman. Color quantization of images. *IEEE Transactions on Image Processing*, 39(12):2677–2690, December 1991.

[2] G. Sharma and H.J. Trussel. Digital color imaging. 6(7):901–932, July 1997.

[3] X. Chen, R. Kothari, and P. Klinkhachorn. Reduced color image based on adaptive palette color selection using neural networks. In *IEEE Proceedings*, pages 555–558, 1994.

[4] J. Ketterer, J. Puzicha, M. Held, M. Fischer, J. M. Buhmann, and D. Fellner. On spatial quantization of color images. In *Proc. of European Conference on Computer Vision*, pages 563–577, Freiburg, Germany, 1998.

[5] C. Fernandez-Maloigne and N. Richard. Coding of color images with cooccurrence matrice descriptors. In *Proc. of ECCV'2000*, Dublin, Ireland, June 2000.

[6] M. C. Larabi, N. Richard, and C. Fernandez. A Fast Color Quantization Using a Matrix of Local Pallets. In *IEEE Applied Imagery Pattern Recognition*, pages 136–140, Washington, USA, octobre 2000.

[7] A. Smolarz and P. Cornu. A color image classification by means of image transformations. In *Proc. of CGIV'2002*, Poitiers, France, June 2002.

[8] P. Cornu and A. Smolarz. Caractérisation d'images par textures associées. *Traitement du Signal*, 19(1):29–35, 2002.

[9] V.I. Arnold and A. Avez. *Problèmes ergodiques de la mécanique classique.* Gauthier-Villars, Paris, 1967.

[10] P. Billingsley. *Ergodic Theory and Information.* John Wiley & Sons Inc., New-York, 1965.

[11] Philippe Cornu and André Smolarz. Caractérisation de la signature texturelle d'une image. In *Proc. du 18ème colloque GRETSI sur le traitement du signal et des images*, Toulouse - France, septembre 2001.

[12] Christophe Montagne and Sylvie Lelandais. Adaptive color quantization using the baker's transform. In *Proc. of CGIV 2004*, Aachen, Germany, April 2004.

[13] Alain Tremeau, Christine Fernandez-Maloigne, and Pierre Bonton. *Image numérique couleur - De l'acquisition au traitement*, chapter 2. Sciences de l'ingénieur. Dunod, Paris, France, 2004.

[14] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae.* 2nd edition. John Wiley & sons, 1982.

[15] P. S. Heckbert. Color Image Quantization for Frame Buffer Display. B.S. thesis, Architecture Machine Group, MIT, mai 1980.

[16] P. S. Heckbert. Color Image Quantization for Frame Buffer Display. *ACM Computer Graphics*, 16(3):297–307, 1982.

[17] L. Brun and M. Mokhtari. Two high speed color quantization algorithms. In *Proc. of CGIP'2000*, Saint-Étienne, 2000.

[18] M. Gervautz and W. Purgathofer. A simple method for color quantization: Octree quantization. *New Trends in Computer Graphics*, pages 219–231, 1988.

[19] M. Gervautz and W. Purgathofer. A simple method for color quantization: Octree quantization. *Graphics Gems*, pages 287–293, 1990.

[20] M. C. Larabi, N. Richard, and C. Fernandez. A new quantification method under colorimetric constraints. In *IS&T Conf. on Color in Graphics, Image and Vision*, pages 412–415, Poitiers, France, 2002.

[21] A. M. Eskicioglu and P. S. Fisher. Image quality measures and their performance. *IEEE Transactions on Communications*, 43(12):2959–2965, December 1995.

[22] C. Charrier. *Vers l'optimisation statistique et perceptuelle de la qualité pour la compression des images couleur par quantification vectorielle.* Thèse de doctorat, Université Jean Monnet Saint-Etienne, 1998.

[23] D. McIntyre. *Colour Blindness: Causes and Effects.* Dalton Publishing, Chester, 2002.

[24] ITU. Itu-r recommendation bt.500-10: Methodology for the subjective assessment of the quality of television pictures. Technical report, ITU, Geneva, Switzerland, 2000.