Inverse Problems and Imaging
Volume 1, No. 2, 2007, 265–287
Web site: http://www.aimSciences.org

AUTOMATIC COLOR PALETTE

J. Delon[†], A. Desolneux[††], J. L. Lisani[‡] and A. B. Petro[‡]

[†]LTCI Télécom Paris, 46 rue Barrault 75013 Paris, France

[††]MAP5, Univ. Paris 5, 45 rue des Saints-Pères, 75006 Paris, France

[‡]Univ. Illes Balears, Ctra. Valldemossa km 7,5 07122 Palma de Mallorca, Spain

(Communicated by Jean-Michel Morel)

Abstract. We present a method for the automatic estimation of the minimum set of colors needed to describe an image. We call this minimal set "color palette". The proposed method combines the well-known K-Means clustering technique with a thorough analysis of the color information of the image. The initial set of cluster seeds used in K-Means is automatically inferred from this analysis. Color information is analyzed by studying the 1D histograms associated to the hue, saturation and intensity components of the image colors. In order to achieve a proper parsing of these 1D histograms a new histogram segmentation technique is proposed. The experimental results seem to endorse the capacity of the method to obtain the most significant colors in the image, even if they belong to small details in the scene. The obtained palette can be combined with a dictionary of color names in order to provide a qualitative image description.

1. Introduction

The human visual system is a complex and precise entity that is able both to distinguish millions of colors and to describe an image by naming just a few of them [24]. This last characteristic derives from its ability to group colors with similar tonality and to assign a unique name to each group. Humans perform this process effortlessly but, computationally, it is not an easy task. We aim at the automation of such a process.

The example in Figure 1 illustrates the main difficulties of the problem of representing a color image with a minimum set of colors. In this figure, the same original image is represented with two different sets of colors. Both sets consist of 12 colors, but why 12 and not 10, 20 or any other value? How many colors are needed for an acceptable representation? Moreover, even if this number can somehow be estimated, the question of finding the best set of colors remains open, as it can be observed in the example. In both cases the number of colors is the same, however the second set represents more accurately the original image. In particular, the color of the small ladybug in the leaf is correctly represented in the second image, while it is missing in the first one.

Based on the previous observations, we consider that any acceptable image representation must satisfy two basic requirements: reduction of redundant colors, and

2000 *Mathematics Subject Classification:* 65J20, 65J15.

Key words and phrases: Color, clustering, vision and scene understanding, scene analysis.

This work has been partially financed by the Centre National d'Etudes Spatiales (CNES), the Office of Naval Research under grant N00014-97-1-0839, the Direction Générale des Armements (DGA), the Ministerio de Ciencia y Tecnología under grant TIC2002-02172.

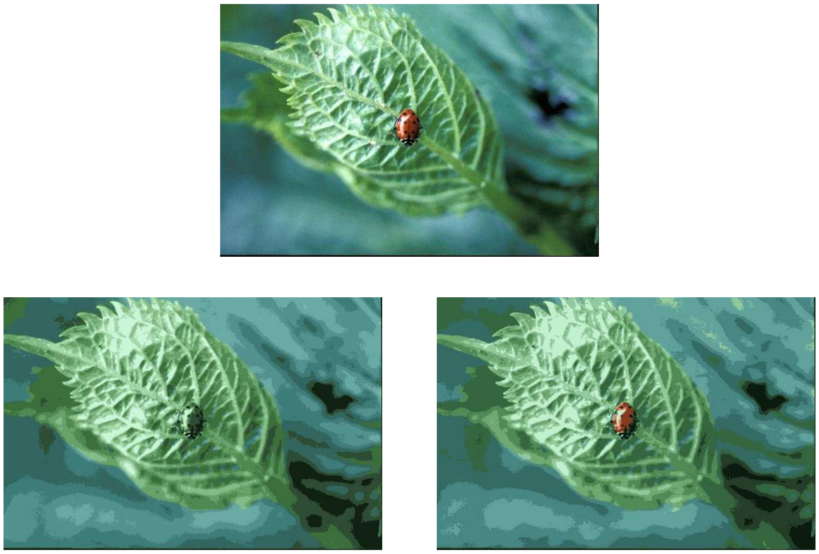


Figure 1. Top, original image (600 × 400 pixels) . Bottom, two representations of the original image with two different sets of 12 colors each.

preservation of rare colors. The first requirement leads to a minimum set of colors, while the second one ensures that colors corresponding to small but perceptually significant objects in the image will be preserved in the final set.

The problem of representing a color image with a minimum set of colors is well known by the Computer Graphics community and it is usually referred to as **color palette estimation**, by an analogy between this set of colors and the palette of a painter. The image description provided by the color palette has several applications: selection of the optimum displayable colors in computer graphics, color-based image indexing and retrieval [37], compression of color information [35], color images segmentation [40], etc.

However, the most popular techniques used in Computer Graphics for obtaining the color palette (Median Cut [22], Popularity algorithm [22] and Octree algorithm [16]) fail to meet the above requirements. First, the user needs to specify the desired number of colors. Moreover, they can not cope with the problem of representing small image details, as will be shown in the experimental section.

More general clustering techniques, such as K-Means and Mean Shift [43], still present the drawback that one needs to estimate the right number of clusters. An additional problem relates to the fact that the obtained results depend on the initial set of cluster seeds. Although some general methods have been proposed to overcome these drawbacks (see Section 2), none of these techniques takes into account the special characteristics of the color clustering problem.

We propose in this paper to combine the K-Means clustering technique with a thorough analysis of the color information of the image in order to obtain an automatic and very accurate representation of any color image. The proposed method decomposes color information into hue, saturation and illumination, which are magnitudes that can be interpreted intuitively and have a physical meaning [44]. Groups of colors are obtained by applying the following hierarchical algorithm:

initially, colors are discriminated by their hue; next, colors with similar hue are discriminated by their saturation; finally, colors with similar hue and saturation are discriminated by their intensity. At each step of the algorithm the discrimination between different values of a given magnitude (hue, saturation or illumination) is performed by analyzing the associated 1D histograms.

As a result of this analysis a small set of color groups is obtained and a representative for each group is chosen. This set of color representatives is used as an initial seed for the K-Means algorithm in order to obtain the final palette. Optionally, the information of the color palette can be combined with a dictionary of color names, thus obtaining a qualitative description of the image.

The paper is organized as follows. In the next section we give an account of the main techniques found in the literature for color clustering, highlighting their advantages and shortcomings. It must be remarked that the review only includes those techniques that do not make use of spatial relations between image pixels. Only methods using global color information are reported. Therefore no references to segmentation methods have been included. Section 3 describes a new technique for analyzing 1D histograms, which is at the core of the method for the automatic estimation of the initial set of significant image colors. This method, the so-called ACoPA algorithm, is described in detail in Section 4. The use of ACoPa as the initialization step of the K-Means algorithm leads to an automatic method for obtaining the color palette. Section 5 explores the combination of this palette with a dictionary of color names in order to obtain a qualitative description of the images. Section 6 displays some results of the proposed algorithm. Some examples illustrating the shortcomings of classical color clustering techniques are also shown, together with the improvements provided by the combination of the ACoPa and the K-Means algorithms. Also some experiments on the use of dictionaries of color names are shown. Finally, some conclusions and final remarks are outlined in the last section of the paper.

To improve the readability of the paper two short appendices have been added, describing the K-Means clustering algorithm and a 1D histogram transformation algorithm (called "Pool Adjacent Violators") needed in Section 3.

2. A review of color clustering methods

The problem of color image representation, using a color palette construction, can be considered as an unsupervised classification of the three-dimensional color space. Several techniques have been proposed to solve this problem. First, there is the class of splitting algorithms that divide the color space into disjoint regions, by consecutively splitting up the color space. From each region, a color is chosen to represent it in the palette. These algorithms are used for the Computer Graphics color palette construction where speed is a required characteristic. Among the most classical approaches we can cite the Popularity algorithm [22], the Octree algorithm [16] or the Median-Cut algorithm [22]. The most popular is the Median Cut since it is faster than the Octree algorithm and produces better results than the Popularity algorithm. This algorithm repeatedly subdivides the color space into smaller and smaller rectangular boxes, until there are as many boxes as requested colors in the final palette. All these methods partition the color space into a number of clusters defined by the user.

Another class of color palette construction techniques considers the problem as a clustering approach and performs clustering of the color space. The clustering techniques [25] can be divided in *hierarchical* and *partitional clustering*. *Hierarchical clustering* aims at obtaining a hierarchy of clusters, called dendogram, that shows how the clusters are related to each other. Most hierarchical clustering algorithms are variants of the single-link [39], complete-link [28] and minimum-variance [32, 42]. A partition of the data items can be obtained by cutting the dendogram at a desired level, but the partitional clustering algorithms are more useful for this purpose. A *partitional clustering* algorithm obtains a single partition of the data instead of a hierarchical structure. Hierarchical algorithms are more versatile than partitional ones but the time and memory requirements of the partitional algorithms are typically lower than those of the hierarchical algorithms. Partitional methods have advantages in applications involving large data sets for which the construction of a dendogram is computationally prohibitive. Many of these methods are based on the iterative optimization of a criterion function reflecting the "agreement" between the data and the partition. The criteria can be divided in three categories:

- *Methods using the squared error* attempt to minimize a cost function that is the sum over all the data items of the squared distance between the item and the prototype of the cluster it is assigned to. The K-Means [43] is the simplest and the most commonly used algorithm employing this criterion.
- *Density-based methods* consider that clusters are dense sets of data items separated by less dense regions. Dense regions correspond to the modes of the data items. Once the location of a mode is determined, the cluster associated with it is delineated based on the local structure of the data items. Some algorithms of this category are the mean-shift [9] and the GDB-SCAN [36].
- *Mixture-resolving methods* assume that the patterns to be clustered are drawn from one of several distributions, and the goal is to identify their parameters. Most of the work in this area has assumed that the individual components of the mixture density are Gaussian. The introduction of the expectation maximization (EM) algorithm in [11] was an important step in solving the parameter estimation problem.

Most of these methods present two important drawbacks:

- they assume that the number of clusters K in the database is known beforehand which, obviously, is not necessarily true in real-world applications,
- as iterative techniques, these algorithms are especially sensitive to initial conditions (initial clusters and instance order).

Clustering methods have been compared and studied in different works [25, 13, 20]. Although K-Means suffers from the previous drawbacks, it is the most popular clustering algorithm since it is easy to implement, and its time complexity is $O(n)$, where n is the number of data items. K-Means presents another important drawback: its convergence to a global minimum is not insured. Several variants and improvements of this method have been reported in the literature to avoid these drawbacks [26, 4]. The problem of determining the number of clusters can be partly solved by adding a regularization term to the cost function. With such solutions, instead of the number of clusters one has to control a regularization parameter, which is often more convenient but it introduces a new parameter in the algorithm. Another usual but rough approach is to try clustering with several values of K and select the "best" solution.

Milligan [31] shows the strong dependence of the K-Means algorithm on initial clustering. Some works [14, 23, 38, 3] propose new clustering methods as initialization steps for K-Means, resulting in a hybrid clustering method. The problem is that these initialization methods suffer from the same problems as the K-Means algorithm and they have to be provided with an initial clustering. Much simpler and more inexpensive initialization methods have been presented in other works [15, 30, 26, 6]. Most of them use the randomization as main tool in their approaches. A more recent work is [1], which is based on finding a set of medians extracted from a dimension with maximum variance.

Moreover, the above-mentioned works of initialization and clustering do not take into account the 3D color space particularities. In particular, some authors [33] have observed that the color clouds of the real world images usually present half-moon shapes in RGB space. Clustering in 3D color space has to imply the detection and separation of these structures. Most of the works in the literature using 3D histograms do not take into account theses characteristics and they divide the space by means of rigid structures such as hyperrectangles or spheres [7, 43, 9]. Other works, such as [33] and [34], take into account the statistical properties, such as shape and distribution of the clusters in the histogram. The main drawback of these latter methods is that they do not have a sound theoretical background.

In this work, we present a new method for the automatic determination of the number of color clusters and for obtaining an initial clustering of the data. The proposed method is based on the statistical analysis of the colors distribution. The new approach allows us to solve the main drawbacks of the K-Means method and to obtain an accurate color image representation with a reduced number of colors. The result of the Automatic Color Palette (ACoPa) algorithm is a set of colors that can be used as an initialization of the K-Means method.

3. A new statistical histogram segmentation approach

The ACoPa algorithm is based on the hierarchical analysis of the hue, saturation and intensity histograms of the image colors. In this section we describe the construction of an automatic and non-parametric method for 1D histogram segmentation. Such a method should be fully unsupervised and should provide a set of relevant modes, representing suitably the data set. The proposed approach is based on the following observation: on the intervals where they concentrate, data generally follow a "unimodal" law. In a color image, for example, the hue values of the pixels of a given object are naturally distributed according to a unimodal law around the average hue of the object. Following this observation, it seems appropriate to look for segments on which the histogram is "statistically unimodal". This term will be defined later, but, intuitively it implies that the distribution of values in the interval is (aproximately) increasing up to some maximum value and decreasing from there on. Remark that the notion of "unimodality" is more general than the one of "Gaussian model": a Gaussian distribution clearly follows the unimodal hypothesis, but not all the unimodal distributions can be modeled as Gaussians.

Thus, we are led to the question of finding the right tests to decide whether an interval is unimodal or not. In a non-parametric setting, we can hypothesize any unimodal density and test whether the observed law is compatible with the hypothesized one. Unfortunately, this leads to a huge number of tests and is therefore impossible. There is, however, a way to address this question.

Consider a discrete histogram $h = (h_i)_{i=1\dots L}$, with N samples on L bins $\{1, \dots L\}$. The number h_i is the value of h in the bin i. It follows that $\sum_{i=1}^{L} h_i = N$. For every interval I of $\{1, ..., L\}$, we note $h_{|I}$ the restriction of h to I.

We call segmentation of h a sequence $1 = s_0 < s_1 < \cdots < s_n = L$. The number n is termed length of the segmentation. Our aim is to find an "optimal" segmentation S of h, such that h is roughly unimodal on each interval $[s_i, s_{i+1}]$ of S. Obviously, deciding if a part $h_{|I}$ of a histogram is the realization of an unimodal law can be deduced easily if we know how to test the assumption that h is the realization of a monotone law on an interval. This can be done thanks to the Grenander estimator [19].

On each interval I of $\{1, ..., L\}$, we can compute the decreasing Grenander estimator of $h_{|I}$, which is defined as the non-parametric maximum likelihood estimator restricted to decreasing densities. Introduced by Grenander in 1956 [19], this estimator can be easily derived from h by an algorithm called "Pool Adjacent Violators" (see [2, 5]). This algorithm, described in Appendix B, leads to a unique decreasing step function on I. In a symmetric way, we can compute the best increasing estimator of a histogram on any interval. An example of a histogram and its Grenander estimator is shown on Figure 2.

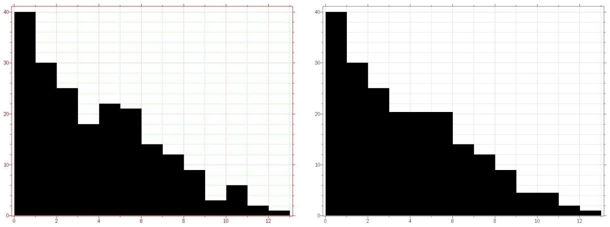


Figure 2. The right histogram is the Grenander estimator of the left histogram on all the interval, computed by the "Pool Adjacent Violators" algorithm.

Now, let T be a statistical test that permits us to decide if a part of a histogram is likely to be the realization of a given law (T could be for example a Kolmogorov-Smirnov, a Chi-2, or even a multiple test).

Definition 1. We say that a histogram h follows the decreasing (resp. increasing) hypothesis on an interval I, if it is likely for the test T that $h_{|I}$ is a realization of the law defined by its decreasing (resp. increasing) Grenander estimator.

Of course, this definition depends on the choice of the statistical test T, and on the significance level of the test. In order to avoid the usual problems linked with significance levels, we used a multiple statistical test, introduced by Desolneux et al. [12] and we further developed in the case of histograms in [10]. This multiple test presents the advantage of having a significance level with a clear physical meaning and easy to fix. However, other tests could be used here.

From the previous definition, we can easily define what it means to be "statistically unimodal" on a segment.

Definition 2. We say that a histogram h follows the unimodal hypothesis on the interval $[a, b]$ if there exists $c \in (a, b)$ such that h follows the increasing hypothesis on $[a, c]$ and h follows the decreasing hypothesis on $[c, b]$.

If s is the segmentation defined by all the minima of h, then h follows obviously the unimodal hypothesis on each of its segments. But this segmentation is not reasonable in general (see the left part of Fig. 3). A segmentation following the unimodal hypothesis on each segment is generally not unique. Then, in order to build a minimal (in terms of number of separators) segmentation, we introduce the notion of "admissible segmentation".

Definition 3. Let h be a histogram on $\{1, ..., L\}$. We will say that a segmentation s of h is admissible if it satisfies the following properties:

- h follows the unimodal hypothesis on each interval $[s_i, s_{i+1}]$,
- there is no interval $[s_i, s_j]$ with $j > i + 1$, on which h follows the unimodal hypothesis.

The first requirement allows us to avoid under-segmentations, and the last one is imposed in order to avoid over-segmentations. It is clear in the discrete case that such a segmentation exists: we can start with the limit segmentation containing all the minima of h and merge recursively the consecutive intervals until both properties are satisfied. This is the principle used in the following algorithm.

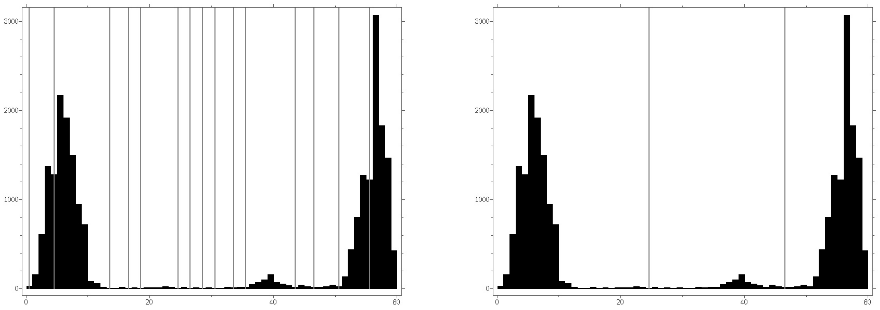


FIGURE 3. Left: Typical histogram of hue values of a color image. This histogram cannot be represented by a gaussian mixture. The vertical lines represent the initialization of the FTC algorithm (all the minima of the histogram). The histogram presents small oscillations, which create several local minima. Right: final result of the segmentation using the FTC algorithm. The histogram is parsed into three modes.

Fine to Coarse (FTC) Segmentation Algorithm:

1. *Define the finest segmentation (i.e. the list of all the local minima, plus the endpoints 1 and L)* $S = \{s_0, \ldots, s_n\}$ *of the histogram.*
2. *Repeat:*

   Choose i *randomly in* $[1, length(S) - 1]$. *If the segments on both sides of* s_i *can be merged into a single interval* $[s_{i-1}, s_{i+1}]$ *following the unimodal hypothesis, group them. Update* S.

   Stop when no more pair of successive intervals follows the unimodal hypothesis.
3. *Repeat step 2 with the unions of* j *segments,* j *going from* 3 *to* $length(S)$.

Remark that the method is completely automatic, in the sense that the user does not need to specify the number of modes in the final segmentation. This quantity is automatically determined by the algorithm.

An example of the application of this algorithm to a typical histogram of hue values is shown in Fig. 3. This kind of histogram cannot usually be modeled as a mixture of Gaussians so the classical techniques of histogram segmentation do not give the desired results. The final segmentation of the histogram gives three modes (shown on the right side of the figure).

A detailed description of the FTC algorithm and its theoretical background can be found in [10].

4. Color palette construction

In this section we present a method for obtaining a statistically meaningful set of color groups from any color image. The meaningfulness of a group is assessed by analyzing, using the FTC algorithm described in the previous section, the 1D histograms of the hue (H), saturation (S) and intensity (I) components of the image colors. This analysis is performed hierarchically:

- First, colors are discriminated by their hue: meaningful modes are detected in the hue histogram of the image and all the color points belonging to the same mode are grouped together.
- Next, colors with similar hue (those belonging to the same mode in the hue histogram) are discriminated by their saturation: the saturation histogram for these colors is computed and parsed using FTC; colors belonging to the same saturation mode are then grouped together.
- Finally, the intensity histogram of colors with similar hue and saturation (the ones belonging to a common mode in hue and saturation) is computed and its meaningful modes are extracted; colors belonging to the same mode are grouped together.

As a result of this process a small set of color groups is obtained. Each group is composed by colors belonging to a common mode in the hue, saturation and intensity histograms.

This algorithm is described in full detail at the end of this section, but some remarks need to be made before this description. First, why using HSI instead of any other color representation (e.g. RGB or CIE Lab)? The reasons are manifold:

- Hue, saturation and intensity are magnitudes that can be interpreted intuitively and have a physical meaning [44, 18]: the intensity is proportional to the energy sent back by the observed object, the hue indicates the dominant wave lengths and the saturation hints about the concentration of wavelengths.
- Red, blue and green histograms are highly correlated, whereas the intensity information is decoupled from the chrominance information represented as hue and saturation. Moreover, the hue and saturation components are intimately related to the way in which human beings perceive colors. For this reason HSI is usually the space chosen when developing algorithms based on the color sensing properties of the human visual system.
- Conversion formulas from RGB to HSI are straightforward (see below), while conversion from RGB to Lab depends on the scene illuminant.

A second point that needs to be clarified is the order of the color components in the hierarchical algorithm: why first hue, then saturation and finally illumination?

In this respect, we have followed the lines marked by other authors [8, 41] which consider the hue as the dominant component for color segmentation, due to its invariance properties with respect to changes in the direction and intensity of the incident light [17].

Finally, a technical problem related to the use of the HSI color system must be addressed. The conversion formulas between RGB and HSI are:

$$\begin{aligned} I &= \frac{R+G+B}{3} \\ S &= \sqrt{(R-I)^2+(G-I)^2+(B-I)^2} \\ H &= \arccos\left(\frac{(G-I)-(B-I)}{S\cdot\sqrt{2}}\right). \end{aligned}$$

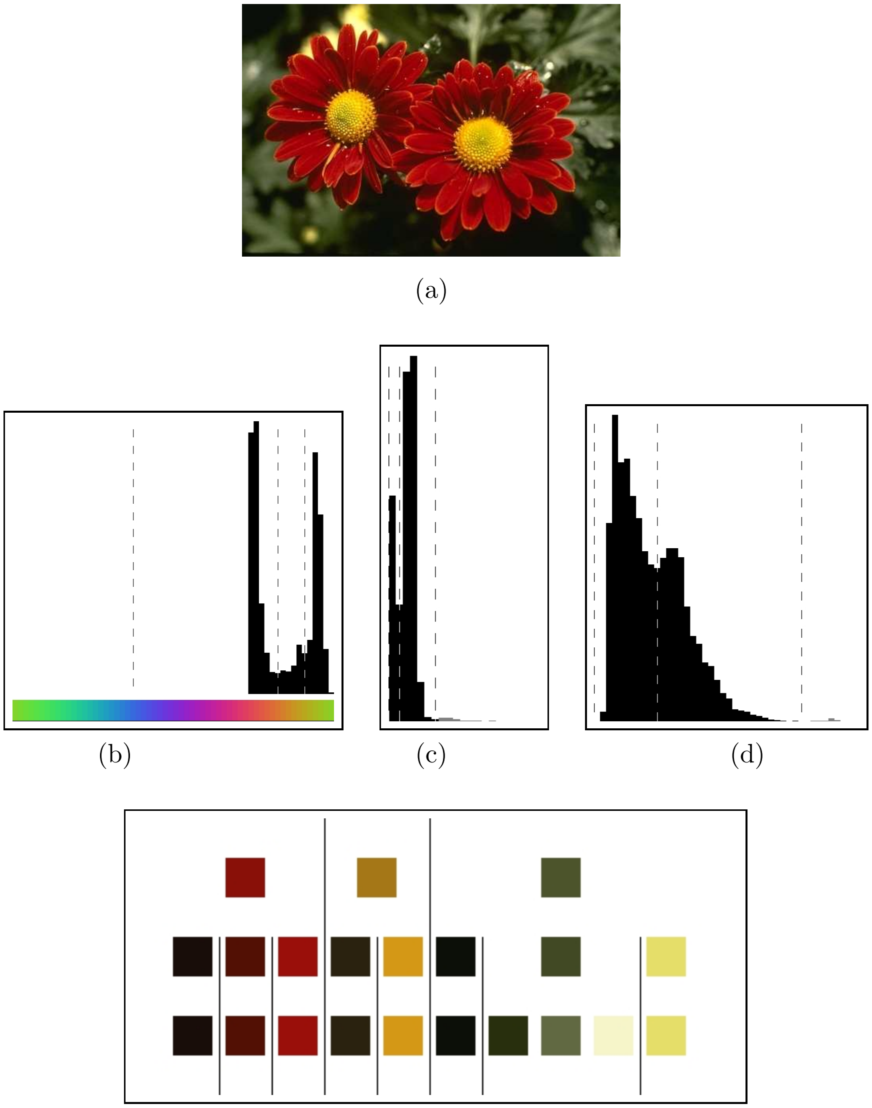
We can observe that the hue component is not defined for zero-saturated colors. These colors are located on the line from pure black to pure white in the RGB color cube (the so-called "grey axis"). In practice, and due to numerical limitations in the representation of the color components, this problem also arises when dealing with low saturated colors. Several authors [21, 29, 41] have remarked this problem and proposed different solutions which basically consist in classifying colors as "chromatic" or "achromatic" according to their saturation value. Only colors whose saturation is above a given threshold are considered as "chromatic" and are used in the computations involving hue values. This threshold is computed in different ways depending on the author. We follow a similar approach but we use a very simple argument to estimate this threshold: for a fixed intensity, at a distance S from the grey axis the maximum allowed number of color points is $2\pi S$, therefore, if we decide to quantize the hue component with Q different values S must be above $\frac{Q}{2\pi}$ to allow this quantization. This requirement defines a cylinder in the HSI color space that we call the *grey cylinder.* All the color points contained in the cylinder will be considered as achromatic.

The final version of the color grouping algorithm is as follows:

ACoPA (automatic color palette) Algorithm:

1. *Apply the FTC algorithm on the hue histogram of the image. Let S be the obtained segmentation.*
2. *Link each pixel of the grey cylinder to its corresponding interval $S_i = [s_i, s_{i+1}]$, according to its hue value.*
3. *For each i, construct the saturation histogram of all the pixels in the image whose hue belongs to S_i. Take into account the pixels of the grey cylinder. Apply the FTC algorithm on the corresponding saturation histogram. For each i, let $\{S_{i,1}, S_{i,2}, \ldots\}$ be the obtained segmentation.*
4. *For each i and each j, compute and segment the intensity histogram of all the pixels whose hue and saturation belong to S_i and $S_{i,j}$, including those in the grey cylinder.*

Observe that the number of colors obtained from the first step of the method increases when we add successively the saturation and intensity information. As a result of the algorithm we end up with a minimum set of color groups. The representative color of a given cluster can be chosen as the mean of the colors present in the cluster This set of colors forms the initial seed that will be used by the K-Means algorithm (see Appendix A) to obtain the final color palette.



(a)

(b) (c) (d)

(e)

Figure 4. Test 1 (I), example of the ACoPa algorithm: (a) Original image “Flower” (481 × 321 pixels). (b) Hue histogram, segmented into 3 modes. (c) Saturation histogram corresponding to the right-most mode of the hue histogram, segmented into 3 modes. (d) Intensity histogram corresponding to the central mode of the histogram in (c), segmented into 3 modes. (e) Hierarchical color palette: each row displays the colors obtained at each step of the algorithm.

It is worth noting that the hue histogram is circular, which means that the hue value 0° is identified with the hue value 360°. Then, the obtained modes of the hue

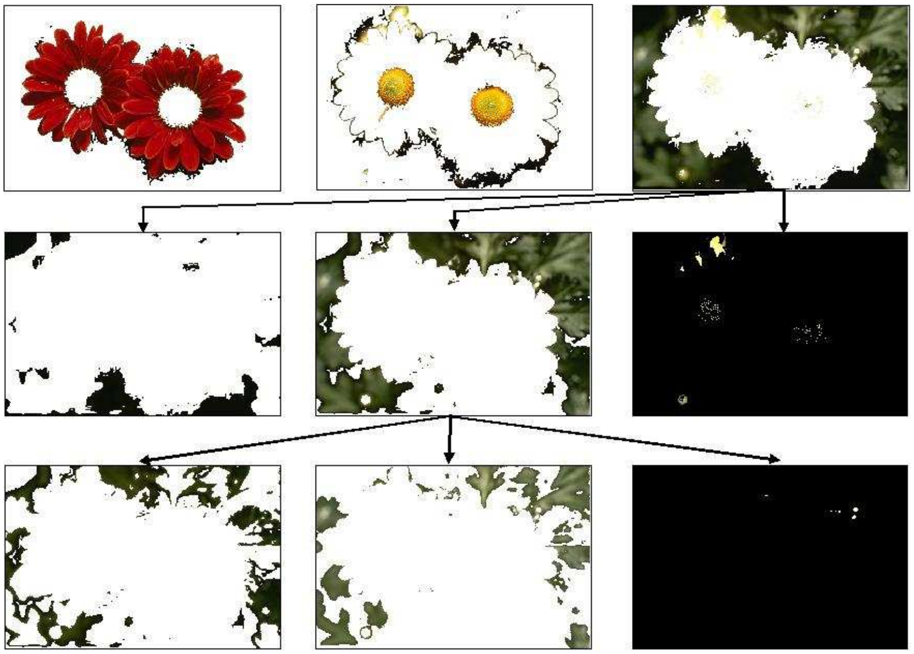


Figure 5. Test 1 (II): (Top) From left to right, pixels contributing to the left-most, central and right-most modes of the hue histogram of Fig. 4(b). (Middle) Idem for the saturation histogram of Fig. 4(c). (Bottom) Idem for the intensity histogram of Fig. 4(d). Pixels not contributing to the modes are set either to black or white.

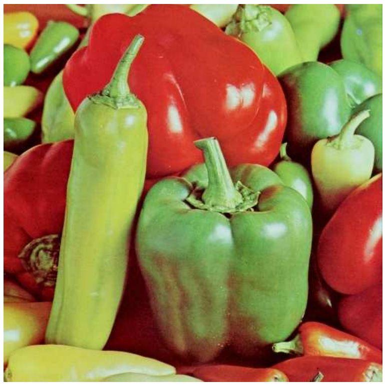
histograms are also circular. For example, in the hue histogram of Fig. 4(b), we can see one circular mode from 330° to 100°.

5. Naming colors

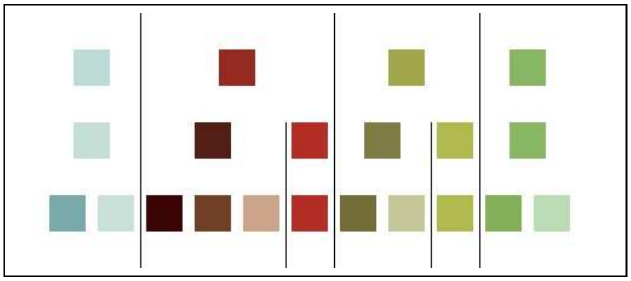
The ultimate goal of the proposed method is to qualitatively describe an image by associating a name to each color in the final palette. To achieve this goal, we seek for a dictionary of color names, with a good representation and distribution of the millions of existing colors and containing a not excessive but sufficient amount of them.

In the literature there are lots of color-names dictionaries (see for example [27]). We have limited our search to the ones published in the world wide web and we have considered two of them: the X11 rgb and NBS/ISCC Centroids dictionaries. X11 is the most replicated dictionary among the ones published online and its last modification took place in 1994. However, this dictionary presents some drawbacks: it severely under-represents the darkest octant of the RGB color cube and it presents some name conflicts. The NBS/ISCC Centroids is an improvement of a NBS/ISCC dictionary, which tries to be the ideal source for surface color names. The advantages of this last dictionary can be summarized into the following items:

- it has no name conflicts,
- it has colors distributed to equalize perceptual distances between them,



(a)



(b)

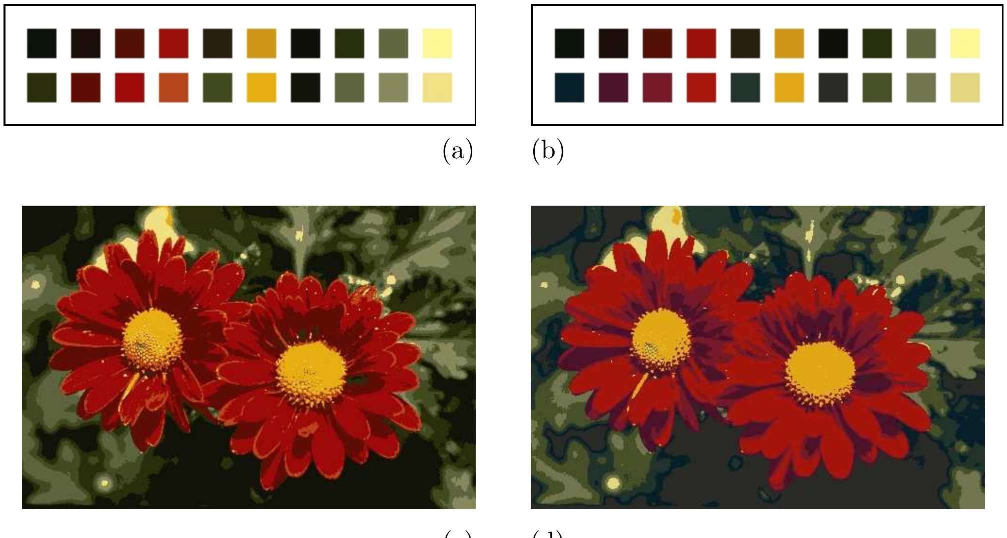
Figure 6. Test 2: (a) Original image "Peppers" (508×507 pixels). (c) Hierarchical color palette (11 colors).

- it is derived from matching physical samples.

The X11 dictionary presents 450 names of colors, and the NBS/ISCC Centroids presents 250. In both dictionaries each color name has an RGB code associated (the "color representative" of the name).

Independently of the used dictionary, the process that associates the color in the palette to its name in the dictionary is always the same. We use the Euclidean distance and the CIE Lab coordinates for measuring the difference between the representative color of the dictionary name and the colors in the palette. The reason for using Lab coordinates is that, in this space, Euclidean distances between colors are proportional to the perceptual distances between them. Thus, we will associate a color of the palette to the name of the dictionary whose representative color is at the minimal Euclidean distance, considering the CIE Lab coordinates of both colors. The final result is a list of names where the colors of the palette are represented by the associated name in the dictionary and the representative color of the name. The colors are ordered by the amount of pixels that contribute to each cluster. Thus, the first name in the list is the most frequent color in the image and the last one the less frequent.

The association of palette colors to color names usually leads to a reduction in the final number of colors needed to describe the image: if different colors in the



(a) (b)

(c) (d)

Figure 7. Test 3 (I): color palettes obtained using AcoPa + K-Means in RGB (a) and Lab spaces (b). The upper row displays the initial seeds and the lower the final colors in the palette (10 colors). The corresponding segmentations are shown in (c) and (d). Compare them to the original image in Fig. 4(a)

palette are associated to the same name, then the colors are merged into a common cluster and the name appears only once in the final list.

Some examples of the combination of the ACoPa+K-Means palette with the X11 and the NBS/ISCC Centroids dictionaries are presented in the next section.

6. Experimental results

We begin this section with a complete and detailed example of the ACoPa algorithm. In Fig. 4 we observe the original "Flower" image (Fig. 4(a)) and its hue histogram (Fig. 4(b)), which is segmented into three modes using the FTC algorithm. The saturation histogram of the colors contributing to each one of these modes is computed and also segmented. Fig. 4(c) shows the saturation histogram corresponding to the right-most mode of the hue histogram and its segmentation. The process is repeated for the intensity histograms of the colors belonging to a common mode in hue and saturation. Fig. 4(d) shows the intensity histogram corresponding to the central mode of the histogram in Fig. 4(c) and its segmentation. Fig. 5 displays, for each one of the modes in these histograms, the pixels in the original image whose color contributes to the mode (the rest of pixels are set either to black or white).

The mean RGB value of the colors contributing to the same mode is chosen as the color representative of the mode. Fig. 4(e) displays the color representatives of all the computed modes. The three rows of the image (from top to bottom) show the colors corresponding to the modes in the hue, saturation and intensity histograms, respectively. Observe how the number of colors increases as additional information is used to compute the modes: only hue information is used to compute

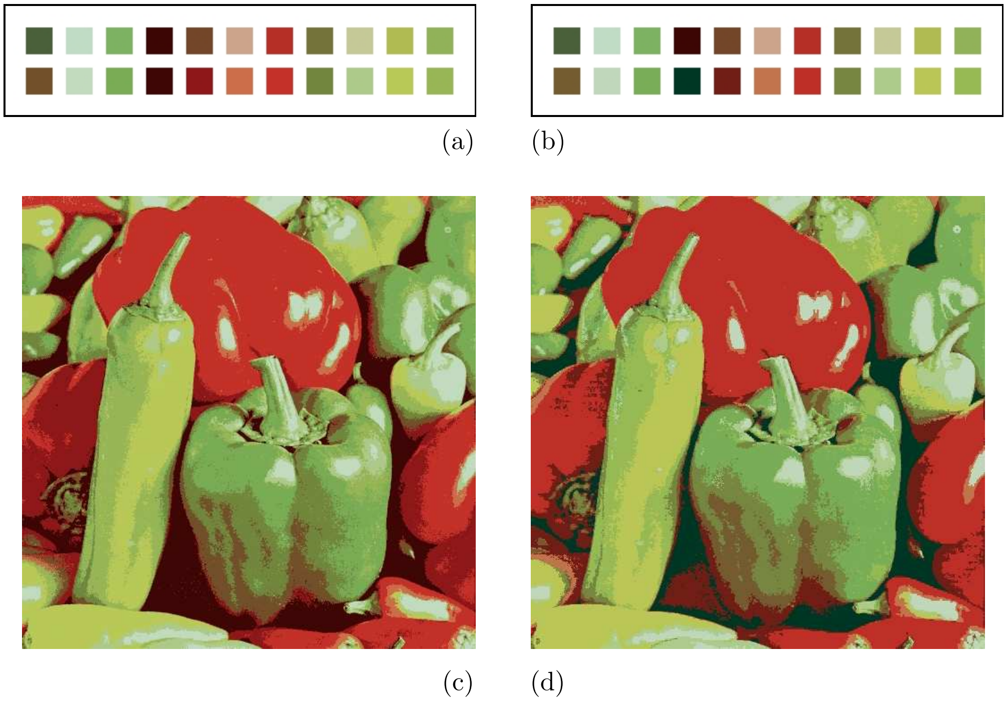


FIGURE 8. Test 3 (II): color palettes obtained using AcoPa + K-Means in RGB (a) and Lab spaces (b). The upper row displays the initial seeds and the lower the final colors in the palette (11 colors). The corresponding segmentations are shown in (c) and (d). Compare them to the original image in Fig. 6(a).

the first row (3 colors); hue and saturation are used in the second row (8 colors); hue, saturation and intensity are used to compute the third row (10 colors). For this reason, we call this representation “hierarchical color palette”. The colors in the last row are the ones that will be used as seeds in the K-Means method (see Appendix A) to obtain the final palette.

A second example of application of the AcoPa algorithm is displayed in Fig. 6. In this case only the original image and the hierarchical color palette are shown.

The third example displays the final color palette for the previous images, obtained by combining the result of ACoPa and the K-Means clustering method. As commented above, the colors in the last row of the hierarchical palettes are used as initial seeds of the K-Means method. Clustering has been performed in both RGB and CIE Lab spaces and the results are shown in Fig. 7 and Fig. 8. In both cases, the initial seeds and the final colors in the palette (the mean RGB colors of the final clusters) are displayed. The segmentations of the original images are also shown. These are obtained by replacing their original image colors by the mean RGB color of the cluster to which they belong. Observe the striking similarity between the original and segmented images (either when using RGB or Lab spaces), even if the latter are represented with a fraction of the original set of colors.

The next test compares the segmentations obtained with classical techniques for computing color palettes (Median Cut, Popularity algorithm, classical K-Means and Mean-Shift) with the results obtained with the proposed method (AcoPa + K-Means). K-Means and Mean-Shift have been tested using both RGB and Lab color spaces. Two images have been used in the tests and the results are shown in Fig. 9 and Fig. 11. In both cases the proposed method outperforms the classical ones, even if their parameters have been adjusted to obtain the same number of colors. Remind that this number of colors is automatically computed by AcoPa.

In the case of the "Ladybug" image (Fig. 9), only Mean-Shift and the proposed method are able to correctly detect the color of the ladybug. Mean-Shift, however, is unable to represent with the same fidelity both the small bug and the background leafs. The reason why most of the classical methods fail to find the red color is that it belongs to a very sparse cluster in RGB space (see Fig. 10(a)). This cluster forms however a small mode in the hue histogram of the image (see Fig. 10(b)) that the FTC algorithm is able to correctly detect.

Similar remarks can be made on the "Olive tree" image (Fig. 11). Only Mean-Shift correctly finds the T-shirts colors, however the landscape in the background is poorly segmented. On the other hand, Median-Cut, Popularity and classical K-Means miss some of the T-shirt colors, since they belong to small clusters in color space. By combining ACoPa with K-Means we obtain better results, but note that only when using the CIE Lab space both the small T-shirts and the landscape in the background are represented with the same fidelity. In general, it has been observed that the use of this color space improves the clustering results.

The last two figures show some experiments on combining the obtained palette with a dictionary of color names. The goal is to obtain a qualitative description of the image.

Fig. 12 displays the original image "Bicycle"(Fig. 12(a)), the final segmentation obtained using the ACoPa+K-Means algorithm in Lab space (Fig. 12(b)) and the hierarchical and final color palettes (Fig. 12(c), (d)). These colors are named according to the procedure described in Section 5 and the results are shown in Fig. 12(d) and Fig. 12(e) (X11 and NBS/ISCC Centroids dictionaries respectively). In these images, the left column corresponds to the palette color, the middle column is its name and the last column displays the representative color of the name in the dictionary.

This example illustrates the reduction in the number of colors induced by the use of a color dictionary. From the initial 17 colors in the palette, just 12 different names are found in the X11 dictionary (Fig. 12(d)) while 16 names are assigned when using the NBS/ISCC Centroids (Fig. 12(e)).

The final lists of names obtained using each one of the dictionaries are different since the names are different, but the associated representative colors are similar. It can be observed in the last example (Fig. 13) however that the X11 dictionary severely under-represents the darkest octant of the RGB color cube. In this figure the lists of color names obtained for the "Flower" image (Fig. 4(a)) are displayed. As a result of this under-representation the three dark colors in the palette are represented by only one name (black). On the other hand, when using the NBS/ISCC Centroids dictionary these colors are represented by three different names (greenishblack, oliveblack and brownishblack).

We can also estimate the adequacy of each dictionary to represent the colors by comparing the representative color in the dictionary with the original color in the palette. Ideally they should be very similar, however this is not always true (see e.g. the second color in the list in Fig. 13(a)).

In general, the reduction in the number of colors is more important when using the X11 dictionary, but the assigned color representatives are not always correct. For this reason we can conclude that the NBS/ISCC dictionary is better suited for our application: it permits a reduction of the colors in the palette while allowing a qualitative description of an extensive gamut of colors.

7. Conclusions

We have presented in this paper a method to automatically estimate the significant colors of an image. This set of colors is inferred from a statistical analysis of the Hue, Saturation and Intensity histograms of the image colors and has been used as the initial seed for the classical K-Means clustering method to obtain a minimum set of colors representing the image (the so-called "color palette"). The performed tests show the capacity of the method to represent with fidelity any color image with a small set of colors (usually less than 30). These results seem to endorse the idea that a statistical analysis of physical variables may be used to assert the perceptual reality.

Acknowledgments

The authors gratefully thank F. Mettini for her kind authorization to use the "Ladybug" image.

Appendix A. K-Means clustering algorithm in RGB or CIE Lab space

Given a cloud of points in RGB space (e.g. the RGB values of the pixels of a color image) and a set of K RGB points or "initial seeds" $(R_1, G_1, B_1), \cdots, (R_K, G_K, B_K)$, do:

1. Compute the Euclidean distance from all the colors in the cloud to each one of the seeds.
2. Associate each color point to its closest seed. A **cluster** is defined as the set of points associated to the same seed. The number of clusters is therefore K.
3. For each cluster, compute the mean RGB value of the colors in the cluster. Use this value as the new seed of the cluster.
4. If, for all the clusters, the distance between the new and old seeds is small (below some user-defined threshold, e.g. 1) then STOP. Else return to step 1.

The same algorithm can be applied in CIE Lab space just by replacing the RGB components by their corresponding Lab values (conversion formulas can be found in [44]).

Appendix B. Pool Adjacent Violators

We call $\mathcal{D}(L)$ the space of all decreasing densities on $\{1, 2, ..., L\}$ and $\mathcal{P}(L)$ the space of probability distributions on $\{1, 2, ..., L\}$, *i.e.* the vectors $r = (r_1, ..., r_L)$ such that:

$$\forall i \in \{1, 2, ..., L\}, \quad r_i \geq 0 \qquad \text{and} \qquad \sum_{i=1}^{L} r_i = 1.$$

The algorithm "Pool Adjacent Violators" (see [2, 5]) performs a simple operation: if r is increasing on successive bins, r is replaced on these bins by its local mean value and the procedure is repeated.

Pool Adjacent Violators

Let $r = (r_1, ..., r_L) \in \mathcal{P}(L)$ *be a normalized histogram. We consider the operator* $D : \mathcal{P}(L) \to \mathcal{P}(L)$ *defined by: for* $r \in \mathcal{P}(L)$*, and for each interval* $[i, j]$ *on which* r *is increasing,* i.e. $r_i \leq r_{i+1} \leq ... \leq r_j$ *and* $r_{i-1} > r_i$ *and* $r_{j+1} < r_j$*, we set*

$$D(r)_k = \begin{cases} \frac{r_i + ... + r_j}{j-i+1} & \text{for } \ k \in [i, j] \\ r_k & \text{otherwise.} \end{cases}$$

This operator D *replaces each increasing part of* r *by a constant value (equal to the mean value on the interval).*

After a finite number M *of iterations of* D*,* $M < L$*, we obtain a decreasing distribution denoted by* $\overline{r}$*:*

$$\overline{r} = D^L(r).$$

This distribution $\overline{r}$ *is exactly the decreasing Grenander estimator of* r*.*

References

[1] Al-Daoud, M., *A new algorithm for cluster initialization*, Transactions on Engineering, Computing and Technology , **4** (2005), 74–76.

[2] Ayer, M., Brunk, H., Ewing, G., Reid, W., and Silverman, E., *An empirical distribution function for sampling with incomplete information*, The Annals of Mathematical Statistics 26, **4** (1955), 641–647.

[3] Babu, G., and Murty, M., *A near-optimal initial seed value selection in k-means algorithm using a generic algorithm*, Pattern Recognition Letters, **14** (1993), 763–769.

[4] Bezdek, J., "Pattern Recognition With Fuzzy Objective Function Algorithms," Plenum Press, 1981.

[5] Birgé, L., *The Grenander estimator: A nonasymptotic approach*, The Annals of Statistics, **17** (1989), 1532–1549.

[6] Bradley, P., and Fayyad, U., *Refining initial points for k-means clustering*, Proceedings of the Fifteenth International Conference on Machine Learning, (1998), 91–99.

[7] Chen, T., Murphey, Y., Karlsen, R., and Gerhart, G., *Color image segmentation in color and spatial domain*, Lecture Notes in Computer Science, **2718** (2003), 72–82.

[8] Cheng, H., and Sun, Y., *A hierarchical approach to color image segmentation using homogeneity*, IEEE Transactions on Image Processing 9, **12** (2000), 2071–2082.

[9] Comaniciu, D., and Meer, P., *Mean shift: A robust approach toward feature space analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence 5, **24** (2002), 603–619.

[10] Delon, J., Desolneux, A., Lisani, J. L., and Petro, A. B., *A non-parametric approach for histogram segmentation*, to appear in IEEE Transactions on Image Processing.

[11] Dempster, A., Laird, N., and Rubin, D., *Maximum likelihood from incomplete data via EM algorithm*, Journal of the Royal Statistical Society, Series B, **39** (1977), 1–38.

[12] Desolneux, A., Moisan, L., and Morel, J.-M., *A grouping principle and four applications*, IEEE Transactions on Pattern Analysis and Machine Intelligence 25, **4** (2003), 508–513.

[13] Everitt, B., "Cluster Analysis," Halsted Press, 1974.

[14] Fisher, D., *Knowledge acquisition via incremental conceptual clustering*, Machine Learning, **2** (1987), 139–172.

[15] Forgy, E., *Cluster analysis of multivariate data: Eficiency vs. interpretability of classifications*, Biometrics, **21** (1965), 768.

[16] Gervautz, M., and Purgathofer, W., "A Simple Method for Color Quantization: Octree Quantization," Academic Press Professional, Inc., 1990, 287–293.

[17] Gevers, T., and Smeulders, A., *Color-based object recognition*, Pattern Recognition, **32** (1999), 453–464.

[18] Gonzalez, R., and Woods, R., "Digital Image Processing," Addison Wesley, 1993.

[19] Grenander, U., "Abstract Inference," Wiley, New York, 1980.

[20] Grira, N., Crucianu, M., and Boujemaa, N., *Unsupervised and semi-supervised clustering: a brief survey*, in "A Review of Machine Learning Techniques for Processing Multimedia Content," MUSCLE European Network of Excellence, 2004.

[21] Hanbury, A., *Circular statistics applied to colour images*, in 8th Computer Vision Winter Workshop, (2003).

[22] Heckbert, P., "Color Image Quantization for Frame Buffer Display," B.s. thesis, Architecture Machine Group, MIT, Cambridge, Mass., 1980.

[23] Higgs, R., Bemis, K., Watson, I., and Wikel, J., *Experimental designs for selecting molecules from large chemical databases*, J. Chem. Inf. Comput. Sci., **37** (1997), 861–870.

[24] Hubel, D., "Eye, Brain and Vision," Paperback, 2000.

[25] Jain, Murty, and Flynn, *Data clustering: A review*, ACM Computing Surveys, **31** (1999), 264–323.

[26] Kaufman, L., and Rousseeuw, P. J., "Finding Groups in Data: an Introduction to Cluster Analysis," John Wiley and Sons, 1990.

[27] Kelly, K., and Judd, D., "Color: Universal Language and Dictionary of Names," National Bureau of Standarts, 1976.

[28] King, B., *Step-wise clustering procedures*, J. Am. Stat. Assoc, **69** (1967), 86–101.

[29] M. Tico, T. Haverinen, P. K., *A method of color histogram creation for image retrieval*, in "Nordic Signal Processing Symposium," 2000, 157–160.

[30] MacQueen, J., *Some methods for classification and analysis of multivariate observations*, in "5th Proceedings of the Symposium on Mathematics and Probability," (1967).

[31] Milligan, G., and Cooper, M., *An examination of procedures for determining the number of clusters in a data set*, Psychometrika , **50** (1985), 159–179.

[32] Murtagh, F., *A survey of recent advances in hierarchical clustering algorithms which use cluster centers*, Comput. J., **26** (1984), 354–359.

[33] Omer, I., and Werman, M., *Color lines: Image specific color representation*, in "Proceedings of the IEEE CVPR," (2004).

[34] Park, S., Yun, D., and Lee, S., *Color image segmentation based on 3-d clustering: morphological approach*, Pattern Recognition, **31** (1998), 1061–1076.

[35] Plataniotis, K., and Venetsanopoulos, A., "Color Image Processing and Applications," Springer, 2000.

[36] Sander, J., Ester, M., Kriegel, H.-P., and Xu, X., *Density-based clustering in spatial databases: The algorithm gdbscan and its applications*, Data Min. Knowl. Discov., **2** (1998), 169–194.

[37] Schettini, R., Ciocca, G., and Zuffi, S., *A survey of methods of colour image indexing and retrieval in image database*, in "Color Imaging Science: Exploiting Digital Media," John Wiley & Sons, 2002, 183–211.

[38] Snarey, M., Terrett, N., Willet, P., and Wilton, D., *Comparison of algorithms for dissimilarity-based compound selection*, J. Mol. Graphics and Modelling, **15** (1997), 372–385.

[39] Sneath, P., and Sokal, R., "Numerical Taxonomy," Freeman, 1973.

[40] Trémeau, A., Fernandez-Maloigne, C., and Bonton, P., "Image numérique couleur, de l'acquisition au traitement," Dunod, 2004.

[41] Vadivel, A.and Mohan, M., Sural, S., and Majumdar, A., *Segmentation using saturation thresholding and its application in content-based retrieval of images*, Lecture Notes in Computer Science, **3211** (2004), 33–40.

[42] Ward, J., *Hierarchical grouping to optimize an objective function*, J. Am. Stat. Assoc, **58** (1963), 236–244.

[43] Weeks, A., and Hague, G., *Color segmentation in the hsi color space using the k-means algorithm*, in "Proceedings of the SPIE - Nonlinear Image Processing VIII," (1997), 143–154.

[44] Wyszecki, G., and Stiles, W., "Color Science: Concepts and Methods, Quantitative Data and Formulae," Wiley, 1984, ch. 3.

Received for publication September 2006.

E-mail address: julie.delon@enst.fr
E-mail address: desolneux@math-info.univ-paris5.fr
E-mail address: joseluis.lisani@uib.es
E-mail address: anabelen.petro@uib.es

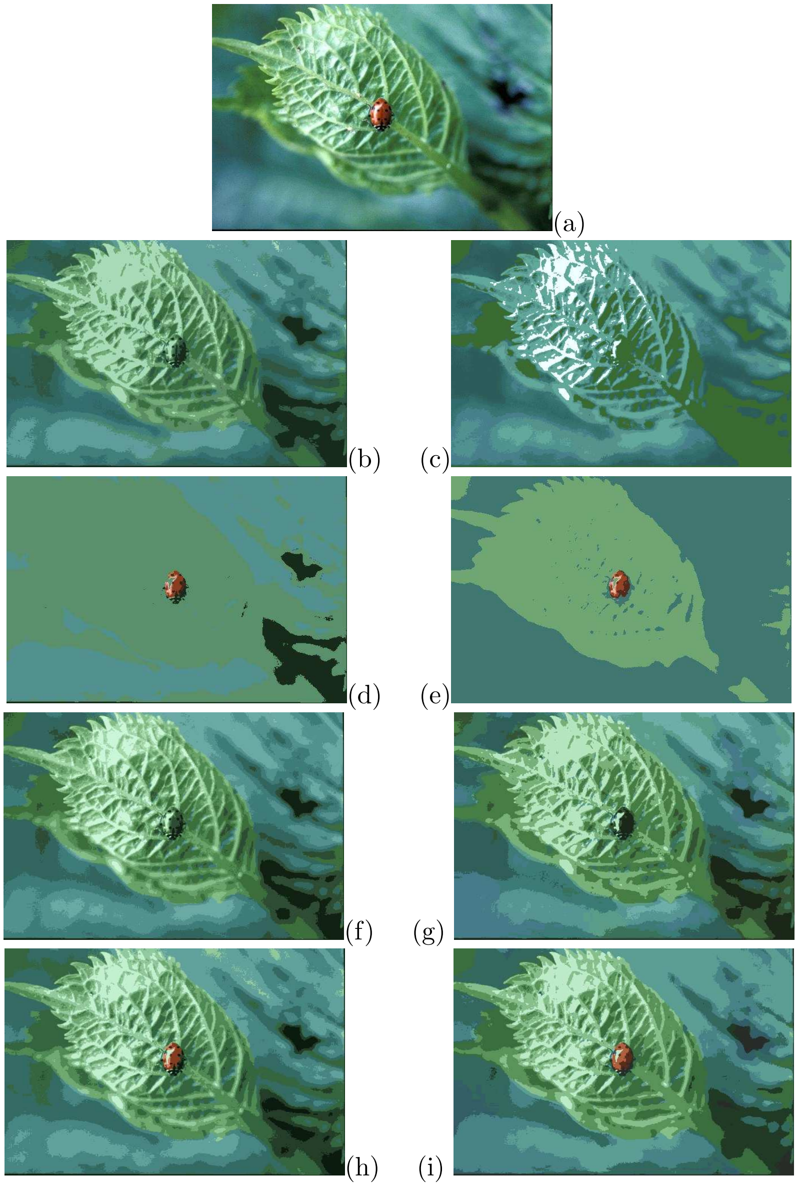
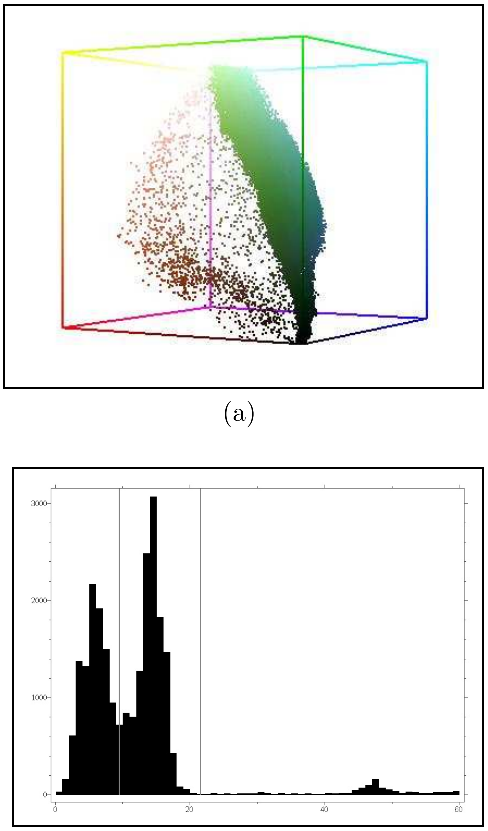


Figure 9. Test 4 (I): comparison of the segmentations resulting from different clustering methods. The parameters of the methods have been adjusted to obtain 12 colors. This value has been automatically estimated using ACoPa. (a) Original image "Ladybug" (600 × 400 pixels). (b) Median Cut. (c) Popularity. (d) Mean-Shift in RGB space (random seeds). (e) Mean-Shift in Lab space (random seeds). (f) K-Means in RGB space (random seeds). (g) K-Means in Lab space (random seeds). (h) ACoPa + K-Means in RGB space. (i) ACoPa + K-Means in Lab space.



(a)

(b)

Figure 10. Test 4 (II): (a) RGB space representation of the pixels corresponding to the “Ladybug” image (Fig. 9(a)). (b) Hue histogram of the same image. The red color corresponding to the ladybug represents a little mode in the hue histogram and a sparse cluster in RGB space.

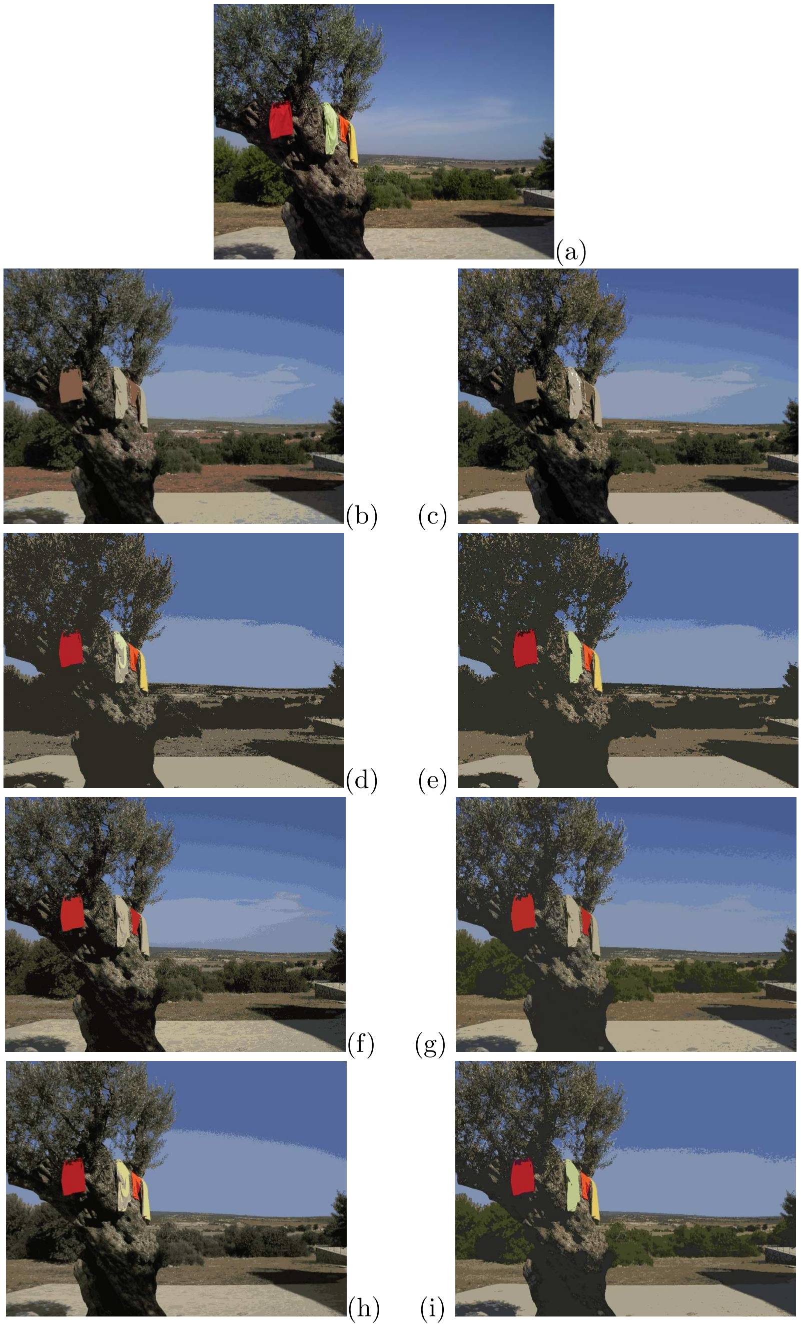
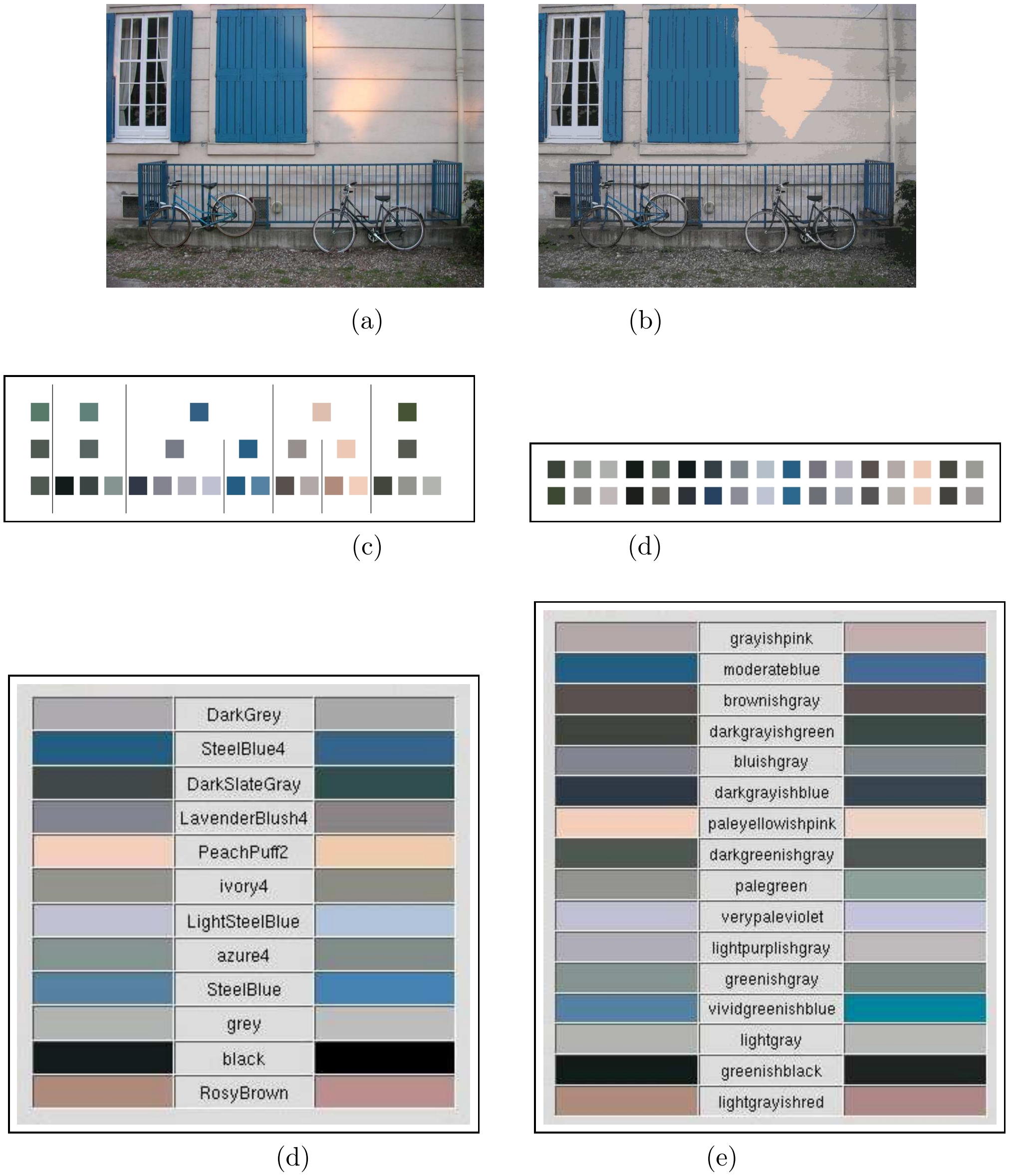


Figure 11. Test 4 (III): comparison of segmentations obtained with different clustering methods. The parameters of the methods have been adjusted to obtain 17 colors (value automatically estimated with ACoPa). (a) Original image "Olive tree" (779×584). (b) Median Cut. (c) Popularity. (d) Mean-Shift (RGB space, random seeds). (e) Mean-Shift (Lab space, random seeds). (f) K-Means (RGB, random seeds). (g) K-Means (Lab, random seeds). (h) ACoPa + K-Means (RGB). (i) ACoPa + K-Means (Lab).



(a) (b)

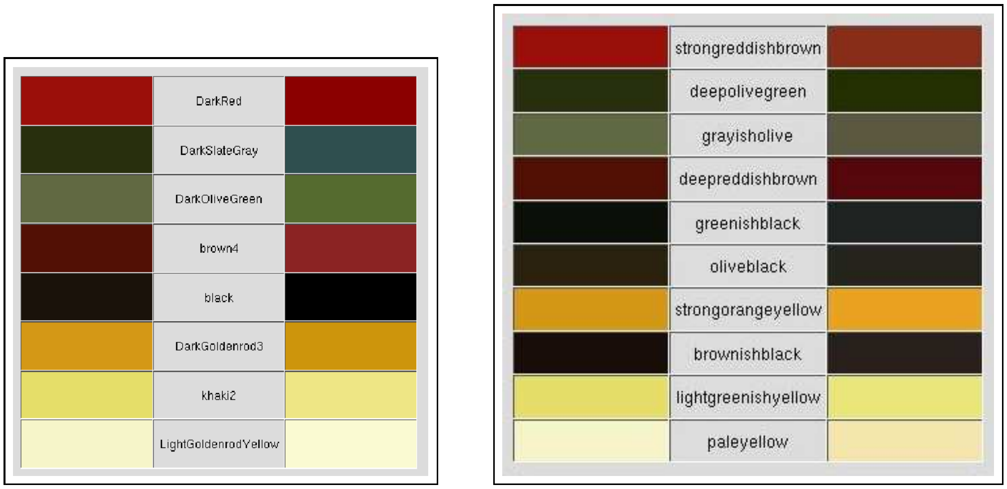
(c) (d)

DarkGrey
SteelBlue4
DarkSlateGray
LavenderBlush4
PeachPuff2
ivory4
LightSteelBlue
azure4
SteelBlue
grey
black
RosyBrown

grayishpink
moderateblue
brownishgray
darkgrayishgreen
bluishgray
darkgrayishblue
paleyellowishpink
darkgreenishgray
palegreen
verypaleviolet
lightpurplishgray
greenishgray
vividgreenishblue
lightgray
greenishblack
lightgrayishred

(d) (e)

Figure 12. Test 5 (I): naming colors. Comparison of the X11 and NBS/ISCC dictionaries. (a) Original image "Bicycle" (800 × 600 pixels). (b) Segmentation using ACoPa+ K-Means (Lab). (c) Hierarchical color palette. (d) Final palette (17 colors). (d) List of names using the X11 dictionary (12 names). (e) List of names using the NBS/ISCC dictionary (16 names).



(a) (b)

FIGURE 13. Test 5 (II): naming colors. Comparison of the X11 and NBS/ISCC dictionaries for image in Fig. 4(a). The number of colors in the final palette is 10. (a) List of names using the X11 dictionary (8 names). (b) List of names using the NBS/ISCC dictionary (10 names).