Unsupervised Low-Frequency Driven Segmentation of Color Images

L. Lucchese[†‡] and S.K. Mitra[†]

[†]Dept. of Electrical and Computer Eng., University of California, Santa Barbara

[‡]Dept. of Electronics and Informatics, University of Padua, Italy

{luca,mitra}@iplab.ece.ucsb.edu

Abstract

This paper presents an algorithm for unsupervised segmentation of color images. The main idea behind it is the use of the low-frequency content of images which allows for smoothing of segments and sharpening of histograms of color attributes. Our algorithm handles images in a palettized format and operates in the feature space constituted by the cylindrical representation of the $L^*u^*v^*$ color space. Within such space, it finds representative colors by determining first the main hue families, through histogram thresholding, and then the main clusters on planes at constant hue, by means of k-means clustering. Two examples of the practical performance of the algorithm are reported and discussed.

1 Introduction

The human visual system performs extraordinarily well in distinguishing and recognizing the various objects portrayed in images. However, for computers it is very difficult to recognize objects in images, even with simple scenes. In image processing and computer vision, segmentation is the low-level operation which consists in partitioning an image into disjoint and homogeneous regions, which should be meaningful for certain applications, before higher-level procedures such as object recognition and semantic interpretation are called for.

Until a few years ago, segmentation techniques were proposed mainly for gray-level images since for a long time these have been the only kind of visual information that acquisition devices were able to take and computer resources to handle. Rather comprehensive surveys on techniques for segmentation of gray-level images can be found in [1]-[4]. In this last decade, there has been a remarkable growth of algorithms for segmentation of color images. Among others, two phenomena in particular have triggered such an interest: the recent and fast evolution of the Internet and the parallel development of digital libraries and large databases of images which have been gathering an impressive amount of visual information. Color represents indeed an effective means for indexing and organizing this kind of information.

Several techniques have recently been contributed to the literature of color segmentation [5, 6, 7]. Most of the times, these are kind of "dimensional extensions" of techniques devised for gray-level images; thus, they exploit the well-established background laid down in that field. In other cases, they are *ad hoc* techniques tailored on the particular nature of color information and on the physics of the interaction of light with colored materials. Basically, the techniques for color image segmentation may be divided into three main categories [7]: 1) feature-space based techniques; 2) image-domain based techniques; and 3) physics based techniques.

In this paper, we propose a feature-space based technique for color segmentation which significantly extends and improves the algorithmic ideas originally devised in [10] and [11]. Our method has the following characteristics: i) works with color palettes; ii) exploits the low-frequency content of color; and iii) finds representative colors in the cylindrical representation of the (CIE) $L^*u^*v^*$ color space [8, 9] by determining first the main hue families and then the main clusters on planes at constant hue.

This paper is organized as follows. Section 2 discusses the various steps of the proposed segmentation algorithm and reports on some experimental results. Section 3 draws the conclusions.

2 Segmentation algorithm

2.1 Color representation

Color images are usually represented and handled in RGB coordinates; in this format, an image $\boldsymbol{\mathcal{F}}$ of size $M \times N$ may be represented as $\boldsymbol{\mathcal{F}} = \{\boldsymbol{\mathcal{R}}^{(\mathcal{F})}, \boldsymbol{\mathcal{G}}^{(\mathcal{F})}, \boldsymbol{\mathcal{B}}^{(\mathcal{F})}\}$, *i.e.*, as a set of three $M \times N$ matrices respectively con-

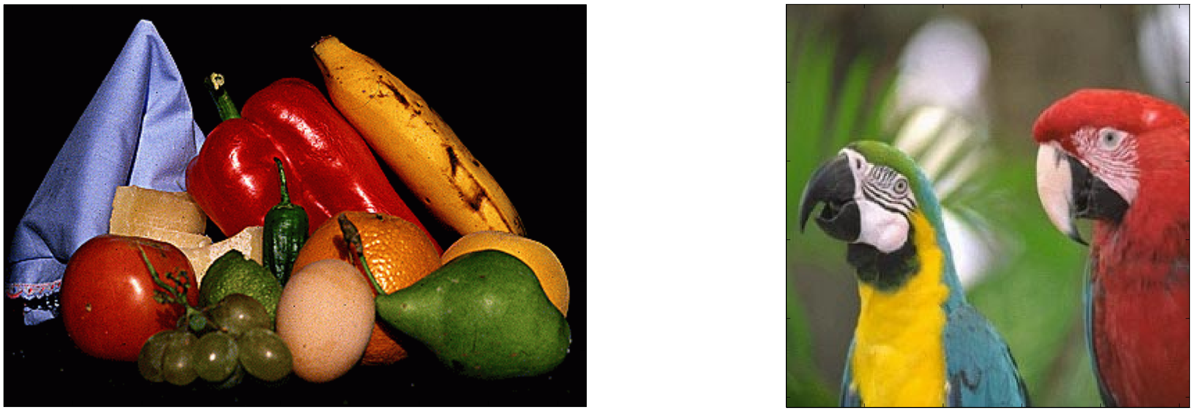

Figure 1. Two examples of color images $\boldsymbol{\mathcal{F}}$ with $C = 256$ and $C = 1024$ colors (left and right, respectively).

taining the red, green, and blue components of $\boldsymbol{\mathcal{F}}$. This image can be also conveniently represented[1] in a palettized format as $\boldsymbol{\mathcal{F}} = \{\boldsymbol{\mathcal{Q}}^{(\mathcal{F})}, \boldsymbol{\mathcal{P}}^{(\mathcal{F})}\}$, where $\boldsymbol{\mathcal{Q}}^{(\mathcal{F})} \in \mathbb{N}^{M\times N}_{[1,C]}$ is a matrix of pointers to a look-up-table of colors $\boldsymbol{\mathcal{P}}^{(\mathcal{F})} \in \mathbb{R}^{C\times 3}_{[0,1]}$, C being the number of possible combinations of the entries of the three color matrices $\boldsymbol{\mathcal{R}}^{(\mathcal{F})}, \boldsymbol{\mathcal{G}}^{(\mathcal{F})}, \boldsymbol{\mathcal{B}}^{(\mathcal{F})} \in \mathbb{R}^{M\times N}_{[0,1]}$. The color palette $\boldsymbol{\mathcal{P}}^{(\mathcal{F})}$ may be structured as $\boldsymbol{\mathcal{P}}^{(\mathcal{F})} = [\mathbf{R}^{(\mathcal{F})}\ \mathbf{G}^{(\mathcal{F})}\ \mathbf{B}^{(\mathcal{F})}]$, where $\mathbf{R}^{(\mathcal{F})}$, $\mathbf{G}^{(\mathcal{F})}$, $\mathbf{B}^{(\mathcal{F})} \in \mathbb{R}^{C}_{[0,1]}$ are, respectively, the vectors containing the red, green, and blue coordinates of the possible combinations of colors within the image $\boldsymbol{\mathcal{F}}$.

As an example, Fig. 1 shows two color images $\boldsymbol{\mathcal{F}}$ having $C = 256$ and $C = 1024$ colors (left and right, respectively).

2.2 Low-pass filtering

The first step of our algorithm consists in applying a low-pass filtering to the image $\boldsymbol{\mathcal{F}}$ in order to smooth it and emphasize this way the main color families within the image. In particular, a low-pass filtered version $\boldsymbol{\mathcal{L}}$ of $\boldsymbol{\mathcal{F}}$ is obtained as $\boldsymbol{\mathcal{L}} \doteq \boldsymbol{\mathcal{F}} * \boldsymbol{\mathcal{H}} = \{\boldsymbol{\mathcal{R}}^{(\mathcal{F})} * \boldsymbol{\mathcal{H}}, \boldsymbol{\mathcal{G}}^{(\mathcal{F})} * \boldsymbol{\mathcal{H}}, \boldsymbol{\mathcal{B}}^{(\mathcal{F})} * \boldsymbol{\mathcal{H}}\}$, where $\boldsymbol{\mathcal{H}}$ is a 5×5 binomial filter [12], chosen for its isotropy properties, and "$*$" denotes 2D discrete convolution. Let $\boldsymbol{\mathcal{L}} = \{\boldsymbol{\mathcal{Q}}^{(\mathcal{L})}, \boldsymbol{\mathcal{P}}^{(\mathcal{L})}\}$ be the palettized representation of $\boldsymbol{\mathcal{L}}$, where $\boldsymbol{\mathcal{P}}^{(\mathcal{L})}$ has size $C' \times 3$.

It is easy to figure out that the number of colors C' is much bigger (a few orders of magnitude!) than the number of colors C due to the averaging operation; in principle, there might exist as many colors in $\boldsymbol{\mathcal{L}}$ as the number of image pixels, namely MN. At first glance, this increased number of colors may appear to be a computational drawback, but, as a matter of fact, it is the key idea of our algorithm and some simple provisions can be adopted to achieve computational efficiency in handling huge amounts of colors.

Some authors have already resorted to low-pass filtering of images to obtain smooth regions in segmentation applications (see references in [6] and [7]); however, the substantial improvement that can be achieved in histogram based segmentation by working with the low-pass content of color images has not been fully exploited yet. This concept is clearly pointed out in [2] where the authors discuss about the sharpening effect, produced by local averaging, of the histogram peaks of certain features of gray level images. An analogous histogram sharpening of color attributes, such as the hue-angle used in this work, is determined by the low-pass filtering of color images; in fact, this operation evens out color differences that, for various reasons, may normally arise among pixels belonging to perceptively uniform color patches within an image. Accordingly, a low-pass smoothing dampens local color fluctuations, and reduces the variability of colors, while preserving their mean values. The general effect is that of generating a great many new colors, somehow nuances of the original colors, which perceptively cluster towards them; this fact can be appreciated very well from a geometric point of view in whatever color space and, perceptually, by comparing the color palette of an image with that of its low-pass filtered version. As a counterpart though, the low-pass filtering introduces false and spurious colors nearby the borders of adjacent uniform regions; but, statistically, they are much less than those belonging to the regions and do not represent an impairment to our procedure.

2.3 Hue-angle histogram

The RGB color space allows for a very straightforward representation of colors but, unfortunately, it has a Riemannian nature; this means that it is not a uniform space and perceived differences among colors

[1] Throughout the text, notation $\boldsymbol{A} \in \mathbb{S}^{M\times N}_{[\alpha,\beta]}$ indicates that each element a_{ij} of the matrix $\boldsymbol{A} \in \mathbb{S}^{M\times N}$ takes in values within the set $[\alpha,\beta] \subset \mathbb{S}$.

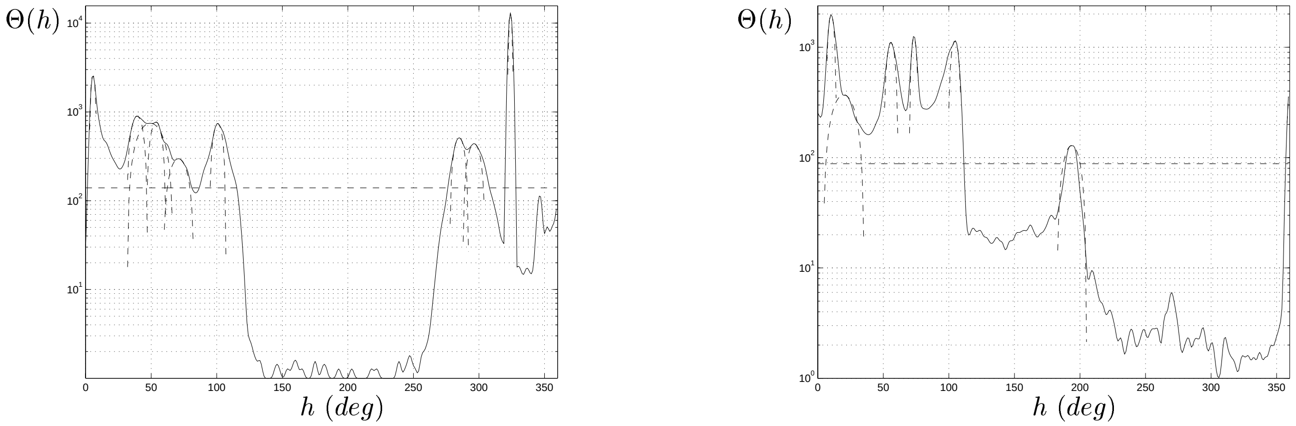

Figure 2. Semilogarithmic plots of the hue-angle histograms $\Theta(h)$ ordinately associated with the two color images of Fig. 1. The horizontal dashed lines represent the threshold τ_H and the curved dashed lines the parabolas $p = f_s(h)$ fitting the histograms at h_s, $s = 1, \ldots, S$.

can be assessed only by means of complicated metrics. Therefore, in order to have at disposal a simple measure for evaluating perceptive distances, we have chosen the uniform color space CIE $L^*u^*v^*$ [8, 9], which is an orthogonal Cartesian coordinate system endowed with the simple Euclidean metric $||(L^*, u^*, v^*)|| = \sqrt{(L^*)^2 + (u^*)^2 + (v^*)^2}$.

In this paper, we resort to the representation in cylindrical coordinates hcL^* of the $L^*u^*v^*$ color space, where the coordinates h and c are respectively called hue-angle and chroma and defined as $h = \arctan(v^*/u^*)$ and $c = \sqrt{(u^*)^2 + (v^*)^2}$.

The colors of the palette $\boldsymbol{\mathcal{P}}^{(\mathcal{L})}$ are mapped into the hcL^* space and the histogram $\Theta(h)$ is built with respect to their hue-angles h's; this also requires the knowledge of the number of pixels per each color entry of $\boldsymbol{\mathcal{P}}^{(\mathcal{L})}$ and such a piece of information is easily derived from the matrix $\boldsymbol{\mathcal{Q}}^{(\mathcal{L})}$. The histogram $\Theta(h)$ is then low-pass filtered to obtain a smooth profile and normalized with respect to its area in such a way that it can also be regarded as a hue-angle probability function. Fig. 2 shows the hue-angle histograms (in semilogarithmic plots to compress the vertical range) relative to the images of Fig. 1.

Hue is the most important attribute of color [8, 9]; thus, the most prominent peaks of the hue-angle histogram $\Theta(h)$ correspond to the main "color families" (*e.g.*, the reds, the oranges, the yellows, and so forth) of an image. These dominant color families are clearly noticeable in the plots of Fig. 2.

We then determine a threshold τ_H such that a pre-established percentage of the area of the histogram $\Theta(h)$ is above τ_H (in the examples reported in this paper we have set $\tau_H = 80\%$). Only the peaks (relative maxima) above τ_H are retained and fitted with parabolas; this gives an easy way of estimating the curvature, and thus the aperture, of the peaks. Let $\{h_s\}_{s=1,\ldots,S}$ be the set of hue-angles corresponding to the maxima of the histogram $\Theta(h)$ satisfying $\Theta(h_s) \geq \tau_H$, $s = 1, \ldots, S$, and let $p = f_s(h) = a_s h^2 + b_s h + c_s$, $s = 1, \ldots, S$, be the parabolas fitting in the least-squares sense the function $\Theta(h)$ in neighborhoods of points h_s (such parabolas are drawn with dashed lines in Fig. 2). We then define $\Delta h_s \propto 1/|a_s|$, $s = 1, \ldots, S$, which is a quantity proportional to the radius of curvature at h_s.

2.4 Search for clusters in planes at constant hue-angle

The next step consists in spanning the space hcL^* along cylindrical sectors centered at angles $\{h_s\}_{s=1,\ldots,S}$ and respectively having angular apertures $\{\Delta h_s\}_{s=1,\ldots,S}$. All the points within each cylindrical sector are projected onto the plane Π_s spanned by the coordinates (c, L^*) and associated with the central hue-angle h_s. According to the criterion defined in the previous section, the greater the aperture, and accordingly the radius of curvature, of a peak of the histogram $\Theta(h)$ in correspondence of h_s, the wider the angle Δh_s and therefore the number of colors (points) projected onto the plane Π_s.

The low-pass filtering produces a few very prominent clusters within the planes Π_s; these clusters can be easily found through an *ad hoc* implementation of the *k-means* algorithm [13, 14] which is detailed in Fig. 3.

It can be readily proven that, to a broad extent, the final number of segments does not depend on the threshold τ_H, but it does depend on the two thresholds τ_1 and τ_2 of the clustering procedure of Fig. 3.

```
for $s = 1$ to $s = S$
  ▷ define $\Pi_s \doteq \{(c, L^*) : h_s - \Delta h_s/2 \leq h \leq h_s + \Delta h_s/2\}$; let $D_s$ be the number of points $(c, L^*)$ belonging to $\Pi_s$ which are disposed in two vectors $\boldsymbol{c}_s, \boldsymbol{\ell}_s \in \mathbb{R}_+^{D_s}$;
  ▷ compute the normalized distribution vector $\boldsymbol{w}_s \in \mathbb{Q}^{D_s}_{[0,1]}$, $\| \boldsymbol{w}_s \| = 1$, containing the number of image pixels per each point of $\Pi_s$;
  ▷ compute the baricenter of the data in the plane $\Pi_s$ as $\boldsymbol{\beta}_s \doteq (\boldsymbol{w}_s^T \boldsymbol{c}_s, \boldsymbol{w}_s^T \boldsymbol{\ell}_s)$;
  ▷ sort the vector $\boldsymbol{w}_s$ in descending order and define the vector $\boldsymbol{p}_s \in \mathbb{N}^{D_s}_{[1, D_s]}$ containing such ordering;
  ▷ set $\boldsymbol{\beta}_s$ as the initial seed for clustering in the plane $\Pi_s$, associate it with a weight $0$, and initialize with them, respectively, two ordered sets $\mathbb{K}_s \subset \mathbb{R}^2_+$ and $\mathbb{W}_s^{(K)} \subset \mathbb{Q}_{[0,1]}$ whose elements are dinamically ordered according to their insertion or extraction order;
  for $l = 1$ to $l = D_s$
    ▷ define $\boldsymbol{y}(l) \doteq (\boldsymbol{c}_s(\boldsymbol{p}_s(l)), \boldsymbol{\ell}_s(\boldsymbol{p}_s(l)))$;
    ▷ compute the minimum distance $d(i)$ between $\boldsymbol{y}(l)$ and the cluster seeds in the set $\mathbb{K}_s$ as $d(i) = \min_{\boldsymbol{\kappa}_s \in \mathbb{K}_s} \| \boldsymbol{y}(l) - \boldsymbol{\kappa}_s \|$; let $\boldsymbol{\kappa}_s(i) \in \mathbb{K}_s$ denote the closest seed;
    if $d(i) > \tau_1$
    then
      ▷ add $\boldsymbol{y}(l)$ to the set $\mathbb{K}_s$ as a new seed and its associated weight $\boldsymbol{w}_s(\boldsymbol{p}_s(l))$ to the set $\mathbb{W}_s^{(K)}$;
    else
      ▷ $\boldsymbol{y}(l)$ belongs to the cluster defined by $\boldsymbol{\kappa}_s(i)$; this induces the shifting of the seed $\boldsymbol{\kappa}_s(i)$ whose new position and associated weight are respectively updated as

          $\boldsymbol{\kappa}_s(i) := \dfrac{w_s^{(K)}(i)\boldsymbol{\kappa}_s(i) + \boldsymbol{w}_s(\boldsymbol{p}_s(l))\boldsymbol{y}(l)}{w_s^{(K)}(i) + \boldsymbol{w}_s(\boldsymbol{p}_s(l))}$ and $w_s^{(K)}(i) := w_s^{(K)}(i) + \boldsymbol{w}_s(\boldsymbol{p}_s(l))$;

      ▷ compute the minimum distance $d(j)$ between $\boldsymbol{\kappa}_s(i)$ and the other seeds in $\mathbb{K}_s$ as $d(j) = \min_{\substack{\boldsymbol{\kappa}_s \in \mathbb{K}_s \\ \boldsymbol{\kappa}_s \neq \boldsymbol{\kappa}_s(i)}} \| \boldsymbol{\kappa}_s(i) - \boldsymbol{\kappa}_s \|$;
      if $d(j) < \tau_2$
      then
        ▷ the clusters associated with the seeds $\boldsymbol{\kappa}_s(i)$ and $\boldsymbol{\kappa}_s(j)$ must be merged together and the position and weight of the new resulting cluster are computed according to the following procedure:
        ▷ define $m \doteq \min(i, j)$ and $M \doteq \max(i, j)$;
        ▷ update

          $\boldsymbol{\kappa}_s(m) := \dfrac{w_s^{(K)}(i)\boldsymbol{\kappa}_s(i) + w_s^{(K)}(j)\boldsymbol{\kappa}_s(j)}{w_s^{(K)}(i) + w_s^{(K)}(j)}$ and $w_s^{(K)}(m) := w_s^{(K)}(i) + w_s^{(K)}(j)$;

        ▷ remove the seed $\boldsymbol{\kappa}_s(M)$ from $\mathbb{K}_s$ and its weight $w_s^{(K)}(M)$ from $\mathbb{W}_s^{(K)}$;
      end
    end
  end
end
```

Figure 3. Algorithm for k-means clustering in the planes Π_s, $s = 1, \ldots, S$.

In fact, the parameter τ_1 determines the average distance among clusters within the planes Π_s whereas the parameter τ_2 is related to the average radius of the clusters, regarding them as circles.

The number of segments wanted in an image may depend on the particular applications and these thresholds should to be chosen according to some suitable criteria on the segmentation coarseness. For instance, in the examples reported in this paper, we have experimentally found that a number of final segments on the order of 15 may be obtained by setting $\tau_1 = 25$ and $\tau_2 = 20$.

2.5 Palette color matching

At the end of clustering procedure we end up with a limited number of representative colors C'' (usually $C'' < 20$) in the hcL^* space which are projected back to the RGB space and arranged into a new color palette $\boldsymbol{\mathcal{P}}^{(\mathcal{Z})}$ of size $C'' \times 3$; the final segmented image is represented as $\boldsymbol{\mathcal{Z}} = \{\boldsymbol{\mathcal{Q}}^{(\mathcal{Z})}, \boldsymbol{\mathcal{P}}^{(\mathcal{Z})}\}$, which is obtained by matching the entries of $\boldsymbol{\mathcal{P}}^{(\mathcal{L})}$ with those of $\boldsymbol{\mathcal{P}}^{(\mathcal{Z})}$ in the $L^*u^*v^*$ space and re-indexing the matrix $\boldsymbol{\mathcal{Q}}^{(\mathcal{L})}$ as a new matrix $\boldsymbol{\mathcal{Q}}^{(\mathcal{Z})}$. These operations as well as the post-processing, which basically consists in performing a median filtering of $\boldsymbol{\mathcal{Q}}^{(\mathcal{Z})}$, can be found in [10].

Fig. 4 shows the segmented versions $\boldsymbol{\mathcal{Z}}$ of the two color images of Fig. 1, where the segments ($C'' = 14$ in both cases) are separated by white borderlines.

It turns out that our algorithm may effectively separate homogeneous regions of color images. Nonetheless, some areas appear to be oversegmented, mainly because of shadows, highlights, and reflections. This

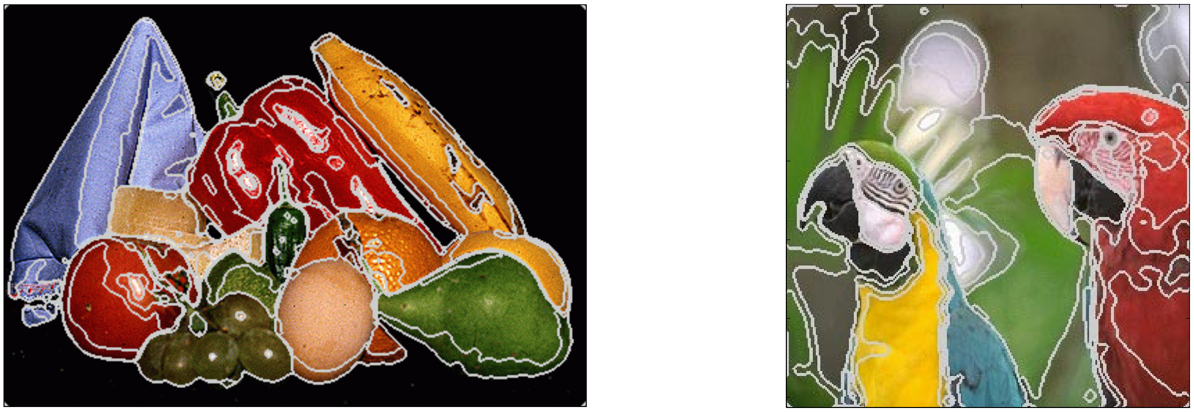

Figure 4. Original color images of Fig. 1 with borderlines drawn between homogeneous regions (there are $C'' = 14$ segments in both examples).

is generally inavoidable with segmentation algorithms operating in some feature space without any physical model of the interaction of light with color. Physics based techniques for color segmentation may be found in the references of [6, 7].

3 Conclusions

We have presented an original algorithm for unsupervised segmentation of color images based on their low-frequency content. Indeed, the low-pass filtering of color images produces smooth segments and sharpens histograms of color attributes. Our algorithm operates in the cylindrical representation of the $L^*u^*v^*$ color space where it finds representative colors by determining first the main hue families, through histogram thresholding, and then the main clusters on planes at constant hue, by means of k-means clustering.

We have reported and discussed two examples which confirm the effectiveness of our technique.

4 Acknowledgments

This work was supported in part by C.N.R., Italy, through a fellowship for research abroad (n. 203.15.9) and in part by a University of California MICRO grant, with matching supports from Lucent Technologies, National Semiconductor Corp., Raytheon Missile Systems, Tektronix Corp. and Xerox Corp.

References

[1] K.S. Fu and J.K. Mui, "A Survey on Image Segmentation," *Pattern Recognition*, Vol. 13, pp. 3-16, 1981.

[2] A. Rosenfeld and A. Kak, *Digital Picture Processing*, Vol. 2, 2^{nd} Ed., Academic Press, New York, NY, 1982.

[3] R.M. Haralick and L.G. Shapiro, "Image Segmentation Techniques," *Comp. Vision, Graphics, and Image Proc.*, Vol. 29 , No. 1, pp. 100-132, Jan. 1985.

[4] N.R. Pal and S. K. Pal, "A Review on Image Segmentation Techniques," *Pattern Recognition*, Vol. 26, No. 9, pp. 1277-1294, 1993.

[5] Q.T. Luong, "Color in Computer Vision," *Handbook of Pattern Recognition and Computer Vision*, L.F. Pau and P.S.P. Wang Eds., World Scientific, Singapore, pp. 311-368, 1993.

[6] W. Skarbek and A. Koschan, "Colour Image Segmentation - A Survey," Technical Report 94-32, Technical University Berlin, Oct. 1994.

[7] L. Lucchese and S.K. Mitra, "Advances in Color Image Segmentation," to appear in *Proc. of Globecom'99*, Rio de Janeiro, Brazil, 5-9 Dec. 1999.

[8] R.W.G. Hunt, *Measuring Colour*, 2^{nd} Ed., Ellis Horwood Ltd. Publ., Chichester, England, 1987.

[9] R.W.G. Hunt, *The Reproduction of Colour*, 5^{th} Ed., Fountain Press, Kingstone-upon-Thames, UK, 1995.

[10] L. Lucchese and S.K. Mitra, "An Algorithm for Fast Segmentation of Color Images," *Proc. of the* 10^{th} *International Tyrrhenian Workshop on Digital Communications - "Multimedia Communications,"* Ischia, Italy, Sept. 1998, pp. 110-119.

[11] L. Lucchese and S.K. Mitra, "An Algorithm for Unsupervised Color Image Segmentation," *Proc. of the 1998 Workshop on Multimedia Signal Processing*, Los Angeles, CA, Dec. 7-9, 1998, pp. 33-38.

[12] B. Jähne, *Digital Image Processing*, Springer-Verlag, Berlin, Germany, 1991.

[13] M.R. Anderberg, *Cluster Analysis for Applications*, Academic Press, New York, NY, 1973.

[14] S.-T. Bow, *Pattern Recognition and Image Preprocessing*, Marcel Dekker, Inc., New York, NY, 1992.