

Detecting the Main Colors of an Image: A Color Quantization Problem

Sergio Marquez-de-Silva, Edgardo Felipe-Riveron

Center for Computing Research, National Polytechnic Institute
Juan de Dios Batiz s/n, Col. Nueva Industrial Vallejo, Mexico D.F., Mexico
sama06@sagitario.cic.ipn.mx ; edgardo@cic.ipn.mx

Abstract. In this paper an algorithm for detecting the main colors in an image is presented. The method is based on finding homogeneous-color regions with a given level of color tolerance computed according the Lab color model using the CIE2000 metric and the application of a size-based selectable filter of a minimum region size. In order to achieve the best results, a particular way to explore the image is used.

Keywords: Color quantization, Essential colors; Color image segmentation; $L^*a^*b^*$; CIE2000; Size-based filtering.

1. Introduction

Color quantization consists of reducing the number of colors in a color image. This could be a 24-bit color image that contains at most $2^{24} = 16,777,216$ different colors. These colors are represented as three dimensional vectors, each vector element having 8-bit dynamic range allowing $2^8 = 256$ different values. These vectors are often called RGB triplets. A smaller set of representative colors is called color palette.

Until recently, quantization was used to reproduce 24-bit images on graphics hardware with a limited number of simultaneous colors. Even though 24-bit graphics hardware is becoming more common, color quantization still has practical application. It lessens space requirements for storage of image data, reduces transmission bandwidth requirements in multimedia applications, and is used in image processing applications, such as multispectral image segmentation.

The principal objective of this work is to use the color quantization as a tool for detecting the main colors composing the image. This technique can be used for factories that produce topographic maps, print the objects by layers and mixes of colors produced by them. Later to extract automatically the most important information from these maps is a very complex task.

A collateral result is a computer solution for image segmentation and image compression. This task is not a trivial one because it depends on the previous analysis of each image. The problem arose when it was necessary to find a procedure to achieve a fast, robust, and error-free way for build a color palette. We considered that the solution given by using the principle of similarity in color matching might be viable for this purpose.

© G. Sidorov, B. Cruz, M. Martínez, S. Torres. (Eds.)
Advances in Computer Science and Engineering.
Research in Computing Science 34, 2008, pp. 81-91

Received 07/04/08
Accepted 26/04/08
Final version 03/05/08

Main colors means in our solution those colors that are necessary to keep in the image in order to maintain the “semantic” of the image, that is, the whole meaning (details included) from the visual point of view.

This paper is organized as follows. Section 2 provides a brief summary of the state of the art. Section 3 reports the color representation used in the method and discusses the metric considered for color difference calculation. Section 4 presents the filter technique for deleting the non-significant color regions. Section 5 presents a dithering technique for rebuilding the image. The proposed method is described in Section 6. Finally, section 7 is dedicated to show some experimental results and finally the conclusions are included in section 8.

2. A Brief Discussion of the State-of-the-Art

2.1. Color Image Quantization

In this section we give a brief overview about color image quantization methods widely used in image processing.

Generally color quantization involves two steps: choosing or building a proper palette and afterwards reconstructing the output image using this palette.

One straightforward way to obtain quantized images with a low computational charge lie in using a pre-selected set of representative colors [7]. Such methods are referred to as image-independent quantization methods. Quantized images with a higher quality are generally obtained by building the set of representative colors according to the color distribution of the input image. These methods are called adaptive quantization methods or image-dependent quantization methods. Our method falls in this last category.

The uniform quantization of a color space Ω divides it into K equal size sub-boxes and defines the set of representative colors as the sub-box centroids. Uniform quantization techniques differ according to the geometry of the color space in which the quantization is performed.

Image-independent color quantization techniques [8] may be viewed as a generalization of uniform quantization where the chosen quantization regions are not of the same size but are still independent of the image to be quantized. Such techniques are often used to divide a color space according to a criterion that varies throughout the color space.

Due to the complexity of color quantization, preprocessing steps are commonly employed for reducing the data processed by quantization algorithms. The major techniques used to reduce the data are pre-quantization, which reduces the range of each coordinate in the color space, and histogram calculation, which allows us to manage the image color set more efficiently. The principal methods that have been developed for the solution of this problem are based on clustering [9]. Some recent color quantization methods, such as those described in [6], [7] y [8] can be referenced.

2.2. Color Image Segmentation

Methods that have been proposed for color image segmentation include active contours, clustering, wavelets and watersheds, among others. Some of these techniques are revised next.

There are several approaches to incorporate spatial information into a segmentation algorithm. One of the most straightforward methods is to include coordinate information as a part of feature sets [10]. However, since the coordinate data does not form a compact mass in feature space, this method will result in loose clusters even for a perfect feature set. Moreover, the choice of a coordinate system will also have a strong impact on the segmentation results.

Sapiro in [11] proposed a framework for object segmentation in vector-valued images, called color snakes. As in the original snakes' formulation for monochromatic images, color snakes present the nice property of smooth contours, but require manual initialization and may face converging problems. Liapis et al. in [12] proposed a wavelet-based algorithm for image segmentation based on color and texture properties. A multichannel scale/orientation decomposition using wavelet frame analysis is performed for texture feature selection, and then histograms in the CIELAB color space are used for color feature extraction. Two labeling algorithms are proposed to obtain the final segmentation results based on either or both features. This technique also achieves nice segmentation results for natural complex images, but the number of different color-texture classes must be selected by the user, which may not be easy to define in practical applications. Ma and Manjunath in [13] and Deng and Manjunath in [14] also proposed approaches for image segmentation based on color and texture. In [15], a predictive coding model was created to identify the direction of change in color and texture at each image location at a given scale, and object boundaries are detected where propagated "edge flows" meet. Other authors have combined watershed segmentation with multi-resolution image representations. Scheunders and Sijbers in [16] used a non-decimated wavelet transform to build a multiscale color edge map, which was filtered by a multivalued anisotropic diffusion. The watershed transform is then applied at each scale, and a hierarchical region merging procedure is applied to connect segmented regions at different scales. Despite the denoising power of both wavelet transform and anisotropic diffusion, the experimental results presented in their paper indicate a large number of segmented regions. Vanhamel et al. in [17] also explored multiscale image representations and watersheds for color image segmentation. In their approach, the scale-space is based on a vector-valued diffusion scheme, and color gradients in the YUV color space are computed at each scale. After applying the watershed transform, the dynamics of contours in the scale space are used to validate detected contours. Results presented in the paper are visually pleasant, but noisy images were not tested. Kazanov in [18] proposed a multiscale watershed-based approach for detecting both small and large objects, focused on scanned pages of color magazines. For detecting small objects, a small support edge detector is used. For larger objects, a multiscale version of the gradient estimator is computed. One potential problem of this technique is its high sensitivity to noise/texture, due to the use of small support edge detectors.

3. Color Representation

Color models, like all mathematical representations of physical phenomena, can be expressed in many different ways, each with its advantages and drawbacks. Some representation are formulated for helping humans to select colors – the Munsell System [19], for example – and others are formulated to ease data processing requirements – like the RGB space. The goal is to minimize formulation complexity and the number of variables, or dimensions. Historically, whatever the meaning assigned to the variables, three of them were enough to describe the majority of colors: RGB, Hue – Saturation – Brightness (HSB) and others HS based models, $L^*a^*b^*$, xyY , etc. Color perception is a brain process that starts in the eye's cone receptors. These receptors are found in three varieties that exhibit somewhat reddish, greenish, and bluish sensitivities. The “probable” sensitivities, according to Hunt [20] can be seen in Figure 1.

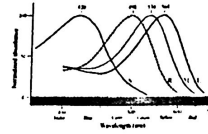


Fig. 1. Spectral absorption (normalized) of the human eye's cones

In 1931, the Commission Internationale de l'Eclairage (CIE) established standards for color representation based on the physiological perception of light. They are built on a set of three color-matching functions, called the tristimulus values. These functions are the base of almost all modern color models.

As good as it may be, the CIE 1931 standard is not without faults. Soon after it was introduced, it was found that it does not represent color gradations in a uniform manner. David L. MacAdam in [21] showed in the early 1940's that the minimum distance between two discernible colors is smaller in the lower left portion, and progressively bigger toward the top. It is also non-uniform; for any given point, tracing the coordinates of the minimally discernible colors around the point forms an ellipse, called a MacAdam ellipse.

Attempts to transform the original diagram into a more uniform representation resulted in the 1960 CIE Uniform Chromaticity Space (UCS), where projective transforms of the x and y space distort it to obtain somewhat more uniform u and v coordinates. More recently, there was an industry-wide “agreement” on two standards, the $L^*a^*b^*$ representation – called either CIE 1976 ($L^*a^*b^*$) or CIELAB – and the $L^*u^*v^*$ space – called either CIE 1976 ($L^*u^*v^*$) or CIELUV –, the later a slightly revised version of the 1960 CIE UCS. Since both spaces have their proponent and preferred applications, it is up to the users to select the most appropriate model; at least until a better “universal” one is defined and accepted. The $L^*a^*b^*$ space will be used in our method.

The $L^*a^*b^*$ is derived from the XYZ data through the following equations:

$$L^* = 116f\left(\frac{Y}{Y_n}\right) - 16 \quad (1)$$

$$a^* = 500 \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] \quad (2)$$

$$b^* = 200 \left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right] \quad (3)$$

Taking into account that:

$$f(I) = \begin{cases} I^{\frac{1}{3}} & \text{for } I > 0.008856 \\ 7.787I + \frac{16}{116} & \text{for } I \leq 0.008856 \end{cases} \quad (4)$$

Where: X_n, Y_n, Z_n are the values of X, Y, Z for a specified reference white. In the color transformation we use the D50 reference white that corresponds to the Daylight used for color rendering. As it was expected, the CIELAB color space is only approximately uniform. That is why color difference formulae have been proposed since acceptance of the CIE standards, most of the difference formulae attempt to use alternate (non-Euclidean) distance measures in the CIELAB space.

The CIE2000 color difference formula denoted as ΔE_{00} is designed to be symmetric by using averages of the standard and sample colors values in the weighting functions. The reader is referred to reference [22] for details.

$$\Delta E_{00} = \sqrt{\left(\frac{\Delta L^*}{K_L S_L}\right)^2 + \left(\frac{\Delta C^*}{K_C S_C}\right)^2 + \left(\frac{\Delta H^*}{K_H S_H}\right)^2 + R_T \left(\frac{\Delta C^*}{K_C S_C}\right) \left(\frac{\Delta H^*}{K_H S_H}\right)} \quad (5)$$

Where: $\Delta L^*, \Delta C^*, \Delta H^*$ are the differences of luminosity, chrome and hue calculated in the CIELAB model; W_L, W_C, W_H are weighting functions that improve the perceptual uniformity of CIELAB and K_L, K_C, K_H are three dependent numbers of the parametric effects and with a value of one for the reference conditions implemented in this work.

4. Filtering Method

Several regions found in images can be too small to be considered their colors in the final solution given by our methodology. This can be avoided by using a size-based filter related to the size of the region. If a region is smaller than the size value established by the user, the region will be erased and its pixels will be reassigned.

- The size of the region is calculated by counting the number of pixels belonging to the region considering 8-neighbor connectivity.
- The pixels that are to be reassigned chose from the neighbor region having a greater influence on them. The influence is determined by the number of pixels of all neighbor regions.
- The algorithm scans the image from left to right, top to bottom, looking for regions whose size is less than the minimum established size. Pixels belonging to the region to be eliminated are reassigned and then subtracted from the

original region. The process continues until all the regions surpass the minimum size established by the user.

The proposed region filtering method is implemented according to the following pseudo code listing:

```

for each y
  for each x
    region:=pixel[x][y]
    size:=region_size(region)
    if size<minimum_size
      add_to_list_of_neighbor_regions(pixel[x-1][y]);
      add_to_list_of_neighbor_regions(pixel[x-1][y-
1]);
      add_to_list_of_neighbor_regions(pixel[x][y-1]);
      add_to_list_of_neighbor_regions(pixel[x+1][y-
1]);
      pixel[x][y]=more_frequent_in_the_list_of_neighbo
r_regions();

```

5. Dithering Technique

Mapping the original colors to the color palette always implies a perceptual loss of image quality. This can be minimized using a dithering method. If the original pixel colors are simply translated into the closest available color from the palette, no dithering occurs. In a dithered image, colors not available in the palette are approximated by a diffusion of colored pixels from within the available palette. The human eye perceives the diffusion as a mixture of the colors.

Dithering is a technique used in computer graphics to create the illusion of color depth in images with a limited color palette. There are several algorithms designed to perform dithering. One of the earliest, and still one of the most popular, is the Floyd-Steinberg dithering algorithm [23], developed in 1975. One of the strengths of this algorithm is that it minimizes visual artifacts through an error-diffusion process. The Floyd-Steinberg algorithm typically produces images that are closer representations of the original than simpler dithering algorithms. It is commonly used by image manipulation software, for example when an image is converted into GIF format that is restricted to a maximum of 256 colors.

The algorithm achieves dithering by diffusing the quantization error of a pixel to its neighboring pixels, according to the distribution:

$$\frac{1}{16} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 3 & 5 & 1 \end{bmatrix} \quad (6)$$

Each time the quantization error is transferred to the neighboring pixels, while not affecting the pixels that have already been quantized. Hence, if a number of pixels have been rounded downwards, it becomes more likely that the next pixel is rounded upwards, so that on average, the quantization error is close to zero. Furthermore, this

method introduces “ghosts” into the picture. The ghosting problem can be ameliorated choosing the coefficients so that add to less than 1.

6. The Proposed Method

In this section the general arrangement, the block diagram and the algorithm of the proposed method is introduced. Figure 2 shows the block diagram of the method. In the first phase the input image is converted to the $L^*a^*b^*$ space. The second phase is dedicated to obtaining homogeneous regions in the image by mean of a growing process that, starting from a pixel that does not belongs to a region, progressively agglomerates points around it satisfying the homogeneity criterion called color tolerance threshold given by the user as a range of the maximum distance calculated by the ΔE_{90} function among colors.

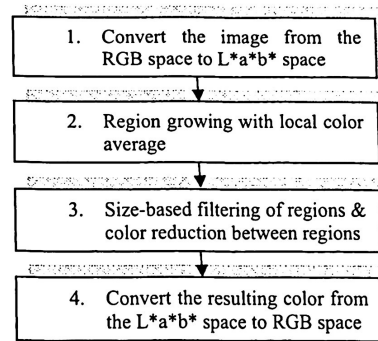


Fig. 2. Block diagram of the proposed method

The colors that are grouped probably change the region color (used for agglomerate successive pixels) according to the color located at the centroid found for the region in the CIELAB space; informally, it is the arithmetic average of colors of all points in the region. The algorithm scans the image in a zigzag manner from top to bottom, left to right and then from bottom to top in order to avoid a draining effect in the regions built like it is shown in Figure 3.

The third phase filters the smaller regions and sorts the regions from bigger to smaller according to the amount of pixels grouped; all the colors of regions with the number of pixels below the tolerance threshold are marked. The non-marked region will be our essential colors used as the palette for rebuilding the image. The output image is built converting the resulting color from the $L^*a^*b^*$ space to RGB space; finally the colors mapping is carried out using the dithering method explained in Section 5.

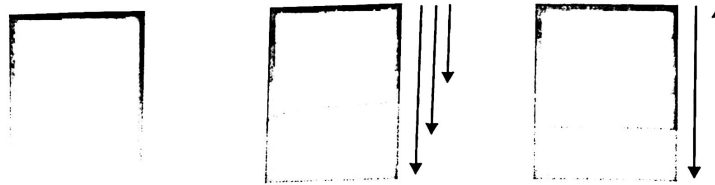


Fig. 3. A test image and their scan versions (a) Original (b) Draining effect (c) With zigzag scan

The proposed method is implemented according to the following pseudo code listing:

```

Original_Image=getImage();
Ima=Convert_to_L*a*b*(Original_Image);
for x = 0 to Ima.width()
    if x mod 2 = 0
        start = 0;
        end = Ima.height();
    else
        start = Ima.height()-1;
        end = 0;
    for y = start to end
        if Add_to_neighbor_region(Ima.pixel[x][y]) =
false
            Create_new_region(Ima.pixel[x][y]);
        Filter_small_regions(Ima);
        Merge_similar_regions(Ima);
    Result=Rebuild(Original_Image, Ima);
    Result=Convert_to_RGB(Result);

```

7. Results and Discussion

Two images were used for the analysis. Figure 4 shows the resultant dithered output images after the application of our color quantization method with a tolerance equal to eight, together with the initial and final number of colors. They are compared with the results obtained from two classical quantization methods: Median-Cut and Principal Components. The images are from the Signal and Image Processing Institute of the University of Southern California (SIPI-USC) database [24].

In the image of Lena, the quantization algorithm reduces automatically from 71068 to 12 colors, due to the little variety of colors that the human eye can appreciate in the original image.

In the Baboon image, a zoom of one eye is shown to highlight how the proposed quantization method preserves the relationship of tones when compared to the other two methods.

In the distribution of colors shown by the 3D-histograms of both images, we can observe that our method groups the colors in a better way. The future work include a complexity analysis of the method compared with other ones.

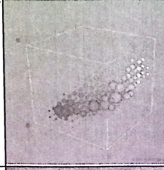
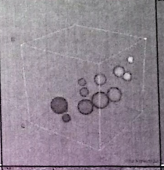
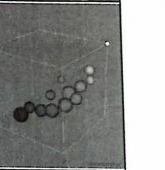
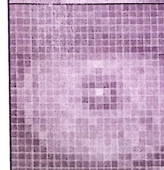
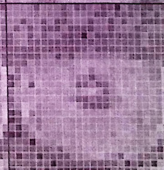
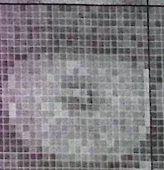
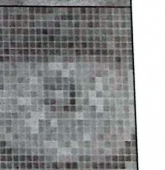
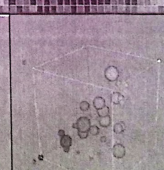
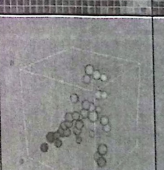
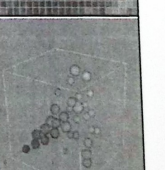
	Original	Proposed Algorithm	Median Cut	Principal Components
Lena 71,068 to 12 Colors				
				
Baboon 256 to 30 Colors				
				
				

Fig. 4. Results obtained after the application of the quantization process

8. Conclusion

In this paper a new color quantization method to find the color homogeneous regions was proposed. The unique parameter in this method simply is set according to the user priorities. This principle greatly increases the flexibility of the algorithm. The new method does not require the previous definition of the total number of colors to be reduced in the image, like the other ones. Besides this, the dithering technique that increases the perceptual result hardly has been used by other techniques to enhance the results in color quantization.

Acknowledgments

The authors of this paper wish to thank the Center for Computing Research (CIC), Mexico; Research and Postgraduate Secretary (SIP), Mexico, and National Polytechnic Institute (IPN), Mexico, for their support.

References

1. Allende, H. and Galbiati, J.: A non-parametric to filter for digital image restoration, using cluster analysis. *Pattern Recognition Letters*, Vol. 25, (2004) 841-847
2. Cortes, C. and Hertz, A.: Network System for Image Segmentation Intelligent Robotics, (1989) 121-125
3. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing* (2nd ed.), Prentice Hall, Upper Saddle River, NJ. (2002)
4. Bing, Z., Junyi, S., Qinke, P. An adjustable algorithm for color quantization, *Pattern Recognition Letters* 25 (2004) 1787-1797
5. Hautamäki, H., *Efficient Color Quantization by Hierarchical Clustering Algorithms*, University of Joensuu, (2005)
6. Petro, A. Analytical methods for the study of color in digital images, Palma de Mallorca, (2006)
7. Gentile, R., Allebach, J., and Walowit, E., Quantization of color images based on uniform color spaces, *J. Imaging Technol.*, 16(1), (1990) 11-21
8. Paeth, A. W., Mapping RGB triples onto four bits, in A.S. Glassner, Ed, *Graphics Gems*, Academic Press, Cambridge, MA, 718, (1990) 233-245
9. Anderberg, M. E., *Cluster Analysis for Applications*, Academic Press, New York (1973)
10. Krishnapuram, R., Freg, C.P.. Fitting an unknown number of lines and planes to image data through compatible cluster merging. *Pattern Recogn.* 25, (1992) 385-400
11. Sapiro, G.: Color snakes. *Computer Vision and Image Understanding* 68 (2), (1997) 247-253
12. Liapis, S., Sifakis, E., Tziritis, G.: Colour and texture segmentation using wavelet frame analysis, deterministic relaxation, and fast marching algorithms. *J. Visual Comm. Image Represent.* 15 (1), (2004) 1- 26
13. Ma, W.Y., Manjunath, B.S.: Edge flow: A technique for boundary detection and image segmentation. *IEEE Trans. Image Process.* 9 (8), (2000) 1375-1388
14. Deng, Y., Manjunath, B.S.: Unsupervised segmentation of color- texture regions in images and video. *IEEE Trans. Pattern Anal. Machine Intel.* 23 (8), (2001) 800-810
15. Nock, R., Nielsen, F.: Statistical region merging. *IEEE Trans. Pattern Anal. Machine Intel.* 26 (11), (2004) 1452-1458

16. Scheunders, P., Sijbers, J.: Multiscale watershed segmentation of multivalued images. In: International Conf. Pattern Recognition, (2002) III: 855–858
17. Vanhamel, I., Pratikakis, I., Sahli, H.: Multiscale gradient watersheds of color images. IEEE Trans. Image Process. 12 (6), (2003) 617–626
18. Kazanov, M.: A new color image segmentation algorithm based on watershed transformation. In: Internat. Conf. Pattern Recognition, (2004) II: 590–593
19. Birren F., Eds.: Munsell: A Grammar of Color, A Basic Treatise on the Color System of Albert H Munsell, Van Nostrand Reinhold, New York, (1969)
20. R.W.G. Hunt.: The reproduction of Colour, 5th ed., Fountain Press, ISBN 0-86343-381-2 (1995)
21. MacAdam D.L.: Visual sensitivities to color differences in daylight, J. Opt. Soc. Am., Vol. 32, (1942) 247-273
22. Luo M. R., Cui G., and Rigg B.: The development of the CIE 2000 colour difference formula: CIEDE2000, Color Res. Appl., 26(5), (2001) 340–350
23. Floyd, R. W. and Steinberg, L., An adaptive algorithm for spatial gray scale, in SID, Ed., *Int. Symp. Dig. Tech. Papers*, 36, (1975)
24. <http://sipi.usc.edu/services/database/Database.html>