Spatial Resolution and Distance Information for Color Quantization

Giuliana Ramella and Gabriella Sanniti di Baja

Istituto di Cibernetica "E. Caianiello", CNR
Via Campi Flegrei 34, 80078, Napoli
{g.ramella,g.sannitidibaja}@cib.na.cnr.it

Abstract. A new color quantization algorithm, CQ, is presented, which includes two phases. The first phase reduces the number of colors by reducing the spatial resolution of the input image. The second phase furthermore reduces the number of colors by performing color clustering guided by distance information. Then, color mapping completes the process. The algorithm has been tested on a large number of color images with different size and color distribution, and the performance has been compared to the performance of other algorithms in the literature.

Keywords: Color Quantization, Image Scaling, Distance Transform, Voronoi Diagram.

1 Introduction

Regions of a color image characterized by color homogeneity can be interpreted as constituting objects (or parts of objects) in the image. Generally, the number of perfectly homogeneous regions in the image overcomes the number of objects (or parts of objects) that a human observer is likely to perceive by looking at the image. In fact, human observers often need just a few colors for image understanding and, accordingly, group colors with similar tonality even if the obtained regions are not perfectly homogeneous. In analogy with the behavior of the human visual system, an automatic process is of interest, which is aimed at reducing the number of colors of a digital image, while causing in the resulting image the smallest possible visual distortion. This process is known as color quantization and is profitably employed in a number of applications such as image display, color based indexing and retrieval from image database, and storage and transmission of multimedia data files.

The literature includes a large number of color quantization methods that can be roughly classified as image independent and image dependent methods, as suggested in [1, 2].

Image independent methods, e.g., [3, 4], have generally lower performance with respect to image dependent methods as concerns the quality of the results, since they do not take into account the distribution of colors in the input image, but are rather convenient from the computational point of view.

In turn, image dependent methods, e.g., [2, 5-16] generally provide higher quality results, at the expenses of a larger computational effort. Image dependent methods are

A. Petrosino (Ed.): ICIAP 2013, Part II, LNCS 8157, pp. 642–651, 2013.

based, for example, on histogram analysis, fuzzy logic, neural network, multi-resolution analysis, and clustering. Actually, most image dependent methods are based on clustering in the color space. In fact, since the colors of pixels of an RGB image are a mixture of red, green and blue, color quantization can be seen as a clustering problem in the 3D space, where the three coordinate axes are the three color components. Each point in the 3D space represents one of the colors in the image. Since the values for the three color components are in the range [0, 255], the 3D space is limited to a cube with edges having length 256. Once a given clustering technique has been taken into account, each obtained cluster is associated a unique representative color, which can be computed, for example, as the average of the points in the cluster. Then, the representative colors replace the colors of the input image.

Image dependent methods can also be divided in pre-clustering methods (e.g., [5-8, 16]) that define a unique color palette by using features derived from the image at hand, and post-clustering methods (e.g., [2, 9-15]) that define an initial palette and improve it by means of an iterative process, which is repeated until the desired reduced number of colors is obtained.

In this paper, we present a new image dependent technique for color quantization that combines pre-clustering and post-clustering approaches. First, a reduced set of colors is determined by a pre-clustering method; then, a post-clustering scheme is adopted only if necessary to furthermore reduce the number of colors. The proposed method CQ is the follow up to a pre-clustering method that we have recently introduced [16]. CQ mainly consists of two phases. During the first phase, an improved version of the pre-clustering method [16] is accomplished to significantly reduce the number of colors. This goal is reached by suitably reducing the spatial resolution of the input image. In fact, the number of colors present in an image cannot be larger than the number of pixels forming the image itself. Then, if the resulting lower resolution image still includes a number of colors larger than that desired by the user, the second phase is activated. During the second phase, distance based clustering is accomplished in the RGB color space. The second phase of the process can be framed among post-clustering methods since the palette built during the first phase is iteratively processed as far as the number of colors overcomes the maximum number of colors fixed by the user. Once the final palette is available, the so identified colors of the 3D cube are taken as sources to compute the Voronoi Diagram. Finally, color mapping is taken into account to produce the output color image.

2 Basic Notions and Definitions

We work with RGB color images. Let I be any such an image. Colors are interpreted as three-dimensional vectors. Each vector element has an 8-bit dynamic range. As already pointed out in Introduction, the RGB color space can be represented by a 3D cube whose edges have length 256. The sides of the cube are aligned along the Cartesian axes. The origin of the Cartesian coordinate system is the black color (0,0,0). Since whichever color is considered its three colors components are integer numbers, the cube can be interpreted as a discrete cube and its points can be referred to as voxels.

The neighbors of a voxel p are the 26 voxels sharing with p a face, an edge, or a vertex. These neighbors are respectively denoted as face-, edge- and vertex-neighbors.

The 3D color histogram of I is built by assigning to the voxel in position (x, y, z) of the 3D cube a value equal to the number of pixels of I whose three color components have values x, y, and z, respectively.

In principle, pixels of an RGB image I can have any color out of a bit more than 16 millions of colors. In turn, I generally includes a number of pixels remarkably smaller than 16 millions. For example, a 512×512 image includes 262144 pixels, which naturally may assume at most 262144 different colors. Thus, most of the voxels of the 3D histogram of I generally have zero value, and the voxels with value different from zero constitute a sparse set of voxels.

The centroid of a discrete object consisting of a given number of voxels is the arithmetic mean position of all the voxels in the object. Thus, given n colors of the image I, i.e., n voxels in the 3D cube representing the RGB color space for I, their centroid is the color, whose color components are obtained by computing the arithmetic means of the color components of the selected n colors. By considering the histogram H of I, the occurrences of the n colors can be used to weight them. In this case, the centroid, computed as the arithmetic mean of all voxels weighted by their occurrences, can be interpreted as the physical center of mass of the object.

The distance transform of an image including two sets, the object and the background, is a replica of the image where the elements of the object are assigned the value of their distance from the background. The background can be seen as the source from which distance information propagates onto the object. In particular, the histogram H is a 3D image including the set of voxels corresponding to colors non existing in I (that we interpret as the object) and the set of voxels corresponding to colors existing in I (that we interpret as the background). We call DT the distance transform of H. Any distance function, including the Euclidean distance, can be used to compute DT. However, since H is a voxel image, it is particularly convenient to use a path-based distance, where the distance between two voxels is defined as the length of a shortest path between them. Each move along the path can be suitably weighted to take into account that the moves towards face-, edge- and vertex-neighbors have different Euclidean length. We use the <3,4,5> weighted distance, where the weights 3, 4 and 5 are respectively used for face-, edge- and vertex-neighbors [17]. Such a distance has been shown to provide a good approximation of the Euclidean distance.

To evaluate the performance of our color quantization method CQ, we use the Peak Signal to Noise Ratio $PSNR$.

For gray-level images, $PSNR$ is computed as:

$$PSNR = 20 \times \log_{10}\left(\frac{255}{\sqrt{MSE}}\right) \tag{1}$$

where

$$MSE = \frac{1}{H \times K} \sum_{i=1}^{H} \sum_{j=1}^{K} \left(v_{i,j} - w_{i,j}\right)^2 \tag{2}$$

and $v_{i,j}$ and $w_{i,j}$ respectively belong to the input image and to the output image of size $H \times K$.

For RGB images, the definition of *PSNR* is still the same, but *MSE* is the sum over all squared value differences divided by image size and by three.

To build from I an image with lower spatial resolution, but still preserving reasonably well shape and color information, we use a classical interpolation method, where to each resized pixel is assigned the average value of the pixels in the corresponding cell of the decimation grid used for image reduction [18]. The reduction factor f is the ratio between the number of rows r_f (columns c_f) that will characterize the lower resolution image and the number of rows r_i (columns c_i) in the input image. Thus, once the desired size of the lower resolution image has been fixed, say to a total of n_f pixels, the reduction factor $f=r_f/r_i$ can be computed by taking into account that $r_f \times c_f = n_f$ and $r_f/c_f = r_i/c_i$.

3 The Color Quantization Method CQ

Let I be an RGB color image with r_i rows and c_i columns, H its 3D histogram, and n_i and n_c the number of colors of I and the maximum number of colors desired for the quantized image, respectively. A running example is shown in Fig. 1 left, where I is a 320×480 image including 43650 colors.

Our color quantization method CQ mainly consists of two phases. During the first phase, the number of initial colors n_i is drastically reduced to at most n_f by reducing the spatial resolution of the input image I in such a way that the reduced size image I' consists of at most n_f pixels.

During the second phase, distance based color clustering is iteratively accomplished as far as the number of colors overcomes the maximum number of colors n_c fixed by the user for the quantized image.

Once the final color palette is available, color mapping is accomplished to produce the output color image.

43650 colors 53 colors 16 colors

Fig. 1. Input image, left, image resulting after the first phase of CQ, middle, and after the second phase of CQ, right

3.1 First Phase

The lower resolution image is built from I by means of the nearest neighbor scaling down process with reduction factor $f=r_f/r_i$. In principle, any reduction factor can be used so as to generate an image with any lower resolution size. Thus, one might directly compute the reduction factor necessary to generate a reduced size image with at most n_c pixels. However, this is not advisable if I is a large size image and n_c is rather small, unless the number of initial colors n_i is comparable with n_c. In general, we suggest to take 64 as the smallest possible value for n_f. In fact, with less than 64 pixels, the lower resolution image I' would not be adequate to represent reasonably well the input image I.

Once scaling down has been applied to I, at most n_f colors will be present in the reduced size image I'. Thus, the histogram H' of I' will include at most n_f colors, which can be grouped into a smaller number n_{cc} of connected components. In fact, some colors of I' may be so similar to each other that their corresponding voxels in H' belong to the same connected component. Only one color should be selected to replace in H' each connected component of colors. To this purpose, connected component labeling is accomplished, so as to assign an identity label to each connected component of colors. Then, the center of mass is computed for each component and replaces in H' the corresponding connected component.

We point out that some of the colors obtained in H' may not correspond to true colors of I, i.e., to colors existing also in H. This is due both to the scaling down process used to build I' and to the replacement of connected components of colors in H' by their corresponding centers of mass. In fact, in both cases a number of true colors of I in H is replaced in H' by a single color, which is definitely the best one to replace those true colors but is not necessarily itself a true color of I. Then, to improve the similarity between the input image and the quantized image, we replace colors that exist in H' but do not exist in H, with their closest colors in H. Actually, we search for the closest colors in a small neighborhood centered in the position of H corresponding to the color at hand. This is done both to limit the computational cost of the above color updating process and to avoid to replace a color with a too different color. In this paper, the neighborhood has size 11×11×11.

If the number n_{cc} of colors present in H' does not overcome the maximum number n_c requested by the user, the quantization process terminates and the final step, aimed at building the output color image, is directly accomplished. Otherwise, the second phase of the process is activated.

With reference to the running example, let us set n_f=64, which implies to use the reduction factor f=0.02. Then, the spatially reduced image I' has actually size 6×9 and includes 54 colors. These colors are grouped into n_{cc}=53 connected components. After center of mass detection and color updating, 45 colors of H' are replaced by their closest colors in H. The image resulting at the end of the first phase is shown in Fig. 1 middle. There, color mapping of I has been performed by using the method that will be described in Section 3.3.

3.2 Second Phase

To reduce the n_{cc} colors resulting after the first phase to at most n_c colors, we perform color clustering guided by distance information. To this purpose we compute the distance transform DT of H'. Colors present in H' are taken as the source from which distance information propagates onto the object, i.e., onto the set of voxels of H'

corresponding to colors that are not present there. Thus, the resulting *DT* will be a replica of *H'* where non existing colors are assigned the value of their <3,4,5> distance to the closest color existing in *H'*.

A binarization of *DT* is accomplished by setting to zero all voxels with an assigned distance value larger than an a priori fixed threshold θ. All other voxels are set to one. Colors of *H'* that were closer to each other than 2θ will result to be included in the binarized *DT* in the same connected component of voxels with value one. Each obtained connected component is a cluster grouping similar colors. Connected component labeling is performed to count the obtained number of clusters. If the resulting number does not overcome n_c, the process terminates. Otherwise, the value of θ is incremented and *DT* is binarized again, by using the updated value of the threshold. Binarization of *DT* and connected component labeling are iterated as far as the number of obtained clusters is larger than n_c.

The value of the binarization threshold is initially set to the smallest possible value when working with the <3,4,5> distance transform, i.e., θ=3. This roughly means that two voxels of *H'* will be merged in the same cluster if they are separated by at most other two voxels. Each time that binarization is iterated, the threshold is incremented by 3. This is done to reduce the risk to create excessive merging. Since the number of colors provided by the first phase is generally not much larger than n_c, a small number of iterations is generally sufficient to obtain the desired result.

Of course, clustering implies the replacement of colors in the same cluster by their corresponding center of mass, which may originate non true colors. Thus, once the number of clusters is at most equal to n_c, color updating is accomplished to replace non true colors as done in the first phase of CQ.

For the running example, let us suppose that the user desires at most 16 different colors. To reach this goal, starting from the first binarization of *DT* obtained with θ=3, 10 iterations are necessary until, with θ=30, the number of finally obtained clusters n_{fin} is at most equal to n_c. Actually, it results n_{fin}=16. After center of mass detection, 15 colors in *H'* are replaced by their closest colors in *H*. The resulting quantized image with n_{fin}=16 colors is shown in Fig. 1 right. Color mapping has been done as described in the next section.

3.3 Color Mapping

Once the final palette *H'* including n_{fin} colors is available, color mapping is accomplished to complete color quantization. To this aim, the Voronoi Diagram of *H'* is computed. In this way, the 3D cube is divided into a number of cells equal to the number of colors detected by the quantization process.

The n_{fin} colors in *H'* are used as the sources from which to compute the distance transform onto the remaining voxels of *H'*. Since the n_{fin} colors have been assigned an identity label, when distance information is propagated from the sources onto the remaining voxels of *H'*, also identity label is propagated. Thus, at the end of the process *H'* will result to be divided into a number of Voronoi cells, each of which including the voxels closer to the color included in that cell than to any other color. Voxels in the same Voronoi cell have the same identity label. The relation between each of the n_{fin} colors and the corresponding identity label is recorded.

Color mapping is done by changing the color of each pixel p of I, whose color components have values x, y, and z respectively, with the color associated to the Voronoi cell including the voxel in position (x, y, z).

4 Experimental Results

We have tested the color quantization method CQ on several color images with different size and color distribution, taken from available repositories, e.g., [19-21]. For illustration purpose, a small set of sixteen test images is shown in Figure 2. The number of colors and the size of the test images are given in Table 1.

a) b) c) d) e) f) g) h)

i) j) k) l) m) n) o) p)

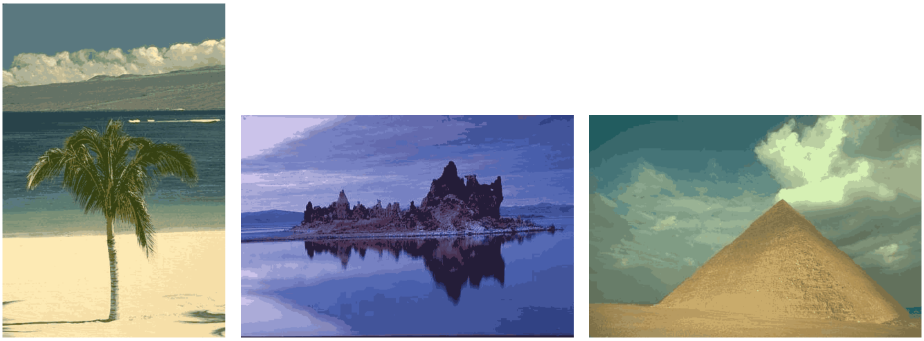
Fig. 2. Sixteen test images and their respective size and number of colors

To achieve a reasonably small number of colors during the first phase, while still preserving shape and color information, we suggest to use n_f=64 as default value. Thus, during the second phase n_c=32 or n_c=16 colors can be efficiently obtained.

A qualitative evaluation of the performance of CQ can be appreciated with reference to Fig. 3, where the quantized images obtained by setting n_f=64 and n_c =32 are shown for six test images. Under each image, the number of final colors n_{fin} is indicated.

We have also quantitatively compared the performance of CQ with that of other well-known methods in the literature, namely with the Median Cut MC [5], the Octree OT [6], and the method XW by X.Wu [7]. To this purpose, we resorted to Peak Signal to Noise Ratio PSNR. Actually, since the number of final colors n_{fin} may be slightly different for the four methods, rather than PSNR we computed the ratio PSNR/n_{fin}. The comparison is summarized for n_f=64 and n_c=32 in Table 1 for the sixteen test images and the four methods CQ, MC, OT and XW. The best values (i.e., the maximal values for PSNR/n_{fin}) are in bold. It can be noted that CQ has in general a better performance with respect to MC, OT and XW. The same holds also for the entire dataset of images we have been working with.

a) 28 colors b) 25 colors e) 30 colors



g) 30 colors i) 21 colors j) 30 colors

Fig. 3. A few resulting images for n_f=64 and n_c=32

Table 1. Results for the sixteen test images, case n_f=64 and n_c=32

image	n_i	size	CQ	MC	OT	XW
a)	71045	512×512	**0.88**	0.78	0.76	0.82
b)	8764	321×481	**1.31**	1.05	1.05	1.06
c)	43650	320×480	**0.99**	0.91	0.89	0.94
d)	111344	512×512	**0.92**	0.83	0.85	0.89
e)	1843	261×388	**1.30**	1.02	0.96	1.15
f)	50990	481×321	**0.97**	0.94	0.89	0.95
g)	34991	481×321	**1.02**	0.95	0.90	0.99
h)	9384	321×481	**2.29**	1.15	1.28	1.20
i)	32294	321×481	**1.44**	0.99	0.95	1.01
j)	23955	321×481	**1.07**	0.95	0.91	1.02
k)	6073	321×481	**1.27**	1.11	1.12	1.18
l)	17474	321×481	**1.08**	1.01	1.06	1.07
m)	21443	321×481	1.06	1.04	1.04	**1.09**
n)	178778	512×512	**0.95**	0.82	0.83	0.90
o)	154605	512×512	**0.91**	0.83	0.86	0.89
p)	77426	320×480	0.89	0.86	0.85	**0.90**

5 Conclusion

A new color quantization algorithm has been presented, which includes two phases. The first phase reduces the number of colors via spatial resolution reduction. The second phase furthermore reduces the number of colors by performing color clustering in the RGB space, based on the distance among colors. Color mapping completes the process.

The combined use of pre-clustering and post-clustering schemes is computationally convenient. In fact, post-clustering schemes, if applied directly to an input image characterized by a large number of colors, would require a large number of iterations to originate a quantized image with a small number of final colors. If pre-clustering is taken into account before using a post-clustering scheme, the necessary number of iterations will be considerably reduced, so limiting the computational burden. On the other hand, if only pre-clustering based on spatial reduction of the input image is taken into account, a very small number of final colors would be difficult to achieve. In fact, scaling down should originate an image with a smaller number of colors but still having a reasonable size, able to represent shape and colors of the input image.

References

1. Brun, L., Trémeau, A.: Digital Color Imaging Handbook, Chapter 9: Color Quantization, Electrical and Applied Signal Processing, pp. 589–638. CRC Press (2002)
2. Celebi, M.E.: Improving the performance of k-means for color quantization. Image and Vision Computing 29, 260–271 (2011)
3. Paeth, A.W.: Mapping RGB triples onto four bits. In: Glassner, A.S. (ed.) Graphics Gems, pp. 233–245. Academic Press, Cambridge (1990)
4. Mojsilovic, A., Soljanin, E.: Color quantization and processing by Fibonacci lattices. IEEE Transactions on Image Processing 10(11), 1712–1725 (2001)
5. Heckbert, P.S.: Color Image Quantization for Frame Buffer Display. In: ACM SIGGRAPH 1982, vol. 16(3), pp. 297–307 (1982)
6. Gervautz, M., Purgtathofer, W.: A Simple Method for Color Quantization: Octree Quantization. Academic, San Diego (1990)
7. Wu, X.: Color quantization by dynamic programming and principal analysis. ACM Transactions on Graphics 11(4), 349–372 (1992)
8. Kanjanawanishkul, K., Uyyanonvara, B.: Novel fast color reduction algorithm for time-constrained applications. Journal of Visual Communication and Image Representation 16, 311–332 (2005)
9. Ozdemir, D., Akarun, L.: A fuzzy algorithm for color quantization of images. Pattern Recognition 35, 1785–1791 (2002)
10. Bing, Z., Junyi, S., Qinke, P.: An adjustable algorithm for color quantization. Pattern Recognition Letters 25, 1787–1797 (2004)
11. Kim, N., Kehtarnavaz, N.: DWT-based scene-adaptive color quantization. Real-Time Imaging 11, 443–453 (2005)
12. Atsalakis, A., Papamarkos, N.: Color reduction and estimation of the number of dominant colors by using a self-growing and self-organized neural gas. Engineering Applications of Artificial Intelligence 19, 769–786 (2006)

13. Chen, T.W., Chen, Y.L., Chien, S.Y.: Fast image segmentation based on K-means clustering with histograms in HSV color space. In: Proc. IEEE 10th Workshop on Multimedia Signal Processing, pp. 322–325 (2008)
14. Ramella, G., Sanniti di Baja, G.: A new technique for color quantization based on histogram analysis and clustering. IJPRAI (2013)
15. Rasti, J., Monadjemi, A., Vafaei, A.: Color reduction using a multi-stage Kohonen Self-Organizing Map with redundant features. Expert Systems with Applications 38, 13188–13197 (2011)
16. Ramella, G., Sanniti di Baja, G.: Color quantization via spatial resolution reduction. In: Battiato, S., Braz, J. (eds.) VISAPP 2013, vol. 1, pp. 78–83. SCITEPRESS, Lisboa (2013)
17. Borgefors, G.: On digital distance transform in three dimensions. CVIU 64(3), 368–376 (1996)
18. Pratt, W.K.: Digital Image Processing. John Wiley & Sons, New York (2001)
19. http://www.hlevkin.com/TestImages/
20. http://r0k.us/graphics/kodak/
21. http://sipi.usc.edu/database/