

Multiresolution Histogram Analysis for Color Reduction

Giuliana Ramella and Gabriella Sanniti di Baja

Istituto di Cibernetica "E.Caianello", CNR
Via Campi Flegrei 34, 80078 Pozzuoli, Naples, Italy
{g.ramella,g.sannitidibaja}@cib.na.cnr.it

Abstract. A new technique for color reduction is presented, based on the analysis of the histograms of an image at different resolutions. Given an input image, lower resolution images are generated by using a scaling down interpolation method. Then, peaks and pits that are present in the histograms at all resolutions and dominate in the histogram of the input image at full resolution are taken into account to simplify the structure of the histogram of the image at full resolution. The so modified histogram is used to define a reduced colormap. New colors possibly created by the process are changed into the original colors closer to them.

1 Introduction

The human visual system is able to distinguish a large number of colors. However, it generally groups colors with similar tonality, since even a few colors are often enough for image understanding. When considering a digital image, color reduction can be used analogously to generate a transformed image, where a smaller number of distinct representative colors are used, while the visual appearance of the original image and of the transformed image is as similar as possible. The increasing number of applications, e.g., [1-3], dealing with multimedia data where millions of distinct colors are present makes color reduction particularly useful, especially for storage and transmission purposes.

Color quantization can be seen as a clustering problem in the 3D space, where the coordinate axes are the color components and each point represents one of the colors in the image. By means of a clustering technique, points are grouped into an a priori fixed number of clusters, each of which is associated a representative color, generally obtained as the average of the points in the cluster [4-6]. The most known methods to build a colormap with an a priori fixed number of colors are the median cut algorithm [1], which is based on the popularity method suggested by Boyle and Lippman in 1978, and the octree color quantization algorithm [7]. Other quantization methods in the literature are based on histogram analysis [8-10], fuzzy logic [11,12], neural network [13,14] and multiresolution analysis [9,10,15].

In this paper, we present a color reduction algorithm, based on color distribution and on the use of multiresolution image representation. The method is a substantial modification and improvement of a method we have recently suggested, which is also in the framework of multiresolution histogram analysis [10]. With respect to our previous method, the main differences are: 1) a different strategy to obtain the

multiresolution image representation, 2) the process for simplifying the structure of the histogram, 3) the introduction of an updating of the colormap (both when building lower resolution representations of the input image and when computing the reduced colormap), and 4) the ability of the method to originate automatically a transformed image with a number of colors in the range established by the user.

One of the advantages of the suggested method is the possibility to obtain different transformed images, each characterized by a different number of colors. This makes our method useful for progressive transmission, where a transformed image quantized with a small/large number of colors can be initially transmitted, and versions with larger/smaller numbers of colors can be provided, if demanded by the receiver.

The paper is organized as follows. Some preliminary notions are given in Section 2; the method is described in Section 3 and experimental results are discussed in Section 4. Concluding remarks are finally given in Section 5.

2 Preliminaries

We work with RGB images and interpret colors as three-dimensional vectors, with each vector element having an 8-bit dynamic range.

Given a color image I , let H be the histogram of the values in any of the color components of I . We consider as peaks and pits of H the values that are relative local maxima and relative local minima, respectively. Formally, if $p-1$, p and $p+1$ are three consecutive values in H , and $\text{height}(p)$ denotes the height of the bin associated to p in H , i.e., the number of pixels with value p , then:

- if $(\text{height}(p-1) \leq \text{height}(p) \text{ and } \text{height}(p+1) < \text{height}(p))$ or $(\text{height}(p-1) < \text{height}(p) \text{ and } \text{height}(p+1) \leq \text{height}(p))$, p is a peak;
- if $(\text{height}(p) \leq \text{height}(p-1) \text{ and } \text{height}(p) < \text{height}(p+1))$ or $(\text{height}(p) < \text{height}(p-1) \text{ and } \text{height}(p) \leq \text{height}(p+1))$, p is a pit.

Peaks and pits of H can be seen as vertices of a polygonal approximation of H . To simplify the structure of the histogram by retaining only the most significant vertices, we associate to each vertex v_i three parameters (see Fig. 1a):

- the area a_i of the region of the histogram dominated by v_i (i.e., the area of the triangle formed by the three successive vertices v_{i-1} , v_i and v_{i+1}),
- the cosine c_i of the angle formed by the two straight lines respectively joining v_i with v_{i-1} and with v_{i+1} ,
- the distance d_i of v_i from the straight line joining v_{i-1} and v_{i+1} .

The above parameters, introduced in [16] in the framework of 2D object's contour analysis to define the dominance of a vertex on the basis of criteria of perceptual significance, are here used in the context of multiresolution histogram analysis (see Section 3).

Lower resolution representations of a color image I can be obtained in different manners, e.g., by means of pyramids, and can be used to simplify the structure of the full resolution histogram. In this paper, we compute lower resolution representations

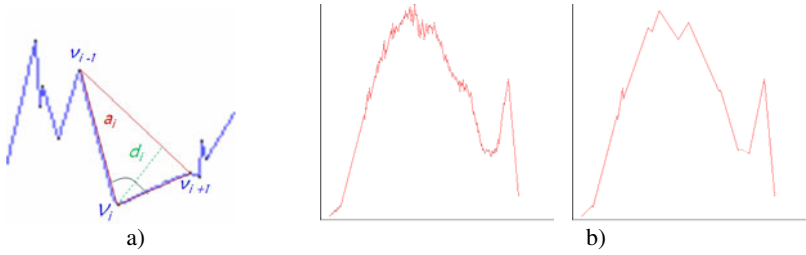


Fig. 1. a) The three parameters a_i, c_i, d_i associated to the vertex v_i , b) An histogram before, left, and after the simplification process, right

of I by means of a scaling down method based on nearest neighbor interpolation. The main advantage of this choice, with respect to using a Gaussian pyramid, is that any reduction factor f can be employed, instead of the reduction factor of 0.5 along each coordinate direction. Thus, lower resolution versions of I can be obtained, where resolution diminishes less abruptly along the levels. Another useful feature is that f can assume different values during the construction of the lower resolution images. In particular, a smaller f can be used to build the first lower resolution image, starting from the full resolution input image, and a larger f to build all successive lower resolution images, each of which is computed starting from the immediately previous one. The use of a larger value for f at the first scaling down step is motivated by the fact that even a rather strong compression does not dramatically alter the information contents, when done on the full resolution image. In turn, a smaller compression is advisable at the successive scaling down steps, i.e., when working with images that have already lost some information due to the resolution reduction.

3 The Algorithm

Our color reduction algorithm, based on the multiresolution analysis of color distribution in I , consists of two steps.

Step 1 is devoted to the following tasks: 1) construction of L lower resolution representations I_k of the input image I ; new colors, possibly created in the images I_k , are changed to their closest colors present in I . 2) computation, for each color component of I , of the histogram H of I as well as the histograms H_k for the L lower resolution images I_k .

Step 2 is devoted to the multiresolution analysis of the histograms to identify values that can be grouped together and be replaced by a unique representative value in H . To this aim, the structure of H is simplified, by using information derived from H as well as from a subset of the L histograms H_k . The representative values found for the histogram H of the color components are combined, to generate the reduced colormap. New colors possibly created during histogram manipulation, are replaced by the color of the original colormap to which they result to be closer.

If the number of colors of the transformed image I' is in the range desired by the user the process terminates. Otherwise, Step 2 is repeated, after the number of lower resolution images taken into account is suitably increased/decreased, until the number

of colors of I is in the desired range. During Step 1, the original colormap is computed and is stored in a 3D array, where position (x,y,z) corresponds to the color with $R=x$, $G=y$, $B=z$. Then, standard OpenCV libraries are used to build the lower resolution images I_k via nearest neighbor interpolation. As pointed out in Section 2, different values for f can be used during scaling down. We have experimentally found, by analyzing images with different size and color distribution, that satisfactory results are obtained, in the average, by using a reduction factor $f=0.5$ for the first resolution reduction and $f=0.8$ for the successive ones. These values will be used as default values in this paper.

In general, a number of images I_k could be built by repeatedly applying the scaling down process, but images I_k with too small size do not adequately preserve the information contents of I . Thus, we consider only the L successive images I_k whose size, expressed in number of pixels, includes at least 32 pixels for row and column. As soon as a lower resolution image I_k is created, the corresponding color map is built. Colors that are present in I_k , but do not exist in I , are changed to their closest colors in the original colormap, before building the successive lower resolution image. The histogram H of I and the histograms H_k of the L images I_k are computed. Peaks and pits are detected on all histograms. Depending on the number of histograms H_k , out of the L computed, that will be used for multiresolution histogram analysis, the transformed image I' will be characterized by a different number of colors. A large number of resolution images will produce a transformed image with a small number of colors and vice versa.

Step 2 starts by examining N histograms, where it is $N= L/2 + 1$ and is aimed at simplifying H . We look for a simplification of H , accounting for the permanence along the N histograms and the dominance in H (intended as perceptual relevance) of a vertex v_i . The permanence of v_i of H is checked by verifying if the value of v_i actually corresponds to a vertex in each H_k . The dominance of v_i is evaluated with respect to geometrical parameters of the triangles associated to all the vertices. The choice of these parameters is done to take into account factors influencing human perception such as the size of the region "viewed" by a vertex (the area a_i of the triangle associated to v_i) and the "cornerity" (roughly related to c_i) of the polygonal line including v_i . Moreover, the distance d_i is related to an often used measure of perceptual significance in the context of polygonal approximation. The dominance of a vertex is evaluated in the context of the overall shape, by referring to the average value of all areas of the triangles associated to all the vertices. The average value is a term of comparison for each area and guides the process of vertex selection by giving more evidence to the vertices having area larger than the average. The dominance of a vertex is also locally evaluated by comparing the geometrical parameters associated to each pair of consecutive vertices.

Let v_1, v_2, \dots, v_n be the vertices of H and let A be the average of the areas a_i of the triangles associated to the vertices v_i , $i=1, \dots, n$. The following process is done to simplify H .

- Any v_i of H that is not present in all the H_k and is such that $a_i < A$ is not retained as a vertex. The values of the three parameters (a_i , c_i , d_i) for all surviving v_i of H , and the average area A are updated.

- Any pair of consecutive vertices (v_i, v_{i+1}) of H such that $(a_i < A$ and $a_{i+1} < A)$ and $(a_i = a_{i+1},$ or $c_i = c_{i+1},$ or $d_i = d_{i+1})$ is not retained as a vertex.
- Any v_i of H which is no longer a relative maximum or a relative minimum of H is not retained as a vertex.

In Fig. 1b, an histogram H is shown before and after the simplification process. Once the structure of H has been simplified, all values from a pit to the successive pit are replaced by a single representative value. Three cases are possible:

- If one single peak is in between two successive pits, the representative value is the value of the peak.
- If more than one peak is in between two successive pits, the representative value is the value of the leftmost peak.
- Otherwise, the representative value is the value of the leftmost pit.

After combining the representative values of the color components the representative colors are obtained. These may not be included in the original colormap. To limit the presence of false colors, each representative color that is not found in the original colormap is replaced by the closest color in the original colormap. Selecting $N=L/2 + 1$ histograms to identify the significant peaks and pits of H is in some cases already adequate to produce a transformed image I' with colors in the range desired by the user. Otherwise, Step 2 is automatically repeated by analyzing either $N+1$ or $N-1$ histograms. In fact, let $C1$ and $C2$ be the minimum and maximum number of colors desired by the user and let RC_1 be the number of resulting representative colors, as obtained when Step 2 is accomplished for the first time, i.e., by analyzing $L/2 + 1$ histograms. If it results $C1 \leq RC_1 \leq C2$, the obtained image I' is the final transformed image. If $RC_1 < C1$ ($RC_1 > C2$), $N=N-1$ ($N=N+1$) histograms are considered so originating a different simplification of H and, as a consequence, producing a new transformed image with a larger (smaller) number of colors. Step 2 is repeated as many times as necessary to obtain a transformed image I' with colors in the desired range.

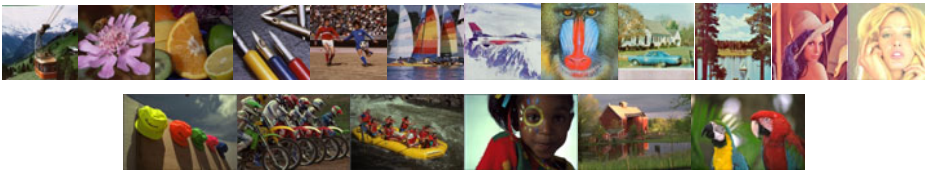


Fig. 2. A set of test images

4 Experimental Results

We have applied the color reduction algorithm to a collection of images with different size and color distribution, taken from available databases, e.g., [17-19]. A small dataset including six 512×480 images, six 512×512 images, and six 768×512 images

is given in Fig. 2. This dataset is used to show the performance of our method in terms of the obtained number of representative colors, accounting for the degree of color compression, and the compression ratio, computed as the ratio between the size of the output stream and the input stream expressed in bit per pixel [20].

Table 1 summarizes the quantitative measures for the images in the dataset, after one application of Step 2, where $L/2 + 1$ histograms are analyzed to simplify H . The number of colors in I and in I' are denoted by OC and RC_I , respectively. CR_I denotes the compression ratio of I' .

If the user desires a transformed image with a number of colors ranging, say, between $C1$ and $C2$ different colors, for some images in the dataset, the condition $C1 \leq RC_I \leq C2$ is not satisfied in one application of Step 2. These images are automatically processed again by using $N+1$ or $N-1$ histograms. In particular, $N+1$ histograms are considered if $RC_I > C2$, since by increasing the number of resolution levels taken into account the number of colors in the reduced colormap decreases. In turn, $N-1$ histograms are analyzed if $RC_I < C1$. If the condition $C1 \leq RC_I \leq C2$ is still not satisfied after Step 2 is applied twice, multiresolution histogram analysis is repeated after furthermore increasing/decreasing the number of histograms. The process terminates as soon as with the current value N' of histograms, for the transformed image it is $C1 \leq RC_{I'} \leq C2$.

Table 1 summarizes also the final results for the images in the data set, in correspondence with $C1=256$ and $C2=512$. The last column of the table denotes how many other histograms more (positive value) or how many histogram less (negative values) have to be used with respect to the starting $L/2 + 1$ histograms to satisfy the requirements of the user (and, hence, how many repetitions of Step 2 are necessary).

At most three repetitions of Step 2 have been necessary in order the number of colors of the transformed image is in the selected range. This has been necessary, for example, for the images “airplane” and “kodim22”, for which the number of histograms leading to 263 and 265 colors respectively is diminished by 3 with respect to the initially selected number. The average compression ratio for the 18 images in the dataset is equal to 0.517, i.e., color information occupies after compression in the average 48,3% of its original size.



Fig. 3. Different resulting images, obtained with a different selection of the number of histograms

To show qualitatively the obtained results, refer to Fig. 3, where three different resulting images are shown for the input image “baboon”, satisfying different requirements of the user as far as the final range of colors is concerned. The three

Table 1. Results after one application of Step 2 (columns 3,4) and final results for $CI=256$ and $C2=512$ (columns 5,6)

Image	OC	RC_I	CR_I	RC_F	CR_F	$N'-N$
cablecar	130416	1328	0,611	258	0,471	2
flower	111841	573	0,546	344	0,502	1
fruits	160476	1136	0,587	452	0,510	2
pens	121057	1036	0,593	364	0,504	2
soccer	139156	934	0,577	383	0,502	2
yacht	150053	648	0,543	381	0,499	1
lena	69904	87	0,400	304	0,513	-2
tiffany	79228	695	0,580	359	0,522	2
airplane	47819	65	0,387	263	0,517	-3
baboon	171045	640	0,536	460	0,509	2
housed	154605	148	0,418	329	0,485	-1
lake	168459	87	0,371	271	0,466	-1
kodim03	34871	474	0,589	474	0,589	0
kodim05	63558	956	0,621	512	0,564	1
kodim14	55117	487	0,567	487	0,567	0
kodim15	44576	446	0,570	446	0,570	0
kodim22	53351	62	0,379	265	0,513	-3
kodim23	72079	284	0,505	284	0,505	0

transformed images, from left to right, are characterized by 640, 460, and 224 different colors. The numbers of histograms that have been used to simplify H are respectively 6, 8, and 9.

5 Concluding Remarks

A color reduction algorithm has been introduced that generates a transformed image with a smaller number of colors but still maintaining the visual aspect of the input image satisfactorily. The algorithm is based on the analysis of the histograms at different resolutions of the input image, obtained by using a scaling down method based on nearest neighbor interpolation. Only peaks and pits present at all resolutions and dominating in the full resolution histogram are considered to identify the representative values defining the reduced colormap. Colors of the reduced colormap that are not present in the original colormap are changed into the closest colors in the original colormap.

The algorithm has a limited computational complexity and is not time consuming. It does not require pre-quantization and generates a transformed image with a number of colors in an *a priori* fixed range. It has been implemented on a Pentium 4 (3.39 GHz, 2 GB RAM) personal computer and has been applied to a large set of images, producing satisfactory results in terms of compression ratio.

Since different transformed images are obtained depending on the number of analyzed histograms, the method can be used for progressive transmission, where an image characterized by strong color reduction, i.e., quantized by using a large number of histograms, can be initially transmitted and better versions can be provided if demanded by the receiver.

References

1. Heckbert, P.S.: Color Image Quantization for Frame Buffer Display. In: Proc. ACM SIGGRAPH 1982, vol. 16(3), pp. 297–307 (1982)
2. Plataniotis, K.N., Venetsanopoulos, A.N.: Color Image Processing and Applications. Springer, Heidelberg (2000)
3. Rui, Y., Huang, T.S.: Image Retrieval: Current Techniques, Promising Directions, and Open Issues. *Journal of Visual Communication and Image Representation* 10, 39–62 (1999)
4. Braquelaire, J.P., Brun, L.: Comparison and Optimization of Methods of Color Image Quantization. *IEEE Transactions on Image Processing* 6(7), 1048–1052 (1997)
5. Bing, Z., Junyi, S., Qinke, P.: An adjustable algorithm for color quantization. *Pattern Recognition Letters* 25, 1787–1797 (2004)
6. 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. of IEEE 10th Workshop on Multimedia Signal Processing, pp. 322–325 (2008)
7. Gervautz, M., Purgathofer, W.: A Simple Method for Color Quantization: Octree Quantization. In: Glassner, A.S. (ed.) *Graphics Gems*, pp. 287–293. Academic Press, London (1990)
8. Delon, J., Desolneux, A., Lisani, J.L., Petro, A.B.: A Nonparametric Approach for Histogram Segmentation. *IEEE Transactions on Image Processing* 16(1), 253–261 (2007)
9. Kim, N., Kehtarnavaz, N.: DWT-based scene-adaptive color quantization. *Real-Time Imaging* 11, 443–453 (2005)
10. Ramella, G., Sanniti di Baja, G.: Color quantization by multiresolution analysis. In: Jiang, X., Petkov, N. (eds.) *CAIP 2009*. LNCS, vol. 5702, pp. 525–532. Springer, Heidelberg (2009)
11. Ozdemir, D., Akarun, L.: A fuzzy algorithm for color quantization of images. *Pattern Recognition* 35, 1785–1791 (2002)
12. Shorter, N., Kasparis, T.: Fuzzy ART for Relatively Fast Unsupervised Image Color Quantization. In: Proc. of 19th Int. Conf. on Pattern Recognition. IEEE CS Press, Los Alamitos (2008) ISBN/ISSN: 978-1-4244-2175-6
13. Papamarkos, N., Atsalakis, A.E., Strouthopoulos, C.P.: Adaptive color reduction. *IEEE Transactions Systems, Man, and Cybernetics* 32(1), 44–56 (2002)
14. 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)
15. Robinson, J.A.: Adaptive Prediction Trees for Image Compression. *IEEE Transactions on Image Processing* 15(8), 2131–2145 (2006)
16. Arcelli, C., Ramella, G.: Finding contour-based abstractions of planar patterns. *Pattern Recognition* 26(10), 1563–1577 (1993)
17. <http://sipi.usc.edu/database/>
18. <http://www.hlevkin.com/TestImages/>
19. <http://r0k.us/graphics/kodak/>
20. Salomon, D.: *Data Compression: The Complete Reference*. Springer, London (2007)