Fast Color Image Quantization using Squared Euclidean Distance of Adjacent Color Points along the Highest Color Variance Axis

Y. Sirisathitkul[a] , S. Auwatanamongkol[a] and B. Uyyanonvara[b]
[a]School of Applied Statistics, National Institute of Development Administration, Bangkok, Thailand
[b]Department of Information Technology, Sirindhorn International Institute of Technology,
Thammasat University, Bangkok, Thailand
kin009@yahoo.com

Abstract

A new color image quantization algorithm that uses the squared Euclidean distance of adjacent color points along the highest color variance axis is proposed. This algorithm is a hierarchically divisive colormap design technique. Colors are sorted along the axis with the highest variance of color distribution. The squared Euclidean distances between any adjacent colors' along the axis are then used to find the cutting plane that divides a color cell into two subcells with approximately equal quantization errors respect to their centroids. The proposed algorithm is effective and yields a short execution time.

1. Introduction

Color image quantization is an important process of representing true color images using a small number of colors in a colormap. The objective of color image quantization is to quantize a true color image, typically $2^{24} \approx 16.8$ million colors for each pixel, into one with fewer colors (256 colors or less). If the image is quantized to 256 colors, the file size will be reduced to one-third of original size. The color image quantization algorithm consists of 4 phases. The original image sampling computes the image histogram for color statistics. The colormap design chooses the best possible set of representative colors. The pixel mapping maps each color in original image to a representative color in colormap. Finally, the image quantizing redraws the image by replacing the original color in every pixel with a representative color.

The existing colormap design algorithm may be categorized into 2 classes, splitting algorithm and clustering-based algorithm. A number of splitting colormap design algorithms [1-4] can be used to perform color image quantization. However, many algorithms are computationally intensive. Consider the well-known variance-based algorithm [2] which, at each step, selects the color cell with the highest variance of colors for division. All colors in the cell are projected onto each coordinate axis. For each projected distribution, the optimal cut-point is computed and the partition plane is chosen perpendicular to the axis with the largest reduction of expected variance among all projected distributions.

The clustering-based algorithms [5-10] cluster the color space into K-desired clusters. The methods involve an initial selection of colormap followed by repeatedly updating cluster representatives.

In this paper, we propose a simple hierarchically divisive algorithm for designing the colormap. The proposed algorithm, described in details in the next section, employs the squared Euclidean distance of adjacent color points along the highest color variance axis. Once the colormap has been constructed, each color pixel can then be replaced by finding the nearest neighbor of a color in the colormap so quantization errors of quantized images can be minimized.

The rest of this paper is organized as follows. First, the proposed algorithm for colormap design is introduced in section 2. In section 3, we compare the performance of the proposed algorithm with other existing algorithms. Finally, conclusions are given in section 4.

2. The proposed algorithm

In this section, we illustrate the proposed algorithm and show how it can be used to construct a colormap. The proposed algorithm is based on a concept that the sum of distances between adjacent colors along the highest color variance axis indicates the dissimilarity of all colors in each color cell. If we divide a cell into two smaller cells by balancing the sum of distances between adjacent colors and frequencies of each cell, it could mean that the colors in each cell are similar and gradually change. Consequently, the cell representative is getting close to every color in each cell and hence the total quantization

IEEE COMPUTER SOCIETY

error becomes minimal.

Before describing the proposed algorithm, let us first examine a diagram to clarify it. As shown in Figure 1, the six given data points are supposedly sorted by its X value and indicated by a number beneath each data point.

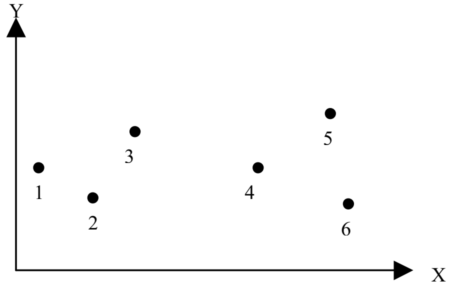

Figure 1. Diagram of six data points in 2D, sorted by its X value, with an ordering number for each data point.

Let $D_j = d\left(c_{j-1}, c_j\right)^2$ be the squared Euclidean distance of adjacent color points along the X-axis (as shown in Figure 2).

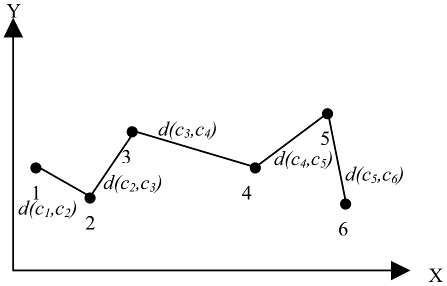

Figure 2. Illustration of six data points, a solid line represents the distance between adjacent colors along the X-axis.

The task of finding the partition point in 2D is thus replaced by finding in one-dimensional line as shown in Figure 3.

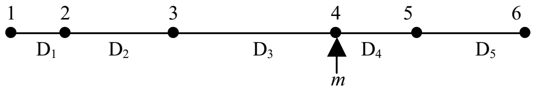

Figure 3. Illustration of six data points on a one-dimensional line and the relevant D_j.

The partition point (m) is thus a centroid on the one-dimensional line (as shown in Figure 3).

Let $dsum_i = \sum_{j=1}^{i} D_j$ and f_i is the relative frequency of i$^{\text{th}}$ color. The centroid on the one-dimensional line, denoted by $centroidDist$, is computed by:

$$centroidDist = \sum_{i=1}^{n} f_i \bullet dsum_i \Big/ \sum_{i=1}^{n} f_i \tag{1}$$

Thus, the total quantization error of the two smaller cells at the data point, closed to the $centroidDist$, are approximately equal.

Although the aforementioned technique partitions data points into only two subcells, we can repeat this technique K-1 times to obtain K mutually disjoint subcells. The proposed colormap design algorithm proceeds in hierarchically divisive manner. The details of this algorithm are summarized as follows:

Algorithm. Colormap design using squared Euclidean distance of adjacent color points along the highest color variance axis.

Input: a data set $S = \left\{\left(r_i, g_i, b_i\right) \mid i = 1, 2, \ldots, |S|\right\}$, a predefined value K and an image histogram.

Output: a colormap $C = \left\{\left(\overline{r}_j, \overline{g}_j, \overline{b}_j\right) \mid 1 \le j \le K\right\}$

Method:

Step 0:Let cell c_l be the entire data set S and go to step 2.

Step 1:Choose the cell c_l from remaining cells for further partition. If the cell contains only one color, choose a new cell.

Step 2: Compute the variance of each color component of the cell c_l.

Step 3: Choose a dimensional axis with the highest variance of color component as the principal axis.

Step 4: Sort all colors in the cell c_l along the principal axis.

Step 5: Compute the squared Euclidean distance $D_j = d\left(c_{j-1}, c_j\right)^2$ of adjacent colors. Compute the $dsum_i = \sum_{j=1}^{i} D_j$.

Step 6: Compute $centroidDist$ of the cell c_l

$$centroidDist = \sum_{i=1}^{n} f_i \bullet dsum_i \Big/ \sum_{i=1}^{n} f_i$$

where f_i is the frequency of i$^{\text{th}}$ color and $dsum_i$ is the summation of distance between the adjacent color.

Step 7: Divide cell c_l into two smaller cells. The partition boundary is the plane passing through a point m whose $dsum_i$ approximately equals to $centroidDist$.

Step 8: Repeat steps 1-7 until the number of cells reaches K.

Step 9: Collect the centroids of K cells to form the K desired representative colors in the colormap. Hence, a K-color colormap C has been produced.

Step 10: Stop.

3. Experimental results

All color image quantization algorithms were implemented on Intel PENTIUM-4 1.8 GHz PC with 256 MB RAM. Experiments were conducted on three true color images, taken from the internet, namely, “Lena”, “Zelda” and “Test”, respectively. Each RGB pixel of the original images is represented by 24 bits and 8 bits per component.

In this paper, the experiments is aimed to evaluate the performance of the proposed algorithm, determined by execution time and quantization errors. The mean squared error (MSE) and the peak-signal-to-noise ratio (PSNR) are used as criteria for evaluating the quantized error. The experiments employed the exhaustive search for pixel mapping since it gave the nearest representative color in the colormap for each color.

The 256-color quantized images of the “Lena”, “Zelda”, and “Test” generated by the proposed algorithm are shown in Figures 4–6, respectively. It can be seen that the quality of reconstructed images is still as good and acceptable.

Table 1 compares MSE values and execution time of the proposed algorithm, the median-cut algorithm, the variance-based algorithm, the octree algorithm and the Dekker’s algorithm using SOM (sampling factor = 1), respectively. It is evident that the quantization results from the proposed algorithm are better than those from other algorithms, except the variance-based algorithm and the Dekker’s algorithm. Figures 7-9 give the comparative performance in term of PSNR value with K = 256, 128 and 64. The PSNR value of the proposed algorithm is slightly above 30 dB, meaning that it is very difficult for human to perceive any differences between the original and the quantized images. Moreover, the proposed algorithm requires less execution time than the variance-based algorithm and the Dekker’s algorithm.

4. Conclusions

In this paper, a simple colormap design algorithm using squared Euclidean distance of adjacent color points along the highest color variance axis is proposed. Experiments show that the proposed algorithm is effective and requires less execution time than other well-known algorithms such as the variance-based, and the Dekker’s algorithm. Especially when applied to images with large number of distinct colors, the proposed algorithm requires a short execution time and produces a relatively low quantization errors.

References

[1] P. Heckbert, “Color image quantization for frame buffer display”, *ACM Trans. Computer Graphics (SIGGRAPH)*, vol. 16, no. 3, pp. 297-307, 1982.

[2] S. J. Wan, S. K. M. Wong, and P. Prusinkiewicz, “An algorithm for multidimensional data clustering”, *ACM Trans. Math. Software*, vol. 14, no. 2, pp. 153-162, 1988.

[3] G. Joy, and Z. Xiang, “Center-cut for color-image quantization”, The *Visual Computer*, vol. 10, no. 1, pp. 62-66, 1993.

[4] M. Gervautz, and W. Purgathofer, “A simple method for color quantization: Octree quantization”, *Graphics Gems*, Academic Press, New York, pp. 287-293, 1990.

[5] A.H. Dekker, “Kohonen neural networks for optimal colour quantization”, Network*: Computation in Neural Systems*, vol. 5, pp. 351-367, 1994.

[6] Y. Linde, A. Buzo, and R. Gray, “An algorithm for vector quantizer design”, IEEE *Trans. Commun*, vol. 28, no. 1, pp. 84-95, 1980.

[7] Z. Xiang, and G. Joy, “Color image quantization by agglomerative clustering”, *IEEE Comput. Graph Appl.*, vol. 14, no. 3, pp. 44-48, 1994.

[8] Z. Xiang, “Color image quantization by minimizing the maximum intercluster distance”, *ACM Trans. on Graphics*, vol. 16, no. 3, pp. 260-267, 1997.

[9] P. Scheunders, “A comparison of clustering algorithms applied to color image quantization”, *Pattern Recognition*, vol. 30, no. 6, pp. 859-866, 1997.

[10] G. Patane, and M. Russo, “The enhanced LBG algorithm”, *Neural* Networks, vol. 14. pp. 1219-1237, 2001.


(a) (b)

Figure 4. Quantization of the image “Lena”. (a) The original image and (b) the 256-color quantized image of the proposed algorithm.

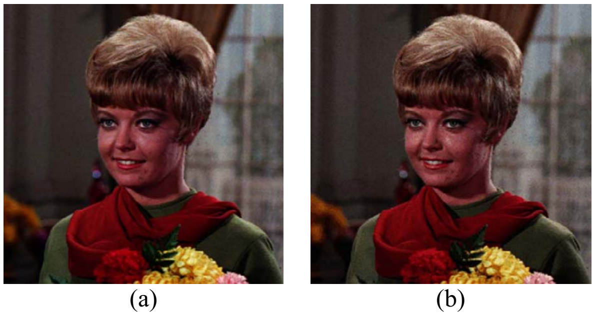
(a) (b)

Figure 5. Quantization of the image “Zelda”. (a) The original image and (b) the 256-color quantized image of the proposed algorithm.


(a) (b)

Figure 6. Quantization of the image “Test”. (a) The original image and (b) the 256-color quantized image of the proposed algorithm.

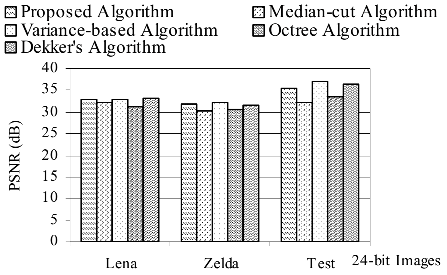

Figure 7. The PSNR of 256 quantized colors of all algorithms.

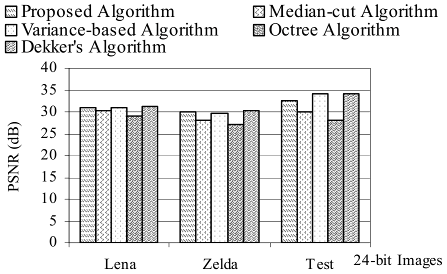

Figure 8. The PSNR of 128 quantized colors of all algorithms.

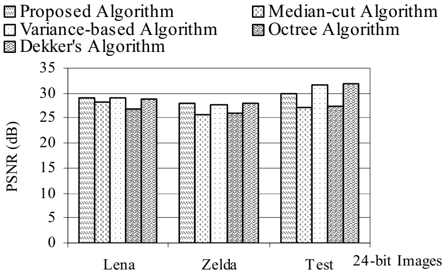

Figure 9. The PSNR of 64 quantized colors of all algorithms.

Table 1. The comparative performance in terms of the MSE value and the execution time (in milliseconds) on 24-bit colors of different algorithms.

24-bit images	No. of colors	Proposed algorithm		Median-cut		Variance-based		Octree		Dekker’s algorithm (SF = 1)	
		MSE	Time	MSE	Time	MSE	Time	MSE	Time	MSE	Time
Lena	64	82.77	455	97.37	430	82.94	2050	138.35	76	83.69	1127
	128	53.35	540	59.95	470	52.65	2520	84.03	97	47.86	1940
	256	33.97	554	39.19	500	33.79	3055	49.18	114	32.07	3525
Zelda	64	104.50	89	174.12	140	109.55	1685	164.41	20	107.64	268
	128	66.45	104	101.87	149	67.57	2024	123.19	25	61.66	480
	256	42.15	115	63.05	156	40.59	2460	56.47	32	46.61	892
Test	64	66.37	79	127.37	130	44.89	1539	115.87	72	43.27	1002
	128	36.19	86	63.93	140	24.03	1940	98.70	76	24.58	1699
	256	18.05	100	38.12	150	12.79	2365	28.28	100	15.36	3153