ELSEVIER

Signal Processing 78 (1999) 201–214

SIGNAL PROCESSING

www.elsevier.nl/locate/sigpro

Color quantization by preserving color distribution features☆

Wu-Ja Lin*, Ja-Chen Lin

Department of Computer and Information Science, National Chiao-Tung University, Hsinchu 30050 Taiwan, People's Republic of China

Received 19 November 1997; received in revised form 12 April 1999

Abstract

A color quantization method is proposed. The method hierarchically uses a bisection tool that preserves certain color-distribution features. The preserved features are average color, variance in each color dimension, and average color radius. An algorithm is given that includes the histogram simplification procedure, color palette generation, and pixel mapping. Experiments show that the proposed method usually obtains acceptable quantized images with competitive speed. © 1999 Elsevier Science B.V. All rights reserved.

Zusammenfassung

Eine Methode zur Farbquantisierung wird vorgestellt. Die Methods verwendet ein Werkzeug zur Farbhalbierung hierarchisch, das gewisse Farbverteilungseigenschaften bewahrt. Diese Eigenschaften sind die durchschnittliche Farbe, die Varianz in jeder Farbdimension und der durchschnittliche Farbradius. Ein Algorithmus wird angegeben, der ein Vereinfachungsverfahren des Histogramms, eine Erzeugung der Farbpalette und eine Pixelkarte beinhaltet. Experimente zeigen, dass die vorgeschlagene Methode akzeptable Quantisierungsbilder mit konkurrenzfähiger Geschwindigkeit liefert. © 1999 Elsevier Science B.V. All rights reserved.

Résumé

Nous proposons une méthode de quantification de la couleur. Elle utilise hiérarchiquement un outil de bissection qui préserve certaines caractéristiques de distribution des couleurs. Celles-ci sont la couleur moyenne, la variance dans chaque dimension de couleur et le rayon de couleur moyen. Nous donnons un algorithme qui inclus une procédure de simplification d'histogramme, de génération de palettes de couleurs et de mise en correspondance de pixels. Des expériences montrent que la méthode proposée obtient d'habitude des images quantifiées à une vitesse compétitive. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Color quantization; Histogram simplification; Color distribution features preserving; Hierarchical division

☆This study was supported by the National Science Council, Republic of China, under contract number: NSC.88-2213-E-009-067.

* Corresponding author. Tel.: + 35-715-900; fax: + 35-721-490.

E-mail address: gis82802@cis.nctu.edu.tw (W.-J. Lin)

PII: S0165-1684(99)00060-2

Nomenclature

P	The input image
P'	The simplified histogram of P
k	The palette size
C_m	The mth color of the palette, $1 \leqslant m \leqslant k$
Q	The quantized image
S	A set of color points in the RGB color space
S_A, S_B	A partition of S, i.e., $S_A \cup S_B = S$ and $S_A \cap S_B = \emptyset$
C_A	The representative color $C_A = (r_A, g_A, b_A)$ of S_A
C_B	The representative color $C_B = (r_B, g_B, b_B)$ of S_B
p_A	The percentage of points contained in S_A
p_B	The percentage of points contained in S_B

1. Introduction

Color quantization of an image is a process that uses a small number of colors to represent the image. The objective is to approximate as closely as possible the original full-color images. This technique is necessary for systems that can display only a few colors. For example, systems with 8 bits/pixel frame buffers can display only 256 colors. Although many modern systems have 24 bits/pixel frame buffers and can display $2^{24} = 16{,}777{,}216$ colors, color quantization is still practical for systems running animations and those used for advanced graphics applications. It reduces storage requirements and saves image transmission time over networks.

In general, color quantization can be divided into two parts: color-palette design, and mapping of pixels in the original image to colors in the designed palette. Several approaches can be used to perform color quantization. Linde et al. proposed the LBG method [15]. Given an initial palette, the LBG method minimizes image quantization errors by repeatedly assigning image pixels to the colors closest to them in the palette, and then updating palette colors using the average colors of pixels assigned to them. The LBG method usually obtains good color palettes when initial palettes (initial guesses) are carefully chosen. However, when a poor initial palette is used, the palette generated might also be poor. In other words, LBG performance is strongly dependent on the initial-palette choice. Scheunders proposed a genetic approach [17] to improve LBG's dependence on initial palettes and to further reduce quantization errors. Taşdizen et al. proposed a genetic method [19] for use in YUV color space. This method minimizes a distortion measure that uses different weights for the Y, U, V components ($Y:U:V = 10:3:3$) when measuring dissimilarities between colors. A common goal of the methods mentioned above is minimizing predefined distortion measures. Unfortunately, this kind of approach usually results in the problem of heavy computation loads, which makes them impractical for on-line quantization of colors.

Many methods use top-down (hierarchically divisive) approaches to accomplish on-line quantization. Among these methods, the median-cut method [11,14] proposed by Heckbert constructs the color palette using the strategy that all colors in the palette should represent approximately equal numbers of image pixels. The method recursively subdivides reduced RGB color space (that is, the input image is first prequantized using 3–3–3 bit-cutting for the RGB components) into rectangular hyperboxes. During subdivision, the hyperbox containing the largest pixels count is picked and subdivided. The division plane passes through the median point of the projected color distribution along the dimension with the largest color spread. The median-cut method has the advantage of being simple and easy to understand. However, the quantization error is not small enough.

Joy and Xiang proposed the center-cut method [12], a modification of the median-cut method. It uses 3–2–4 bit-cutting to prequantize input images, which results in the same memory requirement as

that of the median-cut method, and partially compensates for the nonuniform nature of RGB color space. Center-cut divides the hyperbox along its so-called longest-dimension, and the division plane passes through the center point rather than the median point. The center-cut method is simple and easy to implement. However, as indicated by the authors in [12], uneven bit-cutting (3–2–4) may cause undesirable hue shifts and an auxiliary procedure is required to avoid this problem.

Wu and Witten proposed the mean-split method [22,25], which uses the mean rather than the median of the projected distribution (with the largest spread) as the partition point. Let L be the number of clusters assigned to a splitting hyperbox. The number of clusters, L_i, assigned to each of the two resulting subboxes is then, respectively,

$$L_i = L \times \left(q \frac{n_i}{n_1 + n_2} + (1 - q) \frac{V_i}{V_1 + V_2} \right), \quad i = 1,2. \tag{1}$$

Here, n_i is the number of data points contained in subbox i, and V_i is the volume of subbox i. The value of the heuristic parameter q is constrained to the range [0.5,0.7]. Hyperboxes will not be subdivided further if their size are smaller than a prespecified value.

Wan et al. proposed the variance-based method [21,22] which differs from the median-cut, center-cut, and mean-split methods in that the sum of the square errors in the projected distribution in one of the three color-component axes is repeatedly minimized (see Eq. (11) of [21]). Therefore, (three-dimensional) distortions of quantized images obtained using this method is usually low (though not minimal).

Braquelaire and Brun [4] used a similar strategy. Their method runs in $H_1H_2H_3$ color space ($H_1 = R + G$, $H_2 = R - G$, $H_3 = B - \frac{1}{2}(R + G)$), and minimizes the quadratic errors associated with the two subboxes yielded by splitting along the axis with the largest variance. In general, the median-cut, center-cut, mean-split, variance-based and Braquelaire's methods all require that the division planes are perpendicular to one of the color axes. This keeps the methods simple and their computation loads low. However, eliminating this restriction often leads to smaller quantization errors.

To address this, Orchard and Bouman proposed the binary splitting (BS) algorithm [16]. The BS algorithm uses a division plane that passes through the mean point of the color set being split and is perpendicular to the principal axis. They also proposed the erosion-based binary splitting (EBBS) method based on the BS technique [16] to improve image quality. The EBBS method takes the spatial relationship of colors into account and uses more quantization levels to represent colors which might yield false contours.

Balasubramanian et al. proposed the PQBS (binary splitting with prequantization) algorithm [2] to accelerate the BS algorithm execution speed. The PQBS algorithm uses a sophisticated data structure to store the average colors of the image pixels yielding the same first 7–8–6 bits of the R-G-B components, and uses these average colors as inputs to the BS algorithm. This prequantization stage reduces input data size while retaining image quality. The PQBS method also uses a spatial-activity measure to avoid false contours while designing its color palette.

Wu proposed a color quantization method [23] based on principal analysis and dynamic programming. Wu's method yields much lower quantization errors at the expense of added execution time compared to the median-cut and variance-based methods.

In addition to using the top-down approach mentioned above, some methods [7,8,26] use a bottom-up approach. The bottom-up approach forms the color palette by progressively merging input colors until the number of remaining colors is equal to (or less than) the predetermined palette size.

Balasubramanian et al. proposed a sophisticated quantization method called SSQ (sequential scalar quantization) [3], which individually quantizes the scalar components of 3D color vectors, thereby generating color palettes more efficiently. The quality of quantized images is also high.

In this paper, we propose a method that repeatedly uses average color, variance, and radius preserving (ACVRP) bisection to quantize color images. The bisection method is applied in a hierarchically divisive manner. The idea about the ACVRP bisection method is that: whenever a set of color pixels is to be divided into two subsets, we always require that the representative colors of the two generated subsets, weighted by their corresponding population ratios, could preserve color distribution features of the set being split. Throughout this paper, the color distribution features preserved are: average color, variance in each color dimension, and average color radius. Average color is preserved (Eqs. (3)–(5) in Section 2) because this keeps the mean (the average brightness level) of each color component unchanged. The variance in each color dimension is preserved (Eqs. (6)–(8) in Section 2) so that the individual contrast of the red (R dimension), green (G dimension), and blue (B dimension) colors could be preserved in a certain sense, thus avoiding biasing the color hue. The average color radius is preserved (Eq. (9) in Section 2) to retain the so-called *color activity* ([1,13], respectively, evaluated in terms of l_1-norm and weighted l_2-norm where our evaluation is l_2-norm). (Note that an image with low color activity contains similar colors. A special case is an image with zero-value color activity, which means the image contains only one color.) By preserving these color distribution features, we hope our quantized images will approximate the original images in the viewers' eyes. Preserving average color is also used in dithering techniques (e.g., error diffusion [9]). Our method and dithering techniques differ in that our method tries to preserve average color in the color palette design procedure, whereas dithering tries to compensate (during color-mapping using a given palette) for the shift in the average color of newly mapped pixels. Such dithering is achieved at the cost of a substantial computational overhead, appearance of noise, etc., as indicated in [2,16]. (Note that dithering can be used as a post-processing after most color quantization methods [7,11,16,21,23].) Tsai [20], and Delp and Mitchell [6], proposed 1D (gray-valued) quantization to preserve moments (of the first, second, and third orders) of gray images. The novelty of our method compared to their methods is the generalization from 1D-scalar to 3D-vector quantization (from gray to color) and the use of 3D analytical formulas (derived in Section 2). The differences in constraints on cluster splitting between the proposed method and the other methods are: our constraints are intended to make the representatives of the generated clusters preserve the original data distribution, instead of balancing populations (e.g. median-cut) or minimizing (weighted) quantization errors (variance-based, EBBS, SSQ, etc.). The constraints we use can also yield 3D formulas for bisecting data analytically, thus, our bisection method is parameter-free and noniterative.

The proposed method uses the histogram-simplification procedure illustrated in Section 3 to reduce data size and decrease computation time. This procedure is a little similar to that used in the PQBS method (but we use the first 5–5–5 rather than the 7–8–6 bits when prequantizing input images because, as will be seen later in Tables 2 and 3, data size can be reduced 15 to 20 times smaller without degrading image quality much). Throughout this paper, we consider the proposed method to be run in RGB color space. It can also be run in other color spaces since its design is independent of color space. Some earlier studies showed that better results can often be obtained by using uniform color spaces [10,18], however, the price is an increase in the execution time since the transformation into uniform color spaces is somewhat time-consuming. Other studies [3,16] examined the nonlinear relationship between primary color input values and display intensities of typical monitors, and then compensated it. It should also be noted that using more quantization levels for the luminance term of the color signal can also increase image quality because human vision is more sensitive to changes in luminance than to changes in chromaticity [3,24].

The remainder of this paper is organized as follows. In Section 2, the ACVRP bisection method is illustrated. The whole system for color quantization, including histogram-simplification preprocessing, is introduced in Section 3. An algorithm is also given there to benefit readers. Experimental results are shown in Section 4 and a summary is given in Section 5.

2. ACVRP: a bisection method using average color, variance, and radius preserving

Our method repeatedly bisects data in a hierarchically divisive manner. Below, we introduce the bisection tool, ACVRP, designed for the method we propose.

Suppose an N-point set in RGB color space $S = \{(r_i, g_i, b_i)\}_{i=1}^{N}$ is to be split into two subsets S_A and S_B. Let $C_A = (r_A, g_A, b_A)$ and $C_B = (r_B, g_B, b_B)$ be the respective representative colors of S_A and S_B. We need to find explicit formulas for computing C_A and C_B efficiently. Let the population percentages of S_A and S_B be $p_A = n_A/N$ and $p_B = n_B/N$, respectively. (Here, n_A and n_B are the respective numbers of points [counting multiplicities] contained in S_A and S_B.) Since there are eight unknowns: p_A, r_A, g_A, b_A, p_B, r_B, g_B, b_B, we need eight constraints. The first constraint we will consider is the natural requirement

$$p_A + p_B = 1 = 100\%. \tag{2}$$

In order to make $C_A = (r_A, g_A, b_A)$ and $C_B = (r_B, g_B, b_B)$, weighted, respectively, by p_A and p_B, preserve the average color of S, we also require that

$$p_A r_A + p_B r_B = \frac{1}{N}\sum_{i=1}^{N} r_i = \bar{r} = 0, \tag{3}$$

$$p_A g_A + p_B g_B = \frac{1}{N}\sum_{i=1}^{N} g_i = \bar{g} = 0, \tag{4}$$

$$p_A b_A + p_B b_B = \frac{1}{N}\sum_{i=1}^{N} b_i = \bar{b} = 0. \tag{5}$$

(For simplicity, we temporarily assume that the average color $(\bar{r}, \bar{g}, \bar{b})$ of the set S being bisected is $(0, 0, 0)$. This restriction will not be removed until the final paragraph of this section in which some formulas are useful in the case in which $(\bar{r},\bar{g},\bar{b}) \neq (0,0,0)$ are provided.) The remaining four constraints we use are

$$p_A r_A^2 + p_B r_B^2 = \frac{1}{N}\sum_{i=1}^{N} r_i^2 = \overline{r^2}, \tag{6}$$

$$p_A g_A^2 + p_B g_B^2 = \frac{1}{N}\sum_{i=1}^{N} g_i^2 = \overline{g^2}, \tag{7}$$

$$p_A b_A^2 + p_B b_B^2 = \frac{1}{N}\sum_{i=1}^{N} b_i^2 = \overline{b^2}, \tag{8}$$

and

$$p_A\sqrt{r_A^2 + g_A^2 + b_A^2} + p_B\sqrt{r_B^2 + g_B^2 + b_B^2} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{r_i^2 + g_i^2 + b_i^2} = \bar{R}. \tag{9}$$

Note that $\overline{r^2}, \overline{g^2}, \overline{b^2}$ are the color variances of S (because $(\bar{r}, \bar{g}, \bar{b}) = (0, 0, 0)$ has been assumed), and $\bar{R}$ is the average color radius of S. Using the eight equations listed in (2)–(9), we may now solve for p_A, (r_A, g_A, b_A), p_B, and (r_B, g_B, b_B). Eqs. (3)–(5) imply that

$$r_B = -\frac{p_A}{p_B} r_A, \tag{10}$$

$$g_B = -\frac{p_A}{p_B} g_A, \tag{11}$$

$$b_B = -\frac{p_A}{p_B} b_A. \tag{12}$$

Substituting Eqs. (10)–(12) in Eqs. (6)–(8), we obtain

$$\frac{p_A}{p_B} r_A^2 = \frac{p_A}{p_B}(p_B + p_A) r_A^2 = \overline{r^2}, \tag{13}$$

$$\frac{p_A}{p_B} g_A^2 = \frac{p_A}{p_B}(p_B + p_A) g_A^2 = \overline{g^2}, \tag{14}$$

$$\frac{p_A}{p_B} b_A^2 = \frac{p_A}{p_B}(p_B + p_A) b_A^2 = \overline{b^2}. \tag{15}$$

The analytical formulas for computing $C_A = (r_A, g_A, b_A)$ are therefore

$$r_A = \pm\sqrt{\frac{p_B}{p_A}\,\overline{r^2}}, \tag{16}$$

$$g_A = \pm\sqrt{\frac{p_B}{p_A}\,\overline{g^2}}, \tag{17}$$

$$b_A = \pm\sqrt{\frac{p_B}{p_A}\,\overline{b^2}}. \tag{18}$$

Once C_A is known, $C_B = (r_B, g_B, b_B)$ can also be evaluated using Eqs. (10)–(12). The remaining problem is then determining the signs of r_A, g_A, and b_A, and the values of p_A and p_B. To determine the values of p_A and p_B, note that Eq. (2) implies

$$p_B = 1 - p_A. \tag{19}$$

Substituting Eqs. (10)–(12) in (9), we obtain

$$2p_A\sqrt{r_A^2+g_A^2+b_A^2}=\bar{R},$$

i.e.,

$$4p_A^2(r_A^2+g_A^2+b_A^2)=\bar{R}^2.$$

Then, using Eqs. (16)–(18), we obtain

$$4p_Ap_B(\overline{r^2}+\overline{g^2}+\overline{b^2})=\bar{R}^2.$$

Using (19), we obtain

$$4p_A(1-p_A)=\bar{R}^2/(\overline{r^2}+\overline{g^2}+\overline{b^2}).$$

Hence,

$$4p_A^2-4p_A+[\bar{R}^2/(\overline{r^2}+\overline{g^2}+\overline{b^2})]=0.$$

Therefore,

$$p_A=\frac{1}{2}\pm\frac{1}{2}\sqrt{1-\frac{\bar{R}^2}{\overline{r^2}+\overline{g^2}+\overline{b^2}}}. \tag{20}$$

To prove p_A in Eq. (20) always yields real value, we must show that $\bar{R}^2\leqslant\overline{r^2}+\overline{g^2}+\overline{b^2}$. This is guaranteed by Schwartz's inequality and is proven in Appendix A.

Below, we show how to determine the signs of (r_A,g_A,b_A). Assume there are at least as many color points in S_A as there are in S_B, that is, $p_A\geqslant p_B$. Then

$$p_A=\frac{1}{2}+\frac{1}{2}\sqrt{1-\frac{\bar{R}^2}{\overline{r^2}+\overline{g^2}+\overline{b^2}}}. \tag{21}$$

To determine the signs of (r_A,g_A,b_A) in (16)–(18), we proceed as follows:

Step 1. Choose the color dimension from among $\{r,g,b\}$ that yields the maximal variance. Without loss of generality, r may be chosen, i.e. $\overline{r^2}=\max\{\overline{r^2}+\overline{g^2}+\overline{b^2}\}$. Check whether most of the color points (r_i,g_i,b_i) in S have positive r_i. If they do, set the sign of r_A positive; otherwise, set the sign of r_A negative.

Step 2. Proceed as follows for the remaining color dimensions (here, g and b):

a. Determine whether most of the color points (r_i,g_i,b_i) in S yield positive product r_ig_i. If they do, assign g_A the same sign r_A has; otherwise, assign g_A the opposite sign.

b. Determine whether most of the color points (r_i,g_i,b_i) in S yield positive product r_ib_i. If they do, assign b_A the same sign r_A has; otherwise, assign b_A the opposite sign.

Once values for $C_A=(r_A,g_A,b_A)$ and $C_B=(r_B,g_B,b_B)$ have been obtained using the procedure above, (16)–(18), and (10)–(12), then the partition separating subsets S_A and S_B can be considered a plane perpendicular to the line segment $\overline{C_AC_B}$ that cuts S into two subsets such that Np_A points fall in the subset containing the point $C_A=(r_A,g_A,b_A)$. However, finding this plane is quite time-consuming and may make the whole method inapplicable to real-time color quantization. Therefore, if we note that p_A and p_B are just estimated values, and bear in mind that what we really want is a way to rapidly bisect data, we then realize the faster way to partition data is to apply the nearest neighbor rule: a data point is assigned to S_A if and only if the data point is closer to (r_A,g_A,b_A) than to (r_B,g_B,b_B).

So far, we have assumed that the centroid $(\bar{r},\bar{g},\bar{b})$ of the set S is (0, 0, 0). We now remove this restriction. Whenever the $(\bar{r},\bar{g},\bar{b})$ of S is not (0, 0, 0), we create a new N-point set $\tilde{S}$, which is a translated version of the N-point set S, such that $\tilde{S}$ has the centroid (0, 0, 0). The formulas derived in this section can then be applied to $\tilde{S}$ to obtain the representative colors $\tilde{C}_A$ and $\tilde{C}_B$ for $\tilde{S}$. We may then translate $\tilde{C}_A$ and $\tilde{C}_B$ back to the original coordinate system to obtain the two desired representative colors C_A and C_B for S. Note also that it is then easy to prove that the final values of p_A, p_B, $C_A=(r_A,g_A,b_A)$ and $C_B=(r_B,g_B,b_B)$ obtained in this way (translation and back-translation) can also be expressed directly in terms of the input data S as follows: replace Eq. (21) with

$$p_A=\frac{1}{2}+\frac{1}{2}\sqrt{1-\frac{\overline{\Delta R}^2}{\overline{\Delta r^2}+\overline{\Delta g^2}+\overline{\Delta b^2}}}, \tag{22}$$

where

$$\overline{\Delta r^2}=\frac{1}{N}\sum_{i=1}^{N}(r_i-\bar{r})^2, \tag{23}$$

$$\overline{\Delta g^2} = \frac{1}{N}\sum_{i=1}^{N} (g_i - \bar{g})^2, \tag{24}$$

$$\overline{\Delta b^2} = \frac{1}{N}\sum_{i=1}^{N} (b_i - \bar{b})^2, \tag{25}$$

and

$$\overline{\Delta R} = \frac{1}{N}\sum_{i=1}^{N} \sqrt{(r_i - \bar{r})^2 + (g_i - \bar{g})^2 + (b_i - \bar{b})^2}. \tag{26}$$

The computation formula for p_B is still

$$p_B = 1 - p_A. \tag{27}$$

Then replace Eqs. (10)–(12) with

$$r_B = \frac{\bar{r} - p_A r_A}{p_B}, \tag{28}$$

$$g_B = \frac{\bar{g} - p_A g_A}{p_B}, \tag{29}$$

$$b_B = \frac{\bar{b} - p_A b_A}{p_B}. \tag{30}$$

Finally, replace Eqs. (16)–(18) with

$$r_A = \bar{r} \pm \sqrt{\frac{p_B}{p_A}\overline{\Delta r^2}}, \tag{31}$$

$$g_A = \bar{g} \pm \sqrt{\frac{p_B}{p_A}\overline{\Delta g^2}}, \tag{32}$$

$$b_A = \bar{b} \pm \sqrt{\frac{p_B}{p_A}\overline{\Delta b^2}}. \tag{33}$$

Of course, the signs in the three equations above can be determined by a rule analogous to (although not identical to) that in Steps 1 and 2 above. For example, the following sentences will replace Step 1, and an analogous statement will replace Step 2: Step 1′. Choose the color dimension from among $\{r,g,b\}$ that yields the maximal variance. Without loss of generality, we may let r be the chosen dimension, i.e., $\overline{\Delta r^2} = \max\{\overline{\Delta r^2}, \overline{\Delta g^2}, \overline{\Delta b^2}\}$. Determine whether most of the color points (r_i, g_i, b_i) in S have positive $r_i - \bar{r}$. If they do, use a plus sign in Eq. (31) to evaluate r_A; otherwise, use a minus sign in Eq. (31).

3. The proposed method

Our quantization process has two parts: preprocessing, which we call *histogram simplification*, and the main procedure, which we call *hierarchical divisive quantization*. Each pixel in input image P has a color value expressed as a 24-bit binary number (8 bits for each of the three color components). Part 1 transforms P to P', and this P' is for use as the input to Part 2. In Part 1, we group together those color points whose 5–5–5 most significant bits of the r–g–b-component have identical values. (In other words, each group is formed of color points so similar to one another that their 15-bit expressions (5–5–5) are identical, their 24-bit expressions (8–8–8) might differ.) Assume there are J such groups, e.g., $P = G_1 \cup G_2 \cup \cdots \cup G_J$. We then compute the 24-bit average color V_j for each group G_j, which will yield a simplified histogram for P:

$$P' = \bigcup_{j=1}^{J} \{V_j, V_j, V_j, \ldots (V_j \text{ repeats } |G_j| \text{ times})\},$$

where $|G_j|$ denotes the number of points contained in G_j. Note that Part 1 reduces the number of distinct colors in a 512×512 image from, at most, $512 \times 512 = 262{,}144$ to at most $2^{5+5+5} = 32{,}768$. We found that using the output P' of Part 1 (instead of the original image P) as the input to Part 2 only slightly degrades the reconstructed image quality; however, the computation load in Part 2 is reduced significantly due to the simplification job done in Part 1.

In Part 2 (the hierarchically divisive quantization part), we hierarchically apply the ACVRP bisection method $k-1$ times to divide P', i.e. the sampled colors $\{V_1, V_1, \ldots, V_1\}$; $\{V_2, V_2, \ldots, V_2\}$; …; $\{V_J, V_J, \ldots, V_J\}$ (counting multiplicities) obtained in Part 1, into k subsets (usually $k = 256$). The representative colors of the k subsets are also generated in Part 2 to form the desired k-color palette. The algorithm for this is given below.

Algorithm: The proposed color quantization method (ACVRP-cut).

Input: An integer k and a full-color (24-bit) image $P = \{(r_i, g_i, b_i) \mid i = 1, \ldots, |P|\}$. (Each color is

described in 24-bit, and there are at most $|P|$ distinct colors.)

Output: A quantized image Q containing k colors. (Each color is described in 24-bit, and there are only k distinct colors. $k \ll |P|$.)

Step 1. Reduce P to P' using the histogram simplification procedure, i.e., Part 1 described in the proceeding paragraphs. (Note that each point in P' is still described in 24-bit.)

Step 2. Initially, let $S = P'$ and goto Step 4.

Step 3. Choose from among the existing subsets the subset S whose $\sum_{(r,g,b)\in S}[(r-\bar{r})^2 + (g-\bar{g})^2 + (b-\bar{b})^2]$ is the largest. Here, $(\bar{r},\bar{g},\bar{b})$ is the *local* centroid of the subset S. (Note that each subset has its own local centroid.)

Step 4. Use the $\overline{\Delta r^2}$, $\overline{\Delta g^2}$, $\overline{\Delta b^2}$, and $\overline{\Delta R}^2$ (defined in Eqs. (23)–(26)) of S to evaluate $C_A = (r_A, g_A, b_A)$ and $C_B = (r_B, g_B, b_B)$ according to Eqs. (22), (27), and (31)–(33), followed by (28)–(30).

Step 5. Use the two representative colors C_A and C_B to bisect S into the two subsets according to the nearest-neighbor rule.

Step 6. Repeat Steps 3–5 until there are k subsets.

Step 7. Output the k representatives colors $\{C_1, \ldots, C_k\}$ of these k existing subsets as the expected k colors.

Step 8. Do the pixel mapping $P \to \{C_1, \ldots, C_k\}$ according to the following rule: if Part 1 groups a pixel $(r_i, g_i, b_i) \in P$ into, say, group G_j, and if V_j, which is the average color of the group G_j, is finally located in the mth subset (of the k final subsets mentioned in Step 7), then the input pixel (r_i, g_i, b_i) should be painted using the mth representative color C_m.

4. Experimental results

In this section we show the good cluster-splitting capacity of the proposed ACVRP cut, and compare it with the cluster-splitting capacities of the other methods. To make illustration easier, the 2D projection of the Chernoff Fossil data and Fisher Iris data were used in the experiment. These two data sets have often been used in the field of data analysis [5], and we scaled the 2D projections of these two data such that their distributions could be clearly illustrated.

Fig. 1 shows the result of splitting the Chernoff Fossil data into three clusters, and Fig. 2 shows the result of splitting the Fisher Iris data into two clusters. (Note that, as mentioned in the Introduction, BS algorithm was used in both EBBS and PQBS.) The solid lines denote the cut planes, and the numbers 1 and 2 in Fig. 1 indicate the first and second cuts, respectively. The corresponding root-mean-square error (RMSE), a commonly used measure in data-clustering, and the number of misclassified points, are both listed in Table 1. Here, RMSE is the root-mean-square l_2-norm distance of each point to the center of the class that the point belongs to. Figs. 1 and 2 both show that the proposed method made more reasonable splittings than the other methods. Table 1 also shows the same holds true for quantity values.

Four experimental applications of the proposed ACVRP cut to color quantization were performed using Lena, Boats, Peppers, and Zelda as the input images (see Fig. 3). The results yielded by the median-cut [11,14], mean-split [22,25], variance-based [21,22], center-cut [12], EBBS [16], and PQBS [2] methods are included for purposes of comparison. When using the median-cut, mean-split, variance-based, and center-cut methods, we followed the original suggestions made by the authors of these methods for prequantizing images, i.e., 3–3–3 bit-cutting for the first three methods and 3–2–4 bit-cutting for the center-cut method. When using the EBBS algorithm, we prequantized the images using the proposed histogram simplification procedure (5–5–5 grouping) so that the EBBS method could be compared on equal terms with our method. When using the PQBS algorithm, we followed the prequantization method the authors of PQBS suggested, that is, 7–8–6 grouping. Table 2 shows the number of colors obtained by each prequantization method for the four images in Fig. 3. We note the following with respect to implementing these methods: (1) we used $q = 0.6$ in Eq. (1), and set the minimum hyperbox size equal to 0 for the mean-split method; (2) we used the nearest-neighbor rule (choosing the closest of the k representative colors) to do pixel mapping for the median-cut, mean-split, variance-based and center-cut methods; (3) erosion-based weighting was applied in the last one-third of the runs, as the authors

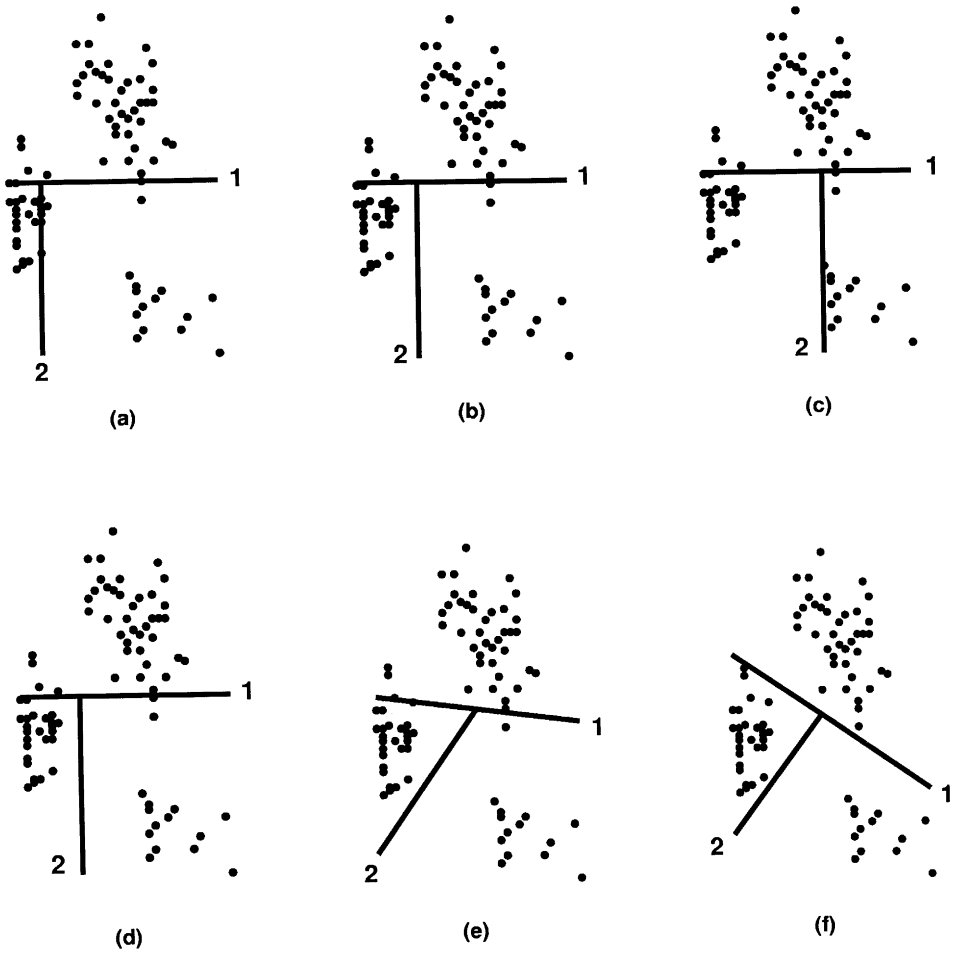

Fig. 1. Three-class clustering results for the Chernoff Fossil data using (a) median-cut, (b) mean-split, (c) variance-based, (d) center-cut, (e) binary splitting (BS) algorithm, and (f) our method.

suggested, for the EBBS method; (4) we directly allocated a huge amount of memory ($2^7 \times 2^8 \times 2^6$) to the PQBS method in order to simplify the prequantization procedure, and reduced the time needed to store and access the prequantized colors; (5) all the methods were implemented in C on an SGI Indygo2 workstation. Fig. 4 shows the results of quantizing Lena into 256 colors. Like EBBS and PQBS, the proposed method generated acceptable color smoothness on both the arm and the right cheek of Lena. Center-cut gave the worst image quality among the seven methods. The results from the median-cut and variance-based methods were better than that of the mean-split method, although the color smoothness on Lena's arm (from the median-cut method) and Lena's right cheek (from the variance-based method) were not very good.

Table 3 lists the results quantizing the images in Fig. 3 into 256 colors from all methods. The results were measured using RMSΔE (root-mean-square ΔE). In the field of image processing, this measurement is considered closer to the human perception [18] (than RMSE). Note that

$$\mathrm{RMS}\Delta\mathrm{E} = \sqrt{\frac{1}{M \times N} \sum_{i=1}^{M \times N} \Delta\mathrm{E}(L^*_{P_i}, a^*_{P_i}, b^*_{P_i}, L^*_{Q_i}, a^*_{Q_i}, b^*_{Q_i})^2}$$

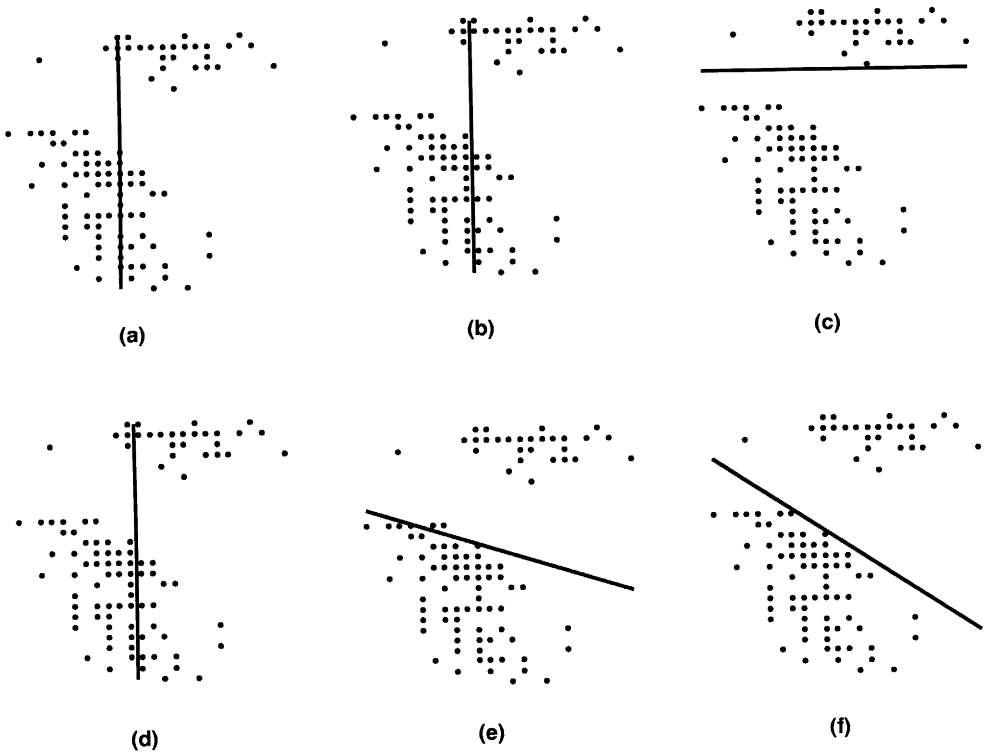

Fig. 2. Two-class clustering results for the Fisher Iris data using (a) median-cut, (b) mean-split, (c) variance-based, (d) center-cut, (e) binary splitting (BS) algorithm, and (f) our method.

Table 1
RMSEs and numbers of mis-classified data points for the various methods

	Median-cut	Mean-split	Variance-based	Center-cut	BS algorithm	Our method
Fossil	36.22/13	29.32/6	29.32/6	29.32/6	28.18/4	26.24/0
Iris	79.37/25	76.75/17	50.46/0	76.75/17	51.33/2	50.46/0

and

$$\Delta E(L^*_{P_i},a^*_{P_i},b^*_{P_i},L^*_{Q_i},a^*_{Q_i},b^*_{Q_i}) = \sqrt{(L^*_{P_i}-L^*_{Q_i})^2+(a^*_{P_i}-a^*_{Q_i})^2+(b^*_{P_i}-b^*_{Q_i})^2}.$$

Here, $M \times N$ denotes the image size; $(L^*_{P_i},a^*_{P_i},b^*_{P_i})$ and $(L^*_{Q_i},a^*_{Q_i},b^*_{Q_i})$ denote the $L^*a^*b^*$ color values of the i-th pixel in the input image and quantized image.

The computation times listed in Table 3 were measured in CPU seconds. Apparently, finer prequantization did help the PQBS method obtain lower RMSΔEs than the other methods yielded. (After prequantization, the number of prequantized colors the PQBS used was about 15–20 times larger than that of the other methods [see Table 2].) The EBBS method also yielded low RMSΔEs. However, the execution time for the EBBS and PQBS

Fig. 3. The four full-color images (left to right, top to bottom): Lena, Boats, Peppers, and Zelda.

Table 2
Numbers of distinct colors using various prequantization methods

	3–2–4 bit-cutting	7–8–6 grouping	5–5–5 grouping and 3–3–3 bit-cutting
Lena	2650	54063	2596
Boats	2451	39561	2353
Peppers	5113	86816	4972
Zelda	2987	50925	2781

Table 3
RMSΔEs and execution time (in s) for the various methods. Images were quantized to 256 colors. The * line shows the average RMSΔEs and time for the four lines above it

	Median-cut	Mean-split	Variance-based	Center-cut	EBBS	PQBS	Our method
Lena	3.34/1.02	3.42/1.04	3.22/1.18	4.25/1.05	2.98/5.09	2.78/5.77	2.92/0.91
Boats	7.32/0.97	6.77/0.95	6.35/1.08	13.34/1.02	3.50/4.45	3.34/4.40	3.38/0.88
Peppers	6.90/1.70	5.79/1.69	5.91/1.84	7.82/1.75	5.83/7.17	5.66/8.07	5.69/1.10
Zelda	6.98/1.07	7.04/1.08	4.03/1.20	10.67/1.15	3.59/5.81	3.56/5.62	3.57/0.95
(*	6.14/1.19	5.76/1.19	4.88/1.33	9.02/1.24	3.98/5.63	3.84/5.97	3.89/0.96)

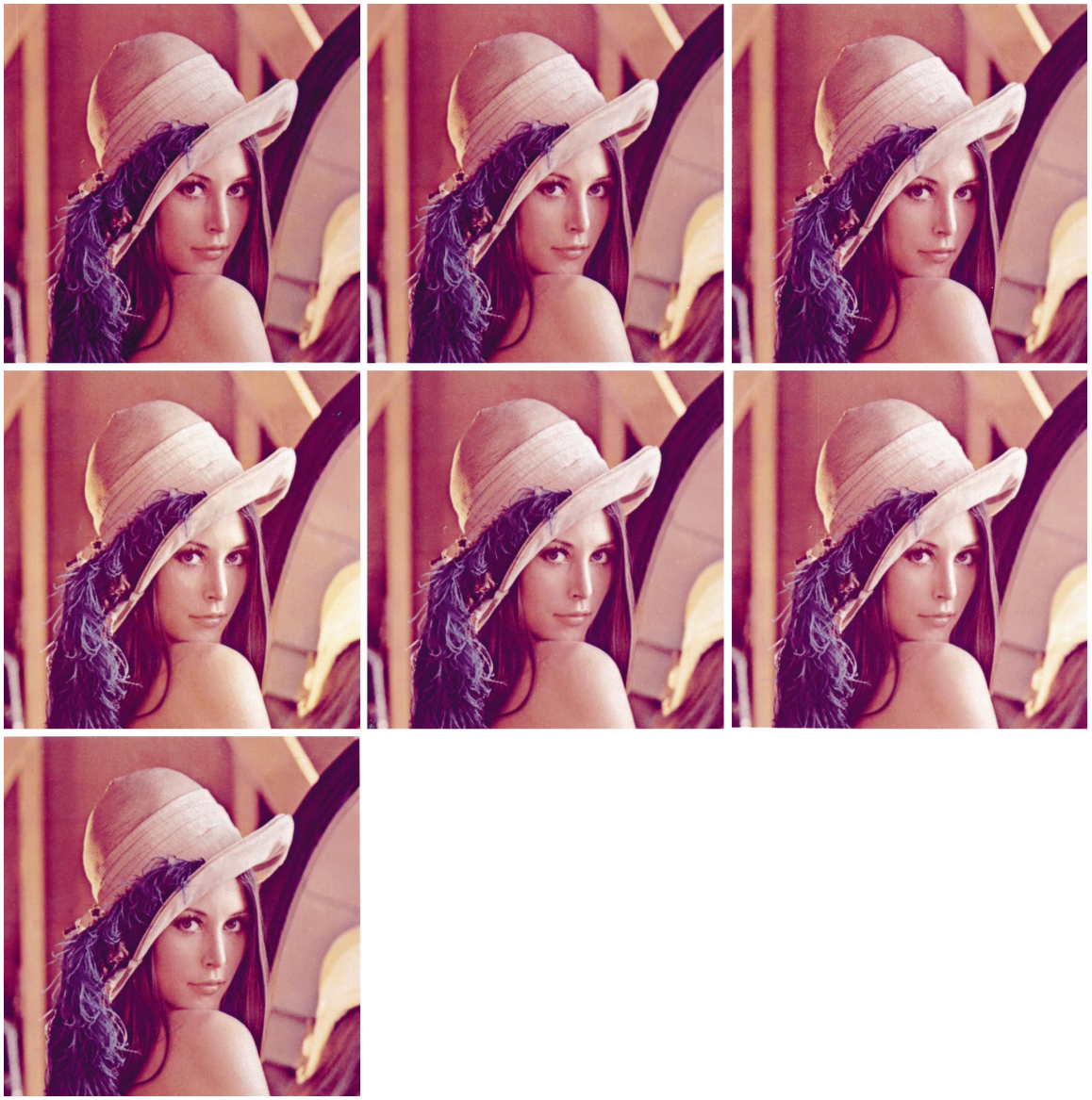

Fig. 4. 256-color quantized images of Lena obtained from the various methods (left to right, top to bottom): median-cut, mean-split, variance-based, center-cut, EBBS, PQBS, and our method.

methods were higher than those of the other methods. On the other hand, although the proposed method used coarser prequantization than the PQBS method, the RMSΔEs of the proposed method was close to that of the PQBS method. (In fact, in RMSΔE terms, the proposed method was a little worse than PQBS, but a little better than EBBS, and much better than the other four methods.) The execution speed of the proposed method was the best (similar to the remaining four methods, but faster than EBBS and PQBS). Therefore, the proposed method represents a good compromise between image quality and execution time.

5. Summary

The color quantization method proposed here repeatedly uses a bisection method called ACVRP, which preserves average color, variance in each color dimension, and average color radius of the data set being bisected. To get a k-color palette, a histogram-simplification procedure is first used to transform the input image P into a simpler form P'. The ACVRP bisection method is then applied $k-1$ times in a hierarchically divisive manner such that P' is split to k subsets. The k colors of the palette, which are the k representative colors of these k subsets, are automatically generated by analytical formulas. In the final step (pixel-mapping), it is not necessary to use the common nearest-neighbor rule to reduce quantization errors because our partition result (the k obtained subsets) are good enough so no re-allocation of pixels from one subset to another is needed to make our errors competitive. As shown by the experimental results, images quantized using the proposed method were visually acceptable. In commercial application, i.e., quantizing images into 256 colors, the proposed method usually gave low RMSΔEs and attractive execution speed.

Appendix A

$\bar{R}^2 \leqslant \overline{r^2} + \overline{g^2} + \overline{b^2}$.

Proof. According to Eq. (9),

$$\bar{R}^2 = \left(\frac{1}{N}\sum_{i=1}^{N}\sqrt{r_i^2+g_i^2+b_i^2}\right)^2 = \left(\frac{1}{N}\sum_{i=1}^{N} R_i\right)^2, \tag{A.1}$$

where $R_i = \sqrt{r_i^2+g_i^2+b_i^2}$. On the other hand,

$$\overline{r^2}+\overline{g^2}+\overline{b^2} = \left(\frac{1}{N}\sum_{i=1}^{N} r_i^2\right)+\left(\frac{1}{N}\sum_{i=1}^{N} g_i^2\right)+\left(\frac{1}{N}\sum_{i=1}^{N} b_i^2\right) = \frac{1}{N}\sum_{i=1}^{N}(r_i^2+g_i^2+b_i^2) = \frac{1}{N}\sum_{i=1}^{N} R_i^2. \tag{A.2}$$

Let $\boldsymbol{X} = (1/N,\ldots,1/N)$, and $\boldsymbol{Y} = (R_1,R_2,\ldots,R_N)$ be two N-dimensional vectors. Then

$$\bar{R}^2 = \left(\frac{1}{N}\sum_{i=1}^{N} R_i\right)^2 = \left(\left(\frac{1}{N},\frac{1}{N},\ldots,\frac{1}{N}\right)\cdot(R_1,R_2,\ldots,R_N)\right)^2 = (\boldsymbol{X}\cdot\boldsymbol{Y})^2 \leqslant (\|\boldsymbol{X}\|\|\boldsymbol{Y}\|)^2 = \|\boldsymbol{X}\|^2\|\boldsymbol{Y}\|^2 = \left[N\left(\frac{1}{N^2}\right)\right](R_1^2+R_2^2+\cdots+R_N^2) = \frac{1}{N}\sum_{i=1}^{N} R_i^2 = \overline{r^2}+\overline{g^2}+\overline{b^2},$$

where the inequality is due to Schwartz's inequality.

References

[1] R. Balasubramanian, J.P. Allebach, A new approach to palette selection for color images, Proc. SPIE: Human Vision Visual Process. Digital Display III 1453 (1991) 58–69.

[2] R. Balasubramanian, J.P. Allebach, C.A. Bouman, Color-image quantization with use of a fast binary splitting technique, J. Opt. Soc. Amer. A 11 (11) (1994) 2777–2786.
[3] R. Balasubramanian, C.A. Bouman, J.P. Allebach, Sequential scalar quantization of color images, J. Electron. Imaging 3 (1) (1994) 45–59.
[4] J.P. Braquelaire, L. Brun, Comparison and optimization of methods of color image quantization, IEEE Trans. Image Process. 6 (7) (1997) 1048–1052.
[5] Y.T. Chien, Interactive Pattern Recognition, Marcel Dekker, New York, 1978, pp. 226–230.
[6] E.J. Delp, O.R. Mitchell, Moment preserving quantization, IEEE Trans. Commun. 39 (11) (1991) 1549–1558.
[7] S.S. Dixit, Quantization of color images for display printing on limited color output devices, Comp. Graph. 15 (4) (1991) 561–567.
[8] W.H. Equitz, A new vector quantization clustering algorithm, IEEE Trans. Acoust. Speech Signal Process. 37 (10) (1989) 1568–1575.
[9] R.W. Floyd, L. Steinberg, An adaptive algorithm for spatial gray scale, Proc. SID 17 (2) (1976) 75–77.
[10] R.S. Gentile, J.P. Allebach, E. Walowit, Quantization of color images based on uniform color spaces, J. Imaging Technol. 16 (1) (1990) 12–21.
[11] P. Heckbert, Color image quantization for frame buffer display, Comp. Graph. 16 (1982) 297–307.
[12] G. Joy, Z. Xiang, Center-cut for color-image quantization, The Visual Comput. 10 (1993) 62–66.
[13] K.M. Kim, C.S. Lee, E.J. Lee, Y.H. Ha, Color image quantization and dithering method based on human visual system characteristics, J. Imaging Sci. Technol. 40 (6) (1996) 502–509.
[14] A. Kruger, Median-cut color quantization, Dr. Dobb's Journal, September 1994, 46–92.
[15] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantifier design, IEEE Trans. Commun. 28 (1980) 84–95.
[16] M.T. Orchard, C.A. Bouman, Color quantization of images, IEEE Trans. Signal Process. 39 (12) (1991) 2677–2690.
[17] P. Scheunders, A genetic c-means clustering algorithm applied to color image quantization, Pattern Recognition 30 (6) (1997) 859–866.
[18] K.E. Spaulding, L.A. Ray, J.R. Sullivan, Secondary quantization of color images for minimum visual distortion, Proc. SPIE: Human Vision Visual Proc. Digital Display IV 1913 (1993) 261–269.
[19] T. Taşdizen, L. Akarun, C. Ersoy, Color quantization with genetic algorithms, Signal Processing: Image Communication 12 (1998) 49–57.
[20] W.H. Tsai, Moment-preserving thresholding: a new approach, Comp. Vision Graphics Image Process. 29 (1985) 377–393.
[21] S.J. Wan, P. Prusinkiewicz, S.K.M. Wong, Variance-based color image quantization for frame buffer display, Color Res. Appl. 15 (1990) 52–58.
[22] S.J. Wan, S.K.M. Wong, P. Prusinkiewicz, An algorithm for multidimensional data clustering, ACM Trans. Math. Software 14 (1988) 153–162.
[23] X. Wu, Color quantization by dynamic programming and principal analysis, ACM Trans. Graph. 11 (4) (1992) 348–372.
[24] X. Wu, YIQ vector quantization in a new color palette architecture, IEEE Trans. Image Process. 5 (2) (1996) 321–329.
[25] X. Wu, I.H. Witten, A fast k-means type clustering algorithm, Technique Report, Department of Computer Science, University of Calgary, Canada, 1985.
[26] Z. Xiang, G. Joy, Color image quantization by agglomerative clustering, IEEE Comput. Graph. Appl. 1 (1994) 44–48.