Color Quantization by Dynamic Programming and Principal Analysis

XIAOLIN WU
University of Western Ontario

Color quantization is a process of choosing a set of K representative colors to approximate the N colors of an image, $K \ll N$, such that the resulting K-color image looks as much like the original N-color image as possible. This is an optimization problem known to be NP-complete in K. However, this paper shows that by ordering the N colors along their principal axis and partitioning the color space with respect to this ordering, the resulting constrained optimization problem can be solved in $O(N + KM^2)$ time by dynamic programming (where M is the intensity resolution of the device).

Traditional color quantization algorithms recursively bipartition the color space. By using the above dynamic-programming algorithm, we can construct a globally optimal K-partition, $K > 2$, of a color space in the principal direction of the input data. This new partitioning strategy leads to smaller quantization error and hence better image quality. Other algorithmic issues in color quantization such as efficient statistical computations and nearest-neighbor searching are also studied. The interplay between luminance and chromaticity in color quantization with and without color dithering is investigated. Our color quantization method allows the user to choose a balance between the image smoothness and hue accuracy for a given K.

Categories and Subject Descriptors: I.3.3 [**Computer Graphics**]: Picture/Image Generation; I.4.1 [**Image Processing**]: Digitization—*quantization*; I.5.3 [**Pattern Recognition**]: Clustering

General Terms: Algorithms

Additional Key Words and Phrases: Algorithm analysis, clustering, color quantization, dynamic programming, principal analysis

1. INTRODUCTION

To render continuous-tone color images on CRT displays, 24 bits (one byte for each of the primary colors: red, green, and blue) are usually used to represent the color of each pixel. However, most images contain only a small subset of the sixteen million colors distinguishable by this encoding scheme. Very often, 256 or fewer carefully chosen representative colors from an image

The research was supported by grants from the Natural Science and Engineering Council of Canada.
Author's address: Department of Computer Science, University of Western Ontario, London, Ontario, Canada N6A 5B7.

suffice to reproduce the image without a noticeable loss of color fidelity. The process of selecting a small number of representative colors from an image of higher color resolution is called *color image quantization*. Color image quantization is necessary when displaying continuous-tone color images on color monitors that lack 24-bits/pixel full-color frame buffers. Even in the future, when 24-bits/pixel frame buffers become affordable for all users, color quantization will still retain its practical value, since it can relieve valuable frame buffer space for animation, transparency, window applications, and other graphics functions. Furthermore, color image quantization lessens the burden of massive image data on storage and transmission bandwidth that are bottlenecks in many applications, particularly when computer graphics is integrated into multimedia systems or when HDTV technology becomes commonplace.

In 1982, Heckbert proposed the popular median-cut algorithm for color quantization [18]. His algorithm recursively splits the RGB color space into two subsets of equal color population using orthogonal cutting planes until K rectangular boxes are formed. This structure of recursive bipartitioning is identical to that of k-d trees [10]. Within the same k-d tree framework, Wu and Witten [43] changed the partitioning criterion from median-cut to variance minimization for better quantizer performance. They only gave a mean-based approximation algorithm for variance minimization. Later, Wu devised a fast exact algorithm for variance minimization [42] for partitioning color spaces. Meanwhile, Wan et al. extended the work of [43] in the direction of marginal variance minimization [39], i.e., sweeping a cutting plane perpendicular to the R, G, and B axes separately and splitting the current box at the position where the variance of the marginal distribution in the corresponding axis is minimized. The above algorithms have a common drawback that is inherent in tree-structured recursive bipartitioning. Namely, intermediate subsets are bipartitioned one at a time in total isolation from each other. As a result, the intercluster interactions are ignored in attempting to minimize the total quantization error. To correct this fundamental flaw, we develop a new, color-space-partitioning strategy to simultaneously optimize multiple cuts orthogonal to the principal axis of the color distribution. The new partitioning strategy enhances quantizer performance, since it permits minimizing quantization error across a much broader scope than the k-d tree structure allows. Other new results in this paper are: the development of new algorithmic techniques for efficient computations of color statistics and the elimination of the prequantization step, which was required for efficient implementation of previous algorithms.

There exist other color quantization techniques in the computer graphics literature that are somewhat less relevant to this work, including the peano curve scan [37], BTC coding [3], simulated annealing [6], and an octree-based method [15]. Color quantization/coding has been studied also by many telecommunication researchers [9, 20, 30, 36]. In particular, color quantization is an instance of vector quantization, a very active research field since the late seventies in the communities of image coding and information theory. The rich literature on vector quantization theory and techniques (see

the reference list of [14]) should prove useful to color quantization researchers and practitioners in computer graphics. Historically, color quantization researchers in image coding and computer graphics have had different emphases. The former group aim primarily at reducing the bandwidth for transmitting color video signals, whereas the latter group focus on frame buffer size reduction with the hardware support of color lookup tables. Besides its other uses in interactive graphics, color lookup tables provide so far the simplest and hence fastest decoding scheme of compressed color images onto a CRT display. This is probably why the quantization algorithms based on color lookup tables gained their popularity in interactive graphics where inexpensive real-time color decoding is a must, even though their compression ratios are modest when compared with what is achievable by other color-coding techniques [20].

This paper is organized as follows. In the next section, we formulate optimal color quantization as a discrete optimization problem and reveal its NP-completeness. In turn, this raises the interest in approximation algorithms to the problem. In Section 3, we expose the inherent drawbacks of recursive, orthogonal bipartitioning of the color space, which is common to many existing color quantization algorithms. Overcoming these drawbacks motivates our research. Section 4 contains the most important results of this paper. In it, an optimal K-partition ($K > 2$) of the color space normal to the principal axis of the data set is proposed for designing better color quantizers. The process of finding such a K-partition is called optimal principal multilevel quantization. A dynamic-programming algorithm is developed for finding this partition. Complexity analysis is given to demonstrate that the new optimization strategy can be made efficient enough to be practical. In Section 5, a locally optimal bipartitioning technique is integrated into the dynamic-programming scheme to finish the color quantizer design process. Section 6 covers nearest-neighbor-searching techniques for mapping input colors to their best representatives. Color dithering to improve the quality of quantized images is also discussed. Experimental results and observations on the perceptual behavior of the new algorithm are given in Section 7. Also, these results are compared with those of Heckbert's algorithm [18] and Wan et al.'s algorithm [39].

2. PROBLEM FORMULATION

It was psychovisually established that color is trivariant long before the birth of computer graphics. All current color models are defined in a three-dimensional space. Thus, a color image of N pixels corresponds to a set S of N points, $\mathbf{c}_i = (c_{i0}, c_{i1}, c_{i2})$, $1 \le i \le N$, in a three-dimensional color space such as RGB, YIQ, $L^*u^*v^*$, HSV, etc. In color image quantization, the point set S is partitioned into K subsets S_k, $1 \le k \le K$ ($S_k \neq \phi$, $S_j \bigcap_{j \neq k} S_k = \phi$, and $\bigcup_{1 \le k \le K} S_k = S$) where all colors $\mathbf{c} \in S_k$ are mapped to or approximated by, a representative color $\mathbf{z}_k = (z_{k0}, z_{k1}, z_{k2})$. Through this mapping an original 24-bit pixel c_i is represented by a $(\log_2 K)$-bit integer that is an index into a

color lookup table of K entries, one entry for each representative color. This simple, indirect addressing mechanism reduces the frame buffer size by a factor of $24/\log_2 K$, achieving a compression ratio of 3:1 for the typical table size $K = 256$.

Mathematically, color quantization can be formulated as a large-scale clustering problem. Our goal is to find the optimal K-partition of the set S of three-dimensional color points to minimize the quantization error

$$E(S_1, S_2, \cdots, S_K) = \sum_{1 \le k \le K} \left\{ \frac{1}{|S_k|} \sum_{\mathbf{c}_i, \mathbf{c}_j \in S_k,\, i<j} \|\mathbf{c}_i, \mathbf{c}_j\| \right\}, \tag{1}$$

where $\|\mathbf{c}_i, \mathbf{c}_j\|$ is a perceptually meaningful color distance between $\mathbf{c}_i$ and $\mathbf{c}_j$. The quantization error is defined to be the sum of in-cluster pairwise color dissimilarities. Note that in the above formulation we allow the set S to have duplicate elements to accommodate images that contain pixels having the same color. This relaxed setting eliminates the need of the color frequency function in (1); hence it saves a multiplication per pixel in evaluating the objective function. The total saving is significant because over 90% of the colors in a 24-bit natural image are distinct.

There are two important aspects of color quantization: a meaningful and computable color measure $\|\mathbf{c}_i, \mathbf{c}_j\|$ to quantify visual aesthetics and an efficient algorithm to minimize the quantization error E. Perceptual color measurement is a challenging color science problem and has attracted much attention [20, 22, 23, 26, 28, 38]. If the ubiquitous Euclidean metric is to be used in (1) then quantization needs to be performed in a perceptually uniform color space in which $\|\mathbf{c}_0, \mathbf{c}_1\| = \|\mathbf{c}_1, \mathbf{c}_2\|$ if $\mathbf{c}_0$ and $\mathbf{c}_1$ differ as much as $\mathbf{c}_1$ and $\mathbf{c}_2$ in visual sensation, and this quantified difference is independent of $\mathbf{c}_1$. Color quantization in perceptual color spaces instead of the device-oriented RGB space was proposed by Kurz [19] and Gentile et al. [13]. However, uniform color space alone does not suffice to quantify perceptual color distance since image context also plays an important role in human color vision. The relative positions in image space of different colors influence our color interpretation. Recently, Balasubramanian and Allebach [1] incorporated a color activity criterion into a prequantization scheme to account for different human observers' sensitivities to quantization errors in different color contexts. Unfortunately, this technique remains highly heuristic and far from offering a mathematical model for context dependency of colors to be integrated into an objective function such as (1) for clustering colors.

In the sequel it is assumed that color quantization is carried out in a perceptually uniform color space. We recommend the uniform $L^*u^*v^*$ color space (CIE 1976) in which the perceptual distance $\|\mathbf{c}_i, \mathbf{c}_j\|$ can be approximated by the Euclidean distance between $\mathbf{c}_i$ and $\mathbf{c}_j$. Note that even $L^*u^*v^*$ color space is not perfectly uniform. But it is better than the RGB space which was previously used solely for convenience. The transforms between $L^*u^*v^*$ and other color spaces can be found in many sources [8, 27, 28] and has been omitted here for brevity. Let $\|\mathbf{c}_i, \mathbf{c}_j\|$ be the Euclidean distance

between $\mathbf{c}_i$ and $\mathbf{c}_j$ in the $L^*u^*v^*$ space; then we have

$$\begin{aligned}\sum_{\mathbf{c}_i,\mathbf{c}_j\in s_k,\, i<j} \|\mathbf{c}_i,\mathbf{c}_j\| &= \sum_{\mathbf{c}_i,\mathbf{c}_j\in S_k,\, i<j} \left[(c_{i,0}-c_{j,0})^2+(c_{i,1}-c_{j,1})^2+(c_{i,2}-c_{j,2})^2\right] \\ &= |S_k| \sum_{\mathbf{c}_i\in S_k} \left[c_{i,0}^2+c_{i,1}^2+c_{i,2}^2\right] \\ &\quad -\left[\sum_{\mathbf{c}_i\in S_k} c_{i,0}\right]^2-\left[\sum_{\mathbf{c}_i\in S_k} c_{i,1}\right]^2-\left[\sum_{\mathbf{c}_i\in S_k} c_{i,2}\right]^2 \\ &= |S_k| \sum_{\mathbf{c}_i\in S_k} \left[(c_{i,0}-z_{k,0})^2+(c_{i,1}-z_{k,1})^2+(c_{i,2}-z_{k,2})^2\right]\end{aligned} \tag{2}$$

where

$$z_{k,j}=\frac{\sum_{\mathbf{c}_i\in S_k} c_{i,j}}{|S_k|}, \qquad j=0,1,2 \tag{3}$$

is the jth component of the centroid of S_k. Thus we simplify (1) to

$$E(S_1,S_2,\cdots,S_K)=\sum_{1\le k\le K}\sum_{\mathbf{c}_i\in S_k}\|\mathbf{c}_i,\mathbf{z}_k\|. \tag{4}$$

Clearly, the above formulation is an image-dependent but context-free color quantization scheme. The context-free treatment may compromise the subjective image quality, but it offers acceptable solutions in practice before a tractable, context-dependent color measure is known.

Minimizing (4) over all possible K-partitions is a K-clustering problem, which was shown to be NP-complete for variable K [2, 12, 26].[1] Consequently, any solution to optimal color quantization will necessarily be heuristic and approximate. From now on we will concentrate on an algorithmic approach to statistical color quantization. The study of color quantization algorithms is important in its own right, independent of color measures for two reasons: (a) color quantization as a frame buffer technique has to be performed under severe time constraints; hence algorithm efficiency is crucial; (b) a good strategy of minimizing (1) under one color measure may lead to an efficient quantization algorithm under a better color measure should it become available. In fact, the following algorithm developments are independent of color spaces.

3. BIPARTITIONING TECHNIQUES AND THEIR LIMITATIONS

Previous color quantization techniques [15, 18, 39, 43] share a common algorithmic structure, namely, recursive orthogonal bipartitioning of color space. Instead of optimizing the K-partition of the set S which is itself an

[1] In [39], the proof was credited to a wrong reference.

NP-complete problem, we try to compute (approximate) the optimal bipartition of S into S_1 and S_2, i.e., to minimize the sum $E(S_1) + E(S_2)$, where

$$E(S_j) = \sum_{\mathbf{c} \in S_j} \|\mathbf{c}, \mathbf{z}_j\|. \tag{5}$$

After S is cut into S_1 and S_2, the same bipartitioning is carried out on S_1 and S_2 and then to the subsequent subsets until K subsets (clusters) are generated. The partitioning process is structured as a k-d tree whose root is S, each of whose $K - 1$ internal nodes corresponds to a bipartition, and whose K leaves are the final K subsets S_j, $1 \le j \le K$. The K centroids $\mathbf{z}_j$ are the K representative colors to be loaded into the color lookup table. In quantization literature, quantizers embedded in BSP trees are called tree-structured vector quantizers [16].

The original k-d tree [10] partitions the space with orthogonal cutting halfplanes, and early color quantization algorithms [15, 18, 39, 43] adopted the orthogonal partitioning scheme. However, realistic color image data are not generally distributed orthogonally in the color space. It is intuitively clear that cutting halfplanes should be set normal to the principal axis along which the color points have the maximum variance rather than normal to one of the three axes, if our goal is to minimize $E(S_1) + E(S_2)$ when splitting S into S_1 and S_2. Indeed, this principal cutting strategy was successfully applied to color quantization by Wu [40]. The principal axis is given by the principal eigenvector of the covariance matrix of the input data. Arbitrary spatial partitioning for color quantization was reported also by Orchard and Bouman [30]. Historically, splitting a vector space with arbitrary halfplanes to cluster multivariate data dates back to 1963 [29]. This sort of spatial partitioning method is not new in computer graphics, either. The color-space-partitioning tree with arbitrary cutting halfplanes described above can be regarded as a particular type of binary spatial-partitioning tree (BSP tree). It was used by Fuchs et al. [11] for efficient visibility determinations. The apparent differences are only in the spaces in which the trees are constructed and the partitioning criteria. For convenience, we simply refer to the partitioning tree in the color space as the BSP tree.

In general, the order in which the BSP tree grows during the clustering process matters to the optimality of the resulting color quantizer. The simplest order of repeated bipartitionings is blind recursion. Before growing to the next level, each internal node of the binary tree at the current level is split regardless of its statistical characteristics. Smaller quantization distortions may be achieved by more elaborate tree-growing techniques. An obvious alternative is to split the node with the largest variance. Still a better criterion is to split the node whose bipartition yields the largest reduction in the total quantization distortion. More precisely, let Ω_j, $1 \le j \le k < K$, be the k current subsets subject to further subdivisions, and let $\Omega_{j,1}$ and $\Omega_{j,2}$ be the two subsets of Ω_j if it is split by the optimal cutting halfplane. Then the next subset to be split is Ω_t such that

$$t = \arg \max_{1 < j \le k} \{E(\Omega_j) - E(\Omega_{j,1}) - E(\Omega_{j,2})\}. \tag{6}$$

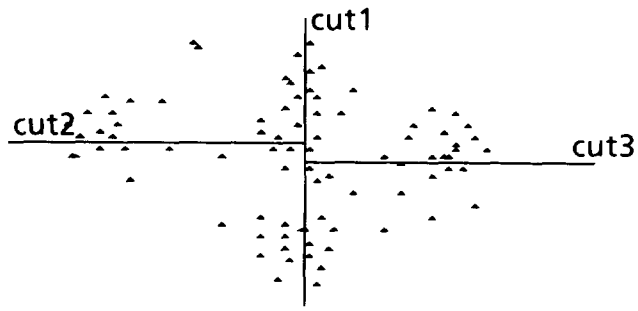


Fig. 1. A bad 4-partition formed by greedy bipartitions.

Although the algorithm combining principal cutting halfplanes and the look-ahead BSP tree construction by (6) [40, 44] can improve orthogonal cutting methods [18, 39], it still suffers from an extremely narrow optimization scope as its predecessors do. The optimality is compromised by the greedy bipartitioning strategy. Namely, the aim of each bipartitioning is the immediate profit of minimizing $E(S_1) + E(S_2)$, regardless of its impact on further subdivisions of S deeper in the BSP tree. This greedy local criterion may contradict the global criterion of minimizing (4). Even in two-dimensional space it is not difficult to find such adverse cases. For instance, in Figure 1, three successive greedy bipartitions clearly yield a bad four-partition of the data set.

Recently, Chou et al. [4] and Lin et al. [21] suggested an improvement to quantizer performance by growing a BSP tree of more than K leaves and then optimally pruning the tree back to a subtree of K leaves. But our experiments showed that the quantization distortion reduction by optimal tree pruning was minimal and hardly justified the incurred heavy computational cost. The main reason for the ineffectiveness of optimal tree pruning is that the decision is binary: a cut is either kept or removed, never adjusted. Furthermore, the top structure of the BSP tree remains immune from the optimal pruning.

The limited optimization scope is an inherent and severe drawback of a tree-structured greedy bipartitioning strategy. We need an approach that permits an optimization of multilevel partitioning in a global sense.

4. OPTIMAL PRINCIPAL MULTILEVEL QUANTIZATION

4.1 Motivation

First, we reveal a statistical characteristic of color image data beneficial to the design of our new color quantizer. Suppose, as in our previous algorithm [40, 44], that the data set S is split into S_1 and S_2 by the optimal cutting halfplane normal to the principal axis of S, and the BSP tree grows in a look-ahead-greedy fashion. Let $\{H_i\colon i = 1, 2, \cdots\}$ be the sequence of optimal cutting halfplanes generated by the above process. We observed consistently

that for some κ, the principal axes of the resulting subsets created by the first few cuts H_i, $1 \le i \le \kappa$, remained approximately the same as the principal axis of the original data set S. Consequently, the halfplanes H_i, $1 \le i \le i_0$, are almost parallel to each other and all approximately normal to the principal axis of S. The critical value of κ varied from four to eight depending on different color images. This finding is due to the fact that the color distribution of a natural scene is not isotropical in the three-dimensional space; rather, the color points spread out in the intensity direction (Y component in YIQ color space, V in HSV space, and L^* in $L^*a^*b^*$ space) much more than in the planes of the other two chromaticity dimensions. The same statistical nature of color image data was previously well documented by many researchers [3, 9, 17, 20, 31, 33], and it has inspired the following better heuristic for optimal color quantization.

4.2 Optimal Principal Quantizer

The performance of a color quantizer can be improved by simultaneously choosing H_i, $1 \le i \le \kappa$, rather than choosing H_i greedily and one at a time, in minimizing the quantization distortion. This accounts for the chain interactions between clusters ignored by the greedy approach and minimizes the quantization distortion in a much broader scope. The question is how this global optimization approach can be made computationally feasible. Fortunately, the fact that the halfplanes H_i, $1 \le i \le \kappa$, are almost parallel facilitates the following constrained global optimization scheme.

The basic idea is to optimize multiple cuts against the principal axis of the data set S. This axis can be determined by the classic principal-component analysis technique [24], that is, by finding the largest eigenvalue λ_{max} and the corresponding principal eigenvector $\mathbf{v}$ of the covariance matrix $\mathbf{C}$ of S, i.e., $\mathbf{Cv} = \lambda_{max}\mathbf{v}$. Then all N color points $\mathbf{c} \in S$ will be sorted by their projections on the principal axis given by $\mathbf{v}$, i.e., $\mathbf{c}_i \le \mathbf{c}_j$ iff $\mathbf{c}_i^T\mathbf{v} \le \mathbf{c}^T j\mathbf{v}$. In the computation, v needs to be normalized. The eigenvector transform and the sorting constitute a map $R : S \to \{1, 2, \cdots, N\}$, with $R(\mathbf{c}) = i$ meaning that the projection value $\mathbf{c}^T\mathbf{v}$ ranks i in the sorted list of N projections. Now we define a finite set

$$Q_n^k \equiv \{\mathbf{q} \mid 0 \equiv q_0 \le q_1 < q_2 < \cdots < q_{k-1} < q_k \equiv n\} \subset \aleph^{k+1}, \tag{7}$$

where $\aleph$ is the set of all natural numbers. Then a $\mathbf{q} \in Q_n^k$ corresponds to a k-partition of the point set $S(0, n] = \{\mathbf{c} : 0 < R(\mathbf{c}) \le n\}$ into subsets: $S(q_{i-1}, q_i] = \{\mathbf{c} : q_{i-1} < R(\mathbf{c}) \le q_i\}$, $1 \le i \le k$. Notice that the intervals involved are open on the left but closed on the right.

The vector $\mathbf{q} \in Q_n^k$ is called a $k : n$ principal quantizer since it quantizes n multivariate points into k parallel cells bounded by $k - 1$ cutting halfplanes normal to the principal axis of $S(0, n]$. The quantization distortion of the ith cell is

$$\mathscr{E}(q_{i-1}, q_i] = \sum_{\mathbf{c} \in S(q_{i-1}, q_i]} \|\mathbf{c}, \mathbf{z}_i\| \tag{8}$$

where $\mathbf{z}_i$ is the centroid of the point set $S(q_{i-1}, q_i]$. Given a critical parameter κ, the total distortion of a $\kappa : N$ principal quantizer $\mathbf{q} \in Q_N^\kappa$ is

$$E(\mathbf{q}) = \sum_{1 \le i \le \kappa} \sum_{\mathbf{c} \in S(q_{i-1},\, q_i]} \|\mathbf{c}, \mathbf{z}_i\|. \tag{9}$$

Under the above formulation, minimizing $E(\mathbf{q})$ over all possible $\kappa : N$ quantizers $\mathbf{q} \in Q_N^\kappa$ means minimizing the total quantization distortion over all possible κ-partitions of S generated by parallel halfplanes normal to the principal axis. We define the optimal $\kappa : N$ principal quantizer $\hat{\mathbf{q}}$ to be the one that minimizes (9), i.e., $\hat{\mathbf{q}} = \arg\min_{\mathbf{q} \in Q_N^\kappa} E(\mathbf{q})$.

It is important to realize that $E(\mathbf{q})$ is still defined in $\mathscr{R}^3$ not in $\mathscr{R}$. Thus the $k : n$ principal quantizer $\mathbf{q} \in Q_n^k$ is a *vector*, not a scalar quantizer. The optimal principal quantization can be considered as a hybrid method of two existing types of color quantization methods: component based and vector based. The component-based methods [9, 31, 36, 39] quantize color components independently as scalars, whereas vector-based methods [15, 18, 40, 43] quantize colors as trivariant data. The component-based methods give higher priority to the luminance component because the dynamic range of luminance is significantly larger than that of the chromaticity components and because errors in chromaticity are less visible than errors in luminance. Another advantage of the component-based methods is that each color component can be treated as a one-dimensional signal and can be quantized with computational ease. In contrast to the NP-completeness of optimal vector quantization, optimal $K : N$ scalar quantization was recently shown [34, 41] to be solvable in $O(KN)$ time. However, the combined result of separate quantization of luminance and chromaticity can be far off the optimal color quantizer since the chromaticity distribution usually varies at different luminance levels. Furthermore, due to the limited gamut of the physical devices [38] the achievable range of chromaticity varies also at different luminance levels. Note that the principal axis of S is very close to the luminance axis since luminance has a much larger dynamic range than chromaticity. The optimal principal quantization preserves the advantages of component-based quantization by setting cutting halfplanes normal to the principal axis of S, thus on a luminance-first basis, but not at the expense of chromaticity quantization since the multilevel partition is optimized under the original three-dimensional distortion measure, not a one-dimensional distortion measure.

4.3 Dynamic-Programming Algorithm

The size of the search domain for minimizing (9) is $|Q_N^\kappa| = \binom{N\text{-}1}{\kappa - 1} = O(N^{\kappa-1})$. Finding the optimum by enumeration is still intractable for modest N and κ, $\kappa \ll N$. A more sophisticated algorithm is needed.

The first t cells of a $k : n$ principal quantizer $\mathbf{q} \in Q_n^k$, $1 < t < k$, $k < n$, give a t-partition of the point set $S(q_0, q_t]$; thus by definition, they form a $t : q_t$ principal quantizer. It can be proven by contradiction that the first t cells of the optimal $\kappa : N$ principal quantizer $\hat{\mathbf{q}}$ must be the optimal $t : q_t$ principal

quantizer on the subset $S(\hat{q}_0, \hat{q}_l]$. This property, called the principle of optimality in the optimization literature,enables a dynamic-programming algorithm to compute the optimal $\kappa : N$ principal quantizer $\hat{\mathbf{q}}$, which is a generalization of the optimal scalar quantization process [41].

Denote by $\hat{\mathbf{q}}_n^k$ the optimal principal $k : n$ quantizer, and let $Ł[k, n]$ be the $(k-1)^{\text{th}}$ parameter of $\hat{\mathbf{q}}_n^k$, i.e., $Ł[k, n] \equiv (\hat{\mathbf{q}}_n^k)_{k-1}$. Then the principle of optimality can be expressed as

$$Ł[k, n] = \arg \min_{k \le i < n} \left\{ E(\hat{\mathbf{q}}_i^{k-1}) + \mathscr{E}(i, n] \right\}, \qquad 2 \le k < n \le N. \tag{10}$$

Hence $Ł[k, n]$ can be determined by a linear search provided that $E(\hat{\mathbf{q}}_i^{k-1})$, $k \le i < n$, are all known. Once $Ł[k, n]$ is determined $E(\hat{\mathbf{q}}_n^k)$ becomes known also by

$$E(\hat{\mathbf{q}}_n^k) = E\left(\hat{\mathbf{q}}_{Ł[k,n]}^{k-1}\right) + \mathscr{E}(Ł[k, n], n], \qquad 2 \le k < n \le N. \tag{11}$$

This suggests that $\hat{\mathbf{q}}_N^\kappa$ can be constructed by bottom-up dynamic programming. First note that the distortions of one-level quantizers are trivially $E(\hat{\mathbf{q}}_n^1) = \mathscr{E}(0, n]$, $1 \le n \le N$. Then, by (10) and (11), $Ł[2, n]$ and $E(\hat{\mathbf{q}}_n^2)$, $2 \le n \le N$, can be computed and stored as intermediate results for later use. In general, the dynamic-programming process determines $Ł[k, n]$ and $E(\hat{\mathbf{q}}_n^k)$, $k \le n \le N$, by referring to $E(\hat{\mathbf{q}}_n^{k-1})$, $k-1 \le n \le N$, and it remembers all the results recently obtained to facilitate the computations of $Ł[k+1, n]$ and $E(\hat{\mathbf{q}}_n^{k+1})$, $k+1 \le n \le N$. The process terminates when k has been incremented from 2 up to κ and when $Ł[\kappa, N]$ is finally obtained. The parameters of optimal $\kappa : N$ quantizer $\hat{\mathbf{q}}_n^\kappa$ can be reconstructed backward then from $\hat{\mathbf{q}}_\kappa = N$ and the relation $(\hat{\mathbf{q}}_N^\kappa)_i = Ł[i, (\hat{\mathbf{q}}_N^\kappa)_{i+1}]$. The pseudocode of the algorithm is given below.

Algorithm. *Optimal quantization by dynamic programming.*
Input: N, k, S, P.
Output: $\hat{\mathbf{q}}_N^\kappa$.
Globals: $E[n] \equiv E(\hat{\mathbf{q}}_n^{k-1})$, $Ł[k, n] \equiv (\hat{\mathbf{q}}_n^k)_{k-1}$;
Initialization: $E[n] := \mathscr{E}(0, n]$, $1 \le n \le N$; $Ł[k, k] := k - 1$, $1 \le k \le \kappa$.

```
begin
    for k := 2 to κ do
      for n := k + 1 to N − κ + k do begin
        cut := n − 1; e := E[n − 1];
        for t := n − 2 downto k − 1 do
        if E[t] + 𝓔(t, n] < e then begin
          cut := t; e := E[t] + 𝓔(t, n];
        end;
        Ł[k, n] := cut; E[n] := e;
      end;
    output Łchain(κ, N) as q̂_N^κ;
end.
function Łchain(k, n) : q ∈ Q_n^k;
begin
    t := n;
    for j := k − 1 downto 1 do q_j := t := Ł[j + 1, t];
    return(q);
end
```

4.4 Complexity Analysis

In this section we will study the time and space complexities for computing an optimal $\kappa : N$ principal quantizer $\hat{\mathbf{q}}_N^{\kappa}$. In general, all color points $\mathbf{c} \in S$ may have N distinct projection values $\mathbf{c}^T\mathbf{v}$, resulting in a huge search domain for the dynamic-programming algorithm. However, on second reflection, real arithmetic on the principal axis is unnecessary in practice since the intensity resolutions of frame buffer displays are discrete and have a small dynamic range, typically being integers from 0 to 255. Therefore, without loss of precision achievable by the digital devices, we put N projections into $M < N$ buckets and approximate $\hat{\mathbf{q}}_n^{\kappa}$ by $\hat{\mathbf{q}}_M^{\kappa}$. This allows also a linear-time ordering of the N projections by the bucket sort algorithm. In our experiments $M = 512$ was found sufficient. By comparison, the previous algorithms [18, 39] employed simple truncation of the least-significant bits of each component of the input data to reduce the problem size. However, this treatment sacrifices the optimality by cutting the precision of the algorithm below that of the device.

The following analysis is based on M not on N, although the algorithm was derived for N for conceptual clarity. In the kernel of the dynamic-programming algorithm, the quantization distortions $\mathscr{E}(a, b]$, $0 \le a \le b \le n$, are repeatedly evaluated. To gain efficiency, we can precompute $\mathscr{E}(a, b]$ for all possible pairs a and b and store them for future reference. Since there are $O(M^2)$ possible pairs a and b, such a preprocessing seemingly needs $O(M^3)$ time and $O(M^2)$ space. But the following manipulations lead to a linear-time scheme. Notice that

$$\mathscr{E}(a, b] = \sum_{a < R(\mathbf{c}) \le b} (\mathbf{c} - \mathbf{z})^T(\mathbf{c} - \mathbf{z}) = \sum_{d=0}^{2} \left\{ \sum_{a < R(\mathbf{c}) \le b} c_d^2 - \frac{[\Sigma_{a < R(\mathbf{c}) \le b} c_d]^2}{\Sigma_{a < R(\mathbf{c}) \le b} 1} \right\}. \tag{12}$$

Now define the quantities

$$\begin{aligned} W_2(n) &\equiv \sum_{0 < R(\mathbf{c}) \le n} \sum_{d=0}^{2} c_d^2, \\ W_1(d, n) &\equiv \sum_{0 < R(\mathbf{c}) \le n} c_d, \qquad 0 \le d \le 2, \\ W_0(n) &\equiv \sum_{0 < R(\mathbf{c}) \le n} 1, \end{aligned} \tag{13}$$

where all W_i are the ith cumulative moments, and use them to rewrite (12) as

$$\mathscr{E}(a, b] = W_2(b) - W_2(a) - \frac{\Sigma_{d=0}^{2}[W_1(d, b) - W_1(d, a)]^2}{W_0(b) - W_0(a)}. \tag{14}$$

Therefore, if the 0th, 1st, and 2nd cumulative moments in (13) are precomputed and stored for $1 \le n \le N$, $0 \le d \le 2$, then $\mathscr{E}(a, b]$ can be evaluated

in $O(1)$ time. Consequently, the inner t loop of the dynamic-programming algorithm can be executed in $O(n-k)$ time. Thus, the total computational cost of the dynamic-programming process is determined by the simple counting:

$$\sum_{k=2}^{\kappa} \sum_{n=k+1}^{M-\kappa+k} (n-k) = \frac{\kappa-1}{2}\left[(M-\kappa)^2 + M - \kappa\right]. \tag{15}$$

Clearly, $O(N)$ times suffices to precompute and save the quantities $W_2(n)$, $W_1(d,n)$, $d=0,1,2$, and $W_0(n)$, $1 \le n \le M$, and $O(\kappa)$ time is taken by the function Łchain(κ, M), both being insignificant in the presence of (15). The cost of the eigenvector transform prior to dynamic programming is dominated by the construction of the covariance matrix C, which takes $O(N)$ time. Computing the N projections on the principal axis and ordering those projections by a bucket sort each takes $O(N)$ time. Adding up all these costs, we conclude that the optimal κ: M principal quantizer can be computed in $O(\kappa M^2 + N)$ time considering that $\kappa \ll M$.

The space complexity of the dynamic-programming algorithm is $O(\kappa M)$ because the $\kappa \times M$ array Ł is maintained to reconstruct $\hat{\mathbf{q}}_M^\kappa$. If this space complexity is too high for a small machine, we can reduce it to $O(M)$ by a slightly more complicated logic and at a small penalty on execution time. In order not to lengthen the paper we only sketch the technique. Consider $\hat{\mathbf{q}} = \hat{\mathbf{q}}_M^\kappa$ and assume that κ is a power of 2 for simplicity. We first determine the middle cutting position of $\hat{\mathbf{q}}_M^\kappa$ by

$$\hat{\mathbf{q}}_{\kappa/2} = \arg \min_{\kappa/2 \le i \le M-\kappa/2} \left\{E(\mathbf{q}_i^{\kappa/2}) + E(\check{\mathbf{q}}_{M-i}^{\kappa/2})\right\}, \tag{16}$$

where $\check{\mathbf{q}}_{M-i}^{\kappa/2}$ is the optimal $\kappa/2:(M-i)$ principal quantizer on the subset $S(i, M]$. The dynamic-programming computation for $\check{\mathbf{q}}_{M-i}^{\kappa/2}$ is the same as for $\hat{\mathbf{q}}_i^{\kappa/2}$ on the subset $S(0, i]$. The only difference is that the former uses the index M while the latter uses the index 0 as the fixed reference point in the search. To save working space in the bottom-up dynamic programming we no longer save the optimal cutting positions of $\check{\mathbf{q}}_{M-i}^{\kappa/2}$ and $\hat{\mathbf{q}}_i^{\kappa/2}$ in Ł arrays. In this way we can only find $\hat{\mathrm{q}}_{\kappa/2}$ by spending $O(\kappa M^2)$ time, with $O((\kappa/2)M^2)$ time for each of $\check{\mathbf{q}}_{M-i}^{\kappa/2}$ and $\hat{\mathbf{q}}_i^{\kappa/2}$ plus $O(M)$ time for the linear search of (16). Then recursively, we can set $\hat{\mathrm{q}}_{\kappa/4} = \arg\min_{\kappa/4 \le i \le \xi-\kappa/4}\{E(\hat{\mathbf{q}}_i^{\kappa/4}) + E(\check{\mathbf{q}}_{\xi-i}^{\kappa/4})\}$ by computing $\hat{\mathbf{q}}_i^{\kappa/4}$ on $S(0, i]$ and $\check{\mathbf{q}}_{\xi-i}^{\kappa/4}$ on $S(i, \xi]$, where $\xi = \hat{\mathrm{q}}_{\kappa/2}$. Thus finding $\hat{\mathrm{q}}_{\kappa/4}$ incurs an $O((\kappa/4), M^2)$ cost but using only linear space. Likewise, with the same costs, we can set $\hat{\mathrm{q}}_{3\kappa/4}$. Now it becomes clear that $\hat{\mathbf{q}}_M^\kappa$ can be done in $2\kappa M^2 \sum_{1 \le j \le \log(\kappa-1)} 2^{-j} = O(\kappa M^2)$ time while using only $O(M)$ space with the described technique. In the worst case the above $O(M)$-space algorithm doubles the execution time of the $O(\kappa M)$-space version.

4.5 Termination of Principal Quantization

As the bottom-up dynamic-programming algorithm proceeds the distributions of color points in the quantization cells $S(\hat{q}_{i-1}, \hat{q}_i]$, $1 \le i \le k$, become less and less biased toward the principal axis of S for increasing k. The algorithm

should terminate with the output $\hat{\mathbf{q}}_M^{\kappa}$ when for some $k = \kappa$ none of the data sets $S(\hat{q}_{i-1}, \hat{q}_i]$, $1 \le i \le \kappa$, has a strongly biased orientation in the principal direction of S. Otherwise, if the algorithm continues for $k > \kappa$, its output $\hat{\mathrm{q}}_M^k$ can differ significantly from the globally optimal k-cell quantizer. The critical parameter κ for the optimal principal quantizer $\hat{\mathbf{q}}_M^{\kappa}$ varies from image to image, but it can be determined during the bottom-up dynamic-programming process by examining the eigenvalues and eigenvectors of $S(\hat{q}_{i-1}, \hat{q}_i]$, $1 \le i \le k$, as k increases, to detect the shift of principal axes. The covariance matrix $\mathbf{C} = E\{\mathbf{cc}^T\} - E\{\mathbf{c}\}[E\{\mathbf{c}\}]^T$ of $S(\hat{q}_{i-1}, \hat{q}_i]$ consists of nine covariances (six of them are distinct due to the symmetry of $\mathbf{C}$), namely,

$$\mathbf{c}_{r,s} = \frac{\sum_{\mathbf{c} \in S(\hat{q}_{i-1}, \hat{q}_i]} c_r c_s}{|S(\hat{q}_{i-1}, \hat{q}_i]|} \frac{\left[\sum_{\mathbf{c} \in S(\hat{q}_{i-1}, \hat{q}_i]} c_r\right]\left[\sum_{\mathbf{c} \in S(\hat{q}_{i-1}, \hat{q}_i]} c_s\right]}{|S(\hat{q}_{i-1}, \hat{q}_i]|^2} \qquad 0 \le r, s \le 2. \tag{17}$$

If the above matrix is to be evaluated straightforwardly for each subset $S(\hat{q}_{i-1}, \hat{q}_i]$, $1 \le i \le k$, at each level of dynamic programming, $O(\kappa N)$ operations are needed just to correctly terminate the dynamic-programming process. But a similar statistical preprocessing to (13) can reduce this cost to $O(N + \kappa^2)$ thanks to the order already established on color points $\mathbf{c} \in S$ by their projections $\mathbf{c}^T\mathbf{v}$. Indeed, let

$$W_{r,s}(n) \equiv \sum_{0 < R(\mathbf{c}) \le n} c_r c_s, \tag{18}$$

then precompute and store $W_{r,s}(n)$ for $1 \le n \le M$ and $0 \le r \le s \le 2$; we can simplify (17) to

$$c_{r,s} = \frac{W_{r,s}(\hat{q}_i) - W_{r,s}(\hat{q}_{i-1})}{W_0(\hat{q}_i) - W_0(\hat{q}_{i-1})} - \frac{[W_1(r, \hat{q}_i) - W_1(r, \hat{q}_{i-1})][W_1(s, \hat{q}_i) - W_1(s, \hat{q}_{i-1})]}{[W_0(\hat{q}_i) - W_0(\hat{q}_{i-1})]^2}, \tag{19}$$

where W_1 and W_0 are the cumulative comments introduced in the previous section. Therefore, a covariance $c_{r,s}$ can be evaluated in $O(1)$ time independent of the size of the subset $S(\hat{q}_{i-1}, \hat{q}_i]$. Clearly, precomputing and saving all $W_{r,s}(n)$ require $O(N)$ time and $O(M)$ space. They will not change the complexity order of the entire algorithm.

5. LOCALLY OPTIMAL BIPARTITIONING

After the optimal principal quantizer $\hat{\mathbf{q}}_M^{\kappa}$ is computed and if $\kappa < K$, we still need to further partition the subsets $S(\hat{q}_{i-1}, \hat{q}_i]$, $1 \le i \le \kappa$, until the required K clusters are formed. Unfortunately, we can no longer carry out a global optimization in the way in which the optimal principal quantizer is constructed, because none of the $S(\hat{q}_{i-1}, \hat{q}_i]$ has now a predominant enough principal axis to have two or more locally optimal cutting halfplanes parallel to each other. So we resort to the local optimization approach and bipartition

the current subsets $S(\hat{q}_{i-1}, \hat{q}_i]$ one at a time and under some optimization criteria as discussed in Section 3.

Many heuristic spatial-division methods [18, 39, 43] can be applied at this stage. A more elaborate, locally optimal bipartitioning method [40] is to sweep a cutting halfplane normal to the principal axis of a data set and split it at the position where the sum of the distortions of the two resulting subsets is minimum. If the data set subject to bipartitioning is $S(\hat{q}_{i-1}, \hat{q}_i]$, a cell of the optimal principal quantizer $\hat{\mathbf{q}}_M^{\kappa}$, then its covariance matrix can be obtained in $O(1)$ time with the technique of (19); thus its principal axis is known through an eigenvector transform. Due to the small dimensionality of our problem, the numerical computation of the principal eigenvector $\mathbf{v}$ of the 3×3-symmetric, positive definite covariance matrix is fast and robust. Next we sort the color points $\mathbf{c} \in S(\hat{q}_{i-1}, \hat{q}_i]$ by their projections onto the principal axis to facilitate the halfplane sweep. Again, as argued in the previous section, a linear-time bucket sort suffices for our purpose.

Let $S_1(\xi)$ and $S_2(\xi)$ be a bipartition of $S(\hat{q}_{i-1}, \hat{q}_i]$ formed by the halfplane normal to and positioned at coordinate ξ of the principal axis. Our goal is to find the optimal cutting position ξ_{opt}, i.e.,

$$\begin{aligned}\xi_{opt} &= \arg\min_{\xi} \{E(S_1(\xi)) + E(S_2(\xi))\} \\ &= \arg\min_{\xi} \left\{ \sum_{\mathbf{c} \in S_1(\xi)} (\mathbf{c} - \mathbf{z}_1)^T (\mathbf{c} - \mathbf{z}_1) + \sum_{\mathbf{c} \in S_2(\xi)} (\mathbf{c} - \mathbf{z}_2)^T (\mathbf{c} - \mathbf{z}_2) \right\}.\end{aligned}$$

It can be verified after some manipulations that

$$\begin{aligned}&\sum_{\mathbf{c} \in S_1(\xi)} (\mathbf{c} - \mathbf{z}_1)^T (\mathbf{c} - \mathbf{z}_1) + \sum_{\mathbf{c} \in S_2(\xi)} (\mathbf{c} - \mathbf{z}_2)^T (\mathbf{c} - \mathbf{z}_2) \\ &\sum_{d=0}^{2} \sum_{\mathbf{c} \in S} c_d^2 - \sum_{d=0}^{2} \left\{ \frac{[\Sigma_{\mathbf{c} \in S_1(\xi)} c_d]^2}{|S_1(\xi)|} + \frac{[\Sigma_{\mathbf{c} \in S_2(\xi)} c_d]^2}{|S_2(\xi)|} \right\} \qquad (20)\end{aligned}$$

Note that the first term is now a constant for any given S; the minimization problem of (20) is equivalent to the maximization one:

$$\begin{aligned}\xi_{opt} &= \arg\max_{\xi} \left\{ \sum_{d=0}^{2} \left[\frac{[\Sigma_{\mathbf{c} \in S_1(\xi)} c_d]^2}{|S_1(\xi)|} + \frac{[\Sigma_{\mathbf{c} \in S_2(\xi)} c_d]^2}{|S_2(\xi)|} \right] \right\} \\ &= \arg\max_{\xi} \left\{ \sum_{d=0}^{2} \left[\frac{[\Sigma_{\mathbf{c} \in S_1(\xi)} c_d]^2}{|S_1(\xi)|} + \frac{[\Sigma_{\mathbf{c} \in S} c_d - \Sigma_{\mathbf{c} \in S_1(\xi)} c_d]^2}{|S_2(\xi)|} \right] \right\}, \qquad (21)\end{aligned}$$

in which $\Sigma_{\mathbf{c} \in S} c_d$ is also a constant for any given S. The primitive operation in the above maximization process is thus $\Sigma_{\mathbf{c} \in S_1(\xi)} c_d$, and it can be done incrementally. Suppose that ξ_1 and ξ_2 are two consecutive ordinates on the major axis after the bucket sort of projections $\mathbf{c}^T\mathbf{v}$, $\mathbf{c} \in S$. Then we have

$$\sum_{\mathbf{c} \in S_1(\xi_2)} c_d = \sum_{\mathbf{c} \in S_1(\xi_1)} c_d + \sum_{\mathbf{c}^T\mathbf{v} = \xi_2} c_d. \qquad (22)$$

Since the data set S is sorted by $\mathbf{c}^T\mathbf{v}$ we can retrieve all those c satisfying $\mathbf{c}^T\mathbf{v} = \xi_2$ by simply marching in the sorted list from ξ_1 to ξ_2 and evaluate $\sum_{\mathbf{c}^T\mathbf{v}=\xi_2} c_d$ along the way. After this, $\sum_{\mathbf{c}\in S_1(\xi_2)} c_d$ can be incrementally obtained in constant time from $\sum_{\mathbf{c}\in S_1(\xi_1)} c_d$. Based on the above analysis we finally conclude that ξ_{opt} defined in (21) can be found in time linear to the size of the data set.

This halfplane placed at ξ_{opt} optimized on the principal axis can be adjusted then by a two-mean, iterative clustering process [25] to achieve a local optimum. This, too, can be done efficiently. Given two tentative centroids $\mathbf{z}_1$ and $\mathbf{z}_2$ of the two subsets S_1 and S_2, we have the three-dimensional halfplane equation $(\mathbf{c} - \mathbf{z}_h)^T(\mathbf{z}_2 - \mathbf{z}_1) = 0$, where $\mathbf{z}_h$ is the midpoint of the line segment connecting $\mathbf{z}_1$ and $\mathbf{z}_2$. Then we can determine the cluster membership of a color point $\mathbf{c} \in S$ in S_j, $j = 1, 2$, by substituting the c into the plane equation and checking the sign of the result. If positive, $\mathbf{c}$ is closer to $\mathbf{z}_2$; otherwise it is closer to $\mathbf{z}_1$. Since the starting halfplane of the two-mean clustering process is already optimized normal to the principal axis, very few iterations suffice for the iterative clustering algorithm to converge to a local optimum.

6. QUANTIZER MAPPING AND DITHERING

Through a dynamic-programming process for optimal principal quantization and the subsequent locally optimal bipartitioning we obtain a good color quantizer; i.e, the final K representative colors $\mathbf{z}_j$, the centroids of S_j, $1 \le j \le K$, are chosen and loaded into the color lookup table for a frame buffer. The next step, called quantizer mapping in the sequel, is to map all input colors of S to some $\mathbf{z}_j$. Clearly, in order to minimize the quantization distortion for the given color quantizer, the quantizer mapping should be a nearest-neighbor match; i.e., a color $\mathbf{c}$ is mapped to $\mathbf{z}_\xi$ such that $\xi = \arg\min_{1\le j\le K}\|\mathbf{c}, \mathbf{z}_j\|$. In other words, the quantizer mapping has the structure of the three-dimensional voronoi diagram on the K centroids $\mathbf{z}_j$. Note that the final K-partition of S created by the new algorithm, S_j, $1 \le j \le K$, generally does not constitute a voronoi partition on K representative colors. However, it was found in our practice that the simple quantization of $\mathbf{c} \in S_j$ to $\mathbf{z}_j$ did not cause a significant increase in numerical quantization error nor a noticeable loss of image quality from the nearest-neighbor mapping. This observation suggests that the cluster membership of $\mathbf{c}$, known from the quantizer design process, can be directly used for a good quantizer mapping, saving the expensive nearest-neighbor search. In contrast, in order for the previous algorithms [18, 39] to get good results, the nearest-neighbor mapping is essential.

For small K, dithering is often necessary to reduce the false contours in the quantized color images. In this case, a $\mathbf{c} \in S_j$ may be quantized to some other centroid $\mathbf{z}_t$, if $\mathbf{c} + \mathbf{e} \in S_t$ where $\mathbf{e} \in \mathscr{R}^3$ is the distributed error vector from the neighboring pixels in the image space. The original cluster membership of $\mathbf{c}$ can no longer serve as a quantizer-mapping function. A three-dimensional range search based on nearest-neighbor rule must be conducted to quantize

the composite color value $\mathbf{c} + \mathbf{e}$. Nearest-neighbor search is a well-studied topic in computational geometry [32]. The classic *k-d* tree [10] method can answer a nearest-neighbor query in $O(\log K)$ expected time. The *k-d* tree can be built in $O(K \log K)$ time. With the support of this data structure the nearest-neighbor quantizer mapping of an entire color image can be effected in $O(N \log K)$ expected time. A faster but suboptimal (not necessarily nearest-neighbor match) alternative is to build the BSP tree explicitly in the quantizer design process and then use it as a binary search tree to determine the relation $\mathbf{c} + \mathbf{e} \in S_j$. The internal nodes of the BSP search tree represent a spatial cut, and its K leaves contain K representative colors. The query point $\mathbf{y} = \mathbf{c} + \mathbf{e}$, given an input pixel value $\mathbf{c}$ and its inherited error vector $\mathbf{e}$ in dithering, is checked against the cutting halfplanes along a path starting from the root of the BSP tree until it reaches a leaf node S_l; then the color $\mathbf{z}_l$ is assigned to the query pixel.

In the computational geometry literature [32] the trees for range search are often height balanced to achieve logarithmic query time in the worst case. But in color quantization, we are concerned with the amortized (not the worst-case) time complexity. It is the total cost of allocating N query points in K quantizer cells that needs to be minimized, not the single-shot query time. Let l_j be the length of the path from the root of the BSP tree to the leaf node containing S_j. The optimal search tree for our problem should minimize $\sum_{1 \le j \le K} |S_j| l_j$ which is the total number of comparisons required by the quantizer mapping of an entire image. To this end we need to balance the BSP search tree in terms of the color population rather than its height. For any internal node we will let its left and right subtrees have approximately the same number of color points. Such a population tree balancing is straightforward with the optimal multilevel principal quantizer $\hat{\mathbf{q}}_M^*$ while being quite tricky with a recursive, greedy, bipartitioning algorithm. The root of the search tree corresponds to the halfplane normal to the principal axis of S and placed at $\hat{q}_t$ such that $t = \arg\min_i \{|\,|S(0, \hat{q}_i]| - N/2|\}$. This equal-population criterion can be applied recursively in shaping the search BSP tree until we move into the subsets of some cell of the optimal principal quantizer $S(\hat{q}_i, \hat{q}_{i+1}]$. Now we can appreciate more the global impact of optimal multilevel principal quantization. Unlike the greedy bipartitioning approach that can only optimize a tree-structured quantizer node by node, the new algorithm optimizes the top of a quantizer tree globally for minimizing both the quantization distortion and the quantizer-mapping time. With a population-balanced tree, the total quantizer-mapping time can be bounded by $O(N \log K)$ in the worst case, which occurs when each quantizer cell has the same color population. Note that the previously claimed $O(N \log K)$ cost for quantizer mapping by the *k-d* tree method is only the expected behavior of the algorithm. Its worst-case time complexity is $O(KN)$, and this could indeed happen for some data sets [35].

7. EXPERIMENTAL RESULTS AND REMARKS

The proposed color quantization algorithm was implemented and tested on twenty 24-bits/pixel color images. The new algorithm obtained significantly

Table I. Quantization Errors of Different Algorithms on Test Images

	Median-Cut		Wan *et. al.*		New	
K	Zelda	Toys	Zelda	Toys	Zelda	Toys
16	1328.9	1076.4	1294.6	1138.3	301.9	211.5
32	1228.1	695.7	581.9	466.6	156.3	103.1
64	713.2	343.2	304.1	253.2	78.7	57.3
256	79.5	68.3	69.6	52.6	24.0	20.4

smaller quantization distortions than the median-cut algorithm [18] and marginal-variance minimization algorithm [39] on every test image. On the average, the quantization error of the new color quantizer is 24.3% of the marginal-variance minimization algorithm and 19.4% of the median-cut-algorithm. The improved quantizer performance is due primarily to a better quantizer adaptability to color statistics that is achieved by the new color-space-partitioning technique of optimal principal multilevel quantization. Another important contributing factor to improved performance is that the new algorithm eliminates the step of prequantization used by previous algorithms [1, 13, 15, 18, 39, 43] just as an expediency to control space complexity. These previous algorithms need color data to be organized in a sparse three-dimensional array whose size is unmanageable without prequantization. By projecting color points onto the principal axis the internal data representation becomes a compact one-dimensional array, rendering prequantization unnecessary. Currently, the common prequantization practice seems to simply chop three least significant bits off as first suggested in [18], losing quantizer precision by three bits even before clustering. This drawback is completely overcome by the new algorithm.

Since the ultimate judgment for any quantization algorithm in practice should be subjective image fidelity, we present in the color plates quantized images by different algorithms for various K. The mean-square quantization errors of two ISO test images, Zelda and Toys (Plate I), are tabulated in Table I. Plates II–VIII show different quantization algorithms applied to the image Zelda. When $K = 256$, the new algorithm reproduced images that are virtually indistinguishable from the originals even without dithering; the median-cut algorithm still produced some false contours; Wan et al.'s algorithm is better than the median-cut algorithm, but it has subtle differences from the new algorithm. For instance, in Plate II(b) Zelda's lip lines are slightly jagged, and the shadow on her neck is not as smooth as in Plate II(c). As the number of colors decreases the three algorithms behaved very differently. Human subjects seemed to prefer the images quantized by the new algorithm to those by the other two.

A characteristic of marginal-variance minimization that was not mentioned in the original paper [39] was observed in our experiments. By always cutting the component with the largest marginal variance this component-based method favors luminance over hue quantization. Consequently, it tends to

(a) Zelda

(b) Toys

Plate I. Two ISO 24bit/pixel color test images.

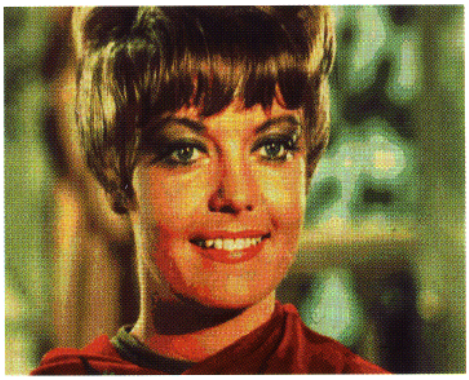
reproduce smoother images than the median-cut algorithm. However, this smoothness was achieved at the expense of distorted hues. To see this, compare the skin tone of Zelda's face and her lips, for $K = 64, 16$, (plates III(b) and IV(b)) with those of the original. Similarly, the Toys images

(a) Median-cut. Quantization is visible along the left side of the face.

(b) Marginal variance minimization.

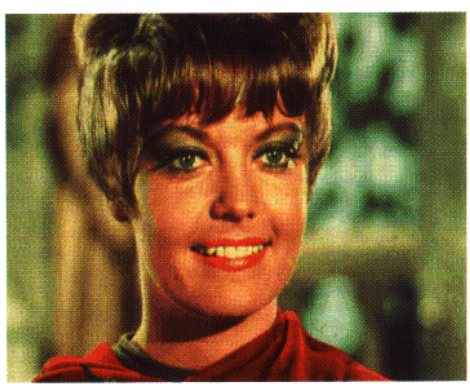
(c) New algorithm.

Plate II. Comparison of median-cut, marginal variance minimization and the new algorithm for K = 256.



(a) Median-cut.

(b) Marginal variance minimization.



(c) New algorithm.

Plate III. Comparison of median-cut, marginal variance minimization and the new algorithm for K = 64.

(a) (b)

Plate IV. Median cut algorithm for K = 16. (a) without dithering (b) with dithering

(a) (b)

Plate V. Marginal variance minimization for K = 16. (a) without dithering (b) with dithering.

(a) (b)

Plate VI. Uniform quantization for K = 16. (a) without dithering (b) with dithering.

(a) (b)

Plate VII. New algorithm for K = 16. (a) without dithering (b) with dithering.

Plate VIII. New algorithm for K = 16, with smoothness emphasis without dithering. Compare to Plate VII (a).

quantized by this algorithm seemed to have a false reddish tone when $K = 64$. For $K = 16$ this algorithm almost reduced the Zelda image to a gray-scale image of dominant hue. We can imagine this smoothness emphasis taken to the extreme. If all K representative colors were chosen as gray scales, then color quantization is reduced to scalar quantization, and the output would be a smooth but monochrome image. The new algorithm aims at a good balance between the smoothness and hue accuracy through optimal principal quantization in Section 4.2, and this goal seems to be realized according to the color plates.

To minimize the total quantization distortion the new algorithm switches from optimal principal quantization to locally optimal bipartitioning when the current data subsets no longer have their principal axes agree with that of the original data set S. But if the smoothness of the quantized image outweighs its hue precision for an individual's taste, we can delay the transition of the algorithm from optimal principal quantization to locally optimal bipartitioning. Perceptually this means that more information capacity is given to luminance than to hue. The user can retain this control by setting the critical parameter κ. To show what happened when a user

instructed the algorithm to give smoothness a higher priority we give in Plate VIII a 16-color image that was quantized with two more cuts in the principal direction than the one in Plate VII(a). If we need to trade hue for luminance when the color quota K is not adequate to satisfy both, principal optimal quantization is a better compromise than marginal-variance minimization. By minimizing three-dimensional color distortion under the constraint of cutting the principal (luminance) axis, the new algorithm does not completely ignore the hue distributions as in [39].

For $K < 64$ even the new algorithm generates objectionable false contours. Floyd-Steinberg's dithering technique [7] was employed to reduce the contour effects. Dithering noticeably enhanced the performance of color quantizers (compare (a)'s and (b)'s in Plates IV–VII). However, a better quantizer designed independently of dithering will not necessarily perform better after dithering. For example, the uniform quantizer performed by far the worst without dithering (Plate VI), while its dithered version is superior to the dithered images of the median-cut algorithm and Wan et al.'s algorithm. The comparison between the dithered images of the uniform quantizer and the new algorithm is less conclusive. Judging by the photographs the reader may find that the higher color contrast in Plate VI(b) makes it more visually appealing than Plate VII(b). The representative colors of a uniform quantizer have a wider span over the color space than a statistical quantizer. Consequently, the dithering can simulate a wider spectrum of colors with the former than with the latter. However, the adjacent pixels in a smooth region may be mapped to drastically different colors by dithering, magnifying the artificial dither textures. Indeed, at a closer look Plate VI(b) is much noisier than Plate VII(b) with more prominent dithering patterns. The problem is far more severe on a color CRT where the image in Plate VI(b) resembles a color television signal corrupted by snow noise. Even in the prints we can observe that the dithered image of the new algorithm retains much finer details (Zelda's eyes, hair, and neck shadow, say) than the dithered image of the uniform quantizer.

For practical considerations, when $K \geq 256$ previous algorithms [15, 18, 39] often give satisfactory results at higher speeds, especially if these orthogonal bipartitioning methods are sped up by the tricks published by Wu [42]. The new algorithm is more advantageous when high-quality color reproduction for small K is required. In a windowing system running multiple applications a common color map is desired. In this case, dithering based on a uniform sampling of the color gamut in a perceptually uniform space should offer a simple and often effective solution if the device can address more than 100 pixels per inch. However, if a window has a strongly biased color distribution the dithering based on an image-independent set of prefixed colors will perform very poorly since many colors in the lookup table cannot be effectively used by dithering, wasting the information capacity of the device. Fidelity gain can be achieved by good image-dependent color quantizer if it can be computed fast enough. Thus, lower time complexity of the quantization algorithm can translate to higher image quality, justifying the research on algorithmic approaches to color quantization.

In addition to the lack of a context-dependent color measure that can be integrated into an optimal color quantizer definition, two more interesting and important problems in color quantization are still open: (1) a definition and approximation algorithm of an optimal color quantizer for a chosen dithering process; (2) an on-line algorithm for color quantization, i.e., dynamically update the contents of the color map according to the change of color distribution in time, for instance, during an animation. Finally, a word of warning for the sake of rigor, the optimal principal quantization defined by (9), under the constraint of cutting orientation, is only a better heuristic method guided by a common statistical characteristic of color images to approximate the solution of the original NP-complete problem of optimal color quantization. A nontrivial bound on the difference between the true global minimum quantization distortion and that of the new algorithm or any other heuristic algorithm still remains elusive.

8. CONCLUSIONS

A novel color-space-partitioning strategy is presented for color quantizer design based on our observation that the colors of an image have a significant statistical bias along the principal axis of the input data. A color space is optimally partitioned into multiple quantizer cells in the principal direction of the input data, rather than bipartitioned recursively and orthogonally as done by current color quantization algorithms. The total quantization error of multiple quantizer cells in the principal direction can be minimized by a new algorithm using dynamic programming. Due to its better adaptability to the color statistics of input images and its ability to minimize quantization error over multiple quantizer cells as a whole, the new color quantizer design algorithm outperforms the existing algorithms for a large set of test images. On the average, the mean-square quantization error of the new algorithm is five times smaller than that of the traditional algorithms. Also, the quantized images produced by the new algorithm seem to look better than those produced by the traditional algorithms. The new algorithm is analyzed and shown to be practical for the mean-square error measure. Indeed, for the 720×576 ISO test images, the new algorithm takes less than three minutes on a Personal IRIS workstation. Furthermore, the algorithm works in all color spaces and can be generalized to other error measures as well.

ACKNOWLEDGMENT

The author thanks all four reviewers and Maureen Stone, the editor of this special issue, for their constructive comments about the context-sensitive nature of human color vision and its impact on color quantizer design. Thanks go also to Dr. Robert Webber and Dr. Robert Mercer for their proofreading of the final manuscript.

REFERENCES

1. BALASUBRAMANIAN, R., AND ALLEBACH, J. A new approach to palette selection for color images. *J. Imaging Tech. 17*, 6 (Dec. 1991), 284–290.
2. BRUCKER, P. On the complexity of clustering problems. In *Optimization and Operations*

Research. R. Henn, B. Korte, and W. Oettli, Eds. Springer-Verlag, New York, 1977, pp. 45–54.

3. CAMPBELL, G., DEFANTI, T., FREDERIKSEN, J., JOYCE, S., LESKE, L., LINDBERG, J., AND SANDIN, D. Two bit/pixel full color encoding. In *Proceedings of SIGGRAPH'86*. ACM, New York, 1986, pp. 215–233.
4. CHOU, P. A., LOOKABAUGH, T., AND GRAY, R. M. Optimal pruning with applications to tree-structured source coding and modeling. *IEEE Trans. Inf. Theory IT-35*, 2, (1989), 299–315.
5. DREZNER, Z. The p-center problem—Heuristic and optimal algorithms. *J. Oper. Res. Soc. 35*, (1984), 741–748.
6. FIUME, E., AND OUELLETTE, M. On distributed, probabilistic algorithms for computer graphics. In *Proceedings of Graphics/Interface '89*. 1989, pp. 211–218.
7. FLOYD, R. W., AND STEINBERG, L. An adaptive algorithm for spatial gray scale. *Int. Symp. Digest Tech. Papers* (1975), 36.
8. FOLEY, J. D., DAM, A. V., FEINER, S. K., AND HUGHES, J. F. *Computer Graphics, Principles and Practice*. Addison-Wesley, Reading, Mass., 1990.
9. FREI, W. Rate-distortion coding simulation for color images. In *Proceedings of SPIE National Seminar on Advances in Image Transmission Technology*, vol. 87. 1976, pp. 197–203.
10. FRIEDMAN, J. H., BENTLEY, J. L., AND FINKEL, R. A. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw. 3*, 3 (Sept. 1977), 209–226.
11. FUCHS, H., KEDEM, G. Z. M., AND NAYLOR, B. F. On visible surface generation by a priori tree structure. *Comput. Graphics 14*, 3 (July 1980), 124–133.
12. GAREY, M. R., JOHNSON, D. S., AND WITSENHAUSEN, H. S. The complexity of the generalized Lloyd-Max problem. *IEEE Trans. Inf. Theory IT-28*, (Mar. 1982), 255–256.
13. GENTILE, R. S., ALLEBACH, J. P., AND WALOWIT, E. Quantization of color images based on uniform color spaces. *J. Imaging Tech. 16*, 1 (Feb. 1990), 11–21.
14. GERSHO, A., AND GRAY, R. M. Vector Quantization and Signal Compression. Kluwer Academic Publishers, Boston, 1991.
15. GERVAUTZ, M., AND PURGATHOFER, W. A simple method for color quantization: Octree quantization. In *Graphics Gems*. Academic Press, New York, 1990.
16. GRAY, R. M. Vector quantization. *IEEE ASSP Mag.* (Apr. 1984), 4–29.
17. HABIBI, A., AND WINTZ, P. A. Image coding by linear transformation and block quantization. *IEEE Trans. Commun. COM-19*, (Feb. 1971), 50–61.
18. HECKBERT, P. Color image quantization for frame buffer display. In *Proceedings of SIGGRAPH'82*. ACM, New York, 1982, pp. 297–307.
19. KURZ, B. Optimal color quantization for color displays. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Washington D.C.). IEEE, New York, 1983, pp. 217–224.
20. LIMB, J. O., RUBINSTEIN, C. B., AND THOMPSON, J. E. Digital coding of color video signals –A review. *IEEE Trans. Comm. COM-25*, 11 (Nov. 1977), 1349–1385.
21. LIN, J., STORER, J., AND COHN, M. On the complexity of optimal tree pruning for source coding. In *Proceedings of Data Compression Conference*. IEEE Computer Society Press, Los Angeles, 1991, pp. 63–72.
22. MACADAM, D. L. Projective transformations of ICI color specifications. *J. Opt. Soc. Amer. 27*, (Aug. 1937), 294–299.
23. MACADAM, D. L. Uniform color scale, *J. Opt. Soc. Amer. 64*, (1974), 1691.
24. MANLY, B. F. J. *Multivariate Statistical Methods*. Chapman and Hall, London, England, 1986.
25. MACQUEEN, J. B. Some methods for classification and analysis of multivariate observations. In *Proceedings 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1. 1967, pp. 281–297.
26. MEGIDDO, N., AND SUPOWIT, K. J. On the complexity of some common geometric location problems. *SIAM J. Comput. 13*, (1984), 182–196.
27. MERRIFIELD, R. M. Visual parameters for color CRTs. In *Color and the Computer*. Academic Press, New York, 1987, pp. 63–81.

28. MEYER, G. W., AND GREENBERG, D. P. Perceptual color space for coputer graphics. In *Color and the Computer*. Academic Press, New York, 1987, pp. 83–100.
29. MORGAN, J. N., AND SONQUIST, J. A. Problems in the analysis of survey data, and a proposal. *J. Amer. Statist. Assoc. 58*, (1963), 415–434.
30. ORCHARD, M., AND BOUMAN, C. Color quantization of images. *IEEE Trans. Signal Processing* (Dec. 1991).
31. PRATT, W. K. Spatial transform coding of color images. *IEEE Trans. Comm. COM-19*, (Dec. 1971), 980–992.
32. PREPARATA, F. P., AND SHAMOS, M. I. *Computational Geometry*. Springer-Verlag, New York, 1985.
33. RUBINSTEIN, C. B., AND LIMB, J. O. Statistical dependence between components of a differentially quantized color signal. *IEEE Trans. Comm. COM-20*, (Oct. 1972), 890–899.
34. SOONG, F. K., AND JUANG, B. H. Optimal Quantization of LSP Parameters. In *Proceedings of ICASP '88* (New York, Apr. 4–11). 1988, pp. 394–397.
35. SPROULL, R. E. Refinements to nearest-neighbor searching in k-d trees. *Algorithmica 6*, (1991), 579–589.
36. STENGER, L. Quantization of TV chrominance signals considering the visibility of small color differences. *IEEE Trans. Comm. COM-25*, (Nov. 1977), 1393–1406.
37. STEVENS, R. J., LEHAR, A. F., AND PRESTON, R. H. Manipulation and presentation of multidimensional image data using the Peano scan. *IEEE Trans. PAMI 5*, 2 (Sept. 1983), 520.
38. STONE, M. C., COWAN, W. B., AND BEATTY, J. C. Color gamut mapping and the printing of digital images. *ACM Trans. Graphics 7*, 3 (Oct. 1988), 249–292.
39. WAN, S., WONG, S., AND PRUSINKIEWICZ, P. An algorithm for multidimensional data clustering. *ACM Trans. Math. Softw. 14*, 2 (June 1988), 153–162.
40. WU, X. Statistical colour quantization for minimum distortion. In *Tutorials and Perspectives in Computer Graphics*. Springer-Verlag, New York, 1992.
41. WU, X. Optimal quantization by matrix-searching. *J. Alg. 12*, 4 (Dec. 1991), 663–673.
42. WU, X. Efficient statistical computations for optimal color quantization. In *Graphics Gems*, vol. II. Academic Press, New York, 1991, pp. 126–133.
43. WU, X., AND WITTEN, I. A fast k-means type clustering algorithm. Res. Rep. No. 85/197/10, Dept. of Computer Science, Univ. of Calgary, 1985.
44. WU, X., AND ZHANG, K. A better tree-structured vector quantizer. In *Proceedings of the IEEE Data Compression Conference*. IEEE Computer Society Press, Los Angeles, 1991, pp. 392–401.

Received September 1991; revised March 1992 and June 1992; accepted July 1992