Color Quantization and Processing by Fibonacci Lattices

Aleksandra Mojsilović, *Member, IEEE,* and Emina Soljanin, *Member, IEEE*

***Abstract*—Color quantization is sampling of three-dimensional (3-D) color spaces (such as *RGB* or *Lab*) which results in a discrete subset of colors known as a color codebook or palette. It is extensively used for display, transfer, and storage of natural images in Internet-based applications, computer graphics, and animation. We propose a sampling scheme which provides a uniform quantization of the *Lab* space. The idea is based on several results from number theory and phyllotaxy. The sampling algorithm is very much systematic and allows easy design of universal (image-independent) color codebooks for a given set of parameters. The codebook structure allows fast quantization and ordered dither of color images. The display quality of images quantized by the proposed color codebooks is comparable with that of image-dependent quantizers. Most importantly, the quantized images are more amenable to the type of processing used for grayscale ones. Methods for processing grayscale images cannot be simply extended to color images because they rely on the fact that each gray-level is described by a single number and the fact that a relation of full order can be easily established on the set of those numbers. Color spaces (such as *RGB* or *Lab*) are, on the other hand, 3-D. The proposed color quantization, i.e., color space sampling and numbering of sampled points, makes methods for processing grayscale images extendible to color images. We illustrate possible processing of color images by first introducing the basic average and difference operations and then implementing edge detection and compression of color quantized images.**

***Index Terms*—Color codebooks, phyllotaxy, processing of color images, quantization of color spaces.**

I. Introduction

COLOR quantization is extensively used for display, transfer, and storage of natural images in Internet-based applications, computer graphics and animation. Each primary color in a digitalized color image is usually quantized with eight-bit resolution. Therefore, a full-color display system requires 24 bits to specify the color of each pixel on the screen. Because of the cost of high-speed monitors, this is impractical for many applications. Thus, currently available monitors display colors with a limited number of bits (8, 12, or 16), where each value is used as an index into a table of colors specified by the user. Therefore, the subset of available colors, called *color codebook*, has to be preselected from the full color set. Color codebooks are also used in many other image processing operations, such as enhancement, edge detection, pseudocoloring, and color adjustment. These operations are more efficiently carried out on a small set of colors.

Design of color codebooks has been studied at length and many color quantization methods have been developed. The algorithms for color quantization can be image-dependent or image-independent. Image-dependent quantization algorithms traditionally adopt a vector quantization (VQ) approach in a three-dimensional (3-D) color space [1]. In general, these methods primarily aim at removing statistical redundancy between colors in the color space. However, standard VQ-based methods do not take into account perceptual qualities and generate codebooks based on the color distribution, without considering their spatial relationships. Many experiments have demonstrated that the best color codebooks are not necessarily those that minimize the mean squared error criterion [1]. Therefore, more advanced algorithms account for spatial activities among the image pixels [1], or exploit redundancies in both the image and the color space [16]. It is well known that image-dependent quantizers provide better image reproduction than image-independent ones; however they are computationally more demanding.

All these algorithms were developed mainly for use in display systems and do not address the problem of further processing of color quantized images. We investigate the possibility to develop an image-independent quantization scheme, which approaches the display quality of the VQ-based quantizers for a reasonably large number of colors and, most importantly, makes methods for processing grayscale images extendible to color images In general, methods for processing grayscale images cannot be simply extended to color images because they rely on the fact that each gray level is described by a single number as well as the fact that a relation of full order can be easily established on the set of those numbers. Color spaces (such as *RGB* or *Lab*) are, on the other hand, 3-D. We are therefore interested in designing image-independent color codebooks which can be easily used for both display and processing of color images. To achieve this goal we were guided by the following:

1) algorithm for constructing color codebooks should be computationally inexpensive;
2) structure of the codebook should enable fast quantization of color images and ordered dither;
3) addressing scheme of the codebook, i.e., the numbering of color space sample points should make methods for processing grayscale images extendible to color quantized images.

We found a solution in sampling the *Lab* space with respect to the spiral geometry and spiral lattices.

Manuscript received February 28, 2000; revised August 13, 2001. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Robert L. Stevenson.

A. Mojsilović is with the IBM T. J. Watson Research Center, Hawthorne, NY 10532 USA (e-mail: alkesand@us.ibm.com).

E. Soljanin is with the Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974 USA (e-mail: emina@lucent.com).

Publisher Item Identifier S 1057-7149(01)09735-4.

1057–7149/01$10.00 © 2001 IEEE

Cylindrical and spiral lattices can be used to construct color palettes in which each color is described by a single number. This feature makes color quantized images more amenable to the type of processing used for grayscale ones. Another technique which was used for construction of such palettes is space-filling curves [22], [23]. An important feature of Peano curves is that they are special grid-scans which visit every cell in the quadrant they enter before exiting the quadrant. Stevens *et al.* made use of this feature for their limited color display palette design [23]. Peano curves have also been successfully applied to image compression, classification and multispectral display, [19]–[23]. However, space-filling curves do not have an important property of the Fibonacci spiral lattices. In Fibonacci lattices, differences of indexes of two points are uniquely determined by the distance between them in the space. All neighboring points are some Fibonacci number apart. We will use this property to introduce several basic image processing operations in the space of color symbols.

Although our image-independent quantization scheme still requires larger number of colors for an accurate representation then the image-dependent one it offers numerous advantages over the other approaches. For example, traditional color quantization methods impose additional complexity on algorithms for processing of color quantized images (since the resulting image has to be again quantized to the colors from the codebook). On the other hand, the algorithm we propose integrates easily in many image processing operations and thus requires no additional quantization at the output. Furthermore, in the traditional codebook design, colors within the codebook appear randomly, color symbols do not correlate with the actual color values and there is no relationship between the symbols of adjacent colors. Here, we propose a method that designs structured codebooks and simplifies many color processing operations.

This paper is organized as follows. In Section II, we give basic definitions for color spaces and images. In Section III, we introduce quantization of color spaces by Fibonacci lattices. We first describe sampling in two dimensions by cylindrical and spiral lattices and present a palette design procedure. We then discuss the structure of such codebooks. In Section IV, we discuss quantization of color images by Fibonacci palettes. In particular, we discuss fast quantization and ordered dither. In Section V, we explain how the special structure of the proposed palette can be used for basic operations on color images, such as the difference between two colors, the average value of two colors, an edge detection, while in Section VI, we are concerned with compression of color quantized images. A color version of the manuscript is available at http://www.research.ibm.com/people/a/aleksand.

II. Color Spaces and Images

An $I \times J$ *image* (*picture*) P is a real-valued vector function defined on a rectangular grid of points (pixels) (i, j), where i, $1 \leq i \leq I$, is the row index and j, $1 \leq j \leq J$, is the column index. The color of pixel (i, j), denoted by $p(i, j)$, takes values in the (possibly quantized) 3-D space of colors. We write the color value as a column vector $c = [r_p\ g_p\ b_p]^\dagger$, $c \in C$, $0 \leq r_p, g_p, b_p \leq 1$, where C is a 3-D color space, the vector elements are the red, green, and blue color components and $\dagger$ denotes the transpose operator. Color quantization is sampling of the color space C which results in a discrete subset of N colors known as the *color codebook* or *palette* $C_Q \subset C$. A *Color-quantized* or *palettized* image is a pair (C_Q, P_Q), where P_Q is an $I \times J$ color image with $p_q(i, j) \in C_Q$. In the quantization scheme we propose, each color in the codebook C_Q is identified by a single index q, $0 \leq q < |C_Q| = N_Q$, i.e., $C_Q = \{c_0, c_1, \ldots, c_{N_Q - 1}\}$. The image description where each color value is an integer from the set $C_S = \{0, 1, \ldots, N_Q - 1\}$ will be denoted by P_S and referred to as the *color-mapped* image. Note that $p_s(i, j) = k$ whenever $p_q(i, j) = c_k$.

For *original* (nonquantized) digital images, the value of each color component is represented by an eight-bit number. Thus each pixel uses up to 24 bits of memory. In many applications the cost of high-speed video memory needed for storage of full color images is formidable. In such applications, color display devices reduce the memory requirements by restricting the number of colors that can be displayed simultaneously. Hence, instead of 24 bits, only 8, 12, or 16 bits of video memory is allocated for each pixel, thus allowing simultaneous display of 2^8, 2^{12}, or 2^{16} colors, respectively. The user has the capability to choose a palette of simultaneously displayable colors from the set of all colors. A palletized image, having only the colors from the palette, is stored in the memory and rapidly displayed using the lookup tables implemented in hardware.

For a typical monitor, the input values of the r, g, or b values and their displayed intensities are connected through the nonlinear relationship, approximated as

$$I_r = r^\gamma$$

where I_r is intensity, r is display input, and $\gamma \approx 2.3$. Therefore, in all image capturing devices, to compensate for this nonlinearity between the input primary and its luminance, it is necessary to predistort their output according to the inverse relationship

$$r = (I_r)^{1/\gamma}.$$

The predistorted primary color values are usually referred to as *gamma-corrected colors*. The gamma-corrected colors have the advantage that they can be displayed directly. Moreover, they are less susceptible to noise than the intensities of the corresponding linear primaries [1]. Therefore, in many methods for the color palette design the original image is specified in terms of the γ-corrected coordinates.

It has been shown that quality of the displayed image as well as the final results of numerous image processing applications (color quantization [2], halftoning [5], coding [9], matching, and retrieval [6]) can be substantially improved by using luminance-chrominance color spaces where the image components are reasonably uncorrelated with each other. Two special cases are *Lab* and *Luv* color systems, which are nonlinearly related to the primary intensities. These color spaces are colorimetric standards designed to mimic experimental data of isotropic color spacing. Their use in image quantization and further processing

improves the performance of the method. Therefore, it will be assumed throughout the paper that the original image is specified in the *Lab* color space. This representation is obtained from the γ-corrected *RGB* representation by first linearizing the input data (removing the γ-correction). Next, the data is transformed into the XYZ color space using a linear operator. In the XYZ space, the data is normalized with respect to the illumination white point and then converted to the *Lab* representation via a nonlinear transform (more details on this procedure as well as on design and properties of the *Lab* can be found in [7]). Consequently, the set of all possible colors C occupies an irregular cone in the *Lab* space. From now on we will refer to this cone as C_{Lab}, while under the color value of the pixel p we assume $c = [L_p\ a_p\ b_p]^\dagger$, $c \in C_{Lab}$, where the vector components are luminance L_p and chrominance (a_p, b_p) values of the pixel.

III. Quantization of Color Spaces by Fibonacci Lattices

We propose a sampling scheme which provides a uniform quantization of the *Lab* (or *Luv*) space and a way to establish a partial order relation on the set of points. The idea is based on several results from number theory and the study of geometric and numerical patterns in plants, *phyllotaxy*. An important subject of phyllotaxy is the family of *spiral lattices*. In this section we briefly introduce spiral lattices, give basic notation and definitions and explain the algorithm that samples the *Lab* space according to the spiral geometry. A more detailed overview of phyllotaxy is given in Appendix I.

A. Cylindrical and Spiral Lattices in Two Dimensions

Sampling in the k-dimensional Euclidean space is commonly represented by a *lattice*, see for example [3]. A k-dimensional lattice is specified by a set $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k$ of linearly independent real vectors; the set of all points

$$\boldsymbol{x} = u_1\boldsymbol{a}_1 + \cdots + u_k\boldsymbol{a}_k$$

with integer $u_1, \ldots, u_k$ comprise the lattice. The sampled subset of the Euclidean space consists of all lattice points which belong to the subset. A point on the lattice is specified by a k-tuple $(u_1, \ldots, u_k)$. As previously mentioned, the idea here is to devise a sampling scheme in which each point is described by a single number.

In two dimensions, we can equivalently consider the complex Euclidean plane $w = u + jv$. The points

$$\begin{aligned} w_{n,m} &= u_{n,m} + jv_n \\ &= (-\alpha + j\beta)n + m, \quad \alpha, \beta \in \boldsymbol{R}, \quad n, m \in \boldsymbol{Z} \end{aligned}$$

form a lattice known as the *extended cylindrical lattice* [4]. Again, two integers, m and n, are necessary to specify a lattice point. Since we are concerned with sampling finite regions of the complex plane, we can (without loss of generality) consider only points $w_{n,m}$ for which $0 < u_{n,m} \leq 1$, as shown in Fig. 1 for $\alpha = \sqrt{2} - 1$ and $\beta = 0.03$. It is easy to see that the requirement $0 < u_{n,m} \leq 1$ translates into $m = [n\alpha] + 1$. Thus, each point can be identified only by n. Parameters α and β define the lattice and thus the distribution of points in the plane. The

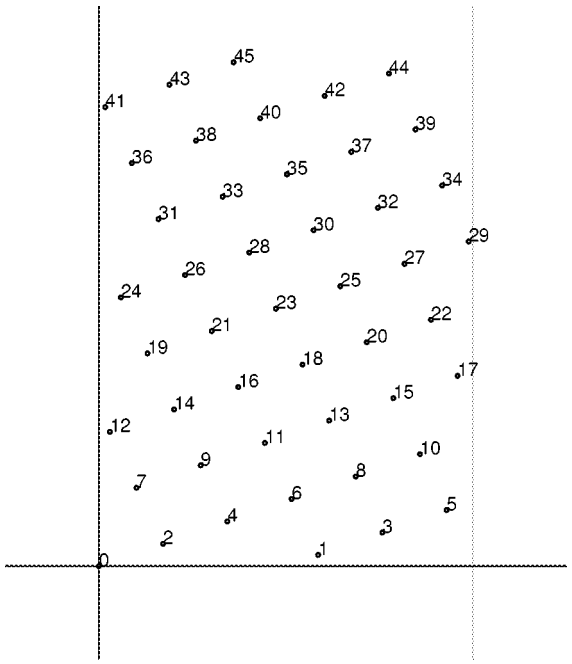


Fig. 1. Points of the cylindrical lattice $w_{n,m} = u_{n,m} + jv_n = (-\alpha + j\beta)n + m$.

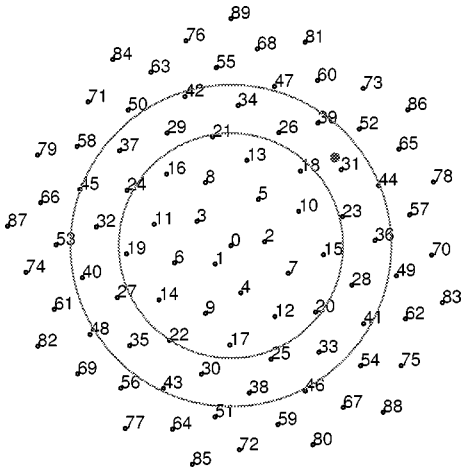


Fig. 2. Points of the Fibonacci lattice in the complex plane.

"more irrational" the value of α the *more uniform* the distribution of points is. Precisely, the point distribution is determined by the continued fractions of α and the most uniform distribution is achieved when α is a so called *Markoff irrational number*.

Because of the central symmetry of several popular color spaces, we consider the complex plane in polar coordinates and define a *spiral lattice* as a convenient means for sampling. A set of points

$$z_n = n^\delta e^{j2\pi \cdot n\tau}, \quad \tau, \delta \in \boldsymbol{R}, \quad n \in \boldsymbol{Z}$$

constitute a spiral lattice. Fig. 2 shows a spiral lattice for $\tau = (\sqrt{5}-1)/2$ and $\delta = 1/2$, which we shall call *Fibonacci lattice*. Again, each point z_n is identified by its index n.

Parameters τ and δ define the lattice and thus the distribution of its points in the plane. For a given δ, parameter τ determines the axial distribution of the points. Examples of spiral lattices for

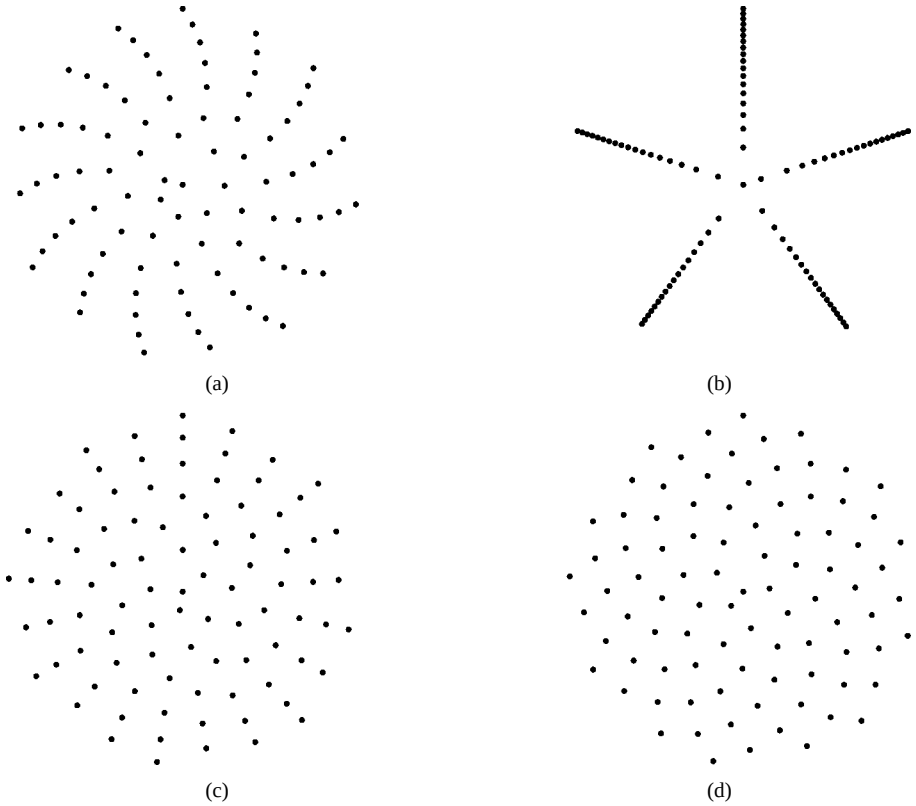


Fig. 3. Spiral lattices for four different values of τ and $\delta = 0.5$.

four different values of τ and $\delta = 0.5$ are shown in Fig. 3. Note that for rational values of τ, e.g., $\tau = p/q$, the lattice points fall on rays radiating from the center at angles $l \cdot 2\pi p/q, 0 \leq l < q$. The axial point distribution is connected to the continued fractions of τ and the most uniform distribution is achieved when τ is a Markoff irrational number. This is discussed in Appendix II. For a given τ, the parameter δ determines the radial distribution of the points. Examples of spiral lattices for four different values of δ and $\tau = (\sqrt{5}-1)/2$ are shown in Fig. 4.

When a spiral lattice is used for sampling color spaces, it may be advantageous to rotate it by a small angle α_0 for better alignment with boundaries of the constant luminance cross section being sampled (this will be explained in the following two subsections). In this case, the lattice points are given by

$$z_n = n^{\delta} e^{j\theta}, \quad \theta = 2\pi n\tau + \alpha_0, \quad \alpha_0, \tau, \delta \in \boldsymbol{R}, \quad n \in \boldsymbol{Z}.$$

B. Color Palette Design

The objective of the palette design algorithm is to sample C_{Lab} into N_Q points, where N_Q is the predetermined palette size. This is done by sampling the *Lab* space with respect to the spiral geometry. In the proposed scheme, we first sample the luminance axis into N_L levels. Then, for each discrete luminance value L_i, $i = 1, \ldots, N_L$, we choose N_p colors in the corresponding (a, b) plane as points z_n of a hexagonal spiral lattice in the complex plane. The points are given by $z_n = s\sqrt{n}e^{j(2\pi n\tau + \alpha_0)}$, where $\tau = (\sqrt{5}-1)/2$ is the golden mean and s is a scaling factor determined as to ensure that the N_p-point lattice covers the region of interest in the (a, b) color plane. The points chosen in this way are uniformly distributed over the area of a disk with a center at the origin (regardless of the disk's size). Hence, there is neither crowding in the center nor sparsity near the edges, providing an optimal packing of the space. In addition to the uniform distribution of points, another advantage of this sampling method is that, for a given n, one can easily determine which points z_i are in the neighborhood of z_n. As a result of this sampling scheme we obtain the set of $N_L \cdot N_p$ points. For each point in the palette, the corresponding symbol is composed by adding its chrominance index n to a multiple of its luminance index i

$$q = n + N_p \cdot i. \tag{1}$$

Consequently, the L, a and b values for any color from the palette can be reconstructed from its symbol q. However, since a cross section of the *Lab* is not circular, there will be points in the Fibonacci lattice which are not valid colors in the *RGB* space. Therefore in the next step, we discard all the points whose r, g, or b value is not within [0,1] range. The remaining N_Q points define the palette. Here, the Fibonacci lattice structure is again useful. Eliminating "invalid" points does not change the structure of the lattice and the labels of the remaining points stay the same. However, since we discard invalid colors, an iterative procedure may be necessary to obtain the required number of colors. An example palette of $N_Q = 139$ colors is shown in Fig. 5. Fig. 6 illustrates a natural image and its color quantized version obtained by using a palette with 1024 colors. Color mapping is performed without additional halftoning, dither, or error diffusion techniques. As illustrated in Fig. 6, the proposed method provides more accurate quantization than commonly

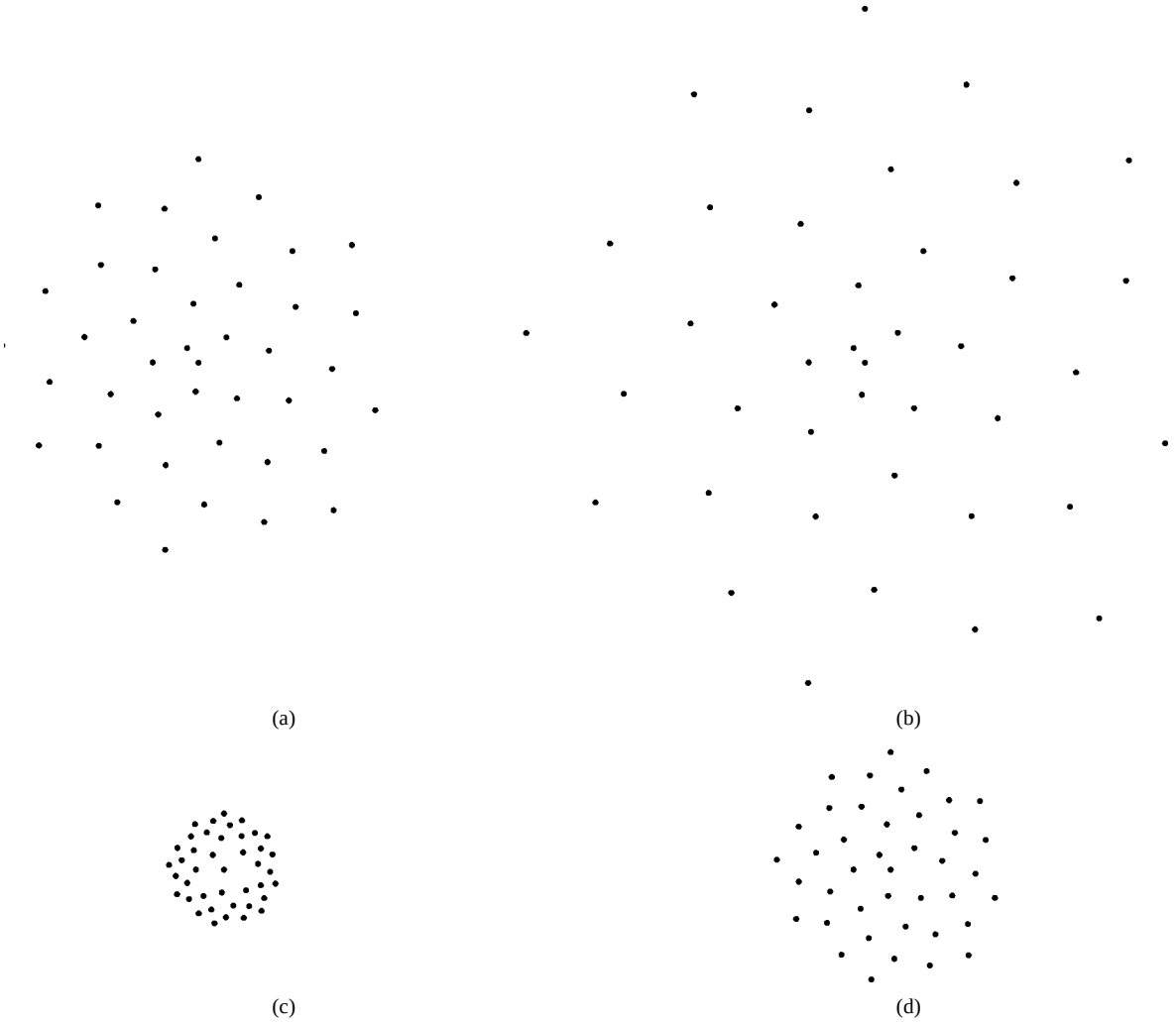


Fig. 4. Spiral lattices for four different values of δ and $\tau = (\sqrt{5}-1)/2$.

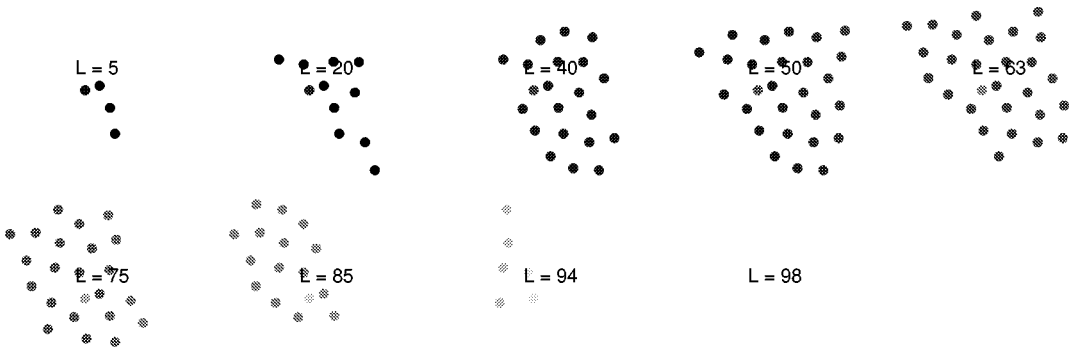


Fig. 5. Palette with $N = 139$ colors.

used techniques based on the uniform quantization of the *RGB* space. It is well-known that (regardless of the method) the use of the fixed palette for image display usually produces unexpectable results. However, images quantized with our method approach the quality of the image dependent techniques for a reasonably large number of colors.

C. Selection of Design Parameters

A very important property of Fibonacci palettes is that they are completely specified by a set of parameters that define and control the sampling process. The parameter list and their descriptions are as follows:

N_Q	total number of colors in the palette;
τ and δ	lattice parameters, defining the distribution of its points in the plane;
N_L	number of luminance levels, i.e., the number of luminance planes to be sampled;
$\{L_i\}_{i=1,\ldots,N_L}$	set of luminance values;
N_p	number of points in each plane; defines the number of chrominance values in the codebook;



(a) (b)

Fig. 6. Color quantized image: (a) original image and (b) color quantized version obtained with a 1024 color palette.

α_0 initial angle; defines the starting point for the sampling in the plane.

Consequently, regardless of the palette size, we do not need to store the real color values for each palette entry. Instead, we need to store only this small set of parameters and use them in real time to generate the required color values. For example, for any 12-bit color palette (4096 colors) we have to store *R*, *G*, and *B* values of each color; this results in 12 kilobytes of data for the pallete; if this is a Fibonacci palette we need less than 50 bytes. This property could be of great importance for storage, transmission and image processing applications.

Depending on the choice of the parameters, we generate codebooks with different properties. The effects of τ and δ are explained in Section III-A. The choice of N_L, N_p and $\{L_i\}$ is crucial for the appearance of colors in the Fibonacci palette. Design of larger palettes, for color display and reproduction, requires larger values of N_L, N_p as well as careful selection of the luminance levels. For example, a palette with 1024 colors is generated with the following parameters:

$$\begin{aligned}\tau =&\frac{(\sqrt{5}-1)}{2},\ \delta = 0.5,\ \alpha_0 = 0.1\\ N_L =&27,\ N_p = 124 \text{ and}\\ \{L_i\} =&\{0,4,10,15,20,25,31,35,40,44,49,52,56,60,\\ &64,68,72,76,80,85,88,92,95,97,98,99,100\}.\end{aligned}$$

The goal of the palette design in this case is to achieve as accurate color representation as possible. Hence, the set of luminance values is selected through the vector quantization of the luminance values, with a large set of training images. The choice of initial angle α_o is not crucial in this case because of the density of sampling. However, this parameter should not be neglected in the design of small codebooks, since some chromaticities could be missed because of the sparse sampling in the plane. For example, if we want to design a color palette for image retrieval, segmentation, or analysis, we need a small numbers of colors to facilitate compact representation and fast searches. In this case, a possible choice of parameters is the following:

$$\begin{aligned}\tau =&\frac{(\sqrt{5}-1)}{2},\ \delta = 0.5,\ \alpha_0 = 0.05,\ N_L = 7,\\ N_p =&56,\ \text{and } \{L_i\} = \{0,10,40,65,85,94,100\}.\end{aligned}$$

The quality of this palette is nicely illustrated in Figs. 7 and 8, which compare the proposed quantization scheme, to commonly used image-independent quantization method based on the uniform sampling of the *RGB* color space.

Although the number of colors in the palette is very small, the representation is still accurate and all the image characteristics are preserved for the future analysis. This is further utilized in [17].

D. Properties of the Fibonacci Lattice

The structure of the proposed color palette makes methods for processing grayscale images extendible to color images. Color image processing methods such as differencing, averaging, edge detection and coding are described and implemented in Sections V and VI. Here, we highlight several important features of the color codebook structure and then briefly explain how it can be used for processing of color images.

Each point z_n in the spiral lattice is identified by a single number, its index n. It is interesting to note that spiral lattices are an important subject of phyllotaxy, the study of geometric and numerical patterns in plants [4]. This is discussed briefly in Appendix I. Before proceeding, it may be useful to look at the images of a daisy and a pinecone shown in Fig. 13 in Appendix I, where certain spiral patterns may be more apparent than in the lattice in Fig. 2. The spiral lattice in Fig. 2 illustrates how the seed-head of a daisy or sunflower might look if magnified, with numbers describing the *age* of the seed. Observe the set of points shown in Fig. 2, forgetting for a moment that their true source is a spiral winding around the center 0. The points appear to form two families of spirals sinking toward the center: one clockwise, one counter clockwise. Notice that the numbers of neighboring points along clockwise spirals differ by $s_r = 21$ and along counter clockwise by $s_l = 13$. Now, choose a point and look at its neighbors in all directions. Note that the differences between the number of the chosen point and its neighbors are Fibonacci numbers. (These numbers are defined and discussed in Appendix I.) Pick another point and note that the differences between its number and its neighbors' numbers are again Fibonacci numbers, as for the first point. This feature of the Fibonacci lattice allows simple and elegant solutions to several problems in color image processing. For example, based on the difference between the color numbers of two neighboring pixels in an image, we can say if there is an edge or not, depending on whether the difference is a Fibonacci number or not.

An important lattice parameter is its *minimum distance* $d_{\min}$. For the Fibonacci lattice, $d_{\min}$ is equal to the Euclidean distance between the lattice point with the largest number and its closest neighbor along the dominant spiral with the larger Fibonacci number. In the example of Fig. 2, the lattice point with the largest number is 89 at the top of the figure, the difference between numbers along the clockwise dominant spiral is $s_r = 21$ and, thus, the closest neighbor to 89 along that spiral is 68. In the same manner, the difference between numbers along the counter clockwise dominant spiral is $s_l = 13$ and thus the closest neighbor to 89 along that spiral is 76. Another distance parameter is the codebook minimum distance $D_{\min}$, defined as



(a) (b) (c)

Fig. 7. Color quantization: (a) original image, (b) color quantized version using a Fibonacci palette with 92 colors, and (c) color quantized version using a palette with 92 colors obtained by uniform sampling of the *RGB* space.



(a) (b) (c)

Fig. 8. Color quantization: (a) the original image, (b) color quantized version using a Fibonacci palette with 92 colors, and (c) color quantized version using a palette with 92 colors obtained by uniform sampling of the *RGB* space.

the minimum Euclidean distance between the sample points of the *Lab* space comprising the color codebook

$$D_{\min} = \min\left(d_{\min}, \min_{i=2,\ldots,N_L} \Delta L_i\right),\ \Delta L_i = L_i - L_{i-1}.$$

To exploit properties of the Fibonacci palettes in color processing applications, it is important to define a *Fibonacci neighborhood* of a color from the palette. For a point q, its Fibonacci neighborhood, $FN(q)$, includes colors closest to q along the dominant spirals in the same luminance plane and the points immediately below and above these points and point q as shown in the equation at the bottom of the page.

IV. Quantization of Color Images By Fibonacci Palettes

A. Fast Quantization Algorithm

The geometry and the numbering of the sample points in the color space significantly simplifies otherwise computationally intensive operation of color quantization. Quantizers which minimize the *mean square error* (MSE) for an arbitrary point in the *Lab* space minimize the Euclidean distance between the point and the palette, i.e., search for the closest point in the palette. It is easy to see that computational requirements to perform this search within a codebook with little structure may be formidable.

We now describe how the structure of the proposed palette can be exploited to narrow the search required for minimum MSE coding. For a pixel p, with color components L_p, a_p, and b_p, the process of determining the closest codebook point starts with finding the closest luminance level L_s from the N_L levels available in the palette. The luminance level L_s determines an (a, b) plane and one of the points z_n, $0 \leq n < N_p$, in that plane is the minimum MSE solution. Recall that the solution is one of the points whose squared distance to the origin is close to $r_p^2 = a_p^2 + b_p^2$. It therefore lies in the ring around the origin whose middle radius is approximately equal to r_p and whose width is slightly larger than the minimum distance of the lattice $d_{\min}$, as shown in Fig. 2. Because of the spiral geometry and the numbering of the lattice, the ring covers points z_n with successive

$$\begin{aligned} FN(q) = \{&q - s_l, q + s_l, q - s_r, q + s_r, && \text{neighbors in the same plane} \\ &q - N_p, q - s_l - N_p, q + s_l - N_p, q - s_r - N_p, q + s_r - N_p, && \text{below} \\ &q + N_p, q - s_l + N_p, q + s_l + N_p, q - s_r + N_p, q + s_r + N_p\} && \text{above} \end{aligned}$$

indexes n belonging to the interval $n \in [n_{\min}, n_{\max}]$, where $n_{\min} = \lfloor (r_p - \alpha d_{\min})^2 \rfloor$ and $n_{\max} = \lceil (r_p + \beta d_{\min})^2 \rceil$. Parameters α and β determine the width of the search ring. When $\alpha = \beta = 0.5$, the width of the search ring is equal to $d_{\min}$. Now, since chromatic components a_s and b_s of the minimum MSE solution are close to a_p and b_p, respectively, we can easily tell when a point in the ring is not a candidate for the MSE solution and do not need to compute the Euclidean distances of p to all the points in the ring.

Hence, for the structured palette with N_L luminance levels the minimum MSE coding is performed by first finding the closest luminance level from the palette. For each image pixel this requires in average $\log N_L$ multiplications, $\log N_L$ subtractions, and $\log N_L$ comparisons. Having found the closest luminance level, the chromatic components that minimize MSE are found by searching only that particular luminance level. For a codebook with N_p colors, the number of points to be examined includes $n_r = (3 \cdot s_l/4)$ pixels, where s_l is the Fibonacci number of the counter clockwise dominant spirals for the particular spiral lattice. Hence, for the Fibonacci palette, minimum MSE quantization of one pixel requires only distance calculation and comparison within a subset of n_r pixels (instead of the total set of N_Q colors in case of unstructured color palettes). This number of possible candidates for the closest color in the palette is even smaller when we discard colors whose r, g, or b value is not within [0,1] range. This is true because eliminating some lattice points does not change the labels of the remaining points.

To illustrate the gain in computational efficiency let us consider a codebook with 1024 colors, designed with $N_L = 28$ and $N_p = 124$. The smaller dominant spiral in this case is 21, which results in $n_r = 16$ candidates. Hence, quantization of one pixel requires calculation of 2-D Euclidean distance and comparison within these candidates only, plus $\log 28$ multiplications, $\log 28$ subtractions, and $\log 28$ comparisons (on average). On the other hand, for unstructured color palettes, quantization of one pixel requires distance calculation in 3-D Euclidean space and comparison with each of 1024 colors from the palette.

B. Ordered Dither

Ordered dither is a method for improving the appearance of a displayed image by adding a pseudorandom noise pattern before the quantization. Hence, instead of quantizing the original image $P_S(i,j)$, we quantize $P'_S(i,j)$ given by

$$P'_S(i,j) = P_S(i,j) + n(i \bmod L, j \bmod L)$$

where $n(k,l)$ is an $L \times L$ matrix of noise (dithering) values. The matrix is chosen to have a zero mean and the minimum of the energy spectrum at the lowest spatial frequencies. The amplitude of the noise has to be determined so that areas of constant color are quantized into a variety of neighboring colors, thus reducing the appearance of false contouring.

Unfortunately, the application of ordered dithering to improve the display of color images is complicated by the nonuniform spread of colors in the palette. This problem is typically found in image-dependent color palettes, since the distance between the neighboring colors may vary significantly across the color space. Therefore, it is sometimes impossible to determine an amplitude needed to dither sufficiently all areas of constant color without adding noticeable noise to other areas. A solution to adjust the dithering to the nonuniform spread of colors in the palette was proposed in [5]. The method, however, has a number of disadvantages and a substantial computational overhead.

Because of the structure of our palette, ordered dithering is easily integrated into the quantization procedure. In that case the amplitude of the pseudorandom noise is determined based on the minimum distance of the relevant Fibonacci lattice or the minimum distance of the entire palette (these distance parameters were defined in Section III-D). Also there is no computational cost of searching for candidate quantization colors, since their symbols are straightforwardly determined by adding or subtracting the appropriate Fibonacci number.

V. Basic Processing of Color Quantized Images

When processing grayscale images, many operators can be defined in a *natural* fashion. For example, a grayscale edge detection operator can be simply introduced as a difference (or gradient) between pixel values in a given neighborhood. However, in case of a color-mapped image, although we clearly see and edge between red and green pixels or an edge between light and dark pixels, such information can not be expressed as a difference between their color symbols. Instead, we have to compute distances along each dimension of the space and combine them in an appropriate manner, or we have to define a metric for the 3-D space. In other words, methods for processing grayscale images cannot be simply extended to color because they rely on the fact that each grayscale is described by a single number and the fact that a relation of full order can be easily established on the set of those numbers. Also, the straightforward application of one-dimensional (1-D) gray-level operators to separately process each color dimension usually does not yield satisfactory results. Therefore, many researchers have addressed the problems of building the optimal representation and optimal approaches for processing of color and color quantized images [9]–[12]. In this and the following section, we demonstrate how the properties of our color palette can be used to efficiently design algorithms for processing of color quantized images. The objective of our approach is to make use of the *natural* Fibonacci ordering of the colors within our palette and define color operators as they are defined for gray-level images.

A. Difference Between Two Colors

In the *Lab* space, the Euclidean distance between two color points carries the information about their visual difference. We show how to compute the Euclidean distance between two colors in the palette based on their symbols q_1 and q_2 and palette parameters. From (1), which relates the the symbol of a color with its luminance and chrominance indexes, we have

$$q_1 = n_1 + i_1 N_p \quad \text{and} \quad q_2 = n_2 + i_2 N_p.$$

Computing the Euclidean distance between two points of the Fibonacci lattice, z_{n_1} and z_{n_2}, is done based on their indexes n_1

and n_2. By applying the Cosine Theorem, we get the following simple expression:

$$||z_{n_1} - z_{n_2}||^2 = s^2 \cdot (n_1 + n_2 - 2\sqrt{n_1 n_2}\cos(2\pi(n_1 - n_2)\tau)).$$

Now, the Euclidean distance between two points q_1 and q_2 in the *Lab* space quantized as described in Section III-B, is given by

$$(L_{i_1} - L_{i_2})^2 + ||z_{n_1} - z_{n_2}||^2 \tag{2}$$

where L_{i_1} and L_{i_2} are the points' luminance values. Note that (2) is just another representation of Euclidean distance between the colors q_1 and q_2 in the 3-D space. Since we "live" in the discrete space where all the lattice points are specified by their integer symbols, we would like this distance to take values from an integer set. This also simplifies further color processing operations on the lattice. Therefore, we rescale (2) to obtain

$$\text{Diff}(q_1, q_2) = \left\lfloor \frac{(L_{i_1} - L_{i_2})^2 + ||z_{n_1} - z_{n_2}||^2}{D_{\min}^2} \right\rfloor. \tag{3}$$

The difference between the points with color symbols q_1 and q_2, as defined by (3), has a physical interpretation: it represents a minimum path that connects q_1 and q_2. In other words, (3) gives the minimum number of steps on the *Lab* space one must take to reach q_2 from q_1 and vice versa.

B. Average Value of Two Colors

In processing grayscale images, the average value of two points q_1 and q_2 is defined as $(q_1 + q_2)/2$. When the resulting number is not an integer, it is rounded to the closest one. The introduced error reflects in the reconstruction as a slightly lighter or darker spot and does not significantly degrade the image quality. It can be seen that rounding in the same manner for a sequence of numbers written in a higher dimensional space, does not give acceptable results regardless of the way the sequence is written [18]. However, the necessary rounding operation can be very elegantly handled for the palette we propose. The *neighborhood* structure of the palette, described in Section III-D, comes in handy here. Suppose, we are to compute the average value of two points with indexes q_1 and q_2, where $q_1 = n_1 + i_1 N_p$, $q_2 = n_2 + i_2 N_p$ and $n_1 > n_2$. The average q_a is given by

$$q_a = n_a + i_a N_p = \text{av}(q_1, q_2) = \frac{(n_1' + n_2)}{2} + \left\lfloor \frac{(i_1 + i_2)}{2} \right\rfloor N_p \tag{4}$$

where n_1' is determined based on the parity of n_1 and n_2 as follows: The value of n_1' is either equal to n_1 (same chroma) or to one of its Fibonacci neighbors with the same parity as n_2(similar chroma). Out of all candidates for n_1', i.e., nearest neighbors of n_1, we select one which guarantees that 1) $n_1 > n_a > n_2$ and 2) $\min\{\theta_{n_1}, \theta_{n_2}\} < \theta_{n_a} < \max\{\theta_{n_1}, \theta_{n_2}\}$. If there are several candidates that fulfill the two requirements, the one closest to n_1 (in Euclidean distance) should be selected. The change in value of n_1 or n_2 reflects in the reconstruction as a spot with a color similar to the original one.

C. Edge Detection

A grayscale edge detection operator can be simply introduced as a gradient between pixel values in a given neighborhood. For color images, edge detection becomes a difficult task not only because of the added dimension of the data, but also because of the added complexity of color perception. Hence, there are different definitions of color edges. An edge in a color image could be defined to exist only if the luminance field contains an edge. However, this definition eliminates discontinuities in hue and saturation in regions of constant luminance. Another approach could be to examine each color component separately and only judge a color edge to be present if an edge exists simultaneously in all three components. A third possibility could be to define an edge to exist if the norm of the vector difference between the 3-D color values of two pixels exceeds some threshold level. The latter two definitions of color edges are quite dependent on the color coordinate system chosen for representation.

The way we defined color difference for the colors from the Fibonacci palette, allows us to extend gradient-based grayscale edge detectors to process color-mapped images. Hence, we first introduce a pair of masks

$$\boldsymbol{H}_x = \begin{bmatrix} -a & 0 & a \\ -b & \underline{0} & b \\ -c & 0 & c \end{bmatrix} \text{ and } \boldsymbol{H}_y = \begin{bmatrix} -a & -b & -c \\ 0 & \underline{0} & 0 \\ a & b & c \end{bmatrix} \tag{5}$$

where $\boldsymbol{H}_x$ measures the gradient of the image along the horizontal direction and $\boldsymbol{H}_y$ along the vertical direction and the underlined elements indicate the location of the origin. The coefficients in (5) depend on the type of operator. For example, for the Prewitt operator $a = b = c = 1$, for the Sobel operator $a = c = 1$, $b = 2$, and for the isotropic operator $a = c = 1$, $b = \sqrt{2}$. Defining the bidirectional gradients

$$g_x(i,j) = \langle I, \boldsymbol{H}_x \rangle(i,j) \text{ and } g_y(i,j) = \langle I, \boldsymbol{H}_y \rangle(i,j) \tag{6}$$

the gradient vector magnitude $g(i,j)$ and direction $\phi(i,j)$ are given by

$$g(i,j) = \sqrt{g_x^2(i,j) + g_y^2(i,j)} \text{ and } \phi(i,j) = \tan^{-1}\frac{g_y(i,j)}{g_x(i,j)}.$$

With the gradient masks as defined in (5), for the grayscale image I the bidirectional gradients (6) in the image location (i,j) are computed as

$$\begin{aligned} g_x(i,j) =& a[I(i-1,j+1) - I(i-1,j-1)] \\ &+ b[I(i,j+1) - I(i,j-1)] \\ &+ c[I(i+1,j+1) - I(i+1,j-1)] \\ g_y(i,j) =& a[I(i-1,j-1) - I(i+1,j-1)] \\ &+ b[I(i-1,j) - I(i+1,j)] \\ &+ c[I(i-1,j+1) - I(i+1,j+1)]. \end{aligned}$$



(a) (b) (c)

Fig. 9. Edge map is computed with the Sobel operator. (a) Original image, (b) edge map obtained by applying the "color" Sobel operator to the color-mapped image, and (c) edge map obtained by applying the traditional grayscale Sobel operator to the red component of the image.

For the color-mapped image P_S, we define the bidirectional gradients in the same manner

$$\begin{aligned} g_x(i,j) =& a \cdot \text{Diff}\left(P_S(i-1,j+1), P_S(i-1,j-1)\right) \\ &+ b \cdot \text{Diff}\left(P_S(i,j+1), P_S(i,j-1)\right) \\ &+ c \cdot \text{Diff}\left(P_S(i+1,j+1), P_S(i+1,j-1)\right) \\ g_y(i,j) =& a \cdot \text{Diff}\left(P_S(i-1,j-1), P_S(i+1,j-1)\right) \\ &+ b \cdot \text{Diff}\left(P_S(i-1,j), P_S(i+1,j)\right) \\ &+ c \cdot \text{Diff}\left(P_S(i-1,j+1), P_S(i+1,j+1)\right) \end{aligned}$$

where $\text{Diff}(\cdot,\cdot)$ denotes the difference between the two symbols from the palette, as defined by (3). As above, coefficients a, b, and c depend on the type of operator. Fig. 9 illustrates the application of the proposed technique.

VI. Compression of Color Quantized Images

A. Background

Traditional lossy compression techniques do not perform well when applied to color-mapped images. This is because most coding schemes assume that neighboring pixels have similar intensity levels (or colors) while most algorithms for generation of color palette do not assign symbols to the colors from the palette in any specific order and symbols pointing to similar colors can be very different. Therefore, coding of color-quantized images is usually performed in the transform domain, independently along each axis of the color space. A luminance–chrominance coordinate space is preferred since the image components are reasonably correlated with each other. The international standard for component video coding relates the digital luminance Y_d and the color difference chrominances C_r and C_b to the *RGB* color system via a linear transform. This transform is usually the first step taken in a DCT-based coding system. Furthermore, since the spatial bandwidth of the chrominance is much smaller than of the luminance, chrominance components are subsampled by the factor two both horizontally and vertically. Finally, the luminance and the chrominance components of the image are individually compressed. The compression is done by applying the DCT on blocks of 8×8 pixels and uniformly quantizing each coefficient. The quantization steps are usually chosen to model the sensitivity of human visual system in different frequency bands. Since most of the quantized coefficients are zero, the resulting image is scanned in a zig-zag pattern, run-length encoded, and then entropy coded. Since each of the color components is coded separately, the pixels of the reconstructed image can take any color value. Thus, when this approach is used to code the color quantized image, the reconstructed image will no longer take values from the original palette and thus has to be quantized again. There are several ways to perform quantization at the decoder. In one approach, the decoded image is color-quantized to the colors of the original color palette. For each image pixel this requires a full search of the palette and, although there are fast algorithms [8], this still involves a significant amount of computation. An alternative approach is to requantize the reconstructed image independently of the original color map [9]. However, this still introduces additional computational complexity at the decoder.

Based on the properties of our color palette, we propose a technique that requires no additional quantization at the decoder, since the reconstructed image contains only colors from the original palette. The main idea behind our technique is to consider each color from the original palette as a symbol and to code the image in the space of the symbols instead of coding it in the 3-D color space. In the case of color palettes produced by the traditional quantization methods, the symbols representing colors have no correlation with the actual color values and within the palette colors appear randomly. Hence, application of any lossy compression technique to symbols from the palette introduces unacceptable distortion and only lossless coding techniques can be used for compression of color-quantized images. An optimal solution to this problem is to order the colors such that consecutive symbols are assigned to visually similar colors. Unfortunately, since the symbol space is one-dimensional (1-D), while colors are defined in the 3-D space, such ordering does not exist for most natural images. A suboptimal solution is to partition the image into small blocks, so that color variation within each block is small and it is very likely that satisfactory color ordering can be performed [9]. Since our sampling method produces symbols that have correlation with color values, we are able to design a compression scheme that operates in the space of color palette symbols.

B. Coding of Color-Quantized Images in the Color Symbols Space

Our method essentially performs multiresolution decomposition of a color-mapped image to obtain a low-resolution approximation of the original color map and a set of images representing the details on different resolutions, along different directions. In our system, one decomposition level consists of 1-D filtering along the preferred direction, followed by the subsampling. Before each decomposition step we use the ordering

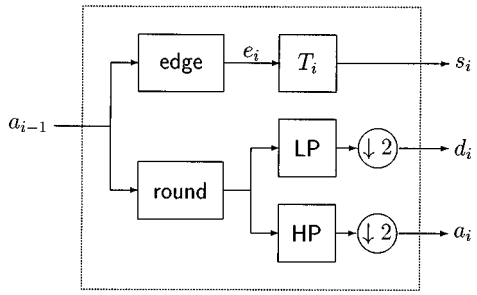


Fig. 10. One decomposition level in the coding algorithm.

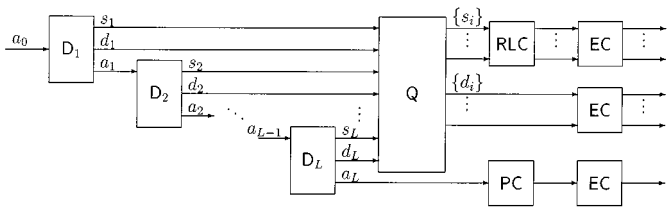


Fig. 11. Structure of the coder.

properties of our color palette to reduce the entropy of color symbols. After the complete decomposition into L-levels, we perform thresholding of the detail signals so that only visually important edges are kept, resulting in significant compression gain. Moreover, in the thresholding stage we exploit the perceptual redundancy among the detail signals. Hence, when a detail coefficient is deleted, the corresponding coefficients from the previous resolution levels are also deleted, providing additional data compression. For each detail image we build the significance map, providing information about the positions of nonzero coefficients. The significance map is first run-length coded and then entropy coded. Nonzero coefficients are entropy coded. Finally, the color-map at the lowest resolution is predictively coded with entropy coding of the prediction errors.

The complete structure of the coder is illustrated in Figs. 10 and 11. Fig. 10 illustrates one decomposition level, while Fig. 11 illustrates the complete coder, consisting of L decomposition levels, coefficient thresholding and lossless compression. We now give the detailed explanation of the whole algorithm. As shown in Fig. 10, the original color-mapped image is taken as an input $a_0(m,n)$ into the decomposition level $i = 1$. Each row (or column) is filtered with the 1-D low-pass and high-pass perfect reconstruction filters and then subsampled by the factor two. Note that the averaging operator [defined in (4)] can be seen as a low-pass filter for the Haar basis. Hence, assuming the Haar filters (with the lowpass filter $l = [1/2\ 1/2]$ and the highpass filter $h = [1/2 - 1/2]$) and input pixels $p_1 = p(m,n)$ and $p_2 = p(m,n+1)$, the filter outputs $a_i(m,n)$ and $d_i(m,n)$ represent the average and difference along the horizontal direction. Note that since the color map values are integers, these operations may result in noninteger values, which increases the entropy of the coefficients. Therefore, prior to filtering we perform the rounding operation on the pixels p_1 and p_2 as explained in Section V-B. Essentially, this operation checks the parity of input pixels and changes it, if necessary, thus forcing the filter outputs to take only integer values. At each decomposition level i, in addition to the approximation and detail signals, we also compute the edge map $e_i(m,n)$ and binary quantize it to obtain the significance map $s_i(m,n)$ as

$$s_i(m,n) = \begin{cases} s_i(m,n) = 1, & \text{if } e_i(m,n) > T_i, \\ s_i(m,n) = 0, & \text{if } e_i(m,n) \le T_i \end{cases}$$

where T_i is the threshold for the particular decomposition level. The edge at the location (m,n) can be computed with any edge detection operator, as explained in Section V-C. The significance maps are used later in the system for quantization of the detail coefficients. Finally, as a result of the decomposition at the level i, we obtain images a_i, d_i, s_i. The resolution of these images along the direction of decomposition is two times lower than the resolution of the input image a_{i-1}. The approximation a_i is then taken as input into the next decomposition level, this time along the opposite direction. As illustrated in Fig. 11, the decomposition is repeated L times, resulting in

1) image approximation a_L having $L/2$ lower resolution than the original color map;
2) set of L detail signals $\{d_i\}$;
3) set of L significance maps $\{s_i\}$, where $i = 1, \ldots, L$.

These signals are then taken as an input into the quantizer $\boldsymbol{Q}$, representing a set of quantization operations $\boldsymbol{Q} = \{q_i\}$, $i = 1, \ldots, L$ in the following way:

$$q_i[d_i(m,n)] = \hat{d}_i(m,n) = \begin{cases} \hat{d}_i(m,n) = d_i(m,n), & \text{if } s_i(m,n) = 1, \\ \hat{d}_i(m,n) = 0, & \text{if } s_i(m,n) = 0. \end{cases}$$

Moreover, to improve the compression of significance maps we take advantage of the spatial redundancy between the detail coefficients. The compression scheme is based on the property that if a detail coefficients at a coarse scale is insignificant with respect to a given threshold T, than all detail coefficient in the same location at finer scales (also called *descendants*) are likely to be insignificant. In our decomposition scheme, at a given scale i, each detail coefficient has two coefficients at the next finer scale $i-1$, corresponding to the same spatial location (except for the detail coefficient at the finest scale $i = 1$). The quantization is performed in such a way, that no finer scale coefficients are quantized before the corresponding coefficients at the coarser scales. Once the descendants of a particular detail coefficient are deleted, the corresponding elements in their significance maps are set to zero. Hence, the final representation we code consists of

1) image approximation a_L (first predictive and then entropy coded);
2) set of L arrays with nonzero detail coefficients for each scale, where each array is entropy coded;
3) set of L significance maps $\{s_i\}_{i=1,\ldots,L}$, where each map is first run-length and then entropy coded.

Since the reconstructed image contains only colors from the original palette, there is no need to perform additional quantization at the decoder. Hence, this coding scheme significantly reduces complexity of the decoder, yet provides comparable quality and the bit rates.

Fig. 12 illustrates the performance of the algorithm. Fig. 12(a) is a full color image "Itl0004" with 330×280 pixels. Fig. 12(b) is image this image quantized to 512 colors (9 bpp). The image



Fig. 12. Compression of palletized images: (top left) original image, (top right) color quantized version obtained with 512 color palette, (bottom left) JPEG compressed and requantized image (0.8 bpp), and (bottom right) compression result with the proposed method (0.8 bpp).

in Fig. 12(c) is obtained by compressing the image in Fig. 12(b) with a conventional JPEG encoder and quantizing it again to the colors from the palette; the bit rate is 0.8 bpp. The image in Fig. 12(d) is obtained by compressing the image in Fig. 12(b) with the proposed method; the bit rate is 0.8 bpp. Decomposition is performed in six levels. The JPEG compressed and requantized image in Fig. 12(c) is clearly superior (in terms of quantization artifacts) to the image in Fig. 12(b), which is compressed with the proposed method. Here, the low complexity and computational advantages of the new method (discussed at the beginning of the section) come at performance cost. However, selection of better filters may eliminate the artifacts.

VII. Conclusions and Future Work

We proposed an entirely novel approach to sampling of the *Lab* color space, based on Fibonacci numbers and spiral lattices. The sampling scheme is to be used to generate color palettes for color quantization, display and processing. The palettes designed with the proposed method have a number of important properties which we illustrated on several examples.

1) Color space sampling and numbering of sampled points makes methods for processing grayscale images extendible to color images.
2) Structure of the Fibonacci palettes enables fast quantization and ordered dither methods can be easily integrated into the quantization procedure.
3) Fibonacci palette is completely specified by a small set of parameters, hence there is no need to store or transmit all the colors in the palette; instead, it is sufficient to store only the parameters that define the palette.
4) Each color in the palette is identified by a single number (symbol) and can be simply reconstructed from it.
5) Display quality of images quantized by the proposed color codebooks is comparable with that of image dependent quantizers for a reasonably large number of colors.

The first property is the most important and opens various directions for future work. Here, we introduced the averaging operation in the symbol space and defined color edge detectors in the same way they are defined for grayscale images. These two operations are essential for many image compression techniques. By using them, we were able to design a color compression algorithm that performs a multiresolution decomposition in the space of color symbols. However, these are only initial examples of image processing algorithms. It remains to be studied how properties of the Fibonacci numbers and Fibonacci lattices can be further exploited and incorporated into many other, more sophisticated processing operations.

Appendix I
Fibonacci Numbers and Phyllotaxy

The sequence of numbers F_n, $n \geq 0$, defined recursively as follows:

$$\begin{aligned} F_0 &= 1,\ F_1 = 1 \\ F_n &= F_{n-1} + F_{n-2},\ n \geq 2 \end{aligned} \tag{7}$$



(a) (b)

Fig. 13. Fibonacci lattice of (a) daisy seedhead and (b) pine cone. (Photos: Scott Camazine, Pennsylvania State University [24].)

is commonly known as the Fibonacci sequence. It was first defined in 1202 by the Italian mathematician Leonardo of Pisa, nicknamed Fibonacci, as a solution to his rabbit population growth puzzle: Assume that at breeding season zero, we start with a single pair of immature rabbits, which mature in one season. Every season, each mature pair produces precisely one immature pair, which again takes one season to mature. Under this assumptions, if rabbits are immortal, the number of pairs of rabbits in each season F_n satisfy recursion (7). The assumptions, of course, do not hold in real life. But Fibonacci numbers, as we shall see in the following, really occur in nature.

The ratios of consecutive Fibonacci numbers converge to the *golden mean*

$$\lim_{n\to\infty} \frac{F_{n-1}}{F_n} = \frac{\sqrt{5}-1}{2}$$

also referred to as golden number, section, or ratio. The golden mean is considered to be one of *nature's special* numbers, (like π or e) particularly related to the geometry of plants and shells. The seed-head of a daisy or sunflower is formed according to the Fibonacci lattice $z_n = \sqrt{n}e^{j2\pi\cdot n\tau}$, where τ is the golden number and n is the number of time units corresponding to the the age of the seed. The lattice can be easily recognized in the plants shown in Fig. 13. More on geometric and numerical patterns in plants can be found in [27] and more technical references therein.

The golden number was known to the ancient Greeks, who thought that a rectangle with the ratio of the sides equal to the golden number has the most pleasing proportions. Such rectangle can be subdivided into a square and a smaller rectangle that has the same proportions as the larger one.

APPENDIX II
CONTINUED FRACTIONS

An expression of the form

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}} \equiv [a_0; a_1, a_2, a_3, \ldots]$$

we shall call a *continued fraction* with value x. The following holds.

Theorem 1

To every real number α, there corresponds a unique continued fraction with value equal to α [25, p. 16]. This fraction is finite if α is rational and infinite if α is irrational.

The rationals p_n/q_n given by

$$\frac{p_n}{q_n} = [a_0; a_1, a_2, a_3, \ldots a_n]$$

are known as *convergents*, or nth step truncations, of the continued fraction expansion. The continued fraction expansion of the golden number is given by

$$\frac{\sqrt{5}-1}{2} = [0; 1, 1, \ldots]$$

and its convergents are ratios of successive Fibonacci numbers, i.e., $p_n/q_n = F_{n-1}/F_n$. *Noble numbers* are those numbers whose continued fraction expansion tails agree with that of the golden ratio.

The convergents p_n/q_n provide good rational approximations of α in the sense that there exists a constant c such that $|\alpha - p_n/q_n| < c/q_n^2$. As discussed in Section III-A, this rational approximation determines the axial distribution of the lattice points $z_n = \sqrt{n}e^{j2\pi\cdot n\alpha}$. Therefore, we are interested in an irrational number x "badly" approximated by its convergents, i.e., the one which maximizes c_α as

$$x = \arg\min_{\alpha} c_\alpha, \text{ where}$$
$$c_\alpha = \inf\left\{c : \left|\frac{\alpha - p_n}{q_n}\right| < \frac{c}{q_n^2} \text{ for infinitely many } q_n\right\}.$$

It turns out that $c_\alpha \le 1/\sqrt{5}$ for all α, with equality if and only if α is a noble number, e.g., $x = (\sqrt{5}-1)/2$.

It is then not surprising that plants chose this (in a way) optimal packing. Although the hexagonal lattice provides the best sphere packing in two dimensions, because of the growth dynamics and circular shape of flower-heads, the seeds follow the spiral lattice rule.

The set of bad irrationals (noble numbers) was extended by Markoff in 1879, who showed that there is a discrete set of values c_i decreasing to 1/3 such that, if $c_\alpha > 1/3$, then$c_\alpha = c_i$ for some $i \in \mathbf{N}$. The numbers c_i are called the *Markoff spectrum* and the corresponding α's Markoff irrationals. Spiral lattices $z_n = \sqrt{n}e^{j2\pi \cdot n\tau}$, where τ is a Markoff irrational, can therefore be used for quantization of color spaces. It is important to note that for practical purposes of designing palettes with finite number of colors, some rational numbers will do a good job as well.This was illustrated by examples of spiral lattices for four different values of τ shown in Fig. 3.

ACKNOWLEDGMENT

The authors are grateful to M. Finston, V. Goyal, J. Kovačević, J. Mazo, and V. Weerackody for their comments on an earlier version of this manuscript.

REFERENCES

[1] M. T. Orchard and C. A. Bouman, "Color quantization of images," *IEEE Trans. Signal Processing*, vol. 39, pp. 2677–2689, Dec. 1991.
[2] R. Gentile, J. Allebach, and E. Walowit, "Quantization of color images based on uniform color spaces," *J. Imag. Technol.*, vol. 16, no. 1, pp. 11–21, February 1990.
[3] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[4] F. Rothen and A. Koch, "Phyllotaxis or the properties of spiral lattices," *J. Physique*, vol. 50, pp. 633–657, Mar. 1989.
[5] R. Gentile, E. Walowit, and J. Allebach, "Quantization and multilevel halftoning of color images for near original image quality," *J. Opt. Soc. Amer. A*, vol. 7, no. 6, pp. 1019–1026, June 1990.
[6] X. Wan and C.-C. J. Kuo, "A new approach to image retrieval with hierarchical color clustering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 442–453, Oct. 1993.
[7] G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*. New York: Wiley, 1982.
[8] C.-M. Huang *et al.*, "Fast full-search equivalent encoding algorithm for image compression using vector quantization," *IEEE Trans. Image Processing*, vol. 1, pp. 413–416, July 1992.
[9] A. Zaccarin and B. Liu, "A novel approach for coding color quantized images," *IEEE Trans. Image Processing*, vol. 2, pp. 442–453, Oct. 1993.
[10] R. Machuca and K. Phillips, "Application of vector fields to image processing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 5, pp. 318–329, 1983.
[11] C. K. Yang, T. C. Wu, J. C. Lin, and W. Tsai, "Color image sharpening by moment-preserving techniques," *IEEE Trans. Signal Processing*, vol. 45, pp. 397–403, 1995.
[12] S. Pei and C. Cheng, "Color image processing by using binary quaternion moment-preserving thresholding technique," *IEEE Trans. Image Processing*, vol. 8, pp. 614–628, May 1999.
[13] R. Balasubramanian, C. A. Bouman, and J. P. Allebach, "Sequential scalar quantization of color images," *J. Electron. Imag.*, vol. 3, pp. 45–59, Jan. 1994.
[14] X. Wu, "YIQ vector quantization in a new color palette architecture," *IEEE Trans. Image Processing*, vol. 5, pp. 321–329, Feb. 1996.
[15] R. Balasubramanian and J. P. Allebach, "A new approach to palette selection for color images," *J. Imag. Technol.*, pp. 284–290, Dec. 1991.
[16] G. Sharma and H. J. Trussell, "Digital color imaging," *IEEE Trans. Image Processing*, vol. 6, July 1997.
[17] A. Mojsilovic and J. Hu, "Extraction of perceptually important colors and similarity measurement for image matching," in *IEEE Int. Conf. Image Processing 2000*.
[18] H. Sagan, *Space-Filling Curves*. New York: Springer-Verlag, 1991.
[19] A. Lempel and J. Ziv, "Compression of two-dimensional data," *IEEE Trans. Inform. Theory*, vol. 32, pp. 2–8, Jan. 1986.
[20] A. Ansari and A. Fineberg, "Image data ordering and compression using Peano scan and lot," *IEEE Trans. Consumer Electron.*, vol. 38, pp. 436–445, June 1992.
[21] K.-M. Yang, L. Wu, and M. Mils, "Fractal based image coding scheme using Peano scan," in *1988 IEEE Int. Symp. Circuits Systems '88*, Helsinki, Finland, June 1988, pp. 2301–2304.
[22] E. A. Patrick, D. R. Anderson, and F. K. Bechtel, "Mapping multidimensional space to one dimension for computer display," *IEEE Trans. Comput.*, vol. 17, pp. 949–953, Oct. 1968.
[23] R. J. Stevens, A. F. Lehar, and F. H. Preston, "Manipulation and presentation of multidimensional image data using the Peano scan," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 5, pp. 520–526, Sept. 1983.
[24] S. Camazine.. Dept. Entomology, Penn. State Univ., University Park. [Online]. Available: http://www.psu.edu/dept/beehive
[25] A. Y. Khinchin, *Continued Fractions, Mineola*. New York: Dover, 1997.
[26] C. Series, "The geometry of Markoff numbers," *Math. Intell.*, vol. 7, pp. 20–29, Mar. 1985.
[27] I. Stewart, *Life's Other Secret*. New York: Wiley, 1998.

Aleksandra Mojsilović (S'93–M'98) was born in Belgrade, Yugoslavia, in 1968. She received the B.S.E.E., M.S.E.E., and Ph.D. degrees from the University of Belgrade in 1992, 1994, and 1997, respectively.

From 1994 to 1998, she was a Member of Academic Staff with the Department of Electrical Engineering, University of Belgrade. From 1998 to 2000, she was with Bell Laboratories, Lucent Technologies. She is currently a Research Staff Member at IBM T. J. Watson Center, Hawthorne, NY. Her main research interests include computer vision, image analysis, and multidimensional signal processing.

Dr. Mojsilović is a member of the IEEE Signal Processing Society and an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING.

Emina Soljanin (S'93–M'95) received the B.S. degree in electrical engineering from Sarajevo University, Bosnia and Herzegovina, in 1986 and the M.S. and Ph.D. degrees in electrical engineering from Texas A&M University, College Station, in 1989 and 1994.

From 1986 to 1988, she was with Energoinvest Company, Bosnia and Herzegovina, developing optimization algorithms and software for power system control. She is a Member of Technical Staff with the Mathematical Sciences Research Center, Bell Laboratories, Murray Hill, NJ. Her research interests are in the broad area of communications, information and coding theory, their applications to storage systems, and, more recently, quantum information theory.

Dr. Soljanin was the recipient of the 1992 Texas A&M University Electrical Engineering Department Fouraker Fellowship. She served as a Technical Proofreader, (1990–1992) and Associate Editor for Coding Techniques (1997–2000) for the IEEE TRANSACTIONS ON INFORMATION THEORY.