A Novel Algorithm for Color Quantization by 3D Diffusion

K. C. Lo, Y.H. Chan, and M.P. Yu
Dept. of Electronic and Information Engineering
The Hong Kong Polytechnic University
Kowloon, HONG KONG

ABSTRACT

A novel algorithm for color quantization, three-dimensional frequency diffusion (3D FD), applies to the histogram of an image based on the principle of error diffusion using a 3D error diffusion filter. With the histogram divided into overlapping cubes, an iterative process is devised to select representative colors from these cubes by a popularity scheme that considers a neighborhood of pixels until a colormap is filled. The algorithm is simple but effective. Results show that images quantized by the proposed algorithm are comparable in quality to other popular algorithms.

1. INTRODUCTION

Full color displays typically use 24 bits to represent the color of each pixel on a screen. With 8 bits for each of the primary components, red, green and blue, approximately 16.7 million possible colors can be generated in a 24 bit full RGB color image. In many imaging applications, it is often desirable to represent images with as few colors as possible while at the same time retain the optimal perceived quality of the images. This is generally achieved by color quantization, in which the number of colors in an image is reduced to a maximum of 256 simultaneously displayable colors and followed by subsequent halftoning to create the illusion of a continuous tone image. Some well-known techniques to create an image colormap or palette are the popularity algorithm [1][2], the Center-cut [3], the Octree quantization [4], Orchard-Bouman splitting algorithm [5] and Wu's algorithm by principal analysis [6]. Important developments in digital halftoning include analog screening, noise encoding, ordered dithering, error diffusion and stochastic screening [7]. Among the various methods, ordered dithering and error diffusion are the two main classes of conventional digital halftoning techniques. While clustered dots are often preferred for printing, error diffusion is perhaps the most widely used approach for displays [8]. Error diffusion is an adaptive algorithm that operates by spreading or diffusing the quantization error of a current pixel to neighboring pixels. This method was first proposed by Floyd and Steinberg [9] and was originally used for gray-scale images. For color images, error diffusion is either performed separately in each independent channel (scalar error diffusion) or simultaneously in a three-dimensional (3D) color space (vector error diffusion) [10]. For color displays with an image dependent palette, vector error diffusion is often used.

We propose a simple but effective color quantization technique, which performs 3D frequency diffusion in a color space. Our approach is to select the colors of an image with the highest frequency of occurrence to be the colormap entries from the corresponding 3D color histogram. To avoid choosing too many entries from clusters nearby, the frequencies of colors in a neighborhood are considered. We identify the color representatives by iteratively divide the 3D color histogram into subspaces in a deterministic manner and compare their frequencies until the subspace contains one target color, i.e., a cluster. Each cluster of the histogram carries a fractional dot value d equal to its frequency density times the number of entries of a colormap. When a cluster is chosen, an integer value nearest to d will be removed from the subspace. An error will thus be introduced and diffused to its neighbors according to a 3D frequency diffusion filter. In a normal image, most clusters carry a d less than 1. In this way, a self-correcting task is performed to penalize the neighbors of a color selected each iteration. The algorithm is applicable to images in any color space.

2. 3D FREQUENCY DIFFUSION

2.1 Neighborhood

Consider a population residing in the RGB color space. Every pixel in an image is treated as a 3×1 vector that constitutes a single point, $v_s = (v_s^R, v_s^G, v_s^B)^T$, where s = 0, 1, 2,… $(L_1 \times L_2)-1$ for an image of size $(L_1 \times L_2)$, and T denotes the transpose of a vector. In this color space, pixels of the same color are accumulated, giving rise to a 3D color histogram. Let X be the input color space of size N×N×N, which contains the histogram, where $N = 2^r$ and r equals to the number of bits per channel. Usually r is smaller than the original bit resolution of the image so as to scale down the histogram to better fit a palette of 256 entries. For example, a 24-bit image having 8 bits/channel would have a histogram of 5 bits/channel or N = 32. There are 32×32×32 = 32,768 clusters or cells. With most images, a 5-bit histogram is sufficient to yield good results.

0-7803-7761-3/03/$17.00 ©2003 IEEE

Our algorithm is a two-step iterative process. Each iteration, we first identify the colors of the image having the highest frequency of occurrence according to the input histogram X. With a naive scheme that picks all the clusters with the highest frequencies, it is likely that the palette will contain many entries of similar colors, neglecting other less popular colors. To prevent this, we consider a neighborhood or subset of a color space:

$$X_k(l_k,m_k,n_k) = \{X_{k-1}(l_k+x, m_k+y, n_k+z) \mid x, y, z \in 0,1,\ldots, 2w-1\} \quad (1)$$

where $k = 1,2, \ldots,.r-1$, $w = 2r\text{-}k\text{-}1$, and l_k, m_k and n_k each takes the values of 0, w and $2w$.

In eq (1), a higher k refers to a smaller subspace such that X_0 represents the original input. A space is divided into 27 overlapping subspaces, each occupying 1/8 of that space. The frequency f_k associated with a subspace X_k is defined as

$$f_k(l_k,m_k,n_k) = \sum_{x=0}^{(2w-1)} \sum_{y=0}^{(2w-1)} \sum_{z=0}^{(2w-1)} X(l_k+x, m_k+y, n_k+z) \quad (2)$$

The highest f_k is selected and the corresponding X_k is subdivided according to eq (1) until X_{r-1}, which is the finest level containing 8 clusters of the original histogram in each subspace. After identifying the particular X_{r-1} with the highest f_{r-1}, the last step is to pick the single cluster having the highest frequency from that subspace, which does not require any summation in eq (2). The color of the final cluster becomes an entry in the palette.

The process allows flexibility by providing more possible paths to reach a particular location without being trapped in a local optimum. With N=32, r = 5, the index k varies from 1 to 4 and each color space is divided into 27 overlapping subspaces as shown in Fig. 1. A palette entry is determined after 4 iterations of dividing color spaces and 5 iterations of seeking the maximum frequency.

2.2 Frequency Diffusion Process

The principle of frequency diffusion is similar to error diffusion that once a pixel has been quantized, error is introduced, which will affect the next color to be picked. Here frequency diffusion is performed in a 3D color space after choosing a cluster to be an entry in the palette. To borrow the concept of quantization, an integer dot value is defined. Let f_t be the total frequency of X and f the frequency of a cluster in X. With an 8-bit palette, we can scale f_t to 256. Then the frequency density $\phi = f / f_t$ can be calculated and a fractional dot value $d = 256\times\phi$ can be assigned to a cluster. When a cluster is chosen, a dot value is assigned by a quantization process Q( ):

$$Q(d) = \text{dot} = \max(1, \text{round}(d)) \quad (3)$$

where round( ) rounds the argument to the nearest integer. This function is required because there may exist in a picture a few clusters with a dot value larger than 1. Assuming that values in X are normalized to 256, the error is:

$$e(l, m, n) = X(l, m, n) - \text{dot}(l, m, n) \quad (4)$$

The quantization error $e(l,m,n)$ is then diffused and added to the frequency of the neighboring cluster, giving rise to an error histogram E. A 3D filter with a convenient geometric shape, e.g. a sphere, and coefficients ω is defined for the diffusion process. The error histogram is updated as follows:

$$E(i,j,k) = \begin{cases} 0, & \text{if } (i,j,k) = (l,m,n) \\ X(i,j,k) + \omega_{ijk} e(l,m,n), & \text{otherwise} \end{cases} \quad (5)$$

where the sum of all $\omega_{ijk} = 1$. When a part of the filter extends outside the boundary of the color space, the part is folded back into the space. After the frequency diffusion, the histogram X is updated. The quantization process is repeated until all palette entries are filled.

The frequency diffusion process regulates the selection of the palette entries such that representative colors may be chosen from clusters not too close together. In a normal picture, most dot values d are less than 1, which will generate negative errors e. If Ω is the window of support for the frequency diffusion filter, all clusters within Ω are penalized in being selected in the following iteration. Hence more variation in the palette will be obtained.

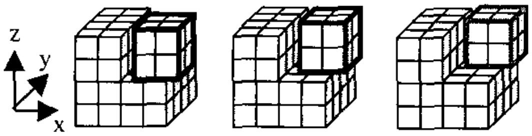

Figure 1. A schematic showing 3 of the 27 overlapping subspaces obtained according to eq (1). Each side of a small cube is w. From left to right, they are $X_k(2w,0,2w)$, $X_k(2w,w,2w)$ and $X_k(2w,2w,2w)$.

2.3 Algorithm

S1: Prepare a suitable color histogram X_0 of dimensions N×N×N in a color space. For example, with a 24-bit image, we can set up a 5-bit histogram in the RGB space, i.e., there are 32×32×32 clusters. A 5-bit histogram is sufficient for most pictures.

S2: Specify the shape and coefficients of a 3D filter for frequency diffusion.

Initialize an error histogram $E(x,y,z)$ having the same size as X_0. Set $r = \log_2 N$ and $k = 1$.

S3: Divide an input histogram X_{k-1} into 27 overlapping subspace according to eq (1). Find the frequency f_k of each subspace X_k according to eq (2). Seek the subspace X^m_k having the highest f_k.

Increment k. Set X^m_k as X_{k-1}, the input histogram, and repeat Step 3 until $k = r - 1$.

S4: Identify the cluster with the highest frequency from the 8 clusters within X_{r-1}, the final histogram found in S3. Select the color of that cluster as an entry of the palette.

S5: Scale the total frequency f_t to 256 dots. Determine the dot value of the cluster found in Step 4 according to eq (3). Compute and diffusion the error as stated in eqs (4) and (5). Update the histogram X_0.

S6: Repeat S3 to S5 until the whole palette is filled.

3. EXPERIMENTAL RESULTS

A representative set of images has been chosen to include both areas of smooth gradation (low frequency) and fine details (high frequency) for evaluation of our proposed quantization algorithm (Three of these images are shown in Fig. 2). To start with, we test our algorithm in the RGB color space. For the frequency diffusion, spherical filters enclosed in 3×3×3 and 5×5×5 cubes are used, and the filter coefficients are given in Fig. 3.

To compare our algorithm with the other color quantization schemes, we use the peak signal-to-noise ratio (PSNR) as a gauge to measure the effectiveness of the resulting palettes. Given the original image I having dimensions of M×N and a quantized image J, the PSNR in decibels (dB) is computed as follows

$$PSNR = 10\log_{10}\left\{\frac{MN \times 255^2}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[I(i,j) - J(i,j)]^2}\right\} \quad (6)$$

The PSNR values of several test images quantized by the 3D frequency diffusion algorithm and by two other algorithms, Median Cut and Octree under RGB color space are given in Table 1. All the images produced by 3D frequency diffusion give high values in PSNR, indicating a good performance of our proposed algorithm compared to the other color quantization schemes.

The appearance of contouring artifacts is one of the major problems encountered by color quantization, especially in regions exhibiting smooth color transition, such as the background in "Woman" and the billiard balls in "Pool". This problem can be solved with spatial error diffusion by using the Floyd and Steinberg error diffusion filter [9] during pixel mapping. To study the overall performance of 3D frequency diffusion with spatial error diffusion, we measure the color reproduction errors by using the S-CIELAB color difference metric [11]. The S-CIELAB calculation is a spatial extension of CIELAB in which factors related to the pattern-color sensitivities of the human eye are incorporated in measuring color differences in digital images. The result is an error map with one ΔE value per pixel indicating the difference between the S-CIELAB representation of the original image and the quantized image. Table 2 shows that our algorithm behaves equally well as Median Cut and more consistently with different types of images than Octree.

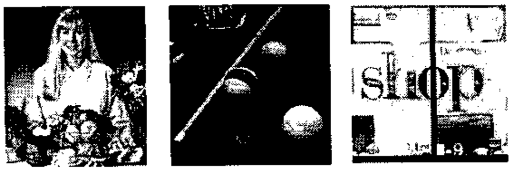

Figure 2. Three of the test images: from left to right, Woman, Pool and Shop

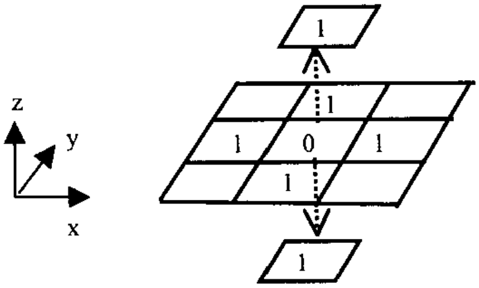

Figure 3. Weights of 3D frequency diffusion filter SP3, where the center carries a weight 0. Radius = 1 in D_4 (4-neighbor) distance. Filter coefficients $\omega_{xyz} = w_{xyz}/6$, where $w_{xyz} = 1$ if distance D = 1 as shown on the diagram; otherwise $w_{xyz} = 0$.

Weights of filter of 5×5×5

SP5a: Radius = 2 in D_4 distance. Weight w = 1 if distance D = 1 and 2, otherwise w = 0. Coefficients $\omega_{xyz} = w_{xyz}/\Sigma w_{xyz}$

SP5b: w = 1 if D = 1 and z = 0; w = 0.5 if D = {1,2} and $z \neq 0$

SP5c: w = 1 if D = 1; w = 0.5 if D = 2

4. CONCLUSIONS

The proposed algorithm, 3D FD, realizes frequency diffusion with the histogram of a color image. The process is simple yet effective. It is applicable to images of any size in any color space. All steps require only simple arithmetical calculations, but the algorithm performs consistently well with various types of pictures. The most time-consuming part of the algorithm is the preparation of the histogram, which has a complexity of N^2 (square class). For the part of frequency diffusion, the size of the histogram is fixed, i.e., independent of the image size, and so is the diffusion filter, which cannot be larger than the histogram itself. Hence the computational complexity of this part is a constant. Other algorithms, such as Median Cut, Principle Component Analysis and Octree also have an N^2 complexity. However, they require more complicated mathematical operations like finding statistical parameters or Eigenvectors of the images.

	Image / Filter*	Woman	Pool	Shop	Blythe	Musi-cians
3D FD	SP3	29.34	30.88	28.82	29.61	30.09
	SP5a	29.23	30.71	28.98	29.54	30.01
	SP5b	29.28	30.76	29.07	29.52	30.07
	SP5c	29.27	30.81	29.01	29.55	30.08
others	Median Cut	29.49	32.63	28.84	29.90	29.82
	Octree	30.50	33.72	28.47	30.44	29.76

Table 1. PSNR (in dB) of pictures reproduced after color quantization by 3D FD, Median cut and Octree under RGB color space.

* Format of filter in 3D FD, SPms :

m – the filter is enclosed in an m×m×m cube, e.g. 3×3×3 a, b, c – different filter coefficients as shown in Fig. 3

5. ACKNOWLEDGEMENT

This work is supported by the Center for Multimedia Signal Processing, The Hong Kong Polytechnic University.

6. REFERENCES

[1] Heckbert, P. "Color image quantization for frame buffer display", Computer Graphics 16, pp. 297–307. 1982.

[2] Braudaway, G. "A procedure for optimum choice of a small number of colors from a large color palette for color imaging", *Proc. Electronic Imaging '87*, San Francisco, pp.75–79, 1987..

[3] Joy, G. and Xiang, Z. "Center-cut for color image quantization", *The Visual Computer* 10, pp. 62-66, 1993.

[4] Gervautz, M. and Purgathofer, W. "A simple method for colour quantization: Octree quantization" In Thalmann and Thalmann, Ed., *New Trends in Computer Graphics*. Springer, New York, pp.219–231, 1988.

[5] Orchard, M. T. and. Bouman, C. A. "Color quantization of images", *IEEE Trans. Signal Processing* 39, pp. 2677–2690, 1991..

[6] Wu, X. "Color quantization by dynamic programming and principal analysis", *ACM Trans. on Graphics* 11, pp. 348–372, 1992.

[7] Kane, H. R. *Digital Color Halftoning*, IEEE Press, USA., 1996.

[8] Sharma, G. and Trussell, H. J. "Digital color imaging", *IEEE Trans. Image Processing* 6 (7), pp. 901-932, 1997.

[9] Floyd, R. and Steinberg, L. "An adaptive algorithm for spatial grey scale", *Proc. Soc. Inf. Disp.* 17 (12), pp. 75-77, 1976.

[10] Knox, K. T. "Evolution of error diffusion", *Journal of Electronic Imaging* 8 (4) , pp. 422-429., 1999.

[11] Zhang, X. and Wandell, B. A. "A spatial extension to CIELAB for digital colour image reproduction" *Soc. Inf. Disp. Symposium Technical Digest*, SID 27, pp. 731-734, 1997.

Image / Filter	Woman			Pool			Shop			Blythe		
	median	mode	>3ΔE (%)	median	mode	>3ΔE (%)	median	mode	>3ΔE (%)	median	mode	>3ΔE (%)
SP3	1.0570	0.54	13.00	0.7816	1.09	3.63	0.9467	0.41	13.69	0.7655	0.03	13.08
SP5a	1.0636	0.54	14.65	0.7792	1.09	3.56	0.9389	0.41	14.99	0.8128	0.04	15.81
SP5b	1.0563	0.54	14.50	0.7681	1.09	3.29	0.9417	0.41	15.56	0.7945	0.03	16.71
SP5c	1.0540	0.57	13.64	0.7757	1.09	3.48	0.9362	0.41	14.45	0.7951	0.04	15.31
MC	1.3110	0.65	16.13	0.7872	1.09	3.63	1.1337	0.41	13.69	0.7856	0.03	9.11
OC	1.1186	0.46	8.20	0.9753	3.19	18.12	1.7045	1.35	19.37	0.8622	0.04	5.94

Table 2. Statistical parameters of S-CIELAB ΔE of 3D FD under RGB color space followed by spatial error diffusion. Median Cut (MC) and Octree (OC) under RGB space with error diffusion are included for comparison.