

Filter Designs for Multiresolution Halftoning

Nancy Breaux
Naval Surface Warfare Center
Dahlgren Division
Dahlgren, Virginia 22448, U.S.A.

Chee-Hung Henry Chu
The University of Southwestern Louisiana
Center for Advanced Computer Studies
Lafayette, Louisiana 70504, U.S.A.

Abstract

Halftoning is used to create the illusion of new colors when only a limited number of colors is available for rendering. Many techniques for color halftoning are based on techniques taken from monochrome halftoning, but modified so that the algorithm is applied to each color component separately. These techniques work well if the palette is separable, but for the case of arbitrary palettes, only error diffusion has been widely used or studied, especially in the context of rendering true-color images on color indexed display systems. We present an alternative algorithm for color halftoning whether the color palette is arbitrary or image dependent. A systematic way of developing suitable filters for this multiresolution halftoning technique is described.

Introduction

Since most halftoning algorithms for printing deal with color as three superimposed fixed color scalar images, they cannot be easily adapted when the colors used to render an image form an arbitrary palette. At present, only a modified form of error diffusion is being used in an attempt to dither color images with arbitrary color tables on color-indexed displays (see, e.g., [1,2,3]). On such displays, this method leads to several artifacts such as color shifts and color spikes. Most research into the problem of improving image quality for color indexed displays has focused on developing better color tables, and little research has been devoted to better halftoning algorithms.

A new multiresolution color halftoning algorithm is presented which greatly improves image quality for low bit depth color images. Unlike error diffusion, it cannot produce a color shift and rarely produces color spikes. An image is first quantized. Its pixels are then altered by considering the image at various scales to form a halftone image. The algorithm is described in Section 2. Experimental results demonstrating the algorithm's performance are shown in Section 3. Finally, our conclusions are drawn in Section 4.

Multiresolution Halftoning

Our algorithm is based on a halftoning algorithm for gray scale images developed by Eli Peli.[4] Peli's algorithm was designed to combine the best properties of ordered dither and error diffusion for halftoning grayscale images. The basic idea of Peli's algorithm is to refine a quantized image by readjusting certain pixels. The pixels adjusted are centers of overlapping square windows over which a weighted average is taken. If switching the center pixel's value results in a lowering of the average error in the window, then the value is readjusted. Initially, 3×3 windows are used, followed by windows of sizes 5×5, 7×7, and 9×9. The largest matrix used depends on the size of the smallest dot or pixel and should correspond roughly to the maximum area over which the human eye takes a local average. For instance, if dots spaced every 3 pixels apart are averaged together to produce a new gray level or color, but pixels spaced 4 pixels apart look like isolated dots, then the maximum size of matrix that should be used is 5×5.

One of the main problems with the original Peli's algorithm is that when applied to a gray ramp, the gray levels are quantized to bands of differing size. These problems of having too few gray levels and the nonlinear division of the gray ramp can be overcome by properly adjusting the weights of the matrix for the averaging process. In our design, the overall shape of the matrix is that of a truncated circular Gaussian distribution, the sum of whose weights are normalized to one. The standard deviation of the distribution determines how flat the matrices will be. Like the maximum size of the matrix used, this parameter depends on dot size and spacing. For the case of color-indexed displays, a standard deviation of one is suitable.

We consider the reasoning behind the development of the 5×5 matrix, denoted W_5, to illustrate the method. Let the gray levels be normalized to the range between 0 and 1. Consider the dithering of a flat gray region of an intensity between 0 and 1/2. Since the gray level is less than 1/2, the initial quantization by thresholding the gray level image will produce a constant output map of 0. The subsequent adjustment of the output map can turn some of the pixels to 1, representing a possibly quantized gray value. If we consider only 3×3 and 5×5 filters, the entire image is tiled by a 4×4 block:

3		3	
	5		
3		3	
			5

where "3" and "5" mark the centers of the 3×3 and 5×5 filters, respectively. Since only those pixels that are centers of windows will be adjusted, there are four possible gray level outputs when dithering a flat gray region: 0, 1/8, 1/4, and 3/8, depending on various combinations of 0 and 1 values taken by the centers of 3×3 and 5×5 windows.

For the gray ramp to be evenly divided, the thresholds dividing the intervals are 1/8, 1/4, 3/8 and 1/2 The ideal behavior of the algorithm for flat gray regions with intensity between 0 and 1/2 is as follows. In (0, 1/8), all output pixels should be 0. In (1/8, 1/4), only the center of 5×5 windows should be 1, corresponding to an average window value of 1/8. In (1/4, 3/8), only the center of 3×3 windows should be 1, corresponding to an average window value of 1/4. In (3/8, 1/2), all the center of windows should output 1, corresponding to an average window value of 3/8.

The task is to adjust the 3×3 and 5×5 Gaussian matrices such that the above behavior is realized. The weights in W_5 can be divided into three classes: ordinary quantized pixels, centers of 3×3 windows, and the center of the 5×5 matrix. The centers of 3×3 windows are the four diagonal neighbors of the center of the matrix, while the ordinary quantized pixels are the remaining 20 pixels. Let *A* be the sum of the weights corresponding to ordinary pixels, *B* be the sum of the weights of 3×3 window centers, and *C* be the center weight of the matrix.

We can find the relative weights of A, B, and C based on the constraints derived from the desired outputs at the three thresholds of the interval [0, 1/2).[5,6] From these constraints, we find that A = 1/2, B = 1/4, and C = 1/4. Since B = 1/4, the sum of the elements corresponding to the centers of windows must be 1/4. Since each of these elements is equidistant from the center of the matrix, each of these elements will have a weight of 1/16 in the final matrix. Since A = 1/2, the rest of the elements must sum to 1/2. These elements are set so that their sum is 1/2, and their weights relative to one another are the same as for the original Gaussian matrix.

The matrices W_3, W_7, and W_9 can be derived using similar reasoning. The coefficients of W_3 and W_5 are shown in Figure 1 and 2, respectively.

.0472	.0778	.0472
.0778	.5000	.0778
.0472	.0778	.0472

Figure 1. Coefficients of the optimum W_3 filter.

.0025	.0111	.0183	.0111	.0025
.0111	.0625	.0820	.0625	.0111
.0183	.0820	.2500	.0820	.0183
.0111	.0625	.0820	.0625	.0111
.0025	.0111	.0183	.0111	.0025

Figure 2. Coefficients of the optimum W_5 filter.

The improved multiresolution halftoning algorithm can be applied to color images. Let *I* be a color image, whose elements are each a vector of three elements. Let *I(i, j; k)* be the *k*th element of the color vector at pixel position *(i,j)*. Let *N* be a palette of colors. The *m*th entry in the palette, *P(m)*, is a color vector of three elements, and can be thought of as the index into the color table. Let *Q* be the quantized and halftoned image. Initially, *Q(i,j) = P(m)* where *P(m)* is the color in the palette that is closest to *P(i,j)*. The quantization error at each pixel is computed. At the *n*th resolution, the quantization error is convolved with the corresponding W_n filter. At each pixel, the quantized pixel *Q(i,j)* is set to a color in the palette *P(m)* such that if *I(i,j) = P(m)*, the averaged error is minimized. The quantization error is updated after each pass.

While the results shown below used colors in the RGB-space, it is also possible to use the same algorithm in the YIQ- or L*a*b*-space. For such color spaces, the matrix used to average over the luminance component can be based on a Gaussian distribution with a smaller standard deviation than the matrices used for the chrominance components.

Experimental Results

The multiresolution algorithm was applied to the Lena image. Figure 3 shows the Lena image quantized to six random colors plus black and white. Figure 4 shows the result using the same eight colors with multiresolution halftoning. Figure 5 shows the result using Floyd and Steinberg error diffusion. Although the error diffused image is smoother, the colors of the multiresolution image are closer to the original Lena image.

Figure 6 shows the Lena image quantized to eight colors chosen using Heckbert's median cut algorithm. Figure 7 shows the result using the same eight colors with multiresolution halftoning. Figure 8 shows the result using the same eight colors with error diffusion. In this figure, there is a noticeable color shift, an unpredictable artifact of error diffusion. The multiresolution color halftoning algorithm does not produce color shifts of this type because errors from quantizing one part of an image cannot be carried over to another part of the image. One may also notice the loss of resolution in the feather region of Figure 8 due to the large number of color spikes.

Conclusion

Until now, the only widely used algorithm for halftoning images using arbitrary palettes was error-diffusion. Unfortunately, using error diffusion could result in color shifts, color spikes, wormy textures, and loss of detail. Multiresolution color halftoning can create new shades of color from a limited palette without unexpected color shifts. When used for color-indexed displays, multiresolution color halftoning reduces the appearance of contouring without introducing objectionable artifacts and with more detail than have images produced using error diffusion.

Acknowledgments

This work was supported in part by the U.S. Department of Energy through grant no. DE-FG02-97ER1220.

References

1. L. Akarun, Y. Yardimci, and E. Cetin, "Adaptive methods for dithering color images," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 950--955, Jul. 1997.
2. C.Y. Kim, I.S. Kweon, Y.S. Seo, "Color and printer models for color halftoning," *Journal of Electronic Imaging*, vol. 6, no. 2, pp. 166--180, Apr. 1997.
3. T. Pappas, "Model-based halftoning of color images," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 1014--1024-955, Jul. 1997.
4. E. Peli, "Multiresolution, error-convergence halftone algorithm," *Journal of the Optical Society of America*, ser. A., vol. 8, no. 4, pp. 625--636, Apr. 1991.
5. N. Breaux, *Data compression and halftone rendering for grayscale and color images*, Ph.D. dissertation, the University of Southwestern Louisiana, Lafayette, La., 1998.
6. N. Breaux and C. H. Chu, "Halftoning for color-indexed displays," to be presented at the *1999 IEEE International Conference on Image Processing,* Kobe, Japan, 1999.

Biography

Nancy Breaux received her B.S.E. (Electrical Engineering), M.S. and Ph.D. degrees in computer engineering, in 1991, 1993, and 1998, respectively, all from the University of Southwestern Louisiana. She is currently with the Naval Surface Warfare Center-Dahlgren Division. Her technical interests are in signal and image processing. She is a member of the IEEE.

Chee-Hung Henry Chu received his B.S.E. (Computer Engineering) and M.S.E. (Computer, Information, and Control Engineering) degrees from the University of Michigan, Ann Arbor, in 1981 and 1982, respectively, and a Ph.D. in Electrical Engineering from Purdue University in 1988. Since 1988 he has been with the University of Southwestern Louisiana. His technical interests are in signal and image processing, computer graphics, and visualization. He is a member of the ACM and IEEE.

Figure 3. Lena image quantized to black, white, and six random colors.

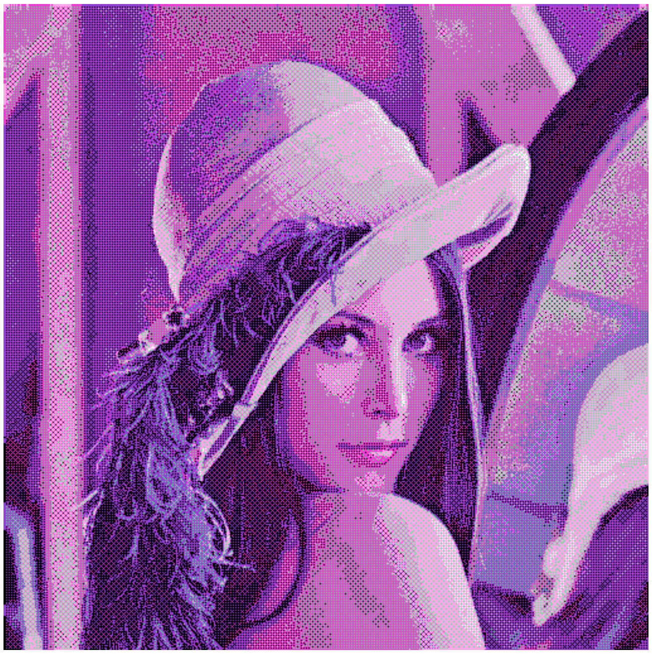

Figure 4. Lena with random palette halftoned using multiresolution halftoning algorithm.

Figure 5. Lena with random palette halftoned using Floyd-Steinberg error diffusion algorithm.

Figure 6. Lena quantized to eight colors using Heckbert's median cut algorithm.

Figure 7. Lena halftoned using multiresolution halftoning algorithm.

Figure 8. Lena halftoned using Floyd-Steinberg error-diffusion algorithm.