

# Multiresolution, error-convergence halftone algorithm

Eli Peli

*Physiological Optics Unit, Eye Research Institute, 20 Staniford Street, Boston,  
Massachusetts 02114*

Received August 29, 1990; accepted November 14, 1990

A new halftone algorithm is described. The algorithm is designed for implementation on a parallel architecture in order to provide fast, progressive coding of moderate-resolution images. The design is based on a multiresolution, hierarchical, pyramidal structure. At each pyramid level, the binarized image is compared with the original, gray-tone image over a successively larger window of pixels for calculation of a weighted averaged error. Within each level, selected binarized pixels are tested for possible changes in the binary assignment. The binary assignment is changed if the change results in a lower average error over the entire window. Varying the selection of test pixels can cause the same process to provide clustered-dot patterns and dithering. A comparison of performance with the best implementation of the error-propagation algorithm is presented visually. Quality is compared also in terms of isotropy of the texture and the appropriate blue-noise characteristics in areas of uniform gray tone. The benefits of this algorithm are realized with moderate-resolution display of the order of 512 dots  $\times$  512 dots. The processing can be carried out on smaller blocks since the results can be combined without any visible seams or edge effects.

## INTRODUCTION

Digital-halftoning techniques are used for the rendition of multivalued gray-tone, continuous-tone images on bilevel displays or printers. Halftone algorithms attempt to compromise between the requirements for spatial resolution and dynamic range<sup>1</sup> and at the same time maintain a visually pleasing image in terms of texture and perceived roughness.<sup>2</sup> These techniques also can be viewed as compression methods for information transmission over limited-capacity channels. Halftoning algorithms create a binary image whose average local density mimics that of the original continuous-tone images. With sufficient resolution of the final display device, good results may be obtained with practically all the known algorithms. However, with the commonly available printers and displays, many algorithms usually suffer from artifacts that result from either a limited dynamic range, i.e., the number of gray tones represented, or a limited spatial resolution in the image.

I chose to classify halftoning algorithms into three types: (1) dithering techniques including constrained-average or edge-emphasis techniques, (2) error-diffusion or propagation techniques, and (3) adaptive methods. Jarvis *et al.*<sup>3</sup> reviewed the first two types and compared their performances visually. Ulichney<sup>4</sup> recently described two measures for quantitative evaluation and comparison of halftoning techniques. The current paper describes an algorithm that attempts to incorporate the advantages of all three. The algorithm implemented performs better for low-to-moderate resolution display devices (of the order of 512 pixels  $\times$  512 pixels or dots). This algorithm is designed to be implemented with a parallel architecture and thus permits fast, possibly real-time, processing as well as progressive transmission in which the image quality is gradually improved.

Dither halftoning algorithms generate a binary image by comparing the original image value with a position-dependent set of thresholds within a small, repeated cell. Com-

monly, a square cell containing  $n$  pixels  $\times$   $n$  pixels is used,<sup>5</sup> but rectangular or hexagonal<sup>4</sup> cells also have been used. The choices of the threshold values and their position within the cell determines the nature of the dither technique. The position of the threshold values within a cell may be random or ordered, with order favored visually. The position of the dots within the ordered screen can be dispersed across the cell or clustered. Clustered-dot-ordered screens are preferred for printing devices in which the fidelity of single-dot production is low. For clustered-dot algorithms, diagonally oriented patterns are usually desired, presumably because of the reduced sensitivity of the human visual system to diagonal patterns.<sup>4</sup> For devices with good reliability of dot reproduction, dispersed-dot-ordered screens are selected because they provide increased spatial resolution at the same dynamic range of gray levels. The method of dispersing the dots proposed by Bayer<sup>6</sup> is considered the best. This technique alternates among horizontal, vertical, and diagonal points at each gray step. The main limitation of the Bayer dispersed-dot dithered halftoning is the distinct visibility of some of the textures, resulting in the appearance of false contours in some images.

Constrained-average methods, proposed independently by Roetling<sup>7</sup> and Jarvis and Roberts,<sup>8</sup> enhance the appearances of edges by calculating local average and then adjusting the dither threshold value locally. The local averaging is performed over both the binary images and the original in one case<sup>7</sup> and over the original image only in the other case.<sup>8</sup> The main advantage of these methods is the apparent increase in resolution resulting from the more effective lock to local gray-level changes and the consequent reduction in aliasing. Jarvis *et al.*<sup>3</sup> found that combining the dispersed-dither and the edge-emphasis techniques provided the best results for the images that they tested.

For the error-diffusion (propagation) algorithm,<sup>9</sup> there is no set, repeated cell of pixels. The algorithm progresses as a pixel-by-pixel, gray-scale control. When the gray-tone pixel is turned into white or black, a local error in the gray

scale is generated. To correct the picture, the error is redistributed among nearby pixels. The results are generally good in terms of the mutually increased dynamic range and good spatial resolution,<sup>1</sup> which suppresses aliasing. The good detail rendition of this algorithm has been shown to be due to an inherent edge enhancement that is built into the error-diffusion process.<sup>10</sup> Ulichney<sup>2</sup> evaluated a number of methods for implementing the error-propagation technique in search of a class most suitable for representing an area of uniform gray level. He showed that the best results are obtained when the final image has blue-noise characteristics. This effect can be achieved by using a modification of the basic algorithm, applying the original filter of Floyd and Steinberg<sup>9</sup> with 50% random weights processed on a serpentine raster. This process reduces the disturbing texture patterns that appear frequently when the original error-propagation technique is used.

These algorithms are accurately called error propagations because they are inherently serial and because the error is being determined based only on pixels already processed. When the processing is on a serpentine raster, the cumulative error determination is limited to at most half the neighborhood of any point. The computational complexity of the method increases with the area over which the error is permitted to propagate, and the inherent serial operation results in excessive processing time for moderate-sized images. These two problems are addressed by the algorithm proposed here. The algorithm is designed as a recursion of operations. At each recursion step, the parallel operations are computationally simple and can be performed with available hardware. The proposed algorithm operates as a symmetrical error-diffusion algorithm.

This multiresolution algorithm is adaptive in nature as it assigns more gray levels per presentation to image areas with low-frequency spatial content, while restricting the number of gray levels representing areas of high-spatial-frequency content. This approach, which is implicit in this algorithm, was explicitly used for improving halftone coding by Carlson and Besslich.<sup>11</sup> The difference in the number of the gray levels that can be seen at any spatial frequency and the need for locally varying the resolution and dynamic range in halftone images was analyzed previously by Roetling.<sup>12,13</sup>

The image is progressively coded into a binary output by using a pyramidal structure that is based on the errors generated by the binary assignment at each pixel. In the first level of the pyramid, all pixels are binarized. At each successive pyramidal level, some binary pixels are modified in order to reduce error averaged over larger areas. The region of support is symmetrical around the tested point and is growing at each successive step up the pyramid. This iterative buildup, starting with fine details and progressively modifying pixels in order to reduce gray-scale errors, is the major point of this paper. The term error convergence is used because the binary assignment at a point is based on the error generated at all neighboring pixels. The use of less than 50% overlapping neighborhoods permits parallel processing. Two modifications of the basic algorithm are presented. The first modification provides a clustered-dot version to enhance printing reproducibility. The second one incorporates position dither to reduce contouring and anisotropy of the texture and to furnish the desirable blue-noise characteristics. The results are evaluated both by the



(a)



(b)

visual inspection of images halftoned with various algorithms and by the use of Ulichney's measures<sup>4</sup> of spatial spectrum and isotropy.

## THE PYRAMID

At the first pyramid level, the original gray-tone image  $I(i, j)$  is thresholded pixel by pixel to produce a binary image  $B(i, j)$  [Fig. 1(a)]:

$$B(i, j) = \begin{cases} 255 & [I(i, j) > 127] \\ 0 & [I(i, j) \leq 127] \end{cases} \quad (1)$$



(c)



(e)



(d)



(f)

Fig. 1. Progressive development of halftone images at the different levels of the pyramidal error-convergence algorithm. Printing was limited to 150 dots per inch to permit clear visibility of the patterns. (a) The binary image resulting from simple thresholding at the first level of the pyramid. (b) The image at the end of the second pyramidal level for which errors were calculated and corrected over  $3 \times 3$  pixels windows. (c) The result following the third level correcting for errors over  $5 \times 5$  windows. (d) The output at the end of the fourth level of the pyramid correcting for  $7 \times 7$  windows. (e) The final output following the fifth level of the pyramid ( $9 \times 9$ ). A few contours are visible despite the availability of 32 gray levels. (f) The same image processed with the dispersed-dot dithered algorithm of Bayer.<sup>6</sup> Note the similarity of textures and the lower spatial resolution in this image as compared with (e), especially at the feathers.

The binary image then is compared with the original image to produce an error array  $E(i, j)$ :

$$E(i, j) = B(i, j) - I(i, j). \quad (2)$$

The sign of the error  $E(i, j)$  thus represents the binary image  $B(i, j)$  except for the case of  $E(i, j) = 0$ , which is ambiguous.

At the second level, the signed error array  $E(i, j)$  is convolved with a  $3 \times 3$  mask  $y^{(3)}$  [Eq. (3b) below]. The resultant output  $\bar{E}^{(3)}(i, j)$  is a local, weighted sum of the error at the first level:

$$\bar{E}^{(3)}(i, j) = E(i, j) * y^{(3)}, \quad (3a)$$

3		3		3		3		3		3	
	5		7		5		7		5		7
3		3		3		3		3		3	
			5		9		5				5
3		3		3		3		3		3	
	5		7		5		7		5		7
3		3		3		3		3		3	
	9		5				5		9		5
3		3		3		3		3		3	
	5		7		5		7		5		7
3		3		3		3		3		3	
			5		9		5				5
3		3		3		3		3		3	
	5		7		5		7		5		7
3		3		3		3		3		3	
	9		5				5		9		5
3		3		3		3		3		3	
	5		7		5		7		5		7

Fig. 2. Pixel positions tested for possible change of the binary assignment at all levels of the pyramid. Pixels marked by 3 are tested and changed at the second (3 × 3) level of the pyramid, etc. The tested pixels positions were selected for providing alternating diagonal horizontal-vertical patterns.

where

$$y^{(3)} = \begin{bmatrix} 1/12 & 1/12 & 1/12 \\ 1/12 & 1/3 & 1/12 \\ 1/12 & 1/12 & 1/12 \end{bmatrix}. \tag{3b}$$

The weights of the terms in  $y^{(3)}$  add to unity.

When a change in a pixel's code (i.e., a change from white to black or vice versa) decreases the weighted average error, the output code at the current level of the pyramid is changed. Every other pixel on every other line is examined for a possible code change. This partial selection provides for independent decision making.

This independence permits completely parallel operations because a pixel's being changed will not affect the averaged error in any other neighboring point that is being considered in the image. The pixels of an image that are examined and changed at the 3 × 3 level are marked in Fig. 2 by the number 3. The output code at this level of the pyramid is illustrated in Fig. 1(b).

For every change of a pixel from black to white or vice versa, the error array is updated according to

$$E(i, j) = \begin{cases} E(i, j) - 255 & \text{(white} \rightarrow \text{black)} \\ E(i, j) + 255 & \text{(black} \rightarrow \text{white)} \end{cases}. \tag{4}$$

Thus the original error array is modified wherever a decision is made to change pixel polarity. The modified error array then is used for calculating the next pyramid level in which 5 × 5 windows are applied.

The 5 × 5 level of the pyramid is processed in a similar way to the 3 × 3 level. The weighted average error array is obtained by convolving the updated error array with a 5 × 5 mask:

$$\bar{E}^{(5)}(i, j) = E(i, j) * x^{(5)}. \tag{5}$$

The actual calculation of the average is accomplished by two successive convolutions of the updated error array  $E(i, j)$  with a uniform 3 × 3 averaging matrix  $x^{(3)}$  (all terms equal 1/9). The use of repeated convolutions permits implementation of the algorithm with hardware that is capable of only 3 × 3 convolutions:

$$\bar{E}^{(5)}(i, j) = E(i, j) * x^{(3)} * x^{(3)}. \tag{6}$$

The result of these two convolutions is equivalent to one convolution with a 5 × 5, matrix  $x^{(5)}$ :

$$x^{(5)} = \begin{bmatrix} 1/81 & 2/81 & 3/81 & 2/81 & 1/81 \\ 2/81 & 4/81 & 6/81 & 4/81 & 2/81 \\ 3/81 & 6/81 & 1/9 & 6/81 & 3/81 \\ 2/81 & 4/81 & 6/81 & 4/81 & 2/81 \\ 1/81 & 2/81 & 3/81 & 2/81 & 1/81 \end{bmatrix}. \tag{7}$$

Once again, a subset of the pixels of the image is tested in order to determine whether a change in polarity (i.e., code) of the pixel from white to black or vice versa results in a decrease of the averaged error within the 5 × 5 window. Pixels that are tested at the 5 × 5 level also are shown in Fig. 2. These positions are arranged in a diagonal pattern. Thus, for uniform gray levels that require a change of only those pixels, the pattern will have the familiar, and visually more acceptable, diagonal screen pattern (Fig. 3).<sup>2</sup> Here, the constraint of obtaining a uniform, diagonal, nonoverlapping distribution contradicts our preference for sampling lines that have not been sampled in a 3 × 3 level. Thus Fig. 2

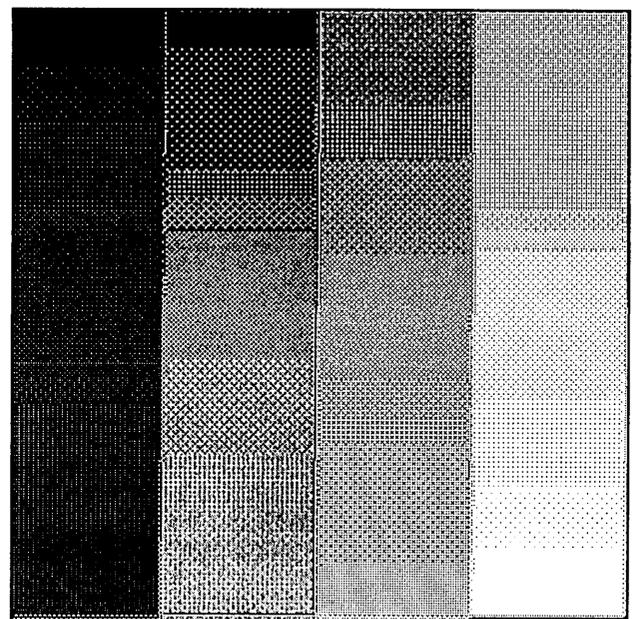


Fig. 3. Linear gray-tone wedge halftoned with the empirically determined values in Table 1. These values provide a more even distribution of values along the wedge (32 gray levels are distinguishable). Printing was limited to 150 dots per inch to permit clear visibility of texture patterns.

**Table 1. Error Thresholds Used to Determine Whether a Pixel Binary Assignment Should Be Changed**

Error Threshold	Pyramid Level			
	3 × 3	5 × 5	7 × 7	9 × 9
Calculated	43-86	14-28	9-18	7-14
Empirical	64	36	10	3

3	7	3		3	7	3		3	7	3		3
5				5				5				5
	3			3				3				3
9	5			9	5			9	5			9
3		3	7	3		3	7	3		3	7	3
5				5				5				5
	3		3		3		3		3		3	
	5	9	5		5	9	5		5	9	5	
3	7	3		3	7	3		3	7	3		3
5			5		5			5				5
	3		3		3		3		3		3	
9	5			9	5			9	5			9
3		3	7	3		3	7	3		3	7	3
5				5				5				5
	3		3		3		3		3		3	
	5	9	5		5	9	5		5	9	5	
3	7	3		3	7	3		3	7	3		3
5			5		5			5				5

Fig. 4. Pixel position tested for the clustered-dot version (compare with Fig. 2). The shaded areas illustrate windows over which error was calculated for the 3 × 3 level (top left) and the 5 × 5 level (bottom).

illustrates a compromise with an anisotropic sampling for this level. This selection results in some window overlaps, which are limited to small weights. However, the resulting small errors will not propagate further because they will be reconsidered and corrected in the following level of 7 × 7 windows. The patterns generated are similar to those patterns generated by the dispersed-dot ordered-dither algorithm of Bayer.<sup>6</sup>

The 7 × 7 window pyramid level starts with the calculation of the 7 × 7 weighted average  $\bar{E}^{(7)}(i, j)$  for each pixel from the modified error array that was updated at the 5 × 5 level. Here again, the actual calculation of the 7 × 7 window is obtained by three successive applications of a 3 × 3 uniform window that result in an equivalent averaging, as would be obtained by convolving the array with the 7 × 7 matrix  $\mathbf{x}^{(7)}$ :

$$\bar{E}^{(7)} = E(i, j) * \mathbf{x}^{(7)} = E(i, j) * \mathbf{x}^{(3)} * \mathbf{x}^{(3)} * \mathbf{x}^{(3)}. \quad (8)$$

The positions of the points tested for change are illustrated in Fig. 2. Those points also are arranged in a horizontal-vertical pattern, interleaving points tested at the 3 × 3 level

diagonally and points tested at the 5 × 5 level of the pyramid horizontally and vertically. The intermediate image at this level is illustrated in Fig. 1(d).

The 9 × 9 level of the pyramid is last. Operations performed there are similar to those for the previous layers. The corresponding matrix may be calculated in the same way by convolving the  $\mathbf{x}^{(3)}$  uniform matrix with itself four times. The positions tested are shown in Fig. 2. The final image [Fig. 1(e)] is compared with the same image generated by the dispersed-dot ordered-dither algorithm of Bayer<sup>6</sup> [Fig. 1(f)].

I have applied the next level that utilizes an 11 × 11

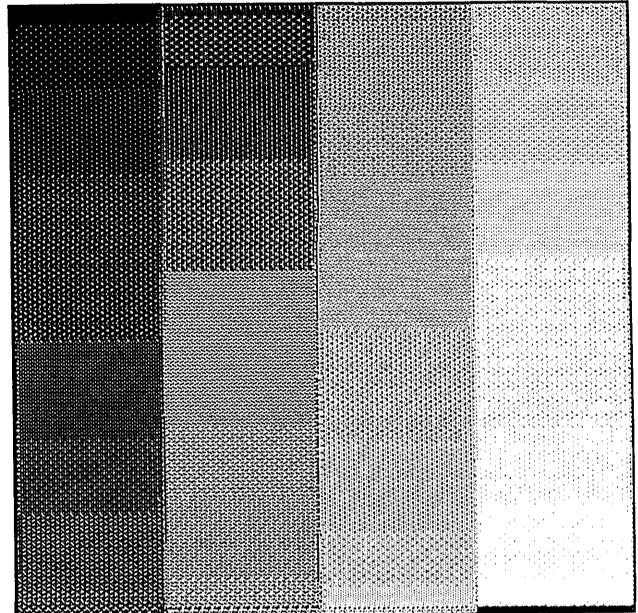


Fig. 5. Linear gray-tone wedge halftoned with the clustered dots by using thresholds of the same values as in Fig. 3. The tested positions are those illustrated in Fig. 4.

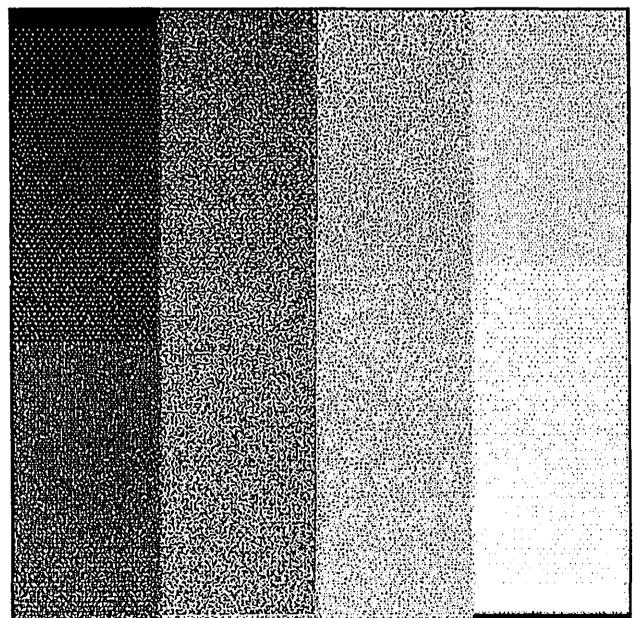
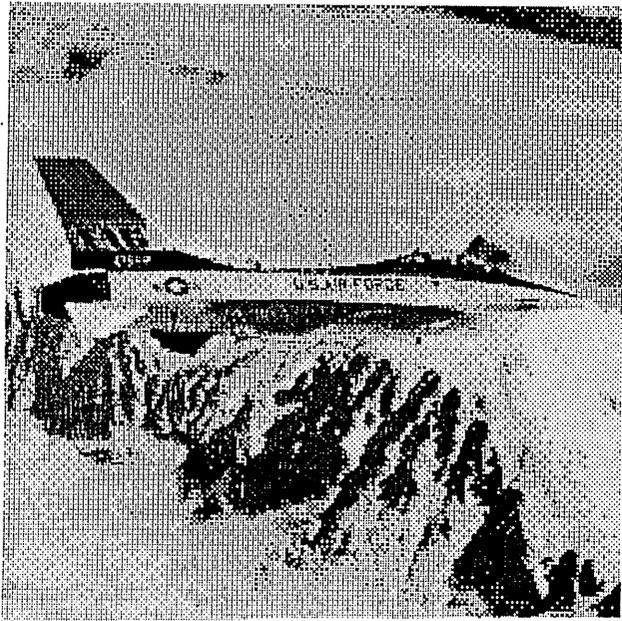
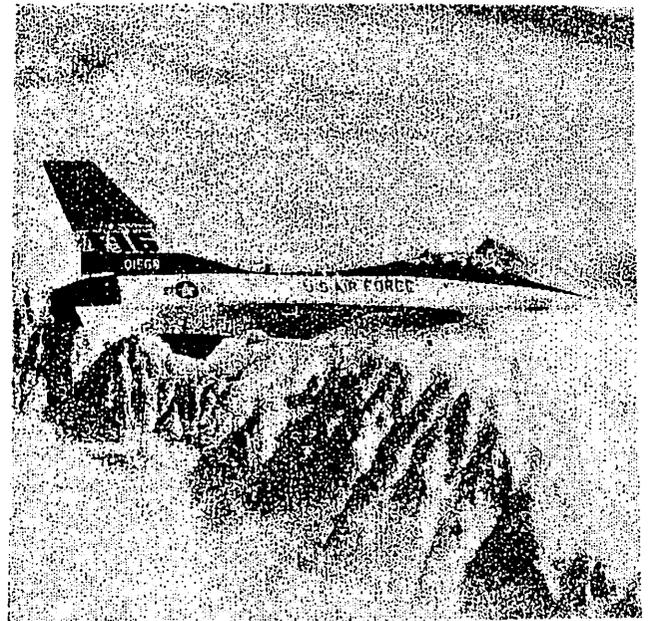


Fig. 6. Linear gray-tone wedge halftoned with the position-dithered algorithm by using the same parameters as Fig. 4.



(a)



(c)



(b)

Fig. 7. Moderate resolution images (512 dots  $\times$  512 dots) processed with three halftoning techniques are presented for comparison. For comparison of spatial resolution, note the serial number under the F-16. (a) An image processed with the dispersed-dot algorithm of Bayer<sup>6</sup> (33 gray levels). (b) The same image processed with the error-propagation algorithm. (c) The same image processed with the pyramidal error-convergence algorithm. Although the textures are slightly less pleasing than those obtained with error propagation, the spatial resolution is superior, and fast implementation is possible.

window; however, the results appear to be visually inconsequential. Therefore in these experiments I have limited the pyramid to five levels (i.e., up to a  $9 \times 9$  window).

#### Averaged-Error Threshold Determination

The prior binary assignment of a point should be changed only if the change reduces the averaged error at the scale associated with the current level of the pyramid. Changing the assignment of a point always changes the sign of the error,  $E(i, j)$ , at that point. Therefore we make such a change only if

$$\text{sign}[E(i, j)] = \text{sign}[\bar{E}^{(n)}(i, j)], \quad (9)$$

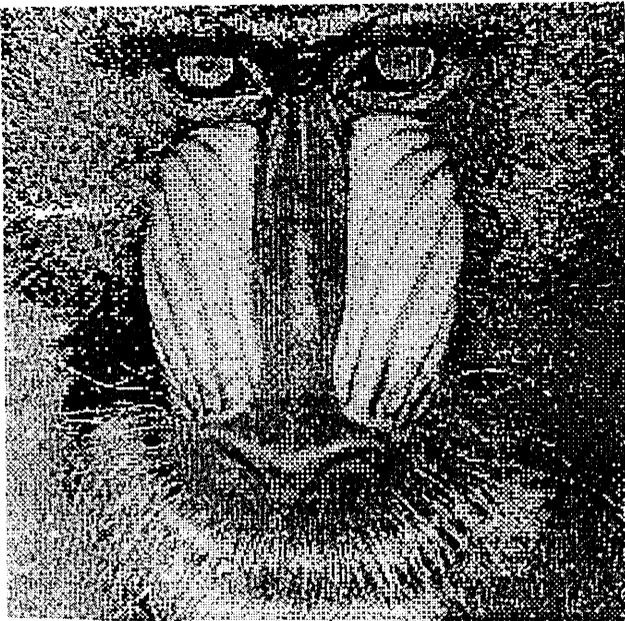
where  $n$  corresponds to the level of the pyramid.

If the signs are not equal, a change in the assignment actually will cause an increase in the local averaged error. In the calculation of the averaged-error threshold for each level of the pyramid, we assume that the weight assigned in the averaged-error calculation is the one perceived by the observer for the center point at that scale. For example, at the  $5 \times 5$  level of the pyramid the center point is weighted as  $1/9$  [Eq. (7)]. Therefore a change in the binary assignment will cause a  $255/9 = 28$  change in the averaged error at that level. Thus pixels with averaged errors larger than 28 will benefit from a change of the binary assignment. If we permit a reduction of absolute value of the error, including cases for which such a change will result in a change of sign of the averaged error, then pixels with an averaged error larger

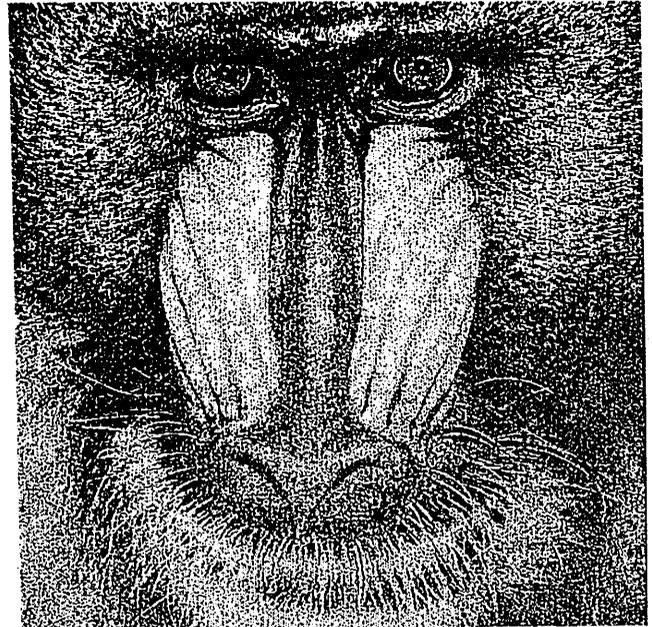
than 14 also could be changed. I found that, by using either of the two criteria, I did not obtain an equal representation of the gray levels of a linear gray-tone wedge. In addition, the number of gray levels represented was smaller than anticipated. These effects are the results of arbitrary selection of weight for the  $y^{(3)}$  matrix and the nonideal placing of the points tested at the  $5 \times 5$  level. Therefore the actual threshold was determined for each level of the pyramid empirically by finding those values that split every previous section of the gray-scale wedge into two almost equal selections (Fig. 3). The calculated threshold values and the empirically determined levels used are given in Table 1.

#### Clustered-Dot Version

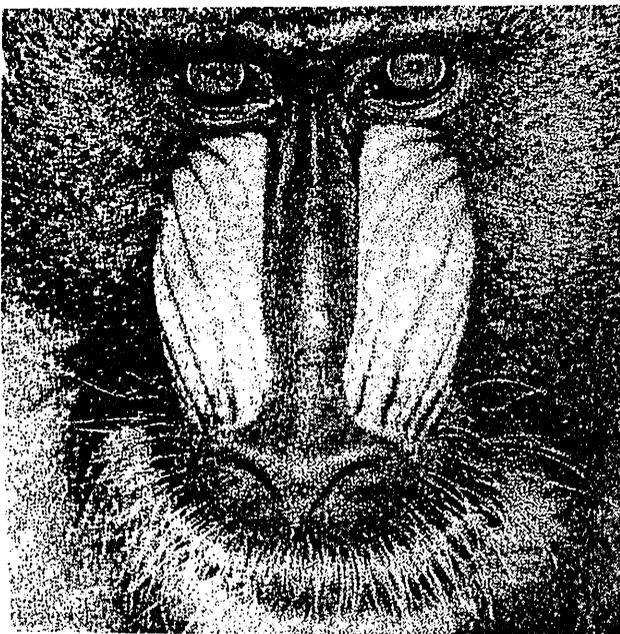
The selection of points tested on the pyramid as depicted in Fig. 2 results in dispersed-dot patterns [Figs. 1(e) and 3] that frequently do not reproduce well in print. Therefore clustered-dot patterns have evolved to improve the quality of reproduction by various laser printers. The unreliable reproduction of a single dot is the reason that I have reproduced most of the images in this paper at 150-dots-per-inch resolution, thus creating a 4-dot cluster to represent each dot in the final image. The pyramidal error-convergence algorithm may be modified in order to accommodate more clustered-dot patterns and improve reproduction in print. The



(a)



(c)



(b)

Fig. 8. Same sequence as in Fig. 7 for the baboon's image. These images illustrate the superiority of the pyramidal error-convergence algorithm in both resolution and texture over the two other techniques.



(a)



(c)



(b)

Fig. 9. Same as Fig. 6 for a face image for which the textures used to render the skin are critical. (a) The simulated optical screen, clustered-dot dither was implemented because it provides smooth skin texture, but the spatial resolution is poor. The dispersed-dot result for the same image is illustrated in Fig. 1(f). The same image halftoned with the error-propagation algorithm. (c) The pyramidal error convergence provides acceptable skin texture and high spatial resolution.

only change required for this modification is a different choice of the points tested at each pyramidal level. One such pattern is illustrated in Fig. 4. This pattern differs from the original (Fig. 2) beginning with the  $5 \times 5$  level of the pyramid. Points at this level, and in the following levels, were always placed adjacent to points already tested at previous pyramidal levels. This design of adjacency provides clustered rather than dispersed dots when possible. The resulting pattern (Fig. 5) represents a compromise between the dispersed-dot and clustered-dot patterns, with which an increased clustering of the dots occurs toward intermediate levels of gray tones.

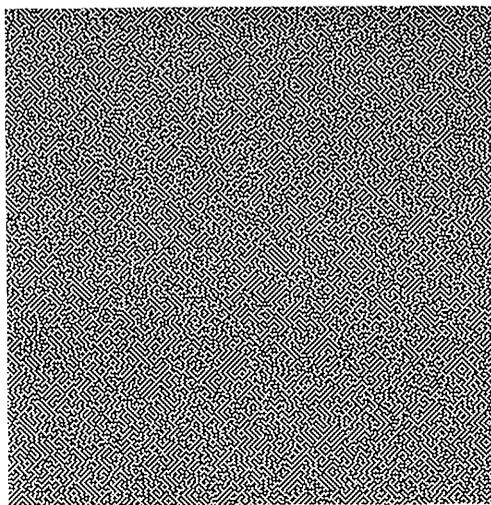
Ink-spread characteristics and the difference between the sizes of a printed dot and a nonprinted dot may result in printed gray levels not linearly related to the number of black dots.<sup>14</sup> Thus an optimal, accurate assignment of error thresholds may be printer and paper dependent and can be obtained only by tedious calibration. However, the empirical values given in Table 1 provide a good first approximation from which such an assignment could be determined.

#### Dithering

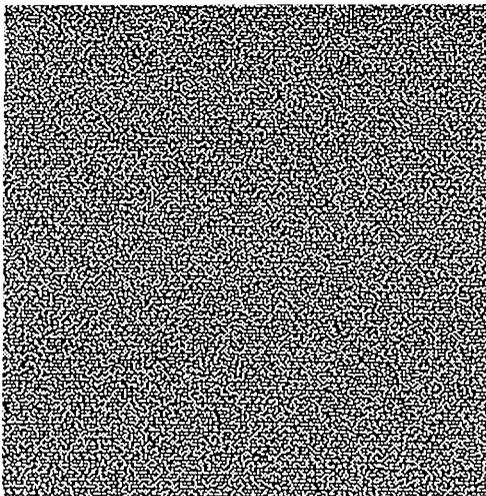
The algorithm described above is successful in combining high resolution with large dynamic range and providing

mostly diagonal textures. These textures for the dispersed-dot version are almost identical to those generated by the dispersed-dot dither of Bayer,<sup>6</sup> yet the spatial resolution is higher [Fig. 1(e) versus Fig. 1(f)]. This improvement in resolution compared with that produced by the algorithm that provides the best spatial resolution among the dither algorithms is the main benefit of this version of the pyramidal error-convergence algorithm. However, despite the large number of gray levels generated (32), there are still visible false contour lines because of the visibility of texture changes in areas of a shallow gradient of gray levels [Fig. 1(e)]. These artifacts, similar to the dither artifacts of Bayer,<sup>6</sup> are quite disturbing in halftone imaging. The texture within areas of uniform gray level is also important. Ulich-

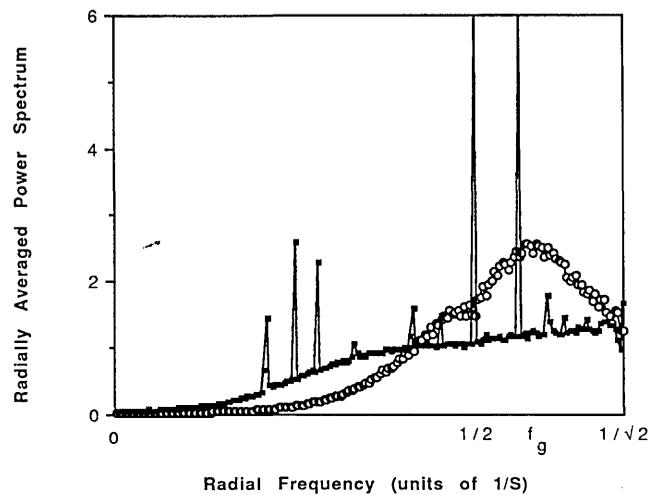
ney<sup>4</sup> demonstrated that halftone patterns of uniform gray levels that have blue-noise characteristics and reduced anisotropy are more acceptable and visually pleasing. To obtain similar blue-noise and isotropy characteristics and reduce false contouring, I included a pseudorandom noise in the choice of points tested at each pyramidal level. Rather than testing points on the orderly grids described in Fig. 2 or Fig. 4, I pseudorandomly shifted the testing to one of the eight neighboring pixels in approximately half the pixels. The decision as to which pixel will be tested and modified was established by a random variable testing. This random selection does not have to be calculated for each image; it may be selected once and then can be fixed and used for every image. In this implementation, horizontal and verti-



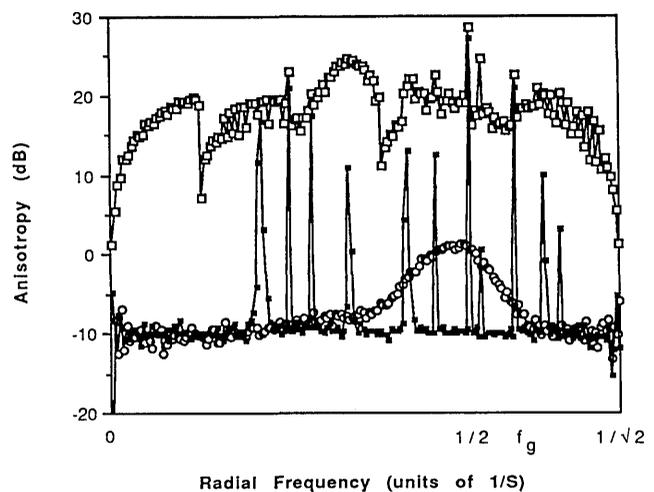
(a)



(b)

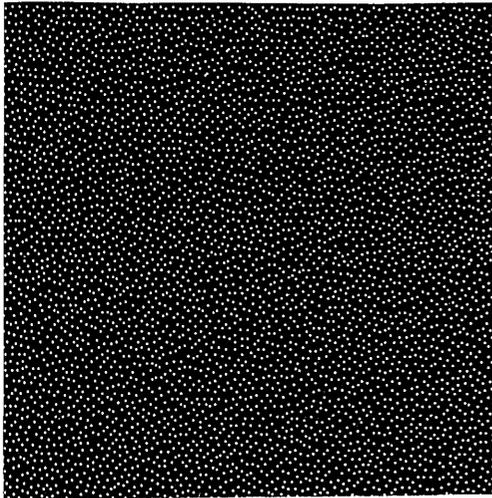


(c)

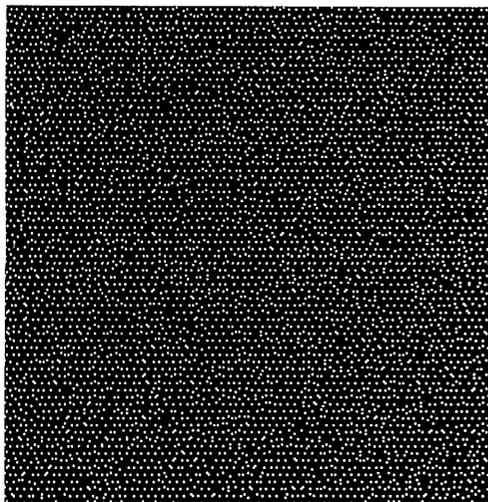


(d)

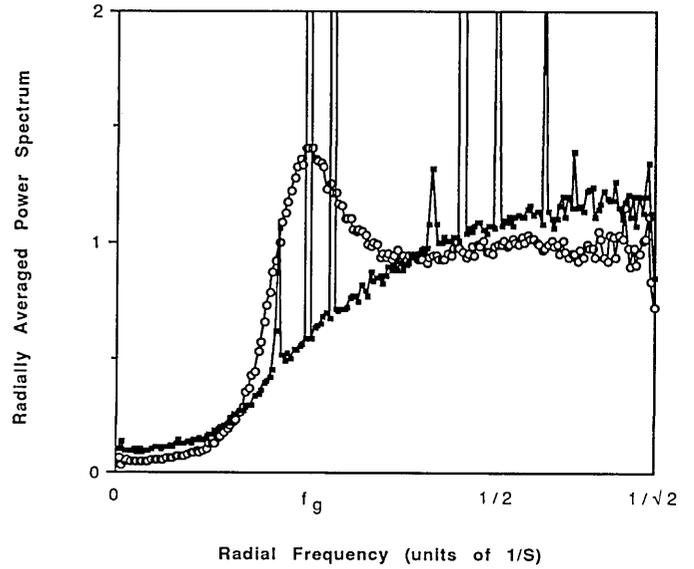
Fig. 10. Characteristics of sample images of uniform gray level 85 ( $g = 1/3$ ). In this case the pyramidal error-convergence algorithm performs better than the error propagation. (a) A sample of the error-propagation image. (b) A sample of the pyramidal error-convergence image with random position dithering. Little directional texture may be noted. (c) Radial spectra obtained from ten samples for each of the images illustrated in (a) and (b). (d) The anisotropy calculated for the pyramidal error convergence (randomized and nonrandomized) compared with the anisotropy of the error-propagation sample: circles, error propagation; open squares, nonrandomized; solid squares, randomized. Flat anisotropy of  $-10$  dB is the optimal result. The random pyramidal case is at that level except for the few spikes. The error-propagation anisotropy is increased at the principal frequency to  $0$  dB. The nonrandom case is presented to illustrate the effect of randomization in reducing the anisotropy by  $25$  dB or more across most frequencies.



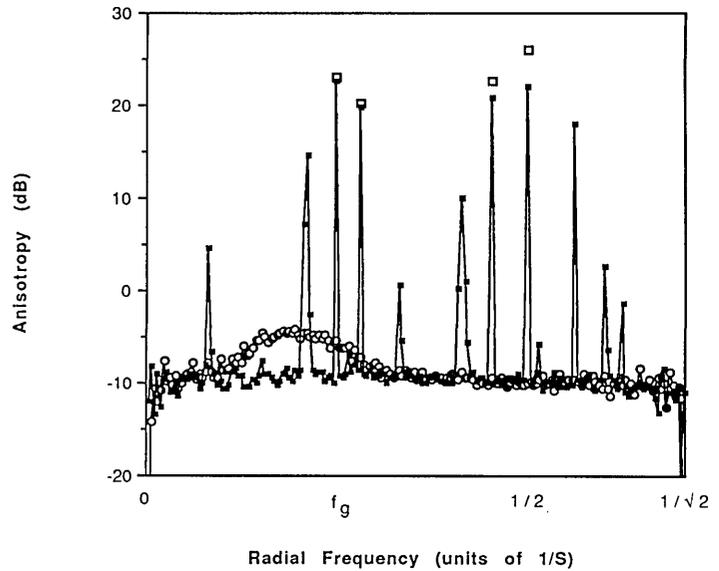
(a)



(b)



(c)



(d)

Fig. 11. Similar to Fig. 8 except for gray level 16 ( $g = 1/16$ ). In this case the error-propagation algorithm is superior in terms of the anisotropy. The spectral characteristics are similar and close to optimal in both cases.

cal shifts were determined separately and independently. The probability of shift of a tested, central point to any of the neighboring positions was designated by the matrix  $P$ :

$$P = \begin{bmatrix} 1/60 & 2/15 & 1/60 \\ 1/15 & 8/15 & 1/15 \\ 1/60 & 2/15 & 1/60 \end{bmatrix}. \quad (10)$$

**RESULTS**

The halftoned wedge obtained with the dithering of the clustered version of the pyramidal error convergence algorithm is presented in Fig. 6. The results of dithering the dispersed-dot version are practically indistinguishable; therefore only one case is illustrated in the following images.

Only a few false contours remain visible, at the darkest and at the brightest ends of the wedge. Halftoned images obtained with three different methods from  $512 \times 512$  gray-tone images with 256 gray levels were compared visually. Three images are presented to the reader for comparison (Figs. 7-9). Images were selected from a widely used commercial package of images distributed by the University of Southern California.<sup>15</sup> Images were halftoned at the spatial resolution of 1 dot per pixel and were printed at a density of 150 dots per inch to permit accurate reproduction and clear visibility of the dots' patterns. Images processed with the pyramidal error-convergence algorithm were compared with the same images processed with both the ordered-dither, dispersed-dot algorithm of Bayer<sup>6</sup> and modified-error propagation.<sup>4</sup> The first algorithm can be implemented quickly

and provides regular textures in uniform shading areas as well as high spatial resolution. However, false contouring is frequently evident in areas of shallow gradients [Figs. 1(f) and 7(a)]. Images processed with the error-propagation algorithm have good spatial resolution and blue-noise texture. The results obtained with this algorithm are the best demonstrated so far; however, fast implementation of this algorithm is limited by its inherent serial nature. As seen in Figs. 7–9, the pyramidal error-convergence algorithm provides spatial resolution superior to both the other algorithms with similar numbers of gray levels. The random nature of the dithered pattern also serves well to blend previously visible false contours [compare Figs. 1(e) and 9(c)]. Thus this algorithm also provides smooth texture transitions and is comparable with the performance of the error-propagation algorithm. Improvement of high-frequency details may be achieved with the other algorithms by preprocessing the gray-tone image.<sup>5</sup> However, such preprocessing increases the computational cost of other methods compared with that of the error-convergence algorithm.

To determine the effect of the position noise (dithering) on the spatial spectral properties of the textures of uniform gray levels images qualitatively, I used the two measures suggested by Ulichney,<sup>2,4</sup> the radial average frequency spectrum and anisotropy. Examples of the analysis for two cases are illustrated in Figs. 10 and 11. I compared these measures calculated for the best implementation of error propagation with the results obtained from this error-convergence algorithm. The error propagation was implemented with 50% random weights, using the best method recommended for a rectangular grid by Ulichney.<sup>4</sup> All images were printed at 107 dots per inch to permit clear visibility of the textures. The error-convergence spectrum contains strong peaks at few frequencies, but both spectra have the preferred characteristics of low-frequency cutoff at the principal frequency  $f_g$ , sharp transition, and relatively flat blue-noise area. Although the spectra of our dithered patterns contained spikes at frequencies corresponding to the underlying ordered patterns (Figs. 10 and 11), I believe that the general characteristics of our spectra were comparable with those of the ideal patterns described by Ulichney.<sup>4</sup> The spectra have the same blue-noise high-pass nature with the transition occurring near the principal frequency. The anisotropy is at the ideal  $-10$ -dB level, except for a few peaks at the underlying frequencies. The anisotropy represented by those peaks may be detectable with careful observation but is not immediately apparent.

The algorithm described was designed to be implemented on a parallel system that can perform the  $3 \times 3$  convolution for the entire  $512 \times 512$  image at once. Such systems are readily available today. Relatively inexpensive systems can perform the convolution at 1/30 sec. The algorithm requires repeated application of this  $3 \times 3$  convolution 1 time for the  $3 \times 3$  level, 2 times for the  $5 \times 5$ , 3 times for the  $7 \times 7$ , and 4 times for the  $9 \times 9$ , for a total of 10 applications of the  $3 \times 3$  convolution operation. This number determines the time needed for calculating a  $512 \times 512$ -square image. For images larger than 512, the images can be sectioned into smaller overlapping parts, processed, and then pasted together after removal of the overlapped edges. The overlapping areas are needed for removing the edge artifacts from all sections. The random method introduced by the posi-

tion noise dithering resulted in an invisible seam on all images that we have processed this way.

## CONCLUSIONS

Halftone algorithms used for depicting continuous-tone images on a binary display have gained considerable attention over the past two decades. These techniques are used mostly in computer preparation of images for printing. Although the introduction of low-cost, extremely high-resolution laser printers may substantially decrease the difference in performance among different algorithms, current laser-printer resolution (300 dots per inch) is still low enough that improved algorithms produce superior images. In addition to their uses in printing, halftone images are used for displaying images on computer binary monitors and a variety of other bilevel displays.<sup>3</sup> Recent introduction of a moderate-resolution binary miniaturized display, the Private Eye,<sup>16</sup> and the increased interest in visual image communication over channels of limited capacity (picture phone and fax) also may contribute to the continued interest in improved and fast halftoning techniques. Halftone images also are considered to be a compressed form suitable for efficient transmission.<sup>12,17</sup> The algorithm presented here is especially fitting for such applications because of its progressive coding property and its design for parallel, fast processing with currently available, relatively inexpensive hardware.

Anastassiou and Kollias<sup>17</sup> introduced a symmetrical error-diffusion neural network in which the errors from all neighboring points participate in determining the binary value at a central point. The process is parallel and recursive in nature and thus can be rapidly completed with neural-net (systolic-array) hardware. Further increases in speed were proposed in another paper by Anastassiou and Kollias, who used a progressive coding scheme of halftone images.<sup>18</sup> In this scheme each image is broken into different classes of image at different resolutions, which are progressively coded or binarized. The process of binarization may be performed in parallel within each class of image, based on values of previously calculated classes. Thus the process combines the ordered-dither and error-propagation approaches, as each position threshold is determined from errors propagated from the previously processed neighboring pixels.

The present method is similar in some ways to that of Anastassiou and Kollias<sup>17</sup> but differs both conceptually and in practical implementation. The most important difference is the direction of progress on the pyramid of scales. Anastassiou and Kollias<sup>17</sup> start the processing at low resolution and progress to higher and higher resolution, while I start at the tip of the pyramid with the highest resolution and progress downward, adding gray level at places where the spatial image content (low frequency) requires such changes.

The multiresolution, error-convergence algorithm was designed for incorporating the advantages of all the previous classes of halftone technique. It provides superior spatial resolution of the edge-emphasis techniques combined with dynamic range similar to that of Bayer's dispersed-dot dither algorithm.<sup>6</sup> The blue-noise characteristics of the textures generated are comparable with those of the best implementation of the slow but visually superior error-propagation technique. In addition, the technique is adaptive in nature

and provides for the rendition of more gray levels where the local spatial frequencies are low. These advantages are important for the rendition of images at moderate resolution of  $512 \times 512$  dots or less. When the display device used and the desired image size permit much higher resolution, comparable quality may be obtained with the simple ordered-dither algorithm.

## ACKNOWLEDGMENTS

The author thanks S. Lubars of the Eye Research Institute, Boston, for valuable programming help. This research was supported in part by grant EYR015957 from the National Institutes of Health.

The author is also with Tufts University School of Medicine, Tufts University, Medford, Massachusetts.

## REFERENCES

1. J. A. Saghri, H. S. Hou, and A. G. Tescher, "Personal computer based image processing with halftoning," *Opt. Eng.* **25**, 499-504 (1986).
2. R. A. Ulichney, "Dithering with blue noise," *Proc. IEEE* **76**, 56-79 (1988).
3. J. F. Jarvis, C. N. Judice, and W. H. Ninke, "A survey of techniques for the display of continuous tone pictures on bilevel displays," *Computer Graphics Image Process.* **5**, 13-40 (1976).
4. R. Ulichney, *Digital Halftoning* (MIT Press, Cambridge, Mass., 1987).
5. H. S. Hou, *Digital Halftoning and Shading in Digital Document Processing* (Wiley, New York, 1983), pp. 83-115.
6. B. E. Bayer, "An optimum method for two level rendition of continuous-tone pictures," in *Proceedings of the IEEE International Communications Conference Record* (Institute of Electrical and Electronics Engineers, New York, 1973), pp. 26-11-26-15.
7. P. G. Roetling, "Halftone method with edge enhancement and moiré suppression," *J. Opt. Soc. Am.* **66**, 985-989 (1976).
8. J. F. Jarvis and C. S. Roberts, "A new technique for displaying continuous tone images on a bilevel display," *IEEE Trans. Commun.* **891-898** (1976).
9. R. W. Floyd and L. Steinberg, "Adaptive algorithm for spatial grey scale," *Proc. Soc. Inf. Disp.* **17**, 75-77 (1976).
10. K. T. Knox, "Edge enhancement in error diffusion," in *Paper Summaries of the SPSE's 42nd Annual Conference* (Society for Imaging Science and Technology, Springfield, Va., 1989), pp. 310-313.
11. M. F. Carlson and P. W. Besslich, "Adaptive selection of threshold matrix size for pseudo-gray rendition of images," *Opt. Eng.* **24**, 655-662 (1985).
12. P. G. Roetling, "Quality measures in digital halftones," in *Topical Meeting on Applied Vision*, Vol. 16 of 1989 OSA Technical Digest Series (Optical Society of America, Washington, D.C., 1989), pp. 59-62.
13. P. G. Roetling, "Visual performance and image coding," in *Image Processing*, J. C. Urbach, ed., *Proc. Soc. Photo-Opt. Instrum. Eng.* **74**, 195-198 (1976).
14. R. L. Stevenson and G. R. Arce, "Binary display of hexagonally sampled continuous-tone images," *J. Opt. Soc. Am. A* **2**, 1009-1013 (1985).
15. A. G. Weber, "Image data base," USC SIPI Rep. 101 (Image Data Base Signal and Image Processing Institute, University of Southern California, Los Angeles, Calif., 1988).
16. E. Peli, "Visual issues in the use of head mounted display," in *Visual Communication and Image Processing IV*, W. A. Pearlman, ed., *Proc. Soc. Photo-Opt. Instrum. Eng.* **1199**, 1164-1176 (1989).
17. D. Anastassiou and S. Kollias, "Digital image halftoning using neural networks," in *Visual Communications and Image Processing III*, T. R. Hsing, ed., *Proc. Soc. Photo-Opt. Instrum. Eng.* **1001**, 1062-1068 (1988).
18. D. Anastassiou and S. Kollias, "Progressive half-toning of images," *Electron. Lett.* **24**, 489-490 (1988).