

# Halftoning via Direct Binary Search using a Hard Circular Dot Overlap Model

*Farhan A. Baqai, Christopher C. Taylor and Jan P. Allebach*  
*Electronic Imaging Systems Lab., School of Electrical and Computer Engineering*  
*Purdue University, W. Lafayette, Indiana*

## Abstract

The direct binary search algorithm is a powerful heuristic for generating high quality halftone images that account for the characteristics of both the output device and viewer. With a hard circular dot overlap model, it yields enhanced detail rendition, suppression of aliasing artifacts, and improved tonal gradation in shadow areas.

## 1. Introduction

With the advent of low cost printers, the number of images being printed has increased manifold. Due to this, there is an increasing demand for high quality printing. Real life images are continuous-tone. Only photographic film and thermally sensitive papers are capable of rendering continuous-tone directly. Almost all printers are binary or multilevel with a few levels. Digital halftoning provides a mechanism of rendering continuous-tone images on such devices.

In halftoning, it is assumed that the perceived gray level is proportional to the fraction of black dots in the pattern. Ideally, we want the printed dots to be square. However, real printer dots are more closely round than square. To ensure full coverage, the diameter of the dots are designed to be larger than the spacing between them. This causes the perceived gray level to be darker than expected. In addition, the interaction between overlapping regions of adjacent dots is not additive. This effect is commonly known as “dot overlap”. In this paper, we propose a model-based iterative technique for digital halftoning. This technique exploits the properties of both output device and viewer in generating high quality images for printing at lower printer resolutions (300 dots per inch).

Several model-based halftoning approaches have been proposed. Roetling and Holladay<sup>1</sup> used a printer model to modify the ordered dither thresholds to ensure linear gray scale. They, however, did not take into account the effect of dot-overlap on texture. Allebach<sup>2</sup> proposed a modification to ordered dither that accounts for overlapping dots. However, the effect of dot-overlap was reduced at the expense of spatial resolution. Stucki<sup>3</sup> suggested an extension of error diffusion that accounts for the distortion due to dot-overlap. Stucki’s algorithm is efficient in computation and is geared to incorporate a dot-overlap model in error diffusion. Dot models for correcting dot gain in error diffusion have also been pro-

posed by Stevenson and Arce<sup>4</sup> and by Pappas and Neuhoff.<sup>5</sup> Rosenberg<sup>6</sup> suggested that a tone correction transformation be derived from the printer model and applied to the image before halftoning. This approach compensates for monotonic printer distortions resulting in an image that has good tone rendition. Lin<sup>7,8</sup> proposed the use of a printer model and an adaptive filter to improve the tonal response of halftones generated by the void-and-cluster algorithm. She also suggested<sup>9</sup> using a dot-model in generating frequency modulated halftone screens to improve pattern uniformity and tonal response.

A search-based approach based on the direct binary search (DBS) heuristic has also been proposed by Analoui and Allebach.<sup>10</sup> Similar algorithms were proposed at the same time by Pappas and Neuhoff<sup>11</sup> and Mulligan and Ahumada.<sup>12</sup> DBS accounts for the characteristics of the printer and human visual system. It has been used for halftoning in variety of ways. For instance, Carrara et al<sup>13</sup> suggested a combination of DBS and pulse density modulation to improve the quality of halftones generated by DBS at extreme absorptances. Some preliminary results of DBS with a dot model were also given. Flohr et al<sup>14</sup> proposed DBS with a stochastic dot model and Allebach et al<sup>15</sup> applied DBS to halftoning of monochrome, color, and time-varying images. Lieberman and Allebach<sup>16</sup> suggest different search heuristics for achieving better quality. DBS generates halftones that are visually pleasing and have fewer artifacts. The reason it outperforms other techniques is that it is iterative and there is no causality constraint.

Motivated by the initial results in [13], we study DBS with dot-overlap in much more detail and show that this approach generates halftones that are not only tone corrected, but also have excellent detail rendition especially in the shadow areas. To make the algorithm computationally tractable, efficient error minimization with the help of look up tables is also addressed. The least-squares approach in [11] has similar qualities. However, the efficient implementation of the proposed approach is computationally less expensive than the least-squares approach, and has the potential to be used in desktop printing applications.

The remainder of the paper is organized as follows. The next section describes the human visual model. In Sec. 3, we discuss the printer model. In Sec. 4, efficient implementation of DBS with dot-overlap is addressed. The experimental results are presented in Sec. 5. Finally, our conclusions are given in Sec. 6.

## 2. Human Visual Models

Halftoning algorithms rely on the inability of the human viewer to resolve the spatial texture in the printed image. Nearly all models for the human visual system have a low-pass characteristic. To keep the model tractable, we use a linear model based on the spatial frequency dependence of the contrast sensitivity of the human viewer. It is based on the exponential function proposed by Näsänen<sup>17</sup>:

$$H(\Omega) = K(L)\exp[-\alpha(L)\Omega] \quad (1)$$

where  $L$  is the average luminance in  $cd/m^2$ ,  $\Omega$  is the magnitude of radial spatial frequency vector  $(u, v)^T$  in cycles per degree,  $K(L) = aL^b$ , and

$$\alpha(L) = \frac{1}{c \ln(L) + d} \quad (2)$$

with  $a = 131.6$ ,  $b = 0.3188$ ,  $c = 0.525$ , and  $d = 3.91$ . Figure 1 shows the plot of  $H(\Omega)$ . The spacing of the dots is given by

$$r = 2 \tan^{-1} \frac{1}{2Rd} \text{ radians} \approx \frac{180}{\pi R d} \text{ degrees} \quad (3)$$

where  $R$  is the printer resolution in dots per inch (dpi) and  $d$  is the viewing distance. The sampling frequency is  $\rho_s = 1/r$ . Sampling  $H(\Omega)$  gives us the discrete version  $H[k, l]$ . The point spread function of the human visual system  $h[m, n]$  is given by the inverse discrete-space Fourier transform of  $H[k, l]$ . In addition, the dot absorptance profile  $p[m, n]$  accounts for the characteristics of the output device. Combining the human visual system model and the dot absorptance profile we get the perceived dot absorptance profile. It is assumed that the extent of  $p[m, n]$  is much smaller than  $h[m, n]$ ; so

$$\tilde{p}[m, n] = p[m, n] * h[m, n] \approx h[m, n] \quad (4)$$

where  $*$  indicates convolution.

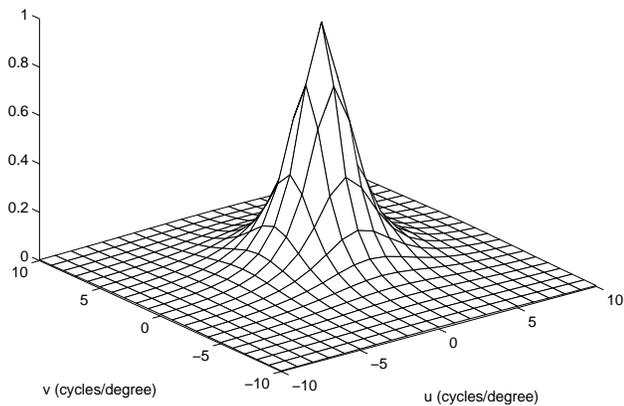


Figure 1. Human visual system model.

## 3. Printer Model

Since printed dots are more closely round than square, it is reasonable to assume that the printer produces equally spaced large circular dots on a Cartesian grid. The grid spacing is  $X$  inches. It is also assumed that the absorptance is uniformly maximum on the surface of the dot and minimum outside.

It should be pointed out that the printing mechanism determines the physical properties of the printed dot. For instance, in electrophotographic printing, the toner particles are transferred to the paper from the photoconductor and fused. Due to the statistical nature of the processes involved, the printer dot is not exactly round. Also, due to mechanical vibrations, paper slippage and gear noise, the scan line separation may vary significantly, i.e., the vertical spacing between dots may be significantly different from the expected value. Unfortunately, all the factors mentioned above are highly probabilistic and cannot easily be modeled. In the following, we develop a deterministic dot overlap model.

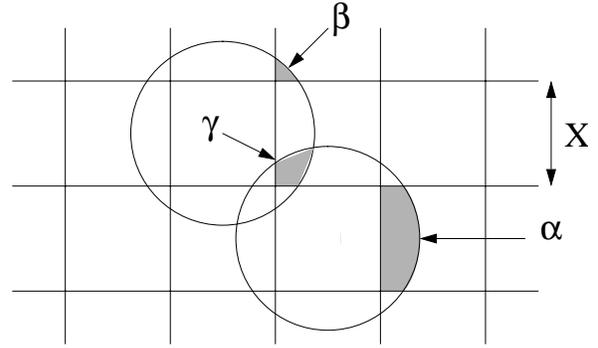


Figure 2. Hard circular dot overlap model.

Let be the input gray-scale digital image we wish to print and be the corresponding halftone. It is assumed that the number of pixels in the input image are equal to the number of dots in the halftoned image. If, a dot will be placed at  $(m, n)$  which is located  $mX$  inches from the left and  $nX$  inches from the top, and if no dot will be placed at this position. Since printed dots overlap, the gray level at any pixel is dependent on itself and its neighbors. We assume that it can be determined by pixels in a  $3 \times 3$  window. These assumptions are combined to form the hard circular dot overlap model.<sup>5,18</sup> In this model, dot-overlap is characterized by three parameters  $\alpha$ ,  $\beta$  and  $\gamma$ . These parameters are the ratios of the areas of the shaded regions shown in Figure 2 to the total area ( $X^2$ ). Equations describing  $\alpha$ ,  $\beta$ , and  $\gamma$  in terms (ratio of the actual dot radius to the ideal dot radius) are given in [5,18]. We account for dot overlap by replacing the digital halftone by the equivalent grayscale image. This is illustrated in Figure 4. So is given by

$$\hat{g}[m, n] = F(W[m, n]) = \begin{cases} 1, & g[m, n] = 1 \\ n_1\alpha + n_2\beta - n_3\gamma, & g[m, n] = 0 \end{cases} \quad (5)$$

where  $W[m, n]$  consists of and its eight neighbors,  $F$  is some function of  $W$ ,  $n_1$  is the number of horizontal and vertical neighboring dots that are black,  $n_2$  is the number of diagonally neighboring dots that are black and not adjacent to any horizontal and vertical neighboring black dot, and  $n_3$  is the number of pairs of neighboring black dots in which one is a horizontal neighbor and the other is a vertical neighbor.

Once  $F$  is evaluated for all possible ( $2^9 = 512$ ) values of  $W$ , we have a model describing the dependence of  $F$  on the eight neighbors of  $[m, n]$ . These values are stored in a look-up-table (LUT), which is useful for efficiently implementing the algorithm.

#### 4. Efficient Implementation of DBS with Dot Overlap Model

The goal of DBS is to determine the binary halftoned image  $g[m, n]$  such that the difference between the perceived versions of the original grayscale image and the rendered halftone is minimized. As a measure of the difference, we use the total squared error

$$\varepsilon = \sum_m \sum_n (\tilde{e}[m, n])^2 \quad (6)$$

where  $\tilde{e}[m, n] = \tilde{f}[m, n] - \tilde{g}[m, n]$  is the error between the perceived original image  $\tilde{f}[m, n]$  and the perceived halftone,  $\tilde{g}[m, n]$  where

$$\tilde{f}[m, n] = f[m, n] * \tilde{p}[m, n] \quad (7)$$

$$\tilde{g}[m, n] = \hat{g}[m, n] * \tilde{p}[m, n] \quad (8)$$

<b>a<sub>1</sub></b> <b>(m<sub>0</sub>-1, n<sub>0</sub>-1)</b>	<b>a<sub>4</sub></b> <b>(m<sub>0</sub>, n<sub>0</sub>-1)</b>	<b>a<sub>7</sub></b> <b>(m<sub>0</sub>+1, n<sub>0</sub>-1)</b>
<b>a<sub>2</sub></b> <b>(m<sub>0</sub>-1, n<sub>0</sub>)</b>	<b>a<sub>5</sub></b> <b>(m<sub>0</sub>, n<sub>0</sub>)</b>	<b>a<sub>8</sub></b> <b>(m<sub>0</sub>+1, n<sub>0</sub>)</b>
<b>a<sub>3</sub></b> <b>(m<sub>0</sub>-1, n<sub>0</sub>+1)</b>	<b>a<sub>6</sub></b> <b>(m<sub>0</sub>, n<sub>0</sub>+1)</b>	<b>a<sub>9</sub></b> <b>(m<sub>0</sub>+1, n<sub>0</sub>+1)</b>

Figure 3.  $3 \times 3$  region in the image.

To minimize the total squared error, the algorithm starts with an initial estimate of the halftone  $g[m, n]$ . This can be obtained efficiently by randomly thresholding  $f[m, n]$ . The halftone image is iteratively scanned,

and at each pixel location, we evaluate the effect of toggling its state or swapping its value with any of its 8 nearest neighbors. The toggle or swap that results in the greatest decrease in the squared error is accepted. An iteration is complete when every pixel in the image has been visited. When no changes are accepted during an iteration, the algorithm has converged to a local minimum of the error metric.

Let  $P \times P$  be the region of support for  $\tilde{p}[m, n]$ . The computation of a trial change in error due to a toggle using (6) requires  $(P + 1)^2$  additions to get  $\tilde{e}[m, n]$ , followed by  $9P^2$  multiplications and  $(P + 1)^2$  additions to get  $\varepsilon$ . However, updating  $\tilde{e}[m, n]$  requires only  $P^2$  additions. It is expected that the number of trial evaluations will be much greater than the number of updates. So it is desirable to minimize the computational cost of evaluating trial changes; even at the cost of increasing update complexity. In the following, we present a procedure for efficiently evaluating the effect of a trial toggle.

Consider minimizing the error for a trial toggle at location  $(m_0, n_0)$  in the image. The  $3 \times 3$  region containing the location  $(m_0, n_0)$  is shown in Figure 3. Any change in  $(m_0, n_0)$  would generally also alter its 8 neighbors. Therefore, the new perceived halftone can be written as

$$\begin{aligned} \tilde{g}'[m, n] = & \tilde{g}[m, n] + a_1 \tilde{p}[m - m_0 + 1, n - n_0 + 1] + \\ & a_2 \tilde{p}[m - m_0 + 1, n - n_0] + a_3 \tilde{p}[m - m_0 + 1, n - n_0 - 1] + \\ & a_4 \tilde{p}[m - m_0, n - n_0 + 1] + a_5 \tilde{p}[m - m_0, n - n_0] + \\ & a_6 \tilde{p}[m - m_0, n - n_0 - 1] + a_7 \tilde{p}[m - m_0 - 1, n - n_0 + 1] \\ & a_8 \tilde{p}[m - m_0 - 1, n - n_0] + a_9 \tilde{p}[m - m_0 - 1, n - n_0 - 1] \end{aligned} \quad (9)$$

where  $a_1, a_2, \dots, a_9$  represent the change in grayscale at the corresponding pixels shown in Figure 3 when we toggle  $(m_0, n_0)$ . The new perceived error is

$$\tilde{e}'[m, n] = \tilde{f}[m, n] - \tilde{g}'[m, n]$$

Using (9) in (6) under the assumptions that

$$c_{\tilde{p}\tilde{p}}[m, n] = c_{\tilde{p}\tilde{p}}[-m, -n]$$

and

$$c_{\tilde{p}\tilde{e}}[m, n] = c_{\tilde{p}\tilde{e}}[-m, -n]$$

the change in the error metric can be expressed as

$$\begin{aligned} \Delta \varepsilon = & c_{\tilde{p}\tilde{p}}[0, 0] \sum_{i=1}^9 a_i^2 - 2 \sum_{i=-1}^1 \sum_{j=-1}^1 a_{3(i+1)+j+2} c_{\tilde{p}\tilde{e}}[m_0 + i, n_0 + j] + \\ & 2(a_1 a_9 c_{\tilde{p}\tilde{p}}[-2, -2] + (a_1 a_8 + a_2 a_9) c_{\tilde{p}\tilde{p}}[-2, -1] + \\ & (a_1 a_7 + a_2 a_8 + a_3 a_9) c_{\tilde{p}\tilde{p}}[-2, 0] + (a_2 a_7 + a_3 a_8) c_{\tilde{p}\tilde{p}}[-2, 1] + a_3 a_7 c_{\tilde{p}\tilde{p}}[2, 2] + \\ & (a_1 a_6 + a_4 a_9) c_{\tilde{p}\tilde{p}}[-1, -2] + (a_1 a_5 + a_2 a_6 + a_4 a_8 + a_5 a_9) c_{\tilde{p}\tilde{p}}[-1, -1] + \\ & (a_1 a_4 + a_2 a_5 + a_3 a_6 + a_4 a_7 + a_5 a_8 + a_6 a_9) c_{\tilde{p}\tilde{p}}[-1, 0] + \\ & (a_2 a_4 + a_3 a_5 + a_5 a_7 + a_6 a_8) c_{\tilde{p}\tilde{p}}[-1, 1] + (a_3 a_4 + a_6 a_7) c_{\tilde{p}\tilde{p}}[-1, 2] + \\ & (a_1 a_3 + a_4 a_6 + a_7 a_9) c_{\tilde{p}\tilde{p}}[0, -2] + \\ & (a_1 a_2 + a_2 a_3 + a_4 a_5 + a_5 a_6 + a_7 a_8 + a_8 a_9) c_{\tilde{p}\tilde{p}}[0, -1] \end{aligned} \quad (10)$$

where  $c_{\tilde{p}\tilde{p}}[m, n]$  with  $m, n \in \{-2, \dots, 2\}$  is the auto correlation function for  $\tilde{p}[m, n]$ , and  $c_{\tilde{p}\tilde{e}}[m, n]$  is the cross correlation between  $\tilde{p}[m, n]$ , and  $\tilde{e}[m, n]$ . Since  $c_{\tilde{p}\tilde{p}}[m, n]$  is independent of the original image and the halftone, it can be precomputed and stored in a LUT. When a trial

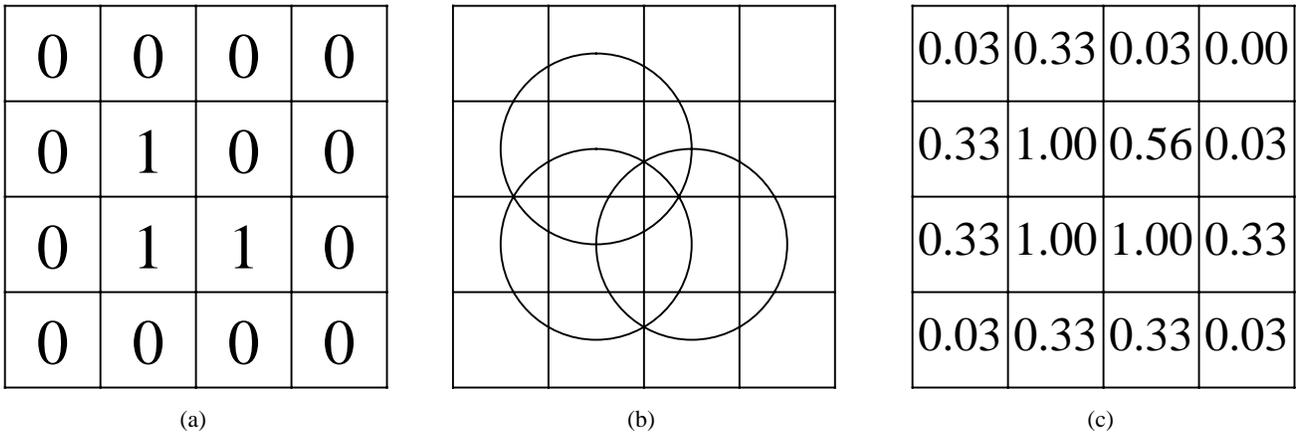


Figure 4. Equivalent grayscale image,  $a = 0.33$ ,  $b = 0.03$ ,  $g = 0.1$ . (a). Digital halftone. (b). Printer model output. (c). Equivalent grayscale image.

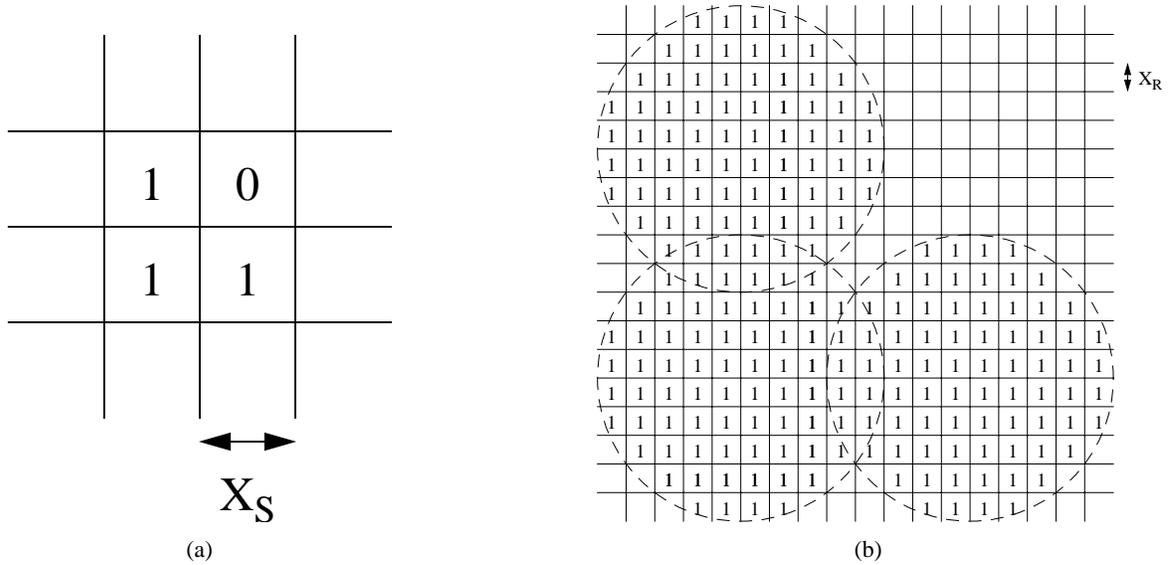


Figure 5. Printer simulation. (a). Bitmap sent to simulated printer. (b). Bitmap sent to real printer.

change is accepted, the LUT containing  $c_{\tilde{p}\tilde{e}}[m, n]$  is updated to as

$$\begin{aligned}
 c'_{\tilde{p}\tilde{e}}[m, n] = & c_{\tilde{p}\tilde{e}}[m, n] - a_1 c_{\tilde{p}\tilde{p}}[m - m_0 + 1, n - n_0 + 1] - \\
 & a_2 c_{\tilde{p}\tilde{p}}[m - m_0 + 1, n - n_0] - a_3 c_{\tilde{p}\tilde{p}}[m - m_0 + 1, n - n_0 - 1] - \\
 & a_4 c_{\tilde{p}\tilde{p}}[m - m_0, n - n_0 + 1] - a_5 c_{\tilde{p}\tilde{p}}[m - m_0, n - n_0] - \\
 & a_6 c_{\tilde{p}\tilde{p}}[m - m_0, n - n_0 - 1] - a_7 c_{\tilde{p}\tilde{p}}[m - m_0 - 1, n - n_0 + 1] - \\
 & a_8 c_{\tilde{p}\tilde{p}}[m - m_0 - 1, n - n_0] = a_9 c_{\tilde{p}\tilde{p}}[m - m_0 - 1, n - n_0 - 1]
 \end{aligned} \quad (11)$$

It can be seen from (10) and (11) that updating the error only involves a few LUTs and some scalar multiplications and additions. Similarly, the error update for swapping pixels can be efficiently implemented with LUTs.

## Experimental Results

To assess the quality of the halftoned image when printer closely matches the hard circular dot overlap model, we

use a simulated printer. The image is halftoned at simulated printer resolution  $R_s = 1/X_s$  and printed at real printer resolution  $R_A = 1/X_R$ , where  $X_s$  and  $X_R$  are the grid spacings for the simulated and the real printers, and  $S = R_A/R_s$  is an integer valued scale factor. After halftoning, each simulated printer dot is replaced by a cluster of real printer dots. The size of the cluster is determined by  $S$  and  $\rho$ . An example illustrating the effect of transforming the image halftoned at simulated printer resolution to the actual printer resolution is shown in Figure 5. The bit pattern sent to the simulated printer at  $R_s = 75$  dpi is shown in Figure 5(a). To achieve  $R_A = 600$  dpi, the required scale factor is eight. The value of  $\rho$  is taken to be 1.25. The  $8 \times 8$  grid of the actual printed dots for each pixel in the halftone image is given in Figure 5(b). The empty positions indicate the absence of a dot.

We generated halftones for three different images using DBS both with and without the hard circular dot overlap model. In all the images, the viewing distance is 24 inches and the real printer resolution is 600 dpi. The first image called ‘‘Ramp’’ is of a gray scale ramp, with

resolution  $256 \times 768$ . The second image called “Bullseye” has resolution  $512 \times 512$ . The third image called “Big-tree” is of one of the authors looking up a redwood tree in Northern California. Its resolution is  $780 \times 548$ . To provide a meaningful comparison, all images halftoned by DBS without an overlap model have been tone corrected prior to halftoning.

Figure 6 contains the halftones of the “Ramp” image. The simulated printer resolution is 85 dpi and the scale factor is seven. The images in Figures 6(b) and 6(d) are the upper left quarter of the images in Figures

6(a) and 6(c), respectively, magnified four times. It can be observed from Figure 6(a) that the image generated by DBS without the overlap model is patchy and mottled at high absorptances. Texture contouring is also present. On the other hand, the halftone generated by DBS with the overlap model, shown in Figure 6(c), is consistently smooth. There is some clipping at low absorptances. This is because no tone correction was applied to the image before halftoning with DBS with the overlap model. Such clipping may be eliminated by tone correction as discussed in [16].

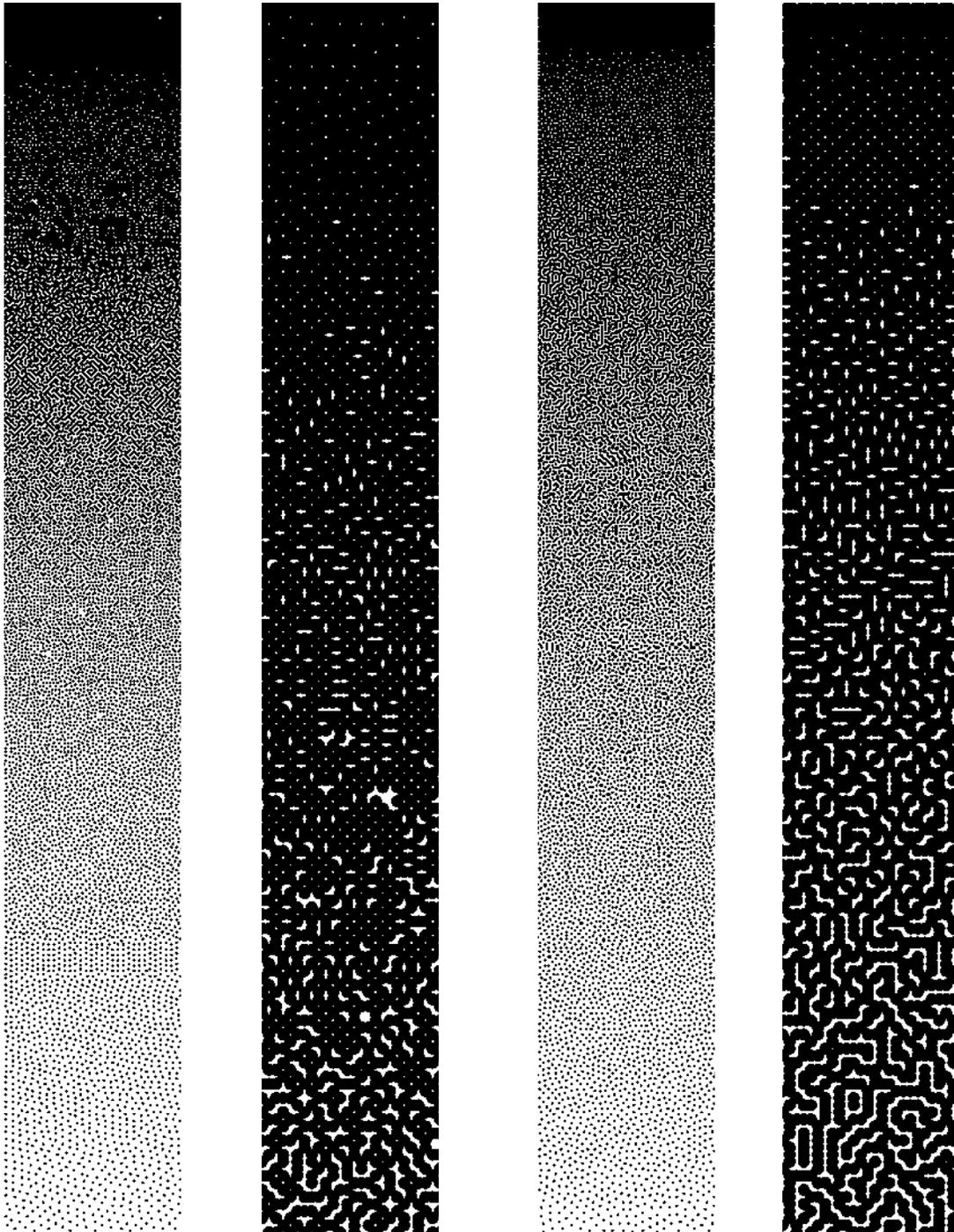


Figure 6. Ramp image halftoned by DBS. (a). Without dot-overlap model. (b). Without dot-overlap model. (c). With dot-overlap model. (d). With dot-overlap model.

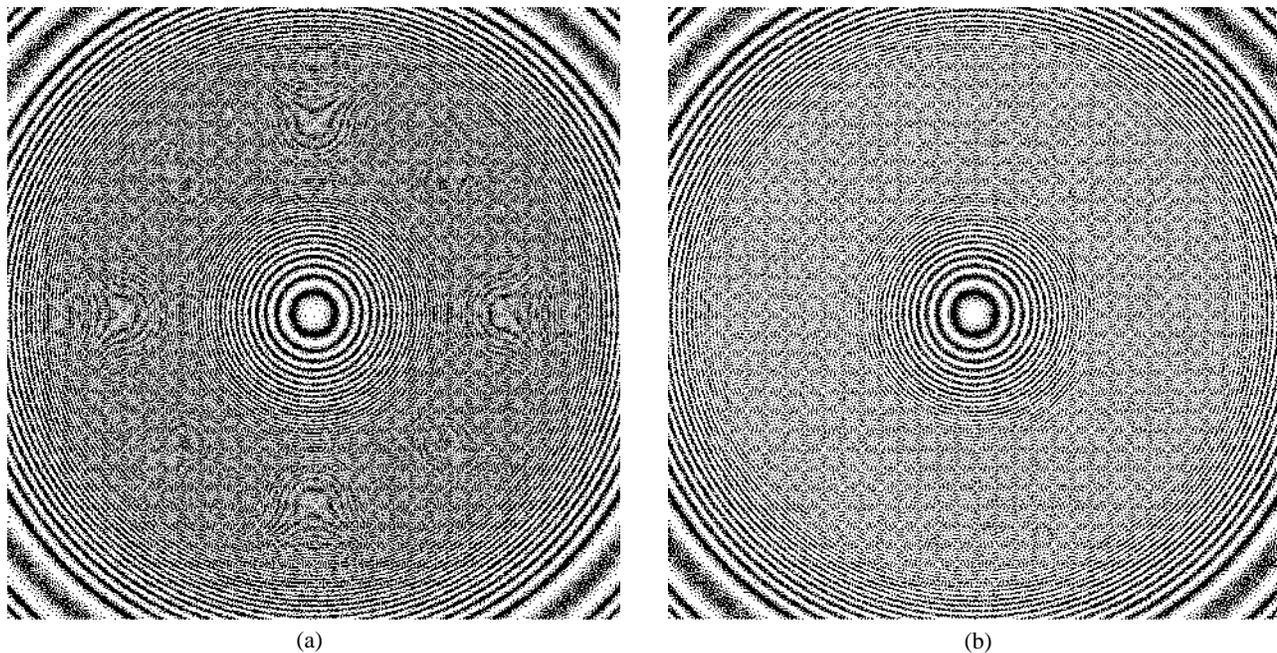


Figure 7. Bullseye image halftoned by DBS. (a). Without dot-overlap model. (b). With dot-overlap model.

The “Bullseye” image is halftoned at a simulated printer resolution of 120 dpi. The scale factor is five. The resulting halftones are in Figure 7. It can be seen from Fig. 7(a) that the halftone generated with DBS without the overlap model contains moiré patterns. No such artifact exists in the image when DBS with the overlap model is used, as is apparent from Figure 7(b).

Figures 8 and 9 contain the halftones of the “Big-tree” image. These images are halftoned at a simulated printer resolution of 85 dpi and the scale factor is seven. In the halftone generated by DBS without the overlap model, shown in Fig. 8, there are objectionable quantization artifacts, especially on the lower left of the tree-trunk. On the other hand, DBS with the overlap model does not contain these artifacts; furthermore it has much better rendition in the shadow areas on either side of the tree trunk which is evident in Figure 9.

### Conclusions

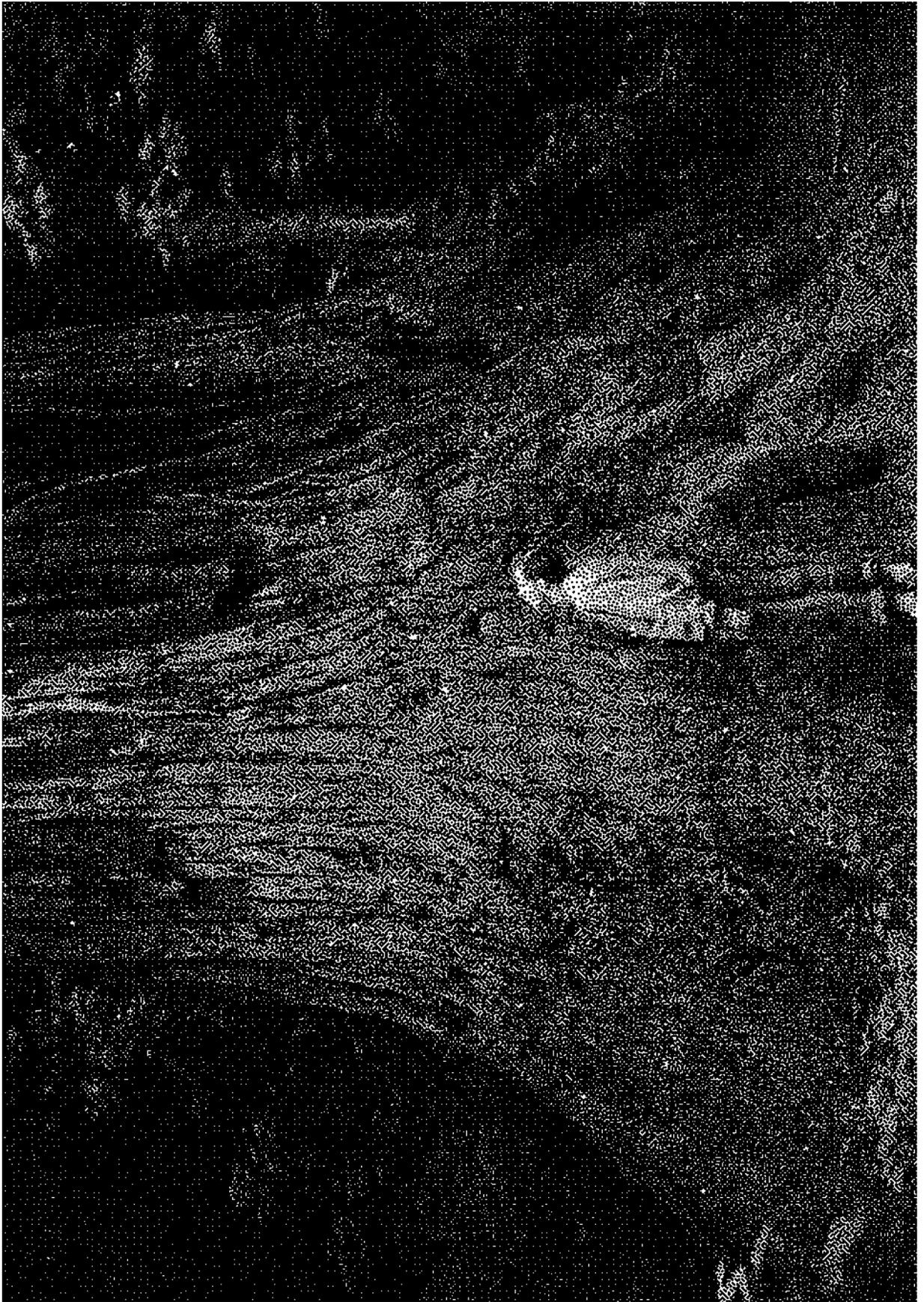
We have shown that DBS coupled with the hard circular dot-overlap model yields images that have accurate tone rendition, high spatial resolution and visually pleasing textures. It has the ability to exploit intermediate gray levels caused by dot overlap. Unlike the traditional DBS algorithm, there is no tone correction required prior to halftoning except to eliminate clipping at the extremes of the gray scale range. We have also shown that the algorithm can be efficiently implemented by precomputing and storing as many terms as possible.

### Acknowledgments

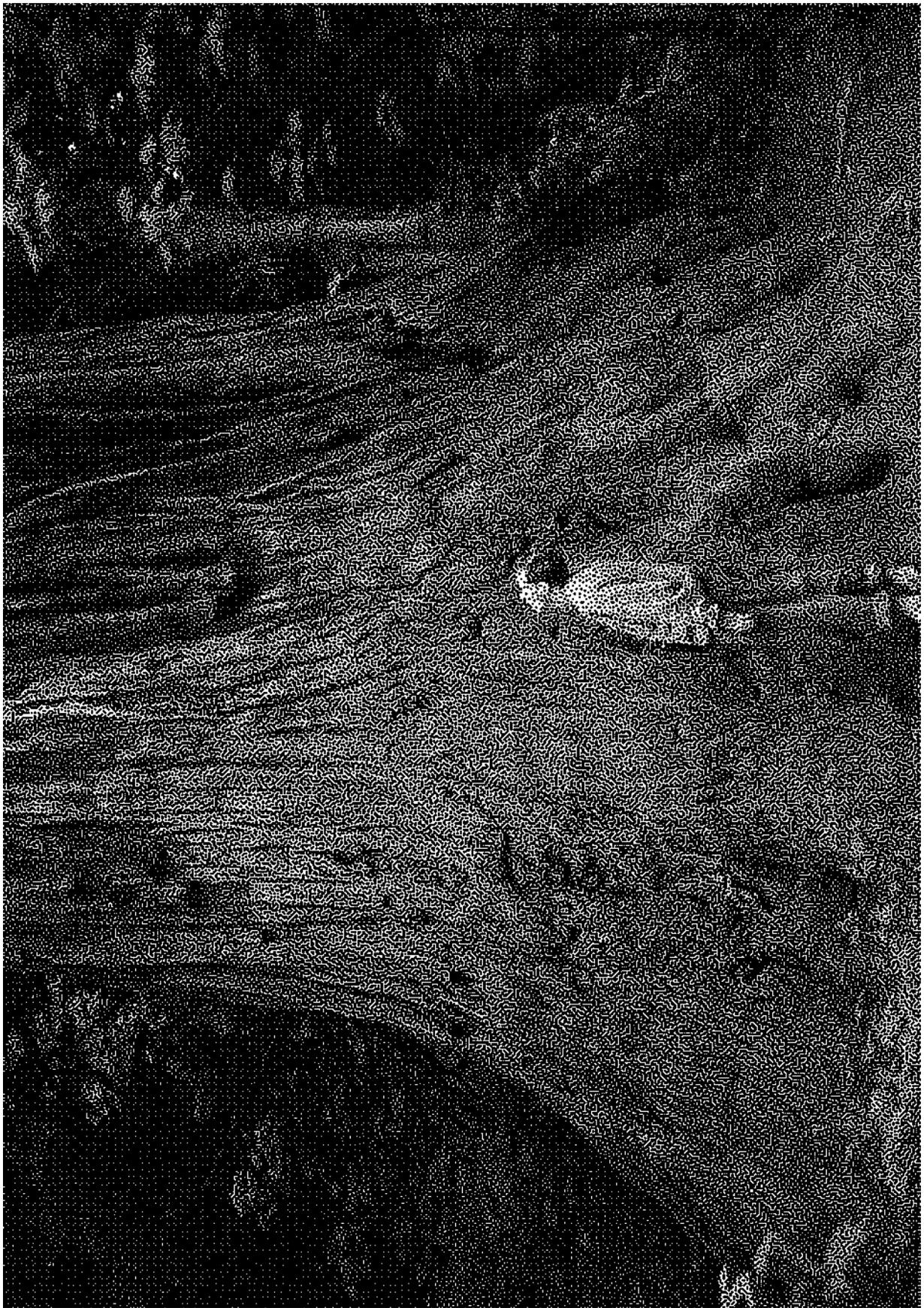
This research was supported by the Hewlett-Packard Co.

### References

1. P. G. Roetling and T. M. Holladay, “Tone reproduction and screen design for pictorial electrographic printing,” *J. Applied Photo. Eng.*, vol. **15**, no. 4, pp. 179–182, 1979.
2. J. P. Allebach, “Binary display of images when spot size exceed step size,” *Applied Optics.*, vol. **19**, pp. 2513–2519, Aug. 1980.
3. P. Stucki, “MECCA - a multiple-error correcting computation algorithm for bilevel image hardcopy reproduction,” res. rep. *RZ1060 IBM Res. Lab*, Zurich, Switzerland, 1981.
4. R. L. Stevenson and G. R. Arce, “Binary display of hexagonally sampled continuous-tone images,” *Journal of the Optical Society of America A*, vol. **2**, pp. 1009–1013, July. 1985.
5. T. N. Pappas and D. L. Neuhoff, “Model-based halftoning,” in *Proc. SPIE, Human Vision, Vis. Proc., and Dig. Display II*, San Jose, CA, Feb. 1991, pp. 244–255.
6. C. J. Rosenberg, “Measurement-based evaluation of a printer dot model for halftone algorithm tone correction,” *J. of Imaging Technology*, vol. **2**, pp. 205–212, July 1993.
7. Q. Lin and J. Wiseman, “Impact of electrophotographic printer dot modeling on halftone image quality,” in *SID Int. Symp. Tech. Papers*, Seattle, WA, May 1993, pp. 147–150.
8. Q. Lin, “Improving halftone uniformity and tonal response,” in *Proc. IS&T’s 10th Int. Cong. on Adv. in Non-Impact Tech.*, New-Orleans, Louisiana, Oct. 1994, pp. 377–380; see *Recent Progress in Digital Halftoning, Vol. I, pg. 6*.
9. Q. Lin, “Screen design for printing,” in *Proc. IEEE Int. Conf. on Image Processing*, vol. II, Washington, D.C., Oct. 1995, pp. 331–334.
10. M. Analoui and J. P. Allebach, “Model based halftoning



*Figure 8. Big-tree image halftoned by DBS without dot-overlap model.*



*Figure 9. Big-tree image halftoned by DBS with dot-overlap model.*

- using direct binary search,” in *Proc. SPIE/IS&T Symp. on Elec. Imag. Sci. and Tech.*, San Jose, CA, Febr. 1992, pp. 96–108.
11. T. N. Pappas and D. L. Neuhoff, “Least-squares model-based halftoning,” in *Proc. SPIE/IS&T Symp. on Elec. Imag. Sci. and Tech.*, San Jose, CA, Febr. 1992, pp. 165–176.
  12. J. B. Mulligan and A. J. Ahumada Jr., “Principled halftoning based on human visual models,” in *Proc. SPIE/IS&T Symp. on Elec. Imag. Sci. and Tech.*, San Jose, CA, Febr. 1992, pp. 109–121.
  13. D. A. Carrara, M. Analoui, and J. P. Allebach, “Recent progress in digital halftoning,” in *Proc. IS&T 8th Int. Cong. Adv. Non-Impact Printing Tech.*, Williamsburg, VA, Oct. 1992, pp. 265–270; *see Recent Progress in Digital Halftoning, Vol. 1, pg. 139.*
  14. T. J. Flohr and J. P. Allebach, “Halftoning via direct binary search with a stochastic dot model,” in *Proc. IS&T 3rd Tech. Sym. on Prepress, Proofing and Printing*, Chicago, IL, Nov. 1993; *see Recent Progress in Digital Halftoning, Vol. 1, pg. 87.*
  15. J. P. Allebach, T. J. Flohr, D. P. Hilgenberg, C. B. Atkins, and C. A. Bouman, “Model-based halftoning via direct binary search,” in *Proc. IS&T’s 47th Annual Conf./ICPS*, Rochester, NY, May 1994, pp. 476–481; *see Recent Progress in Digital Halftoning, Vol. 1, pg. 50.*
  16. D. J. Lieberman and J. P. Allebach, “Digital halftoning using the direct binary search algorithm,” in *Proc. IS&T’s 5th International Conf. on High Tech.*, Chiba, Japan, Sep. 1996.
  17. R. Näsänen, “Visibility of halftone dot textures,” *IEEE Trans. on Syst., Man, Cyber.*, vol. **14**, no. 6, pp. 920–924, 1984.
  18. T. N. Pappas and D. L. Neuhoff, “Printer models and error diffusion,” *IEEE Trans. on Image Processing*, vol. 4, pp. 66–80, Jan. 1995.
- \* Previously published in *IS&T/OSA Optics & Imaging in the Information Age*, pp. 383–390, 1996.
- 
-