Thrasyvoulos N. Pappas,
Jan P. Allebach, and David L. Neuhoff

Model-Based Digital Halftoning

Digital halftoning is the process of generating a pattern of pixels with a limited number of colors that, when seen by the human eye, is perceived as a continuous-tone image. Digital halftoning is used to display continuous-tone images in media in which the direct rendition of the tones is impossible. The most common example of such media is ink or toner on paper, and the most common rendering devices for such media are, of course, printers. Halftoning works because the eye acts as a spatial low-pass filter that blurs the rendered pixel pattern, so that it is perceived as a continuous-tone image.

Although all halftoning methods rely, at least implicitly, on some understanding of the properties of human vision and the display device, the goal of model-based halftoning techniques is to exploit explicit models of the display device and the human visual system (HVS) to maximize the quality of the displayed images. This is illustrated in Figure 1.

Based on the type of computation involved, halftoning algorithms can be broadly classified into three categories [1]: point algorithms (screening or dithering), neighborhood algorithms (error diffusion), and iterative algorithms [least squares and direct binary search (DBS)]. All of these algorithms can incorporate HVS and printer models. The best halftone reproductions, however, are obtained by iterative techniques that minimize the (squared) error between the output of the cascade of the printer and visual models in response to the halftone image and the output of the visual model in response to the original continuous-tone image.

©ARTVILLE

Introduction

We will discuss HVS models. Although all halftoning algorithms rely on the fact that the eye acts as a spatial low-pass filter, many do not make use of an explicit HVS model, and as such, they are not considered to be vision-model based. Vision-model-based halftoning algorithms, on the other hand, explicitly incorporate a model of the HVS and exploit it to produce halftone images of higher visual quality.

Also discussed are display models. We focus on black and white (B&W) printers. However, many of the ideas extend to color printers. For other display devices, such as displays of handheld devices and especially cell phones, techniques similar to those used for printers can be used. The most elementary halftoning techniques for B&W printers assume that the "blackness," i.e., the perceived gray level, of a printed binary pattern is proportional to the fraction of black dots in the pattern. (Recall that the eye acts as a low-pass filter.) In effect, this assumes that the area occupied by each black dot is roughly the same as the area occupied by the white space left where no dot is placed. It follows that the "desired" shape for the black dots produced by a printer would be $T \times T$ squares, where T is the dot spacing. However, actual printers do not obey this assumption. For purely physical reasons, printers produce more or less circular dots that overlap adjacent spaces, causing the perceived gray level to be darker than the fraction of black dots. We refer to this as dot overlap. As we will see in the following sections, a minimal overlap is necessary so that the printer is capable of darkening entirely a portion of the page. In addition, most printers typically produce dots that appear larger than this, a phenomenon that is called dot gain. This can be caused by one or more of the following effects: optical gain (due to scattered light being trapped under the colorant), mechanical gain (spreading of the colorant on the medium), and electric field gain (which occurs in electrophoto-

1053-5888/03/$17.00©2003IEEE

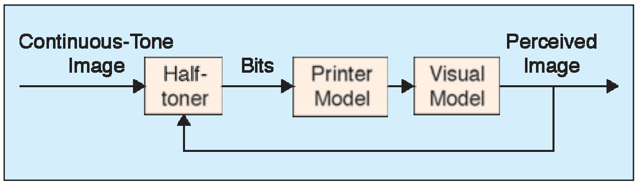
graphic printers). If a halftoning technique does not account for such nonlinearities, the resulting tone scale will be distorted.

The classical approach to mitigating the effects of the phenomena described above is to cluster black dots so the percentage effect of dot gain on perceived gray level is reduced. Such clustering reduces the spatial resolution of the resulting images and increases the visibility of halftone textures. On the other hand, dispersed dot halftoning algorithms can provide high spatial resolution and excellent halftone textures, but they are very sensitive to dot gain as well as other printer distortions. Model-based techniques, on the other hand, can rely on accurate printer models to exploit (rather than avoid) printer distortions in order to maximize the quality of the resulting images. Thus, they make it possible to extend the benefits of dispersed dot techniques to any device for which an accurate model is available. A third approach is to use tone correction, that is, a compensating gray-scale transformation applied to the image before halftoning [50, p. 36]. However, this approach does not give good detail rendition and cannot match the tone levels as precisely as the model-based techniques. Moreover, for some techniques (e.g., when the printer introduces a nonmonotonic nonlinearity) this approach does not work at all. In addition, since dot interactions can be very complex, the tone correction approach does not work well for dispersed dot techniques because there is very little control over the dot microstructure, resulting in severe artifacts. Thus, tone correction is less suitable for dispersed dot printing with electrophotographic printers.

We present a general class of printer models that facilitate the implementation of model-based techniques and examine a number of specific models. We then examine several model-based techniques that exploit HVS and/or printer models to improve the quality of the halftone images. Such techniques include screening, error diffusion, and iterative algorithms (least squares and DBS).

For a given display device and viewing conditions, the performance of a halftoning technique, i.e., the visual quality of the halftone images it produces, can be judged in terms of its spatial resolution (the ability to display detail, such as edges, with sharpness), tone scale resolution (the ability to display many different gray levels), tone scale accuracy (the degree to which the displayed halftones are perceived with proper gray levels), and texture (how visible or annoying the halftone-induced texture is).

These attributes can be tested in different regions of an image. The spatial resolution affects regions of rapidly changing intensity, such as at an edge. In regions of constant intensity, the tone scale resolution and accuracy are important if an exact reproduction is desired. However, the most important consideration is the visibility of the halftone induced textures. Finally, the regions of slowly changing intensities are perhaps the most challenging for a halftoning technique. Here the most important consideration is the compatibility of adjacent tone levels. For if adjacent tone levels are not rendered with compatible halftone patterns, then false contours will be visible in regions where the intensity changes slowly. There can be two causes for this effect: the lack of adequate tone scale resolution and the incompatibility of the halftone textures that correspond to adjacent tone levels.



▲ *1. Model-based halftoning.*

In traditional halftoning techniques (clustered dot screens), the main tradeoff (controlled by the period of the screen) is between spatial resolution and texture (low visibility) on the one hand, and tone scale resolution and accuracy on the other. In contrast, in error diffusion and iterative techniques, spatial resolution is very high in general, and the main tradeoff is between texture and tone scale resolution. There are, however, other important tradeoffs. For example, clustered-dot screening techniques offer robustness to dot gain and other printer distortions, while model-based error diffusion and iterative algorithms offer better spatial resolution and texture for a specific device. The tradeoff between robustness to printer distortions and texture and spatial resolution is one of the key issues in green-noise halftoning techniques discussed in the second article of this issue.

HVS Models

Halftoning works because the eye, i.e., the HVS, acts as a spatial low-pass filter that blurs patterns of printed dots so as to be perceived as various gray levels. Though every halftoning method is based on this understanding (i.e., model) of human vision, certain halftoning methods make explicit use of an HVS model. Such vision-model-based halftoning methods will be described later. In this section, we describe some of the HVS models that they use.

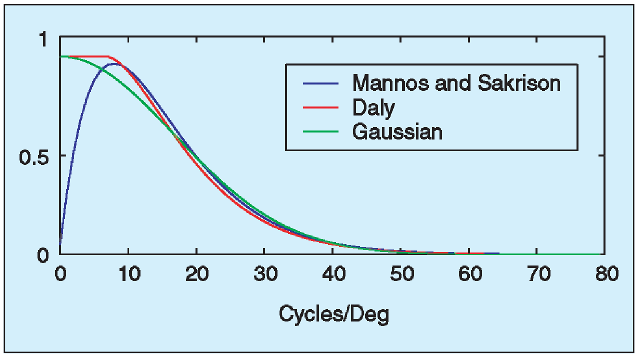
There are two closely related concepts: visual fidelity (VF) metrics and HVS models. A VF metric is a function $V(z,\hat{z})$ that is intended to indicate the degree to which the image $\hat{z}$ is perceived to differ from the image z. Overviews of VF metrics can be found in [2] and [3]. In the context of halftoning, an HVS model is a system S that generates $S(z)$, which is an image like z, except that features have been enhanced or suppressed in proportion to their perceptibility. There is a close connection between VF metrics and HVS models. For example, given an HVS model S, one can obtain a VF metric $V(z,\hat{z})$ by computing the energy in the difference image $S(z)-S(\hat{z})$. This approach is used in many vision-model-based halftoning methods.

The most basic HVS model is simply a two-dimensional, linear, shift-invariant filter. For example, the widely cited paper by Mannos and Sakrison [4] found the following filter frequency response to be good for predicting the subjective quality of coded images:

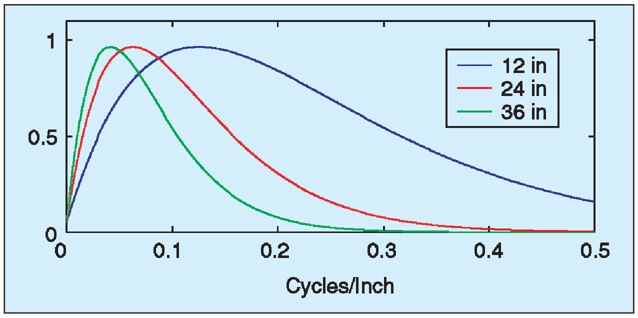
$$H_r(f_r) = 2.6(0.0192 + 0.114 f_r)\exp\left\{-(0.114 f_r)^{1.1}\right\} \quad (1)$$

where frequency $f_r = \sqrt{f_x^2 + f_y^2}$ is the RMS value of the horizontal and vertical frequencies f_x, f_y in units of cycles per degree (cycles/deg). As illustrated in Figure 2, which shows H plotted versus f_r, this filter has a bandpass character, peaking at 7.9 cycles/deg. We view this frequency response as representing the sensitivity of the eye at the various frequencies. Alternatively, direct estimates of the sensitivity of the eye at each frequency have been made and then used to define the frequency response of a filter in an HVS model. Such sensitivities are typically found by measuring the faintest sinusoidal pattern at a given frequency that is distinguishable from the background. The inverse of such threshold sinusoidal amplitudes (actually, the thresholds are divided by the background intensity) is a measure of HVS sensitivity to the corresponding frequency, and a plot of such sensitivities versus frequency is often referred to as the contrast sensitivity function. Direct estimates of the peak sensitivity range from 3 to 10 cycles/deg [5, p. 55]. The frequency responses of various filters that have been used in model-based halftoning can be found in [6]-[14].

The decrease in sensitivity at higher frequencies is due to the optical characteristics of the eye, e.g., properties of the cornea and lens. Indeed, adopting optics terminology, the frequency response of the model is often called the modulation transfer function (MTF) of the eye. The decrease in sensitivity at low frequencies is a consequence of the eye's ability adapt to a broad range of lighting conditions. Because of this, the eye has difficulty determining the intensity of any large constant intensity region.



▲ 2. Several frequency responses.



▲ 3. The Mannos-Sakrison frequency response with frequency expressed in cycles per inch at several viewing distances.

It is well known that the eye is less sensitive to obliquely oriented features than to horizontally and vertically oriented features. This can be exploited by using a filter of the form

$$H(f_r, \theta) = H_r\left(\frac{f_r}{s(\theta)}\right) \quad (2)$$

where $\theta = \arctan(f_y/f_x)$ is the angle corresponding to f_x, f_y. The following choice of $s(\theta)$ was proposed by Daly [6] and used for model-based halftoning methods in [7], [8], [12], and [14]

$$s(\theta) = 0.15\cos(4\theta) + 0.85. \quad (3)$$

A number of more sophisticated models have been proposed for the HVS; a comprehensive reference is [15]. For example, some models add a memoryless nonlinearity before the linear filter and a number of models include a bank of filters [16, p. 295], [17]-[20]. The latter model the multichannel nature of the HVS.

Although there are many more sophisticated HVS models, simple filter-based models have been predominantly used in vision-model-based halftoning. This is especially true in iterative vision-model-based methods, where simplicity is required so that the method is computationally feasible. Thus in the remainder of this section, we focus on models consisting only of a filter.

For use in halftoning, a filter-based HVS model such as described by (1) needs converting in three steps: to frequency in units of cycles per inch, to a spatial domain point-spread function (impulse response), and to a sampled version of the latter. The conversion to frequency in cycles per inch (cycles/in) is accomplished via

$$H_d(f_r) = H_r\left(f_r \frac{d\pi}{180}\right) \quad (4)$$

where f is frequency in cycles/in and d is the viewing distance in inches from the eye to the image. As illustrated in Figure 3, the frequency response changes substantially with viewing distance. For example, as d increases, the peak of the response shifts to lower frequencies, implying that rapid intensity fluctuations become increasingly difficult to observe. Conversely, as d decreases, the fine detail in the image and the individual halftone dots, become visible.

Because viewers cannot be expected to maintain a fixed distance from an image, vision-model-based halftoning methods must, in effect, be designed for a range of HVS models, rather than just one specific model such as in Figure 2. This is one reason why the models used in such methods often consist of low-pass filters, like those shown in Figure 2 with a red line, rather than bandpass filters like the Mannos-Sakrison filter shown with a blue line. A vision-model-based halftoning method that employs a low-pass filter can be expected to respect the low frequencies that will become quite visible at larger viewing distances. Accordingly, several low-pass filters have been proposed. For example, as described in [7], Daly [6] proposed a bandpass frequency response similar to that of Figure 2 and then formed a low-pass filter by simply extending the peak to the origin, as illustrated in Figure 2. This approach has been used in [7], [9]-[14]. As a second form of low-pass filter, several authors have used filters with an exponentially decaying frequency response [8], [21]-[24]. Finally, we mention that Neuhoff and Pappas found that, as illustrated in Figure 2, a low-pass filter with a Gaussian frequency response with standard deviation $\sigma = 16.7$ cycles/deg well matches the high frequency portion of the Mannos-Sakrison frequency response [10], [11].

Converting a frequency response to an impulse response is accomplished in the usual way with the inverse Fourier transform. Typically, the filter in a vision-model-based halftoning method is implemented in the spatial domain by direct convolution. Since the images and halftones are, of course, sampled, the filter impulse responses must be truncated and sampled. For example, Figure 4 shows the Gaussian filter of Figure 2, truncated to the interval ±0.064° and sampled under two different conditions. In general, if $h(\theta)$ denotes a (one-dimensional) impulse response, where θ has units of degrees, then the truncated and sampled impulse response is

$$h_{d,R}[n] = h_r\left(n\frac{180}{dR\pi}\right), \quad \frac{dR\pi}{180}\theta_{\min} \le n \le \frac{dR\pi}{180}\theta_{\max} \tag{5}$$

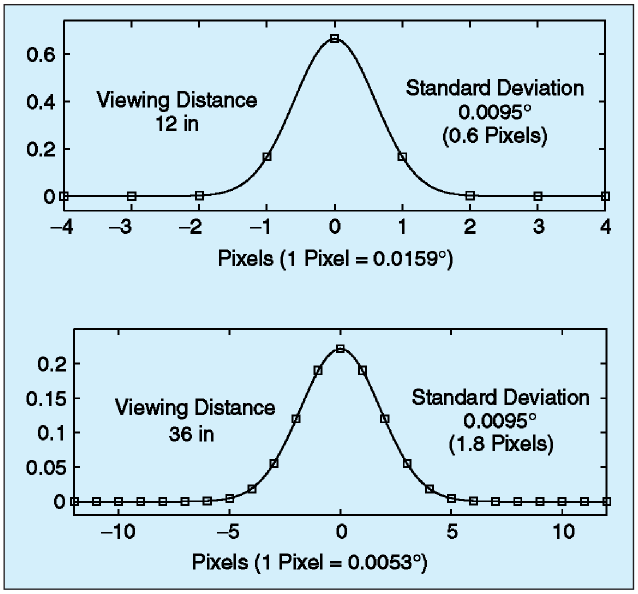
where d is the viewing distance, R is the pixel resolution in dots per inch (dpi) of the image and halftone, and $\theta_{\min}$ and $\theta_{\max}$ specify the degree range to which h is to be truncated. We should mention that for low resolutions and short viewing distances, the filter design should be done more carefully to avoid aliasing.

Since the above depends on d and R only through their product, this product $s = dR$ becomes a key parameter, called the scale factor. It may be interpreted as the perceived resolution in dots per radian, when viewing an image printed with R dpi from distance d. It also has a significant influence on the halftone patterns produced by a model-based halftoning algorithm that uses the HVS model. A large value of d tells the halftoning algorithm that the eye will do much blurring of the image and its halftoned representation. Thus, it will tend to produce coarsely textured halftoned patterns. On the other hand, a small value of d tells the halftoning algorithm that dots and other fine detail are more visible. In this case, the algorithm will micromanage the placement of dots, creating a finer texture. A change in R has a similar effect.

Digital halftoning is the process of generating a pattern of pixels with a limited number of colors that, when seen by the human eye, is perceived as a continuous-tone image.

It is also interesting to consider the effect of changing s on the sampled impulse response $h_{d,R}[n]$. When the scale factor s is small, the sampled impulse response has small support, i.e., few nonzero terms, because the samples of $h(\theta)$ are widely spaced. Thus, when a model-based halftoning algorithm seeks a halftone pattern for a patch of constant gray level, it works with only a small number of dots. This means it can only create a small number of gray levels, which causes the tone scale resolution to be coarse. Moreover, for some gray levels, there will tend to be a single pattern having the desired gray level and the best texture. This, combined with the low grayscale resolution, tends to cause false contouring. To see why, consider a region that increases gradually in gray level. Then there will be an imaginary line on one side of which the halftoning algorithm consistently produces one pattern of dots, while on the other side it produces a different pattern of dots. The result is a visible but false contour line. On the other hand, when the scale factor is large, the support of the filter will include many samples, giving the model-based halftoning technique more flexibility in choosing the best patterns to produce many gray levels without false contouring. As mentioned previously, the resulting halftones will tend to be coarse but not visible due to the longer viewing distance.

Because the scale factor has such impact on the nature of the halftones produced by model-based halftoning, one may wish to consider it to be a parameter to be tuned, even when the viewing distance is known in advance. In



▲ *4. The sampled Gaussian point spread functions (impulse responses) for two viewing distances at 300 dpi.*

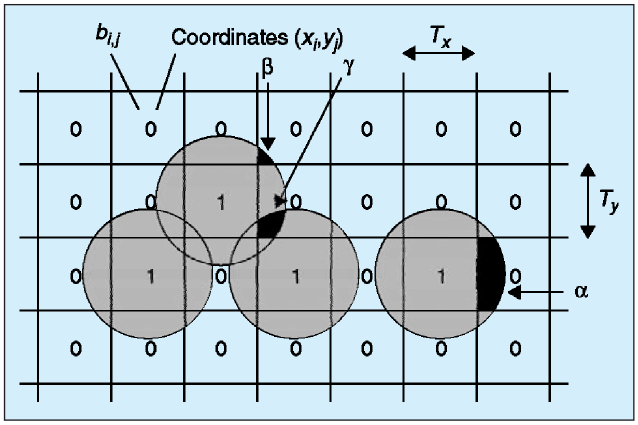
this case, one chooses s to obtain a pleasing compromise between the fineness of the halftone textures on the one hand, and the fineness of the tone scale resolution and the avoidance of false contouring on the other.

Along similar lines in the context of least-squares model-based (LSMB) DBS, Kim and Allebach [23] used a sum of two separable Gaussians (corresponding to different viewing distances) to represent the autocorrelation of the HVS point spread function and visually optimized the parameters of the model. However, they found that no single set of parameter values gave the best performance across the entire tone scale. So they developed a dual metric approach that effectively uses a larger scale factor in the highlights, midtones, and shadows, and a smaller scale factor elsewhere.

Printer Models

The purpose of a printer model is to accurately predict the actual gray levels produced by a printer. In addition, it should take a form that is easy to incorporate in a halftoning algorithm. With accurate printer models, model-based halftoning techniques can exploit the printer characteristics to produce higher quality renditions of digital images.

To a first approximation, laser printers are capable of producing black spots (usually called dots) on a piece of paper, usually on a rectangular array of pixels with horizontal and vertical spacing of T_x and T_y inches, respectively. The reciprocal of T_x (T_y) is the horizontal (vertical) resolution of the printer in dpi. We will use the following terminology and notation. Pixel (i,j) is the T_x by T_y rectangle whose center is the point (x_i, y_j), with $x_i = iT_x + T_x/2$ in from the left of the image, and $y_j = jT_y + T_y/2$ in from the top of the image. We will refer to the collection of all pixels, i.e., all such rectangles, as the printing lattice. The printer is controlled by an $N_x \times N_y$ binary array $[b_{i,j}]$, where $b_{i,j} = 1$ indicates that a black dot is to be placed at pixel center (x_i, y_j), and $b_{i,j} = 0$ indicates that a black dot is not placed there. We will sometimes refer to the latter as a "white" dot. Finally, we assume that the grayscale image (to be halftoned) has been sampled so there is one pixel per dot to be generated; otherwise, interpolation is necessary. Thus, it is also defined on an $N_x \times N_y$ array $[z_{i,j}]$, which takes values in the interval [0,1]. We will assume that these values (gray levels) represent absorptance. Thus, the gray level of a black pixel will be 1, and the gray level of a white pixel will be 0. Reflectance will then be defined as 1 – absorptance. We will refer to the collection of all image pixels as the image lattice.



▲ *5. Circular dot overlap (ρ = 1.25).*

As we saw in the introduction, to a first approximation, actual printers produce roughly circular black dots, as shown in Figure 5. Depending on the printer technology, the size, shape, colorant density (and hence absorption uniformity), and placement of the black dots may vary. In addition, present day electrographic (laser or LED) printers provide dot modulation capabilities (e.g., changing the area of the printed dots), and thus, each pixel may be in one of several hundred states rather than the two states of conventional printers. In such cases, the array that controls the printer will take more than two values. In the following discussion, we will assume a binary printer but the ideas extend to the more general case.

The Sampled Grayscale Printer Model

We now describe a general class of printer models that forms the basis of model-based halftoning techniques. The main idea, introduced by Roetling and Holladay [25] and formalized by Pappas and Neuhoff in [26] and [27], is to estimate the average gray level of each pixel of the printed image as a function of the values of the binary array $[b_{i,j}]$ in the neighborhood of that pixel.

As a result of the phenomena such as those mentioned above, the gray level produced by the printer at any point in the image depends in some complicated way on the surrounding pixels. Let $u(s,t)$ be the gray level produced by the printer at point (s,t) located s inches from the left and t inches from the top of the image. Then

$$u(s,t) = f(s,t;B_{s,t}), \quad 0 \le s \le N_x T_x,\ 0 \le t \le N_y T_y \qquad (6)$$

where $B_{s,t}$ denotes the set of bits in a neighborhood of the point (s,t) and f is some function. As we will see, this function f could be deterministic or probabilistic. However, due to the close spacing of the dots and the limited spatial resolution of the eye, the gray level $u(s,t)$ of the printed image can be modeled as having a constant value $p_{i,j}$ within the area of pixel (i,j) as follows:

$$\tilde{u}(s,t) = p_{i,j}, \quad |x_i - s| < \frac{T_x}{2},\ |y_j - t| < \frac{T_y}{2} \qquad (7)$$

where, as we saw above, (x_i, y_j) are the coordinates of the center of the pixel. Although the gray level is not actually constant, the eye responds, essentially, only to the average gray level over the site. It is this average gray level that $p_{i,j}$ represents, i.e., the average of the function $f(s,t;B_{s,t})$ over the site. Thus the average gray level $p_{i,j}$ depends on the neighboring pixels in the form

$$p_{i,j} = \mathcal{P}(W_{i,j}), \quad 0 \le i < N_x\,, \; 0 \le j < N_y \qquad (8)$$

where $W_{i,j}$ is a window that consists of the bits in some neighborhood of $b_{i,j}$ and $\mathcal{P}$ denotes some function thereof. Note that the printer model is completely specified by the function $\mathcal{P}$, which like f in (6) could be deterministic or probabilistic. Thus, given the binary array $[b_{i,j}]$ that specifies the dot pattern to be printed, the printer model generates a new array $[p_{i,j}]$ of gray levels that has the same dimensions as the binary array as well as the grayscale image array. This is very important, as all processing can be done in the discrete domain without any need for resampling. We refer to this generic model as the sampled grayscale printer model (SGPM).

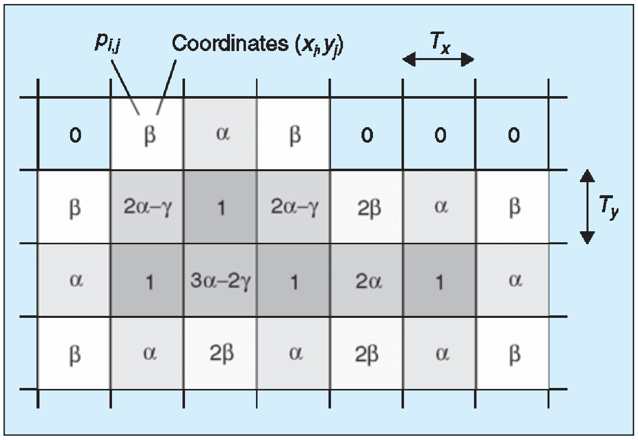
In the remainder of this section, we consider different ways for obtaining this function $\mathcal{P}$. A variety of phenomena may contribute to the appearance of the printed page, and the only place to account for them is in the function $\mathcal{P}$. For computational efficiency, it is essential that $p_{i,j}$ be entirely determined by the bits in a small window around $b_{i,j}$, e.g., a 3×3 window. In this case, the possible values of $\mathcal{P}$ can be listed in a table, e.g., with 2^9 elements. The individual elements of this table might be derived from a detailed physical understanding of the various phenomena effecting gray level or from measurements of the gray level that results when various dot patterns are printed. One example of the first approach uses the "circular dot-overlap model," which we describe below.

Circular Dot-Overlap Model

One of the simplest printer models assumes that the printer produces circularly shaped black dots [25], as shown in Figure 5. The dots are saturated so that the overlapping areas do not get any darker (logical OR). In this section, for simplicity, we will assume that the pixels are square (i.e., $T_x = T_y = T$). Note that neighboring black dots may overlap and that black dots may cover parts of adjacent white dots. The radius of the dots must be at least $T/\sqrt{2}$ so that they are capable of blackening a region of the page entirely. This means that there is always some overlap between black dots and adjacent white dots, resulting in a darkening of the gray level of the pixel that corresponds to the white dot. This results in significant distortion in the printed images. (The area of such a dot is $1.57\ T^2$, i.e., 57% larger than a $T \times T$ square!) Most printers produce black dots that are larger than the minimal size (dot gain), which further distorts the gray level.

The circular dot-overlap model proposed in [26] and [27] accounts for such distortions by estimating the gray level of each pixel of the printed image as the percentage of the pixel covered by ink. This area can be calculated easily from the radius of the dots. More specifically, the printer model takes the form

$$p_{i,j} = \mathcal{P}(W_{i,j}) = \begin{cases} 1, & \text{if } b_{i,j} = 1 \\ f_1\alpha + f_2\beta - f_3\gamma, & \text{if } b_{i,j} = 0 \end{cases} \qquad (9)$$



0	β	α	β	0	0	0
β	2α−γ	1	2α−γ	2β	α	β
α	1	3α−2γ	1	2α	1	α
β	α	2β	α	2β	α	β

6. SGPM output ($\rho = 1.25$).

where the window $W_{i,j}$ consists of $b_{i,j}$ and its eight neighbors. Here f_1 is the number of horizontally and vertically neighboring dots that are black, f_2 is the number of diagonally neighboring dots that are black and not adjacent to any horizontally or vertically neighboring black dot, and f_3 is the number of pairs of neighboring black dots in which one is a horizontal neighbor and the other is a vertical neighbor. The parameters α, β, and γ are the ratios of the areas of the shaded regions shown in Figure 5 to T^2 and can be expressed in terms of the ratio ρ of the actual dot radius to the ideal dot radius $T/\sqrt{2}$ [27]. Thus, the parameter ρ completely specifies the printer model. For typical write-black B&W printers, the value of ρ ranges from 1.0 to 1.7. Figure 6 shows the SGPM output for the dot pattern in Figure 5 with $\rho = 1.25$. The model is not very sensitive to small variations in ρ and applies to a wide range of printers. Note that the circular dot-overlap model can be used to account for both mechanical and optical dot gain.

Roetling and Holladay [25] were the first to propose a circular dot-overlap model and used it to improve the design of clustered dot screens. Several authors have used printer models in their papers [9]-[11], [26]-[31]. We will examine some of these in later sections.

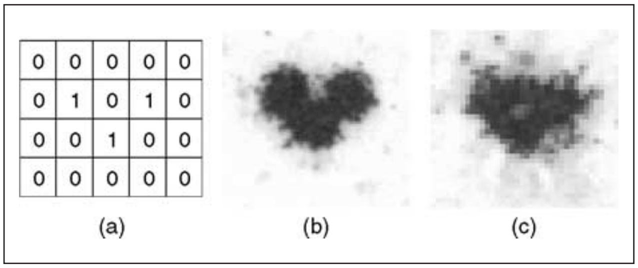
Tabular Models

As we saw earlier, the printer model predicts the gray level (i.e., the absorptance) of each pixel of the printed pattern as a function of the values of the binary array $[b_{i,j}]$ in the neighborhood of that pixel, as specified in (8). Such a function can be specified by a formula, as in the circular dot-overlap model, or by a table that lists a gray level for each input pattern.

Models such as the circular dot-overlap are accurate for many printers but cannot describe the behavior of all printers. Thus, different models may have to be derived for different printer technologies, or as we will see below, different printer resolutions. In addition, for some printers, the dot interactions can be very complex, and deriving a model based on a physical understanding of the printing process can be too difficult or too complicated.

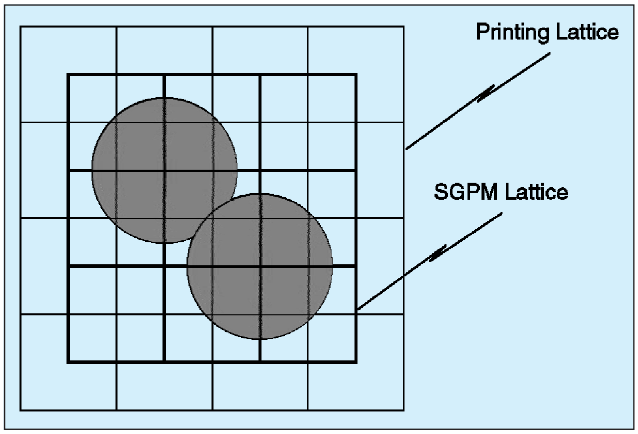
To avoid such problems, Pappas et al. [32] proposed a tabular approach for modeling printers that uses direct measurements of the absorptance of printed test patterns in order to obtain the table entries. This can be applied to any printer and makes very few assumptions about its characteristics. Their approach is based on macroscopic measurements of the absorptance of specially designed periodic test patterns. The average absorptance of each pattern is related to the printer model parameters by a set of linear equations. A constrained optimization problem must then be solved to obtain the model parameters. Note that a simple measuring device can be used and no precise alignment is required.

The estimation of the parameters (table values) in the tabular approach can be simplified considerably if a high resolution device is available for measuring the absorptance of individual pixels. This eliminates the need for solving a set of equations to obtain the table values but requires precise alignment. This microscopic approach was proposed by Baqai and Allebach [24], who used a high resolution (4,000 dpi) drum scanner to measure the absorptance of the central pixel for all possible 3×3 patterns. To solve the alignment problem, they printed reference marks around each pattern and used estimates of their centroids to determine the center of the pattern. Figure 7 shows high resolution scans of the output of an HP LaserJet 4M printer for the bitmap specification in Figure 7(a). Note that the 600 dpi pattern of Figure 7(c) has been magnified by two so that it is the same size as the 300 dpi pattern of Figure 7(b). Note also that at 300 dpi, it appears that the circular dot-overlap model may provide a reasonable approximation of the printer, but it would be hard to justify the same model at 600 dpi. More importantly, Baqai and Allebach [24] found that the 600 dpi patterns are not as stable (repeatable) as the 300 dpi patterns, which necessitates the use of a stochastic printer model. The tabular model they proposed incorporates the mean and variance of each dot combination.



0	0	0	0	0
0	1	0	1	0
0	0	1	0	0
0	0	0	0	0

▲ *7. (a) Digital halftone pattern and 4,000 dpi scan of its printed versions obtained at (b) 3,000 dpi and (c) 600 dpi (magnified by two).*



▲ *8. Offset centering for calculating the sampled grayscale printer model.*

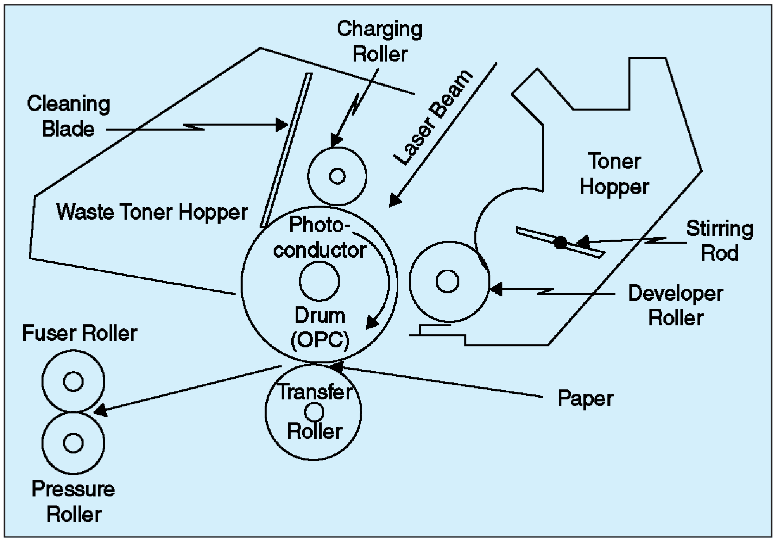
Offset-Centered Model

We described SGPMs that predict the gray level of each pixel site in the printing lattice, that is, they produce grayscale images that predict the output of the printer on the same lattice as the printing lattice. However, there are other possibilities. For example, as proposed by Wang et al. [33] and illustrated in Figure 8, instead of predicting the gray level of each printer pixel site, the SGPM could predict the gray level at pixel sites centered at the corners of the printer pixels. In other words, the lattice on which the SGPM is based could have an offset relative to the printing lattice. An advantage of this is that now each printer model pixel has only four neighboring printer pixels, rather than the eight neighboring printer pixels when there is no offset. Thus a 2×2 tabular model that accounts for the effects of the nearest pixels need only have $2^4=16$ entries, instead of $2^9=512$ for a 3×3 model. This is a considerable savings. Of course, the 2×2 model might not be quite as accurate as the 3×3 model. However, we now see that in addition to the 3×3, 5×5, 7×7, ... neighborhoods, we can add 2×2, 4×4, ... neighborhoods to our toolbox.

Model for Electrophotographic Printers

Previously, we described a tabular printer model that is based on the SGPM. We also described a method for parameterizing the model based on macroscopic or microscopic measurements. Here we show how these parameters can alternatively be obtained from a more analytic, physics-based characterization of an EP printer. This is especially important for EP printers that utilize pulse-width modulation, since in this case the number of possible pixel states is too large to permit use of a completely measurement-based approach to parameterize the model.

The EP process has three main steps [34], which can be understood by referring to Figure 9. The charged organic photoconductor (OPC) is exposed to a modulated light source, which typically is either a scanned laser beam or a 1-D array of light emitting diodes; 2) the exposed regions attract charged toner particles to the surface of the photoconductor; and, finally, 3) the toner particles on the photoconductor are transferred and fused to the paper to get the desired output. Laser EP printers can provide pixel modulation by either turning on the laser beam for less than the full width of the pixel or by keeping the beam on for the full width of the pixel, but varying the beam intensity. Analytical characterization of the EP process can be used within a halftoning algorithm to yield improved textures [35]. Our discussion here follows that in [35].



▲ *9. Cross section of the toner cartridge for a typical EP laser printer.*

The relationship between exposure on the OPC and the resulting charge is linear. The laser beam can be represented by a spatially varying intensity profile $I(x,y)$ that is assumed to be centered at the origin. To write a dot at the (i,j)th pixel, the laser beam is turned on according to the temporal switching waveform $0 \le I^0_{ij}(t) \le 1$, as it is scanned across that pixel. Assuming that the spacing between pixels in the horizontal and vertical directions is T_x and T_y, respectively, and the laser beam is scanned with velocity V, the total exposure at each point (x,y) due to the writing of the (i,j)th pixel can be expressed as

$$E_{i,j}(x,y) = \int I^0_{ij}(t) I(x - iT_x - Vt, y - jT_y)dt. \tag{10}$$

Typically, $I(x,y)$ is Gaussian, and $I^0_{ij}(t)$ is a rising exponential during the turn-on phase and a decaying exponential during the turn-off phase. Assuming that both exponentials have the same time constant, we can find the total exposure $E(x,y)$ on the OPC by simply summing over all the pixels, i.e.,

$$E(x,y) = \sum_{i,j} E_{i,j}(x,y). \tag{11}$$

Unfortunately, the remainder of the EP process is not nearly so easy to characterize. The manner in which toner particles are attracted to the OPC is an extremely complex and highly stochastic process. Transfer of the toner from the OPC to the paper and subsequent fusing of the toner to the paper introduces further changes to the latent image. In lieu of analytical descriptions for these parts of the EP process, an empirical approach can be taken in which the relation between exposure $E(x,y)$ at any point on the OPC and the absorptance $u(x,y)$ at the corresponding point in the final print is modeled by a simple point-to-point transformation $\tau(\cdot)$; so $u(x,y) = \tau(E(x,y))$.

The transformation τ is obtained in two steps. In the first step, the relation between a constant potential on the OPC and the resulting absorptance on the paper is determined. This is done by printing and measuring the absorptance of test patches with constant pulse width, which due to the spread of the Gaussian beam results in nearly constant exposure on the OPC. In the second step, this relationship is modified to account for halftone texture dependent factors such as dot gain discussed earlier. This step is accomplished by measuring the absorptance from constant tone patches that have been rendered with the target halftoning algorithm.

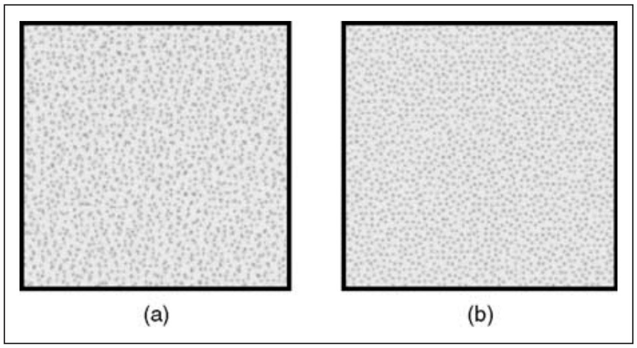
Once the transformation τ has been obtained, we have an analytic expression for the absorptance (x,y) at any point on the paper due to the underlying digital halftone image $b_{i,j}$, which in this case might not be binary. To parameterize the tabular model, the absorptance is calculated in an $M \times M$ grid of points within each $T_x \times T_y$ pixel of the SGPM, and this is used to compute $p_{i,j}$ in (8).

Model for Ink-Jet Printers

Ink-jet devices print by passing a printhead containing many small nozzles over the paper and ejecting drops of ink from the nozzles. The print head may contain nozzles for one or more colorants, for example, cyan, magenta, and yellow; and each colorant is typically assigned two columns of nozzles; so for a three-color print head, there will be a total of six columns of nozzles. Compared to EP printers, ink-jet printers render dots that have a greater integrity than those shown in Figure 7 and are much more stable. However, ink-jet printers exhibit dot displacement errors and dot irregularity that are caused by misaligned nozzles in the printhead, the dynamics of the carriage motion and drop formation, and turbulence in the air between the print head and the paper surface. The displacement errors that result from these sources appear largely random in nature. Ink-jet printers support various print options such as different print resolutions, speeds, directions (uni/bi), number of printing passes, and number of ink drops at each pixel. The artifacts that result are very dependent on the print mode. In general, slower printing modes will result in less visible artifacts. Multipass print modes are usually employed to reduce printing artifacts [36]. In a multiple-pass mode, the pen visits each pixel more than once and puts a drop there during a certain pass. A binary array called the print mask is used to control this process.

For certain print modes, dot displacement errors may cause significant artifacts in the printed halftone texture. For example, when an HP DeskJet 970 printer is operated in a 600 dpi, 10 in/s unidirectional mode, veining artifacts such as those shown in Figure 10(a) will occur. These artifacts can be traced to the fact that the dots printed in even-numbered rows are displaced to the right with a bias of 0.25 pixels, whereas those printed in odd-numbered rows are displaced to the left by 0.25 pixels. This is indicated by Figure 11, which shows the histograms of the horizontal dot displacement for even- and odd-numbered rows. These displacement biases are a consequence of the print head construction. Figure 11 also shows that the dot displacements for this particular print mode are quite random, being approximately uniformly distributed over an interval of width equal to the pixel dimension.

To eliminate these artifacts [37], one can again turn to the SGPM and tabular model discussed earlier. A dot overlap model similar to that discussed earlier is used to account for the effect of horizontal dot displacement. However, the stochastic nature of these displacements adds an additional level of complexity to the model and intimately links this modeling stage to the error metric employed with the halftoning algorithm. Since this topic has not yet been discussed, it will suffice to say here that for an LSMB approach like DBS [22], the expectation of the mean-squared error is taken with respect to the ensemble of all possible dot displacement configurations given the fixed digital halftone image. For tone-dependent error diffusion (TDED) [38], at each gray level, an estimate of the power spectrum of the printed halftone is computed and the parameters of the algorithm are chosen for that gray level to minimize the difference between this spectrum and an ideal halftone power spectrum based on DBS textures. Figure 10(b) illustrates the improvement that results.



▲ 10. Constant gray patch halftoned using TDED (a) with ideal model and (b) with inkjet model discussed here, and printed with an HP DeskJet 970 printer operated in the 600 dpi, 10 in/s, unidirectional print mode.

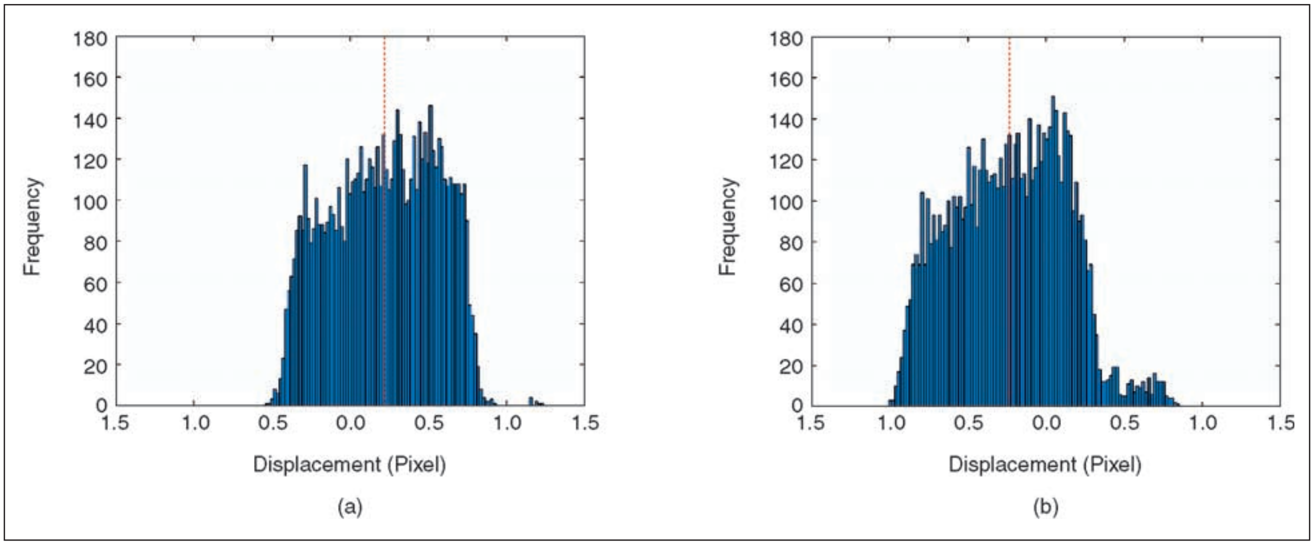
Model-Based Halftoning Algorithms

LSMB Halftoning with DBS

Given accurate models for the HVS and display device, we now consider algorithms that maximize the visual quality of the displayed images using a fidelity metric. In the LSMB approach this takes the form of minimizing the squared error between the output of the cascade of the printer and visual models in response to the halftone image specification and the output of the visual model in response to the original continuous-tone image. As we will see below, the optimal solution, and in this sense the "optimal" halftone reproduction, can only be obtained by iterative techniques that in effect search through the space of all possible halftone images and pick the one that minimizes the error criterion. Since the typical halftone specification is binary, we refer to this process as DBS. The overall approach has thus been known as either the DBS [14], [22], [24], [35] or the LSMB approach [9]-[11]. Similar approaches (not including a display model) have been proposed in [39]-[42].

We now look at a more detailed formulation of the problem. Let $[z_{i,j}]$ denote a continuous-tone image. As illustrated in Figure 12, the LSMB approach seeks the halftone image $[b_{i,j}]$ that minimizes the squared error

$$E = \sum_{i,j} \left(\tilde{z}_{i,j} - \tilde{p}_{i,j} \right)^2 \tag{12}$$



▲ 11. Horizontal dot displacement for pixels in (a) even and (b) odd rows for an HP DeskJet 970 printer operated in the 600 dpi, 10 in/s, unidirectional print mode.

where

$$\tilde{z}_{i,j} = z_{i,j} * g_{i,j} \tag{13}$$

$$\tilde{p}_{i,j} = p_{i,j} * g_{i,j} = \mathcal{P}\left(W_{i,j}\right) * g_{i,j}. \tag{14}$$

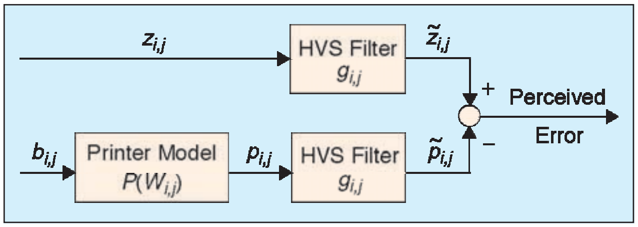
Here $W_{i,j}$ consists of $b_{i,j}$ and its neighbors as in (8), and $*$ indicates convolution. The boundary conditions assume that no colorant is placed outside the image borders.

In principle, the optimal solution can be obtained by an exhaustive search over all possible binary patterns for the entire image. This approach is not computationally feasible, however. The number of possible patterns for an $N_x \times N_y$ image is $2^{N_x N_y}$ (e.g., 1.16×10^{77} just for a 16×16 image). Thus, the least-squares solution must be obtained by iterative optimization techniques. Such techniques find a solution that is only a local optimum. They assume that an initial estimate of the binary image $[b_{i,j}]$ is given. This could be a trivial image, e.g., a constant or random image, or the output of any halftoning algorithm. Depending on the optimization strategy, the visual quality of the resulting halftone image may be influenced by this starting point.

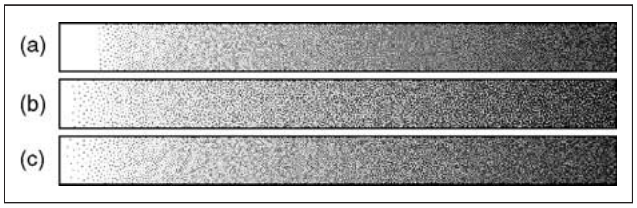
A search strategy that has been found to work very well traverses the image in a raster scan. At each pixel, we consider toggling the state of that pixel from 0 to 1 or 1 to 0, whichever is appropriate, and also swapping the state of that pixel with the state of any of its eight nearest neighbors, if they are different. Among these possible nine changes, we retain that single change, if any, that reduces the overall error (12) the most. When the HVS filter has finite impulse response, the binary value of each pixel affects only the model outputs $\tilde{p}_{k,l}$ in its neighborhood, and thus the error need only be computed locally. Here we should mention that the error value should not be computed from scratch each time; the value of $\tilde{z}_{i,j}$ never changes and only a few terms in the convolution sum that determines $\tilde{p}_{i,j}$ must be updated (those that involve the current pixel and its neighbors). By exploiting the bilinearity of the error metric, we can achieve a dramatic reduction in the cost of evaluating trial changes from $O(\Omega(g_{i,j}))$ arithmetic operations, where $\Omega(g_{i,j})$ is the number of pixels in the support of the filter $g_{i,j}$, to a handful of arithmetic operations, independent of $\Omega(g_{i,j})$ [22]. An iteration is complete when the minimization is performed once at each image pixel. The number of iterations depends on the starting point and the effective filter width. The resulting halftones, however, are practically independent of the starting point. More sophisticated (and computationally intensive) schemes use simulated annealing [42], [43] but have not yet shown any significant improvements in image quality.

Figure 13(a) and (b) shows a magnified detail of a grayscale ramp halftoned with LSMB using HVS filters corresponding to viewing distances of 0.5 and 2 ft at 300 dpi. Note that, as predicted earlier, as the viewing distance increases, the textures become coarser. If each image is viewed at the right viewing distance, the number of perceived gray levels should increase with coarser textures (which become less visible as the actual viewing distance increases). An immediate consequence of this observation is the fact that the white regions at the left of the ramps grow as the viewing distance decreases. This is an artifact directly related to the coarseness of the grayscale. Specifically, when the dots are farther apart than the eye can average, the LSMB metric finds it better to produce no dots to represent a nearly white region, rather than placing a few widely separated dots. Figure 13(c) shows the ramp halftoned with the dual-metric DBS technique [23] that we discussed earlier. This approach effectively changes the viewing distance to improve the halftone patterns in different parts of the image.

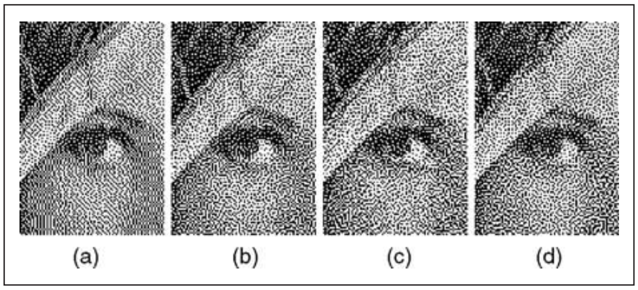
Figure 14 shows a magnified detail (shown at 100 dpi) of an image halftoned with (a) the Floyd/Steinberg error diffusion [44], (b) the TDED [38], (c) the dual-metric DBS screening technique [23], [45], and (d) the dual-metric DBS technique [23]. Observe the superior patterns of the dual-metric DBS, followed by TDED, Floyd/Steinberg, and the screening technique. Note that



▲ 12. LSMB halftoning.



▲ 13. LSMB/DBS halftoning for different viewing distances at 300 dpi (shown magnified by three): (a) 0.5 ft, (b) 2 ft, and (c) dual-metric DBS.



▲ 14. Different halftoning techniques shown at 100 dpi: (a) Floyd/Steinberg error diffusion, (b) TDED, (c) dual-metric DBS screen, and (d) dual-metric DBS.

the latter is the most constrained, while Floyd/Steinberg exhibits well-known artifacts.

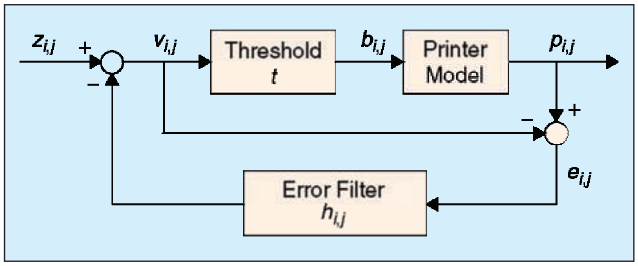
Model-Based Error Diffusion

The error-diffusion algorithm [44] is a neighborhood technique that produces sharper images than point (screening) techniques and generates visually pleasant textures. However, as we discussed earlier, it is very sensitive to dot-overlap and other printer distortions. Therefore, it is necessary to incorporate a printer model in the algorithm. In addition, we consider the use of HVS models in error diffusion. The basics of error diffusion and a number of fundamental issues are discussed in the Lau et al. and Eschbach et al. articles in this issue, while fundamentals of color error diffusion are discussed in the Damera-Venkata et al. article.

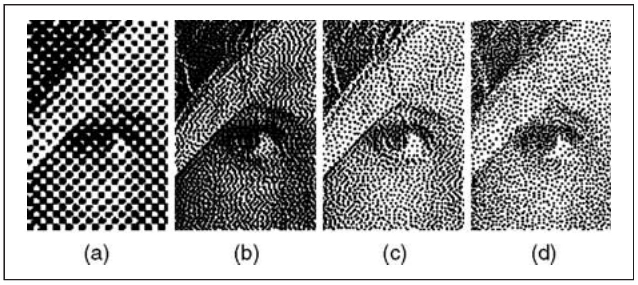
Printer Model-Based Error Diffusion

The main idea behind error diffusion is very simple. As illustrated in Figure 15 it keeps track of "past" quantization errors and compensates for them when it quantizes the next pixel value. If we ignore the printer model in Figure 15, the $e_{i,j}$s represent the quantization errors, and compensating for them is accomplished by subtracting a filtered version of the errors from the image values $z_{i,j}$. Equally simple is the extension of error diffusion to include a printer model. In addition to quantization errors, the algorithm must take into account the printer effects, as shown in Figure 15.

Stucki [29], [30] was the first to suggest the use of a dot-overlap model to account for printer effects in error diffusion. At each pixel in the image, Stucki's algorithm accounts for the newly placed colorant. While this scheme produces the correct tone scale, it also results in a loss of sharpness. That's because part of the newly placed colorant may be outside the pixel boundaries, while some previously placed colorant may be inside the pixel boundaries. In effect, this increases the cell size of the SGPM, thus resulting in a loss of spatial resolution. Pappas and Neuhoff [26], [27] used the dot-overlap model of (9), which accounts only for the colorant within the pixel boundaries. Thus, it preserves the sharpness of the original error-diffusion algorithm. They referred to the resulting algorithm as modified error diffusion (MED).



▲ *15. Modified error diffusion.*



▲ *16. (a) "Classical" screen, (b) error diffusion (no printer model), (c) MED, and (d) blue-noise screen.*

A block diagram of the MED algorithm is shown in Figure 15. Without loss of generality, we assume that the image is scanned left to right, top to bottom. The binary image $[b_{i,j}]$ is obtained by thresholding a "corrected" value $v_{i,j}$ of the grayscale image. The MED algorithm uses a printer model to estimate the gray level $p_{i,j}$ of the printed pixels. The difference between this gray level and the "corrected" grayscale image is defined as the error $e_{i,j}$ at the location (i,j). Previous errors are filtered and subtracted from the current image value $z_{i,j}$ to obtain the "corrected" value of the grayscale image. The threshold t is typically fixed at 0.5. The MED equations are

$$v_{i,j} = z_{i,j} - \sum_{m,n} h_{m,n}\, e^{i,j}_{i-m,j-n} \qquad (15)$$

$$b_{i,j} = \begin{cases} 1, & \text{if } v_{i,j} > t \\ 0, & \text{otherwise} \end{cases} \qquad (16)$$

$$e^{i,j}_{m,n} = p^{i,j}_{m,n} - v_{m,n} \quad \text{for } (m,n) < (i,j) \qquad (17)$$

where $(m,n)<(i,j)$ means (m,n) precedes (i,j) in the scanning order and

$$p^{i,j}_{m,n} = \mathcal{P}(W^{i,j}_{m,n}) \quad \text{for } (m,n) < (i,j) \qquad (18)$$

where $W^{i,j}_{m,n}$ consists of $b_{m,n}$ and its neighbors as in (8). Here the neighbors $b_{k,l}$ have been determined only for $(k,l)<(i,j)$; they are assumed to be zero (i.e.,white) for $(k,l)\geq(i,j)$. Since only the dot-overlap contributions of the previous pixels can be used in (18), the previous errors keep getting updated as more binary values are computed. This is why the error and the printer model output depend on the location (i,j). The assumption that the undetermined pixels are white leads to a bias in the grayscale of the printed image. This bias is very small and difficult to detect in practice and can be eliminated by the multipass version of the MED algorithm [32].

Figure 16 shows magnified details of halftone images as they would be printed on a 300 dpi printer, if it obeyed the circular dot-overlap model with $\rho=1.25$. They are reproduced with simulated dot gain at one third of the printer resolution. In particular, it shows (a) the "classical" clustered-dot screening technique without any compensation for printer effects, (b) error diffusion with a Jarvis-Judice-Ninke filter [46] and no printer model, (c) MED with a Jarvis-Judice-Ninke filter and a circular dot-overlap model with $\rho=1.25$, and (d) a blue-noise screening (BNS) technique with the same printer model [43]. Observe that without a printer model, the error diffusion results in noticeable grayscale distortion (darkening), while the "classical" screen is very robust to printer effects. Observe also that the MED image is sharper and has better texture than either of the screening techniques. In particular, the texture that the BNS produces is considerably grainier than that of the MED.

Tone-Dependent Error Diffusion

Recently, a number of researchers have considered varying the weights in the error-diffusion kernel as a function of the gray level of the continuous-tone input image to improve the control of the texture at each gray level [38], [47]-[49]. Ostromoukhov [49] trained his weights to yield error-diffusion textures that have blue noise spectra. In addition to making the weights tone dependent, Li and Allebach [38] used a serpentine raster and replaced the threshold t in Figure 15 by two tone-dependent thresholds $t_z^l \geq t_z^u$, where z denotes the gray level. If the input z to the threshold step is less than t_z^l, the output is 0. If it is greater than t_z^u, the output is 1. If z falls between t_z^l and t_z^u, the binary output is taken from a binary texture patch with absorptance 0.5 generated by DBS [22]. They trained all the parameters of the algorithm to minimize a cost function based on an HVS model. In the highlights and shadows, this cost function is given by (12). In the midtones, they found that this cost function did not result in textures with satisfactory homogeneity and texture variety; so they replaced it with the total squared error between the power spectrum of the halftone generated by TDED and that generated by DBS. So here the cost function is indirectly linked to an HVS model via the power spectrum of the DBS halftone. In general, TDED can yield halftone textures that have almost the same level of quality as DBS. Because of the large number of degrees of freedom that TDED possesses, it is well suited to use with a printer model. As discussed earlier, Lee and Allebach [37] have based the training of the parameters on an inkjet printer model and successfully eliminated very visible artifacts that were directly due to the printer mechanism.

Other Halftoning Algorithms

Model-Based Screening and Lookup Table Halftoning

In screening, the binary image is generated by comparing each pixel of a continuous-tone image to an array of image-independent thresholds [50]. The binary image is black when the gray level of the image pixel is greater than the corresponding threshold and white otherwise. The thresholds can be generated randomly (random dither) or can be periodic (ordered dither). The main advantage of screening techniques is that the required amount of computation is minimal and can be carried out in parallel.

Traditional screening techniques make use of the properties of the eye only implicitly. The "classical" clustered-dot screen has been the most popular for printing because of its robustness to printer distortions and its similarity to traditional analog halftoning techniques. Dispersed-dot screening techniques produce images with better spatial resolution and better texture than clustered-dot techniques but are more sensitive to printer distortions.

Dispersed-dot screens can be designed by using any dispersed-dot halftoning algorithm to generate halftone textures for each constant gray level between zero and 255. To be implementable by thresholding, these textures must obey a stacking constraint so that once a black dot is turned on at a given pixel location, that dot will remain turned on for all darker gray levels. To minimize the visibility of the fundamental period in the halftone patterns and to enable the design of higher quality textures, it is common to use screens that are much larger than the 16×16 minimum that would be required to generate 256 gray levels. Sizes of 128×128 or 256×256 are typical. A number of approaches have been used to design the binary textures at each gray level. Some of these methods use an explicit model for the HVS and directly minimize an error metric [7], [43] [45], [51]. Others attempt to force the spectrum of the halftone textures to have a blue noise characteristic [52] or to eliminate voids and clusters in the halftone texture [53]. With this latter method, the Gaussian filter used to identify the largest void and the tightest cluster can be interpreted as the point spread function of an HVS. It is also possible to generalize the screening concept to a model-based lookup-table-based approach that can yield some improvement over screening [21].

Printer models can be used with screening techniques to account for printer distortions. They can be used to modify the thresholds of an already designed screen. For example, this was done by Roetling and Holladay [25], who used a circular dot-overlap model. In addition, they also used the printer model to optimize screen design. In [43], Schulze and Pappas used the circular dot-overlap printer model described above to optimize the design of BNSs using the void-and-cluster method. The images generated using such model-based BNSs are very similar in appearance to those generated by BNSs with modified thresholds. Figure 16(a) and (d) compares a "classical" screen and a model-based BNS, respectively. Note that the "classical" screen is fairly robust to printer distortions (no printer compensation was used), while the model-based BNS is sharper and has better texture.

AM/FM Halftoning

The halftoning algorithms we have discussed so far achieve a given level of absorptance by either modulating the size of dots that are placed on a fixed lattice of points or by modulating the spacing or density of dots that have fixed size. The best example of the first group of methods is the traditional clustered dot screen. In the second group, we have error diffusion and LSMB halftoning. Algorithms in the first group are more stable for printing with EP printers, whereas algorithms in the second group provide better detail resolution. The AM/FM algorithm [54] combines both modes of modulation to yield a tradeoff between stability and detail resolution that is optimized for a given EP print mechanism. It is especially well suited for EP printers with pixel modulation capability.

The basic idea of AM/FM halftoning is that each intended gray level is reproduced by a prespecified macrodot density and macrodot size. For example, assuming no dot gain, to attain gray level 0.2, the density could be 0.1 and the size might be 2, meaning that macrodots consisting of two adjacent printer dots are placed at 10% of the printer lattice sites. Alternatively, the density might be 0.4 and the macrodot size might be 0.5, indicating that macrodots are placed at 40% of the sites, each with half the usual size. This basic idea is implemented with some dispersed dot halftoning method, such as tone-dependent error diffusion, along with two point-to-point nonlinear mappings, stored as lookup tables. For each gray level, one table determines the density and the other determines the macrodot size. The AM/FM algorithm operates by first applying the density mapping to the image, with the effect of reducing gray levels that are to be reproduced with macrodot sizes greater than one, and increasing gray levels corresponding to macrodot sizes less than one. Next, the dispersed dot halftoning is applied, which, at least approximately, produces the desired macrodot density for each gray level of the image. Finally, for each pixel (i,j) at which there is a macrodot, the original image gray level $z_{i,j}$ is used to address the size table to determine the size of the macrodot at this location.

There are obviously many combinations of density and size that can attain each gray level. In [54], the tables were designed by using the AM/FM algorithm to generate halftone patches with every possible combination of macrodot size and density, and then measuring the average absorptance and visually weighted mean-squared error of these patches. The dot size and density mappings were found using a multiresolution iterative coordinate descent algorithm, with a cost function that was regularized to penalize abrupt changes in dot size or density as a function of input gray level (which might cause false contouring).

For EP printers, the AM/FM algorithm produces more stable and higher quality halftones than conventional error diffusion, while providing better detail rendition than clustered dot screens. It is particularly useful for scan-to-print applications where the susceptibility of clustered dot screens to moire in scanned material that contains periodic halftone dot patterns is unacceptable.

Thrasyvoulos N. Pappas received the S.B., S.M., and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1979, 1982, and 1987, respectively. From 1987 until 1999, he was a Member of the Technical Staff at Bell Laboratories, Murray Hill, New Jersey. In 1999, he joined the Department of Electrical and Computer Engineering at Northwestern University as an associate professor. His research interests are in image and multidimensional signal processing. He is a Senior Member of the IEEE. He is electronic abstracts editor and associate editor for *IEEE Transactions on Image Processing*. He is chair of the IEEE Image and Multidimensional Signal Processing Technical Committee and a member of the Multimedia Signal Processing Technical Committee. He served as Technical Program cochair for ICIP-2001 in Thessaloniki, Greece. He is also cochair for the IS&T/SPIE Conference on Human Vision and Electronic Imaging.

Jan P. Allebach received his B.S.E.E. from the University of Delaware in 1972 and his Ph.D. from Princeton University in 1976. He was on the faculty at the University of Delaware from 1976 to 1983. Since 1983, he has been at Purdue University in the School of Electrical and Computer Engineering where he is Michael J. and Katherine R. Birck Professor of Electrical and Computer Engineering. His current research interests include image rendering, image quality, color imaging and color measurement, document management, and wireless application of imaging and printing. He is a member of the IEEE Signal Processing Society, the Society for Imaging Science and Technology (IS&T), and SPIE. He is a Fellow the IEEE and IS&T and has served as distinguished/visiting lecturer and as an officer and on the Board of Directors of both societies. He was an associate editor for *IEEE Transactions on Signal Processing* and *IEEE Transactions on Image Processing*. He is currently editor for the *IS&T/SPIE Journal of Electronic Imaging*. He received the Senior (best paper) Award from the IEEE Signal Processing Society and the Bowman Award from IS&T.

David L. Neuhoff received the B.S.E. from Cornell University in 1970 and the M.S. and Ph.D. in electrical engineering from Stanford University in 1972 and 1974, respectively. In 1974, he joined the University of Michigan, Ann Arbor, where he is now professor of electrical engineering and computer science. He spent a sabbatical at Bell Laboratories, Murray Hill, New Jersey. His research and teaching interests are in communications, information theory, and signal processing. He is a Fellow of the IEEE. He was associate editor for *IEEE Transactions on Information Theory* and served on the Board of Governors of the IEEE Information Theory Society. He cochaired several meetings, including the 1986 IEEE International Symposium on Information Theory.

References

[1] J.P. Allebach, Ed., *Selected Papers on Digital Halftoning*, vol. MS 154. SPIE: Bellingham, WA, 1999.

[2] M.P. Eckert and A.P. Bradley, "Perceptual quality metrics applied to still image compression," *Signal Processing*, vol. 70, pp. 177-200, 1998.

[3] T.N. Pappas and R.J. Safranek, "Perceptual criteria for image quality evaluation," in *Handbook of Image and Video Processing*, A.C. Bovik, Ed. New York: Academic, 200, pp. 669-684.

[4] J.L. Mannos and D.J. Sakrison, "The effects of a visual fidelity criterion on the encoding of images," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 525-536, July 1974.

[5] A.K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[6] S. Daly, "Subroutine for the generation of a two dimensional human visual contrast sensitivity function," Eastman Kodak, Rochester, NY, Tech. Rep. 233203Y, 1987.

[7] J. Sullivan, L. Ray, and R. Miller, “Design of minimum visual modulation halftone patterns,” *IEEE Trans. Syst., Man, Cybern.*, vol. 21, pp. 33-38, Jan./Feb. 1991.

[8] B.W. Kolpatzik and C.A. Bouman, “Optimized error diffusion for image display,” *J. Electron. Imaging*, vol. 1, pp. 277-292, July 1992.

[9] T.N. Pappas and D.L. Neuhoff, “Least-squares model-based halftoning,” in *Proc. SPIE, Human Vision, Visual Proc., and Digital Display III,* San Jose, CA, Feb. 1992, vol. 1666, pp. 165-176.

[10] D.L. Neuhoff, T.N. Pappas, and N. Seshadri, “One-dimensional least-squares model-based halftoning,” *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 14, pp. 1707-1723, Aug. 1997.

[11] T.N. Pappas and D.L. Neuhoff, “Least-squares model-based halftoning,” *IEEE Trans. Image Processing*, vol. 8, pp. 1102-1116, Aug. 1999.

[12] J. Sullivan, R. Miller, and G. Pios, “Image halftoning using a visual model in error diffusion,” *J. Opt. Soc. Am. A*, vol. 10, pp. 1714-1724, Aug. 1993.

[13] T. Mitsa and K. Vakur, “Evaluation of contrast sensitivity functions for the formulation of quality measures incorporated in halftoning algorithms,” in *Proc. ICASSP-93*, Minneapolis, MN, Apr. 1993, vol. V, pp. 301-304.

[14] M. Analoui and J.P. Allebach, “Model based halftoning using direct binary search,” in *Proc. SPIE, Human Vision, Visual Proc., and Digital Display III*, San Jose, CA, Feb. 1992, vol. 1666, pp. 96-108.

[15] B.A. Wandell, *Foundations of Vision*. Sunderland, MA: Sinauer, 1995.

[16] A.N. Netravali and B.G. Haskell, Eds., *Digital Pictures: Representation and Compression.* New York: Plenum, 1988.

[17] A.B. Watson, “The cortex transform: Rapid computation of simulated neural images,” *Comput.Vis., Graph., Image Process.*, vol. 39, pp. 311-327, 1987.

[18] S. Daly, “The visible differences predictor: an algorithm for the assessment of image fidelity,” in *Digital Images and Human Vision*, A.B. Watson, Ed. Cambridge, MA: MIT Press, 1993, pp. 179-206.

[19] D.J. Heeger and P.C. Teo, “A model of perceptual image fidelity,” in *Proc. ICIP-95*, Washington, DC, Oct. 1995, vol. II, pp. 343-345.

[20] J. Lubin, “The use of psychophysical data and models in the analysis of display system performance,” in *Digital Images and Human Vision*, A.B. Watson, Ed. Cambridge, MA: MIT Press, 1993, pp. 163-178.

[21] P. Li and J.P. Allebach, “Look-up-table based halftoning algorithm,” *IEEE Trans. Image Processing*, vol. 9, pp. 1593-1603, Sept. 2000.

[22] D.J. Lieberman and J.P. Allebach, “A dual interpretation of direct binary search and its implications for tone reproduction and texture quality,” *IEEE Trans. Image Processing*, vol. 9, pp. 1950-1963, Nov. 2000.

[23] S.H. Kim and J.P. Allebach, “Impact of HVS models on model-based halftoning,” *IEEE Trans. Image Processing*, vol. 11, pp. 258-269, Mar. 2002.

[24] F.A. Baqai and J.P. Allebach, “Halftoning via direct binary search using analytical and stochastic printer model,” *IEEE Trans. Image Processing*, vol. 12, pp. 1-15, Jan. 2003.

[25] P.G. Roetling and T.M. Holladay, “Tone reproduction and screen design for pictorial electrographic printing,” *J. Appli. Phot. Eng.*, vol. 15, no. 4, pp. 179-182, 1979.

[26] T.N. Pappas and D.L. Neuhoff, “Model-based halftoning,” in *Proc. SPIE, Human Vision, Visual Proc., and Digital Display II*, San Jose, CA, Feb. 1991, vol. 1453, pp. 244-255.

[27] T.N. Pappas and D.L. Neuhoff, “Printer models and error diffusion,” *IEEE Trans. Image Processing*, vol. 4, pp. 66-80, Jan. 1995.

[28] J.P. Allebach, “Binary display of images when spot size exceeds step size,” *Appl. Opt.*, vol. 19, pp. 2513-2519, Aug. 1980.

[29] P. Stucki, “MECCA—A multiple-error correcting computation algorithm for bilevel image hardcopy reproduction,” IBM Research Laboratory, Zurich, Switzerland, Res. Rep. RZ1060, 1981.

[30] P. Stucki, “Advances in digital image processing for document reproduction,” in *VLSI Engineering*, T. L. Kunii, Ed. Tokyo: Springer-Verlag, 1984, pp. 256-302.

[31] R.L. Stevenson and G.R. Arce, “Binary display of hexagonally sampled continuous-tone images,” *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 2, pp. 1009-1013, July 1985.

[32] T.N. Pappas, C.-K. Dong, and D.L. Neuhoff, “Measurement of printer parameters for model-based halftoning,” *J. Electron. Imaging*, vol. 2, pp. 193-204, July 1993.

[33] S.-G. Wang, K.T. Knox, and N. George, “Novel centering method for overlapping correction in halftoning,” in *Proc. IS&T’s 47th Annu. Conf.*, Rochester, NY, May 15-20, 1994, pp. 482-486.

[34] L.B. Schien, *Electrophotography and Development Physics*, 2nd ed. Morgan Hill, CA: Laplacian, 1996.

[35] D. Kacker, T. Camis, and J.P. Allebach, “Electrophotographic process embedded in direct binary search,” *IEEE Trans. Image Processing*, vol. 11, pp. 243-257, Mar. 2002.

[36] J. Yen, M. Carlsson, M. Chang, J.M. Carcia, and H. Nguyen, “Constraint solving for inkjet print mask design,” *J. Imaging Sci. Technol.*, vol. 44, pp. 391-397, Sept./Oct. 2000.

[37] J.-H. Lee and J.P. Allebach, “Inkjet printer model-based halftoning,” in *Proc. ICIP-02*, Rochester, NY, Sept. 2002, vol. I, pp. 461-464.

[38] P. Li and J.P. Allebach, “Tone-dependent error diffusion,” in *Proc. SPIE, Color Imaging: Device Independent Color, Color Hardcopy, and Applications VII*, San Jose, CA, Jan. 2002, vol. 4663, pp. 310-321.

[39] D. Anastassiou and S. Kollias, “Digital image halftoning using neural networks,” in *Proc. SPIE, Visual Communications and Image Processing*, vol. 1001, pp. 1062-1069, 1988.

[40] D. Anastassiou, “Error diffusion coding for A/D conversion,” *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1175-1186, Sept. 1989.

[41] A. Zakhor, S. Lin, and F. Eskafi, “A new class of B/W halftoning algorithms,” *IEEE Trans. Image Processing*, vol. 2, pp. 499-509, Oct. 1993.

[42] J.B. Mulligan and A.J. Ahumada, Jr., “Principled halftoning based on models of human vision,” in *Proc. SPIE, Human Vision, Visual Proc., and Digital Display III,*, San Jose, CA, Feb. 1992, vol. 1666, pp. 109-121.

[43] M.A. Schulze and T.N. Pappas, “Blue noise and model-based halftoning,” in *Proc. SPIE, Human Vision, Visual Proc., and Digital Display V*, San Jose, CA, Feb. 1994, vol 2179, pp. 182-194.

[44] R.W. Floyd and L. Steinberg, “An adaptive algorithm for spatial grey scale,” in *Proc. Society for Information Display*, 1976, vol. 17/2, pp. 75-77.

[45] J.P. Allebach and Q. Lin, “FM screen design using DBS algorithm,” in *Proc. ICIP-96*, Lausanne, Switzerland, Sept. 1996, vol. 1, pp. 549-552.

[46] J.F. Jarvis, C.N. Judice, and W.H. Ninke, “A survey of techniques for the display of continuous-tone pictures on bilevel displays,” *Comp. Graph. Image Proc.*, vol. 5, pp. 13-40, 1976.

[47] R. Eschbach, “Reduction of artifacts in error diffusion by means of input-dependent weights,” *J. Electron. Imaging*, vol. 2, pp. 352-358, Oct. 1993.

[48] J. Shu, “Adaptive filtering for error diffusion quality improvement,” in *SID Digest of Technical Papers*, Orlando, FL, May 1995, pp. 833-836.

[49] V. Ostromoukhov, “A simple and efficient error diffusion algorithm,” in *Proc. SIGGRAPH ‘01 28th Intl. Conf. Computer Graphics and Interactive Techniques*, Los Angeles, CA, Aug. 2001, pp. 567-572.

[50] R. Ulichney, *Digital Halftoning*. Cambridge, MA: MIT Press, 1987.

[51] J.R. Sullivan and L. Ray, “Digital halftoning with correlated minimum visual modulation patterns,” U.S. Patent 5 214 517, 1993.

[52] T. Mitsa and K.J. Parker, “Digital halftoning technique using a blue-noise mask,” *J. Opt. Soc. Am. A, Opt.Image Sci.*, vol. 9, pp. 1920-1929, Nov. 1992.

[53] R. Ulichney, “The void-and-cluster method for dither array generation,” in *Proc. SPIE, Human Vision, Visual Proc., and Digital Display IV*, San Jose, CA, Feb. 1993, vol. 1913, pp. 332-343.

[54] Z. He and C.A. Bouman, “AM/FM halftoning: Digital halftoning through simultaneous modulation of dot size and dot density,” in *Proc. SPIE, Color Imaging: Device Independent Color, Color Hardcopy, and Applications VII,*, San Jose, CA, Jan. 2002, vol. 4663, pp. 322-334.