AM/FM Halftoning: Digital Halftoning Through Simultaneous Modulation of Dot Size and Dot Density

Zhen He and Charles A. Bouman
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN, 47907-1285

January 26, 2004

Abstract

Conventional digital halftoning approaches function by modulating either the dot size (amplitude modulation or AM) or the dot density (frequency modulation or FM). Generally, AM halftoning methods have the advantages of low computation and good print stability, while FM halftoning methods typically have higher spatial resolution and resistance to Moiré artifacts.

In this paper, we present a new class of AM/FM halftoning algorithms that simultaneously modulate dot size and density. The major advantages of AM/FM halftoning are:

- Better stability than FM methods through the formation of larger dot clusters.
- Better Moiré resistance than AM methods through irregular dot placement.
- Improved quality through the systematic optimization of the dot size and dot density at each gray level.

We present a general method for optimizing the AM/FM method for specific printers, and we apply this method to an electro-photographic printer using pulse width modulation (PWM) technology.

1 Introduction

Conventionally, halftoning is accomplished by either changing the size of printed dots or changing the relative density of dots on the page. These two approaches are analogous to amplitude modulation (AM) or frequency modulation (FM) used in communications.

In AM halftoning, the density of dot clusters, which we define as the number of clusters per unit area, is fixed. Tone rendition is achieved by varying the size of each dot. The most commonly used AM halftoning algorithm is clustered dot screening [1]. Cluster dot screening has the advantages of low computational load, stable dot formation, and resistance to such printer artifacts as dot gain and banding. Thus, it is widely used in electro-photographic (laser) printers where a single isolated dot may not develop stably. One drawback of cluster dot screening is its limited ability to render fine detail. Moreover, the regular dot placement also makes it vulnerable to Moiré patterns when periodic patterns in the image are similar to the clustered dot frequency [2]. Thus, it is not suitable for halftoning images scanned from printed material.

In FM halftoning, dot size is fixed and the tone rendition is achieved by varying the dot density. Commonly used FM halftoning algorithms include disperse dot screening [3, 4, 5], error diffusion [6, 7, 8], and search-based halftone methods such as direct binary search (DBS) [9, 10, 11]. Error diffusion is perhaps the most popular FM halftoning algorithm. Though it requires more computation as compared to screening, it is still very efficient. In general, FM halftoning achieves higher spatial resolution than AM halftoning and is free of Moiré artifacts. However, it may lack of the print stability required for electro-photographic printing.

Levien proposed output-dependent error diffusion in [12], which increases the print stability of error diffusion through the formation of larger dot clusters. The spectrum of such methods has been characterized by Lau *et al.* as green noise [13], which lacks both low frequency and high frequency components. In [14], the authors also gave a proceedure for designing masks capable of producing green-noise halftone patterns. In [15], both output-dependent error diffusion and green-noise masks are generalized to color halftoning. Adaptive threshold modulation for green noise halftoning was also proposed in [16].

In this paper, we present a new class of halftoning algorithms which combine the advantages of both AM and FM halftoning methods. These methods, which we call AM/FM halftoning, are distinct from previous methods because they directly modulate the dot size and dot density to produce the best quality halftone at each gray level. Unlike the green-noise halftoning, AM/FM halftoning renders each gray level with a single fixed dot size optimized for this gray level. To halftone an image, AM/FM halftoning first determines the position of each dot using a dispersed dot halftoning algorithm and a dot density curve that relates the input gray level to the density of dots on the page. The size of each dot is then modulated according to a dot size curve. A measurement-based parameter design system is developed to optimize both the dot size and dot density curves for a particular printer.

A specific implementation of AM/FM halftoning is developed for use with electro-photographic printers having subpixel modulation ability such as Hewlett-Packard's pulse width modulation (PWM) technology [17, 18]. Dot size and density curves are designed via the minimization of a regularized cost function which takes both print quality and smooth texture transition into account. Experiments demonstrate that AM/FM halftoning achieves high spatial resolution, smooth highlight halftone textures, good printing stability, and Moiré resistance.

Section 2 develops the framework of AM/FM halftoning algorithm, and Section 3 presents a specific implementation of it based on error diffusion and dot pair clusters formed using PWM. Section 3.3 shows how dot size diffusion can eliminate quantization artifacts when the dynamic range of the PWM system is limited, and Section 4 presents experimental results.

2 AM/FM Halftoning Framework

In the following section, we present a general framework for AM/FM halftoning. Once we adopt this framework, we must then determine how to choose the dot size and dot density curves in a manner which produces the best quality rendering. We address this question in Section 2.1 by introducing a parameter design methodology which uses empirical measurements of printer response to optimize the AM/FM algorithm's behavior.

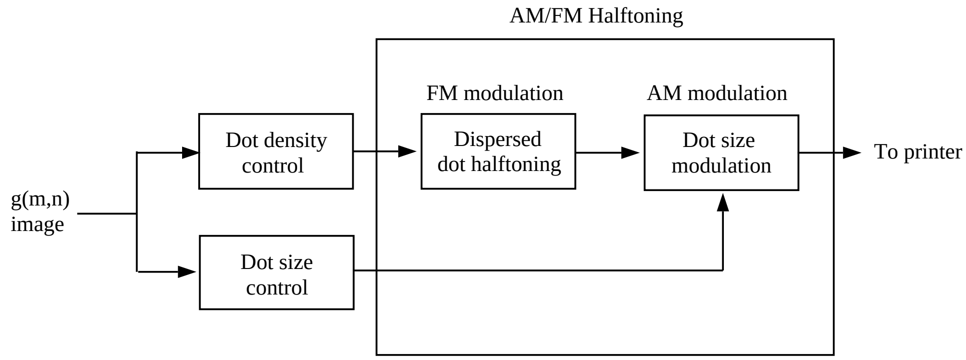


Figure 1: Diagram of the AM/FM halftoning algorithm. The dispersed dot halftoning algorithm determines the placement of dots, but in addition the size of each dot is independently modulated. The dot size and density are controlled using a pair of look-up tables.

Figure 1 illustrates how the AM/FM halftoning algorithm functions. Two look-up tables (LUT's) are first used to determine the dot density and dot size at each pixel. The dot density is then used as the input to a dispersed dot halftoning algorithm which determines the position of dots on a discrete printing grid. The size of each printed dot is then independently modulated based on the computed dot size at that position. The resulting output is then printed.

The essential attribute of AM/FM halftoning is that it simultaneously modulates both the dot density (i.e. spacing between dots) and the dot size. If the dot size is fixed to 1 everywhere, AM/FM halftoning degenerates to conventional disperse dot halftoning. Thus, AM/FM halftoning is a more general class of algorithms than dispersed dot halftoning. Importantly, neither the dot density nor the dot size independently control the printed tone. Instead the printed tone is controlled through a combination of the two. This allows one to incorporate tone correction directly into the dot density and size LUT's.

In general, the disperse dot halftoning algorithm can be any one of a wide variety of methods including error diffusion, dispersed dot screening, or iterative search based halftoning such as direct binary search (DBS) [9]. The specific method used to modulate dot size will, in general, depend on the particular printing technology. It may be any one of a number of methods which vary the size of a printed dot either by grouping clusters of dots together, or by directly modulating the size of the printed dot using a technique such as pulse width modulation (PWM) [18].

The dot density and dot size LUT's are critical parameters of any AM/FM halftoning algorithm.

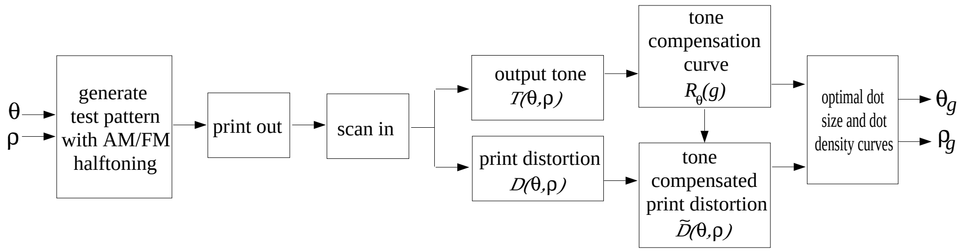


Figure 2: Diagram showing process used to design optimized LUT's for AM/FM halftoning algorithm. The process is based on direct measurement of printed halftone quality.

These LUT's must be selected to obtain the desired absorptance for each input gray level value $g(m,n)$. However, this leaves an additional degree of freedom which may be used to optimize a variety of printing attributes including print quality or print stability. Therefore, it is possible to achieve the best quality rendering of a desired absorptance by either varying the dot size or the dot density.

2.1 Parameter Design for AM/FM Halftoning Algorithm

In this section, we present a general procedure for designing dot size and dot density LUT's in order to maximize print quality while producing the desired tone response. Suppose we have selected a disperse dot halftoning algorithm to generate an FM pattern and a method to modulate the dot size. Then the disperse halftoning algorithm is first applied to generate a FM pattern with dot density ρ, and each FM dot is modulated with dot size θ.

There are two important quantities which must be measured in order to design AM/FM LUTs. The first quantity is output tone level, $T(\theta,\rho)$. Here we assume the $T(\theta,\rho)$ is linear in absorptance, so that $T(\theta,\rho)=1$ is a perfect black and $T(\theta,\rho)=0$ is white. Intermediate values of $T(\theta,\rho)$ vary linearly with luminance Y. The second quantity is print distortion, $D(\theta,\rho)$, which measures the visual difference between the rendered halftone pattern and an ideally rendered gray level. For now, we only assume that $D(\theta,\rho)$ is positive, and that it decreases with improving print quality. Our objective will be to minimize the print distortion $D(\theta,\rho)$ subject to a constraint that the tone $T(\theta,\rho)$ is correct.

Figure 2 shows the general approach used for measurement based design of the dot size and dot density LUT's. We first generate halftone patterns by printing gray patches with varying dot sizes, θ, and dot densities ρ. The print-out is then scanned in as a gray level image, and $T(\theta, \rho)$ is estimated by averaging the absorptance values of the pixels in each patch. We also compute $D(\theta, \rho)$ using a metric of print distortion that is appropriate for our problem.

The next step is to compute the tone compensation curve with respect to the desired tone curve of the printer. Let g be an integer value between 0 and 255 that specifies the nominal input gray level to the printer. The desired tone curve is denoted by $A(g)$ and is equal to the desired printed absorptance for input g. We assume that the desired tone curve is chosen so that $A(0) = A_W$ and $A(255) = A_B$, where A_B and A_W are the black and white points of the printer, respectively. For a given dot size θ and input gray level g, we may find the dot density ρ that achieves the desired absorptance by solving the equation

$$T(\theta, \rho) = A(g) \ . \tag{1}$$

Notice that for each fixed dot size θ, $T(\theta, \rho)$ is a monotone increasing function of the dot density ρ. Therefore, one can invert $T(\theta, \rho)$ with respect to ρ. We denote this inverse function by $T_\theta^{-1}(\cdot)$. We then compute the dot density curve which achieves the desired tone level as

$$\rho = R_\theta(g) \stackrel{\triangle}{=} T_\theta^{-1}(A(g)) \ . \tag{2}$$

We call this the tone compensation curve for dot size θ. For each input gray value g and dot size θ, $\rho = R_\theta(g)$ is the dot density required to produce the absorptance $A(g)$. In practice, some absorptance levels may not be achievable for certain dot size values θ, even when the dot density is chosen to be its maximum value, ρ_{max}. In this case $A(g) > T(\theta, \rho_{max})$, and we set $R_\theta(g) = \rho_{max}$. After determining $R_\theta(g)$, we may compute the print distortion of gray level g using dot size θ as

$$\tilde{D}(\theta, g) \stackrel{\triangle}{=} D(\theta, \rho)\Big|_{\rho = R_\theta(g)} = D(\theta, R_\theta(g)) \ . \tag{3}$$

We call this the tone compensated print distortion. For any gray level g with the property that $A(g) > T(\theta, \rho_{max})$, we set $\tilde{D}(\theta, g) = \infty$, so that print distortion is infinite.

The question remains of how to choose θ and ρ for each gray level g. Denote the dot size and the dot density selected to render gray level g as θ_g and ρ_g respectively. In order to optimize the choice of (θ_g, ρ_g), we define a cost function as an optimization objective. One choice is to use a minimum print distortion criterion, which minimizes $\tilde{D}(\theta, g)$ for each input gray level g. This optimization objective may be expressed as

$$\theta_g = \arg \min_{\theta \in [\theta_{min}, \theta_{max}]} \tilde{D}(\theta, g) \tag{4}$$

where the arg min operator returns the value of the argument that minimizes the cost function, and $[\theta_{min}, \theta_{max}]$ is the interval of possible values for θ. Equation (4) yields the value of θ_g that results in the best quality rendering for each individual gray level g. However, it may produce abrupt changes of dot size and dot density for adjacent input gray levels, which will cause visible contouring artifacts when rendering continuous tone images. To achieve smooth tone and texture transitions, we enforce smoothness constraints in the dot size and dot density curves by using regularization techniques. More specifically, we augment the cost function with quadratic penalty functions that favor solutions that are smooth. The dot size penalty function has the form

$$\frac{1}{2\sigma_1^2} \sum_{g=1}^{255} (\theta_{g-1} - \theta_g)^2 \ . \tag{5}$$

This function discourages large changes in dot size from one gray level to the next. The value σ_1^2 is used to control the amount of smoothing applied. When σ_1 is small, the penalty term is heavily weighted and the resulting θ_g will be very smooth. Similarly, the quadratic function applied to penalize the change of dot density from one gray level to the next is given by

$$\frac{1}{2\sigma_2^2} \sum_{g=1}^{255} (\rho_{g-1} - \rho_g)^2 \tag{6}$$

where σ_2 controls the regularization or smoothness of ρ_g. We would like to formulate our overall optimization problem in terms of $\{\theta_g\}_{g=0}^{255}$, so we can rewrite (6) using the relationship of (2) as

$$\frac{1}{2\sigma_2^2} \sum_{g=1}^{255} (R_{\theta_{g-1}}(g-1) - R_{\theta_g}(g))^2 \ . \tag{7}$$

Combining the expressions in (4), (5), and (7) results in the cost function

$$\mathcal{C}(\theta_0, \vec{\theta}, \theta_{255}) = \sum_{g=1}^{255} \left\{ \tilde{D}(\theta_g, g) + \frac{1}{2\sigma_1^2}(\theta_{g-1} - \theta_g)^2 + \frac{1}{2\sigma_2^2}(R_{\theta_{g-1}}(g-1) - R_{\theta_g}(g))^2 \right\} \tag{8}$$

where $\vec{\theta} \stackrel{\triangle}{=} [\theta_1, \cdots, \theta_{254}]$ is a vector which represents the contents of the dot size LUT. The optimal $\vec{\theta}$ is obtained by minimizing (8)

$$\vec{\theta}^* = \arg \min_{\vec{\theta} \in [\theta_{min}, \theta_{max}]^{254}} \mathcal{C}(\theta_0, \vec{\theta}, \theta_{255}) \tag{9}$$

where $[\theta_{min}, \theta_{max}]^{254}$ is the set of feasible values for the vector $\vec{\theta}$. Notice that in (9), the boundary points θ_0 and θ_{255} are fixed in the minimization. Generally, θ_0 is set to θ_{min} to insure high quality rendering in the highlight area, and θ_{255} is set to θ_{max} to achieve the full darkness in the shadow area. When $\sigma_1 = \sigma_2 = \infty$, the cost function degenerates to the minimum print distortion criterion in (4).

Minimization of (8) is, in general, a difficult problem because the function is not convex. In practice, we have found that this cost function has complex structure with many local minima, Therefore, we need a robust optimization method. In the following section, we develop a multiresolution iterative coordinate decent (MICD) optimization algorithm which we empirically show can consistently minimize the cost function (8) without becoming trapped in local minima.

2.2 Multiresolution Iterative Coordinate Decent Algorithm (MICD)

One commonly used optimization method is iterative coordinate decent (ICD). This method works by iteratively updating the coordinates to minimize a cost functional. For our problem, each element of the parameter vector $\vec{\theta}$ is sequentially updated to minimize the cost function of (8). The procedure is iterated until no further decrease in the cost function is achieved. A disadvantage of ICD is that it can become trapped in local minima, as will be demonstrated in the Section 4. Here we propose a more robust method for optimization, which we call multiresolution iterative coordinate decent (MICD).

The MICD algorithm works by optimizing the cost function after transformation of the parameter vector $\vec{\theta}$ by a series of coarse-to-fine transformations. These transformations range from a coarsest resolution of K to a finest resolution of 1. At each resolution k, ICD optimization is applied, and the result is used as an initial condition to the next finer resolution.

Consider the optimization problem for resolution k. Let $x^{(k)}$ be a vector of length $254 - k$ which is related to the vector $\vec{\theta}$ by the transformation

$$\vec{\theta} = \vec{\theta}_{int}^{(k)} + \mathbf{M}^{(k)} x^{(k)}$$

where $\vec{\theta}_{int}^{(k)}$ is the initial value of $\vec{\theta}$ at resolution k, and the elements of $\mathbf{M}^{(k)}$ are given by

$$\mathbf{M}_{ij}^{(k)} = \begin{cases} 1 & \text{if } j < i \leq j + k \\ 0 & \text{otherwise} \end{cases} .$$

Intuitively, the matrix $\mathbf{M}^{(k)}$ is chosen so that $x_j^{(k)}$ perturbs k components of $\vec{\theta}$ ranging from $\vec{\theta}_{j+1}$ to $\vec{\theta}_{j+k}$. The coarse resolution cost function is then given by

$$\mathcal{C}^{(k)}(\theta_0, x^{(k)}, \theta_{255}) = \mathcal{C}(\theta_0, \vec{\theta}_{int}^{(k)} + \mathbf{M}^{(k)} x^{(k)}, \theta_{255}) \ . \tag{10}$$

The MICD algorithm works by optimizing the cost function $\mathcal{C}^{(k)}(\theta_0, x^{(k)}, \theta_{255})$ as will be demonstrated in the Section 4.

The MICD algorithm works by optimizing the cost function $\mathcal{C}^{(k)}(\theta_0, x^{(k)}, \theta_{255})$ at resolution k using ICD optimization. The result is then used as an initial condition for ICD optimization at resolution $k - 1$. Formally, the MICD algorithm is as follows:

1. Set $\vec{\theta}_{int}^{(K)}$ to the minimum distortion solution of (4).
2. For $k = K$ to 1
   (a) Apply ICD optimization to the variable $x^{(k)}$
   $$x^{(k)} \leftarrow \text{ICD} \arg\min_{x^{(k)}} \mathcal{C}(\theta_0, \vec{\theta}_{int}^{(k)} + \mathbf{M}^{(k)} x^{(k)}, \theta_{255})$$
   (b) If $k \neq 1$,
   $$\vec{\theta}_{int}^{(k-1)} = \vec{\theta}_{int}^{(k)} + \mathbf{M}^{(k)} x^{(k)}$$
3. Set $\vec{\theta} \leftarrow \vec{\theta}_{int}^{(0)}$

Generally, we use $K = 10$ as our coarsest resolution in the optimization.

3 Detailed Implementation of AM/FM Halftoning

In this section, we describe a specific implementation of AM/FM halftoning that is suitable for the electro-photographic printers using pulse width modulation (PWM) technology [18]. PWM

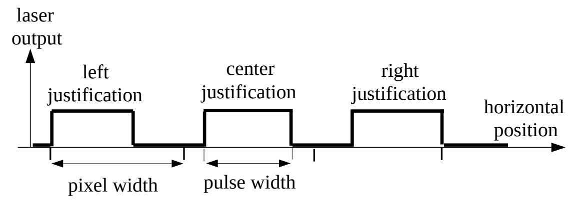


Figure 3: Illustration of how the laser output can be modulated to control dot size and placement.

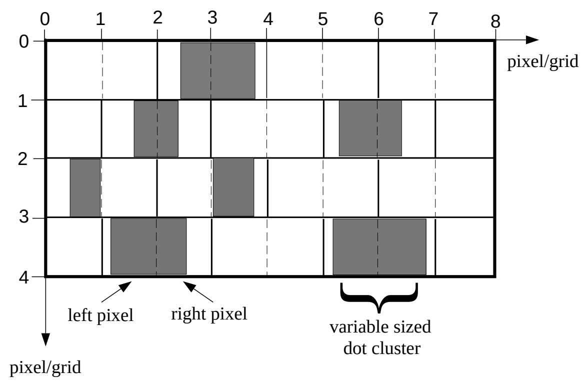


Figure 4: Illustration of how PWM technology is used to form clustered pixel pairs arranged on a diagonal grid.

technology allows the size of printed dots to be modulated and is therefore ideally suited to the AM/FM method. Section 3.1 describes how the AMFM algorithm can be implemented using PWM to modulate the size of dot clusters on a diagonal grid. Section 3.2 gives the details on the computation of tone and distortion measurements, and proposes a HVS-weighted mean square error as the print distortion metric. Finally, Section 3.3 introduces a dot size diffusion technique for use when the number of quantization levels of PWM is insufficient.

3.1 AM/FM Halfoning using Pulse Width Modulation

In a conventional EP printer, the laser is either "on" or "off" during the entire period it passes a pixel's location. This results in either a fully exposed dot, or no dot. In PWM technology, the dot can be modulated by changing both the time that the laser is on, and the registration of the exposure in the pixel grid. Figure 3 illustrates the typical laser output patterns generated by a PWM system. For each pixel, two attributes can be controlled, the pulse width and the pulse justification. A greater pulse width increases exposure and therefore creates a darker printed pixel. The pulse justification determines whether a pixel is printed in the left, center, or right positions

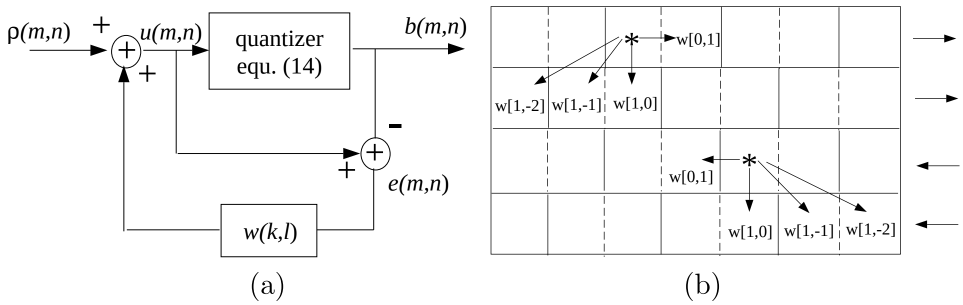


Figure 5: Modified error diffusion algorithm for the FM component of the AM/FM algorithm: a) Basic flow diagram, b) Positioning of weights and pixel processing order.

of the pixel grid.

In electro-photographic printing, small isolated dots tend to be unstable. their development can vary substantially with environmental conditions or small changes in roller speed or charge voltage. This type of instability can result in a variety of defects, including color/tone shifts, and banding artifacts. PWM technology can be used to form a more stable cluster of variable size by exposing pixel pairs as shown in Figure 4. For each pixel pair, the left pixel is right justified, and the right pixel is left justified. This results in pixels that are clustered together in pairs. The size of each pixel is also independently modulated using a pulse width value consisting of an integer between 0 and 63. Thus, the total cluster size ranges from a minimum dot size of $\theta = 0$ (total PWM value of 0) to a maximum dot size of $\theta = 2$ (total PWM value of 126).

Figure 4 shows how the 2-pixel clusters are arranged in a diagonal grid to fill all the locations on a rectangular lattice. Pairs of adjacent pixels are grouped together, but each row is offset relative to adjacent rows. Notice that the size of each pixel is modulated independently. This allows for more precise representation of edge detail. Experimentally, we have found that this diagonal packing of the pixel clusters yields effective dot placement with no overlap between adjacent clusters.

The FM modulation component determines which of the pixel pair clusters are enabled or “turned on”. For this purpose, we use a modification of conventional error diffusion which only allows dots to be fired on the diagonal grid associated with the cluster positions. The modified error diffusion algorithm is illustrated in Figure 5. Let the values (m, n) denote the row and column of a pixel respectively. Then

$$\theta(m,n) \triangleq \theta_{g(m,n)} \tag{11}$$

$$\rho(m,n) \triangleq \rho_{g(m,n)} \tag{12}$$

are the dot size and dot density at position (m,n) that results from application of the dot size and dot density look-up-tables. The modified error diffusion algorithm is then specified by the following three equations

$$u(m,n) = \rho(m,n) + \sum_{l>0} w(0,l)e(m,n-l) + \sum_{l} w(1,l)e(m-1,n-l) \tag{13}$$

$$b(m,n) = \begin{cases} 1 & \text{if } \{u(m,n) \geq t(m,n,\rho(m,n)\} \text{ and } \{(m+n)\text{mod}2 == 0\} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

$$e(m,n) = u(m,n) - b(m,n) \tag{15}$$

where $w(k,l)$ are error diffusion weights that normally sum to one. The major modification compared to conventional error diffusion is in the quantization step of (14) where the additional condition that $(m+n)\text{mod}2 == 0$ is added; and in general, the threshold value $t(m,n,\rho(m,n))$ will be dependent on the location and the input dot density value at current pixel. The positions such that $(m+n)\text{mod}2 == 0$ correspond to the left hand pixels in the pixel pair clusters of Figure 4. So this quantizer suppresses every other dot and only allows pixels on a diagonal grid to be turned on. Thus the maximum fraction of pixels that can be turned on is 50%. Consequently, the input dot density is limited to the range $[0, 0.5]$. The positioning of filter weights is also shown in Figure 5.

In our experiments, both the filter weights and threshold values of the modified error diffusion are dependent on input dot density $\rho(m,n)$. The weights and thresholds are optimized for each input gray level with the method described in [19]. Using the threshold modulation method developed in [19], the threshold function $t(m,n,\rho(m,n))$ is given by

$$t(m,n,\rho(m,n)) = t_0(\rho(m,n)) + t_1(\rho(m,n))p(m\text{mod}M, (n/2)\text{mod}M) \tag{16}$$

where $p(m,n)$ is a DBS midtone pattern with 45 degree rotation to match the diagonal grid, and $M = 64$ is the period of the DBS pattern. The rotated DBS pattern p(m,n;0.5) is shown in Figure 7 where $p(m,n) = 1$ is white, and $p(m,n) = 0$ is black. In (16), $t_0(\rho(m,n))$ controls the basic threshold value, while $t_1(\rho(m,n))$ controls the intensity of threshold modulation. The threshold modulation is used to break up regular halftone patterns near the area of dot densities of $1/8, 1/4, 3/8$. The error diffusion algorithm uses 2-row serpentine order with the values of the weights and thresholds illustrated in Figure 6. The halftone result for a serpentine ramp is shown in Figure 8. Overall, it achieves fairly uniform dot placement.

The values of the error diffusion output, $b(m,n)$, at the positions corresponding to $(m+n)\text{mod}2 == 0$ determine whether each pixel pair is enabled or disabled. More specifically, the output PWM code at position (m,n) is given by

$$PWM(m,n) = \lfloor \frac{63}{2}\theta(m,n) \rfloor * b(m, n - ((m+n)\text{mod}2)) \qquad (17)$$

where the maximum PWM code is assumed to be 63. Notice that the PWM code for each pixel is modulated independently through the choice of $\theta(m,n)$. This improves resolution by allowing fine edge details to be rendered more accurately.

3.2 Measurement of Tone Curve and Print Distortion.

Accurate estimation of the dot size and dot density curves depends on careful measurement of the tone curves, $T(\theta,\rho)$, and the print distortion, $D(\theta,\rho)$. Both curves are measured for N_1 discrete values of dot size denoted by $\{\theta^{(i)}\}_{i=0}^{N_1-1}$, and N_2 discrete values of dot density denoted by $\{\rho^{(i)}\}_{i=0}^{N_1-1}$.

Figure 9 shows a typical test pattern used to measure the tone and distortion curves for a single dot size $\theta^{(i)}$. The test pattern is produced by printing an AM/FM halftoned gray level test image using a fixed dot size of $\theta^{(i)}$, and a linearly varying dot density LUT, $\rho_g = g/254$. The test pattern is broken into 8 sub-patterns. Each sub-pattern consists of a 16×8 array of gray patches with dot densities ranging from the minimum of 0 to the maximum of 0.5. The randomization of the patches is critical to avoid the measurement of systematic tone variations across the printed page. We have also found that the use of sub-patterns improves accuracy by increasing the spatial localization of the measurements.

For each dot size $\theta^{(i)}$, the tone curve, $T(\theta^{(i)},\rho)$, and the print distortion curve, $D(\theta^{(i)},\rho)$, are computed by averaging measurements over M scanned test patches, each with dot density ρ. Let $s^{(j)}(m,n)$ be the $N \times N$ sampled version of the j^{th} scanned test patch where $0 \leq m < N$ and $0 \leq n < N$. Here we assume that $s^{(j)}(m,n)$ is measured in units of absorptance which are linear with reflected energy. Furthermore, let $S^{(j)}(k,l)$ be the DFT of $s^{(j)}(m,n)$ computed using a 2-D Hanning window to minimize boundary effects.

The initial value of the tone curve, denoted by $\tilde{T}(\theta,\rho)$, is then computed by averaging the absorptance values of the M test patches.

$$\tilde{T}(\theta^{(i)},\rho) = \frac{1}{M\,N^2} \sum_{j=0}^{M-1} \sum_{(m,n)} s^{(j)}(m,n) \tag{18}$$

The distortion metric we use is based on a linear shift-invariant low-pass filter proposed in [20] with the form of

$$H(u,v) = a \exp(-b\sqrt{u^2+v^2}) \tag{19}$$

where u and v are horizontal and vertical spatial frequency in units of cycles per degree, and a and b are constants that are dependent on average luminance of the image. Experimentally, b has been measured to be 5.169 *degree/cycle*; and for our purposes, we may assume that $a = 1$. The initial the print distortion is computed by

$$\tilde{D}(\theta,\rho) = \sqrt{\frac{1}{M} \sum_{j=0}^{M-1} \sum_{(k,l)\neq(0,0)} F(k,l)\,|S^{(j)}(k,l)|^2} \tag{20}$$

where $F(k,l)$ is the filter given by

$$F(k,l) = H\left(\frac{\pi d f_s}{180N}\left[\left(\left(k+\frac{N}{2}\right)\mathrm{mod}N\right)-\frac{N}{2}\right], \frac{\pi d f_s}{180N}\left[\left(\left(l+\frac{N}{2}\right)\mathrm{mod}N\right)-\frac{N}{2}\right]\right)\ , \tag{21}$$

d is the viewing distance in inches, f_s is the sampling frequency for $s(m,n)$ in samples per inch, and N is assumed even.

The initial tone and distortion curves are then smoothed in the ρ domain using a combination of median and linear filters. The initial tone curve is smoothed using two applications of a 5-point median filter followed by 10 applications of a linear filter with impulse response $[0.25, 0.5, 0.25]$. The initial print distortion curve is smoothed using three applications of a 5-point median filter followed by 12 applications of the same linear filter. After smoothing, the measured tone curve may not be completely monotonic. To insure monotonicity, we apply the following operation.

$$T(\theta^{(i)}, \rho) \leftarrow \max_{x \leq \rho} T(\theta^{(i)}, x) \ . \tag{22}$$

Since $T(\theta, \rho)$ and $D(\theta, \rho)$ are computed for discrete values of θ and ρ, the intermediate values of these functions must be smoothly interpolated to allow effective optimization of the cost function defined in (8). Generally, N_2 is large, so we use linear interpolation to compute intermediate values of ρ. Once this is done, the functions $R_\theta(g)$ and $\tilde{D}(\theta, g)$ may also be computed using (2) and (8). Then for each i, we then have the functions $R(\theta^{(i)}, g)$ and $\tilde{D}(\theta^{(i)}, g)$ for any g in the interval $[0, 255]$. We evaluate these functions for intermediate values of θ using cubic interpolation since this reduces interpolation error for a sparse sampling of dot size values.

3.3 Data Bandwidth Reduction with Dot Size Diffusion (DSD)

In some applications, the number of quantization levels used to specify the dot size may be limited. We will denote this set as $\Theta_p = \{\theta_{p_1}, \theta_{p_2}, \cdots, \theta_{p_L}\}$. This limitation may result from bandwidth constraints of the printing systems, or limited dynamic range of the PWM electronics. In either case, a small number of possible dot sizes for each dot cluster will result in quantization artifacts in the resulting AM/FM halftone. In order to eliminate these artifacts, we introduce a method of dithering the dot size using a second level of error diffusion.

Dot size diffusion diffuses dot size error across dot clusters formed by pixel pairs. The size of the $(i,j)^{th}$ dot cluster is given by

$$\theta_c(i,j) = \frac{1}{2} \left(\theta(i, 2j + i\text{mod}2) + \theta(i, 2j + i\text{mod}2 + 1) \right) * b(i, 2j + i\text{mod}2) \ . \tag{23}$$

However, due to the quantization of the PWM codes, only certain discrete values of $\theta_c(i,j)$ are possible. Denote this set of possible values as Ω. Then the dot size diffusion is specified by the following three equations illustrated in Figure 10.

$$u(i,j) \ = \ \theta_p(i,j) + \sum_{l>0} w(0,l) e_p(i, j-l) + \sum_{l} w(1,l) e_p(i-1, j-l) \tag{24}$$

$$\theta_q(i,j) = \begin{cases} \arg\min_{s\in\Omega}(|u(i,j)-s|) & \text{if } b(i,2j+i\text{mod}2) == 1 \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

$$e_\theta(i,j) = u(i,j) - \theta_q(i,j) \tag{26}$$

where $w(k,l)$ are the error diffusion weights. In our experimental results, we use the Floyd-Steinberg weights shown in Figure 11. The DSD method differs from conventional error diffusion primarily in quantization step of (25). In DSD, if a cluster is turned off, it is always quantized to 0. Otherwise, the multilevel quantizer selects the available cluster size that minimizes the dot size error between $u(i,j)$ and $\theta_q(i,j)$. After determining $\theta_q(i,j)$, we need to decide how to assign dot size values of the left pixel θ_l and that of the right pixel θ_r. If $\theta_q(i,j) = 0$, one simply sets both θ_l and θ_r to be zero. Otherwise, we first calculate the dot size ratio r_0 of the left and right pixels for the case of partial dotting

$$r_0 = \frac{\theta(i,2j+i\text{mod}2)}{\theta(i,2j+i\text{mod}2+1)} \ . \tag{27}$$

The values of θ_l and θ_r is then determined by

$$(\theta_l,\theta_r) = \arg \min_{\substack{(\theta_a,\theta_b),\\ \theta_a,\theta_b\in\Theta_p,\ \theta_a+\theta_b=\theta_q(i,j)}} \left|\frac{\theta_a}{\theta_b} - r_0\right| \ . \tag{28}$$

4 Experimental Results

All experiments were performed on an HP4000 printer that was modified to allow pulse width modulation as described in Section 3.1. The dot size was uniformly sampled from a minimum PWM value of $\theta = 28$ to a maximum PWM value of $\theta = 112$ using a sample interval of size 7. We choose the minimum value of $\theta = 28$ because it was the smallest dot size that could stably deposit toner when printed in isolation. The maximum value of $\theta = 112$ corresponds to two PWM codes of 56, and was choosen because it produced solid black fill regions with the same absorptance as the maximum PWM code of 63. The test pattern contained $0.25in \times 0.25in$ test patches and was scanned at 600dpi resolution using an HP Scanjet 6100c whose output was calibrated using a GreTag reflection spectrophotometer. The scans were then converted to a linear reflectance scale. The viewing distance for the human visual system model in all experiments was chosen to be $d = 6$ inches, and the target tone curve was chosen to be

$$A(g) = A_B + (A_W - A_B)\left(1 - \frac{g}{255}\right)^{\gamma}$$

where the parameter γ approximately corresponds the gamma-correction of the printers calibration. We found that the value $\gamma = 1.5$ worked well in all our experiments.

In our experiments, we tested two versions of the AM/FM algorithm. The standard implementation of AM/FM used 6 bit PWM codes to specify pulse widths. However as discussed above, the 6-bit PWM codes were restricted to the of range 0 to 56. This 6-bit version of AM/FM allows almost continuous variation of the pixel width. A second version of the AM/FM algorithm was tested that used only 2-bits per pixel to specify 4 possible PWM code values corresponding to 0, 28, 42, and 56. In this case, the available dot sizes for each pixel pair in units of pulse width are given by $\Omega = \{0, 28, 42, 56, 70, 84, 98, 112\}$. The 2-bit AM/FM then used DSD to eliminate contouring artifacts due to the quantization. Both the 6-bit and 2-bit versions of AM/FM used the same pixel justification scheme as shown in Fig. 4.

4.1 AM/FM LUT Design Results

We first present results for AM/FM LUT design using AM/FM halftoning with 6-bit PWM codes and no DSD. Figure 13 illustrates the initial and smoothed versions of the tone curve and print distortion curve for a dot size of $\theta = 1.0$. Notice that they track each other accurately, but the smoothed tone curve is guaranteed monotonic.

Figure 14 shows the tone and print distortion curves for each dot size. In general, the larger dot size achieves greater absorptance output at a fixed dot density. Notice that, for small dot sizes, it is impossible to achieve a full back ($A = A_B$). As dot size becomes larger, the dot overlap becomes severe, which results in the saturation of the output absorptance at high dot densities.

The tone compensation curves $R_{\theta^{(i)}}(g)$ are plotted in Figure 15. As expected, greater dot density is needed for a smaller dot size to achieve a given absorptance. Figure 16 shows the tone compensated print distortion, $\tilde{D}_{\theta^{(i)}}(g)$. Importantly, it is clear that at each gray level, g, the minimum distortion is achieved for varying values of dot size $\theta^{(i)}$.

Figure 17 shows the dot size and dot density curves that result from different choices of smoothing parameters and optimization algorithms. A smaller value of σ_1 or σ_2 constrains the dot size or dot density curve to be smoother as indicated in (8). However, as with any regularized optimization problem, there is a tradeoff between smoothness of the dot size/density curves and maximization of halftone quality at individual gray levels. For this experiment, we found that $\sigma_1 = 0.05$ and $\sigma_2 = 0.1$ yields the best overall subjective quality.

Notice that the smaller dot size is selected in the highlight area. Intuitively, this is because the small dots are less visible in highlights. In the midtone, the dot size stays near a value of 0.70. In our experiments, we observed that dot size 0.70 achieved the highest overall quality halftone texture over a range of dot sizes for this particular printer. In the dark regions, Larger dot size is preferable because this creates large "holes" which are more stable for printing.

For the case $\sigma_1 = 0.05$ and $\sigma_2 = 0.1$, we also compared the results of MICD and ICD optimization. Both the optimal dot size and dot density curves of MICD are smoother compared with those of ICD. The smooth dot size and dot density transition is crucial to avoiding abrupt halftone texture change for adjacent gray levels, which would otherwise be very visible. Figure 18 shows the comparison of cost function value change of MICD and ICD optimization. The converged cost function value is 86.45 for MICD optimization versus 86.74 for ICD optimization. Thus, ICD optimization is trapped in a local minimum. Generally, we have observed that the MICD optimization results in consistently lower values of the cost function.

Figure 19 shows the tone compensated print distortion curves for AM/FM halftoning with 2-bit PWM codes and DSD. The results of MICD optimization using $\sigma_1 = 0.05$ and $\sigma_2 = 0.1$ are given in Figure 20. As we see, in the highlight area, the optimal dot size curve stays with dot size $\theta = 0.44$. Thus, AM/FM halftoning uses purely FM modulation. $\theta = 0.44$ corresponds to pulse width 28, which is an available pulse width. Since the DSD uses the combination of the adjacent available dot sizes to render an intermediate dot size, it makes the halftone texture of the intermediate dot size noisier. Therefore, permissible dot sizes are preferred. For the midtone area, the optimal dot size stays around $\theta = 0.67$, which corresponds to available pulse width 42. For the shadow region, the dot size goes up quickly to achieve the desired absorptance and better print stability.

4.2 Print Samples

In this section, we compare the print-outs produced using AM/FM halftoning, Floyd-Steinberg error diffusion, and a clustered dot screening algorithm called PhotoTone [21]. PhotoTone is designed to use the printer's PWM capability and has a screen frequency of 141 lines per inch. Floyd-Steinberg error diffusion was performed at 600dpi and used PWM codes of 0 and 56 as "off" and "on". The PWM code of 56 was used because it allows Floyd-Steinberg error diffusion to produce solid black regions with the same absorptance as those produced with the AM/FM algorithms. The results of Floyd-Steinberg error diffusion and PhotoTone screening were both tone compensated to match the desired target tone scale.

All figures were obtained by scanning portions of actual 600dpi print-outs. Therefore the figures show the quality and defects that were generated in the final prints. The scan resolution was 1200dpi and all the scans are displayed at approximately 300dpi (i.e. $4\times$ magnification).

First, we compare several gray level patches from a ramp. Figure 21 shows the results. In the highlight region, AM/FM halftoning achieves superior print quality because isolated dots are much less visible. Floyd-Steinberg error diffusion produces large dots in the highlight regions because dot size can not be modulated as a function of gray level. The PhotoTone result also has visible dots in the highlight region, but these dots are less visible than those in Floyd-Steinberg error diffusion. This is because the PhotoTone algorithm uses the PWM capability to reduce dot size, but it is still limited to a fixed grid of dot positions, so it can not achieve the same high density of small dots that that the AM/FM method achieves. Floyd-Steinberg error diffusion also has wormy halftone patterns. In the midtone regions, the halftone texture transition for AM/FM halftoning is smooth, and dots are more uniformly distributed. PhotoTone screening achieves the smoothest texture transition in 600dpi print-outs due to the regular dot placement of cluster dot screening. The midtone region of Floyd-Steinberg error diffusion produces the smoothest textures of the four algorithms. The Floyd-Steinberg algorithm has an advantage in this respect because it uses the full 600dpi grid, rather than being restricted to the diagonal pixel-pairs of the AM/FM algorithm.

However, the regular directional halftone textures and fine dot structure of the Floyd-Steinberg error diffusion make it much more prone to texture discontinuities and printing instability than either AM/FM or clustered dot screening. In the shadow regions, AM/FM halftoning creates desirable "hole" structures which increase the printing stability. Floyd-Steinberg error diffusion produces relatively little halftone graininess in the dark area. But it also tends to show worse banding artifacts in practice due to reduced stability. PhotoTone algorithm also creates a regular "hole" structure.

Overall, the PhotoTone screening is the most resistant to banding artifacts. In highlight and midtone regions, the banding artifacts of AM/FM halftoning and Floyd-Steinberg error diffusion are comparable; however, in the shadow regions, Floyd-Steinberg error diffusion shows worse banding artifact.

Figure 22 through 25 compare the four halftoning algorithms using a synthesized test image consisting of 120 and 160 line per inch scan bars, scanned text, and a continuous tone image. The line scan bar are particularly important for evaluating Moiré resistance.

For the scanned bars, both AM/FM halftoning and Floyd-Steinberg error diffusion show high spatial resolution and are free of Moiré artifacts. Floyd-Steinberg achieves slightly better spatial resolution probably due both to the inherent sharpening of conventional error diffusion [22] [1] and the pixel grouping of AM/FM halftoning. However, PhotoTone algorithm has much lower spatial resolution and serious Moiré artifacts due to the interactions between its own screen frequency and the periodic signals in the test image.

On the scanned text, the AM/FM algorithm produces a clean result, whereas the Floyd-Steinberg error diffusion looks muddy in the background. The AM/FM result with 2-bit PWM codes and DSD has softer text edges, and PhotoTone algorithm has the softest text edge rendering.

For the scanned continuous tone image, the major differences between AM/FM halftoning and Floyd-Steinberg error diffusion are in the highlight areas. For example, AM/FM renders the clouds much more smoothly than Floyd-Steinberg error diffusion. In the midtone regions, Floyd-Steinberg

[1] For this implementation of AM/FM halftoning, the error diffusion in the FM part was trained to remove the sharpening effect.

error diffusion is somewhat less grainy due to the smaller dot size, but this reduced grainyness causes greater printing instability. For example, Floyd-Steinberg error diffusion has more noticable horizontal streaks through the sky region and women's legs. These streaks are caused by banding artifacts and are enhanced by printer instability. In the dark areas such as the mountain region above the water, the AM/FM algorithm produces more distinct holes which can increase grainyness. However, these holes are less noticable in the original printout due to the 4× magnification of our figures.

5 Conclusions

In this paper, we present both a general theory and specific implementation for AM/FM halftoning. AM/FM halftoning uses a combination of dot size and dot density modulation to produce the best possible print quality at each gray level. Our approach to design the required dot size and dot density curves is based on regularized optimization of print quality as measured from actual printed and scanned halftones. We propose a multi-resolution iterative coordinate decent optimization algorithm for robustly optimizing the resulting non-convex functional, and show that it performs better than a fixed scale method. Finally, we introduce a dot size diffusion method for use when the available dot sizes must be quantized.

The results show that the AM/FM halftoning algorithm produces high quality halftone images on electro-photographic printers with pulse width modulation (PWM) capability. Generally, AM/FM halftoning produces more stable and higher quality halftones than conventional error diffusion, while eliminating the Moiré artifacts typical of clustered dot screens. By using small dot sizes in highlights, AM/FM halftoning reduces or eliminates visible isolated dots; and by using larger dots in dark regions, it increases the stability of printed output. Moreover, the combination of detail rendition, stability, and resistance to Moiré make it particularly suitable for scan-to-print applications of electro-photographic printing.

Acknowledgements

This research is funded by Hewlett-Packard Inc. We would like to thank Professor Jan P. Allebach, Peter Bauer, George Kerby and Qian Lin for their valuable inputs during the discussions. We also would like to thank Tichiun Chang for designing the weights and thresholds of the modified error diffusion used in this work.

References

[1] J. C. Stoffel and J. F. Moreland, "A survey of electronic techniques for pictorial reproduction," *IEEE Trans. on Communications*, vol. 29, pp. 1898–1925, 1981.

[2] J. P. Allebach and b. Liu, "Analysis of halftone dot profile and aliasing in the discrete binary representation of images," *J. Optical Society. America*, vol. 67, no. 9, pp. 1147–1154, September 1977.

[3] B. E. Bayer, "An optimum method for two-level rendition of continuous-tone pictures," *IEEE International Conference on Communications*, vol. 1, June 11-13 1973, pp. 11–15.

[4] J. Sullivan, L. Ray, and R. L. Miller, "Design of minimum visual modulation halftone patterns," *IEEE Trans. on Systems Man and Cybernetics*, vol. 21, no. 1, pp. 33–38, January 1991.

[5] K. E. Spaulding, R. L. Miller, and J. Schidkraut, "Methods for generating blue-noise dither matrices for digital halftoning," *Journal of Electronic Imaging*, vol. 6, no. 2, pp. 208–230, April 1997.

[6] R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial greyscale," *Journal of the Society for Information Display*, vol. 17, no. 2, pp. 75–77, 1976.

[7] R. A. Ulichney, "Dithering with blue noise," *Proc. of the IEEE*, vol. 76, pp. 56–79, January 1988.

[8] B. Kolpatzik and C. A. Bouman, "Optimized error diffusion for image display," *Journal of Electronic Imaging*, vol. 1, no. 3, pp. 277–292, July 1992.

[9] M. Analoui and J. P. Allebach, "Model-based halftoning using direct binary search," *Proc. of SPIE/IS&T Conf. on Electronic Imaging Sci. and Tech.*, vol. 1666, Feb 1992, San Jose, CA, pp. 96–108.

[10] T. N. Pappas and D. L. Neuhoff, "Least-squares model-based halftoning," *Proc. of SPIE/IS&T Conf. on Electronic Imaging Sci. and Tech.*, vol. 1666, February 1992, San Jose, CA, pp. 165–176.

[11] J. B. Mulligan and J. A. J. Ahumada, "Principled halftoning based on models of human vision," *Proc. of SPIE/IS&T Conf. on Electronic Imaging Sci. and Tech.*, vol. 1666, February 1992, San Jose, CA, pp. 109–121.

[12] R. Levien, "Output dependent feedback in error diffusion halftoning," *IS&T's Eighth Int. Congr. on Adv. in Non-Impact Printing Tech.*, October 1992, pp. 280–282.

[13] D. Lau, G. R. Arce, and N. Gallagher, "Green-noise digital halftoning," *Proc. of the IEEE*, vol. 86, no. 12, pp. 2424–2444, December 1998.

[14] D. Lau, G. R. Arce, and N. Gallagher, "Digital halftoning via green-noise masks," *J. Optical Society America A*, vol. 16, no. 7, pp. 1575–1586, July 1999.

[15] D. Lau, G. R. Arce, and N. Gallagher, "Digital color halftone with generalized error diffusion and multichannel green-noise masks," *IEEE Trans. on Image Processing*, vol. 9, no. 5, pp. 923–935, December 1998.

[16] N. Damera-Venkata and B. L. Evans, "Adaptive threshold modulation for error diffusion halftoning," *IEEE Trans. on Image Processing*, vol. 10, no. 1, pp. 104–116, 2001.

[17] G. Overall and P. Wright, "Ibm laser printer 4029 series print quality enhancements," *Personal Systems*, pp. 7–12, 1992.

[18] C. Tung, "Resolution enhancement technoogy in hewlett-packard laser jet printers," *Proc. SPIE Color Hard Copy and Graphic Arts II*, 1993, pp. 440–448.

[19] P. Li and J. P. Allebach, "Tone dependent error diffusion," *IEEE Trans. on Image Processing*, submitted, 2001.

[20] R. Nasanen, "Visibility of halftone dot textures," *IEEE Trans. on Systems Man and Cybernetics*, vol. 14, no. 6, pp. 920–924, November/December 1984.

[21] A. Frazier, "System and method for enhancing graphic features produced by marking engines." US Patent 5515480, May 1996.

[22] R. Eschbach and K. T. Knox, "Error diffusion algorithm with edge enhancement," *J. Optical Society America A*, vol. 8, no. 12, pp. 1844–1850, July 1991.

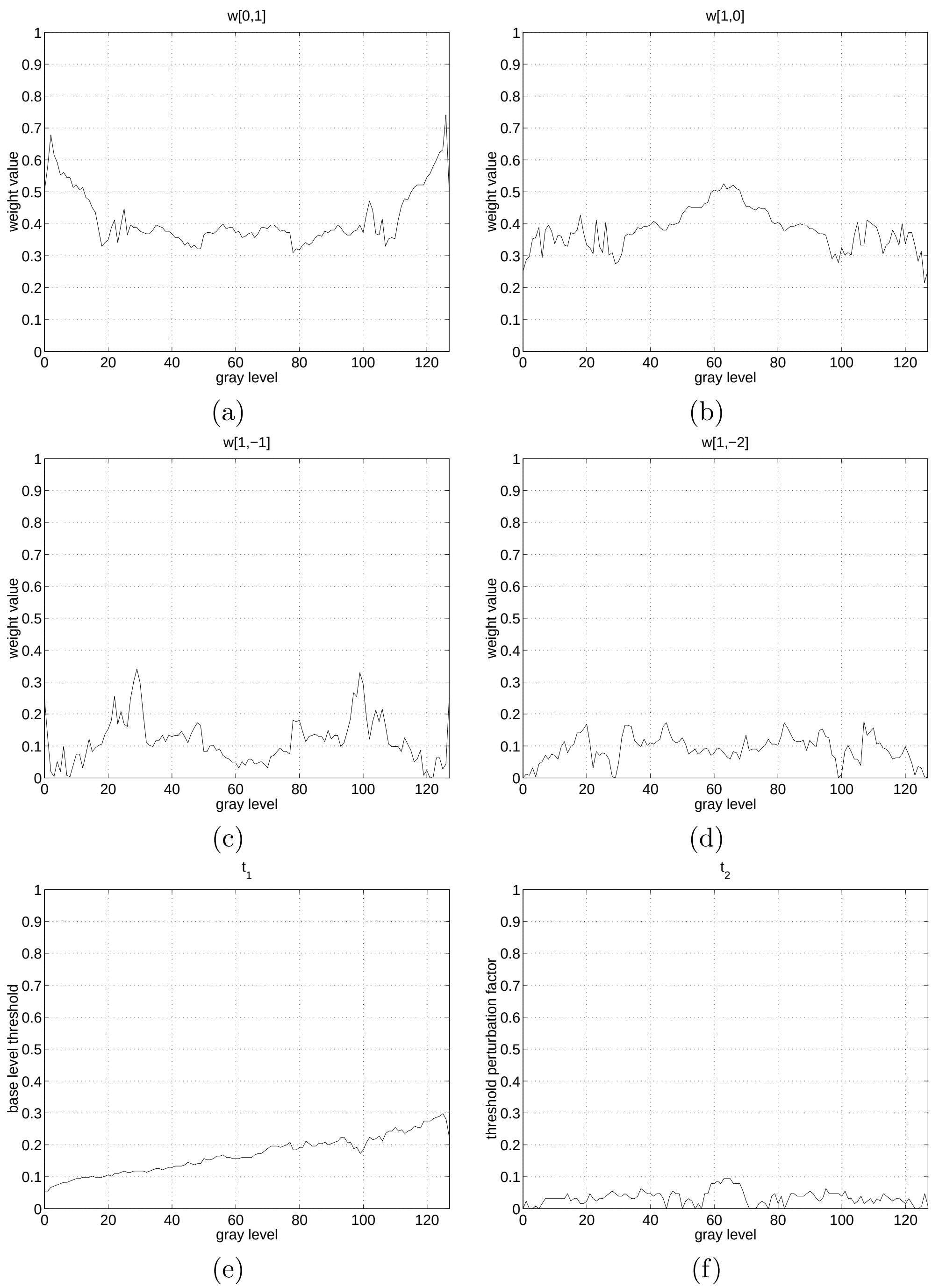


Figure 6: Error filter weights and threshold values of tone-dependent modified error diffusion.

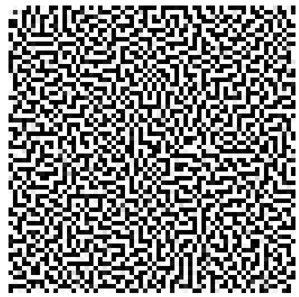


Figure 7: DBS midtone pattern used in threshold modulation.

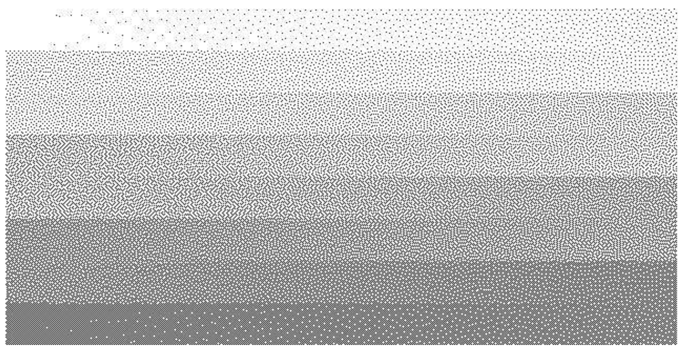


Figure 8: Serpentine ramp of the modified ED.

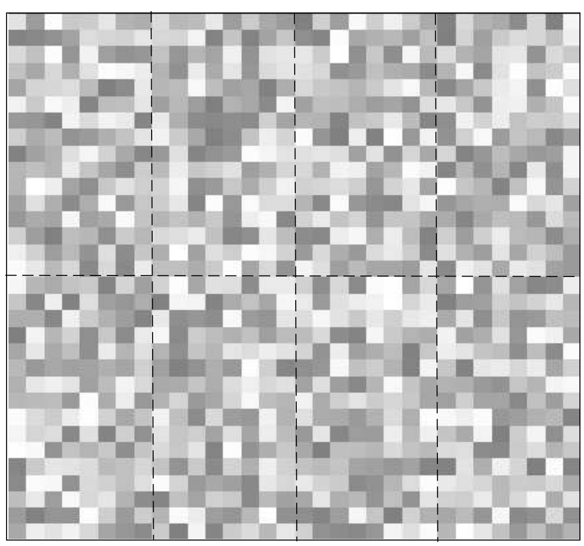


Figure 9: Test pattern used to measure $T(\theta, \rho)$ and $D(\theta, \rho)$. Notice that gray patches are distributed randomly to reduce the effects of printer variation.

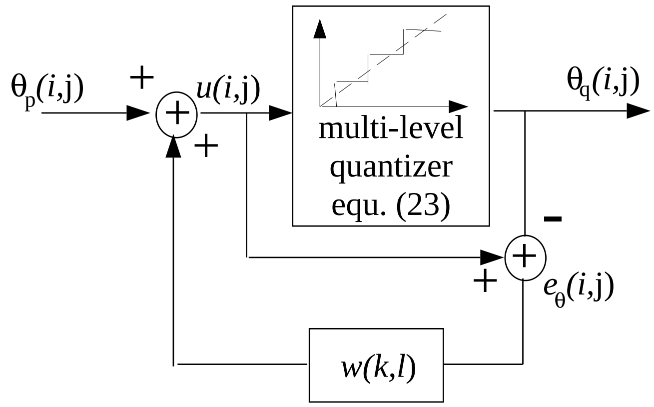


Figure 10: Flow diagram for dot size diffusion (DSD) process.

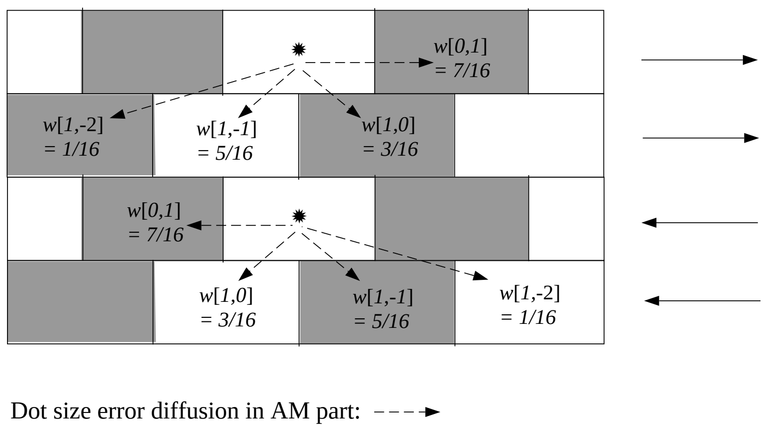


Figure 11: Diagram illustrating diffusion positions and weights for dot size diffusion (DSD) process.

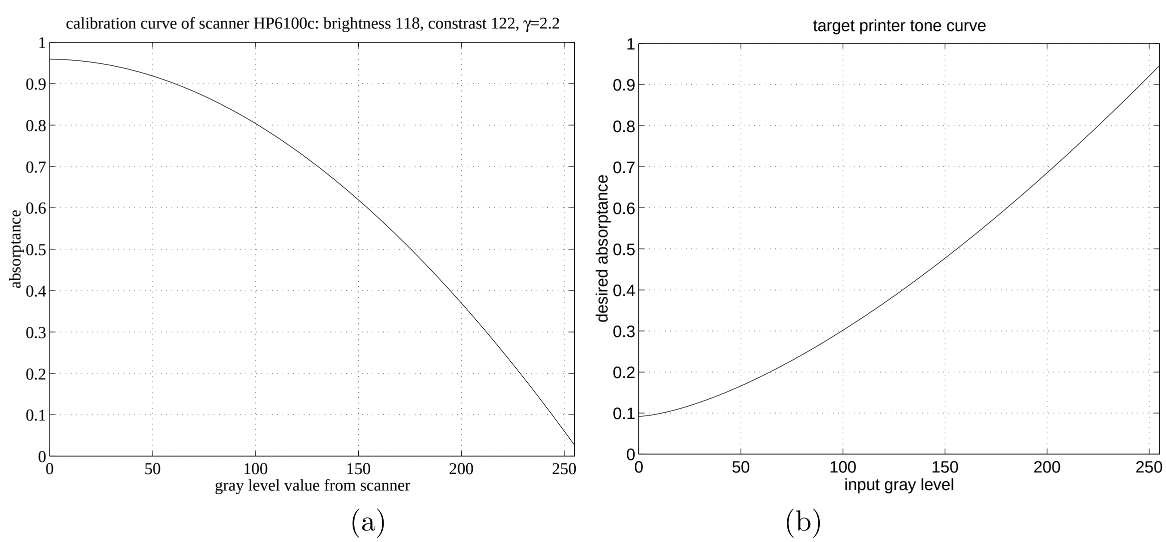
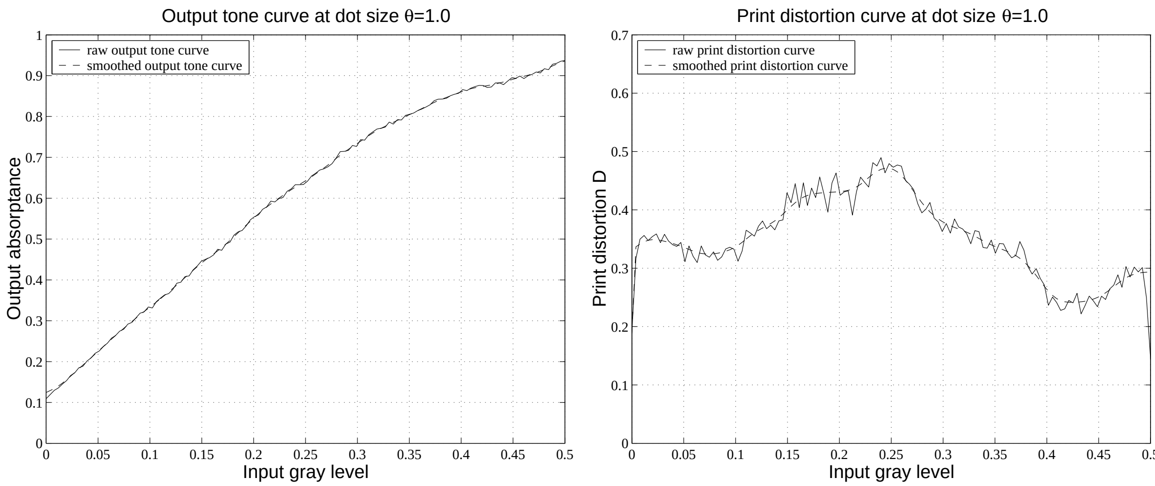
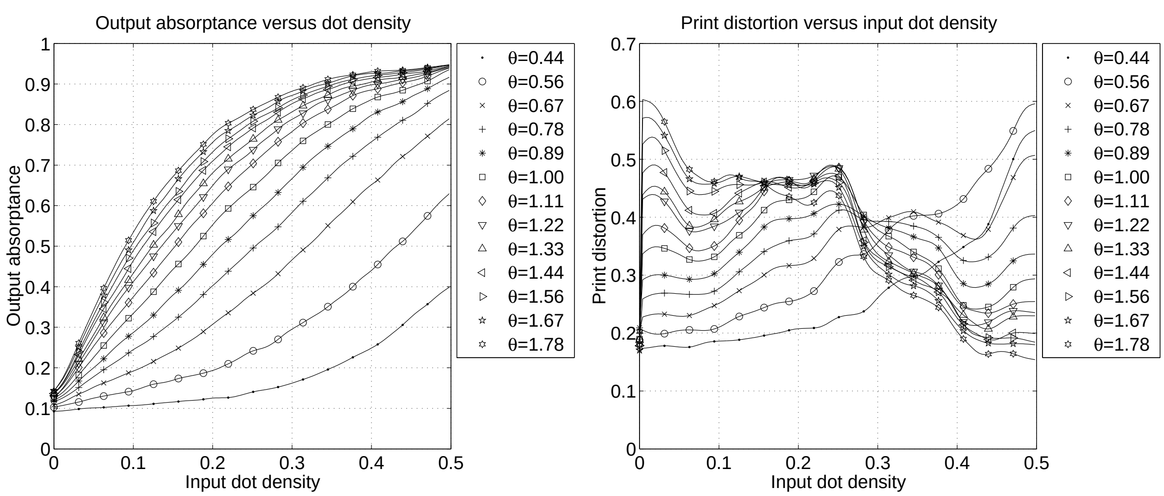


Figure 12: a) Calibration curve of HP6100c scanner with brightness 118, contrast 122, gamma 2.2. b) Target printer tone curve used in the experiments.



(a) (b)

Figure 13: Raw and smoothed curves measured for 6-bit AM/FM halftoning with $\theta = 1.0$. a) Raw and smoothed tone curves. b) Raw and smoothed print distortion curves.



(a) (b)

Figure 14: Smoothed tone and print quality curves for 6-bit AM/FM halftoning using different dot sizes.

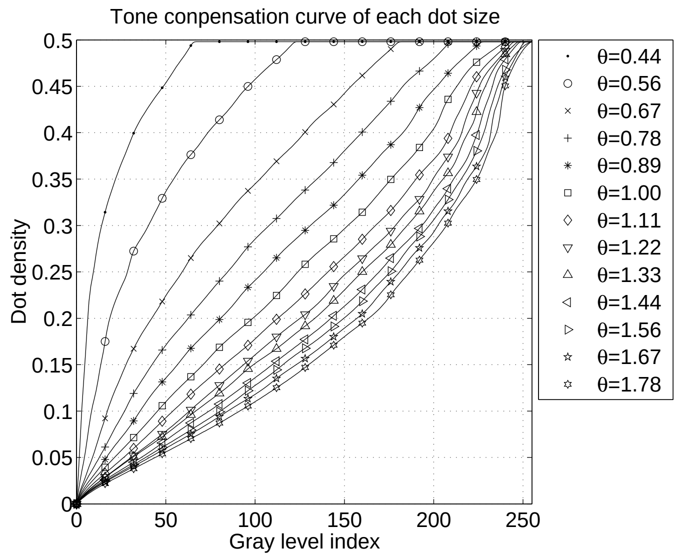


Figure 15: Tone compensation curves required for each dot size, $R_{\theta^{(i)}}(g)$. Each curve is designed to achieve the desired target tone curve for the specified dot size using the AM/FM halftoning with 6-bit PWM codes.

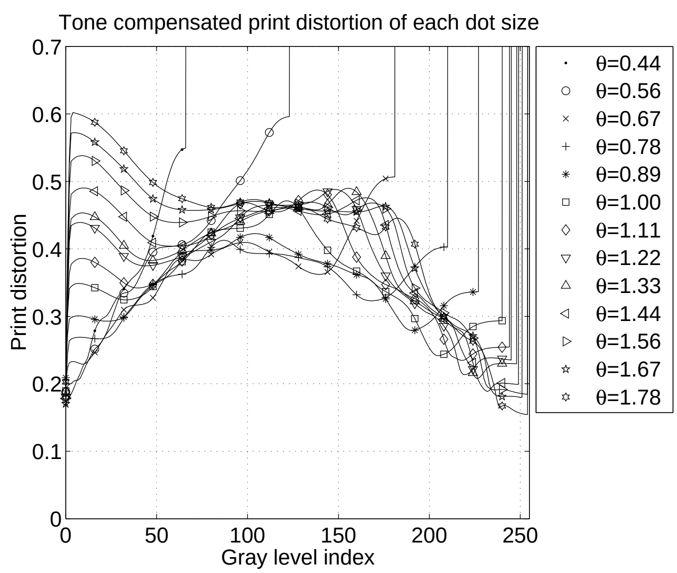
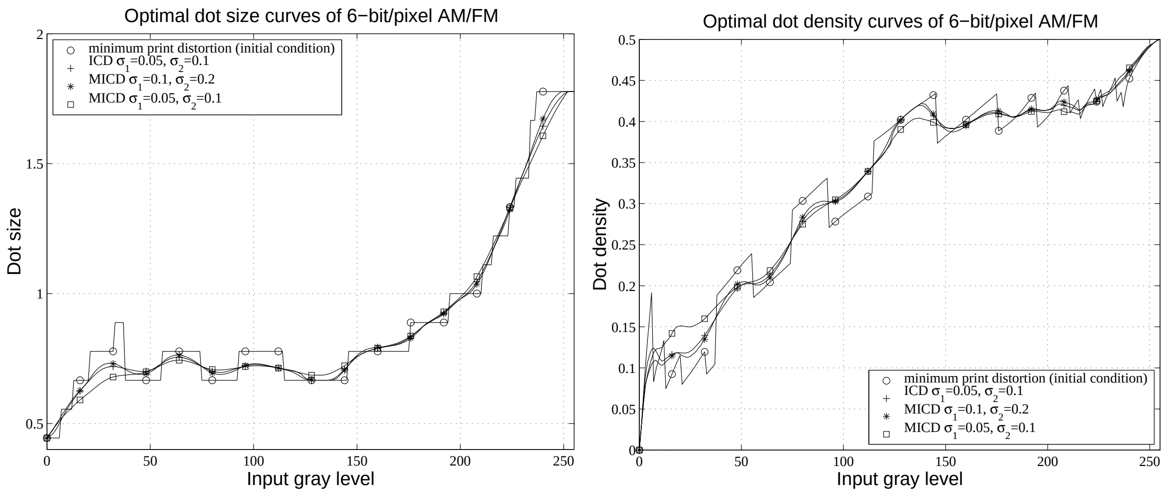


Figure 16: Print distortion as a function of the desired gray scale output value using the AM/FM halftoning with 6-bit PWM codes. Notice that at each gray level different dot sizes result in substantially different quality.



(a) (b)

Figure 17: Plots of a) optimal dot size curves and b) optimal dot density curves for AM/FM halftoning with 6-bit PWM codes. Each plot represents a different choice of smoothing parameters or different optimization algorithm.

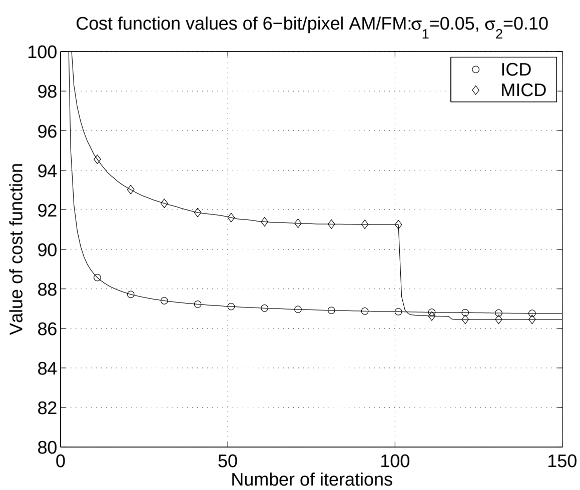


Figure 18: Plot of cost as a function of iteration number using both the MICD and ICD optimization algorithms for AM/FM halftoning with 6-bit PWM codes. Notice that the MICD algorithm achieves a slightly lower final value of the cost.

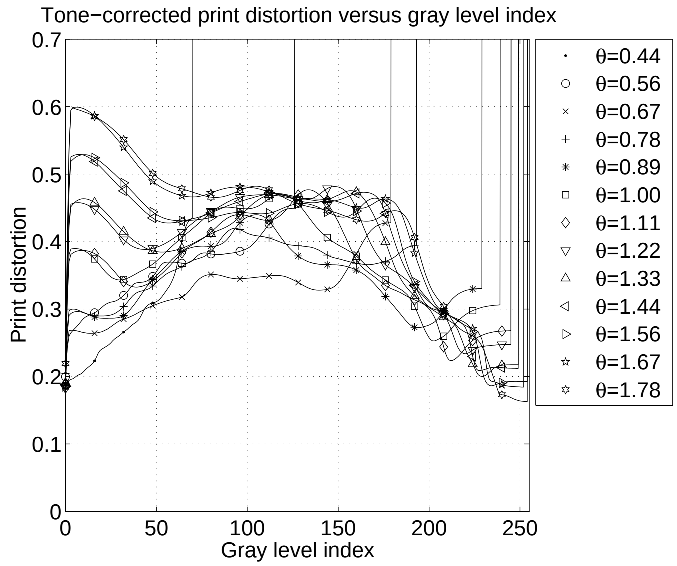


Figure 19: Print distortion as a function for AM/FM halftoning with 2-bit PWM codes and DSD with DSD. Notice that at each gray level different dot sizes result in substantially different quality.

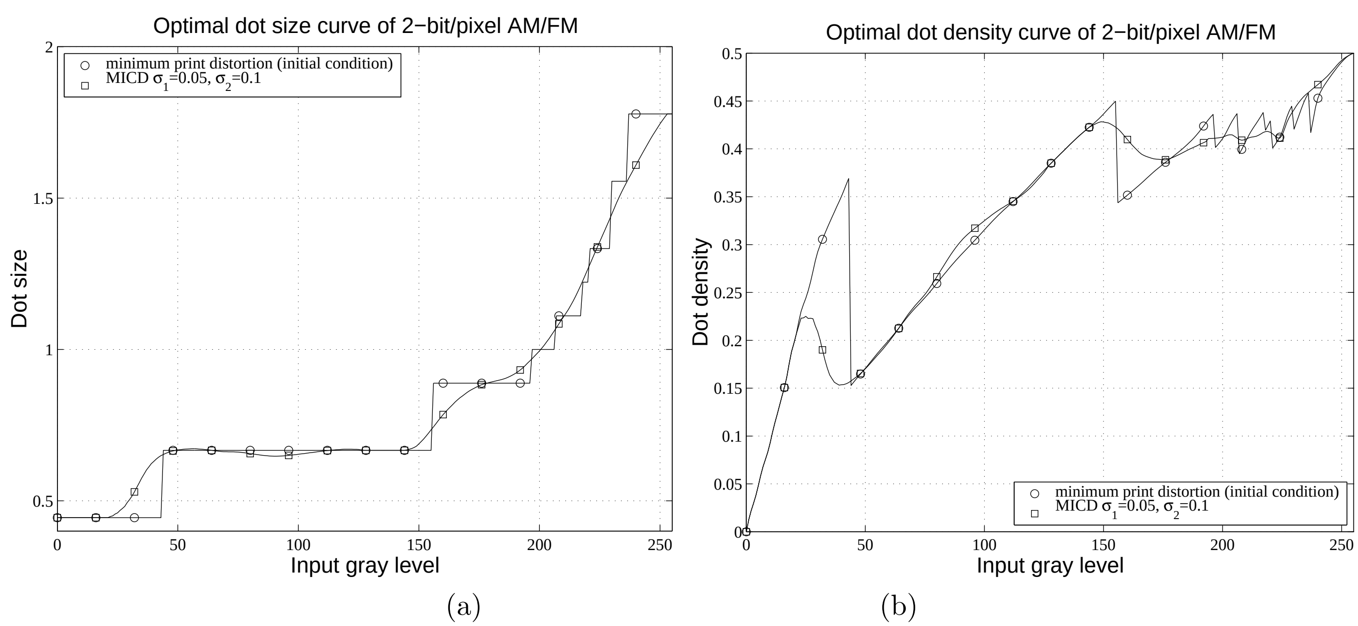
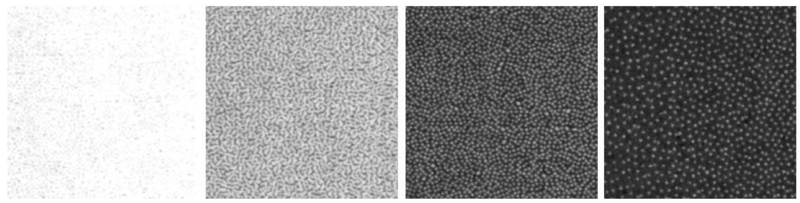
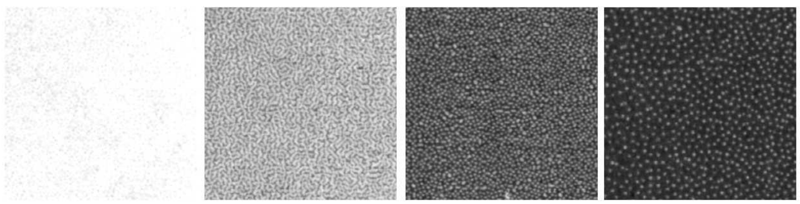


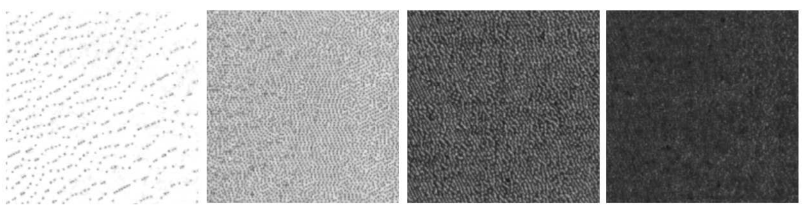
Figure 20: Plots of a) optimal dot size curves and b) optimal dot density curves for AM/FM halftoning with 2-bit PWM codes and DSD.



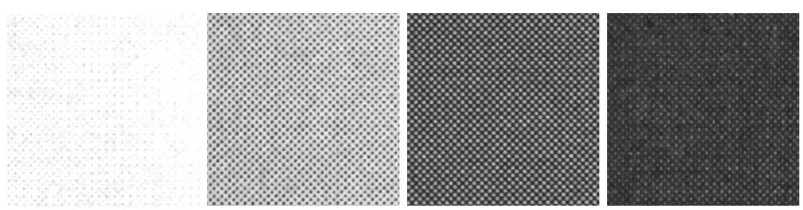
(a)



(b)



(c)



(d)

Figure 21: Scanned patches from the print-outs of a ramp using a) AM/FM halftoning with 6-bit PWM codes; b) AM/FM halftoning with 2-bit PWM codes and DSD; c) Floyd-Steinberg error diffusion; d) PhotoTone cluster dot screening.

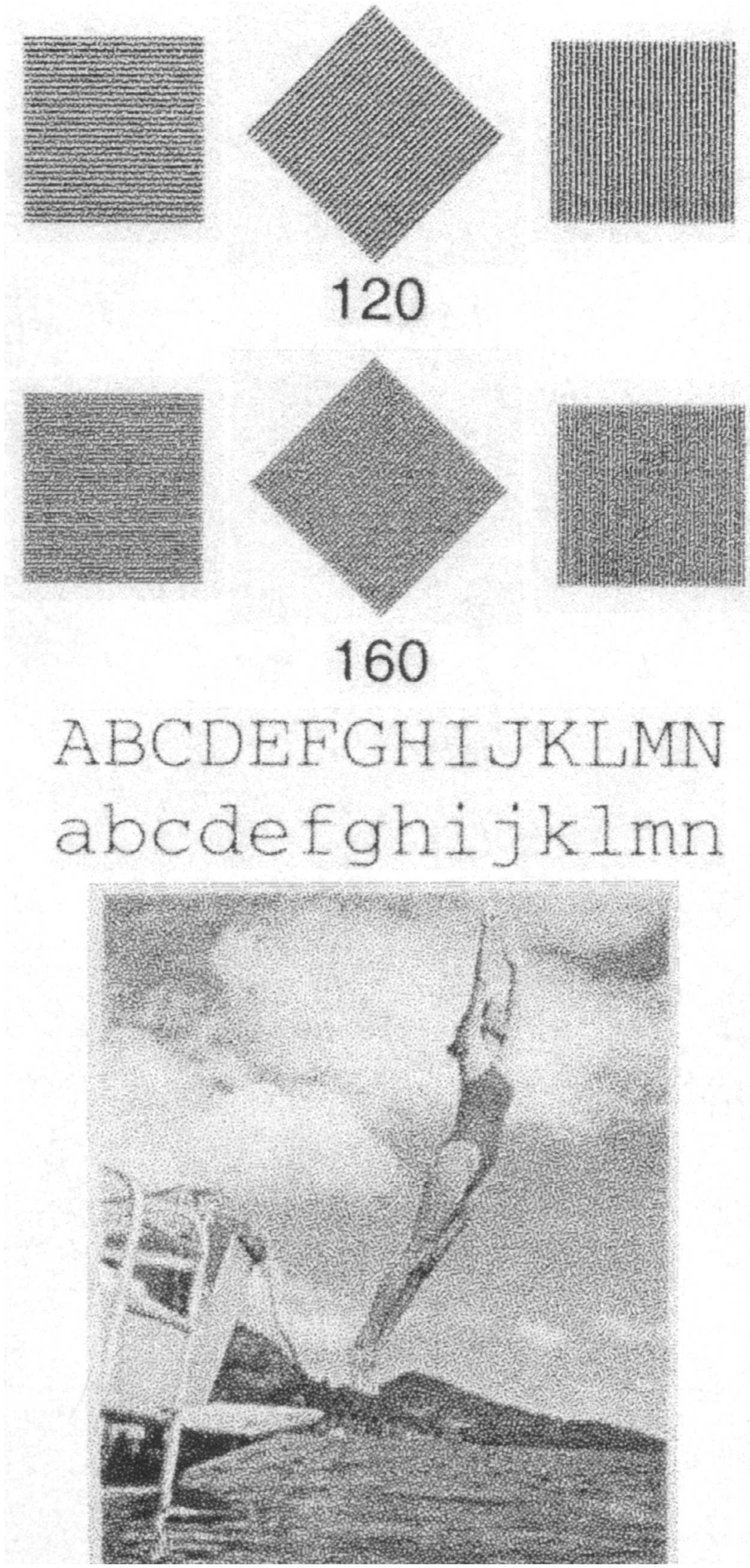


Figure 22: Rendering of a scanned image using AM/FM halftoning with 6-bit PWM codes.

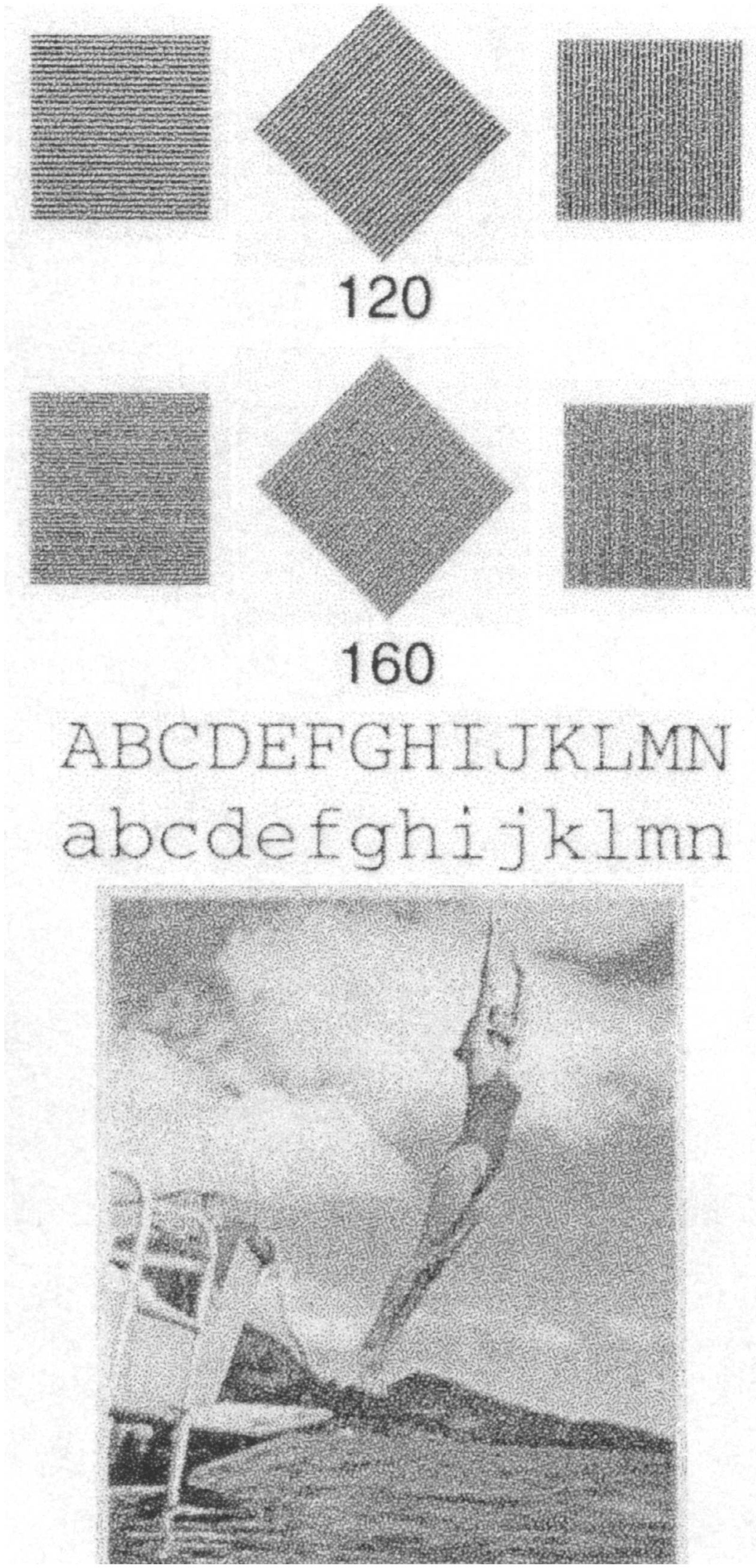


Figure 23: Rendering of a scanned image using AM/FM halftoning with 2-bit PWM codes and DSD.

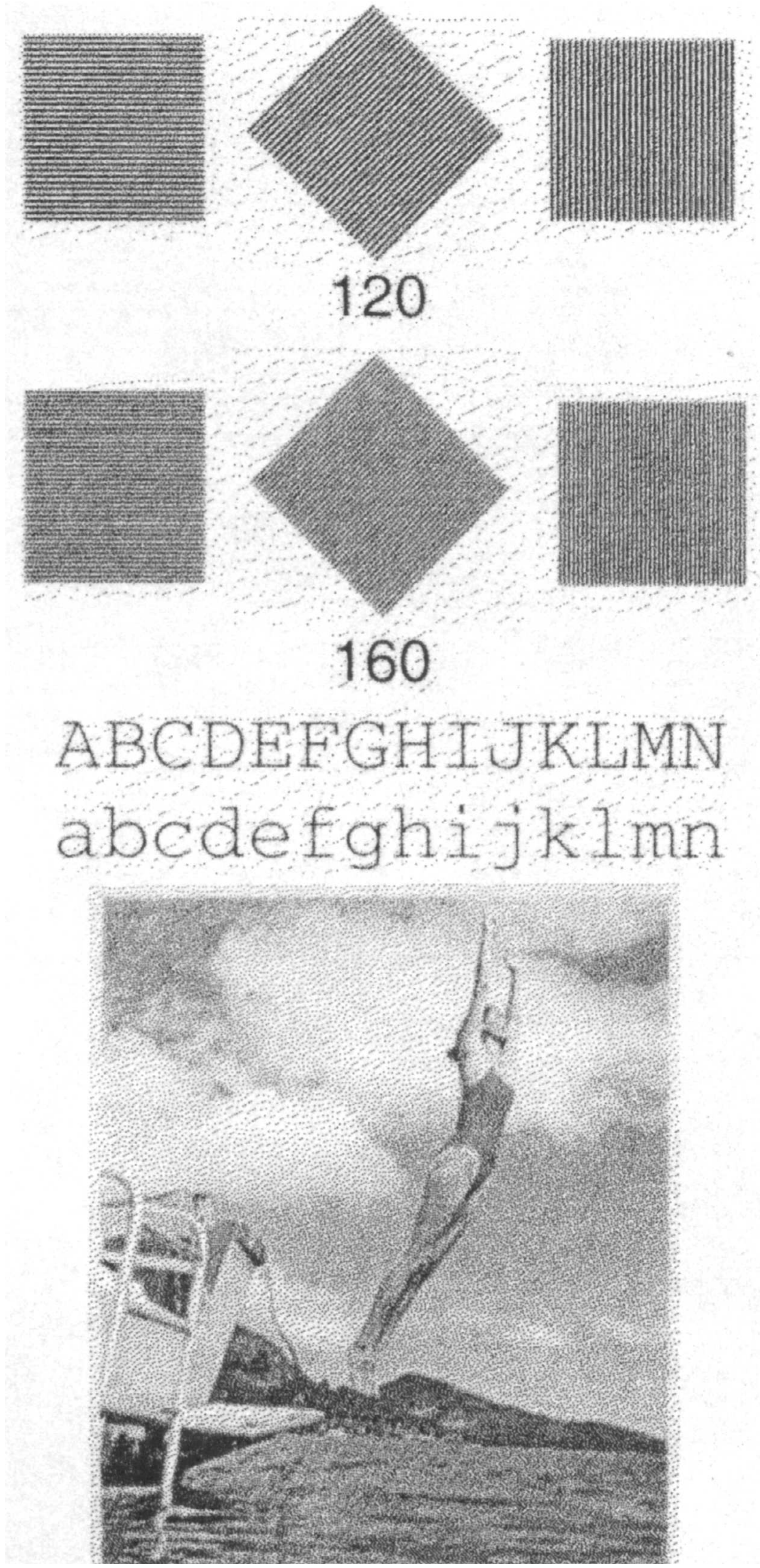


Figure 24: Rendering of a scanned image using Floyd-Steinberg error diffusion.

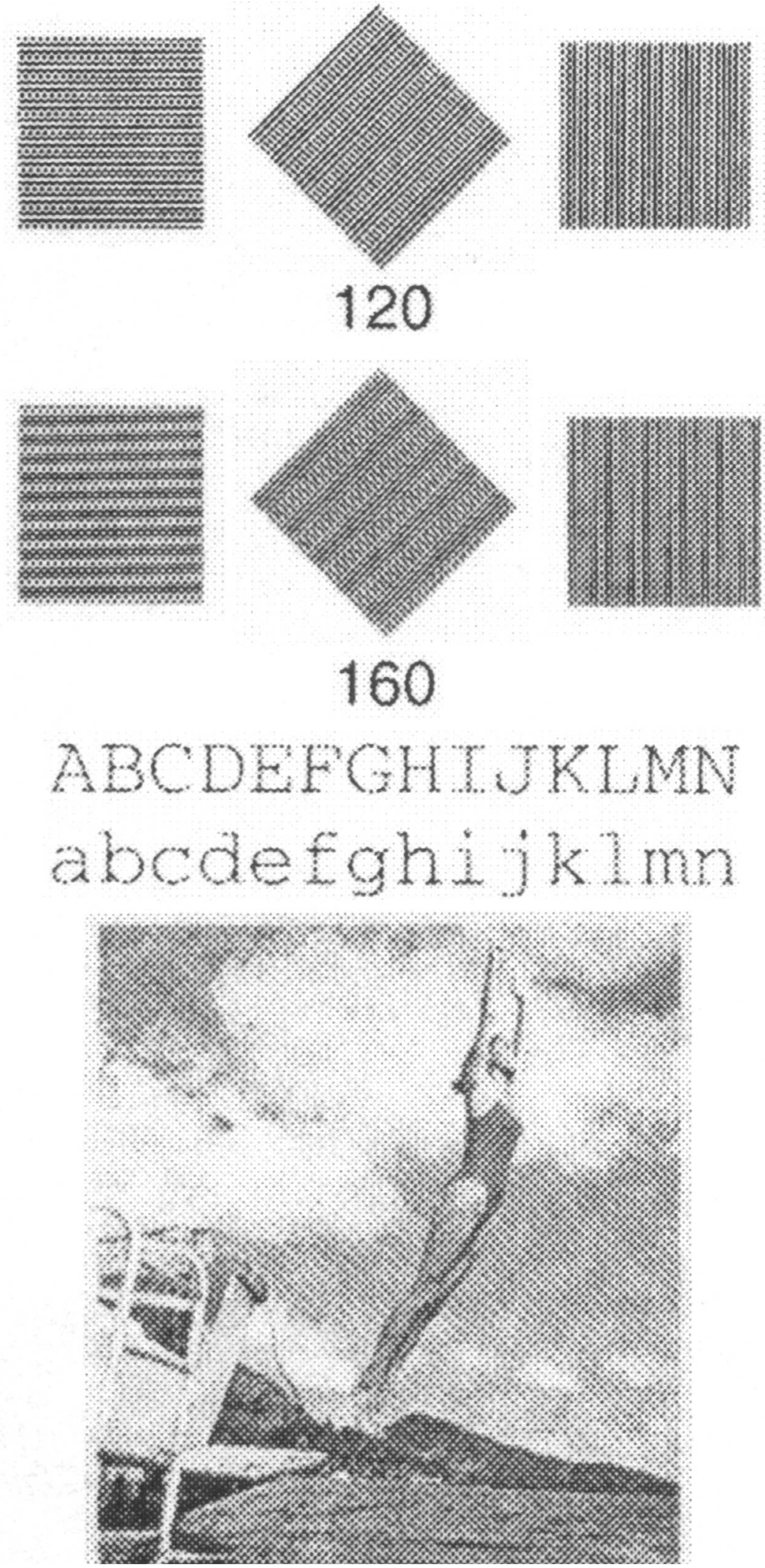


Figure 25: Rendering of a scanned image using PhotoTone cluster dot screening.