

Methods for Generating Blue-Noise Dither Matrices for Digital Halftoning

*Kevin E. Spaulding, Rodney L. Miller and Jay Schildkraut
Eastman Kodak Company*

*Imaging Research and Advanced Development, Rochester, New York 14650
E-mail: spaulding@kodak.com*

Abstract

Blue-noise dither halftoning methods have been found to produce images with pleasing visual characteristics. Results similar to those generated with error-diffusion algorithms can be obtained using an image processing algorithm that is computationally much simpler to implement. This paper reviews and compares the various techniques that have been used to design blue-noise dither matrices. In particular, a series of visual cost function based methods, a several techniques that involve designing the dither matrices by analyzing the spatial dot distribution are discussed. Ways to extend the basic blue-noise dither techniques to multilevel and color output devices are also described, including recent advances in the design of jointly optimized color blue-noise dither matrices.

1 Introduction

Many printing devices are binary in nature, and are therefore incapable of producing continuous tone images. Digital halftoning techniques are used to create the appearance of intermediate tone levels by controlling the spatial distribution of the binary pixel values. Ordered-dither, sometimes referred to as periodic dither, is a simple digital halftoning algorithm that has been used for many applications. The origin of this technique can be traced back at least as far as 1969.^{1,2} One common implementation of the basic ordered-dither algorithm is shown in Fig. 1. In this case, a dither value $d(x_d, y_d)$ is determined by modularly addressing a “dither matrix” with the row and column addresses of the image pixels. The size of the dither matrix in this example is $M_x \times M_y$. The dither value is then used to threshold the input continuous tone image value $I(x, y)$ to determine a halftoned output value $O(x, y)$. If the input value is greater than the dither value, the output pixel is set to “on.” Conversely, if the input value is less than the dither value, the output pixel is set to “off.” The modulo operations have the effect of tiling the dither matrix across the image in a repeating pattern.

There are several other implementations of the basic ordered-dither algorithm that are commonly used. One variation, shown in Fig. 2, is quite similar to that shown in Fig. 1, except that the dither value is added to

the input value and compared to a fixed threshold, instead of using the dither value itself as the threshold. It can easily be shown that these two implementations are mathematically equivalent, however, in some cases, the second approach may be more computationally efficient.

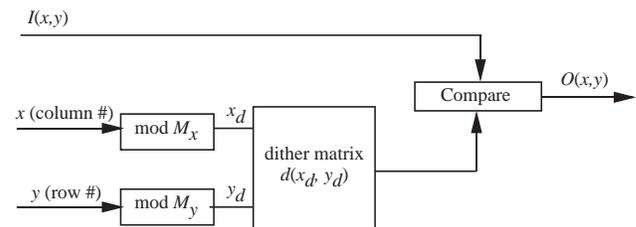


Figure 1. Flow diagram for basic ordered-dither algorithm.

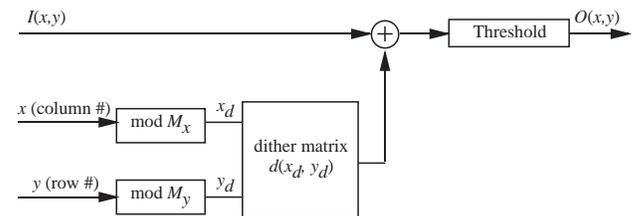


Figure 2. Flow diagram for additive ordered-dither implementation.

A third implementation of the ordered-dither algorithm is shown in Fig. 3. In this case, instead of storing a dither matrix, a set of bitmaps are stored corresponding to the halftone pattern that should be used for each gray level. The input value is used to select one of the bitmaps, and the pixel row and column address are used to modularly address the selected bitmap to determine the output pixel value. It can be seen that this implementation has the advantage that fewer computations are required to process each pixel, at the expense of a larger memory requirement. This method can be used to obtain results that are equivalent to those obtained in the first two implementations if the bitmaps correspond to those that would be obtained by thresholding the dither matrix at each of different input values.

Note that any series of halftone patterns that can be generated with the methods of Figs. 1 and 2, can also be

generated with the method of Fig. 3. However, the reverse is not true. This is because in the dither matrix implementation the halftone patterns are forced to be correlated so that once a certain pixel in the dither pattern has been turned on at some input value, it will stay on for all higher input levels. With the dither bitmaps approach, this constraint is not present. For example, a certain pixel could be “off” for an input value of 150, “on” for an input value of 151, and “off” again for an input value of 152. In practice, the limitations associated with the methods of Figs. 1 and 2 do not pose any serious problems because it is usually desirable, from an image quality point of view, to use correlated patterns that can be implemented either way. The basic equivalence between the bitmap and threshold matrix implementations for correlated halftone patterns was noted by Granger *et al.*³

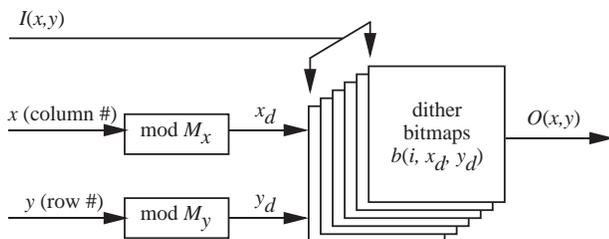


Figure 3. Flow diagram for ordered-dither bitmap implementation.

The number of unique entries in the dither matrix determines the number of gray levels, which can be produced in the output image. For example, if 16×16 dither matrices are used, it is possible to produce 257 ($M_x M_y + 1$) gray levels. If smaller matrices are used, the period of the dither pattern is smaller, but the number of producible gray levels falls correspondingly. However, despite the loss of gray levels, the reduction in the period of the dither patterns can sometimes provide an image quality advantage, particularly when using a clustered-dot dither matrix. A variation of the ordered-dither approach has recently been developed to allow the use of smaller dither matrices while maintaining the full number of gray levels at the expense of adding a small amount of randomness to the image.⁴

The arrangement of the values stored in the dither matrix determines the dot pattern that is formed at each gray level. Clustered-dot dither matrices represent one class of dot patterns. This approach attempts to mimic the halftone dot patterns produced by conventional graphic arts halftone screens. Another type of dither matrix is the well-known “Bayer matrix.”⁵ This solution was designed to maximize the lowest spatial frequency in the halftone pattern for any given gray level. The idea being that high frequencies are less visible to a human observer than low frequency values. It can be seen that the Bayer matrix is closely related to the deterministic dither described by Limb.¹ One characteristic of the Bayer matrix is that the perceived texture associated with the halftone patterns changes quite dramatically as a function of gray level. As a result, images having smooth gradations are apt to exhibit texture contours. The Bayer

matrix represents an example of a class of dither matrices sometimes referred to as “dispersed-dot dither.” The general characteristic of dispersed dot dither matrices is that they avoid placing dots near each other whenever possible.

A particular form of the dispersed-dot ordered-dither technique known as “blue-noise dither” is the topic of the current paper. This approach was first reported by Sullivan *et al.*,⁷ and has since been investigated by many other researchers including Mitsa and Parker,⁸ and Ulichney.⁹ Blue-noise dither techniques are related to the methods of Limb¹ and Bayer,⁵ in that they are all directed towards reducing the visibility of the halftone patterns by controlling the frequency spectrum of the spatial modulation. Generally, blue-noise dither matrices are designed to minimize the low frequency content and maximize the high frequency content of the halftone patterns. (The term “blue-noise” originates from the fact that blue light corresponds to the high frequency portion of the visible spectrum.) Because the human visual system is less sensitive to high spatial frequencies, blue-noise dither patterns are less visible to a human observer. Generally, the halftone patterns associated with blue-noise dither are significantly less visible than clustered-dot dither, and do not have the annoying texture changes associated with Bayer dither.

Blue-noise dither patterns are actually quite similar to the halftone patterns that are produced with error diffusion halftoning algorithms, but the implementation is computationally more efficient. In fact, error diffusion methods have sometimes been referred to as “dithering with blue-noise”,¹⁰ which can be a source of some confusion. In the graphic arts halftoning field, the terms “stochastic screening” and “frequency-modulation screening” have been used to describe both blue-noise dither techniques, as well as error-diffusion methods.

2 Review of Blue-Noise Matrix Generation Methods

This section reviews the various techniques that have been used to generate blue-noise dither patterns.

2.1 Minimum Visual Cost Techniques

Sullivan, and Miller and Sullivan^{6,7} have described a series of methods for designing blue-noise dither matrices based on the use of stochastic optimization methods to minimize a visual cost function.

2.1.1 Uncorrelated Patterns. In their initial investigation, the halftone patterns designed by Sullivan *et al.* were optimized for individual gray levels to be used for tint fills of uniform image regions. The halftone patterns were determined using stochastic annealing techniques to minimize a cost function based on the frequency response of the human visual system. The halftone patterns for each gray level were uncorrelated with the halftone patterns for any other gray level. Because the patterns were uncorrelated, it was necessary to use an implementation like that shown in Fig. 3 to apply the patterns to an image.

The cost function used to estimate the visibility of the halftone patterns is given by

$$\text{cost} = \iint |P(f_x, f_y) V(f_x, f_y)|^2 df_x df_y, \quad (1)$$

where $p(f_x, f_y)$ is the frequency spectrum of the halftone pattern, and $V(f_x, f_y)$ is the frequency response of the human visual system. It can be seen that the effect of this cost function is to weight the frequency content of the halftone pattern by the ability of the human visual response to detect it. Thus halftone patterns that move the frequency content to a region of the frequency spectrum where the human visual system is less sensitive will have a correspondingly lower cost.

A model of the low-contrast photopic modulation transfer function was used to characterize the human visual system:

$$V(f_x, f_y) = \begin{cases} a(b + c\tilde{f}) \exp[-(c\tilde{f})^d] & \text{if } f > f_{\max}, \\ 1.0, & \text{otherwise} \end{cases} \quad (2)$$

where the constants a , b , c , and d are calculated from empirical data to be 2.2, 0.192, 0.114, and 1.1 respectively, \tilde{f} is the normalized radial spatial frequency in cycles/degree of visual subtense, and f_{\max} is the frequency at which the weighted exponential peaks. To account for angular variations in the human visual function sensitivity, the normalized radial spatial frequency is computed from the actual radial spatial frequency using an angular-dependent scale function

$$\tilde{f} = \frac{f}{s(\theta)}, \quad (3)$$

where $f = (f_x^2 + f_y^2)^{1/2}$, and $s(\theta)$ is given by

$$s(\theta) = \frac{1-w}{2} \cos(4\theta) + \frac{1+w}{2}, \quad (4)$$

with w being a symmetry parameter, and

$$\theta = \arctan\left(\frac{f_y}{f_x}\right). \quad (5)$$

A plot of this visual MTF function is shown in Fig. 4, which illustrates the low-pass nature of visual system, and the reduced sensitivity at 45°. To apply this MTF function for a specific viewing distance and dot pitch, the angular frequency values must be related to the corresponding spatial frequency values using straightforward geometrical relationships.

Because the dither pattern is specified for a discrete set of pixel values, a discrete form of Eq. (1) was used to compute the cost

$$\text{cost} = \sum_{i=0}^{M_x-1} \sum_{j=0}^{M_y-1} |P_{ij} V_{ij}|^2, \quad (6)$$

where P_{ij} is the (i,j) th element of the discrete Fourier transform of the halftone pattern, and V_{ij} is the corre-

sponding human visual system sensitivity. The fundamental periodic nature of the discrete Fourier transforms properly accounts for the fact that the dither patterns are applied in a repeating tiled fashion.

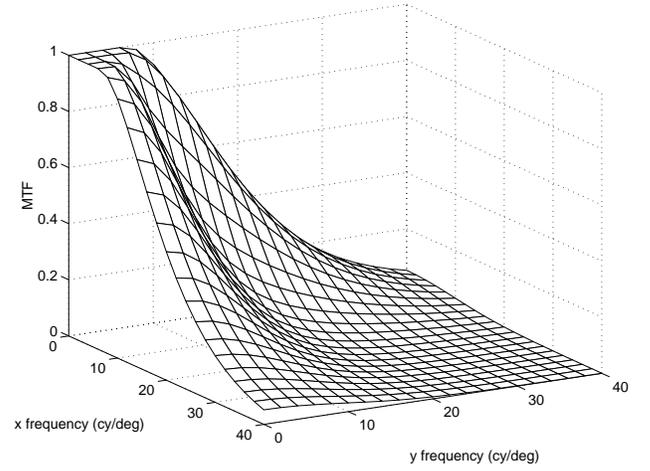


Figure 4. Low-contrast photopic modulation transfer function of human visual system.

Ideally, the cost for every possible arrangement of the dots in the halftone pattern could be computed and the halftone pattern with the lowest cost could then be selected. However, in practice, the number of possible arrangements is so large that this is impossible. As a result, it is necessary to use combinatorial optimization techniques such as stochastic annealing^{11,12} or genetic algorithms^{13,14} to minimize the cost function. For most of the blue-noise dither optimization work described in this section, the stochastic annealing approach was used. A flowchart showing an implementation of the stochastic annealing process for optimizing halftone patterns is shown in Fig. 5. The algorithm comprises the following steps:

1. Define an initial halftone pattern having the desired number of dots, and compute a corresponding initial cost value. The initial halftone pattern can be formed randomly, or alternatively some predefined halftone pattern can be used. The initial cost is temporarily labeled as the “old cost.”
2. Randomly select a pair of pixels in the halftone pattern and interchange them to form a new halftone pattern.
3. Compute a cost value for the new halftone pattern.
4. Calculate a random number z between 0 and 1, and the Boltzmann test statistic q ,

$$q = \exp\left(-\frac{\Delta \text{cost}}{T}\right), \quad (7)$$

where Δcost is the new cost minus the old cost, and the parameter T is set initially so that a large percentage, e.g., 80%, of new halftone patterns are accepted in the following step.

5. Compare q to z . If $q > z$ the new halftone pattern is retained and the new cost computed in step 3 is re named as the old cost. if $q \leq z$, the halftone pattern is returned to its previous state.
6. After many iteration of steps 2-5 above, e.g., 1000, reduce the parameter T to κT , where $\kappa < 1$, e.g., $\kappa = 0.95$.
7. When T is sufficiently small so that the costs at successive values of T are no longer changing significantly, or after a fixed number of changes have been made to T , e.g., 500, the process is complete and the final halftone pattern is stored.

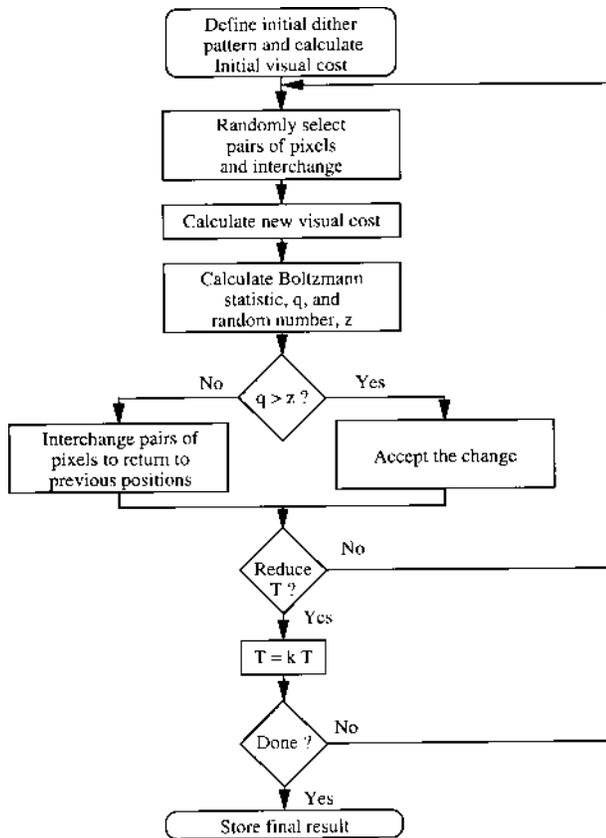


Figure 5. Flowchart for stochastic annealing algorithm.

In the initial investigation of Sullivan *et al.*, the stochastic annealing process was repeated independently for every gray level to form a set of uncorrelated blue-noise halftone patterns. Because of the large number of iterations that were required to optimize the patterns, powerful computers were necessary to determine solutions in a reasonable amount of time. Even with the use of supercomputers, it can still take several hours or more to design a set of patterns for even modest size (16×16) dither matrices. Figure 6 shows an example of an initial randomly designed pattern, compared with an optimized pattern determined using the above process. In this case, a 128×128 bitmap was computed. The corresponding angularly averaged frequency spectra for these two patterns are shown in Fig. 7. It can be seen that the low frequency content of the optimized pattern is greatly reduced relative to that of the random pattern. Figure 8

shows the value of the visual cost function at successive temperatures in the optimization process. It can be seen that the stochastic annealing process gradually reduces the visual cost until a plateau is reached where the cost cannot be improved further.

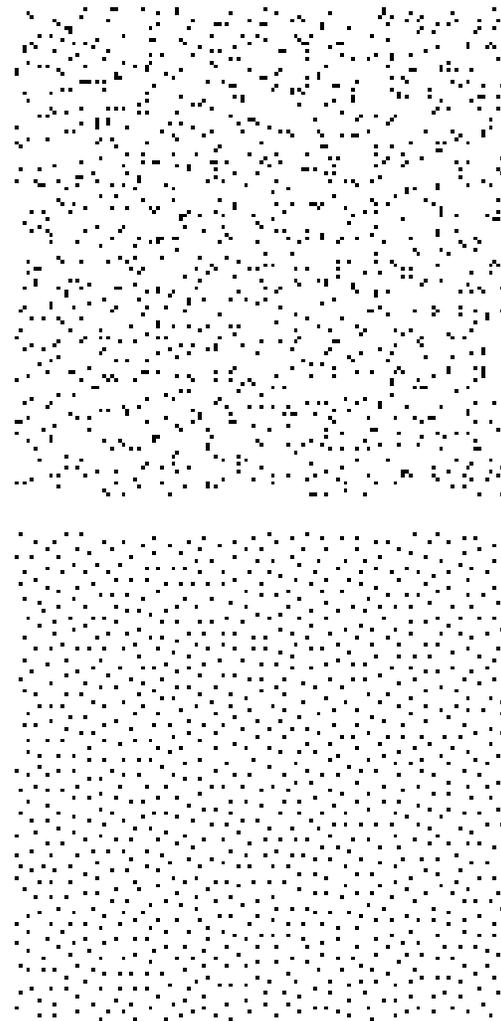


Figure 6. Pixel positions of (a) initial random pattern, and (b) final minimum cost pattern.

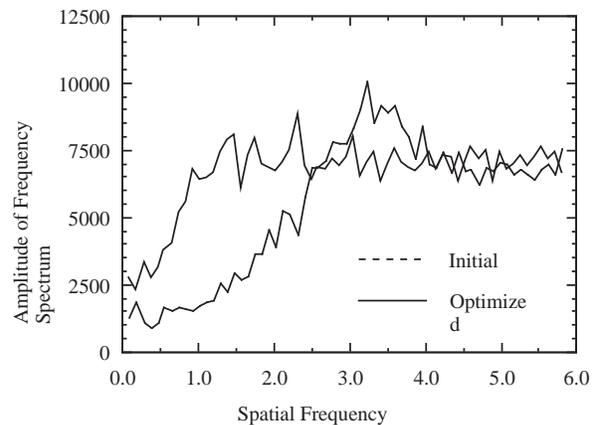


Figure 7. Amplitude of frequency spectrum averaged over all angles for (a) initial random pattern, and (b) final minimum cost pattern.

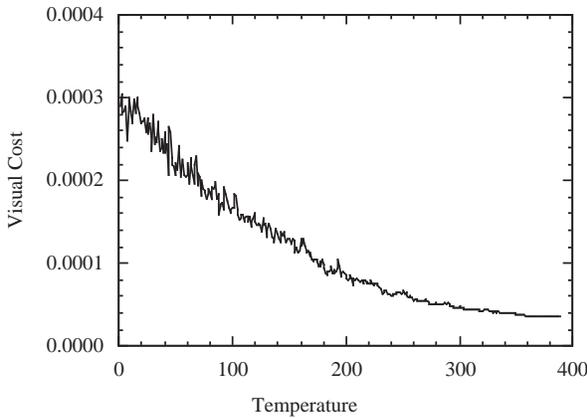


Figure 8. Visual cost at successive temperatures in the stochastic annealing optimization process.

Other types of cost functions can also be used to estimate the visibility of the halftone patterns. One approach is to calculate the variance of the visually weighted spectral powers rather than summing them as in Eq. (6). This has the effect of trying to suppress any strong spikes in the perceived power spectrum. Another approach would be to use a spatial channel model of the visual system to determine the visual cost.¹⁵

Although the results obtained using this method are quite pleasing when applied to uniform regions, they do not produce desirable results when applied to continuously varying images. This can be traced to the uncorrelated nature of the halftone patterns where the pattern at one gray level has been derived independent of all other gray levels. As a result, objectionable dot patterns are formed in regions of the image that contain smooth gradations from one gray level to another.

2.1.2 Correlated Patterns. Realizing the limitations of the uncorrelated blue-noise patterns, Sullivan and Ray later generalized the blue-noise dither pattern generation techniques to produce correlated blue-noise patterns.¹⁶ Because of the correlated nature of these patterns, they produced much better results when applied to continuous tone images, and additionally they could be implemented using the conventional dither matrix techniques shown in Figs. 1 and 2.

The determination of the correlated halftone patterns becomes somewhat more complicated than the uncorrelated case due to the fact that the halftone pattern for one level must be designed taking into account the halftone patterns for the other levels. The first attempts to compute correlated blue-noise halftone patterns used a sequential optimization approach to determine the patterns for each level one at a time. Although later methods were developed to optimize all of the levels simultaneously, the sequential approach still holds significant advantages from a computation time point of view.

A flowchart showing the basic process for sequentially optimizing correlated blue-noise dither patterns is shown in Fig. 9. The steps of this process are as follows:

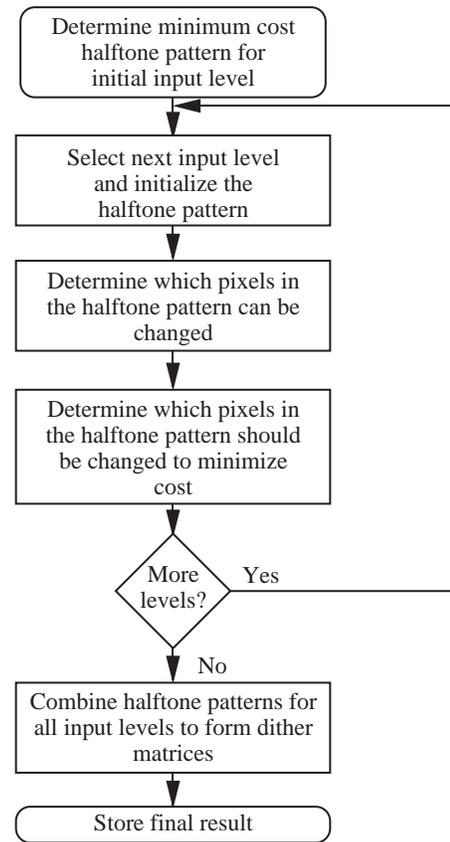


Figure 9. Flowchart for computing correlated blue-noise patterns using sequential optimization

1. Form an optimized pattern for an initial input level. Generally this can be done using the stochastic annealing method described in Sec. 2.1.1 for the uncorrelated halftone patterns. The initial input level is commonly taken to be 128, although it has been found that better results can sometimes be obtained with other initial values.
2. Select the next input level in the sequence and initialize the halftone pattern. A consecutive sequence could be used where the input levels were optimized by first incrementally decreasing, and then increasing the level relative to the initial input level, (e.g., 127, 126, ..., 1, 0, 129, 130, ..., 254, 255). Alternatively, many other arrangements are possible such as a sequence based on a binary tree where the intervals are sequentially subdivided, (e.g., 64, 192, 32, 96, 140, 224, ...).
3. Determine which elements of the halftone patterns can be changed. Generally this is done by finding the pixels that are simultaneously “off” in the halftone pattern corresponding to the nearest input level that has already been optimized below the current input level, and “on” in the halftone pattern corresponding to the nearest input level that has already been optimized above the current input level.
4. Determine which changeable elements should be changed in order to minimize the visual cost. This can be done using stochastic annealing or other optimization techniques.

5. Repeat steps 2-4 for each of the input levels.
6. Combine the resulting halftone patterns for all of the input levels into a corresponding dither matrix and store the result.

Because the levels are determined sequentially, the levels that are optimized earlier in the sequence generally have a higher quality (lower cost) than those determined later in the sequence. This is because the decisions about pixel placement at one level constrain the choice of where pixels can be placed for subsequent levels. This is the motivation for using a binary tree sequence. For example, consider the case where an initial level of 128 is used. If the levels are optimized incrementally, as many as 128 previous optimizations may constrain the optimization for a single level. With the binary tree approach, the maximum number of previous optimizations that affect the current optimization will be much smaller.

The blue-noise dither matrices that were initially computed using this approach produced much better results than those obtained using the uncorrelated patterns. However, the images were still somewhat marginal in quality. One of the principle reasons for this is that the practical dither matrix size was limited to approximately 16×16 due to computational constraints. With this matrix size, it was not possible to totally eliminate the appearance of low-frequency periodic artifacts. The use of larger dither matrices not only reduces the fundamental frequency of the halftone pattern, but additionally provides more degrees of freedom during the optimization process.

An effort was made to speed up the optimization process to make it possible to compute larger dither matrices. The use of more efficient programming techniques and faster computers provided some improvement, but by far the largest speed gains came by moving the computation of the cost function from the frequency domain to the spatial domain. By Parseval's Theorem it can be shown that the cost function given in Eq. (1) can be rewritten as

$$\text{cost} = \iint |p(x,y) * v(x,y)|^2 dx dy, \quad (8)$$

where $*$ represents convolution, $p(x,y)$ is the halftone pattern, and $v(x,y)$ is the inverse Fourier transform of the human visual system sensitivity, which can be interpreted as the human visual system point-spread function (PSF).

The corresponding discrete form of the spatial domain cost function is

$$\text{cost} = \sum_{i=0}^{M_x-1} \sum_{j=0}^{M_y-1} |(p * v)_{ij}|^2, \quad (9)$$

where $(p * v)_{ij}$ is the (i,j) 'th element of the perceived halftone pattern given by discrete convolution of the halftone pattern with the human visual function system PSF. Noted that the halftone pattern generated by an ordered-dither process will be periodic, hence the discrete convolution must take this into account by including contributions from surrounding dither arrays. In practice, this is most easily done by computing an effective

visual system PSF that is the sum of a series of visual system PSFs that repeat at the dither pattern period

$$v_e(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} v(x - mM_x\Delta x, y - nM_y\Delta y), \quad (10)$$

where Δx and Δy are the pixel spacing in the x and y directions, respectively. One method to conveniently determine the effective PSF is to perform an inverse FFT of the human visual function MTF. The periodic boundary condition assumptions associated with the FFT, naturally account for the repeating nature of the effective PSF. An example of an effective PSF is shown in Fig. 10. The x and y axes in this case give the pixel offset values. It can be seen that the same PSF value will be obtained when $x = 1$ and when $x = 15$, which reflects the fact that the dither pattern repeats every 16 pixels. Thus $x = 1$ is 1 pixel to the right of a dot centered at $(0,0)$, and $x = 15$ is 1 pixel to the left of the same dot in the next period of the dither pattern. An equivalent result can be obtained by performing the summation given in Eq. (10), but a large number of terms must be included to obtain accurate results.

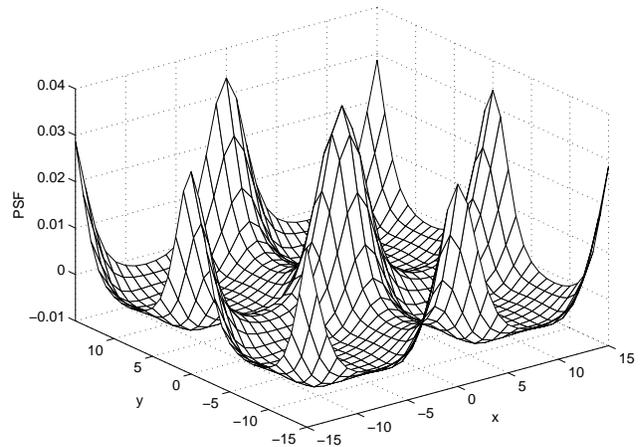


Figure 10. Effective point spread function for 16×16 dither matrix. Viewing distance was assumed to be 30 inches, and printer was assumed to be 300 dpi.

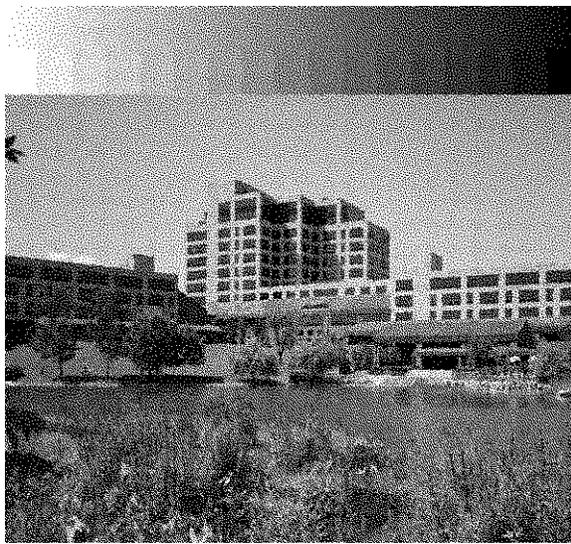
Generally, one would expect it to be faster to calculate the Fourier transform required to compute the cost function using Eq. (6), rather than the convolution necessary to compute the cost function using Eq. (9). However, in this case there are several factors that make it possible to obtain much faster results with the convolution. First of all, since the halftone pattern consists only of ones and zeros, the multiplications normally associated with the computation of a convolution can be eliminated. In this case, the convolution simply becomes a summation of a set of point-spread functions centered at the dot positions. Secondly, during the optimization process the convolution can quickly be recalculated as the bitmap is perturbed, since it is only necessary to modify the contributions to the summation corresponding to the pixels that are changed. The move from the frequency domain cost computation to the spatial domain cost computation resulted in more than a $100\times$ improvement in

computation speed. It is currently possible to compute a 128×128 solution in about 2 days on a DEC Alpha 2100-4/275 computer. With the frequency domain approach it would only have been possible to compute a 16×16 matrix in that same time frame.

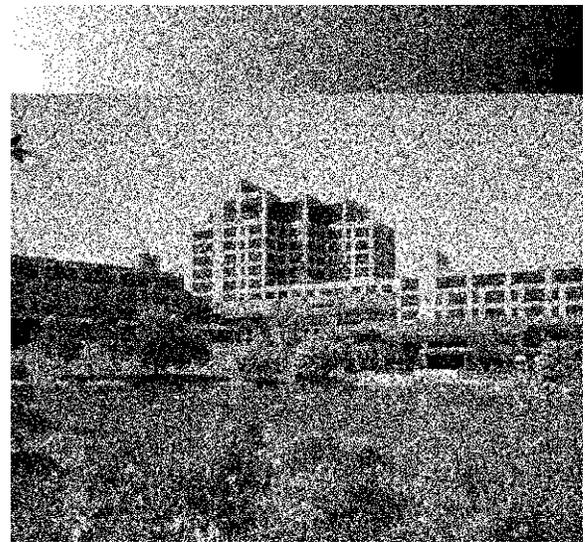
One approach that can be used to speed up the computation of the blue-noise patterns further is to use a steepest-descent type of optimization technique, rather than a stochastic optimization technique. In this case, instead of randomly adding/subtracting the dots to eventually minimize the visual cost, a visual cost value is determined corresponding to every possible place that a dot can be added/subtracted. The position corresponding to the lowest cost position is then chosen. For this approach, the initial pattern for the first level will generally need to be determined using a stochastic optimization technique, or some alternate approach, but the remainder of the halftone patterns can be determined by sequentially adding/subtracting dots that minimize the

visual cost. The advantage of this steepest-descent approach is that the computation time is greatly reduced. However, the disadvantage is that the solution does not necessarily represent a global minimum for the visual cost function.

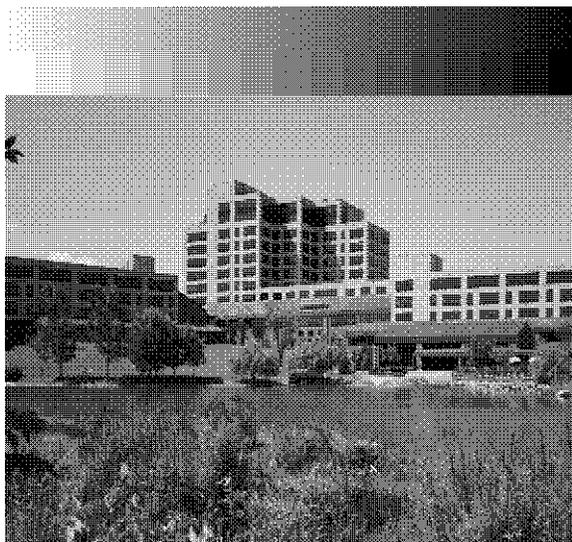
A sample image produced using a 128×128 blue-noise matrix is shown in Fig. 11(a). The dither matrix used to make this image was generated by using stochastic annealing to determine an optimized halftone pattern for gray level 100. The steepest descent optimization approach described above was then used to sequentially optimize the remaining gray levels. For comparison, images generated using a 128×128 random (white-noise) dither matrix, a 16×16 Bayer dither matrix, and a 16×16 clustered-dot dither matrix are shown in Figs. 11(b) to 11(d). (The clustered-dot dither pattern was created by tiling an 8×8 pattern into a 16×16 matrix and adjusting the thresholds to obtain 256 unique values.)



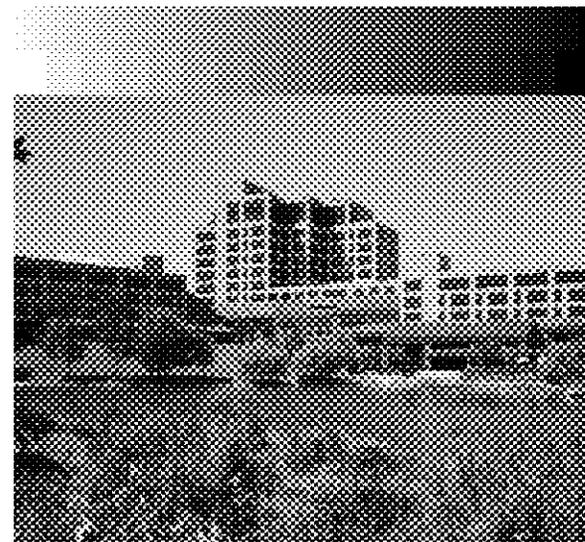
(a)



(b)



(c)



(d)

Figure 11. Images generated using (a) 128×128 blue-noise dither matrix, (b) 128×128 random (white-noise) dither matrix, (c) 16×16 Bayer dither matrix, and (d) 16×16 clustered-dot dither matrix.

The visual cost as a function of gray level for these matrices is shown in Fig. 12. The visual cost was computed using Eq. (9) with a viewing distance of 20 in., and a dot pitch of 300 dpi. For purposes of comparison the smaller Bayer and clustered-dot matrices were tiled to a 128×128 size before computing the cost values. It can be seen that the clustered-dot and random dither matrices have the highest visual cost. This is consistent with the overall higher visibility of the halftone patterns for the corresponding images. It should be noted that although the visibility of the white-noise and clustered-dot halftone patterns may be similar, the clustered-dot patterns are generally found to be less objectionable. This demonstrates that the visual cost may not always be the only factor that influences the visual objectionability.

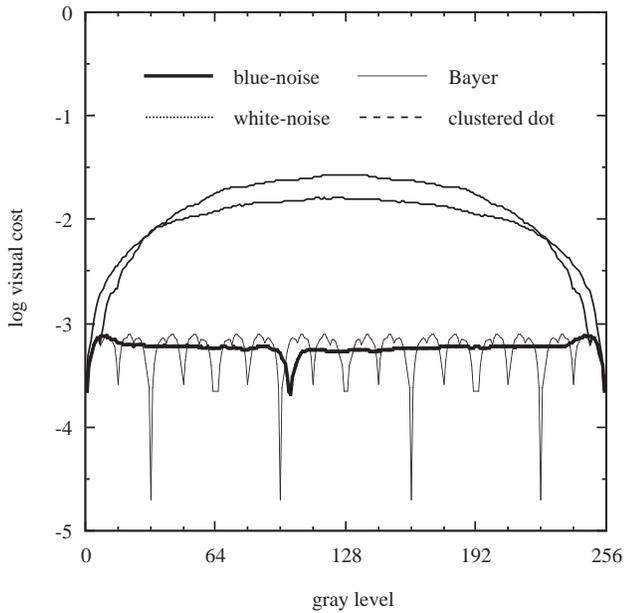


Figure 12. Visual cost as a function of gray level for the dither matrices used in Fig. 11.

Overall, the visual costs for the Bayer matrix and the blue-noise matrix are substantially lower than the visual costs for the clustered-dot and white-noise patterns. Although the average visual cost levels are quite similar, it can be seen that the visual cost for the Bayer matrix fluctuates much more widely with gray level than does the visual cost for blue-noise matrix. This reflects the fact that the Bayer matrix produces distinct changes in the texture as a function of gray level. At the gray levels where very regular patterns are formed, the cost for the Bayer matrix is lower than that of the blue-noise matrix, however, at intermediate gray levels, the Bayer matrix gives a higher visual cost. The visual cost for the blue-noise matrix can be seen to be much more uniform than that of the Bayer matrix, reflecting the fact that the overall texture is quite constant with gray level. There is a small dip in the visual cost near the gray level of 100, which corresponds to the starting gray level for the optimization process. This results from the fact that the optimization process cannot move the dots that are al-

ready placed due to the correlated nature of the halftone patterns. The patterns must therefore be slightly less optimal than the solution for the first level. Most people would judge that the blue-noise matrix result is preferable to the Bayer matrix result because of the more consistent image texture characteristics.

2.1.3 Simultaneously-Derived Correlated Patterns. Another related approach for computing correlated blue-noise dither matrices has been developed.¹⁷ In this case, the dither patterns for all the gray levels are computed simultaneously rather than sequentially. The advantage of this approach is that it will not penalize the image quality for patterns that are computed later in the sequence. To accomplish this, a cost function is defined that combines the visual cost for all of the different gray levels. One of the simplest forms for the combined cost function is

$$C = \left[\sum_{t=0}^{255} (C_t)^p \right]^{1/p}, \quad (11)$$

where C_t is the visual cost for a particular tone level t , and p is a positive constant. If p is taken to be 2, this result is simply the RMS visual cost for all threshold levels. Larger values of p have the effect of weighting the cost values for tone levels with larger errors more strongly.

One problem with the combined cost function given in Eq. (11) is that the fundamental visual cost can vary quite widely as a function of tone level. In particular, tone levels near the dark and light end of the tone scale typically have smaller cost values than midtone levels. As a result, the cost function preferentially treats the tone levels with higher cost values. One approach that can be used to minimize this effect is to normalize the cost values before they are combined

$$C = \left[\sum_{t=0}^{255} \left(\frac{C_t}{\langle C_{t0} \rangle} \right)^p \right]^{1/p}, \quad (12)$$

where $\langle C_{t0} \rangle$ is the average cost calculated for a series of random variations of the matrix elements.

The optimization process used for the simultaneously derived patterns can be very similar to that described above to determine optimized patterns for individual gray levels. For example, the stochastic annealing process shown in Fig. 5 can be used with only small modification:

1. Define an initial dither matrix and compute the corresponding initial combined cost value. The initial dither matrix can be formed randomly, or alternatively some predefined dither matrix can be used. The initial combined cost will temporarily be known as the "old total cost."
2. Randomly select a pair of elements in the dither matrix and interchange them to form a new dither matrix.
3. Compute a new combined cost value for the new dither matrix.

4. Calculate the Boltzmann test statistic q , and a random number z between 0 and 1.
5. Compare q to z . If $q > z$ the new dither matrix is kept and the new total cost computed in step 3 is renamed as the old total cost. If $q \leq z$, the dither matrix is returned to its previous state.
6. After many iteration of steps 2-5 above, e.g., 1000, reduce the parameter T to κT , where $\kappa < 1$, e.g., $\kappa = 0.95$.
7. When T is sufficiently small so that the total costs at successive values of T are no longer changing significantly, or after a fixed number of changes have been made to T , e.g., 500, the process is complete and the final dither matrix is stored.

While the simultaneous optimization of the different tone levels would seem to be advantageous in theory, there are several practical factors that limit the usefulness of this technique in practice. First, the computational complexity of the joint cost function slows the optimization process substantially. This puts a practical limitation on the size of the dither matrix that can be computed. With current computer resources a 64×64 dither matrix takes as long as a week to converge. Additionally, the optimization process seems to be more prone to getting caught in a local minimum of the cost function than the sequential optimization methods. For example, a 16×16 solution was computed using this technique that yielded moderate image quality. A 32×32 solution was then computed that produced a lower perceived image quality. To compare the visual costs for the two solutions, a 32×32 matrix was made by tiling together four copies of the 16×16 matrix. The total cost for the replicated 16×16 was substantially lower than that of the actual 32×32 solution. Because a lower cost solution was shown to exist, this indicates that the optimization process for the 32×32 matrix was not able to converge on the global minimum of the cost function. This is probably due to the high dimensionality of the problem, combined with the complexity of the cost function. It is expected that if the true global minimum could be found, this approach would give very good results. However, the optimization techniques that have been tested have been unable to get very near to the global minimum for dither matrices of reasonable size.

2.2 Spatial Dot Distribution Techniques

A number of other blue-noise halftone pattern generation methods have been developed that are based on evaluating the spatial distribution of dots in the halftone pattern, rather than the computation of a visual cost function value.

2.2.1 Iterative Constraint Techniques. An approach for designing blue-noise dither matrices has been developed by Parker and Mitsa.¹⁸ that attempts to produce halftone patterns that simultaneously satisfy two different criteria. First, the patterns should have particular power spectrum characteristics. Second, the patterns must be binary in nature, and must have the desired fractional area coverage. It is quite simple to design patterns that meet one or the other of these criteria, but not both of them at the

same time. To determine halftone patterns that simultaneously meet both requirements, an optimization technique that is similar to a “projection-on-convex-sets” approach is used. Essentially, a set of independent constraints are cyclically applied until a convergent solution is reached.

Figure 13 shows a flow diagram of the process used to design the halftone pattern for a first gray level (typically a 50% gray).

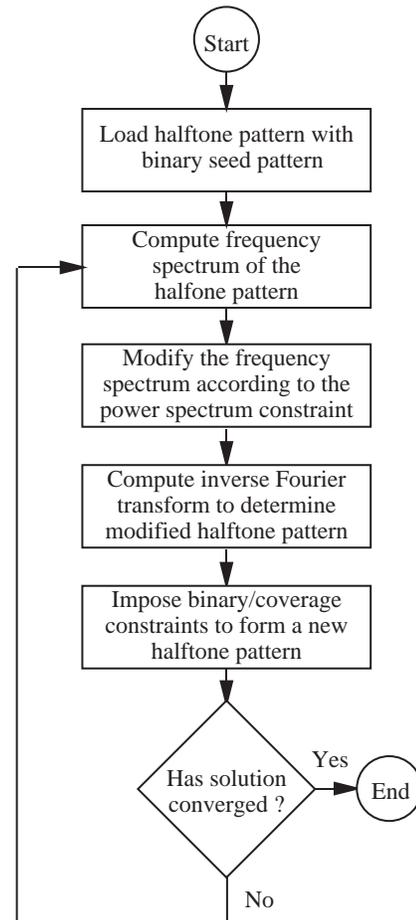


Figure 13. Iterative constraint technique for optimizing blue-noise halftone patterns.

1. First the halftone pattern is initialized with a seed pattern. Typically this would be a white-noise pattern.
2. The Fourier transform of the halftone pattern is computed to determine its frequency spectrum.
3. The frequency spectrum is modified according to a power spectrum constraint. In general, the power spectrum constraint can be used to impose any particular power spectrum characteristics to the halftone pattern. In the examples presented by Parker and Mitsa, they are attempting to produce patterns with power spectrum having no energy below a principle frequency. This principle frequency is a function of the halftone pattern gray level. By imposing this constraint they are attempting to mimic the frequency spectrum characteristics of an error diffusion pattern. They impose the constraint by comparing the

radially averaged power spectrum of the halftone pattern to the aim power spectrum to compute a frequency domain filter

$$D_r(f_r, g) = \frac{P'_r(f_r, g)}{P_r(f_r, g)}, \quad (13)$$

where f_r is the radial frequency, g is the gray level, $P_r(f_r, g)$ is the radially averaged power spectrum for the halftone pattern, and $P'_r(f_r, g)$ is the desired radially averaged power spectrum. The frequency domain filter $D_r(f_r, g)$ is applied to the frequency spectrum of the halftone pattern to determine a modified frequency spectrum.

4. Once the frequency domain constraint has been applied, an inverse Fourier transform is applied to the modified frequency spectrum to determine a modified halftone pattern.
5. This modified halftone pattern now has the desired blue-noise characteristics, however, it no longer is a binary pattern. It is, therefore, necessary to impose the constraint that the halftone pattern must be binary, and that a certain number of dots must be turned on. This is done by computing an error array that is given by the difference between the previous halftone pattern, $h(i, j)$ and the modified halftone pattern, $h'(i, j)$,

$$e(i, j) = h'(i, j) - h(i, j). \quad (14)$$

The error values are rank ordered, and the pairs of 1's and 0's corresponding to the largest error values are interchanged to form a new halftone pattern.

6. If the total mean square error for this iteration is below some threshold, the solution is accepted. Otherwise, the steps 2 to 5 are repeated using the new halftone pattern until convergence is reached.

Once the halftone pattern for the first gray level is determined, a similar method is used to sequentially determine the halftone patterns for the remaining gray levels. In this case, a single pass technique is applied, rather than the iterative approach just described. To construct the blue-noise pattern for the next highest gray level above a previously computed pattern, the previous solution is used as seed level and steps 2-4 of the above method are used to determine a modified halftone pattern. As before, an error array $e(i, j)$ is computed, and the errors are rank ordered. The number of 1's that must be added to the halftone pattern to produce the desired gray level are determined. The 0's in the previous solution corresponding to the highest error values are replaced with 1's to form a new halftone pattern. These steps are repeated to determine the halftone patterns for each gray level above the initial gray level. The gray levels below the initial gray level are computed in a similar fashion. The only difference being that ones in the previous halftone pattern are replaced by 0's. Throughout this process, the aim power spectrum is adjusted at each gray level to account for the change in the principle frequency.

A small modification to this basic technique was proposed by Yao and Parker.¹⁹ In this case, instead of

adaptively computing the filter necessary to produce the desired power spectrum as was shown in Eq. (13), a fixed filter was used at each gray level. In general, a smoothly varying low-pass filter was used, such as a Butterworth filter, or a Gaussian filter. The cutoff frequency of the filter is still adjusted as a function of gray level to account for the variation in the principle frequency of the halftone pattern.

Rolleston and Cohen²⁰ have also presented a similar approach where they create noise patterns with prescribed two-dimensional correlation functions for the purpose of using them as masks for digital halftoning. In this case, instead of designing the halftone patterns in a sequential fashion and afterwards combining the patterns to form a dither matrix, the entire dither matrix is optimized simultaneously. This is done by initializing the dither matrix with uniformly distributed random noise. The noise matrix is then transformed to the frequency domain, where a power spectrum constraint is applied. Generally, the application of this constraint involves simply multiplying the frequency spectrum by a binary masking function. The filtered spectrum is then transformed back to the spatial domain where a constraint is imposed that forces the noise to have a uniform probability density function. This insures that the number of dots in the resulting halftone pattern will be linear with the input tone level. These steps are then applied iteratively until convergence is reached. In this manner, it is possible to generate halftone patterns with a variety of power spectra, including those with blue-noise characteristics.

2.2.2 Void-and-Cluster Technique. Ulichney has proposed a simple technique for generating blue-noise patterns that he refers to as the "void-and-cluster" method.⁹ The general principle employed in the design of dither matrices using this technique is that large clusters or voids of dots should be avoided. The optimization process involves two basic phases. First, a dither pattern is formed for an initial gray level. Next, dither patterns are sequentially formed for the remaining gray levels.

A flow chart for the first phase of the void-and-cluster optimization process is shown in Fig. 14. The first step is to choose an initial binary dither pattern. This initial pattern may be randomly formed, or may correspond to the pattern that is obtained with an existing solution, such as the Bayer matrix. The initial pattern is then examined to find the tightest "cluster" of dots. All of the pixel locations containing 1's are considered in this process. The "1" at the location of the tightest cluster is then removed. The new pattern is then examined to find the location of the largest "void," and a "1" is inserted at this location. If the location of the largest void is the same as the location that the "1" was just removed from, then the process has converged. In this case the "1" is restored to initial position and the procedure is terminated. With each iteration, the voids should get smaller, and the clusters should get looser.

To determine the locations of the largest void and cluster, Ulichney defined a filter function that he convolved with the dot pattern. The location of the minimum and maximum values of the filtered dot pattern then define the largest void and the largest cluster, respec-

tively. As with the spatial domain cost function calculation described above in Sec. 2.1.2, it is necessary to account for the periodic nature of the dither pattern when performing the convolution. This is accomplished using a modified convolution

$$p_f(x, y) = \sum_{m=-M_x/2}^{M_x/2} \sum_{n=-M_y/2}^{M_y/2} p(m', n') f(m, n). \quad (15)$$

where M_x and M_y are the dither array dimensions, $p(x, y)$ is the dot pattern, $f(x, y)$ is the filter function, $p_f(x, y)$ is the filtered dot pattern, and

$$\begin{aligned} m' &= (M_x + x - m) \bmod M_x, \\ n' &= (M_y + y - n) \bmod M_y. \end{aligned} \quad (16)$$

Ulichney used a Gaussian filter function of the form

$$f(x, y) = \exp[-(x^2 + y^2)^{1/2} / (2\sigma^2)] \quad (17)$$

for his investigation. He found that a value of $\sigma = 1.5$ produced the best results.

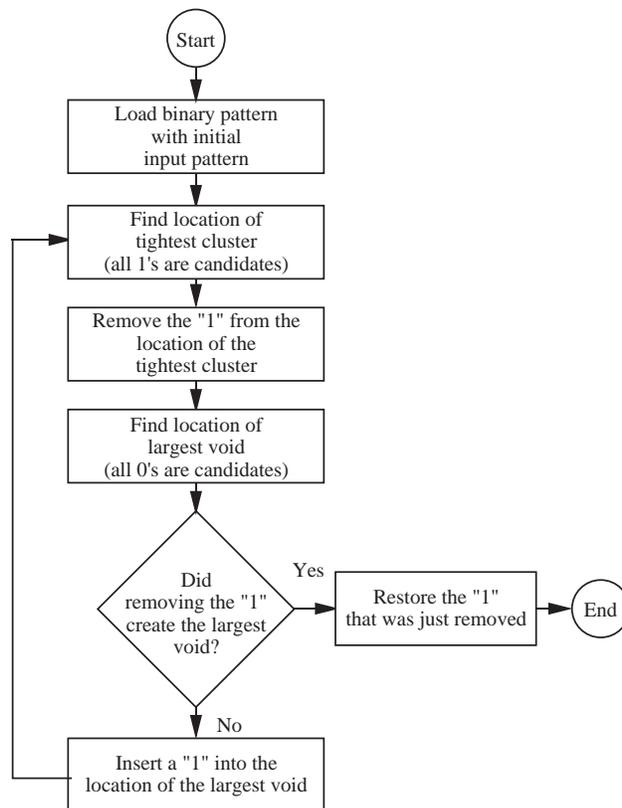


Figure 14. Flow chart for the first phase of the void-and-cluster optimization process.

Once the halftone pattern for the first gray level is determined, the next phase of the void-and-cluster approach is to sequentially compute the patterns for the remaining gray levels. A flowchart for this process is shown in Fig. 15. (It should be noted that this procedure

has been generalized slightly relative to that described by Ulichney.) First the patterns between the initial gray level and black, and then the patterns between the initial gray level and white are computed. The first step is to load an array with an initial binary pattern. The solution for the previous gray level is used as the initial binary pattern for the current gray level. Next, the number of pixels that need to be changed to produce the correct number of dots for the current gray level is determined. Depending on whether the gray level is currently being incremented or decremented, dots either need to be added or removed. If the gray level is being decremented, a number of 1's need to be turned into 0's to remove dots. Conversely, if the gray level is being incremented, a number of 0's need to be turned into 1's to add more dots. The number of dots that need to be added/subtracted will depend on the size of the dither array and the number of allowable gray levels.

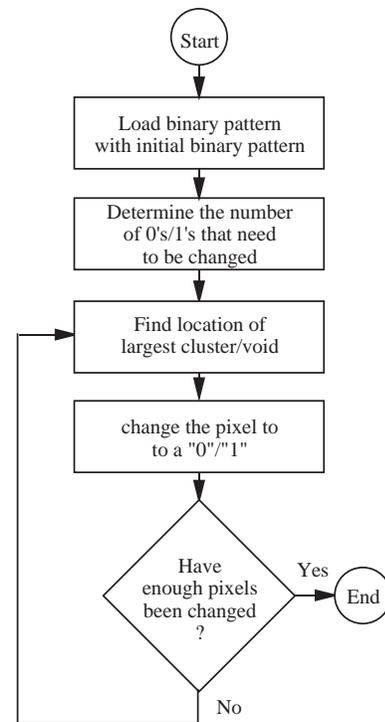


Figure 15. Flow chart for the second phase of the void-and-cluster optimization process.

The next step is to identify the location of the largest void/cluster in the dot pattern. If dots need to be added, the largest void is identified. If dots need to be removed, the tightest cluster is identified. Once the location has been found, the pixel is changed to a 1 or a 0 accordingly. This process is repeated until the required number of pixels have been changed for the current gray level. The remainder of the gray levels are then processed in an identical fashion until a complete set of halftone patterns have been determined for all of the allowable gray levels. The resulting halftone patterns can finally be combined to form a single dither matrix that can be used in any of the standard dither algorithms.

Since the void-and-cluster method is effectively a “steepest descent” optimization approach, the results that are obtained using this technique are quite dependent on the initial starting conditions. For example, if an initial binary pattern containing a single 1 is used, the final matrix is identical to the Bayer matrix solution (also known as a “recursive tessellation” solution). If randomly formed initial binary patterns are used the results are quite similar to the blue-noise patterns computed using the visual cost function methods described above. The largest advantage of the void-and-cluster approach is the computation speed of the optimization process.

It can be seen that the void-and-cluster method has much in common with Mitsa and Parker’s iterative constraint technique described in Sec. 2.2.1. In both cases, a filtered dot pattern is determined. However, Mitsa and Parker perform the filtering operation in the frequency domain, whereas a spatial domain convolution is applied for the void-and-cluster technique. Once the filtered dot pattern has been computed, it can be seen that the pixel positions having the largest errors computed for with the iterative constraint technique are analogous to the pixel positions with the largest voids and clusters determined by Ulichney.

2.2.3 Visual Potential Techniques. We have recently investigated a variation of the void-and-cluster technique that incorporates some of the desirable features of the sequential visual cost function method described in Sec. 2.1.2. The most significant difference is that the filter function is designed using a visual response model. Therefore, the largest “visible voids” and “visible clusters” in the halftone pattern will be determined. Additionally, a visual cost function method is generally used to determine the initial halftone pattern. Alternatively, a pattern from a previously determined blue-noise pattern, or even a pattern generated using an error diffusion algorithm can also be used for the initial pattern.

Once the initial pattern has been defined, a perceived halftone pattern $p_v(x,y)$ can be computed by convolving the halftone pattern $p(x,y)$ with an approximation of the point-spread function of the human visual system $v(x,y)$:

$$p_v(x,y) = p(x,y) * v(x,y). \quad (18)$$

As noted above, the convolution must take into account the periodic nature of the dither pattern. The perceived halftone pattern is analogous to the filtered halftone pattern computed for the void-and-cluster technique, except that a visual model is used for the filter function. For an “ideal” halftone pattern, the resulting perceived halftone pattern is perfectly uniform. However, for a real halftone pattern, the perceived halftone pattern is comprised of a lumpy surface having local minima and maxima corresponding to the voids and clusters in the halftone pattern.

To determine halftone patterns for the remaining gray levels in a sequential manner, it is necessary to add (or subtract) an appropriate number of dots from the halftone pattern. It is intuitive that the dots are typically added to pixel locations where the perceived halftone pattern is high, or conversely are subtracted from pixel

locations where the perceived halftone pattern is low. Using this approach to determine a halftone pattern is essentially equivalent to defining a new cost function that penalizes large peaks and valleys in the perceived luminance distribution. Although it could be argued that the final result obtained with this method may be less optimal than those obtained with the stochastic optimization methods, the quality improvements associated with the fact that larger dither matrices can be determined may outweigh any quality loss from the more simplistic optimization approach.

An appropriate name for this technique may be “visual void-and-cluster.” It has also been referred to as “electrostatic dither” because of a close analogy to an electron distribution problem. Consider a set of electrons distributed on a planar surface. An electrostatic potential can be computed by summing the potential for each of the electrons. The potential for a point charge is given by

$$\phi(x,y) = \frac{A}{R}, \quad (19)$$

where A is a constant, and $R = [(x - x_0)^2 + (y - y_0)^2]^{1/2}$, where x_0 and y_0 are the coordinates of the point charge. The overall potential for a set of electrons is given by the sum of the potentials for each of the individual electrons. This can be seen to be a convolution of the potential for a single point charge at the origin, with a set of delta functions corresponding to the positions of the electrons. If these electrons were locked in place, and an additional electron was added to the surface, it would want to go to the location with the minimum potential. After the electron is placed, the potential function would need to be modified to account for the new electron.

It can be seen that this is analogous to the present problem where halftone dots are being positioned instead of electrons. The analogy to the potential function for a single point charge is the point-spread function for the human visual system. Since the perceived halftone pattern is given by the convolution of the halftone pattern with the visual point-spread function, this can be directly related to the total potential function for the electron distribution. A method recently described by Yu *et al.* also uses an “electrostatic force” model to optimize blue-noise dither patterns.²¹

2.2.4 Other Void-and-Cluster Variants. A number of other variations of the void-and-cluster technique have also been investigated. Lin has developed a void-and-cluster based algorithm that uses a filter function that varies with gray level.²² In particular, the width of the filter function is varied to account for the fact that the average distance between the minority pixels changes as a function of gray level. (The minority pixels are those that occur less frequently in the halftone pattern; *e.g.*, for halftone patterns that are lighter than a 50% gray level, the minority pixels will be black.)

During the optimization process for a given gray level, the average separation between the minority pixels is determined. For gray levels near the 50% level, the average separation is relatively small. For gray levels near the extreme ends of the tone scale the average

separation is large. The average separation is calculated by dividing the total number of pixels in the halftone pattern by the number of elements with the minority value and taking the square root.

A function is determined by “trial and error” to relate the average separation value to a region of support value for the filter function that gives pleasing results when examined visually. The region of support value is related to the width of the filter function. Lin reports that the use of this variable filter function results in fewer objectionable artifacts in the halftone pattern. This approach can be compared to using a visual model where the viewing distance is varied as a function of the gray level. Tuning the filter size as a function of gray level allows the pattern to be optimized for the viewing distance that will be most susceptible to artifacts. This may be quite appropriate in practice where observers frequently will view an image from a series of viewing distances to evaluate quality.

Lin has also described a refinement of this basic method that incorporates a model of the dot structure for the printer.²³ In this case, instead of applying the filter function directly to the binary halftone pattern, a model of the dot reproduction characteristics for the particular printer being addressed is used to predict the density distribution of an image formed using the halftone pattern. The resulting density distribution is filtered to identify the largest clusters and voids. Several types of dot models were discussed including overlapping circular dots and Gaussian dots. To use this approach, it is necessary to form a higher resolution representation of the halftone image where each image pixel is broken down into a set of sub-pixels.

Barton has described another approach for designing halftone patterns that involves examining the spatial dot distribution to look for the largest voids and clusters of dots.²⁴ In this case, the halftone patterns are designed by growing the patterns from the lightest tone level to the darkest tone level. As in the methods described by Ulichney and Lin, a filter function is used to determine the pixel location where a dot should be added to the halftone pattern. (Barton refers to the filter function as a “cost function”.) Similar to Lin, Barton varies the filter function with tone level. However, instead of varying the function form of the filter itself, he varies the search radius used to determine whether a dot should be included in the summation. This is equivalent to multiplying the filter function, $f(x, y)$, by a cylinder function that varies in radius:

$$f'(x, y) = f(x, y) \text{cyl}[(x^2 + y^2)^{1/2}/R], \quad (20)$$

where R is the search radius, and

$$\text{cyl}(r) = \begin{cases} 1, & 0 < r \leq 1 \\ 0, & r > 1 \end{cases}. \quad (21)$$

To force diagonal correlation of adjacent dots in the output image, Barton introduces an angular dependent filter function

$$f(x, y) = \frac{1}{r + (C - \sqrt{2})(\sin 2\alpha)(\sqrt{2}/r)}, \quad (22)$$

where

$$r = (x^2 + y^2)^{1/2}, \quad (23)$$

C is a constant (nominally 2), and α is the relative angle from the center pixel. Because the filter function has a lower value in the diagonal direction, dots are preferentially placed in diagonal arrangements. Barton argues that this is preferable in many cases due to the dot interaction characteristics of typical printers. Qualitatively, this is also consistent with the angular dependence of the human visual system sensitivity.

3 Blue-Noise Matrices for Multilevel Halftoning

Periodic halftoning techniques can easily be generalized for use with multilevel output devices.^{25,26} Typically, such techniques are referred to as multilevel halftoning, or sometimes as multitoning. Figure 16 shows a generalization of the ordered-dither technique for application to multilevel devices. It can be seen that this is equivalent to the binary implementation shown in Fig. 2, except that the threshold operation has been replaced by a quantization operation. The dither signal is added to the input signal to form a modulated input signal. The quantizer then associates one of the allowable output levels with each of the possible values of the modulated input signal. The quantizer can conveniently be implemented as a look-up table (LUT). If a uniform quantizer is used, it can also be implemented using a divide operation, or possibly as a binary bit shift.

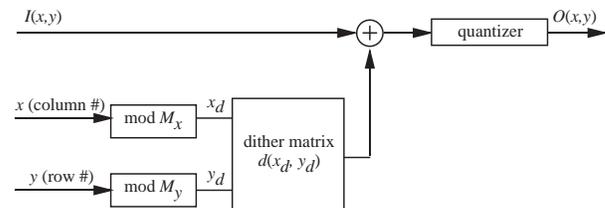


Figure 16. Basic multilevel dither implementation.

Generally, it is desirable to adjust the amplitude of the dither signal to equal the size of the quantization interval. This produces dither patterns with the minimum possible modulation level. For example, consider the case where an input image $I(x, y)$ has m different levels, and it is desired to produce an output image $O(x, y)$ having n possible output levels. If a uniform quantizer is used, the size of the quantization interval in terms of the input levels will be given by the maximum input level divided by the maximum output level: $(m-1)/(n-1)$. If the dither matrix, $d(x_d, y_d)$ stores dither values in the range 0 to $p-1$, the amplitude of the dither signal must be rescaled to match the quantization interval.

When the dither matrix is properly rescaled, the multilevel dither process can be represented by the equation

$$O(x, y) = \text{int} \left\{ \frac{I(x, y) + [d(x_d, y_d) / p](m-1)/(n-1)}{(m-1)/(n-1)} \right\}, \quad (24)$$

where $\text{INT}()$ indicates an integer truncation. It can be seen that the dither matrix value is scaled and added to the input image. The resulting modulated input value is quantized by dividing the result by the size of the quantization interval, $(m-1)/(n-1)$, and rounding down to the next integer value. Equation (24) can be rearranged into a form that is more convenient in many cases:

$$O(x,y) = \text{int} \left\{ \frac{[m/(m-1)][(n-1)/n]I(x,y) + (m/np)d(x_d, y_d)}{m/n} \right\} \quad (25)$$

If both m and n are powers of 2, the divide operation associated with the quantizer can now be accomplished using a binary bit-shift. However, the input value must now be scaled before the dither signal is added to it. This equation can be simplified somewhat by defining a series of constants

$$O(x,y) = \text{int} \left[\frac{C_1 I(x,y) + C_2 d(x_d, y_d)}{C_3} \right], \quad (26)$$

where

$$C_1 = \left(\frac{m}{m-1} \right) \left(\frac{n-1}{n} \right), \quad (27)$$

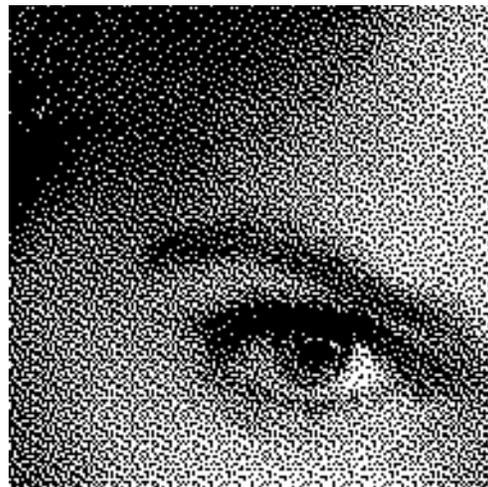
$$C_2 = \frac{m}{np}, C_3 = \frac{m}{n}.$$

In practice, for computational efficiency, the scale factor C_1 can be implemented as a LUT, the dither matrix values can be premultiplied by the value C_2 , and the divide by C_3 can be implemented using a LUT or a binary bit-shift operation.

This multilevel dither algorithm can be used to adapt any conventional binary dither matrices, including blue-noise dither matrices, for use with multilevel output devices. Figure 17 shows a series of images generated using the multilevel halftoning method given in Fig. 16. In this case, a 16×16 blue-noise dither matrix generated using the simultaneous visual cost optimization technique described in Sec. 2.1.3 was used. It can be seen that even in the magnified state, the multi-toned image approaches the quality of the original image when 16 output levels



(a)



(b)



(c)



(d)

Figure 17. Images generated using multilevel blue-noise dither algorithms with 16×16 dither matrix: (a) original, (b) 2 output levels, (c) 4 output levels, and (d) 16 output levels.

are used. When the images are printed at 300 dpi, the difference is essentially indistinguishable.

Although any blue-noise dither matrix can be used with the above implementation for multilevel output devices, the results are not necessarily optimal. Spaulding and Ray have investigated methods to generate blue-noise dither matrices that were optimized specifically for multilevel output devices.²⁷ The dither matrix design approach used in this case is very similar to the simultaneous visual cost optimization technique described in Sec. 2.1.3. The difference being that the costs for each of the levels within a quantization interval are combined, rather than the cost for all of the input levels. A slight modification to the implementation architecture was also introduced that permitted the use of non-uniform quantization functions. As shown in Fig. 18, a variable scale factor is included that permits the magnitude of the dither value to be adaptively adjusted as a function of the size of the quantization interval. In this particular implementation, the dither values are nominally stored as values in the range between $\pm 1/2$ and the scale factor is simply the size of the quantization interval. Since the size of the quantization interval may vary as a function of the input level, this value can be determined using a LUT addressed by the input code value.

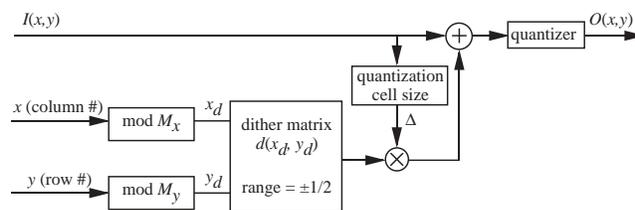


Figure 18. Multilevel dither implementation using non-uniform quantizer.

Multilevel dither can also be implemented using other algorithm architectures. One such approach, described by Miller and Smith,²⁸ and Miller²⁹ is shown in Fig. 19. In this case, the modularly addressed matrix is used to store pointers to a series of dither LUTs, rather than storing actual dither values. The results of the dither process for each of the possible input levels are pre-calculated and stored in these dither LUTs. The algorithm can now be executed with only table look-ups rather than the adds and multiplies necessary in the first implementation. Effectively, this trades off memory requirements for a faster execution speed. This approach is somewhat analogous to the binary dither arrangement shown in Fig. 3.

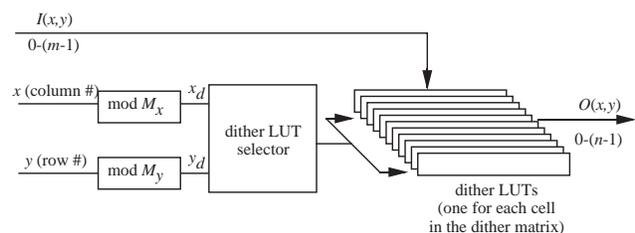


Figure 19. Multilevel dither implementation using dither LUTs.

The LUT-based approach also has the advantage that any conceivable dot growth pattern can be specified. With the conventional implementations, as the input gray level is increased all of the pixels in the dither pattern are generally increased to the second output level before increasing any of the pixels to the third output level. In some cases, it might be desirable to increase the gray level of one pixel in the dither pattern through all of its levels before starting to increase the gray level of a second pixel. This would be possible with the LUT-based approach, but would not be possible with the first implementation. Any pattern that can be created with the first implementation can also be created using the LUT-based approach, but the reverse is not true. Although blue-noise dither matrices have not been designed to specifically take advantage of this extra flexibility, the texture parameter described by Miller *et al.* can be used to design LUTs with a variety of characteristics from a conventional dither matrix. This feature can be particularly important for devices, such as electrophotographic printers, that do not produce uniform density regions very well.

4 Blue-Noise Matrices for Color Images

Many color imaging devices produce binary output for each of the color planes. Typically the color planes are red, green, and blue (RGB) for additive devices, or cyan, magenta, and yellow (CMY), or cyan, magenta, yellow, and black (CMYK) for subtractive devices. Generally halftone images for these applications are generated for each of the color planes independently following any color correction/calibration steps. It is important to note that the color reproduction characteristics of the device are usually a function of the halftoning method that is used. It is, therefore, necessary to design the color correction process knowing the halftoning method.

For the present discussion we assume that an ordered-dither algorithm is used to halftone each of the channels independently. In general, a different dither matrix can be used for each of the different color channels. Therefore, an important question is how the dither matrices used for the different color channels should be designed to maximize image quality.

4.1 Dot-on-Dot Matrices. If the same halftoning pattern is used for each of the color channels, this is known as “dot-on-dot” printing. A sample image generated using this method with a 128×128 dither matrix is shown in Fig. 20(a). Although this approach is the easiest to implement, it is rarely used in practice because it results in the highest level of luminance modulation, and additionally the color reproduction characteristics will be most sensitive to registration errors.

To understand these effects consider a neutral patch created using equal levels of cyan, magenta, and yellow. With the dot-on-dot approach, the halftone patterns produced for each color channel are identical. Therefore, whenever a cyan dot is printed, a magenta and yellow dot are also printed in the same location. As a result, the image is formed using a series of black dots on a white background. This is illustrated in the neutral regions in the image shown in Fig. 20(a). With a little thought, it

can be seen that this must produce a larger level of luminance modulation than the case where the colored dots are not coincident, therefore, the halftone patterns are generally more visible to an observer.

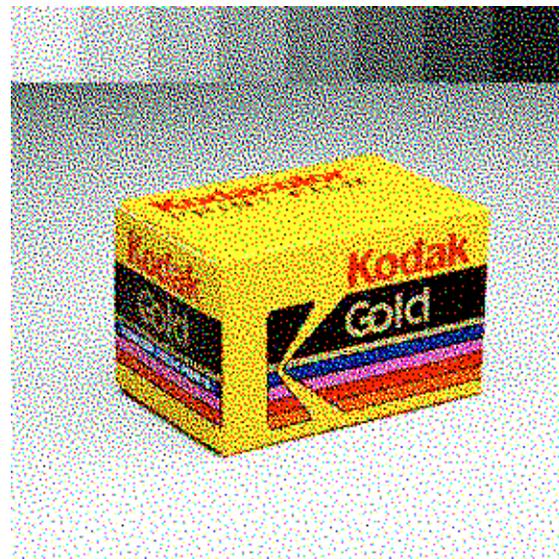
There is also an increased sensitivity to registration errors. This can cause the reproduced color to drift from page-to-page, or even within a page, as the registration characteristics change. This increased sensitivity to registration errors results in the fact that if one of the color planes is misregistered relative to the others, all of the dots shift together. This could cause a neutral patch to be formed by cyan dots next to a set of red dots, instead of a set of black dots on a white background. Since a cyan dot next to a red dot does not generally produce the same integrated color as a black dot, this causes the reproduced color to change.

4.2 Independently Derived Matrices. Because of the problems associated with dot-on-dot printing, it is usually desirable to decorrelate the halftone patterns used for each of the color channels.³⁰ In the graphic arts field, the preferred solution has typically been to decorrelate the halftone patterns by rotating the halftone patterns to different “screen angles.” For conventional graphic arts halftone methods, this rotation can either be accomplished optically or digitally. However, this solution is generally not practical for low-resolution printing applications where ordered-dither is useful.

A number of techniques can be used to decorrelate the halftone patterns used for ordered-dither. Parker and Mitsa have enumerated several of these methods³¹ One solution is to use independently derived dither matrices for each color channel. To effectively decorrelate the



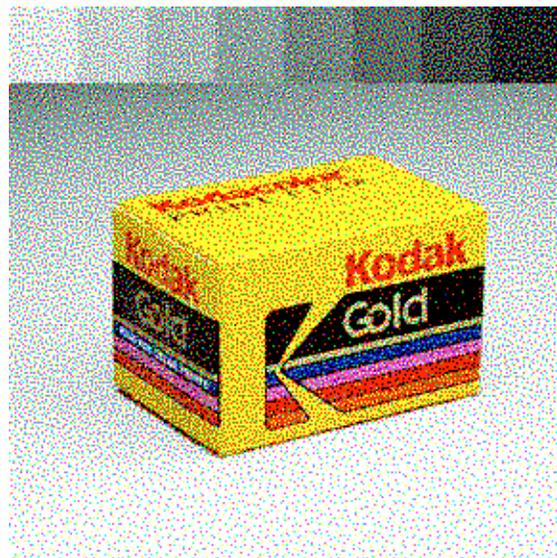
(a)



(b)



(c)



(d)

Figure 20. Images generated using 128×128 blue-noise dither matrices: (a) dot-on-dot matrices, (b) shifted matrices, (c) inverted magenta/shifted yellow matrices, and (d) jointly optimized matrices.

matrices for each channel, different random seeds could be used to initialize the blue-noise dither optimization process. Using uncorrelated matrices significantly reduces the amount of visible luminance modulation produced in the final image, however, it is not optimum because there is still a random overlap between the dots of the different color planes.

4.3 Shifted Matrices

Another approach that can be used to decorrelate the matrices is to spatially shift the phase of the dither matrix patterns for each channel relative to each other. For example, a blue-noise dither matrix, determined using one of the design techniques discussed above, can be used for the cyan color channel. This dither matrix can then be altered by shifting the phase of the dither pattern some number of pixels to the right for use with the magenta color channel. Similarly, the phase can be shifted by some number of pixels in the vertical direction for the yellow color channel. The phase shifts can be accomplished by computing a new set of dither matrices, or by simply adding an offset to the image row and column addresses before the modulo operations are applied to determine the dither matrix indices. The results should be essentially equivalent to those obtained with totally uncorrelated matrices.

A sample image generated using shifted dither matrices is shown in Fig. 20(b). The same 128×128 dither matrix that used to generate Fig. 20(a) was used here as well. However, the magenta dither pattern was shifted 64 pixels horizontally, and the yellow dither pattern was shifted 64 pixels vertically. It can be seen that the overall level of pattern visibility is substantially lower than for the dot-on-dot case. Any overall color difference between the two images is due to the fact that the color reproduction characteristics of a halftone device will be a function of the halftoning algorithm. In a real application, the output would be color corrected for the specific approach that is implemented. However, for this example, no custom correction was done.

The amount of decorrelation can be controlled somewhat by adjusting the amount of phase shift that is applied. The greatest decorrelation is typically obtained by shifting the phase by half the dither matrix size. In general, the argument could be made that the phase of the patterns should be shifted by half of the dominant halftone period to ensure that the dots have a higher probability of not overlapping. However, since the dominant period will be a function of gray level, this is generally not a practical solution.

4.4 Inverted Matrices

Another approach is to use an inverse dither matrix for one or more of the color channels. In this case, a new dither matrix is calculated by inverting the polarity of a first matrix. If the dither matrix has values in the range 0 to 255, the dither values for new matrix will simply be equal to 255 minus the dither values for the first matrix. Using this approach will minimize the number of dots that will overlap with each other for two of the color channels. In fact, for a 50% gray level, the two matrices will produce halftone patterns that are exact inverses of

each other. For lighter gray levels the dots for the first color channel will tend to be placed in the voids of the second color channel. The primary limitation of this approach is that most color imaging applications use more than two color channels. Therefore, in practice, some other method must be used to determine at least one of the color channels. For example, a first dither matrix could be used for the magenta channel, an inverted version of the magenta dither matrix could be used for the cyan channel, and a shifted version of the magenta dither matrix could be used for the yellow channel. In many cases, this may produce acceptable results because the yellow channel generally contributes less to the luminance modulation.

A sample image generated using this approach is shown in Fig. 20(c). Again, the same basic 128×128 dither matrix is used. However, in this case, the magenta matrix was inverted relative to the cyan matrix, and the yellow matrix was shifted 64 pixels vertically. Overall, this image shows a small improvement in the luminance modulation relative to the shifted matrices image. However, there are some mildly disturbing texture contours that can be seen just below the gray scale. These are formed near a code value of 128 where the cyan and magenta patterns are exact inverses of each other.

4.5 Jointly Optimized Matrices

Each of the approaches discussed in Sections 4.2 to 4.4 has the advantage that the halftone patterns for the different color channels are effectively decorrelated, therefore, the resulting halftone image should produce lower amounts of luminance modulation and sensitivity to registration errors relative to the nominal dot-on-dot printing case. However, none of these techniques produce images that will exhibit optimal behavior, particularly with respect to the visibility of the halftone patterns.

We have recently developed a method that can be used to simultaneously design a set of blue-noise dither matrices for each of the color channels for a color output device. The matrices are designed to provide an output image having halftone patterns with minimum visibility to a human observer. In the simplest version of this method, the halftone pattern visibility is minimized by determining the dither matrices that minimize the visible luminance modulation. This is frequently appropriate since the human visual system is more sensitive to luminance modulation than it is to chrominance modulation. However, the method can easily be extended to include terms in the cost function that reflect the visibility of the chrominance modulation as well as the luminance modulation.

The preceding monochrome dither matrix design methods can be adapted to determine the set of color dither matrices that have minimum visibility to a human observer. To use the visual cost function optimization approaches, it is first necessary to define a cost function related to the visibility of the halftone pattern. For the case where only the luminance component of the halftone pattern visibility is considered, the first step is to compute a spatial luminance distribution from the halftone patterns that are used for each color channel.

One method that can be used to determine the spa-

tial luminance distribution is by measuring the luminance values that result when each of the individual colorants (e.g., cyan, magenta, and yellow) are used, as well as the luminance values that result for the possible combinations of the colorants (e.g., red = magenta+yellow, green = cyan+yellow, blue = cyan+magenta, and black = cyan+magenta+yellow). To determine the luminance distribution, the halftone patterns for each of the color channels can be superimposed and the luminance value corresponding to the resulting combination of colorants can be assigned to each of the pixel locations.

Another approach that can be used to compute the spatial luminance distribution is to estimate its value by calculating a weighted sum of the individual halftone patterns

$$l(x, y) = \sum_{i=1}^{\text{channels}} w_i O_i(x, y), \quad (28)$$

where $O_i(x, y)$ is the output image bitmap for the i 'th color channel, and w_i is a weighting factor for each channel. The weighting factors should generally reflect the relative contributions of each channel to the luminance signal. Typically, yellow would have the smallest contribution, and magenta would have the largest contribution. An example of typical weights would be $w_c = 0.3$, $w_m = 0.6$, $w_y = 0.1$.

Once the luminance distribution $l(x, y)$ for a set of halftone patterns has been computed, the visibility of the luminance modulation can be estimated using a cost function analogous to that shown in Eq. (1):

$$\text{cost}_{\text{CMY}} = \iint |L_{\text{CMY}}(f_x, f_y) V(f_x, f_y)|^2 df_x df_y. \quad (29)$$

In this case, the Fourier transform the luminance distribution, $L_{\text{CMY}}(f_x, f_y)$, is used in the formula instead of the Fourier transform of the halftone pattern itself. The CMY subscript reflects the fact that the cost is associated with the halftone pattern for a particular color value. It is frequently useful to work with a discrete form of this cost function

$$\text{cost}_{\text{CMY}} = \sum_{i=0}^{M_x-1} \sum_{j=0}^{M_y-1} |L_{\text{CMY},ij} V_{ij}|^2, \quad (30)$$

where $L_{\text{CMY},ij}$ is the (i, j) 'th element of the discrete Fourier transform of the luminance distribution for a particular CMY color value, and V_{ij} is the human visual system sensitivity for the frequency corresponding to element (i, j) . As was discussed above, it may be desirable from a computational efficiency point of view to use the spatial domain versions of these cost functions. These forms follow directly using Parseval's theorem. Other types of cost functions can also be defined, which include terms relating to the visibility of the chrominance modulation in addition to terms relating to the visibility of the luminance modulation. These types of cost functions can be used to more accurately reflect the overall halftone pattern visibility. However, in many cases the improvement in the final results do not justify the added complexity in the optimization process.

As with the monochrome dither matrix design methods discussed above, there are a number of ways that

the visual cost associated with the halftone patterns can be minimized. For example, the bit patterns for all of the pixel values can be optimized simultaneously. In this case a total cost value is computed that is a combination of the individual cost values for a set of CMY color values. An optimization technique such as stochastic annealing can then be used to determine the bit patterns that produce the minimum total cost value.

One form of the total cost that can be used is a weighted sum of the individual cost values

$$\text{cost}_{\text{total}} = \sum_{\text{CMY}} w_{\text{CMY}} \text{cost}_{\text{CMY}}, \quad (31)$$

where w_{CMY} is a weighting factor, and the summation is computed over a certain set of CMY color values. The choice of the particular subset of CMY color values, as well as the weights assigned to each color value determine the relative importance of various parts of color space during the optimization process.

For example, if the neutral colors are believed to be the most important color values for a particular printing application the set of CMY color values having equal amounts of cyan, magenta, and yellow can be included in the summation. If all of the neutral colors are equally important, then the weighting factors can be defined to normalize the individual cost values so that they have similar magnitudes. Other color values that can be included in the set of CMY color values are primary color series where one color channel is varied throughout its range, and the other color channels are set to 0. CMY color values for particularly important colors, such as skin-tones, sky colors and grass colors can also be included. Yet another set of CMY color values that could be used would be all of the possible combinations of the allowable color values for the different color channels.

Alternatively, the halftone patterns for each input level can be determined sequentially rather than simultaneously. In this case, a halftone pattern for a first input level is initially determined, and then halftone patterns for the remaining input levels are determined one-by-one. One method that has been used successfully is to first determine the cyan halftone pattern for a certain input level, and then to determine the magenta and yellow halftone patterns for that input level in turn. When the halftone patterns for each of the color channels has been determined for that level, the next level can then be considered. Other optimization sequences are also possible. For example, the halftone patterns for each of the input levels for one color channel can be determined first. When the halftone patterns for the first color channel have all been determined, the remaining color channels can be optimized sequentially. As before, the individual levels may be optimized using a consecutive sequence, or using other types of sequences such as a binary-tree.

The step of determining halftone patterns for an initial input level can be performed in several ways. For example, a first set of halftone patterns can be randomly formed, and a stochastic annealing procedure can be used to minimize a visual cost function. Alternatively, the initial halftone patterns can be formed from monochrome

blue-noise dither matrices. For example, to initialize a set of dither matrices for a neutral input value, the method shown here in pseudo-code can be used:

```

for channel = 0 to num_channels
  for each i,j
    {
      if ((d(i,j) >= channel*level) and (d(i,j)
        <(channel + 1) * level)
        patternchannel(i,j) = 1
      else
        patternchannel(i,j) = 0
    },

```

(32)

where $num_channels$ is the number of color channels, $d(i,j)$ is the monochrome dither matrix, $pattern_{channel}(i,j)$ is the initial halftone pattern for a given color channel, and $level$ is the input level for the initial halftone pattern. It should be noted that the input level for the initial halftone pattern must be less than, or equal to, the number of levels in the monochrome dither matrix divided by the number of color channels. For a monochrome dither matrix having 256 different levels, and a system having three color channels this limits the input level to be less than or equal to 85.

If the initial halftone patterns are formed using this approach, the resulting patterns are non-overlapping, and reflect the basic frequency distribution characteristics of the monochrome dither matrix. Both of these characteristics are desirable, but the resulting patterns may not be optimum relative to the global minimum visibility solution. For this reason, it may be desirable to form patterns using this method, and then refine them using an optimization method such as stochastic annealing.

When a sequential optimization approach is used, the cost value can be computed from a single color value (typically a neutral color value corresponding to the input value currently being optimized), or can be a combination of cost values for a set of color values. Using a combination of cost values ensures that the visibility of the halftone patterns generated for color values in one part of color space is not optimized at the expense of other parts of color space. One combination of cost values that has been found to be useful is to compute a weighted average of the cost value for a neutral color value, and a corresponding pure color value. For example, if the bitmap for the cyan channel for input value 100 was being optimized, the combined cost value can be a weighted average of the cost for the neutral color where cyan, magenta, and yellow all have a value of 100, with the cost for the color where cyan has a value of 100, and magenta and yellow are zero.

When large dither matrices are being optimized, the visual cost based optimization techniques have been found to take a substantial amount of processing time to complete even with very powerful computers. For this reason, it may be desirable to use other optimization techniques, such as the visual potential technique described above, that converge more quickly, even if the final solution is slightly less optimal than could be found using the slower techniques.

The visual potential technique discussed in Section 2.2.3 can be extended to the current problem of

jointly optimizing blue-noise matrices for a color output device by simply replacing the halftone pattern with the luminance distribution for the set of halftone patterns. The perceived luminance distribution is calculated by convolving the luminance distribution with an approximation to the human visual system point-spread function. Dots can then be added/subtracted at the minima/maxima of the perceived luminance distribution. If more than one dot needs to be added or subtracted, the dots should generally be added/subtracted one at a time and the perceived luminance distribution recomputed after each step.

This approach can also be extended to include contributions from the perceived luminance distributions for other colors in addition to the neutral colors. One way to do this is to compute a combined perceived luminance distribution

$$l_p(x, y) = \sum_{CMY} w_{CMY} l_{p,CMY}(x, y), \quad (33)$$

where w_{CMY} is a weighting factor, $l_{p,CMY}(x,y)$ is the perceived luminance distribution for a certain CMY color value, and the summation is computed over a set of CMY color values. Once the combined perceived luminance distribution is computed, dots can be added/subtracted by finding the respective minimum or maximum. One set of CMY color values that has been found to be quite useful is a neutral value, combined with the corresponding pure color. For example, if the bitmap for the cyan channel for input value 100 were being optimized, the combined perceived luminance distribution can be calculated as a weighted average of the perceived luminance distribution for the neutral color where cyan, magenta, and yellow all have a value of 100, with the perceived luminance distribution for the pure color where cyan has a value of 100, and magenta and yellow are zero.

A sample image generated using jointly optimized 128×128 blue-noise matrices is shown in Fig. 20(d). A visual potential optimization approach was used starting from a set of initial patterns at level 85 formed from a monochromatic blue-noise dither matrix using Eq. (32). The dither matrices were formed by sequentially adding/subtracting pixels from the halftone patterns at the min/max of the combined perceived luminance distribution. In this case, the combined perceived luminance distribution was determined using a simple average of the luminance distribution for a neutral patch, and a pure color for the current color channel. It can be seen that the overall halftone pattern visibility is noticeably improved relative to the three other examples in Fig. 20.

An interesting way to compare the color halftone patterns is to look at the fraction of the image covered by the cyan, magenta, and yellow dots, as well as their two- and three-color combinations as a function of gray level for a neutral patch. Figure 21 shows a series of plots generated for the dither matrices used to produce the images in Fig. 20. Considering Fig. 21(a) it can be seen that only black and white dots are used to produce the neutral patches at every gray level. As the gray level is increased, the proportion of black dots increases, and the proportion of white dots decreases correspondingly.

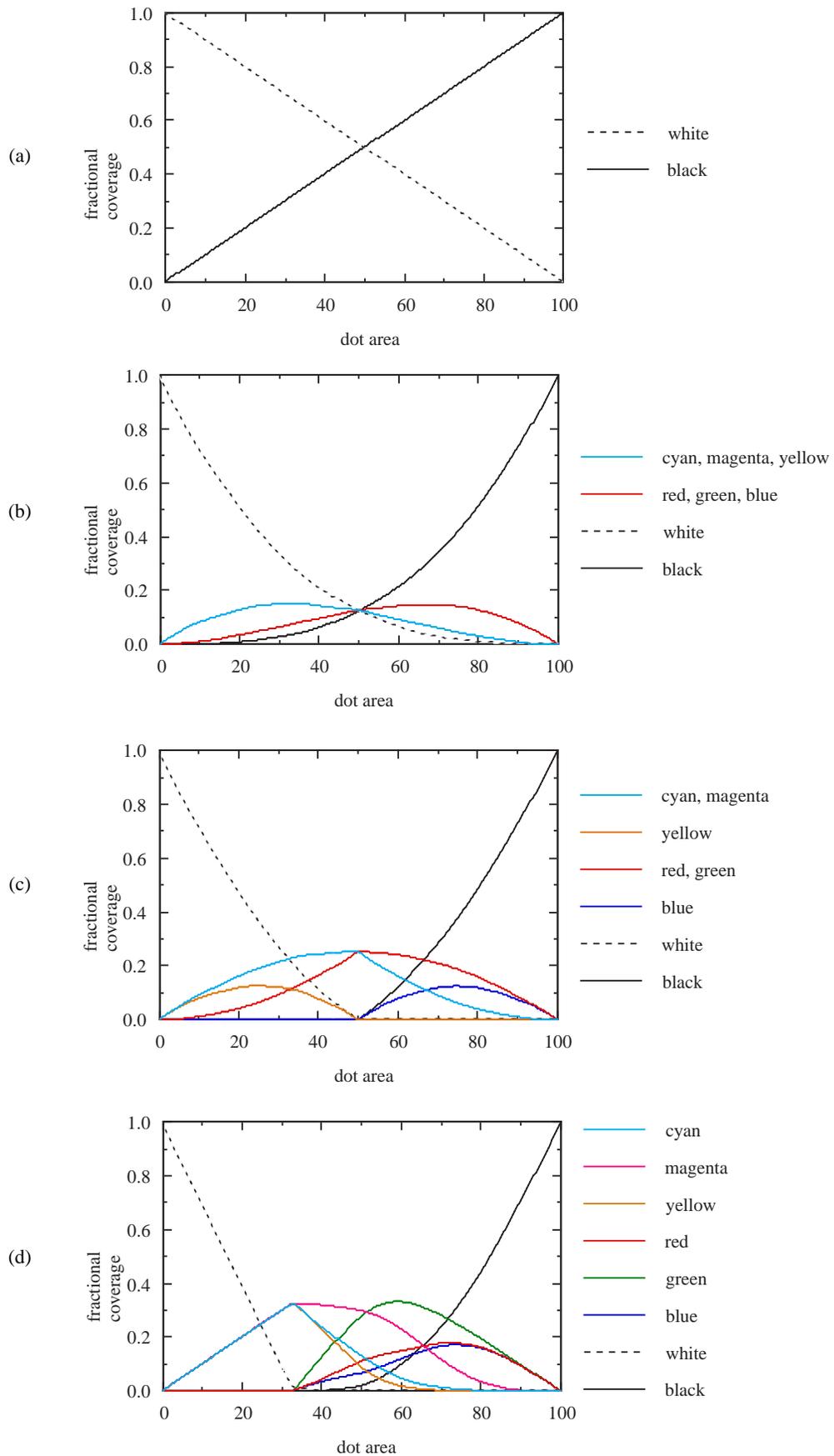


Figure 21. Fractional coverage of paper as a function of dot area for neutral patches: (a) dot-on-dot matrices, (b) shifted matrices, (c) inverted magenta./shifted yellow matrices, and (d) jointly optimized matrices.

This is characteristic of the dot-on-dot approach.

In Fig. 21(b) it can be seen that the characteristics of the shifted matrices approach are quite different. The fractional coverage of the cyan, magenta, and yellow dots are approximately equal at all gray levels, as are the coverages for the red, green, and blue two-color combinations. Since the shifted matrices produce halftone patterns that are effectively uncorrelated we would expect that the fractional coverages should approximately follow the Neugebauer equations,³²

$$\begin{aligned} f_w &= (1 - c)(1 - m)(1 - y), \\ f_c &= c(1 - m)(1 - y), \quad f_m = (1 - c)m(1 - y), \\ f_y &= (1 - c)(1 - m)y, \\ f_r &= (1 - c)my, \quad f_g = c(1 - m)y, \\ f_b &= cm(1 - y), \quad f_k = cmy, \end{aligned} \quad (34)$$

where c , m , and y are the fractional dot areas for the cyan, magenta, and yellow halftone patterns, and f_w , f_c , f_m , f_y , f_r , f_g , f_b , and f_k are the resulting fractions of the paper covered by white (no ink), cyan, magenta, yellow, red (magenta+yellow), green (cyan+yellow), blue (cyan+magenta), and black (cyan+magenta+yellow), respectively. In this case the values for c , m , and y are all equal since the neutral colors are being considered. It can be shown that the curves in Fig. 21(b) are indeed consistent with the expected values from Eq. (34).

The fractional coverage characteristics for the inverted matrices approach, shown in Fig. 21(c), are quite different from either of the previous graphs. Because the cyan and magenta matrices are inverses of each other, the blue and black coverages remain at zero until the 50% dot area is reached. This is because cyan and magenta dots never occur at the same pixel location until all of the pixel locations are filled with one or the other. However, since the yellow halftone pattern is effectively uncorrelated with the cyan and magenta patterns, two-color combinations that include yellow (red and green) start to form even for the smallest dot areas.

Figure 21(d) shows the fractional coverage characteristics for the jointly optimized matrices. It can be seen that up to a 33% dot area there are only single color ink colors present. This is due to the fact that the initial pattern was formed using non-overlapping bitmaps at this dot area. Above the 33% dot area, the two-color combinations begin to form. Note that green dots are formed most rapidly due to the fact that green contributes least strongly to the luminance modulation since it is the lightest of the two-color combinations. Black dots are the last to start forming since they have the largest effect on the luminance modulation.

5 Comparison of Blue-Noise Dither and Error Diffusion

Blue-noise dither halftoning methods have been found to produce images with pleasing visual characteristics. The resulting image structure resembles the results that can be obtained using error-diffusion algorithms. However, the computational complexity associated with the ordered-dither approach is much simpler than that of error diffusion. Because of the fact that the blue-noise dither patterns must be correlated as a function of gray

level, error diffusion fundamentally maintains a small image quality advantage.

To illustrate this, consider a halftone pattern generated for a gray level of 240. To obtain this tone value, a dot needs to be formed at one out of every 16 pixels. Error diffusion algorithms accomplish this by producing a halftone pattern having a fundamental period of about 4 pixels in both the vertical and horizontal directions. If a blue-noise dither pattern were designed specifically for this gray level it would have the same basic characteristics. Now consider a gray level of 239. To produce this tone value it is necessary to add one more dot for every 256 pixels. Error diffusion naturally adjusts the fundamental period of the halftone to an average of about 3.9 pixels to accomplish this. However, to design a blue-noise pattern for a gray level of 239 that is correlated with the pattern for a gray level of 240, the other dots must be left in the same locations and the new dots must be inserted into the pattern. Therefore, instead of adjusting the dot spacing to be an average of 3.9 pixels apart, some dots are still 4 pixels apart, and some dots are closer where an extra dot has been inserted in the middle of a void in the first pattern. This necessarily adds some spectral power in the low spatial frequency regions. Therefore, the resulting halftone pattern are less optimal than the error diffusion patterns. Consequently, this implies that blue-noise dither will never be able to produce optimal halftone patterns over the entire tone scale to the extent that error diffusion can. However, for many applications, the quality level that can be obtained with blue-noise dither is adequate and the simplicity and speed of the algorithm out-weigh any quality loss.

6 Conclusions

This paper has reviewed and compared the various techniques that have been used to develop blue-noise dither matrices. In particular, a series visual cost function based techniques were presented that have been developed by Sullivan *et al.* Additionally another family of techniques was discussed that involves designing the dither matrices by analyzing the spatial dot distribution. This group included the iterative constraint method developed by Mitsa and Parker, and Ulichney's "void-and-cluster" technique, as well as a number of other variations.

The visual cost function based techniques have the closest tie to the response of the human visual system. Therefore, they should offer the greatest opportunity to design halftone patterns that have the minimum visibility to the human observer. The current approaches utilize a simple linear system model of the human visual system, although other more sophisticated models could be easily investigated. The largest drawback of this approach is the amount of computation time needed to design a dither matrix of reasonable size. Additionally, the stochastic optimization techniques that have been investigated have been found to rarely converge to a global minimum. This is largely due to the high dimensionality of the problem.

The techniques, such as the iterative constraint methods and the void-and-cluster algorithm, that attempt to

optimize the blue-noise patterns by simple analysis of spatial dot distribution have the advantage that they are relatively fast to compute compared to the other methods. Because of their speed, they can frequently be used to design larger matrices than can practically be done otherwise. Although these methods generally do not produce patterns that are truly optimal relative to a visual cost function, it has been found that the quality increase associated with the larger matrix size usually makes up for any deficiencies. As a result, these methods have been found to produce good quality results. Although the methods that use a filter size that is adaptive with gray level may offer some advantages, very acceptable dither matrices have been designed with even the simpler algorithms.

All of these methods can produce dither matrices that exhibit similar visual characteristics, although there are subtle differences between the resulting patterns. In all cases, the results are quite sensitive to the form of the filter function that is used. Even the approaches that are tightly coupled to a model of the human visual system have parameters, such as viewing distance, that can have a large effect on the quality of the final results. In fact, the differences between the results obtained using different forms of the filter function/visual cost function have generally been found to be larger than the differences between the different optimization techniques.

Ways to extend the basic blue-noise dither techniques to multilevel and color output devices were also described. At 300 dpi, the multilevel blue-noise dither approach has been found to produce photographic quality images using only 16 output levels. Recent advances were described for the design of jointly optimized dither matrices for color output devices. It was demonstrated that these results provide an incremental improvement in the quality over the conventional solutions for color output devices. However, to take full advantage of these improvements requires that the registration errors associated with the output device be relatively small. It is expected that if the registration errors are very significant, the optimum result will move toward uncorrelated matrices.

Acknowledgement

The authors would like to acknowledge the contributions of Jim Sullivan and Larry Ray who played a large part in the development of many of the blue-noise dither optimization techniques discussed in this paper, and have provided many valuable insights and suggestions for various aspects of this work.

References

1. J. O. Limb, "Design of dither waveforms for quantized visual signals," *Bell Syst. Tech. J.* **48**, 2555–2583 (1969).
2. R. J. Klensch, D. Meyerhofer, and J. J. Walsh, "Electronically generated halftone pictures," *RCA Rev.* **31**, 517–533 (1970).
3. E. M. Granger, J. F. Hamilton and L. G. Wash, "Electronic graphic arts screener," U.S. Patent No. 4,918,622 assigned to Eastman Kodak Company (1990).
4. A. J. Leone and K. E. Spaulding, "Method for generating halftone image data with randomly selected threshold array," U.S. Patent No. 5,150,428 assigned to Eastman Kodak Company (1992).
5. B. E. Bayer, "An optimum method for two-level rendition of continuous-tone pictures," *Proc. IEEE Int. Conf. Commun.*, (26-11)–(26-15) (1973)
6. J. R. Sullivan and R. L. Miller, "Digital halftoning with minimum visual modulation patterns," U.S. Patent No. 4,920,501 assigned to Eastman Kodak Company (1990).
7. J. Sullivan, L. Ray, and R. Miller, "Design of minimum modulation halftone patterns," *IEEE Trans. Syst. Man Cybern.* **SMC-21**(1), 33–38 (1991).
8. T. Mitsa and K. J. Parker, "Digital halftoning technique using a blue-noise mask," *J. Opt. Soc. A* **9**(11), 1920–1929 (1992).
9. R. Ulichney, "The void-and-cluster method for dither array generation," in *Hum. Vision, Visual Process., Digital Disp. IV, Proc. SPIE* **1913**, 332–343 (1993).
10. R. Ulichney, *Digital Halftoning*, MIT Press, Cambridge, MA (1987).
11. M. P. Vecchi and S. Kirkpatrick, "Global wiring by simulated annealing," *IEEE Trans. Comp. Aid. Des.* **CAD-2**(4), 215–222 (1983).
12. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing* Cambridge Univ. Press, Cambridge, pp. 343–352 (1988).
13. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc. (1989).
14. G. J. E. Rawlings, *Foundations of Genetic Algorithms*, Morgan Kaufmann Publishers (1991).
15. J. Dalton, "Perception of binary texture and the generation of stochastic halftone screens," in *Hum. Vision, Visual Process., Digital Disp. VI, Proc. SPIE* **2411**, 207–220 (1995).
16. J. R. Sullivan and L. A. Ray, "Digital halftoning with correlated minimum visual modulation patterns," U.S. Patent No. 5,214,517 assigned to Eastman Kodak Company (1993).
17. L. A. Ray, "Method and apparatus for generating simultaneously derived correlated digital halftone patterns," U.S. Patent Application No. 07/848,779 (filed March 10, 1992).
18. K. J. Parker and T. Mitsa, "Method and apparatus for halftone rendering of a gray scale image using a blue noise mask," U.S. Patent 5,111,310 assigned to Research Technologies Corporation (1992).
19. M. Yao and K. J. Parker, "Modified approach to the construction of a blue noise mask," *J. Electron. Imaging* **3**(1), 92–97 (1994).
20. R. J. Rolleston and S. J. Cohen, "Halftoning with random correlated noise," *J. Electron. Imaging* **1**, 209–217 (1992).
21. Q. Yu, K. J. Parker and M. Yao, "Optimality of blue noise mask binary patterns," in *Proceedings of IS&T's NIP12: International Congress on Digital Printing Technologies*, 66–69 (1996); see pg. 151, this publication.
22. Q. Lin, "Halftone images using special filters" U.S. Patent No. 5,317,418 assigned to Hewlett-Packard Company (1994).
23. Q. Lin, "Halftone image formation using dither matrix gen-

- erated based upon printed symbol models," U.S. Patent No. 5,469,515 assigned to Hewlett-Packard Company (1995).
24. D. C. Barton, "Digital halftone rendering of a gray scale image with frequency-dependent diagonal correlation," European patent application No. 0 647 058 A2 assigned to Hewlett-Packard Company (filed April 5, 1995).
 25. R. S. Gentile, E. Walowit, and J. P. Allebach, "Quantization and multilevel halftoning of color images for near original image quality," *J. Opt. Soc. Am. A* **7**(6), 1019-1026 (1990).
 26. P. A. Delabastita, "Multilevel halftoning," in *Fourth technical symposium on prepress, proofing and printing*, 68-73 (1995).
 27. K. E. Spaulding and L. A. Ray, "Method and apparatus for generating a halftone pattern for a multilevel output device," U.S. Patent application No. 08/131,801 assigned to Eastman Kodak Company (filed October 4, 1993).
 28. R. Miller and C. Smith, "Mean preserving multilevel halftoning algorithm," in *Hum. Vision, Visual Process., Digital Disp. IV, Proc. SPIE* **1913**, 367-377 (1993).
 29. R. L. Miller, "Apparatus and method for generating multilevel output values for pixels in a halftone cell," U.S. Patent No. 5,291,311 assigned to Eastman Kodak Company (1994).
 30. J. B. Mulligan, "Digital halftoning methods for selectively partitioning error into achromatic and chromatic channels," in *Hum. Vision Electron. Imaging: Models, Methods, Appl., Proc. SPIE* **1249**, 261-270 (1990).
 31. K. J. Parker and T. Mitsa, "Method and apparatus for halftone rendering of a gray scale image using a blue noise image," U.S. Patent No. 5,341,228 assigned to Research Technologies Corporation (1994).
 32. A. C. Hardy and F. L. Wurzburg, Jr., "Color correction in color printing," *J. Opt. Soc. Am.* **38**(4), 300-307 (1948).
- * Previously published in *JEI* **6**(2), pp. 208-229, 1997.
-
-