Stochastic Screen Design using Symmetric Error Compensation

Qing Yu and Kevin J. Parker
Department of Electrical Engineering
University of Rochester, Rochester, NY 14627

Abstract

Stochastic screen halftoning combines the speed of ordered dither and high quality of error diffusion. The general approach to stochastic screen design employs either spatial domain or frequency domain filtering to generate binary patterns with "blue noise" characteristic at each gray level, which can be summed to form a single threshold array. In this paper, we will propose a novel technique to design stochastic screens using an error diffusion approach called symmetric error compensation (SEC) [1], where all neighbors of the current pixel are involved in the computation. The screen is still designed at each gray level, but each level is constructed using SEC. Typically, SEC is used to identify those pixels to be removed or added for building neighbor binary patterns. The processing sequence for SEC is determined by a distance variable for each unprocessed pixel. The screen from this method will be compared with the Blue Noise Mask[2], which is the prototypical stochastic screen generated from filtering technique. We will present the gray ramp and dot patterns at certain gray level with radial power spectrum for both screens.

Keywords: halftone, stochastic screen, ordered dither, blue noise mask, symmetric error compensation, binary pattern, radial power spectrum.

1 Introduction

Stochastic screen halftoning is the subject of active research in recent years. It combines the simplicity of ordered dither with the blue noise characteristic of error diffusion. Stochastic screen halftoning is a point comparison process, so it is easily implemented. Thus, devices currently using ordered dither techniques may be switched to stochastic screen halftoning simply by replacing the original dither array with a stochastic screen. The halftone image from a stochastic screen will have the visually pleasing blue noise characteristic, which is guaranteed when screens are generated from blue noise dot patterns of individual gray levels.

To construct a stochastic screen, we usually start from an initial binary pattern for some intermediate level g ($0 < g < 255$, assuming an 8 bit mask). Once this pattern is optimized, level g - 1 is processed. For this level, the binary pattern is created by converting the appropriate number (the total number of pixels in the binary pattern divide by the total number of levels) of 1's to 0's in the previous pattern g. This process is repeated until binary patterns are generated for all the levels below g to level 0. Analogous procedures are used to construct binary patterns for all the levels above g to level 255. Finally, these binary patterns are summed to form a single threshold array, a stochastic screen. As we can see, the quality of a stochastic screen is directly related to the quality of individual binary patterns at each gray level.

The general approach to binary dot pattern design employs either spatial domain or frequency domain filtering to generate binary patterns with blue noise characteristics. These approaches [2, 3, 4, 5] have been reviewed in a recent paper [6]. We have shown that the direct filtering of binary patterns can be useful in selecting pixels which can be changed to produce a desired result in the spatial domain, and correspondingly approximate a desired power spectrum in the frequency domain.

Since the major advantage of stochastic screen hafltoning is its ability to generate halftone patterns similar to those from error diffusion, it will be very natural to introduce error diffusion process into the design of binary dot patterns. In this paper, we will present a technique which employes an error diffusion scheme, symmetric error compensation (SEC) [1], to design dot patterns for stochastic screens.

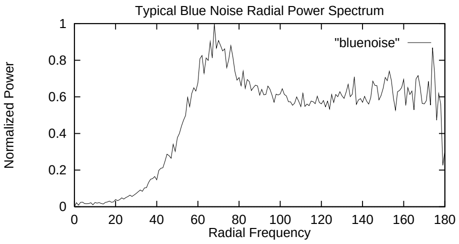


Figure 1: *Left: A blue noise radial average power spectrum*

2 Symmetric Error Compensation

Error diffusion is an adaptive algorithm that produces patterns with different spatial frequency content depending on the input image value. It forces total tone content to remain the same and attempts to localize the distribution of tone levels. The halftone image from error diffusion generally contains sharp edges and many image details [7]. Most part of the success of error diffusion lies in the fact that it is a "good blue noise generator"[8]. In the academic literature, the nature of noise is often described by a color name, i.e., white noise, so named because its flat shaped power spectrum. Blue noise, on the other hand, has most its energy located at a high frequency band with very little low frequency component. A typical blue noise radial average power spectrum (RAPS) is shown in Figure 1. Patterns with blue noise characteristics generally enjoy the benefits of aperiodic uncorrelated dot patterns without low frequency graininess.

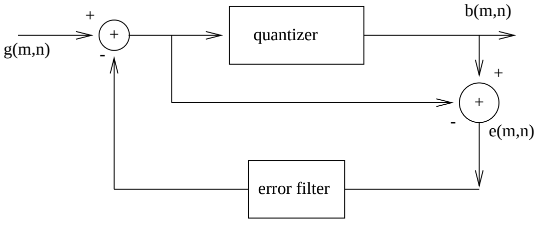
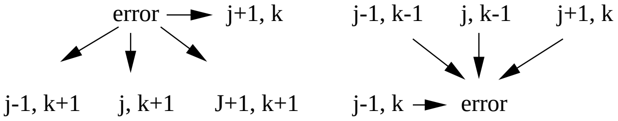


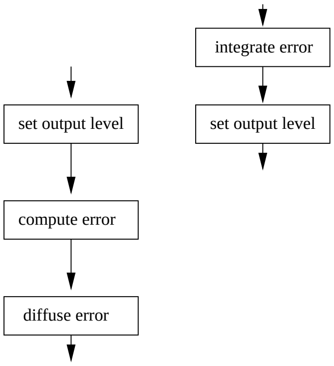
Figure 2: *Standard error diffusion*

Figure 2 shows the flow chart for standard error diffusion, which was first presented by Floyd and Steinberg [9]. An alternative scheme, called back error compensation, is later proposed by Marcu and Abe [10]. The difference bewteen these two schemes is that, in standard error diffusion, the error due to selection of the output pixel value 1 or 0 is diffused forward to the unprocessed pixels, which is illustrated in Figure 3.a and in the diagram in Figure 4.a; in back error compensation, output is selected that minimizes the error between input and the equivalent gray level of binary image, which is computed by weighting the error produced in the previous processed pixels as illustrated in Figure 3.b and in the diagram in Figure 4.b.



(a) standard error diffusion (b) back error compensation

Figure 3: *error manipulation*



a. diffuse error forward b. back error compensation

Figure 4: *processing diagram*

Later on, Marcu and Abe [1] demonstrated that these two approaches are mathmatically equivalent, and furthered unified these two approaches and proposed the symmetric error compensation (SEC) method. A symmetric error kernel is considered with the center in the current pixel to be processed. All pixels from the kernel are considered in computation. The kernel has associated a symmetric weighting mask of a 2D low pass filter. To process each input pixel, the symmetric error compensation passes twice over all the pixels in the kernel. In the first pass, all the pixels which have already been processed are considered and an "equivalent gray error" term is computed. This term is used to compute the error between the gray level input pixel and the equivalent gray level of the output image. The output value that minimize the error is selected as the output pixel value. In the second pass, the unprocessed pixels absorb the error between the modified input pixel and the output pixel. Figure 5 illustrates the steps in SEC.

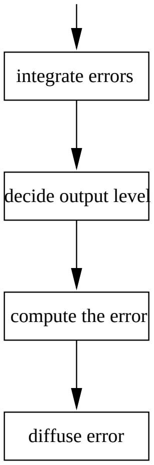


Figure 5: *follow chart of SEC*

With SEC, we are able to process the image pixels in an arbitary sequence. The error diffusion can start anywhere in the image and the processing sequence can be defined in advance, since any pixel in the kernel is automatically considered either to contribute an error term to the modified input pixel or to absorb the error diffused from the output pixel.

3 Dot pattern design with SEC

We can apply SEC directly to dot patterns design for stochastic screen. As we pointed out previously, in order to construct a single value screen, every dot pattern (except the starting pattern) is built from one of its neighboring dot patterns. Assume we have an optimized starting pattern P_g at level g, which could be got from filtering process or from error diffusion of a uniform gray patch of level g. To desgin the dot pattern P_{g-1} for level g-1, we need to keep all the "0" pixels of P_g untouched and switched certain number of "1"s to "0". Therefore, in a sense, some pixels of dot pattern P_{g-1} (those "0" pixels in pattern P_g) have already been processed, we need to diffuse error at those locations to their neighborhood and identify those "1"s to be switched to "0"s. As we can see, SEC could be well fit for this task. What we need to do is to define a processing sequence and to design a filter kernel.

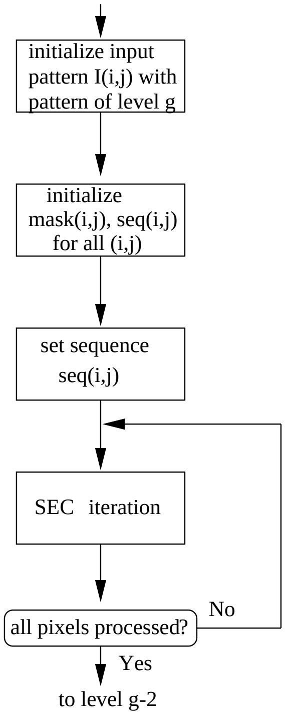


Figure 6: *follow chart of pattern design*

3.1 processing sequence

Since those "0" in dot pattern of level of g are inherited by level g-1, we should define a processing sequence such that errors at these locations are gradually propagated to their neighborhood. This will minimize the possibility of setting any pixels to "0" around these locations so as to eliminate clumping. One way to realize this is to relate the processing sequence of any pixel directly to its distance to those "inherited" "0"s from dot pattern of level of g. The follow chart of this algorithm is shown in Figure 6, assuming dot pattern P_g of level g has already been constructed. The detailed steps are given below.

[1] First, initialize input pattern I_{g-1} with P_g in the following way:

$$I_{g-1}(i,j) = \begin{cases} 0, & for \quad P_g(i,j) = 0 \\ g', & for \quad P_g(i,j) \neq 0 \end{cases} \qquad (1)$$

g' is set to a value such that the average value of input pattern I_{g-1} is g-1.

$$g' = \frac{g-1}{g} * 256 \qquad (2)$$

[2] Initialize mask(i.j) and seq(i,j) to 0 for each pixel in the dot pattern, mask(i,j) will monitor the current state of each pixel (processed or not), seq(i,j) will be used to store the processing sequence for each pixel.

[3] For each pixel I(i,j), if I(i,j) = 0, set mask(i,j) as 1; otherwise, find its nearest neighboring pixel of value "0", compute the distance d(i,j) between these two pixels, and map d to some integer value to set seq(i,j) (1, 2, 3 ...). This mapping should maintain the order such that smaller d always be map to smaller integer value.

[4] Starting the SEC process with the defined sequence from those pixels with seq(i,j) = 1. Set mask(i,j) to 1 when a pixel is processed. Continue the iterations till all the pixels are processed or mask(i,j) = 1 for all pixels.

[5] Continue to design dot pattern for level g-2 and so on.

3.2 filter kernel design

As we mentioned before, the filter kernel generally takes on a shape of a 2D low pass filter with symmetric weighting. The simplest example is shown in below. Intuitively, two parameters could be adjusted for this kernel. One is the weighting factors, another is the size of the kernel (3 X 3 or 5 X 5). Investigation is currently undergoing to study the effects of these two parameters.

1/16	3/16	1/16
3/16	1	3/16
1/16	3/16	1/16

3.3 thresholding value setting

In the SEC algorithm, each modified input pixel is compared with a thresholding value T, and based on the outcome, the output pixel is either set to 1 or 0. T should be made adaptive so that the output dot pattern is mean preserved.

4 Experimental Result

Using the 3x3 gaussian kernel as shown above, and starting from the same optimized dot pattern at level 250, we designed one screen with the SEC scheme and another one with the direct filtering technique. A ramp for each screen is shown in Figure 8. (only enlarged partial is shown for better illustration).

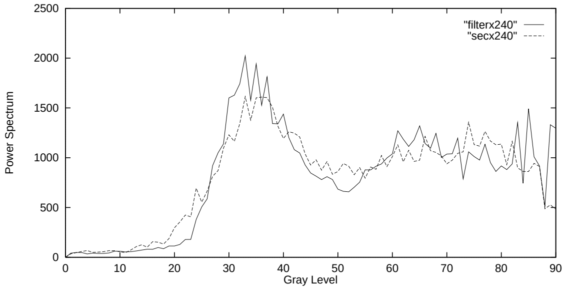


Figure 7: *radially averaged power spectrum*

Figure 9 shows the dot patterns of level 240 for each screen. Figure 7 shows the radially averaged power spectra for these two patterns.

As we can see, the quality of the dot pattern from the SEC algorithm is comparable to that of direct filtering technique and clumping is minimized with the SEC algorithm.

5 Discussion

We realize that this SEC algorithm is very effective to generate good dot patterns at highlight gray levels (for example, above 192 and below 64). However, it doesn't work very well around mid-gray levels, mainly due to the fact that most of the unprocessed pixels have the same sequence number. To get over this problem, we could work on another efficient way to define the processing sequence or switch to filtering process for these levels.

There are rich literature on error diffusion techniques, which offer a lot of room for us to improve the SEC scheme. One possible improvement could employ a visual model to modify the error propagation process, as proposed by Sullivan et al.[11]

6 Conclusion

In this paper, we present a stochastic screen design technique that employs a specific error diffusion sheme, SEC to design dot patterns for each gray level. We have shown that this new algorithm will enable us to generate dot patterns of blue noise characteristics for the design of stochastic screens.

References

[1] G. Marcu and S. Abe, "Symmetric error compensation for digital halftoning and applications," *IS&T/SPIE Symposium on Electronic Imaging Science & Tech.*, 1997.

[2] T. Mitsa and K. J. Parker, "Digital halftoning using a blue noise mask," *Image Processing Algorithms and Techniques III, SPIE 1452*, pp. 47–56, 1991.

[3] T. Mitsa and K. J. Parker, "Digital halftoning using a blue-noise mask," *Journal of the Optical Society of America A*, vol. 9, pp. 1920–1929, Nov. 1992.

[4] R. Ulichney, "The void-and-cluster method for dither array generation," in Allebach and Rogowitz [12], pp. 332–343.

[5] M. Yao and K. J. Parker, "Modified approach to the construction of a blue noise mask," *J. Elec. Imag.*, vol. 3, no. 1, pp. 92–97, 1994.

[6] Q. Yu, K. J. Parker, and M. Yao, "On filter techniques for generating blue noise mask," in *Proceedings, IS&T's 50th Annual Conference*, (Boston, MA), IS&T, May 1997.

[7] K. T. Knox, "Error diffusion: A theoretical view," in Allebach and Rogowitz [12], pp. 326–331.

[8] R. Ulichney, *Digital Halftoning.* MIT Press, 1987.

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

[10] G. Marcu and S. Abe, "Halftoning by back error compensation," in *Proceedings, NIP12: International Conference on Digital Printing Technologies*, (San Antonio, TX), pp. 132–135, IS&T, Oct. 1996.

[11] J. Sullivan, R. Miller, and G. Pios, "Imaging halftoning using a visual model in error diffusion," *Journal of Optical Society America*, no. 10, pp. 1714–1724, 1993.

[12] J. P. Allebach and B. E. Rogowitz, eds., *Proceedings, SPIE—The International Society for Optical Engineering: Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, (San Jose, California), SPIE, Feb. 1993.

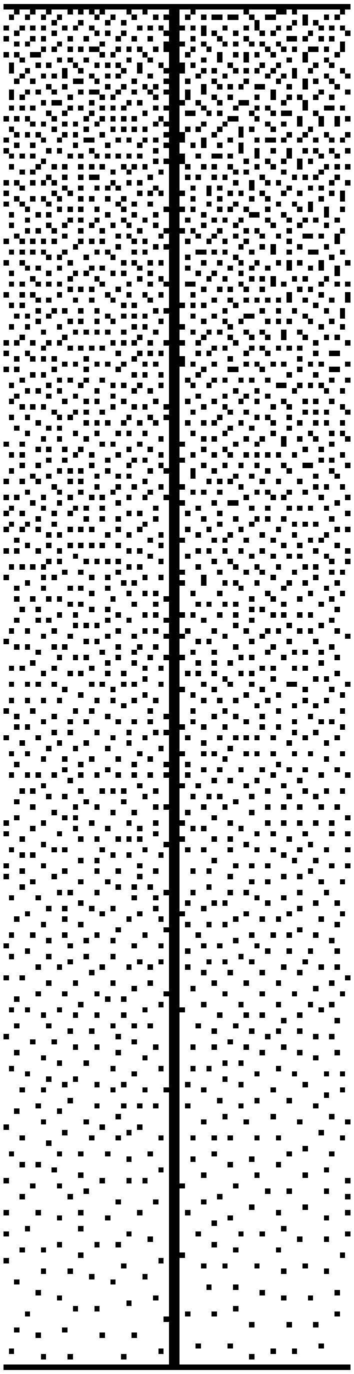
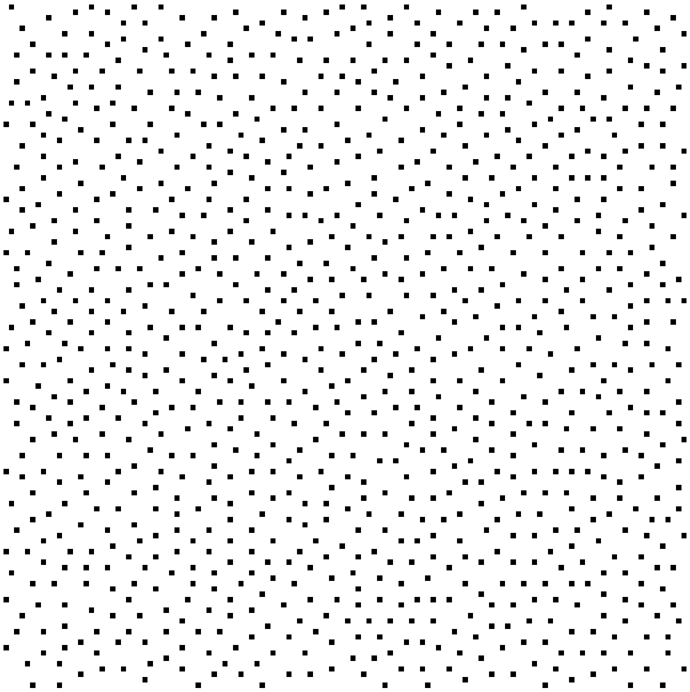


Figure 8: *left: screen from SEC; right: screen from filtering*



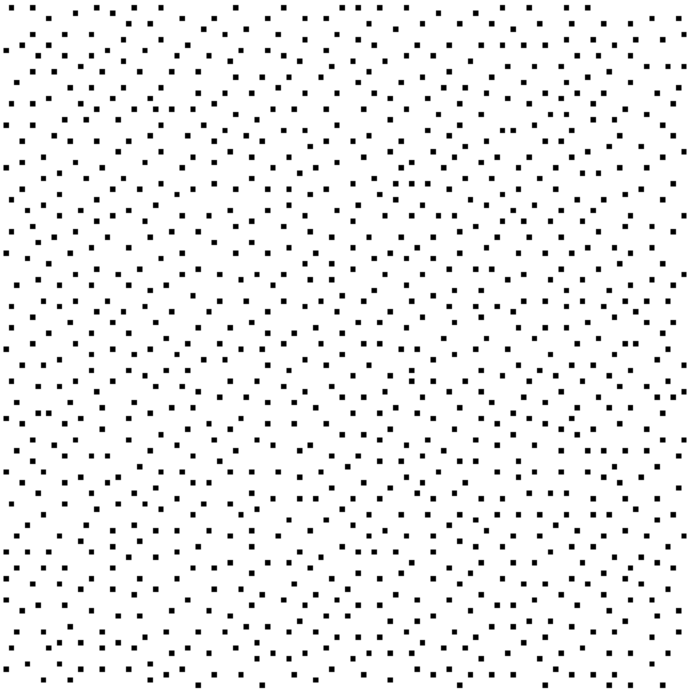


Figure 9: *Up: filtering; down: SEC*