Spatial Extent of Void and Cluster Finding Filters

Robert Ulichney
Cambridge Research Lab, Digital Equipment Corp.
Cambridge, Massachusetts

Abstract

The Void-and-Cluster algorithm for designing dither arrays is both simple and general. Considerable attention has been placed on the selection of various linear and nonlinear cluster-finding filters. In this paper, the importance of separating the treatment of the shape and spatial extent of cluster-finding filters is illustrated. Lack of such separation may be a source of confusion in other filter design efforts. The effect of finite precision on the implementation of these filters is analyzed. For a Gaussian filter, the limits imposed by the floating-point exponent and mantissa are shown to define the limits of spatial extent and discrimination.

Introduction

High quality dithered images can result from processes that employ neighborhood operations, such as error diffusion, or from multipass processes that use iterative optimization algorithms. But for high speed and simplicity of implementation, ordered-dither methods are in a distant first place. This is particularly true for video applications where a high premium are placed on these requirements. Ordered dither or screening techniques are point operations that simply compare the current pixel with a periodic and deterministic threshold array.

The first step in creating an ordered dither threshold array is the design of a template which ranks the order of the thresholds. Subsequent steps normalize this template by including tone correction and range adjustment to account for the number of input and output levels. The design of the template is central to the nature of the perceived textures in the output.

The method of recursive tessellation[1] generates dispersed-dot ordered dither templates of which the familiar patterns described by Beyer[2] are a subset. The method builds the template by recursively filling the center of the largest voids in the intermediate patterns. This method was generalized by the void-and-cluster method[3]. It is generally agreed that blue noise[4] properties are the most pleasant for display devices where disperse-dot dither patterns can be accommodated. While the void-and-cluster method is extremely simple, it is versatile enough to generate arrays with blue noise properties as well as recursive-tessellation arrays.

Recent work[5,6] include the incorporation of the visual system and printer models in the design of blue noise ordered dither arrays. A very different method for generating blue noise ordered dither arrays based on manipulation of power spectra is described by Mitsa and Parker[7].

Filter Width vs. Extent

Central to the void-and-cluster method is the mechanism, or filter, for finding the center of voids or groups of majority pixels, and finding the center of clusters or groups of minority pixels. Such filters should have at least the following three properties:

1. Isotropic about the center point,
2. Higher weight for locations closer to the center point, and
3. Unbounded in extent.

While a number of mechanisms can satisfy these properties, a symmetric Gaussian is perhaps the simplest to describe:

$$G(r) = e^{\frac{r^2}{2\sigma^2}} \text{ where } r^2 = x^2 + y^2.$$

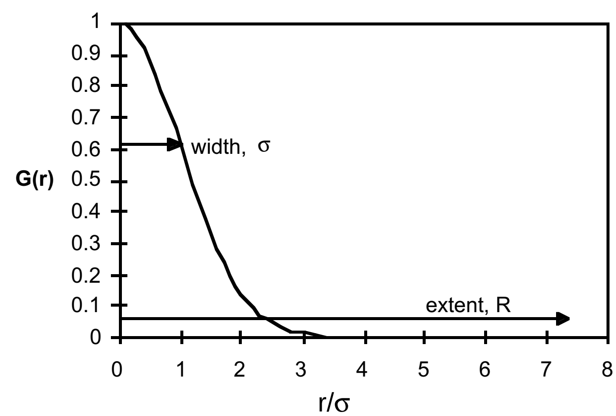


Figure 1. Width vs. extent of a radial slice of a 2D Gaussian function

This function is shown in Figure 1 where the two characteristics of width, σ, and extent (or region of support), R, are illustrated.

It was first reported[3] that a good example of a void-and-cluster finding filter was a simple Gaussian with a fixed width of 1.5, in units of pixel periods. More details of the associated experiments with the filter width were later presented[8].

In typical signal processing applications, the contribution of such a filter looses its significance beyond an extent of 3σ or 4σ. The key point of this paper is that this is not the case when used as a void or cluster finding filter! As can be seen in Figure 2, the ability of a Gaussian to discriminate between adjacent concentric “shells” of more distant pixels continues unbounded, and this should not be prematurely constrained.

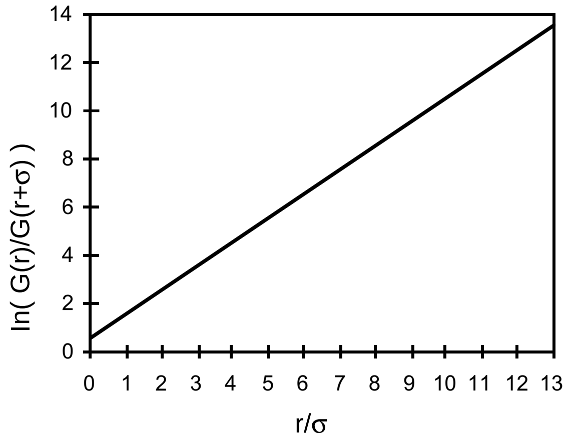


Figure 2. Ratio between neighboring concentric shells of a Gaussian increases with distance

The need to make the filter width adapt to the principal wavelength (average distance between minority pixels) in subsequent studies[9,10], may be due to the limiting of the filter extent. If untruncated, the use of a Gaussian in the array generation process results in excellent patterns at all gray levels.

Of course the use of finite computational tools will set some limit on the extent, but this limit turns out to be quite generous. There are two numbers that restrict the effective extent in the computation of Gaussian filtering, the size of the exponent and mantissa of the floating point representation, which will now be examined separately.

Exponent Limit

Figure 3 illustrates the components of a floating point number. The exponent is allocated E bits, and the mantissa M bits. Typical values for single precision are E=7 and M=23; For double precision, E=10 and M=52.

Sign	**Sign**	**Exponent**	**Mantissa**
1 bit	1 bit	E bits	M bits

Figure 3. Typical bit allocation in a floating point number

The value of E limits the extent R of G(r) in cases of long principle wavelength. The smallest number that can be digitally represented is

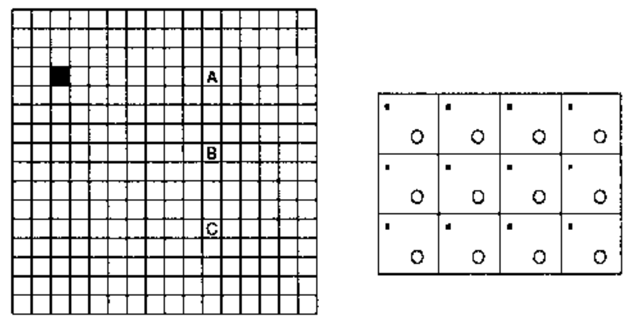
$$G(R) = e^{-\frac{R^2}{2\sigma}} \approx 2^{-2^E},$$

or

$$R \approx \sigma\sqrt{2^{E+1}\ln 2}\ .$$

So for single precision, R ≈ 13.3σ, and for double precision, R ≈ 37.7σ.

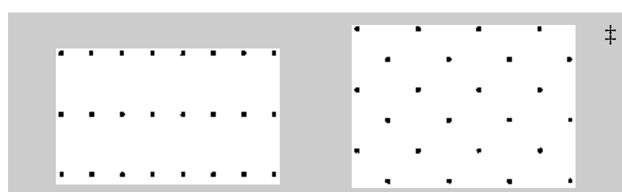
Figure 4 illustrates the problem of finding the center of the largest void in a long wavelength situation. In this simple example, a single minority pixel is present in a 16 by 16 array. In Figure 4b, the entire 16 by 16 period is tiled a few times to help visualize the center of the void; the void location is circled. The void-finding filter will identify incorrect void centers if the extent is limited, regardless of the value of σ. In Figure 4a, the correct pixel C will be located for all extents greater than $\sqrt{128}$. Examples of incorrect identifications are pixel A when R=6, and pixel B when R=10.



(a) (b)

Figure 4. Locating the center of the largest void in a long wavelength pattern. (a) The resulting locations, A, B, and C, using filters with different extents, R=6, R=10, and R>11.3, respectively. (b) Tiling the pattern in (a) with the true void center circled.

Figure 5a shows a g=1/256† pattern at 50 dpi resulting from the use of a dither array with the incorrectly identified void pixel A, and Figure 5b shows the pattern with the correct void pixel C identified.



(a) (b)

(a) (b)

Figure 5. Resulting periodic patterns based on different void finding criteria. (a) Using location A from Figure 4a, and (b) using the correct location C.

* R=6 and R=10 *are now replaced with* R=5 and R=8.3, *respectively*

† g=1/256 *should now be read as* g=2/256

‡ Figure 5a has now been replaced. Original Figures are shown in the gray box.

Mantissa Limit

The number of mantissa bits, M, limits the extent R of G(r) in cases of short principle wavelength that occur near g=1/2. In such cases, the problem is not the absolute value of the contribution of a pixel, but its value relative to the strongest contributing pixel. Since pixels closer to the filter center are "seen" much brighter, the situation is analogous to detecting candles next to the sun. In the presence of a closer pixel at r_0, the effective filter extent is the farthest pixel at R that will still contribute to the convolution. This is governed by the size of M:

$$\frac{G(r_0)}{G(R)} \approx 2^M,$$

or

$$R \approx \sqrt{2M \ln 2\sigma^2 + r_0}.$$

In short wavelength cases, the worst case is $r_0 = 1$. For the case of $\sigma = 1.5$, $R \approx 8.53$ for single precision and $R \approx 12.78$ for double precision.

Figure 6 illustrates the problem of finding the center of the largest cluster in a short wavelength pattern. This example recursive tessellation pattern is chosen because the correct answer is well known. The same extents are used as in Figure 4, resulting in the identification of pixels A, B, and C. The tiny contribution from the absence of the one black pixel not in the checkerboard must be felt in the presence of many other much weightier contributions surrounding the candidate central pixels to make a difference. Note again that these results are independent of the filter width.

Figure 7a shows a g=126/256 pattern at 50 dpi resulting from the incorrect identification of cluster pixel A, and Figure 7b shows the pattern with the correct cluster pixel C identified.

*

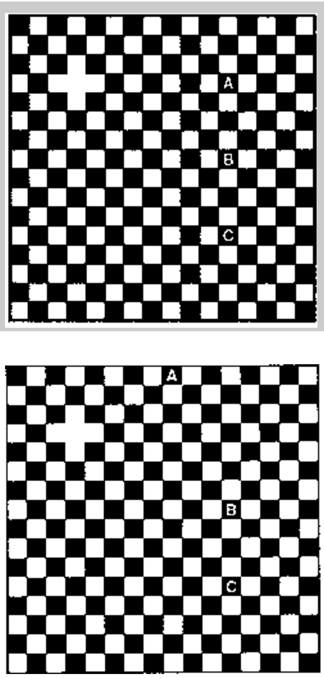


Figure 6. Locating the center of the largest cluster in a short wavelength pattern. The resulting locations A, B, and C, using filters with different extents, R=6, R=10,† and R>11.3, respectively.

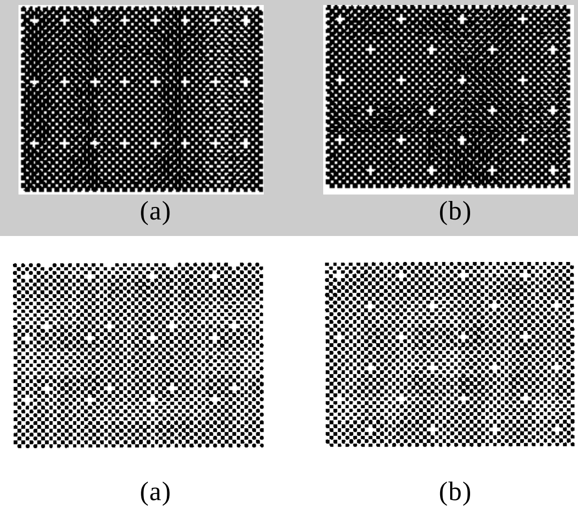


Figure 7. Resulting periodic patterns based on different cluster finding criteria. (a) Using location A from Figure 6, and (b) using the correct location C.

Blue Noise Example

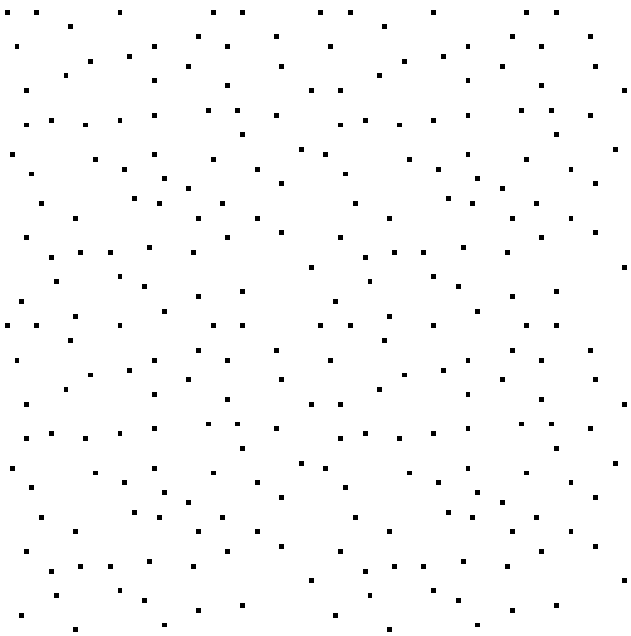
The examples in Figure 4 and Figure 6 indicate how the void-and-cluster method can be used to automatically generate recursive tessellation arrays. The more important use of the method is, of course, the generation of dither templates that can produce blue noise patterns. In Figure 8, two 64 by 64 dither patterns of gray level g=4/256 are displayed at 50 dpi with the void-and-cluster method using the same width of $\sigma = 1.5$, but different filter extents. The patterns in both (a) and (b) include 4 periods to illustrate the wrap-around effects, Figure 8a resulted from a Gaussian filter with an extent of only $3\sigma = 4.5$, and resembles the Figure 4 from Lin's paper[10] presented at this conference last year. In that paper, the lack of an adaptive width was cited as the shortcoming. The actual shortcoming is due to the truncation of the filter in that simulation. Figure 8b, shows the more uniform pattern that results when a filter with any extent greater than 8 is used.

Conclusion

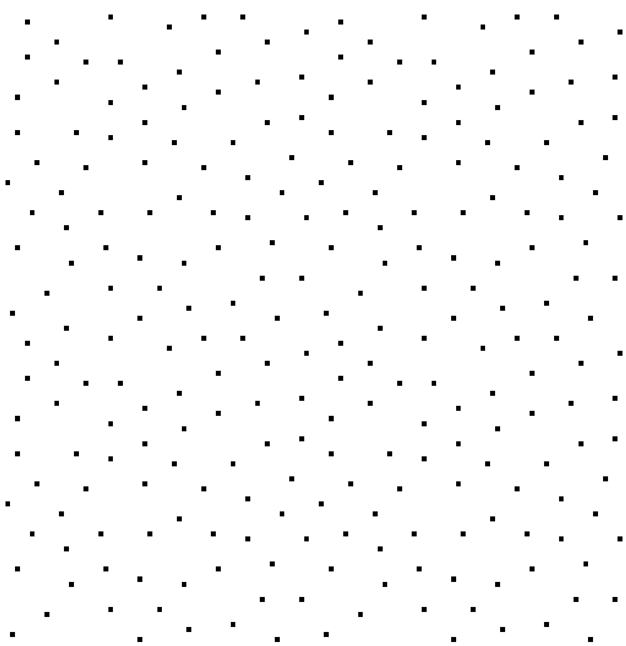
A key property of a Gaussian filter as a void-and-cluster finding filter is its unbounded extent. Unnecessarily truncating its extent can produce unwanted asymmetries in both long *and* short principle wavelength patterns. This paper demonstrated this for the cases of recursive tessellation and blue noise dither templates. Adapting filter widths to the principle wavelength (or gray level) does not appear to be have an appreciable effect on quality.

* Figures 6 and 7a have now been replaced. Original Figures are shown in the gray boxes.

† R=6 and R=10 *are now replaced with* R=5 and R=8.3, *respectively*



(a)



(b)

Figure 8. Long wavelength patterns generated with identical filter widths of σ =1.5, but different extents, (a) R=4.5, and (b) R>8.

References

1. R. Ulichney, *Digital Halftoning*, The MIT Press, Cambridge, chapter 6, 1987.
2. B. E. Bayer, “An optimum method for two level rendition of continuous-tone pictures,” *Proc. IEEE Int. Conf. Commun.,Conference Record*, pp. (26-11)-(26-15), 1973.
3. R. Ulichney, “The Void-and-Cluster Method for Generating Dither Arrays,” *IS&T/SPIE Symposium on Electronic Imaging Science & Technology*, San Jose, CA, vol. **1913**, Feb. 1-5, 1993, pp. 332-343.
4. R. Ulichney, “Dithering with blue noise,” *Proc. IEEE*, vol. **76**, no. 1, pp. 56-79, 1988.
5. J. Dalton, “Perception of binary textures and the generation of stochastic halftone screens”, *Human Vision, Visual Processing, and Digital Display VI, Proc. SPIE*, vol. **2411**, pp. 207-220, 1995.
6. M. Schulze and T. Pappas, “Blue noise and model-based halftoning,” *Human Vision, Visual Processing, and Digital Display VI, Proc. SPIE*, vol. **2179**, pp. 182-194, 1994.
7. T. Mitsa and K.J. Parker, “Digital halftoning technique using a blue-noise-mask,” *J. Opt. Soc. Am. A*, vol. **9**, pp. 1920-1929, 1992.
8. R. Ulichney, “Filter Design for void and cluster dither arrays”, *Proc. SID Int. Symposium*, June, 1994.
9. T. Mitsa and P. Brathwaite, “Wavelets as a tool for the construction of a halftone screen”, *Human Vision, Visual Processing, and Digital Display VI, Proc. SPIE*, vol. **2411**, pp. 228–238, 1995.
10. Q. Lin, “Improving halftone uniformity and tonal response”, *IS&T 10th Int. Congress on Advances in Non-Impact Printing*, New Orleans, Oct, 1994.

❋ Previously published in *IS&T’s NIP 11*, 1995, pp. 430–433.