Error Diffusion: Wavefront Traversal and Contrast Considerations

Avi C. Naiman and David T. W. Lam
Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

Phone: +852 2358-6997 Fax: +852 2358-1477
e-mail: avi@cs.ust.hk and dada@cs.ust.hk

Abstract

Error diffusion algorithms are used widely to halftone grayscale images. We present three enhancements to the standard approach: *wavefront* traversal of the pixels, error weights based on the distances to a pixel's neighbors and contrast considerations in determining error weights dynamically. The novel algorithm is demonstrated to yield improved detail over standard and serpentine error diffusion and to provide greater control over the contrast in the halftone image.

Keywords: Error Diffusion, Halftoning, Wavefront, Contrast.

Introduction

When grayscale images are to be displayed on black and white devices, a set of pixels must be generated that optimizes the image quality. To compute the optimal bilevel image, information about the display technology, the human visual system and the intended viewing environment are necessary. Unfortunately, we do not yet know all of the factors that play roles in human perception and image formation, nor do we have quantitative metrics that correlate well with subjective image quality.

In the meantime, a variety of algorithms for *halftoning* grayscale images for bilevel presentation have been developed, and trial and error has led to substantial image quality improvements over the past few years. Major classes of halftoning algorithms include *ordered dither*, *dot diffusion*, *blue-noise masking*, *iterative search* and *error diffusion*, as well as interesting combinations of these. For summaries and pointers to the original literature on the various techniques, see [Geist *et al.* 93, Jarvis *et al.* 76, Knox 90, Knox 94, Ulichney 88]. Recent work has included attempts to take the human visual system into account [Lin 93, Mitsa *et al.* 93, Mulligan and Ahumada 92, Ostromoukhov *et al.* 94, Sullivan *et al.* 93] as well as to model the display technology in use [Lin and Wiseman 93, Pappas *et al.* 93, Rosenberg 93]. Here, we focus on improvements to the error diffusion algorithm, originally introduced in [Floyd and Steinberg 76] and improved upon by numerous researchers (see summaries above).

Binarization – the process of turning a grayscale pixel into either a black or white one – generates an error term for any pixels that were not originally black or white. The goal in error diffusion is to minimize inaccuracies in local mean luminance, by distributing the errors to neighboring pixels that have yet to be binarized.

For each pixel in an image, error diffusion involves the steps of:

- choosing a threshold value for binarization (the threshold is often fixed at 0.5, but adaptive thresholding has been introduced to take into consideration the most-recent binary outputs and/or to enhance edges; see, for example, [Billotet-Hoffman and Bryngdahl 83, Eschbach and Knox 91, Fawcett and Schrack 86, Knox and Eschbach 93]);
- diffusing the error to a set of neighboring pixels, by use of (predetermined) *weights* ([Eschbach 93, Floyd and Steinberg 76, Fawcett and Schrack 86]); and
- choosing the next pixel to process (see [Witten and Neal 82, Velho and Gomes 91, Zhang and Webber 93]).

We report here a novel technique to generate weights dynamically, with a predetermined component based on a neighbor's distance from the diffusing pixel as well as a dynamically-computed component based on the *contrast* between the two pixels.

We also introduce a novel traversal of the pixels designed to propagate errors outward from the center of an image, pushing all of the undiffused errors to the perimeter.

Error Diffusion: Radially-Symmetric Weights

In the error diffusion approach to halftoning, each pixel's gray value, $G(P)$ (in the range [0.0–1.0]) is binarized by comparing it with a threshold – typically 0.5 – to determine whether to set the output pixel at that location to 0 (black) or 1 (white), $B(P)$. The error that this step incurs is $E(P) = G(P) - B(P)$, which is then *diffused* to its neighbors. The goal is to preserve the mean luminance in any localized region, by including any 'missing' luminance and removing any 'added' luminance.

A pixel's error, $E(P)$, must be divided amongst its neighbors that have not yet been binarized. Determining how to split up this error has, in general, been based on trial and error, rather than any principled approach (see, for example, [Floyd and Steinberg 76]). We propose that the error should be diffused in a radially-symmetric fashion, with each of the active neighbors (i.e., those that have not yet been binarized) receiving a proportion of the error that is determined by taking the physical layout of the output raster into consideration.

Fig. 1 demonstrates this concept for determining the amount of error – or *weight* – for each of a pixel P's neighbors. Here we assume that the output device's raster grid is rectilinear and its pixels are square and abutting, and we consider diffusing a pixel's error only to its 8 immediately-adjacent neighbors. The circle shown represents the area of diffusion that maximizes coverage of the neighbors' pixels without extending beyond their extents. The gray areas delineate the portions of the neighboring pixels that are covered by the circle, and the percentages superimposed on the gray areas indicate the relative proportions of error that each neighbor should receive, assuming all 8 neighbors have yet to be binarized. When only a subset of the neighbors are to be considered, these proportions must be normalized, in order to diffuse 100% of the error. The appropriate (normalized) weights are also shown for the case when only the 3 neighbors to the bottom-right of the pixel are active.

Wavefront Error Diffusion

A fundamental issue for any error diffusion algorithm is the *order in which pixels are traversed*.

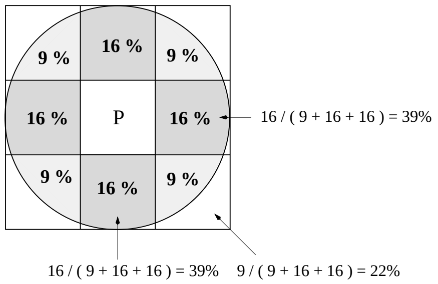

Figure 1: Radially-Symmetric Weights.

The initial error diffusion proposal suggested traversing the image row by row from top to bottom, and from left to right within each row [Floyd and Steinberg 76]. However, this technique exhibits a systematic directional dependency that results in a textured pattern. Improvements to this approach have included serpentine traversal – where alternating rows are traversed either from left to right or from right to left [Witten and Neal 82] – and space-filling curves [Witten and Neal 82, Velho and Gomes 91, Zhang and Webber 93].

We propose to apply the underlying concept of radially-symmetric error diffusion weights to the traversal of the pixels themselves. That is, instead of traversing the pixels along rows from one corner of the image to the other, we will traverse the image beginning at a seed starting pixel (e.g., the center of the image) and progressing outward in (square) rings. We call this approach *wavefront error diffusion*, because the processing proceeds in a fashion reminiscent of a wavefront in water emanating from an initial point of disturbance.

Fig. 2 introduces the technique of wavefront error diffusion. Consider the central, starting pixel, labeled '0' (all pixels are labeled with the order in which they are traversed). Since it is the first pixel to be processed, all of its 8 neighbors are active, as represented by the 8 arrows emanating from it. We proceed to process the rest of the pixels in square 'rings' (demarked by the dashed lines) emanating outward from the starting pixel. Within each ring, processing begins in the middle of the ring's top row and proceeds outward in both directions (potentially in parallel), stopping just before the corner pixels. The bottom row is then processed in a similar fashion (this can also be done in parallel with the top row), followed by the left and right columns (in parallel). Finally, the four corner pixels are processed. For the purposes of the discussion below, we call this traversal *row-first*, since the rows of a ring are processed before the columns.

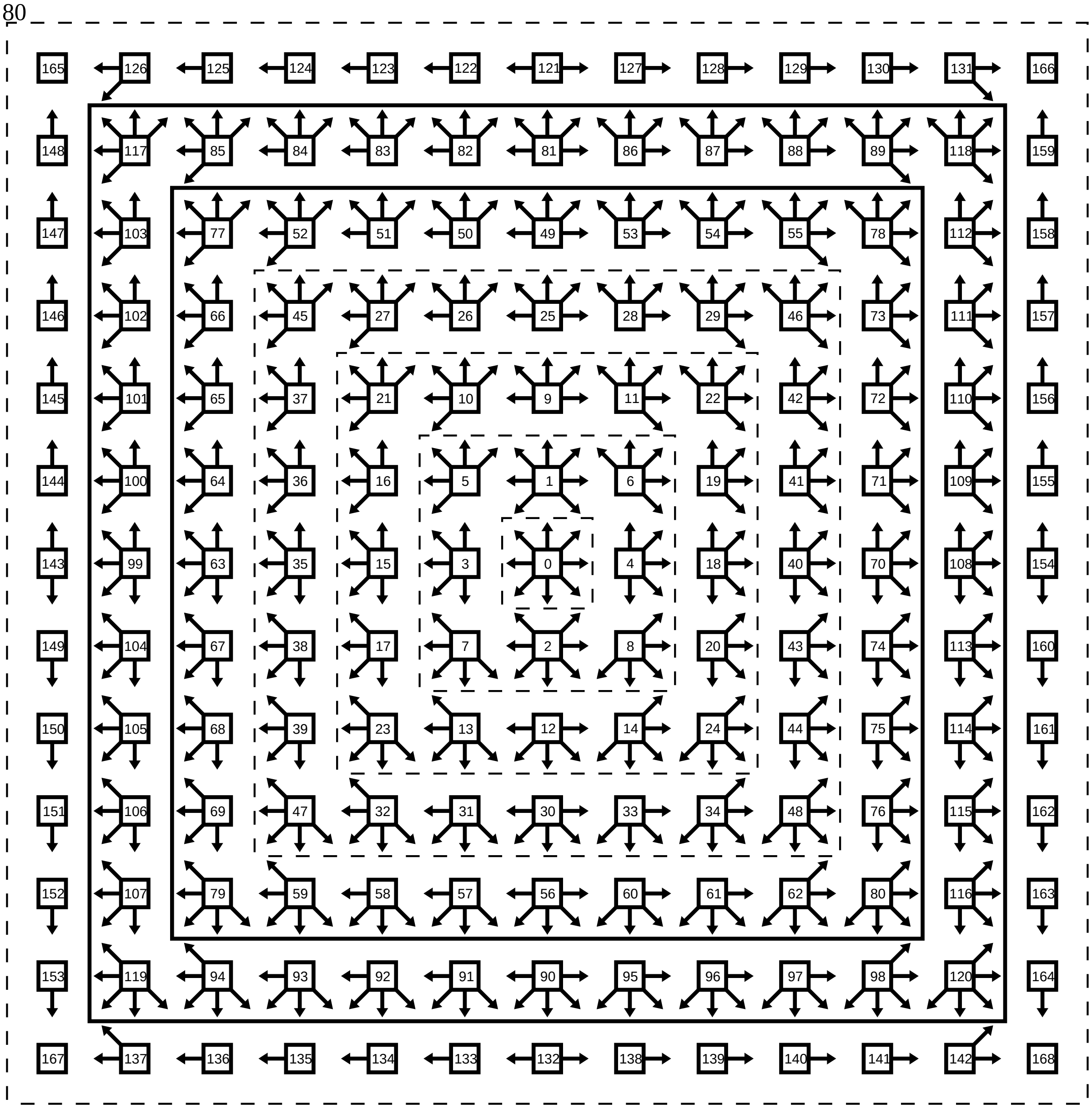

Figure 2: Wavefront Error Diffusion Traversal Order.

For example, consider ring 5 (the starting pixel is ring 0). The middle pixel of its top row, labeled 81, is processed first, and it has 5 active neighbors. The pixels to its left, 82-85, are processed next, followed by the pixels to its right, 86-89. The bottom row (90-98), left column (99-107), and right column (108-116) are then processed, finishing with the four corner pixels (117-120).

Note that, in this scenario, the starting pixel is the *only* pixel that has 8 active neighbors. Beginning with ring 3, all pixels have 4 active neighbors, except the middle pixels (in each row and column), the corner pixels, and the corner pixels' horizontally-adjacent pixels, all of which have 5 active neighbors. In ring 5, this row-first-influenced asymmetry can be seen in pixels 85, 89, 94 and 98, which each have 5 neighbors, whereas pixels 103, 107, 112 and 116 each have only 4 neighbors. We will deal with this issue below.

Fig. 3 is a grayscale image which we will use for comparisons. Fig. 4 is the result obtained from the row-first implementation of wavefront error diffusion, using the radially-symmetric weights introduced earlier. We immediately notice objectional

Figure 3: Original Grayscale Image.

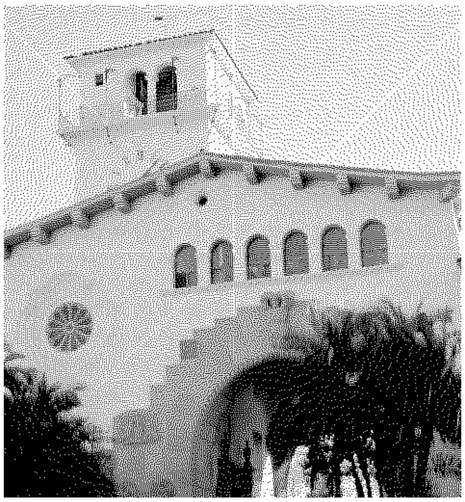

Figure 4: Row-First Wavefront Error Diffusion.

vertical, horizontal and diagonal lines emanating from the center of the image. We will first deal with the vertical and horizontal lines and then address the diagonal ones.

Vertical and Horizontal Lines

To understand the source of the vertical and horizontal lines, consider the error that is *accumulated* in (i.e., diffused *into*) a pixel from other pixels *in the same ring*. With reference again to ring 5 in Fig. 2, note that the middle pixels – 81, 90, 99 and 108 – all diffuse error to 2 neighbors within the same ring and do not accumulate any error at all from neighbors within the same ring. Except for the pixels at (or near) the corners, all other pixels in the ring both diffuse error to 1 neighbor within the ring and accumulate error from 1 neighbor within the ring.

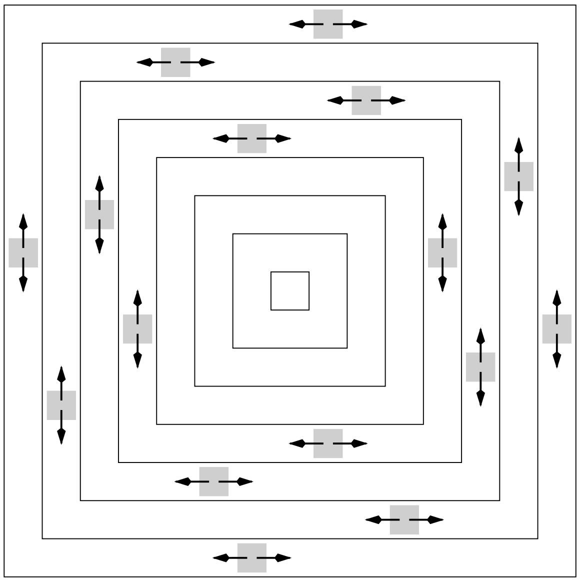

Figure 5: Random Starting Positions Within Rings.

Consequently, the central column and row of pixels distribute error outward but do not receive any error. This results in a coherent, differential error diffusion pattern which is readily visible.

We could process rings starting from the corners and working inward, but this would introduce an opposite, but similar, bias, with the central column and row now accumulating relatively more error from within the ring than the other pixels in the ring. A further possibility would be to alternate between diffusions outward and inward and this, indeed, appears to eliminate the vertical and horizontal lines. However, as we shall see below, diffusing error inward exacerbates the diagonal lines, so we shall avoid it. The solution, then, is to always diffuse error outward, but, rather than always beginning the diffusion at the middle of a ring's rows and columns, randomly selecting a starting pixel for each ring (Fig. 5), thereby eliminating the coherence of differential within-ring diffusion.

Diagonal Lines

To understand the source of the diagonal lines, consider the error that is accumulated in a pixel from pixels in the *previous* ring. With reference again to ring 5 in Fig. 2, note that most pixels accumulate error from 3 pixels in ring 4, but that the pixels adjacent to the corner pixels (85, 89, 94, 98, 103, 107, 112 and 116) accumulate error from only 2 pixels in ring 4, while the corner pixels (117-120) accumulate error from only 1 pixel in ring 4. Once again, this

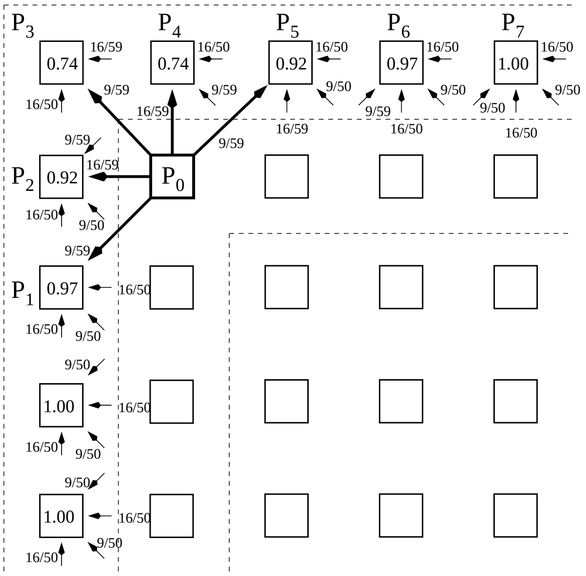

Figure 6: Shortfall in Error Accumulation Near Corners.

results in a coherent, differential error diffusion pattern which is readily visible, this time along the diagonals. Note that, if we diffuse the errors beginning at the corners and working our way inward, this differential accumulation would be further exacerbated, as no error would be accumulated in corner pixels from other pixels within the same ring. Hence, we always progress outward, as mentioned above.

The solution to the diagonal-lines problem can best be understood with reference to Fig. 6, which shows pixels in the top-left corner of a few rings. Consider the pixel, P_0, the top-left corner pixel from the ‘previous’ ring. It diffuses its error to 5 neighbors, all of which are in the next ring. Three of these neighbors are corner neighbors and 2 are side neighbors. The normalized, radially-symmetric weights, then, are 9⁄59 for the corner neighbors and 16⁄59 for the side neighbors, as shown by the arrows’ labels.

Now consider the pixel in the top-right corner of the diagram, labeled P_7. This pixel accumulates error from 4 neighbors, as represented by the 4 incoming arrows (keep in mind that the labels on each arrow represent radially-symmetric weights, normalized for the *source* pixels’ diffusion computations). For most pixels, these incoming weights sum to 1.0 (except in small rings). However, as we get closer to the top-left corner, the incoming weights sum to less than 1.0, as shown for each of the pixels P_1 through P_6. Hence, there is a coherent pattern of diminished error accumulation toward each of the corners, which is readily visible as diagonal lines.

To correct for these shortfalls, when diffusing the error from P_0, we scale the weights for each of its neighbors, so as to bring each of their accumulated errors to 1.0. For example, P_3’s accumulated error is only 0.74, so an extra 0.26 must come from the error diffused to it from P_0; i.e., the diffusion weight from P_0 to P_3 is increased from 9⁄59 to $9/59 + (1.0 - 0.74) = 0.41$. The adjusted weights for the other pixels to which P_0 diffuses error are: P_1: 0.18, P_2: 0.35, P_4: 0.53 and P_5: 0.23. However, P_6, which also accumulates less than 1.0 error, cannot be compensated for in this manner, as P_0 does not diffuse any error to P_6. Fortunately, this shortfall (0.03) is quite small and is likely to be inconspicuous.

Note that there is an asymmetry between the shortfalls in the rows and columns, as a function of pixel distance from the corner. This is a result of the rows being processed before the columns. To minimize this effect, for each ring, we randomly choose whether to process the rows before the columns or vice versa.

Fig. 7 shows the result of wavefront error diffusion with radially-symmetric weights and error compensation at the corners of each ring. A comparison with serpentine error diffusion [Witten and Neal 82] – but here using radially-symmetric weights – is shown in Fig. 8. Note the vertical streaking in the serpentine error diffusion image and the more-faithful rendition of the radial detail around the rosette-patterned window in the wavefront error diffusion image. However, strong worm-like substructure is visible in Fig. 7, particularly in the vertical and horizontal directions in bright areas.

Contrast Considerations in Error Diffusion

A variety of error diffusion weights have been proposed in the literature, primarily obtained through trial and error. One common feature is that the weights are all predetermined; i.e., they do not change throughout the entire image being processed. Consequently, when diffusing a pixel’s error to a set of neighbors, there is no possibility of taking into consideration the destination pixels’ values.

We propose that the error diffusion weights be computed dynamically, taking into consideration how visible the errors might be. Because the human visual system is sensitive to luminance *differences*, not absolute luminances [Blackwell 46], we use

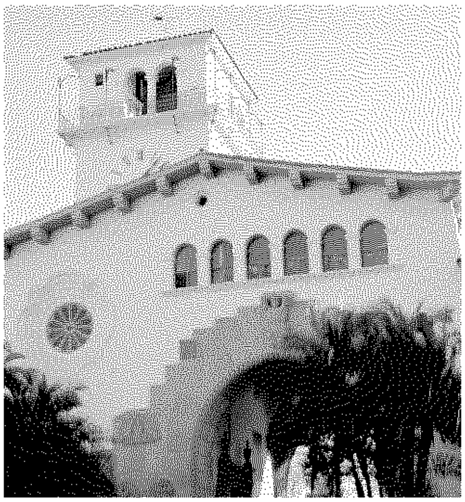

Figure 7: Radially-Symmetric Weights and Compensation at Corners.

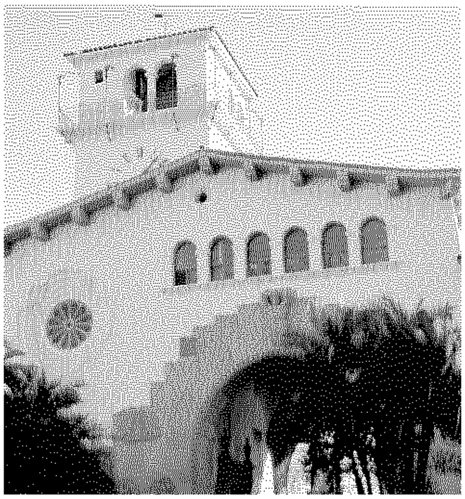

Figure 8: Serpentine Error Diffusion.

weights that are proportional to the *contrasts* between the pixel and its neighbors. That is, we diffuse proportionately more error to neighbors that are supposed to have high contrast with the pixel and less error to neighbors that are supposed to have low contrast. Although the same total error will be diffused to the set of neighbors, by apportioning the error based on contrast, the likelihood of diffused errors being objectionably visible – as high-frequency spatial modulation rather than luminance modulation – should be minimized.

For example, consider the case of a pixel with a (normalized) grayscale value of 0.3 and three neighbors with grayscale values of 0.4, 0.4 and 0.8. Binarizing the pixel to 0 results in an error of 0.3 to be diffused to the neighbors. If each of them receives an equal amount of the error, all three neighbors will (at least temporarily) rise to the binarization threshold of 0.5 and it is possible that all three will be binarized to 1. However, if proportionately more error is diffused to the neighbor with high contrast (i.e., 0.8), the two neighbors with relatively low contrast will (at least temporarily) remain below the binarization threshold, increasing the likelihood that they will eventually be binarized to 0.

We define the contrast between a pixel P and its j'th active neighbor, P_j, as the absolute difference between their *original* grayscale values $O(P)$ and $O(P_j)$ (i.e., before error diffusion / accumulation): $C(P_j) = |O(P) - O(P_j)|$. (Other possible values that could be used for computing contrast include the accumulated grayscale values before binarization and the binary values. However, it seems to us that, in considering contrast to determine diffusion weights, we should use the *intended* contrast, i.e., the original grayscale values.) The mean contrast of a set of n active neighbors of P, then, is: $C_{mean}(P) = \sum_{j=1}^{n} C(P_j)/n$. The diffusion weight for neighbor P_j is:

$$W(P_j) = \frac{E(P)}{n}\left(1 + \frac{C(P_j) - C_{mean}(P)}{C_{mean}(P)}\right).$$

If each of the neighbors has the same, non-zero contrast with $O(P)$, then each receives an identical portion of the error. If C_{mean} is 0 – i.e., a pixel and all its active neighbors have the same original grayscale value – we must take appropriate measures to assign the diffusion weights based solely on the radially-symmetric weights. When the contrast of the neighbors varies, those with higher contrast receive more of the error than those with lower contrast. As these weights are independent of the radially-symmetric weighting discussed above, the two weights for each neighbor must be multiplied before normalizing over all of the neighbors. Fig. 9 shows the result of using both types of weights when determining how to diffuse a pixel's error. Note that, although areas of uniform luminance – such as the sky in the upper right – have less noticeable substructure, the definition of edges – e.g., the left side of the tower – has been blurred.

Several error diffusion algorithms use a preprocessing step to enhance – and, thereby, preserve – edges [Jarvis *et al.* 76, Knuth 87]. Recent work has demonstrated the effectiveness of simple edge detection heuristics [Buchanan and Verevka 95, Velho and Gomes 95]).

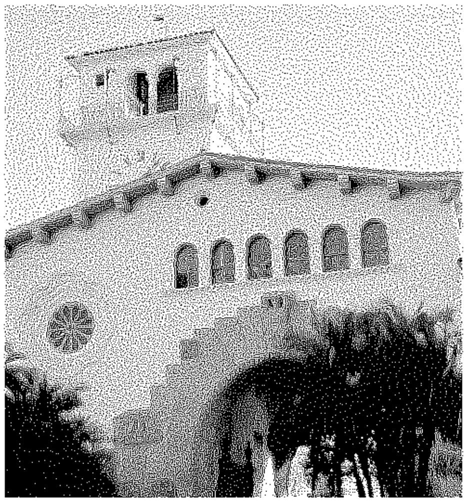

Figure 9: Weights Computed Based on Contrast.

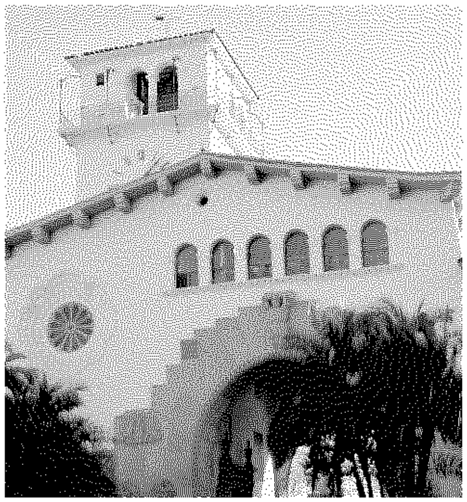

Figure 11: Added Minimum Contrast: 100 / 256.

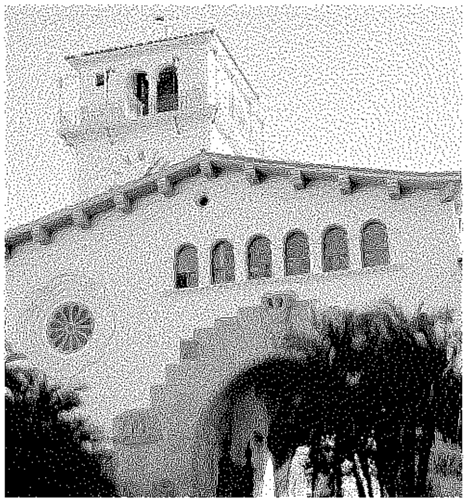

Figure 10: Added Minimum Contrast: 0.01 / 256.

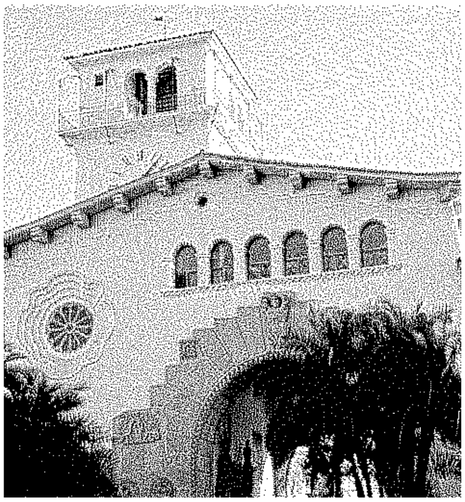

Figure 12: Avoiding Contrast Reversals.

Avoiding Contrast Reversals

Our approach is to directly incorporate contrast manipulation by adding a minimum contrast C_{min} to both the numerator and denominator of the contrast computation:

$$C(P_j) = \frac{|O(P) - O(P_j)| + C_{min}}{1 + C_{min}} \quad .$$

This has the added benefit of obviating the need to test for a zero mean contrast. For example, Fig. 10 uses an added minimum contrast of 0.01/256 while Fig. 11 uses an added minimum contrast of 100/256. Note how straight lines – for example, the left, vertical edge of the clock tower, the arm of the clock and the ledge beneath the 6 windows – have greater clarity with increasing C_{min}. On the other hand, the circular detail around the rosette-patterned window becomes muted with increasing C_{min}.

Consider the case in which a pixel's original grayscale value $O(P)$ is *less* than the original grayscale value of one of its neighbors $O(P_j)$. If the accumulated pixel value, $G(P)$, is greater than the threshold, then $B(P)$ will be 1 and the error will be negative (i.e., some light has to be removed from the neighbors). However, if the (negative) error diffused to a neighbor reduces $G(P_j)$ below the threshold, this results in a *contrast reversal* between these two pixels, the worst of all possibilities on a local scale. Consequently, we may want to exclude all such neighbors from consideration when determining the diffusion weights, as shown in Fig. 12. Note that, although the image has considerably greater contrast, its white-noise component is too pronounced.

Conclusions

Error diffusion has the potential to produce higher-quality images than many other digital halftoning approaches, such as ordered dither and clustered-dot diffusion. Despite the numerous improvements that have been made since error diffusion was introduced 20 years ago, a variety of possible advances have yet to be explored. We introduced three such possibilities here:

- wavefront traversal of the pixels;
- radially-symmetric error weights based on the distances to a pixel's neighbors; and
- contrast-based error weights that distribute more error to neighbors that are supposed to have relatively more contrast with the pixel, along with contrast manipulation that provides edge enhancement.

We have only begun to explore the power of these enhancements, as well as the interactions between them. The novel approach of taking the destination pixels into consideration to determine diffusion weights dynamically provides heretofore unrealized flexibility that we expect will yield substantial novel twists on the basic error diffusion approach.

Future research in halftoning algorithms is likely to explore better models of output devices and improve quantitative metrics for image quality. Aspects of the techniques presented here that are worthy of immediate investigation include enhancements to the contrast function, minimization of the computational overhead incurred by computing diffusion weights dynamically, and evaluation of the effects of these techniques on synthetic images. The most challenging task will be to carry out visual experiments to quantify the relative image-quality improvements that the new techniques presented here make possible.

Acknowledgements

This research was supported under RGC grant CERG HKUST600/94E (Hong Kong). We thank Walter Makous and Alain Fournier for illuminating discussions on contrast and ring equalization. We also thank the anonymous reviewers for several helpful suggestions.

References

Billotet-Hoffman and Bryngdahl 83
C. Billotet-Hoffman and O. Bryngdahl, ''On the Error Diffusion Technique for Electronic Halftoning,'' *Proceedings of the Society for Information Display*, Volume 24, Number 3, 1983, pp. 253-258.

Blackwell 46
H. Richard Blackwell, ''Contrast Thresholds of the Human Eye,'' *Journal of the Optical Society of America*, Volume 36, Number 11, November 1946, pp. 624-643.

Buchanan and Verevka 95
John W. Buchanan and Oleg Verevka, ''Edge Preservation with Space-Filling Curve Halftoning,'' *Graphics Interface 1995*, 1995, pp. 75-82.

Eschbach 93
Reiner Eschbach, ''Reduction of Artifacts in Error Diffusion by Means of Input-Dependent Weights,'' *Journal of Electronic Imaging*, Volume 2, Number 4, October 1993, pp. 352-358.

Eschbach and Knox 91
Reiner Eschbach and Keith T. Knox, ''Error-Diffusion Algorithm with Edge Enhancement,'' *Journal of the Optical Society of America A*, Volume 8, Number 12, December 1991, pp. 1844-1850.

Geist *et al.* 93
Robert Geist, Robert Reynolds, and Darrell Suggs, ''A Markovian Framework for Digital Halftoning,'' *ACM Transactions on Graphics*, Volume 12, Number 2, April 1993, pp. 136-159.

Fawcett and Schrack 86
G. S. Fawcett and G. F. Schrack, ''Halftone Techniques Using Error Correction,'' *Proceedings of the Society for Information Display*, Volume 27, 1986, pp. 305-308.

Floyd and Steinberg 76
Robert W. Floyd and Louis Steinberg, ''An Adaptive Algorithm for Spatial Greyscale,'' *Proceedings of the Society for Information Display*, Volume 17, Number 2, 1976, pp. 75-77.

Jarvis *et al.* 76
J. F. Jarvis, C. N. Judice, and W. H. Ninke, ''A Survey of Techniques for the Display of Continuous-Tone Pictures on Bilevel Displays,'' *Computer Graphics and Image Processing*, Volume 5, 1976, pp. 13-40.

Knox 90
Keith T. Knox, ''Digital Halftoning Algorithms and Parameters,'' *Proceedings of the International Conference on Lasers '90,* 1990, pp. 619-625.

Knox 94
Keith T. Knox, ''Introduction to Digital Halftones,'' *IS&T's 47th Annual Conference*, 1994, pp. 456-458.

Knox and Eschbach 93
Keith T. Knox and Reiner Eschbach, ''Threshold Modulation in Error Diffusion,'' *Journal of Electronic Imaging*, Volume 2, Number 3, July 1993, pp. 185-192.

Knuth 87
Donald E. Knuth, ''Digital Halftones by Dot Diffusion,'' *ACM Transactions on Graphics*, Volume 6, Number 4, October 1987, pp. 245-273.

Lin 93
Qian Lin, ''Halftone Image Quality Analysis Based on a Human Vision Model,'' *SPIE Proceedings: Human Vision, Visual Processing and Digital Display IV*, Volume 1913, February 1993, pp. 378-388.

Lin and Wiseman 93
Qian Lin and Jeanne Wiseman, ''Impact of Electrophotographic Printer Dot Modeling on Halftone Image Quality,'' *Society for Information Display 93 Digest*, Volume 24, 1993, pp. 147-150.

Mitsa *et al.* 93
Theophano Mitsa, Krishna L. Varkur, and Jennifer R. Alford, ''Frequency-Channel-Based Visual Models as Quantitative Quality Measures in Halftoning,'' *SPIE Proceedings: Human Vision, Visual Processing and Digital Display IV*, Volume 1913, February 1993, pp. 390-401.

Mulligan and Ahumada 92
Jeffrey B. Mulligan and Albert J. Ahumada, ''Principled Halftoning Based on Human Vision Models,'' *SPIE Proceedings: Human Vision, Visual Processing and Digital Display III*, Volume 1666, February 1992, pp. 109-121.

Ostromoukhov *et al.* 94
Victor Ostromoukhov, Roger D. Hersch, and Isaac Amidror, ''Rotated Dispersed Dither: A New Technique for Digital Halftoning,'' *Computer Graphics*, August 1994, pp. 123-130. Proceedings of SIGGRAPH 94; Annual Conference Series.

Pappas *et al.* 93
Thrasyvoulos N. Pappas, Chen-Koung Dong, and David L. Neuhoff, ''Measurement of Printer Parameters for Model-Based Halftoning,'' *Journal of Electronic Imaging*, Volume 2, Number 3, July 1993, pp. 193-204.

Rosenberg 93
Charles J. Rosenberg, ''Measurement-Based Evaluation of a Printer Dot Model for Halftone Algorithm Tone Correction,'' *Journal of Electronic Imaging*, Volume 2, Number 3, July 1993, pp. 205-212.

Sullivan *et al.* 93
James R. Sullivan, R. Miller, and G. Pios, ''Image Halftoning Using a Visual Model in Error Diffusion,'' *Journal of the Optical Society of America A*, Volume 10, Number 8, August 1993, pp. 1714-1724.

Ulichney 88
Robert A. Ulichney, *Digital Halftoning,* The MIT Press, Cambridge, Massachusetts, 1988.

Velho and Gomes 91
Luiz Velho and Jonas de Miranda Gomes, ''Digital Halftoning with Space Filling Curves,'' *Computer Graphics*, Volume 25, Number 4, July 1991, pp. 81-90. SIGGRAPH 1991 Proceedings.

Velho and Gomes 95
Luiz Velho and Jonas de Miranda Gomes, ''Stochastic Screening Dithering with Adaptive Clustering,'' *Computer Graphics*, August 1995, pp. 273-276. Proceedings of SIGGRAPH 95; Annual Conference Series.

Witten and Neal 82
Ian H. Witten and Radford M. Neal, ''Using Peano Curves for Bilevel Display of Continuous-Tone Images,'' *IEEE Computer Graphics and Applications*, Volume 2, Number 5, May 1982, pp. 47-52.

Zhang and Webber 93
Yuefeng Zhang and Robert E. Webber, ''Space Diffusion: An Improved Parallel Halftoning Technique Using Space-Filling Curves,'' *Computer Graphics*, August 1993, pp. 305-312. SIGGRAPH 93 Proceedings.