

Block Interlaced Pinwheel Error Diffusion

Pingshan Li* and Jan P. Allebach†

*Sony Electronics Inc., 3300 Zanker Rd., San Jose, CA 95134

†School of Electrical and Computer Engineering,
Purdue University, West Lafayette, IN 47907

Abstract

Error diffusion is a popular halftoning algorithm that in its most widely used form, is inherently serial. As a serial algorithm, error diffusion offers limited opportunity for large-scale parallelism. In some implementations, it may result in excessive bus traffic between the on-chip processor and the off-chip memory used to store the modified continuous-tone image and the halftone image. We introduce a new error diffusion algorithm in which the image is processed in two groups of interlaced blocks. Within each group, the blocks may be processed entirely independently. In the first group, the error diffusion proceeds along an outward spiral from the center of the block. Errors along the boundaries of blocks in the first group are diffused into neighboring blocks in the second group, within which the error diffusion spirals inward. A tone-dependent error diffusion training framework is used to eliminate artifacts associated with the spiral scan paths. We demonstrate image quality which approaches that achieved by conventional line-by-line error diffusion.

1. Introduction

Digital halftoning is the process of transforming a continuous-tone image to a binary image so that it can be printed or displayed with a bi-level device. Error diffusion is a popular algorithm that can be used to produce high quality halftone images. However, it is inherently serial. We cannot process the last pixel until the rest of the image has been processed.

To increase the parallelism associated with error diffusion, authors have considered modifications to the order of processing or scan pattern, and/or modifications to the error diffusion architecture itself. Modifications to the error diffusion architecture include grouping pixels into cosets which are binarized in successive passes¹ according to an optimal ordering.² Within each pass, all pixels may be processed independently and in parallel. This strategy has also been combined with the use of a space-filling curve.³ Other strategies based on space-partitioning include Refs. 4 and 5. Another approach is to partition the processing among multiple processors which are tightly coupled.⁶

In this paper, we introduce a pinwheel error diffusion algorithm in which the image is divided into two groups of non-overlapping blocks. The blocks in each group can be processed in parallel. Boundary artifacts are minimized by allowing interaction between the blocks from different groups. Of the references cited above, Ref. 4 is perhaps most similar to our work, in that it too is block-based with some communication of error information across block boundaries. However, aside from these very broad similarities, the two approaches are really quite different.

2. Tone Dependent Error Diffusion

In this paper, we develop block interlaced error diffusion based on the tone dependent error diffusion (TDED) introduced in Ref. 7, since this algorithm produces high quality halftone images compared with the traditional Floyd-Steinberg error diffusion.⁸ The input continuous-tone image $f[m, n]$ is modified by previous quantizer errors. Let $u[m, n]$ denote the modified pixel value. The error-weighting matrix $w_{k,\ell}(a)$ and threshold matrices $t[m, n; a]$ are functions of input pixel value a . The binary output $g[m, n]$ is determined by

$$g[m, n] = \begin{cases} 1, & \text{if } u[m, n] \geq t[m, n, f[m, n]], \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

The quantizer error $d[m, n]$ is given by

$$d[m, n] = g[m, n] - u[m, n]. \quad (2)$$

The continuous-tone pixel values are updated according to

$$u[m+k, n+\ell] \leftarrow u[m+k, n+\ell] - w_{k,\ell}(f[m, n])d[m, n]. \quad (3)$$

The threshold matrix we use is based on a halftone pattern for the absorbance 0.5.

$$t[m, n; a] = \begin{cases} t_u(a), & \text{if } p[m, n; 0.5] = 0, \\ t_\ell(a), & \text{otherwise.} \end{cases} \quad (4)$$

Here $t_u(a)$ and $t_\ell(a)$ are tone dependent parameters satisfying $t_u(a) \geq t_\ell(a)$. The function $p[m, n; 0.5]$ is a halftone pattern generated by direct binary search (DBS)⁹ to represent a constant patch with absorbance 0.5. It is doubly periodic in m and n with period 128.

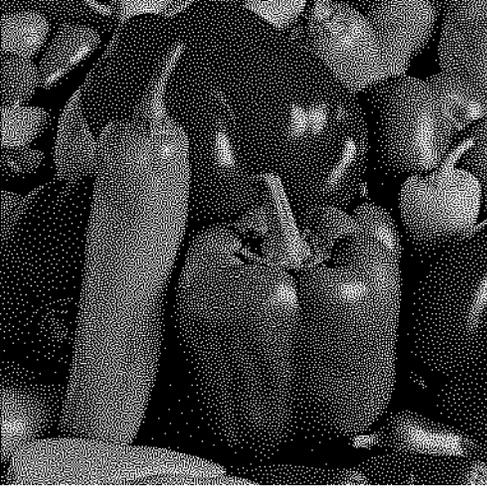


Figure 1: Halftone image generated by tone dependent error diffusion printed at 150 dpi.

The error weighting matrix and the thresholds are optimized level by level based on a human visual system (HVS) model. For highlight and shadow gray levels where the minority pixels are sparse, we optimize the TDED filters by minimizing the perceived total squared error between the constant value image and the corresponding halftone image.

The frequency response of the HVS filter is given by

$$H(\bar{u}, \bar{v}) = \exp \left[- \left(\frac{180}{\pi} \right) \frac{\sqrt{\bar{u}^2 + \bar{v}^2}}{c \ln(L) + d} \right], \quad (5)$$

where \bar{u} and \bar{v} are the frequency variables in cycles/radian subtended at the retina, L is the average luminance in cd/m^2 , $c = 0.525$, and $d = 3.91$. The point spread function $h(\bar{x}, \bar{y})$ is obtained by taking the inverse Fourier transform of $H(\bar{u}, \bar{v})$. Here (\bar{x}, \bar{y}) is in units of radians subtended at the retina. Since a length x when viewed at a distance D subtends an angle of $\bar{x} = \tan^{-1}(x/D) \approx x/D$ for $x \ll D$, the HVS filter with units measured on the printed media can be computed as

$$p_{\text{HVS}}(x, y) = \frac{1}{D^2} h\left(\frac{x}{D}, \frac{y}{D}\right). \quad (6)$$

Let $e[m, n]$ denote the error image obtained by $e[m, n] = g[m, n] - f[m, n]$. Then the perceived error between the halftone image and the continuous-tone image is represented as

$$\tilde{e}(x, y) = \sum_m \sum_n e[m, n] \tilde{p}(x - mX, y - nX), \quad (7)$$

where X corresponds to the lattice of addressable points for the output device; and $\tilde{p}(x, y) = p_{\text{HVS}}(x, y) ** p_{\text{dot}}(x, y)$ is the printer spot profile convolved with the HVS filter.

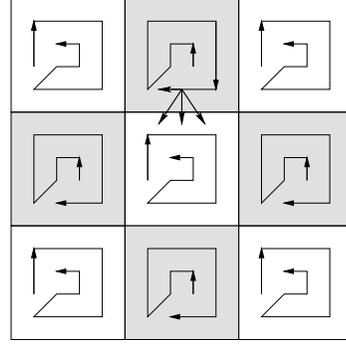


Figure 2: Block interlaced pinwheel error diffusion with serpentine scan. The shaded blocks are outward spiral blocks. The white blocks are inward spiral blocks.

We shall assume that $p_{\text{HVS}}(x, y)$ has much larger extent than $p_{\text{dot}}(x, y)$ and hence that $\tilde{p}(x, y) \approx p_{\text{HVS}}(x, y)$. In this paper we choose $L = 10 \text{ cd/m}^2$, $D = 11 \text{ in}$ and $1/X = 300 \text{ dpi}$. The error metric used as a cost function is the total squared error given by

$$E = \iint |\tilde{e}(x, y)|^2 dx dy. \quad (8)$$

For midtones, we found that minimizing the cost function (8) did not yield satisfactory results. Instead we use the binary textures generated by DBS⁹ under the same HVS model as the reference. We obtain the optimal TDED weights and thresholds by minimizing the mean squared error between the expected magnitude images of the spectra of halftones generated by TDED and by DBS.⁷

Figure 1 shows an image obtained by TDED with serpentine scan.

3. Block Interlaced Pinwheel Error Diffusion

In this section, we present a block-interlaced pinwheel error diffusion (BIPED) algorithm. It attempts to retain both the quality of serial error diffusion and the parallelism of screening without causing visible artifacts at the boundary between blocks. For pinwheel error diffusion, we divide the image into two groups of blocks. The blocks in each group can be processed independently. Blocks from the two groups are interlaced in a checkerboard pattern on the image. The first group contains outward spiral blocks. For each outward spiral block, the error diffusion starts from the center; and the error is diffused toward the block boundary as shown in Fig. 2. At the boundary, the error is diffused to the neighboring blocks in the second group.

A block in the second group can be processed after its four neighboring blocks from the first group have been processed. Blocks in the second group are inward spiral

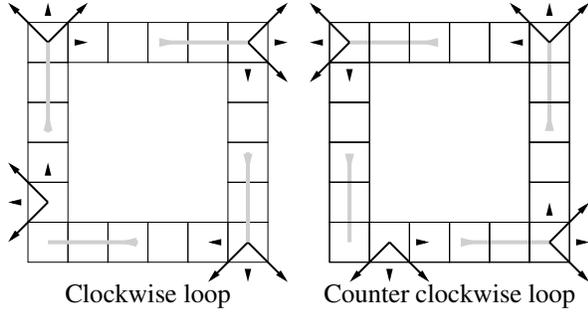


Figure 3: Corner weight configuration for pinwheel error diffusion outward spiral blocks.

blocks as is also shown in Fig. 2. The error diffusion starts from the boundary, and the error is diffused toward the center. The first loop along the boundary picks up the errors diffused from the neighboring outward spiral blocks. The pixels on either side of the boundary between blocks are scanned in the opposite direction, so that the error diffusion around the first loop in the inward spiral blocks is a regular error diffusion process with serpentine scan.

We will extend the TDED optimization technique to BIPED. In order to obtain high quality and the consistency of the texture appearance in the vertical and horizontal scanning regions, we use the following cost function for the midtones:

$$\varepsilon = \sum_k \sum_\ell \left[\left(|\hat{G}^{TDED}[k, \ell]| - |\hat{G}^{DBS}[k, \ell]| \right)^2 + \alpha \left(|\hat{G}^{TDED}[k, \ell]| - |\hat{G}^{TDED}[k, \ell]^T| \right)^2 \right], \quad (9)$$

where $|\hat{G}^{TDED}[k, \ell]|$ and $|\hat{G}^{DBS}[k, \ell]|$ are the expected magnitude images of the DFT of the halftone patterns created by TDED and by DBS, respectively. $(\cdot)^T$ denotes matrix transpose, and α is a nonnegative weight parameter. The second squared term on the right hand side of (9) measures the texture consistency in the vertical and horizontal directions. If α is large, the texture is less homogeneous, but appears to be smoother and more consistent in the vertical and horizontal directions. The results shown in the rest of this paper are obtained by choosing $\alpha = 0.25$. This produces high quality halftone images when applied to pinwheel error diffusion. In the highlight and shadow areas, we still use the cost function (8) to optimize the error diffusion filters.

For the corners of the spiral paths, we design special filters to obtain a smooth transition. The error diffusion systems for the outward spiral blocks and for the inward spiral blocks can be optimized independently. Figure 3 shows how we define the terms in the error-weighting matrix for the outward spiral corners. The same set of error

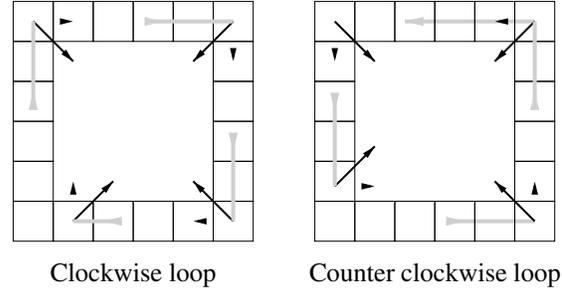


Figure 4: Corner weight configuration for pinwheel error diffusion inward spiral blocks.

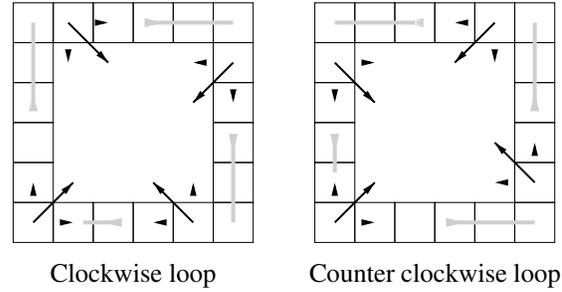


Figure 5: Next-to-corner weight configuration for pinwheel error diffusion inward spiral blocks.

weights is used for all the corners. We want the texture appearance at the corners to be consistent with that along the straight portions of the scan path. Therefore, we use the halftone patterns generated by the optimal TDED obtained by minimizing (9) as the reference when optimizing the corner filters for pinwheel error diffusion. Because the quality of the halftones in the outward spiral blocks is independent of that for the inward spiral blocks, we treat the entire image as a single block, and apply an outward spiral scan when optimizing the weights and thresholds for outward spiral blocks. The cost function for optimizing the corner parameters is given by

$$\varepsilon = \sum_k \sum_\ell \left[\left(|\hat{G}^{BIPED}[k, \ell]| - |\hat{G}^{TDED}[k, \ell]| \right)^2 + \left(|\hat{G}^{BIPED}[k, \ell]| - |\hat{G}^{TDED}[k, \ell]^T| \right)^2 \right], \quad (10)$$

where $|\hat{G}^{BIPED}[k, \ell]|$ denotes the expected magnitude of the DFT of the halftone pattern obtained by block interleaved pinwheel error diffusion. Here $|\hat{G}^{TDED}[k, \ell]|$ is the expected magnitude of the DFT of the halftone pattern generated by TDED optimized using (8) and (9) with regular serpentine scan. The two squared terms on the right hand side of (10) are weighted equally because the areas of the vertical and horizontal scanning regions are the same.

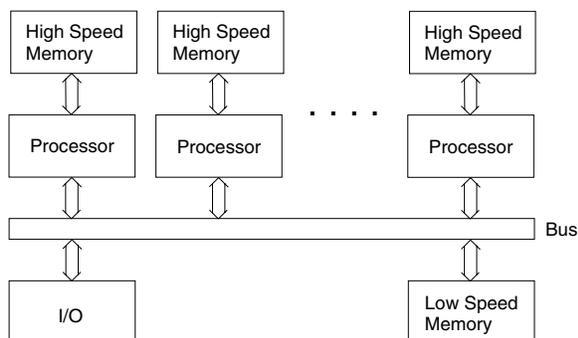


Figure 6: A generic system that includes one or more processors.

1	3	1	3	1
5	2	5	2	5
4	7	4	7	4
8	6	8	6	8

Figure 7: Stages for stripe-based processing of the image. Blocks are processed in the order as numbered. Blocks with the same number are processed independently. Shaded blocks are outward spiral blocks; white blocks are inward spiral blocks.

When optimizing the weights and thresholds at the corners, the error diffusion filters for the straight portions of the scan path are chosen to be the optimal filters obtained by using (8) and (9), and they are fixed. Similar to regular TDED optimization, the corner filters are optimized level by level. For each level, the initial values of the weights and thresholds are chosen to be the optimal values for regular TDED at the same level. At each level, the mean of the upper and lower thresholds remains constant during the optimization procedure.

For the inward spiral blocks, the corner weights are configured as in Fig. 4. The error weighting matrix to be used for the locations next to the corners has three terms as illustrated in Fig. 5. To obtain the optimal value of these parameters, we jointly optimize the weights at the corner and those at the next location.

4. Efficient Computation

The fundamental advantage of the BIPED algorithm is that the blocks within a group can be processed completely independently. Figure 6 shows a generic system architecture that may include one or more processors. The continuous-tone input and output halftone images are kept in the low-speed storage. The local high-speed storage associated with each processor is only large enough to hold a single block of pixels from the input image and a single block



Figure 8: Halftone image generated by pinwheel error diffusion with block size 8×8 , printed at 150 dpi.

of pixels from the output image, plus boundary pixels and other temporary variables. We distinguish three cases depending on the manner in which the input image is written into the low speed storage, and the manner in which the output image is read from the low speed storage.

Case 1: The input image is fully loaded into storage before the processing begins; and the output image is not read from storage until the halftoning process is completed.

The processors first independently transfer data from group 1 (outward spiral) blocks in the input image to their local storage, generate the corresponding halftone image blocks, and write them back out to the low-speed storage. The input image blocks will include a one-pixel border from adjacent group 2 blocks so that the continuous-tone image pixel values can be updated during the outermost error diffusion pass along the block edges. These updated pixel values are then written over the original continuous-tone image values kept in the low-speed storage. When all group 1 blocks have been processed, the group 2 blocks are processed independently.

Case 2: The input image arrives in stripes; and the halftone image is sent out in stripes to the marking engine as the stripes are completed.

In this case, the low-speed storage must be large enough to hold at least 2 entire rows of blocks of both the halftone and modified continuous-tone images, plus an additional row of pixels from a third row of blocks in the modified continuous-tone image. Figure 7 shows how the processing will take place. Initially the first two rows of input blocks are available. The group 1 blocks in both rows (stages 1 and 2) are processed. Then the group 2 blocks in row 1 can be processed (stage 3). At this point, the first row of blocks is completed, and can be sent to the marking

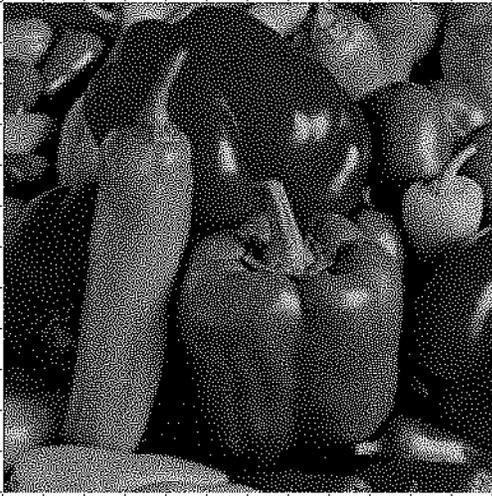


Figure 9: Halftone image generated by pinwheel error diffusion with block size 32×32 , printed at 150 dpi.

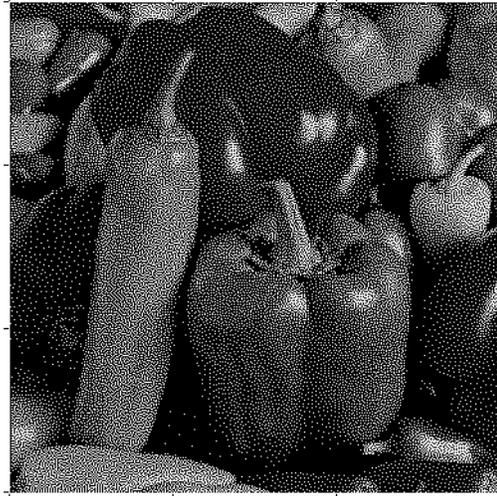


Figure 10: Halftone image generated by pinwheel error diffusion with block size 128×128 , printed at 150 dpi.

engine. Once the third row of input blocks is available, the group 1 blocks in row 3 can be processed (stage 4), followed by the group 2 blocks in row 2 (stage 5). At this point, row 2 is completed, and can be sent to the marking engine. The process continues in this manner until the entire image has been processed.

Case 3: The input image arrives at the low-speed data storage in randomly ordered superblocks. The entire output image is stored in low speed storage. We use the term superblock to distinguish from the blocks that comprise BIPED. A superblock will typically contain many blocks. The superblock boundary may not be perfectly aligned with the block boundaries. Group 1 blocks can be processed as soon as they are available. Any Group 2 block can be processed whenever its neighboring Group 1 blocks have been processed.

5. Results and Conclusion

Figures 8-10 show the results obtained with block size 8×8 , 32×32 , and 128×128 . They are printed at 150 dpi. There are tick marks around each image to indicate the boundaries of blocks. It can be seen that as the block size increases, the quality of the halftone image improves. The improvement is more noticeable in the highlight and shadow areas where the minority pixels are sparse, such as the dark pepper at the left edge of the picture. The halftone image shown in Fig. 10 is very similar to that generated with regular TDED shown in Fig. 1.

BIPED offers substantial opportunities for parallelism. In certain architectures, it will also reduce bus traffic between the on-chip processor and off-chip low-speed storage, assuming that the on-chip local high speed storage has

sufficient capacity for at least one entire block of the modified continuous-tone image. The disadvantage of block-interlaced error diffusion is that the low-speed memory must hold more intermediate data compared with a conventional line-by-line error diffusion algorithm.

Acknowledgments

This work was supported by the Hewlett-Packard Company.

References

1. D. E. Knuth, "Digital Halftones by Dot Diffusion," *ACM Trans. on Graphics* **6**, 245–273 (1987).
2. M. Meşe and P. P. Vaidyanathan, "Optimized Halftoning Using Dot Diffusion and Methods for Inverse Halftoning," *IEEE Trans. Image Processing* **9**, 691–709 (2000).
3. Y. Zhang and R. E. Webber, "Space Diffusion: An Improved Parallel Halftoning Technique Using Space-Filling Curves," *Proc. ACM SIGGRAPH 93 Conf. Computer Graphics*, 305–312 (1993).
4. Y. Takeuchi and H. Kunieda, "Space Partitioning Image Processing Technique for Parallel Recursive Half Toning," *IEICE Trans. on Fundamentals of Electronics Communications & Computer Sciences* (4), 603–612 (1993).
5. Y. Zhang, "Line Diffusion: A Parallel Error Diffusion Algorithm for Digital Halftoning," *Visual Computer* **12**, 40–46 (1996).
6. P. T. Metaxas, "Optimal Parallel Error-Diffusion Dithering," *Color Imaging: Device-Indep. Color, Color Hardcopy, and Graphic Arts IV, Proc. SPIE* **3648**, 485–494 (1999).
7. P. Li and J. P. Allebach, "Tone Dependent Error Diffusion," *Color Imaging: Device-Independent Color, Color Hardcopy, and Applications VII, Proc. SPIE* **4663** (2002).
8. R. W. Floyd and L. Steinberg, "An Adaptive Algorithm for Spatial Greyscale," *Proc. Soc. Inf. Disp.* **17**, 75–77 (1976).
9. M. Analoui and J. P. Allebach, "Model-Based Halftoning Using Direct Binary Search," *Human Vision, Visual Proc. and Digital Display III, Proc SPIE* **1666**, 96–108 (1992).