Improved Error Diffusion Algorithm Incorporating a Visual Model

Kevin E. Spaulding, Douglas W. Couwenhoven and Rodney L. Miller
Eastman Kodak Company
Imaging Research and Advanced Development
Rochester, New York 14650-1816
E-mail: spaulding@kodak.com

Abstract

This paper describes an improved error diffusion algorithm for the purpose of digitally halftoning images. In one variation of the algorithm an error signal is calculated by the difference between a visually perceived input value and a visually perceived output value. This is accomplished by applying a causal visual blur function to both the input and output images. This approach has the advantage that it minimizes the appearance of worm artifacts in the output image while simultaneously eliminating the edge artifacts associated with a previous visual error diffusion algorithm. In a second variation of the improved error diffusion algorithm, a local image activity detector is used to adaptively modify the input and output blur filters. This allows the error diffusion parameters to be optimized for different types of image content.

Keywords: digital halftoning, error diffusion, visual modeling

1 Introduction

Digital halftoning is a digital image processing technique used to produce a halftone output image from a continuous-tone input image.[1] A continuous-tone image is typically represented as a set of discrete pixel values ranging from 0 to 255. To reproduce this image on an output device capable of printing dots of one tone level (e.g., black) it is necessary to create the sensation of intermediate tone levels by suitably distributing the printed dots in the output image. This is accomplished by converting the continuous-tone image to a binary output image using some form of halftoning algorithm.

One type of digital halftoning is known as error diffusion. Figure 1 shows a block diagram describing a basic error diffusion algorithm developed by Floyd and Steinberg.[2] The continuous-tone input value for column i, and row j of the input image is given by $y_{i,j}$. For purposes of illustration it will be assumed that the continuous-tone input values span the range from 0 to 255. The continuous-tone input value for the current input pixel is thresholded to form the output value $b_{i,j}$. The threshold operator will return a 0 for any continuous-tone input value below the threshold, and a 255 for any continuous-tone input value above the threshold. A difference signal is computed between the continuous-tone input value and the output value, representing the error introduced by the thresholding process. The difference signal is multiplied by a series of error weights, $W_{i,j}$, and is added to the continuous-tone input values of nearby pixels that have yet to be processed to form modified continuous-tone input values. The propagation of the errors made during the quantization process to the nearby pixels ensures that the mean of the transformed pixel values is preserved over a local image region. Figure 2 illustrates the error weights use by Floyd and Steinberg to distribute the errors to the nearby pixels. Figure 3 shows an image generated using this simple error diffusion algorithm, together with the corresponding continuous-tone original.

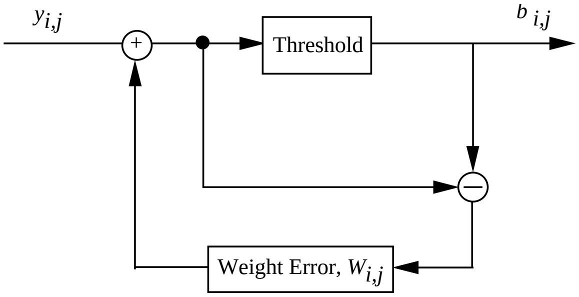

Figure 1. Basic Floyd-Steinberg error diffusion algorithm.

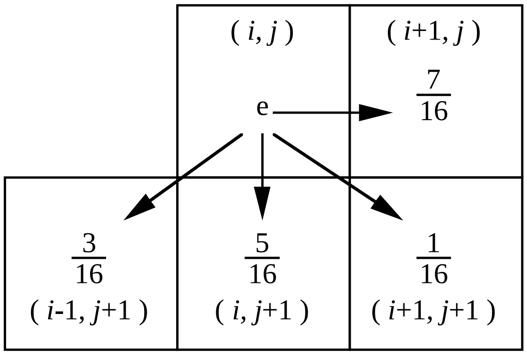

Figure 2. Error weights for Floyd-Steinberg error diffusion algorithm.

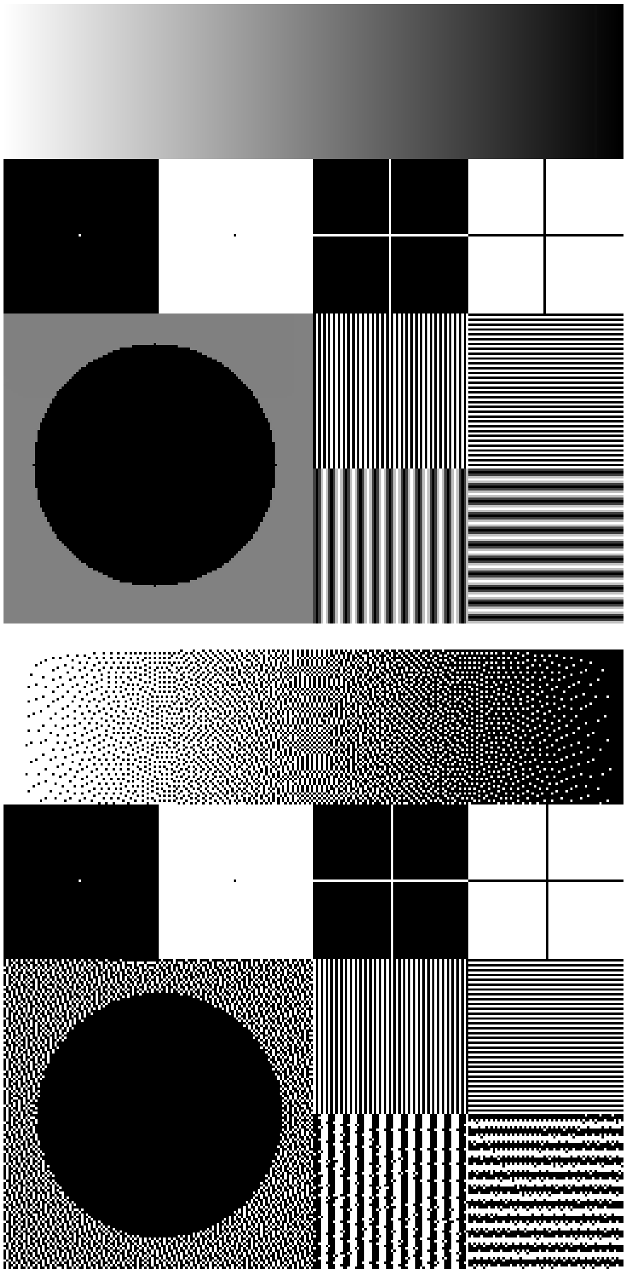

Figure 3. (a) Original image, and (b) sample image generated with basic error diffusion.

An artifact that is typically associated with error diffusion halftoning algorithms is known as “worms.” Worms are formed when the black or white output pixels appear to string together in an area that should be otherwise uniform. Worm artifacts can be clearly seen in the light and dark ends in the gray wedge along the top of Fig. 3. Other objectionable artifacts include the near-periodic patterns that are found at some of the intermediate gray levels. Many modifications to the basic error diffusion algorithm have been proposed to attempt to eliminate these artifacts and improve the overall quality.[3–12] These algorithms vary greatly in their complexity, as well as in the associated image quality.

One particular variation of the basic error diffusion algorithm that is relevant to the present work has been described by Sullivan *et al.*[13] This method will be referred to as visual error diffusion. The fundamental concept incorporated into this algorithm is that the error signal is computed relative to the tone value that would be observed by the human visual system, rather than the tone value for a single pixel. With this approach, a blur filter derived from the response of the human visual system is used to compute a visually perceived output value. The output level is chosen that gives the smallest error between the continuous-tone input value and the visually perceived output value. Likewise, the corresponding error signal that is propagated to the nearby image pixels is computed by taking the difference between the continuous-tone input value and the visually perceived output value rather than the output pixel value itself.

Figure 4 illustrates a flow diagram for the visual error diffusion method. The simple threshold in conventional error diffusion has been replaced by a selection criterion that is used to determine the output pixel value $b_{i,j}$. The selection is made by using a causal visual blur function, $v_{i,j}$, to blur the previously computed output pixel values together with each of the possible output levels for the current pixel to compute a set of visually perceived output values. For a binary output device there will be two possible output levels corresponding to a black or a white pixel. For multilevel output devices, a visually perceived output value is computed for each of the possible output levels. The output pixel value is chosen that gives the smallest difference between the continuous-tone input value and the visually perceived output value. The error signal is given by the difference associated with the chosen output level. As with the conventional error diffusion algorithm, this error is then weighted by a series of error weights $W_{i,j}$, and is added to the continuous-tone input values of nearby pixels which have yet to be processed.

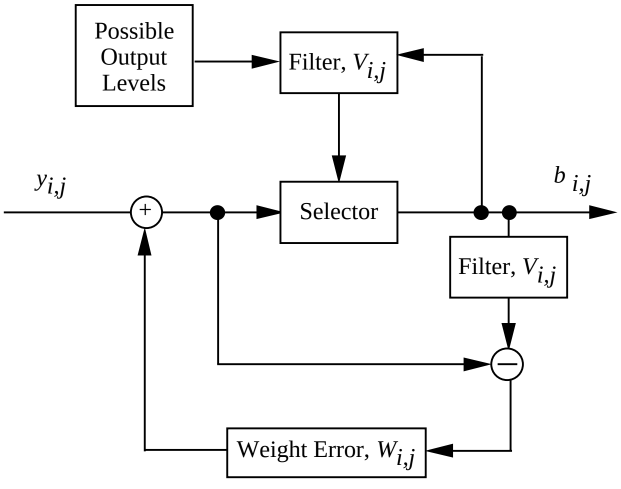

Figure 4. Visual error diffusion algorithm.

The causal visual blur function used in this method is computed from the frequency response of the human visual system. An example of a 4 × 7 causal visual blur function is shown in Fig. 5. The indicated array element is used to weight the possible output levels for the current pixel, and the remaining array elements are used to weight the nearby output values which have been previously computed. (This causal visual blur function is a normalized version of the one described by Sullivan *et al*.[14]) A similar 8 × 15 causal visual blur function is shown in Fig. 6. The size of the causal visual blur function array is related to parameters, such as viewing distance and sample spacing on the document, in the equations given by Sullivan. Figure 7 shows a sample image generated using this algorithm with the 8 × 15 causal visual blur function. Note that the appearance of the worm artifacts is substantially reduced relative to the conventional error diffusion image shown in Fig. 3. Also, it can be seen that the algorihm suppresses many of the near-periodic patterns at the intermediate gray levels. It is believed that the basic reason for the improvement in these artifacts can be traced to the fact that the visually percieved output values are computed based on a relatively large region of influence. Therefore, the algorithm will be less sensitive to the errors made in the neighboring pixels, and will therefore be less likely to get into a "ringing" type mode where a placing a black dot creates a large error which must be corrected by placing a white dot in the next pixel, and so on.

-0.009	-0.010	0.004	0.021	0.004	-0.010	-0.009
-0.010	-0.018	0.007	0.051	0.007	-0.018	-0.010
0.004	0.007	0.079	0.190	0.079	0.007	0.004
0.021	0.051	0.190	0.368			

current pixel

Figure 5. Example of a 4 × 7 causal visual blur filter used for visual error diffusion.

Although the visual error diffusion algorithm has the result of reducing the appearance of artifacts to a large degree without the introduction of undesirable noise, a side effect of this method is that artifacts are generated near edges, lines, and isolated pixels in the image. Although these artifacts do have a sharpening effect on the halftoned image, it is not necessarily a desirable effect because the degree of sharpening can not be controlled independent of the artifact reduction, and additionally the effect is anisotropic so that the apparent sharpening is not symmetric. As a result, different amounts of sharpness will be observed on an edge going from dark gray to light gray, as opposed to one going from light gray to dark grey. Additionally ghost pixels may be formed near high-contrast edges. These artifacts can be observed around the circles and the other edges in the image shown in Fig. 7.

The origin of these artifacts can be traced to the fact that Sullivan *et al.* did not blur the input image as well as the output image. The selection process, therefore, is comparing the visually perceived output image to the unblurred input image. For example, consider the case where the input image contains an isolated black pixel on a white background. The algorithm will place white pixels in most of the area corresponding to the white background, but when it comes time to make a selection for the black pixel it will compute visually perceived output values by convolving the output pixels in the surrounding area. Since most of these pixels will be white, the visually perceived output level for a white output value will be white, but the visually perceived output value for a black output pixel will be light gray rather than black. Because the light gray visually perceived output level will be closer to the desired black input level than the white visually perceived output level, the correct black output level will be chosen. However, the resulting error signal corresponding to the difference between the black input level and the light gray visually perceived output level will be quite large. This error will then be propagated to the surrounding continuous-tone input pixels. As a result, when the nearby white back-

-0.002	-0.002	-0.002	-0.002	-0.001	0.000	0.002	0.003	0.002	0.000	-0.001	-0.002	-0.002	-0.002	-0.002
-0.002	-0.003	-0.003	-0.003	-0.002	0.001	0.004	0.006	0.004	0.001	-0.002	-0.003	-0.003	-0.003	-0.002
-0.002	-0.003	-0.004	-0.005	-0.003	0.001	0.007	0.010	0.007	0.001	-0.003	-0.005	-0.004	-0.003	-0.002
-0.002	-0.003	-0.005	-0.005	-0.004	0.002	0.011	0.017	0.011	0.002	-0.004	-0.005	-0.005	-0.003	-0.002
-0.001	-0.002	-0.003	-0.004	-0.002	0.007	0.022	0.031	0.022	0.007	-0.002	-0.004	-0.003	-0.002	-0.001
0.000	0.001	0.001	0.002	0.007	0.020	0.043	0.057	0.043	0.020	0.007	0.002	0.001	0.001	0.000
0.002	0.004	0.007	0.011	0.022	0.043	0.076	0.096	0.076	0.043	0.022	0.011	0.007	0.004	0.002
0.003	0.005	0.010	0.017	0.031	0.057	0.096	0.118							

current pixel

Figure 6. Example of a 8 × 15 causal visual blur filter used for visual error diffusion.

ground pixels are processed, one or more of these pixels may be rendered as black output pixels. The artifacts can be even more severe on edges between uniform patches of different gray levels where anisotropic overshoots are formed.

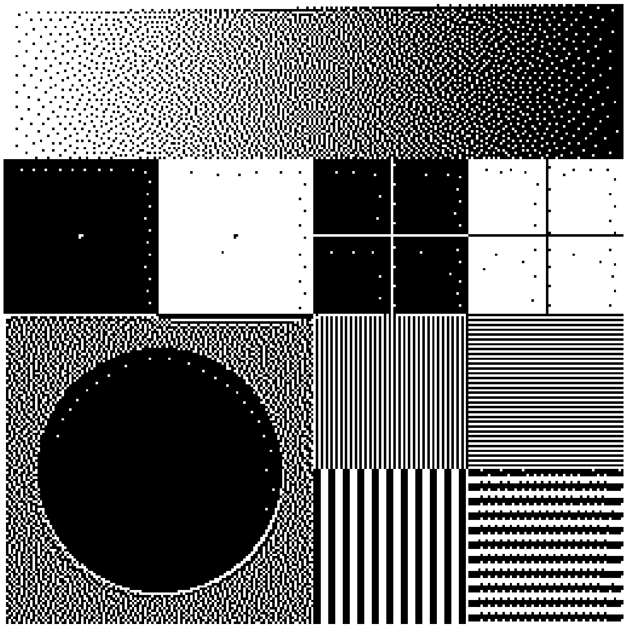

Figure 7. Sample image generated using visual error diffusion.

The new error diffusion algorithms that will be described here eliminate these artifacts while simultaneously maintaining the benefits associated with the visual error diffusion algorithm.

2 Visual Error Diffusion With Input Blur

As was noted above, the source of many of the artifacts that are associated with the visual error diffusion algorithm lies in the fact that the error is computed by comparing the visually perceived output value to the unblurred input value. This suggests that these artifacts could be reduced by applying the same causal visual blur filter to the continuous-tone input pixels that is applied to the halftoned output pixels. As a result, the visually perceived output pixel values will be compared to the visually perceived input pixel values rather than the input pixel values themselves during the output pixel selection step and the error calculation step as is shown schematically in Fig. 8. This method is identical to the method of Sullivan *et al.* shown in Fig. 4, with the exception that the causal visual blur filter $V_{i,j}$ is also applied to the continuous-tone input pixels to compute visually perceived input values. The selection of the output pixel value is done by using the same causal visual blur function to blur the previously computed output pixel values along with each of the possible output levels for the current pixel to compute the visually perceived output value. For a binary output device there will be two possible output levels corresponding to a black or a white pixel, but this method can be extended to multilevel output devices by considering more than two possible output levels. The output pixel value is chosen that gives the smallest error between the visually perceived continuous-tone input value and the visually perceived output value. The resulting error is then calculated for the current pixel by finding the difference between the visually perceived continuous-tone input value and the visually perceived output value for the selected output level. This error is then multiplied by a series of error weights, $W_{i,j}$, and is added to the visually blurred continuous-tone input values of nearby pixels which have yet to be processed.

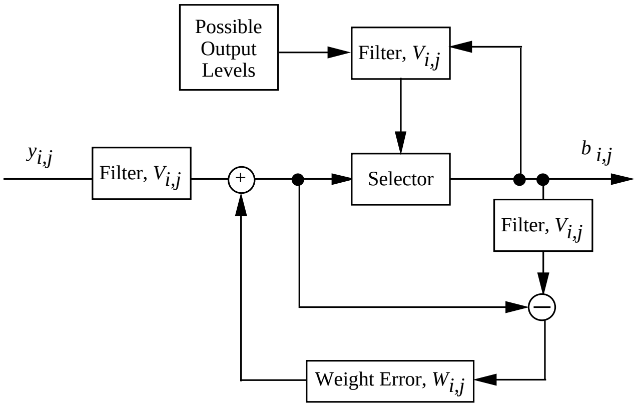

Figure 8. Visual error diffusion algorithm using input blur.

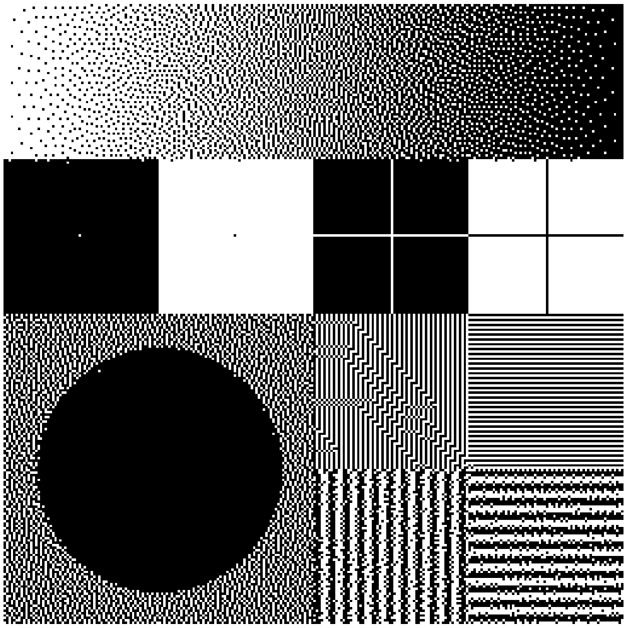

Figure 9. Sample image generated using visual error diffusion algorithm with input blur.

The result of visually blurring the continuous-tone input values as well as the output values is that the calculated errors will now represent the true visually perceived difference between the input image and the output image. This eliminates the anisotropic sharpening artifact as well as the ghost pixels associated with Sullivan's visual error diffusion algorithm. Figure 9 shows a sample image generated using this approach. The 8 × 15 causal visual blur function shown in Fig. 6 was used to visually blur both the input and output pixels. It can be seen that the worms found in the conventional error diffusion have been eliminated as they were with Sullivan's visual error diffusion, but that the edge artifacts associated with that algorithm have been eliminated.

Although the anisotropic sharpening effect caused by the Sullivan *et al.* visual error diffusion algorithm is undesirable in many respects, it does have the effect of increasing the overall image sharpness which can be a positive attribute in many cases. The use of this new method eliminates most of this sharpening effect. As a result, an image generated with the input visual blur will be preferable from an artifact standpoint, but may appear quite soft relative to an image generated with the original visual error diffusion technique. This can be compensated for by applying a sharpening filter to the continuous-tone input image prior to the application of the improved visual error diffusion algorithm, as shown in Fig. 10. This algorithm is identical to that shown in Fig. 8 with the exception that a sharpening step is applied to the continuous-tone input pixels before the application of the visual blur operation. One advantage of applying the sharpening as a separate step, rather than simply using Sullivan's algorithm, is that the amount of sharpening can be easily controlled by adjusting the sharpening coefficients, and additionally the sharpening can be made to be more isotropic. The sharpening filter can be implemented as a simple convolution, or using other known techniques such as un-sharp masking. A typical 3 × 3 sharpening filter that can be used for this purpose is shown in Fig. 11. Because both the sharpening step and the visual blurring of the input signal are sequential convolution operations that are applied to the continuous-tone input pixels, they can be combined into a single convolution operation where the convolution kernel is formed by the convolution of the sharpening filter with the visual blur filter.

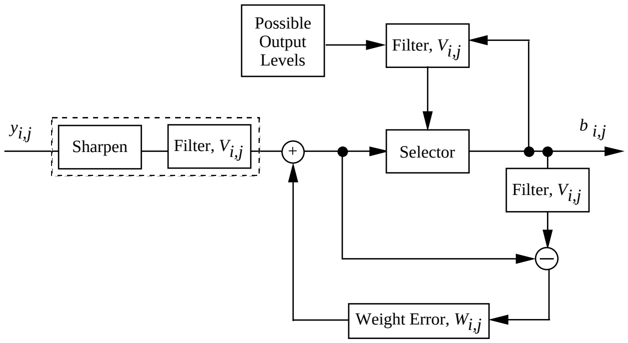

Figure 10. Visual error diffusion algorithm with input blur and pre-sharpening.

Figure 12 shows a sample image generated using the method of Fig. 10 with the sharpening kernel of Fig. 11. It can be seen that the improved algorithm introduces a desirable amount of sharpening and eliminates the majority of the ghost pixel and anisotropic sharpening artifacts associated with the visual error diffusion algorithm, while maintaining the benefits of the worm reduction. It should be noted that performing the convolution of the visual blur filter with the input values is significantly more costly from a computational point of view than the convolution with the output values. This is due to the fact that the output values will generally be binary, and, therefore, the covolution basically reduces to simply a series of add operations. This may be the main reason that the original Sullivan *et al.* algorithm did not include the input blur step. This drawback is addressed in the next section where it is shown that an adaptive algorithm can be used to produce superiour results, while minimizing the need to use an input blur filter.

-0.197	-0.373	-0.197
-0.373	3.28	-0.373
-0.197	-0.373	-0.197

Figure 11. Sample 3 × 3 sharpening filter.

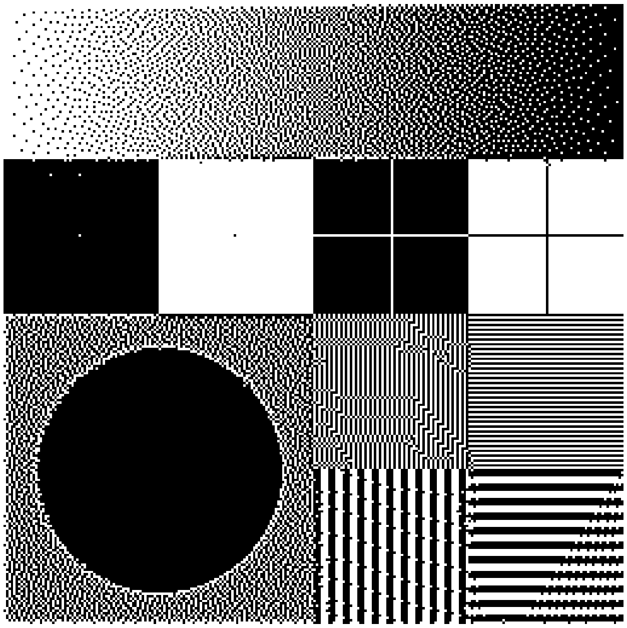

Figure 12. Sample image generated using method of Fig. 10.

3 Adaptive Visual Error Diffusion

While the improved algorithm described in the last section produces a higher quality output than the earlier methods, it does introduce undesirable noise into certain areas of the image that are high in contrast and spatial frequency. For example, consider the vertical and horizontal bars near the center right of the sample image of Fig. 12. This noise results from the fact that the visual blur filter makes it impossible for the algorithm to distinguish the high-frequency bars and a uniform gray patch. Ideally, it is desirable to combine the performance of the simple error diffusion algorithm of Fig. 1 in areas of the image that contain high spatial frequency content, with that of the improved error diffusion algorithm shown in Fig. 10 in areas of the image that are smoothly varying. A new error diffusion algorithm will now be described that uses a local activity detector to combine these two approaches in a locally adaptive fashion to take advantage of the best features of each approach.

This adaptive visual error diffusion algorithm is shown schematically in Fig. 13. To process an input pixel $y(i,j)$ located in row i, column j of the input image, an activity detector is first used to determine the nature of the image content in the surrounding pixel neighborhood. In general, the activity detector may take many forms, including a local range detector, local variance estimator, convolution filter, or edge detection operator such as a Sobel or Prewitt edge detection filter.[15] An activity detector using a local range detector would compute the activity signal as the difference between the maximum and minimum input pixel values of neighboring pixels in the vicinity of $y(i,j)$. The neighboring pixels may include the current pixel, adjacent pixels, and/or other nearby pixels. An activity detector using a local variance estimator would compute the activity signal as the statistical variance of the input pixel values for neighboring pixels in the vicinity of $y(i,j)$. An activity detector using a convolution filter would compute the activity signal as the convolution of an edge detection convolution filter and the input pixel values of neighboring pixels in the vicinity of $y(i,j)$. An activity detector using an edge detection operator would compute the activity signal as the sum of convolutions of the input pixels of neighboring pixels in the vicinity of $y(i,j)$ with the edge detection filters. Of these options, it has been found that the local range activity detector produces good results and is relatively efficient to compute. This form of the activity detector will be used for the examples given in the remainder of this report.

The value of the activity signal is used as an index to an activity function that computes a weighting vector $Q = \{q_0, q_1, ..., q_k, ..., q_{N\text{-}1}\}$ that is used to specify the amount of weight to be applied to each of a set of N different error diffusion algorithms. Typically, the activity function would be implemented as a lookup table that is indexed by the activity signal. In its simplest form, the activity function can be used to switch between two different algorithms. An example of such an activity function is shown in Fig. 14, where S is the value of the activity signal. In this particular example, the value of the activity signal ranges from 0 to 255. For values of the activity signal between 0 and 10, $Q = \{1,0\}$, and therefore the first algorithm will be used. Likewise, for values of the activity signal between 10 and 255, $Q = \{0,1\}$, so that the second algorithm will be used. In this way, the processing is locally switched between the two algorithms based on the value of the activity signal. This

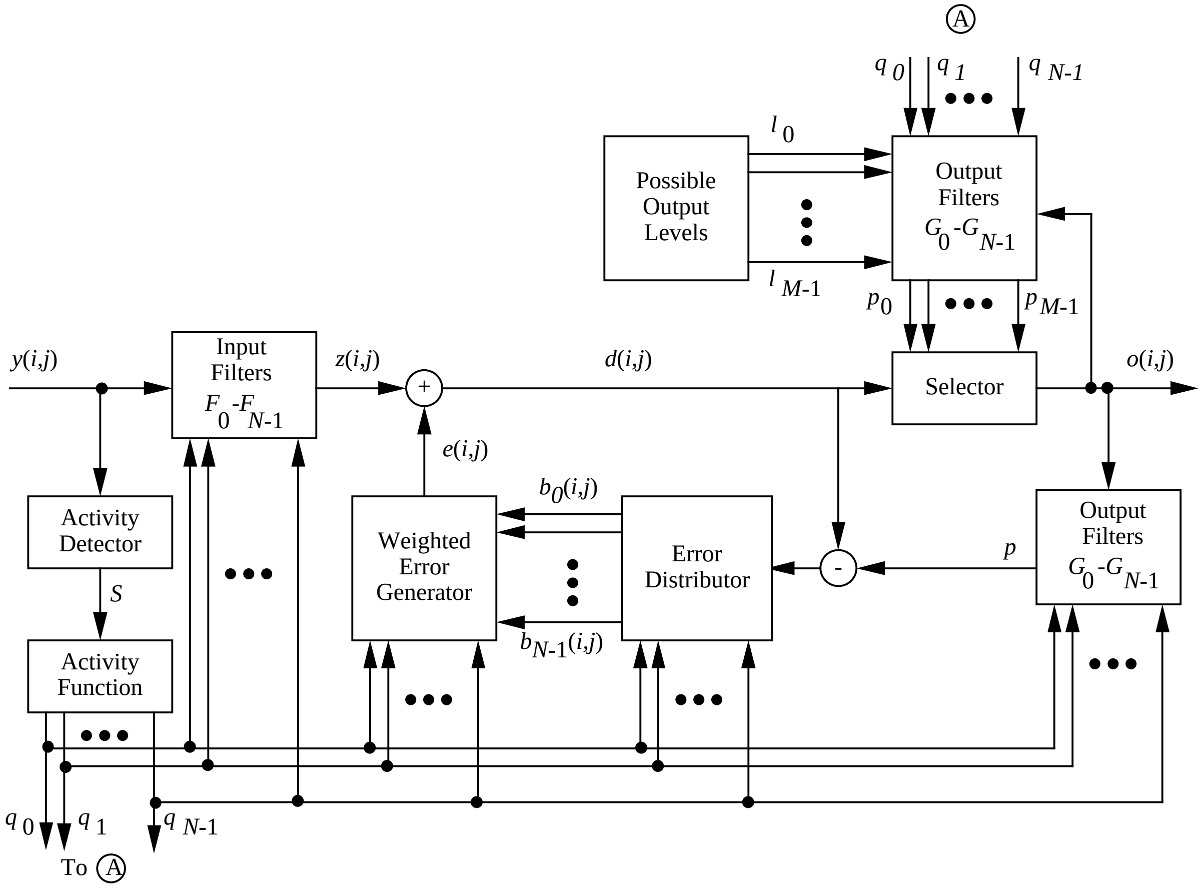

Figure 13. Adaptive visual error diffusion algorithm.

concept can be generalized so that a different algorithm is selected for each of N types of local scene content in the input image as discriminated by the activity detector. Additionally, the results can be blended in regions of the image where it is unclear exactly what type of scene content is present. An example of an activity function that will accomplish this is shown in Fig. 15.

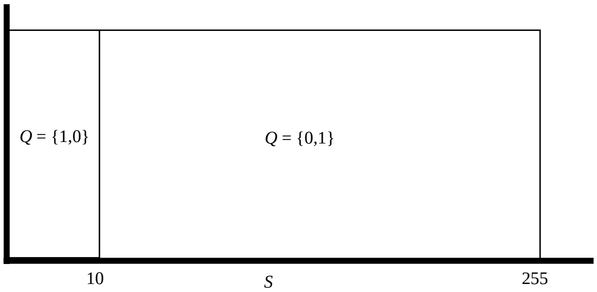

Figure 14. Example activity function to switch between two error diffusion algorithms.

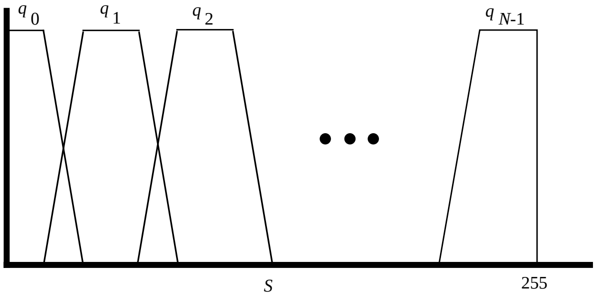

Figure 15. Example activity function to blend N different error diffusion algorithms.

The input image is also processed through an input filter processor, which digitally filters the input image in the vicinity of $y(i,j)$ using a bank of N separate filters $F_0(i,j)$ through $F_{N-1}(i,j)$ and combines the results. The N filtered input images are then weighted using the corresponding coefficients of the weighting vector Q and are added together to produce a weighted filtered input signal $z(i,j)$,

$$z(i,j) = \sum_{k=0}^{N-1} q_k \{y(i,j) * F_k(i,j)\}, \qquad (1)$$

where k is the filter number, and $*$ is the convolution operator. It should be noted that the filtered input values only need to be computed for cases where the weights q_k are non-zero.

For uniform regions of the input image, it is desirable that the average value of $z(i,j)$ be equal to the average value of $y(i,j)$. One way of satisfying this mean preserving relationship is to use filters $F_0(i,j)$ through $F_{N-1}(i,j)$ that are normalized (i.e., that satisfy $\Sigma_i\Sigma_j F_k(i,j) = 1$), and impose the constraint that Q satisfies $\Sigma q_k = 1$.

After the weighted filtered input signal $z(i,j)$ is computed, the errors made by processing previous pixels are weighted by the error weights $W(i,j)$ shown in Fig. 2 and by the weighting vector Q using a weighted error generator. During this step, the error values stored in N separate error buffers $b_0(i,j)$ through $b_{N-1}(i,j)$, corresponding to the N types of local scene content, are individually weighted by the spatial diffusion function $W(i,j)$, then the resulting values are weighted by the weighting vector Q, and added together to produce an error signal $e(i,j)$

$$e(i,j) = \sum_{k=0}^{N-1} q_k \{W(i,j) * b_k(i,j)\}, \qquad (2)$$

where a convolution notation has been used to represent the operation of weighting the error buffer with the spatial diffusion function. In this way, the errors generated in areas of the image that contain one type of scene content will not propagate to an area of the image that contains a different type of scene content. This type of separate error buffering is desirable to avoid artifacts that can be generated when the image contains a sharp boundary between areas of different scene content.

The error signal $e(i,j)$ is then added to the weighted input signal $z(i,j)$ to compute the desired signal $d(i,j)$.

$$d(i, j) = z(i, j) + e(i, j) \qquad (3)$$

An output filter processor computes a set of M weighted filtered output values p_0 through $p_{M\text{-}1}$. In this step, the previously computed output values $o(i,j)$ are convolved together with each of the M possible output levels for the current pixel (l_0 through l_{M-1}) using the bank of N filters $G_0(i,j)$ through $G_{N\text{-}1}(i,j)$ to compute filtered output values. The filtered output values only need to be computed for cases where the weights q_k are non-zero. For each of the M possible output levels, the filtered output values corresponding to the N filters are weighted by the weighting vector Q, and added together to generate a set of weighted filtered output values p_0 through $p_{M\text{-}1}$,

$$P_m = \sum_{k=0}^{N-1} q_k \{o_m(i,j) * G_k(i,j)\}, \qquad (4)$$

where $o_m(i,j)$ is the output image that would be formed if the output level l_m were used for the current pixel.

After the weighted filtered output values have been computed, the selector shown in Fig. 13 chooses the output levels (l_0 through $l_{M\text{-}1}$) which minimizes the difference between the desired signal $d(i,j)$ and the weighted filtered output values (p_0 through $p_{M\text{-}1}$). This step is intended to determine which of the possible output levels results in the smallest effective error between the actual output signal, and the desired output signal. The weighted filtered output value P_i corresponding to the selected output value will be needed to compute the error signal which will be distributed to future pixels. For clarity of the figure, a second output filter processor is shown which digitally filters the final output signal to compute a weighted filtered output value p. However, it should be noted that p is simply equal to the weighted filtered output value P_i corresponding to the selected output value. Since this result has already been determined, it should be unnecessary to recompute this value.

The error signal is generated by computing the difference between the weighted filtered output value and the desired signal $d(i,j)$. The error signal is then distributed into N error buffers b_0 through $b_{N\text{-}1}$ by an error distributor that weights the error signal by the weighting vector Q,

$$b_k(i, j) = q_k\{d(i, j) - p\}. \qquad (5)$$

The values stored in the N error buffers b_0 through $b_{N\text{-}1}$ are then processed by the weighted error generator and added to the weighted filtered input values of pixels that have not yet been processed, as described earlier.

As was the case for the visual error diffusion algorithm using input blur that was discussed in the last section, it may be desirable to apply a sharpening pre-filter to the input image in order to obtain a desirable level of edge sharpness. If the sharpening is implemented in the form of a convolution, then it is possible to incorporate the sharpening pre-filter directly into the input filtering operation by convolving the sharpening pre-filter with the filters $F_0(i,j)$ through $F_{N\text{-}1}(i,j)$ to arrive at a new set of input filters. It may be desirable to only apply the sharpening filter to portions of the image that contain large amounts of image activity. This can be accomplished by only incorporating the sharpening pre-filter into a subset of the filters $F_0(i,j)$ through $F_{N\text{-}1}(i,j)$ that correspond to large values of the activity signal.

A sample image generated using this adaptive visual error diffusion method is shown in Fig. 16. The activity function shown in Fig. 14 was used to switch between image regions having high and low activity signal values. The activity detector used in this example consisted of a local range detector. The set of pixels used in the local range detection operation was a 5 × 5 pixel neighborhood centered on the current pixel. In regions of low activity, the input filter $F_0(i,j)$ used was a delta function $\delta(i,j)$, where $\delta((i,j)$ is defined as 1 for $i = j = 0$ and 0 otherwise. As a result, it can be seen that the input filter operation reverts to a null operation and $z(i,j)$ is equal to $y(i,j)$. The output filter used for low activity regions $G_0(i,j)$ was the 8 × 15 causal visual filter $V(i,j)$ shown in Fig. 6. In can be seen that this pair of input and output filters simply corresponds to the conventional visual error diffusion algorithm shown in Fig. 4. In this case, it was found that there was no significant advantage to including the output visual blur filter since the low activity regions of the image will generally not have a significant amount of high spatial frequency content. In the high activity regions, the input filter $F_1(i,j)$ used was the sharpening filter shown in Fig. 11. The output filter $\mathrm{G}_1(i,j)$ used was the delta function $\delta(i,j)$. It can be seen that this is equivalent to the conventional error diffusion algorithm of Fig. 1 where a pre-sharpening step is applied.

From examination of Fig. 16, it can be seen that the image incorporates all of the benefits of the visual error diffusion algorithm, without any of the undesirable artifacts. The ghost pixels have been effectively eliminated, the anisotropic edge sharpening has been replaced by a well-controlled sharpening effect, and the noise previously introduced in regions of high spatial frequency content has been eliminated. An additional advantage of this technique is that the algorithm can potentially be computationally faster due to the fact that the time-consuming visual blur function only needs to be applied in flat image regions. Another example image generated using this method is shown in Fig. 17.

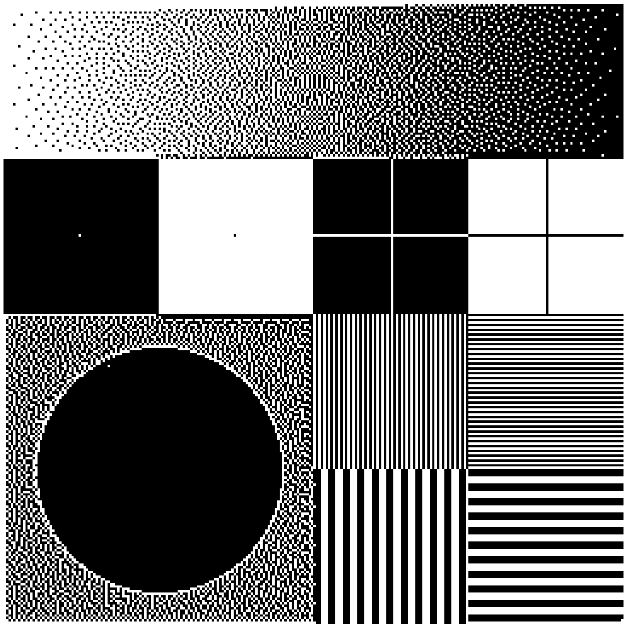

Figure 16. Sample image generated using adaptive visual error diffusion algorithm.

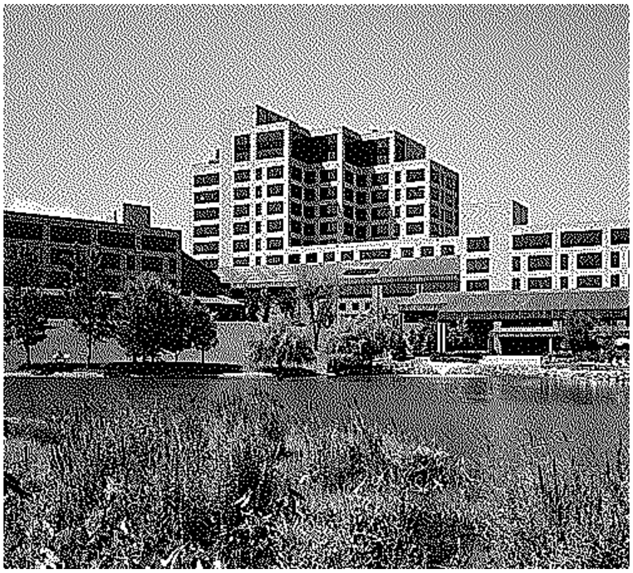

Figure 17. Sample image generated using adaptive visual error diffusion algorithm.

5 Conclusions

Improved error diffusion algorithms have been described that produce digitally halftoned images having fewer artifacts. In a first variation, a causal visual blur function is applied to both the input and output images during the process of selecting the output value for the current pixel. The error signal that is propagated to nearby unprocessed continuous-tone input pixels is determined by computing the difference between the visually perceived input value and the visually perceived output value. This approach has the advantage that it minimizes the appearance of worm artifacts in the output image, while eliminating the edge artifacts associated with a previous visual error diffusion algorithm. However, it was shown that undesirable noise characteristics result in image regions having high spatial frequency content.

A second variation of the algorithm was described that uses a local image activity detector to adaptively modify the input and output blur filters. This allows the error diffusion algorithm to be optimized for different types of image content. The adaptive visual error diffusion approach incorporates all of the advantages of the other visual error diffusion algorithms, while simultaneously eliminating the objectionable artifacts. Example images were given to compare the various techniques.

Acknowledgment

The authors would like to acknowledge the contributions of Jim Sullivan and Gene Pios who were instrumental in the development of the original visual error diffusion algorithm upon which the current work was based.

References

1. R. Ulichney, *Digital Halftoning*, MIT Press, Cambridge (1990).
2. R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial greyscale," *Proc. SID* **17**(2), 75-77 (1976).
3. J. F. Jarvis, C. N. Judice and W. H. Ninke, "A survey of techniques for the display of continuous tone pictures on bilevel displays," *Comput. Graphics Image Process.* **5**, 13-40 (1976).
4. C. Billotet-Hoffman and O. Bryngdahl, "On the error diffusion technique for electronic halftoning," *Proc. SID* **24**(3), 253-258 (1983).
5. G. Goertzel and G. R. Thompson, "System for reproducing multi-level digital images on a bi-level printer of fixed dot size," U. S. Patent 4,654,721 (1987).
6. R. A. Ulichney, "Dithering with blue noise," *Proc. IEEE* **76**(1), 56-79 (1988).
7. R. Eschbach, "Pulse density modulation on rastered media: Combining pulse-density modulation and error diffusion," *J. Opt. Soc. Am. A* **7**(4), 708-716 (1990).
8. R. L. Miller and C. M. Smith, "Image processor with error diffusion modulated threshold matrix," U. S. Patent 5,150,429 (1992).
9. R. Levien, "Output dependent feedback in error diffusion halftoning," in *Proc. IS&T 46th Annual Conference*, 115-118 (1993); ***see Recent Progress in Digital Halftoning, Vol. I, pg. 106.***
10. R. Eschbach, "Reduction of artifacts in error diffusion by means of input-dependent weights," *J. Electron. Imaging* **2**(4), 352-358 (1993).
11. T. Zeggel and O. Bryndahl, "Halftoning with error diffusion on an image-adaptive raster," *J. Electron. Imaging* **3**(3), 288-294 (1994).
12. K-M. Wang, S-W. Kang and C-W. Kim, "A modified error diffusion scheme based on the human visual system," in *Proc. IS&T 49th Annual Conference*, 341-345 (1996); ***see pg. 30, this publication.***
13. J. Sullivan, R. Miller and G. Pios, "Image halftoning using a visual model in error diffusion," *J. Opt. Soc. Am. A* **10**(8), 1714-1724 (1993).
14. J. Sullivan, "Digital halftoning with error diffusion," U. S. Patent 5,051,844 (1991).
15. For example see: W. K. Pratt, *Digital Image Processing*, John Wiley & Sons, New York, pp. 491-556 (1991).

❋ Previously published in the *Journal of Electronic Imaging*, **7**(3), 655–663 (1998).