

Error diffusion without contouring effect

Wei-Yu Han
Ja-Chen Lin

National Chiao Tung University
Department of Computer and Information Science
Hsinchu, Taiwan 30050

Abstract. A modified error-diffusion algorithm that eliminates the false-texture contour phenomenon that usually appears in the halftone images is presented. The main idea behind our modification is to introduce a local filter that can smooth the false-texture contours of the low variation regions in the images generated by the standard error-diffusion algorithm. Experimental results are included. © 1997 SPIE and IS&T.

1 Introduction

Due to physical limitations, some display or recording media, such as ink printers, can represent only binary data. Therefore, many techniques for converting continuous-tone images to halftone images (a type of binary image that preserves the perception of gray and image detail of the original gray level image) has long been investigated. These techniques can be categorized roughly into two classes, namely, (1) thresholding (including the ordered-dither technique) and (2) error diffusion. A more detailed survey can be found in Refs. 1, 2, and 3.

Recently, several modifications were proposed to improve the reproduced binary image quality or extend the capability of the standard error-diffusion algorithm introduced by Floyd and Steinberg.⁴ Eschbach and Knox⁵ used image information to modulate the threshold of the standard error-diffusion algorithm⁴ so that edge enhancement effect can be obtained. Sullivan *et al.*⁶ combined a technique of visual modeling with the error feedback of the conventional error diffusion to suppress the unwanted textures and improve the sharpness of halftone images. Levien⁷ showed that applying an output dependent function to the threshold of the standard error-diffusion algorithm would reduce the worm-like patterns. Fan⁸ incorporated ordered dither into error diffusion to overcome the drawbacks of ordered dither, while preserving the goodness of both methods. Eschbach⁹ described a multiple weight-matrix error-diffusion algorithm for reducing the artifacts that can be found in the traditional single weight-matrix error-diffusion algorithm. Xie and Rodriguez¹⁰ proposed a two-pass modified error-diffusion algorithm in which, according to their description, the first pass uses a random threshold to minimize the artifacts in flat tonal areas and the second

pass attempts to preserve the bandwidth of the continuous-tone image.

Almost every existing algorithm has false-texture contours. More precisely, although a few reported algorithms, such as Eschbach's,⁹ tried to reduce the artifacts, we found that false-texture contours were still visible in their experimental results. (As for the elegant method proposed by Xie and Rodriguez,¹⁰ the processing time is not short. Besides, as seen later in our experiments, if the magnitude of the so-called threshold randomization is not chosen properly, the false-texture contours might still be visible.) In this paper, we focus completely on the elimination of the false-texture contours generated by the standard error-diffusion algorithm. (Hereafter, whenever we mention "the standard error-diffusion algorithm" or "the standard ED," we mean the one introduced in Ref. 4. It is called "standard" because many reported methods are basically derived from it. Moreover, these advanced methods still have the false-texture contour problem that the standard ED has. Therefore, we started our modification from the standard ED instead of the other error-diffusion algorithms because we do not want to make our design too complicated. Note, however, that after the design, we compare our method with not only the standard ED but also many other kinds of error-diffusion algorithms and find that our new method works the best.)

2 Review of Error Diffusion (ED)

ED is a popular technique to render continuous-tone images using only two gray values. Note that all images mentioned in this paper are digital images, and each pixel located at address (i, j) has gray value $g(i, j)$ and the gray values are scaled from $[0, 255]$ to the range $0 \leq g(i, j) \leq 1$. In an ED algorithm, each pixel of the input image is compared with a fixed threshold value, if the current pixel's gray value is greater than the threshold value, the output is assigned as 1, otherwise, a 0 is placed. The error between the input and output of the current pixel is propagated to the unprocessed neighboring pixels (the propagation ratios among these neighbours are determined by a weight matrix). Theoretically, if the weights in the weight matrix have a sum of 1, the average gray value of the output image will be the same as that of the input image.¹¹ Fig. 1 depicts the block diagram and the associated parameters of a typical ED. Fig. 2 depicts a set of commonly used weight matrices.^{1,4,12}

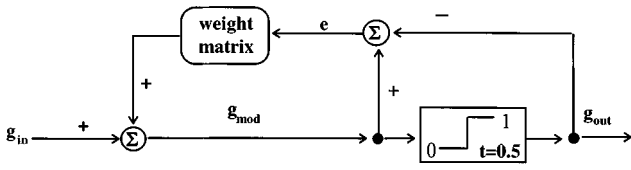


Fig. 1 Typical ED algorithm.

As mentioned, before the comparison with the threshold value, every pixel's gray value (except that of the first pixel) was modified by means of the error feedback of the previously processed neighboring pixels; the relationship of the input gray value $g_{in}(i,j)$, modified gray value $g_{mod}(i,j)$, error feedback $e(i,j)$, and output binary value $g_{out}(i,j)$, can be expressed as

$$g_{mod}(i,j) = g_{in}(i,j) + \sum_{k,m \in \Omega} W_{k,m}(i,j)e(i-k,j-m), \quad (1)$$

$$g_{out}(i,j) = \begin{cases} 1 & g_{mod}(i,j) > t \\ 0 & \text{otherwise} \end{cases}, \quad (2)$$

$$e(i,j) = g_{mod}(i,j) - g_{out}(i,j). \quad (3)$$

Here $w_{k,m}$ is the weight coefficient for the error e incurred at location $(i-k,j-m)$, Ω is the neighborhood of pixel (i,j) , which is defined by the nonzero positions of the weight matrix (see Fig. 2), and t is the threshold value usually taken to be the constant 0.5.

3 Observations of the g_{mod} Image Generated by the Standard ED

We first discuss the causes of the false-texture contours. It is well known that all the halftoning algorithms imitate gray values by varying dot patterns (some distribution of the black and white dots, usually, a dot pattern has two observable properties, namely, intensity and structure). The dot patterns generated by the standard ED can reproduce intensity very well; however, there exist in the standard ED some dot patterns that are quite distinct visually, although they represent the adjacent gray value ranges. This phenomenon of the standard ED is shown in Fig. 3. In Fig. 3, for a slowly varied area of the image, when the gray values gradually change from 0.41 to 0.42, then to 0.43, . . . , then to 0.62 [see Fig. 3(a)], two quite distinct dot patterns appear on the two sides of the vertical boundary across which the gray values change from 0.50 on one side to 0.51 on the other side [see Fig. 3(b)]. The boundary just mentioned then becomes a false-texture contour. This dilemma will

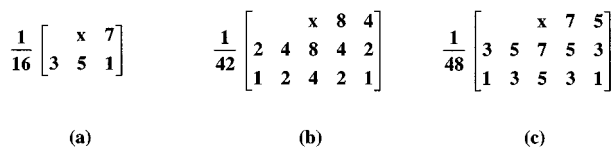


Fig. 2 Some commonly used weight matrices ["x" is the location of the current pixel (i,j)]: (a) Floyd and Steinberg,⁴ (b) Stucki,¹² and (c) Jarvis *et al.*¹

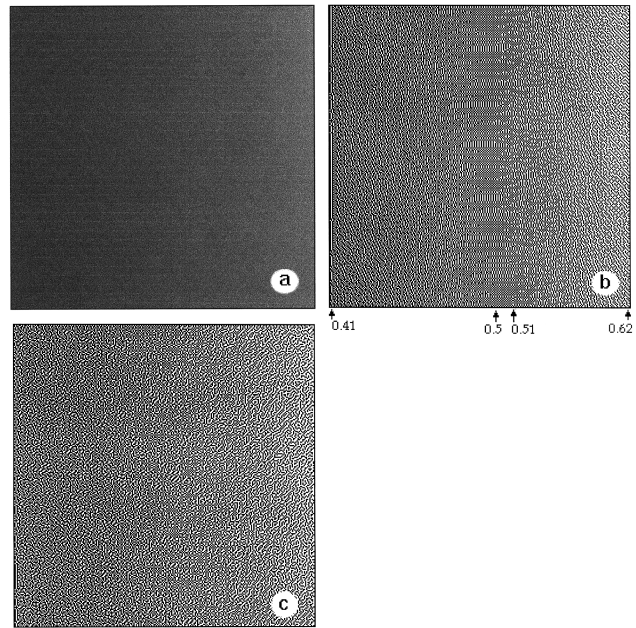


Fig. 3 Slowly varied image (gray value range from 0.41 to 0.62): (a) original gray level image, (b) output image generated by the standard ED, and (c) output image generated by the proposed method.

not be a problem for the proposed method [see Fig. 3(c)]. A complete list of the dot patterns used for each gray-value range 0.0 to 0.1, 0.1 to 0.2, . . . , 0.9 to 1.0 are provided in Fig. 4. We can see that many false-texture contours, i.e., sudden changes of dot patterns, appear for the standard ED, but not for ours. This phenomenon is more obvious if we inspect a wider gray-value range, e.g., 0.4 to 0.7, as shown in Fig. 5, which combines three pieces 0.4 to 0.5, 0.5 to 0.6, and 0.6 to 0.7 in Fig. 4(a) [Fig. 4(b)] to get Fig. 5(a) [Fig. 5(b)].

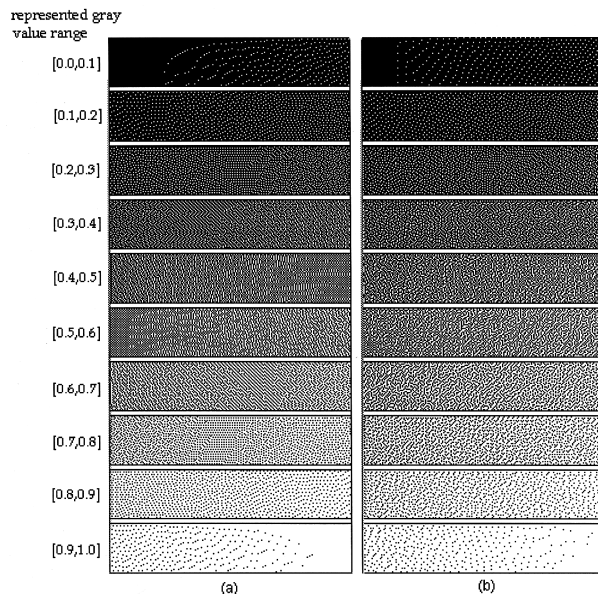


Fig. 4 Dot patterns for gray values range from 0.0 (completely black) to 1.0 (completely white): (a) standard ED and (b) our ED.

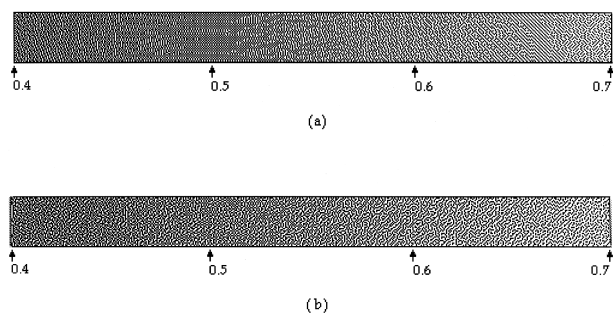


Fig. 5 Dot patterns for gray values range from 0.4 to 0.7: (a) standard ED and (b) our ED.

To clarify how we obtained the idea needed to design the method, we discuss in this paragraph the relationship between the modified and the output images defined by Eqs. (1) and (2), respectively. Fig. 6(a) shows an image whose intensity changes slowly from 0 to 1 in the antidiagonal direction. When we applied the standard ED to Fig. 6(a), the g_{mod} image [defined by Eq. (1)] would be Fig. 6(b) and the output image [defined by Eq. (2)] would be Fig. 6(c). From Figs. 6(b) and 6(c), we can see that a very close

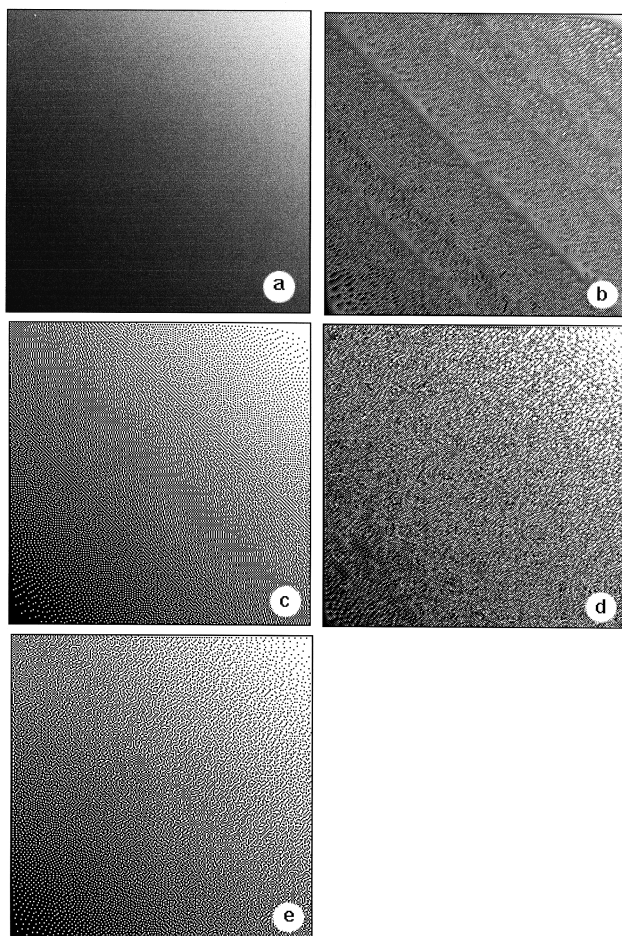
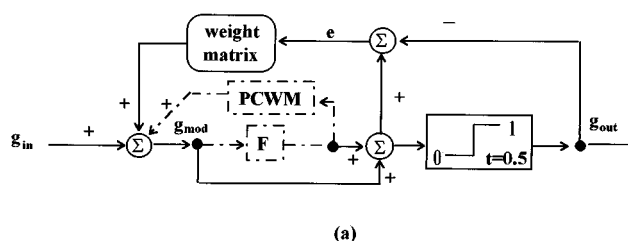


Fig. 6 (a) Original gray level image, (b) the g_{mod} image of the standard ED, (c) output image of the standard ED, (d) the g_{mod} image of our method, and (e) our output image.



$$PCWM = \frac{1}{30} \begin{bmatrix} & & & x & -1 & -5 & -3 \\ -1 & -3 & 0 & 0 & 0 & -3 & -1 \\ 0 & -1 & -3 & -5 & -3 & -1 & 0 \end{bmatrix}$$

(b)

Fig. 7 (a) Proposed modification of the standard ED and (b) perturbation compensation weight matrix (PCWM).

relation exists between the positions of the bands (each band is an area composed of similar texture) in the g_{mod} image and the positions of the dot patterns in the output image. We therefore know that the dot patterns in the output image can be redesigned to get a better outlook by means of changing the texture structure of the bands in the modified image g_{mod} .

4 Proposed Modification of ED

As noted in the last sentence of Section 3, in this section we try to alleviate the banding effect of g_{mod} so that g_{out} will include no false-texture contours. Our procedure is to apply a filter to prohibit or destroy the formation of the bands in the g_{mod} image; as a result, the dot patterns in the g_{out} image will not change abruptly and thus alleviate the phenomenon of false-texture contours. We tried to apply to the g_{mod} image some commonly seen smoothing operators, e.g., the mean filter or the median filter, and found that the results were not satisfactory. We therefore use a more advanced filter to correct the g_{mod} image. The details are illustrated in the following. The block diagram of the proposed modification of the standard ED is provided in Fig. 7(a). Note that the solid lines there represent the standard ED and the dotted lines are the proposed modification. Also note that F is a perturbing operator whose value at pixel (i, j) is defined by the product of three terms; more precisely,

$$[F(g_{mod})]_{(i,j)} = P \cdot Z \cdot g_{mod}(i, j), \quad (4a)$$

$$Z = (1 - \exp\{-[g_{mod}(i, j) - \mu]^2 / \text{var}\}), \quad (4b)$$

$$P = \begin{cases} 1 & g_{mod}(i, j) > \mu \\ -1 & \text{otherwise} \end{cases}, \quad (4c)$$

where μ is the local mean value and var is the local variance of the g_{mod} image. Both values μ and var are calculated using a 3×3 neighborhood centered at the current pixel $g_{mod}(i, j)$ being processed. The function of the dotted block PCWM in Fig. 7(a) is to correct the gray-value shift induced by the operator F . The details of the PCWM are discussed at the end of this section. The matrix PCWM used throughout the paper is shown in Fig. 7(b).

The major idea of designing the proposed filter F is to reduce the chance of gathering some fixed texture structures in an area (i.e., to reduce the banding effect in the g_{mod} image) by means of perturbing the gray values of the g_{mod} image. In the following, we explain the function of F and show that the filter F can meet the design goal. Eq. (4a) is a combination of g_{mod} and $(P \cdot Z)$, and $(P \cdot Z)$ is a perturbing term. Eq. (4b) shows that Z is a complement of Gauss function, and the response value of Z depends on the distance between $g_{\text{mod}}(i, j)$ and the local mean μ ; moreover, the value of var controls the shape of Z (the lower the var value, the sharper the shape), which means that, with same local mean μ and $g_{\text{mod}}(i, j)$, the response value of Z is stronger in the area with low variation than in the high variation area. An example to illustrate this phenomenon is given next.

Example 1. Let $A1$ and $A2$ be two 3×3 areas (of the g_{mod} image) with the same local mean μ and the same gray value $g_{\text{mod}}(i, j)$ for the center pixel (the underbared pixel shown below). If $A1$ has higher local variance, then the response value of Z will be less than the one of $A2$. For example, if $A1$ and $A2$ are

$$\begin{array}{ccc}
 0.2 & 0.4 & 0.4 \\
 A1 = 0.4 & \underline{0.5} & 0.6 \\
 0.3 & 0.5 & 0.3
 \end{array}
 \quad
 \begin{array}{ccc}
 0.35 & 0.35 & 0.5 \\
 A2 = 0.35 & \underline{0.5} & 0.35 \\
 0.5 & 0.35 & 0.35
 \end{array}$$

then, for both $A1$ and $A2$, we have $g_{\text{mod}}(i, j) = 0.5$, $\mu = 0.4$, and $[g_{\text{mod}}(i, j) - \mu]^2 = (0.1)^2 = 0.01$. The only difference is var , which is 0.013 for $A1$, and 0.005 for $A2$. As a result, Eq. (4b) implies that $[Z_{A1} \cdot g_{\text{mod}}(i, j)] = 0.263$ and $[Z_{A2} \cdot g_{\text{mod}}(i, j)] = 0.432$.

Eq. (4c) determines the sign of the perturbation $[Z \cdot g_{\text{mod}}(i, j)]$. More precisely, P ensures that the perturbation will make the $g_{\text{mod}}(i, j)$ away from the local mean μ , and hence, amplifies the local variance. Example 2 illustrates this.

Example 2. Let $A1$ and $A2$ be two 3×3 areas that are identical everywhere in gray values except at the center pixel $g_{\text{mod}}(i, j)$ (the underbared pixel shown below). Note that after applying the operator F , the sign of P makes the local variances of $A1$ and $A2$ both increased.

Originally,

$$\begin{array}{ccc}
 0.2 & 0.4 & 0.4 \\
 A1 = 0.4 & \underline{0.5} & 0.6 \\
 0.3 & 0.5 & 0.3 \\
 \mu_{A1} = 0.4, \\
 \text{var}_{A1} = 0.0133, \\
 [P \cdot Z \cdot g_{\text{mod}}(i, j)]_{A1} = +0.263,
 \end{array}$$

$$\begin{array}{ccc}
 0.2 & 0.4 & 0.4 \\
 A2 = 0.4 & \underline{0.25} & 0.6 \\
 0.3 & 0.5 & 0.3 \\
 \mu_{A2} = 0.372, \\
 \text{var}_{A2} = 0.0139, \\
 [P \cdot Z \cdot g_{\text{mod}}(i, j)]_{A2} = -0.164.
 \end{array}$$

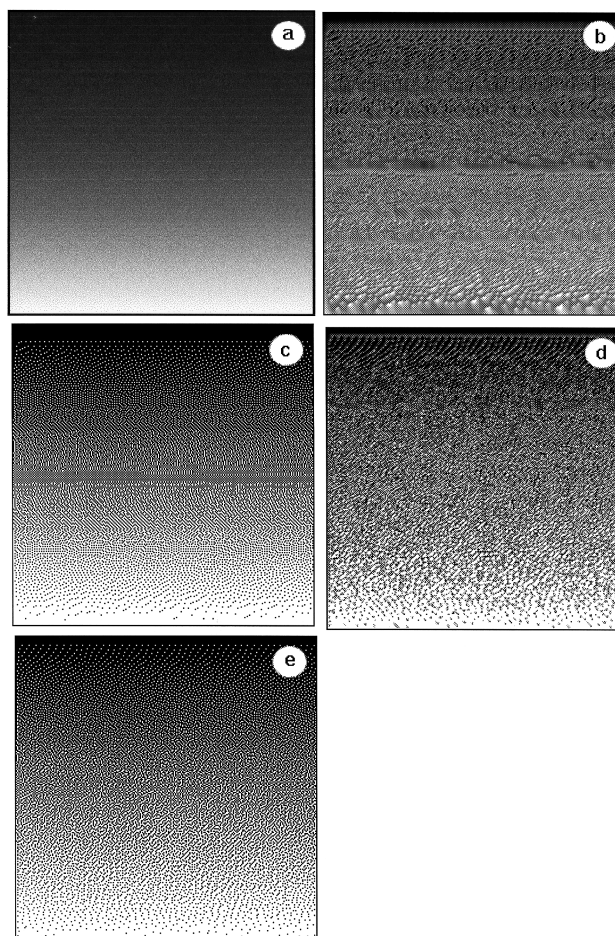


Fig. 8 (a) Original gray level image, (b) the g_{mod} image of the standard ED, (c) output image of the standard ED, (d) the g_{mod} image of ours, and (e) the output image of our method.

After applying the operator F ,

$$\begin{array}{ccc}
 0.2 & 0.4 & 0.4 \\
 A1 = 0.4 & \underline{0.763} & 0.6 \\
 0.3 & 0.5 & 0.3 \\
 \mu_{A1} = 0.429, \\
 \text{var}_{A1} = 0.026,
 \end{array}
 \quad
 \begin{array}{ccc}
 0.2 & 0.4 & 0.4 \\
 A2 = 0.4 & \underline{0.086} & 0.6 \\
 0.3 & 0.5 & 0.3 \\
 \mu_{A2} = 0.354, \\
 \text{var}_{A2} = 0.021.
 \end{array}$$

In general, just like the two preceding examples, the operator F always enlarges the local variance. The larger the local variance, the smaller the chance of gathering some fixed texture patterns in an area (and gathering some other fixed patterns in the other area) that visually causes the banding effect occurs (because two adjacent areas have two quite distinct texture patterns, and both areas are quite large). Fig. 6(d) shows the result after applying the proposed filter, Eqs. (4) to Fig. 6(b). It is obvious that the banding effect is indeed alleviated significantly. The final binarized image is presented in Fig. 6(e); and as we expected, almost no false-texture contours exist and the output is better than in Fig. 6(c). One more example is given in Fig. 8 in which the intensity of the original image changes slowly from 0 to 1 in the vertical direction. The reason that

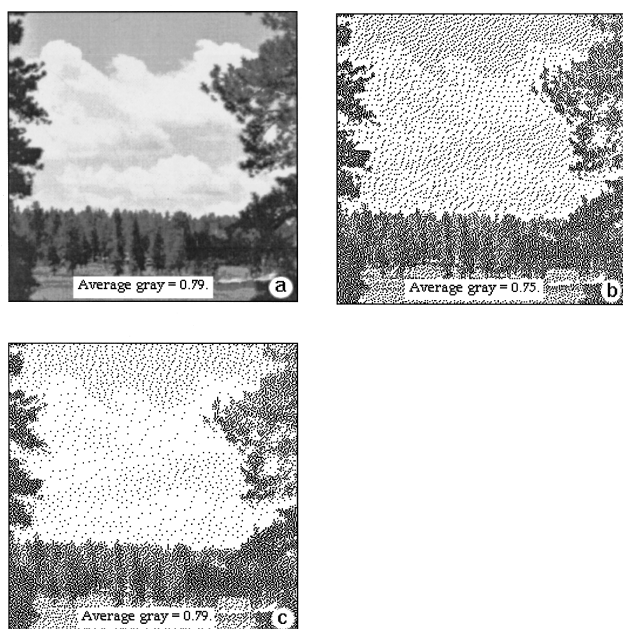


Fig. 9 Perception of gray (with or without PCWM): (a) original continuous-tone image, (b) halftone image generated by our method (when the PCWM is not applied), and (c) halftone image generated by our method (when the PCWM is applied).

the structure of the g_{mod} image is random is probably because that the perturbation (which always enlarge the local variance) destroys the regularity (position and gray value) of the original g_{mod} image generated by the standard ED.

We explain below why the PCWM is needed; Example 3 is also provided to illustrate the need. As mentioned, after applying the filter F , the gray value of each pixel (in the g_{mod} image) had been changed. To make our method have the benefit (reproducing the intensity well) of the standard ED, we use the PCWM to compensate each pixel's gray-value shift; as a result, the average of gray of the original continuous-tone digital image can be preserved. More precisely, the gray-value shift of the current pixel (i.e., the pixel being processed) in the g_{mod} image is propagated to the unprocessed neighboring pixels in the g_{mod} image (the propagation ratios among these neighbors are determined by the weight matrix PCWM). Note that the content of the PCWM is independent of the printer being used. An example to demonstrate the need of the PCWM is given in Example 3.

Example 3. Let A be a 4×7 area (of the g_{mod} image) in which the underbared pixel shown below is the pixel being processed. For example, if the original A in the g_{mod} image is

$$A = \begin{matrix} 0.4 & 0.4 & 0.2 & 0.4 & 0.4 & 0.4 & 0.4 \\ 0.5 & 0.6 & 0.4 & \underline{0.5} & 0.6 & 0.4 & 0.5 \\ 0.5 & 0.3 & 0.3 & 0.5 & 0.3 & 0.3 & 0.5 \\ 0.4 & 0.4 & 0.2 & 0.4 & 0.2 & 0.4 & 0.4 \end{matrix}$$

then we have average gray value (local mean) $\mu_A = (0.4+0.4+0.2+\dots)/28=0.4$.

After applying the operator F without PCWM, A becomes

$$A = \begin{matrix} 0.4 & 0.4 & 0.2 & 0.4 & 0.4 & 0.4 & 0.4 \\ 0.5 & 0.6 & 0.4 & \underline{0.763} & 0.6 & 0.4 & 0.5 \\ 0.5 & 0.3 & 0.3 & 0.5 & 0.3 & 0.3 & 0.5 \\ 0.4 & 0.4 & 0.2 & 0.4 & 0.2 & 0.4 & 0.4 \end{matrix}$$

The new local mean is $\mu_A=0.41$. It is clear that the local mean had been changed from 0.4 to 0.41 (because the filter F shift the gray value of the underbared pixel from 0.5 to 0.763). To compensate the gray-value shift, we use the PCWM in our framework [see Fig. 7(b)], and the result is shown below:

$$A = \begin{matrix} 0.4 & 0.4 & 0.2 & 0.4 & 0.4 & 0.4 & 0.4 \\ 0.5 & 0.6 & 0.4 & \underline{0.763} & 0.5912 & 0.3562 & 0.4737 \\ 0.4912 & 0.2737 & 0.3 & 0.5 & 0.3 & 0.2737 & 0.4912 \\ 0.4 & 0.3912 & 0.1737 & 0.3562 & 0.1737 & 0.3912 & 0.4 \end{matrix}$$

After simple calculation we can find that the resulting local mean is $\mu_A=0.4$, which is identical to the original local mean. Therefore, the gray-value shift of the underbared pixel has been "compensated" by means of distributing the gray-value shift to its several neighbors (according to the weights described by PCWM), and the local mean is preserved. Note that due to the fact that the coefficients in Fig. 7(b) has sum $-1=-100\%$, the gray-value shift occurred at the current pixel (the underbared pixel) will be compensated quite well. Some readers might think that the change of the local mean is so small (only 0.01 in the preceding example) that correcting this change is not necessary. However, the preceding illustrative example showed just the effect of 1 pixel; and if we do not correct the change of each pixel, the gray-value shift will be accumulated, and eventually change the overall perception of gray of the halftone image. An experimental example to show the perception of gray is given in Fig. 9. Fig. 9(a) is the input continuous-tone image. Figs. 9(b) and 9(c) are the halftone images generated by our method without the PCWM and with the PCWM, respectively. We can find that the average gray value (when 0 to 255 is normalized to 0 to 1) for Figs. 9(a), 9(b), and 9(c), are, respectively, 0.79, 0.75, and 0.79. That is Fig. 9(c) really preserves the average gray of Fig. 9(a) while Fig. 9(b) does not. In other words, the one with PCWM is a little better. [The loss of illumination in Fig. 9(b) is 0.04, or equivalently, the loss is 10 in the traditional 256-level system.]

5 More Experimental Results

More images and more algorithms were tested to be more convincing. Fig. 10 was the original gray level image. Fig. 11 depicted the halftone result of our method, while Fig. 12 used the algorithm of Stucki,¹² Fig. 13 used the Eschbach's algorithm,⁹ and Fig. 14 was the result of replacing the role of our filter [Eqs. (4)] by a mean filter. Note that in Figs. 12 to 14, the false-texture contours were serious, especially in the sky. In Fig. 15, which shows the results generated by Xie and Rodriguez's algorithm,¹⁰ it is evident that Fig. 15(a) had a few false-texture contours but Fig. 15(b) had



Fig. 10 Original gray level image.



Fig. 13 Result of Eschbach's algorithm.⁹



Fig. 11 Result of the proposed algorithm.

almost none. This means that, with suitable value assignment in the so-called threshold randomization, the visual perception of the elegant method introduced in Ref. 10 can also compete with that of our method. As pointed out in Ref. 10, however, their method is computational intensive. In fact, according to our experience, our method is at least five times faster than theirs. Finally, as compared with those methods using random noise,³ the method proposed by us has the advantage that the trial-and-error process used in the approach based on random noise is no longer needed.

6 Conclusions

According to the observation of the intermediate product g_{mod} image generated by the standard ED algorithm, we found the strong connection between the false-texture contours in the output image and the banding effect in the g_{mod} image. Based on this observation, an approach to generate halftone images without false-texture contours is proposed in this paper. The approach applied a local filter to the g_{mod} image generated by the standard ED before creating the



Fig. 12 Result of Stucki's algorithm.¹²



Fig. 14 Result of using the mean filter to replace the role of the filter F defined in Eqs. (4).



(a)



(b)

Fig. 15 Results of applying the technique of Xie and Rodriguez¹⁰ to Fig. 10. Note that (a) and (b) are, respectively, the results when the magnitude of the so-called threshold randomization is unsuitably (suitably) assigned. Also note that, as was pointed out by the authors in Ref. 10, their method was computational intensive.

output image using g_{mod} . Experimental results show that the false-texture contours are really alleviated significantly.

Acknowledgments

This work was supported by the National Science Council, Republic of China, under grant NSC85-2213-E009-111. The authors also wish to thank the referees for their valuable comments, which led to the improvement of the paper.

References

1. 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. Graph. Image Process.* **5**, 13–40 (1976).
2. J. C. Stoffel and J. F. Moreland, "A survey of electronic techniques for pictorial reproduction," *IEEE Trans. Commun.* **C-29**, 1898–1925 (1981).
3. R. A. Ulichney, *Digital Halftoning*, MIT Press, Cambridge, MA (1987).
4. R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial greyscale," *Proc. Soc. Inf. Disp.* **17**, 75–77 (1976).
5. R. Eschbach and K. T. Knox, "Error-diffusion algorithm with edge enhancement," *J. Opt. Soc. Am. A* **8**, 1844–1850 (1991).
6. J. Sullivan, R. Miller, and G. Pios, "Image halftoning using a visual model in error diffusion," *J. Opt. Soc. Am. A* **10**, 1714–1724 (1993).
7. R. Levien, "Output dependent feedback in error diffusion halftoning," in *Proc. IS&T's 46th Ann. Conf.*, pp. 115–118 (1993).
8. Z. Fan, "Dot-to-dot error diffusion," *J. Electron. Imaging* **2**(1), 62–66 (1993).
9. R. Eschbach, "Reduction of artifacts in error diffusion by means of input-dependent weights," *J. Electron. Imaging* **2**(4), 352–358 (1993).
10. Z. Xie and M. Rodriguez, "A bandwidth preservation approach to stochastic screening," in *Proc. IS&T's 3rd Technical Symp. on Prepress, Proofing, & Printing*, pp. 113–116 (1993).
11. S. Weissbach and F. Wyrowski, "Error diffusion procedure: theory and applications in optical signal processing," *Appl. Opt.* **31**, 2518–2534 (1992).
12. P. Stucki, "MECCA-A multiple-error correcting computation algorithm for bilevel image hardcopy reproduction," *Research report RZ1060*, IBM Research Laboratory, Zurich, Switzerland (1981).



Wei-Yu Han received his BS degree in 1984 from National Chung Cheng Institute of Technology, Taiwan. Since 1994 he has been studying toward a PhD degree and he is currently a PhD candidate in the Computer and Information Science Department of Chiao Tung University. His research interests include image processing, digital halftoning and texture analysis.



Ja-Chen Lin received his BS degree in computer science in 1977 and his MS degree in applied mathematics in 1979, both from National Chiao Tung University, Taiwan. In 1988 he received his PhD degree in mathematics from Purdue University. In 1981 and 1982, he was an instructor at National Chiao Tung University. From 1984 to 1988, he was a graduate instructor at Purdue University. In August 1988, he joined the Computer Vision Laboratory of the Department of Computer and Information Science at National Chiao Tung University, where he is currently an associate professor. His recent research interests include pattern recognition, image processing, and parallel computing. Dr. Lin is a member of the Phi-Tau-Phi Scholastic Honor Society.