Journal of Electronic Imaging 13(3), 639–645 (July 2004).

Feature-preserving multiscale error diffusion for digital halftoning

Yuk-Hee Chan
Sin-Ming Cheung
Hong Kong Polytechnic University
Department of Electronic and Information Engineering
Centre for Multimedia Signal Processing
Hung Hom, Kowloon, Hong Kong
E-mail: enyhchan@polyu.edu.hk

Abstract. *Multiscale error diffusion is superior to conventional error diffusion methods in digital halftoning as it can eliminate directional hysteresis completely. However, there is a bias to favor a particular type of dots in the course of the halftoning process. A new multiscale error diffusion method is proposed to improve the diffusion performance by reducing the aforementioned bias. The proposed method can eliminate the pattern noise in flat regions and the boundary effect found in some other conventional multiscale error diffusion methods. At the same time, it can preserve the local features of the input image in the output. This is critical to quality, especially when the resolution of the output is limited by the physical constraints of the display unit.* © 2004 SPIE and IS&T. [DOI: 10.1117/1.1758728]

1 Introduction

Digital halftoning is a process that uses bilevel pixels (black and white pixels) to simulate a gray-scale image on a bilevel output device. There are many methods to implement digital halftoning.[1–3] Among them, error diffusion is currently one of the most popular methods to generate halftone images as it can produce images of good quality at a reasonable cost. However, as the quantization error is diffused to a predefined direction with a causal filter during the halftoning process, conventional error diffusion algorithms introduce directional hysteresis to the output halftones. Various modifications to the original error diffusion algorithm[4] have been proposed to reduce this problem. The approaches include using longer error diffusion filters,[5] using nonraster scans,[3,6,7] postprocessing the input,[8] and exploiting threshold modulation.[2,9,10] However, though most of them can effectively reduce directional hysteresis to a certain extent, none of these approaches can root directional hysteresis out as causal diffusion filters are still used. They just try to hide directional hysteresis by averaging out to zero the diffused error components in a local region. From this point of view, Peli's approach[11] is a bit better as no directional error diffusion is involved in its realization. It iteratively modifies selected binarized pixels to reduce the weighted averaged error of local regions.

Recently, an error diffusion algorithm was proposed by Katsavounidis and Kuo.[12] This algorithm exploits multiscale error diffusion technique and forms another approach to tackle directional hysteresis. This technique is superior to conventional error diffusion methods such as that in Ref. 4 in a way that no sequential predetermined order is required for error diffusion. Accordingly, noncausal filters can be used in diffusing quantization error to avoid directional hysteresis. This algorithm was improved by Chan to reduce the boundary effect and the pattern noise appeared in its diffusion output.[13]

Both Katsavounidis and Kuo's[12] multiscale error diffusion and its improved version[13] are basically realized with a two-step iterative algorithm as follows. Consider one wants to apply digital halftoning to a gray-level input image **X** whose values are within $[0,1]$ to obtain an output binary image **B**. Without loss of generality, it can be assumed that they are of size $2^k \times 2^k$ each, where k is a positive integer. At the very beginning, an error image **E** is initialized to be **X**. Then the positions of the white dots in the output image **B** are determined and the error image **E** is updated iteratively. At each iteration, a white dot (value$=1$) is first introduced at one location of the output image **B**. The location is chosen in a greedy way based on the current **E**. Then the error at that position is diffused to the neighbors of that pixel with a noncausal diffusion filter to update the error image **E**. These procedures are repeated until the sum of all elements of **E** is bounded in absolute value by 0.5. This forms the framework of the multiscale error diffusion technique.

An interesting observation we have had is that, in this multiscale error diffusion technique, no matter whether Ref. 12 or Ref. 13 is concerned, the major activity is to assign white dots to **B**. Black dots in **B** are passively assigned during the process. They are only the dots left behind after assigning the white dots and have no say to determine their positions. This implies that the diffusion technique favors bright areas or bright features. Conceptually, such a bias should be avoided. A dark area is not necessarily less important. At least dark features in bright regions are as important as bright features in dark regions.

Paper 03026 received Feb. 24, 2003; revised manuscript received Jul. 11, 2003, and Jan. 8, 2004; accepted for publication Jan. 16, 2004.
1017-9909/2004/$15.00 © 2004 SPIE and IS&T.

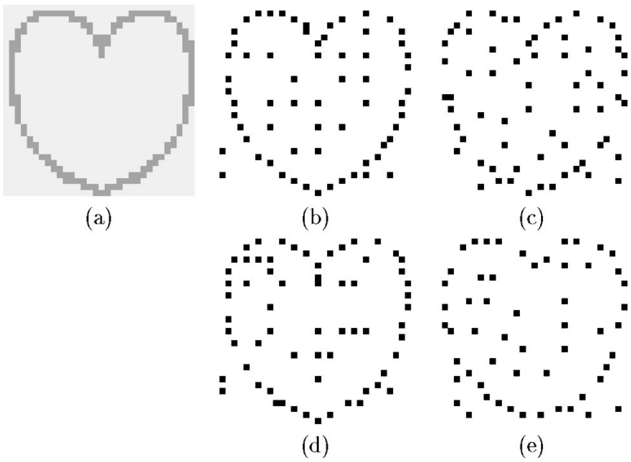


Fig. 1 (a) Original image "heart" and diffusion results of (b) Ref. 12 by assigning black dots, (c) Ref. 12 by assigning white dots, (d) Ref. 13 by assigning black dots, and (e) Ref. 13 by assigning white dots.

Another observation we have is that, in a bright region, the positions of black dots are more critical than those of white dots. Minority dots are generally more outstanding in a region and hence they should be used to highlight the local features in the region. Accordingly, in a bright region, black dots instead of white dots should be handled first due to their higher priority. Similar consideration should be made when handling a dark region. In this case, white dots should be of higher priority.

Figure 1 shows how assigning different types of dots in different scenarios affects the diffusion results of Refs. 12 and 13. The original image shown in Fig. 1(a) is a bright image the average intensity level of which is 0.936 $\in[0,1]$. Black dots are the minority. Figures 1(b) and 1(c) show, respectively, the results of iteratively assigning black dots and white dots to the output image with Katsavounidis and Kuo's algorithm.[12] As the original algorithm of Ref. 12 assigns white dots only, to obtain the simulation result of assigning black dots, we halftoned the negative image of the original image with the algorithm and then produced the negative image of the result. Note that this is equivalent to the result of iteratively assigning black dots to the output image with the algorithm. One can see that the contour of the heart is broken into pieces of dotted line segments in Fig. 1(c) but it is in a good shape in Fig. 1(b). A similar case occurs in Figs. 1(d) and 1(e), which show the results of using Chan's algorithm. These examples show that minority dots are more important in a halftoned output and they should be handled first when this multiscale error diffusion technique is exploited.

Based on the idea we have presented, in this paper, a new diffusion algorithm is proposed to improve the multiscale error diffusion technique by reducing the bias to a particular type of dots and preserving the features in a particular region of the image in the output to a certain extent. A simple trick is also presented in this paper to reduce the boundary effect introduced in Katsavounidis and Kuo's algorithm.[12]

2 Algorithm

Our proposed algorithm is a two-step iterative algorithm as well. First, the default type of dots used in the halftoning process is determined. The total energy of the image is estimated to determine how many white dots one must introduce during the halftoning process. If this is more than half of the total number of pixels, it is better to introduce black dots instead of white dots so as to reduce the realization effort. Black dots are used as the default type of dots in this case. Otherwise, white dots are used by default. Here, without losing the generality, that white dots are the default is assumed. If it is the opposite, one can invert $\mathbf{X}$ before carrying out the proposed algorithm and invert the output at the end. In this paper, inverting an image means replacing the image with its negative image.

An error image $\mathbf{E}$ is initialized to be $\mathbf{X}$ at the beginning. Note that we assume white dots are the default type of dots. If this is not the case, $\mathbf{X}$ should have been inverted and the $\mathbf{X}$ we are now referring to is actually the negative image of the original $\mathbf{E}$. Dots, either black or white, are then assigned to appropriate locations of the output image $\mathbf{B}$ one by one and $\mathbf{E}$ is updated iteratively. Basically, there are two steps at each iteration. These steps are repeated until the sum of all elements of $\mathbf{E}$ is bounded in absolute value by 0.5. The details of the two steps are as follows.

2.1 *Step 1: Determine the Right Location of a New Dot*

At the beginning of each iteration, we assume a white dot is to be introduced in the iteration. Note that this assumption is based on the default type of dots that has been selected. In contrast to conventional multiscale error diffusion algorithms, the location where a new white dot should be introduced is determined via the "extreme error intensity guidance" instead of the "maximum error intensity guidance."

The process starts with the error image $\mathbf{E}$ as the region of interest. Then the region of interest is divided into subregions and the subregion with the largest sum of its all elements is selected to be the new region of interest. This step is repeated until a particular subregion of a particular size is reached. Then, whether the average energy of $\mathbf{E}$ in that subregion is larger than 0.5 is investigated. If the criterion is satisfied, the subregion is declared to be a bright region. In this case, if the number of black dots introduced to the region does not exceed the total number of black dots that should be introduced, the dot to be introduced should be changed to a black one in this iteration. This can be realized by inverting the active elements of $\mathbf{E}$ in the corresponding subregion with $e_{i,j}=1-e_{i,j}$, where $e_{i,j}$ denotes the value of the pixel of $\mathbf{E}$ at location (i,j). Active elements are those elements whose corresponding elements in $\mathbf{B}$ have not yet been assigned a dot.

After making the decision and working accordingly, the aforementioned division procedure is repeated as before until a particular pixel location is reached. Note that even though the region with the largest sum of its all elements is always selected, it is not following the "maximum error intensity guidance" as the active elements of $\mathbf{E}$ may be inverted in the course. It is actually following the "minimum error intensity guidance" when they are inverted.

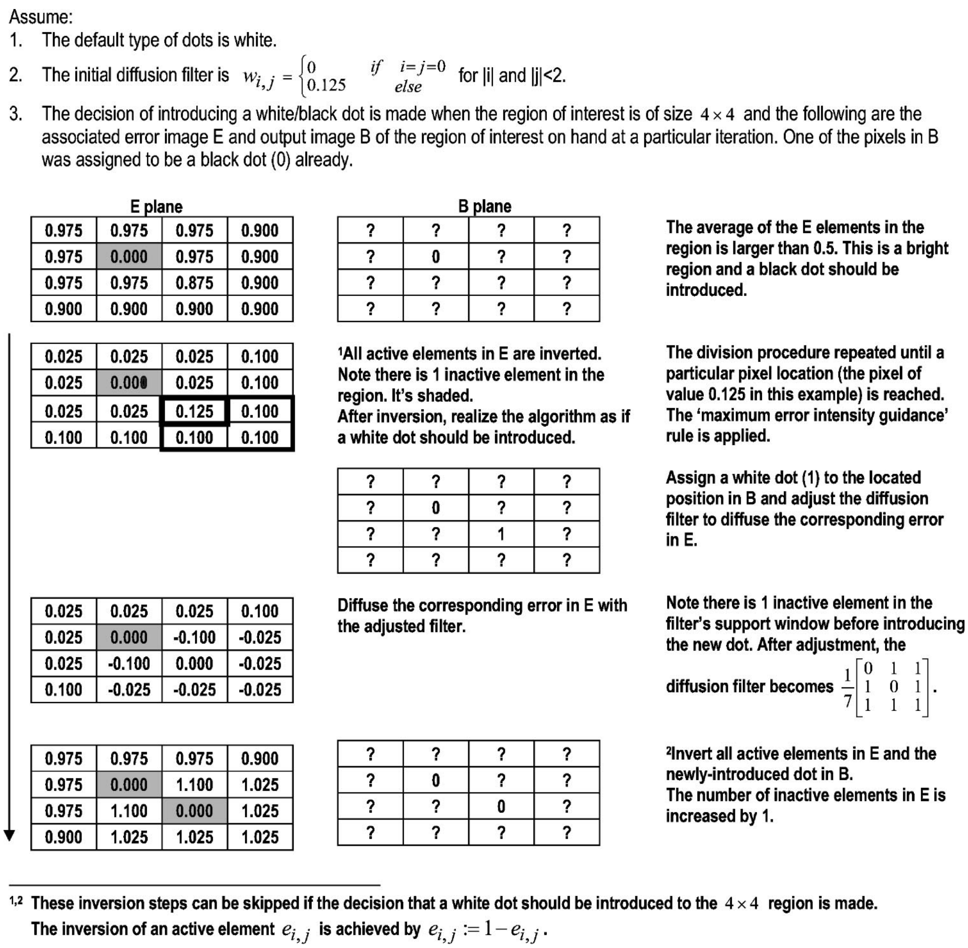


Fig. 2 Example of how the algorithm introduces a black dot to a bright region.

In Katsavounidis and Kuo's approach,[12] a region of interest is partitioned into four nonoverlapped subregions to locate the next region of interest. At a particular scale, the borders of regions are fixed, which restricts how to locate the next region of interest to a certain extent. In other words, boundary effect exists. Chan solved this problem by overlapping subregions.[13]

A simple trick is introduced in the proposed algorithm to reduce this effect. Before realizing the proposed two-step algorithm, a 1-pixel frame of value 0 and a 1-pixel frame of value 1 are, respectively, added to **E** and **B** first. At each iteration, the size of the starting region of interest remains to be that of the original **E** but the region is shifted by a random offset $(x_{\text{off}}, y_{\text{off}})$, where x_{off} and $y_{\text{off}} \in [-1,0,1]$. By doing so, the borders of regions vary at different iterations and hence the boundary effect can be reduced. After all dots are assigned, the frame of **B** is removed to get the final halftoned output.

2.2 *Step 2: Update Error Image* E

After the right pixel position is located, a dot is assigned to it. The dot could be a white dot or a black dot and it depends on the decision made in step 1. Here, it is assumed that a white dot is assigned to the selected position, say, (m,n), by making $b_{m,n}=1$ and a noncausal diffusion filter H with a support window $\Omega \equiv \{(x,y) | 0 \leqslant |x|,|y| \leqslant \text{half window size}\}$ is used.

Let $e_{i,j}$ and $b_{i,j}$ be, respectively, the values of the pixels of **E** and **B** at location (i,j) after the dot assignment but before the error diffusion. Then, after the error diffusion, the new value of $e_{i,j}$, say $e'_{i,j}$ is assigned to be

$$e'_{i,j} = \begin{cases} 0 & \text{if } (i,j)=(m,n) \\ e_{i,j} - w_{i-m,j-n} a_{i,j} (1-e_{m,n})/s & \text{else,} \end{cases} \tag{1}$$

where $w_{u,v}$'s are the filter weights of filter H, $a_{i,j}$ is an $e_{i,j}$-dependent parameter defined as

$$a_{i,j} = \begin{cases} 1 & \text{if } e_{i,j} \text{ is an active element and } (i,j) \neq (m,n) \\ 0 & \text{else,} \end{cases} \tag{2}$$

and

$$s = \sum_{(i-m,j-n)\in\Omega} w_{i-m,j-n} \times a_{i,j}. \tag{3}$$

Note this assignment causes no error leakage in the error diffusion and the algorithm works with any choice of filter H, producing different results. In the case where $s=0$, a filter with a larger support window should be exploited to make $s\neq 0$ and keep the algorithm working.

In Katsavounidis and Kuo's approach,[12] quantization error at (m,n) will be diffused to its neighbors even though they are already assigned to be white dots. This amount of error will then be stored in these locations forever and will not contribute to the following quantization and diffusion stages. This effect results in an uneven error image $\mathbf{E}$ at the end and makes the introduced dots not properly distributed in a local region. The proposed approach can obviously solve this problem.

If the dot to be assigned to the position is a black one, the same procedures for assigning a white dot can still be exploited. Since the active elements of $\mathbf{E}$ were inverted earlier in this case, carrying out the same procedures is actually equivalent to assigning a black dot to the position as long as an adjustment is performed afterward. Specifically, the adjustment is made by inverting the active elements of the updated $\mathbf{E}$ in the concerned subregion with $e_{i,j}=1-e_{i,j}$ and inverting $b_{m,n}$ to zero after performing the procedures already described.

Figure 2 shows an example of how the algorihm works when the point at which the decision of introducing a black or white dot should be made is reached. In this example, the decision is made when the region of interest is of size 4×4 and the region concerned is a bright region, as shown in the figure.

3 Simulation Results

Simulations were carried out to evaluate the performance of the proposed algorithm. First, we compare the performance of various algorithms that try to tackle directional hysteresis with different approaches. The selected algorithms cover all kinds of approaches and Fig. 3 shows their simulation results. The 3×3 noncausal diffusion filter used in Ref. 12 was used in realizing Katsavounidis and Kuo's algorithm. This filter was also used as the initial version of H in realizing Chan's[13] algorithm and the proposed algorithm. In the realization of these two algorithms, the actual diffusion filter operated on a particular local region of $\mathbf{E}$ was adjusted based on H, as described in Eqs. (1) to (3) according to the distribution of the active elements in the region. When $s=0$, H evolved to have a larger support. In formulation, the H used in our simulation can be generalized as

$$w_{i,j} = \begin{cases} 0 & \text{if } i=j=0 \\ (2d+1-|i|-|j|)/S & \text{else} \end{cases} \quad \text{for } |i|,|j| \leq d, \tag{4}$$

where S is the total sum of all $w_{i,j}$'s, $(2d+1)\times(2d+1)$ is the size of the filter support, and d is a positive integer. When the side and corner pixels were handled, some $w_{i,j}$'s might fall out of the boundary. In such a case, they were set to be zero. Note that, when $d=1$, the filter defined in Eq. (4) is exactly the same as the one used in Ref. 12. When $s=0$ in a particular local region, the H used for that region was extended by increasing d by 1. In the realization of the proposed algorithm, the decision of introducing a white dot or a black dot was made when the region of interest was of size 16×16.

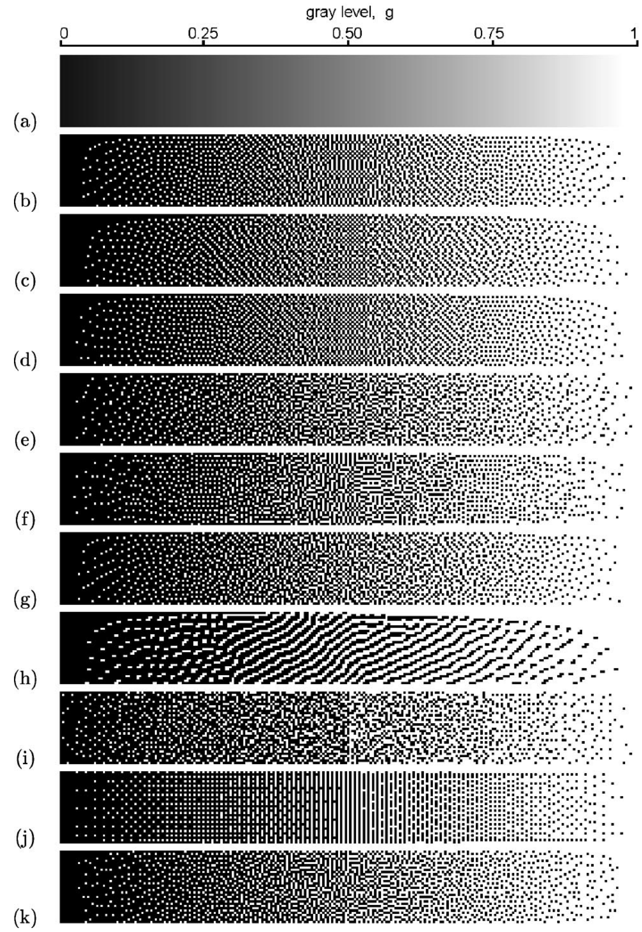


Fig. 3 (a) Original ramp image and diffusion results of (b) Ref. 4, (c) Ref. 5, (d) Ref. 6, (e) Ref. 3, (f) Ref. 8, (g) Ref. 9, (h) Ref. 2; (i) Ref. 11, (j) Ref. 12, and (k) the proposed algorithm.

Figure 3(b) shows the performance of standard error diffusion.[4] The original image is the ramp image shown in Fig. 3(a). The diffusion starts from the left upper corner of the image. The artifacts caused by directional hysteresis are obvious in Fig. 3(b), and can be identified by the virtual arcs formed by connecting the front-line minority dots in the upper corner of the output. The directional diffusion of quantization error diffuses the error to the next line and hence, when the next line is processed, it will be sooner when the accumulated error is larger than the threshold so as to produce a minority dot. This causes a corresponding directional shift of minority dots even though the energy distribution of each row is identical.

These virtual arcs can also be found in Figs. 3(c), 3(d), 3(g), and 3(h). This implies that using a longer error diffusion filter,[5] using a serpentine scan,[6] and using threshold modulation[2,9] are not able to eliminate directional hysteresis. These approaches can handle directional hysteresis

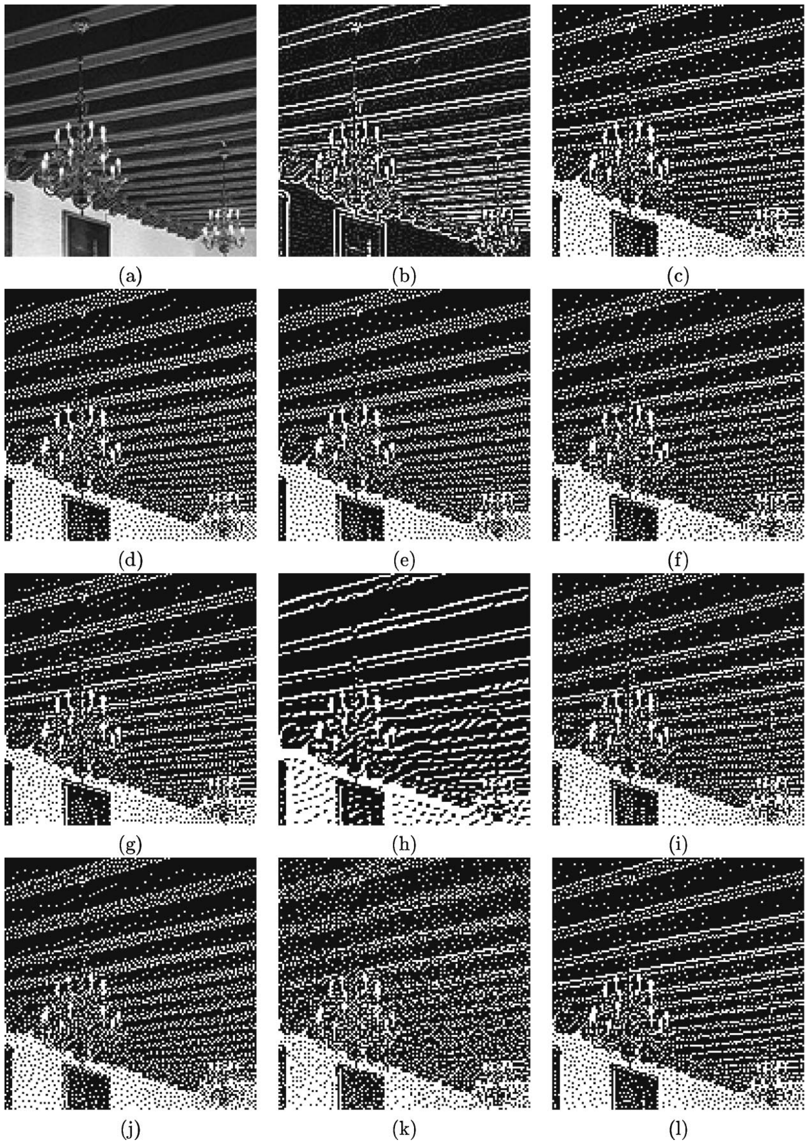


Fig. 4 (a) Original image; (b) edges detected in (a); and diffusion results of (c) the proposed algorithm, (d) Ref. 4, (e) Ref. 6, (f) Ref. 3, (g) Ref. 8, (h) Ref. 2, (i) Ref. 12, (j) Ref. 9, (k) Ref. 11, and (l) Ref. 13.

along the rows, but they cannot handle downward hysteresis.

Meşe and Vaidyanathan's dot diffusion[3] and Kumar and Makur's postprocessing approach[8] can handle vertical hysteresis to a certain extent and hence the virtual arcs disappear in Figs. 3(e) and 3(f). Meşe and Vaidyanathan's algorithm divides the input into a number of blocks and then processes the pixels of each block in a predefined order. As a result, periodic patterns appear in Fig. 3(e). One can see that there are pairs of ''eyes'' located at $g=0.26$, 0.44, 0.58, 0.7, and 0.88. The 16×16 class matrix suggested in Table IV of Ref. 7 was used in the simulation, and hence the 32×256 halftone output shows two complete cycles along the vertical direction. Kumar and Makur's algorithm[8] diffuses the intermediate output in different directions iteratively so as to remove directional hysteresis. Every single iteration corresponds to a halftoning process of the image and introduces some noise to the image. The larger the number of iterations, the more the image is degraded.

Katsavounidis and Kuo's[12] algorithm exploits multiscale error diffusion. Theoretically, multiscale error diffusion is superior to other approaches in removing directional hysteresis as it can eliminate it completely. The regular shift of minority dots caused by directional error diffusion cannot be found in Fig. 3(j). From Fig. 3(j), one can see that dots are uniformly distributed according to the gray level of each column of pixels. However, as the error diffusion is not biased to a particular direction, regular dot patterns are everywhere when the input is a uniformly distributed pattern such as Fig. 3(a). Peli's multiscale processing algorithm[11] removes the pattern noise by introducing some random disturbance. However, simulation result shows that there is a visible step change at $g=0.50$, as shown in Fig. 3(i).

Figure 3(k) shows the simulation result of the proposed algorithm. Like Katsavounidis and Kuo's algorithm, as multiscale error diffusion technique is used, there is no directional hysteresis. As compared with Fig. 3(j), the pattern noise is removed.

Figure 4 shows some other simulation results for evaluating the feature preserving performance of the proposed algorithm as compared with other algorithms. Though the feature-preserving performance of multiscale error diffusion is the concern in the paper, we purposely include the results of some other halftoning algorithms for reference. The pictures in Fig. 4 are actually the halftoning results of image ''Ceiling.'' Edge is one of the important features in an image. Figure 4(b) shows the edges detected in the original shown in Fig. 4(a) and can be used as a reference to compare the performance of various algorithms. From Fig. 4(a), one can see that there are strips on the ceiling and there are white lines on the edges of each strip. These white lines are missing in Figs. 4(d), 4(e), 4(f), 4(j), and 4(k), but they are highly visible in the results of the proposed algorithm and Chan's algorithm, as shown in Figs. 4(c) and 4(l), respectively. Figure 4(i) shows Katsavounidis and Kuo's result. Some white lines are missing and the ''virtual'' edges of the first and the second strips from the top of the image are not as straight as those in Fig. 4(c). The performance of the Ref. 13 algorithm is very close to that of the proposed algorithm, but one can still tell the difference when comparing the diffusion results of the canopy of the chandelier on the left of the image. One can tell from Fig. 4(c) that the canopy is cone-shaped but it is impossible to determine this with Fig. 4(l). One can also compare the corresponding region in Fig. 4(b) to find that Fig. 4(c) can report the canopy in a more faithfully way. The algorithm proposed in Ref. 8 is an iterative algorithm. The output of the standard error diffusion[4] was used as its initial estimate in the simulation. By using Fig. 4(b) as a reference, one can see that the canopy of the chandelier on the right of the image and the first corbel on the left of the image are missing in Fig. 4(g), while they are reported in Fig. 4(c).

The features of an image are not limited to edges. The smoothness of flat regions is also an important feature in an image. Figure 5 shows the diffusion results of different algorithms when they are used to process a flat gray-level image of intensity level 50/255. This intensity level is randomly selected. One can see that there is serious pattern noise in Fig. 5(a). As Katsavounidis and Kuo's algorithm partitions a region into nonoverlapped subregions in a fixed manner, it results in a fixed dot assignment pattern and hence pattern noise is unavoidable in a flat region. The proposed algorithm can solve this problem.

In the realization of the proposed algorithm, the region of interest can be divided into nine overlapped subregions instead of four nonoverlapped subregions to locate the next region of interest at a particular scale as in Ref. 13. Figures 5(b) and 5(c), respectively, show the case of using the 1-to-4 scheme and that of using the 1-to-9 scheme. Theoretically speaking, the proposed algorithm works better with the 1-to-9 scheme as it is less restricted. However, simulation results show that the visible difference is very small. In contrast to Ref. 13, the proposed algorithm removes the boundary effect by shifting the window used to define a region of interest instead of overlapping subregions. This would be an advantage as it handles fewer regions at a time and in turns reduces the complexity. The complexity of the proposed algorithm is more or less the same as that of Ref. 12. As parallel processing is allowed in its realization, real-time halftoning can be achieved with the processed algorithm.

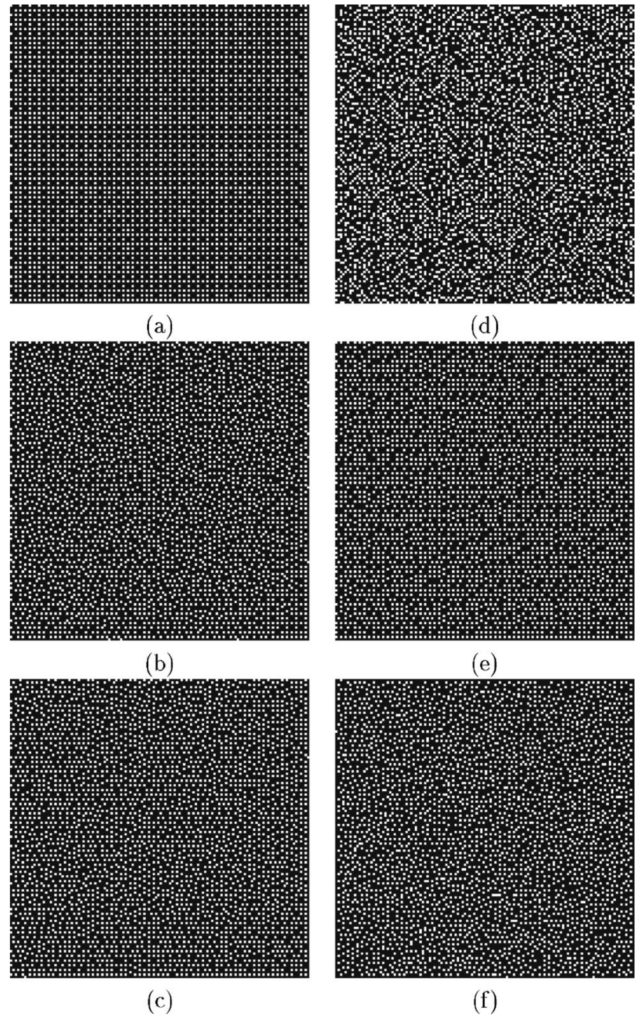


Fig. 5 Diffusion results of a flat image of gray level 50/255: (a) Katsavounidis and Kuo's algorithm,[12] (b) the proposed algorithm with the 1-to-4 division scheme, (c) the proposed algorithm with the 1-to-9 division scheme, (d) Peli's algorithm,[11] (e) Chan's algorithm,[13] and (f) Kumar and Makur's algorithm.[8]

As compared with Figs. 5(b) and 5(c), Fig. 5(d) looks like a rough surface that is full of defects. It appears that black holes of large size are everywhere. The appearance of Figs. 5(e) and 5(f) is better, but it is not as fine as Figs. 5(b) and 5(c). The size of the apparent black holes scattered over Figs. 5(e) and 5(f) is somewhat larger than that scattered over Figs. 5(c) and 5(d).

Note here that the proposed algorithm preserves the local features of the image by putting minority dots in the right positions during halftoning. The features are not highlighted by enhancing the edges before halftoning the original image. Obviously, preprocessing distorts the original image and introduces noise to the image. Edge enhancement also leads to disturbance to the intensity of the original. This does not happen in our case.

Some algorithms may enhance edges via adjusting some tuning parameters to make the thresholding process more sensitive to the intensity difference between adjacent pixels along the scan. To a certain extent, one can consider this as embedding image enhancement in halftoning. In our approach, no parameter is tuned for this purpose and the feature is preserved naturally. After all, there is no tuning parameters for multiscale halftoning algorithms including those of Refs. 11, 12, and 13 to enhance or preserve edges.

4 Conclusions

The performance of various halftoning algorithms proposed for removing directional hysteresis was evaluated. It was found that multiscale error diffusion can effectively eliminate directional hysteresis with a noncausal diffusion filter. A new digital halftoning algorithm based on multiscale error diffusion was also proposed. In contrast to Refs. 12 and 13, the ''extreme error intensity guidance'' principle was adopted. This reduced the bias to bright areas and preserved the local features in the image. In addition, the proposed algorithm can effectively remove the boundary effect and the pattern noise introduced by Ref. 12.

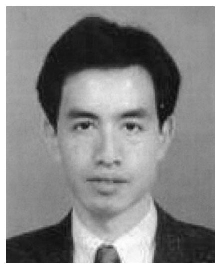
When the resolution of the processing image is high, whether the minority dots can be located in the critical positions to show local features of an image may not be critical because dots are eventually filtered by the eyes and their exact positions are fuzzy to the viewer. However, when both the original and the output halftone are of limited resolution, say 150×200 dots, this is a critical matter as dots can be visible at the viewing distance. This case is not uncommon when a user views images with a small display unit such as a mobile phone or a handheld device.

Acknowledgments

The work described in this paper was substantially supported by Grant No. A046 from the Center for Multimedia Signal Processing, Department of Electronic and Information Engineering, the Hong Kong Polytechnic University.

References

1. R. Ulichney, ''Dithering with blue noise,'' *Proc. IEEE* **76**(1), 56–79 (1988).
2. D. L. Lau, G. R. Arce, and N. C. Gallagher, ''Green-noise digital halftoning,'' *Proc. IEEE* **86**(12), 2424–2444 (1998).
3. M. Meşe and P. P. Vaidyanathan, ''Recent advances in digital halftoning and inverse halftoning methods,'' *IEEE Trans. Circuits Syst.* **49**(6), 790–805 (2002).
4. R. W. Floyd and L. Steinberg, ''An adaptive algorithm for spatial gray scale,'' in *Proc. SID Int. Symp. Digest of Technical Papers*, pp. 36–37, Society for Information Displays, (1975).
5. P. Stucki ''MECCA-A multiple-error correcting computation algorithm for bilevel hardcopy reproduction,'' Res. Rep. RZ1060, IBM Res. Lab., Zurich (1981).
6. Z. Fan, ''Error diffusion with a more symmetric error distribution,'' *Proc. SPIE* **2179**, 150–158 (1994).
7. M. Meşe and P. P. Vaidyanathan, ''Optimized halftoning using dot diffusion and methods for inverse halftoning,'' *IEEE Trans. Image Process.* **9**(4), 691–709 (2000).
8. A. Kumar and A. Makur, ''On the phase response of the error diffusion filter for image halftoning,'' *IEEE Trans. Image Process.* **8**(9), 1282–1292 (1999).
9. N. Damera-Venkata and B. L. Evans, ''Adaptive threshold modulation for error diffusion halftoning,'' *IEEE Trans. Image Process.* **10**(1), 104–116 (2001).
10. K. Knox and R. Eschbach, ''Threshold modulation in error diffusion,'' *J. Electron. Imaging* **2**, 185–192 (1993).
11. E. Peli, ''Multiresolution, error-convergence halftone algorithm,'' *J. Opt. Soc. Am. A* **8**(4), 625–633 (1991).
12. I. Katsavounidis and C. C. J. Kuo, ''A multiscale error diffusion technique for digital halftoning,'' *IEEE Trans. Image Process.* **6**(3), 483–490 (1997).
13. Y. H. Chan, ''A modified multiscale error diffusion technique for digital halftoning,'' *IEEE Signal Process. Lett.* **5**(11), 277–280 (1998).



Yuk-Hee Chan received his BSc degree with honors in electronics from Chinese University of Hong Kong in 1987, and his PhD degree in signal processing from Hong Kong Polytechnic University in 1992. Between 1987 and 1989, he worked as an R&D engineer at Elec & Eltek Group, Hong Kong. He joined Hong Kong Polytechnic University in 1992 and is now an associate professor in the Department of Electronic & Information Engineering. Dr. Chan has published more than 85 research papers in various international journals and conferences. His research interests include image and video compression, image representation, image restoration, and fast computational algorithms in digital signal processing. Dr. Chan is a member of IEEE and a member of IEE. He obtained the Sir Edward Youde Memorial fellowship in 1991 and 1992.

Sin-Ming Cheung received his Higher Diploma, BEng(Hons.), and MPhil degrees in electronic engineering from Hong Kong Polytechnic University in 1997, 1999, and 2003, respectively. He is currently working in the R&D Department in COMedia Ltd. (Hong Kong) to improve the performance of CMOS-image-sensor-based applications. His research interests are in digital halftoning, lossy compression, digital signal processing, and embedded systems.