

A MULTISCALE ERROR DIFFUSION TECHNIQUE FOR DIGITAL HALFTONING WITH DOT-OVERLAP COMPENSATION

Yuk-Hee Chan, H.T. Kwong, C.K. Li and K.T. Lo

Department of Electronic and Information Engineering
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong

ABSTRACT

In this paper, a digital halftoning method is proposed based on a recently-proposed multiscale error diffusion technique for digital halftoning. It can definitely improve the diffusion performance by effectively removing pattern noise and eliminating boundary and 'blackhole' effects. A dot-overlap compensation scheme is also proposed to eliminate the bias in the gray scale of the printed images.

EDICS: IP 3.2 Quantization and Halftoning

1. INTRODUCTION

Digital halftoning is a method that uses bilevel pixels (black and white pixels) to simulate a gray-scale image on a bilevel output device. Recently, a multiscale level error diffusion has been proposed[1]. This method is superior to the conventional error diffusion methods such as [2] in a way that no sequential predetermined order is required for error diffusion. Accordingly, non-causal filters can be used in diffusing quantization error to avoid directional hysteresis.

In this letter, two contributions will be made. First, two modifications will be presented to improve the multiscale level error diffusion scheme such that a more pleasant output can be provided. Second, we shall introduce a compensation scheme to remove the dot-overlap distortion when the algorithm is used to generate printer output.

2. ALGORITHM

Consider we want to apply digital halftoning to a gray-level input image \mathbf{X} whose values are within $[0,1]$ to obtain an output binary image \mathbf{B} . Without loss of generality, we assume they are of size $2^k \times 2^k$ each, where k is a positive integer. Our proposed algorithm

is a two-step iterative algorithm. At each iteration, we first introduce a white dot (value=1) at one location of the output image \mathbf{B} . The location is chosen in a greedy way based on a corresponding diffused error image \mathbf{E} . Then we diffuse the error to the neighbors of that pixel to update the error image \mathbf{E} . These procedures are repeated until the sum of all elements of \mathbf{E} is bounded in absolute value by 0.5. The error image \mathbf{E} is initialized to be \mathbf{X} at the beginning.

1. Determine the right location of a new dot:

The location where a new white dot should be introduced is determined via the so-called 'maximum error intensity guidance'. We start with the error image \mathbf{E} as our region of interest. Then we divide the region of interest into subregions and select the one with the largest sum of its all elements to be the new region of interest. This step is repeated until a particular pixel location is reached.

Our approach is different from Katsavounidis's approach[1] in a way that, instead of partitioning the region of interest into 4 non-overlapped subregions to locate the next region of interest in each step, we divide it into 9 overlapped subregions as shown in figure 1. Flexibility is introduced by doing so as more possible paths are available to reach a particular location. Comparatively, it is potentially more difficult to be trapped in a local optimum and hence the region of the most significant error can be taken care in an earlier stage. Besides, it is more difficult to fall into a particular dot assignment pattern caused by the limited number of paths, which is helpful to reduce the pattern noise in the output. Finally, since we allow subregions to overlap each other, there will not be any boundary effect.

2. Update error image \mathbf{E} :

After locating the right pixel position, a white dot is assigned to it. Without loss of generality, we assume we assign the white dot to the pixel at location

This work was supported by POLYU Grant 0759.000.00.420.

(m,n) by making $b_{m,n}=1$ and a noncausal diffusion filter with a support window $\Omega \equiv \{(x,y) | 0 \leq |x|, |y| \leq \text{half window size}\}$ is used. Let $e_{i,j}$ and $b_{i,j}$ be respectively the values of the pixels of \mathbf{E} and \mathbf{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

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

where $w_{u,v}$'s are the filter weights, $\bar{b}_{i,j} = 1 - b_{i,j}$, and

$$s = \sum_{(i-m,j-n) \in \Omega} w_{i-m,j-n} \times \bar{b}_{i,j} \quad (2)$$

Note this assignment causes no error leakage in the error diffusion and the algorithm works with any choice of filter, producing different results. In the case when $s=0$, we exploit a filter with larger support window to make $s \neq 0$ and keep the algorithm work.

In Katsavounidis's approach[1], 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. For the sake of reference, we call this effect as 'blackhole' effect as energy is absorbed to these locations. This effect results in an uneven error image \mathbf{E} at the end and makes the white dots not properly distributed in a local region. Our proposed approach can obviously solve this problem.

3. DOT-OVERLAP COMPENSATION

Although error diffusion schemes including the multiscale level error diffusion scheme can generate very pleasant textures, it suffers from the dot-overlap effect introduced in laser printers, and leads to a bias in gray scale. In order to compensate for this effect, we suggest carrying out a compensation step between procedures 1 and 2 in each iteration.

Let's assume that an ideal pixel occupies an unity square. When a circular dot is put on pixel $p_{m,n}$, some area of $p_{m,n}$'s neighbors will also be covered by the dot as shown in figure 2. Accordingly, we should compensate for this effect by adjusting \mathbf{E} with $e''_{m+u,n+v} = e_{m+u,n+v} - a(u,v)$ before carrying out procedure 2, where $e''_{m+u,n+v}$ is the compensated value, $a(u,v)$ is the area covered in pixel (m+u,n+v) due to the dot assignment and $(u,v) \in \{(x,y) | x, y \in \{-1, 0, 1\}\} \setminus \{(0,0)\}$. For instance, in the case shown in Fig.2, we have $a(0,-1)=a(-1,0)=\alpha$, $a(-1,1)=a(-1,-1)=a(1,-1)=\beta$, $a(0,1)=a(1,0)=\alpha - \gamma$ and $a(1,1)=0$, where α , β and γ are respectively the

area of the shaded areas shown in Fig.2. The values of $a(u,v)$'s can be precalculated and stored in a table such that $a(u,v)$'s can be determined easily with a table lookup process.

Table 1 can be used for such a purpose. Parameters α , β and γ are respectively the area of the shaded areas shown in figure 2. The input pattern ψ is made up with the $b_{i,j}$'s in the corresponding positions of the most updated image \mathbf{B} . Table 1 shows the cases for getting $a(1,0)$ and $a(-1,1)$. Due to the geometrical symmetry, one can make use of the same table to get $a(u,v)$'s of other $p_{m,n}$'s neighbors by providing proper ψ 's after rotating the images. For instance, in the case as shown in figure 2, we let ψ be {00010} and then after a table lookup we have $a(0,1)=\alpha - \gamma$.

Unlike those compensation algorithms such as [3], [4] and [5] for typical error diffusion schemes, the compensation in this scheme is not biased to the diffusion direction, and, directly takes effect on $p_{m,n}$'s neighbors where the area is actually covered instead of $p_{m,n}$ itself.

Note here we assume a white ink dot is printed on a black paper. If it is the opposite, one can invert \mathbf{X} before carrying out the proposed algorithm and complement the output at the end.

Table 1: Area to be covered in adjacent pixels by printing a new dot on $p_{m,n}$.

Input pattern ψ^*	$a(1,0)$
{X X 1 X X #}	0
{0001X} or {X 1000}	$\alpha - \gamma$
{X 101X}	$\alpha - 2\gamma$
{1001X} or {X 1001}	$\alpha - \beta - \gamma$
{10000} or {00001}	$\alpha - \beta$
{10001}	$\alpha - 2\beta$
{00000}	α

* $\psi = \{b_{m,n+1}, b_{m+1,n+1}, b_{m+1,n}, b_{m+1,n-1}, b_{m,n-1}\}$
for getting $a(1,0)$

Input pattern ψ^{**}	$a(-1,1)$
{000}	β
{X X X} \setminus {000}	0

** $\psi = \{b_{m-1,n}, b_{m-1,n+1}, b_{m,n+1}\}$ for getting $a(-1,1)$
X is don't care.

4. SIMULATION RESULTS

Simulations have been carried out to evaluate the performance of the proposed algorithm. Figure 3a shows a

256 × 256 point light source pattern. Figures 3b and 3c show the results obtained with Katsavounidis's and our algorithms. The 3×3 non-causal diffusion filter used in [1] was used to realize both algorithms. However, the size of the filter used was adjusted when $s=0$ happened in realizing our algorithm. We deliberately printed the figures using a 600dpi laser printer so that the individual dots can be clearly printed and the effect of dot overlapping is not dominant. One can clearly see the pattern noise in Fig.3b especially in the upper right quadrant while hardly find any in Fig.3c. Figures 4a-c show another simulation result on a ramp image of size 256 × 256. No compensation was done to get Fig.3c and Fig.4c.

Figures 3d and 3e show the effect of dot-overlap compensation. A printer model of $\chi = 1.1$, which is defined as the ratio of the actual dot radius to the ideal dot radius $T/\sqrt{2}$, was used for compensation and generating the printed outputs. The connection among χ , α , β , and γ is given in [3]. A darker output is obtained without compensation as shown in Fig.3d.

5. CONCLUSION

In this letter, we proposed a new digital halftoning algorithm based on multiscale error diffusion. Compare with [1], it is more faithful to the "maximum error intensity guidance" principle. The proposed algorithm can definitely reduce pattern noise and eliminate the boundary effect and the 'blackhole' effect. Finally, a compensation scheme for this algorithm is proposed to compensate for the dot-overlap effect introduced by printers.

6. REFERENCES

- [1] I. Katsavounidis and C.C. J. Kuo, "A multiscale error diffusion technique for digital halftoning," *IEEE trans. on image processing*, Vol.6, No., Mar 1997, pp.483-490.
- [2] R. W. Floyd and L. Steinberg, "An Adaptive Algorithm for Spatial Gray Scale," *Proc. SID International Symposium Digest of Technical Papers*, Society for Information Displays, 1975, pp.36-37.
- [3] Y. Lin and T.C. Ko, "A modified model-based error diffusion," *IEEE Signal Processing Letters*, Vol. 4, No.2, Feb 1997, pp.36-38.
- [4] T. N. Pappas and D. L. Neuhoff, "Printer models and error diffusion," *IEEE trans. on Image Processing*, Vol.4, No.1, Jan 1995, pp.66-79.
- [5] Y.H.Chan, H.T.Kwong, K.T.Lo and C.K.Li, "An error diffusion technique with reduced directional

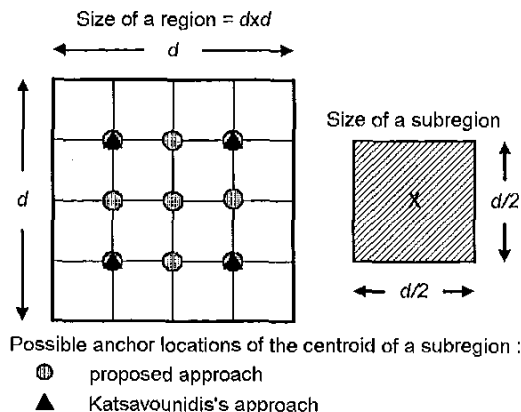


Figure 1: Subregions of a region of interest.

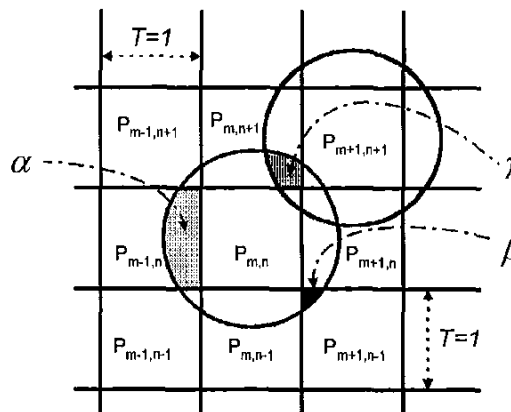
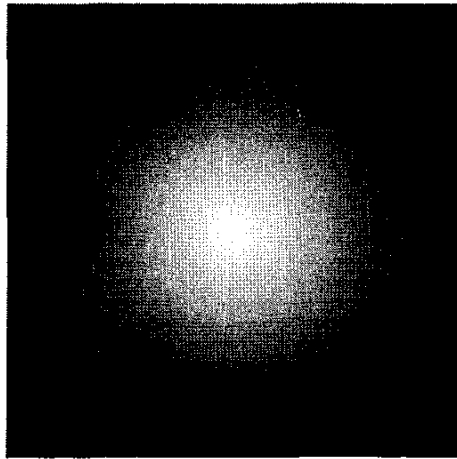
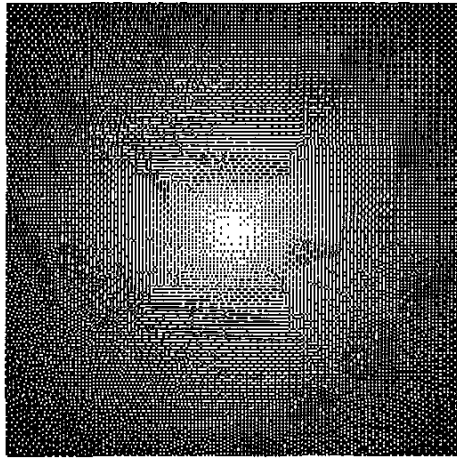


Figure 2: Circular dot-overlap model

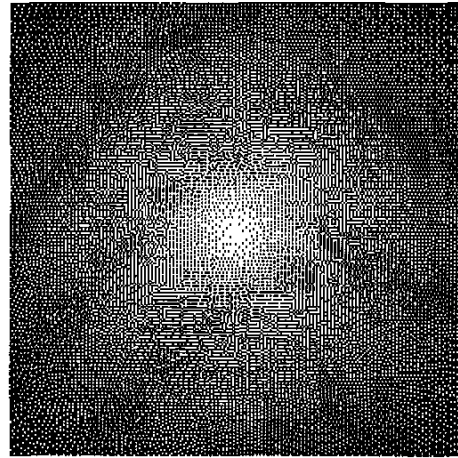
hysteresis," *Proceedings, IEEE ISCAS'99*, May 30 - June 2, 1999, Orlando, Florida, USA. Vol. IV, pp.263-266.



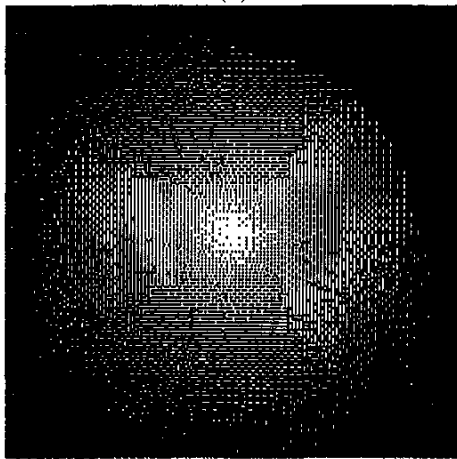
(a)



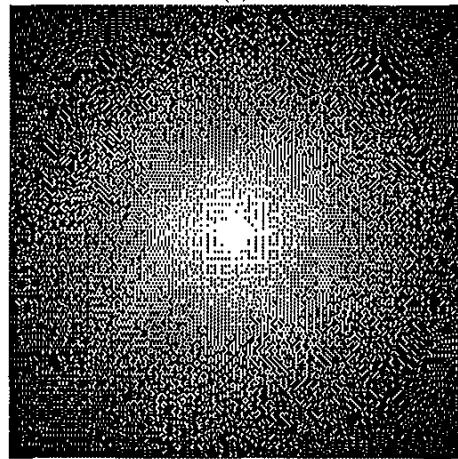
(b)



(c)

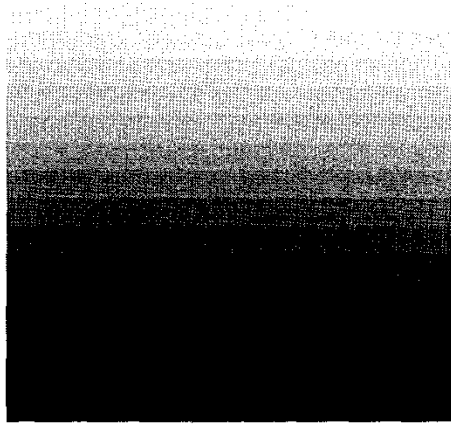


(d)

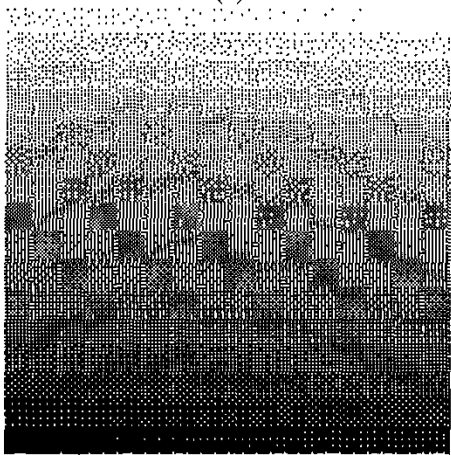


(e)

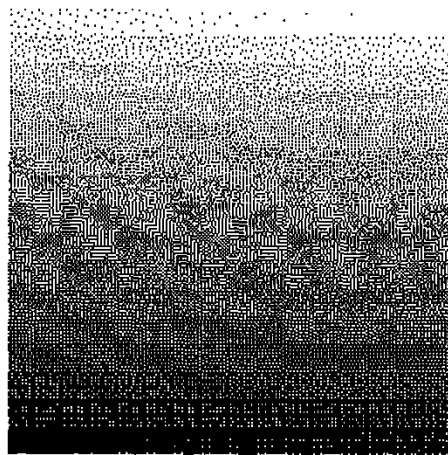
Figure 3: (a): Original test pattern; (b): Diffusion result of Katsavounidis's algorithm [1]; (c): Diffusion result of the proposed algorithm; (d): Actual printed output of Fig.3b; (e): Actual printed output of the compensated result of the proposed algorithm.



(a)



(b)



(c)

Figure 4: (a): Original ramp image; (b): Diffusion result of Katsavounidis's algorithm [1]; (c): Diffusion result of the proposed algorithm.