A Low-complexity Multiscale Error Diffusion Algorithm for Digital Halftoning

King-Hong Chung, Yik-Hing Fung, Ka-Chun Lui and Yuk-Hee Chan

Centre for Multimedia Signal Processing
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong

ABSTRACT

Multiscale error diffusion (MED) digital halftoning technique outperforms classical conventional error diffusion techniques as it can produce a directional- hysteresis-free bi-level image. However, extremely large computation effort is required for its implementation. In this paper, a fast MED-based digital halftoning technique is proposed to produce a halftone image without directional hysteresis at a significantly reduced computational cost. The amount of reduction is monotonic increasing with the image size. For an image of size 512×512, the proposed algorithm can save 40% of arithmetic operations as compared with MED. Moreover, since it supports parallel processing, processing time can further be squeezed.

I. INTRODUCTION

In digital halftoning, a gray scale image is digitally processed, via either software or hardware, to generate a black and white image such that an illusion of a continuous tone picture can be emulated for binary devices like fax machines, printer and bi-level display panels. There are many methods to perform digital halftoning [1-3]. Error diffusion is one of the most popular digital halftoning techniques nowadays due to its simplicity and fairly good overall visual quality. However, because causal error diffusion filters and deterministic scanning paths are generally used in conventional error diffusion algorithms, visual artifact of directional hysteresis usually appears in their output halftone images.

Multiscale error diffusion (MED) [4] is a digital halftoning technique which tackles this problem by introducing a non-causal error diffusion filter and a non-deterministic scanning path and hence provides a superior result. It iteratively assigns a white dot by searching the brightest region of the given image according to the maximum intensity guidance until the total energy of the output image approximately equals that of the given image. However, heavy computation effort is required to search the locations for error diffusion especially when the image size is large.

In this paper, a fast MED-based halftoning technique is proposed to render a bi-level image with significantly reduced complexity without degrading the output image quality as compared with MED. This paper is organized as follows. First, a brief introduction of MED is given in Section II. The proposed algorithm is then introduced in Section III. In Section IV, simulation results for evaluating the computation complexity and the output quality of the proposed algorithm are presented. Finally, a conclusion is made in Section VI.

II. CONVENTIONAL MED

Let X be a given gray scale image of size $N\times N$ and B be the corresponding bi-level output image. $x(i,j)$ and $b(i,j)$ are, respectively, the $(i,j)^{th}$ pixels of X and B. Without loss of generality, $x(i,j)$ is bounded by [0,1] and $b(i,j)$ is either 0 or 1.

The original MED halftoning technique is an iterative algorithm [4]. At the very beginning, B is initialized to be a black image by assigning $b(i,j)=0$ for all (i,j) and the total intensity value of the given gray scale image X, say I , is first calculated as follows.

$$I=\sum_{i=1}^{N}\sum_{j=1}^{N}x(i,j) \tag{1}$$

White dots are then assigned to the output bi-level image B one by one iteratively. At each iteration, a location in B is located by searching the image X with the maximum intensity guidance. The corresponding quantization error is then diffused in X and I, which is now the total intensity value of the current X, is updated accordingly. The iteration terminates when $I<0.5$. The details of each iteration are given as follows.

Step 1: Select a pixel for quantization

The current version of image X is divided into four non-overlapped sub-images of size $N/2\times N/2$ each. The total intensity value for each sub-image is calculated and the one of the highest total intensity value is selected. The selected sub-image is further divided into four sub-images for selection and so on. The division- and-selection process is repeated until the size of the sub-image is of size 1×1. This 1×1 sub-image defines a pixel and its coordinates, say (i_s,j_s), defines the location to where one should assign a white dot in output B. When there are more than one sub-images having the same maximum total intensity value during sub-image selection, one of them is randomly picked.

0-7803-9266-3/05/$20.00 ©2005 IEEE.

Step 2: Quantize and diffuse error

Once a pixel is selected, a white dot is assigned to the output bi-level image B by setting $b(i_s,j_s)=1$. The quantization error at $x(i_s,j_s)$ is then given as

$$e(i_s,j_s)=x(i_s,j_s)-1. \quad (2)$$

and is diffused to $x(i_s,j_s)$'s neighboring pixels with a 3×3 non-causal diffusion filter defined as

$$H_{interior}=\frac{1}{12}\begin{bmatrix}1 & 2 & 1\\ 2 & -12 & 2\\ 1 & 2 & 1\end{bmatrix} \quad (3)$$

To avoid error leakage, when the selected pixel is a side or corner pixel of the image, a modified diffusion filter such as H_{side} and H_{corner} is used instead, where

$$H_{side}=\frac{1}{8}\begin{bmatrix}0 & 0 & 0\\ 2 & -8 & 2\\ 1 & 2 & 1\end{bmatrix} \text{ and } H_{corner}=\frac{1}{5}\begin{bmatrix}0 & 0 & 0\\ 0 & -5 & 2\\ 0 & 2 & 1\end{bmatrix}.$$

Diffusion filters of larger support regions such as 5×5, 7×7 and 9×9 can also be used. One can read [4] for the details.

Step 3: Update X and I

After diffusing the error in step 2, X is updated and its total intensity value I should also be updated. This can be achieved with $I=I-1$.

Steps 1 to 3 are carried out sequentially in each iteration and they are repeated until $I<0.5$. The bi-level halftone image can then be finalized. Note that, by using this termination criterion, the absolute value of the overall intensity value difference between the output bi-level image and the given gray scale image are bounded by 0.5.

III. PROPOSED ALGORITHM

Similar to MED [4], the proposed algorithm is also an iterative algorithm. After computing the total intensity value of the given gray scale image X, which is the initial value of I, the given image X is partitioned into non-overlapped blocks of size $n\times n$ each and the total intensity value of each block is calculated. Let $I_B(k)$ be the total intensity value of the k-th block and M be the mean of the total intensity values of all blocks. In formulation, we have

$$M=(\frac{n}{N})^2\sum_{k=1}^{(N/n)^2} I_B(k)=\left(\frac{n}{N}\right)^2 I \quad (4)$$

Here, without loss of generality, we assume that N/n is an integer for formulating the proposed algorithm in a simpler form.

Note that, in this proposed iterative algorithm, X is updated after each iteration and its associated parameters such as I, M and $I_B(k)$ are also updated accordingly after an iteration.

In each iteration, the blocks whose total intensity values are larger than the current mean intensity value M are selected. For each selected block, a pixel is selected to assign a white dot. Quantization and error diffusion are then performed on the selected pixels of image X. $I_B(k)$ for all selected k's and M are then updated. The iteration stops until the total intensity value of the updated image X is below 0.5. The details are described as follows.

Step 1: Select image blocks for assigning white dots

The blocks whose total intensity values are larger than or equal to the current mean intensity value M as defined in Eq.(4) are selected. If the total number of the selected blocks is larger than the current total image intensity value I, the first m blocks with higher total intensity values will be chosen, where m is the rounded value of I. By using M as the threshold, it guarantees that white dots are assigned to the blocks of higher total intensity values first such that white dots are not over-assigned to the low intensity areas.

Step 2: Select a pixel for error diffusion in each selected block

For each selected block, a pixel is located to assign a white dot by following the steps 1 and 2 described in Section II. Similarly, a non-causal diffusion filter is used to avoid directional hysteresis. Note that the search is block-based instead of image-based in the proposed algorithm and hence the complexity is significantly reduced. Since the quantization error of a pixel is allowed to diffuse across the block boundary if the selected pixel is at the boundary of a block, blocking artifacts can be avoided to a certain extent.

Step 3: Update I, M and $I_B(k)$

The total block intensity values $I_B(k)$ and the total image intensity value I should be updated after error diffusion. Accordingly, the mean of the total intensity value of the blocks, M, should be re-calculated as well with eqn.(4) for the next iteration. In particular, updating these parameters can be achieved by $I=I-m$ and $M=I(n/N)^2$, where m is the total number of blocks selected in step 1 of the current iteration.

Repeat steps 1 to 3 for the next iteration until the total intensity value I is below 0.5. In the proposed algorithm, the error leakage is bounded by $\pm$ 0.5.

IV. SIMULATIONS

Simulations were carried out to evaluate the performance of the proposed algorithm in terms of implementation complexity and visual quality. Fig. 1 shows 15 512×512 gray-scale images used in the test. In the simulation, the block size n was set to be 16 in the realization of the proposed algorithm.

A. Computational complexity

For an image of size $N\times N$ with total intensity value I, totally *round*(*I*) white dots are assigned to the output bi-level image. Table 1 summarizes the total number of arithmetic operations required by MED and the proposed algorithm.

Fig. 1 Testing images (Refers as Image 1-15, from top-to-bottom and left-to-right)

Process		ADD	MUL	CMP
Initialization		(N^2-1)	0	0
Iter.	Step 1: Select pixel	0	0	$3\log_2 N$
	Step 2: Diffuse error	$z(\log_2 N+1)$	2	0
	Step 3: Update I	1	0	0

(a) MED [4]

Process		ADD	MUL	CMP
Initialization		(N^2-1)	1	0
Iter.	Step 1: Select blocks	0	0	$(N/n)^2$
	Step 2: Select pixels and diffuse error	$mz(\log_2 n+1)$	$2m$	$3m\log_2 n$
	Step 3: Update $I_B(k)$ and M	$mz+1$	1	0

(b) proposed algorithm

Table 1 Realization complexity

Size	64×64	128×128	256×256	512×512
MED	18×10^4	81×10^4	366×10^4	1629×10^4
Proposed	15×10^4	60×10^4	240×10^4	961×10^4
Ratio (=Ours/MED)	0.8394	0.7345	0.6540	0.5896

Table 2 Average total number of operations required for the valuated algorithms

In Table 1(b), parameter m is the total number of blocks selected in step 1 of a particular iteration. Parameter z is the number of neighboring pixels being affected in diffusing the quantization error of a pixel. It is equivalent to the number of non-zero elements of the diffusion filter used in the error diffusion. Accordingly, its value can be 9, 6 or 4 when $H_{interior}$, H_{side} and H_{corner} are used in our case. Since parameter m varies in each iteration and it depends on the nature of the given image, it is hard to formulate the overall computational complexity of the proposed algorithm. Simulations were carried out to count the operations required for halftoning the 15 test images. This was done to evaluate the complexity of MED and the proposed algorithm in real situation.

Table 2 shows the average number of operations required for the two algorithms to halftone images of different sizes. It shows that the average number of operations of the proposed algorithm is significantly reduced as compared with that of MED. On average, the proposed algorithm saved 40% of the operations used in MED to generate a 512×512 halftone image. The reduction is monotonic increasing with the image size.

B. Visual quality

The HVS-based measurement proposed in [1] was used to evaluate the visual quality of the proposed algorithm. This tool quantitatively measures how close a halftone image B to its original gray-scale image X is by measuring the mean squared error (MSE) between the HVS-filtered X and the HVS-filtered B. In particular, it is defined as

$$MSE_v = \left\| hvs(X, dis, dpi) - hvs(B, dis, dpi) \right\|^2 / N^2 \quad (5)$$

where $hvs(Y,dis,dpi)$ is the HVS filter function defined in [1]. Note that Y is the image of query, dis is the viewing distance in inches and dpi is the printer resolution. In our simulations, the viewing distance was fixed at 20 inches and printer resolutions of 600dpi and 1200dpi were considered.

Table 3 shows the visual distortion of the halftone images generated with different algorithms with respect to the original gray-scale images in terms of MSE_v. The images involved were of size 256×256 each. On average, the difference between the halftones produced with MED and those produced with the proposed algorithm is in the order of 10^{-5} no matter whether the printer resolution is 600dpi or 1200dpi. To a certain extent, it reflects that, in terms of MSE_v, the proposed algorithm can produce a halftone the visual quality of which is very close to that can be achieved by MED.

Img.	MSE_v $(\times10^{-3})$					
	600 dpi			1200dpi $(\times10^{-3})$		
	MED [4]	Ours	MED - Ours	MED [4]	Ours	MED - Ours
1	30.6638	30.6904	-0.0266	43.2991	43.3705	-0.0714
2	51.4678	51.4721	-0.0043	60.7124	60.7224	-0.0100
3	31.8694	31.8457	0.0237	41.3630	41.3524	0.0106
4	44.7648	44.7285	0.0363	64.0336	64.0565	-0.0229
5	54.1260	54.1854	-0.0594	63.2660	63.3663	-0.1003
6	43.2872	43.3142	-0.0270	54.7066	54.7149	-0.0083
7	45.2958	45.2219	0.0739	65.6964	65.6119	0.0845
8	35.0173	34.8668	0.1505	50.8251	50.5760	0.2491
9	24.8586	24.8316	0.0270	36.5069	36.4585	0.0484
10	38.6846	38.7048	-0.0202	54.1784	54.2626	-0.0842
11	42.3316	42.2476	0.0840	60.8101	60.6515	0.1586
12	27.4713	27.4378	0.0335	35.2178	35.1153	0.1025
13	45.0759	45.0700	0.0059	49.1231	49.1032	0.0199
14	36.5533	36.5138	0.0395	46.8408	46.8043	0.0365
15	65.1764	65.1670	0.0094	73.7522	73.7636	-0.0114
Avg.	**41.1096**	**41.0865**	**0.0231**	**53.3554**	**53.3287**	**0.0268**

Table 3 Visual quality of halftones produced with different algorithms in terms of MSE_v

To explore if the proposed block-based approach introduces more artifacts to the boundary region of a block,

block boundary pixels and block interior pixels of a HVS-filtered *B* were separated as shown in Figure 2. Their contributions to MSE_v are then separately evaluated. Table 4 shows the evaluation result. |Δ| is the difference between their contributions. One can see that the difference between their contributions is actually very little. In fact, the contribution of the interior pixels is even higher. This minor difference may be explained by that error can be diffused across the boundary of a block in the halftoning process. Accordingly, the diffusion of error does not blocked by the block boundary and hence error does not accumulate at the boundary region.

Figure 3 shows some halftoning outputs for subjective evaluation. As shown in Fig. 3, the halftoning outputs of the two evaluated algorithms are more or less the same. Details can be preserved and the contrast can be kept equally in both outputs. Theoretically, since both algorithms exploit a non-causal diffusion filter and a non-deterministic scanning path, artifacts caused by directional hypothesis can be eliminated in their outputs.

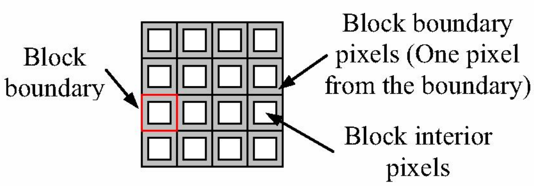


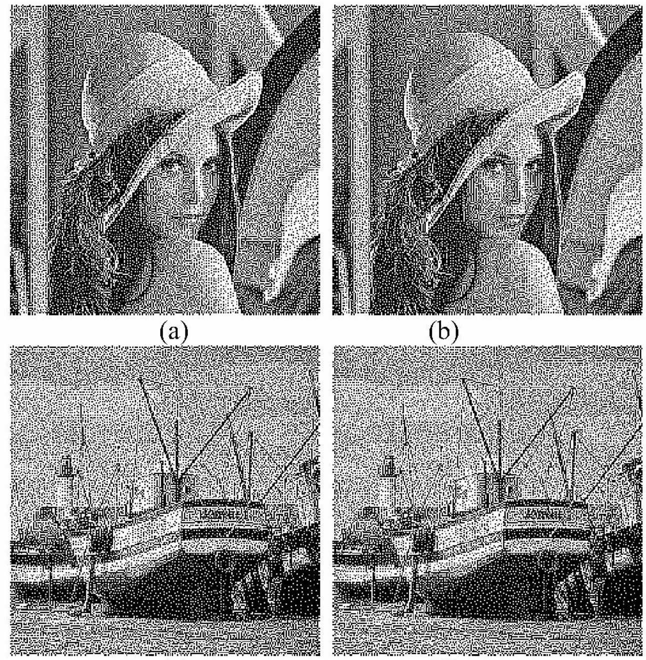
Fig. 2 Block boundary pixels and block interior pixels in a HVS-filtered image

Img.	MSE_v ($\times 10^{-3}$)					
	600 dpi			1200dpi ($\times 10^{-3}$)		
	Boundary pixels	Interior pixels	|Δ|	Boundary pixels	Interior pixels	|Δ|
1	30.16	30.81	0.66	41.85	43.65	1.80
2	50.57	51.72	1.15	59.14	61.22	2.08
3	31.31	32.01	0.70	39.84	41.80	1.96
4	44.28	44.92	0.64	63.07	64.39	1.33
5	53.49	54.35	0.86	61.83	63.79	1.96
6	42.77	43.44	0.67	53.98	54.90	0.92
7	44.10	45.67	1.57	62.84	66.64	3.80
8	34.56	35.01	0.45	49.51	50.99	1.48
9	24.30	24.99	0.69	35.13	36.88	1.75
10	38.40	38.82	0.41	53.59	54.46	0.86
11	41.41	42.51	1.10	58.91	61.26	2.35
12	27.15	27.59	0.45	34.54	35.42	0.88
13	44.26	45.35	1.09	47.53	49.63	2.11
14	36.46	36.63	0.17	46.77	46.93	0.16
15	63.78	65.53	1.75	69.84	74.87	5.03
Avg.	**40.47**	**41.29**	**0.82**	**51.89**	**53.79**	**1.90**

Table 4 Contributions of pixels of different nature to MSE_v when the proposed algorithm is used.

V. CONCLUSIONS

In this paper, a fast block-based MED halftoning algorithm is proposed. This algorithm reduces the searching effort to select a location for assigning a white dot by dividing an image into blocks such that more than one white dots can be assigned in each iteration loop. Simulation results show that the proposed algorithm can significantly reduce the complexity without degrading the visual quality of the output as compared with MED[4]. Since blocks can be separately processed in parallel, the proposed algorithm supports parallel processing. In other words, processing time can be further squeezed and real-time processing is possible.



(a) (b) (c) (d)

Fig. 3 Halftone images of size 256×256 produced by (a)&(c) MED and (b)&(d) the proposed algorithm.

VI. ACKNOWLEDGEMENT

This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (PolyU 5205/04E).

VII. REFERENCES

[1] D. L. Lau, G. R. "Arce, "Modern Digital Halftoning", Marcel Dekker, Inc., 2001.

[2] M. Mese and P. P. Vaidyanathan, "Recent advances in digital halftoning and inverse halftoning methods", *IEEE Trans. Circuits and Systems-I*, pp.790-805, Jun. 2002.

[3] 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*, pp.36-37, 1975.

[4] I. Katsavounidis and C. C. J. Kuo, "A multiscale error diffusion technique for digital halftoning", *IEEE Trans. Image Processing*, pp.483-490, Mar. 1997.

[5] Y. H. Chan and S. M. Cheung, "Feature- preserving multiscale error diffusion for digital halftoning", *Journal of Electronic Imaging*, pp.639-645, Jul. 2004.