

4-ROW SERPENTINE TONE DEPENDENT FAST ERROR DIFFUSION

Y. Mao, L. Abello, U. Sarkar, R. Ulichney and J. Allebach

School of Electrical and Computer Engineering, Purdue University

Abstract: Error diffusion is a popular technique widely used in desktop printers, especially inkjet printers. However, since the conventional error diffusion is computed in raster order, it produces worm artifacts in the highlights and shadows. In addition, as a serial algorithm, it limits the efficiency and flexibility of hardware implementations. To address these two issues, we propose a novel serpentine based error diffusion algorithm that uses tone dependent error weights and thresholds. We also propose an expanded error weight location matrix to improve the halftone quality in the extreme tones. With this new algorithm, we achieve better halftones comparing to the original tone dependent fast error diffusion, especially in the quarter tones.

Index Terms: halftoning, modified tone dependent fast error diffusion, parallel implementation

1 Introduction

Digital halftoning is a method of creating the illusion of continuous-tone output through the use of a series of dots arranged differently in size or in spacing. Halftoning allows one to simulate various shades of color with one or two colors, so it is an widely used technique in rendering devices that are only capable of producing limited number of tone levels, for example printers and some displays.

According to the level of computational complexity, halftoning algorithms can be classified into three general categories: screening, error diffusion, and search-based methods. Since error diffusion renders better detail than screening while maintaining lower cost than search-based methods, it is the most popular algorithm for marking engine technologies that can stably render isolated dots, such as inkjet. In this paper, we will focus on error diffusion.

As originally proposed by Floyd and Steinberg [1], error diffusion is a neighborhood operation that moves through the input image in a raster order, quantizing each pixel in the scan line, and feeding the error ahead to the neighboring pixels that have not yet been binarized. Despite excellent detail rendition, error diffusion sometimes creates worm-like patterns and visible structures. To solve these problems, a number of derivations and modifications of error diffusion were developed in previous research, including use of alternative scan paths [2]–[4], threshold modulation [5]–[9], variable weights [4], [10]–[12], and tone dependent parameters [13], [14]. Reference [15] is of particular note, as it provides an excellent summary of recent methods based on directly training the weights to match a desired blue noise spectral characteristic, and proposes an improvement to this approach. It also incorporates the training of the thresholds to eliminate edge sharpening.

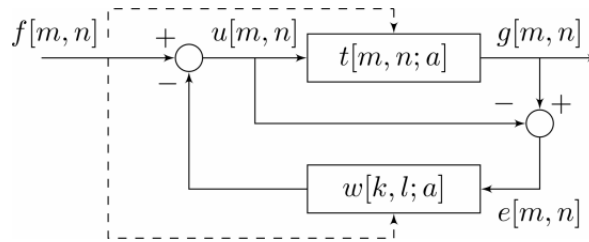


Fig. 1: Tone dependent fast error diffusion system.

Aside from generating visually unpleasant textures, another disadvantage of error diffusion is lack of locality. This means that when either a conventional raster or serpentine scan path [4] is used, we

are not allowed to process a pixel until all pixels in its preceding scan line have been quantized. As a consequence, hardware must store the information associated with the states of the pixels that are spatially far away, which is inefficient. With Peano scan [2], a pioneer of parallel scan path, however, the output quality of error diffusion is not satisfactory. In fact, customers in the printing market base their judgment on both print quality and implementation efficiency. Therefore, it is extremely beneficial to enable the hardware to decide the binary output locally without losing quality. In this regard, we design a novel 4-row serpentine scan path. The novelty of our approach lies in using a compound of conventional raster and serpentine scan patterns, which greatly enhances hardware efficiency. Other prior work that incorporated the concept of a serpentine raster with novel error diffusion architecture include [16], [17].

In this paper, we apply the 4-row serpentine scan path with the tone dependent fast error diffusion (TDFED) [18] algorithm. To reduce worm-like textures, a tone-dependent 4-weight location matrix that diffuses errors further back along the next line is designed. To further refine the halftone outputs, the weights and thresholds values for each gray level are optimized in an offline training process based on a visual cost function developed in [19]. However, the primary focus of this paper is the development of a new error diffusion architecture that is suitable for efficient hardware implementation, rather than methods and cost functions for optimization of the weights and thresholds, which is the primary focus of [15]. In fact, the concepts introduced in this paper could also be deployed with weights and thresholds optimized according to the methods introduced in [15]. The rest of the paper is organized as follows: Section 2 demonstrates the 4-row serpentine TDFED algorithm. Section 3 discusses the experimental results. Section 4 draws the conclusions.

2 4-Row Serpentine Tone Dependent Fast Error Diffusion

We will start by providing an overview of TDFED in Section 2.1 and then present the details of the scan path. Section 2.3 will illustrate the expanded error location matrix. Lastly, Sec. 2.4 will discuss the training process.

2.1 Overview of Tone Dependent Fast Error Diffusion

Since we are presenting an error diffusion algorithm that is designed for monochrome printing devices, the pixel value is represented in units of absorbance $0 \leq a \leq 1$, where 0 corresponds to white, and 1 corresponds to black.

Figure 1 illustrates the block diagram of the TDFED system. In this figure, $f[m, n] = a$ is the pixel absorbance of the continuous-tone image, $u[m, n]$ is the updated pixel value, and $g[m, n]$ is the binary output. Unlike Floyd Steinberg error diffusion, the thresholds $t[m, n; a]$ and error weights $w[k, l; a]$ of TDFED depend on the input absorbance, where k, l are the relative position indices indicating the location of the neighboring pixels of $f[m, n]$. The neighboring pixels of $f[m, n]$ are defined by Floyd and Steinberg to be on its right, lower right, below, and lower left, assuming the image is scanned from left to right in a raster order. Since TDFED moves through the input image in a serpentine raster order, a mirror image of the weight location is adopted when scanning from right to left.

The binary output of the system is determined by thresh-olding the updated pixel value:

$$g[m, n] = \begin{cases} 1, & \text{if } u[m, n] \geq t[m, n; a], \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

The updated continuous-tone pixel value $u[m, n]$ is computed as:

$$u[m + k, n + l] \leftarrow u[m + k, n + l] - w[k, l; a] \cdot e[m, n], \quad (2)$$

where $e[m, n]$ is the quantization error. It is computed as:

$$e[m, n] = g[m, n] - u[m, n], \quad (3)$$

and the weights $w[k, l; a]$ satisfy $\sum_{k,l} w[k, l; a] = 1$ to preserve the average local tone. To eliminate the checkboard patterns in the midtone, the threshold matrix $t[m, n; a]$ of TDFED is defined based on a halftone pattern $p[m, n; 0.5]$ generated by DBS with period 128×128 for absorptance 0.5 as:

$$t[m, n; a] = \begin{cases} t_u(a), & \text{if } p[m, n; 0.5] = 0, \\ t_l(a), & \text{otherwise.} \end{cases} \quad (4)$$

where $t_u(a)$ and $t_l(a)$ are tone dependent parameters that serve as upper and lower thresholds satisfying $t_l(a) \leq t_u(a)$. Substituting (5) into (1) yields:

$$g[m, n] = \begin{cases} 1, & \text{if } u[m, n] \geq t_u(a), \\ 0, & \text{if } u[m, n] \leq t_l(a), \\ p[m, n; 0.5], & \text{otherwise.} \end{cases} \quad (5)$$

In order to reduce the computational complexity, Li and Allebach chose the optimal filters for tone levels higher than 127/256 according to $t_u(a) = t_u(1 - a)$ and $w[k, l; a] = w[k, l; 1 - a]$. The same strategy is used for all experiments in this paper.

2.2 4-Row Serpentine Scan Path

It has been shown in [4] that implementing error diffusion in serpentine order effectively reduces the worm artifacts in the extreme gray levels. However, serpentine error diffusion is intrinsically a serial process, we must finish an entire scan line before moving on to the next one. This limits the parallelism and locality required by hardware implementations. Thus, it is necessary to mix the serpentine and raster scan together to solve the problem.

In 4-row serpentine TDFED, every 4 rows of an input image are grouped as a swath. As depicted by the arrows in Fig. 2, four scan lines in the same swath are processed in one direction and the consecutive four lines are processed in the opposite direction. By doing so, the serpentine property is preserved between adjacent swaths.

To be more specific, inside each swath, error diffusion starts from the first pixel of the first row, and then it advances rightward along the first row in raster scan order until the next row is activated. The activation condition is that the binarization of d pixels in the preceding row has been finished, where d is defined as the delay between the processing of sequential lines in the 4-row swath. It can be adapted according to the need of the actual hardware architecture. Error diffusion travels back and forth across the activated scan lines. After the last pixel of the fourth row has been visited, the scan path then traverses leftward to process the next swath in the same manner described above. Figure 2 is an instance of the 4-row serpentine scan pattern in which d equals 3 pixels. Each entry of the matrix represents a pixel of the continuous-tone input image, and the number indicates the order in which the pixel at that location is processed.

1	2	3	4	6	8	10	13	16	19	23	27
5	7	9	11	14	17	20	24	28	31	34	37
12	15	18	21	25	29	32	35	38	40	42	44
22	26	30	33	36	39	41	43	45	46	47	48
75	71	67	64	61	58	56	54	52	51	50	49
85	82	79	76	72	68	65	62	59	57	55	53
92	90	88	86	83	80	77	73	69	66	63	60
96	95	94	93	91	89	87	84	81	78	74	70
...

Fig. 2: Scan path of 4-row serpentine scan with 3 pixel delay

2.3 Error Weight Location Matrix

To reduce worm-like patterns with a conventional raster scan error diffusion, randomized weights [4] and a low frequency modulated threshold matrix [6] have been proposed. Unfortunately, these approaches introduce noise to the halftone image. Jarvis et al and Stucki [20] proposed a larger set of error weights with 24 terms, which requires heavy computation. Shiau and Fan [21] moved the $1/16$ term in Floyd Steinberg weights from location $(1, 1)$ to $(-2, 1)$. Li and Allebach [18] developed a set of wider matrices that diffuse the errors further back. Our approach is based on [18].

As it can be seen from Fig. 4(a), the 4-row serpentine scan pattern does not produce long diagonal structures but generates lots of short diagonal worms in the highlights and shadows. Therefore, we need to spread the quantization error over a wider region in the problematic gray levels to disperse the worms. In 4-row serpentine TDFED, diffusing the errors to a further location requires an increase in the delay. We explored delays of 2, 3, and 6 pixels, which allow an increasingly larger spatial spread in the tone-dependent weight location matrix. To ensure minimum computations, we chose to use 4 non-zero weights as originally described in Floyd and Steinberg error diffusion. The matrix set designed for 3-pixel delay is presented in Table 1. The weights allocation and input partition are determined empirically. The values of the weights and thresholds are obtained using a search-based method, which we will discuss in the following section.

			*	w_1
w_4		w_3	w_2	

$$\frac{1}{255} \leq a \leq \frac{63}{255}, \frac{192}{255} \leq a \leq \frac{254}{255}$$

	*	w_1
w_4	w_3	w_2

$$\frac{64}{255} \leq a \leq \frac{191}{255}$$

	*	
	w_2	

$$a = 0, a = 1$$

Table 1: Expanded error weight location matrices designed for 4-row serpentine TDFED with 3 pixel delay. The asterisk sign denotes the current pixel.

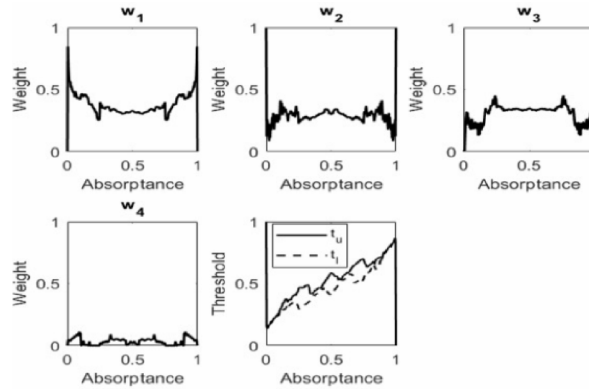


Fig. 3: Optimal tone dependent weights and thresholds for 4-row serpentine scan with 3 pixel delay

2.4 Training System for Tdfed Parameters

Although the expanded weight location matrix reduces worm artifacts, if the weights and thresholds are not properly designed, there will be correlated patterns and non-homogeneous textures in the output image. Thus, they must be optimized to achieve the best visual quality. We propose to use four non-zero error weights and two thresholds. There are only 4 degrees of freedom due to the constraints $\sum_{k,l} w[k, l; a] = 1$ and $t_u(a) + t_l(a) = 1$. We choose to optimize $w[0, 1; a]$, $w[1, 0; a]$, $w[1, 1; a]$, and $t_u(a)$. Generally speaking, the TDFED training system searches for the optimal parameters by minimizing a cost function ε .

Li and Allebach [18] used two different cost functions depending on the tone level. For the extreme gray levels, the perceived mean squared error between the constant valued continuous-tone patch and a TDFED halftone is minimized based on Nasanen's HVS model [22]. For the midtones, the total squared error between the power spectra of the halftone patch generated by TDFED and by DBS is minimized, which is given by:

$$\varepsilon_{Li} = \sum_u \sum_v (\bar{G}_{DBS}[u, v; a] - \bar{G}_{TDFED}[u, v; a])^2, \quad (6)$$

where $\bar{G}_{TDFED}[u, v; a]$ and $\bar{G}_{DBS}[u, v; a]$ represent the average magnitude of the 2D Fourier amplitude spectra obtained from TDFED and DBS halftone patches, respectively.

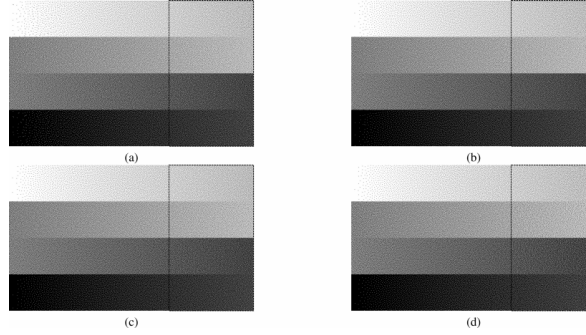


Fig. 4: Halftones of a folded ramp image generated by: (a) 4-Row serpentine TDFED with 1 pixel delay, (b) 4-row serpentine TDFED with 3 pixel delay, (c) 4-row serpentine TDFED with 6 pixel delay, (d) the original 1-row serpentine tdfed. It is recommended to zoom in on the figure so that individual pixels are clearly displayed and to view from a sufficient distance at which these individual pixels are not visually resolved.

Chang and Allebach [16] presented a cost function with a normalization as:

$$\varepsilon_{Chang} = \frac{\sum_u \sum_v (\bar{G}_{DBS}[u, v; a] - \bar{G}_{TDFED}[u, v; a])^2}{\bar{G}_{DBS}[u, v; a]^2}. \quad (7)$$

Han and Allebach [19] further modified the cost function by adding the power spectra of TDFED to the denominator:

$$\varepsilon_{Han} = \frac{\sum_u \sum_v (\bar{G}_{DBS}[u, v; a] - \bar{G}_{TDFED}[u, v; a])^2}{(\bar{G}_{DBS}[u, v; a] + \bar{G}_{TDFED}[u, v; a])^2}. \quad (8)$$

We concluded from our experiments that Han and Allebach's cost function yields the most reliable results for all levels. Thus, their cost function is adopted as the error metric.

As for the search strategy, we compared the performance of pattern search [23] with the downhill search [18], and established that downhill search is better suited to our application. The optimized weights and thresholds of 4-row serpentine TDFED with 3 pixel delay are shown in Fig. 3.

3 Experimental Results

Figures 4(a)-(c) show the result of 4-row serpentine TDFED with 1 pixel, 3 pixel, and 6 pixel delay, respectively. Figure 4(d) is the original 1-row serpentine TDFED result. It can be observed that in the

highlights and shadows, both 3-pixel and 6-pixel delay 4-row serpentine TDFED significantly reduce the short diagonal structures seen in (a) and are comparable to 1 row serpentine TDFED. Moreover, Fig. 4(d) contains some vertical veining patterns in the region marked by the dotted square box. However, all three 4-row halftones are very smooth and homogeneous in that area.

4 Conclusions

With a modest delay value, 4-row serpentine TDFED can achieve essentially the same or better image quality than that provided by the original 1-row serpentine TDFED, except perhaps in the extreme gray levels. Besides, it will also boost efficiency and reduce memory cost in some hardware implementations.