

译文 1

作者: Yafei Mao; Lluis Abello; Utpal Sarkar; Robert Ulichney; Jan Allebach

出处: IEEE International Conference on Image Processing. 2018:3973-3977.

四行蛇形色调自适应快速误差扩散算法

摘要: 误差扩散是桌面打印机（尤其是喷墨打印机）中广泛应用的技术。然而，传统误差扩散采用光栅扫描顺序计算，会在高光和阴影区域产生“蠕虫伪影”；同时，其串行执行的特性限制了硬件实现的效率与灵活性。为解决这两个问题，本文提出一种基于蛇形扫描的新型误差扩散算法，该算法采用色调自适应的误差权重与阈值，并设计了扩展误差权重位置矩阵以提升极端色调下的半色调质量。与原始色调自适应快速误差扩散算法相比，本文算法能生成更优的半色调图像，尤其在四分之一色调区域表现突出。

关键词: 半色调、改进型色调自适应快速误差扩散、并行实现

1 引言

数字半色调技术通过调整点的大小或间距，利用有限的色调层级模拟连续色调的输出效果，广泛应用于打印机、显示器等仅支持有限色调的设备中。

根据计算复杂度，半色调算法可分为三类：加网算法、误差扩散算法和搜索型算法。误差扩散的细节呈现效果优于加网算法，同时成本低于搜索型算法，因此成为喷墨打印机等能稳定渲染孤立点的标记引擎的主流算法。本文将聚焦于误差扩散技术。

弗洛伊德-斯坦伯格（Floyd and Steinberg）提出 的传统误差扩散是一种邻域操作：按光栅顺序遍历输入图像，量化扫描线中的每个像素，并将量化误差传递给后续未二值化的邻域像素。尽管细节呈现效果出色，但误差扩散有时会产生蠕虫状图案和可见结构。为解决这些问题，已有研究提出了多种改进方案，包括采用替代扫描路径 [2]-[4]、阈值调制 [5]-[9]、可变权重 [4]、[10]-[12] 及色调自适应参数 [13]、[14] 等。文献 [15] 值得特别关注：它总结了基于权重训练以匹配蓝噪声频谱特性的近期方法，并提出了改进方案，同时将阈值训练纳入其中以消除边缘锐化。

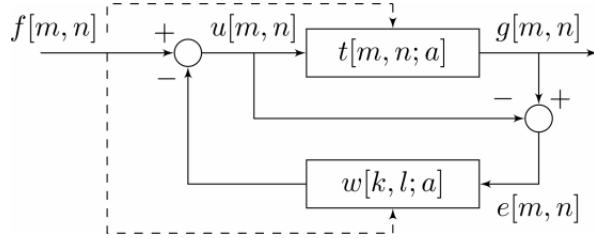


图 1: 色调自适应快速误差扩散系统。

除了生成视觉上不愉悦的纹理外，误差扩散的另一缺点是缺乏局部性。这意味着无论采用传统光栅扫描还是蛇形扫描路径 [4]，必须等前一条扫描线的所有像素完成量化后，才能处理当前像素。因此，硬件需存储空间上较远的像素状态信息，效率较低。而皮亚诺扫描 [2]（并行扫描路径的先驱）虽能并行处理，但误差扩散的输出质量无法令人满意。实际上，打印市场的客户同时关注打印质量与实现效率，因此在不损失质量的前提下实现硬件的局部二值输出决策具有重要意义。为此，本文设计了一种新型四行蛇形扫描路径，其创新点在于结合传统光栅与蛇形扫描模式，大幅提升了硬件效率。其他将蛇形光栅与新型误差扩散架构结合的相关工作包括 [16]、[17]。

本文将四行蛇形扫描路径与色调自适应快速误差扩散 (TDFED) [18] 算法结合：为减少蠕虫状纹理，设计了色调自适应的四权重位置矩阵，将误差沿下一行进一步向后扩散；为进一步优化半色调输出，基于文献 [19] 提出的视觉代价函数，通过离线训练过程优化每个灰度级的权重与阈值。但本文的核心是开发适用于高效硬件实现的新型误差扩散架构，而非权重与阈值的优化方法及代价函数（这是文献 [15] 的核心内容）。实际上，本文提出的概念也可与文献 [15] 中的权重与阈值优化方法结合使用。

本文后续结构如下：第 2 节介绍四行蛇形色调自适应快速误差扩散算法；第 3 节讨论实验结果；第 4 节总结全文。

2 四行蛇形色调自适应快速误差扩散算法

本节首先在 2.1 小节概述色调自适应快速误差扩散算法，随后介绍扫描路径的细节，2.3 小节阐述扩展误差位置矩阵，最后在 2.4 小节讨论训练过程。

2.1 色调自适应快速误差扩散概述

由于本文提出的误差扩散算法面向单色打印设备，像素值以吸收率表示 ($0 \leq a \leq 1$)，其中 0 对应白色，1 对应黑色。

图 1 展示了 TDFED 系统的框图： $f[m, n] = a$ 是连续色调图像的像素吸收率， $u[m, n]$ 是更新后的像素值， $g[m, n]$ 是二值输出。与弗洛伊德-斯坦伯格误差扩散不同，

TDFED 的阈值 $t[m, n; a]$ 和误差权重 $w[k, l; a]$ 依赖于输入吸收率，其中 k, l 是相对位置索引，用于表示 $f[m, n]$ 的邻域像素位置。弗洛伊德-斯坦伯格定义的邻域像素为当前像素的右侧、右下、下方及左下（假设图像按光栅顺序从左到右扫描）；由于 TDFED 采用蛇形光栅顺序遍历图像，当从右到左扫描时，权重位置会采用镜像形式。

系统的二值输出通过对更新后的像素值进行阈值化得到：

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

更新后的连续色调像素值 $u[m, n]$ 计算如下：

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

其中 $e[m, n]$ 是量化误差，计算方式为：

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

且权重 $w[k, l; a]$ 满足 $\sum_{k,l} w[k, l; a] = 1$ ，以保持局部平均色调。为消除中间调的棋盘格图案，TDFED 的阈值矩阵 $t[m, n; a]$ 基于吸收率为 0.5 时由 DBS 生成的半色调图案 $p[m, n; 0.5]$ （周期为 128×128 ）定义：

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

其中 $t_u(a)$ 和 $t_l(a)$ 是色调自适应参数，分别作为上阈值和下阈值，满足 $t_l(a) \leq t_u(a)$ 。将式 (5) 代入式 (1) 可得：

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

为降低计算复杂度，Li 和 Allebach 选择当色调层级高于 $127/256$ 时，最优滤波器满足 $t_u(a) = t_u(1 - a)$ 且 $w[k, l; a] = w[k, l; 1 - a]$ 。本文所有实验均采用此策略。

2.2 四行蛇形扫描路径

文献 [4] 表明，采用蛇形顺序实现误差扩散可有效减少极端灰度级的蠕虫伪影，但蛇形误差扩散本质上是串行过程：必须完成整条扫描线的处理后才能进入下一条，

这限制了硬件实现所需的并行性与局部性。因此，需将蛇形与光栅扫描结合以解决该问题。

在四行蛇形 TDFED 中，输入图像的每 4 行被分为一个“条带（swath）”。如图 2 中的箭头所示，同一条带内的 4 条扫描线按同一方向处理，相邻条带则按相反方向处理，从而在条带间保留蛇形特性。

具体而言，在每条带内，误差扩散从第一行的第一个像素开始，按光栅顺序沿第一行向右推进，直到下一行被激活。激活条件是前一行中 d 个像素完成二值化，其中 d 定义为四行条带中连续行处理的延迟，可根据实际硬件架构的需求调整。误差扩散在激活的扫描线间来回遍历；当第四行的最后一个像素处理完成后，扫描路径向左移动，按上述方式处理下一个条带。图 2 是 $d = 3$ 像素时四行蛇形扫描模式的实例：矩阵中的每个元素代表连续色调输入图像的一个像素，数字表示该位置像素的处理顺序。

| | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 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 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

图 2：具有 3 像素延迟的四行蛇形扫描路径

2.3 误差权重位置矩阵

为减少传统光栅扫描误差扩散的蠕虫状图案，已有研究提出了随机权重 [4] 和低频调制阈值矩阵 [6]，但这些方法会给半色调图像引入噪声。Jarvis 等人和 Stucki[20] 提出了包含 24 项的更大误差权重集，但计算量较大；Shiau 和 Fan[21] 将弗洛伊德-斯坦伯格权重中的 1/16 项从位置 (1, 1) 移至 (-2, 1)；Li 和 Allebach[18] 开发了一组更宽的矩阵，将误差进一步向后扩散。本文方法基于文献 [18]。

从图 4(a) 可以看出，四行蛇形扫描模式不会产生长对角线结构，但会在高光和阴影区域生成大量短对角线蠕虫，因此需要在这些问题灰度级中将量化误差扩散到更宽的区域以分散蠕虫。在四行蛇形 TDFED 中，将误差扩散到更远位置需要增加延迟。本文测试了 2、3、6 像素的延迟，这些延迟允许色调自适应权重位置矩阵的空间

扩散范围逐渐增大。为确保计算量最小，本文选择弗洛伊德-斯坦伯格误差扩散中最初描述的 4 个非零权重。表 1 展示了为 3 像素延迟设计的矩阵集，权重分配和输入分区由经验确定，权重与阈值的值通过基于搜索的方法获得（下一小节将讨论）。

| | | | | |
|-------|--|-------|-------|-----------------------------------------------------------------------------------------|
| | | | * | 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 |
| | | | | |
| | | | | $a = 0, a = 1$ |

表 1：为具有 3 像素延迟的四行蛇形 TDFED 设计的扩展误差权重位置矩阵。星号表示当前像素。

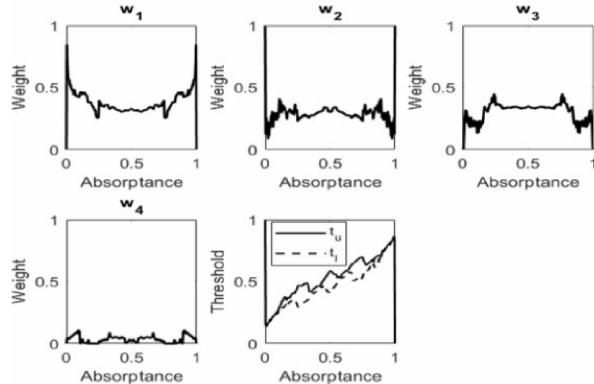


图 3：具有 3 像素延迟的四行蛇形扫描的最优色调自适应权重与阈值

2.4 TDFED 参数的训练系统

尽管扩展权重位置矩阵可减少蠕虫伪影，但如果权重与阈值设计不当，输出图像会出现相关图案和非均匀纹理，因此必须对其进行优化以获得最佳视觉质量。本文采用 4 个非零误差权重和 2 个阈值，由于约束条件 $\sum_{k,l} w[k, l; a] = 1$ 和 $t_u(a) + t_l(a) = 1$ ，仅存在 4 个自由度，因此选择优化 $w[0, 1; a]$ 、 $w[1, 0; a]$ 、 $w[1, 1; a]$ 和 $t_u(a)$ 。一般而言，TDFED 训练系统通过最小化代价函数 ε 来搜索最优参数。

Li 和 Allebach[18] 根据色调层级使用两种不同的代价函数：对于极端灰度级，基于 Nasanen 的人类视觉系统（HVS）模型 [22]，最小化恒定值连续色调块与 TDFED

半色调块之间的感知均方误差；对于中间调，最小化 TDFED 生成的半色调块与 DBS 生成的半色调块的功率谱之间的总平方误差，即：

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

其中 $\bar{G}_{TDFED}[u, v; a]$ 和 $\bar{G}_{DBS}[u, v; a]$ 分别表示从 TDFED 和 DBS 半色调块获得的二维傅里叶振幅谱的平均值。

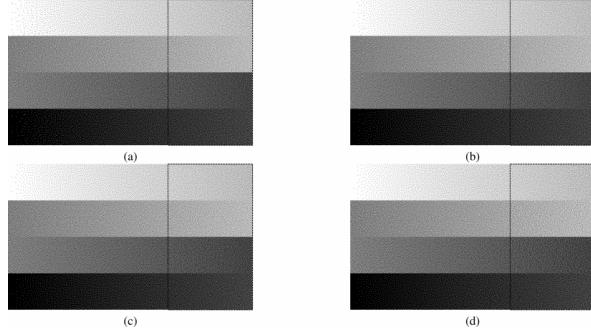


图 4：折叠斜坡图像的半色调结果：(a) 具有 1 像素延迟的四行蛇形 TDFED；(b) 具有 3 像素延迟的四行蛇形 TDFED；(c) 具有 6 像素延迟的四行蛇形 TDFED；(d) 原始一行蛇形 TDFED。建议放大图像以清晰显示单个像素，并从足够远的距离观察（此时单个像素无法被视觉分辨）。

Chang 和 Allebach[16] 提出了带归一化的代价函数：

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

Han 和 Allebach[19] 进一步修改了代价函数，在分母中加入 TDFED 的功率谱：

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

实验表明，Han 和 Allebach 的代价函数在所有层级均能产生最可靠的结果，因此本文将其作为误差度量。

在搜索策略方面，本文比较了模式搜索 [23] 与 downhill 搜索 [18] 的性能，发现 downhill 搜索更适合本文应用。图 3 展示了具有 3 像素延迟的四行蛇形 TDFED 的优化权重与阈值。

3 实验结果

图 4(a)-(c) 分别展示了具有 1 像素、3 像素和 6 像素延迟的四行蛇形 TDFED 的结果，图 4(d) 是原始一行蛇形 TDFED 的结果。可以观察到，在高光和阴影区域，3 像素和 6 像素延迟的四行蛇形 TDFED 显著减少了 (a) 中出现的短对角线结构，与一行蛇形 TDFED 的效果相当；此外，图 4(d) 中虚线方框标记的区域存在一些垂直纹理，而所有三种四行半色调图像在该区域均非常平滑均匀。

4 结论

通过适度的延迟值，四行蛇形 TDFED 的图像质量与原始一行蛇形 TDFED 基本相当或更优（极端灰度级可能除外）；同时，它还能提升部分硬件实现的效率并降低内存成本。