

# 基于离散 Sibson 插值的反半色调算法

王 磊,李学庆

(山东大学计算机科学与技术学院,山东济南 250101)

**摘 要:** 本文提出了一种基于离散 Sibson 插值的盲反半色调方法.算法建立在统一的离散 Voronoi 图框架之上,通过自适应 Voronoi 区域估计得到种子点的灰度值,然后通过离散 Sibson 插值将传统的反半色调问题转化为插值问题并计算获取连续色调图像.算法不需要知道任何先验知识,因此算法具有更广的适用性,可以被应用于任意方法生成的半色调图像的反半色调处理.在效果处理方面,算法能够很好的保留原始图像的信息,同时能够更好的去除平滑区域的噪声.

**关键词:** 半色调;反半色调;误差扩散;离散 Voronoi 图;离散 Sibson 插值

**中图分类号:** TP391 **文献标识码:** A **文章编号:** 0372-2112 (2011) 01-0070-06

## Inverse Halftoning Based on Discrete Sibson Interpolation

WANG Lei, LI Xue-qing

(College of Computer Science and Technology, Shandong University, Jinan, Shandong 250101, China)

**Abstract:** We proposed a blind inverse halftoning algorithm with discrete Sibson interpolation. In our algorithm, a unified framework based on discrete Voronoi diagram is used to recover continuous-tone (contone) images from halftone images. At first, the halftone dots' intensity is estimated by the corresponding Voronoi diagram information. We further perform the discrete Sibson interpolation based on discrete Voronoi diagram to get the contone image. This algorithm is universal to all types of halftone images since it is independent of any prior information. Compared with existed algorithms, this algorithm is able to remove artifacts in both sparse and dense halftone dots regions and produces visually pleasant contone images.

**Key words:** digital halftoning; inverse halftoning; error diffusion; discrete Voronoi diagram; discrete Sibson interpolation

## 1 引言

半色调技术是一种将连续色调图像转化为二值图像的技术,它被广泛应用在印刷、传真、图像传输等相关应用,例如报纸、书籍、杂志的印刷过程中都广泛使用了半色调技术<sup>[1,2,6]</sup>.但是,半色调图像存在不易于进行操作的缺点,常用的图像处理技术包括缩放、增强和恢复等都难以在半色调图像上进行应用.为了能够对半色调图像进行处理,需要将半色调图像转化为连续色调图像,该转化过程被称为反半色调技术.由于半色调处理过程中使用的方法、模型的多样化,反半色调技术不存在单一的模型和方法,其转化过程属于一对多的映射关系.在本文中,一种基于离散 Voronoi 图的 Sibson 插值方法<sup>[3]</sup>被应用到反半色调的计算过程中,该算法能够适用于不同方法生成的半色调图像.

对于半色调方法来说,误差扩散<sup>[1,2,4,15]</sup>和点扩散

方法<sup>[18]</sup>是其中两种常用的传统方法.其中,依据误差扩散核函数的定义差别,误差扩散方法又分为 Floyd 方法<sup>[1]</sup>,Jarvis 方法<sup>[2]</sup>和 Ostromoukhov 方法<sup>[4]</sup>.依据不同的半色调方法,研究人员也提出了很多不同的反半色调方法.依据方法的策略不同,它基本上被分为三类.第一类方法将反半色调过程看作是去噪声的过程,通过对半色调过程中产生的高频噪声进行滤波而得到连续色调图像.其中,最简单的方法是直接对半色调图像进行低通滤波,但是该方法会模糊掉图像中的高频信息.另外,该类方法还包括了很多改进的方法,包括 Sigma 滤波、快速盲反半色调<sup>[5]</sup>、基于小波的方法<sup>[7]</sup>、快速梯度滤波<sup>[8]</sup>等.该类方法一般简单、快速,但是由于滤波方法本质上为低通滤波器,因此生成的反半色调图像往往会边缘比较模糊.

第二类方法将反半色调过程看作是图像重建问题,大量的图像重建方法被应用到反半色调的处理过程中.

其中包括基于最大后验概率(MAP)方法<sup>[9]</sup>和凸集投影方法(POCS)<sup>[10]</sup>. 该类重建方法的反半色调结果具有比较好的视觉效果. 但是, 此类方法只适用于相关的误差扩散半色调方法, 同时该类方法需要进行大量迭代运算从而需要大量的运算时间. 第三类方法为基于机器学习的方法, 通过训练集合建立半色调图像和连续色调图像之间的对应关系, 然后通过映射关系来估计给定的半色调图像所对应的连续色调图像. 其中, 基于查找表(LUT)<sup>[11]</sup>和决策树学习的方法<sup>[12]</sup>属于这一类别. 尤其针对基于查找表的反半色调方法, 目前有大量的工作对其效率和恢复效果进行了改进, 比如说基于树形结构的查找表方法<sup>[14]</sup>和边缘增强的查找表方法<sup>[13]</sup>. 该类方法的优点在于可以适用于不同的半色调方法得到的半色调图像. 但是, 该类方法的局限性在于需要大量的训练集合进行离线训练得到相对应的查找表, 恢复效果受限于训练集合和训练方法的选取. 另外, 该类方法在进行反半色调处理时, 需要知道当前半色调图像的处理方法从而选取相对应的训练集合进行处理, 从而限制了该类方法的适用性.

在实际的应用中, 给定的半色调图像原始处理方法及相关参数是未知的, 需要已知先验知识的反半色调方法的适用性将受到影响. 因此, 反半色调方法依据处理过程中给定图像的半色调方法是否知晓, 划分为盲反半色调和非盲反半色调方法. 本文提出了一种新的盲反半色调方法, 该方法不需要知道任何相关的先验知识, 同时也不需要额外的训练图像集合. 本文算法建立在统一的离散 Voronoi 图框架之上, 将传统的反半色调问题转化为插值问题并计算获取连续色调图像. 另外, 算法处理过程中充分利用了图形处理器(GPU)的并行计算能力提高了算法的计算速度. 首先, 在快速离散 Voronoi 图计算基础上, 利用 Voronoi 图信息计算得到种子点的灰度值. 然后, 充分利用黑色和白色种子点对图像种子点灰度信息估计的互补性, 对种子点灰度估计值进行了修正. 最后, 通过离散的 Sibson 插值方法得到最终的连续色调图像.

## 2 相关定义

### 2.1 离散 Voronoi 图

传统的 Voronoi 图是在一定区域内给定有限集合的种子点, 然后依据种子点将区域分成很多小的区域, 同时确保每个小的区域内的点距离当前小区域的种子点距离最近. 在  $d$  维空间内, 给定一区域  $\Omega$  和有限集合  $S = \{S_1, S_2, \dots, S_n\}$  包含  $n$  个种子点; Voronoi 图  $V(S)$  是指将区域  $\Omega$  划分为很多小的区域  $V(S_i)$ , 对于小区域  $V(S_i)$  中的任意点, 其距离当前区域种子点  $S_i$  的距离小于与其他种子点  $S_j \in S (j \neq i)$ . 区域  $V(S_i)$  和所对应的

种子点被称为 Voronoi 图, 其形式化描述如下:

$$V_i = \{\omega \in \Omega \mid d(\omega, S_i) < d(\omega, S_j), \quad (1)$$

$$\text{for: } i, j = 1, \dots, n; j \neq i\}$$

离散 Voronoi 图是在光栅化空间内 Voronoi 图的表示形式. 在图像处理领域, 光栅化空间是指规则化表示的像素空间. 在本文算法的计算过程中, 我们所需要的就是当前种子点所在 Voronoi 区域的相关信息, 包括 Voronoi 区域的大小、相邻 Voronoi 区域等. 因此, 算法所需要的信息就是每个像素点所属的种子点位置, 通过离散 Voronoi 算法的计算就可以得到以上相关信息. 传统的离散 Voronoi 图的计算量过高, 为了提高算法的计算效率大量基于 GPU 的并行运算算法被应用于其计算过程, 本文的算法采用了基于 GPU 的 Jump Flooding 离散 Voronoi 算法<sup>[16]</sup>.

### 2.2 离散 Sibson 插值

Sibson 插值是指充分利用邻域种子点信息作为插值过程中的权重计算参数的一种自然插值方法, 该方法是由 Sibson 在文献[17]中提出来的. 该方法的特点在于充分利用相邻种子点的信息进行权重计算. 如图 1 所示, 通过图例的方式给出了该方法在只有四个种子点的计算过程描述. 给定一组种子点集合及其对应的 Voronoi 区域如图 1(a)所示. 在种子点值已知的情况下, 需要对  $q$  点的值进行插值计算.  $q$  点被插入已有的 Voronoi 图, 如图 1(b)所示. 将  $q$  点当成新加入的种子点, 其对应的 Voronoi 图  $V(q)$  包括  $k$  个邻接 Voronoi 图  $V(S_1), \dots, V(S_k)$ .  $k$  个种子点  $S_1, \dots, S_k$  称为待插值点  $q$  的自然邻居, 在图 1(c)中的重叠区域( $u_1$  到  $u_4$ )所对应的区域比例作为相应原种子点 Voronoi 区域内像素的权重值.  $q$  点的像素值由以下公式(2)计算得到, 如图 1(d)给出了 Sibson 插值的最终结果.

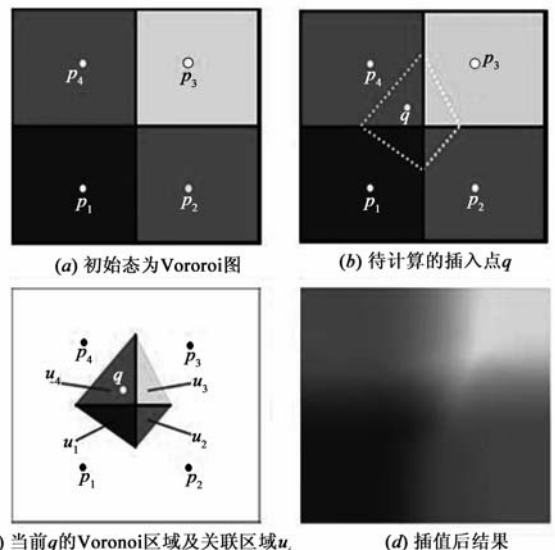


图1 Sibson插值的基本思路

$$f(q) = \frac{\sum_{i=1}^k U_i * f(S_i)}{\sum_{i=1}^k U_i} \quad (2)$$

离散 Sibson 插值概念类似于离散 Voronoi, 是 Sibson 插值在像素空间的计算. 传统的 Sibson 插值方法需要多次计算 Voronoi, 其计算费时从而影响了该方法的推广使用. 文献[3]给出了快速计算离散 Sibson 插值的算法, 该方法充分利用了离散空间的特性将插值过程转化为扩散过程进行分析, 只需要进行一次循环迭代即可得到插值结果. 另外, 算法计算过程中使用了 GPU 将每个像素的灰度值计算转化为并行计算, 从而节省了算法的计算时间.

### 3 基于离散 Sibson 插值的反半色调方法

本文算法的提出是基于 Voronoi 图能够很好地反映图像细节和灰度信息的假设. 通过对半色调图像进行分析, 发现灰度值比较接近于黑色的位置其半色调处理后存在的点更加密集; 同时半色调图像通过点的分布能够很好地反映出图像的结构和细节信息. 基于以上的观察, 结合 Voronoi 图区域划分及其在图像近似和压缩领域使用的启发, 提出了本文算法. 其整体流程和详细描述将在本章中分别进行分析和论述.

#### 3.1 离散 Voronoi 图的计算

在本文算法中, 输入为任意方法的半色调图像  $H$ , 其中包括了半色调点和非半色调点. 为了简化说明, 接下来的描述中将半色调点描述为黑点, 非半色调点描述为白点. 依据以上的假设, 本文算法将半色调图像中的黑点或者是白点当做 Voronoi 图计算过程中的种子点. 接下来算法中, 针对白点或者是黑点的 Voronoi 计算是一致的, 所以只针对黑色点重分布及其 Voronoi 计算进行说明.

一般的半色调图像点分布过于密集, 其 Voronoi 信息不能够很好反映灰度及其变化. 为了提高灰度估计的正确性, 算法首先需要将半色调图像  $H$  放大得到放大后的半色调图像  $H'$ . 在算法的实现过程中, 对于放大的参数通过变量  $blowup$  进行控制; 通常该参数被设置为常数, 通过对结果的分析本文算法中该参数被设置为常数 6 时其灰度估计正确性更高. 对于每个黑点位置  $(x, y)$ , 算法对黑点位置按照公式(3)进行重新分布:

$$H'(x', y') = blowup \times blowup H(x, y) \quad (3)$$

上式中,  $x'$  大小为  $blowup \times x$ ,  $y'$  大小为  $blowup \times y$ . 为了保证半色调图整体灰度值在放大过程中的不变性, 算法在放大过程中保持黑点数目不发生变化. 同时, 放大后的黑点凝聚的灰度值变为  $blowup \times blowup$ . 在完成对半色调图像的放大操作后, 将黑点当做种子

点进行 Voronoi 划分计算. 由于算法是针对图像空间进行处理, 因此 Voronoi 图的计算属于离散 Voronoi 算法的计算. 为了提高其计算的运行效率和速度, 本文采用了基于 Jump Flooding<sup>[16]</sup> 的 GPU 算法进行 Voronoi 图计算. 通过 GPU 的并行加速, 将 Voronoi 图的计算复杂性降低到了常数时间内.

#### 3.2 基于 Voronoi 图信息的种子点灰度值估计

依据半色调图像的 Voronoi 图能够反映图像的结构信息的假设, 使用当前种子点所属的 Voronoi 图信息对种子点的灰度值进行估计. 本文算法中, 主要依据 Voronoi 图的大小信息对灰度值进行估计. 使用计算得到的放大后半色调图像的离散 Voronoi 图, 算法中使用  $voronoi(P_i)$  进行表示.  $voronoi(P_i)$  存储了每个像素  $P_i$  所对应的种子点坐标信息. 依据每个像素点对应的种子点信息, 计算获取每个 Voronoi 区域的面积信息, 算法中使用  $area(S_i)$  表示,  $S_i$  代表每个种子点. 获取每个种子点对应的 Voronoi 区域的面积信息后, 将依据公式(4)计算种子点对应的灰度值, 其中  $blowup$  表示半色调图像的放大参数,  $area(S_i)$  表示当前种子点对应的区域大小.

$Intensity(S_i) =$

$$\begin{cases} (1.0 - \frac{blowup \times blowup}{area(S_i)}) \times 255; & \text{if (black \cdot dots)} \\ \frac{blowup \times blowup}{area(S_i)} \times 255; & \text{if (white \cdot dots)} \end{cases} \quad (4)$$

#### 3.3 种子点灰度值的修正

使用种子点灰度值的估计获取了图像的基本估计, 但是通过对半色调图像的 Voronoi 图的观察分析得出在平滑区域的 Voronoi 估计存在较大偏差. 比如, 对于一部分灰度接近于黑色的区域, 半色调的处理过程往往会引入高频噪声从而造成 Voronoi 估计偏离实际的图像灰度值. 通过对黑色、白色点的 Voronoi 图的分析得到, 两者对于引入的高频噪声处理恰好具有互补性.

因此, 在种子点灰度值的初步估计获取后, 将通过白点的种子点估计值对其进行灰度值修正. 本文采用了一种比较简单直观的方法解决连续灰度区域的噪声问题. 通过观察分析发现, 通常情况下以黑色点作为种子点的估计相对正确性更高. 因此, 算法中将黑色点的灰度估计作为基准灰度, 通过将黑点在一定窗口范围内的密度作为区别连续黑色区域的判别标准. 算法中通过阈值  $threshold$  设定判定条件, 大于阈值的其种子点灰度修正为白色点的估计值. 本文中给定的窗口大小为  $5 \times 5$ , 阈值  $threshold$  一般设定为 0.7. 其修正过程依据  $Density(S_i)$  进行判断, 其值超过 0.7 其像素值修正为  $IntensityWhite(S_i)$  对应位置的像素值; 否则, 当  $Density(S_i)$  值小于 0.7 时取  $IntensityBlack(S_i)$  对应位置的像素值. 其中,  $Density(S_i)$  表示当前黑色种子点所在位置在

指定窗口内的黑点密度值.

### 3.4 非种子点灰度值的插值计算

在完成种子点灰度值的估计和修正后,半色调图像转化为待插值计算的灰度图像.在插值方法方面,本文充分利用了已计算获取的离散 Voronoi 信息,采用了基于该信息的 Sibson 插值方法.同时,为了提高离散 Sibson 插值计算速度,本文对基于 GPU 硬件加速的离散 Sibson 插值方法<sup>[3]</sup>进行了改进.将计算费时的 kd-tree 构建过程通过离散 Voronoi 图的计算结果进行替代,其插值计算过程描述如下:

**Step1** 已知种子点对应的灰度值  $Intensity(S_i)$

**Step2** 构建  $n$  个种子点的离散 Voronoi 图  $voronoi(S_i)$

**Step3** 对于每个非种子点的像素点  $P_i$ :初始化每个像素点的灰度和  $cumultae(P_i)$ 和范围内像素点数目  $number(P_i)$ 等于零,计算当前像素与其种子点  $voronoi(S_i)$ 的距离  $d$ ,在以  $P_i$  为圆心,半径为  $d$  的圆内所有的像素  $V_c(i)$ :将  $V_c(i)$  对应的种子点灰度值加入  $cumultae(P_i)$ ,  $number(P_i)$ 加 1

**Step4** 对于每个非种子点的像素点  $P_i$ :  $Intensity(P_i) = cumulate(P_i) / number(P_i)$

最后,在计算得到种子点和非种子点的灰度值后,算法得到了放大的连续色调图像.然后,依据前述的放大参数  $blowup$  进行放缩采样得到最终原始大小的半色调图像  $I$ .在以下算法 1 的描述中,给出了本文算法的整体处理过程.

算法 1 基于离散 Sibson 插值的盲反半色调算法

**Step1** 通过参数  $blowup$  对原半色调图像  $H$  进行放大得到  $H'$ ,同时对黑点进行重新分布保留其数目不变.

**Step2** 分别将黑点或白点当作种子点进行 Voronoi 图计算,得到对应的 Voronoi 图  $V_B$  和  $V_W$ .

**Step3** 对于每个 Voronoi 图,依据其区域大小信息计算其种子点的灰度值,分别得到  $IntensityBlack(S_i)$ 和  $IntensityWhite(S_i)$ .

**Step4** 对于黑点估计的种子点灰度值,依据其相对应位置白点估计值对其灰度值估计进行修正得到  $Intensity(S_i)$ .

**Step5** 在种子点灰度的估计基础上,通过改进离散的 Sibson 插值算法计算所有非种子点的灰度值.

**Step6** 对现有的图像通过参数  $blowup$  进行向下采样得到最终的图像  $I$ .

## 4 算法实现及结果分析

为了能够提高算法的计算速度,算法的实现过程中充分利用了并行运算硬件图形处理器 GPU 的计算能力,算法实现过程中结合了 CPU 和 GPU 进行编程实现.其中,离散 Voronoi 的计算和离散 Sibson 插值计算涉及大量可并行运算的过程在 GPU 上进行了实现.对于灰

度值的估计以及灰度值修正等涉及大量逻辑判断的过程在 CPU 上进行了实现.在算法的实现过程中,使用 OpenGL 库、Jump Flooding 算法<sup>[16]</sup>和 NVIDIA 的 Cg 语言进行 GPU 编程,实验采用的显卡为普通电脑的 NVIDIA GeForce 8600GS.为了提高计算的精度,对离散 Voronoi 图和 Sibson 插值计算过程中在 GPU 上的数据存储使用了 4 通道的 32 位浮点类型进行存储.在 Sibson 插值的计算过程中,算法采用了文献<sup>[3]</sup>的高效计算方法,同时通过离散 Voronoi 图信息的计算取代了算法中的 Kd-tree 计算,从而节省算法的计算时间.

本文算法处理过程中,不需要知道现有半色调图像的任何先验知识,可以适用于任意的误差扩散和点扩散等不同的半色调处理方法.在实验过程中,本文分别对不同方法的半色调图像进行了处理,包括了 Floyd 误差扩散方法<sup>[1]</sup>、其它核函数的误差扩散方法以及基于点扩散的半色调方法<sup>[18]</sup>.

### 4.1 Floyd 误差扩散半色调处理结果分析

在本段中,给出了本文算法的反半色调计算结果,同时针对以上两类包括只适用于误差扩散方法的反半色调和盲反半色调方法中的经典算法进行了比较.其中,基于查找表<sup>[11]</sup>、边缘增强查找表<sup>[13]</sup>的方法属于盲反半色调算法,快速梯度滤波算法<sup>[8]</sup>只能适用于误差扩散方法.

图 2 给出了本文算法与基于查找表算法、边缘增强的查找表算法、快速滤波算法的处理结果,在下图中使用  $512 \times 512$  大小的 Lena 作为测试图像,所对应的半色调处理方法为 Floyd 误差扩散方法<sup>[1]</sup>.在上例中,本文算法所使用的  $blowup$  参数为 6、修正参数  $threshold$  为 0.7,其处理的结果如图 2(d)所示.图 2(a)和图 2(b)分别为基于查找表算法和边缘增强查找表算法的反半色



图 2 对于 Floyd 误差扩散半色调方法的计算结果

调处理结果. 相对于以上两类方法来说, 本文结果能够更好的去除平滑区域的噪声并保留了更多的细节信息. 同时, 本文算法不需要额外的针对半色调方法的训练集合和离线训练过程, 也不需要已知当前的待处理图像所属的半色调方法. 图 2(c) 结果为基于梯度的快速滤波方法<sup>[8]</sup>的处理结果, 相对于本文算法和基于查找表的算法其恢复结果更加平滑, 但是细节相对于以上两类方法则略微模糊. 快速滤波方法只能应用于部分误差扩散半色调算法, 对于自适应误差扩散方法 Ostromoukhov 误差扩散半色调方法<sup>[4]</sup>, 由于不能够估计其误差扩散核因此将不能够适用.

图 3 给出了图像 Lena 恢复结果中肩膀部分的放大效果比较, 其中图 3(a) 对应的是边缘增强查找表方法的处理结果, 图 3(b) 对应的是本文算法的计算结果. 相对于查找表方法, 边缘增强的查找表方法能够更好的保持图像边缘信息. 在图 3 中, 本文算法的结果较边缘增强的 LUT 来说更加平滑和清晰, 尤其是在大区域的连续区域本文算法平滑性更好. 从以上实验结果来说, 本文算法能够更好的适用于各类不同的半色调方法. 由于不需要已知当前图像所属的半色调方法和该类方法所对应的训练集合, 因此本文算法具备了更好的算法适用性.

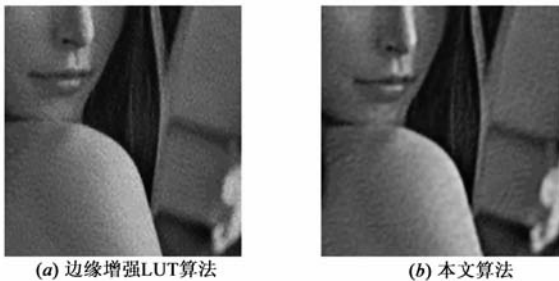


图3 针对Lena图像肩部放大区域的结果比较

#### 4.2 其它核函数的误差扩散半色调方法

本文算法除了可以被应用于 Floyd 核误差扩散方法外还可以被应用到其他核误差扩散方法, 比如 Jarvis 误差扩散<sup>[2]</sup>和 Ostromoukhov 误差扩散半色调方法<sup>[4]</sup>. 图 4(a) 和图 4(b) 分别为 Jarvis 误差扩散的半色调图像和本文算法所计算的连续色调图像; 图 4(c) 和图 4(d) 分别为 Ostromoukhov 误差扩散的半色调图像和本文算法所计算的连续色调图像. 针对于不同的误差扩散核, 本文算法无需估计其误差扩散核. 图 4 中算法采用的参数与上图的 Floyd 方法相同, *blowup* 参数为 6、修正参数 *threshold* 为 0.7. 相对于 Floyd 误差扩散方法的恢复结果来说, 图 4 中的重建结果能够保持稳定的效果. 无论从图像的平滑程度还是细节保留来说, 本文算法都具有很好的重建效果.



图4 其它核函数误差扩散半色调方法处理结果

#### 4.3 点扩散半色调方法

除了不同的误差扩散半色调方法外, 本文算法还可以被应用于其它的半色调处理方法, 比如基于点扩散的半色调方法<sup>[18]</sup>. 图 5 算法中, *blowup* 参数为 6、修正参数 *threshold* 为 0.72. 图 5(a) 和 (b) 分别为针对 Lena 图像的点扩散的半色调图像和本文算法所计算的连续色调图像; 图 5(c) 和 (d) 分别为针对 Pepper 图像的点扩散的半色调图像和本文算法所计算的连续色调图像. 从半色调图像来看, 点扩散方法更好的保留了图像的细节, 同时噪声看上去更加严重. 从本文算法的重建结果来看, 该方法能够很好的确保半色调图像保留下来的细节信息. 同时, 针对半色调图像中大幅连续区域的噪声也能够很好的去除. 从整体效果来说, 基本上保持了图像的平滑和细节. 但是, 在重建图像的平滑效果来说, 仍然存在一些噪声现象, 该类噪声可以通过近一步的后处理比如小波滤波<sup>[19]</sup>等方法进行处理.



图5 点扩散半色调方法处理结果

## 5 结论

本文提出了一种新的盲反半色调方法,该方法不依赖于任何半色调方法及其相关的先验参数,同时也不需要额外的图像训练集合.本文算法建立在统一的离散 Voronoi 图框架之上,将传统的反半色调问题转化为插值问题并计算获取连续色调图像.相对于已有的反半色调方法,本文算法具有更广的适用性,可以被应用于绝大部分的半色调图像.在效果处理上,本文算法能够更好的去除平滑区域的噪声同时很好的利用了 GPU 的并行运算能力,提高了算法的计算效率.但是,针对点扩散的半色调图像,算法处理结果仍然存在一些噪声.在未来的工作中,我们将分析点扩散半色调方法的特点,通过引进保留图像边缘信息的非线性滤波方法的思想<sup>[19]</sup>对算法进一步改进从而得到更好的结果.

### 参考文献:

- [1] R W Floyd, L Stienberg. An adaptive algorithm for spatial grayscale[J]. Proceedings of the Society for Information Display, 1976, 17(2): 75 – 77.
- [2] J Jarvis, et al. A survey of techniques for displaying of continuous-tone pictures on bilevel displays[J]. Computer Graphics and Image Processing, 1976, 5(1): 13 – 40.
- [3] Sung Park, Lars Linsen, et al. Discrete sibson interpolation[J]. IEEE Transactions on Visualization and Computer Graphics, 2006, 12(2): 243 – 253.
- [4] Ostromoukhov V. A simple and efficient error-diffusion algorithm[A]. Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques[C]. New York, NY, USA: ACM, 2001. 567 – 572.
- [5] N D Venkata, T D Kite, et al. Fast blind inverse halftoning[A]. Proceedings of IEEE International Conference on Image Processing[C]. Chicago, IL, USA, 1998. 64 – 68.
- [6] 牛少彰, 钮心忻, 杨义先, 胡文庆. 半色调图像中数据隐藏算法[J]. 电子学报, 2004, 32(7): 1180 – 1183.  
Niu Shao-zhang, Niu Xin-xin, Yang Yi-xian, Hu Wen-qing. Data hiding algorithm for halftone images[J]. Acta Electronica Sinica, 2004, 32(7): 1180 – 1183. (in Chinese)
- [7] Z Xiong, M T Orchard, K Ramchandran. Inverse halftoning using wavelets[J]. IEEE Transactions on Image Processing, 1996, 8(10): 1479 – 1483.
- [8] T D Kite, N D Venkata, B L Evans, A C Bovik. A high quality, fast inverse halftoning algorithm for error diffusion halftones[A]. Proceedings of IEEE International Conference on Image Processing[C]. USA: IEEE, 1998. 59 – 63.
- [9] R L Stevenson. Inverse halftoning via MAP estimation[J]. IEEE Transactions on Image Processing, 1997, 6(4): 574 – 583.
- [10] Bozkurt Unal G, A E Cetin. Restoration of error-diffused im-

ages using projection onto convex sets[J]. IEEE Transactions on Image Processing, 2001, 10(12): 1836 – 1841.

- [11] M Mese, P P Vaidyanathan. Look up table (LUT) method for inverse halftoning[J]. IEEE Transactions on Image Processing, 2001, 10(10): 1566 – 1578.
- [12] H Y Kim, R de Queiroz. Inverse halftoning by decision tree learning[A]. Proceedings of IEEE International Conference Image Processing[C]. USA: IEEE, 2003. 913 – 916.
- [13] K L Chung, S T Wu. Inverse halftoning algorithm using edge-based lookup table approach[J]. IEEE Transactions on Image Processing, 2005, 14(10): 1583 – 1589.
- [14] M Mese, P P Vaidynathan. Tree-structured method for LUT inverse halftoning and for image halftoning[J]. IEEE Transactions on Image Processing, 2002, 11(6): 644 – 655.
- [15] R Eschbach, et al. Error-diffusion algorithm with edge enhancement[J]. Journal of the Optical Society of America A: Optics, Image Science, and Vision, 1991, 8(12): 1844 – 1850.
- [16] Guodong Rong, et al. Jump flooding in GPU with applications to voronoi diagram and distance transform[A]. Proceedings of the Symposium on Interactive 3D Graphics and Games[C]. Redwood City, California: ACM, 2006. 109 – 116.
- [17] R Sibson. A vector identity for the dirichlet tessellation[J]. Mathematical Proceedings of the Cambridge Philosophical Society, 1980, 87(1): 151 – 155.
- [18] D E Knuth. Digital halftones by dot diffusion[J]. ACM Transactions on Graphics, 1987, 6(4): 245 – 273.
- [19] 蔡超, 丁明跃, 周成平, 张天序. 小波域中的双边滤波[J]. 电子学报, 2004, 32(1): 128 – 131.  
Cai Chao, Ding Ming-yue, Zhou Chen-ping, Zhang Tian-xu. Bilateral filter in wavelet domain[J]. Acta Electronica Sinica, 2004, 32(1): 128 – 131. (in Chinese)

### 作者简介:



王 磊 男, 1982 年 8 月出生于山东淄博. 山东大学计算机科学与技术学院博士研究生, 主要从事图形图像处理、机器学习等方面的研究.  
E-mail: wanglpqq@gmail.com



李学庆 男, 1964 年 4 月出生于山东德州. 山东大学教授、博士生导师, 主要从事软件工程、人机交互、图形图像处理等方面的研究.  
E-mail: xqli@sdu.edu.cn