

# 基于重心 Delaunay 三角剖分的蓝噪声点采样算法

齐若同<sup>1,3)</sup>, 肖艳阳<sup>1,3)</sup>, 曹娟<sup>2)</sup>, 陈中贵<sup>1,3)\*</sup>

<sup>1)</sup> (福建省智慧城市感知与计算重点实验室(厦门大学) 厦门 361005)

<sup>2)</sup> (厦门大学数学科学学院 厦门 361005)

<sup>3)</sup> (厦门大学信息科学与技术学院 厦门 361005)  
(chenzhonggui@xmu.edu.cn)

**摘要:** 为了生成带有高质量蓝噪声性质的采样分布, 提出一种基于重心 Delaunay 三角剖分的点采样算法. 该算法将 Delaunay 三角剖分与 1-邻域三角片重心相结合, 迭代地将每个采样点移至其 1-邻域三角片的重心处并更新采样点之间的拓扑连接关系; 重心通过给定的密度函数计算得出. 实验结果表明, 本文算法在运行效率与鲁棒性方面均有一定优势.

**关键词:** 蓝噪声采样; Delaunay 三角剖分; 重心 1-邻域三角化  
**中图法分类号:** TP391.41 **DOI:** 10.3724/SP.J.1089.2018.16760

## Blue-Noise Point Sampling Based on Centroidal Delaunay Triangulation

Qi Ruotong<sup>1,3)</sup>, Xiao Yanyang<sup>1,3)</sup>, Cao Juan<sup>2)</sup>, and Chen Zhonggui<sup>1,3)\*</sup>

<sup>1)</sup> (Fujian Key Laboratory of Sensing and Computing for Smart City, School of Information Science and Engineering, Xiamen University, Xiamen 361005)

<sup>2)</sup> (School of Mathematical Sciences, Xiamen University, Xiamen 361005)

<sup>3)</sup> (School of Information Science and Engineering, Xiamen University, Xiamen 361005)

**Abstract:** We present an algorithm for generating point distributions with high-quality blue noise characteristics based on centroidal Delaunay triangulation. The method combines Delaunay triangulation with centroidal patch triangulation, and iteratively moves each vertex to the centroid of its 1-ring neighborhood and updates the topological connectivity between the sampling points. The centroid of a patch is calculated by using a given density function. The experimental results demonstrate the effectiveness and robustness of the proposed algorithm.

**Key words:** blue-noise sampling; Delaunay triangulation; centroidal patch triangulation

### 1 相关工作

采样在计算科学中是一项重要技术, 计算机图形学从本质上来讲是一个采样的过程, 对于采样问题可以理解为关于如何生成一组带有某种分

布特征点集. 该技术是指在给定空间内的有界采样域的前提下, 将这些采样点放置于采样域内合理的位置, 使得形成的采样分布满足用户需要的一些特性.

如何生成符合用户需求的采样模式是计算机图形学中的一重要课题. 通常情况下, 希望采样点

收稿日期: 2017-07-05; 修回日期: 2017-08-30. 基金项目: 国家自然科学基金(61472332); 福建省自然科学基金(2018J01104). 齐若同(1992—), 女, 硕士研究生, 主要研究方向为计算机图形学; 肖艳阳(1991—), 男, 博士研究生, 主要研究方向为计算机图形学; 曹娟(1983—), 女, 博士, 副教授, 硕士生导师, CCF 会员, 主要研究方向为 CAGD&CG; 陈中贵(1982—), 男, 博士, 副教授, 硕士生导师, CCF 会员, 论文通讯作者, 主要研究方向为计算机图形学、数字几何处理.

均匀且随机地分布在采样区域内, 这种分布结构之所以表现出色, 原因是它们表现出了低差异性和随机性. 这与灵长类动物视网膜上的感光器的分布十分相似, 因此人类的眼睛也更倾向于拾取这些模式.

Ulichney<sup>[1]</sup>将具有上述性质的采样称为“蓝噪声”采样, 本意是指具有最小低频分量和没有集中的能量尖峰的任何噪声, 也指采样点的均匀、非结构化分布, 这意味着没有由低频造成的结构混叠现象, 也没有间隔很小的局部点簇. 对应到频域中, 低频处接近于 0, 而高频处接近白噪声, 如图 1b<sup>[2]</sup>所示. 蓝噪声采样在各领域迅速普及, 可以被认为是广泛应用的最佳采样模式. 图 2 分别展示了一幅图像的随机采样(如图 2a 所示)与蓝噪声采样(如图 2b 所示)的点采样分布对应的视觉效果<sup>[3]</sup>. 蓝噪声采样在渲染、点画图、动态采样及网格生成等方面有重要的应用<sup>[4]</sup>.

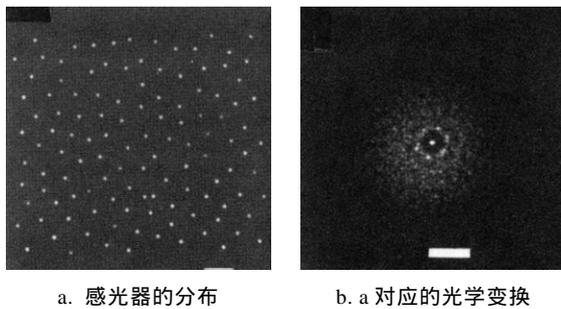


图 1 猴眼感光器的分布及其对应的光学变换

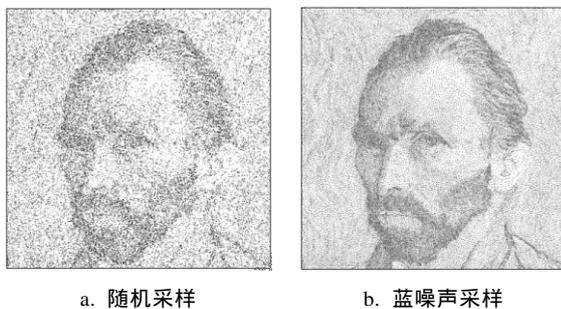


图 2 图像的随机采样和蓝噪声采样视觉效果对比

本文提出一种基于重心 Delaunay 三角剖分的蓝噪声点采样算法. 该算法结合 Delaunay 三角剖分与 1-邻域三角片重心的概念, 迭代更新采样点位置与采样点之间的拓扑关系, 最终得到高质量的具有蓝噪声性质的采样分布.

用于计算机图形学中蓝噪声采样的一类经典技术是基于 Poisson 圆盘分布<sup>[5]</sup>, 该分布所包含的点在空间中均匀且随机地分布, 采样点之间以一

个最小距离  $d_{\min}$  约束, 从而使彼此离散. 当这些采样点被当做信号解释为辐射波长时, 它们的平均傅里叶功率谱产生蓝光, 证明了其具有前文所述的蓝噪声频谱性质.

目前已经有很多这类生成算法, 最早的是 1986 年 Cook<sup>[2]</sup>提出的“掷飞镖”算法. 该算法从一个空的采样点集合开始, 随机产生候选点  $x$ , 如果这个点与采样点集里的其他点  $x_i$  之间的距离大于给定的阈值  $r$ , 即 Poisson 圆盘的半径, 则将该点加入到采样点集合中, 如此迭代下去, 直到满足一定的采样点数目; 但随着采样点的增加, 这种方法也许会出现多次连续选点失败的现象, 也就是多次选点都出现与集合中的点  $x_i$  之间的距离小于阈值  $r$ . 故 McCool 等<sup>[6]</sup>通过适当收缩半径, 以允许添加更多采样点的方法来增大采样点的覆盖率. 然而这类方法难以操纵或改变所产生的样本分布, 并且运算代价较高, 采样点的个数也无法精确控制. 为此, 许多方法对其进行改进和加速. Dunbar 等<sup>[7]</sup>提出一种扇形结构来记录剩余的可接受的采样区域, 加速效果明显; 但该方法具有较强的偏向性, 采点随机性降低. Yan 等<sup>[8]</sup>基于正则三角剖分和 Power 图系统地讨论了 Poisson 圆盘之间的间隙, 并给出相应的算法.

另一类比较经典的算法则是基于松弛算法的采样, 该类算法将采样点分布与某些拓扑结构相结合, 如 Voronoi 图或其对偶 Delaunay 三角剖分等, 进而将问题转化为关于采样点坐标的能量函数优化问题. 这类方法主要包含 2 个步骤: (1) 生成初始采样点集合; (2) 基于集合与拓扑的能量函数来优化点集中点的位置, 迭代直至收敛.

Du 等<sup>[9]</sup>最先提出重心 Voronoi 图(centroidal Voronoi tessellation, CVT)的概念, 所有采样点均位于各自 Voronoi 区域的重心. 经典的 Lloyd 算法<sup>[10]</sup>正是基于 CVT 来实现的, 该算法也被称为 CVT 算法, 其生成的采样点分布满足蓝噪声应有的均匀性; 但随着迭代次数的增加, 这样的点分布呈现出一定的规则性, 因为 Voronoi 细胞是一种接近正六边形的栅格, 如图 3 所示, 这与蓝噪声采样应具有随机性相悖. Reinert 等<sup>[11]</sup>提出将采样点投影到低维空间并与 Lloyd 方法相结合, 有效地降低了规整性这一缺陷. 结合测地距离, 陈中贵等<sup>[12]</sup>提出了一种保持图像特征的 CVT 算法, 由于特征附近的采样点无法越过边界, 避免了最终结果的规整性, 并呈现良好的均匀性.

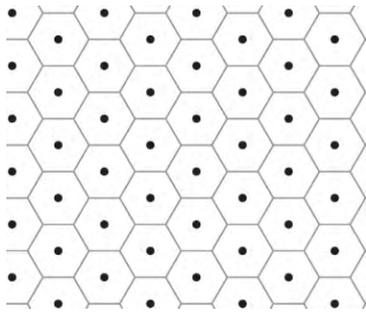


图 3 规则 Voronoi 划分

基于容量约束的思想, Balzer 等<sup>[13]</sup>提出容量约束的 Voronoi 划分(capacity-constrained Voronoi tessellation, CCVT)算法, 其中, 每个采样点仍位于其相应 Voronoi 区域质心位置, 且每个 Voronoi 区域的容量都尽量相等. 在二维空间中, 面积则是容量的一种最基本表现形式. 当给定某个密度函数时, 容量的概念则替换为每个 Voronoi 区域的密度积分. CCVT 采样已被证明具有良好的蓝噪声性质, 但由于 Voronoi 区域的容量计算复杂, 该算法通过将采样区域离散为额外的采样点来近似计算每个区域的容量, 导致样本数比给定的采样点数  $n$  大出几个数量级, 非常影响算法效率. 后来, Li 等<sup>[14]</sup>基于 GPU 实现一种快速 CCVT 算法(fast capacity constrained Voronoi tessellation, FCCVT), 但其思想仍然采用离散化近似容积, 对于大数量级采样点的优化加速仍然受限.

基于变分的思想, Chen 等<sup>[15]</sup>将生成 CCVT 的过程转化为一个非线性优化问题, 在极小化其目标函数的过程中不断地更新点集. de Goes 等<sup>[16]</sup>基于最优传输的连续约束最小化问题重构了 CCVT 算法, 并且采用 Power 图代替 CCVT 算法中的 Voronoi 图, 最终得到了良好的蓝噪声采样效果.

Xu 等<sup>[17]</sup>提出基于容量约束的 Delaunay 三角方法(capacity-constrained Delaunay triangulation tessellation, CCDT), 将复杂的 Voronoi 区域替换为 Delaunay 三角剖分, 此时容量也就转化为每个 Delaunay 三角片的容量, 二维情况下即三角片的面积. 该方法很好的解决了运行效率问题, 但算法略微复杂. 陈中贵等<sup>[18]</sup>提出一种结合拓扑优化的最优 Delaunay 三角剖分(optimal Delaunay triangulation, ODT)方法<sup>[19]</sup>, 有利于跳出 ODT 的局部最优解, 获得更好的顶点分布. Ahmed 等<sup>[20]</sup>也提出一种基于 Delaunay 三角剖分的蓝噪声采样方法(简称为 push-pull), 通过不断地移动 1-邻域顶点(太近的推开, 太远的拉近), 同时优化对应的 Voronoi 细胞的

容积来获取蓝噪声采样.

相似地, 本文也采用松弛算法, 利用基于 1-邻域三角化重心(centroidal patch triangulation, CPT)<sup>[21]</sup>的原理, 采用 Delaunay 三角剖分作为拓扑结构与采样点相结合的算法进行蓝噪声采样, 并将其命名为基于重心 Delaunay 三角剖分(centroidal Delaunay triangulation, CDT)的蓝噪声采样算法; 同时结合 OpenGL 中的渲染功能, 实现一种原理简单、收敛速率快, 且具有一定鲁棒性的算法.

## 2 本文算法

### 2.1 CPT 原理

对于  $\mathbb{R}^N$  空间中一组给定的采样点集, 表示为  $X = \{x_i\}_{i=1}^n \subset \mathbb{R}^N$ . 记  $T$  为采样点集  $X$  的三角剖分, 对任意一点  $x_i \in X$ , 所有与该采样点相邻的三角片构成的集合称为点  $x_i$  的 1-邻域区域, 用  $\Omega_i$  来表示, 如图 4 所示. 每个  $\Omega_i$  的重心会参照一个定义在凸域  $\Omega \subset \mathbb{R}^N$  上的密度函数  $\rho(x)$  表示,  $\rho(x) > 0$ ,  $\int_{\Omega} \rho(x) dx < \infty$ .  $\Omega_i$  的重心定义为

$$c_i = \frac{\int_{\Omega_i} \rho(x) x d\sigma}{\int_{\Omega_i} \rho(x) d\sigma} \quad (1)$$

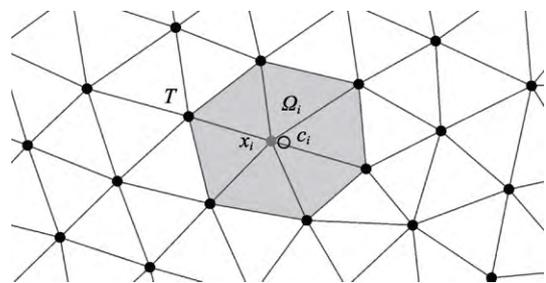


图 4 采样点的 1-邻域及其重心示意图

若任意一点  $x_i \in T$  都与该采样点区域的重心  $c_i$  重合, 则对点集  $X$  的三角剖分  $T$  被称为重心 1-邻域三角剖分. Chen 等<sup>[22]</sup>采用 CPT 获得了高质量的曲面网格.

### 2.2 算法描述

Delaunay 三角剖分有诸多优良性质, 如空圆特性、最大化最小角等, 在计算机图形学的多个领域中都有应用. 文献[19]中已经证明了当  $\Omega$  为一个凸包, 对于一个固定数量的点集  $X$ , 其最优的连接关系, 也即加权插值误差最小时, 就是对该点集的 Delaunay 三角剖分. 故文献[21]中提出了基于

Delaunay 三角剖分的拓扑结构来生成 CPT 的目标方程, 称之为重心 Delaunay 三角剖分, 定义为

$$F(X, T) = \sum_{i=1}^n \int_{\Omega_i} \rho(x) \|x - x_i\|^2 d\sigma \quad (2)$$

其中,  $T$  为点集  $X$  的 Delaunay 三角剖分,  $F(X, T)$  的最小值是通过最小化每个采样点  $x_i$  的 1-邻域  $\Omega_i$  的能量值来实现的, 即  $x_i$  最优的位置  $x_i^*$  满足

$$x_i^* = \min_{x_i \in \Omega_i} \int_{\Omega_i} \rho(x) \|x - x_i\|^2 d\sigma \quad (3)$$

事实上,  $\Omega_i$  的重心  $c_i$  正是式(3)的解, 文献[21]中将该问题分解为 2 个子问题来求解, 并设计一个基于两步迭代的算法用来生成这个 1-邻域  $\Omega_i$  的重心:

(1) 更新拓扑结构, 即更新对点集  $X$  的 Delaunay 三角剖分.

(2) 更新采样点位置, 逐一调整点集中的点  $x_i$ , 使其与相应的 1-邻域  $\Omega_i \in T$  的重心  $c_i$  重合, 即

$$x_i^* = c_i \quad (4)$$

这一步骤与前文所提到的 Lloyd 算法<sup>[10]</sup>很相似. Lloyd 算法将采样点的位置移至该点相应的 Voronoi 区域重心, CDT 算法则是将采样点位置移至该点相应的 1-邻域 Delaunay 三角片的重心. 本文将 CDT 算法应用于图像的蓝噪声采样中, 不仅展现了不错的视觉效果, 而且从与现有方法的对比来看, 在运算效率方面也有很大优势.

初始化一个采样点集  $X$ , 然后迭代进行下面的步骤, 直到满足一定的停止条件. 算法步骤如下:

输入. 采样点数  $n$ , 迭代次数  $N$ .

输出. 具有蓝噪声性质的采样分布.

Step1. 适应性采样初始化, 建立初始采样点集.

Step2. 更新拓扑关系, 对点集建立 Delaunay 三角剖分.

Step3. 计算重心, 计算采样点 1-邻域 Delaunay 三角片重心.

Step4. 更新采样点几何位置, 将每个采样移至该点相应的 1-邻域重心处.

Step5. 若迭代次数超过给定值  $N$ , 算法结束; 否则, 转 Step2.

### 2.2.1 适应性采样初始化

在计算机图形学的实际应用中, 人们期望采样点集的分布能够根据某个给定的密度函数  $\phi(x)$  来进行. 当该密度函数为一个常量时, 对应的采样为均匀采样. 对于这种采样, 本文通过随机算法在采样区域内随机地生成给定点数的采样点, 建立初始采样点集  $X$ , 此时的采样分布具有前文所述

的白噪声性质. 当密度函数为非常量时, 对应为非均匀采样. 例如, 在图像点画风格处理技术中, 给定一幅灰度图, 以不同的分布密度  $\phi(x)$  拟合图像上对应的灰度, 从而使输出的采样分布在视觉上与初始图片十分相似. 具体过程如下: 先将输入的图片转化为灰度图, 将灰度归一化后作为相应的密度值; 再从(0,1)随机生成一个数  $m$ , 比较两者大小, 若后者比前者值大, 则保留该点. 这样最终的效果则是更多的点位于图像密度高的区域, 更少的点位于密度小的区域.

然而, 均匀采样情况下仅采用随机化方法生成的初始采样点难免会有分布不均的局部点簇, 同样地, 非均匀采样情况中初始采点的方法也不能保证初始分布的连续性, 在某些局部仍然会出现分布不均的情况. 由于本文算法是基于局部优化来进行迭代操作, 当达到局部最优时, 会对满足蓝噪声性质中的均匀性产生一定的副作用. 本文的目标是尽可能将初始化做得更好, 故在初始采点之后还用到文献[16]中的带容积约束的 Power 图生成方法, 使得初始点集分布在相同面积区域上的点的个数相同.

### 2.2.2 建立 Delaunay 三角剖分

根据初始采样点集构建出初始的 Delaunay 三角化结构, 本文采用的是计算几何算法库<sup>[23]</sup>中的标准 Delaunay 三角剖分算法.

可以看出, 式(3)只考虑了内部顶点的情况, 即有完整的 1-邻域三角片的采样点. 由于该算法使用重心位置来更新采样点的位置, 根据重心的性质, 计算出的位置  $c_i$  一定会落在采样点相应的 1-邻域内部  $\Omega_i$ . 如果边界点也按照该公式计算并更新边界上采样点的位置与相关的连接关系, 那么边界点就会逐渐向区域  $\Omega_i$  的内部移动, 导致该采样点集的 Delaunay 三角剖分所覆盖的区域会随着迭代次数的增多而不断萎缩, 无法完整地剖分指定的区域, 会得到错误的结果. 为了避免边界萎缩效应, 需要更好的保持边界, 本文采用 2 种方法: (1) 周期采样. 图 5 所示为根据周期采样生成的结果; (2) 通过给定的一个凸多边形  $D$  作为一个有界域, 采用一个系数  $\lambda$  与采样域的大小乘积计算出一个阈值, 当采样点与采样区域边界之间的距离小于该阈值时, 该采样点就会被映射到边界上, 其中, 系数  $\lambda$  根据采样点的数量而定. 根据大量的实验发现, 该值取 0.1~0.2 效果较好. 将投射到边界的点作为边界点集, 在之后的迭代中, 边界点与

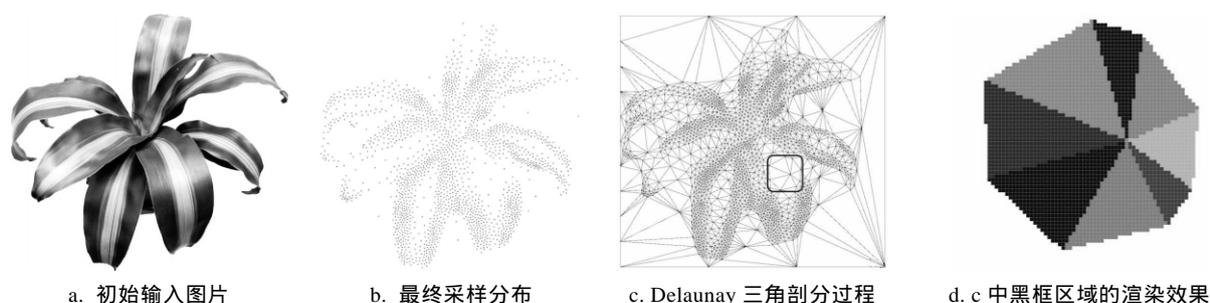


图 5 渲染优化的过程

内部点的处理方式有所不同. 对于边界点, 本文算法限定其只能沿着边界移动, 并且其位置只受其边界上相邻点的影响; 计算出该点与其相邻点的位置平均值作为该边界点更新的位置. 随着迭代次数的增加, 边界点也逐渐表现为均匀的采样分布, 在后续输出最终的采样分布时, 再将限定在边界的采样点去掉即可. 本文大部分的实验结果都采用了第 2 种方法.

### 2.2.3 采样点位置更新

计算采样点位置有 2 个步骤:

Step1. 依次计算采样点 1-邻域中的所有三角片的重心;

Step2. 根据三角片重心计算该采样点对应的 1-邻域的重心.

在均匀采样情况下, 结合前文所述的三角形与多边形重心计算方法, 可以快速、准确的计算出每一个邻域的重心, 从而快速迭代更新点的位置. 该算法每一次迭代的时间复杂度为  $O(n)$ , 其中  $n$  为采样点个数. 然而, 对于生成点画图等非均匀采样, 每个采样点 1-邻域范围的三角片重心及该区域重心的计算都要结合相应的密度函数, Step2 的计算要依赖 Step1 的计算结果, 而 Step1 中三角片重心计算的关键在于如何快速地寻找到每个三角片中包含的像素点与该像素对应的灰度值. 本文需要精确地计算出每个三角片重心的位置, 不能仅仅通过三角片 3 个顶点的像素位置与其相应密度值来进行计算, 因为像素是离散化的, 故采用栅格化的方法遍历求得重心位置.

### 2.3 算法优化

本文算法最初通过包围盒与 Bresenham 算法<sup>[24]</sup>相结合的方法来实现, 但是实验表明该步骤十分耗时, 进一步实验发现, 时间主要消耗在判断图像中像素点所属三角片这一步操作. 因此解决该步骤的运行效率将是提升算法效率的关键.

本文采用一种更快的基于渲染着色的实现方法: 利用 OpenGL 中的着色器并行地将每个三角片

渲染为不同的颜色, 并将每个三角片的索引值编码为该三角片相应的颜色值; 根据屏幕上每个像素点的颜色信息判断其所属的三角片, 达到快速分类像素的目的. 该方法的中间步骤如图 5 所示, 为了视觉上看得更清晰, 图中并没有将区域内所有的三角片着色后的效果体现出来. 从图 5d 中可以清晰地看出每个着色三角形中包含的像素, 完成这一步操作之后, 再通过三角片内像素对应的密度, 即每个像素的灰度值, 来求解出每个三角片的重心, 进而求得每个采样点对应的 1-邻域重心, 提高算法效率. 另外, 本文将对多幅图片作为输入进行非均匀采样, 并对比了使用渲染着色方法与栅格化方法分别采样所用时间对比.

如图 6 所示, 以图 5a 的图片作为输入, 对应采样点数为 2 048, 图 6a 所示为 CDT 算法分别从 0 迭代到 100 次所用时间对比, 可以看出, 渲染分类的方法用时明显比栅格化方法用时短, 且随着迭代次数的增多, 这种优势会越来越明显, 反映在图中则是渲染着色方法的斜率比栅格化方法的斜率小, 且采样点数越多, 2 种方法的斜率差越大. 图 6b 所示为 2 种方法迭代 30 次, 不同采样点数所用时间的对比, 可以看出, 采样点的数目对渲染着色方法的影响较小. 相同的迭代次数、不同采样点所用的时间相差较小, 反映在图中则是折线图斜率比较平缓. 实验结果表明, 在采样点为 16k 时, 对应的时间仅为 3.83 s; 而栅格化方法受到采样点数量的影响较大, 相同迭代次数的条件下, 不同采样点数目对应的的时间差很大, 采样点数为 16k 的时间已经达到 23.54 s. 此时, 渲染着色方法的优势就凸显出来了, 它更适用于大规模点集的采样分布, 实用性更强.

另外, 前文中通过局部地调整每个采样点在其一领域中的位置, 容易使采样点陷入局部最优状态. 因为最终结果很大程度上依赖于初始的输入, 所以需要充分随机化的分布作为输入.

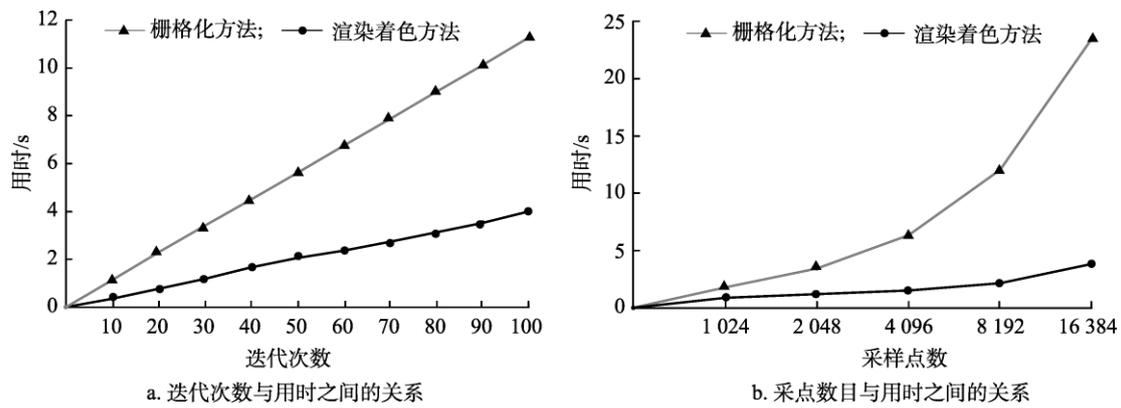


图 6 渲染着色方法与栅格化方法所用的时间对比

本文采用模拟退火算法(simulated annealing, SA)<sup>[25]</sup>,在局部稳定的情形下进行全局扰动,从而适当地减小非均匀初始化带来的影响.与其他全局优化方法一样,SA算法通常在计算代价上是昂贵的,它从初始状态开始,然后重复移动到当前状态的随机选择的邻居.通常,SA算法需要在达到全局最小值之前测试数千个状态.

本文对当前点集随机扰动后再进行局部优化,当它达到一个比上一个局部最优更好的状态时才将它移动到对应位置;否则,设定概率随机接受.本文称这一操作为“扰动优化”,每个采样点的扰动位置可以由

$$x_i^* = x_i + \lambda l_i (u \cos 2\pi v, u \sin 2\pi v) [t_1, t_2]^T \quad (5)$$

指定.其中, $l_i$ 是与点 $x_i$ 相连接的所有边的长度平均值, $u$ 和 $v$ 是2个范围在 $[0,1]$ 的随机数, $t_1$ 和 $t_2$ 是 $x_i$ 的切平面中的2个相互正交的单位向量.

事实上,SA算法的效率依赖于算法参数的选取,但没有适当选择这些参数的一般规则,它们取决于问题的性质.特别地, $\lambda$ 是在本文中的SA算法中唯一需要设置的参数,体现了扰动的量级.如果 $\lambda$ 太小,扰动优化后的结果将很可能与之前局部最小化的结果无异,计算工作将被浪费;如果 $\lambda$ 太大,则相当于随机初始化重新开始优化.本文经过大量实验发现, $\lambda$ 取值0.2时,SA算法在本文的所有试例中均能取得较好的效果.

## 2.4 算法讨论

本文提出的CDT算法能够产生优质的蓝噪声采样效果.式(2)所定义的目标函数是三角剖分中每个顶点到其1-邻域范围内点的加权距离之和.如前所述,极小化式(2)这个目标函数的解等价于该顶点位于其1-邻域范围的重心上,因此CDT能够产生这样的最优Delaunay三角剖分,即顶点之间尽可能均匀,但并不要求1-邻域的顶点个数,规

律性低,非常适用于蓝噪声采样.

基于Poisson盘最大化采样有这样一个事实,即当点集中每个点的Voronoi细胞都能够被其Poisson圆盘覆盖时,该点集即为最大化采样<sup>[8]</sup>.因此,对某个顶点来说,Voronoi细胞的最远角点必须在其1-邻域范围内,即顶点的1-邻域范围内没有细长的三角形.由于CDT采用Delaunay三角剖分,并且不断地将顶点移动到其1-邻域重心上,能够驱使点集均匀化,因此非边界附近的内部点都能满足上述条件;同时边界上的点也会移动到相应的中心位置,因此CDT在边界上的点足够多时能产生最大化的Poisson盘采样.

## 3 实验结果及分析

下面通过实验来说明本文算法快速生成优质蓝噪声采样分布的有效性.本文算法由C++实现,在3.20 GHz的Intel Xeon处理器,12 GB的RAM的PC机上运行.

### 3.1 频谱分析

频谱分析是用于评估点分布质量的常用方法,由Ulichney<sup>[1]</sup>首次用于抖动模式的研究,随后对采样点的分布周期图作平均化处理来估计其功率谱.周期图由该分布的傅里叶变换决定,每一次采样分布都有一个相应的周期图,通过对 $k$ 组采样分布的周期图取平均获取其功率谱.

具有蓝噪声性质的采样周期图是径向对称的,从功率谱可以推导出2个有用的一维统计数据:(1)径向平均频谱.该统计应该呈现出典型的蓝噪声性质,以低频截止的尖锐过渡区域开始,随后具有一系列逐渐衰减的结构波峰,最后是一个较为平坦的高频区域;(2)各向异性.用来衡量频谱的径向对称性.该统计应该平稳地保持在较低值处,

以表明优异的对称性. 频谱分析方法被 Lagae 等<sup>[26]</sup>用来比较不同方法生成泊松分布的差别. 后来 Schlmöer 等<sup>[27]</sup>扩展了文献[26]的工作, 提出一种更精确地用于分析平面上采样分布的频谱分析方法.

本文算法生成的均匀采样点分布与 Poisson 圆盘分布十分接近. 故在二维平面上, 采用 Schlmöer 等<sup>[27]</sup>提供的频谱分析工具(PSA)进行频谱分析, 进而验证本文算法生成的采样分布的蓝噪声声性质.

### 3.2 均匀采样

均匀采样即在采样区域内密度函数的值处处

相等. 为了对比不同方法生成的采样分布, 本文在单位正方形内均匀采样, 并将采样点个数设定为 1 024. 对于初始输入, 采用 10 组不同的采样数据集, 之后再对 10 组采样结果作平均化处理. 另外, 分别将蓝噪声采样领域比较流行的几种算法与本文算法对比, 包括经典的 CVT 算法<sup>[10]</sup>, CVT 算法的改进算法 CCVT 算法<sup>[13]</sup>, 采用 Power 图作为拓扑结构的 BNOT 算法<sup>[16]</sup>, 采用带容积约束的 Delaunay 三角剖分作为拓扑结构的 CCDT 算法<sup>[17]</sup>和 push-pull 算法<sup>[20]</sup>. 图 7 所示为不同的算法所生成

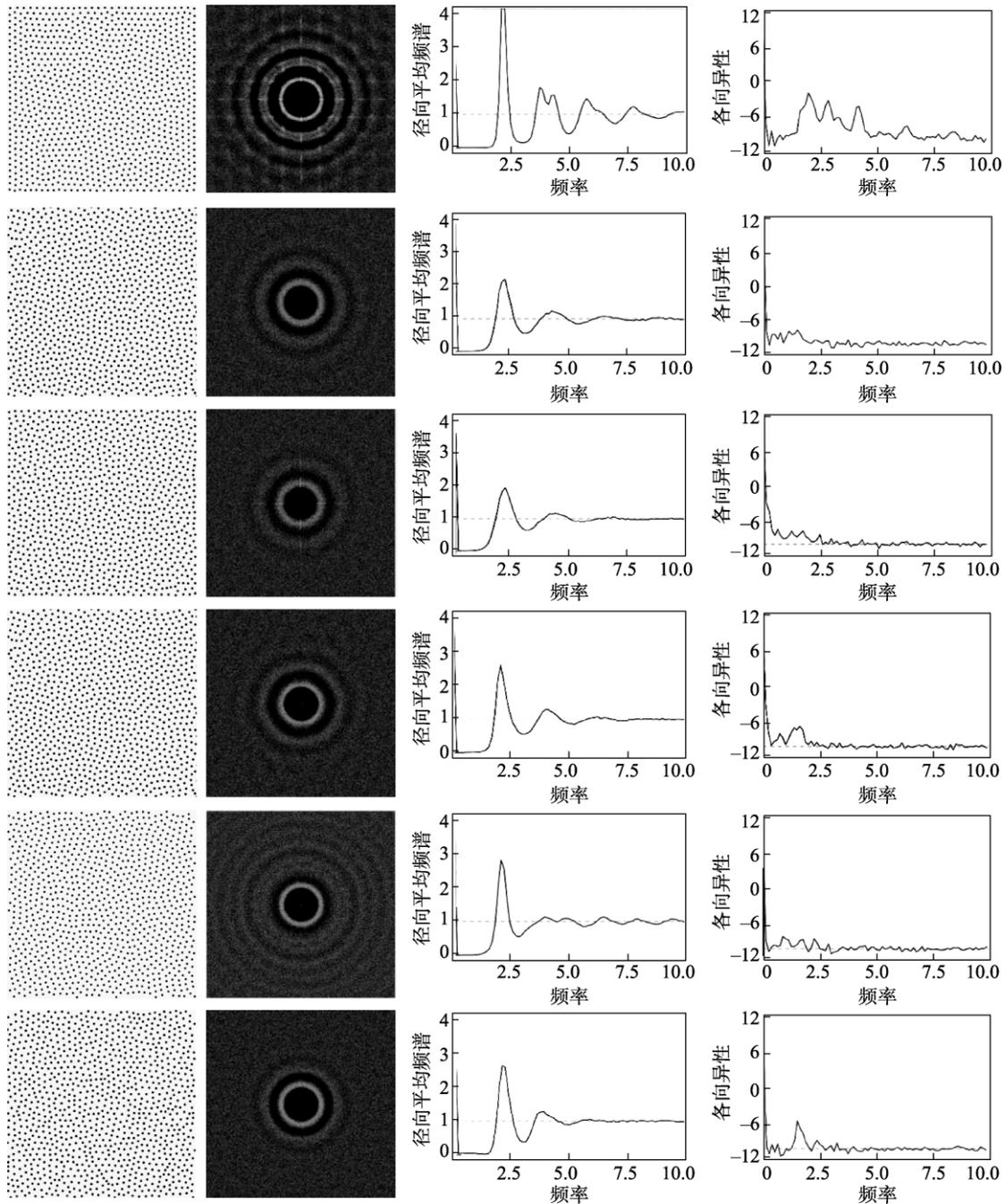


图 7 算法结果和频谱分析对比示意图

的采样点分布图、频谱分析图, 以及相应的一维上的径向平均功率谱与各向异性数据分析. 从上到下分别是 CVT, CCVT, BNOT, CCDT, push-pull 及本文算法得到的结果, 从左到右分别是最终点集分布, 点集对应的功率谱图, 径向平均频谱图及各向异性结果图.

从图 7 第 1 列的最终采样点集分布图可以看出, 6 种算法均满足蓝噪声应具有均匀性; CVT 算法生成的采样分布具有明显的规则性缺陷, 该缺陷会随着迭代次数的增多更加明显, 因其总是倾向于寻找全局最优解, 采样分布的效果会逐渐接近正六边形的平铺效果, 其径向平均功率谱和各项异性均不稳定, 故该算法不具有蓝噪声性质; 其余 5 种算法的功率谱均表现出了较好的蓝噪声特性, 相比 CVT 算法, 在规则性方面均有很大的改进; 从各向异性数据分析来看, 5 种算法的各向异性值最终均能平稳保持在一个较低值处, 尽管在图中 CCDT 算法与本文算法会在低频处会稍有波动, 但整体仍然位于较低的数值段内.

蓝噪声频谱分析并不是一个严谨的数学层面的定义, 它是从统计学层面给出的一种概念, 通过人类的视觉感知来达成某种共识, 目前也没有一种精准的评判规则来评判一种采样分布是否具有蓝噪声性质. 但从采样分布的整体效果来看, 蓝噪

声是指既具有空间分布的均匀性又无偏向性的一种分布. 本文算法与 CCDT 算法均是从一个充分随机的初始采样分布开始, 基于局部算法来优化采样点位置, 使算法的能量函数最终收敛到各个局部极小值, 而不是全局极小值. 这就解释了各向异性值比较低的原因, 因而不具有规则性这一缺陷. 可以说, 本文算法的采样分布具备良好的蓝噪声性质.

### 3.3 非均匀采样

非均匀蓝噪声采样可以直接应用于模拟非真实感图像绘制领域中的点画风格, 通过点的疏密程度不同来表现色调的明暗. 利用蓝噪声采样可以快速生成一幅图像对应的点画风格图, 即给定一张灰度图, 用点作为采样基元, 将图像的灰度值作为密度函数, 将采样点以不同的密度分布在图像区域中, 拟合原图中内容从而生成点画风格图像.

图 8 所示为基于非均匀 CDT 算法生成点画风格图像的过程, 采点数目为 4 096. 可以看出, 图 8a 虽然无法达到精准, 但确实可以模拟出对应输入图片的密度函数; 在图 8d 以后就开始呈现出质量较高的点分布效果, 基本达到稳定状态; 图 8e 之后的迭代中只有少量点需要更新位置以及相应的拓扑关系. 整个过程用时仅为 2.4s, 说明本文的算法在计算效率上十分有优势.



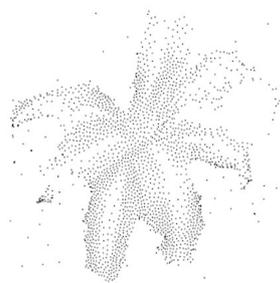
图 8 本文算法的非均匀采样过程

在图 9 中分别对比了 CCDT 算法、BNOT 算法、push-pull 算法和本文算法作用于相同输入图片生成非均匀点采样的效果, 从上到下依次采样数依次为 2k, 16k, 20k 和 8.5k. 由于 CCDT 算法在文献中只需 5 次迭代即可达到很好的效果, 但该算法每次迭代中都包含 5 次的几何优化, 故本文采

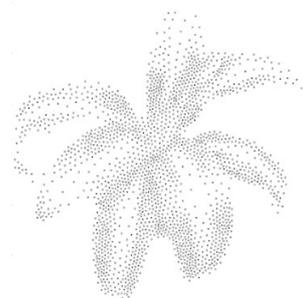
用迭代 25 次的效果用来对比, 与 BNOT 算法和 push-pull 算法的比较则根据它们各自的终止条件达到时对应的次数作为本文算法的迭代次数. 由于 CCDT 算法也是简单地移动顶点和构造 Delaunay 让每个三角形的加权面积尽量相等, 因此本文将主要与该算法进行对比.



a. 初始图片花



b. a 的 CCDT 算法最终采样分布



c. a 的本文算法最终采样分布



d. 初始图片 Lena



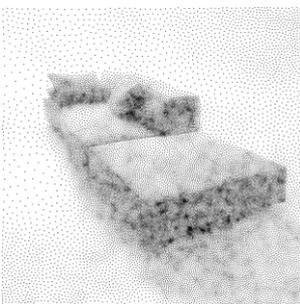
e. d 的 CCDT 算法最终采样分布



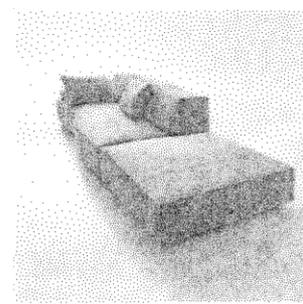
f. d 的本文算法最终采样分布



g. 初始图沙发



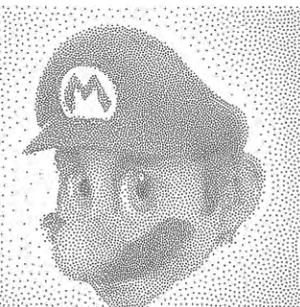
h. g 的 BNOT 算法最终采样分布



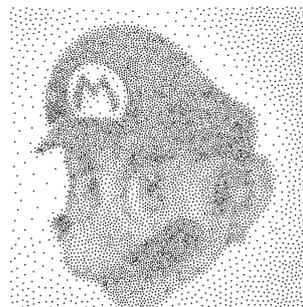
i. g 本文算法最终采样分布



j. 初始图片 Mario



k. j 的 push-pull 算法最终采样分布



l. j 的本文算法最终采样分布

图 9 4 种算法的点画图效果对比

对比图 9b 和图 9c 可以看出, 本文算法生成的效果明显好于 CCDT 算法, 因为本文将初始采样

点映射到图像边界, 并在迭代过程中保持边界上的点在边界上移动, 保证完整的拓扑结构, 并且在

最终输出采样分布图时边界点是不被考虑在内的。同样地,落在无效区域的采样点也会被处理掉,进而保证了高质量点画图的输出。对于 CCDT 算法,因为其算法迭代过程中边界在初始化采样时就已经确定好,并且在迭代过程中这些边界点的位置始终保持不变,所以在输出的采样点分布图中采样边界并没有被很好地保持,如图 9b 所示,CCDT 算法由于边界的处理不够完善,使得花瓣部分有一种残缺感;相比之下,本文算法针对这类图像的采样输出质量都比较高,如图 9c 所示。图 9e 中出现了明显的条纹,说明其产生了规整性缺陷,与蓝噪声采样的特性有些相悖;而图 9f 中不存在该问题,仍然呈现了一个较高质量的采样分布。另外本文算法还可以通过 SA 算法进行扰动优化,从而获得一个更加均匀的采样分布效果。

如图 9h 所示, BNOT 算法产生的采样效果在空白高亮处仍然体现出较好的均匀性,但在沙发的侧面及阴影处则出现了点相对集中的现象,这可能是由于该算法在顶点密集处的能量较低,表现出较低的优化能力,因此提早跳出优化框架;图 9i 效果则相反,在阴影处表现出更好的均匀效果。因为高亮处周围的点不断向非高亮处移动,使得该处的顶点数偏少;这与初始化也有一定的关系,总体来说,本文算法产生的点采样效果在一定程度上优于 BNOT 算法产生的效果。

从视觉效果上看,图 9k 所示的结果均匀性较好,由于 push-pull 算法本身对顶点位置的三重优化,使顶点之间的距离不会太小也不会太大;而本文算法并没有这种约束,因此比较容易陷入局部最优解,从图 9l 可以看出,有少部分点出现聚集的效果,虽然采用了 SA 算法,但较少的迭代次数很难求解全局最优值。

本文算法还与 CCDT 算法在时间消耗上进行对比,表 1 所示为 2 种算法对应不同采样点数的时间。本文在 Matlab 上运行 CCDT 算法,一般比 C++ 实现时间稍长。但从整体来看,本文算法在速率方面仍然有一定优势,耗时与采样点数之间基本呈现出线性关系,且随着点数的增多时间优势将会越来越明显。然而,本文算法也有不足之处,如表 1 中最后 2 行数据可知,采样点个数都为 16k 时,2 幅图像对应的时间却相差较大,也就是说会受到具体图像特征的影响,而 CCDT 算法在这方面则相对鲁棒。

表 1 与 CCDT 算法在不同采样点数时的时间对比

图片	采样点数	图像大小	算法耗时/s	
			CCDT	本文
花	2048	512×480	4.130	1.005
沙发	4096	468×468	7.934	1.589
小女孩	8192	1024×1037	19.667	3.448
小女孩	16384	1024×1037	40.538	4.128
Lena	16384	1024×1024	41.538	9.617

## 4 结 语

本文提出基于重心 Delaunay 三角剖分的蓝噪声点采样算法,用于生成具有优质蓝噪声性质的点分布采样。该算法在三角剖分中以每个采样点的 1-邻域为基本单位,采用一种简单的迭代算法,通过更新采样点位置与拓扑结构达到每个采样点均为 1-邻域区域的重心的目的;采用适应性初始采点与渲染分类等优化方法使该算法的效果与运行时间均得到进一步提升。

通过频谱分析,验证了利用本文算法的采样分布满足蓝噪声性质;此外,该算法具有算法原理简单、时间复杂度低、对大规模点集鲁棒等优势。目前,该算法仍有不足,在点的整体分布中会出现一些少量视觉上分布不均的情况;在几何优化部分也可以采样一些并行化方法,如 CPU 并行编程等进一步提高求解速率。

下一步的研究方向是将 CDT 算法推广到曲面,在曲面对采样点集进行基于重心 Delaunay 三角剖分,使采样分布在曲面上同样具有蓝噪声特性。

## 参考文献(References):

- [1] Ulichney R A. Dithering with blue noise[J]. Proceedings of the IEEE, 1988, 76(1): 56-79
- [2] Cook R L. Stochastic sampling in computer graphics[J]. ACM Transactions on Graphics, 1986, 5(1): 51-72
- [3] Wei L Y, Wang R. Differential domain analysis for non-uniform sampling[J]. ACM Transactions on Graphics, 2011, 30(4): Article No.50
- [4] Yan D M, Guo J W, Wang B, *et al.* A survey of blue-noise sampling and its applications[J]. Journal of Computer Science and Technology, 2015, 30(3): 439-452
- [5] Crow F C. The aliasing problem in computer-generated shaded images[J]. Communications of the ACM, 1977, 20(11): 799-805
- [6] McCool M, Fiume E. Hierarchical Poisson disk sampling distributions[C] //Proceedings of the Conference on Graphics In-

- terface 92. San Francisco: Morgan Kaufmann Publishers Inc, 1992: 94-105
- [7] Dunbar D, Humphreys G. A spatial data structure for fast Poisson-disk sample generation[J]. *ACM Transactions on Graphics*, 2006, 25(3): 503-508
- [8] Yan D M, Wonka P. Gap processing for adaptive maximal Poisson-disk sampling[J]. *ACM Transactions on Graphics*, 2013, 32 (5): Article No.148
- [9] Du Q, Faber V, Gunzburger M. Centroidal Voronoi tessellations: applications and algorithms[J]. *SIAM Review*, 1999, 41(4): 637- 676
- [10] Lloyd S P. Least squares quantization in PCM[J]. *IEEE Transactions on Information Theory*, 1982, 28(2): 129-137
- [11] Reinert B, Ritschel T, Seidel H, *et al.* Projective blue-noise sampling[J]. *Computer Graphics Forum*, 2016, 35(1): 285-295
- [12] Chen Zhonggui, Ouyang Yongsheng, Cao Juan. Feature-preserving method for mosaic image generation[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2014, 26(4): 520-527 (in Chinese)  
(陈中贵, 欧阳永昇, 曹娟. 特征保持的马赛克图像生成方法[J]. *计算机辅助设计与图形学学报*, 2014, 26(4): 520-527)
- [13] Balzer M, Schlmöer T, Deussen O. Capacity-constrained point distributions: a variant of Lloyd's method[J]. *ACM Transactions on Graphics*, 2009, 28(3): Article No.86
- [14] Li H W, Nehab D, Wei L Y, *et al.* Fast capacity constrained Voronoi tessellation[C] //Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. New York: ACM Press, 2010: Article No.13
- [15] Chen Z G, Yuan Z, Choi Y K, *et al.* Variational blue noise sampling[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2012, 18(10): 1784-1796
- [16] de Goes F, Breeden K, Ostromoukhov V, *et al.* Blue noise through optimal transport[J]. *ACM Transactions on Graphics*, 2012, 31(6): Article No.171
- [17] Xu Y, Liu L G, Gotsman C, *et al.* Capacity-constrained Delaunay triangulation for point distributions[J]. *Computers & Graphics*, 2011, 35(3): 510-516
- [18] Chen Zhonggui, Cao Juan, Yang Chenhui. Topology improvement for constructing optimal Delaunay triangulation[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2011, 23(12): 1967-1974(in Chinese)  
(陈中贵, 曹娟, 杨晨晖. 构造最优 Delaunay 三角剖分的拓扑优化方法[J]. *计算机辅助设计与图形学学报*, 2011, 23(12): 1967-1974)
- [19] Chen L, Xu J C. Optimal Delaunay triangulations[J]. *Journal of Computational Mathematics*, 2004, 22(2): 299-308
- [20] Ahmed A G M, Guo J W, Yan D M, *et al.* A simple push-pull algorithm for blue-noise sampling[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2017, 23(12): 2496-2508
- [21] Chen L, Holst M. Efficient mesh optimization schemes based on Optimal Delaunay Triangulations[J]. *Computer Methods in Applied Mechanics and Engineering*, 2011, 200(9-12): 967-984
- [22] Chen Z G, Cao J, Wang W P. Isotropic surface remeshing using constrained centroidal Delaunay mesh[J]. *Computer Graphics Forum*, 2012, 31(7): 2077-2085
- [23] Fabri A, Pion S. CGAL: the computational geometry algorithm library[C] //Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. New York: ACM Press, 2009: 538-539
- [24] Bresenham J E. Algorithm for computer control of a digital plotter[J]. *IBM System Journal*, 1965, 4(1): 25-30
- [25] Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by simulated annealing[J]. *Science*, 1983, 220(4598): 671-680
- [26] Lagae A, Dutré P. A comparison of methods for generating Poisson disk distributions[J]. *Computer Graphics Forum*, 2008, 27(1): 114-129
- [27] Schlmöer T, Deussen O. Accurate spectral analysis of two-dimensional point sets[J]. *Journal of Graphics, GPU and Game Tools*, 2011, 15(3): 152-160