

计算几何

Voronoi

① C语言

程序设计

32

86-87

Voronoi 图画法的改进与实现

The Improvement and Implementation of Voronoi Diagram Drawing Technique

张有会¹ 李秀丽² 杨立平³ 王运格⁴

024

TP311

(河北师范大学计算机系 石家庄050016)¹ (石家庄市科技干部进修学院 石家庄050031)²

Abstract Voronoi diagram is one of the most important branches in computational geometry. In this paper, pixel scanning as a new drawing method is proposed based on the traditional drawing method. Finally the source program written in C language is provided.

Keywords Computational geometry, Voronoi diagram, Pixel scanning

1 引言

计算几何在计算机辅助设计、计算机图形学及机器人等领域有着重要的应用。Voronoi 图是计算几何的一个重要分支。在计算几何中，Voronoi 图理论成功地解决了找最近点、求最大空圆、求 n 个点的凸包、求最小树等问题。另外，Voronoi 图在物理、生态、城市规划等许多领域都有重要应用。所谓 Voronoi 图，简单地说，就是对平面上任意给定的 n 个点，根据这些点的位置，将平面分割成 n 部分，得到一种对平面的分割图（如图1）。当平面上 n 个点给定以后，精确、高效地绘制出 Voronoi 图是 Voronoi 图领域的一个具有实际应用价值的研究方向。随着绘图工具的不断改进，特别是计算机软硬件技术的发展，绘图方法也在不断发展。本文在简要介绍 Voronoi 图的传统画法的基础上，给出了一种新的画法—像素扫描法，并用 C 语言进行了实现，给出了用 C 语言编写的源程序清单。

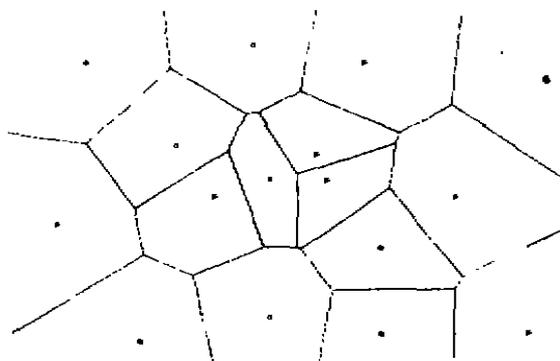


图1

2 Voronoi 图的数学定义和传统画法

2.1 Voronoi 图的数学定义^[1]

设 $P_i (i=1, 2, \dots, n)$ 为二维欧氏空间(平面)上的 n 个点，将由

$$V_i(P_i) = \bigcap_{j=1, j \neq i}^n \{P \mid d(P, P_i) < d(P, P_j)\} (i=1, 2, \dots, n)$$

所给出的对平面的分割，称为以 $P_i (i=1, 2, \dots, n)$ 为母点的 Voronoi 图，通常简称为 Voronoi 图，见图1，称 $V_i(P_i)$ 为 Voronoi 多边形 ($i=1, 2, \dots, n$)。其中 $d(P, P_i)$ 为 P 和 P_i 间的欧几里得距离。

2.2 传统画法简介

(1) 画法1(递归二分法^[2]):

第一步 把平面 S 划分为两个大小近似相等的子集 S1 和 S2;

第二步 递归地构造 $V(S1)V(S2)$;

第三步 “合并” $V(S1)$ 和 $V(S2)$ 得到 $V(S)$ 。

特点: ①从理论上讲, 效率比较高; ②对现代计算机软硬件来说, “合并”操作实现起来很困难; ③需保存大量相关数据, 内存占用量大。

(2) 画法2(逐次添加法^[1]):

第一步 对 P_1, P_2 两母点, 画 Voronoi 图(即画出 P_1, P_2 间的垂直平分线);

第二步 添加母点 P_3 , 利用对 Voronoi 图添加母点的方法得到由 P_1, P_2, P_3 三个母点构成的 Voronoi 图。

第三步 逐个母点添加, 直到所有母点添加完毕为止。

特点: ①理论上效率较高; ②随着画出母点个数的增多, 构造母点添加后的 Voronoi 图对现代计算机软硬件来讲, 处理起来很麻烦。③需保存大量数据, 内存

开销大。

3 像素扫描法及其实现

3.1 基本思想

从理论上讲,二维欧氏空间(平面)是由无穷多个点构成。然而用计算机作图,不论屏幕显示还是用输出设备(如打印机)输出,都是对有限个点操作。像素扫描法就是对屏幕上的每个像素点,计算其是否位于 Voronoi 边上。如是,就将该像素点用前景色画出,否则,该像素点就保持原色(背景色),从而在屏幕上画出了 Voronoi 图。

3.2 实现方法

首先画出 n 个母点 $P_i (i=1, 2, \dots, n)$; 然后对屏幕上每个像素点 $p(i, j) (i=1, 2, \dots, XMAX, j=1, 2, \dots, YMAX)$, 计算 p 和 $P_i (i=1, 2, \dots, n)$ 间的距离的最小值 m_1 和次最小值 m_2 。其中 $XMAX$ 和 $YMAX$ 分别为屏幕横纵方向的像素点数。如果 m_1 和 m_2 之差的绝对值小于某个很小的正数, 就将 p 用前景色画出, 否则, 让 p 保持原色。

3.3 C 源程序清单

```

1 #define XMAX getmaxx()
2 #define YMAX getmaxy()
3 #include "math. h"
4 #include "stdio. h"
5 #include "conio. h"
6 #include "graphics. h"
7 main()
8 { int graphdriver = DETECT, graphmode t, j, k, n, p1,
9 p2;
10 double d;
11 double a[100][2]; /* 数组 a 用来存放母点的坐标 */
12 double m1, m2; /* m1 和 m2 用来存放每次计算得到的最
13 小值和次最小值 */
14 clrscr(); /* 清屏 */
15 initgraph(&graphdriver, &graphmode, "");
16 /* 初始化图形系统 */
17 printf("input point num:");
18 scanf("%d", &n); /* 通过键盘输入母点的个数 */
19 for(i=0; i<n; i++)
20 scanf("%lf%lf", &a[i][0], &a[i][1]);
21 /* 通过键盘输入 n 个母点的屏幕坐标 */
22 cleardevice(); /* 清屏 */
23 rectangle(0, 0, XMAX-1, YMAX-1);
24 /* 沿屏幕四周画一矩形 */
25 for(i=0; i<n; i++)
26 circle(a[i][0], a[i][1], 1);
27 /* 在屏幕上画出 n 个母点 */
28 for(i=0; i<XMAX; i++)
29 for(j=0; j<YMAX; j++)
30 { m1 = m2 = sqrt(XMAX * XMAX + YMAX *
31 YMAX) + 1;
32 for (k=0; k<n; k++)
33 { d=(a[k][0]-i) * (a[k][0]-i) + (a[k][1]-
34 j) * (a[k][1]-j);
35 d=sqrt(d);
36 if(d<m2)
37 if(d<m1) {m2=m1; p2=p1; m1=d; p1=k;}
38 else {m2=d; p2=k;}
39 }
40 d=(a[p1][0]-a[p2][0]) * (a[p1][0]-a[p2]
41 [0]) + (a[p1][1]-
42 a[p2][1]) * (a[p1][1]-a[p2][1]);
43 d=sqrt(d);

```

```

34 if(fabs(m1-m2)<0.4 * d / ((m1>d)?m1:d))
35 {putpixel(i, j, 15); /* 在(i, j)处画一黑色点 */
36 m1=m2=10000;
37 }
38 getch();
39 closegraph();

```

关于源程序的说明

- (1) 本程序在 Turbo C 环境下编译通过。
- (2) 语句 10 定义的数组第一维的下标 100 为母点个数的上限, 该数值可根据母点个数进行调整。
- (3) 由语句 16、17 知, 母点的屏幕坐标为 $(a[k][0], a[k][1]) (k=0, 1, 2, \dots, n-1)$ 。
- (4) 语句 26、27 计算屏幕像素点 (i, j) 到母点 $(a[k][0], a[k][1])$ 之间的 Euclid 距离。
- (5) 语句 28~30 是将 (i, j) 点到 n 个母点的距离的最小值和次最小值分别赋给 m_1 和 m_2 , 同时记下距 (i, j) 最近的母点为 $(a[p1][0], a[p1][1])$, 距 (i, j) 次最近的母点为 $(a[p2][0], a[p2][1])$ 。
- (6) 语句 32、33 求两母点 $(a[p1][0], a[p1][1])$ 和 $(a[p2][0], a[p2][1])$ 之间的距离 d 。
- (7) 语句 34 判断 m_1 和 m_2 之差的绝对值是否足够小, 其中常量 0.4 可调, 其值大小决定画出的 Voronoi 边的粗细, 而乘以 $d / ((m1 > d) ? m1 : d)$ 是为了使 Voronoi 边画得均匀。

3.4 像素扫描法的特点

前已述及, 传统画法尽管在理论上效率较高, 但在计算机上实现起来, 都存在内存开销大, 程序编写复杂, 程序可读性差等问题, 与其相比, 像素扫描法具有以下特点:

- (1) 占用内存较少, 其实只需存放母点坐标和一些临时变量的内存空间即可。
- (2) 程序简洁、明快, 可读性好。
- (3) 算法适应面广, 不仅能用 C 语言及其系列语言(如 C++) 实现, 还可以用其他一些高级语言实现。
- (4) 使用简便, 使用时只需输入母点个数和母点坐标, 计算机自动画出母点和由其确定的 Voronoi 边。
- (5) 算法在理论上效率不高, 好在现代计算机运行速度都很高, 画图所用时间已不是主要矛盾。

参考文献

- 1 浅野孝夫, 计算几何学, bit 别册, 日本东京, 共立出版株式会社, 1986
- 2 Preparata F P, Shamos M L, 计算几何导论, 庄心谷译, 北京: 科学出版社, 1990
- 3 张有会, 川口凌, 母点到重みをつけた Voronoi 图, 日本国福井大学工学部研究报告, 1991, 39(2): 303~308
- 4 蒋国瑞, C 语言简明教程, 石家庄: 河北科学技术出版社, 1997