维普资讯 http://www.cqvip.com

计算机科学2001Vol.28№.6

加权 Voronoi 图画法的研究

A Method to Draw Weighted Voronoi Diagram

张有会

(河北师范大学计算机科学系　石家庄050016)

Abstract Voronoi Diagram is an important branch of computational geometry. Weighted Voronoi Diagrams are extensions of Voronoi diagrams. In this paper, a method for drawing weighted Voronoi diagrams is proposed. With this method, given generating points with their weights, the Voronoi diagram can be constructed by computing Voronoi arcs between two Voronoi vertices based on an incremental method. The obtained data of every closed domain are then stored in a circular linked list. The Voronoi arcs then are drawn according to the data stored in the lists. Thus a weighted Voronoi diagram is drawn.

Keywords Computational geometry, Voronoi diagram, Weight, Linked list

1 引言

随着计算机理论与应用研究的深入发展，尤其是计算机在图形图像处理方面的广泛应用，对计算几何理论与应用的研究，越来越受到重视并日益蓬勃开展起来。计算几何研究的是，如何高效处理通过视觉器官等途径获取的几何信息、从理论上探讨几何计算的可行性与复杂性，开发高速处理几何信息的方法，并对其性能做出评价。计算几何在计算机辅助设计，计算机图形学，图像处理、地理信息处理，机器人等许多领域都有重要应用。Voronoi 图是计算几何的一个重要分支，在计算几何的理论和应用中发挥着很大作用。另外，Voronoi 图在城市规划，优化配置、生态学、物理学等许多领域也有应用。

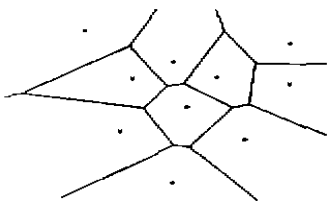

图1　Voronoi 图

任何一种 Voronoi 图的计算机作图问题，不论在理论上，还是在应用上，都是最基本、最重要的问题。对于点 Voronoi 图，已有如逐点添加法，二分法等成熟的算法。对于加权 Voronoi 图，文[2]给出了对平面上的像素点进行逐点扫描画法，并用 C 语言编写了源程序。该方法思路清晰，源程序代码少，可读性好，但效率太低($\Omega(n^3)$，n 为母点个数)。本文利用文[3]提出的画图思想，给出了加权 Voronoi 图的另一种画法。

为简化书写，在下面的叙述中，将"Voronoi"简记为"V-"，如"V-图"指的是"Voronoi 图"，"V-边"指的是"Voronoi 边"，等等。

2 V-图与加权 V-图

2.1 V-图的数学定义

定义1　设 $p_i(i=1,2,\cdots,n)$为二维欧氏空间(平面)上的 n 个互不相同的点，将由

$$V_n(p_i)=\bigcap_{j\neq i}\{p\mid d(p,p_i)<d(p,p_j)\}\quad (i=1,2,\cdots,n) \tag{1}$$

所给出的对平面的分割，称为以 $p_i(i=1,2,\cdots,n)$为母点(或生成元)的 V-图，通常简称为 V-图(图1)。其中 d(p,p)为 p 和 p_i 间的 Euclid 距离。

在 V-图中，平面被母点间的垂直平分线(段)分割成了 n 部分。

2.2 加权 V-图的数学定义

定义2　设 $p_i(i=1,2,\cdots,n)$为二维欧氏空间(平面)上的 n 个互不相同的点，$\lambda_i(i=1,2,\cdots,n)$是给定的 n 个正实数，称

$$V_n(p_i,\lambda_i)=\bigcap_{j\neq i}\left\{p\mid\frac{d(p,p_i)}{\lambda_i}<\frac{d(p,p_j)}{\lambda_j}\right\} \tag{2}$$

为点 p 的权重为 λ_i 的 V-区域，其中 $d(p,p_i)$为 p 和 p_i 间的 Euclid 距离，如果 $i\neq j$ 时，$\overline{V_n(p_i,\lambda_i)}\cap\overline{V_n(p_j,\lambda_j)}$非空且非单点集，则称$\overline{V_n(p_i,\lambda_i)}\cap\overline{V_n(P_j,\lambda_j)}$为 p_i 和 p_j 间的加权 V-边，其中$\overline{V_n(p_i,\lambda_i)}$是 $V_n(p_i,\lambda_i)$的闭包。两条以上加权 V-边的交点，称为加权 V-点，也就是加权 V-

维普资讯 http://www.cqvip.com

区域的顶点，将 $V_n(p_i,\lambda_i)(i=1,2,\cdots,n)$ 及其边界，称为在以 $p_i(i=1,2,\cdots,n)$ 为母点（或生成元），$\lambda_i(i=1,2,\cdots,n)$ 为权重的点上加权的 V-图，通常简称为加权 V-图，记作 V_n（图2，母点附近的数字表示母点的权重）。

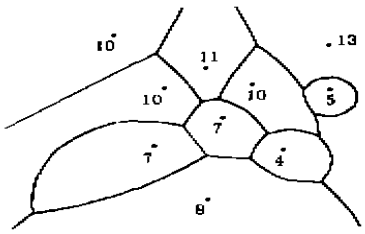

图2 加权 Voronoi 图

当 $\lambda_1=\lambda_2=\cdots=\lambda_n$ 时，(2)式等价于(1)式，即 V-图是加权 V-图当所有权重均相等时的特例。

在下面的叙述中，在不引起混淆的情况下，常将“加权”二字省略，将 $V_n(p_i,\lambda_i)$ 简记为 $V_n(p_i)$。

2.3 加权 V-图的性质

与本文相关的加权 V-图的几个重要性质如下[3]：

设 p_i 和 p_j 是加权 V-图 V_n 中的两个具有公共 V-边 l_{ij} 的母点，它们的权重分别为 λ_i 和 λ_j。

(1) 如果 $\lambda_i=\lambda_j$，则 l_{ij} 为线段 p_ip_j 的垂直平分线的一部分。

(2) 如果 $\lambda_i\neq\lambda_j$，则 l_{ij} 为圆或圆弧。设 l_{ij} 的圆心为 O，半径为 R，p_i 的坐标为 (x_i,y_i)，p_j 的坐标为 (x_j,y_j)，则 O、p_i、p_j 共线，且

① 圆心 O 的坐标为 $\left(\frac{\lambda_i^2x_i-\lambda_j^2x_j}{\lambda_i^2-\lambda_j^2},\frac{\lambda_i^2y_i-\lambda_j^2y_j}{\lambda_i^2-\lambda_j^2}\right)$；

② 半径 $R=\frac{\lambda_i\lambda_j}{|\lambda_i^2-\lambda_j^2|}d(p_i,p_j)$；

③ $d(O,p_i)=\frac{\lambda_i^2}{|\lambda_i^2-\lambda_j^2|}d(p_i,p_j)$，$d(O,p_j)=\frac{\lambda_j^2}{|\lambda_i^2-\lambda_j^2|}d(p_i,p_j)$；

④ p_i、p_j 关于圆 O 互为镜像点（因为 p_i、p_j 位于圆 O 的异侧，且 $d(O,p_i)d(O,p_j)=R^2$）。

3 加权 V-图的画法

加权 V-图画法的基本思想是，对给定的 n 个母点和权重，先用逐点添加的方法，求出各 V-区域边界上相邻两顶点间 V-边的与作图有关的数据，将每个 V-区域的闭合边界的数据存放在一个循环链表中，最后根据保存在各个循环链表中的数据画出 V-边，从而得到加权 V-图。为便于应用和实现，本文将加权 V-图限定在一矩形区域 D 内，在向循环链表中存放数据时，将 D 的每条边与 V-边同样对待。

3.1 数据结构

在给出具体画法之前，先介绍画法中有关数据的存储结构。

3.1.1 *母点和权重的数据结构* 将所有母点及其权重存放在一个由 $n\times3$ 个元素组成的二维数组中，n 为母点个数，数组的第 i 行元素存放母点 p_i 的横坐标 x_i，纵坐标 y_i 和权重 λ_i。

3.1.2 *V-区域边界的数据结构* 由于加权 V-区域的边界未必都是连续的，即一母点的 V-区域的边界可能由两个或两个以上互不相连的闭合曲线组成（如图3，母点 p_1 的 V-区域边界由两个闭合曲线 $(a_1a_2a_3a_4a_5a_6)$ 和 $(b_1b_2b_3b_4)$ 组成），为此，定义一个由 $n\times m$ 个元素组成的二维循环链表数组 $L_{n,m}$，存放 $V_n(p_i)$ 的第 j 个闭合曲线 l_{ij} 的数据，$(i=1,2,\cdots,n,j=1,2,\cdots,m_i,1\leqslant m_i\leqslant m)$，具体规定如下：

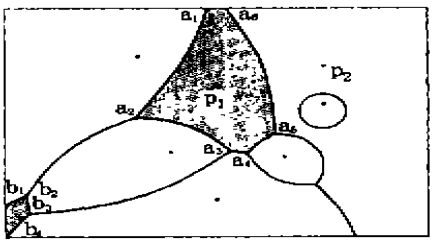

图3 母点 p_1，p_2 的 Voronoi 区域均有两个闭合的边界

(1) 从 l_{ij} 的任一顶点 e 开始，按围绕区域内部的逆时针顺序（即沿该顺序前进时，区域始终在其左侧），将从顶点 a 开始的边界数据，依次存放在 $L_{i,j}$ 的从头结点开始的各个结点中。第 k 个结点存放第 k 个顶点到第 k+1 个顶点之间的边界数据，$(k=1,2,\cdots)$。因为 l_{ij} 是闭合的，所以最后一个顶点的下一个顶点即为第1个顶点。

(2) 各循环链表的结点，存放形如 {x_0，y_0，angle1，angle2，r} 的数据，对于 V-区域边界上的任一条边 l：①如果 l 是圆或圆弧，则 (x_0,y_0) 为圆心坐标，angle1 和 angle2 为起始角和终止角，r 为半径。②如果 l 是线段，则令 r=-1，且 (x_0,y_0) 和 (angle1，angle2) 分别为围绕区域内部为逆时针顺序的线段的起点坐标和终点坐标。

3.2 准备工作

先给出作图时将要用到的两个基本方法。

3.2.1 *与一 V-区域相邻的全部 V-区域的查找方法* 设 p_c 为加权 V-图 V_m 中的一个母点，要想求出与 $V_m(p_c)$ 相邻的全部 V-区域，对 $V_m(p_c)$ 的每个连通区域 U 进行如下操作即可：

设 U 边界中的所有 V-边为 l_i，$(i=0,1,2,\cdots,t)$。

维普资讯 http://www.cqvip.com

做法如下.

(1)$j:=0$;

(2)如果 l 为线段($r<0$),则 p_0 关于 l 的对称点即为 V_m 中的一个母点,该母点的 V-区域即为与 $V_m(p_0)$ 相邻的 V-区域;

(3)如果 l 为圆($r>0$且 angle1-angle2为360的整数倍),则 $U=V_m(p_0)$,其周围的区域即为与 $V_m(p_0)$ 相邻的区域;找 p_0 关于圆 l 的镜像点,该点即为所求区域的母点;

(4)如果 l 为圆弧($r>0$且 angle1-angle2不是360的整数倍),求 p_0 关于圆弧 l 所在圆的镜像点,该点即为一母点,该母点的 V-区域即为 $V_m(p_0)$ 的以 l 为邻边的 V-区域.

(5)如果 $j<t$,$j:=j+1$,转(2),否则,结束.

3.2.2 查找一点所在的 V-区域的方法 设由母点 p_i,权重 λ_i,($i=1,2,\cdots,m$)所确定的加权 V-图为 V_m,现对平面上的一点 p,求 p 所在的 V-区域。

(1)在母点 p_i($i=1,2,\cdots,m$)中任选一母点,不妨设为 p_i,$k:=i$.

(2)在 V_m 中对所有与 $V_m(p_k)$ 相邻的 V-区域 $V_m(p_j)$,找满足 $\frac{d(p,p_j)}{\lambda_j}<\frac{d(p,p_k)}{\lambda_k}$ 的母点 p_j;

(3)如果这样的 p_j 不存在,则 p_k 即为所求,结束;否则,令 $k:=j$,转(2)。

3.3 创建和更新循环链表的过程

设 V_k 是由母点 p_i,权重 λ_i,($i=1,2,\cdots,k$)所确定的加权 V-图 设 L_{ij} 是循环链表,用来存放 $V_k(p_i)$ 第 j 个闭合的、相对于区域内部为逆时针排列的边界数据,($i=1,2,\cdots,k$,$j=1,2,\cdots,m_i$,$1\leqslant m_i\leqslant m$)。现在对于 V_k,添加母点 p_{k+1},构造 V_{k+1},为此需更新部分相关的循环链表并创建新循环链表 $L_{k+1,j}$($j=1,2,\cdots$)。

在 V-图 V_k 中,按3.2.2中的方法,找 p_{k+1} 所在的区域,设为 $V_k(p_{i0})$;按2.3给出的结果,求两母点 p_{k+1},p_{i0} 之间的 V-边,设为 l。以下分三种情形讨论:

3.3.1 l 与 $V_k(p_{i0})$ 无交点 此时,l 为圆(图4),它从 $V_k(p_{i0})$ 的某区域中圈出一个区域,即 l 所围的区域。该区域成为 p_{k+1} 的 V-区域,其余部分仍为 p_{i0} 的 V-区域。将 l 的数据按要求分别存放到 $L_{k+1,1}$ 和 L_{i0}、*(L_{i0} ,为 $L_{i0,j}$($j=1,2,\cdots,m_{i0}$)中编号最小的空循环链表)中即可.

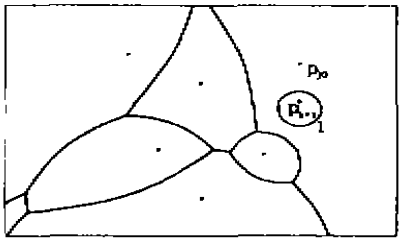

图4

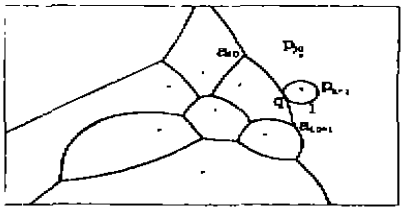

图5

3.3.2 l 与 $V_k(p_{i0})$ 只交于一点 此时 l 为圆(图5)。设 l 与 $V_k(p_{i0})$ 的某连通区域的边界相交,不妨设该连通区域边界的循环链表为 $L_{i0,1}$($=a_1a_2\cdots$),交点为 q,开辟新结点 a 存放 l 的数据。由于 l 与 V_k 只交于一点,所以有两种可能:l 与 $L_{i0,1}$ 的某个结点相交、或 l 与 $L_{i0,1}$ 的某条边相切。下面分别讨论:

(1)如果 l 与 $L_{i0,1}$ 的结点 a_{i2} 相交,即 $q=a_{i0}$。此时,更新 $L_{i0,1}$ 为($a_1a_2\cdots a_{i0-1}aa_{i0}a_{i0+1}\cdots$);创建 $L_{k+1,1}$ 为(a);

(2)如果 l 与 $L_{i0,1}$ 的两结点 a_{i0},a_{i0+1} 之间的边相切(图5),则更新 a_{i0} 存放 a_{i0} 和 q 之间的边界数据,开辟新结点 b 存放 q 和 a_{i0+1} 之间的边界数据;更新 $L_{i0,1}$ 为($a_1a_2\cdots a_{i0-1}a_{i0}aba_{i0+1}\cdots$),创建 $L_{k+1,1}$ 为(a)。

3.3.3 l 与 $V_k(p_{i0})$ 有两个交点 设 l 与 $V_k(p_{i0})$ 的某区域的边界相交,交点为 q_1,q_2,不妨设存放该边界数据的循环链表为 $L_{i0,1}=(a_1a_2\cdots)$,以下再分三种情况讨论:

1 q_1,q_2 均位于 D 的边界上

假设 q_1,q_2 相对于 $V_{k-1}(p_{i0})$ 的区域为逆时针顺序,开辟新结点 a 存放 q_1 为始点,q_2 为终点的 l 的数据,开辟新结点 b 存放 q_2 为始点,q_1 为终点的 l 的数据。

(1)如果 q_1 与 a_{i0} 重合,q_2 与 a_{i1} 重合,则更新 $L_{i0,1}$ 为($a_1a_2\cdots a_{i0-1}aa_{i1}a_{i1+1}\cdots$),创建 $L_{k+1,1}$ 为($a_{i0}a_{i0+1}\cdots a_{i1-1}b$)(图6)

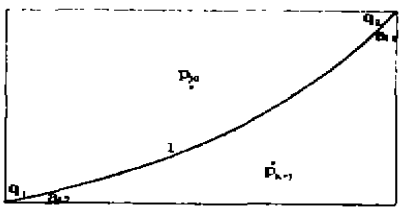

图6

(2)如果 q_1 位于 a_{i0},a_{i2+1} 间的边上,q_2 位于 a_{i1},a_{i1+1} 间的边上,则更新结点 a_{i0} 存放 a_{i0} 到 q_1 间的数据,更新结点 a_{i1} 存放 a_{i1} 到 q_2 间的数据,另开辟结点 a' 存放 q_1 到