基链分治算法与 Voronoi 区的面积计算定理研究*

付 庄[1], 王树国[1], 王剑英[2], 蔡鹤皋[1]

[1](哈尔滨工业大学 机器人研究所,黑龙江 哈尔滨 150001);

[2](中国科技大学 计算机科学与技术系,安徽 合肥 230027)

E-mail: fuzhuang@263.net

http://www.hit.edu.cn

摘要: 基于一般曲线多边形 Voronoi 图的面向对象数据结构,提出了一种改进的 Voronoi 图生成算法——基链分治算法.该算法与经典的分治法相比更容易被实现.同时,在欧氏米制中,由于 Voronoi 区的边界包含抛物线或双曲线,因而 Voronoi 区的面积很难被计算.为此提出了 Voronoi 区的面积计算定理,并给出了定理证明和算例,从而为某些工程应用中的面积计算提供了一种方法.

关键词: Voronoi 区;面向对象的数据结构;候选轮廓平分线;基链分治法

中图法分类号: TP391 **文献标识码**: A

Voronoi 图在 VLSI、机器人轨迹规划、形态表达与变换、曲面生成以及型腔加工等许多领域有着广阔的应用前景.因此,Voronoi 图及其相关技术作为一种理论性和实践性很强的图形学技术越来越受到国内外学者的关注.Blum[1]首先提出了关于 Voronoi 图平分线的概念.Persson[2]应用 Voronoi 图来生成一般型腔加工的等距线,并给出了 Voronoi 图平分线的参数方程表达式,但他对算法却未作任何描述.

Voronoi 图的构造算法有增量法、Lee 的分治法以及 Held 的波前-传播算法等.其中增量法简单,但却费时,它通过每次增加一个位置点来计算 Voronoi 图[3].Lee 的分治法[4]的时间复杂度为 $O(N\log N)$,算法很复杂.Held[5]提出并分析了波前-传播算法,并将其与 Lee 的分治法进行了比较.比较结果表明,波前-传播算法在某些方面优于分治法,例如,针对等距连接区,它的运算时间接近线性.但经分析后发现,其求交策略实际上也采用了 Lee 和 Drysdale 的扫描原理,而且对于病态情况有时会产生更坏的结果.相比之下,本文的基链分治法时间复杂度为 $O(N(h-1))$(其中 $h<N+1$),并更易于编程.

Voronoi 图数据结构的研究是 Voronoi 图算法研究的重要方面.Srinivasan. V.[6]等人的研究还只限于模块化编程,而关于面向对象的 Voronoi 图数据结构研究是一个比较新的课题.Voronoi 图有许多有趣的性质,Voronoi 区的"最近性"就是其中之一.利用面向对象技术研究 Voronoi 区的面积计算,对于 VLSI 的临界面积计算等实际问题具有一定的意义.

1 Voronoi 图分析

考虑一个封闭的平面单连通域 A,假设边界 c 由 n 个直线段或圆弧组成,则称该边界为一般曲

* 收稿日期: 1999-08-23; 修改日期: 2000-01-05

基金项目: 国家自然科学基金资助项目(69685006)

作者简介: 付庄(1972-),男,山东招远人,博士生,主要研究领域为 CG/CAD;王树国(1959-),男,黑龙江人,教授,博士生导师,主要研究领域为机器人技术,虚拟现实,计算机图形学;王剑英(1948-),男,安徽合肥人,副教授,主要研究领域为计算机图形学,CAD;蔡鹤皋(1934-),男,吉林长春人,教授,博士生导师,中国工程院院士,主要研究领域为机器人学.

线多边形.假设外轮廓的方向为逆时针,内轮廓的方向为顺时针,即沿边界方向运动时,区域 A 永远在边界元素的左侧.下面我们来介绍几个相关概念[5].

定义 1(边界元素). 直线段、圆弧是最基本的边界元素.如果两个相连的边界元素所夹的内角大于 π,则内角顶点称为优顶点,优顶点实质上是一个半径为 0 的圆弧,它也是一种边界元素.

定义 2(影响区). 当沿边界元素 e_i 的正方向运动时,其左侧有效区域称为该元素的影响区,记为 $CI(e_i)$.

定义 3(平分线、轮廓平分线和节点). 边界 c 在 A 内的平分线(bisector)是一系列点集,点集内每个点至少与两个边界元素距离相等.相邻边界元素之间的平分线以边界元素的一个控制点为起点,称为轮廓平分线(contour bisector).平分线之间的交点称为节点.

定义 4($V_R(e,c)$). Voronoi 图的平分线把 A 分成许多仅与一个边界元素 e 相关的子区域,称为 Voronoi 区,记为 $V_R(e,c)=\{p\in CI(e):d(p,e)\leqslant d(p,c)\}$.

定义 5(峡,单调区,内点). Voronoi 图上不相邻的边界元素之间的局部最窄点称为峡(strait).单调的平分线所在的 Voronoi 图局部区域称为单调区.单调区内有且只有一个间距最大的节点,称为内点.

2 面向对象的 Voronoi 图数据结构

Voronoi 图的边界元素、节点、平分线以及包含这些要素的多连通域之间有着密切的关系,其数据结构非常复杂.我们利用面向对象的数据结构技术来定义 Voronoi 图,这不但有利于对 Voronoi 图的理解,而且简化了 Voronoi 图的构造过程.下面我们将一一介绍 Voronoi 图的各个定义类.

2.1 TMultiConnectedPolygon

Voronoi 图最重要的数据是多连通多边形,定义为类 TMultiConnectedPolygon,它包括外轮廓(OutContour)、内轮廓集(InnerContours)、Voronoi 图(TheVoronoi)、候选节点集(Intersections)、峡集(Straits)、Voronoi 图处理完毕标志量(VoronoiIsComputed)等属性.其中,内轮廓集是用来存放多个单连通域的队列,队列指针指向当前被计算的元素;TheVoronoi 用来存放被计算的 Voronoi 图;候选节点集用来存放算法初始化阶段可能的 Voronoi 节点;峡集是用来存放多连通域峡的队列;若 Voronoi 图处理完毕,标志量 VoronoiIsComputed 置 True.可见,类 TMultiConnectedPolygon 几乎容纳了计算 Voronoi 图的所有信息,再通过合适的函数,若生成平分线的函数、平分线的求交函数、峡的有效性判别函数、候选节点集的初始化函数以及合并平分线的扫描函数等,就从整体上完成了 Voronoi 图数据结构的定义.下面具体介绍各个子类的情况.

2.2 TContourObject

内外轮廓集中的每个元素是边界段组成的线性表,表中的元素被定义为类 TContourObject.它包括边界类型(ObjectTag)、边界序号(No.)、边界参数(Param)、边界的控制点(ControlPoints)以及起始链(StartBisectorChain)和终止链(EndBisectorChain)等属性.其中 ObjectTag 可取直线段(OLine)、优顶点(OReflex)或圆弧(OArc)等具体类型.StartBisectorChain 和 EndBisectorChain 用来存放边界元素的左、右平分线链表,通过访问该链表就可以取得平分线的起始点、终止点和类型等信息.

2.3 TBisector

Voronoi 图的平分线被定义为类 TBisector.它包括平分线的类型(BisectorTag)、参数

(Param)、取舍(IsDiscard)、起始节点(StartNode)、终止节点(EndNode)、左定义边(LeftDefineObject)及右定义边(RightDefineObject)等属性.属性 Param 用来保存不同类型平分线的方程参数.根据左、右定义边,平分线的分类见表 1.

Table 1 The type of bisector

表 1 平分线的类型

Type①	Left and right definition boundary②
Straight line segment③(BLine)	Two non-parallel straight line segments, or the reflex vertex and its adjacent straight line segment④
Parallel line segment⑤(BPLine)	Two parallel and non-overlapping straight line segments⑥
Parabolic segment⑦(BParabola)	The straight line segment and the reflex vertex, or the straight line segment and the circular arc⑧
Concentric circular arc⑨(BPCircle)	Two concentric circular arcs with different radii⑩
Hyperbolic segment⑪(BHyperbola)	Four combinations of the reflex vertex and the circular arc⑫

①类型,②左、右定义边,③直线段,④两条不平行的直线段或优顶点以及相邻的直线段,⑤平行线段,⑥两条平行且不重合的直线段,⑦抛物线段,⑧直线段和优顶点,或直线段和圆弧,⑨同心圆弧,⑩两个半径不同的同心圆弧,⑪双曲线段,⑫优顶点和圆弧的 4 种组合.

2.4 TVoronoiNode

Voronoi 图的节点被定义为类 TVoronoiNode.该类包括位置(Position)、与边界的间距(Clearance)、取舍(IsDiscard)、相关的平分线队列(AssociatedBisectors)以及阶(Rank)等属性.与节点相连的平分线个数称为节点的阶,最终的 Voronoi 图节点至少是三阶的.在 Voronoi 图的生成过程中,有的平分线可以暂时终止于无穷远处,因此它的终止节点是一阶的.通过访问相关的平分线队列,我们就可以找到与节点相连的平分线.

2.5 TStrait

Voronoi 图的峡使 Voronoi 图的等距连接区是分离的.峡被定义为类 Tstrait,它包括真伪布尔量(IsReal)、峡所在的平分线(Strai:Bisector)、到边界元素的间距(Clearance)以及位置(Position)等属性.

2.6 TVoronoi

Voronoi 图定义为类 TVoronoi,它包括节点集(VoronoiNodes)和平分线集(Bisectors)等属性.

3 算法初始化概述

本节主要介绍单连通域的 Voronoi 图生成算法.在开始基链分治法之前,要进行许多初始化工作.例如,从图形 CAD 系统中读入多边形的边界数据、判断边界元素的类型和生成轮廓平分线等.比较重要的初始化工作还有候选轮廓平分线的判断[6](终止节点是最终 Voronoi 图的一部分)、峡的判断和边界元素子链的划分等.

3.1 求候选的轮廓平分线

取$\otimes$为平分线的求交符号.依次取 4 个连续边界元素 e_i,e_{i+1},e_{i+2}和 e_{i+3},则有 $v_1(e_i,e_{i+2})=b(e_i,e_{i+1})\otimes b(e_{i+1},e_{i+2})$,表示平分线 $b(e_i,e_{i+1})$与 $b(e_{i+1},e_{i+2})$相交于 v_1 点,接下来一条可能的平分线始于 v_1 并以 e_i 和 e_{i+2}为定义边.同样地,有 $v_2(e_{i+1},e_{i+3})=b(e_{i+1},e_{i+2})\otimes b(e_{i+2},e_{i+3})$.于是,如果 $d(v_1,e_{i+1})<d(v_2,e_{i+1})$,则取节点 v_1 为一个候选节点,将其存入候选节点集,并使相应的两条候选轮

廓平分线终止于该节点.同样,如果 $d(v_1,e_{i+1})>d(v_2,e_{i+1})$,则取节点 v_2.如果 $d(v_1,e_{i+1})=d(v_2,e_{i+1})$,则取节点 v_1 或 v_2,下一条平分线的定义边为 e_i 和 e_{i+3},同时,3 条候选的轮廓平分线终止于该节点.令 i 递增,重复上述过程,得到候选节点集.将其按节点间距递增排序,就完成了这步初始化工作.

3.2 求 峡

Held 认为,在生成 Voronoi 图以后,令间距递增的方向为平分线的方向,如果找到 Voronoi 图的所有内点,则相邻的内点之间一定有峡存在.根据"峡线"是以峡为间距最低点这个特征,通过图形搜索方法就可以找到峡.但分析发现,在 Voronoi 图没有被求出的情况下,上述方法在实现上有一定困难.根据峡的数学特征,我们给出了一种在初始化阶段求峡的方法.该方法遵循如下原则:

· 峡产生于抛物线或双曲线等"峡线"上,即优顶点和直线、优顶点和优顶点、优顶点和圆弧以及在圆弧和圆弧之间产生峡;

· 峡一定在它所属的两个边界元素的影响区内;

· 除了峡所属的边界元素以外,没有比到峡间距更小的边界元素存在.

Voronoi 图求解完毕后,还可根据下列原则加以验证,即

· "峡线"的两个节点本身是内点;

· 或先找到与"峡线"的两个节点相连的其他非轮廓平分线,再判断该平分线的另一个节点是否为内点;

· 或"峡线"的两个节点分别满足上述两种情况.

峡找到后,"峡线"就被峡分割成两条单调的平分线.

3.3 划分边界元素子链

在 Voronoi 图生成算法之前,还要把边界元素分成多个子链[4].由于开始搜索的起始元素可能不是第 1 条子链的第 1 个元素,所以,首先应该建立一条中间子链,最后再将它与最后一条子链比较、合并.如果边界元素类型是优顶点,则优顶点的左右元素都被归入一条子链中,否则将建立下一条新的子链.

4 基链分治算法(main-chain divide-and-conquer algorithm)

假设轮廓 c 包括 N 个边界元素,初始化完毕后轮廓 c 被划分成 h 条子链:$C_1,C_2,C_3,\ldots,C_h$,其中每条子链包含的元素个数为 $t_1,t_2,t_5,\ldots,t_h$.经典的分治法用递归的方法,先求第 1 层每对子链的 Voronoi 图(最后剩余的一条单独子链留给下一层),再将每对子链合并成一条子链,重复上述操作,直到合并成两条独立的子链,再求出最后的合并平分线.由于该递归方法实现起来很复杂,我们提出了一种改进算法,称为基链分治算法(简称 MCDCA).首先,用子链 C_1 初始化基链(main-chain),然后求基链与 C_2 之间的合并平分线,随后将 C_2 加到基链中去.重复上述过程,直到基链与最后一条子链合并为止.算法描述与合并树如图 1 和图 2 所示.

具体的合并过程如下:令边界元素的左、右平分线链表(见第 2.2 节)为 S 和 E.设合并平分线为 $b(e_i,e_{i+1})$,先令它与 e_i 的 S 相交,产生 0、1 或 2 个交点,其中最多只有一个有效点.再与 e_{i+1} 的 E 相交,产生类似的结果.如果左、右均有有效点,则取 $b(e_i,e_{i+1})$ 先遇到的点.如果左、右的两个有效点重合或左、右只有一个有效点,则取重合点或惟一的有效点[4].如果求交的两条平分线是轮廓平分线,则还要判断其是否有终止节点.如果有终止节点,则判断该节点是否在候选节点集内.如果在

候选节点集内并且上述交点的间距大于候选节点的间距,则本次求交无效,直接将子链加到基链中,从而大大减少不必要的合并工作.如果合并平分线 $b(e_i,e_{i+1})$ 与 S 中的平分线 $b(e_l,e_r)$ 相交于有效节点 v,则下一条合并平分线 $b(e_l,e_{i+1})$ 从 v 出发,记为 $v(e_l,e_{i+1})=b(e_l,e_i)\otimes b(e_i,e_{i+1})$;如果合并平分线 $b(e_i,e_{i+1})$ 与 E 的平分线 $b(e_{i+1},e_r)$ 相交于有效点 v,则下一条合并平分线 $b(e_i,e_r)$ 从 v 出发,记为 $v(e_i,e_r)=b(e_i,e_{i+1})\otimes b(e_{i+1},e_r)$.

```
Mainchain→Add(C1);
for i=2 to h do
begin
    VD(mainchain+Ci)=merge (mainchain, Ci);
    mainchain→Add (Ci);
end
```

Fig. 1 Main-Chain divide-and-conquer algorithm
图 1 基链分治算法

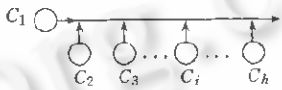


Fig. 2 The merge tree of the MCDCA
图 2 基链分治算法的合并树

可见,基链分治算法采用非递归的方法,更易于编程实现.算法的时间复杂度分析如下:

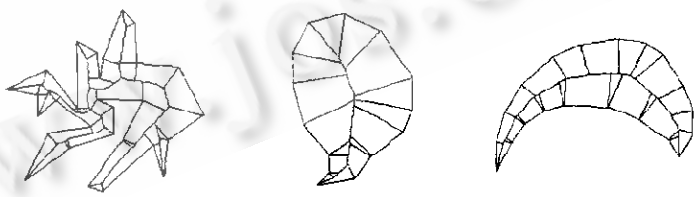
考虑 $t_1+t_2+t_3+\ldots+t_h=N$,有

$$\sum_{j=2}^{h}\sum_{i=1}^{j}O(t_i)\leqslant\sum_{j=2}^{h}\sum_{i=1}^{h}O(t_i)=\sum_{j=2}^{h}O(N)=O(N\cdot(h-1)).$$

因此,算法的时间复杂度上限为 $O(N(h-1))$,其中 $h<N+1$.从上面的合并过程可见,由于每条轮廓平分线都与其候选节点进行比较,避免了大量的合并操作,所以该时间复杂度分析是比较保守的.

5 基链分治法实验

为了从实验上验证基链分治法的有效性,我们设计了一个可以生成任意自由曲线的 CAD 图形系统,将自动生成的多边形分为随机、平滑和更平滑 3 种类型.生成的 Voronoi 如图 3 所示.实验是用相同的计算机和编程语言完成的.图 3 中,边数为 29,15 和 20 的多边形生成 Voronoi 图分别耗时 24ms,13ms 和 17ms.实验表明,基链分治法生成的 Voronoi 图十分精确,算法的效率较高.



(a) Random (b) Smooth (c) Smoother
(a) 随机 (b) 平滑 (c) 更平滑

Fig. 3 Three classes of the tested polygons
图 3 被测试的 3 类多边形

6 Voronoi 区的面积计算定理

6.1 定 理

设边界元素 e_i 的 EndBisectorChain 与 StartBisectorChain 分别有 k 和 h 条平分线,则

$$\text{EndBisectorChain}=\bigcup_{ii_i=0}^{k-1} b_{ii}(e_i,e_j),\ \text{StartBisectorChain}=\bigcup_{ii_i=k}^{k+h-1} b_{ii}(e_i,e_j),$$

其中 e_j 是平分线的另外一条定义边($i\langle\rangle j$). 为了方便起见,下文 $b_{ii}(e_i,e_j)$ 的下标 ii 从略. 于是,Voronoi 区的面积 $S_i=S_1^*+S_2^*+S_3^*$. 其中 S_1^*,S_2^*,S_3^* 分别表示 EndBisectorChain,StartBisectorChain 与边界元素 e_i 到 x 轴的投影区所包围的面积,公式如下:

$$S_1^*=-\sum_{i_1=0}^{k-s_e-1}\left(\sum_{t_i=t_{ie}}^{t_{ie}} y(t)\cdot[x(t+\mathrm{d}t)-x(t)]\right)-\sum_{i_1=k-s_e}^{k-1}\left(\sum_{t_i=t_{is}}^{t_{strait}} y(t)\cdot[x(t+\mathrm{d}t)-x(t)]+\sum_{t_i=t_{strait}}^{t_{ie}} y(t)\cdot[x(t+\mathrm{d}t)-x(t)]\right), \tag{1}$$

$$S_2^*=-\sum_{i_1=k}^{k+h-s_s-1}\left(\sum_{t_i=t_{ie}}^{t_{is}} y(t)\cdot[x(t+\mathrm{d}t)-x(t)]\right)-\sum_{i_1=k+h-s_s}^{k+h-1}\left(\sum_{t_i=t_{ie}}^{t_{strait}} y(t)\cdot[x(t+\mathrm{d}t)-x(t)]-\sum_{t_i=t_{strait}}^{t_{is}} y(t)\cdot[x(t+\mathrm{d}t)-x(t)]\right), \tag{2}$$

$$S_3^*=-\sum_{x_i=x_s}^{x_e} y(x)\cdot\mathrm{d}x. \tag{3}$$

s_e 与 s_s 分别表示 EndBisectorChain 与 StartBisectorChain 的峡数,t_{is},t_{ie} 和 t_{strait} 分别表示平分线的起始节点、终止节点及峡与边界的间距. x_s 与 x_e 表示边界元素的起点与终点的横坐标.

6.2 推 论

一般曲线多边形的面积等于各 *Voronoi* 区的面积之和.

6.3 定理证明

因为 Voronoi 区的边界由 EndBisectorChain 与 StartBisectorChain 组成. EndBisectorChain 和 e_i 的方向是逆时针的,StartBisectorChain 的方向是顺时针的. 假设 e_i 的类型是直线段,则根据格林(Green)公式有

$$S_i=\iint_{\overline{\lambda}_i}\mathrm{d}x\mathrm{d}y=-\int_C y\mathrm{d}x=-\sum_{i_1=0}^{k-s_e-1}\int_{C_i} y\mathrm{d}x-\sum_{i_1=k-s_e}^{k-1}\left(\int_{C_{i1}} y\mathrm{d}x+\int_{C_{i2}} y\mathrm{d}x\right)-\sum_{t_i=k}^{k+h-s_s-1}\int_{C_i} y\mathrm{d}x-\sum_{i_1=k+h-s_s}^{k+h-1}\left(\int_{-C_{i1}} y\mathrm{d}x+\int_{-C_{i2}} y\mathrm{d}x\right)-\int_{C_{ei}} y\mathrm{d}x.$$

式中 c 表示 A_i 的边界轮廓,c_{i1},c_{i2} 表示峡的左、右平分线,$-c$ 表示反向的 StartBisectorChain(即积分区域是$[t_{ie},t_{is}]$). 把 x 与 y 的参数方程代入,根据定积分的定义将上式离散化,从而 Voronoi 区的面积计算定理得证(如果 e_i 是圆弧,令 y 对 x 取极值,使 x 递增时,y 单调变化,则在单调区间内证明仍成立). 计算的误差主要是由差分的舍入误差引起的,在效率允许的情况下,减小积分步长 $\mathrm{d}t$ 可以减小舍入误差.

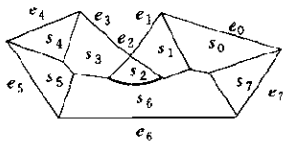


Fig. 4 A sample for the area computation of Voronoi cells

图4 Voronoi区的面积计算示例

6.4 算 例

以有一个峡的多边形为例,多边形的边界元素数据如图 4 和表 2 所示.

Table 2 Data of the contour elements

表 2 边界元素数据

No.①	e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7
Type②	Oline	Oline	Oreflex	Oline	Oline	Oline	Oline	Oline
Start point③	(515,145)	(392,122)	(341,181)	(341,181)	(277,112)	(169,142)	(238,230)	(460,230)
End point④	(392,122)	(341,181)	–	(277,112)	(169,142)	(238,230)	(460,230)	(515,145)

①序号,②类型,③起点,④终点.

表 3 中 S 和 E 分别表示起始链和终止链,t 表示平分线终止节点的边界间距,s_i 表示第 i 个 Voronoi 区的面积.抛物线在峡处($b(e_2,e_6)$的中点)被分成两段,t_{strait}表示峡距离边界的间距,$dt=0.001$. 对各 Voronoi 区的面积求和,有 $S=s_1+\ldots+s_7=26967.4540267928$,与真实值 $S'=26967.5$相减得舍入误差 0.045 97.一般地,适当调整步长,计算误差就可以限制在允许的范围内.

Table 3 The bisector, its end node clearance and the area of Voronoi cell for each contour element

表 3 每个边界元素的平分线、平分线的终止节点间距与 Voronoi 区的面积

Boundary①	Bisector②	The clearance of the end node and the area③	Boundary	Bisector	The clearance of the end node and the area
e_0	$E:b(e_0,e_1)$ $S:b(e_7,e_0)$ $S:b(e_6,e_0)$	$t=51.7104801641265$ $t=49.7412537125597$ $t=51.7104801641265$ $s_0=3692.37054678657$	e_1	$E:b(e_1,e_2)$ $S:b(e_0,e_1)$ $S:b(e_6,e_1)$	$t=29.6259813544904$ $t=51.7104801641265$ $t=29.6259813544904$ $s_1=2731.7529319667$
e_2	$E:b(e_2,e_3)$ $S:b(e_1,e_2)$ $S:b(e_6,e_2)$ $S:b(e_6,e_2)$	$t=29.1659126262907$ $t=29.6259813544904$ $t=t_{\text{strait}}=24.5$ $t=29.1659126262907$ $s_2=572.279935724535$	e_3	$E:b(e_3,e_4)$ $S:b(e_2,e_3)$ $S:b(e_6,e_3)$ $S:b(e_5,e_3)$	$t=53.1814592819843$ $t=29.1659126262907$ $t=53.4349767269919$ $t=53.1814592819843$ $s_3=3485.6338561321$
e_4	$E:b(e_4,e_5)$ $S:b(e_3,e_4)$	$t=53.1814592819843$ $t=53.1814592819843$ $s_4=2980.69734020246$	e_5	$E:b(e_5,e_6)$ $E:b(e_5,e_3)$ $S:b(e_4,e_5)$	$t=53.4349767269919$ $t=53.1814592819843$ $t=53.1814592819843$ $s_5=3138.7471913408$
e_6	$E:b(e_6,e_7)$ $E:b(e_6,e_0)$ $E:b(e_6,e_1)$ $E:b(e_6,e_2)$ $E:b(e_6,e_2)$	$t=49.7412537125597$ $t=51.7104801641265$ $t=29.6259813544904$ $t=t_{\text{strait}}=24.5$ $t=29.1659126262907$	e_6	$E:b(e_6,e_3)$ $S:b(e_5,e_6)$	$t=53.4349767269919$ $t=53.4349767269919$ $s_6=7848.12616403341$
			e_7	$E:b(e_7,e_0)$ $S:b(e_5,e_7)$	$t=49.7412537125597$ $t=49.7412537125597$ $s_7=2517.84606060626$

①边界,②平分线,③终止节点的间距和面积.

7 结 论

本文提出了一种改进的分治算法——基链分治法,以及面向对象的 Voronoi 图数据结构,提出了 Voronoi 区的面积计算定理.算法的复杂度分析和实验证明,基链分治法思路清晰、编程简单,生成的 Voronoi 图准确无误,是一种值得推荐的 Voronoi 图生成算法.Voronoi 区的面积计算定理为 Voronoi 区面积的求取提供了理论依据,在工程中具有一定的应用价值.

References:

[1] Blum, H. A transformation for extracting new descriptors of shape. In: Whaten-Dunn, W. ed. Proceedings of the

Symposium on Models for Perception of Speech and Visual Form. Cambridge, MA: MIT Press, 1967. 362~380.

[2] Persson, H. NC machining of arbitrarily shaped pockets. Computer-Aided Design, 1978,10(3):169~174.

[3] Ohya, T., Iri M., Murota K. Improvements of the incremental method for the Voronoi diagram with computational comparison of various algorithms. Journal of Operation & Research Society, 1984,27(4):306~336.

[4] Lee, D. T. Medial axis transformation of a planar shape. IEEE Transactions on PAMI, 1982,4(4):363~369.

[5] Held, M. Voronoi diagrams and offset curve of curvilinear polygons. Computer-Aided Design, 1998,30(4):287~300.

[6] Srinivasan, V., Nackman, R. Voronoi diagrams for multiply connected polygonal domains-I. Algorithm IBM Journal of Research and Development, 1987,31(3):361~372.

The Main-Chain Divide-and-Conquer Algorithm and the Area Computation Lemma for Voronoi Cell*

FU Zhuang[1], WANG Shu-guo[1], WANG Jian-ying[2], CAI He-gao[1]

[1](*Robotic Research Institute, Harbin Institute of Technology, Harbin* 150001, *China*);

[2](*Department of Computer Science and Technology, University of Science and Technology of China, Hefei* 230027, *China*)

E-mail: fuzhuang@263.net

http://www.hit.edu.cn

Abstract: Based on the object-oriented data structure of Voronoi diagram for normal curvilinear polygon, an improved Divide-and-Conquer algorithm is presented in this paper for Voronoi diagram generation called Main-Chain Divide-and-Conquer algorithm. It is easier to implement compared with the classical one. Meanwhile, because the boundary of a Voronoi cell contains parabolic or hyperbolic curves in the Euclidean metric, it is difficult to compute the Voronoi cell area in practice. For this reason, an area computation lemma of the Voronoi cell is presented. The lemma proof and an example are also given. Thus, a way is provided for the area calculation in some engineering applications.

Key words: Voronoi cell; object-oriented data structure; candidate contour bisector; main-chain divide-and-conquer algorithm

* Received August 23, 1999; accepted January 5, 2000

Supported by the National Natural Science Foundation of China under Grant No. 69685006