第 4 卷(A 版) 第 4 期 中国图象图形学报 Vol.4(A),No.4
1999 年 4 月 Journal of Image and Graphics Apr.1999

# 封闭图元求并、交、差的方法

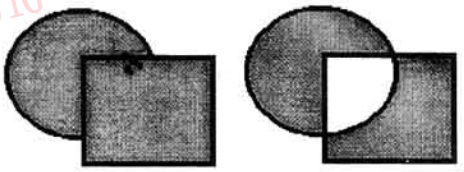
袁灯山 伍 江
(北京大学计算机研究所文字图形处理国家重点实验室,北京 100871)

**摘 要** 简述了问题提出的原因,讨论了求封闭图元的并、交、差的方法,阐述了算法及实现的具体过程。该算法的特点是思路简捷,将这三种运算统一了起来,同时指出了它们的不同点。

**关键词** 封闭图元 内点 外点 交点 线段

## 0 引 言

我们在利用计算机绘制复杂的几何图形时,总是首先生成基本图元,然后将它们组合成复杂的图形。组合的方法有很多,得到的效果也各不一样。大致可能有 3 种方法,(1)简单地将基本图元互相叠在一起,达到所需要的效果;(2)各基本图元的轮廓依然保持,它们的内部区域统一按奇偶规则进行处理;(3)各基本图元的轮廓被打断并重新连接,从而形成新的轮廓和区域。图 1 是这 3 种处理的例子:

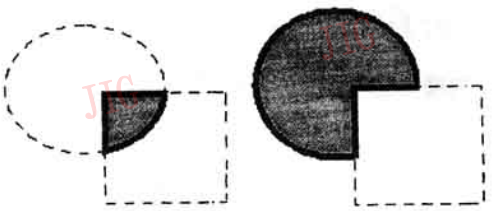


(a)简单地叠在一起的效果 (b)保持轮廓,统一按奇偶规则处理,使中间区域被挖空

(c) 轮廓被连接在一起,形成一个新的闭合连通区域,该区域是原来区域的并

图 1 基本图元组合方法

文中我们要讨论的是第 3 种方法的实现。图 1 显示的还只是一个求并的运算,对图元轮廓能进行的基本操作还可以有以下 2 个:



(a) 新生成的轮廓包围的区域是原来区域的交 (b)新生成的轮廓包围的区域是原来区域的差

图 2 图元轮廓的基本操作

我们将这 3 种操作分别称为求封闭图元的并、交、差。

## 1 名词定义

封闭图元:所谓封闭图元是指这样一条封闭轮廓,该轮廓我们用 $T$ 表示,它可以是一个分段参数函数,也可以是一个单一的函数,但它一定是连续的,且可以取出其中的任意一段。这些封闭图元可以包括椭圆、扇形、矩形、菱形和任意的曲线或直线多边形。

线段:从一个封闭轮廓 $T$ 上取出的一段,它一定是不封闭的。我们用 $T(t_0,t_1)$ 表示一条线段,$t_0$

收稿日期:1998-07-17;收到修改稿日期:1998-09-09

表示线段的起点，$t_1$ 表示线段的末点。

边界点：若点 $p$ 位于封闭轮廓 $T$ 上，称 $p$ 为 $T$ 的边界点。

内点：若 $p$ 不是边界点，从点 $p$ 向任一方向作射线 $L$，$L$ 与封闭轮廓 $T$ 有奇数个交点，则称 $p$ 为 $T$ 的内点。

外点：若 $p$ 不是边界点，从点 $p$ 向任一方向作射线 $L$，$L$ 与封闭轮廓 $T$ 有偶数个交点，则称 $p$ 为 $T$ 的外点。

内部区域：由 $T$ 的所有内点构成的区域，称为 $T$ 的内部区域。

外部区域：由 $T$ 的所有外点构成的区域，称为 $T$ 的外部区域。

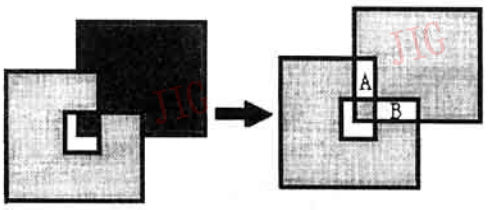
两个封闭轮廓的交点(简称交点)：若点 $p$ 既是轮廓 $T$ 的边界点，又是轮廓 $T'$ 的边界点，则称 $p$ 是 $T$ 和 $T'$ 的交点(若发生重合，则取重合的2个端点为交点，2个端点间的点不作为交点)。

## 2 求并算法

### 2.1 问题描述

设：封闭轮廓 $T$，$T'$

目标：在图形学中，经常使用奇偶法则确定一个或多个轮廓的内部区域，因为奇偶法则在很多时候处理起来都比较容易。对于 $T$ 和 $T'$，对它们分别使用奇偶法则得到的内部区域和对它们统一使用奇偶法则得到的内部区域往往是不一样的。如图3所示：

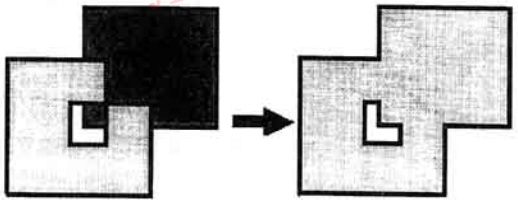



(a) (b)

图3

图3(a)的两个轮廓统一使用奇偶法则构成的内部区域如图3(b)所示，可见，图3(b)的区域 $A$ 和区域 $B$ 被掏空了，这通常不是我们所要的结果，当分别对图3(a)的两个轮廓使用奇偶法则时区域A和B是不应该被掏空的。

本文的目的是希望求出这样一个新的轮廓集 $W$，它内部的所有轮廓统一按照奇偶法则得到的内部区域刚好是 $T$ 和 $T'$ 分别按照奇偶法则得到的内部区域。我们把这种运算称做求 $T$ 和 $T'$ 的并。如图4所示：



(a) (b)

图4

图4(a)的两个轮廓经过求并运算构成图4(b)的两个轮廓。图4(b)的两个轮廓统一按照奇偶法则得到的内部区域刚好是原来分别按照奇偶法则得到的内部区域的并。

### 2.2 算法思想

设 $T$ 与 $T'$ 的交点集合为 $C\{c_0, c_1, c_2, \cdots, c_k\}$，把 $C$ 分别在 $T$ 与 $T'$ 上排序，得到序列 $(c_{i_0}, c_{i_1}, c_{i_2}, \cdots, c_{i_k})$、$(c_{j_0}, c_{j_1}, c_{j_2}, \cdots, c_{j_k})$，$(i_0, i_1, i_2, \cdots, i_k \in \{0, 1, \cdots, k\}$，且互不相等；$j_0, j_1, j_2, \cdots, j_k \in \{0, 1, \cdots, k\}$，且互不相等)。可以用线段 $T_1(c_{i_0}, c_{i_1})$，$T_2(c_{i_1}, c_{i_2})$，$T_3(c_{i_2}, c_{i_3})$，$\cdots$，$T_{k+1}(c_{i_k}, c_{i_0})$ 连接起来表示 $T$，用 $T'_1(c_{j_0}, c_{j_1})$，$T'_2(c_{j_1}, c_{j_2})$，$T'_3(c_{j_2}, c_{j_3})$，$\cdots$，$T'_{k+1}(c_{j_k}, c_{j_0})$ 连接起来表示 $T'$。

设线段集合 $Seg\{T_1(c_{i_0}, c_{i_1})$，$T_2(c_{i_1}, c_{i_2})$，$\cdots$，$T_{k+1}(c_{i_k}, c_{i_0})$，$T'_1(c_{j_0}, c_{j_1})$，$T'_2(c_{j_1}, c_{j_2})$，$\cdots$，$T'_{k+1}(c_{j_k}, c_{j_0})\}$，则 $T$ 和 $T'$ 的并 $W$ 可以表达为一个 $Seg^*$ 集合：$Seg^*$ 是按照一定原则从 $Seg$ 中提取的一个子集，从 $Seg^*$ 中我们再按照一定的原则抽取出若干组线段，每组线段都可连接成一个封闭轮廓，这些轮廓刚好能满足我们上面的要求。

首先讨论得到 $Seg^*$ 的方法：

对于 $Seg$ 中任意线段 $T_n(c_{i_{n-1}}, c_{i_n})$，$(k+1 \geq n > 0)$，若 $T_n(c_{i_{n-1}}, c_{i_n})$ 上存在任意一点 $p(p \neq c_{i_{n-1}}, p \neq c_{i_n})$ 为 $T'$ 的内点，则 $T_n$ 不属于 $Seg^*$。

对于 $Seg$ 中任意线段 $T'_n(c_{j_{n-1}}, c_{j_n})$，$(k+1 \geq n$

$>0)$,若 $T_n'(c_{j_{n-1}}, c_{j_n})$ 上存在任意一点 $p(p \neq c_{j_{n-1}}, p \neq c_{j_n})$为 $T$ 的内点,则 $T_n'$ 不属于 $Seg^*$。(证明从略。)

### 2.3 算法实现

为方便起见,定义函数:

(1) $GetPointIndex(S,t)=m:\{Seg^*,\{0,1\}\}\rightarrow\{0,1,2,\cdots,k\}$。

即求线段 $S$ 的两个端点在 $T$ 与 $T'$ 的交点集合 $C$ 中的索引 $m$,0 表示 $S_p$ 的前端点,1 表示后端点

对于 $S$ 若 $GetPointIndex(S,0)=m$,则 $C$ 中的 $c_m$ 正好是 $S$ 的前端点。若 $GetPointIndex(S,1)=m$,则 $C$ 中的 $c_m$ 正好是 $S$ 的后端点。

(2) $FindNextLine(S,t)=(S',t'):\{Seg^*,\{0,1\}\}\rightarrow\{Seg^*,\{0,1\}\}$。

若 $t$ 为 0 时,在 $Seg^*$ 中寻找 $S'$,$S'$ 与 $S$ 的首点相连,$t'$ 指出与 $S$ 相连的是 $S'$ 的首点还是末点(0 表示首点,1 表示末点);

若 $t$ 为 1 时,在 $Seg^*$ 中寻找 $S'$,$S'$ 与 $S$ 的末点相连,$t'$ 指出与 $S$ 相连的是 $S'$ 的首点还是末点(0 表示首点,1 表示末点);

因此:

若 $GetPointIndex(S',0)=GetPointIndex(S,0)$,

则 $FindNextLine(S,0)=(S',0)$

若 $GetPointIndex(S',0)=GetPointIndex(S,1)$,

则 $FindNextLine(S,1)=(S',0)$

若 $GetPointIndex(S',1)=GetPointIndex(S,0)$,

则 $FindNextLine(S,0)=(S',1)$

若 $GetPointIndex(S',1)=GetPointIndex(S,1)$,

则 $FindNextLine(S,1)=(S',1)$

算法:

(1) 计算出 $T$ 与 $T'$ 所有的交点;

(2) 按要求计算出 $Seg$[1];

(3)通过筛选得到 $Seg^*$;

(4)$k=0$, $tap=1$;

(5)如果 $Seg^*$ 为空则结束;

(6)从 $Seg^*$ 中取出第一条线段赋给 $W_0$,记

$start=GetPointIndex(W_0,0)$;

(7)从 $Seg^*$ 中删除 $W_0$;

(8)求$(W_{k+1},tap')=FindNextLine(W_k,tap)$;

(9)从 $Seg^*$ 中删除 $W_{k+1}$, $k=k+1$; $tap=1-tap'$;

(10)If( $GetPointIndex(W_{k+1},tap)==start$) 则由 $W_0$ 到 $W_{k+1}$产生一个封闭图元,转到(4);

(11)转到(8)。

### 2.4 考虑重合

前面的讨论都是假设 $T$ 和 $T'$ 没有发生边重合的情况,若发生了边重合,要正确地求并,首先要求 $T$ 和 $T'$ 各自的边都没有发生自交。

重合发生时,只对筛选 $Seg^*$ 有影响。

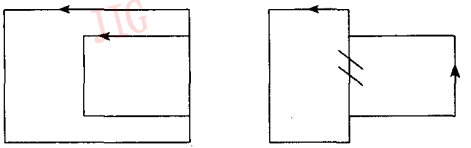
假设无穷远点是 $T$ 和 $T'$ 的外点,根据这个假设来调整 $T$ 和 $T'$ 的方向,使 $T$ 或 $T'$ 的内点始终在 $T$ 或 $T'$ 方向的左侧(当有自交时,这个条件无法满足,只能将自交打断,方法从略)。

首先调整 $T$ 和 $T'$ 的方向使其满足上述条件。

设 $T_p(c_{i_{p-1}}, c_{i_p})$, $T'_q(c_{j_{q-1}}, c_{j_q})$为重合线段,考虑下列情况来对 $T_p$、$T'_q$ 来处理:

若 $c_{i_{p-1}}=c_{j_{q-1}}$, $c_{i_p}=c_{j_q}$ 重合线段方向相同,则在 $Seg^*$ 中保留 $T_p$ 和 $T'_q$ 的任意一条线段,另一条删去(见图 5(a))。

若 $c_{i_{p-1}}=c_{j_q}$, $c_{i_p}=c_{j_{q-1}}$ 重合线段方向相反,$T_p$、$T'_q$ 都从 $Seg^*$ 中删除,见图 5(b)。(证明从略。)



(a) 重合线方向相同 保留一条　　(b) 重合线方向相反 全都删除

图 5

## 3 求交和求差

对于求交、求差,在算法思想上与求并总体上是相同的,只是在筛选 $Seg^*$ 上有所不同。

求交:

对于 $Seg$ 中任意线段 $T_n(c_{i_{n-1}}, c_{i_n})$,$(k+1\geq n>0)$,若 $T_n(c_{i_{n-1}}, c_{i_n})$ 上任意一点 $p(p\neq c_{i_{n-1}}, p\neq c_{i_n})$为 $T'$ 的外点,则 $T_n$ 不属于 $Seg^*$。

对于 $Seg$ 中任意线段 $T'_n(c_{j_{n-1}}, c_{j_n})$,$(k+1\geq n>0)$,若 $T'_n(c_{j_{n-1}}, c_{j_n})$ 上任意一点 $p(p\neq c_{j_{n-1}}, p\neq c_{j_n})$为 $T$ 的外点,则 $T'_n$ 不属于 $Seg^*$。

重合线的方向相同时,保留一条重合线;方向相反时,不保留。

求差:$T-T'$

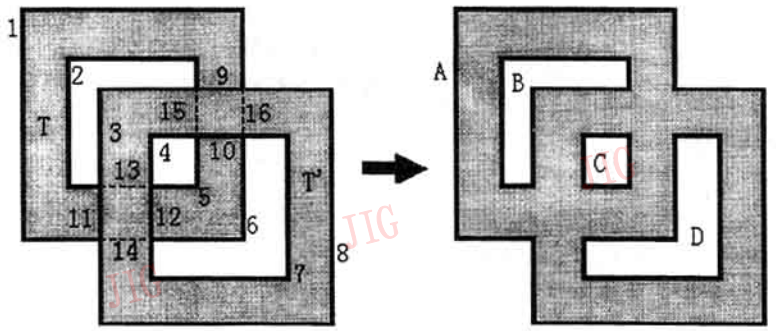
对于 *Seg* 中任意线段 $T_n(c_{i_{n-1}}, c_{i_n})$,$(k+1 \geq n > 0)$,若 $T_n(c_{i_{n-1}}, c_{i_n})$ 上任意一点 $p(p \neq c_{i_{n-1}}, p \neq c_{i_n})$ 为 $T'$ 的内点,则 $T_n$ 不属于 $Seg^*$。

对于 *Seg* 中任意线段 $T'_n(c_{j_{n-1}}, c_{j_n})$,$(k+1 \geq n > 0)$,若 $T'_n(c_{j_{n-1}}, c_{j_n})$ 上任意一点 $p(p \neq c_{j_{n-1}}, p \neq c_{j_n})$ 为 $T$ 的外点,则 $T'_n$ 不属于 $Seg^*$。

重合线的方向相同时,不保留;方向相反时,保留其中一条。

## 4 多轮廓的情况

以上只讨论了用于求交、并、差的两个封闭区域只包含单轮廓的情况,但在绘图中还可能涉及到多轮廓的情况,如图6所示:




(a) (b)

图6

这种情况下,T和T'都是带有2条轮廓的图元,生成的W则有4条轮廓。算法的思路和单轮廓时基本一样。只不过在 *Seg* 中应加入T和T'的所有轮廓两两生成的交线段,其结果如图6(a)所示,共有16条线段;另外,在判断内、外点时要注意,比如,判断一个点 $P$ 是否是 $T$ 的内点时,应对 $T$ 的2条轮廓统一使用奇偶法则,不能分别使用。这样,图6(a)中的9、10、11、12因为包含 $T$ 的内点而被筛掉,13、14、15、16因为包含 $T'$ 的内点而被筛掉。而1和8连成图6(b)的A,2和3连成 $B$,4和5连成 $C$,6和7连成 $D$,由 $A$、$B$、$C$、$D$ 统一按照奇偶法则得到的内部区域刚好是 $T$ 和 $T'$ 分别按照奇偶法则得到的内部区域。

## 5 结束语

经过测试,这种求并、交、差的方法是很有效的,当没有重合线时,对于有自交的、多轮廓的复杂封闭图元都能很好处理。当有重合线时,对于无自交的简单情况也能正确处理。

与文献[1]中的求并算法比较,本文算法的复杂度在于线段的筛选,而文献[1]中算法的复杂度在于(环)合并,(环)遍历。

### 参考文献

1 孙家广,杨长贵. 计算机图形学. 北京:清华大学出版社, 1995

**袁灯山** 1973年11月出生,1996年毕业于北京航空航天大学应用数学专业,现在北京大学计算机研究所攻读计算机应用硕士学位,主要研究方向为计算机文字与图形。

**伍 江** 1973年8月出生,1996年毕业与四川大学计算机系,现在北京大学计算机研究所攻读计算机应用博士学位,主要研究方向为计算机文字与图形。

(下转第306页)

# Topograpy Simulation by Controlling Parameters of Fourinterpolatory Scheme

Luo YanLin
(*Computer Center, Beijing Normal University, Beijing* 100875)
Wang GuoZhao
(*Department of Applied Mathematics, Zhejing University, Hangzhou* 310027)

**Abstract** In this paper, the four-interpolatory scheme is applied in simulating topography. In order to control the form of topography, the proper "control-parameters" are chosen. For the sake of getting variant ones, geometric texture-rendering methods are applied. In the meantime, patches are really displayed in GL environment by modifying the property of the material, the light and the light model. The experimental results show that the algorithm offers an efficient means for simulation of natural scene which effected by the surrounding, the trees, the climate and so on.

**Keywords** Topography simulation, Tour-interpolatory scheme, Shape parameters, Really display

(上接第302页)

# A Method for Union, Intersection and Trim of Closed Contours

Yuan Dengshan, Wu Jiang
(*National Key Laboratory of Text & Graphics Processing Institute of Computer Science & Technology, Peking University, Beijing* 100871)

**Abstract** This paper discusses the method for counting the union, intersection and trim of closed contours. The "reason" for bringing up the question is introduced. The theory and implementation of the algorithm are described detailedly. The algorithm is simple which combines the three counting operations in one. And the differences among the three counting operations are given.

**Keywords** Closed contour, Intern point, External point, Intersect, Point segment