计算几何

Computational Geometry

张思容
zhangsirong@buaa.edu.cn

数学与系统科学学院，北京航空航天大学
Department of Mathematics, Beihang University

September 14, 2011

引言: 我们需要更多的几何学!

- 大学数学的几何课程：解析几何,微分几何，拓扑学：
- 几何分支：欧氏几何，仿射几何，射影几何；
  解析几何，微分几何，代数几何；
- 计算几何：可以计算的几何问题。

Chapter 0: 计算几何引论

What is Computational Geometry?

计算几何是研究用计算机解决几何问题的算法分析。

- 主要内容：几何建模+几何计算。
  几何模型：多边形(多面体)模型(即单纯型), 曲线曲面模型；
  几何计算：求交,搜索,排列；
- 不同分类：
  算术计算几何：1978 Shamos 博士论文；
  数值计算几何：1971 CAGD
- 相关发展：数字几何，离散几何(组合)，离散微分几何;

计算几何的应用

- 计算机图形学：相交，鼠标选取，隐藏面消除；
  算法效率(平面扫描算法)
- 计算机辅助几何设计CAGD,CAM：求交，求并，模拟测试；
  有理样条曲面Nurbs, Coons 曲面等
- 机器人学：Robotics 运动规划(路径寻找)
- 地理信息系统GIS：系统信息查询与组合；

课程内容

预备要求：兴趣
主要目标：
学习常见几何算法， 掌握常见几何模型和概念，深入理解某个算法；
教学参考书:

- (参考教材)计算几何：算法与应用(第三版)。de Berg et al 清华大学出版社.
- (参考教材)计算几何 in C: （英文版)，ORouke. 机械工业出版社。
- (参考教材) 计算几何教程：王仁宏等，科学出版社。
- Graphics Gems I-V: 图形学宝藏： 非常适合课程设计；

课程安排

教学日历：上课16周 (9/6-12/27)

- 第零章：介绍：1周
- 第一章：多边形模型：4周
- 第二章：几何计算：4周
- 第三章：几何模型：曲线与曲面 4周
- 第四章：高级主题 2周
- 课程设计答辩 1周

考核：

- 作业3-4次(包含上机作业)
- 课程设计：1-2人
- 成绩：作业60+课程设计30+课堂参与10=100分
- 要求：提问参与!!!

计算几何的基本2D模型:多边形

Definition (多边形)

由有限个线段，包含一片平面区域的封闭曲线。

- 大多数计算几何的研究对象是多边形；
  同一个几何对象可以用不同多边形逼近；
- 约旦曲线定理：每个简单封闭曲线将平面分成两部分.
  一般研究简单曲线(或多边形)
- 定义：顶点v_i,边$e_i = v_i \to v_{i+1}$;
- 真实感图像的模型：上百万个多边形连接到一起。

艺术画廊问题：Art Gallery Problem

Problem (1973,Klee)

假设艺术画廊的地面是一个有n个顶点的多边形。 问需要安排多少个固定警卫看守整个画廊？或多少灯可以照亮所有地方？

- 可见性：假定每个警卫可以看任何方向(不能穿墙),互相不遮挡。
- 数学描述：设多边形是P,每个警卫是一个点. x可以看到y,指线段$xy \subset P$.
  问题变成是构造点集$S = \{x_i\}$,使得任一点$y \in P$,存在$x_i y \subset P$. 问点集S的大小？
- 极大与极小： 给定多边形P,记$g(P) = \min_S(S$是可以看见整个P的点集$)$,
  给定顶点数n, 记$G(n) = \max_P g(P)$(P是所有顶点数为n的多边形。)
  $1 \le G(n) \le n$. $G(n)$不确定。

艺术画廊定理：Art Gallery theorems

实验考查：

- $1 \le G(n) \le n$
- $G(3) = 1, G(4) = 1$
- $G(5) = 1, G(6) = 2$

Theorem (Art Gallery theorem)

假设艺术画廊的地面是一个有n个顶点的多边形,则 至少需要$floor(n/3)$个固定警卫看守整个画廊.

注记1：也是必要条件：存在梳形多边形，需要$n/3$个警卫。
注记2：可以考查其他的类似定理(比如仅仅覆盖多边形的边界即可).

艺术画廊定理的证明Fisk(1978)

Fisk,1978.

1. 每个多边形可以分解成三角形之并。
2. 可三角化的多边形可以用三色涂色。(使得所有相邻顶点的颜色不同)。
3. 选择同一种颜色的顶点附近安排警卫可以覆盖所有多边形。
4. 存在一种颜色的顶点数少于或等于$n/3$。

□

三角剖分定理:相关证明细节

严格凸指顶点的内角小于π.

Lemma (凸顶点)

每个多边形至少有一个严格凸顶点。

Lemma (对角线)

每个多边形($n \ge 4$)至少有一条对角线。

Theorem (三角剖分定理)

每个n顶点多边形可以用对角线$(n-3)$剖分为$(n-2)$个三角形。

三色定理:相关证明细节

三角剖分的对偶图：每个三角形定义一个结点(node),每两个有相邻(边）的三角形定义一条边连接两个结点。

Lemma (对偶)

每个三角剖分的对偶图是一棵树(顶点度至多为3).

耳朵：多边形的三个相邻顶点bac满足bc是对角线. 称a为耳尖。

Theorem (Meisters双耳定理)

每个多边形($n \geq 4$)至少有两个不重叠的耳朵。

注：$n > 2$的树至少有两个叶子结点。

Theorem (三色定理)

每个三角剖分的对偶图可以用三色着色。

多边形的符号面积计算

Lemma (三角形符号面积)

设三角形T的三个顶点坐标$(a_0, a_1), (b_0, b_1), (c_0, c_1)$,则$2A(T) = (b_0 - a_0)(c_1 - a_1) - (c_0 - a_0)(b_1 - a_1)$.

Lemma (四边形的面积)

设四边形$Q = (a, b, c, d)$,则$2A(Q) = A(a, b, c) + A(a, c, d) = \sum(x_i y_{i+1} - x_{i+1} y_i)$.

Theorem (多边形面积)

设多边形$P = (v_0, v_1, \ldots, v_{n-1})$,$p$是平面上任何一点，则
$A(P) = A(p, v_0, v_1) + A(p, v_1, v_2) + \cdots + A(p, v_{n-1}, v_0)$.
特别坐标表示$2A(P) = \sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1} y_i) = \sum_{i=0}^{n-1}(x_i + x_{i+1})(y_{i+1} - y_i)$.

注记：可以推广到三维体积和高维单纯型的体积。

多边形：数据和结构

- 点的表示：二维数组(不用结构);$tPointi[2]$
- 多边形表示：双链接表：
  struct tVertex{int vnum;
  tPointi v;
  tVertex next,prev; }
  tVertex vertices=NULL;
- 面积函数：$Area2(a, b, c), AreaPoly2()$.

找对角线I：线段相交

给定多边形$P = v_i$, 线段$s = v_i v_j$是对角线的充要条件：

- $s = v_i v_j$和P的所有(不含v_i, v_j)边e不相交
- $s = v_i v_j$在P的内部;

注记：可以直接计算线段求交,但容易有错误!(垂直和平行线？)
利用符号面积判断是否相交(不求交点)

- 点c在有向线段ab的方位：$left(a, b, c) = area2(a, b, c) > 0$
- 线段相交一般情形: 线段cd在线段ab的两侧且反之也是；(要排除三点共线情形);
- 线段相交特殊情形：三点共线且一点在另一个线段中间；

找对角线II：内部或外部对角线

给定多边形$P = v_i$，$s = ab$是内部对角线的判断：

- s位于a_-aa_+和b_-bb_+的扇形里；(凸凹情形)
- *incone*函数：利用*left*函数判断；

通过找对角线直接三角剖分

- 给定$s = v_iv_j$判断是否对角线；判断是否在内部；
- 切割多边形变成两个，递归找到剩下$n-3$个对角线即可；
- 计算量：$O(n^4)$. 对角线$O(n^2)$,相交$O(n)$,递归$O(n)$.
  特别：由双耳定理，仅仅找对角线v_iv_{i+2}即可，得到$O(n^3)$算法

三角剖分实现：切割耳 ear removal

算法：$O(n^2)$

- 初始化判断顶点是否是耳尖(ear tip)；
- WHILE $n > 3$
- 找到一个耳尖v_2
- 输出对角线v_1v_3
- 去掉v_2
- 更新v_1, v_3是否是耳尖；
- END

注记：可以进一步加快速度$O(n)$?

Chapter 1: 多边形模型

1. 多边形剖分
   - 三角剖分
   - 其他剖分
2. 凸包
3. Voronoi 图
4. Delaunay 三角剖分
5. 若干应用

找对角线I：线段相交

给定多边形$P = v_i$，线段$s = v_i v_j$是对角线的充要条件：

- $s = v_i v_j$和P的所有(不含v_i, v_j)边e不相交
- $s = v_i v_j$在P的内部；

注记：可以直接计算线段求交,但容易有错误!(垂直和平行线？)
利用符号面积判断是否相交(不求交点)

- 点c在有向线段ab的方位：$left(a, b, c) = area2(a, b, c) > 0$
- 线段相交一般情形：线段cd在线段ab的两侧且反之也是；(要排除三点共线情形);
- 线段相交特殊情形：三点共线且一点在另一个线段中间；

找对角线II：内部或外部对角线

给定多边形$P = v_i$，$s = ab$是内部对角线的判断：

- s位于$a_- aa_+$和$b_- bb_+$的扇形里；(凸凹情形)
- $incone$函数：利用$left$函数判断；

通过找对角线直接三角剖分：

- 给定$s = v_i v_j$判断是否对角线；判断是否在内部；
- 切割多边形变成两个，递归找到剩下$n - 3$个对角线即可；
- 计算量：$O(n^4)$. 对角线$O(n^2)$,相交$O(n)$,递归$O(n)$.
  特别：由双耳定理，仅仅找对角线$v_i v_{i+2}$即可，得到$O(n^3)$算法

三角剖分实现：切割耳 ear removal

算法：$O(n^2)$

- 初始化判断顶点是否是耳尖(ear tip)；
- WHILE $n > 3$
- 找到一个耳尖v_2
- 输出对角线$v_1 v_3$
- 去掉v_2
- 更新v_1, v_3是否是耳尖；
- END

例子：18点多边形（Figure1.27） 代码 tri.c

多边形的其他剖分

三角化算法的进化：
1911 $O(n^2)$, 1978 $O(n\log n)$, 1986 $O(n\log\log n)$
1991 Chazzelle $O(n)$

- 梯形剖分 trapezoidalization：每个顶点用水平线剖分，适当加对角线。
  实现算法：plane sweep $o(n\log n)$
- 单调分解 monotone
  注记：每个单调块可以在线性时间三角化。
- 凸分解：
  应用：扫描字符识别；

凸包的定义convex hull

凸集：集合内任两点的线段都包含在集合内。

Definition (集合的凸包$\mathcal{H}(S)$)

集合S的凸包由所有集合内所有点的凸组合$(\sum a_i p_i, \sum a_i = 1, a_i > 0)$构成。

等价定义

- n维点集的凸包由集合中$n+1$个点的凸组合构成；
- 凸包是包含集合的所有凸集的交（最小的凸集包含集合S）;
- 凸包是包含集合的所有半空间的交。
  半空间：超平面划分的空间的一半，在二维即半平面。
- 有限点集的凸包是包含点集的最小凸多边形。（面积，周长都最小，证明？）
- 平面点集的凸包是所有点构成的三角形的集合并。我们仅考察二维凸包！！！

应用：机器人行走，形状分析；

平面凸包与极端点extreme points

Remark (凸包的边界)

平面凸包是凸多边形，通常由其边界的顶点*vertex*确定。（组成一个顶点环路*circle*)
边界的顶点属于集合S,其中极端点*extreme points*（严格凸）是凸多边形的角顶点。给出极端点即得到凸包。
其他内部的点和边上的点不是极端点。

简单算法：

- 引理：非极端点当且仅当其在某个三角形的内部（含边界）且不是三角形的顶点。
  存在$O(n^4)$算法找出所有非极端点。
- 极端边：如果集合内每一个点都在某条边的一侧。
  存在$O(n^3)$算法找出所有极端边（但可能包含非极端点);

礼物包裹算法：Gift Wrapping

Chand,Kapur(1970)：类似于以一根绳环绕礼物盒包装的过程。找到一个极端边再逐步找后面的极端边（利用旋转角）。
可以用于任意维的凸包构造，但输出包含非极端点。
算法复杂性：$O(n^2)$, 特别输出决定$O(nh)$,其中h是凸包的边数。

1. 找到最低点（y坐标）(最右边)，初始极端边为水平线；
2. 记为i_0,$i = i_0$
3. DO
4. for all $j \neq i$ 计算和前面极端边的夹角θ
5. 找到最小角对应点k
6. (p_i, p_k)为新的极端边。
7. UNTIL $i = i_0$.

三点共线?
推广到三维?

快速凸包算法：QuickHull

Preparata，Shamos(1985): 类似于QuickSort(Knuth 1973)。先找两个极端点(a, b)（比如最左，最右）；再找一个极端点c；去掉三角形abc中的点，递推的找$(a, c), (c, b)$. 算法：$O(n^2)$

1. function QuickHull（a,b,S)
2. IF $S = \emptyset$ return()
3. ELSE
4. 找到c:离(a, b)最远的极端点；
5. 构造集合A为(a, c)右边的点集；
6. 构造集合B为(c, b)右边的点集；
7. return QUickHull(a,c,A)+(c)+QUickHull(c,b,B)

算法分析：$T(n) = O(n) + T(\alpha) + T(\beta)$,其中$\alpha, \beta$为集合$A, B$的大小；
最好情形$O(n \log n)$,最差$O(n^2)$.

Graham算法

Graham(1972):最差$O(n \log n)$.
基本思想：从两点出发，旋转扫描增加“凸”顶点且去掉凹顶点。
基本算法：

1. 找到一个内部点（p）
2. 所有点围绕p按逆时针顺序排序，记为$p_1, \ldots, p_n$
3. 初始化堆栈$S = (p_2, p_1) = (p_t, p_{t-1}), i = 3$
4. WHILE $i < n$
5. if p_i 在(p_t, p_{t-1})的左边；
6. $push(p_i, S)$: p_i放入堆S
7. else pop(S): 取出堆的顶点。
8. END

可能的问题：内部点？共线点？

Graham算法细节

起点到终点：从一个极端点出发；（从而循环必然中止）
共线点选择：选取最长的点作为极端点；
排序问题：不用$atan()$,斜率，用符号面积(p_0, p_i, p_j)
基本算法：

1. 找到最右边的最低点（p_0）
2. 所有点围绕p_0按逆时针顺序排序，记为$p_1, \ldots, p_n$ （去掉一些共线点）
3. 初始化堆栈$S = (p_1, p_0) = (p_t, p_{t-1}), i = 2$
4. WHILE $i < n$
5. if p_i 在(p_t, p_{t-1})的严格左边；
6. $push(p_i, S)$: p_i放入堆S，$i++$
7. else pop(S): 取出堆的顶点。
8. END

Graham算法的实现和例子: 参见graham.c

```
main()
{
tStack top;
n = ReadPoints();
FindLowest();
PrintPoints();
qsort( &P[1], /* pointer to 1st elem */ n-1, /* number of elems */ sizeof(
tsPoint ), /* size of each elem */ Compare /* -1,0,+1 compare function
*/ );
printf("After sorting, ndelete = PrintPoints();
if (ndelete > 0) { Squash(); printf("After squashing:"); }
top = Graham();
printf("Hull:"); PrintStack( top ); PrintPostscript( top ); }
```

二维凸包的算法分析：最坏情形

能否找到更好的$O(n)$算法?

Theorem

任何计算二维凸包的算法的复杂性最坏情形必然是$\Omega(n\log n)$.

注：$O(g(n)) : f(n) \leq Mg(n)$
$\Omega(g(n)) : f(n) \geq Kg(n)$,
特别$\Theta(g(n))$ 同时存在上下界。
解释:

- 定理：排序算法的下界是$\Omega(n\log n)$.
- 任何一个二维凸包的算法可以得到一个排序算法。$x \to (x, x^2)$ 反之，未知！
- 结论：二维凸包的算法的下界为$\Omega(n\log n)$。

可以推广的$O(n\log n)$二维凸包算法

Graham算法不能用于三维：不能角度排序。
增量算法：incremental algorithm 逐点插入

1. 初始化凸包$H_2 = conv(p_2, p_1, p_0)$
2. FOR $k = 3 : n-1$
3. $H_k = conv(H_{k-1} + p_k)$
4. END

分治算法：Divide and Conquer:

1. 按x坐标排序所有点;
2. 初始化集合A, B各含一半的点;
3. 递归计算$H(A), H(B)$
4. 合并$H(A) + H(B)$.

注：合并$H(A) + H(B)$可以得到$O(n)$线性速度!

从多边形到多面体：三维凸包的相关内容

多面体的边界：多边形曲面，存在拓扑限制。

Theorem

仅仅存在五种正多面体。（*Platonic Solids*）面数为*4*，*6*，*8*，*12*，*20*.

解释：.

设每个顶点有f个面相交，每个面有p个顶点，角为$\pi(p-2)/p$
关键限制：凸多面体在该点的各面角和小于2π.
即$f*(p-2)/p < 2$, 即$(f-2)(p-2) < 4$. □

Theorem

欧拉数：设简单多面体的顶点数，边数，面数为V, E, F,则有$V - E + F = 2$.

注：一般的有g个洞的多面体满足$V - E + F = 2 - 2g$，g成为亏格。证明参见拓扑书。

Voronoi 图:例子和应用

EXAMPLE (邮局问题)

假定某个区域内有若干邮局（*sites*），区域内人们都选择最近的邮局投信，则区域可以划分为 一些子区域。这种划分是基于*sites*的*Voronoi*图。

应用举例

- 传染病学的起源：1850 英国医生Snow 发现 Cholera 病人集中于Broad Street的water pump.
- 社会地理学： 在区域内新开一个商店或服务站的选址问题。
- 气候与地理学： 雨量计算，森林火灾模型，
- 其他：机器人学(自动行走，避免障碍), 材料科学（晶体增长）。

历史：1644 Descartes， 1850 Dirichlet，1854 Snow;
1908 Voronoi：一般n维Voronoi图。
地球物理学： Thiessen泰森多边形。

平面Voronoi 图的定义：Voronoi diagram

欧式距离：平面两点p, q的距离是$dist(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$.

Definition (平面Voronoi 划分)

给定平面点集$P = \{p_1, p_2, \ldots, p_n\}$(称为基点*sites*），可以将平面划分为*Voronoi*区域$V(p_i)$的并。其中 $V(p_i)$是所有最近点为p_i的点的集合。

例子

- 两点情形：
- 三点情形：
- 四点情形：

Proposition

任意两点p, q的闭半空间$H(p,q) = \{x : dist(x,p) \le dist(x,q)\}$,则$V(p_i) = \cap_j H(p_i, p_j)$. $V(P)$是所有子区域的边界的集合（顶点和边）。

Voronoi 图的性质

基本属性：

- $V(p_i)$是开的凸多边形；
- 如果不是所有基点共线，则$V(P)$是连通的，且边是线段或射线。
- $n \ge 3, V(P)$的顶点数不超过$2n - 5$,边数不超过$3n - 6$.
  利用欧拉数定理，构造封闭平面图。

Theorem (判定法则)

定义$C(p)$是以p为圆心内部不包含任何基点的最大园。则有
1； x是顶点当且仅当最大空园$C(x)$上至少有三个基点。
2： p, q的平分线包含一条边当且仅当这条线上存在点x使得$C(x)$的边界只经过p, q两个基点。

Voronoi 图的构造算法

普通算法：

- 简单算法：$o(n^2 \log n)$ 直接求半平面的交(类似求凸包算法);
- 增量算法：1977 incremental algorithm 逐点插入 $O(n^2)$
- 分治算法：1975 Divide and Conquer: $o(n \log n)$

Fortune 算法： 1987， $O(n \log n)$.

- 平面扫描算法： plane sweep
  使用夹角为$\pi/4$的平面扫描。
- 跟踪圆锥交线：抛物线段的出现与消失：beach line
- 找边：
- 找顶点：

Delaunay 三角剖分的定义

给定平面点集$P = \{p_1, p_2, \ldots, p_n\}$(称为基点sites）,得到平面划分Voronoi 图$V(P)$。
定义对偶图G：顶点是基点，两个顶点有一条边如果两个基点的Voronoi多边形有一条公共边。

Proposition

任何平面点集的对偶图是平面图。（*Delaunay* 图）

Definition (Delaunay 三角剖分,1934)

如果没有四个基点共圆，则对偶图G用直线画可以得到一个关于基点集合的三角剖分。

- 两点情形
- 三点情形：
- 四点情形：

Delaunay 图的性质

基本属性：记Delaunay 图为$D(P)$

- 对偶：$D()$的面对应$V(P)$的顶点，$D(p)$的边对应$V(P)$的边，$D(p)$的顶点对应$V(P)$的区域，
- Delaunay 定理：如果没有四个基点共圆，则每个面都是三角形。
- $D(p)$的边界是点集P的凸包。

Theorem (判定法则)

定义$C(x)$是以x为圆心内部不包含任何基点的最大园。则有
1；v是$V(p)$的顶点，则$C(v)$是包含v的三角形的外接圆。特别$C(v)$内没有基点；
2：p,q是$D(P)$的一条边当且仅存在点x使得$C(x)$的边界只经过p,q两个基点。

可以从Voronoi图直接得到Delaunay三角剖分;
也从Delaunay三角剖分可得到Voronoi图(其顶点是三角形外接圆心).

Delaunay 图的构造算法：高维凸包算法

Delaunay 图 与凸包

- 一维Delaunay 图：直线点集P,边为(p_i,p_{i+1}).
  构造二维抛物线上点(x,x^2)，得到凸包，去掉顶边，投射即可。
- 重要观察： 抛物线上任意点$x=a$的切线$z=2ax-a^2$,垂直平行移动切线r^2,交抛物线与$a\pm r$两点(圆盘)。
- 二维情形：构造抛物面点集(x,y,x^2+y^2),计算凸包，去掉顶面，投射到平面得到三角剖分！
  说明：过(a,b)的切平面$z=2ax+2by-(a^2+b^2)$，垂直平行移动切平面r^2,交抛物面为椭圆，投射为圆。 即任何一个最低的凸包三角形对应平面的一个最大空心圆，投射得到Delaunay 三角形。
- 推论： 四点共圆当且仅当四点提升的四面体的体积为零。

算法：

- dt4.c:$O(n^4)$,$O(n^2)$ 对任意三个基点得到三角形，提升到空间，$O(n)$ 判断其是否为最低面(其他点在其上面)；
- dt2.c:$O(n^2)$.$O(n^2)$ 对任意基点，提升到空间，得到凸包，比如Qhull $O(n^2)$ 判断其是否为最低面(法向量判别);投射；

Delaunay 三角剖分的边翻转方法 edge flipping

给定平面点集P,连接各点得到的区域划分是极大的(增加边必然与其他边相交) 则称为极大平面子区域划分。又称三角剖分。

Theorem (Delaunay 三角剖分是角度最优的)

给定点集的任意三角剖分，将三角形的内角排序，则 *Delaunay* 三角剖分的角度排序是"最大"的。*Delaunay* 三角剖分极大化最小角。

- 引理：任意非共线的n点平面点集P,设在P的凸包上点数为k,则P的任意三角剖分都有$2n-k-2$个三角形，$3n-3-k$个边。
- 边翻转：任一内部边是一个四边形的对角线,取四边形另一个对角线可以得到不同的三角剖分。
- 角度排序：所有三角形的内角从小到大排序得到角度向量，可以定义三角剖分的排序。
  注：凸四边形不共圆则可能通过边翻转得到角度更优三角剖分。
- 简单翻转算法：给定三角剖分，不断进行边翻转，算法必然中止。
- 随机增量算法：随机加入新顶点，判断新的三角形是否是Delaunay，不是可以通过边的翻转得到。 $O(n\log n)$

最近邻点问题

Definition (平面点集中的最近邻点)

给定平面点集P,平面任一点a，称$b\in P$是a的最近*(*邻*)*点，如果$|b-a|\le\min|c-a|, c\in P$.

注： 最近邻点不是对称关系。
相关应用：

- 点的区域定位：任一点寻找最近的邮局等。
  算法思想：构造Voronoi图，再定位点所在区域。(参见后面算法)
- 平面点集的最近邻点图Nearest Neighbors Graph：任一点连接其最近邻点得到平面图。
  引理：$NNG\subset D(P)$.
  算法思想：构造Delaunay图，再每点判断最近邻点。

最大空圆问题

Definition (平面点集中的最大空圆)

给定平面点集P,寻找圆心位于凸包$H(P)$内，内部不含其他顶点的最大圆。

- 应用：选择核电站或商场的地址。
- 引理：如果最大空圆的圆心必然是Voronoi图的顶点或Voronoi边和凸包的交点。
- 算法:(Toussaint 1983) 构造凸包和Voronoi图，寻找最大空圆的可能圆心。$O(n\log n)$.

最小支撑树 Minimum Spanning Tree

Definition (最小支撑树MST)

给定平面点集P,存在一个树恰好包含所有点(称为支撑树)。其中树的长度最小,即边的长度和最小，称为MST。 一般是欧几里德长度。

Theorem (最小支撑树包含于Delaunay三角剖分)

$MST \subset D(P)$.

- 应用：比如网络连线方案。
- Kruskai算法(1956): 排序边的长度（小到大),依次加入图中，如果没有生成新环路(是树)，继续加。
  复杂度： $O(E\log E) = O(n^2\log n)$.
- 改进算法:构造Delaunay三角剖分，边数为$O(n)$,其他同上。
  $O(n\log n)$.

旅行推销员问题 Traveling Salesperson Problem

Definition (TSP)

寻找经过平面点集中的每一个点的最短路线。

- 定理：TSP是NP问题。可以构造逼近算法。
- 结论：用双向MST可以逼近TSP. 特别其长度小于TSP最优值的两倍。
- 算法:构造Delaunay三角剖分，得到双向MST,再寻找shortcut(短线)进行微调路线。 $O(n\log n)$.
  Christoifes 方法可以得到上界不超过50%.
  可以得到更好上界但算法可能复杂。

中间轴问题 medial axis

Definition (中间轴medial axis 或骨架skeleton)

给定平面简单形状的边界B,定义内部点x到边界的距离$d(p,B) = \min d_{q\in B}(p,q)$. 特别如果边界上有两点(或多点)满足到$x$是最小距离，则记所有$x$的集合为中间轴。

- Voronoi 图的推广：给定若干线段(多边形的边),可以按最近点给出平面区域的一个划分，特别区域的边界上的点是中间轴。
  参见线段的Voronoi图。
  比喻：中间轴是边界同时点火得到的内部消失点集合。
- 例子：圆的中间轴是圆心，椭圆是一个线段(焦点间).
  简单多边形？ 可能包含抛物线段。(注：其点到一个边的端点最短，到另一个边的中间点最短，可得)。
- 定理：简单多边形(含凹多边形)的中间轴是树。
- Blum（1967）：研究生物形状时引入，一般是非凸，光滑曲线。
- 算法:对多边形存在$O(n\log n)$算法.

Chapter 2: 几何计算

平面线段相交

给定平面点a, b, c, d，求线段$L1 = a \to b$和线段$L2 = c \to d$的交。

- 直接数学方法：直线求交$y_i = k_i x_i + b_i, i = 1, 2$，解方程判断交点位置。
- 参数表示方法：线段$L1 = a + t(b - a)$,线段$L2 = c + t(d - c)$,求解：
  $s = [a_0(d_1 - c_1) + c_0(a_1 - d_1) + d_0(c_1 - a_1)]/D$
  $t = [a_0(c_1 - b_1) + b_0(a_1 - c_1) + c_0(b_1 - a_1)]/D$
  $D = a_0(d_1 - c_1) + b_0(c_1 - d_1) + c_0(a_1 - b_1) + d_0(b_1 - a_1)$
  特别：解$0 \leq s, t \leq 1$.

问题与改进：

- 平行线段： $Area(a, b, c) = 0$共线+判断重合部分；
- 不同的相交：区别输出；
- 乘法溢出： 用浮点。
  计算：正方形对角线交点$D = -4r^2, r = 10^5$溢出。

可以改进计算线段和射线或直线相交。

平面线段集或多边形相交

给定两个多边形$P1, P2$分别有n, m条边，求所有边的交。

- 复杂度：一般情形$\Omega(nm)$；
- 复杂度：凸多边形$O(n + m)$
- 目标：依赖于输出大小的快速算法：输出大小k是相交点的个数。

Theorem (Chazelle, Edelsbrunner 1992)

平面n条线段的交集可以用$O(n \log n + k)$算法实现,算法存储空间$O(n)$*(Balaban 1995)*

注：可以用相关算法计算两个凹多边形的交。

平面线段集求交的扫描算法

平面扫描算法:(Bentley,Ottmann 1979) $O((n + k) \log n)$
主要思想：仅仅求相邻线段的交点。

- 平面水平扫描线从上到下：L包含相交的线段;
  求交事件集Q(包含线段端点)；
- 求交事件集Q从上到下(,从左到右)排序；
- 依次事件求交：交点排序$s_0, s_1, s_2, \ldots$
  事件可能：上端点，下端点，交点；
- 相邻交点对应线段求交
- 增加交点到求交事件集；
- END

线段与三角形相交

给定空间三角形$\Delta = (a, b, c)$，求线段$L = p \to r$和它的交q。

- 直接数学方法：解方程判断交点位置。
- 参数表示方法：线段$L = p + t(r - p)$,设三角形平面 $Ax + By + Cz = D$，法向量$N = (A, B, C)$
- 求线段和平面的交点：$t = \frac{(D-p)\cdot N}{(r-p)\cdot N}$.
  特别：解$0 \le t \le 1$.
- 判断交点是否在三角形内：
  简单方法：计算三角形重心坐标$q = \alpha a + \beta b + \gamma c$,利用系数判断；
  投射方法：投射到平面考察是等价的。
  实现：投射到法向量坐标最大的平面(避免奇异情形)．利用符号面积计算。

应用：光线追踪算法的基础；

平面点定位：多边形

给定点q，简单多边形P，判断在内部或其他。

- 硬件算法：直接得到；
- 凸多边形：简单边检测算法：$O(\log n)$，类似三角形。
- 一般情形：环绕数计算。winding number $O(n)$
  沿着边界走一周： 内部点环绕角度为2π,外部点为0,
  定义：$W = theta/2\pi$。
- 一般情形：射线求交数。$O(n)$
  正则的射线交多边形偶数为外部点，奇数为内部点；
  注意：存在退化情形!!!

空间点定位：多面体

给定点q，简单多面体P，判断在内部或其他。

- 凸多边形：简单面检测算法：$O(\log n)$。
- 一般情形：环绕数计算。winding number $O(n)$
  沿着边界面： 内部点环绕角度为4π,外部点为0,
  定义：$W = theta/4\pi$。
- 一般情形：射线求交数。$O(n)$
  正则的射线交多面体偶数为外部点，奇数为内部点；
  注意：存在退化情形更复杂!!!OPEN PROBLEM。
  随机化算法：随机生成一条射线，如果有退化情形，换一条，直到成功！
  例子：1000000次实验，8121次退化0.8%!

正交区域查找与数据库查询

数据库查询：完全匹配，部分匹配，区间查找。
快速查找：建立合适的数据结构。
转化为数学(几何)问题：数据库是一个d维空间；
完全匹配：点定位；字典排序(二分树）；
部分匹配：子空间查找，kd树；
区域查找：正交区域查找，四叉树，区域树；

Definition (正交区域查找)

设数据库在一个d维空间中，寻找位于空间的某个平行与坐标轴的长方体内所有数据的问题，称为 矩形区域查找或正交区域查找。

一维区域查找

给定一维点集$P = \{P_1, P_2, \ldots, P_n\}$，求位于区间$[a, b]$内的点。
平衡二分查找树：

- 建立二分查找树T,输入a, b
- 找到分叉点V
- 向左输出：每个子树的右边叶子；
  向左输出：每个子树的左边叶子；
- 判断是否输出终点的叶子：

算法复杂度：空间$O(n)$,(构造树$o(n \log n)$),
查询:$O(k + \log n)$,k是输出。

二维区域查找

给定二维点集$P = \{P_1, P_2, \ldots, P_n\}$，求位于矩形$[a0, a1] \times [b0, b1]$内的点。
*kd*树：*k* dimension tree：

- 建立*kd*树,输入矩形
- 找到所有与矩形区域相交的节点V；
- 输出在矩形中的叶子；

算法复杂度：空间$O(n)$,(构造树$o(n \log n)$),
查询:$O(k + \sqrt{n})$,k是输出点。
推广：区域树，空间$O(n \log n)$,(构造树$o(n \log n)$),
查询:$O(k + \log^2 n)$. 改进:$O(k + \log n)$。

排列 arrangements

Definition (线或平面的排列)

给定平面上的一族直线，或空间的一族平面，给出平面或空间的一个划分， 该划分称为一个排列。

- 平面的一个排列由顶点，边，线段，面（或胞腔cell)组成。
  一般边，面看成开集。划分是互不相交的。
- 应用： 计算机图形学：隐藏面消除；
- 应用：最大空凸多边形问题；给定平面点集包含最多点的空凸多边形；
  Erdos OPEN问题：是否充分大的平面点集中必然包含一个空六边形。
- 应用：
  火腿三明治定理（Ham-Sandwich cuts）：任何火腿三明治可以用一个平面切开使得两边有相同的面包，cheese 和火腿。
  二维情形：任何两个平面点集可以被一条直线平分。

排列 arrangements的性质

Theorem (排列的复杂度)

任何平面上n条直线的一个简单排列，
有$V = (n-1)n/2, E = n^2, F = C_2^n + n + 1$。一般的排列的复杂度不超过以上数值。

- 简单排列：任意两条直线必相交，没有三线交与一点。
- 简单排列：欧拉公式。
- 一般排列：扰动得到简单排列，增加复杂度。

注意：没有一条线有太多边。

排列 arrangements的构造

Definition (带状区域zone)

给定一个排列A,任意直线L,则$Z_A(L)$是L和A相交的面（区域）， 特别记$z_A(L)$为相交的区域的总边数。

Theorem (带状区域Zone定理)

给定一个n条直线排列A,任意直线L,则$z_A(L) = O(n)$，特别$z_n \leq 6n$.

递增算法：算法复杂度：空间和时间 $O(n^2)$

- 初始化A_0
- 加入一条直线
- 找到一个交点x, 沿$Z_{A_i}(L)$更新A_{i+1}
- 循环：

推广高维情形：d维情形的面复杂度为$O(n^d)$,带状区域复杂度$O(n^{d-1})$,构造算法的复杂度$O(n^d)$。

对偶映射

Definition (平面的点线对偶)

给定平面上的一条直线可以用两个参数唯一确定，对应参数空间的一个点，指定平面到参数平面的一个一一对应称为一个对偶。 记映射为$D(l)$

- 斜率直线:$y = kx + b \to (k, b)$
  注：垂线无定义！
- 极对偶： $ax + by = 1 \to (a, b)$
- 抛物线切线： $y = 2ax - b \to (a, b)$
  注：抛物线$y = x^2$在a点的切线$y = 2ax - a^2$

定义:$D(l) = p, D(p) = l$.

对偶映射的性质

性质

- 对偶映射是非垂线到平面点的一一对应。
- 任意点$p \in L$ 当且仅当$D(L) \in D(p)$.
- $p = l_1 \cap l_2$ 当且仅当$D(l_1), D(l_2) \in D(p)$.
- 如果p在直线l的上面，则$D(p)$在$D(l)$的下面，对称的有p在l下面，$D(p)$在$D(l)$的上面。

例子：一个线段的对偶$[p, q]$是一个double wedge.

若干应用

Voronoi 图和排列

- 抛物线两条切线的交是一维Voronoi图的顶点。$(x + y)/2$
  抛物面的两个切平面的交投影得到平面Voronoi图的边。
  注：切线的排序对应顶点的距离。
- 高阶Voronoi 图：二阶Voronoi 图指有前两个相同邻近点的区域。
  对应与2阶排列(有一条边在其上的排列)。

火腿三明治定理（Ham-Sandwich cuts）：任何火腿三明治可以用一个平面切开使得两边有相同的面包，cheese 和火腿。
二维情形：任何两个平面点集可以被一条直线平分。
引理：点集的平分线对偶到对偶直线集的中位分割线。

运动规划

机器人：robotics 理解(学习)+执行（行动)
运动规划：在一个环境中机器人从一点到另一点不发生碰撞的运动路线(free path)。
记机器人R,障碍物P,起点s,终点t.

- 基本问题：是否存在一条自由路线？ 构造一条自由路线。找到一条最短自由路线。
  简化:R是一点，圆盘，多边形或线段；
  障碍物:若干多边形；
  运动：任意平移，(不可旋转);
- 工作空间和configure 空间：
  工作空间： 包含若干障碍物的环境。机器人R是多边形，参考点$(0,0)$.
  configure 空间：机器人的定位， $R(x,y)$平移；$R(x,y,\phi)$包含旋转。→ 特殊群! C-空间：包含禁止空间和自由空间；
  运动规划:即在其中寻找一条自由曲线。

注：相切不是碰撞。

点机器人

记机器人R是一点,障碍物P,起点s,终点t，则一般存在唯一一条最短自由路线。

- 直接构造区域的梯形图，可以找到一条自由路线，不一定最短!
- 引理：最短路线必然经过多边形（障碍物)的顶点。
- 可见性图：visibility 图：顶点为节点，
  存在连接v,w的弧，如果v,w可以互相看见(线段VW不和障碍物多边形内部交)
  推论:最短路线是可见性图的一条道路。
- 构造可见性图:类似与找到所有对角线。
  基本算法 $O(n^3)$, 可以用线段的排列改进$O(n^2)$ ，特别有$O(n\log n+E)$.
- 最短路线算法:图论经典算法Dijkstra(1959) $O(n\log n+E)$
  类似与油漆的spread：

圆盘机器人

记机器人R是一个圆盘,障碍物P,起点s,终点t。

- 基本想法:加大多边形障碍，缩小机器人到一点。
- 例子： $P^+=P+R$
- Minkowski 和: $A+B=\{x+y:x\in A,y\in B\}$.
  点和集合的加:$x+B$等价与集合的平移；
  特别:$A+B=\cup_{x\in A}\{x+B\}$
- 最短路线算法:
  加大障碍$P+R$;
  s,t是否在一个分支；
  利用可见性图寻找道路。

多边形机器人

例子:假设机器人是正方形，参考点是左下角。

- Minkowski 加法:$P^+=P-R$
- 定理:如果R移到P^+的内部，则机器人和障碍相交。
  如果R移到P^+的边界，则机器人和障碍相切。
  如果R移到P^+的外部，则机器人和障碍不交。
- Minkowski 和的构造算法： $O(n),)(n\log n),O(n^2\log n)$
  构造边向量的星形图即可得到多边形的卷积(P^+的边界)，再构造P^+.

注：由Minkowski 和容易得到一般的最短路线算法:
一般的有：凸多边形机器人可以在n点障碍中找到最短路线的算法$O(n\log n)$.

计算几何

Computational Geometry

张思容
zhangsirong@buaa.edu.cn

数学与系统科学学院，北京航空航天大学
Department of Mathematics, Beihang University

December 14, 2011

Chapter 3: 数值计算几何

参考书：计算几何教程；
金字塔算法：电子工业出版社：

几何形状的表示

几何形状：
确定形状：（微分几何，代数几何），简单形状：解析几何的二次曲线
随机形状：（随机过程？） 几何形状的计算机表示：

- 几何方法：参数表示(曲线与曲面)
- 分形几何：函数迭代的参数；
- 水平集：隐函数 $f(x,y,z)=C$
- 函数空间逼近: 多项式逼近；

Weierstrass 第一逼近定理

Theorem (Weierstrass定理 1885)

设$f(x)\in C[0,1]$,则任意给定$\epsilon>0$,存在多项式$p(x)$, 满足$\max_{x\in[0,1]}|p(x)-f(x)|<\epsilon$.

1912年Bernstein给出多项式构造证明，存在很多证明和相关逼近定理；

- 定义n次多项式 $P_n(x)=\sum_k f(k/n)C_k^n x^k(1-x)^{n-k}$, 后面为Bernstein多项式$B_n^k(x)$. 待证: $P_n(x)\to f(x)$ 一致收敛!
- 分解求和
  $P_n(x)-f(x)=\sum_{k/n\sim x}(f(k/n)-f(x))B_n^k+\sum_{k/n>x}(f(k/n)-f(x))B_n^k$
  记为$|P_n(x)-f(x)|=A+B$
- 关于A: f一致连续，存在小区间 $|k/n-x|<1/n^{1/4}$，满足$|f(k/n)-f(x)|<\epsilon_n;B_n^k$有界(1)，
- 关于B; $|k/n-x|>1/n^{1/4}$时，$|k-nx|<n^3/2$; 求和$n^3/2\sum_k' B_n^k\le\sum_k(k-nx)^2B_n^k=nx(1-x)\le n/4$,
  引理: $\sum_k(k-nx)^2B_n^k=nx(1-x)$
- $A+B$可得。$A<\epsilon_n$ $B\le M(1/n)^{1/2}$ M是$f(x)$上界。

切比雪夫逼近问题

Problem (切比雪夫最佳问题)

设$f(x) \in C[0,1]$,则任意n次多项式逼近$p(x)$，定义偏差$\Delta(p) = \max_{x \in [0,1]} |p(x) - f(x)|$,则最佳逼近多项式满足$E(p^*) = \inf \Delta(p)$. 也称为极大极小逼近。

- Borel 定理：存在多项式取到下界。 注:n次多项式是一个闭集。
- 切比雪夫定理：$p(x)$是n次最佳逼近多项式的充要条件是$p(x) - f(x)$在$n+2$个点列上交替取正负号。 且最佳多项式是唯一的。
  引理：$n+2$个交错点的最小误差即$\Delta(p)$. (反证法).
- 例子：0次最佳逼近 $p_0(x) = (\max f + \min f)/2$;
  1次最佳逼近,设$f'' > 0$, $p_1(x) = (f(1) - f(0))x + d$,$d$由相关切线决定；
- 切比雪夫多项式：$T_n(x) = \cos(n \arccos x)$。
  验证其为n次多项式，有n个实零点，特别是$f(x) = 0$的最佳逼近。

注：切比雪夫多项式代替泰勒展开，可以得到低阶的精确逼近。

函数空间和多项式逼近

常见函数空间(几何形状):
$P_n \subset P_\infty \subset C^\infty \subset C^k \subset C[0,1] \subset L^2[0,1]$.
一般函数空间是无穷维线性空间。

- Weierstrass 第一逼近定理：P_∞在$C[0,1]$中稠密。
- Weierstrass 第二逼近定理： 三角函数可以一致逼近连续可微周期函数。
- 函数空间的不同收敛：(距离) 一致收敛；几乎处处收敛，平分可积收敛
- 傅立叶分析: 三角函数可以平分可积逼近任意L^2可积函数。
  定义:$L^2 : |f|_2^2 = \int f^2(x)dx$

傅立叶分析是信号处理的基本工具。

一元样条函数空间

多项式逼近的误差：Runge现象，高阶多项式逼近必然存在局部振荡数值不稳定。 应用中选择样条函数-分段连续的多项式函数。

Definition (一元样条函数)

给定一组节点 $-\infty = x_0 < x_1 < x_2 < x_n < \cdots < x_N < x_{N+1} = \infty$是实数的一个划分； 定义其上的分段函数$S(x)$满足 $S(x)$在$[x_j, x_{j+1}]$上是次数不超过n次的多项式； 且$S(x)$在$(-\infty, \infty)$上有$n-1$阶连续导数；
称$S_n(x_1, x_2, \cdots, x_N)$为$n$次样条函数。

Theorem (样条函数空间的基)

$S_n(x_1, x_2, \cdots, x_N) = p_n(x) + \sum_{j=1}^N c_j (x - x_j)_+^n$. 即$S_n(x_1, x_2, \cdots, x_N)$是一个$n+N+1$维空间，基为$1, x, x^2, \cdots, x^n, (x - x_1)_+^n, \cdots, (x - x_N)_+^n$.
其中截断函数$(x - x_j)_+ = x - x_j, x > x_j; 0, x \le x_j$.

证明： 左到右依次叠加，考察节点处光滑性。
注:插值问题： 给定$n+N+1$个值，参数满足一定条件(节点相关),则存

Lagrange 插值

$$
\begin{array}{lllll}
x_1: & y_1 = P_1 & & & \\
 & & P_{12} & & \\
x_2: & y_2 = P_2 & & P_{123} & \\
 & & P_{23} & & P_{1234} \\
x_3: & y_3 = P_3 & & P_{234} & \\
 & & P_{34} & & \\
x_4: & y_4 = P_4 & & &
\end{array}
$$

Problem (一元插值问题)

给定$(x_i, y_i), i = 0, \ldots n$,寻找曲线$f(t)$,满足$f(t_i) = (x_i, y_i)$，称为插值问题。

- 多项式插值定理：给定参数节点t_i,控制点p_i,存在唯一的n次多项式是插值问题的解。 样条函数插值定理： 参见教材。
- 线性插值:$P_{01}(t) = (t_1 - t)P_0 + (t - t_0)P_1$，注： 此处分母$t_1 - t_0$省略。下同。
- 高阶插值：通过线性插值的迭代得到n次插值解。
  $P_{012}(t) = (t_2 - t)P_{01}(t) + (t - t_0)P_{12}(t)$
  $P_{0123}(t) = (t_3 - t)P_{012}(t) + (t - t_0)P_{123}(t)$
- 算法实现：Neville算法：通过从下到上(从左到右)建立三角形.
  Dynamic programming $O(n^2)$，储存$O(n)$.
  Aitken 算法：利用递推关系从上到下得到三角形。不快。

三角形插值与多项式基

- Neville算法：从给定控制点P_i到三角形顶点的路径的乘积的和为n次多项式，
- 插值多项式的表示：$P_{0\cdots n}(t)=\sum_{k=0}^{n}P_kL_k^n$，其中$L_k^n$即Lagrange 多项式基；
- 一般$L_k^n(t|t_0,\ldots,t_n)=\frac{\prod_{j\neq k}(t-t_j)}{\prod_{j\neq k}(t_k-t_j)}$.
- 定理：L_k^n即Lagrange 多项式是一组基!
- 不同的插值方法可以得到不同的多项式基：Hermite 基，牛顿基等；

同样的三角形算法改进可以用于逼近问题: Bezier曲线；

De-castliau 算法

简化Neville算法的各个路径为左右边分别为$1-t,t$，得到De-castliau 算法,用于表示过控制点P_k的Bezier曲线；

- Bernstein多项式：从一个控制点到顶点的路径乘积的和是一个n次多项式；正好是Bernstein多项式基。其中每个中间顶点也是低次的Bernstein多项式。Bezier曲线；$B(t)=\sum B_k^n(t)P_k$
- Bezier曲线的性质：线性不变，凸包，对称性；端点插值；由算法直接得到；
- 升阶：任何低阶曲线可以用高阶曲线表示(Bernstein多项式有相关公式),等价与基变换；
  构造:线性插值$n+2$个新控制点$Q(k)=(kP_{k-1}+(n+1-k)P_k)/(n+1)$;
  注：升阶是割角过程。控制多边形逼近Bezier曲线(即Weierstrass逼近定理);
- 细分:$B(t):t\in[0,1]$变成$B(s):s\in[0,r]$; $B(s):s\in[r,1]$.
  三角形构造: 取$t=r$得到三角形，两边Q_k,R_k分别为新三角形的控制点，得到两个新三角形(即Bezier曲线).
- ***其他： 微分(可由三角形得到),张量曲面(金字塔算法)

De-castliau 算法的开花

De-castliau 算法:中间顶点的标记：可以用三元多项式表示$a^ib^jt^k$,特别顶点为t^n,控制点为$a^{n-k}b^k$; 对不同层的插值引入不同变量u_i,则多项式$B(t)=p(u_1,u_2,\ldots,u_n)$;称为多项式$B(t)$的开花 blossom

Definition (开花)

$p(u_1,u_2,\ldots,u_n)$是多项式$P_n(t)$开花，如果满足p是n重对称的；n重线性的， 且有对角线性质$p(t,\ldots,t)=P(t)$.

- 例子: $1,t^1,t^2,\ldots$
- 定理：设$P(t)$是Bezier曲线，$p(u_1,u_2,\ldots,u_n)$是其开花，
  则$P_k=p(a,\ldots,a,b,\ldots,b)$,其中有$k$个$b$,$n-k$个 a.
- 定理:任意多项式$P_n(t)$,存在唯一的开花函数$p(u_1,u_2,\ldots,u_n)$
- 开花与升阶公式：$p_{n+1}(u_1,u_2,\ldots,u_{n+1})=\frac{1}{n+1}\sum_k p_n(u_1,\ldots,u_{k-1},u_{k+1},u_{n+1};$
- 开花与细分:$B(t):t\in[a,b]$变成$B(s):s\in[a,r]$; $B(s):s\in[r,b]$.
  三角形构造: 开花有$Q_k=p(a,,,r,,,)$; $R_k=p(r,r,...,b,b)$
- ***其他： 微分可用齐次开花得到， 张量Bezier曲面的开花

De-boor算法与开花

推广De-castliau 算法:保证每个三角形的相邻节点仅仅只有一个参数不同，不必取相同的路径；
$p(t_2,t_3,t)=((t_4-t)p(t_1,t_2,t_3)+(t-t_1)p(t_2,t_3,t_4))/(t_4-t_1)$.
用$t_{(k+1)(k+2)\ldots(k+n)}$代替$a^{n-k}b^k$,得到一段B样条曲线的德波尔算法；

- 德波尔算法由$n+1$个开花值得到n次多项式$P(t)$;$t_1,\ldots,t_{2n}$称为节点； 是一个渐进序列，要求$t_k\neq tk+n$.
- 局部B样条基:b_k^n是一个控制点到顶点的所有路径乘积的和；
  有$P(t)=\sum b_k^n(t)p(t_{(k+1)(k+2)\ldots(k+n)})$.
- 渐进节点序列的基：$t_1=\cdots=t_n,t_{n=1}=\cdots=t_{2n}$ 对应Bernstein基；
- ***其他： 不同序列得到不同的基；

B样条曲线

每段B样条曲线需要2n个节点，$n+1$个控制点，
利用充分多的节点把每段B样条曲线连接起来，得到光滑的B样条曲线；

- 增加一个节点和控制点得到一个平移的三角形；
- 两个三角形的合并得到一个光滑的两段B样条曲线，
  相交端点的光滑度$n-1$，重复节点：减低相交端点的光滑度；
- 依次类推可以得到无限长B样条曲线；

B样条曲线的性质

- 分段多项式，局部可控；
- 仿射不变，局部凸包性；
- 节点重复度$=n$,过该点；
  一般不用重复控制点过该点,不光滑!
- B样条基是一元样条空间的一组基。

离散微分几何DDG简介

离散微分几何: 直接从曲面的网格(通常是三角形)定义几何和几何不变量的方法。

- 已知信息：顶点V,边E,和面F
- 简单几何量：点，长度，夹角，面积；面的法向量；
- 法向量和切平面:
  顶点的法向量由面的法向量加权平均：权未知！
- 曲率: 法向量的微分；离散差分；
  平均曲率:$H=\sum\cot\alpha_{ij}(p_i-p_j)$;
  高斯曲率:$G=2\pi-\alpha_{jk}$,由夹角定义；满足Gauss-Bonnet定理。
- 其他: 离散Laplace 算子，外微分，联络等
- 应用： 计算机模拟，几何图像处理，几何建模优化；

EXERCISE ONE

作业与算法作业*11/1*日前交有效，带*****可以不做。

1. (看护画廊墙壁)构造一个多边形并布置若干警卫，使得警卫可以看到多边形的所有边界（墙），但至少有一个内部点不被任何警卫看到。

2. (凸包是最小面积)证明：包含一个点集所有点的最小面积的凸多边形是该点集的凸包。 如果多边形不要求是凸的，给出反例。

3. 证明不存在七条边的多面体。

4. 证明：一个Voronoi 图的每个区域的平均边数不超过6.

5. 举例说明一个非凸多边形的中间轴可以包含曲线段。***这些曲线段有什么特殊吗?

6. 证明：最近邻近图（NNG)包含于$D(P)$.即如果b是到a的最近点，则线段$ab \in D(P)$.

算法作业：分上机和不上机两个选择

上机作业可以两到三人为一个小组共同完成

- OPTION 1(计算验证：可以不上机) 给出正五角星的多边形顶点（可由正五边形顶点计算得到);

  A:验证三角剖分算法：给出算法的每一步和最终结果;

  B:验证凸包GRAHAM算法：给出算法的每一步和最终结果;

  C：给出其Voronoi图和Delaunay三角剖分:

  D: ***验证Fortune算法：给出算法的每一步和最终结果;

- OPTION 2(编程验证)

  A:给出正五角星的多边形顶点，用O'Rouke的代码：tri.c,graham.c,dt2.c给出其三角剖分，凸包，Delaunay三角剖分。

  或者用MATLAB的相关函数（qhull,voronoi)等。

  B: 随机生成100组n个在单位正方形中的点，计算其凸包，验证：正好在凸包上的点的个数的期望值是$O(\log n)$.n的选择可以从100开始。

  可用graham.c，或MATLAB.

EXERCISE TWO

作业与算法作业*12/14*日前交有效，带***可以不做。

1. (射线交)给出射线(包含一个起点a和线上一点b)和线段cd的求交方程和判别法则。 (包含两线平行的判别法则)。
2. (重心坐标)设三角形abc内点p的重心坐标为α, β, γ,连接p到abc得到三个三角形，证明：三个三角形的面积比是重心坐标的比。
3. (区域判别)给定凸四边形，延长四条边得到九个区域，给出平面任一点位于某个区域的判别法则。 （可以用整数编号判别).
4. (四线排列)给出平面四条直线得到的一个区域划分(排列)的所有可能情形。给出顶点数，边数和面数。
5. (点集对偶)给出以下点集的对偶集。

   A: n个共线的点； B：三角形的边；

   C：正n边形的边；D：单位园；
6. (Minkowski和)给出正方形和三角形可能的Minkowski和集，特别给出和可能有的最大边数。

   ***证明：两个n和m条边的凸多边形的Minkowski和的边数不超过$c(n+m)$.c是常数(可以取1).

算法作业：以下题目可以 伪代码和手工计算 或 MATLAB实现

给出正五角星的多边形顶点（可由正五边形顶点计算得到);

1. 给出计算多边形环绕数的算法，计算原点和正五角星外一点的环绕数。
2. 给出平面点集kd树的构造算法。构造正五角星和正五边形的顶点集的kd树。

   ***可以每条边增加若干等平分点，kd树是什么?
3. 给出正五角星的顶点的对偶集，说明正五角星和正五边形的对偶集。
4. ***给出正五角星和一个三角形的Minkowski和。

EXERCISE THREE

作业与算法作业*2012/1/14*日前交电子版或办公室西配楼*501*有效

1. (切比雪夫多项式)证明:$T_n(x) = \cos(n \arccos x)$ 为n次多项式，且有n个实零点。

   给出直到三阶的多项式表达式。

2. (Lagrange 多项式)对$n = 3$,验证$L_k^n(t|t_0, \dots, t_n) = \frac{\prod_{j\neq k}(t-t_j)}{\prod_{j\neq k}(t_k-t_j)}$是3次多项式的一组基； 证明$\sum_{k=0}^{k=n} L_k^n(t) = 1$

3. (升阶公式)证明Bernstein多项式的等式

   $B_k^n = ((n+1-k)B_k^{n+1} + (k+1)B_{k+1}^{n+1})/(n+1)$,

   证明：任何n阶Bezier曲线可以用$n+1$阶Bezier曲线表示,其$n+2$个新控制点满足 $Q(k) = (kP_{k-1} + (n+1-k)P_k)/(n+1)$;

4. (开花)给出三次Bernstein多项式的基的开花多项式(4个)，并用它证明: $\sum_{k=0}^{k=3} B_k^3(t) = 1$.

算法作业：以下题目可手工计算或MATLAB实现

给定正方形四个顶点$P_0 = (0,0), P_1 = (1,0), P_2 = (1,1), P_3 = (0,1)$, 利用不同的三角形算法计算以下多项式曲线;

1. 给出顶点对应节点为$(0,1,2,3)$的Lagrange 插值曲线;

2. 给出以四个顶点为控制点的Bezier曲线,定义区间$[a,b]$满足$a = 0, b = 1$;

3. 给出以四个顶点为控制点，节点序列$(0,1,2,3,4,5)$的B样条曲线段，$2 \leq t \leq 3$;

   验证:节点序列$(0,0,0,1,1,1)$对应的B样条曲线段是上面的Bezier曲线;

4. 给出以四个顶点为控制点，节点序列为$(0,1,2,3,4,5,6,7,8)$的整个B样条曲线;