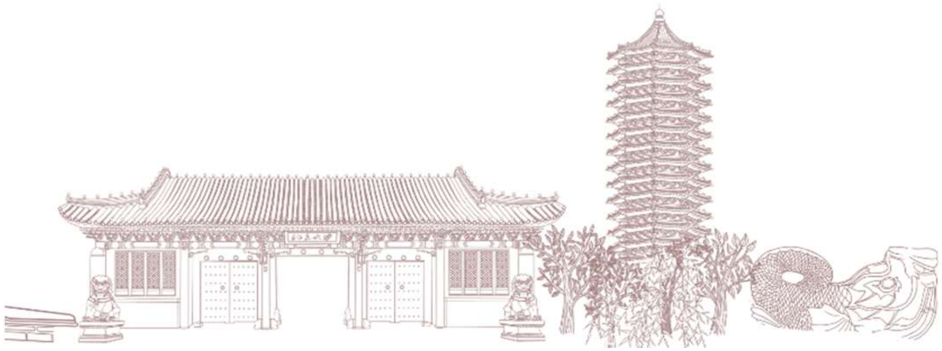
计算几何

Computational Geometry

主讲人：倪星宇

2024年3月18日



什么是计算几何

北京大学 PEKING UNIVERSITY

所属题库 洛谷 主题库 入门与面试 CodeForces SPOJ AtCoder UVA

筛选条件 题目难度 算法/来源/时间/状态 关键词 算法、标题或题目编号 搜索题目内容

已选择 计算几何 ×

共计 115 条结果 清除所有筛选条件

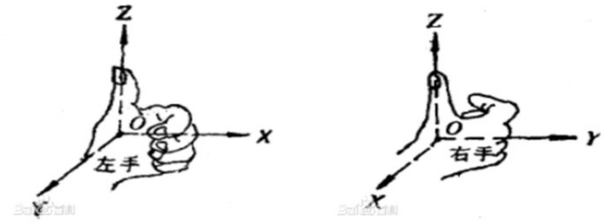
状态	题号	题目名称	显示算法 标签	难度	通过率
—	P1027	[NOIP2001 提高组] Car 的旅行路线	NOIp 提高组 2001	普及+/提高	
—	P1034	[NOIP2002 提高组] 矩形覆盖	NOIp 提高组 2002	普及+/提高	
—	P1142	轰炸		普及/提高−	
—	P1153	点和线		提高+/省选−	
—	P1183	多边形的面积		普及/提高−	
—	P1222	三角形		省选/NOI−	
—	P1257	平面上的最接近点对		普及−	
—	P1325	雷达安装		普及/提高−	

什么是计算几何

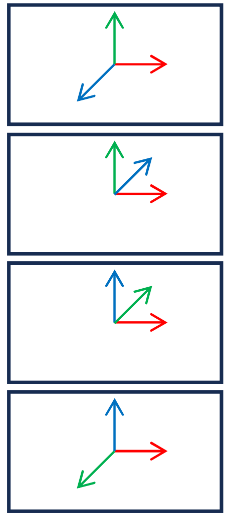
- 计算几何： ✗数学 ✓计算机
  - 起源于 1971 年，计算机图形学 (CG) 与计算机辅助设计 (CAD) 的推动
  - 主要关注解决几何问题的算法
    - 三角剖分、凸包、扫描线、旋转卡壳、半平面交、最小圆覆盖……
  - 特点：需要处理实数而非整数、关心（低维）向量而非标量

- 为什么需要计算几何
  - 降低在计算机中解决几何问题的复杂度
  - 降低在计算机中解决几何问题的数值误差

点、线、面的表示

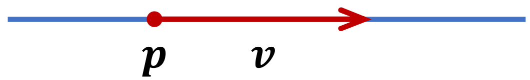


- 坐标系选取
  - 笛卡尔坐标 (Cartesian coordinates) x, y, z
  - 原点选取具有任意性；y/z 轴负方向选为重力方向
    - y 轴为竖直方向、右手系：Houdini、Maya
      - x 正方向指向屏幕右侧；y 正方向指向屏幕上方；z 正方向垂直于屏幕朝外
    - y 轴为竖直方向、左手系：Cinema4D、Unity3D、ZBrush、LightWave3D
      - x 正方向指向屏幕右侧；y 正方向指向屏幕上方；z 正方向垂直于屏幕朝内
    - z 轴为竖直方向、右手系：Blender、3DSMax、CryEngine、SketchUp
      - x 正方向指向屏幕右侧；y 正方向垂直于屏幕朝内；z 正方向指向屏幕上方
    - z 轴为竖直方向、左手系：Unreal
      - x 正方向指向屏幕右侧；y 正方向垂直于屏幕朝外；z 正方向指向屏幕上方
    - OpenGL 为右手系；Direct3D 为左手系

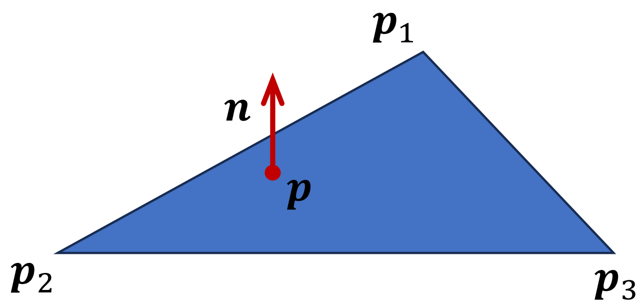


点、线、面的表示

- 点的表示
  - 一般等同于矢量，在单位正交基底下用二维或三维坐标表示
    - 物理上对应于从原点到该点的位置矢量 (position vector)
  - 也可以与矢量进行区分，class Point ≠ class Vector
    - Point − Point → Vector; Point + Point → undefined
    - Point ± Vector → Point; Vector ± Point → undefined
- 线的表示
  - 直线/射线：$\boldsymbol{x}(t) = \boldsymbol{p} + t\boldsymbol{v},\ t \in (-\infty, +\infty)$ or $[0, +\infty)$
    - $\boldsymbol{p}$ 是直线/射线上一点，$\boldsymbol{v}$ 是直线/射线的方向向量
  - 线段：$\boldsymbol{x}(t) = \boldsymbol{p}_1 + t(\boldsymbol{p}_2 - \boldsymbol{p}_1) = (1-t)\boldsymbol{p}_1 + t\boldsymbol{p}_2,\ t \in [0,1]$
    - $\boldsymbol{p}_1, \boldsymbol{p}_2$ 分别是线段的两个端点



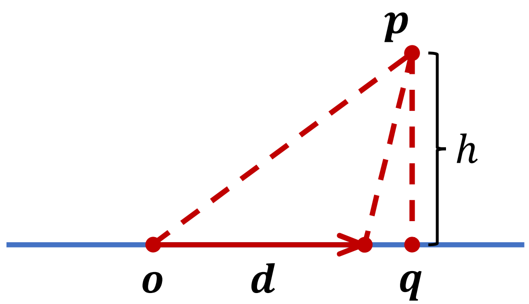
点、线、面的表示



- 面的表示
  - 无穷大平面：$(\boldsymbol{x}-\boldsymbol{p})\cdot\boldsymbol{n}=0$
    - $\boldsymbol{p}$ 为平面上任一点，$\boldsymbol{n}$ 为平面的法向量
  - 半平面：$(\boldsymbol{x}-\boldsymbol{p})\cdot\boldsymbol{n}\le 0$
    - 半“平面”不是平面，而是一半的空间，此时法向量的正负有意义
  - 三角面：$\boldsymbol{x}=\boldsymbol{p}_1+u(\boldsymbol{p}_2-\boldsymbol{p}_1)+v(\boldsymbol{p}_3-\boldsymbol{p}_1)=(1-u-v)\boldsymbol{p}_1+u\boldsymbol{p}_2+v\boldsymbol{p}_3$
    - $\boldsymbol{p}_1,\boldsymbol{p}_2,\boldsymbol{p}_3$ 分别为三角形的三个顶点
    - 若想将 $\boldsymbol{x}$ 限制在三角形内
      - 先在 $\boldsymbol{p}_1,\boldsymbol{p}_2$ 间找一点：$\boldsymbol{x}'=(1-t')\boldsymbol{p}_1+t'\boldsymbol{p}_2$
      - 再在 $\boldsymbol{x}',\boldsymbol{p}_3$ 间找一点：$\boldsymbol{x}=(1-t)\boldsymbol{x}'+t\boldsymbol{p}_3=(1-t)(1-t')\boldsymbol{p}_1+(1-t)t'\boldsymbol{p}_2+t\boldsymbol{p}_3$
      - 令 $u\coloneqq(1-t)(1-t')=tt'-t-t'+1, v\coloneqq t'-tt', w\coloneqq t$
      - $\boldsymbol{x}=u\boldsymbol{p}_1+v\boldsymbol{p}_2+w\boldsymbol{p}_3,\ u+v+w=1$

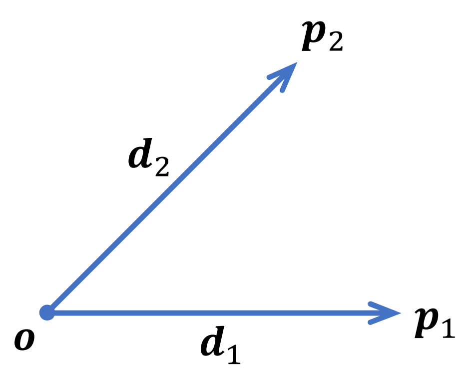
点与线的距离

- 点与直线的距离
  - 点 $\boldsymbol{p}$; 直线 $\boldsymbol{x} = \boldsymbol{o} + t\boldsymbol{d}$
  - $\mathrm{Dist} = h = \frac{\|(\boldsymbol{p}-\boldsymbol{o})\times\boldsymbol{d}\|}{\|\boldsymbol{d}\|}$
- 点在直线上的投影
  - 点 $\boldsymbol{p}$; 直线 $\boldsymbol{x} = \boldsymbol{o} + t\boldsymbol{d}$
  - $(\boldsymbol{o} + t\boldsymbol{d} - \boldsymbol{p}) \cdot \boldsymbol{d} = 0$
  - $\boldsymbol{o} \cdot \boldsymbol{d} + t\|\boldsymbol{d}\|^2 - \boldsymbol{p} \cdot \boldsymbol{d} = 0$
  - $t = \frac{(\boldsymbol{p}-\boldsymbol{o})\cdot\boldsymbol{d}}{\|\boldsymbol{d}\|^2}$
  - $\boldsymbol{q} = \boldsymbol{o} + t\boldsymbol{d} = \boldsymbol{o} + \frac{(\boldsymbol{p}-\boldsymbol{o})\cdot\boldsymbol{d}}{\|\boldsymbol{d}\|^2}\boldsymbol{d}$
- 点与射线的距离
  - 点 $\boldsymbol{p}$; 直线 $\boldsymbol{x} = \boldsymbol{o} + t\boldsymbol{d}$
  - $t = \frac{(\boldsymbol{p}-\boldsymbol{o})\cdot\boldsymbol{d}}{\|\boldsymbol{d}\|^2}$
    - $t \geq 0, \mathrm{Dist} = h$
    - $t < 0, \mathrm{Dist} = \|\boldsymbol{p} - \boldsymbol{o}\|$
  - 点与线段的距离同理



线与线的位置关系 (2D)

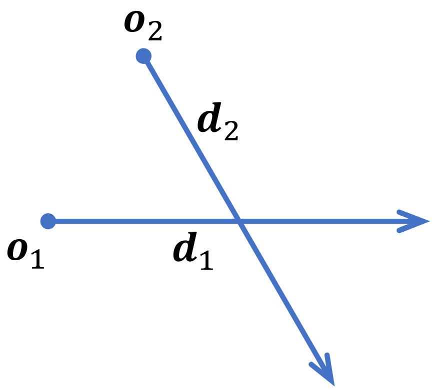
- 共端点射线/线段的绕序
  - 射线 1、2 的端点为 $\boldsymbol{o}$，方向向量分别为 $\boldsymbol{d}_1, \boldsymbol{d}_2$
  - 线段 1 的端点为 $\boldsymbol{o}, \boldsymbol{p}_1$；线段 2 的端点为 $\boldsymbol{o}, \boldsymbol{p}_2$
    - $\boldsymbol{d}_1 = \boldsymbol{p}_1 - \boldsymbol{o}, \boldsymbol{d}_2 = \boldsymbol{p}_2 - \boldsymbol{o}$
  - $\boldsymbol{d}_1 \times \boldsymbol{d}_2$
    - > 0, 射线 2 在射线 1 的"左"边（逆时针转角小于 180°）
    - < 0, 射线 2 在射线 1 的"右"边（顺时针转角小于 180°）
    - $= 0$, 射线 2 与射线 1 重合或为反方向
      - 通过点乘进一步判断



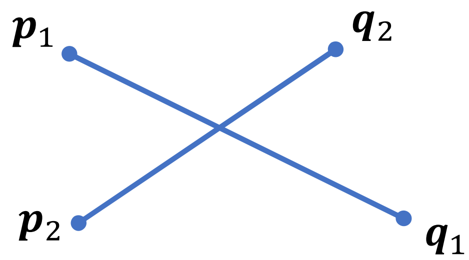
线与线的位置关系 (2D)

PEKING UNIVERSITY

- 直线与直线的交点
  - 直线 1： $\boldsymbol{x}_1 = \boldsymbol{o}_1 + t_1\boldsymbol{d}_1$； 直线 2： $\boldsymbol{x}_2 = \boldsymbol{o}_2 + t_2\boldsymbol{d}_2$
  - $[(\boldsymbol{o}_2 + t_2\boldsymbol{d}_2) - \boldsymbol{o}_1] \times \boldsymbol{d}_1 = 0$
    - $(\boldsymbol{o}_2 - \boldsymbol{o}_1) \times \boldsymbol{d}_1 + t_2(\boldsymbol{d}_2 \times \boldsymbol{d}_1) = 0$
    - $t_2 = \frac{\boldsymbol{d}_1 \times (\boldsymbol{o}_2 - \boldsymbol{o}_1)}{\boldsymbol{d}_2 \times \boldsymbol{d}_1}, \boldsymbol{p} = \boldsymbol{o}_2 + t_2\boldsymbol{d}_2$

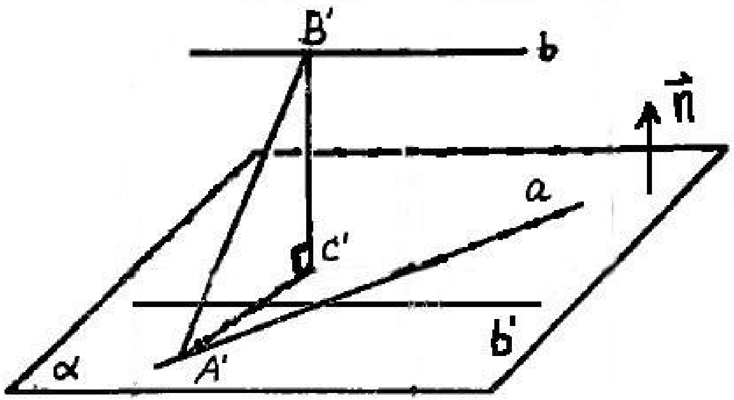


- 线段的跨立测试
  - 线段 1： 端点 $\boldsymbol{p}_1, \boldsymbol{q}_1$； 线段 2： 端点 $\boldsymbol{p}_2, \boldsymbol{q}_2$
  - $s_1 = [(\boldsymbol{q}_1 - \boldsymbol{p}_1) \times (\boldsymbol{p}_2 - \boldsymbol{p}_1)] \times [(\boldsymbol{q}_1 - \boldsymbol{p}_1) \times (\boldsymbol{q}_2 - \boldsymbol{p}_1)]$
  - $s_2 = [(\boldsymbol{q}_2 - \boldsymbol{p}_2) \times (\boldsymbol{p}_1 - \boldsymbol{p}_2)] \times [(\boldsymbol{q}_2 - \boldsymbol{p}_2) \times (\boldsymbol{q}_1 - \boldsymbol{p}_2)]$
  - 两线段相交当且仅当 $s_1, s_2 \leq 0$
    - 取等号时交点在端点处



线与线的位置关系 (3D)

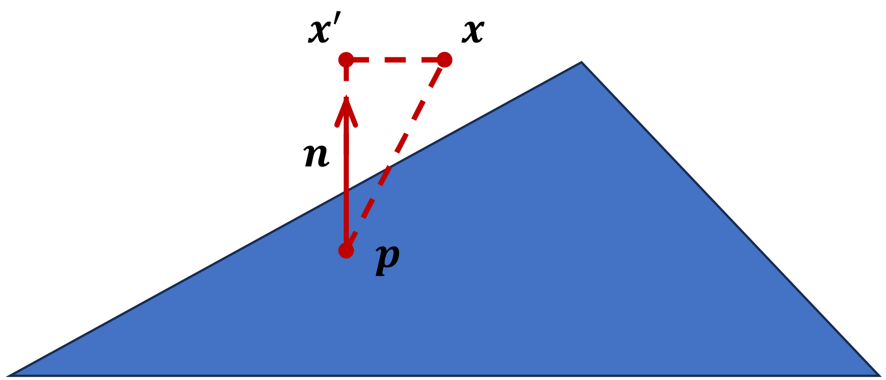
- 异面直线之间的距离
  - $\boldsymbol{x}_1 = \boldsymbol{o}_1 + t_1\boldsymbol{d}_1$
  - $\boldsymbol{x}_2 = \boldsymbol{o}_2 + t_2\boldsymbol{d}_2$
  - 法 1：求两直线公垂线并投影
    - $\boldsymbol{n} = \frac{\boldsymbol{d}_1 \times \boldsymbol{d}_2}{\|\boldsymbol{d}_1 \times \boldsymbol{d}_2\|}$
    - $\mathrm{Dist} = \boldsymbol{n} \cdot (\boldsymbol{o}_2 - \boldsymbol{o}_1)$
  - 法 2：联立方程求极值
    - $\|\boldsymbol{x}_2 - \boldsymbol{x}_1\| = \|\boldsymbol{o}_2 + t_2\boldsymbol{d}_2 - \boldsymbol{o}_1 - t_1\boldsymbol{d}_1\|$
    - 上式对 t_1, t_2 分别取偏导为零



- 异面线段之间的距离
  - 投影至同平面进行跨立测试；优化问题求极值

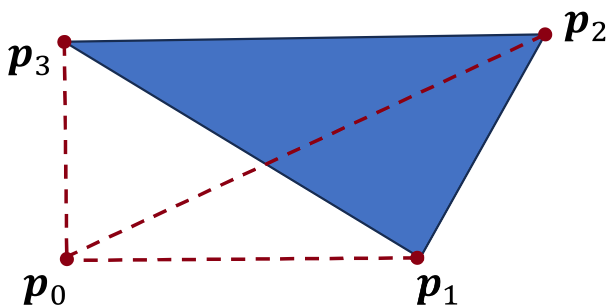
点与面的距离

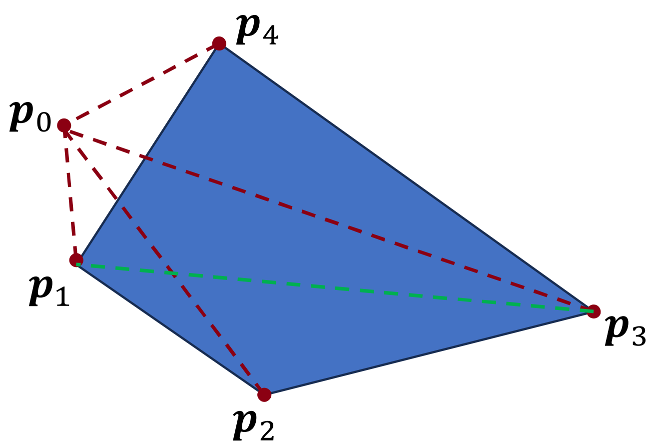
- 点与面
  - 平面：$(\boldsymbol{x}-\boldsymbol{p})\cdot\boldsymbol{n}=0$
    - $\boldsymbol{p}$ 为平面上任一点，$\boldsymbol{n}$ 为平面的法向量
  - 平面外一点 $\boldsymbol{x}$
- 点与面的距离
  - $\text{Dist}=\|(\boldsymbol{x}-\boldsymbol{p})\cdot\boldsymbol{n}\|/\|\boldsymbol{n}\|$
    - 该距离不受 $\boldsymbol{p}$ 点选取的影响（为什么?）
  - 向量形式与解析形式的关系
    - $(x-p_x, y-p_y, z-p_z)\cdot(n_x, n_y, n_z)=0 \Longrightarrow Ax+By+Cz+D=0$
    - $\text{Dist}=\frac{\|(\boldsymbol{x}-\boldsymbol{p})\cdot\boldsymbol{n}\|}{\|\boldsymbol{n}\|}=\frac{\|Ax+By+Cz+D\|}{\sqrt{A^2+B^2+C^2}}$
    - 性质：参数表示与隐式表示；自动适用于二维



多边形的周长与面积 (2D)

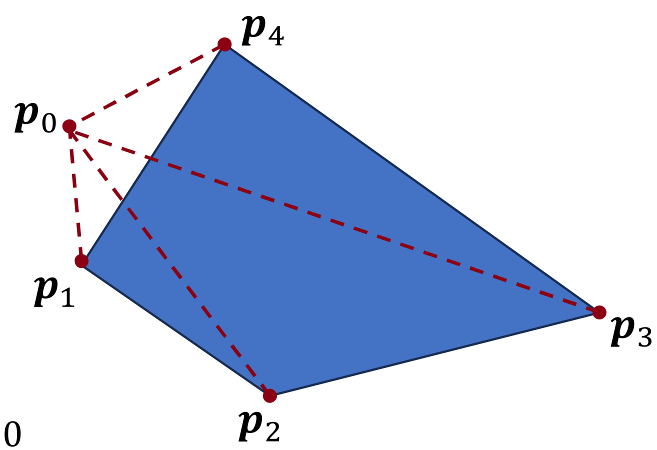
- 多边形的周长
  - 遍历每条边计算模长求和
- 多边形的面积（逆时针序给出多边形）
  - 三角形的面积
    - $A_{123} = \frac{1}{2}\|(p_2 - p_1) \times (p_3 - p_1)\|$
    - $A_{123} = A_{012} + A_{023} - A_{013} = A_{012} + A_{023} + A_{031}$
  - 凸多边形的面积
    - $A_{1234} = A_{123} + A_{134}$
  - 任意多边形的面积
    - $A_{1234} = A_{012} + A_{023} + A_{034} + A_{041}$
    - $A = \sum_{i=1, j=i\%n+1}^{n} A_{0ij}$





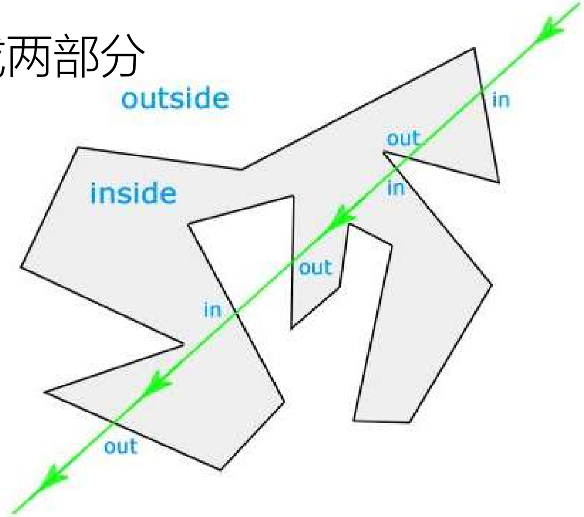
点与多边形的位置关系 (2D)

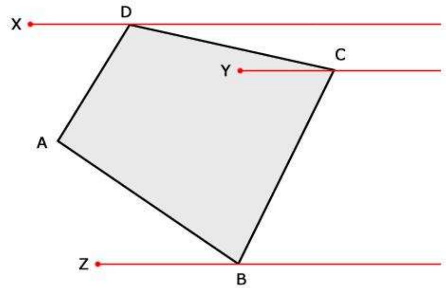
- 点与凸多边形的位置关系
  - 面积法（逆时针序给出多边形）
    - $A = \sum_{i=0,j=(i+1)\%n}^{n-1} A_{0ij}$
    - $A' = \sum_{i=0,j=(i+1)\%n}^{n-1} |A_{0ij}|$
    - 检测 A 与 A' 的大小关系（相等则 $\boldsymbol{p}_0$ 在内）
  - 绕序法（逆时针序给出多边形）
    - 检测 $\boldsymbol{p}_0$ 是否在每一条边的“左边”：$(\boldsymbol{p}_i - \boldsymbol{p}_0) \times (\boldsymbol{p}_{i+1} - \boldsymbol{p}_0) > 0$
  - 思考：多面体的面积、点与凸多面体的位置关系 (3D)
- 点与任意多边形的位置关系
  - 光线投射 (ray casting) 算法
  - 回转数 (winding number) 算法



点与多边形的位置关系 (2D)

- 光线投射算法
  - 若尔当 (Jordan) 曲线定理：任意一条简单闭曲线可以将平面分成两部分
    - 连接同一部分任意两点的弧与该曲线不相交或相交偶数次
    - 连接不同部分的两点的弧与该曲线相交奇数次
  - 观察：取多边形为简单闭曲线，取从判断点出发的射线为弧
    - 弧每与多边形相交一次即改变内/外关系，无穷远点一定在多边形外
    - 相交点个数为奇数：点在多边形内
    - 相交点个数为偶数：点在多边形外
  - 思考：该射线与顶点/边重合
    - 取射线的极角为 π 的无理数倍，降低重合概率
    - 规定射线及射线以上的点在“上方”，否则在“下方”
    - 只有一条边的两个点分别在上方和下方才算相交

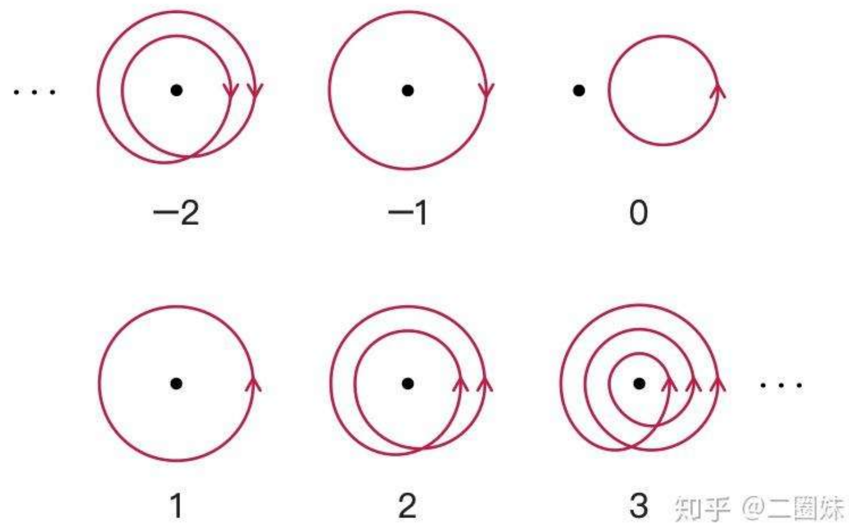




点与多边形的位置关系 (2D)

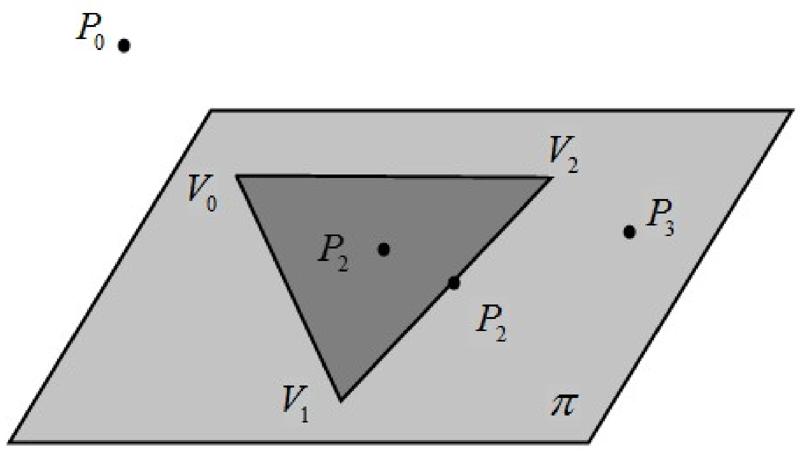
北京大学 PEKING UNIVERSITY

- 回转数算法
  - 回转数 (winding number)：曲线绕过一点的次数
    - 对于多边形内的点，多边形绕过该点的次数为 1
    - 对于多边形外的点，多边形绕过该点的次数为 0
  - 如何计算回转数（逆时针给出多边形）
    - 回转数等于回转角 θ 除以 2π
    - $\theta = \sum_{i=1, j=i\%n+1}^{n} \arcsin \frac{\|(\boldsymbol{p}_i - \boldsymbol{p}_0) \times (\boldsymbol{p}_j - \boldsymbol{p}_0)\|}{\|\boldsymbol{p}_i - \boldsymbol{p}_0\| \|\boldsymbol{p}_j - \boldsymbol{p}_0\|}$



点到三角形的距离

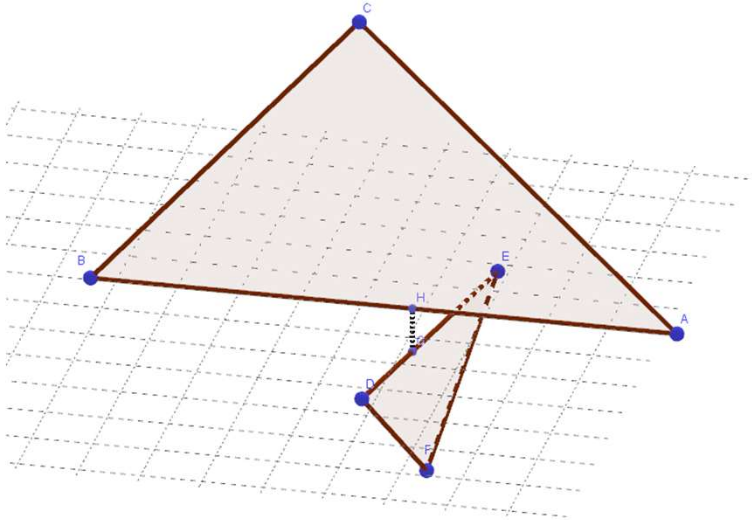
- 二维情形
  - 点在三角形内：距离为零
  - 点在三角形外
    - 设 Dist 为点–顶点距离的最小值
    - 将点分别投影到三条边上
      - 若投影点不在边的延长线上，则用点–边距离更新 Dist
- 三维情形
  - 将点投影到三角形所在平面
    - 记投影距离为 D_1、投影点到三角形的距离为 D_2
    - $D = \sqrt{D_1^2 + D_2^2}$
  - 其它方法：利用三角形的参数表达转换为优化问题



三角形之间的距离

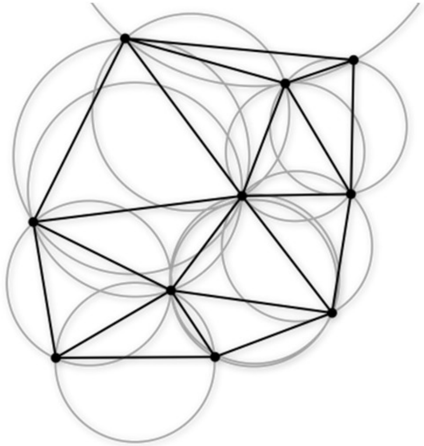
北京大學 PEKING UNIVERSITY

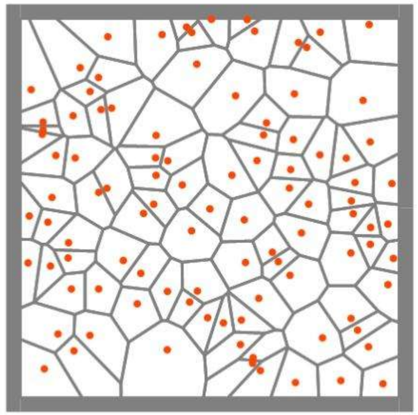
- 二维情形（保证不相交）
  - 6 组点—面/18 组点—边距离的最小值
- 三维情形（保证不相交）
  - 6 组点—面距离的最小值
  - 9 组边—边距离的最小值
- 三角形距离与碰撞检测
  - 离散碰撞检测 (discrete collision detection, DCD)
    - 求两个三角形网格 (triangular mesh) 之间的距离
  - 连续碰撞检测 (continuous collision detection, CCD)
    - 三角形网格的每个顶点都有速度，求两个网格经过多长时间发生接触
      - 先考察"点—面"和"边—边"何时共面（均为一元三次方程组）再判断共面时是否接触



德洛内 (Delaunay) 三角剖分 (2D)

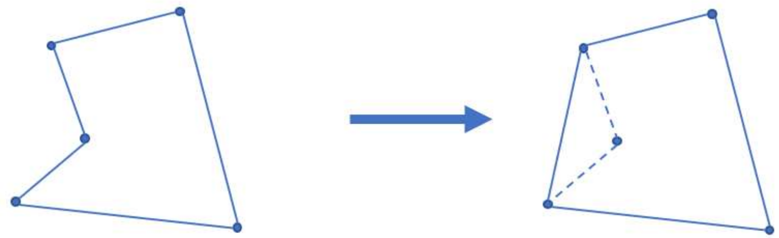
- 三角剖分
  - 将点集连接为三角网格，使得各三角形的边不相交
- 德洛内三角剖分
  - 空圆性：任意三角形的外接圆内无其它点
  - 规则化：最大化最小角
  - 其它性质
    - 所得结果唯一
    - 增删改顶点只影响局部
    - 最外层边界形成凸多边形
  - 沃罗诺伊 (Voronoi) 图：三角剖分的对偶图
    - 将平面划分为若干由最近控制点决定的区域

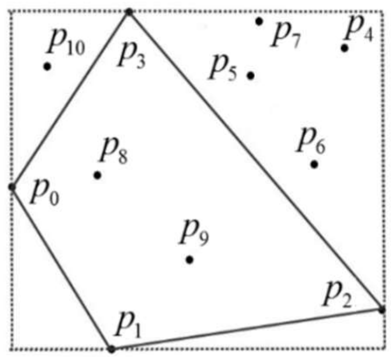




凸包 (2D)

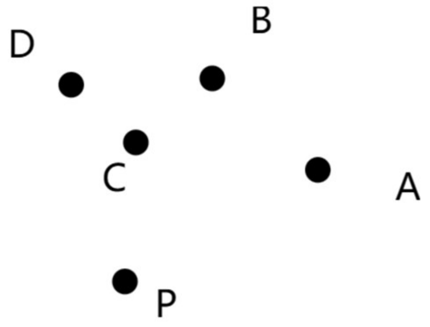
- 凸包的定义
  - 在平面上能包含给定点的最小的凸多边形
  - 凸多边形：所有内角均小于 180° 的多边形
- 凸包与包围盒
  - 轴对齐包围盒 (axis aligned bounding box, AABB)
    - 包含给定点的与坐标轴平行的矩形
    - 凸包一定在 AABB 内；AABB 是凸包最粗糙的近似
  - 方向包围盒 (oriented bounding box, OBB)
    - 包含给定点的最小的矩形
  - 固定方向凸包 (fixed directions hulls, FDH)
    - “凸包”各边（面）的法向只能在给定集合中选取

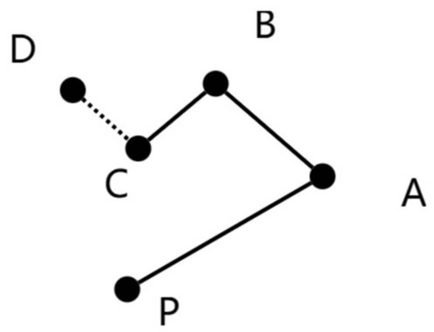


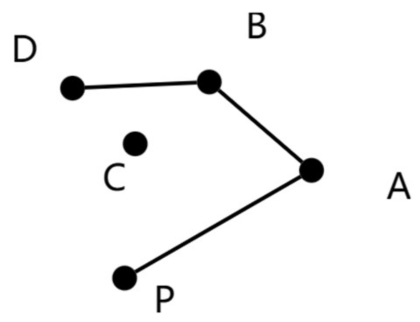


凸包 (2D)

- 葛立恒 (Graham) 扫描法
  - 观察 1：点集中纵坐标最小的点一定在凸包内
  - 观察 2：在凸包上逆时针走，所经过的边一定是“左拐”的
  - 算法流程
    - 取点集中纵坐标最小的点 $\boldsymbol{p}_0$
    - 将 $\boldsymbol{p}_0$ 作为原点，对其余所有点按极角排序
    - 维护一个栈，将 $\boldsymbol{p}_0, \boldsymbol{p}_1$ 压入栈中
    - 按极角序依次考察 $\boldsymbol{p}_2, \boldsymbol{p}_3, \ldots$
      - 设当前考察 $\boldsymbol{p}_i$
      - 记栈顶元素为 $\boldsymbol{p}_\mathrm{f}$，栈顶下方的元素为 $\boldsymbol{p}_\mathrm{s}$
      - 若 $(\boldsymbol{p}_\mathrm{f} - \boldsymbol{p}_\mathrm{s}) \times (\boldsymbol{p}_i - \boldsymbol{p}_\mathrm{f}) > 0$ 则将 $\boldsymbol{p}_i$ 压入栈中
      - 否则弹出栈顶并重复上述过程
  - 时间复杂度 $O(n \log n)$

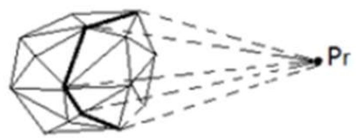






凸包 (3D)

- 增量法
  - 给定 $n-1$ 个点的凸包，考察新点的加入
    - 若新点在凸包内（如何判断?）则凸包不变
    - 若新点不在凸包内，从该点为光源发射光线
      - 点光源会在凸包上形成明暗分界线（凸包的棱组成）
      - 删除亮面，并连接点光源和分界线将形成新的凸包
  - 时间复杂度 $O(n^2)$
  - 如何处理多点共面的情况?
    - 对输入点进行随机扰动
- 三维凸包的其它算法
  - 暴力法：枚举有向三角形，判断它在不在凸包上
  - 快速凸包 (quick hull) 法：增量法的变种，每次选择最远的点



CH(Pr-1)相对于Pr的水平面

图1

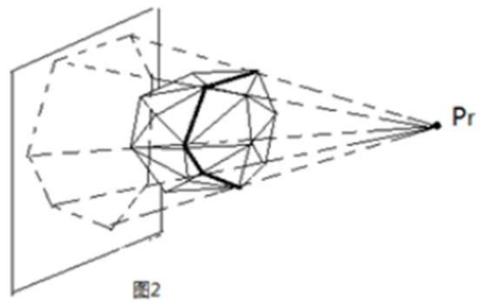
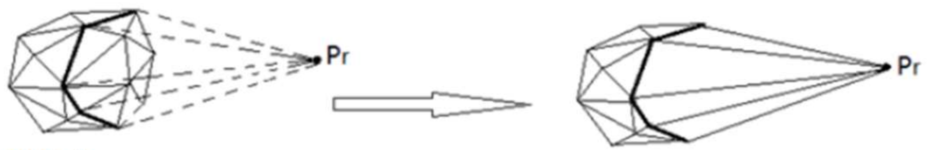


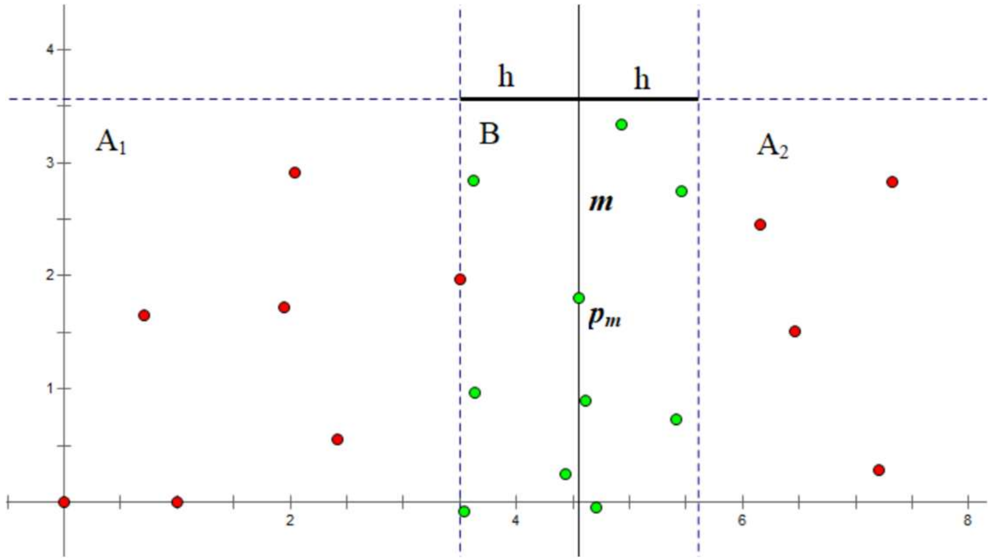
图2



CH(Pr-1)　　　CH(Pr)

平面最近点对 (2D)

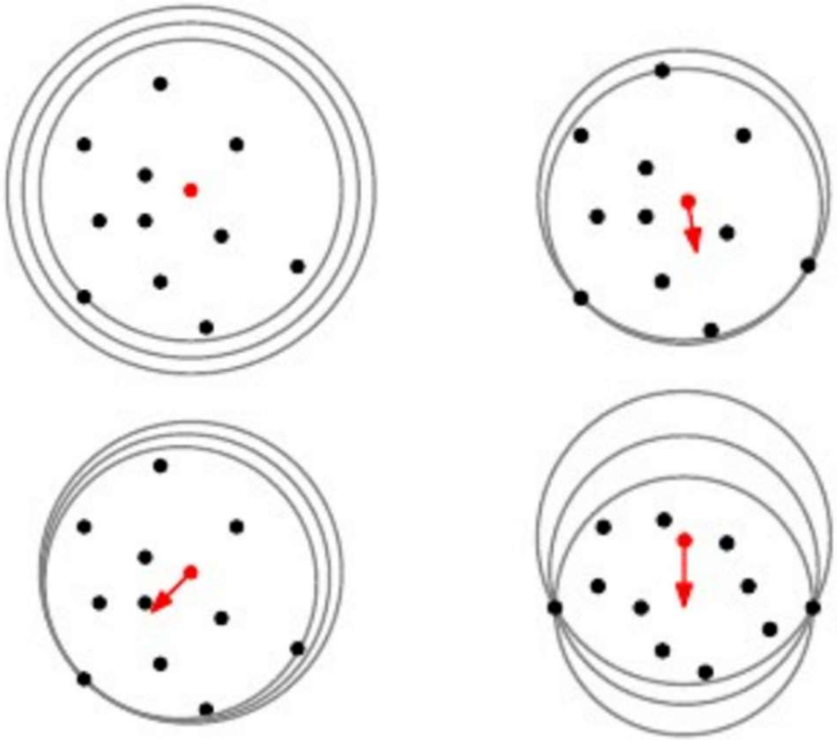
- 分治算法
  - 将点集 S 沿中线划分为两个子集 S_1, S_2
  - S 的最近点对的可能来源
    - S_1 的最近点对
    - S_2 的最近点对
    - 横跨 S_1 和 S_2 的最近点对
  - 如何求解横跨两点集的最近点对
    - 设 S_1 的最近点对距离为 h_1
    - 设 S_2 的最近点对距离为 h_2
    - 只需考虑距中线距离小于 $h = \min(h_1, h_2)$ 的点，构成集合 B
    - 在集合 B 内按 y 序扫描



最小圆覆盖 (2D)

北京大学 PEKING UNIVERSITY

- 最小圆覆盖：包围盒→包围球 (2D)
  - 包围给定点集的**唯一的**最小的圆
- 最小覆盖圆的性质
  - 凸包一定在最小覆盖圆内
  - 最小覆盖圆的三种情况
    - 圆心在某一点处，半径为零
    - 圆心在两点中点，直径为两点距离
    - 圆心为三点外心，半径为外接圆半径
- 最小圆覆盖的增量构造（与凸包一致）
  - 给定 $n-1$ 个点的最小覆盖圆
  - 第 n 个点或在上述圆内，或新覆盖圆必过该点



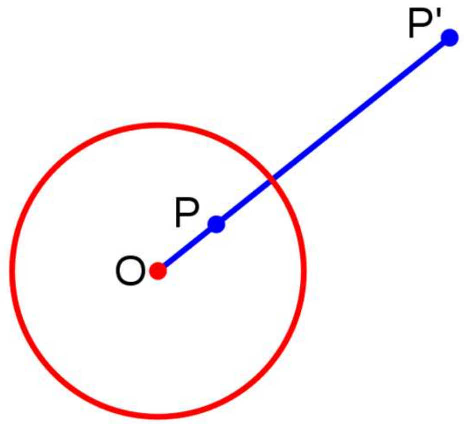
最小圆覆盖 (2D)

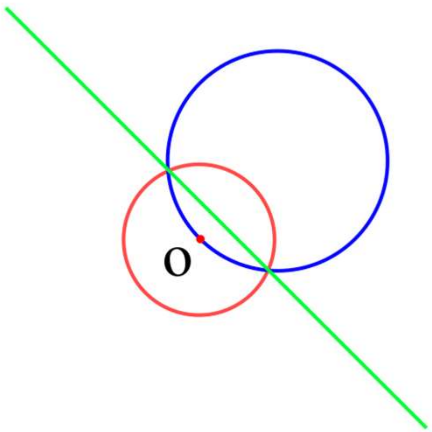
- 随机增量法
  - 算法流程（三层循环）
    - $\odot_{\text{ans}} = \text{Circle}(\boldsymbol{p}_1)$
    - For $i = 2 \rightarrow n$
      - 若 $\boldsymbol{p}_i \in \odot_{\text{ans}}$ 则直接进入下一次循环
      - $\odot_{\text{ans}} = \text{Circle}(\boldsymbol{p}_i)$
      - For $j = 1 \rightarrow i - 1$
        - 若 $\boldsymbol{p}_j \in \odot_{\text{ans}}$ 则直接进入下一次循环
        - $\odot_{\text{ans}} = \text{Circle}(\boldsymbol{p}_i, \boldsymbol{p}_j)$
        - For $k = 1 \rightarrow j - 1$
          - 若 $\boldsymbol{p}_k \in \odot_{\text{ans}}$ 则直接进入下一次循环
          - $\odot_{\text{ans}} = \text{Circle}(\boldsymbol{p}_i, \boldsymbol{p}_j, \boldsymbol{p}_k)$

- 随机增量法
  - 期望时间复杂度 $O(n)$
    - 最小覆盖圆至多只需要三个特定点确定
    - 每次进入下层循环的概率仅为 $O\left(\frac{3}{i}\right)$
  - 最小球覆盖
    - 三种情况→四种情况（四点确定球面）
    - 随机增量法期望渐进复杂度不变
      - 但求外接球的过程繁琐
- 模拟退火法
  - 随机圆/球心，取距离最大值为半径

反演变换

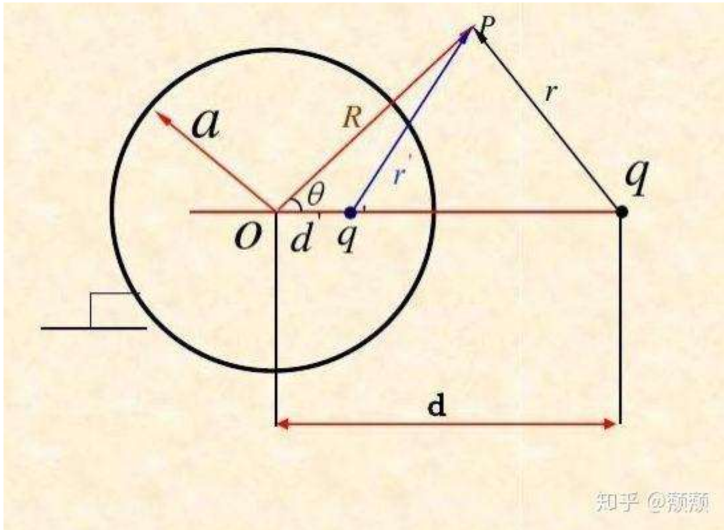
- 点的反演
  - 给定反演中心 $\boldsymbol{o}$ 和反演半径 r
  - $\boldsymbol{p}$ 与 $\boldsymbol{p}'$ 互为反演点
    - $\boldsymbol{o}, \boldsymbol{p}, \boldsymbol{p}'$ 三点共线
    - $\|\boldsymbol{p}-\boldsymbol{o}\|\|\boldsymbol{p}'-\boldsymbol{o}\| = r^2$
  - 性质：圆 $\boldsymbol{o}$ 外（内）的点，反演点在圆 $\boldsymbol{o}$ 内（外）
- 过点 $\boldsymbol{o}$ 的圆的反演 (2D)
  - $(x-x_0)^2+(y-y_0)^2 = a^2$
  - $x_0^2+y_0^2 = a^2$
  - $(x,y) \rightarrow (x',y') = \left(\frac{r^2}{x}, \frac{r^2}{y}\right)$





反演变换

- 不过点 $\boldsymbol{o}$ 的圆的反演 (2D)
  - 依然是不过点 $\boldsymbol{o}$ 的圆
- 反演的应用
  - 求过两圆外一点与两圆相切的所有的圆
- 三维推广
  - 圆→球；直线→平面
- 反演的意义
  - 几何图形关于球的"镜像"
  - 微分方程求解中的"电像法"



总结

- 参考资料
  - Mark de Berg et al. 邓俊辉译. 计算几何: 算法与应用 (第 3 版). 清华大学出版社. 2009.
  - 周培德. 计算几何: 算法设计与分析. 清华大学出版社. 2005.
  - 王华民. GAMES103: 基于物理的计算机动画入门, Lecture 09.
- 课后练习 (POJ)
  - 点、线、面
    - 2318, 2398, 3304, 1269, 2653, 1066, 1410, 1696, 3449, 1584, 2074
  - 凸包
    - 1113, 2007, 1228, 3348
  - 圆、球
    - 1375, 1329, 2354, 1106, 1673