扎实的编码能力和良好的编码习惯

精通几何图形算法，扎实数据结构与算法基础

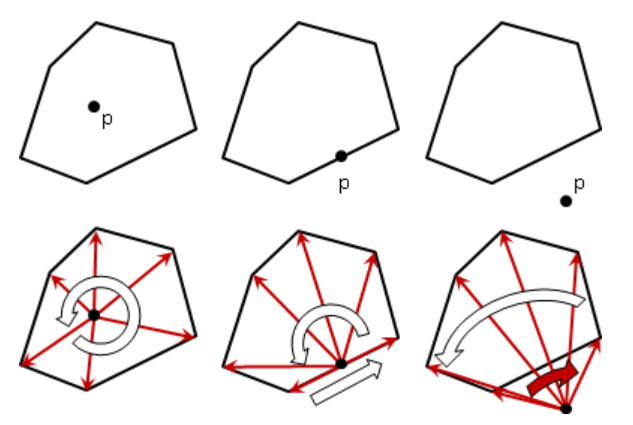
3d空间关系构建与建模算法研发

几何图形算法的架构设计与落地

凹包、点线面、几何、图形

点是否在凸包内？

1 外积计算法（只适用于凸多边形）O(N) 逐个判断是否同向



2 区域判别法 适用于快速判断多个点是否在一个凸多边形内 O(lgn)，找一个基准点和基准方向，极坐标角排序，二分得到极坐标角区域，判断待判断点与三角形顶点是否同向。

3 射线相交法 O(n) 进行n条线段的求交判断

从给定的点开始，往随便一个方向（习惯水平向右）引一条无限长的射线，看看穿过多少条边。如果穿过偶数条边，说明点在多边形外；穿过奇数条边，说明点在多边形内。 预先将与顶点重合短路、射线与目标线段平行时，额外判断是否在线上。

二维就逐个求交点，三维就参数方程求解析解。

4 面积比较法/夹角和

通过积分方法和基点方法分别计算面积，然后通过比较两种方法的面积判断基点的位置。如果面积相等，则说明基点在多边形内；如果不相等，则说明基点在多边形外。

1 顶点与所有多边形计算面积 2 鞋带公式算面积 xiyi+1-x+1yi

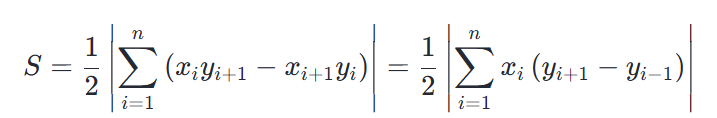
指定任意一点A，采取逆时针/顺时针进行标号，鞋带公式计算面积

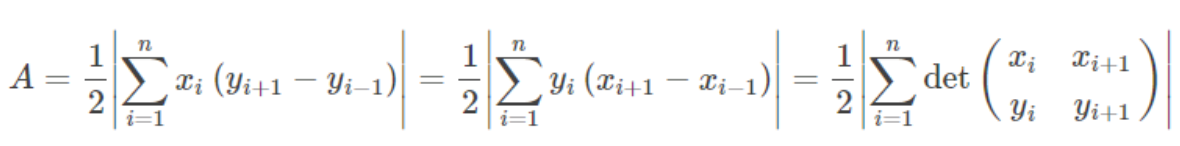
余弦定理(画图可证) c2 = a2 + b2 -2abcos theta

换算为向量 则有点乘公式向量a·向量b=a·b·cosTheta

向量叉乘，得到的是平行四边形面积，其方向通过右手定则确定，a·b·sin<a, b>

鞋带公式算面积





determinant行列式

5 共求凸包法

将凸包上的点与内部的点共同求凸包，如果最后求得的凸包中出现了原先没有出现的点，则证明有些点不严格在凸包内部。

计算多边形的形心

计算多边形的形心需要能反映顶点的顺序，不能直接做均值，因为相同的顶点集在不同的连接顺序（不同的边集）下，会生成不同的多边形。如果顶点沿一个共同的中心均匀分布，直接做均值才可行(比如规则多边形，矩形，正五边形…)；

凹多边形切割算法

启发于Graham扫描算法，构建当前直线与观察点数据，按序扫描多边形顶点逐个观察，观测到凹角点，出栈后不丢弃顶点，而是采用递归方式探测子凸多边形，直到探测到下一个凹角，切出子凸多边形后，回溯至上一层继续观测顶点，直到观测完毕所有顶点。

N+1维的数据才能将N维数据分割

一个栈不能模拟队列，只有两个栈可以，一个栈用来入队，一个栈用来出队。

# 凸包算法

根据三角形公式，凸多边形在周长上一定是最优的。

## Jarvis步进算法

jarvis算法O(n2)核心思想是双重遍历，选定一个初始点后，逐步迭代寻找与当前点构成的极坐标角度最小的点，检测到环路(寻找到最开始的初始点)时终止，注意在迭代寻找的过程中，考虑到多点共线的情况，全量查找时会查找到查找过的点，使用visited数组控制访问次数。

## Graham扫描算法

Graham扫描的时间复杂度为O(nlogn)，其时间复杂度主要前处理的排序上。主干凸包逻辑其实是O(n)的。把所有点放到二维坐标系中，选择最左边/最下边的点，建立局部参考坐标系，计算所有点距离点的极坐标向量，排序，将最小点压栈，记录最大点后，开始查找凸包：以原点与栈顶元素连接直线，扫描剩余点，尝试迭代扩大凸包面积，有两种情况：1 新的点与前向线方位一致（都是向左转），此时将新的点压入栈，表示新的凸包；2 新的点与前向线方位不一致（向右反转），此时将栈顶元素弹出，压入新的点；直到扫描至最后一个点。

## Andrew扫地机算法

Andrew算法与Graham算法流程类似，区别在于排序方式的不同，graham算法为角度排序，Andrew算法为x, y字典序排序后正反两次扫描。

# 扫描线

线段树比较多，矩形的面积并，周长并等

# 凹包算法

【滚球法】Alpha Shape 是一种基于Delaunay三角剖分的凹包计算方法。它通过设置一个参数α（通常是一个正数），来控制包围点集的“松紧程度”，即当α越小时，计算得到的边界越“凹”。Alpha Shape通常被认为是一种可以计算任意形状的凹包的有效方法。

Graham Scan与凹包算法的结合

Quickhull算法

K-nearest Neighbor算法

Monotone Chain算法的凹包变种

Shrinkage算法

Greedy方法

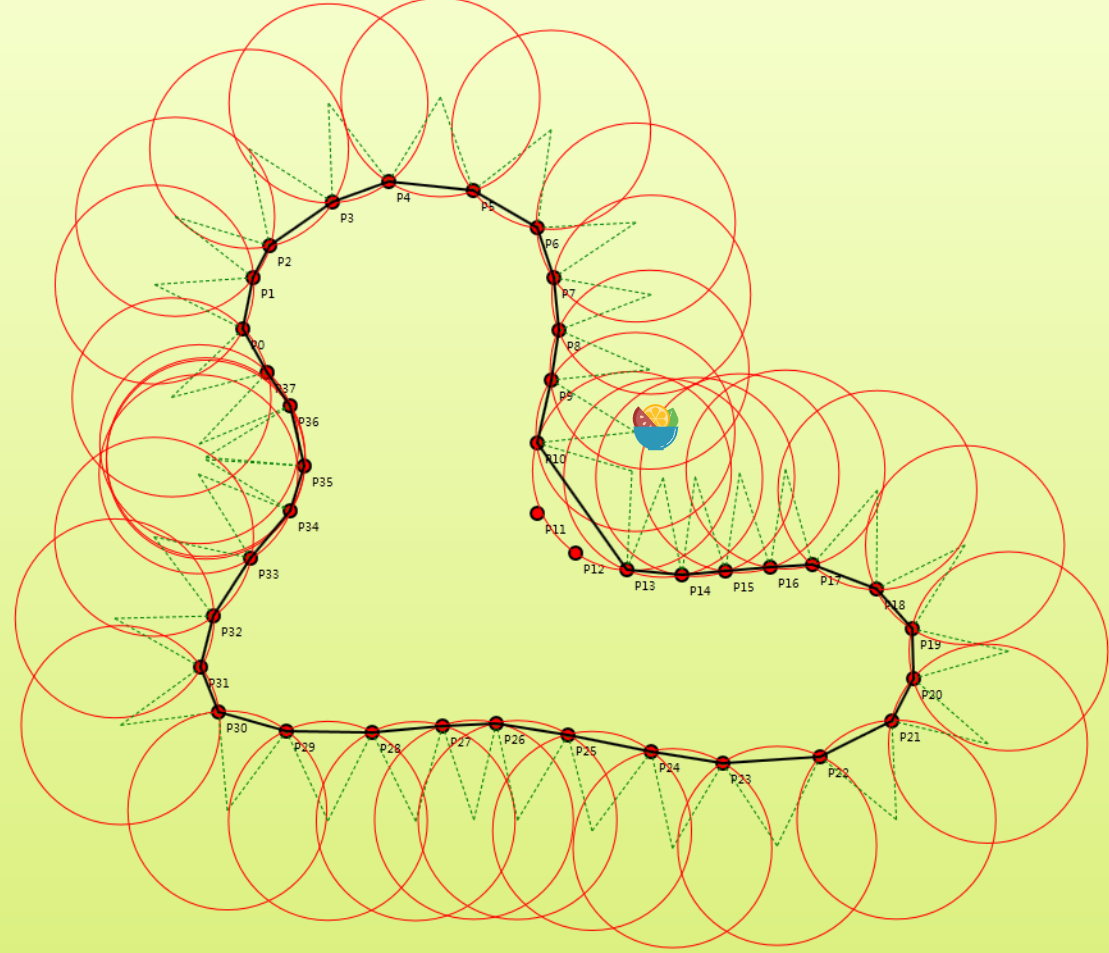
## 滚球法

求初始点p0，排序选定旁边的一个点，作为p1, 滚球法使用的球的半径，实际上代表的是一个扫描范围，过滤出来固定半径范围内的点：nearestPoints。

遍历 nearestPoints，以 P0->P1 角度为基准，逆时针旋转寻找最近的点。

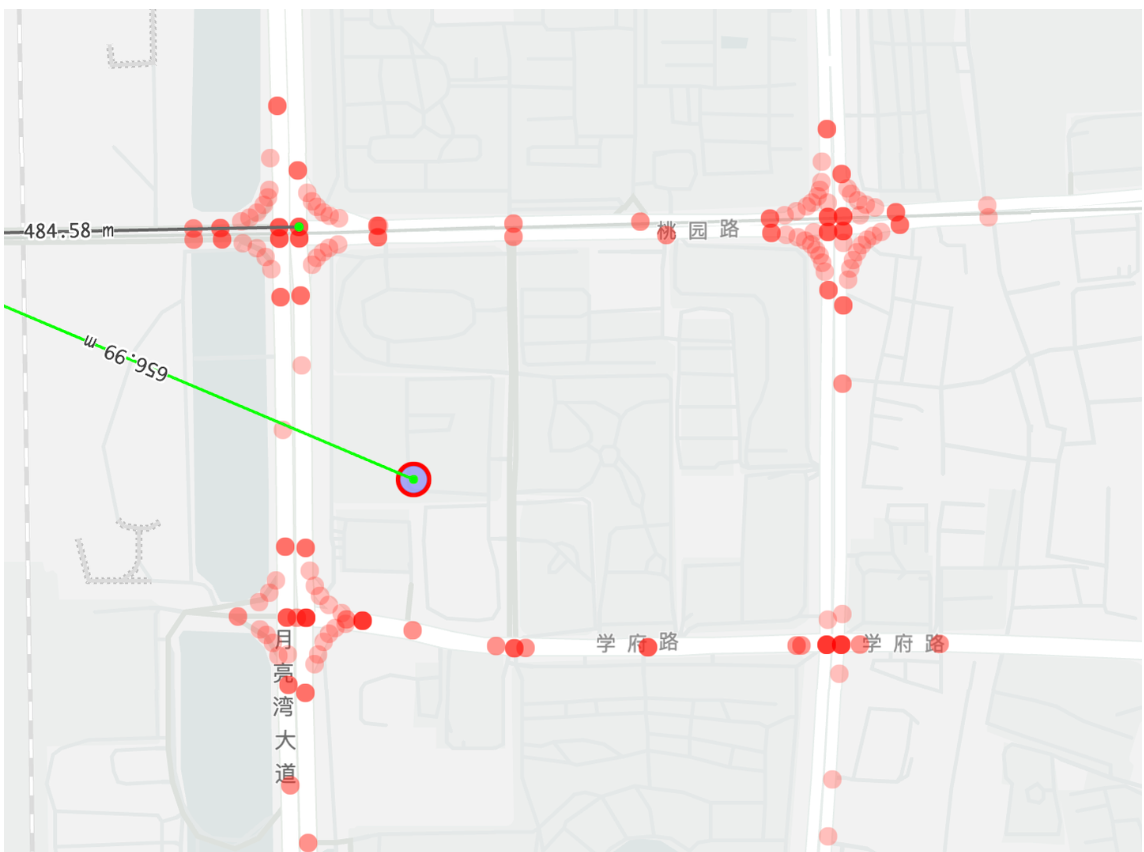
假设当前遍历到 P2

计算 P1->P2 的角度，然后比对 P0->P1和 P1->P2 在逆时针上最接近的 P2



改进凹包——自动伸缩滚球法

在初始半径内没有搜寻到复合结果的解，按某种规则（可以是指数关系）增加搜寻范围。直到找到结果



# 内凸包

内凸包的凸包不一定是绝对的凸包

网上关于内凸包的资料很少，讨论也比较少

我是这样理解内凸包的：在点的集合的一处塞入一个气球，给气球充气，在某个时刻，气球挤出去的形状就是内凸包。。

内凸包有一个中心参考点（气球的起点）

如下图，实现一个当鼠标在一群点中移动的时候，能够识别出内凸包，之后可以快速用比较合理的形状填充空白区域。

德劳内递归分割点集。

# 点线面空间几何

* 空间中两线段是否相交？

1 解析方法，求出解析式，看是否有解，要考虑除数为0等情况

2 计算几何，分别计算A,B两点是否在CD两侧；同时计算C、D两点是否在AB两侧。

三维上考虑异面情景：

问题的关键是求出这两条任意直线之间的最短距离，以及在这个距离上的两线最接近点坐标，判断该点是否在线段AB和线段CD上。

求向量AB在向量N方向的投影即为两异面直线间的距离了（就是最短距离啦）。

**最短距离的求法：d=|向量N\*向量AB|/|向量N|（上面是两向量的数量积，下面是取模）。**

* 空间中两直线相交的交点

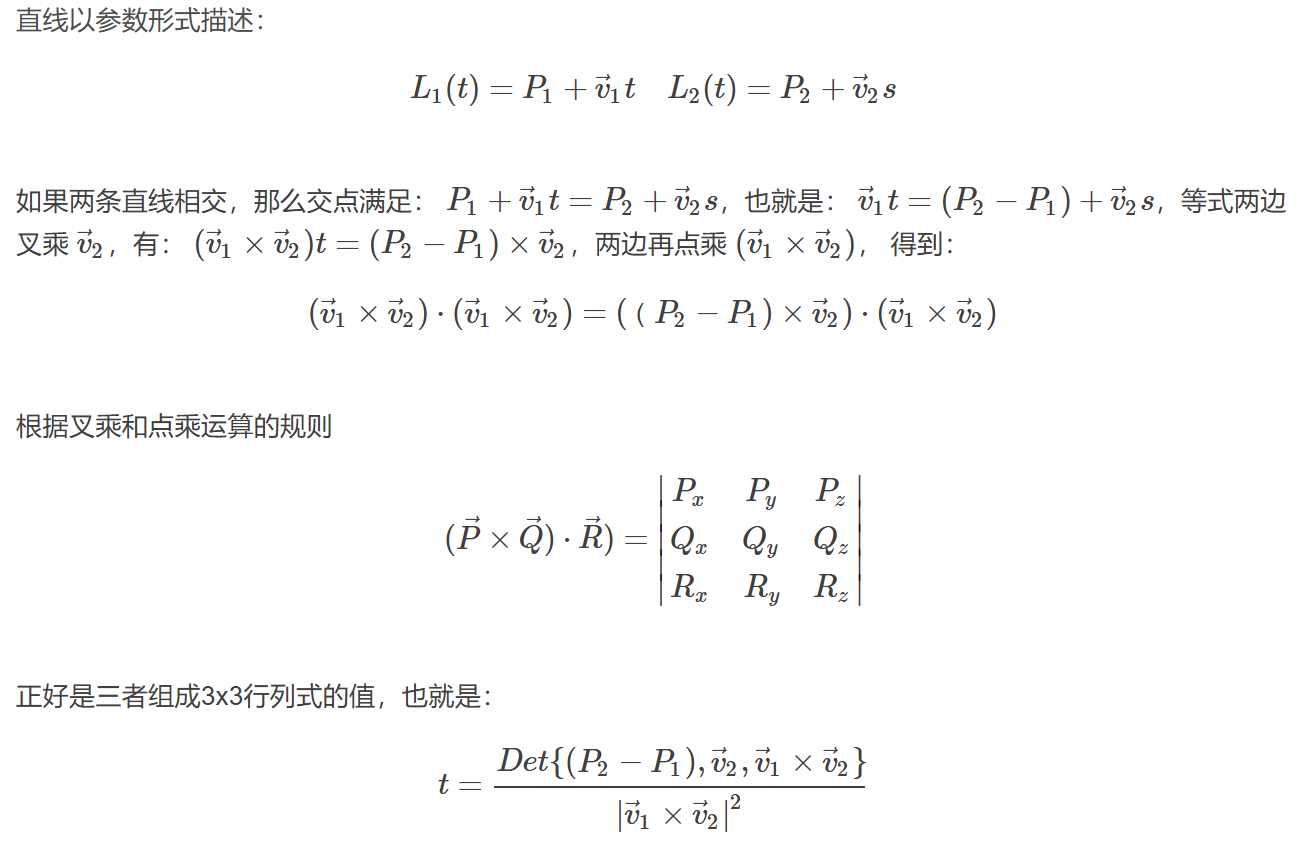
注意可以先利用叉乘，判断是否有交点，然后执行求交算法。

先利用单位向量叉乘，得到点到直线的距离；利用三角函数转换，tantheta = √1/cos2theta – 1

然后利用单位向量点乘，求直角边，得到投影点

利用向量运算，可以得到两个候选根，距L2近的即为所求根。

也可以直接利用参数方程，求解函数参数



* 空间中点到直线的距离

在直线中取两个点AB，计算AC与AB的外积，除以AB的模. 或者直接将AB取为单位向量也可以。

* 空间中点到直线的投影点

算出三角形距离后，从直线中运动下来

* 空间中点到线段的距离

我们有集中不同的方法来判断这种特殊情况。第一种情况是计算点积AB Bc来判定两线段间夹角。如果点积大于等于零，那么表示AB到BC是在-90到90度间，也就是说C到AB的垂线在AB外，那么AB上到C距离最近的点就是B。同样，如果BAAC大于等于零，那么点A就是距离C最近的点。如果两者均小于零，那么距离最近的点就在线段AB中的某一点。

* 判断点与射线是否相交

将射线端点分别与线段两端点连结，得一锥形区域，判定射线方向是否在这个锥里。

* 空间中点到空间三角形的距离

计算点到平面的投影点，如果在三角形内，就是该点；如果不在三角形内，则最近点一定在三角形上，分别求点到三角形三条边的最短距离。

也可以采用解析式求解，将三角形内的点用参数化描述，则两点距离可以表述为两个点的参数化2范数，在空间上是一个抛物面，然后求偏导处理，具体结果我这边不太记得了。

* 点到面的距离

设某三维平面表达式为

a\*x+b\*y+c\*z+d = 0;

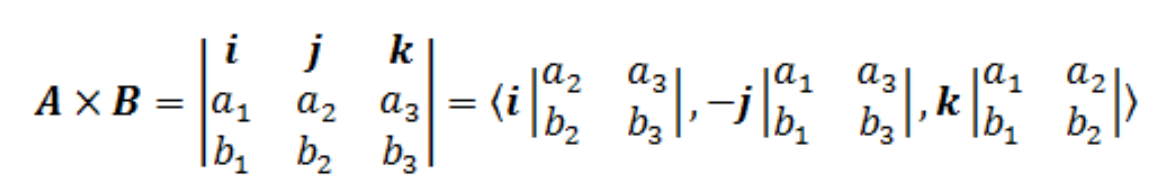
则其法向量即为: (a,b,c).

任意一点 p = (x1, y1, z1)到该平面的距离为：

(a\*x1 + b\*y1 + c\*z1+d) / (a\*a + b\*b +c\*c)，任意选一点在法单位向量上的投影

# 计算几何数学

* 点乘表征角度（相关性），叉乘表征距离（变化幅度）。定位问题涉及到距离+方向，需要综合的三角运算。
* 三维向量叉积的公式



* 先叉乘，后点乘的向量可以转换为三阶行列式处理
* 行列式的计算中，二四象限对角线为主对角线。
* 空间直线的方程表示：

Ax+by+cz=d能表示一个平面，三个变量，一个方程，具有二维的自由度；(a, b, c是平面的法向量)

直线可以由两个参数平面联立表示

其点向式表示形式为（和对称式相同）（x－x0）／l＝（y－y0）／m＝（z－z0）／n，其[方向向量](https://zhidao.baidu.com/search?word=%E6%96%B9%E5%90%91%E5%90%91%E9%87%8F&fr=iknow_pc_qb_highlight)就是（l，m，n）或反向量（－l，－m，－n）。

但是点向式表示法存在某一维度可能退化的问题，0不能做除数，计算机常用直线的参数方程P = P0 + tL, 来表示直线/线段，L是方向向量，t是参数，直线是-无穷到正无穷。

直线的方向向量可以由两个平面的法向量外积得出。

异面直线距离可以把交线求出来后做投影

两直线方向向量的夹角可以称为两直线的夹角，可以由内积反算得出

* 给定两个点，怎么表示一条空间直线

给定两个点(x0, y0, z0), (x1, y1, z1)，实际上就可以确认直线的方向向量（x1-x0, y1-y0, z1-z0），可以使用点向式表示x-x0/x1-x0=y-y0/y1-y0=z-z0/z1-z0

* 为什么截距式平面方程的法向量是ABC？

取平面上任意一点x0(a, 0, 0)，则基于该点的向量x0x = (x- a, y, z)

X0x · ABC向量最后可以得到恒等于0，则有截距式平面的法向量为(A,B,C)。

从另一个角度来说，在一个R3的空间中确定一个点和一个方向，就能确定一个平面，平面上所有点与P构成的直线与法向量垂直，即内积为0，据此，可以推导出截距式方程。

* 行列式为0的意义

每一个线性变换都对应着一个变换矩阵，被变换后的空间，相对之前来说也发生了一定的形变，**而行列式的意义则是线性变换前后，空间形变的倍数**。行列式为0表示行（或列）向量组线性相关。对一个几何采用该行列式的变换，会产生降维且不可逆的结果，三维体会被压缩为二维面；二维面会被压缩为1维线；线会被压缩为一个0维的点。

仿射矩阵实际对应的是基向量的变化。

* 矩阵A的秩的几何意义

矩阵A的秩刻画了其对于空间几何体的压缩能力

* 牛顿法算平方根