

GIS 中的计算几何

GIS 是一个图形系统，必然会涉及到几何学上理论应用，比如，图形的可视化，空间拓扑分析，GIS 图形编辑等都需要借助几何。向量几何是用代数的方法来研究几何问题，首先，请大家翻一翻高等数学里有关向量的章节，熟悉一下几个重要的概念：向量、向量的模、向量的坐标表示、向量的加减运算、向量的点积、向量的叉积...下面我们将用这些基本概念来解答 GIS 中一些几何问题。

一，点和线的关系。

点是否在线段上，这样的判断在图形编辑，拓扑判断(比如，GPS 跟踪判断是否跑在线上)需要用到这样的判断。通常的想法是：先求线段的直线方程，再判断点是否符合这条直线方程，如果符合，还要判断点是否在线段所在的矩形区域(MBR)内，以排除延长线上的可能性，如果不符合，则点不在线段上。这种思路是可行的，但效率不高，涉及到建立方程，解方程。借助向量的叉积（也叫向量的向量积，结果还是向量，有方向的）可以很容易的判断。设向量 $\mathbf{a}=(X_a,Y_a,Z_a)$ $\mathbf{b}=(X_b,Y_b,Z_b)$ 向量叉积 $\mathbf{a} \times \mathbf{b}$ 如下：

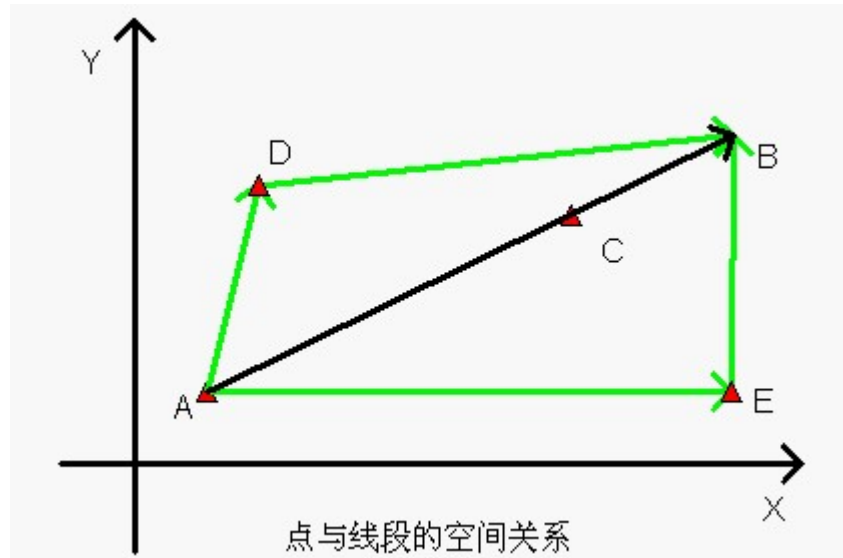
$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ X_a & Y_a & Z_a \\ X_b & Y_b & Z_b \end{vmatrix} = (Y_a \cdot Z_b - Z_a \cdot Y_b) \cdot \mathbf{i} - (X_a \cdot Z_b - Z_a \cdot X_b) \cdot \mathbf{j} + (X_a \cdot Y_b - Y_a \cdot X_b) \cdot \mathbf{k}$$

$\mathbf{i}, \mathbf{j}, \mathbf{k}$ 分别为 X, Y, Z 方向上的单位向量

$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ X_a & Y_a & 0 \\ X_b & Y_b & 0 \end{vmatrix} = (X_a \cdot Y_b - Y_a \cdot X_b) \cdot \mathbf{k}$$

如果在二维平面上，则 Z_a, Z_b 都为 0

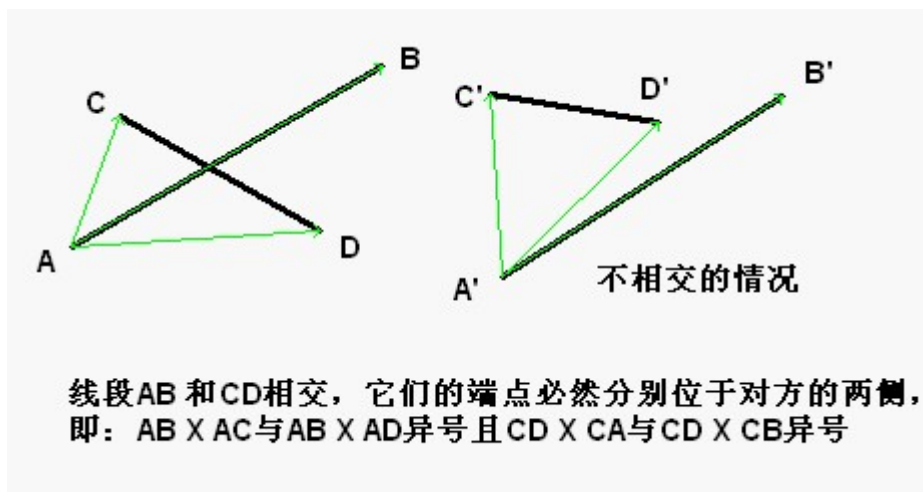
二维向量叉积的模 $|\mathbf{a} \times \mathbf{b}| = |\mathbf{a}| \cdot |\mathbf{b}| \cdot \sin \alpha = |X_a \cdot Y_b - Y_a \cdot X_b|$ (α 是向量 \mathbf{a}, \mathbf{b} 之间的夹角)，向量叉积模的几何意义是以向量 \mathbf{a}, \mathbf{b} 为邻边的平行四边形的面积。可以推测：如果两向量共线，向量叉积模(所代表的平行四边形的面积为零) 则 $|\mathbf{a} \times \mathbf{b}| = |\mathbf{a}| \cdot |\mathbf{b}| \cdot \sin \alpha = |X_a \cdot Y_b - Y_a \cdot X_b| = 0$ ，否则不共线，叉积的模为非零，根据这样条件可以很轻松的判断点和线的关系，避免了建立方程和解方程的麻烦。



向量叉积的模 $|AB \times AC| = 0$ 即可判断 C 点在 AB 所确定的直线上, 再结合 C 点是否在 AB 所在的 MBR 范围内, 就可以最终确定 C 是否在 AB 线段上。关于点和线段的其他关系, 都可以通过叉积求得, 比如 判断点在线的哪一侧, 右手法则, 可以通过 $a \times b = (X_a * Y_b - Y_a * X_b) * k$ 中的 $(X_a * Y_b - Y_a * X_b)$ 正负来判断。留给大家思考, 很简单的, 呵呵...

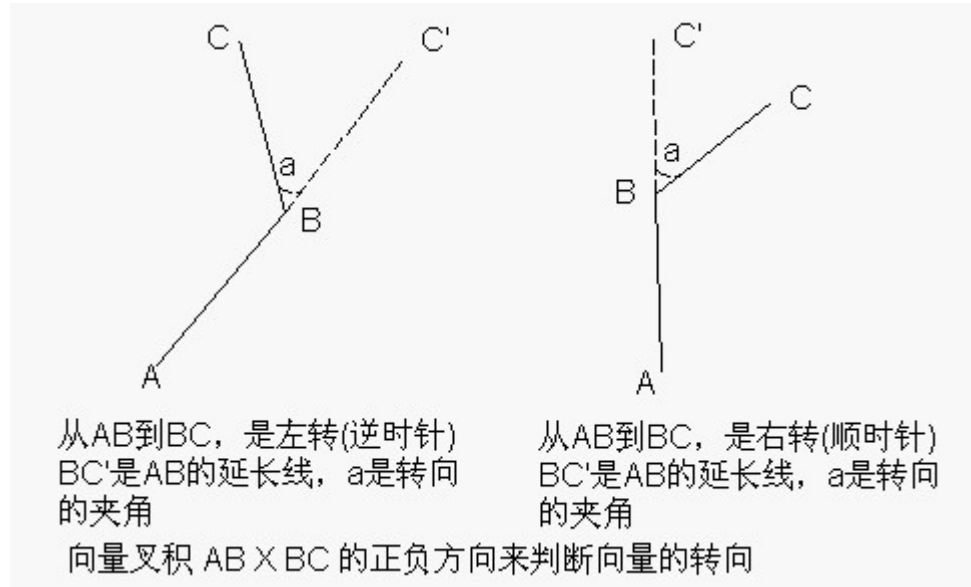
二, 线和线的关系

判断两条线段是否相交, 在很多拓扑判断和图形编辑 (比如, 线的打断来构建拓扑, 编辑线对象, 叠置分析, 面与面关系的判断等) 中都需要用到线线相交的判断, 如果两条线段相交, 一条线段的两个端点必然位于另一条线段的两侧 (不考虑退化情况, 也就是一条线段的端点在另一条线段上, 这个很容易判断)



两向量的叉积 $a \times b = (X_a * Y_b - Y_a * X_b) * k$, 分别判断 $AB \times AC$ 的方向与 $AB \times AD$ 的方向是否异号, 再判断 $CD \times CA$ 的方向与 $CD \times CB$ 的方向是否异号即可判断两线段是否相交。

退化情况, 一条线的端点在另一条线上, 则 $AB \times AC$ 、 $AB \times AD$ 、 $CD \times CA$ 、 $CD \times CB$ 是否存在有一个为零的, 存在则表明肯定有一条线的端点在另一个线上或者共用一个端点。详细区分留给大家思考。呵呵... 利用向量的方向还可以判断线段的转向, 这个在道路导航中有所应用



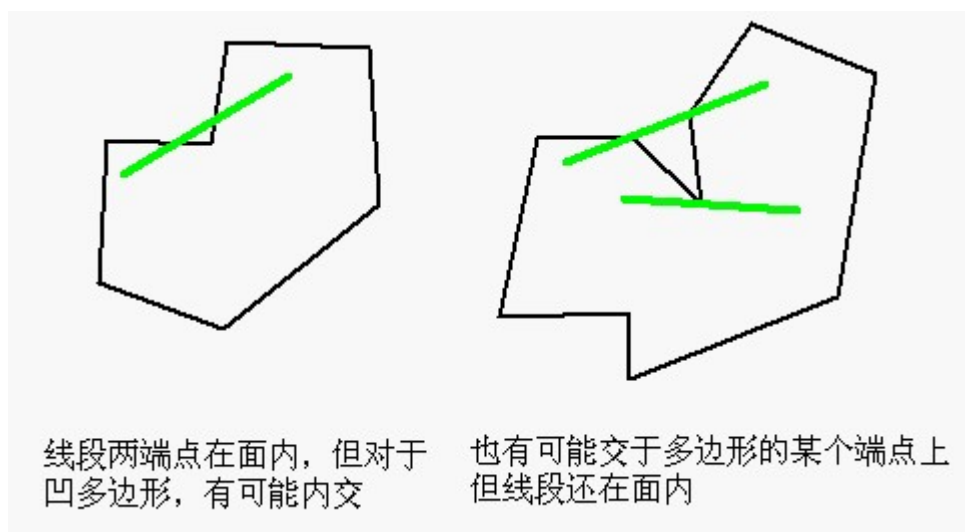
三，点和面的关系

在各种拓扑判断中（比如，面对象的选取，包含关系的判断等）有时需要判断一个点是否位于某个面内，经典的方法就是“垂线法”，在直角坐标系中，从这个点向 X 轴作射线，判断射线与多边形的交点个数（不考虑退化情况，退化情况下，判断点或者射线与多边形端点或者边的关系），如果为奇数，则点在面内，为偶数，则点在面外。

四，线和面的关系

线面关系的判断相对比较复杂，线在面内，线和面相交，相离，相接等关系。**线段**在面内，第一个必要条件是，线段的两个端点都要在内。但由于多边形可能为凹，所以这不能成为判断的充分条件，于是有第二个必要条件线段与多边形的边，没有内部交点。

线段和多边形交于线段的两个端点并不会影响线段是否在多边形内；但是如果多边形的某个顶点和线段相交，还必须判断两相邻交点之间的线段是否包含于多边形内部，如果在面内，则线段在面内，否则不在面内。



所以，算法思路如下（本算法引用网络上一篇文章）：

```

if 线段 PQ 的端点不都在多边形内
then return false;
点集 pointSet 初始化为空;
for 多边形的每条边 s
do if 线段的某个端点在 s 上
then 将该端点加入 pointSet;
else if s 的某个端点在线段 PQ 上
then 将该端点加入 pointSet;
else if s 和线段 PQ 相交 // 这时候已经可以肯定是内交了
then return false;
将 pointSet 中的点按照 X-Y 坐标排序;
for pointSet 中每两个相邻点 pointSet[i] , pointSet[i+1]
do if pointSet[i] , pointSet[i+1] 的中点不在多边形中
then return false;
return true;

```

注：X-Y 坐标排序，X 坐标小的排在前面，对于 X 坐标相同的点，Y 坐标小的排在前面，这种排序准则也是为了保证水平和垂直情况的判断正确
点在面内，线段相交情况的判断见上面的思路。

这个过程排序因为交点数目肯定远小于多边形的顶点数目 n ，所以最多是常数级的复杂度，几乎可以忽略不计。因此算法的时间复杂度也是 $O(n)$ 。

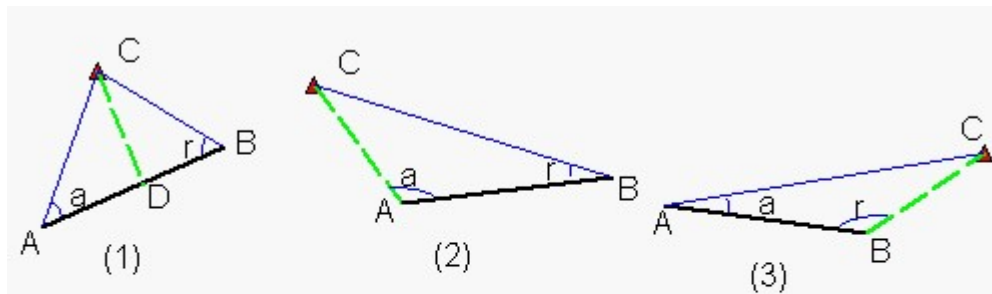
有了线段和面的关系，再判断折线与面的关系，也就可以 for 循环，同理进行判断了。

五，面和面的关系

面面的空间关系，可能要更复杂一些，在拓扑判断，多边形叠置分析，面对象的编辑中，有着广泛的应用。这个将在以后的章节中介绍一种时间复杂度为 $O(n \log n)$ 的算法“平面扫描算法”。

六，点到线段的距离

点到线段的距离，在各种测量，拓扑判断(比如，线对象的选取中需要比较距离)中都需要用到。大家对点到直线的距离，都很熟悉，那点到线段距离又该怎么计算呢？



- (1) α 、 r 都是锐角，则C到AB的最短距离为CD
 (2) α 为钝角，则C到AB的最短距离为CA
 (3) 与(2)情况类似， r 为钝角，则C到AB的最短距离为CB

问题的关键是判断 α 、 r 的角度，向量的点积能判断一个角是钝角还是锐角，先复习一下向量的点积，也叫向量的数量积，结果是一个数，没有方向。设向量 $\mathbf{a}=(X_a,Y_a,Z_a)$ $\mathbf{b}=(X_b,Y_b,Z_b)$

$\mathbf{a} \cdot \mathbf{b}=|\mathbf{a}|*|\mathbf{b}|\cos\alpha=X_a*X_b+Y_a*Y_b+Z_a*Z_b$ 向量点积的几何意义是，高中物理中，求作用力在一个方向上所作的功。如果 $\mathbf{a} \cdot \mathbf{b}>0$ ，则 α 为锐角， $\mathbf{a} \cdot \mathbf{b}<0$ ，则 α 钝角。

熟悉了利用向量的点积来判断角度， $\mathbf{AC} \cdot \mathbf{AB}$ 判断夹角 α ， $\mathbf{BA} \cdot \mathbf{BC}$ 判断夹角 r ，即可确定三种情况中，具体是哪一种。至于第一种情况，求点到垂足的距离，可以绕开建立方程求垂足，再求两点距离的思路，因为运算就复杂了，多耗了 CPU 资源。利用向量叉积的几何意义来求，向量的叉积表示两向量为邻边的平行四边形的面积 $|\mathbf{AC} \times \mathbf{AB}|$ 为 $\triangle ABC$ 的面积的两倍，求平行四边形的高，只要用面积除以底边 AB 的长度，高 CD 的长度= $|\mathbf{AC} \times \mathbf{AB}|/\text{distance}(\text{AB})$ 。

这些复杂的几何判断，都将在空间索引的过滤下，在少量数据集(候选集)上进行。计算几何算法，通常是比较复杂，比较耗 CPU 的运算，而且还要考虑各种退化的情况，在这里，并不试图向大家穷举各种情况，只想起一个抛砖引玉的作用，或许还有人会有这样的疑问：有没考虑“投影”的问题？关于投影将在相应的章节中给予解释，但有一点是可以肯定的，空间分析计算几何算法，都是在直角平面上运算的，不会在球面上。

参考文献：计算机地图制图 蔡先华 东南大学 2000.08

作者：陈玉进 geochenyj@hotmail.com

07.01.30