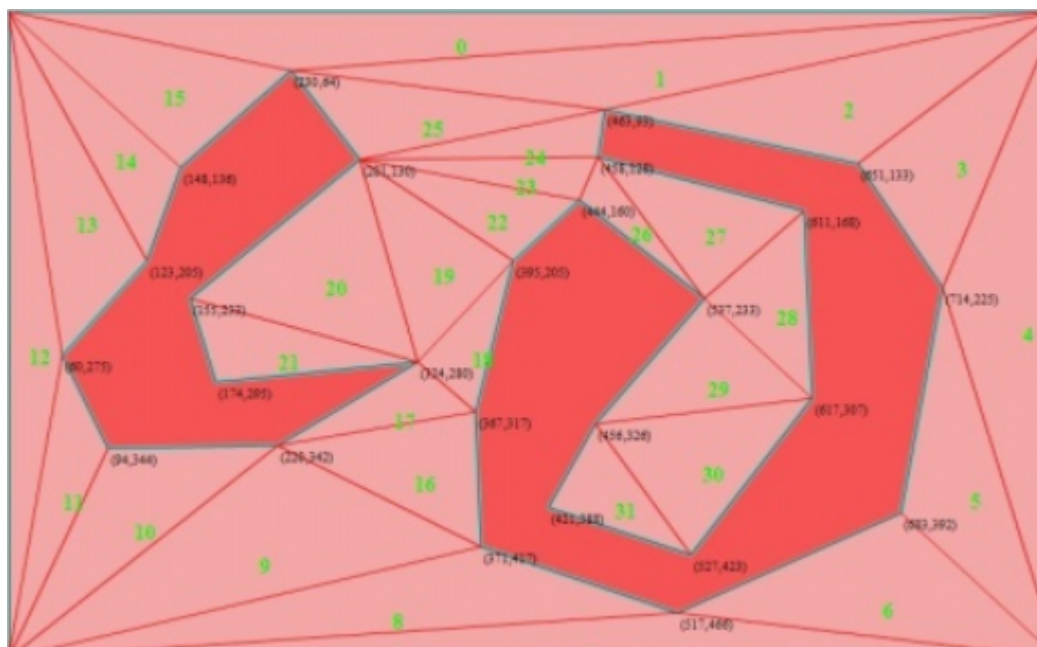


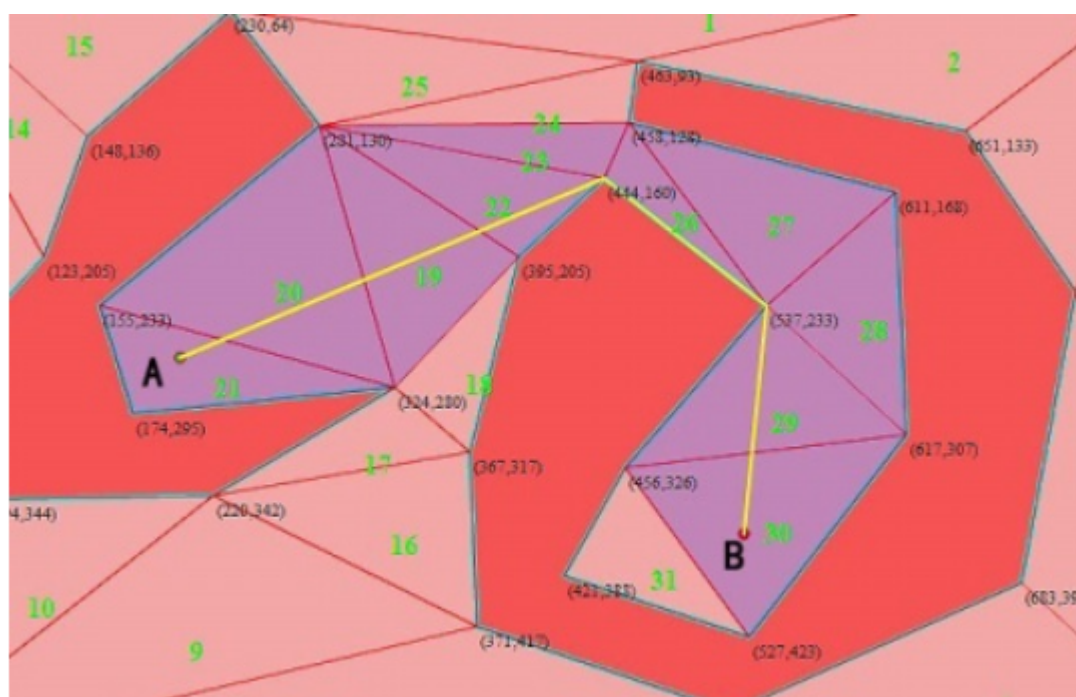
## NAV导航网格寻路--寻路方法

### 一. 使用A\*寻找所经过网格路径

下图为一个已经生成nav网格的地图，深红色区域为不可行走区域，浅红色区域为可以行走的区域。



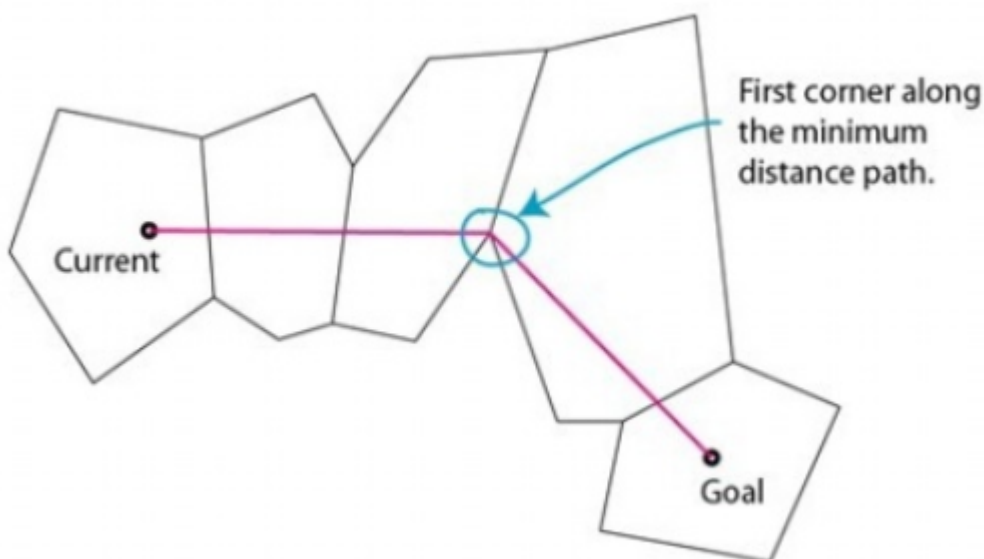
如下图，现在如果要寻找从A点到B点的路径，首先要从所有可行走的网格中找到一条最优的网格路径（图中紫色的网格），然后再根据这些网格生成所需要的路径点。



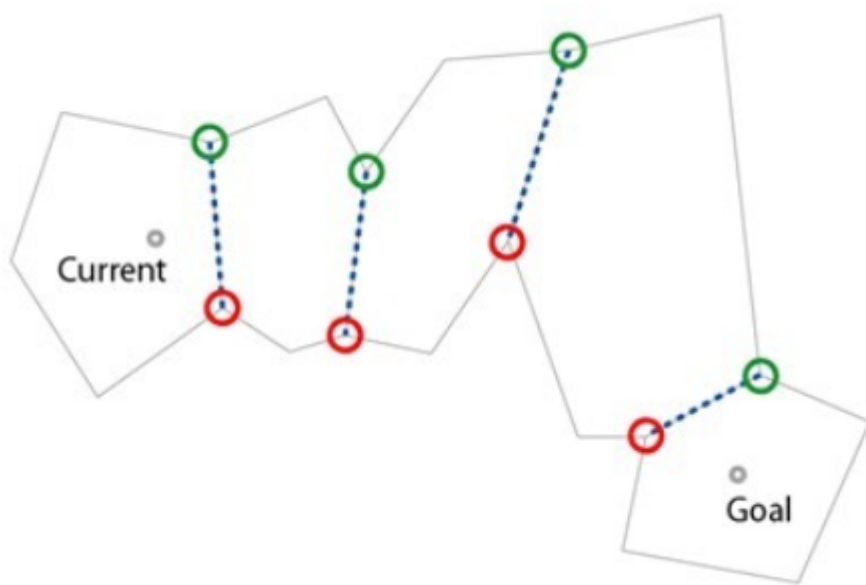
### 二. 生成路径点

拐角点法，如下图

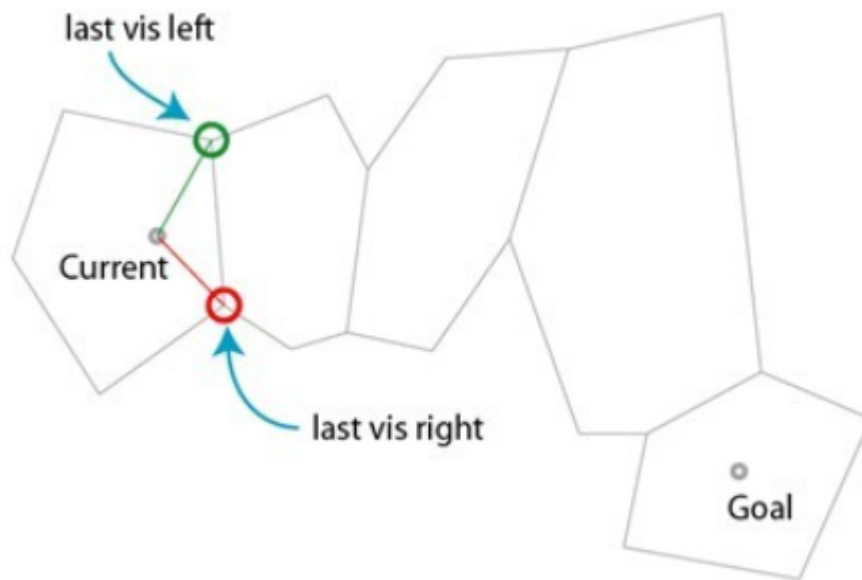
下图的5个凸多边形是已经生成的导航网格，多边形外部的区域为不可行走区域，current为起点，goal为终点，从图中就可以看出最短路径为图中红线，蓝色圈出的点为我们需要找出的点。所有多边形顶点均按逆时针方向存储（这些均在生成导航网格时处理，以后会讲到）。



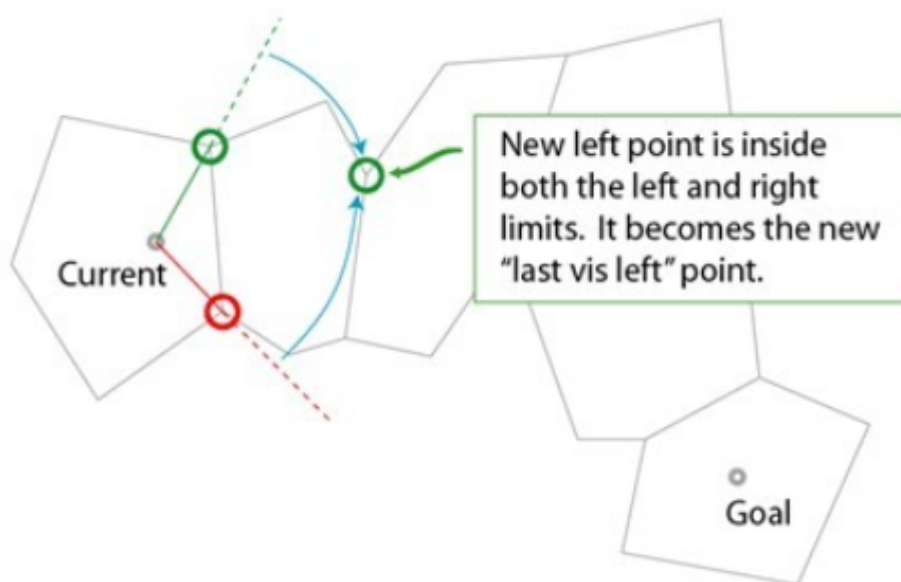
（1）下图显示出各区域之间的入口，即多边形的临边。由图中可以看出每个临边均为起点穿出该多边形区域的边，故以下称该边为穿出边。



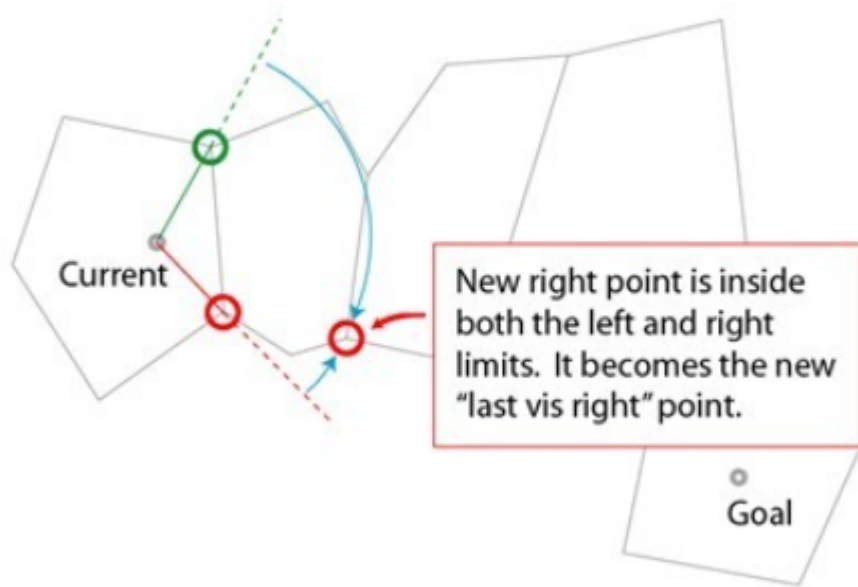
（2）首先找到起始点所在的多边形和穿出边的两个端点，由起点连接两个端点，形成两个线段lineLeft 和 lineRight。如下图。绿色圈表示左点，红色表示右点（左点、右点是根椐多边形顶点保存顺序而来）。



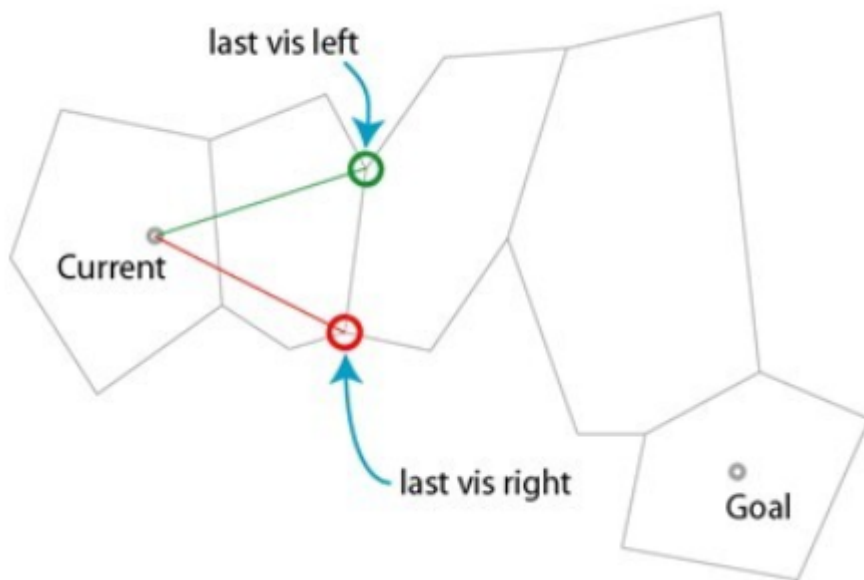
(3) 继续找到下一个穿出边的两个端点，判断新的左点是否在lineLeft 和lineRight之间，如果在，则更新lineLeft为起点到新左点的线段。



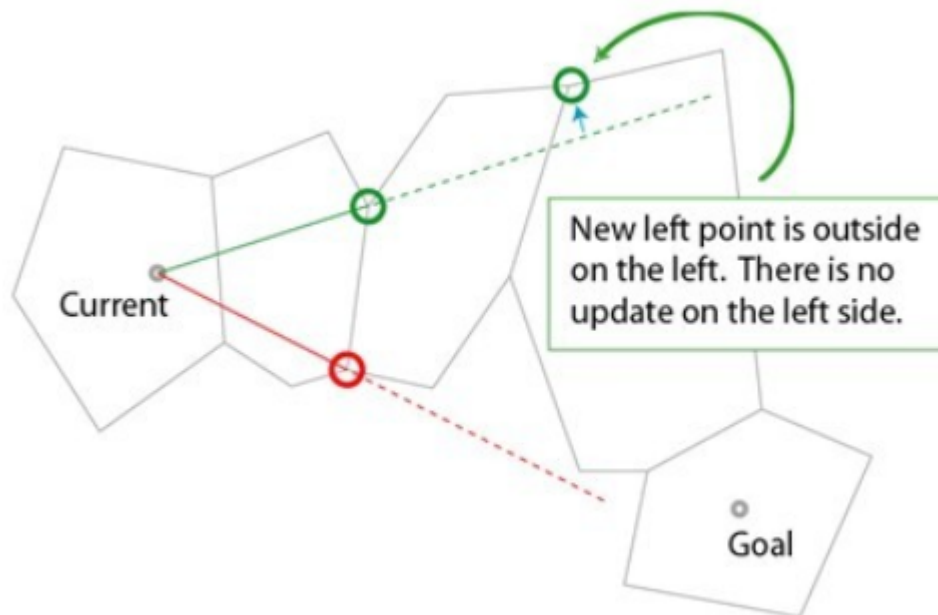
同样处理新穿出边的右点，如下图



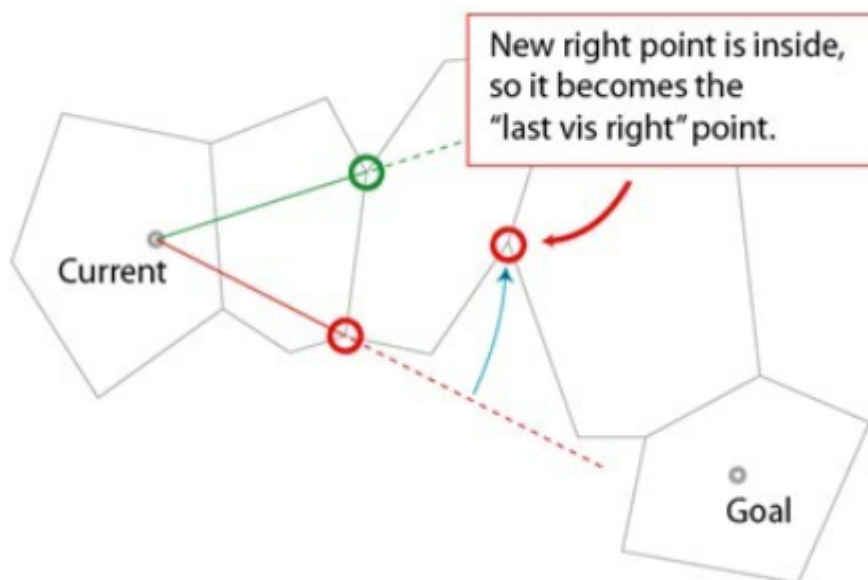
该步最后得到两个新的线段，如下图。



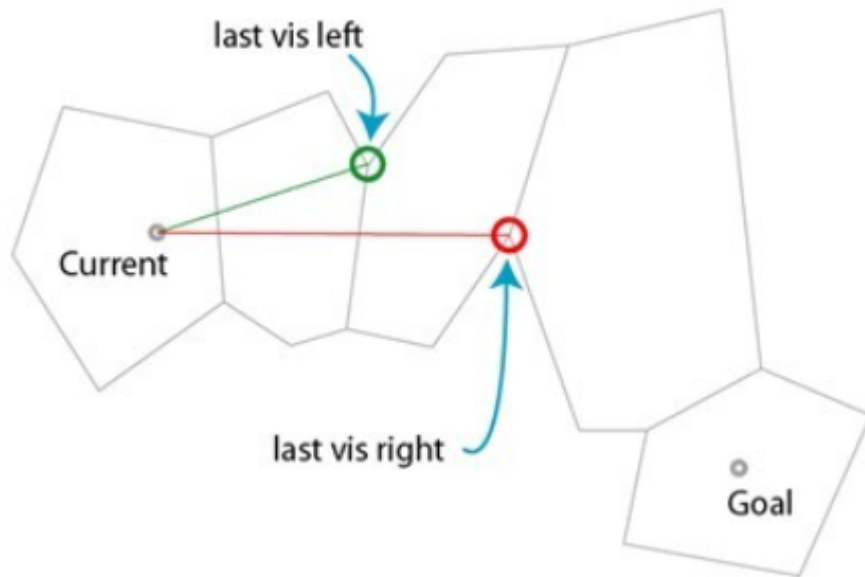
(4) 继续判断下一个穿出边的两个端点，如下图，新的左点在lineLeft和lineRight的外面，则不更新线段。



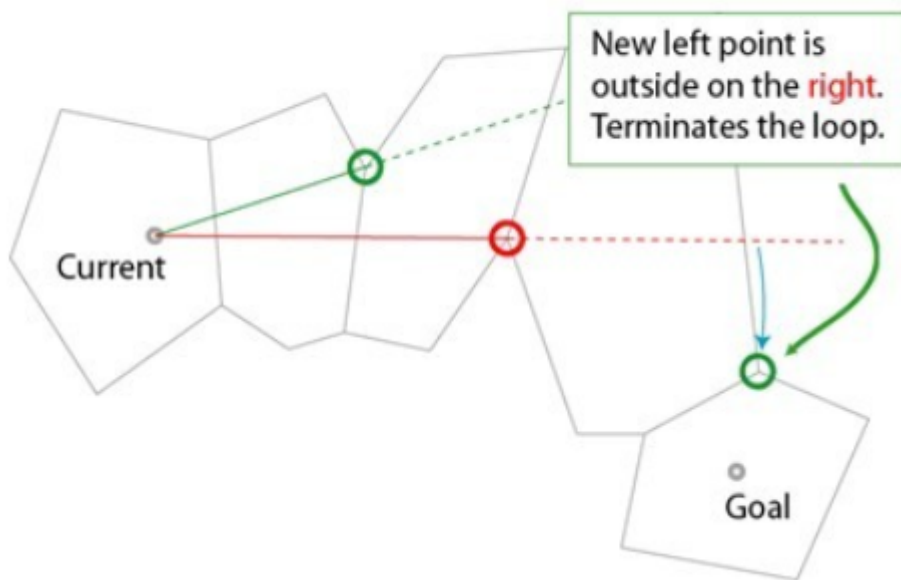
下图说明新的右点在两条直线之间，更新lineRight。



该步最后得到两个新的线段，如下图。



(5) 继续循环判断下一个穿出边的两个端点，该穿出边的两个端点都在lineRight的右侧，表示lineRight的终点即为路径的一个拐角点。



(6) 循环以上步骤都可以找到从起点到终点的一条完整路径