基于可见多边形生成算法的 PlanarSight 游戏

计算几何课程实验选题报告

罗必成，温佺，陈翔

意义动机：

数学当中的很多定义都来源于日常的生活当中。而且，也正是日常生活当中的灵感不断的灌注其中，数学的世界才有了如此鲜活的生命力。在几何学当中，可见性的问题就是一个很好的例子。所谓可见性问题是指，在欧式空间当中放置一系列障碍物，如果该空间当中的两个点的连线不与任何障碍物相交，那么我们就定义他们两个点是互为可见的。当然，这是一个基本的问题。随着问题的进一步拓展，我们会出现连线只能与障碍物相交 k 次，甚至连线可以凭借障碍物发生反射等等特殊的要求。这些要求让可见性这个问题变得更加丰富多彩起来。

可见性的判断与运算是计算几何当中非常基础的一类问题，有着丰富的理论研究价值。但除了理论研究以外，可见性在计算机图形学或者运动规划等实际问题当中，也有着非常重要的作用。例如，渲染一个网格模型时，通过判断哪些点或者哪些面是可见的，从而有选择地渲染一部分数据，最终实现降低性能耗损，实现实时渲染的目的。再例如，一个清洁机器人在楼层里打扫卫生，它需要再不撞到任何问题的要求下，判断下一步需要进行的动作序列是什么。通过楼层当中物体可见性的运算，可以在给定约束条件下，找到最优的运动规划。

由于可见性问题的应用具有一定的代表意义，并且也是计算几何研究当中的重中之重，除此之外也富有展示性和教学性。所以，我们组选择的课程实验选题是"基于可见多边形生成算法的 PlanarSight 游戏"。

PlanarSight 是这样一款小游戏，用户可以在窗口当中随意地绘制出一个带障碍物的多边形房间，并且在多边形房间当中布置一系列的随意走动的 Monster。然后游戏一开始，这些 Monster 就开始在房间当中巡视，他们有着相应的朝向和视角大小。然后玩家要用鼠标移动控制游戏的主角 Hero，在不被 Monster 发现（也就是不在 Monster 的可视多边形范围内）的情况下逃离这个房间。这个游戏的实现背后所需要的理论基础就是对平面含洞多边形中的某个点求其可见多边形，再判断某个点是不是在这个可见多边形内。与游戏联系起来，还会有动态更新，计算的需求。

通过这个项目的研发，我们希望可以更好地对可见性问题产生深入的理解。我们也希望通过这个小游戏，从寓教于乐的角度出发，让大家对计算几何问题的研究产生兴趣。

技术路线：

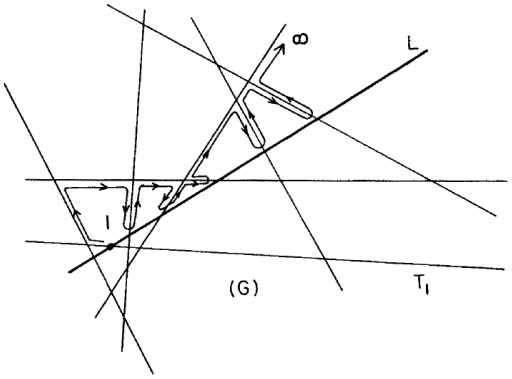
如上文所述，我们需要解决的最基本的问题是对于一个平面含洞多边形 P 和其中的一

点 q，求 q 关于 P 的可见多边形。我们计划采用的算法是 Asano[1]，算法需要 $O(n^2)$的时间和空间进行预处理（n 指的是平面含洞多边形的边数），但仅需线性时间计算可见多边形。下面将介绍算法的主要步骤。

预处理的部分分为计算直线排布和计算三角剖分：

（1）计算直线排布

由于在后面的算法中需要计算多边形 P 所有顶点关于 q 的极角序，并且一般的排序算法在 O(nlogn)时间内完成无法满足线性时间的条件，因此在这里其参考了 CHazelle[2]中的算法。其中将平面内的点和直线按一定关系转化为相应的直线和点，这样就将原先的极角（q 与 P 顶点所在直线的斜率）转化为变换后直线间交点的横坐标。接着需要计算一组直线排布，如下图所示，其中的直线由原先多边形 P 的顶点转化而来，我们可以在 $O(n^2)$的时间内计算它们的交点以及将其表示为 DCEL 的形式。当加入一条新的直线 L（由 q 转化）时，我们可以根据 DECL 表示中边之间的关系在线性时间内得到 L 与所有直线交点横坐标的有序关系，从而得到了多边形 P 所有顶点关于 q 的极角序。



（2）计算三角剖分

三角剖分用于计算多边形 P 中的边关于点 q 的一种偏序关系，有很多成熟的算法可在 O(nlogn)的时间内完成。

算法的主体步骤包含计算顶点极角序，计算多边形边关于输入点的偏序关系以及线性集合，并计算可视多边形三个方面：

（1）计算顶点极角序

计算极角序的预处理工作已在前文介绍，当给定一个输入点 q 时，使用直线排布辅助便可以在线性时间内计算极角序从而使整个计算过程在线性时间内完成，对于提高实时更新可视多边形的效率是非常有利的。

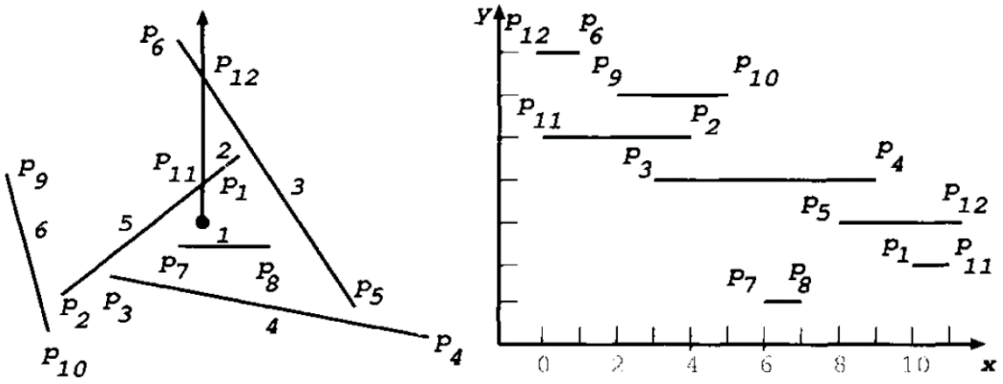
（2）计算多边形关于输入点的偏序关系

在此处，算法中定义了一种多边形 P 中所有边关于输入点 q 建立起的偏序的关系，即对于两条线段 s_i 和 s_j，若存在从 q 引出的一条射线与两线段均相交，且这条射线先与 s_i 相交再与 s_j 相交，则记作 $s_i <_q s_j$。利用预处理得到的多边形 P 的三角剖分 T，我们需要根据输入点 q 在 T 上作一些更新，并根据 T 中三角形与 q 的位置关系得到一个关于$<_q$关系的有向无

环图 G（DAG），对 G 作拓扑排序后便可得到多边形 P 中的顶点关于 q 的$<_q$关系。

（3）线性集合并计算可视多边形

最终计算可见多边形的算法利用了前面两步的结果，其将多边形顶点的极角序转化为 x 坐标，多边形边关于输入点 q 的偏序关系转化为排序的序号值并进一步转化为 y 坐标，从而将原先如下面左图中的问题转化为了右图中对于每个 x 坐标求其对应的最低的线段。之后可以使用 Gabow[3]中的线性集合并算法计算，在线性时间内得出点 q 的可见多边形。对于我们设计的应用，算法可在最后一步轻松的求解某个 x 区间内的可见多边形，也就是守卫在一定角度范围内的可见区域。



进度安排：

如下表所示，是我们小组计划完成整个项目的时间进度安排：

开始时间	结束时间	工作任务
2015.5.11	2015.5.17	完成数据结构以及算法接口的设计，包括： 1. 表示节点偏序关系的图，使用邻接链表表示 2. 顶点、向量，线段，环，三角形，三角网格，多边形的数据表示 3. DCEL 数据结构 完成基于 MFC 与 OpenGL 的 C++程序框架的初始搭建
2015.5.18	2015.5.24	1. 完成根据带孔多边形生成受约束 Delaunay 三角剖分以及相应的更新 2. 完成根据边的远近排序结果以及顶点的极角序生成最后的线性集合 3. 多条直线相交生成 DCEL 结构 4. 实现 OpenGL 进行绘图操作
2015.5.25	2015.5.31	1. 完成有向图的拓扑排序

		2. 完成根据视线范围对线性集合求出最终的可见多边形 3. 完成判断点是否在可见多边形内 4. 插入新的直线更新 DCEL 结构，并输出所有交点，表示出所有顶点的极角序 5. 实现基本的游戏逻辑控制
2015.6.1	2015.6.7	1. 完成逻辑功能的整合和调试 2. 实现一个完整的游戏 demo，并调优
2015.6.8	2015.6.14	机动时间，用于 bug 处理，界面完善，性能优化等

组内分工：

如下表所示，是我们小组对任务的一个基本分工：

组员	数据结构部分	算法部分	平台框架部分
罗必成	表示节点偏序关系的图，使用邻接链表表示	1．根据带孔多边形生成受约束 Delaunay 三角剖分 2．并对新插入的顶点做三角剖分的更新 3．有向图的拓扑排序	基于 MFC 与 OpenGL 的 C++程序框架的初始搭建
温佺	顶点、向量，线段，环，三角形，三角网格，多边形的数据表示	1. 根据边的远近排序结果以及顶点的极角序生成最后的线性集合 2. 根据视线范围对线性集合求出最终的可见多边形 3. 判断点是否在可见多边形内	使用 OpenGL 进行绘图操作，游戏的逻辑控制
陈翔	DCEL 数据结构	1. 多条直线相交生成 DCEL 结构 2. 插入新的直线更新 DCEL 结构，并输出所有交点，表示出所有顶点的极角序 3. 研究算法的改进和优化	界面部分的体验优化

参考文献：

[1] Asano T, Asano T, Guibas L, et al. Visibility of disjoint polygons[J]. Algorithmica, 1986, 1(1-

4): 49-63.

[2] Chazelle B, Guibas L J, Lee D T. The power of geometric duality[J]. BIT Numerical Mathematics, 1985, 25(1): 76-90.

[3] Gabow H N, Tarjan R E. A linear-time algorithm for a special case of disjoint set union[C]//Proceedings of the fifteenth annual ACM symposium on Theory of computing. ACM, 1983: 246-251.