

# 计算几何第三次大实验：梯形图

## 一. 问题介绍。

点定位问题是指：在一个平面区域的划分中，找到一个点属于哪个区域。梯形图结构是解决点定位问题的一种高效算法，点查询的平均时间复杂度为  $O(\lg n)$ ，而梯形图本身的构造时间复杂度为  $O(n \lg n)$ 。

## 二. 梯形图构造算法介绍。

输入为平面上的一组互不相交的线段集，输出为线段集对应的梯形图和查找树结构。

算法的整体思路是递推，每一步递推都有对应的初始梯形图和查找树结构，考虑当前的线段，对当前的梯形图存储结构和查找树进行更新，具体更新算法见参考资料 1 中的详细介绍。

为了获得  $O(n \lg n)$  的平均时间性能，采用了随机增量式的算法结构，随机地获得下一个输入线段。

## 三. 程序界面和使用说明

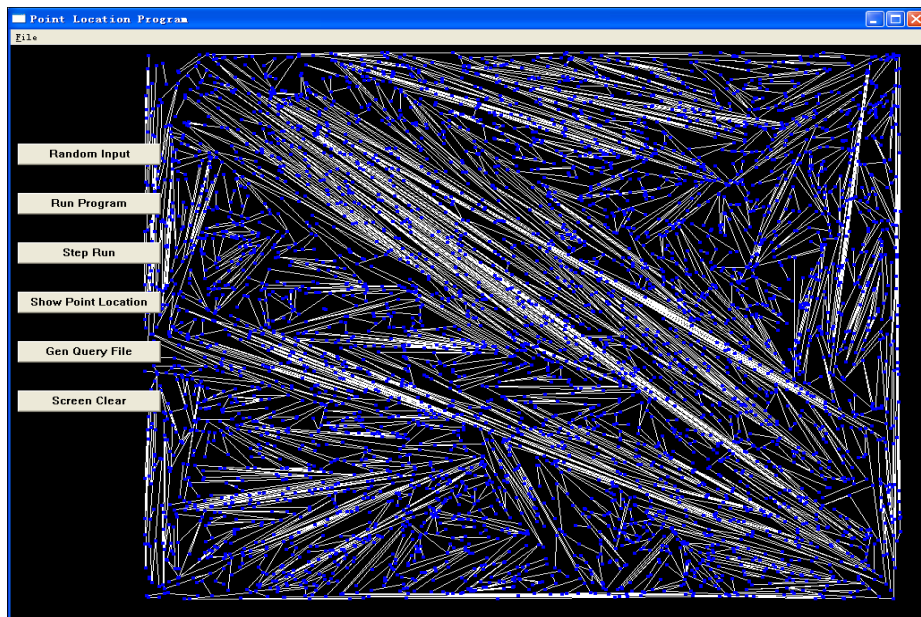
### 1. 线段集的输入：

分为三种方式，鼠标输入，文件输入，随机生成。

鼠标输入时，点击鼠标左键开始输入，点击左键结束一条线段的输入，这时程序会以这条线段的终点为新的起点再次画线段，若要结束这种输入，点击鼠标右键即可；再次点击左键可以开始新的输入。

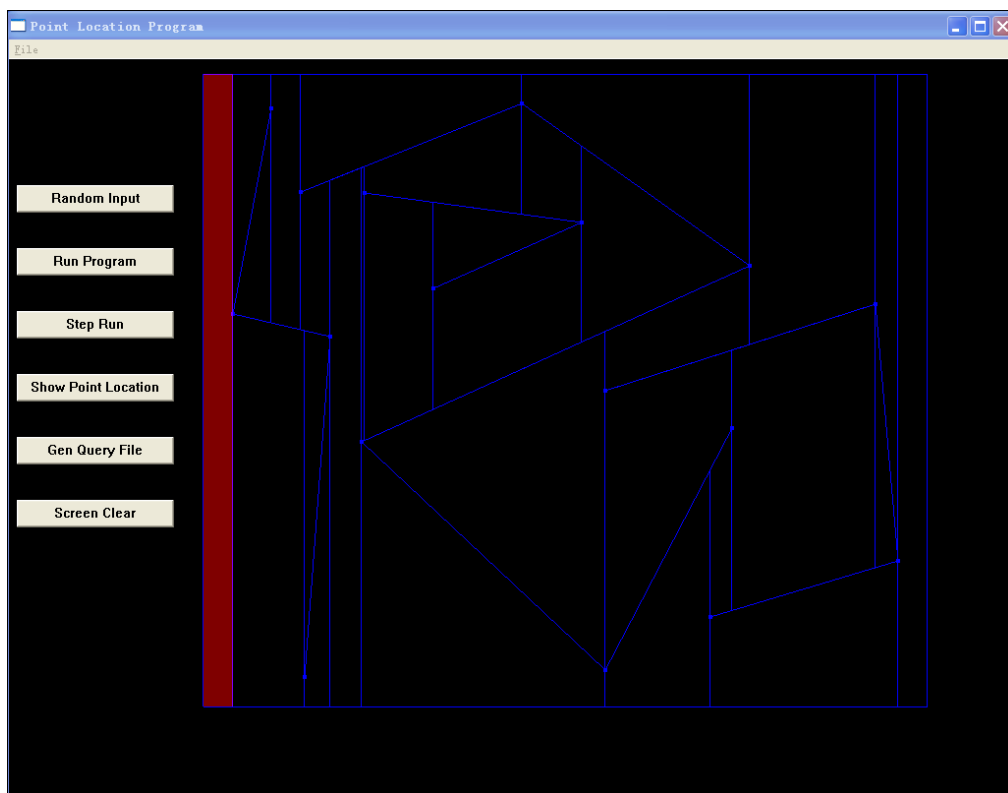
文件输入时，点击 File 菜单栏的 OpenSegmentFile 选项，选择要导入的线段文件，线段端点坐标系的原点取的是屏幕的左下角，X 轴正向向右，Y 轴正向向上。

随机生成时，点击屏幕左边的“Random Input”按钮，输入线段数目，获得随机线段。因为本梯形图算法不能处理线段相交的情况，所以要保证线段集无交，本算法采用了逐个比较，发现相交则重新生成的方法，效率较低，这一点生成 2000 个以上的线段时会明显感到，下图为生成 3000 条随机不相交的线段的结果。

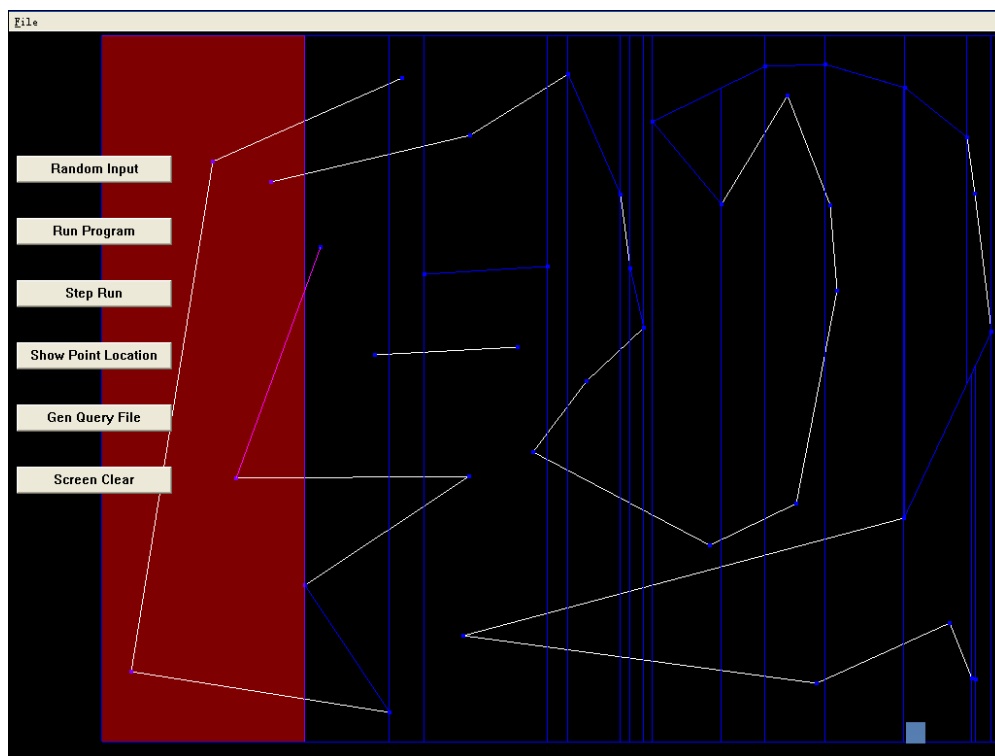


## 2. 程序的运行。

提供了两种运行方式，一种是直接运行，一种是 Step 运行，Step 运行需要点击“Step Run”按钮，下一个要处理的线段被标记为红色，按下“Step Run”则完成梯形图的一步更新。下面为直接运行和分步运行的截图。



直接运行的结果



分步运行中的一步，品红的线段为下一条线段，红色区域是当前点定位的结果

### 3. 点定位的显示。

提供了两种点定位的显示，一种是鼠标移动时，即时标记出鼠标点所在的梯形为红色，另一种是单击鼠标，输入查询点，标记出从这一点垂直向上相交的第一条线段，这一步给出了从查找树中查询的可视过程，在查找树中进行查找时，若遇到 X 节点，则绘出坐标为 X 的竖线，若遇到 Y 节点，则显示出相应的线段；最后标记出找到的线段。

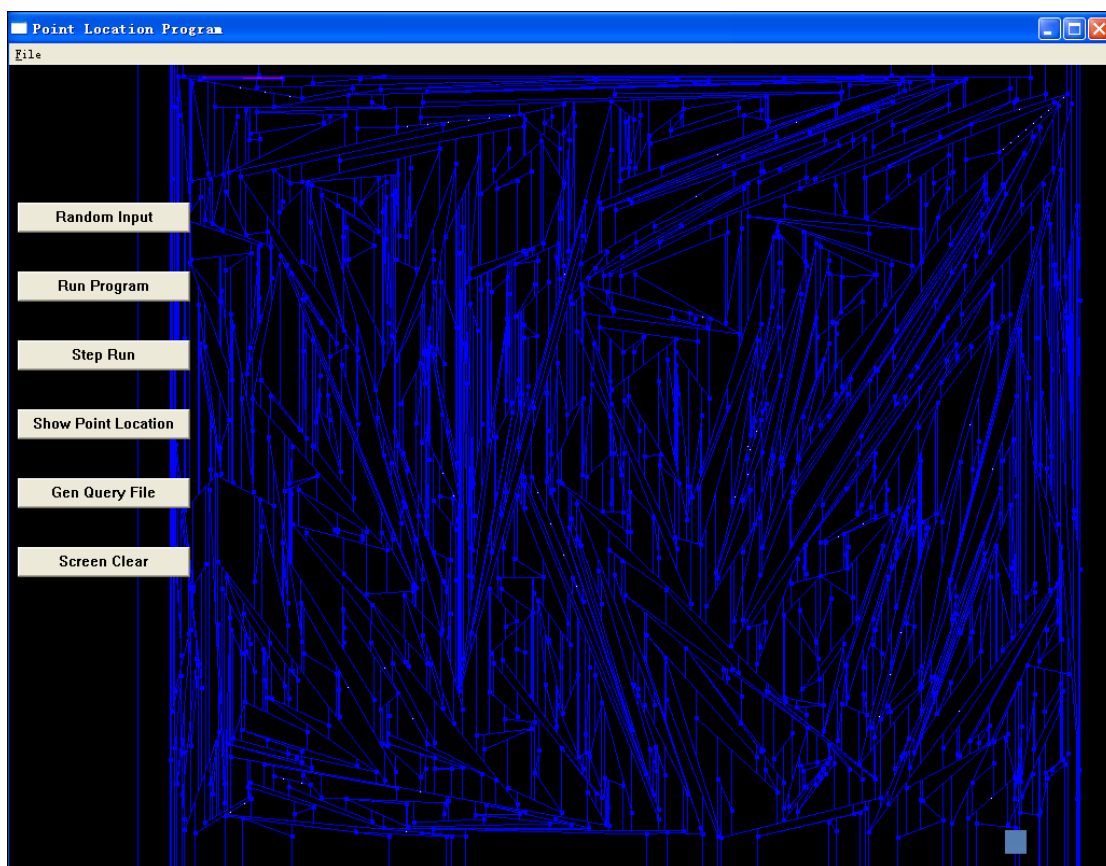
按钮“Show Point Location”，提供了两种显示方式的切换。

### 4. 清空屏幕。

如果需要重新输入，需要先按下“Screen Clear”按钮。

### 5. 批量查询和时间统计。

导入查询文件：点击 File 菜单的 OpenPointFile 菜单项，即导入查询点数据，并开始查询，查询结束后给出查询时间统计（不包括 I/O 时间）。



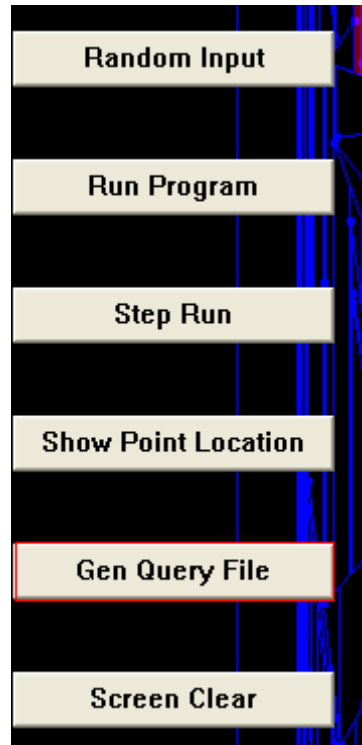
梯形图计算结果



对上面梯形图查找 10000 个点的执行结果统计

#### 6. 生成批量查询点文件。

为了方便程序的调试和性能分析，程序提供了生成查询点文件的功能，点击“Gen Query File”，在弹出对话框中输入生成点的数目和文件名称，就可以生成查询点文件。



#### 四. 性能分析和程序鲁棒性。

不同的输入顺序会导致查询树的高度不同，从理论上说，高度越高的树查询性能越低，而趋于平衡的树平均查找效率最好。

程序在处理超过 1000 个线段的输入时，鲁棒性不够理想，往往会产生 Heap Corruption。希望以后可以有时间完善它。

#### 五. 参考资料。

1. 《计算几何—算法与应用》第 6 章。