

平面点集三角剖分实验

一、实验目的

1. 基于分治策略，实现对平面点集的三角剖分；
2. 利用 DCEL 结构对剖分后的图进行存储，以便进行后续操作；
3. 熟悉简单的 MFC 编程。

二、实验分析

1. 算法分析

我们的算法是采用分治策略的，即根据点的 x 坐标将点分成左右两个部分，分别将左右两个点集进行三角剖分，再通过 zig-zag 的方法合并左右两个部分。详细请参考后面算法描述部分。

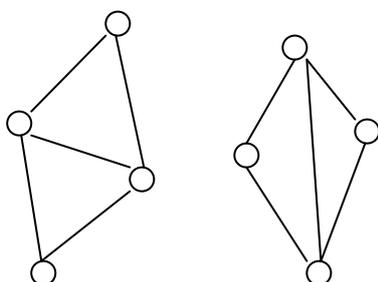
2. 图的存储

为了使得剖分的结果能执行后续操作，我们需要使用 DCEL 结构来存储当前的图。在我们的实验中采用了翻墙实验来验证剖分结果的正确性，具体可以参考实验说明部分。

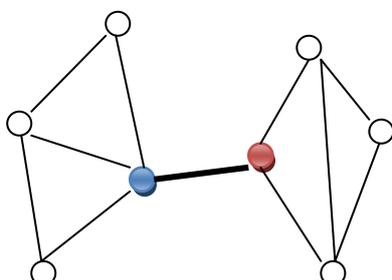
三、算法描述

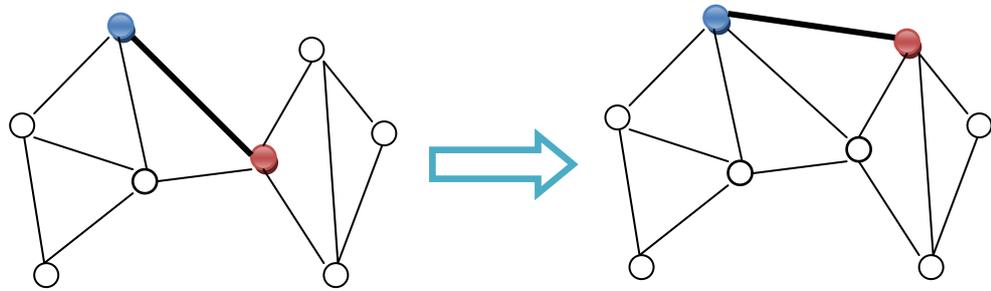
1. 核心思想

我们采用分治的算法，对平面点集进行处理，如下图所示，



假设左右两边已经处理完毕，那么可以通过 zig-zag 的方法求出结果。具体做法是先找到最右和最左点。





左边多边形依次向上找，知道某个点满足将左边所有点划分到同一侧，这样再接着右边的点，如此往复下去，最后可以完成整个 zig-zag 过程。
下面的部分同理可以完成。

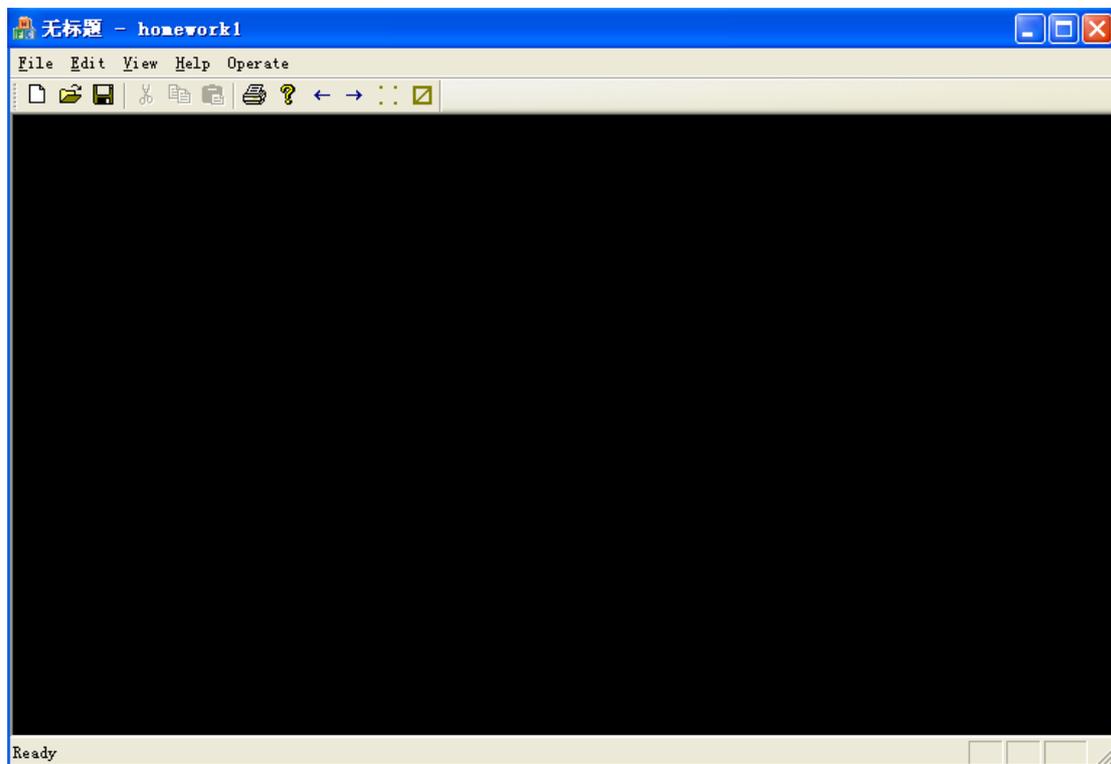
2. 退化情况分析

对于可能出现的同一 X 可能坐标出现多次的情况，我们可以通过将坐标轴旋转一个微小的角度来解决。当然在实际处理中，不必真的作旋转，因为那样会产生精度误差，我们只需要假想旋转即可。例如，在排序的时候，当 X 坐标相同时，我们比较 Y 坐标，将 Y 坐标从小到大排序，实际上与旋转后再比较是等价的。

3. 优化

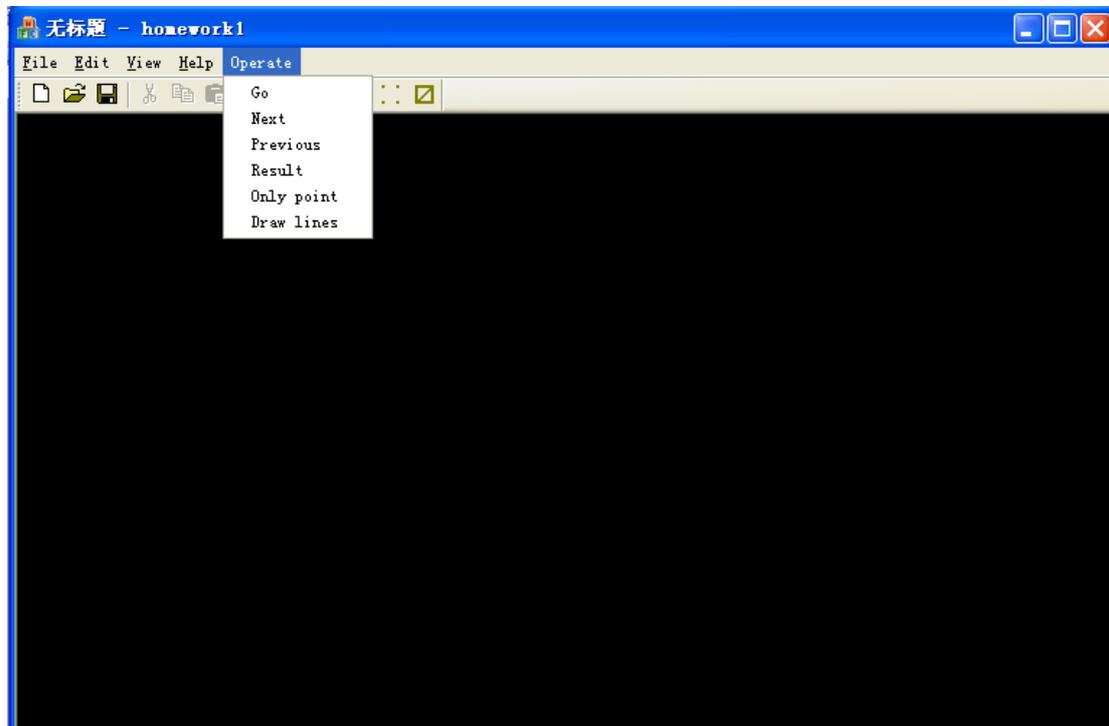
在找最左边点和最右边点的时候，我们不必每次都去寻找，而是对每个多边形纪录它的最左和最右点即可，每次 zig-zag 合并完后可以通过 $O(1)$ 的时间就更新这对左右点。这样就不必要每次都去查找一遍了。

四、 软件介绍



打开后可以直接在黑幕上点点，或者通过 File 菜单中的 Open 按钮打开文件读取数据

Operate 菜单



点击 Go 后可获得所设置点集的三角剖分

Next 与 Previous 都是实现单步操作的，同时他们对应<-和->按钮

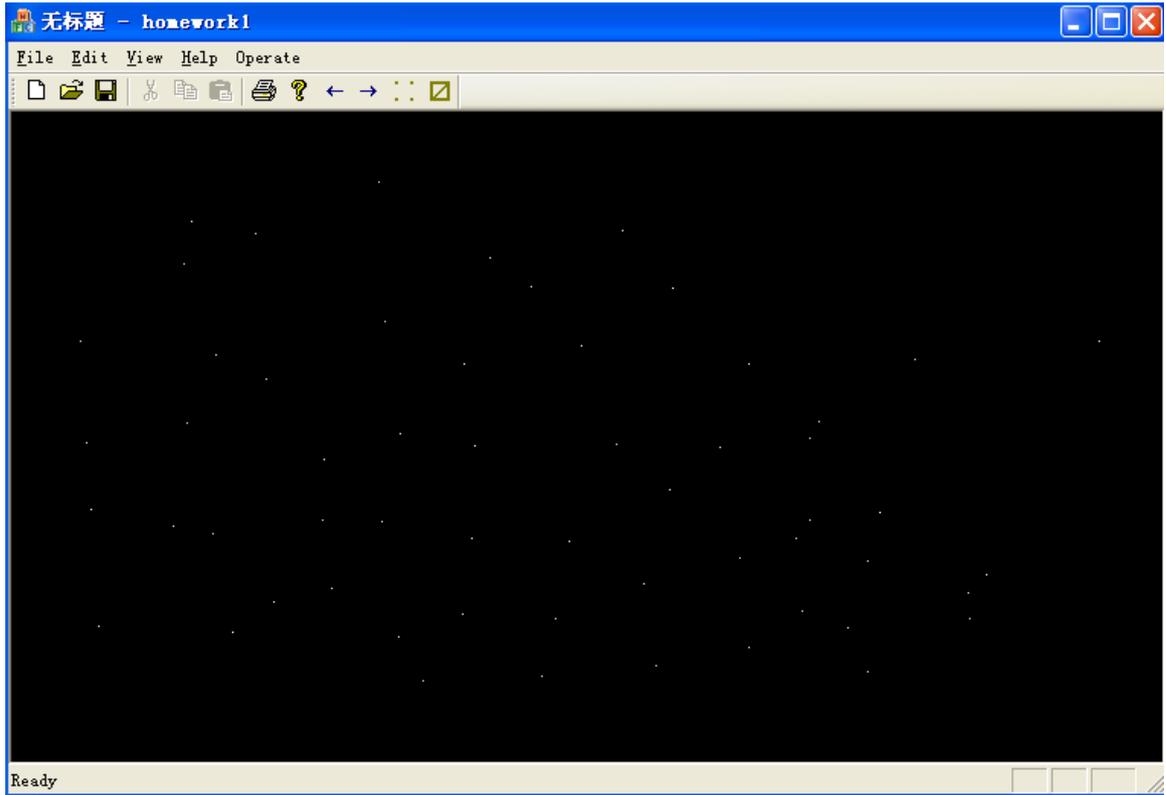
Result 实现在单步过程中，可直接得到三角剖分的，同时对应着  按钮

Only point 实现了实验中获取原点集的功能，同时对应着  按钮，它是三角剖分单步执行的第一步

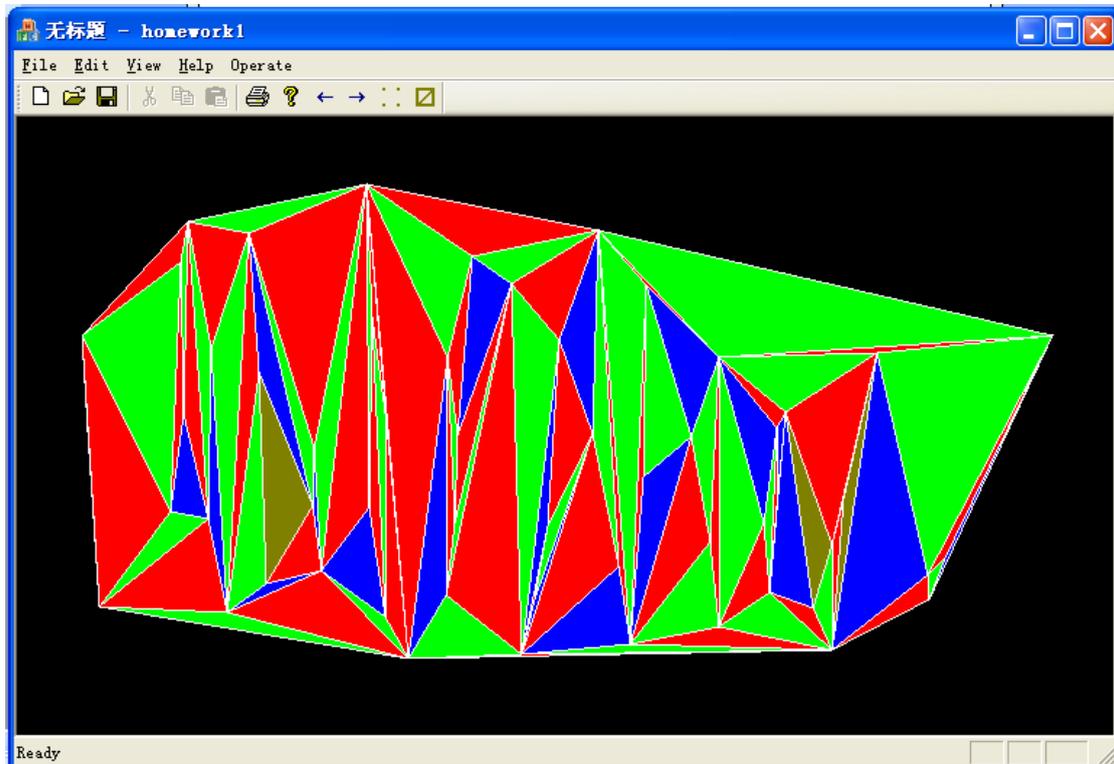
Draw lines 实现了 DCEL 结构验证性实验中打开和关闭画直线的功能，如果不关闭，其他三角剖分的功能不能实现

五、 实验结果

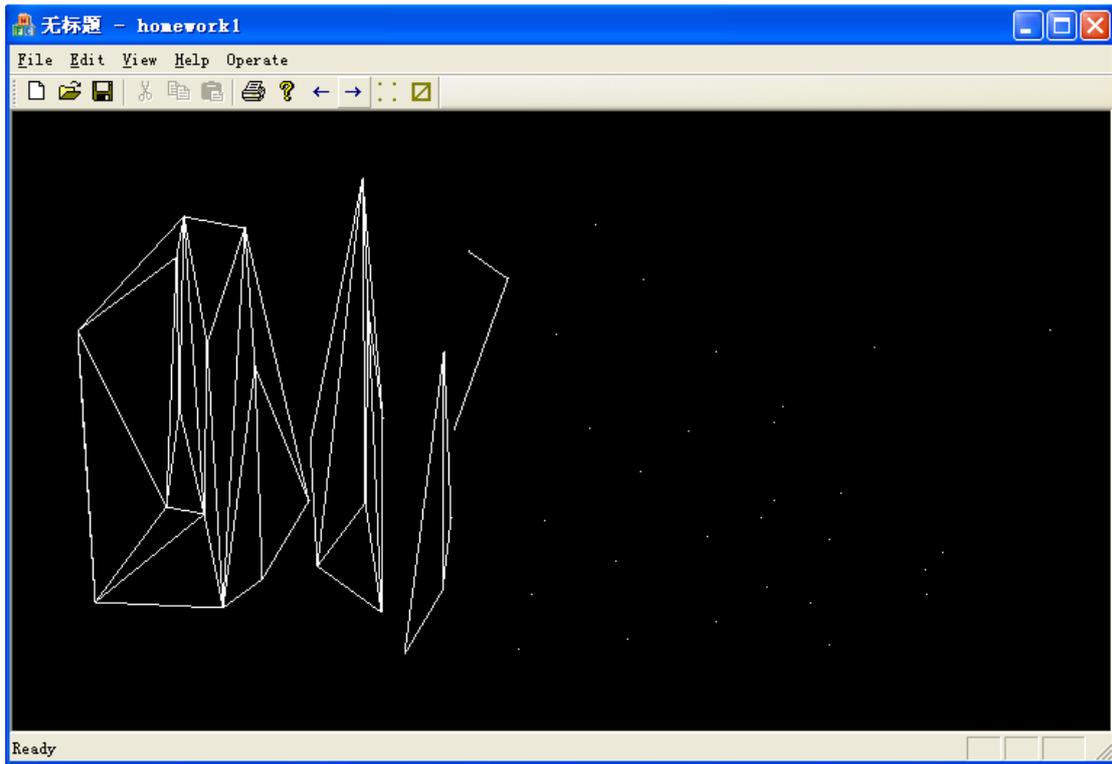
1.三角剖分



对上面点集，在 Operate 菜单中选择 Go，即可得出剖分并着色的三角剖分图如下所示

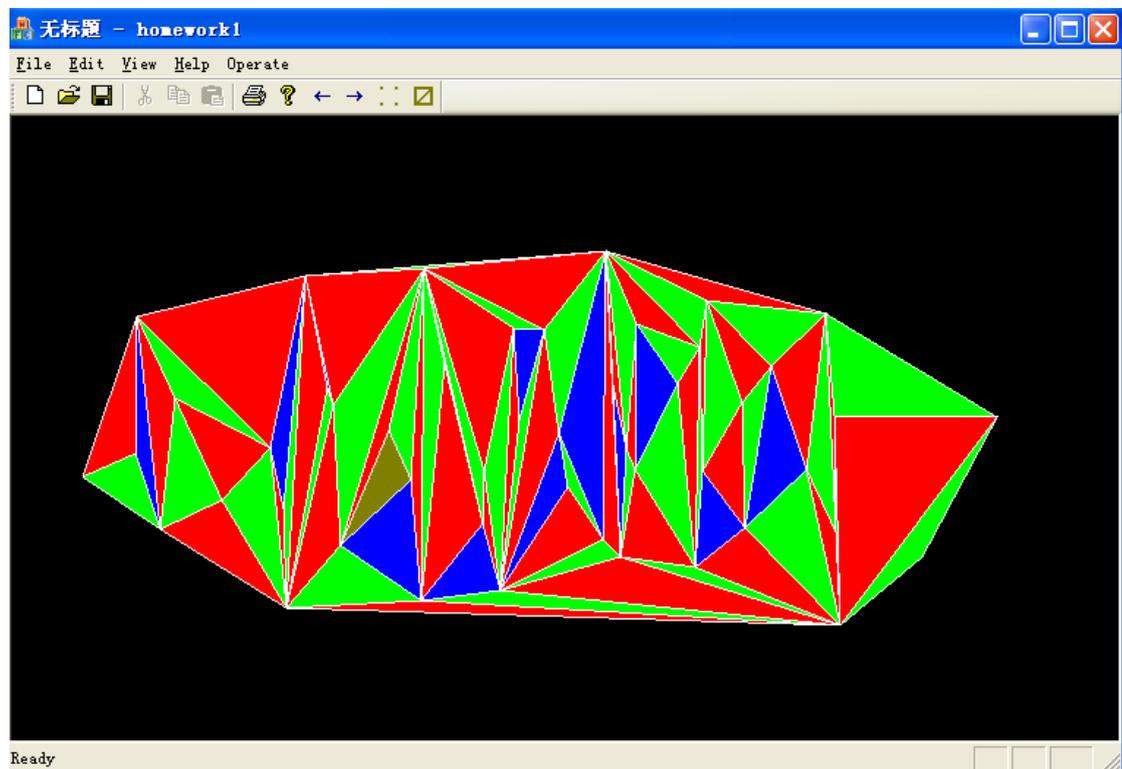


然后选择 Operate 菜单中的 Only point 得到点后就可以单步执行三角剖分过程

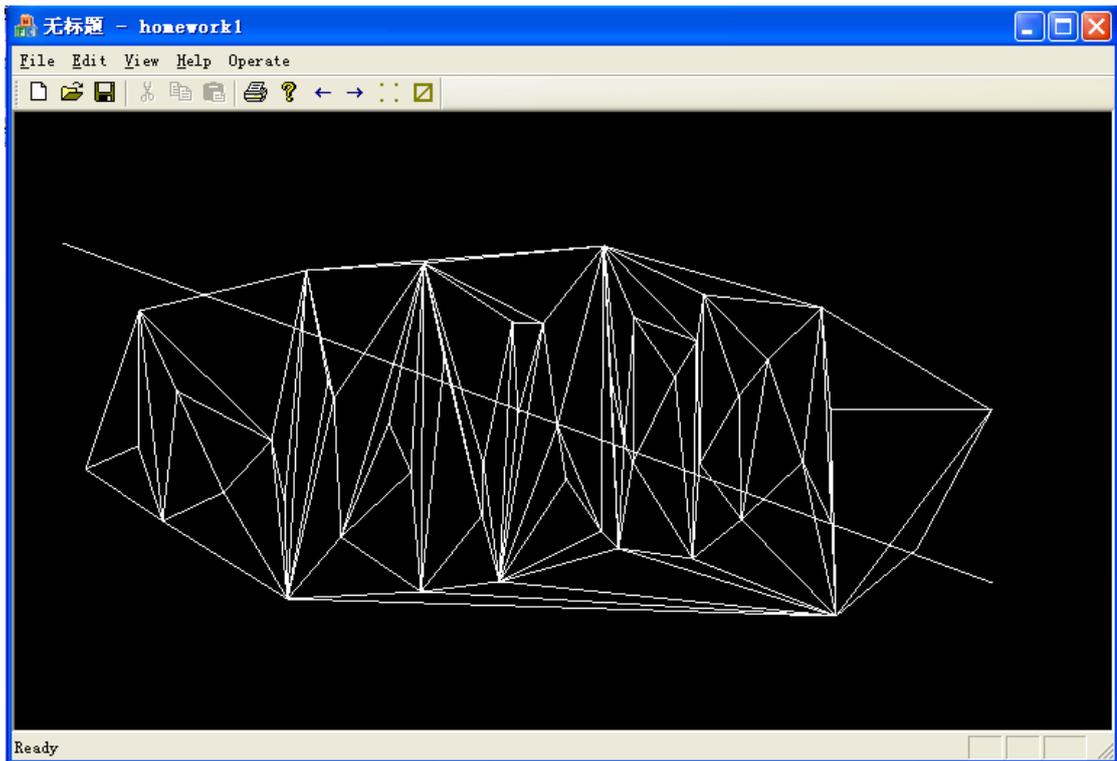


2.翻墙实验

对一个已经三角剖分过的平面点集，如下

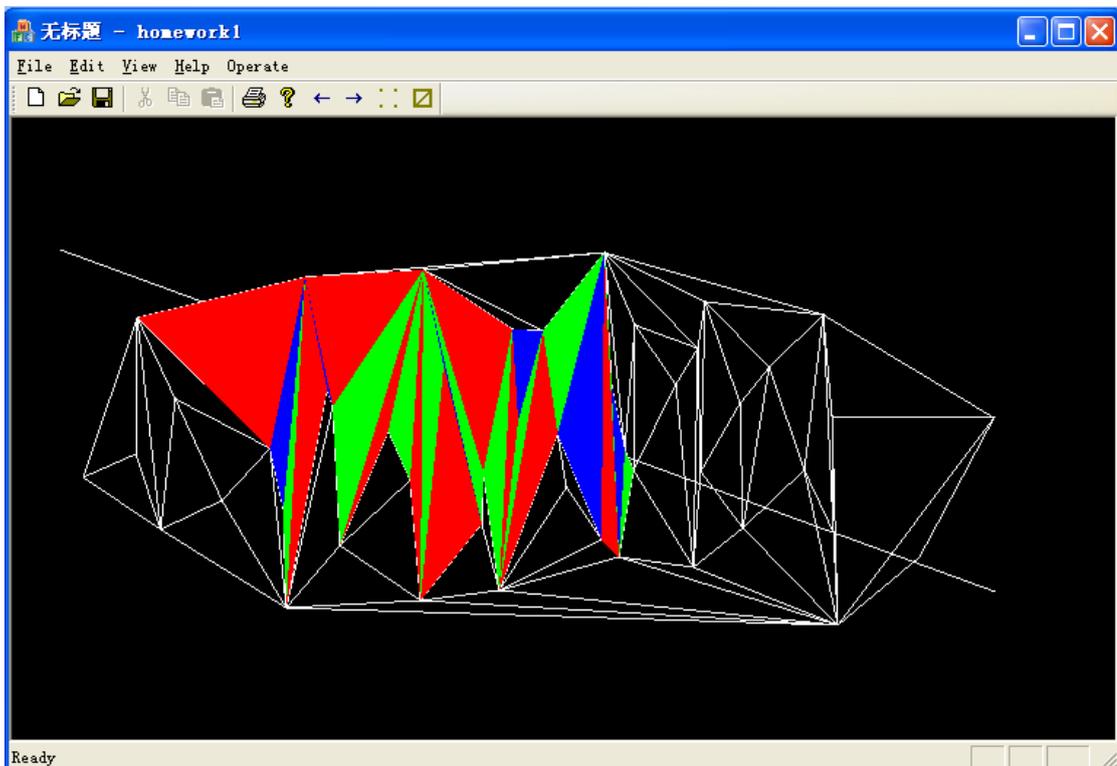


在 operate 菜单中选择 Draw lines，然后可以在图中随便画一条直线，如下图

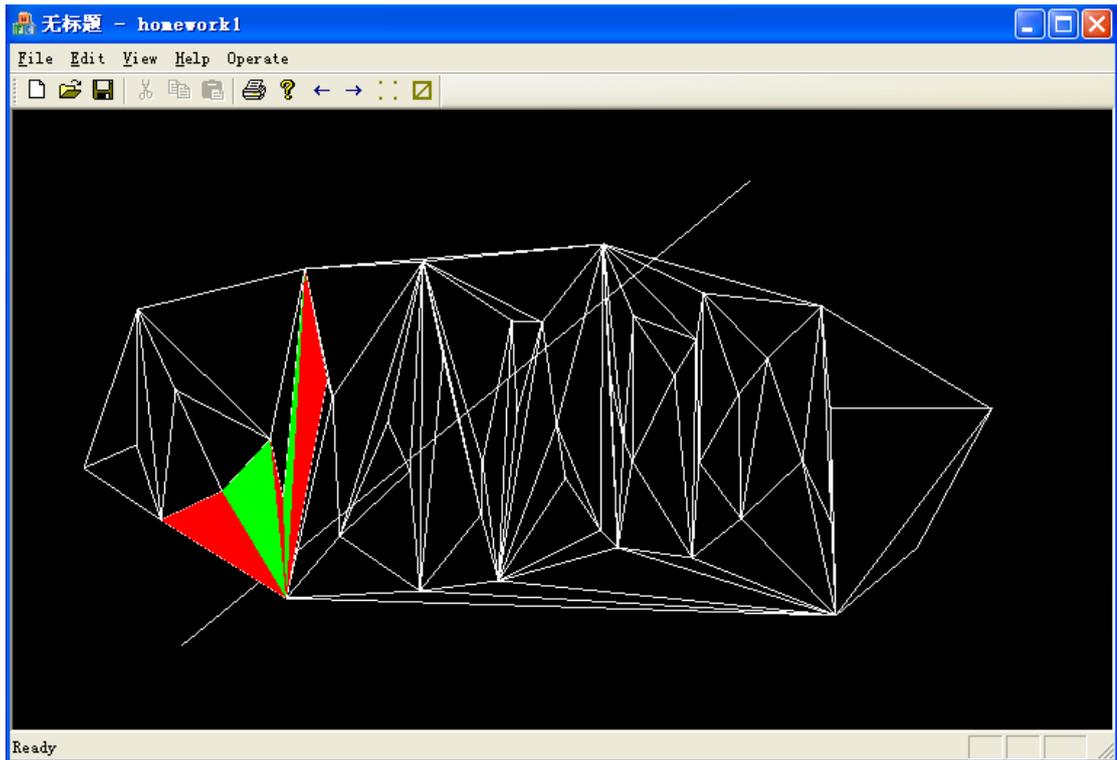


注意线的两端必须在多边形外部。

然后执行单步执行操作，就可以一个一个的将直线经过的三角形涂色



在这个过程中我们还可以再随画一条直线，并单步执行对此条直线所过三角形涂色

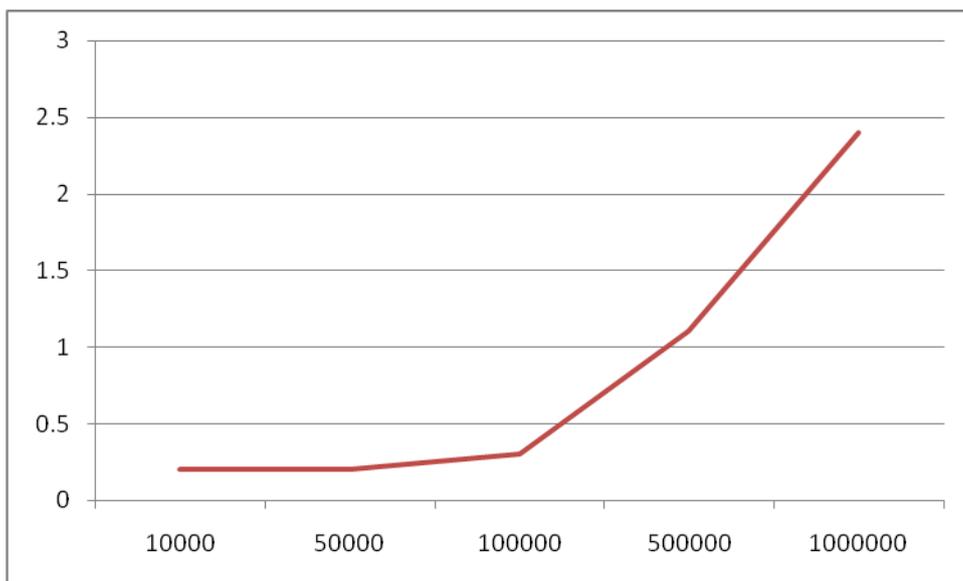


如此证明利用 DCEL 结构储存可以进行各种后续操作。

3. 时间分析

根据点数的不同，我们测试了 10000-1000000 的不同情况，程序用时如下：

点数	10000	50000	100000	500000	1000000
用时(秒)	0.2	0.2	0.3	1.1	2.4



六、实验收获

经过此次实验我们对平面点集的三角剖分有了清楚的了解，并且首次使用了 DCEL 结构进行图的存储。不仅如此，实验还使我们对所学内容有了更深刻的理解，更使我们对以后的所学内容有了大致性的概念，最后感谢两位老师对我们实验的建议，使我们对整个实验的原理掌握得更充分。