面向对象程序设计个人大作业

Euclidean Minimal Spanning Tree设计文档

计34 何钦尧 2012010548

## 整体设计

整体采用了Strategy的设计模式，主要是为了在朴素方法和Delaunay三角剖分方法之间进行切换。定义了一个EMSTAbstract类，用于规定求解生成树的通用接口。EMSTPrim和EMSTDelaunay类分别继承EMSTAbstract并实现。其中定义一个solve函数，接受一个点的vector作为参数，返回的是点对的vector（即vector<pair<Point, Point>），点对即表示最终求出的生成树中的一条边。

使用EMSTSolver类接受EMSTAbstract的一个实例，用于计算。EMSTVisualizer类用于接受一个点对的数组，并把它显示出来。

最小生成树的求解使用了Prim算法。图的表示使用邻接表的方法。这是考虑Prim算法需要遍历每个点的边，用邻接表能有更好地性能。如果使用Kruscal算法求最小生成树的话，应该使用边列表。写了一个Graph类对类的相关操作进行了封装。

## 技术实现

项目的组织管理使用cmake工具。Cmake的优点是跨平台，而且可以生成多种格式的项目文件，便于使用各种环境进行开发。图形的可视化采用opencv。项目同时依赖boost，opencv和cgal库。

两种方法（Delaunay和Prim）分别作了不同的计算。Prim中，对每对点之间都连一条边，建立一个完全图然后求最小生成树。Delaunay中，先用CGAL求出三角剖分，然后在图中添加三角剖分计算的边，再在这个图上求最小生成树。

## 使用方法

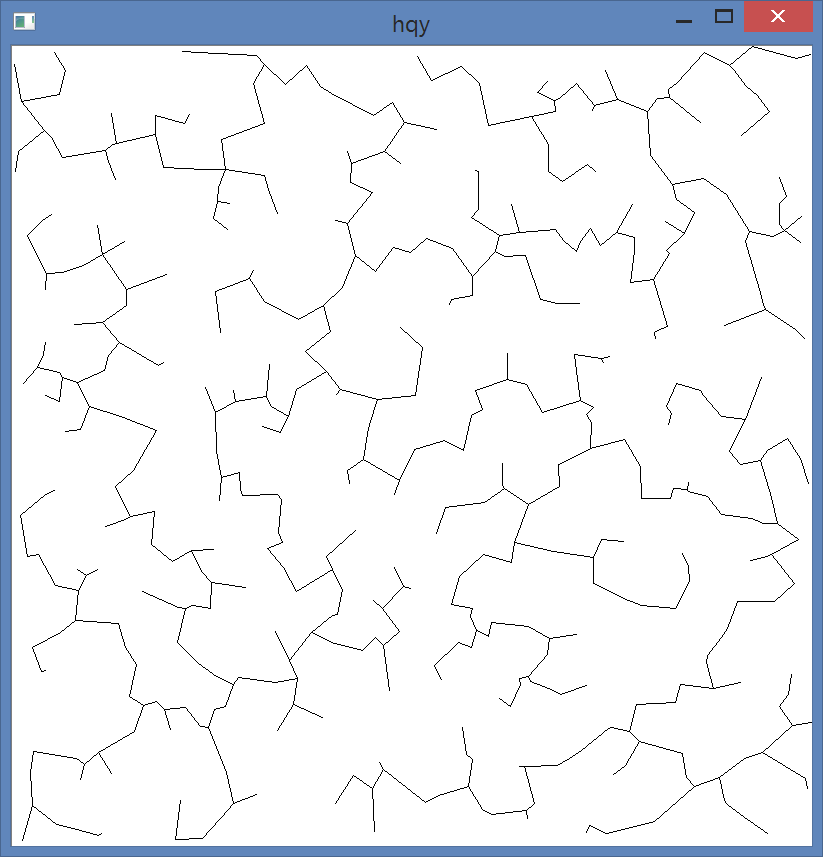
Cmake项目中添加了两个可执行文件。一个是EMST，一个是EMST\_test，顾名思义。项目主要代码在src文件夹中，test文件夹中为测试程序的代码。Testcase中有预先生成的测试数据文件。EMST主程序，不加参数的运行，会随机生成500个点，然后计算欧几里得最小生成树，默认使用Delaunay剖分的办法。可带一个参数，表示输入的数据文件。

数据文件的格式为，第一行一个整数，表示点的个数。接下来每行一个点，两个数字分别表示x和y坐标。

Test程序主要做单元测试用。不带参数运行会使用testcase下面的input1.txt到input5.txt文件，分别运行Prim和Delaunay算法，并对其结果进行比较。这里面我比较的方面有，所有的边是否相同，最后生成的树的总距离是否相同。测试结果会在输出中加以显示。可以带参数运行，格式为“generator n filename”用于指定生成新的测试数据，n代表多少个点，filename代表输出的文件。

为了保证运行，请将运行目录设置在bin下面（程序中用了比较笨的办法把路径写死了）。

## 测试结果



图为500个点的运行结果。

在测试程序中，有时会出现failed的情况。但是总的cost又是一致的。经过我的检查，发现两种算法的求解几乎一样，只在个别地方出现了不同的边的连接，但是两种连法的长度都是一样的。所以可能他们分别选择了不同的方案。在我的测试中，随机生成的数据点里面，这种情况还比较常见。