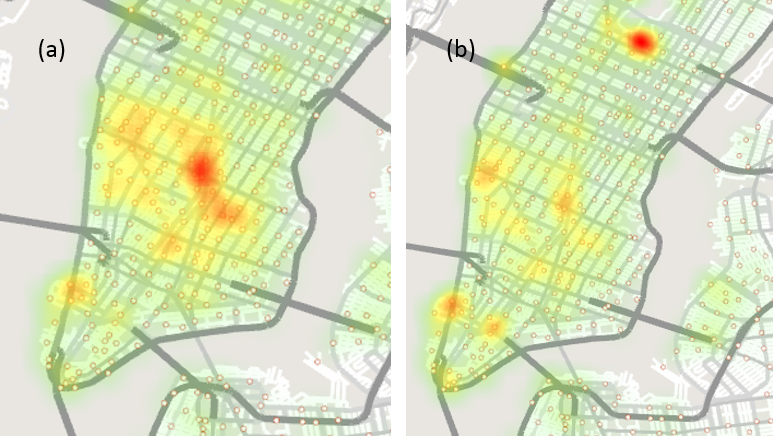
实习7 空间索引编程

**实习目的：**了解空间数据类型层次结构，熟悉包围盒Envelope在空间查询中作用，掌握四叉树的创建，掌握区域查询和最邻近查询方法，了解空间数据挖掘基本概念，掌握K-Means聚类方法。

**实习内容：**

纽约市的自行车租赁点和出租车上车，如下图为自行车租赁点在休息日租车数量对应的热力图。



(a) 个体休息日租车热力图 (b) 群体(>2人)休息日租车热力图

数据在data目录下，程序代码包含以下文件：

|  |  |
| --- | --- |
| 文件名 | 说明 |
| Shapelib | Shapefile文件读取类库，包含shapefil.h, shpopen.cpp, safileio.cpp和dbfopen.cpp，无需修改任何代码。 |
| Common.h | 符号定义，无需修改任何代码。 |
| Geometry.h/cpp | Envelope, Geometry, Point, LineString, Polygon类，需要实现距离计算和Envelope空间关系代码 |
| QuadTree.h/cpp | Feature类封装Geometry类，QuadNode四叉树节点类，QuadTree四叉树类，需要实现QuadNode和QuadTree的创建和查询代码 |
| KMeans.h/cpp | Cluster和KMeans类，需要实现KMeans算法 |
| Lab7.cpp | 界面和数据读入等函数，无需修改任何代码 |
| Test.cpp | 测试代码，需要实现四叉树性能分析 |

建议先浏览每个类的功能和已有的函数，再填充相应函数的C++代码。实习总分60分，其中1-7是必做题，8-16是选做题，选做题最多加20分。实习可以2人一组，每人得分是实习得分×0.8，即需要做75分，能拿60分，做100分，能拿到实习最高80分。

1. 实现点与点之间的球面距离计算(Geometry.cpp，4分)

double Point::distanceOnSphere(const Point\* point) const

球面距离计算公式参考http://tech.meituan.com/lucene-distance.html

2. 包围盒空间关系判断(Geometry.cpp，10分)

bool Envelope::contain(const Envelope& envelope) const

bool Envelope::intersect(const Envelope& envelope) const

Envelope Envelope::unionEnvelope(const Envelope& envelope) const

空间索引是基于几何特征的包围盒创建，首先通过判断几何特征的包围盒是否和查询区域相交，降低复杂的几何特征空间关系计算的次数。contain含义和PostGIS中的contain不同，可以是相同的包围盒。

3. 四叉树构建(QuadTree.cpp，10分)

bool QuadTree::constructQuadTree(vector<Feature>& features)

void QuadNode::split(size\_t capacity)

四叉树创建输入一组几何特征，将节点分裂为四个子节点，每个特征加到包围盒重叠的子节点中（即一个特征可能在多个节点中），删除当前节点的几何特征记录（即所有特征只存储在叶节点中），如果子节点的几何特征个数大于capacity，递归分裂子节点。

4. 基于四叉树的区域查询(QuadTree.cpp，10分)

void QuadNode::rangeQuery(Envelope& rect, vector<Feature>& features)

void QuadNode::rangeQuery(Envelope& rect, vector<Feature>& features)

区域查询输入区域rect，查询与区域rect相交的几何特征，存储在features。区域rect如果与当前节点的包围盒bbox相交，递归遍历四叉树，查询哪些几何特征的包围盒和查询区域相交；再获得可能和区域rect相交的候选几何特征后，精确判断几何特征是否与区域rect相交。由于实习使用点数据，因此精确判断可以忽略，但对于线和面，此步不能忽略。

通过鼠标选择查询区域，验证区域查询。

5. 基于四叉树的最邻近几何特征查询(QuadTree.cpp，10分)

bool QuadTree::NNQuery(double x, double y, Feature& feature)

QuadNode\* QuadNode::pointInLeafNode(double x, double y)

最邻近几何特征查询(K-NN)输入查询点(x, y)，返回与该点最邻近的几何特征，存储在feature。首先，通过pointInLeafNode查询点(x, y)所在的叶节点，计算查询点(x, y)与该叶节点内的几何特征的最小距离minDist，然后，构造查询区域Envelope(x – minDist, x + minDist, y – minDist, y + minDist)，查询与查询点(x, y)距离最近的几何特征。

通过鼠标移动选择离鼠标最近的几何特征，验证最邻近几何特征查询。

6. 基于空间位置聚类(KMeans.cpp，10分)

void KMeans::cluster(vector<Feature>& features, int k, int maxIterNum)

bool Cluster::resetMean()

聚类输入几何特征集合，聚类个数和最大迭代次数，其中resetMean根据Cluster中的几何特征集合，计算平均空间位置。KMeans算法参考http://baike.baidu.com/view/3066906.htm

7. 四叉树性能分析(test.cpp，6分)

void QuadTreeAnalysis(vector<Feature>& features)

使用纽约taxi上车点数据分析四叉树性能。四叉树的性能决定于参数capacity，即每个叶节点最多的几何特征数目，数值太多，每个叶节点判断次数太大，数值太小，树的层次太高。当capacity在[70, 100]时，计算四叉树的高度和叶节点数目，评估1000次随机最邻近几何特征查询的时间。

**选做题**

8. 实现Point-Line，Point-Polygon，Line-Line，Line-Polygon，Polygon-Polygon的欧式距离计算(Geometry.cpp，10分，每个欧式距离计算2分)

9. 实现Polygon的内环几何数据存储，并修改欧式距离计算代码(Geometry.cpp, 4分)

10. 扩展几何类型，增加MultiPoint, MultiLineString, 和MultiPolygon类的数据存储和欧式距离计算(Geometry.cpp，12分，每个类4分)。

11. 扩展四叉树，实现spatial joins的查询函数，即道路和打车位置关联分析，计算每条道路上的打车数量(10分)。

12. 基于Voronoi diagram计算自行车租赁点的服务空间区域(10分)。

13. 基于Taxi数据绘制打车位置热力图(10分)。

14. DEBS 2015挑战赛Query 1: Frequent Routes (find the top 10 most frequent routes during the last 30 minutes)，http://www.debs2015.org/call-grand-challenge.html，只需分析2015年一个月的数据，约2G数据(10分)。

15. DEBS 2015挑战赛Query 2: Profitable Areas (identify areas that are currently most profitable for taxi drivers) ，http://www.debs2015.org/call-grand-challenge.html，只需分析2015年一个月的数据，约2G数据(10分)。

16. SIGSPATIAL CUP 2016挑战赛: Using spatial statistics to identify statistically significant clusters or outliers in spatial data，http://sigspatial2016.sigspatial.org/giscup2016/，只需分析2015年一个月的数据，约2G数据(10分)。

**实习提交内容：**

1. 实习代码，删除不必要的文件，如lab7.sdf、data、Debug和Release目录，压缩后在1M以内。

2. 实习报告，第4-6题的运行效果截图，第7题的capacity分析，实现了哪些选做题，相应的截图等内容。