计算几何报告

Enclosing k points in the smallest axis parallel rectangle

单　　位：　计算机系软件所

指 导 教 师：　　邓 俊 辉

作　　者：　童锡鹏（**014899**）

　　杨中　（**014887**）

　　尹一瓠（**014960**）

摘要：

*平面上任意 **n** 个点，给定常数 **k**，求边与坐标轴平行最小矩形（按面积、周长或任一给定规则定义，这里采用按面积衡量矩形大小）恰好包含这个 **n** 个点中的 **k** 个。*

问题背景

有许多计算几何学家研究过这一问题，并给出了多种算法，这里简单列举如下：

- Aggarwal 提出时间复杂度为 $O(k^2 n\log n)$空间复杂度为 O(kn)的算法；
- Eppstein and Datta 提出时间复杂度为 $O(n\log n+k^2 n)$空间复杂度为 O(kn)的算法；

进一步深入考察这个问题，可以发现当 k 较大当时候，该问题当时间复杂度有大幅度改善当余地，考虑极端情况：k=n 时，只存在一个矩形能够恰好包含这 k 个点，因此当 k=n 时，在 O(n)当时间复杂度内就可以求得问题的解。而上面当两个算法没有很好的体现这一特点，当 k 较小当时候有比较好的性能，随着 k 的增大，复杂度逐渐提高。我们研究了 Michael Segal and Klara Kedem 提出的算法，这一算法主要集中解决了 k 较大情况下算法时间复杂度较大的问题，使得当 n/2<k<n 时算法具有较好当性能。

另外，这个问题在实际应用中主要用于统计聚类(Statistical Clustering)和模式识别中，关于这两方面的应用请具体参见以下论文：

1．H. C. Andrews, Introduction to Mathematical Techniques in Pattern Recognition, WileyInterscience, New York, 1972.

2．J. A. Hartigan, Clustering Algorithms, John Wiley, New York, 1975.

算法及原理

Michael Segal and Klara Kedem 提出的算法是一个扫描线算法，使用两条扫描线对按 x 排序的这 n 个点进行扫描，第一条扫描线 L1 记录可能矩形点左边界，第二条扫描线 L2 记录可能矩形点右边界。扫描线 L1 初值为 1，L2 初值为 L1 所在位置 r+k，当 L2 扫描到最后一个点时 L1 前进一步，L2 重新设为 r+k，下面详细说明整个算法：

- 基本数据结构：S、L、K1、K2、MinRect、RectInProc

  S： 平面内任意 n 个点；

  L： 左边界 r 起的 k 个最小点（按 x 坐标排序）；

  K1：当前左边界与右边界内所有点中 y 坐标最小点 k 个点；

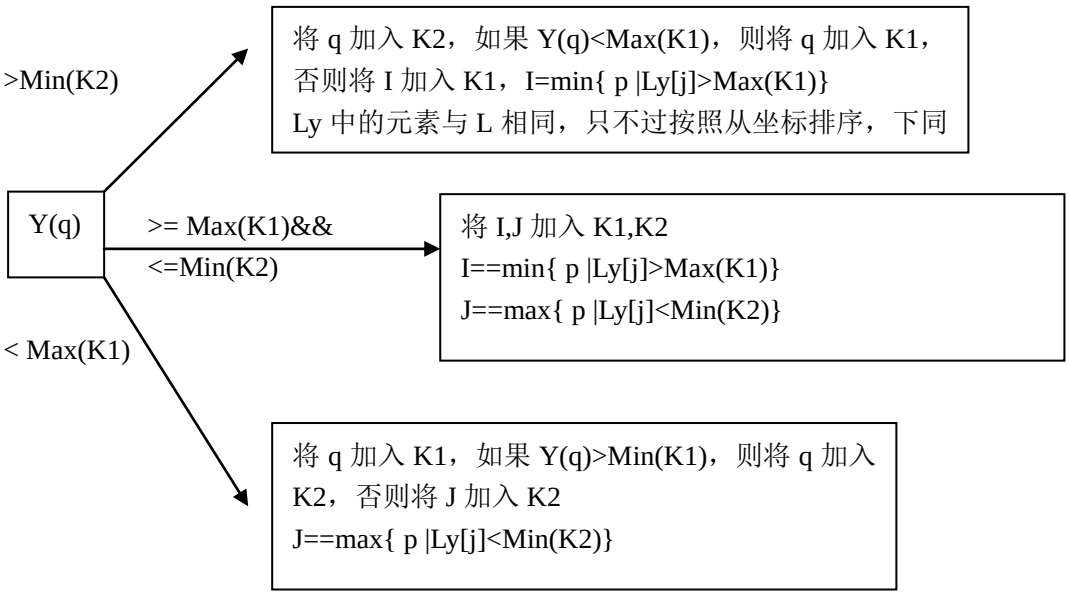
  K2：当前左边界与右边界内所有点中 y 坐标最大点 k 个点；

  MinRect： 已扫描过矩形中按定义最小的一个矩形；

  RectInProc： 扫描线算法每前进一步生成的所有新的矩形中按定义最小的一个矩形；

- 算法流程：

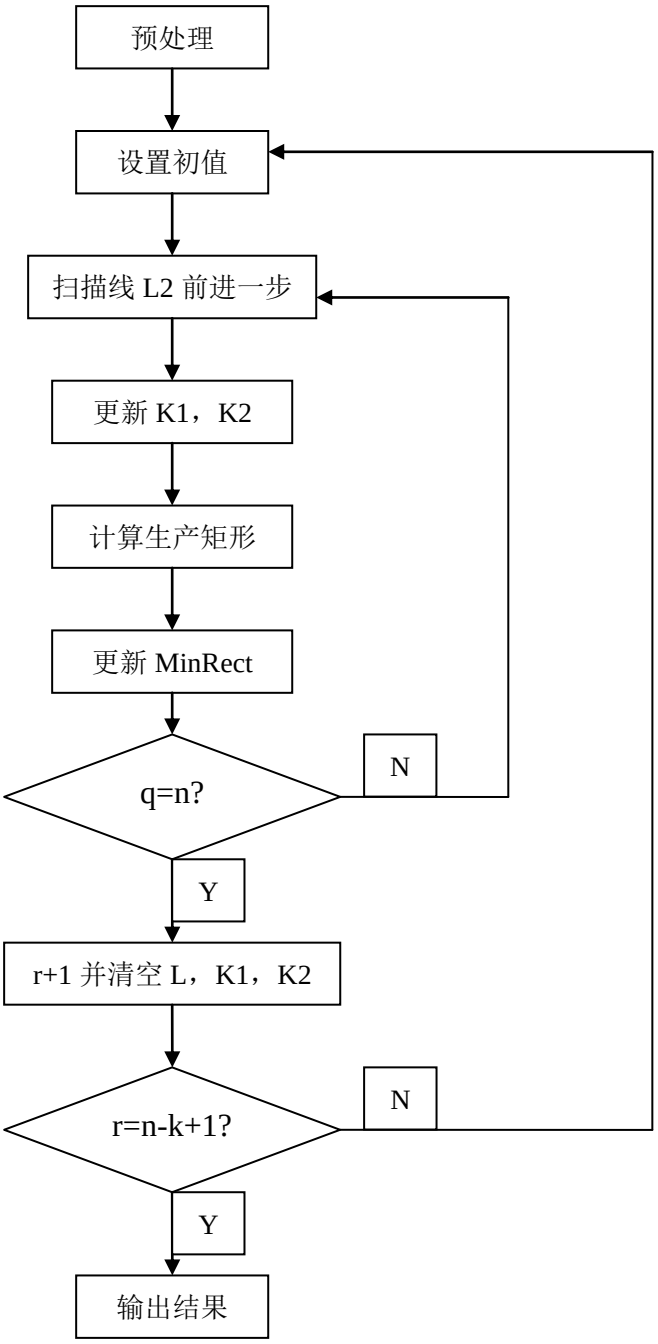
  1. 预处理。将 n 个点按 x 坐标取出存入 S；将 L、K1、K2 清空；扫描线 L1 所在位置 r=1，初始矩形为无穷大；
  2. 对各个元素进行初始化。
     i. 将排好序的第 r,r+1,...,r+k-1 个元素装入 L，将 L 中按纵坐标排序的最小最大点放到 K1，K2 中，扫描线 L2 所在位置设为 q=r+k-1；
     ii. 找到 min(L,x)，max(L,x)，min(K1,y)，max(K2,y)并将之存入 MinRect；
  3. 扫描线 L2 前进一步。
     i. q+1，即新加入一个点 p；
     ii. 将 p 加入 L 并对 K1，K2 进行更新，更新方式如下：



iii. 求出新生成的 q-(r+k-1)+1 个矩形中按定义最小的一个 RectInProc，与 MinRect 比较并将较小的一个存入 MinRect；

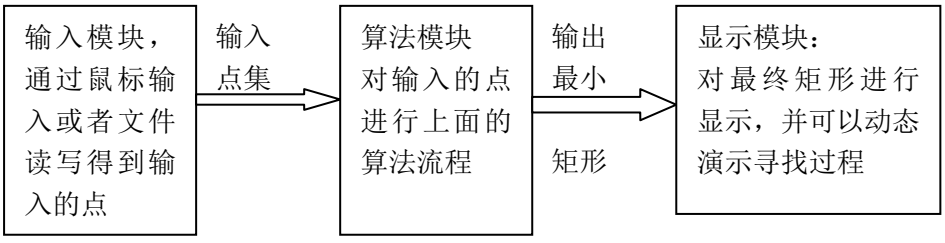
4. 重复 3。
5. 当扫描线 L2 前进至第 n 个点，扫描线 L1 前进一步并转 2 重新初始化。
   i. r+1，即扫描线 L1 前进一步；
   ii. 清空 L、K1、K2；
   iii. 转到 1

- 算法流程图



系统设计

程序采用 Windows 下 Visual C++进行程序设计，主要系统模块如下所示：



系统主要由以上三部分组成，采用 Windows SDK 写成。各个模块的作用由上面所示。支持文件读写操作，便于测试和理解算法流程。

主要文件如下：

- 主要程序文件： BST.cpp(算法实现), FindKRect.cpp(输入，输出，显示，存取等)
- 数据测试文件：general.pt, sp1.pt, sp2.pt, sp3.pt, sp4.pt, sp4_1.pt

数据文件采用二进制格式存取，存放的是点数和点的坐标信息。

复杂性分析

- *完备性*:

从扫描线 L1，L2 的整个扫描过程可以看出，这个算法确保了所有可能出现的恰好包含 k 个点的矩形的左右边界都被取到。同时当 L1，L2 确定时，这个算法保证了当前左右边界确定情况下所有恰好包含 k 个点的矩形的所有可能上下界都被取到。由以上分析可以看出这个算法保证搜索到了全部可能到恰好包含 k 个点的矩形，因此必能找到所求的解（当解存在时）。

- *时间复杂度*:

1. 预处理时间复杂度为 O(nlogn)；
2. 每次初始化 K1，K2 所需点时间复杂度为 O(k)，对整个过程总共为 O(k(n-k))；
3. 每次更新 K1，K2 所需对时间复杂度为 O(logk)，因此整个过程中总共为 o(k(n-k)^2logk+(n-k)^3)；
4. 总共将搜索的矩形数量为 o((n-k)^3)，每个矩形比较所需时间为 o(1)，所以总时间复杂度为 o((n-k)^3)；
5. 当 n/2<k<n 时，总的时间复杂度为 O(n+k(n-k)^2)；

- *难点分析及对策:*

1. 原算法对 K1，K2 对存储采用了二分堆，由于堆同时进行最大值最小值及第 n 大，第 n 小值的操作很不方便，我们这里使用了二叉树进行存储。
2. 本实验的使用的基础数据结构是二叉树，但是与普通的二叉树又有不少不同的地方，主要是由于每个点有横坐标和纵坐标，算法要求有时要根据横坐标排序进行操作，有时要根据纵坐标排序进行操作，所以我们相应在数据的基本操作中加入了 Flag 标志变量，0 时按照横坐标操作，1 是按照纵坐标操作。
3. 本实验中的特例情况比较多，下面有详细的论述。

- *我们的工作*:

我们注意到就原算法本身而言，只能解决 n/2<k<n 这种情况，而且这时这个算法有比较好的性能，这个限制不成立算法就失效了。对算法进行如下修改，即可使之也可以对 0<k<=n/2 的情况适用：

将 K1，K2 的元素数量修改为最大为 n-k+1，初始为 1。当加入一个点 p 时，则按照上面的方法更新 K1,K2。

进行这样的修改后，K1，K2 的大小至多为 n-k，在 n/2<k<n 时候有更好的性能，而且在第二条扫描线的移动过程中树只有插入操作，开销小，而且树中节点的数目就是需要比较的矩形的数目，比较方便。

特例处理

几种特例的情况：

- 有横坐标或者纵坐标相同的点。
  处理方法：首先，我们在每个节点的中加入 id 这个唯一标识每个节点的分量，以至于不会删错点。我们在树的插入和删除节点的操作中保持了顺序的一致性，也就是说，如果插入的时候将大于等于当前节点的节点放入有分支，则删除节点的时候也在有分支中找大于等于当前节点的节点。

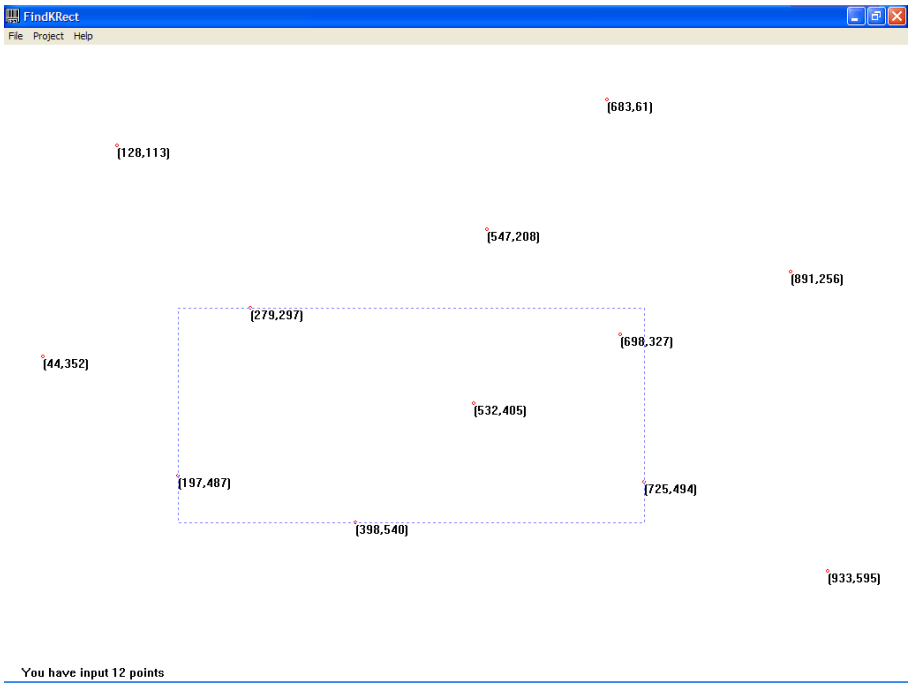
- 结果为一条线段。例如一共 5 个点，从中找包含 3 个点的最大矩形，如果其中有 4 个点共线，求出的矩形应该是包含其中的三个点的线段，按照算法步骤会出现线段长度超出三个最大和最小的坐标。
  出现原因：出现这种情况的原因是一旦找到的矩形为线段，则目标函数无法更新，因为线段的面积为 0。
  处理方法：在输出的时候进行坐标检测，即判断线段端点的坐标是否为所给点的坐标，如果不是，则修改线段的端点。

- 无解或者解不合理情况
  出现原因：出现这种情况的原因是由于有横坐标或者纵坐标相同的点，导致求得的矩形中无意中包含了其他的点，是所求矩形不符合要求。
  处理方法：在每次将要替换当前最小矩形的时候进行检测，也就是检测所求矩形中的包含的点数是否为 k 个，如果不是则排除这个矩形。

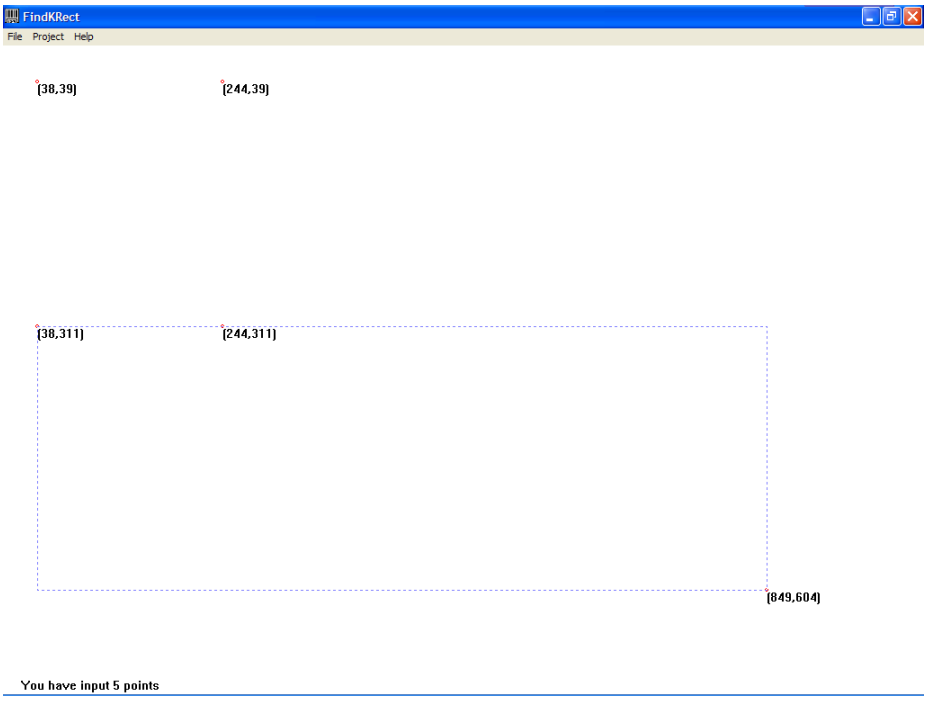
实验结果分析

- 一般情况

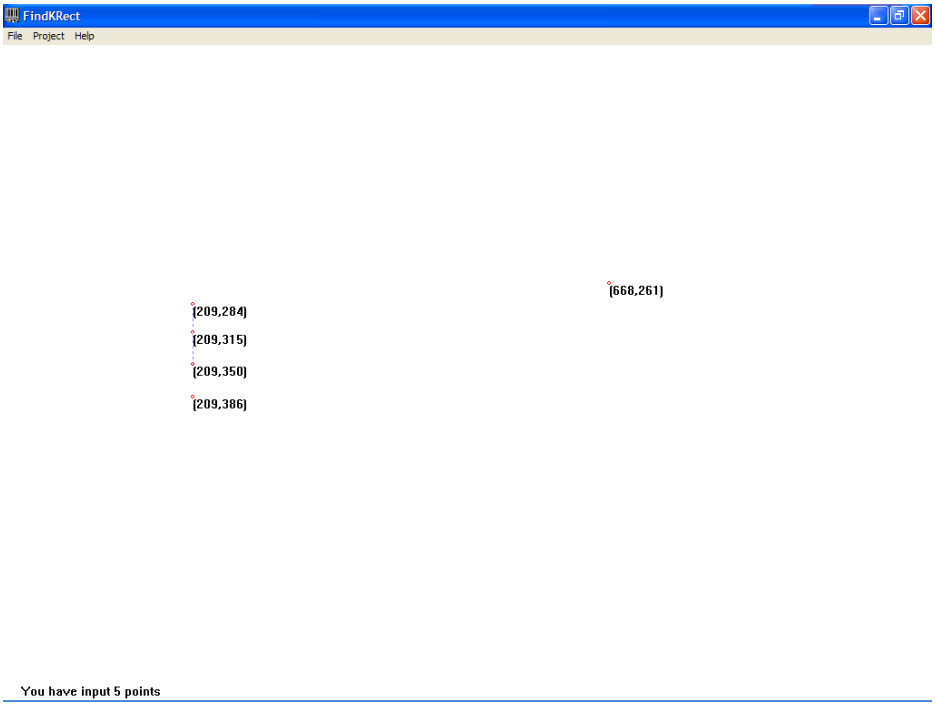
这种情况下点都处于 General Position，即不存在同行，同列的点，如下所示：在 12 个点中寻找包围 6 个点的最小矩形。



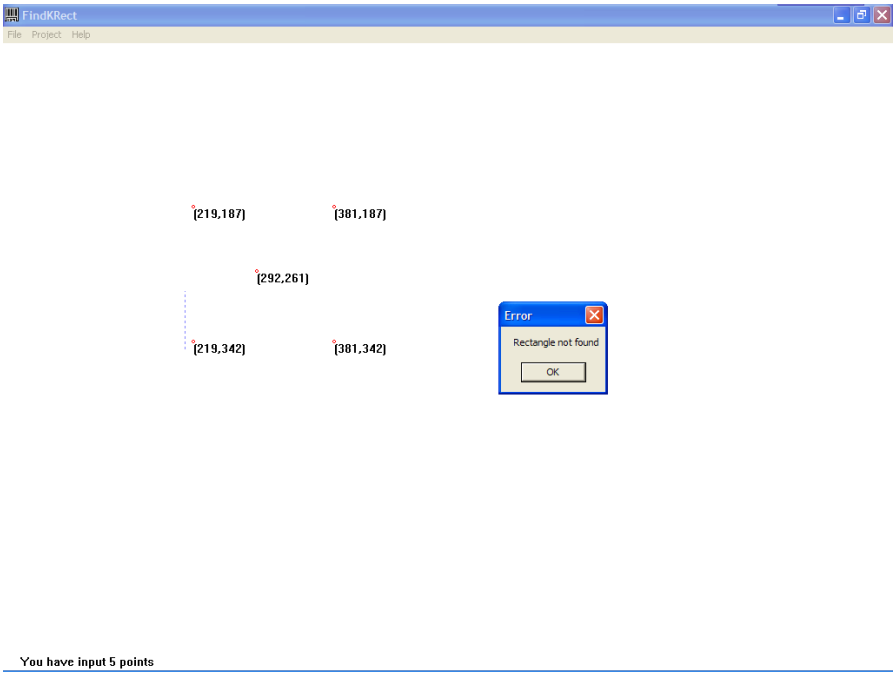
- 特殊情况
  - 存在同行、同列情况：5 个点中找包围 3 个点最小矩形



- 结果为一条线段情况：5 个点中有 4 个在一条直线上，找包围 3 个点的最小矩形



- 无解情况：如下 5 个点中找包围 4 个点的最小矩形，不存在。



实验总结

这学期计算几何课程应该说是一门比较新颖的一门课，跟以前的课不一样，这门课所讲的东西比较新，也比较有意思。在最后小组的 Project 中，我们小组成员通过互相间的合作，从选题，到程序编写，到程序测试，大家都投入了很高的热情。这其中既有选题不知如何选取，也有程序调试的烦恼，更有调试成功的喜悦。从这个实验中，学习到怎样去选择一个适合自己的题目，怎样分工做一件较大的事情，怎样和别人合作等等。

参考文献

[1] Martin Aigner Combinatorical search, Wiley-Teubner Series in CS John Wiley and Sons, 1988

[2] A.Aggarwal, H.Imai, N.Katoh, S.Suri, Finding k points with minimum diameter and related problems. Journal of algorithms. 12(1991) pp.38-56

[3]A.Glozman, K.Kedem, G.Shpitalnik, On some geometric selection and optimization problems via sorted matrices. Lecture Notes in Computer Science

[4]A.Datta, H.-P. Lenhof, C.Schwarz, M.Smid, Static and dynamic algorithms for k_pointclustering problems. Lecture Notes in Computer Science

[5]D. Eppstein, J. Erickson, Iterated nearest neighbors and finding minimal polytopes. Disc and Computer Geom