

# 伪正交区域的凸四边形分解算法的实现与可视化

周伯威 (2016213588)、潘毅铃 (2016311947)、肖剑楠 (2016213589)

## 问题介绍

### 概念简介

伪正交区域 (pseudo-rectilinear region) :

- 水平边和倾斜边交替出现
- 所有内角不大于  $270^\circ$
- 任意倾斜边的影子都不包含顶点(边  $e$  的影子: 是一个区域, 这个区域里每个点都能与  $e$  上(端点不算)的某个点垂直连起来, 且连线在多边形内)

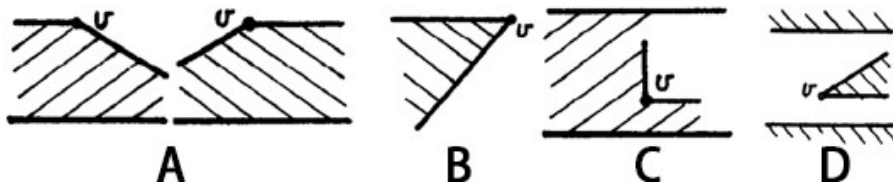
### 选题介绍

我们这次选题的题目为伪正交区域的凸四边形剖分, 主要参考了[Lubiw]和[KKK]这两篇论文。[KKK]这篇论文证明了正交多边形区域能够分解为凸四边形, [Lubiw]在[KKK]这篇论文的基础上做出了新的工作, 展示了两类多边形区域时“CQ hereditary”<sup>1</sup>的, 分别为 1-正交多边形区域, 记为  $P_1$ , 和伪正交多边形区域, 记为  $P_2$ 。其中  $P_2$  多边形区域的分解更具有普适性, 因为我们这次选题主要实现  $P_2$  类多边形区域分解算法的实现与展示。在算法的时间复杂度方面, 对于包含  $n$  个顶点的  $P_2$  区域, 可以在  $O(n \log n)$  时间内分解成多个凸四边形。特别地, 当这个区域是正交区域时,  $O(n \log n)$  是下界。

### 算法实现

算法实现中使用了扫描线技术。节点按照  $x$  坐标进行排序, 算法复杂度为  $O(n \log n)$ 。一条垂直的扫描线从  $x = -\infty$  扫描至  $x = \infty$ , 在遇到节点时暂停。在扫描过程中, 将节点所连的边加入到一个有序队列中, 同时检查该区域的合法性。

在扫描过程中, 需要搜索每个节点的右邻居<sup>2</sup>, 为下一步的剖分做准备。在扫描过程中, 分情况讨论, 共有如下图所示的四种情况。



在情况 A, B 和 C 中, 节点  $v$  所两侧属于同一“进出对”的两条边可能为某些节点带来 initial right neighbor。以一条右边与节点  $v$  连接的节点没有 initial right neighbor, 而任何一个以

<sup>1</sup> “CQ hereditary”的定义是: 一个多边形区域类, 对于其内部的一个多边形区域, 如果它本身不是凸四边形, 那么它有一个“可移除的四边形”, “可移除的四边形”指的是区域中的凸四边形, 它的边是区域的边或者弦, 移除了它区域仍属于这一类多边形区域中。显然, CQ hereditary 类里面的每个多边形区域都是可以被分解成凸四边形的。

<sup>2</sup> 右邻居的定义为, 该区域内  $x$  坐标大于节点  $v$ , 且节点  $v$  可见的节点中,  $x$  坐标最小的节点即为节点  $v$  的右邻居。

一条左边与节点 相连的节点可以继续等待一个 initial right neighbor。同时，在情况 A, C 和 D 中，节点 自身也需要等待一个 initial right neighbor。

## 系统框架

算法实现与效果演示基于 JavaScript 语言，多边形区域及结果演示基于 Canvas。由于算法的输入由用户手工输入，因而数据量较小，使用 JavaScript 这样的解释型语言来实现算法并不会会有明显的负担。

系统除了依赖 jQuery 及 materialize 这两个工具包来支持对 DOM 元素的操作和布局的实现以外，并不依赖其他计算库，线段求交等几何函数为手动实现。

## 交互部分

用户输入：由于伪正交多边形具有诸多特点，所以随机选点输入并不现实。经过讨论，我们设计的输入方式共分为两步：

- 1) 用户根据辅助线提示，在画板上连续取点，并获得多个正交区域；
- 2) 在第一步得到的正交区域基础上，调整节点水平位置，从而得到任意伪正交多边形区域；
- 3) 扫描及分解。

## 算法展示

考虑到扫描线算法的特点，我们提供了两部分的效果展示，具体的交互及细节请查看 manual.pdf。

- 1) 扫描线过程展示
- 2) 分解过程展示

## 成员及分工

任务	参与者
前端框架	周伯威
交互及动画	周伯威、肖剑楠
扫描线算法及区域分解	潘毅铃、肖剑楠

## 小结

这个题目对于我们每个人来说都是一个全新的题目。每个小功能的实现，都是课程所学知识的实践和再检验。与其他的项目不同的是，为了实现这样一个算法并得到好的结果，需要反复研读论文中的每句话，才能够确保最终结果的可靠。

文章中对于扫描线过程中出现的四种情况介绍的比较简略，初读的时候自认为理解的比较全面，考虑到了全部的情况，但是在落实到键盘的时候，发现还是有很多细节并不清楚。为了弄懂作者的意图，我们在项目中往往会加入自己的设想进行尝试，如果猜想的不对，就再阅读一遍原文。最终，算法就是在“读论文-做假设-实践”的这样一个循环中逐渐完善了。还是古人的那句话说的好，“纸上得来终觉浅，绝知此事要躬行。”

## 参考文献

- [Lubiw] Lubiw, Anna. "Decomposing polygonal regions into convex quadrilaterals." Proceedings of the first annual symposium on Computational geometry. ACM, 1985.
- [KKK] Kahn, Jeff, Maria Klawe, and Daniel Kleitman. "Traditional galleries require fewer watchmen." SIAM Journal on Algebraic Discrete Methods 4.2 (1983): 194-206.
- [T] Toussaint, Godfried T. "Pattern recognition and geometrical complexity." Proceedings of the 5<sup>th</sup> International Conference on Pattern Recognition. Vol. 334. 1980.
- [S] Sack, J., and O. An. "Algorithm for Decomposing Simple Rectilinear Polygons into Convex Quadrilaterals,." Proceedings of the 20th Allerton Conference on Communication, Control, and Computing, Monticello. 1982.
- [GJPT] Garey, Michael R., et al. "Triangulating a simple polygon." Information Processing Letters 7.4 (1978):175-179.
- [PLLML] Pagli, L., et al. "On two dimensional data organization 2." Fund. Inform 2 (1979): 211-226.
- [LPRS] Lingas, Andrzej, et al. "Minimum edge length partitioning of rectilinear polygons." Proc. 20th Allerton Conf. Commun. Control Comput. 1982.
- [CD] Chazelle, Bernard, and David Dobkin. "Decomposing a polygon into its convex parts." Proceedings of the eleventh annual ACM symposium on Theory of computing. ACM, 1979.
- [Ln] Lingas, Andrzej. "The power of non-rectilinear holes." International Colloquium on Automata, Languages, and Programming. Springer Berlin Heidelberg, 1982.
- [EOW] Edelsbrunner, Herbert, Joseph O'Rourke, and Emmerich Welzl. "Stationing guards in rectilinear art galleries." Computer vision, graphics, and image processing 27.2 (1984): 167-176.