学号：__2019302110028__

密级：_____

武汉大学本科毕业论文

任意平面多边形切分算法设计与实现

院（系）名 称 ： 计算机学院

专 业 名 称 ： 软件工程

学 生 姓 名 ： 逄博

指 导 教 师 ： 傅佑铭 讲师

二〇二三年五月

郑 重 声 明

本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权归属于培养单位。

本人签名: _____ 日期: _____

摘　　要

计算机在渲染一个形状时，需要首先将形状的曲边转化为折线，由此将待渲染的形状转为多边形，再将其切分为三角形并依次渲染。对于其中将多边形分割为三角形的步骤，设计效率更高和兼容性更好的算法可以提高切分的速度、准确性以及普适性。

本文实现了一种算法，可在不添加新顶点的情况下快速将任意形状带孔凹多边形划分成数个互不重合的三角形，能正确处理多种复杂情况，且产生的划分规整，便于后续处理。

关键词：计算几何；平面多边形；三角化

目　　录

1 引言 …… 1

1.1 概述 …… 1

1.2 研究现状 …… 1

1.3 理论基础 …… 1

1.4 概念定义 …… 2

1.5 代码结构 …… 2

1.6 预期效果 …… 3

2 对凸多边形进行三角化 …… 4

2.1 概述 …… 4

2.2 朴素的划分方案 …… 4

2.3 改进的朴素划分 …… 4

2.4 完美三角剖分 …… 4

3 引用与链接 …… 9

3.1 脚注 …… 9

3.2 引用文中小节 …… 9

3.3 引用参考文献 …… 9

3.4 链接相关 …… 9

4 其它格式 …… 10

4.1 代码 …… 10

4.1.1 原始代码 …… 10

4.1.2 代码高亮 …… 10

4.1.3 算法描述/伪代码 …… 10

4.2 绘图 …… 11

4.3 写在最后 …… 11

参考文献 …… 12

致谢 …… 13

附录 A 数据 …… 14

A.1 第一个测试 …… 14

1 引言

1.1 概述

计算机在对一个平面形状进行栅格化时，一般需要首先将形状的曲边近似转化为折线，由此将待渲染的形状转为多边形，再将其切分为三角形并依次渲染。在切分出的三角形中，具有两个极小锐角的细长三角形被称为“sliver triangles”[1]，不便于栅格化时的计算。因此，在设计多边形栅格化的算法时，常常需要在做到高效的同时避免产生细长三角形。

本文实现了一种近似算法，可在不添加新顶点的情况下快速将任意形状带孔凹多边形划分成数个互不重合的三角形，能正确处理多种复杂情况，且产生的细长三角形数目较少，便于后续处理。

1.2 研究现状

平面多边形三角化作为计算机图形学的基础课题，当下研究较为成熟，有着较多的相关研究。近些年的论文以对已有的三角划分算法应用为主，基于三角划分实现了 ***，***，*** 等功能。而对三角划分算法本身的研究则多发表于二十一世纪初甚至更早之前。

1.3 理论基础

三角剖分（triangulation）：假设在平面有一点集 V，三角形 s 是由点集中的点作为端点的非退化三角形，S 为 s 的集合。则该点集 V 的一个三角剖分 T=（V，S）是一个平面图，满足以下条件：

- 平面图中的任意两个三角形交集面积为 0
- 平面图中所有三角形的并集是点集 V 的凸包

Ear Clipping（耳切法）[2] 是一种较为简易的多边形三角化算法，其思路简单易懂，但缺点是可能会产生较多的细长三角形。原理为不断在多边形内寻找可被切除①的连续三个顶点构成的三角形。对于任意多边形，至少存在两个可被切除的

①当一个三角形内部不包含其他顶点且该三角形属于原多边形一部分时认为其可切除

三角形。轮流切除所有的三角形后算法结束。朴素情况下该算法的复杂度为 $O(n^2)$。

Delaunay triangulation[3] 算法则会生成一组相对更加优秀的解，其满足形成的三角形中最小角尽可能大。但该算法只适用于凸多边形的划分。虽然计算得出的解在处理后也可用于凹多边形划分，但该划分结果可能会产生新的顶点，导致最终的三角形个数增加。Delaunay 三角划分有数种实现方式，其中较为常用的算法如 Bowyer-Watson 算法，其原理是维护一个合法的 Delaunay 三角划分，每次向点集中加入新点，并修改新增点附近的三角划分规则。其复杂度为 $O(n^2)$。或者也可以采用分治法，将点集划分为两部分并合并生成的 Delaunay 三角划分，其时间复杂度为 $O(n * \log_2 n)$[4]。

1.4 概念定义

定义 N_i 为二维平面上的点，N_{ix}, N_{iy} 分别为该点的横纵坐标；

定义有序集合 $S: N_0, N_1, \cdots, N_{n-1}$ 表示有 n 个节点的无孔洞简单多边形，对任意 $i \in [0, n-1]$ 满足 $N_i, N_{(i+1) \bmod n}$ 之间存在一条连边。

定义有序集合 $T: S_0, S_1, \cdots, S_{n-1}$ 表示有 n 段边界的有孔洞简单多边形，其中 S_0 表示多边形的外边界，其余则表示多边形中的孔洞。

1.5 代码结构

本代码采用 C++ 语言自主实现，使用双向链表存储节点及其连边，每个多边形内记录一个指向链表节点的指针，整个图对象内使用数组记录指向每一个多边形的指针。

为了便于数据处理和计算，每个链表节点同时存储了当前节点的标号，部分代码处理时利用节点的标号辅助确定节点的相对位置。

图对象内同时存储了指向全部初始结点的指针，可根据需求进行排序以便于部分处理。

IO 部分实现了以 svg 图片格式输出中间过程和最终结果，可以直观的展示图片分割结果。

1.6 预期效果

代码支持从标准输入或文件读取多边形，并将分割结果以图片或多边形集合形式输出。多边形分割完成后由数个三角形构成，且各三角形之间交集为空，并集为原多边形。分割完成后的多边形中细长三角形的数目较少，大部分三角形较为规则。

2　对凸多边形进行三角化

2.1　概述

凸多边形[①]是平面多边形中最简单最易处理的一类。定义一个多边形 S 满足对于一个凸多边形 S，存在多种方案将其分割为 n-2 个三角形。其中 Delaunay 划分可以生成满足最小角最大特性，细长三角形较少的优秀划分方案。

2.2　朴素的划分方案

由于凸多边形的性质，多边形任意两个顶点之间连边均在多边形内部。因此最简单的划分方案即为对于 $i \in [2, n-2]$，沿边 S_0, S_i 进行一次分割，即可产生总共 n-2 个三角形。该方案最为简便，且产生的三角形均有一个共同顶点，但产生的三角形形状均为细长型，相对形态较劣。

2.3　改进的朴素划分

由于对凸多边形分割后产生的两个多边形仍为凸多边形，我们可以采用分治法来一定程度的优化上述方案。一种可行的方案如下：

- 若当前多边形为三角形则结束算法；
- 随机选择两个不相邻的顶点连边，将当前多边形分割为两部分；
- 对分割出的两个子多边形调用本算法进行分割，并将分割的结果加入答案。

该方案仍然有较大优化空间，例如选择点对进行分割时可以选择多边形较短的轴进行切割以使两部分更加规整，但实际应用中优化意义不甚显著，且计算复杂度大幅增加。

2.4　完美三角剖分

Delaunay triangulation（简称 DT）是一种三角剖分方式，其结果满足以下性质：

- 对于三角剖分中产生的每一个三角形，其外接圆中不包含点集 V 中的其他点。

①每个内角均小于 180 的多边形

- 若对点集的三角剖分结果中每一个三角形的最小角升序排列，则 DT 所得到的数值比其他三角剖分的数值大。
- 若点集中任意四点均不共圆，则产生的三角剖分唯一。

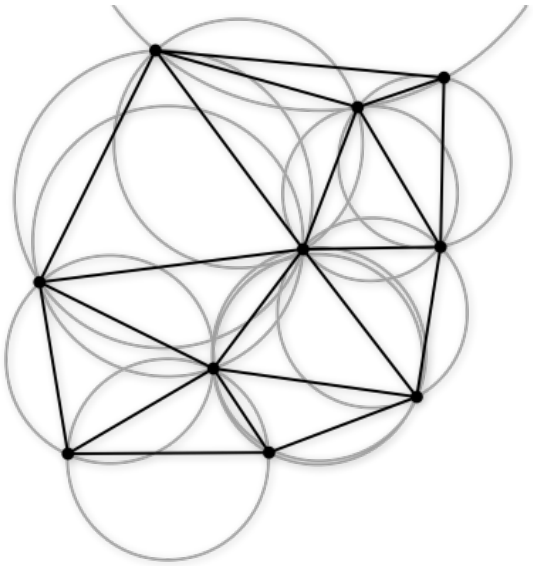

图 2.1　DT 示例

若进行三角剖分的点集中存在四点共圆，虽然计算其 Delaunay triangulation 时得到的结果不唯一，但仍不破坏其余性质。由于最小角最大的特性，DT 可以产生理论最优的划分。而根据三角剖分的性质，对凸多边形顶点点集进行三角剖分所得到的三角形集合即为待求的三角化结果。

一种利用分治思想在 $O(n \log n)$ 复杂度内计算点集的 Delaunay triangulation 方法如下：

首先将给定点集按照 x 坐标升序排列，如下图所示。

图 2.2　排好序的大小为 10 的点集

一旦点集有序，我们就可以不断地将其平分成两个部分，直到子点集大小不

超过 3。然后这些子点集可以立刻剖分为一个三角形或线段。

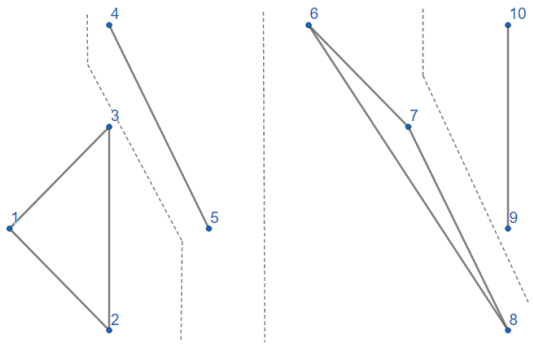

图 2.3　分治为包含 2 或 3 个点的点集

在分治回溯的过程中，已经剖分好的左右子点集可以依次合并。合并后的剖分包含 LL-edge（左侧子点集的边）。RR-edge（右侧子点集的边），LR-edge（连接左右剖分产生的新的边），如图 LL-edge（灰色），RR-edge（红色），LR-edge（蓝色）。对于合并后的剖分，为了维持 DT 性质，我们可能需要删除部分 LL-edge 和 RR-edge，但我们在合并时不会增加 LL-edge 和 RR-edge。

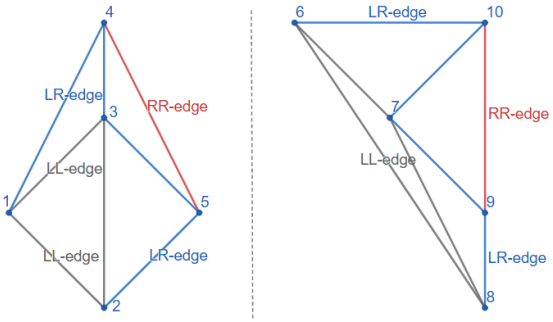

图 2.4　三种不同的边

合并左右两个剖分的第一步是插入 base LR-edge，base LR-edge 是最底部的不与任何 LL-edge 及 RR-edge 相交的 LR-edge。

然后，我们需要确定下一条紧接在 base LR-edge 之上的 LR-edge。比如对于右侧点集，下一条 LR-edge 的可能端点（右端点）为与 base LR-edge 右端点相连的 RR-edge 的另一端点（6, 7, 9 号点），左端点即为 2 号点。

对于可能的端点，我们需要按以下两个标准检验：

1. 其对应 RR-edge 与 base LR-edge 的夹角小于 180 度。
2. base LR-edge 两端点和这个可能点三点构成的圆内不包含任何其它可能点。

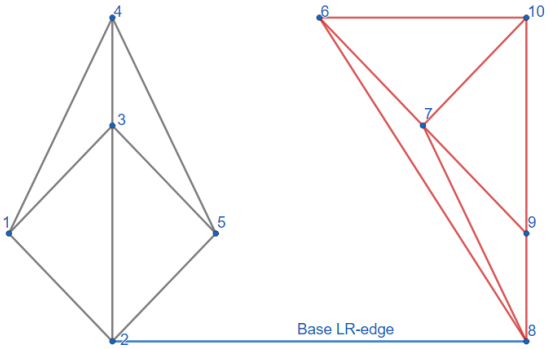

图 2.5　合并左右剖分

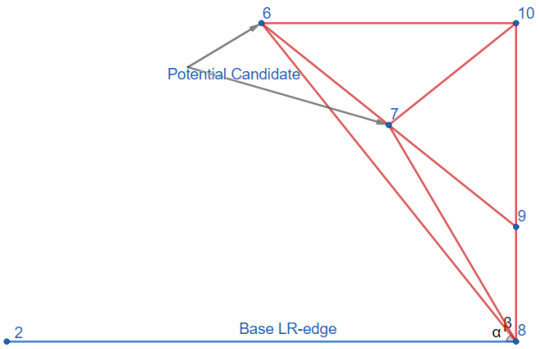

图 2.6　下一条 LR-edge 的可能位置

如上图，6 号可能点所对应的绿色圆包含了 9 号可能点，而 7 号可能点对应的紫色圆则不包含任何其它可能点，故 7 号点为下一条 LR-edge 的右端点。

对于左侧点集，我们做镜像处理即可。

当左右点集都不再含有符合标准的可能点时，合并即完成。当一个可能点符合标准，一条 LR-edge 就需要被添加，对于与需要添加的 LR-edge 相交的 LL-edge 和 RR-edge，将其删除。

当左右点集均存在可能点时，判断左边点所对应圆是否包含右边点，若包含则不符合；对于右边点也是同样的判断。在未出现四点共圆的情况下每次只会有一个可能点符合标准，否则任选其一即可。

重复以上步骤即可完成 Delaunay triangulation 的构建。

由此我们完成了对凸多边形的三角化。

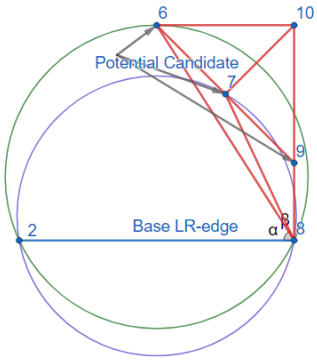

图 2.7　检验可能点

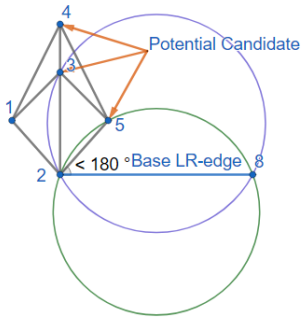

图 2.8　镜像处理另一半点集

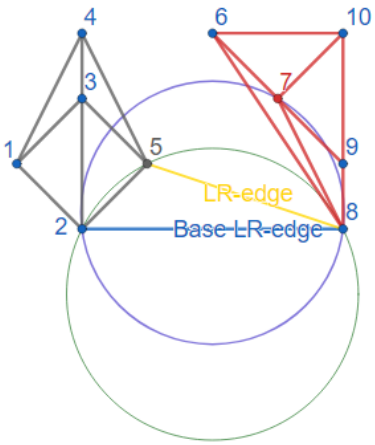

图 2.9　下一条 LR-edge

3 引用与链接

3.1 脚注

注释是对论文中特定名词或新名词的注解。注释可用页末注或篇末注的一种。选择页末注的应在注释与正文之间加细线分隔，线宽度为 1 磅，线的长度不应超过纸张的三分之一宽度。同一页类列出多个注释的，应根据注释的先后顺序编排序号。字体为宋体 5 号，注释序号以“①、②”等数字形式标示在被注释词条的右上角。页末或篇末注释条目的序号应按照“①、②”等数字形式与被注释词条保持一致。示例：这里有个注释①。

3.2 引用文中小节

如引用小节 3.2

3.3 引用参考文献

这是一个参考文献引用的范例

还可以采用上标的引用方式

引用多个文献

文献引用需要配合 BibTeX 使用，很多工具可以直接生成 BibTeX 文件（EndNote, NoteExpress, 百度学术，谷歌学术），此处不作介绍。

3.4 链接相关

模板使用了 hyperref 处理相关链接，使用href可以生成超链接，链接周围的方框在打印时不会出现。可以在 cls 文件中修改相应的 hypersetup 项来关闭方框：\hypersetup{hidelinks}。如果需要输出网址，可以使用url包，示例：https://github.com。

①我是解释注释的

4 其它格式

4.1 代码

4.1.1 原始代码

朴实的代码块：

使用 verbatim 可以得到原样的输出，如下：

```
print("Hello world!")
```

使用listings环境可以对代码进行进一步的格式化，如下：

```
import numpy as np

a = np.zeros((2,2))
print(a)
```

4.1.2 代码高亮

还可以对代码进行高亮，请参考 Code Highlighting with minted。请先到 cls 文件中启用 minted 库。注意使用 Minted 库时，需要系统默认 Python 有 Pygments 库，可以通过$ pip install Pygments 来进行安装。且需要在编译时加上--shell-escape参数，否则会报错。

4.1.3 算法描述/伪代码

参考 Algorithms，下面是一个简单的示例：

Result: Write here the result

initialization;

while *While condition* **do**

    instructions;

    if *condition* **then**

        instructions1;

    else

        instructions3;

    end

end

算法 1: How to write algorithms

4.2 绘图

关于使用 LaTeX 绘图的更多例子，请参考 Pgfplots package 中的例子。一般建议使用如 Photoshop、PowerPoint 等制图，再转换成 PDF 等格式插入。

4.3 写在最后

工具不重要，对工具的合理运用才重要。希望本模板对大家的论文写作有所帮助。

参考文献

[1] EDELSBRUNNER H, LI X-Y, MILLER G, et al. Smoothing and cleaning up slivers[A]. Proceedings of the thirty-second annual ACM symposium on Theory of computing[C], 2000 : 273 – 277.

[2] EBERLY D. Triangulation by ear clipping[J]. Geometric Tools, 2008 : 2002 – 2005.

[3] 闵卫东, 唐泽圣. 二维任意域内点集的 Delaunay 三角划分生成算法 [J]. 计算机学报, 1995, 18(5) : 365 – 371.

[4] LEE D-T, SCHACHTER B J. Two algorithms for constructing a Delaunay triangulation[J]. International Journal of Computer & Information Sciences, 1980, 9(3) : 219 – 242.

致谢

以简短的文字表达作者对完成论文和学业提供帮助的老师、同学、领导、同事及亲属的感激之情。

附录 A　数据

A.1　第一个测试

测试公式编号

$$1 + 1 = 2. \tag{A.1}$$

表格编号测试

表 A.1　测试表格

11	13	13	13	13
12	14	13	13	13