

Alphashape 轮廓线提取—实验总结报告

马靖	吉晓飞	徐峰
2014213475	2014213472	2014213510
软件学院	软件学院	软件学院
1554594081@qq.com	296722640@qq.com	517326756@qq.com

1 简介	2
2 功能描述	2
2.1 实验环境	2
2.2 使用手册	2
2.3 项目清单	3
3 算法	4
3.1 图片边缘点提取	4
3.2 delaunay 三角剖分	4
3.3 Alpha Shape	6
4 总结与展望	8
5 参考文献	8

1 简介

计算几何中，alpha shape (α -shape) 是描述欧氏平面有限点集形状的一组分段线性简单曲线。Alpha Shape 最早在 1981 年被构想出来，随后它被应用与多种科学与工程领域，其中包括模式识别，数字采样和处理，结构分子生物学等等。

Alpha-Shapes 可以用来从一堆无序的点集中提取边缘。设有一点集 s 的 Alpha-Shapes 是一个多边形，这个多边形是由点集 s 和半径参数 α 决定的且唯一。它的原理可以想象成一个半径为 α 的圆在点集 s 外滚动，当 α 足够大时，这个圆就不会滚到点集内部，其滚动的痕迹就是这个点集的边界线。因此，当 α 值很小，则每个点都是边界；如果 α 很大 ($\alpha \rightarrow \infty$) 时，则求出的边界线为点集 s 的凸包。

本实验在图片中运用 alphashape 算法提取特征曲线，采用标准的 Web 技术，只需一个浏览器就能观赏到我们的成果，方便展示给他人。Web 技术中包含 HTML、CSS 和 JavaScript。HTML 负责文档结构，CSS 负责外观样式，JavaScript 负责交互与执行算法。以下为实验主要内容。

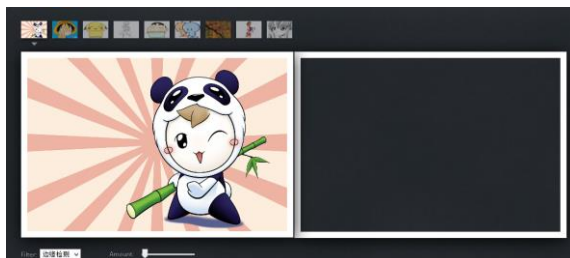
2 功能描述

2.1 实验环境

Firefox (or Safari)浏览器

2.2 使用手册

选取图片：



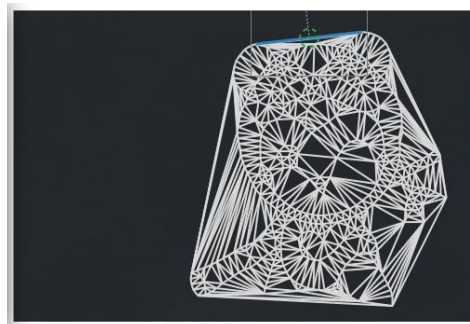
提取边缘:



像素点采样:



三角剖分:



Alpha Shape



2.3 项目清单

demo 主页———[alpha/index.html](#)

主页样式 ———[alpha/css/playground.css](#)

示例图片 ———[alpha/images](#)

javascript 算法———[alpha/script/](#)

3 算法

3.1 图片边缘点提取

本部分需要提取图片的边缘并离散成一系列的像素点，主要分为以下三个部分：

(1) 边缘检测

图像的边缘指的是图像局部强度变化最显著的部分，图像的边缘有方向和幅度两个属性，沿边缘方向像素变化平缓，垂直于边缘方向像素变化剧烈，这种边缘上的剧烈变化可以用微分算子检测出来，通常用一阶或者二阶导数来检测边缘。在本实验中我采用了二阶微分算子拉普拉斯算子，它的定义为：

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

其中：

$$\frac{\partial^2 f}{\partial x^2} = f[x+1, y] - 2f[x, y] + f[x-1, y]$$

$$\frac{\partial^2 f}{\partial y^2} = f[x, y+1] - 2f[x, y] + f[x, y-1]$$

可以用多种方式将其表示为数字形式，对于 3*3 的区域，经验上被推荐最多的形式为：

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

(2) 彩色图像灰度化

灰度化一般有三种方法：

最大值法：使 R、G、B 的值等于三个值最大的一个。

平均值法：使 R、G、B 的值等于三个值的平均值。

加权平均值法：使 R、G、B 的值等于三个值为三者的加权平均数。

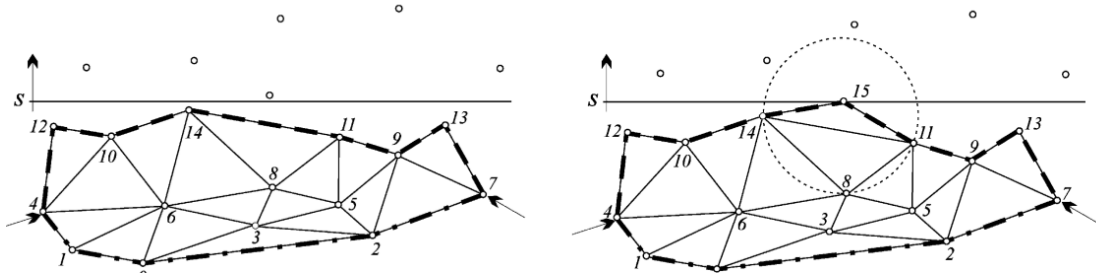
(3) 采样

通过以上两个步骤得到的是连续的边缘像素点，可以通过选取离散区域的大小来控制离散的程度，比如选取 3*3 的区域，则在此区域中只选取一个采样点。

3.2 delaunay 三角剖分

本部分三角剖分算法为扫描线算法计算给定点集的三角剖分结构，主要参考论文 An efficient sweep-line Delaunay triangulation algorithm [Borut Z'alik 2005.]，

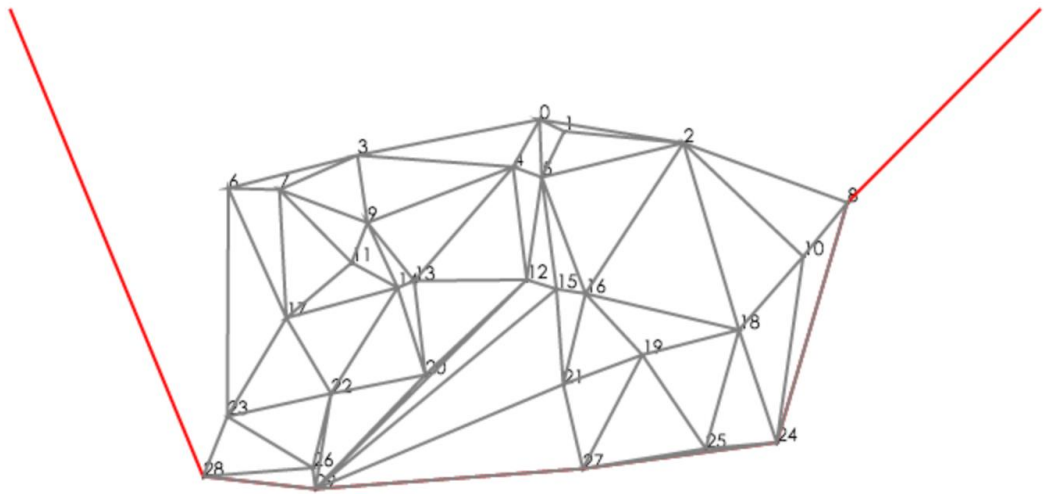
复杂度为 $O(n \lg n)$ 。此算法中涉及到 **advancing front** 的概念，将所有点按 y 坐标排序，如下图点 12 到点 7 为此时的 **advancing front**，当扫描到点 15 时若连接的三角形 14-15-11 符合 **delaunay** 三角形性质则加入结果集。



主要分为三步：

(1) 初始化三角形和 **advancing front**

初试的三个点可能组成的 **advancing front** 集合可能有很多种情况，此算法在开始时根据给定点集包围盒增加两个位于最左下和最右下辅助点，这样在下一步扫描点时就可以只考虑点的 x 坐标投影永远落在 **advancing front** 上的情况。最后去掉所有与辅助点相连的三角形。如下图，红色折现为 **advancing front**，灰色网格为最后三角剖分结果。



(2) 三角剖分

每扫描到一个新点 p 后做其 x 坐标的投影，落在 **advancing front** 的点 a, b 之间，连接 abp ，递归检测空圆性质，若不符合则 **flip**。然后依次将 p 与左右两边可能形成三角形的点相连，如上方法检测空圆属性。

(3) 凸包找丢失的边缘三角形

最后的 **advancing front** 边界可能并不是凸包，此时使用了类似 **Gram**

Scan 的方法将丢失的三角形找回来并对这些三角形进行 delaunay 空圆性质检测，符合条件的加入最后的结果集合中。

3.3 Alpha Shape

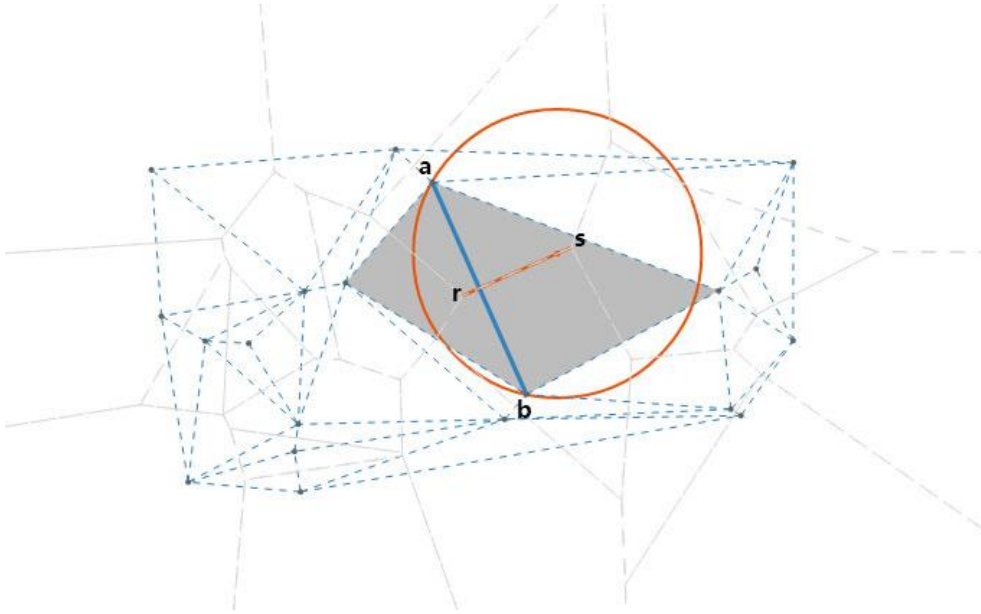
Alpha Shape 的一般定义是：

P 是包含一系列点的集合，对于点集 P 中的任何两个点 p 和 q，如果满足：存在一个半径为 α 的圆，这个圆经过 p 和 q，点集 P 中的点都不在这个圆的内部，则由 pq 定义的线段是 Alpha Shape 中的一条边。

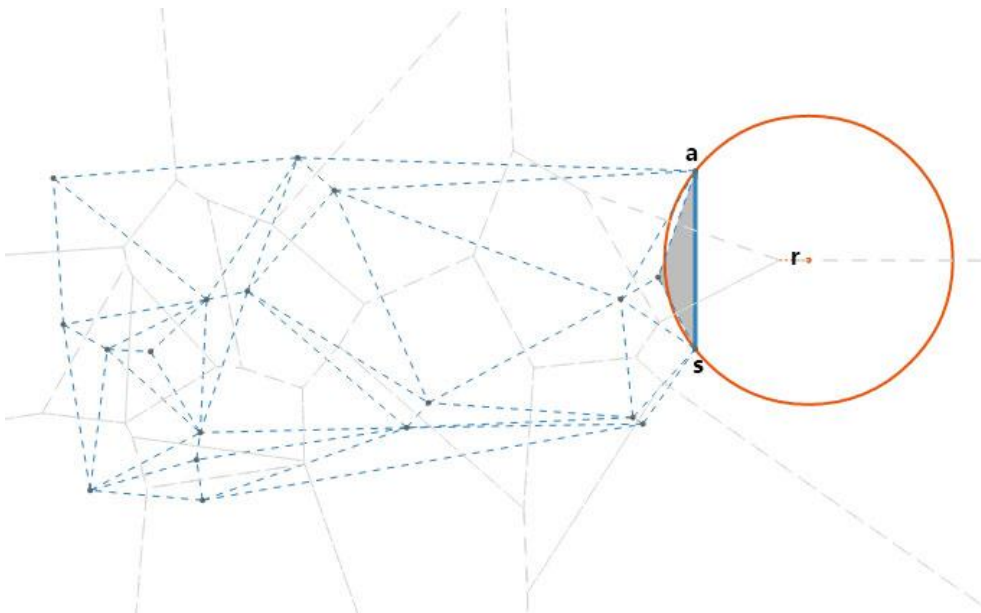
如果根据 Alpha Shape 的定义，考虑点集 P 中的每一对点，判断是否存在满足要求的圆，则需要 $O(n^3)$ 的时间复杂度。根据文献[2]可以知道，Alpha Shape 是点集 P 的 delaunay 三角剖分的子图，而且利用 delaunay 三角剖分的性质，对于每条 delaunay 三角剖分中的边，可以在 $O(1)$ 时间内判断它是否属于 Alpha Shape 中的边。

我们可以发现，对于一条属于 delaunay 三角剖分中的边，要判断它是否属于 Alpha Shape 中的边，可以采用如下策略进行判断：

情况 1: 当这条 delaunay 三角剖分中的边 ab (如下图蓝色实线所示) 有两个相关联的 delaunay 三角形时 (如下图灰色阴影三角形所示)。下图中的实心点表示原来点集 P，蓝色虚线表示 delaunay 三角剖分，灰色虚线表示 voronoi 图。要判断蓝色实线 ab 是否属于 Alpha Shape 中的边，可以在两个相关的灰色阴影 delaunay 三角形的两外心所连成的线段 rs (如下图的橘黄色实线所示) 上找出满足要求的 Alpha 圆。我们可以知道，这两个灰色阴影三角形对应的外心 r 和 s，就是 voronoi 图中的顶点。可以证明：如果能找出对应的 Alpha 圆，则 Alpha 圆的圆心必然在 rs 上。并且，圆心在 rs 上且经过 a 和 b 的圆的半径满足大于等于 $\alpha_{\min}(a,rs)$ ，小于等于 $\alpha_{\max}(a,rs)$ 。 $\alpha_{\min}(a,rs)$ 表示点 a 到线段 rs 的最近距离， $\alpha_{\max}(a,rs)$ 表示点 a 到线段 rs 的最远距离。因此，如果 α 满足 $\alpha_{\min}(a,rs) \leq \alpha \leq \alpha_{\max}(a,rs)$ ，则存在 Alpha 圆经过 ab，即 ab 是 Alpha Shape 中的边；反之，如果 α 不满足 $\alpha_{\min}(a,rs) \leq \alpha \leq \alpha_{\max}(a,rs)$ ，则 ab 不是 Alpha Shape 中的边。



情况 2: 当这条 delaunay 三角剖分中的边 ab (如下图蓝色实线所示) 只有一个相关联的 delaunay 三角形时 (如下图灰色阴影三角形所示)。与情况 1 类似, 只不过原来的线段变成了一条射线 r , 点 r 是灰色阴影 delaunay 三角形的外心。这时, $\alpha_{\min}(a,r)$ 表示点 a 到射线 r 的最短距离, 如果 α 满足 $\alpha_{\min}(a,rs) \leq \alpha$, 则 ab 是 Alpha Shape 中的边; 反之 α 不满足 $\alpha_{\min}(a,rs) \leq \alpha$, 则 ab 不是 Alpha Shape 中的边。



4 总结与展望

本次实验实现了 Alpha Shape 算法，并把 Alpha Shape 应用到轮廓提取中。可以看出，给出一定的采样点，选取适当的 Alpha 值，构造出的 Alpha Shape 可以比较接近地还原原来的形状。但是，如果希望通过 Alpha Shape 尽可能近似的还原原来形状，对采样点的密度以及选取的 Alpha 值有一定要求。那么如何确定采样点的密度和 Alpha 值呢？目前的情况是通过用户的手动操作调整采样点的密度和 Alpha 值来得到希望得到的 Alpha 值。今后的工作可以对两个的数据进行一系列的研究，希望能通过算法自动得到采样点的密度和 Alpha 值。

5 参考文献

- [1] Žalik B. An efficient sweep-line Delaunay triangulation algorithm[J]. Computer-Aided Design, 2005, 37(10): 1027-1038.
- [2] H. Edelsbrunner, D.G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. IEEE Trans. Inform. Theory, 29:551–559, 1983