P.O.L.Y
实时的LOW-POLY风格影像生成器

计76 王西平 2017011388

Abstract

- 动机与起源
- 目前效果的演示
- 应用的处理流程与主要结构
- 难点
  - 边缘检测
  - 三角剖分
- 在影像优化中做的尝试
- 总结

动机与起源

- low-poly style 一种图片处理的风格,还算流行
- 在去年的情系母校培训中第一次接触到这种风格
- 觉得非常的酷炫
- 给自己学校也做了一张
- 从晚上十点做到早上六点
- 那么问题来了

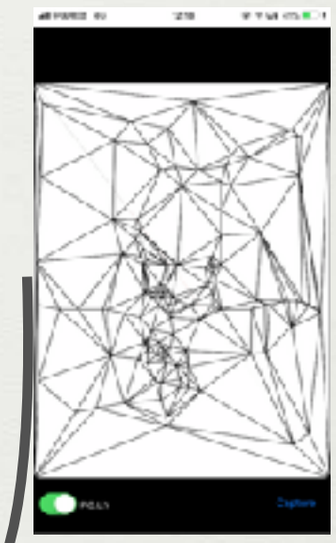
目前效果演示

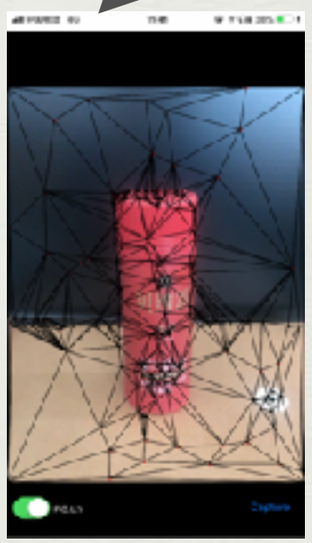
——一段小小的视频

获取图像的边缘信息

选取关键点 并加入随机散点

三角剖分





合并

提取指定像素的颜色信息

绘图层

难点

- 关键点与随机点的选取
- 处理效率
  - 获取图像的边缘信息
  - 三角剖分

边缘检测

- (图像大小 640*480)

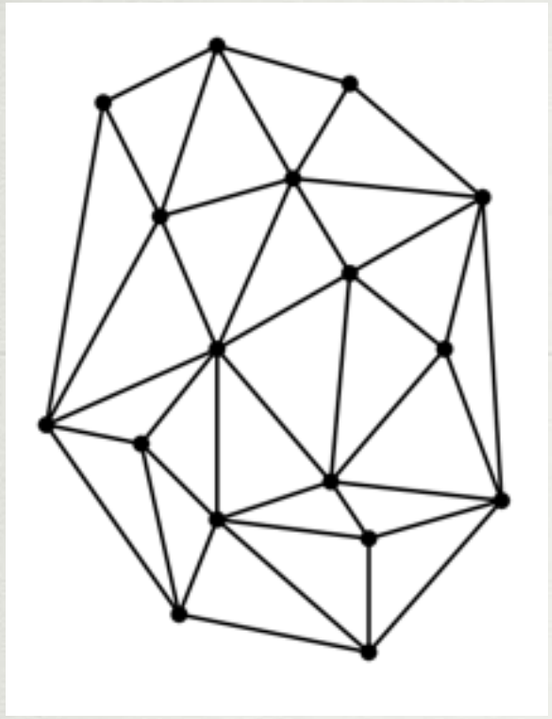
- 主要思想:“相近像素间的差异度大 -> 位于边缘的像素”

- 一开始尝试单线程朴素算法,计算每个像素与四联通像素的RGB的欧几里得距离之和,人工设定阈值(单帧 500-700 ms)

- 后基于GPUImage2 框架,写了一个opengl 的滤镜,GPU加速(单帧 20ms)

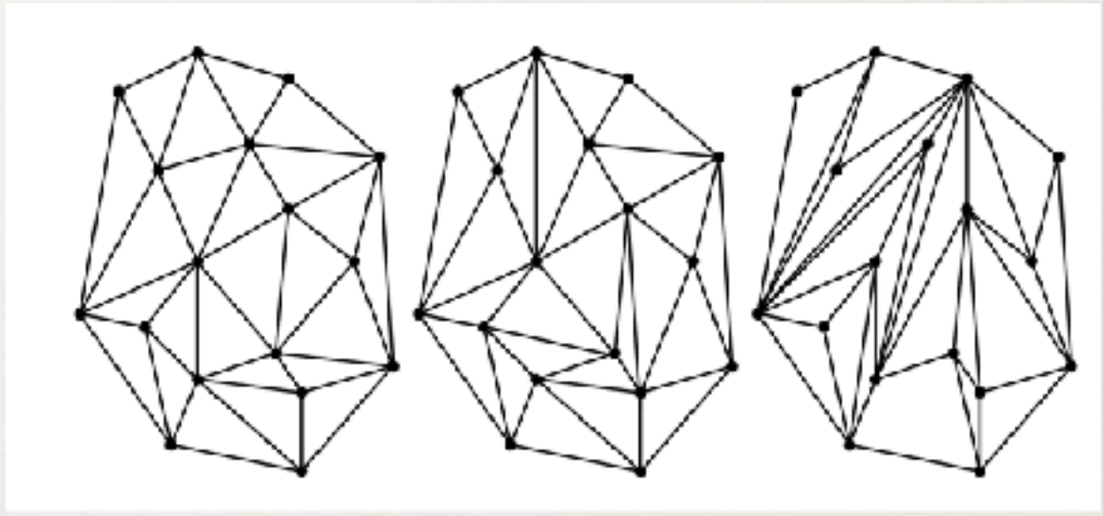
    - 参考Sobel算子 减少乘法次数,扩大监测面积(单帧 <13ms)

- 从的到的灰度图中随机选择一定数量的关键点



三角剖分

- 将平面上的点连线组成若干三角形
- 外边缘形成一个凸包
- 2n - 2 - k 个三角形
- 3n - 3 - k 条边
- k 是凸包上点的数量



有若干种连线的方式满足条件

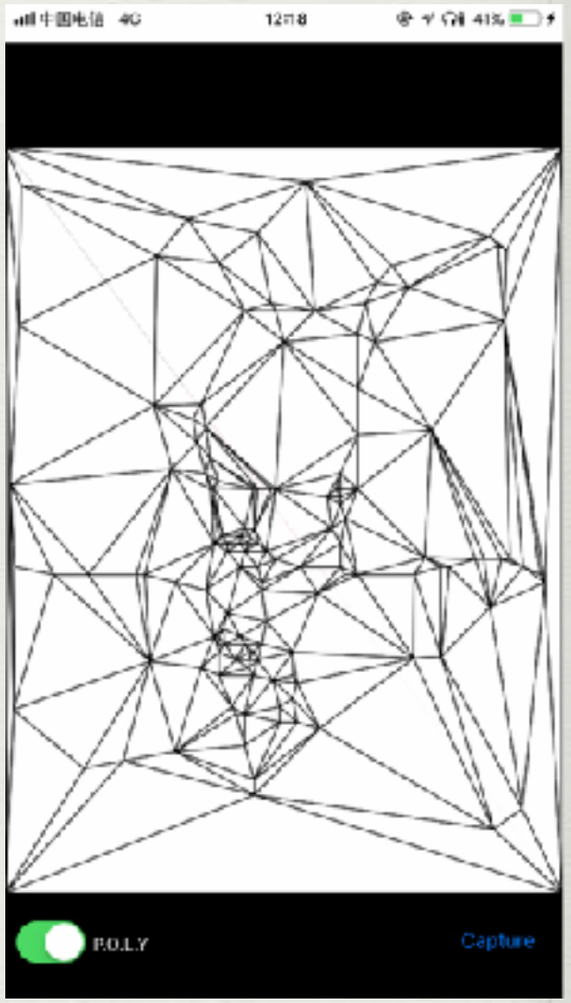
Which one is “better” ?

考虑最小的一个角的大小

Delaunay triangulation

红色随机点的用处?
让三角形不那么尖锐

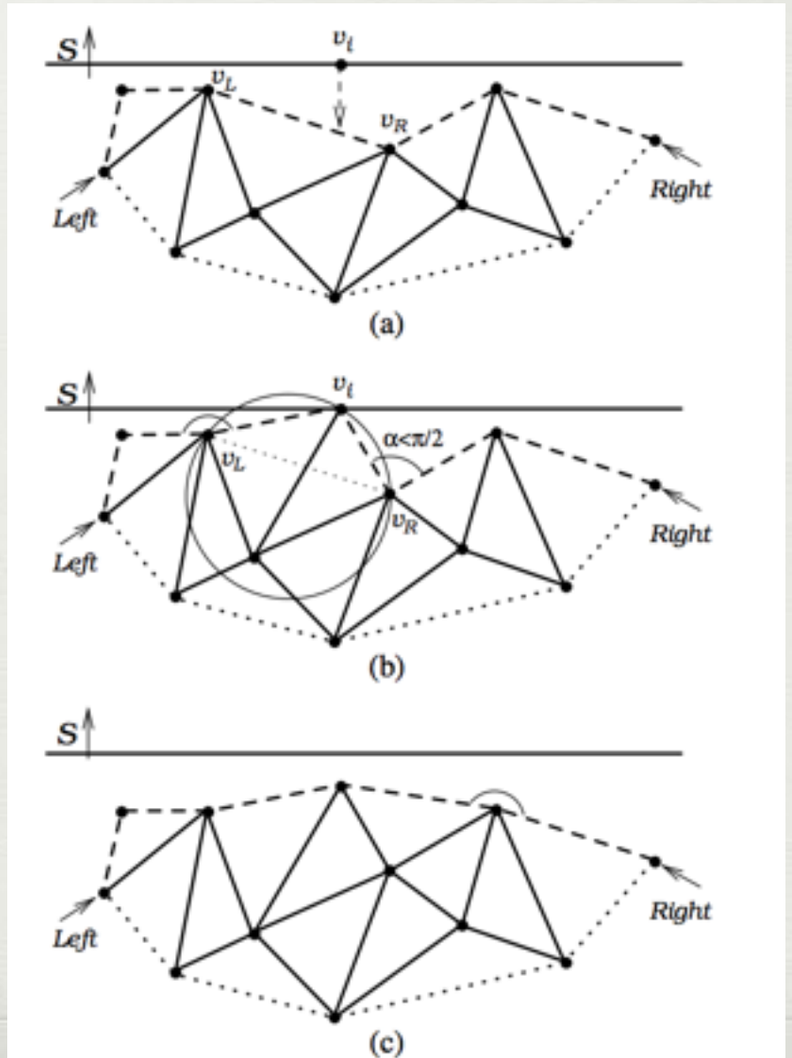
- 平面图的Voronoi图的对偶图
- “最大化最小角”
- 唯一性(空圆特性)



各种算法

- N^2 的朴素算法
- 一系列NlogN算法
- 扫描线/扫描圆
- 分治算法
- 随机增量算法
- V图的对偶图

我实现过的:
一个N^4的三维凸包
朴素的插入算法
带强剪枝的插入算法
扫描线算法



一些成功的尝试/创新

- 针对视频的特殊优化:
  - 关键点的稳定性:若上一帧的关键点在这一帧依旧处于边缘,则大大增加被选中的概率
  - 为所有的随机点设置一个较低更新率,减少闪动
  - 新的边缘点被选中的概率与当前已经存在的关键点的数量成反比:关键点数量稳定

一些失败的尝试尝试过

- 用更快的NlogN的三角剖分——常数太大,实现复杂,在N<=500的数量级不占优势

- 在更新帧与帧之间的关键点的同时动态维护三角剖分— 删除复杂度高,且快速移动时,关键点变化较快,速度慢于直接重建

工作量

- 学了下Swift 与IOS开发
- 最终项目的总代码量1500排
- Open-GL 照着sobel轮子手撸了两个滤镜(朴素的相邻像素差)
- 读了些有关Delaunay triangulation的paper,并对不同的算法进行了对比

运用到的工具&算法

- Swift&Xcode
- Open-GL & GPU
- low pass filter
- Delaunay triangulation
- graph-based object detection
- Delaunay triangulation & Voronoi

谢谢!