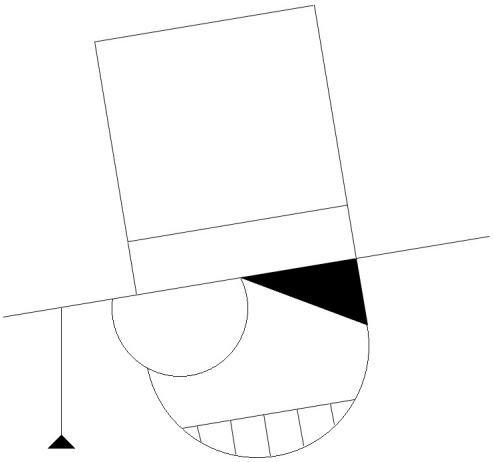
光栅图形学作业

顾煜贤 **2017011421**

0. 成品——怪盗基德



1. 基本选题

基本选题为区域填充，同时实现了画线、画圆。加分项实现了抗锯齿（圆形的抗锯齿没有实现）。编程语言使用的是c++，第三方库使用opencv。

程序入口在main.cpp中。utils.h和utils.cpp中分别是各种方法定义与实现。

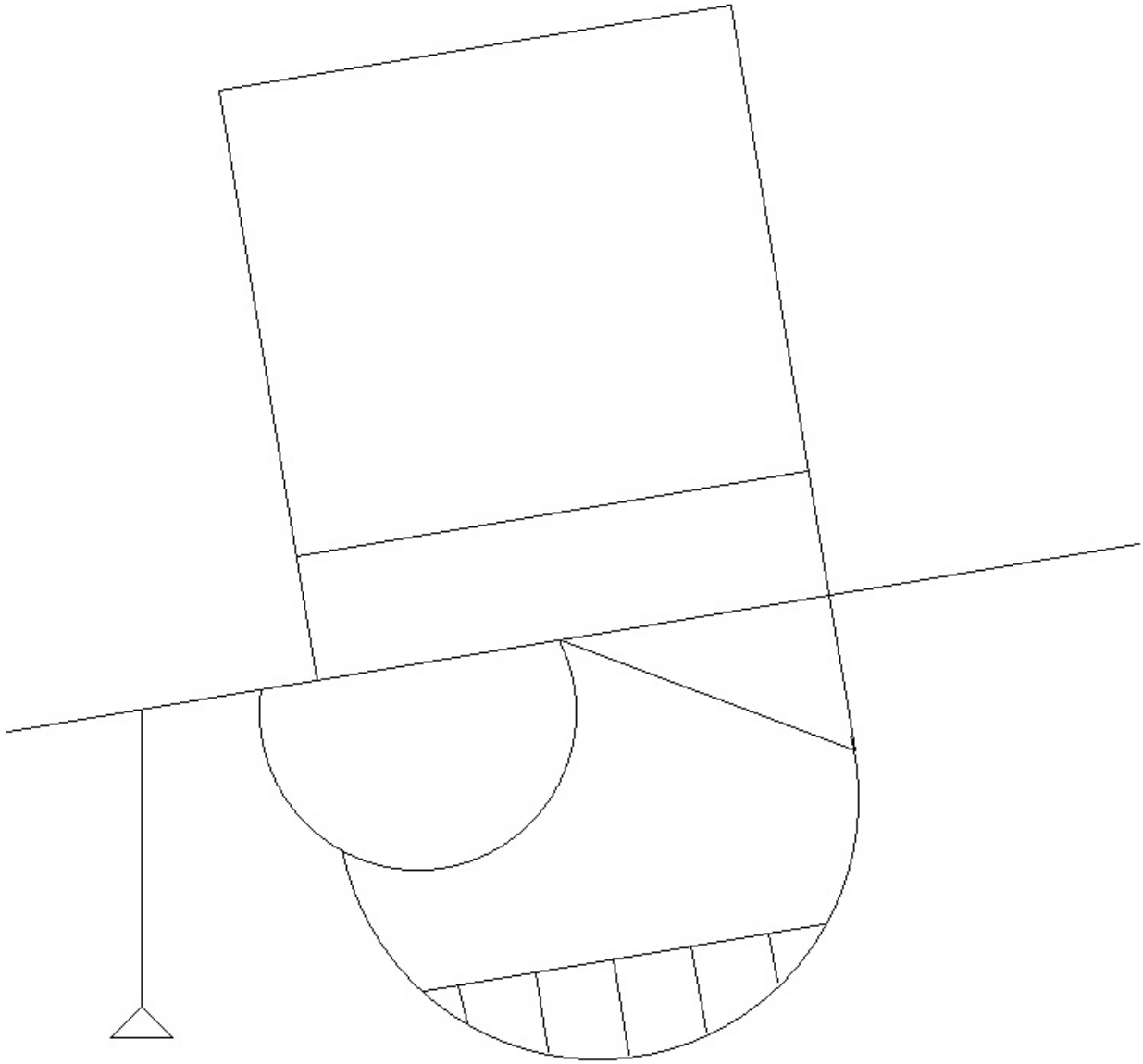
`make`编译，编译完成后`./kid`出图。

1.1 画线

实现的画线函数为`void drawLine(cv::Mat &image, int x1, int y1, int x2, int y2, const Color3 &c)`。

函数行为如下：输入两个端点坐标，(x1, y1)，(x2, y2)，画出过两个端点的线段。颜色定义在Color3对象中，有rgb三个通道。

实现效果如下图所示，可以看到明显的毛刺效果。



1.2 画圆

画圆实现了两个算法，一个是书上的中点画圆算法，另一个是使用多边形逼近圆形的算法。

实现画圆的函数为

- `void drawCircle(cv::Mat &image, int x, int y, int r, int ang1, int ang2, const Color3 &c)`

  统一画圆接口，输入圆心位置，半径，起始和终止角度，以及颜色，可以画出圆弧。
- `void midCircle(cv::Mat &image, int x, int y, int r, int ang1, int ang2, const Color3 &c)`

使用中点画圆法画圆。

- `void polygonCircle(cv::Mat &image, int x, int y, int r, int ang1, int ang2, const Color3&c)`

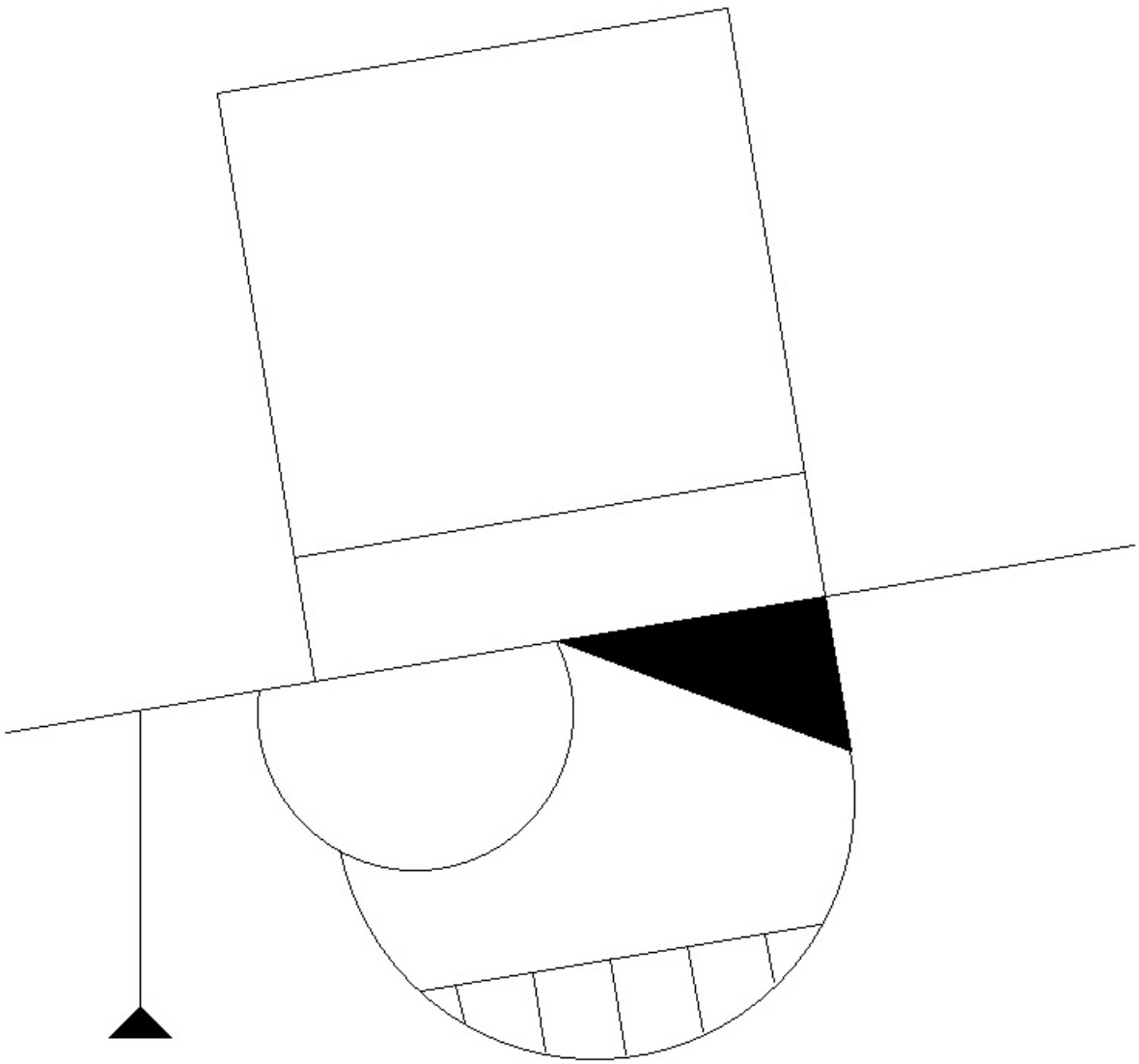
  使用多边形逼近算法。

1.3 区域填充

实现函数为`void fill(cv::Mat &image, const std::vector<Point2> &vp, const Color3 &c)`。传入一个二维的点列，填充颜色定义在Color3对象中。

实现方法为队列实现。即维护一个活性边表，按y坐标逐层扫描。可以实现任意形状多边形填充（包括凸多边形和凹多边形）。

效果如下图：

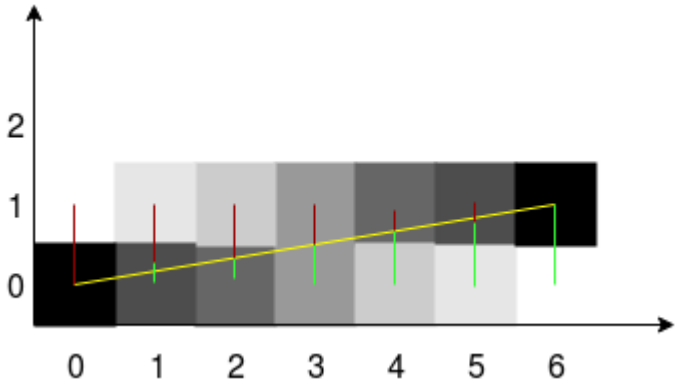


可以看到毛刺效果并没有消失。

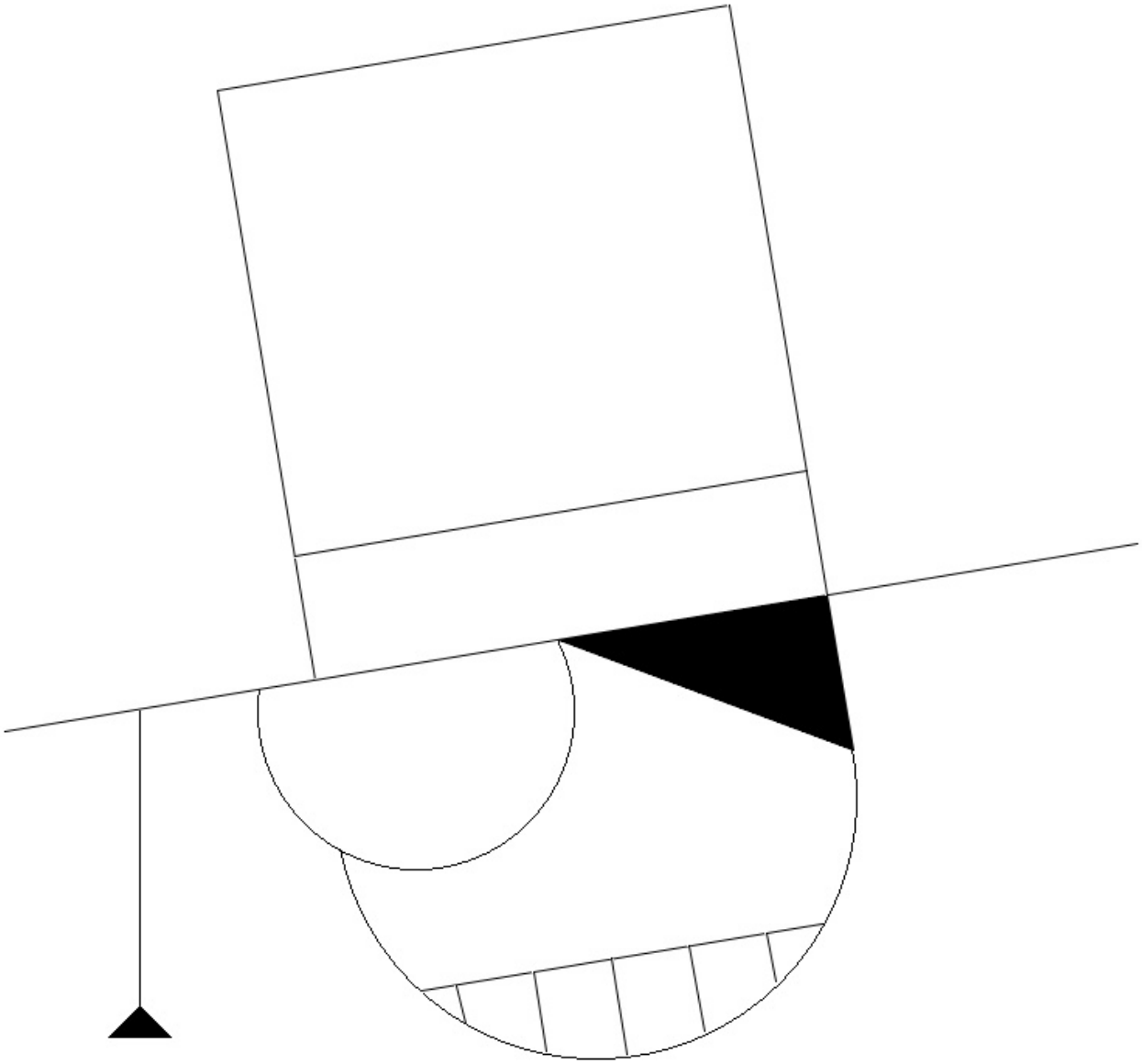
2. 加分项——反走样

反走样使用快速方法（参考https://www.geeksforgeeks.org/anti-aliased-line-xiaolin-wus-algorithm/）

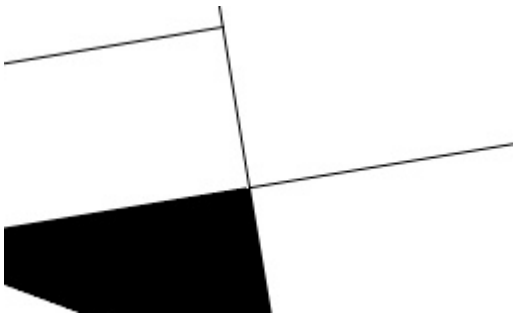
即根据Bresenham算法过程中直线与两个像素中心的距离来决定两个像素的灰度。



效果如下图：



放大之后的边缘：



可以看到边缘更加平滑。