

# 光栅图形学实验报告

张钰晖 (2015011372, yuhui-zh15@mails.tsinghua.edu.cn) - 2017年3月17日

## 一、文件组织

文件夹分为两个文件夹

- lodepng文件夹（使用lodepng轻量级图像处理库实现，主代码文件为main.cpp，含有两个lodepng库文件分别为lodepng.h和lodepng.cpp，内含makefile文件，编译文件为a.out，运行后将得到5个png）
- opencv文件夹（使用opencv图像处理库实现，主代码文件为main.cpp，，内含makefile文件，编译文件为a.out，运行后将得到5个png，和lodepng功能完全相同）

每个文件夹下生成的5个图像文件为：

- circle\_1.png文件为未进行反走样画圆，画布大小500\*500（采用改进的不含浮点运算操作的中点画圆算法）
- circle\_2.png文件对circle\_1.png的圆进行区域填充，画布大小500\*500（采用扫描线区域填充算法）
- circle\_3.png文件为进行反走样画圆，画布大小500\*500（采用自己提出的算法，下面会详细阐释）
- circle\_4.png文件为进行反走样画圆，画布大小500\*500（采用SSAA算法，仅限opencv程序）
- circle\_5.png文件为进行反走样画彩色圆，画布大小600\*600（采用自己提出的算法）

## 二、测试方法

为了方便使用调试，采用命令行传参的形式，在命令行输入指令`./a.out x0 y0 r`三个参数即可在circle\_1.png~circle\_4.png得到以(x0, y0)为圆心r为半径的采用不同算法生成的圆(注意画布大小为500\*500，超出画布部分将无法显示，程序对超出部分进行了处理保证程序的鲁棒性)。如果参数不全或者过多，则会自动使用程序默认参数。所有的参数也可直接在源码中更改测试（可调参数均在main()函数内）。

## 三、算法描述

### 1、改进的中点画圆法

对应于程序中的DrawCircle(x0, y0, r, color)函数，实现了未进行反走样的画出圆心在(x0, y0)半径为r颜色为color的圆。采用课本24页中点画圆算法的主体思想，利用了圆的八对称性，避免了复杂的根号运算操作。并且进行了改进，所有操作均为整数运算，不含浮点数运算，故进一步提高了算法的效率。

### 2、扫描线区域填充算法

对应于程序中的Fill(x0, y0, oldcolor, newcolor)函数，实现了从(x0, y0)开始将oldcolor转换为newcolor的区域填充算法。采用课本29页的扫描线区域填充算法，大致分为四个步骤，初始化、出栈、填充并确定种子点所在区段、确定新的种子点。相比递归的FloodFill算法，对于每一个待填充区段，只需压栈一次，提高了区域填充的效率。

### 3、自己提出的圆的反走样算法

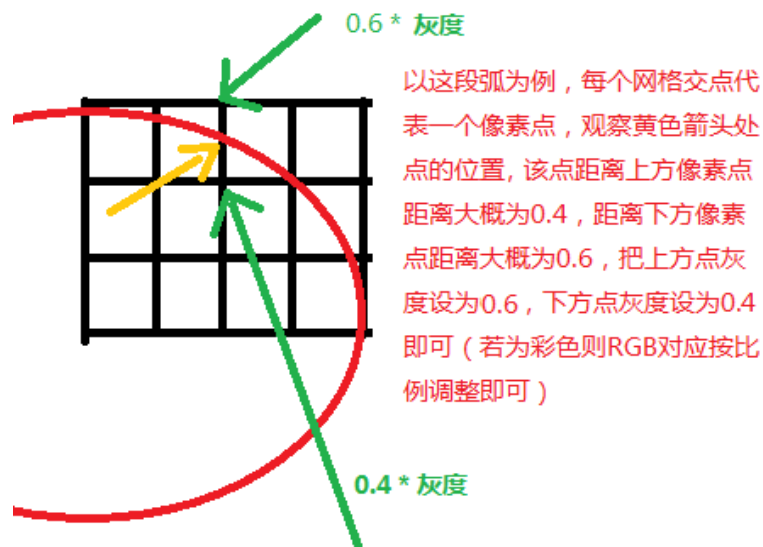
对应于程序中的AntiAliasingDrawCircle(x0, y0, r, color)函数，实现了反走样的画出圆心在(x0, y0)半径为r颜色为color的圆。通过观察Photoshop生成的圆，猜想出了一种算法，实际效果还不错。算法简介见下页图。实际上该算法和区域填充算法、Kernel算法的思想有很大类似，前两个算法基于面积进行反走样，比如面积占了这个像素40%，则灰度设为40%，该算法基于长度进行反走样，比如长度距离这个像素40%，则灰度设为（1-40%）。仅通过一维长度计算，实现简洁，避免了复杂的面积计算或分格计算，效率较高，效果也不错。

### 4、SSAA算法（仅限opencv实现的程序）

没有专门对应的函数，因为实现比较简单，对应于程序中main()函数生成circle\_4的部分，先以4倍分辨率放大再缩小得到，使用了cv::resize()函数。但是该算法效果并不好（可能实现方式有一些问题）。

### 5、Bresenham画线算法

对应于程序中的DrawLine(x0, y0, x1, y1, color)函数，实现了未进行反走样的画出从(x0, y0)到(x1, y1)的直线。实现了课本23页的直线算法，但未进行反走样处理，故不进行展示。实际上，我将自己提出的反走样算法思想告诉一些同学后，有的同学采用该算法实现了直线的反走样，效果也非常好。



#### 四、一点疑问

该部分不属于程序设计部分，但我发现如果使用opencv的imwrite函数保存jpg格式文件会自动进行反走样！而保存为png格式的文件则不会，参加下面的两张图，可以清晰地看出填充轮廓的边缘部分自动进行了反走样处理（可以和生成的circle\_2.png对比观察）。在阅读了jpg的编码方式后，我猜想这可能是因为jpg复杂的编码形式造成的，还待进一步探索。

