

《计算机图形学》4 月报告

161220062 李昇程 161220062@smail.nju.edu.cn

2019 年 4 月 15 日

1 综述

完成的内容 本月我启动了大作业项目，目前完成了命令行接口部分的参数分析和错误处理任务，实现一个对用户相对友好的命令行交互式前端；同时开始实现基本功能，目前完成了用DDA画线(对drawLine函数的支持)，并且进行了简单的性能测试。

计划将要完成的任务 按照课上进度完成其他图元的生成算法；启动图形界面的设计，并将命令行部分的算法迁移到图形界面上去。

2 算法介绍

2.1 DDA

Digital Differential Analyzer 数字差分分析，每次根据斜率在一个坐标轴上以单位间隔对线段取样($\Delta x = 1$ 或 $\Delta y = 1$)，据其计算另一个坐标轴上最靠近线段路径对应的整数值，以此作为最后生成的线元的整点[1]。记斜率为 m ，若 $|m| > 1$ ，则对于间隔 $\Delta x = 1$ ，顺序 y 值可计算为：

$$y_{k+1} = y_k + m \quad (1)$$

对 $|m| \leq 1$ 类似有

$$x_{k+1} = x_k + \frac{1}{m} \quad (2)$$

然而，实际实现时，可能存在直线垂直于 x 轴或 y 轴的情况，我采用的办法是特殊处理，保证斜率取值合法再进行DDA，否则只用在某一条平行线上赋值，这样的话存在高效存取方法实现。

3 系统介绍

总体架构 其分为命令行界面和图形界面两部分，分别位于两个文件夹下，命令行界面主要由三部分构成：

main函数 接收用户输入，显示相关信息；

parser部分 作为系统前端，分析用户输入，进行基础的词法语法分析，拒绝非法输入，并将合法输入传到后端进行处理；

panel部分 其为画板，作为系统后端，完成各种图形操作，并按照用户要求显示与保存图片。其中panel部分主要包含使用的算法，计划当图形界面开始设计的时候移植过去，以减少重复工作。目前图形界面部分未开始设计。

另外由于其需要支持clip操作，所以图元需要设计合理的数据结构来存储。在这里我是用字典来存放图元，以支持clip操作中对该区域的分析，并在分析结束后重新画图。

实现方法 命令行界面(CLI)部分我选用python来实现，与画图相关的矩阵操作与保存图片等我选用opencv来实现(python里对应的包为cv2和numpy)。GUI部分还没想好要怎么实现。

运行 命令行界面部分在commandLine文件夹下python main.py或者./main.py即可运行。可使用./main.py input.txt 或者 ./main.py j input.txt 读取测试文件输入运行，查看效果。

4 总结

目前由于课上算法只讲到直线的生成算法，所以只完成了些系统的架构部分的设计，但是预计下个月内所有算法都会陆陆续续讲到，所以下个月应该会完成较多的算法部分。目前应用了部分编译原理课上了解的前后端的思想，感觉设计出来的命令行界面的交互性和容错性都比以前有了较大的提升，并且使整个设计更加结构化。

总体来说感觉用python实现十分畅快，但还是发现python的opencv实际上有很多坑点，比如其图片的存储顺序和我们的认知顺序相反(从左上到右下)，而且其中可能有部分性能问题是由于低效地矩阵访问导致，而非算法导致，这种问题可能导致性能分析和提升时候出现问题，希望能在以后的设计中尽快解决，并使整个系统更加完美。

参考文献

- [1] 计算机图形学教程 孙正兴主编 周良，郑洪源，谢强编著