

《计算机图形学》5 月报告

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

2019 年 5 月 26 日

1 综述

完成的内容 本月我继续进行大作业项目，目前完成了Bresenham和Line画线算法，并且实现了 Line 和 Polygon 两种图元，使其支持Bresenham和DDA生成，设计完成了更为合理的模块架构，并进行了正确性测试。

计划将要完成的任务 抓紧时间完成其他图元的算法和裁剪算法，并将图元部分的代码移植到GUI上去。

2 算法介绍

2.1 线画图元介绍

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。

Bresenham算法 详细算法课本上有，所以在此用我的语言归纳总结一下，其针对DDA中可能出现的累积误差的问题，选择使用决策函数来在每一点决定下一个点时的选择，并通过动态更新决策函数有力的消除了累计误差的问题，而且其全部使用整数计算，开销比全部使用浮点数的DDA大大减少。

具体的，对于第一象限的直线，斜率小于1时，在点 (x_k, y_k) 处， p_k 的更新公式为

$$p_{k+1} = \begin{cases} p_k + 2\Delta y - 2\Delta x & p_k > 0 \\ p_k + 2\Delta y & p_k < 0 \end{cases} \quad (3)$$

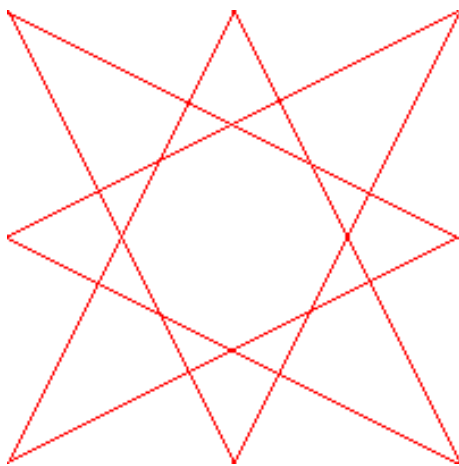


图 1: Bresenham 测试图样

且初始化其为

$$p_0 = 2\Delta y - \Delta x \quad (4)$$

而对于其他象限以及其他斜率的直线，可对应变换 x , y , p_k 的相应项完成。由于象限间的对称性，在我实现中通过规定 y 的相对关系将需要处理的情况减为4种，再通过对称性仅处理两种情况即可。

测试 由于Bresenham需要处理8种情况，所以应该使测试用例覆盖8个朝向的边保证正确地覆盖所有可能。我使用了图1中的八条线来检测，其结果清晰可见，并且能很快的显示出错误的可能原因，加速了我的debug过程。上图也用于检测DDA的实现的正确性。

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。

图元接口 在一个月学习后，以及受到sklearn中类设计的启发，我意识到实际上可以将后端的功能进一步分离，只要设计好每种图元自身的数据结构，并使其支持draw, clip, rotate等操作，就可以直接通过panel保存图元字典来直接调用，实现更好的数据封装。目前相关实现在commandLine/Line.py 和 commandLine/Polygon.py中。

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部分拟采用PyQt来实现，方便直接调用在命令行界面下的功能代码。

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 < input.txt 读取测试文件输入运行，查看效果。

另外执行./main.py后进入交互界面，可以通过show指令展示当前的画布，并使panel同步图元信息到画布上。

4 总结

目前课上相关部分知识已经上完，由于最近作业较多这个的进展比较慢，而且发现好的测试用例能暴露出很多之前没有意识到的问题，希望能在接下来的1个月内投入更多的时间，成功完成这个项目。

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