

光栅图形学 作业报告

作者

魏家栋，计72，学号2017011445

实现内容

- Bresenham算法画线
- 中点法画圆
- Bresenham算法画多边形
- 扫描线算法多边形的扫描转换
- 扫描线算法区域填充
- 各个算法的测试

实现思路

- Bresenham算法画线

参照课本P23算法程序2.4。Bresenham算法的思想是，对误差项 d 进行增量计算，通过 d 判断下一个像素应为右方像素还是右上方像素。

本程序采用整数替代浮点数以避免除法，提高程序执行效率并消除浮点误差。值得注意的是，课本算法程序2.4只适用于从左端点到右端点，而且斜率 k 满足 $|k| \leq 1$ 的直线的绘制。若给出的是直线的两个端点，则需要判断端点的左右；若直线斜率 k 满足 $|k| > 1$ ，则需要交换 x 与 y 的地位，即 y 每增长1， d 的值相应递增 $1/k$ 。

- 中点法画圆

参照课本P24算法程序2.5。中点法画圆的思想是，构造判别式 $d=F(M)$ ，其中 F 为圆的方程， M 为右方像素与右下方像素的中点。对 d 进行增量计算，通过 d 判断下一个像素应为右方像素还是右下方像素。再利用圆的八对称性，只需扫描转换 $1/8$ 圆弧。

本程序采用整数替代浮点数，以提高程序执行效率并消除浮点误差。为了能够绘制圆心在任意一点的圆，本程序先类似于课本算法程序2.5得到圆心在原点的圆上各点坐标，再进行一次平移得到圆心在给定圆心的圆上各点坐标并进行绘制。

- Bresenham算法画多边形

给定顶点序列，对于每对相邻的顶点，调用Bresenham画线算法绘制一条线段，连起来就得到一个多边形。如果输入要求多边形闭合，则在第一个顶点和最后一个顶点之间，也调用Bresenham画线算法绘制一条线段。

- 扫描线算法多边形的扫描转换

参照课本P27算法程序2.6。扫描转换指的是，把多边形的顶点表示（边）转换为点阵表示（面）。扫描线算法的思想是，对每一条扫描线，计算扫描线与多边形边的交点，得到扫描线在多边形内的区段，并填充为指定的颜色。

为了减少扫描线与多边形边界求交、提高效率，采用活性边表（active_edge_table）记录与当前扫描线相交的多边形的边。该活性边表为一个链表，链表的结点存储3项内容：当前扫描线与多边形边的交点的x坐标，当前扫描线到下一条扫描线与多边形边的交点的x坐标的增量delta_x，多边形的边的最高扫描线号y_max。为了方便活性边表的建立与更新，为每一条扫描线采用一个新边表（new_edge_table）存放最低扫描线为该扫描线的多边形的边。

对于扫描线在多边形内的区段，填色包括区段的两端点。对于扫描线与多边形顶点相交的特殊情况，若扫描线与多边形相交的边分处扫描线的两侧，则计一个交点；若扫描线与多边形相交的边处于扫描线的一侧，若为高侧则计两个交点（填色），若为低侧则计零个交点（不填色）。对于扫描线与多边形边相交的特殊情况，若多边形该边相邻的两边分处扫描线的两侧，则计三个交点，即该边的两个端点，其中一个端点计两次；否则计两个交点，即该边的两个端点。

- 扫描线算法区域填充

参照课本P30算法程序2.10。区域填充指的是，将区域内一点赋予指定的颜色，然后将该颜色扩展到整个区域。扫描线算法的思想是，利用一种序列结构保存所有已扫描的扫描线上的种子点，每次从序列中取出一个种子点，填充该扫描线上种子点所在的区段，然后确定该扫描线上下两条扫描线上的种子点，并保存到序列。

本程序中用栈保存所有已知且未被填充的种子点。与课本算法程序2.10有所不同的是，算法程序2.10把每一区间的最右像素作为种子点入栈，本程序把每一区间的最左像素（或者是(xl, y+1)）作为种子点入栈。

- 各个算法的测试

对于每个算法，考虑各类情况，比如Bresenham画线算法，考虑直线的斜率k为0、为正、为负、为无穷、 $|k|=1$ 、 $|k|>1$ 的各种情况，绘制出相应的图形，并且与opencv内置的函数绘制的图形相对比，观察结果是否一致。

遇到的问题 and 解决方法

- 对算法的适用范围不太清楚

比如Bresenham画线算法，课本P23算法程序2.4只适用于从左端点到右端点，而且斜率k满足 $|k| \leq 1$ 的直线的绘制，而我刚开始并没有意识到这一点，导致了绘制图形出现问题。事实上对于其他情况，只需稍微改动代码即可正确地绘制。这说明算法的适用范围是确保算法正确性的重要一点。

- 特殊情况的判断

在多边形的扫描转换扫描线算法中，需要对扫描线与多边形顶点相交、扫描线与多边形边相交等特殊情况进行判断并处理。这些特殊情况非常容易出错，导致绘制结果错误。事实上我觉得课本在这些特殊情况的处理上并不太正确，在开始时甚至有些误导我。比如课本在完成配对、进行填色时，认为区间填色应该左闭右开，但我认为这会导致边界一个填色一个不填色，没有道理，而且在扫描线与多边形只有一个交点时，这会导致多边形这一点不填色，从而导致多边形在这一点“断开”，因此我改为对左闭右闭区间进行填色。再比如课本认为扫描线与多边形边界重合时，计一个交点，但这可能会导致需要配对的交点数为奇数，显然并不正确，我进行了分析后认为计交点数应该与这条边的两条邻边位于扫描线两侧还是同侧有关，修复了这一特殊情况。