

# 《计算机图形学》9 月报告

181860155 朱晓晴 [heloize@126.com](mailto:heloize@126.com)

2020 年 9 月 30 日

## 1 综述

在 9 月提交中, 主要完善了 `cg_cli.py` 和 `cg_algorithms.py` 两个模块。在 `cg_cli.py` 中, 补充了对 `drawPolygon`、`drawEllipse` 等指令的解析。在 `cg_algorithms.py` 中, 完成了绘制线段的所有算法 (即 `draw_line` 函数) 和绘制椭圆的算法 (即 `draw_ellipse` 函数)。

## 2 算法介绍

### 2.1 算法原理

#### 2.1.1 绘制线段

**要求:** 根据给定两点  $(x_0, y_0)$  和  $(x_1, y_1)$  绘制线段。

绘制线段共需完成 3 种算法: Naive 算法、DDA 算法和 Bresenham 算法。其中 Naive 算法已提供, 9 月提交中完成了 DDA 算法和 Bresenham 算法。

**DDA 算法 [1]**

斜率  $m = \frac{y_1 - y_0}{x_1 - x_0}$ , 若  $|m| \leq 1$ , 以  $x_0 < x_1$  为例进行说明。以单位间隔 ( $\Delta x = 1$ ) 对  $x$  进行采样, 并计算对应的  $y$  值:

$$y_{k+1} = y_k + m \quad (k = 0, 1, \dots)$$

若  $|m| > 1$ , 以  $y_0 < y_1$  为例进行说明。以单位间隔 ( $\Delta y = 1$ ) 对  $y$  进行采样, 并计算对应的  $x$  值:

$$x_{k+1} = x_k + \frac{1}{m} \quad (k = 0, 1, \dots)$$

**Bresenham 算法 [1]**

将  $(x_0, y_0)$  作为第一个点, 计算决策参数的第一个值  $p_0 = 2\Delta y - \Delta x$  ( $\Delta x = x_1 - x_0$ ,  $\Delta y = y_1 - y_0$ )。

若  $|m| \leq 1$ , 以  $m > 0$  且  $x_0 < x_1$  的情况为例进行说明。在每个  $x_k$  处进行检测:

$p_k < 0$ , 下一个点为  $(x_{k+1}, y_k)$ , 下一个决策参数  $p_{k+1} = p_k + 2\Delta y$ ;

$p_k \geq 0$ , 下一个点为  $(x_{k+1}, y_{k+1})$ , 下一个决策参数  $p_{k+1} = p_k + 2\Delta y - 2\Delta x$ 。

上述步骤共进行  $\Delta x$  次。

若  $|m| > 1$ , 以  $m > 0$  且  $y_0 < y_1$  的情况为例进行说明。在每个  $y_k$  处进行检测:

$p_k < 0$ , 下一个点为  $(x_k, y_{k+1})$ , 下一个决策参数  $p_{k+1} = p_k + 2\Delta x$ ;

$p_k \geq 0$ , 下一个点为  $(x_{k+1}, y_{k+1})$ , 下一个决策参数  $p_{k+1} = p_k + 2\Delta x - 2\Delta y$ 。

上述步骤共进行  $\Delta y$  次。

### 2.1.2 绘制椭圆

**要求：**根据给定的椭圆矩形包围框左上角坐标  $(x_0, y_0)$  和右下角坐标  $(x_1, y_1)$  绘制椭圆。

**中点圆生成算法 [1]**

计算出椭圆中心  $(x_c, y_c)$ ，长短轴径  $r_x$  和  $r_y$ 。

(1) 区域 1 中 ( $|$  切线斜率  $\leq 1$ )

计算得中心在原点的椭圆上的第一个点  $(0, r_y)$ ，计算区域 1 决策参数的初值  $p1_0 = r_y^2 - r_x^2 r_y + \frac{r_x^2}{4}$ 。在区域 1 每个  $x_k$  处，对决策参数进行检测：

$p1_k < 0$ ，下一个点为  $(x_{k+1}, y_k)$ ，下一个决策参数  $p1_{k+1} = p1_k + 2r_y^2 x_k + 3r_y^2$ ；

$p1_k \geq 0$ ，下一个点为  $(x_{k+1}, y_k - 1)$ ，下一个决策参数  $p1_{k+1} = p1_k + 2r_y^2 x_k - 2r_x^2 y_k + 2r_x^2 + 3r_y^2$ 。

循环直到  $2r_y^2 x \geq 2r_x^2 xy$ 。

(2) 区域 2 中 ( $|$  切线斜率  $> 1$ )

取第一个点  $(r_x, 0)$ ，计算区域 2 决策参数的初值  $p2_0 = r_x^2 - r_x r_y^2 + \frac{r_y^2}{4}$ 。在区域 2 每个  $y_k$  处，对决策参数进行检测：

$p2_k \leq 0$ ，下一个点为  $(x_k, y_{k+1})$ ，下一个决策参数  $p2_{k+1} = p2_k + 2r_x^2 y_k + 3r_x^2$ ；

$p2_k > 0$ ，下一个点为  $(x_k - 1, y_{k+1})$ ，下一个决策参数  $p2_{k+1} = p2_k + 2r_x^2 y_k - 2r_y^2 x_k + 2r_y^2 + 3r_x^2$ 。

循环直到  $2r_y^2 x < 2r_x^2 xy$ 。

最后，计算出其他三个象限中的点，并将所有的点平移到中心为  $(x_c, y_c)$  的椭圆轨迹上。

### 2.1.3 拟采用算法

重置画布、保存画布和设置画笔颜色的功能在 demo 中已经实现，9 月提交则完成了绘制线段和绘制椭圆的功能，剩余绘制多边形、绘制曲线、图元平移、图元旋转、图元缩放、裁剪线段等功能待后续完成。

其中，绘制多边形将采用 DDA 算法和 Bresenham 算法，绘制曲线将采用 Bezier 算法和 B-spline 算法，裁剪线段将采用 Cohen-Sutherland 算法和 Liang-Barsky 算法。

## 2.2 对比分析

### 2.2.1 绘制线段

如图 1 所示，向绘制线段的 3 种算法给定参数。其中，Naive 算法和 DDA 算法获得的参数完全一致，Bresenham 算法获得的参数为另两种算法参数向下平移 10 个像素。

绘制结果如图 2 所示，从上至下三条线段分别由 Naive 算法、DDA 算法和 Bresenham 算法生成。根据图 2，易观察得 Naive 算法的结果与另两种算法的结果差异较大。Naive 算法不区分斜率绝对值小于 1 和大于 1 的情况，同时浮点增量的连续叠加中取整误差的积累会使长线段所计算的像素位置偏离实际线段 [1]。而 DDA 算法和 Bresenham 算法优化了上述两个问题，计算结果与实际线段更接近。因此，图 2 中 DDA 算法和 Bresenham 算法结果无显著区别，而 Naive 算法的结果较前两者差异较大。

```

5  setColor 255 0 0
6  drawLine 600 200 400 400 250 Naive
7  setColor 55 100 255
8  drawLine 610 200 400 400 250 DDA
9  setColor 55 0 255
10 drawLine 666 200 390 400 240 Bresenham

```

图 1: 线段绘制指令

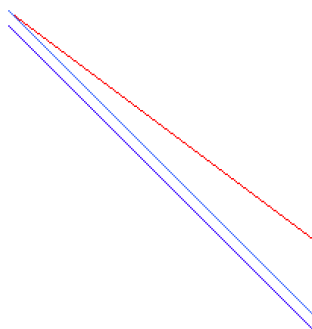


图 2: 线段绘制算法比较

### 2.2.2 绘制椭圆

绘制椭圆采用了中点圆生成算法，根据课本对中点圆生成算法的定义，当取得区域 1 中的最后一个点，不妨设其为  $(x_a, y_a)$ ，应根据  $(x_a, y_a)$  计算出区域 2 中决策参数的初值  $p_{2_0}$ ，然后以单位间隔对  $y$  取样，直到取到点  $(r_x, 0)$ 。课本定义的中点圆生成算法计算出的结果如图 3 所示：

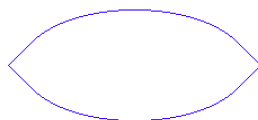


图 3: 课本中点圆生成算法结果

分析结果错误的原因，区域 2 以  $(x_a, y_a)$  为起点，并不能确保最后一个点取到  $(r_x, 0)$ 。参考相关博客 [2] 后，将区域 2 的算法修改为：以  $(r_x, 0)$  为起点，计算决策参数的初值  $p_{2_0}$ ，并按照 2.1.2 中的算法计算，直到切线斜率绝对值大于等于 1。上述算法计算出的结果如图 4 所示：

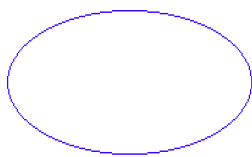


图 4: 优化后的中点圆生成算法结果

## 3 系统介绍

### 3.1 交互逻辑

目前，系统沿用 demo 的交互逻辑，即命令行和可视化界面两种运行方式。

以命令行方式运行时，系统逐行读取指令文件，解析指令并调用 `cg_algorithms.py` 模块中的相关算法。

以可视化界面运行时，通过鼠标事件获取所需参数，调用 `cg_algorithms.py` 模块中的相关算法，再以可视化方式直接呈现结果。

### 3.2 设计思路

后续，在完成要求的功能后，将着重优化图形界面。目前，计划在图形界面中做出以下修改：

- (1) 对用户隐藏图元序号，直接通过鼠标操作选中图元，并对图元进行编辑；
- (2) 优化颜色设置方式，通过显示调色盘简化用户设置颜色的过程；
- (3) 添加可绘制的图元类型，如填充图元、虚线等。

## 4 总结

9 月提交中，运用了目前已讲到图形学基础知识，并自学了解了绘制图元的若干算法。通过实现绘制线段和绘制椭圆的算法，加深了对图形显示原理的理解。但目前对图形界面的理解较浅，尚未基于 demo 的图形界面做出显著改动，后续将着重优化图形界面，使得整个图形学系统功能更为强大，使用更为简易。

## 参考文献

- [1] 孙正兴等. 计算机图形学教程 [M]. 机械工业出版社, 2006.
- [2] cumtv80668. 中点画椭圆算法. CSDN, 2020.  
链接:<https://blog.csdn.net/cumtv80668/article/details/107792162>