

图形学大作业系统报告

陈劲源 (161240004)
sy_chen@smail.nju.edu.cn

April 13, 2019

1 综述

2 算法介绍

绘制曲线 $f(x, y) = 0$ 的基本原则是：当 $\left| \frac{dy}{dx} \right|_{(x_0, y_0)} \leq 1$ 时，沿 x 轴递进采样画点；当 $\left| \frac{dy}{dx} \right|_{(x_0, y_0)} > 1$ 时，沿 y 轴递进采样画点。这样可以保证相邻两个绘制点 $(x_i, y_i), (x_{i+1}, y_{i+1})$ 之间满足 $\max\{|x_i - x_{i+1}|, |y_i - y_{i+1}|\} = 1$ 。

2.1 DDA 算法

DDA 算法是利用对曲线微分方程积分的方法来绘制曲线的。DDA 算法通常用于绘制线段、多边形等，但也可用来绘制非线性曲线 [1]。对于直线 $y = kx + b$ ($|k| \leq 1$)，DDA 算法在每次递增 x 时，对 y 增加 k ，并将取整后的值作为当前绘制点。利用 DDA 算法绘制线段的伪代码如下：

Algorithm 1 DDA 画线算法

Require: 线段的两个端点 $(x_1, y_1), (x_2, y_2)$ 。假定 $x_1 < x_2, |x_2 - x_1| \geq |y_2 - y_1|$ 。

```
1:  $y = y_1, k = \frac{y_2 - y_1}{x_2 - x_1}$ 
2: for  $x = x_1$  to  $x_2$  do
3:   绘制点  $([x], [y])$ 
4:    $y = y + k$ 
5: end for
```

2.2 Bresenham 算法

Bresenham 算法的基本思想是，通过判断下两个绘制点的中点在直线的哪一侧来决定选取哪一个决策点。判断中点在直线哪一侧可以通过维护一个决策变量 Δ 来实现，而决策变量的维护通常可以利用整数的加减法实现 [2]，因此 Bresenham 算法比 DDA 算法更加高效。

对于以 $(x_1, y_1), (x_2, y_2)$ （假设 $x_1 < x_2, y_1 \leq y_2, |x_1 - x_2| \geq |y_1 - y_2|$ ）为端点的线段，它的直线方程为

$$(y - y_1)(x_2 - x_1) = (y_2 - y_1)(x - x_1)$$

故可取决策变量 $\Delta(x, y) = 2[(y_2 - y_1)(x - x_1) - (y - y_1)(x_2 - x_1)]$ ，并根据 $\Delta(x_i + 1, y_i + 0.5)$ 的符号决定绘制点。当决策变量为正时，递增 y ，否则不递增 y 。决策变量可用以下方式维护：

$$\Delta(x_i + 1, y_i + 0.5) = \Delta(x_i, y_i) + 2\Delta y - \Delta x$$

$$\Delta(x_i + 1, y_i + 1) = \Delta(x_i + 1, y_i + 0.5) - \Delta x$$

$$\Delta(x_i + 1, y_i) = \Delta(x_i + 1, y_i + 0.5) + \Delta x$$

Bresenham 算法的伪代码如下

Algorithm 2 Bresenham 画线算法

Require: 线段的两个端点 $(x_1, y_1), (x_2, y_2)$ 。假定 $x_1 < x_2, y_1 \leq y_2, |x_2 - x_1| \geq |y_2 - y_1|$ 。

```
1:  $y = y_1, \Delta x = x_2 - x_1, \Delta y = y_2 - y_1$ 
2:  $\Delta = -\Delta x$ 
3: for  $x = x_1$  to  $x_2$  do
4:   if  $\Delta \geq 0$  then
5:      $y = y + 1, \Delta = \Delta - \Delta x$ 
6:   else
7:      $\Delta = \Delta + \Delta x$ 
8:   end if
9:   绘制点  $(x, y)$ 
10:   $\Delta = \Delta + 2\Delta y - \Delta x$ 
11: end for
```

2.3 ...

...

3 系统介绍

4 总结

...

参考文献

- [1] Wikipedia. Digital differential analyzer (graphics algorithm) — Wikipedia, the free encyclopedia, 2019. [Online; accessed 12-April-2019].
- [2] Wikipedia. Bresenham's line algorithm — Wikipedia, the free encyclopedia, 2019. [Online; accessed 12-April-2019].

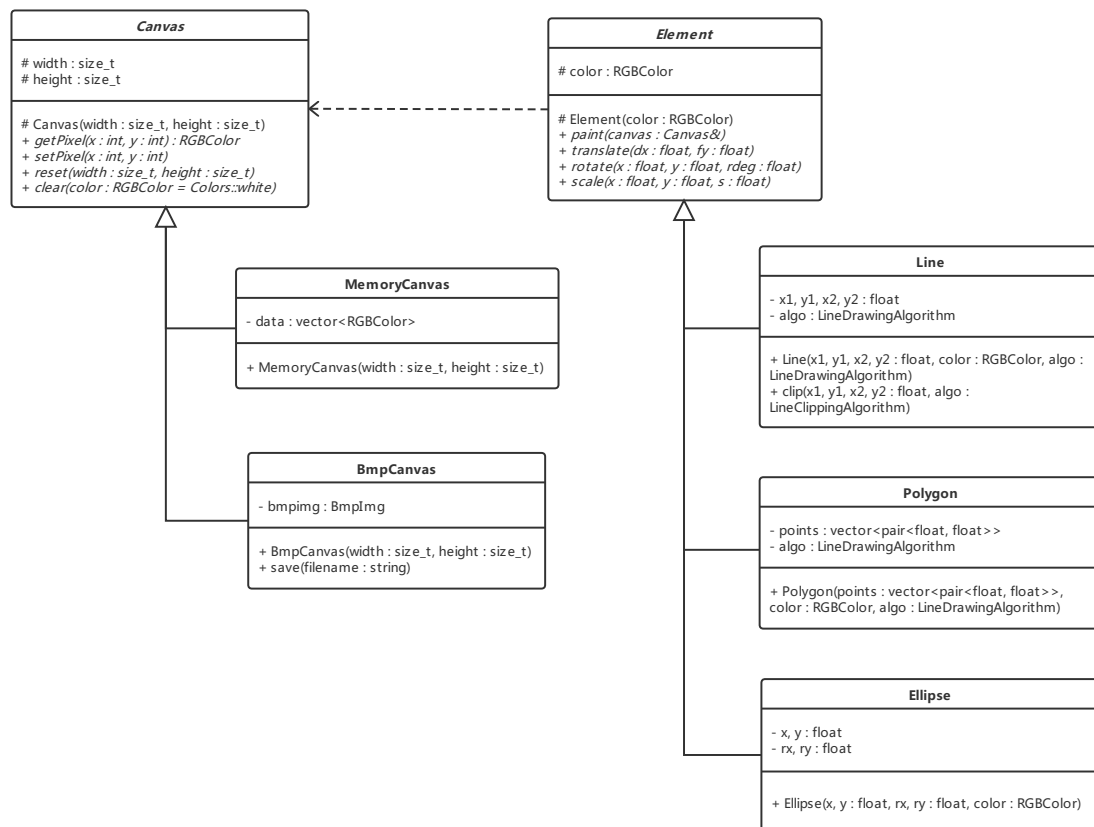


Figure 1: 系统的 UML 类图