Lab01

- Lab01
  - 文件解释
  - 实验原理
    - Bresenham椭圆画法
    - 旋转椭圆
  - 实验效果
    - 无旋转
    - 旋转

文件解释

ppm_rotate.cpp是通过直接坐标变化实现椭圆旋转的算法实现

ppm_shear.cpp是通过Shear实现旋转椭圆的改进算法实现

如果要查看最好结果，直接使用ppm_shear即可。

实验原理

Bresenham椭圆画法

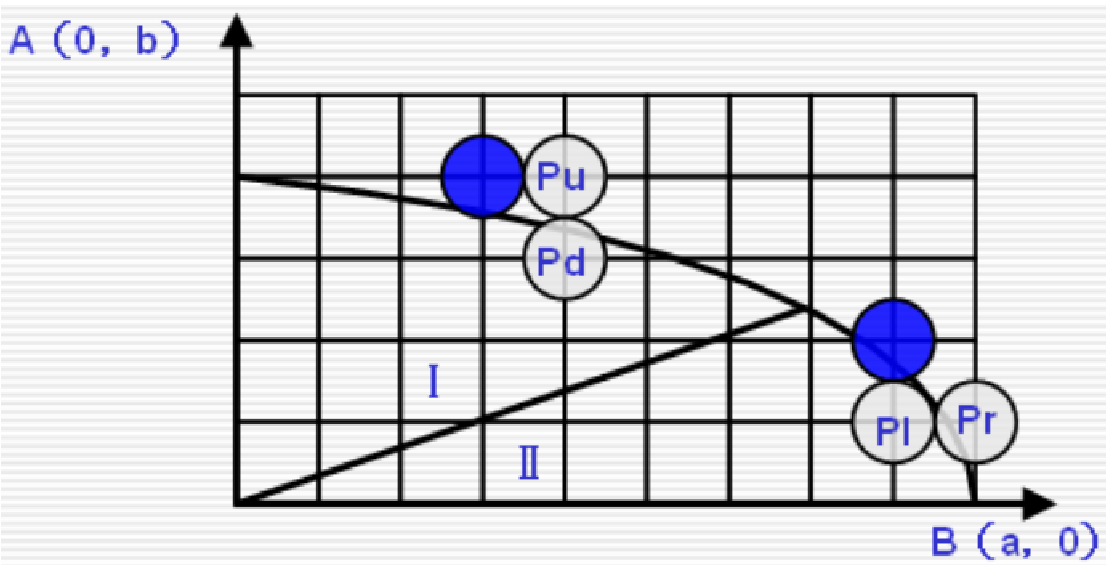
本实验采用中点 Bresenham 的方法绘制椭圆，大体上算法的思想是根据一个确定的点寻找它根据x轴/y轴步进一个确定的步长来确定下一个点，而根据哪个方向步进就需要参考x/y轴方向的变化率。

我们先通过一个标准椭圆来说明算法过程，椭圆的方程如下：

$b^2x^2 + a^2y^2 = a^2b^2$

由于椭圆具有对称性，我们仅需画出1/4椭圆即可根据对称画出整个椭圆，我们以第一象限为例，并且使用(0, b)这个上顶点作为绘制的开始位置。

为确定步进方向，我们对椭圆方程分别求x, y方向上的偏导，并且通过偏导判断出$(\frac{a^2}{(a^2+b^2)^{\frac{1}{2}}}, \frac{b^2}{(a^2+b^2)^{\frac{1}{2}}})$点为x/y轴方向变化率的转折点，也即下图中分界点的上半部分x方向变化大，按照x方向步进，而下半部分y轴方向变化大，按照y方向步进。



对于上半部分，计算其中点偏差判别式，但从$P(x_i, y_i)$步进到x_i+1时，步进后的点的纵坐标只有可能是y_i或者y_i-1，如果我们使用椭圆方程进行两次偏移量的计算，工作量会比较大，而我们可以直接使用这两点的中点，通过判断中点在椭圆内还是椭圆外来判断两点中更加接近的点。所以，偏移量的判别方程式可以写成

$d = b^2(x_i+1)^2 + a^2(y_i-0.5)^2 - a^2b^2$

也就是说当d > 0时，说明中点在椭圆外，则y_i-1更接近椭圆，反之y_i更接近椭圆。

每次都计算一个如此复杂的计算式自然不可取，我们发现此偏移方程式存在一个递推关系，当找到一个点继续找下一个点时，有两种情况

- 若找到的此点是y_i，则需要找的下一个点只能是(x_{i+2}, y_i), (x_{i+2}, y_i-1)同样通过中点计算偏移量:

$$\begin{aligned} d_{(i+1)} &= b^2(x_i+2)^2 + a^2(y_i-0.5)^2 - a^2b^2 \\ &= b^2(x_i+1)^2 + a^2(y_i-0.5)^2 - a^2b^2 + b^2\left(2x_i+3\right) = d_i + b^2\left(2x_i+3\right) \end{aligned}$$

- 若找到的点为y_i-1，则需要找的下一个点只能是(x_{i+2}, y_i-1), (x_{i+2}, y_i-2)

$$\begin{aligned} d_{(i+1)} &= b^2(x_i+2)^2 + a^2(y_i-1.5)^2 - a^2b^2 \\ &= b^2(x+1)^2 + a^2(y_i-05)^2 - a^2b^2 + b^2(2x+3) + a^2\left(-2y_i+2\right) \\ &= d_i + b^2\left(2x_i+3\right) + a^2\left(-2y_i+2\right) \end{aligned}$$

对于下半部分按照y轴步进的推导过程类似，在此不再赘述。

旋转椭圆

在绘制旋转椭圆的方案上，我一开始选用绘制好标准椭圆后，直接使用旋转坐标变化公式绘制出旋转椭圆，但这样做的效果非常差，较多的舍入会产生比较大的偏差。(具体实现代码也附在文件中)

后来进过数学的思考与推导，我想到了**旋转的椭圆始终可以由标准椭圆进过错切(Shear)得到**，所以我使用**先通过Bresenham画一个标准椭圆，再将标准椭圆进行Shear从而得到旋转椭圆**的方式。而我们知道错切一般是接近线性的变化，这样绘制出来的旋转椭圆效果会非常好，并且不同于极坐标直接花旋转椭圆的方法，这种方式保留了Bresenham的大部分优点。

进过数学的推导，我得到以下新椭圆的参数

$$t_0 = \tan^{-1}\left(-\frac{b}{a}\tan\phi\right)$$

$$\begin{aligned} x_0 = a\cos t_0\cos\phi - b\sin t_0\sin\phi \\ y_0 = a\cos t_0\sin\phi + b\sin t_0\cos\phi \end{aligned}$$

我们需要先画一个半长轴$a^{'} = |x_0|$和半短轴$b^{'} = \frac{ab}{a^{'}}$的新标准椭圆，再通过以下错切变换绘制出旋转椭圆:

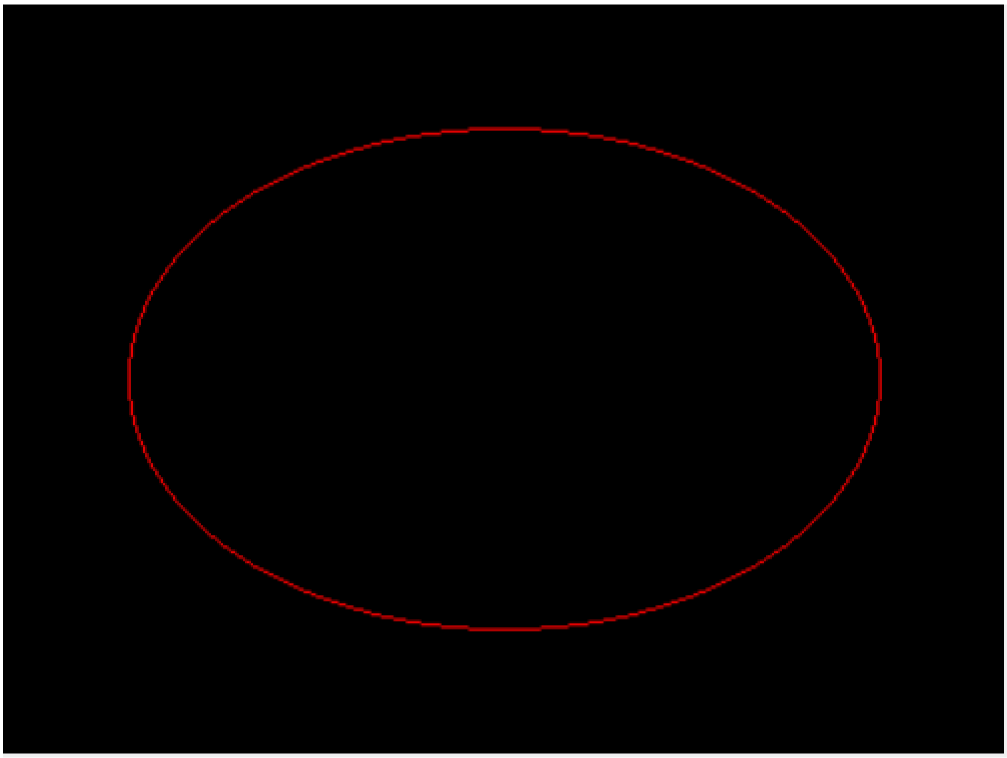
$$\begin{pmatrix} x^{'} \\ y^{'} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \frac{y_0}{x_0} & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$$

实验效果

在图片上，左上角的点为原点，原点向下为y轴正方向，原点向右为x轴正方向，旋转的正方向为顺时针旋转。

无旋转

```
please input the origin (x, y): 200 150
please input the semi-major axis a: 150
please input the semi-minor axis b: 100
please input the rotation angle theta: 0
```



旋转

```
please input the origin (x, y): 200 150
please input the semi-major axis a: 150
please input the semi-minor axis b: 100
please input the rotation angle theta: 30
```