圆、椭圆的扫描转换

ray@mail.buct.edu.cn

2017/10/18

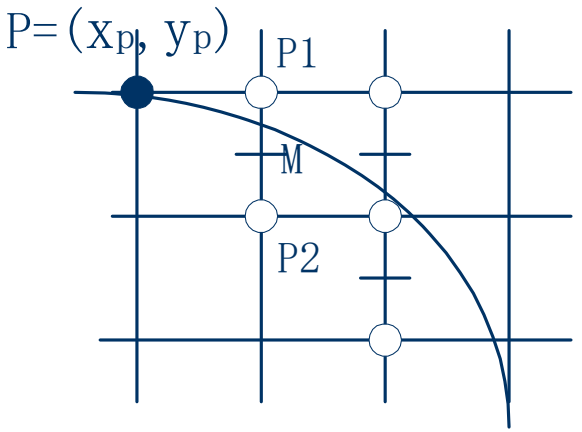
内 容

- 圆的扫描转换
  - 中点画圆法
  - Bresenham画圆法
- 椭圆的扫描转换

圆的性质与中点画圆法

- 圆的特征八对称性
- 中点画圆法(Midpoint circle algorithm)
  - 考虑中心在原点，半径为R的第二个8分圆，构造判别式（圆方程）

$$d = F(M) = F(x_p + 1, y_p - 0.5)$$
$$= (x_p + 1)^2 + (y_p - 0.5)^2 - R^2$$



- 若 d<0，则取P_1为下一象素，而且再下一象素的判别式为

$$d' = F(x_p + 2, y_p - 0.5) = (x_p + 2)^2 + (y_p - 0.5)^2 - R^2 = d + 2x_p + 3$$

- 若d>=0, 则应取P_2为下一象素，而且下一象素的判别式为

$$d' = F(x_p + 2, y_p - 1.5) = (x_p + 2)^2 + (y_p - 1.5)^2 - R^2 = d + 2(x_p - y_p) + 5$$

- 第 一个象素是（0,R），判别式d的初始值为

$$d_0 = F(1, R - 0.5) = 1.25 - R$$

中点画圆法程序代码

```
MidPointCircle(int r int color)
{  int x,y;
     float d;
     x=0; y=r; d=1.25-r;
     circlepoints (x,y,color); //显示圆弧上的八个对称点
     while(x<=y)
    {  if(d<0)    d+=2*x+3;
       else    { d+=2*(x-y)+5;  y--;}
x++;
circlepoints (x,y,color);
   }
}
```

改进(1)

- 为了进一步提高算法的效率，可以将上面的算法中的浮点数改写成整数，仅用整数实现中点画圆法。
  - e=d-0.25代替d
  - e0=1-R

改进(1) 的程序代码

```
MidPointCircle(int r int color)
{  int x,y,d;
   x=0; y=r; d=1-r;
    circlepoints (x,y,color); //显示圆弧上的八个对称点
    while(x<=y)
   {  if(d<0)    d+=2*x+3;
      else    { d+=2*(x-y)+5;  y--;}
x++;
circlepoints (x,y,color);
   }
}
```

改进(2)

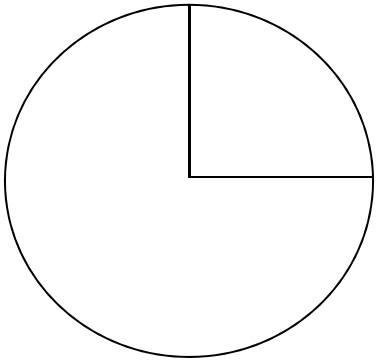
- 判别式d的增量是x、y的线性函数，因此可引进增量的变量deltax和deltay，使算法中不出现乘法。

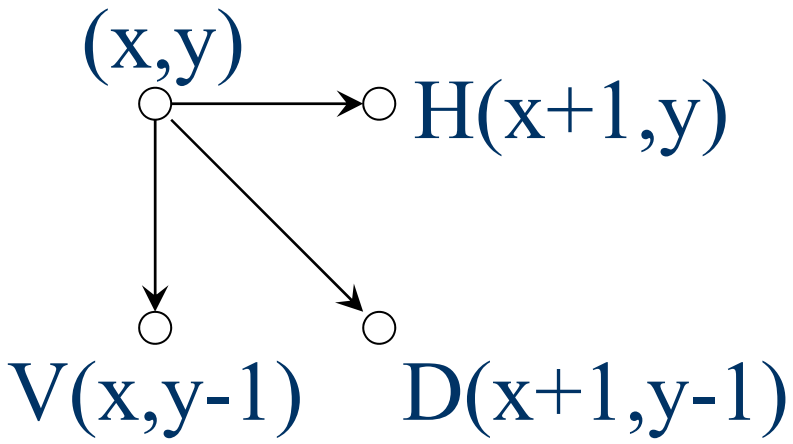
改进(2)后的程序代码

```
MidPointCircle(int r int color)
{  int x,y,d;
   x=0; y=r;
   deltax=3;deltay=2-r-r;d=1-r;
   circlepoints (x,y,color); //显示圆弧上的八个对称点
    while(x<=y)
   {  if(d<0)    {d+=deltax; deltax+=2;}
      else    { d+=deltax+deltay;  deltax+=2; deltay+=2; y--;}
      x++;
      circlepoints (x,y,color);
   }
}
```

Bresenham画圆算法

- 考虑4分圆的情形。
  - $\triangle H=(x+1)^2+y^2-R^2$
  - $\triangle D=(x+1)^2+(y-1)^2-R^2$
  - $\triangle V=x^2+(y-1)^2-R^2$





- 分3种情况：
  1. H在圆外，D、V在圆内；
  2. D在圆上，H在圆外，V在圆内；
  3. H、D在圆外，V在圆内。

- 当△D<0时，δHD≤0，则取H，否则取D；
- 当△D=0时，取D；
- 当△D>0时，δDV≤0，则取D，否则取V；

δHD=| △H |-| △D |
δDV=| △D |-| △V |

- δHD和 δDV均可以通过△D 推算出来
- 可采用增量算法计算△’D ：
  - 如果下一个象素是H：

    （x’,y’)=(x+1,y)

    $\triangle'D=((x+1)+1)^2+(y-1)^2-R^2=\triangle D+2x+1$
  - 如果下一个象素是D：

    $\triangle'D=\triangle D+2x-2y+2$
  - 如果下一个象素是V：

    $\triangle'D=\triangle D-2y+1$

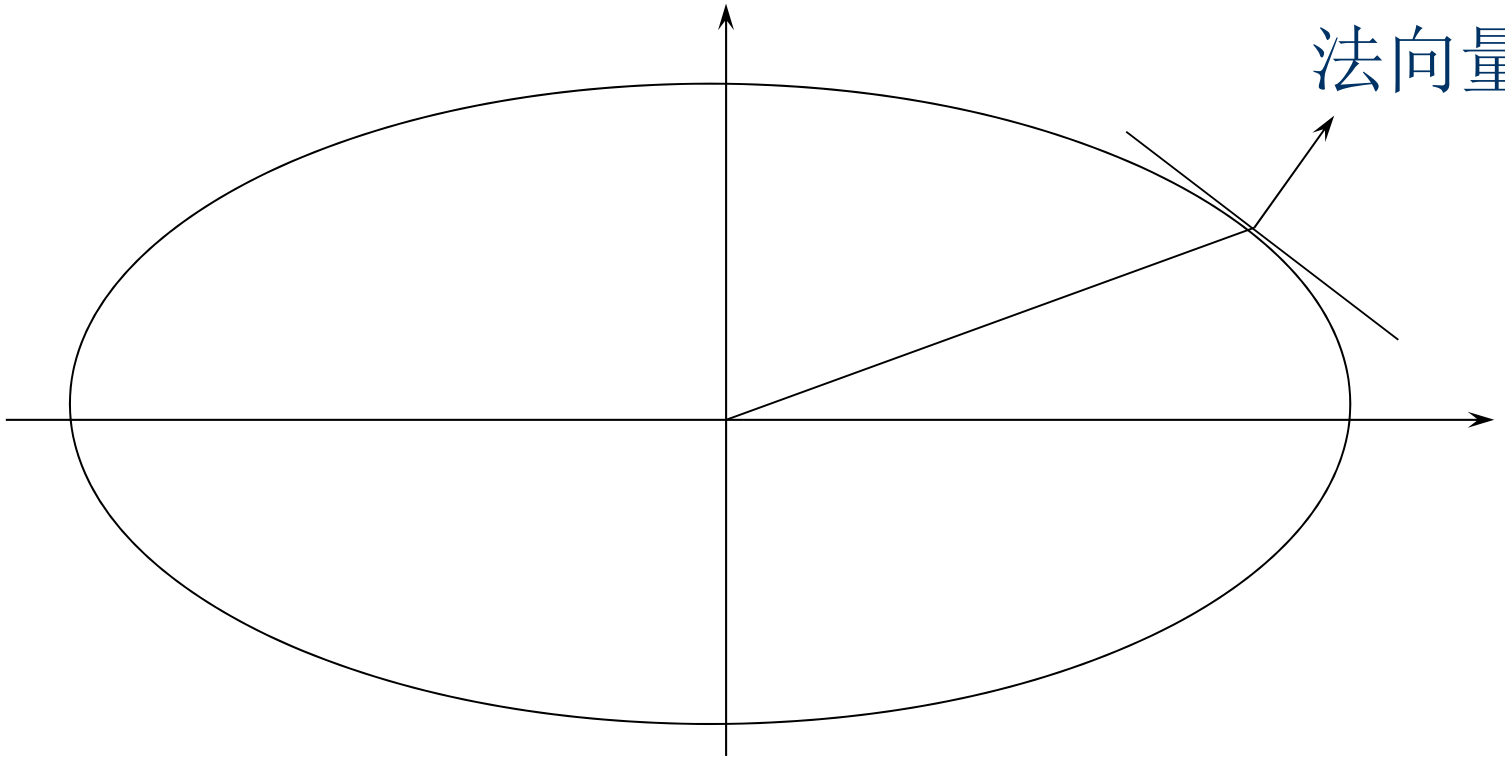
Bresenham画圆法程序代码

```
Bresenham_Circle(r,color)
int r, color;
{
int x,y,delta,delta1,delta2,direction;
x=0;y=r;
delta=2*(1-r);
while(y>=0){
        circlepoints (x,y,color);
        if (delta<0)
        {delta1=2*(delta+y)-1;
        if (delta1<=0) direction=1;
        else direction=2;}
```

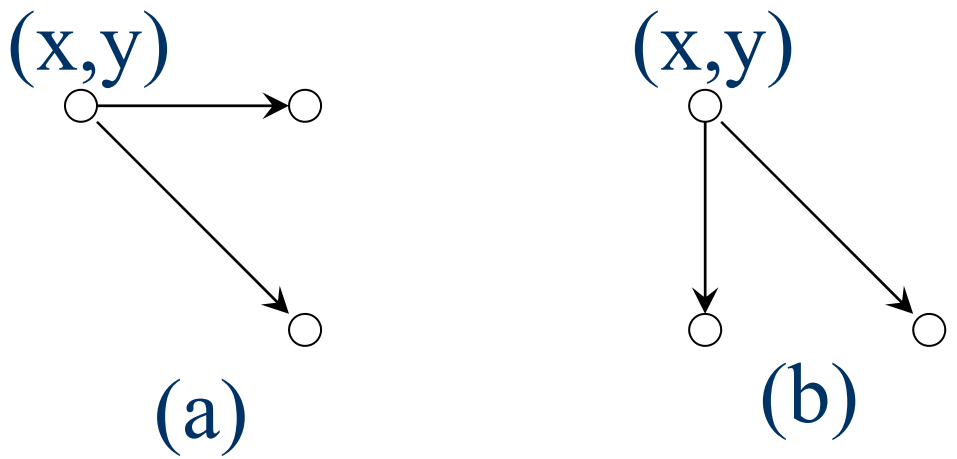
```
        else if (delta>0)
        {delta2=2*(delta-x)-1;
        if(delta2<=0) direction=2;
        else direction =3;}
        else
        direction=2;
        switch (direction)
        {
        case 1: x++;delta+=2*x+1;break;
        case 2:x++;y--;delta+=2*(x-y+1);break;
        case 3:y--;delta+=(-2*y+1);break;
        }/*switch*/
}/*while*/
}/*Bresenham_Circle*/
```

椭圆的扫描转换

$$F(x,y)=b^2x^2+a^2y^2-a^2b^2=0$$



– 法向量的y分量大时适用(a)
– 法向量的x分量大时适用(b)



法向量计算

法向量

$(2b^2(x+1),\ 2a^2(y-0.5))$

开始时

$2b^2(x+1) < 2a^2(y-0.5)$

当

$2b^2(x+1) < 2a^2(y-0.5)$

不再成立时，

说明椭圆弧从上半部分移动到下半部分。

上半部分

下一对候选象素的中点是

(x+1,y-0.5)

判别式

$d_1=F(x+1,y-0.5)$

$= b^2(x+1)^2+a^2(y-0.5)^2-a^2b^2$

增量方法

求△D

d_1<0时

d_1'=F(x+2,y-0.5)

$=d_1+b^2(2x+3)$

d_1>0时

d_1' =F(x+2,y-1.5)

$=d_1+b^2(2x+3)+a^2(-2y+2)$

下半部分

下半部分的终止条件

y=0

下半部分的中点

(x+0.5,y-1)

x和y互换

椭圆弧扫描转换代码

```
MidpointEllipse(a,b,color)
int a,b,color;
{
int x,y;
float d1,d2;
x=0;y=b;
d1=b*b+a*a*(-b+0.25);
drawpixel(x,y,color);
while (b*b*(x+1)<a*a*(y-0.5)){
   if(d1<0){
      d1+=b*b*(2*x+3);
      x++;}
```

```
else {
        d1+=(b*b*(2*x+3)+a*a*(-2*y+2);
        x++;y--;}
    drawpixel(x,y,color);
}/*上半部分*/
d2=sqr(b*(x+0.5)+sqr(a*(y-1))-sqr(a*b);
while(y>0){
    if (d2<0){
        d2+=b*b*(2*x+2)+a*a*(-2*y+3);
        x++;y--;}
```

```
        else {
                d2+=a*a*(-2*y+3);
                y--;}
        drawpixel(x,y,color);
    }/* while */
}/* MidpointEllipse*/
```