# 第 3 章　基本图形的扫描转换

**本章学习目标**

- 扫描转换的基本概念。
- 绘制像素点函数 SetPixel 的用法。
- 直线、圆和椭圆的中点 Bresenham 原理。
- 直线反走样原理。

直线、圆和椭圆是图形设计的最基本图形(图元),尽管在 VC++ 中已有绘制这三种基本图形的现成语句,但是从光栅扫描显示器的显示原理和真实感图形生成技术的需要出发,需要使用像素点函数来绘制这些基本图形。在第 1 章已介绍了光栅扫描显示器是画点设备,不能从像素点阵的一个可编像素点直接画一条直线到达另一个可编址的像素点,只能用靠近这条直线的像素点集来近似地表示这条直线。光栅扫描显示器的绘图过程就是在像素点阵中确定最佳逼近于理想图形的像素点集的过程,这个过程称为“图形的扫描转换”。本章主要使用像素点函数:

```
COLORREF SetPixel(int x,int y,COLORREF crColor)
```

进行基本图形的扫描转换,该函数的三个参数分别为像素点的位置坐标$(x,y)$和像素点的颜色值 crColor。这里请注意屏幕设备坐标系的位置坐标$(x,y)$只能取整数值,而理想图形扫描转换后的坐标值一般为浮点型数值,进行显示图形时需要对计算结果进行四舍五入处理。

## 3.1　直线的扫描转换

直线的扫描转换就是在屏幕像素点阵中用指定颜色点亮最佳逼近于理想直线的像素点集的过程,如图 3-1 所示。对于蓝色的连续理想直线,不能直接在光栅扫描显示器上绘出,只能用点亮了的蓝色像素点集近似地逼近。从图中可以看出,蓝色像素点都是离直线距离最近的像素点。计算机图形学要求直线的绘制速度要快,即尽量使用加减法实现,避免乘、除、开方和三角等复杂运算。有许多算法可以实现直线的扫描转换,如数值微分法(digital differential analyzer,DDA)、逐点比较法等,最著名的是由 Bresenham 提出的中点 Bresenham 算法。

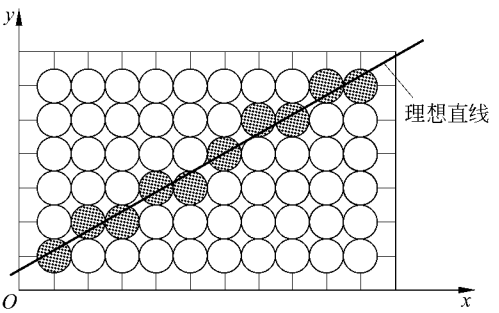



图 3-1　直线的扫描转换

### 3.1.1 算法原理

直线的中点 Bresenham 算法的原理：每次在主位移方向上走一步，另一个方向上走不走步取决于中点偏差判别式的值。

给定理想直线的起点坐标为 $P_0(x_0, y_0)$，终点坐标为 $P_1(x_1, y_1)$，则直线的隐函数方程为

$$F(x,y) = y - kx - b = 0 \tag{3-1}$$

其中，直线的斜率 $k=\dfrac{\Delta y}{\Delta x}=\dfrac{y_1-y_0}{x_1-x_0}$，$\Delta x=x_1-x_0$ 为直线水平方向位移，$\Delta y=y_1-y_0$ 为直线垂直方向位移。

理想直线将平面划分成三个区域：对于直线上的点，$F(x,y)=0$；对于直线上方的点，$F(x,y)>0$；对于直线下方的点，$F(x,y)<0$。假设直线的斜率为 $0\leqslant k\leqslant 1$，则 $\Delta x\geqslant\Delta y$，所以确定 $x$ 方向为主位移方向。按照 Bresenham 原理，$x$ 方向上每次加 1，$y$ 方向上加不加 1 取决于中点偏差判别式的值。

假定直线的当前点是 $P(x_i, y_i)$，沿主位移 $x$ 方向走一步，下一点只能在 $P_u(x_i+1, y_i+1)$ 和 $P_d(x_i+1, y_i)$ 两点中选取。$P_u$ 和 $P_d$ 的中点为 $M(x_i+1, y_i+0.5)$，如图 3-2 所示。显然，若中点 $M$ 在理想直线的下方，则 $P_u$ 点距离直线近，点亮 $P_u$；否则点亮 $P_d$。

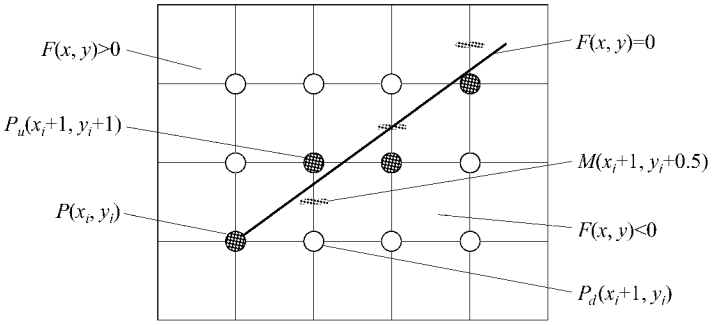



图 3-2　直线中点 Bresenham 算法原理

### 3.1.2 构造中点偏差判别式

从 $P(x_i, y_i)$ 点走第一步后，为了进行下一像素点的选取，需将 $P_u$ 和 $P_d$ 的中点 $M(x_i+1, y_i+0.5)$ 代入隐函数方程，构造中点偏差判别式 $d$。

$$d = F(x_M, y_M) = F(x_i+1, y_i+0.5) = y_i + 0.5 - k(x_i+1) - b \tag{3-2}$$

当 $d<0$ 时，中点 $M$ 在直线的下方，$P_u$ 点离直线距离近，下一像素点应点亮 $P_u$，即 $y$ 方向上走一步；当 $d>0$ 时，中点 $M$ 在直线的上方，$P_d$ 点离直线距离近，下一像素点应点亮 $P_d$，即 $y$ 方向上不走步；当 $d=0$ 时，中点 $M$ 在直线上，$P_u$、$P_d$ 与直线的距离相等，点亮 $P_u$ 或 $P_d$ 均可，约定取 $P_d$，如图 3-2 所示。

因此

$$y_{i+1} = \begin{cases} y_i + 1, & d < 0 \\ y_i, & d \geqslant 0 \end{cases} \tag{3-3}$$

### 3.1.3 递推公式

图 3-2 中，根据当前点 $P(x_i, y_i)$确定下一点是点亮 $P_u$还是点亮 $P_d$时，使用了中点偏差判别式 $d$。为了能够继续判断直线上的每一个点，需要给出中点偏差判别式的递推公式和初始值。

**1. 中点偏差判别式的递推公式**

在主位移 $x$ 方向上已走一步的情况下，考虑沿主位移方向再走一步，应该选择哪个中点代入中点偏差判别式以决定下一步该点亮的像素，如图 3-3 所示，分两种情况讨论。

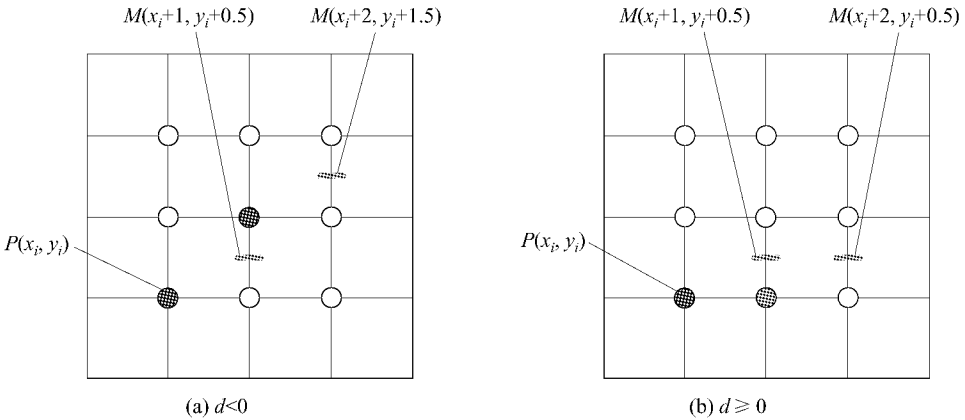



图 3-3　中点偏差判别式的递推

(1) 当 $d<0$ 时，下一步进行判断的中点坐标为 $M(x_i+2, y_i+1.5)$。所以下一步中点偏差判别式为

$$\begin{aligned} d_{i+1} &= F(x_i+2, y_i+1.5) = y_i+1.5-k(x_i+2)-b \\ &= y_i+0.5-k(x_i+1)-b+1-k = d_i+1-k \end{aligned} \tag{3-4}$$

(2) 当 $d \geqslant 0$ 时，下一步的中点坐标为 $M(x_i+2, y_i+0.5)$。所以下一步中点偏差判别式为

$$\begin{aligned} d_{i+1} &= F(x_i+2, y_i+0.5) = y_i+0.5-k(x_i+2)-b \\ &= y_i+0.5-k(x_i+1)-b-k = d_i-k \end{aligned} \tag{3-5}$$

**2. 中点偏差判别式的初始值**

直线的起点坐标为 $P_0(x_0, y_0)$，$x$ 为主位移方向。因此，第一个中点是$(x_0+1, y_0+0.5)$，相应的 $d$ 的初始值为

$$\begin{aligned} d_0 &= F(x_0+1, y_0+0.5) = y_0+0.5-k(x_0+1)-b \\ &= y_0-kx_0-b-k+0.5 \end{aligned}$$

其中，因为$(x_0, y_0)$在直线上，所以 $y_0-kx_0-b=0$，则

$$d_0 = 0.5-k \tag{3-6}$$

## 3.2　圆的扫描转换

圆的扫描转换就是在屏幕像素点阵中用指定颜色点亮最佳逼近于理想圆的像素点集的过程。圆的绘制可以使用简单方程画圆算法和极坐标画圆算法，但这些算法涉及三角运算

和开方运算，效率很低。本节主要讲解仅包含加减操作的顺时针绘制 1/8 圆的中点 Bresenham 算法原理。

### 3.2.1 算法原理

圆心在原点、半径为 $R$ 的圆方程的隐函数表达式为

$$F(x,y) = x^2 + y^2 - R^2 = 0 \tag{3-7}$$

圆将平面划分成三个区域：对于圆上的点，$F(x,y)=0$；对于圆外的点，$F(x,y)>0$；对于圆内的点，$F(x,y)<0$，如图 3-4 所示。

根据坐标系对称性，首先考虑扫描转换 1/4 圆弧。处理第一象限的 1/4 圆弧时，进一步以法矢量的两个分量相等的点把它分为两部分：上半部分Ⅰ和下半部分Ⅱ，如图 3-5 所示。由微分几何可知，该圆上任一点 $P(x,y)$ 处的法矢量为

$$\boldsymbol{N}(x,y) = \frac{\partial F}{\partial x}\boldsymbol{i} + \frac{\partial F}{\partial y}\boldsymbol{j} = 2x\boldsymbol{i} + 2y\boldsymbol{j} \tag{3-8}$$

式中，$\boldsymbol{i}$ 和 $\boldsymbol{j}$ 是沿 $x$ 轴向和 $y$ 轴向的单位矢量。

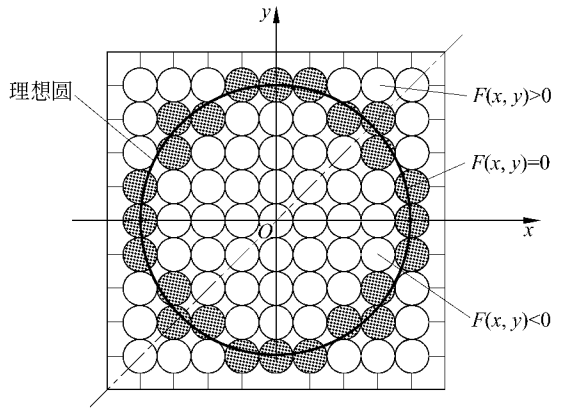



图 3-4　圆的扫描转换

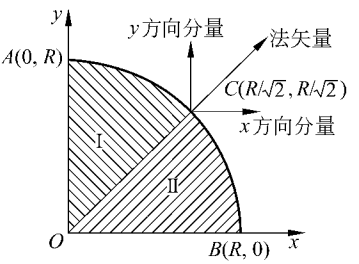



图 3-5　圆的法矢量

在部分Ⅰ的 $AC$ 圆弧段，法矢量的 $x$ 方向分量小于 $y$ 方向分量，斜率 $k$ 处处满足 $|k|<1$，即 $|\Delta x|>|\Delta y|$，所以 $x$ 方向为主位移方向；在 $C$ 点，法矢量的 $x$ 方向分量等于 $y$ 方向分量，斜率 $k$ 满足 $k=-1$，即 $|\Delta x|=|\Delta y|$；在部分Ⅱ的 $CB$ 圆弧段，法矢量的 $x$ 方向分量大于 $y$ 方向分量，斜率 $k$ 处处满足 $|k|>1$，即 $|\Delta y|>|\Delta x|$，所以 $y$ 方向为主位移方向。

圆的中点 Bresenham 算法的原理如下。

在部分Ⅰ：每次在主位移 $x$ 方向上走一步，$y$ 方向上退不退步取决于中点偏差判别式的值。

在部分Ⅱ：每次在主位移 $y$ 方向上退一步，$x$ 方向上走不走步取决于中点偏差判别式的值。

事实上，考虑到圆在第一象限内的对称性，本算法可以进一步简化。因为 $AC$ 段圆弧和 $CB$ 段圆弧以对称轴 $x=y$ 完全对称，如图 3-6 所示，所以可以用 4 条对称轴 $x=0$、$y=0$、$x=y$ 和 $x=-y$ 把圆分成 8 等份。只要绘制出第一象限内的 1/8 圆弧Ⅰ，根据对称性就可绘制出整圆，这称为八分法画圆算法。假定第一象限内的任意点为 $P(x,y)$，可以顺时针确

定另外 7 个点：$P(y,x)$、$P(y,-x)$、$P(x,-y)$、$P(-x,-y)$、$P(-y,-x)$、$P(-y,x)$和$P(-x,y)$。

本算法只考虑图 3-6 所示阴影部分的 45°圆弧，即第一象限内 $x\in\left[0,\frac{R}{\sqrt{2}}\right]$的 1/8 圆弧。此时中点 Bresenham 算法要从$(0,R)\sim(R/\sqrt{2},R/\sqrt{2})$顺时针确定最佳逼近于该段圆弧的像素点集。从前述讨论知道，$x$ 方向为主位移方向，因此中点 Bresenham 算法的原理简化如下：$x$ 方向上每次加 1，$y$ 方向上减不减 1 取决于中点偏差判别式的值。

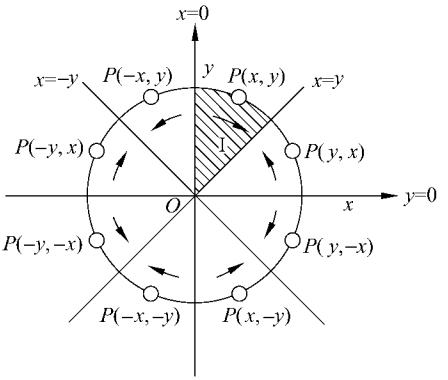



图 3-6　圆的对称性

假定圆当前点是 $P(x_i,y_i)$，下一点只能在 $P_u(x_i+1,y_i)$和 $P_d(x_i+1,y_i-1)$中选取，如图 3-7 所示。$P_u$和$P_d$的中点为$M(x_i+1,y_i-0.5)$，若 $M$ 点在理想圆弧的下方，则 $P_u$点离圆弧近，点亮 $P_u$；否则应点亮 $P_d$。

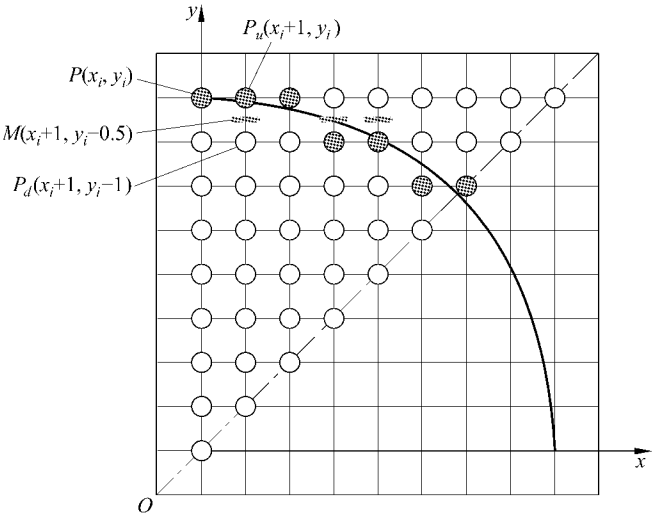



图 3-7　圆中点 Bresenham 算法原理

### 3.2.2　构造中点偏差判别式

从 $P(x_i,y_i)$开始，为了进行下一像素点的选取，需将 $P_u$和 $P_d$的中点 $M(x_i+1,y_i-0.5)$代入隐函数，构造中点偏差判别式

$$d=F(x_M,y_M)=F(x_i+1,y_i-0.5)=(x_i+1)^2+(y_i-0.5)^2-R^2 \qquad (3\text{-}9)$$

当 $d<0$ 时，中点 $M$ 在圆内，下一像素点应点亮 $P_u$，即 $y$ 方向不退步；当 $d>0$ 时，中点 $M$ 在圆外，下一像素点应点亮 $P_d$，即 $y$ 方向退一步；当 $d=0$ 时，中点 $M$ 在圆上，$P_u$、$P_d$和圆的距离相等，点亮 $P_u$或 $P_d$均可，约定取 $P_d$。

因此

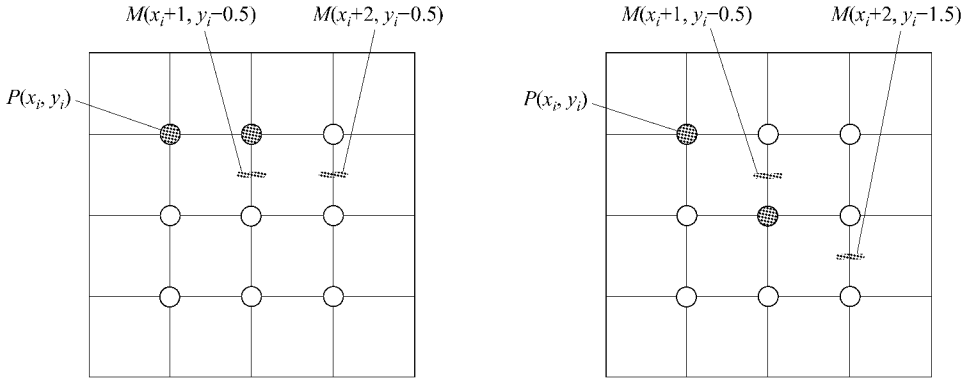
$$y_{i+1}=\begin{cases}y_i, & d<0\\ y_i-1, & d\geqslant 0\end{cases} \tag{3-10}$$

### 3.2.3 递推公式

图 3-7 中，根据当前点 $P(x_i,y_i)$确定了下一点是点亮 $P_u$还是 $P_d$时，使用了中点偏差判别式 $d$。为了能够继续判断 1/8 圆上的每个点，需要给出中点偏差判别式的递推公式和初始值。

**1. 中点偏差判别式的递推公式**

现在如果考虑主位移方向再走一步，应该选择哪个中点代入中点偏差判别式以决定下一步应该点亮的像素，如图 3-8 所示，分两种情况讨论。




(a) $d<0$　　(b) $d\geqslant 0$

图 3-8　中点偏差判别式的递推

(1) 当 $d<0$ 时，下一步的中点坐标为 $M(x_i+2,y_i-0.5)$。所以下一步中点偏差判别式为

$$\begin{aligned}d_{i+1}&=F(x_i+2,y_i-0.5)=(x_i+2)^2+(y_i-0.5)^2-R^2\\&=(x_i+1)^2+(y_i-0.5)^2-R^2+2x_i+3=d_i+2x_i+3\end{aligned} \tag{3-11}$$

(2) 当 $d\geqslant 0$ 时，下一步的中点坐标为 $M(x_i+2,y_i-1.5)$。所以下一步中点偏差判别式为

$$\begin{aligned}d_{i+1}&=F(x_i+2,y_i-1.5)=(x_i+2)^2+(y_i-1.5)^2-R^2\\&=(x_i+1)^2+(y_i-0.5)^2-R^2+2x_i+3+(-2y_i+2)\\&=d_i+2(x_i-y_i)+5\end{aligned} \tag{3-12}$$

**2. 中点偏差判别式的初始值**

圆的起点为 $P_0(0,R)$，$x$ 为主位移方向。因此，第一个中点是$(1,R-0.5)$，对应的 $d$ 的初始值为

$$d_0=F(1,R-0.5)=1+(R-0.5)^2-R^2=1.25-R \tag{3-13}$$

## 3.3　椭圆的扫描转换

椭圆的扫描转换就是在屏幕像素点阵中用指定颜色点亮最佳逼近于理想椭圆像素点集的过程。椭圆是长半轴和短半轴不相等的圆，椭圆的扫描转换和圆的扫描转换有类似之处。

本节主要讲解顺时针绘制 1/4 椭圆的中点 Bresenham 算法原理。

### 3.3.1 算法原理

圆心在原点、长半轴为 $a$、短半轴为 $b$ 的椭圆方程的隐函数表达式为

$$F(x,y) = b^2x^2 + a^2y^2 - a^2b^2 = 0 \tag{3-14}$$

椭圆将平面划分成三个区域：对于椭圆上的点，$F(x,y)=0$；对于椭圆外的点，$F(x,y)>0$；对于椭圆内的点，$F(x,y)<0$，如图 3-9 所示。

考虑到椭圆的对称性，可以用对称轴 $x=0$，$y=0$，把椭圆分成 4 等份。只要绘制出第一象限内的 1/4 椭圆弧，如图 3-10 的阴影部分Ⅰ和Ⅱ所示，根据对称性就可绘制出整个椭圆，这称为四分法绘制椭圆算法。已知第一象限内的点 $P(x,y)$，可以顺时针得到另外 3 个对称点：$P(x,-y)$、$P(-x,-y)$和 $P(-x,y)$。

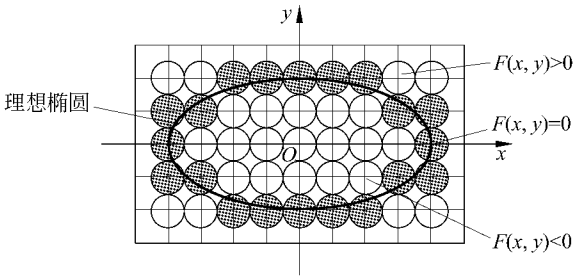



图 3-9　椭圆的扫描转换

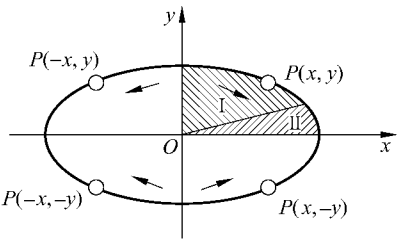



图 3-10　椭圆的对称性

在处理第一象限的 1/4 椭圆弧时，进一步以法矢量两个分量相等的点把它分为两部分：上半部分Ⅰ和下半部分Ⅱ。该椭圆上一点 $P(x,y)$处的法矢量为

$$\boldsymbol{N}(x,y) = \frac{\partial F}{\partial x}\boldsymbol{i} + \frac{\partial F}{\partial y}\boldsymbol{j} = 2b^2x\boldsymbol{i} + 2a^2y\boldsymbol{j} \tag{3-15}$$

式中，$\boldsymbol{i}$ 和 $\boldsymbol{j}$ 是沿 $x$ 轴向和沿 $y$ 轴向的单位矢量。

在图 3-11 所示的部分Ⅰ的 $AC$ 椭圆弧段，法矢量的 $x$ 方向分量小于 $y$ 方向分量，斜率 $k$ 满足 $|k|<1$，$|\Delta x|>|\Delta y|$，所以 $x$ 方向为主位移方向；在 $C$ 点，法矢量的 $x$ 方向分量等于 $y$ 方向分量，斜率 $k$ 满足 $k=-1$，$|\Delta x|=|\Delta y|$；在部分Ⅱ的 $CB$ 椭圆弧段，法矢量的 $x$ 方向分量大于 $y$ 方向分量，斜率 $k$ 满足 $|k|>1$，$|\Delta y|>|\Delta x|$，所以 $y$ 方向为主位移方向。

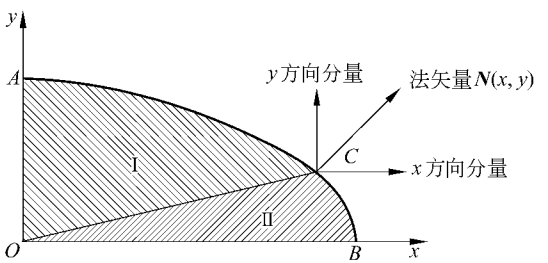



图 3-11　椭圆的法矢量

椭圆的中点 Bresenham 算法的原理如下。

在部分Ⅰ：每次在主位移 $x$ 方向上走一步，$y$ 方向上退不退步取决于中点偏差判别式

的值。

在部分Ⅱ：每次在主位移 $y$ 方向上退一步，$x$ 方向上走不走步取决于中点偏差判别式的值。

本算法需要分两个部分顺时针来绘制 1/4 椭圆。在上半部分Ⅰ，$x$ 方向上每次加 1，$y$ 方向上减 1 不减 1 取决于中点偏差判别式的值；在下半部分Ⅱ，$y$ 方向上每次减 1，$x$ 方向上加 1 不加 1 取决于中点偏差判别式的值。

先考虑图 3-12 所示部分Ⅰ的 $AC$ 段椭圆弧。此时中点 Bresenham 画椭圆算法要从 $A(0,b)\sim(a^2/\sqrt{a^2+b^2},b^2/\sqrt{a^2+b^2})$ 顺时针确定最佳逼近于该段椭圆弧的像素点集。由于 $x$ 方向为主位移方向，假定当前点是 $P(x_i,y_i)$，下一步只能在正右方的像素 $P_u(x_i+1,y_i)$ 和右下方的像素 $P_d(x_i+1,y_i-1)$ 中选取。

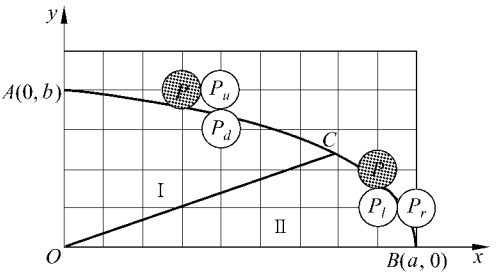


图 3-12　椭圆中点 Bresenham 算法原理

再考虑图 3-12 所示部分Ⅱ的 $CB$ 段椭圆弧。此时中点 Bresenham 画椭圆算法要从 $C(a^2/\sqrt{a^2+b^2},b^2/\sqrt{a^2+b^2})$ 到 $B(a,0)$ 顺时针确定最佳逼近于该段椭圆弧的像素点集。由于 $y$ 方向为主位移方向，假定当前点是 $P(x_i,y_i)$，下一步只能在正下方的像素 $P_l(x_i,y_i-1)$ 和右下方的像素 $P_r(x_i+1,y_i-1)$ 中选取。

### 3.3.2　构造上半部分Ⅰ中点偏差判别式

在上半部分Ⅰ，$x$ 方向每次加 1，$y$ 方向上减不减 1 取决于中点偏差判别式的值。从 $P(x_i,y_i)$ 走第一步，为了选取下一像素点，需将 $P_u(x_i+1,y_i)$ 和 $P_d(x_i+1,y_i-1)$ 的中点 $M(x_i+1,y_i-0.5)$ 代入隐函数，构造中点偏差判别式

$$d_1 = F(x_M,y_M) = F(x_i+1,y_i-0.5) = b^2(x_i+1)^2 + a^2(y_i-0.5)^2 - a^2b^2 \tag{3-16}$$

当 $d_1<0$ 时，中点 $M$ 在椭圆内，下一像素点应点亮 $P_u$，即 $y$ 方向不退步；当 $d>0$ 时，中点 $M$ 在椭圆外，下一像素点应点亮 $P_d$，即 $y$ 方向退一步；当 $d=0$ 时，中点 $M$ 在椭圆上，$P_u$、$P_d$ 和椭圆的距离相等，点亮 $P_u$ 或 $P_d$ 均可，约定取 $P_d$，如图 3-13 所示。

因此

$$y_{i+1} = \begin{cases} y_i, & d_1 < 0 \\ y_i - 1, & d_1 \geqslant 0 \end{cases} \tag{3-17}$$

### 3.3.3　上半部分Ⅰ的递推公式

图 3-13 中，根据当前点 $P(x_i,y_i)$ 确定了下一点是点亮 $P_u$ 还是 $P_d$ 时，使用了中点偏差判别式 $d_1$。为了能够继续判断椭圆上的每个点，需要给出中点偏差判别式 $d_1$ 的递推公式和初始值。

**1. 中点偏差判别式的递推公式**

现在如果考虑主位移方向再走一步，应该选取哪个中点代入中点偏差判别式以决定应

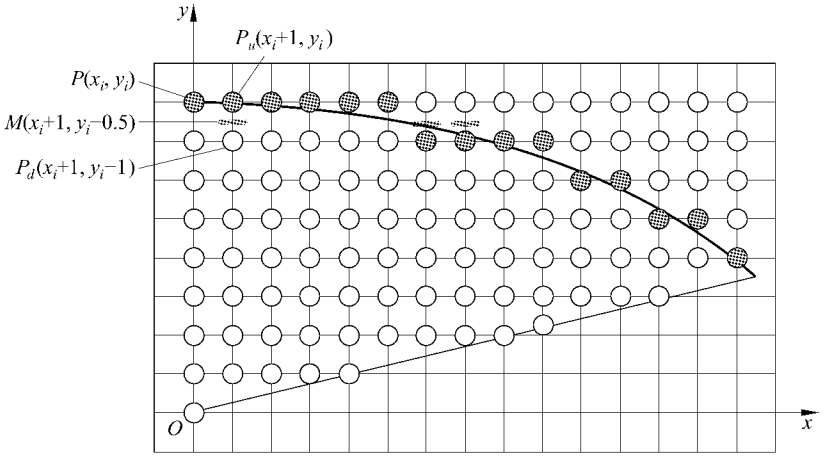



图 3-13　上半部分像素点的选取

该点亮的像素，如图 3-14 所示，分两种情况讨论。

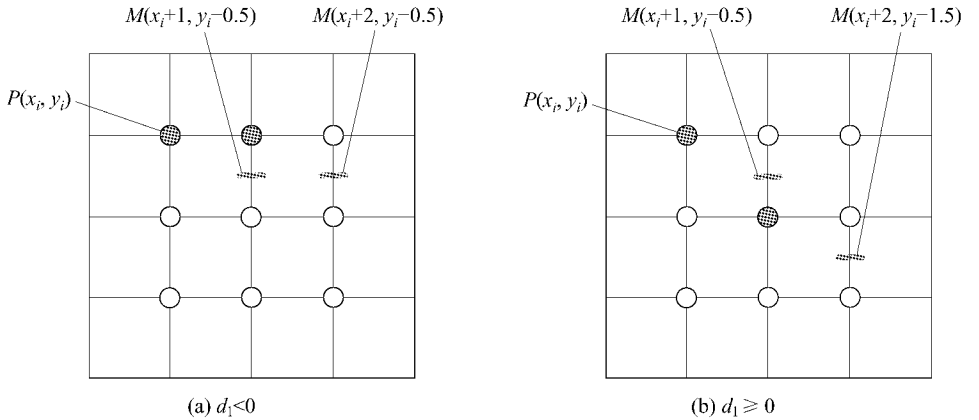



图 3-14　上半部分中点偏差判别式的递推

(1) 当 $d_1<0$ 时，下一步的中点坐标为 $M(x_i+2, y_i-0.5)$。所以下一步中点偏差判别式为

$$\begin{aligned} d_{1(i+1)} &= F(x_i+2, y_i-0.5) = b^2(x_i+2)^2 + a^2(y_i-0.5)^2 - a^2b^2 \\ &= b^2(x_i+1)^2 + a^2(y_i-0.5)^2 - a^2b^2 + b^2(2x_i+3) \\ &= d_{1i} + b^2(2x_i+3) \end{aligned} \tag{3-18}$$

(2) 当 $d_1 \geqslant 0$ 时，下一步的中点坐标为 $M(x_i+2, y_i-1.5)$。所以下一步中点偏差判别式为

$$\begin{aligned} d_{1(i+1)} &= F(x_i+2, y_i-1.5) = b^2(x_i+2)^2 + a^2(y_i-1.5)^2 - a^2b^2 \\ &= b^2(x_i+1)^2 + a^2(y_i-0.5)^2 - a^2b^2 + b^2(2x_i+3) + a^2(-2y_i+2) \\ &= d_{1i} + b^2(2x_i+3) + a^2(-2y_i+2) \end{aligned} \tag{3-19}$$

**2. 中点偏差判别式 $d_1$ 的初值**

上半部分椭圆的起点为 $A(0, b)$，因此，第一个中点是 $(1, b-0.5)$，对应的 $d_1$ 的初值为

$$d_{10} = F(1, b-0.5) = b^2 + a^2(b-0.5)^2 - a^2b^2 = b^2 + a^2(-b+0.25) \quad (3\text{-}20)$$

### 3.3.4 构造下半部分Ⅱ中点偏差判别式

在下半部分Ⅱ，主位移方向发生变化，中点 Bresenham 算法原理为：$y$ 方向上每次减 1，$x$ 方向上加 1 不加 1 取决于中点偏差判别式的值。从上半部分Ⅰ的终止点 $P(x_i, y_i)$ 开始，为了进行下一像素点的选取，需将 $P_l(x_i, y_i-1)$ 和 $P_r(x_i+1, y_i-1)$ 的中点 $M(x_i+0.5, y_i-1)$ 代入隐函数，构造中点偏差判别式

$$d_2 = F(x_M, y_M) = F(x_i+0.5, y_i-1) = b^2(x_i+0.5)^2 + a^2(y_i-1)^2 - a^2b^2 \quad (3\text{-}21)$$

当 $d_2<0$ 时，中点 $M$ 在椭圆内，下一像素点应点亮 $P_r$，即 $x$ 方向上走一步；当 $d_2>0$ 时，中点 $M$ 在椭圆外，下一像素点应点亮 $P_l$，即 $x$ 方向上不走步；当 $d_2=0$ 时，中点 $M$ 在椭圆上，$P_l$、$P_r$ 和椭圆的距离相等，点亮 $P_l$ 或 $P_r$ 均可，约定取 $P_l$，如图 3-15 所示。

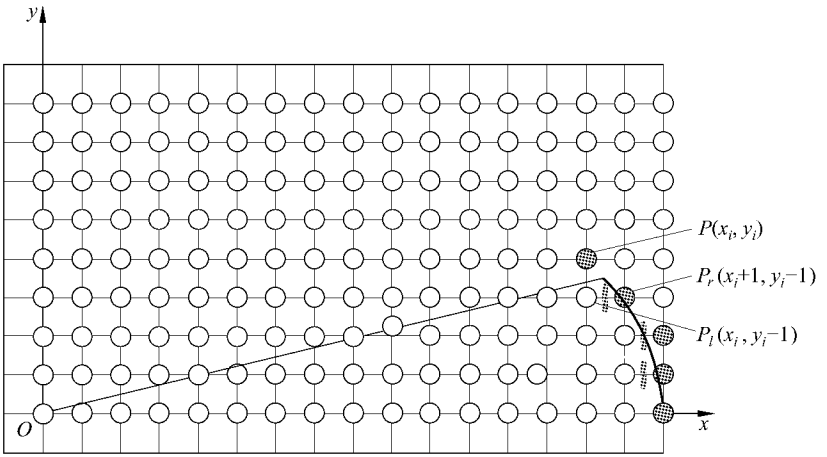



图 3-15　下半部分像素点的选取

因此

$$x_{i+1} = \begin{cases} x_i+1, & d_2<0 \\ x_i, & d_2 \geqslant 0 \end{cases} \quad (3\text{-}22)$$

### 3.3.5 下半部分Ⅱ的递推公式

图 3-15 中，根据上半部分Ⅰ的终止点 $P(x_i, y_i)$ 确定了下一点是点亮 $P_l$ 还是 $P_r$ 时，使用了中点偏差判别式 $d_2$。为了能够继续判断椭圆上的每一个点，需要给出中点偏差判别式 $d_2$ 的递推公式和初始值。

**1. 中点偏差判别式的递推公式**

现在如果考虑主位移方向上再走一步，应该选择哪个中点代入中点偏差判别式以决定应该点亮的像素，如图 3-16 所示，分两种情况讨论。

(1) 当 $d_2<0$ 时，下一步的中点坐标为 $M(x_i+1.5, y_i-2)$。所以下一步中点偏差判别式为

$$d_{2(i+1)} = F(x_i+1.5, y_i-2) = b^2(x_i+1.5)^2 + a^2(y_i-2)^2 - a^2b^2$$