《计算机图形学原理实践》

实验报告

|  |  |  |
| --- | --- | --- |
| 实验名称 | ： | 绘制任意斜率直线 |
|  |  |  |
| 姓 名 | ： |  |
| 学 号 | ： |  |
| 专业班级 | ： |  |
| 实验时间 | ： |  |

西南科技大学计算机科学与技术学院

**一、实验目的**

1、掌握任意斜率直线段的中点 Bresenham 扫描转化算法

2、掌握 Cline 直线类的设计方法

3、掌握状态栏编程技术

**二、实验步骤**

1、打开参考代码，创建MFC应用程序

2、根据直线的中点Bresenham算法原理设计代码

3、1）确定最大位移方向并计算误差初值e = 2 \* min - max;（min和max分别为水平距离和垂直距离的最值）

（2）求下一误差ei+1；如果 ei > 0 则ei+1 = e - 2 \* max;；否则ei+1 = ei + 2 \* min；

（3）根据不同象限，确定x和y变化符号的正负，进行下一次循环；

（4）如果没有结束，则转到步骤2，3；否则结束。

**三、实验结果**



**四、实验总结**

在本次实验中，通过查阅资料和相关学习，我学会了如何创建一个MFC应用程序，也能将中点Bresenham算法理论运用达到实践中去，掌握了绘制任意斜率直线的办法，过程中遇到了很多困难，但最终通过努力将其完成了，收获良多。

**附录：源代码（关键代码）**

void CLine::LineTo(CDC \*pDC,CP2 p1)//绘制直线，不含终点

{

P1=p1;

CP2 p,t;

CRGB clr=CRGB(0.0,0.0,0.0);//黑色像素点

if (fabs(P0.x - P1.x) < 1e-6)//绘制垂线

{

if (P0.y > P1.y)

{

t = P0;

P0 = P1;

P1 = t;

}

for (p = P0;p.y<P1.y;p.y++)

{

pDC->SetPixel(Round(p.x), Round(p.y), RGB(clr.red \* 255, clr.green \* 255, clr.blue \* 255));

}

}

else

{

double k, d;

k = (P1.y - P0.y) / (P1.x - P0.x);

if (k > 1.0)

{

if (P0.y > P1.y)

{

t = P0;

P0 = P1;

P1 = t;

}

d = 1 - 0.5 \* k;

for (p = P0; p.y < P1.y; p.y++)

{

pDC->SetPixel(Round(p.x), Round(p.y), RGB(clr.red \* 255, clr.green \* 255, clr.blue \* 255));

if (d>= 0)

{

p.x++;

d += 1 - k;

}

for (p = P0; p.y < P1.y; p.y++)

{

pDC->SetPixel(Round(p.x), Round(p.y), RGB(clr.red \* 255, clr.green \* 255, clr.blue \* 255));

if (d >= 0)

{

p.x++;

d += 1 - k;

}

else

d += 1;

}

}

}

if (0.0 <= k && k <= 1.0)

{

if (P0.x > P1.x)

{

t = P0;

P0 = P1;

P1 = t;

}

d = 0.5 - k;

for (p = P0; p.x < P1.x; p.x++)

{

pDC->SetPixel(Round(p.x), Round(p.y), RGB(clr.red \* 255, clr.green \* 255, clr.blue \* 255));

if (d < 0)

{

p.y++;

d += 1 - k;

}

else

d -= 1;

}

}

if (k >= -1.0 && k < 0.0)

{

if (P0.x > P1.x)

{

t = P0;

P0 = P1;

P1 = t;

}

d = -0.5 - k;

for (p = P0; p.x < P1.x; p.x++)

{

pDC->SetPixel(Round(p.x), Round(p.y), RGB(clr.red \* 255, clr.green \* 255, clr.blue \* 255));

if (d > 0)

{

p.y--;

d -= 1 + k;

}

else

d -= k;

}

}

if (k < -1.0)

{

if (P0.y > P1.y)

{

t = P0;

P0 = P1;

P1 = t;

}

d = -1 - 0.5 \* k;

for (p = P0; p.y < P1.y; p.y++)

{

pDC->SetPixel(Round(p.x), Round(p.y), RGB(clr.red \* 255, clr.green \* 255, clr.blue \* 255));

if (d < 0)

{

p.x++;

d -= 1 + k;

}

else

d -= 1;

}

}

}

P0=p1;

}