《计算机图形学原理实践》

实验报告

|  |  |  |
| --- | --- | --- |
| 实验名称 | ： | 直线段裁剪算法 |
|  |  |  |
| 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

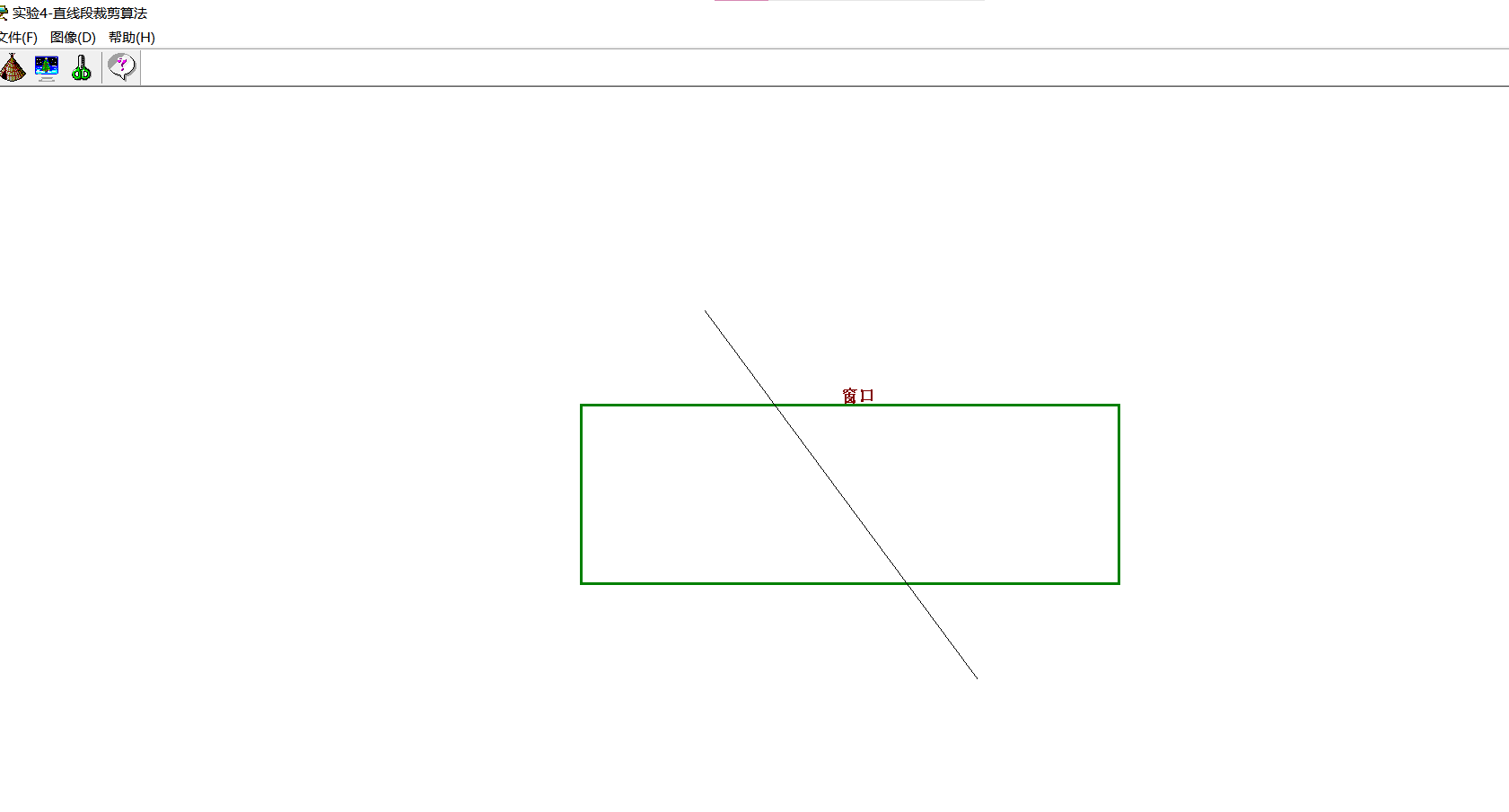
**实验目的**

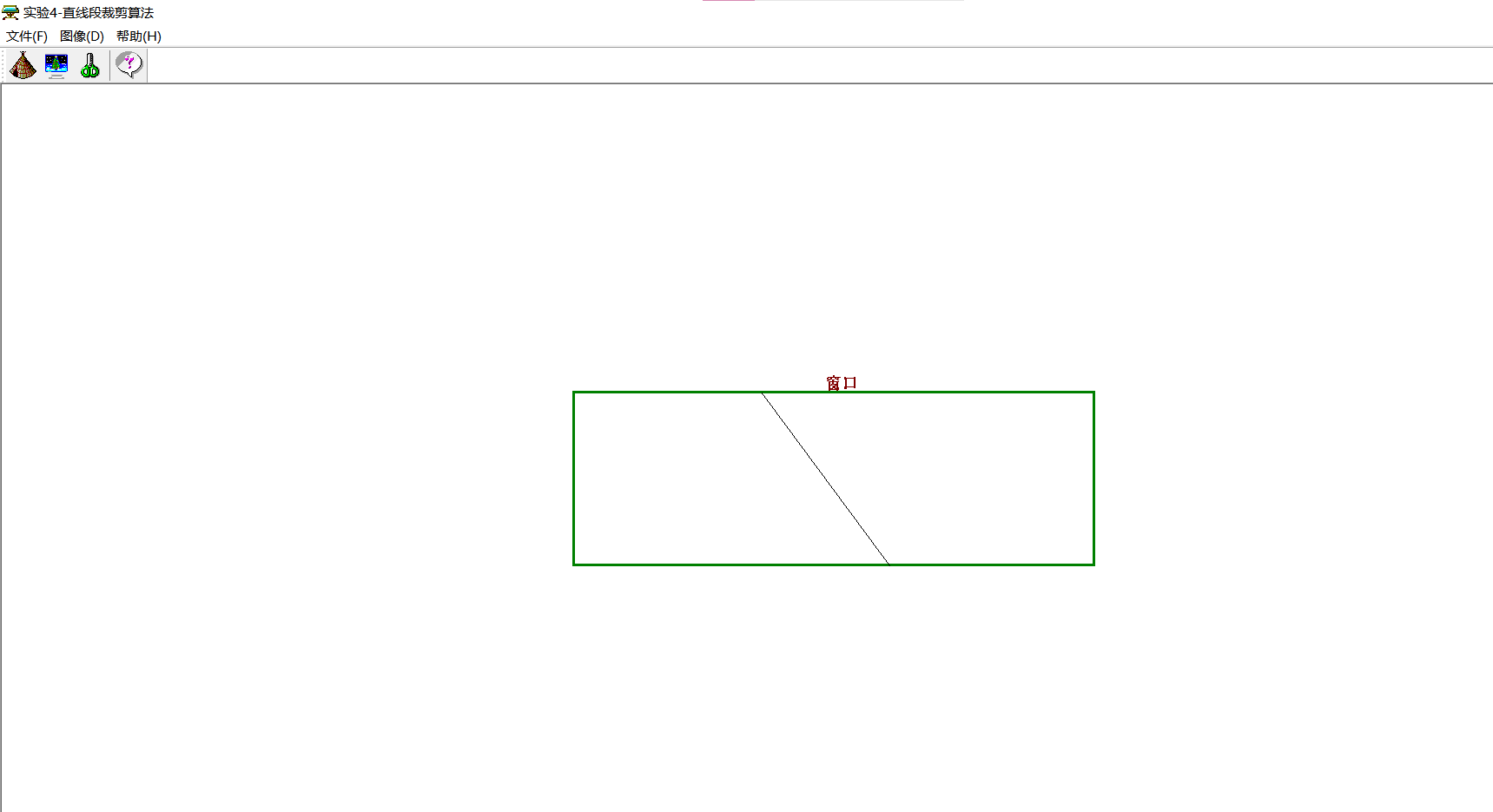
1、掌握 Cohen - Sutherland 直线段裁剪算法的直线段端点编码原理。

2、掌握“简取”、“简弃”和“求交”的判断方法。  
3、掌握直线段与窗口边界交点的计算公式。

**二、实验步骤**  
第一步：在屏幕显示左上角点为（Wx1, Wyt )，右下角为（ Wxrr Wyb ）的窗口。  
第二步：使用鼠标绘制两个端点坐标分别为 P0 (x0 : y0 ),P1(x1, y ）的直线段。  
第三步：将P0点编码为RC0，将P点编码为RC1。  
第四步：循环处理至少一个端点在窗口之外的情况。  
第五步：若RC0&RC≠0,“简弃”之。  
第六步：当RC0&RC=0，确保P0在裁剪窗口外部，若P。在窗口内部，则交换 P 和 P 的坐标位置及编码。  
第七步：按左、右、下、上的顺序计算窗口边界与直线段的交点P，并将该点的坐标值和编码赋值给P0，转步骤(第四步)。  
第八步：输出裁剪后的直线段。

**三、实验结果**





1. **实验总结**

本次实验利用Cohen-Suther land 算法，这是一种基于编码的裁剪方法。从中理解了直线段裁剪的基本知识，掌握了直线裁剪的程序设计方法。其中学到的知识让我受益匪浅，“简取”、“简弃”和“求交”的判断方法也可以运用到实际生活中来，对实验每一次的参透，都会打开生活的新世界。

**附录：源代码（关键代码）**

void CTestView::Cohen()//Cohen-Sutherland算法

{

CP2 p;

EnCode(P[0]);

EnCode(P[1]);

while (P[0].rc != 0 || P[1].rc != 0)

{

if ((P[0].rc & P[1].rc) != 0)

{

PtCount - 0;

return;

}

if (0 == P[0].rc)

{

CP2 Temp;

Temp = P[0];

P[0] = P[1];

P[1] = Temp;

}

UINT RC = P[0].rc;

double k = (P[1].y - P[0].y) / (P[1].x - P[0].x);

if (RC & LEFT)

{

p.x = Wxl;

p.y = k \* (p.x - P[0].x) + P[0].y;

}

else if (RC & RIGHT)

{

p.x = Wxr;

p.y = k \* (p.x - P[0].x) + P[0].y;

}

else if (RC & BOTTOM)

{

p.y = Wyb;

p.x = (p.y - P[0].y) / k + P[0].x;

}

else if (RC & TOP)

{

p.y = Wyt;

p.x = (p.y - P[0].y) / k + P[0].x;

}

EnCode(p);

P[0] = p;

}

}

void CTestView::EnCode(CP2& pt)//端点编码函数

{

pt.rc = 0;

if (pt.x < Wxl)

pt.rc = pt.rc | LEFT;

else if (pt.x > Wxr)

pt.rc = pt.rc | RIGHT;

if (pt.y < Wyb)

pt.rc = pt.rc | BOTTOM;

else if (pt.y > Wyt)

pt.rc = pt.rc | TOP;

}

void CTestView::OnLButtonDown(UINT nFlags, CPoint point)

{

// TODO: Add your message handler code here and/or call default

if(bDrawLine)

{ if(PtCount<2)

{

P[PtCount]=Convert(point);

PtCount++;

}

}

CView::OnLButtonDown(nFlags, point);

}

void CTestView::OnMouseMove(UINT nFlags, CPoint point)

{

// TODO: 在此添加消息处理程序代码和/或调用默认值

if(bDrawLine)

{

if(PtCount<2)

{

P[PtCount]=Convert(point);

Invalidate(FALSE);

}

}

CView::OnMouseMove(nFlags, point);

}

CP2 CTestView::Convert(CPoint point)//设备坐标系向自定义坐标系转换

{

CRect rect;

GetClientRect(&rect);

CP2 ptemp;

ptemp.x=point.x-rect.Width()/2;

ptemp.y=rect.Height()/2-point.y;

return ptemp;

}

void CTestView::OnClip()

{

// TODO: 在此添加命令处理程序代码

Cohen();

bDrawLine=FALSE;

Invalidate(FALSE);

}