

江南大学物联网工程学院实验报告

课程名称 《计算机图形学》 实验名称 实验 3 线段的裁剪与参数化曲线 实验日期 2017.11.16

班级 计科 1404 姓名 阎覃 学号 1030414414

实验报告要求 1. 实验目的 2. 实验内容 3. 实验步骤 4. 运行情况 5. 实验体会

1 实验内容

- 实现编码算法或 liang-barsky 算法
- 绘制一条 Bezier 曲线及其控制多边形

2 实验步骤及运行情况

Problem 1 实现编码算法或 liang-barsky 算法

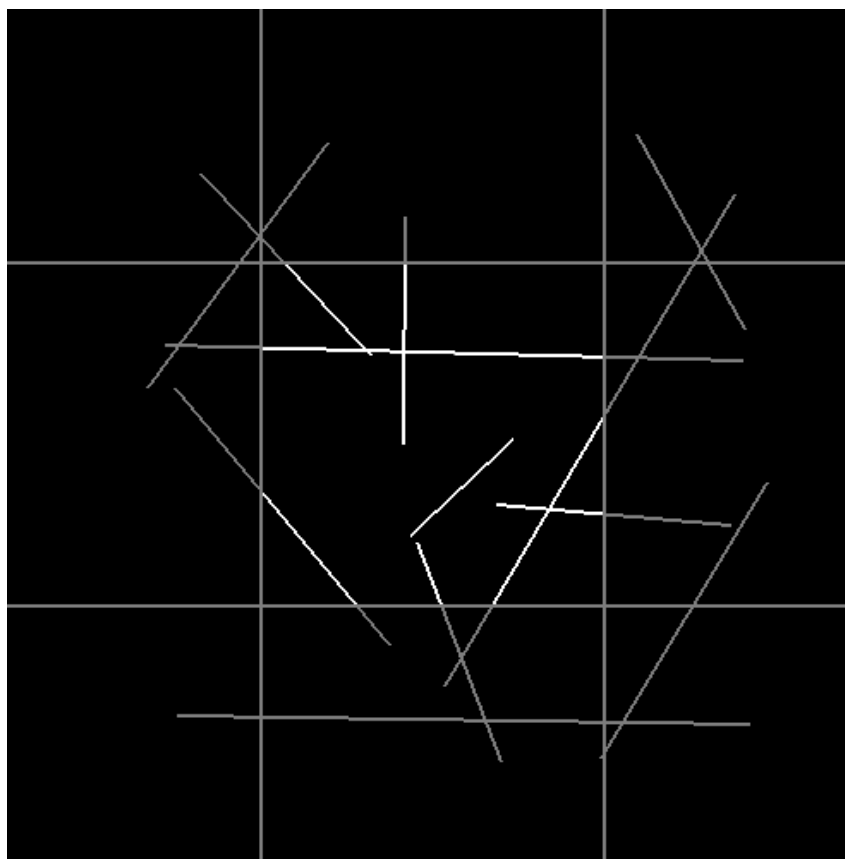
Entities/LineClipLB2D.cpp

```
1 //
2 // Created by Ethan on 2017/11/20.
3 //
4
5 #include "LineClipLB2D.h"
6
7 void LineClipLB2D::drawMe() {
8     float xL = -40, xR = 40, yB = -40, yT = 40;
9     float x2 = eP.x, x1 = sP.x, y2 = eP.y, y1 = sP.y;
10    glBegin(GL_LINES);
11    glColor3f(0.5, 0.5, 0.5);
12    glVertex2f(-100, yT);
13    glVertex2f(100, yT);
14    glVertex2f(-100, yB);
15    glVertex2f(100, yB);
16    glVertex2f(xL, -100);
17    glVertex2f(xL, 100);
18    glVertex2f(xR, -100);
19    glVertex2f(xR, 100);
20    glVertex2f(x1, y1);
21    glVertex2f(x2, y2);
22    glEnd();
23
24    float u1 = 0, u2 = 1, dx = x2 - x1, dy;
25    if (clipTest(-dx, x1 - xL, u1, u2)) {
26        if (clipTest(dx, xR - x1, u1, u2)) {
27            dy = y2 - y1;
28            if (clipTest(-dy, y1 - yB, u1, u2)) {
29                if (clipTest(dy, yT - y1, u1, u2)) {
30                    if (u2 < 1) {
31                        x2 = x1 + u2 * dx;
32                        y2 = y1 + u2 * dy;
33                    }
34                    if (u1 > 0) {
35                        x1 = x1 + u1 * dx;
36                        y1 = y1 + u1 * dy;
37                    }
38                    glBegin(GL_LINES);
39                    glColor3f(1, 1, 1);
40                    glVertex2f(x1, y1);
41                    glVertex2f(x2, y2);
42                    glEnd();
43                }
44            }
45        }
46    }
```

```
44     }  
45     }  
46 }  
47 }  
48  
49 int LineClipLB2D::clipTest(float p, float q, float &u1, float &u2) {  
50     float r;  
51     int retVal = 1;  
52     if (p < 0) {  
53         r = q / p;  
54         if (r > u2) retVal = 0;  
55         else if (r > u1) u1 = r;  
56     } else if (p > 0) {  
57         r = q / p;  
58         if (r < u1) retVal = 0;  
59         else if (r < u2) u2 = r;  
60     } else if (q < 0) retVal = 0;  
61     return retVal;  
62 }
```

本程序参考书上代码 P123。

运行截图



Problem 2 绘制一条 Bezier 曲线及其控制多边形

n 次 Bezier 曲线的数学表达式如下

$$Q(t) = \sum_{i=0}^n P_i B_{i,n}(t), t \in [0, 1] \quad (1)$$

$$B_{i,n}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}, i = 0, 1, \dots, n \quad (2)$$

de Casteljau 算法用下面的迭代公式计算型值点 $Q(t)$

$$P_i^r = \begin{cases} P_i & r = 0 \\ (1-t)P_i^{r-1} + tP_{i+1}^{r-1} & r = 1, 2, \dots, n \end{cases} \quad i = 0, 1, \dots, n-r$$

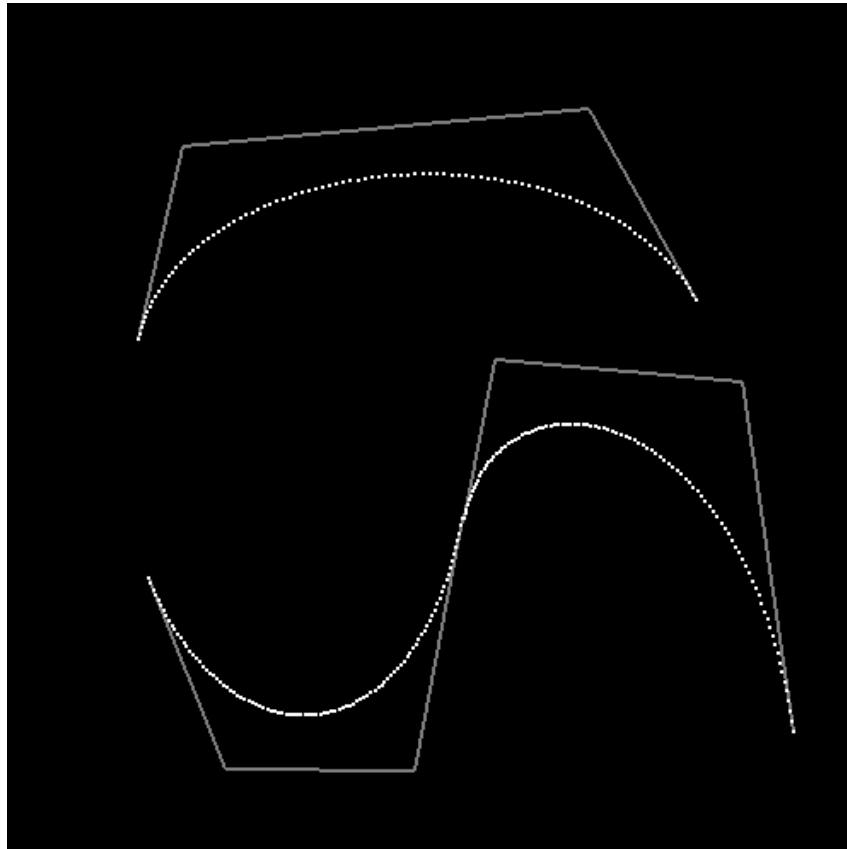
可以用数学归纳法证明, 型值点 $Q(t)$ 就是 P_0^n

Entities/BezierCas2D.cpp

```

1 //
2 // Created by Ethan on 2017/11/20.
3 //
4
5 #include "BezierCas2D.h"
6
7 void BezierCas2D::drawMe() {
8     glBegin(GL_LINES);
9     glColor3f(0.5, 0.5, 0.5);
10    for (int i = 0; i < top; i++) {
11        glVertex2f(p[i].x, p[i].y);
12        glVertex2f(p[i + 1].x, p[i + 1].y);
13    }
14    glEnd();
15
16    if (top == n) {
17        glBegin(GL_POINTS);
18        glColor3f(1, 1, 1);
19        for (int count = 0; count <= 100; count++) {
20            Point2D c[n + 1];
21            float t = count / 100.0f;
22            float t1 = 1.0f - t;
23
24            for (int i = 0; i <= n; i++) {
25                c[i] = p[i];
26            }
27
28            for (int r = 1; r <= n; r++) {
29                for (int i = 0; i <= n - r; i++) {
30                    c[i].x = t1 * c[i].x + t * c[i + 1].x;
31                    c[i].y = t1 * c[i].y + t * c[i + 1].y;
32                }
33            }
34            glVertex2f(c[0].x, c[0].y);
35        }
36        glEnd();
37    }
38 }
```

运行截图



3 实验体会

线段裁剪算法有很多, 本次试验实现的是 Liang-Barsky 算法。程序中设定了一个裁剪窗口, 在窗口外的线用灰色表示, 窗口内被裁剪的线段用白色表示。本程序可以实现绘制 n 阶 Bezier 曲线, 在程序中修改 n 的值即可。控制多边形用灰色表示。

实验报告采用 \LaTeX 排版, 完整代码托管至 GitHub:
<https://github.com/Ethan-yt/JNU-CG-exp>

教师评价	优	良	中	及格	不及格	教师签名	日期
------	---	---	---	----	-----	------	----