**HW3 DRAWLINES**

1 projection matrix：

输入x1 = [x,y,z,1]

输入相机参数

x’ = -x\*n/z

y’ = -y\*n/z

输出[x’,y’]

2 3d-lineclipping：

输入x1 = [x,y,z,1]

输入x2 = [x’,y’,z’,1]

输入相机参数

投影矩阵

m = np.mat([[2\*n/(r-l),0,(r+l)/(r-l),0],[0,2\*n/(t-b),(t+b)/(t-b),0],[0,0,(f+n)/(n-f),2\*f\*n/(n-f)],[0,0,-1,0]])

n1 = m.I

m为投影矩阵，n为逆矩阵。

之后得到投影点A(a1,b1,c1) B(a2,b2,c2)

若它们两点是同侧的外点，直接删去。（如A(2,2,2,1) B(3,3,3,1)）

如果不是，做以下计算：

若x1≠x2，计算AB直线与x=1，x=-1的交点C,D。

若y1≠y2，计算AB直线与y=1，y=-1的交点E,F。

若z1≠z2，计算AB直线与z=1，z=-1的交点G,H。

这六个交点中在正方形面上的最多只有两个。

若不足两个，该直线舍去。若恰为两个，记为M,N

创建数组[A,B,M,N],并对其排序（sorted）

取数组中第1和第二个元素（上述的ABMN则取B和M）加入新的数组。

设新数组中的两点是P,Q。

将P,Q转化为齐次坐标并投影回原来的坐标系下，记为P1,Q1.

输出[P1,Q1]

3 2d-lineclipping：

输入x1 = [x,y,1]

输入x2 = [x’,y’,1]

输入相机参数

对其编码（二进制码）

若p1 | p2 == 0:

返回这两点初始坐标

若 p1 & p2 != 0 :

返回空(None)

其余情况：执行下面的计算：

若它们两点是同侧的外点，直接删去。（如A(2,2,1) B(3,3,1)）

如果不是，做以下计算：

若x1≠x2，计算AB直线与x=right，x=left的交点C,D。

若y1≠y2，计算AB直线与y=top，y=bottom的交点E,F。

这四个交点中在正方形上的最多只有两个。

若不足两个，该直线舍去。若恰为两个，记为M,N

创建数组[A,B,M,N],并对其排序（sorted）

取数组中第1和第二个元素（上述的ABMN则取B和M）加入新的数组。

设新数组中的两点是P,Q。

输出[P,Q]

4 rasterization\_DDA:

输入两点A(X1,Y1) B(X2,Y2)

通过变换矩阵将其变换为相纸的对应坐标

判断斜率kAB 若其>1，则选择y方向为步长，步长1or-1

若其<1，则选择x方向为步长，步长1or-1

画出上面的每一点，连为直线。

5 main程序

依靠sys.stdin读取数据，将其分组。

对每一组点：

先做3d-lineclipping，之后做projection，最后做rasterization\_DDA，输出直线。

图1：

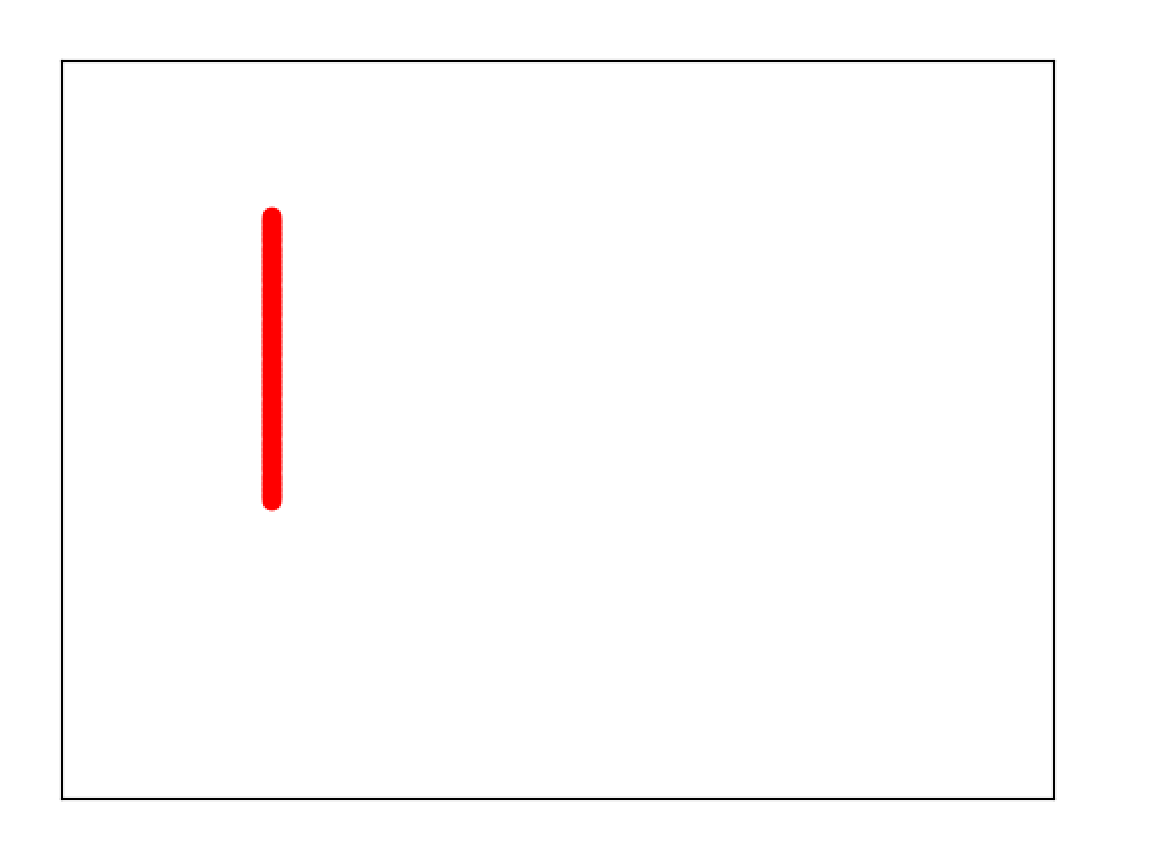


图2：

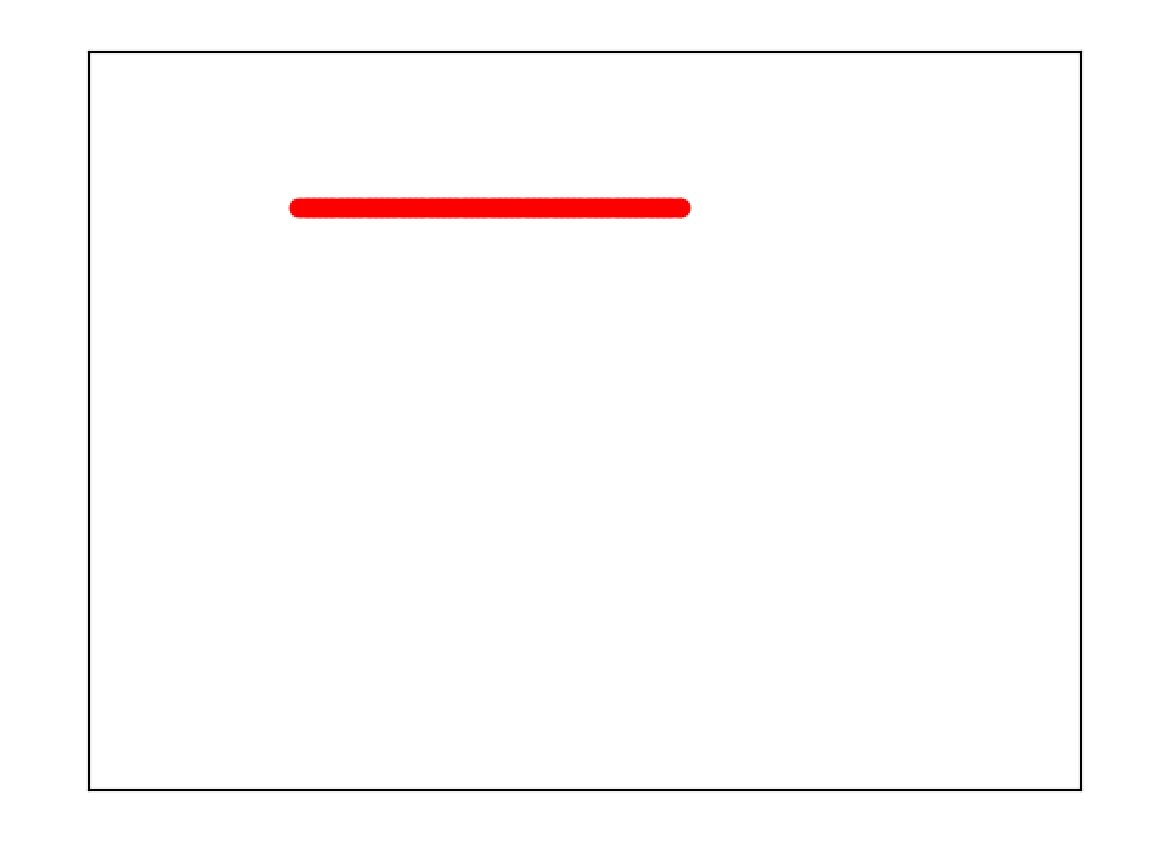


图3

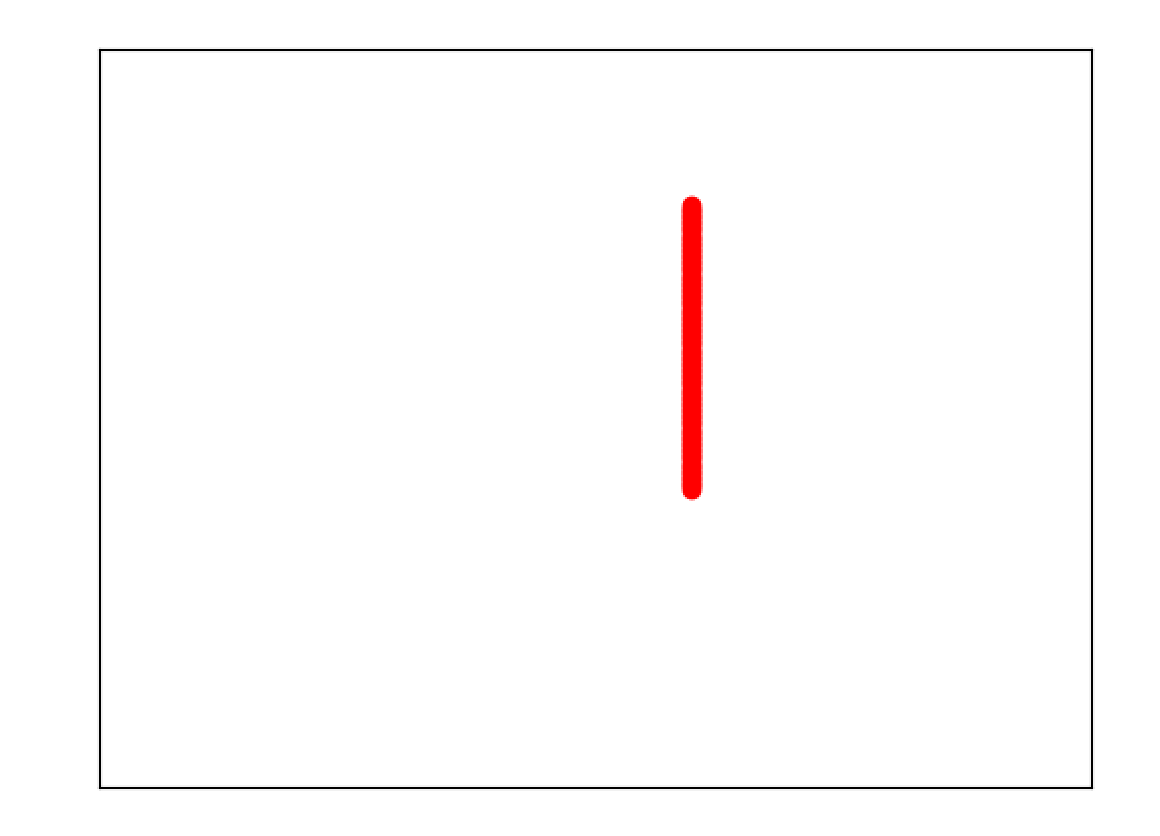


图4

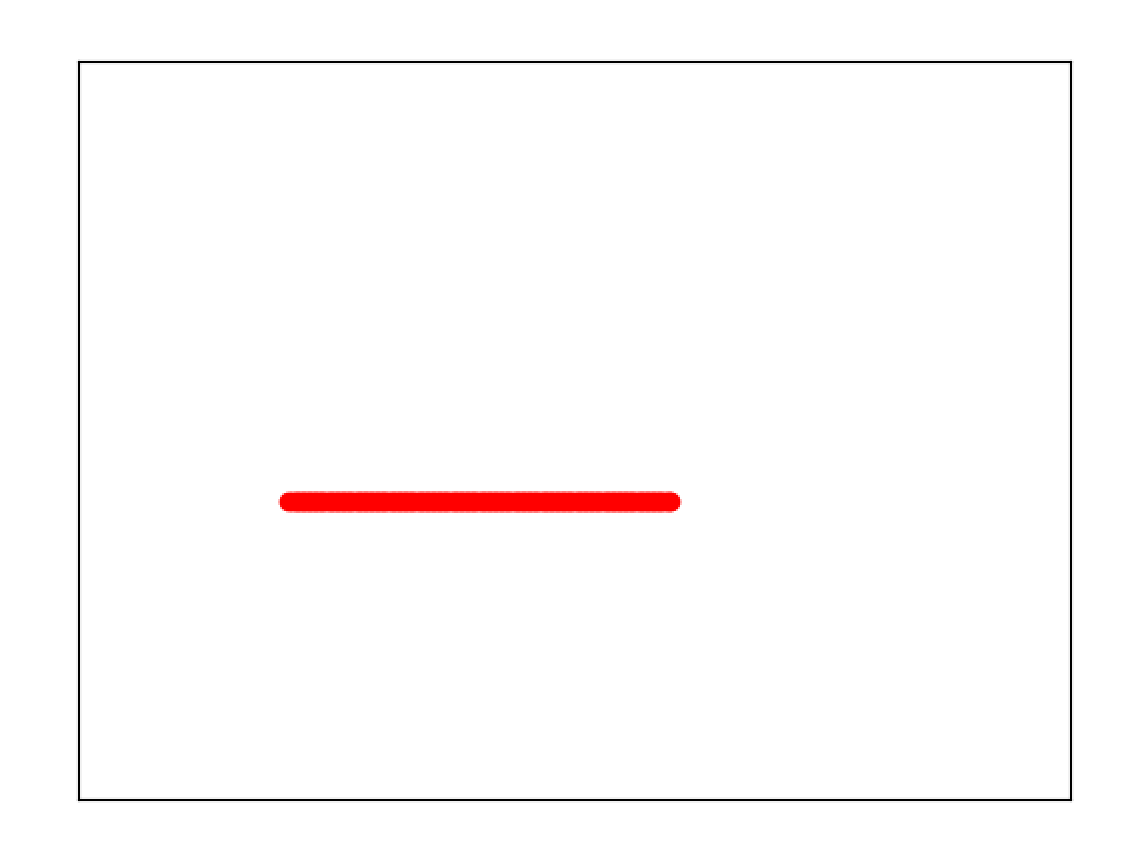


图5



图6



图7

