

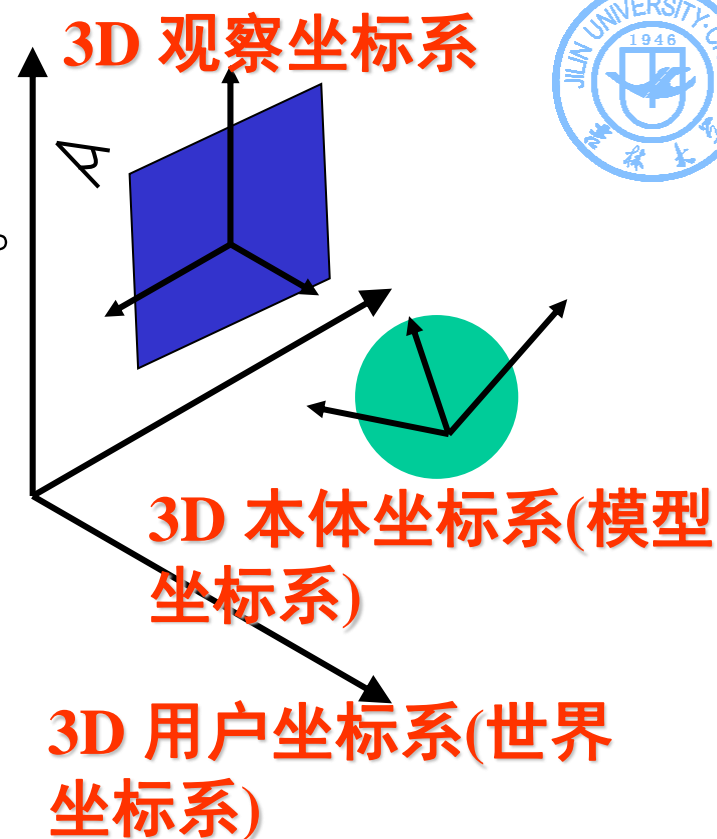


# 第三章 图形变换

- 第一节 齐次坐标
- 第二节 二维图形变换
- 第三节 二维视见变换
- 第四节 三维图形变换
- 第五节 投影变换
- 第六节 裁剪



- 本体坐标系  
也称模型坐标系，是为规定基本形体而引入的便于描述的坐标系。
- 用户坐标系  
也称世界坐标系，是用户引入描述整个形体的坐标系。
- 观察坐标系  
也称视坐标系或目坐标系，为说明观察的姿态而引入，也就是观察者所处的位置。



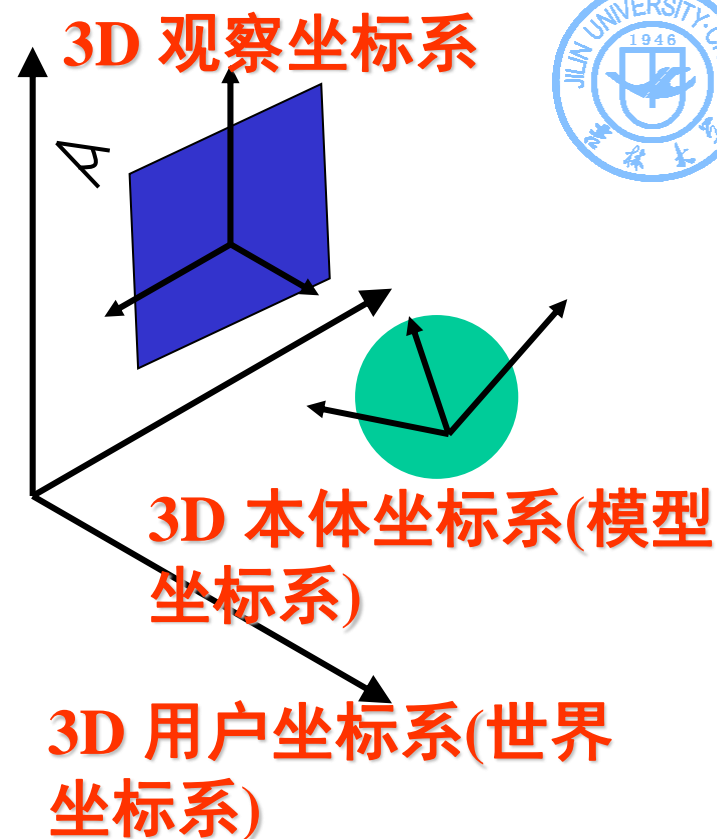


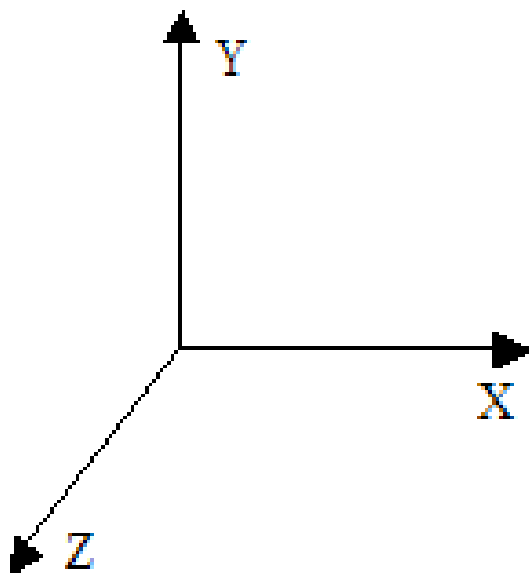
- 设备坐标系

也称显示器坐标系或屏坐标系，是各种图形设备自身规定的在显示表面上采用的坐标系。要显示的景物最终要在这个坐标系中显示出来。

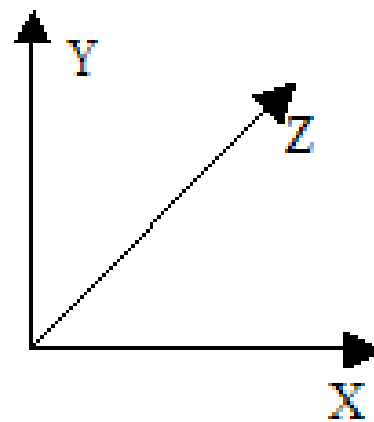
- 规范化设备坐标系

不同图形设备的设备坐标系一般不同，图形系统为了使绘制与具体设备无关而引入的坐标系，约定它是二维正方形，即各坐标范围规定为从0至1。





(1) 右手系



(2) 左手系

图 3.2 左手系和右手系

左手系：在计算机图形学中，通常约定为假想眼睛位于原点，x轴方向是水平向右，y轴方向竖直向上，z轴方向离开眼睛射向前方，这样形成的观察坐标系是左手系。（本书采用左手系）

右手系：如果让z轴的方向面向眼睛方向，这样得到的观察坐标系就是右手系。



# 坐标系

- 本体坐标系：三维坐标系（右手系）
- 用户坐标系：三维坐标系（右手系）
- 观察坐标系：三维坐标系（左手系）
- 规范化设备坐标系：二维坐标系
- 设备坐标系：二维坐标系

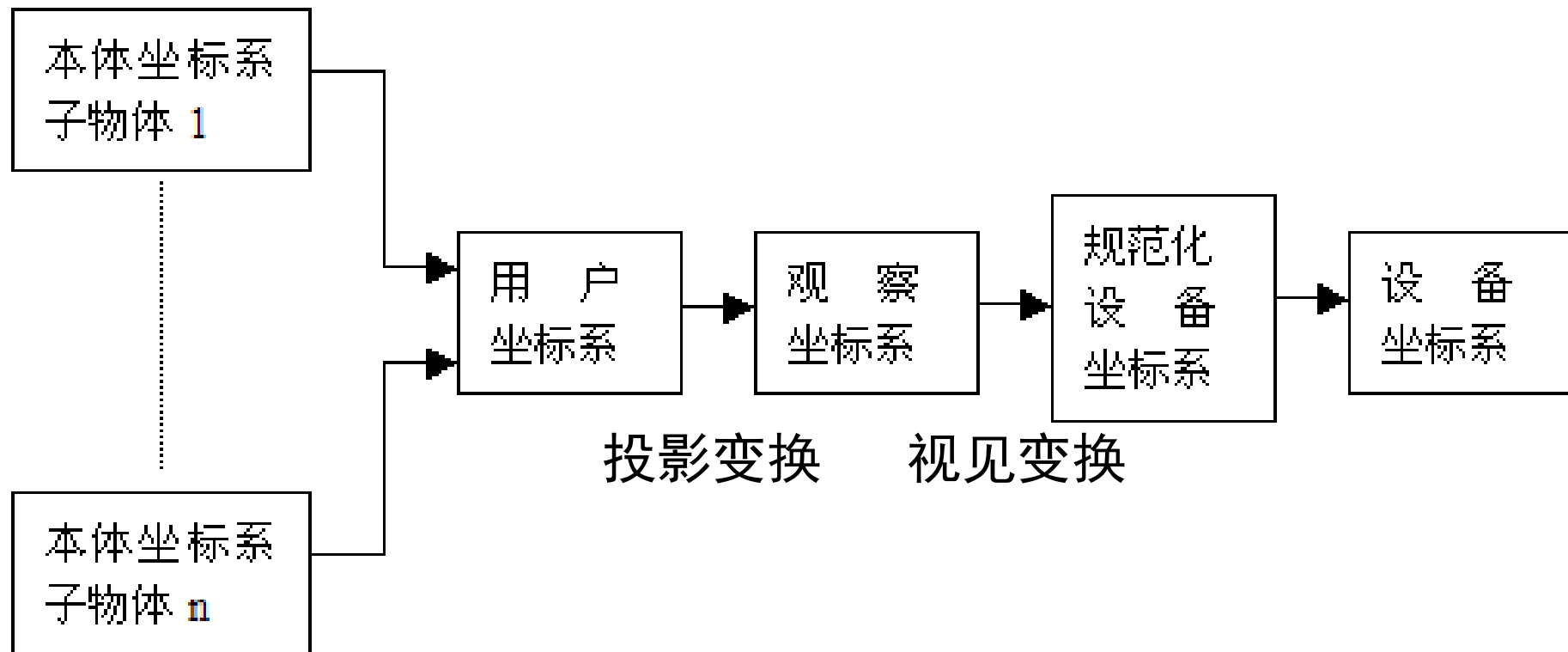


# 坐标系

- 图形（几何）变换：为了从不同方向去观察对象，而对对象所作的旋转、放大、缩小和平移等变换。将景物在本地坐标系中的描述转变为在用户坐标系中的描述。
- 投影变换：三维图形投影在二维投影平面上的变换。将景物在用户坐标系中的描述转变为在观察坐标系中的描述。
- 视见变换：绘图过程中要用窗口来规定要显示的内容，用视区来规定在屏幕上显示的位置所应用的变换。将景物在观察坐标系中的描述转变为在设备坐标系中的描述。



# 图形显示的坐标变换过程



图形（几何）变换

图 3.1 图形显示的坐标变换过程



# 第一节、齐次坐标

齐次坐标表示法就是用 $n+1$ 维向量表示一个 $n$ 维向量。

$n$ 维空间中的点的位置向量用非齐次坐标表示时， $(P_1, P_2, \dots, P_n)$ 具有 $n$ 个坐标分量，并且是唯一的。如果用齐次坐标表示时，该向量有 $n+1$ 个坐标分量， $(hP_1, hP_2, \dots, hP_n, h)$ 并且是不唯一的。

通常都使 $h=1$ 。如果 $h \neq 0$ ，就可以用除齐次坐标的各分量，这一方法称之为齐次坐标的规范化。经过规范化后的齐次坐标就是唯一的。





# 第一节、齐次坐标

如二维点 $(x,y)$ 的齐次坐标表示为 $(hx,hy,h)$ ，  
则 $(h_1x,h_1y,h_1)$ ， $(h_2x,h_2y,h_2)$ ， $\dots$ ，  
 $(h_mx,h_my,h_m)$ 都是表示二维空间中的同一个点  
三维空间中的坐标点的齐次坐标可表示为 $(x,y,1)$ 。

应用齐次坐标可以有效地用矩阵运算把二维、  
三维甚至更高维空间中点集从一个坐标系转换到  
另一个坐标系中。



二维齐次坐标变换矩阵的形式为

$$T_{2D} = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$

三维齐次坐标变换矩阵的形式为

$$T_{3D} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$



## 第二节 二维图形变换

对一个图形作几何变换就是对该图形上的每一个点作相应的几何变换。

常见的基本二维图形几何变换有  
**平移变换、比例变换和旋转变换。**



# 平移变换

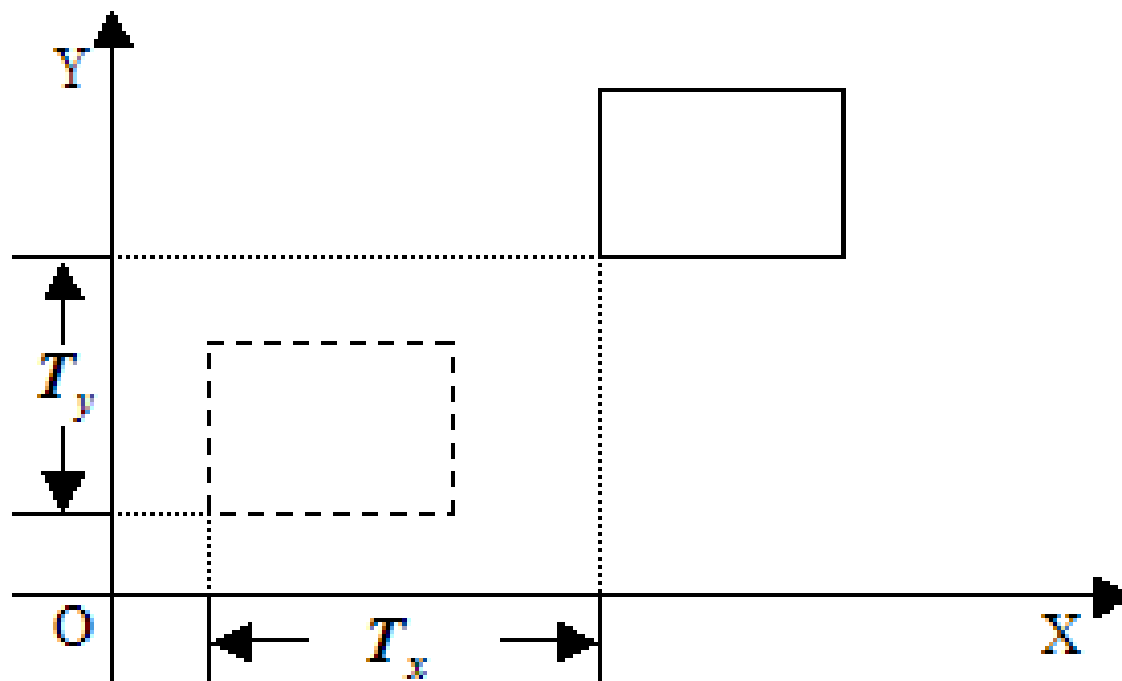


图 3.3 平移变换

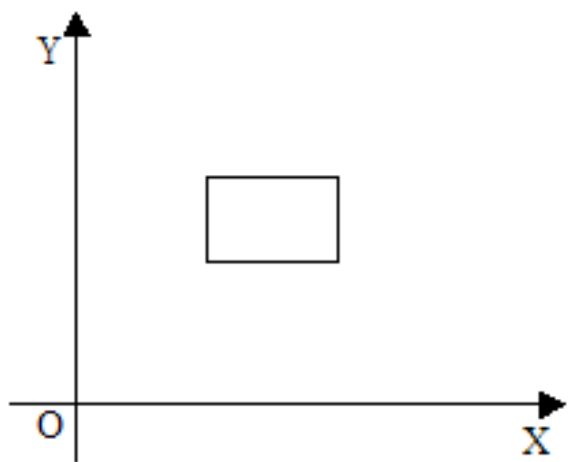
$(x, y)$

$(x', y')$

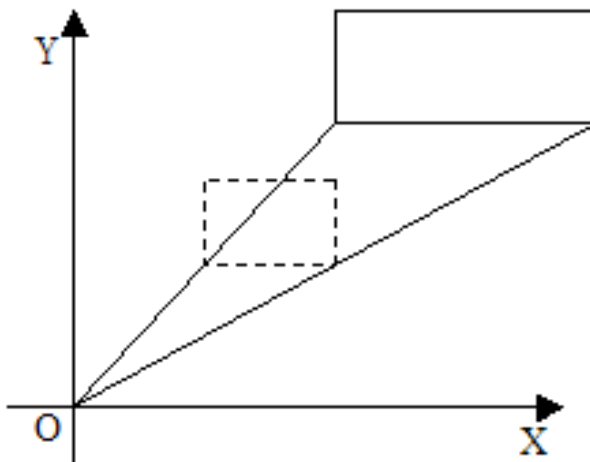
$$x' = x + T_x, \quad y' = y + T_y$$



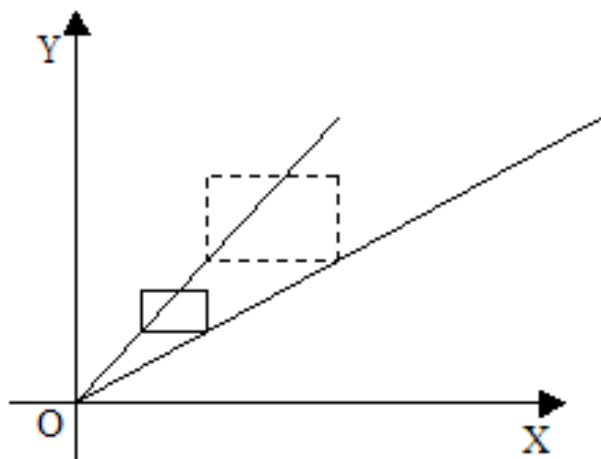
# 比例变换



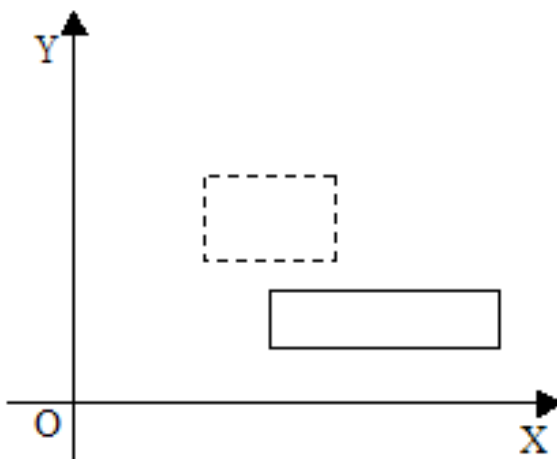
变换前



变换后  $S_x = S_y > 1$



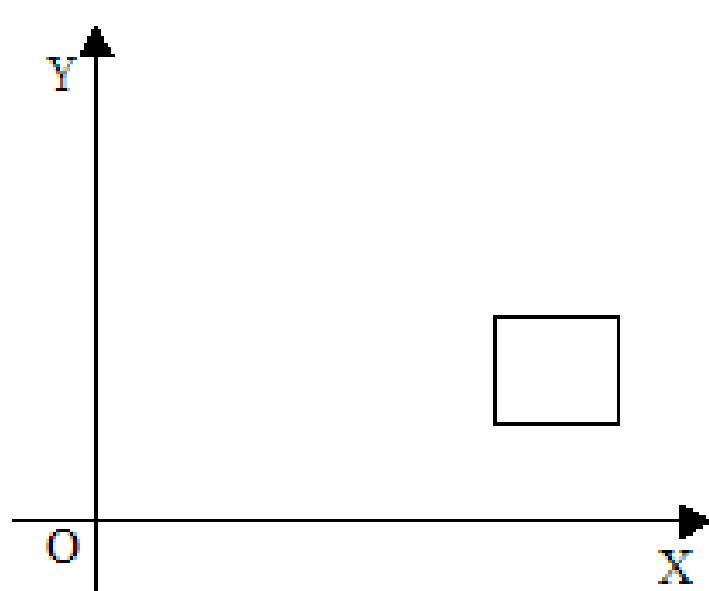
变换后  $S_x = S_y < 1$



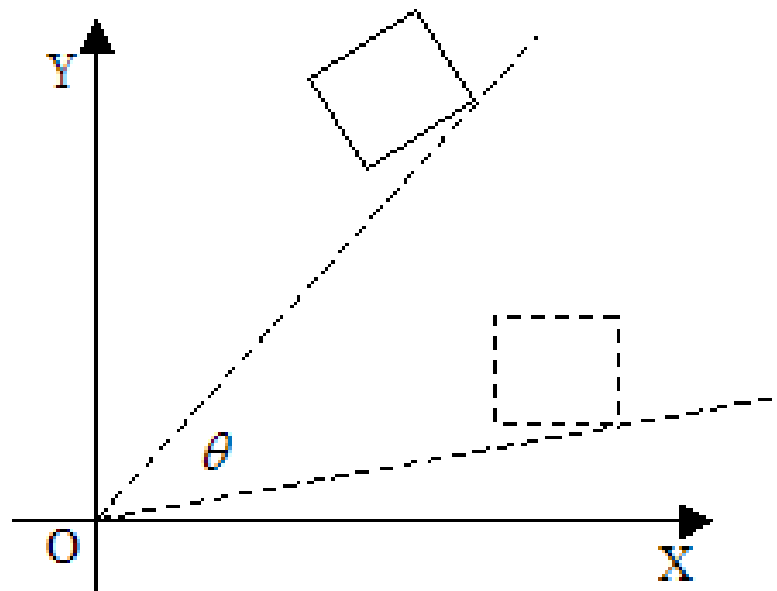
变换后  $S_x \neq S_y$

$$x' = x \bullet S_x$$
$$y' = y \bullet S_y$$

图 3.4 比例变换



变换前



变换后

图 3.5 旋转变换

旋转变换

$$\begin{aligned}x' &= x \cos \theta - y \sin \theta \\y' &= x \sin \theta + y \cos \theta\end{aligned}$$



# 1. 平移变换

$$x' = x + T_x, \quad y' = y + T_y$$

$$p' = [x' \quad y' \quad 1] = [x \quad y \quad 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x + T_x & y + T_y & 1 \end{bmatrix} = p \cdot T(T_x, T_y)$$



## 2. 比例变换

$$x' = x \bullet S_x \quad y' = y \bullet S_y$$

$$p' = [x' \quad y' \quad 1] = [x \quad y \quad 1] \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x \cdot S_x & y \cdot S_y & 1 \end{bmatrix}$$

$$= p \cdot S(S_x, S_y)$$





### 3. 旋转变换

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

$$p' = \begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x \cdot \cos \theta - y \cdot \sin \theta & x \cdot \sin \theta + y \cdot \cos \theta & 1 \end{bmatrix}$$

$$= p \cdot R(\theta)$$



可以用  $T(T_x, T_y)$ 、 $S(S_x, S_y)$ 、 $R(\theta)$  来表示平移、比例、旋转这三种基本的几何变换。多数常见的二维几何变换都可以通过这三种基本几何变换的组合来达到。

比例变换和旋转变换都是相对于坐标原点进行的，如果希望相对于任意一点作变换，可以先平移到原点，相对于原点作变换后，在平移回去。



**例一：求图形绕平面上任意一点  $(x_0, y_0)$  的旋转变换矩阵，设旋转角度为  $\theta$**

- (1) 平移  $T(-x_0, -y_0)$ , 使旋转中心与原点重合**
- (2) 绕原点逆时针旋转  $R(\theta)$**
- (3) 平移  $T(x_0, y_0)$ , 使旋转中心移回原处**



$$H = T(-x_0, -y_0) \cdot R(\theta) \cdot T(x_0, y_0)$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_0 & -y_0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_0 & y_0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ x_0(1 - \cos \theta) + y_0 \sin \theta & y_0(1 - \cos \theta) + x_0 \sin \theta & 1 \end{bmatrix}$$

$$x' = x_0 + (x - x_0) \cos \theta - (y - y_0) \sin \theta$$

$$y' = y_0 + (x - x_0) \sin \theta + (y - y_0) \cos \theta$$



**例二：相对于平面上任意一点 $(x_0, y_0)$ 的比例变换  
如下：**

**(1) 平移 $T(-x_0, -y_0)$ , 使比例中心与原点重合**

**(2) 相对于原点的比例变换  $S(S_x, S_y)$**

**(3) 平移 $T(x_0, y_0)$ , 比例中心移回原处**



$$H = T(-x_0, -y_0) \cdot S(S_x, S_y) \cdot T(x_0, y_0)$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_0 & -y_0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_0 & y_0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ x_0(1-S_x) & y_0(1-S_y) & 1 \end{bmatrix}$$

$$x' = x_0 + (x - x_0)s_x$$

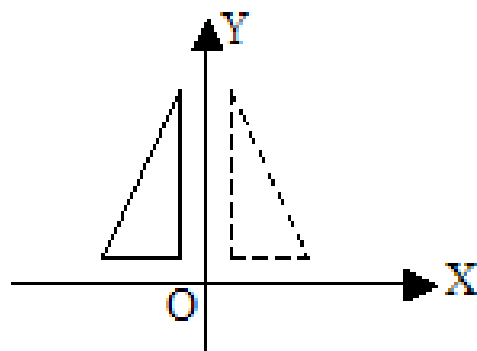
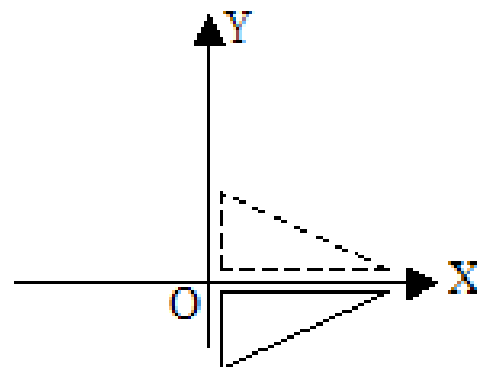
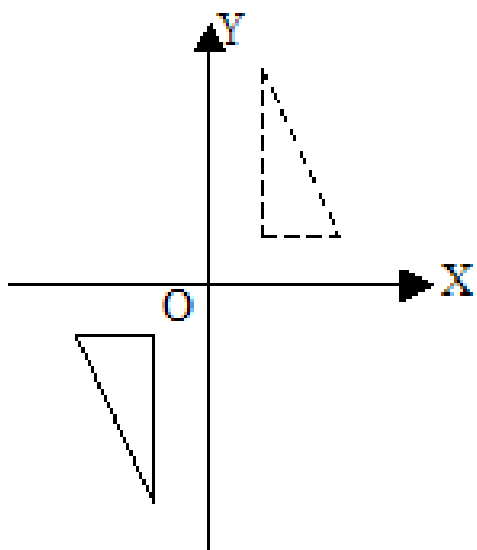
$$y' = y_0 + (y - y_0)s_y$$



# 常见的变换

## 1. 对称变换(反射、镜像变换)

$$p' = [x' \quad y' \quad 1] = [x \quad y \quad 1] \begin{bmatrix} a & d & 0 \\ b & e & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= [ax + by \quad dx + ey \quad 1]$$

 $Y$ 轴对称 $X$ 轴对称

中心对称

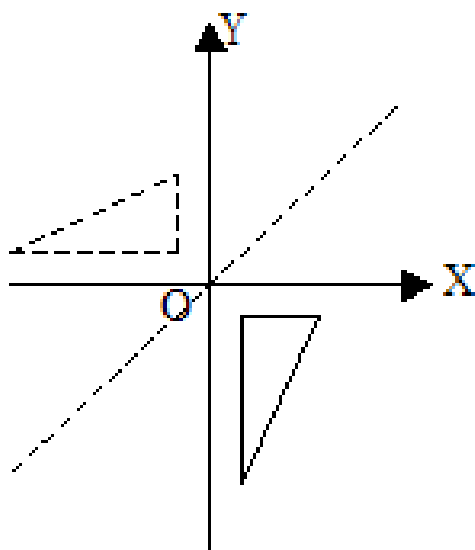
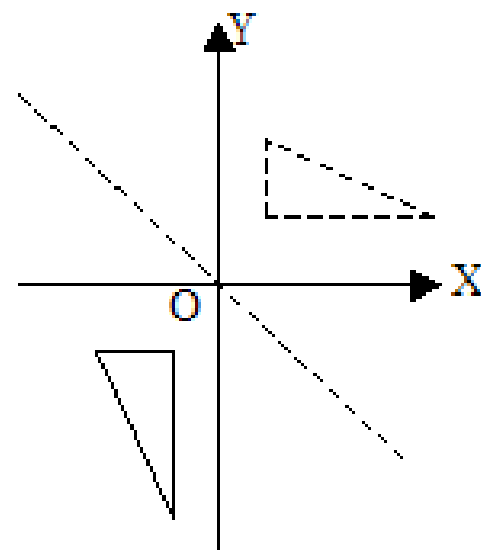
 $Y = X$ 对称 $Y = -X$ 对称

图 3.6 对称变换

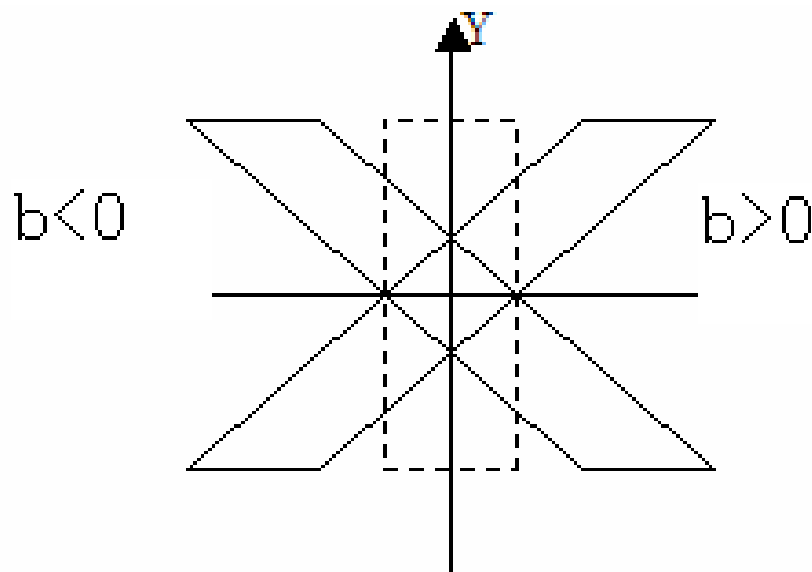




## 2. 错切变换(错移变换)

(1) 沿X轴方向关于Y的错切 (Y不变)

$$p' = \begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x+by & y & 1 \end{bmatrix}$$

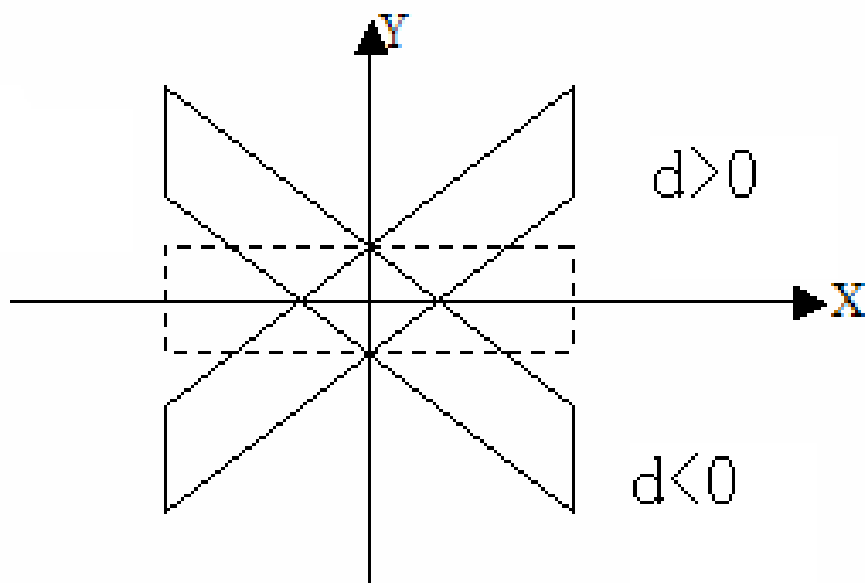


**X**方向错切



## (2) 沿Y轴方向关于X的错切 (X不变)

$$p' = \begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & d & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{pmatrix} x & y+dx & 1 \end{pmatrix}$$

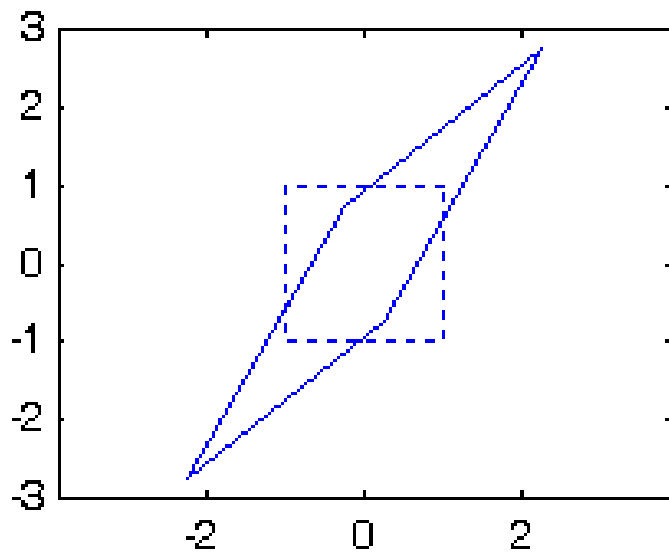
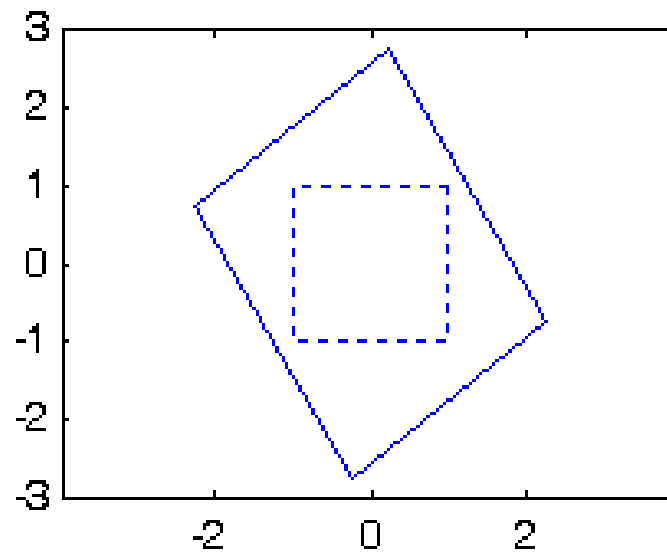
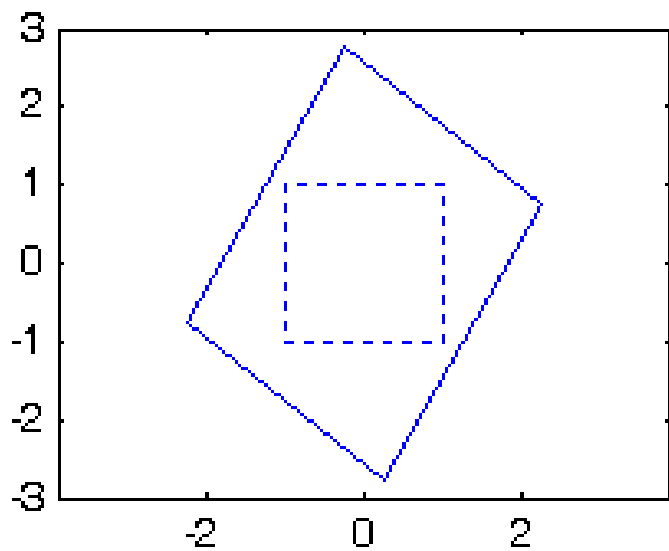
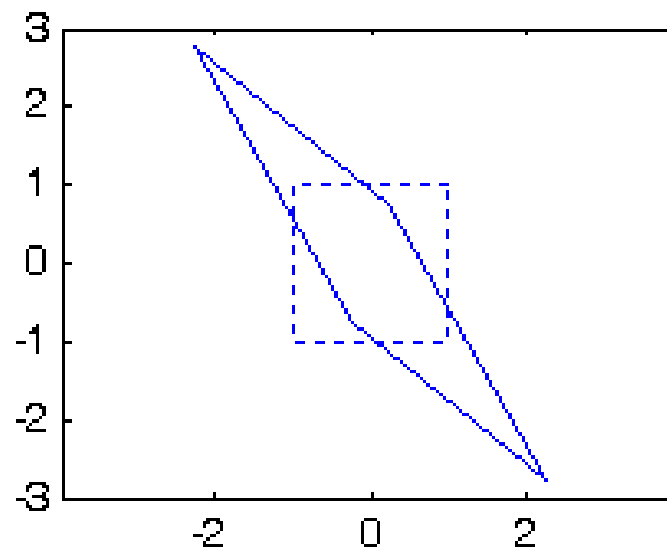


**Y** 方向错切



### (3) 沿X轴Y轴两个方向的错切

$$p' = [x' \quad y' \quad 1] = [x \quad y \quad 1] \begin{bmatrix} 1 & d & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= [x + by \quad dx + y \quad 1]$$

 $d > 0, b > 0$  $d < 0, b > 0$  $d > 0, b < 0$  $d < 0, b < 0$



## 复合变换的性质

$$\mathbf{T}(T_x, T_y) \cdot \mathbf{T}(T'_x, T'_y) = \mathbf{T}(T_x + T'_x, T_y + T'_y)$$

$$\mathbf{S}(S_x, S_y) \cdot \mathbf{S}(S'_x, S'_y) = \mathbf{S}(S_x \cdot S'_x, S_y \cdot S'_y)$$

$$\mathbf{R}(\theta) \cdot \mathbf{R}(\theta') = \mathbf{R}(\theta + \theta')$$



## 二维图形变换总结

平移变换：只改变图形的位置，不改变图形的大小和几何形状；

旋转变换：保持图形的大小和几何形状；

比例变换：改变图形的大小和几何形状；

错切变换：引起图形各部分之间角度关系的变化，会导致图形发生畸变；

拓扑不变的几何变换不改变图形的连接关系和平行关系。平移、旋转、比例、错切变换都属于拓扑不变的几何变换。



## 二维图形变换总结:

$$T_{2D} = \left[ \begin{array}{cc|c} a & d & g \\ b & e & h \\ \hline c & f & i \end{array} \right]$$



## 第三节 二维视见变换

**视见变换**：就是将观察坐标系窗口中指定的图形转换至设备坐标系视区中显示的过程。（Donald Hearn）

**窗口**就是在观察坐标系中指出的要显示出来的区域，这一区域通常为**矩形区域**。

**视见区**是屏幕域中的一个子区域，通常为矩形区域，它最大与屏幕域等同。

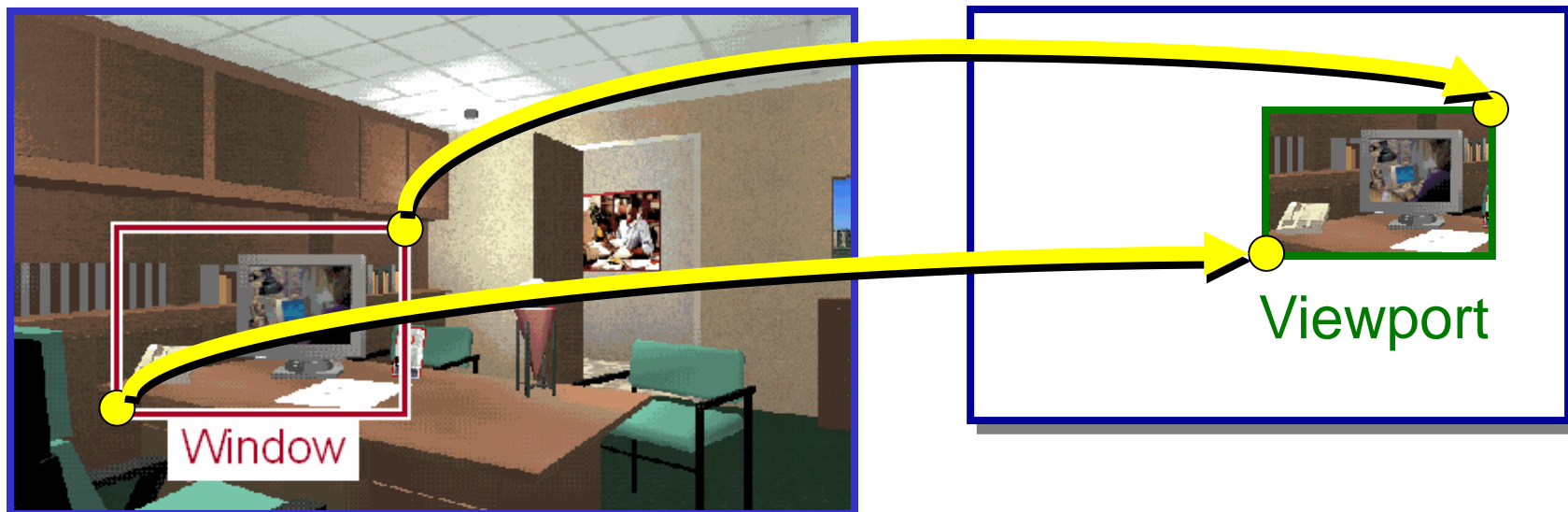




## 窗口到视见区

投影平面

屏幕



**窗口与视见区的差别：**

**窗口：** 在**观察坐标系**中确定，它指出了要显示的图形；

**视见区：** 在**设备坐标系**中确定，它指出了实际显示的图形处于显示屏幕的哪一部分。

视见区用于显示窗口中的图形。

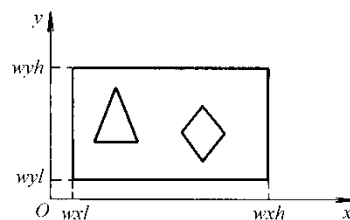


$$H = T_1(-w_{xl}, -w_{yl}) \bullet S(s_x, s_y) \bullet T_2(v_{xl}, v_{yl})$$

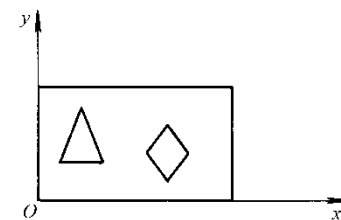
$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -w_{xl} & -w_{yl} & 1 \end{bmatrix} \begin{bmatrix} \frac{v_{xh}-v_{xl}}{w_{xh}-w_{xl}} & 0 \\ 0 & \frac{v_{yh}-v_{yl}}{w_{yh}-w_{yl}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ v_{xl} & v_{yl} & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{v_{xh}-v_{xl}}{w_{xh}-w_{xl}} & 0 \\ 0 & \frac{v_{yh}-v_{yl}}{w_{yh}-w_{yl}} \\ v_{xl}-w_{xl} \frac{v_{xh}-v_{xl}}{w_{xh}-w_{xl}} & v_{yl}-w_{yl} \frac{v_{yh}-v_{yl}}{w_{yh}-w_{yl}} \end{bmatrix}$$

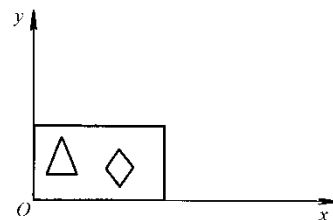
$$s_x = \frac{v_{xh}-v_{xl}}{w_{xh}-w_{xl}}, s_y = \frac{v_{yh}-v_{yl}}{w_{yh}-w_{yl}}$$



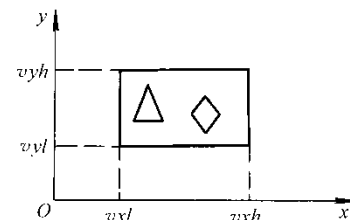
a) 用户坐标系中的窗口



b) 平移变换  $T_1$



c) 比例变换  $S$



d) 平移变换  $T_2$



设窗口中图形上的某一点坐标为  $(x, y)$ ，该点显示在视见区中的坐标为  $(x', y')$ ，利用视见变换矩阵可得出以下计算公式：

$$x' = vxl + (x - wxl) \frac{vxh - vxl}{wxh - wxl}$$

$$y' = vyl + (y - wyl) \frac{vyh - vyl}{wyh - wyl}$$



## 第四节 三维图形变换

设三维空间中任意一点P的齐次坐标为  $P=(x,y,z,1)$ ，做了三维图形几何变换后得到的点P'的齐次坐标为  $P'=(x',y',z',1)$ ，可得下面的三维图形几何变换矩阵。

### 1. 平移变换

$$P' = \begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ D_x & D_y & D_z & 1 \end{bmatrix}$$
$$= P \cdot T(D_x, D_y, D_z) = \begin{bmatrix} x+D_x & y+D_y & z+D_z & 1 \end{bmatrix}$$

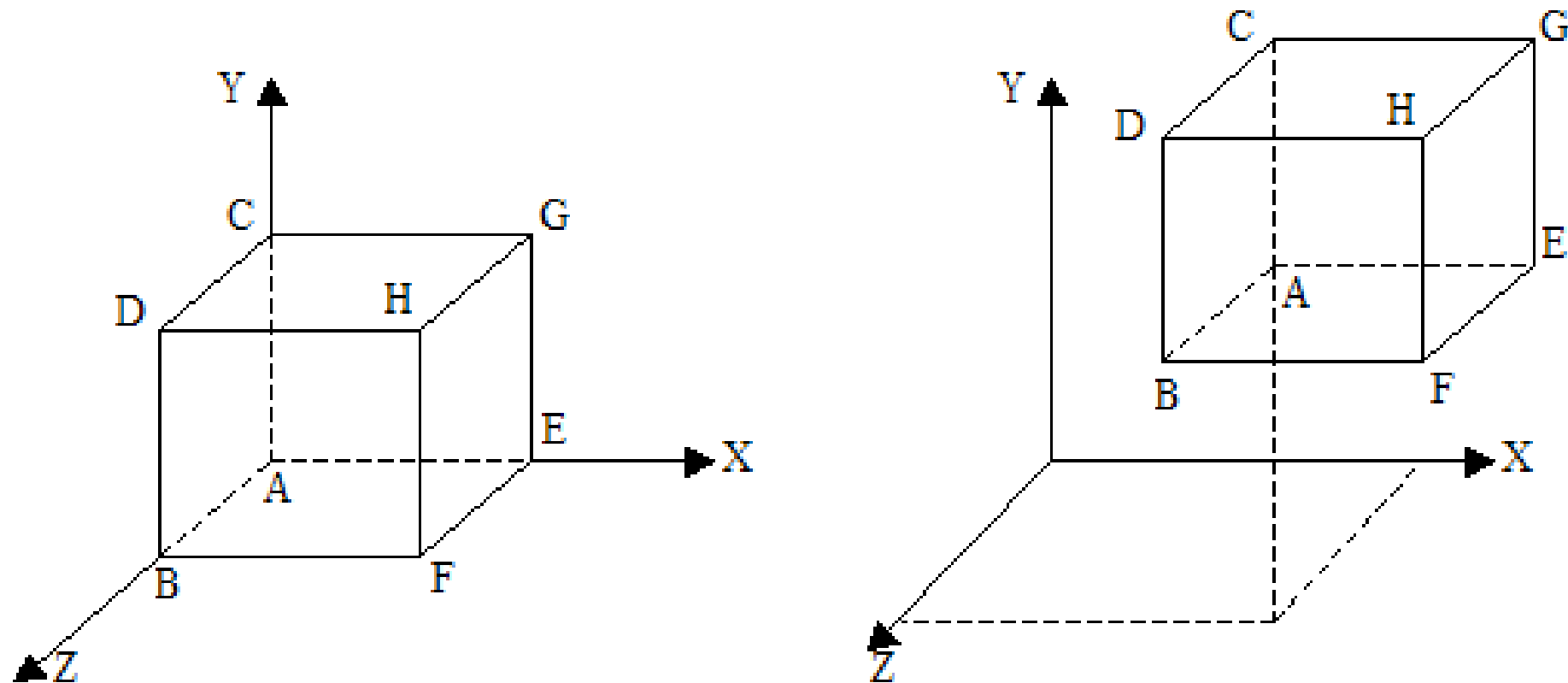


图 3.9 三维平移变换



## 2. 比例变换

$$P' = \begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$= P \cdot S(S_x, S_y, S_z) = \begin{bmatrix} S_x \cdot x & S_y \cdot y & S_z \cdot z & 1 \end{bmatrix}$$

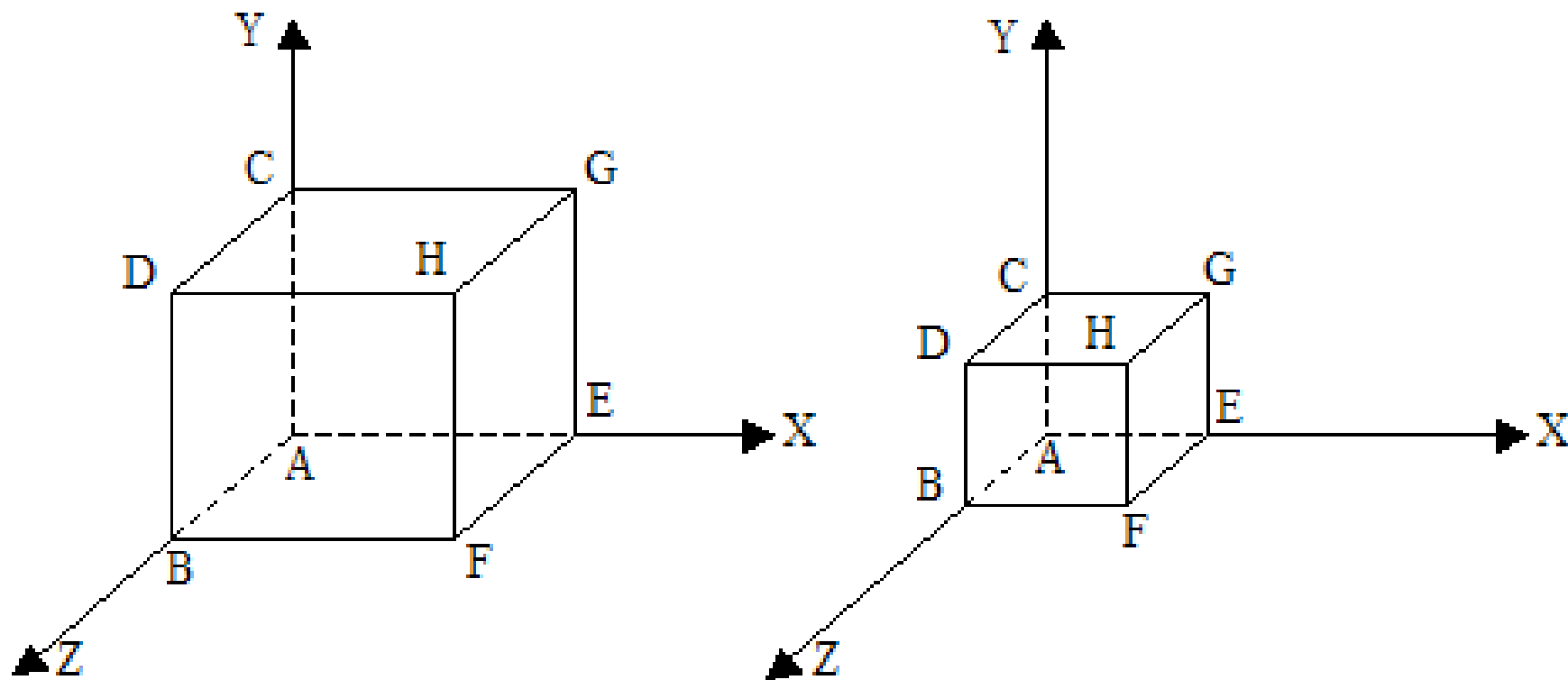


图 3.10 以坐标原点为参考点的三维比例变换

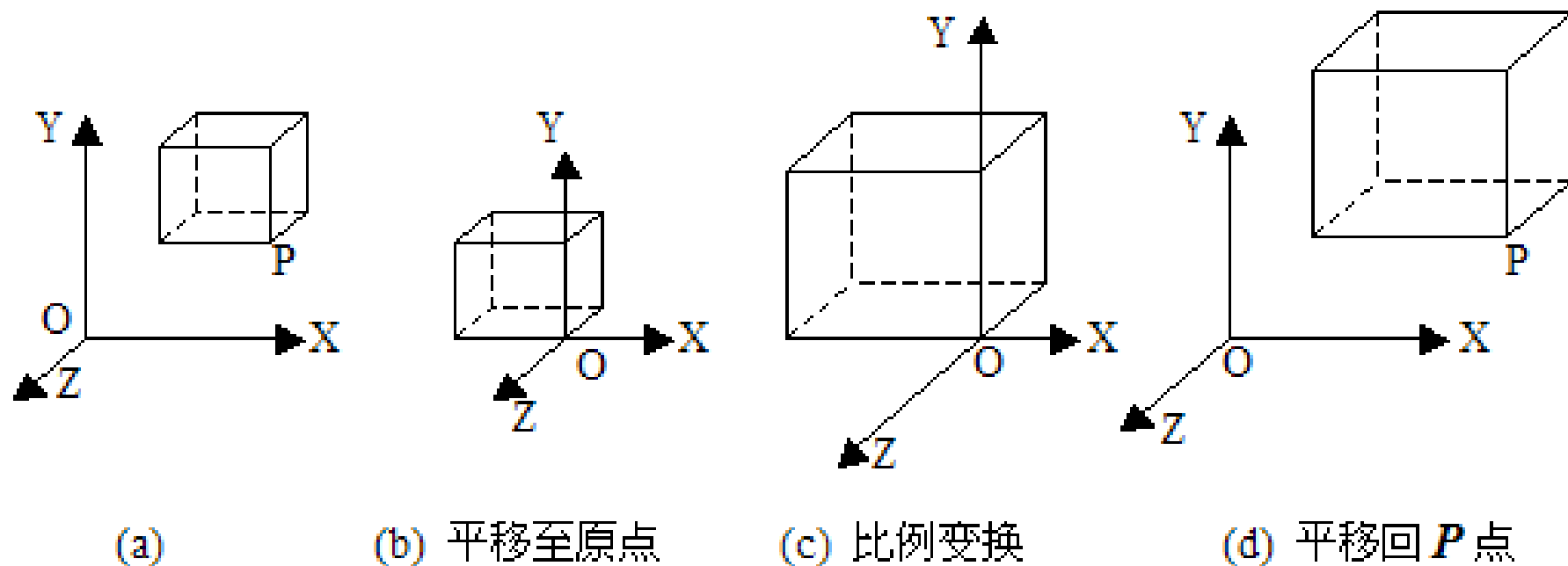


图 3.11 以任意点  $P$  为参考点的三维比例变换过程





### 3. 旋转变换

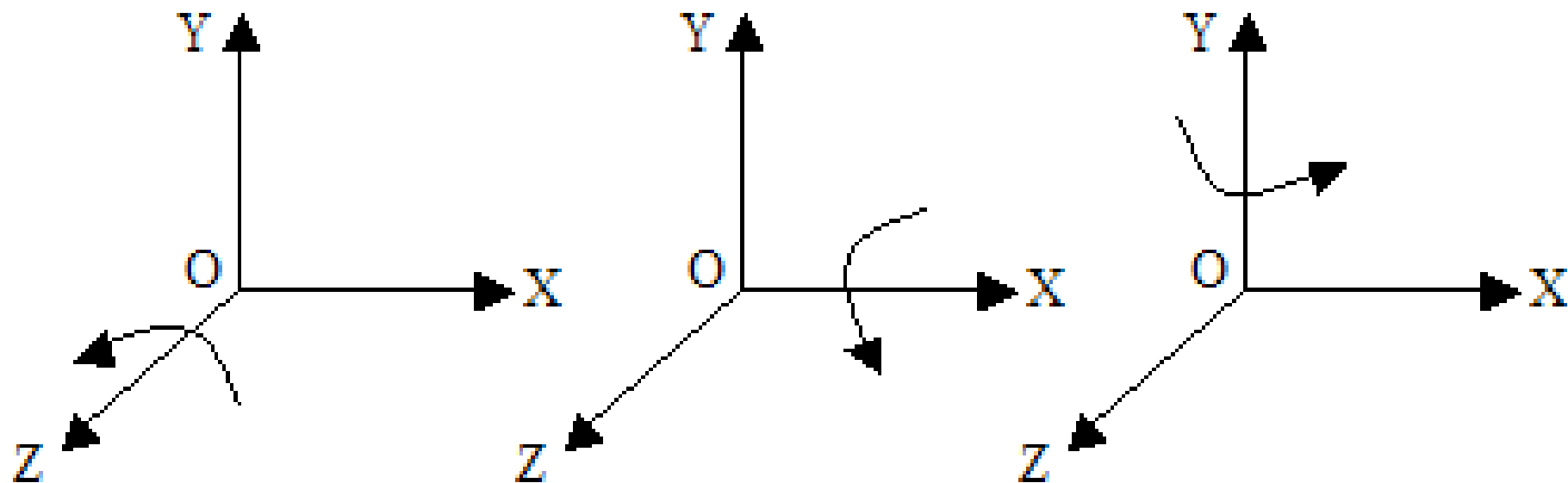


图 3.12 绕三根坐标轴旋转的正方向



# ① 绕Z轴旋转

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

$$z' = z$$

$$P' = \begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= P \cdot R_z(\theta) = \begin{bmatrix} x \cos \theta - y \sin \theta & x \sin \theta + y \cos \theta & z & 1 \end{bmatrix}$$



## ②绕X轴旋转

$$x' = x$$

$$y' = y \cos \theta - z \sin \theta$$

$$z' = y \sin \theta + z \cos \theta$$

$$P' = \begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= P \cdot R_x(\theta) = \begin{bmatrix} x & y \cos \theta - z \sin \theta & y \sin \theta + z \cos \theta & 1 \end{bmatrix}$$



### ③绕Y轴旋转

$$x' = x \cos \theta + z \sin \theta$$

$$y' = y$$

$$z' = -x \sin \theta + z \cos \theta$$

$$P' = \begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= P \cdot R_y(\theta) = \begin{bmatrix} x \cos \theta + z \sin \theta & y & -x \sin \theta + z \cos \theta & 1 \end{bmatrix}$$

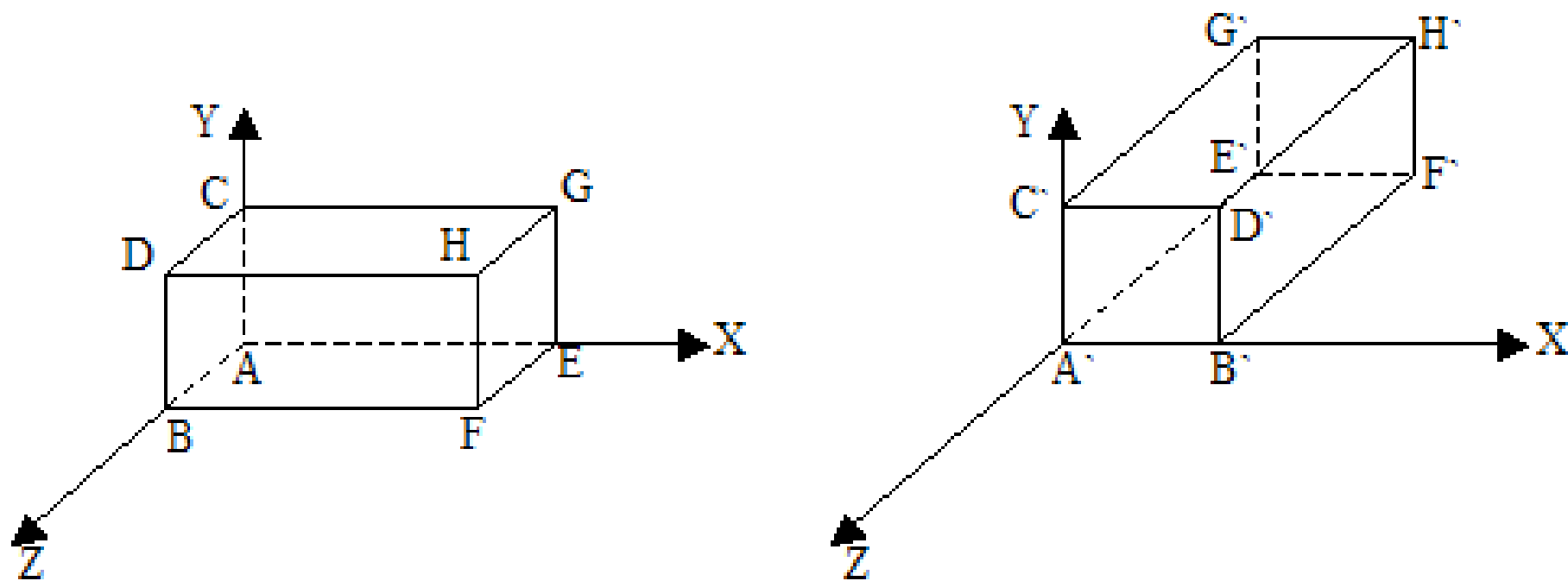


图 3.13 绕  $y$  轴旋转  $90^\circ$



**例：设三维空间中有一条任意直线，它由直线上一点Q和沿直线方向的单位方向向量n确定。**

**Q点坐标为** $(x_0, y_0, z_0)$

**直线向量** $\mathbf{n} = \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix}$   $|\mathbf{n}| = \sqrt{n_1^2 + n_2^2 + n_3^2} = 1$

**求绕这条直线旋转 $\theta$ 角的旋转变换矩阵。**

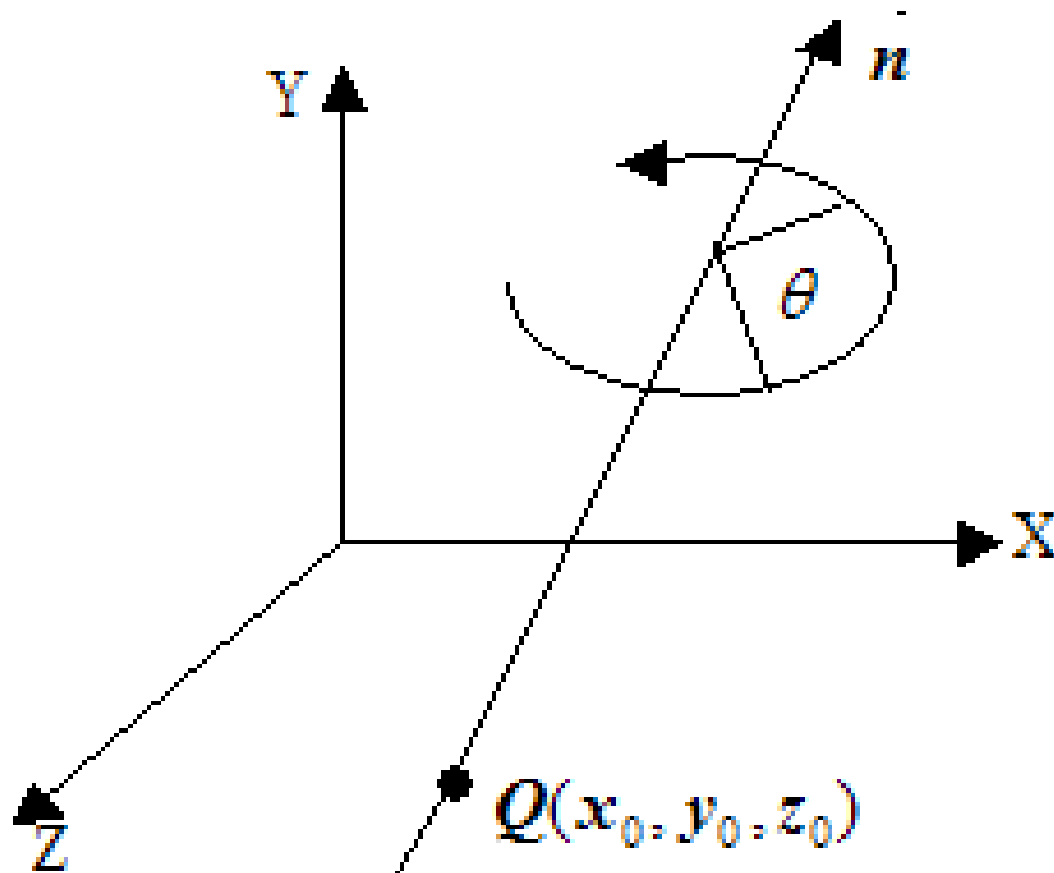


图 3.14 以空间任意直线为轴的三维旋转变换



1. 实现该变换可以先做平移变换，使旋转轴成为通过坐标原点的一条直线，然后做绕通过坐标原点的旋转轴旋转 角的旋转变换，最后再做平移变换，即开始所作平移变换的逆变换，使旋转轴平移回到原来的位置。

$$T(-x_0, -y_0, -z_0) \bullet R(\theta) \bullet T(x_0, y_0, z_0)$$





## 2. 过坐标原点的任意直线为旋转轴的旋转变换可分为五步实现：

- (1) 做绕X轴旋转  $\alpha$  角的变换，使旋转轴落在 $Y=0$ 上。
- (2) 做绕Y轴旋转  $\beta$  角的变换，使旋转轴与Z轴重合。
- (3) 做绕Z轴旋转  $\gamma$  角的旋转变换。
- (4) 做第2步的逆变换，即做绕Y轴 $-\beta$  旋转变换。
- (5) 做第1步的逆变换，即做绕X轴 $-\alpha$  旋转变换。



$$v = \sqrt{n_2^2 + n_3^2}$$

$$\cos \alpha = \frac{n_3}{v}, \quad \sin \alpha = \frac{n_2}{v}$$

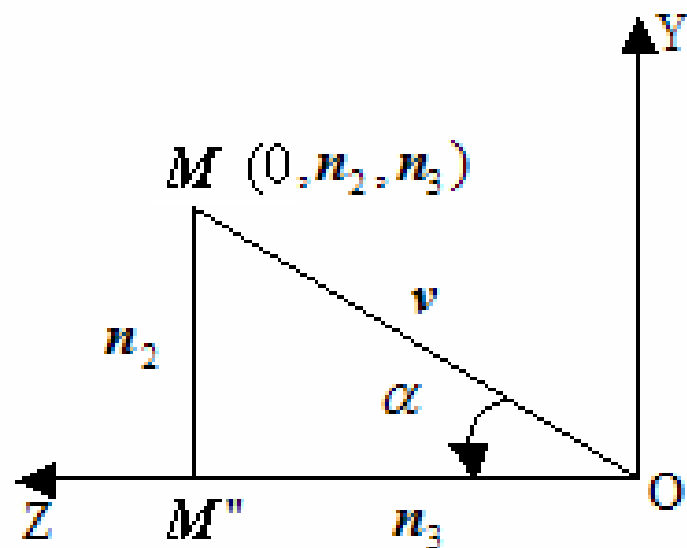
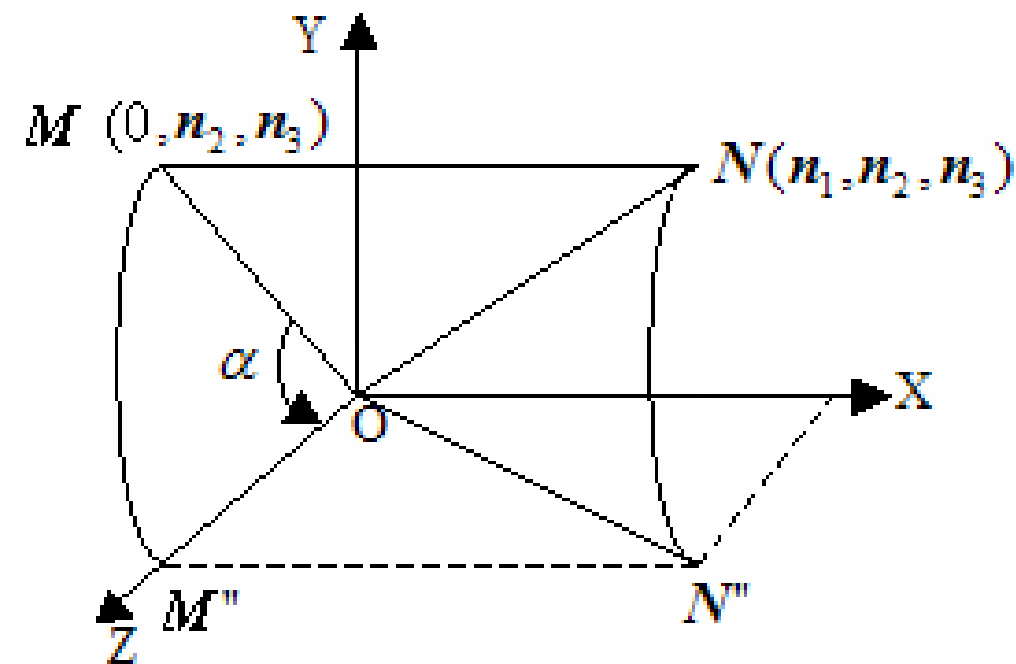


图 3.15 绕 X 轴旋转  $\alpha$  角 (右面是在  $X=0$  平面上的投影)



$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{n_3}{v} & \frac{n_2}{v} & 0 \\ 0 & -\frac{n_2}{v} & \frac{n_3}{v} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$N' = \begin{bmatrix} n_1 & n_2 & n_3 & 1 \end{bmatrix} R_x(\alpha)$$

$$= \begin{bmatrix} n_1 & 0 & v & 1 \end{bmatrix}$$



$$d = \sqrt{n_1^2 + n_2^2 + n_3^2} = 1$$

$$\cos \beta = \frac{v}{d} = v$$

$$\sin \beta = -\frac{n_1}{d} = -n_1$$

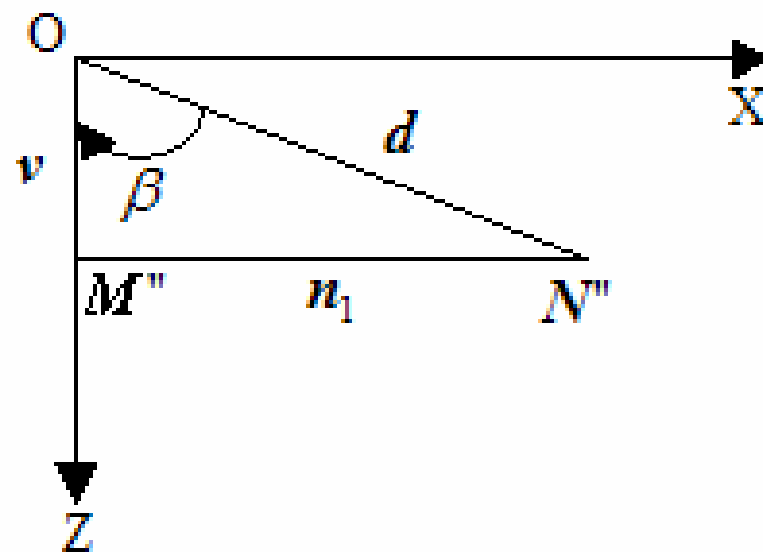
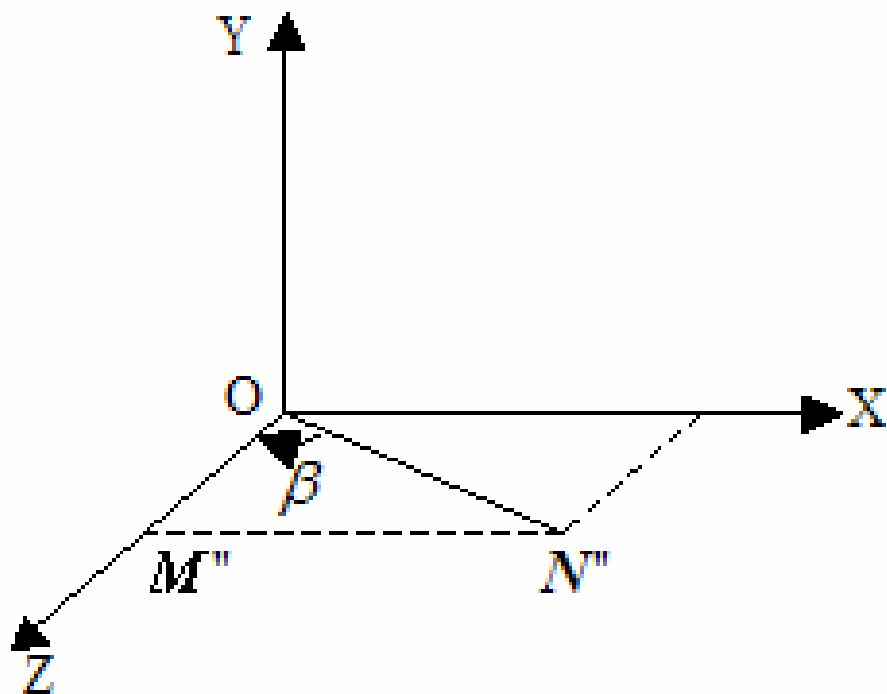


图 3.16 绕 Y 轴旋转  $\beta$  角（右面是在  $Y=0$  平面上的投影）



$$R_y(\beta) = \begin{bmatrix} v & 0 & n_1 & 0 \\ 0 & 1 & 0 & 0 \\ -n_1 & 0 & v & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} N'' &= \begin{bmatrix} n_1 & 0 & v & 1 \end{bmatrix} R_y(\beta) \\ &= \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix} \end{aligned}$$

$$R_z(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y(-\beta) = \begin{bmatrix} v & 0 & -n_1 & 0 \\ 0 & 1 & 0 & 0 \\ n_1 & 0 & v & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_x(-\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{n_3}{v} & -\frac{n_2}{v} & 0 \\ 0 & \frac{n_2}{v} & \frac{n_3}{v} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$





$$R(\theta) = R_x(\alpha) \bullet R_y(\beta) \bullet R_z(\theta) \bullet R_y(-\beta) \bullet R_x(-\alpha)$$

$$= \begin{bmatrix} n_1^2 + (1 - n_1^2) \cos \theta & n_1 n_2 (1 - \cos \theta) + n_3 \sin \theta & n_1 n_3 (1 - \cos \theta) - n_2 \sin \theta & 0 \\ n_1 n_2 (1 - \cos \theta) - n_3 \sin \theta & n_2^2 + (1 - n_2^2) \cos \theta & n_2 n_3 (1 - \cos \theta) + n_1 \sin \theta & 0 \\ n_1 n_3 (1 - \cos \theta) + n_2 \sin \theta & n_2 n_3 (1 - \cos \theta) - n_1 \sin \theta & n_3^2 + (1 - n_3^2) \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



# 三维图形变换

$$\begin{bmatrix} a & b & c & | & p \\ d & e & f & | & q \\ g & h & i & | & r \\ l & m & n & | & s \end{bmatrix}$$

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ l & m & n \end{bmatrix}$$

旋转，比例，对  
称，错切变换

平移变换

$p$

$q$

$r$

投影变换

$s$

整体比例变换





## 第五节 投影变换

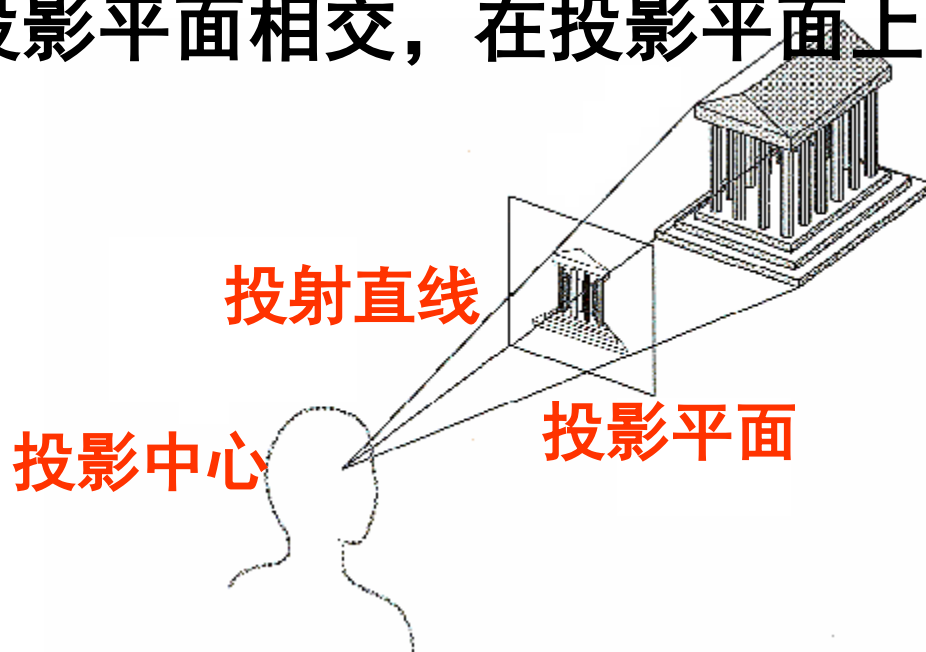
三维观察过程:

用户（世界）坐标系→观察坐标系（投影平面）

投影：把 $n$ 维空间中的点投射到小于 $n$ 维的空间中去

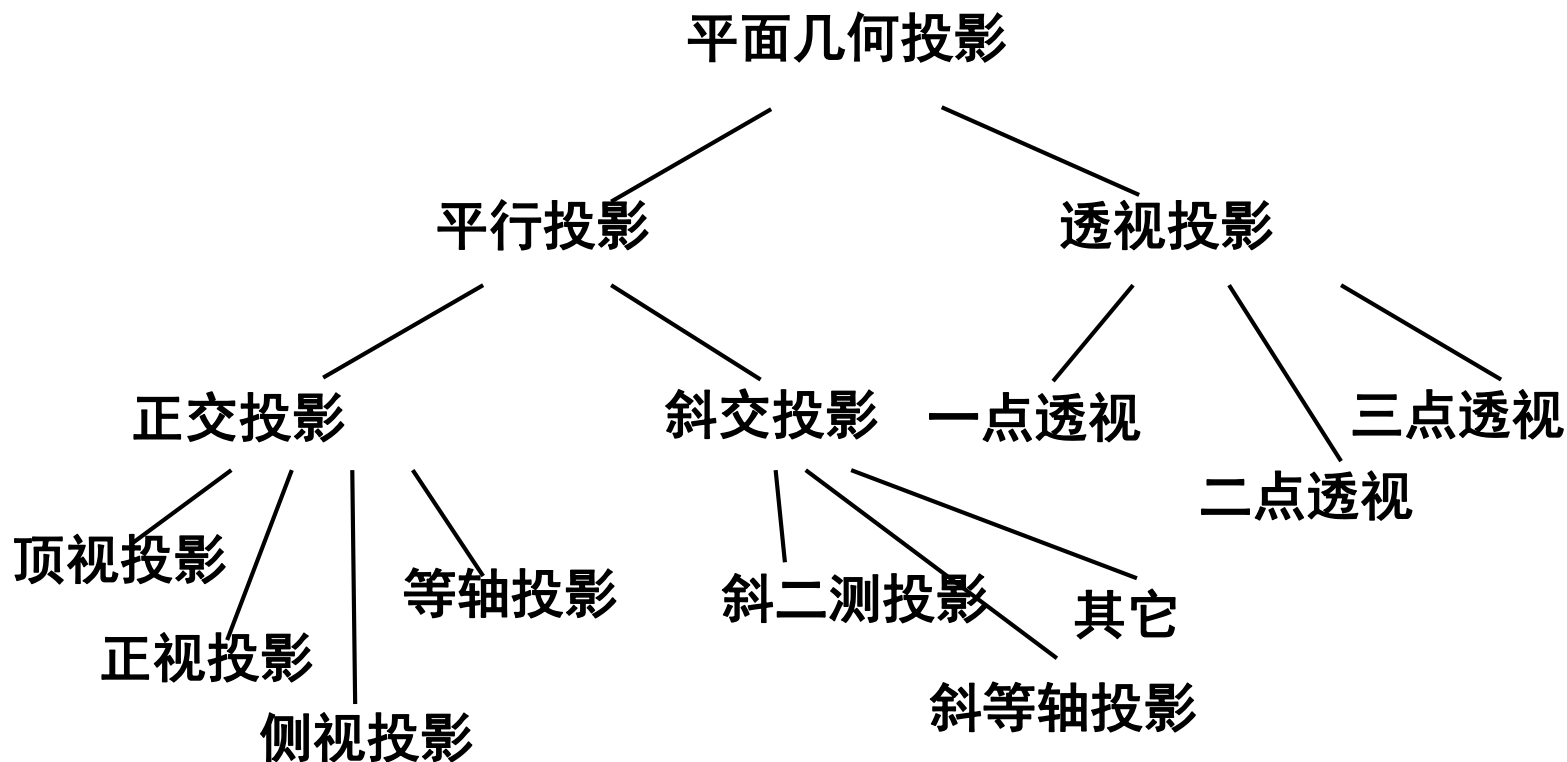
投影的形成：

- 1.在三维空间中确定一个**投影中心**和一个**投影平面**,
- 2.形体上的每一点与投影中心相连, 这些连线称为**投射直线**, 它们与投影平面相交, 在投影平面上就形成了形体的投影。





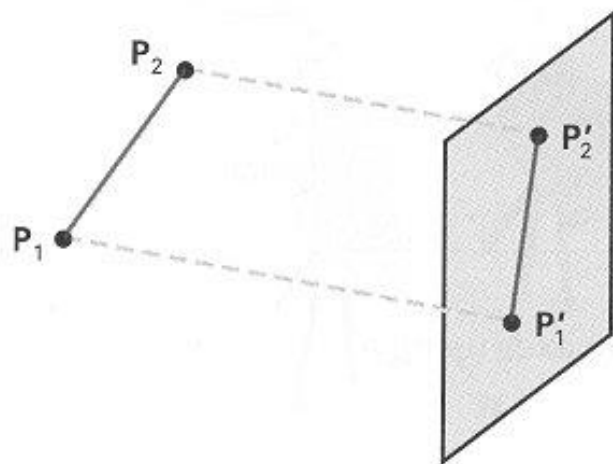
# 投影分类



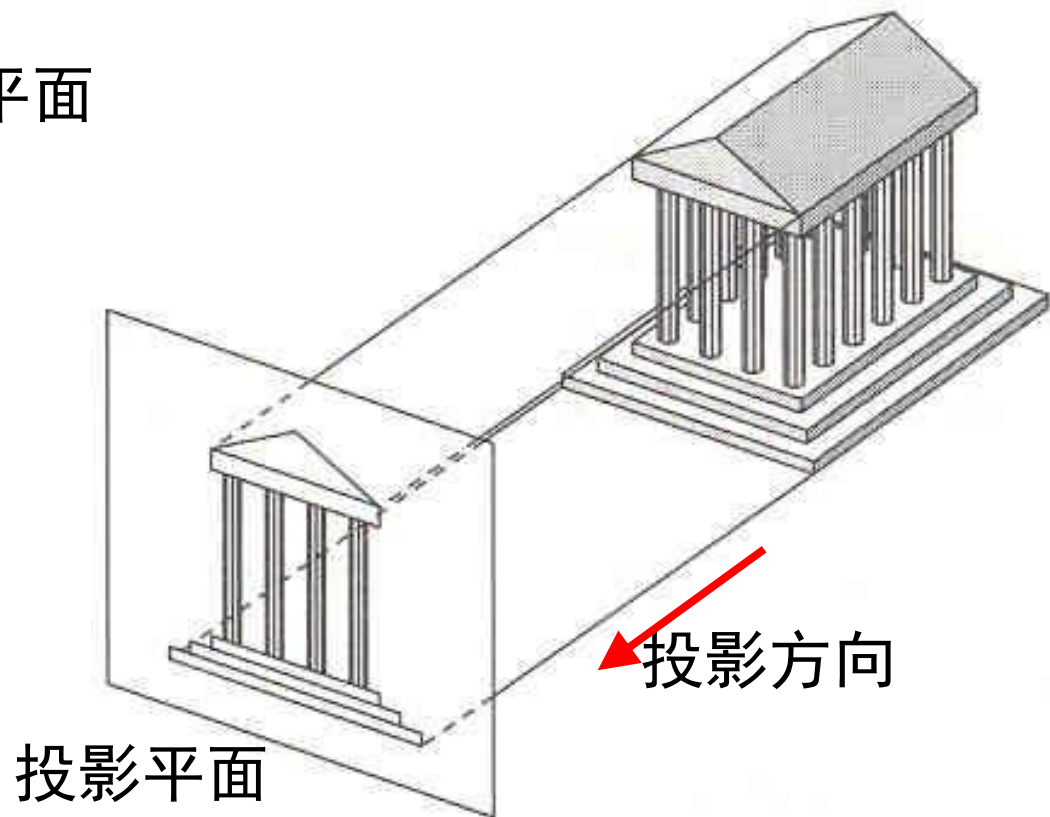


# 投影中心与投影平面之间距离为 $d$

$d$ 无限 **平行投影**，当投影中心与投影平面的距离为无穷远时，投射直线成为一组平行线。  
投射线平行，立体感差，能保持比例关系



投影平面

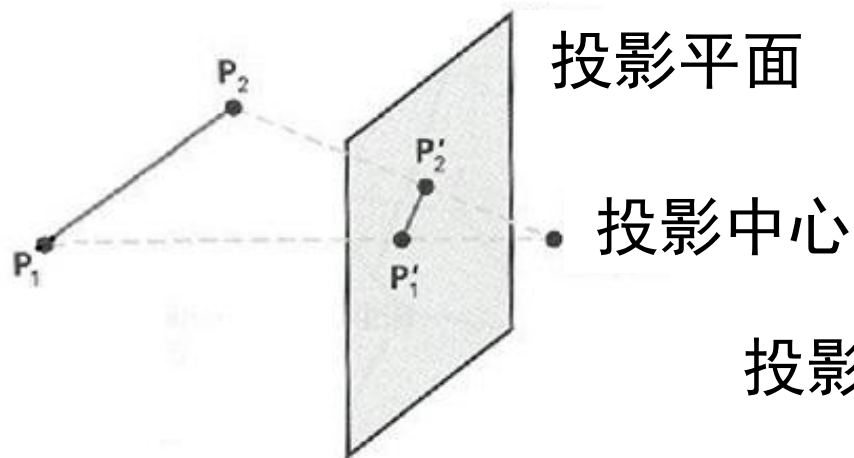




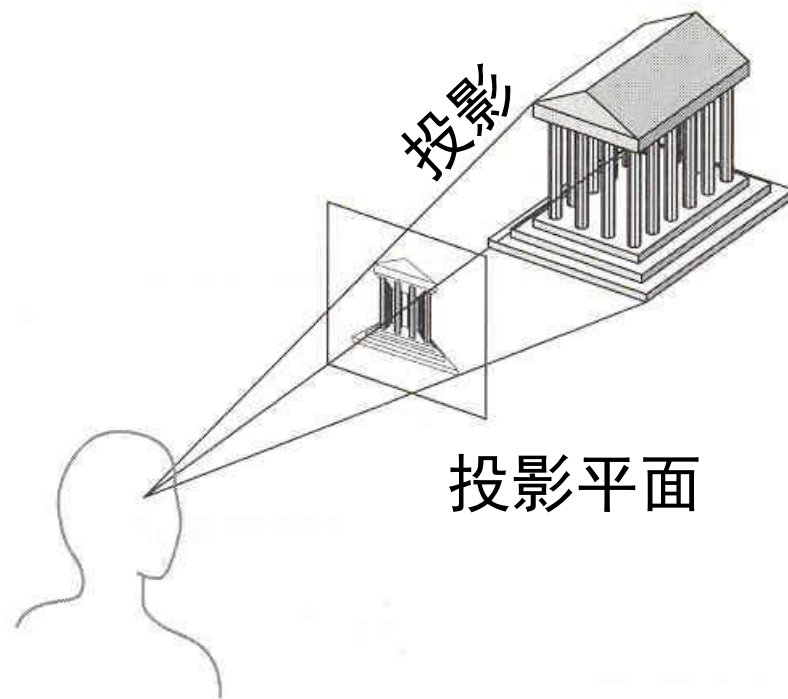
## 投影中心与投影平面之间距离为 $d$

$d$ 有限 **透视投影**，当投影中心与投影平面的距离是有限数值时，投射直线交于一点，形成灭点。

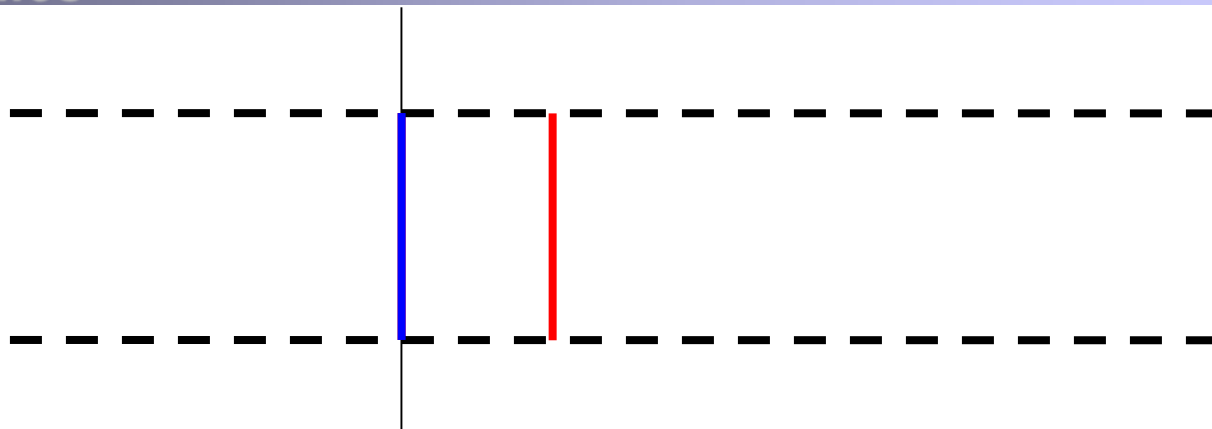
立体感强，更真实，但不能保持原来的比例关系



投影中心



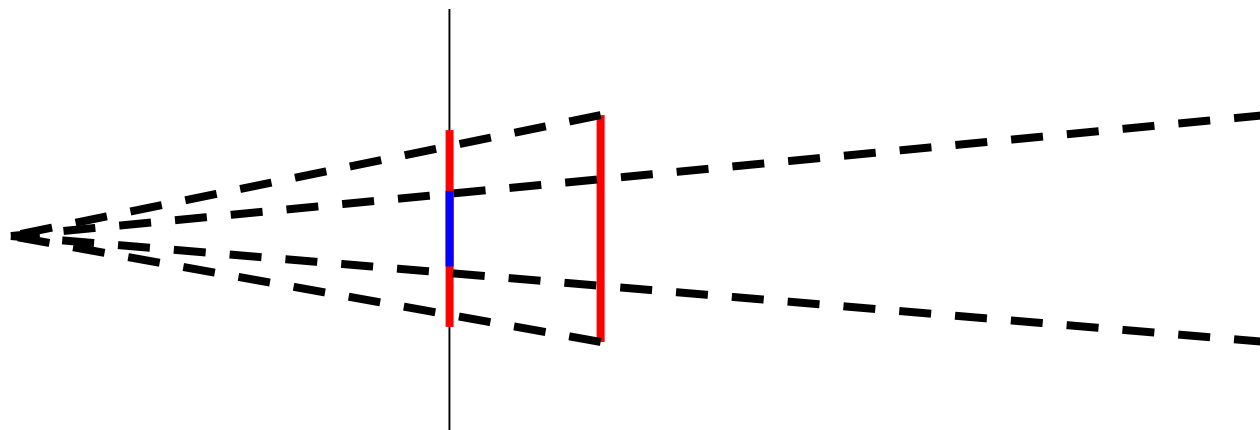
投影平面



投影平面

平行投影

投影中心



投影平面

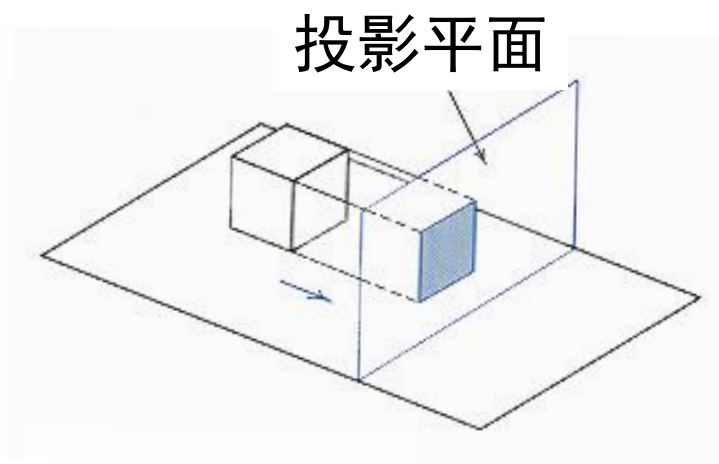
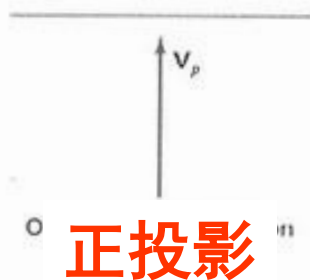
透视投影



# 一. 平行投影（正投影和斜投影）

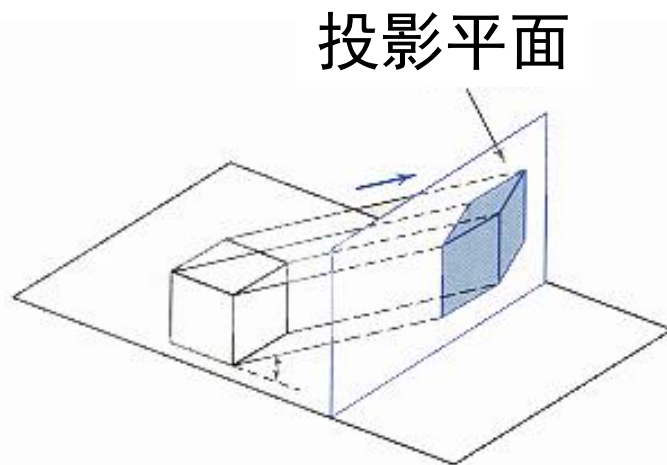
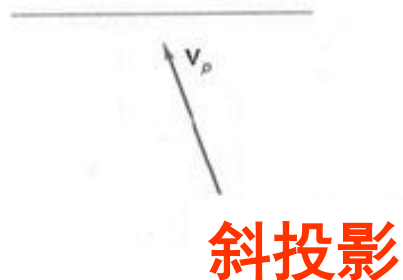
## • 正投影

– 垂直于投影平面



## • 斜投影

– 倾斜于投影平面





# 一. 平行投影（正投影和斜投影）

## 1. 正投影 投影方向与投影平面的法向相同

常见的正投影是正视投影、顶视投影和侧视投影

如果投影平面 $Z=0$ ，投影方向是沿 $Z$ 轴, 设三维空间中有点 $P$ ，普通坐标为 $(x,y,z)$ ，投影后，成为点 $P'$ ，普通坐标为 $(x',y',z')$  可知：

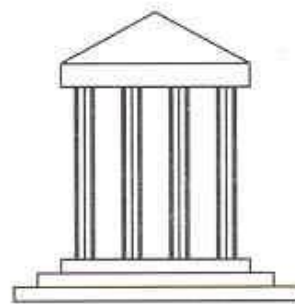
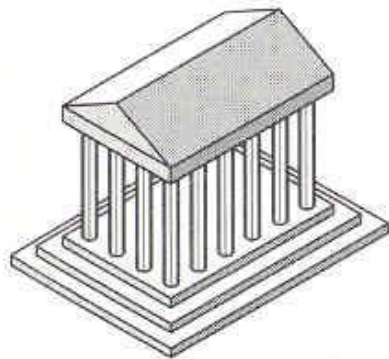
$$x' = x, \quad y' = y, \quad z' = 0$$

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & y & 0 & 1 \end{bmatrix}$$

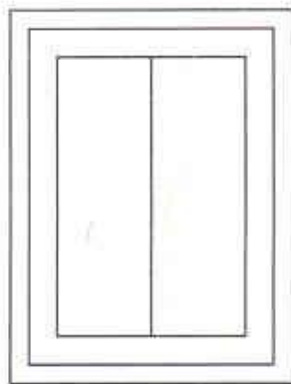


## 正投影

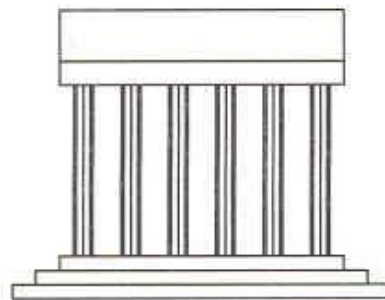
- 垂直于投影平面的投影方向



正视图



顶视图



侧视图



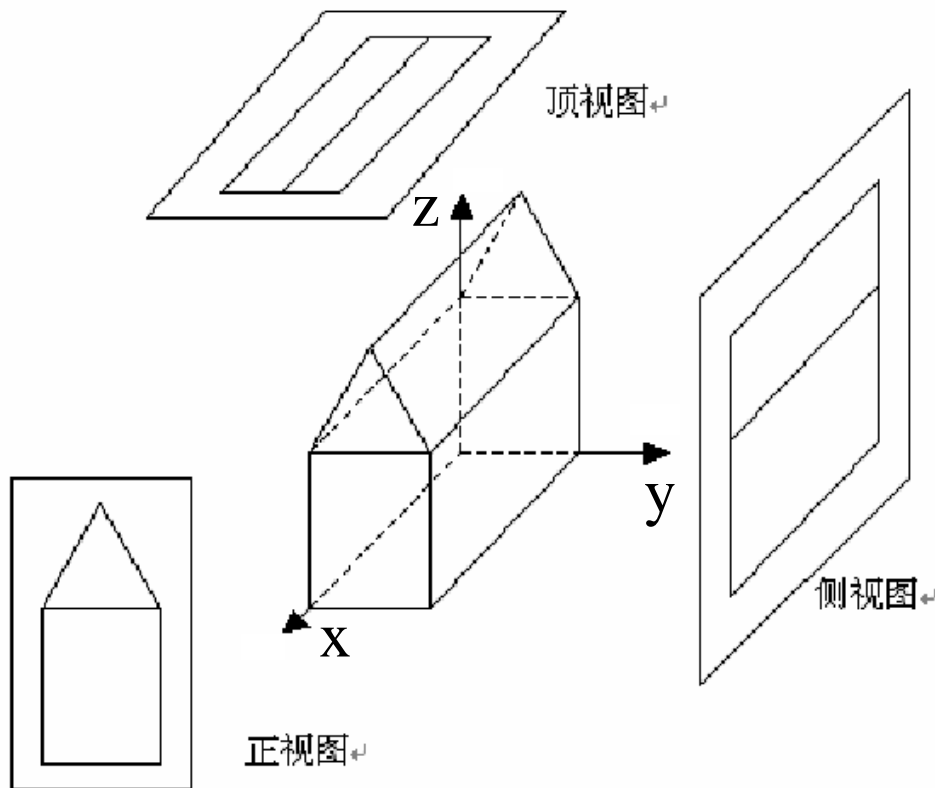


图 3.17 三视图

$$M_{\text{正}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_{\text{侧}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_{\text{顶}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



**正投影:** 投影平面垂直于坐标轴的正交投影。

**等轴投影:** 投影平面的法线方向，即投影方向与三个坐标轴的夹角都相等。这种投影能使在三个坐标轴方向上有相等的透视缩短。

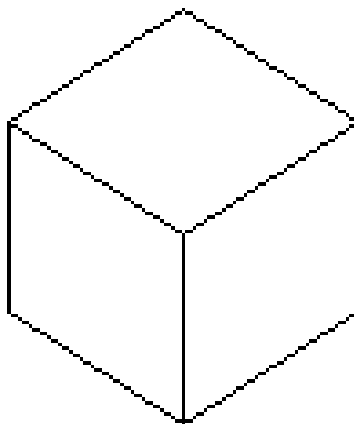
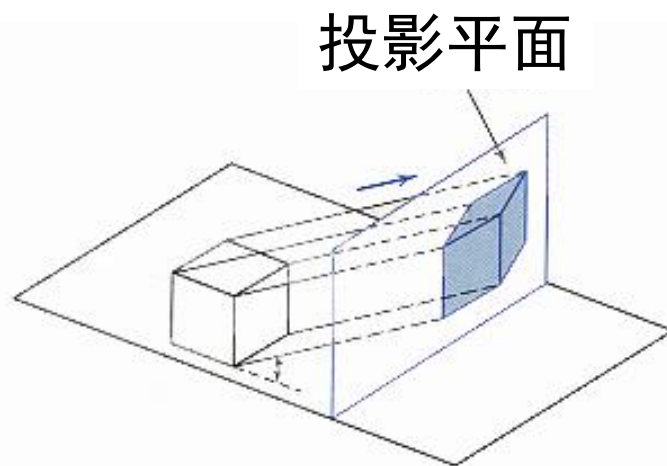
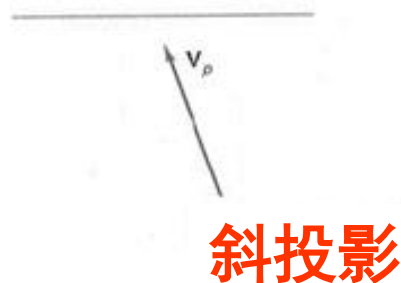
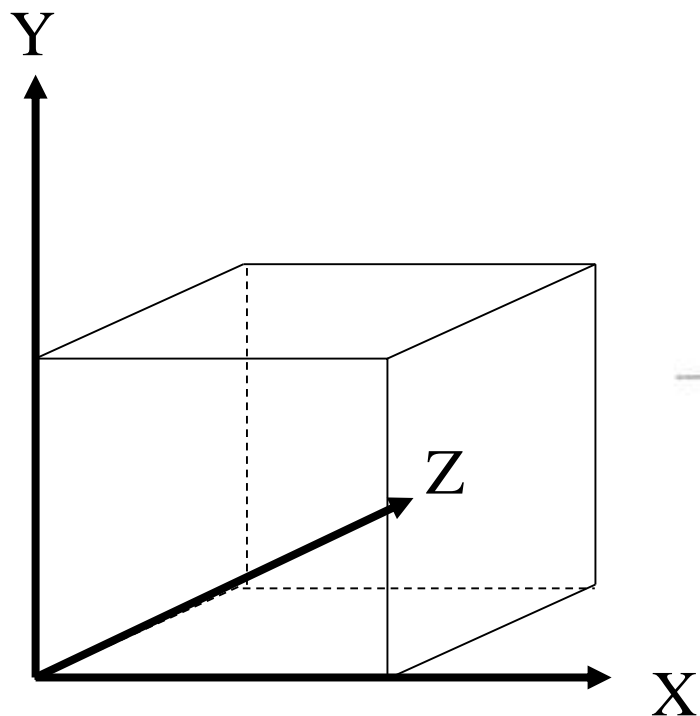


图 3.18 单位正方体的等轴投影图

## 2. 斜投影



当平行投影中投影平面的法线方向与投影方向不同时就得到斜交投影。在斜交投影中，投影平面一般取坐标平面。





设三维空间中有普通坐标为的任意一点  $(x, y, z)$ ，经斜交投影后所得投影点普通坐标为  $(x', y', z')$ 。显然  $z' = 0$ ，有：

$$\frac{x' - x}{z} = \frac{l \cos \alpha}{1}, \quad \frac{y' - y}{z} = \frac{l \sin \alpha}{1}$$

$$x' = x + z \bullet (l \bullet \cos \alpha), \quad y' = y + z \bullet (l \bullet \sin \alpha)$$

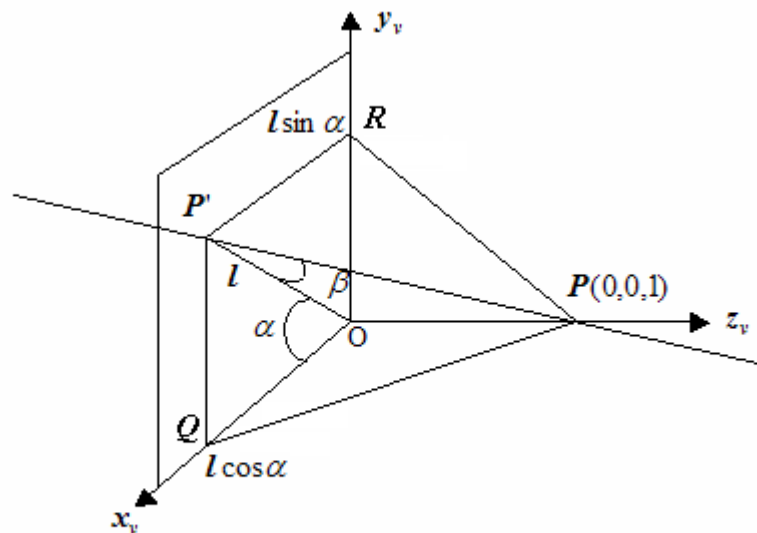


图 3.19 对点  $P(0,0,1)$  作斜平行投影到点  $P'(l \cos \alpha, l \sin \alpha, 0)$



设三维空间中有普通坐标为的任意一点  $(x, y, z)$ ，经斜交投影后所得投影点普通坐标为  $(x', y', z')$ 。显然  $z' = 0$ ，有：

$$\frac{x' - x}{z} = \frac{l \cos \alpha}{1}, \quad \frac{y' - y}{z} = \frac{l \sin \alpha}{1}$$

$$x' = x + z \bullet (l \bullet \cos \alpha), \quad y' = y + z \bullet (l \bullet \sin \alpha)$$

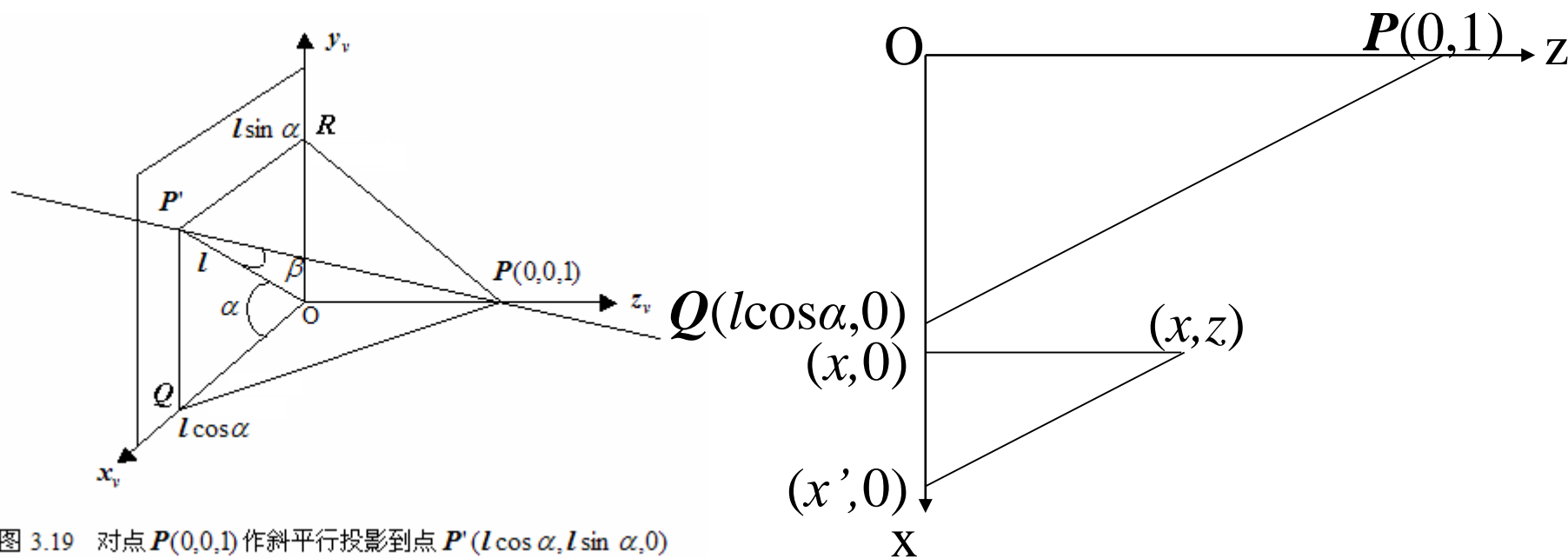


图 3.19 对点  $P(0,0,1)$  作斜平行投影到点  $P'(l \cos \alpha, l \sin \alpha, 0)$



设三维空间中有普通坐标为的任意一点  $(x, y, z)$ ，经斜交投影后所得投影点普通坐标为  $(x', y', z')$ 。显然  $z' = 0$ ，有：

$$\frac{x' - x}{z} = \frac{l \cos \alpha}{1}, \quad \frac{y' - y}{z} = \frac{l \sin \alpha}{1}$$

$$x' = x + z \bullet (l \bullet \cos \alpha), \quad y' = y + z \bullet (l \bullet \sin \alpha)$$

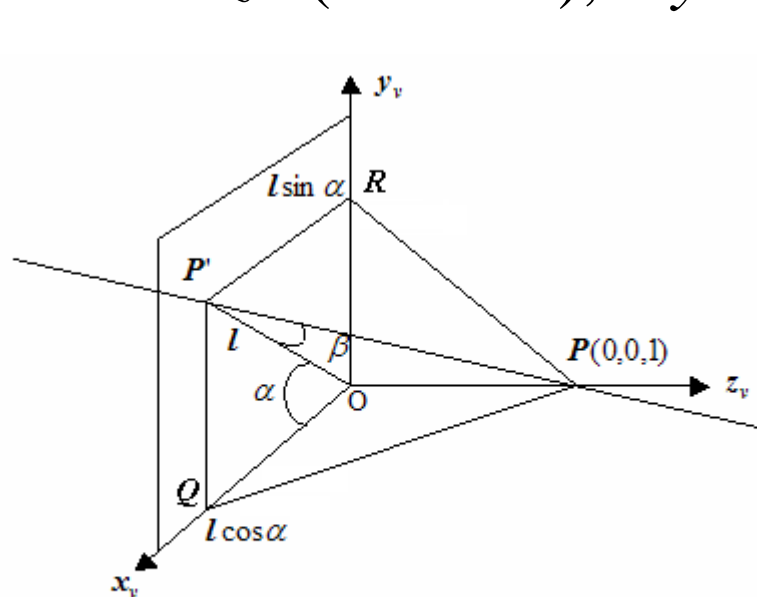
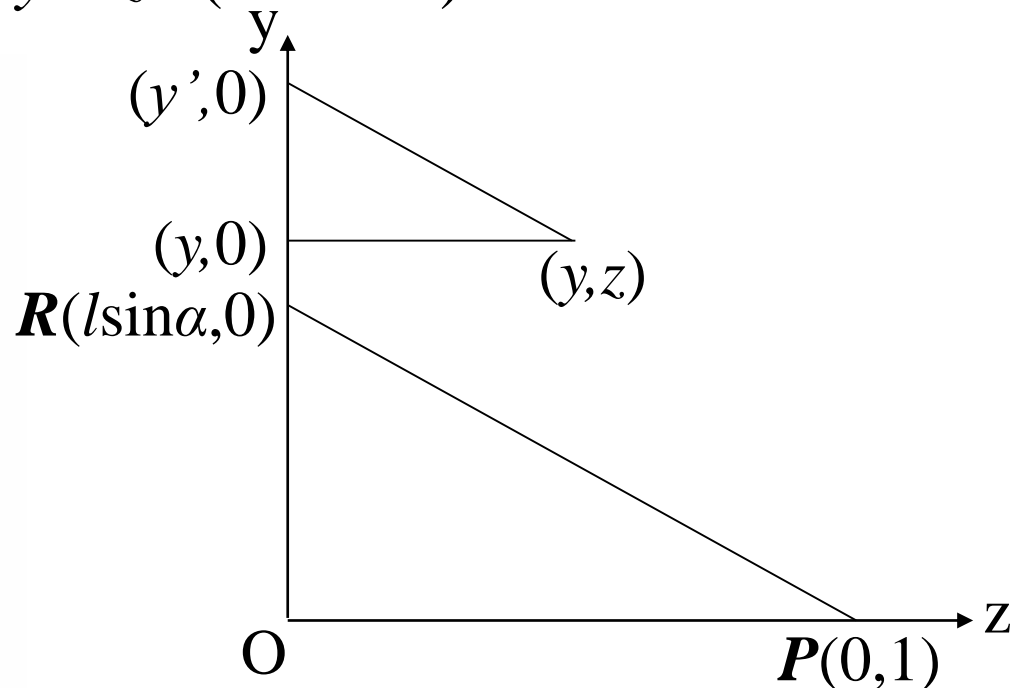


图 3.19 对点  $P(0,0,1)$  作斜平行投影到点  $P'(l \cos \alpha, l \sin \alpha, 0)$





$$x' = x + z \bullet (l \bullet \cos \alpha) \quad y' = y + z \bullet (l \bullet \sin \alpha)$$

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ l \cos \alpha & l \sin \alpha & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_{ob} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ l \cos \alpha & l \sin \alpha & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



**斜二测投影：**垂直于投影平面的线段长度缩短为原来的一半；

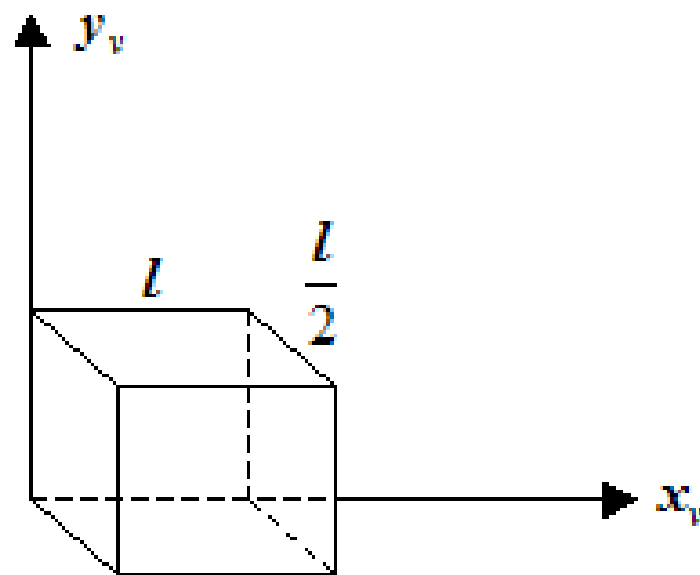
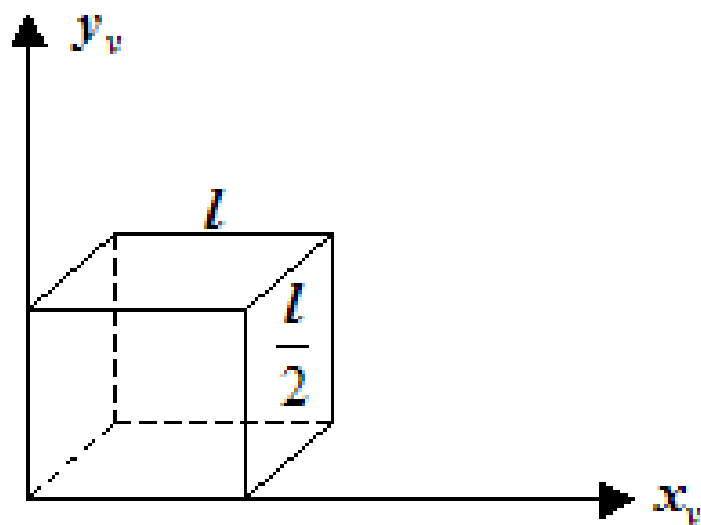
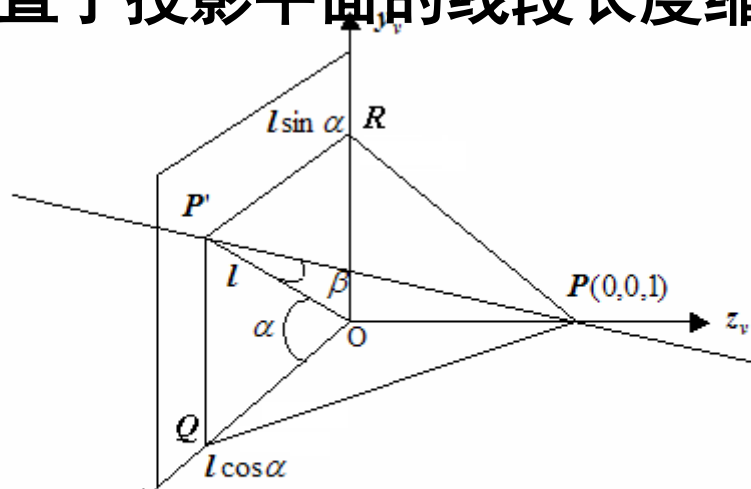


图 3.20 边长为  $l$  的正方体的两种斜二测投影的示意图





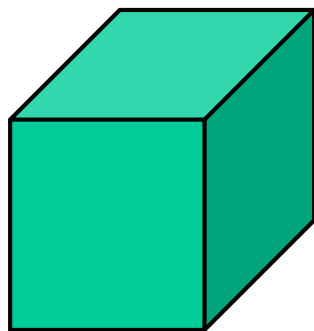
斜交投影中两个比较重要的情形是斜二测投影和斜等轴投影。

斜二测投影：垂直于投影平面的线段长度缩短为原来的一半；

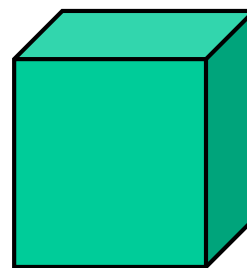
$$|OP'| = \frac{1}{2}|OP|, \tan \beta = 2, \beta = 63.4^\circ, l = \frac{1}{2}, \alpha \text{任意取}$$

斜等轴投影：使垂直于投影平面的线段仍保持长度。

$$|OP'| = |OP|, \beta = 45^\circ, l = 1, \alpha \text{任意取}$$



斜等轴投影  
投影方向( $45^\circ$ )



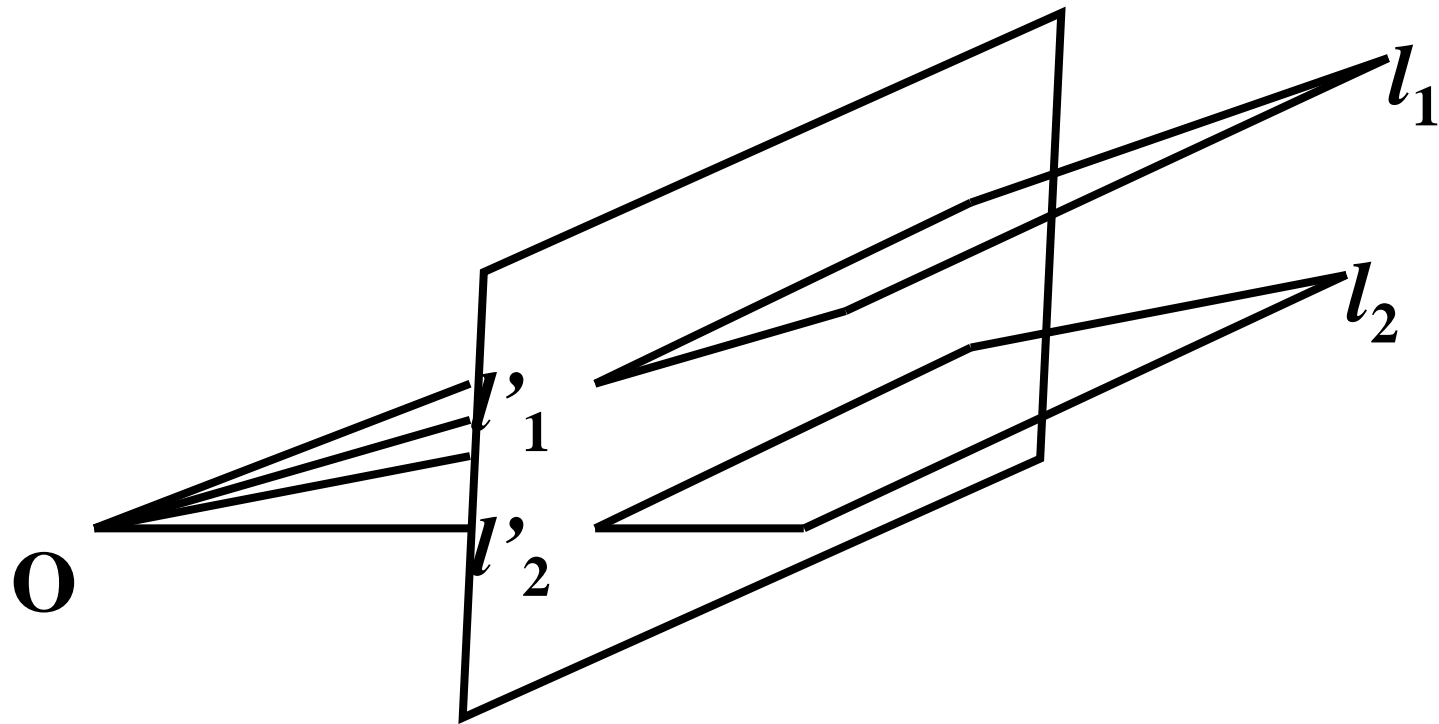
斜二测投影  
投影方向( $63.4^\circ$ )



## 二. 透视投影

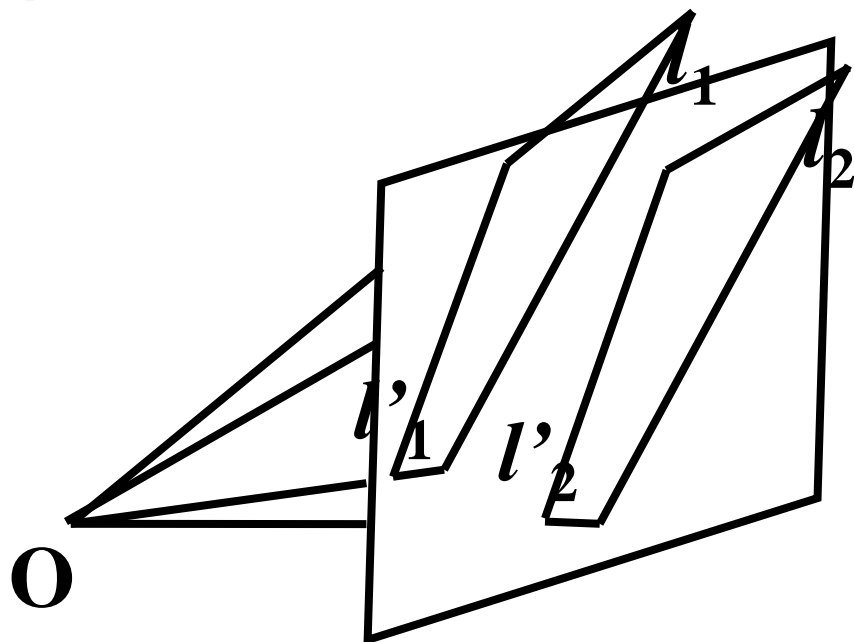
性质： 任意一组平行直线，如果平行于投影平面，则经透视投影后所得到的直线**或者重合，或者仍保持平行**；如果不平行于投影平面，将不再保持平行，并且必会汇聚于同一点，这个点称为**消失点**，也称为**灭点**。

空间中可以取得任意多组不平行于投影平面的平行直线，所以**消失点**也可以取得**任意多个**。



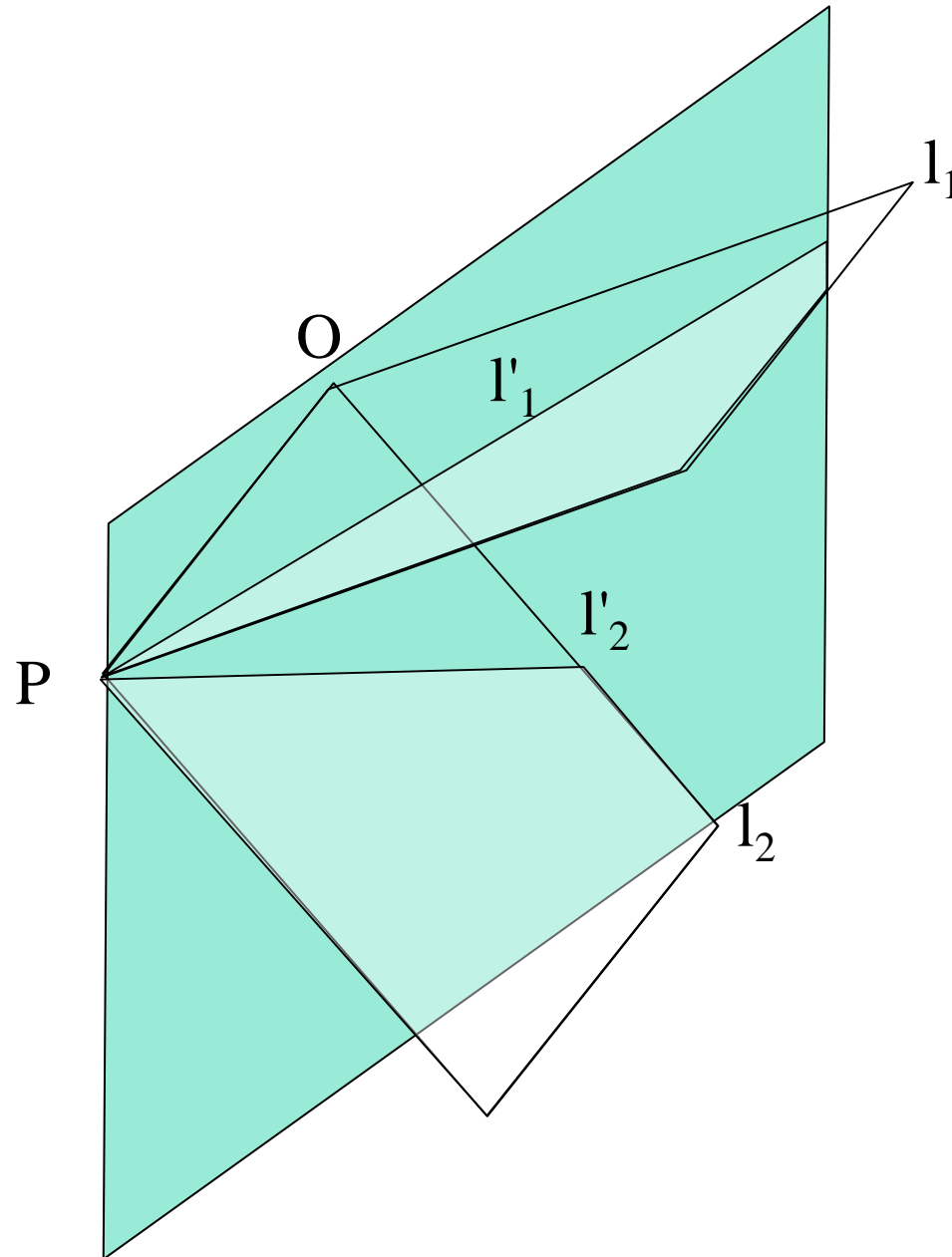
任意一组平行线，如果平行于投影平面，则投影后所得直线仍然保持平行。

投影中心 $O$ 和 $l_1$ 确定的平面和投影平面交于 $l'_1$ ，且 $l_1$ 平行于投影平面，所以 $l_1$ 平行于 $l'_1$ 。同理， $l_2$ 平行于 $l'_2$ 。又因 $l_1$ 平行于 $l_2$ ，所以 $l_2$ 平行于 $l'_1$ ， $l_1$ 平行于 $l'_2$ 。再次利用 $l_1$ 平行于 $l'_1$ ， $l_2$ 平行于 $l'_2$ ，可得 $l'_1$ 平行于 $l'_2$ 。



任意一组平行线，如果不平行于投影平面，则投影后所得直线会聚于称为消失点的同一点。

投影中心 $O$ 和 $l_1$ 确定的平面和投影平面交于 $l'_1$ ，和 $l_2$ 确定的平面和投影平面交于 $l'_2$ 。因为平行线 $l_1 l_2$ 与投影平面不平行，且 $l_1 l_2$ 分别与 $l'_1 l'_2$ 位于同一平面上，所以 $l_1 l_2$ 分别与 $l'_1 l'_2$ 相交。假设投影后所得的直线不相交，即 $l'_1$ 平行于 $l'_2$ ，又因 $l_1$ 平行于 $l_2$ ， $l_1 l_2$ 分别与 $l'_1 l'_2$ 相交，则推出 $l_1$ 与 $l'_1$ 确定的平面平行于 $l_2$ 与 $l'_2$ 确定的平面，但实际上两平面相交于经过透视点中心 $O$ 点的一条直线，矛盾。所以 $l'_1$ 不平行于 $l'_2$ ，即投影后所得直线相交。





**主消失点：**平行于一个坐标轴中的一组平行线对应的消失点称为**主消失点**。

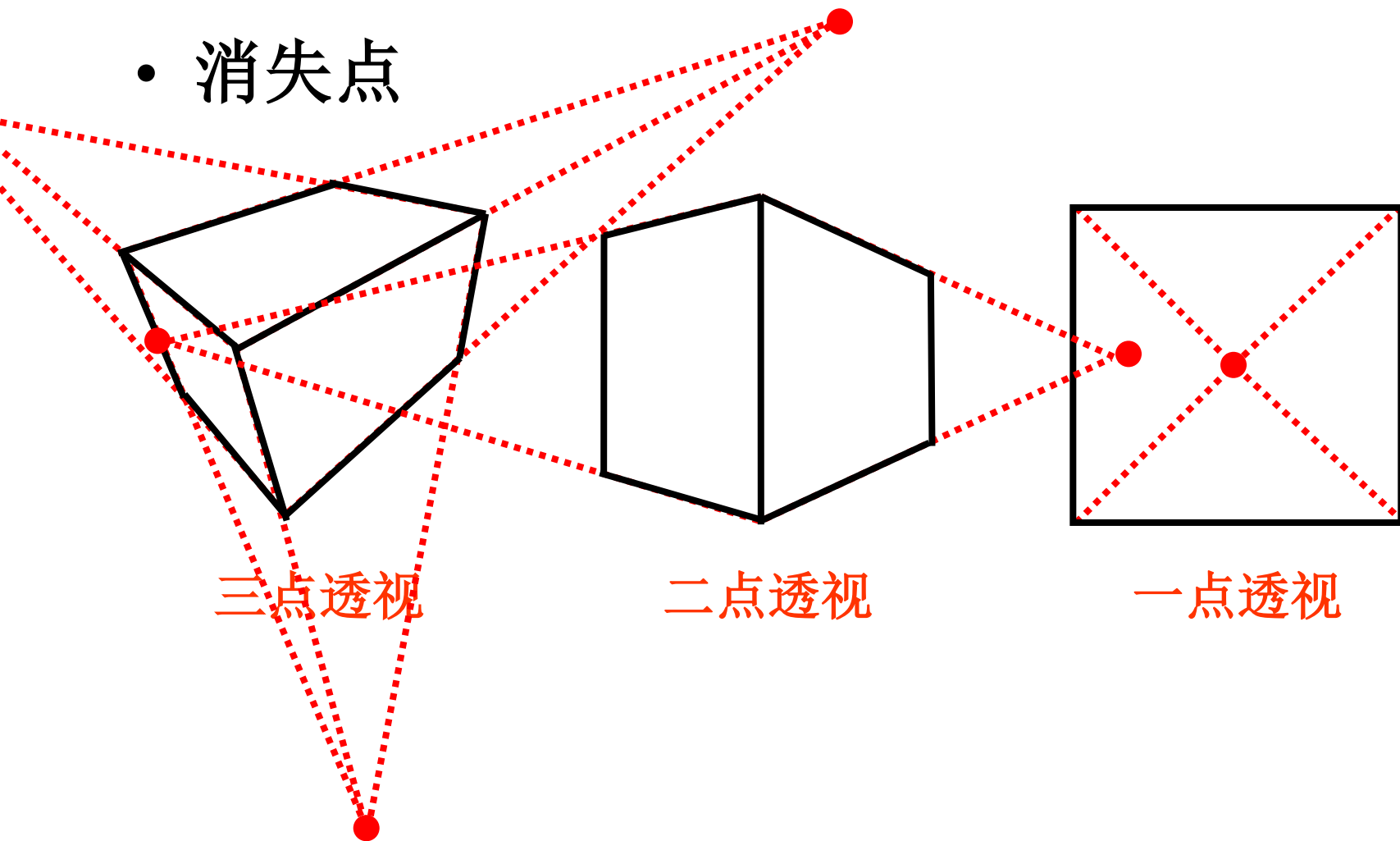
因为只有三个坐标轴，所以最多只有三个主消失点。根据**主消失点的数目**，透视投影可以分为**一点透视、二点透视、三点透视**。

如果投影平面截Z轴并与它垂直，这时就只能在Z轴方向上有主消失点。



# 透视投影

- 消失点





# 单消失点的透视投影计算

左手坐标系

投影中心是坐标原点

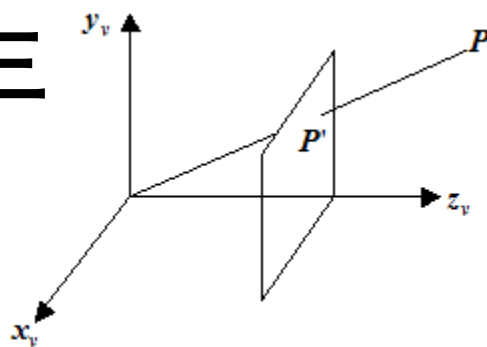
投影平面垂直于Z轴

设投影平面位于处 $Z=d$ ，可设 $d>0$ 。对空间中任意一点 $P$ ，其普通坐标为 $(x,y,z)$ ，它在投影平面上的投影点 $P'$ 的普通坐标为 $(x_p, y_p, z_p)$

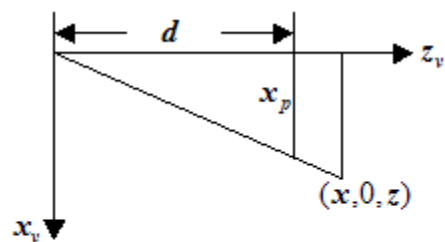




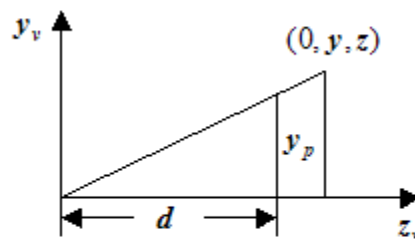
利用后两个图中三角形的相似关系，  
可以得出：



(1) 求点  $P$  的透视投影  $P'$



(2) 沿  $y_v$  轴顶视



(3) 沿  $x_v$  轴侧视

图 3.23 单消失点透视投影计算

$$\frac{x_p}{d} = \frac{x}{z} \Rightarrow x_p = \frac{d \bullet x}{z} = \frac{x}{z/d}$$

$$\frac{y_p}{d} = \frac{y}{z} \Rightarrow y_p = \frac{d \bullet y}{z} = \frac{y}{z/d}, z_p = d$$



使用齐次坐标可以得出：

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{d} \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} x & y & z & z/d \end{bmatrix}$$

对齐次坐标进行规范化后得：

$$\begin{bmatrix} \frac{x}{z/d} & \frac{y}{z/d} & d & 1 \end{bmatrix}$$



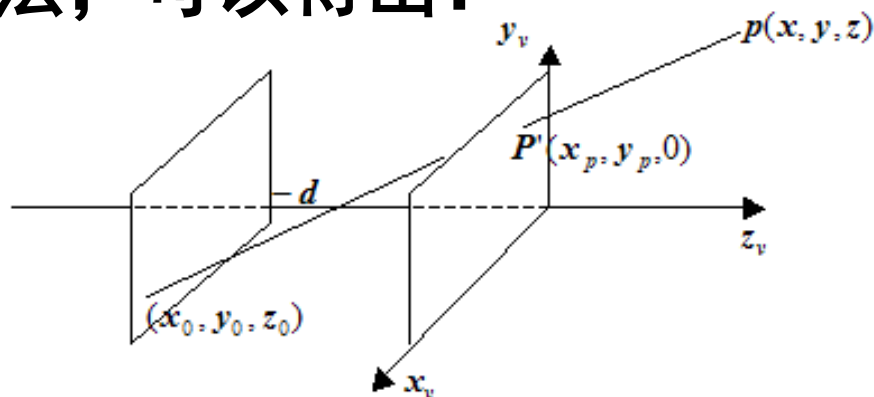
透视投影变换矩阵为:

$$M_{per-z} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{d} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

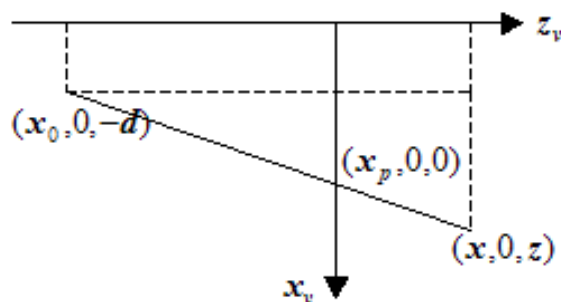
同理易求当投影平面是平面  $x=d$  和  $y=d$  时, 此时的透视投影变换矩阵分别为:

$$M_{per-x} = \begin{bmatrix} 1 & 0 & 0 & \frac{1}{d} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad M_{per-y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{1}{d} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

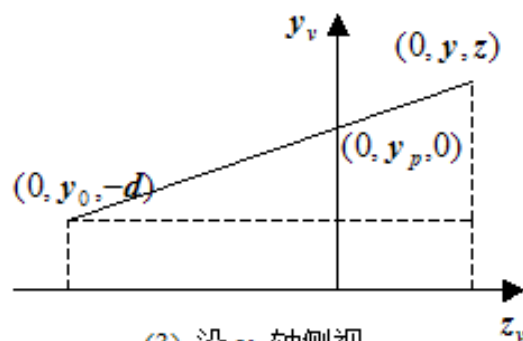
实用中,常取 $z=0$ 为投影平面,这时投影中心可取空间中任意一点 $(x_0, y_0, z_0)$ 。假定 $z_0$ 是一个负数,即 $z_0=-d$ ,  $d>0$ 。用前面相同的方法,可以得出:



(1) 求点  $P$  的透视投影  $P'$



(2) 沿  $y_v$  轴顶视



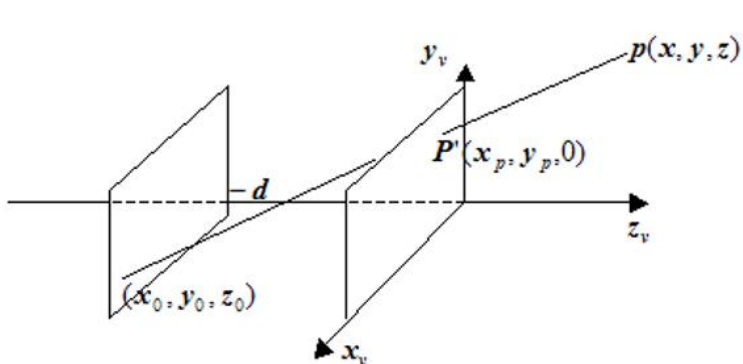
(3) 沿  $x_v$  轴侧视

图 3.24 投影平面是  $z = 0$ , 投影中心是  $(x_0, y_0, -d)$  时的透视投影

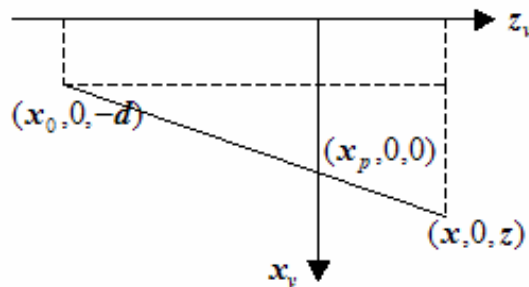
实用中,常取 $z=0$ 为投影平面,这时投影中心可取空间中任意一点 $(x_0, y_0, z_0)$ 。假定 $z_0$ 是一个负数,即 $z_0=-d$ ,  $d>0$ 。用前面相同的方法,可以得出:

$$\frac{x_p - x_0}{d} = \frac{x - x_0}{z + d}, \quad \frac{y_p - y_0}{d} = \frac{y - y_0}{z + d}$$

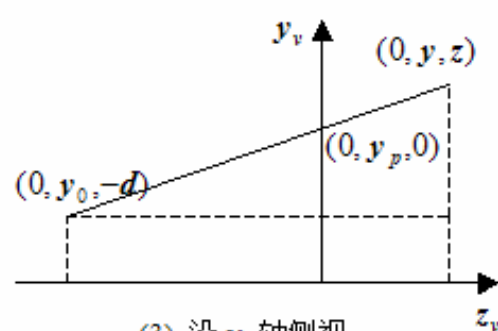
$$x_p = x_0 + \frac{d \bullet (x - x_0)}{z + d} \quad y_p = y_0 + \frac{d \bullet (y - y_0)}{z + d}$$



(1) 求点  $P$  的透视投影  $P'$



(2) 沿  $y$  轴顶视



(3) 沿  $x$  轴侧视

图 3.24 投影平面是  $z=0$ , 投影中心是  $(x_0, y_0, -d)$  时的透视投影



$$M'_{per} = T(-x_0, -y_0, d) \cdot M_{per-z} \cdot T(x_0, y_0, -d)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_0 & -y_0 & d & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{d} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_0 & y_0 & -d & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{x_0}{d} & \frac{y_0}{d} & 0 & \frac{1}{d} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



可以用齐次坐标验证，有：

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{x_0}{d} & \frac{y_0}{d} & 0 & \frac{1}{d} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x + \frac{x_0 \bullet z}{d} & y + \frac{y_0 \bullet z}{d} & 0 & \frac{z}{d} + 1 \end{bmatrix}$$

将结果规范化后得到齐次坐标为：

$$\begin{bmatrix} x_0 + \frac{d \bullet (x - x_0)}{z + d} & y_0 + \frac{d \bullet (y - y_0)}{z + d} & 0 & 1 \end{bmatrix}$$



# 投影总结:

## 1 平行投影

- 1) 正交: 正投影, 等轴
- 2) 斜交: 斜二测, 斜等轴

## 2 透视投影

- 1) 一点透视
- 2) 二点透视
- 3) 三点透视



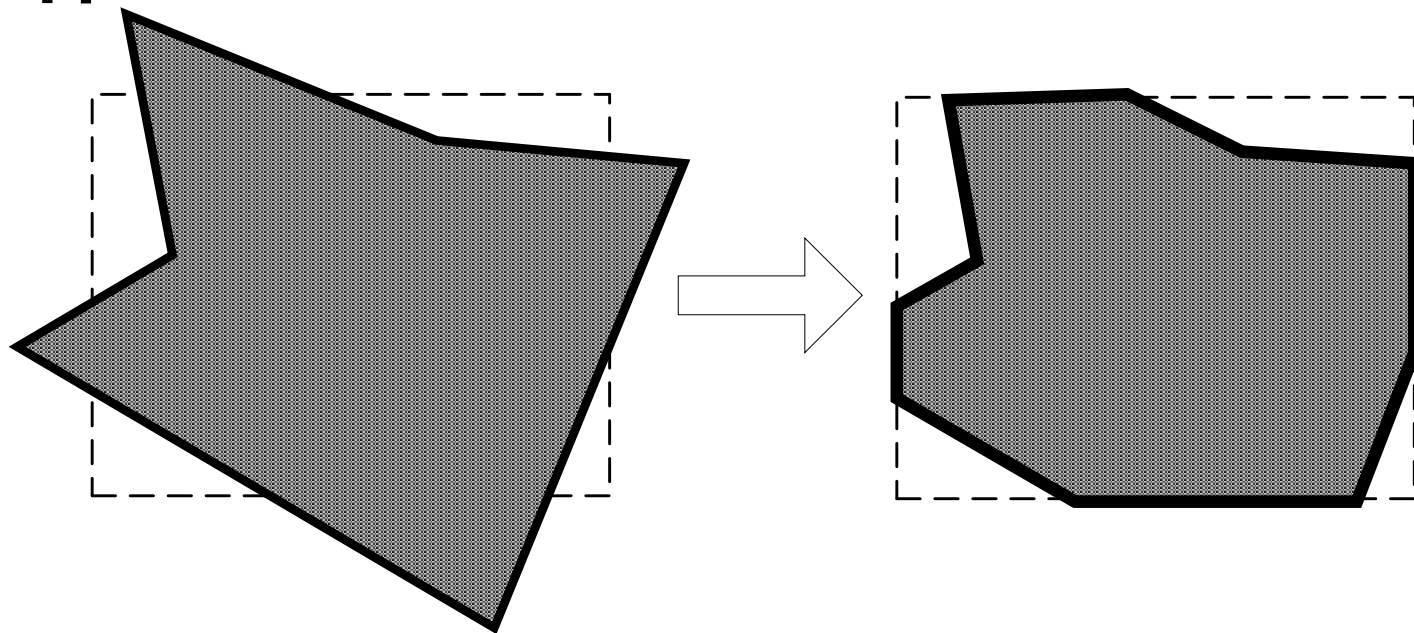


## 第六节 裁剪

**裁剪**就是去掉窗口外的不可见部分，保留窗口内的可见部分的过程。

**裁剪区域**：矩形、任意图形

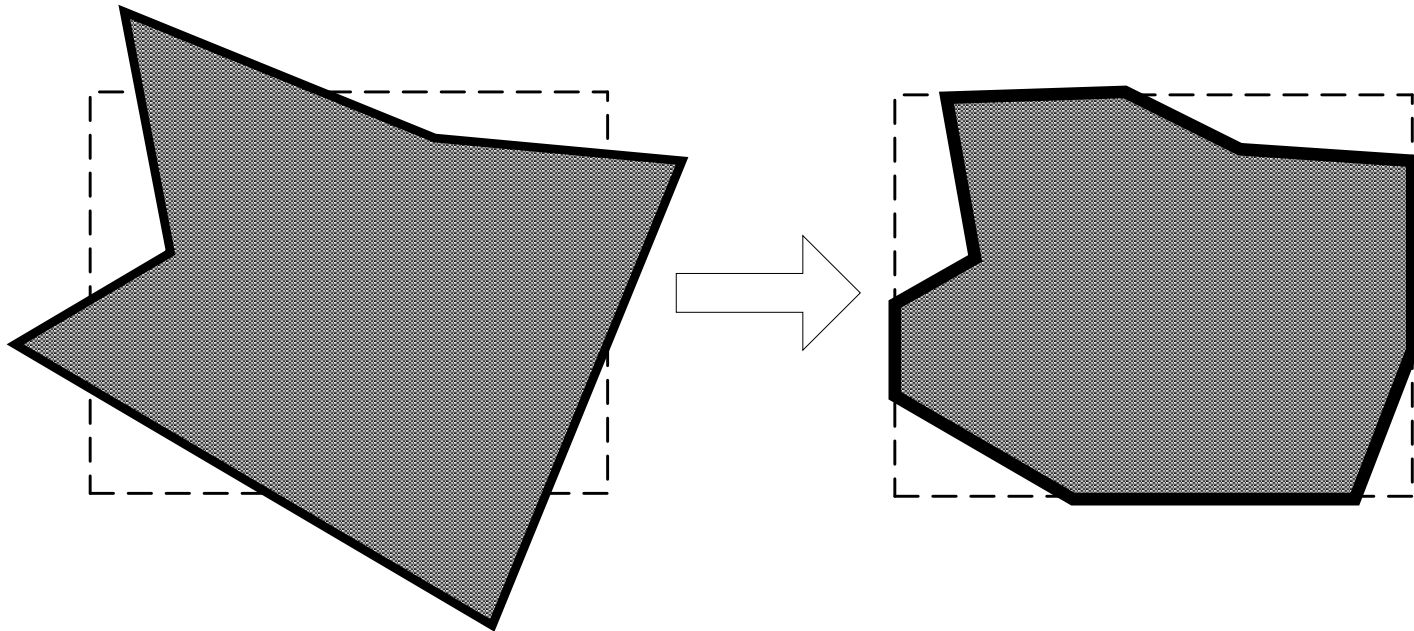
**裁剪对象**：点、线段、多边形、二维或三维形体





假设窗口的两个对角顶点分别是  $(x_l, y_b)$ 、 $(x_r, y_t)$ ，则同时满足下列不等式的点  $(x, y)$  是要保留的点，否则就要被舍弃：

$$x_l \leq x \leq x_r, \quad y_b \leq y \leq y_t$$





# 一、直线段裁剪算法

## 1. Cohen-Sutherland 算法

对象的可见性问题：

- ①判断直线段完全可见，若是，则绘制，结束，保留
- ②判定完全不可见，若是，则不绘制，结束，舍弃
- ③若① ②都不满足，部分可见，则求线段与窗口边界的交点（分割），对子线段继续① ~ ③，直至所有直线段和由直线段分割出来的子线段都已经确定了是保留还是舍弃为止。

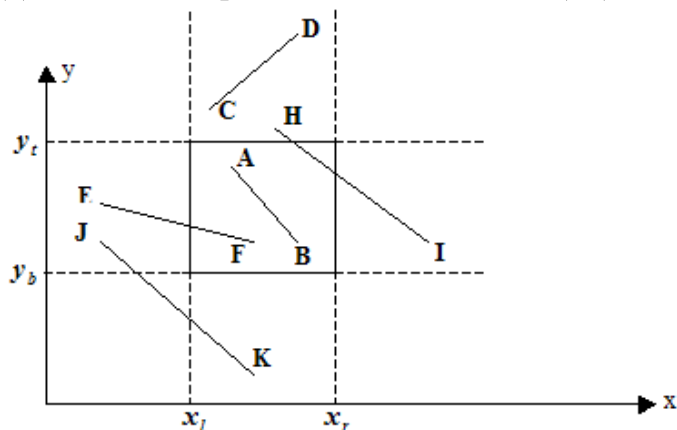


图 3.26 Cohen-sutherland 裁剪算法的例子



# Cohen-Sutherland算法

算法步骤如下：

第一步：编码

第二步：判别

第三步：求交

第四步：对剩下的线段重复以上各步



# 1.编码:

每个区域用一个4位二进制编码来标识，代码确定如下：

若区域在窗口上方：则第1位为1；

若区域在窗口下方：则第2位为1；

若区域在窗口右侧，则第3位为1；

若区域在窗口左侧，则第4位为1。

设线段的两个端点 $p_1(x_1, y_1), p_2(x_2, y_2)$ , 求出 $p_1, p_2$ 所处区域的代码 $c_1$ 和 $c_2$

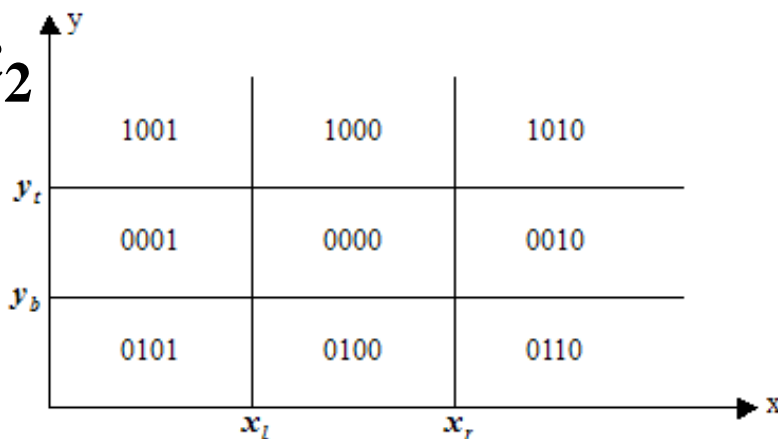


图 3.25 区域编码



## 2.判别:

(1)  $c_1=c_2=0$ , 完全可见, 保留, AB

(2)  $c_1 \& c_2 \neq 0$  完全不可见, 舍弃 CD

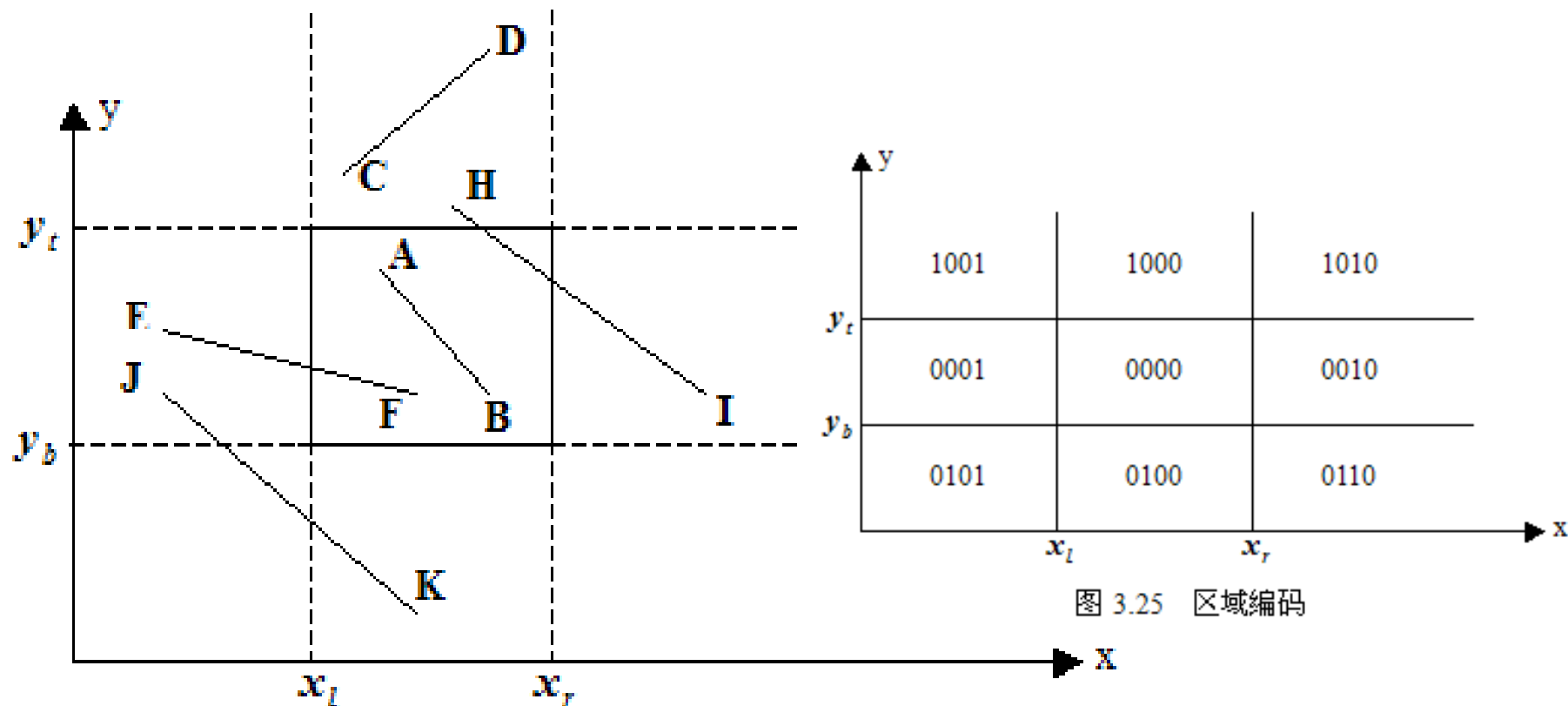


图 3.25 区域编码

图 3.26 Cohen-sutherland 裁剪算法的例子



### (3)不满足(1)(2)可分为三种情况

① 一个端点在内另一个端点在外 EF

② 两个端点都在外，且直线段在外JK

③ 两个端点都在外，但直线中部跨越窗口 HI

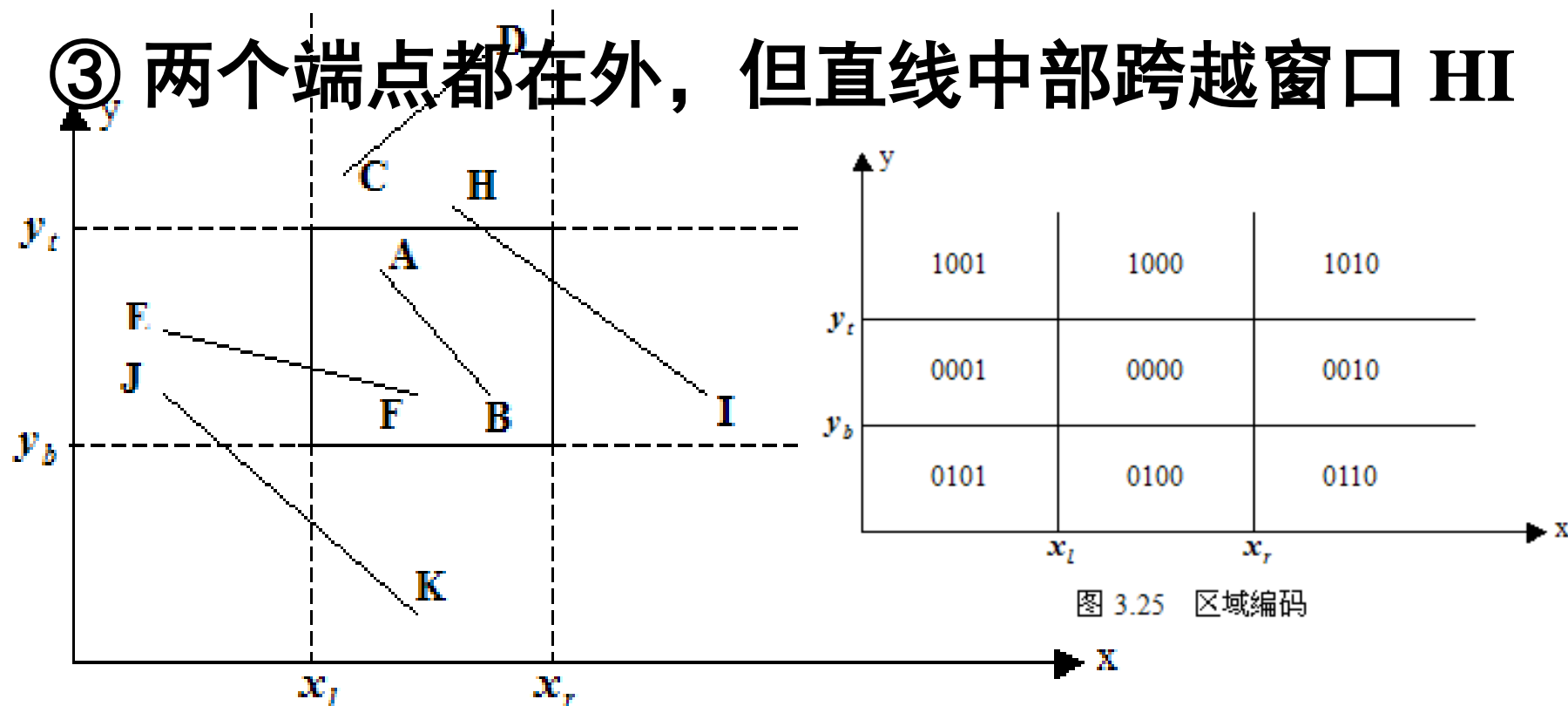


图 3.26 Cohen-sutherland 裁剪算法的例子



### 3. 求交。

#### 求线段与窗口边界及其延长线的交点

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1} \Rightarrow \begin{cases} y = y_1 + \frac{y_2 - y_1}{x_2 - x_1} (x - x_1) \\ x = x_1 + \frac{x_2 - x_1}{y_2 - y_1} (y - y_1) \end{cases}$$

将  $x = xl, x = xr$  , 代入上式, 求出左右边界交点坐标

将  $y = yb, y = yt$  , 代入上式, 求出上下边界交点坐标

### 4. 对剩下的子线段重复以上各步, 从而完成对直线段的裁剪





# Cohen-Sutherland算法

算法的程序实现如下

函数Cohen\_Sutherland用来实现算法

函数makecode用来编码，利用数值位运算

**double xl, xr, yt, yb;** (事先给出窗口的位置，  
四个数值是已知的)



# Cohen-Sutherland算法

端点 $P_0$ 编码

端点 $P_2$ 编码

while( $P_0P_2$ 不完全可见)

{

    if( $P_0P_2$ 完全不可见)return;

    求交点 $P$ ;

    if( $P_0$ 在裁剪区域外)

$P_0=P$ ;

    else

$P_2=P$ ;

}

绘制 $P_0P_2$ ;



```
int makecode(double x, double y)  
{  
    int c=0;  
    if (x<xl) c=1;  
    else if (x>xr) c=2;  
    if (y<yb) c=c+4;  
    else if (y>yt) c=c+8;  
    return c;  
}
```



```
void Cohen_Sutherland(double x0, double y0, double x2, double y2)
{
```

```
    int c, c0, c1;
```

```
    double x, y;
```

```
    c0=makecode(x0, y0); c1=makecode(x2, y2);
```

```
    while (c0!=0 || c1!=0)
```

```
    {
```

```
        if (c0&c1!=0) return;
```

```
        c=c0; if (c==0) c=c1;
```

```
        if (c&1==1) {y=y0+(y2-y0)*(xl-x0)/(x2-x0); x=xl;}
```

```
        else if (c&2==2) {y=y0+(y2-y0)*(xr-x0)/(x2-x0); x=xr;}
```

```
        else if (c&4==4) {x=x0+(x2-x0)*(yb-y0)/(y2-y0); y=yb;}
```

```
        else if (c&8==8) {x=x0+(x2-x0)*(yt-y0)/(y2-y0); y=yt;}
```

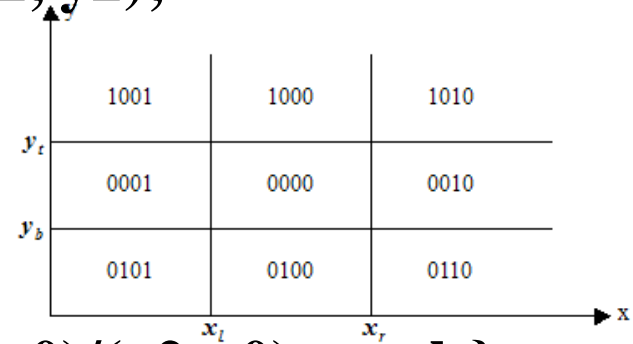
```
        if (c==c0){x0=x; y0=y; c0 = makecode(x, y);}
```

```
        else{x2=x; y2=y; c1 = makecode(x, y);}
```

```
    }
```

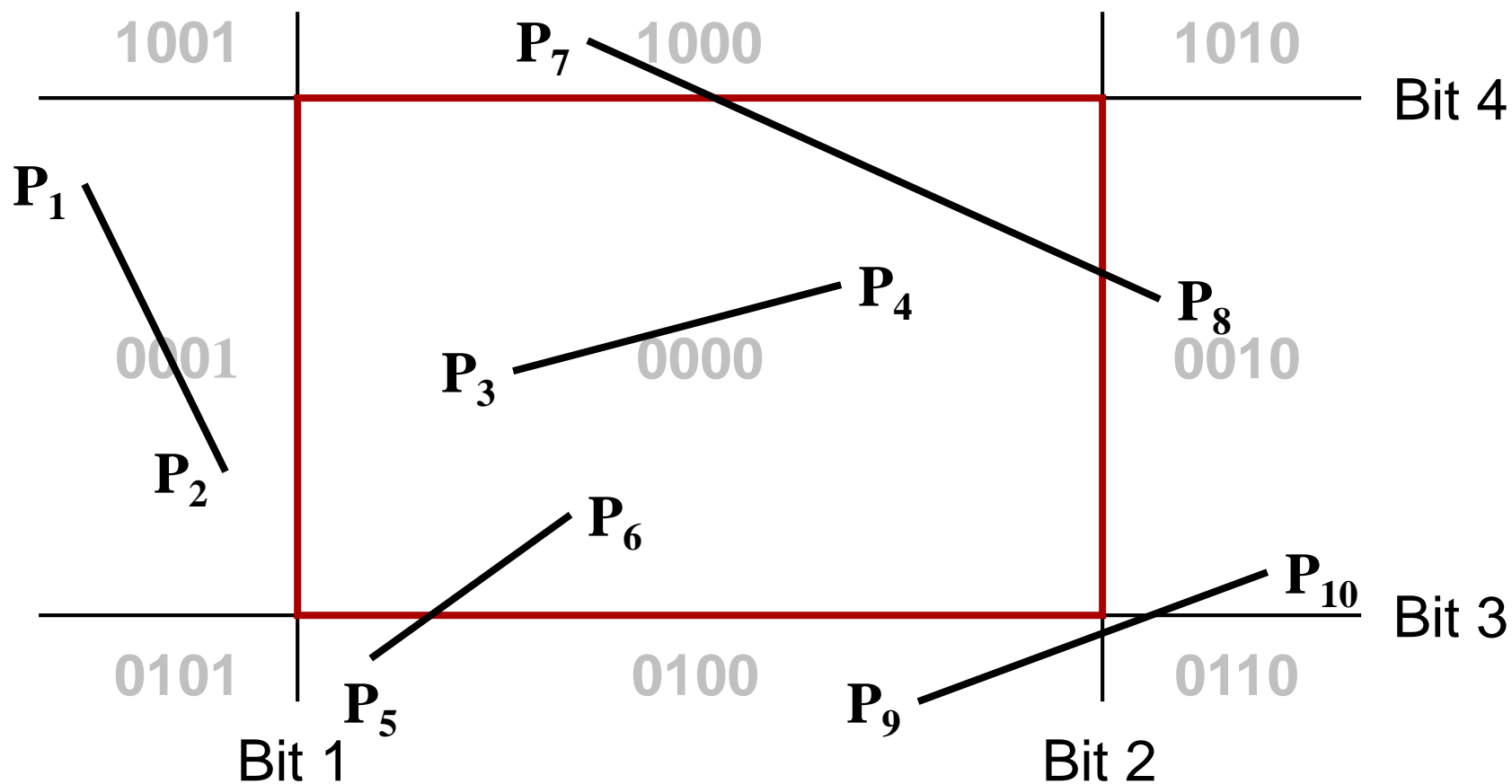
```
    showline(x0, y0, x2, y2); //显示可见线段
```

```
}
```



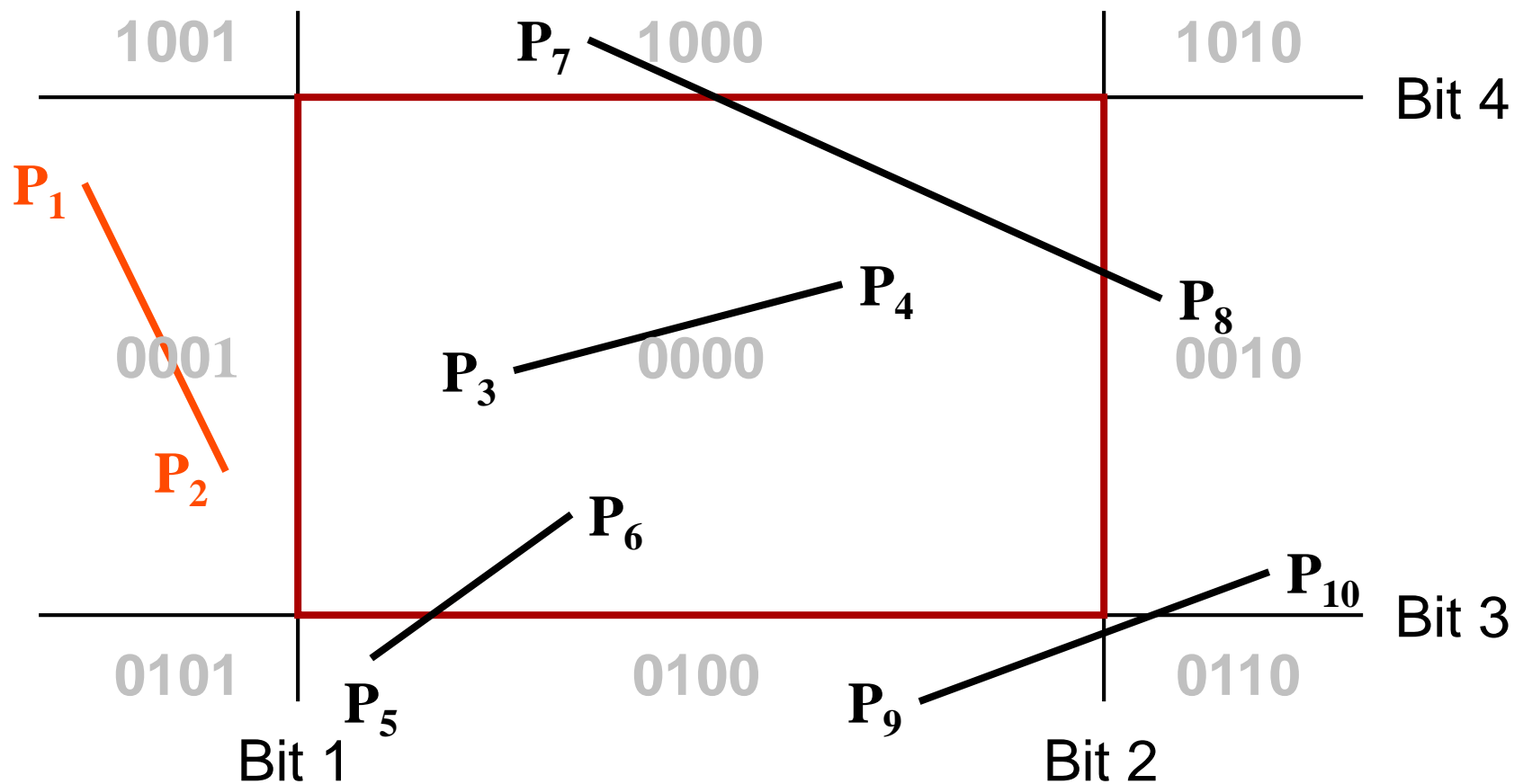


# Cohen-Sutherland 直线段裁剪



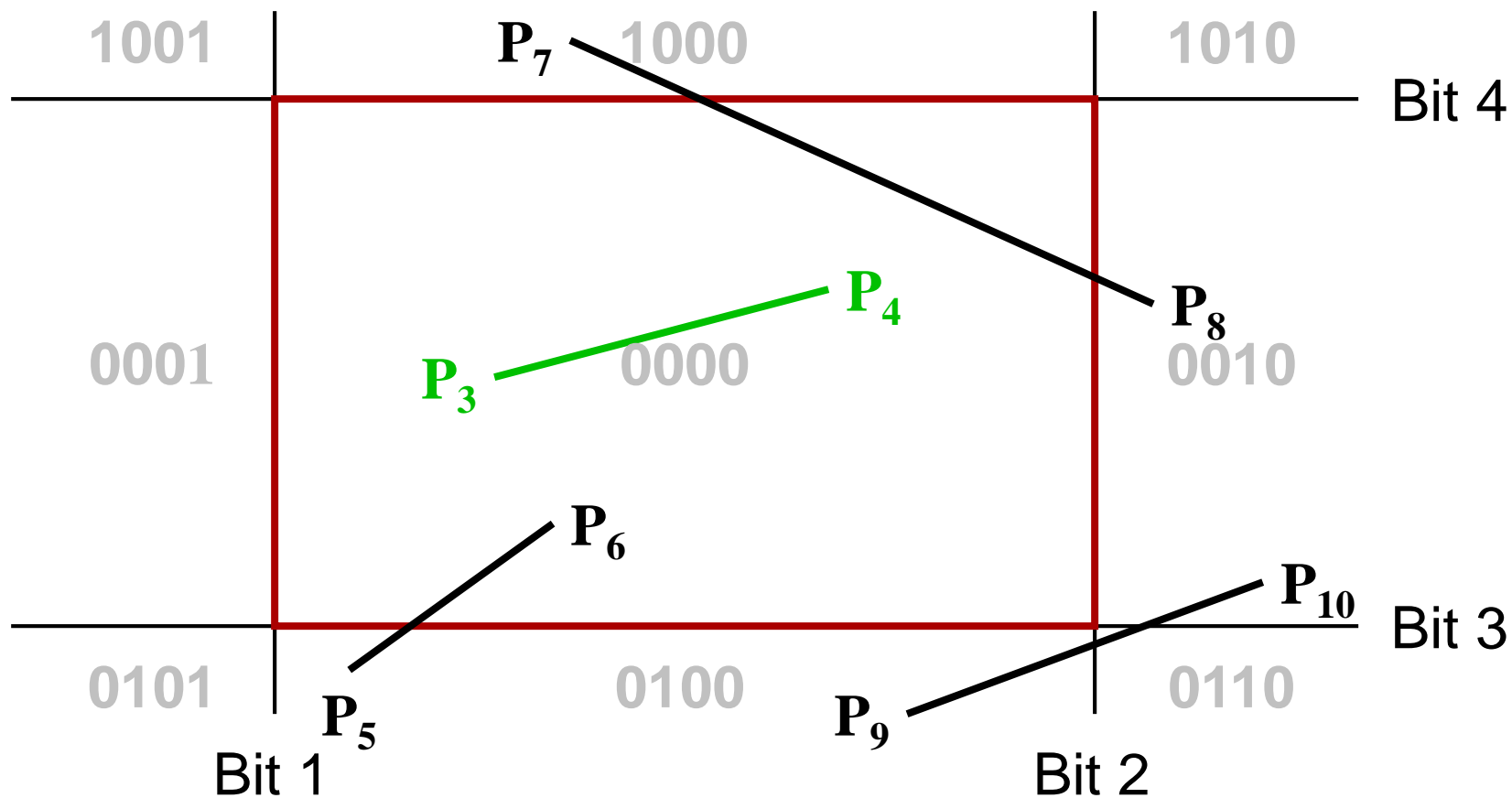


# Cohen-Sutherland 直线段裁剪



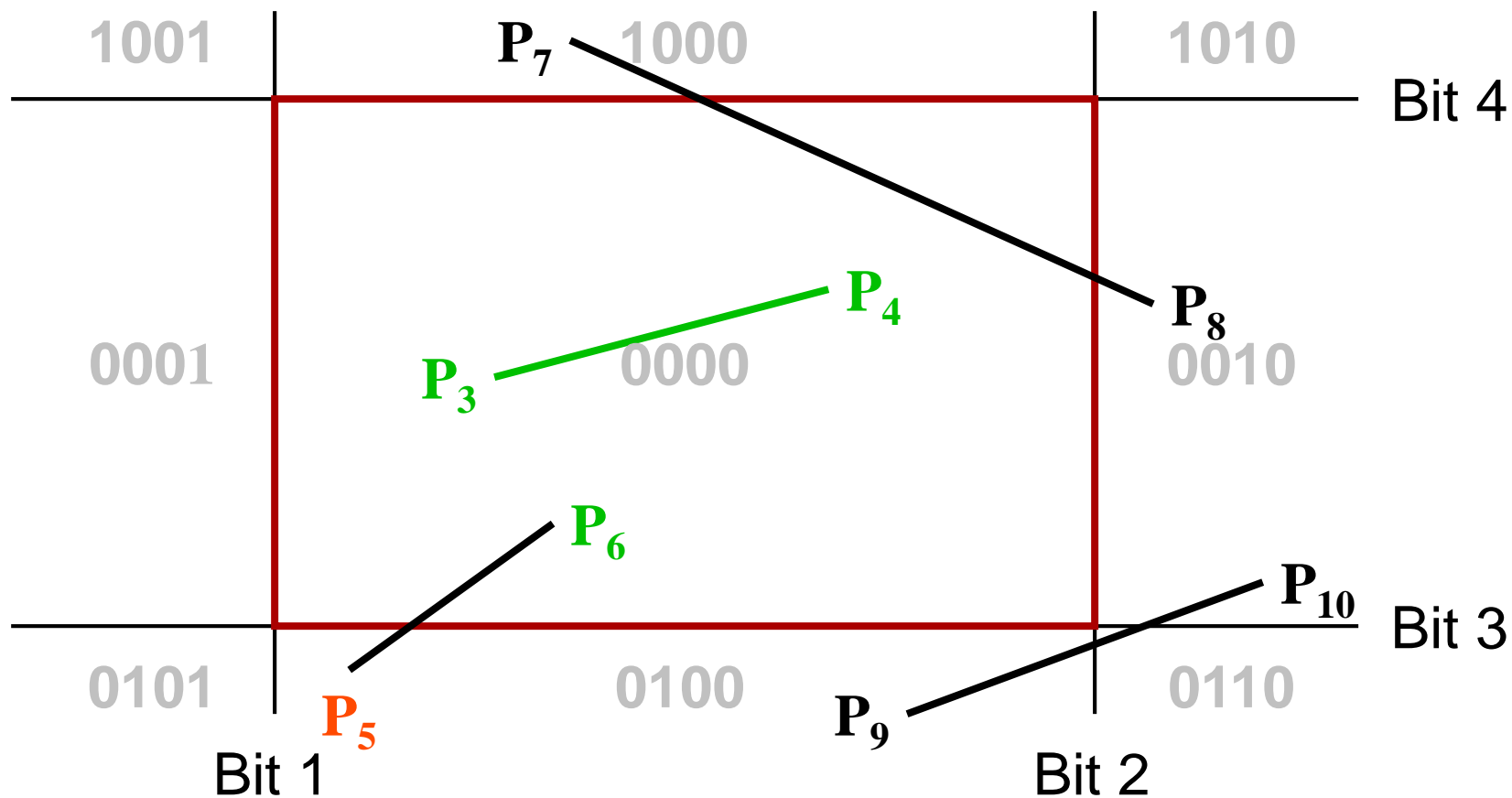


# Cohen-Sutherland 直线段裁剪





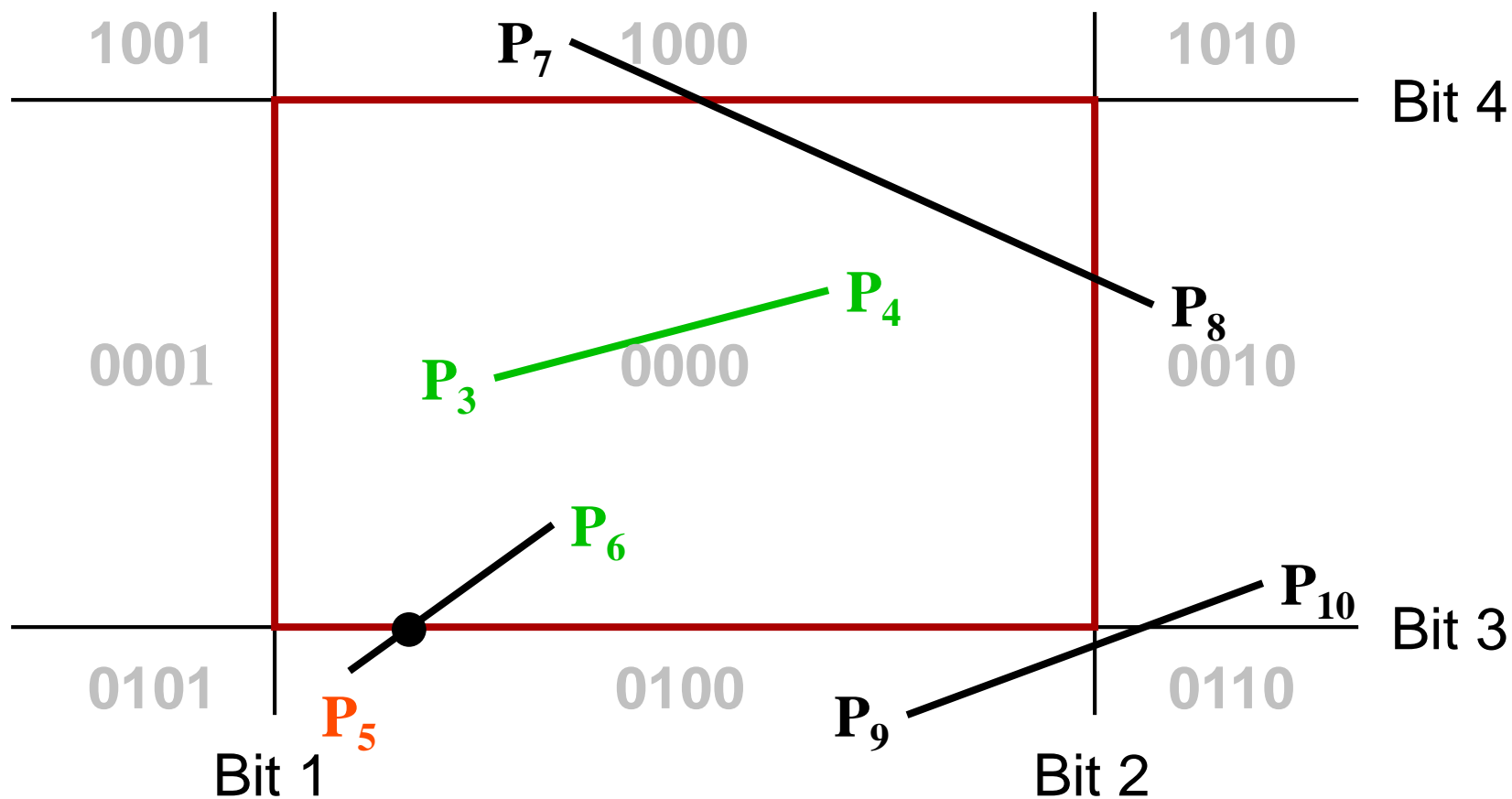
# Cohen-Sutherland 直线段裁剪





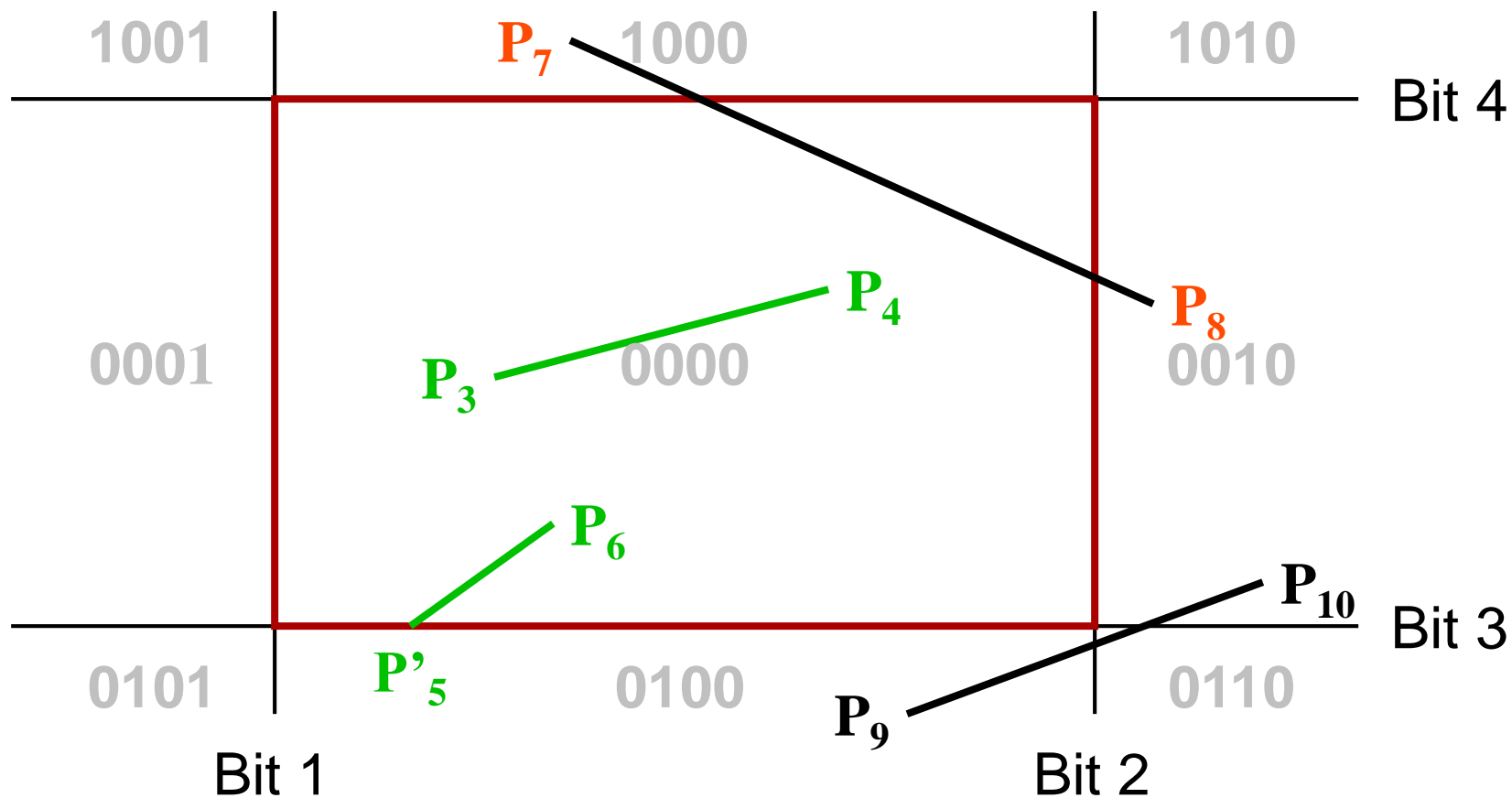


# Cohen-Sutherland 直线段裁剪



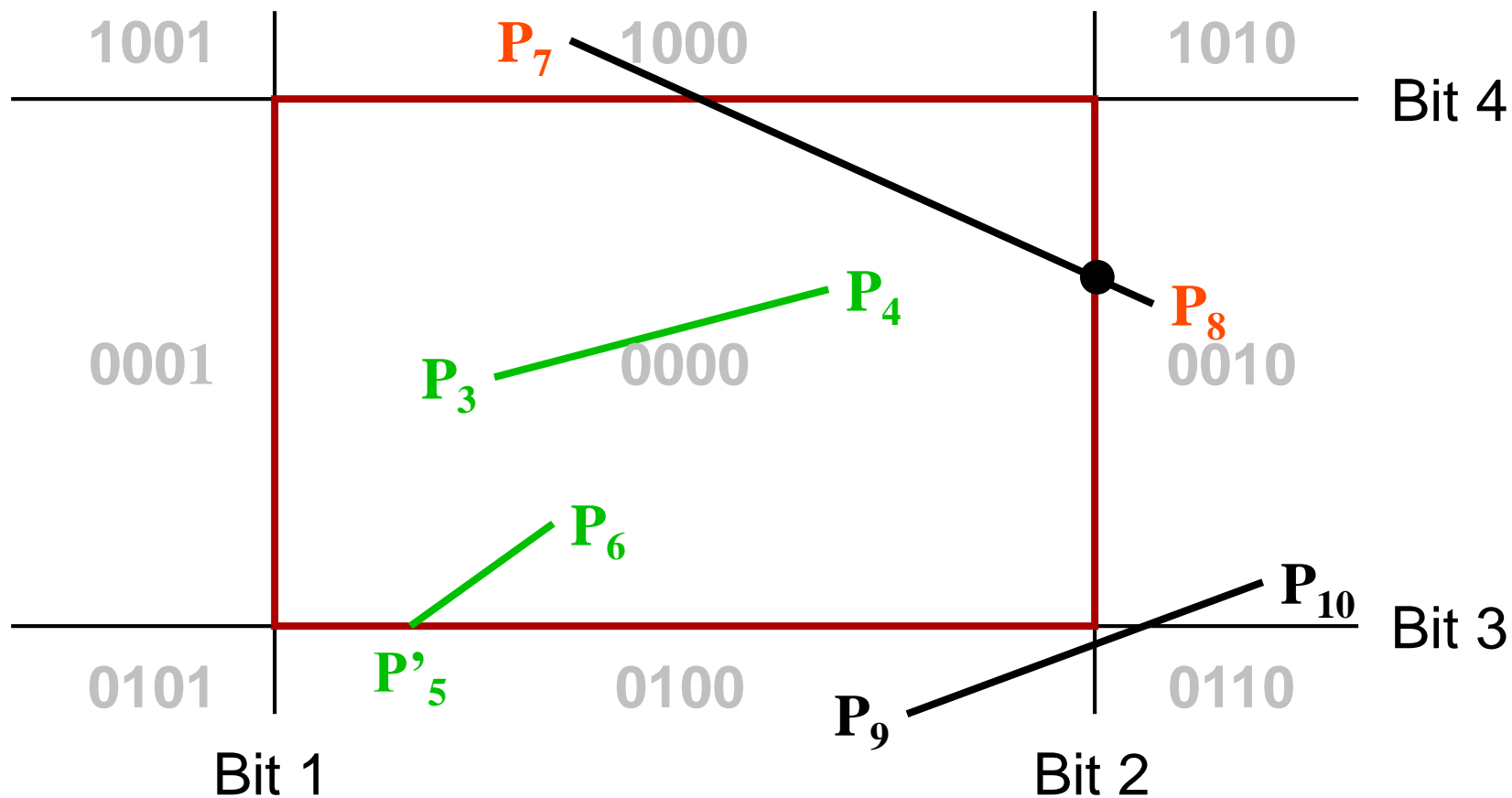


# Cohen-Sutherland 直线段裁剪



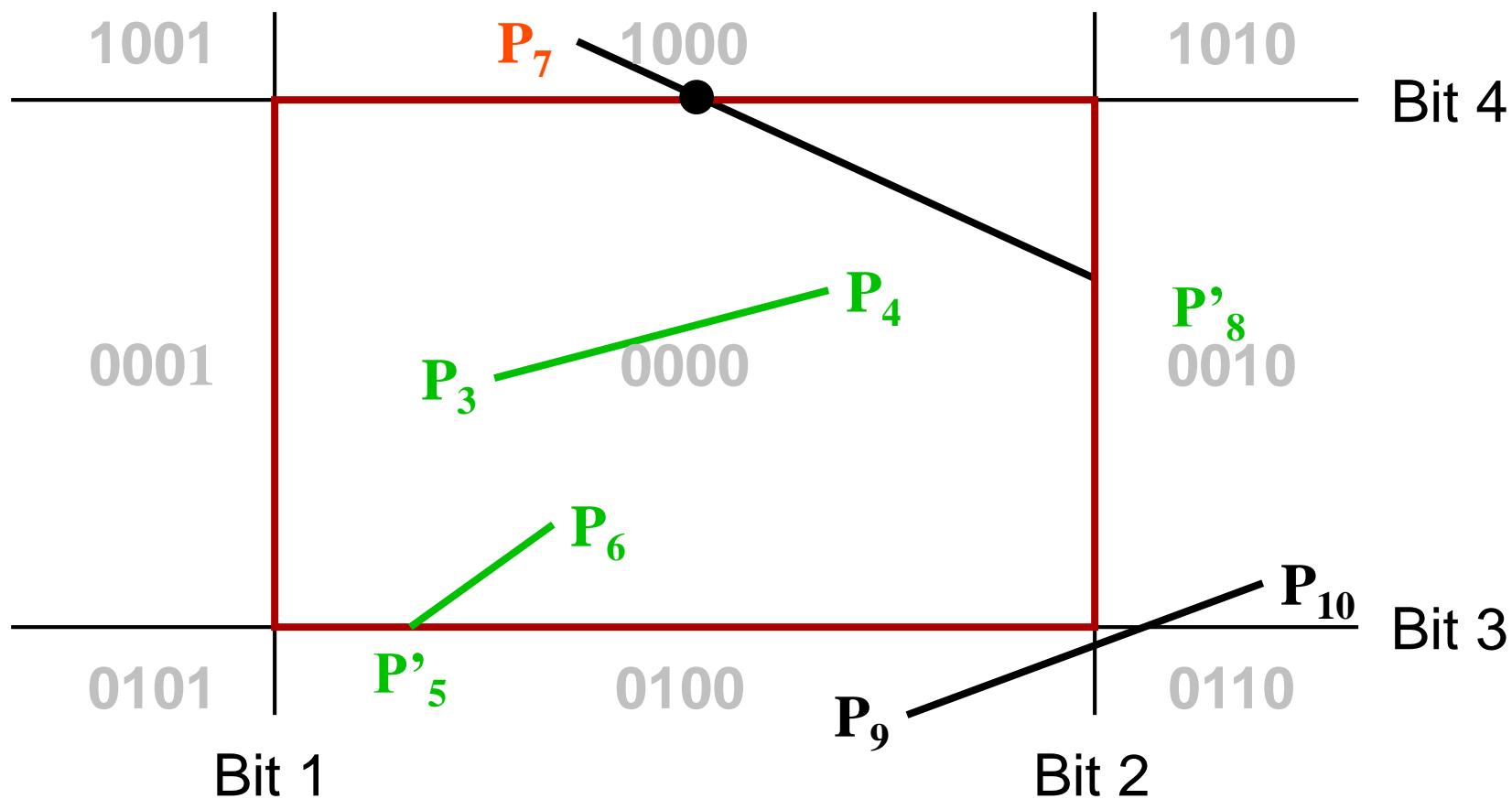


# Cohen-Sutherland 直线段裁剪



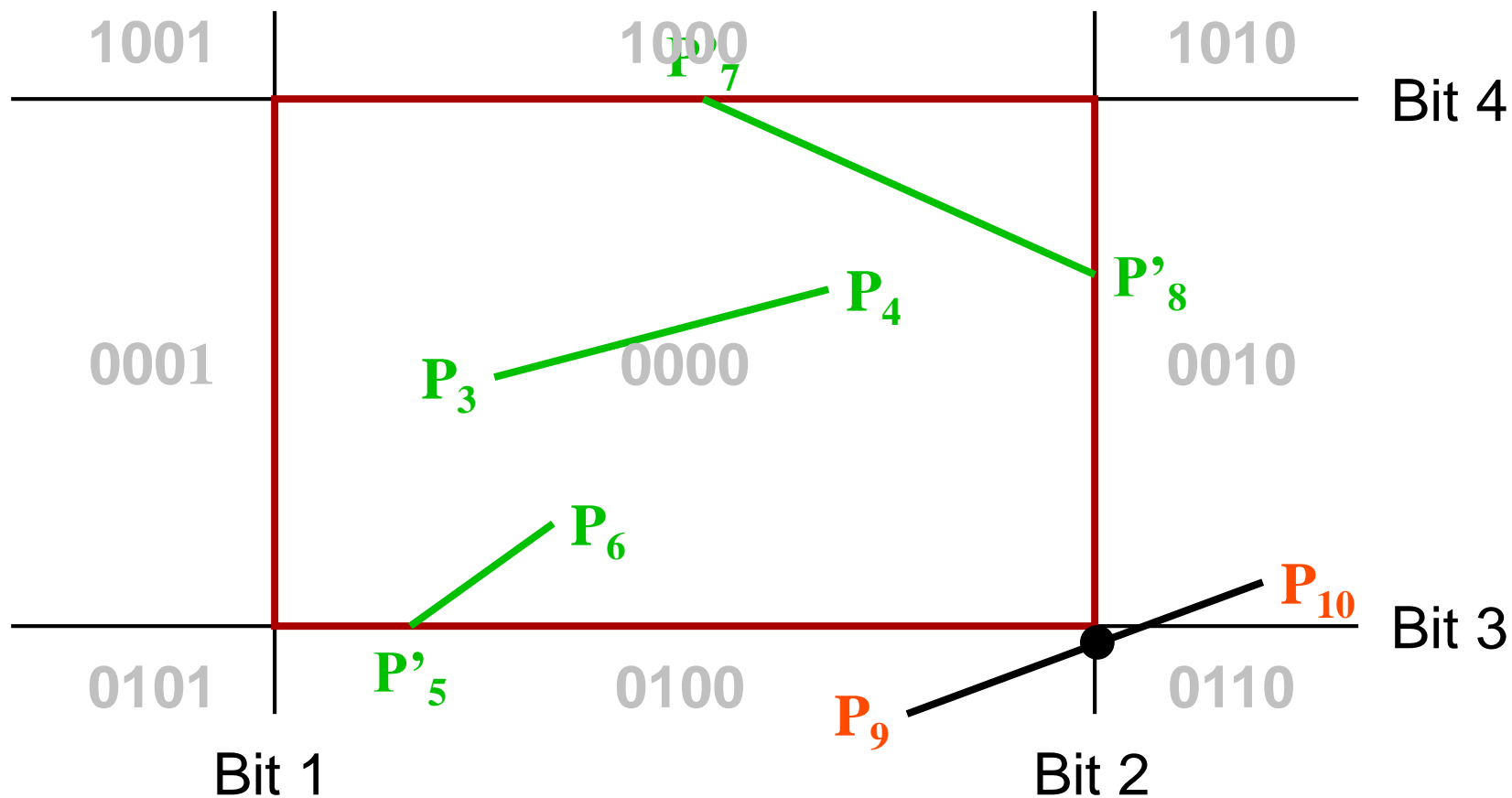


# Cohen-Sutherland 直线段裁剪



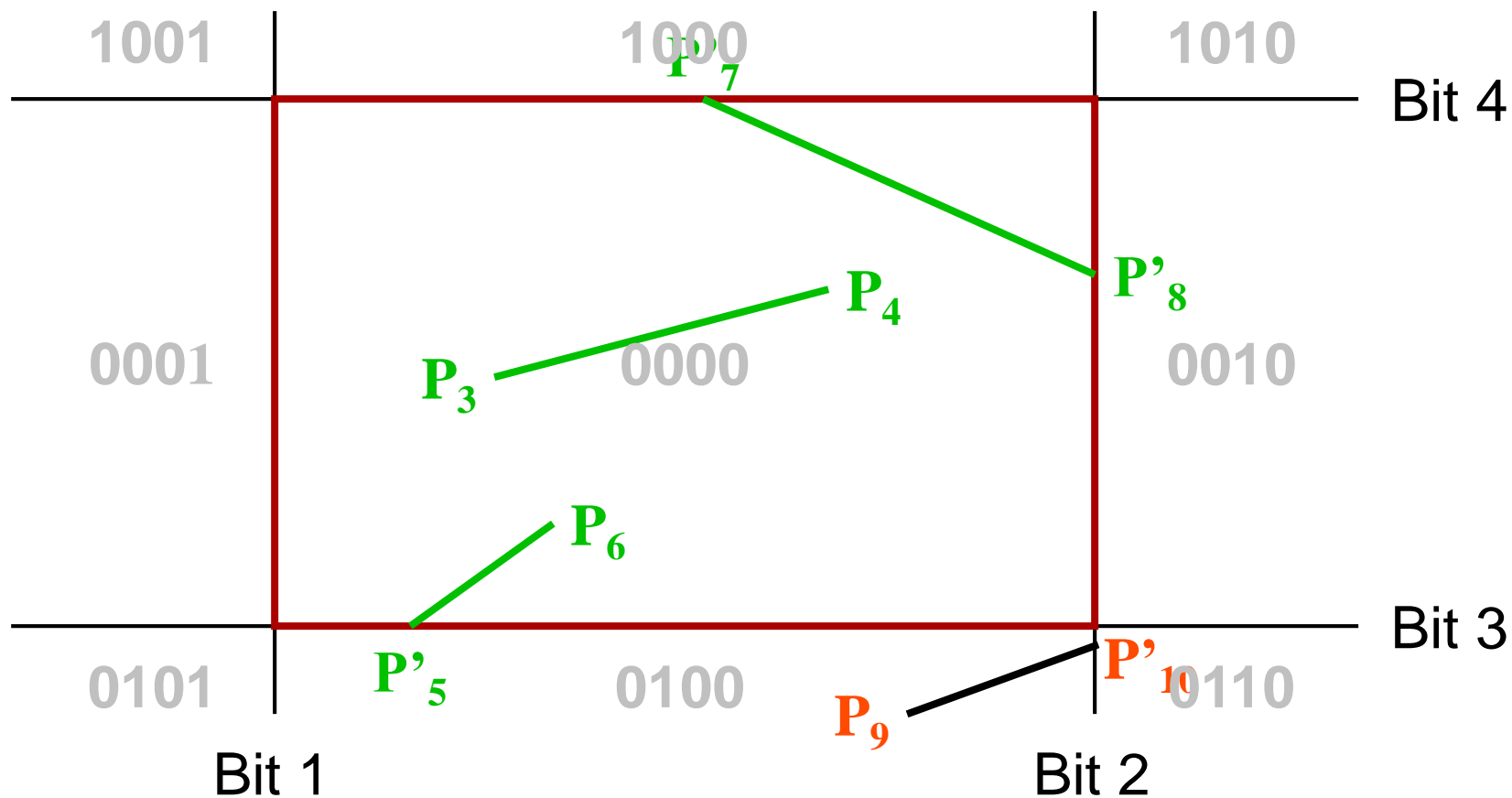


# Cohen-Sutherland 直线段裁剪



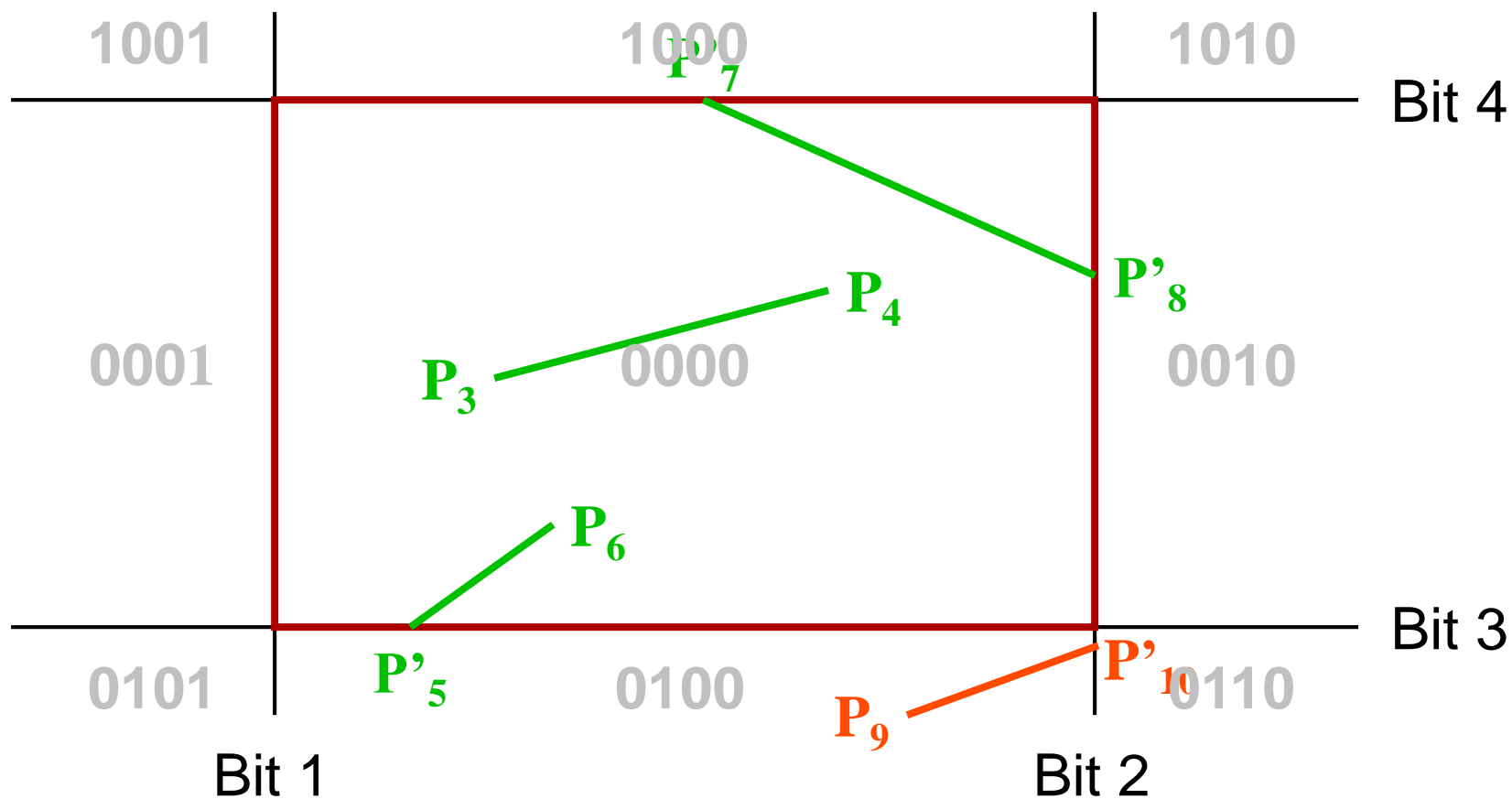


# Cohen-Sutherland 直线段裁剪



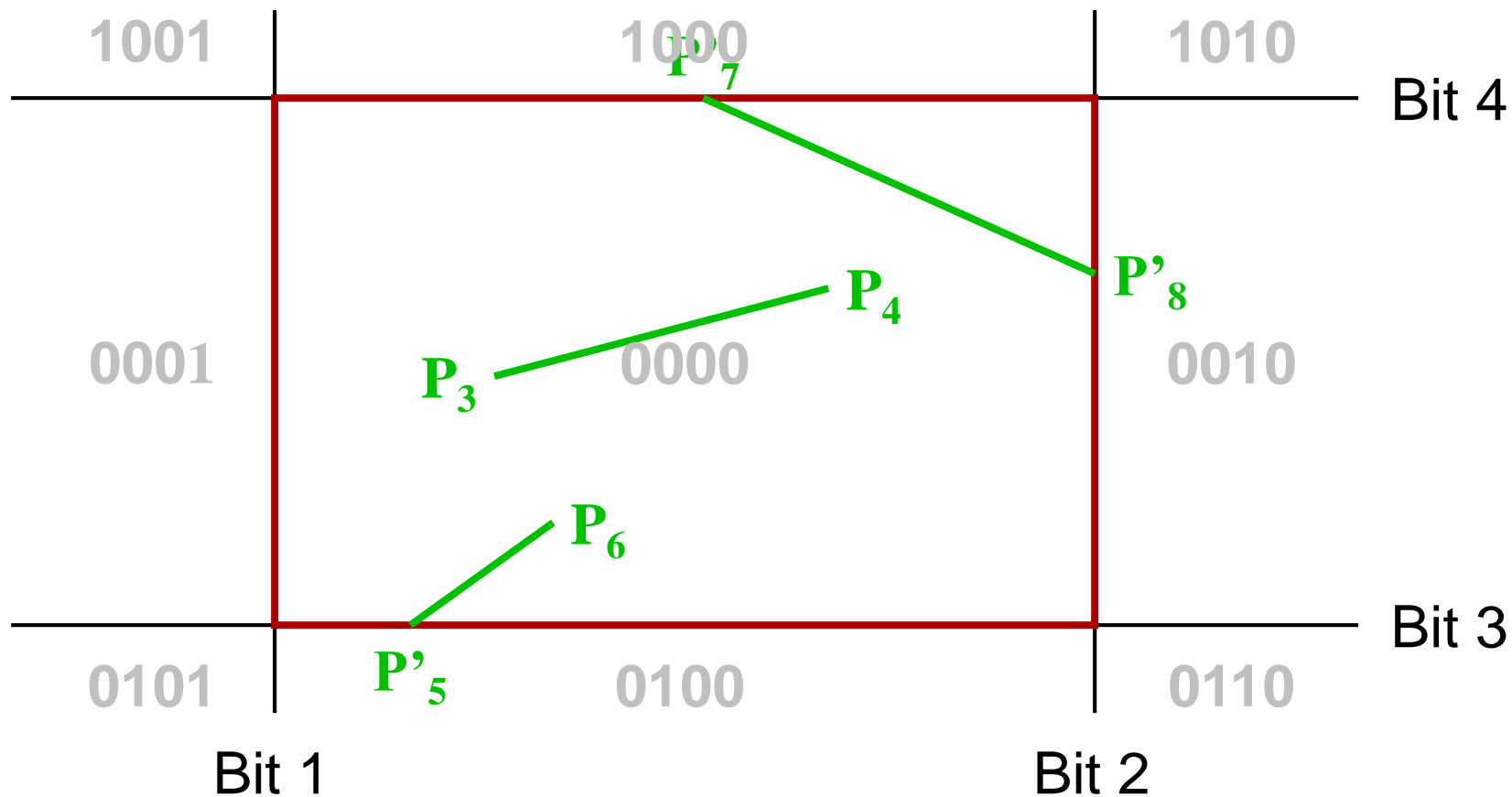


# Cohen-Sutherland 直线段裁剪





# Cohen-Sutherland 直线段裁剪







## 2、中点分割裁剪算法

- 基本思想：

与前一种Cohen-Sutherland算法一样首先对线段端点进行编码，并把线段与窗口的关系分为三种情况：

1)完全可见，绘制线段

2)完全不可见，线段不绘制，舍弃

3)求中点 $p_m$ ,分割成 $p_0p_m, p_mp_1$ ,再分别进行判断，直到

$$|p_mp_1| < \varepsilon$$



## 中点分割算法

可分成两个过程平行进行，

从 $p_0$ 点出发找出离点 $p_0$ 最近的可见点，

从 $p_1$ 点出发找出离点 $p_1$ 最近的可见点。

这两个最近可见点的连线就是原直线段的可见部分。

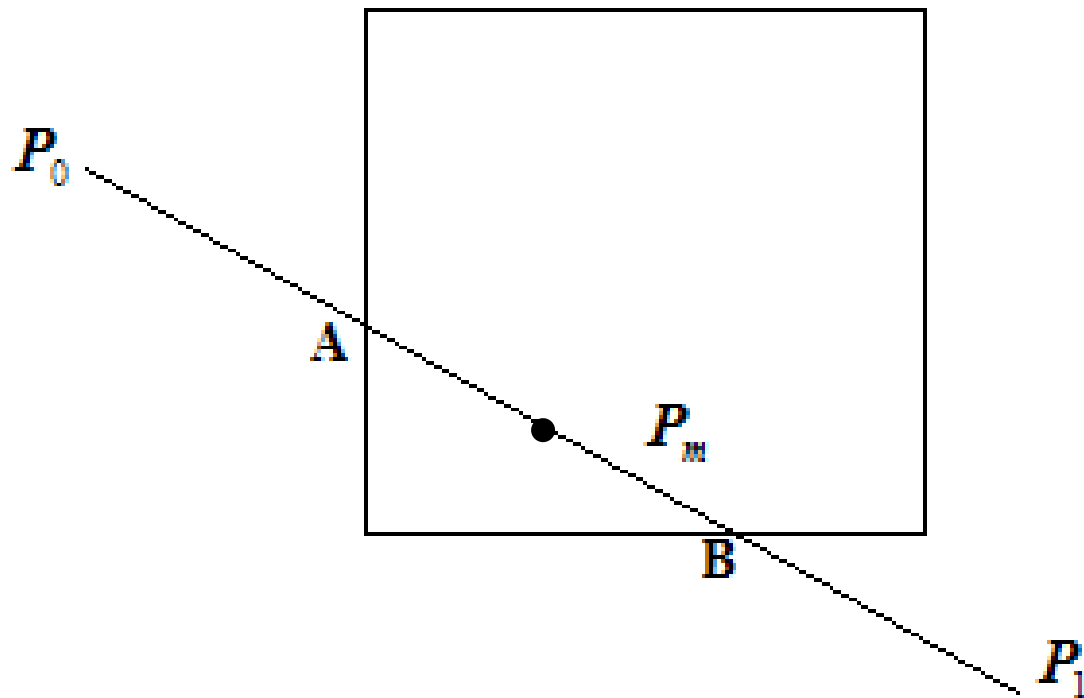


图 3.27 中点分割算法

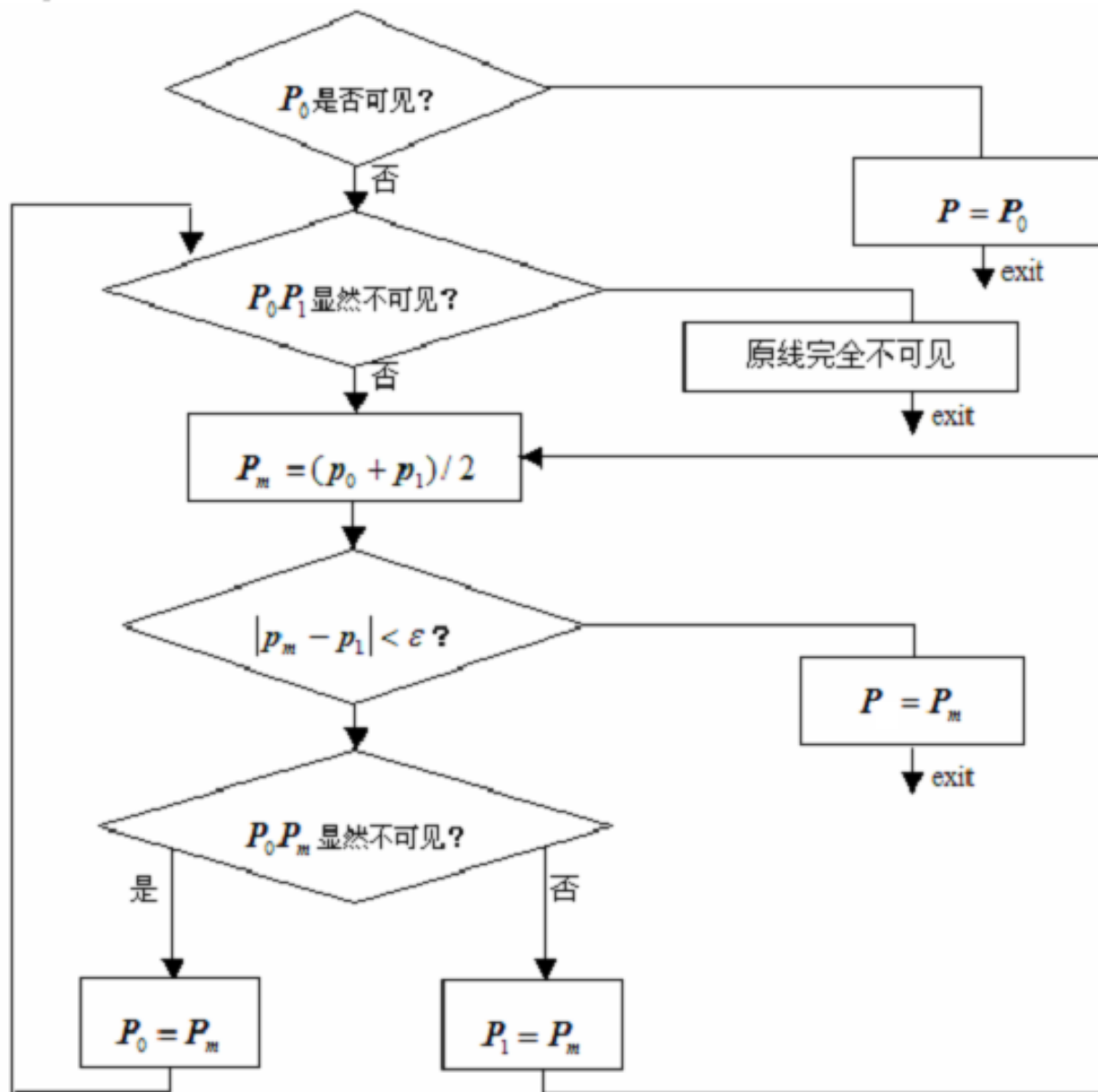
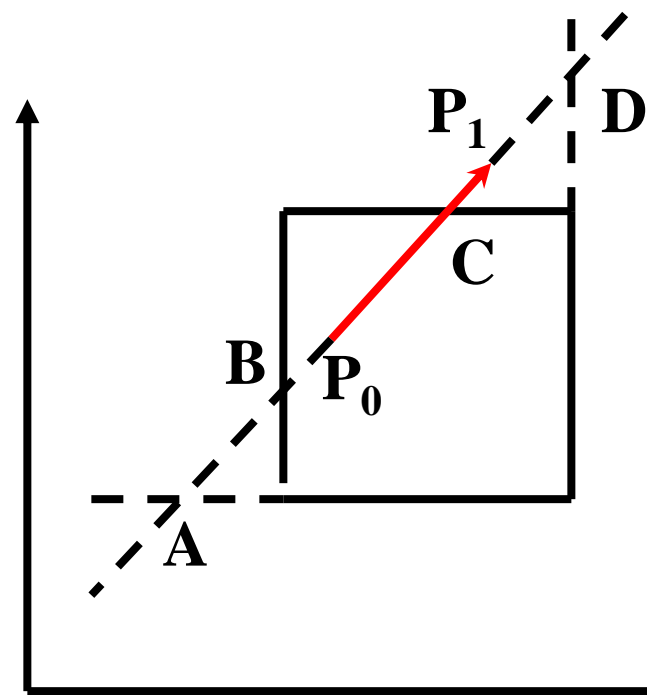


图 3.28 中点分割算法框图

从 $P_0$ 出发找最近可见点的方法



# 梁友栋—Barsky算法



设要裁剪的直线段为  $P_0P_1$  和窗口边界交于A、B、C和D四个点。

基本思想：从A、B(始边交点)和 $P_0$ 三点中找出最靠近 $P_1$ 的点，该点是 $P_0$ 。从C、D(终边交点)和 $P_1$ 点中找出最靠近 $P_0$ 的点，该点是点C。那么 $P_0C$ 就是线段上的可见部分。



在具体计算时，可以把 $P_0P_1$ 写成参数方程：

$$x = x_0 + \Delta x \bullet t \quad \Delta x = x_1 - x_0$$

$$y = y_0 + \Delta y \bullet t \quad \Delta y = y_1 - y_0$$

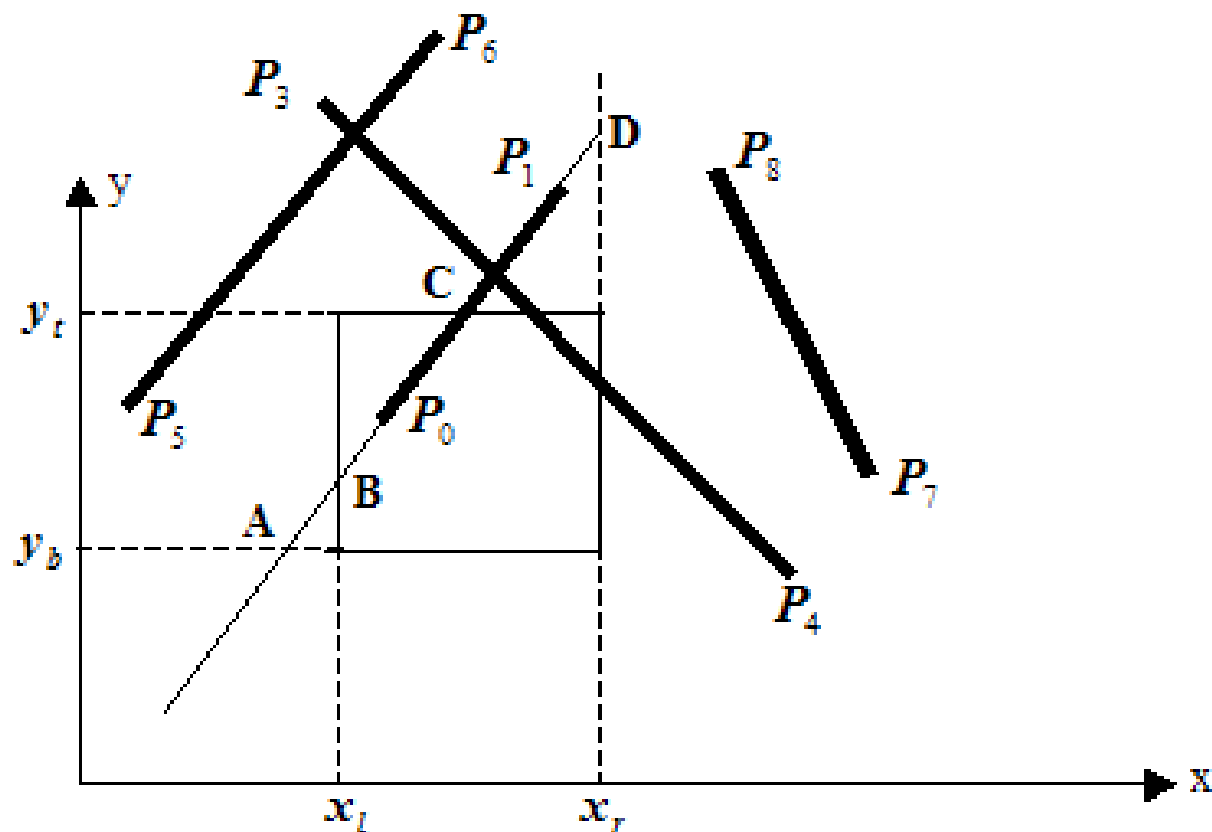


图 3.29 梁友栋-Barsky 裁剪算法



把窗口边界的四条边分成两类，一类称为**始边**，另一类称为**终边**。

当  $\Delta x \geq 0 (\Delta y \geq 0)$  时，  
称  $x = x_l (y = y_b)$  为始边， $x = x_r (y = y_t)$  为终边。  
当  $\Delta x < 0 (\Delta y < 0)$  时，  
称  $x = x_r (y = y_t)$  为始边， $x = x_l (y = y_b)$  为终边。

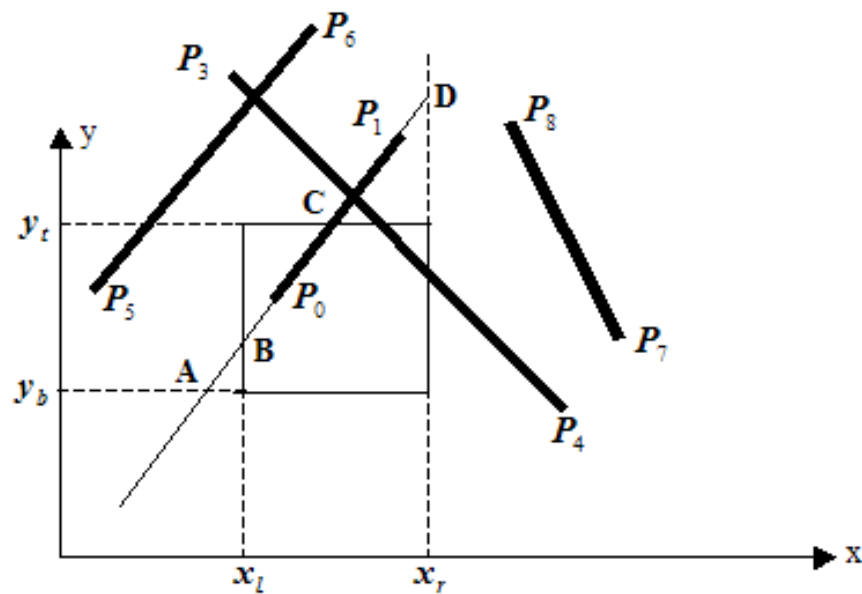
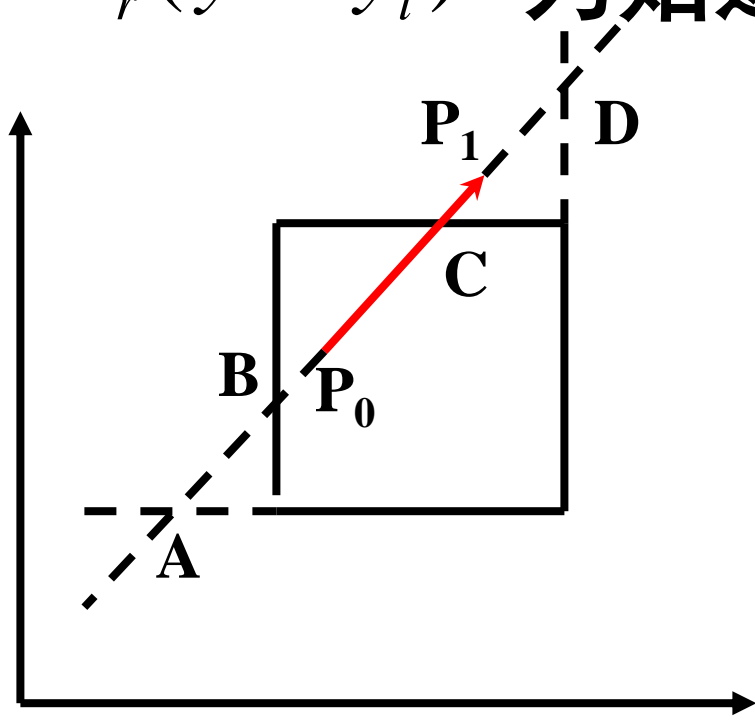


图 3.29 梁友栋-Barsky 裁剪算法

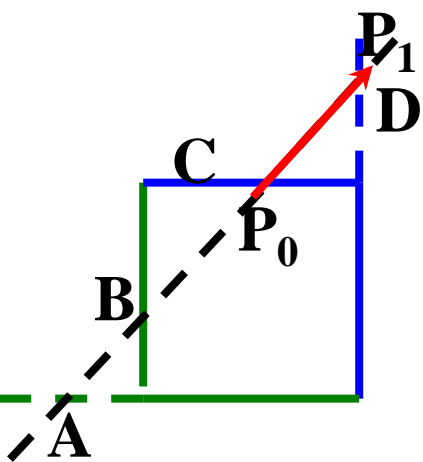


求出 $P_0P_1$ 和两条始边的交点的参数 $t'_0$ 和 $t''_0$ ，令：

$$t_0 = \max(t'_0, t''_0, 0)$$

求出 $P_0P_1$ 和两条终边的交点的参数 $t'_1$ 和 $t''_1$ ，令：

$$t_1 = \min(t'_1, t''_1, 1)$$

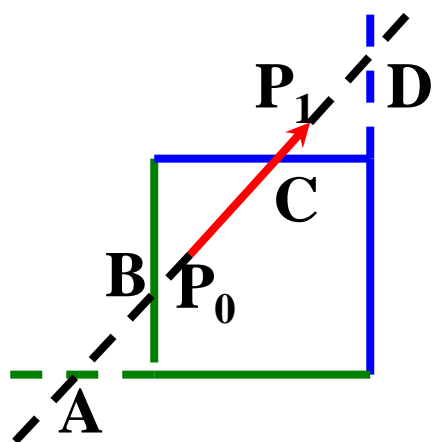


$$t'_0 < 0$$

$$t''_0 < 0$$

$$t'_1 < 1$$

$$t''_1 < 1$$

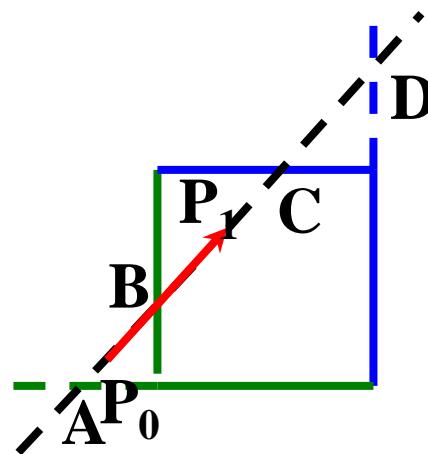


$$t'_0 < 0$$

$$t''_0 < 0$$

$$t'_1 < 1$$

$$t''_1 > 1$$

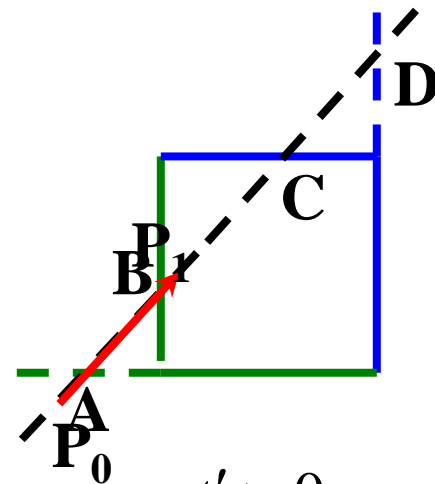


$$t'_0 < 0$$

$$t''_0 > 0$$

$$t'_1 > 1$$

$$t''_1 > 1$$



$$t'_0 > 0$$

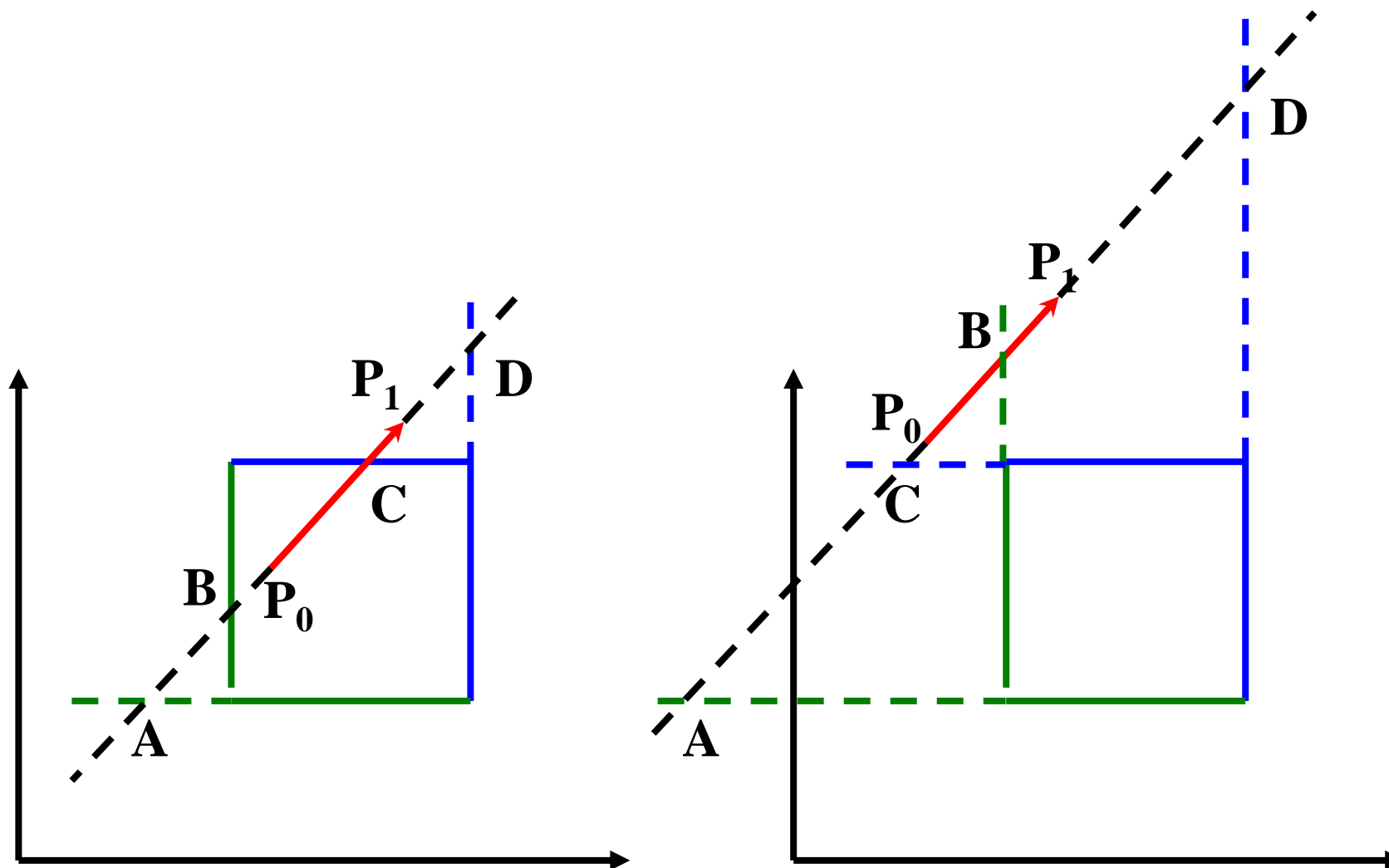
$$t''_0 > 0$$

$$t'_1 > 1$$

$$t''_1 > 1$$



当 $t_1 > t_0$ 时，参数方程中参数的线段就是的可见部分。当 $t_1 < t_0$ 时，整个直线段为不可见







求出 $P_0P_1$ 和两条始边的交点的参数 $t'_0$ 和 $t''_0$ ，令：

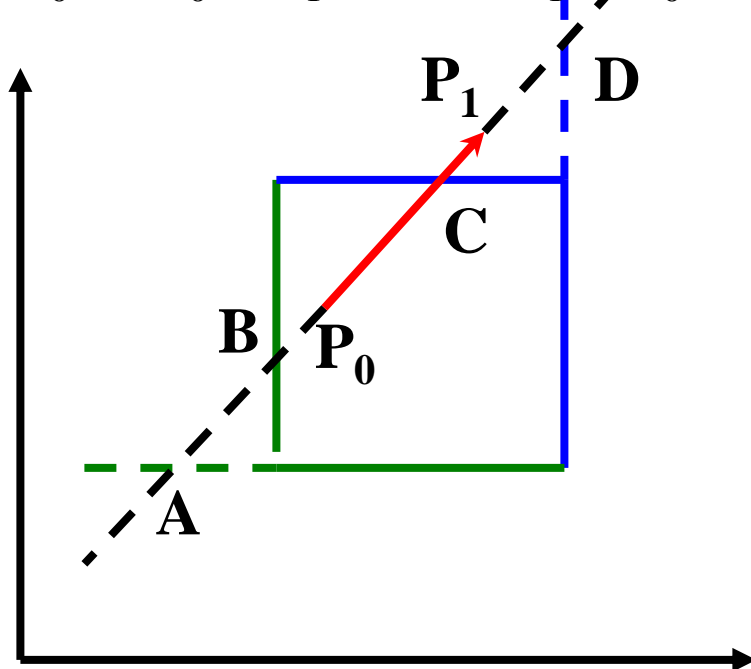
$$t_0 = \max(t'_0, t''_0, 0)$$

求出 $P_0P_1$ 和两条终边的交点的参数 $t'_1$ 和 $t''_1$ ，令：

$$t_1 = \min(t'_1, t''_1, 1)$$

交点的参数为：

$$t_l = \frac{x_l - x_0}{x_1 - x_0} = \frac{x_0 - x_l}{x_0 - x_1} \quad t_r = \frac{x_r - x_0}{x_1 - x_0} \quad t_b = \frac{y_b - y_0}{y_1 - y_0} = \frac{y_0 - y_b}{y_0 - y_1} \quad t_t = \frac{y_t - y_0}{y_1 - y_0}$$



$t_l$ 是 $P_0P_1$ 与左边界的交点参数

$t_r$ 是 $P_0P_1$ 与右边界的交点参数

$t_b$ 是 $P_0P_1$ 与下边界的交点参数

$t_t$ 是 $P_0P_1$ 与上边界的交点参数



令：

$$t_i = \frac{D_i}{Q_i}, \quad i = l, r, b, t$$

$$\text{则 } Q_l = -\Delta x = x_0 - x_1, \quad D_l = x_0 - x_l$$

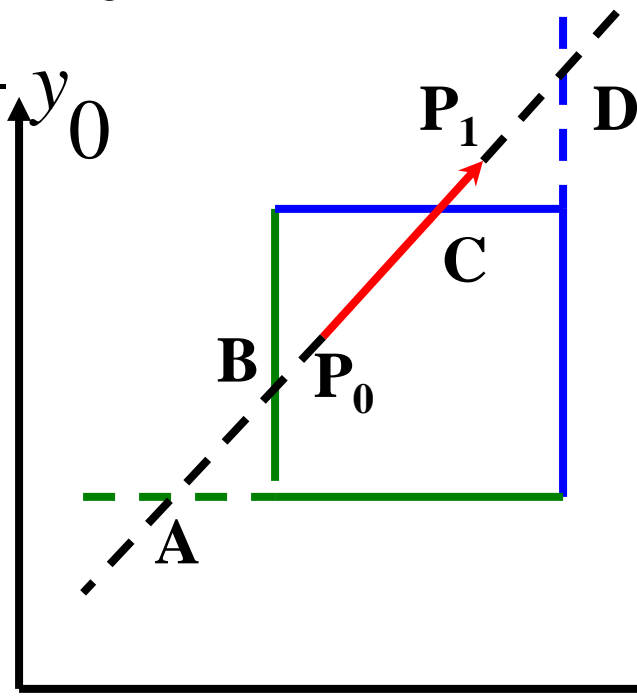
$$Q_r = \Delta x = x_1 - x_0, \quad D_r = x_r - x_0$$

$$Q_b = -\Delta y = y_0 - y_1, \quad D_b = y_0 - y_b$$

$$Q_t = \Delta y = y_1 - y_0, \quad D_t = y_t - y_0$$

$$t_l = \frac{x_0 - x_l}{x_0 - x_1} \quad t_r = \frac{x_r - x_0}{x_1 - x_0}$$

$$t_b = \frac{y_0 - y_b}{y_0 - y_1} \quad t_t = \frac{y_t - y_0}{y_1 - y_0}$$





	始边	终边			始边 参数
$\Delta x > 0$	$l$	$r$	$Q_l < 0$	$Q_r > 0$	$t_l$
$\Delta x < 0$	$r$	$l$	$Q_l > 0$	$Q_r < 0$	$t_r$
$\Delta y > 0$	$b$	$t$	$Q_b < 0$	$Q_t > 0$	$t_b$
$\Delta y < 0$	$t$	$b$	$Q_b > 0$	$Q_t < 0$	$t_t$

当 $\Delta x > 0$ ( $\Delta y > 0$ )时, 称 $x_l(y_b)$ 为始边,  $x_r(y_t)$ 为终边。

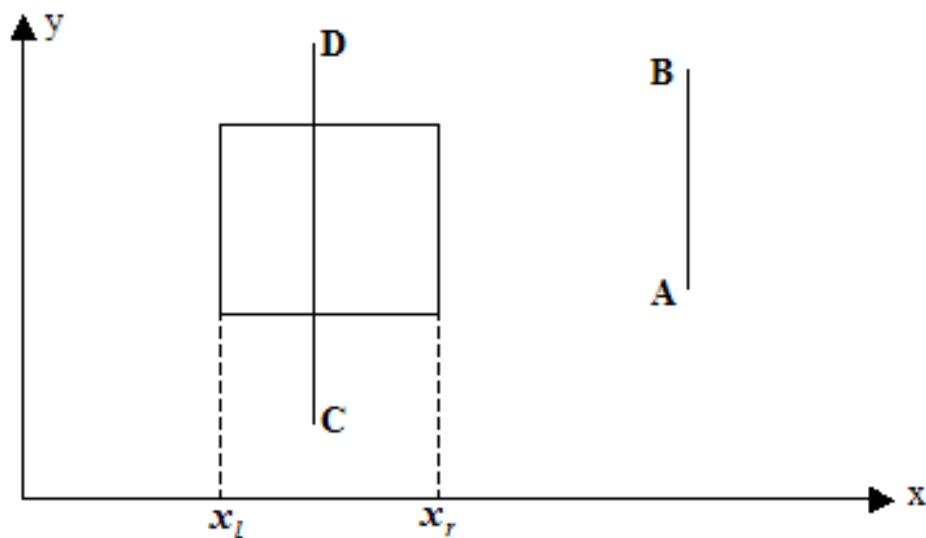
当 $\Delta x < 0$ ( $\Delta y < 0$ )时, 称 $x_r(y_t)$ 为始边,  $x_l(y_b)$ 为终边

当 $Q_i < 0$ 时,  $t_i$ 是 $P_0P_1$ 和始边的交点参数。

当 $Q_i > 0$ 时,  $t_i$ 是 $P_0P_1$ 和终边的交点参数。



当  $Q_i < 0$  时,  $t_i$  是  $P_0P_1$  和始边的交点的参数。  
 当  $Q_i > 0$  时,  $t_i$  是  $P_0P_1$  和终边的交点的参数。  
 当  $\exists i, Q_i = 0, D_i < 0$  时, 则  $p_0p_1$  是完全不可见的  
 (如图3.30中的直线段AB, 使  $Q_r = 0, D_r < 0$ )。  
 当  $Q_i = 0, D_i \geq 0$  时, 如CD,  $Q_l = 0, D_l > 0; Q_r = 0, D_r > 0$



$$D_l = x_0 - x_l$$

$$D_r = x_r - x_0$$

$$D_b = y_0 - y_b$$

$$D_t = y_t - y_0$$

图 3.30  $Q_i = 0$  的情况



**bool cansee(double q, double d, double & t0, double & t1)**// 函数cansee用于判断直线段是否可见。

```
{  
    double r;  
    if (q<0){//计算与始边的交点参数  
        r=d/q;  
        if (r>t1) {return false;}  
        else if (r>t0) t0=r;//  $t_0 = \max(t'_0, t''_0, 0)$   
    }else if (q>0){//计算与终边的交点参数  
        r=d/q;  
        if (r<t0) {return false;}  
        else if (r<t1) t1=r;//  $t_1 = \min(t'_1, t''_1, 1)$   
    }else if (d<0) {return false;}  
    return true;  
}
```



# 函数L\_Barsky实现算法

```
void L_Barsky(double x0, double y0, double x1,  
double y1)// double xl, xr, yt, yb; (窗口位置)  
{   double t0, t1, deltax, deltay;  
    t0=0.0; t1=1.0;  
    deltax=x1-x0;  
    if (!cansee(-deltax, x0-xl, t0, t1)) return;  
    if (!cansee(deltax, xr-x0, t0, t1)) return;  
    deltay=y1-y0;  
    if (!cansee(-deltay, y0-yb, t0, t1)) return;  
    if (!cansee(deltay, yt-y0, t0, t1)) return;  
    x1=x0+t1*deltax;y1=y0+t1*deltay;  
    x0=x0+t0*deltax;y0=y0+t0*deltay;  
    showline(x0, y0, x1, y1); //显示可见线段  
}
```



## 二、其它图形的裁剪

- 1、字符的裁剪
- 2、多边形的裁剪
- 3、圆弧的裁剪
- 4、裁剪区域为凸多边形区域的直线段裁剪



# 字符裁剪

**串精度**: 将包围字串的外接矩形对窗口作裁剪

**字符精度**: 将包围字的外接矩形对窗口作裁剪

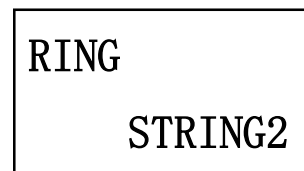
**笔画\象素精度**: 将笔划分解成直线段对窗口作裁剪



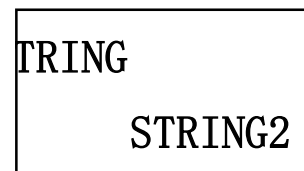
待裁剪字符串



串精度裁剪



字符精度裁剪



象素精度裁剪



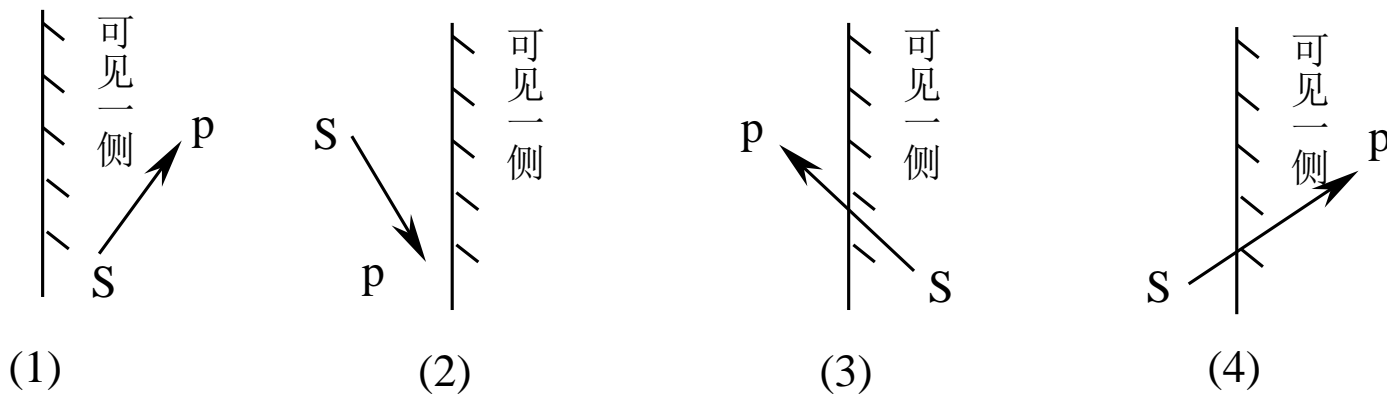


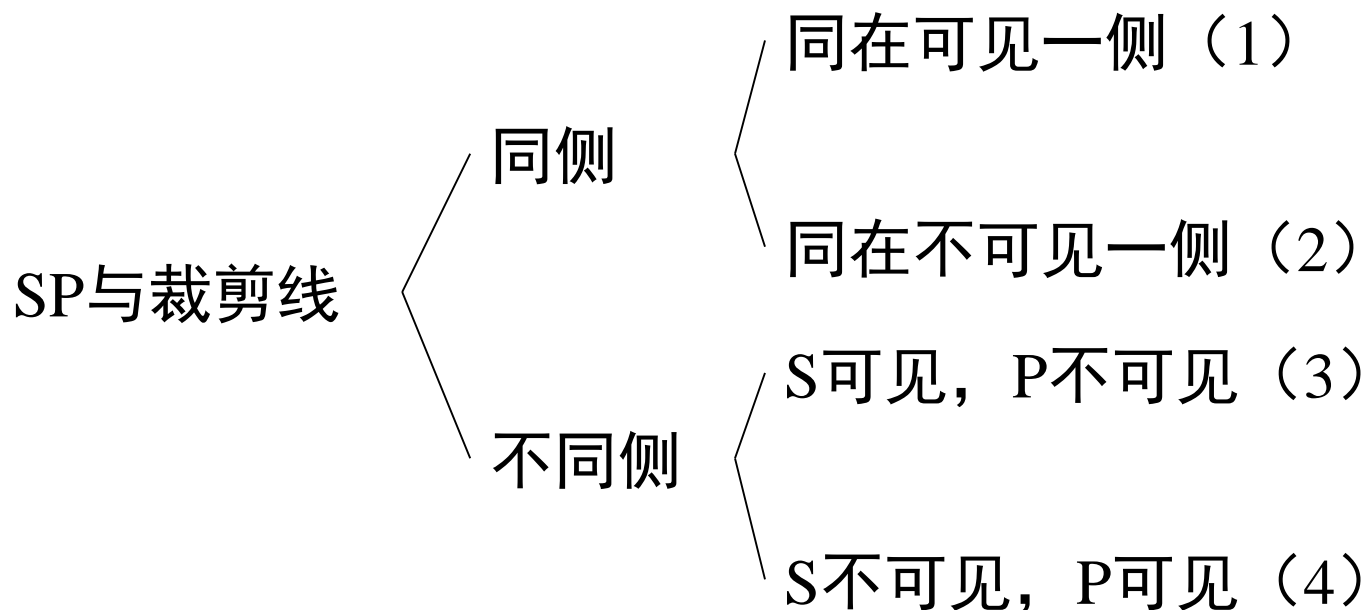
# 多边形裁剪

基本思想：一次用窗口的一条边裁剪多边形。

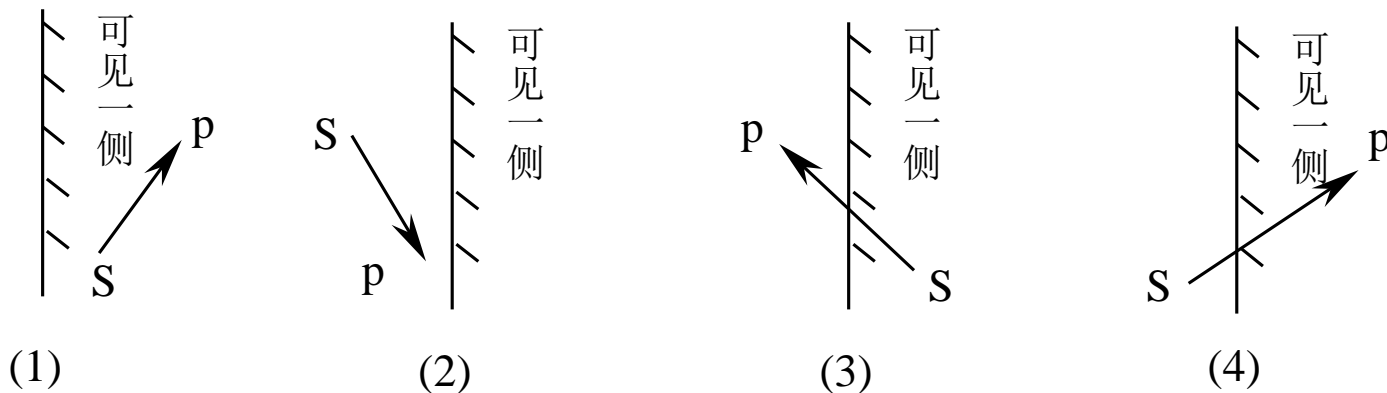
窗口的一条边以及延长线构成的裁剪线把平面分成两个部分：可见一侧；不可见一侧

多边形的各条边的两端点S、P。它们与裁剪线的位置关系只有四种



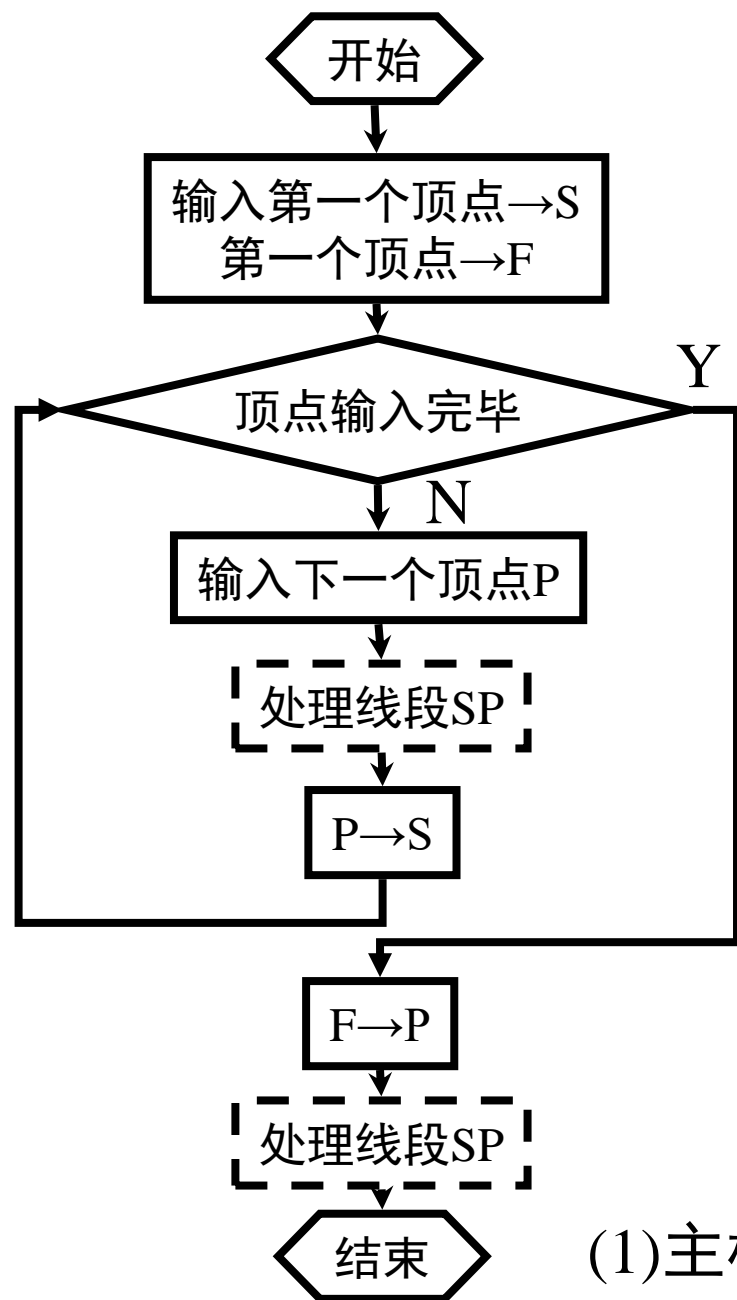


对于情况 (1) 仅输出顶点P; 情况 (2) 输出0个顶点;  
情况 (3) 输出线段SP与裁剪线的交点I;  
情况 (4) 输出线段SP与裁剪线的交点I和终点P

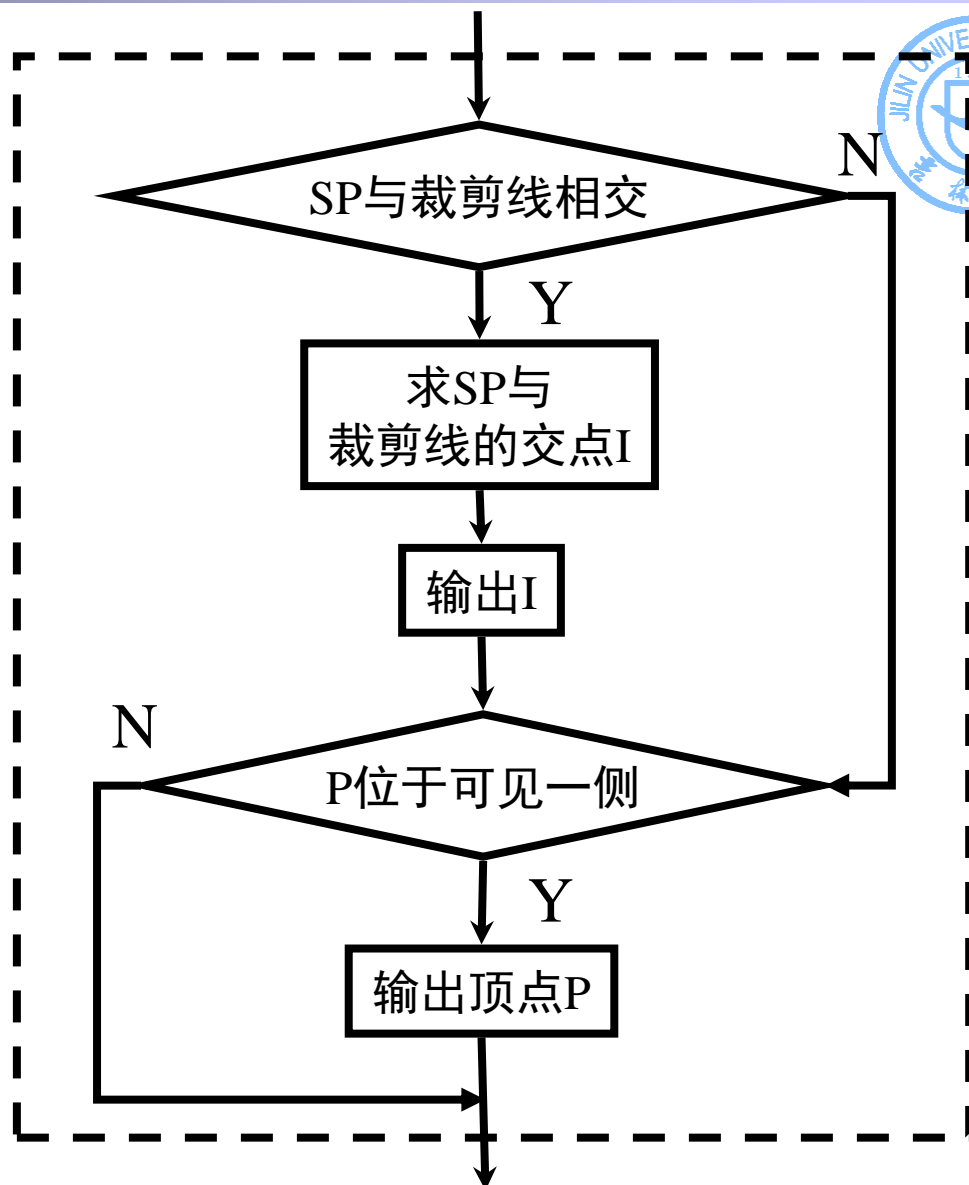
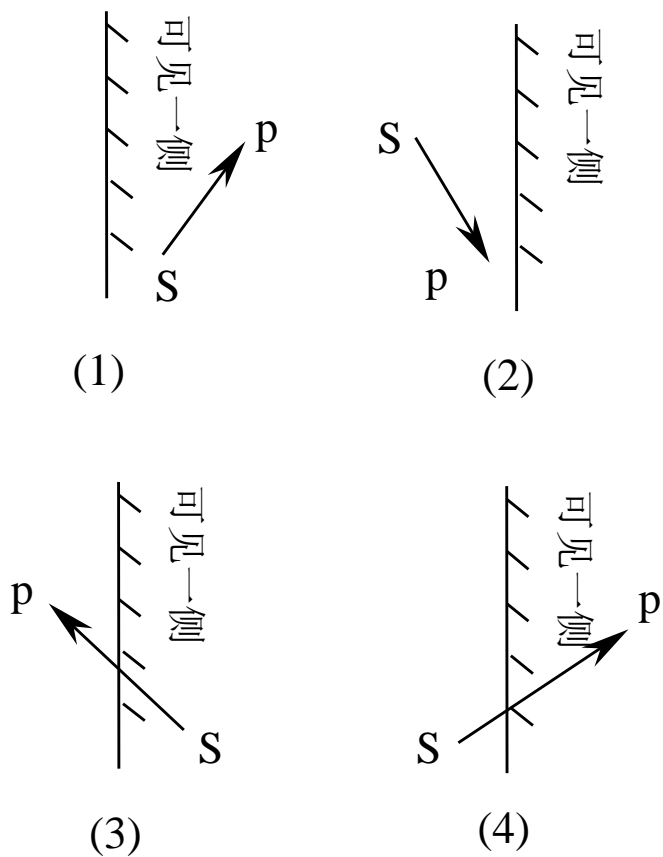




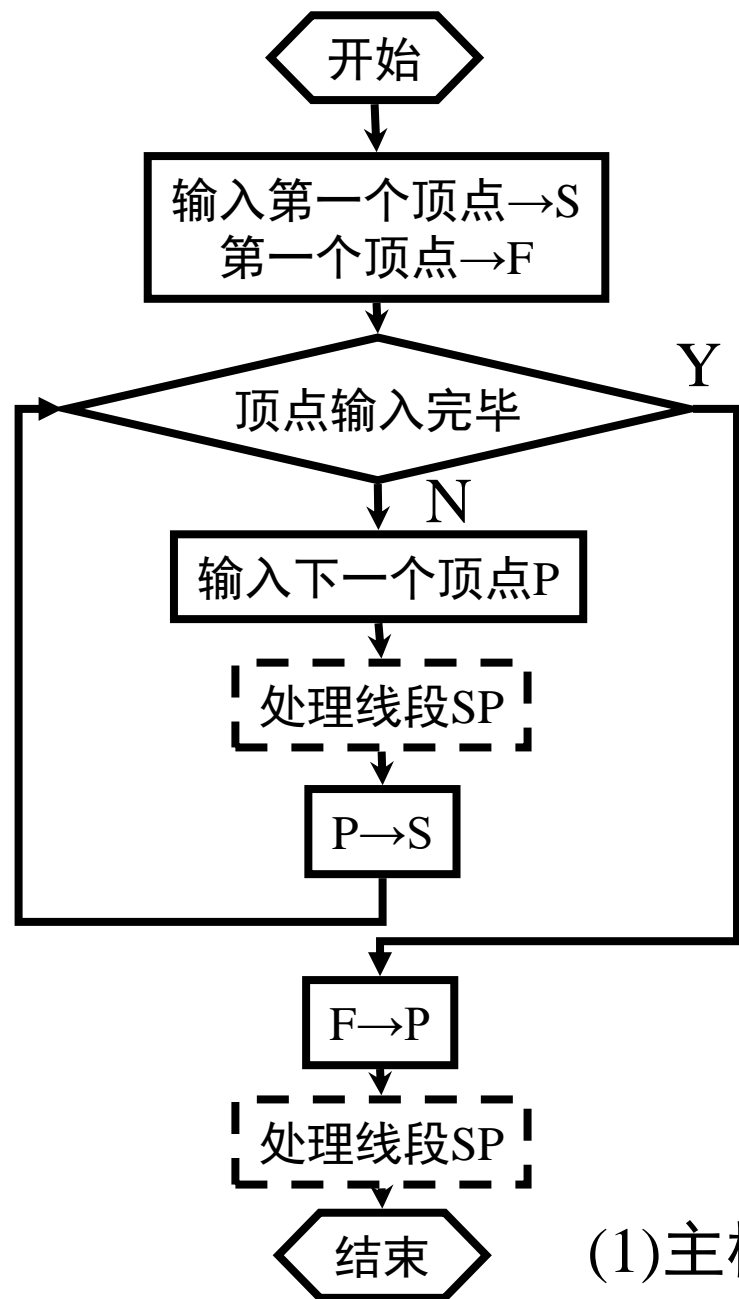
该算法仅用一条裁剪边对多边形进行裁剪，得到一个顶点序列，作为下一条裁剪边处理过程的输入。



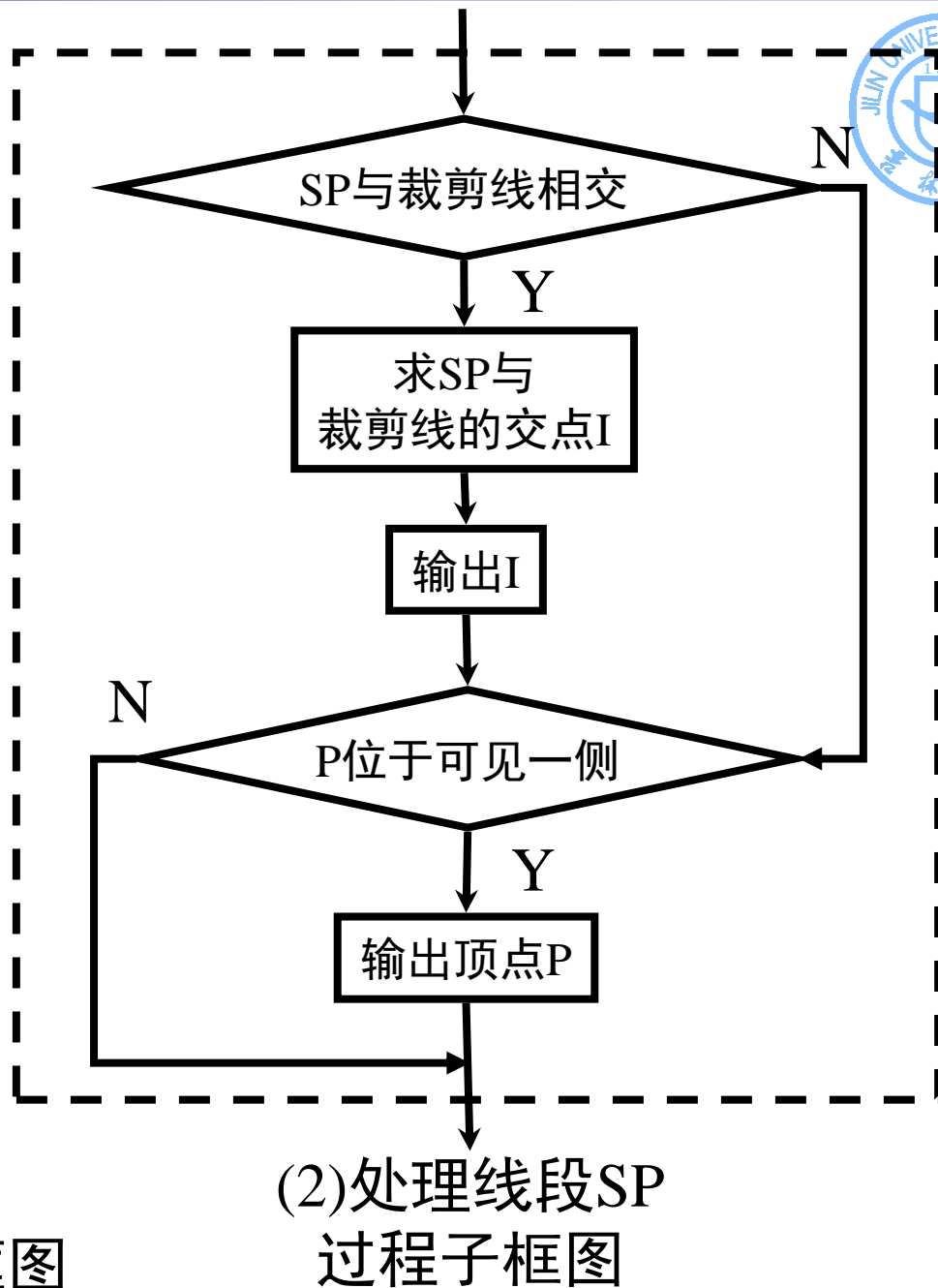
(1)主框图



(2)处理线段SP  
过程子框图



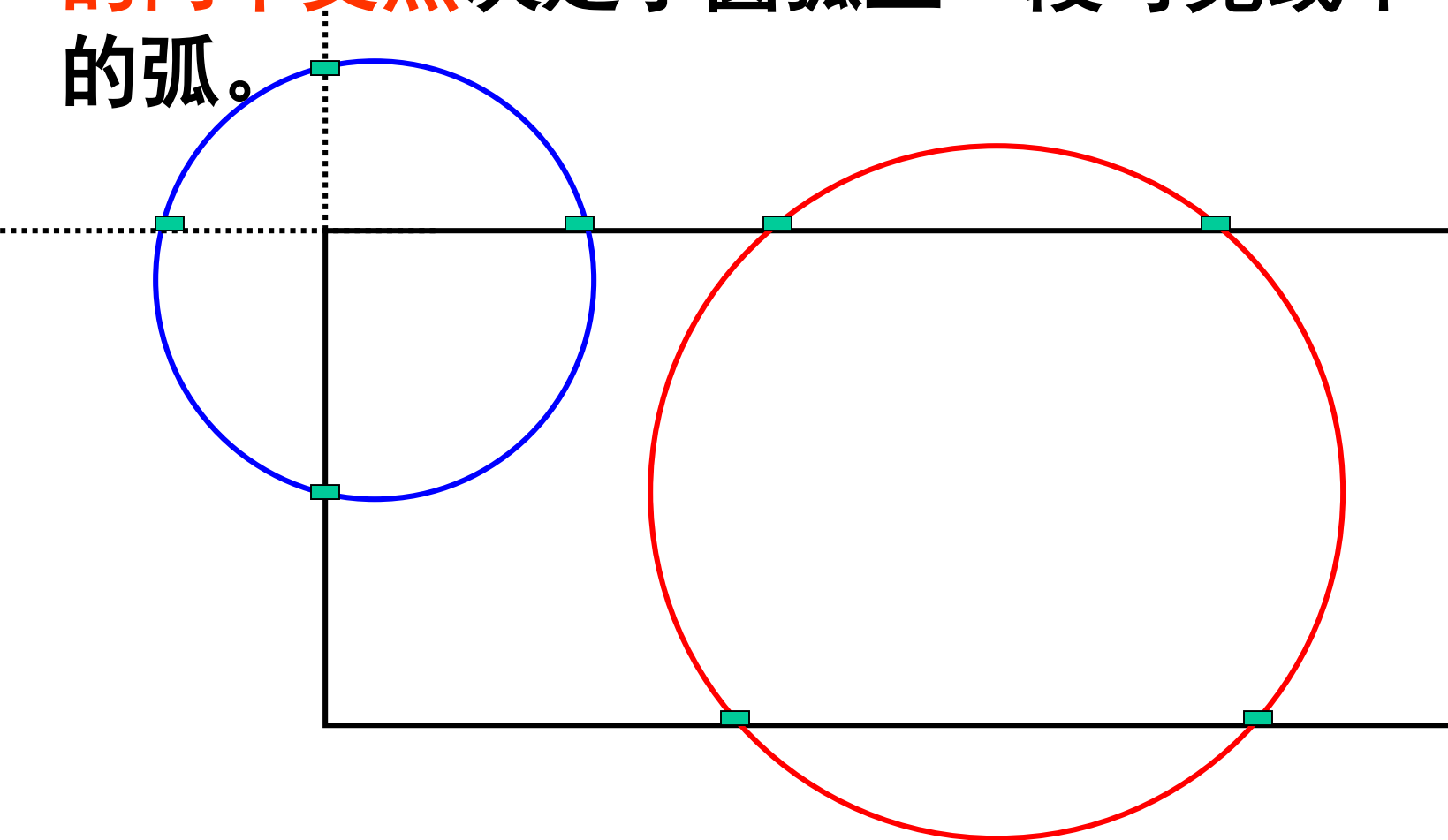
(1)主框图

(2)处理线段SP  
过程子框图



### 3、圆弧裁剪

可把圆弧和窗口四条边的交点求出来，再按交点对圆心幅角的大小排序，排序后，**相邻的两个交点**决定了圆弧上一段可见或不可见的弧。





### 3、圆弧裁剪

#### (1) 计算圆弧与裁剪边的交点

- 相切交点不计在内
- 在延长线上的交点不计在内

#### (2) 交点与圆心连线夹角按逆时针排序

#### (3) 可见圆弧与不可见圆弧交替出现

第一段圆弧的可见性根据  $(x_0+R, y)$  是否在裁剪区域内判断

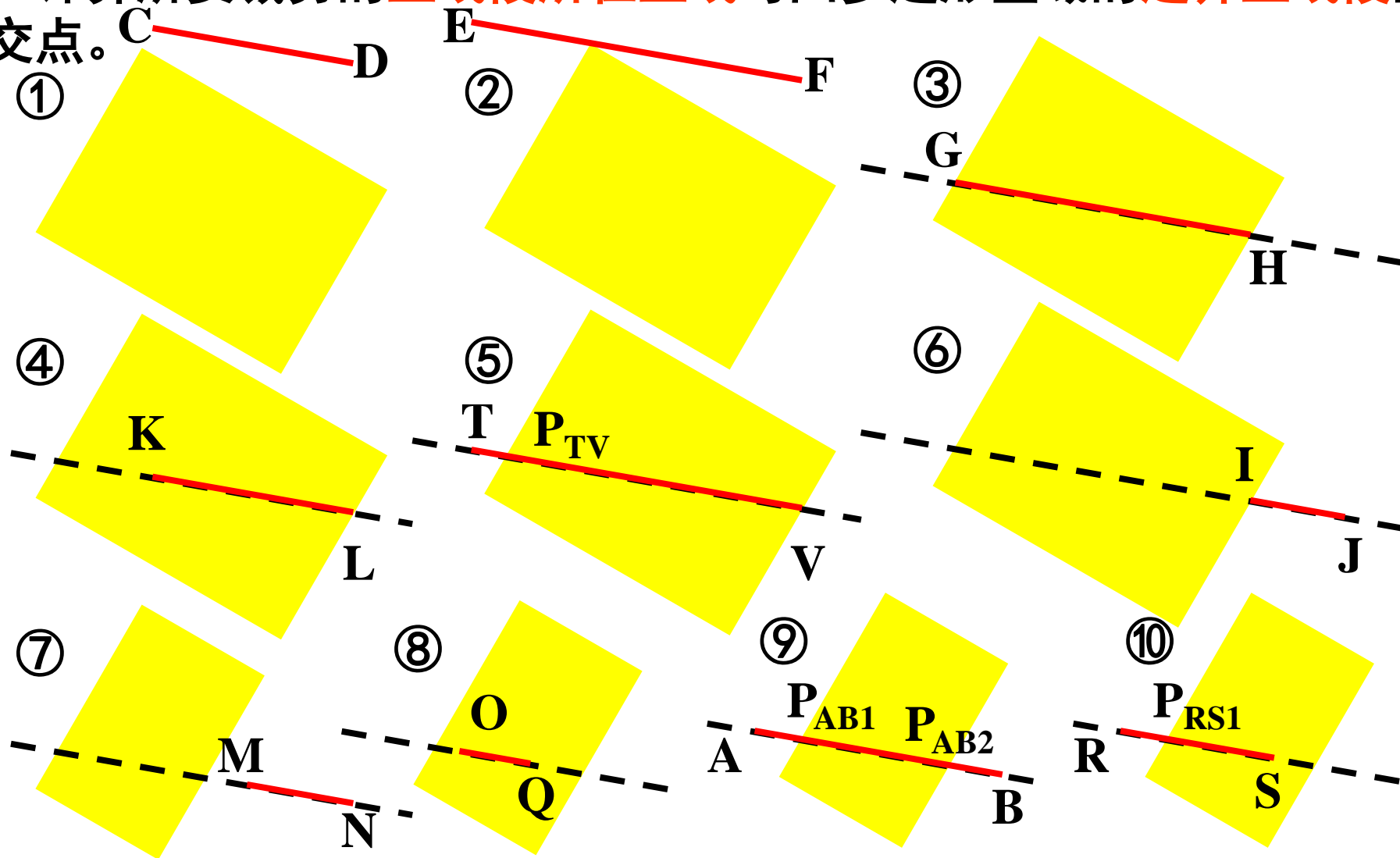
归纳为：

- (1) 圆弧端点在一条裁剪边上，不可见；
- (2) 圆弧端点在相邻裁剪边或对称裁剪边上，可见。



# 裁剪区域为任意凸多边形区域时的直线段的裁剪算法

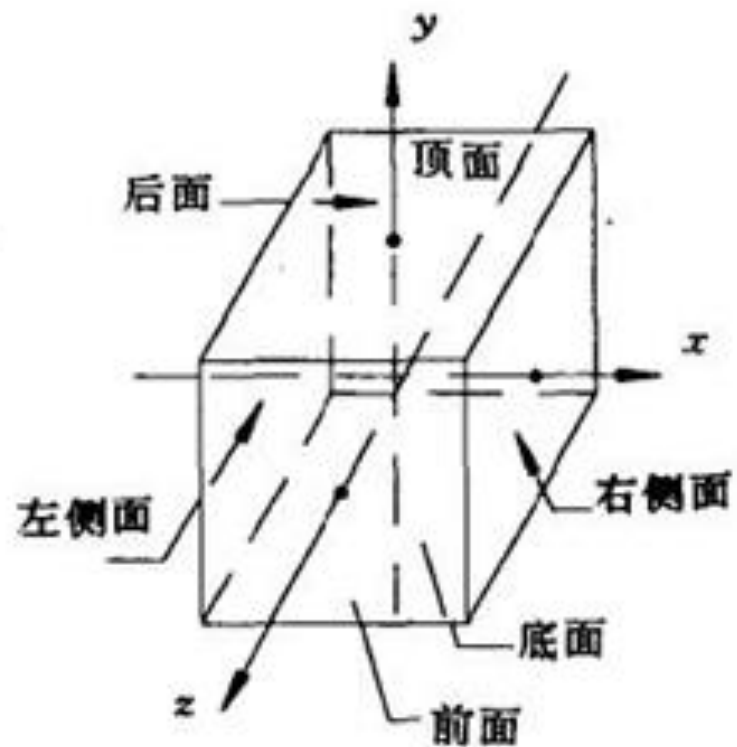
1. 计算所要裁剪的直线段所在直线与凸多边形区域的边界直线段的交点。



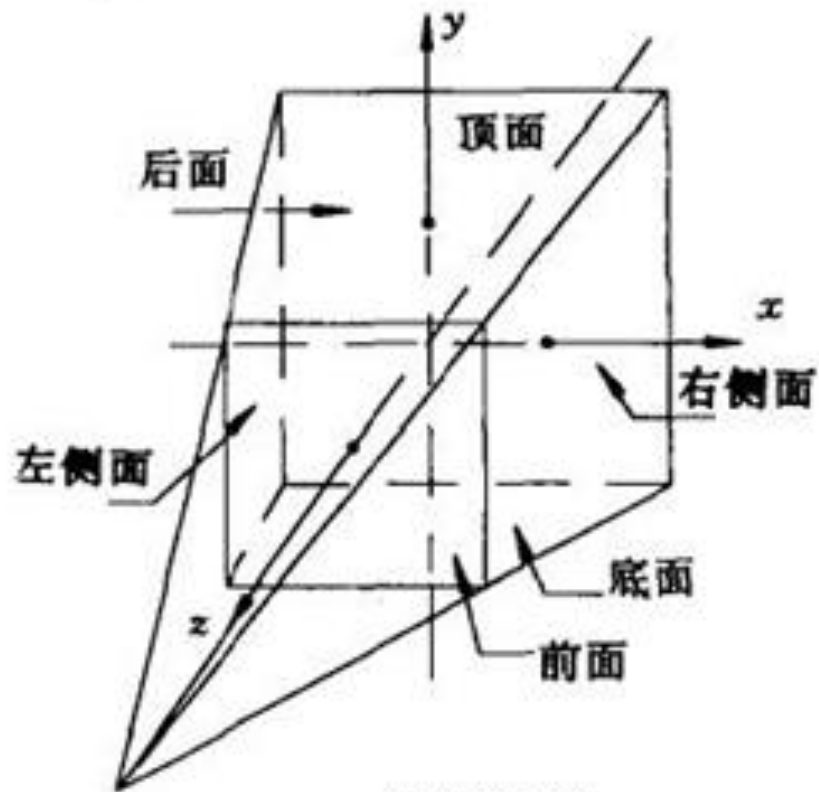




### 三、三维图形的裁剪



(a) 平行投影



(b) 透视投影

图 3.35 两种三维视域



**Cohen-Sutherland 算法推广至三维时，用于判断显然不可见的线段的编码应为六位，这六位的安排是：**

**第一位为1——端点在裁剪体左侧**

**第二位为1——端点在裁剪体右侧**

**第三位为1——端点在裁剪体下侧**

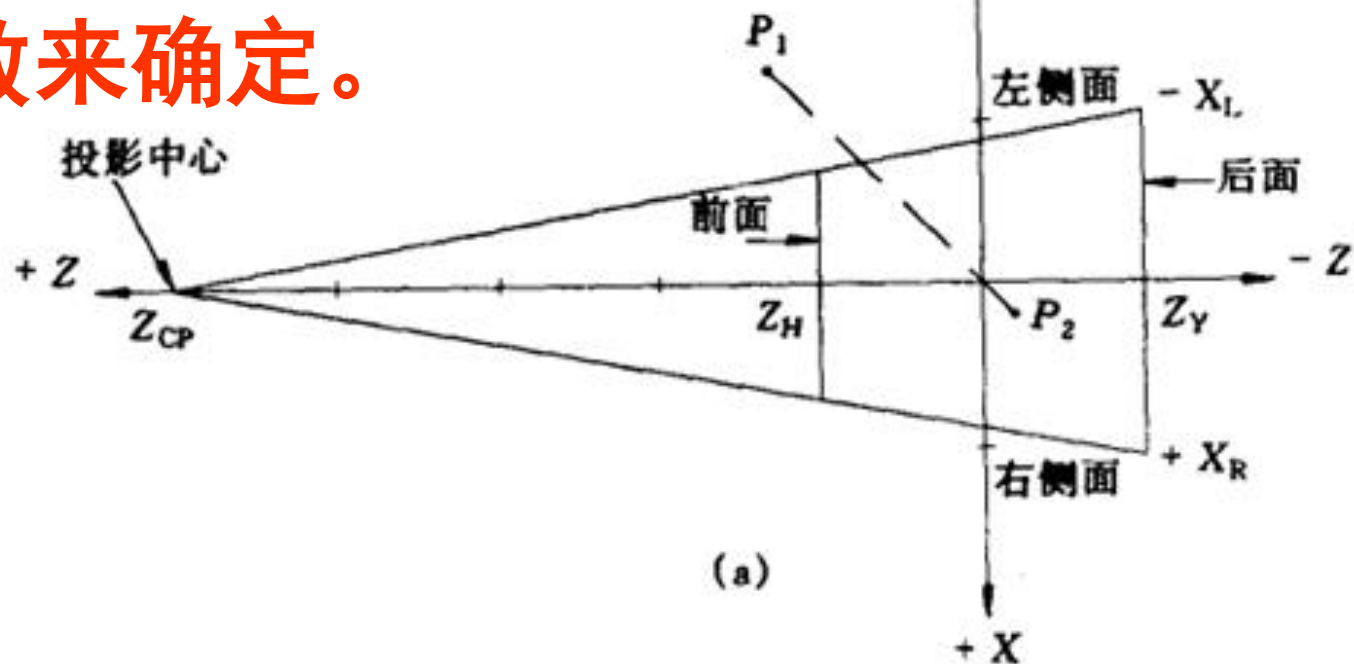
**第四位为1——端点在裁剪体上侧**

**第五位为1——端点在裁剪体前面**

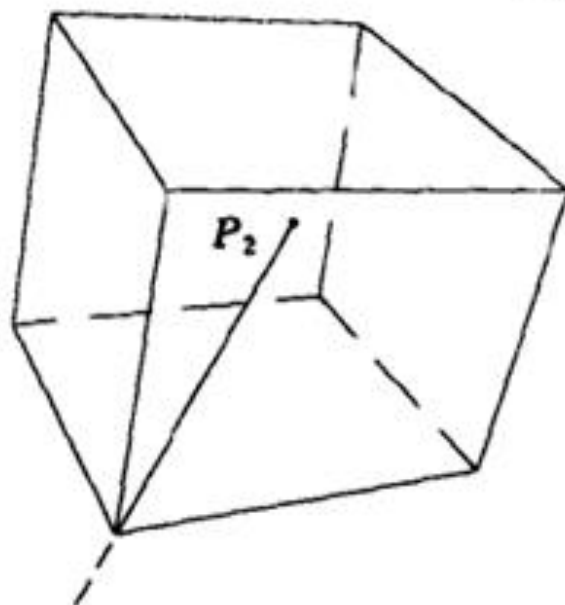
**第六位为1——端点在裁剪体后面**

**当端点在裁剪体内时六位全置零。**

对于透视裁剪体来说，此时需要采用判别函数来确定。



(a)





$$x = \frac{z - Z_{cp}}{Z_Y - Z_{cp}} X_R = z\alpha_1 + \alpha_2$$

其中,  $Z_{CP}$  为投影中心,  $Z_Y$  为裁剪体后面在Z轴上的截距,

$$\alpha_1 = \frac{X_R}{Z_Y - Z_{CP}}$$

$$\alpha_2 = -\alpha_1 Z_{CP}$$



将要确定端点编码的点  $P(x, y, z)$  的  $x$  和  $y$  坐标代入，如果

$$f_R = x - z\alpha_1 - \alpha_2 > 0 \quad P(x, y, z)$$

点在裁剪体右侧面右方

$$f_R = x - z\alpha_1 - \alpha_2 = 0 \quad P(x, y, z)$$

点在裁剪体右侧面上

$$f_R = x - z\alpha_1 - \alpha_2 < 0 \quad P(x, y, z)$$

点在裁剪体右侧面左方



左侧面的判别函数是：

$$f_L = x - z\beta_1 - \beta_2 > 0 \quad P(x, y, z)$$

点在裁剪体左侧面右方

$$f_L = x - z\beta_1 - \beta_2 = 0 \quad P(x, y, z)$$

点在裁剪体左侧面上

$$f_L = x - z\beta_1 - \beta_2 < 0 \quad P(x, y, z)$$

点在裁剪体左侧面左方

其中

$$\beta_1 = \frac{X_L}{Z_Y - Z_{CP}} \quad \beta_2 = -\beta_1 Z_{CP}$$



顶面的判别函数是：

$$f_T = y - z\gamma_1 - \gamma_2 > 0 \quad P(x, y, z)$$

点在裁剪体顶面上方

$$f_T = y - z\gamma_1 - \gamma_2 = 0 \quad P(x, y, z)$$

点在裁剪体顶面上

$$f_T = y - z\gamma_1 - \gamma_2 < 0 \quad P(x, y, z)$$

点在裁剪体顶面下方

其中,

$$\gamma_1 = \frac{Y_T}{Z_Y - Z_{CP}} \quad \gamma_2 = -\gamma_1 Z_{CP}$$



底面的判别函数是：

$$f_B = y - z\delta_1 - \delta_2 > 0$$

$$P(x, y, z)$$

点在裁剪体底面上方

$$f_B = y - z\delta_1 - \delta_2 = 0$$

$$P(x, y, z)$$

点在裁剪体底面上

$$f_B = y - z\delta_1 - \delta_2 < 0$$

$$P(x, y, z)$$

点在裁剪体底面下方

其中,

$$\delta_1 = \frac{Y_B}{Z_Y - Z_{CP}} \quad \delta_2 = -\delta_1 Z_{CP}$$





前面的判别函数是：

$$f_H = z - Z_H > 0 \quad P(x, y, z)$$

点在裁剪体前面前方

$$f_H = z - Z_H = 0 \quad P(x, y, z)$$

点在裁剪体前面上

$$f_H = z - Z_H < 0 \quad P(x, y, z)$$

点在裁剪体前面后方



后面的判别函数是：

$$f_Y = z - Z_Y > 0$$

$$P(x, y, z)$$

点在裁剪体后面前方

$$f_Y = z - Z_Y = 0$$

$$P(x, y, z)$$

点在裁剪体后面上

$$f_Y = z - Z_Y < 0$$

$$P(x, y, z)$$

点在裁剪体后面后方

裁剪体后面的四个边界值

$$X_L \quad X_R \quad Y_T \quad Y_B$$

裁剪体前、后面在Z轴上的截距  $Z_H \quad Z_Y$



# 计算线段与裁剪体的面相交

设直线段的起点和终点分别为  $P_0(x_0, y_0, z_0)$  和  $P_1(x_1, y_1, z_1)$ ，直线方程可以表示成如下的参数方程形式：

$$x = x_0 + (x_1 - x_0)t$$

$$y = y_0 + (y_1 - y_0)t$$

$$z = z_0 + (z_1 - z_0)t$$



当裁剪体为长方体时，其右侧面方程为  $x = X_R$

可由下式求出交点的参数  $t'$ ，从而求出直线与裁剪体边界面的交点坐标：

$$X_R = (x_1 - x_0)t' + x_0, \quad t' = \frac{X_R - x_0}{x_1 - x_0}$$



当裁剪体为棱锥体时，根据裁剪体的右侧面方程可以求出交点的参数  $t'$

$$t' = \frac{z_0 \alpha_1 + \alpha_2 - x_0}{(x_1 - x_0) - \alpha_1 (z_1 - z_0)}$$

其中

$$\alpha_1 = \frac{X_R}{Z_Y - Z_{CP}} \quad \alpha_2 = -\alpha_1 Z_{CP}$$

同理可求得裁剪体为长方体或棱锥体时，直线与其它五个边界面的交点坐标



梁友栋-Barsky算法也可以推广到三维情况下。当视域为立方体时，这种推广是直接的。当视域为棱台时，对于  $x = \pm z$ ， $y = \pm z$  四个平面来说，对应于二维裁剪时的  $Q$ 值和  $D$ 值可如下取值：

$$Q_l = -(\Delta x + \Delta z),$$

$$Q_r = (\Delta x - \Delta z),$$

$$Q_b = -(\Delta y + \Delta z),$$

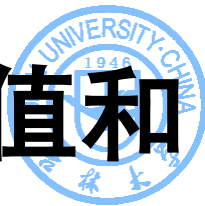
$$Q_t = (\Delta y - \Delta z),$$

$$D_l = z_0 + x_0$$

$$D_r = z_0 - x_0$$

$$D_b = y_0 + z_0$$

$$D_t = z_0 - y_0$$



对 $z=Z_H$ 和 $z=Z_Y$ 两个平面，相应的  $Q$  值和  $D$  值可如下取值：

$$Q_f = -\Delta z, \quad D_f = z_0 - Z_H$$

$$Q_{ba} = \Delta z, \quad D_{ba} = Z_Y - z_0$$

其中， $\Delta x = x_1 - x_0$ ， $\Delta y = y_1 - y_0$ 和  $\Delta z = z_1 - z_0$ 可知，  
相应平面与  $P_0/P_1$  的交点的参数值为

$$t_i = \frac{D_i}{Q_i} \quad (i = l, r, b, t, f, ba)$$



```
void T_Barsky(double x0, y0,z0, x2, y2,z2)
{
double t0, t1, deltax, deltay,deltaz;
t0=0.0; t1=1.0;
deltax=x2-x0;
deltaz=z2-z0;
if (!cansee(-deltax-deltaz, x0+z0, t0, t1)) return;
if (!cansee(deltax-deltaz, z0-x0, t0, t1)) return;
deltay=y2-y0;
if (!cansee(-deltay-deltaz, y0+z0, t0, t1)) return;
if (!cansee(deltay-deltaz, z0-y0, t0, t1)) return;
if(!cansee(-deltaz,z0-zh,t0,t1)) return;
if(!cansee(deltaz,zy-z0,t0,t1)) return;
```





```
x2=x0+t1*deltax;  
y2=y0+t1*deltay;  
z2=z0+t1*deltaz;  
x0=x0+t0*deltax;  
y0=y0+t0*deltay;  
z0=z0+t0*deltaz;  
showline(x0, y0,z0, x2, y2,z2);  (显示可见线  
段)  
}
```

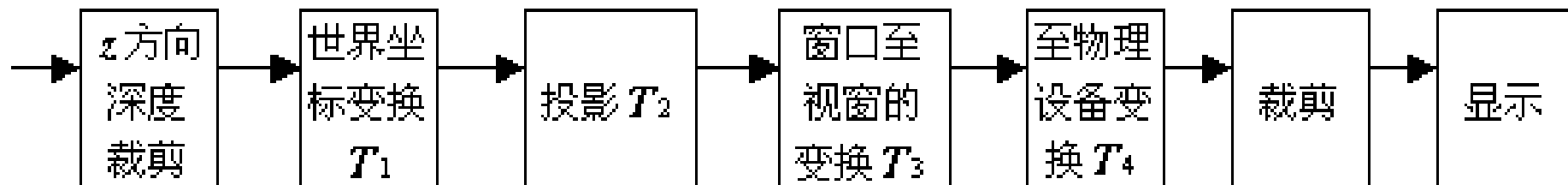


图 3.36 简单的图形处理流程

