

# 6 观察变换

---

□ 二维观察

□ 三维观察

# 6.1 二维观察

---

- 基本概念
- 二维观察变换
- 二维裁剪

## 二维观察——基本概念

---

- 在计算机图形学中，将在用户坐标系中需要进行观察和处理的一个坐标区域称为窗口（**Window**）。
- 将窗口映射到显示设备上的坐标区域称为视区（**Viewport**）。

# 二维观察——基本概念

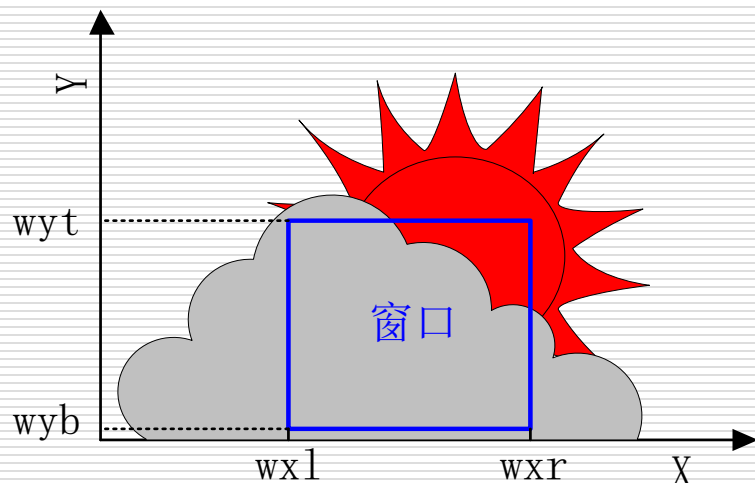
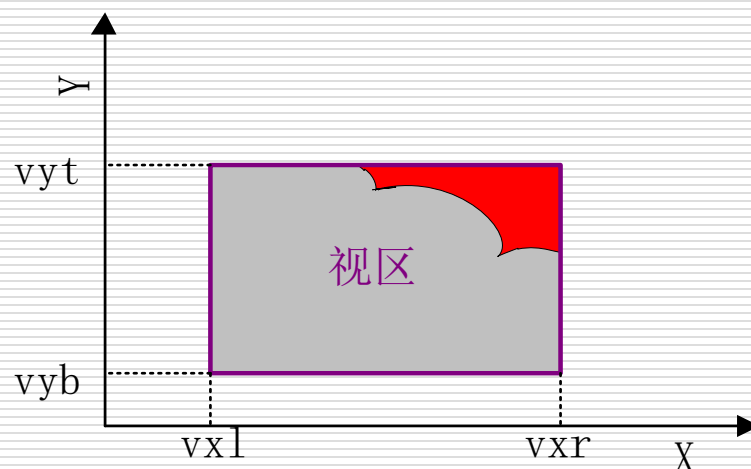


图6-1 (a) 用户坐标系中的窗口



(b) 屏幕坐标系中的视区

- 要将窗口内的图形在视区中显示出来，必须经过将窗口到视区的变换（**Window-Viewport Transformation**）处理，这种变换就是观察变换（**Viewing Transformation**）。

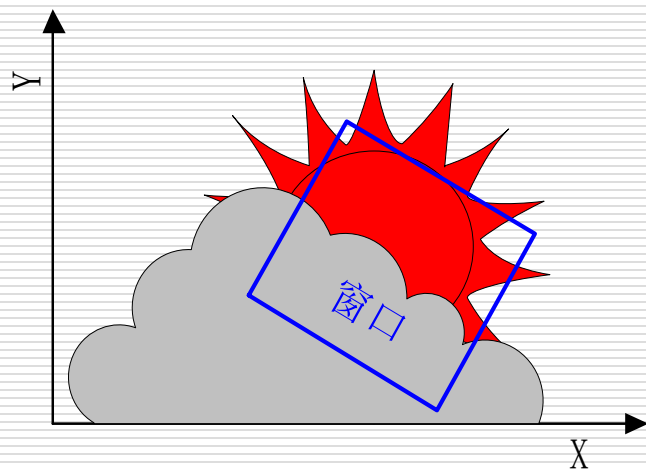
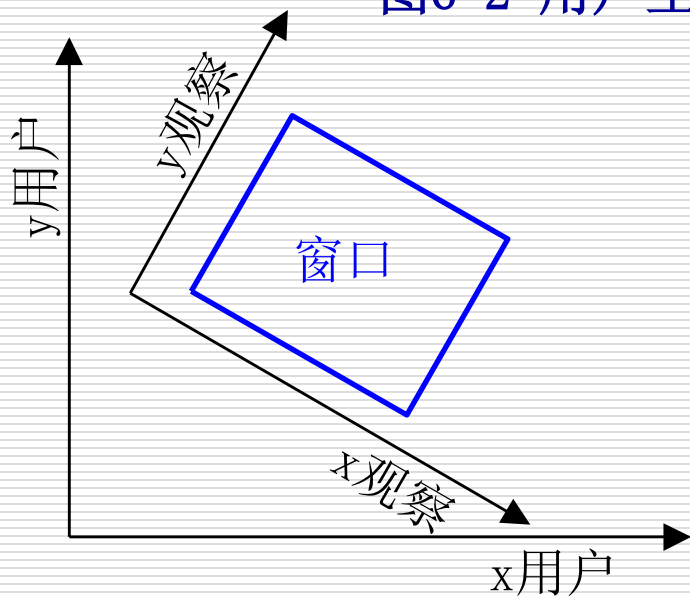
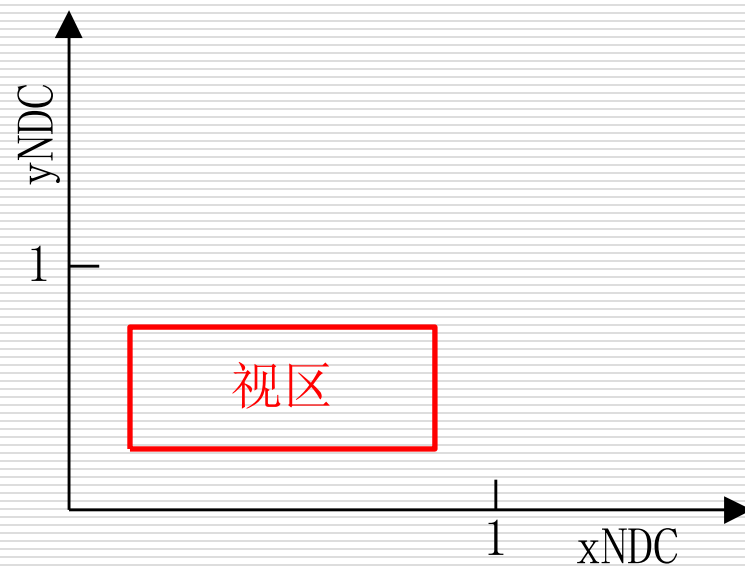


图6-2 用户坐标系中旋转的窗口



(a) 观察坐标系



(b) 规格化设备坐标系

图6-3 观察坐标系和规格化设备坐标系

## 二维观察——基本概念

---

- 观察坐标系(**View Coordinate**)是依据窗口的方向和形状在用户坐标平面中定义的直角坐标系。
- 规格化设备坐标系 (**Normalized Device Coordinate**)也是直角坐标系，它是将二维的设备坐标系规格化到  $(0.0, 0.0)$  到  $(1.0, 1.0)$  的坐标范围内形成的。

## 二维观察——基本概念

- 引入了观察坐标系和规格化设备坐标系后，观察变换分为如下图所示的几个步骤，通常称为**二维观察流程**。



图6-4 二维观察流程

# 二维观察——基本概念

## □ 变焦距效果

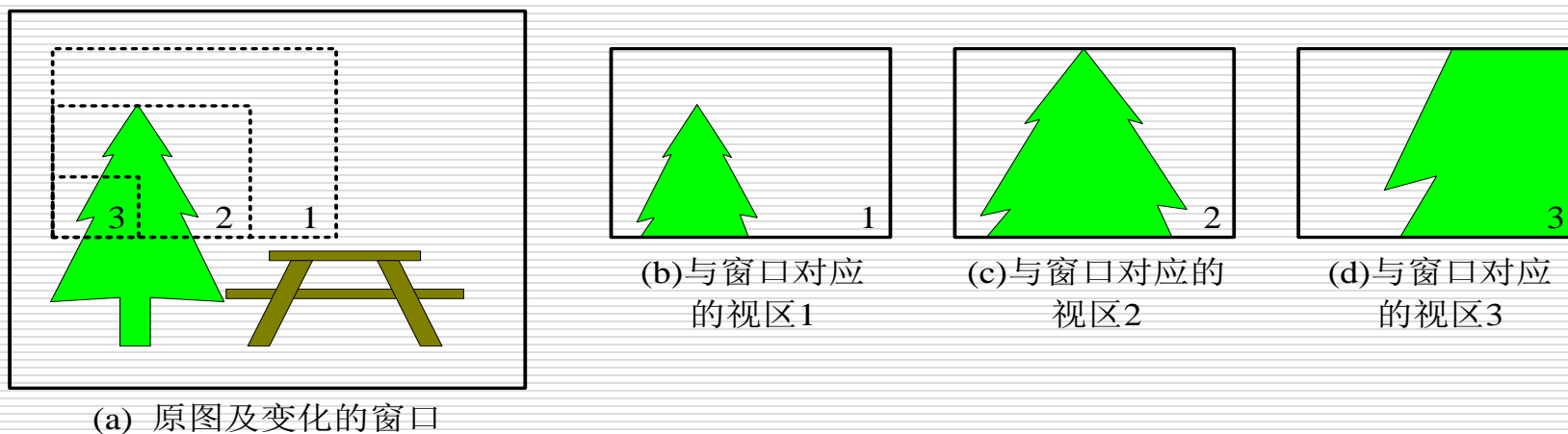


图6-5 变焦距效果（窗口变、视区不变）



## 二维观察——基本概念

### □ 整体放缩效果

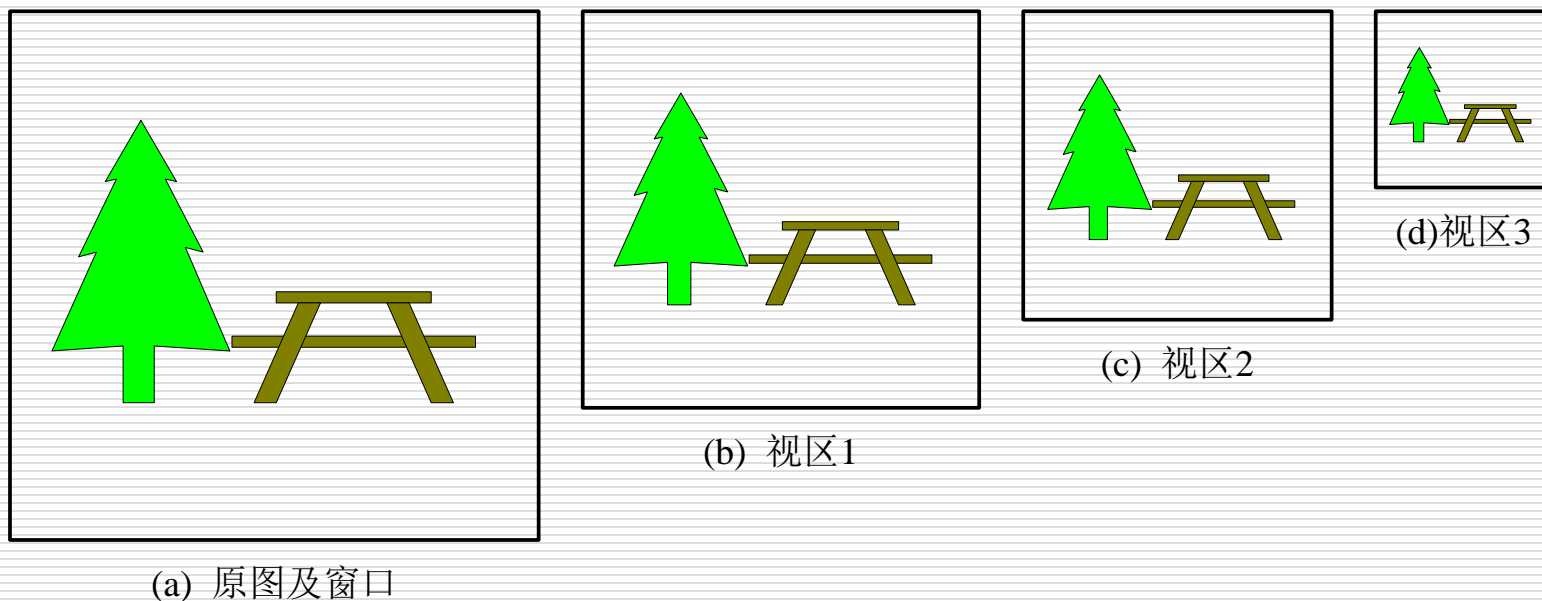


图6-6 整体放缩效果（窗口不变、视区变）

### □ 漫游效果

# 用户坐标系到观察坐标系的变换

- 用户坐标系到观察坐标系的变换分由两个变换步骤合成：
  - ◆ 将观察坐标系原点移动到用户坐标系原点；

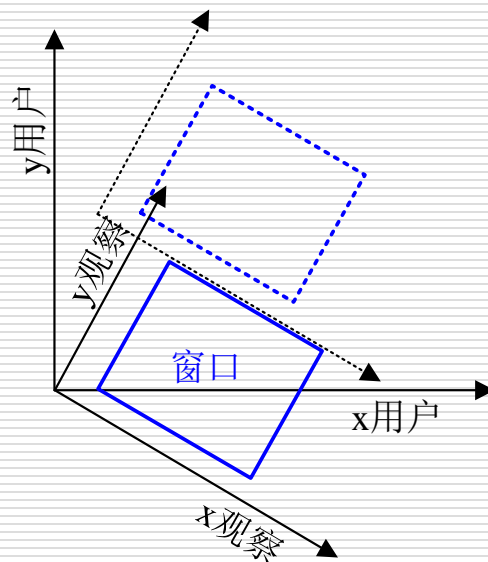


图6-7 (a) 平移使坐标原点重合

# 用户坐标系到观察坐标系的变换

## ◆ 绕原点旋转使两坐标系重合

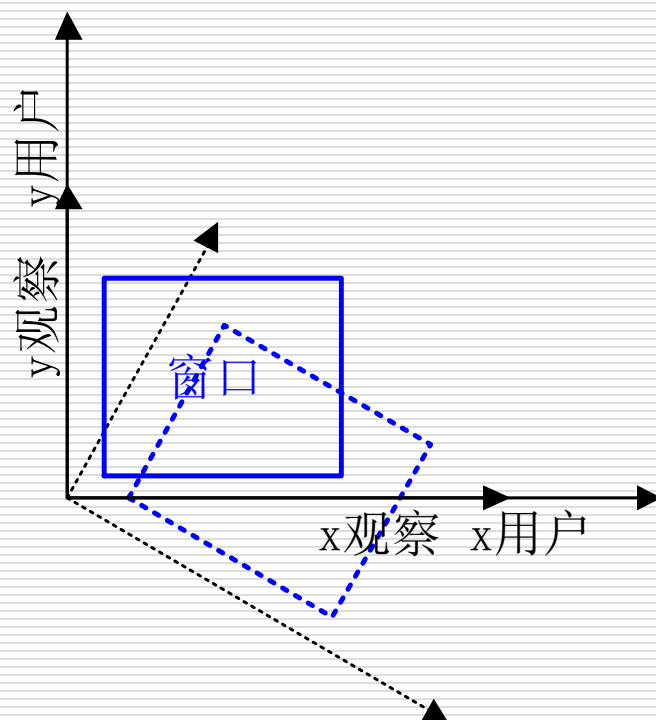
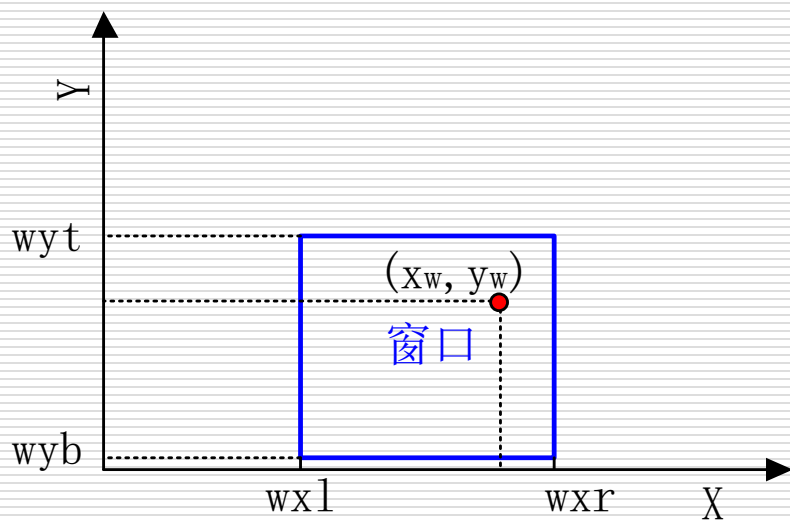
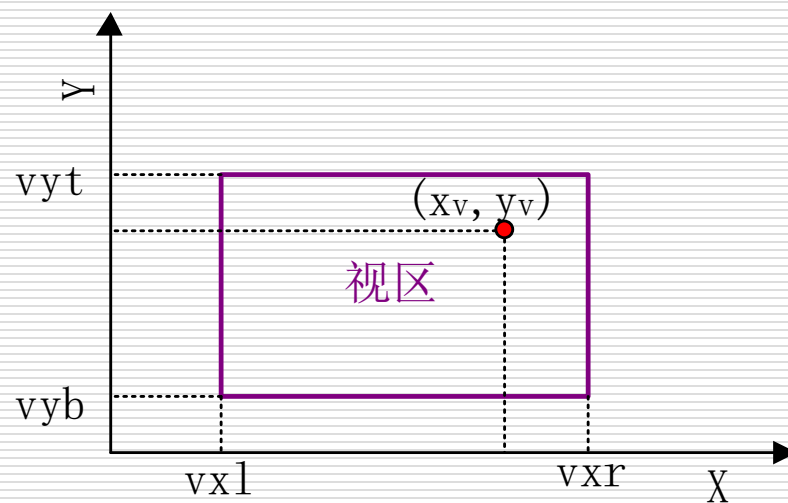


图6-7 (b) 旋转使坐标轴重合

# 窗口到视区的变换



(a) 窗口中的点



(b) 视区中的点

图6-8 窗口到视区的变换

# 窗口到视区的变换

---

□ 要将窗口内的点  $(x_w, y_w)$  映射到相对应的视区内的点  $(x_v, y_v)$  需进行以下步骤：

(1) 将窗口左下角点移至用户系统系的坐标原点；

(2) 针对原点进行比例变换；

(3) 进行反平移。

# 裁剪

- 在二维观察中，需要在观察坐标系下对窗口进行裁剪，即只保留窗口内的那部分图形，去掉窗口外的图形。
- 假设窗口是标准矩形，即边与坐标轴平行的矩形，由上（ $y=wyt$ ）、下（ $y=wyb$ ）、左（ $x=wxl$ ）、右（ $x=wxr$ ）四条边描述。

# 裁剪——点的裁剪

---

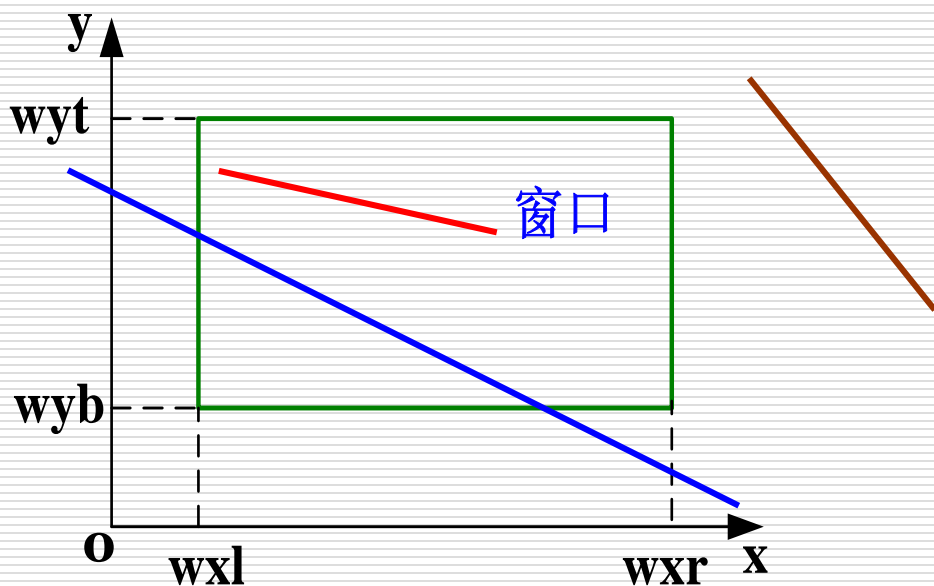
$$wxl \leq x \leq wxr,$$

$$\text{且 } wyb \leq y \leq wyt$$

# 二维直线段的裁剪

已知条件:

- (1) 窗口边界  $wxl$ ,  $wxr$ ,  $wyb$ ,  $wyt$  的坐标值;
- (2) 直线段端点  $p_1p_2$  的坐标值  $x_1, y_1, x_2, y_2$ 。





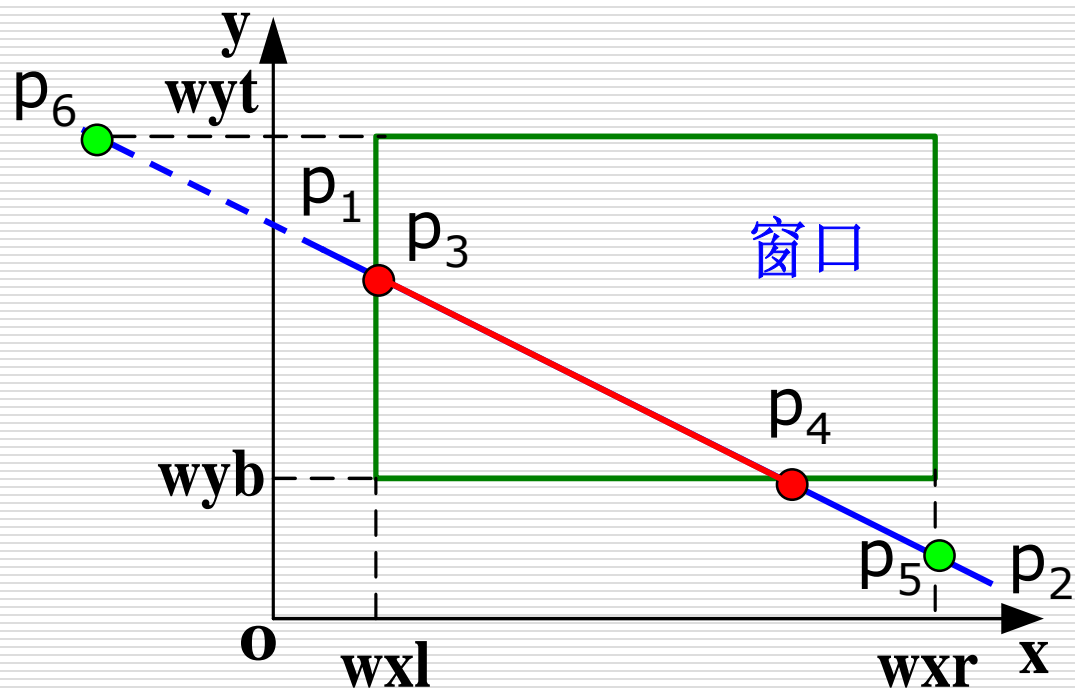


图6-10 实交点与虚交点

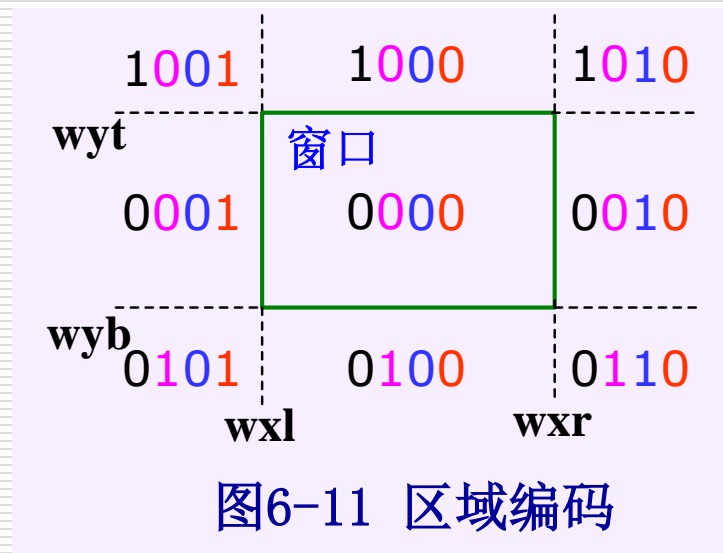
- 实交点：直线段与窗口矩形边界的交点；
- 虚交点：处于直线段延长线或窗口边界延长线上的交点。

# Cohen-Sutherland算法

编码：对于任一端点 $(x,y)$ ，根据其坐标所在的区域，赋予一个4位的二进制码 $D_3D_2D_1D_0$ 。

编码规则如下：

- (1) 若 $x < wxl$ ， $D_0=1$ ，否则 $D_0=0$ ；
- (2) 若 $x > wxr$ ， $D_1=1$ ，否则 $D_1=0$ ；
- (3) 若 $y < wyb$ ， $D_2=1$ ，否则 $D_2=0$ ；
- (4) 若 $y > wyt$ ， $D_3=1$ ，否则 $D_3=0$ 。



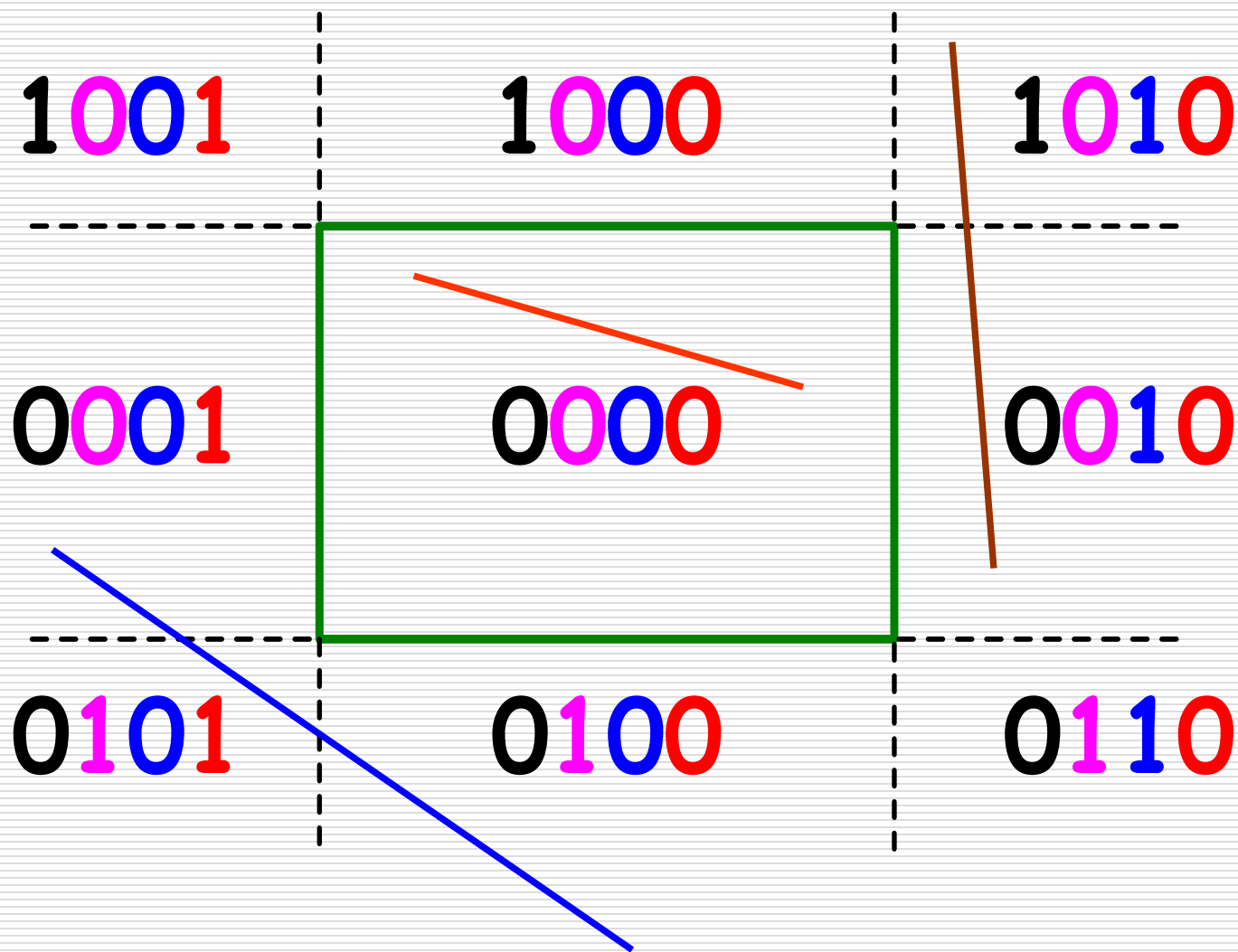


图6-12 区域编码规则

# Cohen-Sutherland算法

---

## (1) 判断

裁剪一条线段时，先求出直线段端点 $p_1$ 和 $p_2$ 的编码 $code1$ 和 $code2$ ，然后：

a. 若 $code1 | code2 = 0$ ，对直线段简取之；

b. 若 $code1 \& code2 \neq 0$ ，对直线段简弃之；

# Cohen-Sutherland算法

---

## (2) 求交

若上述判断条件不成立，则需求出直线段与窗口边界的交点。

a. 左、右边界交点的计算： $y = y_1 + k(x - x_1)$ ;

b. 上、下边界交点的计算： $x = x_1 + (y - y_1)/k$ 。

其中， $k = (y_2 - y_1) / (x_2 - x_1)$ 。

# Cohen-Sutherland算法

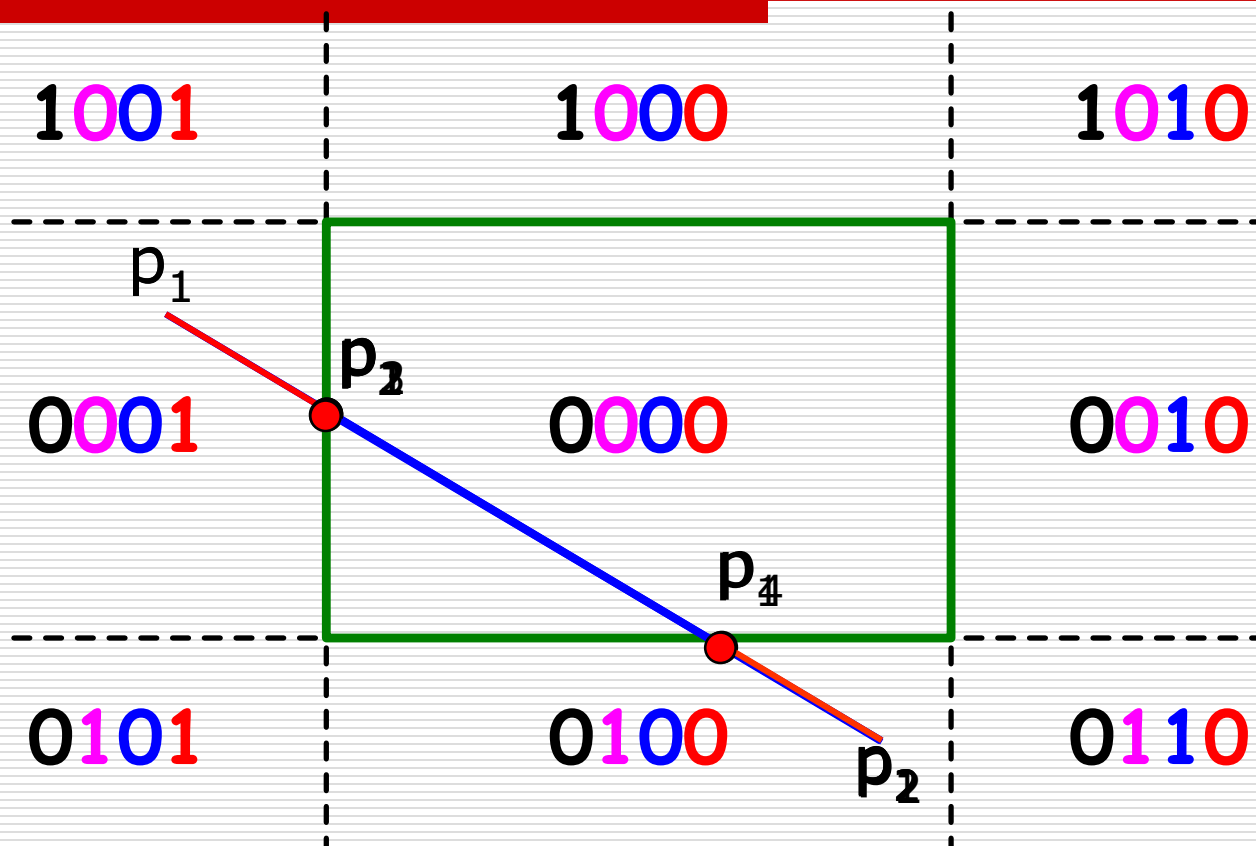


图6-13 区域编码算法

# Cohen-Sutherland算法

---

- 用编码方法实现了对完全可见和不可见直线段的快速接受和拒绝；
- 求交过程复杂，有冗余计算，并且包含浮点运算，不利于硬件实现。

# 中点分割算法

□ 中点分割算法的核心思想是通过二分逼近来确定直线段与窗口的交点。

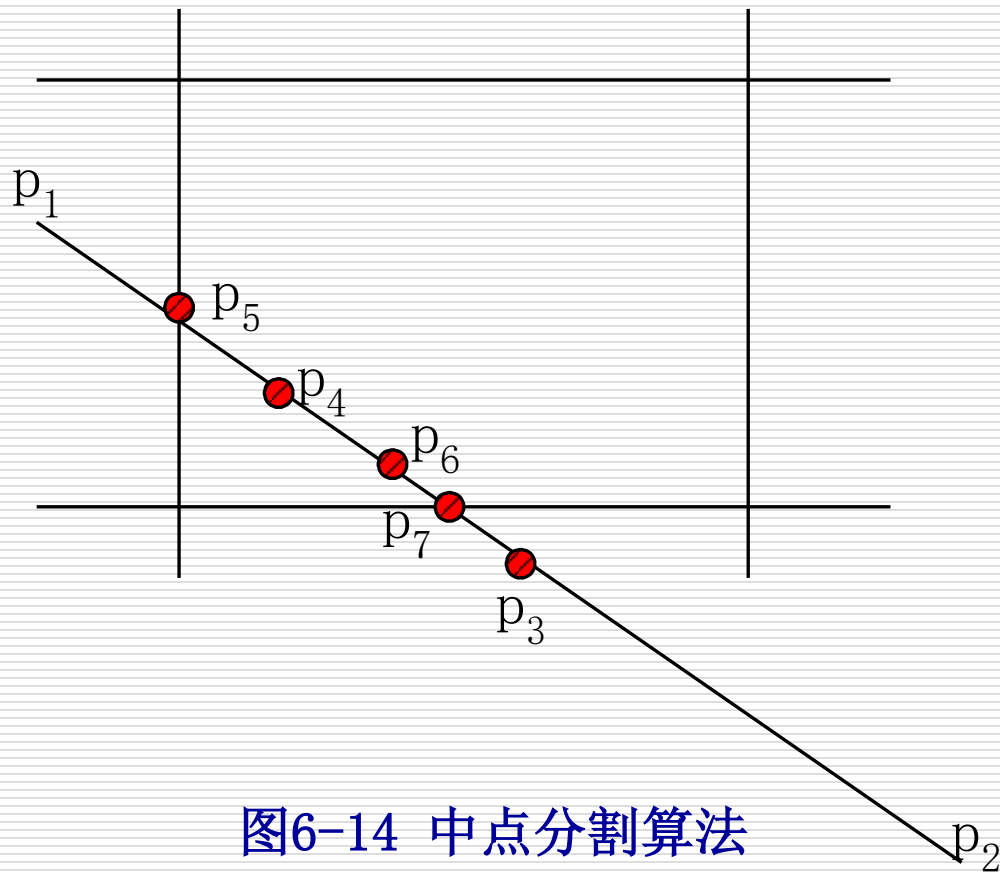


图6-14 中点分割算法



# 中点分割算法

□ 特点：主要计算过程只用到加法或位移运算，易于硬件实现，同时适合于并行计算。

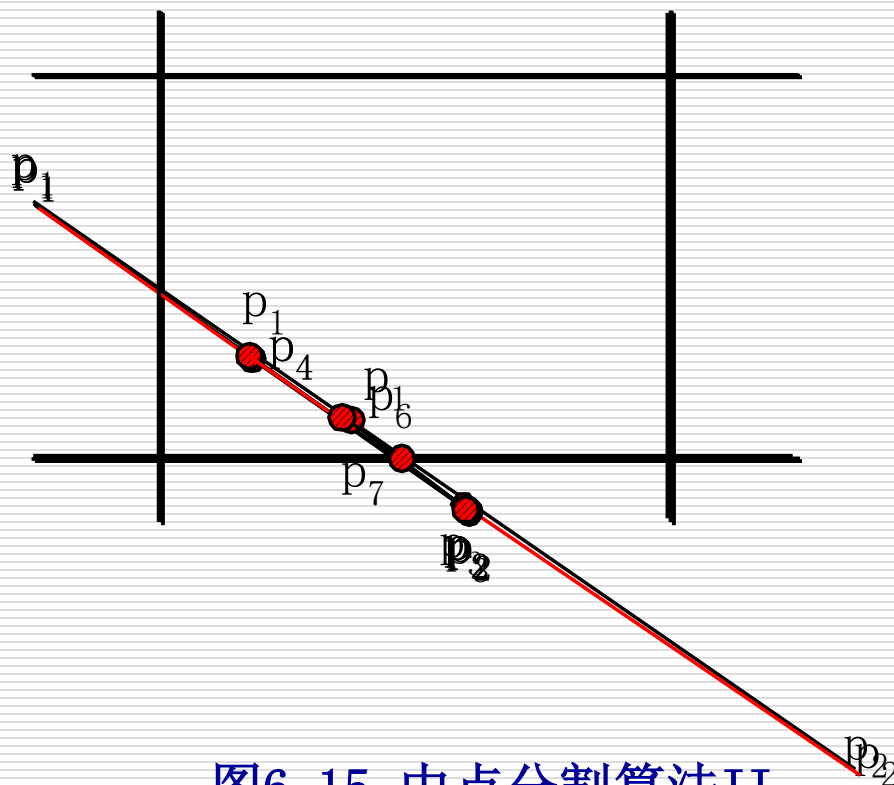


图6-15 中点分割算法II

# Liang-Barsky算法

分析

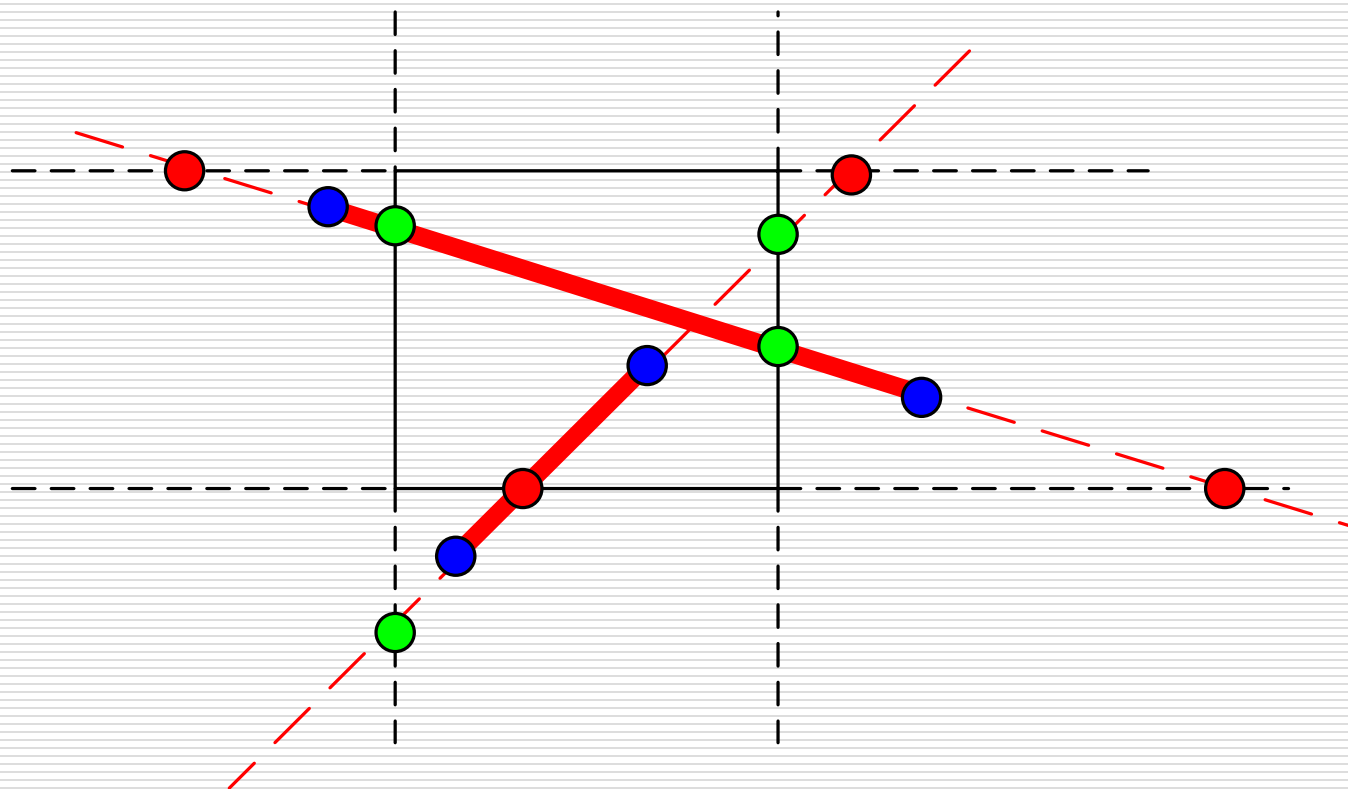


图6-16 Liang-Barsky算法

# Liang-Barsky算法

---

直线的参数方程

$$\begin{aligned}x &= x_1 + u \cdot (x_2 - x_1) \\ y &= y_1 + u \cdot (y_2 - y_1)\end{aligned} \quad 0 \leq u \leq 1$$

对于直线上一点  $(x, y)$ , 若它在窗口内则有

$$wxl \leq x_1 + u \cdot (x_2 - x_1) \leq wxr$$

$$wxb \leq y_1 + u \cdot (y_2 - y_1) \leq wyt$$

$$u \cdot (x_1 - x_2) \leq x_1 - wxl$$

$$u \cdot (x_2 - x_1) \leq wxr - x_1$$

$$u \cdot (y_1 - y_2) \leq y_1 - wyb$$

$$u \cdot (y_2 - y_1) \leq wyt - y_1$$

$$p_1 = -(x_2 - x_1) \quad q_1 = x_1 - wxl$$

$$p_2 = x_2 - x_1 \quad q_2 = wxr - x_1$$

$$p_3 = -(y_2 - y_1) \quad q_3 = y_1 - wyb$$

$$p_4 = y_2 - y_1 \quad q_4 = wyt - y_1$$

令

则有  $u \cdot p_k \leq q_k$

# Liang-Barsky 算法

$$\text{由 } u \cdot p_k \leq q_k$$

$$\left\{ \begin{array}{l} \frac{q_k}{p_k} (p_k < 0) \leq u \leq \frac{q_k}{p_k} (p_k > 0) \quad k = 1, 2 \\ \frac{q_k}{p_k} (p_k < 0) \leq u \leq \frac{q_k}{p_k} (p_k > 0) \quad k = 3, 4 \\ 0 \leq u \leq 1 \end{array} \right.$$

$$p_1 = -(x_2 - x_1)$$

$$p_2 = x_2 - x_1$$

$$p_3 = -(y_2 - y_1)$$

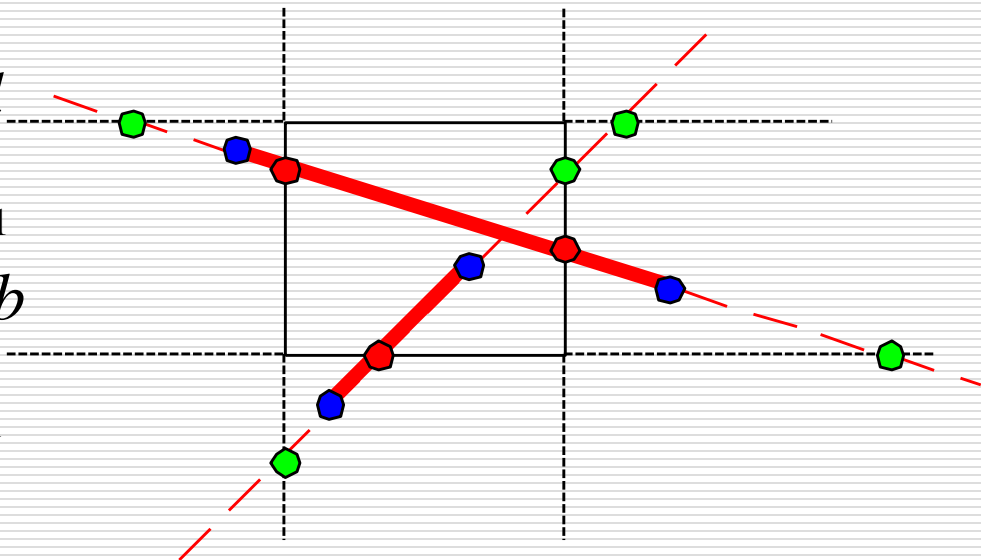
$$p_4 = y_2 - y_1$$

$$q_1 = x_1 - wxl$$

$$q_2 = wxr - x_1$$

$$q_3 = y_1 - wxb$$

$$q_4 = wyt - y_1$$



一般情况:

图6-17 Liang-Barsky算法

$$u_{\max} = \max(0, u_k \mid p_k < 0, u_k \mid p_k < 0)$$

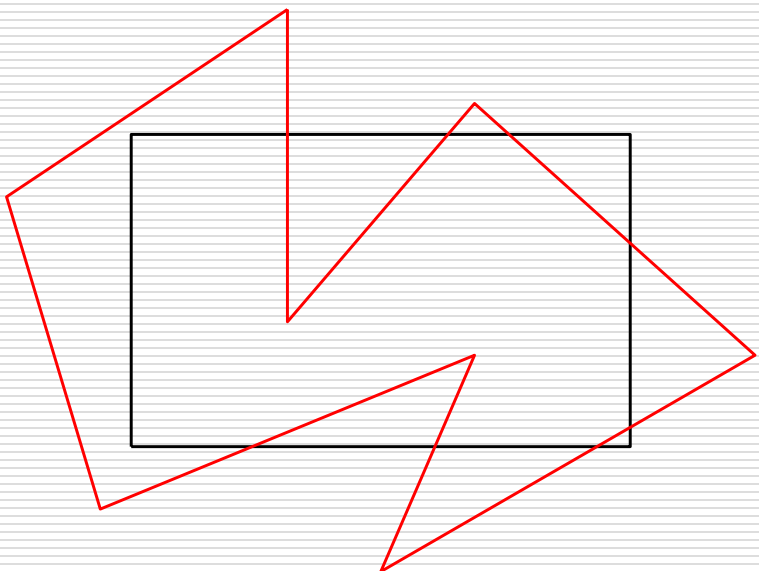
$$u_{\min} = \min(u_k \mid p_k > 0, u_k \mid p_k > 0, 1)$$

算法步骤:

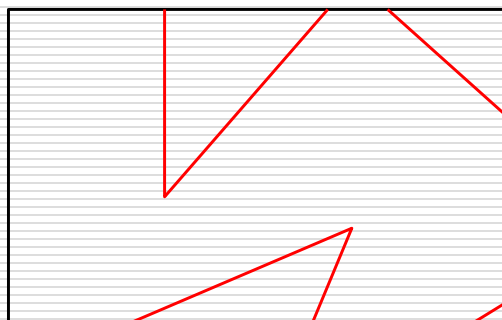
- (1) 输入直线段的两端点坐标:  $(x_1, y_1)$  和  $(x_2, y_2)$ , 以及窗口的四条边界坐标:  $wyt$ 、 $wyb$ 、 $wxl$  和  $wxr$ 。
- (2) 若  $\Delta x = 0$ , 则  $p_1 = p_2 = 0$ 。此时进一步判断是否满足  $q_1 < 0$  或  $q_2 < 0$ , 若满足, 则该直线段不在窗口内, 算法转(7)。否则, 满足  $q_1 > 0$  且  $q_2 > 0$ , 则进一步计算  $u_1$  和  $u_2$ 。算法转(5)。
- (3) 若  $\Delta y = 0$ , 则  $p_3 = p_4 = 0$ 。此时进一步判断是否满足  $q_3 < 0$  或  $q_4 < 0$ , 若满足, 则该直线段不在窗口内, 算法转(7)。否则, 满足  $q_1 > 0$  且  $q_2 > 0$ , 则进一步计算  $u_1$  和  $u_2$ 。算法转(5)。
- (4) 若上述两条均不满足, 则有  $p_k \neq 0$  ( $k=1, 2, 3, 4$ )。此时计算  $u_1$  和  $u_2$ 。
- (5) 求得  $u_1$  和  $u_2$  后, 进行判断: 若  $u_1 > u_2$ , 则直线段在窗口外, 算法转(7)。若  $u_1 < u_2$ , 利用直线的参数方程求得直线段在窗口内的两端点坐标。
- (6) 利用直线的扫描转换算法绘制在窗口内的直线段。算法结束。

# 多边形的裁剪

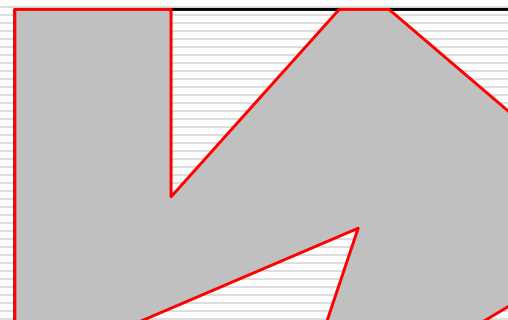
问题的提出:



(a) 裁剪前



(b) 直接采用直线段  
裁剪的结果



(c) 正确的裁剪结果

图6-18 多边形裁剪



# Sutherland-Hodgeman 多边形裁剪

---

- 基本思想：将多边形的边界作为一个整体，每次用窗口的一条边界对要裁剪的多边形进行裁剪，体现分而治之的思想。

# Sutherland-Hodgeman 多边形裁剪

## □ 算法实施策略

- 为窗口各边界裁剪的多边形存储输入与输出顶点表。在窗口的一条裁剪边界处理完所有顶点后，其输出顶点表将用窗口的下一条边界继续裁剪。
- 窗口的一条边以及延长线构成的裁剪线把平面分为两个区域，包含窗口区域的区域称为可见侧；不包含窗口区域的域为不可见侧。

# Sutherland-Hodgeman 多边形裁剪

■ 沿着多边形依次处理顶点会遇到四种情况：

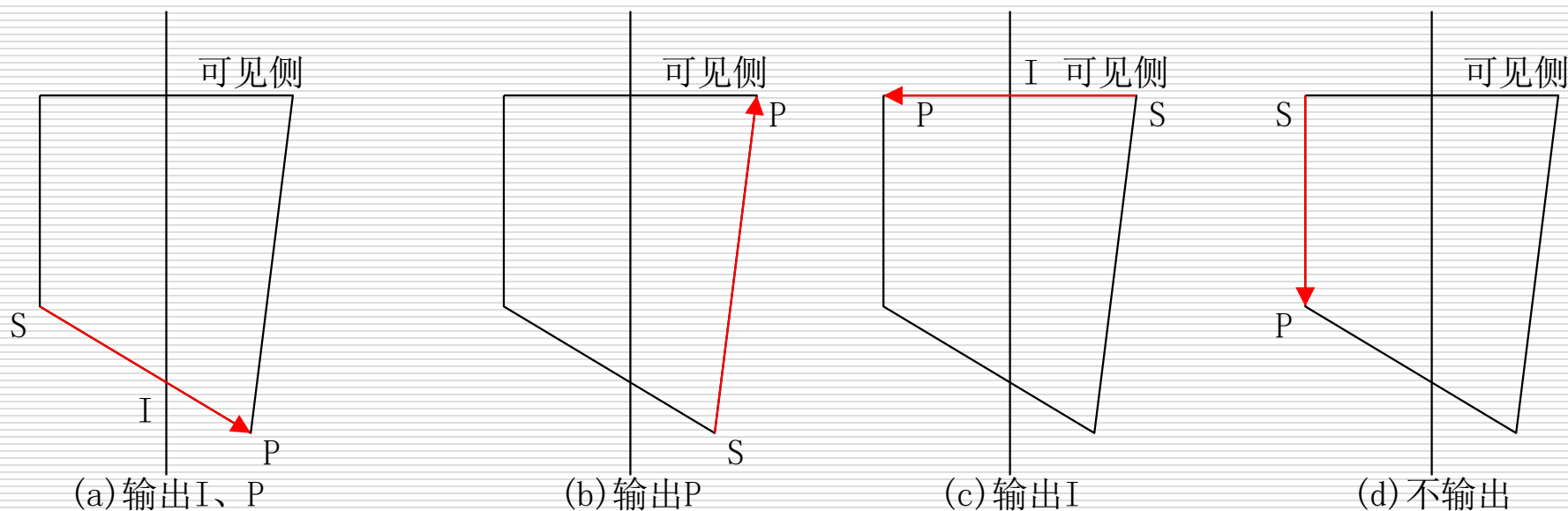
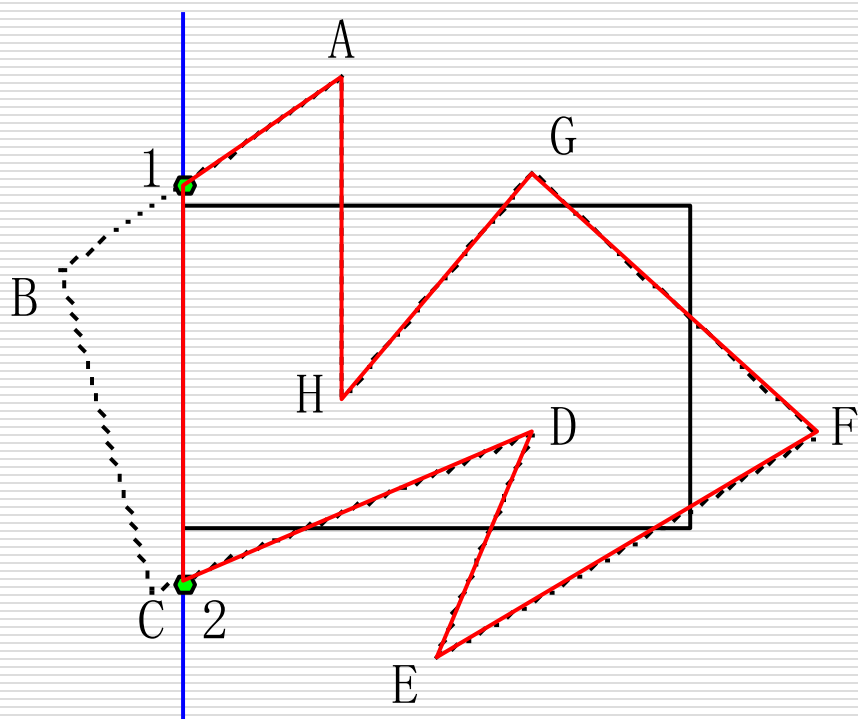
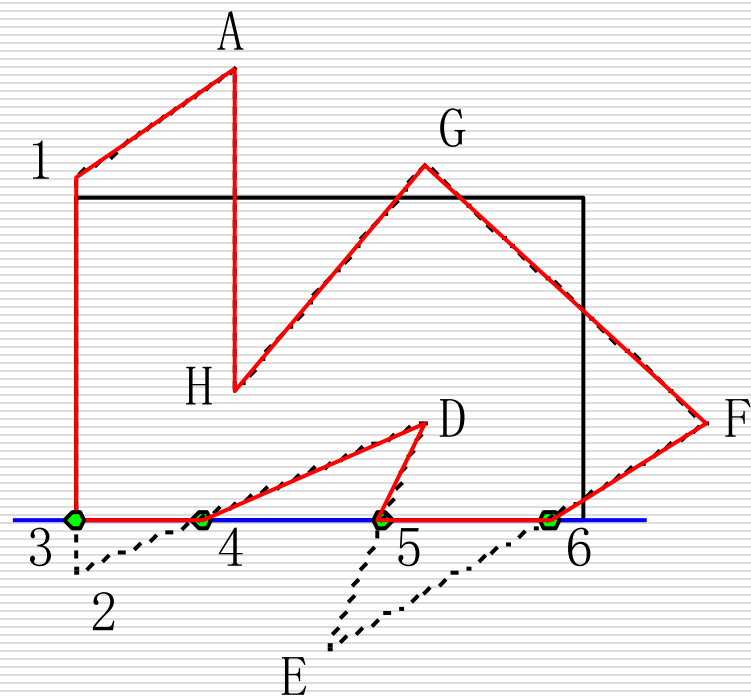


图6-19 多边形裁剪策略

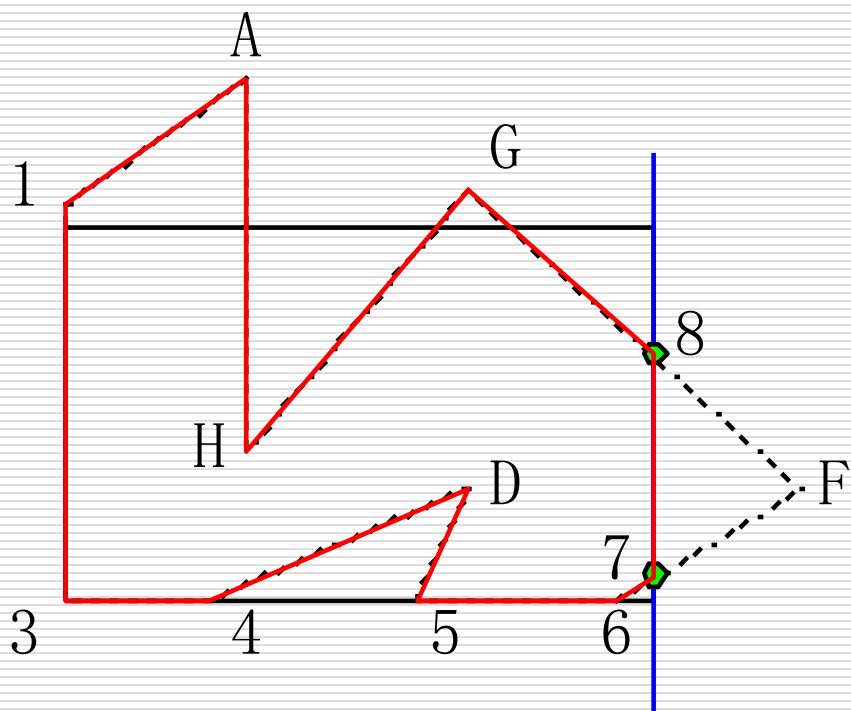


输入：ABCDEFGH  
 输出：12DEFGHA  
 (a) 用左边界裁剪



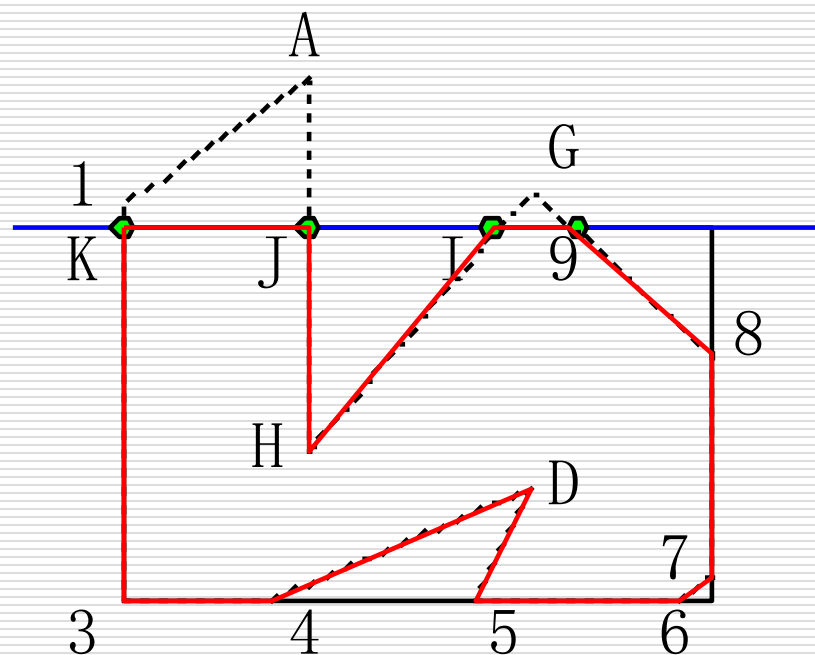
输入：12DEFGHA  
 输出：34D56FGHA1  
 (b) 用下边界裁剪

图6-20 多边形裁剪过程



输入：34D56FGHA1

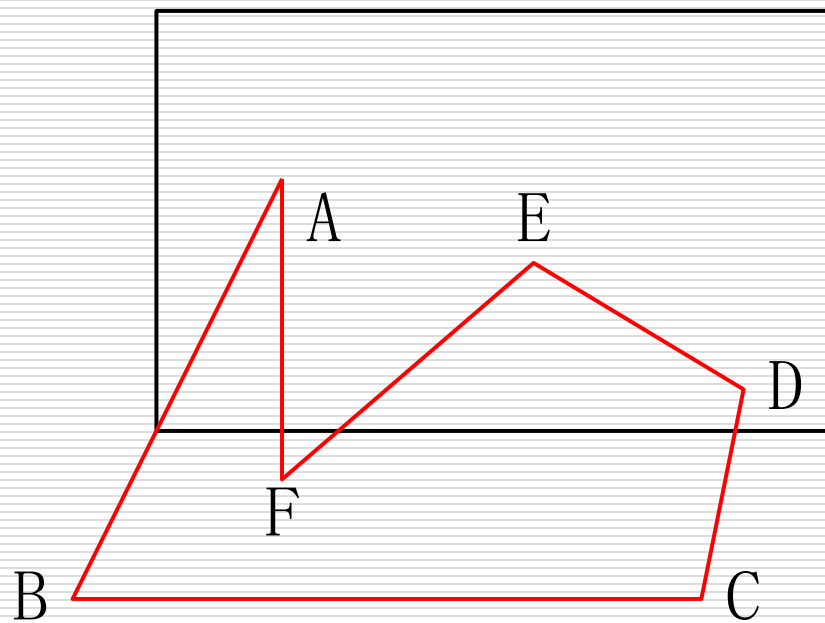
(c) 用右边界裁剪



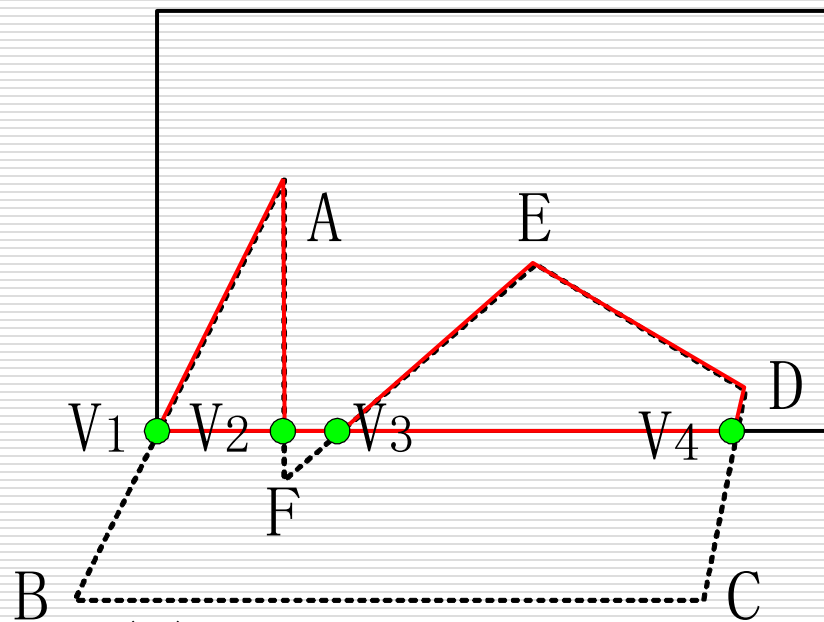
(d) 用上边界裁剪

图6-20 多边形裁剪过程

# 特点



(a) 裁剪前



(b) Sutherland-Hodgeman  
算法的裁剪结果

图6-21 凹多边形的裁剪

# Weiler-Atherton 多边形裁剪

- 假定按顺时针方向处理顶点，且将用户多边形定义为 $P_s$ ，窗口矩形为 $P_w$ 。算法从 $P_s$ 的任一点出发，跟踪检测 $P_s$ 的每一条边，当 $P_s$ 与 $P_w$ 相交时（实交点），按如下规则处理：

(1)若是由不可见侧进入可见侧，则输出可见直线段，转(3)；

# Weiler-Atherton 多边形裁剪

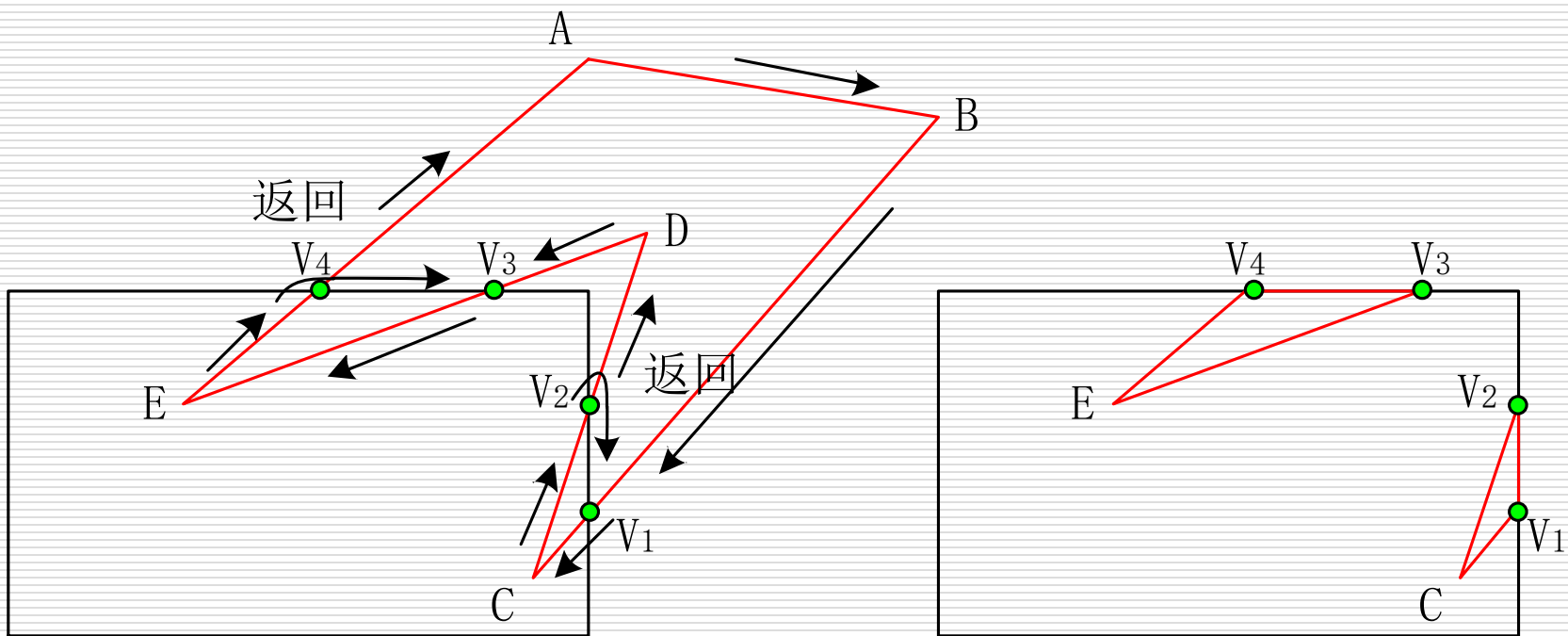
---

(2)若是由可见侧进入不可见侧，则从当前交点开始，沿窗口边界顺时针检测 $P_w$ 的边，即用窗口的有效边界去裁剪 $P_s$ 的边，找到 $P_s$ 与 $P_w$ 最靠近当前交点的另一交点，输出可见直线段和由当前交点到另一交点之间窗口边界上的线段，然后返回处理的当前交点；

(3)沿着 $P_s$ 处理各条边，直到处理完 $P_s$ 的每一条边，回到起点为止。



□ 下图示了Weiler-Atherton算法裁剪凹多边形的过程和结果。



(a) 裁剪前

(b) Weiler-Atherton算法的裁剪结果

图6-22 Weiler-Atherton算法裁剪凹多边形

# 其他裁剪

---

## 2. 文字裁剪

文字裁剪的策略包括几种：

- 串精度裁剪
- 字符精度裁剪
- 笔划、像素精度裁剪

## 3. 外部裁剪

保留落在裁剪区域外的图形部分、去掉裁剪区域内的所有图形，这种裁剪过程称为外部裁剪，也称空白裁剪。

## 6.2 三维观察变换

---

- 观察坐标系
- 观察空间
- 三维观察流程
- 三维裁剪

# 观察坐标系

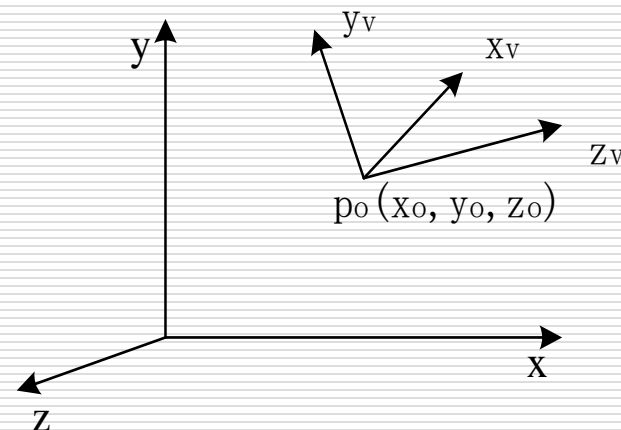


图6-23 用户坐标系与观察坐标系

- 观察参考坐标系 (View Reference Coordinate)
- 观察参考点 (View Reference Point)

# 观察坐标系

□ 观察平面（View Plane），即投影平面。

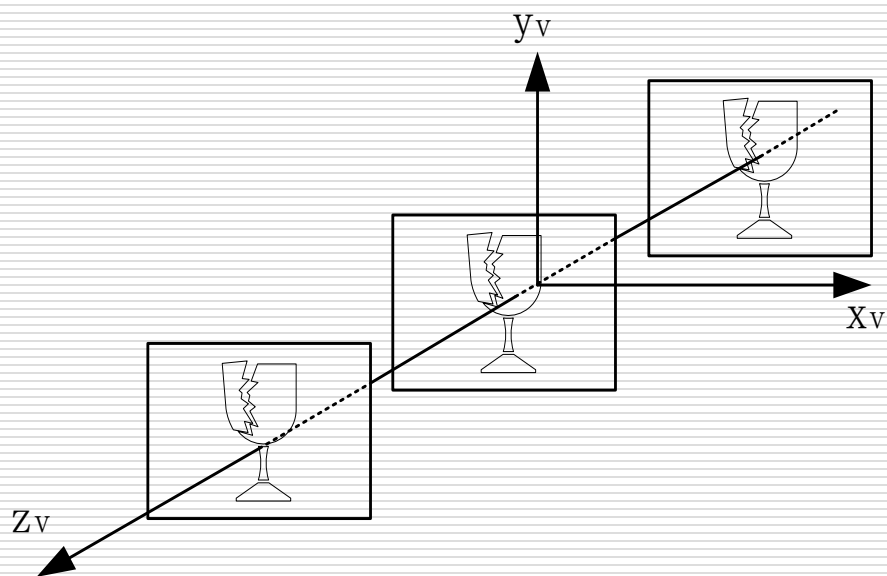


图6-24 沿 $z_v$ 轴的观察平面

# 观察坐标系

## □ 观察坐标系（ $uvn$ 坐标系）的建立

### （1）法矢量 $N$

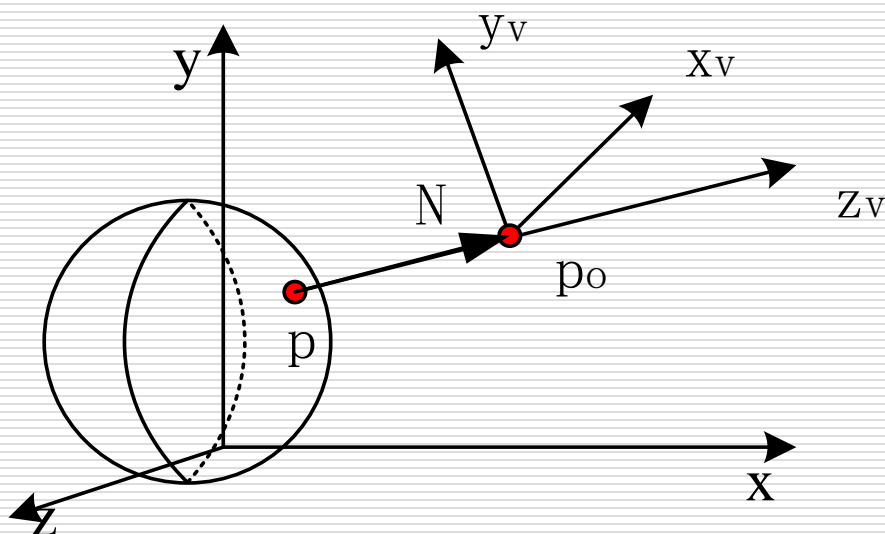


图6-25 法矢量 $N$ 的定义

# 观察坐标系

## (2) 法矢量 $V$ 的建立

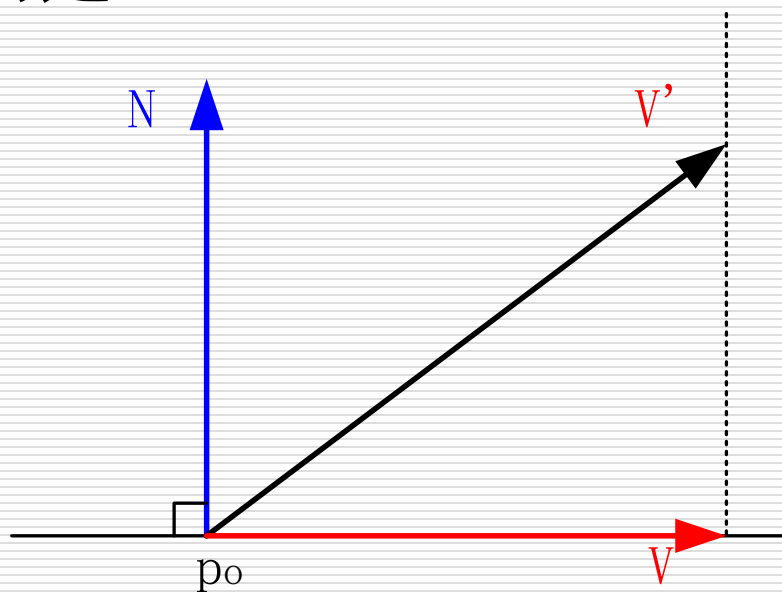


图6-26 法矢量 $V$ 的定义

## (3) 法矢量 $U$

# 观察坐标系

□ 通过改变观察参考点的位置或改变 $N$ 的方向可以使用户在不同的距离和角度上观察三维形体。

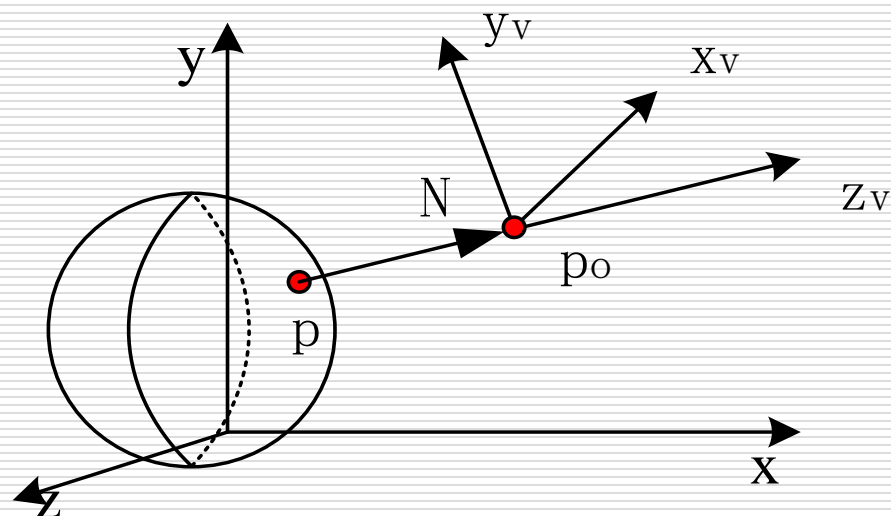


图6-27 三维观察



# 观察空间

## □ 观察窗口

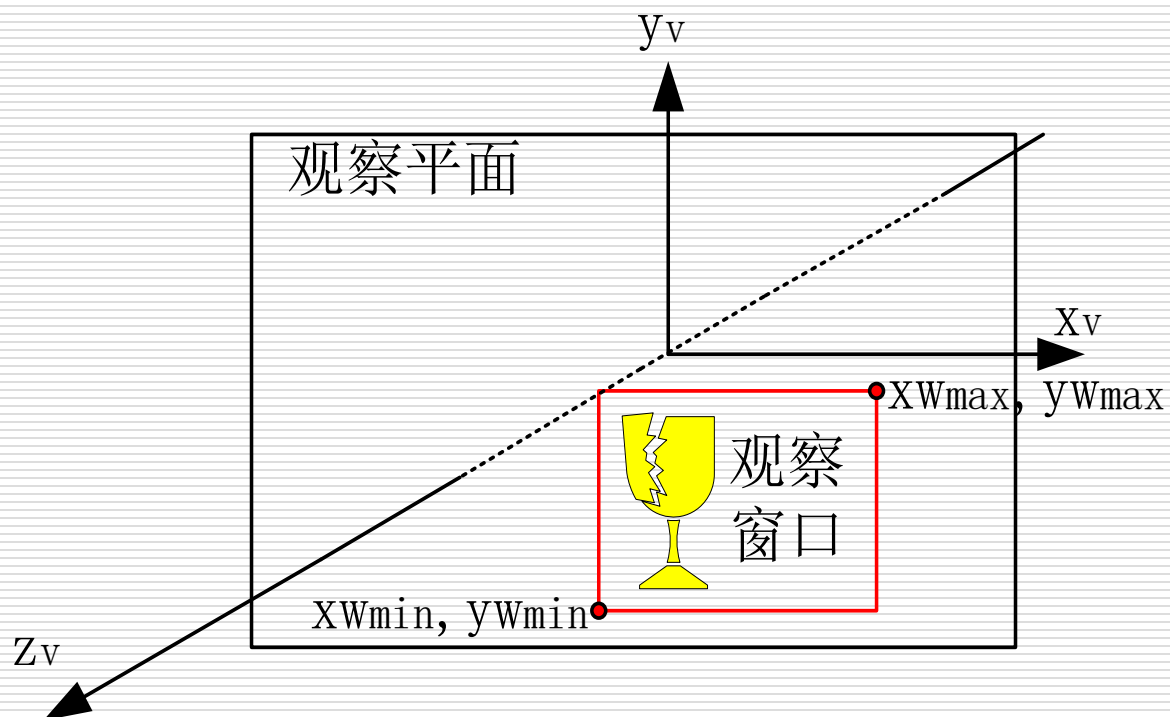


图6-28 观察窗口

# 观察空间

- 观察空间：将观察窗口沿投影方向作平移运动产生的三维形体。
- 观察空间的大小和形状依赖于窗口的大小及投影类型。

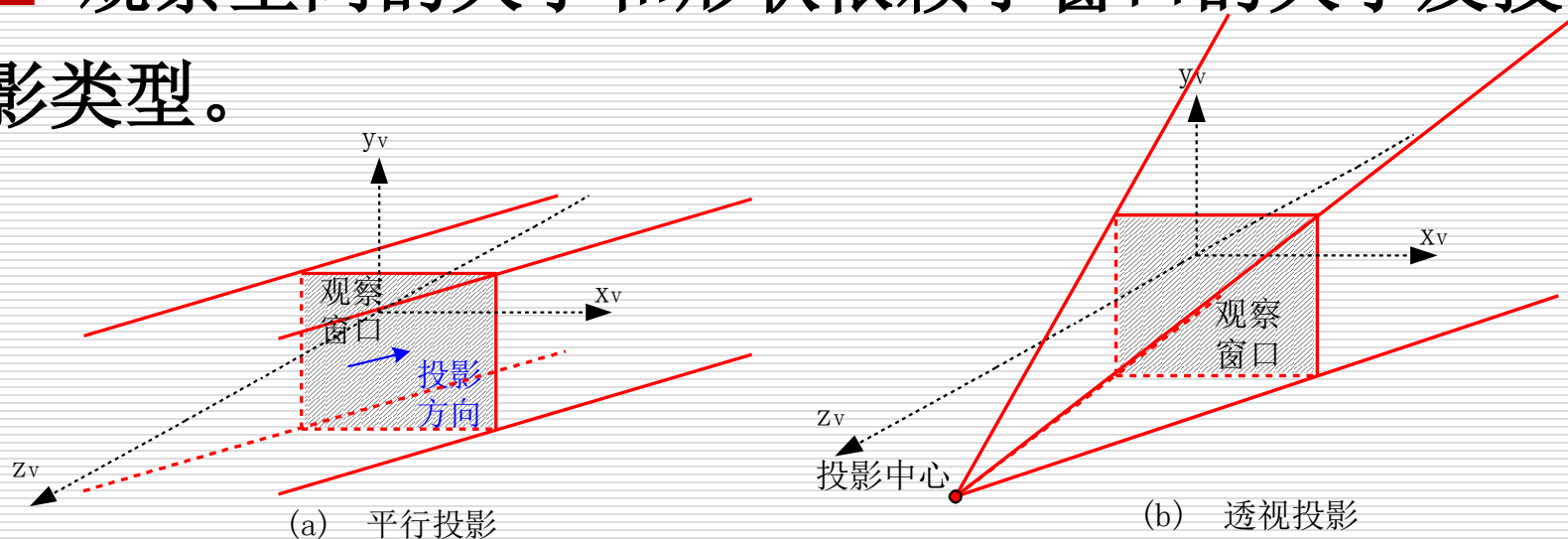


图6-29 观察空间

# 观察空间

- 无限观察空间、有限观察空间
- 前后截面:  $Z=Z_{\text{front}}$ ,  $Z=Z_{\text{back}}$

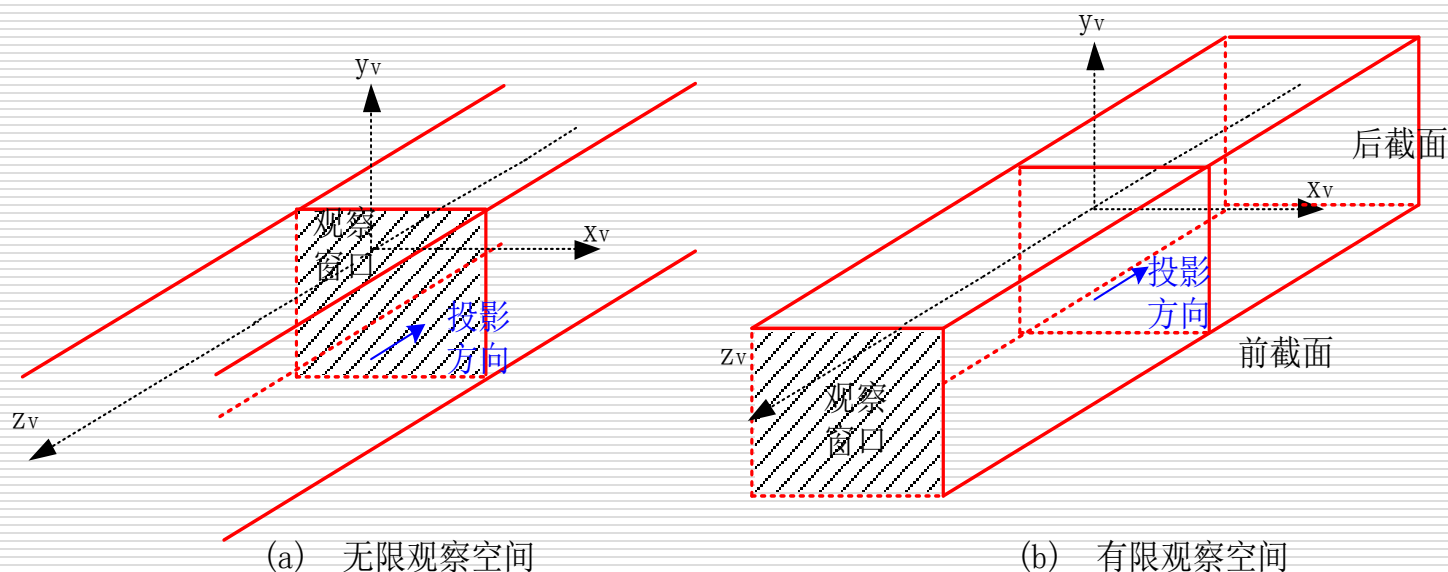


图6-30 正投影的观察空间

# 观察空间

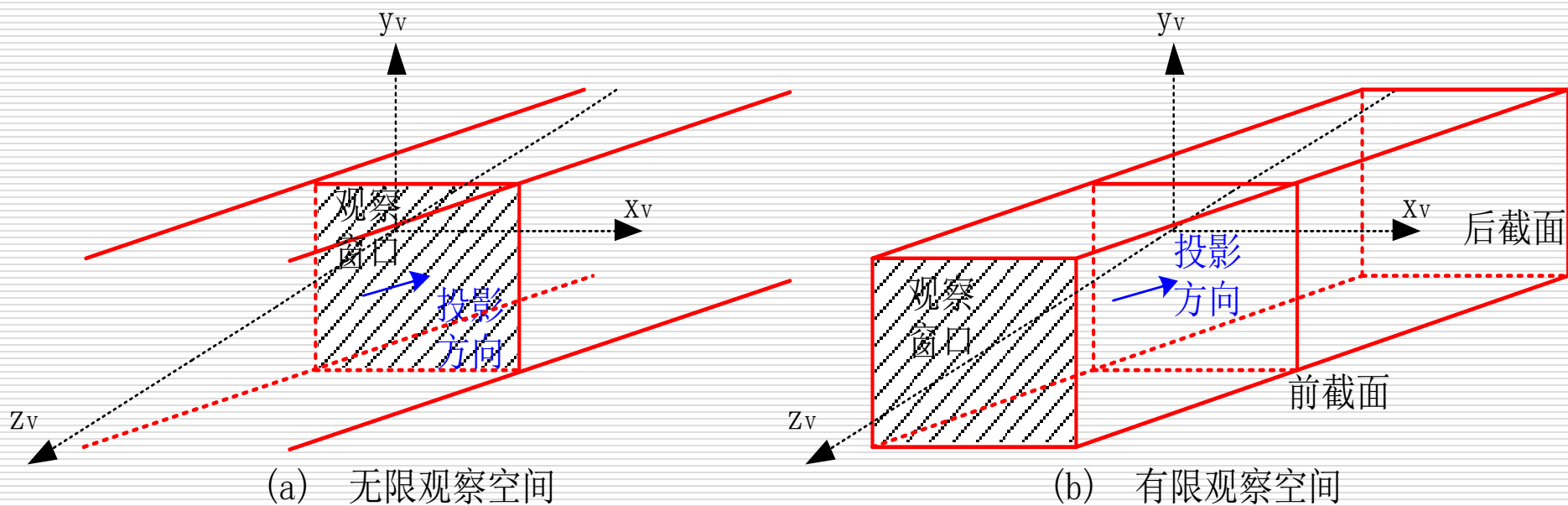


图6-31 斜投影的观察空间

# 观察空间

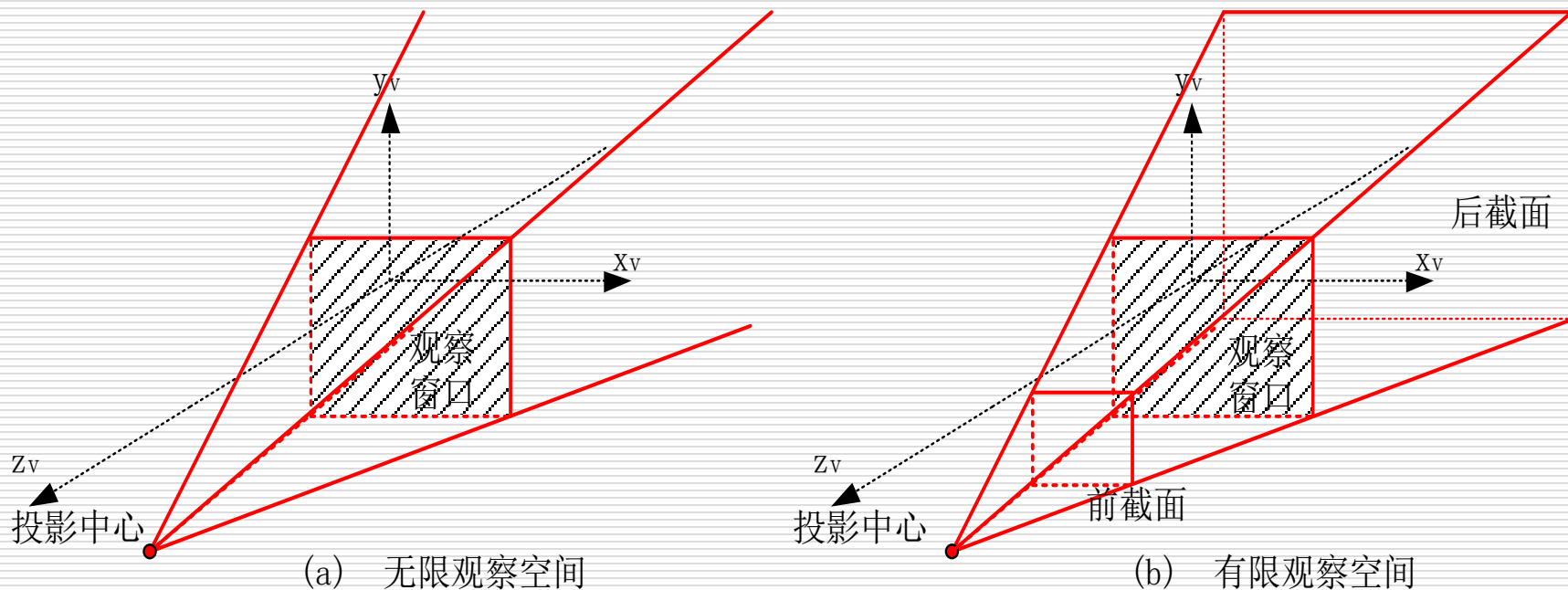


图6-32 透视投影的观察空间

□ 需注意，对于透视投影，前截面必须在投影中心和后截面之间。

# 观察空间

□ 观察平面和前后截面的有关位置取决于要生成的窗口类型及特殊图形包的限制。



图6-33 观察平面及前后截面的位置安排

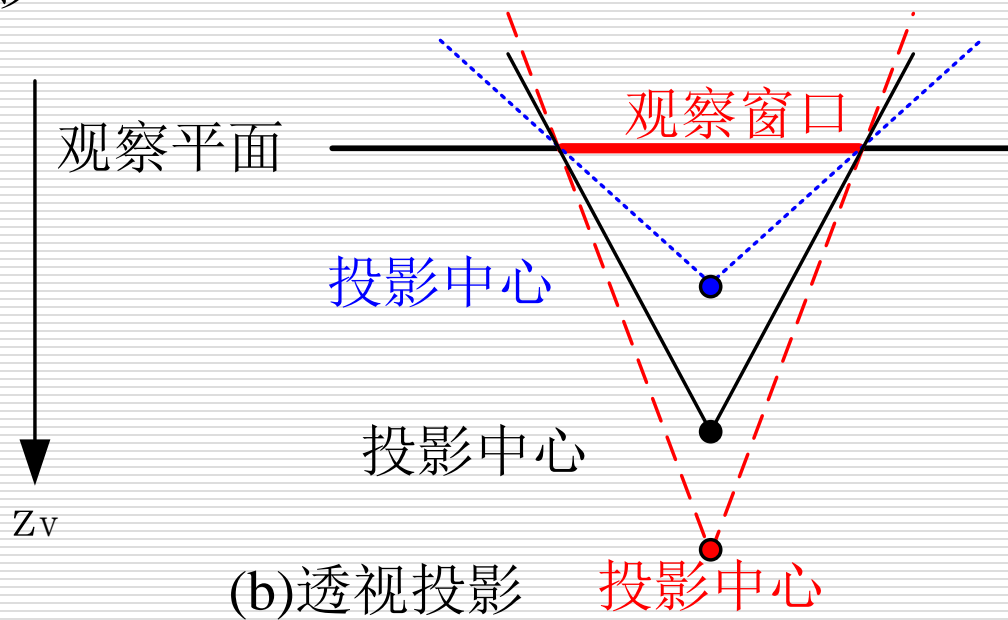
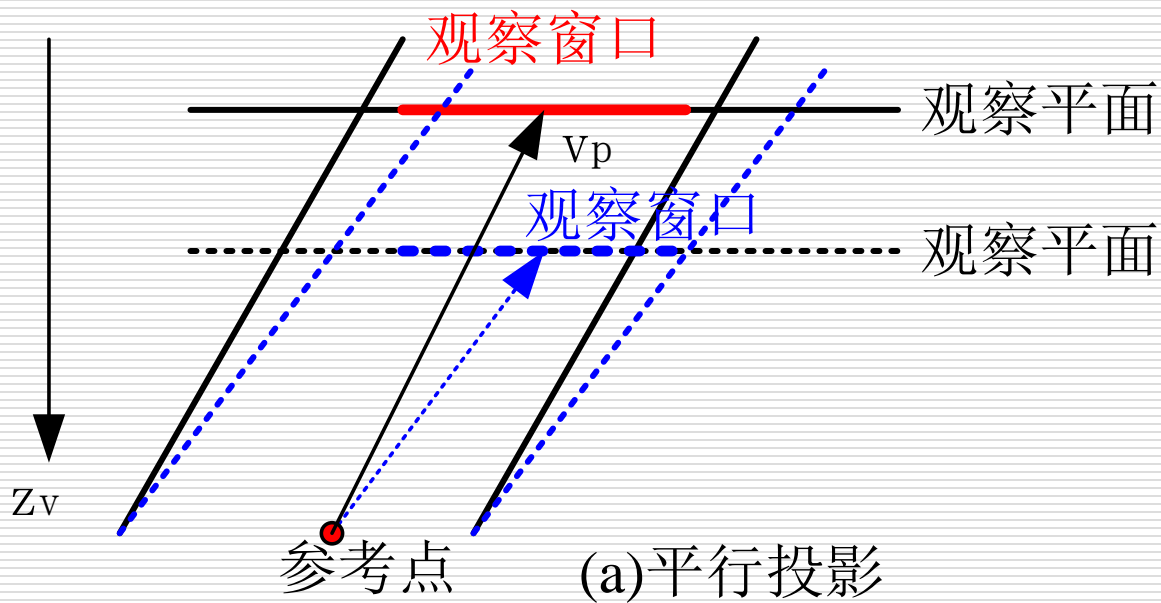


图6-34 观察空间形状的变化

# 观察空间

## □ 规范化观察空间

■ 平行投影的规范化观察空间定义为：

$$x_v = 1, x_v = -1$$

$$y_v = 1, y_v = -1$$

$$z_v = 0, z_v = 1$$

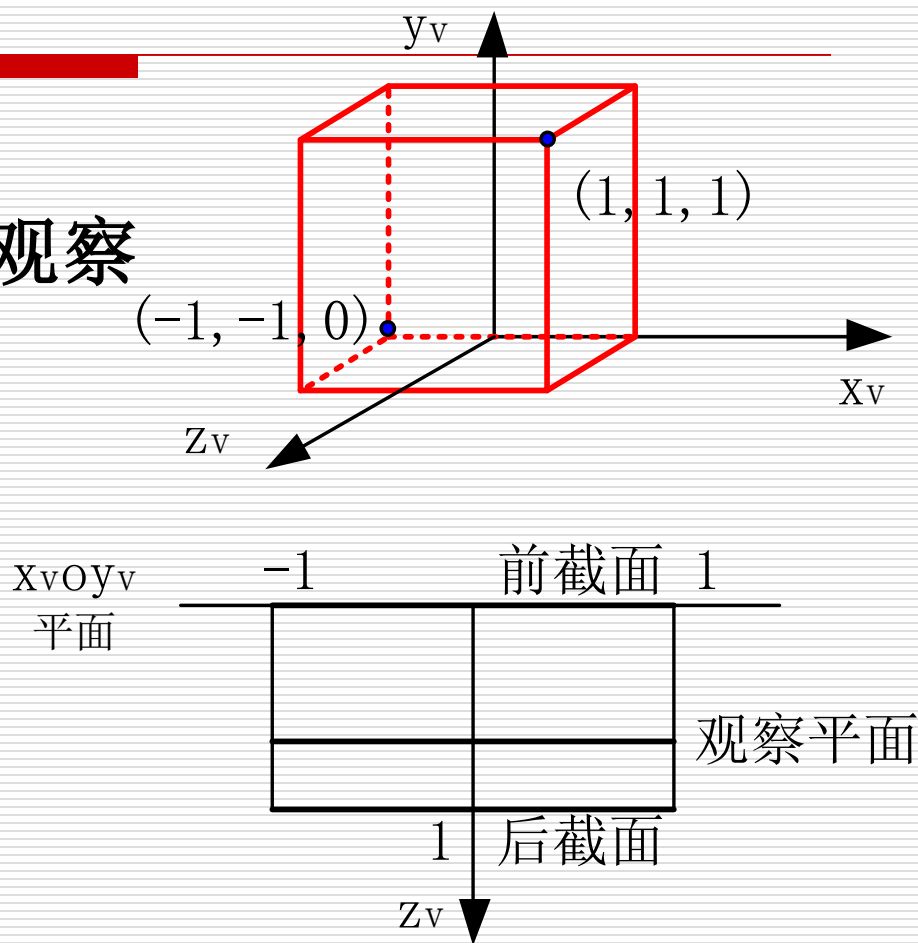


图6-35 平行投影的规范化观察空间



# 观察空间

□ 透视投影的规范化观察空间为：

$$x_v = z_v, x_v = -z_v$$

$$y_v = z_v, y_v = -z_v$$

$$z_v = z_{\min}, z_v = 1$$

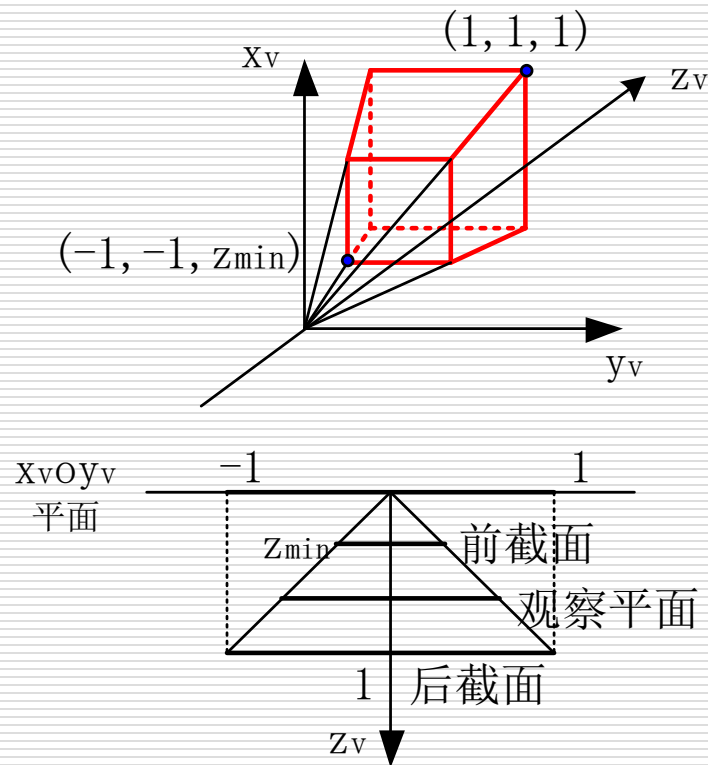


图6-36 透视投影的规范化观察空间

# 三维观察流程

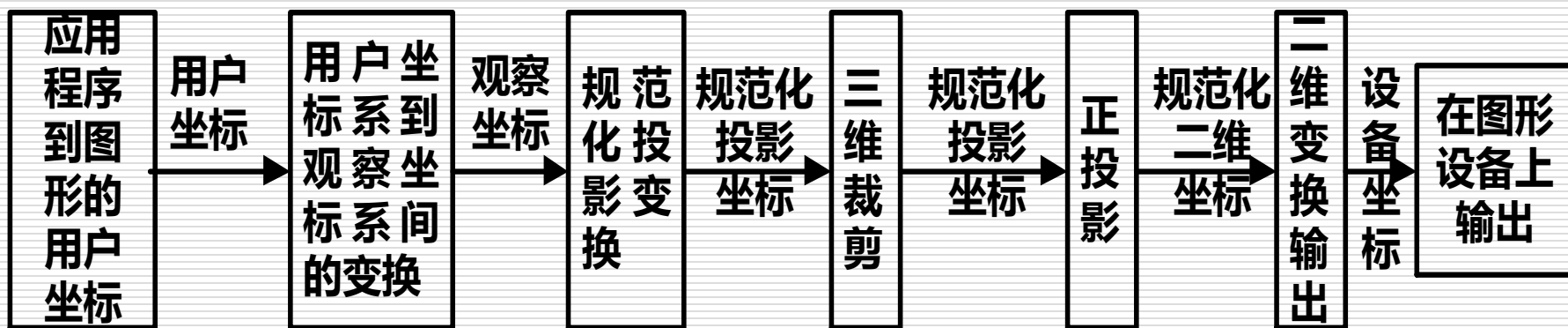


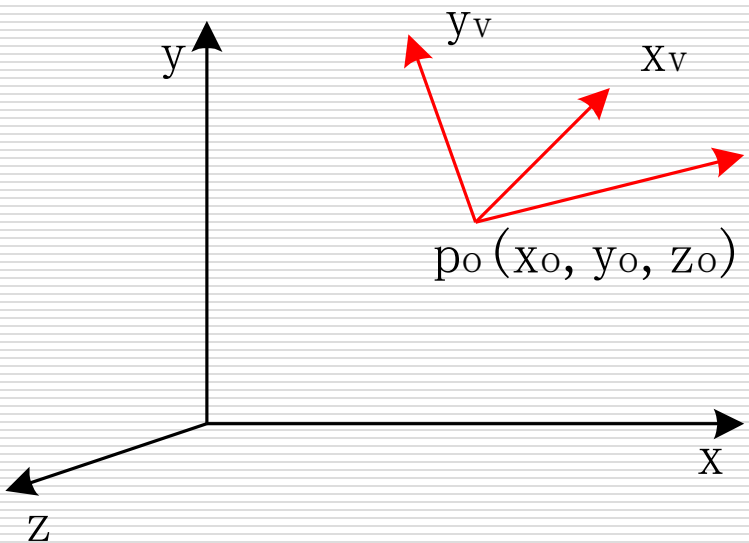
图6-37 三维观察流程

# 用户坐标系到观察坐标系变换

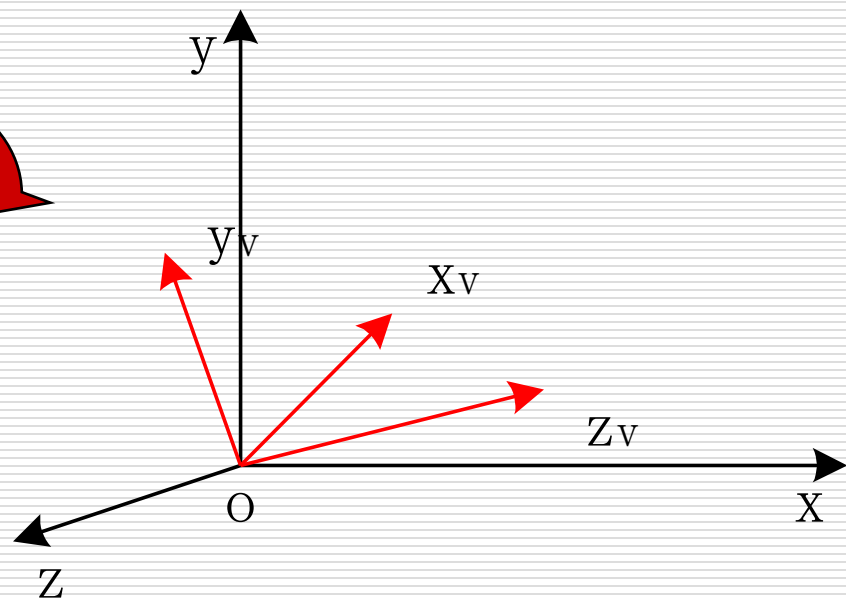
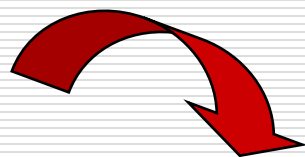
---

具体变换步骤:

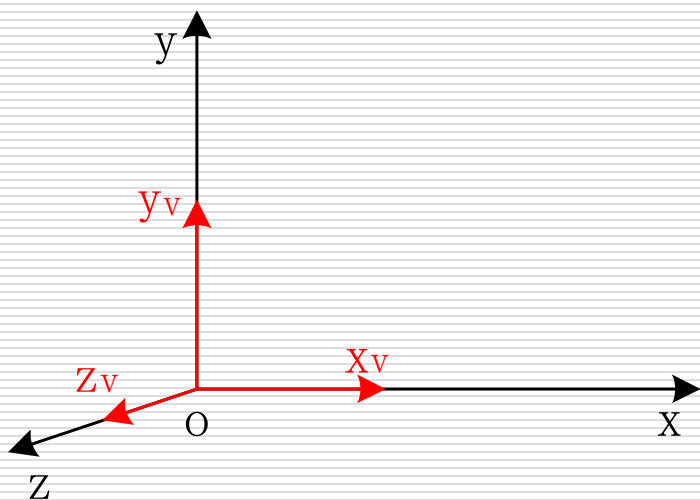
- (1) 平移观察参考点到用户坐标系原点;
- (2) 进行旋转变换分别让 $x_v$ 、 $y_v$ 和 $z_v$ 轴对应到用户坐标系中的 $x$ 、 $y$ 和 $z$ 轴。



(a) 用户坐标系与观察坐标系



(b) 平移观察坐标系



(c) 旋转观察坐标系

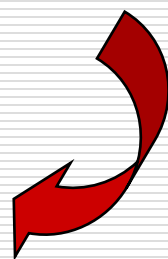


图6-38 用户坐标系到观察坐标系的变换

# 平行投影的规范化投影变换

分析:

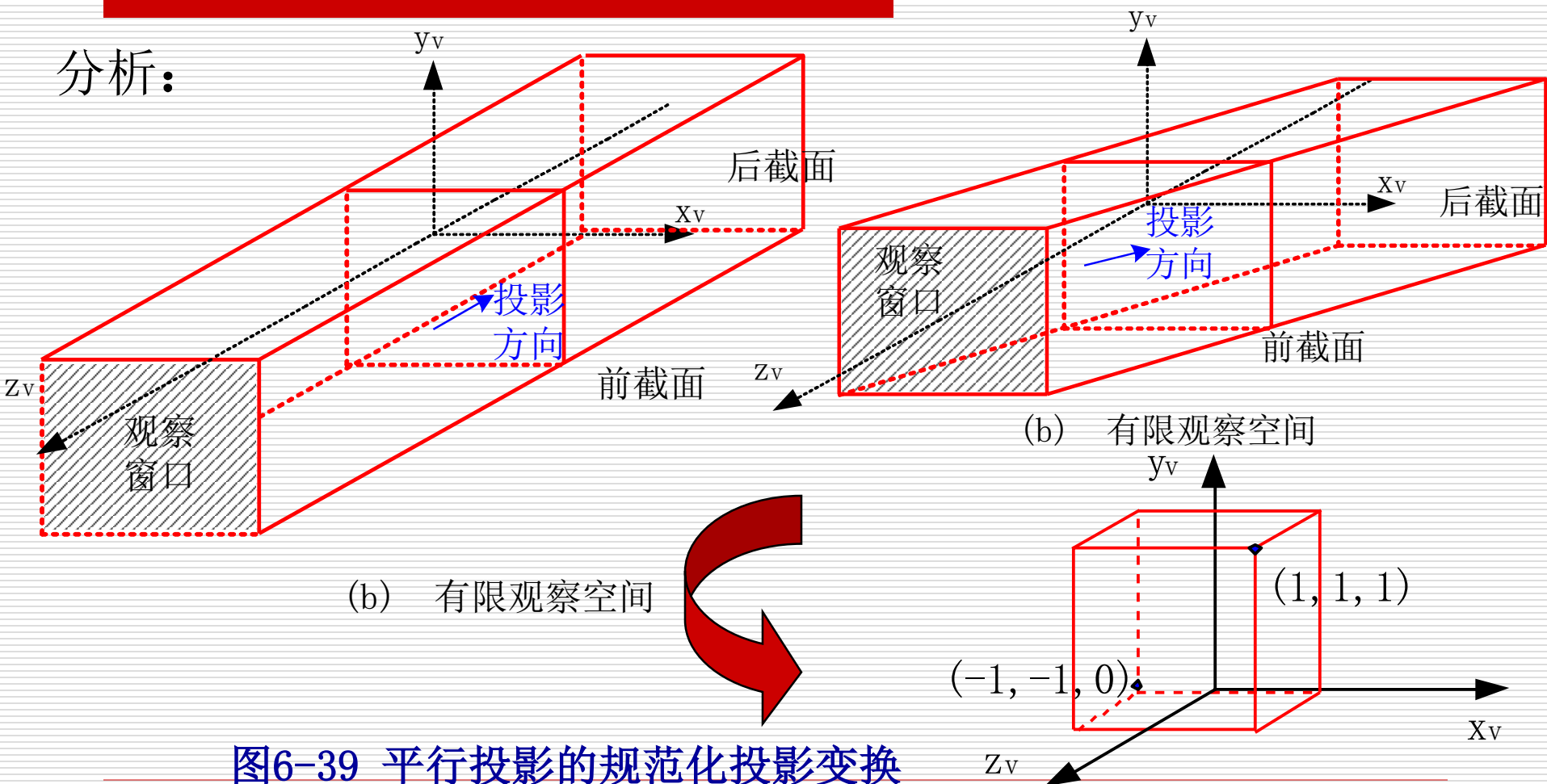


图6-39 平行投影的规范化投影变换

# 平行投影的规范化投影变换

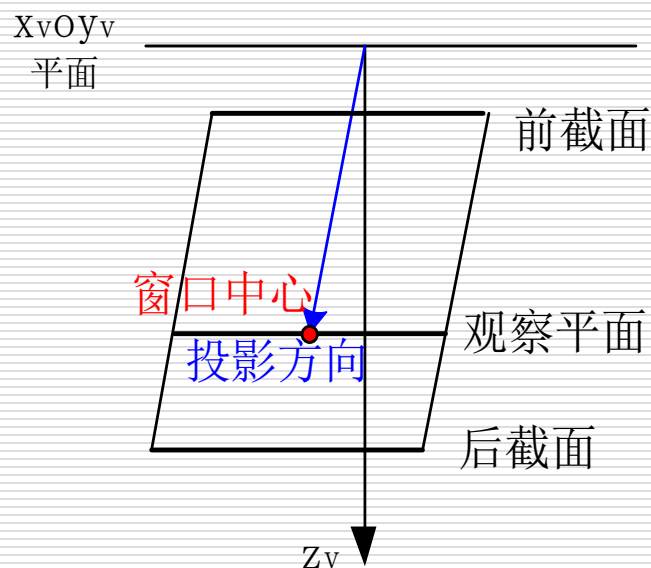
---

- 观察窗口：左下角点 ( $xw_{min}, yw_{min}$ )  
                    右上角点 ( $xw_{max}, yw_{max}$ )
- 参考点：( $x_{prp}, y_{prp}, z_{prp}$ )
- 前后截面： $Z=Z_{front}$ ,  $Z=Z_{back}$
- 观察平面： $Z=Z_{vp}$
- 投影方向为从参考点到观察窗口中心点的坐标矢量。

# 平行投影的规范化投影变换

□ 平行投影的规范化投影变换可由以下三步组成。

(1) 将参考点平移到观察坐标系原点；



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_{prp} & -y_{prp} & -z_{prp} & 1 \end{bmatrix}$$

图6-40 平移变换

# 平行投影的规范化投影变换

(2)对坐标系进行错切变换，使参考点和窗口中心的连线错切到 $z_v$ 轴；

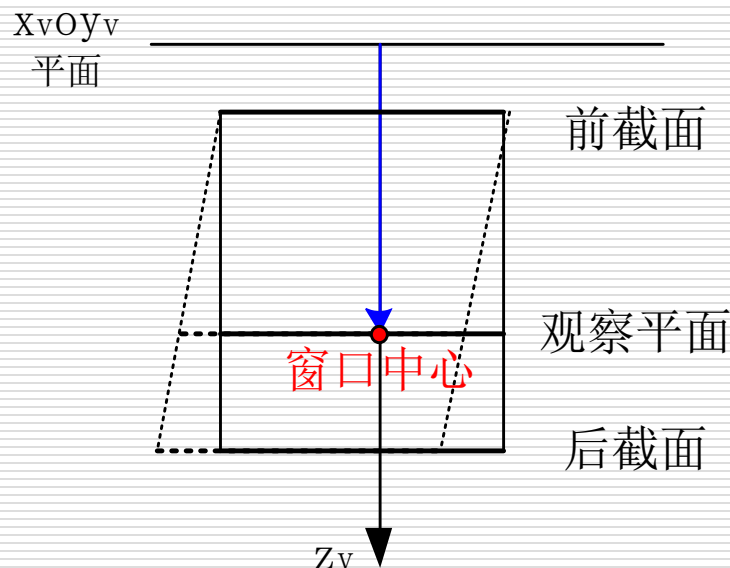


图6-41 错切变换



# 平行投影的规范化投影变换

□ 平移变换后，窗口中心点的坐标

$$\begin{cases} x_{cw} = (xw_{\min} + xw_{\max}) / 2 - x_{prp} \\ y_{cw} = (yw_{\min} + yw_{\max}) / 2 - y_{prp} \\ z_{cw} = z_{vp} - z_{prp} \end{cases}$$

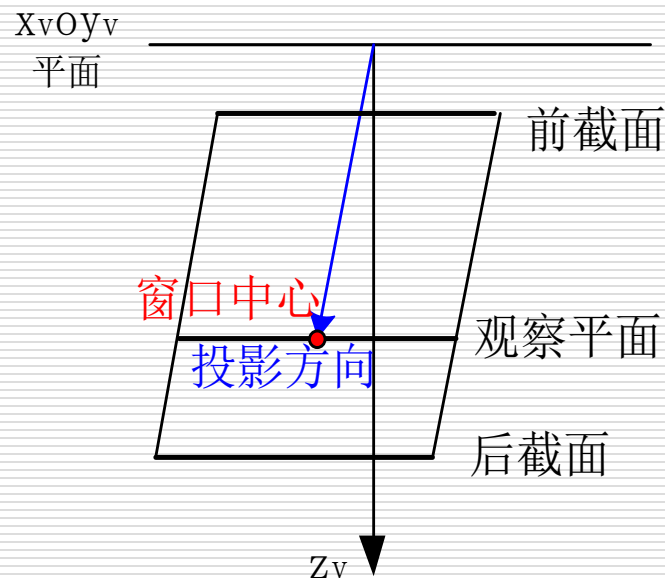


图6-42 平移变换

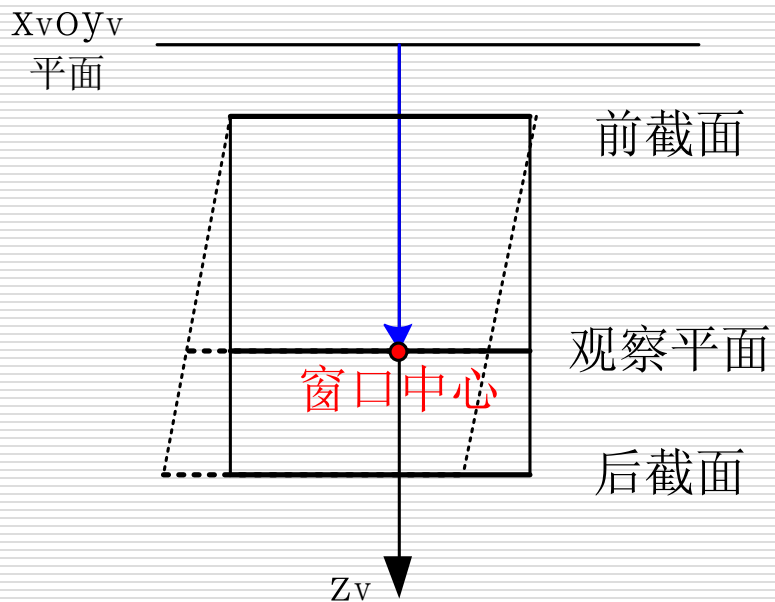


图6-43 错切变换

解得：

$$a = -x_{cw} / z_{cw}$$

$$b = -y_{cw} / z_{cw}$$

$$\begin{bmatrix} 0 & 0 & z_{cw} & 1 \end{bmatrix} = \begin{bmatrix} x_{cw} & y_{cw} & z_{cw} & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & b & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# 平行投影的规范化投影变换

(3)进行坐标的归一化变换;

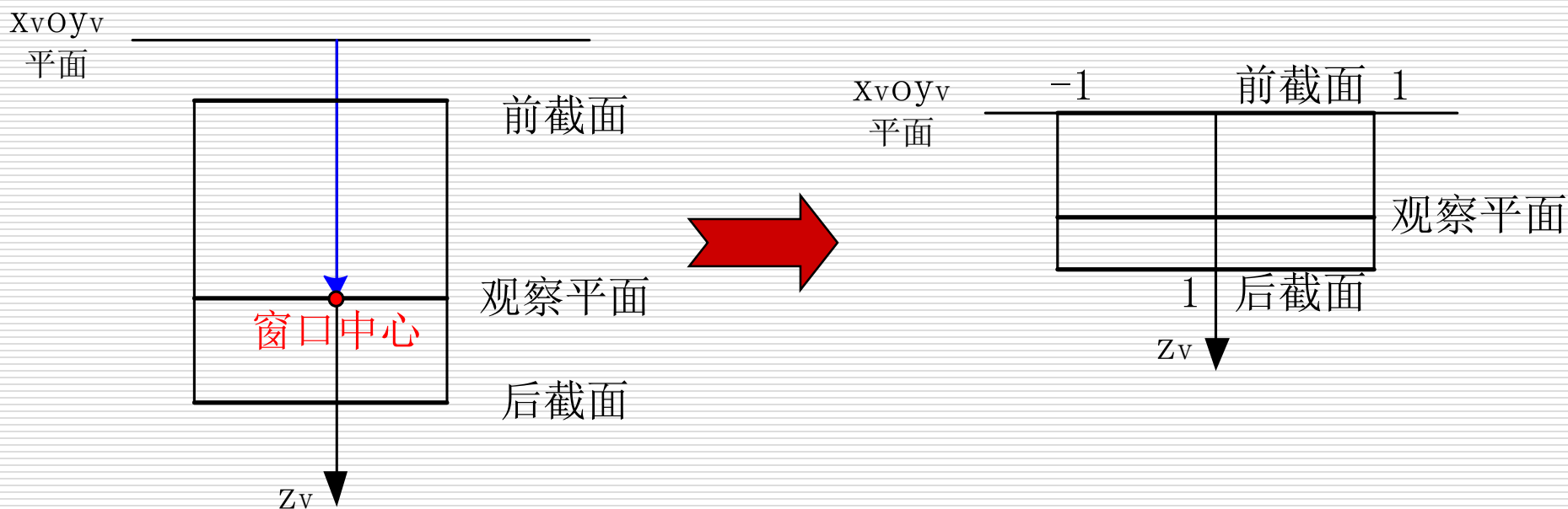
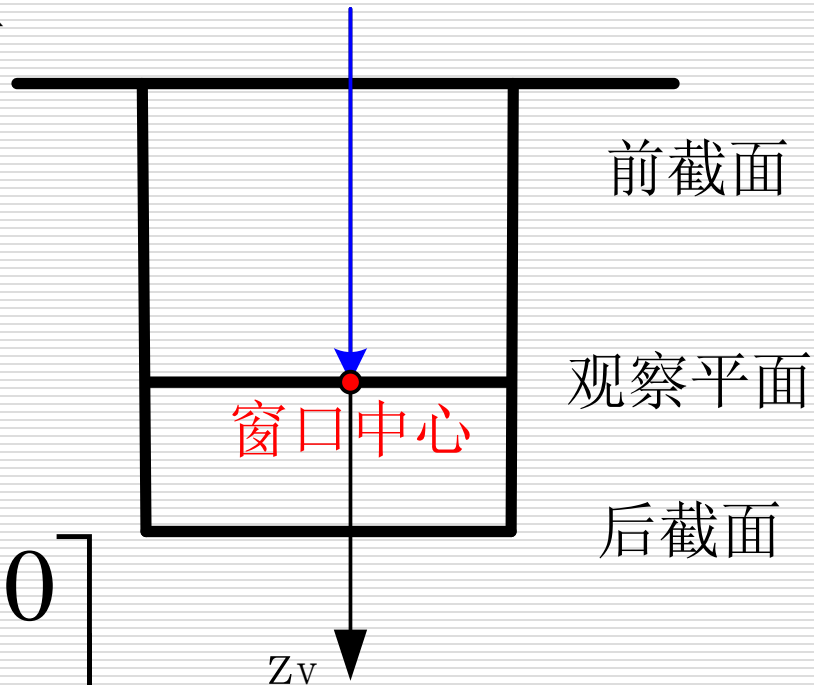


图6-44 比例变换

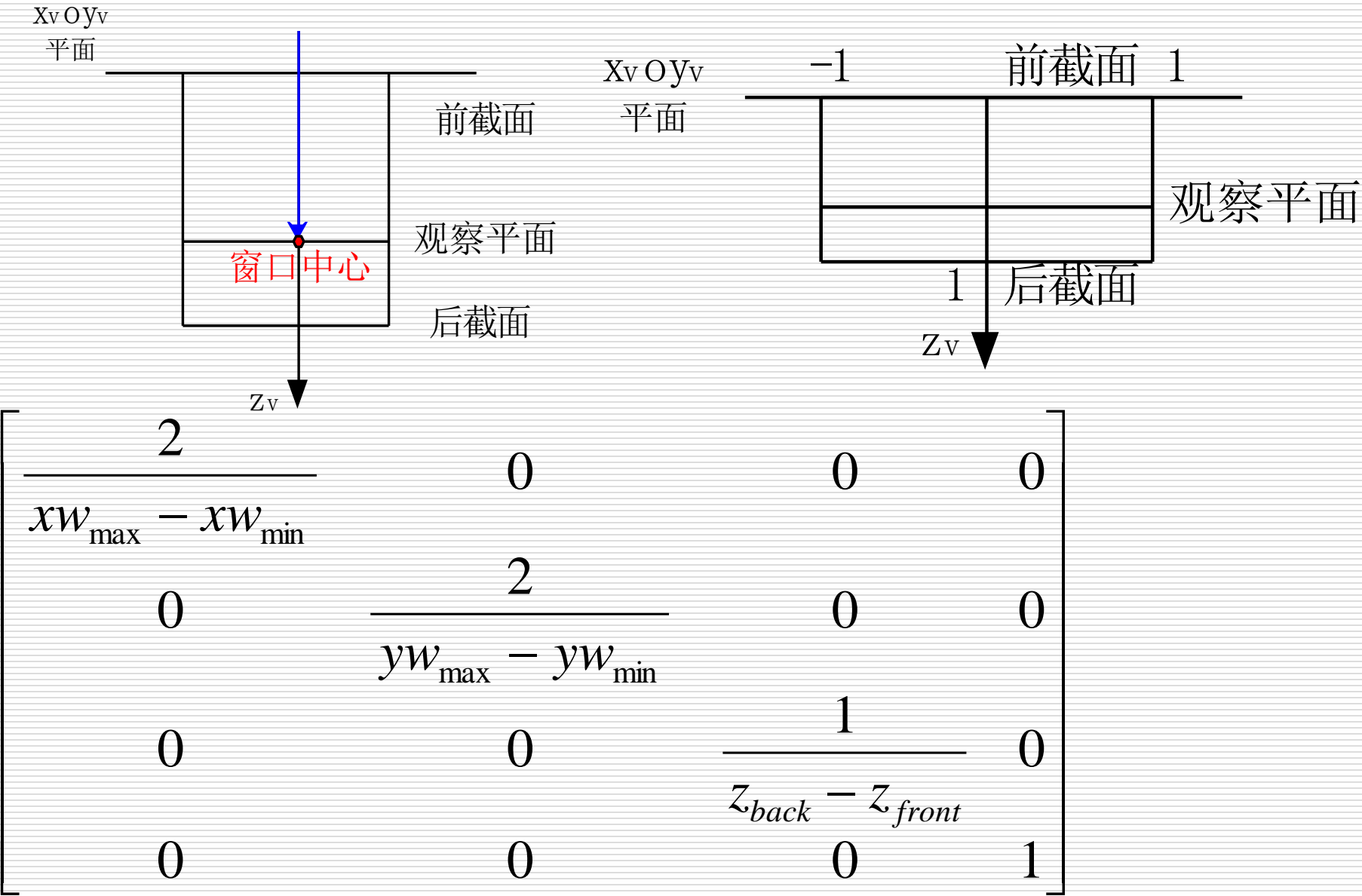
$x_v O y_v$   
平面

## a) 平移变换

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -(z_{front} - z_{prp}) & 1 \end{bmatrix}$$



b) 比例变换



# 透视投影的规范化投影变换

分析：透视投影的规范化投影变换分两步进行  
(1)

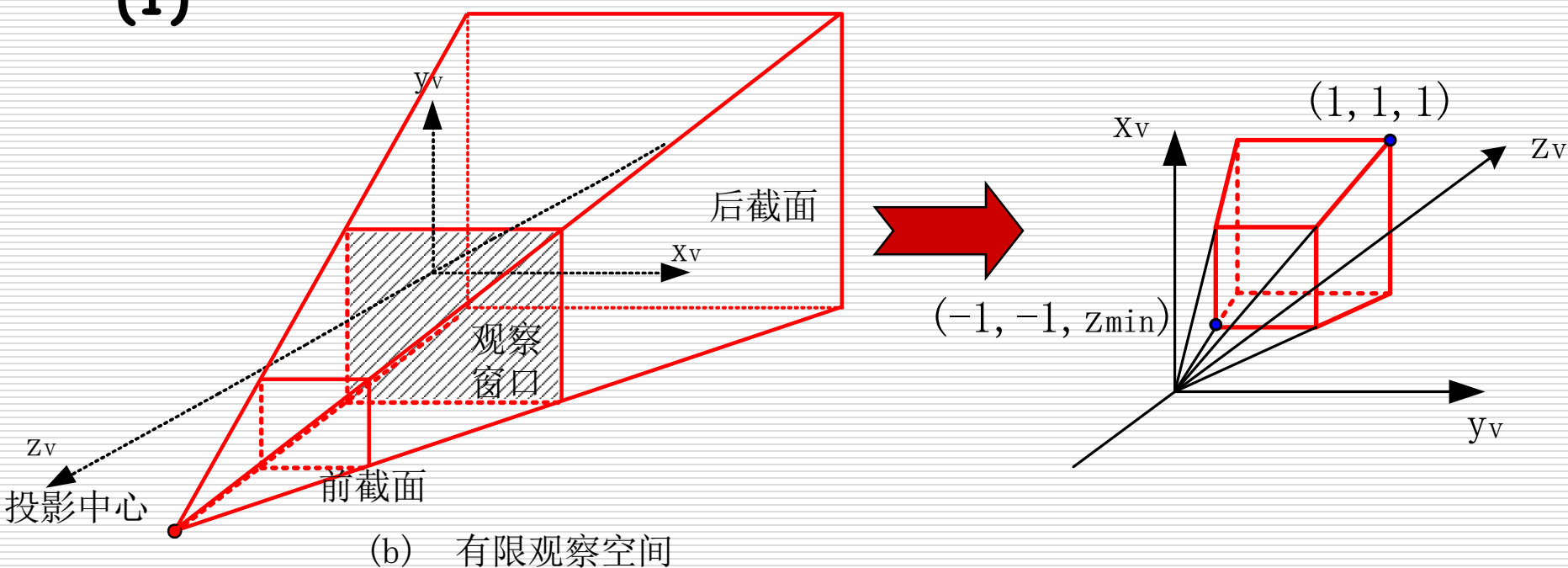


图6-45 透视投影的规范化投影变换

(2)

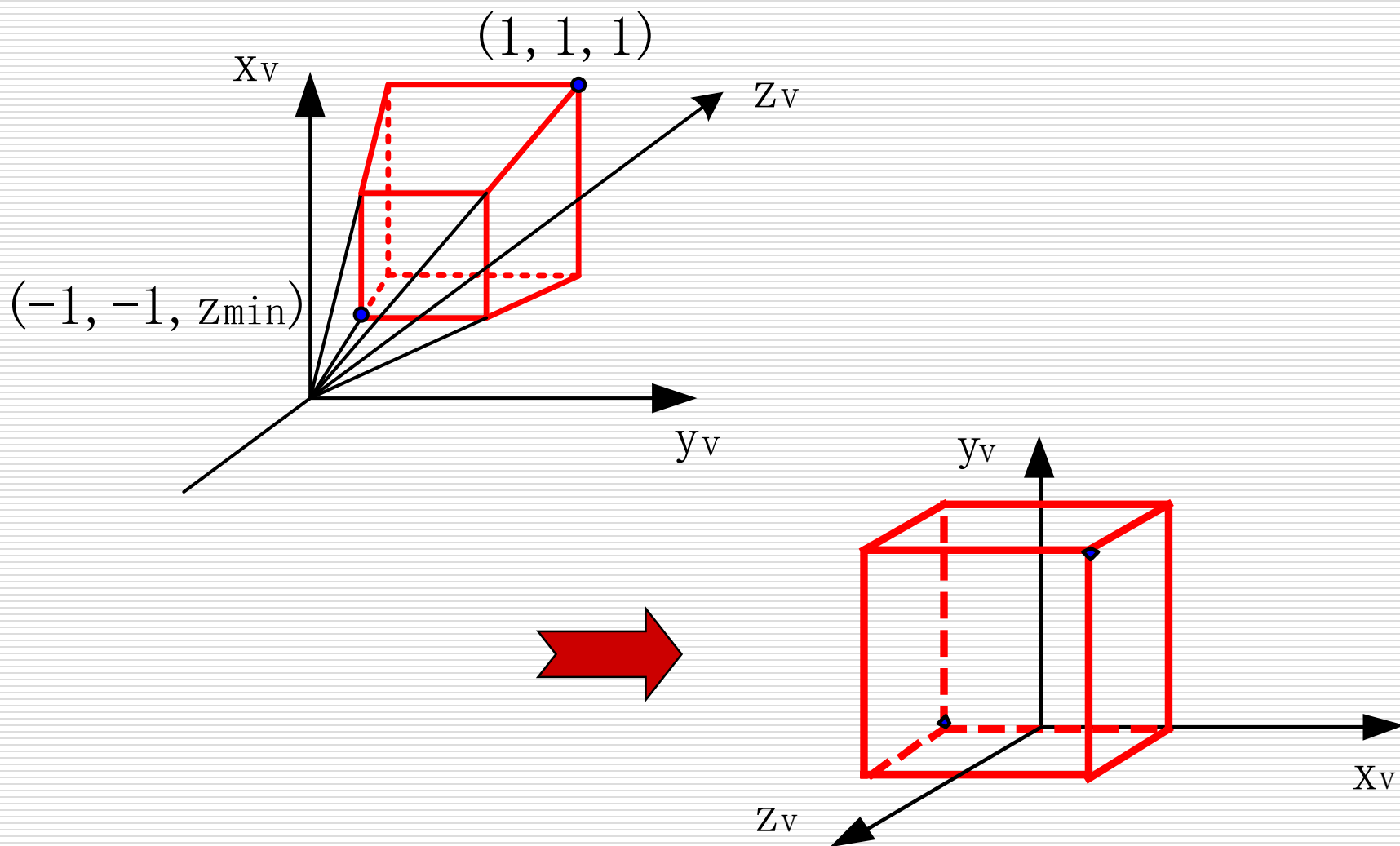


图6-46 透视投影的规范化投影变换II

# 透视投影的规范化投影变换

---

- 观察窗口：左下角点 ( $xw_{\min}, yw_{\min}, 0$ )  
                    右上角点 ( $xw_{\max}, yw_{\max}, 0$ )
- 参考点：( $x_{\text{prp}}, y_{\text{prp}}, z_{\text{prp}}$ )
- 前后截面： $Z=Z_{\text{front}}, Z=Z_{\text{back}}$
- 观察平面： $Z=Z_{\text{vp}}$



变换步骤:

**(1)**将投影中心平移到观察坐标系原点

**(2)**对坐标系进行错切变换

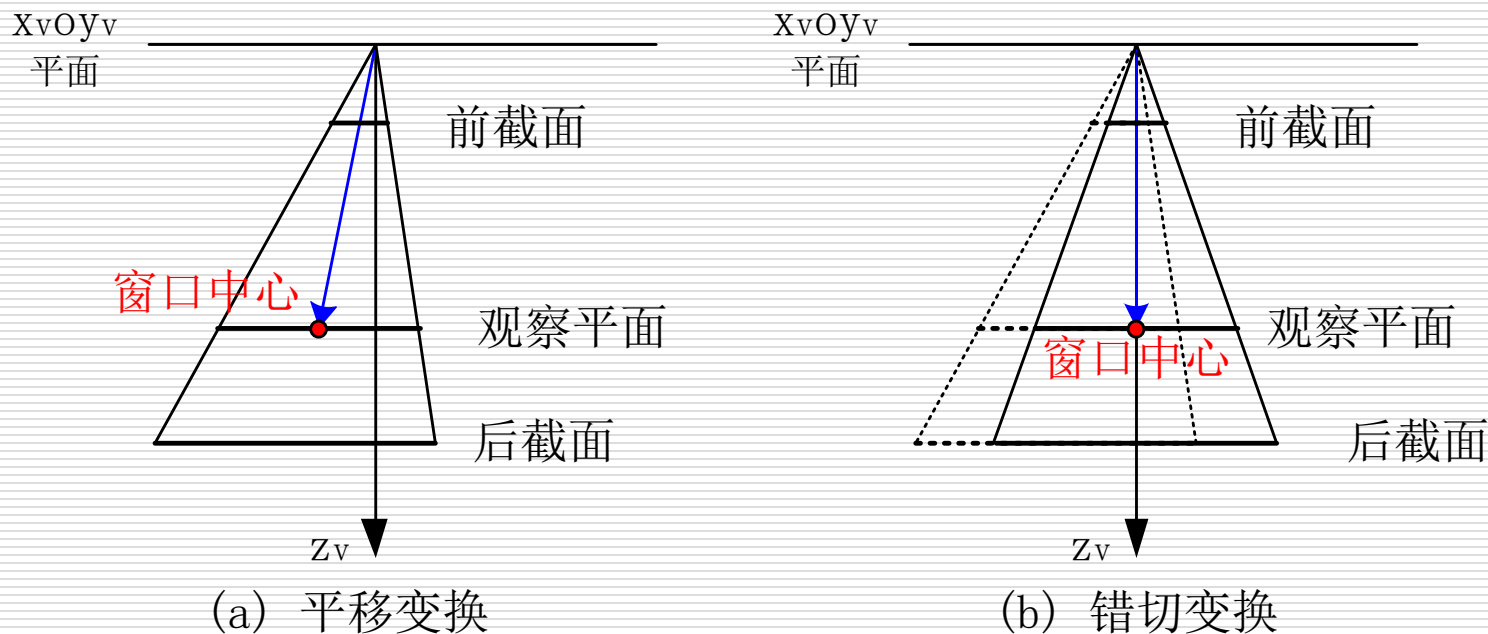


图6-47 透视投影的规范化投影变换步骤I

# 透视投影的规范化投影变换

---

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{x_{cw}}{z_{cw}} & -\frac{y_{cw}}{z_{cw}} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# 透视投影的规范化投影变换

(3) 进行比例变换。

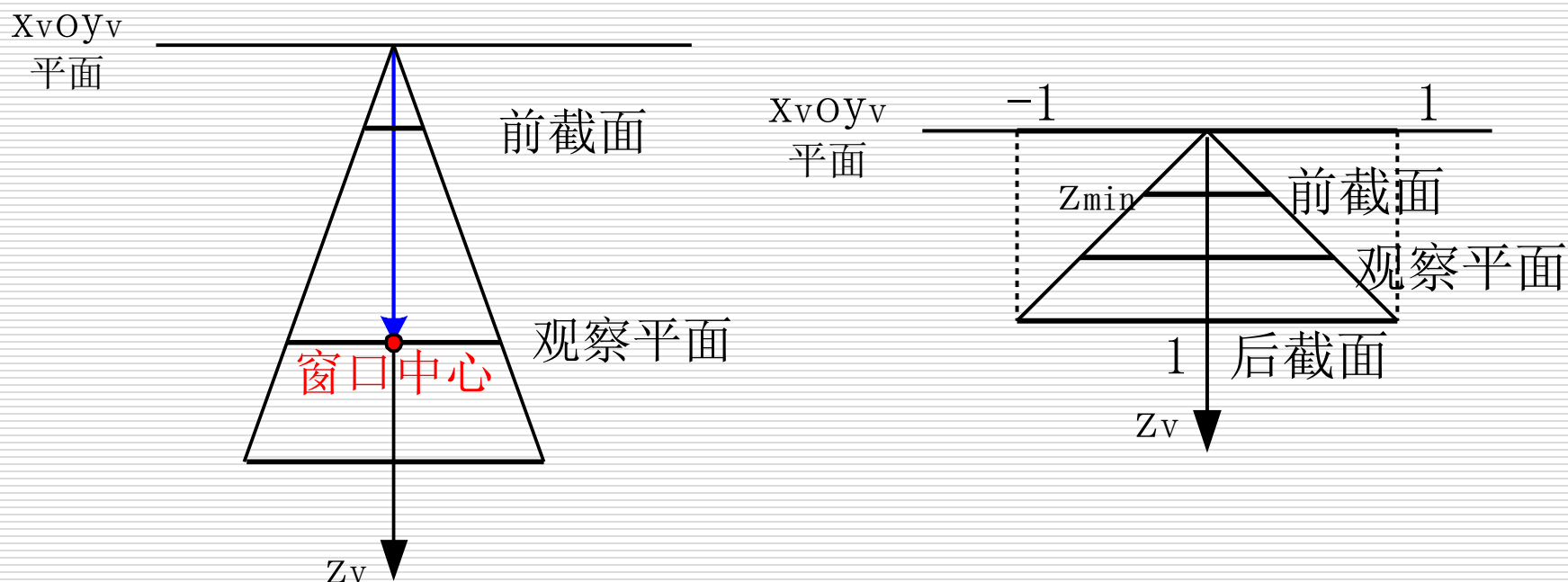


图6-48 透视投影的规范化投影变换II

$$\begin{bmatrix} \frac{2}{xw_{\max} - xw_{\min}} \cdot \frac{z_{vp} - z_{prp}}{z_{back} - z_{prp}} & 0 & 0 & 0 \\ 0 & \frac{2}{yw_{\max} - yw_{\min}} \cdot \frac{z_{vp} - z_{prp}}{z_{back} - z_{prp}} & 0 & 0 \\ 0 & 0 & \frac{1}{z_{back} - z_{prp}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# 透视投影的规范化投影变换

(4) 将透视投影的规范化观察空间变换为平行投影的规范化观察空间。

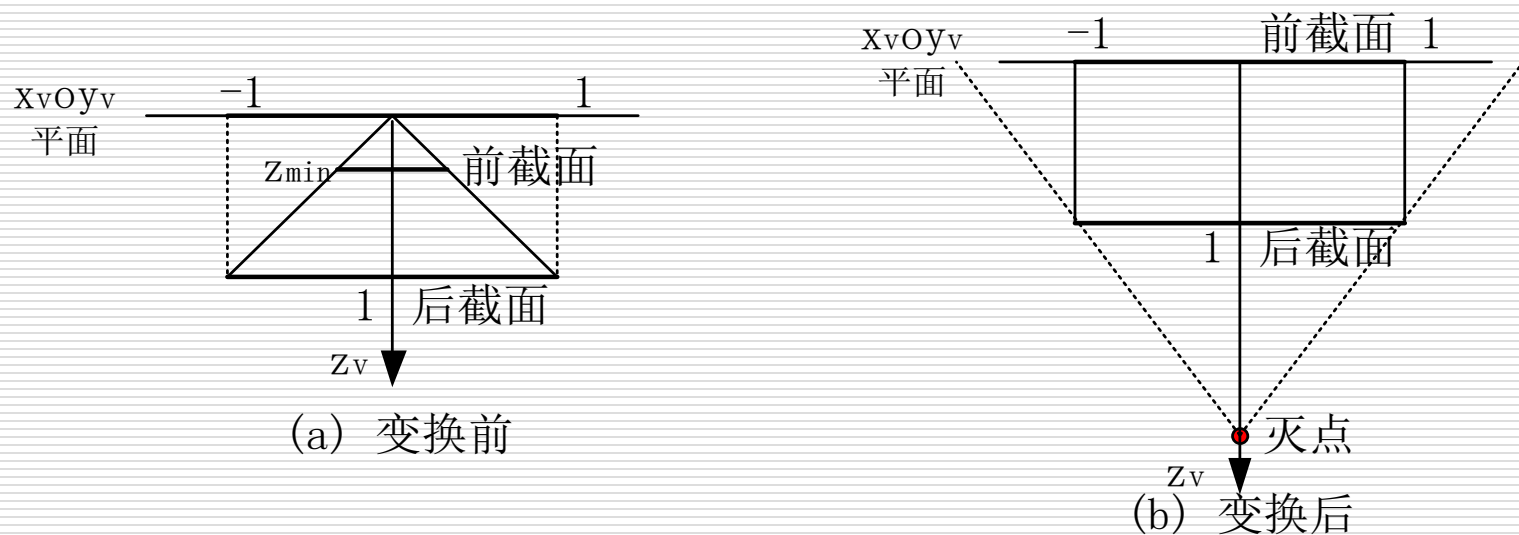


图6-49 规范化投影变换的透视变换

# 三维裁剪

---

- 三维裁剪保留所有在观察空间内的图形以便在输出设备中显示，所有在观察空间外的图形被丢弃。
- 三维直线段的裁剪
- 多边形面的裁剪

# 三维裁剪

---

四维齐次坐标表示的图形裁剪：

- 一是将齐次坐标转换为三维坐标，在三维空间中关于规范化观察空间剪裁；
- 一是直接在齐次坐标空间中进行裁剪。