

第 4 章 图形坐标变换与裁剪

本章导读

图形坐标变换与裁剪是计算机图形学中应用较为普遍的基础内容之一，是重要的图形处理技术。本章首先介绍二维图形坐标变换的基本原理、坐标变换的矩阵表示形式及不足、齐次变换矩阵和二维复合变换。然后介绍三维图形的坐标变换及三维图形变换的应用。最后在介绍开窗与裁剪的基本概念的基础上，介绍二维图形的裁剪，重点介绍直线裁剪的 Cohen-Sutherland 算法和直线的矢量裁剪方法。

本章要点

- 二维图形的坐标变换
- 三维图形的坐标变换
- 三维图形变换的应用
- 开窗与裁剪

4.1 二维图形的坐标变换

在图形显示过程中，用户需对图形进行平移、放大、旋转等基本的几何变换操作。图形的平移、放大、旋转从数学上看都是几何性质的“变换”，故又称为图形的几何变换。

对二维图形的坐标变换，就是指在不改变图形连线次序的情况下，对一个平面点集进行线性变换。

1. 二维图形坐标变换的基本原理

(1) 平移变换。一个点 $P(x,y)$ 平移到 $P^*(x^*,y^*)$ ，如图 4.1 (a) 所示，平移后产生的新坐标如下：

$$\begin{cases} x^* = x + Dx \\ y^* = y + Dy \end{cases}$$

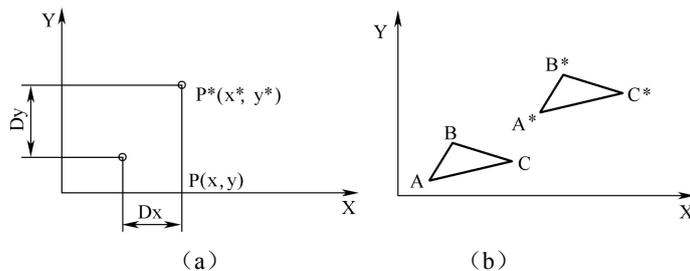


图 4.1 平移变换



对一个三角形来说，平移可只对其三个顶点进行。在平移后的顶点位置，再重新画出移动后的三角形，就可以实现三角形上所有点的平移，如图 4.1 (b) 所示。

(2) 变比例变换。在二维平面上，对一个点 $P(X,Y)$ 进行变比例变换，是将该点的两个坐标值分别按比例系数 S_x 和 S_y 进行变化。变换后，新点的坐标值是：

$$\begin{cases} X^* = S_x X \\ Y^* = S_y Y \end{cases}$$

其中： S_x 和 S_y 分别是 X 方向和 Y 方向上的比例系数。

对一个三角形进行变比例变换，同样仅需将该三角形的三个顶点进行变比例变换。然后在变换后的三顶点间按 $A、B、C$ 次序连线，就可得到变比例变换后的三角形，如图 4.2 (b) 所示。

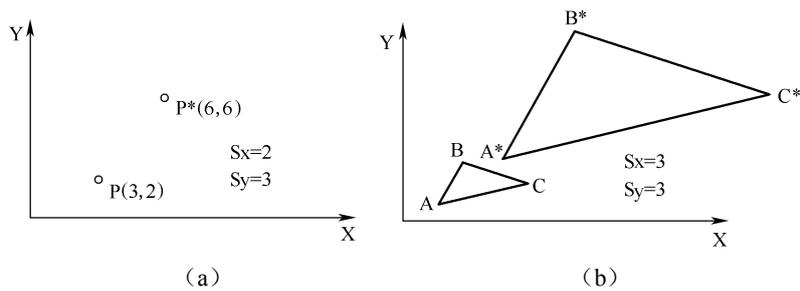


图 4.2 变比例变换

(3) 旋转变换：在二维平面上，对一个点 $P(X,Y)$ 进行旋转变换，是指将该点绕坐标原点旋转一定的角度 α 。旋转后新点的坐标值为：

$$\begin{cases} X^* = X \cos \alpha - Y \sin \alpha \\ Y^* = X \sin \alpha + Y \cos \alpha \end{cases}$$

2. 坐标变换的矩阵表示形式：

前面已介绍对二维图形的坐标变换，实际上是对一个平面点集进行线性变换。由于点集可用矩阵的方式来表达，因此对点的变换可以通过相应的矩阵运算来实现，即：

$$\text{旧点 (集)} \times \text{变换矩阵} \xrightarrow{\text{矩阵运算}} \text{新点 (集)}$$

一个点的坐标可以用矩阵形式 $[x \ y]$ 或 $\begin{bmatrix} x \\ y \end{bmatrix}$ 表示，坐标变换的矩阵表示形式为：

$$[X^* \ Y^*] = [X \ Y] \begin{bmatrix} a & b \\ c & d \end{bmatrix} = [aX+cY \ bX+dY]$$

其中： $T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ 为变换矩阵。

当 T 矩阵各元素取不同值时，便可得到不同的变换结果。

(1) 当 $T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ 时：



$$[X^* \ Y^*]=[X \ Y] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}=[X \ Y]$$

变换后 M^* 的坐标不变。

(2) 当 $T=\begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix}$ 时:

$$[X^* \ Y^*]=[X \ Y] \begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix}=[aX \ dY]$$

变换后 M 点各坐标相对原点伸缩, 即变比例变换。

(3) 当 $T=\begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$ 时:

$$[X^* \ Y^*]=[X \ Y] \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}=[X\cos\alpha-Y\sin\alpha \ X\sin\alpha+Y\cos\alpha]$$

即 M 点的位置绕坐标原点旋转 α 角, 为旋转变换。

(4) 当 $T=\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$ 时:

$$[X^* \ Y^*]=[X \ Y] \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}=[-X \ Y]$$

即点 M 与点 M^* 对称于 Y 轴, 称为对 Y 轴的反射 (对称) 变换。

(5) 当 $T=\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ 时:

$$[X^* \ Y^*]=[X \ Y] \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}=[Y \ X]$$

即点 M 与点 M^* 对称于 $Y=X$ 直线, 称为对直线 $Y=X$ 的反射 (对称) 变换。

注意, 不论变换矩阵中的元素 a 、 b 、 c 、 d 为何值, 都不能使图形产生平移变换, 即用 2 行 2 列的变换矩阵不能实现图形的平移变换。这就需要使用图形的另一种表示方法——齐次坐标。

3. 齐次坐标与齐次变换矩阵

上述几种基本变换的变换矩阵 $T=\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ 没有平移变换的功能。欲使二维图形产生平移, 必须在变换后的坐标中引入平移量 m 、 n , 即

$$\begin{cases} x^* = x + m \\ y^* = y + n \end{cases}$$

显然, 运用 $[x^* \ y^*]=[x \ y] \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ 是无法实现的。为了进行平移变换, 要给二维点的位置矢量增加一个附加坐标, 使之成为三维行向量 $[x \ y \ 1]$, 即用点的齐次坐标表示, 这样便可进行运算了。

(1) 齐次坐标。齐次坐标是将一个 n 维空间的点用 $n+1$ 维坐标来表示。如在直角坐标系



中，二维点 $[x \ y]$ 的齐次坐标通常用三维坐标 $[Hx \ Hy \ H]$ 表示，一个三维点 $[x \ y \ z]$ 的齐次坐标通常用四维坐标 $[Hx \ Hy \ Hz \ H]$ 表示。在齐次坐标系中，最后一维坐标 H 称为比例因子。

三维直角坐标与其齐次坐标的关系是：

$$\begin{cases} x = Hx / H \\ y = Hy / H \\ z = Hz / H \end{cases}$$

由于 H 的取值是任意的，所以任一点可用多组齐次坐标表示。在一般使用中，总是将 H 设为“1”，以保持两种坐标的一致。

(2) 齐次变换矩阵。对于二维坐标系上的点，齐次坐标为 $[Hx \ Hy \ H]$ ，而齐次坐标变换矩阵的形式是：

$$T = \begin{bmatrix} a & b & p \\ c & d & q \\ m & n & s \end{bmatrix}$$

将该矩阵分为四部分，其中 $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ 的作用是对被变换点的坐标进行放大、缩小和旋转变

换； $[m \ n]$ 的作用是对变换点的坐标进行平移变换； $\begin{bmatrix} p \\ q \end{bmatrix}$ 的作用是对变换点的坐标进行投影变

换； $[s]$ 是整个变换中的比例系数。

下面分别介绍利用齐次变换矩阵进行平移变换、变比例变换和旋转变换。

①当 $T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix}$ 时：

$$[XH \ YH \ H] = [X \ Y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix} = [X+m \ Y+n \ 1]$$

$$\begin{cases} X^* = XH/H = X + m \\ Y^* = YH/H = Y + n \end{cases}$$

这时为平移变换。

②当 $T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{bmatrix}$ 时：

$$\text{则 } [XH \ YH \ H] = [X \ Y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{bmatrix} = [X \ Y \ s]$$

$$\begin{cases} X^* = XH/H = X/s \\ Y^* = YH/H = Y/s \end{cases}$$

此时为变比例变换。



③当 $T = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$ 时:

$$[XH \ YH \ H] = [X \ Y \ 1] \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} = [X \cos \alpha - Y \sin \alpha \quad X \sin \alpha + Y \cos \alpha \quad 1]$$

$$\begin{cases} X^* = XH/H = X \cos \alpha - Y \sin \alpha \\ Y^* = YH/H = X \sin \alpha + Y \cos \alpha \end{cases}$$

此时为旋转变换。

4. 二维复合变换

实际上,上述介绍的几种基本变换一般不单独出现,通常出现的都是复合变换。有的图形须经过多次基本变换才能完成,这种由两个以上基本变换构成的变换称为复合变换。设各次变换的变换矩阵分别为 T_1, T_2, \dots, T_n , 则复合变换矩阵是各次变换矩阵的乘积。

当图形要对画面中的某一点 (x_0, y_0) 作放大时,可通过如下三种基本变换复合而成。

(1) 将坐标原点 $(0,0)$ 平移至 (x_0, y_0) , 变换矩阵为:

$$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_0 & -y_0 & 1 \end{bmatrix}$$

(2) 图形以 (x_0, y_0) 为中心作放大, 变换矩阵为:

$$T_2 = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(3) 将坐标原点自 (x_0, y_0) 的位置移回原处 $(0,0)$, 变换矩阵为:

$$T_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_0 & y_0 & 1 \end{bmatrix}$$

则以点 (x_0, y_0) 为中心, 放大系数分别为 s_x, s_y 的复合变换矩阵为:

$$\begin{aligned} T = T_1 \cdot T_2 \cdot T_3 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_0 & -y_0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_0 & y_0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ x_0(1-s_x) & y_0(1-s_y) & 1 \end{bmatrix} \end{aligned}$$

同理,当图形绕坐标原点以外的任意点 (x_0, y_0) 作旋转时,也可以通过三种基本变换复合而成,即将旋转中心平移到坐标原点,其变换矩阵为 T_1 ; 然后使图形绕坐标原点旋转 α 角,变换矩阵为 T_2 ; 最后将旋转中心平移回原来的位置,其变换矩阵为 T_3 。则绕坐标原点以外的任意点旋转 α 角的复合变换矩阵为:



$$\begin{aligned}
 T=T_1 \cdot T_2 \cdot T_3 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_0 & -y_0 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_0 & y_0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ x_0(1-\cos \alpha)+y_0 \sin \alpha & -x_0 \sin \alpha+y_0(1-\cos \alpha) & 1 \end{bmatrix}
 \end{aligned}$$

4.2 三维图形的坐标变换

三维图形的坐标变换是二维图形坐标变换的简单扩展。在进行二维图形坐标变换时，应用了二维点的三维齐次坐标 $[X \ Y \ 1]$ 及其相应的变换矩阵。同样，在进行三维图形的坐标变换时，可用四维齐次坐标 $[X \ Y \ Z \ 1]$ 来表示三维空间的点 $[X \ Y \ Z]$ 。三维图形的齐次变换矩阵为：

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

把矩阵 T 分为四块，每个子块对图形变换的作用如下：

$$\begin{bmatrix} a & b & c \\ d & e & f \\ h & i & j \end{bmatrix} \text{ —— 产生变比例变换、旋转变换、反射（对称）变换等。}$$

$[l \ m \ n]$ —— 产生沿 x 、 y 、 z 三个方向的平移变换。

$[s]$ —— 产生全比例变换。

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} \text{ —— 产生透视变换。}$$

4.2.1 三维基本变换

1. 比例变换

空间立体图形各个顶点坐标按规定比例放大或缩小属于三维比例变换。三维齐次变换矩阵左上角的 3×3 矩阵的主对角线上的元素 a 、 e 、 j 的作用是使图形产生比例变换。比例变换的齐次变换矩阵为：

$$T = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & e & 0 & 0 \\ 0 & 0 & j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

空间一点 (X, Y, Z) 的变化结果为



$$[X \ Y \ Z \ 1] \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & e & 0 & 0 \\ 0 & 0 & j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [aX \ bY \ jZ \ 1] = [X^* \ Y^* \ Z^* \ 1]$$

2. 反射（对称）变换

三维反射变换包括对原点、对坐标轴和对坐标平面的反射变换，常用的是对坐标平面的反射变换，下面介绍对坐标平面的反射变换。

(1) 对 xoy 平面的反射变换。齐次变换矩阵为：

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(2) 对 xoz 平面的反射变换。齐次变换矩阵为：

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(3) 对 yoz 平面的反射变换。齐次变换矩阵为：

$$T = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. 平移变换

平移变换是指立体在空间沿 x、y、z 三个方向移动一个位置，而立体本身的大小和形状并不改变。齐次变换矩阵为：

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & m & n & 1 \end{bmatrix}$$

空间一点(X,Y,Z)的平移变化结果为：

$$[X \ Y \ Z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & m & n & 1 \end{bmatrix} = [X+l \ Y+m \ Z+n \ 1] = [X^* \ Y^* \ Z^* \ 1]$$

l、m、n 分别为 X、Y、Z 三个方向的平移量，它们的正负决定了平移方向。

4. 旋转变换

三维旋转变换指空间立体绕某一轴旋转一个角度 α ，一般是绕坐标轴旋转 α 角。 α 角的正负按右手法则确定：右手大拇指指向旋转轴的正向，其余四个手指的指向即为 α 角的正向。



(1) 绕 X 轴旋转 α 角。空间立体绕 X 轴旋转 α 角后，各顶点的 X 坐标不变，只是 Y 和 Z 坐标发生变化。齐次变换矩阵为：

$$T_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(2) 绕 Y 轴旋转 α 角。空间立体绕 Y 轴旋转 α 角后，各顶点的 Y 坐标不变，只是 X 和 Z 坐标发生变化。齐次变换矩阵为：

$$T_y = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(3) 绕 Z 轴旋转 α 角。空间立体绕 Z 轴旋转 α 角后，各顶点的 Z 坐标不变，只是 X 和 Y 坐标发生变化。齐次变换矩阵为：

$$T_z = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

已知某长方体各顶点坐标为 A(3,2,1.5)、B(1,2,1.5)、C(1,2,2.5)、D(3,2,2.5)、E(3,1,1.5)、F(1,1,1.5)、G(1,1,2.5)、H(3,1,2.5)，将该长方体对 yoz 平面进行反射变换，试求变换结果。

解：将长方体各顶点坐标进行反射变换如下：

$$\begin{matrix} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{matrix} \begin{pmatrix} 3 & 2 & 1.5 & 1 \\ 1 & 2 & 1.5 & 1 \\ 1 & 2 & 2.5 & 1 \\ 3 & 2 & 2.5 & 1 \\ 3 & 1 & 1.5 & 1 \\ 1 & 1 & 1.5 & 1 \\ 1 & 1 & 2.5 & 1 \\ 3 & 1 & 2.5 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -3 & 2 & 1.5 & 1 \\ -1 & 2 & 1.5 & 1 \\ -1 & 2 & 2.5 & 1 \\ -3 & 2 & 2.5 & 1 \\ -3 & 1 & 1.5 & 1 \\ -1 & 1 & 1.5 & 1 \\ -1 & 1 & 2.5 & 1 \\ -3 & 1 & 2.5 & 1 \end{pmatrix} \begin{matrix} A^* \\ B^* \\ C^* \\ D^* \\ E^* \\ F^* \\ G^* \\ H^* \end{matrix}$$

其变换结果如图 4.3 所示。

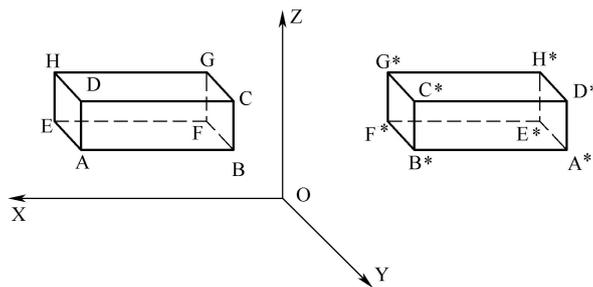


图 4.3 对 yoz 平面的反射变换



4.2.2 三维基本变换矩阵的组合

和二维图形的复合变换一样，通过对三维基本变换矩阵的组合，可以实现对三维立体的较为复杂的变换。

1. 绕通过坐标原点的任意轴的旋转变换矩阵

设 on 为通过坐标原点的任意方向的直线，其方向余弦分别为 n_1 、 n_2 、 n_3 ，则点 p 绕 on 轴转动 α 角到达 p' 点[见图 4.4 (a)]的变换矩阵为：

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

上述矩阵 T_R 是由几个基本变换矩阵组合而成的，它可通过以下几个步骤获得：

(1) 将 on (即 $\triangle onM$) 绕 x 轴逆时针旋转到 xoz 平面[见图 4.4 (b)]，旋转角 ϕ 根据已知条件很容易求得。此旋转由变换矩阵 T_x 实现。

(2) 将 xoz 平面上的 on' 绕 y 轴旋转到与 oz 轴重合[见图 4.4 (c)]，旋转角 θ 可根据已知条件求得。此旋转由变换矩阵 T_y 实现。

(3) 将 p 点绕 z 轴旋转 α 角，由旋转变换矩阵 T_z 实现。

(4) 沿步骤 (2) 的相反方向旋转 θ 角，即步骤 (2) 的逆变换，由旋转变换矩阵 T_y^{-1} 实现。

(5) 沿步骤 (1) 的相反方向旋转 ϕ 角，即步骤 (1) 的逆变换，由旋转变换矩阵 T_x^{-1} 实现。

将上述五步进行级联，便可得到绕通过坐标原点的任意轴 on 旋转的变换矩阵 T_R ，即

$$T_R = T_x \cdot T_y \cdot T_z \cdot T_y^{-1} \cdot T_x^{-1}$$

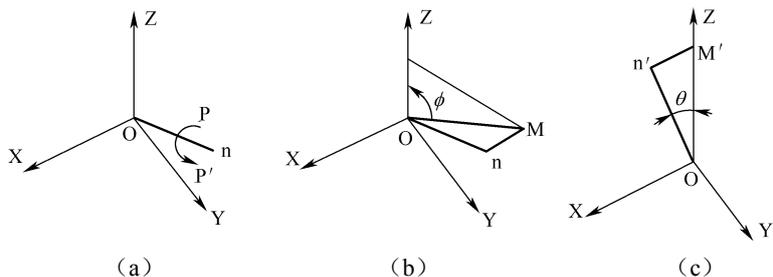


图 4.4 绕过坐标原点的任意轴 on 的旋转变换

2. 绕通过任意点 $P(l,m,n)$ ，方向余弦分别为 n_1 、 n_2 、 n_3 的轴的旋转变换矩阵

如图 4.5 所示，该变换可通过以下三个步骤来实现：

(1) 对空间立体和旋转轴一起进行平移变换，使旋转轴通过坐标原点，变换矩阵为：

$$T_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -l & -m & -n & 1 \end{bmatrix}$$

(2) 使立体绕通过坐标原点的轴旋转 α 角，旋转变换矩阵 T_R 。

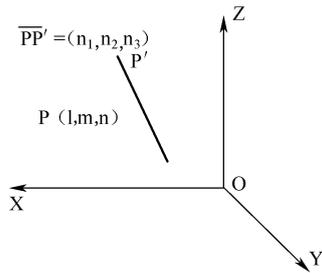


图 4.5 绕过任意点 P 的轴的旋转变换

(3) 将旋转轴和旋转后的空间体一起反向平移，使旋转轴回到原来的位置，变换矩阵为：

$$T_p^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & m & n & 1 \end{bmatrix}$$

绕通过任意点 P(l,m,n)的轴的旋转变换矩阵为：

$$T = T_p \cdot T_R \cdot T_p^{-1}$$

4.3 三维图形变换的应用

4.3.1 正投影变换

正投影变换可得到国家标准规定的六个基本视图——主视图、俯视图、左视图、右视图、仰视图和后视图。现以工程中广泛应用的主视图、俯视图和左视图为例，说明其变换矩阵的求法。

1. 主视图变换矩阵

如图 4.6 所示的三维立体，要得到其主视图，须向 V 面投影，即将立体的全部 y 坐标变为零。其变换关系如下：

$$[x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \ 0 \ z \ 1] = [x^* \ y^* \ z^* \ 1]$$

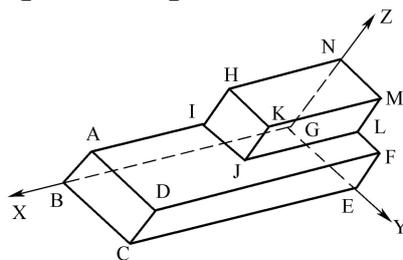


图 4.6 三维立体模型

由此可知，主视图变换矩阵为：



$$T_V = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. 俯视图变换矩阵

如图 4.7 所示, 将三维立体的点集先向 H 面投影 (即令 $z=0$), 再按右手系统绕 x 轴旋转 -90° , 然后沿 z 轴方向平移 $-n$ ($n>0$), 以使 V、H 投影保持一定间距 n 。上述变换过程可表示为:

$$[x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -n & 1 \end{bmatrix}$$

向 H 面投影 绕 x 轴旋转 沿 z 轴平移

$$= [x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -n & 1 \end{bmatrix} = [x \ 0 \ -y-n \ 1] = [x^* \ y^* \ z^* \ 1]$$

由此可知, 俯视图变换矩阵为:

$$T_H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -n & 1 \end{bmatrix}$$

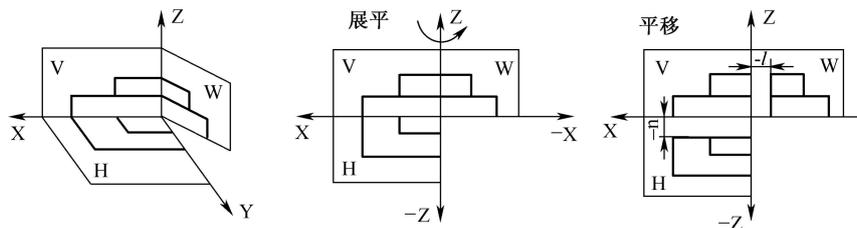


图 4.7 三视图的展开

3. 左视图变换矩阵

如图 4.7 所示, 将三维立体的点集向 W 面投影, 即令 $x=0$, 再按右手系统绕 z 轴旋转 90° , 使之与 V 面重合, 然后沿 x 轴方向平移 $-l$ ($l>0$), 以使 V、W 投影保持一定间距 l 。上述变换过程可表示为:

$$[x \ y \ z \ 1] \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

向 W 面投影 绕 z 轴旋转 沿 x 轴平移



$$= [x \ y \ z \ 1] \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} = [-y-1 \ 0 \ z \ 1] = [x^* \ y^* \ z^* \ 1]$$

由此可知，左视图变换矩阵为：

$$T_w = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

4.3.2 正轴测投影变换

正轴测投影图是工程上应用广泛的二维图形。

若将图 4.8 (a) 所示的立方体直接向 V 面投影，就得到 (b) 图；若将立方体绕 z 轴正转 θ 角，再向 V 面投影，就得到 (c) 图；若将立方体先绕 z 轴正转 θ ，再绕 x 轴旋转 $-\phi$ ($\phi > 0$) 角，然后向 V 面投影，就得到 (d) 图，即立方体的正轴测投影图。其变换矩阵为：

$$T_{\text{正轴测}} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ = \begin{bmatrix} \cos\theta & 0 & -\sin\theta\sin\phi & 0 \\ -\sin\theta & 0 & -\cos\theta\sin\phi & 0 \\ 0 & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

在上述所示的正轴测投影变换矩阵中，只要给 θ 、 ϕ 不同的值，就可得到不同的正轴测投影图。

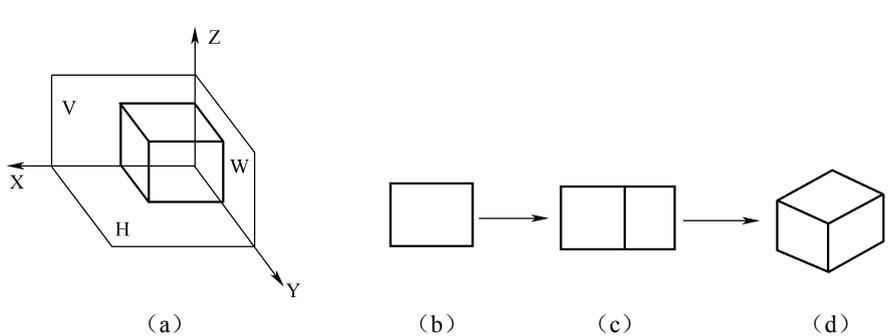


图 4.8 轴测投影图的生成过程

1. 正等轴测投影变换矩阵

按国家标准规定，以 $\theta=45^\circ$ 、 $\phi=35.2644^\circ$ 代入式 (4.1)，即可得到正等轴测投影变换矩阵如下：



$$T = \begin{bmatrix} 0.7071 & 0 & -0.4082 & 0 \\ -0.7071 & 1 & -0.4082 & 0 \\ 0 & 0 & 0.8165 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. 正二等轴测投影变换矩阵

按国家标准规定,以 $\theta=20.7^\circ$ 、 $\phi=19.47^\circ$ 代入式(4.1),即可得到正二等轴测投影变换矩阵如下:

$$T = \begin{bmatrix} 0.9354 & 0 & -0.1178 & 0 \\ -0.3535 & 0 & -0.3117 & 0 \\ 0 & 0 & 0.9428 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4.4 开窗与裁剪

在计算机辅助设计中,常常需要对一幅画面中的特定区域进行处理。例如,区域内画面的放大,细节研究与修改,或突出其所表达的内容等,有时还要求把区域外的画面全部擦去。上述过程可通过开窗口和裁剪来完成。本节将介绍这两种操作的基本方法。

4.4.1 基本概念和术语

(1) 用户坐标系(世界坐标系):用来定义三维物体的形状、大小和位置,指定观察点位置及观察方向等,且符合右手法则的三维笛卡尔坐标系。

(2) 屏幕坐标(设备坐标):与设备相关联的坐标系。例如,屏幕显示用坐标通常取其左上角或左下角为顶点,以水平向右方向为x轴的方向,以垂直向下或向上为y轴的方向,取屏幕分辨率刻度值作为坐标刻度值。

(3) 窗口:在实际画面中(用户坐标系),由用户指定的矩形区域。通常表示以矩形框取出物理世界中的一部分,相当于透过窗口观察这部分图形。

(4) 视区:在屏幕上(屏幕坐标系)用来显示窗口内的画面内容的矩形区域。

(5) 裁剪:保留窗口内的画面,擦去窗口外的全部画面的方法。用以从一幅大的画面中抽取所需的具体信息,以显示某一局部画面或视图。

4.4.2 窗口—视区变换

由于窗口和视区是在不同的坐标系中定义的,所以当窗口内的图形信息拿到视区去输出之前,必须进行坐标变换,这就是窗口—视区变换。

窗口—视区变换要求在保持一定比例关系的前提下,把窗口中的点 (X_w, Y_w) 变换到视区中 (X_v, Y_v) 位置上,即保持点在闭合矩形中的相对位置不变。如图4.9所示,设在用户坐标系中选定窗口的左下角坐标为 (X_{w1}, Y_{w1}) ,窗口高度为 H_w ,宽为 L_w ;而在屏幕坐标系中,视区左下角坐标为 (X_{v1}, Y_{v1}) ,高度为 H_v ,宽为 L_v ,则窗口中的点 (X_w, Y_w) 和视区中对应点 (X_v, Y_v) 的相互变换关系式为:

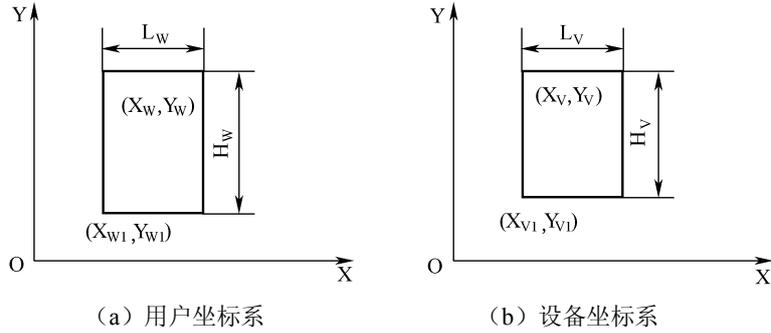


图 4.9 窗口、视区保持比例

$$\begin{cases} X_v = X_{v1} + \frac{L_v}{L_w}(X_w - X_{w1}) \\ Y_v = Y_{v1} + \frac{H_v}{H_w}(Y_w - Y_{w1}) \\ X_w = X_{w1} + \frac{L_w}{L_v}(X_v - X_{v1}) \\ Y_w = Y_{w1} + \frac{H_w}{H_v}(Y_v - Y_{v1}) \end{cases}$$

当图形需要或可能在多种输出设备上输出时，宜先将窗口变换到规格化输出设备坐标系中的视区内，然后按不同输出设备的分辨率再变换到具体输出设备。

在上述变换中，固定视区尺寸，改变窗口的大小（改边参数 H_w 、 L_w ），则改变显示的比例，即放大或缩小；而固定视区尺寸，改变窗口的位置，则可改变显示部位，即实现所谓的摇视。

4.4.3 二维图形的裁剪

1. 点的裁剪

在图形剪裁中，最基本的是点的裁剪。图 4.10 中，直线 $x=x_{\max}$ ， $x=x_{\min}$ ， $y=y_{\max}$ ， $y=y_{\min}$ 是确定窗口 ABCD 的四条边界，对于某一点 $P(x,y)$ ，只要满足

$$x_{\min} \leq x \leq x_{\max} \quad y_{\min} \leq y \leq y_{\max} \quad (4.2)$$

则该点一定落在边界所围成的矩形框内，为可见点。由此可知，只要根据式 (4.2) 就可以对任意图形进行裁剪。但是，若把图形的所有元素都转换成点，再用式 (4.2) 来判断，显然这样进行裁剪的效率是很低的。

2. 直线裁剪的 Cohen-Sutherland 算法

直线的裁剪已提出了许多算法，Cohen-Sutherland 算法是其中较著名的一种，其步骤如下：

(1) 如图 4.10 所示，延长窗口的边界，将屏幕分为九个小区域，中央小区域就是要裁剪的区域。每个小区域用一个四位二进制代码表示，其中各位数字的含义如下：

- 小区域在裁剪区域之左，即 $x < x_{\min}$ ，则 $c_1=1$ ，否则 $c_1=0$ ；
- 小区域在裁剪区域之右，即 $x > x_{\max}$ ，则 $c_2=1$ ，否则 $c_2=0$ ；
- 小区域在裁剪区域之下，即 $y < y_{\min}$ ，则 $c_3=1$ ，否则 $c_3=0$ ；
- 小区域在裁剪区域之上，即 $y > y_{\max}$ ，则 $c_4=1$ ，否则 $c_4=0$ ；

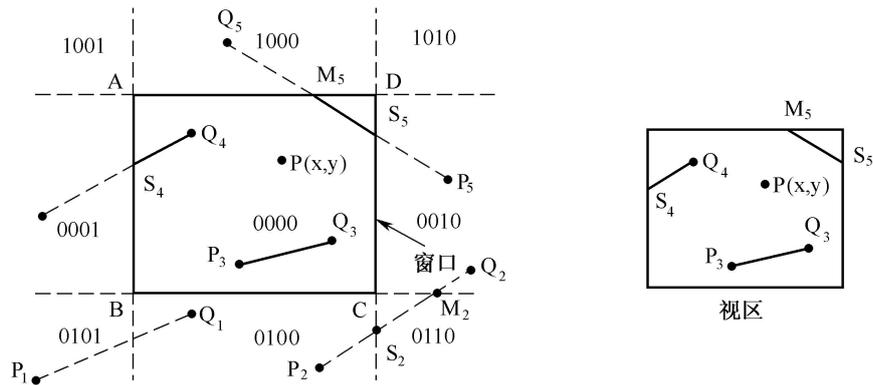


图 4.10 图形裁剪

(2) 按上述规定确定被裁剪线段两端点所在小区域的代码。

(3) 按被裁剪线段两端点代码，对被裁剪直线段进行如下可见性判断：

如果两端点的代码全部由数字“0”组成，则此线段必全部位于窗口之内，为可见线段，应将整个线段画出，如线段 P_3Q_3 ；如果两端点的代码按相同位置的位进行逻辑与运算，结果不全为 0（即两代码中至少有一个相同位置的位的数字同时为 1），则此线段两端点都在裁剪区一个边界的外侧，为不可见线段，应该裁剪掉，如 P_1Q_1 ；如果两端点代码不全部由数字“0”组成，且其按位进行逻辑与运算的结果为 0。则此线段可能与裁剪区域相交，需要进行进一步的计算和判断。

(4) 求被裁剪直线段与裁剪区域边界线的交点，舍弃在区域外的线段部分，对留下部分重新进行步骤（2）以后各步。

3. 直线的矢量裁剪方法

这种裁剪方法与上面所说的方法相类似，只是判别端点是否落在窗口框内所采用的过程不一样。在该算法中，同样用四条窗口边框直线把平面分割成九个区域，为了叙述算法过程方便起见，我们把这九个区域分别标上号码，如图 4.11 所示。

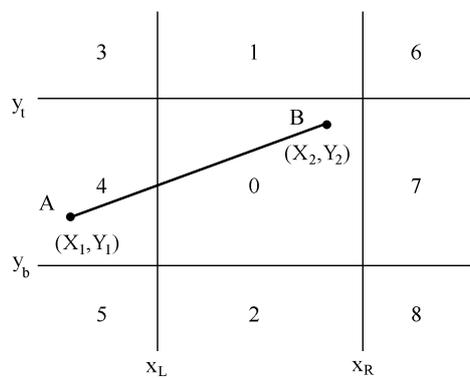


图 4.11 矢量裁剪法

以图 4.11 中的线段 AB 为例来说，该裁剪算法的执行步骤如下：

(1) 若直线 AB 满足下述四个条件之一，即：



$$\max(x_1, x_2) < x_L; \quad \min(x_1, x_2) > x_R;$$

$$\max(y_1, y_2) < y_L; \quad \min(y_1, y_2) > y_R;$$

则线段 AB 不会处于窗口内，过程就此结束，且无输出线段。

(2) 若线段 AB 满足 $x_L \leq x_1 \leq x_R$ ，且 $y_L \leq y_1 \leq y_R$ ，则 AB 的始点 A 在 0 区内，那么窗口内可见线段的新始点坐标即为： $x_s = x_1$ ， $y_s = y_1$ 。

否则，线段 AB 与窗口的关系及其新始点坐标 (x_s, y_s) 的求解公式可讨论如下：

(3) 若 $x_1 < x_L$ ，则：

$$\begin{cases} x_s = x_1 \\ y_s = y_1 + (x_L - x_1)(y_2 - y_1)/(x_2 - x_1) \end{cases}$$

若求得的 y_s 满足 $y_b \leq y_s \leq y_t$ ，则 x_s ， y_s 求解有效；否则：

① 若 (x_1, y_1) 在 4 区，则线段 AB 与窗口无交点，过程结束，且无输出线段；

② 当 (x_1, y_1) 在 5 区，且 $y_s > y_t$ ，或者当 (x_1, y_1) 在 3 区，且 $y_s < y_b$ 时，则线段 AB 与窗口无交点，过程结束，且无输出线段；

③ 若 $y_1 < y_b$ ，则：

$$\begin{cases} x_s = x_1 + (y_b - y_1)(x_2 - x_1)/(y_2 - y_1) \\ y_s = y_b \end{cases} \quad (4.3)$$

若 $y_1 > y_t$ ，则：

$$\begin{cases} x_s = x_1 + (y_t - y_1)(x_2 - x_1)/(y_2 - y_1) \\ y_s = y_t \end{cases} \quad (4.4)$$

式 (4.3)，(4.4) 求出的 x_s 如果满足 $x_L \leq x_s \leq x_R$ ，则结果有效，否则线段 AB 与窗口无交点，过程结束，且无输出线段。

(4) 当 $x_1 > x_R$ 时，可用以上类似的过程求出线段 AB 与窗口右边框的交点。

(5) 当 (x_1, y_1) 在 1, 2 区时，即可用式 (4.3) 和式 (4.4) 求出线段 AB 与窗口上、下两边框线的交点。如求得的 x_s 满足 $x_L \leq x_s \leq x_R$ ，则求解结果有效，否则线段 AB 与窗口无交点，过程结束，且无输出线段。

以上过程，我们仅求出了线段 AB 在窗口内的可见段的起点坐标 (x_s, y_s) 。同样，用类似的过程，可以求出线段 AB 在窗口内的可见段的终点坐标 (x_e, y_e) 。

4. 多边形裁剪的 Sutherland-Hodgman 算法

Sutherland-Hodgman 算法是解决多边形裁剪的较好算法，其处理对象是多边形的顶点，依次用窗口的四条边框线对多边形进行裁剪，即先用一条边框线对整个多边形进行裁剪，得到一个或者若干个新的多边形，再用第二条边对这些新产生的多边形进行裁剪，如此进行下去，直到用四条边框线都裁剪完为止。如图 4.12 所示，连接多边形的顶点 $P_1P_2, P_2P_3, \dots, P_6P_7, P_7P_1$ ，组成多边形的条边。经过一次边框线对多边形的裁剪，就产生一组新的顶点序列，如图 4.12 中的 $Q_1, Q_2, \dots, Q_7, Q_8, R_1, R_2, \dots, R_8, R_9, S_1, S_2, \dots, S_7, S_8$ 和 $T_1, T_2, \dots, T_7, T_8$ 。裁剪顺序是左边框线，顶边框线，右边框线，底边框线。作裁剪的边框线 e 依次检验多边形的每个顶点 P_i ，处于该边框线可见侧的顶点被列入新的顶点序列中，而处于不可见侧的顶点



则被删除掉。此外还要检验 P_i 点和它的前一点 P_{i-1} 是否处于边框线的同侧 (P_1 点除外), 如果不处在同侧, 则必须求出边框线和线段 $P_i P_{i-1}$ 的交点, 并把这个交点作为新的顶点列入新的输出顶点序列中, 如 Q_3 、 Q_4 。当最后一点 P_n 被检验完后, 还要检验线段 $P_n P_1$ 是否和边框线 e 相交。

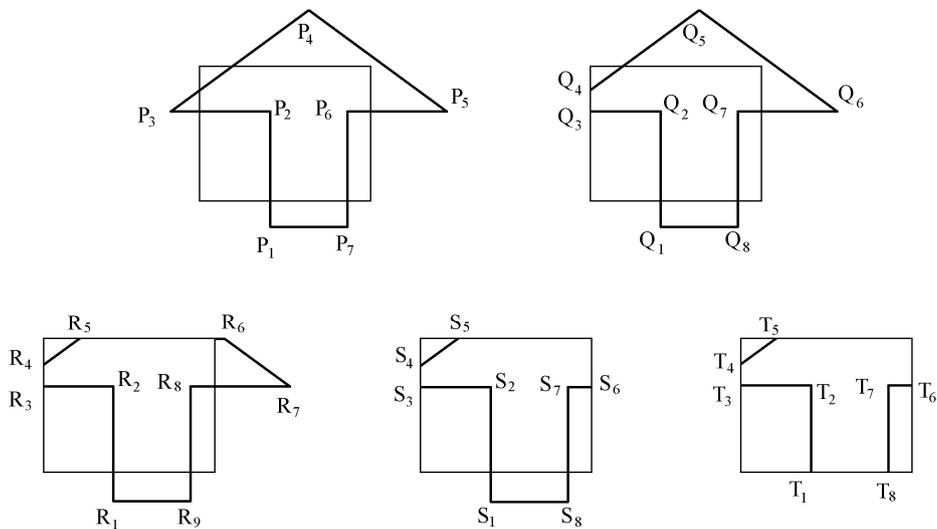


图 4.12 多边形裁剪过程

习题四

1. 已知 $\triangle ABC$ 各顶点的坐标分别为: $A(20,15)$ 、 $B(20,40)$ 、 $C(40,30)$, 对此分别进行下列比例变换:

- (1) 使长度方向 (x 向) 缩小一半, 高度方向 (y 向) 增加一倍;
- (2) 使整个三角形放大为原来的 1.5 倍。

2. 对题 1 的三角形 ABC 分别进行下列反射变换:

- (1) 对 x 轴反射;
- (2) 对 y 轴反射;
- (3) 对原点反射;
- (4) 对 45° 线反射。

3. 已知四边形 $ABCD$ 各顶点的坐标分别为: $A(9,9)$ 、 $B(30,9)$ 、 $C(30,24)$ 、 $D(9,24)$, 试用齐次变换矩阵对此进行下列变换, 并画出变换前后的图形。

- (1) 沿 x 向平移 10, 沿 y 向平移 20, 再绕坐标原点逆时针旋转 90° ;
- (2) 绕坐标原点逆时针旋转 90° , 再沿 x 向平移 10, 沿 y 向平移 20。

4. 将题 3 中的四边形 $ABCD$ 绕点 $P(12,35)$ 逆时针旋转 60° , 试求变换结果。

5. 已知正方体的棱边长为 60mm, 其中一个顶点在坐标原点, 且正方体位于第一卦角内, 将其沿 x 向平移 10, 沿 y 向平移 20, 沿 z 向平移 15, 试用齐次变换矩阵求变换结果。



6. 对题 5 变换后的立体进行下列旋转变换：
 - (1) 绕 x 轴旋转 30° ；
 - (2) 绕 y 轴旋转 45° ；
 - (3) 绕 z 轴旋转 30° 。
7. 按照自己的理解，叙述窗口和裁剪的定义及它们的用途。
8. 试用 C 语言编程来实现直线裁剪的 Cohen-Sutherland 算法。