第五章 三维图形变换

三维图形的几何变换是指对三维图形的几何信息经过平移、比例、旋转等变换后产生新的图形。三维基本几何变换都是相对于坐标原点和坐标轴进行的几何变换。正如在二维几何变换中提到的那样，用齐次坐标表示点的变换将非常方便，因此在本节中所有的几何变换都将采用齐次坐标进行运算。

5.1 三维图形几何变换矩阵

由于用齐次坐标表示，三维几何变换的矩阵是一个 **4** 阶方阵，其形式如下：

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1] \cdot T_{3D} = [x \quad y \quad z \quad 1] \cdot \begin{bmatrix} a & b & c & p \\ d & e & f & q \\ j & h & i & r \\ l & m & n & s \end{bmatrix}$$

T_{3D} 分为四个矩阵，

$T_1 = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$ 对图形进行比例、旋转、对称等变换；

$T_2 = [l \quad m \quad n]$ 对图形进行平移变换；

$T_3 = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$ 对图形做透视投影变换。

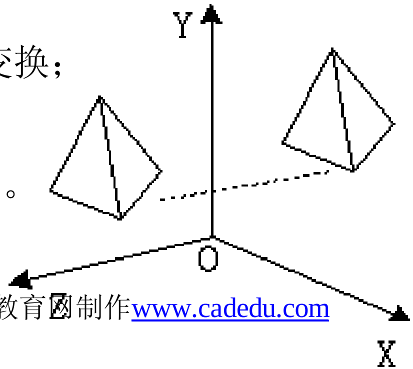


图5-1 三维图形平移示意图

$T_4 = [s]$产生整体比例变换。

5.2 三维图形基本变换矩阵

5.2.1 平移变换

三维坐标系中图形的平移如图 **5-1** 所示。

坐标平移变换：$x' = x + l$　　$y' = y + m$　　$z' = z + n$

参照二维的平移变换，我们很容易得到三维平移变换矩阵：

$$[x' \quad y' \quad z' \quad 1] = T \bullet [x \quad y \quad z \quad 1] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{bmatrix} [x \quad y \quad z \quad 1]$$

$$\text{其中}\ T=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ \mathbf{l} & m & n & 1\end{bmatrix}\text{为三维平移变换矩阵。}$$

5.2.2 比例变换

坐标**比例**变换：

$$\begin{aligned}x' &= S_x x\\ y' &= S_y y\\ z' &= S_z z\end{aligned}$$

三维比例变换：

$$\begin{bmatrix} X' & Y' & Z' & 1 \end{bmatrix} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_x \cdot X & S_y \cdot Y & S_z \cdot Z & 1 \end{bmatrix}$$

相对原点三维图形比例变换矩阵$T = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

5.2.3 绕坐标轴的旋转变换

考虑右手坐标系下相对坐标原点绕坐标轴旋转θ角的变换矩阵。三维基

本旋转变换：绕坐标系 X、Y、Z 轴旋转。

A. 绕 **x** 轴旋转，

三维变换

$$[X' \quad Y' \quad Z' \quad 1] = [X \quad Y \quad Z \quad 1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

其中变换矩阵：$T_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ □□> 0 右手法则

B. 绕 **y** 轴旋转

三维变换

$$[X' \quad Y' \quad Z' \quad 1] = [X \quad Y \quad Z \quad 1]\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

其中变换矩阵：$T_y = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ □□> 0 右手法则

C. 绕 **z** 轴旋转

三维变换

$$[X' \quad Y' \quad Z' \quad 1] = [X \quad Y \quad Z \quad 1]\begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

其中变换矩阵：$T_Z(q) = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $q > 0$相当于二维逆时针，是指符合右手法则

D. 绕任意轴的旋转变换

设旋转轴 **AB** 由任意一点 **A**（x_a，y_a，z_a）及其方向数（**a**，**b**，**c**）定义，空间一点 **P**（x_p，y_p，z_p）绕 **AB** 轴旋转 θ 角到 $P'(x_p'$，y_p'，$z_p')$，如图 **5-2** 所示，则：

$$[x'_p \quad y'_p \quad z'_p \quad 1] = [x_p \quad y_p \quad z_p \quad 1]R_{ab}(q)$$

可以通过下列步骤来实现 **P** 点的旋转：

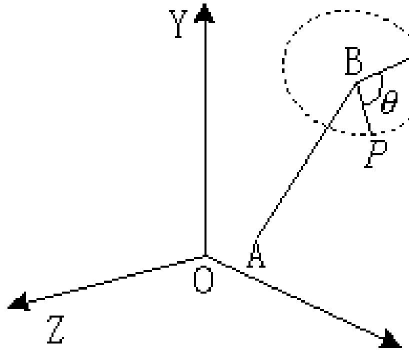


图 5-2 三维图形旋转示意图

1.　将 **A** 点移到坐标原点。

2.　使 **AB** 分别绕 **X** 轴、**Y** 轴旋转适当角度与 **Z** 轴重合。

3.　将 **AB** 绕 **Z** 轴旋转 θ 角。

4.　作上述变换的逆操作，使 **AB** 回到原来位置。

所以实现 **P** 点的旋转可写为：

$$R_{ab}(\theta) = T^{-1}(x_a, y_a, z_a)R_x^{-1}(\alpha)R_y^{-1}(\beta)R_z(\theta)R_y(\beta)R_x(\alpha)T(x_a, y_a, z_a)$$ 其中各

个矩阵的形式参照上面所讲的平移、选择矩阵，而α、β分别是 **AB** 在 **YOZ** 平面与 **XOZ** 平面的投影与 **Z** 轴的夹角。

5.2.4 对称变换

(1) 相对于 **X** 轴对称，坐标变换有

$$\begin{cases} X' = X \\ Y' = -Y \\ Z' = -Z \end{cases},$$

即

$$
\begin{bmatrix} X' & Y' & Z' & 1 \end{bmatrix} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

(2)　　相对于 **Y** 轴对称，坐标变换有

$$
\begin{cases} X' = -X \\ Y' = Y \\ Z' = -Z \end{cases}
$$

即

$$[X' \quad Y' \quad Z' \quad 1] = [X \quad Y \quad Z \quad 1]\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(3) 相对于 **Z** 轴对称，坐标变换有

$$\begin{cases} X' = -X \\ Y' = -Y \\ Z' = Z \end{cases}$$

即

$$\begin{bmatrix} X' & Y' & Z' & 1 \end{bmatrix} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(4)　相对于坐标原点对称，坐标变换有

$$\begin{cases} X' = -X \\ Y' = -Y \\ Z' = -Z \end{cases}$$

即

$$\begin{bmatrix} X' & Y' & Z' & 1 \end{bmatrix} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

（5）关于三个坐标平面 XY、YZ 及 XZ 的对称变换分别为：

$$T_{xy} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{yz} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{xz} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5.2.5 错切变换

图形沿 X 轴、Y 轴、Z 轴方向错切时，其变换矩阵的一般表达式：

$$T=\begin{bmatrix}1 & b & c & 0\\ d & 1 & f & 0\\ g & h & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

对应坐标变换表示为：

$$\begin{aligned}x' &= x+dy+gz\\ y' &= bx+y+hz\\ z' &= cx+fy+z\end{aligned}$$

根据不同的错切方向，三维错切变换矩阵有以下几种。

1．沿 X 轴方向错切，错切变换矩阵：

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

当 **d=0** 时，错切平面离开 **Z** 轴，沿 **X** 方向沿移动；

当 **g=0** 时，错切平面离开 **Y** 轴，沿 **X** 方向沿移动；

2．沿 **Y** 轴方向错切，错切变换矩阵：

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

当 **b=0** 时，错切平面离开 **Z** 轴，沿 **Y** 方向沿移动；

当 **h=0** 时，错切平面离开 **X** 轴，沿 **Y** 方向沿移动；

3．沿 **Z** 轴方向错切，错切变换矩阵：

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

当 **c=0** 时，错切平面离开 **Y** 轴，沿 **Z** 方向沿移动；

当 f **=0** 时，错切平面离开 **X** 轴，沿 **Z** 方向沿移动；

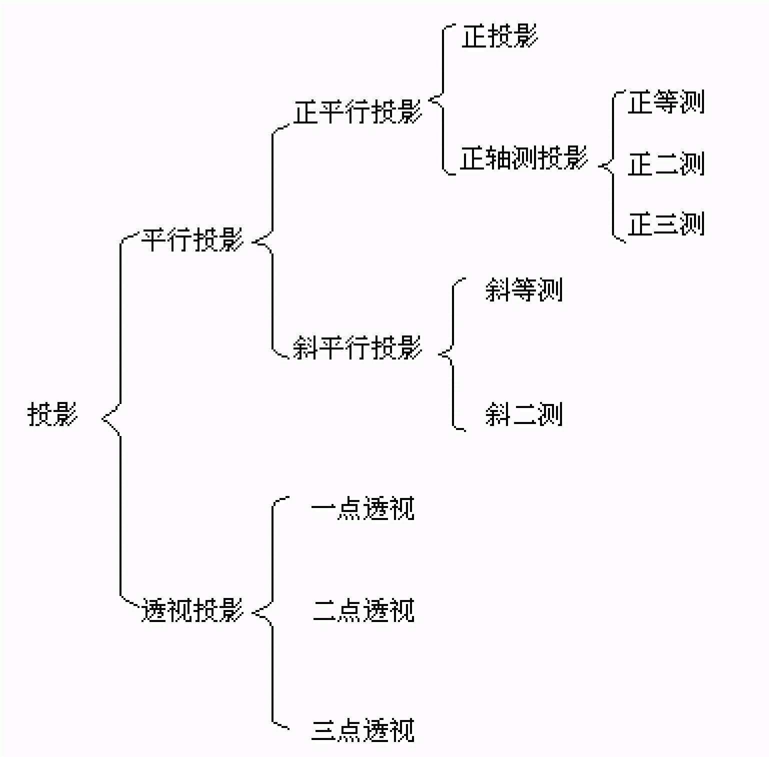
5.3 图形的投影变换

前面我们讨论了二维、三维图形的几何变换，由于显示屏幕是二维的，所以要输出三维形体，需要将三维坐标表示的几何形体变换成二维坐标表示的图形，这就是图形的投影变换。

在三维空间中，选择一个点，记该点为投影中心，不经过这个点再定义一个平面，称该平面为投影面，从投影中心向投影面引出任意条射线，称这些射线为投影线；穿过物体的投影线将与投影面相交，在投影面上形成物体的像，称这个像为三维物体在二维投影面上的投影。这样将三维空间的物体变换到二维平面上的过程称为投影变换。

5.3.1 投影变换分类

投影变换又分为透视投影和平行投影，其主要区别在于透视投影的投影中心到投影面之间的距离是有限的，而平行投影的投影中心到投影面之间的距离是无限的。当投影中心在无穷远时，投影线互相平行，所以，平行投影表示时只给出投影线方向即可，而透视投影要明确指定投影中心的位置。投影变换的分类情况如下表所示：

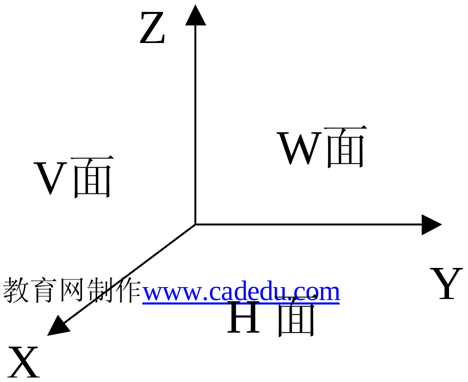

正投影
正平行投影
正等测
正轴测投影
正二测
正三测
平行投影
斜等测
斜平行投影
投影
斜二测
一点透视
透视投影
二点透视
三点透视

5.3.2 平行投影

平行投影根据投影方向与投影面的夹角分为两类，即正投影与斜投影，当投影方向垂直于投影面时称为正投影，否则为斜投影。

5.3.2.1 正平行投影（三视图）

正平行投影的投影中心到投影面之间的距离是无限的正投影。正平行投影又包括：三视图和正轴测。



1．三视图

工程中通常将三维坐标系 OXYZ 三个坐标平面分为：H 面（XOY 面）、V 面（XOZ 面）和 W 面（YOZ 面），如图 5-3 所示。三维图形在 V 面上的投影称为主视图、在 H 面上的投影称为俯视图、在 W 面上的投影称为侧视图。

（1）视图　将三维形体向 xoz 面（又称 V 面）作垂直投影（即正平行投影），得到主视图。主视图变换矩阵：

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

2）俯视图　三维形体向 xoy 面（又称 H 面）作垂直投影得到俯视图，

俯视图变换矩阵：

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

3）侧视图　获得侧视图是将三维形体往 **yoz** 面（侧面 **W**）作垂直投影，步骤：

侧视图变换矩阵：

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

H 面、V 面、W 面的三个正投影以一定方式摆在某一个平面上——三视图

2. 正轴侧投影变换

若将空间立体绕某个投影面所包含的两个轴向旋转，在向该投影面作正投影，即可得到立体正轴测图。通常选 V 面为轴侧投影面，所以将立体图绕 Z 轴正向（逆时针方向）旋转 θ 角，再绕 X 轴反向（顺时针方向）旋转 φ 角，最后向 V 面正投影。因此将绕 Z 轴旋转变换矩阵 T_z，绕 X 轴旋转变

换矩阵 T_x 和向 V 面正投影变换矩阵 T_v 连乘，即可得到正轴侧变换矩阵：

$$T_E = T_z \cdot T_x \cdot T_v = \begin{bmatrix} \cos f & 0 & -\sin f \cos q & 0 \\ -\cos f & 0 & -\cos q \sin f & 0 \\ 0 & 0 & \cos f & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

（1）正等侧投影为正轴侧投影中 x、y、z 三个方向上缩放率相等时的变换，即 $q = 45^0, j = 35^0 16'$，变换矩阵为：

$$T_{正等} = \begin{bmatrix} 0.707 & 0 & -0.408 & 0 \\ -0.707 & 0 & -0.408 & 0 \\ 0 & 0 & 0.816 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

（2）正二侧投影为正轴侧投影中 x、z 三个方向上缩放率相等时的变换，即为：$q=20^{0}42',j=19^{0}28'$时，变换矩阵：

$$T_{正二}=\begin{bmatrix} 0.935 & 0 & -0.118 & 0 \\ -0.354 & 0 & -0.312 & 0 \\ 0 & 0 & 0.943 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5.3.2.2 斜平行投影

投影方向不垂直于投影平面的平行投影被称为斜平行投影，如图 **5-4** 所示，假设 **Z=0** 的坐标平面为观察平面（**H** 面），点（**x**，**y**）为点（**x**，**y**，

z）在观察平面上的正平行投影坐标，点（**x**′，**y**′）为斜投影坐标。（**x**，**y**）与（**x**′，**y**′）的距离为 **L**。

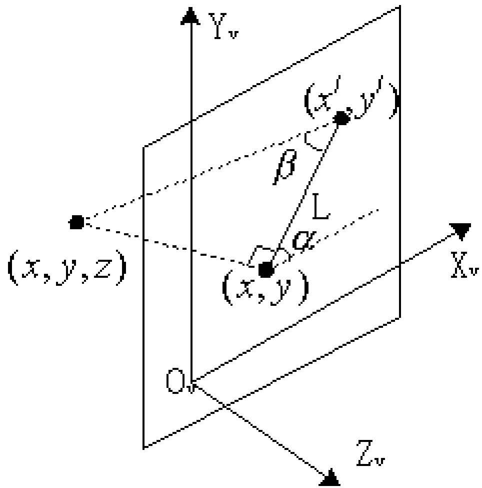


图 5-4 斜平行投影示意图

显然，$x' = x + L\cos\alpha \quad y' = y + L\sin\alpha$

而 **L** 的长度依赖于 z, β **,** 即 $tg\beta = z / L$， $L = z / tg\beta$

所以 $x' = x + z\dfrac{1}{tg\beta}\cos\alpha \quad y' = y + z\dfrac{1}{tg\beta}\sin\alpha$

令 $l_1 = \dfrac{1}{tg\beta}$，则 $x' = x + zl_1\cos\alpha \quad y' = y + zl_1\sin\alpha$，

由此可得斜平行投影的变换矩阵：

$$[x' \ y' \ z' \ 1] = [x \ \ y \ \ z \ \ 1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ l_1\cos a & l_1\sin a & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

(1)斜等测平行投影，$l_1 = \frac{1}{tgb} = 1, b = 45^0$

(2)斜二测平行投影，$l_1 = \frac{1}{tgb} = \frac{1}{2}, b = tg^{-1}a$

其中 β 为投影线与投影平面的夹角，α 为投影与 X 轴的夹角。

5.3.2.3 透视投影

透视投影的视线（投影线）是从视点（观察点）出发，视线是不平行的。不平行于投影平面的视线汇聚的一点称为灭点，在坐标轴上的灭点叫做

主灭点。主灭点数和投影平面切割坐标轴的数量相对应。按照主灭点的个数，透视投影可分为一点透视、二点透视和三点透视，如图 **5-5** 所示。

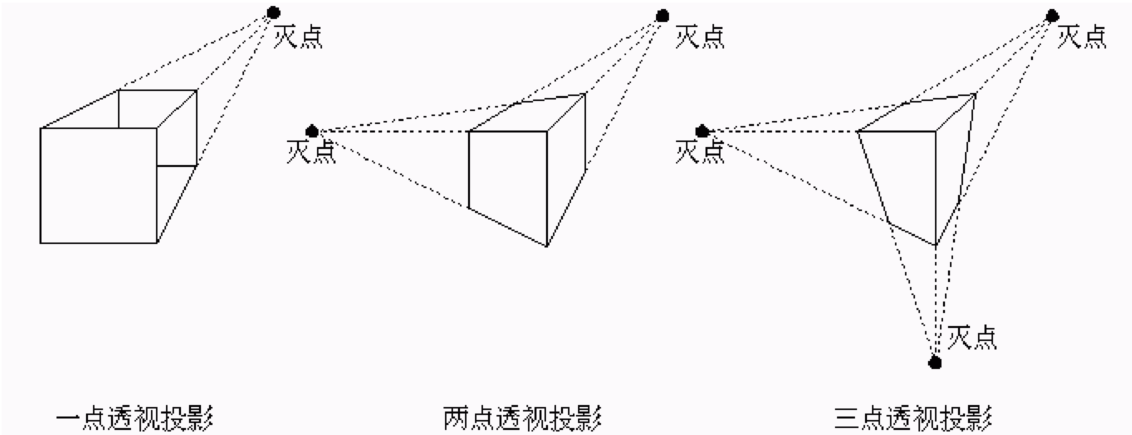


图 **5-5** 透视投影示意图

1. 一点透视

Z 轴上有一个观察点 V（0，0，h），由 V 点出发将物体的点 P（x，y，z）投影到 XOY 平面上得到(X，Y，Z)。

变换矩阵：灭点在 Z 轴上（0，0，-h），

$$M_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{h} & 1 \end{bmatrix}$$

灭点在 X 轴上（-h，0，0），

$$M_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{1}{h} & 0 & 0 & 1 \end{bmatrix}$$

灭点在 **Y** 轴上（**0**，**-h**，**0**），

$$M_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -\frac{1}{h} & 0 & 1 \end{bmatrix}$$

M_x, M_y, M_z 均称为一点变换，可得到一点透视（以灭点在 **Z** 轴上为例）有

$$\begin{bmatrix} X \\ Y \\ Z \\ H \end{bmatrix} = M_z \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$，其中 $H = 1 - \dfrac{z}{h}$

2. 二点透视

在变换矩阵中的四行的三个参数起透视变换作用。

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p & q & r & 1 \end{bmatrix}, \quad X' = MX$$

若 **p**，**q** 二个参数不为零，则即可得到二点透视，

$$\begin{bmatrix} X \\ Y \\ Z \\ H \end{bmatrix} = M_{pq} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ pX + qY + 1 \end{bmatrix}$$

若固定 **Y**，**Z** 令 $X \to \infty$，则得灭点为 $(\frac{1}{p},0,0)$；

若固定 **X**，**Z** 令 $Y \to \infty$，则得另一灭点为 $(0,\frac{1}{q},0)$，

即坐标轴上有两个灭点，因而称为二点透视。

3.　三点透视

同理，可以讨论由三个主灭点的透视变换，其变换矩阵为

$$M=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\p&q&\gamma&1\end{bmatrix},$$

三个灭点分别为 $(\frac{1}{p},0,0),(0,\frac{1}{q},0),(0,0,\frac{1}{\gamma})$ 。

透视投影技巧：实际进行透视投影时，为了获得理想的透视投影图，往往先对物体进行旋转和平易，然后再进行透视投影变换。

5.4 三维变换程序设计案例

一、程序设计功能说明

本程序为三维几何变换的实用程序，可实现以上介绍的种三维几何图形变换的绘制。如图 **5-7** 所示为程序运行时的主界面，通过单击菜单及下拉菜单的各功能项完成分别各种几何变换。

说明：为了使变换结果得更清楚、直观，程序中的几何变换和透视图是对正方体进行的；而投影变换中的平行变换是对类似“椅子”图形进行的变换。

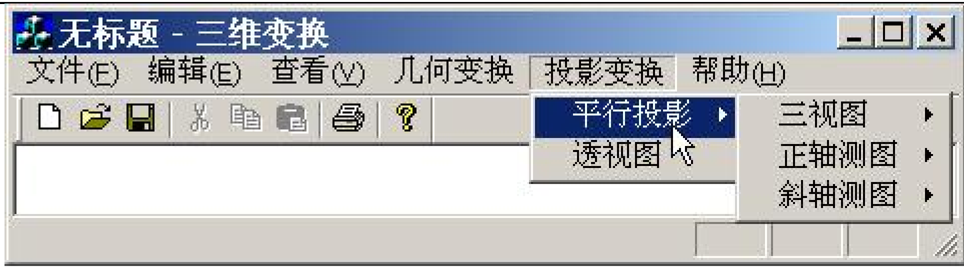


图 **5-7**

二、程序设计步骤

1. 创建工程名称为“三维变换”单文档应用程序框架

2. 编辑菜单资源

根据 **5.1** 表中的定义编辑菜单资源，设计如图 **5-7** 所示的菜单项。

表 **5.1** 菜单资源表

基本图形变换	平移			ID_TRANSLATION
	旋转	绕 X 轴		ID_ROTATION_X
		绕 Y 轴		ID_ROTATION_Y
		绕 Z 轴		ID_ROTATION_Z
	变比	沿 XYZ 变比		ID_SCALING_XYZ
		整体变比		ID_SCALING_S
	对称	关于 X 轴对称		ID_MIRROR_X
		关于 Y 轴对称		ID_MIRROR_Y
		关于 Z 轴对称		ID_MIRROR_Z
		关于 OXY 平面对称		ID_MIRROR_OXY
		关于 OYZ 平面对称		ID_MIRROR_OYZ
		关于 OZX 平面对称		ID_MIRROR_OZX
		关于原点对称		ID_MIRROR_O
		三视图	主视图	ID_V
			俯视图	ID_H
			侧视图	ID_W
		正轴测图	正等测图	ID_VE

3. 添加消息处理函数

利用 **ClassWizard**（建立类向导）为应用程序添加与菜单项相关的消息处理函数，**ClassName** 栏中选择 **CMyView**，根据表 **5.2** 建立如下的消息映射函数。

表 **5.2** 菜单项的消息处理函数

菜单项 ID	消息	消息处理函数
ID_TRANSLATION	CONMMAN	OnIDTRANSLATION
ID_ROTATION	CONMMAN	OnIDROTATION
ID_SCALING	CONMMAN	OnIDROTATION
D_MIRROR_X	CONMMAN	OnIDMIRRORX
ID_MIRROR_Y	CONMMAN	OnIDMIRRORY
ID_MIRROR_O	CONMMAN	OnIDMIRRORO
ID_SCALINGXY	CONMMAN	OnIDSCALINGXY
ID_ROTATIONXY	CONMMAN	OnIDROTATIONXY

4. 添加三维变换绘图基类

基类 **BaseClass** 类。在工程中单击【文件】|【新建】，在弹出的新建对话框中，选择 **C/C++ Header File**，在【文件】名称输入栏中输入“**BaseClass**”；同样，在工程中单击【文件】|【新建】，在弹出的新建对话框中，选择 **C++ Source File**，在【文件】名称输入栏中输入“**BaseClass**”。在工作区中系统自动创建的相应的空文件中，分别添加以下此基类的头文件（**.h** 文件）和应用文件**(.cpp** 文件**)**。

```
// BaseClass.h: interface for the CMyClass class.
//
//////////////////////////////////////////////////////////////////////////

#if !defined(AFX_BaseCLASS_H__6250EB80_113B_11D4_81FF_D19FE195501C__INCLUDED_)
```

```
#define
AFX_BaseCLASS_H__6250EB80_113B_11D4_81FF_D19FE195501C__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
#define      Scale        1.35

typedef  double   array2d[5][5];
typedef  double   array[24];

/************************************************************
* 变量说明：Aux1, Aux2, Aux3, Aux4, Aux5, Aux6, Aux7,
* Aux8 是全局变量，用于存取计算用户坐标系点到观察坐标系点的 *
* 坐标值公式中的正余弦值。X, Y, Z, C, XP, YP, ZP, CP 为一维 *
```

```
* 数组，存放立体顶点齐次坐标，XT，YT，ZT 亦为一维数组，存放 *
* 立体顶点经变换后的坐标值。A，Ah，Aw 二维数组用来接收轴测  *
* 图的变换矩阵与三视图的变换矩阵参数值。                   *
**************************************************************/

class CBaseClass  // 定义一个基类
{
```

程序代码见纸书

```
#endif
// !defined(AFX_MYCLASS_H__6250EB80_113B_11D4_81FF_D19FE195501C__INCLUDED_)
```

说明：

```
* 变量说明：Aux1，Aux2，Aux3，Aux4，Aux5，Aux6，Aux7，  *
* Aux8 是全局变量，用于存取计算用户坐标系点到观察坐标系点的 *
```

```
* 坐标值公式中的正余弦值。X, Y, Z, C, XP, YP, ZP, CP 为一维 *
* 数组，存放立体顶点齐次坐标，XT, YT, ZT 亦为一维数组，存放 *
* 立体顶点经变换后的坐标值。A, Ah, Aw 二维数组用来接收轴测  *
* 图的变换矩阵与三视图的变换矩阵参数值。                    *
// BaseClass.cpp: implementation of the CMyClass class.
//
//////////////////////////////////////////////////////////////////////
////

#include "stdafx.h"
#include "三维变换.h"
#include "BaseClass.h"
#include "三维变换View.h"
#include "math.h"
```

```
#define PI 3.141592654

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif


//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////
```

```
CBaseClass::CBaseClass()
{
 ed=2000,eh=100,od=400,hl=1,ps=0;
}

CBaseClass::~CBaseClass()
{


}

// 此函数赋轴测图中立体上顶点的齐次坐标值
void   CBaseClass::ReadWorkpiece()
{
```

```
/*****************************************************
* 此函数分别用于三个视图的投影变换，统一用变换后顶 *
* 点的三个坐标计算公式求其坐标值。这三个公式是由点 *
* 的齐次坐标乘以变换矩阵得来的。实际上每个视图投影 *
* 只有二个非零坐标需要计算求得，而另一个坐标是零勿 *
* 需计算。因此也可以根据三个不同视图的投影，分别采 *
* 用其投影后的二个坐标计算式来求坐标值。函数中可用 *
* 条件语句选择不同视图投影采用不同坐标计算式求值。 *
*****************************************************/
void   CBaseClass::Calculate(array2d B)
{
  程序代码见纸书
}
```

代码说明：

以上 **CMyClass** 基类中 **DrawView**（）函数考虑用户坐标到屏幕坐标的变

换，其中水平像素与垂直像素比例为 **1**。**ReadWorkpiece()**函数给图形的顶点赋予齐次坐标值。**Calculate()**函数计算原图形中顶点的齐次矩阵与变换矩阵的相乘，并存储相乘的结果。**DrawView ()**绘制原图形与变换后的图形。**Display ()**在屏幕上显示图形

5. 在几何图形变换 **View.h**、几何图形变换 **View.cpp** 添加完成各个菜单消息处理函数，实现既定功能，

```
// 三维变换View.h : interface of the CMyView class

// 三维变换View.h : interface of the CMyView class
//
////////////////////////////////////////////////////////////////////////////////
```

```
#if !defined(AFX_VIEW_H__BFB29846_C242_48D0_8CA7_60B59B46A728__INCLUDED_)
#define AFX_VIEW_H__BFB29846_C242_48D0_8CA7_60B59B46A728__INCLUDED_
```

程序代码见纸书// RedrawWindow();

```
}
```

5.5　课后练习

1. 请写出三维几何变换矩阵,并说明各功能子矩阵作用。
2. 给定一个单位立方体，一个顶点在（0，0，0），相对另一顶点在（1，1，1），将单位立方体绕过此两点所连直线旋转θ角，求变换矩阵；

3. 编程实现将上述单位矩阵作平移、缩放、和旋转变换。

4. 设三棱锥各点坐标为（0，0，20），（20，0，20），（20，0，0），（10，20，10），编程实现三面正投影图；

5. 编程实现一三棱锥的正等轴测和正二测图形。