第3章 几何变换

CHAPTER 3

3.1 仿射变换

变换是一个函数，其把一个点（或向量）映射成另一个点（或向量）。如图 3.1 所示，$Q=T(P)$是点的变换，而$\boldsymbol{v}=T(\boldsymbol{u})$是向量的变换。

如果使用齐次坐标，那么就能把向量和点都表示成四维列矩阵，因此点和向量的变化可以用同一个函数来定义。

如果 f 是线性函数，即对于任何标量 α 和 β 以及任何两个顶点（或者向量）$\boldsymbol{p}$ 和 $\boldsymbol{q}$，都有 $f(\alpha\boldsymbol{p}+\beta\boldsymbol{q})=\alpha f(\boldsymbol{p})+\beta f(\boldsymbol{q})$，这样的变换就称为仿射变换。因为变换的线性组合等于线性组合的变换，所以只需要知道 $\boldsymbol{p}$ 和 $\boldsymbol{q}$ 的变换，就可以求出它们的线性组合的变换，这样就无须计算每个线性组合的变换。

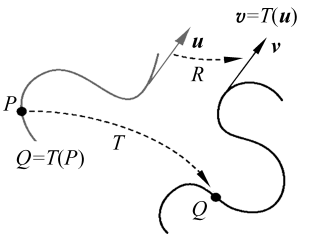


图 3.1 仿射变换

由于变换是将点（或向量）由一种形式转换为另外一种形式，因此当用齐次坐标表示变换的点（或向量）时，点（或向量）可以用一维行矩阵或者一维列矩阵表示：

$$\boldsymbol{p}=\begin{bmatrix}\beta_1\\ \beta_2\\ \beta_3\\ 1\end{bmatrix},\quad \boldsymbol{u}=\begin{bmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\\ 0\end{bmatrix}$$

因此，变换也就可以表达为矩阵乘法的形式$\boldsymbol{v}=\boldsymbol{C}\boldsymbol{u}$，其中 $\boldsymbol{C}$ 为 4×4 的矩阵，形式如下：

$$\boldsymbol{C}=\begin{bmatrix}\alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14}\\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24}\\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34}\\ 0 & 0 & 0 & 1\end{bmatrix}$$

从该矩阵可以看出，点的变换有 12 个自由度，而向量的变换只有 9 个自由度。

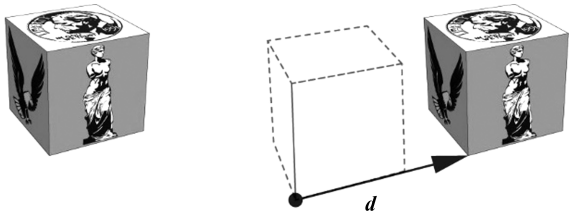
仿射变换将直线映射为直线。因为 $p(\alpha)=p_0+\alpha\boldsymbol{d}$，$\boldsymbol{C}p(\alpha)=Cp_0+\alpha\boldsymbol{C}\boldsymbol{d}$，所以为了确定变换后的线段，只需对两个端点进行变换即可。因此，可以用流水线模型来实现图形系统，流水线中的仿射变换单元只需对端点进行变换，在光栅化阶段根据端点的处理结果就可以生成内部的点。计算机图形学中用到的大多数变换是仿射变换，包括旋转、平移（translate，

放置）和缩放。

3.2 平移、旋转和缩放

3.2.1 平移

平移变换就是把点沿着给定的方向移动固定距离，如图 3.2 所示。



(a) 原始位置　　(b) 移动$\boldsymbol{d}$后的位置

图 3.2　平移变换

因此，平移变换只需要给定平移向量即可，即 $\boldsymbol{P}'=\boldsymbol{P}+\boldsymbol{d}$。平移变换有 3 个自由度，即平移向量 3 个坐标轴方向的分量。如果用齐次坐标表示，则有：

$$\boldsymbol{P}=[x\ y\ z\ 1]^{\mathrm{T}}$$
$$\boldsymbol{P}'=[x'\ y'\ z'\ 1]^{\mathrm{T}}$$
$$\boldsymbol{d}=[d_x\ d_y\ d_z\ 0]^{\mathrm{T}}$$

因此有：

$$x'=x+d_x$$
$$y'=y+d_y$$
$$z'=z+d_z$$

平移变换也可用 4×4 矩阵 $\boldsymbol{T}$ 来表示，即 $\boldsymbol{P}'=\boldsymbol{T}\boldsymbol{P}$。其中：

$$\boldsymbol{T}=\boldsymbol{T}(d_x,d_y,d_z)=\begin{bmatrix}1&0&0&d_x\\0&1&0&d_y\\0&0&1&d_z\\0&0&0&1\end{bmatrix}$$

表达式 $\boldsymbol{P}'=\boldsymbol{T}\boldsymbol{P}$ 是仿射变换的通用表达，并且多个变换可以叠加。

平移的物体再移回原位置，就得到了平移的逆变换，其变换矩阵为

$$\boldsymbol{T}^{-1}(d_x,d_y,d_z)=\boldsymbol{T}(-d_x,-d_y,-d_z)=\begin{bmatrix}1&0&0&-d_x\\0&1&0&-d_y\\0&0&1&-d_z\\0&0&0&1\end{bmatrix}$$

3.2.2 旋转

1. 二维旋转

如图 3.3 所示，假设点(x,y)基于原点旋转 θ 到达点(x',y')，半径 r 不变。由于点$(x,$

y)可看成由 x 轴旋转 ϕ 得到，即 $x=r\cos\phi$，$y=r\sin\phi$；而点(x'，y') 可看成由 x 轴旋转 $\phi+\theta$ 得到，即 $x'=r\cos(\phi+\theta)$，$y'=r\sin(\phi+\theta)$，展开三角函数，则有：

$$x'=r\cos(\phi+\theta)=r\cos\phi\cos\theta-r\sin\phi\sin\theta=x\cos\theta-y\sin\theta$$

$$y'=r\sin(\phi+\theta)=r\cos\phi\sin\theta+r\sin\phi\cos\theta=x\sin\theta+y\cos\theta$$

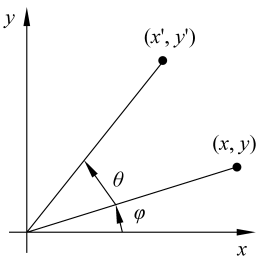


图 3.3　旋转变换

如果点(x，y)和点(x'，y')用齐次坐标表示，二维旋转变换写成矩阵形式，则有：

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

$R_\theta=\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$ 即为二维旋转变换的变换矩阵。

2. 三维旋转

由于二维点可以看成 z 值为 0 的三维点，因此二维旋转所有点的 z 值不变，这与在三维中做关于 z 轴的旋转操作等价，则将点(x，y，0)基于原点绕 z 轴旋转 θ 到达点(x'，y'，0)变换的矩阵形式为 $\boldsymbol{P}'=\boldsymbol{R}_{Z(\theta)}\boldsymbol{P}$。其中：

$$\boldsymbol{R}_{Z(\theta)}=\begin{bmatrix}\cos\theta & -\sin\theta & 0 & 0\\ \sin\theta & \cos\theta & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

与关于 z 轴的旋转类似，关于 x 轴旋转，x 不变，变换矩阵为

$$\boldsymbol{R}_{X(\theta)}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \cos\theta & -\sin\theta & 0\\ 0 & \sin\theta & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

与关于 z 轴的旋转类似，关于 y 轴旋转，y 不变，变换矩阵为

$$\boldsymbol{R}_{Y(\theta)}=\begin{bmatrix}\cos\theta & 0 & \sin\theta & 0\\ 0 & 1 & 0 & 0\\ -\sin\theta & 0 & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

与平移变换的逆变换类似，旋转变换的逆变换也是从变换后的位置旋转回原来的位置，所以有：

$$\boldsymbol{R}^{-1}\theta=\boldsymbol{R}(-\theta)$$

由于有：

$$\sin(-\theta)=-\sin\theta,\quad \cos(-\theta)=\cos\theta$$

可推知：

$$\boldsymbol{R}^{-1}\theta=\boldsymbol{R}^{\mathrm{T}}\theta$$

3.2.3 缩放

与平移变换和旋转变换不同，缩放变换是一类非刚性的变换，其可使对象变大或者缩小，如图 3.4 所示。

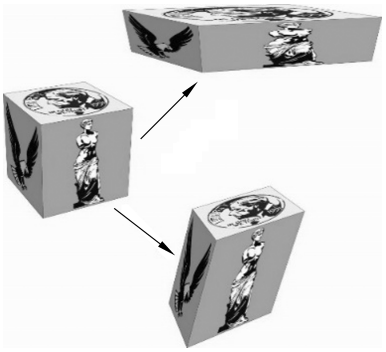


图 3.4 缩放变换

与旋转变换一样，缩放变换需要针对某一定点进行，为方便起见，该定点一般选择为原点。各坐标轴方向变换的比例可相同，也可不同。缩放变换的表达式为

$$x' = \beta_x x$$
$$y' = \beta_y y$$
$$z' = \beta_z z$$

将此 3 个公式合在一起，可表示为

$$\boldsymbol{P}' = \boldsymbol{S}\boldsymbol{P}$$

其中：

$$\boldsymbol{S} = \boldsymbol{S}(\beta_x, \beta_y, \beta_z) = \begin{bmatrix} \beta_x & 0 & 0 & 0 \\ 0 & \beta_y & 0 & 0 \\ 0 & 0 & \beta_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

缩放变换的逆变换是按缩放因子倒数进行缩放，因此有：

$$\boldsymbol{S}^{-1}(\beta_x, \beta_y, \beta_z) = \boldsymbol{S}\left(\frac{1}{\beta_x}, \frac{1}{\beta_y}, \frac{1}{\beta_z}\right)$$

所以：

$$\boldsymbol{S}^{-1}(\beta_x, \beta_y, \beta_z) = \begin{bmatrix} \dfrac{1}{\beta_x} & 0 & 0 & 0 \\ 0 & \dfrac{1}{\beta_y} & 0 & 0 \\ 0 & 0 & \dfrac{1}{\beta_z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.2.4　反射变换

反射变换是缩放变换在特殊比例下的变换，当 $\beta_x=-1$、$\beta_y=1$、$\beta_z=1$ 时，得到相对于 y 轴的反射图形；当 $\beta_x=1$、$\beta_y=-1$、$\beta_z=1$ 时，得到相对于 X 轴的反射图形；当 $\beta_x=-1$、$\beta_y=-1$、$\beta_z=1$ 时，得到相对于原点的反射图形，如图 3.5 所示。

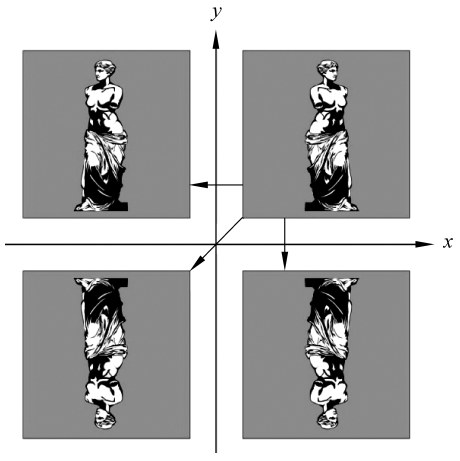


图 3.5　反射变换

3.2.5　错(剪)切变换

错(剪)切变换也是非刚性变换，是指对象在某一方向发生了错(剪)切，如图 3.6 所示。

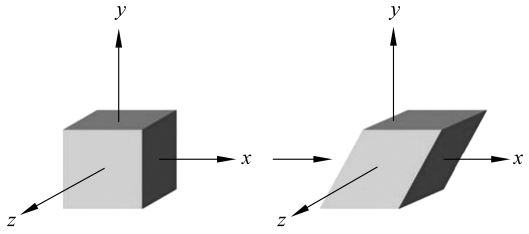


图 3.6　错(剪)切变换

以 x 方向的错切为例，对象变化如图 3.7 所示，可以看到，错切变换和错切角 θ 有关，变换式为

$$x'=x+y\cot\theta$$
$$y'=y$$
$$z'=z$$

则可得到错切变换矩阵为

$$\boldsymbol{H}_x(\theta)=\begin{bmatrix}1 & \cot\theta & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

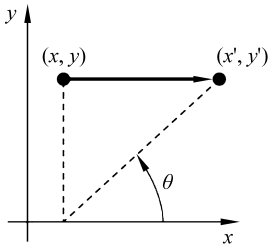


图 3.7　x 方向错切

其逆变换为

$$\boldsymbol{H}_x^{-1}(\theta)=\boldsymbol{H}_x(-\theta)$$

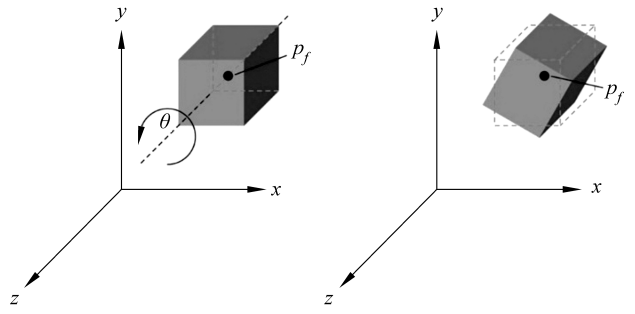
3.3　变换的级联（联合）

基于以上基本变换可以看到，通过矩阵相乘（级联），可以将这些基本变换组合起来。

假设对 p 执行了 A、B、C 3 个变换得到 q，因为矩阵乘法满足结合律，所以变换序列可以表示为 $q=CBAp=C(B(Ap))$。其中，右边的矩阵是首先被应用的矩阵，变换的顺序是不可交换的。如果要对多点进行变换，可以首先计算 $M=CBA$，然后对每个点计算$q=Mp$。

3.3.1　基于任一定点的旋转变换（绕 z 轴旋转）

如图 3.8 所示，需要对立方体基于定点 p_f 绕平行于 z 轴的直线旋转 θ 角。



(a) 旋转前　　(b) 旋转后

图 3.8　基于定点的旋转变换

由于基本旋转变换只能基于原点进行，因此将此变换的策略定为：①将定点移动至原点；②旋转；③将定点移动到原来的位置，如图 3.9 所示。

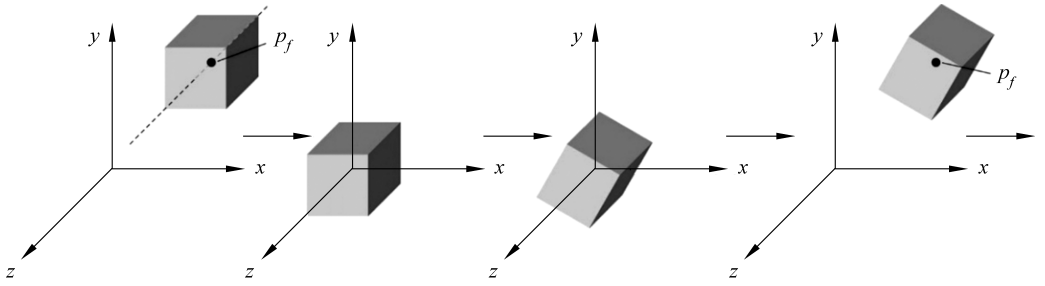


图 3.9　基于定点的旋转变换分解

因此，变换矩阵表示为

$$\boldsymbol{M}=\boldsymbol{T}(P_f)\boldsymbol{R}_z(\theta)\boldsymbol{T}(-P_f)$$

式中，$\boldsymbol{T}(-P_f)$为将 P_f 移动至原点；$\boldsymbol{R}_z(\theta)$为基于原点绕 z 轴旋转 θ 角；$\boldsymbol{T}(P_f)$为将原点移动至 P_f。

经计算，变换矩阵为

$$\boldsymbol{M}=\begin{bmatrix}\cos\theta & -\sin\theta & 0 & x_f-x_f\cos\theta+y_f\sin\theta\\ \sin\theta & \cos\theta & 0 & y_f-x_f\sin\theta-y_f\cos\theta\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

3.3.2　实例变换

由复合变换可以引申出一种建模方法，即一次定义一类对象，对象的大小、位置和方向可以视如何更有利于对象建模而定。对象在场景中的每一次出现都是这种对象原型的一个实例，图 3.10 为立方体、球体、四面体通过实例变换后得到的实体。

实例变换过程一般如图 3.11 所示：首先定义原始大小的对象，再根据需要依次进行缩放、旋转和平移变换，即可得到最终的实例，变换矩阵即为 $\boldsymbol{M}=\boldsymbol{TRS}$。

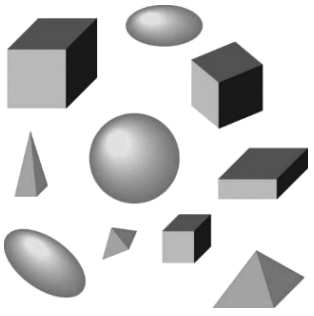


图 3.10　实例变换示例

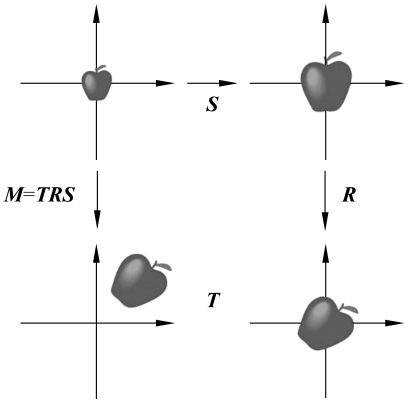


图 3.11　实例变换过程

从前文可以看到，平移、旋转、缩放甚至错切等基本变换以及复合变换都用 4×4 矩阵表示，因此可用 4×4 矩阵操作统一表示二维和三维几何变换。该矩阵可分为 1×1、3×1、1×3、3×3 4 个小矩阵，分别对应整体缩放、平移、投影以及缩放、旋转、反射、错切等变换，如图 3.12 所示。

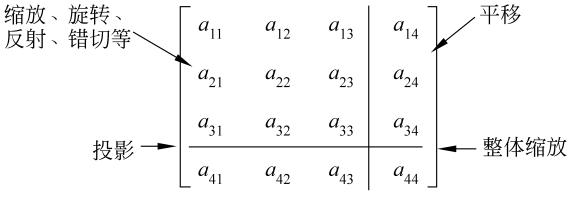


图 3.12　变换的一般矩阵形式

3.4 窗口到视区的坐标变换

窗口到视区的坐标变换也称为视口变换。

1. 窗口与视区的概念

如图 3.13 所示，窗口是在二维世界坐标系或观察坐标系的观察平面（一个与观察坐标系坐标平面平行的投影面，也称画面）中定义的一个矩形区域，只有在该区域内的图形才能在设备坐标系下输出，而窗口外的部分则被裁掉。视区是在屏幕坐标系（是设备坐标系）中定义的一个矩形区域，用于输出窗口中的图形。视区决定了窗口中的图形要显示于屏幕上的位置和大小，其是一个有限的整数域，应小于或等于屏幕区域。可以在同一屏幕上定义多个视区，用来同时显示不同的图形。一般来说，窗口图形坐标值为实数，而视区图形坐标值为正整数。通常人们在世界坐标系中描述图形时，为了把所描述或观察的图形全部或部分地显示在屏幕上，必须将世界坐标或观察坐标变换为显示设备的屏幕坐标。变换时需要在世界坐标系或观察平面中定义一个平行于坐标轴的窗口，框住自己感兴趣的图形区域，并映射到屏幕视区中显示。

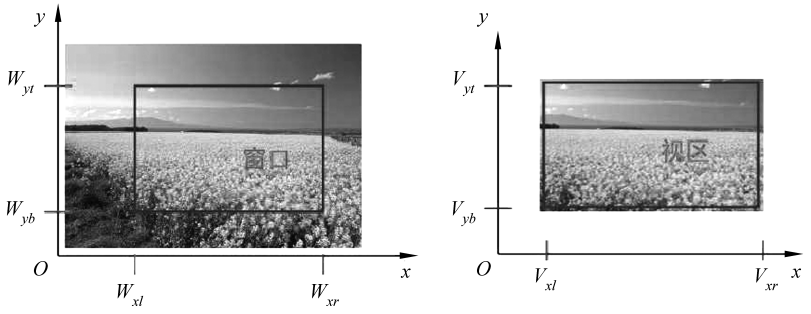


图 3.13 窗口与视区

2. 窗口-视区变换

由于窗口和视区在不同的坐标系中定义，因此把窗口中的图形信息输出到视区之前需要进行坐标变换，即把用户坐标系的坐标值转化为设备（屏幕）坐标系的坐标值，该变换过程为窗口-视区变换。窗口与视区可分别用其 4 条边界的坐标来表示，如图 3.14 所示。

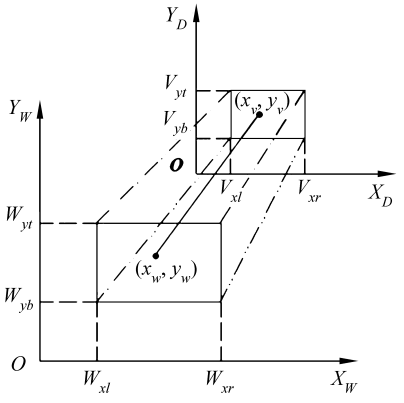


图 3.14 窗口-视区变换

设在用户坐标系下定义的窗口如下：左下角点坐标(W_{xl},W_{yb})，右上角点坐标(W_{xr},W_{yt})；在设备坐标系下定义的视区如下：左下角点坐标(V_{xl},V_{yb})，右上角点坐标(V_{xr},V_{yt})。将用户坐标系中的点(x_w,y_w) 映射到设备坐标系中的点(x_v,y_v)，有下列等式：

$$\begin{cases}\dfrac{x_v-V_{xl}}{V_{xr}-V_{xl}}=\dfrac{x_w-W_{xl}}{W_{xr}-W_{xl}}\\[2ex]\dfrac{y_v-V_{yb}}{V_{yt}-V_{yb}}=\dfrac{y_w-W_{yb}}{W_{yt}-W_{yb}}\end{cases}$$

因此，窗口中一点(x_w,y_w)变换到视区中的对应点(x_v,y_v)，二者之间的关系为

$$\begin{cases}x_v=\dfrac{V_{xr}-V_{xl}}{W_{xr}-W_{xl}}(x_w-W_{xl})+V_{xl}\\[2ex]y_v=\dfrac{V_{yt}-V_{yb}}{W_{yt}-W_{yb}}(y_w-W_{yb})+V_{yb}\end{cases}$$

若令：

$$\begin{cases}a=\dfrac{V_{xr}-V_{xl}}{W_{xr}-W_{xl}},b=V_{xl}-\dfrac{V_{xr}-V_{xl}}{W_{xr}-W_{xl}}W_{xl}\\[2ex]c=\dfrac{V_{yt}-V_{yb}}{W_{yt}-W_{yb}},d=V_{yb}-\dfrac{V_{yt}-V_{yb}}{W_{yt}-W_{yb}}W_{yb}\end{cases}$$

则有：

$$\begin{cases}x_v=ax_w+b\\y_v=cy_w+d\end{cases}$$

写成矩阵形式有：

$$\begin{bmatrix}x_v\\y_v\\1\end{bmatrix}=\begin{bmatrix}a&0&b\\0&c&d\\0&0&1\end{bmatrix}\begin{bmatrix}x_w\\y_w\\1\end{bmatrix}$$

由此可见，窗口-视区变换是比例变换与平移变换的组合。

具体变换时，应使视区的宽高比与窗口的宽高比保持一致，即要求 $a=d$，以免输出的图形变形失真。其方法是从计算出的 a、d 中取较小者 $s=\min\{a,d\}$，令 $a=d=s$。

3.5　虚拟跟踪球

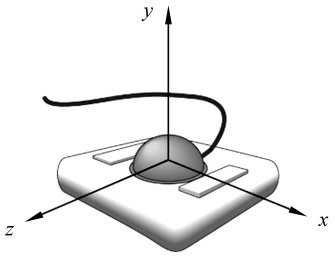
前面介绍的几何变换除了对三维对象进行变换外，也为人们观察和操作三维对象提供了接口。然而，由于屏幕是平面的，现有的鼠标和键盘是针对二维设计的，因此并不能直接提供一个友好的三维应用程序接口。有两种方法可以构造一个友好的三维应用程序接口：一种是在屏幕区域利用鼠标指针位置来控制两个轴的旋转；一种是利用鼠标和显示器生成一个虚拟跟踪球。

和物理跟踪球一样，虚拟跟踪球可用来支持对象的连续旋转，而且可以改变旋转的速度和旋转轴的方向。

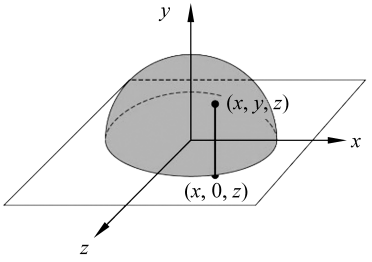
根据跟踪球的特点，可以建立图 3.15(a)所示的坐标系，并假定该跟踪球的半径为 1 个长度单位。这样，半球面和 xz 平面的圆存在一一对应关系，给定平面上一点(x,0,z)，即可

算出在半球面上的位置(x,y,z)，其中$y=\sqrt{1-x^2-z^2}$，这样就可通过鼠标移动来计算和追踪三维信息如，图 3.15(b)所示。

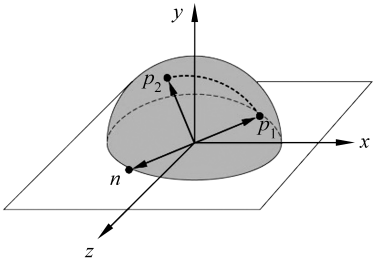
假定在半球面上有两个位置$\boldsymbol{p}_1$和$\boldsymbol{p}_2$，如图 3.15(c)所示，则从原点到这两个位置的两个向量确定了一个平面，且该平面的法向量为$\boldsymbol{n}=\boldsymbol{p}_1\times\boldsymbol{p}_2$。



(a) 跟踪球坐标系



(b) 在平面上的投影



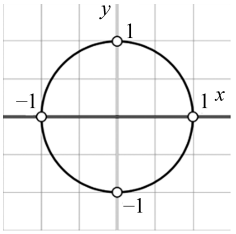
(c) 旋转点的计算

图 3.15　虚拟跟踪球

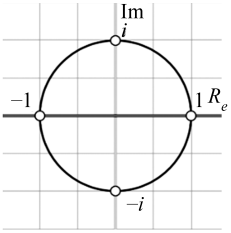
从$\boldsymbol{p}_1$到$\boldsymbol{p}_2$可看成$\boldsymbol{p}_1$绕$\boldsymbol{n}$旋转了一定的角度到达了$\boldsymbol{p}_2$，该角度即为两个向量之间的夹角$\sin\theta=(\boldsymbol{p}_1\times\boldsymbol{p}_2)/(|\boldsymbol{p}_1||\boldsymbol{p}_2|)$。由于半球面的半径为 1，有$|\boldsymbol{p}_1|=|\boldsymbol{p}_2|=1$，因此有$|\sin\theta|=|\boldsymbol{p}_1\times\boldsymbol{p}_2|=|\boldsymbol{n}|$。如果采样率较高，从$\boldsymbol{p}_1$到$\boldsymbol{p}_2$变化较小，一般可采用近似量$\sin\theta=\theta$，以减少计算量。

3.6　四元数

如果把 2D 网格平面的横轴看成实轴，纵轴看成虚轴，则可以得到复数平面。这样，只需要把实数映射到横轴，虚数映射到纵轴，就可以把复数映射到一个 2D 网格平面中，如图 3.16 所示。



(a) 2D网格平面



(b) 复数平面

图 3.16　复数平面的表示