4 几何对象和变换

4.1 标量、点和向量

4.1.1 几何对象

点(point)

点是最基本的几何对象。点所具有的唯一性质就是它的空间位置。数学上的点既没有大小，也没有形状。

标量

我们可以用点来定义几何对象，但是仅有点还不够。需要用实数来确定数量，例如两点之间的距离。实数是标量的例子，因此我们需要进入标量。

向量

为了能够处理方向，还需要另外一种类型，：**向量**，在图形学中即为**有向线段**。

向量的加法

三角形法则，**和向量**，**零向量**，**向量的逆**

向量的乘法

增加或者减小向量的长度而不改变方向

点-向量加法

点 + 向量 = 点

点-点减法

任何两个点都定义了一条有向线段(向量）。点-点 = 向量

4.1.2 与坐标无关的几何

定义点或者向量并不需要坐标系。

4.1.3 数学的观点：向量空间和仿射空间

如果把标量、点和向量看成是某些数学几何中的元素，那么让我们来看看各种各样的抽象空间，这些抽象空间是用来表示和处理这三类对象的。我们仅仅感兴趣空间中的元素都是几何类型的空间——— **向量空间，仿射空间，欧式空间**

向量空间(线性空间)

包含的实体

向量和标量

定义的运算

标量之间的运算

标量-向量乘法——>向量

向量-向量加法—>向量

例子

实数n元组

有向线段

欧式空间

向量空间中不一定有度。欧式空间是向量空间的扩展，它增加了对距离的度量，可以定义线段的长度等概念。欧式空间中线段的大小就是它的长度。

仿射空间

是向量空间的扩展

实体

标量、向量、点

运算

向量-点加法—>点

点-点减法运算——>向量

在这些抽象空间中，对象的定义不依赖于任何特定的表示，对象只是各种集合中的元素。向量空间的主要概念之一是：通过一组基向量来表示向量，这种表示把抽象的对象和它们的实现关联在一起。在不同的表示之间可以进行转换，这就引出了集合变换。

4.1.4 计算机科学的观点

计算机更愿意把几何类型看成是抽象数据类型(Abstract Data Type),ADT是在数据上定义的一些运算，这些运算的定义不依赖于数据在计算机内部的表示方式和运算的具体实现形式。

4.1.5 集合ADT

标量、点、向量遵循仿射空间的运算规则

4.1.6 直线

从点和向量之和(或者两个点之差)可以引出仿射空间中直线的概念。

直线的参数形式

$P(\alpha) = P_0 + \alpha d$

P_0是任意的一个点，d是任何一一个向量，α为任意值

射线

当α只能取非负值的时候，则就是一条射线，当然也可以取非正值。

4.1.7 仿射加法(仿射和)

$P = \alpha R + (1-\alpha)Q$

R, Q, P都是点。当$\alpha \in [0,1]$时，P是线段RQ上的点。

我们可以将放射和的概念扩展到n个点，$P_1, P_2, P_3, ..., P_n$定义的对象。

$P = \alpha_1 P_1 + \alpha_2 P_2 + \alpha_3 P_3 + ... + \alpha_n P_n$ 定义域为$\alpha_1 + \alpha_2 + \alpha_3 + ... + \alpha_n = 1$

4.1.8 凸性

如果对于一个对象中的任意两个点，连接它们的线段上的所有的点仍然位于这个对象中，那么这个对象是凸的。

对于$P = \alpha_1 P_1 + \alpha_2 P_2 + \alpha_3 P_3 + ... + \alpha_n P_n$ ，当$\alpha_i \geq 0, i = 1, 2, ..., n$时，由这n点的放射和所定义的点集称为这n个点的凸包。

容易验证，连接$\{P_1, P_2, ..., P_n\}$中任何两点的线段都在这n个点的凸包中。从几何上看，如果用一张绷紧的曲面包围给定的一组点，就得到了这些点的凸包。凸包是包含这些点的最小凸对象。

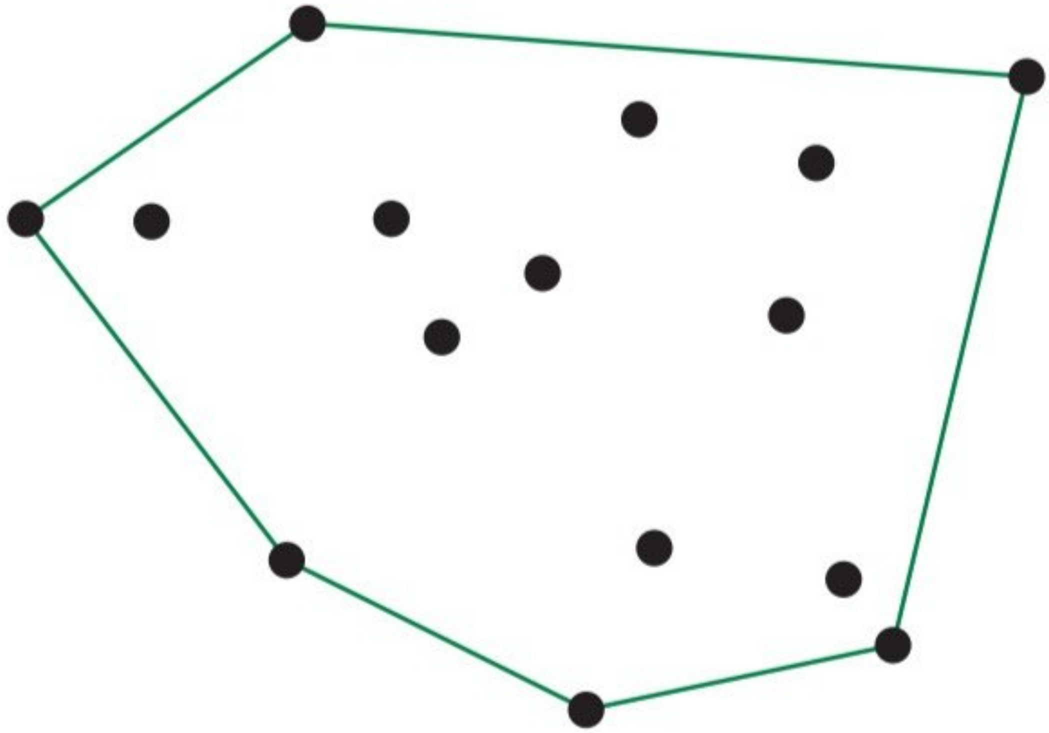

FIGURE 4.13 Convex hull.

4.1.9 点积和叉积

与两个向量的方向关系有关的许多几何概念都是用这两个向量的点积(内积)和叉积来表述的。

正交

如果$u \cdot v = 0$,我们就说u和v是正交的。

在欧式空间中，向量的长度是有定义的。一个向量的长度的平方可以由点积给出

$|u|^2 = u \cdot u$

线性无关

在**向量空间**中，如果一组向量中的任何一个向量都不能由其他向量通过标量-向量乘法和向量-向量加法表示，则称这组向量是线性无关的。向量空间的**维数**是线性无关向量组中向量数目的最大值。

叉积

由不平行的两个向量u, v求出第三个向量n,使得n与u和v都正交。这个向量是通过叉积得到的。

$n = u \times v$

利用叉积可以求出向量夹角的正弦

$|sin\theta| = \frac{|u \times v|}{|u||v|}$,方向由右手坐标系确定。

4.1.10 平面

不在一条直线上的三点唯一确定了一个平面，假设P、Q、R是仿射空间中这样的三个点。该平面上的任意一点的方程为

$T(\alpha', \beta', \gamma') = \alpha' P + \beta' Q + \gamma' R$,其中，$\alpha' + \beta' + \gamma' = 1$

可以使用$(\alpha', \beta', \gamma')$表示一个点，这是一种称为T关于P,Q,R的质心坐标的表示形式。

4.2 三维图元

4.3 坐标系和标架

一个三维向量空间中，可以用任意三个线性无关的向量v_1, v_2, v_3把任意一个向量w表示为

$w = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3$

基

v_1, v_2, v_3

分量

$\alpha_1, \alpha_2, \alpha_3$

w关于这个基的表示

$$a = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} \tag{1}$$

坐标系

通常认为基向量定义了一个坐标系。然而，除了向量，还要处理点和标量，所以还要有一种更加一般的方法。

点

因为**仿射空间**中包含点，所以一旦这样的空间里选择了一个特定的参考点(原点)，就可以以一种明确的方式来表示所有的点。

习惯上将原点作为坐标轴的起点，这在仿射空间里是有意义的。因为我们既要表示点，又要表示向量。不过，这种表示除了要确定基向量以外，还要确定参考点。

标架

原点和基向量决定了一个**标架(frame)**。

标架把向量坐标系的原点固定在了某个点P_0 处。在一个给定的标架下，每个向量可以唯一地表示为

$W = \alpha_1 v_1 \alpha_2 v_2 + \alpha_3 v_3 = \alpha^T v$

这和向量空间中的表示一样；此外，每个点可以唯一地表示为：
$P = P_0 + \beta_1 v_1 + \beta_2 v_2 + \beta_3 v_3 = P_0 + b^T v$

因此，在一个标架下，表示一个向量需要三个标量，而表示一个点需要三个标量和原点的位置。

4.3.1 向量的表示和n元组

假设向量e_1, e_2, e_3构成了一个基，任一向量在这个基下的表示可由一个向量α的分量($\alpha_1, \alpha_2, \alpha_3$)给出，其中

$v = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3$ 。

我们把这些基向量自身的表示记为e_1, e_2, e_3

$e_1 = (1, 0, 0)^T$
$e_2 = (0, 1, 0)^T$
$e_3 = (0, 0, 1)^T$

其中(1,0,0)等就是**三元组**

因此，无需考虑抽象的向量，可以用三元组来思考问题，并且可以把任何向量v的标识写成列矩阵α 或者三元组($\alpha_1, \alpha_2, \alpha_3$)的形式

4.3.2 坐标系的变换

对象标架(建模标架)

我们经常在基向量改变时，求出向量在新坐标系下的表示。例如，在WebGL中，为了便于场景中的各个模型的建模，使用一种对模型很自然的坐标系或者标架来定义几何对象，这个标架被称为**对象标架**或者**建模标架**，。

世界标架

然后这些对象标架的模型又被放到**世界标架**中。

照相机标架或者眼标架

有时候我们需要知道这些对象在照相机看来是什么样子的，此时就需要从世界标架变换到**照相机标架**或者眼标架。从对象标架到眼标架的转换是通过模-视变换矩阵来完成的。

4.3.3 举例：不同基下的表示之间的变换

矩阵的逆

参见线性代数和大脑。

4.3.4 齐次坐标

齐次坐标对三维空间中的点和向量的表示都是四维的。在由(v_1, v_2, v_3, P_0)确定的标架中，任一点P可以唯一地表示为

$P = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 + P_0$

如果把点与标量0和1的"乘法"定义为

$$0.P = 0$$
$$1.P = P$$

那么可以用矩阵乘法把点P形式地表示为：

$$P = [\,\alpha_1, \alpha_2, \alpha_3, 1\,] \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix}$$

其中$[\alpha_1 \quad \alpha_2 \quad \alpha_3, 1]$是点P在由$v_1, v_2, v_3$和$P_0$所确定的标架的**齐次坐标表示**。

我们可以说P的表示是下面的列矩阵。

$$P = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ 1 \end{bmatrix}$$

标架变换的矩阵表示

$$u_1 = \gamma_{11}v_1 + \gamma_{12}v_2 + \gamma_{13}v_3$$

$$u_2 = \gamma_{21}v_1 + \gamma_{22}v_2 + \gamma_{23}v_3$$

$$u_3 = \gamma_{31}v_1 + \gamma_{32}v_2 + \gamma_{33}v_3$$

$$Q_0 = \gamma_{41}v_1 + \gamma_{42}v_2 + \gamma_{43}v_3 + P_0.$$

These equations can be written in the form

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ Q_0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix},$$

where now $\mathbf{M}$ is the 4×4 matrix

$$\boxed{\mathbf{M} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} & 0 \\ \gamma_{21} & \gamma_{22} & \gamma_{23} & 0 \\ \gamma_{31} & \gamma_{32} & \gamma_{33} & 0 \\ \gamma_{41} & \gamma_{42} & \gamma_{43} & 1 \end{bmatrix}.}$$

4.3.5 举例：标架变换

4.3.6 对表示进行处理

利用新标架在原标架下的表示可以得到一个矩阵，该矩阵的逆可以把原标架下的表示变换成新标架下的表示。

4.4 WebGL中的标架(Frame)

4.5 矩阵和向量类型

4.6 建模一个彩色立方体

4.7 仿射变换

变换(transformation)是一个函数，它把一个点(或向量)映射成另一个点(或向量)。

线性函数

一个函数f是线性函数(linear function),当且仅当对于任何标量α和β以及任何两个顶点(或者向量)p和q，都有

$$f(\alpha p+\beta q)=\alpha f(p)+\beta f(q) \tag{2}$$

线性函数的重要性在于：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p和q的变换就可以求出它们的线性组合的变换，这样就无需直接计算每个线性组合的变换。

自由度

变换矩阵中可以任意取值的元素个数。例如：

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

这个变换矩阵总共有12个元素可以任意取值，则称这个变换的自由度为12.

但是在仿射空间中，点和向量的齐次坐标表示稍有不同，

任何向量都可以表示成：

$$u=\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ 0 \end{bmatrix}$$

任何点可以表示为：

$$p = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ 1 \end{bmatrix}$$

如果把任意一个变换应用到一个向量，$v = Cu$

那么可以看到C中只有9个元素起作用，因此向量的变换只有9个自由度。

点的仿射变换具有全部的12个自由度。

计算机图形学中用到大多数变换都是仿射变换，这些变换包括旋转、平移和缩放。

4.8 平移、旋转和缩放

变换可以看成是把点移动到新位置的某种方法，这些点定义了一个或者多个几何对象。

尽管许多变换可以把点移动到新的位置，但是如果要求把一组点移动到新的位置并且保持它们之间的空间关系。

刚体变换

旋转和平移都是刚体变换。旋转和平移的任何组合都不能改变对象的形状和体积，旋转和平移只能改变对象的位置和方向。因此，单靠旋转和平移不能给出所有可能的仿射变换。

缩放是一种仿射变换，但不是刚体变换。

4.8.1 平移

4.8.2 旋转

4.8.3 缩放

为确定一个缩放变换，我们应该指定不动点，缩放的方向和缩放因子(α)

因为不动点需要有3个独立的值来指定，缩放的方向需要两个值确定，缩放因子需要一个值确定，所以缩放有6个自由度。

反射变换

当α是负的时，那么就得到了以不动点为中心沿缩放方向的**反射变换(reflection)**

4.9 变换的齐次坐标表示

所有的图形API都要求我们使用参考系。

在每个标架下，每个仿射变换都可以由如下形式的4x4矩阵来表示。

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

4.9.1 平移

$$p' = p + d \tag{4}$$

把p, p'和d写成齐次坐标的形式：

$$p = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} p' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} d = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \\ 1 \end{bmatrix} \tag{5}$$

则将变换表示成矩阵形式如下：

$$T = \begin{bmatrix} 1 & 0 & 0 & \alpha_x \\ 0 & 1 & 0 & \alpha_y \\ 0 & 0 & 1 & \alpha_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

T被称为**平移矩阵**。

有时候我们把T写成$T(\alpha_x, \alpha_y, \alpha_z)$来强调三个独立的参数。

平移矩阵求逆的新方法

因为把一个点平移d之后再平移-d就回到了原来的位置，所以还可以用另一种方法求出平移矩阵的逆。

$$T^{-1}(\alpha_x, \alpha_y, \alpha_z) = T(-\alpha_x, -\alpha_y, -\alpha_z) = \begin{bmatrix} 1 & 0 & 0 & -\alpha_x \\ 0 & 1 & 0 & -\alpha_y \\ 0 & 0 & 1 & -\alpha_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

4.9.2 缩放

缩放和旋转都有一个不动点，这个点变换以后保持不变。现在把不动点设为原点，以后可以通过变换的复合来得到不动点在任意位置的变换。

不动点位于原点的缩放变换可以沿着三个坐标轴的方向进行彼此独立的缩放。这三个方向的缩放可以表示为：

$$\begin{aligned} x' &= \beta_x x \\ y' &= \beta_y y \\ z' &= \beta_z z \end{aligned} \tag{8}$$

利用齐次坐标，这三个方程可以在一起

$$p' = Sp \tag{9}$$

其中S

$$S = S(\beta_x, \beta_y, \beta_z) = \begin{bmatrix} \beta_x & 0 & 0 & 0 \\ 0 & \beta_y & 0 & 0 \\ 0 & 0 & \beta_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

缩放矩阵的逆

$$S^{-1}(\beta_x, \beta_y, \beta_z) = S(\frac{1}{\beta_x}, \frac{1}{\beta_y}, \frac{1}{\beta_z}) \tag{11}$$

4.9.3 旋转

不动点为原点的旋转

这样的旋转有3个自由度，分别对应于相互独立的绕着3个坐标轴的旋转。必须注意，矩阵乘法不满足交换律。所以绕x轴旋转再绕y轴旋转，和先绕y轴旋转，再绕x周旋转是**不一样的**

绕z轴旋转角度θ

$$R_z = R_z(\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0 & 0 \\ sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

绕x轴旋转角度θ

x轴坐标不变

$$R_x = R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta & -sin\theta & 0 \\ 0 & sin\theta & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

绕y轴旋转角度θ

y轴坐标不变

$$R_y = R_y(\theta) = \begin{bmatrix} cos\theta & 0 & sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -sin\theta & 0 & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

正交矩阵

如果一个矩阵的逆等于它的转置，那么我们称它为正交矩阵。

任何旋转矩阵都是正交矩阵。

任何不动点在原点的旋转矩阵都可以表示成绕三个坐标轴的旋转矩阵的乘积。乘积的顺序按照旋转的顺序。左乘。

4.9.4 错切

错切变换是一类特殊的，基本的仿射变换。

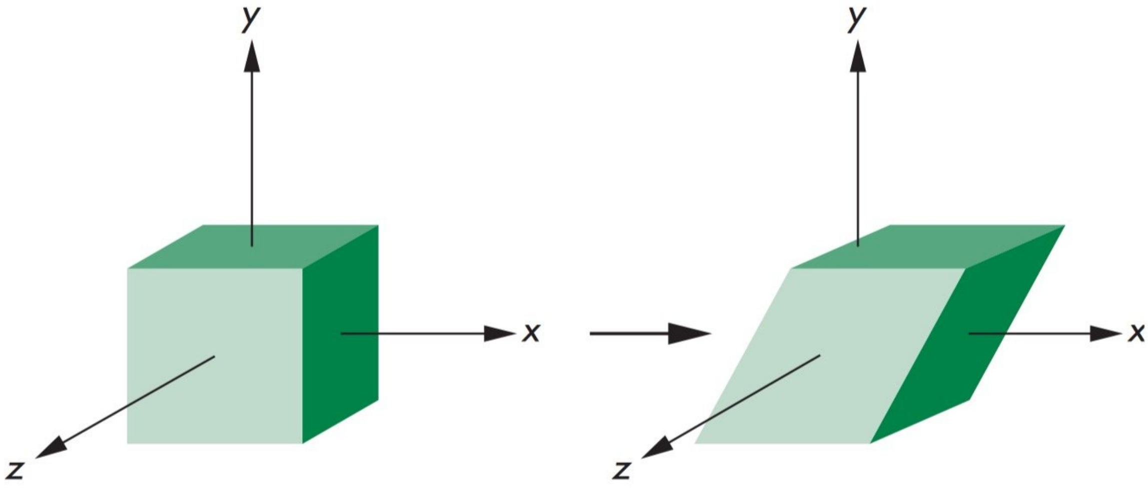

FIGURE 4.41 Shear.

如图4.41所示，如果向右拉立方体的顶面，并且想做拉其底面，那么我们把立方体沿着x方向进行了错切。其中y坐标和z坐标都保持不变，因此为了和其他方向的错切相区别，我们把这个变换叫做**x错切。**

用一个角度θ就可以表征一个x错切。

错切矩阵

$$H_x(\theta) = \begin{bmatrix} 1 & cot\theta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

4.10 变换的级联

4.10.1 不动点在任意位置的旋转

Our first example shows how we can alter the transformations that we defined with a fixed point at the origin (rotation, scaling, shear) to have an arbitrary fixed point. We demonstrate for rotation about the z-axis; the technique is the same for the other cases.

4.10.2 一般的旋转

4.10.3 实例变换

实例

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。

我们可以通过应用一个仿射变换，即**实例变换**使得对象具有期望的大小、方向和位置。

4.10.4 绕任意轴的旋转

这个是个重点，一定要会推导，会看懂。

我们将导出 绕空间中任意点或任意直线旋转的变换矩阵，而且还要说明如何利用方向角确定方向。我们以下图为例

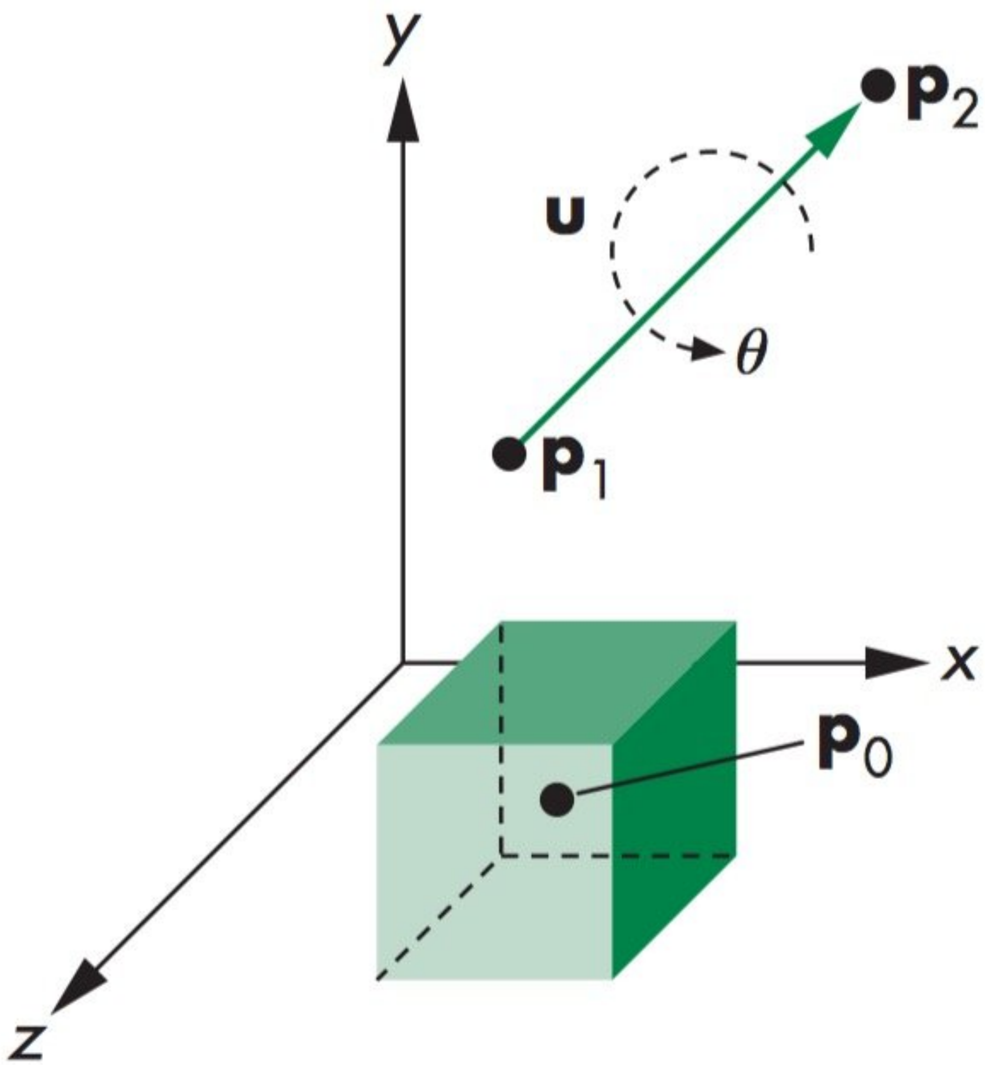

FIGURE 4.52 Rotation of a cube about an arbitrary axis.

需要三个实体指定这个旋转

- fixed point P_0 ,我们假定是这个立方体的中心
- a vector about which we rotate
- an angle of rotation

然而，为了求出表示这个变换的仿射矩阵，我们必须使用某个标架。

绕任意轴的选装的变换矩阵的详细计算方法

设不动点为P_0,单位方向向量为$(\alpha_x,\alpha_y,\alpha_z)$. 所以

$$\alpha_x^2+\alpha_y^2+\alpha_z^2=1 \tag{16}$$

设该单位向量与三个坐标轴的方向角分别为ϕ_x,ϕ_y,ϕ_z

则有

$$\begin{aligned}1\times cos\phi_x&=\alpha_x\\1\times cos\phi_y&=\alpha_y\\1\times cos\phi_z&=\alpha_z\end{aligned} \tag{17}$$

1.将不动点移动到原点

第一个变换$T(-p_0)$

2.将单位方向向量(即旋转轴)绕x轴旋转θ_x，绕到xoz平面。

这个方向要自己把控。如果是满足右手螺旋，则角度为正值，否则角度为负值。

$$R(\theta_x)=\begin{bmatrix}1&0&0&0\\0&cos\theta_x&-sin\theta_x&0\\0&sin\theta_x&cos\theta_x&0\\0&0&0&1\end{bmatrix} \tag{18}$$

3.将旋转轴绕y轴旋转θ_y,绕到与z轴平行的地方

方向同理要遵循右手螺旋定则

$$R(\theta_y)=\begin{bmatrix}cos\theta_y&0&sin\theta_y&0\\0&1&0&0\\-sin\theta_y&0&cos\theta_y&0\\0&0&0&1\end{bmatrix} \tag{19}$$

4.将旋转轴绕z轴旋转角度θ

$$R_z=R_z(\theta)=\begin{bmatrix}cos\theta&-sin\theta&0&0\\sin\theta&cos\theta&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix} \tag{20}$$

5.将图形绕y轴旋转角度$-\theta_y$

$$R(-\theta_y)=\begin{bmatrix} cos\theta_y & 0 & -sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ sin\theta_y & 0 & cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{21}$$

6.将图形绕x轴旋转角度$-\theta_x$

$$R(-\theta_x)=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta_x & sin\theta_x & 0 \\ 0 & -sin\theta_x & cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{22}$$

7. 将不动点移动到p_0

$T(p_0)$

所以旋转部分的整体矩阵为：

$$R(-\theta_x)R(-\theta_y)R(\theta)R(\theta_y)R(\theta_x) \tag{23}$$

下面，我一项一项计算：

$$\begin{aligned} R(\theta_y)R(\theta_x) &= \begin{bmatrix} cos\theta_y & 0 & sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -sin\theta_y & 0 & cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta_x & -sin\theta_x & 0 \\ 0 & sin\theta_x & cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} cos\theta_y & sin\theta_y sin\theta_x & sin\theta_y cos\theta_x & 0 \\ 0 & cos\theta_x & -sin\theta_x & 0 \\ -sin\theta_y & cos\theta_y sin\theta_x & cos\theta_y cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{24}$$

$$\begin{aligned} R(-\theta_x)R(-\theta_y) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta_x & sin\theta_x & 0 \\ 0 & -sin\theta_x & cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\theta_y & 0 & -sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ sin\theta_y & 0 & cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} cos\theta_y & 0 & -sin\theta_y & 0 \\ sin\theta_x sin\theta_y & cos\theta_x & sin\theta_x cos\theta_y & 0 \\ cos\theta_x sin\theta_y & -sin\theta_x & cos\theta_x cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{25}$$

$$R(-\theta_x)R(-\theta_y)R_z(\theta) = \begin{bmatrix} cos\theta_y & 0 & -sin\theta_y & 0 \\ sin\theta_x sin\theta_y & cos\theta_x & sin\theta_x cos\theta_y & 0 \\ cos\theta_x sin\theta_y & -sin\theta_x & cos\theta_x cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\theta & -sin\theta & 0 & 0 \\ sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{26}$$

$$= \begin{bmatrix} cos\theta_y cos\theta & -cos\theta_y sin\theta & -sin\theta_y & 0 \\ sin\theta_x sin\theta_y cos\theta + cos\theta_x sin\theta & -sin\theta_x sin\theta_y sin\theta + cos\theta_x cos\theta & sin\theta_x cos\theta_y & 0 \\ cos\theta_x sin\theta_y cos\theta - sin\theta_x sin\theta & -cos\theta_x sin\theta_y sin\theta - sin\theta_x cos\theta & cos\theta_x cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{final} = \begin{bmatrix} cos\theta_y cos\theta & -cos\theta_y sin\theta & -sin\theta_y & 0 \\ sin\theta_x sin\theta_y cos\theta + cos\theta_x sin\theta & -sin\theta_x sin\theta_y sin\theta + cos\theta_x cos\theta & sin\theta_x cos\theta_y & 0 \\ cos\theta_x sin\theta_y cos\theta - sin\theta_x sin\theta & -cos\theta_x sin\theta_y sin\theta - sin\theta_x cos\theta & cos\theta_x cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{27}$$

$$\begin{bmatrix} cos\theta_y & sin\theta_y sin\theta_x & sin\theta_y cos\theta_x & 0 \\ 0 & cos\theta_x & -sin\theta_x & 0 \\ -sin\theta_y & cos\theta_y sin\theta_x & cos\theta_y cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

由于矩阵过于复杂。设最终的矩阵为A，下面我将表示出每一项。

$$A_{11} = cos^2\theta_y cos\theta + sin^2\theta_y \tag{28}$$

$$A_{12} = cos\theta_y cos\theta sin\theta_y sin\theta_x - cos\theta_y sin\theta cos\theta_x - sin\theta_y cos\theta_y sin\theta_x \tag{29}$$

$$A_{13} = cos\theta_y cos\theta sin\theta_y cos\theta_x + cos\theta_y sin\theta sin\theta_x - sin\theta_y cos\theta_y cos\theta_x \tag{30}$$

$$A_{14} = 0 \tag{31}$$

$$A_{21} = (sin\theta_x sin\theta_y cos\theta + cos\theta_x sin\theta)cos\theta_y - sin\theta_x cos\theta_y sin\theta_y \tag{32}$$

$$A_{22} = (sin\theta_x sin\theta_y cos\theta + cos\theta_x sin\theta)sin\theta_y sin\theta_x + (-sin\theta_x sin\theta_y sin\theta + cos\theta_x cos\theta)cos\theta_x + sin\theta_x cos^2\theta_y sin\theta_x \tag{33}$$

$$A_{23} = (sin\theta_x sin\theta_y cos\theta + cos\theta_x sin\theta)sin\theta_y cos\theta_x - (-sin\theta_x sin\theta_y sin\theta + cos\theta_x cos\theta)sin\theta_x + sin\theta_x cos^2\theta_y cos\theta_x \tag{34}$$

$$A_{31} = (cos\theta_x sin\theta_y cos\theta - sin\theta_x sin\theta)cos\theta_y - cos\theta_x cos\theta_y sin\theta_y \tag{35}$$

$$A_{32} = (cos\theta_x sin\theta_y cos\theta - sin\theta_x sin\theta)sin\theta_y sin\theta_x + (-cos\theta_x sin\theta_y sin\theta - sin\theta_x cos\theta)cos\theta_x + cos\theta_x cos^2\theta_y sin\theta_x \tag{36}$$

$$A_{33} = (cos\theta_x sin\theta_y cos\theta - sin\theta_x sin\theta)sin\theta_y cos\theta_x - (-cos\theta_x sin\theta_y sin\theta - sin\theta_x cos\theta)sin\theta_x + cos\theta_x cos^2\theta_y cos\theta_x \tag{37}$$

其余的好说！

好了，有关这些角度怎么求的问题，看图就懂了！注意旋转角度的正负号！

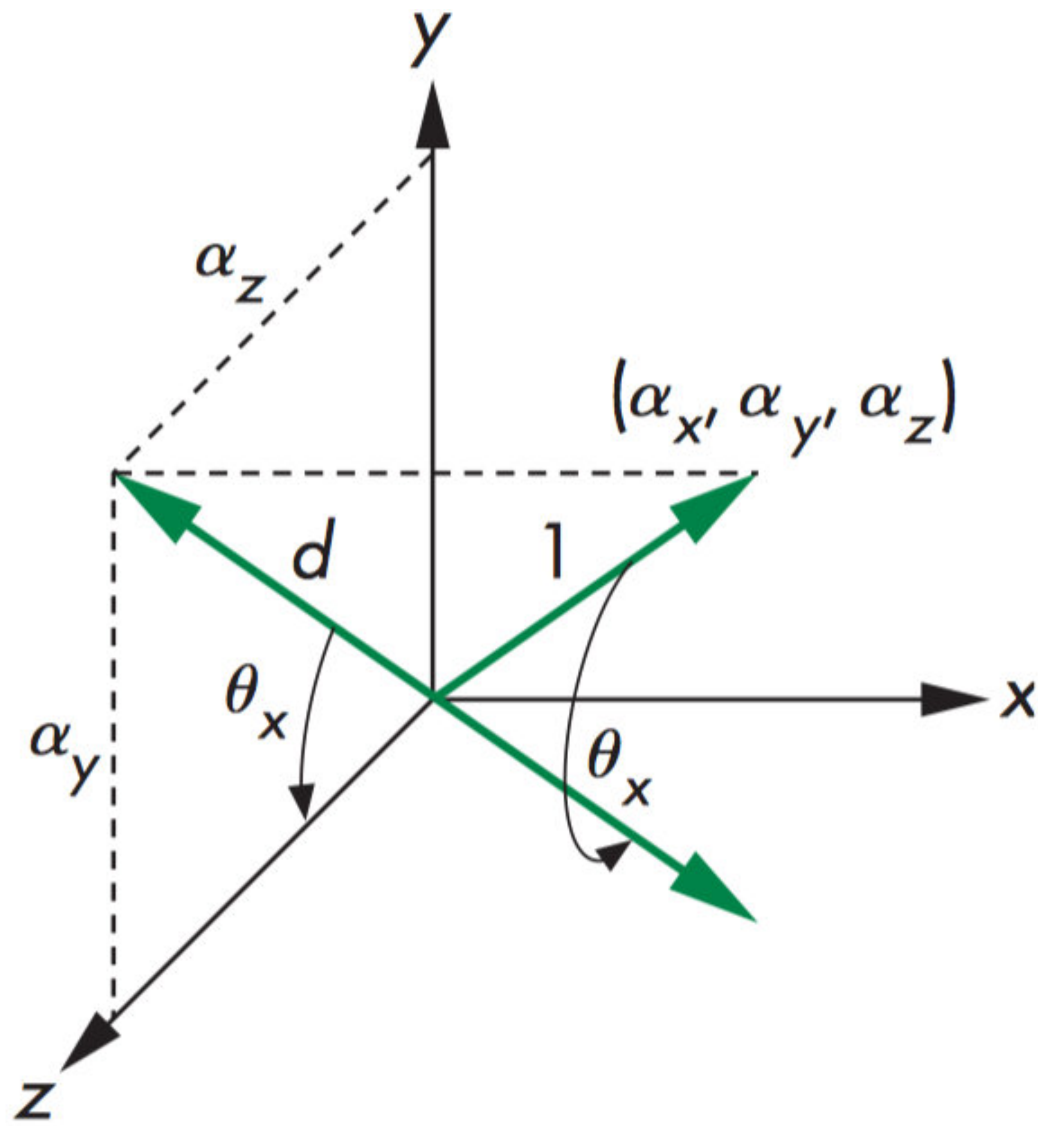

FIGURE 4.56 Computation of the x rotation.

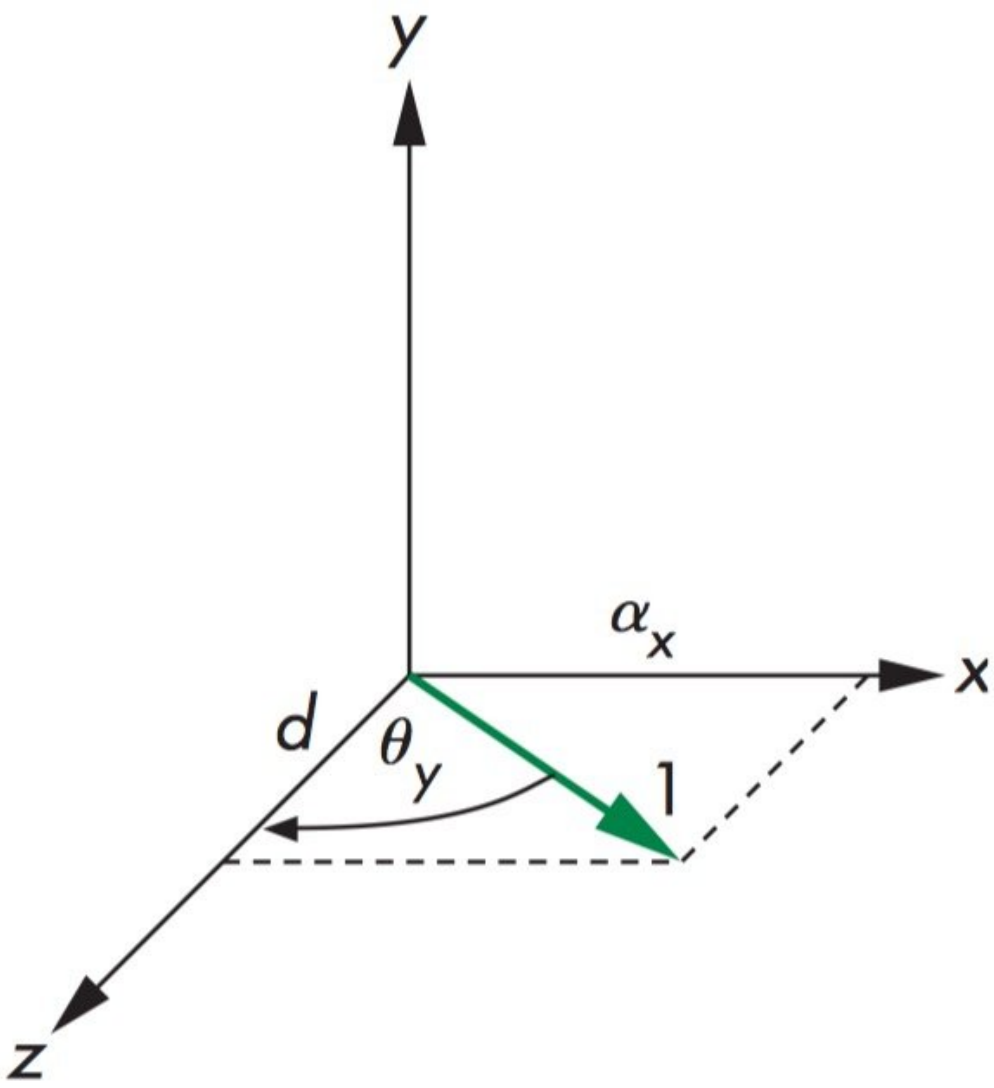

FIGURE 4.57 Computation of the y rotation.

归一化向量

单位向量

第一个变换

第一个变换是平移$T(-p_0)$

最后一个变换是$T(p_0)$

在经过平移$T(-p_{)}$之后，旋转轴将通过原点。

任意一个旋转都可以通过三次绕坐标轴的旋转来合成。

现在的问题更困难，因为我们不知道每次旋转的角度。

这里采用的方法是通过**两次旋转**使得旋转轴v和z轴重合。然后就可以绕z轴旋转角度θ . 此后再把那两次使得旋转轴和z轴重合的旋转抵消掉。因此，要求的旋转矩阵可以写成下面的形式。

$$R = R_x(-\theta_x)R_y(-\theta_y)R_z(\theta)R_y(\theta_y)R_x(\theta_x) \tag{38}$$