中国科学技术大学
University of Science and Technology of China

计算机图形学

计算机学院 黄章进
zhuang@ustc.edu.cn

5.1 几何

5.2 表示

5.3 变换

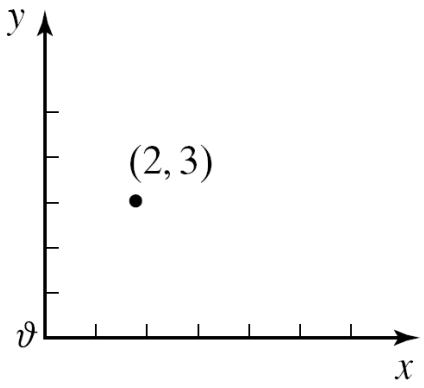
5.1 几何

- 与坐标无关的几何
- 仿射空间
  - 标量
  - 向量
  - 点
- 基本几何图元
  - 线段
  - 多边形

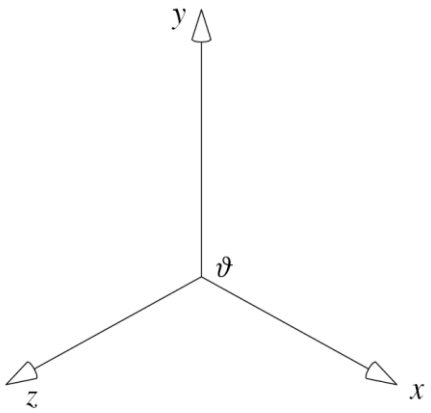
基本几何要素

- 几何研究n维空间中对象之间的关系
  - 在计算机图形学中，我们对三维空间中的对象感兴趣
- 希望得到一个几何形状的最小集合，根据这个集合可以建立起更复杂的对象
- 需要三个基本元素
  - 标量
  - 向量
  - 点

坐标系

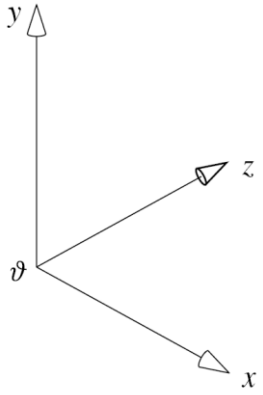


二维坐标系



三维坐标系

（右手系）



三维坐标系

（左手系）

与坐标无关的几何

- 在初等解析几何中，主要应用的是直角坐标系（笛卡尔坐标系）
  - 点在空间中的位置是**p** = (x,y,z)
  - 通过对这些坐标进行代数运算导出结果
- 这种方法不是基于物理的
  - 从物理的角度来讲，点的存在性是与坐标系的具体位置无关的
  - 绝大多数几何结果是不依赖于坐标系的
  - 例如，在欧氏几何中，两个三角形全等是指它们有两个对应边和夹角相等

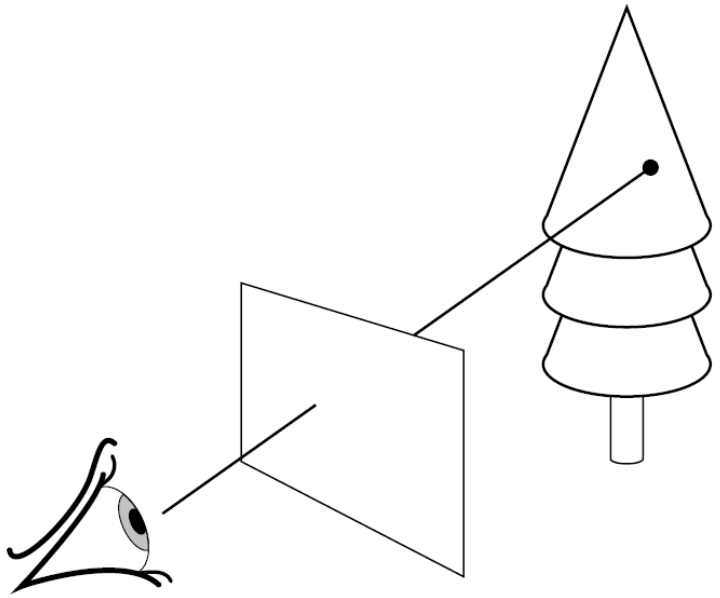
标量

- 在几何中需要三个基本元素
  - 标量、向量、点
- 标量可以定义为集合中的成员，集合中具有两种运算：加法和乘法
  - 两种运算满足交换律、结合律、分配律
  - 加法单位元(0)和乘法单位元(1)
  - 加法逆元和乘法逆元（隐含定义了减法和除法）
- 例：实数或复数全体，通常的加法与乘法
- 标量自身没有几何属性

为什么需要向量？

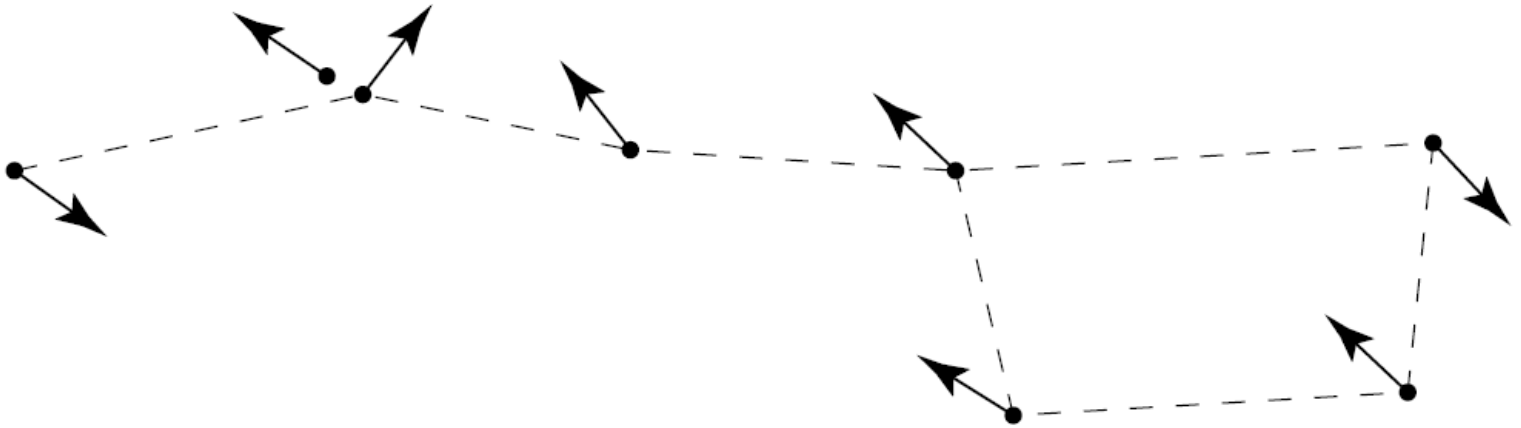
场景：树与照相机

- 照相机需要在视平面上形成一幅图像表示这棵树。视平面上哪些点需要被激活？
- 透视投影需要利用向量来构造



视平面

为什么需要向量？

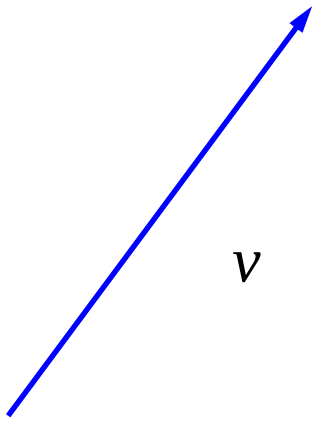


北斗七星的当前位置及移动方向

箭头末尾的位置表示五万年后各星的位置

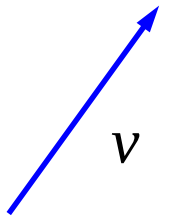
向量

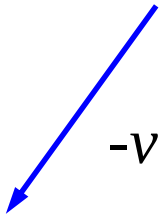
- 物理定义：向量是具有如下两种属性的量
  - 方向
  - 大小或长度: $|v|$
- 例:
  - 力
  - 速度
  - 有向线段
    - 这也是图形学中最重要的例子
    - 可以对应到其它类型上
- 用小写字母表示

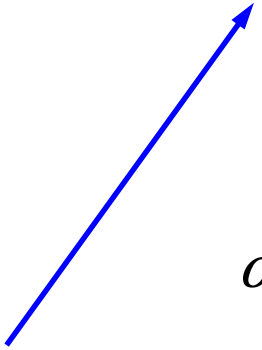


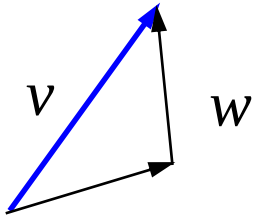
向量运算

- 每个向量都有逆
  - 同样长度但是指向相反的方向
- 每个向量都可以与标量相乘
- 有一个零向量
  - 零长度，方向不定
- 两个向量的和为向量
  - 三角形法则









$v = u + w$

$u = v - w$

(线性)向量空间

- 处理向量的数学系统
- 运算一：向量-向量加法 $w = u+v$
  - 封闭性： $u, v \in V, u+v \in V$
  - 交换律： $u+v = v+u$
  - 结合律： $u+(v+w) = (u+v)+w$
  - 零向量0： $u \in V, u+0 = u$
  - 加法逆元-u： $u+(-u) = 0$

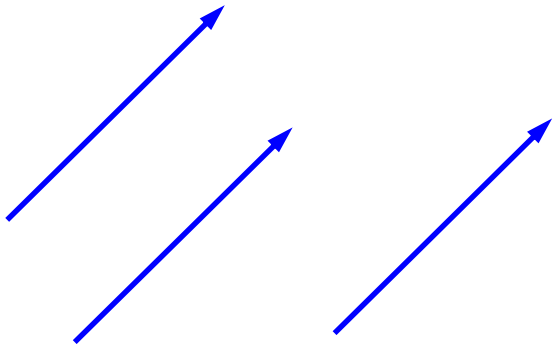
(线性)向量空间

- 运算二：标量-向量乘法 $u = \alpha v$
  - 分配律：$\alpha (u+v) = \alpha u + \alpha v$
  - $(\alpha+\beta) u = \alpha u + \beta u$

- 在向量空间中，表达式 $v = u + 2w - 3r$有意义
- 向量空间例子：有向线段、实数的n元组

向量没有位置

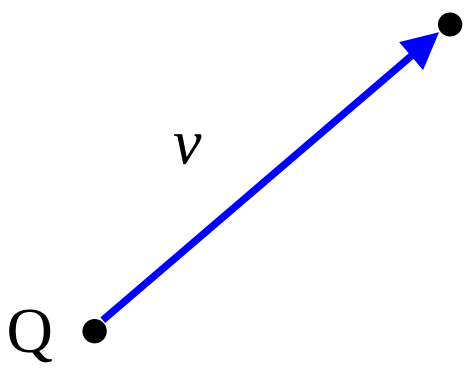
- 下述向量是相等的
  - 因为它们具有相同的方向与长度



- 对几何而言只有向量空间是不够的
  - 还需要点

点

- 空间中的位置。数学上，点没有大小和形状。
  - 用大写字母表示
- 点与向量之间可进行的运算
  - 点与点相减得到一个向量
  - 等价地，点与向量相加得到新点



$v = P - Q$

$P = v + Q$

有意义：$P+3v$, $P-2C+3v$

无意义：$P+3Q-v$

仿射空间

- 仿射空间 = 点 + 向量空间
- 运算:
  - 向量-向量加法 → 向量
  - 标量-向量乘法 → 向量
  - 点-向量加法 → 点（等价地，点-点减法→ 向量）
  - 标量-标量运算 → 标量
  - 上述运算均是与坐标无关的
- 对于任意点P，定义
  - $1 \cdot P = P$
  - $0 \cdot P = \mathbf{0}$ (零向量)

向量与点的线性组合

- 给定n个向量$v_1, v_2, \ldots, v_n$, 以及n个标量$\alpha_1, \alpha_2, \ldots, \alpha_n$, 则由归纳法可以证明

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n$$

也是向量，称为这组向量的线性组合

- 给定n个点$P_1, P_2, \ldots, P_n$, 以及n个标量$\alpha_1, \alpha_2, \ldots, \alpha_n$, 则

$$P = \alpha_1 P_1 + \alpha_2 P_2 + \ldots + \alpha_n P_n$$

是什么？

  - 所给的定义需要与坐标无关

点的线性组合

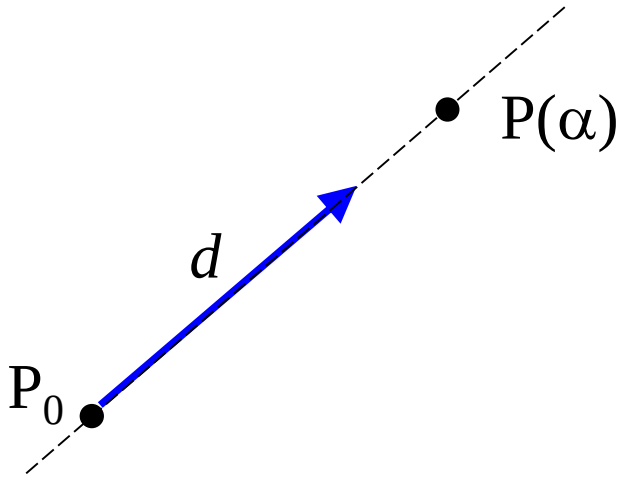
- 固定坐标系，取定其中的两点，那么$P_1 + P_2$是什么？
  - 当P_1为原点时，$P_1 + P_2$等于P_2
  - 当P_1与P_2关于原点对称时，$P_1 + P_2$为原点
  - 所以$P_1 + P_2$的位置与坐标系有关
- 组合系数不能是任意数

点的特殊线性组合

- 由归纳法，从“点 – 点 = 向量” 和“标量·向量 = 向量”可知当组合系数和$\alpha_1 + \alpha_2 + \ldots + \alpha_n = 0$时，点的线性组合为向量
- $\frac{1}{2} P_1 + \frac{1}{2} P_2 = P_1 + \frac{1}{2}(P_2 - P_1)$ = 点 + 向量 = 点
  - 实际上，$\frac{1}{2} P_1 + \frac{1}{2} P_2$表示两点的中点，这是与坐标无关的定义
- 当$\alpha_1 + \alpha_2 + \ldots + \alpha_n = 1$时，点的线性组合为点，称为给定点的仿射组合
- 除此之外，其它形式的线性组合没有与坐标无关的意义

直线

- 考虑具有下述形式的所有点
  - $P(\alpha) = P_0 + \alpha\, d$
  - 即所有过P_0点，与P_0连线平行于向量d的点



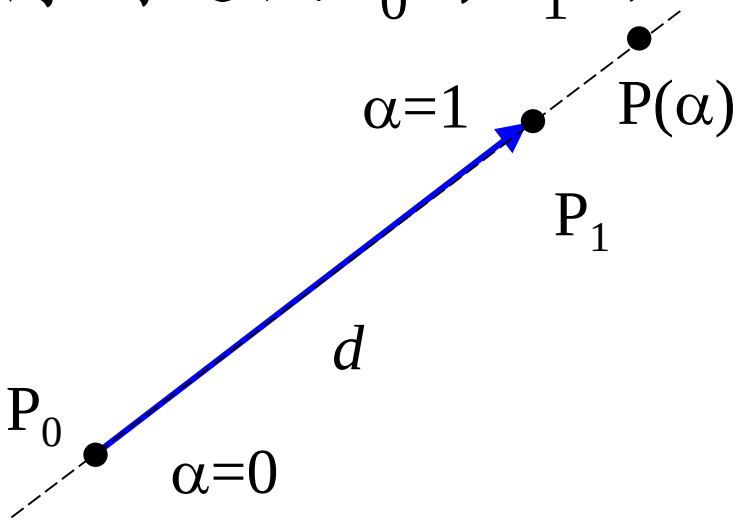
参数形式

- 上述定义直线的形式称为参数形式
  - 比其它形式更一般和稳定
  - 可以推广到曲线和曲面
- 二维形式（坐标系相关）
  - 显式：$y = mx + h$
  - 隐式：$ax + by + c = 0$
  - 参数形式：

$$x(\alpha) = \alpha\, x_0 + (1 - \alpha)\, x_1$$
$$y(\alpha) = \alpha\, y_0 + (1 - \alpha)\, y_1$$

射线与线段

- 如果限定$\alpha \geq 0$, 那么$P(\alpha)$就是从P_0出发，方向为 d 的射线(ray)
- 如果采用两点定义向量d, 那么
  $P(\alpha) = P_0 + \alpha (P_1 - P_0) = (1 - \alpha) P_0 + \alpha P_1$
- 当 $0 \leq \alpha \leq 1$，那么就会得到连接P_0与P_1两点的线段(line segment)



两点线性插值

- 给定两点A, B，那么它们的仿射组合

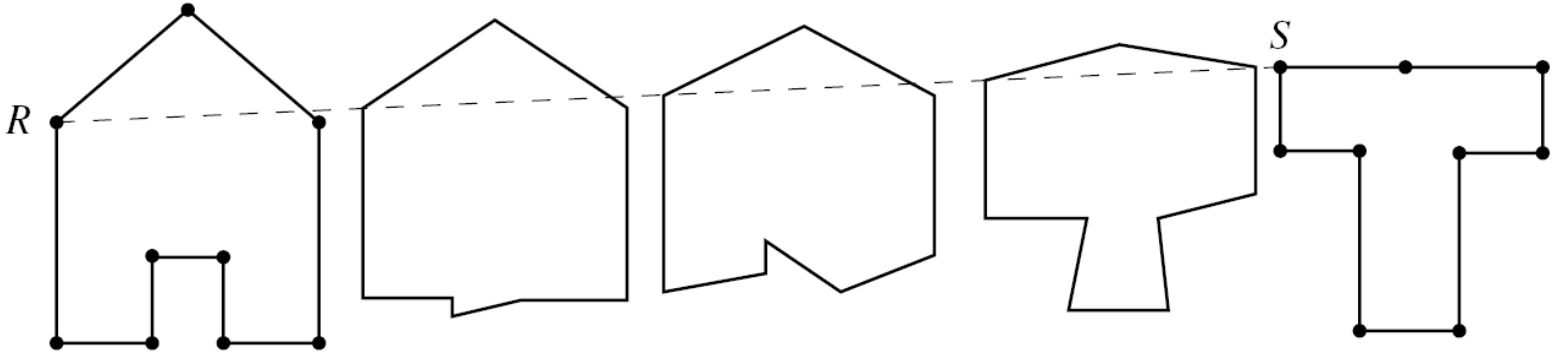
$$P(t) = (1 - t)\,A + t\,B$$

就定义了过这两点的一条直线

- 线性插值在艺术和计算机动画有许多有趣的应用
  - 关键帧

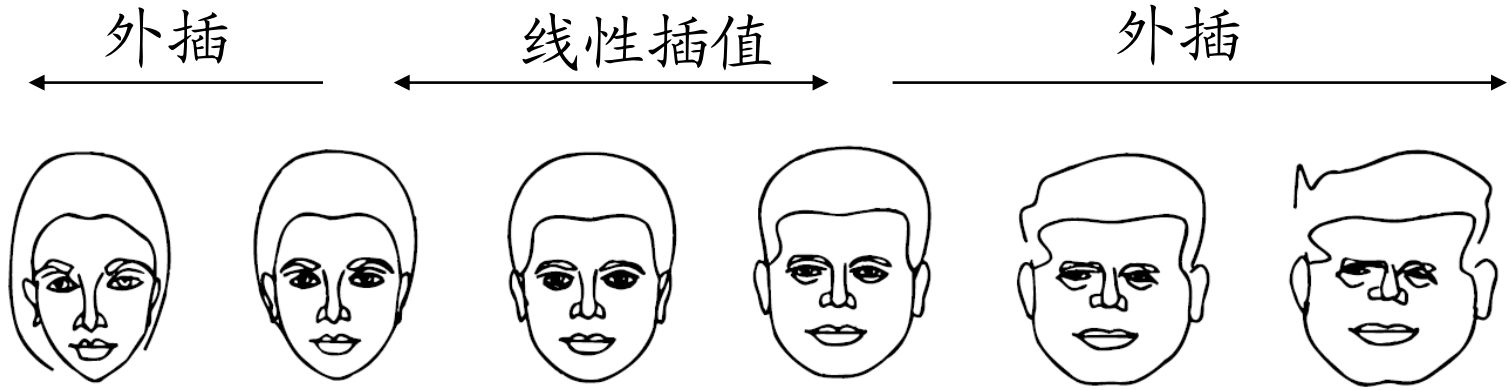
多边形的变形

- 给定两个有同样数目顶点的折线，那么利用线性插值可以给出从第一个折线到第二个折线的平滑过渡

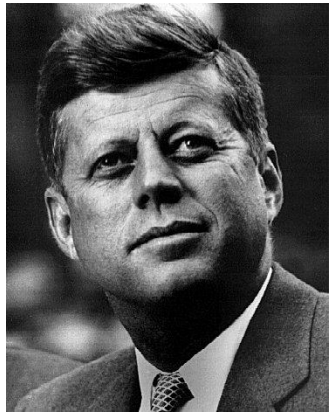


男人→女人

名人脸



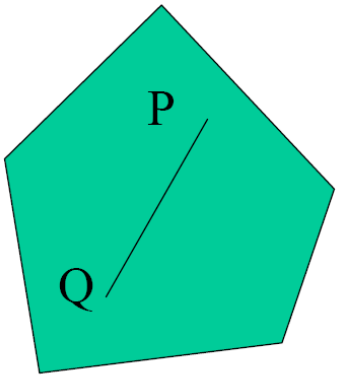
Elizabath Taylor



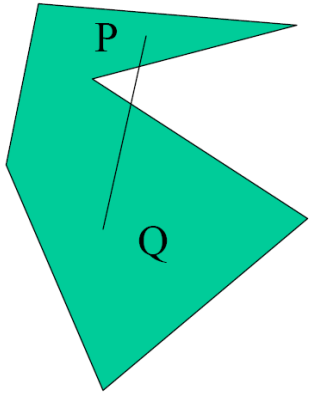
John F. Kennedy

凸性

- 一个对象为凸的(convex)，当且仅当在对象中任何两点的连接线段也在该对象内



凸



非凸

仿射和

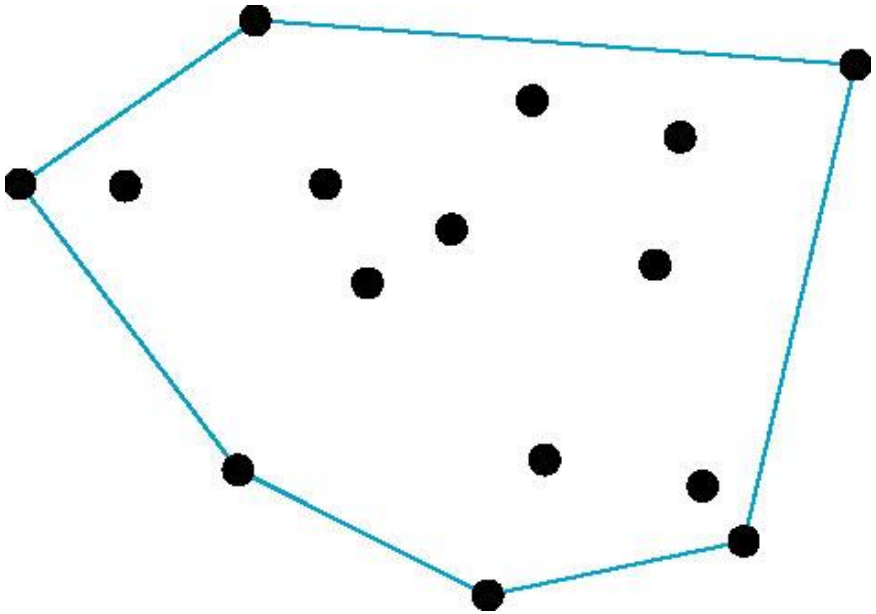
- 考虑“和”式

$$P = \alpha_1 P_1 + \alpha_2 P_2 + \ldots + \alpha_n P_n$$

- 当$\alpha_1 + \alpha_2 + \ldots + \alpha_n = 1$时上述和式有意义，此时结果就称为点$P_1, P_2, \ldots, P_n$的仿射和(affine sum)
- 另外，如果$\alpha_i \geq 0$，那么得到$P_1, P_2, \ldots, P_n$的凸包(convex hull)

凸包

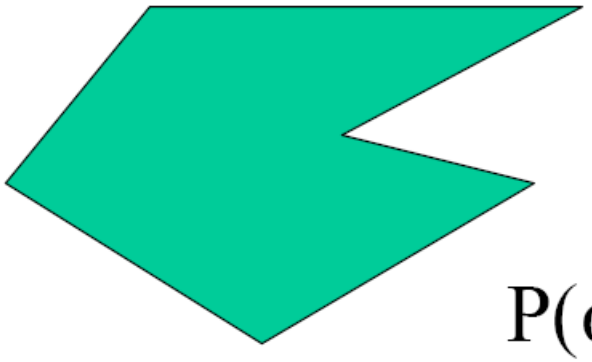
- 最小的包含$P_1, P_2, \ldots, P_n$的凸体
- 可以用“收缩包装”的方式得到



曲线与曲面

- 曲线是形式为P(α)的单参数定义的几何体，其中的函数为非线性
- 曲面是由形式为P(α, β)的两个参数定义的几何体
  - 线性函数对应于平面和多边形

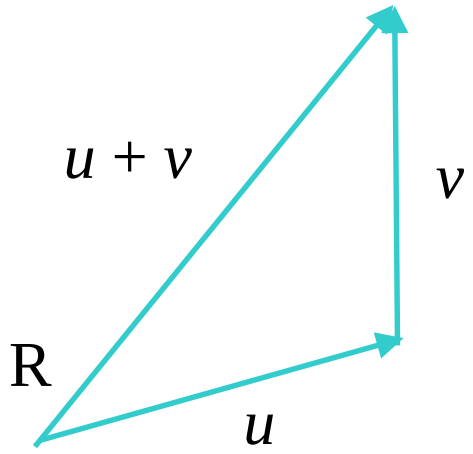
P(α)



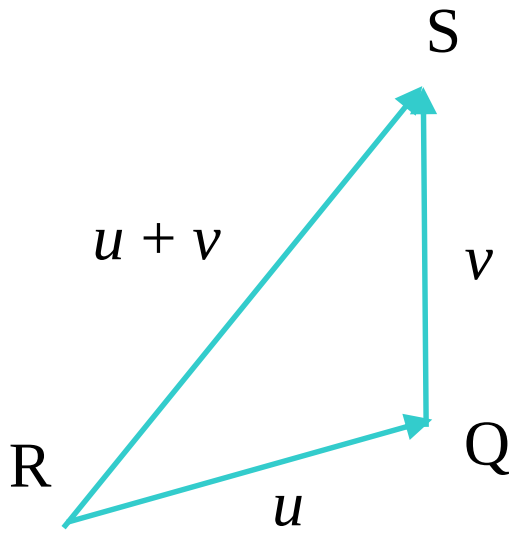
P(α, β)

平面

- 平面是由一个点与两个不平行的向量或者三个不共线的点确定的

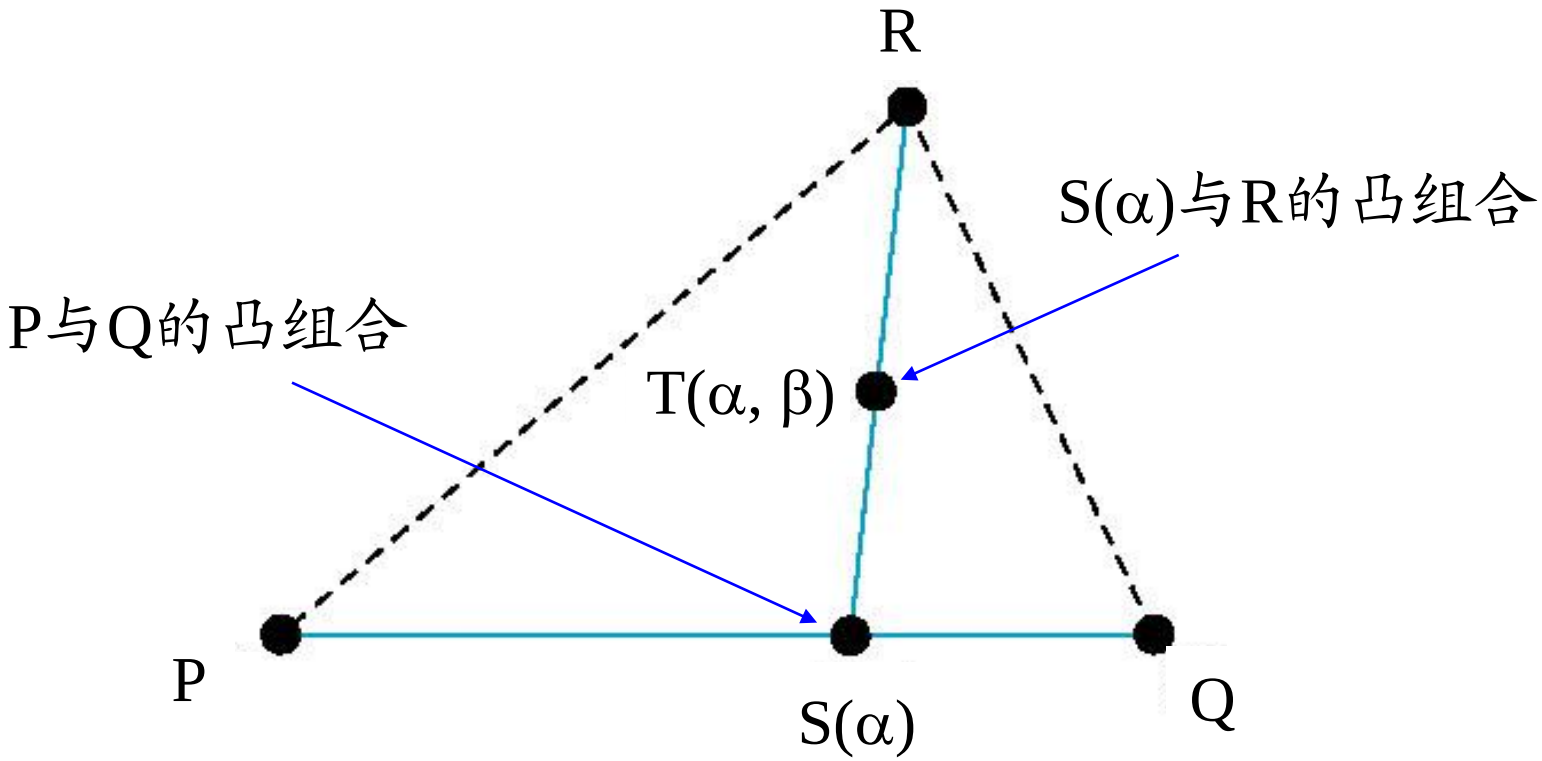


$$P(\alpha, \beta) = R + \alpha u + \beta v$$



$$P(\alpha, \beta) = R + \alpha(Q - R) + \beta(S - Q)$$

三角形



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

$$T(\alpha, \beta) = \beta S(\alpha) + (1-\beta)R$$

当$0 \le \alpha, \beta \le 1$时定义在三角形内的点

重心坐标

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

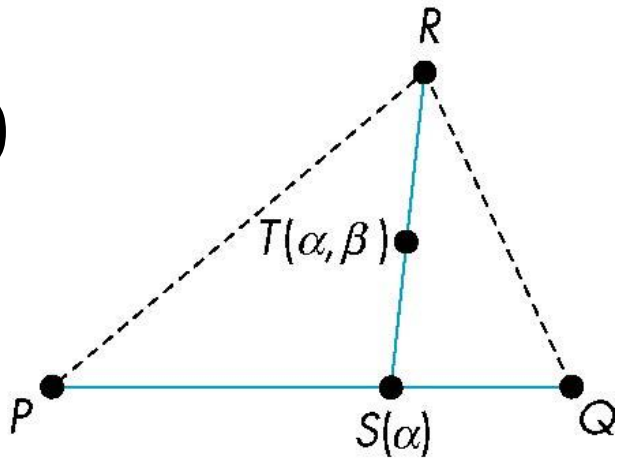
$T(\alpha, \beta) = \beta S(\alpha)+(1- \beta)R$

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

$= \alpha_1 P+\alpha_2 Q+\alpha_3 R$

$= T(\alpha_1 , \alpha_2 , \alpha_3)$

这里， $\alpha_1 +\alpha_2 +\alpha_3 = 1, \alpha_i >=0$

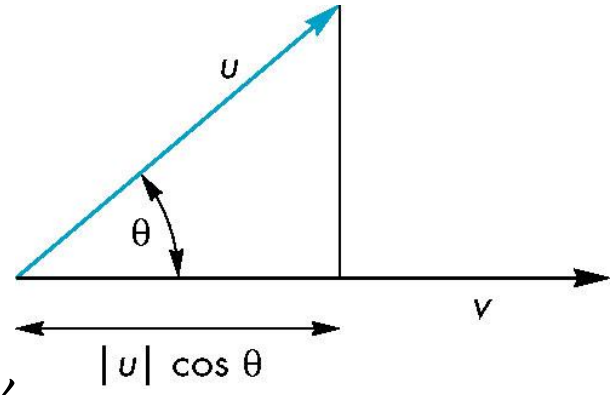


$(\alpha_1 , \alpha_2 , \alpha_3)$称为T点的重心坐标表示

http://www.inf.usi.ch/hormann/barycentric/

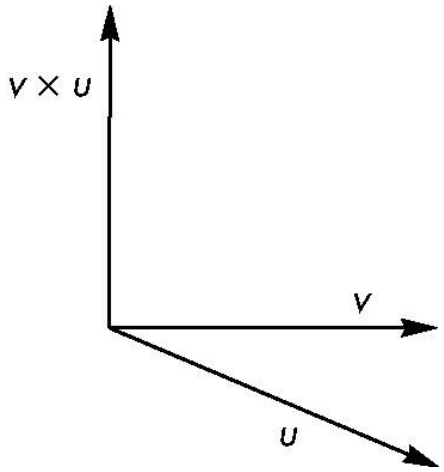
向量的内积

- 欧氏空间是向量空间的扩展，通过定义内积运算，增加了长度或者距离的度量
  - 仿射空间中不能度量两个点的距离，或者说向量空间中不能度量向量的长度
- 内积或点积：$u \cdot v = |u|\,|v| \cos\theta$, θ为两个向量的夹角
  - 正交：$u \cdot v = 0 \Leftrightarrow u \perp v$
  - 向量长度的平方：$|u|^2 = u \cdot u$
  - 夹角的余弦$\cos\theta = u \cdot v / (|u|\,|v|)$
  - $|u| \cos\theta\, v$ 是u在v上的正交投影



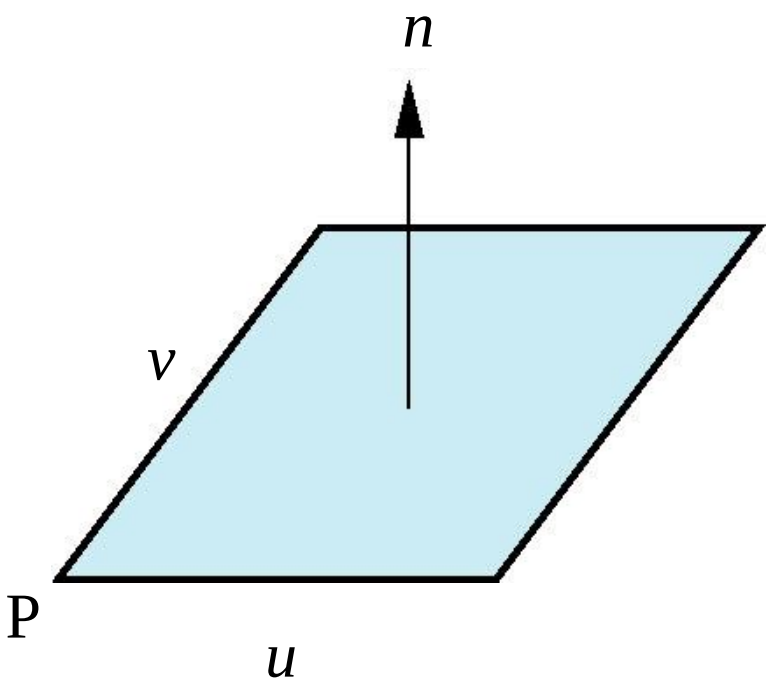
向量的外积

- 外积或叉积：$u \times v$为向量，其长度等于$|u|\,|v|\sin\theta$，方向垂直于u, v所在的平面，并且保证$u, v, u \times v$成为右手系，其中θ为两个向量的夹角。
  - $u \times v = \mathbf{0} \Leftrightarrow u \,//\, v$
  - 夹角的正弦 $|\sin\theta| = |u \times v|/(|u|\,|v|)$



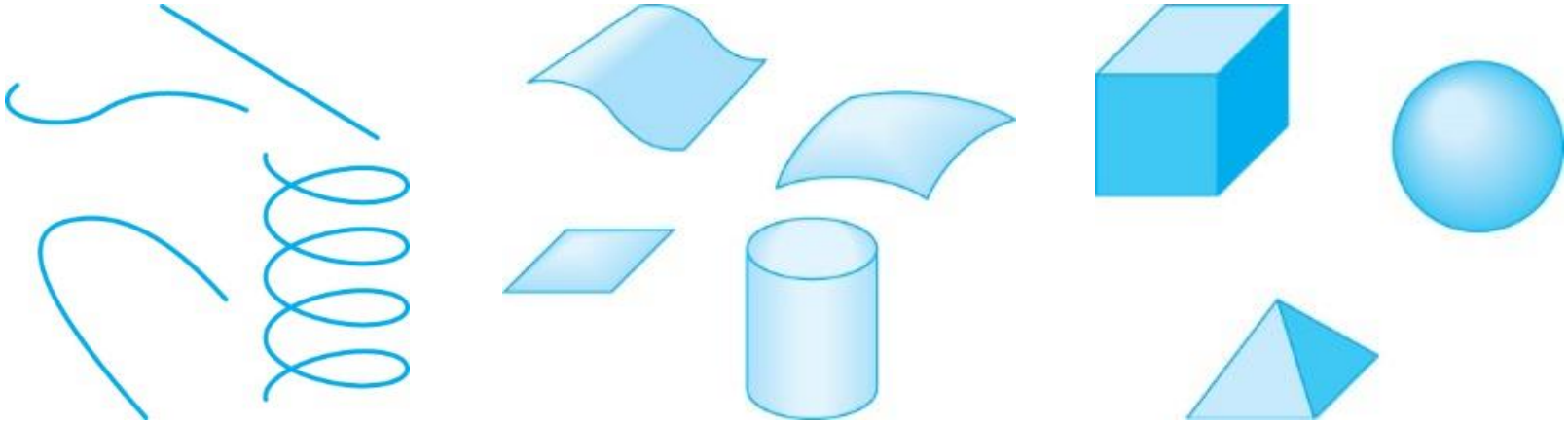
法向量

- 每个平面都有一个垂直于自身的向量n
- 在平面的点与二向量形式
  $P(\alpha, \beta) = R + \alpha u + \beta v$
  中，可以应用向量的外积得到
  $$n = u \times v$$
- 平面方程的等价形式:
  $$(P(\alpha, \beta)-R) \cdot n=0$$



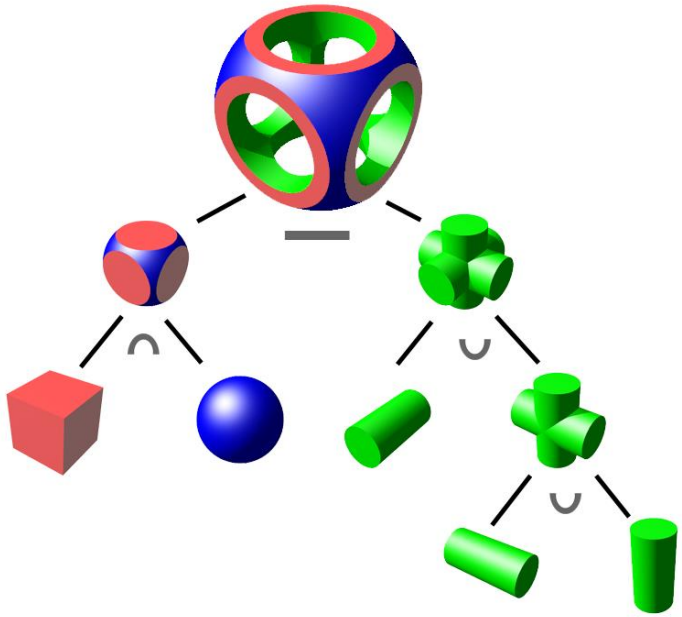
三维图元

- 现有图形系统不能支持所有的三维对象，除非通过近似方法
  - 对象的数学定义可能很复杂
  - 只关注能被高效实现的对象



三维图元

- 适合当今图形软硬件的三维对象的特征
  - 可以用表面来表示，并认为是中空的
    - 2维B-rep（边界表示）vs CSG（构造体几何）
  - 可以由三维空间中的一组顶点来确定
  - 对象可由平面凸多边形组成或近似



5.1 几何

5.2 表示

5.3 变换

5.2 表示

- 维数与基
- 向量空间的坐标系
- 仿射空间的标架
- 齐次坐标
- 基与标架的变换
- OpenGL中的标架

线性无关

- 一组向量$v_1, v_2, \ldots, v_n$称为线性无关的，是指
  $\alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n = 0$
  当且仅当$\alpha_1 = \alpha_2 = \ldots = \alpha_n = 0$
- 如果一组向量是线性无关的，那么不能把其中一个向量表示成其它向量的线性组合
- 如果一组向量是线性相关的，那么其中至少有一个向量可以表示为其它向量的线性组合

维数

- 在向量空间中，最大的线性无关向量组的元素个数是固定的，这个数就称为空间的维数(dimension)
- 在n维空间中，任意n个线性无关的向量构成空间的一组基(basis)
  - 基不唯一
- 给定空间的一组基$v_1, v_2, \ldots, v_n$，空间中任意向量v都可以表示为

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n$$

其中$\{\alpha_i\}$是唯一的

表示

- 到现在为止我们只是讨论几何对象，而没有使用任何参考标架，例如坐标系
- 需要一个参考标架把点和对象与物理世界中的对象联系在一起
  - 例如，点在哪儿？如果没有参考系的话，就无法回答这个问题
  - 世界坐标
  - 照相机坐标

坐标系

- n维向量空间的一组基$v_1, v_2, \ldots, v_n$定义了一个坐标系(coordinate system)
- 一个向量v可以表示为$v = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n$
- 标量组$\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$就称为$v$相对于给定基的表示(representation)
  - 可以把表示写成标量的列矩阵

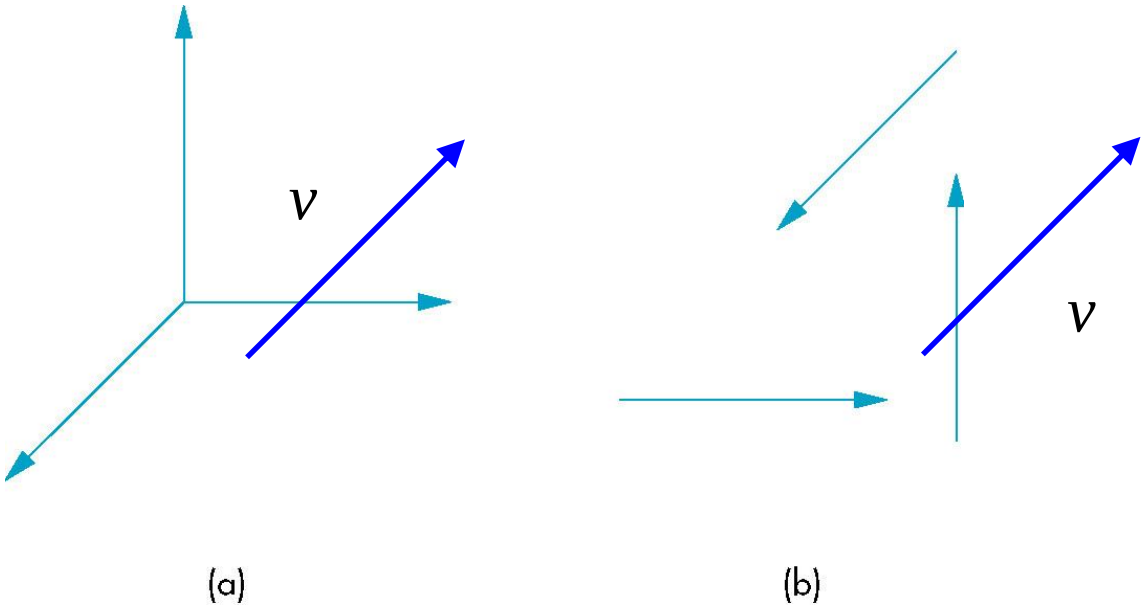
$$\mathbf{a} = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$$

示例

- $v = 2v_1 + 3v_2 - 4v_3$
- $\mathbf{a} = [2, 3, -4]^T$
- 注意上述表示是相对一组特定的基而言的
- 例如，在OpenGL中刚开始是相对于世界坐标系表示向量的，稍后要把这个表示变换到照相机坐标系(或者称为视点坐标系)中

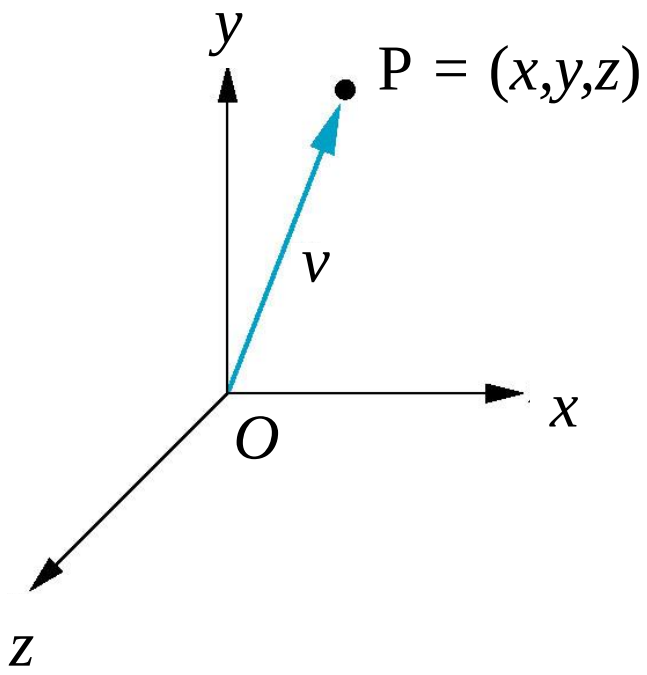
坐标系

- 哪个正确？
  - 都正确，因为向量没有固定位置
- 向量空间没有点，坐标系没有原点



标架

- 坐标系是不足于表示点的
- 如果要在仿射空间中考虑问题，那么可以在基向量组中增加一个参考点(称为原点)，从而构成一个标架(frame)
  - 标架 = 原点 + 坐标系
  - v和P



在标架中的表示

- 标架是由$(O, v_1, v_2, \ldots, v_n)$确定的
- 在这个标架中，每个向量可以表示为

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n$$

- 每个点可以表示为

$$P = O + \beta_1 v_1 + \beta_2 v_2 + \ldots + \beta_n v_n$$

点与向量的混淆

考虑点与向量

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n$$

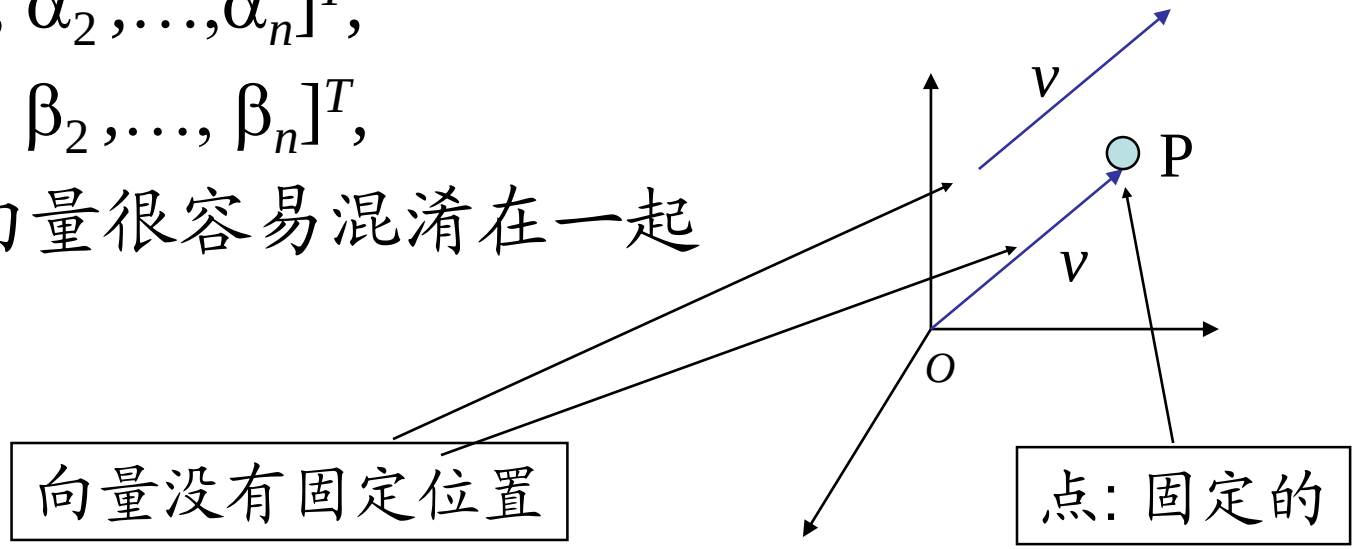
$$P = O + \beta_1 v_1 + \beta_2 v_2 + \ldots + \beta_n v_n$$

它们看起来具有相似的表示:

$$\mathbf{v} = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T,$$

$$\mathbf{p} = [\beta_1, \beta_2, \ldots, \beta_n]^T,$$

这导致点与向量很容易混淆在一起



统一的表示

如果定义$0 \cdot \mathrm{P} = \mathbf{0}$(零向量)，$1 \cdot \mathrm{P} = \mathrm{P}$，那么

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n$$
$$= [v_1, v_2, \ldots, v_n, O][\alpha_1, \alpha_2, \ldots, \alpha_n, 0]^T$$
$$\mathrm{P} = O + \beta_1 v_1 + \beta_2 v_2 + \ldots + \beta_n v_n$$
$$= [v_1, v_2, \ldots, v_n, O][\beta_1, \beta_2, \ldots, \beta_n, 1]^T$$

从而得到n+1维齐次坐标(homogeneous coordinates)表示

$$\mathbf{v} = [\alpha_1, \alpha_2, \ldots, \alpha_n, 0]^T$$
$$\mathbf{p} = [\beta_1, \beta_2, \ldots, \beta_n, 1]^T$$

齐次坐标

四维齐次坐标的一般形式为

$$\mathbf{p} = [x, y, z, w]^T,$$

可以通过透视除法给出三维点(当$w \neq 0$):

$$x \leftarrow x/w, \quad y \leftarrow y/w, \quad z \leftarrow z/w$$

当$w = 0$时，表示对应的是一个向量

注意：齐次坐标表示中把四维空间中过原点的一条直线对应于三维空间中的一个点

齐次坐标与计算机图形学

- 齐次坐标是所有计算机图形系统的关键
  - 所有标准变换(旋转、平移、放缩)都可以应用4×4阶矩阵的乘法实现
  - 硬件流水线体系采用四维表示
  - 对于正交投影，可以通过$w = 0$保证向量，$w = 1$保证点
  - 对于透视投影，需要进行特别的处理：透视除法(perspective division)

变换坐标系

- 考虑同一个向量w相对于两组不同基**v**和**u**的表示。假设表示分别是

$$\mathbf{a} = [\alpha_1, \alpha_2, \alpha_3]^T \quad \text{基:} \quad \mathbf{v} = [v_1, v_2, v_3]^T$$

$$\mathbf{b} = [\beta_1, \beta_2, \beta_3]^T \quad \text{基:} \quad \mathbf{u} = [u_1, u_2, u_3]^T$$

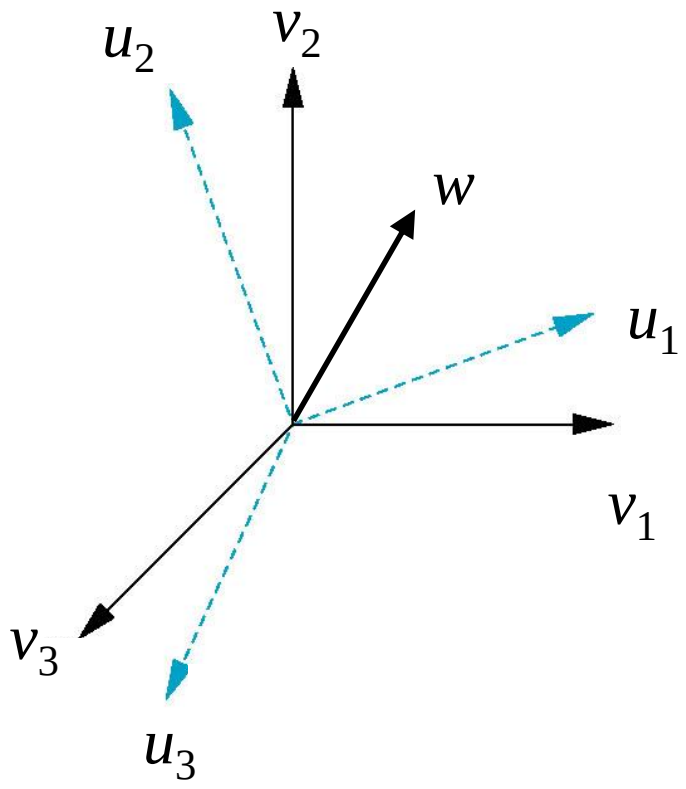
其中

$$w = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 = [v_1, v_2, v_3]\,[\alpha_1, \alpha_2, \alpha_3]^T$$
$$= \beta_1 u_1 + \beta_2 u_2 + \beta_3 u_3 = [u_1, u_2, u_3]\,[\beta_1, \beta_2, \beta_3]^T$$

用第一组基表示第二组基

每个基向量u_1,u_2,u_3都可以用第一组基表示出来

$$u_1 = \gamma_{11}v_1 + \gamma_{12}v_2 + \gamma_{13}v_3$$
$$u_2 = \gamma_{21}v_1 + \gamma_{22}v_2 + \gamma_{23}v_3$$
$$u_3 = \gamma_{31}v_1 + \gamma_{32}v_2 + \gamma_{33}v_3$$



矩阵形式

所有系数定义了一个3×3阶矩阵

$$\mathbf{M} = \begin{bmatrix} \gamma_{11} & \gamma_{21} & \gamma_{31} \\ \gamma_{12} & \gamma_{22} & \gamma_{32} \\ \gamma_{13} & \gamma_{23} & \gamma_{33} \end{bmatrix} = \left[\mathbf{u}_1 \mid \mathbf{u}_2 \mid \mathbf{u}_3\right]$$

这样两组基就可以如下联系在一起

$$\mathbf{u}^T = \mathbf{v}^T \mathbf{M}$$

M称为坐标系变换的矩阵表示

坐标系变换

$$w = \beta_1 u_1 + \beta_2 u_2 + \beta_3 u_3 = \mathbf{u}^T \mathbf{b} = \mathbf{v}^T \mathbf{M}\, \mathbf{b}$$
$$= \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 = \mathbf{v}^T \mathbf{a}$$

从而，有

$$\mathbf{a} = \mathbf{M}\, \mathbf{b}$$

变换矩阵$\mathbf{T} = \mathbf{M}^{-1}$把旧表示$\mathbf{a}$变换到新表示$\mathbf{b}$

$$\mathbf{b} = \mathbf{T}\, \mathbf{a}$$

坐标系变换的例子

向量$w = v_1+2v_2+3v_3$相对于基v_1,v_2,v_3的表示为

$$\mathbf{a} = [1, 2, 3]^T$$

构造一组新的基u_1,u_2,u_3

$$u_1 = v_1$$

$$u_2 = v_1 + v_2$$

$$u_3 = v_1 + v_2 + v_3$$

坐标系变换的例子

矩阵

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{T} = \mathbf{M}^{-1} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

向量$w = v_1+2v_2+3v_3$相对于基u_1,u_2,u_3新表示为

$$\mathbf{b} = \mathbf{T}\,\mathbf{a} = [-1, -1, 3]^T$$

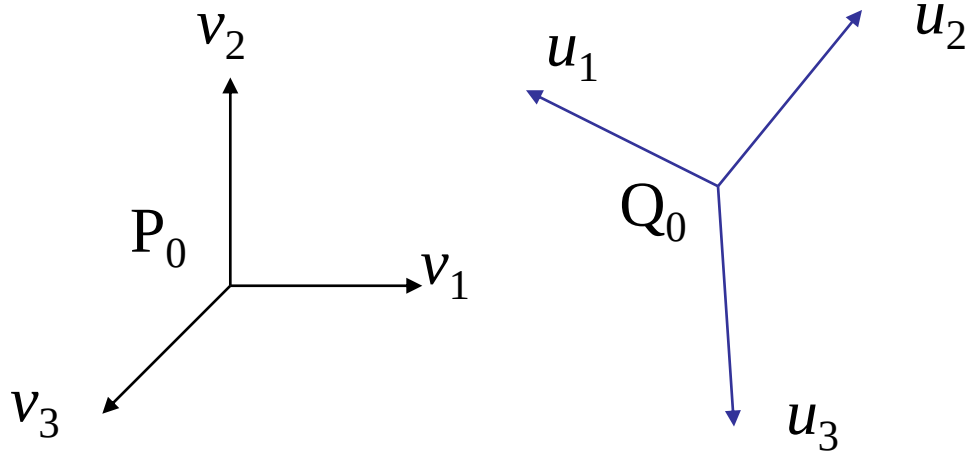
即$w = -u_1-u_2+3u_3$

标架变换

- 可以对同时表示点与向量的齐次坐标进行类似的操作
- 考虑两个标架

  (P_0, v_1, v_2, v_3)

  (Q_0, u_1, u_2, u_3)



- 任何点和向量都可以用它们中的一个表示出来
- 用P_0, v_1, v_2, v_3来表示Q_0, u_1, u_2, u_3

用一个标架表示另一个标架

- 把基的变换方法推广，有

$$
\begin{aligned}
u_1 &= \gamma_{11}v_1 + \gamma_{12}v_2 + \gamma_{13}v_3 \\
u_2 &= \gamma_{21}v_1 + \gamma_{22}v_2 + \gamma_{23}v_3 \\
u_3 &= \gamma_{31}v_1 + \gamma_{32}v_2 + \gamma_{33}v_3 \\
\mathrm{Q}_0 &= \gamma_{41}v_1 + \gamma_{42}v_2 + \gamma_{43}v_3 + \mathrm{P}_0
\end{aligned}
$$

$$
\begin{bmatrix} u_1 & u_2 & u_3 & \mathrm{Q}_0 \end{bmatrix} = \begin{bmatrix} v_1 & v_2 & v_3 & \mathrm{P}_0 \end{bmatrix} \mathbf{M}
$$

这里，4x4阶矩阵**M**为：

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

表示的变换

在两个标架中任意点和向量具有同样形式的表示

在第一个标架中：$\mathbf{a} = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]^T$

在第二个标架中：$\mathbf{b} = [\beta_1, \beta_2, \beta_3, \beta_4]^T$

其中表示的是点时，$\alpha_4 = \beta_4 = 1$；表示的是向量时，$\alpha_4 = \beta_4 = 0$，并且

$$\mathbf{a} = \mathbf{M}\,\mathbf{b}$$

$$\mathbf{b} = \mathbf{M}^{-1}\,\mathbf{a} = \mathbf{T}\,\mathbf{a}$$

矩阵**M**是4×4阶，定义了齐次坐标下的一个仿射变换

仿射变换

- 每个线性变换等价于一次标架改变
- 所有的仿射变换保持共线性
- 然而，一个仿射变换只具有12个自由度(degrees of freedom)，因为所有仿射变换只是由4×4阶矩阵定义的线性变换的子集
  - 矩阵的16个元素中有四个元素是固定的，即最后一行是[0,0,0,1]

标架变换的例子

新标架(Q_0, u_1, u_2, u_3)在原标架(P_0, v_1, v_2, v_3)中的表示为

$$u_1 = v_1$$
$$u_2 = v_1 + v_2$$
$$u_3 = v_1 + v_2 + v_3$$
$$Q_0 = v_1 + 2v_2 + 3v_3 + P_0$$

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

标架变换的例子

$$\mathbf{T} = \mathbf{M}^{-1} = \begin{bmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- 点(1, 2, 3)在原标架中表示为$\mathbf{p} = [1, 2, 3, 1]^T$，新表示为$\mathbf{p}' = \mathbf{T}\,\mathbf{p} = [0, 0, 0, 1]^T$
- 向量(1, 2, 3)在原标架中表示为$\mathbf{a} = [1, 2, 3, 0]^T$，新表示为$\mathbf{b} = \mathbf{T}\,\mathbf{a} = [-1, -1, 3, 0]^T$

OpenGL中的标架

- 根据绘制流水线中出现的先后顺序:
  - 对象或模型标架 4D
  - 世界标架 4D
  - 眼或照相机标架 4D
  - 裁剪标架 4D
  - 规范化设备标架 3D
  - 窗口或屏幕标架 3D or 2D

OpenGL中的表示

- 在模型标架中定义各个对象
- 把对象放置到应用程序场景（世界标架）中，用模型变换改变它们的大小、方向和位置
- 用视图变换把场景变换到照相机标架中
  - 标架变换对应到一个仿射变换，从模型坐标->世界坐标->眼坐标的变换都可以用4x4矩阵表示
  - OpenGL把模型变换和视图变换合并为模-视变换，对应矩阵为模-视矩阵

OpenGL中的表示

- 投影变换把场景变换到裁剪标架中，视见体变换为裁剪立方体，然后把视见体外的对象从场景里裁剪掉
- 顶点的齐次裁剪坐标经过透视除法，即用w分量去除其他分量，得到三维的规范化设备坐标表示
- 根据视口信息，把规范化设备坐标变换为三维的窗口坐标
  - 窗口坐标单位是像素，保留了深度信息
- 去掉深度分量，就得到了二维的屏幕坐标

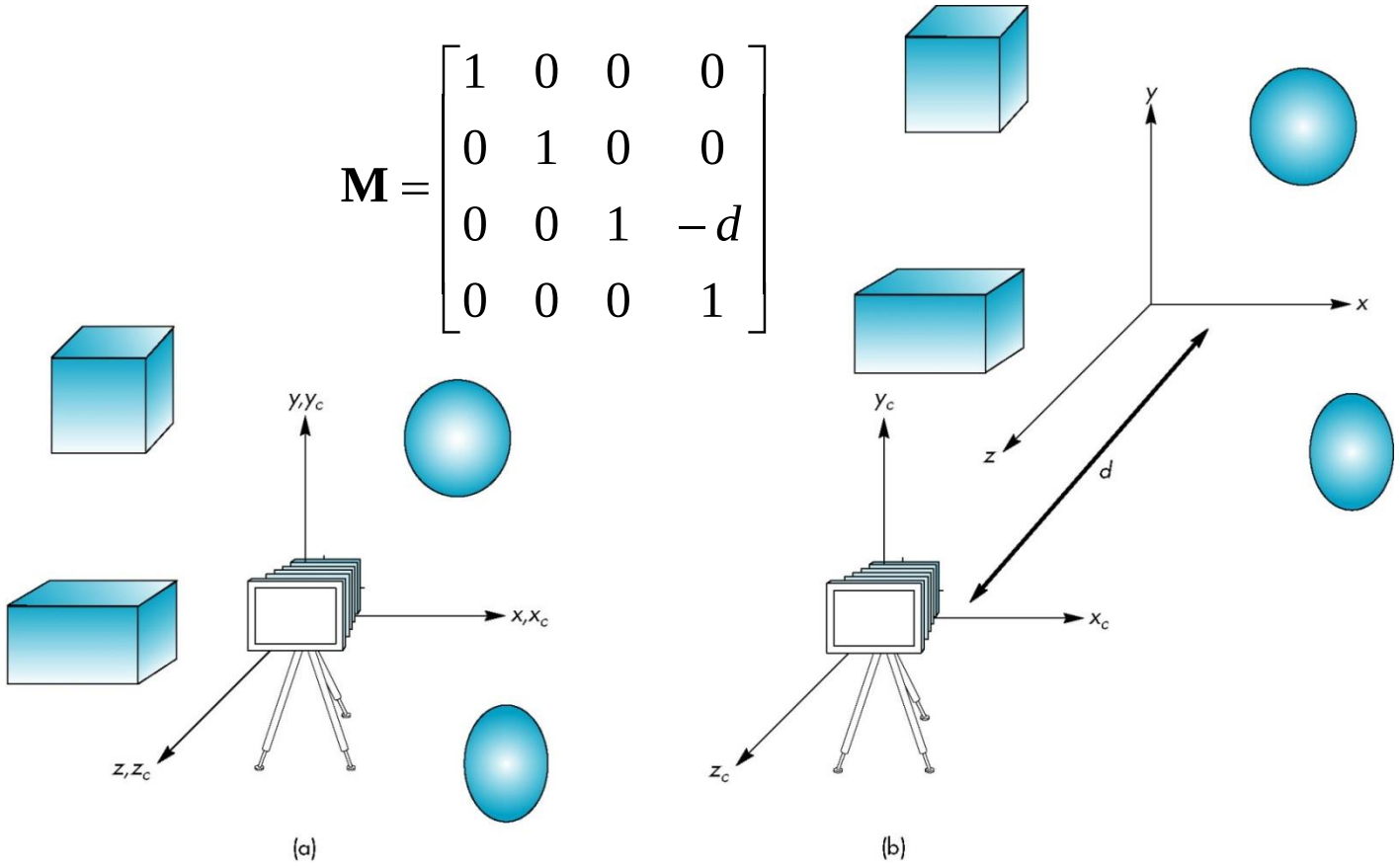
世界标架与照相机标架

- 当提及表示的时候，所指的是由n个标量构成的有序数组，即n元组
- 这样标架的改变就是由一个4×4阶矩阵定义
- 在OpenGL中开始的基本标架是世界标架
- 最终我们是在照相机标架中表示几何体，这是用模型-视图矩阵进行变换的
  - 照相机位于眼标架的原点
  - y轴正方向：照相机的观察正向
  - z轴负方向：照相机正对的方向
  - x轴与y轴、z轴正交，构成右手标架
- 初始状态时这两个标架是相同的 (**M** = **I**)

照相机的移动

当物体位于z=0平面两侧时，我们必须移动照相机标架：$(x, y, z) \rightarrow (x, y, z\text{-}d)$

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



5.1 几何

5.2 表示

5.3 变换

5.3 变换

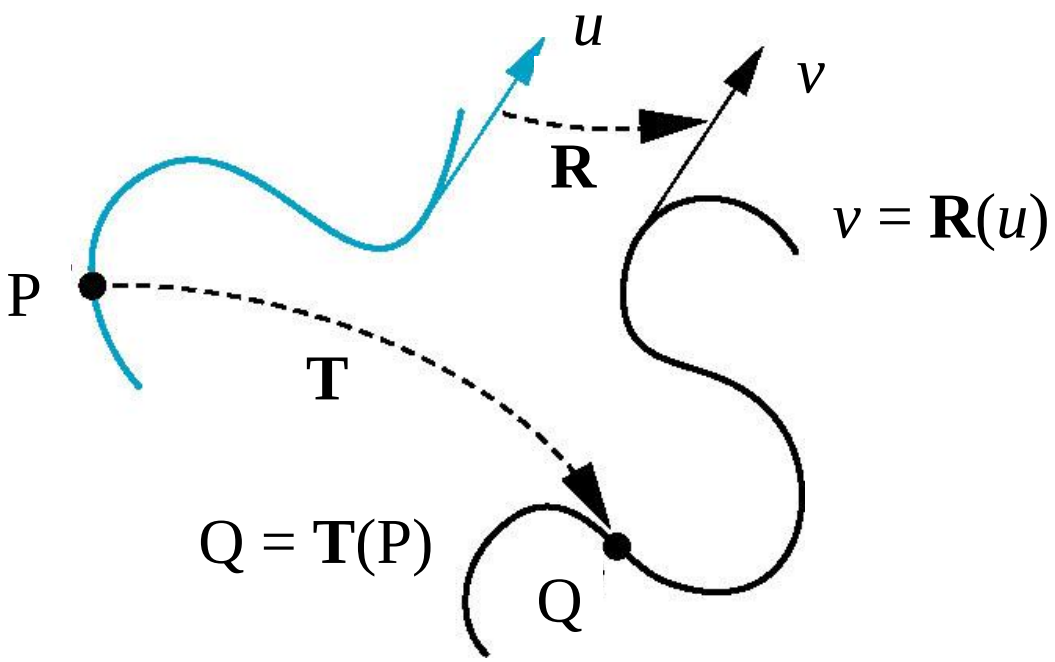
- 线性变换与仿射变换
- 标准变换
  - 平移 Translation
  - 旋转 Rotation
  - 缩放 Scaling
  - 错切 Shear
- 复杂变换

记号

- 在后面的处理中，既会考虑变换的坐标无关的表示，也会考虑在特定标架下的表示
  - P, Q, R: 在仿射空间中的点
  - *u*, *v*, *w*: 在仿射空间中的向量
  - α, β, γ: 标量
  - **p**, **q**, **r**: 点的表示
    - 在齐次坐标中为由四个标量构成的数组
  - **u**, **v**, **w**: 向量的表示
    - 在齐次坐标中为由四个标量构成的数组

一般变换

- 所谓变换就是把点映射到其它点，把向量映射到其它向量



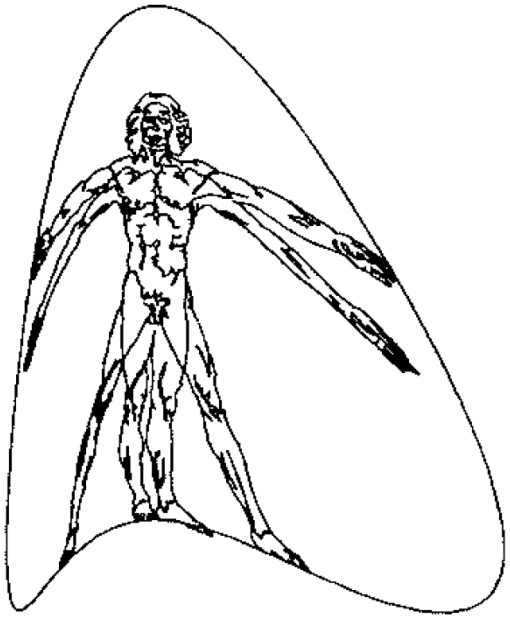
一般变换

- 用齐次坐标表示，变换可定义为一个函数f

$$\mathbf{q} = f(\mathbf{p}) \quad \text{点}$$

$$\mathbf{v} = f(\mathbf{u}) \quad \text{向量}$$

  - 在一般变换下，直线的像是由直线上每个点的像构成的，像一般不再是一条直线
  - 当变换是连续的时候，直线的像就是一条连续曲线



线性变换

- f 是线性函数，即

$$f(\alpha p + \beta q) = \alpha f(p) + \beta f(q)$$

  - 变换的线性组合等于线性组合的变换
- 线性变换可用矩阵乘法表示为：

$$\mathbf{v} = \mathbf{C}\mathbf{u}$$

  - **C**是方阵，**u**和**v**是点（向量）变换前后的表示
  - **C**非奇异时，对应于向量空间的一个坐标系变换
  - 几何变换：绕原点的旋转、缩放、错切
  - 三维空间中，**C**是3x3方阵

仿射变换

- 仿射变换：包含平移的线性变换
  - 平移、绕非原点的旋转
- 为了表示仿射变换，需要使用齐次坐标
- 在变换矩阵中增加一列与一行，除右下角的元素为1外其它部分填充为0，通过这种方法，所有的线性变换都可以转换为仿射变换。

仿射变换

- 三维空间中，仿射变换矩阵为4x4方阵

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

  - 四维齐次空间的线性变换
  - 齐次坐标的w分量保持不变
  - 点的仿射变换有12个自由度
  - 向量的仿射变换只有9个自由度

仿射变换

- 仿射变换把直线映射成直线
- 假定直线的方程为:

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

其中， P_0是一个点， d是一个向量。

- 在某个标架下，直线可以表示为:

$$\mathbf{p}(\alpha) = \mathbf{p}_0 + \alpha\, \mathbf{d}$$

其中，$\mathbf{p}_0$和$\mathbf{d}$分别是P_0和d在这个标架下的表示。

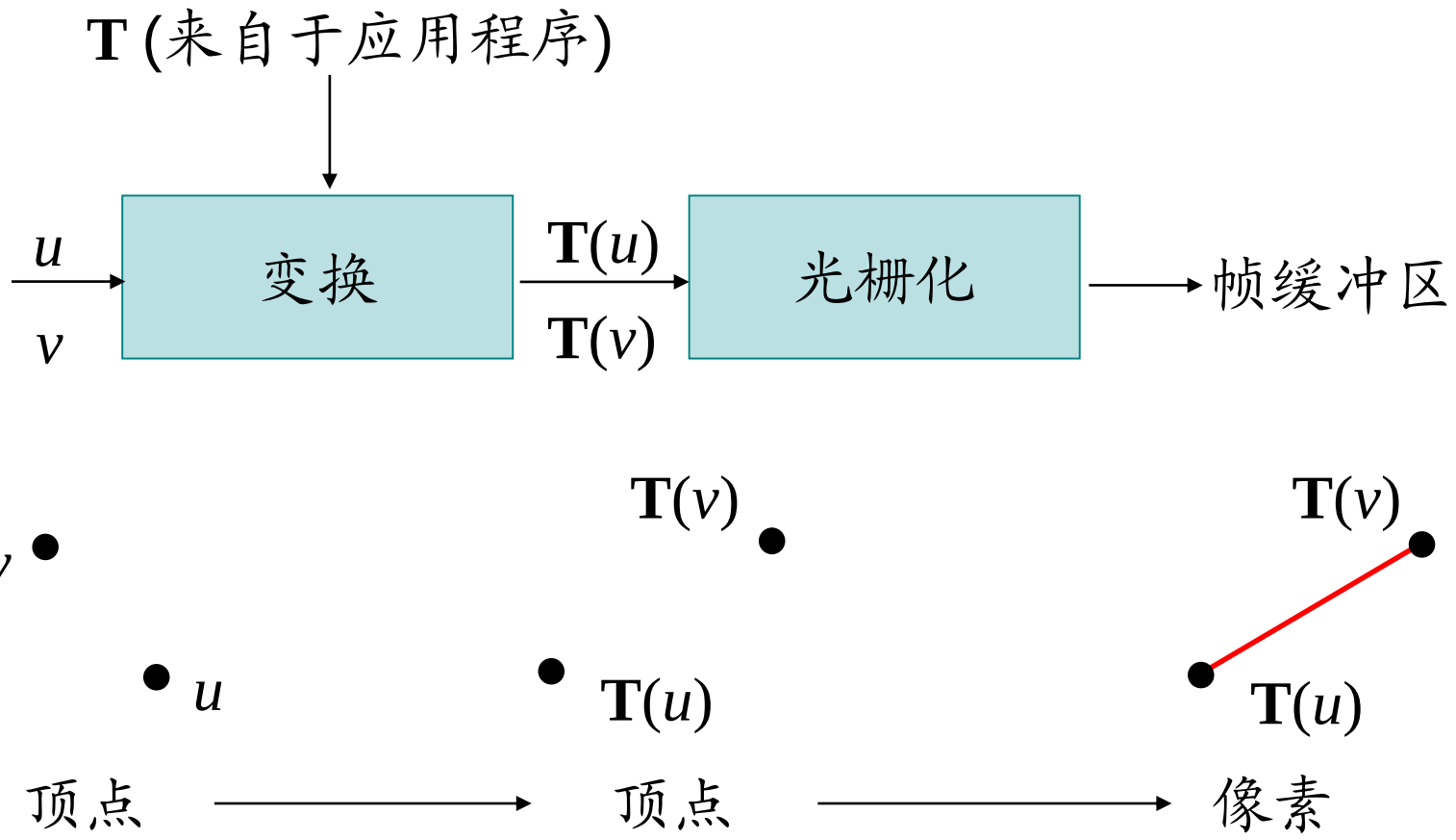
- 对任意仿射变换矩阵$\mathbf{C}$，有

$$\mathbf{C}\,\mathbf{p}(\alpha) = \mathbf{C}\,\mathbf{p}_0 + \alpha\,\mathbf{C}\,\mathbf{d}$$

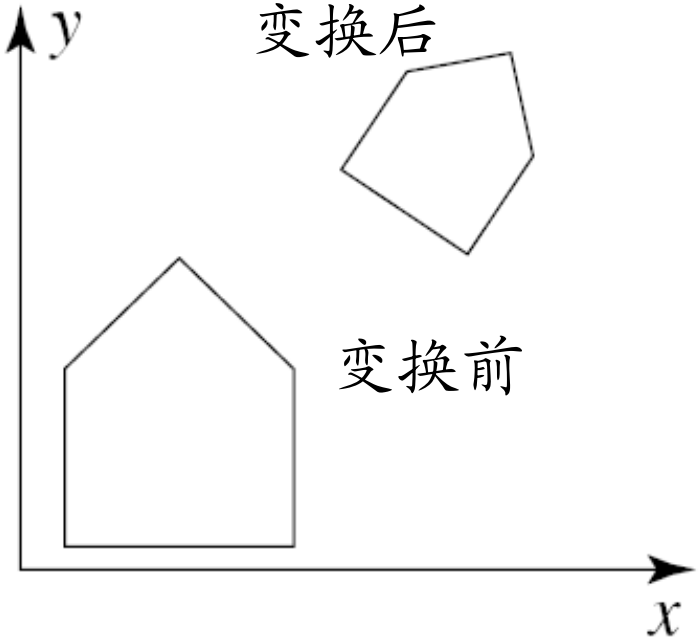
仿射变换

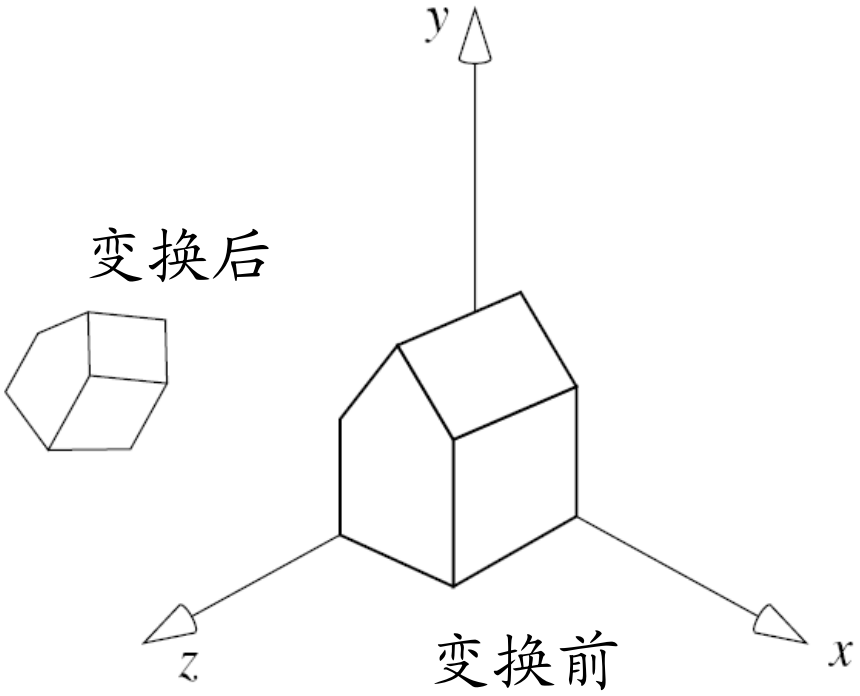
- 保持共线性
- 许多物理上重要变换的特征
  - 刚体变换：旋转、平移
  - 缩放、错切
- 在图形学中的重要性来自于，此时我们只需要变换线段的两个端点，而由系统自动画出变换后两个端点间的线段

流水线实现



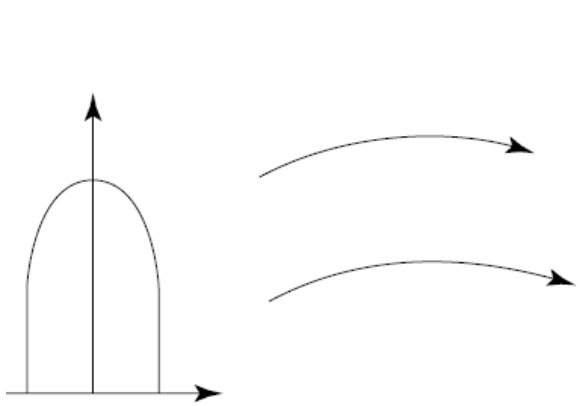
为什么需要变换?

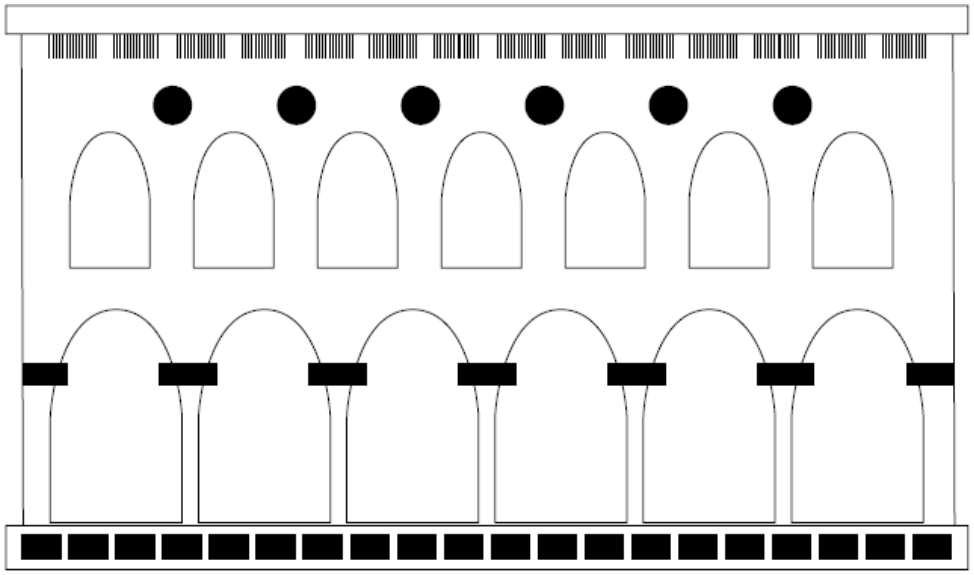




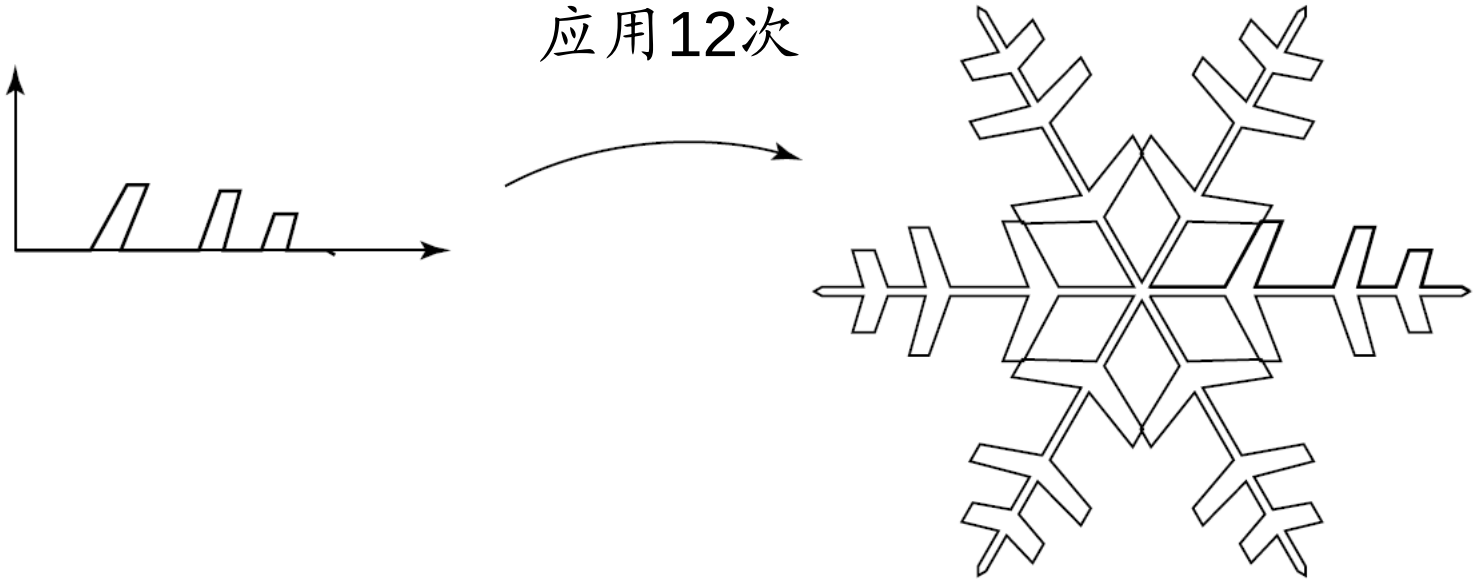
变换的作用之一

- 组合场景

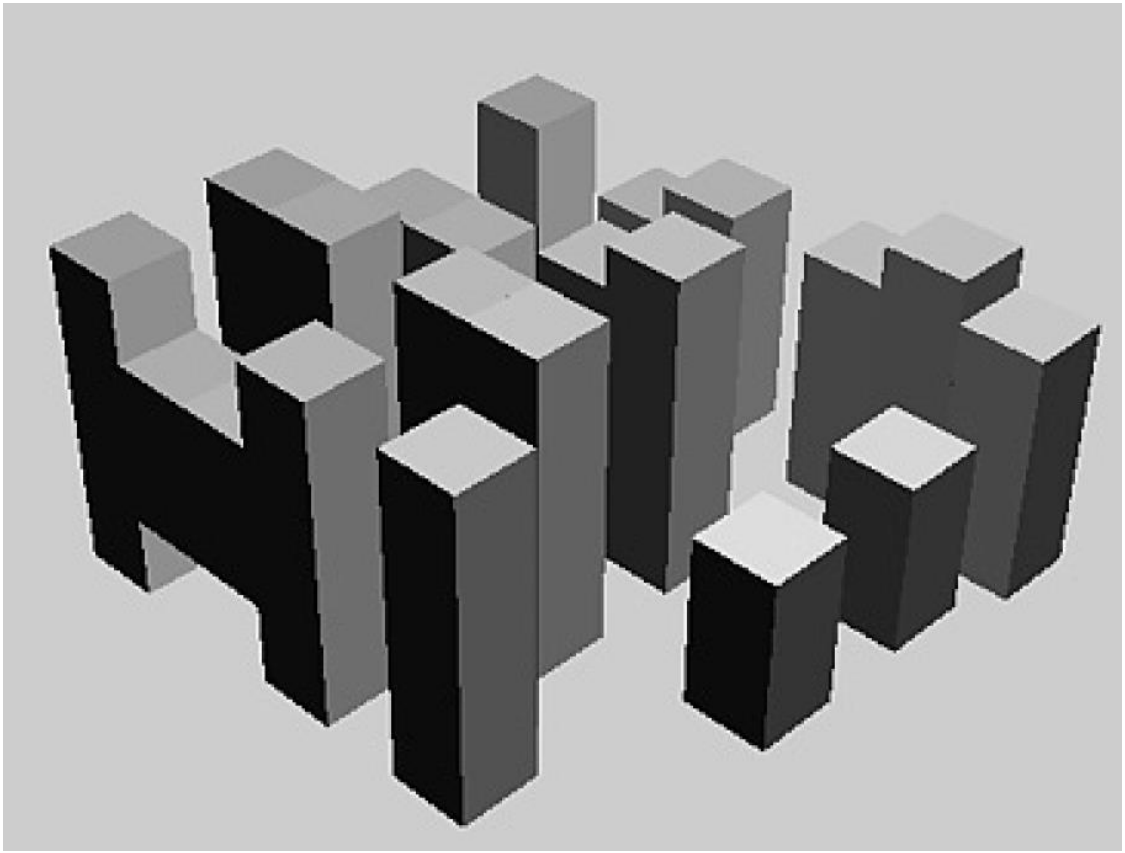




雪花的构造

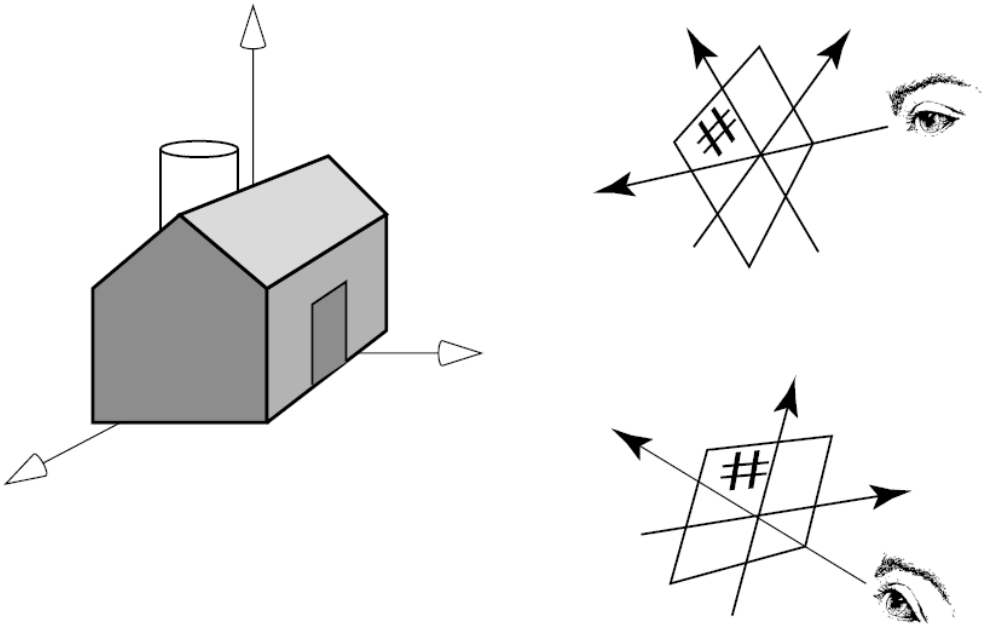


构造三维场景



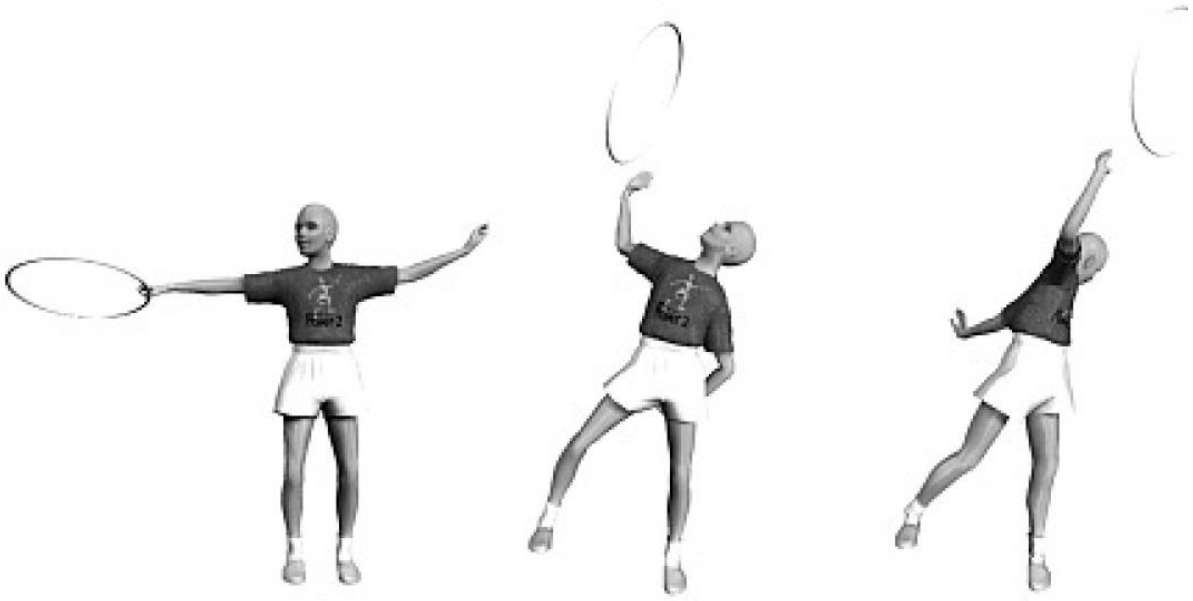
变换的作用之二

- 设计者可能需要从不同的角度查看同一个场景，此时自然的方法是对象不变，而变换照相机的位置



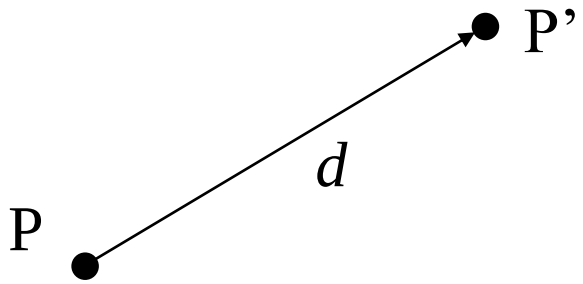
变换的作用之三

- 在计算机动画中，在相邻帧图像中，几个对象相对彼此的位置进行移动。这可以通过平移和旋转局部坐标系实现



平移

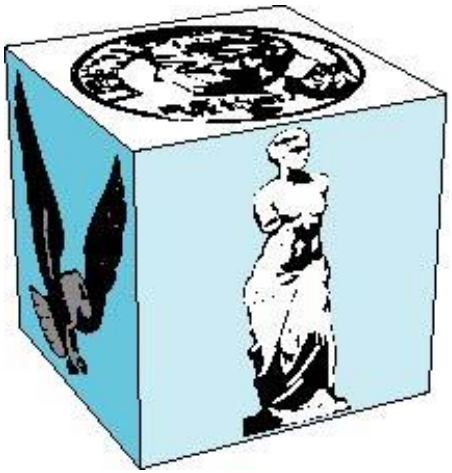
- 把一个点移到新的位置



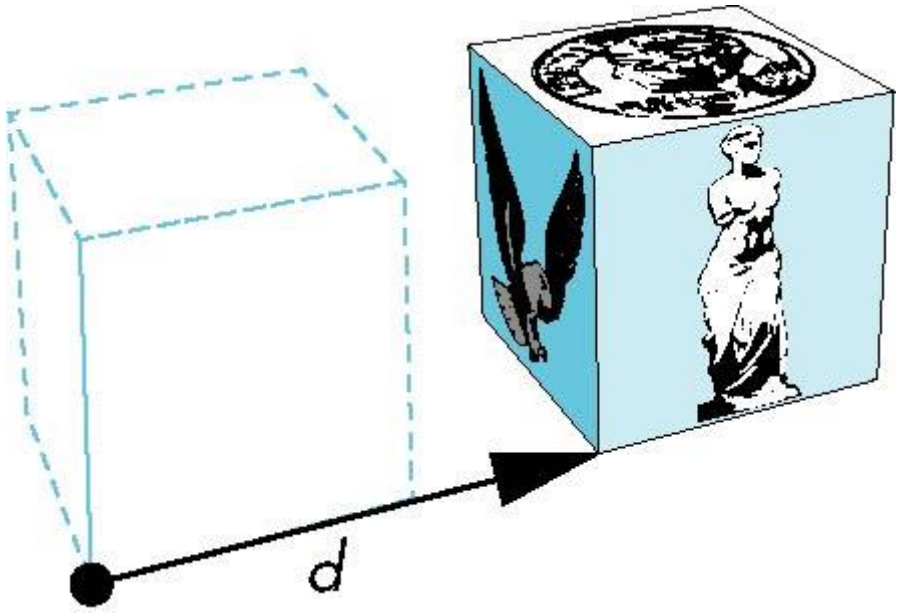
- 平移由一个向量d确定
  - 三个自由度
  - $P' = P + d$

对象的平移

把一个对象上的所有点沿同一方向移动相同距离



原始对象



平移后的对象

平移的表示

应用在某个标架中的齐次坐标表示

$\mathbf{p} = [x, y, z, 1]^T$

$\mathbf{p'} = [x', y', z', 1]^T$

$\mathbf{d} = [d_x, d_y, d_z, 0]^T$

那么 $\mathbf{p'} = \mathbf{p} + \mathbf{d}$ 或者

$x' = x + d_x,$

$y' = y + d_y,$

$z' = z + d_z.$

注意：这个表达式是四维的，而且表示的是

点 = 点+ 向量

平移矩阵

- 可以用在齐次坐标中一个4×4的矩阵**T**表示平移: $\mathbf{p}' = \mathbf{T}\,\mathbf{p}$, 其中

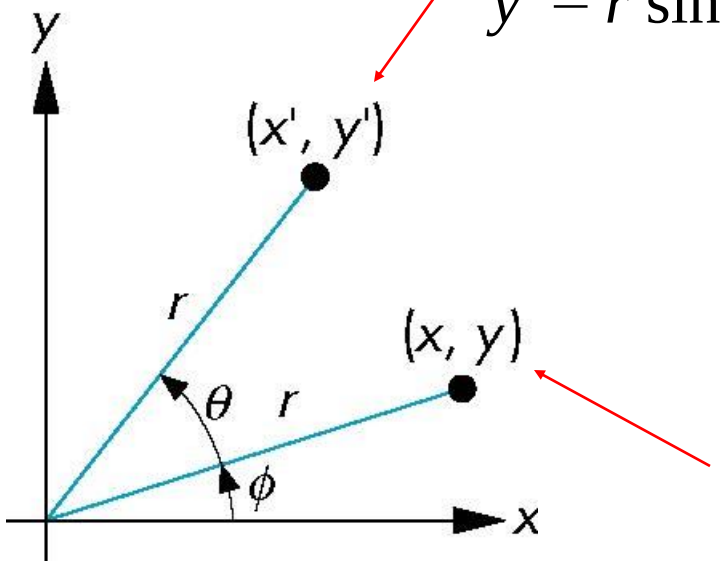
$$\mathbf{T} = \mathbf{T}(d_x, d_y, d_z) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- 这种形式更容易实现，因为所有的仿射变换都可以用这种形式表示，矩阵乘法可以复合在一起

二维旋转

- 考虑绕原点旋转θ度
  - 半径保持不变，角度增加了θ

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

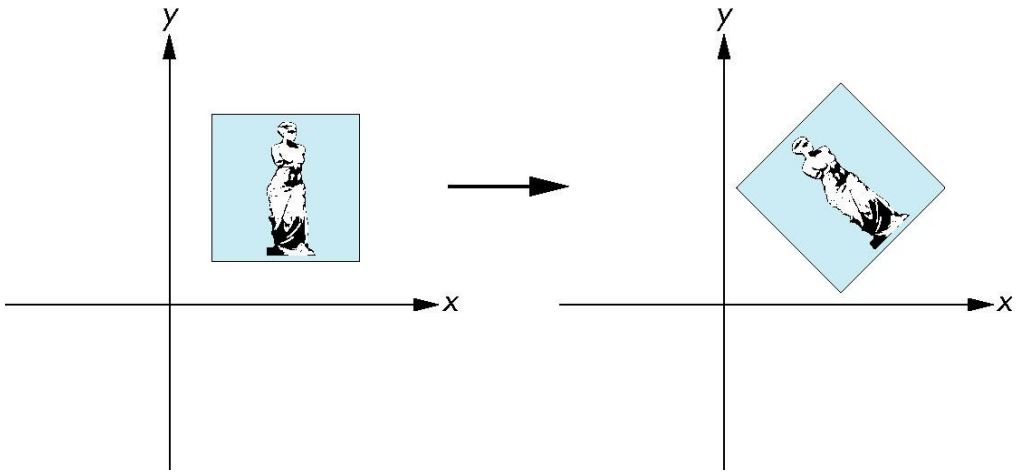


$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$

$$x = r\cos\phi$$
$$y = r\sin\phi$$

二维旋转

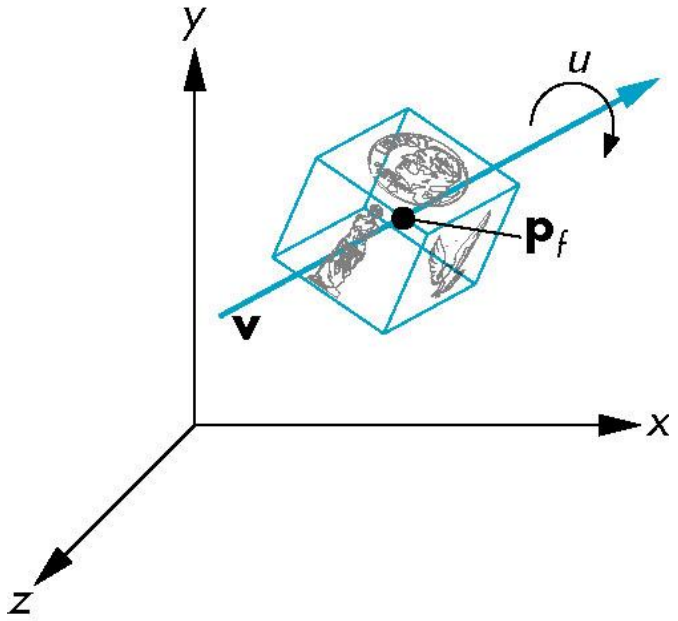
- 不动点（固定点）：绕对象中心旋转
- 旋转轴：等效于三维空间绕z轴旋转
- 旋转角：（从z轴正方向看）逆时针方向为正向



三维旋转

- 不动点$\mathbf{p}_f$
- 旋转轴向量$\mathbf{v}$
- 旋转角θ

- 几种特殊情形
  - 分别绕x,y,z轴的旋转
  - 绕过原点的轴旋转
  - 绕任一轴旋转



绕z轴的旋转

- 在三维空间中绕z轴旋转，点的z坐标不变
  - 等价于在z=常数的平面上进行二维旋转

$$x' = x\cos\theta - y\sin\theta$$

$$y' = x\sin\theta + y\cos\theta$$

$$z' = z$$

  - 其齐次坐标表示为

$$\mathbf{p}' = \mathbf{R}_z(\theta)\,\mathbf{p}$$

旋转矩阵

$$\mathbf{R}_z = \mathbf{R}_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

绕x轴和y轴的旋转

- 与绕z轴的旋转完全类似
  - 对于绕x轴的旋转，x坐标不变

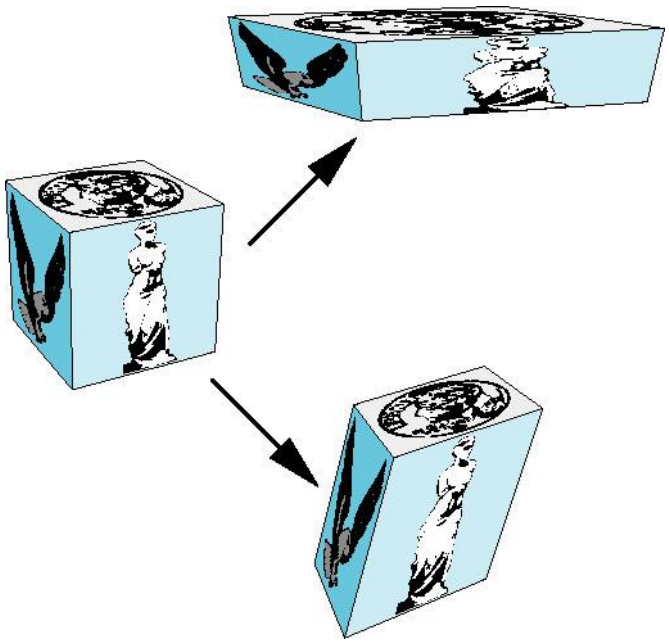
$$\mathbf{R}_x = \mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - 对于绕y轴的旋转，y坐标不变

$$\mathbf{R}_y = \mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

刚体变换

- 旋转与平移是两种刚体变换
  - 这两种变换的复合只能改变对象的位置与定向
  - 保角度和长度
- 其它的仿射变换会改变对象的形状和体积



缩放

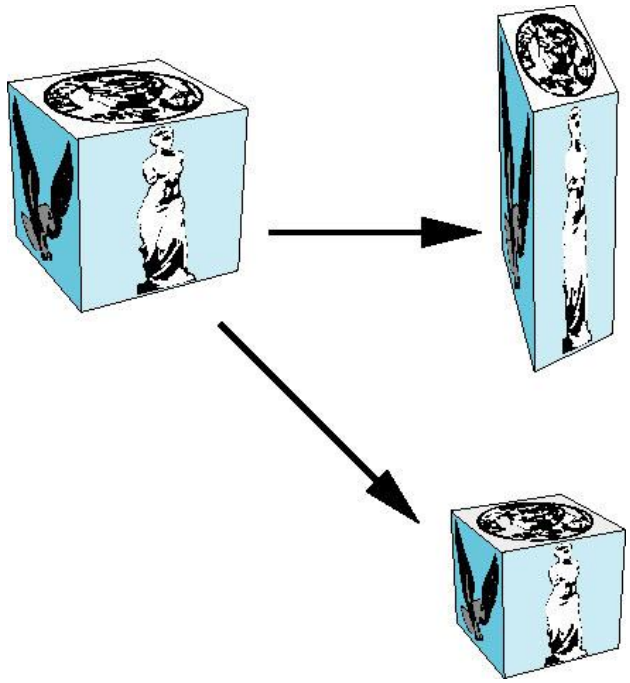
- 沿每个坐标轴伸展或收缩(原点为不动点)

$$x' = s_x x$$

$$y' = s_y y$$

$$z' = s_z z$$

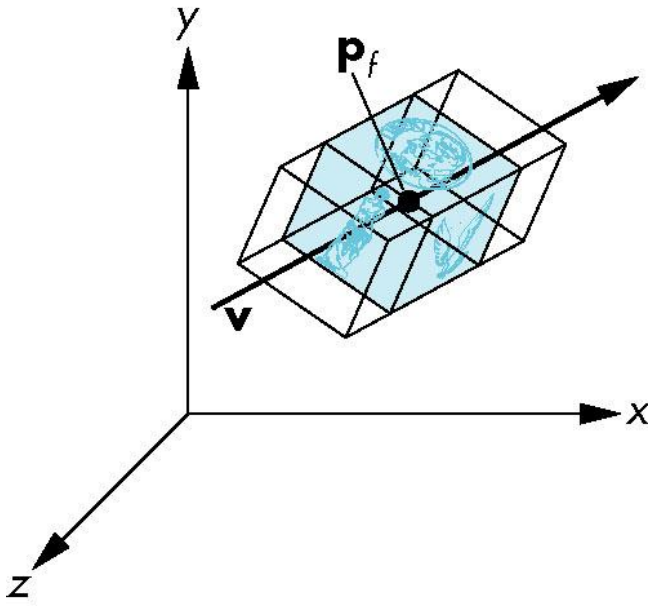
$$\mathbf{p}' = \mathbf{S}\,\mathbf{p}$$



$$\mathbf{S} = \mathbf{S}(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

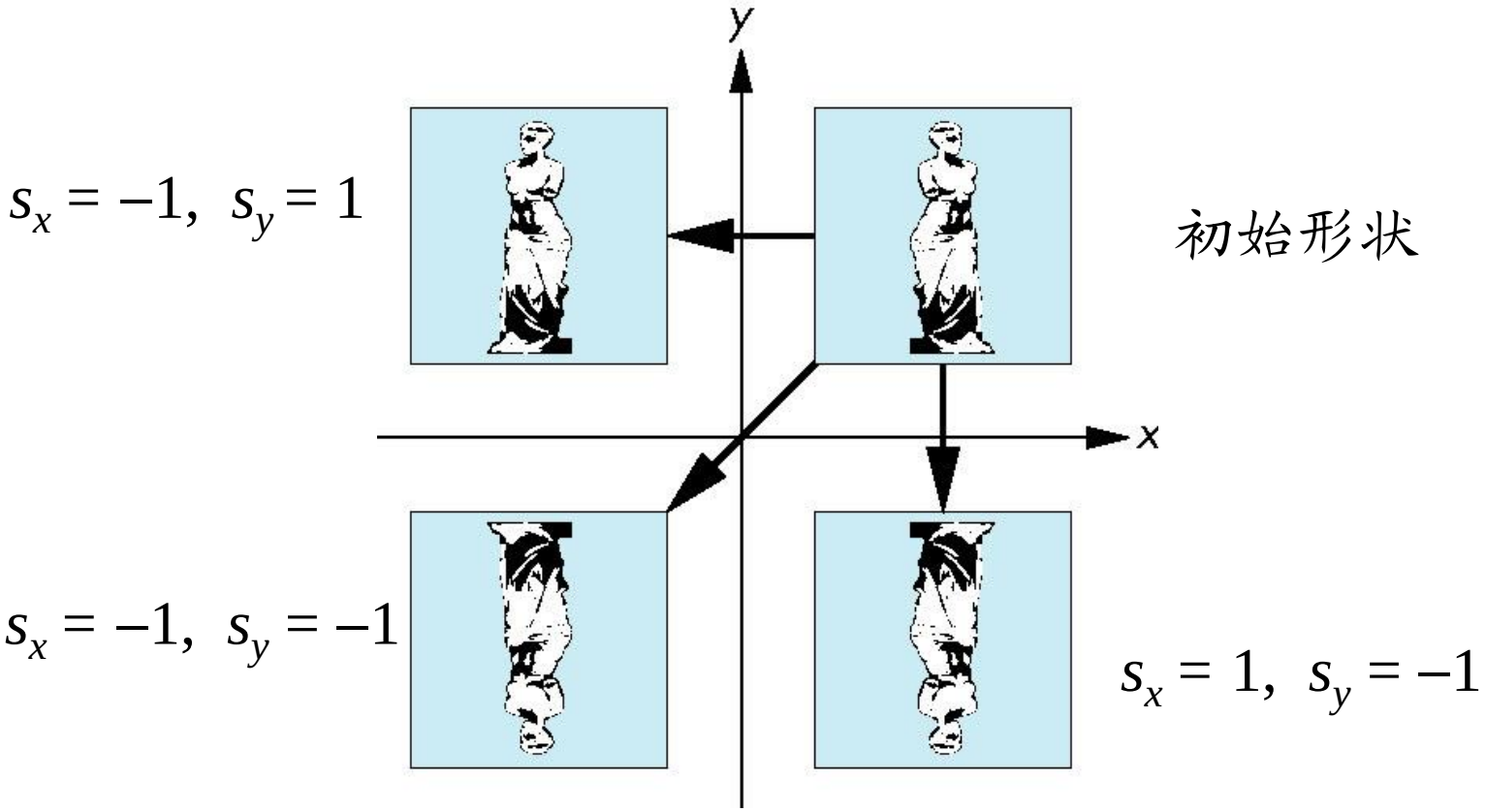
缩放因子

- 缩放变换有一个不动点
- 为了定义缩放变换，需要指定不动点、缩放方向和缩放因子
- 当缩放因子大于1时，对象在指定方向上变长



反射

- 对应于负的缩放因子

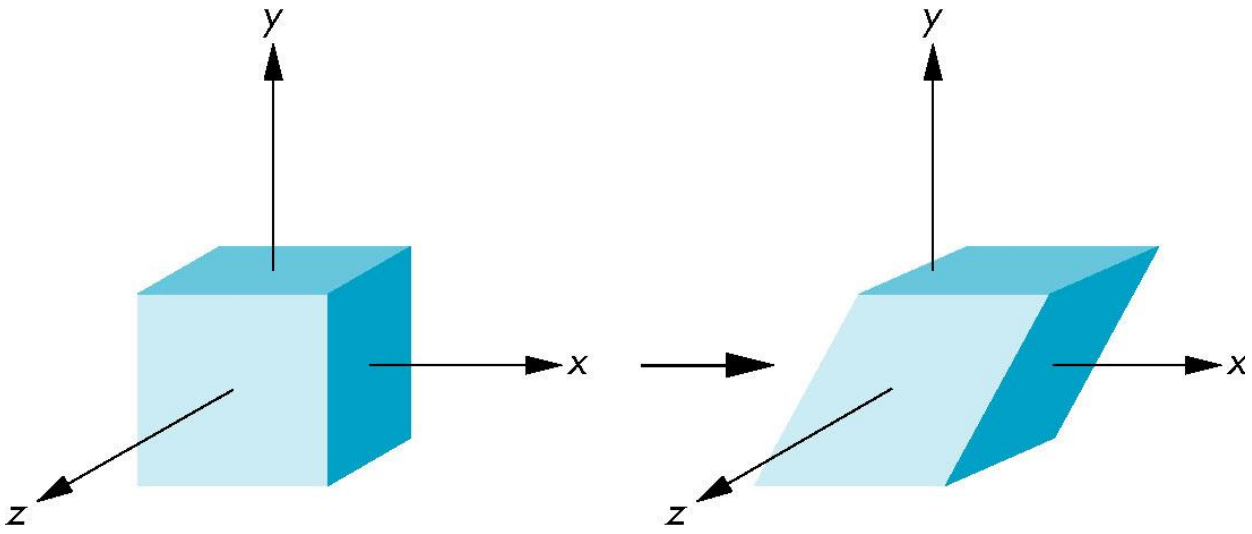


逆变换

- 虽然可以直接计算矩阵的逆，但根据几何意义可以给出各种变换的逆
  - 平移：$\mathbf{T}^{-1}(d_x, d_y, d_z) = \mathbf{T}(-d_x, -d_y, -d_z)$
  - 旋转：$\mathbf{R}^{-1}(\theta) = \mathbf{R}(-\theta)$
    - 对任一旋转矩阵成立
    - 注意$\cos(-\theta) = \cos\theta$, $\sin(-\theta) = -\sin\theta$, 从而
      $$\mathbf{R}^{-1}(\theta) = \mathbf{R}^{T}(\theta)$$
    - 旋转矩阵为正交矩阵
  - 缩放：$\mathbf{S}^{-1}(s_x, s_y, s_z) = \mathbf{S}(1/s_x, 1/s_y, 1/s_z)$

错切

- 另外一种实用的基本变换
- 等价于把面向相反方向拉



错切矩阵

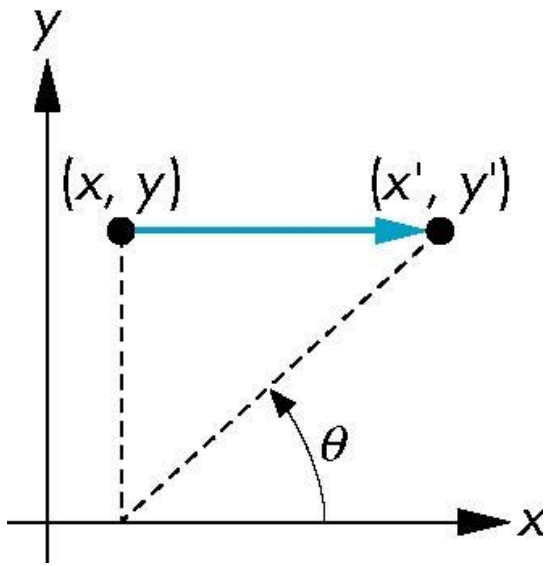
- 考虑沿x轴的错切

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

$$y' = y$$

$$z' = z$$

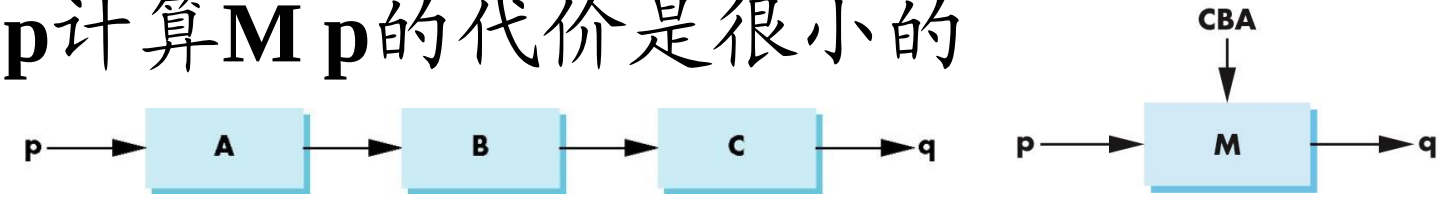
$$\mathbf{H}(\theta) = \begin{bmatrix} 1 & \cot\theta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



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

变换的复合

- 可以通过把旋转、平移与缩放矩阵相乘从而形成任意的仿射变换
- 由于对许多顶点应用同样的变换，因此构造矩阵**M = C B A**的代价相比于对许多顶点**p**计算**M p**的代价是很小的



- 难点在于如何根据应用程序的要求构造出满足要求的变换矩阵

变换的顺序

- 注意在右边的矩阵是首先被应用的矩阵
- 从数学的角度来说，下述表示是等价的

$$\mathbf{p'} = \mathbf{C}\,\mathbf{B}\,\mathbf{A}\,\mathbf{p} = \mathbf{C}\,(\mathbf{B}\,(\mathbf{A}\,\mathbf{p}))$$

- 有些书籍文献用行矩阵来表示点，变换的级联可以表示为：

$$\mathbf{p'}^T = (\mathbf{C}\,\mathbf{B}\,\mathbf{A}\,\mathbf{p})^T = \mathbf{p}^T\,\mathbf{A}^T\,\mathbf{B}^T\,\mathbf{C}^T$$

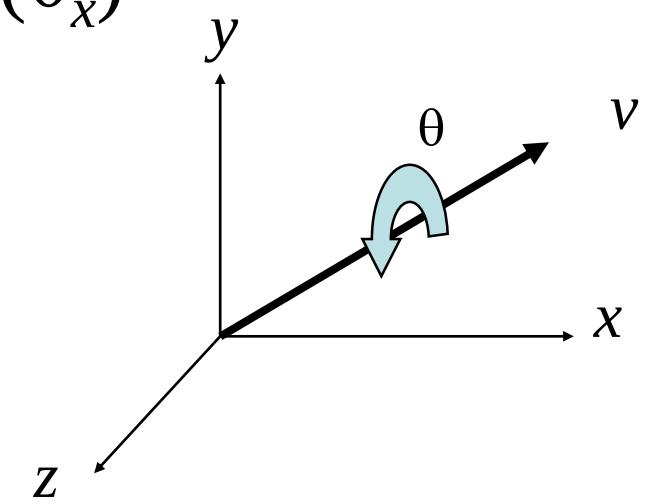
  - 这种形式的优点在于：书写变换的顺序和执行变换的顺序是一样的
- 变换的顺序是不可交换的
  - 矩阵乘法不满足交换律

绕原点的一般旋转

绕过原点任一轴旋转θ角可以分解为绕x, y, z轴旋转的复合

$$\mathbf{R}(\theta) = \mathbf{R}_z(\theta_z)\,\mathbf{R}_y(\theta_y)\,\mathbf{R}_x(\theta_x)$$

$\theta_x, \theta_y, \theta_z$称为Euler角

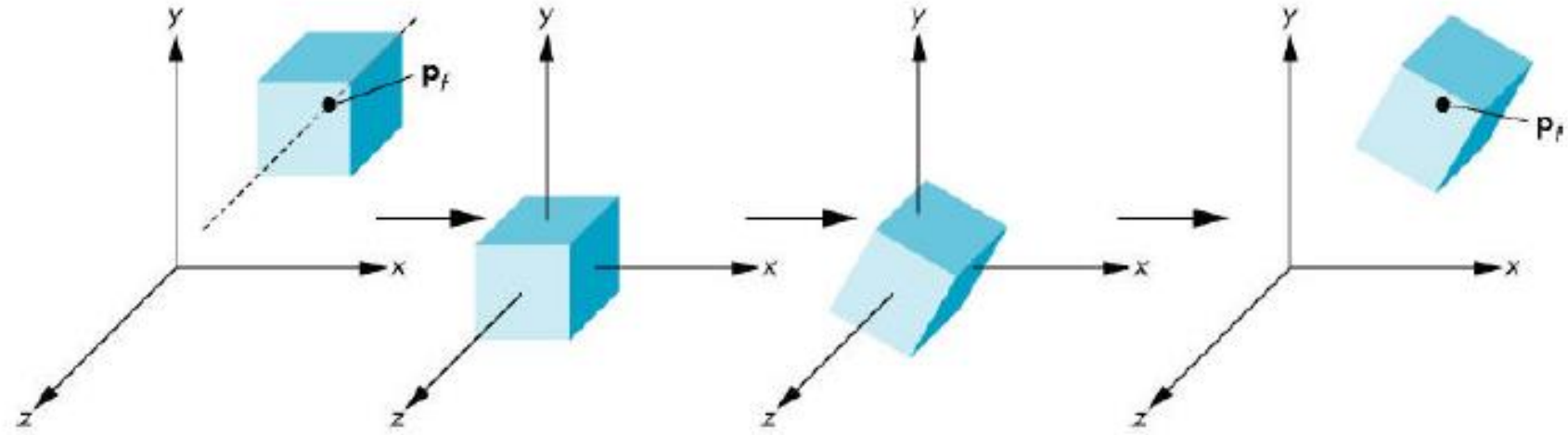


- 注意旋转没有交换性
- 可以用不同的旋转顺序，不同的旋转角度得到同样的效果

绕不同于原点的不动点旋转

- 把不动点移到原点 $\mathbf{T}(-\mathbf{p}_f)$
- 旋转 $\mathbf{R}(\theta)$
- 把不动点移回到原来位置 $\mathbf{T}(\mathbf{p}_f)$

$$\mathbf{M} = \mathbf{T}(\mathbf{p}_f)\,\mathbf{R}(\theta)\,\mathbf{T}(-\mathbf{p}_f)$$



绕任意轴的旋转

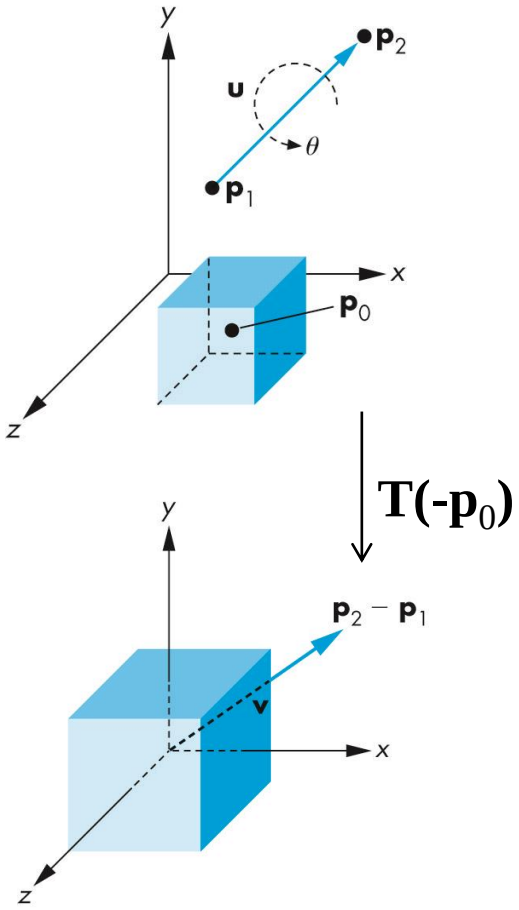
- 不动点：立方体中心$\mathbf{p}_0$
- 旋转轴方向向量：

$$\mathbf{u} = \mathbf{p}_2 - \mathbf{p}_1$$

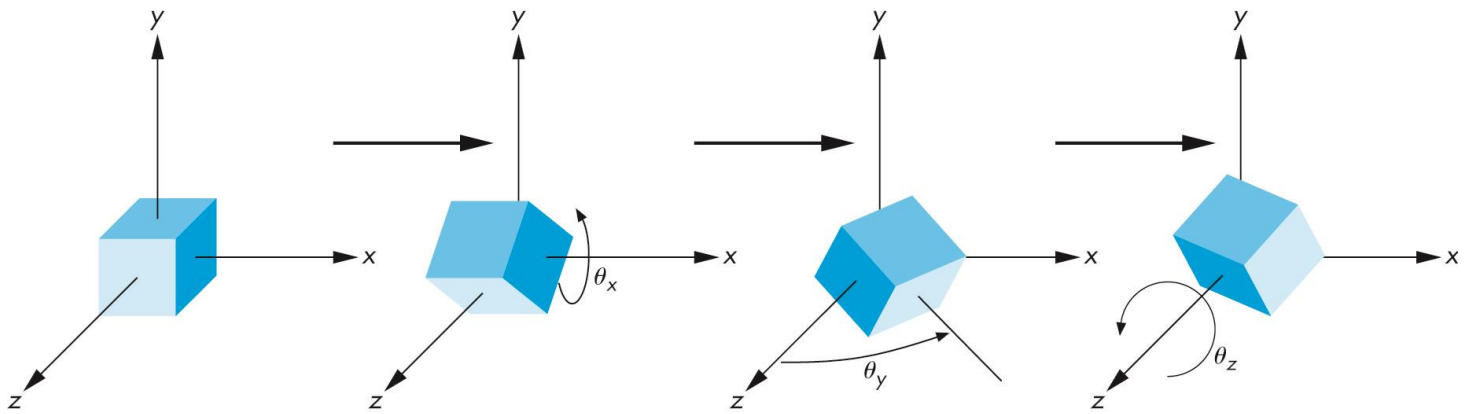
单位化为：

$$\mathbf{v} = \mathbf{u} / |\mathbf{u}| = [\alpha_x, \alpha_y, \alpha_z]^T$$

- $\mathbf{T}(-\mathbf{p}_0)$平移不动点到原点



绕任意轴的旋转



策略：先经过两次旋转使旋转轴**v**与z轴对齐，然后绕z轴旋转角度θ

$$\mathbf{R} = \mathbf{R}_x(-\theta_x)\,\mathbf{R}_y(-\theta_y)\,\mathbf{R}_z(\theta)\,\mathbf{R}_y(\theta_y)\mathbf{R}_x(\theta_x)$$

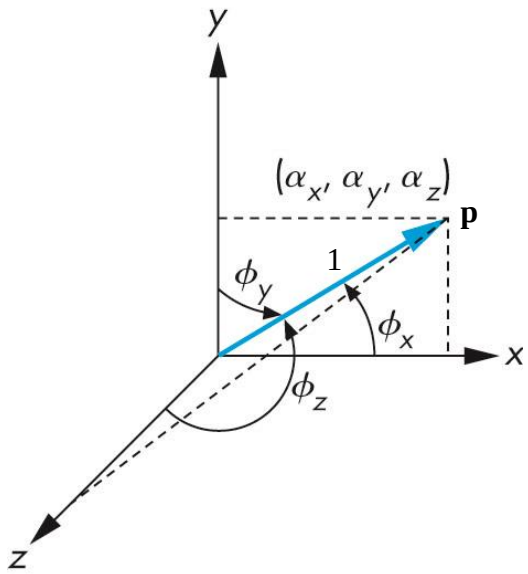
绕任意轴的旋转

从原点画一条线段到点**p** $(\alpha_x, \alpha_y, \alpha_z)$，从点**p**到坐标轴画垂线，得到三个方向角$\phi_x, \phi_y, \phi_z$:

$$\cos\phi_x = \alpha_x$$

$$\cos\phi_y = \alpha_y$$

$$\cos\phi_z = \alpha_z$$



只有两个方向角是独立的，因为

$$\cos^2\phi_x + \cos^2\phi_y + \cos^2\phi_z = 1$$

根据方向角计算θ_x和θ_y

绕任意轴的旋转

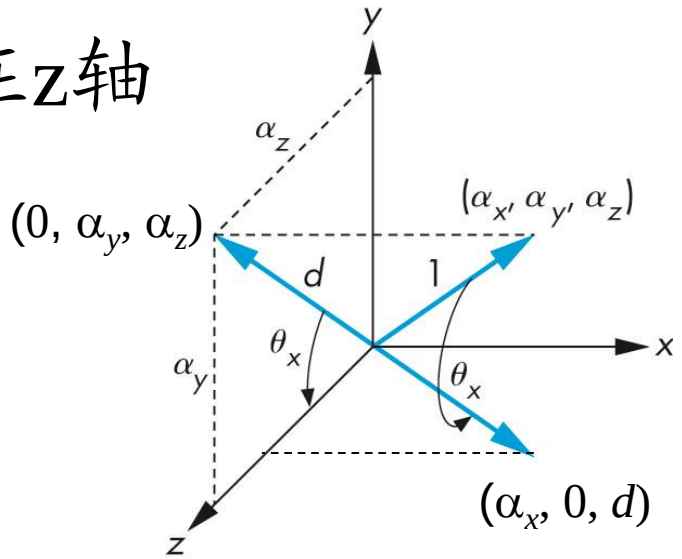
绕x轴的旋转把单位轴旋转到y=0平面，旋转前单位轴在x=0平面上投影线长度为

$$d = (\alpha_y^2 + \alpha_z^2)^{1/2}$$

旋转角度θ_x就是投影线与z轴的夹角

旋转后，单位轴的投影落在z轴

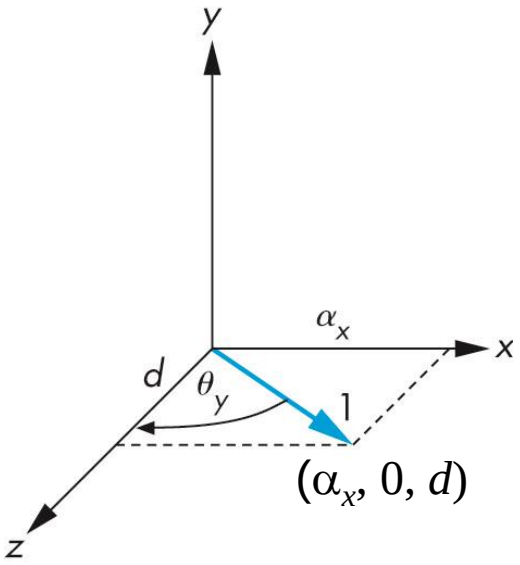
$$\mathbf{R}_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha_z/d & -\alpha_y/d & 0 \\ 0 & \alpha_y/d & \alpha_z/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



绕任意轴的旋转

类似地可求得$\mathbf{R}_y$，这里旋转角是绕y轴顺时针旋转:

$$\mathbf{R}_y(\theta_y)=\begin{bmatrix} d & 0 & -\alpha_x & 0 \\ 0 & 1 & 0 & 0 \\ \alpha_x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

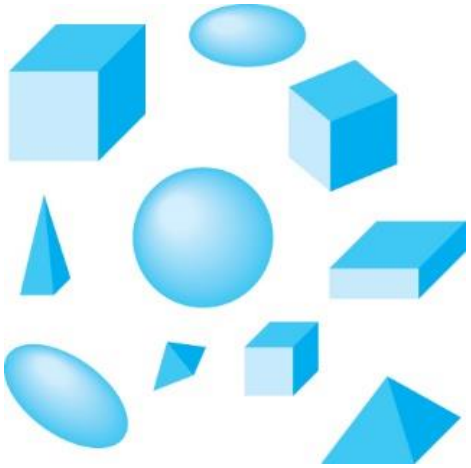


串乘所有矩阵，得到:

$$\mathbf{M} = \mathbf{T}(\mathbf{p}_0)\,\mathbf{R}_x(-\theta_x)\,\mathbf{R}_y(-\theta_y)\,\mathbf{R}_z(\theta)\,\mathbf{R}_y(\theta_y)\,\mathbf{R}_x(\theta_x)\,\mathbf{T}(-\mathbf{p}_0)$$

实例变换

- 建模一个由许多简单对象构成的场景
  - 方法一：直接指定对象的顶点使得每个对象出现在想要的位置并且具有期望的方向和大小
  - 方法二：定义一些中心在原点，相对于坐标轴定向的标准尺寸的对象。对象在场景中的每一次出现都是该对象原型的一个实例(instance)。通过应用一个仿射变换，即实例变换(instance transformation)使得对象具有期望的大小、方向和位置



实例变换

- 构建一个对象数据库，通过一个对象标识符及其对应实例变换的列表来描述场景
- 实例变换**M = TRS**
  - 缩放 **S**
  - 定向（旋转**R**）
  - 定位（平移**T**）