2018-2019年度第二学期 00106501

计算机图形学

童 伟 华　管理科研楼1205室

E-mail: tongwh@ustc.edu.cn

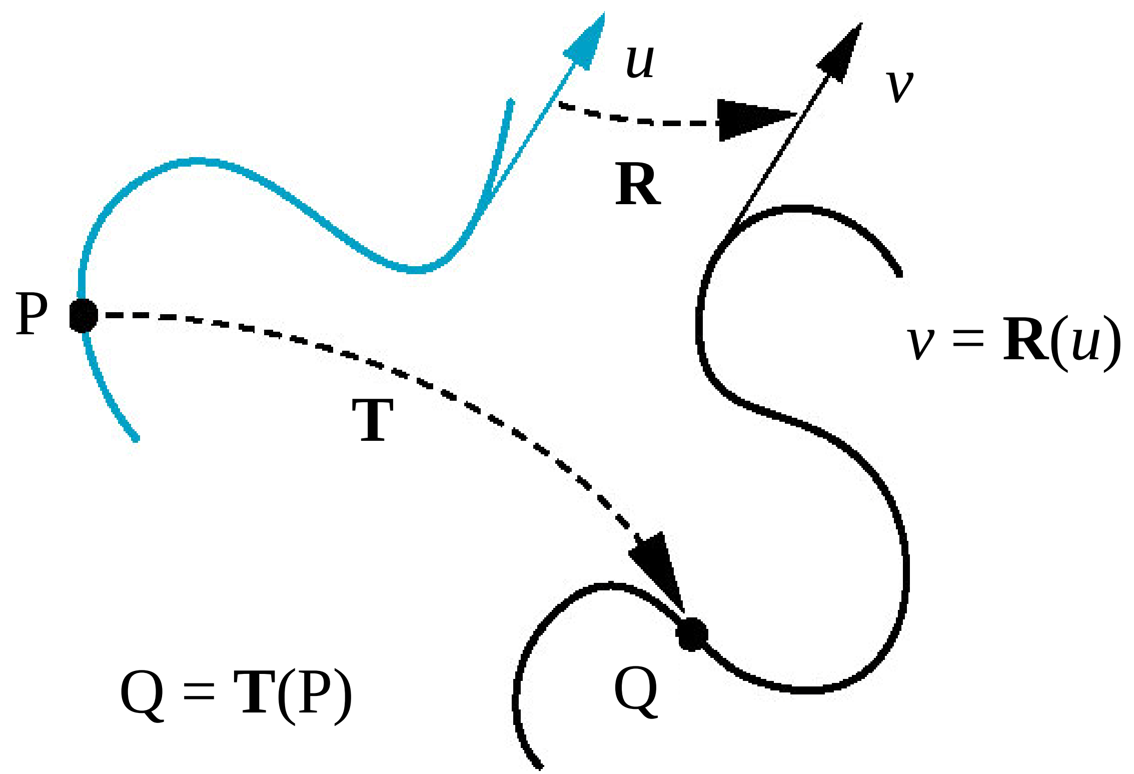
中国科学技术大学　数学科学学院

http://math.ustc.edu.cn/

第三节 变换

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



连续变换

- 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

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
  - 放缩、错切
- 数学定义：设映射 $\Phi:\mathbb{R}^3 \to \mathbb{R}^3$，若 Φ 保持重心组合系数不变，即

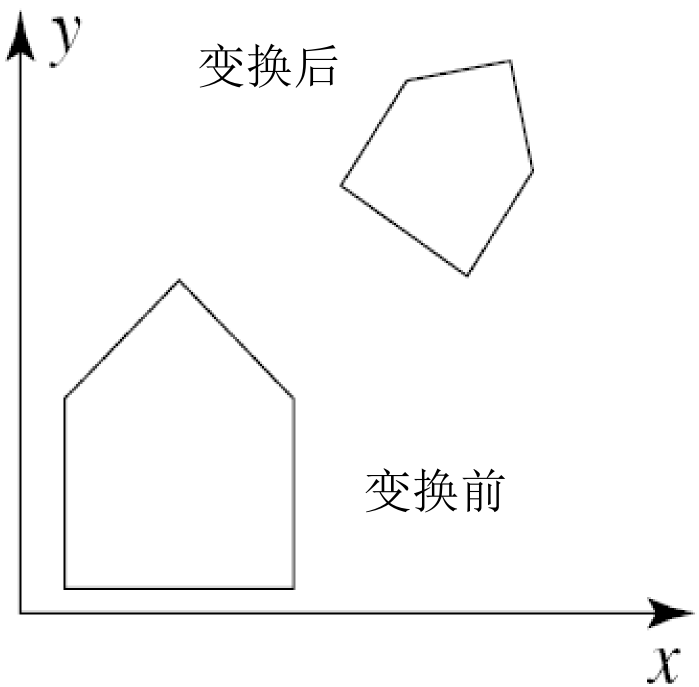
$$\Phi(\mathbf{x}) = \Phi(\sum \alpha_j \mathbf{a}_j) = \sum \alpha_j \Phi(\mathbf{a}_j)$$

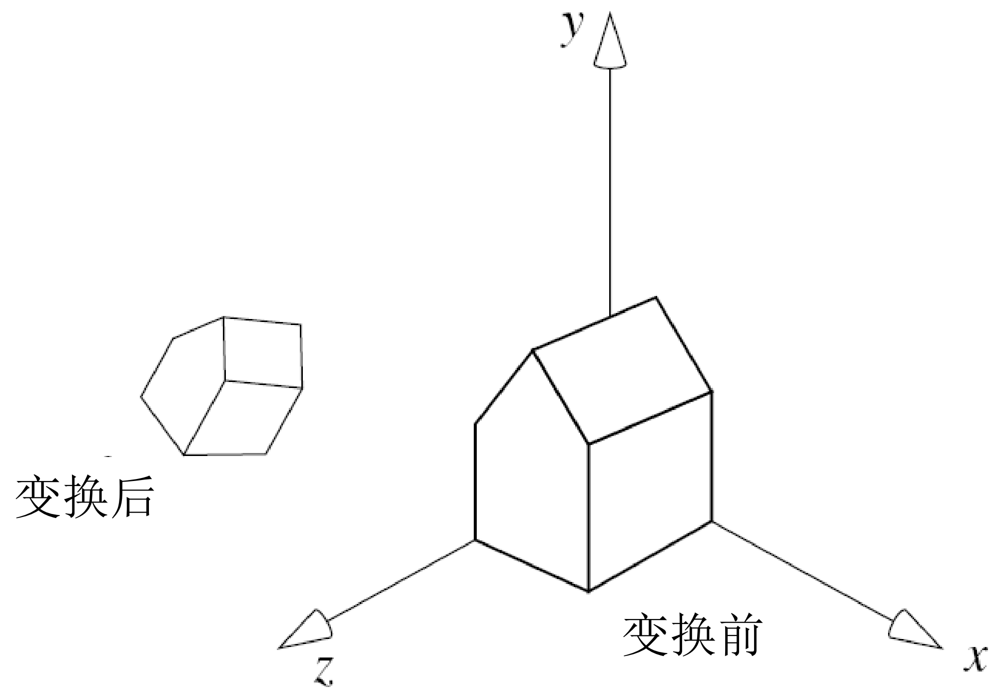
其中 $\mathbf{x}, \mathbf{a}_j \in \mathbb{R}^3$, $\sum \alpha_j = 1$

- 可以证明：任一仿射变换可表示为

$$\Phi(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$$

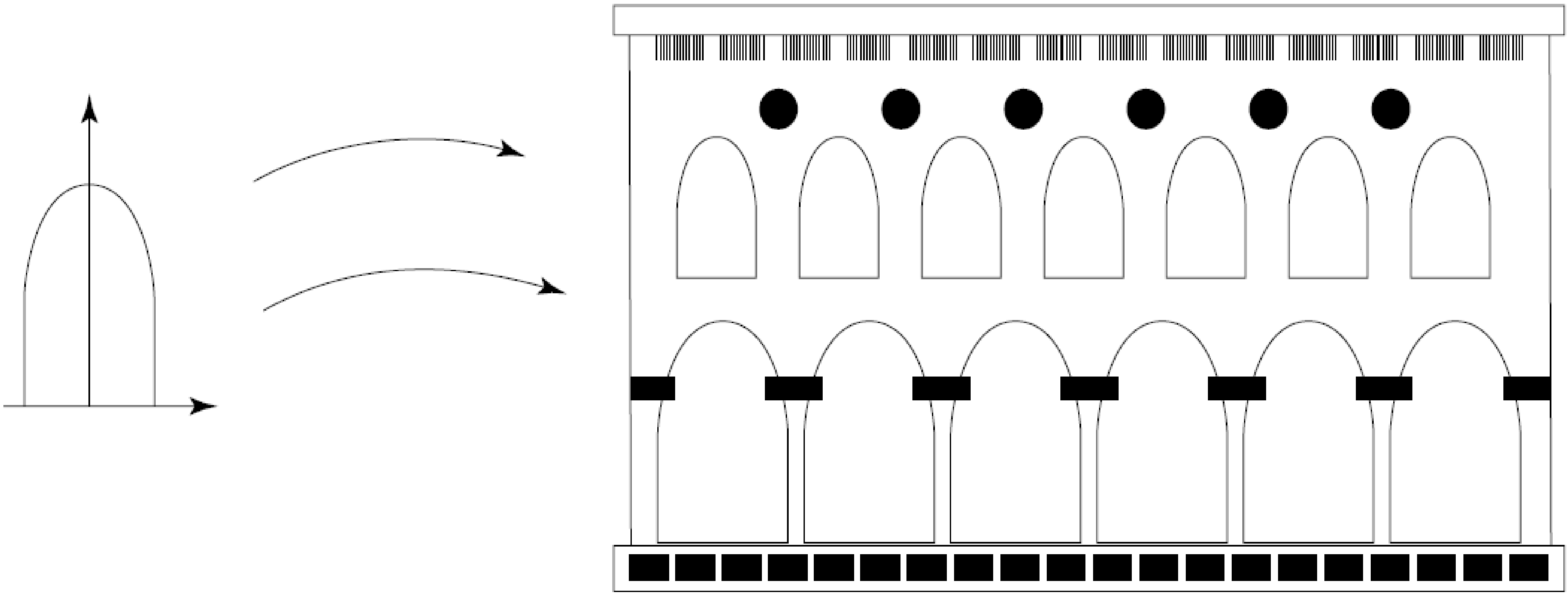
为什么需要变换?



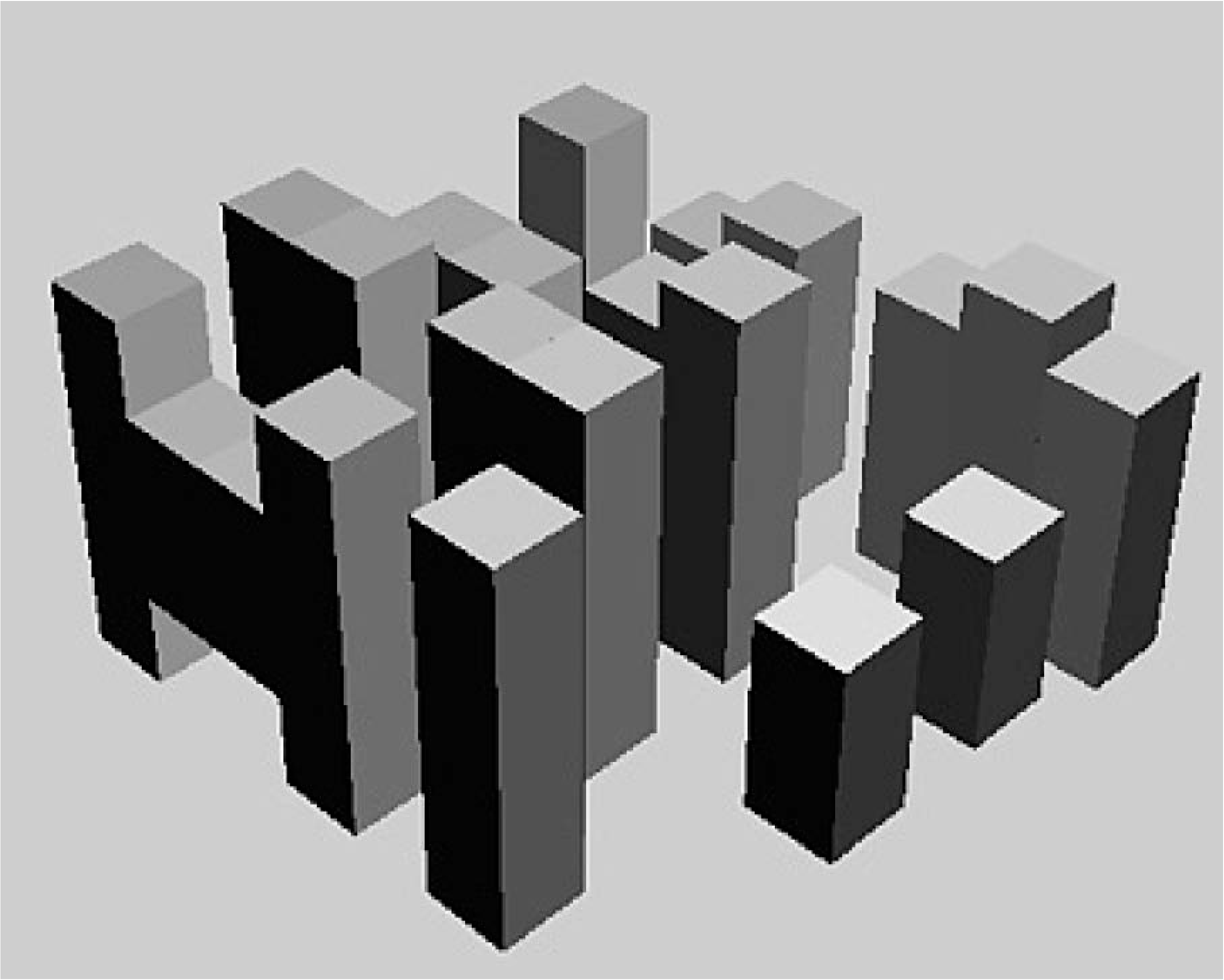


变换的作用之一

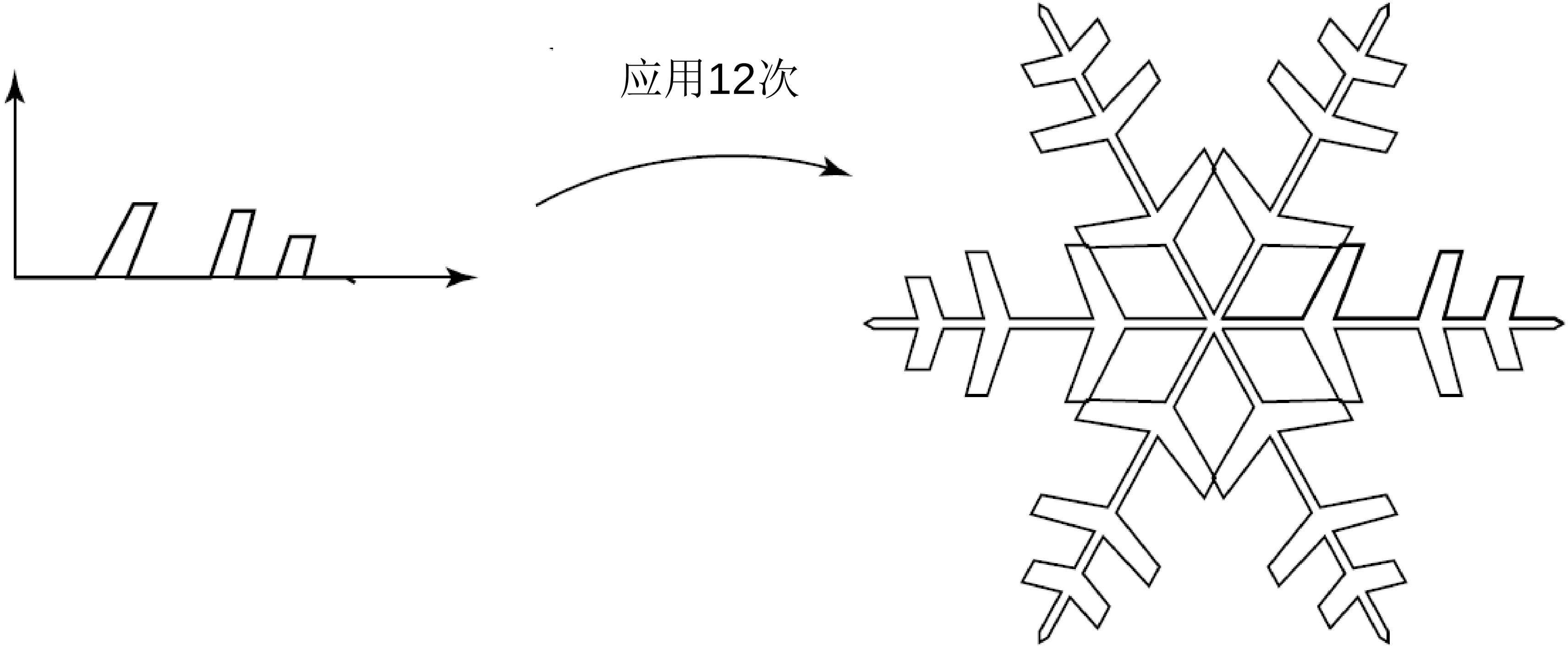
- 组合场景



构造三维场景

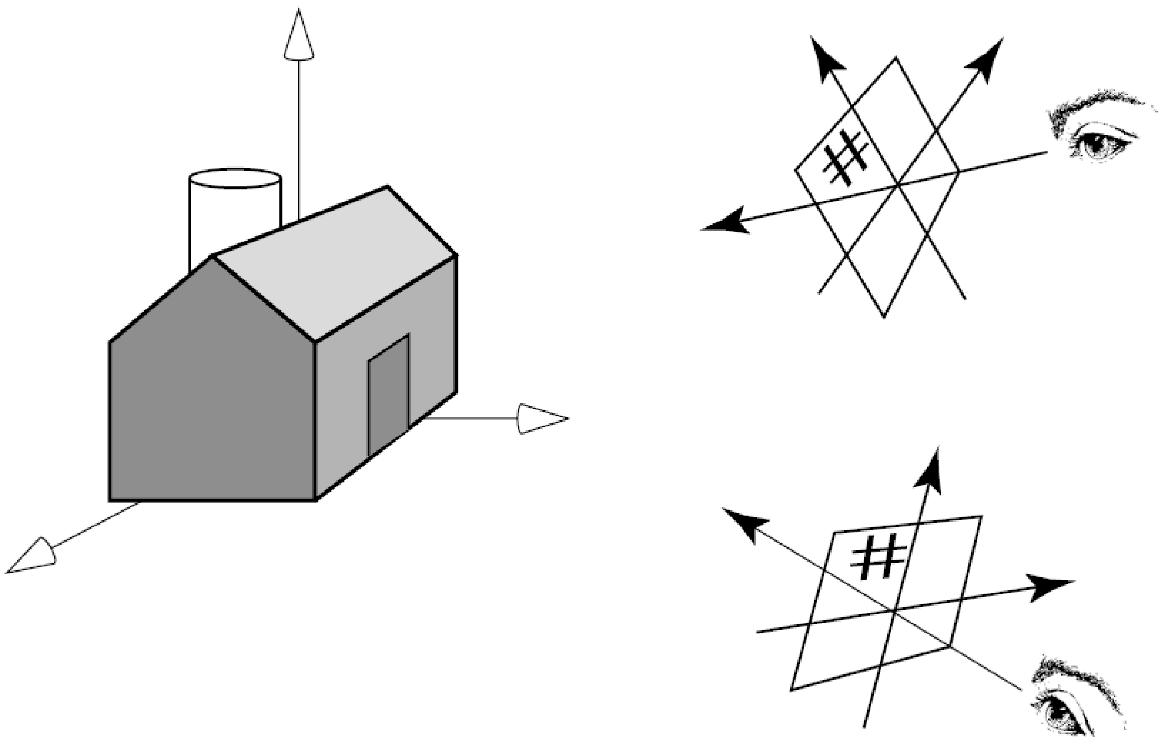


雪花的构造



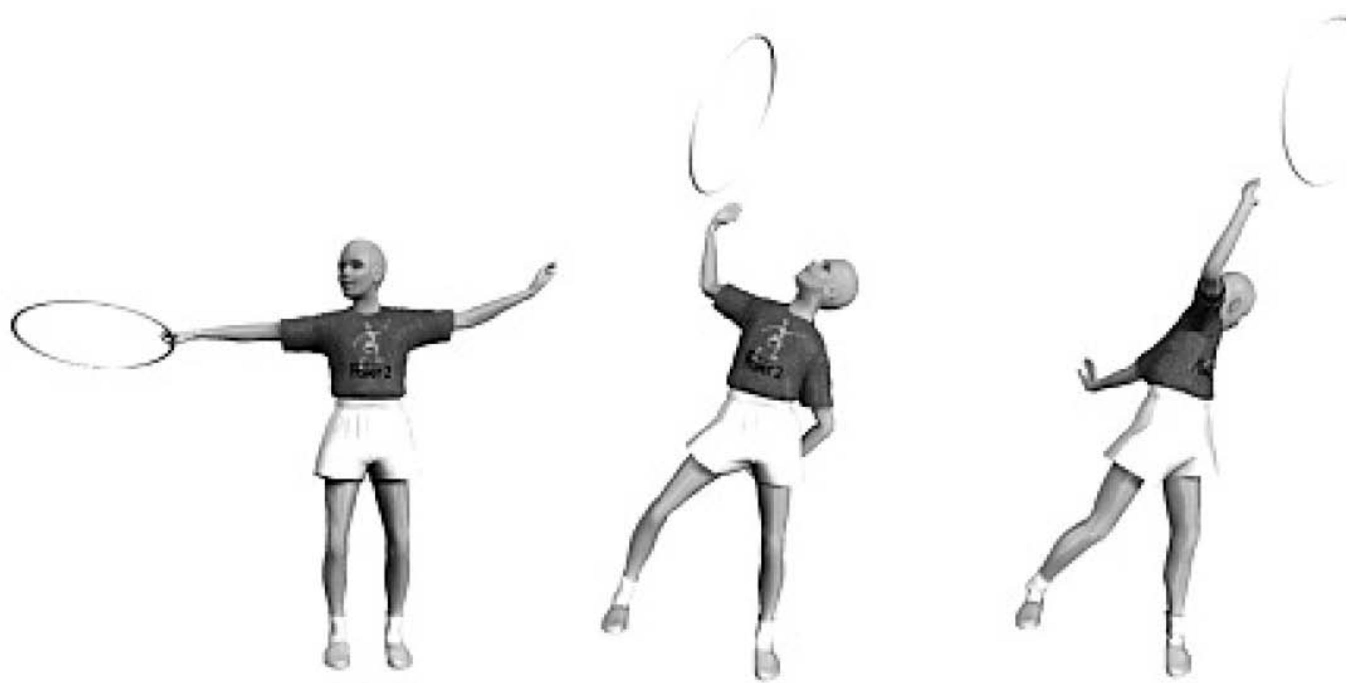
变换的作用之二

- 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

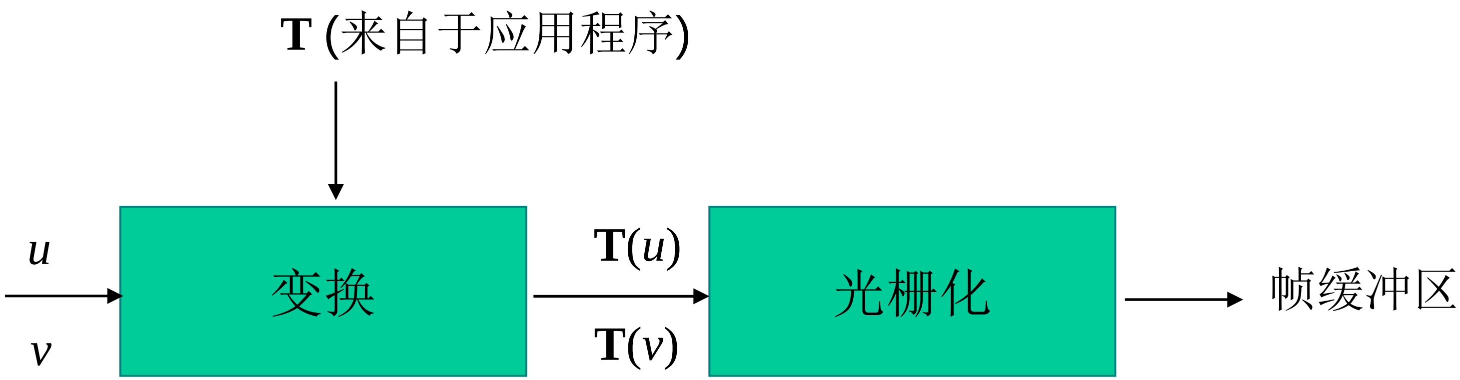


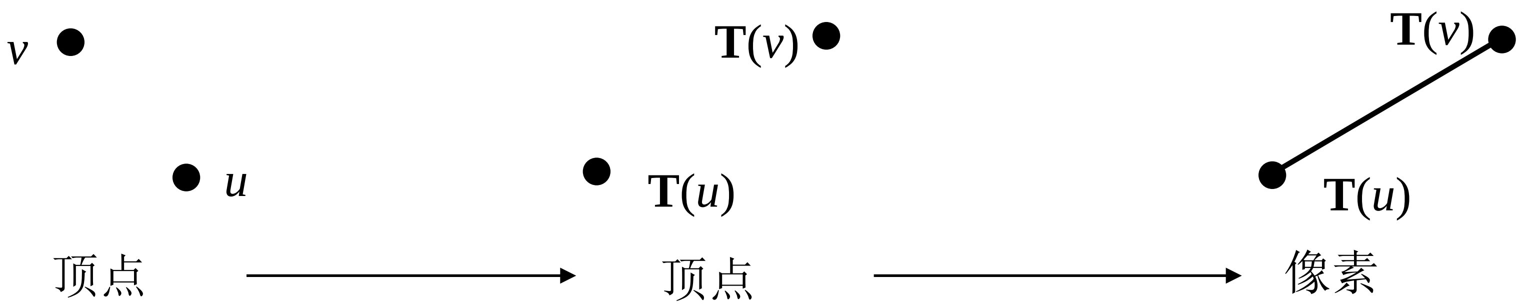
变换的作用之三

- 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



流水线实现





记号

- 在后面的处理中，既会考虑变换的无坐标表示，也会考虑变换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

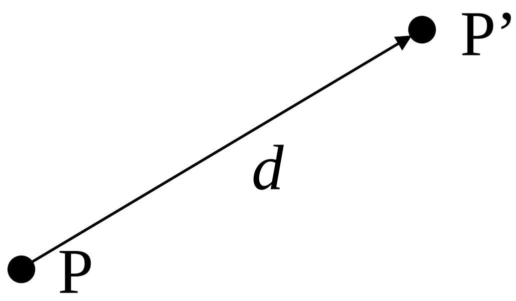
  在齐次坐标中为由四个标量构成的数组

  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

平移

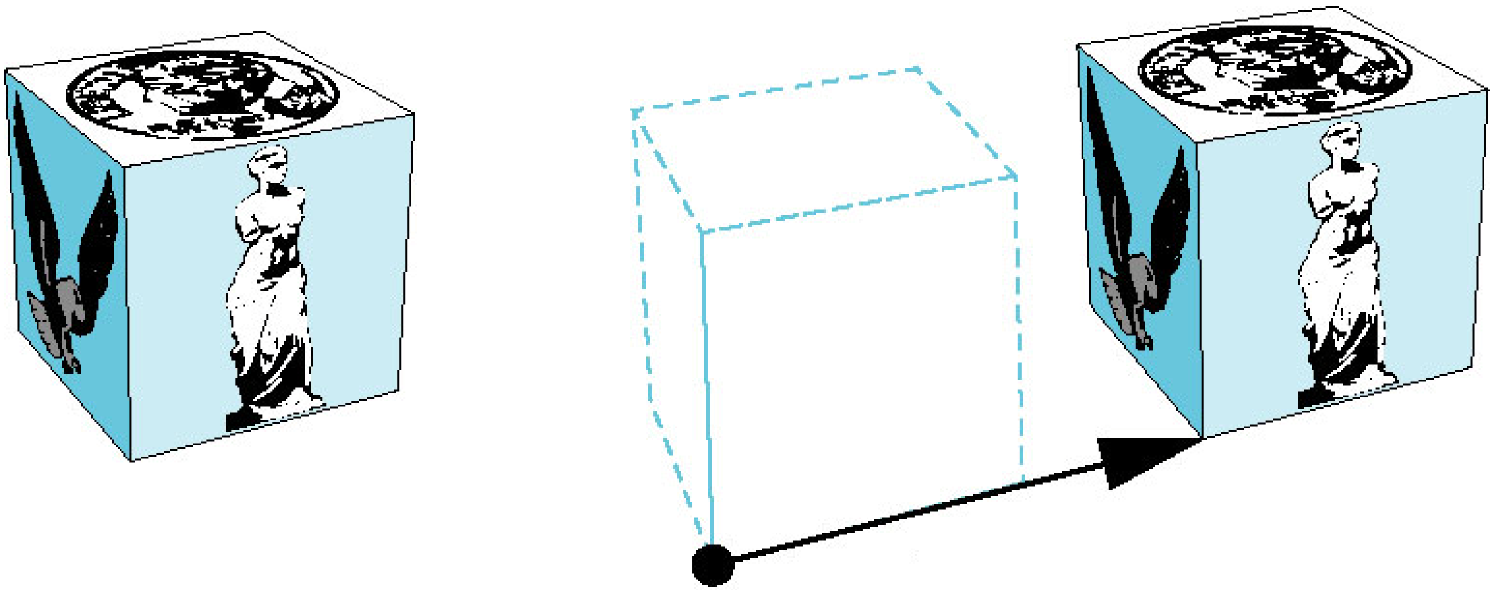
- 把一个点移到新的位置



- 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向量平移



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

$$p = [x, y, z, 1]^T$$

$$p' = [x', y', z', 1]^T$$

$$d = [d_x, d_y, d_z, 0]^T$$

那么 $p' = p + d$ 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
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：p’ = Tp, 其中

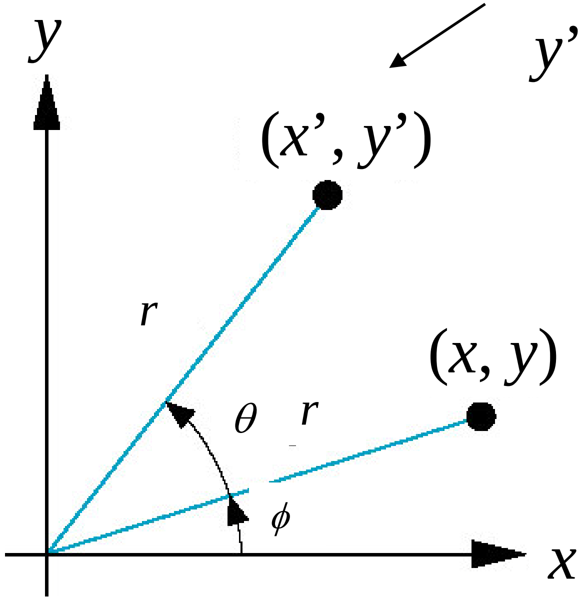
$$\mathbf{T} = \mathbf{T}(d_x, d_y, d_z) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

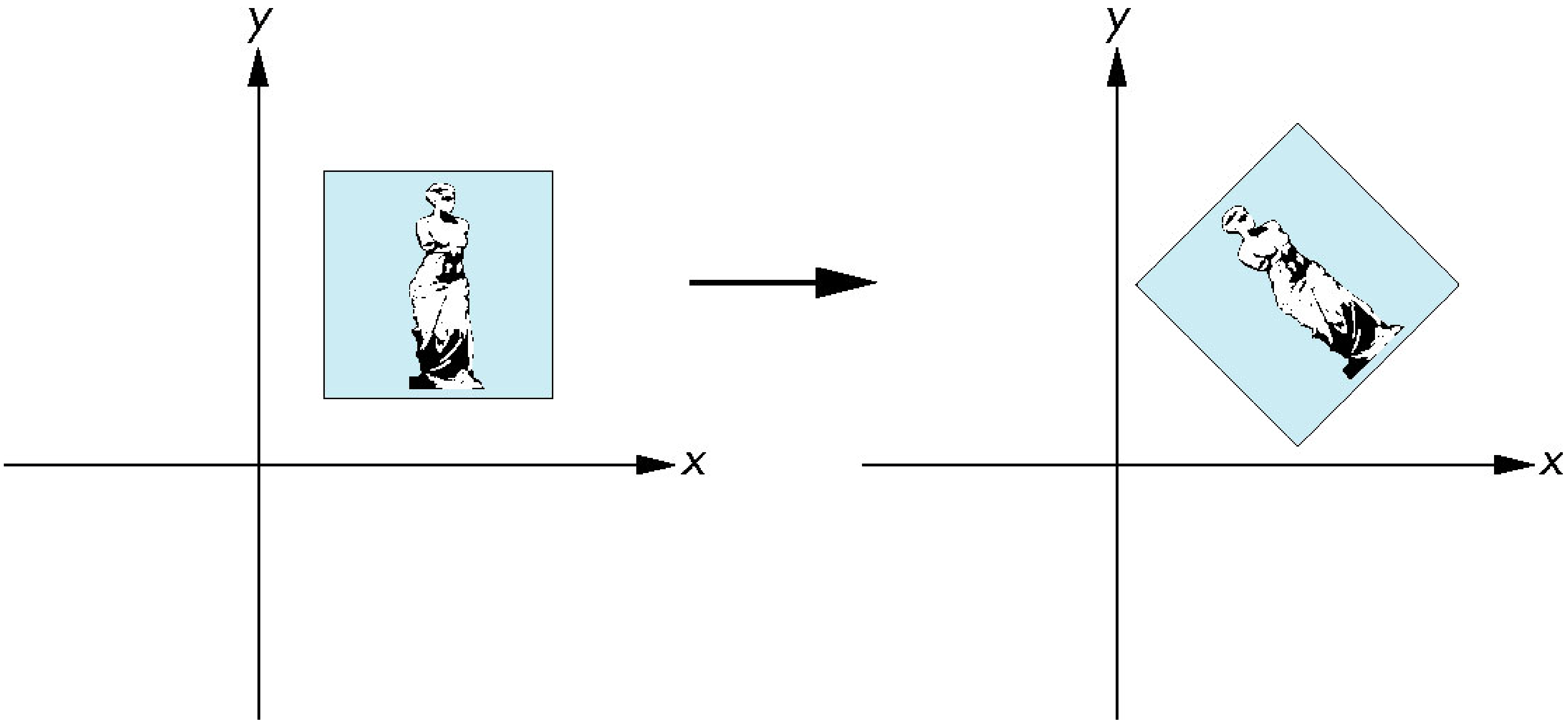
$$x' = r\cos(\phi + \theta)$$
$$y' = r\sin(\phi + \theta)$$



$$\boxed{\begin{aligned} x' &= x\cos\theta - y\sin\theta \\ y' &= x\sin\theta + y\cos\theta \end{aligned}}$$

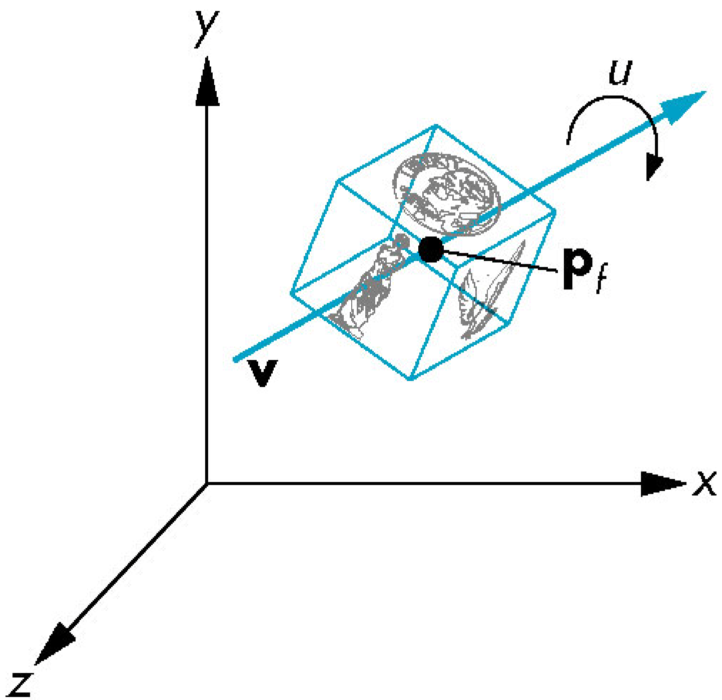
$$x = r\cos\phi$$
$$y = r\sin\phi$$

二维旋转



三维旋转

- 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

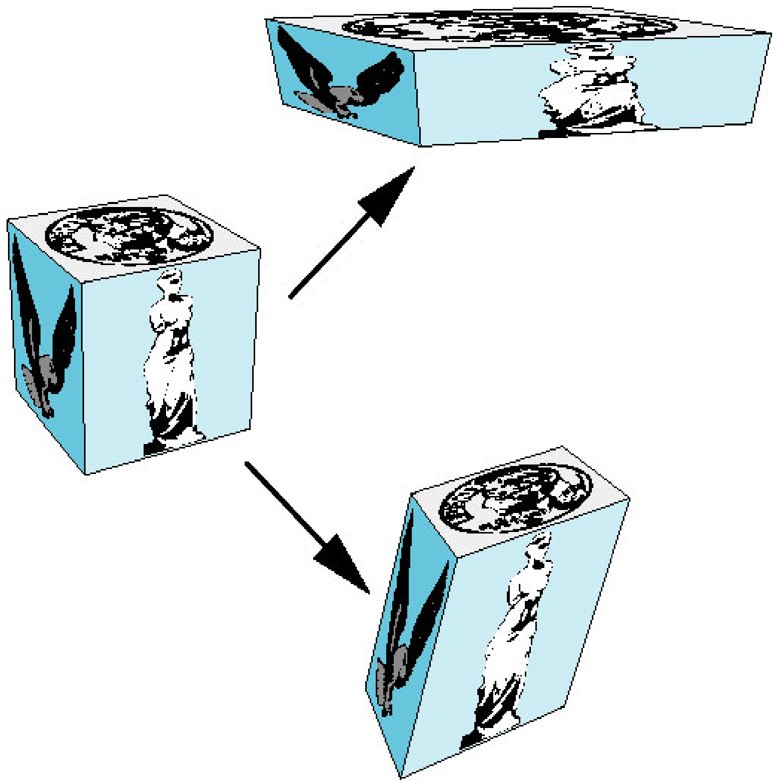
- 与绕z轴的旋转完全类似
  - 对于绕x轴的旋转，x坐标不变
  - 对于绕y轴的旋转，y坐标不变

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

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

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
- 其它的仿射变换会改变对象的形状



放缩

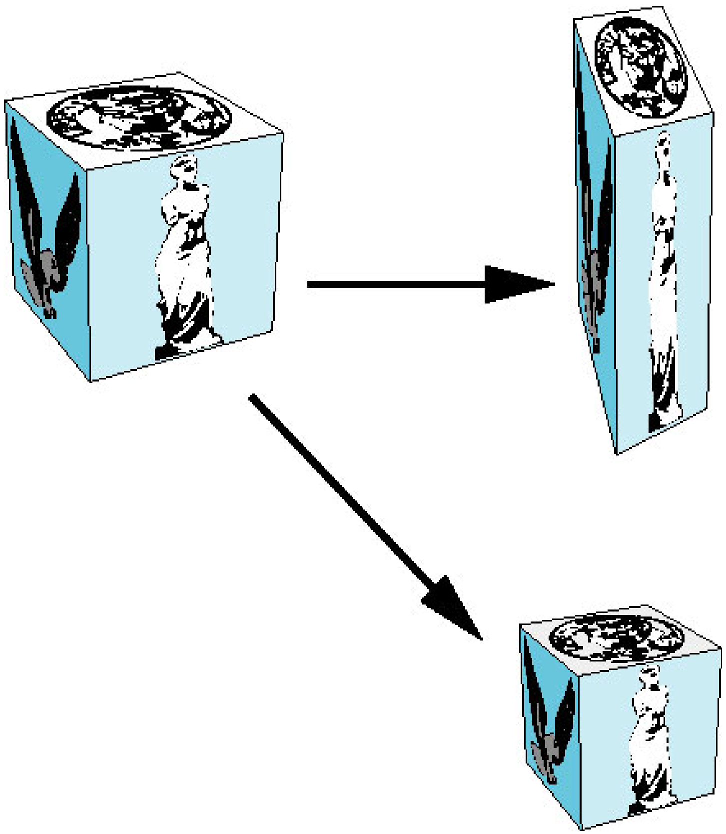
- 沿每个坐标轴伸展或收缩（原点为不动点）

$$x' = s_x x$$

$$y' = s_y y$$

$$z' = s_z z$$

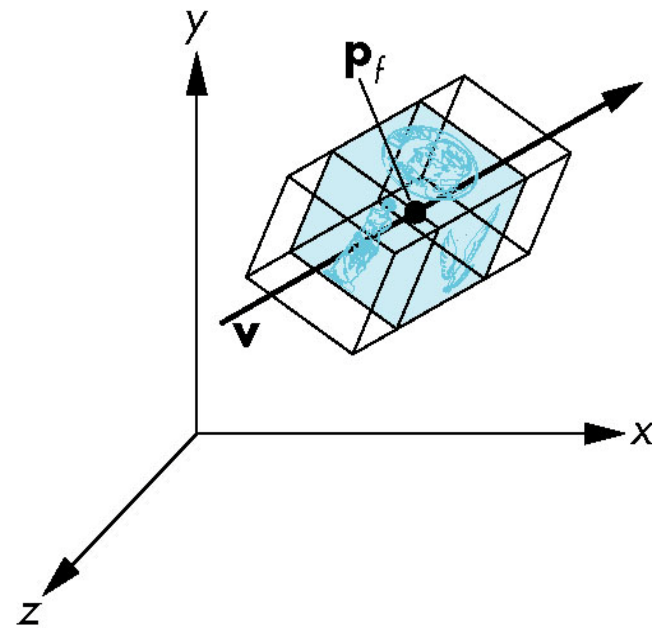
$$p' = Sp$$



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

放缩因子

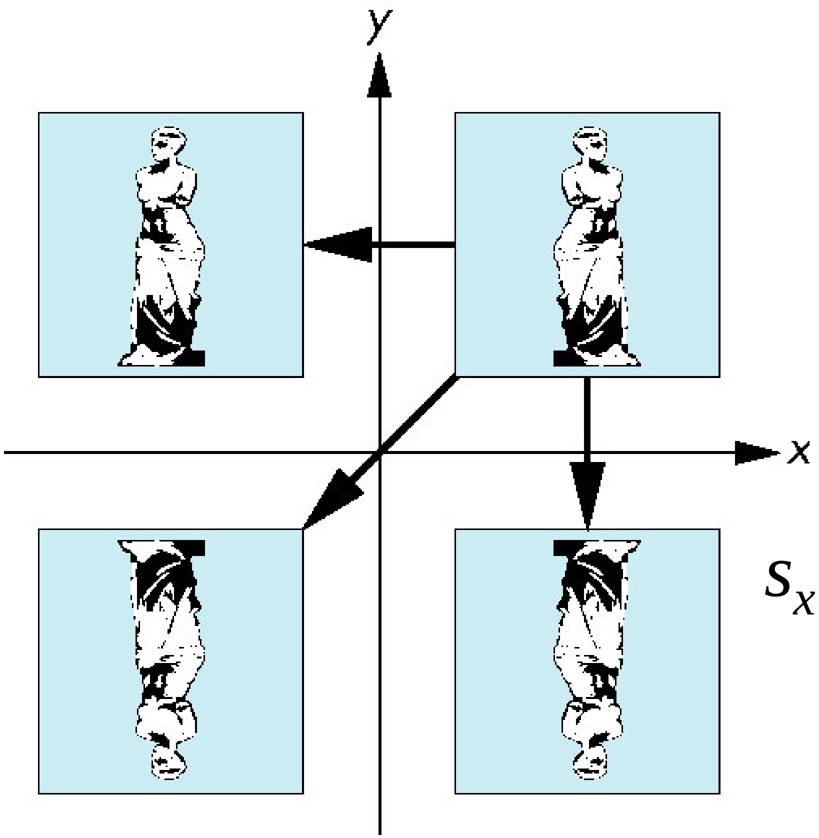
- 放缩变换必定有一个不动点
- 为了定义放缩变换，可以指定其不动点，一个放缩方向，以及沿该方向的放缩因子
- 当放缩因子大于1时，对象在指定方向上变长



反射

■ 对应于负的放缩因子

$s_x = -1,\ s_y = 1$



初始形状

$s_x = -1,\ s_y = -1$

$s_x = 1,\ s_y = -1$

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：$\mathbf{T}^{-1}(d_x,d_y,d_z) = \mathrm{T}(-d_x, -d_y, -d_z)$
  - 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

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

  - 放缩：$\mathbf{S}^{-1}(s_x,s_y,s_z) = \mathbf{S}(1/s_x, 1/s_y, 1/s_z)$

变换的复合

- 可以通过把旋转、平移与放缩矩阵相乘从而形成任意的仿射变换
- 因为对许多顶点应用同样的变换，所以构造矩阵M = ABCD的代价相比于对许多顶点p计算Mp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

- 从数学的角度来说，下述表示是等价的

$$p' = ABCp = A(B(Cp))$$

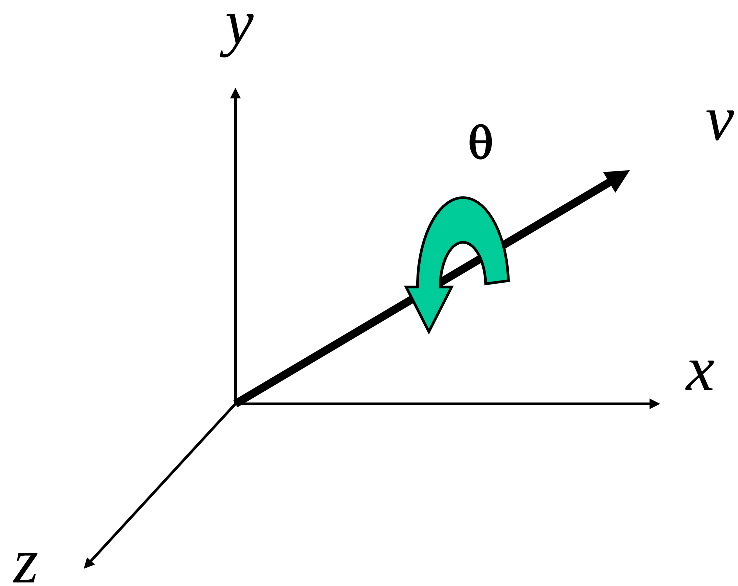
- 变换的顺序是不可交换的
- 注意
  - 在右边的矩阵是首先被应用的矩阵
  - 对于函数调用来说，离顶点数据越近的变换越先作用，譬如

```
glTranslate3f(…)  //A
glScale3f(…)  //B
glRotate3f(…) //C
glVertex3f(…) //p
```

  - OpenGL中的矩阵乘法是右乘：C ← C M

绕原点的一般旋转

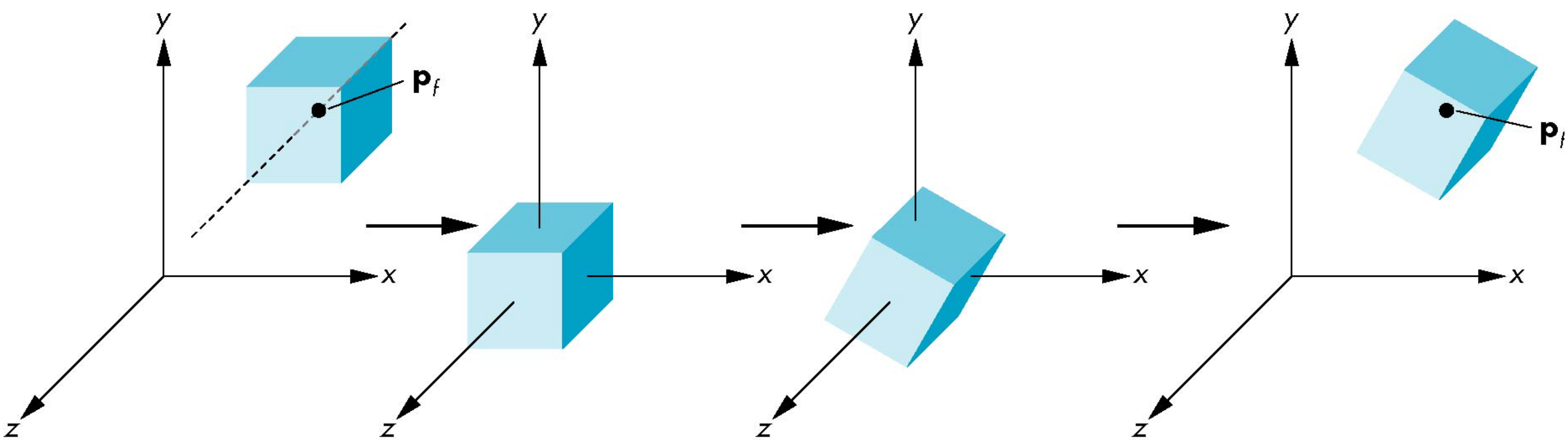
- 绕过原点任一轴旋转θ角可以分解为绕x, y, z轴旋转的复合：$R(\theta) = R_z(\theta_z)R_y(\theta_y)R_x(\theta_x)$
  $\theta_x, \theta_y, \theta_z$称为Euler角。
- 注意：旋转顺序不可交换；可以用不同的旋转顺序，不同的旋转角度得到同样的效果



绕不同于原点的固定点旋转

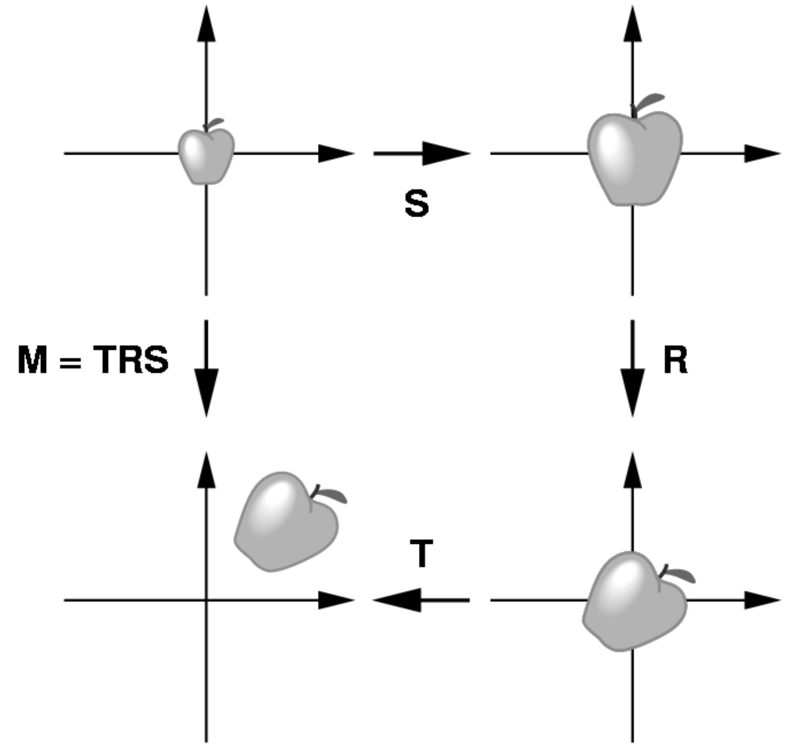
- 把固定点移到原点
- 旋转
- 把固定点移回到原来位置

$$M = T(p_f)R(\theta)T(-p_f)$$



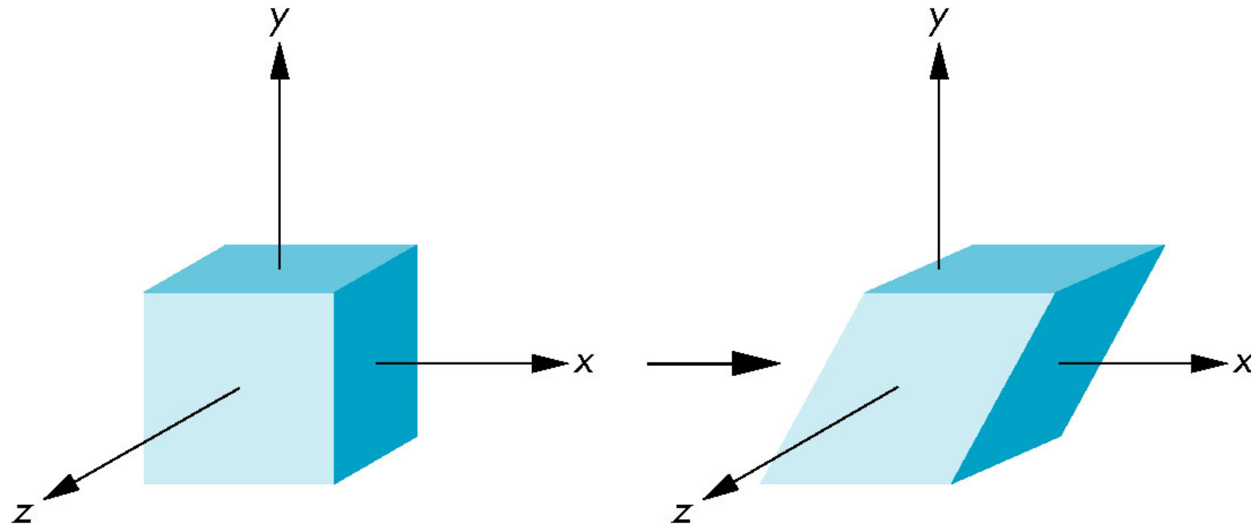
实例

- 在进行造型时，通常从一个中心在原点，相对于坐标轴定向的标准尺寸的对象开始
- 接着对这个对象的顶点应用一个实例变换



剪切变换

- 剪切变换（shear transformation）：一种实用的基本变换
- 等价于把面向相反方向倾斜



剪切矩阵

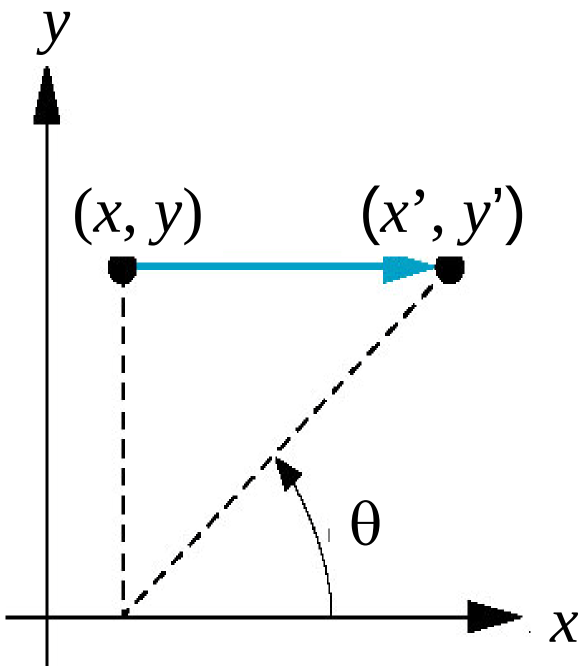
- 考虑沿x轴的剪切

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

$$y' = y$$

$$z' = z$$

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



进一步的性质

- 一个二维旋转相当于三个剪切变换的复合

$$\begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix} = \begin{pmatrix} 1 & \tan\frac{\alpha}{2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -\sin\alpha & 1 \end{pmatrix} \begin{pmatrix} 1 & \tan\frac{\alpha}{2} \\ 0 & 1 \end{pmatrix}$$

- 但剪切变换可以表示为旋转、放缩、平移的复合
  - 如何表示？
- 三维空间中上述结论如何？
- 任一仿射变换一定可以分解成一系列平移、旋转和缩放变换的复合！

Thanks for your attention!