虚拟世界的几何模型
The Geometry of Virtual Worlds

2019.11.11

Outline

1. 几何建模
2. 刚体的变换
3. 联接多个变换

几何建模

右手坐标系

- 让 W 为一个 3D 世界，其中 $W \subseteq R^3$
- 空间中每个点可用坐标 (x, y, z) 表示，其中 $(x, y, z) \in W$

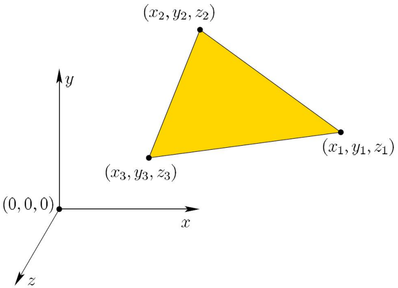


Figure 3.1: Points in the virtual world are given coordinates in a right-handed coordinate system in which the y axis is pointing upward. The origin $(0, 0, 0)$ lies at the point where axes intersect. Also shown is a 3D triangle is defined by its three vertices, each of which is a point in $\mathbb{R}^3$.

图片来自 [3]

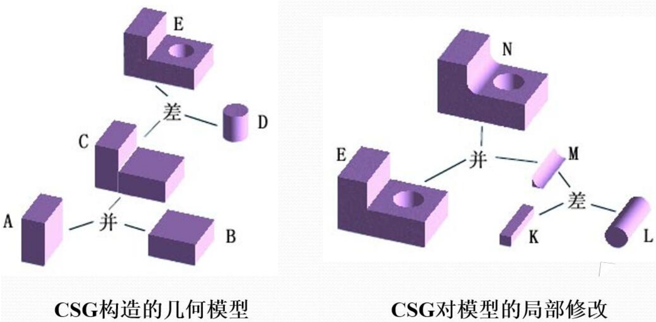
两种模型

在虚拟世界中有两类模型

- 固定模型：
  - 用世界坐标系来描述
- 可移动模型：
  - 有多种可能的变换
  - 通常由刚体组成，每个刚体用自身坐标系

建模方法

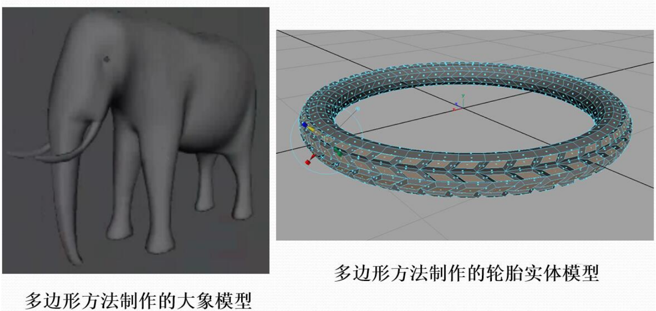
构造立体几何 (Constructive Solid Geometry, CSG)



图片来自 [1]

建模方法

多边形（Polygon）表示法



多边形方法制作的大象模型

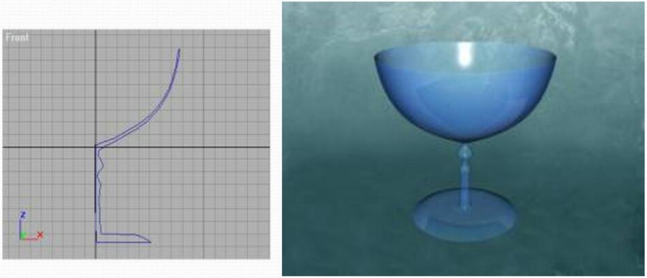
多边形方法制作的轮胎实体模型

图片来自 [1]

建模方法

其他建模方法

- 非统一有理 B 样条（Non-uniform Rational B-Splines NURBS）
- 半代数曲面（semi-algebraic surfaces)



(a) 酒杯截面造型　　(b) 酒杯造型

酒杯截面造型过程

图片来自 [1]

建模方法

建模方法

- 实体表示法 solid representation：3D 基本单元
- 边界表示法 boundary representation：2D 基本单元

考虑障碍物和碰撞测试

我们将讨论使用三角形作为基本单元。

三角形网格组成的几何模型

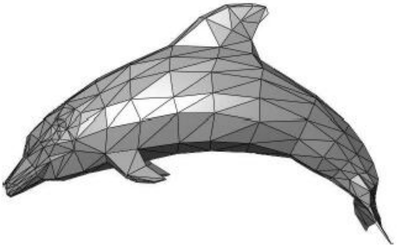


Figure 3.2: A geometric model of a dolphin, formed from a mesh of 3D triangles (from Wikipedia user Chrschn).

- 如何确定 VR 用户查看每个三角形时，它们是什么样子？
- 如何让对象“移动”？

图片来自 [3]

Outline

为什么要进行变换？

变换是一个将空间中的点 x 映射成其他点 x' 的函数。
为什么要进行变换？

- 运动的模型 movable models
- 对静止的感知 perception of stationary:
  - 需要对固定在人身上的刺激进行反向旋转
  - 反向变换需要进行跟踪 (tracking)
- 广泛应用于：morphing,deformation,viewing, projection, real-time shadows,...

刚体的变换

刚体的变换：对三角形的每个端点进行变换

在理论力学中，物体的自由度（Degree of Freedom, DOF）是确定物体的位置所需要的独立坐标数。
3 种情况:

- 简单：平移 2D-2DOFs；3D-3DOFs
- 更难：旋转 2D-1DOFs；3D-3DOFs
- 最难：平移 + 旋转 2D-3DOFs；3D-6DOFs

平移

对三角形平移 (x_t, y_t, z_t)

$(x_i, y_i, z_i) \rightarrow (x_i + x_t, y_i + y_t, z_i + y_t)$，其中 $i = 1, 2, 3.$

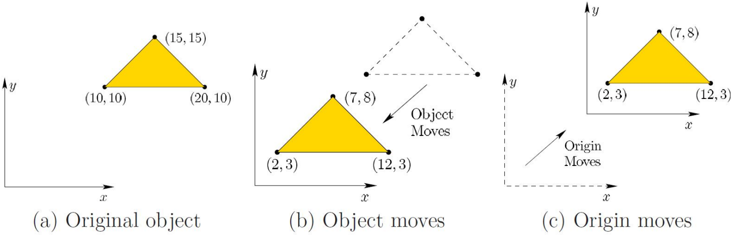


Figure 3.4: Every transformation has two possible interpretations, even though the math is the same. Here is a 2D example, in which a triangle is defined in (a). We could translate the triangle by $x_t = -8$ and $y_t = -7$ to obtain the result in (b). If we instead wanted to hold the triangle fixed but move the origin up by 8 in the x direction and 7 in the y direction, then the coordinates of the triangle vertices change the exact same way, as shown in (c).

图片来自 [3]

2D 线性变换

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

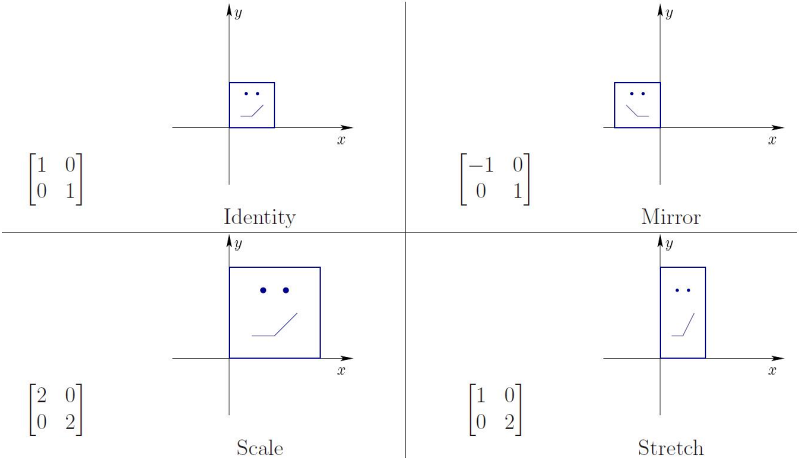
$(x, y) \to (x', y')$

对笛卡尔坐标系的基坐标进行变换：

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} m_{11} \\ m_{21} \end{bmatrix}$$

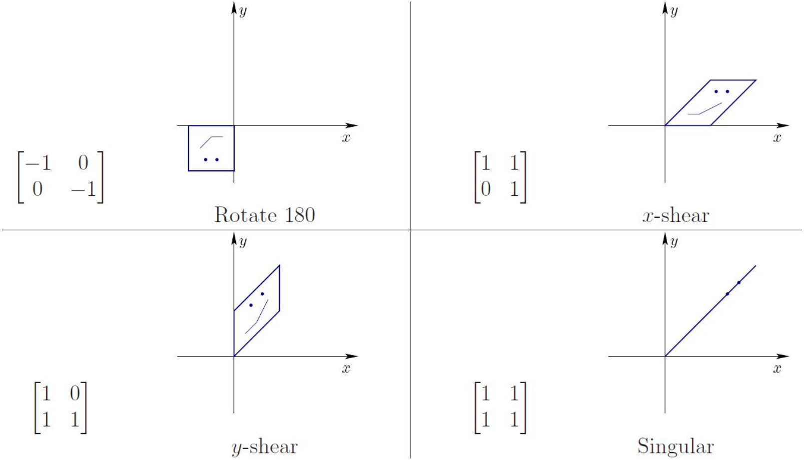
$$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} m_{12} \\ m_{22} \end{bmatrix}$$

2D 变换示例



图片来自 [3]

2D 变换示例



图片来自 [3]

2D 旋转

2D 旋转规则：

- 坐标轴没有拉伸 no strenching of axes：${m_{11}}^2 + {m_{21}}^2 = 1$ 和 ${m_{12}}^2 + {m_{22}}^2 = 1$
- 无错切 no shearing：$m_{11}m_{12} + m_{21}m_{22} = 0$
- 无镜像 no mirror images: $det(\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}) = 1$ 而不是-1

最终的矩阵为：$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

反旋转：$R^T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$

3D 旋转

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \text{9DOFs}$$

让 $v_1 = \begin{bmatrix} m_{11} \\ m_{21} \\ m_{31} \end{bmatrix}, v_2 = \begin{bmatrix} m_{12} \\ m_{22} \\ m_{32} \end{bmatrix}, v_3 = \begin{bmatrix} m_{13} \\ m_{23} \\ m_{33} \end{bmatrix}$

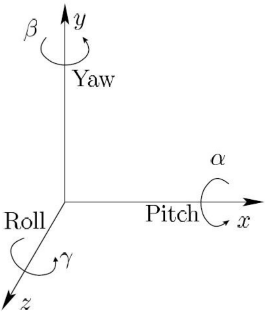
3D 旋转规则:

- $\|v_1\| = \|v_2\| = \|v_3\| = 1$, 减少 3DOFs
- $v_1 \cdot v_2 = v_2 \cdot v_3 = v_1 \cdot v_3 = 0$, 减少 3DOFs
- $det(\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}) = 1$ 而不是-1

剩余 3DOFs

3 种典型旋转

3 种典型旋转：pitch 俯仰；yaw 偏航；roll 滚动



图片来自 [1]

3 种典型旋转

Pitch 俯仰 $R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$

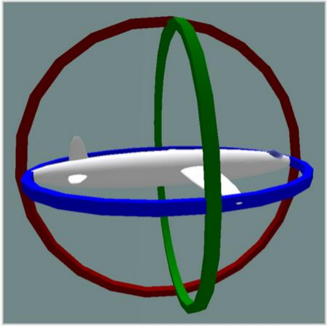
Yaw 偏航 $R_y = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$

Roll 滚动 $R_z = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$

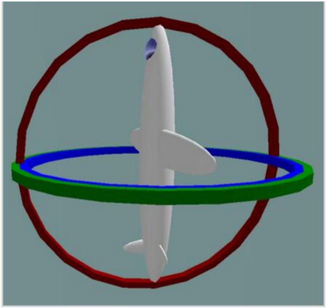
$R = R(\alpha)R(\beta)R(\gamma)$，其中 $\alpha \in [-\pi/2, \pi/2]$, $\beta \in [-\pi, \pi]$, $\theta \in [-\pi, \pi]$

- 顺序很重要,3D 旋转不满足交换律
- 运动奇异性 (kinematic singularities) 和不均匀表示（non-uniform representation）

万向节死锁 Gimbal Lock



Normal situation: the three gimbals are independent



Gimbal lock: two out of the three gimbals are in the same plane, one degree of freedom is lost

欧拉旋转定理

所有 3D 旋转都有一种坐标轴-角度的表示方法（axis-angle representaion）:$(\mathbf{v}, \theta)$，其中 $\mathbf{v} = (v_1, v_2, v_3), \|\mathbf{v}\| = 1$

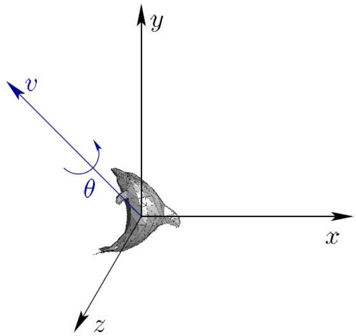


Figure 3.9: Euler's rotation theorem states that every 3D rotation can be considered as a rotation by an angle θ about an axis through the origin, given by the unit direction vector $v = (v_1, v_2, v_3)$.

图片来自 [3]

四元数 quaternion

单元四元数 unit quaternion 更好的表示了 3D 旋转

- $q = (a, b, c, d) \in R^4$ ， $a^2 + b^2 + c^2 + d^2 = 1$

所有单位四元数的集合是一个超球面 (S^3)

坐标轴-角度表示法与四元数：
$(\mathbf{v}, \theta) \leftrightarrow (\cos\theta/2, v_1 \sin\theta/2, v_2 \sin\theta/2, v_3 \sin\theta/2)$

从四元数 (a, b, c, d) 恢复 $(\mathbf{v}, \theta)$:
$\theta = 2arccos\theta, \mathbf{v} = \frac{1}{\sqrt{1-a^2}}(b, c, d)$

四元数 quaternion 示例

Quaternion	Axis-Angle	Description
$(1,0,0,0)$	$(\text{undefined},0)$	Identity rotation
$(0,1,0,0)$	$((1,0,0),\pi)$	Pitch by π
$(0,0,1,0)$	$((0,1,0),\pi)$	Yaw by π
$(0,0,0,1)$	$((0,0,1),\pi)$	Roll by π
$(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}},0,0)$	$((1,0,0),\pi/2)$	Pitch by $\pi/2$
$(\frac{1}{\sqrt{2}},0,\frac{1}{\sqrt{2}},0)$	$((0,1,0),\pi/2)$	Yaw by $\pi/2$
$(\frac{1}{\sqrt{2}},0,0,\frac{1}{\sqrt{2}})$	$((0,0,1),\pi/2)$	Roll by $\pi/2$

Figure 3.11: For these cases, you should be able to look at the quaternion and quickly picture the axis and angle of the corresponding 3D rotation.

图片来自 [3]

反变换和多种表示

$$
\begin{tikzcd}
(a, b, c, d) \arrow[r, leftrightarrow, "\text{equivalent}"'] \arrow[d, leftrightarrow, "\text{inverses}"'] & (-a, -b, -c, -d) \arrow[d, leftrightarrow, "\text{inverses}"] \\
(a, -b, -c, -d) \arrow[r, leftrightarrow, "\text{equivalent}"'] & (-a, b, c, d)
\end{tikzcd}
$$

Figure 3.12: Simple relationships between equivalent quaternions and their inverses.

图片来自 [3]

四元数运算

如何将四元数 $h=(a,b,c,d)$ 表示的旋转作用于物体?
可将四元数转换成旋转矩阵或直接进行四元数运算

$$R(h)=\begin{bmatrix} 2(a^2+b^2)-1 & 2(bc-ad) & 2(bd+ac) \\ 2(bc+ad) & 2(a^2+c^2)-1 & 2(cd-ab) \\ 2(bd-ac) & 2(cd+ab) & 2(a^2+d^2)-1 \end{bmatrix}$$

四元数运算

两个四元数的乘积 $q_1 * q_2 = q_3$ 可表示为:
$a_3 = a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2$
$b_3 = a_1 b_2 + a_2 b_1 + c_1 d_2 - c_2 d_1$
$c_3 = a_1 c_2 + a_2 c_1 + b_2 d_1 - b_1 d_2$
$d_3 = a_1 d_2 + a_2 d_1 + b_1 c_2 - b_2 c_1.$

对点 (x, y, z) 进行旋转，旋转用 $q = (a, b, c, d)$ 表示
让 $p = (0, x, y, z)$，$p' = q * p * q^{-1} = (0, x', y', z')$

齐次坐标变换矩阵 Homogeneous transformation matrices

齐次坐标的本质是使用四维数组来表示三维空间中的点和向量。

平移：$x \longrightarrow x' = x + x_t$, $y \longrightarrow y' = y + y_t$, $z \longrightarrow z' = z + z_t$

把平移矢量记为 $T = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix}$

平移可用齐次矩阵表示为：

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_t \\ 0 & 1 & 0 & y_t \\ 0 & 0 & 1 & z_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$，变换矩阵可记为 $\begin{bmatrix} I & T \\ 0 & 1 \end{bmatrix}$

反变换矩阵：$\begin{bmatrix} 1 & 0 & 0 & -x_t \\ 0 & 1 & 0 & -y_t \\ 0 & 0 & 1 & -z_t \\ 0 & 0 & 0 & 1 \end{bmatrix}$，可记为 $\begin{bmatrix} I & -T \\ 0 & 1 \end{bmatrix}$

齐次坐标变换矩阵 Homogeneous transformation matrices

旋转： $\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

把 $R_{3\times3}$ 表示旋转矩阵，齐次旋转矩阵可表示为：

$\begin{bmatrix} m_{11} & m_{12} & m_{13} & 0 \\ m_{21} & m_{22} & m_{23} & 0 \\ m_{31} & m_{32} & m_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$，变换矩阵可记为 $\begin{bmatrix} R_{3\times3} & 0 \\ 0 & 1 \end{bmatrix}$

反变换矩阵： $\begin{bmatrix} m_{11} & m_{21} & m_{31} & 0 \\ m_{12} & m_{22} & m_{32} & 0 \\ m_{13} & m_{23} & m_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$，可记为 $\begin{bmatrix} (R_{3\times3})^T & 0 \\ 0 & 1 \end{bmatrix}$

齐次坐标变换矩阵 Homogeneous transformation matrices

先平移再旋转：

$$\begin{bmatrix} R_{3\times3} & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} I & T \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{3\times3} & R_{3\times3}T \\ 0 & 1 \end{bmatrix}$$

反变换：先反旋转再反平移

$$\begin{bmatrix} I & -T \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \left(R_{3\times3}\right)^T & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \left(R_{3\times3}\right)^T & -T \\ 0 & 1 \end{bmatrix}$$

齐次坐标变换矩阵 Homogeneous transformation matrices

先旋转再平移：

$$\begin{bmatrix} I & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{3\times3} & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{3\times3} & T \\ 0 & 1 \end{bmatrix}$$

反变换：先反平移再反旋转

$$\begin{bmatrix} (R_{3\times3})^T & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -T \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} (R_{3\times3})^T & -(R_{3\times3})^T T \\ 0 & 1 \end{bmatrix}$$

齐次坐标变换矩阵 Homogeneous transformation matrices

先旋转再平移：

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix}$$

等效的 4×4 矩阵变换：

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & x_t \\ m_{21} & m_{22} & m_{23} & y_t \\ m_{31} & m_{32} & m_{33} & z_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

反变换是什么？

Outline

1. 几何建模
2. 刚体的变换
3. 联接多个变换

要解决的问题

从虚拟的三维场景及相机的位置信息中，生成出一幅二维图像

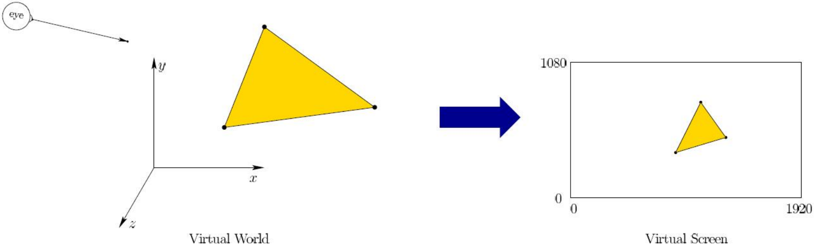


Figure 3.13: If we placed a virtual eye or camera into the virtual world, what would it see? Section 3.4 provides transformations that place objects from the virtual world onto a virtual screen, based on the particular viewpoint of a virtual eye. A flat rectangular shape is chosen for engineering and historical reasons, even though it does not match the shape of our retinas.

图片来自 [3]

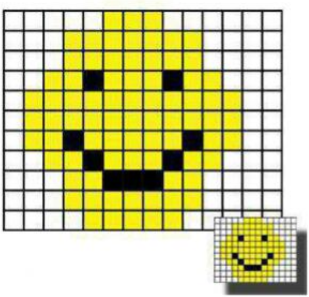
多个变换的联接 the chain of transformations:

$T = T_{vp} T_{can} T_{eye} T_{rb}$

图像和像素

图像

- 图像可以看成是一个二维离散函数:$f(x, y)$
- 函数 f 的定义域是由矩阵排列着的许多格子组成, 这些格子被称为 像素 **(pixel)**
- 图像 f 的取值则为各个像素的色彩: 对于彩色图像, 可以是 RGB 或者是 RGBA; 对于灰度图像, f 为单值函数。



视图变换 Eye transformation

T_{eye}：世界坐标系 (World coordinate frame) $\longrightarrow$ 眼睛坐标系（Eye coordinate frame）

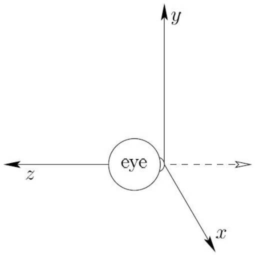


Figure 3.14: Consider an eye that is looking down the z axis in the negative direction. The origin of the model is the point at which light enters the eye.

正交坐标系

3 维空间中的任何 3 个矢量, 满足

- $\|u\| = \|v\| = \|w\| = 1$
- $u \cdot v = v \cdot w = u \cdot w = 0$
- $w = u \times v$ (右手)

除了 XYZ, 还有很多坐标系：global, local, world, model, parts of model(head, eye, hands, ...)

视图变换 Eye transformation

- 眼球的位置： e
- 视线方向： $\hat{c} = \frac{p-e}{\|p-e\|}$
- 向上的方向： $\hat{u}$

眼睛坐标系

- $\hat{x} = \hat{u} \times \hat{z}$
- $\hat{y} = \hat{z} \times \hat{x}$
- $\hat{z} = -\hat{c}$

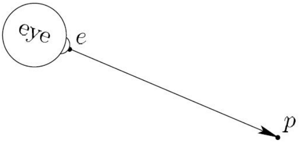


Figure 3.15: The vector from the eye position e to a point p that it is looking at is normalized to form $\hat{c}$ in (3.36).

图片来自 [3]

视图变换 Eye transformation

眼球的旋转：$\begin{bmatrix} \hat{x}_1 & \hat{y}_1 & \hat{z}_1 \\ \hat{x}_2 & \hat{y}_2 & \hat{z}_2 \\ \hat{x}_3 & \hat{y}_3 & \hat{z}_3 \end{bmatrix}$ 平移：$\hat{e}$

视图变换矩阵：$T_{eye} = \begin{bmatrix} \hat{x}_1 & \hat{x}_2 & \hat{x}_3 & 0 \\ \hat{y}_1 & \hat{y}_2 & \hat{y}_3 & 0 \\ \hat{z}_1 & \hat{z}_2 & \hat{z}_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -e_1 \\ 0 & 1 & 0 & -e_2 \\ 0 & 0 & 1 & -e_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

视图变换 Eye transformation

瞳孔间距 Interpupillary distance (IPD)-t 均值 0.064m

左眼视图变换：$\begin{bmatrix} 1 & 0 & 0 & t/2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T_{eye}$

右眼视图变换：$\begin{bmatrix} 1 & 0 & 0 & -t/2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T_{eye}$

透视投影 Perspective projection

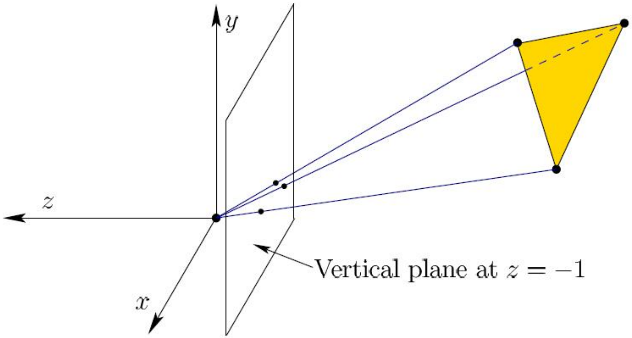


Figure 3.17: An illustration of perspective projection. The model vertices are projected onto a virtual screen by drawing lines through them and the origin $(0, 0, 0)$. The "image" of the points on the virtual screen corresponds to the intersections of the line with the screen.

图片来自 [3]

透视投影 Perspective projection

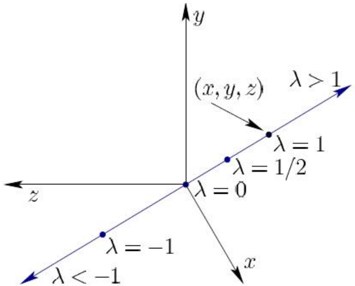


Figure 3.16: Starting with any point (x, y, z), a line through the origin can be formed using a parameter λ. It is the set of all points of the form $(\lambda x, \lambda y, \lambda z)$ for any real value λ. For example, $\lambda = 1/2$ corresponds to the midpoint between (x, y, z) and $(0, 0, 0)$ along the line.

图片来自 [3]

标准视图变换 Canonical viewing transformation

T_{can}：为了把 3D 物体映射到 2D 图像上，眼睛坐标系 (Eye coordinate frame) $\longrightarrow$ 屏幕坐标系（Screen coordinate frame） within the range of [-1, 1]

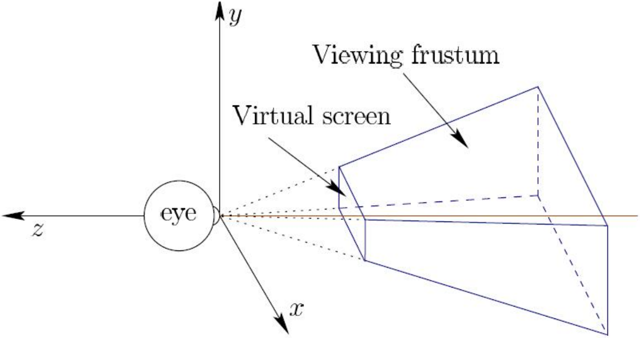


Figure 3.18: The viewing frustrum.

图片来自 [3]

标准视图变换 Canonical viewing transformation

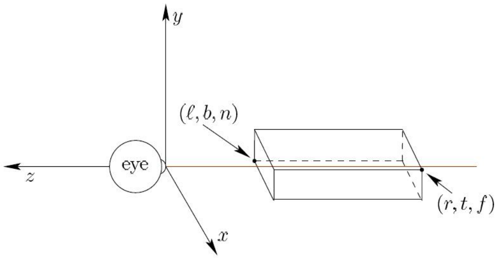
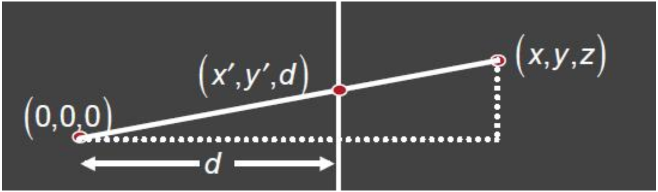


Figure 3.19: The rectangular region formed by the corners of the viewing frustum, after they are transformed by T_p. The coordinates of the selected opposite corners provide the six parameters, ℓ, r, b, t, n, and f, which used in T_{st}.

图片来自 [3]

标准视图变换 Canonical viewing transformation

$$T_p = \begin{bmatrix} n & 0 & 0 & -t/2 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

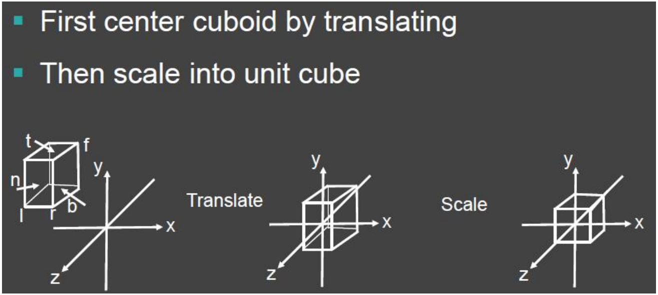


图片来自 Ravi Ramamoorthi Edx course

标准视图变换 Canonical viewing transformation

$$T_{st} = \begin{bmatrix} 2/(r-l) & 0 & 0 & -(r+l)/(r-l) \\ 0 & 2/(t-b) & 0 & -(t+b)/(t-b) \\ 0 & 0 & 2/(n-f) & -(n+f)/(n-f) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{can} = T_{st} T_p$$



图片来自 Ravi Ramamoorthi Edx course

视口变换 Viewport transformation

T_{vp}：屏幕坐标系 (Screen coordinate frame) within [-1, 1]$\longrightarrow$ 像素坐标系（pixel coordinate frame）

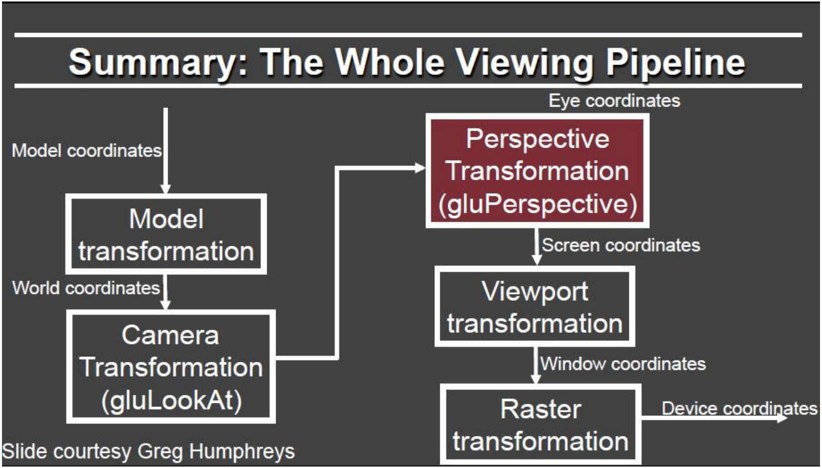
m: 每行的像素个数；n：每列的像素个数

$$T_{vp} = \begin{bmatrix} m/2 & 0 & 0 & (m-1)/2 \\ 0 & n/2 & 0 & (n-1)/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

总结

多个变换的联接 the chain of transformations:

$T = T_{vp} T_{can} T_{eye} T_{rb}$



图片来自 Ravi Ramamoorthi Edx course

Any Questions?