Transformation

3D Rotation

$$R_{xyz}(\alpha,\beta,\gamma)=R_x(\alpha)R_y(\beta)R_z(\gamma)$$

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

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

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

对任意轴旋转（Rodrigues' Rotation Formula: 罗德里格旋转公式）：

设 v_{rot} 是一个三维空间向量，***k***是旋转轴的单位向量，则***v***在右手螺旋定则意义下绕旋转轴***k***旋转角度θ得到的向量可以由三个不共面的向量***v***, ***k***和***k***×***v***构成的标架表示：

$$v_{rot}=cos\theta v+(1-cos\theta)(v\cdot k)k+sin\theta k\times v$$

而在计算机图形学里，用此公式来填写旋转矩阵，把 k 和 v 分别写为列向量，则旋转以后的向量可以表示为 $v_{rot}=Rv$，其中：

$$R=cos(\alpha)I+(1-cos(\alpha))kk^T+sin(\alpha)\begin{pmatrix}0 & -k_z & k_y\\ k_z & 0 & -k_x\\ -k_y & k_x & 0\end{pmatrix}$$

I 为对应阶单位矩阵，kk^T乘出来应为一3×3矩阵。

Model transformation：模型变换

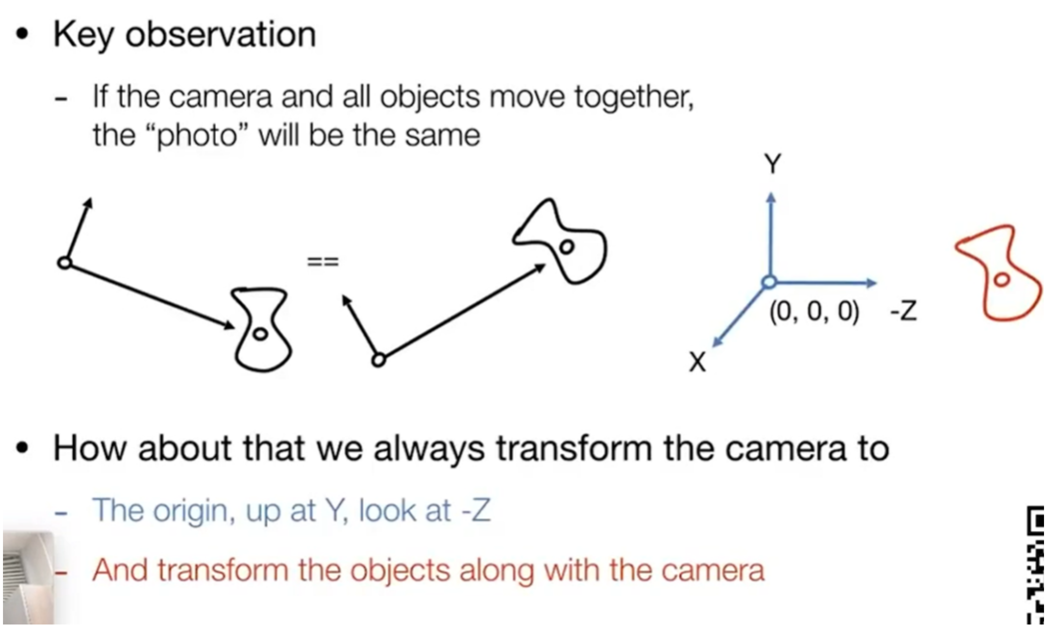
相当于将模型摆好，放到好的位置上。（通过平移其他东西）

Viewing transformation：观测变换

View / Camera Transformation：视图变换

- Position：位置 $\vec{e}$
- Look-at / gaze direction：往哪看 $\hat{g}$
- Up direction：向上方向（把旋转固定住） $\hat{t}$

约定俗成：



将相机移到原点，上方是Y轴，看向-Z轴，需要做一个变换。

而要把$\vec{e}, \hat{g}, \hat{t}$变成三个轴并不容易，所以采用逆变换，先求逆矩阵（将X，Y，Z轴转成$\vec{e}, \hat{g}, \hat{t}$）而由于**旋转矩阵**是正交矩阵，所以直接转置即可。

因此，我们可以移动相机，通过矩阵M_{view}。

$$M_{view} = R_{view} T_{view}$$

其中：T_{view}为平移矩阵，将矩阵平移到原点：

$$T_{view} = \begin{pmatrix} 1 & 0 & 0 & -x_e \\ 0 & 1 & 0 & -y_e \\ 0 & 0 & 1 & -z_e \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

旋转矩阵上面已说明如何求：

$$R_{view} = \begin{pmatrix} x_{\hat{g}\times\hat{t}} & y_{\hat{g}\times\hat{t}} & z_{\hat{g}\times\hat{t}} & 0 \\ x_t & y_t & z_t & 0 \\ x_{-g} & y_{-g} & z_{-g} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Also known as ModelView Transformation

不用坐标系变换是因为不直观。

Projection Transformation：投影变换

Orthographic projection：正交投影

认为相机离得无限远。

更多用于工程制图，正交投影并不会带来一种近大远小的现象，但透视投影会（一叶障目）。

一种简单的投影方法：去除Z轴

常用的方法：先平移（先把中心移到原点上）再缩放（把x, y, z都变成2）（注意是先平移）。

变换矩阵：

$$M_{ortho} = \begin{pmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & \frac{2}{n-f} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -\frac{r+l}{2} \\ 0 & 1 & 0 & -\frac{t+b}{2} \\ 0 & 0 & 1 & -\frac{n+f}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Perspective projection：透视投影

将摄像机认为一个点。平行线不再平行。

将透视投影的Frustum变为一个立方体（$M_{persp->ortho}$），然后做正交投影。

在这里，对于$(x, y, z, 1)$这一点，变形后应是$(\frac{nx}{z}, \frac{ny}{z}, unknown, 1)$，（为什么是unknown，之后会说明，只能说在近和远的两个平面上z不变。）

利用齐次坐标系的性质，$(x, y, z, 1)$和(xz, yz, y^2, z)表示的是同一个点。

证：对于透视投影除最近平面和最远平面上的任意一点$(x, y, z, 1)$，变形为正交投影后z发生变换，且往原点靠近：

首先，$M_{persp\to ortho}^{4\times 4}$为：

$$\begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -nf \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

因此，变形后的点应为：$(nx, ny, (n+f)z - nf, z)$，若全部按比例缩放，原本应为：(nx, ny, nz, z)

因为显然，$(n+f)z - nf - nz = f(z-n) > 0$，也就是说，实际缩放后的点，要比按比例缩放的点离 z 轴更近，所以其实往“相机”方向推动。

最后：

$$M_{persp} = M_{ortho} M_{persp\to ortho}$$