课程介绍

吴雨婷
ytwu1@bjtu.edu.cn

朱桂萍
22110144@bjtu.edu.cn

《GAMES101:现代计算机图形学入门》

平时成绩0.5

> 4次课后作业，包含课堂小测试(20%)
> 2次大作业(80%)
>
> 大作业自由组队: 5~6人一组

课程论文0.5

计算机图形系统

交互式计算机图形处理系统

交互式 = 计算机 + 人

图形输入设备

> 键盘、鼠标
> 光笔
> 触摸屏
> 操纵杆
> 数据手套
> 数字化仪
> 图形扫描仪

> 声频输入系统
> 视频输入系统
> 真实物体的三维信息输入

图形输出设备

硬拷贝: 打印机和绘图仪

阴极射线管: CRT监视器(指标: 分辨率、显示速度)

彩色阴极射线管

CRT图形显示器

随机扫描的图形显示器

> 速度快，不用全屏扫描
> 为画线应用设计的，不能显示逼真的有阴影场景

光栅扫描显示器

> 电子束横向扫描屏幕，一次一行，从顶到底顺次进行。
> 电子束横向沿每一行移动时，电子束强度不断变化来建立亮点的图案

LCD液晶显示器

液晶是一种介于液体和固体之间的特殊物质，受电压影响，改变物理性质而发生形变

图形处理器

图形处理器(显卡)

> 主要配件有**显示主芯片、显存和数字模拟转换器(RAMDAC)**
>
> **显示主芯片**是显卡的核心，俗称GPU,它的主要任务是对系统输入的视频信息进行构建和渲染，各图形函数基本上都集成在这里
>
> **显存**用于存储将要显示的图形信息及保存图形运算的中间数据,它与显示主芯片的关系就像计算机的内存与CPU一样密不可分
>
> **RAMDAC**就是视频存储数字模拟转换器。在视频处理中,它的作用就是把二进制的数字转换成为和显示器相适应的模拟信号

OpenGL编程

变换

二维变换

线性变换

> **线性变换保持向量加法和标量乘法不变**
>
> 可加性
> 齐次性

$$
\begin{aligned}
x' &= a\,x + b\,y \\
y' &= c\,x + d\,y
\end{aligned}
$$

$$
\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}
$$

$$
\mathbf{x}' = \mathbf{M}\,\mathbf{x}
$$

缩放变换

×系数，**对角矩阵**（缩放矩阵）

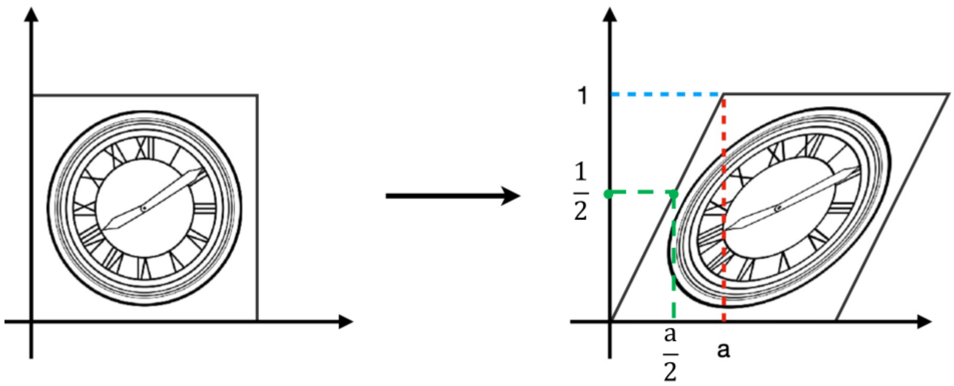
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

反射变换

(-1 0
 0 1)

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

切变(Shear)



可以观察到：

1. y坐标不变；
2. 当$y=0$时，x坐标不变；
3. 当$y=1$时，x坐标变为$x+a$;

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

想要写出变换，找出变换前后的x, y坐标的关系

旋转(Rotate)

默认绕原点为中心，逆时针方向旋转

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

齐次坐标

平移不是线性变换

Add a third coordinate (*w-coordinate*)

- 2D point = $(x, y, 1)^T$
- 2D vector = $(x, y, 0)^T$

Matrix representation of translations

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + t_x \\ y + t_y \\ 1 \end{pmatrix}$$

向量的平移不变性

矩阵左上为二维变换，右侧为平移变换，最下一行是0, 0, 1

In homogeneous coordinates,

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} \text{ is the 2D point } \begin{pmatrix} x/w \\ y/w \\ 1 \end{pmatrix}, \quad w \neq 0$$

点+点是中点

仿射变换(Affine Transformations)

Affine map = linear map + translation

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Using homogenous coordinates:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Scale

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

Rotation

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

Translation

$$\mathbf{T}(t_x, t_y) = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

逆变换(Inverse Transform)

逆矩阵，**旋转矩阵逆矩阵=转置**(因为是正交矩阵)

组合变换

多个变换进行组合，先旋转，矩阵乘法不满足交换律
从右往左

三维变换

类比二维

Use homogeneous coordinates again:

- **3D point = (x, y, z, 1)^T**
- **3D vector = (x, y, z, 0)^T**

In general, (x, y, z, w) (w != 0) is the 3D point:

(x/w, y/w, z/w)

仿射变换

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c & t_x \\ d & e & f & t_y \\ g & h & i & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

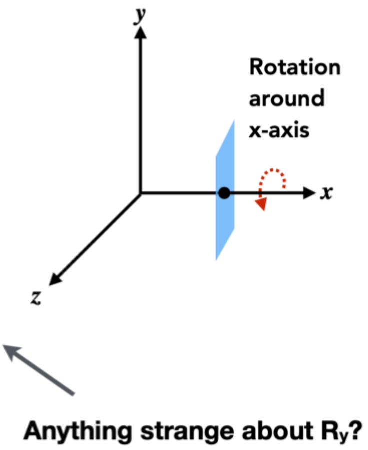
三维旋转

Rotation around x-, y-, or z-axis

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

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

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



任意三维旋转可以变成这三种转轴的**旋转的组合**

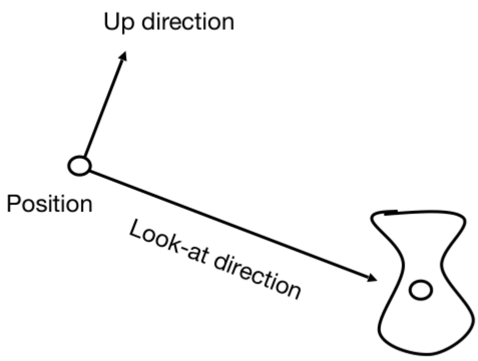
观测变换

MVP = model, view, projection

视图变换(View / Camera transformation)

定义相机

> Position
> Look-at /gaze direction
> Up direction



always

- Position位置 - **原点**，up direction向上方向 - **y轴**，look-at看 - **-z方向**
- 随着相机变换物体

视图变换实现相机

- M_{view} in math?
  - Let's write $M_{view} = R_{view}T_{view}$
  - Translate e to origin

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

  - Rotate g to -Z, t to Y, (g x t) To X
  - Consider its inverse rotation: X to (g x t), Y to t, Z to -g

$$R_{view}^{-1} = \begin{bmatrix} x_{\hat{g}\times\hat{t}} & x_t & x_{-g} & 0 \\ y_{\hat{g}\times\hat{t}} & y_t & y_{-g} & 0 \\ z_{\hat{g}\times\hat{t}} & z_t & z_{-g} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{\text{WHY?}} R_{view} = \begin{bmatrix} x_{\hat{g}\times\hat{t}} & y_{\hat{g}\times\hat{t}} & z_{\hat{g}\times\hat{t}} & 0 \\ x_t & y_t & z_t & 0 \\ x_{-g} & y_{-g} & z_{-g} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

不知道角度，不能用θ

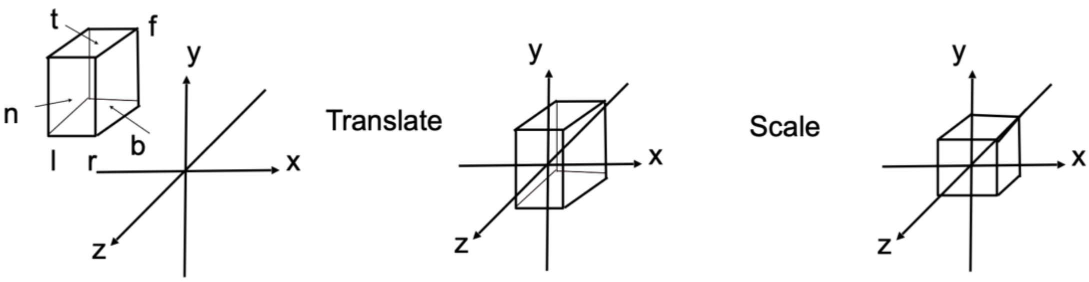
投影变换

正交投影(Orthographic projection)

不会受近大远小影响

移动到原点然后**缩放**

- We want to map a cuboid [l, r] x [b, t] x [**f, n**] to the “canonical (正则、规范、标准)” cube $[-1, 1]^3$



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

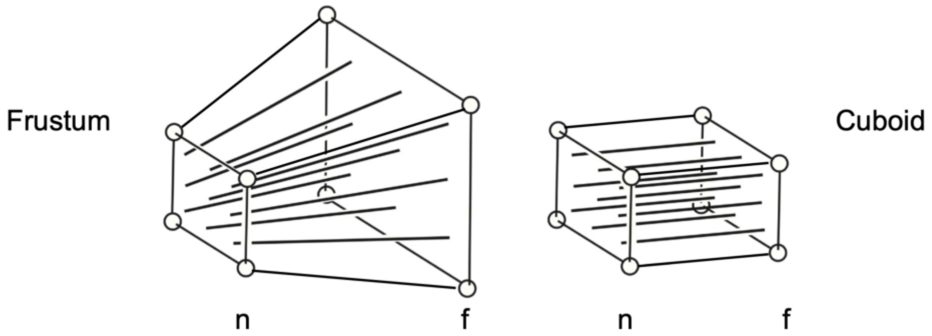
注意!

OpenGL使用的是左手系，所以有些区别

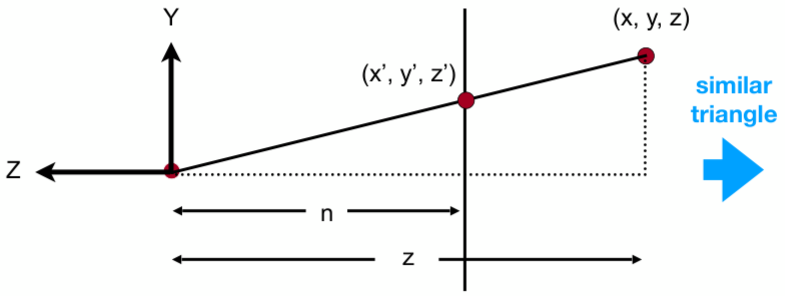
透视投影(Perspective projection)

一个点乘一个不为0的constant还是那个点

> 先把frustum压缩变为cuboid，再做**正交投影**
>
> 近平面所有点不变，中心点不变
>
> 通过相似三角形来求解



- In order to find a transformation
  - Recall the key idea: Find the relationship between transformed points (x’, y’, z’) and the original points (x, y, z)



$$y' = \frac{n}{z}y$$

- In order to find a transformation
  - Find the relationship between transformed points (x’, y’, z’) and the original points (x, y, z)

$$y' = \frac{n}{z}y \qquad x' = \frac{n}{z}x \quad \text{(similar to y')}$$

- In homogeneous coordinates,

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} nx/z \\ ny/z \\ \text{unknown} \\ 1 \end{pmatrix} \overset{\text{mult. by z}}{==} \begin{pmatrix} nx \\ ny \\ \text{still unknown} \\ z \end{pmatrix}$$

不能直接判断z
但可以**通过f中点来计算**

- Any point on the near plane will not change

$$M^{(4\times4)}_{persp\to ortho}\begin{pmatrix}x\\y\\z\\1\end{pmatrix}=\begin{pmatrix}nx\\ny\\\text{unknown}\\z\end{pmatrix} \xrightarrow{\text{replace } z \text{ with } n} \begin{pmatrix}x\\y\\n\\1\end{pmatrix}\Rightarrow\begin{pmatrix}x\\y\\n\\1\end{pmatrix}==\begin{pmatrix}nx\\ny\\n^2\\n\end{pmatrix}$$

- So the third row must be of the form (0 0 A B)

$$\begin{pmatrix}0 & 0 & A & B\end{pmatrix}\begin{pmatrix}x\\y\\n\\1\end{pmatrix}=n^2$$

n² has nothing to do with x and y

- What do we have now?

$$\begin{pmatrix}0 & 0 & A & B\end{pmatrix}\begin{pmatrix}x\\y\\n\\1\end{pmatrix}=n^2 \Rightarrow An+B=n^2$$

- Any point's z on the far plane will not change

$$\begin{pmatrix}0\\0\\f\\1\end{pmatrix}\Rightarrow\begin{pmatrix}0\\0\\f\\1\end{pmatrix}==\begin{pmatrix}0\\0\\f^2\\f\end{pmatrix} \Rightarrow Af+B=f^2$$

- Solve for A and B

$$\begin{aligned}An+B&=n^2\\Af+B&=f^2\end{aligned} \Rightarrow \begin{aligned}A&=n+f\\B&=-nf\end{aligned}$$

- Finally, every entry in $M_{persp\text{->}ortho}$ is known!

最后进行正交透视

$$M_{persp}=M_{ortho}M_{persp->ortho}$$

光栅化

着色

几何

高级图形技术