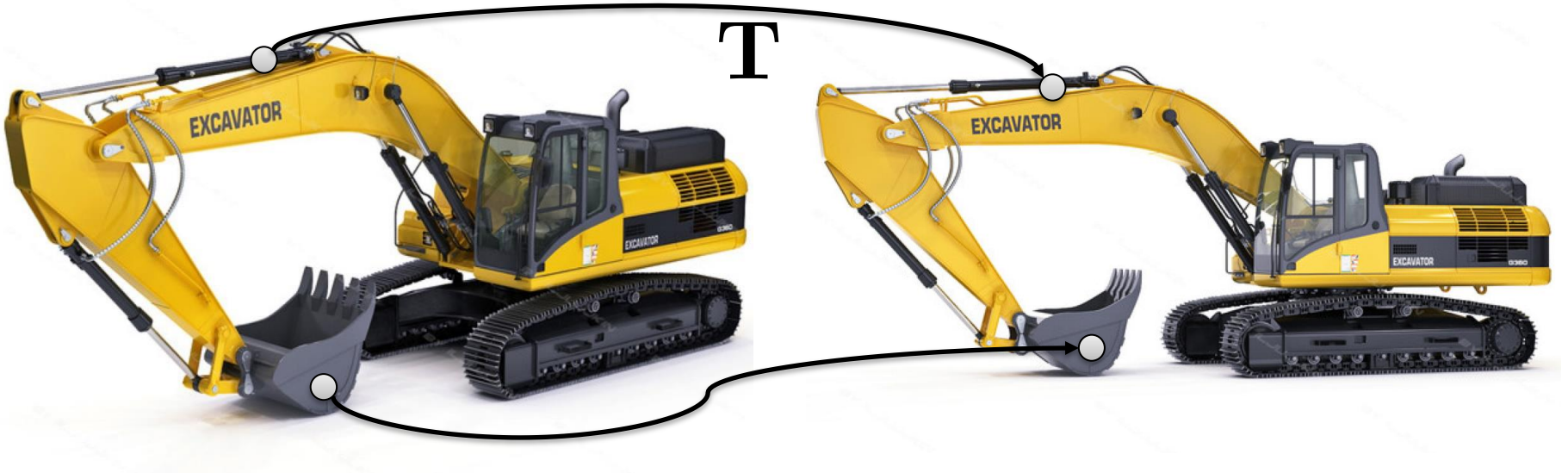
Transformations

Prof. Dr. Markus Gross

Transformations

- Transformations map geometry



Transformations

- Transformations in the graphics pipeline:
  - Change position & orientation of objects
  - Project objects to screen
  - Animate objects

Notation

- Points and vectors are represented as

$$\mathbf{p} = \begin{pmatrix} x \\ y \end{pmatrix}$$

- Matrices are represented as $\mathbf{A}$
- A point is transformed as $\mathbf{p}' = \mathbf{A}\mathbf{p}$
- Transpose: $\mathbf{p}^T = (x, y) \qquad (\mathbf{A}^T)_{i,j} = \mathbf{A}_{j,i}$

Definitions

- Linear maps

$$A(\alpha x + \beta y) = \alpha A(x) + \beta A(y)$$

- Represented by matrices

$$\mathbf{A}(\alpha \mathbf{x} + \beta \mathbf{y}) = \alpha \mathbf{A}\mathbf{x} + \beta \mathbf{A}\mathbf{y}$$

- Affine maps

$$\mathbf{A}\mathbf{x} + \mathbf{b}$$

2D Transformations

- Translation

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \Rightarrow \mathbf{p}' = \mathbf{p} + \mathbf{t}$$

- Scaling

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \Rightarrow \mathbf{p}' = \mathbf{S}\mathbf{p}$$

2D Transformations

- Rotation by angle θ

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

- In matrix form

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \Rightarrow \mathbf{p}' = \mathbf{R}\mathbf{p}$$

Homogeneous Coordinates

- Affine maps are linear maps in **homogeneous coordinates**

$$\mathbf{p} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Homogeneous Coordinates

- Translation is represented as a matrix

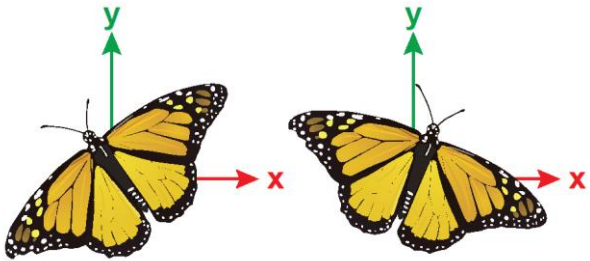
$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Rightarrow \mathbf{p}' = \mathbf{T}\mathbf{p}$$

$\mathbf{p}'$ $\mathbf{T}$ $\mathbf{p}$

Homogeneous Coordinates

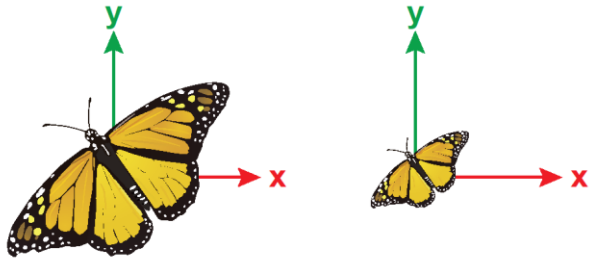
- Rotation

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



- Scaling

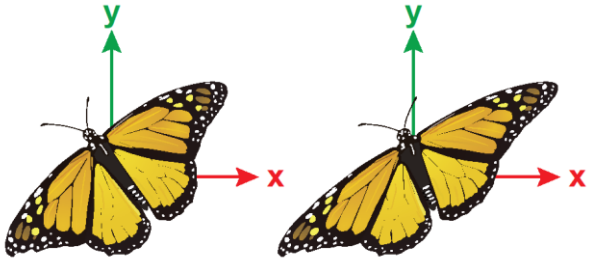
$$\mathbf{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

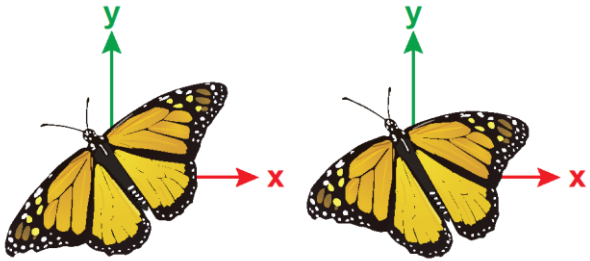


Homogeneous Coordinates

- Shear along x- and y- axis

$$\mathbf{SH}_x = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{SH}_y = \begin{bmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$





Homogeneous Coordinates

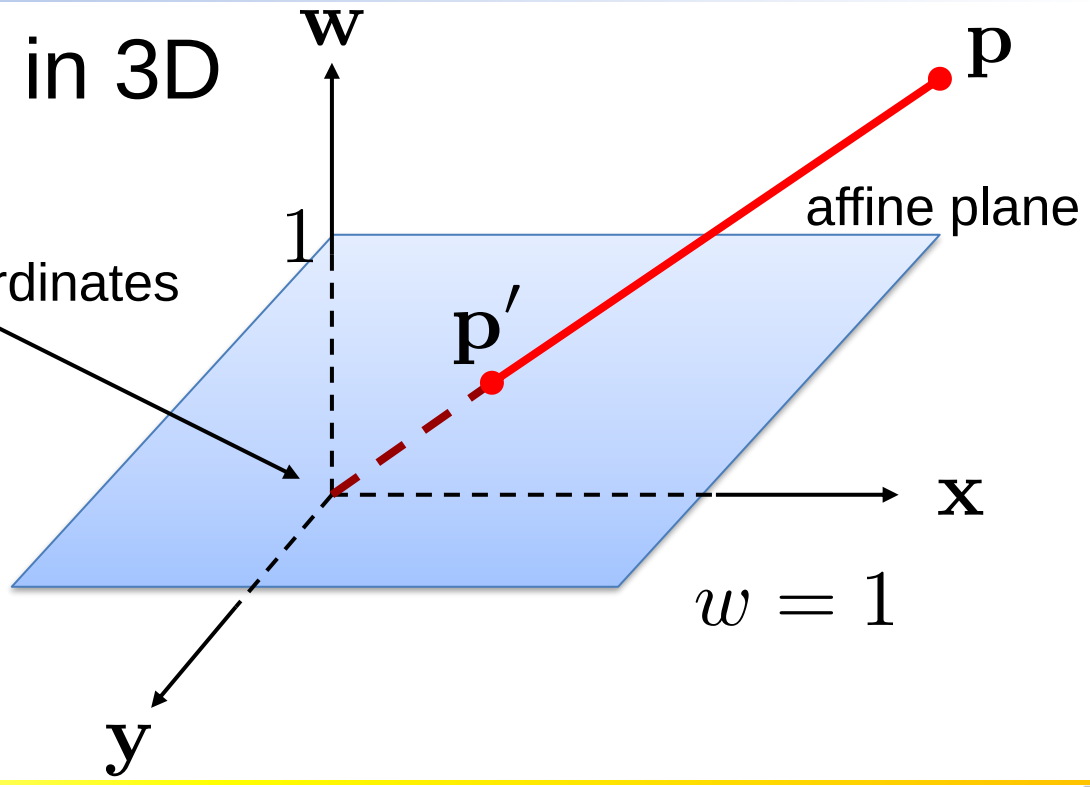
- A point has infinitely many homogeneous coordinates, for any w

$$\mathbf{p} = \begin{pmatrix} wx \\ wy \\ w \end{pmatrix}$$

Homogeneous Coordinates

- Point $\mathbf{p}$ as a line in 3D

$$\mathbf{p} = \begin{pmatrix} wx \\ wy \\ w \end{pmatrix}$$



Combining Transformations

- Combine via matrix multiplication
  - Example: rotation followed by translation

$$\mathbf{TR} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & 0 \\ r_{21} & r_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Combining Transformations

- Commutativity

$$\mathbf{M}_1\mathbf{M}_2 = \mathbf{M}_2\mathbf{M}_1$$

Matrix $\mathbf{M}_1$	Matrix $\mathbf{M}_2$
Translation	Translation
Rotation	Rotation
Scaling	Scaling
Scaling	Rotation

Only for 2D!

3D Transformations

- Homogeneous coordinates: 4x4 matrices
- Project $\mathbf{p} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$ onto the hyperplane $\begin{bmatrix} x/w \\ y/w \\ z/w \\ 1 \end{bmatrix}$

3D Transformations

- Translation

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Scaling

$$\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3D Transformations

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

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3D Transformations

- Rotation of angle θ around an axis $\mathbf{u}$

$$\mathbf{R}(\mathbf{u},\theta) = \begin{bmatrix} u_x^2 + \cos\theta(1-u_x^2) & u_x u_y(1-\cos\theta) - u_z\sin\theta & u_x u_z(1-\cos\theta) - u_y\sin\theta & 0 \\ u_x u_y(1-\cos\theta) - u_z\sin\theta & u_y^2 + \cos\theta(1-u_y^2) & u_y u_z(1-\cos\theta) - u_x\sin\theta & 0 \\ u_x u_z(1-\cos\theta) - u_y\sin\theta & u_y u_z(1-\cos\theta) - u_x\sin\theta & u_z^2 + \cos\theta(1-u_z^2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

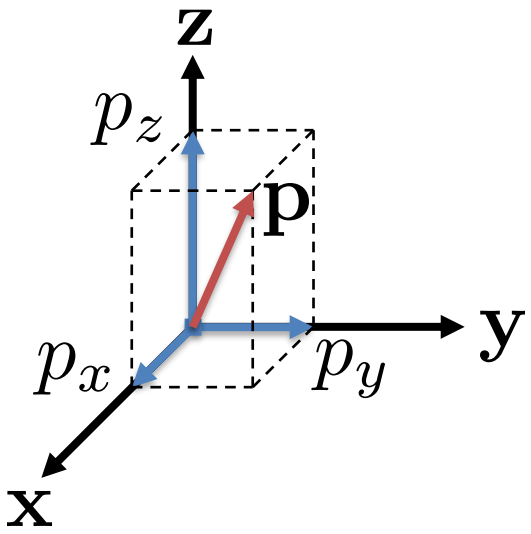
3D Transformations

- Shearing parallel to the principal planes

$$\mathbf{SH}_{xy} = \begin{bmatrix} 1 & 0 & sh_x & 0 \\ 0 & 1 & sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{SH}_{xz} = \begin{bmatrix} 1 & sh_x & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & sh_z & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{SH}_{yz} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ sh_y & 1 & 0 & 0 \\ sh_z & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Coordinate Systems

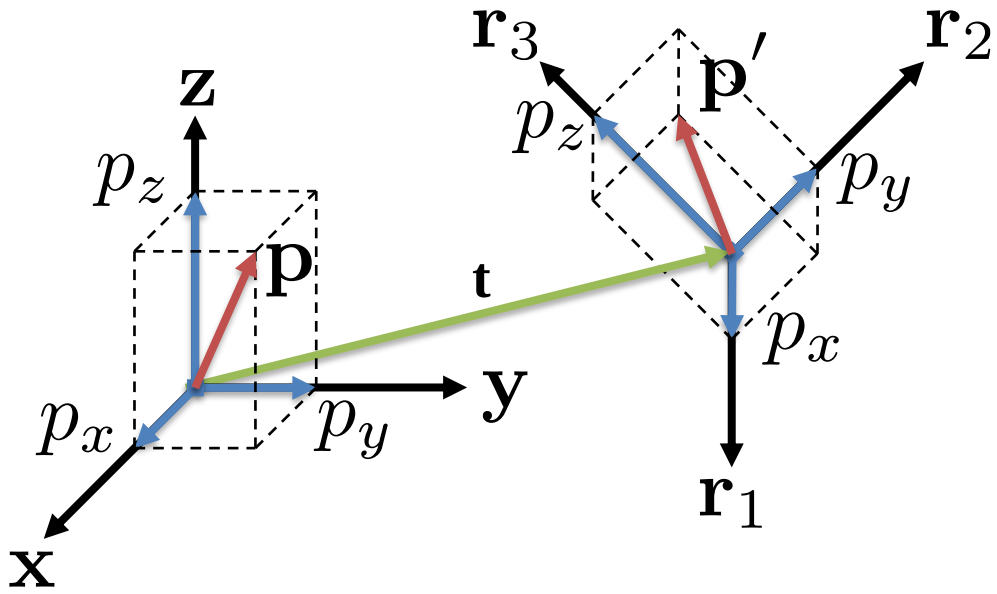
- Represent a point/vector as a linear combination of orthonormal basis vectors



$$\mathbf{p} = p_x\mathbf{x} + p_y\mathbf{y} + p_z\mathbf{z}$$

Coordinate Systems

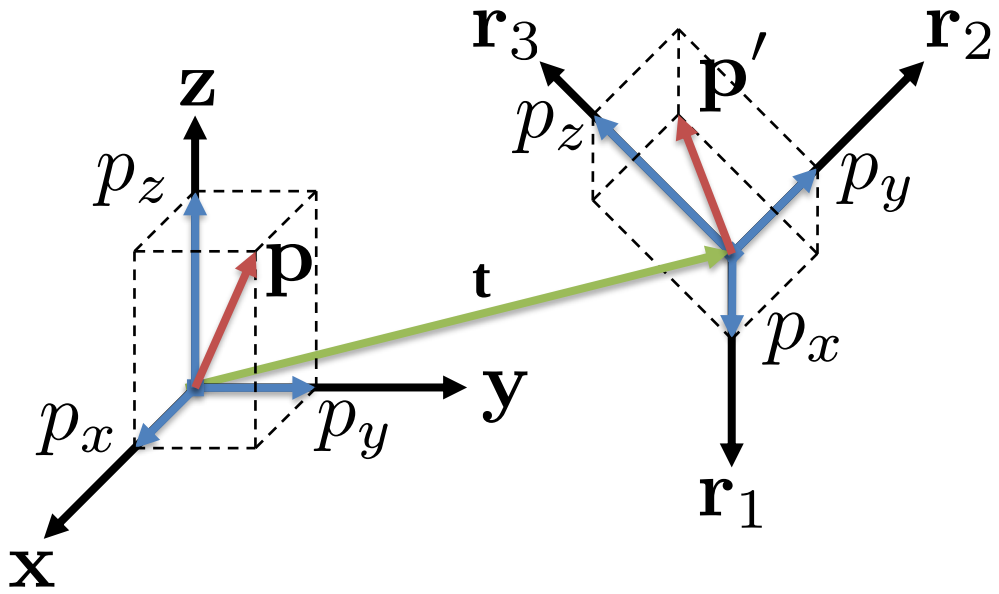
- Change of coordinate systems



$$\mathbf{p}' = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

Coordinate Systems

- Change of coordinate systems

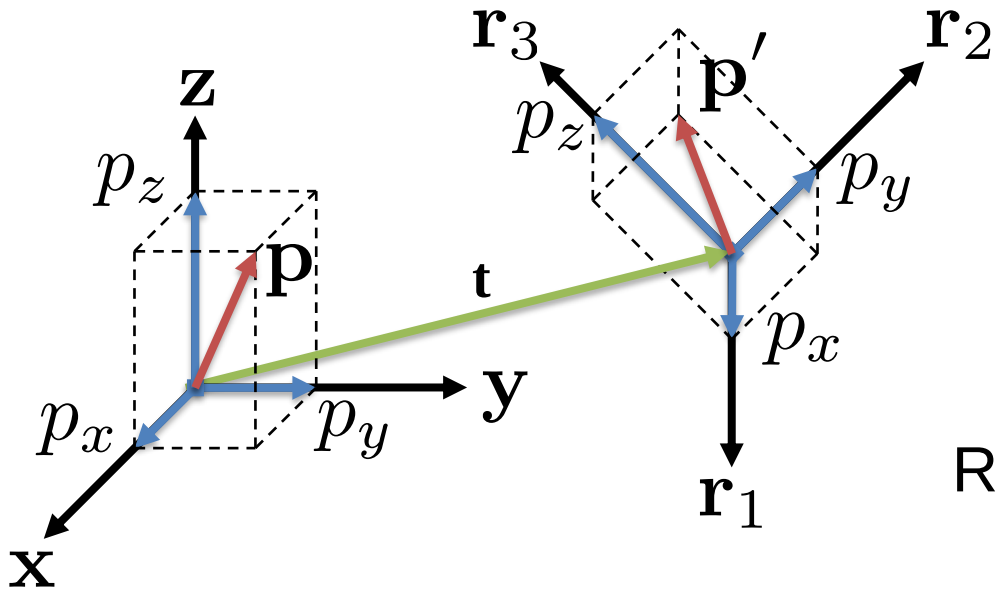


$$\mathbf{p}' = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{t} + p_x\mathbf{r}_1 + p_y\mathbf{r}_2 + p_z\mathbf{r}_3 \\ 1 \end{bmatrix}$$

Coordinate Systems

- Change of coordinate systems

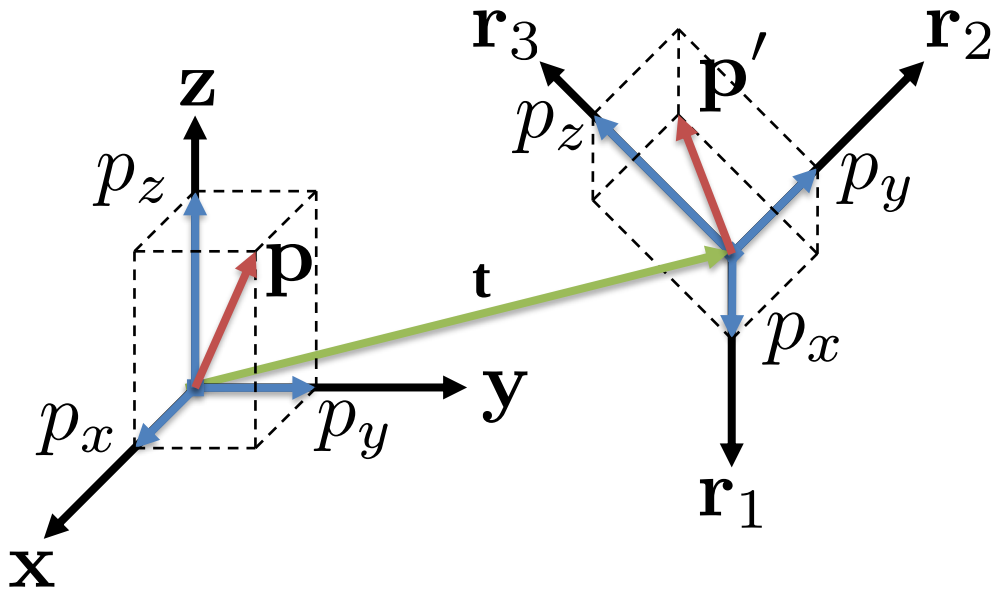


$$\mathbf{p}' = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

Rotation Translation

Coordinate Systems

- Change of coordinate systems

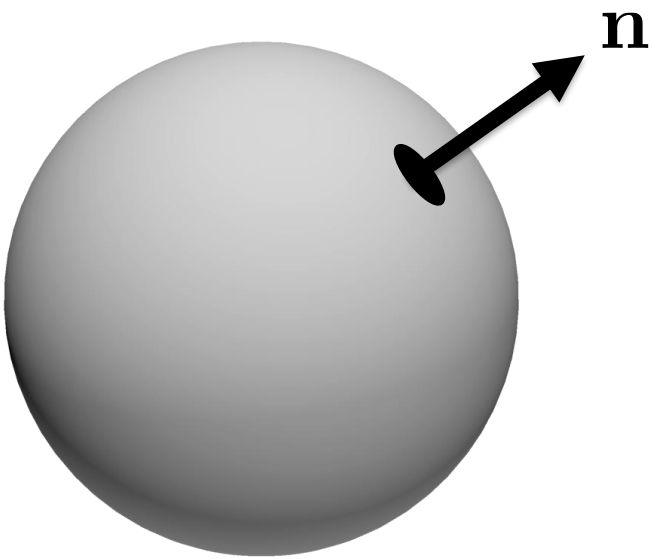


$$\mathbf{p}' = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

$$= \mathbf{TRp}$$

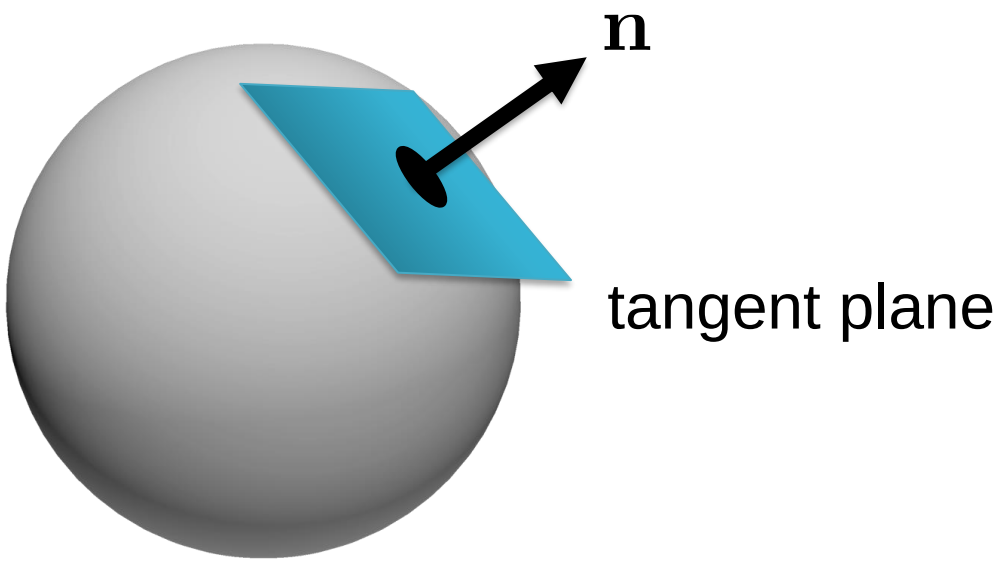
Transforming Normal Vectors

- Surface normal



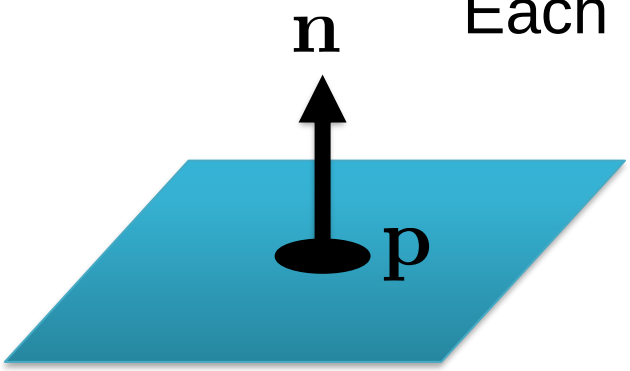
Transforming Normal Vectors

- Surface normal



Transforming Normal Vectors

- How to transform a normal when $\mathbf{p}' = \mathbf{M}\mathbf{p}$



Each $\mathbf{p} = (x, y, z, 1)$ on the plane satisfies

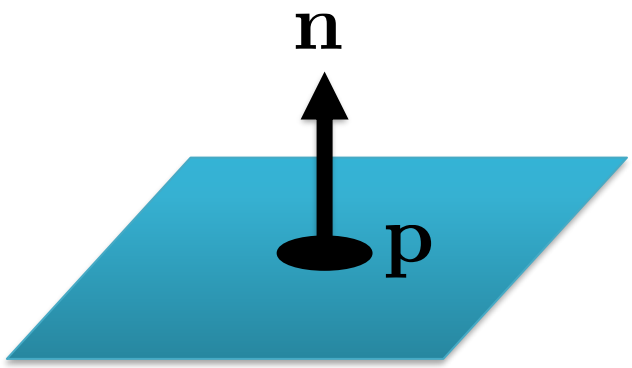
$$Ax + By + Cz + D = 0$$

Then the normal is given by

$$\mathbf{n} = \begin{pmatrix} A & B & C & D \end{pmatrix}$$

Transforming Normal Vectors

- How to transform a normal when $\mathbf{p}' = \mathbf{M}\mathbf{p}$



Current normal

$$\mathbf{n} = \begin{pmatrix} A & B & C & D \end{pmatrix}$$

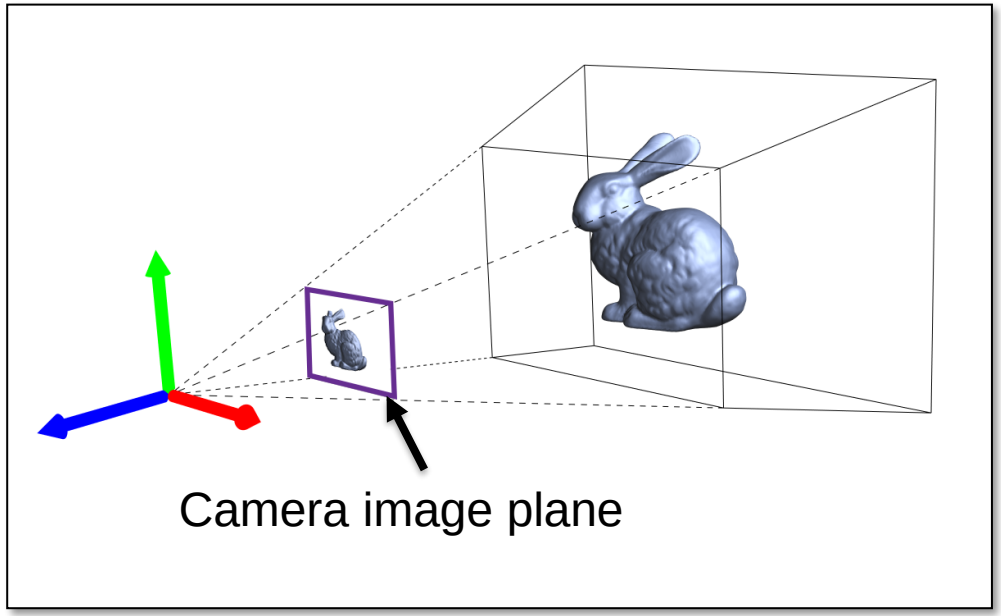
Transformed normal

$$\mathbf{n}' = (\mathbf{M}^{-1})^T \mathbf{n}$$

Verify by some algebra!
(Hint: the plane is given by $\mathbf{n}^T \mathbf{p} = 0$)

Projection

- From 3D to 2D space

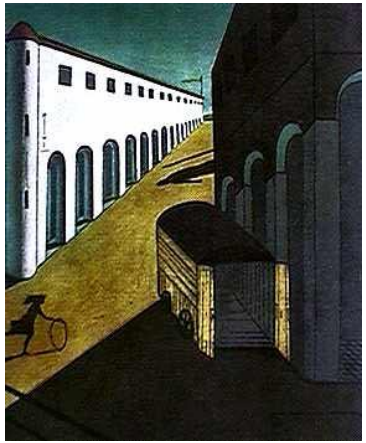


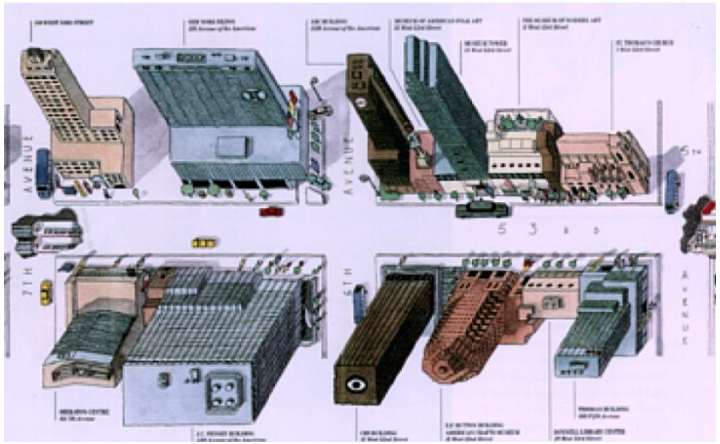
Projection

- From 3D to 2D space

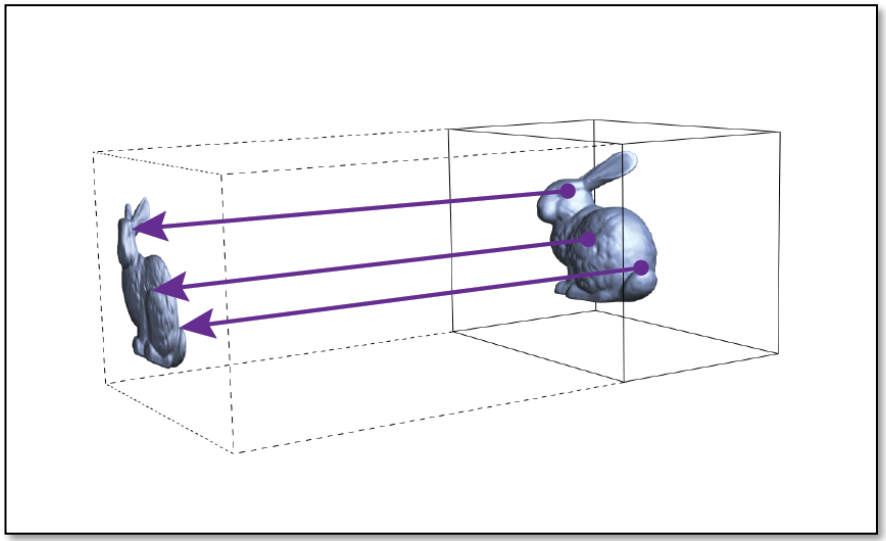
Projection

- Perspective

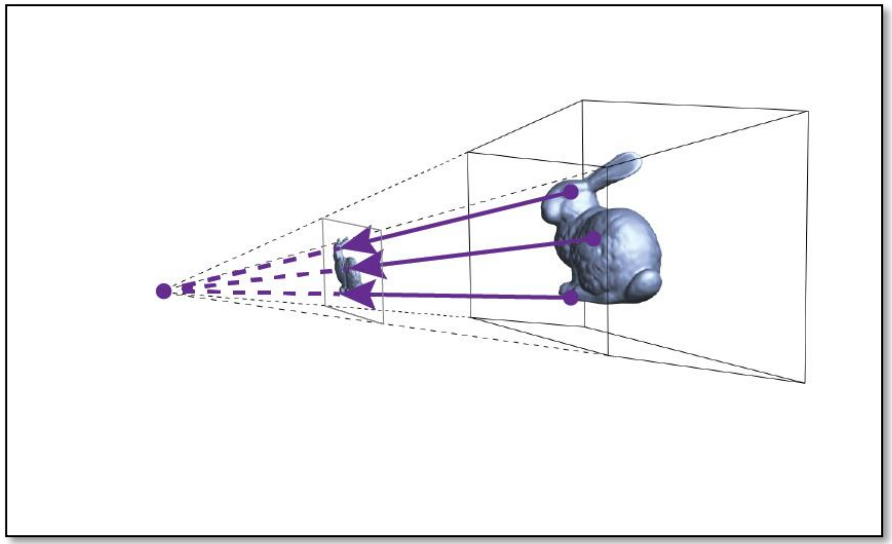




Parallel vs. Perspective Projection

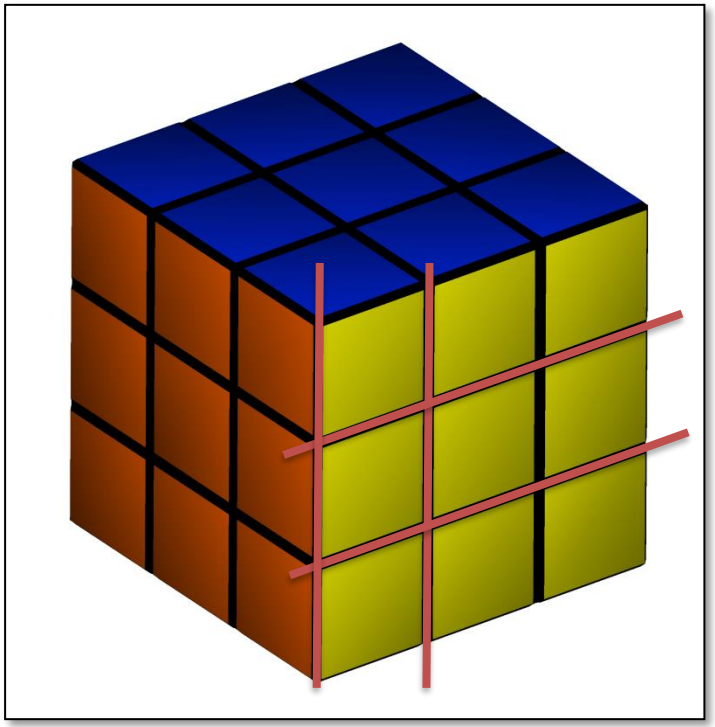


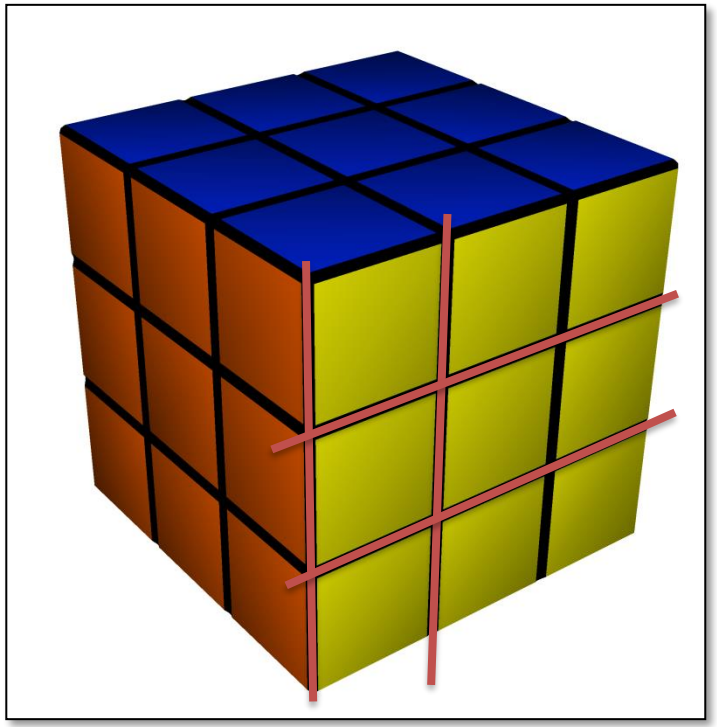
Parallel Projection



Perspective Projection

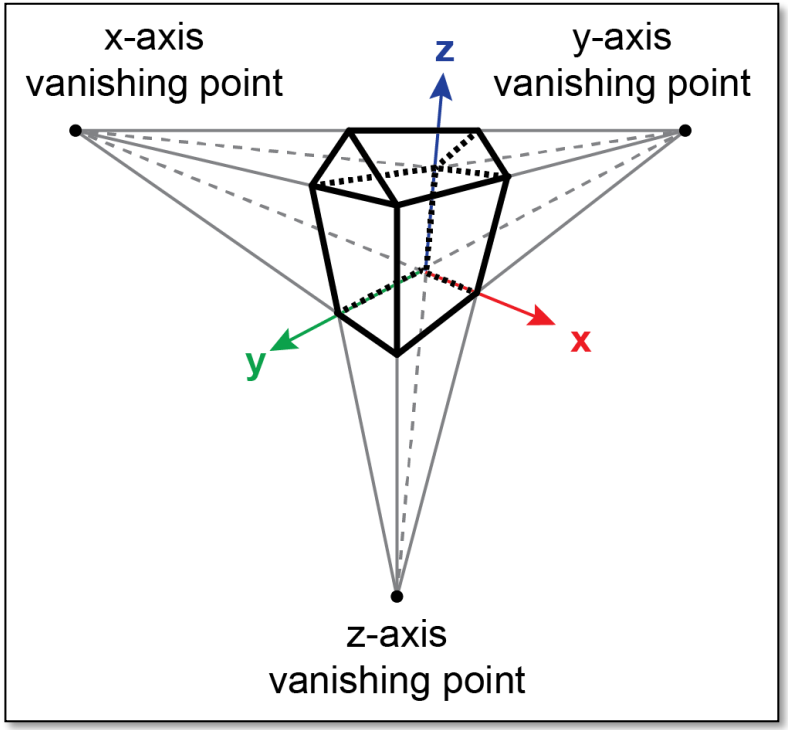
Parallel vs. Perspective Projection





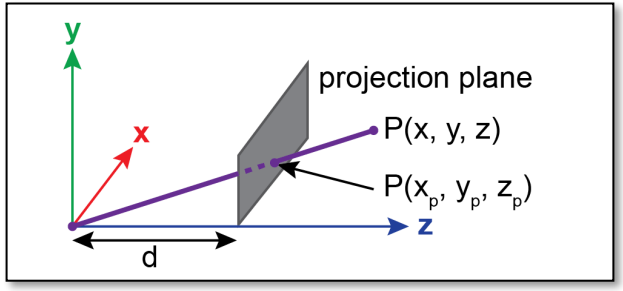
Perspective Projection

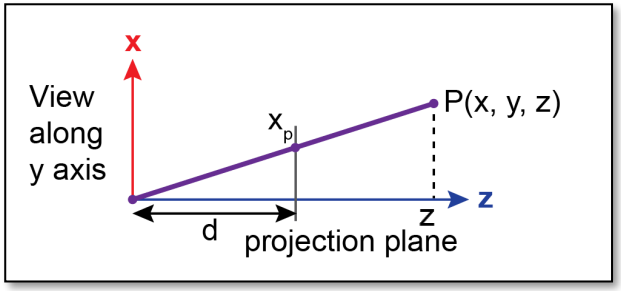
- Vanishing points

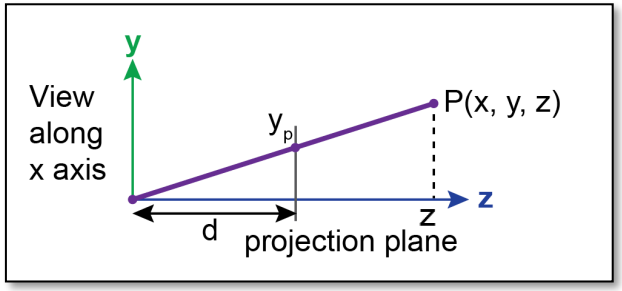


Perspective Projection

- Mathematics of perspective projection

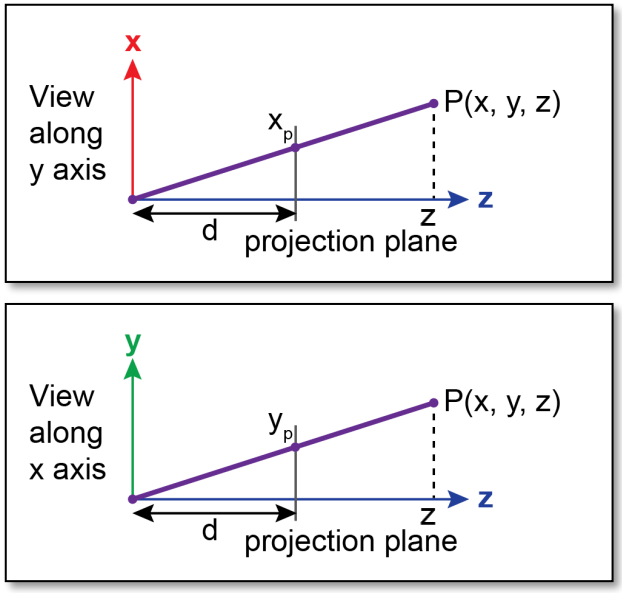






Perspective Projection

- Mathematics of perspective projection



$$x_p = dx/z \qquad y_p = dy/z$$

$$\mathbf{M}_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}$$

Perspective Projection

- Mathematics of perspective projection

3D Coordinate

$$\mathbf{M}_{per}\mathbf{p} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix} \qquad \begin{bmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ d \end{bmatrix}$$

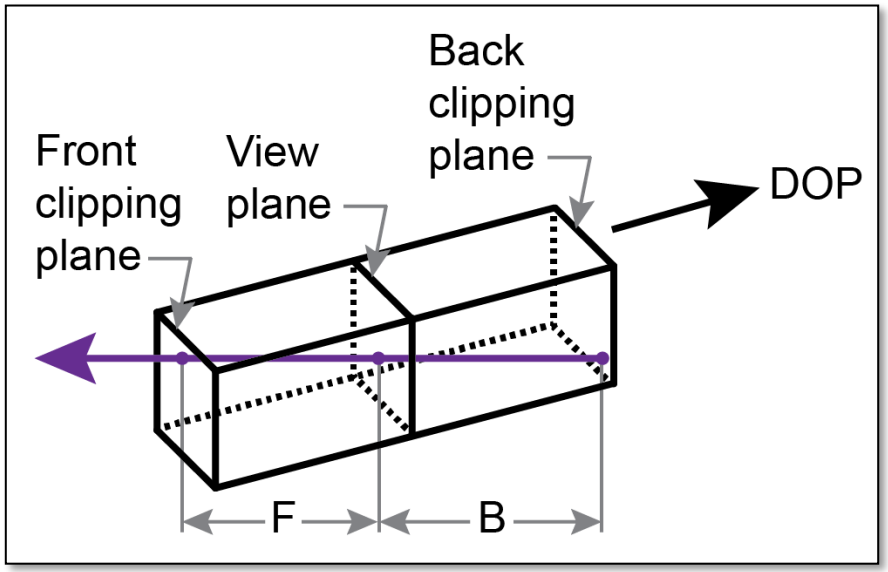
Parallel Projection

- Mathematics of parallel projection

$$\mathbf{M}_{ort} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

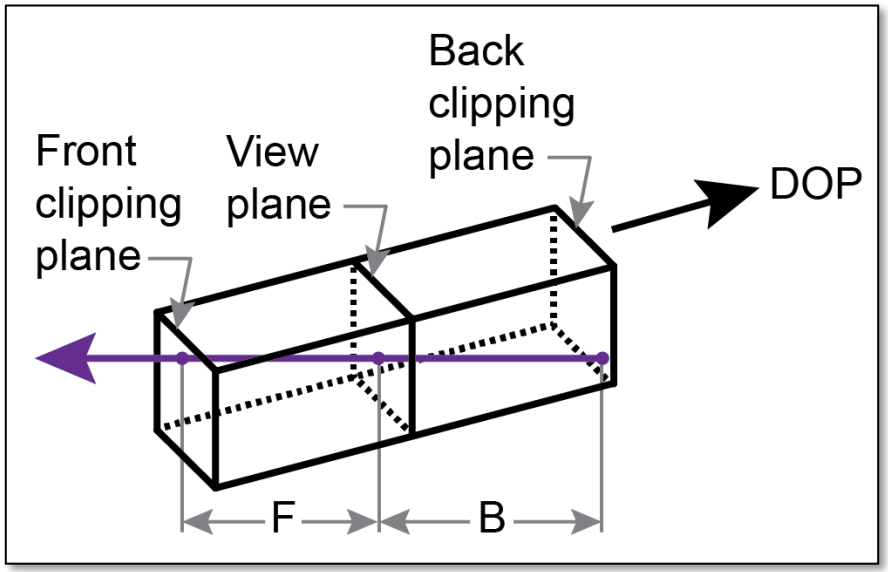
Clipping Planes

- Parallel projection

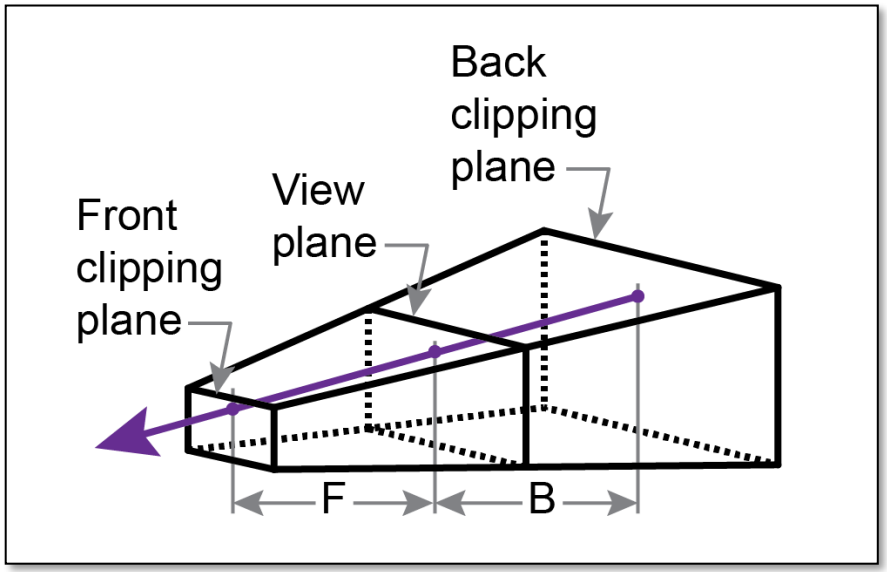


Clipping Planes

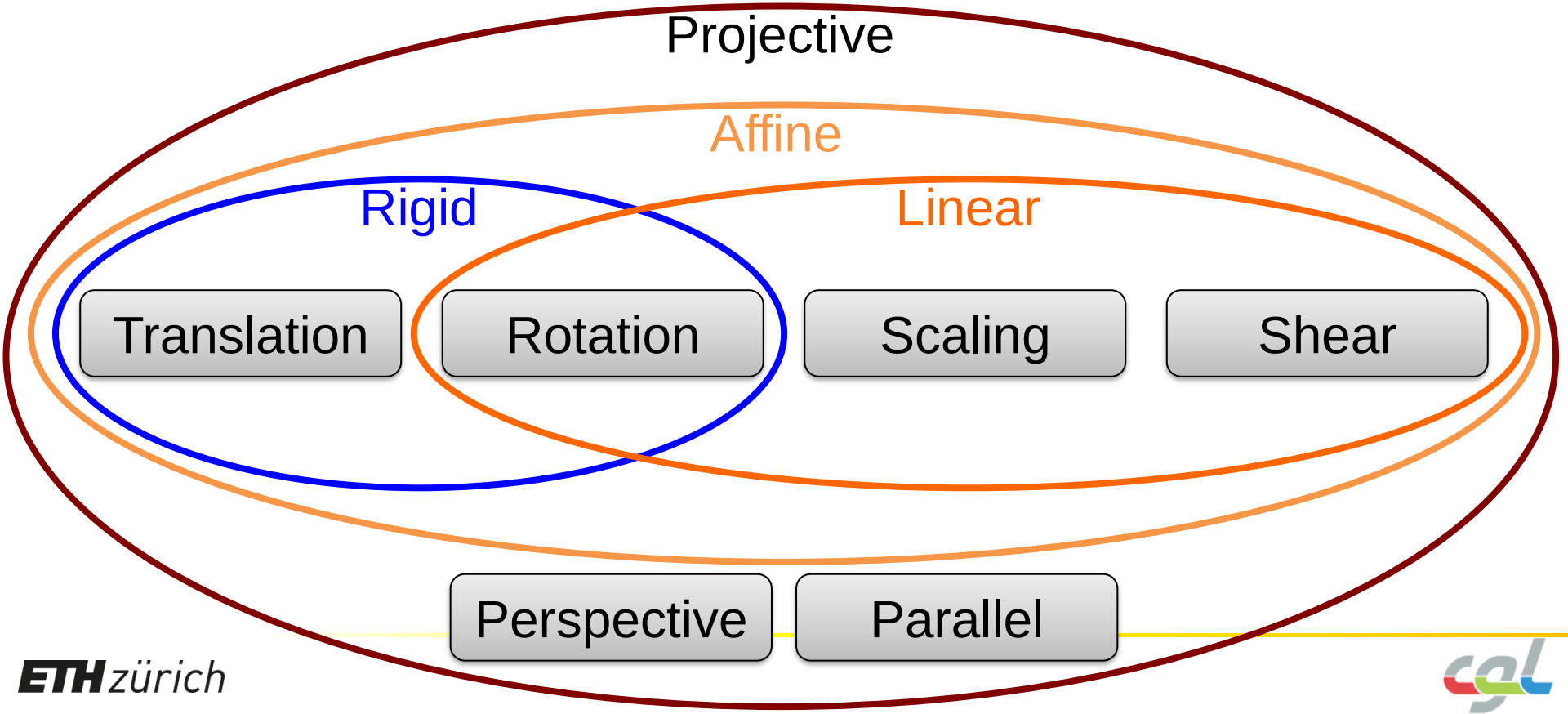
- Parallel projection



- Perspective projection

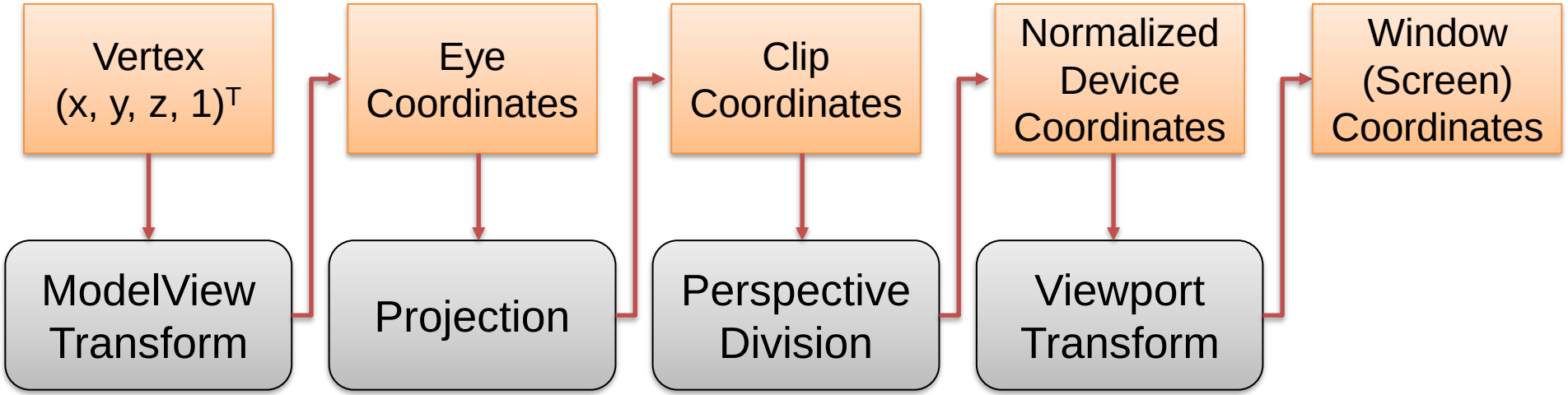


Summary of Transformations



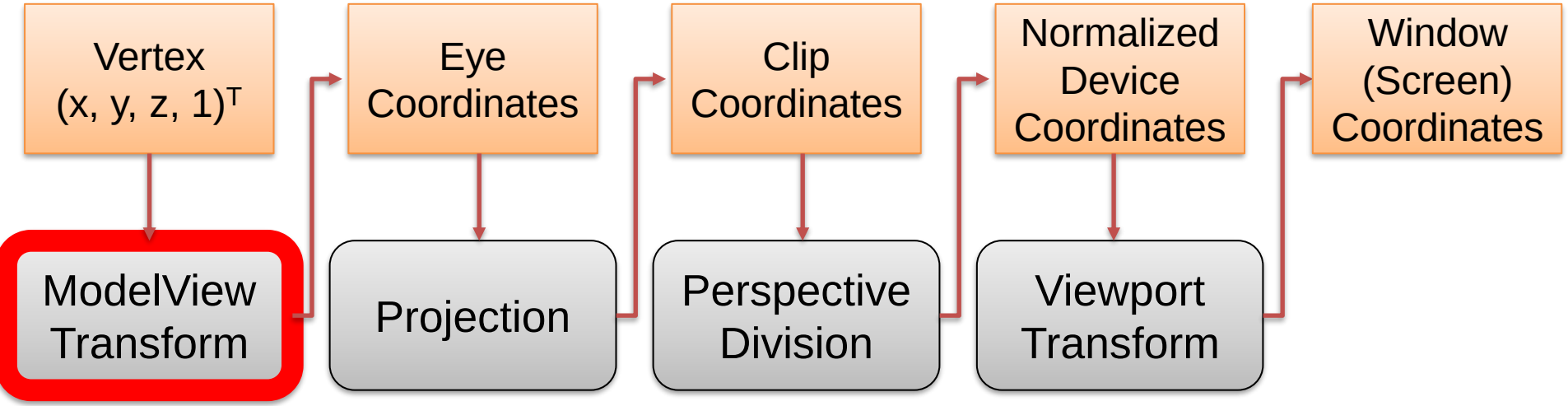
Transformations in OpenGL

- Stages of transformations



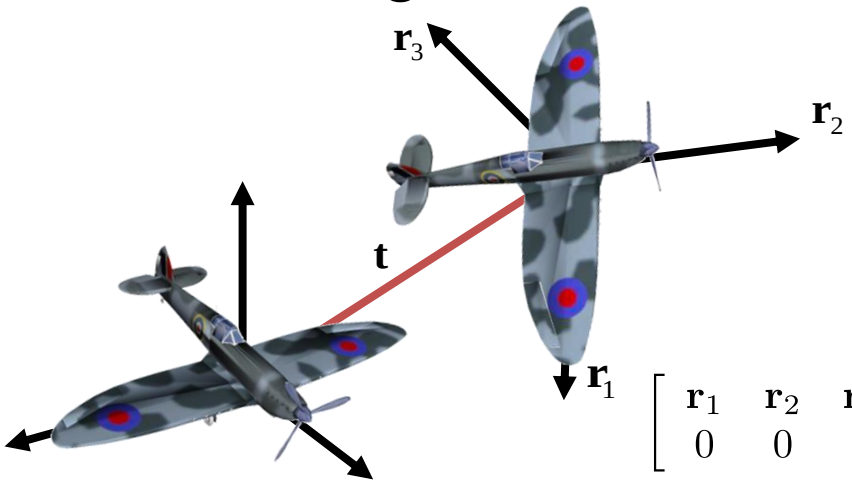
Transformations in OpenGL

- Stages of transformations



Transformations in OpenGL

- ModelView Transform
  - Stage 1: Model to world coordinates



Model Coordinates

World Coordinates

$$\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_x \\ m_y \\ m_z \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{t} + m_x\mathbf{r}_1 + m_y\mathbf{r}_2 + m_z\mathbf{r}_3 \\ 1 \end{bmatrix} = \begin{bmatrix} w_x \\ w_y \\ w_z \\ 1 \end{bmatrix}$$

Transformations in OpenGL

- ModelView Transform
  - Stage 2: World to camera coordinates

$$\begin{bmatrix} \mathbf{left} & \mathbf{up} & -\mathbf{dir} & \mathbf{eye} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_x \\ c_y \\ c_z \\ 1 \end{bmatrix} = \begin{bmatrix} w_x \\ w_y \\ w_z \\ 1 \end{bmatrix}$$

Eye (Camera) Coordinates

World Coordinates

Default in OpenGL:

$$\mathbf{left} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T$$

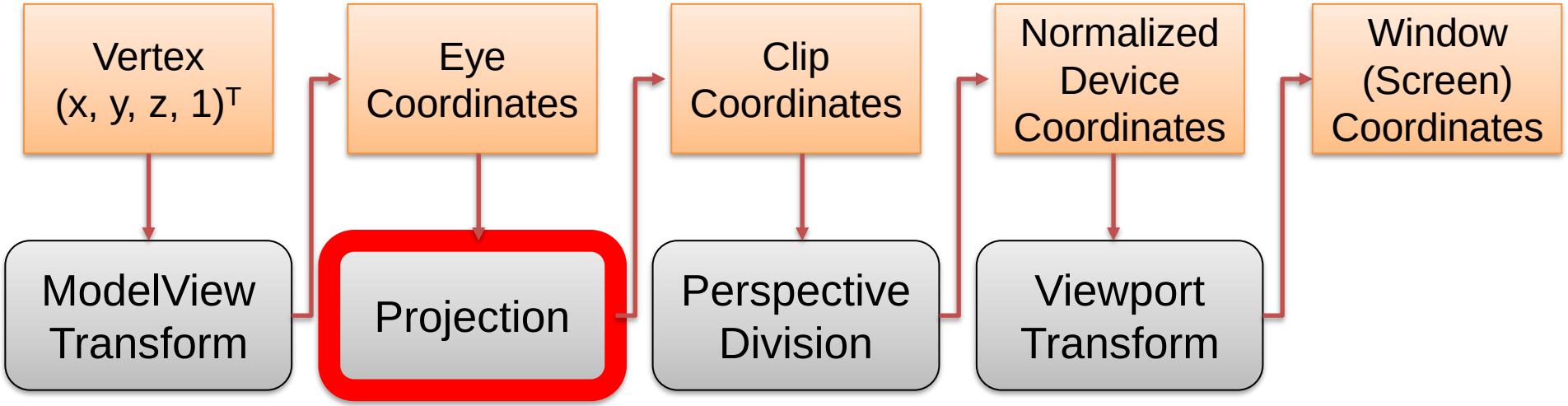
$$\mathbf{up} = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}^T$$

$$\mathbf{dir} = \begin{pmatrix} 0 & 0 & -1 \end{pmatrix}^T$$

$$\mathbf{eye} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$$

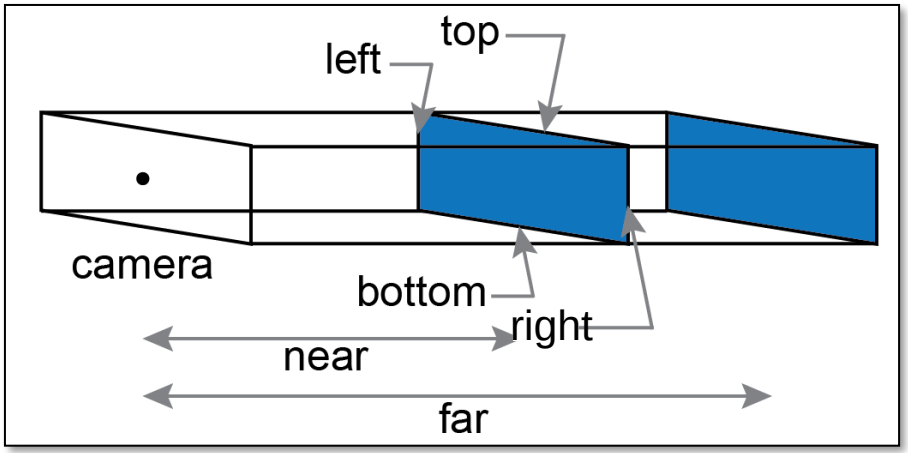
Transformations in OpenGL

- Stages of transformations



Transformations in OpenGL

- Projection
  - Option 1: Parallel projection




```
glOrtho(left, right, bottom, top, near, far);
```

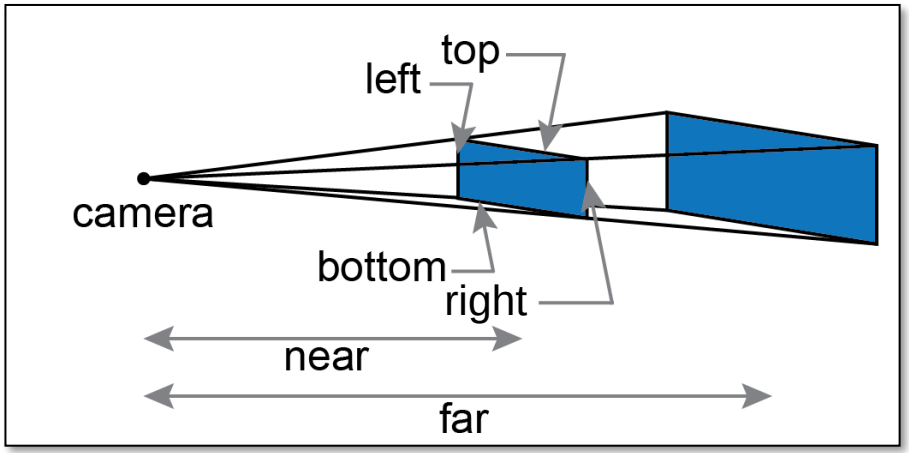
Transformations in OpenGL

- Projection
  - Option 1: Parallel projection

$$\begin{bmatrix} c'_x \\ c'_y \\ c'_z \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{right-left} & 0 & 0 & -\frac{right+left}{right-left} \\ 0 & \frac{2}{top-bottom} & 0 & -\frac{top+bottom}{top-bottom} \\ 0 & 0 & -\frac{2}{far-near} & -\frac{far+near}{far-near} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_x \\ c_y \\ c_z \\ 1 \end{bmatrix}$$

Transformations in OpenGL

- Projection
  - Option 2: Perspective projection




```
glFrustum(left, right, bottom, top, near, far);
```

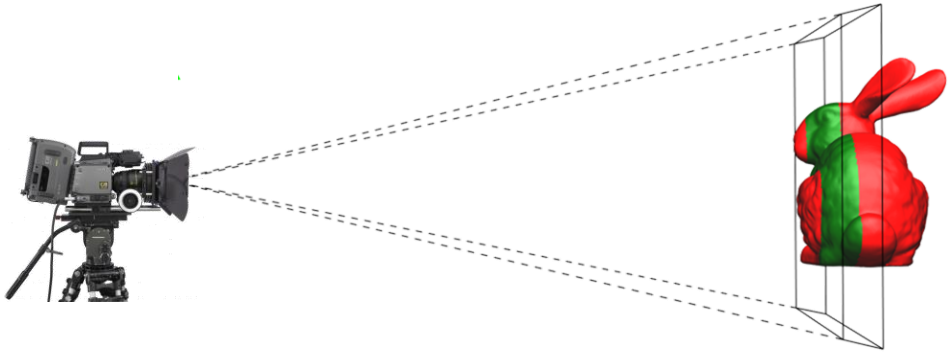
Transformations in OpenGL

- Projection
  - Option 2: Perspective projection

$$\begin{bmatrix} \frac{2\cdot near}{right-left} & 0 & \frac{right+left}{right-left} & 0 \\ 0 & \frac{2\cdot near}{top-bottom} & \frac{top+bottom}{top-bottom} & 0 \\ 0 & 0 & -\frac{far+near}{far-near} & -\frac{2\cdot far\cdot near}{far-near} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} c_x \\ c_y \\ c_z \\ 1 \end{bmatrix} = \begin{bmatrix} c'_x \\ c'_y \\ c'_z \\ -c_z \end{bmatrix}$$

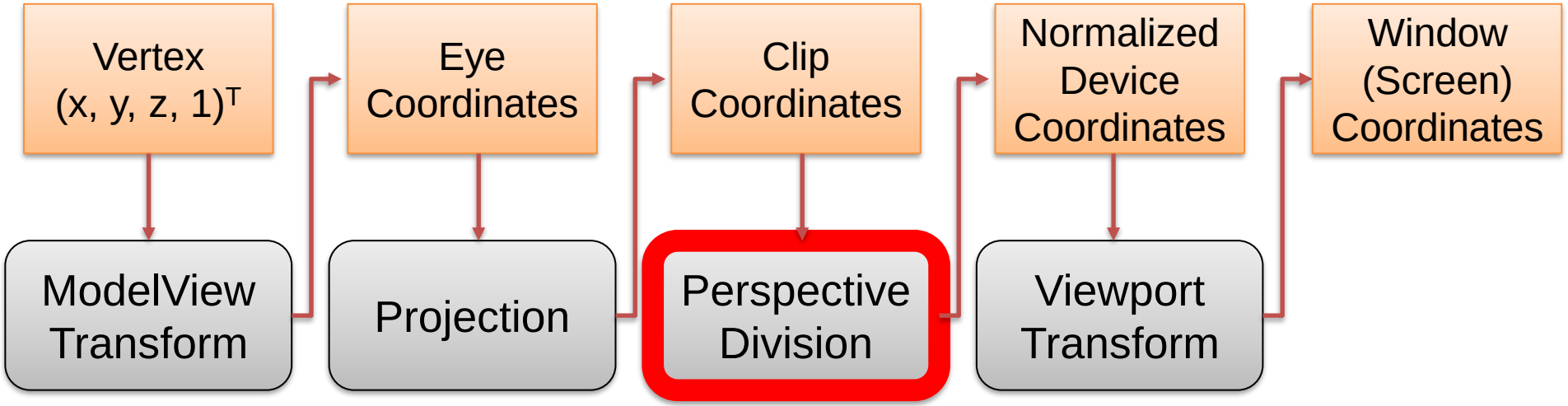
Transformations in OpenGL

- Projection
  - Clip the points $\mathbf{p} = (c'_x, c'_y, c'_z, 1)$ by comparing c'_x , c'_y and c'_z with 1



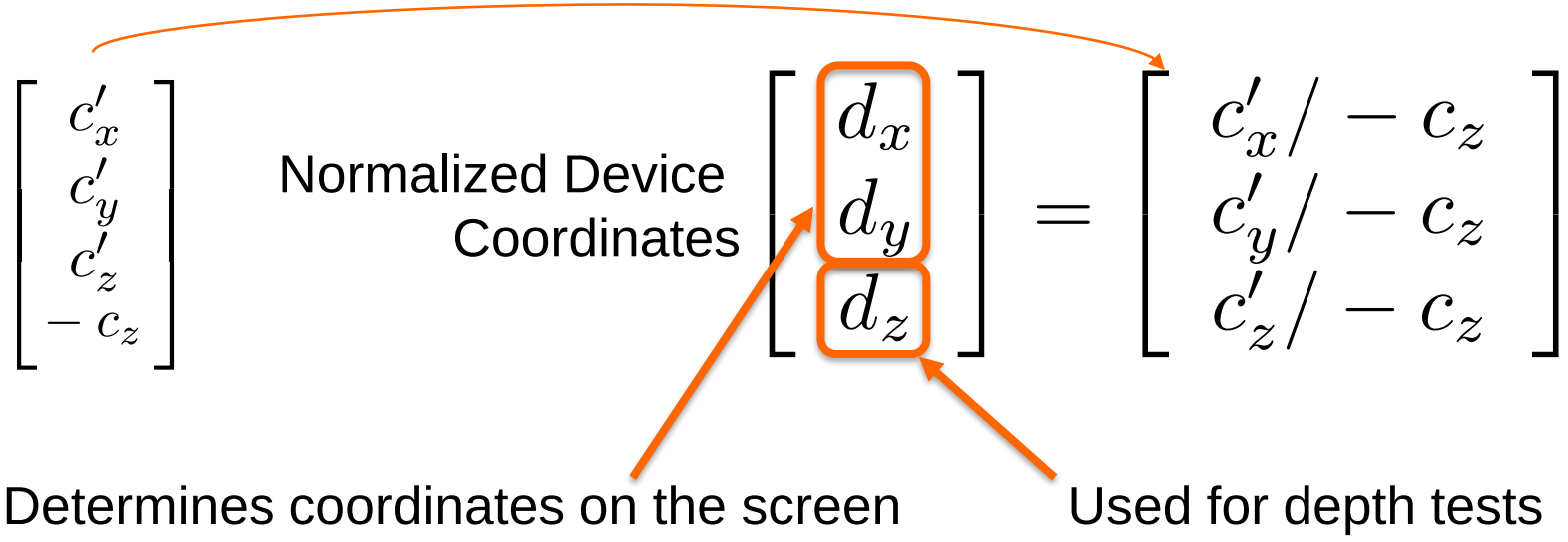
Transformations in OpenGL

- Stages of transformations



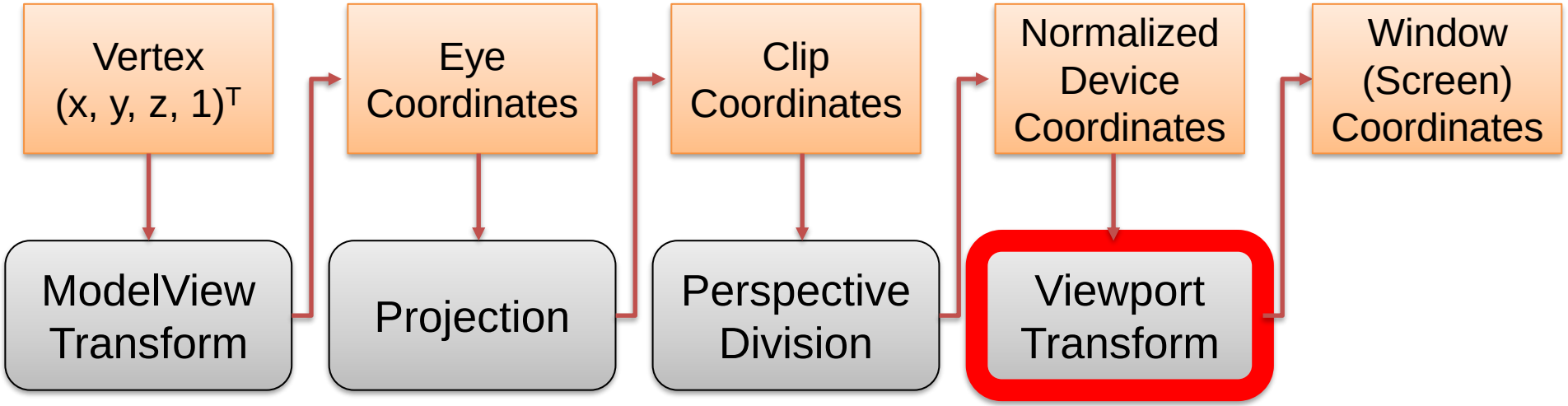
Transformations in OpenGL

- Perspective division



Transformations in OpenGL

- Stages of transformations

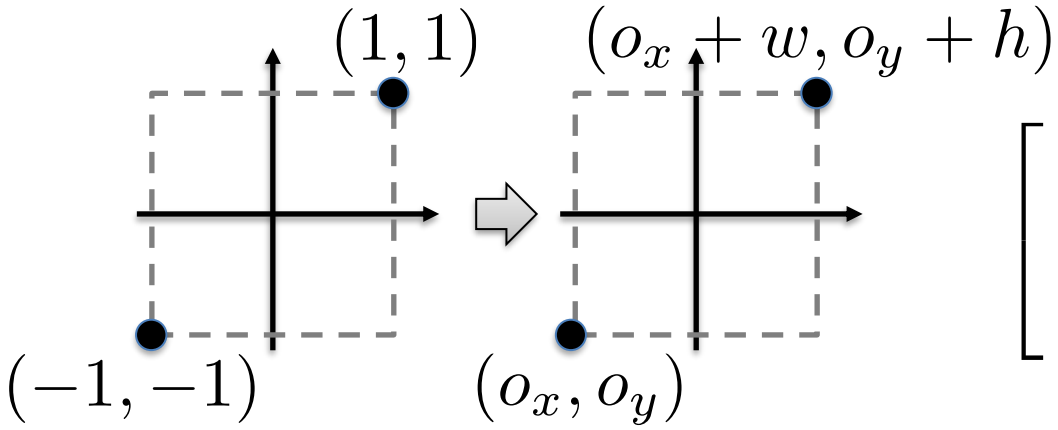


Transformations in OpenGL

- Viewport Transform

Normalized Device Coordinates

Screen Coordinates



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{w}{2} d_x + (o_x + \frac{w}{2}) \\ \frac{h}{2} d_y + (o_y + \frac{h}{2}) \\ \frac{f-n}{2} d_z + \frac{f+n}{2} \end{bmatrix}$$

```
glViewport(ox, oy, w, h);
glDepthRange(n, f);
```

END