3D Viewing

Hakan Bilen
University of Edinburgh

Computer Graphics
Fall 2017

Some slides are courtesy of Steve Marschner and Taku Komura

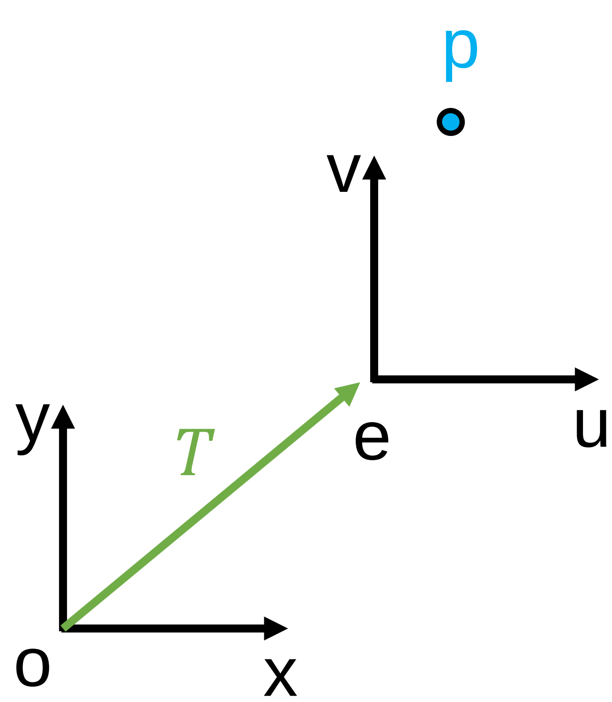
Slides

(Office hrs: Thursday 11.00-12.00 @Inf 1.41A)

WEBGL2

Review: Change of coordinates

Change of Coordinates



Which is bigger x_p or u_p?

$$T = \begin{pmatrix} 1 & 0 & e_x \\ 0 & 1 & e_y \\ 0 & 0 & 1 \end{pmatrix}$$

p in global coordinates (x,y)

$$p = \boldsymbol{o} + x_p \boldsymbol{x} + y_p \boldsymbol{y}$$

$$\begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \boxed{\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix}}$$

p in local coordinates (u,v)

$$p = \boldsymbol{e} + u_p \boldsymbol{u} + v_p \boldsymbol{v}$$

$$\begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} = \begin{pmatrix} u_x & v_x & e_x \\ u_y & v_y & e_y \\ 0 & 0 & 1 \end{pmatrix} \boxed{\begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix}}$$

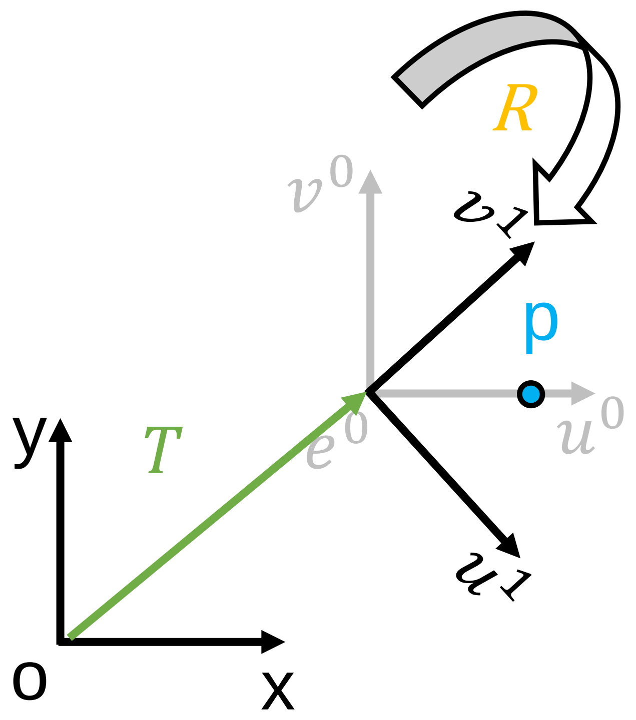
From local to global

$$\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = \begin{pmatrix} u_x & v_x & e_x \\ u_y & v_y & e_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix}$$

From local to global and from global to local

$$\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = T \begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} = T^{-1} \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix}$$

Change of Coordinates



p in different coordinate frames

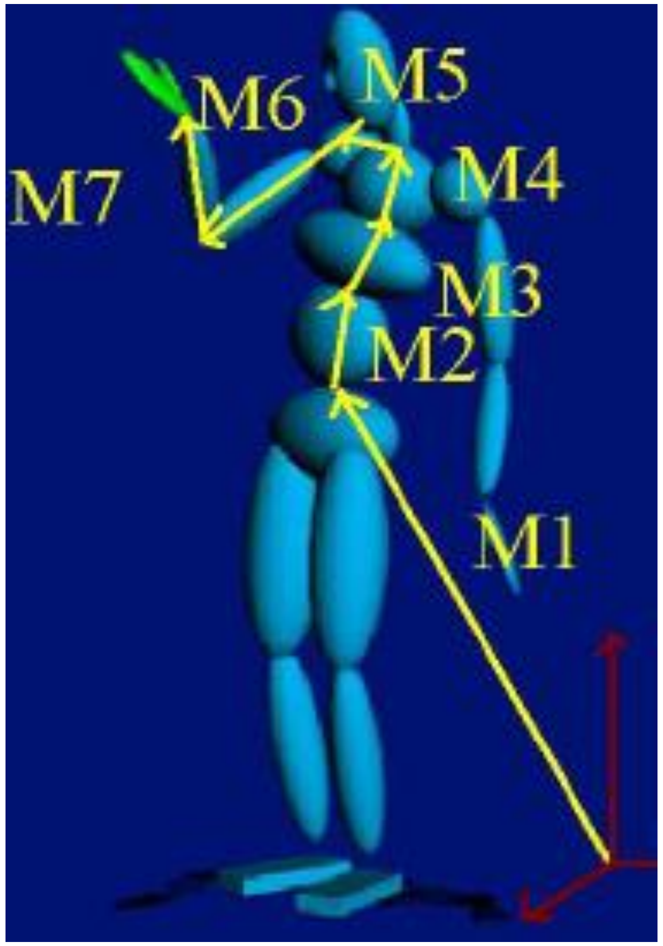
$$\begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = \begin{pmatrix} u_x^0 & v_x^0 & e_x^0 \\ u_y^0 & v_y^0 & e_y^0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_p^0 \\ v_p^0 \\ 1 \end{pmatrix} = \begin{pmatrix} u_x^1 & v_x^1 & e_x^1 \\ u_y^1 & v_y^1 & e_y^1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_p^1 \\ v_p^1 \\ 1 \end{pmatrix}$$

T from $\begin{pmatrix} u_p^0 \\ v_p^0 \\ 1 \end{pmatrix}$ to $\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix}$ and R from $\begin{pmatrix} u_p^0 \\ v_p^0 \\ 1 \end{pmatrix}$ to $\begin{pmatrix} u_p^1 \\ v_p^1 \\ 1 \end{pmatrix}$

$$\begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = T \begin{pmatrix} u_p^0 \\ v_p^0 \\ 1 \end{pmatrix} = TR \begin{pmatrix} u_p^1 \\ v_p^1 \\ 1 \end{pmatrix}$$

Change of Coordinates

Robot example (assignment)



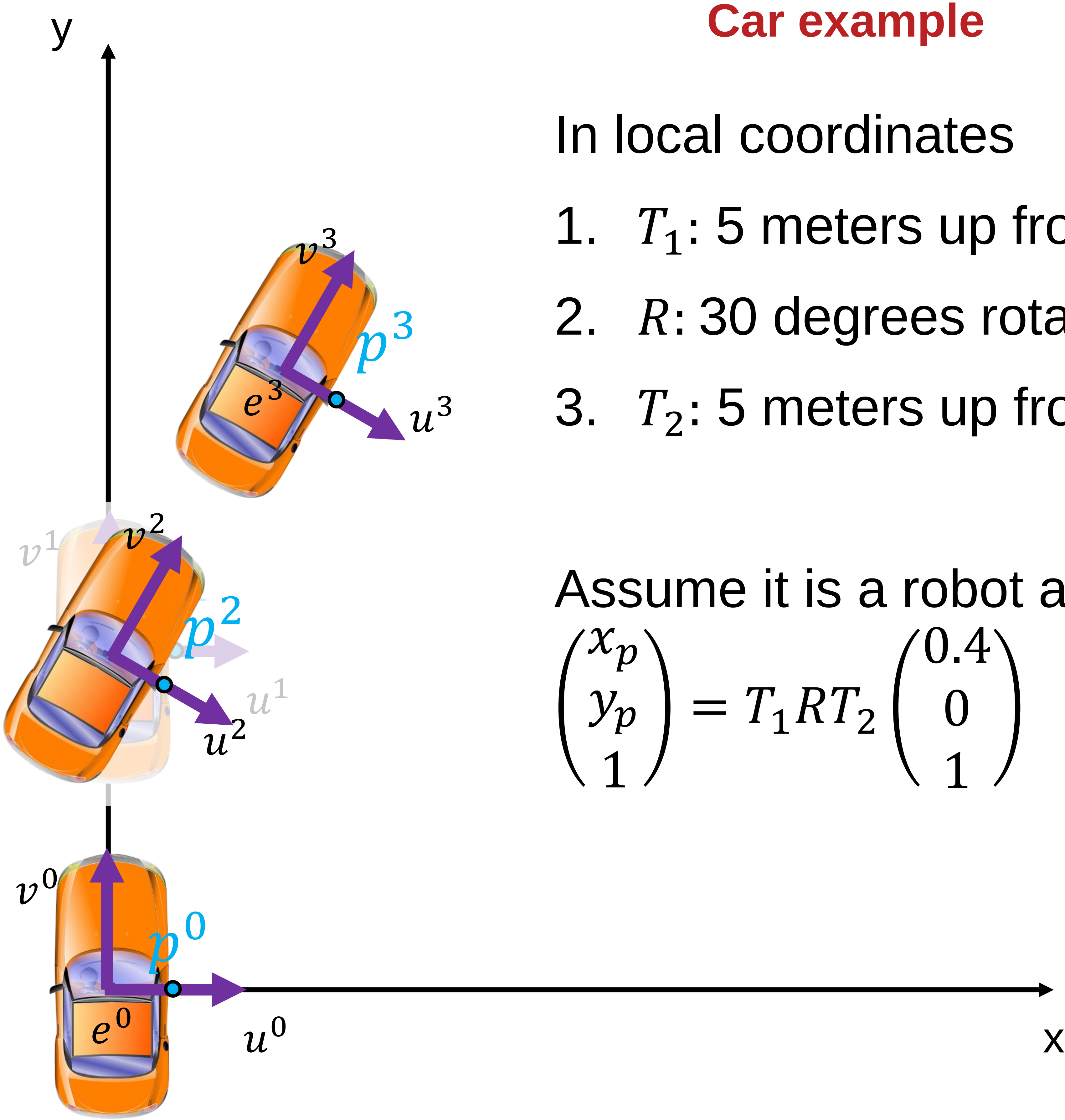
$$\boldsymbol{y} = \boldsymbol{M}^1 \boldsymbol{M}^2 \ldots \boldsymbol{M}^7 \boldsymbol{x}$$

$\boldsymbol{x}$ is in local coordinates

$\boldsymbol{y}$ is in world (global) coordinates

Change of Coordinates

Car example



In local coordinates

1. T_1: 5 meters up from (u^0, v^0) to (u^1, v^1)
2. R: 30 degrees rotate from (u^1, v^1) to (u^2, v^2)
3. T_2: 5 meters up from (u^2, v^2) to (u^3, v^3)

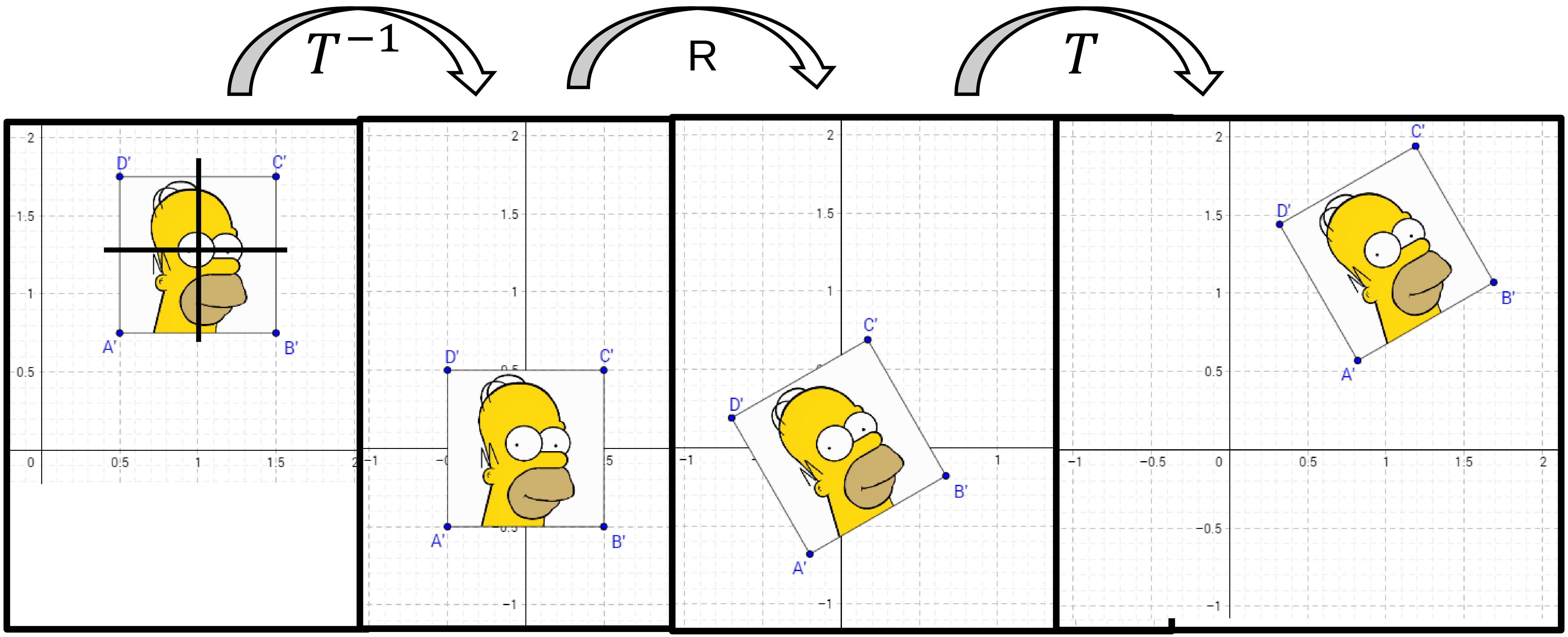
Assume it is a robot arm:

$$\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = T_1 R T_2 \begin{pmatrix} 0.4 \\ 0 \\ 1 \end{pmatrix}$$

Rotate about a particular point

The mysterious connection with change of coordinates

$$Mp = TRT^{-1}p$$

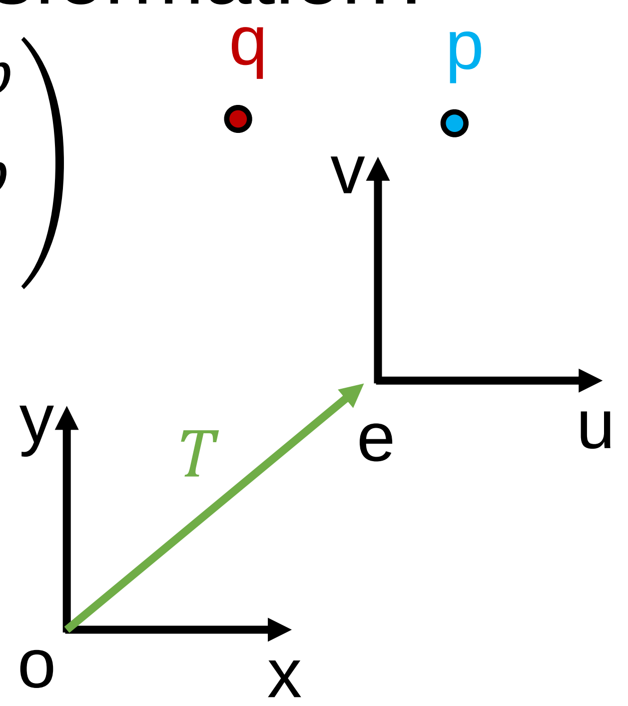


Rotate about a particular point

The mysterious connection with change of coordinates

How the heck is it different from local to global transformation?

$$(1)\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = T\begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} \qquad (2)\begin{pmatrix} u_q \\ v_q \\ 1 \end{pmatrix} = R\begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix}$$



Then the solution should be

$$\begin{pmatrix} x_q \\ y_q \\ 1 \end{pmatrix} = TR\begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix}$$

We don't have the local coordinates $\begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix}$ but global ones $\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix}$

and $\begin{pmatrix} x_q \\ y_q \\ 1 \end{pmatrix} = TRT^{-1}\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix}$

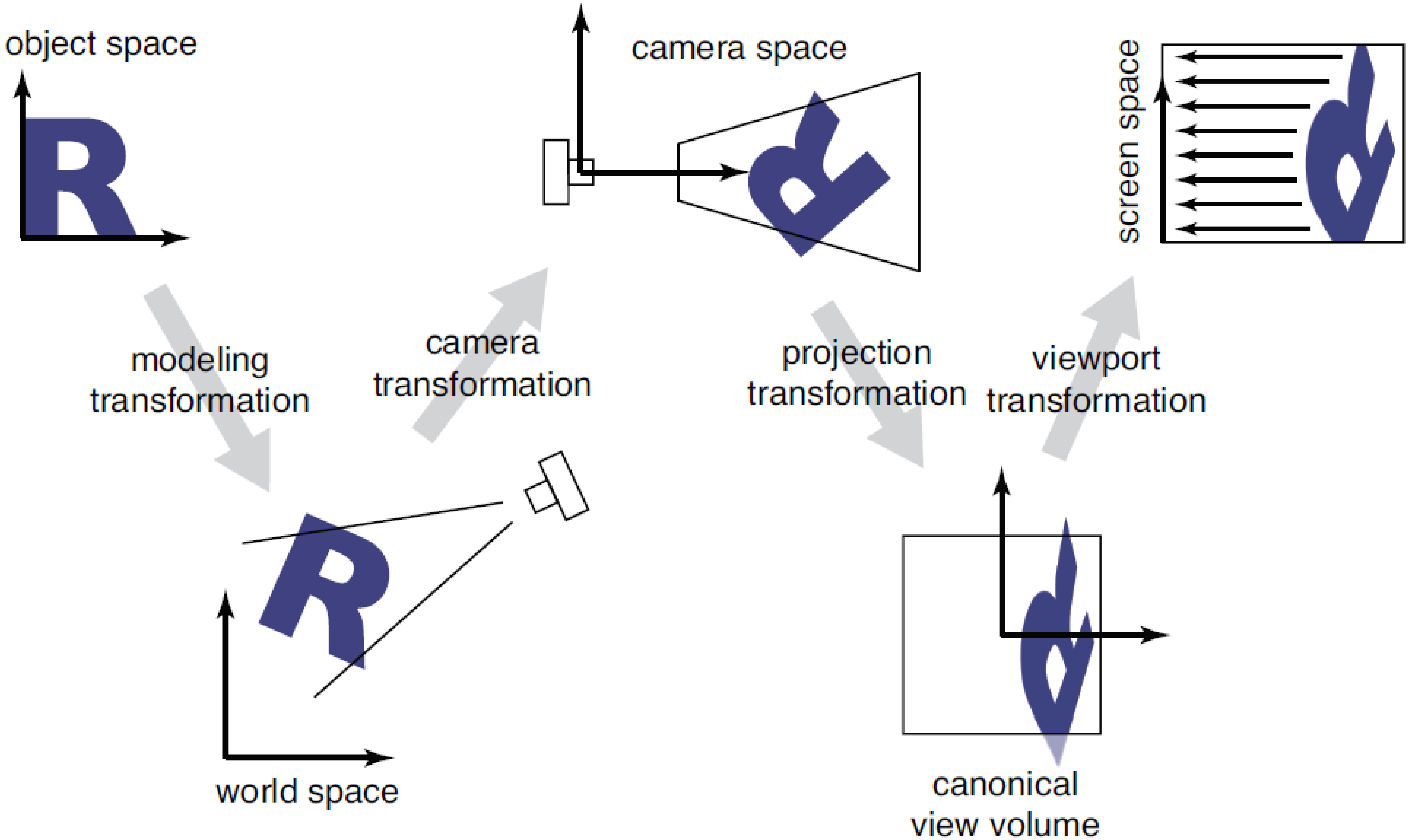
Today

View transformation 3D→2D

- Camera transformation
- Projective transformation
- Viewport transformation

Pipeline of transformations

Sequence of transformations

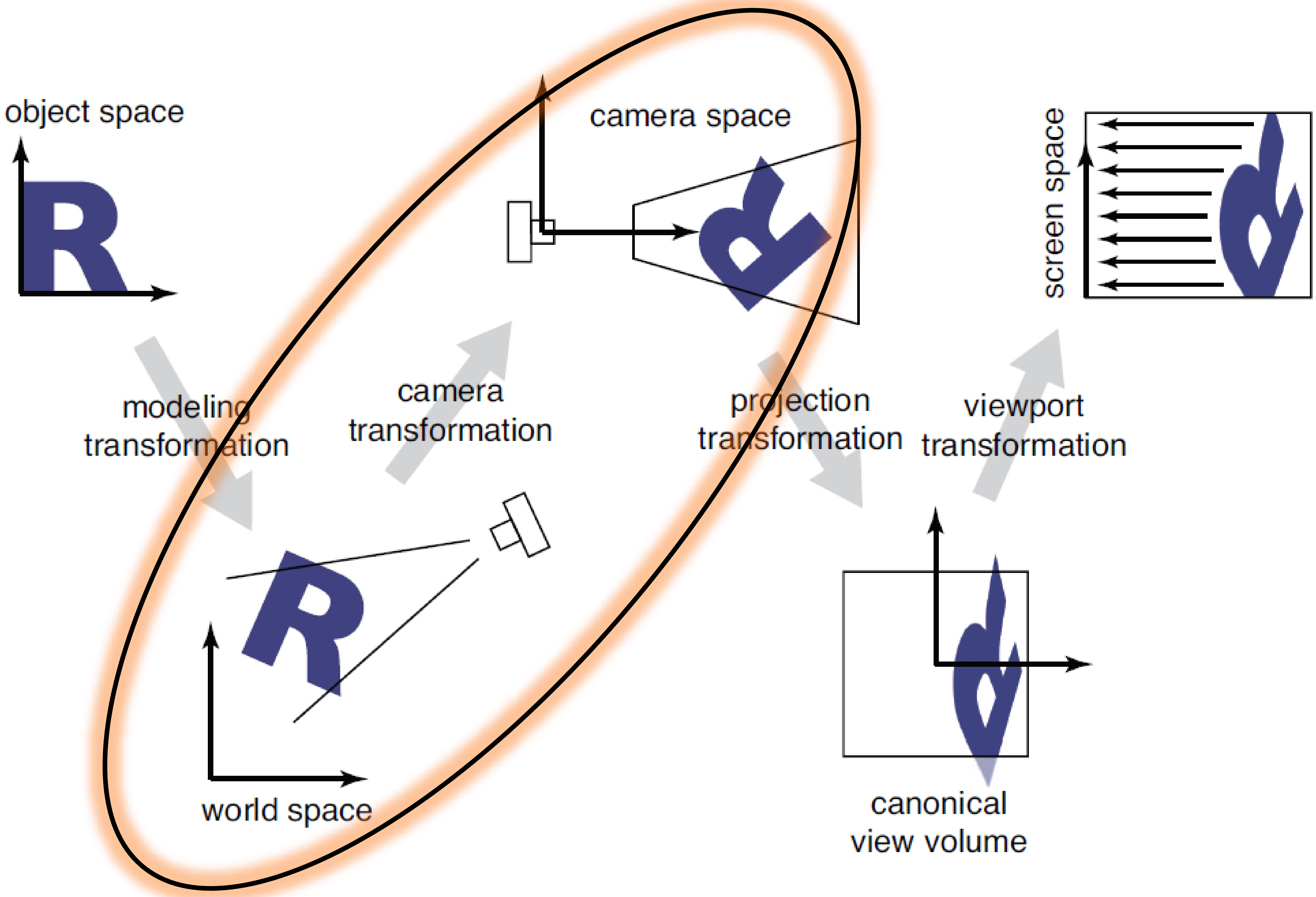


Which transformations are from continuous to continuous?

Slide credits: S. Marschner

Pipeline of transformations

Sequence of transformations

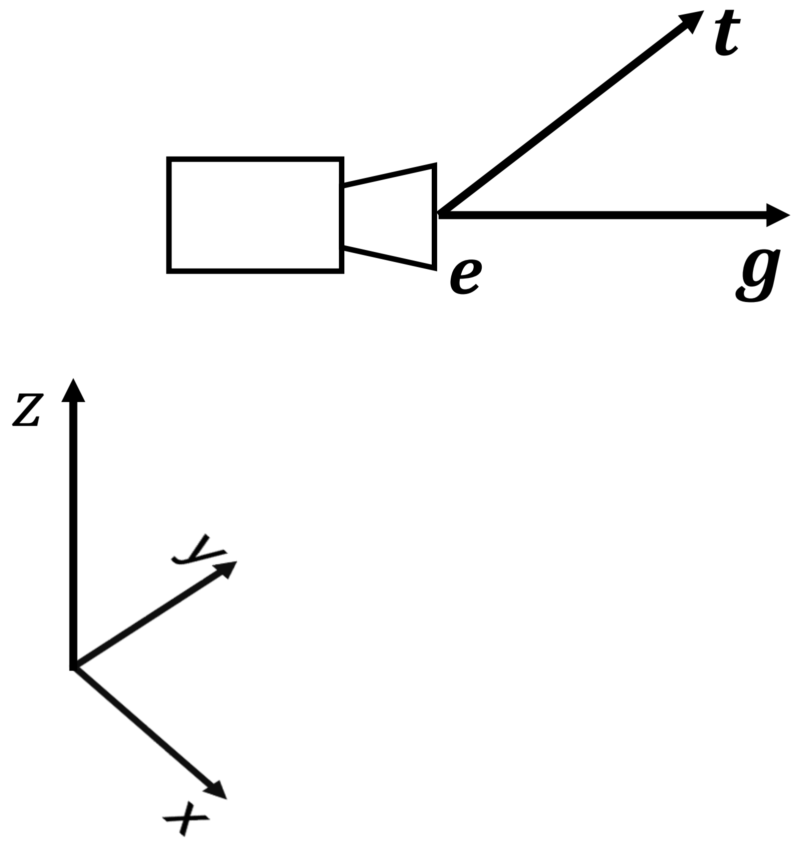


Slide credits: S. Marschner

Camera (Eye) Transformation

World space to camera space

- Depends on pose of camera
  - eye position ($\boldsymbol{e}$)
  - gaze direction ($\mathbf{g}$)
  - view-up vector ($\boldsymbol{t}$)

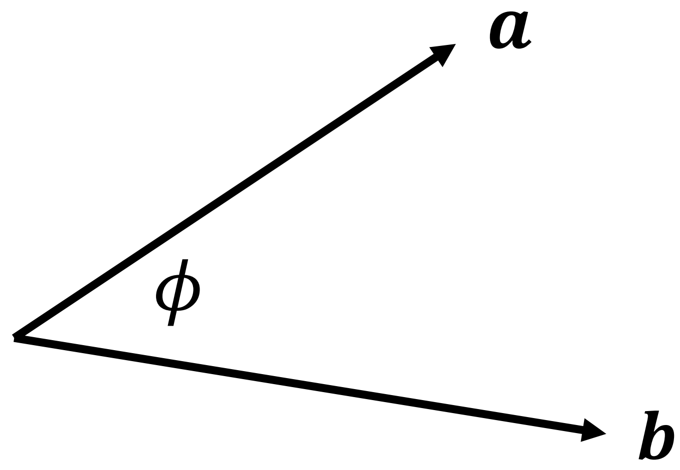


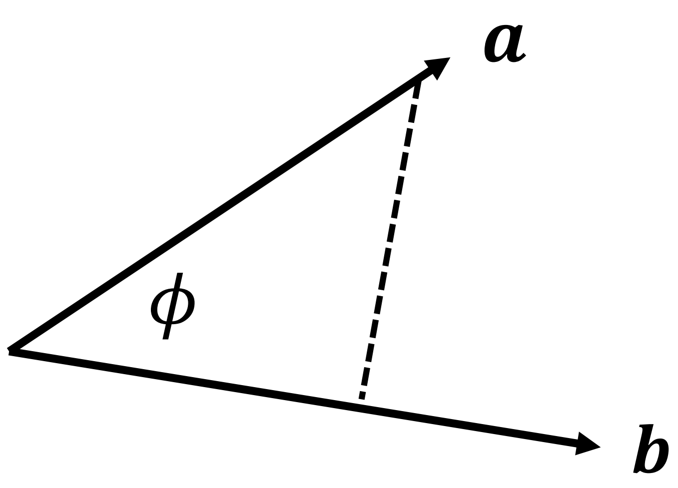
Review: Linear Algebra

Dot product

- $\boldsymbol{a} \cdot \boldsymbol{b} = \left|\left|\boldsymbol{a}\right|\right|\left|\left|\boldsymbol{b}\right|\right| cos\phi$
- $\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{x_a x_b + y_a y_b}$
- Projection a onto b

$$\boldsymbol{a} \rightarrow \boldsymbol{b} = \left|\left|\boldsymbol{a}\right|\right| cos\phi = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{||\boldsymbol{b}||}$$

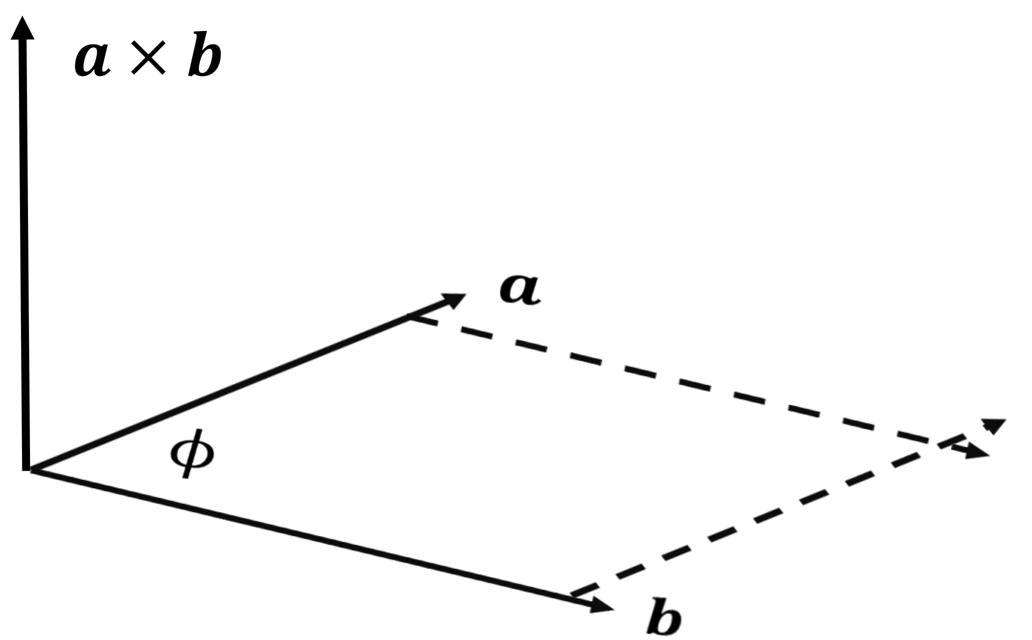




- $\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{b} \cdot \boldsymbol{a}$, $\boldsymbol{a} \cdot (\boldsymbol{b} + \boldsymbol{c}) = \boldsymbol{a} \cdot \boldsymbol{b} + \boldsymbol{a} \cdot \boldsymbol{c}$, $k\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a} \cdot k\boldsymbol{b}$

Cross product

- $||\boldsymbol{a} \times \boldsymbol{b}|| = \left|\left|\boldsymbol{a}\right|\right|\left|\left|\boldsymbol{b}\right|\right| \boldsymbol{sin\phi}$
- $\boldsymbol{x \times y = z, y \times x = -z}$
- $\boldsymbol{y \times z = x, z \times y = -x}$
- $\boldsymbol{x \times z = -y, z \times x = -y}$
- $\boldsymbol{a} \times \boldsymbol{b} = \begin{pmatrix} \boldsymbol{y_a z_b - z_a y_b} \\ \boldsymbol{z_a x_b - x_a z_b} \\ \boldsymbol{x_a y_b - y_a x_b} \end{pmatrix}$



Construct a new coordinate system in 3D

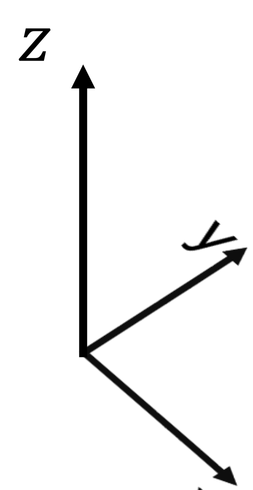
Orthonormal basis

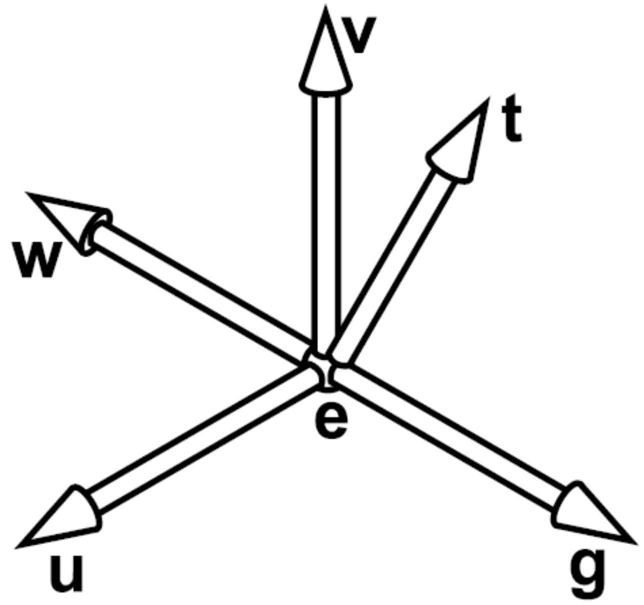
1. Give me two vectors (x, y) which are not collinear
2. Normalize $\boldsymbol{x}$ such that $\|\boldsymbol{x}\| = 1$, $\quad \widetilde{\boldsymbol{x}} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$
3. If $\widetilde{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{y}} \neq 0, \boldsymbol{w} = \widetilde{\boldsymbol{x}}\,, \boldsymbol{u} = \frac{\boldsymbol{w} \times \widetilde{\boldsymbol{y}}}{\|\boldsymbol{w} \times \widetilde{\boldsymbol{y}}\|}$, otherwise, $\boldsymbol{w} = \widetilde{\boldsymbol{x}}\,, \boldsymbol{u} = \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|}$
4. $\mathbf{v} = \boldsymbol{w} \times \boldsymbol{u}$

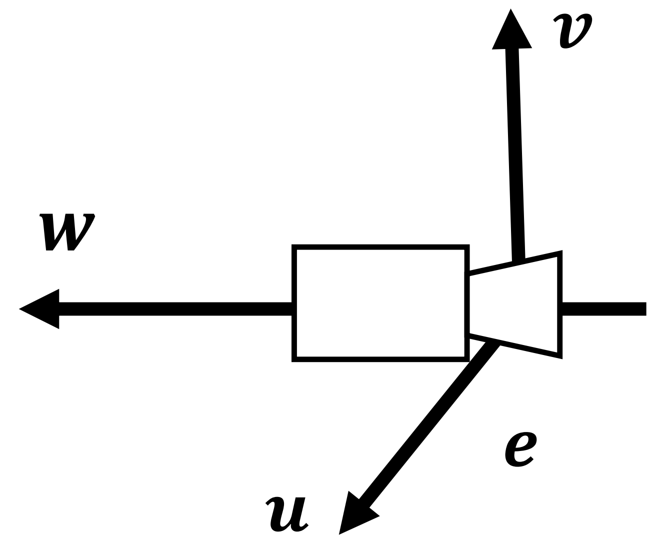
Construct camera coordinates

Given $\boldsymbol{e}, \boldsymbol{g}, \boldsymbol{t}$

- $\boldsymbol{w} = -\frac{\boldsymbol{g}}{||\boldsymbol{g}||}$
- $\boldsymbol{u} = -\frac{\boldsymbol{t} \times \boldsymbol{w}}{||\boldsymbol{t} \times \boldsymbol{w}||}$
- $\boldsymbol{v} = \boldsymbol{w} \times \boldsymbol{u}$



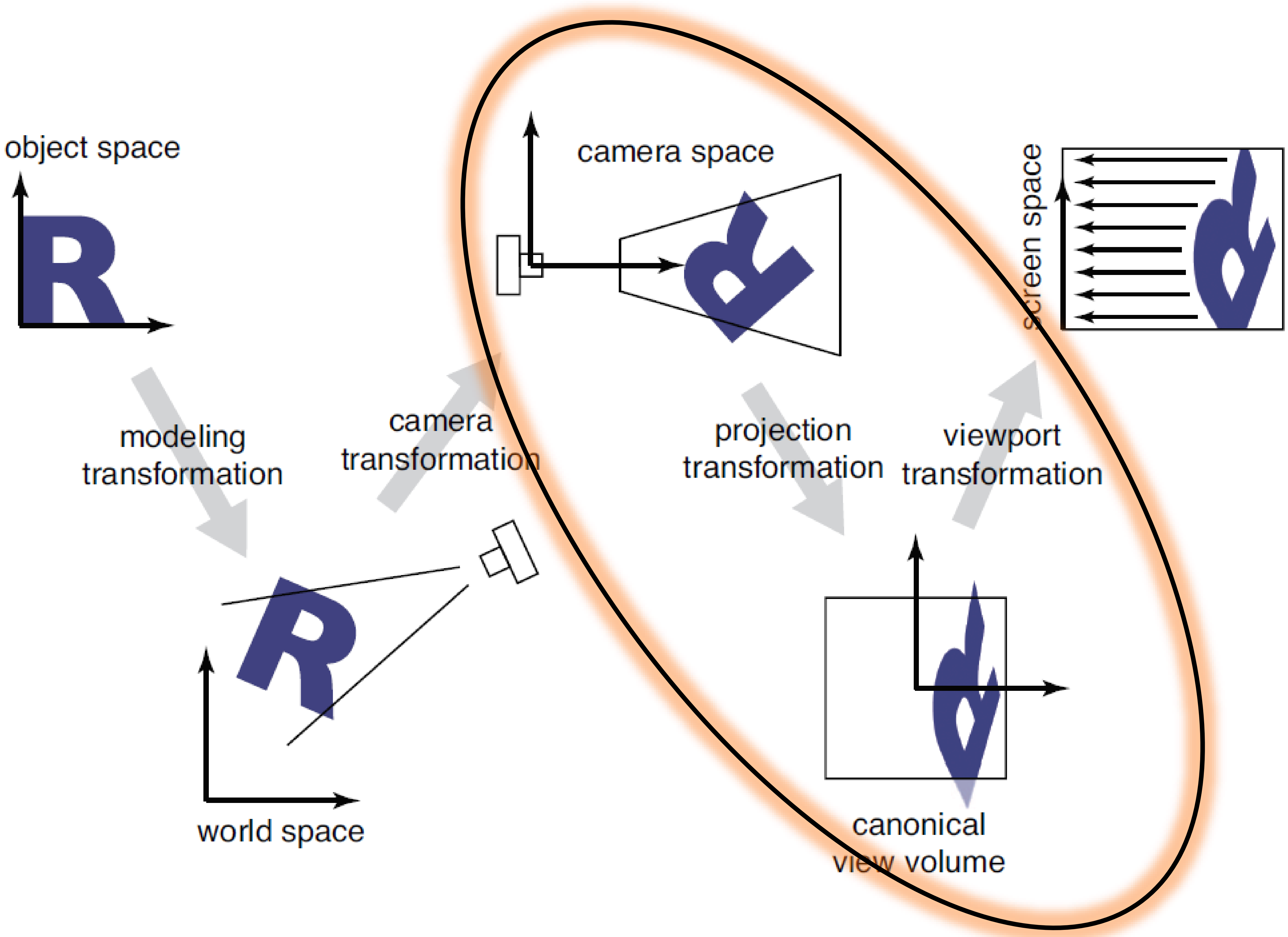




From world to camera

$$\boldsymbol{M_{cam}} = \begin{pmatrix} \boldsymbol{u} & \boldsymbol{v} & \boldsymbol{w} & \boldsymbol{e} \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} x_u & y_u & z_u & 0 \\ x_v & y_v & z_v & 0 \\ x_w & y_w & z_w & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -x_e \\ 0 & 1 & 0 & -y_e \\ 0 & 0 & 1 & -z_e \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Pipeline of transformations



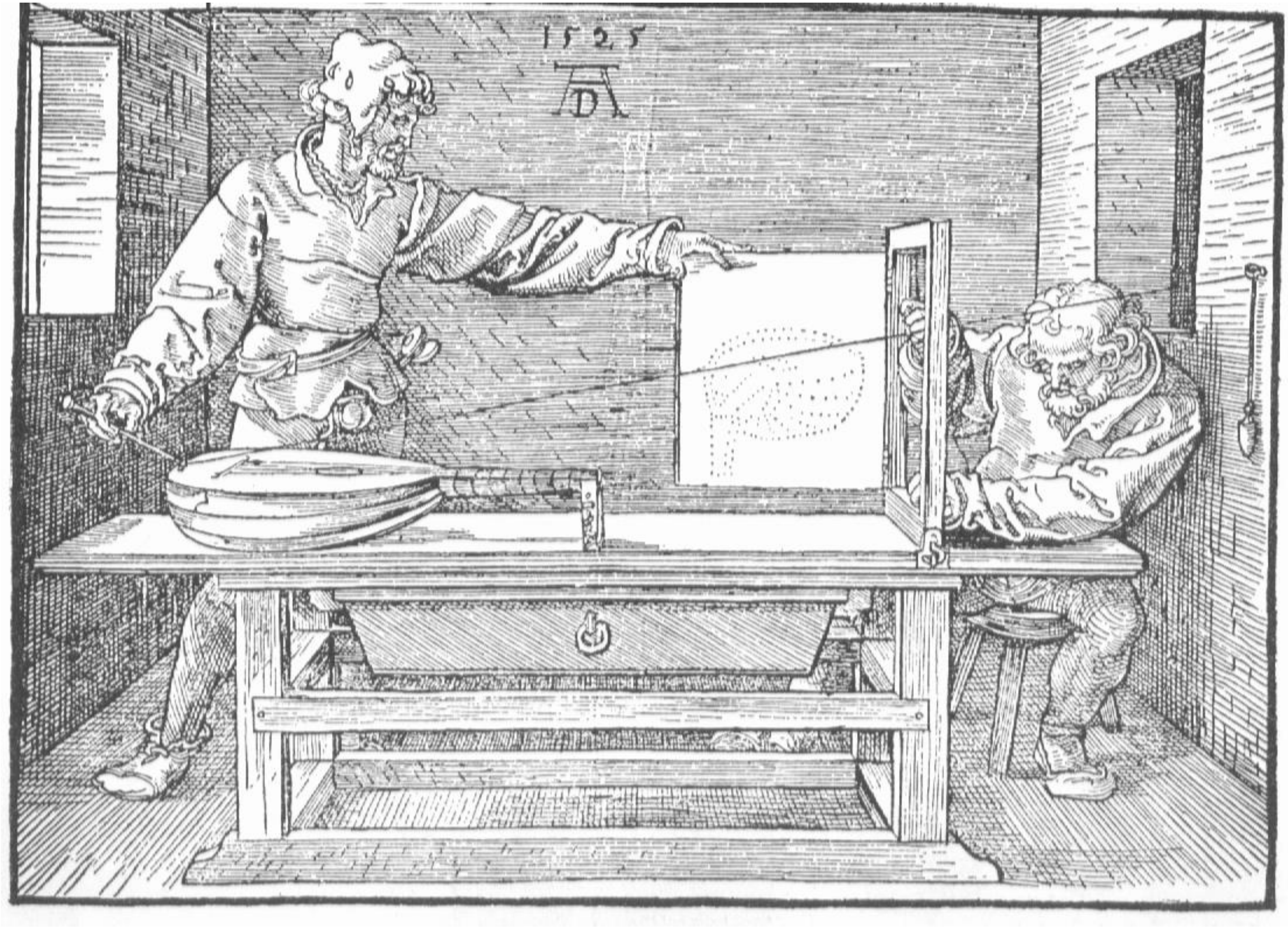
Slide credits: S. Marschner

Perspective Projection

- Objects far away appear smaller, closer objects appear bigger

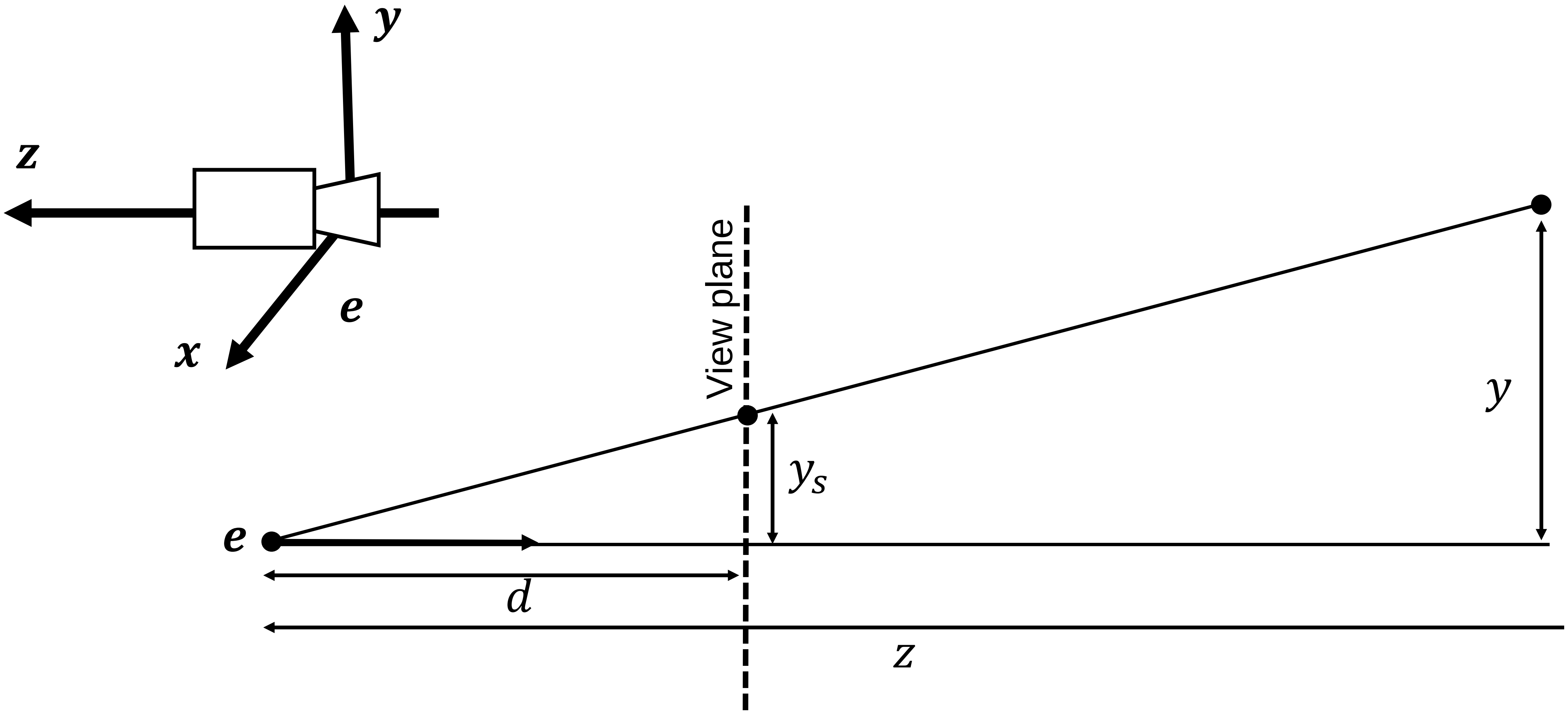
- Specified by
  - center of projection
  - focal distance (distance from the eye to the projection plane)

Perspective Projection



Der Zeichner der Laute, Dürer

Perspective Projection



similar triangles:

$$\frac{y_s}{d} = \frac{y_s}{-z} \quad \rightarrow \quad y_s = -\frac{dy}{z}$$

How to encode perspective?

$$x_s = -\frac{dx}{z}, \qquad y_s = -\frac{dy}{z}$$

$$\begin{pmatrix} x_s \\ y_s \\ z_s \\ 1 \end{pmatrix} = \begin{pmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Homogeneous Coordinates

Revisited

- Introduced to combine linear and translation part (in Lecture 5)

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \to \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

- True purpose of homogenous coordinates is projection
- Perspective projection requires division

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \sim \begin{pmatrix} wx \\ wy \\ wz \\ w \end{pmatrix}$$

- If $w > 0$, divide by w to convert into Cartesian coordinates
- If $w = 0$, it is a point at infinity

Equivalence of homogenous coordinates

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \sim \begin{pmatrix} wx \\ wy \\ wz \\ w \end{pmatrix}$$

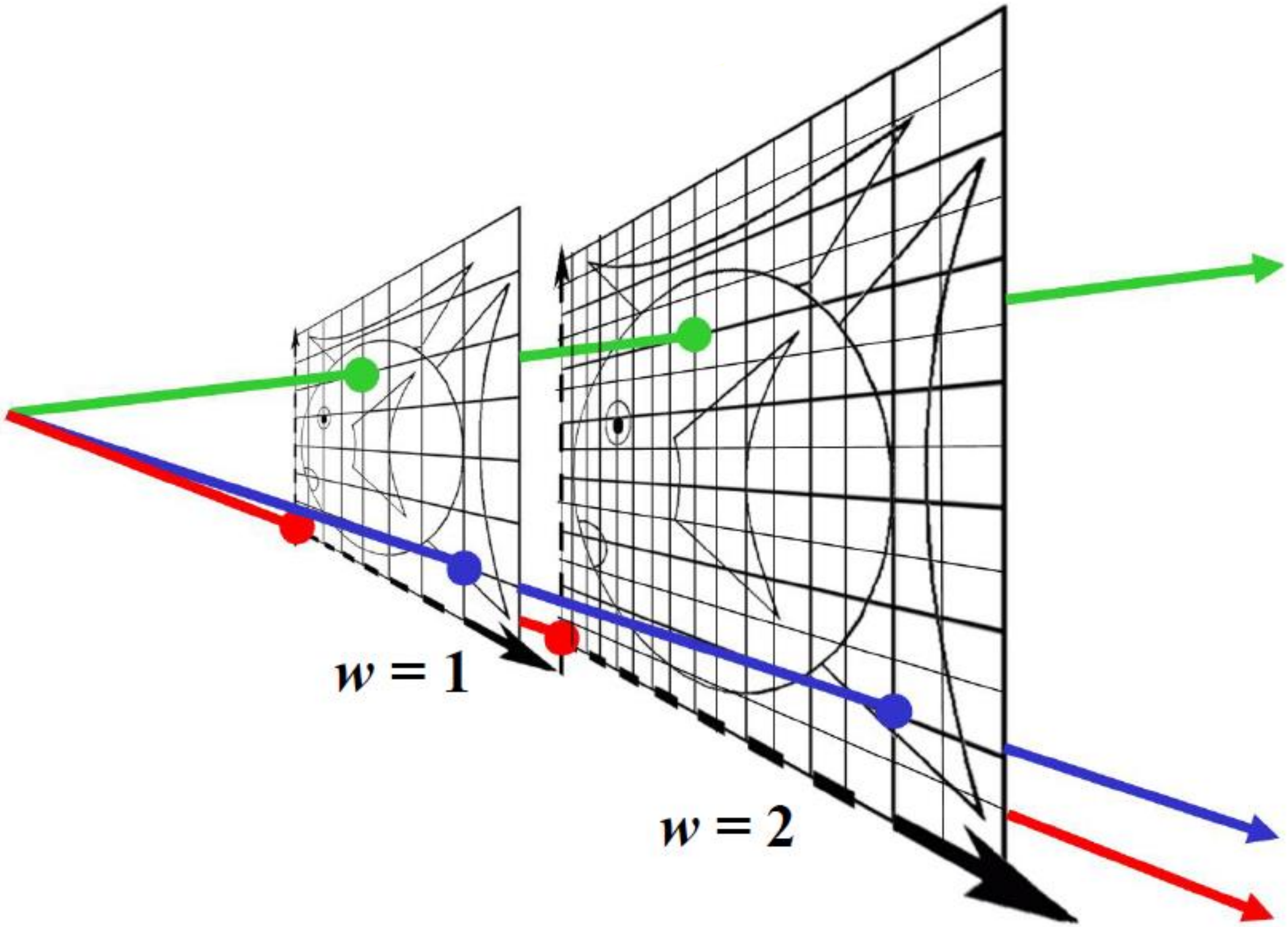
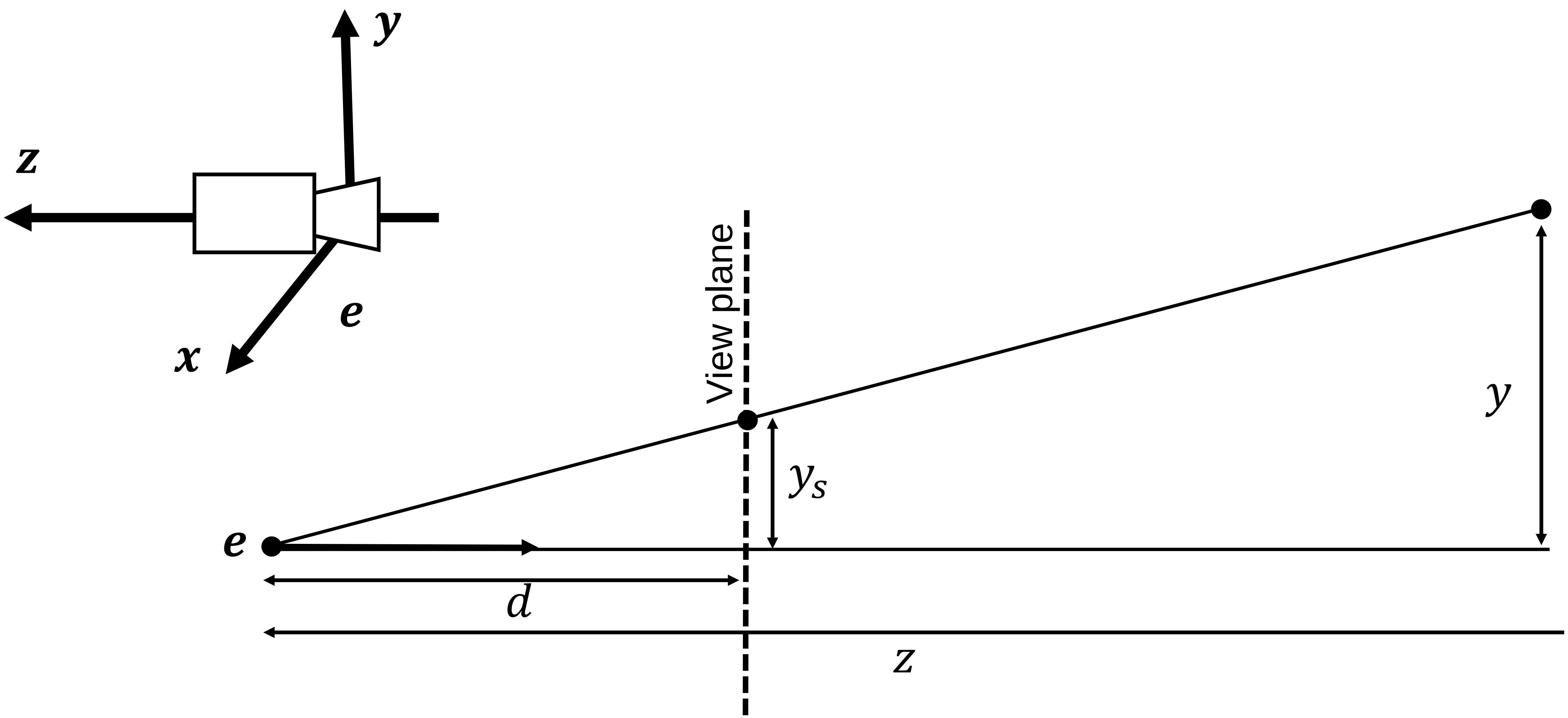


Image credit: W. Matusik

Perspective Projection



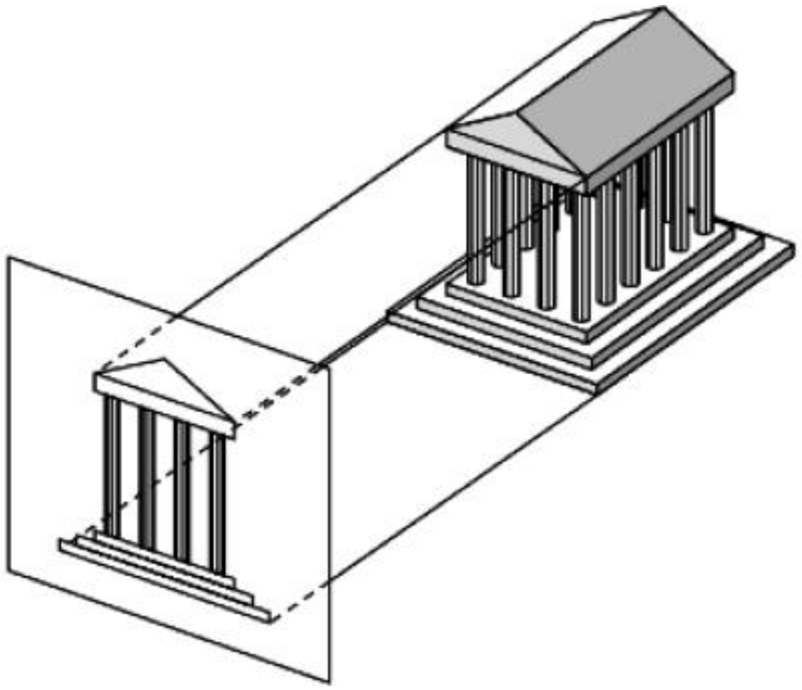
Move z to w:

$$\begin{pmatrix} x_s \\ y_s \\ 1 \end{pmatrix} = \begin{pmatrix} -dx/z \\ -dy/z \\ 1 \end{pmatrix} \sim \begin{pmatrix} dx \\ dy \\ -z \end{pmatrix} = \begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Parallel Projection

- Focal length (d) is infinite

$x_s = x,\ y_s = x$



- Rays are parallel and orthogonal to image
- Toss out z

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Bounding view volume

In practice, we are interested to visualize the objects

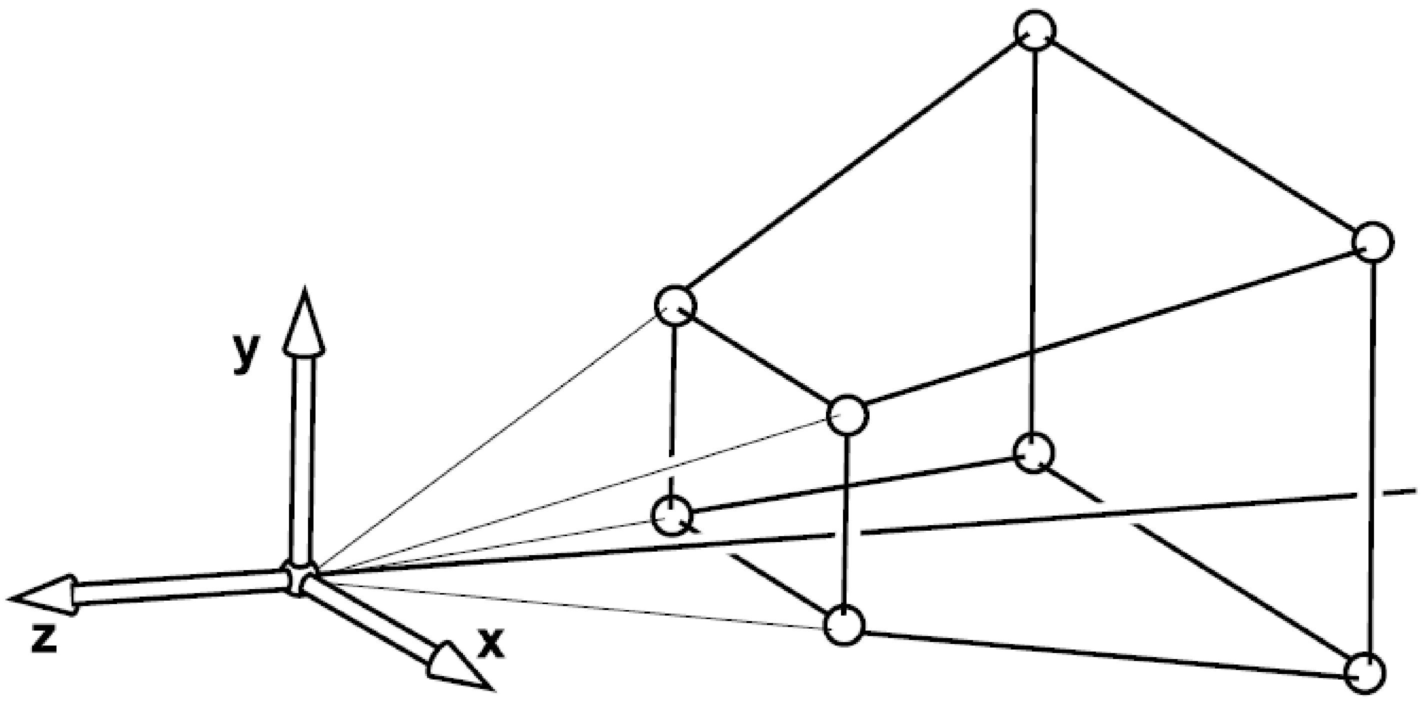
- in front of the camera
- in a bounded volume

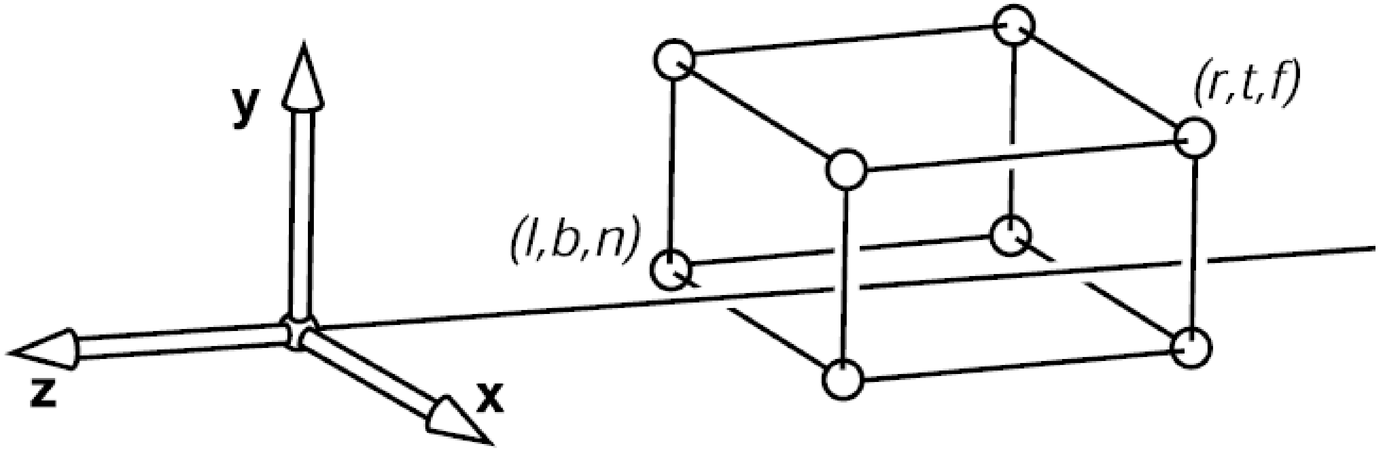
  - not too close, far

Volume shapes

- A box for parallel projection
- a frustum (truncated pyramid) for perspective projection

Clip surfaces outside the view volume are clipped





Bounding view volume

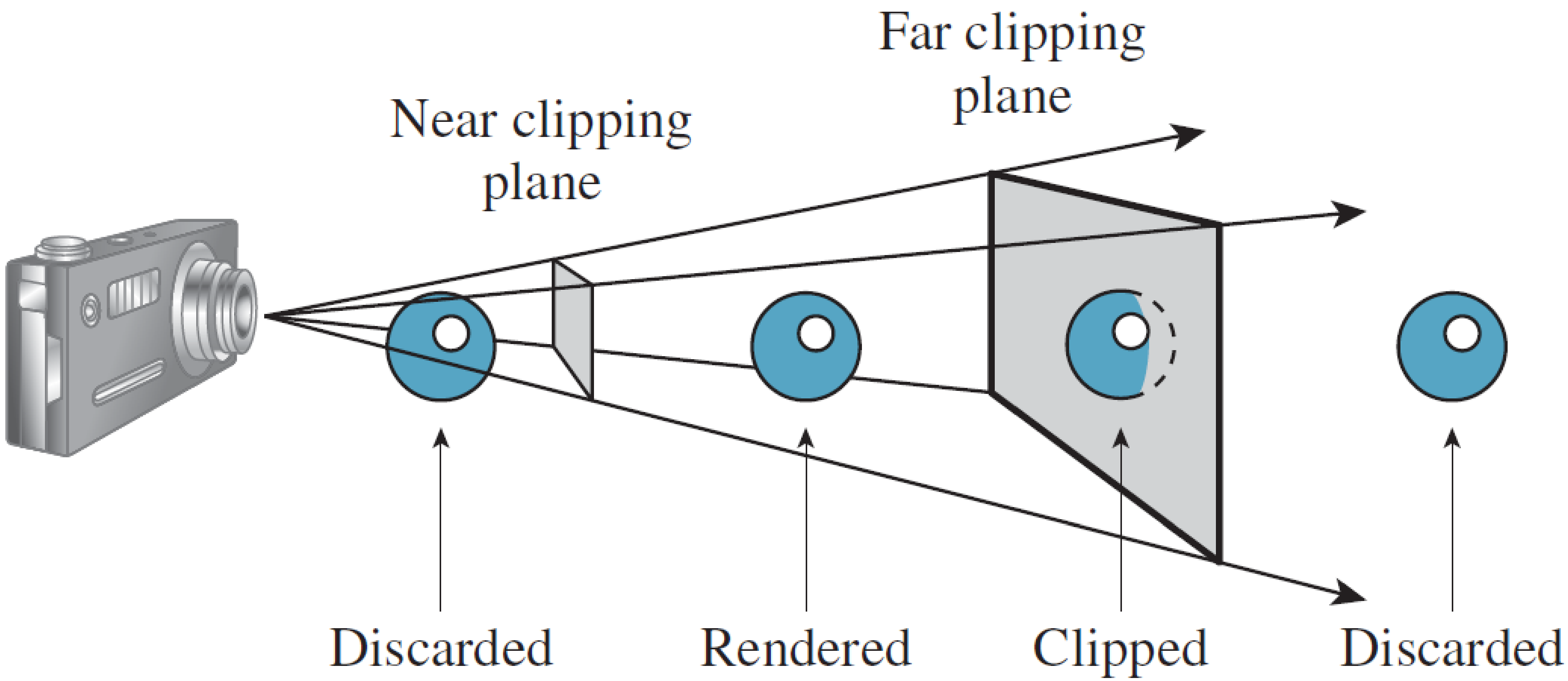
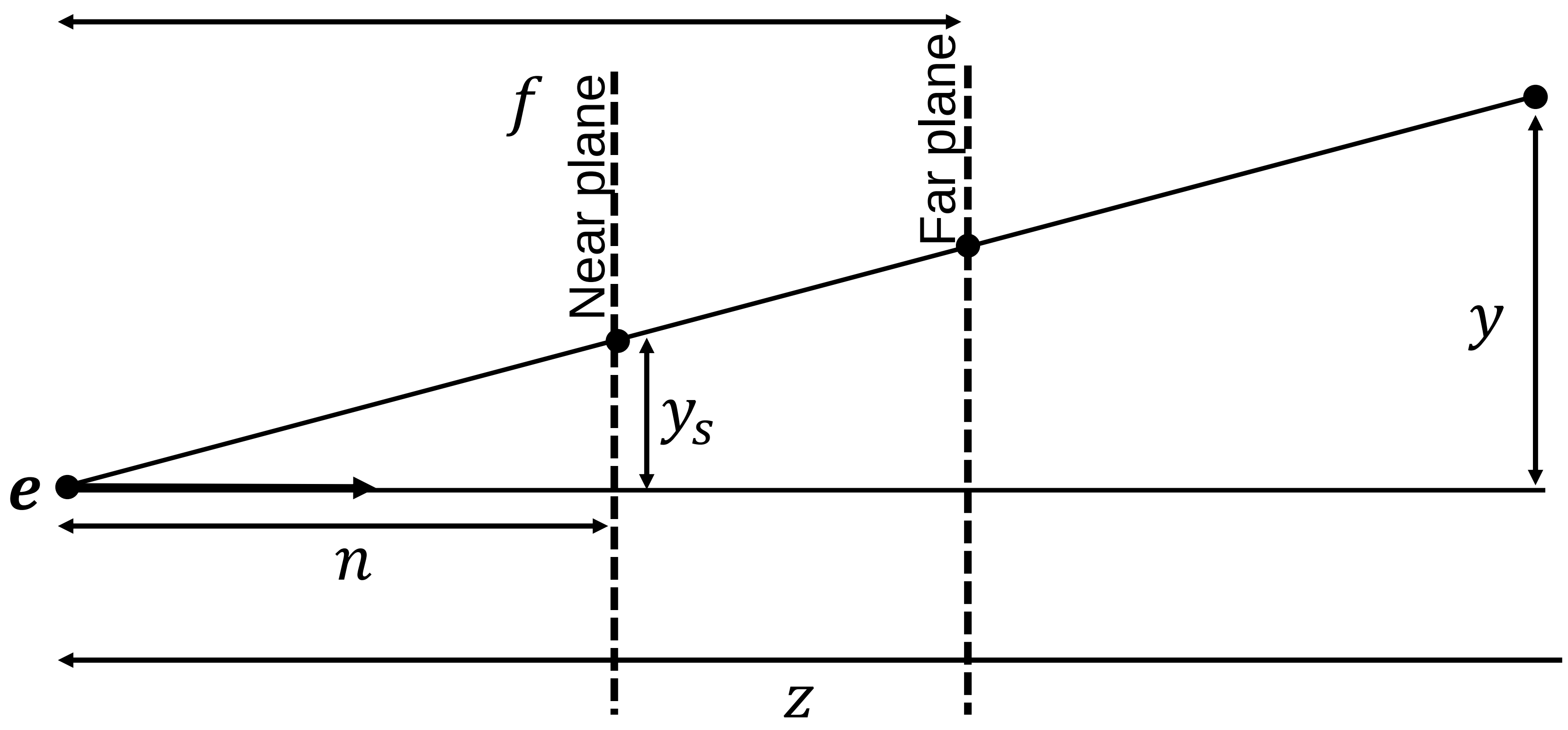


Image credit: Hughes

Bounded View Volume



- Use near plane distances as the projection distance ($d = -n$)
- Scale by -1 to have fewer minus sign

$$\begin{pmatrix} x_s \\ y_s \\ 1 \end{pmatrix} = \begin{pmatrix} nx/z \\ ny/z \\ 1 \end{pmatrix} \sim \begin{pmatrix} nx \\ ny \\ -z \end{pmatrix} = \begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Problem: Keep depth info for the later hidden surface elimination

Canonical View Volume

- Preserve depth on near and far planes in z_s

$$\begin{pmatrix} x_s \\ y_s \\ z_s \\ 1 \end{pmatrix} \sim \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ z \end{pmatrix} = \begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

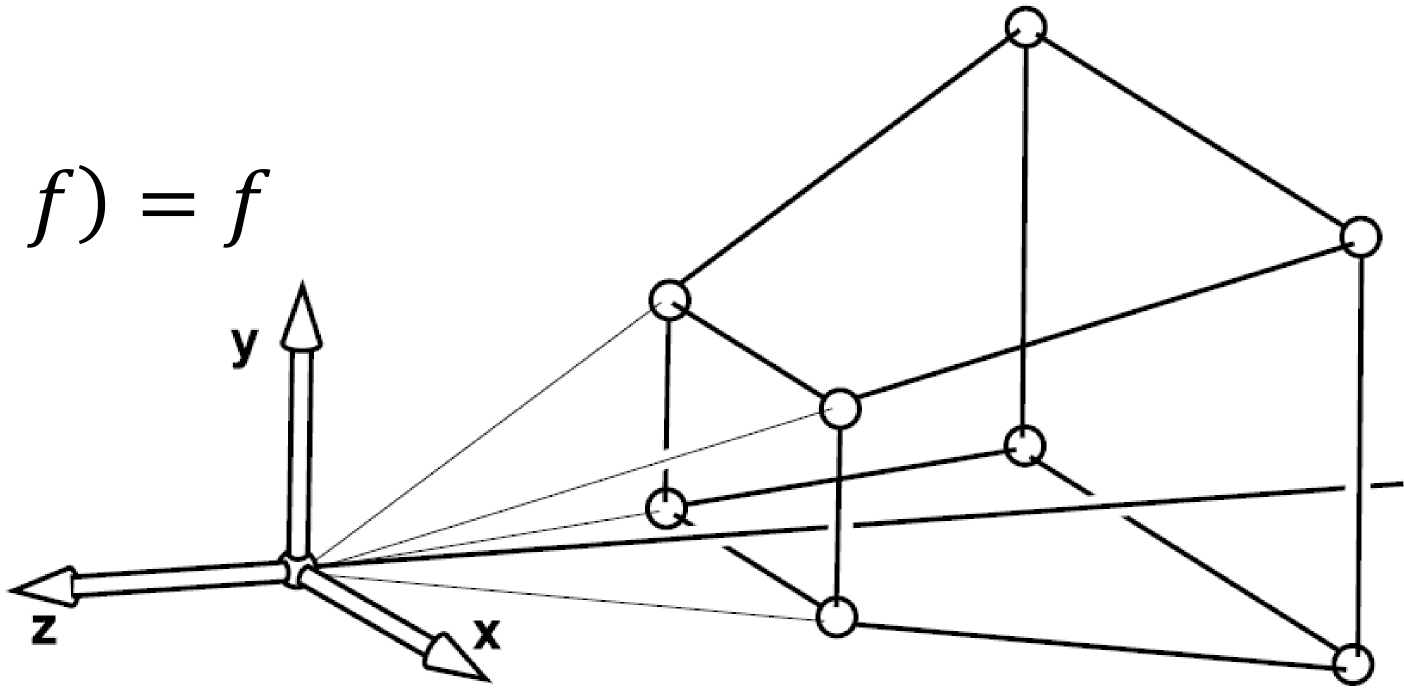
- choose a and b so that $z_s(z = n) = n$ and $z_s(z = f) = f$.

$\tilde{z}(z) = az + b$

$z_s(z) = \dfrac{\tilde{z}}{z} = \dfrac{az + b}{z}$

want $z_s(z = n) = n$ and $z_s(z = f) = f$

$a = (n + f)$ and b $= -nf$



Perspective Matrix

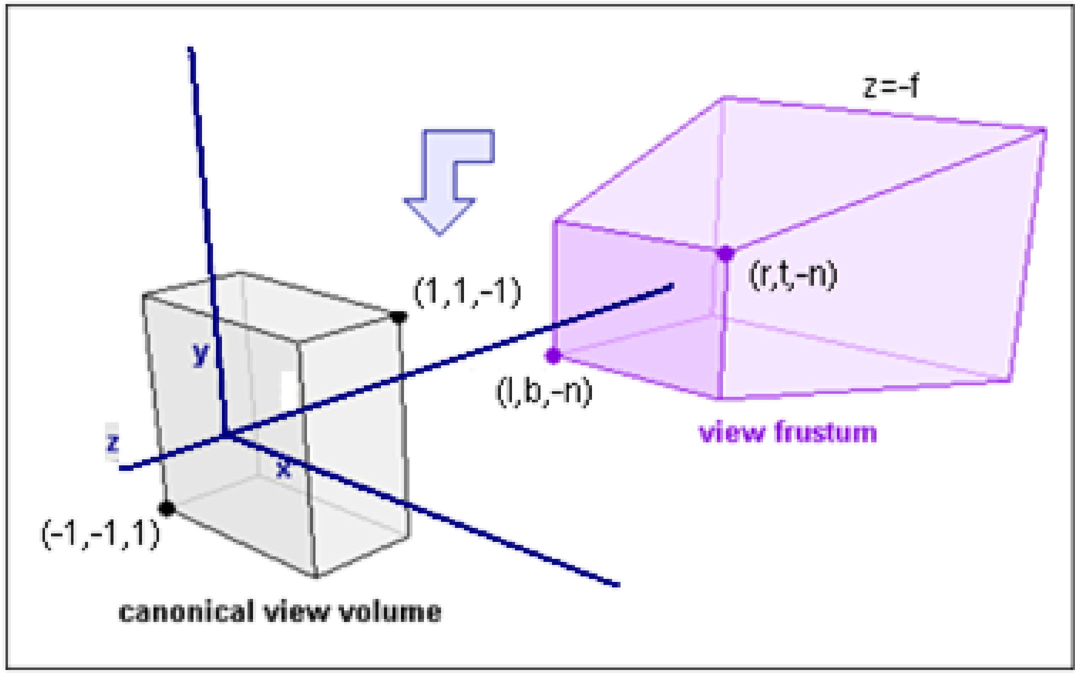
$$\boldsymbol{P}\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{nx}{z} \\ \frac{ny}{z} \\ \frac{(n+f)z - fn}{z} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{nx}{z} \\ \frac{ny}{z} \\ n + f - \frac{fn}{z} \\ 1 \end{pmatrix}$$

Check whether $\boldsymbol{P}$ preserves the relative order of z values ($f < z < n$)

Transforming the View Frustum

Canonical view volume

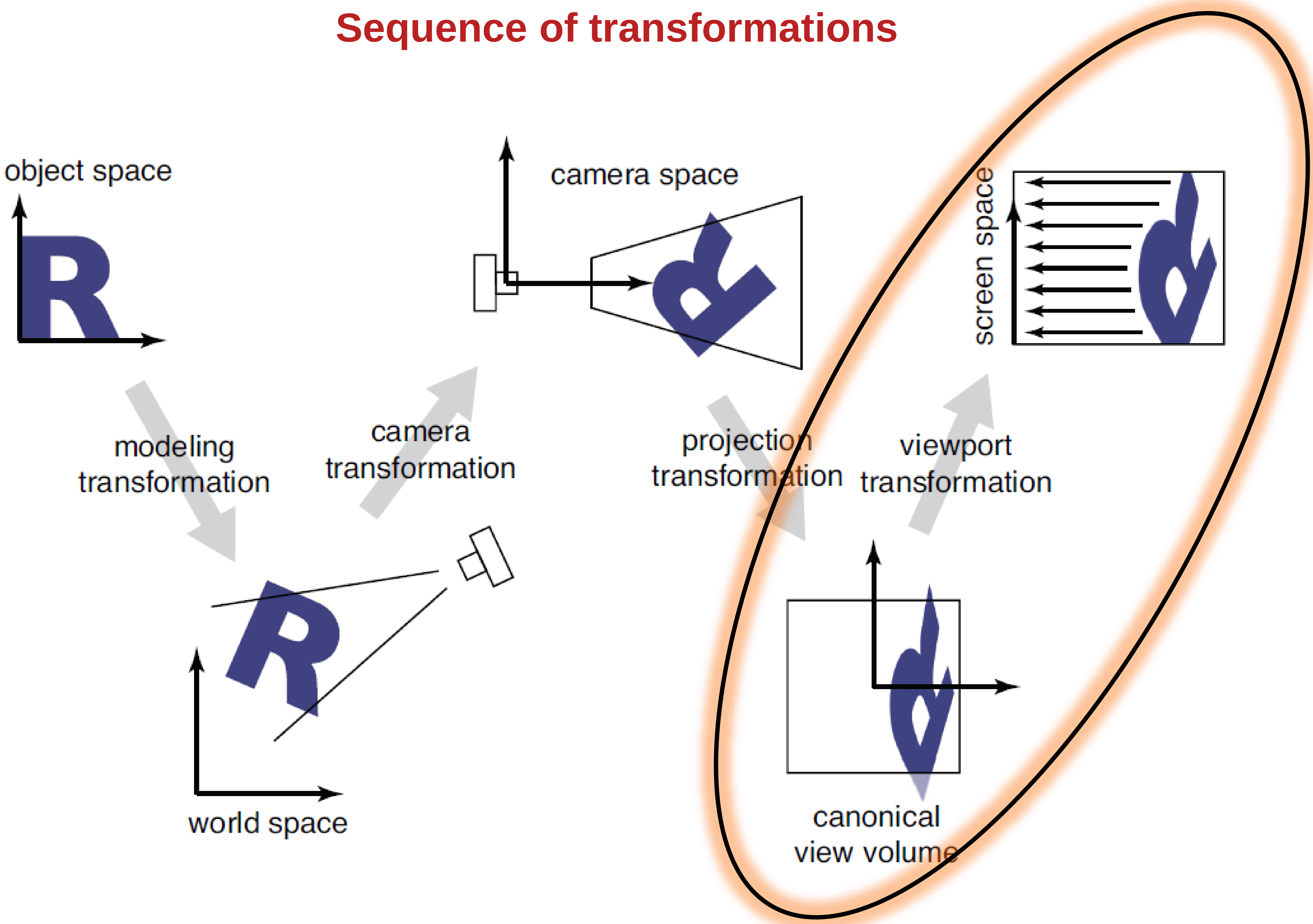


- Frustrum dimensions
  - $left \leq x \leq right,$
  - $bottom \leq y \leq top,$
  - $near \leq z \leq far$

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

Pipeline of transformations

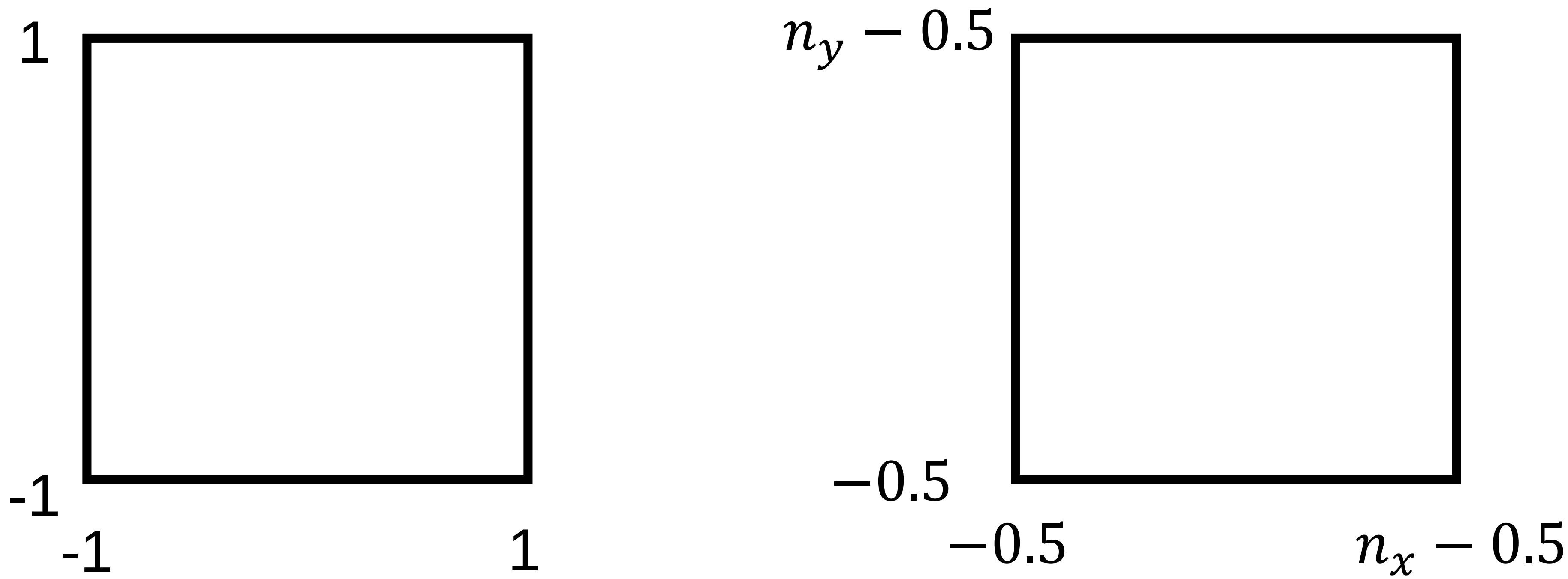
Sequence of transformations



Slide credits: S. Marschner

Viewport Transform

Canonical space to screen space



$$\begin{pmatrix} x_{screen} \\ y_{screen} \\ 1 \end{pmatrix} = \overbrace{\begin{pmatrix} \frac{n_x}{2} & 0 & \frac{n_x - 1}{2} \\ 0 & \frac{n_y}{2} & \frac{n_y - 1}{2} \\ 0 & 0 & 1 \end{pmatrix}}^{M_{vp}} \begin{pmatrix} x_{canonical} \\ y_{canonical} \\ 1 \end{pmatrix}$$

Pipeline of transformations

Canonical space to screen space

1. Transform into world coords (modeling transform)
2. Transform into camera coords (camera transform)
3. Perspective matrix,
4. Orthographic projection,
5. View Transform

$$\boldsymbol{p_{screen}} = \boldsymbol{M_{vp} M_{orth} P M_{cam} M_m p_{object}}$$

$$\begin{pmatrix} x_{screen} \\ y_{screen} \\ z_{screen} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{n_x}{2} & 0 & 0 & \frac{n_x - 1}{2} \\ 0 & \frac{n_y}{2} & 0 & \frac{n_y - 1}{2} \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2}{n-f} & -\frac{n+f}{n-f} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{pmatrix} \boldsymbol{M_{cam} M_m} \begin{pmatrix} x_o \\ y_o \\ z_o \\ 1 \end{pmatrix}$$

Application

Our projection is designed to both preserve local shape and maintain straight scene lines that are marked by the user with our interactive tool

Optimizing Content Preserving Projections for Wide-Angle Images

[Carroll et al., SIGGRAPH 2009]

http://vis.berkeley.edu/papers/capp/

Perspective

Mercator

Stereographic

Our Result

Reading

B1: Chapter 7