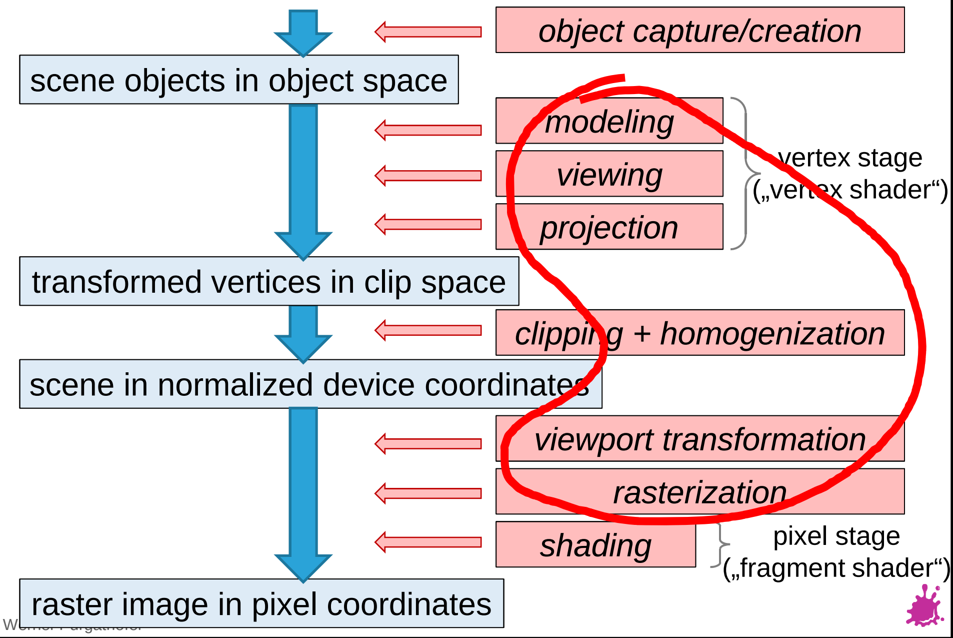
Einführung in Visual Computing

186.822

Viewing

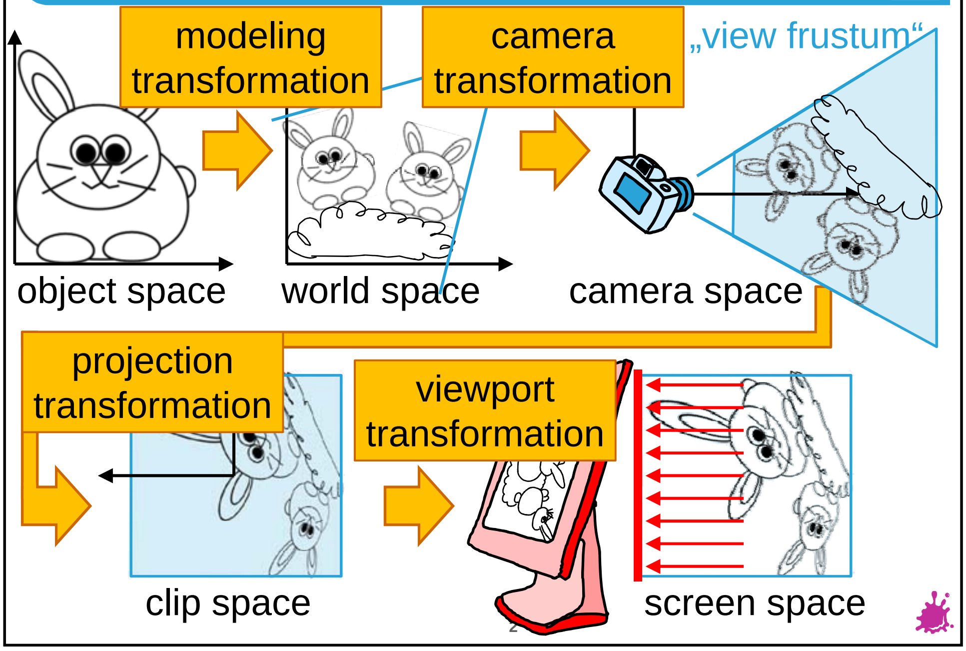
Werner Purgathofer

Rendering Pipeline

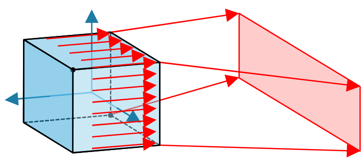


Werner Purgathofer

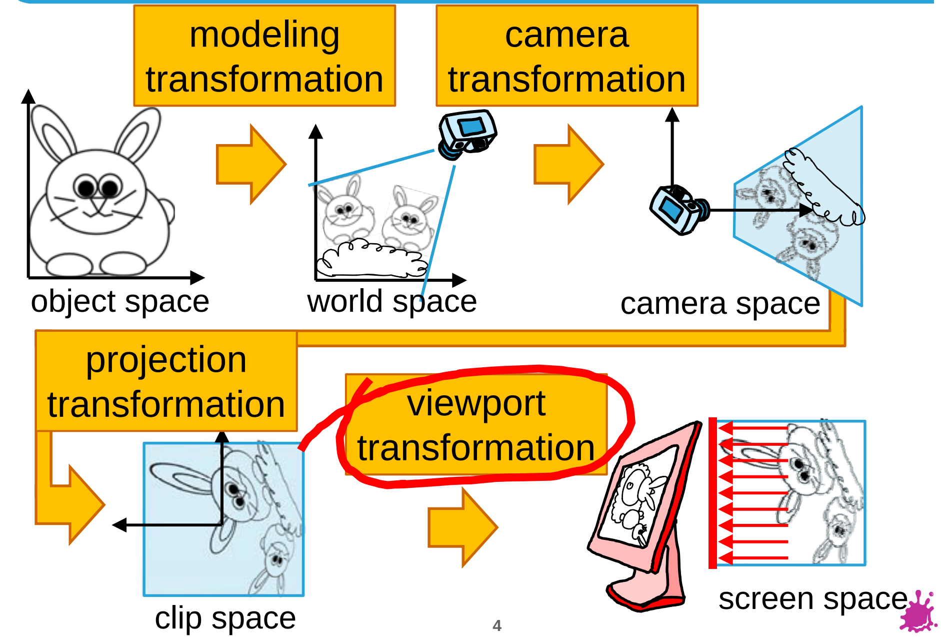
From Object Space to Screen Space



Viewport Transformation

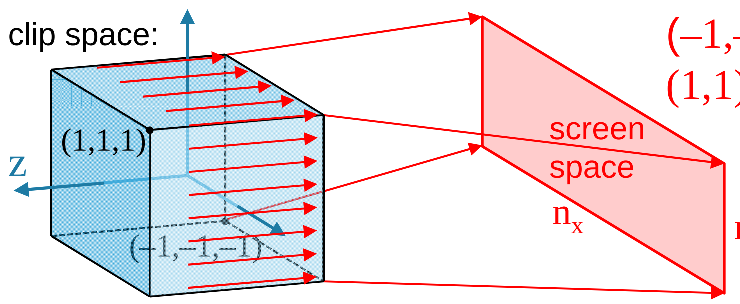


From Object Space to Screen Space



Viewport Transformation (1)

- assumption: scene is in clip space !
- clip space = $[-1,1] \times [-1,1] \times [-1,1]$
- orthographic camera looking in $-z$ direction
- screen resolution $n_x \times n_y$ pixels



$(-1,-1) \rightarrow (0,0)$

$(1,1) \rightarrow (n_x,n_y)$

Viewport Transformation (2)

can be done with the matrix

$$\begin{bmatrix} x_{screen} \\ y_{screen} \\ 1 \end{bmatrix} = \begin{bmatrix} n_x/2 & 0 & n_x/2 \\ 0 & n_y/2 & n_y/2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$(-1,-1) \rightarrow (0,0)$
$(1,1) \rightarrow (n_x,n_y)$

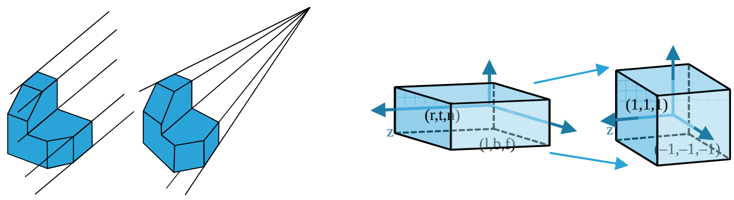
this ignores the z-coordinate, but…

Viewport Transformation (3)

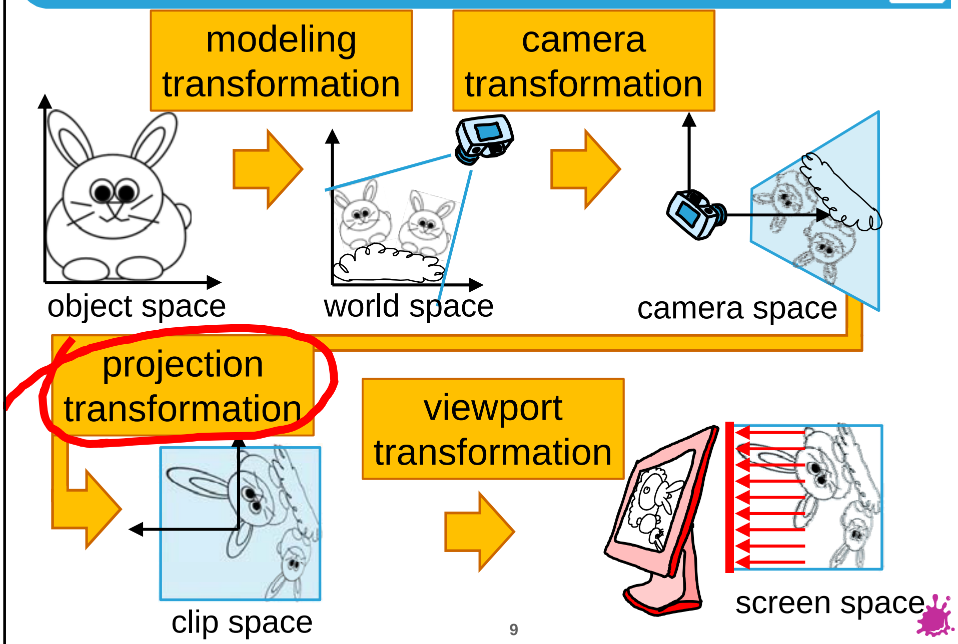
… we will need z later to remove hidden parts of the image, so we add a row and a column to keep z :

$$\begin{bmatrix} x_{screen} \\ y_{screen} \\ \mathbf{z} \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} n_x/2 & 0 & \mathbf{0} & n_x/2 \\ 0 & n_y/2 & \mathbf{0} & n_y/2 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ 0 & 0 & \mathbf{0} & 1 \end{bmatrix}}_{M_{vp}} \cdot \begin{bmatrix} x \\ y \\ \mathbf{z} \\ 1 \end{bmatrix}$$

Projection Transformation

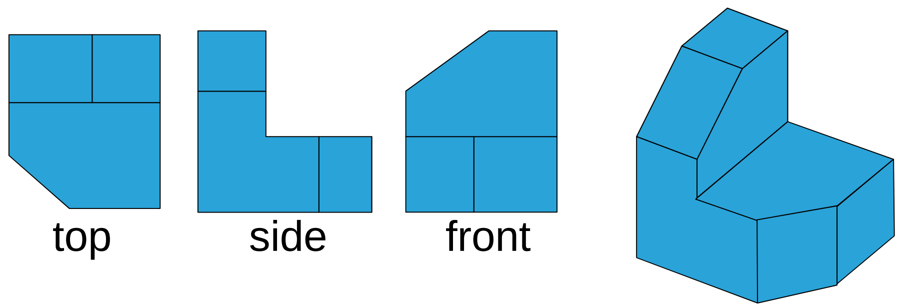


From Object Space to Screen Space



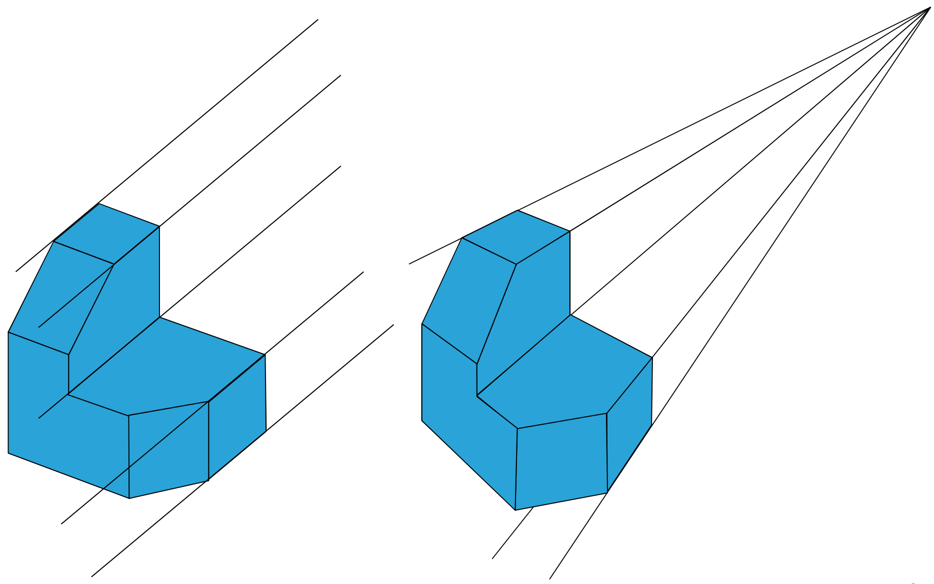
9

Parallel Projection (Orthographic Projection)

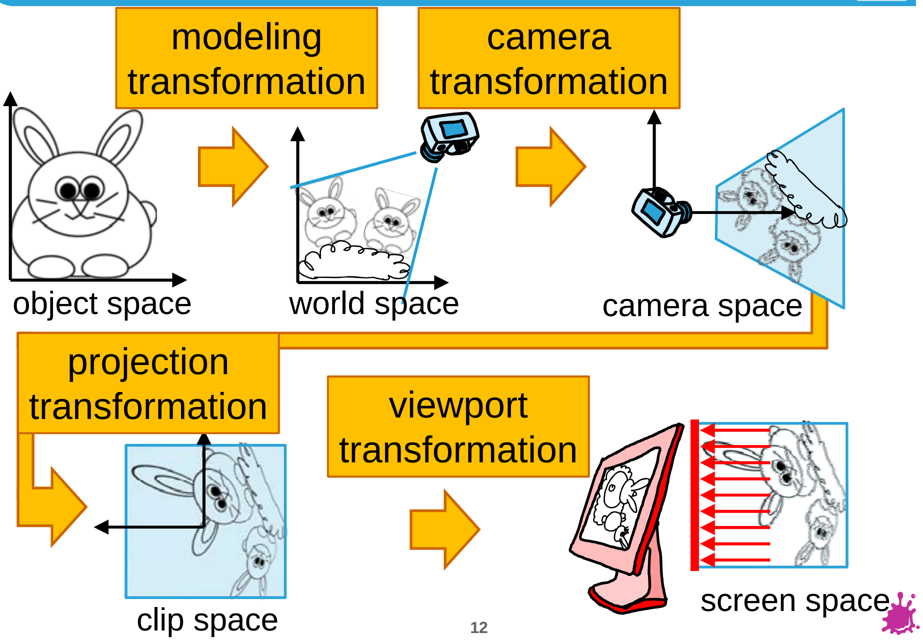


3 parallel-projection views of an object, showing relative proportions from different viewing positions

Parallel vs. Perspective Projection

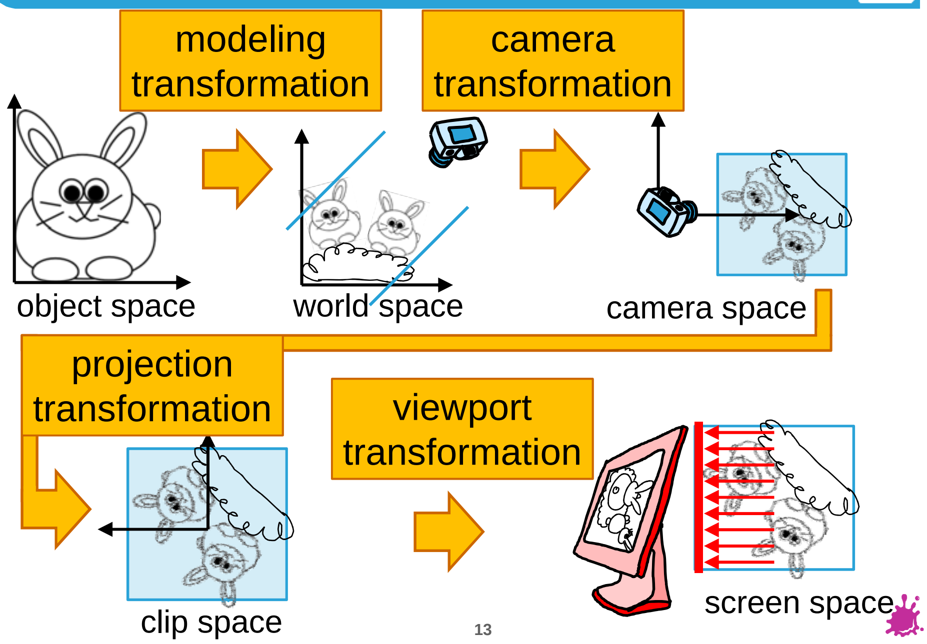


For Now: Parallel (Orthographic) Projection



12

For Now: Parallel (Orthographic) Projection

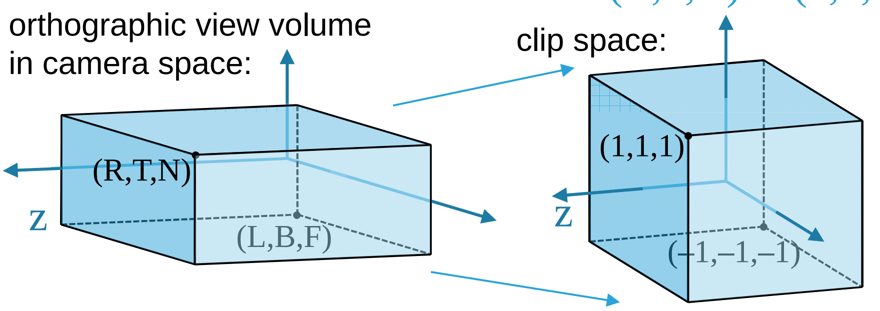


13

Projection Transformation (Orthographic)

- assumption: scene in box [L,R]×[B,T]×[F,N]
- orthographic camera looking in –z direction
- transformation to clip space

(L,B,F) → (–1,–1,–1)
(R,T,N) → (1,1,1)



Projection Transformation (Orthographic)

(L,B,F) → (–1,–1,–1)
(R,T,N) → (1,1,1)

$$M_{orth} = \begin{bmatrix} \frac{2}{R-L} & 0 & 0 & -\frac{R+L}{R-L} \\ 0 & \frac{2}{T-B} & 0 & -\frac{T+B}{T-B} \\ 0 & 0 & \frac{2}{N-F} & -\frac{N+F}{N-F} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

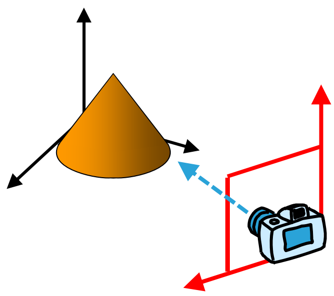
Parallel Projection (1)

orthographic projection

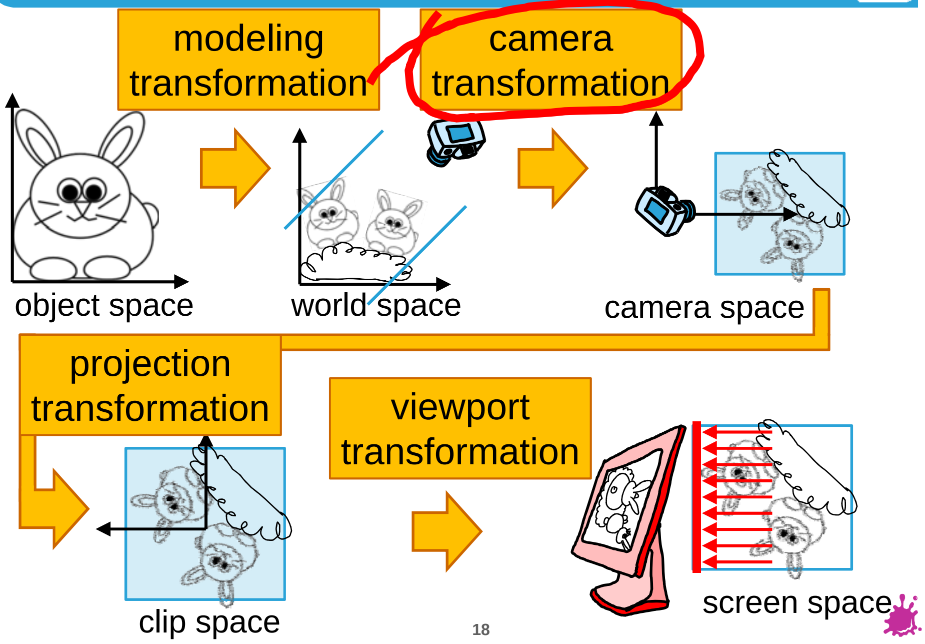
oblique projection

orientation of the projection vector –g

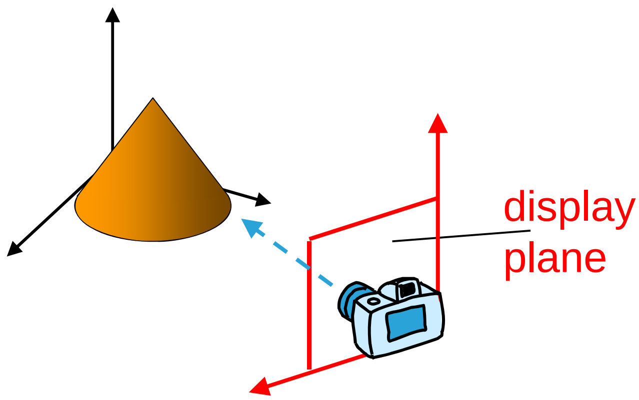
Camera Transformation



For Now: Parallel (Orthographic) Projection



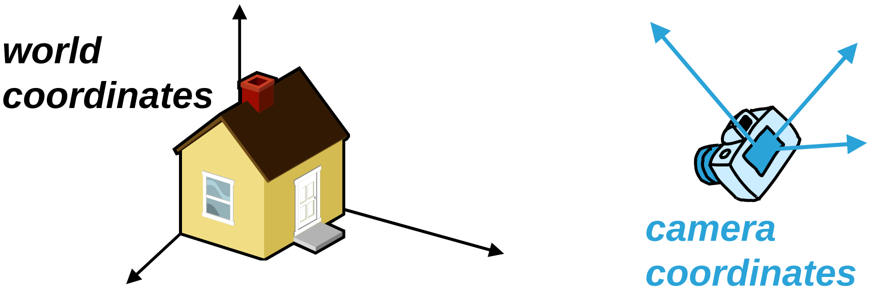
Viewing: Projection Plane



coordinate reference for obtaining a selected view of a 3D scene

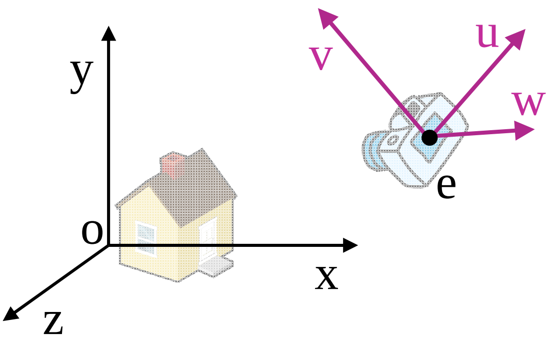
Viewing: Camera Definition

- similar to taking a photograph
- involves selection of
  - camera position
  - camera direction
  - camera orientation
  - “window” (aperture) of camera



Viewing: Camera Transformation (1)

- view reference point
  - origin of camera coordinate system
  - camera position or look-at point



right-handed camera-coordinate system, with axes u, v, w, *relative to world-coordinate scene*

Viewing: Camera Transformation (2)

e … eye position

g … gaze direction

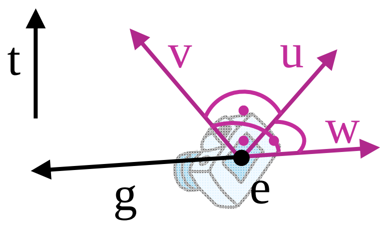
(positive w-axis points to the viewer)

t … view-up vector

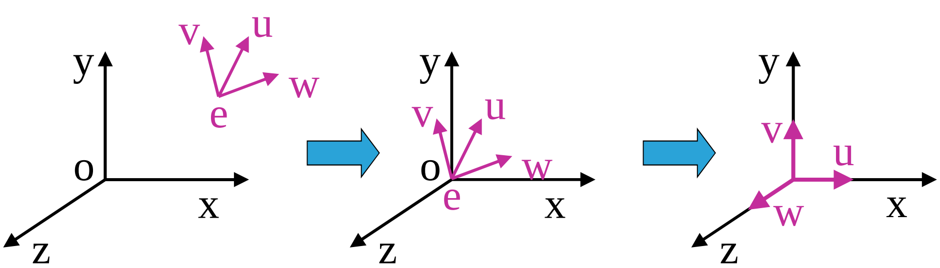
$$w = -\frac{g}{\|g\|}$$

$$u = \frac{t \times w}{\|t \times w\|}$$

$$v = w \times u$$



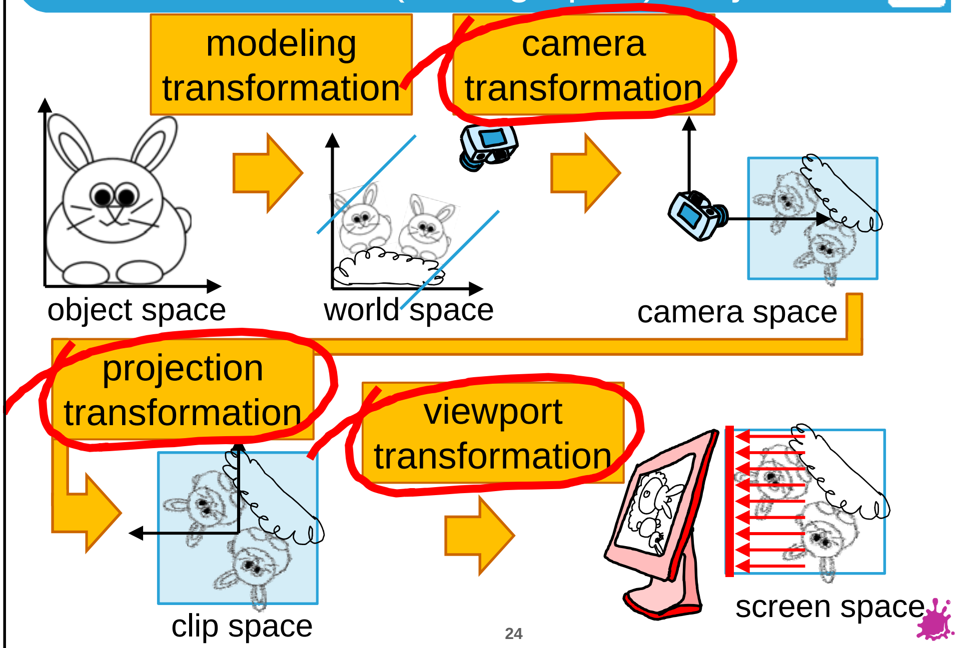
Viewing: Camera Transformation (3)



$$\mathbf{M_{cam}} = \mathbf{R_z} \cdot \mathbf{R_y} \cdot \mathbf{R_x} \cdot \mathbf{T}$$

aligning viewing system with world-coordinate axes using translate-rotate transformations

For Now: Parallel (Orthographic) Projection



Viewing: Camera + Projection + Viewport

$$\begin{bmatrix} x_{screen} \\ y_{screen} \\ z \\ 1 \end{bmatrix} = (M_{vp} \cdot M_{orth} \cdot M_{cam}) \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

pixels on the screen

viewport transformation

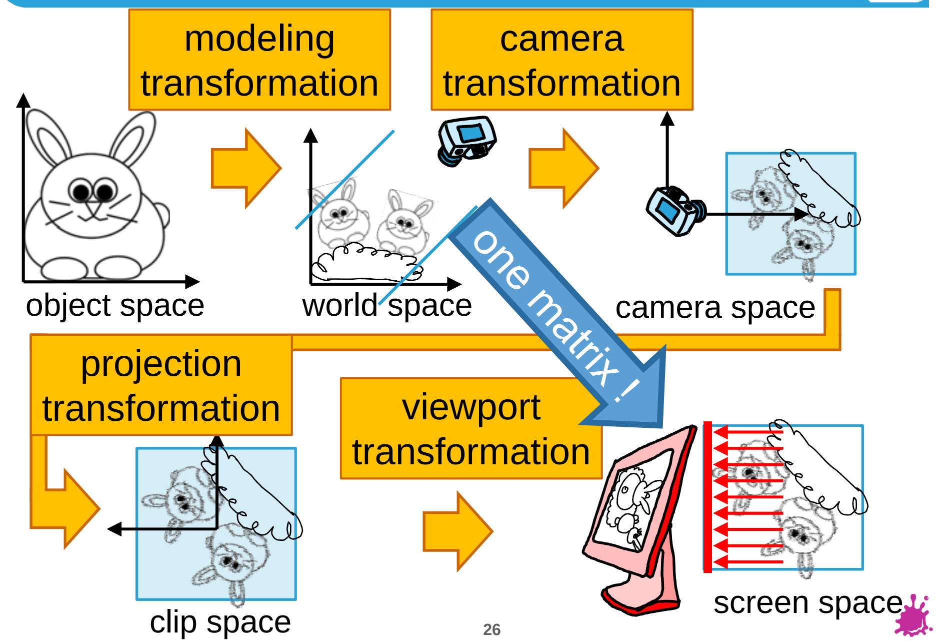
projection transformation

camera transformation

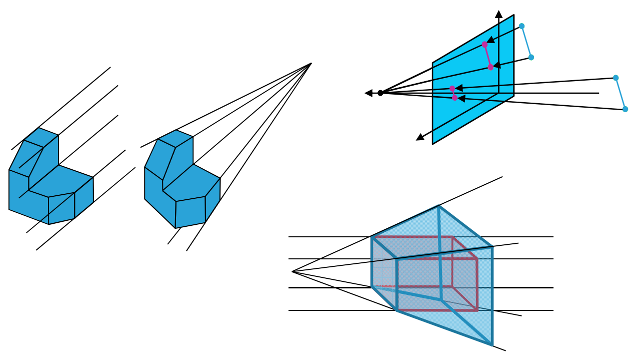
world coordinates

Werner Purgathofer

For Now: Parallel (Orthographic) Projection

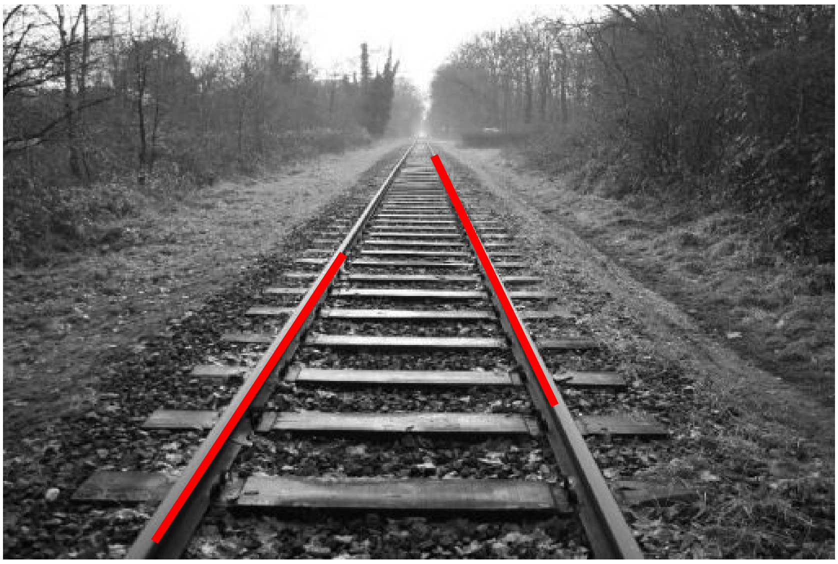


Perspective Projection

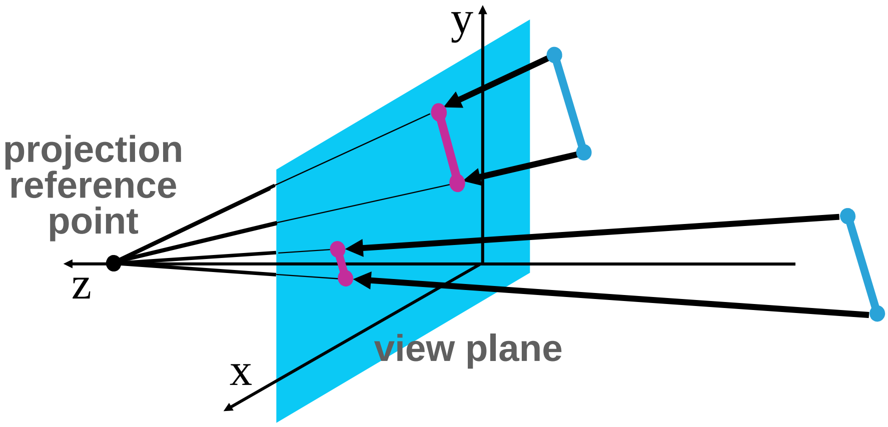


Perspective Projection

Perspective Projection

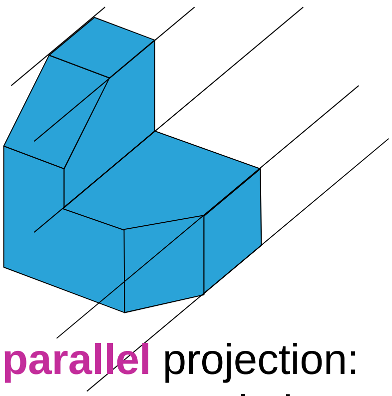


Perspective Projection

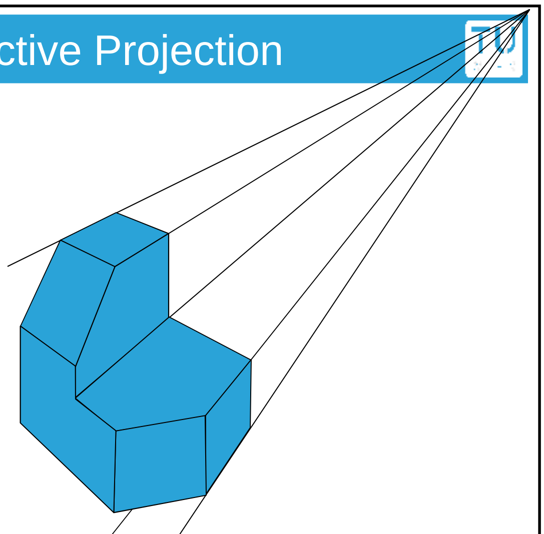


perspective projection of equal-sized objects at different distances from the view plane

Parallel vs. Perspective Projection

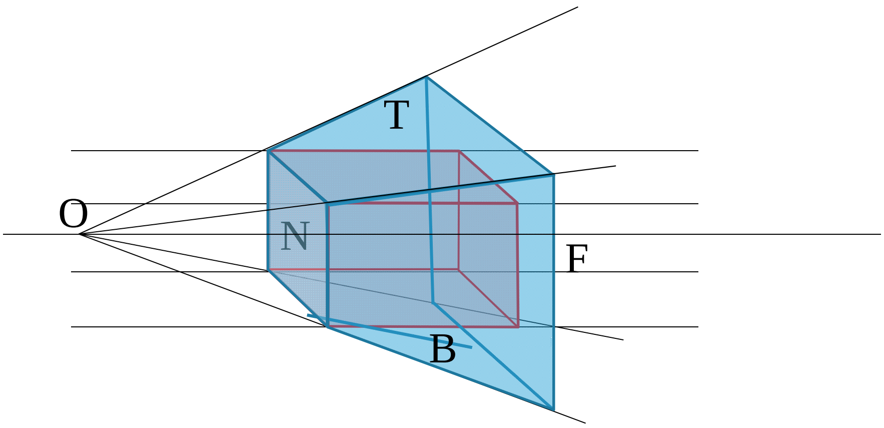


parallel projection:
preserves relative proportions & parallel features (affine transform.)

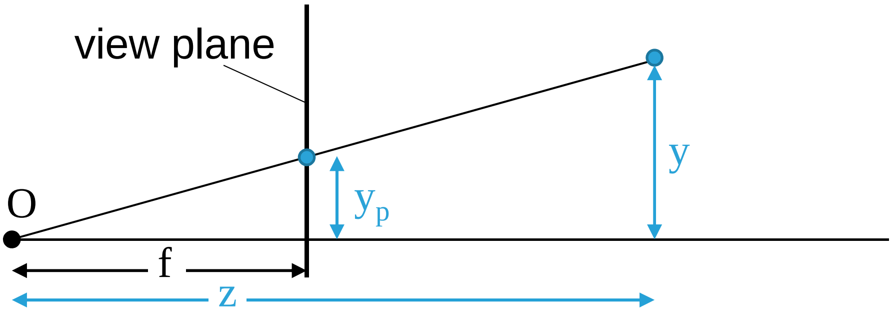


perspective projection:
center of projection, realistic views

Perspective Transform



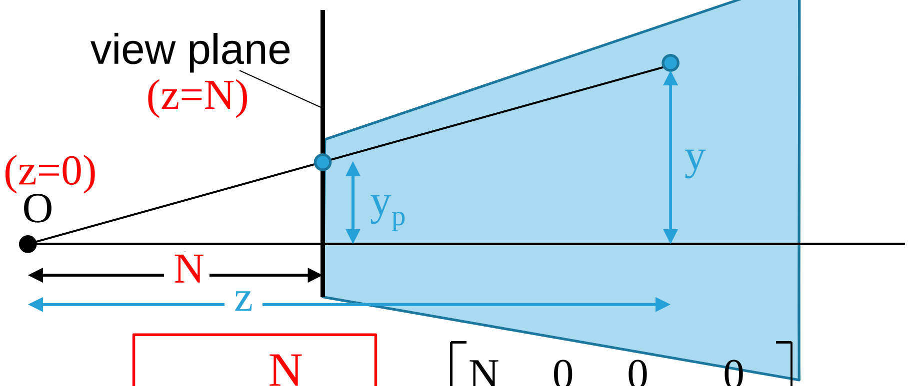
Perspective Transformation (1)



$$y_p = \frac{f}{z}\,y$$

f … focal length

Perspective Transformation (2)



$$y_p = \frac{N}{z}\,y$$

analogous:

$$x_p = \frac{N}{z}\,x$$

$$\begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N+F & -F\cdot N \\ 0 & 0 & 1 & 0 \end{bmatrix} = \mathbf{P}$$

Perspective Transformation (3)

$$\mathbf{P} = \begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N+F & -F\cdot N \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x\cdot N \\ y\cdot N \\ z\cdot(N+F) - F\cdot N \\ z \end{bmatrix}$$

homogenization: divide by z

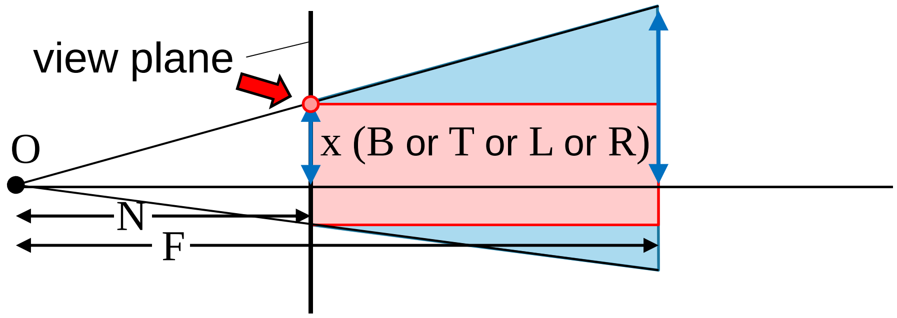
$$y_p = \frac{N}{z}\,y$$

$$x_p = \frac{N}{z}\,x$$

$$\sim> \begin{bmatrix} x\cdot N/z \\ y\cdot N/z \\ (N+F) - F\cdot N/z \\ 1 \end{bmatrix}$$

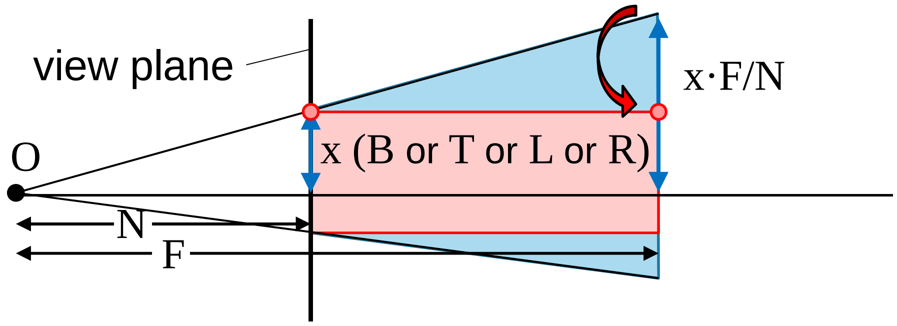
Example: Right Top Near Corner

$$\begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N+F & -F\cdot N \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} R \\ T \\ N \\ 1 \end{bmatrix} = \begin{bmatrix} R\cdot N \\ T\cdot N \\ N\cdot(N+F) - F\cdot N \\ N \end{bmatrix} \sim> \begin{bmatrix} R \\ T \\ N \\ 1 \end{bmatrix}$$

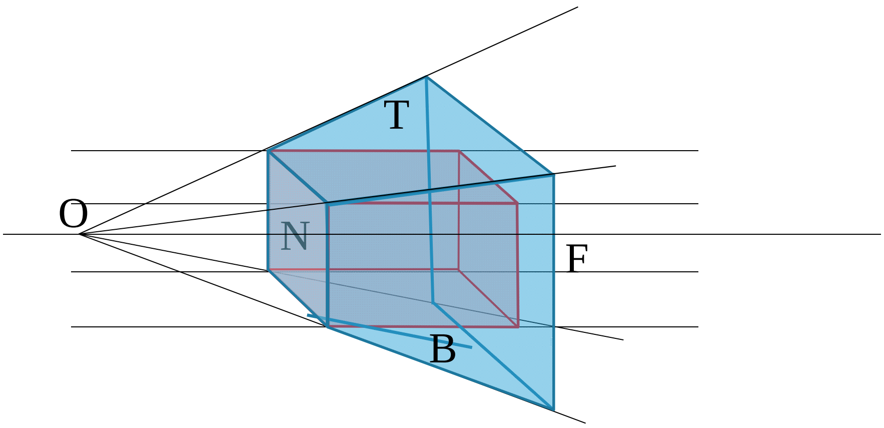


Example: Right Top Near Corner

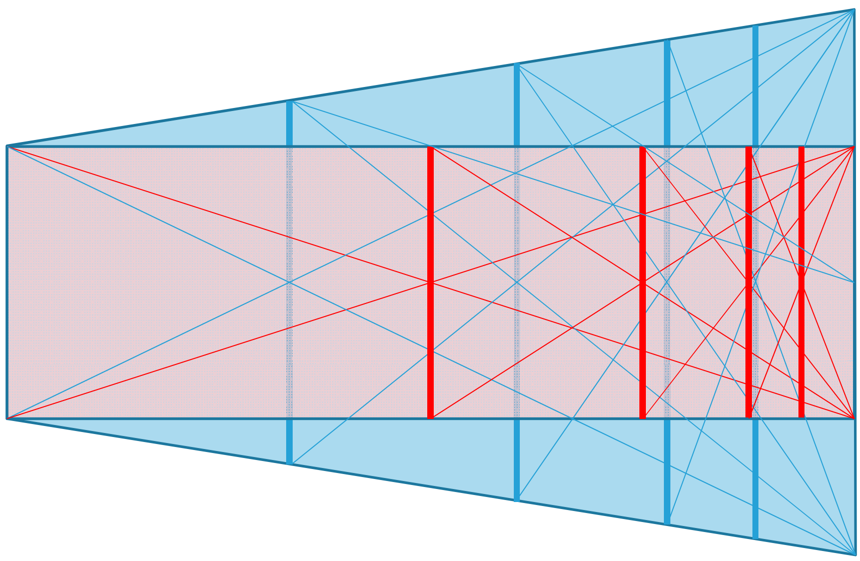
$$\begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N+F & -F\cdot N \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} L\cdot F/N \\ B\cdot F/N \\ F \\ 1 \end{bmatrix} = \begin{bmatrix} L\cdot F \\ B\cdot F \\ F\cdot(N+F) - F\cdot N \\ F \end{bmatrix} \sim> \begin{bmatrix} L \\ B \\ F \\ 1 \end{bmatrix}$$



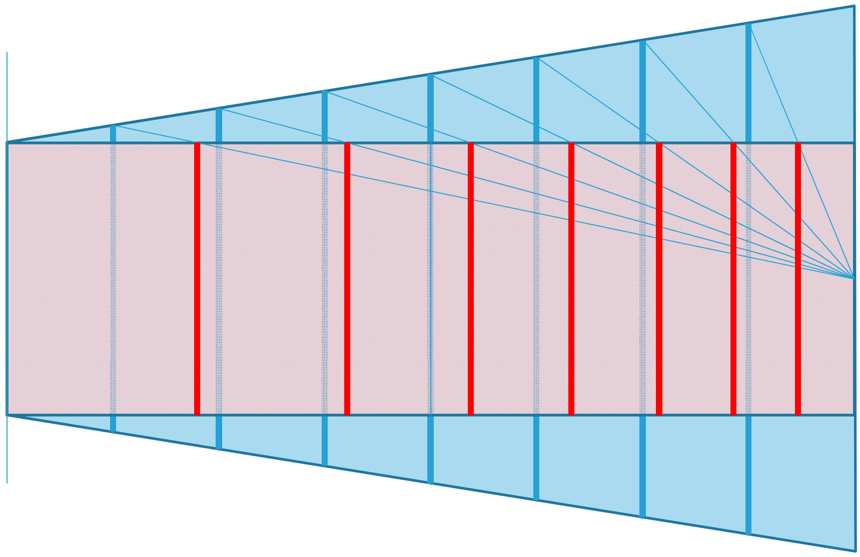
Perspective Transform



Nonlinear z-Behaviour

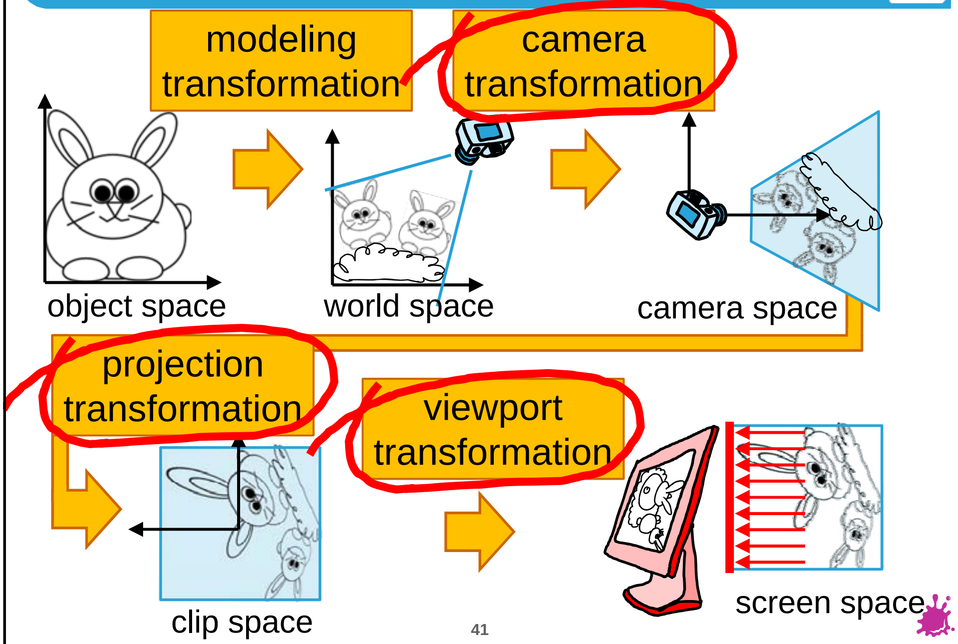


Nonlinear z-Behaviour



Werner Purgathofer

From Object Space to Screen Space



Viewing: Camera + Projection + Viewport

$$\begin{bmatrix} x_{screen} \\ y_{screen} \\ z' \\ 1 \end{bmatrix} = \Big(M_{vp} \cdot \overbrace{M_{orth} \cdot P}^{M_{per}} \cdot M_{cam} \Big) \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

pixels on the screen

viewport transformation

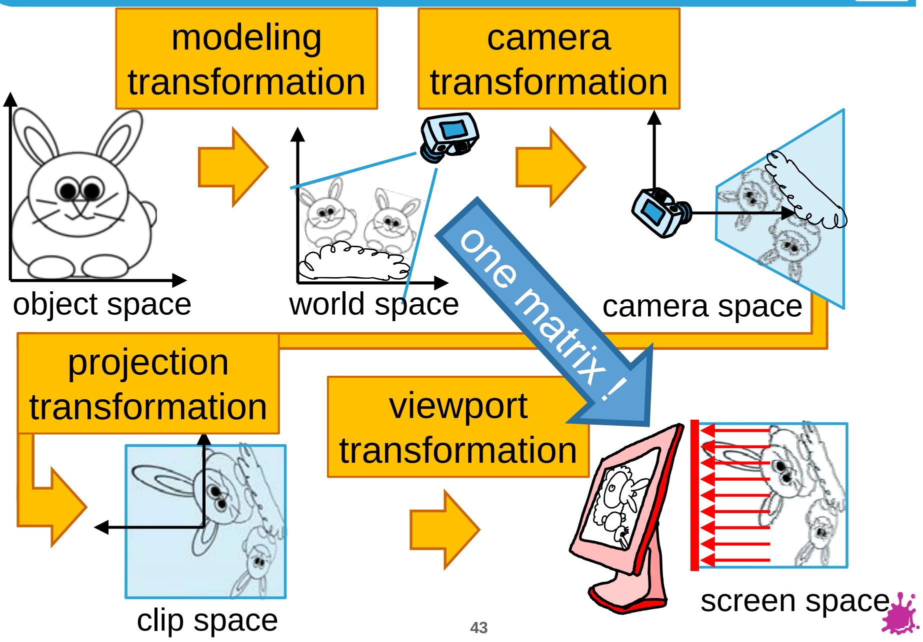
projection transformation

perspective projection

camera transformation

world coordinates

From Object Space to Screen Space



z-Values Remain in Order

$$\begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N+F & -F \cdot N \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \sim> \begin{bmatrix} x \cdot N/z \\ y \cdot N/z \\ (N+F) - F \cdot N/z \\ 1 \end{bmatrix}$$

$z_1, z_2, N, F < 0$

$z_1 < z_2$

$1/z_1 > 1/z_2 \qquad | \cdot (-F \cdot N) \qquad (<0)$

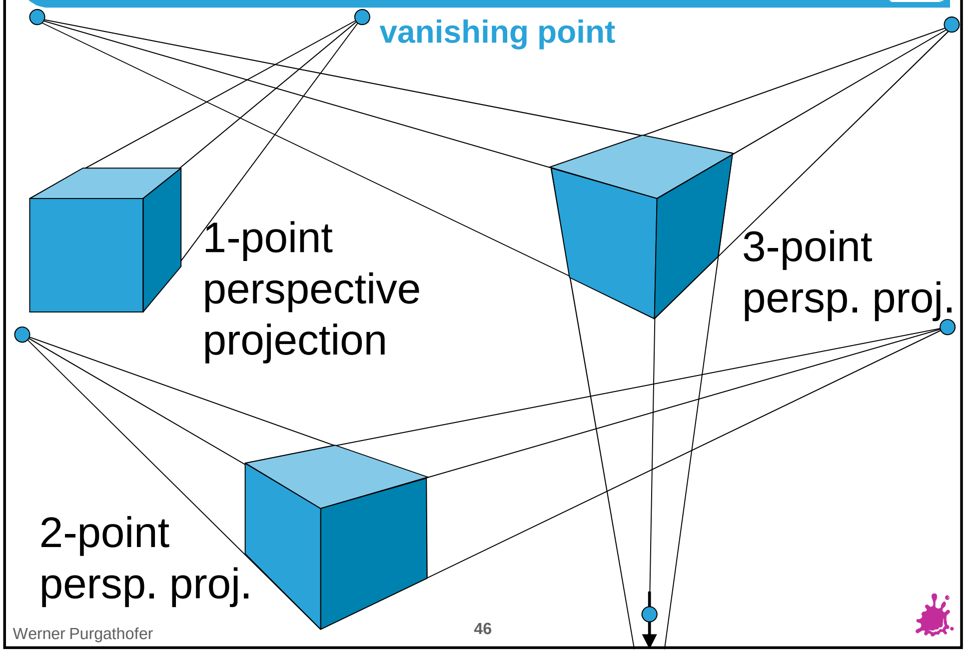
$-F \cdot N/z_1 < -F \cdot N/z_2 \qquad | + (N+F)$

$(N+F) - F \cdot N/z_1 < (N+F) - F \cdot N/z_2$

Perspective Projection Properties

- parallel lines parallel to view plane ⇒ parallel lines
- parallel lines not parallel to view plane ⇒ converging lines (vanishing point)
- lines parallel to coordinate axis ⇒ principal vanishing point (one, two or three)

Principle Vanishing Points



Outlook: Illumination and Shadows

- perspective projection
- local and global illumination models
- shadow generation