Einführung in die Erweiterte Realität

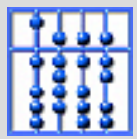
2. Coordinate Systems and Projective Geometry (2D)

Prof. Gudrun Klinker, Ph.D.
Institut für Informatik,Technische Universität München
klinker@in.tum.de

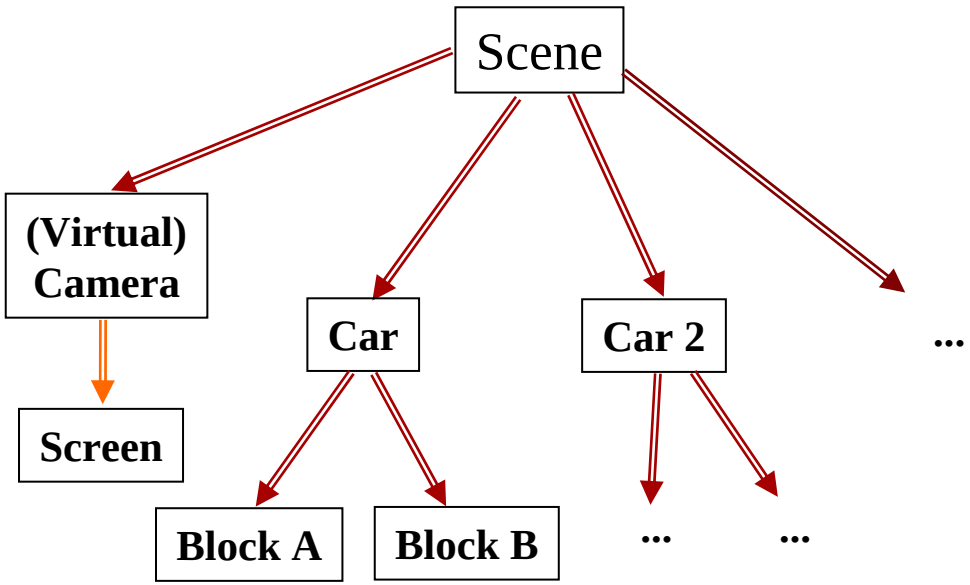
Oct 26, 2004

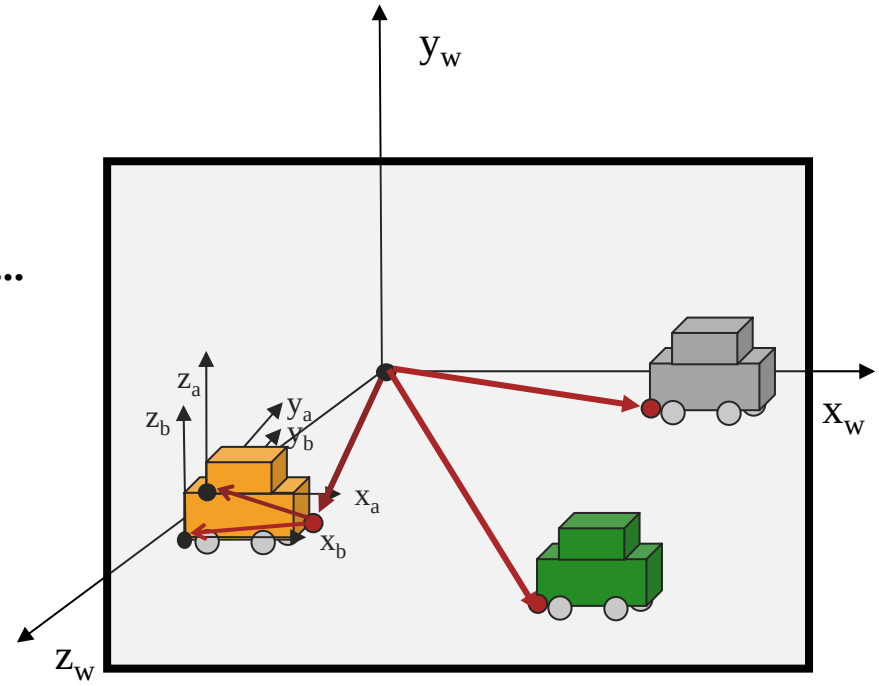
Schedule

- 19.10. 1. Introduction
- **26.10. 2. Coordinate Systems, Projective Geometry (2D)**
- 2.11. - no class -
- 9.11. 3. Rendering (Open GL, Occlusions)
- 16.11. 4. APIs (VRML, Open Inventor)
- 23.11. 5. Displays
- 30.11. 6. Tracking Devices
- 7.12. 7. Camera Calibration (Tsai)
- 14.12. 8. Basics of Computer Vision
- 21.12. 9.Sensor Fusion
- 28.12., 4.1. – winter holidays –
- 11.1. 10. Mixed Reality, Information Presentation
- 18.1. 11. System Architectures for Augmented Reality
- 25.1. 12. Context Toolkit
- 1.2. 13. Intelligent Environments and Ubiquitous Computing
- 8.2. 14. Discussion



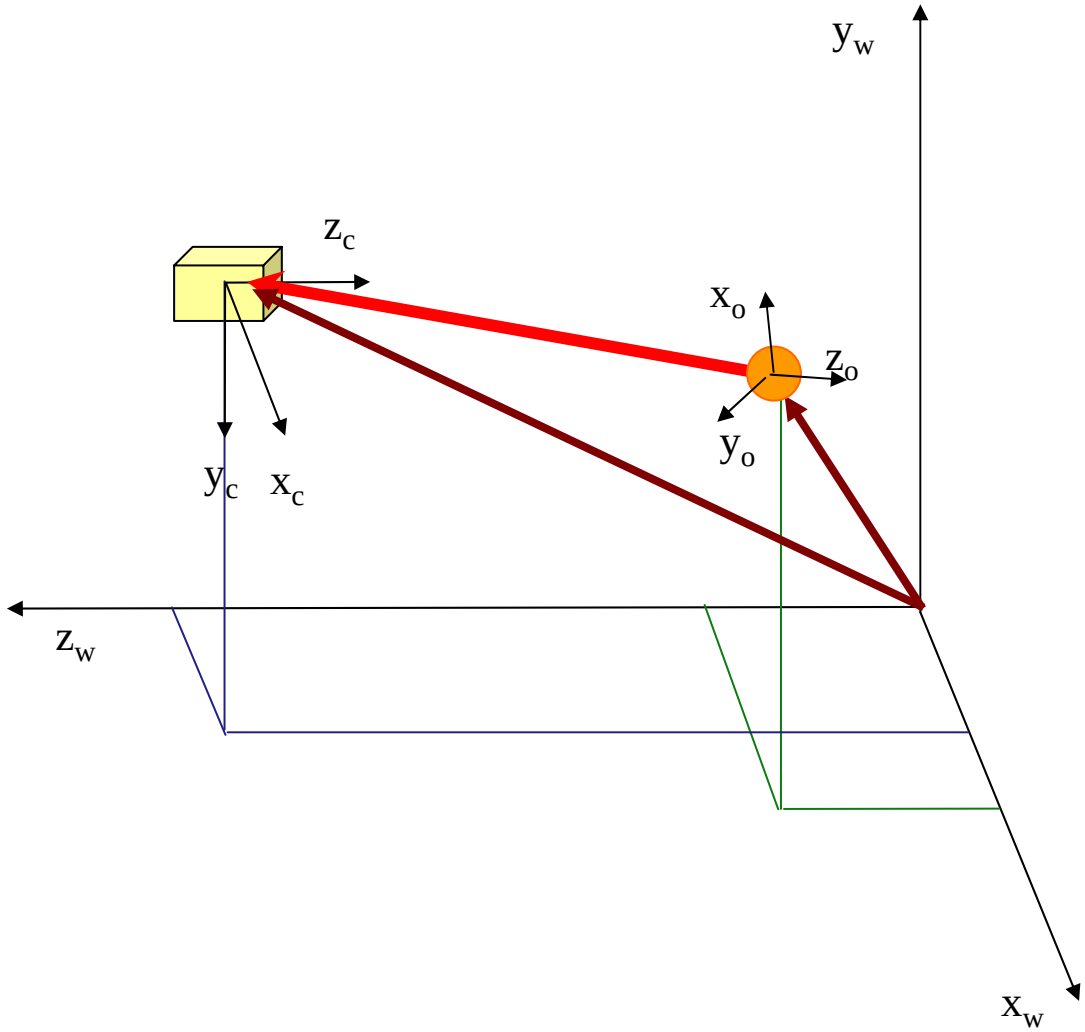
Scene Graph


Scene
(Virtual) Camera
Screen
Car
Car 2
...
Block A
Block B
...
...


y_w
x_w
z_w
z_a
z_b
y_a
y_b
x_a
x_b

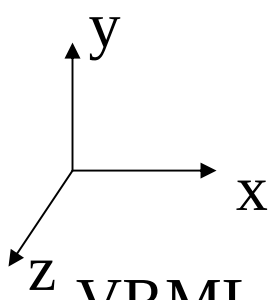
Rendering = Traversal of the Scene Graph

- **Homogeneous coordinates**
  3D Transformations between
  - Object coordinates
  - World coordinates
  - Camera coordinates
- **Geometry Pipeline in OpenGL**

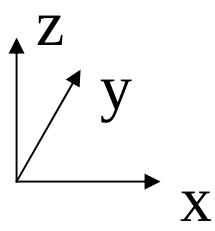


Different Coordinate Systems

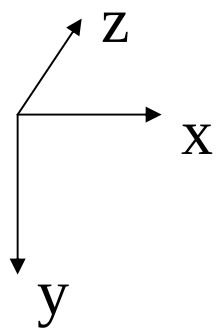
- Righthanded Systems



VRML, openGL

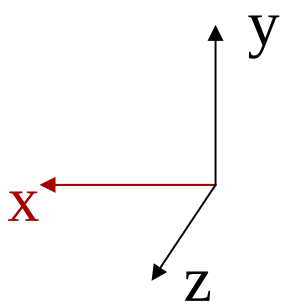


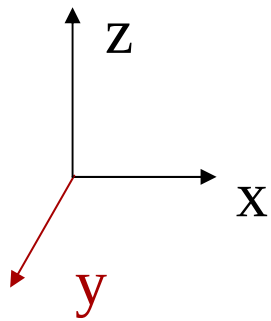
Apps

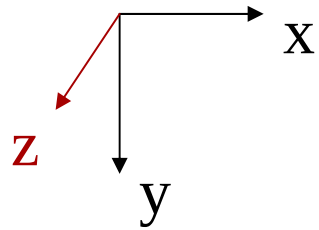


Video

- Lefthanded Systems



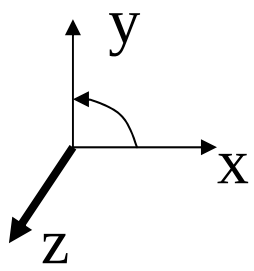




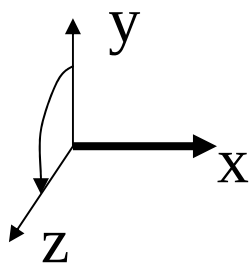
Video

Righthanded Rotations

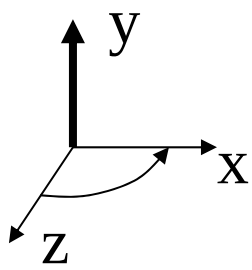
- Around z: $x \rightarrow y$



- Around x: $y \rightarrow z$



- Around y: $z \rightarrow x$

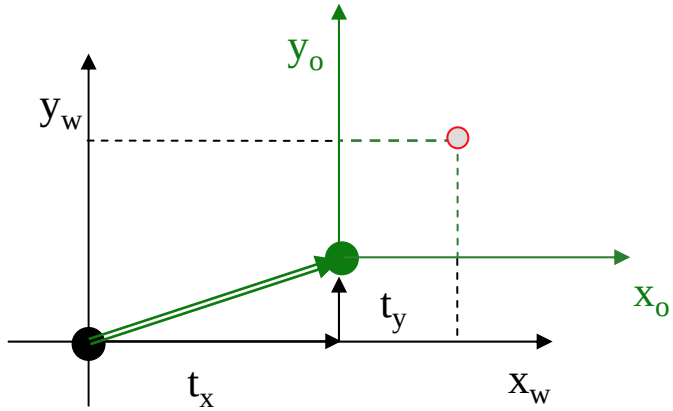


Counterclockwise rotation around respective axis

2D Transformations

- Translation:

$$x_w = x_o + t_x$$
$$y_w = y_o + t_y$$

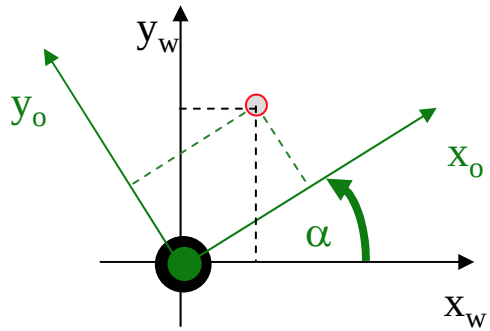


- Scaling:

$$x_w = s_x * x_o$$
$$y_w = s_y * y_o$$

- Rotation:

$$x_w = \cos\alpha * x_o - \sin\alpha * y_o$$
$$y_w = \sin\alpha * x_o + \cos\alpha * y_o$$



2D Transformations Using Homogeneous Coordinates

- Translation:

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} x + wt_x \\ y + wt_y \\ w \end{bmatrix}$$

- Scaling:

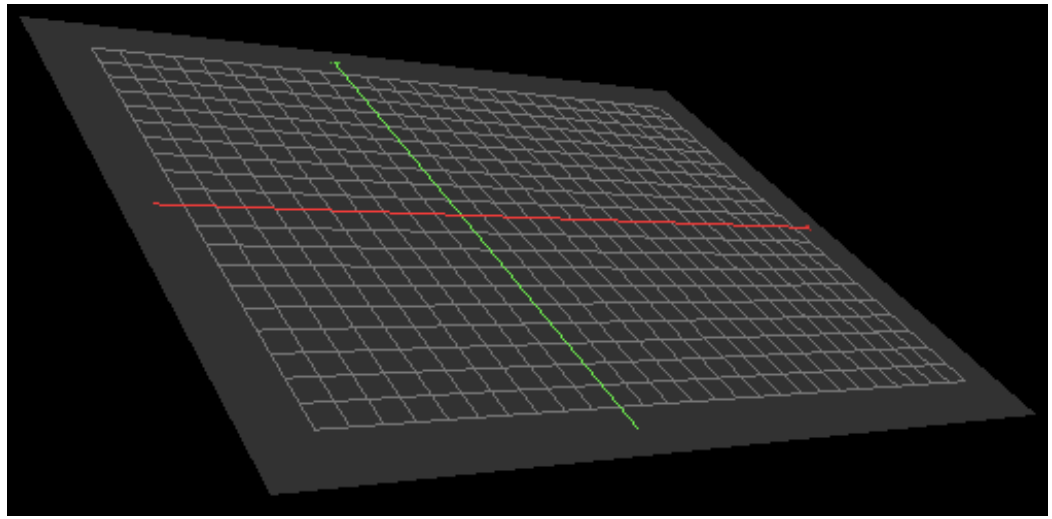
$$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \\ w \end{bmatrix}$$

- Rotation:

$$\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} \cos\alpha * & -\sin\alpha & *y \\ \sin\alpha * x + \cos\alpha & *y \\ w \end{bmatrix}$$

Projective Geometry and Transformations in 2D

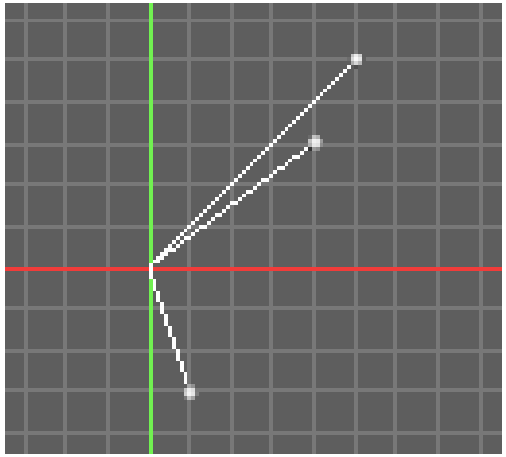
- Geometric distortion due to perspective Projection
- *Invariant*:
  - Straight lines
- Not invariant:
  - Angles
  - Parallel lines



2D Projective Plane - Points -

Inhomogeneous notation in R^2

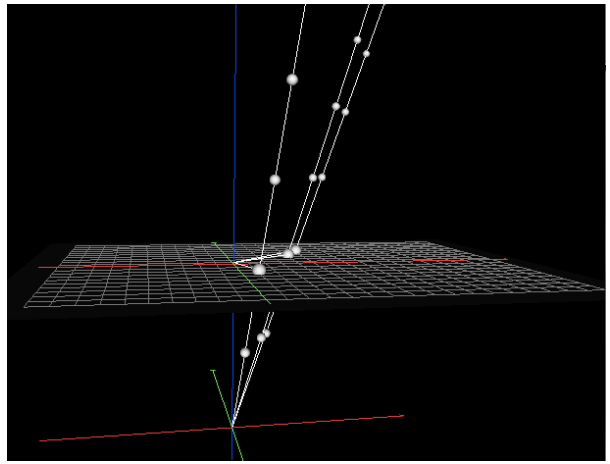
$$P = \begin{pmatrix} x, & y \end{pmatrix}^{\mathrm{T}}$$



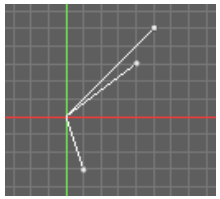
Homogeneous notation in P^2 (*projective space*)

$$\mathbf{x} = \begin{pmatrix} wx, & wy, & w \end{pmatrix}^{\mathrm{T}}$$

$$= w\begin{pmatrix} x, & y, & 1 \end{pmatrix}^{\mathrm{T}}$$



Examples

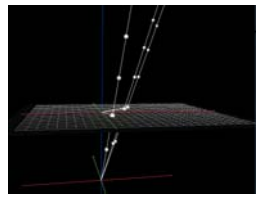


$\mathbb{R}^2$

$$P_1 = (0.4, \quad 0.3)^T$$

$$P_2 = (0.1, \quad -0.3)^T$$

$$P_3 = (0.5, \quad 0.5)^T$$



$\mathbb{P}^2$

$$\mathbf{x}_1 = (0.4w, \quad 0.3w, \quad w)^T$$

$$= (0.4, \quad 0.3, \quad 1.0)^T$$

$$= (0.8, \quad 0.6, \quad 2.0)^T$$

$$\mathbf{x}_2 = (0.3, \quad -0.9, \quad 3.0)^T$$

$$\mathbf{x}_3 = (0.5, \quad 0.5, \quad 1.0)^T$$

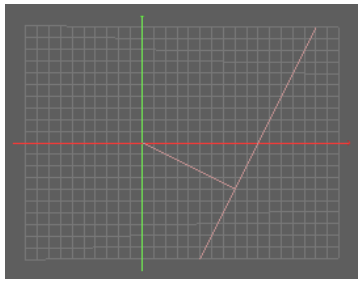
2D Projective Plane - Lines -

- Line equation:

$$ax + by + c = 0$$

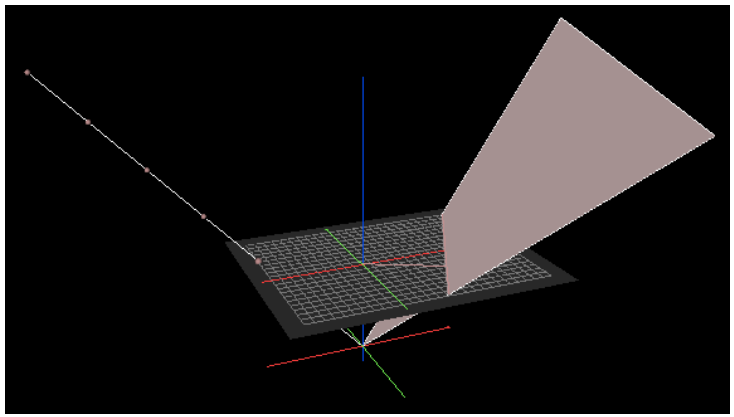
- Line normal:

n = (a, b)/|**n**|



- Homogeneous line notation in P^2 (projective space):

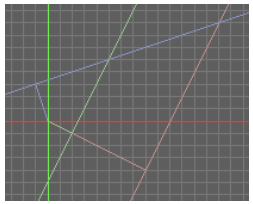
$$\mathbf{l} = k(a, \quad b, \quad c)^T$$

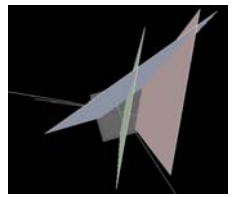


$$w(x, \quad y, \quad 1) \bullet k \begin{pmatrix} a \\ b \\ c \end{pmatrix} = 0$$

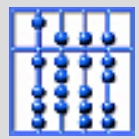
$$\mathbf{x}^T\mathbf{l} = \mathbf{l}^T\mathbf{x} = 0$$

Examples





$\mathbb{R}^2$	$\mathbb{P}^2$
$l_1 : 2x - y - 2 = 0$	$\mathbf{l}_1 = (2, \quad -1, \quad -2)^T$
	$= (-1, \quad 0.5, \quad 1.0)^T$
$l_2 : 2x - y - 0.5 = 0$	$\mathbf{l}_2 = (2, \quad -1, \quad -0.5)^T$
	$= (-4, \quad 2, \quad 1)^T$
$l_3 : x - 3y + 1 = 0$	$\mathbf{l}_3 = (1, \quad -3, \quad 1)^T$



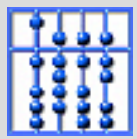
2D Projective Plane - Comparison -

$\mathbb{R}^2$

- Degrees of Freedom (*DOF*): 2
- Point P = (x,y)
- Line l: ax+by+c=0
  Normal **n**=(a,b)/|**n**|

$\mathbb{P}^2$

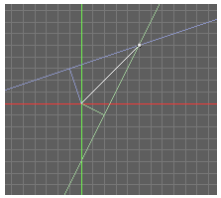
- Degrees of Freedom (DOF): 2
- Vector **x** = w(x,y,1)^T
- Vector **l** = k(a,b,c)^T
- *Duality*: $\mathbf{x}^T\mathbf{l} = \mathbf{l}^T\mathbf{x} = 0$



- Lines $\mathbf{l_1} = (a_1,b_1,c_1)^T$ and $\mathbf{l_2} = (a_2,b_2,c_2)^T$ intersect at a point $\mathbf{x} = (x,y,w)^T$.
- $\mathbf{x}$ is on $\mathbf{l_1}$ and on $\mathbf{l_2}$: $\mathbf{x}^T\mathbf{l}_1 = 0\,,\ \mathbf{x}^T\mathbf{l}_2 = 0$
- $\mathbf{x}$ is perpendicular to $\mathbf{l_1}$ and $\mathbf{l_2}$.
- $\mathbf{x}$ is cross product of $\mathbf{l_1}$ and $\mathbf{l_2}$.

$$\mathbf{x} = \mathbf{l_1} \times \mathbf{l_2} = \begin{vmatrix} i & j & k \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{vmatrix} = \begin{pmatrix} b_1c_2 - c_1b_2 \\ c_1a_2 - a_1c_2 \\ a_1b_2 - b_1a_2 \end{pmatrix}$$

Example

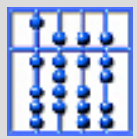


$\mathbb{R}^2$	$\mathbb{P}^2$
$l_2 : 2x - y - 0.5 = 0$	$\mathbf{l_2} = (-4.0, \quad 2.0, \quad 1.0)^T$
$l_3 : x - 3y + 1 = 0$	$\mathbf{l_3} = (1.0, \quad -3.0, \quad 1.0)^T$
$P_3 = (0.5 \quad 0.5)^T$	$\mathbf{x_3} = (0.5, \quad 0.5, \quad 1.0)^T$

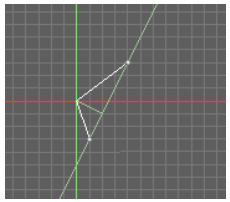
$$\begin{pmatrix} -4 \\ 2 \\ 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ -3 \\ 1 \end{pmatrix} = \begin{pmatrix} 5 \\ 5 \\ 10 \end{pmatrix}$$

- Points $\mathbf{x_1} = (x_1,y_1,w_1)^T$ and $\mathbf{x_2} = (x_2,y_2,w_2)^T$ define a line $\mathbf{l} = (a,b,c)^T$.
- $\mathbf{l}$ goes through $\mathbf{x_1}$ and $\mathbf{x_2}$: $\mathbf{x_1}^T\mathbf{l} = 0, \quad \mathbf{x_2}^T\mathbf{l} = 0$
- $\mathbf{l}$ is perpendicular to both vectors.
- $\mathbf{l}$ is cross product of $\mathbf{x_1}$ and $\mathbf{x_2}$.

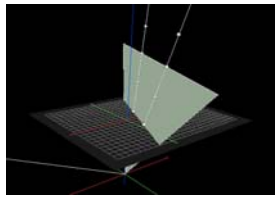
$$\mathbf{l} = \mathbf{x_1} \times \mathbf{x_2}$$



Example



$\mathbb{R}^2$	$\mathbb{P}^2$
$P_1 = (0.4, \quad 0.3)^T$	$\mathbf{x_1} = (0.4, \quad 0.3, \quad 1.0)^T$
$P_2 = (0.1, \quad -0.3)^T$	$\mathbf{x_2} = (0.1, \quad -0.3, \quad 1.0)^T$
$l_2 : 2x - y - 0.5 = 0$	$\mathbf{l_2} = (2.0, \quad -1.0, \quad -0.5)^T$



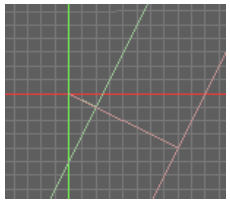
$$\begin{pmatrix} 0.4 \\ 0.3 \\ 1.0 \end{pmatrix} \times \begin{pmatrix} 0.1 \\ -0.3 \\ 1.0 \end{pmatrix} = \begin{pmatrix} 0.6 \\ -0.3 \\ -0.15 \end{pmatrix} = -6.66 \begin{pmatrix} -4 \\ 2 \\ 1 \end{pmatrix}$$

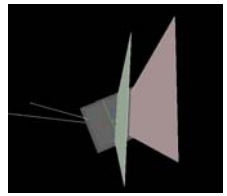
- Intersection of parallel lines $\mathbf{l}_1 = (a,b,c)^T$ and $\mathbf{l}_2 = (a,b,c')^T$

$$\mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2 = (c'-c)\begin{pmatrix} b \\ -a \\ 0 \end{pmatrix}$$

- Parallel lines intersect “at infinity”.
- *Ideal points* lie on plane w=0 (*Points at infinity*).

Example

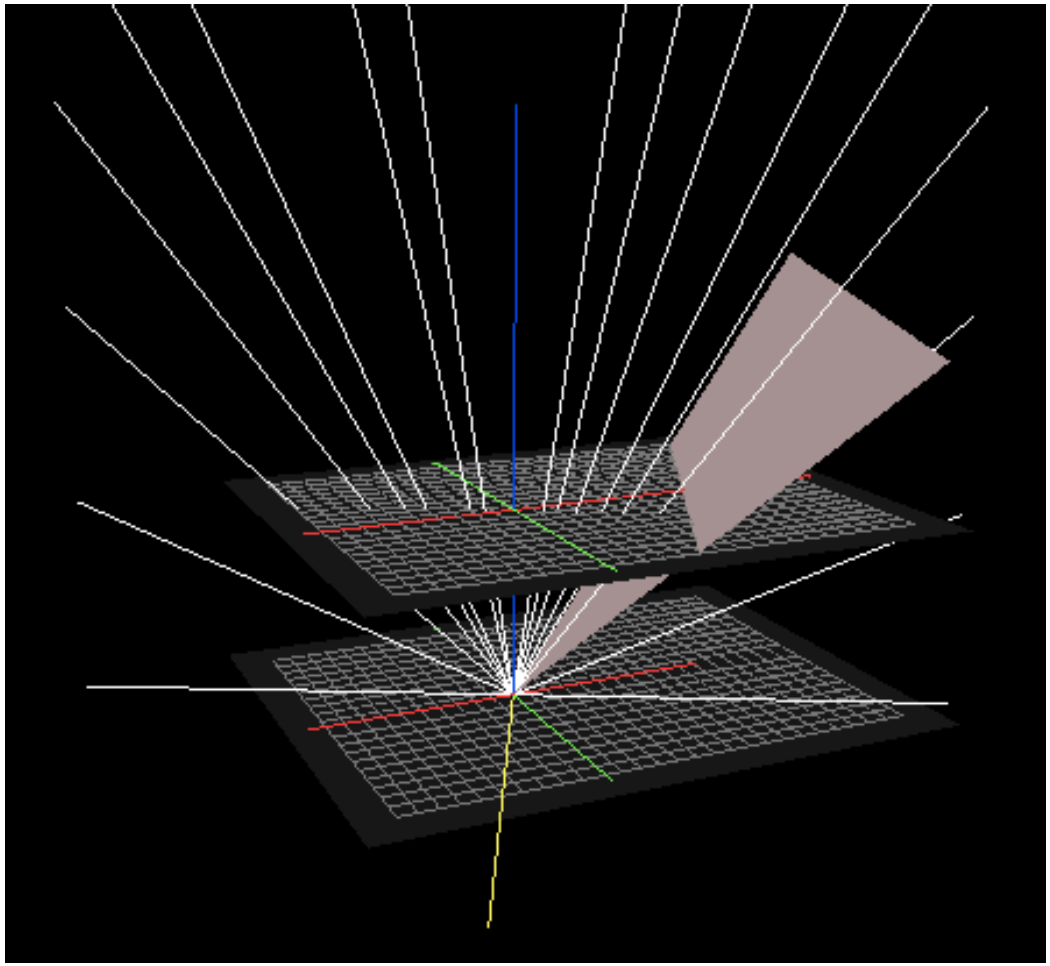




$\mathbb{R}^2$	$\mathbb{P}^2$
$l_1 : 2x - y - 2 = 0$	$\mathbf{l}_2 = (2, \quad -1, \quad -2)^T$
$l_2 : 2x - y - 0.5 = 0$	$\mathbf{l}_3 = (2, \quad -1, \quad -0.5)^T$

$$\begin{pmatrix} 2 \\ -1 \\ -2 \end{pmatrix} \times \begin{pmatrix} 2 \\ -1 \\ -0.5 \end{pmatrix} = \begin{pmatrix} -1.5 \\ -3 \\ 0 \end{pmatrix} = 1.5 \begin{pmatrix} -1 \\ -2 \\ 0 \end{pmatrix}$$

2D Projective Plane - Ideal Points -

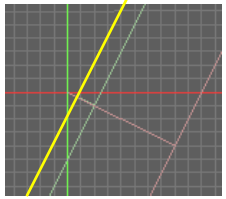


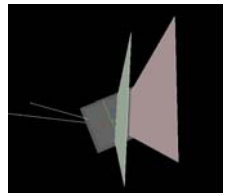
- Set of all ideal points (*points at infinity*):

$$\mathbf{x}_{\mathbf{Id\_i}} = (x_i, y_i, 0)^T$$
$$= s(x_i/y_i, 1, 0)^T$$

i.e.: all ideal points lie in plane, w = 0.

Example

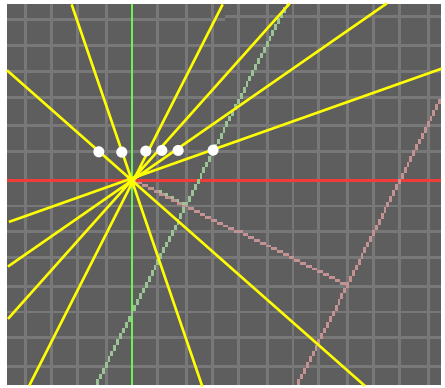




$\mathbb{R}^2$	$\mathbb{P}^2$
$l_1 : 2x - y - 2 = 0$	$\mathbf{l}_2 = (2, \quad -1, \quad -2)^T$
$l_2 : 2x - y - 0.5 = 0$	$\mathbf{l}_3 = (2, \quad -1, \quad -0.5)^T$

$$\begin{pmatrix} 2 \\ -1 \\ -2 \end{pmatrix} \times \begin{pmatrix} 2 \\ -1 \\ -0.5 \end{pmatrix} = \begin{pmatrix} -1.5 \\ -3 \\ 0 \end{pmatrix} = 1.5 \begin{pmatrix} -1 \\ -2 \\ 0 \end{pmatrix} = -3 \begin{pmatrix} 0.5 \\ 1 \\ 0 \end{pmatrix}$$

- Set of all ideal points (*points at infinity*):
  $\mathbf{x}_{Id\_i} = (x_i, y_i, 0)^T = s(x_i/y_i, 1, 0)^T$
  i.e.: all ideal points lie in plane, w = 0.
- The *line at infinity* represents all ideal points.



$$\mathbf{l}_\infty = \mathbf{x}_{Id_1} \times \mathbf{x}_{Id_2} = \begin{pmatrix} m_1 \\ 1 \\ 0 \end{pmatrix} \times \begin{pmatrix} m_2 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ m_1 - m_2 \end{pmatrix} = t \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Normal to the plane w=0.
Set of the directions of all lines in the plane.

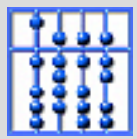
3D Transformations Using Homogeneous Coordinates

- Translation: glTranslate* (t_x, t_y, t_z)

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} x + wt_x \\ y + wt_y \\ z + wt_z \\ w \end{bmatrix}$$

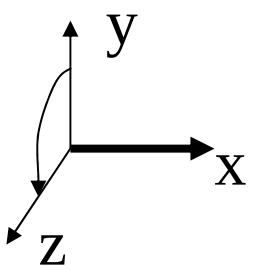
- Scaling: glScale* (s_x, s_y, s_z)

$$\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \\ s_y z \\ w \end{bmatrix}$$



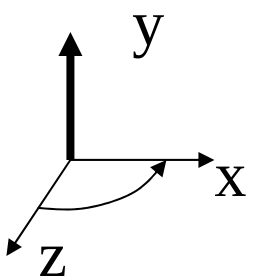
3D Transformations Using Homogeneous Coordinates

- Rotation: glRotate* (α, e_x, e_y, e_z)
  - around x



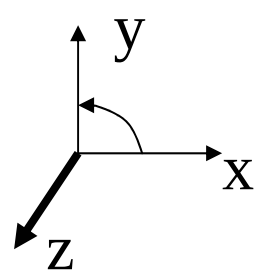
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} x \\ \cos\alpha\, y - \sin\alpha\, z \\ \sin\alpha\, y + \cos\alpha\, z \\ w \end{bmatrix}$$

  - around y



$$\begin{bmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} \cos\alpha\, x + \sin\alpha\, z \\ y \\ -\sin\alpha\, x + \cos\alpha\, z \\ w \end{bmatrix}$$

  - around z



$$\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} \cos\alpha\, x - \sin\alpha\, y \\ \sin\alpha\, x + \cos\alpha\, y \\ z \\ w \end{bmatrix}$$

Composition of Transformations

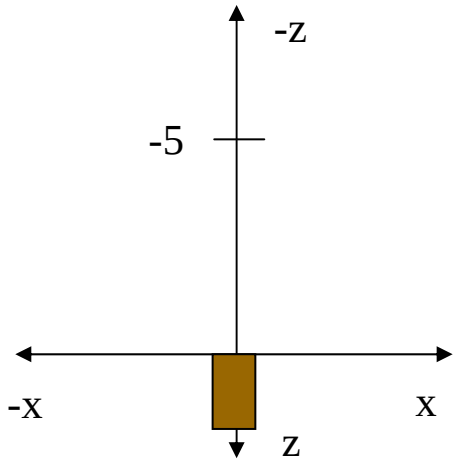
- Initialize accumulative matrix C_0 with identity matrix I
- Post-multiply accumulative matrix C_i with transformation matrix M_i according to command. $C_{i+1} := C_i M_i$
- Accumulated result represents complete transformation (independent of the number of transformation steps)
- Transformation sequence not commutative!!!

Example: translate1 - scale1 - rotate1 - translate2 - rotate2

$$I * T1 * S1 * R1 * T2 * R2 * \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix}$$

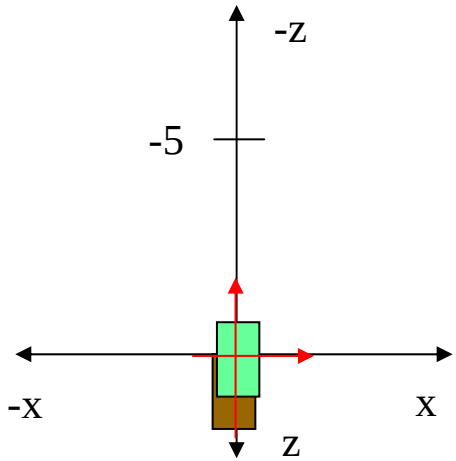
Viewing/Modelling Transformation Examples

glTranslatef (0.0, 0.0, -5.0); // along z-axis
glRotatef (45.0, 0.0, 1.0, 0.0); // around y-axis



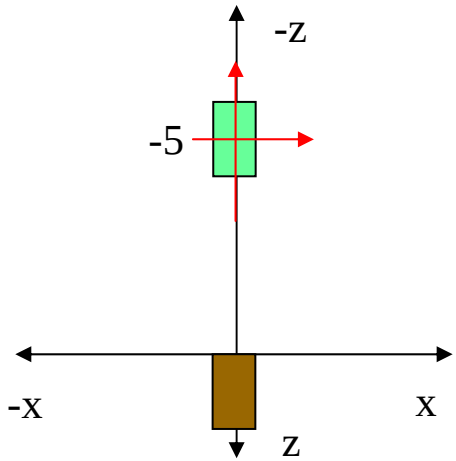
Viewing/Modelling Transformation Examples

```
glTranslatef (0.0, 0.0, -5.0);    // along z-axis
glRotatef (45.0, 0.0, 1.0, 0.0); // around y-axis
```



Viewing/Modelling Transformation Examples

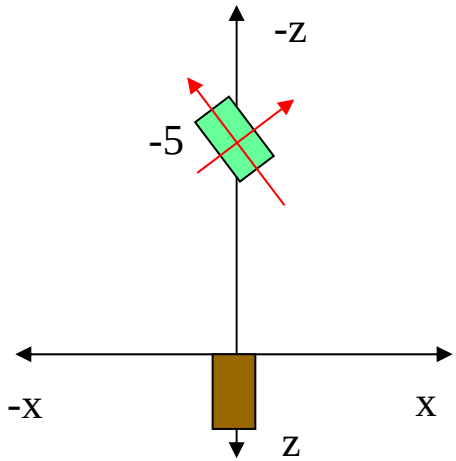
glTranslatef (0.0, 0.0, -5.0); // along z-axis
glRotatef (45.0, 0.0, 1.0, 0.0); // around y-axis



Viewing/Modelling Transformation Examples

glTranslatef (0.0, 0.0, -5.0); // along z-axis
glRotatef (45.0, 0.0, 1.0, 0.0); // around y-axis
...

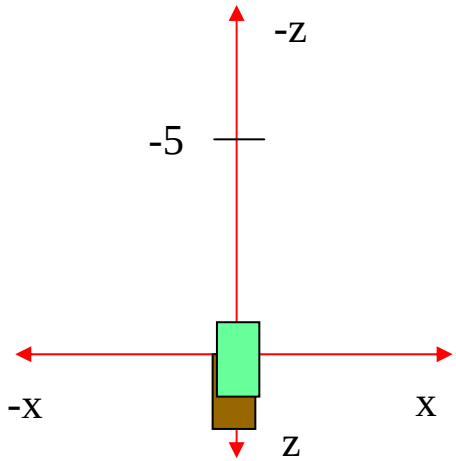
“Examine”



Viewing/Modelling Transformation Examples

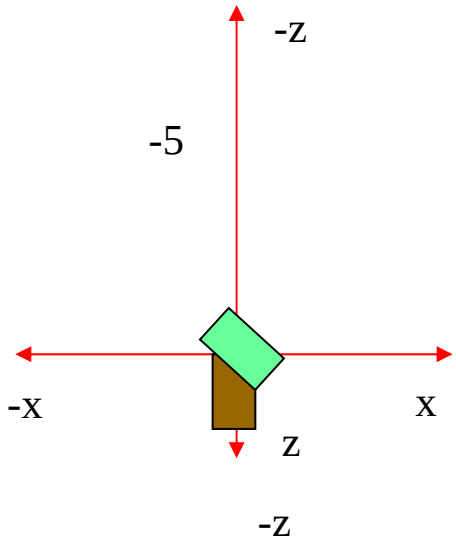
```
glTranslatef (0.0, 0.0, -5.0);    // along z-axis
glRotatef (45.0, 0.0, 1.0, 0.0); // around y-axis
```

ALTERNATIVE:
“Fly-Through”



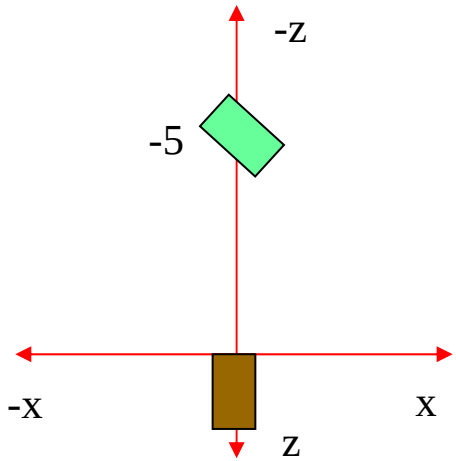
Viewing/Modelling Transformation Examples

glTranslatef (0.0, 0.0, -5.0); // along z-axis
glRotatef (45.0, 0.0, 1.0, 0.0); // around y-axis



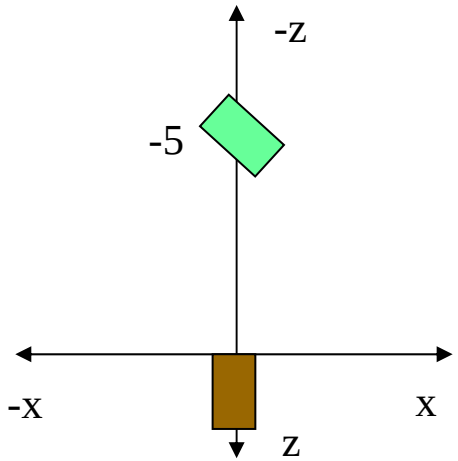
Viewing/Modelling Transformation Examples

glTranslatef (0.0, 0.0, -5.0); // along z-axis
glRotatef (45.0, 0.0, 1.0, 0.0); // around y-axis

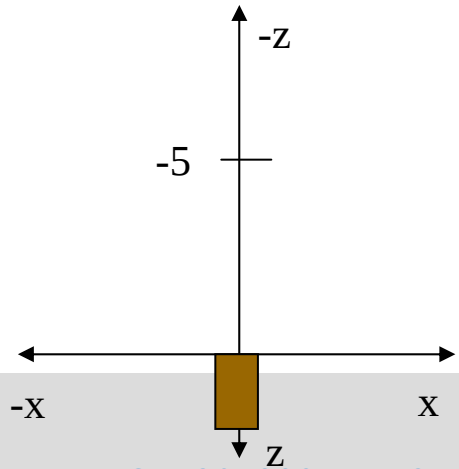


Viewing/Modelling Transformation Examples

glTranslatef (0.0, 0.0, -5.0); // along z-axis
glRotatef (45.0, 0.0, 1.0, 0.0); // around y-axis

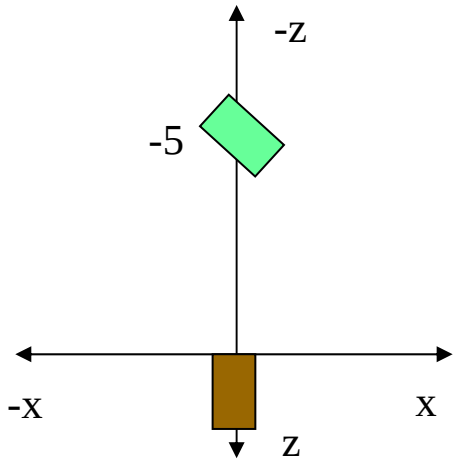


Question:
glRotatef (45.0, 0.0, 1.0, 0.0); // around y-axis
glTranslatef (0.0, 0.0, -5.0); // along z-axis

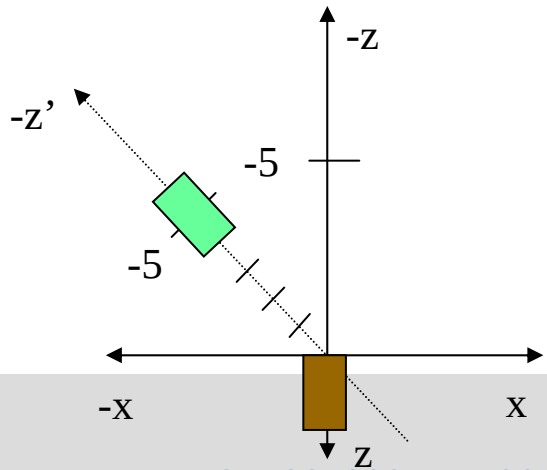


Viewing/Modelling Transformation Examples

```
glTranslatef (0.0, 0.0, -5.0);    // along z-axis
glRotatef (45.0, 0.0, 1.0, 0.0); // around y-axis
```



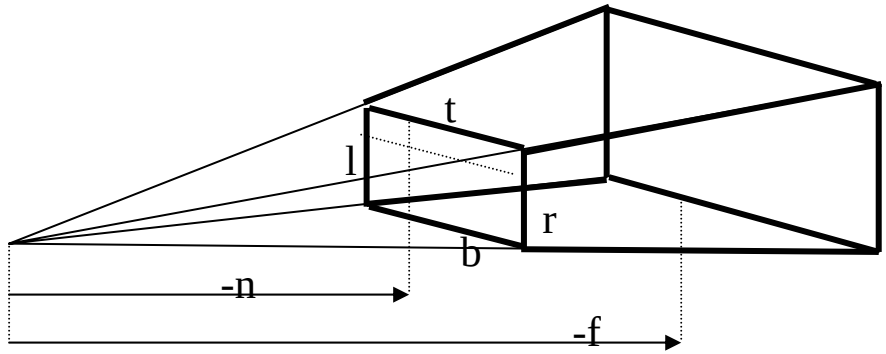

```
glRotatef (45.0, 0.0, 1.0, 0.0); // around y-axis
glTranslatef (0.0, 0.0, -5.0);    // along z-axis
```



3D Projections

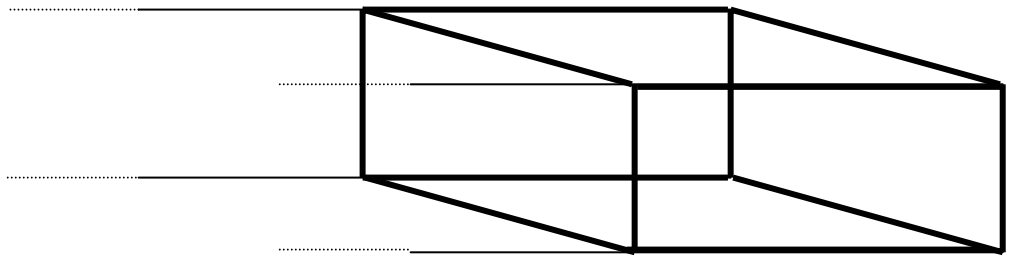
- Perspective Projection: glFrustum* (l,r,b,t,n,f)

$$\begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$



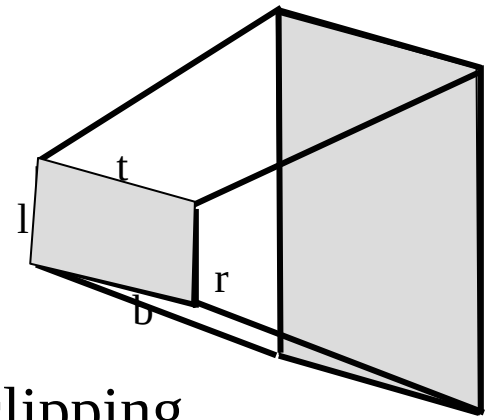
- Orthographic Projection: glOrtho* (l,r,b,t,n,f)

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



Viewport Transformation

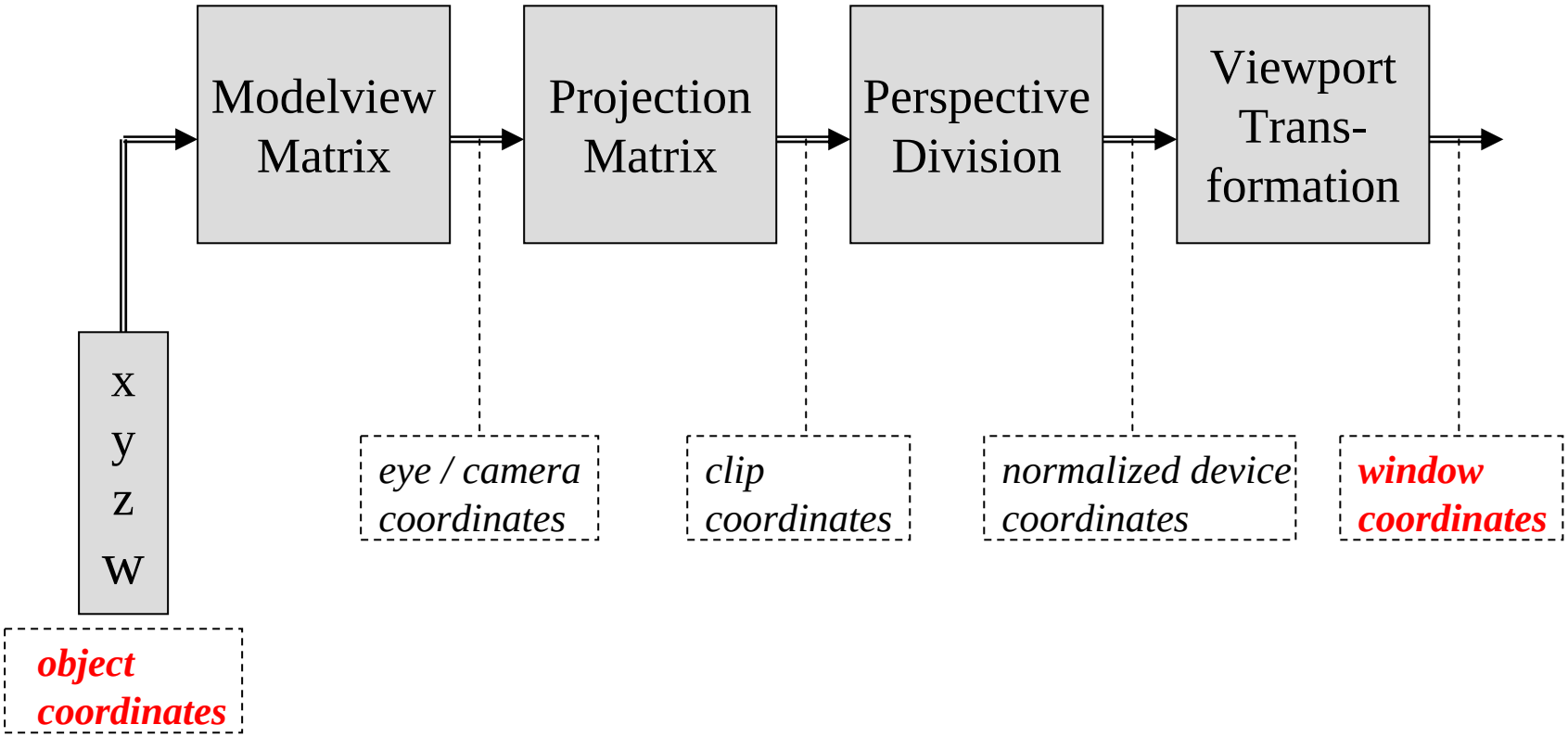
- glViewport (x, y, width, height)



window on the screen

near clipping plane of frustum

Geometry Pipeline (OpenGL)



Sample Code (Open GL)

```
// Viewport Transformations
glViewport (0.0, 0.0, w, h);

// Projection Transformations
glMatrixMode (GL_PROJECTION);
glLoadIdentity ();
glFrustum (l,r,t,b,n,f);   or gluPerspective (60.0, w/h, 1.0, 20.0);

// Viewing Transformations
glMatrixMode (GL_MODELVIEW);
glLoadIdentity ();
glTranslatef (0.0, 0.0, -5.0);  // move world away from camera
glRotatef (45.0, 0.0, 1.0, 0.0);  // rotate world in front of the camera around y-axis
…
// Modeling Transformations
...
// Draw Object
…
```