CSE4203: Computer Graphics

Chapter – 7 (part - C)

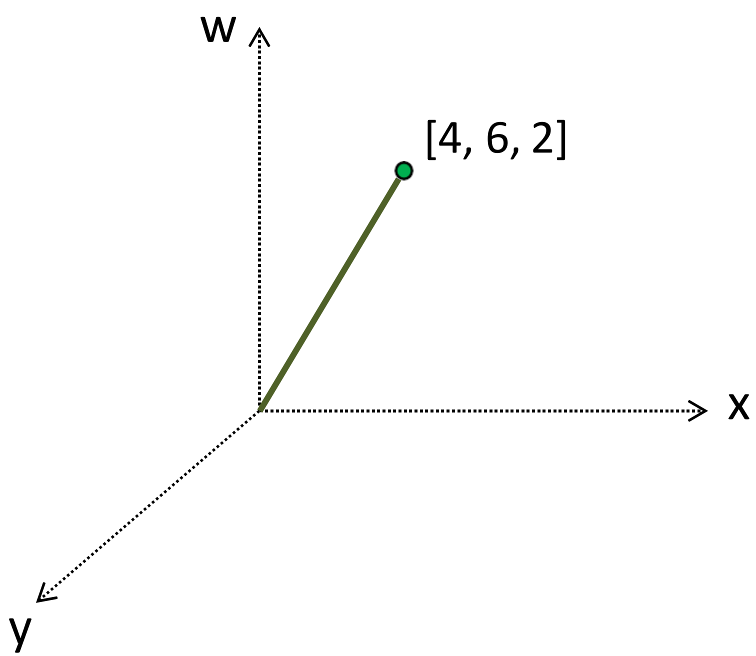
Viewing

Outline

– Perspective projection matrix

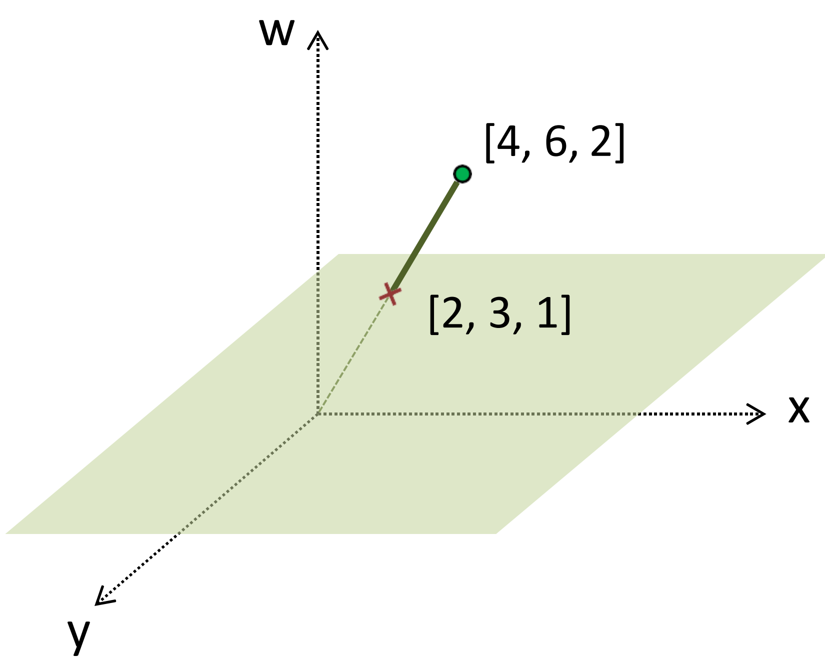
Homogeneous Coordinates (1/3)

- What is a homogeneous coordinate?
- Why do we need it?



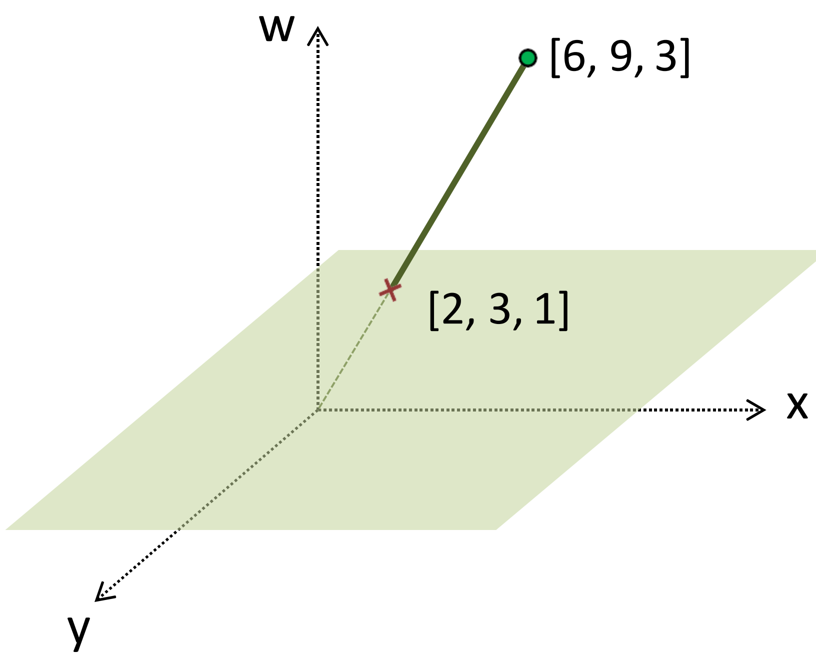
[x, y, w] ➡ [4, 6, 2]

Homogeneous Coordinates (2/3)



[x, y, w] ➡ [x/w, y/w, 1]

Homogeneous Coordinates (3/3)

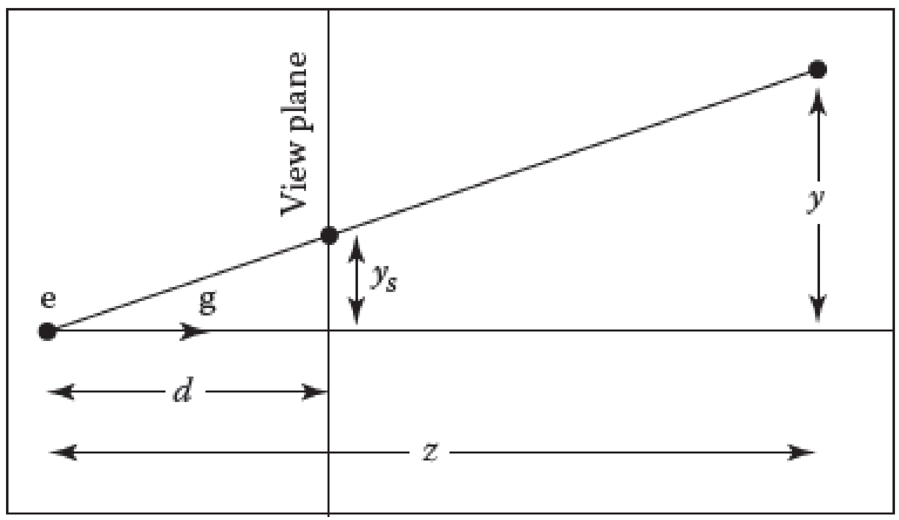


[x, y, w] ➡ [x/w, y/w, 1]

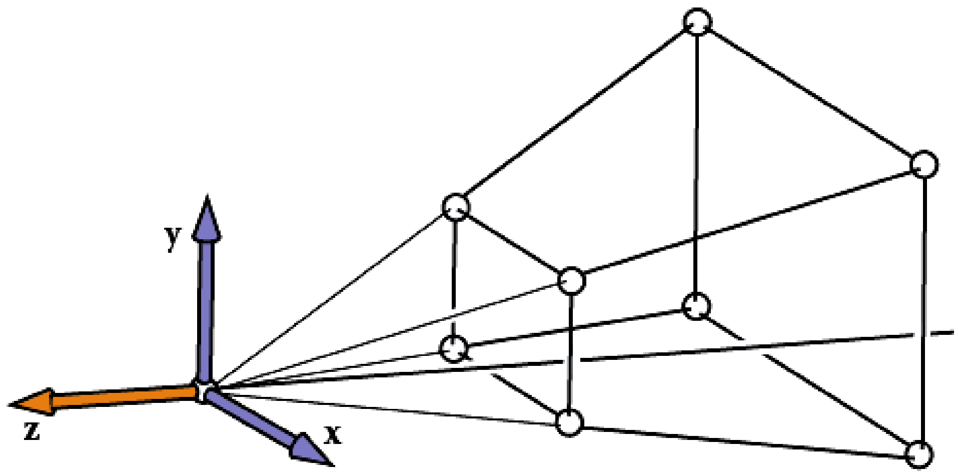
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \sim \begin{bmatrix} x/w \\ y/w \\ w/w \end{bmatrix} \sim \begin{bmatrix} x/w \\ y/w \\ 1 \end{bmatrix}$$

Key property of perspective

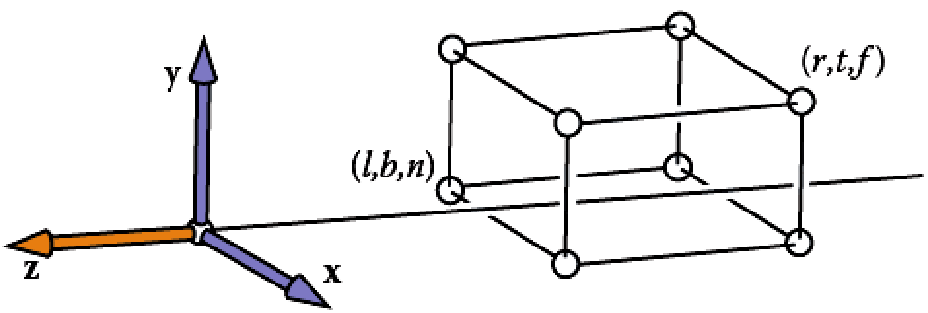
- Size of an object on the screen is proportional to 1/*z*



Perspective Projection (1/17)

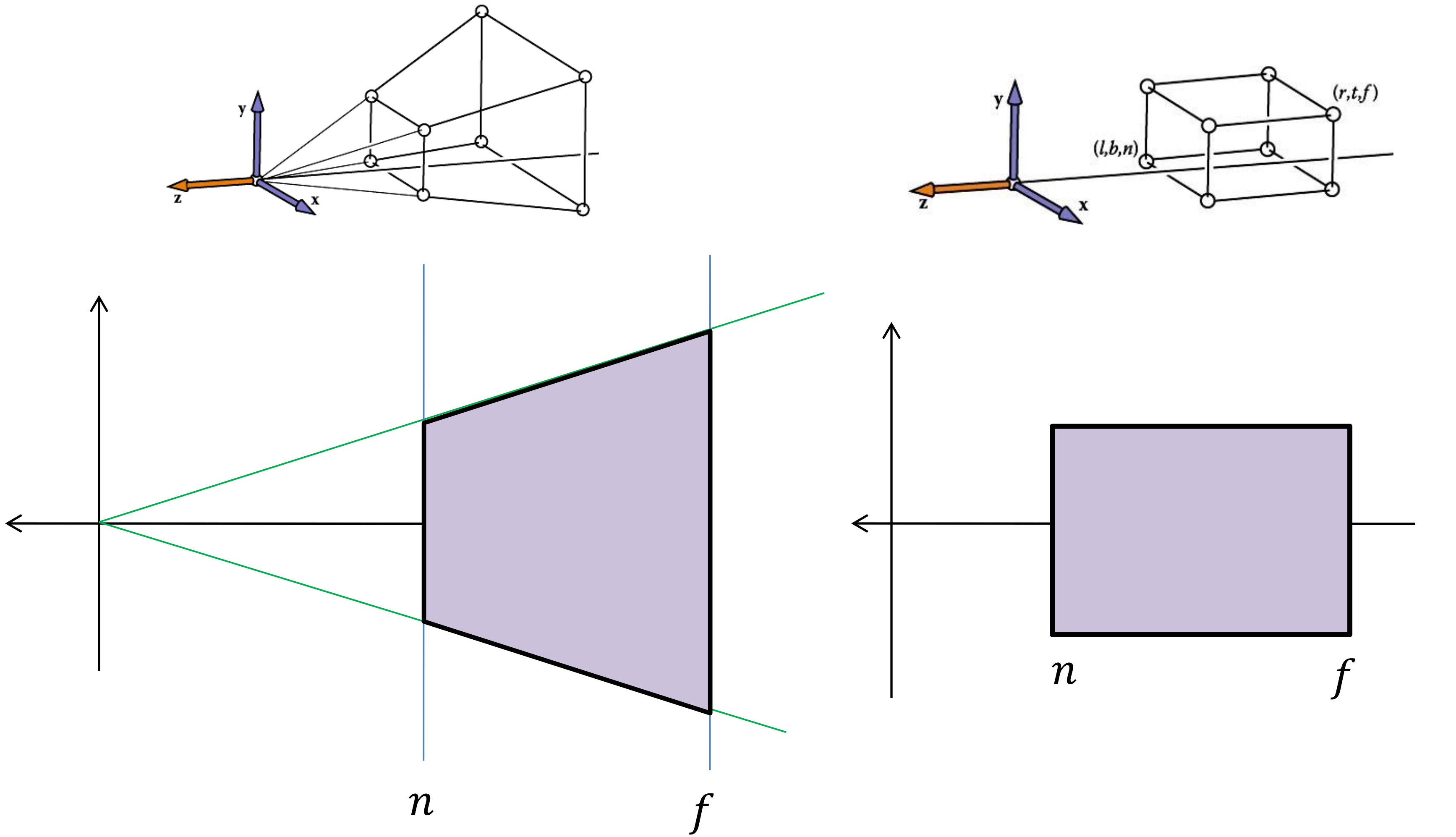


perspective view volume
(viewing frustum)

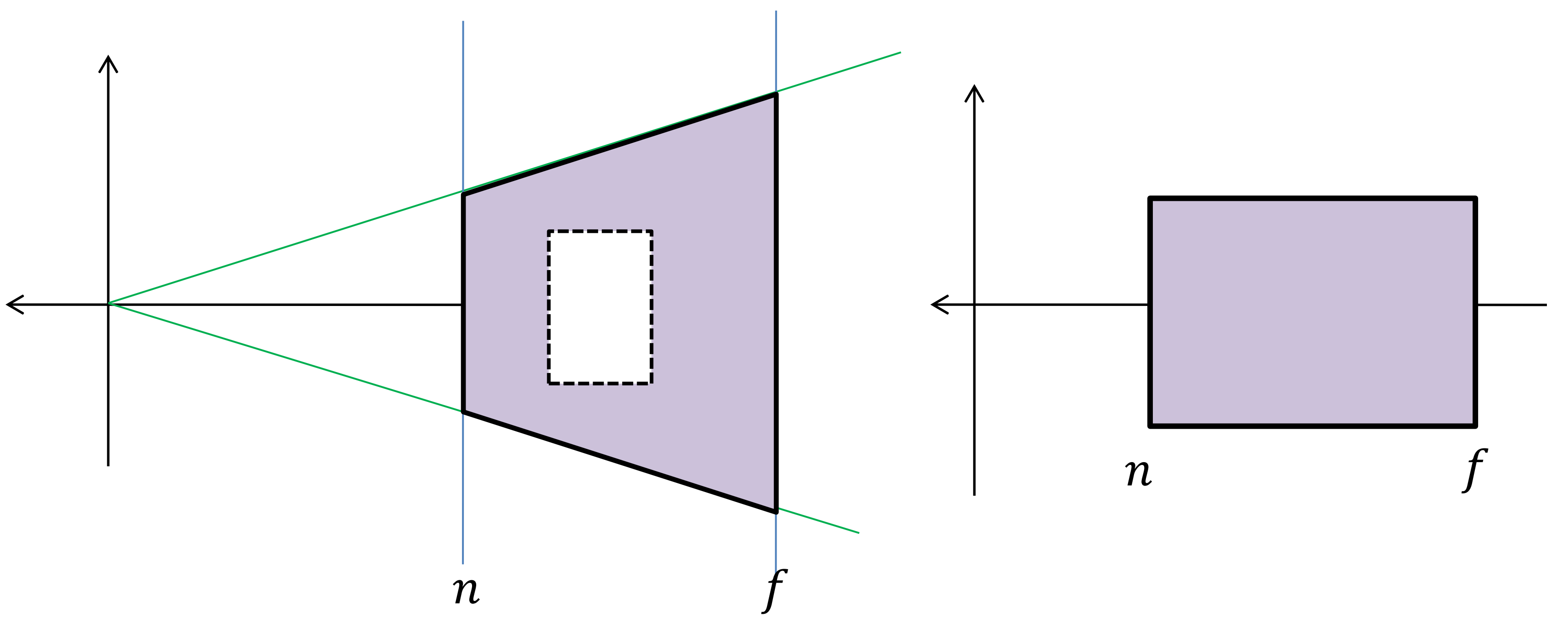


orthographic view volume

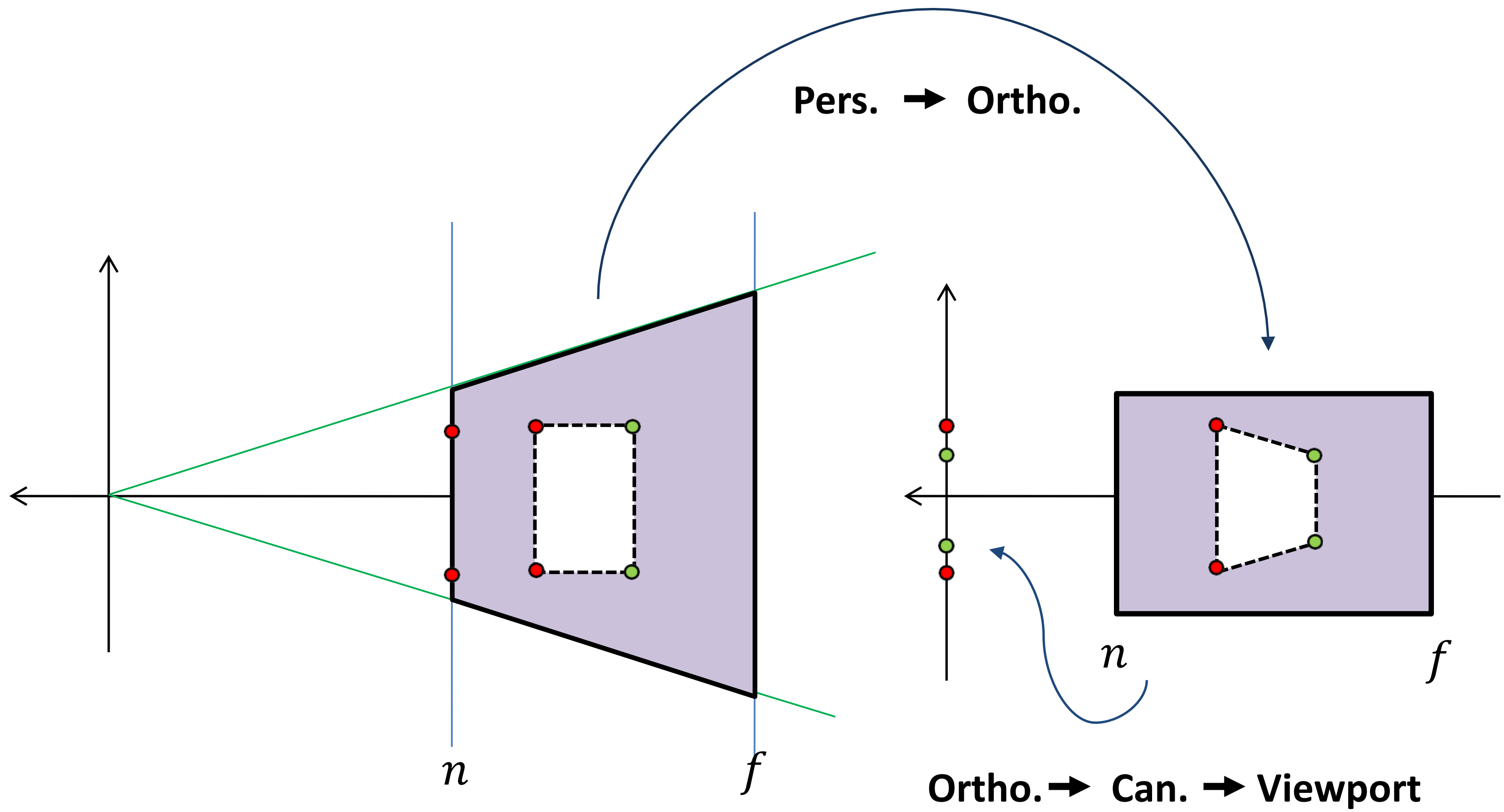
Perspective Projection (2/17)



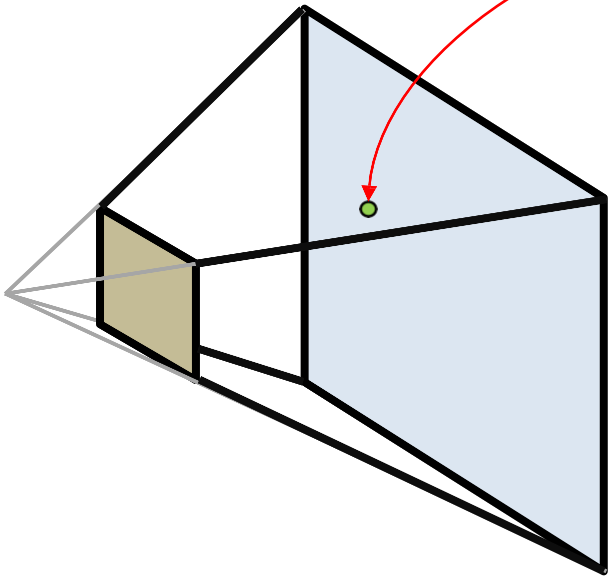
Perspective Projection (3/17)



Perspective Projection (4/17)

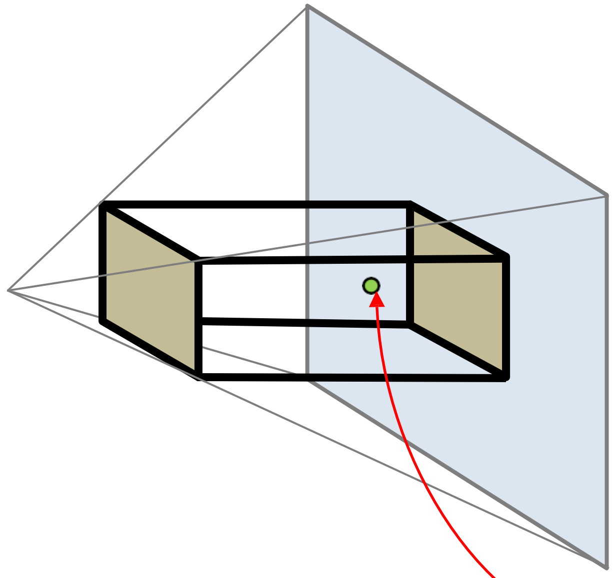


Perspective Projection (5/17)



$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Perspective Projection (6/17)



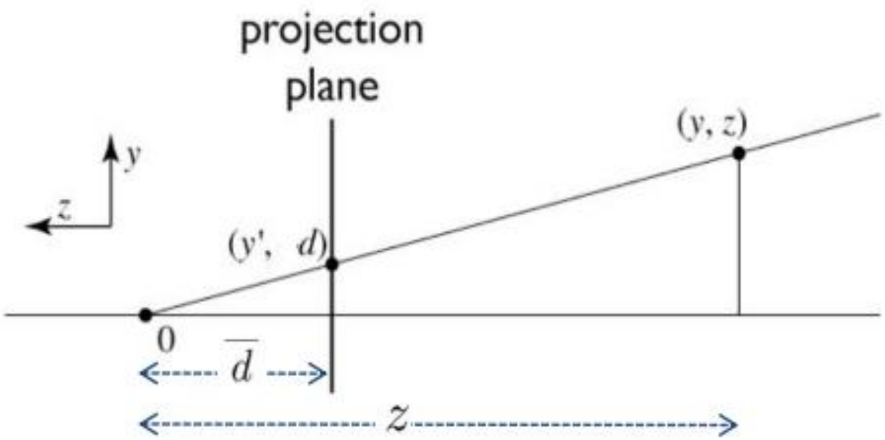
Non Affine Transformation

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \boldsymbol{M} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Perspective Projection (7/17)

<ins>*For 1D:*</ins>

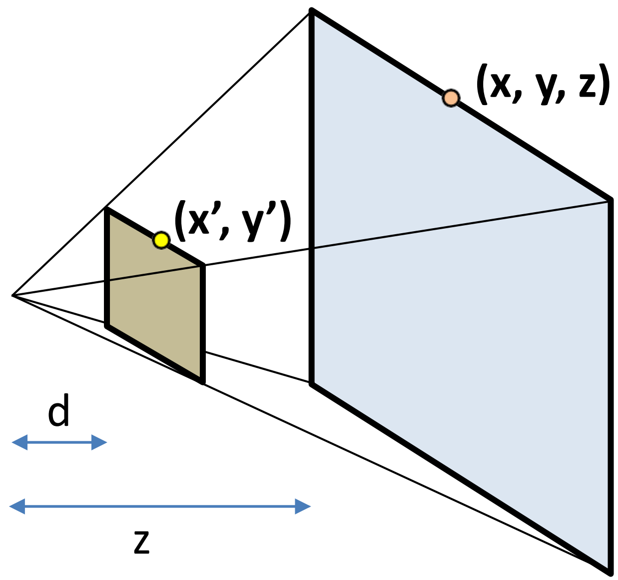
$$\frac{y'}{d} = \frac{y}{z}$$
$$y' = dy/z$$



$$\begin{bmatrix} d & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y \\ z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} dy \\ z \end{bmatrix} \sim \begin{bmatrix} dy/z \\ 1 \end{bmatrix} = \begin{bmatrix} y' \\ 1 \end{bmatrix}$$

Perspective Projection (8/17)



For 2D:

$$y' = dy/z$$

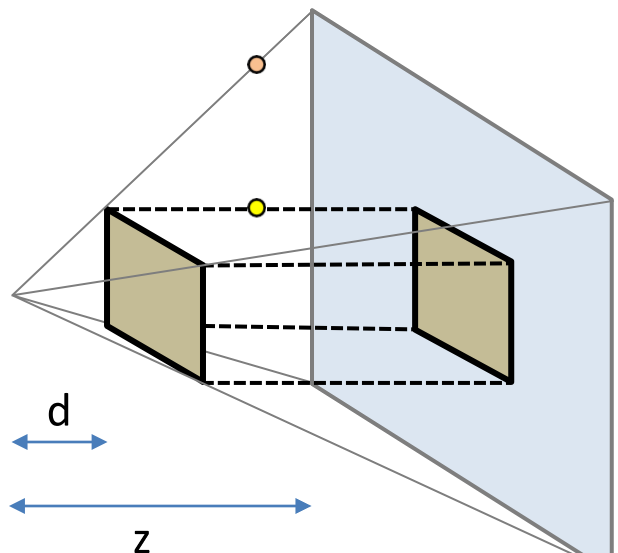
$$x' = dx/z$$

$$\begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} dx \\ dy \\ z \end{bmatrix} \sim \begin{bmatrix} dx/z \\ dy/z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Perspective Projection (9/17)

For 3D:



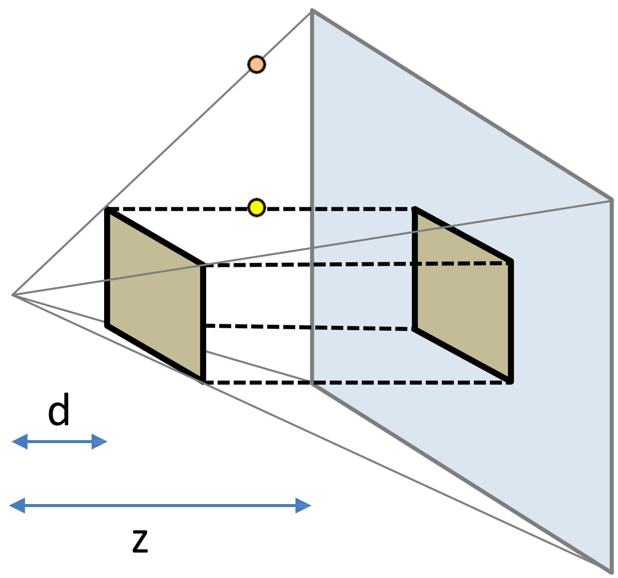
$$y' = dy/z$$

$$x' = dx/z$$

$$z' = z$$

There will be always division by z, so z'= z is not possible.

Perspective Projection (10/17)



Scaling ***z*** with ***a*** and translating it by ***b***.

For 3D:

$$y' = dy/z$$

$$x' = dx/z$$

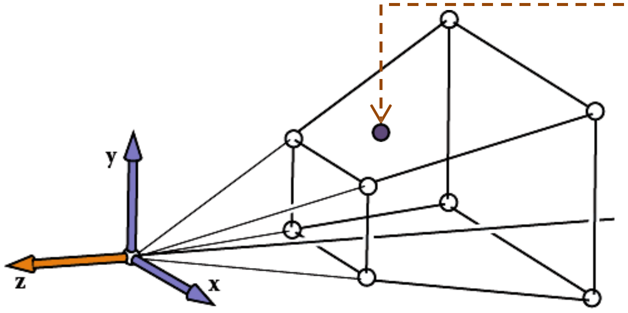
$$z' = (az+b)/z$$

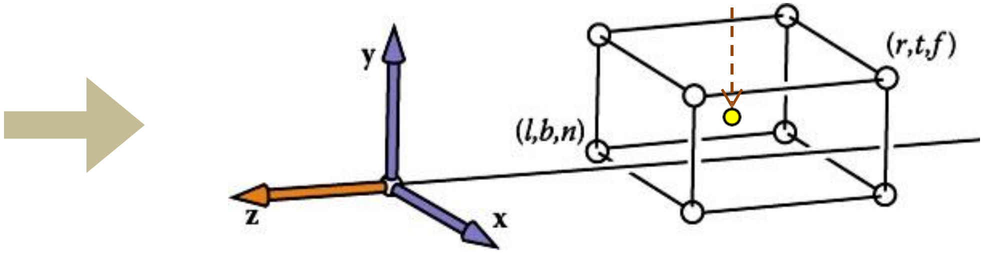
$$\begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} dx \\ dy \\ az+b \\ z \end{bmatrix} \sim \begin{bmatrix} dx/\ z \\ dy/\ z \\ (az+b)/\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

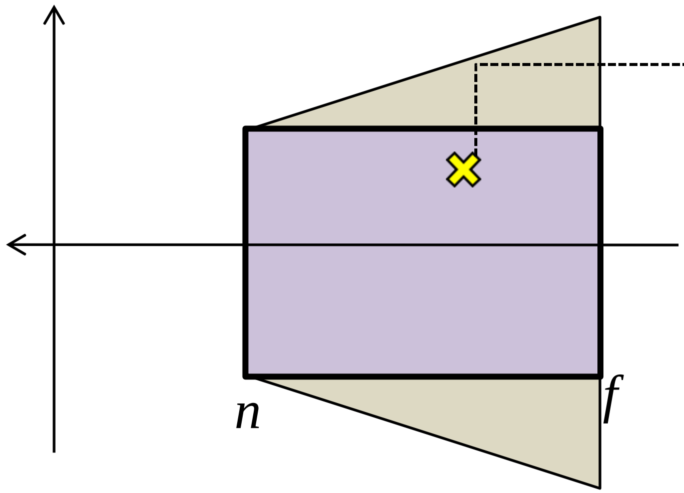
Perspective Projection (11/17)

$$\begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & 1 & 0 \end{bmatrix} \boxed{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}} = \boxed{\begin{bmatrix} dx/\ z \\ dy/\ z \\ (az+b)/\ z \\ 1 \end{bmatrix}}$$



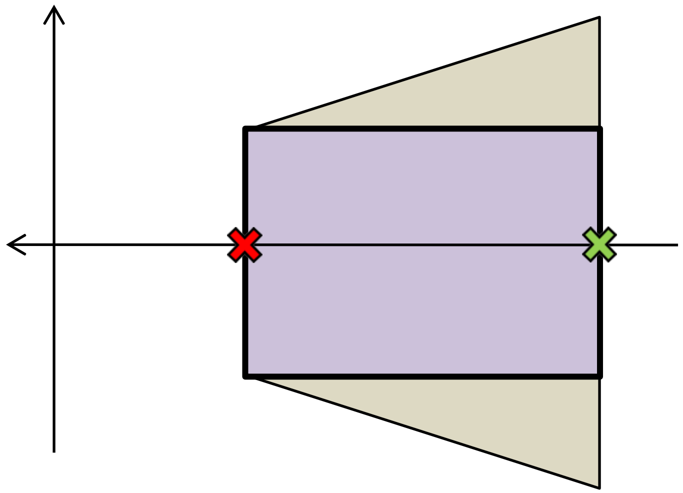


Perspective Projection (12/17)



$$\begin{bmatrix} dx/\ \ z \\ dy/\ \ z \\ (az+b)/\ \ z \\ 1 \end{bmatrix}$$

Perspective Projection (13/17)

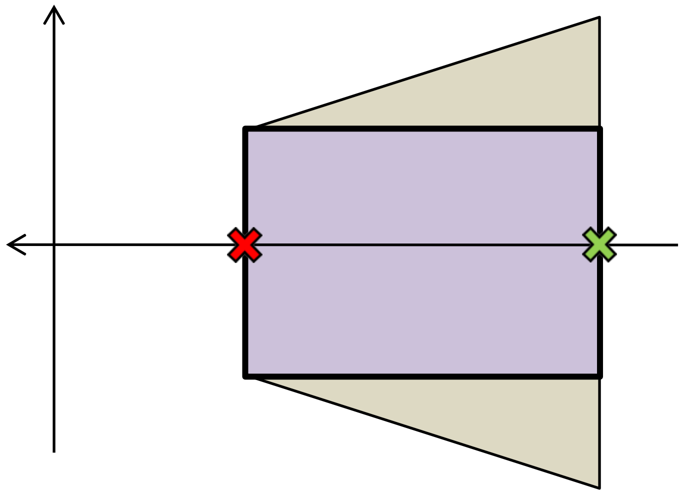


$z = n$

$z = n$

$z = f$

Perspective Projection (14/17)

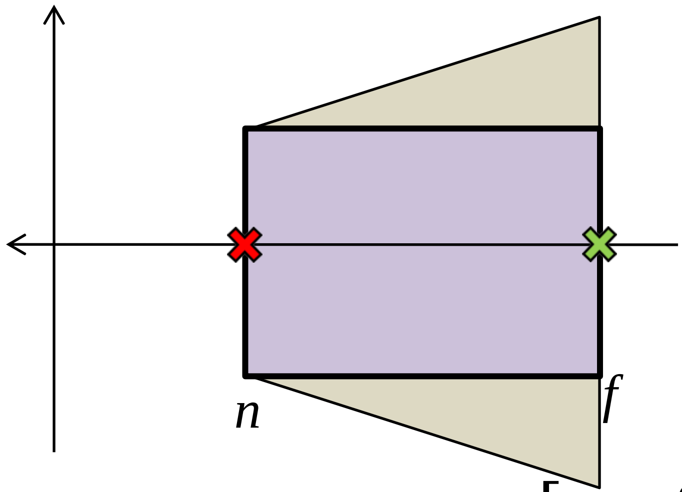


$$z = n$$

$$z = n \qquad \begin{bmatrix} 0 \\ 0 \\ n \\ 1 \end{bmatrix}$$

$$z = f \qquad \begin{bmatrix} 0 \\ 0 \\ f \\ 1 \end{bmatrix}$$

Perspective Projection (15/17)

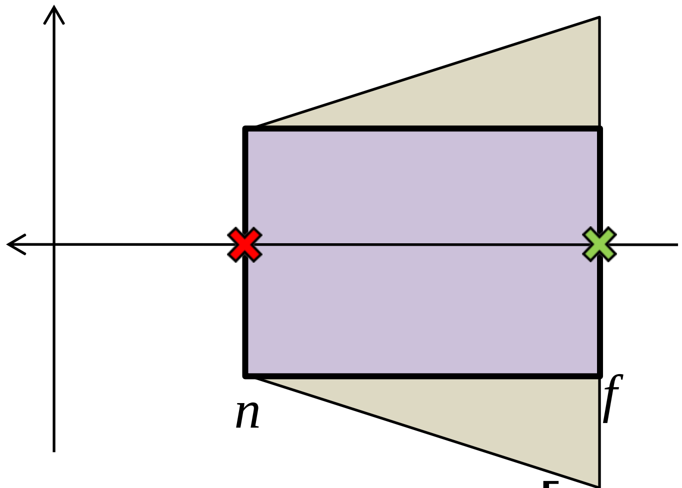


$$z = n$$

$$z = n \quad \begin{bmatrix} 0 \\ 0 \\ (an+b)/n \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ n \\ 1 \end{bmatrix}$$

$$z = f \quad \begin{bmatrix} 0 \\ 0 \\ (af+b)/f \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ f \\ 1 \end{bmatrix}$$

Perspective Projection (16/17)



$$\begin{bmatrix} dx/\ \ z \\ dy/\ \ z \\ (az+b)/\ \ z \\ 1 \end{bmatrix}$$

$a = (n+f)$

$b = \text{-}fn$

$z = n \quad \begin{bmatrix} 0 \\ 0 \\ (an+b)/n \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ n \\ 1 \end{bmatrix}$

Q: But how?

$z = f \quad \begin{bmatrix} 0 \\ 0 \\ (af+b)/f \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ f \\ 1 \end{bmatrix}$

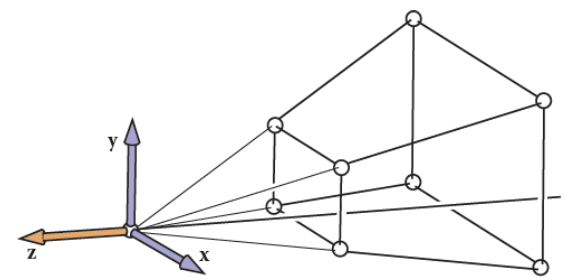
Perspective Projection (17/17)

$$P = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$a = (n+f)$
$b = -fn$
$d = ?$

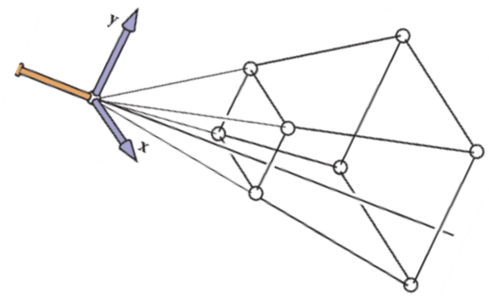
perspective matrix: $$P = \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Summary (1/6)

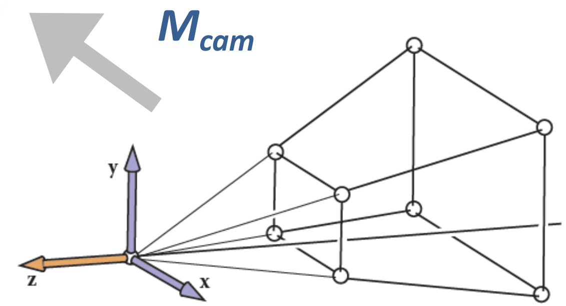


pers. view. vol.

Summary (2/6)

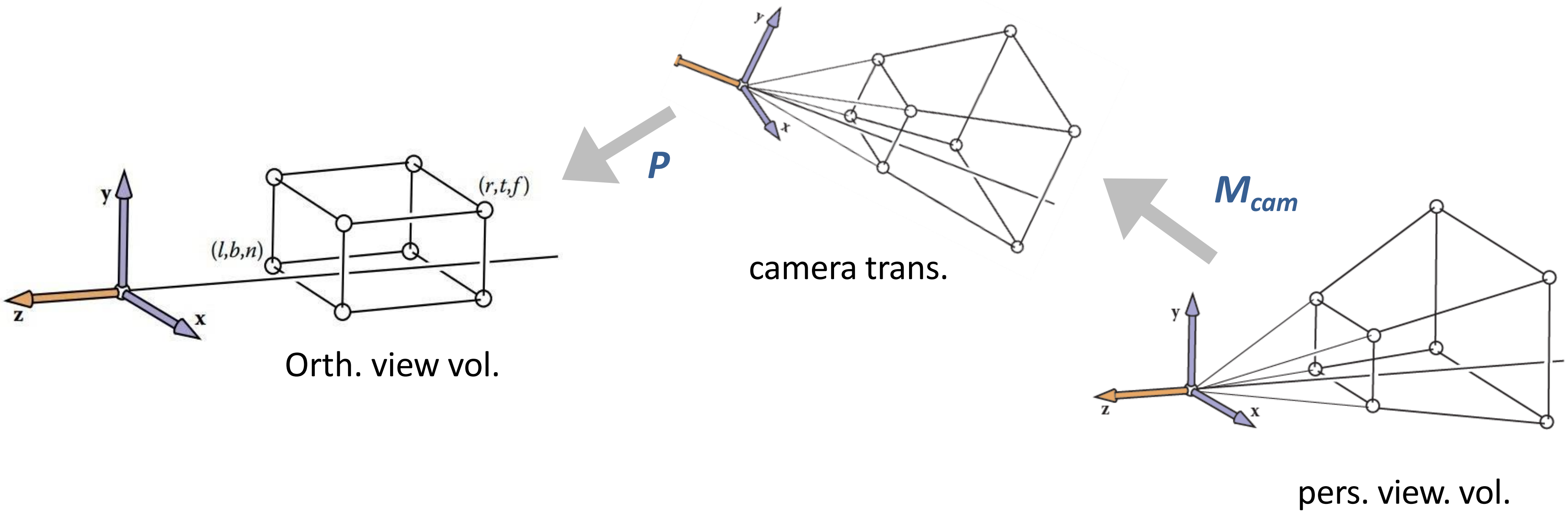


camera trans.



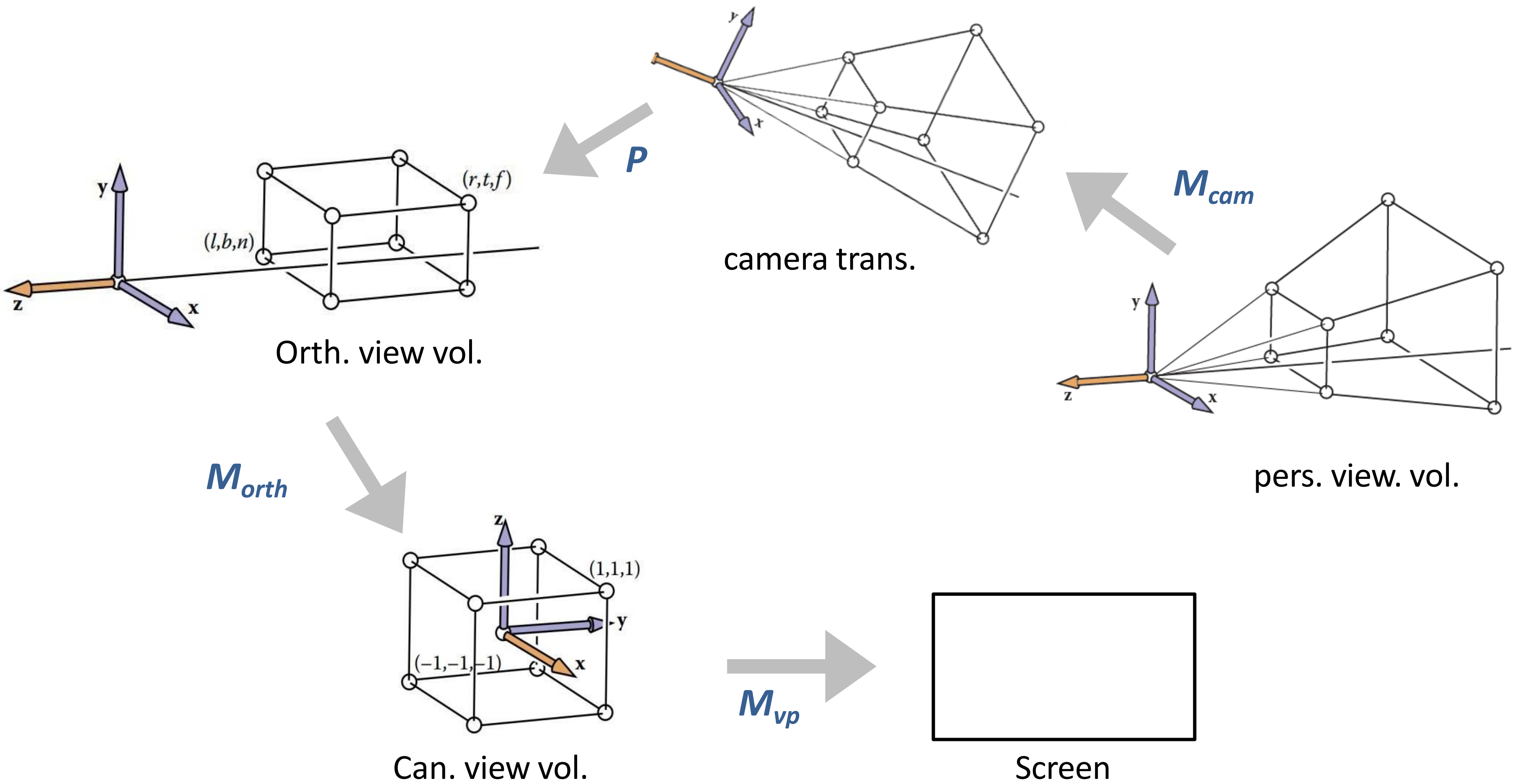
pers. view. vol.

Summary (3/6)

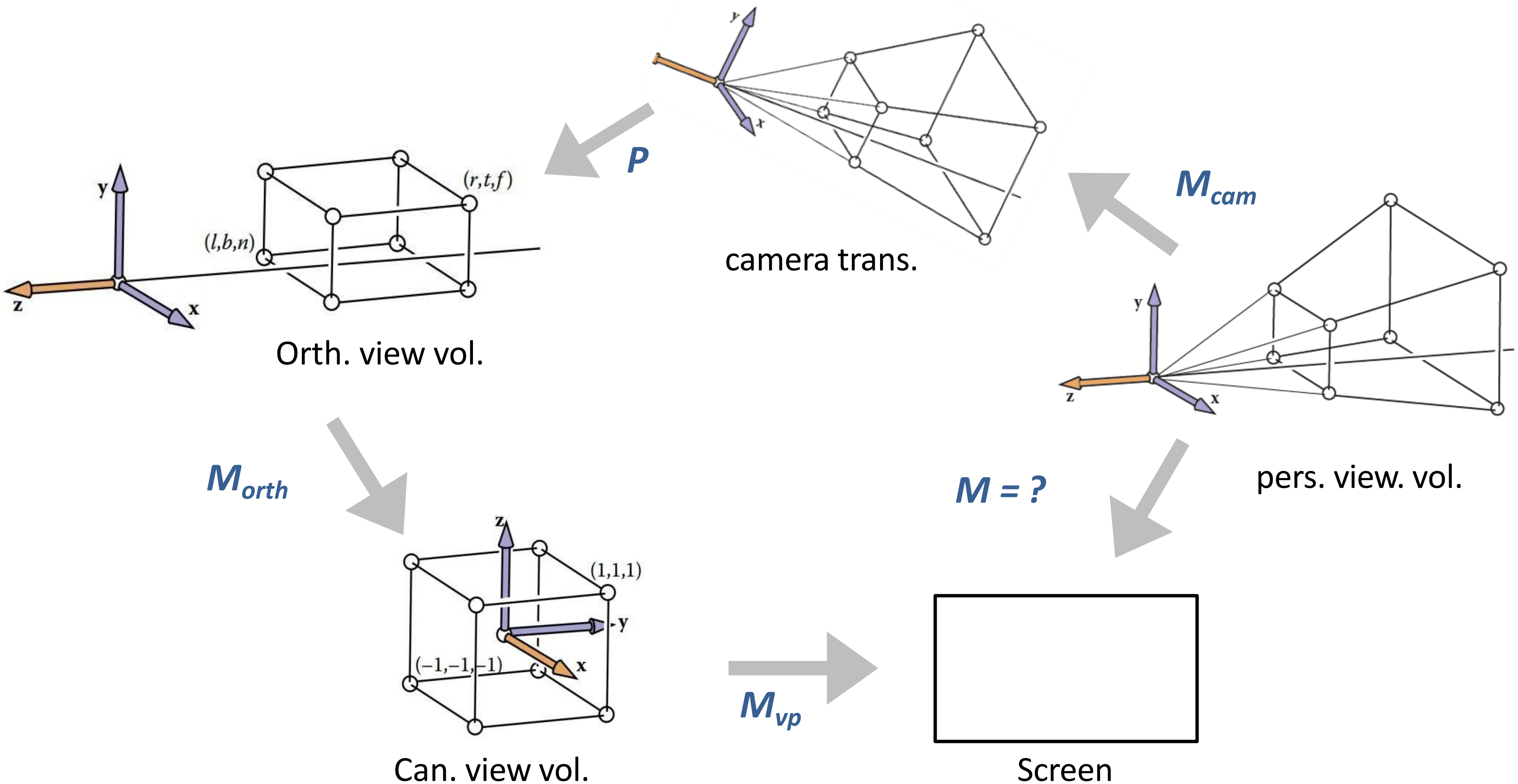


y
z
x
(l,b,n)
(r,t,f)
Orth. view vol.
P
y
x
camera trans.
M_{cam}
y
z
x
pers. view. vol.

Summary (4/6)

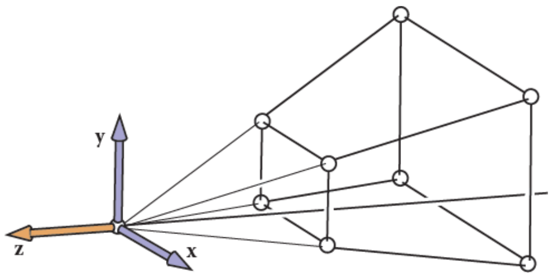

y
x
P
camera trans.
M_{cam}
y
z
x
pers. view. vol.
y
(r,t,f)
(l,b,n)
z
x
Orth. view vol.
M_{orth}
z
(1,1,1)
y
(−1,−1,−1)
x
Can. view vol.
M_{vp}
Screen

Summary (5/6)



y
x
camera trans.
P
M_{cam}
y
z
x
(r,t,f)
(l,b,n)
Orth. view vol.
y
z
x
pers. view. vol.
M_{orth}
M = ?
z
(1,1,1)
y
(−1,−1,−1)
x
Can. view vol.
M_{vp}
Screen

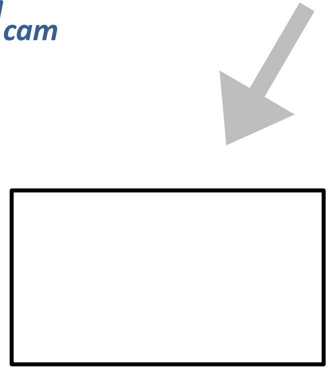
Summary (6/6)



pers. view. vol.

$$M = M_{vp} * M_{orth} * P * M_{cam}$$

$$M = M_{vp} * M_{per} * M_{cam}$$



Screen

Perspective Matrix (1/1)

$$\mathbf{M}_{\text{per}} = \mathbf{M}_{\text{orth}}\mathbf{P}$$

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

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

$$\mathbf{M}_{\text{OpenGL}} = \begin{bmatrix} \frac{2|n|}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2|n|}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{|n|+|f|}{|n|-|f|} & \frac{2|f||n|}{|n|-|f|} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Perspective Transformation Chain (1/1)

1. Modeling transform: M_m
2. Camera Transformation: M_{cam}
3. Perspective: P
4. Orthographic projection: M_{orth}
5. Viewport transform: M_{vp}

$$\mathbf{p}_s = \mathbf{M}_{\mathrm{vp}}\mathbf{M}_{\mathrm{orth}}\mathbf{P}\mathbf{M}_{\mathrm{cam}}\mathbf{M}_{\mathrm{m}}\mathbf{p}_o$$

$$\begin{bmatrix} x_s \\ y_s \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{n_x}{2} & 0 & 0 & \frac{n_x-1}{2} \\ 0 & \frac{n_y}{2} & 0 & \frac{n_y-1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2}{n-f} & -\frac{n+f}{n-f} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{M}_{\mathrm{cam}}\mathbf{M}_{\mathrm{m}} \begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix}$$

Code: *Orth. to Screen v.3* (1/2)

Drawing many 3D lines with endpoints a_i and b_i:

```
Construct M_vp
Construct M_per
Construct M_cam

M = M_vp*M_per*M_cam

for each line segment (a_i, b_i) do:
      p = M*a_i
      q = M*b_i
      drawline (x_p/W_p ,y_p/W_p, x_q/W_q, y_q/W_q)
```

Code: *Orth. to Screen v.3* (2/2)

Drawing many 3D lines with endpoints a_i and b_i:

Construct M_{vp}

Construct M

Cons

M =

$$\begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} dx \\ xy \\ az+b \\ z \end{bmatrix} \sim \begin{bmatrix} dx/\ z \\ dy/\ z \\ (az+b)/\ z \\ 1 \end{bmatrix}$$

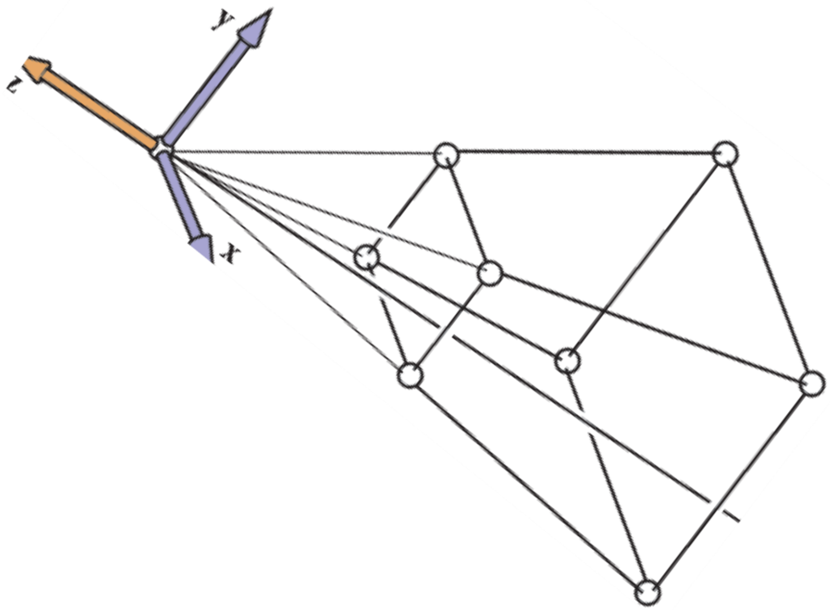
for each line segment (a_i, b_i) do:

  $p = M*a_i$

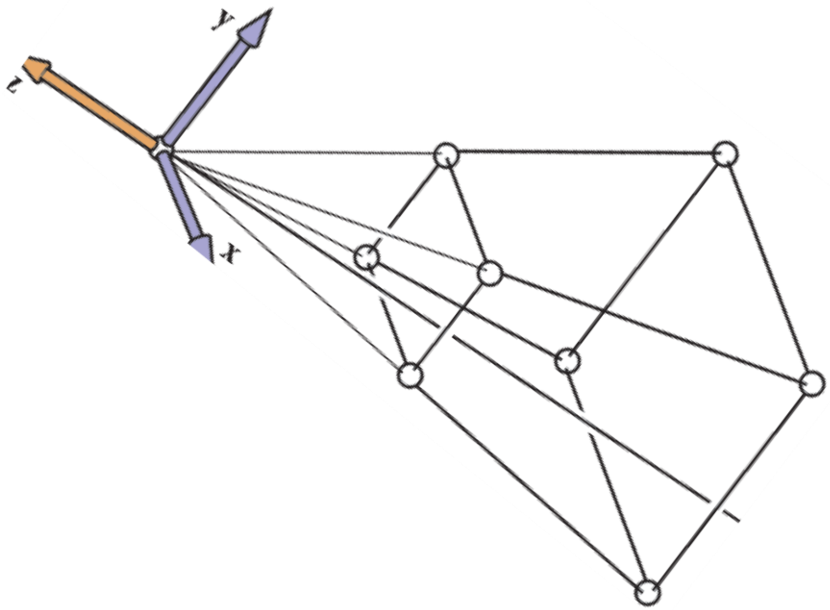
  $q = M*b_i$

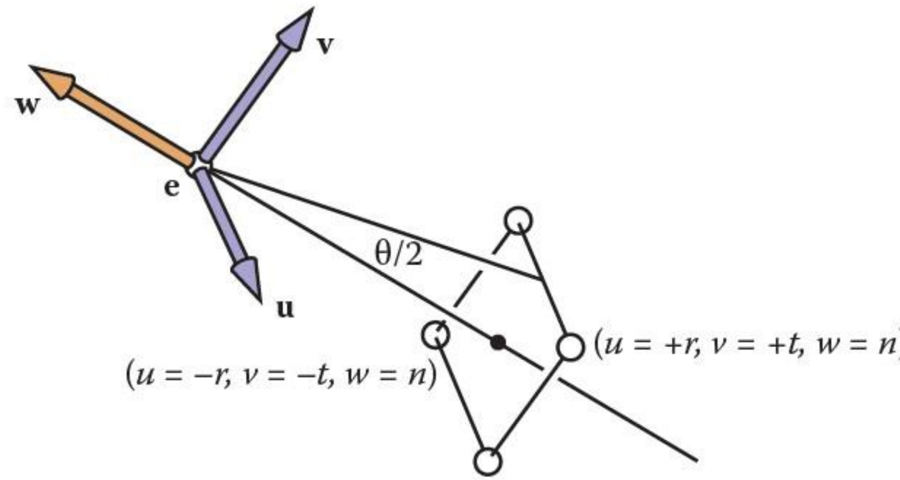
  drawline $(x_p/W_p\ , y_p/W_p,\ x_q/W_q,\ y_q/W_q)$

Field-of-View (1/6)

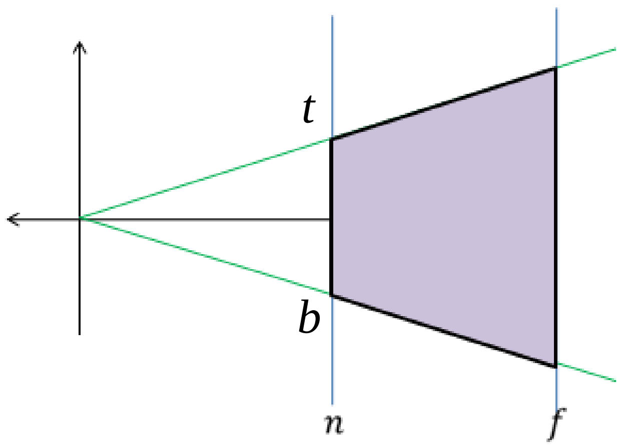


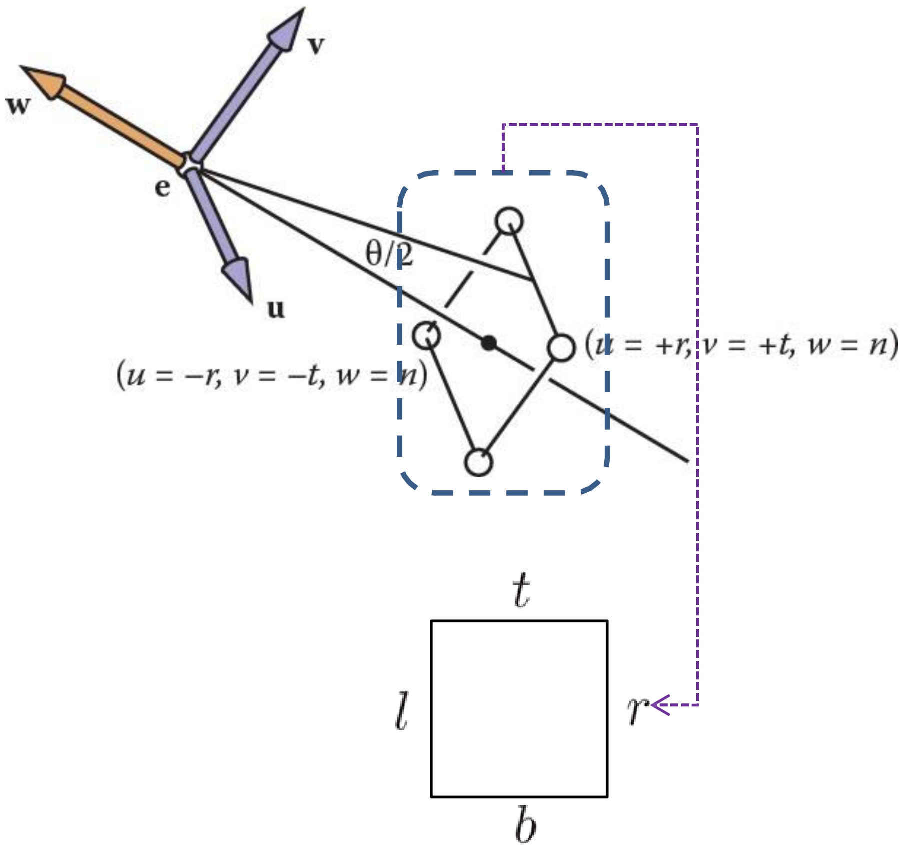
Field-of-View (2/6)



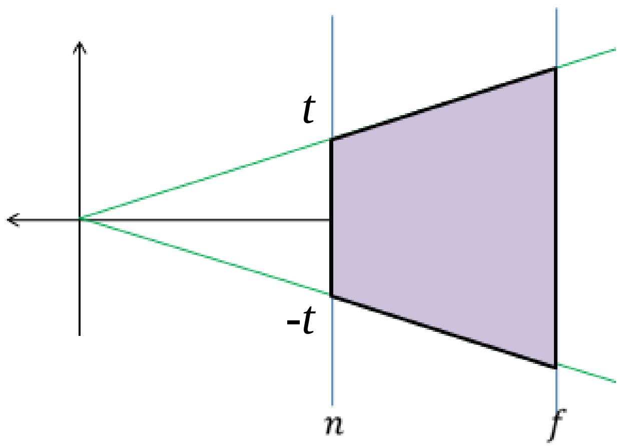


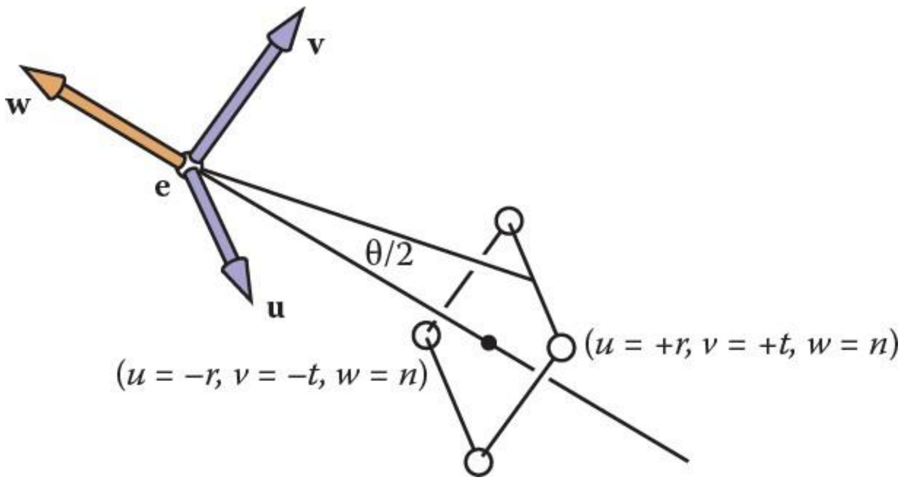
Field-of-View (3/6)



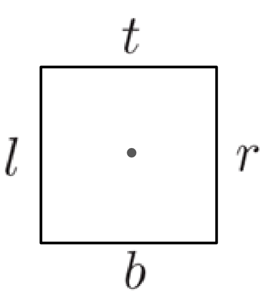


Field-of-View (4/6)

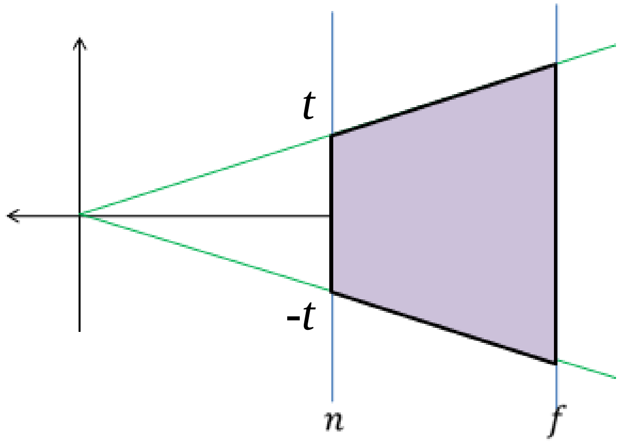


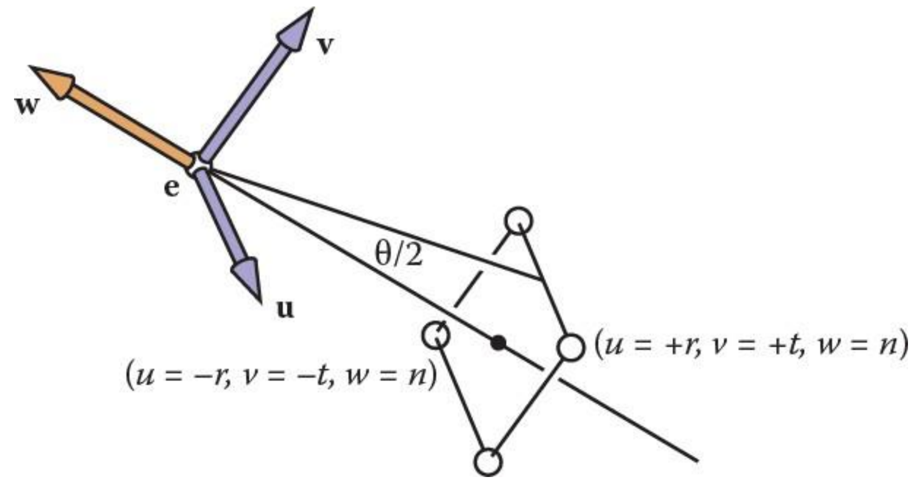


$$l = -r,$$
$$b = -t.$$



Field-of-View (5/6)

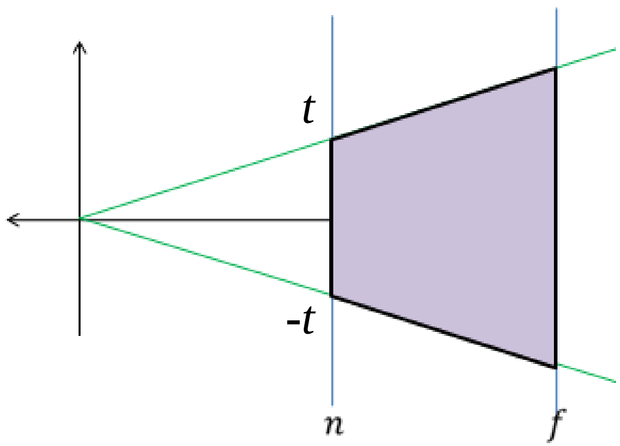


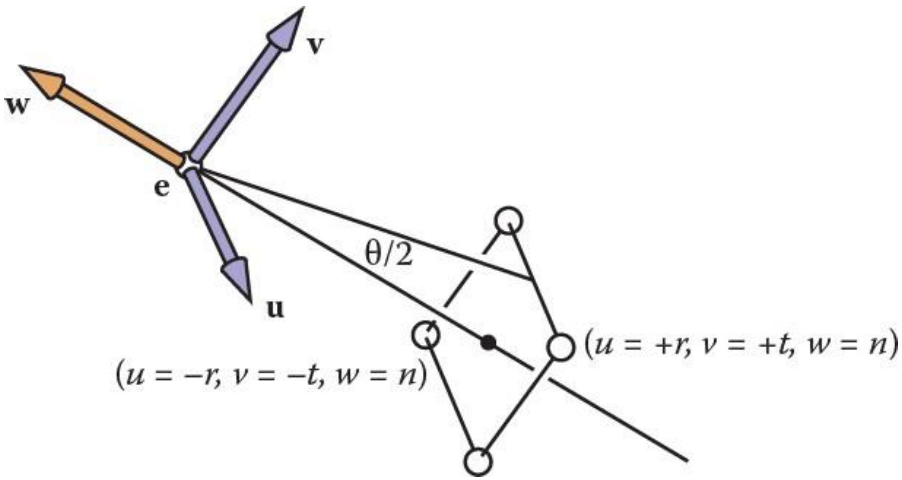


$$l = -r,$$
$$b = -t.$$

$$\tan\frac{\theta}{2} = \frac{t}{|n|}$$

Field-of-View (6/6)





$$l = -r,$$
$$b = -t.$$

$$\tan \frac{\theta}{2} = \frac{t}{|n|}$$

Field-of-View (FoV)

Practice Problem (1/2)

- Show that, the M_{OpenGL} can be written as follows –

$$\mathbf{M}_{\mathrm{OpenGL}} = \begin{bmatrix} \frac{2|n|}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2|n|}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{|n|+|f|}{|n|-|f|} & \frac{2|f||n|}{|n|-|f|} \\ 0 & 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{aspect * \tan(\frac{fov}{2})} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan(\frac{fov}{2})} & 0 & 0 \\ 0 & 0 & -\frac{far+near}{far-near} & -\frac{2*far*near}{far-near} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

here, aspect = ratio of the width to the height of the view. vol. = ?

Practice Problem (2/2)

- Derive all the matrices (using your own words):
  - *a)* M_{vp}
  - *b)* M_{orth}
  - *c)* M_{cam}
  - *d)* P and M_{per}

- Rotate a camera by -45 degree along x-axis with the eye position at 0, 0.5, -4. For a point $P_{xyz} \equiv (0, 0, 3)$, $P_{uvw} \equiv$?
- Exercise:
  - 1, 7, 8, 10