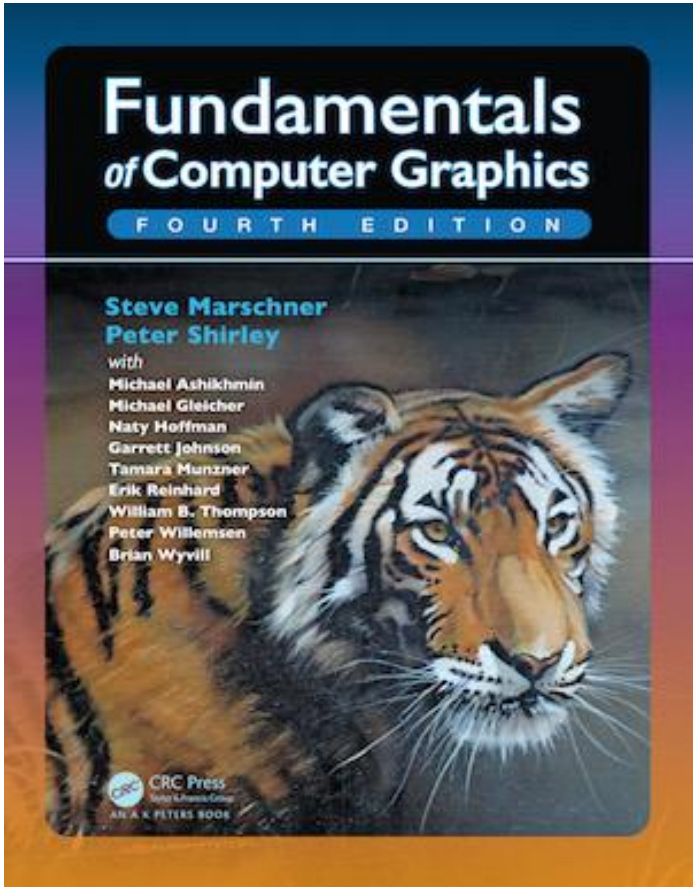
CSE4203: Computer Graphics
Chapter – 7 (part - B)
Viewing

Mohammad Imrul Jubair

Outline

– Coordinate Transformation
– Camera Transformation

Credit



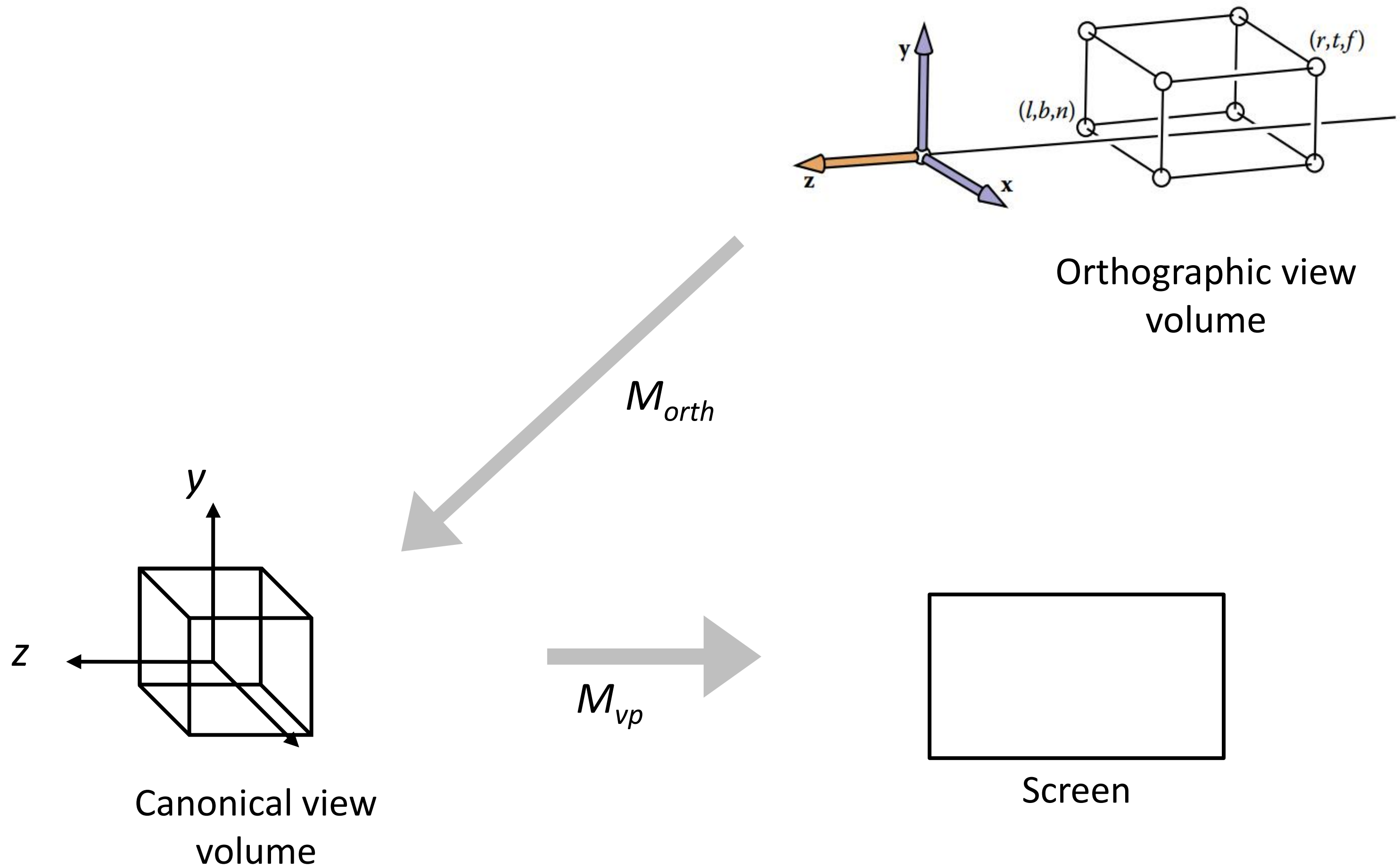
CS4620: Introduction to Computer Graphics

Cornell University

Instructor: Steve Marschner

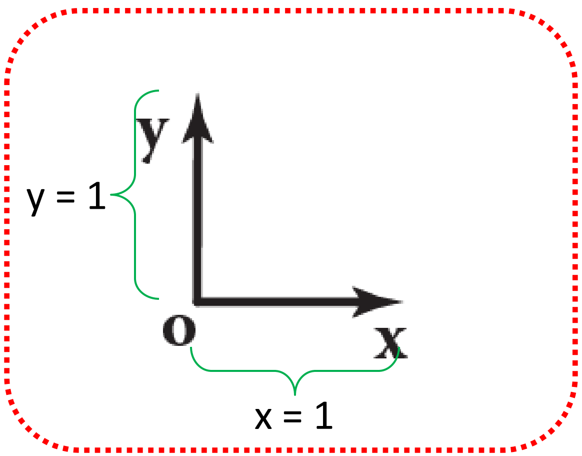
http://www.cs.cornell.edu/courses/cs4620/2019fa/

Recap (1/1)


y
z
x
(l,b,n)
(r,t,f)
Orthographic view volume
M_{orth}
y
z
Canonical view volume
M_{vp}
Screen

Coordinate System (1/1)

- A coordinate system, or coordinate frame, consists of **an origin** and **a basis**: *a set of three (two for 2D) orthonormal vectors.*



2D example

Canonical coordinate system:

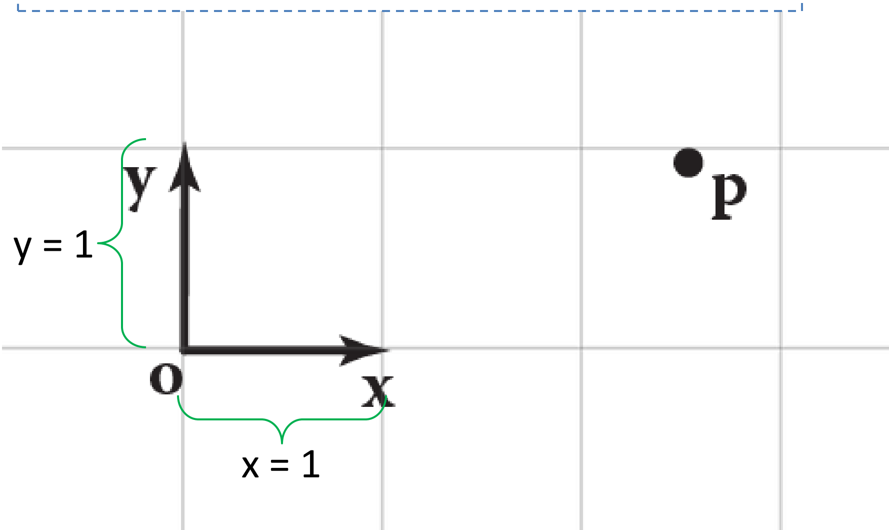
- origin **o**
- orthonormal basis vectors **{x, y}**.
- Also called: *World* coordinates

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Coordinate Transformation (1/20)

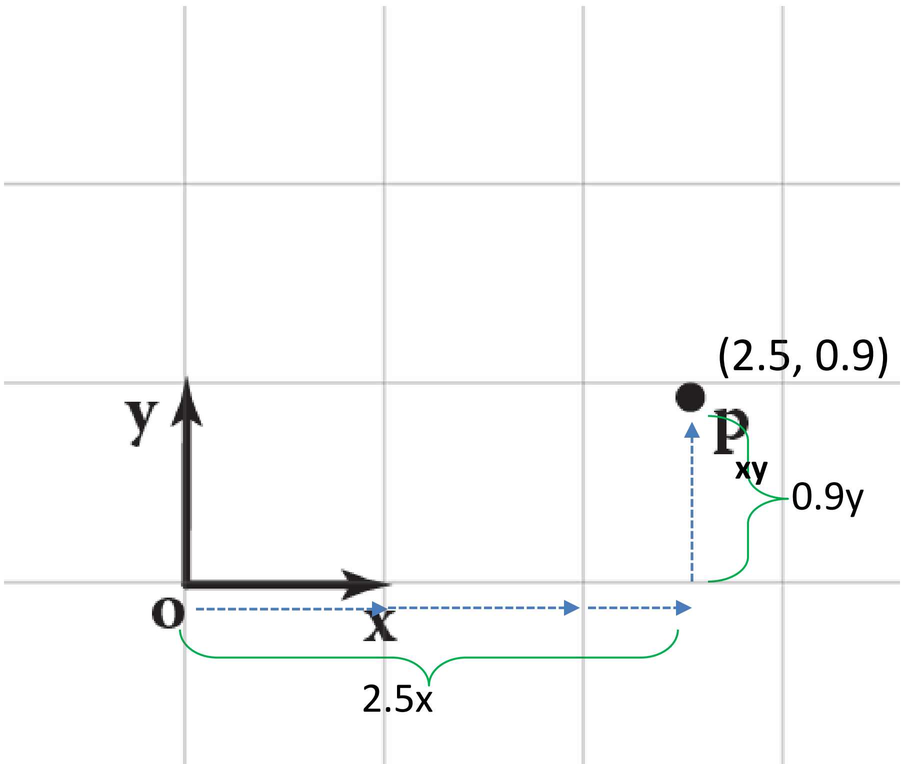
In a frame with origin **o** and basis {**x**, **y**}, the coordinates (*x*, *y*) describe the point:

$$\mathbf{o} + x\mathbf{x} + y\mathbf{y}$$



Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Coordinate Transformation (4/20)



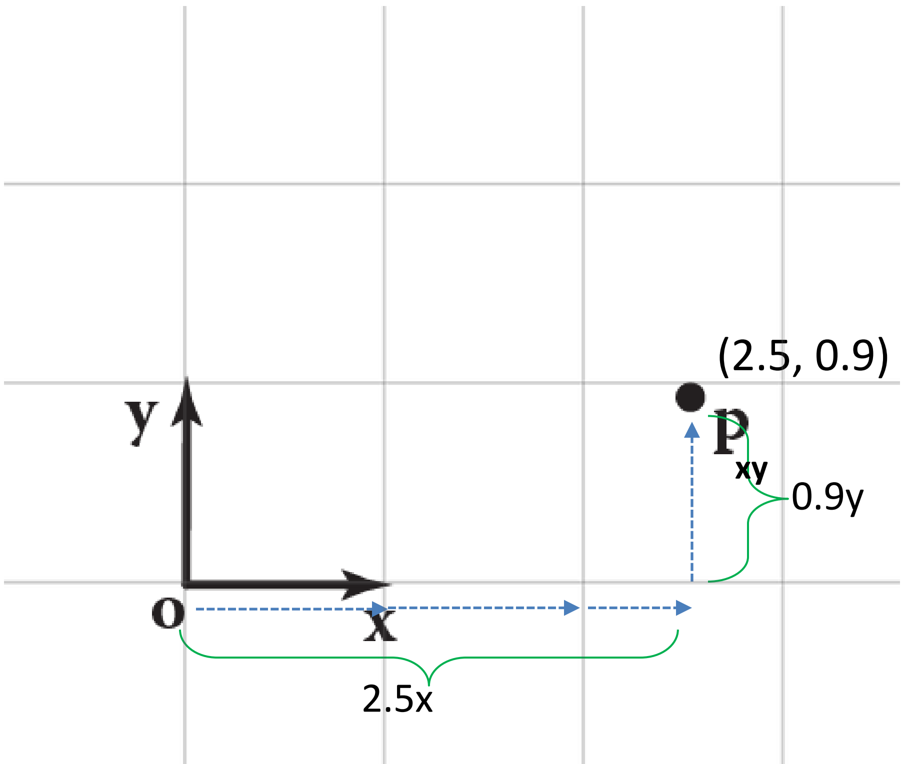
$$\mathbf{p_{xy}} = (x_p, y_p) \equiv \mathbf{o} + x_p\,\mathbf{x} + y_p\,\mathbf{y}$$

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} ? \\ ? \end{bmatrix} + x_p \begin{bmatrix} ? \\ ? \end{bmatrix} + y_p \begin{bmatrix} ? \\ ? \end{bmatrix}$$

$$= \begin{bmatrix} 2.5 \\ 0.9 \end{bmatrix}$$

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Coordinate Transformation (5/20)



$$\mathbf{p_{xy}} = (x_p, y_p) \equiv \mathbf{o} + x_p\,\mathbf{x} + y_p\,\mathbf{y}$$

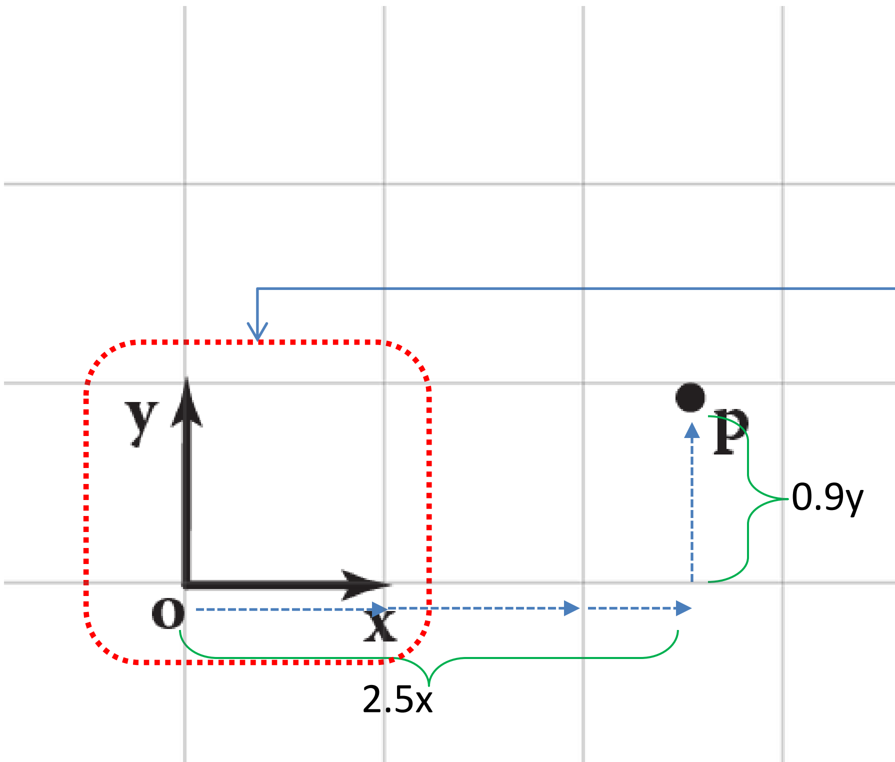
$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 2.5\begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.9\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2.5 \\ 0.9 \end{bmatrix}$$

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

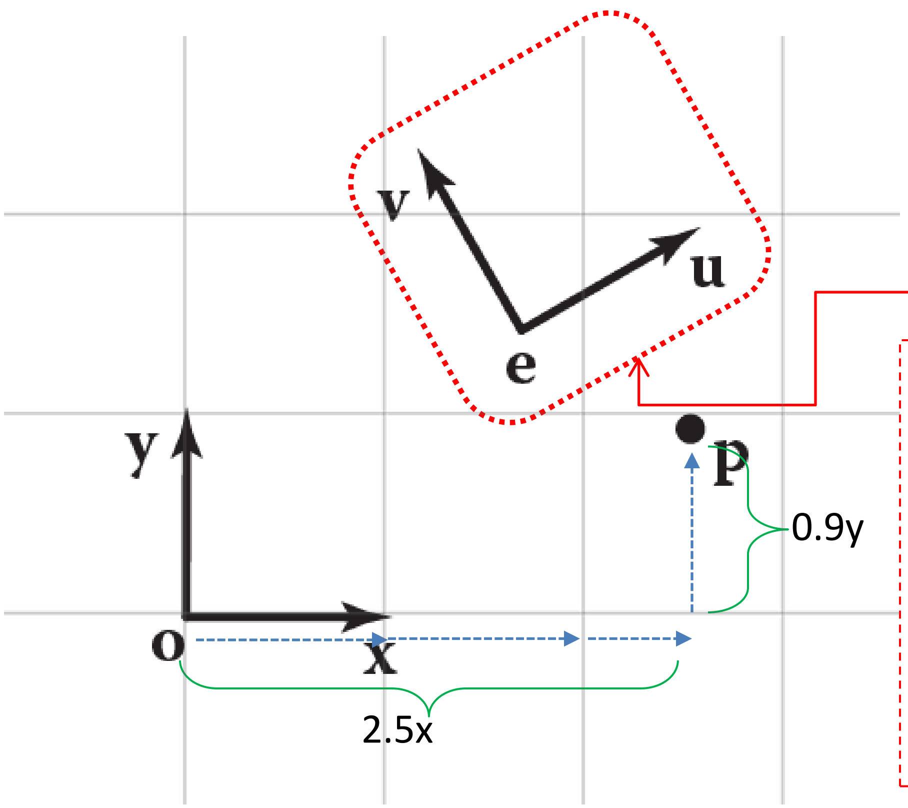
Coordinate Transformation (6/20)

Canonical coordinate system is the **frame-of-reference** for all other coordinate systems.



Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Coordinate Transformation (7/20)



Canonical coordinate system is the frame-of-reference for all other coordinate systems.

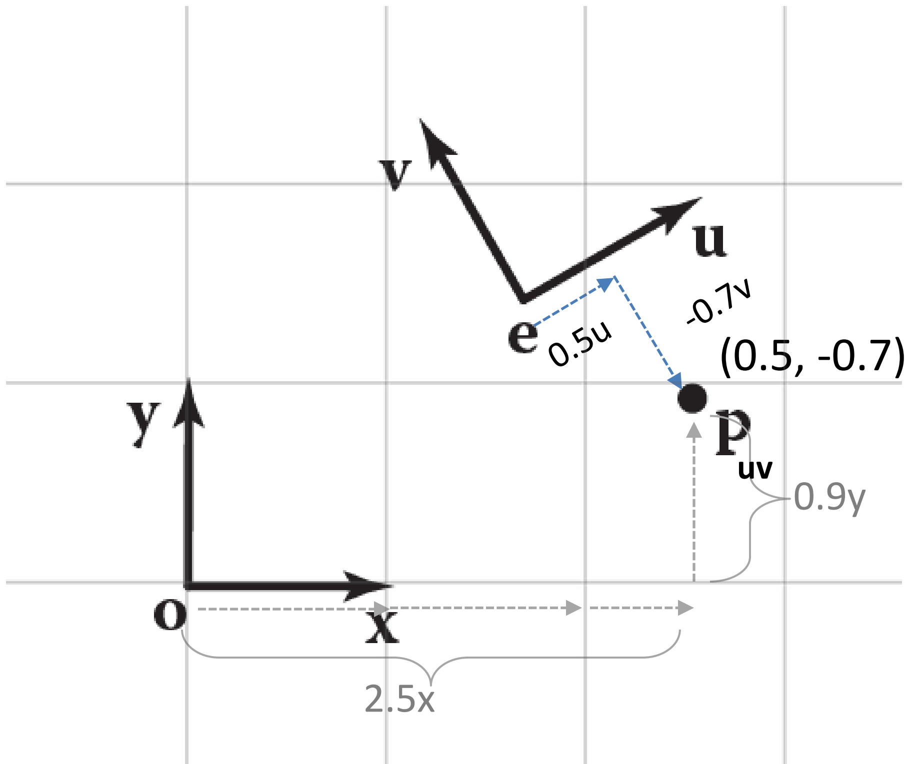
Another coordinate system (**frame coordinate**):

- **within canonical coord.**
- origin **e**
- orthonormal basis vectors **{u, v}**

$\mathbf{p}_{\mathbf{uv}} = (u_p, v_p) \equiv \mathbf{e} + u_p \mathbf{u} + v_p \mathbf{v}$

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Coordinate Transformation (8/20)



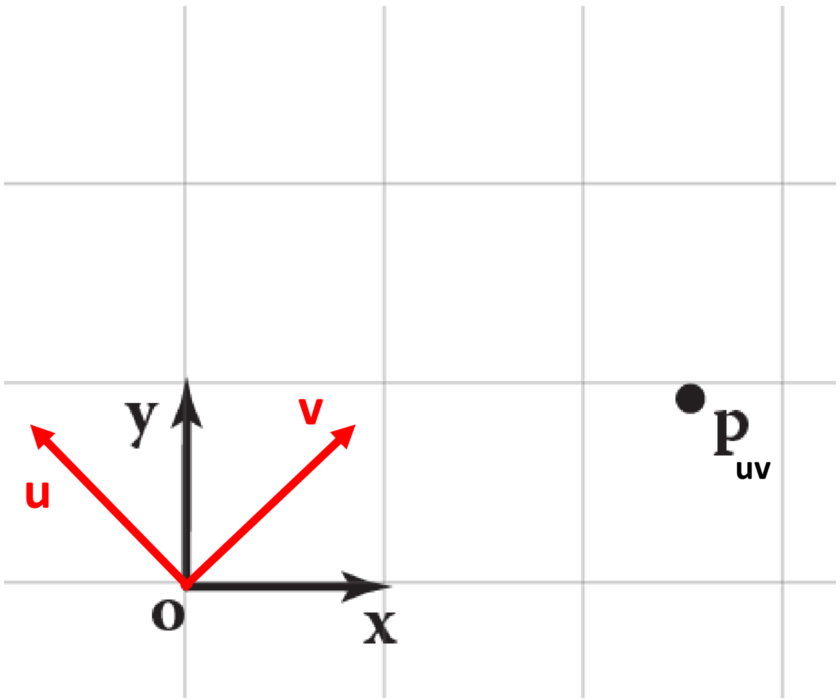
$$\mathbf{p_{uv}} = (u_p, v_p) \equiv \mathbf{e} + u_p\,\mathbf{u} + v_p\,\mathbf{v}$$

$$\begin{pmatrix} u_p \\ v_p \end{pmatrix} = \begin{pmatrix} x_e \\ y_e \end{pmatrix} + 0.5 \begin{pmatrix} x_u \\ y_u \end{pmatrix} + (-0.7) \begin{pmatrix} x_v \\ y_v \end{pmatrix}$$

$$= \begin{pmatrix} 0.5 \\ -0.7 \end{pmatrix}$$

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Coordinate Transformation (8/20)



$\mathbf{p}_{\mathbf{uv}} = (u_p, v_p) \equiv \mathbf{e} + u_p\,\mathbf{u} + v_p\,\mathbf{v}$

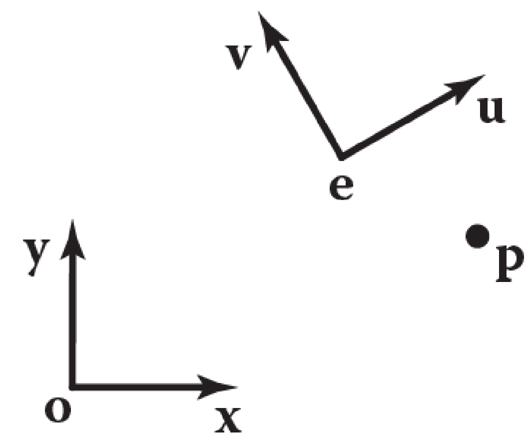
Q: Suppose the basis vectors of a frame coordinate **{e, u, v}** is achieved by rotating the basis vectors of the canonical coordinate system **{o, x, y}** by 45°. ***Determine the basis vectors of the frame coordinate system.***

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Coordinate Transformation (9/20)

$\mathbf{p_{xy}} = (\mathrm{x_p}, \mathrm{y_p}) \equiv \mathbf{o} + x_p\,\mathbf{x} + y_p\,\mathbf{y}$

$\mathbf{p_{uv}} = (\mathrm{u_p}, \mathrm{v_p}) \equiv \mathbf{e} + u_p\,\mathbf{u} + v_p\,\mathbf{v}$

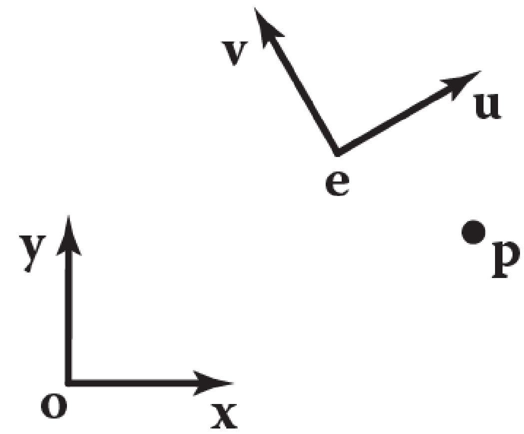


Coordinate Transformation (10/20)

$\mathbf{p_{xy}} = (x_p, y_p) \equiv \mathbf{o} + x_p\,\mathbf{x} + y_p\,\mathbf{y}$

$\mathbf{p_{uv}} = (u_p, v_p) \equiv \mathbf{e} + u_p\,\mathbf{u} + v_p\,\mathbf{v}$

$\mathbf{p_{xy}} \leftrightarrow \mathbf{p_{uv}}$

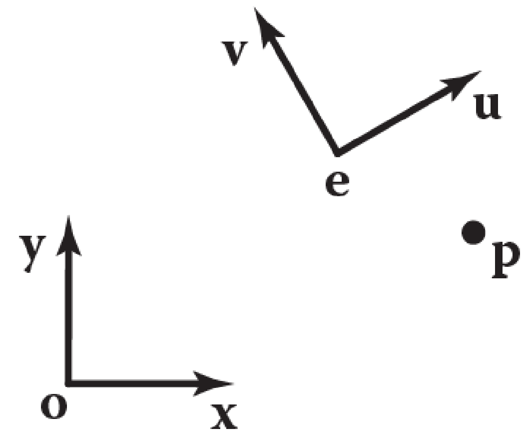


Coordinate Transformation (11/20)

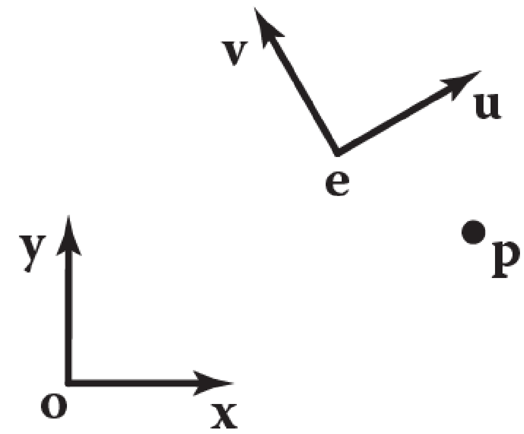
$\mathbf{p}_{\mathbf{xy}} = (x_p, y_p) \equiv \mathbf{o} + x_p\,\mathbf{x} + y_p\,\mathbf{y}$

$\mathbf{p}_{\mathbf{uv}} = (u_p, v_p) \equiv \mathbf{e} + u_p\,\mathbf{u} + v_p\,\mathbf{v}$

$\mathbf{p}_{\mathbf{xy}} = \mathbf{e} + u_p\,\mathbf{u} + v_p\,\mathbf{v}$



Coordinate Transformation (12/20)

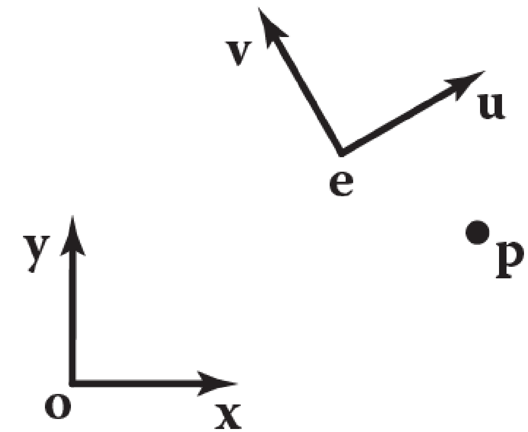


$\mathbf{p}_{xy} = \mathbf{e} + u_p\,\mathbf{u} + v_p\,\mathbf{v}$

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} x_e \\ y_e \end{bmatrix} + u_p \begin{bmatrix} x_u \\ y_u \end{bmatrix} + v_p \begin{bmatrix} x_v \\ y_v \end{bmatrix}$$

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Coordinate Transformation (13/20)

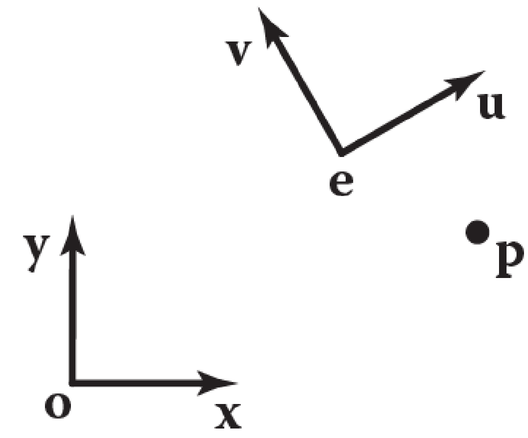


$\mathbf{p}_{\mathbf{xy}} = \mathbf{e} + u_p \mathbf{u} + v_p \mathbf{v}$

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} x_e \\ y_e \end{bmatrix} + \begin{bmatrix} x_u & x_v \\ y_u & y_v \end{bmatrix} \begin{bmatrix} u_p \\ v_p \end{bmatrix}$$

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Coordinate Transformation (14/20)



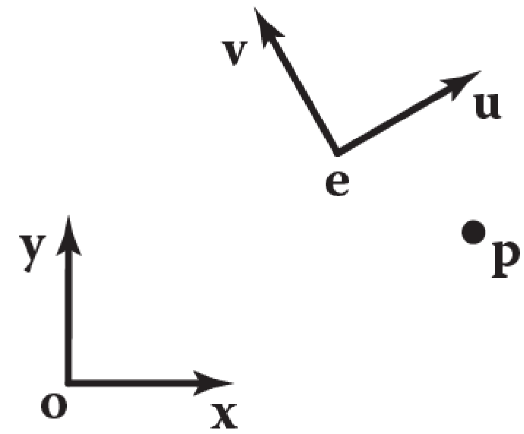
$$\mathbf{p}_{xy} = \mathbf{e} + u_p\,\mathbf{u} + v_p\,\mathbf{v}$$

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} x_e \\ y_e \end{bmatrix} + \begin{bmatrix} x_u & x_v \\ y_u & y_v \end{bmatrix} \begin{bmatrix} u_p \\ v_p \end{bmatrix}$$

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_e \\ 0 & 1 & y_e \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_u & x_v & 0 \\ y_u & y_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_p \\ v_p \\ 1 \end{bmatrix}$$

Coordinate Transformation (15/20)

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} x_u & x_v & x_e \\ y_u & y_v & y_e \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_p \\ v_p \\ 1 \end{bmatrix}$$

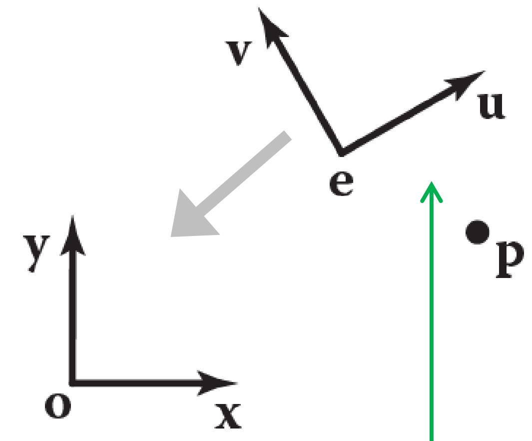


$$\mathbf{p}_{xy} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{e} \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}_{uv}$$

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_e \\ 0 & 1 & y_e \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_u & x_v & 0 \\ y_u & y_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_p \\ v_p \\ 1 \end{bmatrix}$$

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Coordinate Transformation (18/20)



$$\mathbf{p}_{xy} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{e} \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}_{uv}$$

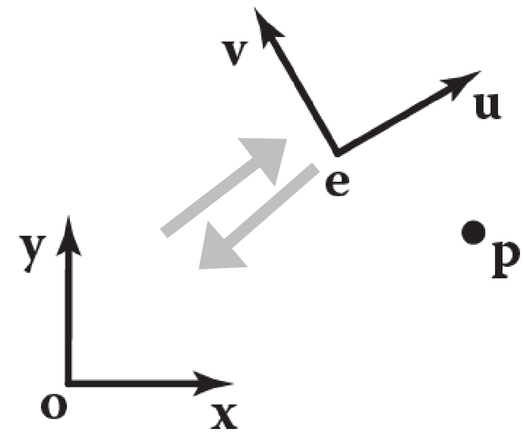
frame-to-canonical matrix

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Coordinate Transformation (19/20)

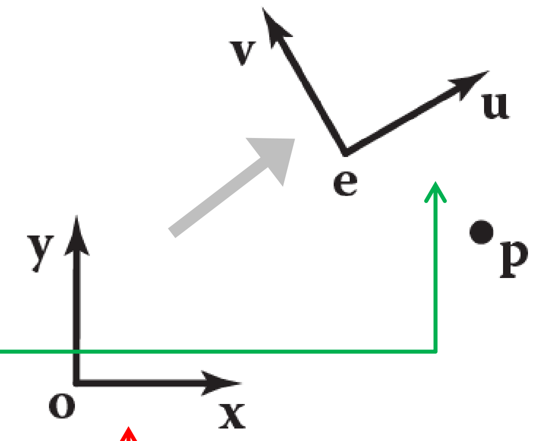
$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_e \\ 0 & 1 & y_e \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_u & x_v & 0 \\ y_u & y_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_p \\ v_p \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u_p \\ v_p \\ 1 \end{bmatrix} = \begin{bmatrix} x_u & y_u & 0 \\ x_v & y_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_e \\ 0 & 1 & -y_e \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix}$$



Coordinate Transformation (20/20)

$$\begin{bmatrix} u_p \\ v_p \\ 1 \end{bmatrix} = \begin{bmatrix} x_u & y_u & 0 \\ x_v & y_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_e \\ 0 & 1 & -y_e \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix}$$



$$\mathbf{p}_{uv} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{e} \\ 0 & 0 & 1 \end{bmatrix}^{-1} \mathbf{p}_{xy}$$

canonical-to-frame matrix

3D Coordinate Transformation (2/2)

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_e \\ 0 & 1 & 0 & y_e \\ 0 & 0 & 1 & z_e \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_u & x_v & x_w & 0 \\ y_u & y_v & y_w & 0 \\ z_u & z_v & z_w & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_p \\ v_p \\ w_p \\ 1 \end{bmatrix}$$

$$\mathbf{p}_{xyz} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{e} \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{p}_{uvw},$$

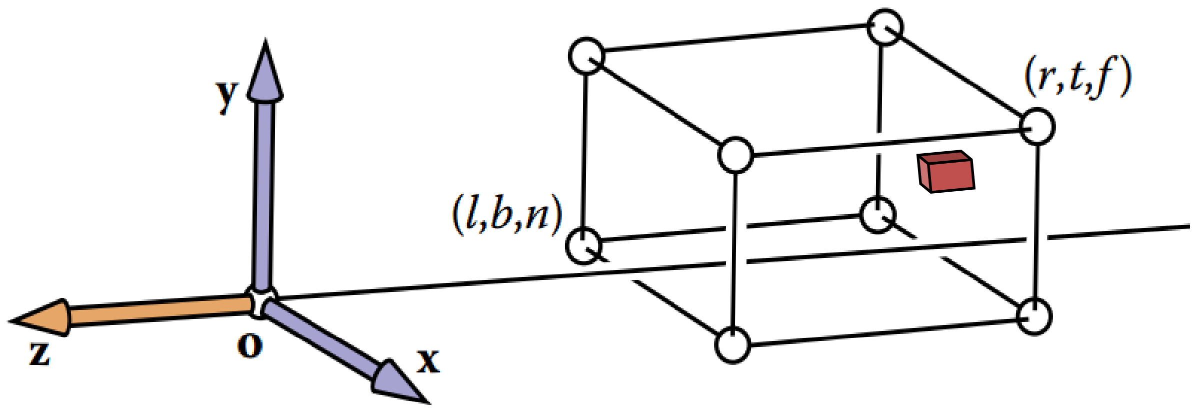
$$\begin{bmatrix} u_p \\ v_p \\ w_p \\ 1 \end{bmatrix} = \begin{bmatrix} x_u & y_u & z_u & 0 \\ x_v & y_v & z_v & 0 \\ x_w & y_w & z_w & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_e \\ 0 & 1 & 0 & -y_e \\ 0 & 0 & 1 & -z_e \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}$$

$$\mathbf{p}_{uvw} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{e} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \mathbf{p}_{xyz}.$$

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

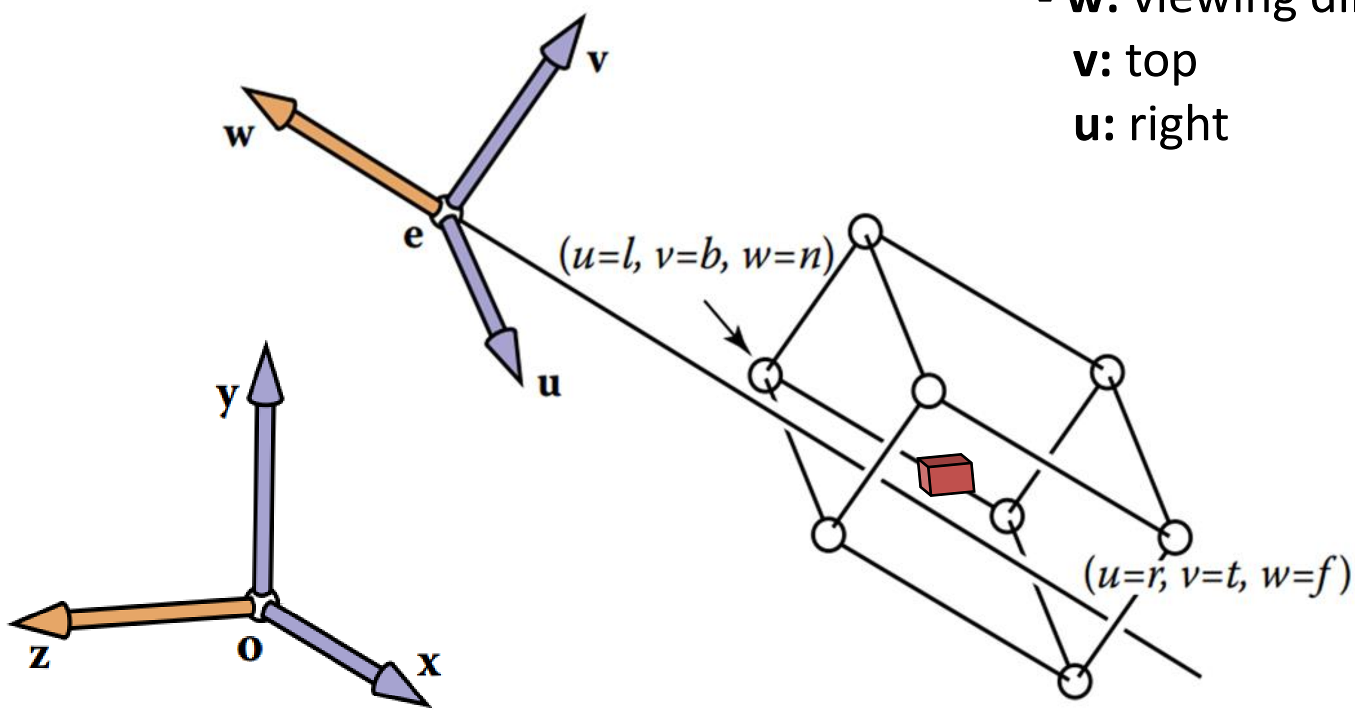
Camera Transformation (2/6)

- We'd like to be able to change the viewpoint in 3D and look in any direction.



Camera Transformation (3/6)

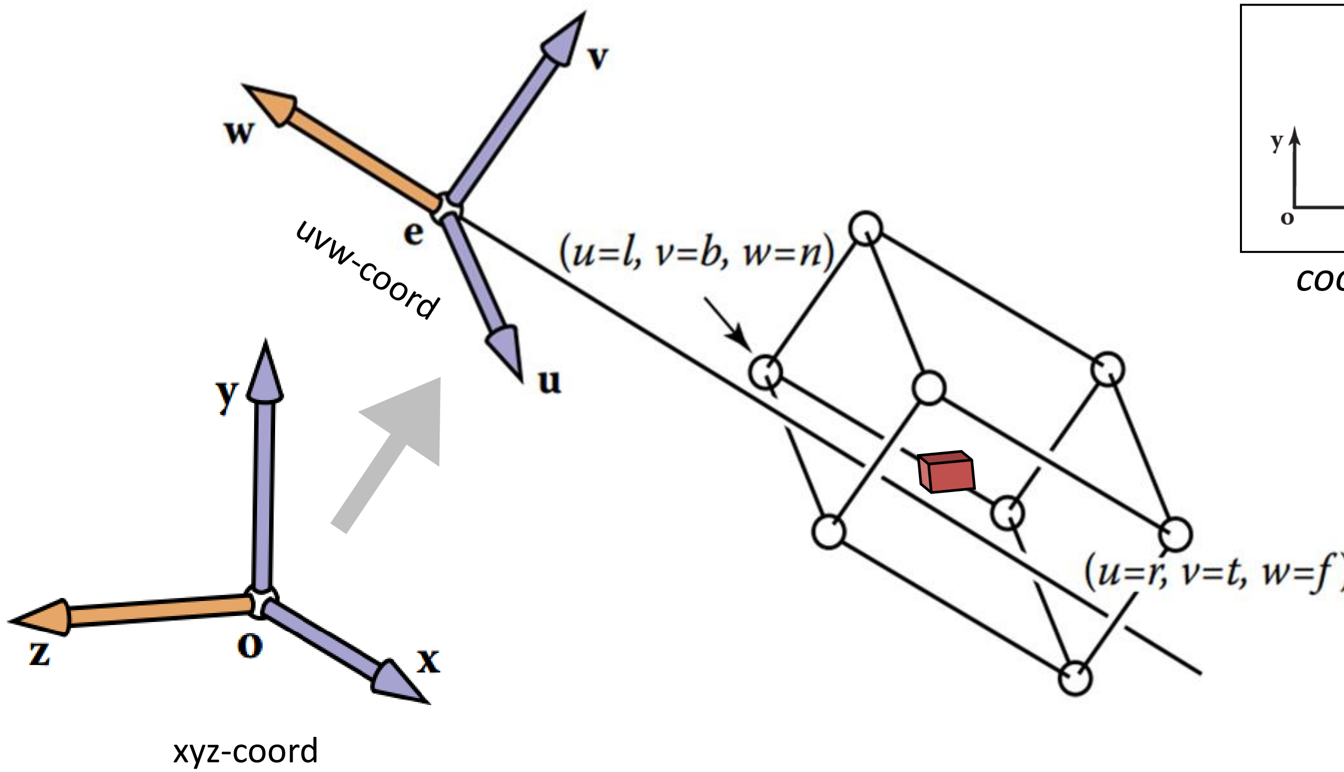
e: camera/eye position
- **w:** viewing direction
v: top
u: right

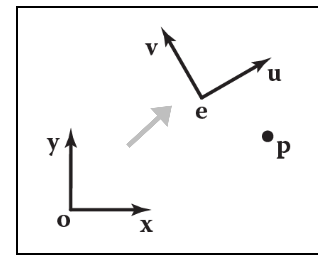


Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Camera Transformation (4/6)

from **xyz-coordinates** into **uvw-coordinates**





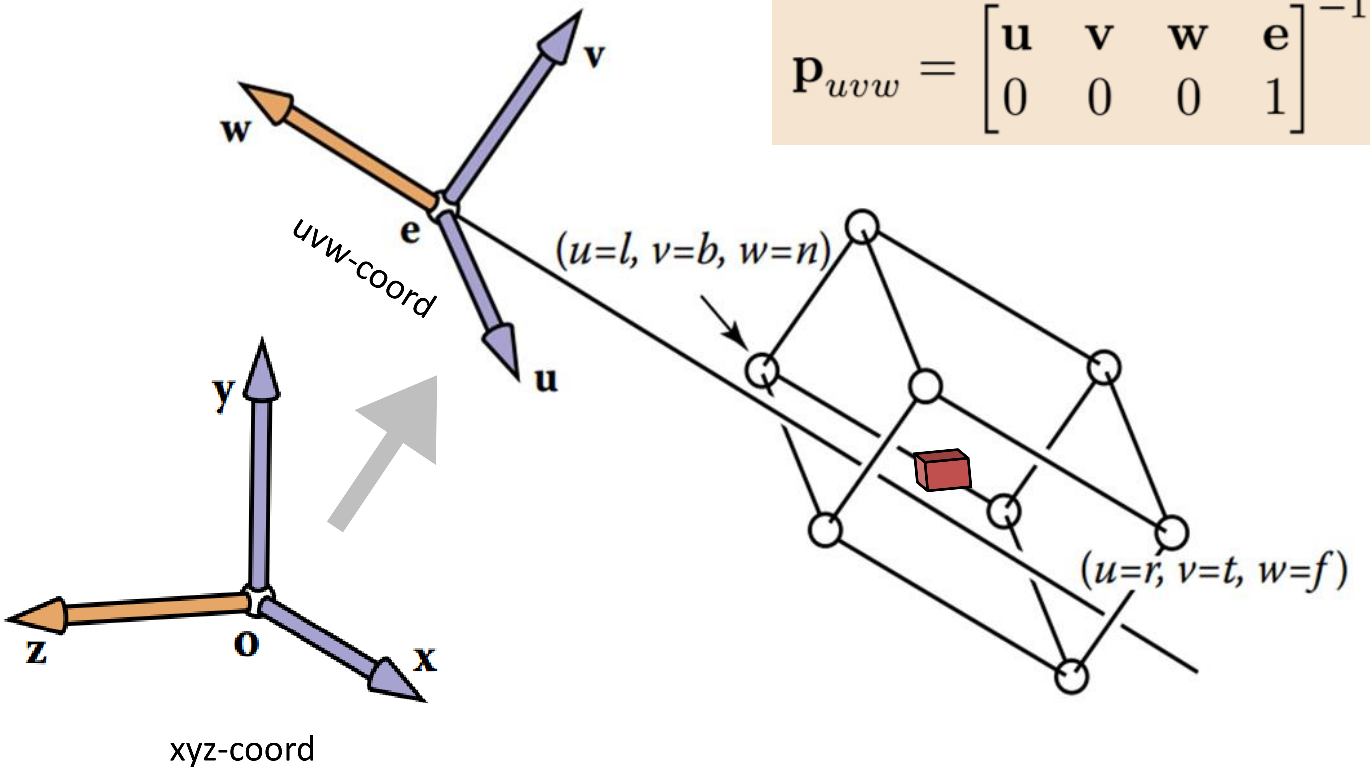
coord. trans.

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Camera Transformation (5/6)

from **xyz-coordinates** into **uvw-coordinates**

$$\mathbf{p}_{uvw} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{e} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \mathbf{p}_{xyz}.$$



Camera Transformation (6/6)

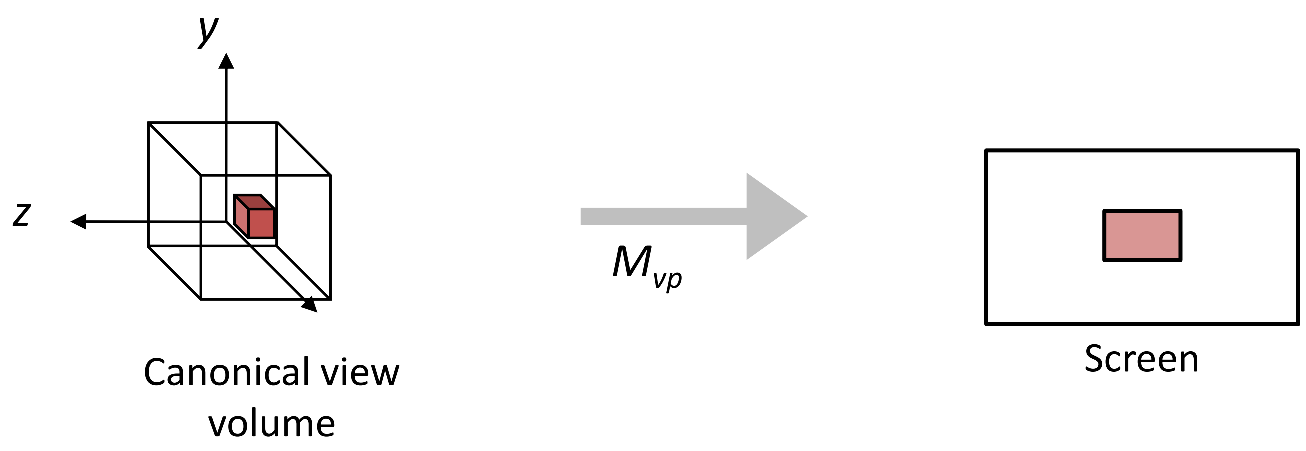
$$\mathbf{p}_{uvw} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{e} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \mathbf{p}_{xyz}.$$

<u>canonical-to-frame matrix:</u>

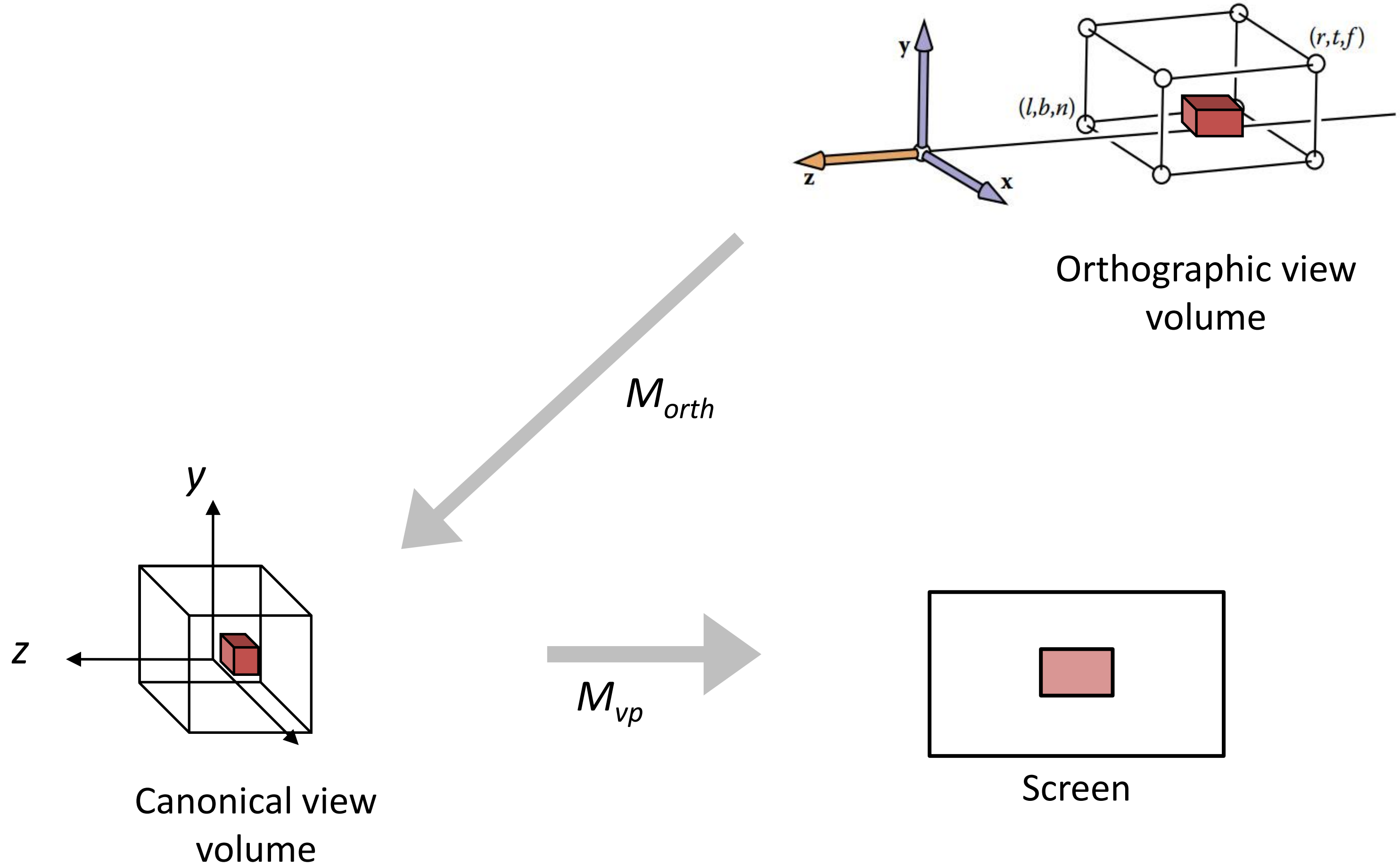
$$\mathbf{M}_{\text{cam}} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{e} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} x_u & y_u & z_u & 0 \\ x_v & y_v & z_v & 0 \\ x_w & y_w & z_w & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_e \\ 0 & 1 & 0 & -y_e \\ 0 & 0 & 1 & -z_e \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

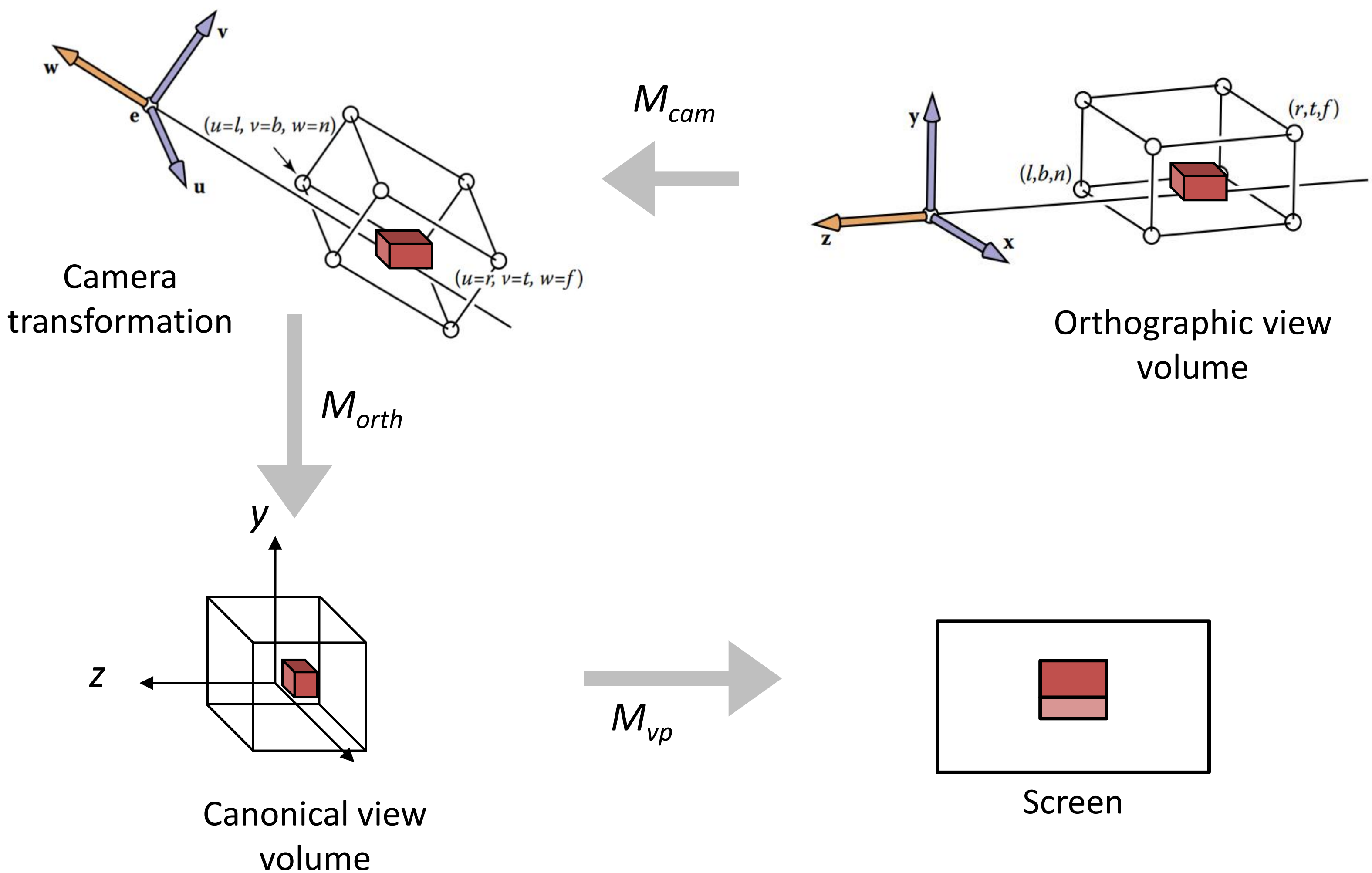
Summary (1/5)



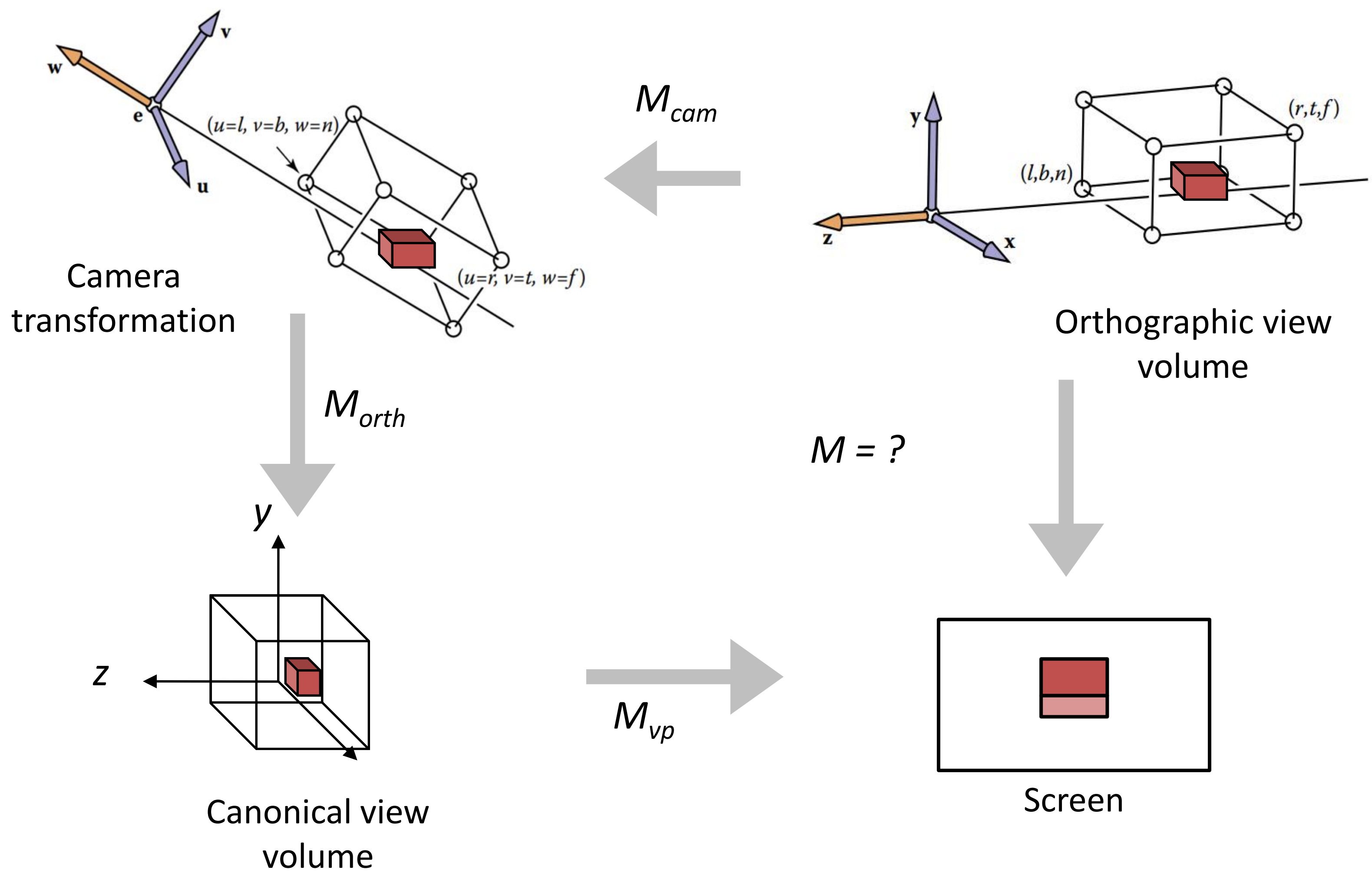
Summary (2/5)


y
z
x
(l,b,n)
(r,t,f)
Orthographic view volume
M_{orth}
y
z
Canonical view volume
M_{vp}
Screen

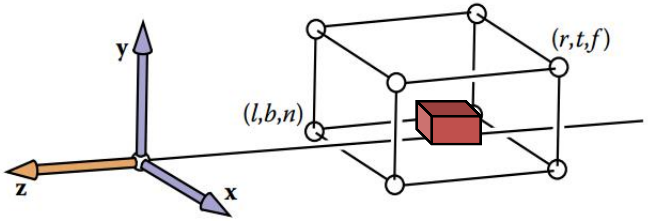
Summary (3/5)


w
v
e
u
(u=l, v=b, w=n)
(u=r, v=t, w=f)
Camera transformation
M_{cam}
y
z
x
(l,b,n)
(r,t,f)
Orthographic view volume
M_{orth}
y
z
Canonical view volume
M_{vp}
Screen

Summary (4/5)

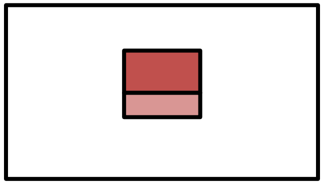

w
v
e
u
(u=l, v=b, w=n)
(u=r, v=t, w=f)
Camera transformation
M_{cam}
y
z
x
(r,t,f)
(l,b,n)
Orthographic view volume
M_{orth}
$M = ?$
y
z
Canonical view volume
M_{vp}
Screen

Summary (5/5)



Orthographic view volume

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



Screen

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

Drawing many 3D lines with endpoints a_i and b_i:

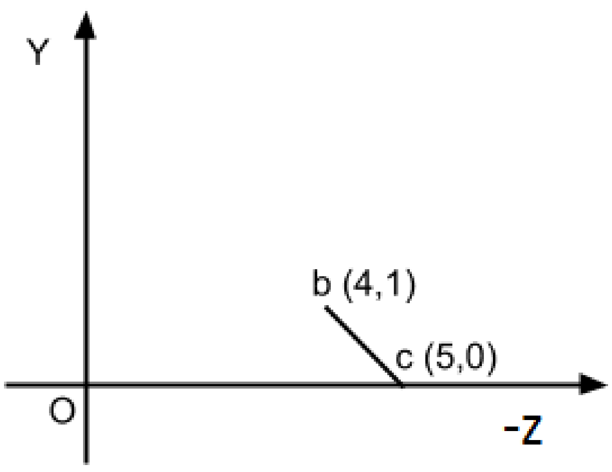
```
Construct M_vp
Construct M_orth
Construct M_cam

M = M_vp*M_orth*M_cam

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

Practice Problem - 1

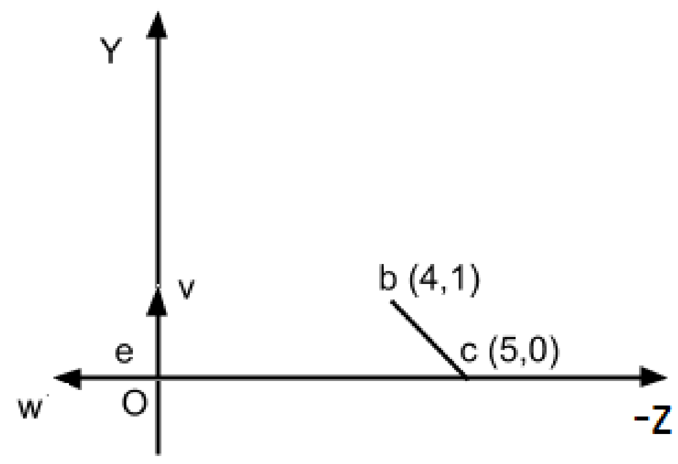
- Origin ***O*** and basis ***{y, z}*** construct a 2D canonical coordinate system. Within this, line ***bc*** is our model ($\boldsymbol{P_{xy}}$). We want to view it from a new 2D camera (frame) with origin ***(4, 8)*** looking downward.

(a) Determine *canonical-to-frame* matrix

(b) Calculate and plot $\boldsymbol{P_{wv}}$.

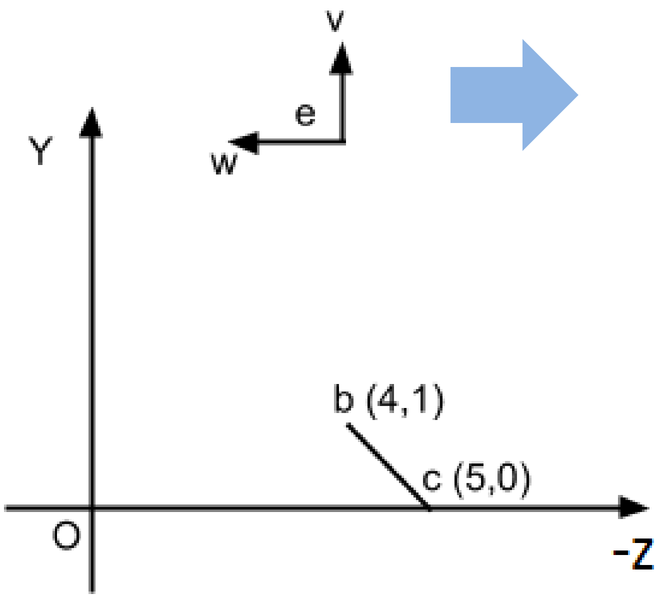


Practice Problem - 1



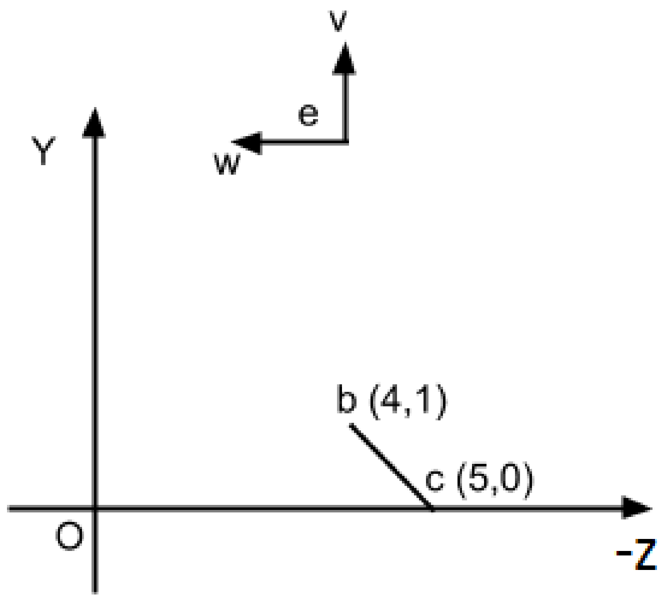
Practice Problem - 1

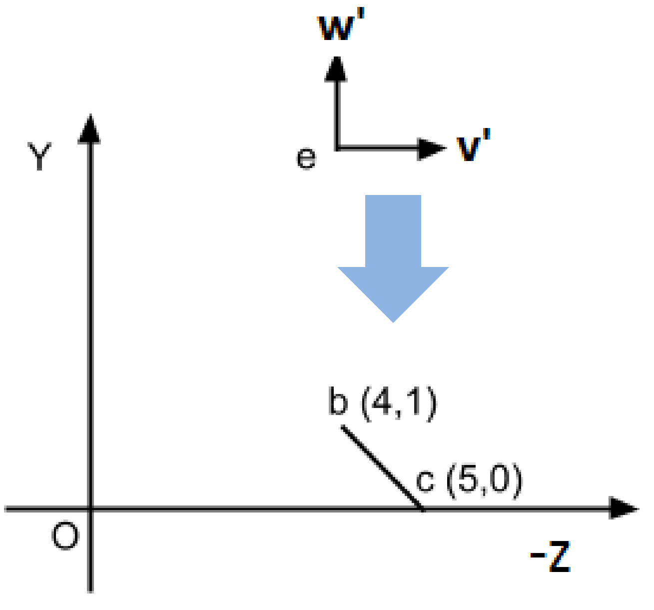
- e ≡ (4,8); w = ? ; v = ?



Practice Problem - 1

- e ≡ (4,8); w' = ? ; v' = ?



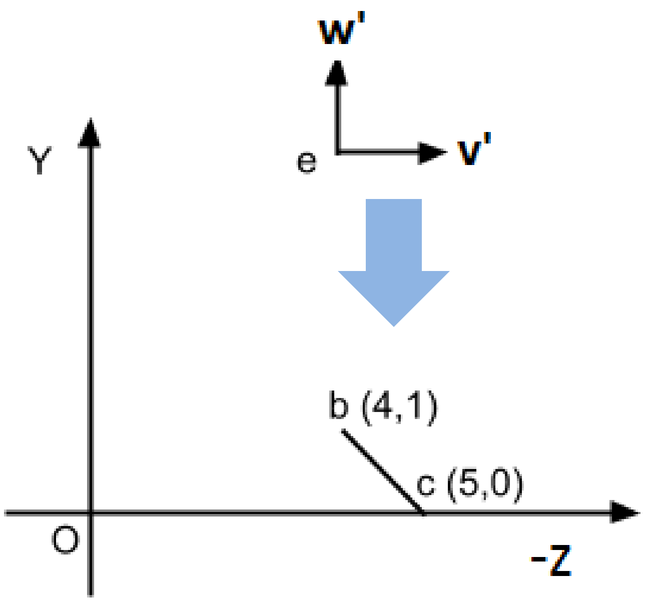


Practice Problem - 1

- e ≡ (4,8); w’ = ? ; v’ = ?

$$\mathbf{p}_{uv} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{e} \\ 0 & 0 & 1 \end{bmatrix}^{-1} \mathbf{p}_{xy}$$

$$\begin{bmatrix} u_p \\ v_p \\ 1 \end{bmatrix} = \begin{bmatrix} x_u & y_u & 0 \\ x_v & y_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_e \\ 0 & 1 & -y_e \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix}$$



Practice Problem - 2

a) Explain with appropriate example that the frame-to-canonical transformation can be expressed as a rotation followed by a translation.
   – Hint: section 6.5

b) Explain with appropriate example that canonical-to-frame transformation is a translation followed by a rotation; they are the inverses of the rotation and translation we used to build the frame-to-canonical matrix.
   – Hint: section 6.5