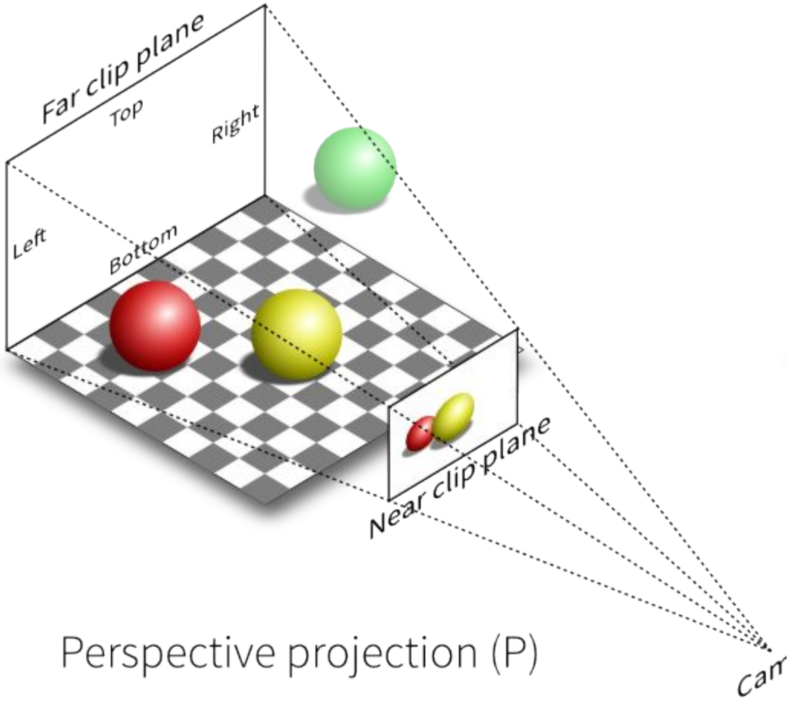
# Projection++

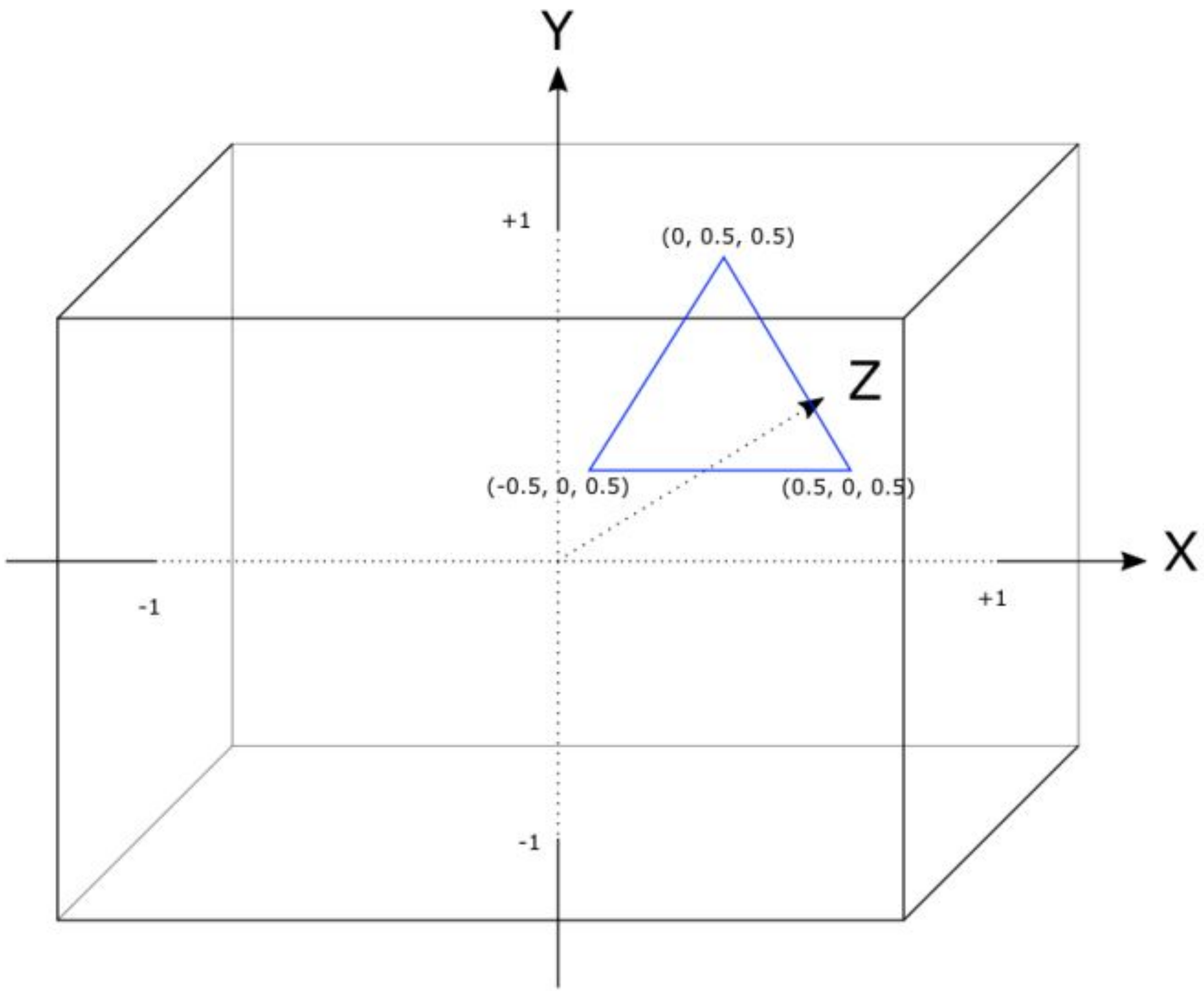
CSE 409 Computer Graphics

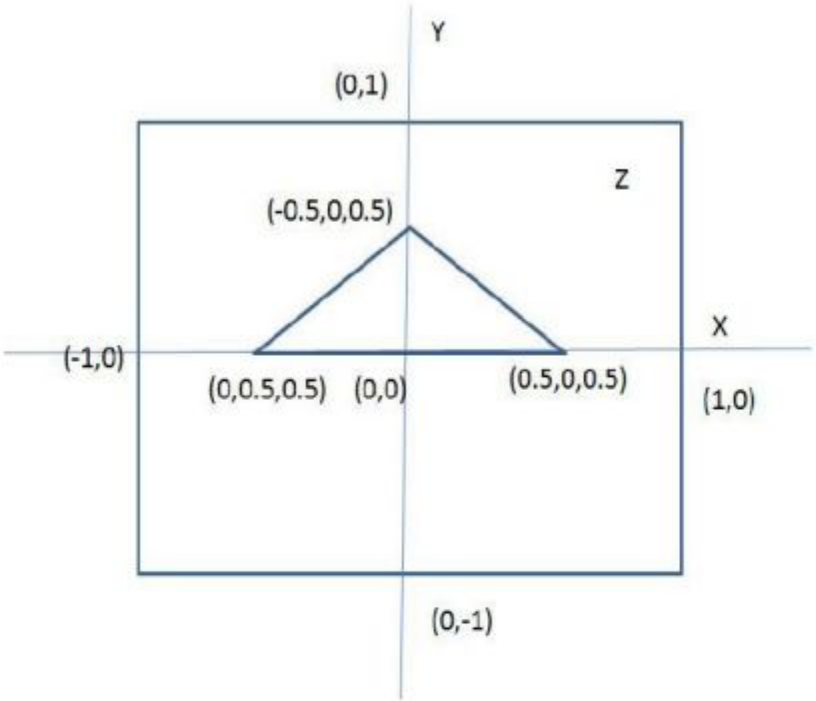
Kowshic Roy

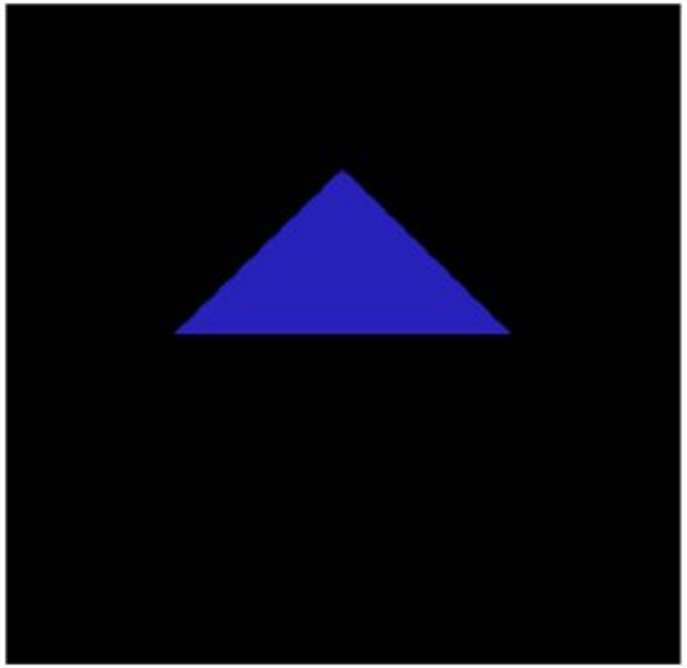
Department of CSE, BUET

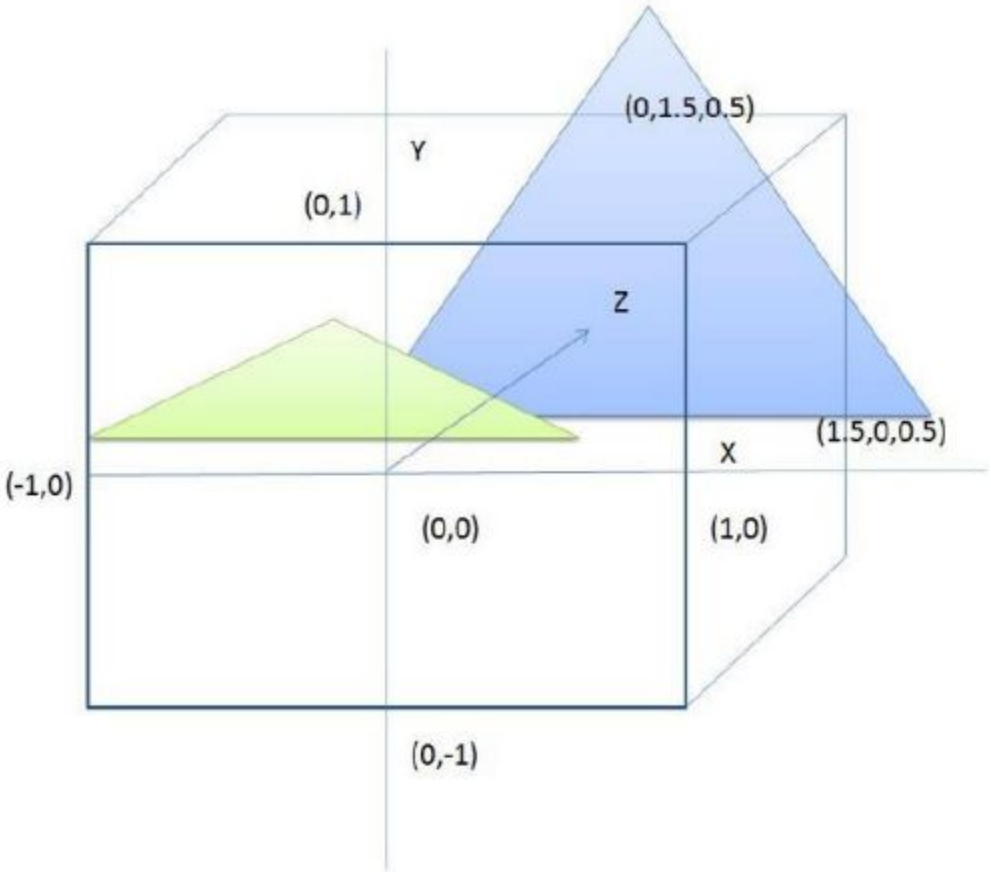


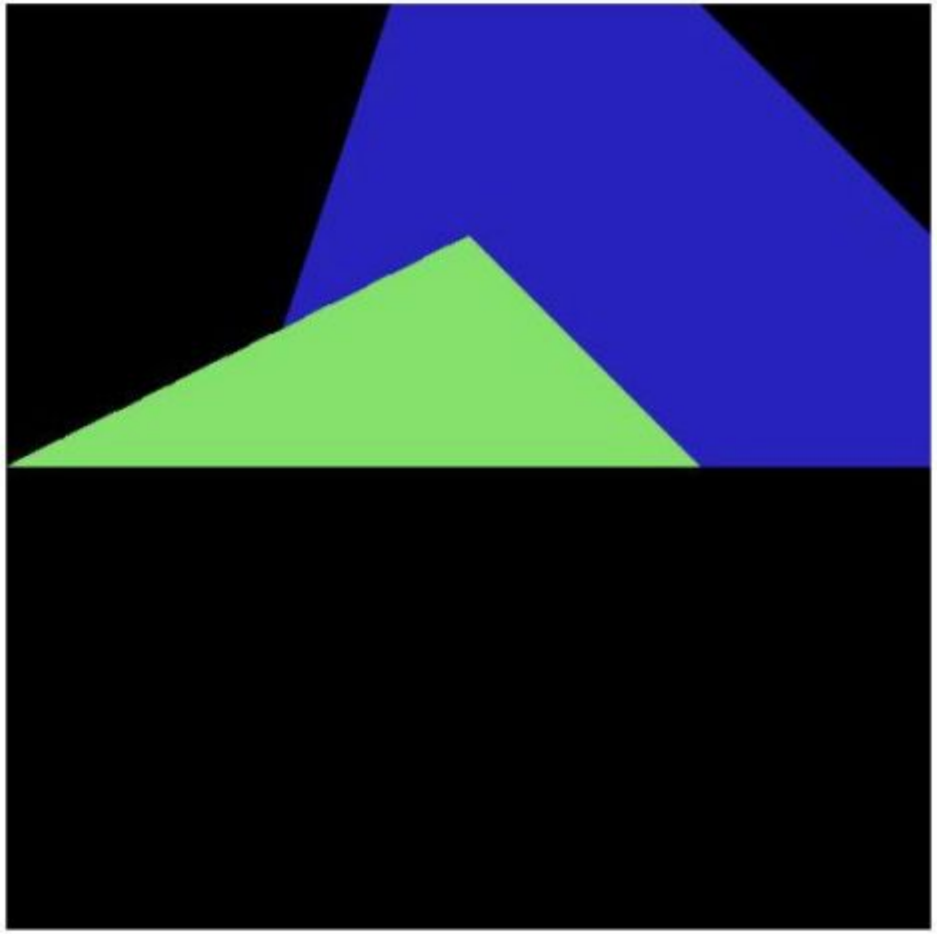

Perspective projection (P)


Y
+1
(0, 0.5, 0.5)
Z
(-0.5, 0, 0.5)
(0.5, 0, 0.5)
X
-1
+1
-1


Y
(0,1)
Z
(-0.5,0,0.5)
X
(-1,0)
(0,0.5,0.5)
(0,0)
(0.5,0,0.5)
(1,0)
(0,-1)

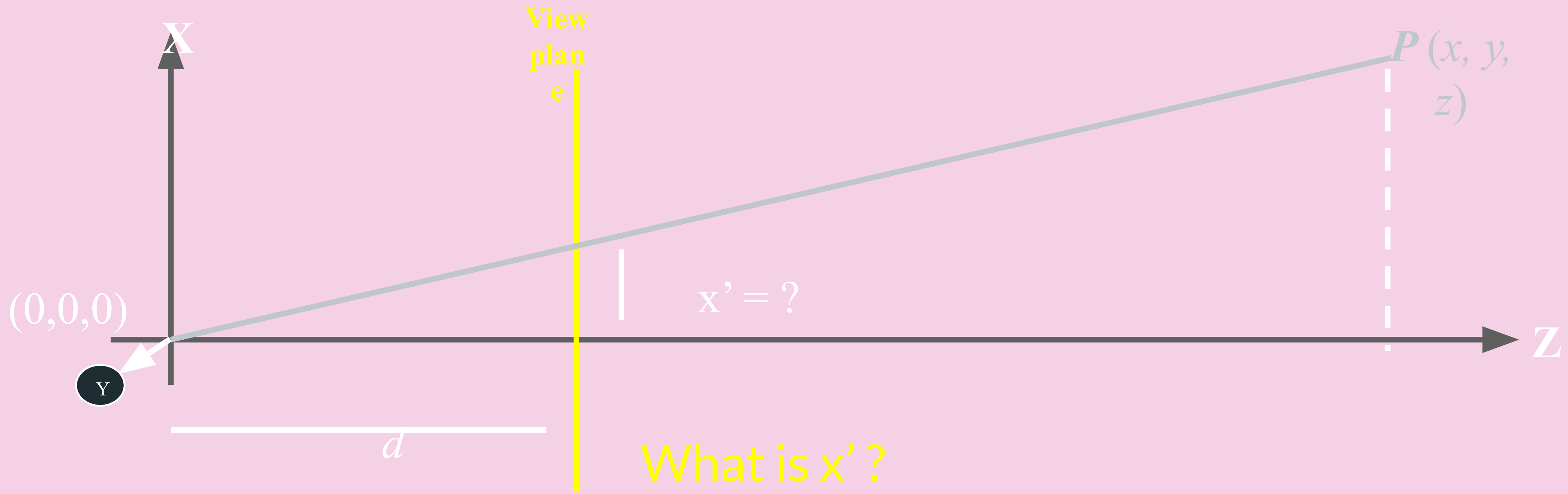



(0,1.5,0.5)
Y
(0,1)
Z
(1.5,0,0.5)
X
(-1,0)
(0,0)
(1,0)
(0,-1)



# Perspective Projection

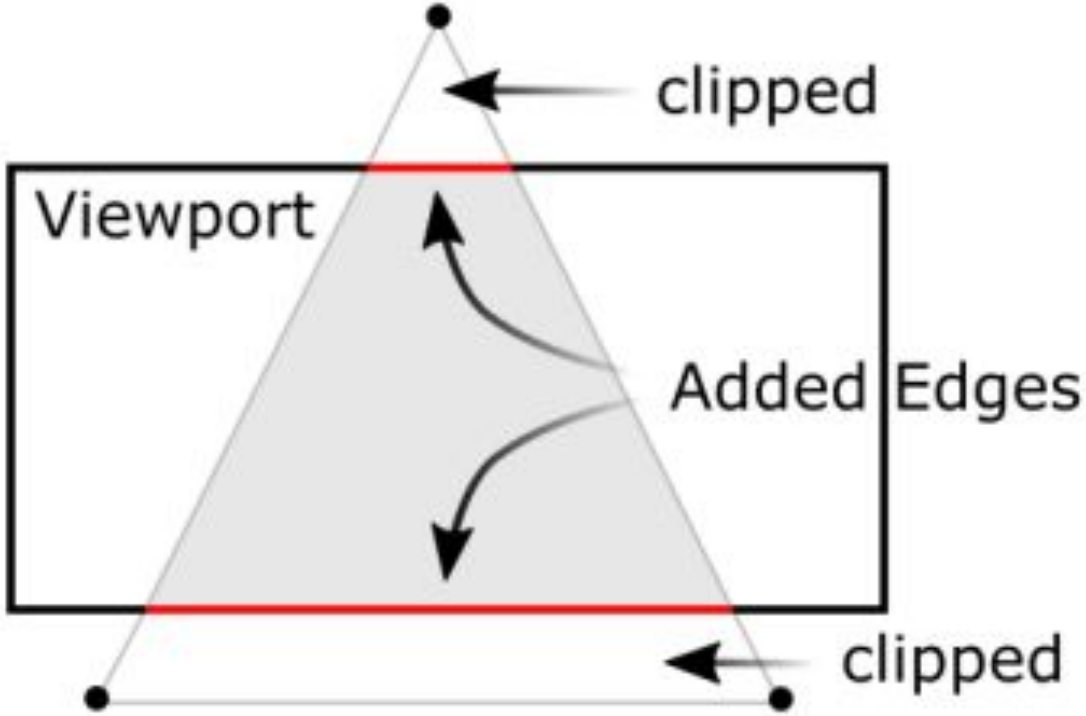
The geometry of the situation is that of similar triangles. View from Y-axis:



# Perspective Projection Matrix

$$M_{perspective} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

# Why don’t we live happily ever after?

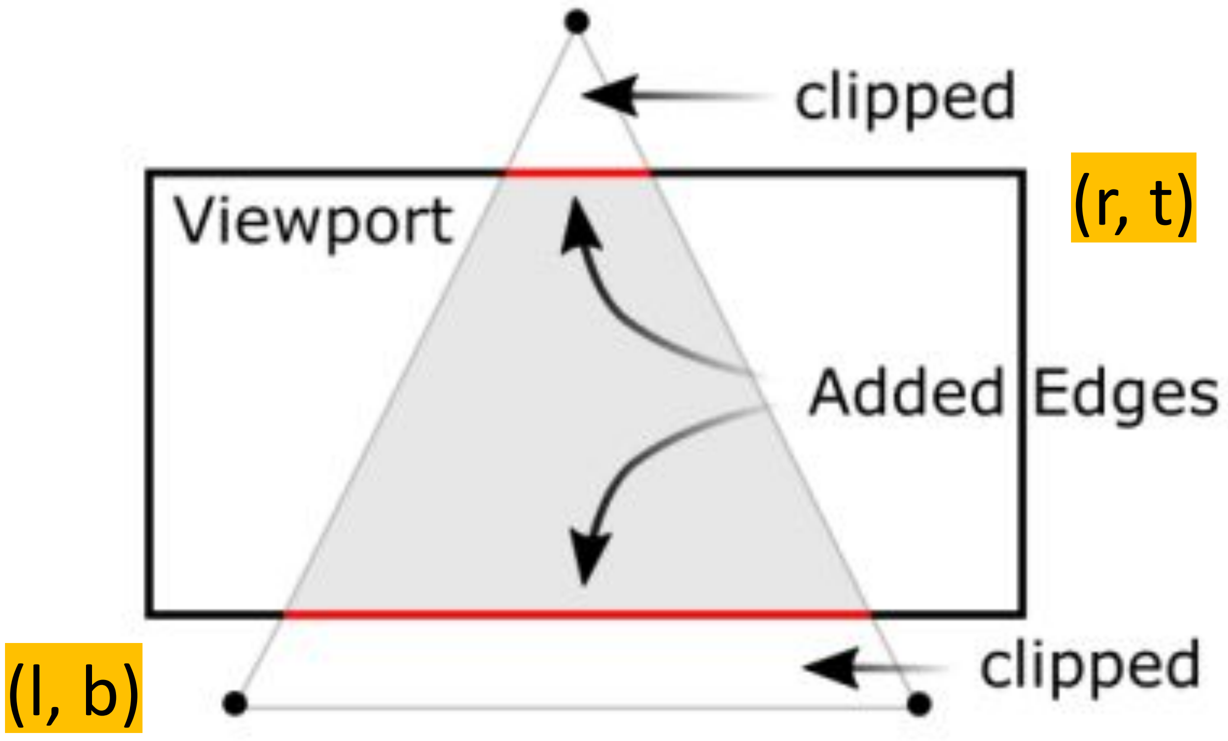


# gluPerspective

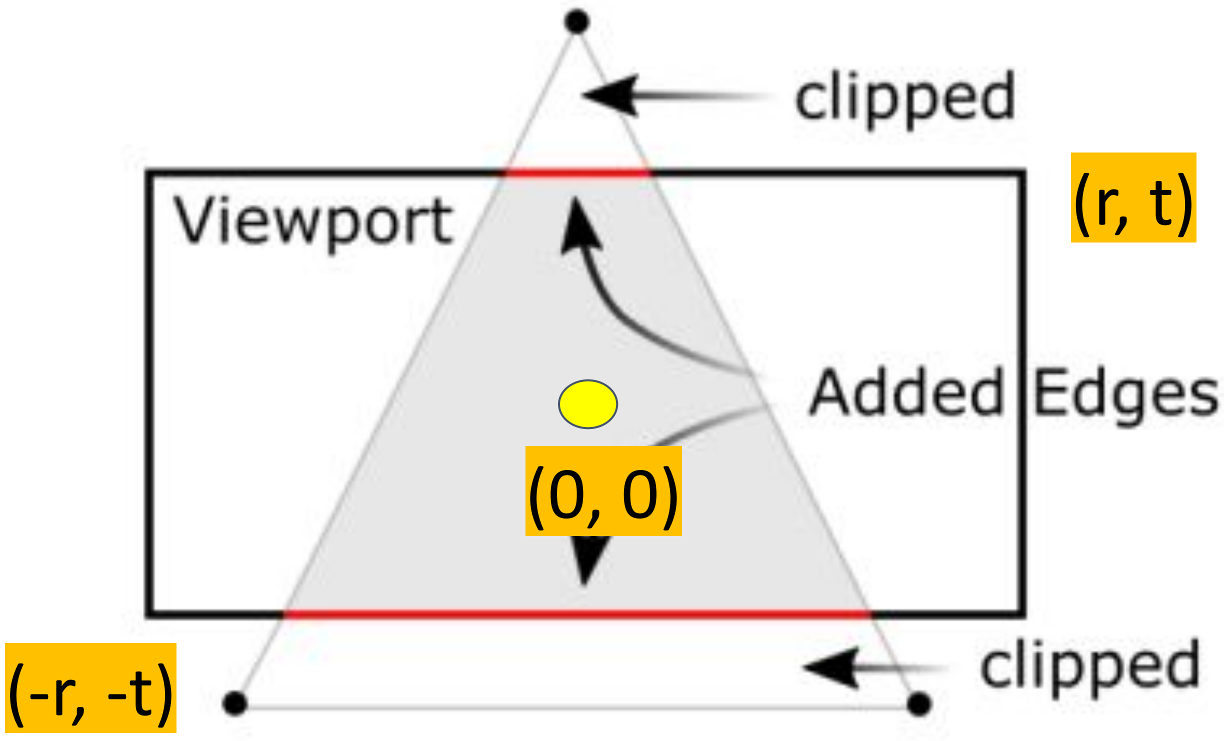
```
void gluPerspective(
    GLdouble fovy,
    GLdouble aspect,
    GLdouble zNear [near],
    GLdouble zFar[far]
);
```

$fov_x = aspect * fov_y$

# Why don’t we live happily ever after?



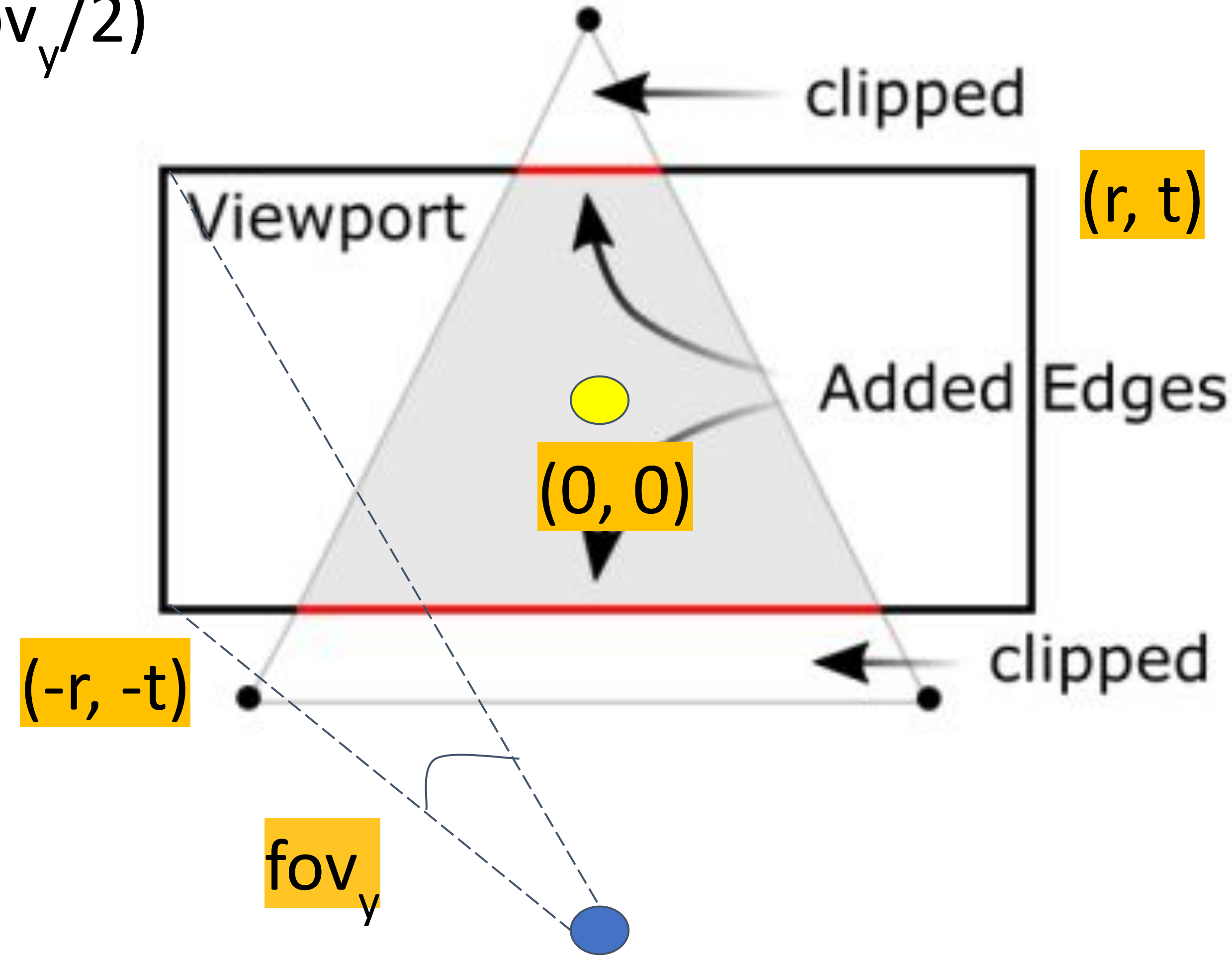
# Why don't we live happily ever after?



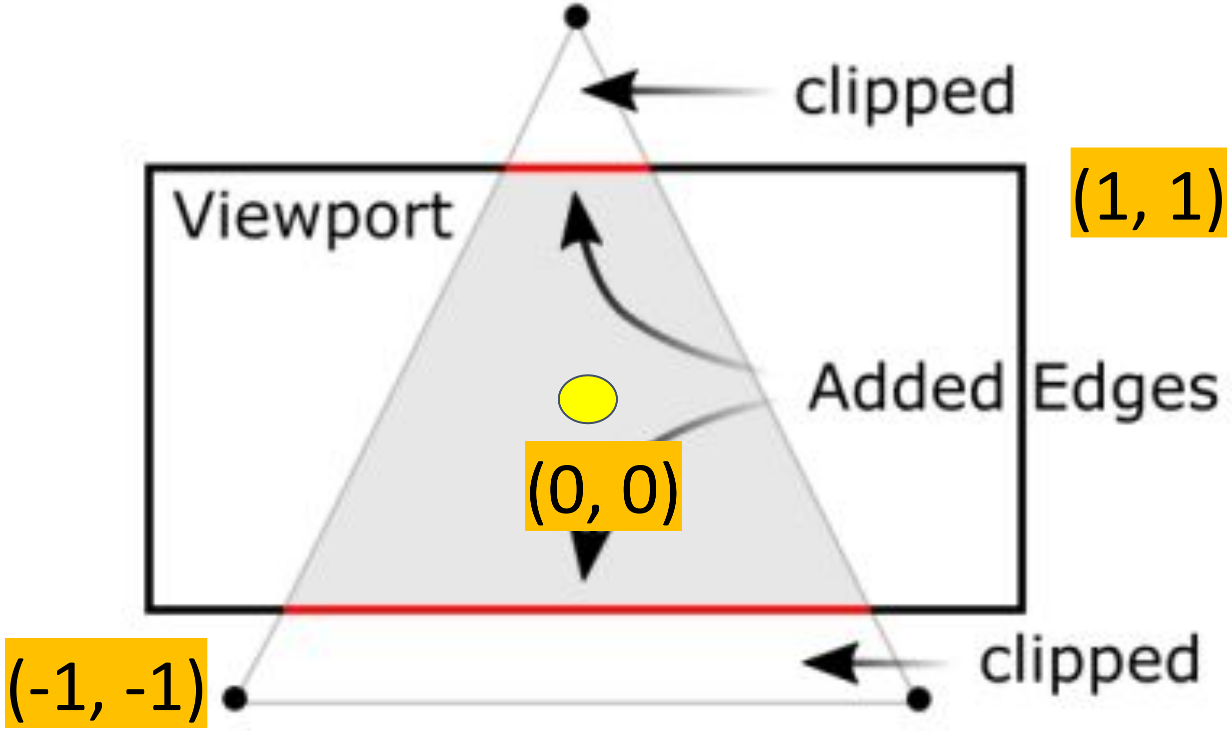
# Why don't we live happily ever after?

$t = near * \tan(fov_y/2)$

r= ..



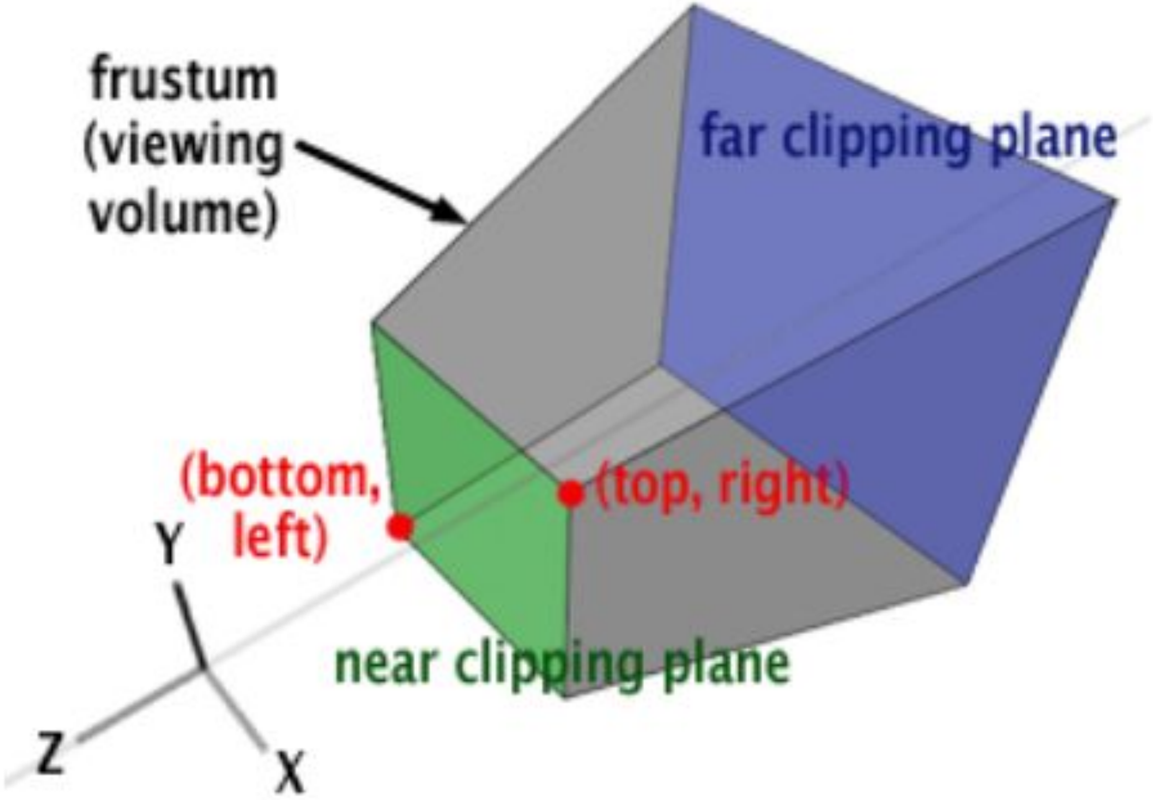
# Why don't push more?



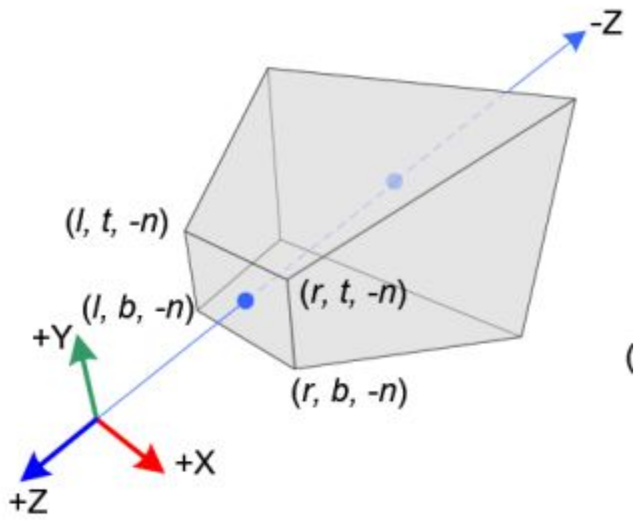
# Why don't push more?

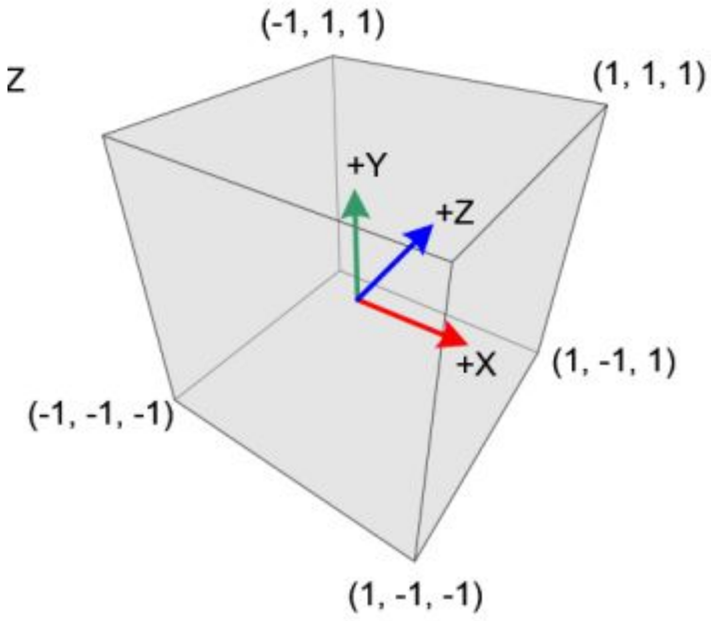
$$M_{perspective} = \begin{bmatrix} d/r & 0 & 0 & 0 \\ 0 & d/t & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

# Frustum

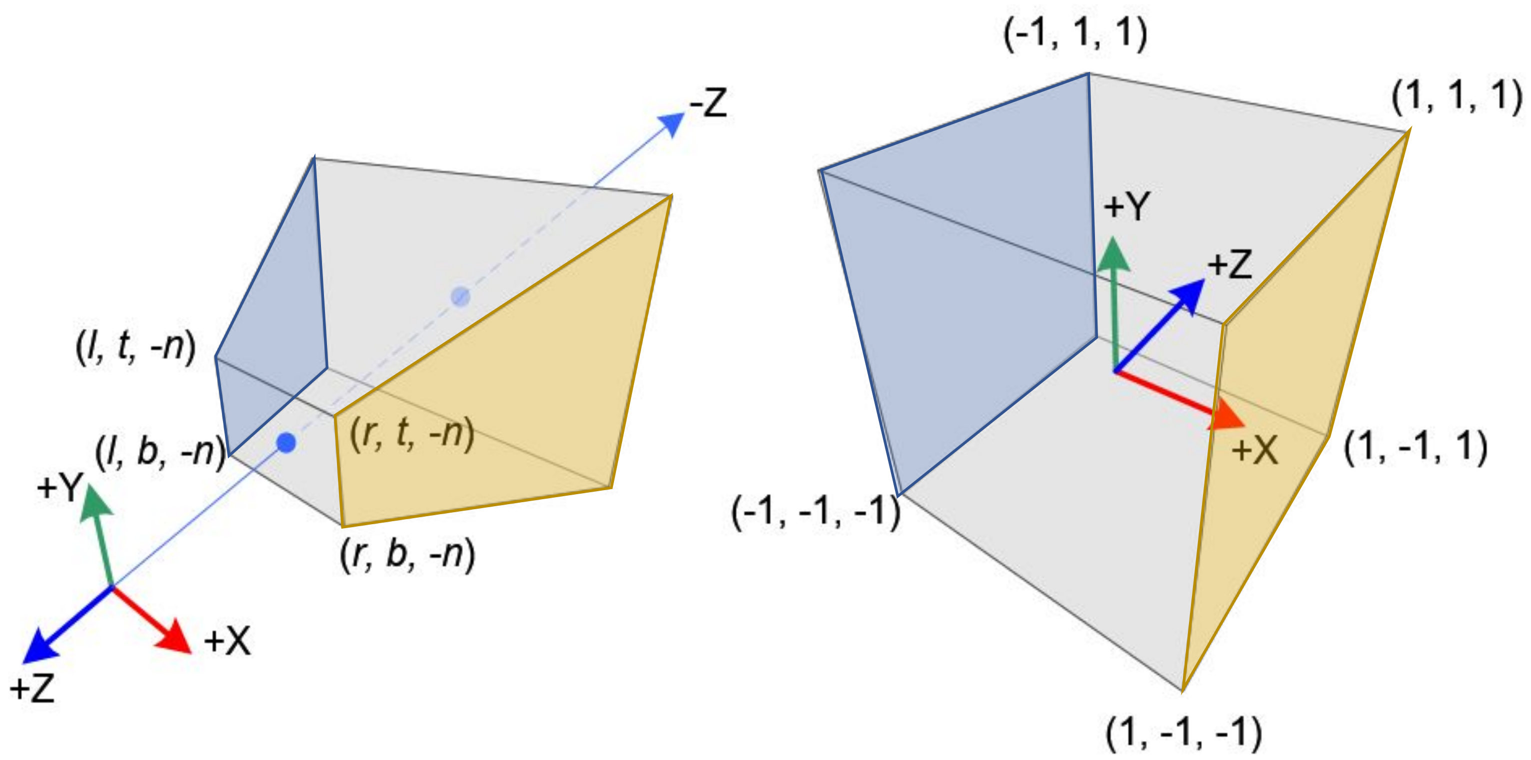

frustum
(viewing
volume)
far clipping plane
(bottom,
left)
(top, right)
near clipping plane
Y
Z
X

# Frustum to NDC

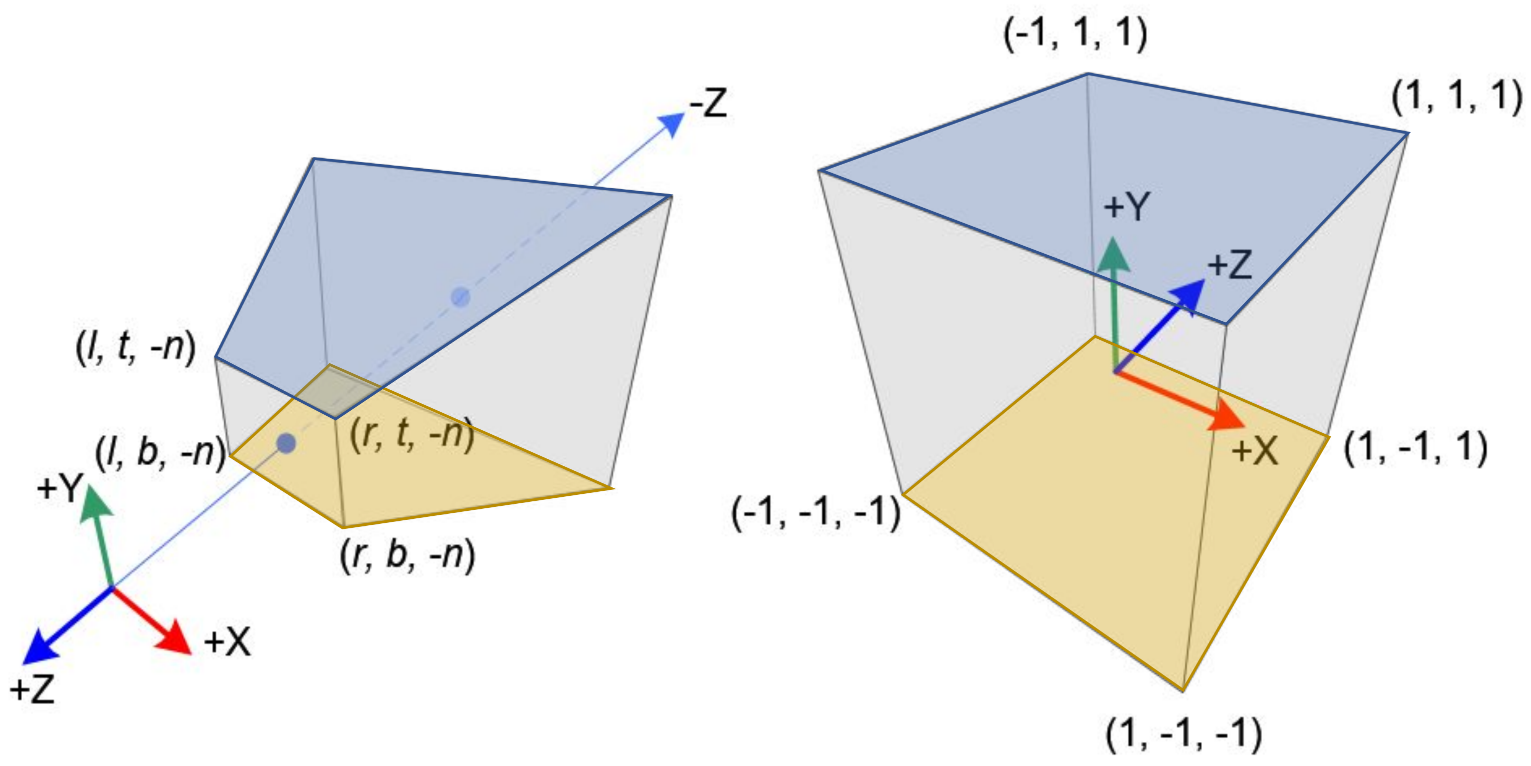


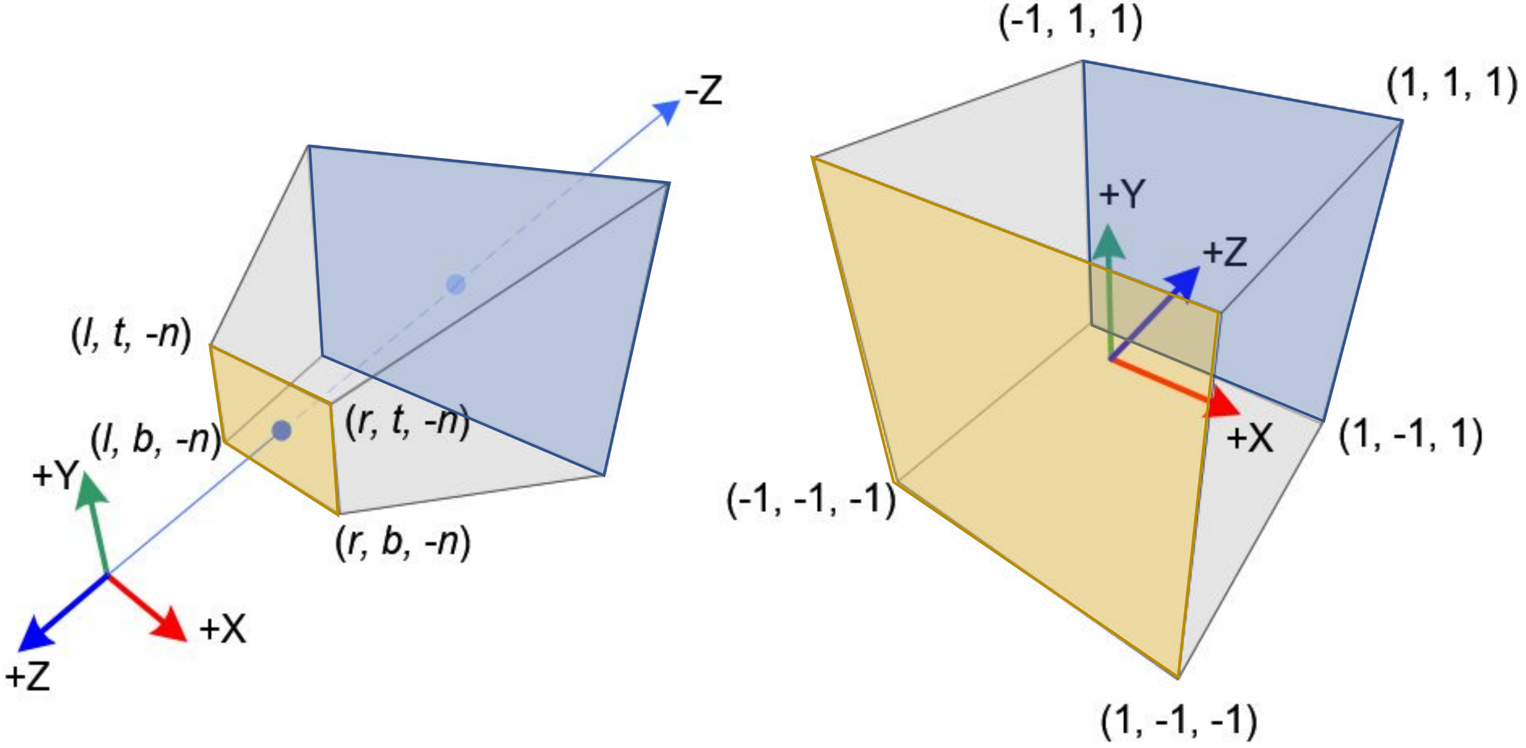




Normalized Device Coordinate



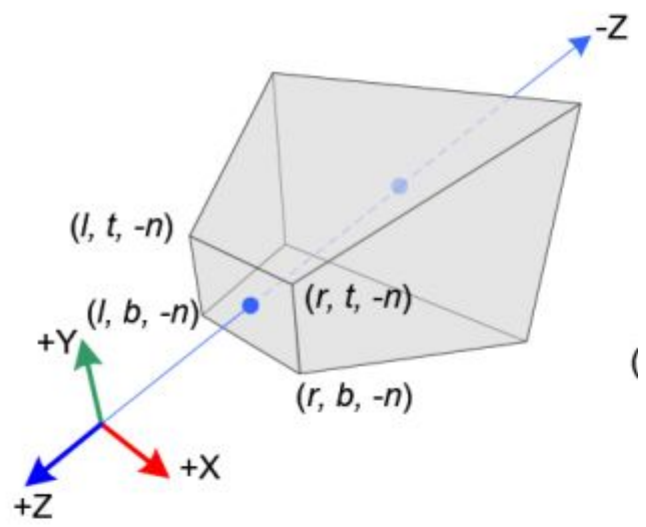
-Z
(l, t, -n)
(l, b, -n)
(r, t, -n)
(r, b, -n)
+Y
+X
+Z
(-1, 1, 1)
(1, 1, 1)
+Y
+Z
+X
(1, -1, 1)
(-1, -1, -1)
(1, -1, -1)


-Z
(l, t, -n)
(l, b, -n)
(r, t, -n)
(r, b, -n)
+Y
+X
+Z
(-1, 1, 1)
(1, 1, 1)
+Y
+Z
+X
(1, -1, 1)
(-1, -1, -1)
(1, -1, -1)

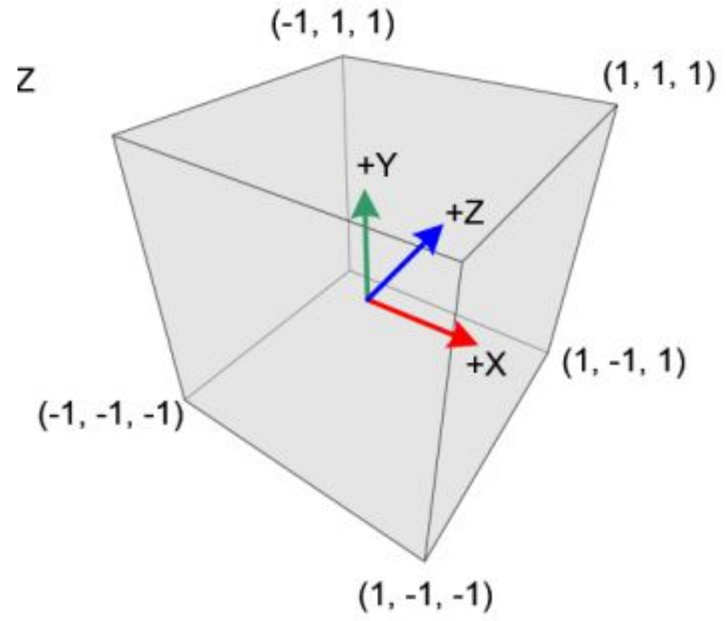


(l, t, -n)
(l, b, -n)
(r, t, -n)
(r, b, -n)
-Z
+Y
+X
+Z
(-1, 1, 1)
(1, 1, 1)
+Y
+Z
+X
(1, -1, 1)
(-1, -1, -1)
(1, -1, -1)

# Frustum to NDC
# Right vs Left




Right handed coordinate system




Left handed coordinate system

# The difference !

$$\begin{bmatrix} x_{eye} \\ y_{eye} \\ z_{eye} \\ 1 \end{bmatrix}$$

$$M_{perspective} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{eye} \\ y_{eye} \\ z_{eye} \\ 1 \end{bmatrix}$$

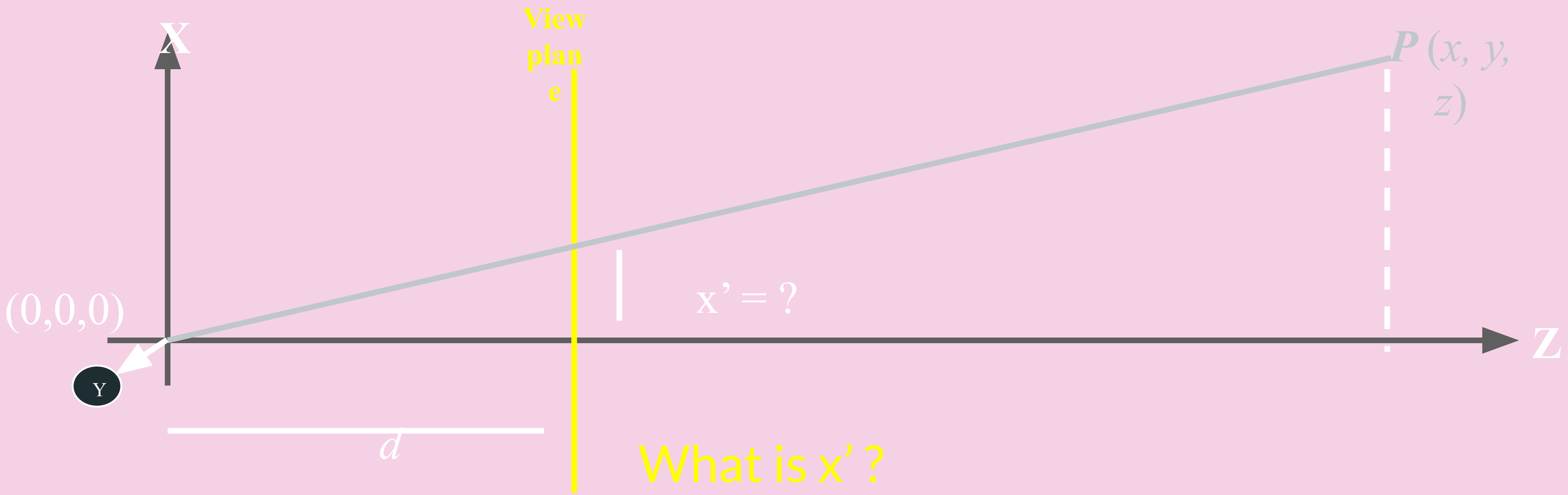
$$\cancel{\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}}$$

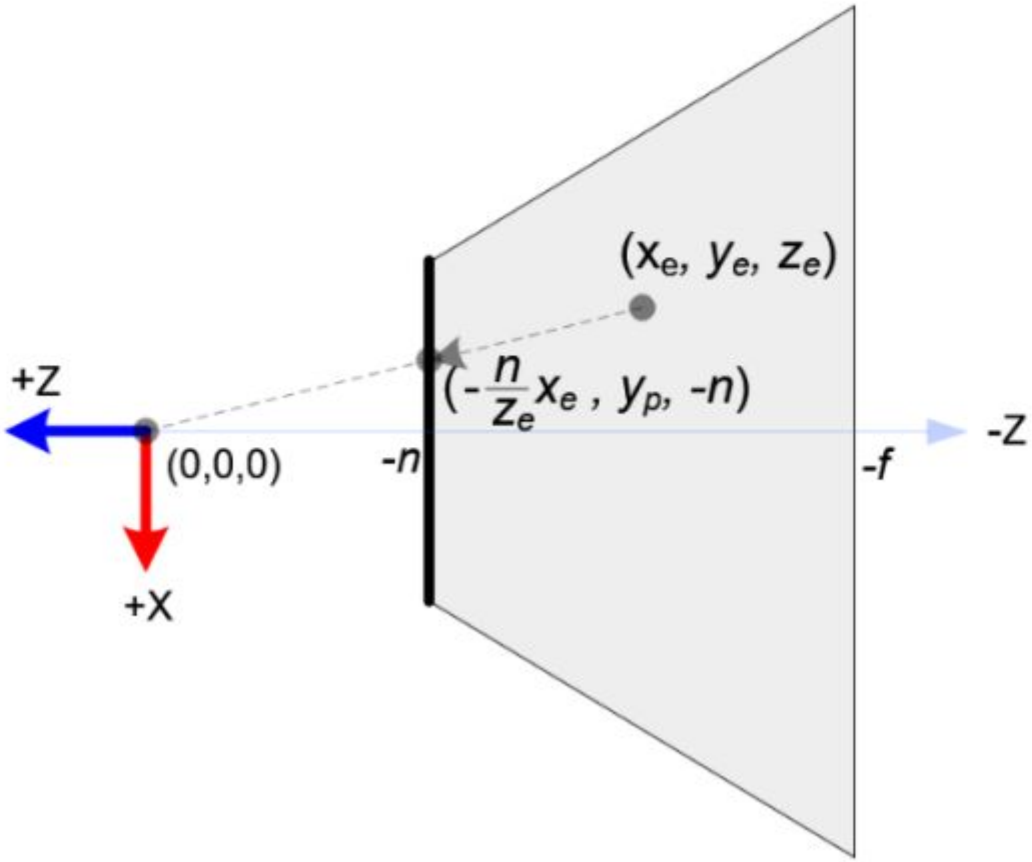
$$\begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix}$$

$$M'_{perspective} = ?$$

# Perspective Projection

The geometry of the situation is that of similar triangles. View from Y-axis:






$$\frac{x_p}{x_e} = \frac{-n}{z_e}$$

$$x_p = \frac{-n \cdot x_e}{z_e} = \frac{n \cdot x_e}{-z_e}$$

# Building the matrix

$$M_{perspective} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\begin{pmatrix} x_n \\ y_n \\ z_n \\ 1 \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \\ z_c \\ w_c \end{pmatrix} = \begin{pmatrix} x_c/w_c \\ y_c/w_c \\ z_c/w_c \\ 1 \end{pmatrix}$$

$w_c = -z_{eye}$

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ w_c \end{pmatrix} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x_e \\ y_e \\ z_e \\ w_e \end{pmatrix}$$

# Building the matrix

$-r <= x_p <= r$

$=> -1 <= x_p / r <= 1$

Therefore $x_n = x_{eye} * d / (-z_{eye} * r)$

Finally $x_c = x_{eye} * (d / r)$

# Building the matrix

$$\begin{pmatrix} d/r & 0 & 0 & 0 \\ 0 & d/t & 0 & 0 \\ ? & ? & ? & ? \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

# Building the matrix

$$\begin{pmatrix} d/r & 0 & 0 & 0 \\ 0 & d/t & 0 & 0 \\ 0 & 0 & A & B \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

# Building the matrix

$z_c = (A*z_{eye} + B)$

$z_n = (A*z_{eye} + B) / -z_{eye}$

Near corresponds to -1

Far corresponds to 1

-near = - A * near + B

far = - A * far + B

# Finally!!

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

# Concrete Proof

http://www.songho.ca/opengl/gl_projectionmatrix.html

# Full Pipeline

$$M = M_{projection} * M_{view} * M_{model}$$

# Thank You