CS559: Computer Graphics

Lecture 9: Rasterization

Li Zhang

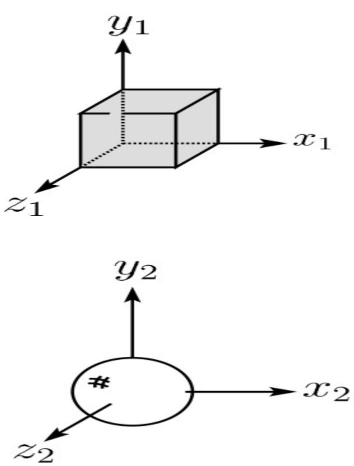
Spring 2008

Today

- Finish Projection
- Rasterization
- Antialising
- Hidden Surface Removal

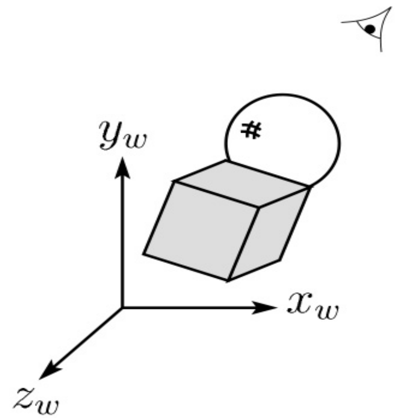
- Reading:
  - Shirley ch 7,4,8

3D Geometry Pipeline



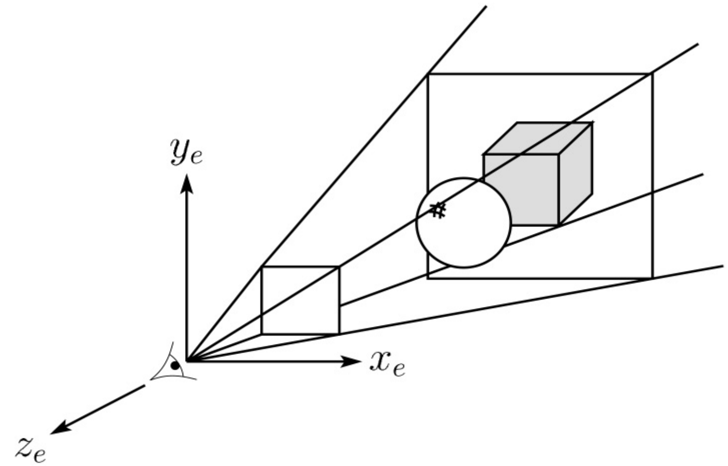
Model Space
(Object Space)

Rotation
Translation
Resizing



World Space

Rotation
Translation



Eye Space
(View Space)

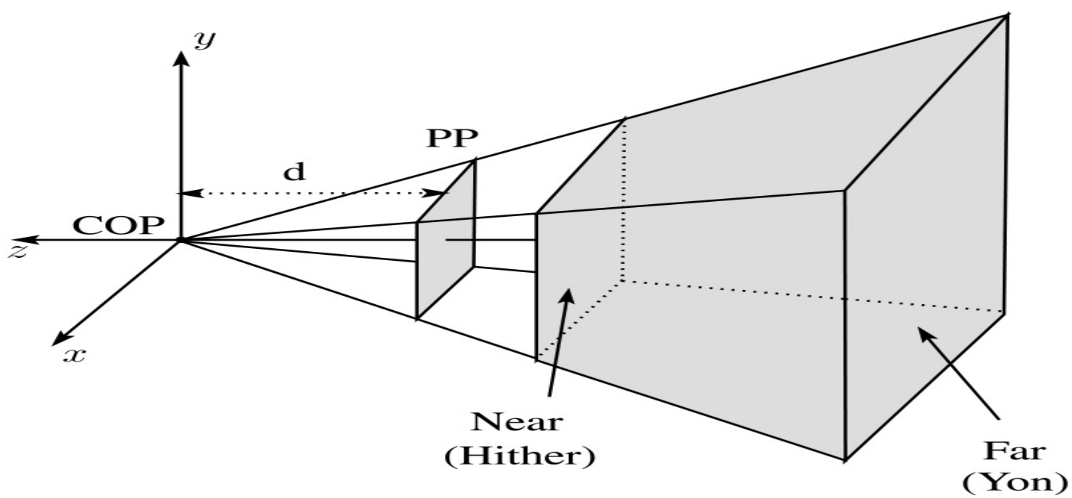
Projective,
Scale
Translation

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} (d/z)x \\ (d/z)y \\ d \\ 1 \end{bmatrix}$$

Clipping and the viewing frustum

The center of projection and the portion of the projection plane that map to the final image form an infinite pyramid. The sides of the pyramid are **clipping planes**.

Frequently, additional clipping planes are inserted to restrict the range of depths. These clipping planes are called the **near** and **far** or the **hither** and **yon** clipping planes.

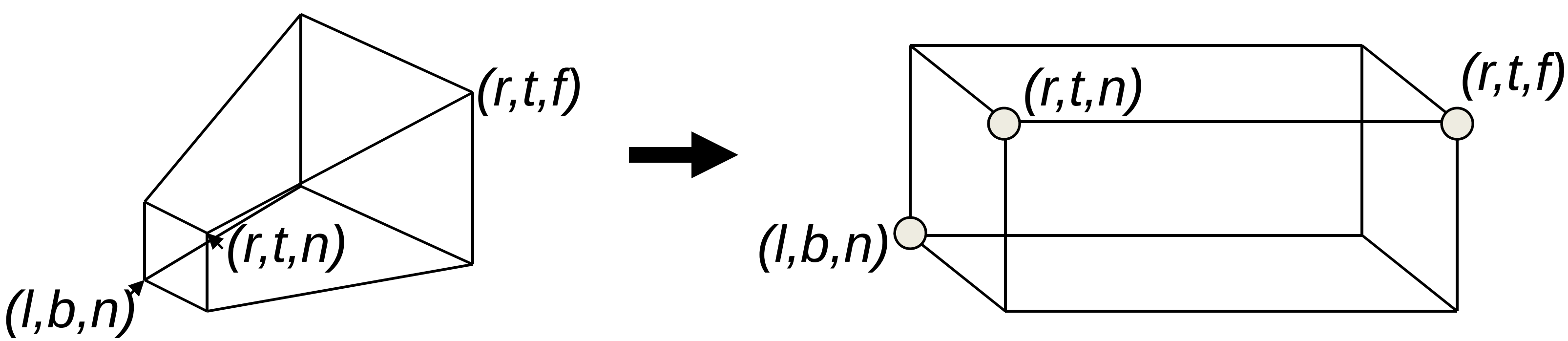


$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} (d/z)x \\ (d/z)y \\ d \\ 1 \end{bmatrix}$$

All of the clipping planes bound the **viewing frustum**.

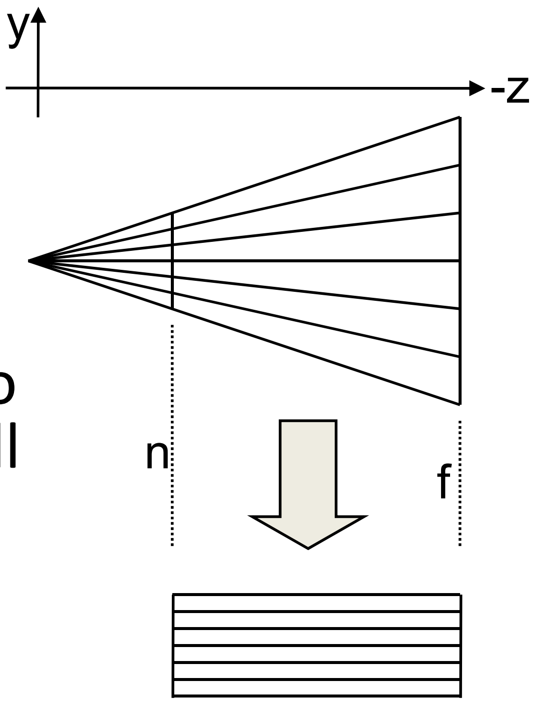
View Volume Transformation

- We want a matrix that will take points in our perspective view volume and transform them into the orthographic view volume
  - This matrix will go in our pipeline before an orthographic projection matrix



Mapping Lines

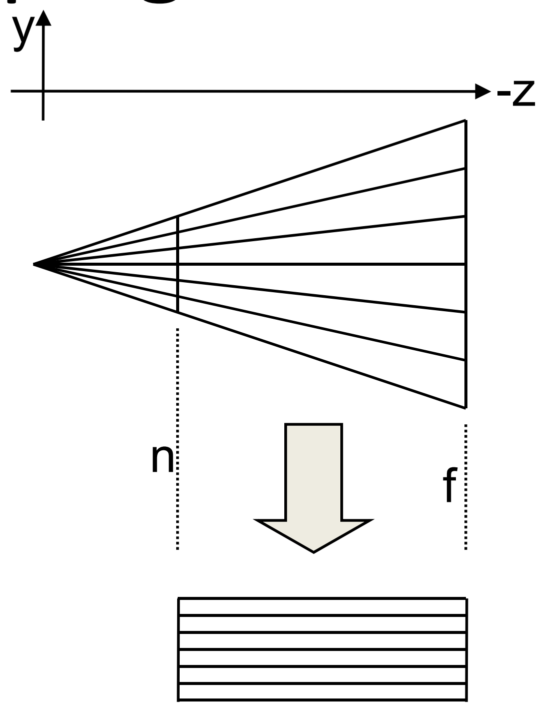
- We want to map all the lines through the center of projection to parallel lines
  - This converts the perspective case to the orthographic case, we can use all our existing methods
- The relative intersection points of lines with the near clip plane should not change
- The matrix that does this looks like the matrix for our simple perspective case



How to get this mapping?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \boxed{1 \quad 0} \\ 0 & 0 & 1/n & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} (n/z)x \\ (n/z)y \\ n \\ 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \boxed{A \quad B} \\ 0 & 0 & 1/n & 0 \end{bmatrix}$$

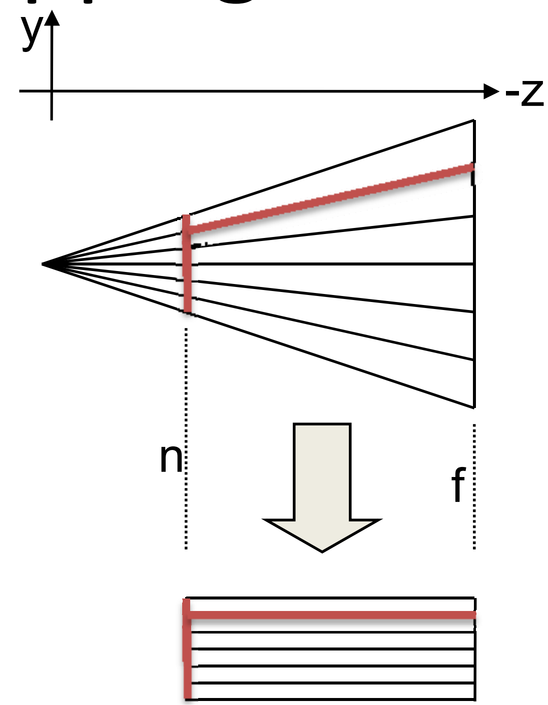


$$M \begin{bmatrix} x \\ y \\ n \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ \boxed{(An+B) \quad = n} \\ 1 \end{bmatrix} \qquad M \begin{bmatrix} x \\ y \\ f \\ 1 \end{bmatrix} = \begin{bmatrix} (n/f)x \\ (n/f)y \\ \boxed{(n/f)(Af+B) \quad = f} \\ 1 \end{bmatrix} \quad \Longrightarrow \quad \begin{aligned} A &= 1 + \frac{f}{n} \\ B &= -f \end{aligned}$$

Properties of this mapping

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \boxed{1 \quad 0} \\ 0 & 0 & 1/n & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} (n/z)x \\ (n/z)y \\ n \\ 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \boxed{(n+f)/n \quad -f} \\ 0 & 0 & 1/n & 0 \end{bmatrix}$$



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{nx}{z} \\ \frac{ny}{z} \\ f+n-\frac{fn}{z} \\ 1 \end{bmatrix}$$

If z = n, M(x,y,z,1) = [x,y,z,1]
near plane is unchanged

If x/z = c1, y/z = c2, then x'=n*c1, y'=n*c2
bending converging rays to parallel rays

If z1 < z2, z1' < z2'
z ordering is preserved

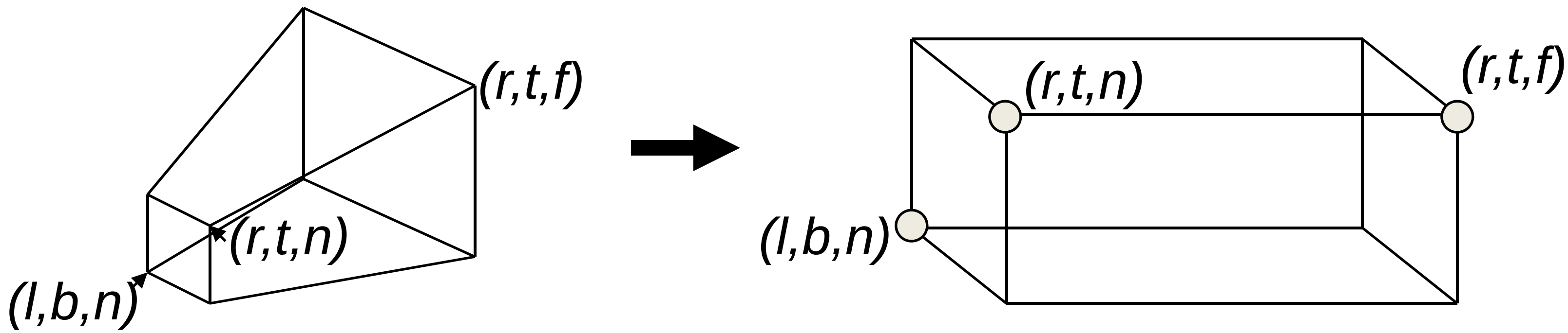
In pyramid ⇔ In cube

General Perspective

$$\mathbf{M}_{View \to OrthoView} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & (n+f)/n & -f \\ 0 & 0 & 1/n & 0 \end{bmatrix} \equiv \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -nf \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

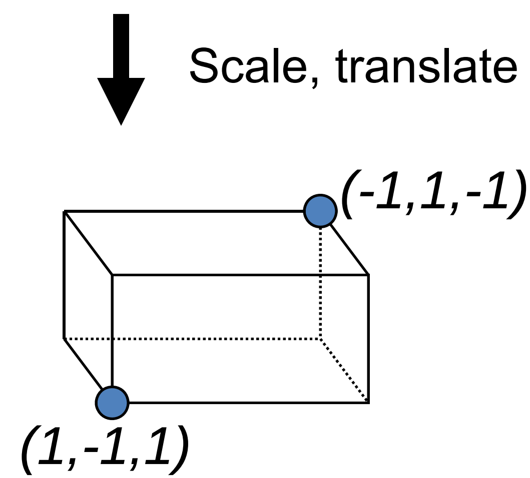
To Canonical view volume

- Perspective projection transforms a pyramid volume to an orthographic view volume



$$\mathbf{M}_{View \rightarrow OrthoView}$$

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



Canonical view volume [-1,1]^3

Orthographic View to Canonical Matrix

$$\begin{bmatrix} x_{canonical} \\ y_{canonical} \\ z_{canonical} \\ 1 \end{bmatrix} = \begin{bmatrix} 2/(r-l) & 0 & 0 & 0 \\ 0 & 2/(t-b) & 0 & 0 \\ 0 & 0 & 2/(n-f) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -(r+l)/2 \\ 0 & 1 & 0 & -(t+b)/2 \\ 0 & 0 & 1 & -(n+f)/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{orthoView} \\ y_{orthoView} \\ z_{orthoView} \\ 1 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} 2/(r-l) & 0 & 0 & -(r+l)/(r-l) \\ 0 & 2/(t-b) & 0 & -(t+b)/(t-b) \\ 0 & 0 & 2/(n-f) & -(n+f)/(n-f) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{orthoView} \\ y_{orthoView} \\ z_{orthoView} \\ 1 \end{bmatrix}}$$

$$\mathbf{x}_{canonical} = \mathbf{M}_{orthoView->canonical} \mathbf{x}_{orthoView}$$

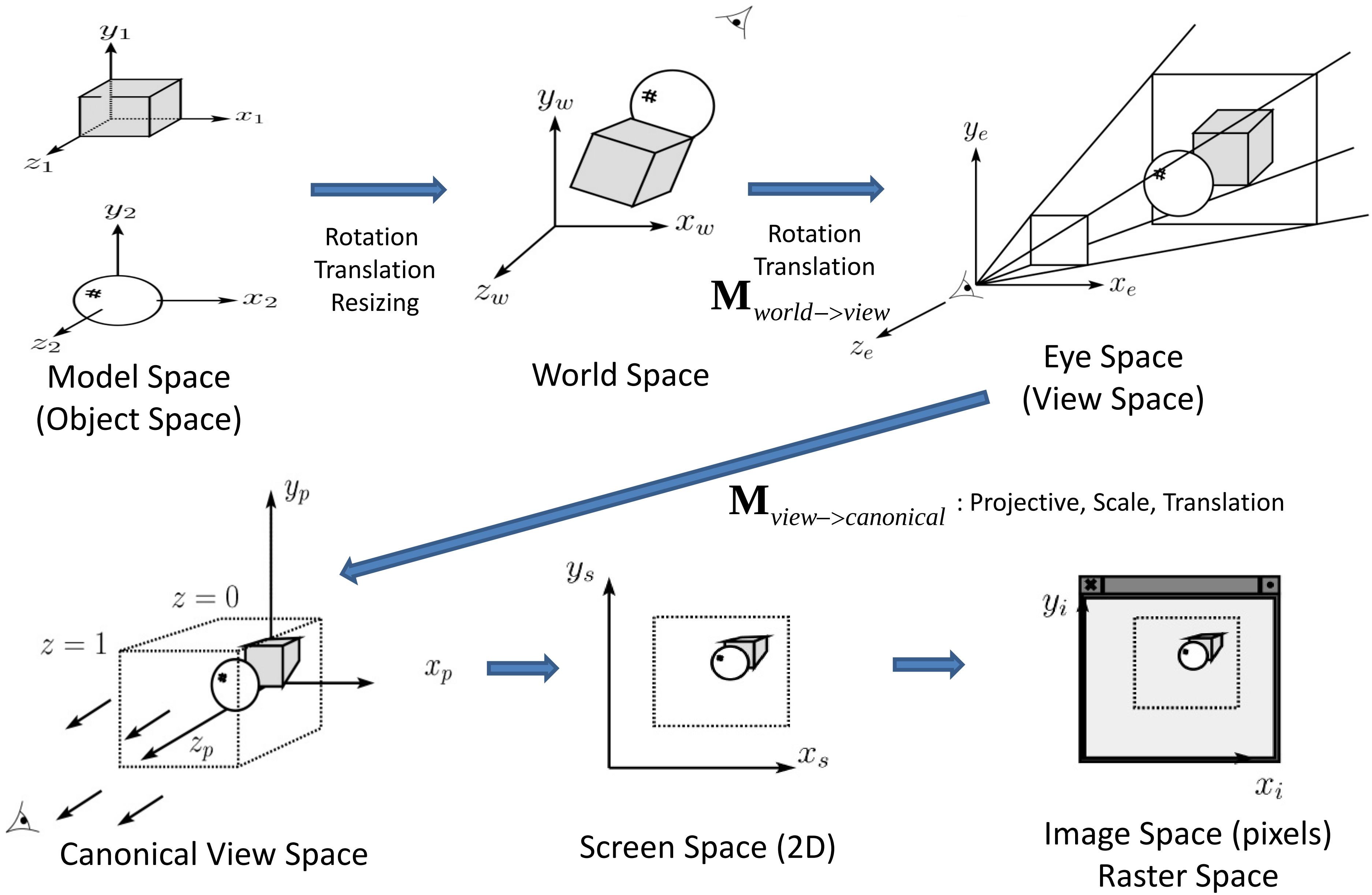
Complete Perspective Projection

- After applying the perspective matrix, we map the orthographic view volume to the canonical view volume:

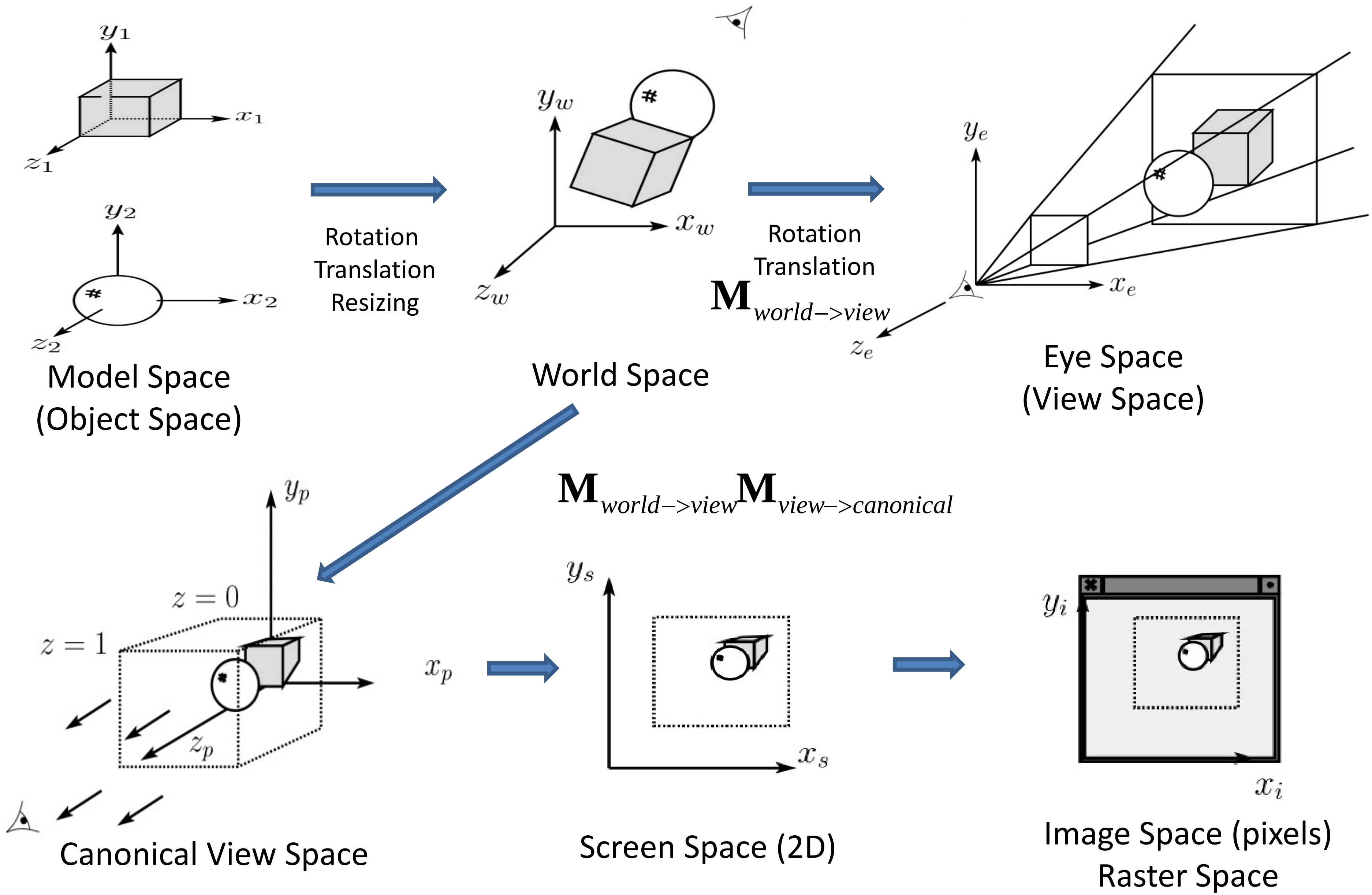
$$\mathbf{M}_{view->canonical} = \mathbf{M}_{orthoView \rightarrow Canonical} \mathbf{M}_{view \rightarrow orthoView}$$

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

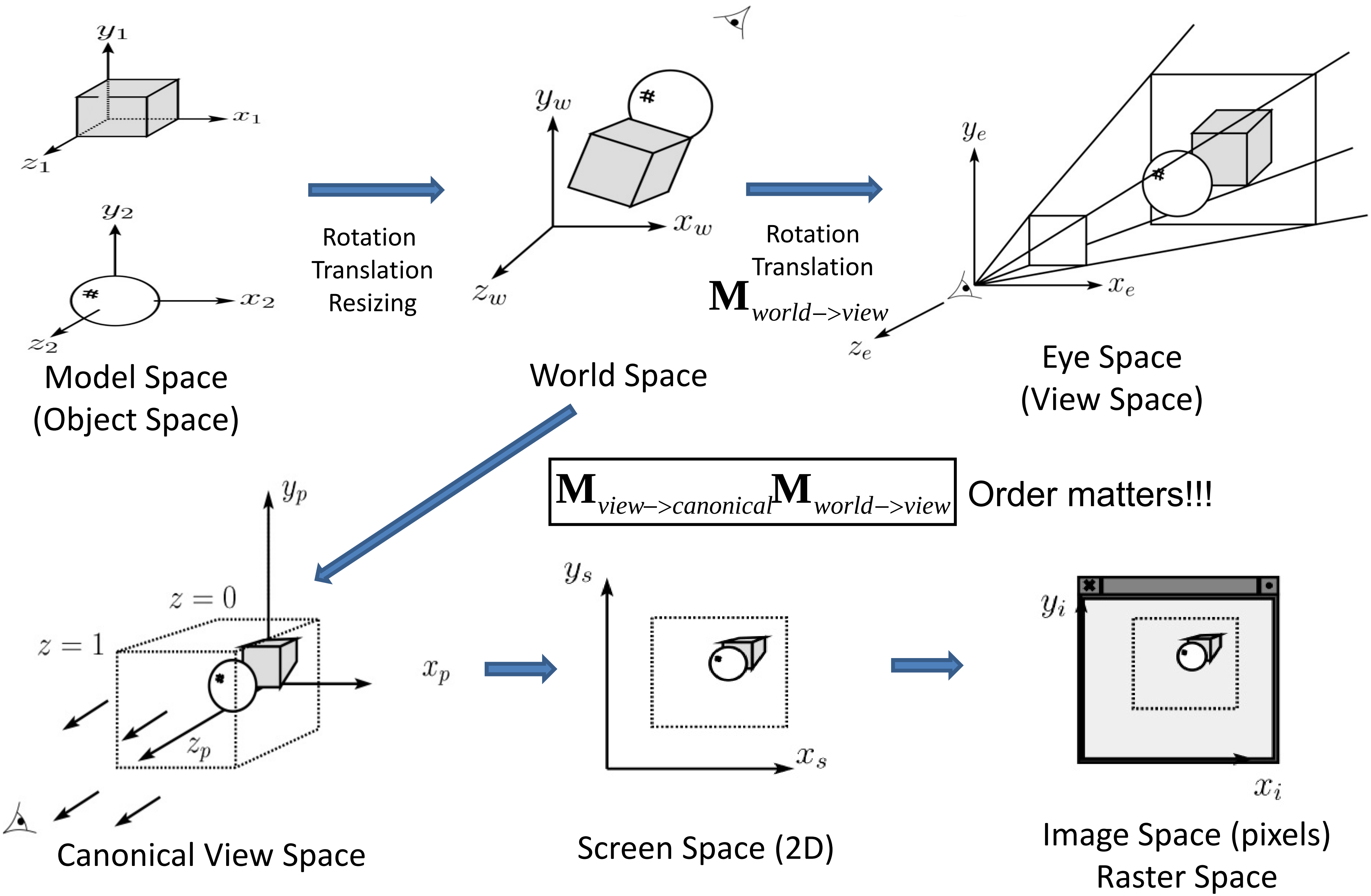
3D Geometry Pipeline



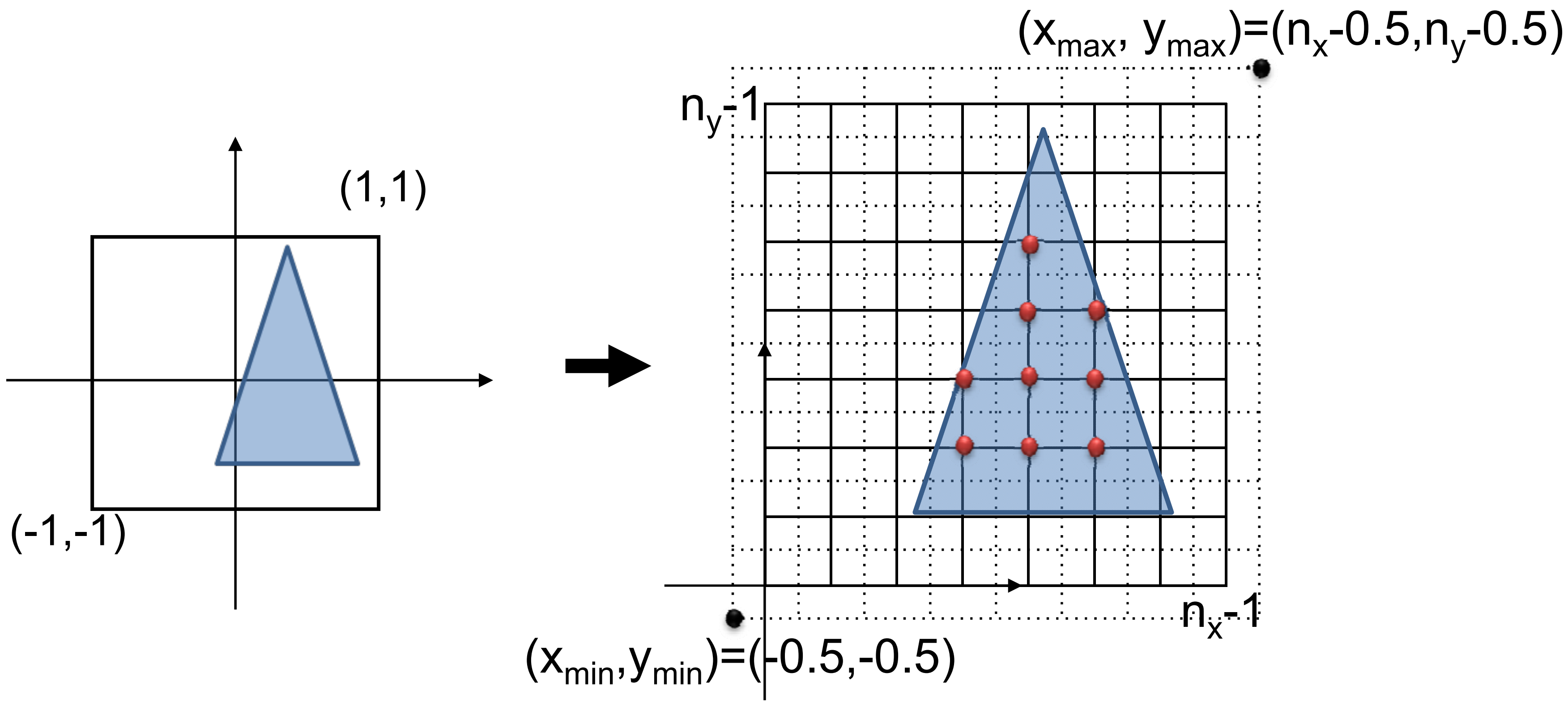
3D Geometry Pipeline



3D Geometry Pipeline

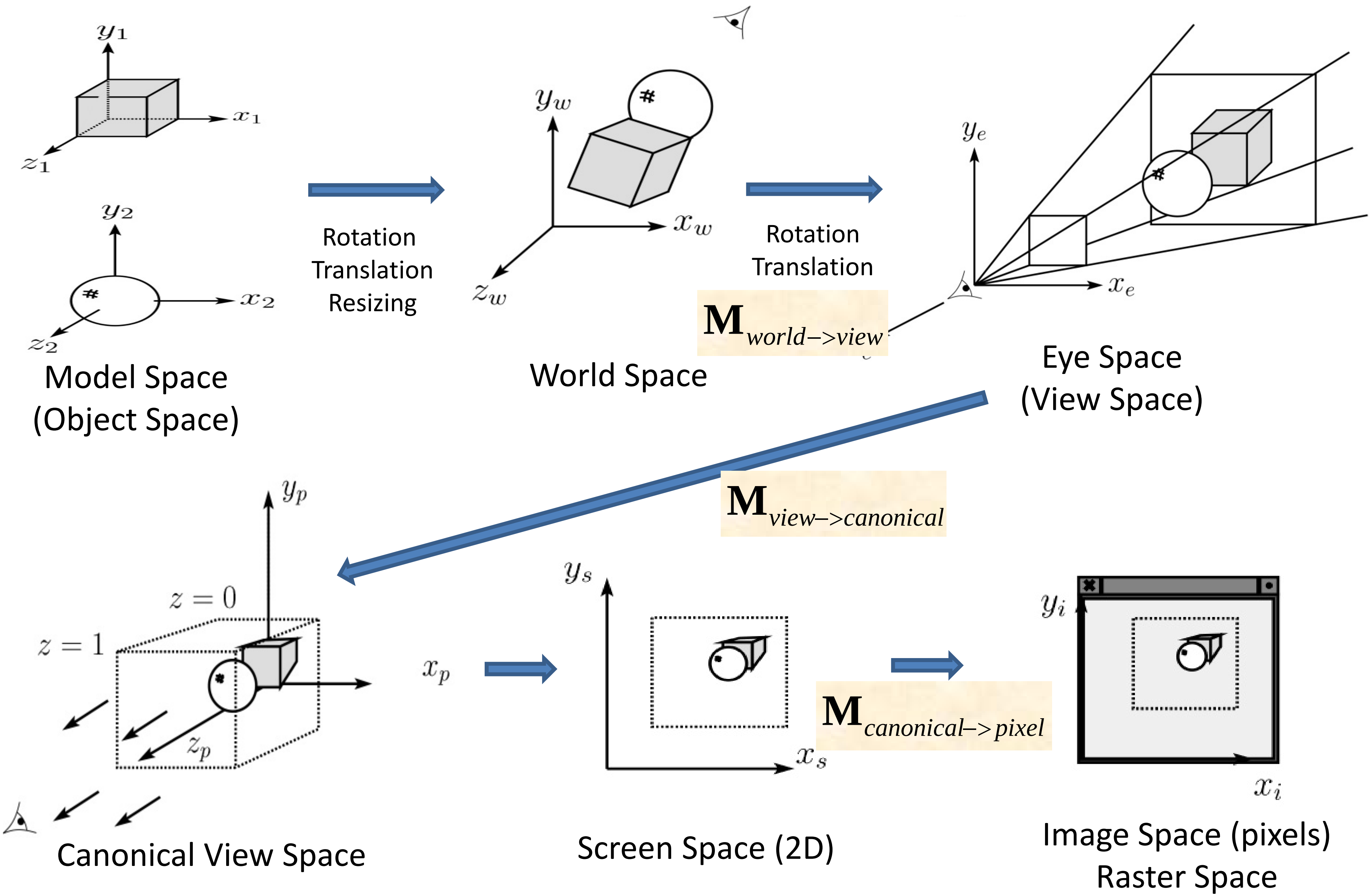


Canonical → Window Transform



$$\underbrace{\begin{bmatrix} x_{pixel} \\ y_{pixel} \\ z_{pixel} \\ 1 \end{bmatrix} = \begin{bmatrix} (x_{\max}-x_{\min})/2 & 0 & 0 & (x_{\max}+x_{\min})/2 \\ 0 & (y_{\max}-y_{\min})/2 & 0 & (y_{\max}+y_{\min})/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{M}_{canonical->pixel}} \begin{bmatrix} x_{canonical} \\ y_{canonical} \\ z_{canonical} \\ 1 \end{bmatrix}$$

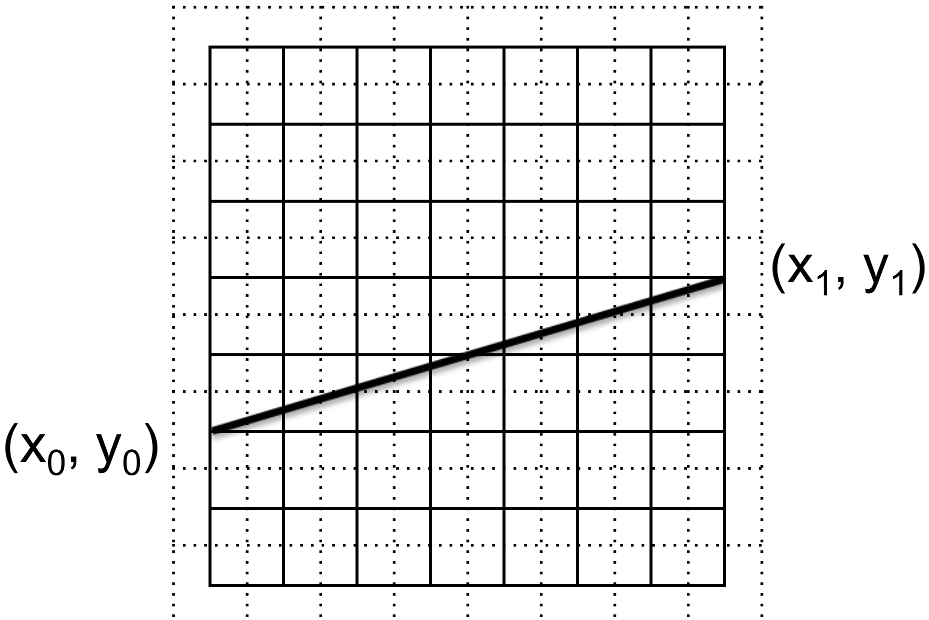
3D Geometry Pipeline



Polygons are basic shape primitives

- Any 3D shape can be approximated by a polygon using a locally linear (planar) approximation. To improve the quality of fit, we need only increase the number of edges

Line Drawing

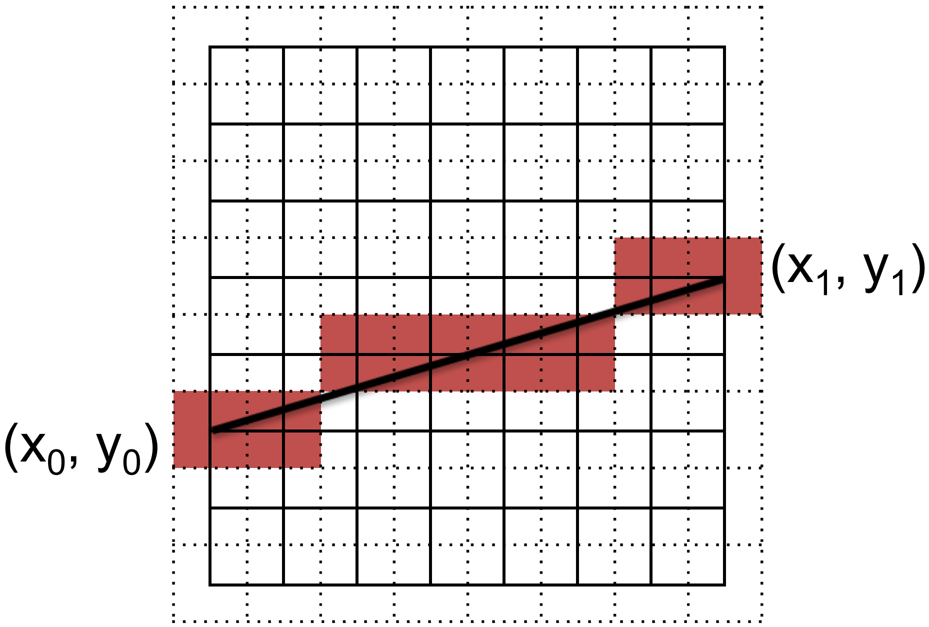


$$y = \frac{\Delta y}{\Delta x}(x - x_0) + y_0$$

$$\Delta x = x_1 - x_0, \Delta y = y_1 - y_0$$

Line Drawing

for x = x0 to x1 **do**
  y=y0+(Δy/Δx)(x-x0)
  draw(x,round(y))

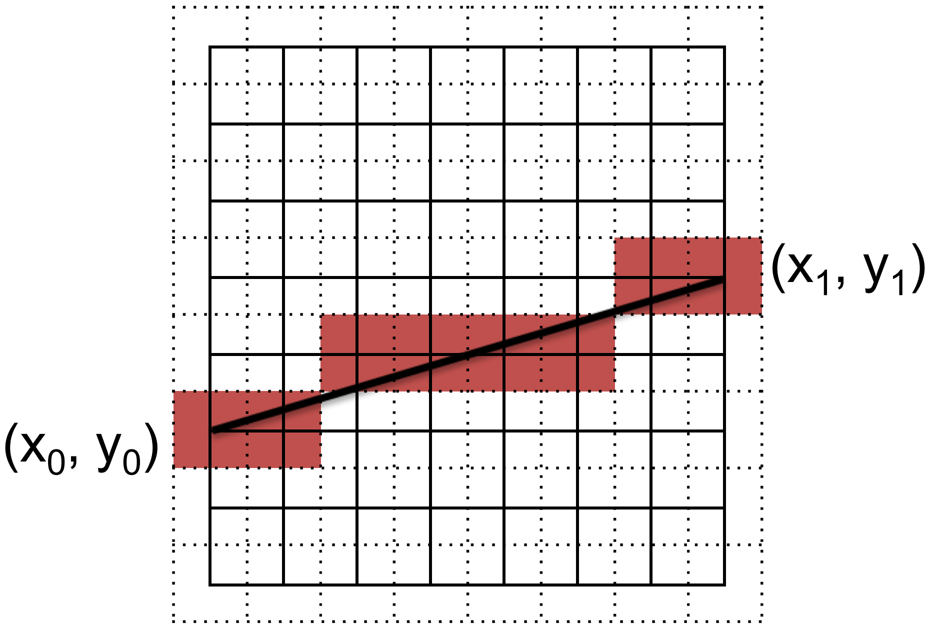


$$y = \frac{\Delta y}{\Delta x}(x - x_0) + y_0$$

$$\Delta x = x_1 - x_0, \Delta y = y_1 - y_0$$

Line Drawing

```
y = y0
for x = x0 to x1 do
    draw(x,y)
    if (some condition) then
        y=y+1
```

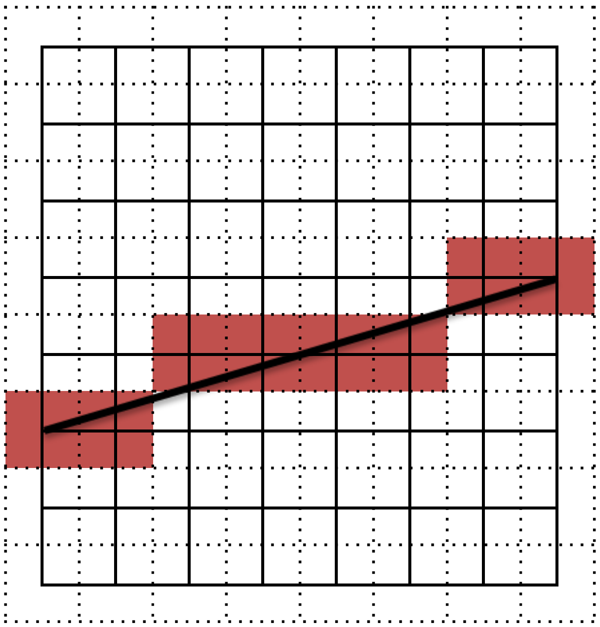


$$y = \frac{\Delta y}{\Delta x}(x - x_0) + y_0$$

$$\Delta x = x_1 - x_0, \Delta y = y_1 - y_0$$

Implicit Line Drawing

```
y = y0
for x = x0 to x1 do
    draw(x,y)
    if f(x+1,y+0.5)<0 ) then
       y=y+1
```



$$y = \frac{\Delta y}{\Delta x}(x - x_0) + y_0$$

$$\Delta x = x_1 - x_0, \Delta y = y_1 - y_0$$

$$f(x, y) = \Delta x \cdot (y - y_0) - \Delta y \cdot (x - x_0)$$

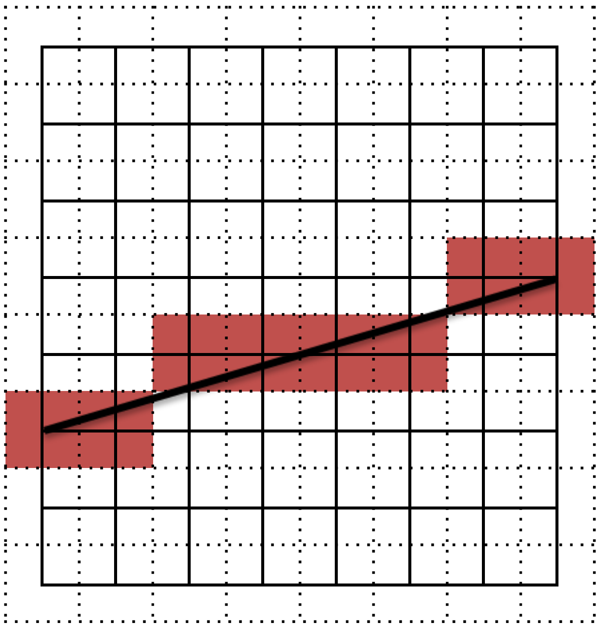
$$f(x, y) = 0 \Rightarrow \text{on the line}$$

$$f(x, y) > 0 \Rightarrow \text{above the line}$$

$$f(x, y) < 0 \Rightarrow \text{below the line}$$

Implicit Line Drawing

```
y = y0
for x = x0 to x1 do
    draw(x,y)
    if f(x+1,y+0.5)<0 ) then
        y=y+1
```



$$y = \frac{\Delta y}{\Delta x}(x - x_0) + y_0$$

$$\Delta x = x_1 - x_0, \Delta y = y_1 - y_0$$

$$f(x, y) = \Delta x \cdot (y - y_0) - \Delta y \cdot (x - x_0)$$

Incremental Evaluation

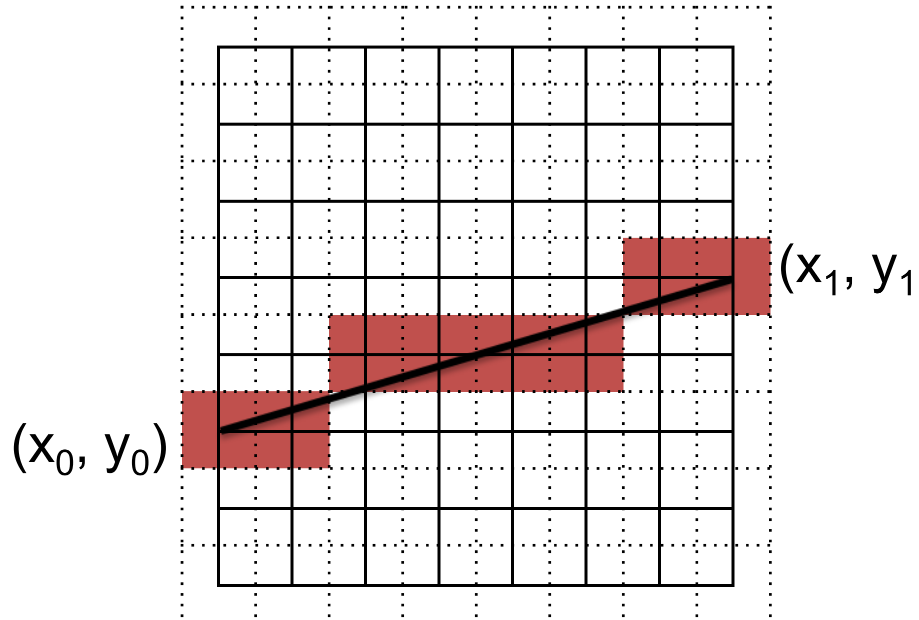
$$f(x+1, y) = f(x, y) - \Delta y$$

$$f(x+1, y+1) = f(x, y) - \Delta y + \Delta x$$

Implicit Line Drawing

```
d=f(x+1,y+0.5)
y = y0
for x = x0 to x1 do
    draw(x,y)
    if d<0 ) then
        D=d-Δy+Δx
        y=y+1
    else
        D=d-Δy
```

Incremental Evaluation



$$y = \frac{\Delta y}{\Delta x}(x - x_0) + y_0$$

$$\Delta x = x_1 - x_0, \Delta y = y_1 - y_0$$

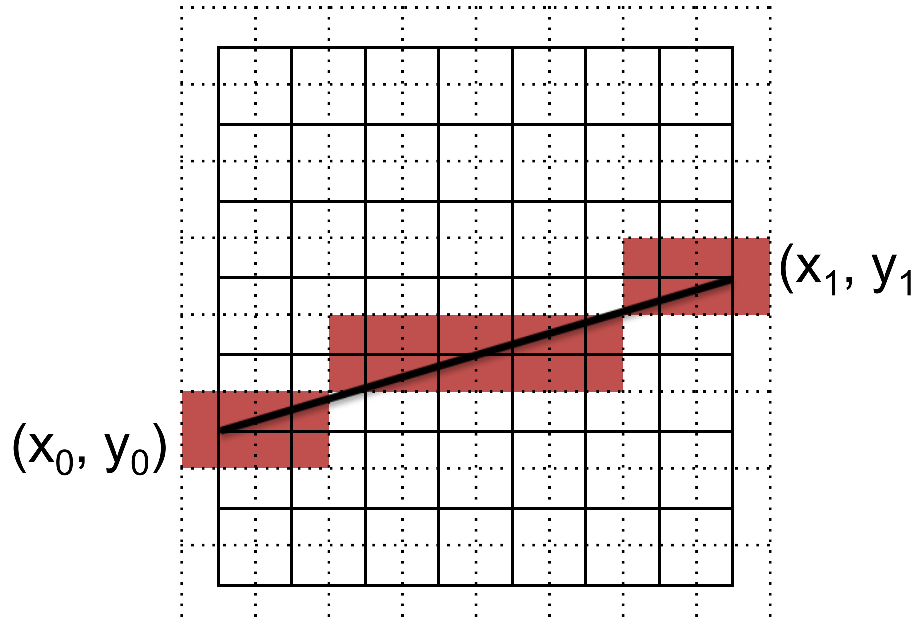
$$f(x, y) = \Delta x \cdot (y - y_0) - \Delta y \cdot (x - x_0)$$

$$f(x+1, y) = f(x, y) - \Delta y$$

$$f(x+1, y+1) = f(x, y) - \Delta y + \Delta x$$

Implicit Line Drawing

```
d=2f(x+1,y+0.5)
y = y0
for x = x0 to x1 do
    draw(x,y)
    if d<0 ) then
        D=d-2Δy+2Δx
        y=y+1
    else
        D=d-2Δy
```



$$y = \frac{\Delta y}{\Delta x}(x - x_0) + y_0$$

$$\Delta x = x_1 - x_0, \Delta y = y_1 - y_0$$

$$f(x, y) = \Delta x \cdot (y - y_0) - \Delta y \cdot (x - x_0)$$

$$f(x+1, y) = f(x, y) - \Delta y$$

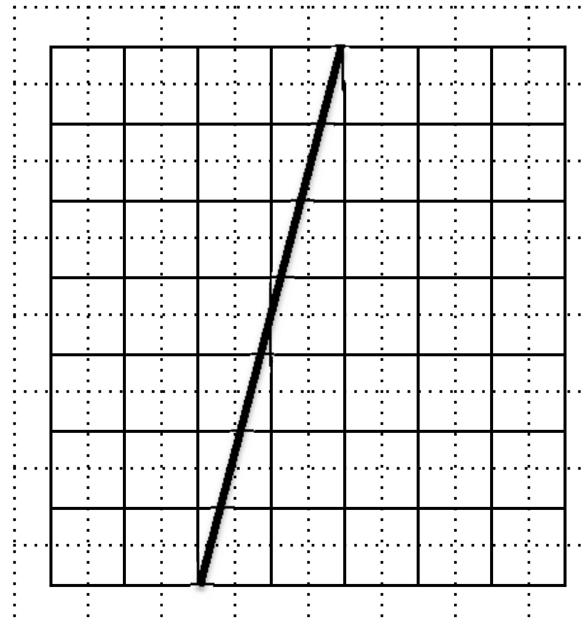
$$f(x+1, y+1) = f(x, y) - \Delta y + \Delta x$$

Incremental Evaluation

Integer operation

Implicit Line Drawing

```
d=2f(x+1,y+0.5)
y = y0
for x = x0 to x1 do
    draw(x,y)
    if d<0 ) then
        D=d-2Δy+2Δx
        y=y+1
    else
        D=d-2Δy
```



$$y = \frac{\Delta y}{\Delta x}(x - x_0) + y_0$$

$$\Delta x = x_1 - x_0, \Delta y = y_1 - y_0$$

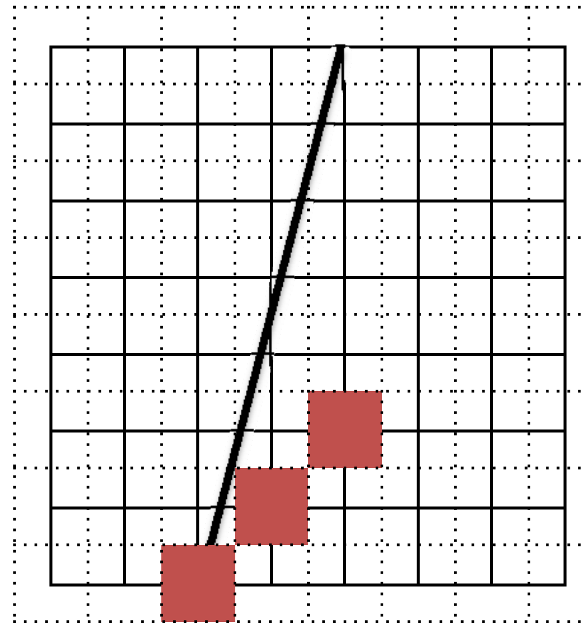
$$f(x, y) = \Delta x \cdot (y - y_0) - \Delta y \cdot (x - x_0)$$

$$f(x+1, y) = f(x, y) - \Delta y$$

$$f(x+1, y+1) = f(x, y) - \Delta y + \Delta x$$

Implicit Line Drawing

```
d=2f(x+1,y+0.5)
y = y0
for x = x0 to x1 do
    draw(x,y)
    if d<0 ) then
        D=d-2Δy+2Δx
        y=y+1
    else
        D=d-2Δy
```



$$y = \frac{\Delta y}{\Delta x}(x - x_0) + y_0$$

$$\Delta x = x_1 - x_0, \Delta y = y_1 - y_0$$

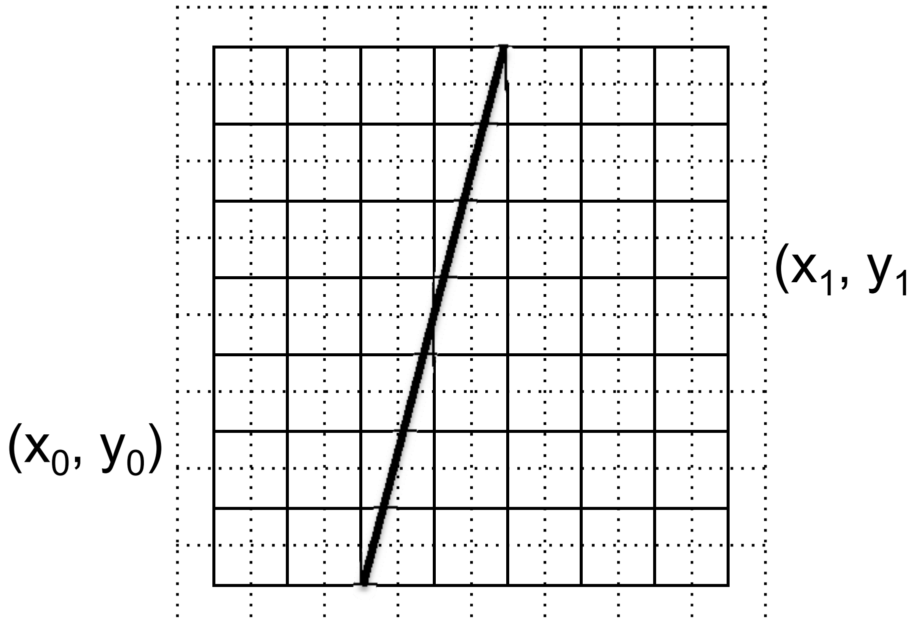
$$f(x, y) = \Delta x \cdot (y - y_0) - \Delta y \cdot (x - x_0)$$

$$f(x+1, y) = f(x, y) - \Delta y$$

$$f(x+1, y+1) = f(x, y) - \Delta y + \Delta x$$

Implicit Line Drawing

```
y = y0
for x = x0 to x1 do
    draw(x,y)
    if (some condition) then
        y=y+1
```



$$y = \frac{\Delta y}{\Delta x}(x - x_0) + y_0$$

$$\Delta x = x_1 - x_0, \Delta y = y_1 - y_0$$

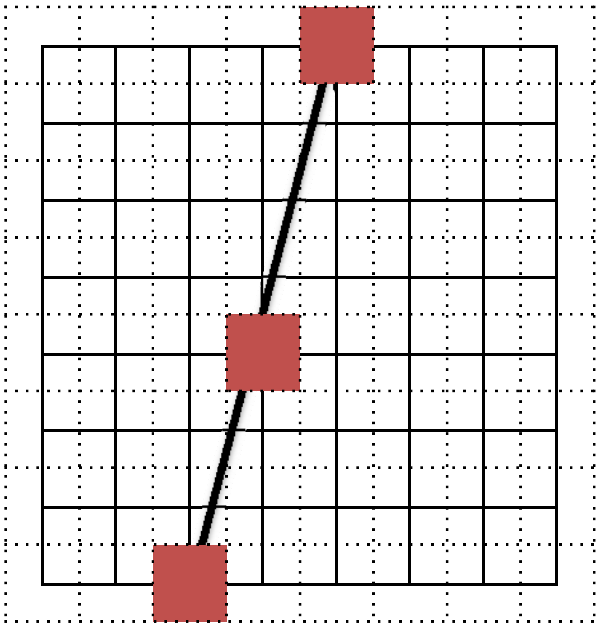
$$f(x, y) = \Delta x \cdot (y - y_0) - \Delta y \cdot (x - x_0)$$

$$f(x+1, y) = f(x, y) - \Delta y$$

$$f(x+1, y+1) = f(x, y) - \Delta y + \Delta x$$

Implicit Line Drawing

```
y = y0
for x = x0 to x1 do
    draw(x,y)
    if (some condition) then
        y=y+1
```



$$y = \frac{\Delta y}{\Delta x}(x - x_0) + y_0$$

$$\Delta x = x_1 - x_0, \Delta y = y_1 - y_0$$

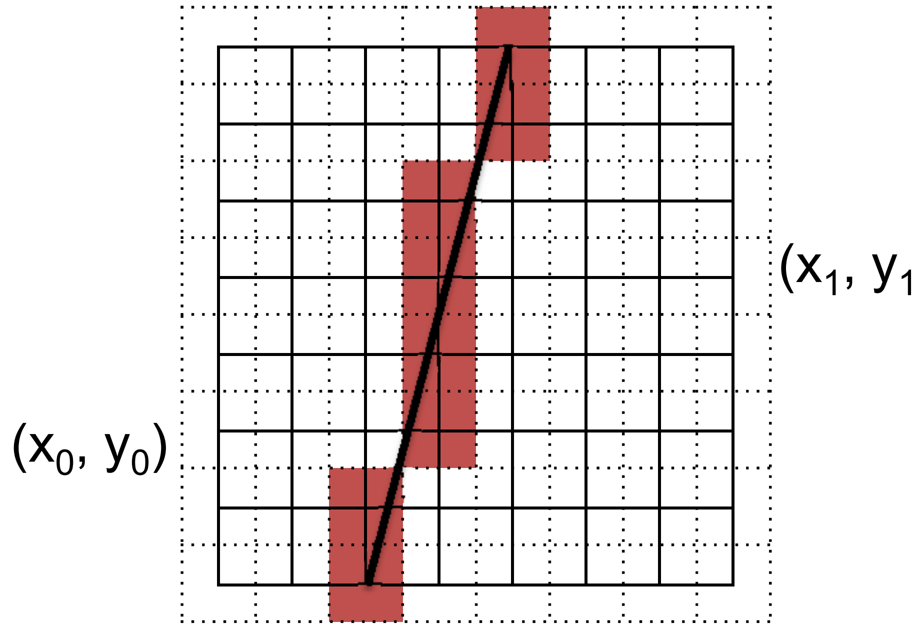
$$f(x, y) = \Delta x \cdot (y - y_0) - \Delta y \cdot (x - x_0)$$

$$f(x+1, y) = f(x, y) - \Delta y$$

$$f(x+1, y+1) = f(x, y) - \Delta y + \Delta x$$

Implicit Line Drawing

```
d=2f(x+1,y+0.5)
y = y0
for x = x0 to x1 do
    draw(x,y)
    if d<0 ) then
        D=d-2Δy+2Δx
        y=y+1
    else
        D=d-2Δy
```



$$y = \frac{\Delta y}{\Delta x}(x - x_0) + y_0$$

$$\Delta x = x_1 - x_0, \Delta y = y_1 - y_0$$

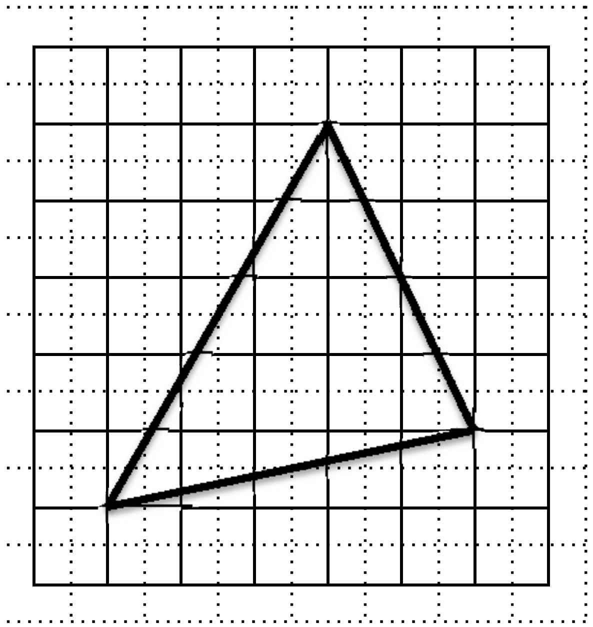
$$f(x, y) = \Delta x \cdot (y - y_0) - \Delta y \cdot (x - x_0)$$

If $\Delta y > \Delta x$, switch x and y

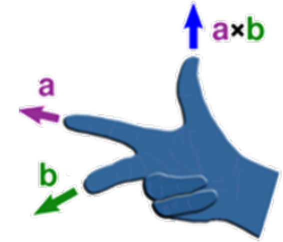
$$f(x+1, y) = f(x, y) - \Delta y$$

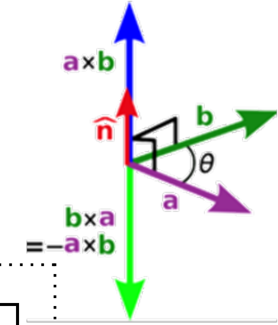
$$f(x+1, y+1) = f(x, y) - \Delta y + \Delta x$$

Drawing Triangle

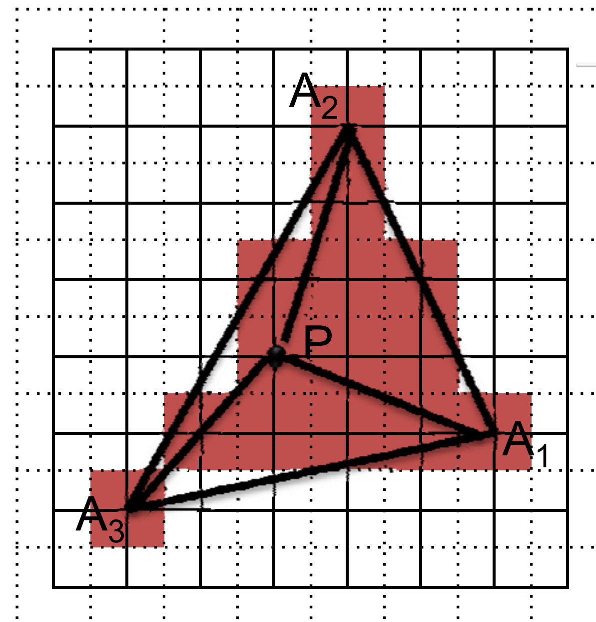


Drawing Triangle





for all y **do**
  for all x **do**
    if (in triangle) **then**
      draw(x,y) with color c



$A_1 : (x_1, y_1), c_1$
$A_2 : (x_2, y_2), c_2$
$A_3 : (x_3, y_3), c_3$

$$\begin{bmatrix} x_P \\ y_P \end{bmatrix} = t_1 \cdot \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + t_2 \cdot \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + t_3 \cdot \begin{bmatrix} x_3 \\ y_3 \end{bmatrix}$$

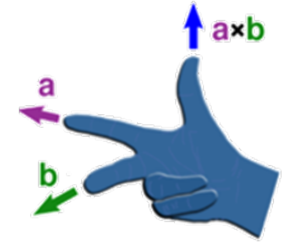
$$t_1 + t_2 + t_3 = 1$$

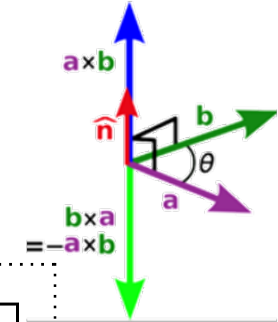
$$area(PA_2A_3) = \frac{1}{2} \overrightarrow{PA_2} \times \overrightarrow{PA_3}$$

$$t_1 = \frac{area(PA_2A_3)}{area(A_1A_2A_3)} \quad t_2 = \frac{area(PA_3A_1)}{area(A_1A_2A_3)} \quad t_3 = \frac{area(PA_1A_2)}{area(A_1A_2A_3)}$$

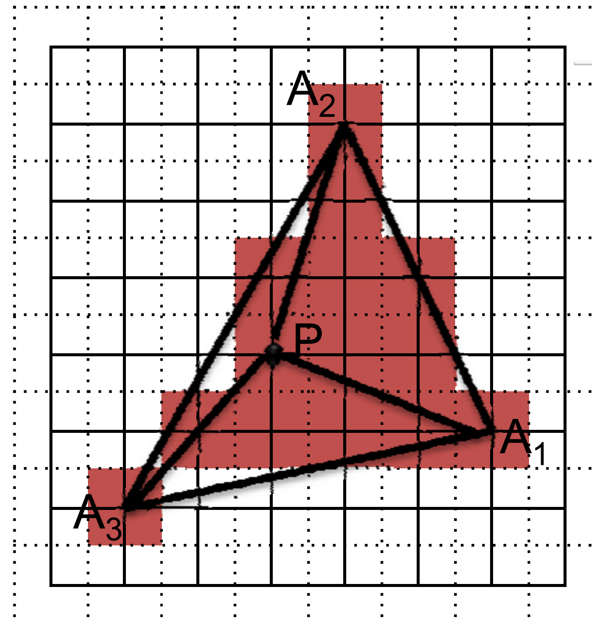
$$\left|area(PA_2A_3)\right| = \frac{1}{2} \left|\overrightarrow{PA_2}\right| \cdot \left|\overrightarrow{PA_3}\right| \sin(\angle A_2PA_3)$$

Drawing Triangle





for all y **do**
    for all x **do**
        compute t_1,t_2,t_3
        if t_1,t_2,t_3 in [0,1] **then**
            draw(x,y) with color
            $t_1c_1+t_2c_2+t_3c_3$



$$A_1 : (x_1, y_1),\ c_1$$
$$A_2 : (x_2, y_2),\ c_2$$
$$A_3 : (x_3, y_3),\ c_3$$

$$\begin{bmatrix} x_P \\ y_P \end{bmatrix} = t_1 \cdot \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + t_2 \cdot \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + t_3 \cdot \begin{bmatrix} x_3 \\ y_3 \end{bmatrix}$$

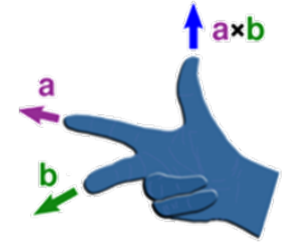
$$t_1 + t_2 + t_3 = 1$$

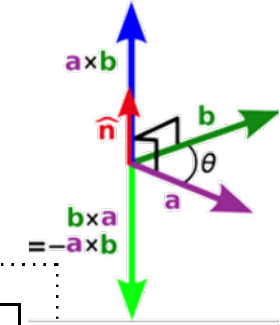
$$area(PA_2A_3) = \frac{1}{2}\overrightarrow{PA_2} \times \overrightarrow{PA_3}$$

$$t_1 = \frac{area(PA_2A_3)}{area(A_1A_2A_3)} \quad t_2 = \frac{area(PA_3A_1)}{area(A_1A_2A_3)} \quad t_3 = \frac{area(PA_1A_2)}{area(A_1A_2A_3)}$$

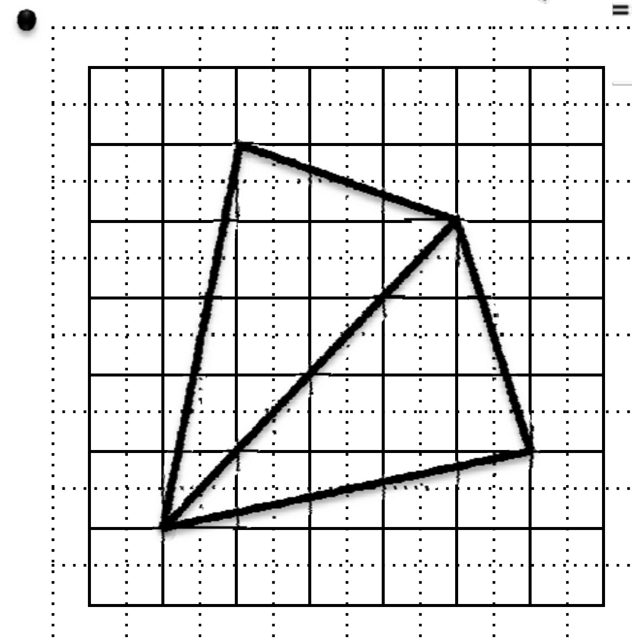
$$\left|area(PA_2A_3)\right| = \frac{1}{2}\left|\overrightarrow{PA_2}\right| \cdot \left|\overrightarrow{PA_3}\right| \sin(\angle A_2PA_3)$$

Common Edge





for all y **do**
  for all x **do**
    compute t_1,t_2,t_3
    if t_1,t_2,t_3 in [0,1] **then**
      draw(x,y) with color
      $t_1c_1+t_2c_2+t_3c_3$



$A_1 : (x_1, y_1),\ c_1$

$A_2 : (x_2, y_2),\ c_2$

$A_3 : (x_3, y_3),\ c_3$

$$\begin{bmatrix} x_P \\ y_P \end{bmatrix} = t_1 \cdot \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + t_2 \cdot \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + t_3 \cdot \begin{bmatrix} x_3 \\ y_3 \end{bmatrix}$$

$$t_1 + t_2 + t_3 = 1$$

$$area(PA_2A_3) = \frac{1}{2} \overrightarrow{PA_2} \times \overrightarrow{PA_3}$$

$$t_1 = \frac{area(PA_2A_3)}{area(A_1A_2A_3)} \quad t_2 = \frac{area(PA_3A_1)}{area(A_1A_2A_3)} \quad t_3 = \frac{area(PA_1A_2)}{area(A_1A_2A_3)}$$

$$|area(PA_2A_3)| = \frac{1}{2} \left|\overrightarrow{PA_2}\right| \cdot \left|\overrightarrow{PA_3}\right| \sin(\angle A_2PA_3)$$