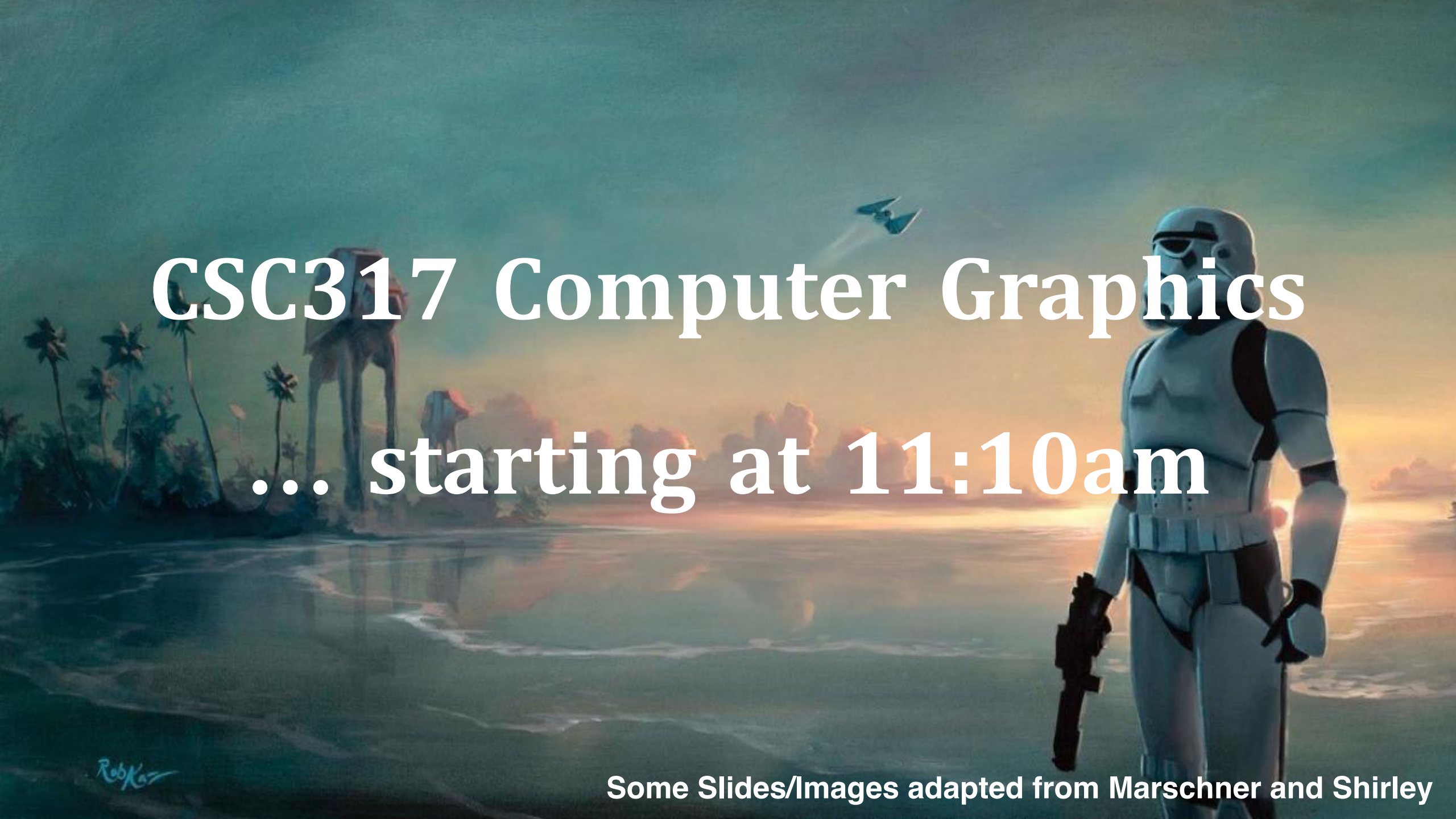
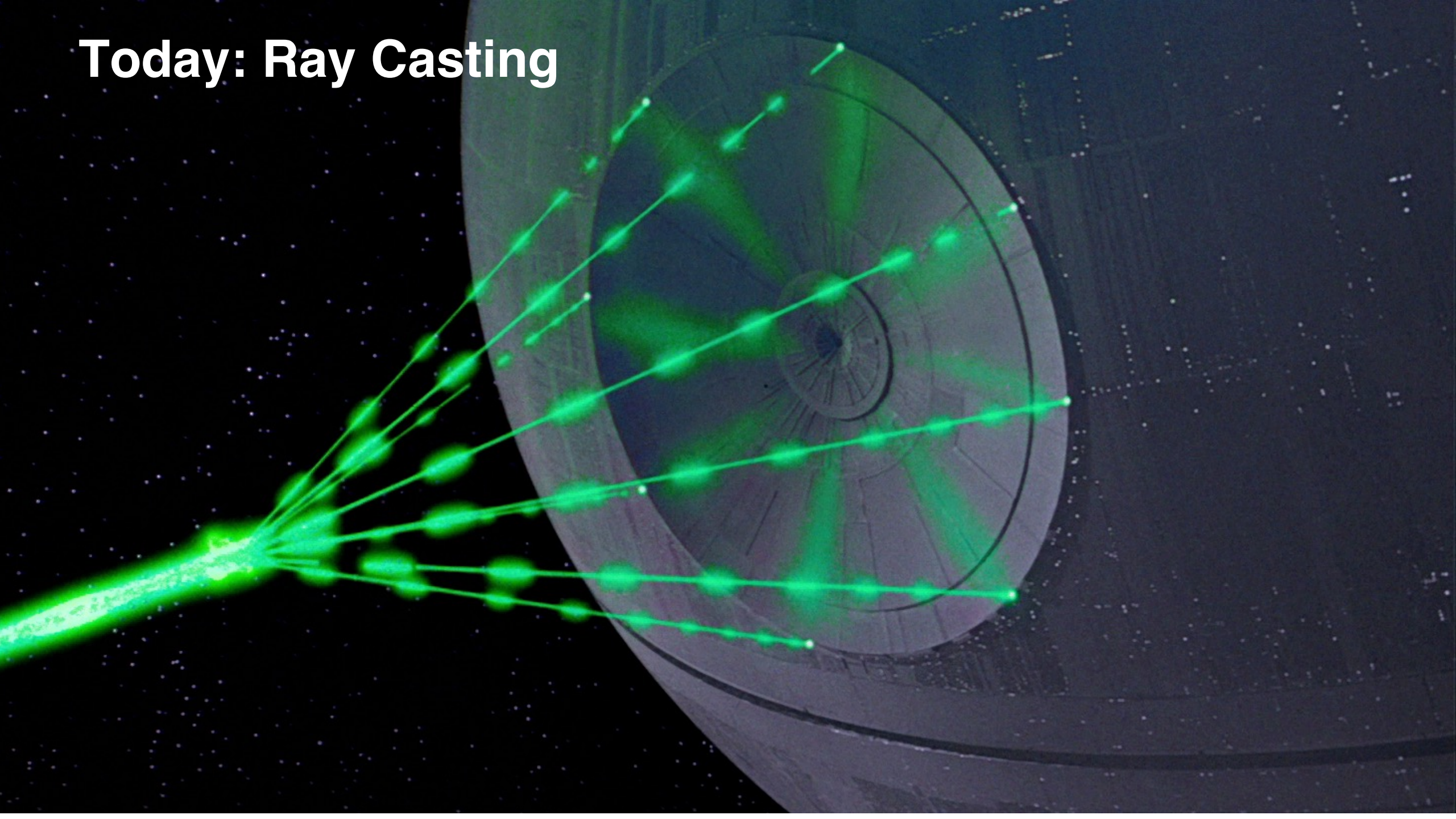

CSC317 Computer Graphics
… starting at 11:10am
Some Slides/Images adapted from Marschner and Shirley

Today: Ray Casting



Announcements

Assignment 1 is due this Wednesday

Assignment 2 available today!! (due 27/09)

Any Questions ?

Today: Ray Casting

The Ray Casting Algorithm

Introduction to Rays

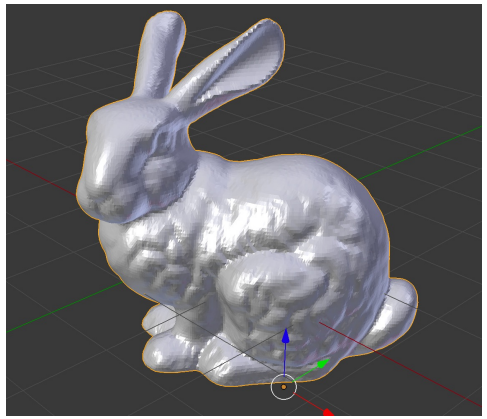
The Camera

Ray-Object Intersection

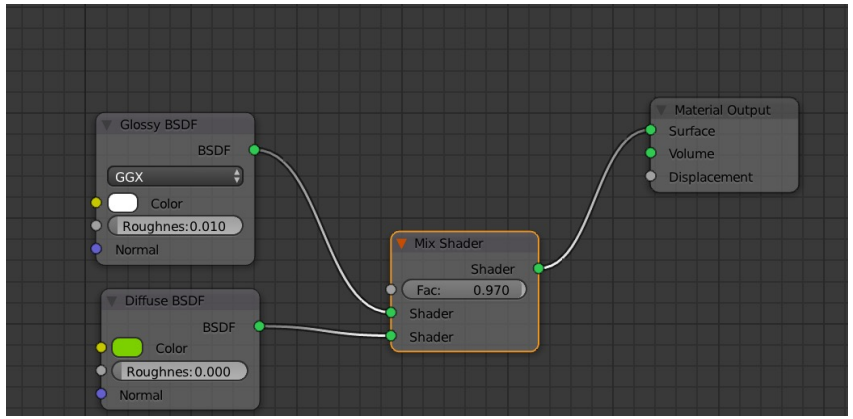
- Ray-Plane Intersection
- Ray-Sphere Intersection
- Ray-Triangle Intersection

Rendering

Input:

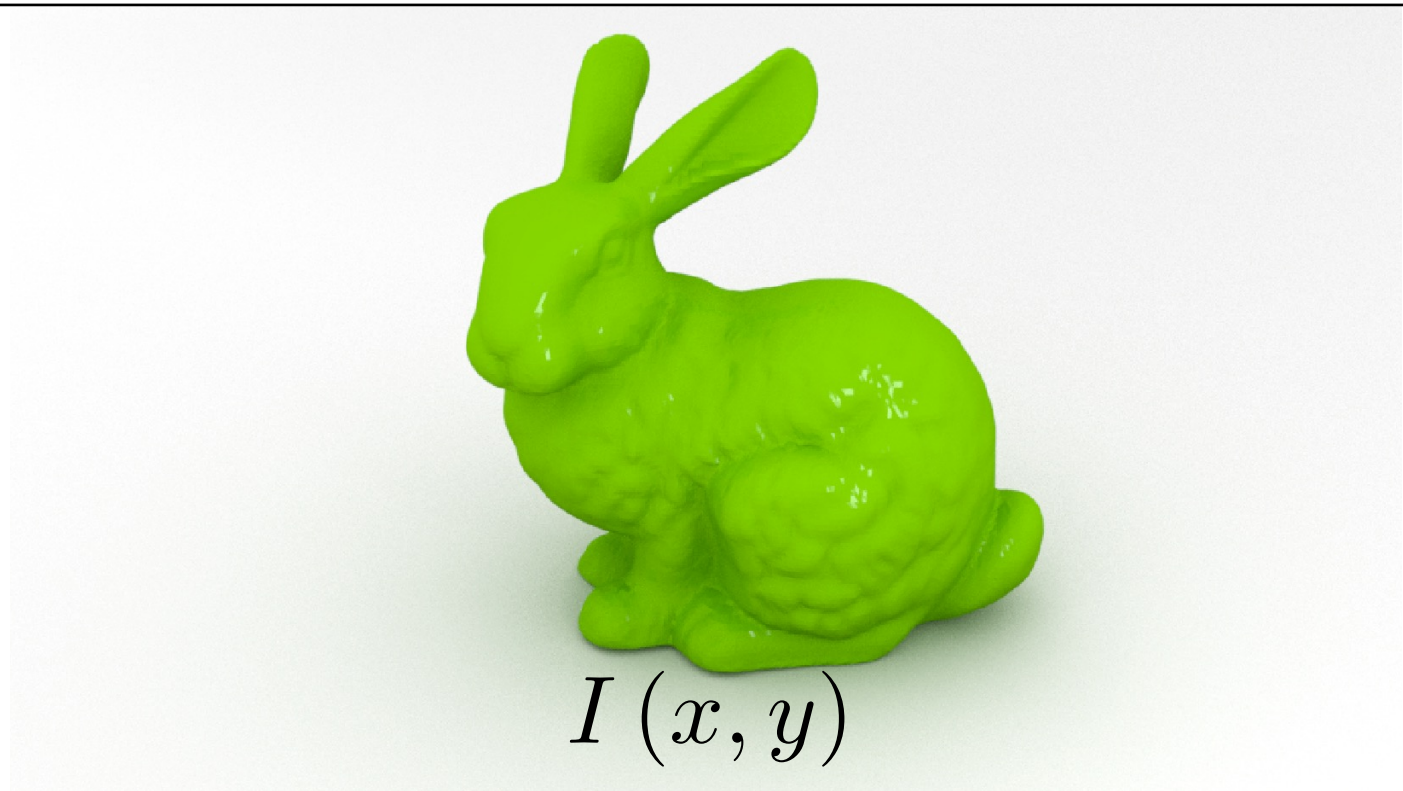


Objects



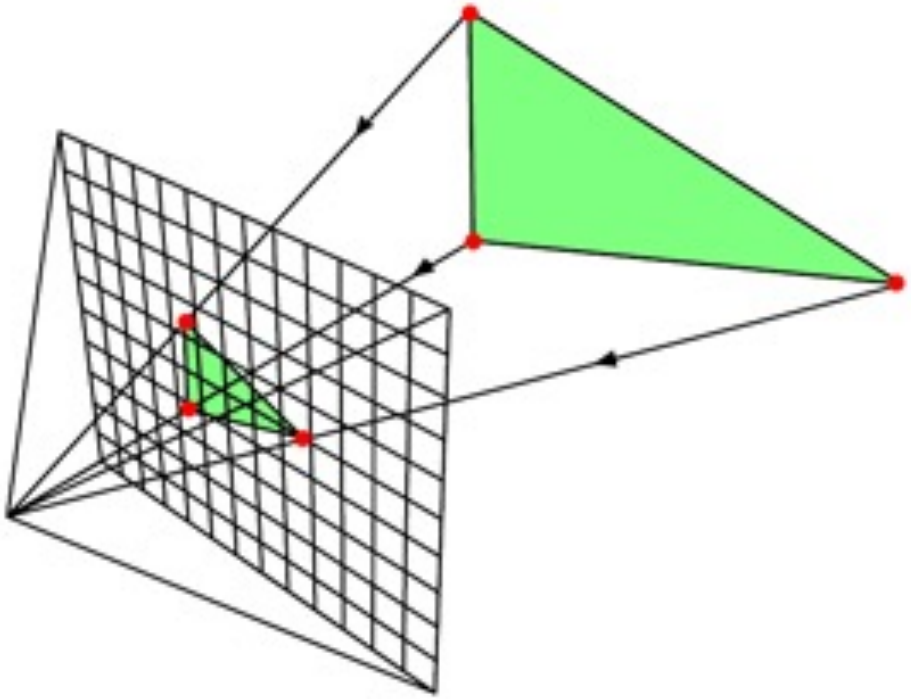
Materials

Output:

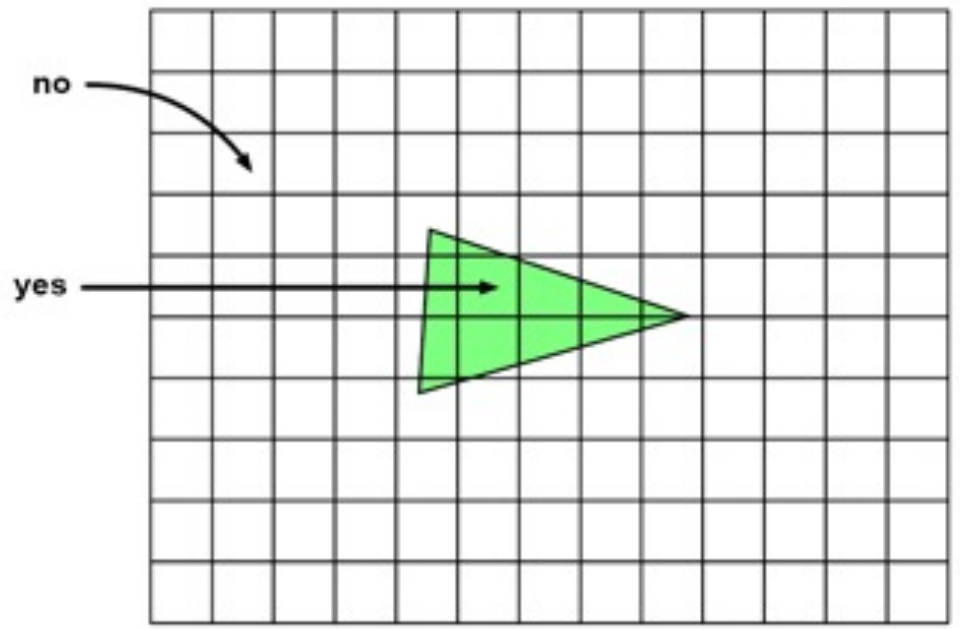



WWW.GRY-ONLINE.PL
zvarownik
13878049a9f142c2b036e491198e2149
WARIOR_GAMING_57
YZx-_Vulka
danielek185
zvarownik
danielek185
285 300 NW 330 345 N 15 30 NE 60 75 E
WARIOR_GAMING_57
0:52
94
0
0 5
0 0 0
0 100
100 100
Przytrzymaj
Fortnite I Epic Games

Rasterization



1. Project Vertices to Image Plane



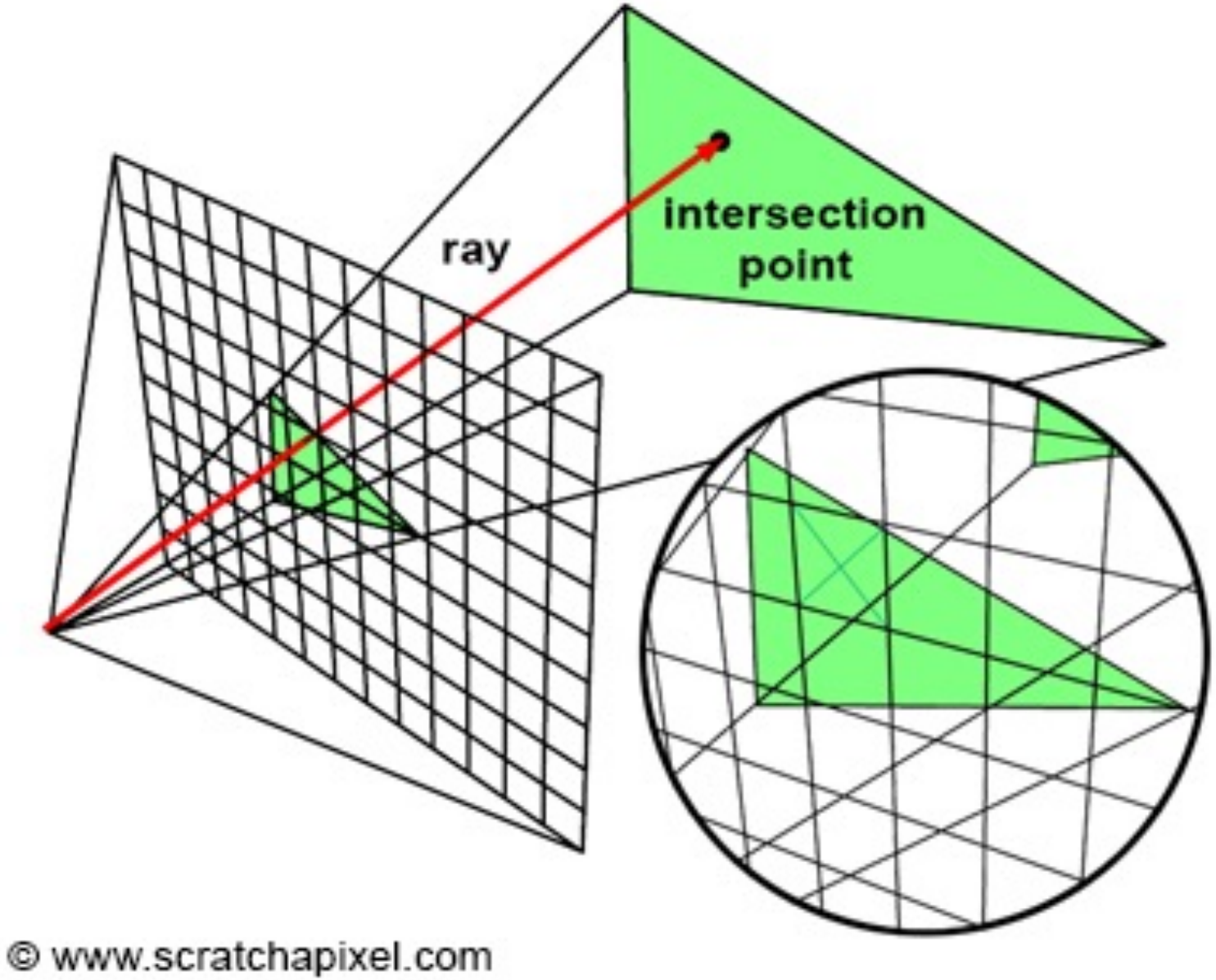
2. Turn on pixels inside triangle

https://www.scratchapixel.com/lessons/3d-basic-rendering/rasterization-practical-implementation

Rasterization

```
for each object in the scene {
    for each pixel in the image {
        if (object affects pixel) {
            do something
        }
    }
}
```

Ray Casting



https://www.scratchapixel.com/lessons/3d-basic-rendering/rasterization-practical-implementation

Ray Casting

```
for each pixel in the image {
    Generate a ray
    for each object in the scene {
        if (Intersect ray with object) {
            Set pixel colour
        }
    }
}
```

Basic Components of Ray Casting

Ray

Camera

Intersection Tests

Basic Components of Ray Casting

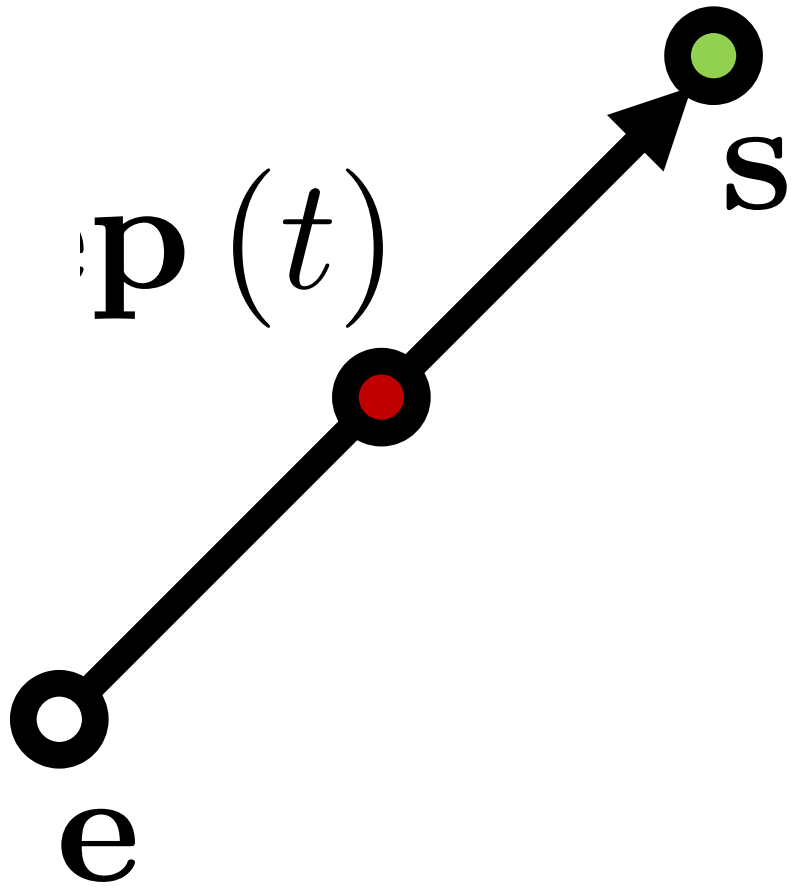
Ray

Camera

Intersection Tests

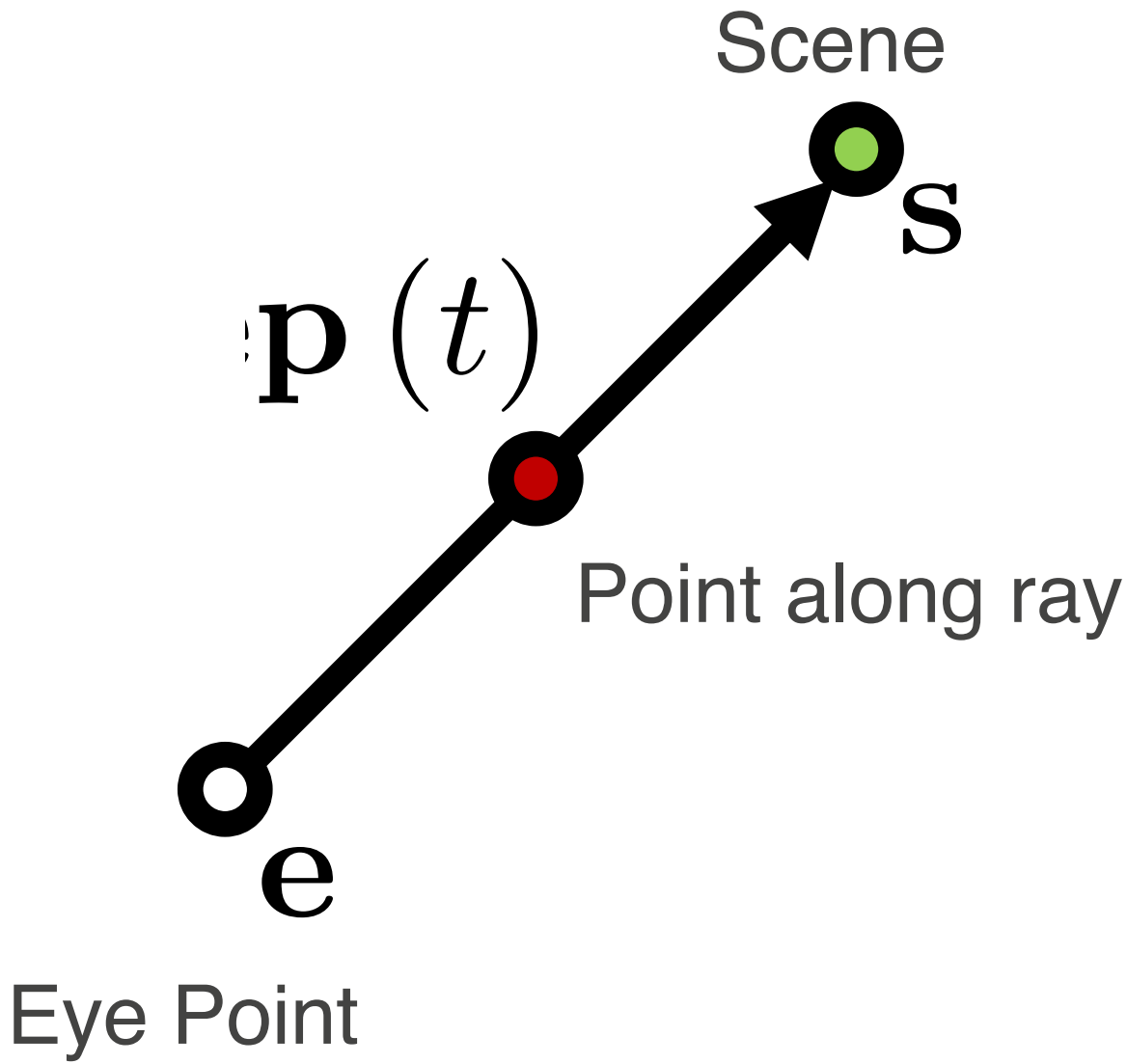
The Ray

$$\mathbf{p}(t) = \mathbf{e} + t(\mathbf{s} - \mathbf{e})$$

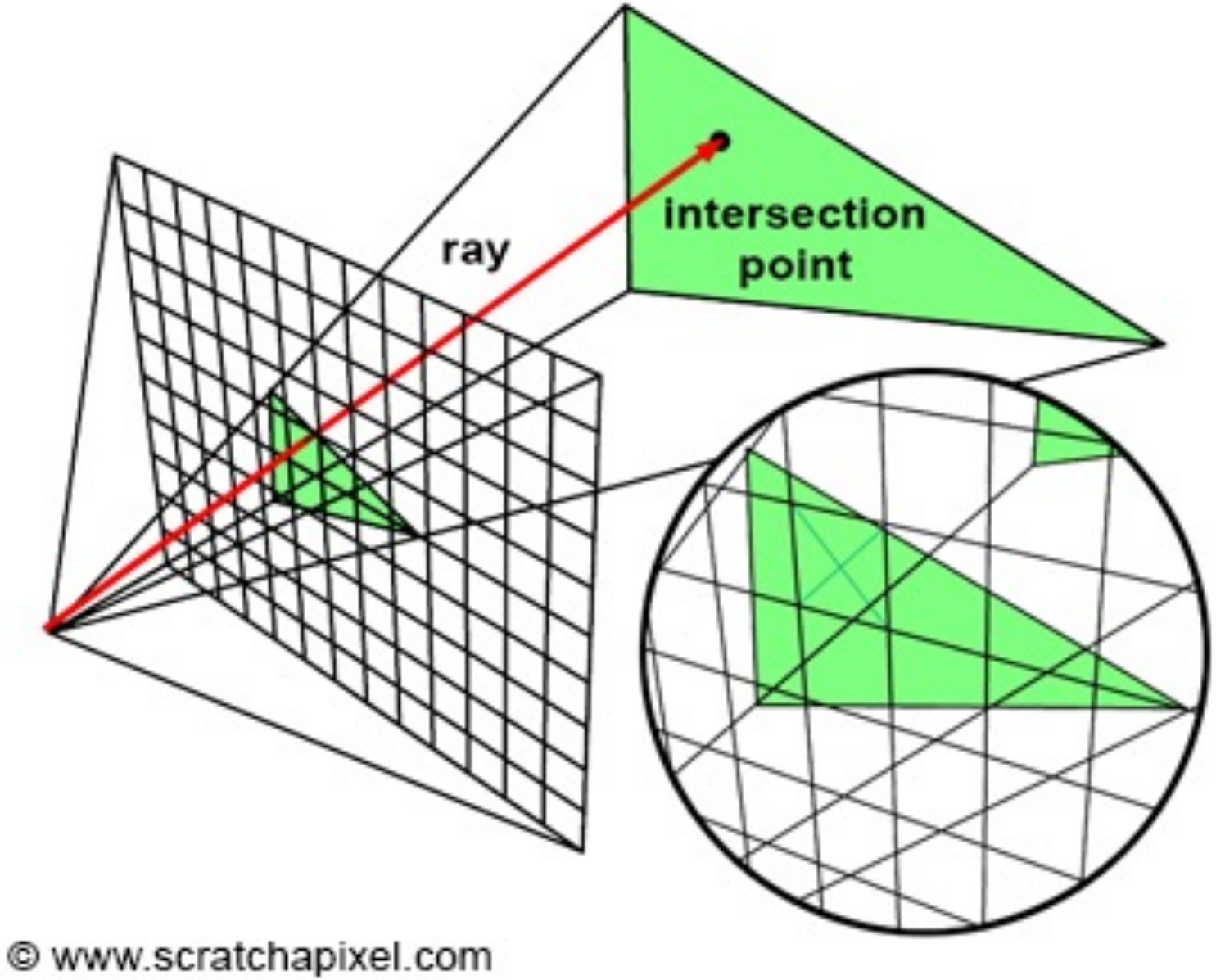


The Ray

$$\mathbf{p}(t) = \mathbf{e} + t(\mathbf{s} - \mathbf{e})$$



Ray Casting



https://www.scratchapixel.com/lessons/3d-basic-rendering/rasterization-practical-implementation

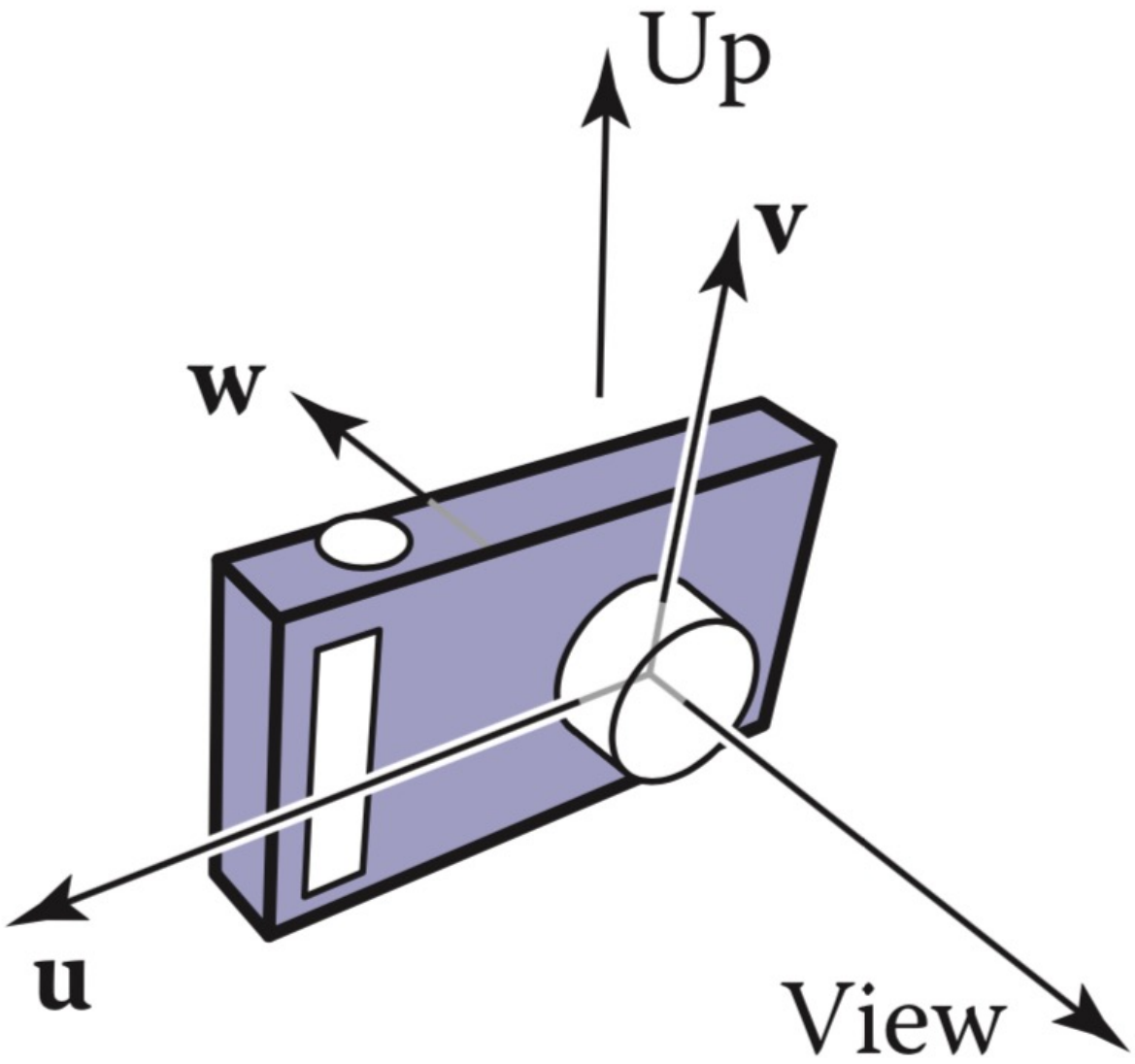
Basic Components of Ray Casting

Ray

Camera

Intersection Tests

The Camera

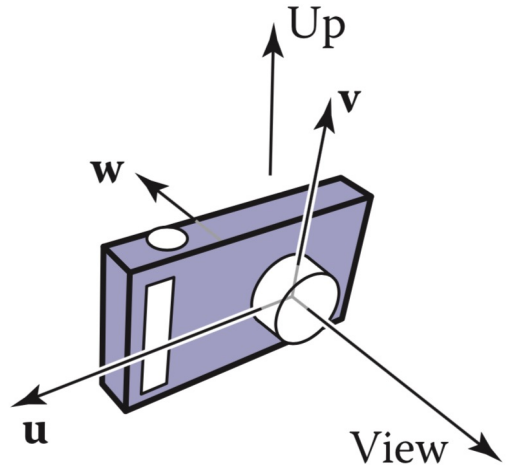


The Camera

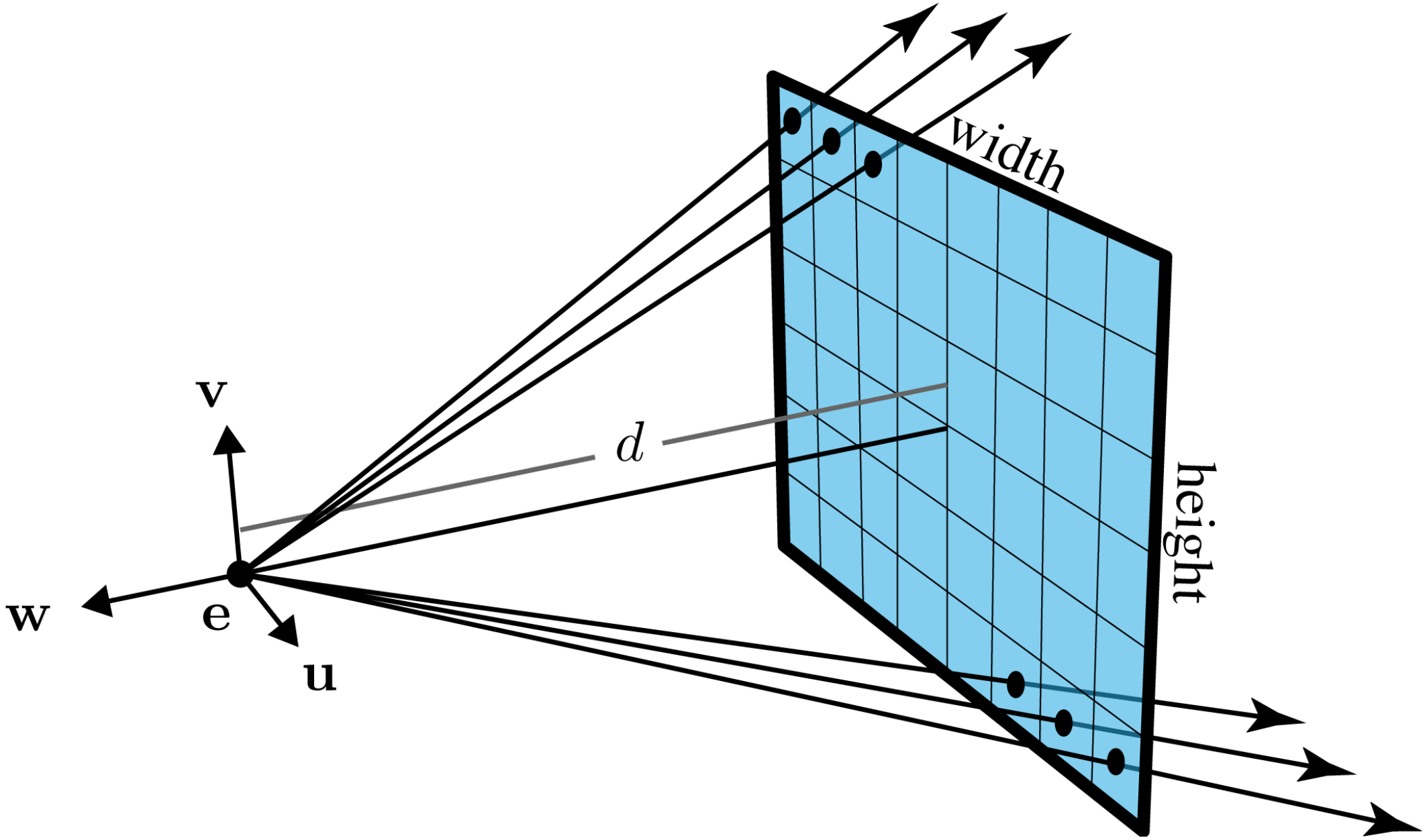
$$\mathbf{w} = -\frac{\text{View}}{\|\text{View}\|}$$

$$\mathbf{u} = \text{View} \times \text{Up}$$

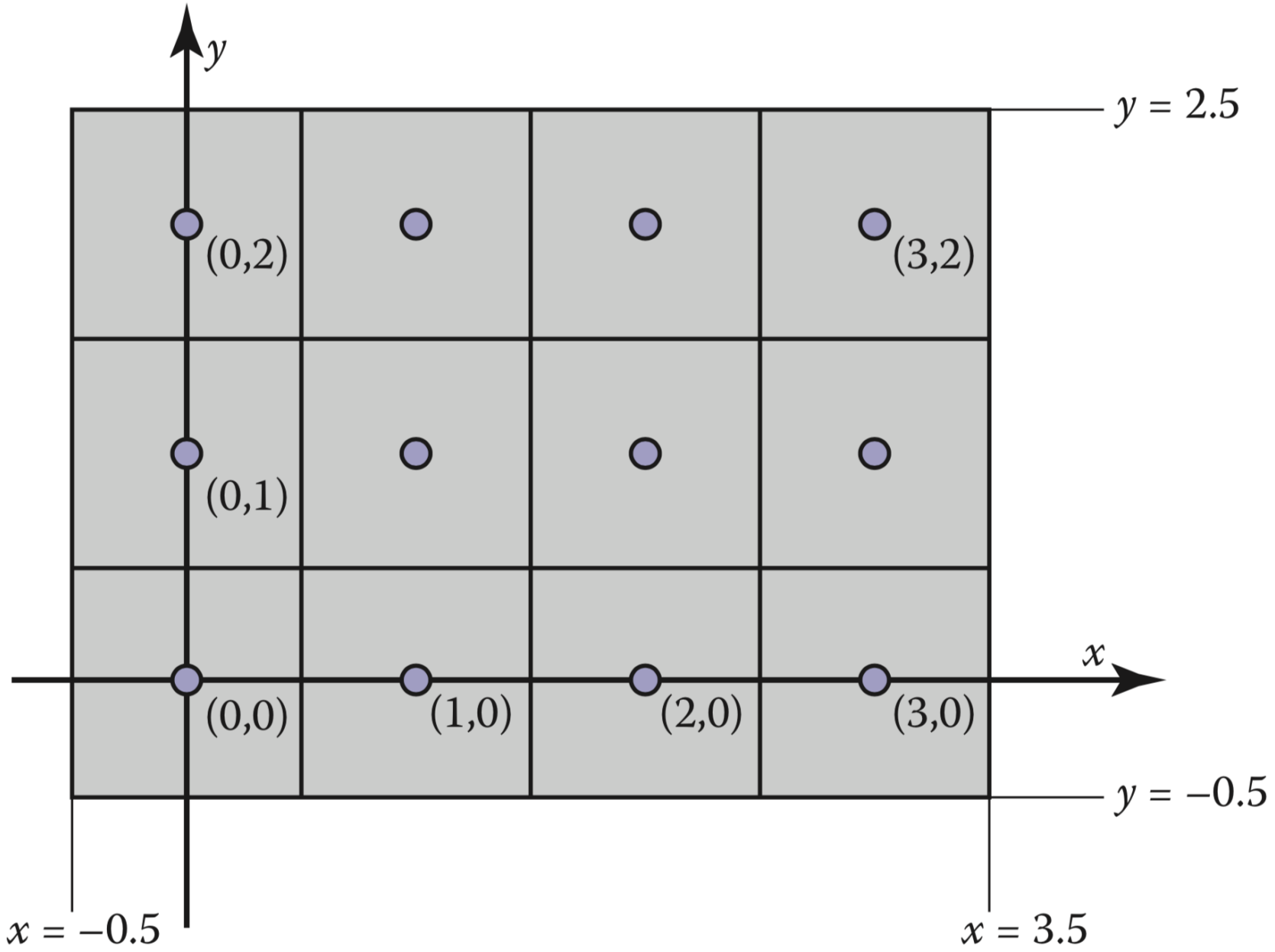
$$\mathbf{v} = \mathbf{w} \times \mathbf{u}$$

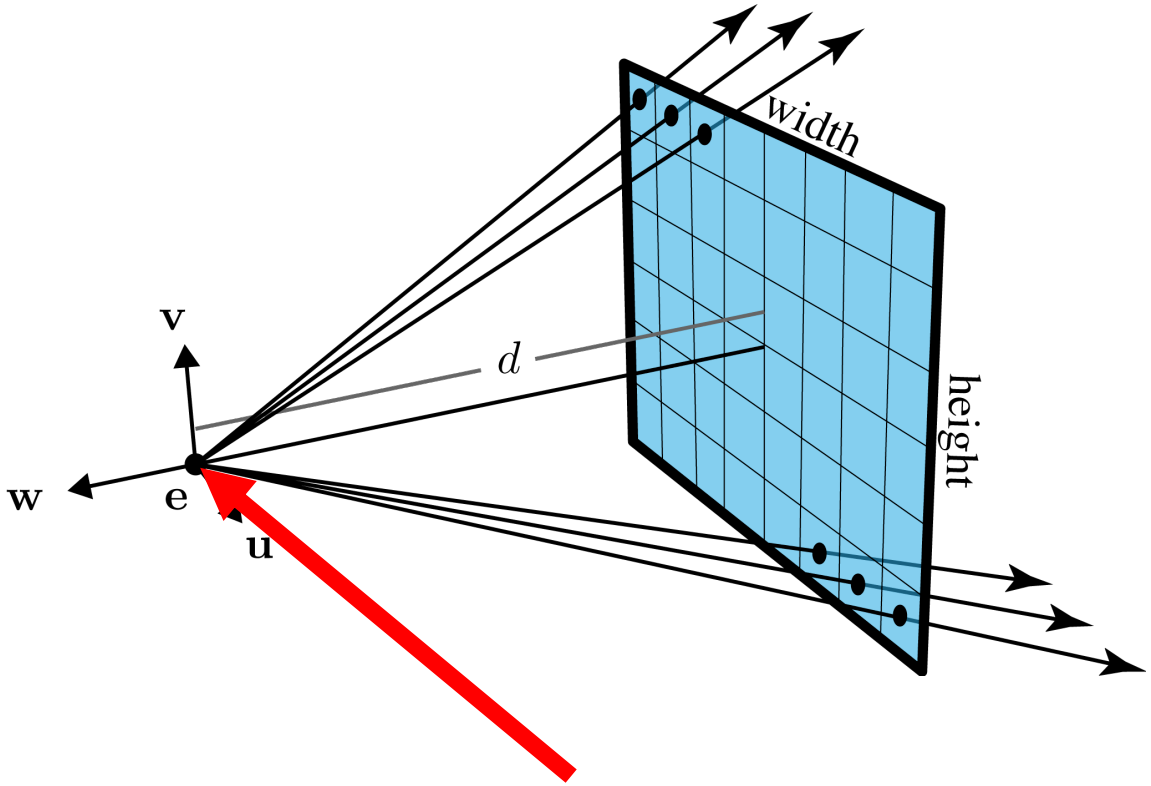


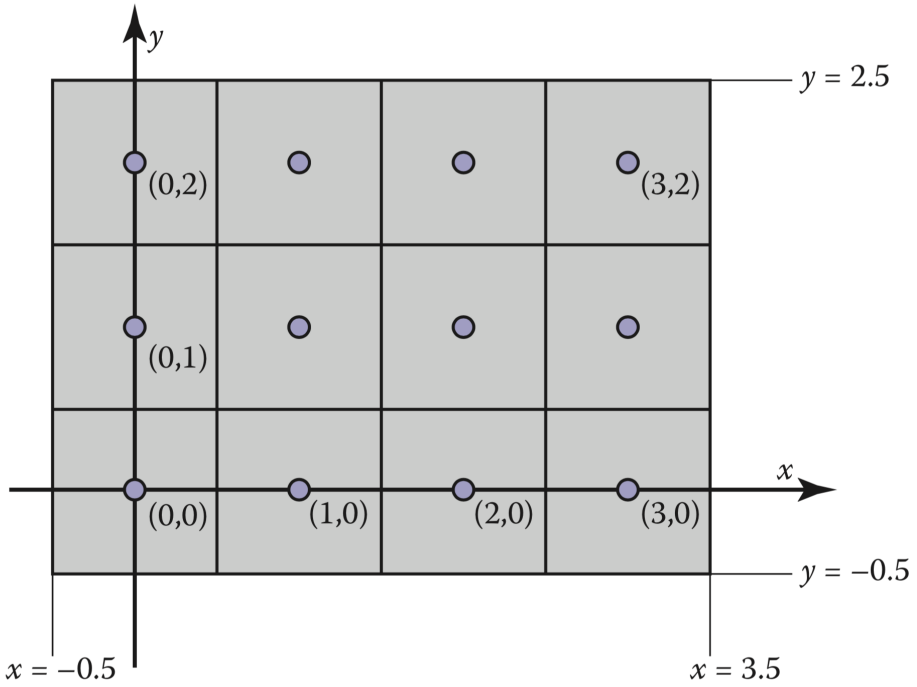
Generating Rays



Recall: Standard Pixel Coordinate System

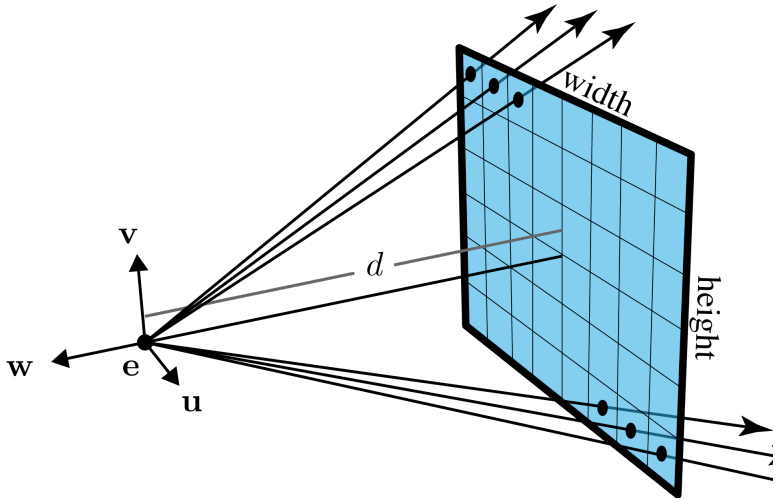


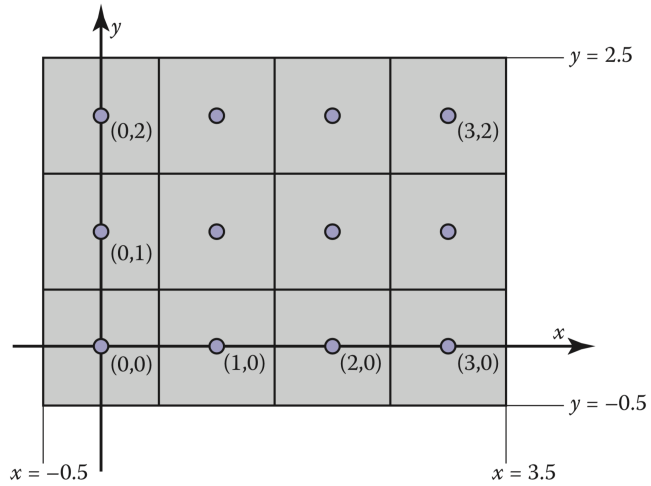




Origin of camera frame (the eye)

What are the coordinates for pixel (i, j) in the camera frame?



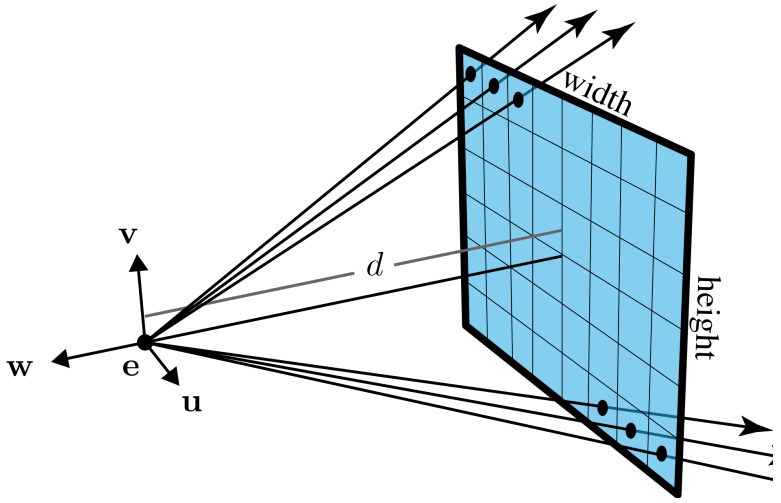


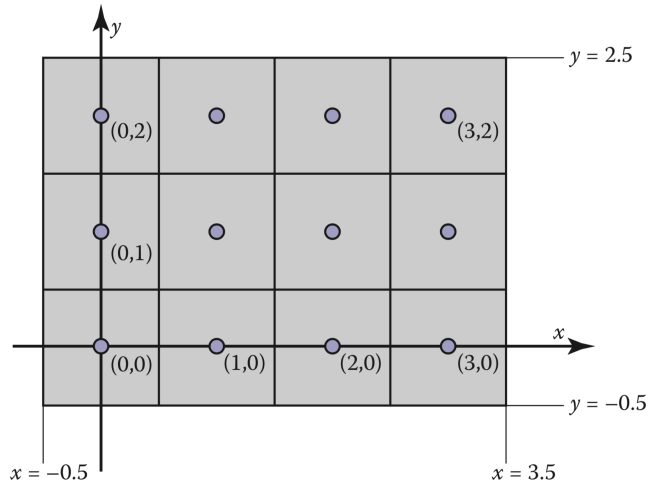
Bottom Left Corner (i, j): ?

Top Right Corner (i, j): ?

Bottom Left Corner (u, v): ?

Top Right Corner (u, v): ?



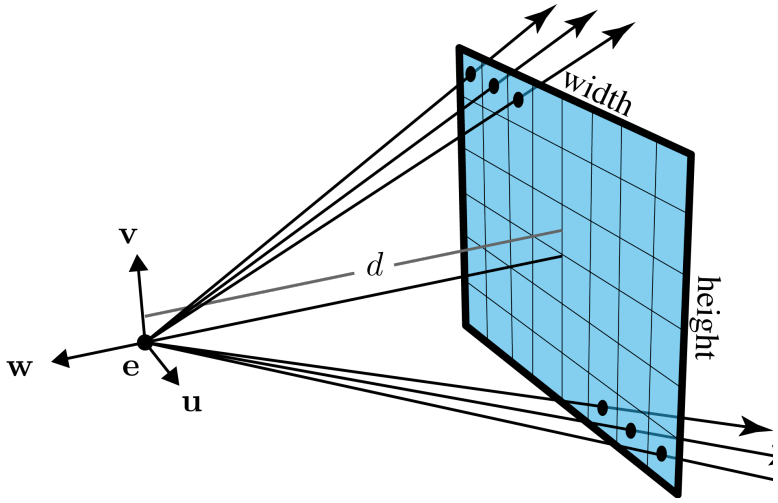


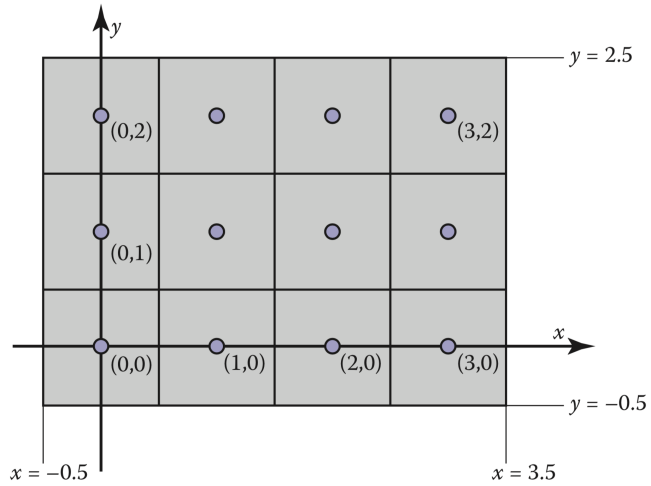
Bottom Left Corner (i, j): $(-\frac{1}{2}, -\frac{1}{2})$

Top Right Corner (i, j): $(n_x - \frac{1}{2}, n_y - \frac{1}{2})$

Bottom Left Corner (u, v): $\left(-\frac{\text{width}}{2}, -\frac{\text{height}}{2}\right)$

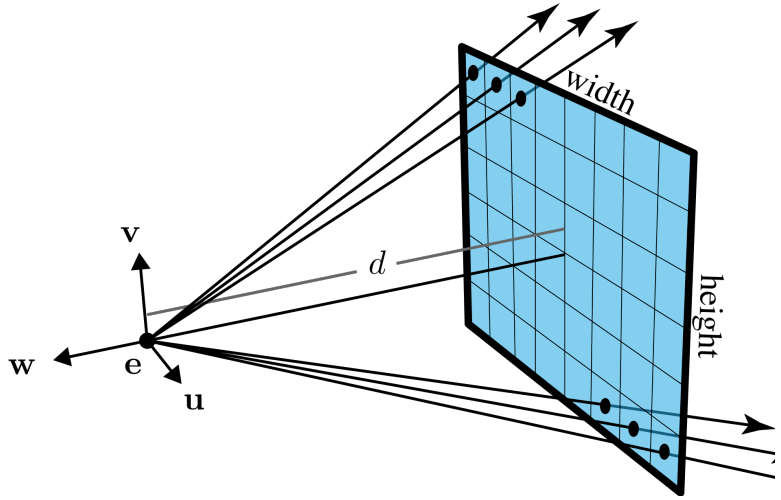
Top Right Corner (u, v): $\left(\frac{\text{width}}{2}, \frac{\text{height}}{2}\right)$

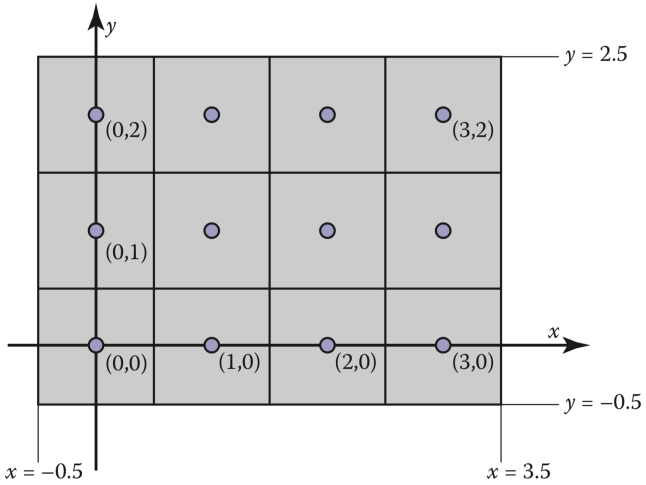




$$u = a \cdot i + b$$

$$v = c \cdot j + d$$





Physical Width of Image in 3D Space

$$u = \text{width} \cdot \frac{\left(i + \frac{1}{2}\right)}{n_x} - \frac{\text{width}}{2}$$

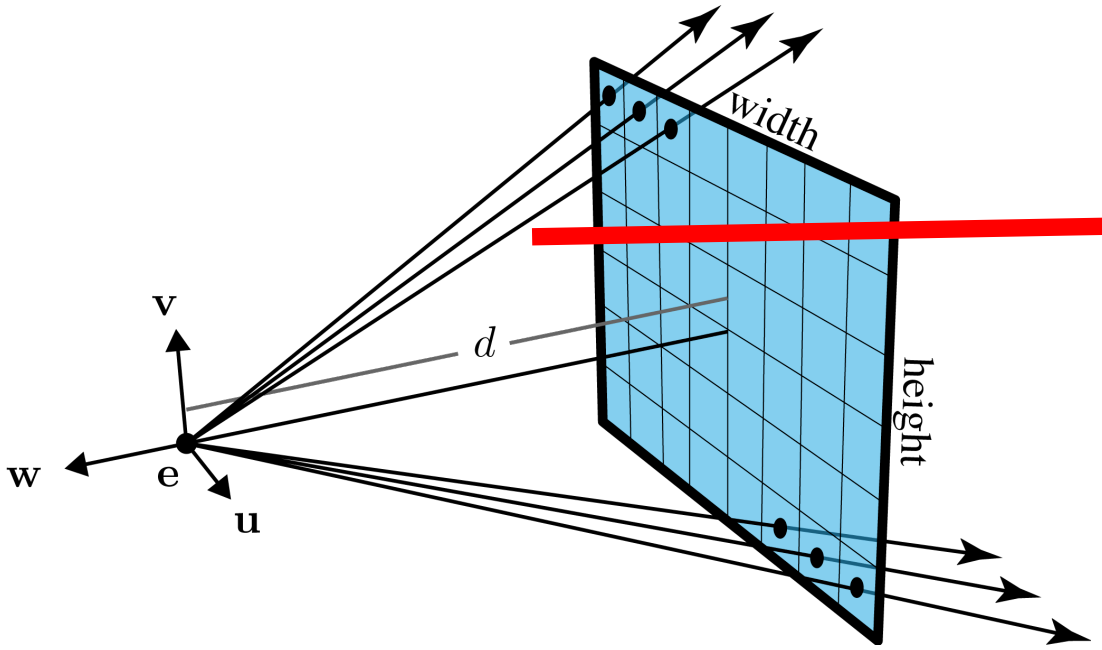
n_x # Pixels X

Physical Height of Image in 3D Space

$$v = \text{height} \cdot \frac{\left(j + \frac{1}{2}\right)}{n_y} - \frac{\text{height}}{2}$$

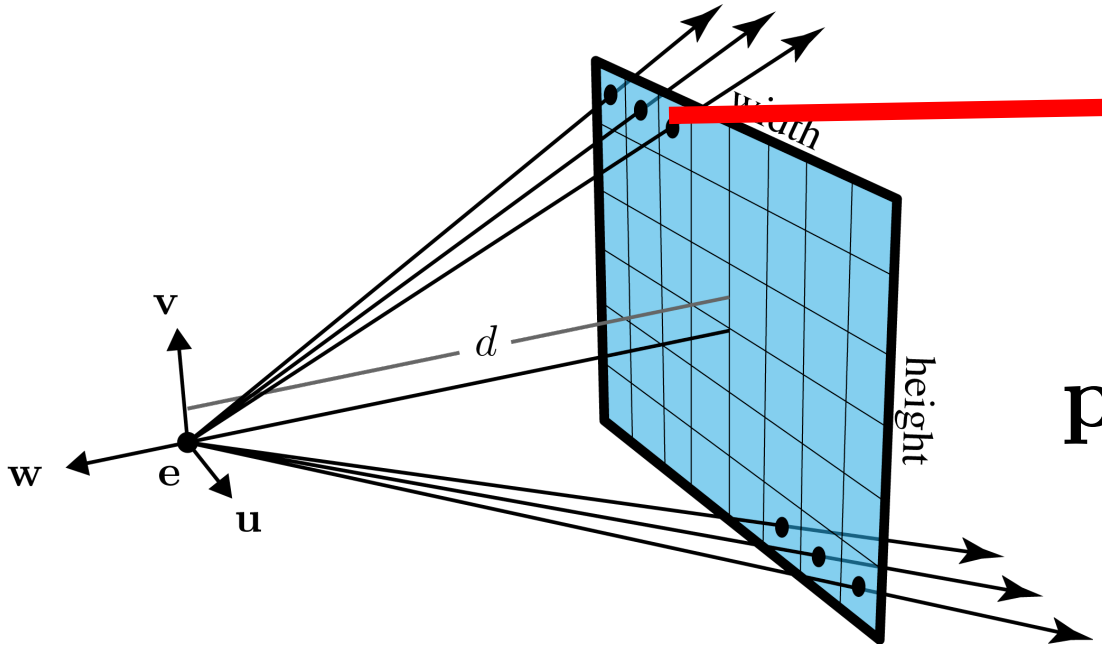
n_y # Pixels Y

Ray Equation in Camera Space



$$\mathbf{p}(t) = \mathbf{e} + t(\mathbf{s} - \mathbf{e})$$

Ray Equation in Camera Space



$$\mathbf{p}(t) = \mathbf{e} + t(\mathbf{s} - \mathbf{e})$$

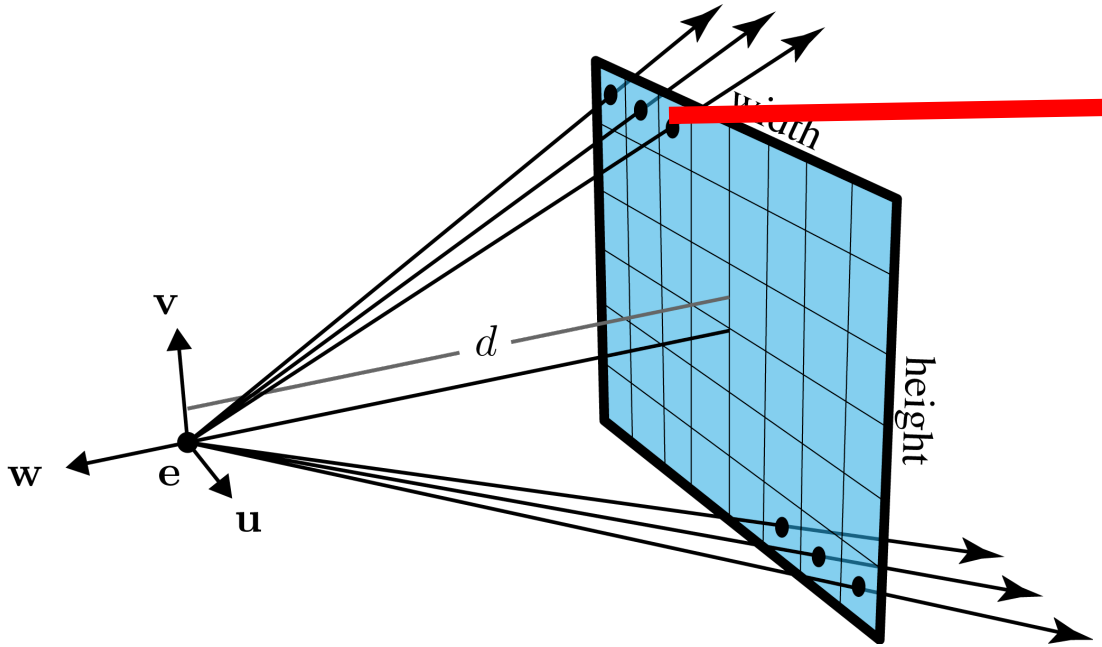
$$\mathbf{p}\,(t) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + t \left(\begin{bmatrix} u\,(i) \\ v\,(j) \\ -d \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right)$$

$$\mathbf{p}\,(t) = t \begin{bmatrix} u\,(i) \\ v\,(j) \\ -d \end{bmatrix}$$

$$u = \text{width} \cdot \frac{\left(i + \frac{1}{2}\right)}{n_x} - \frac{\text{width}}{2}$$

$$v = \text{height} \cdot \frac{\left(j + \frac{1}{2}\right)}{n_y} - \frac{\text{height}}{2}$$

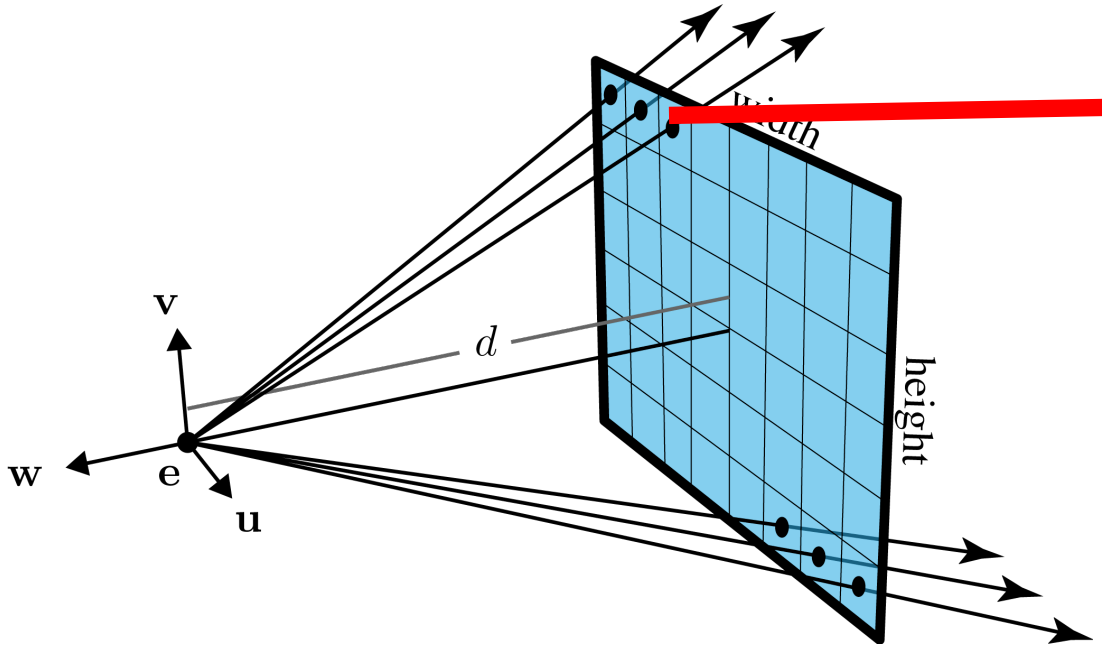
Ray Equation in World Space



$$\mathbf{p}(t) = \mathbf{e} + t(\mathbf{s} - \mathbf{e})$$

$$\mathbf{p}\,(t) = t\,(u\,(i)\,\mathbf{u} + v\,(j)\,\mathbf{v} + -d\mathbf{w}) + \mathbf{e}$$

Ray Equation in World Space



$$\mathbf{p}(t) = \mathbf{e} + t(\mathbf{s} - \mathbf{e})$$

$$\mathbf{p}(t) = t \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} \end{bmatrix} \begin{bmatrix} u(i) \\ v(j) \\ -d \end{bmatrix} + \mathbf{e}$$

Camera Transformation Matrix

Ray Casting

```
for each pixel in the image {
    Generate a ray
    for each object in the scene {
        if (Intersect ray with object) {
            Set pixel colour
        }
    }
}
```

Basic Components of Ray Casting

Ray

Camera

Intersection Tests

Intersection Tests

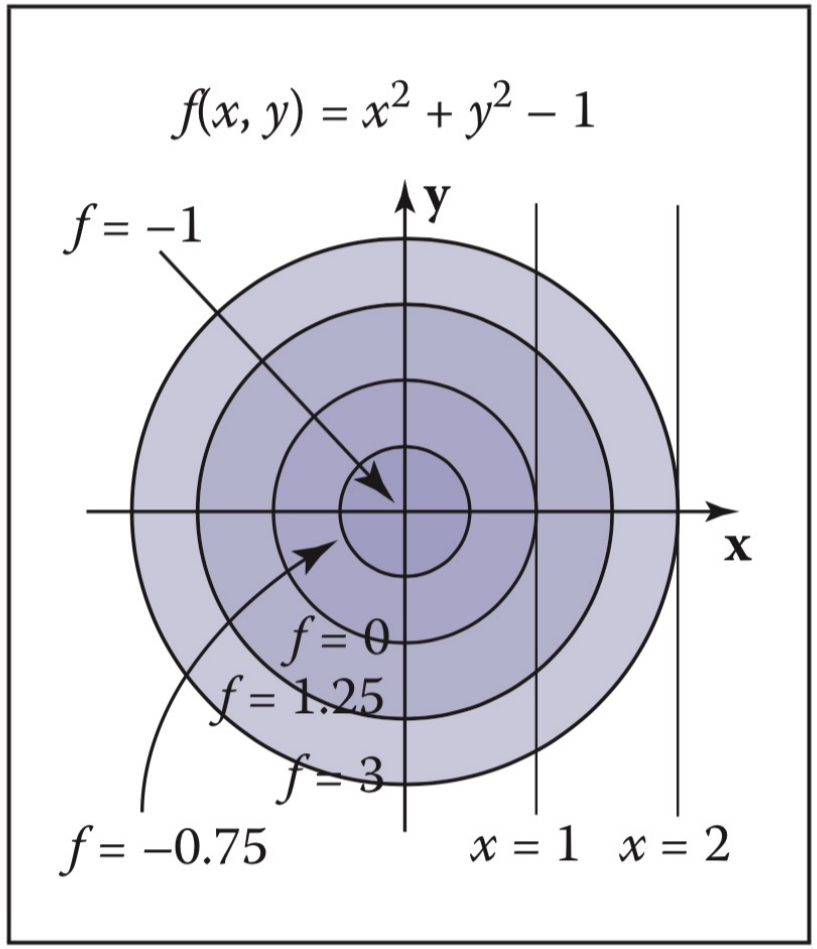
Plane

Sphere

Triangle

Aside: Types of Surface

Implicit Surface

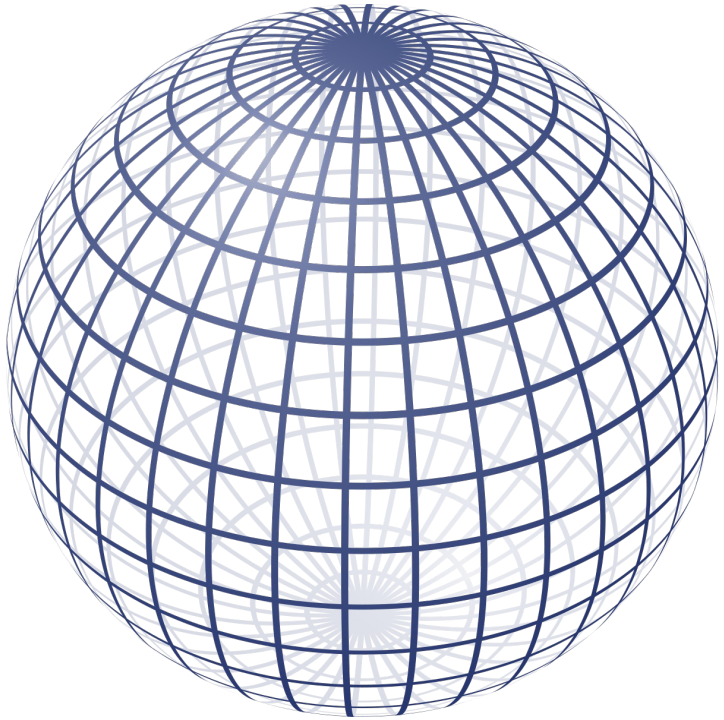


Parametric Surface

$$x = r \cos\phi \sin\theta,$$

$$y = r \sin\phi \sin\theta,$$

$$z = r \cos\theta.$$



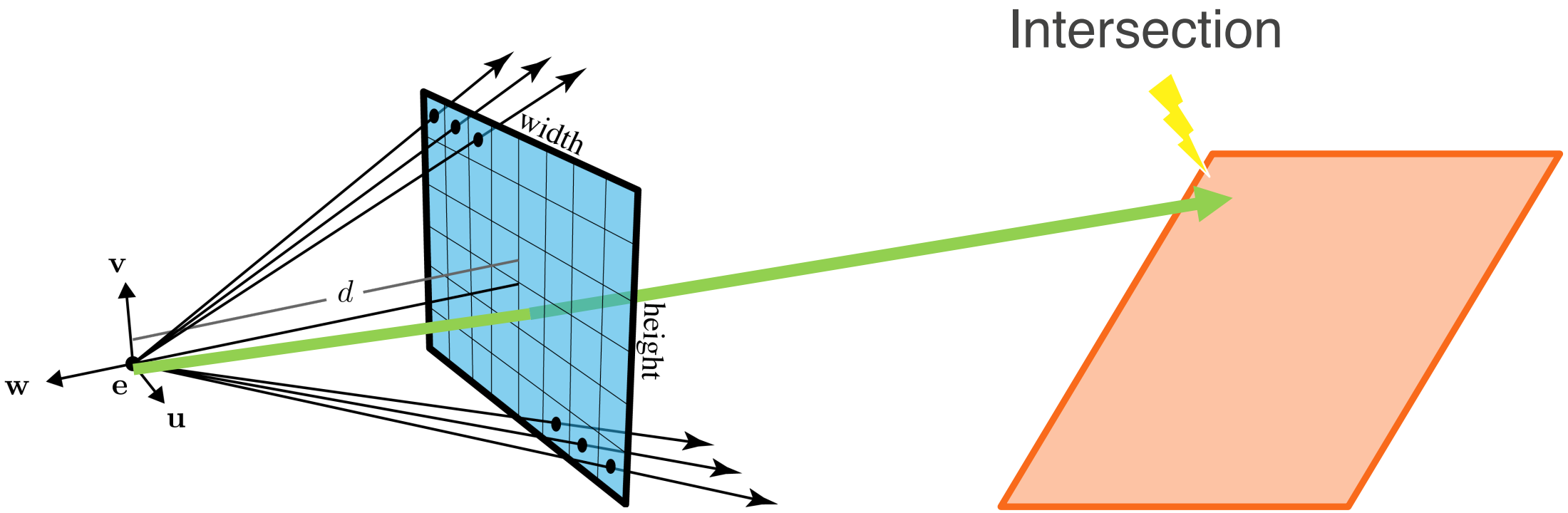
Intersection Tests

Plane

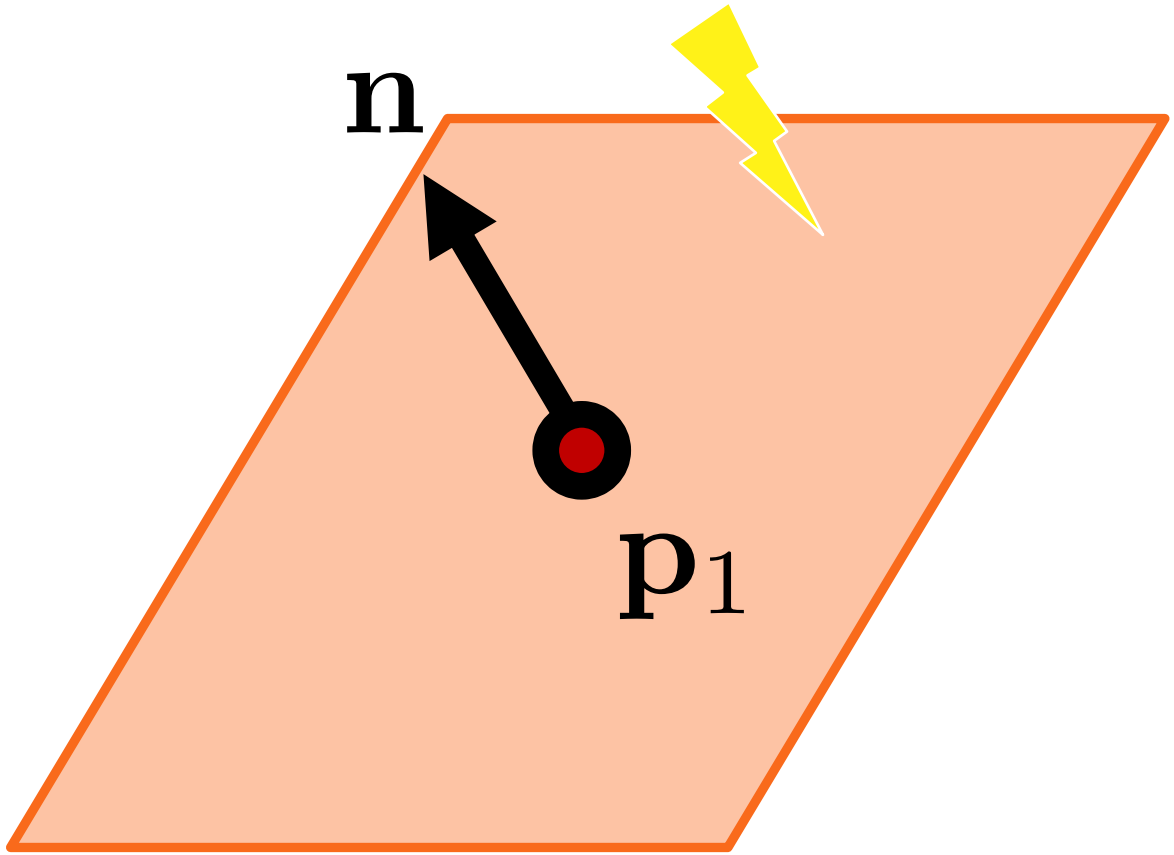
Sphere

Triangle

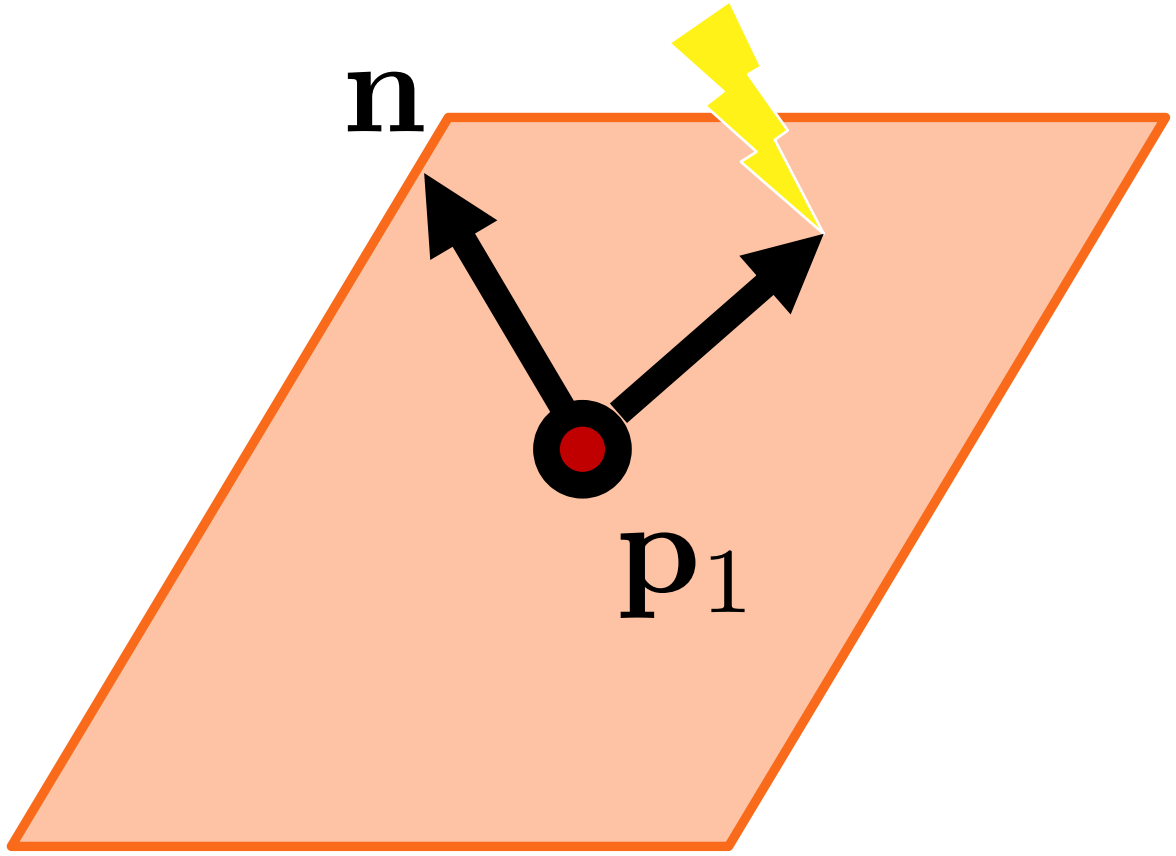
Ray-Plane Intersection



Plane Equation



Plane Equation

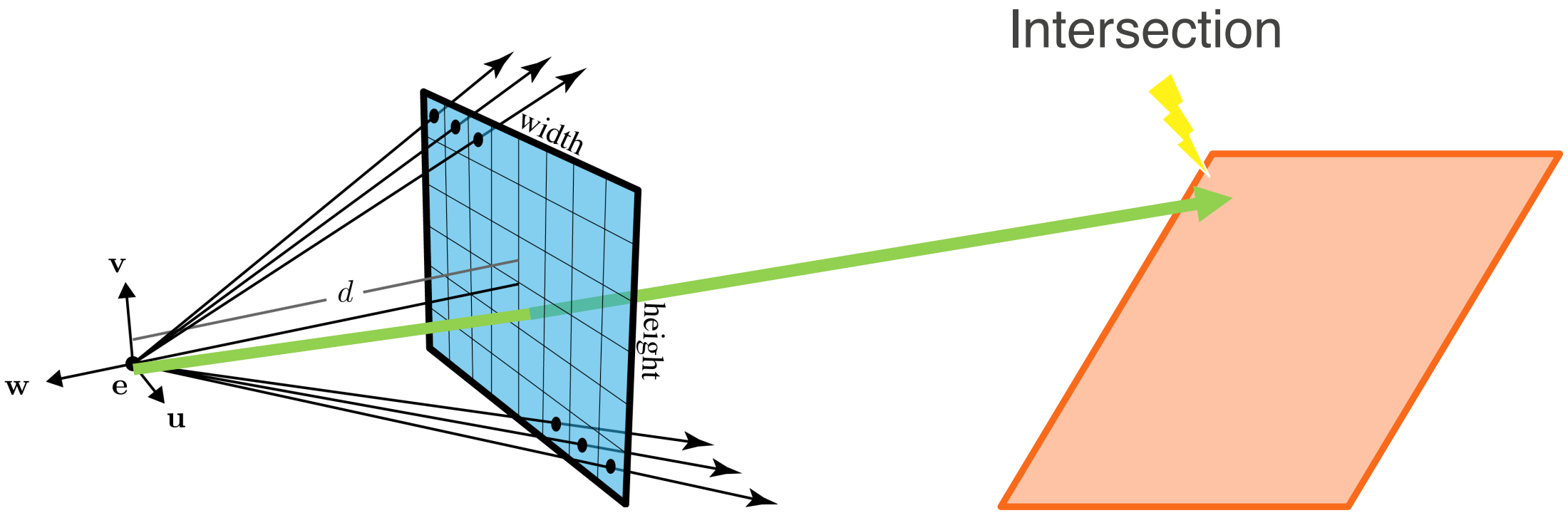


$$\mathbf{n}^T (\mathbf{p} - \mathbf{p}_1) = 0$$

$$\mathbf{n}^T \mathbf{p} - \mathbf{n}^T \mathbf{p}_1 = 0$$

$$\mathbf{n}^T \mathbf{p} - q = 0$$

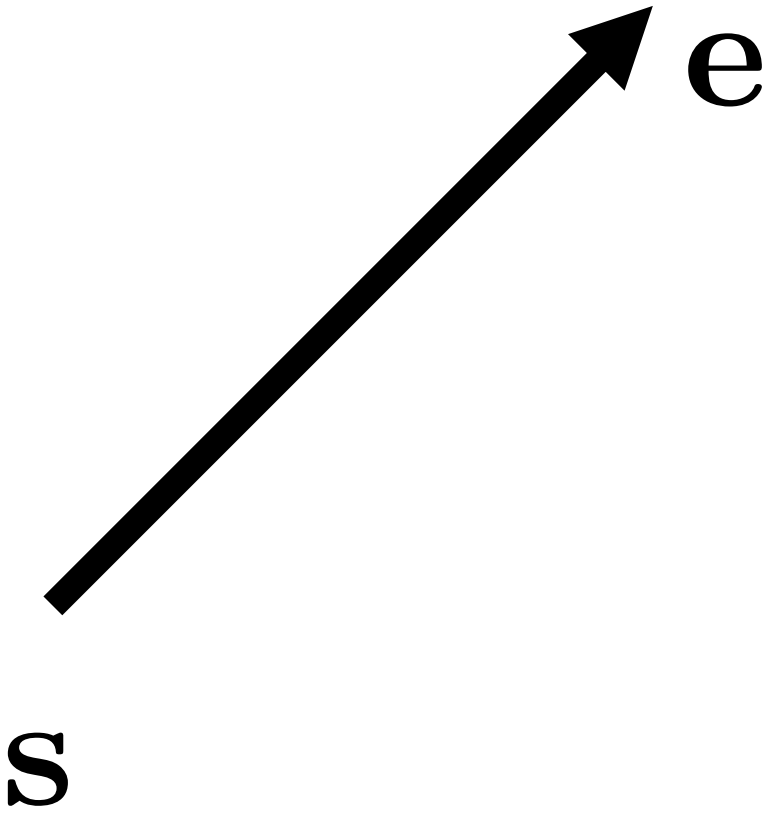
Ray-Plane Intersection



Ray-Plane Intersection

$$\mathbf{n}^T \mathbf{p}(t) - q = 0$$

$$\mathbf{p}(t) = \mathbf{e} + t(\mathbf{s} - \mathbf{e})$$

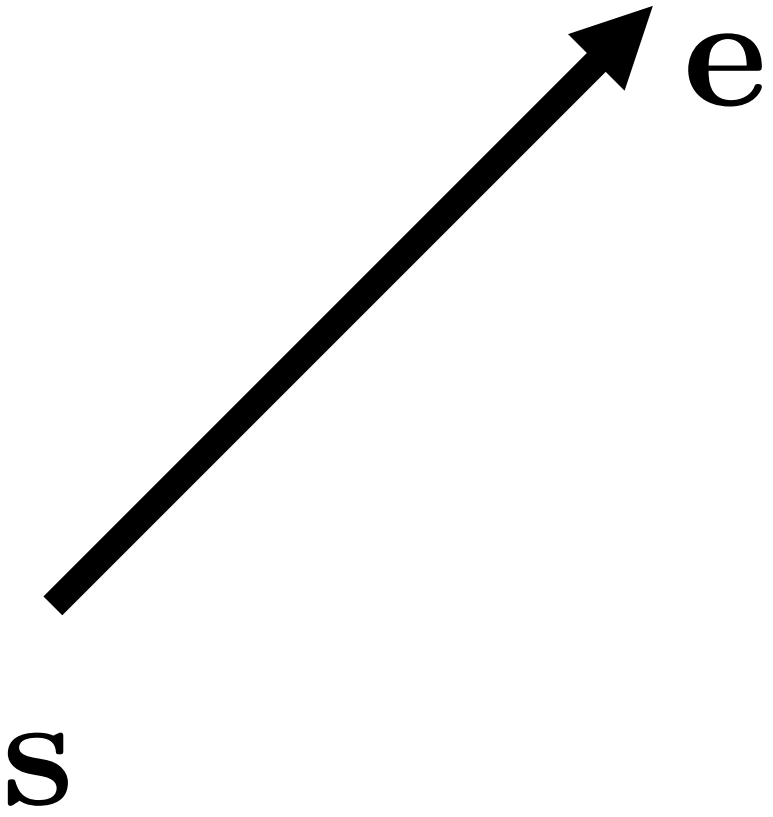


Ray-Plane Intersection

$$\mathbf{n}^T \mathbf{p}(t) - q = 0$$

$$t = \frac{q - \mathbf{n}^T \mathbf{e}}{\mathbf{n}^T (\mathbf{s} - \mathbf{e})}$$

$$\mathbf{p}(t) = \mathbf{e} + t(\mathbf{s} - \mathbf{e})$$



Intersection Tests

Plane

Sphere

Triangle

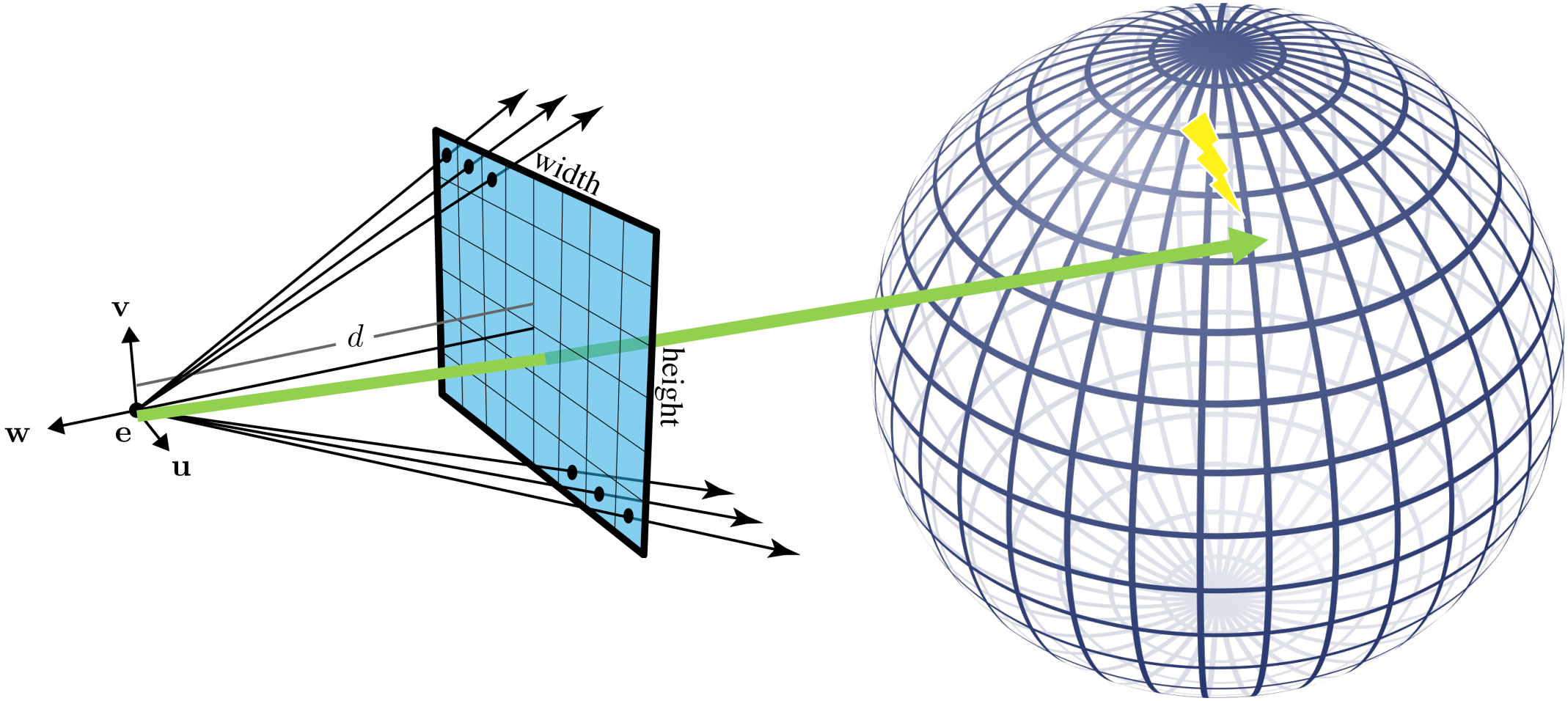
Intersection Tests

Plane

Sphere

Triangle

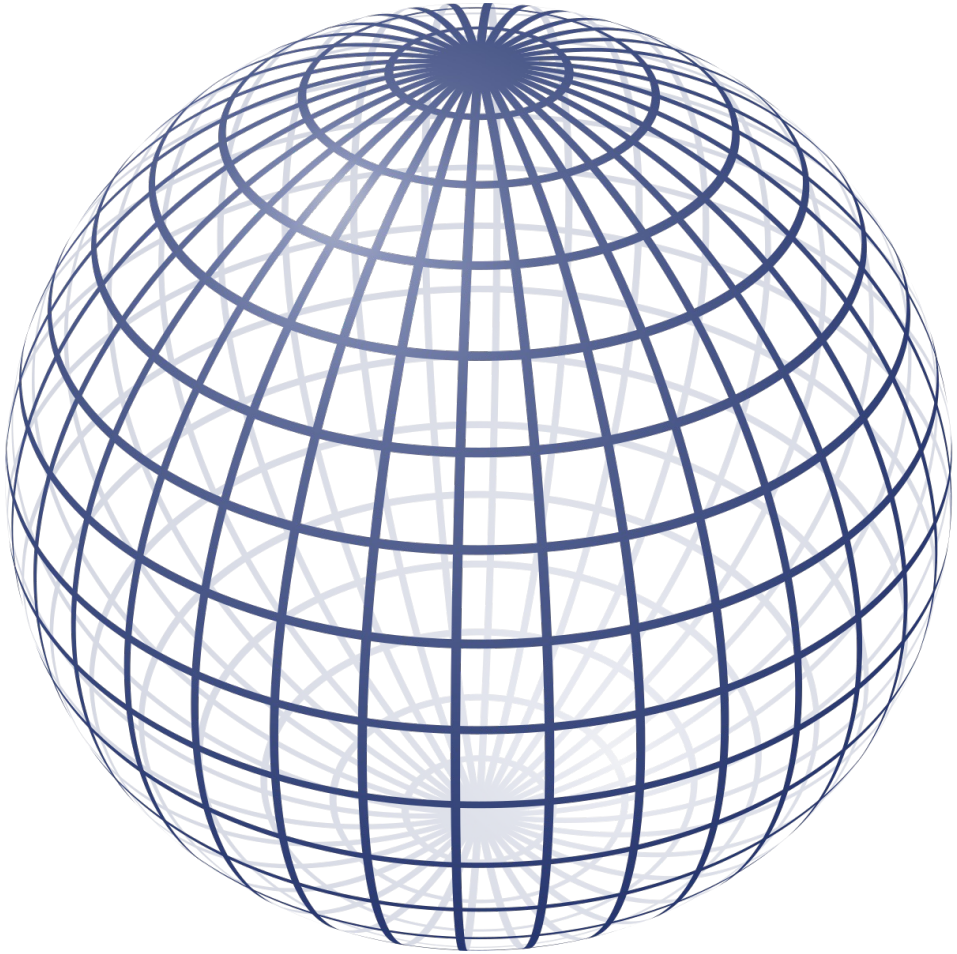
Ray-Sphere Intersection



Implicit Equation of a Sphere

$$\mathbf{x}^T\mathbf{x} - r^2 = 0$$

Sphere centered at origin with radius r



Ray-Sphere Intersection

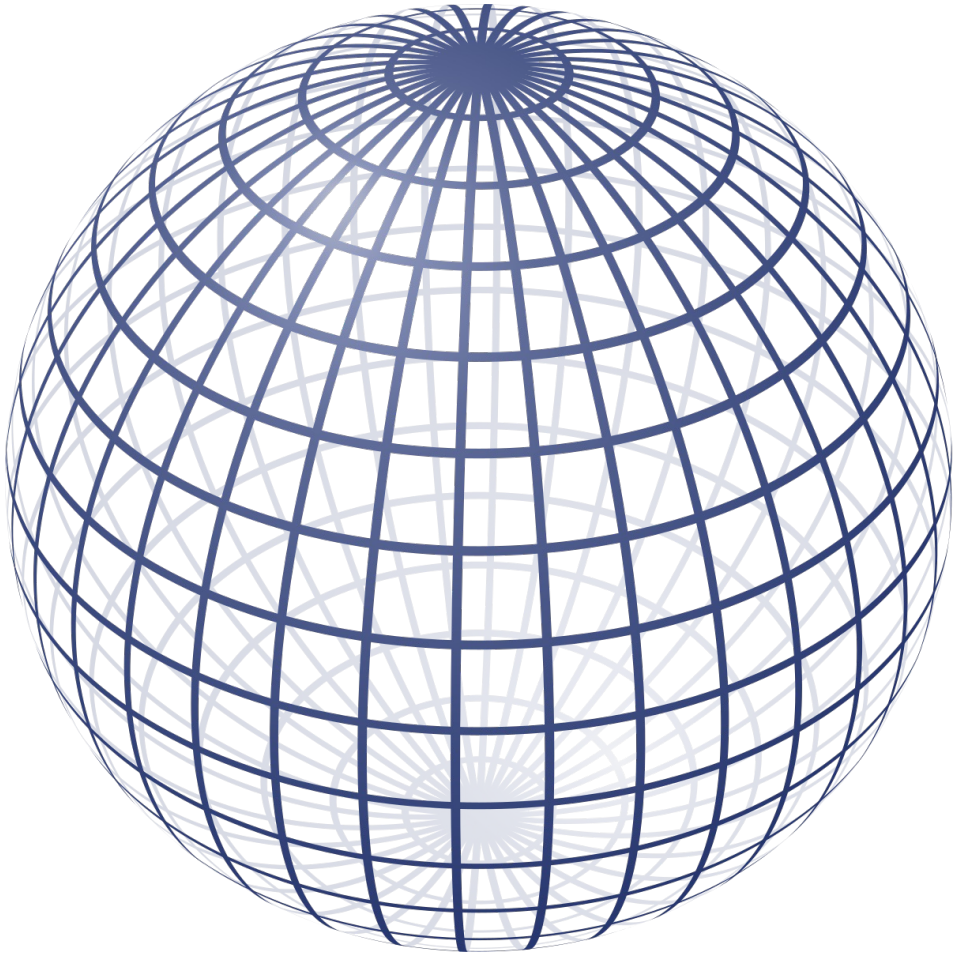
$$\mathbf{p}(t)^T \mathbf{p}(t) - r^2 = 0$$

$$a \cdot t^2 + b \cdot t + c = 0$$

$$a = (\mathbf{s} - \mathbf{e})^T (\mathbf{s} - \mathbf{e})$$

$$b = 2\mathbf{e}^T (\mathbf{s} - \mathbf{e})$$

$$c = \mathbf{e}^T \mathbf{e} - r^2$$



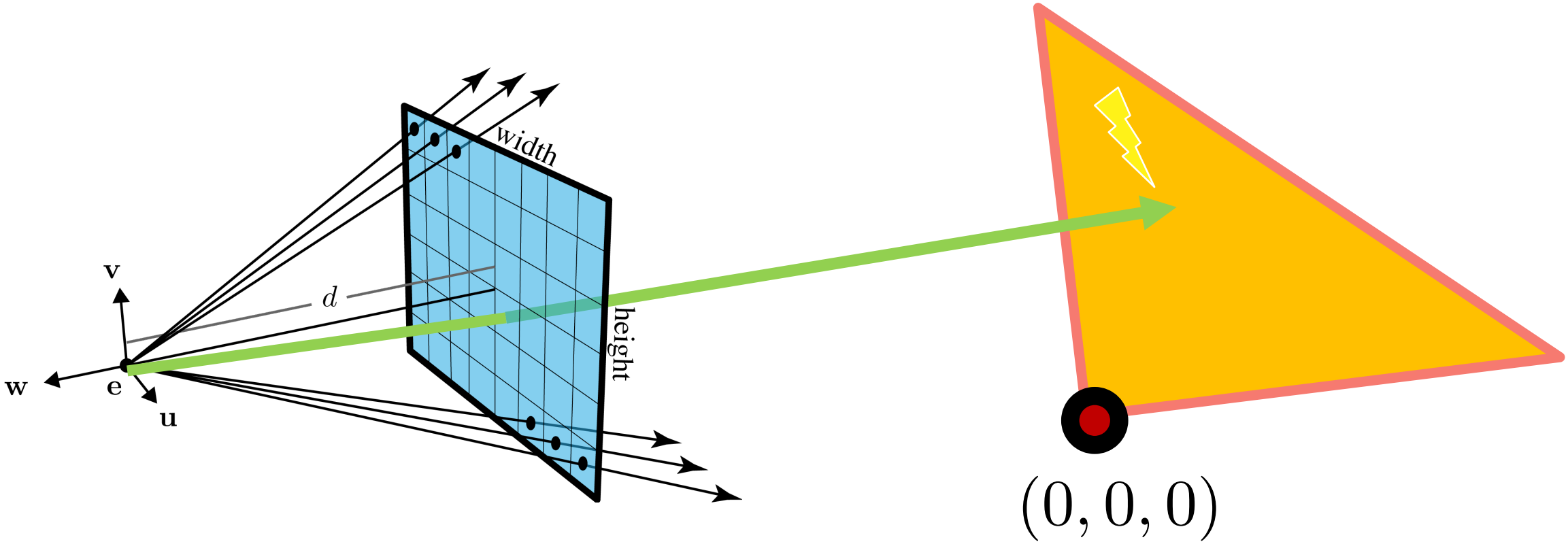
Intersection Tests

Plane

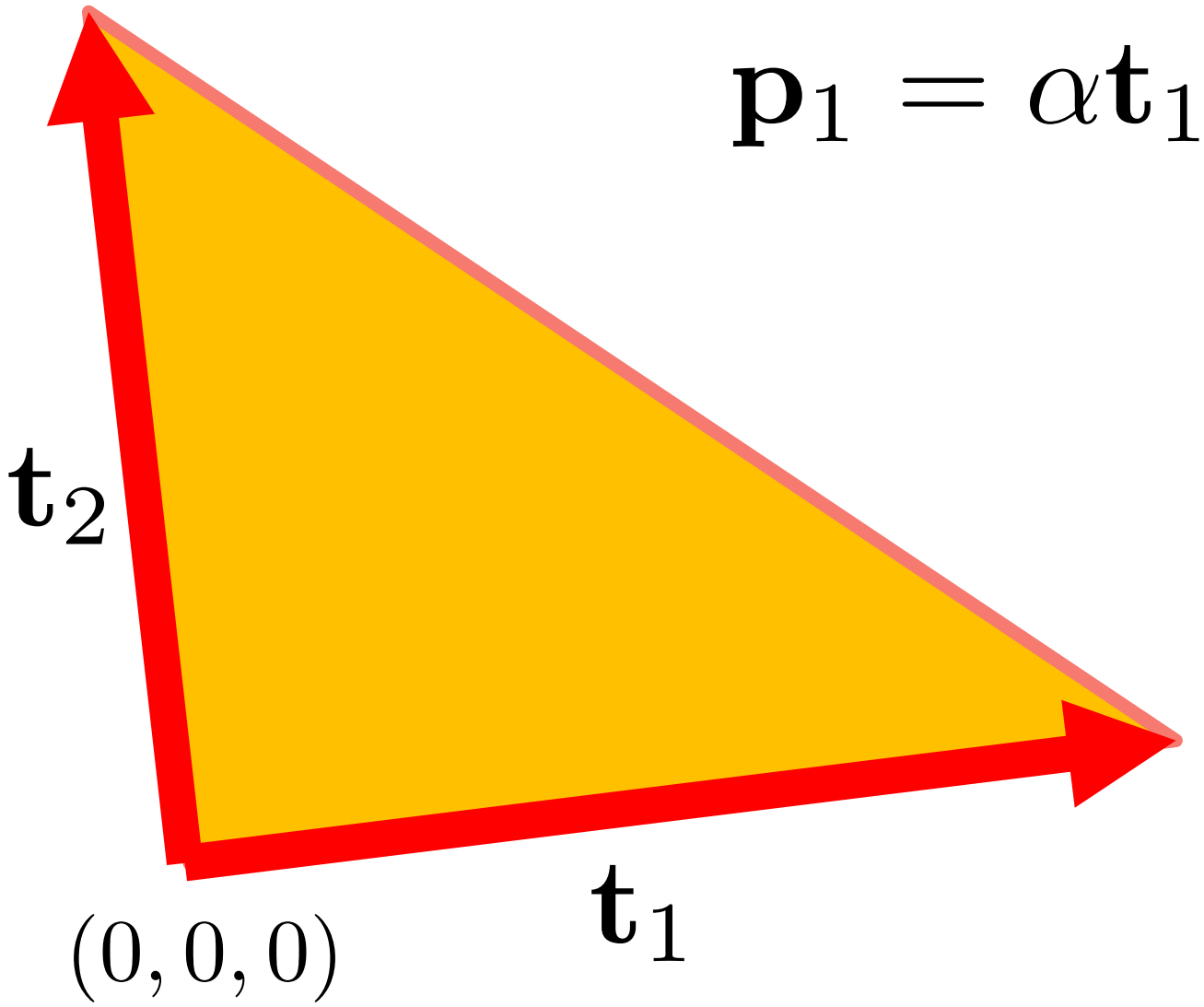
Sphere

Triangle

Ray-Triangle Intersection



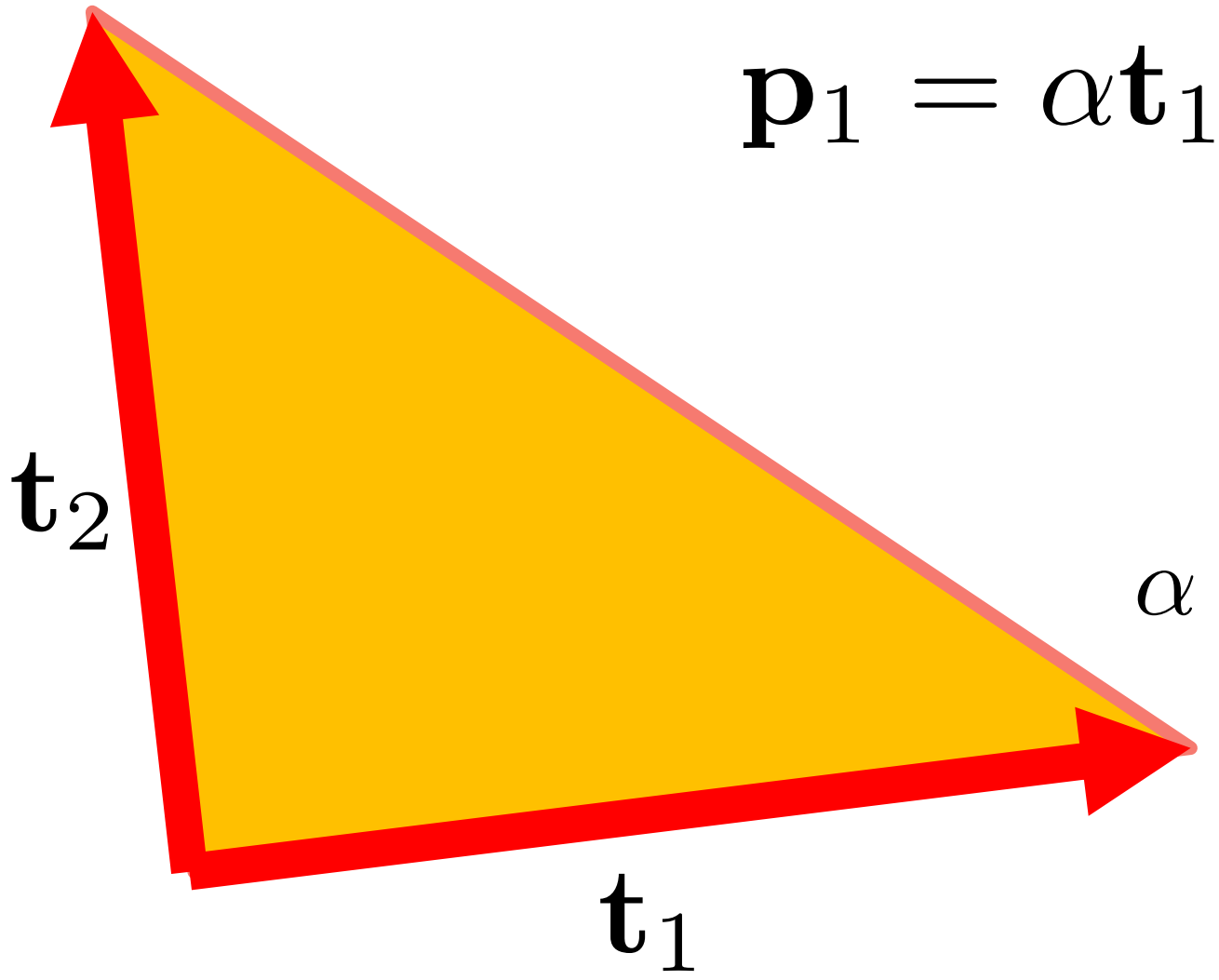
Equations for a Triangle



$$\mathbf{p}_1 = \alpha\mathbf{t}_1 + \beta\mathbf{t}_2$$

You will have to generalize this for your assignment!

Equations for a Triangle



$$\mathbf{p}_1 = \alpha\mathbf{t}_1 + \beta\mathbf{t}_2$$

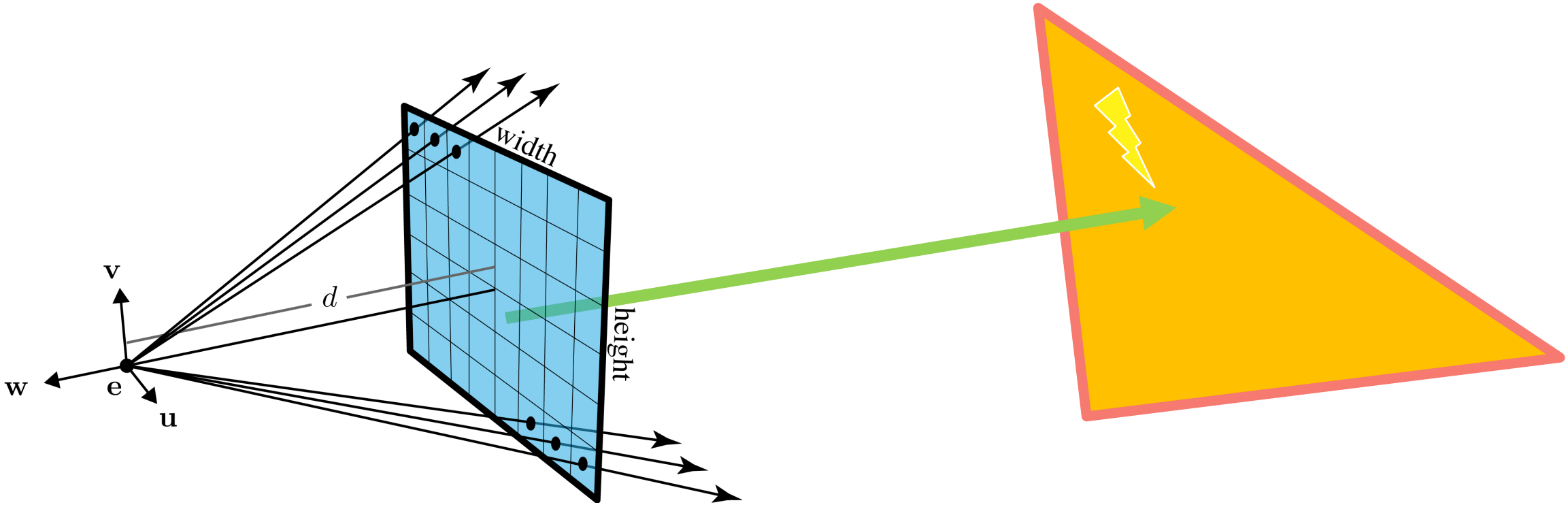
$$\alpha \geqslant 0$$

$$\beta \geqslant 0$$

$$\alpha + \beta \leqslant 1$$

Intersection with a Triangle (Parametric Surface)

Check via equating point on surface with point on ray



Intersection with a Triangle (Parametric Surface)

Check via equating point on surface with point on ray

$$\mathbf{p}(t) = \alpha\mathbf{t}_1 + \beta\mathbf{t}_2$$

$$\mathbf{e} + t(\mathbf{s} - \mathbf{e}) = \alpha\mathbf{t}_1 + \beta\mathbf{t}_2$$

$$\alpha\mathbf{t}_1 + \beta\mathbf{t}_2 - t(\mathbf{s} - \mathbf{e}) = \mathbf{e}$$

Intersection with a Triangle (Parametric Surface)

Check via equating point on surface with point on ray

$$\alpha \mathbf{t}_1 + \beta \mathbf{t}_2 - t\,(\mathbf{s} - \mathbf{e}) = \mathbf{e}$$

$$\begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 & -(\mathbf{s} - \mathbf{e}) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ t \end{bmatrix} = \mathbf{e}$$

Intersection with a Triangle (Parametric Surface)

Check via equating point on surface with point on ray

$$\alpha \mathbf{t}_1 + \beta \mathbf{t}_2 - t\,(\mathbf{s} - \mathbf{e}) = \mathbf{e}$$

$$\begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 & -(\mathbf{s} - \mathbf{e}) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ t \end{bmatrix} = \mathbf{e}$$

Check t, α and β

Ray Casting

```
for each pixel in the image {
    Generate a ray
    for each object in the scene {
        if (Intersect ray with object) {
            Set pixel colour
        }
    }
}
```

Output Type

Object ID

Surface Normal

Depth

Ray Casting

```
for each pixel in the image {
    Generate a ray
    for each object in the scene {
        if (Intersect ray with object) {
            Set pixel colour
        }
    }
}
```

Done for Today

Assignment 1 is due this Wednesday

Assignment 2 Available today (due 27/09)