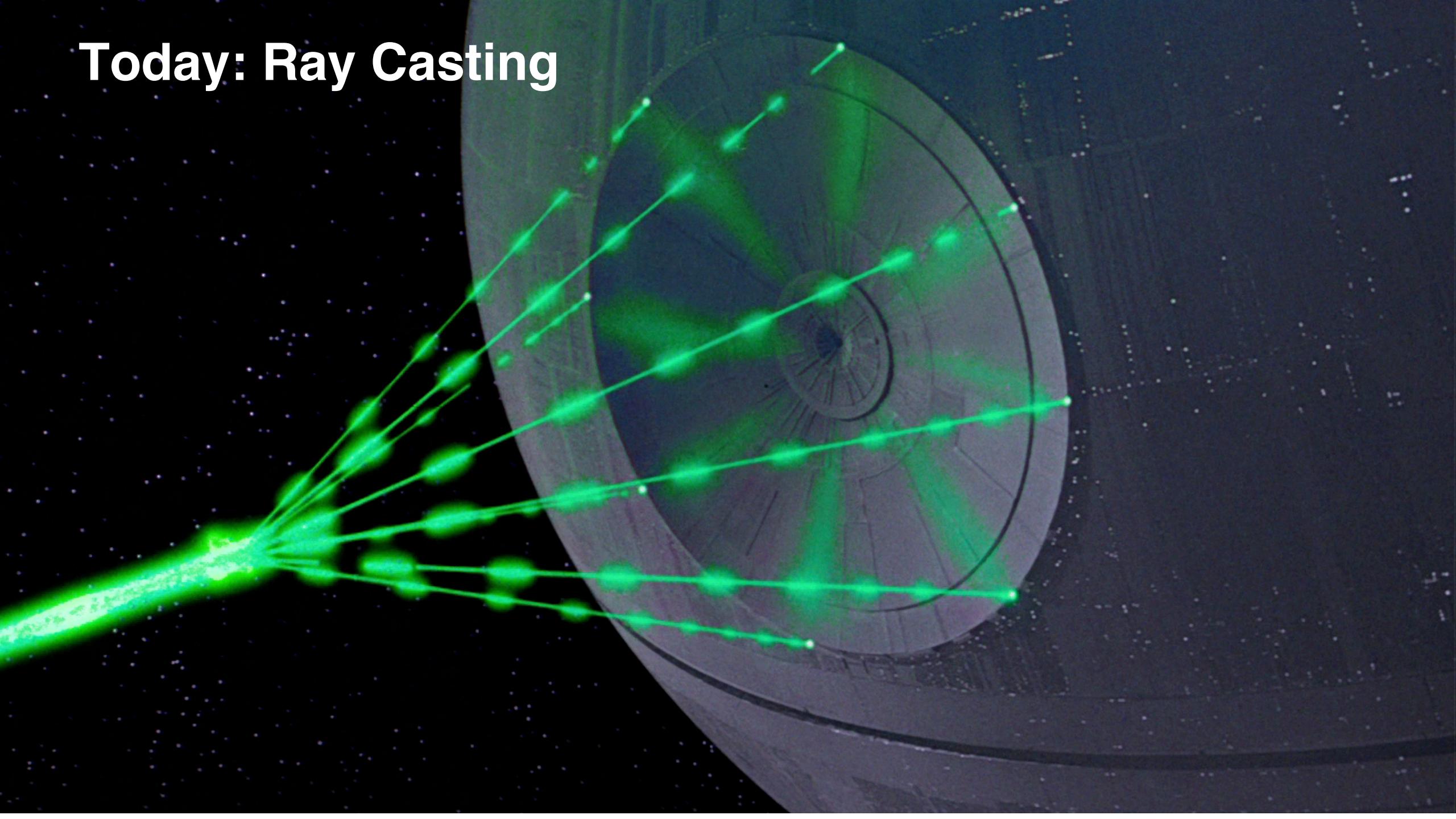


# csc418/2504 Computer Graphics

Rob Katz

Some Slides/Images adapted from Marschner and Shirley

# Today: Ray Casting



# Announcements

Assignment 1 is due this Friday

Assignment 2 is available now (due 25/01)

TA Office Hours This Week:

***Thursday, 17th January, 11:30-12:30pm***

***You'll get an email about the room***

# Any Questions ?



**SNL**  
SUBSCRIBE



# Today: Ray Casting

The Ray Casting Algorithm

Introduction to Rays

The Camera

Ray-Object Intersection

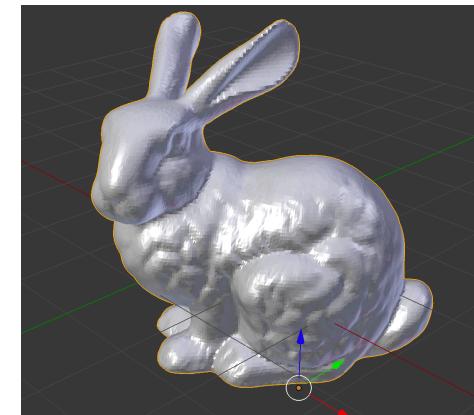
Ray-Plane Intersection

Ray-Sphere Intersection

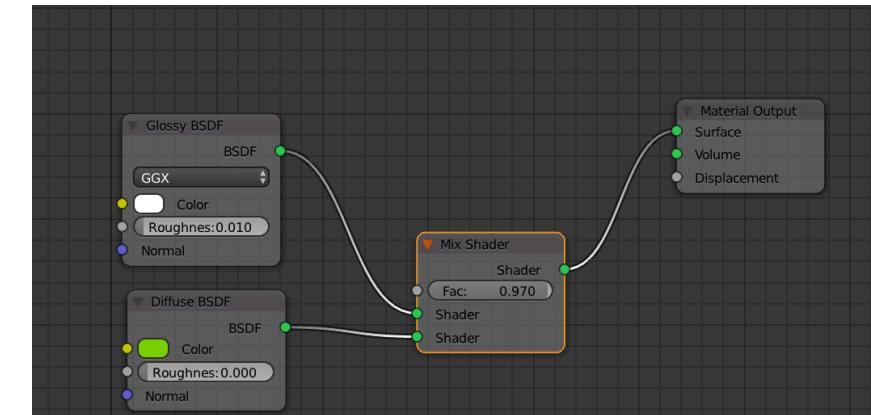
Ray-Triangle Intersection

# Rendering

Input:



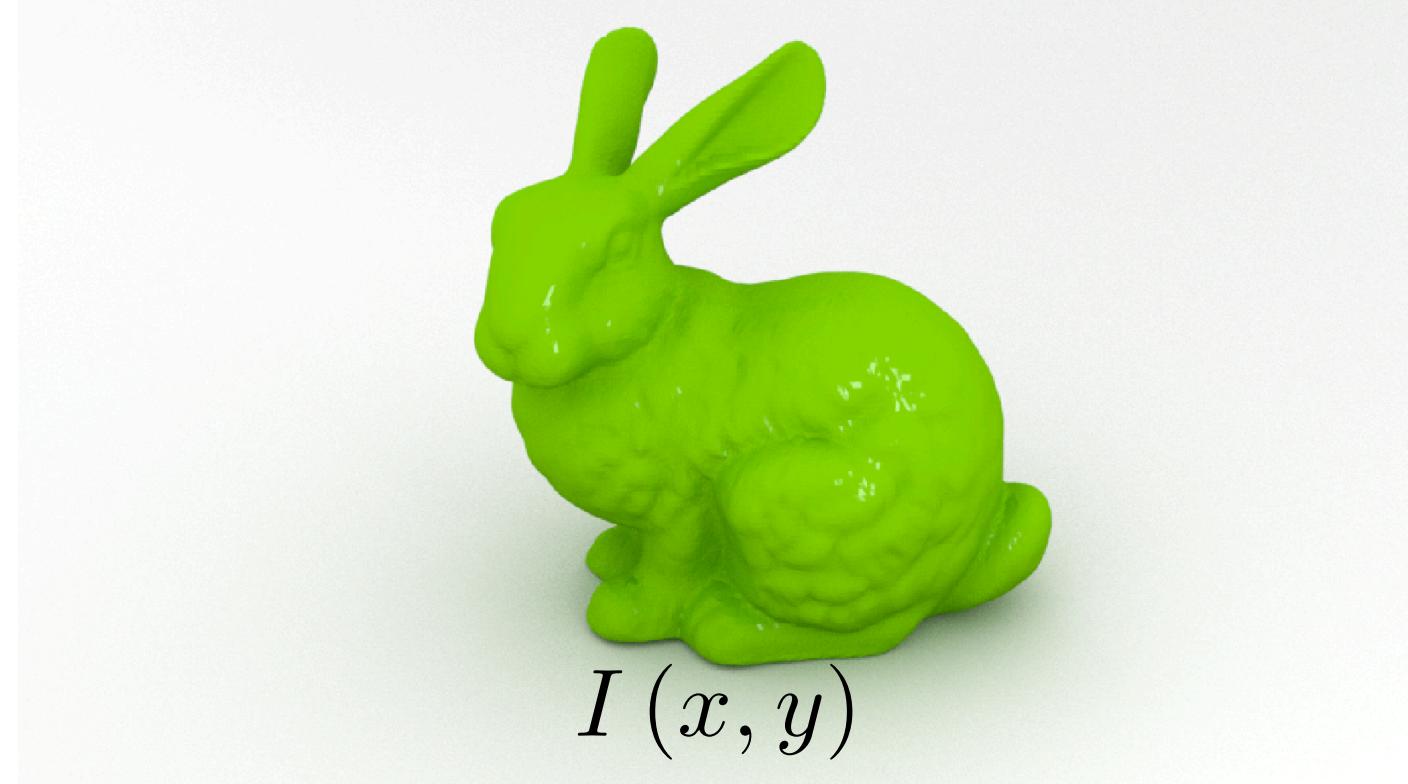
Objects



Materials

---

Output:



$$I(x, y)$$

zvarownik  
13878049a9f142c2b036e491198e2149

◆ WARIOR\_GAMING\_57

● YZx\_Vulka

◆ danielek185

◆ zvarownik

NW 300  
330 345

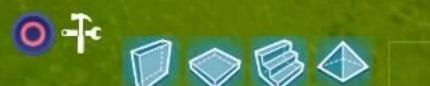
danielek185

N 15  
30 60  
NE 75

E



██████████	██████████	0   5
██████████	██████████	0   0
██████████	██████████	0   0



0 | 100

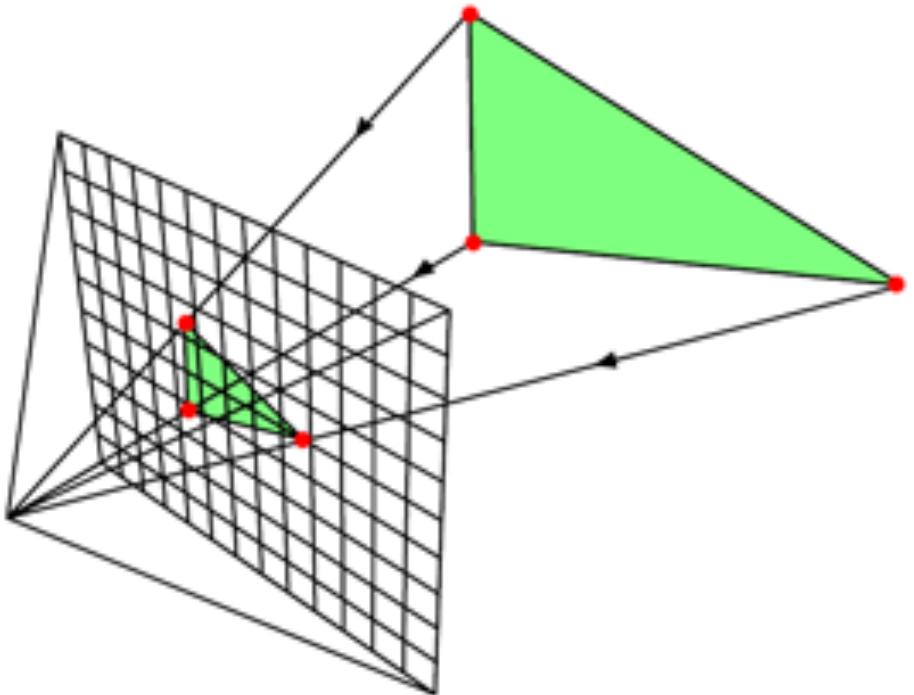
+ 100 | 100



Przytrzymaj

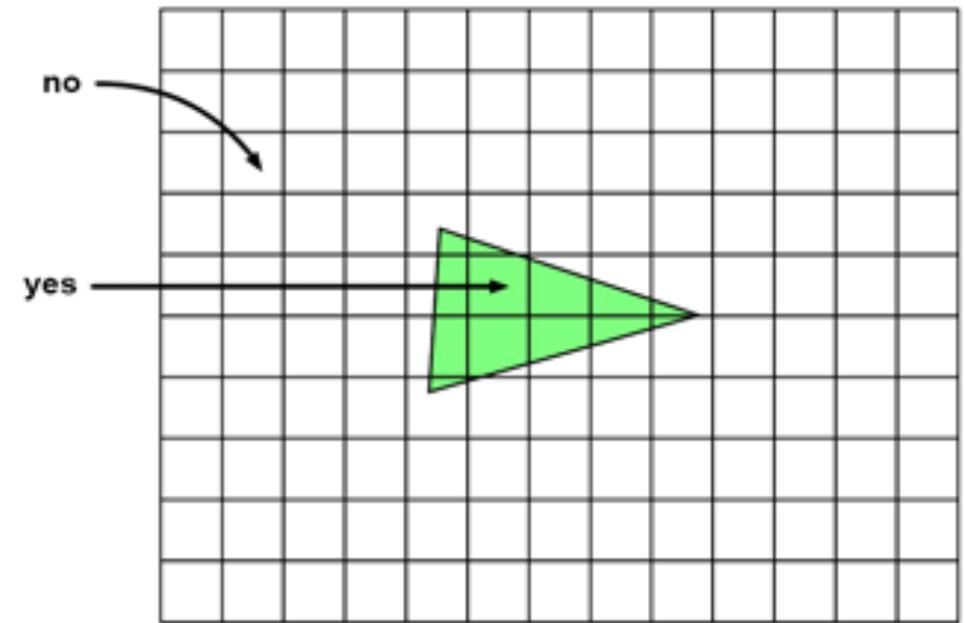


# Rasterization



1. Project Vertices to Image Plane

© www.scratchapixel.com



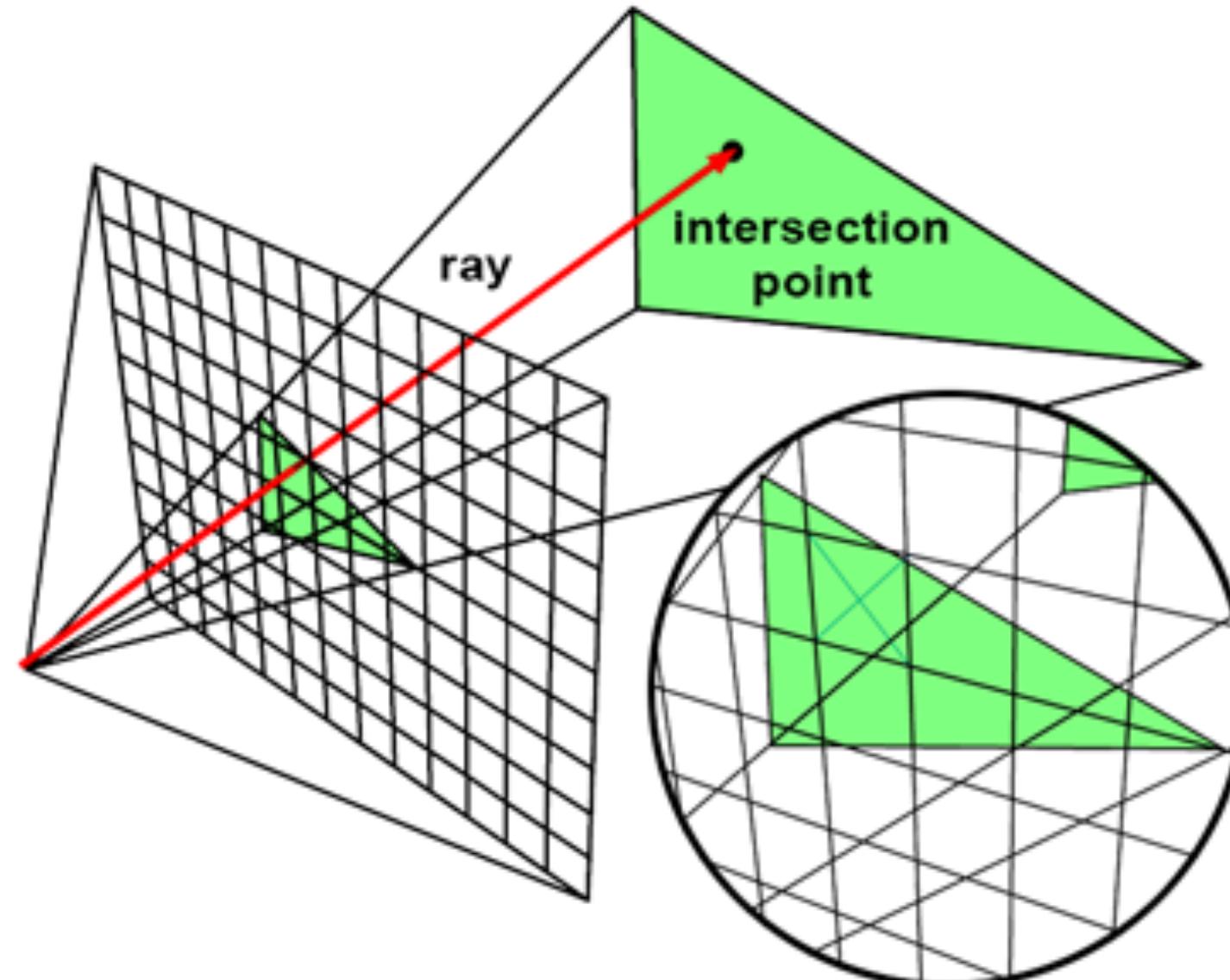
2. Turn on pixels inside triangle

# Rasterization

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



# Ray Casting



# 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  
        }  
    }  
}
```



[https://en.wikipedia.org/wiki/Ray\\_tracing\\_\(graphics\)](https://en.wikipedia.org/wiki/Ray_tracing_(graphics))

# **Basic Components of Ray Casting**

Ray

Camera

Intersection Tests

# Basic Components of Ray Casting

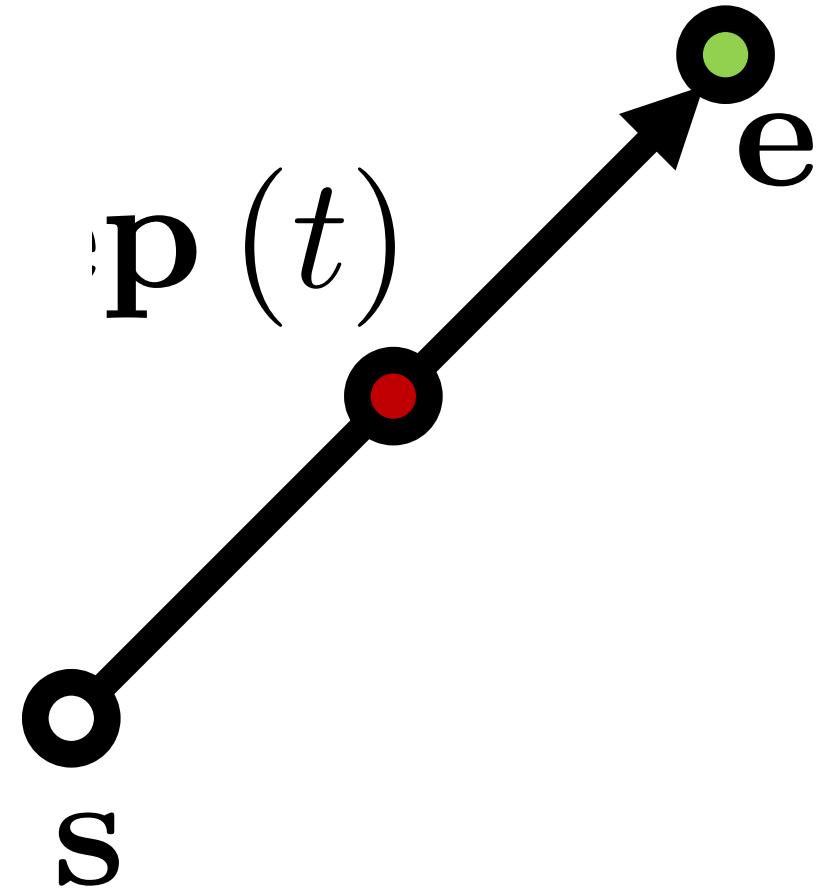
Ray

Camera

Intersection Tests

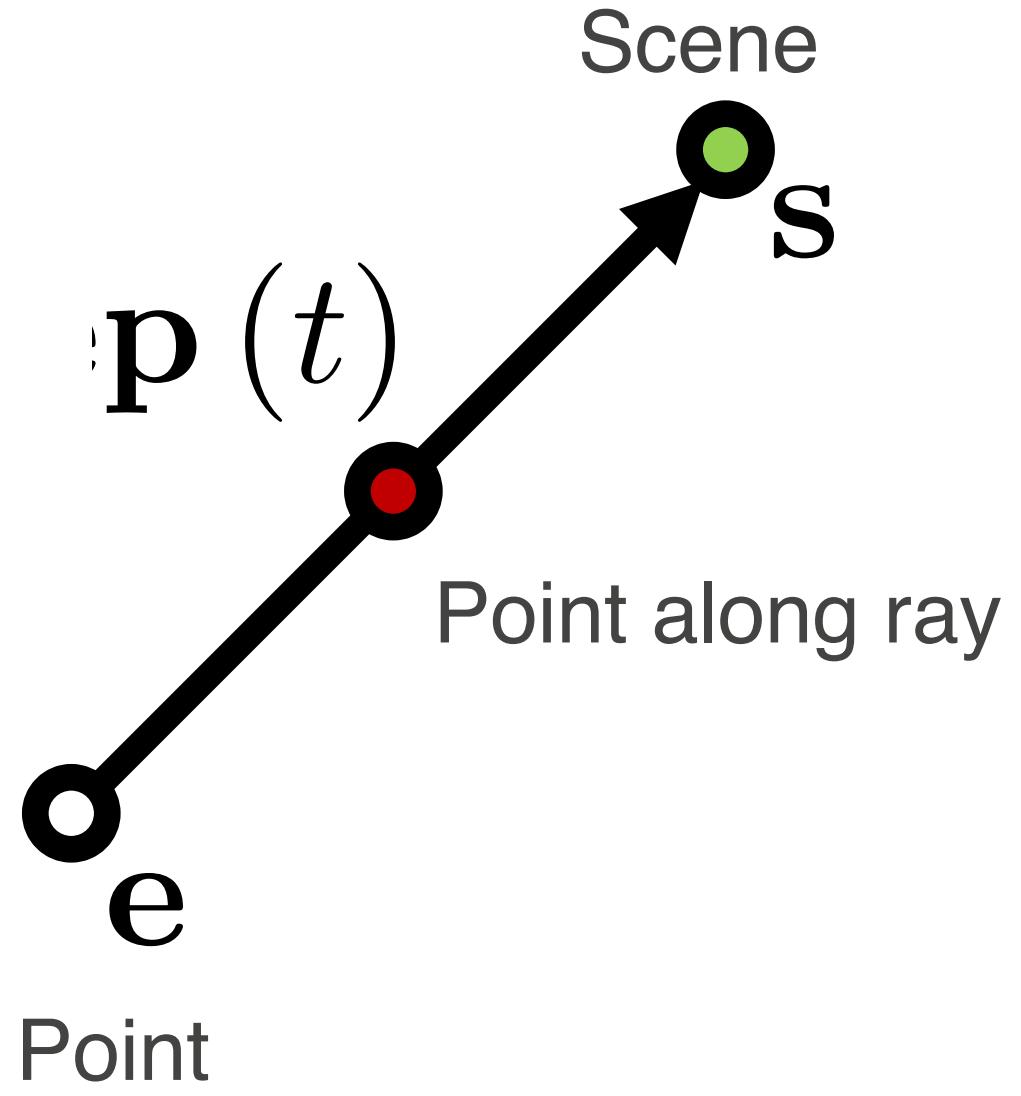
# The Ray

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

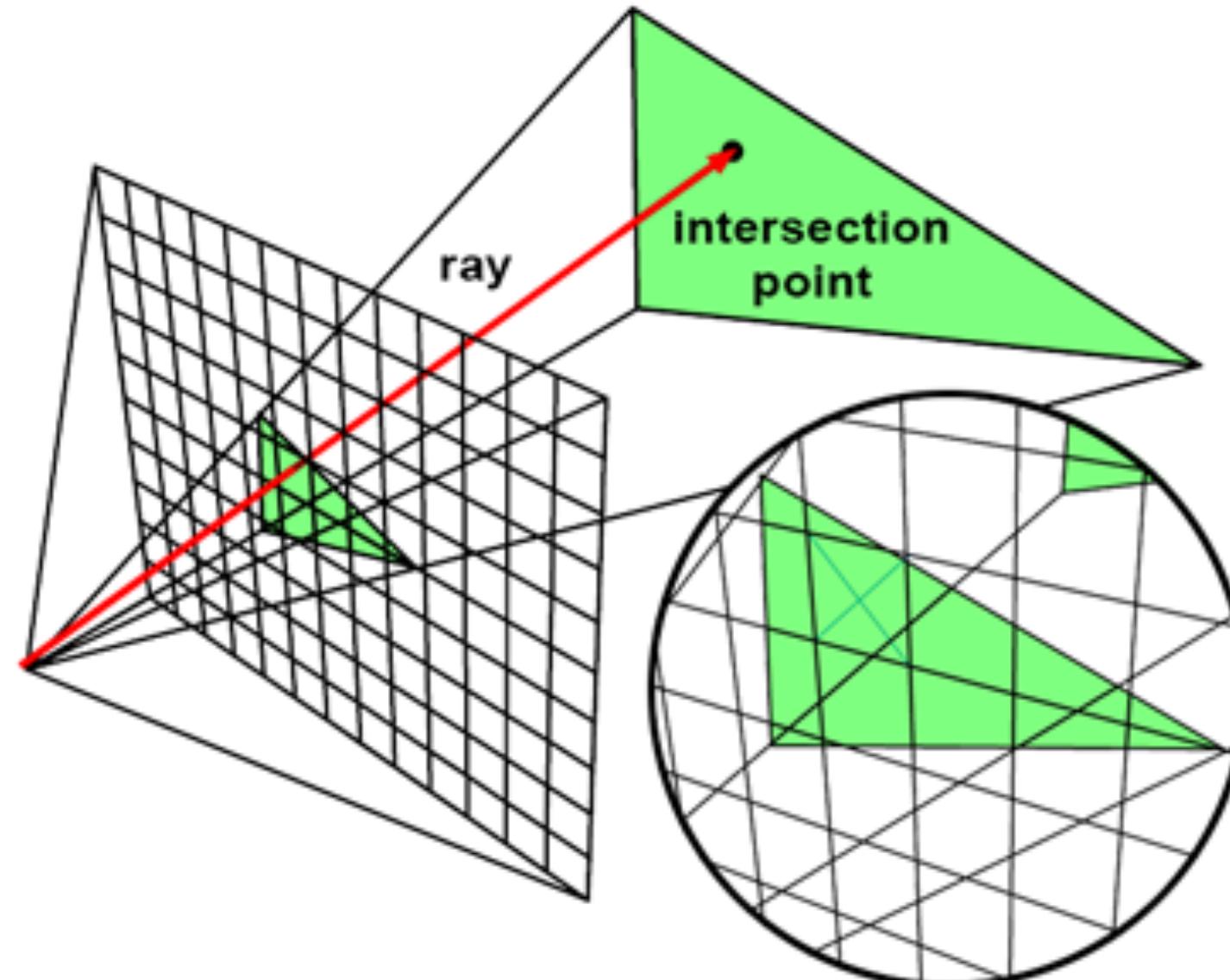


# The Ray

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



# Ray Casting



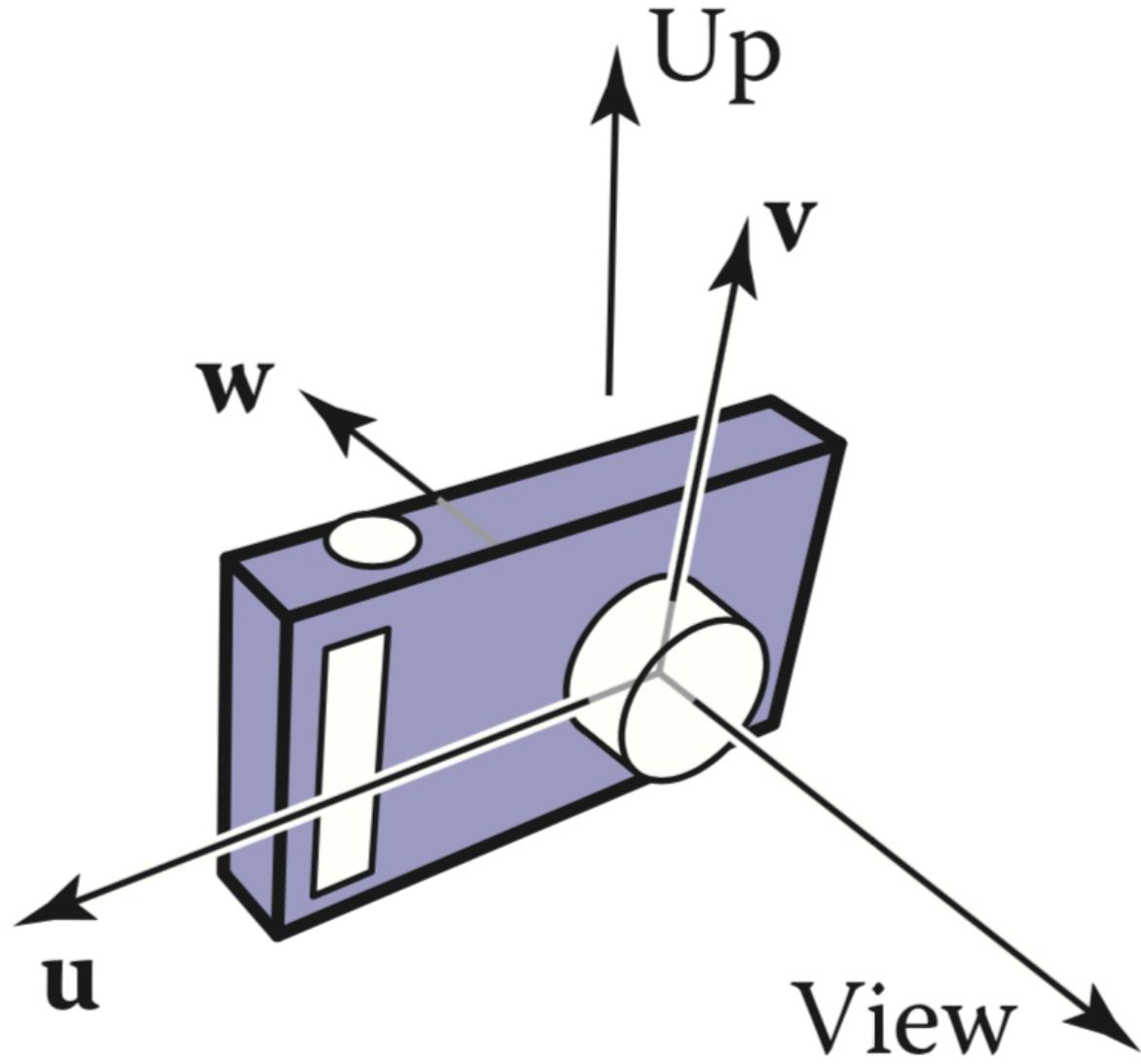
# **Basic Components of Ray Casting**

Ray

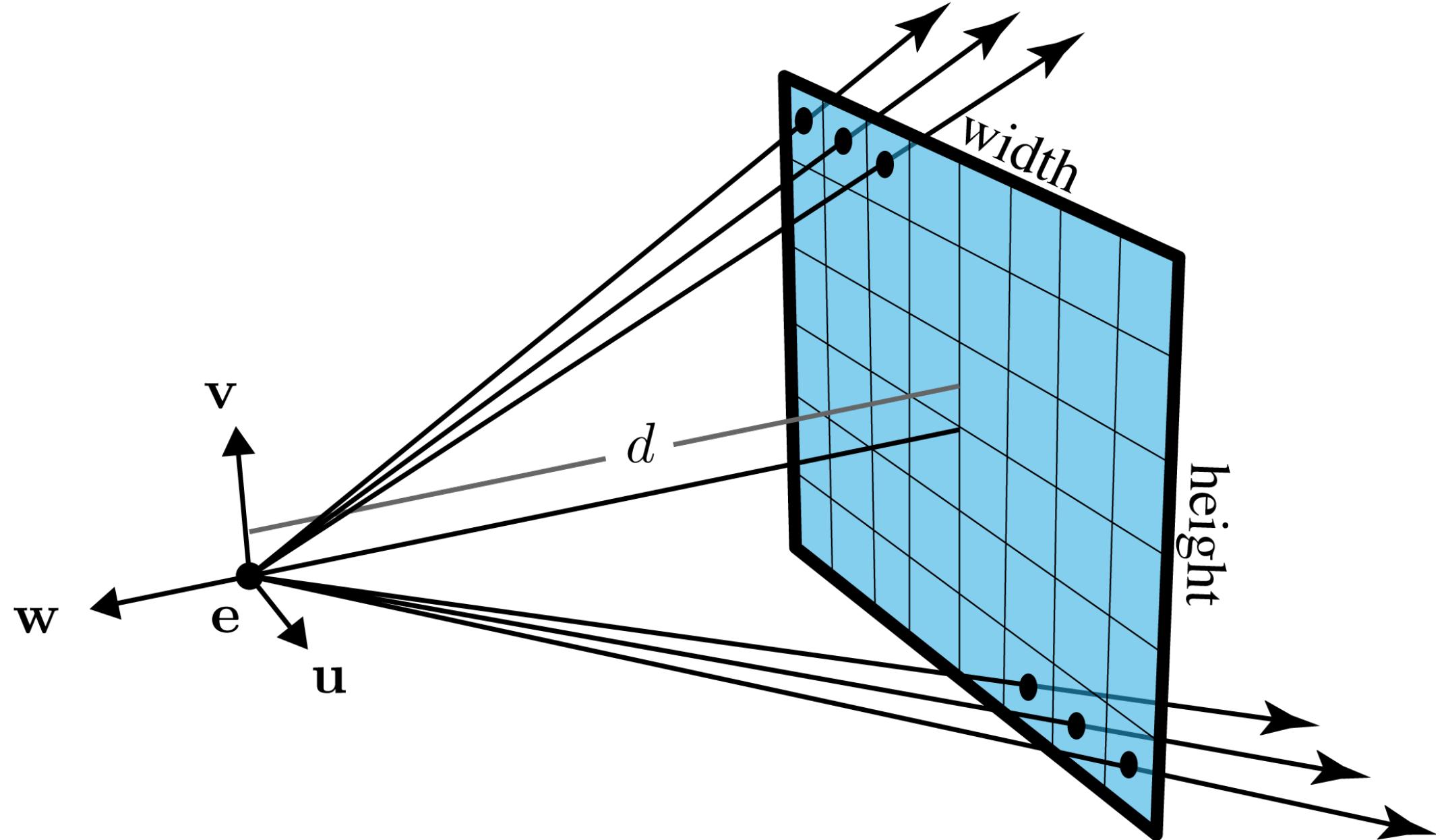
Camera

Intersection Tests

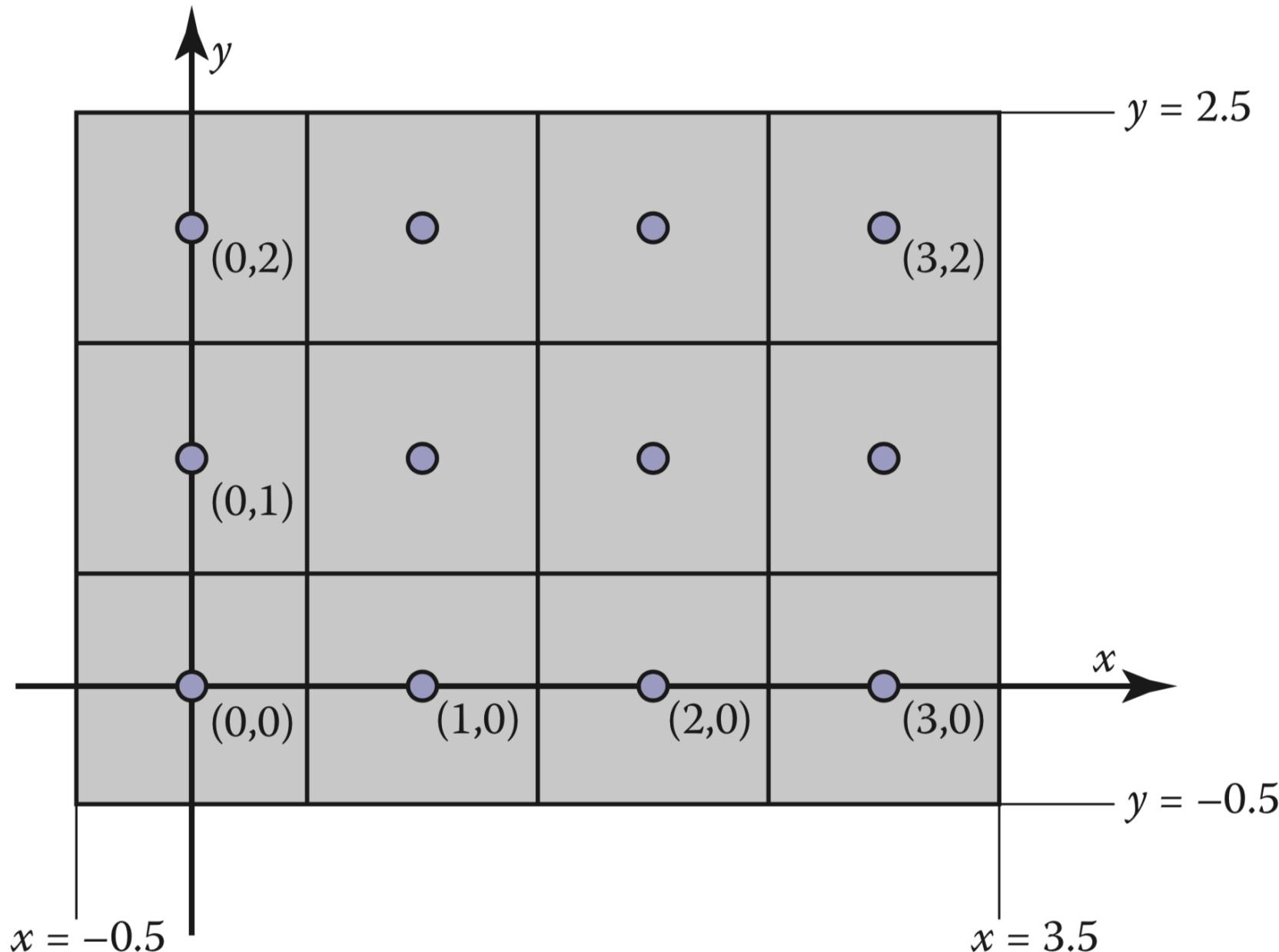
# The Camera

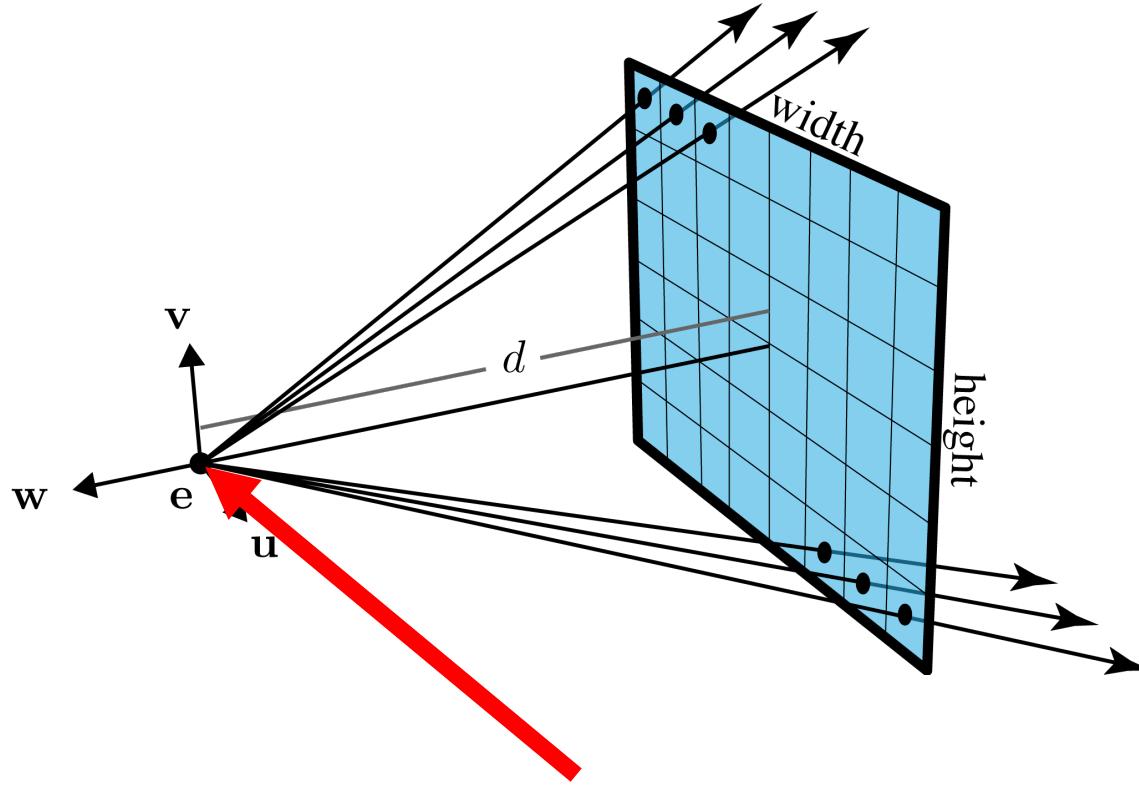


# Generating Rays

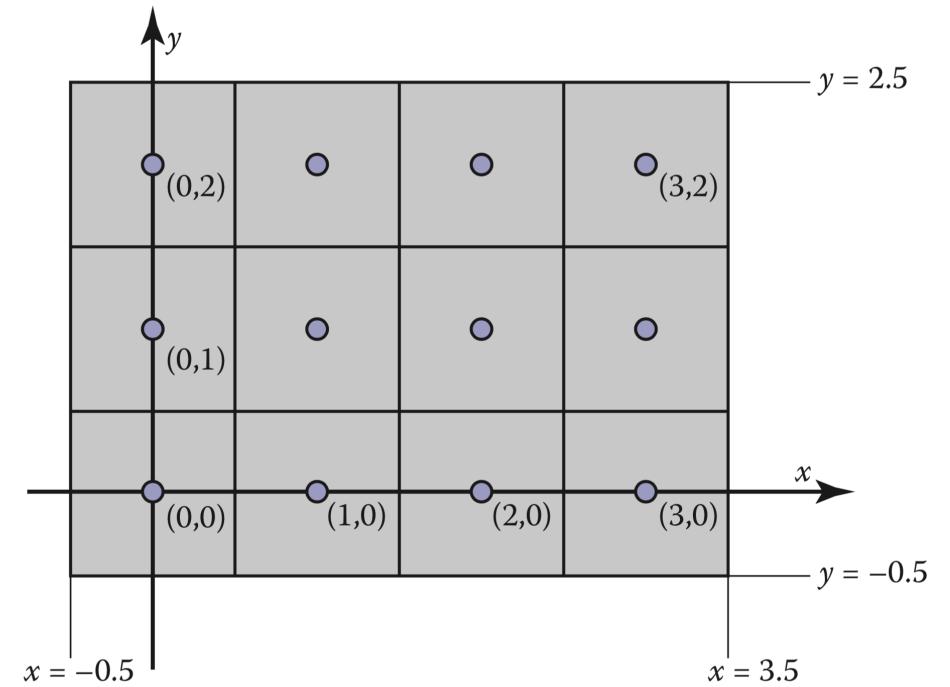


# Recall: Standard Pixel Coordinate System



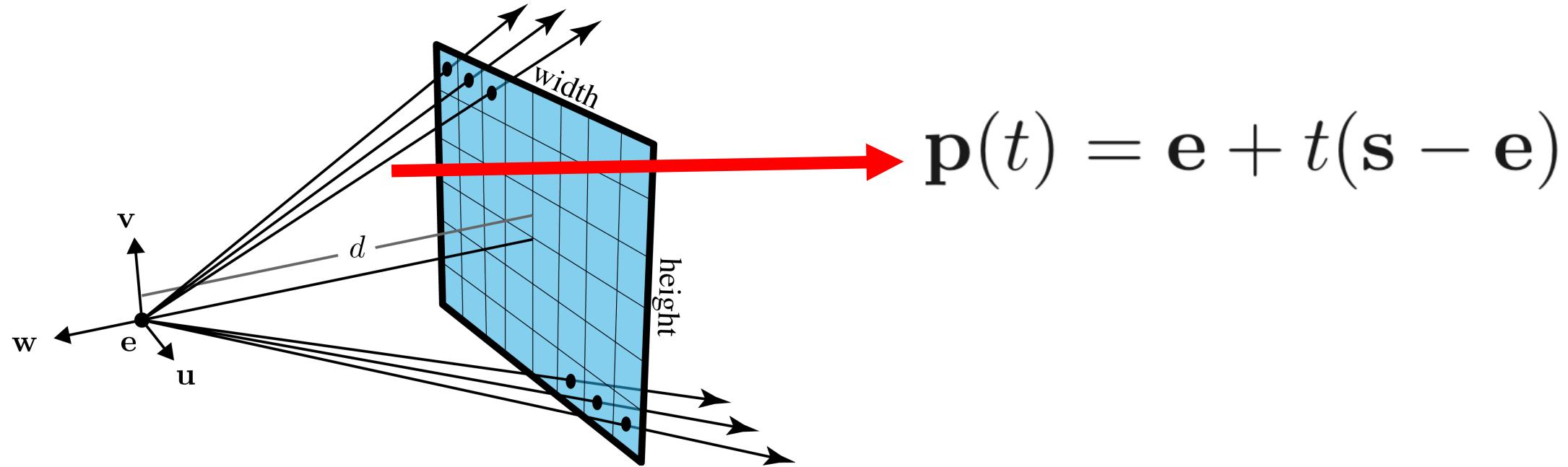


Origin of camera frame (the eye)



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

# Ray Equation in Camera Space



# 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

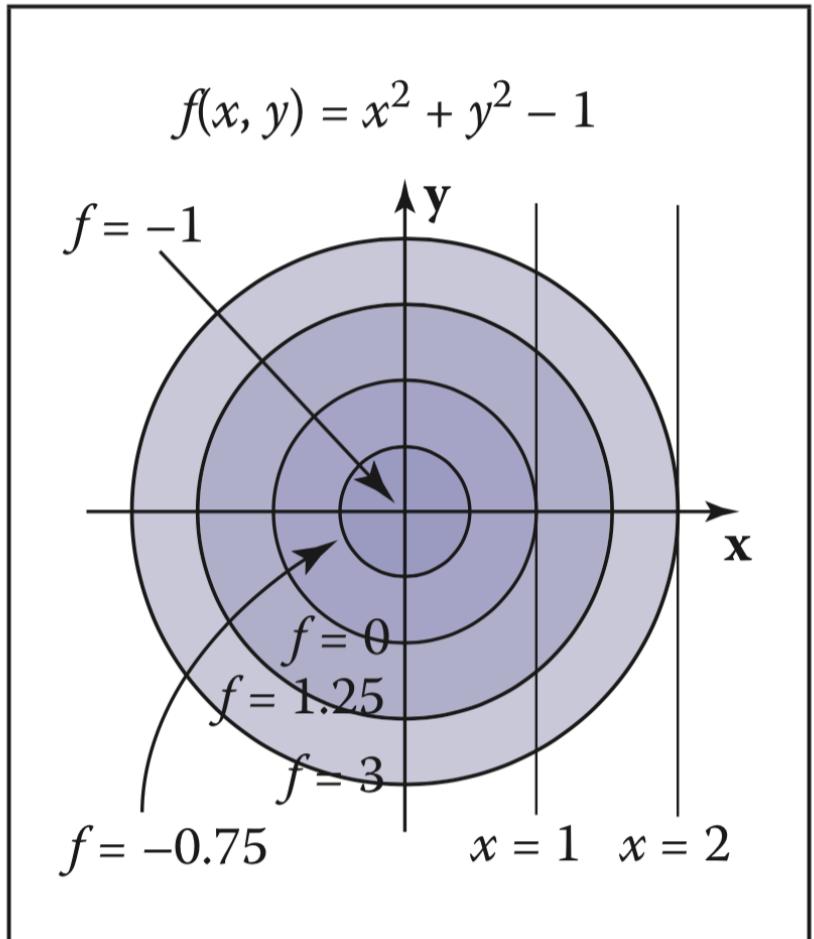
Plane

Sphere

Triangle

# Aside: Types of Surface

## Implicit Surface



## Parametric Surface

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

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

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



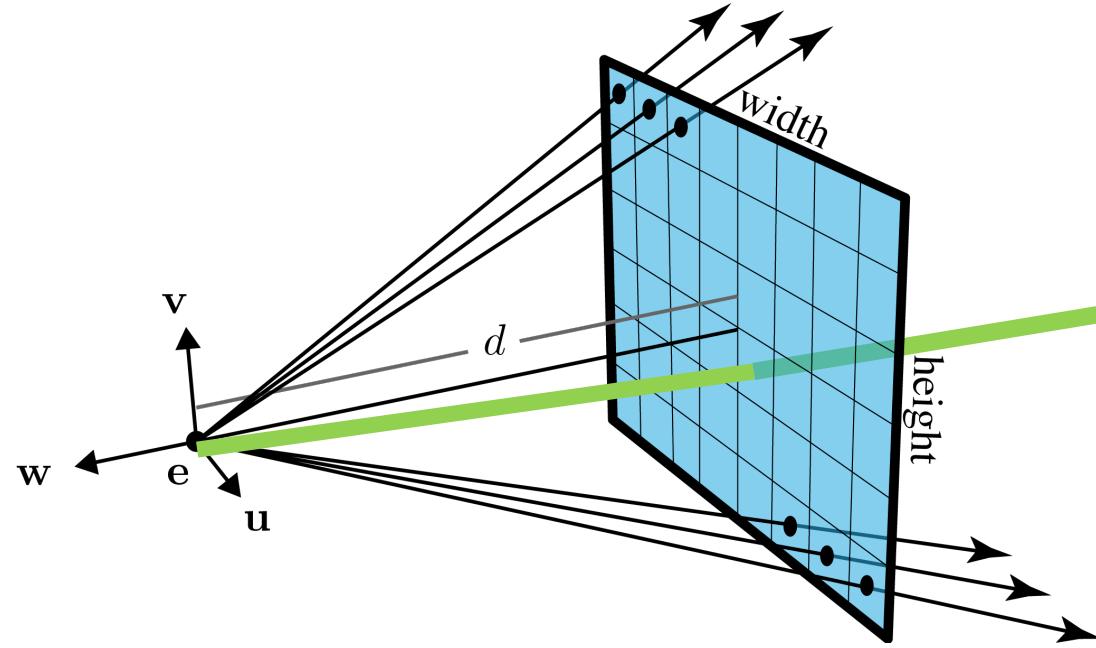
# Intersection Tests

Plane

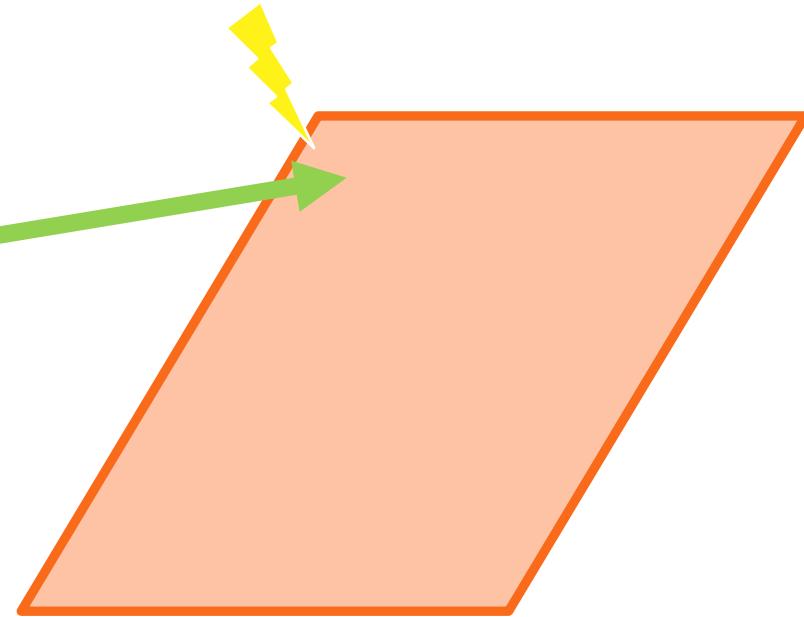
Sphere

Triangle

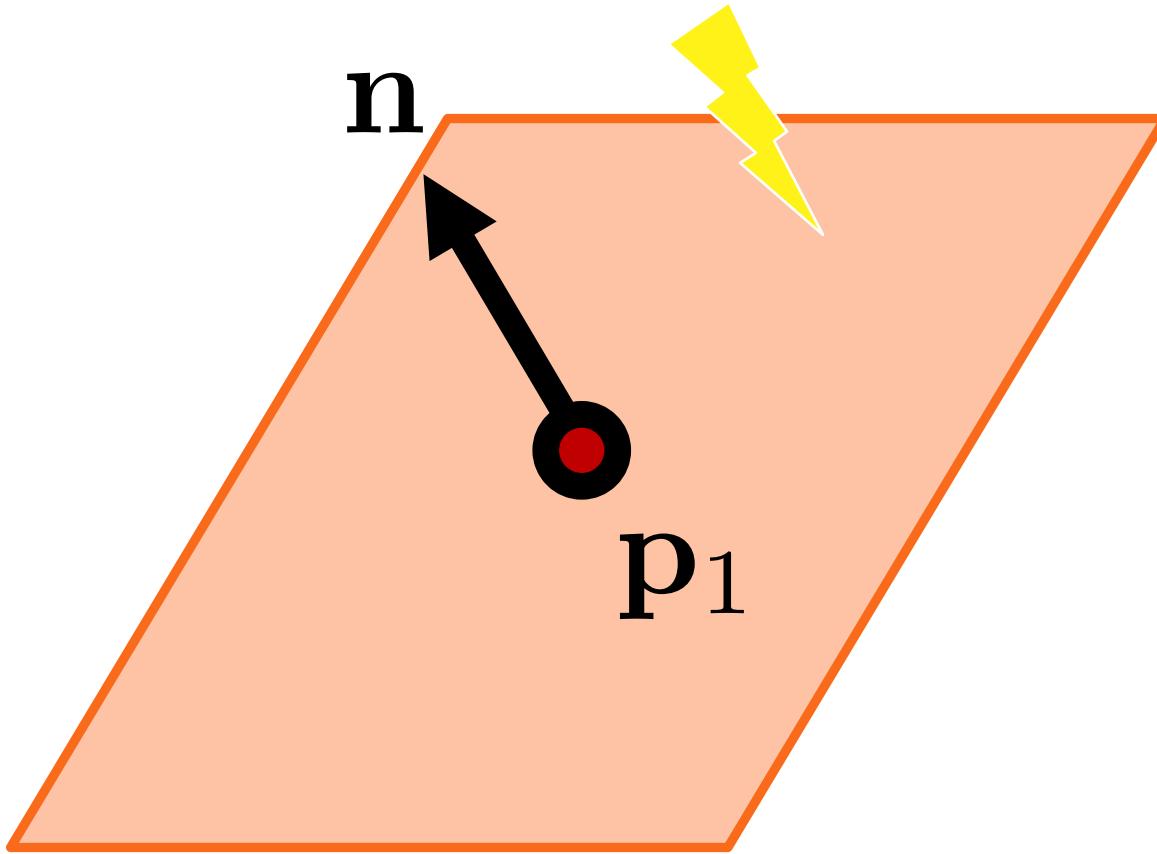
# Ray-Plane Intersection



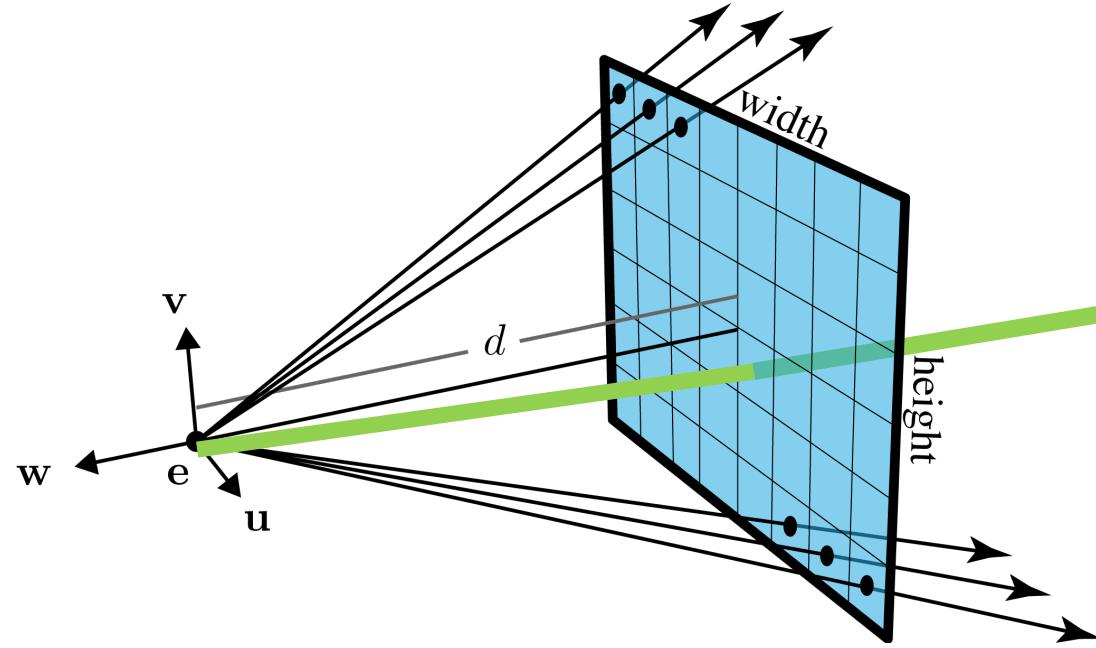
Intersection



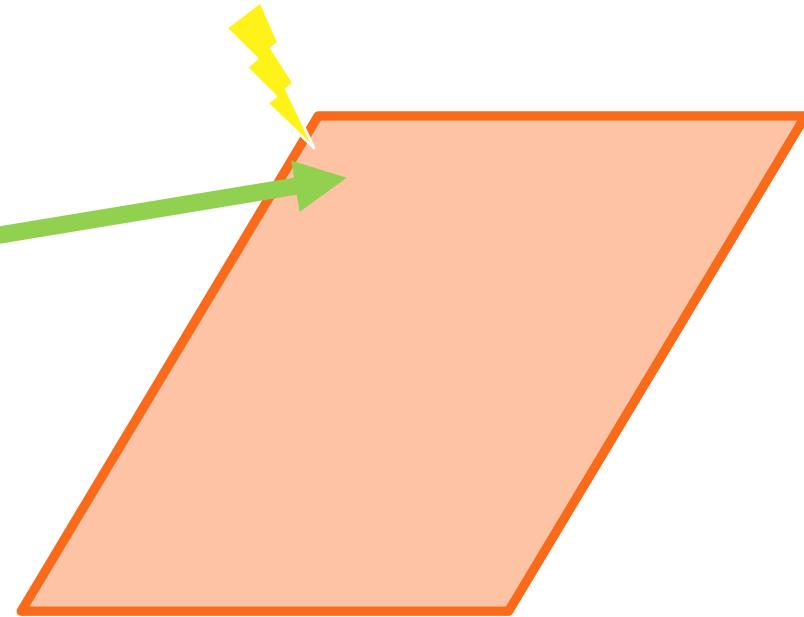
# Plane Equation



# Ray-Plane Intersection



Intersection



# Intersection Tests

Plane

Sphere

Triangle

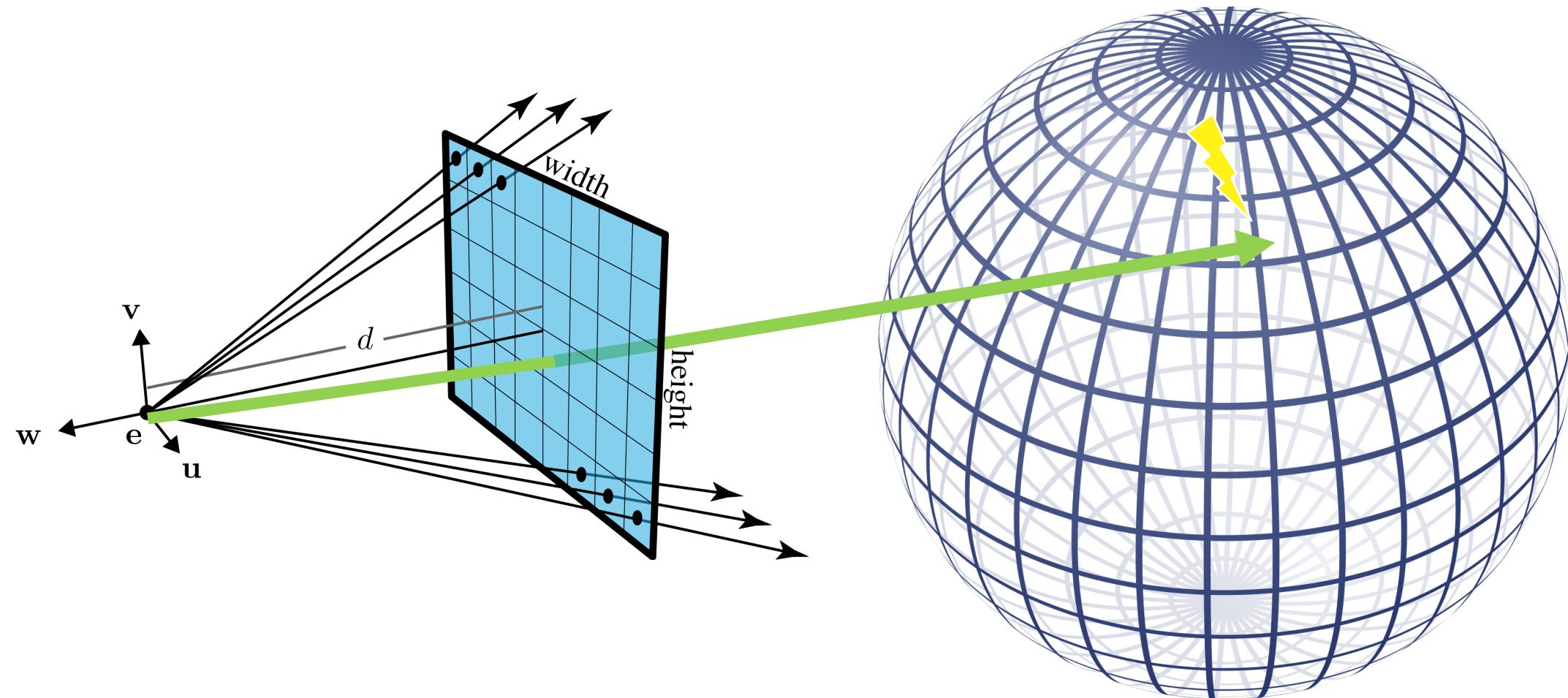
# Intersection Tests

Plane

Sphere

Triangle

# Ray-Sphere Intersection



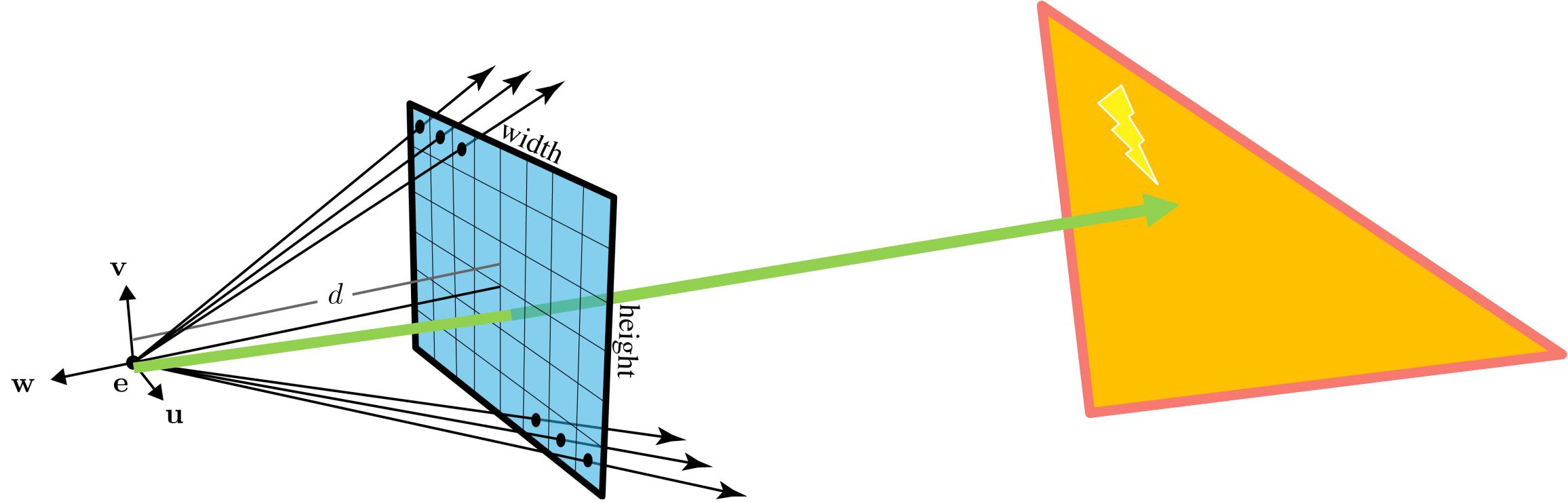
# Intersection Tests

Plane

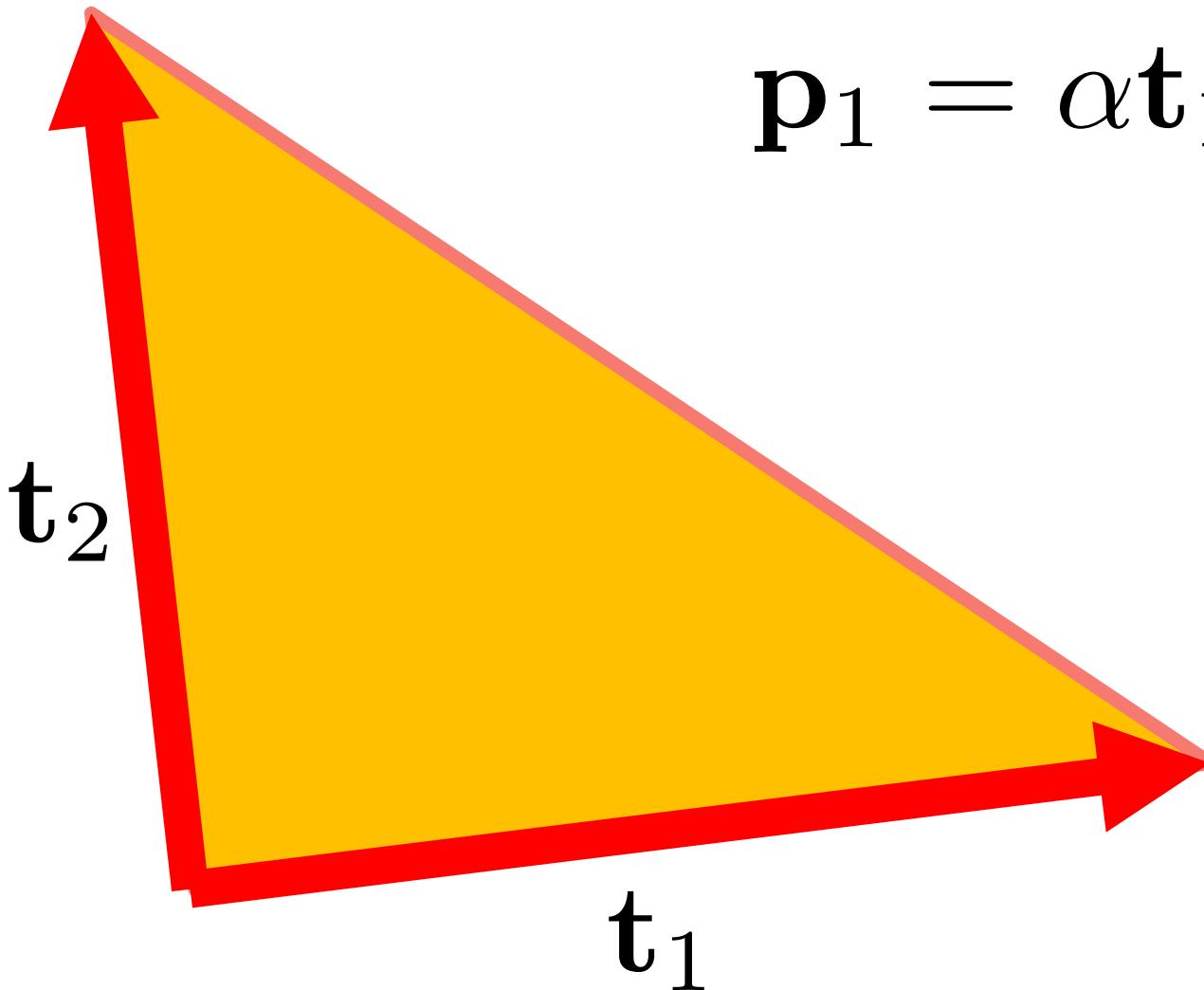
Sphere

Triangle

# Ray-Triangle Intersection



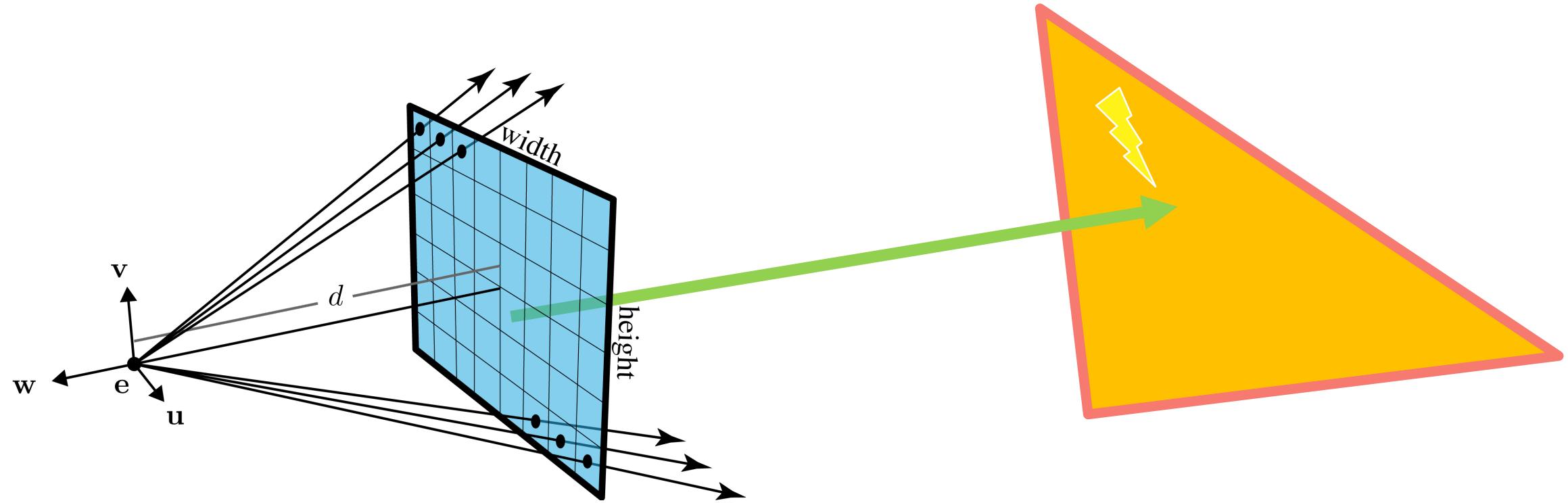
# Equations for a Triangle



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

# Intersection with a Triangle (Parametric Surface)

Check via equating point on surface with point on ray



# 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  
        }  
    }  
}
```