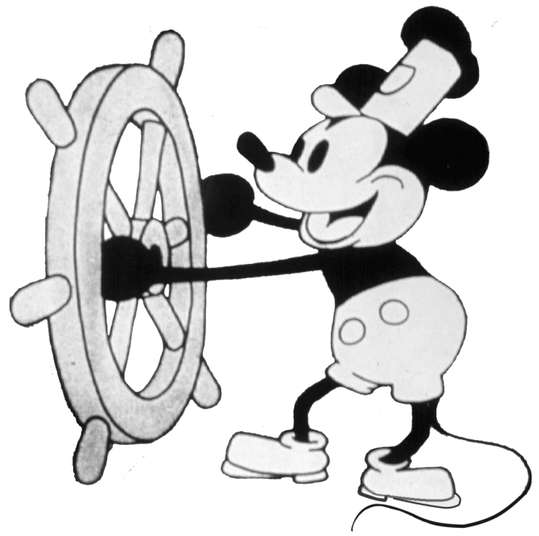


CSCI 461: Computer Graphics

Middlebury College, Spring 2025

Lecture 03: Ray Tracing

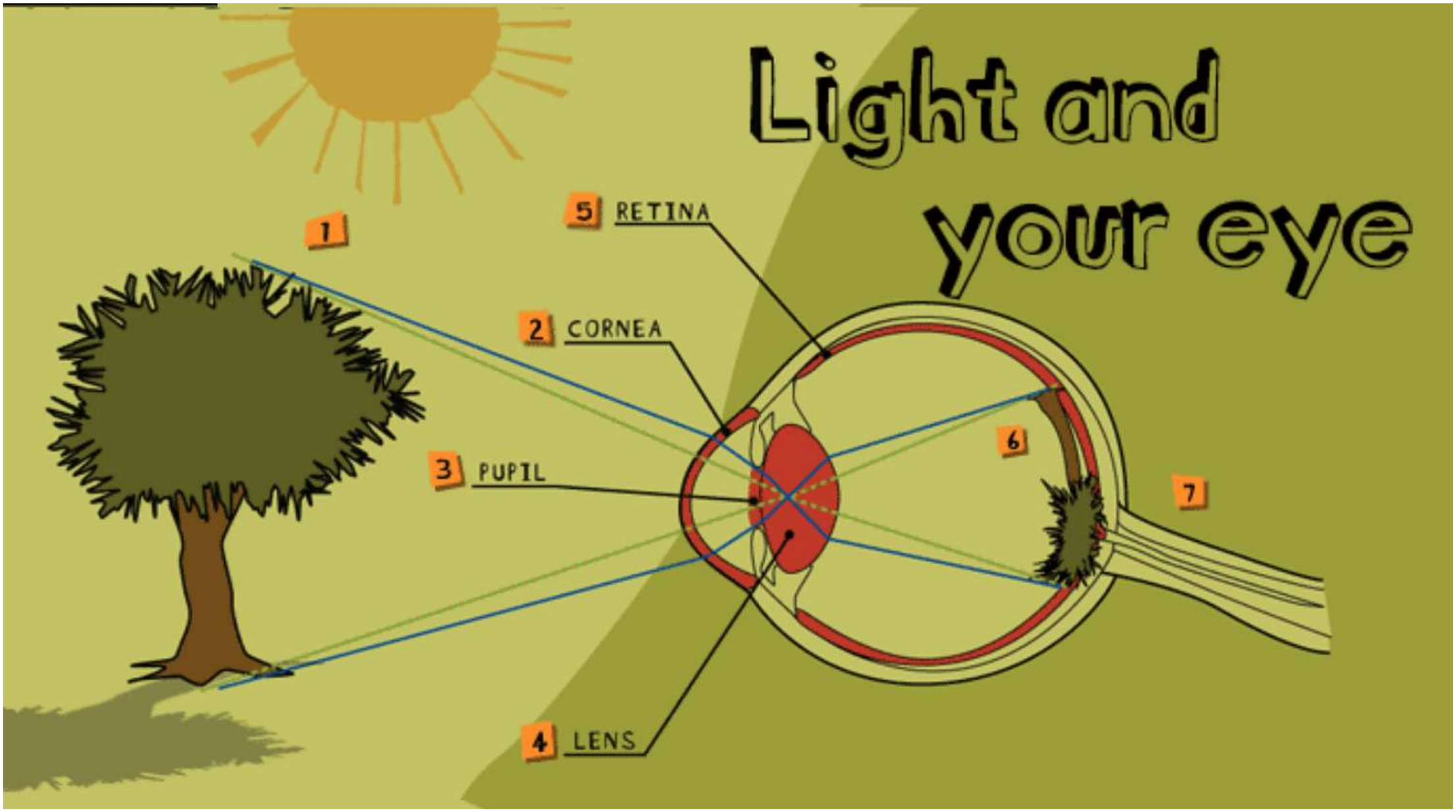
By the end of today's lecture, you will be able to:

- generate camera rays for a view directly aligned with the z-axis,
- intersect rays with simple geometric primitives, such as planes, spheres and triangles,
- practice some more with the math used in computer graphics.

Search Google Images for "ray tracing."

Look for images that show ray tracing on/off. What kinds of differences do you observe?

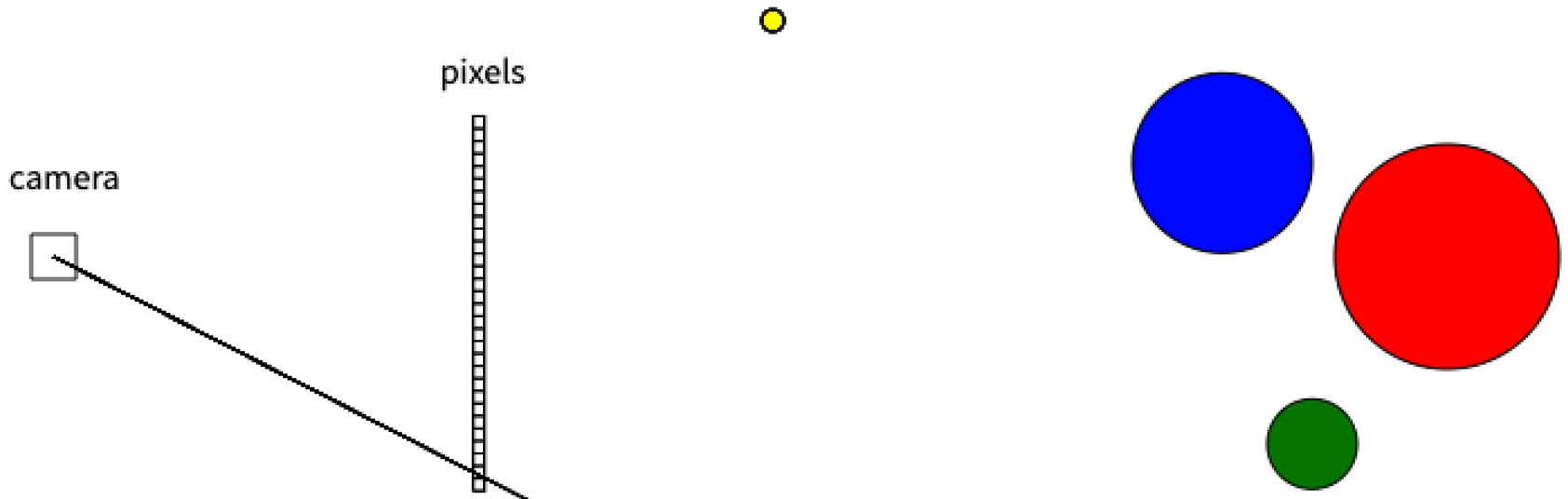
How do we see things?



(source)

The main idea of ray tracing.

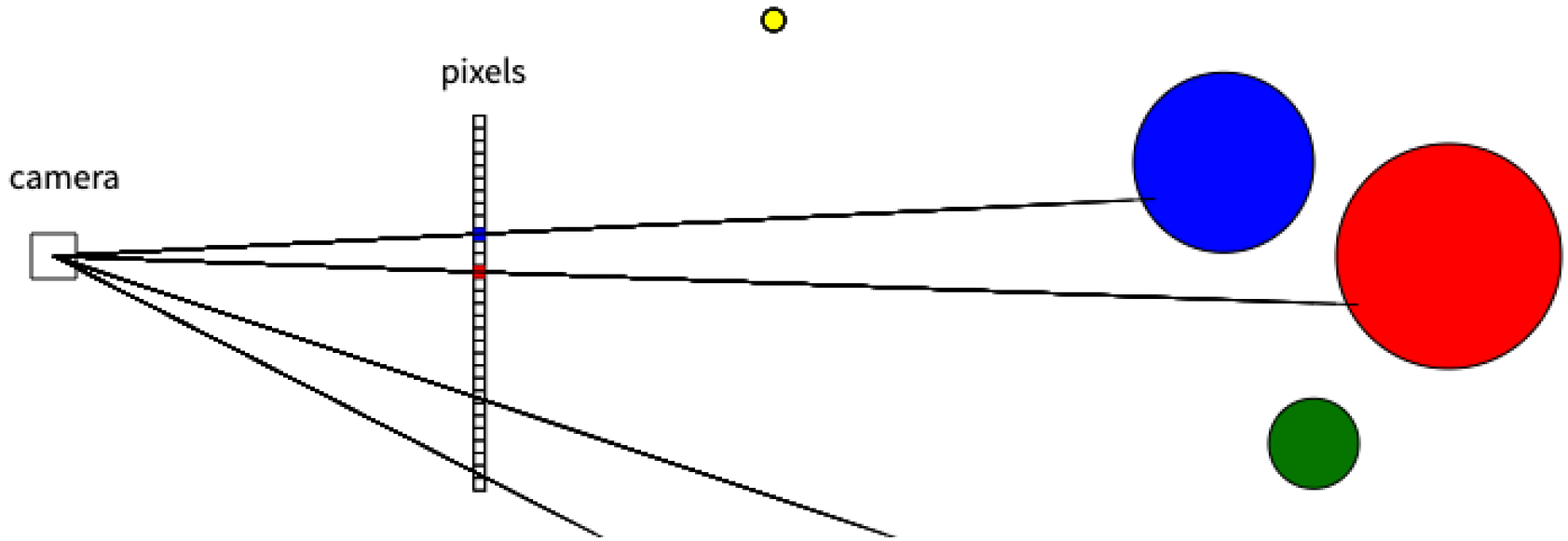
Send "rays" from an observation point (e.g. a camera) through each pixel in an image and into the (virtual) 3d world. Whatever is hit by the ray defines the color of the pixel.



Click to open the ray tracing demo.

Ingredients for a ray tracer.

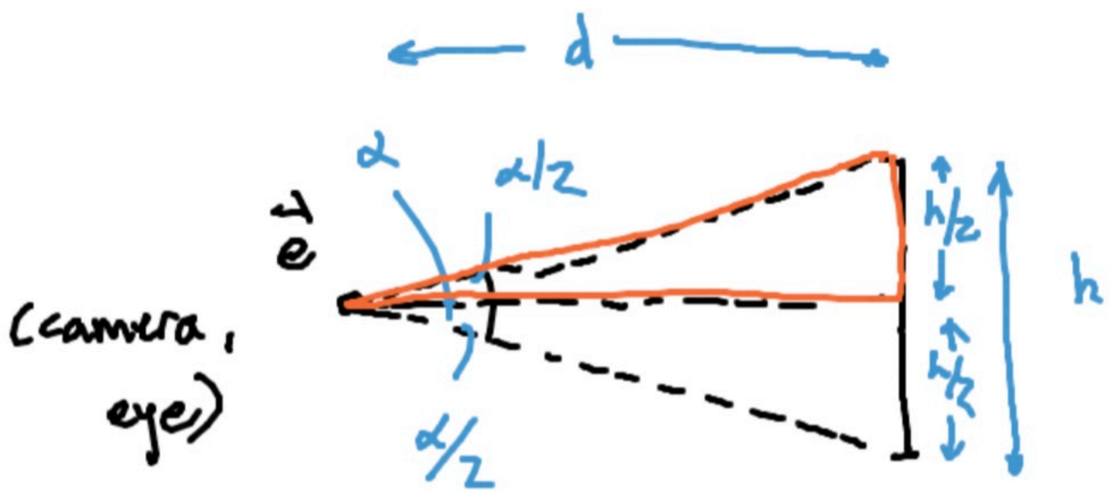
1. Set up your image and place your observer (camera/eye).
2. For each pixel in your image:
    a. Create a ray originating at your camera position that passes through this pixel.
    b. Determine the closest object in your scene intersected by the ray.
    c. Determine the color of the pixel, based on the intersection.

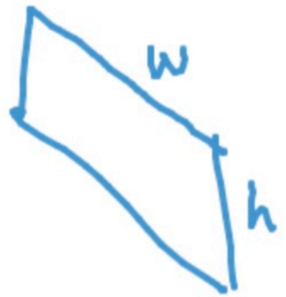


Step 1: setting up a camera and image plane.

position
fov, α
direction

What are the dimensions of the image plane in 3d space?





$$\frac{h/2}{d} = \tan(\alpha/2)$$

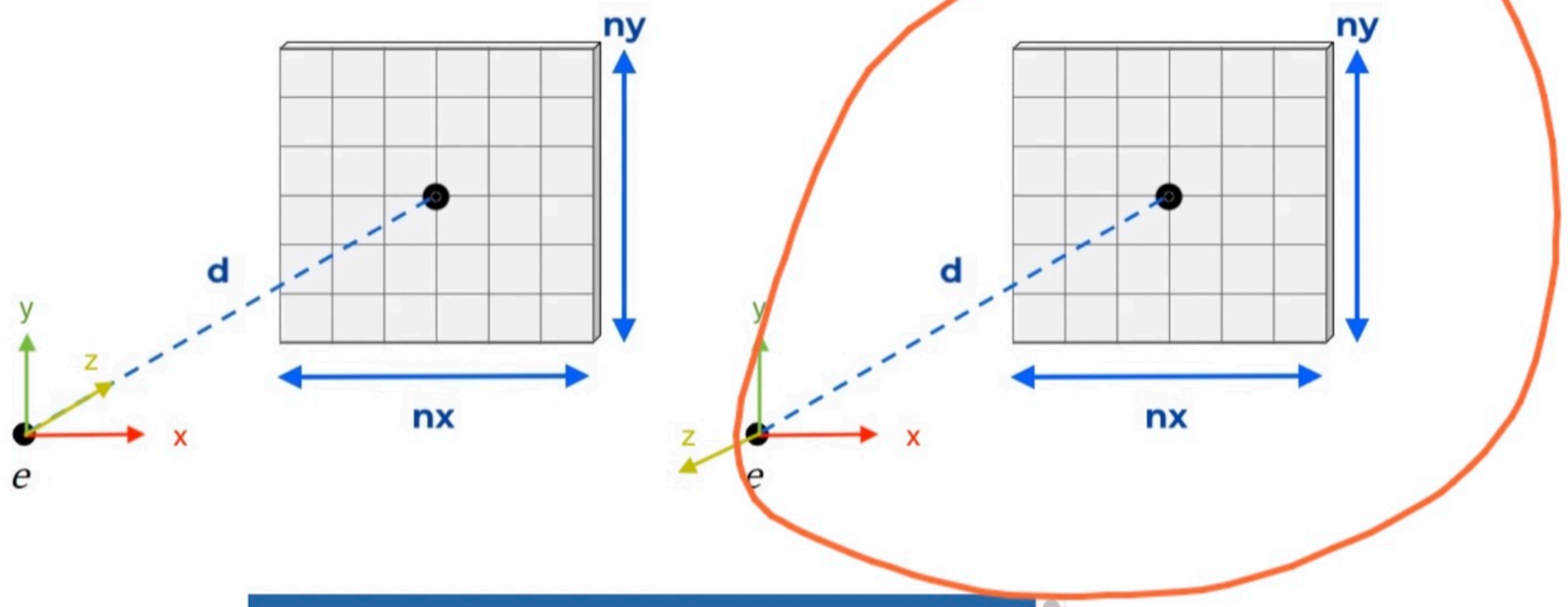
$$h = 2d\tan(\alpha/2)$$

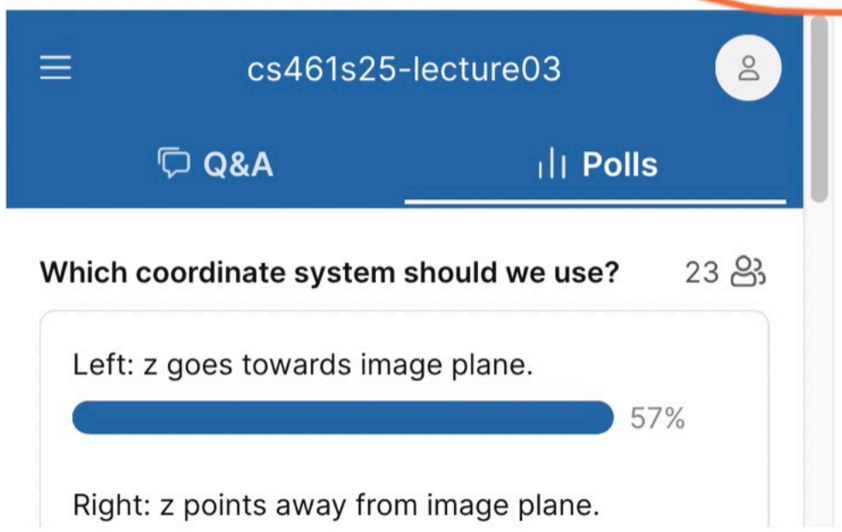
$$w = 2d(AR)\tan(\alpha/2) = (AR)h.$$

$$AR = \frac{nx}{ny} = \frac{w}{h}$$

$$w = h(AR)$$

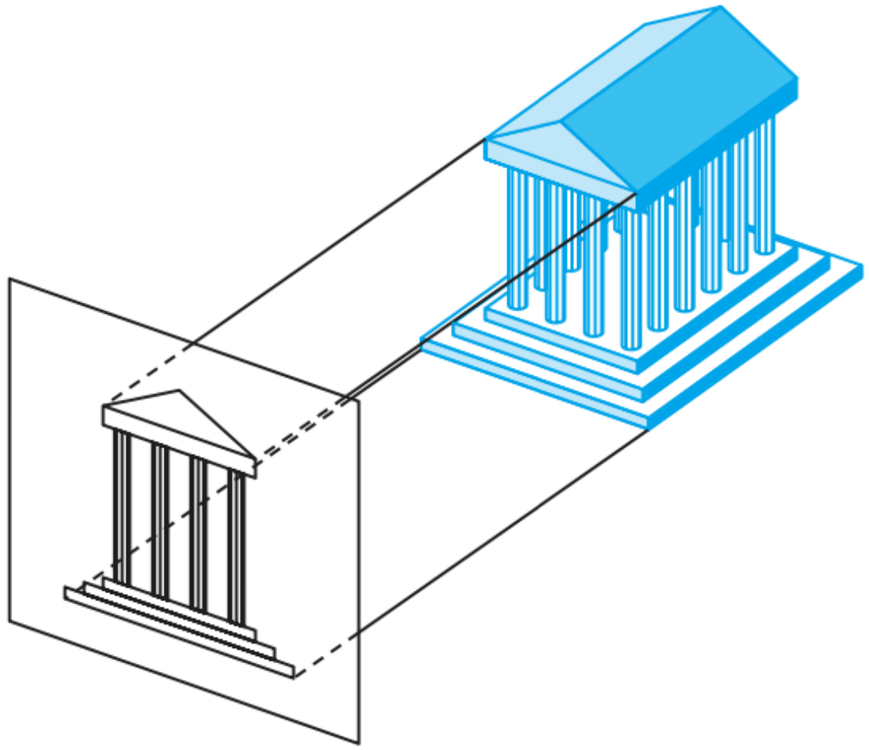
Which coordinate system should we use? (slido.com # 3952613)



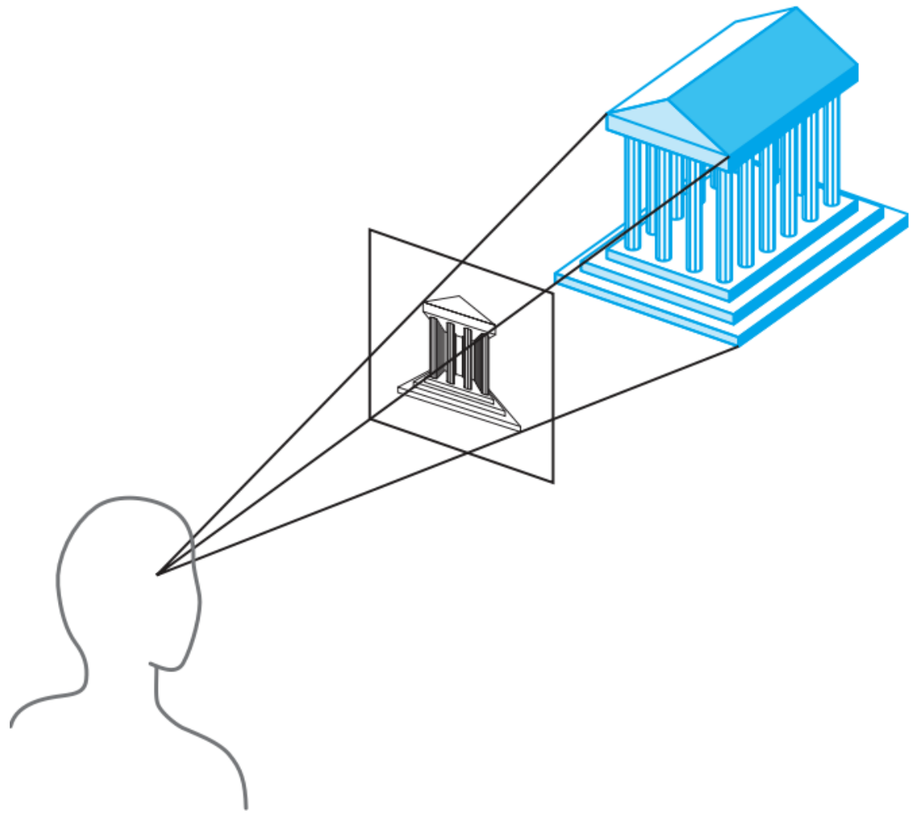


For a perspective view, rays will start at eye and pass through each pixel.

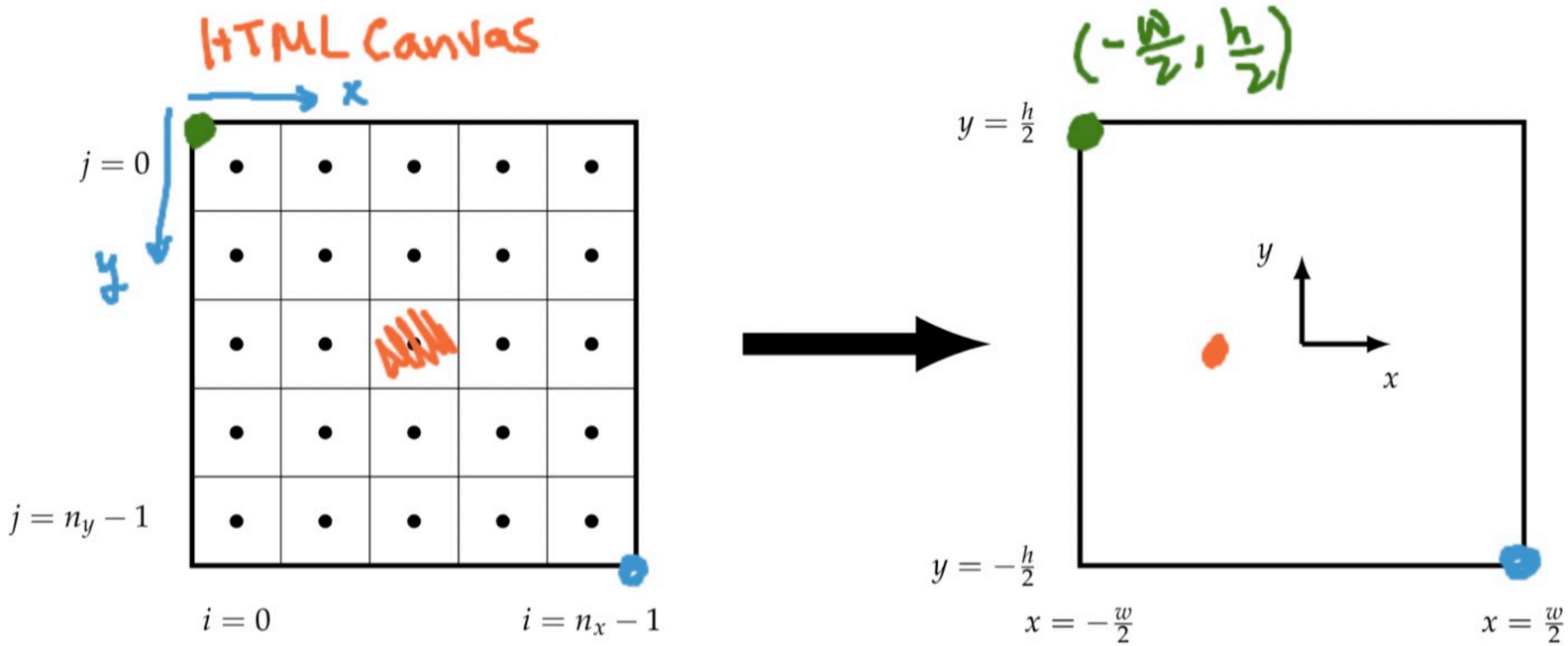
orthographic



perspective



Step 2a: Calculating the 3d coordinates of a pixel.



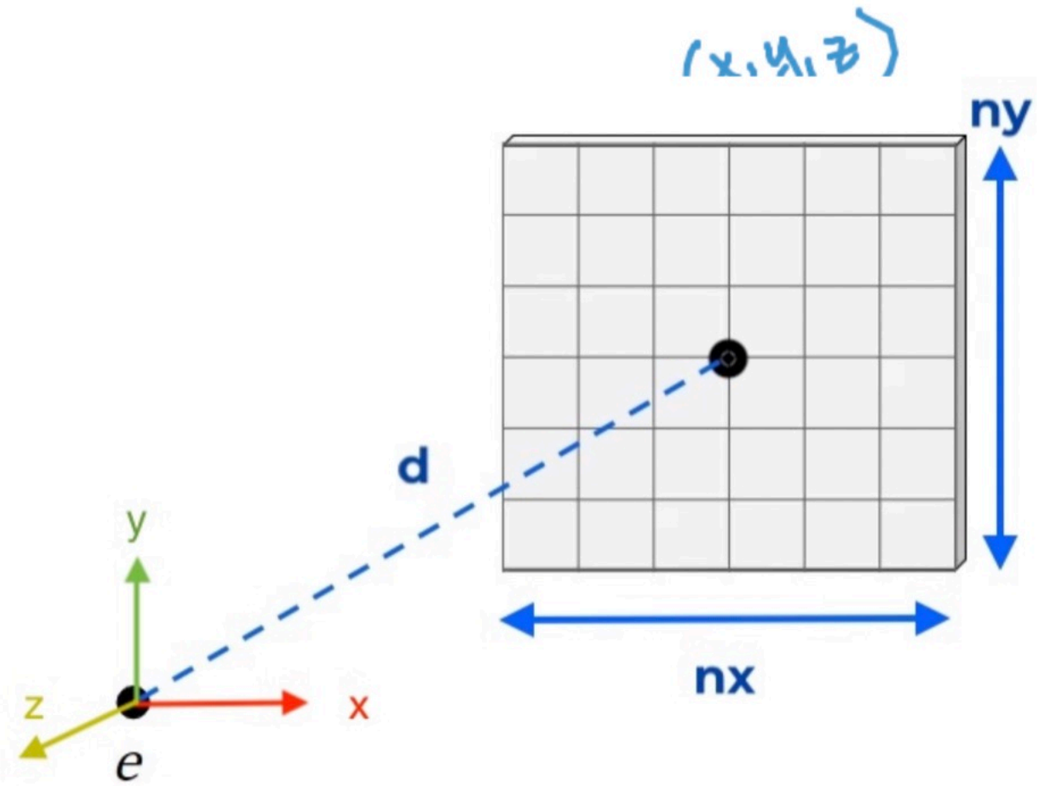
$$x = -\frac{w}{2} + w\frac{(i+0.5)}{n_x} \qquad y = -\frac{h}{2} + h\frac{(n_y - 0.5 - j)}{n_y}$$

approximate check:

- $(i,j) = (0,0) \quad (x,y) = \left(-\frac{w}{2}, -\frac{h}{2} + h\right) = \left(-\frac{w}{2}, \frac{h}{2}\right)$ ✓
- $(i,j) = (n_x, n_y) \quad (x,y) = \left(-\frac{w}{2} + w, -\frac{h}{2} + h(0)\right) = \left(\frac{w}{2}, -\frac{h}{2}\right)$

(approximate) — $-\frac{w}{2} + w = w/2$

What is the z-coordinate of each pixel in our coordinate system?



cs461s25-lecture03

Q&A

Polls

What is the z-component of each pixel in our coordinate system?

26

-d

85%

+d

15%

0

0%

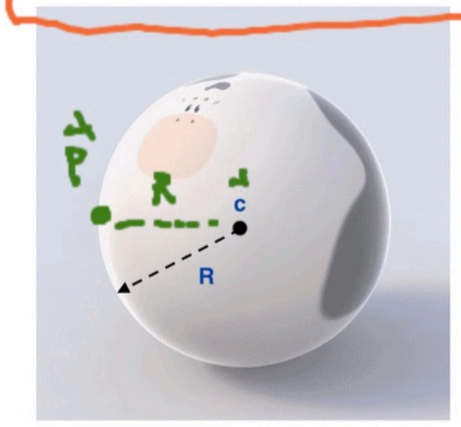
Step 2b: Calculating the intersection of a ray with a sphere.

Why? because lots of things can be approximated as spheres!

Calculating the intersection of a ray with a sphere.

$\vec{r}$ is a unit vector.

ray: (line) $\vec{x}(t) = \vec{e} + t\vec{r}$



A = 1

sphere: $\|\vec{p} - \vec{c}\|^2 = R^2$

$\vec{p} = \vec{x} = \vec{e} + t\vec{r}$

$\|\vec{u}\|^2 = \vec{u} \cdot \vec{u}$

$(\vec{p} - \vec{c}) \cdot (\vec{p} - \vec{c}) = R^2$

$(\underbrace{\vec{e} + t\vec{r}}_{\vec{p}} - \vec{c}) \cdot (\vec{e} + t\vec{r} - \vec{c}) = ((\vec{e} - \vec{c}) + t\vec{r}) \cdot ((\vec{e} - \vec{c}) + t\vec{r})$

$= \|(\vec{e} - \vec{c})\|^2 + [t(\vec{e} - \vec{c}) \cdot \vec{r}]2 + t^2\|\vec{r}\|^2 = R^2$

$t^2 \underbrace{\|\vec{r}\|^2}_{A} + 2t\underbrace{(\vec{e} - \vec{c}) \cdot \vec{r}}_{B} + \underbrace{[\|\vec{e} - \vec{c}\|^2 - R^2]}_{C} = 0$

$At^2 + 2tB + C = 0$ quadratic eq'n

$t = \frac{-2B \pm \sqrt{(2B)^2 - 4AC}}{2A}$

$= \frac{-B \pm \sqrt{B^2 - AC}}{A}$

$= \frac{-B \pm \sqrt{B^2 - C}}{1}$

$4B^2 - 4AC$

Calculating the intersection of a ray with a sphere.

$$t = -B \pm \sqrt{B^2 - C}$$

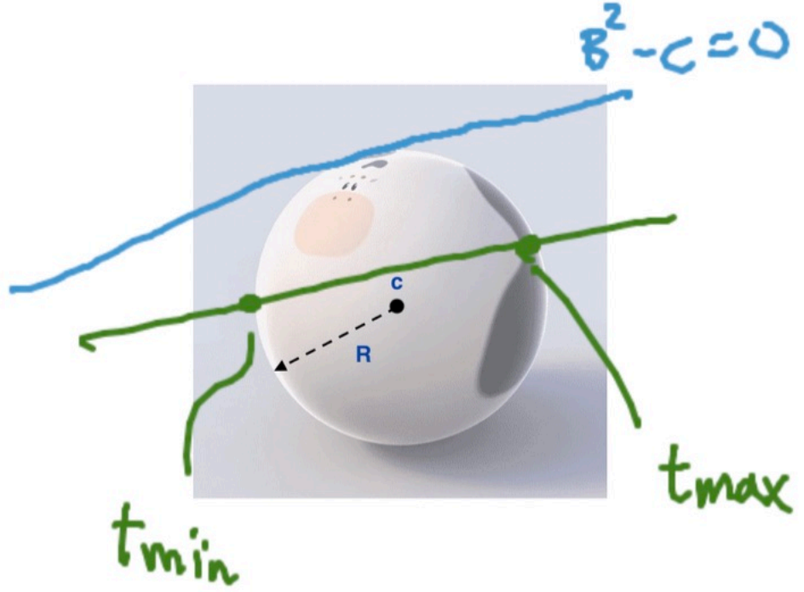
$B^2 - C < 0$? no intersection

$B^2 - C = 0$

$B^2 - C > 0$ $t_{max} = -B + \sqrt{B^2 - C}$

$t_{min} = -B - \sqrt{B^2 - C}$

for now $\vec{x} = \vec{e} + t\, t_{min} \vec{r}$



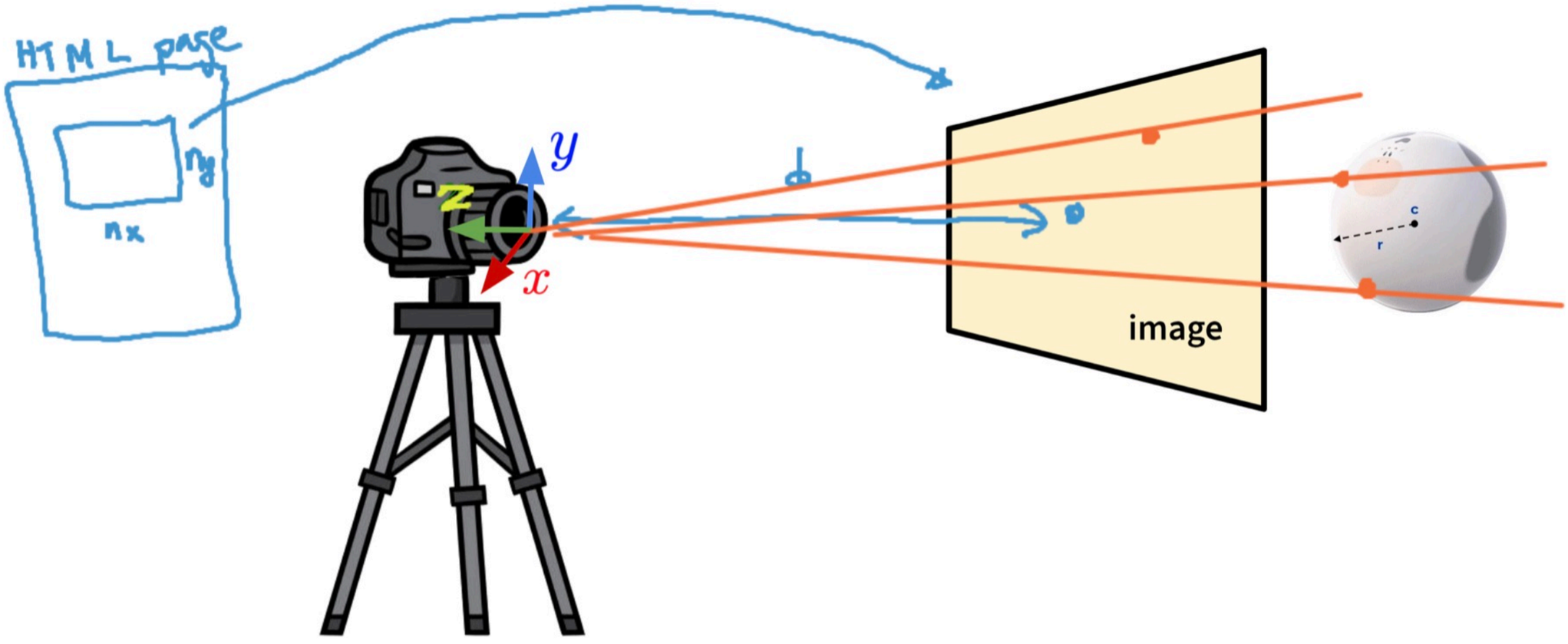
Implementing our own ray tracer! (and more practice with `glMatrix`)

Please download the `exercises.zip` file in the row for today's class (on calendar).

- Part 1: calculate image plane dimensions (w, h).
- Part 2: calculate 3d coordinates of pixel.
- Part 3a: complete `Sphere intersect` function.
- Part 3b: create `Ray` object and assign pixel color.

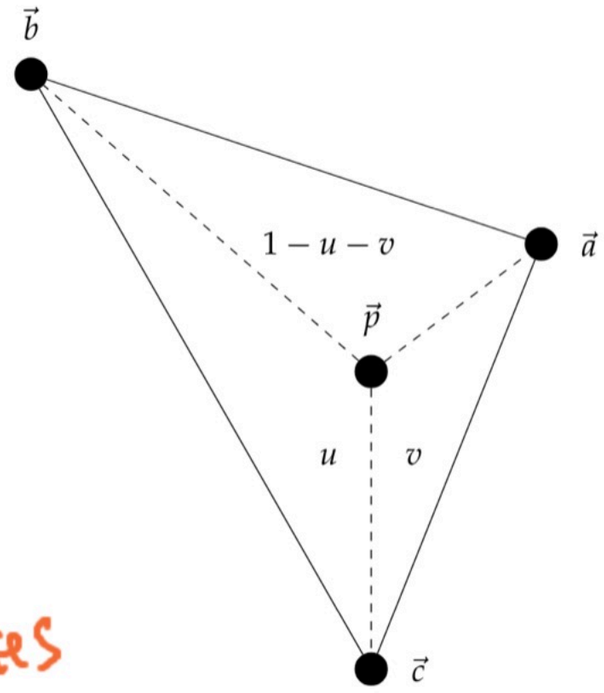
Summary of where we left off on Tuesday.

- We are currently assuming the eye (camera) is at the origin (more general setups in a few weeks).
- Define FOV (α) and calculate image plane dimensions (w , h).
- For each pixel: calculate its 3d coordinates (x, y, z), create ray, intersect with objects, assign color. $z = -d$



Step 2b: Calculating the intersection of a ray with a triangle.

Why? because most things are not spheres :(

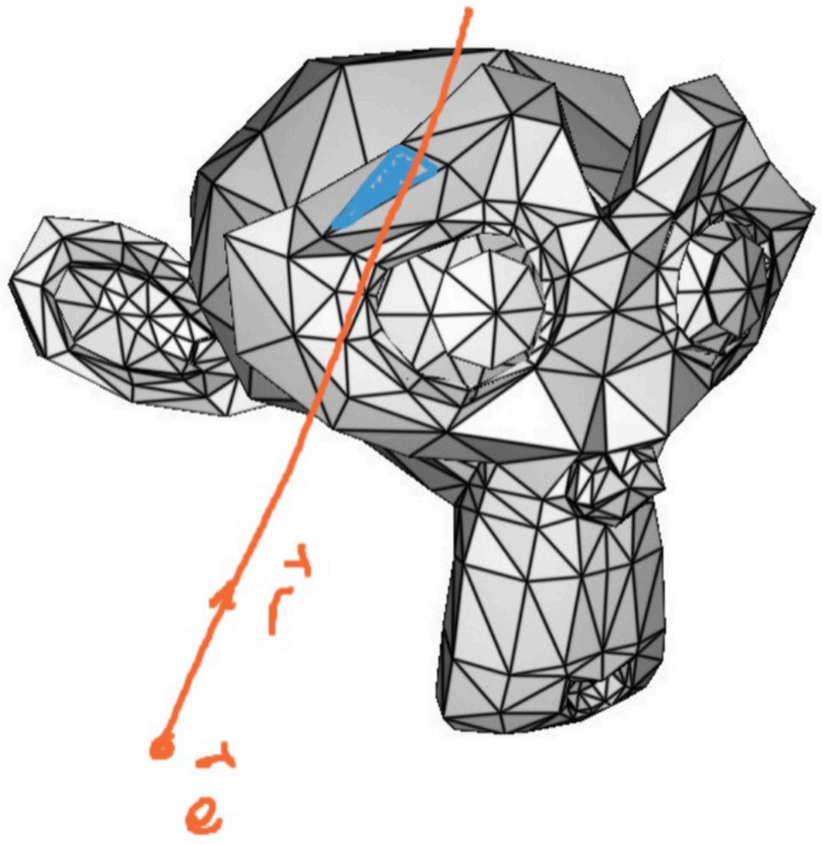


barycentric coordinates

$w = 1-u-v$

$$\vec{p} = u\vec{a} + v\vec{b} + w\vec{c}$$

$$= u\vec{a} + v\vec{b} + (1-u-v)\vec{c}$$



Calculating the intersection of a ray with a triangle.

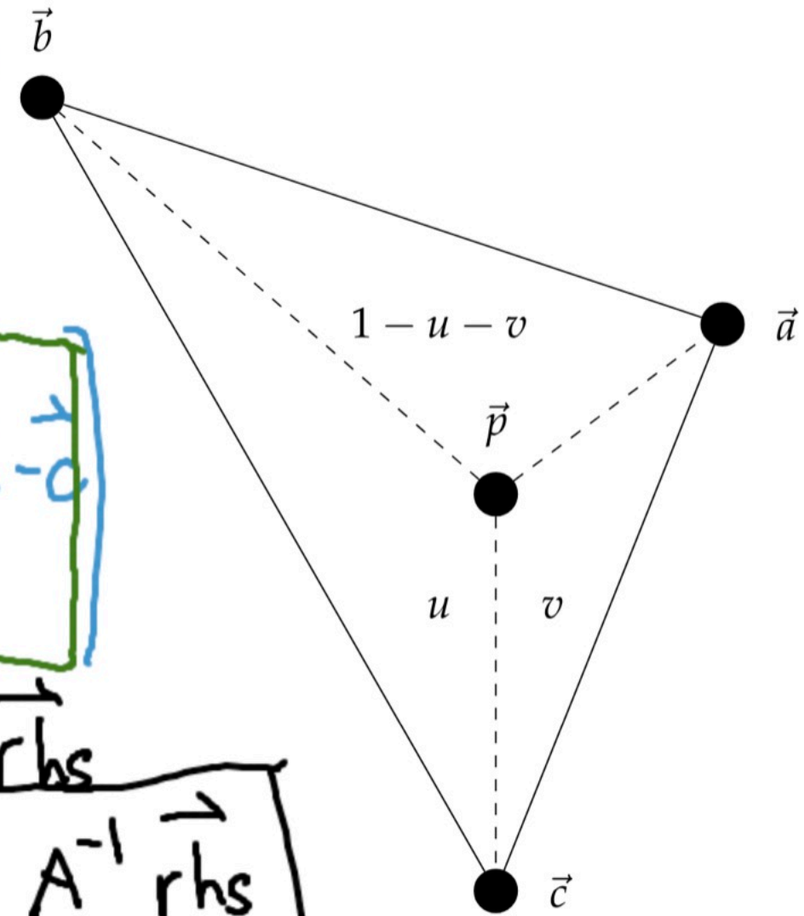
$$\vec{p}(u,v) = \vec{x}(t) \quad \rightarrow \quad \boxed{\vec{a}u + \vec{b}v + (1-u-v)\vec{c} = \vec{e} + \vec{r}\,t.}$$

$$(\vec{a}-\vec{c})\,u + (\vec{b}-\vec{c})\,v + (-\vec{r})\,t = \vec{e}-\vec{c}$$

$$\begin{bmatrix} | & | & | \\ (\vec{a}-\vec{c}) & \vec{b}-\vec{c} & -\vec{r} \\ | & | & | \end{bmatrix} \begin{bmatrix} u \\ v \\ t \end{bmatrix} = \begin{bmatrix} | \\ \vec{e}-\vec{c} \\ | \end{bmatrix}$$

A $\qquad$ $\vec{s}$ $\qquad$ $\vec{rhs}$

$$\vec{s} = A^{-1}\,\vec{rhs}$$



check:

$0 \le u \le 1$

$0 \le v \le 1$

$0 \le 1-u-v \le 1$

if all 3 satisfied then intersection within triangle

Summary

- Calculate dimensions of image plane (w and h) from field-of-view (α) and image aspect ratio.
- Calculate 3d coordinates of each pixel and direction of ray from eye to pixel.
- For now, we are always assuming the eye $\vec{e}$ is at the origin, and that we are looking in the $-z$ direction.
- Intersect ray with objects in scene (spheres, triangles, etc.).
- **Use closest intersection point:** smallest t value.
- **What's missing?** looks 2d :(shading next week!
- I will be at a conference on Tuesday 3/4, so I'll record a lecture and share it early next week. Please watch the video before Lab 3 on Thursday 3/6.