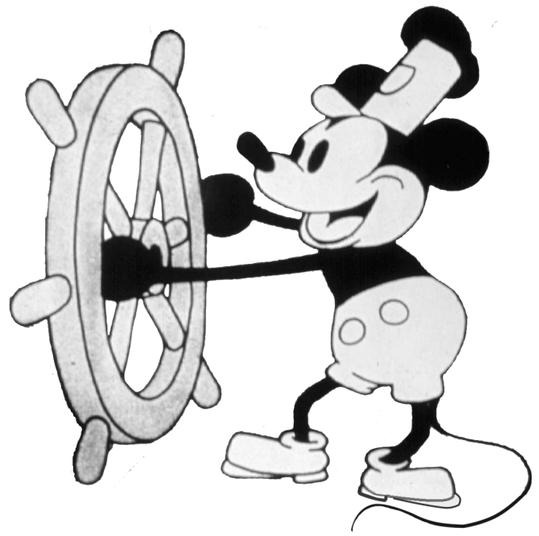


CSCI 461: Computer Graphics

Middlebury College, Fall 2025

Lecture 03: Ray Tracing

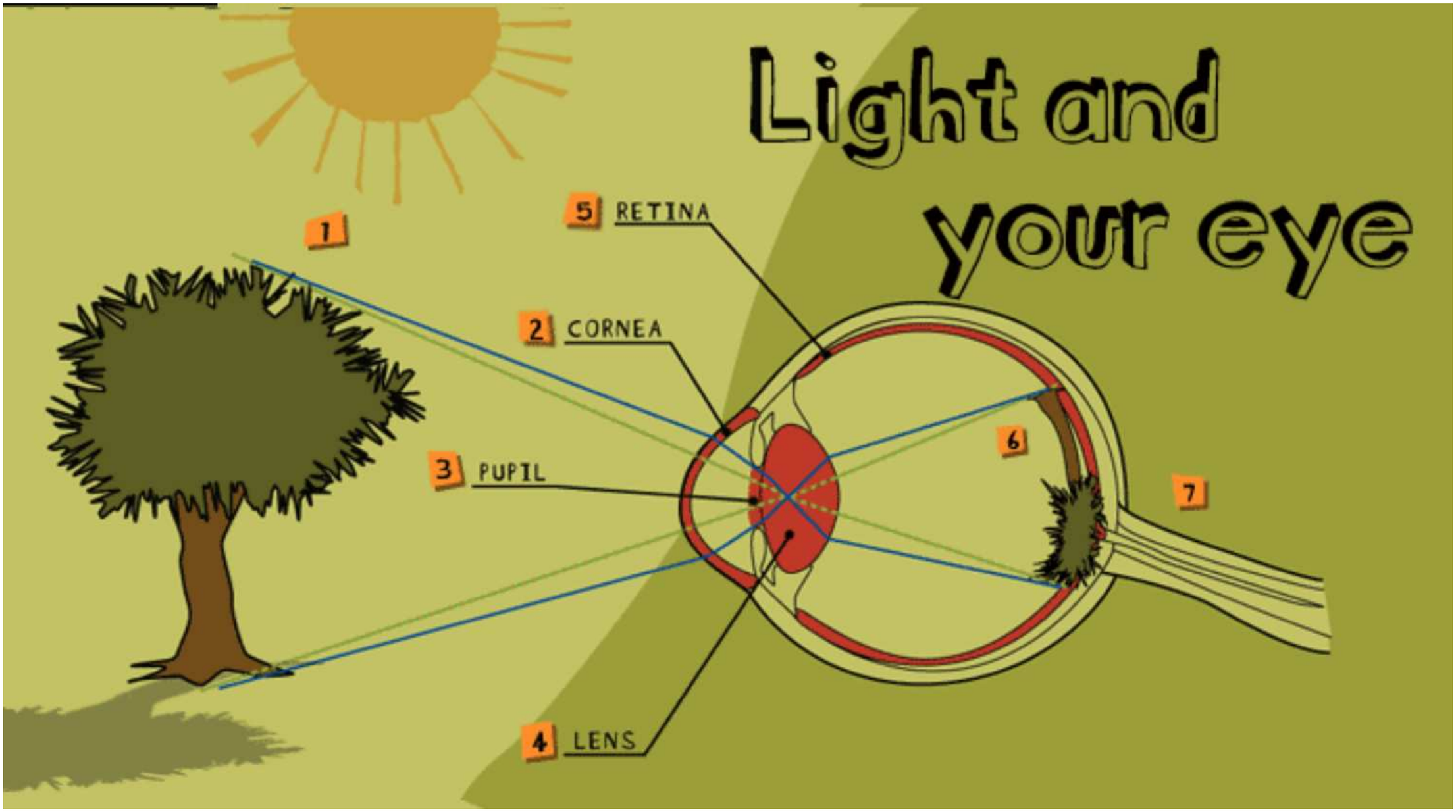
By the end of today's lecture, you will be able to:

- 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?

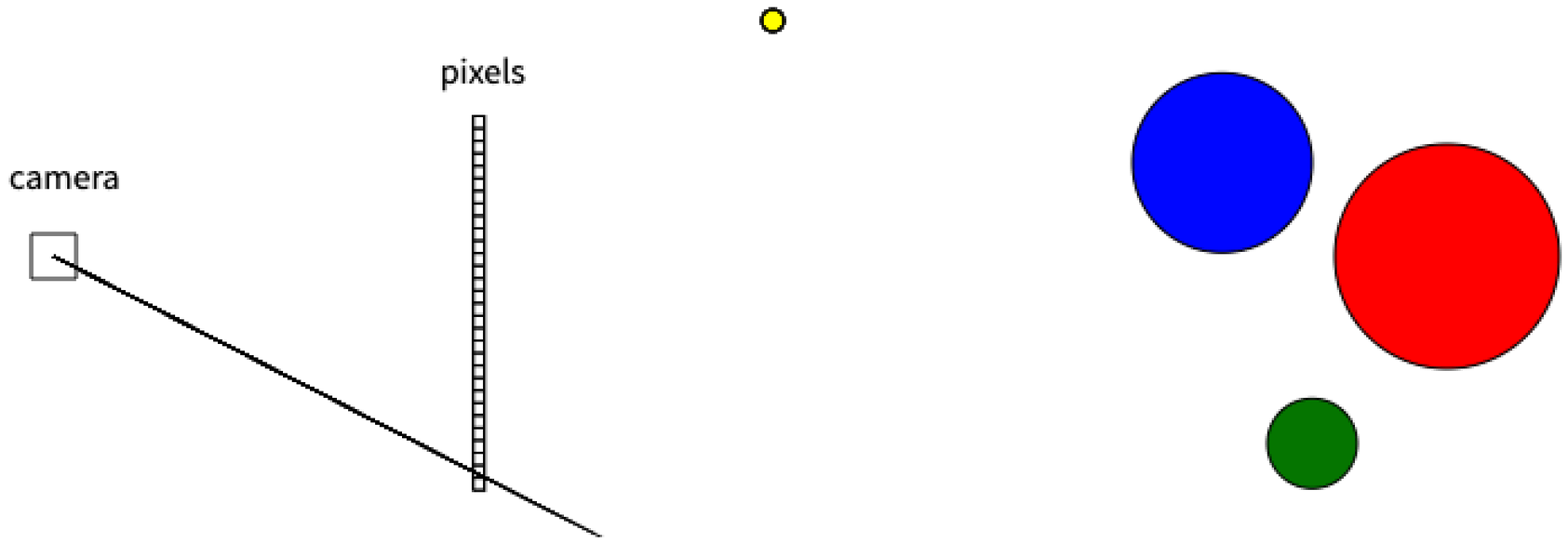
How do we see things?



(source)

The main idea of ray tracing.

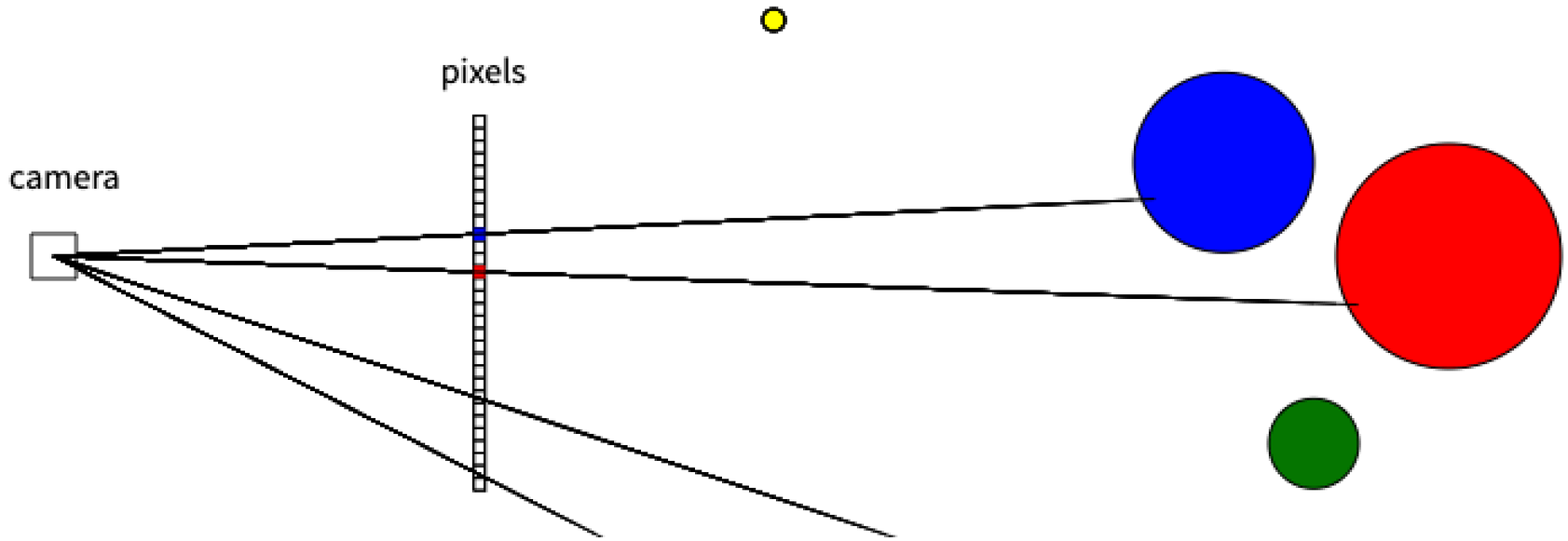
Send "rays" from an observation point (e.g. a camera) through each pixel in an image and into the (virtual) 3d world. Whatever is hit by the ray defines the color of the pixel.



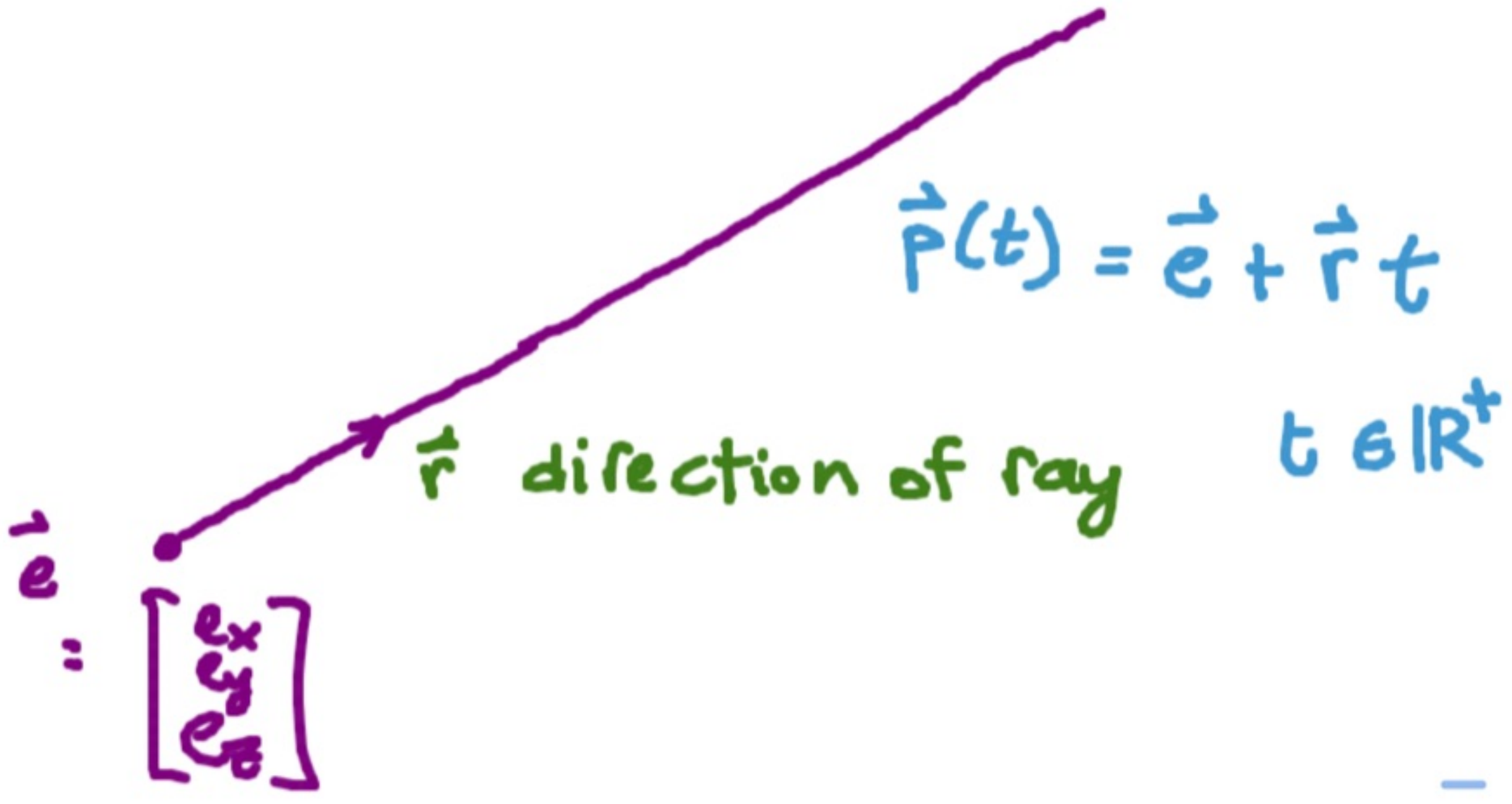
Click to open the ray tracing demo.

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.



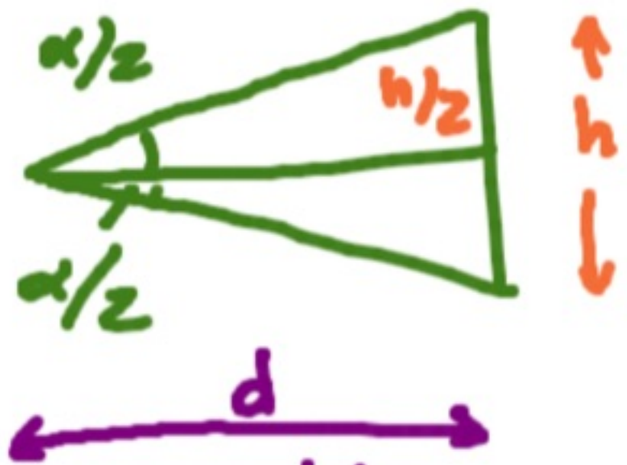
First things first: we need to represent rays mathematically.



Step 1: setting up a camera and image plane.

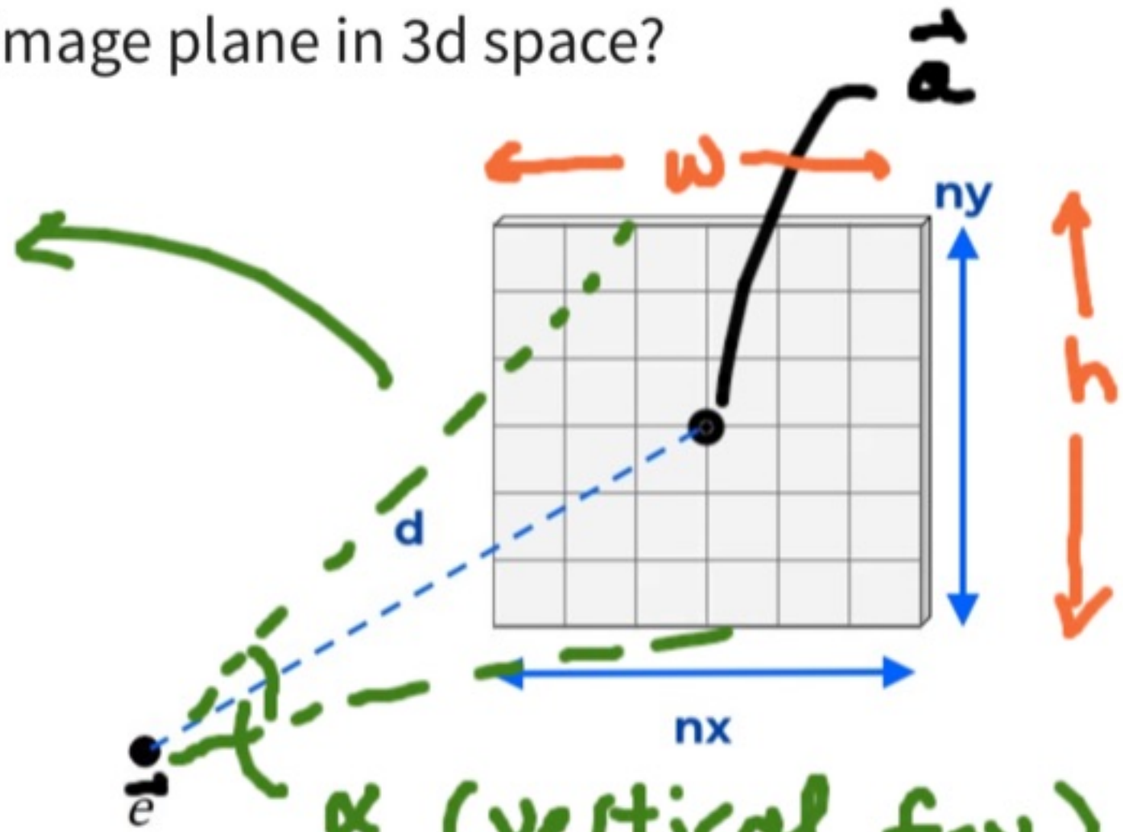
$$d = \|\vec{a} - \vec{e}\|$$

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



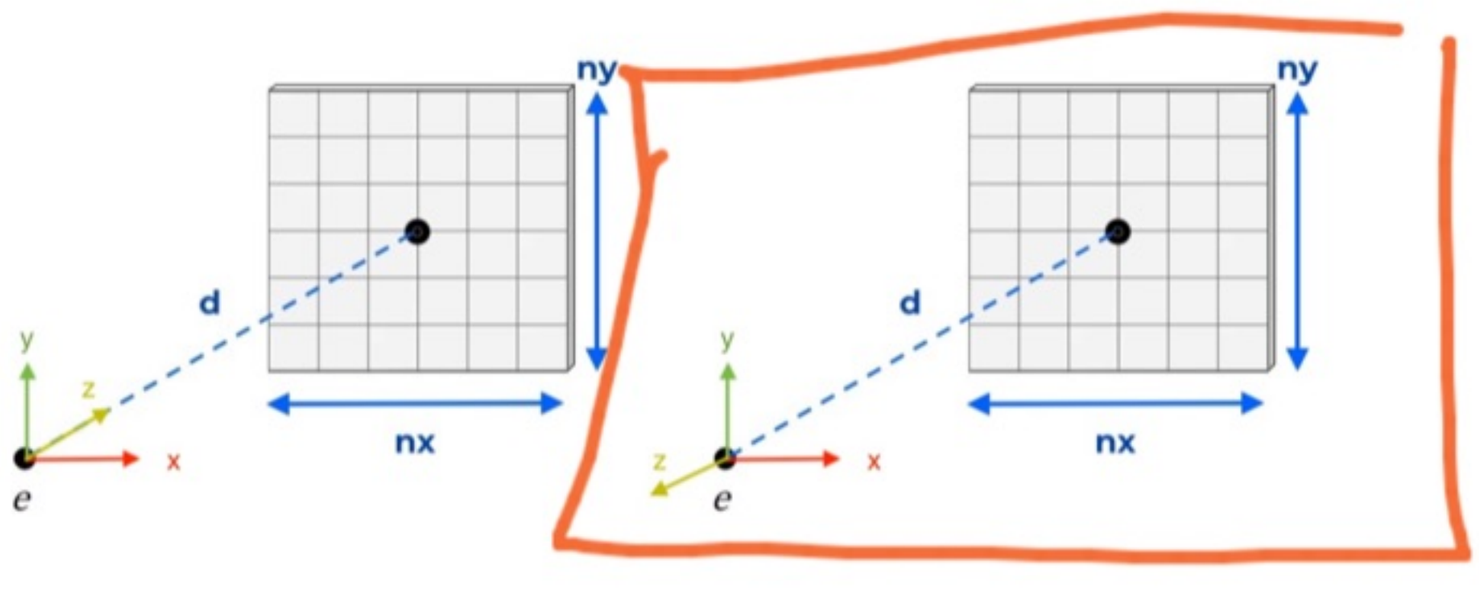
$$\tan(\alpha/2) = \frac{h/2}{d}$$

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



w ?? $AR = \frac{w}{h} = \frac{n_x}{n_y}$ → $\boxed{w = h\left(\frac{n_x}{n_y}\right)}$

Which coordinate system should we use?



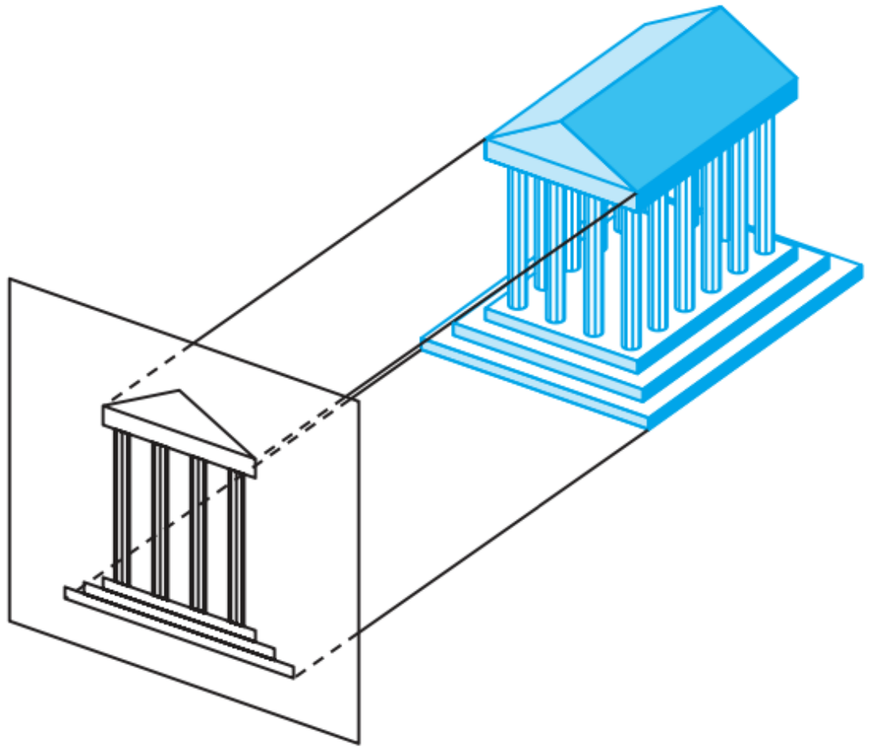
Vote using course **Reactions** page:

- A: coordinate system on the left.
- B: coordinate system on the right.

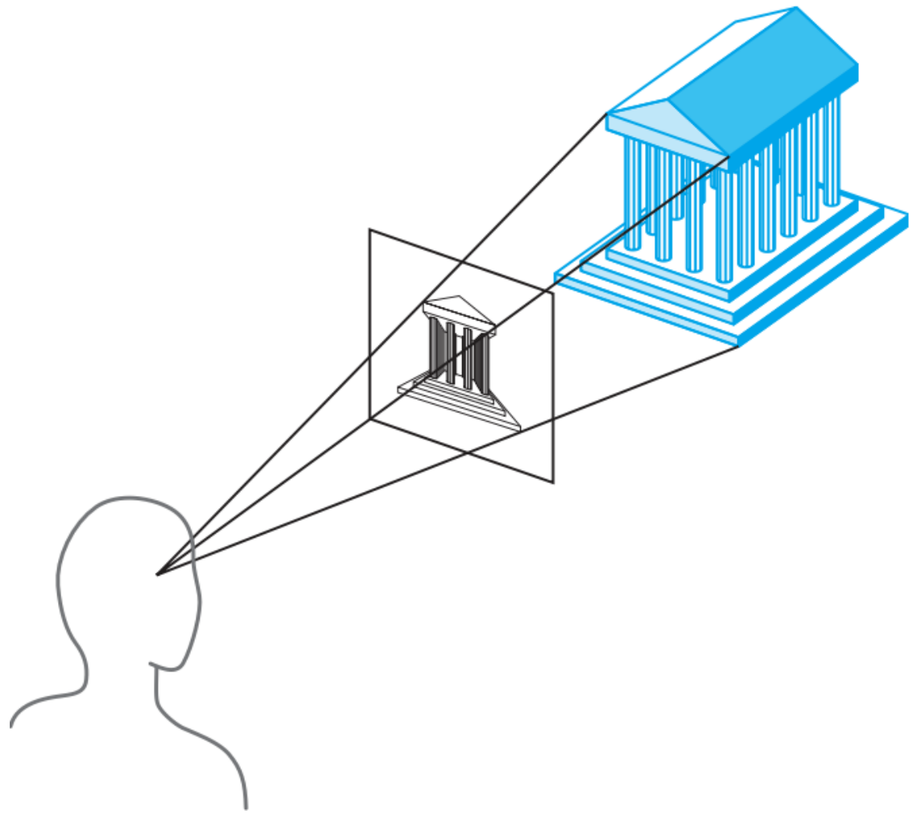
turtle
bird

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

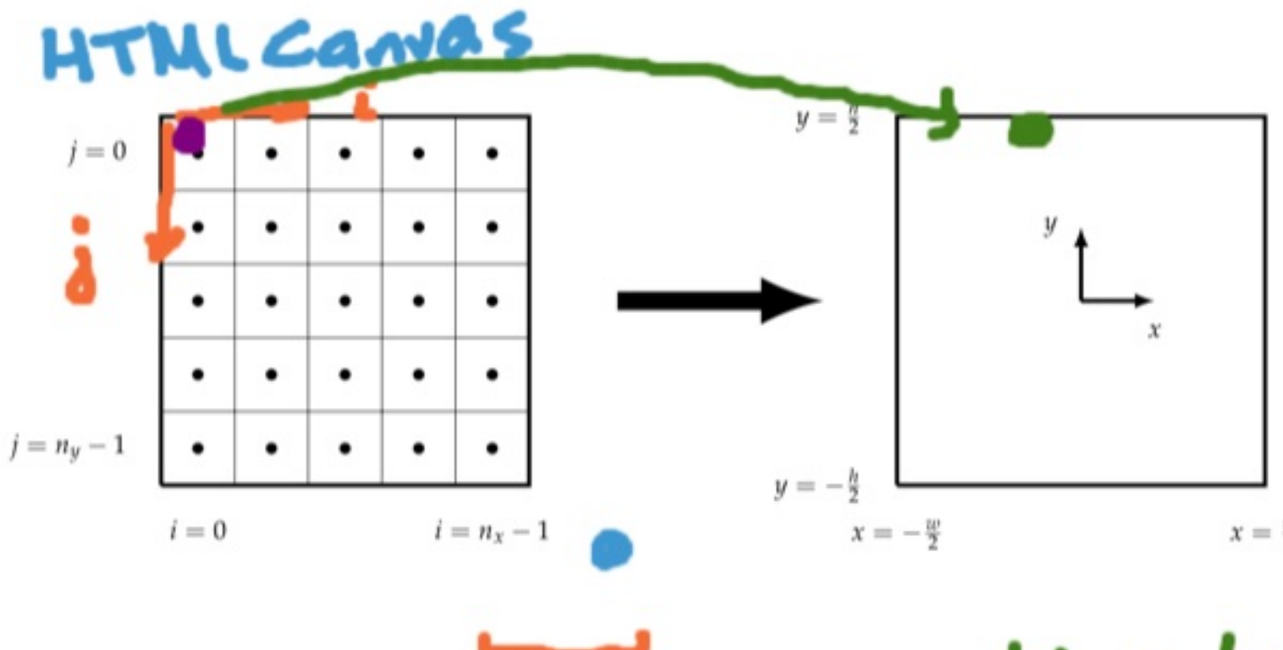
orthographic



perspective



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



$-h/2 + h = +h/2$

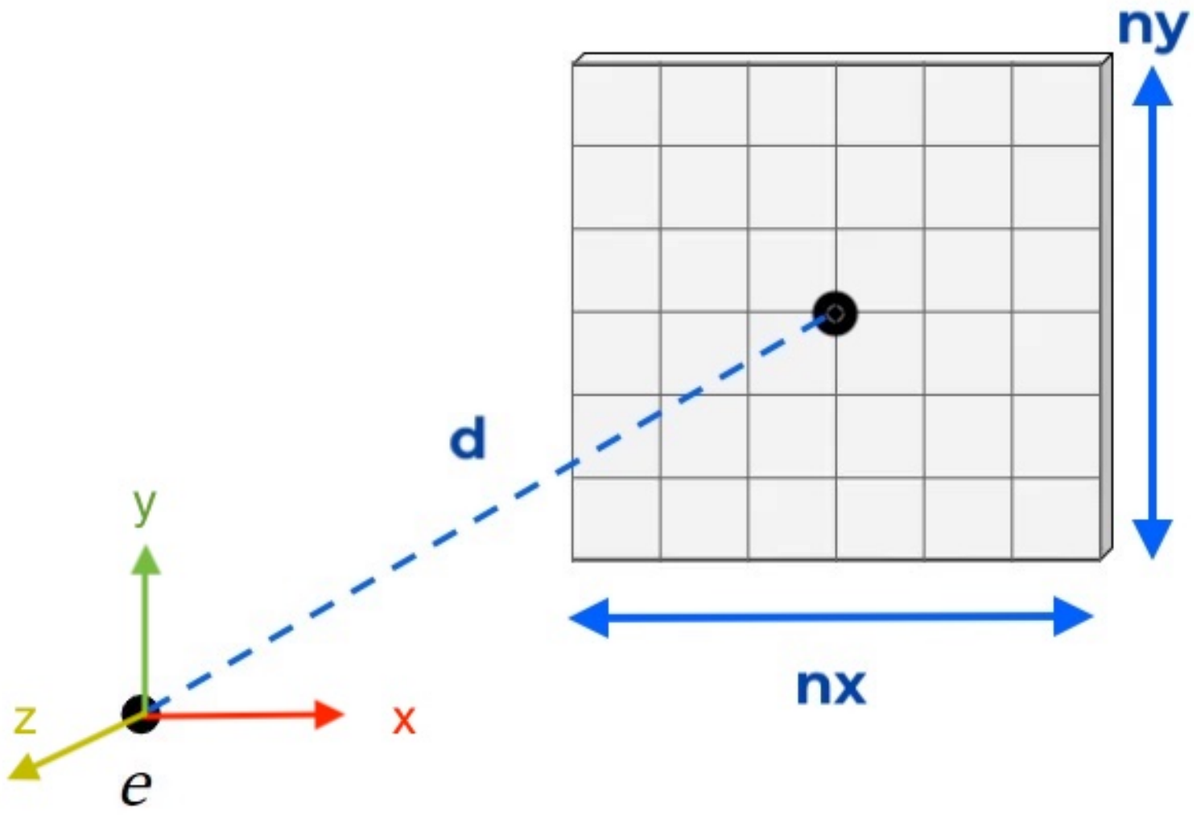
$$x = -\frac{w}{2} + w\frac{(i + 0.5)}{n_x} \qquad y = -\frac{h}{2} + h\frac{(n_y - 0.5 - j)}{n_y}$$

drop 0.5 for simplicity

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

$(i,j) = (n_x, n_y) \rightarrow (x,y) = \left(\frac{w}{2}, -h/2\right)$

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



Vote using course **Reactions** page:

- **A**: $-d$
- **B**: 0
- **C**: $+d$

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.

sphere: all points are the same distance to center

$= R$

$\|\vec{p} - \vec{c}\|^2 = R^2 \qquad \|\vec{u}\|^2 = \vec{u} \cdot \vec{u}$

plug into

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

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

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

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

$A = \vec{r} \cdot \vec{r}$ $\quad B = (\vec{e} - \vec{c}) \cdot \vec{r}$ $\quad C = \|\vec{e} - \vec{c}\|^2 - R^2$

$\vec{p}(t) = \vec{e} + \vec{r}\,t$

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

$A = \vec{r} \cdot \vec{r} = \|\vec{r}\|^2 = 1$ unit vector

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

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

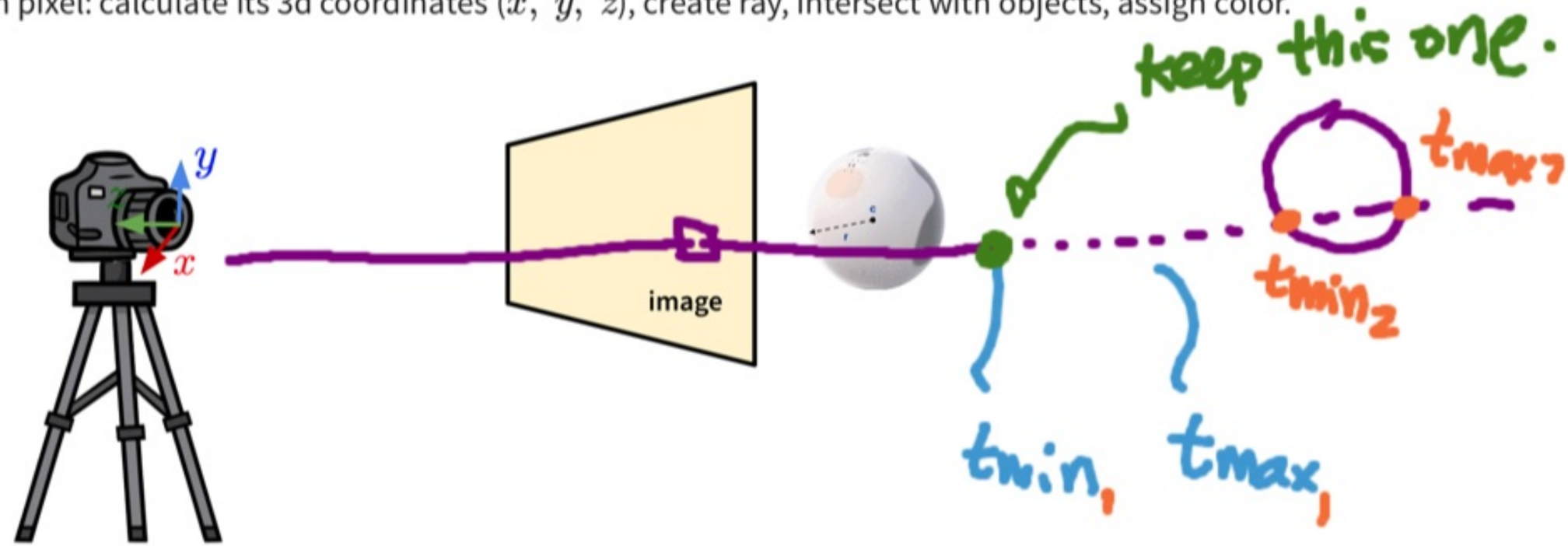
Implementing our own ray tracer! (and more practice with `glMatrix`)

Please open the repository linked in `exercises` 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.

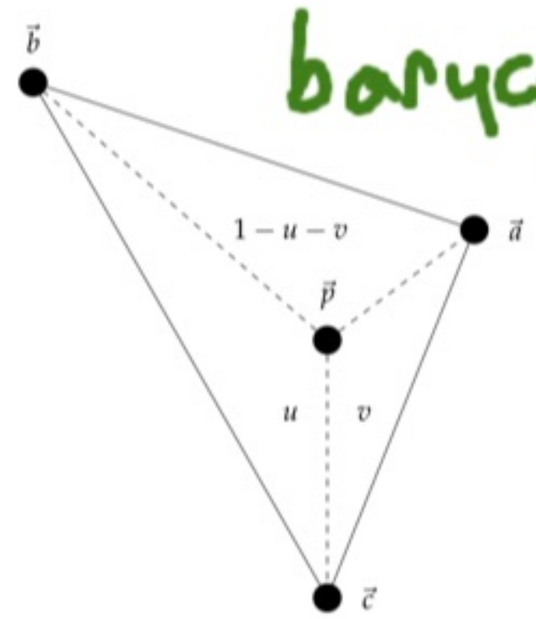
Summary (so far)

- 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.



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

Why? because most things are not spheres :(



barycentric coordinates = $\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$

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

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

Calculating the intersection of a ray with a triangle.

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

unknowns: u, v, t

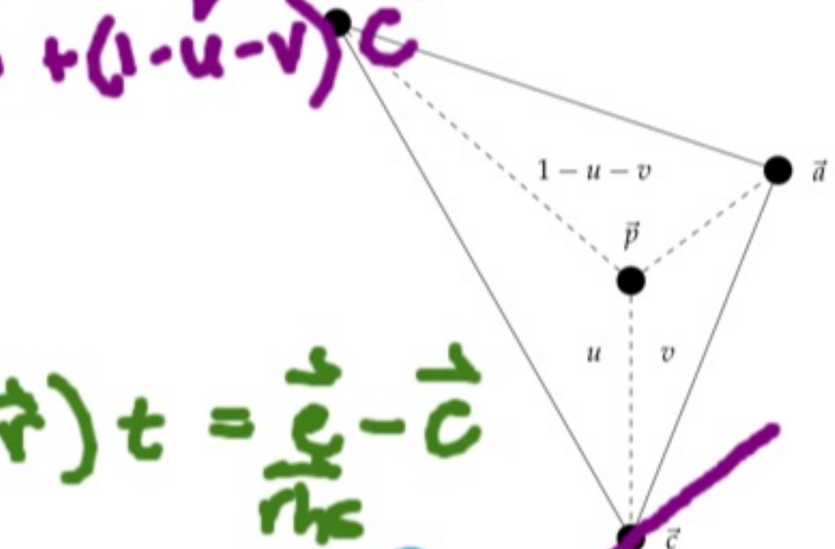
(group into u, v, t)

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

rhs

$$\begin{bmatrix} a_x-c_x & b_x-c_x & -r_x \\ a_y-c_y & b_y-c_y & -r_y \\ a_z-c_z & b_z-c_z & -r_z \end{bmatrix} \begin{bmatrix} u \\ v \\ t \end{bmatrix} = \begin{bmatrix} e_x-c_x \\ e_y-c_y \\ e_z-c_z \end{bmatrix}$$

A $\quad \vec{s}$



$\vec{e}$

then check

$0 \le u \le 1$

$0 \le v \le 0$

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

solve $A\vec{s} = \overrightarrow{rhs}$

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!
- Complete pre-lab 2 before the lab on Thursday!