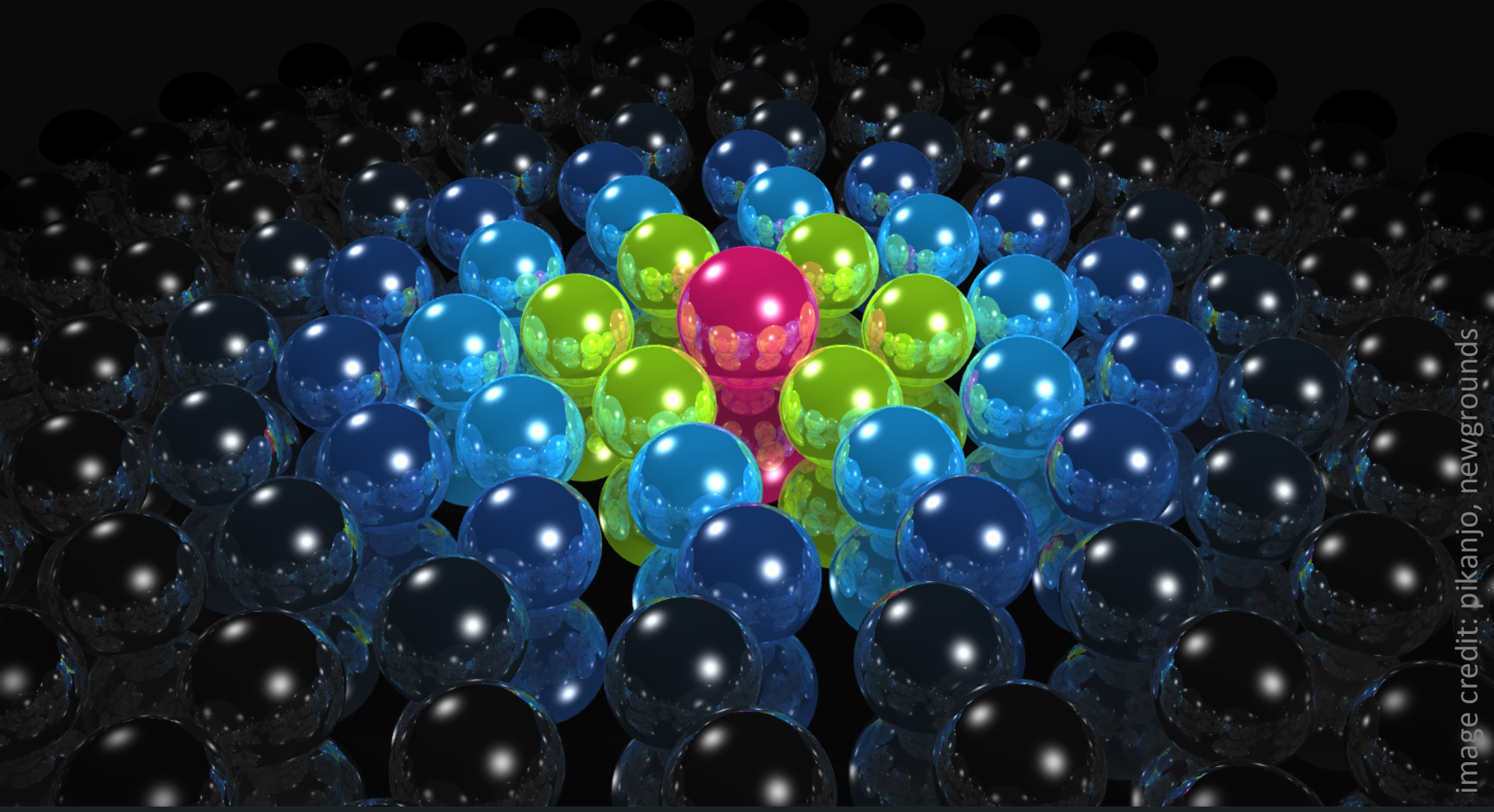


Raytracing

also today: giving a 2 minute talk

CS 4300/5310

Computer Graphics

ANNOUNCEMENTS

Upcoming Deadlines

- 2D Project main deadline: February 5th
  - That's in 5 days!

My office has moved!

- New office: Meserve Hall, Room 146

- Directions
  - Go through unmarked grey door facing Centennial Common
  - Go up a few stairs, left through the door at top
  - Follow signs for game design, along right hallway

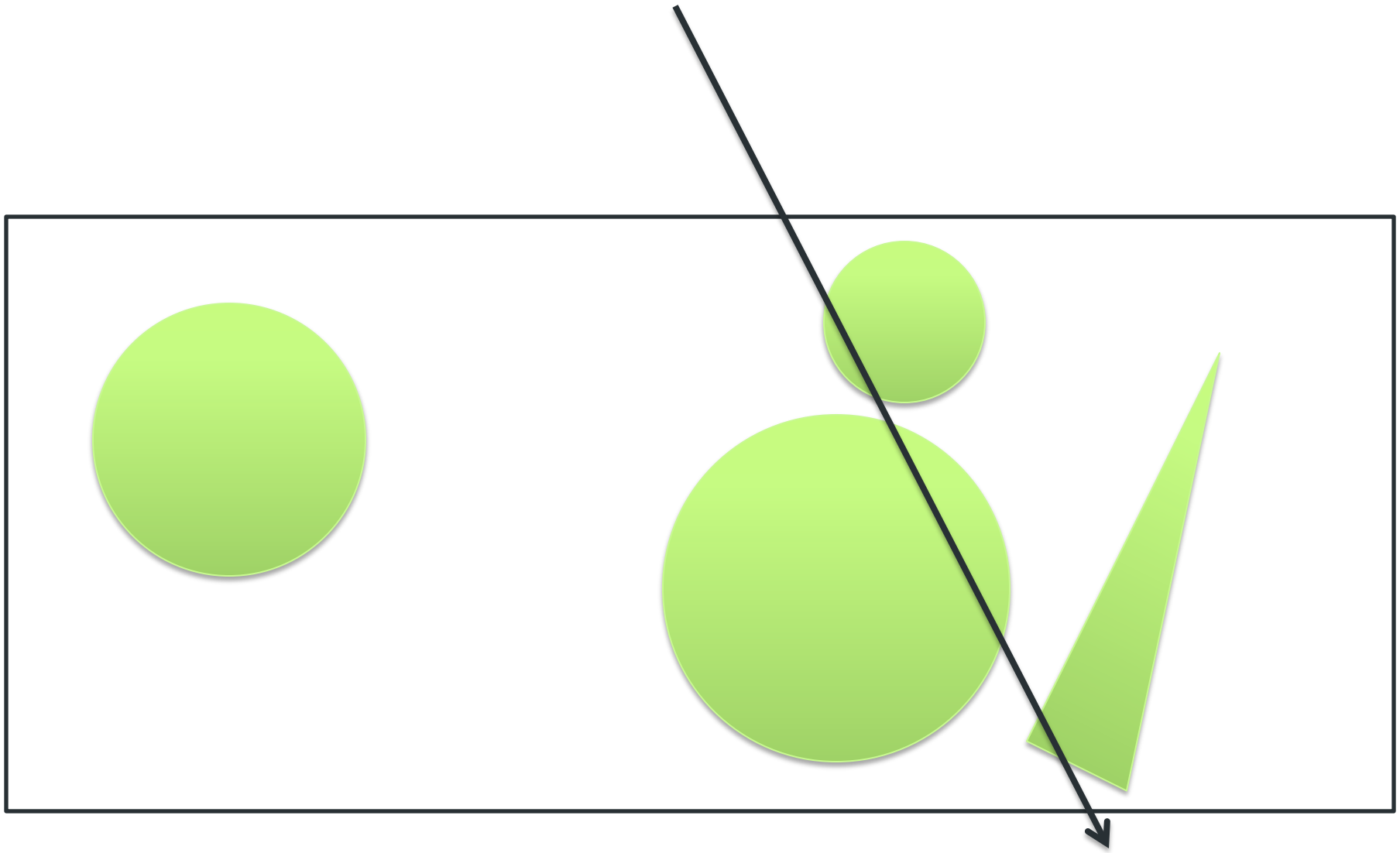
continued from last time…

RAYCASTING

Algorithm Overview

```
for every pixel, p, in the image
        let r be the ray from camera through pixel
        calculate intersection between r and scene objects
        if (!intersection)
                set p to "background color"
        else
                p = color calculated from first object it hits
```

Intersecting with Scene Objects

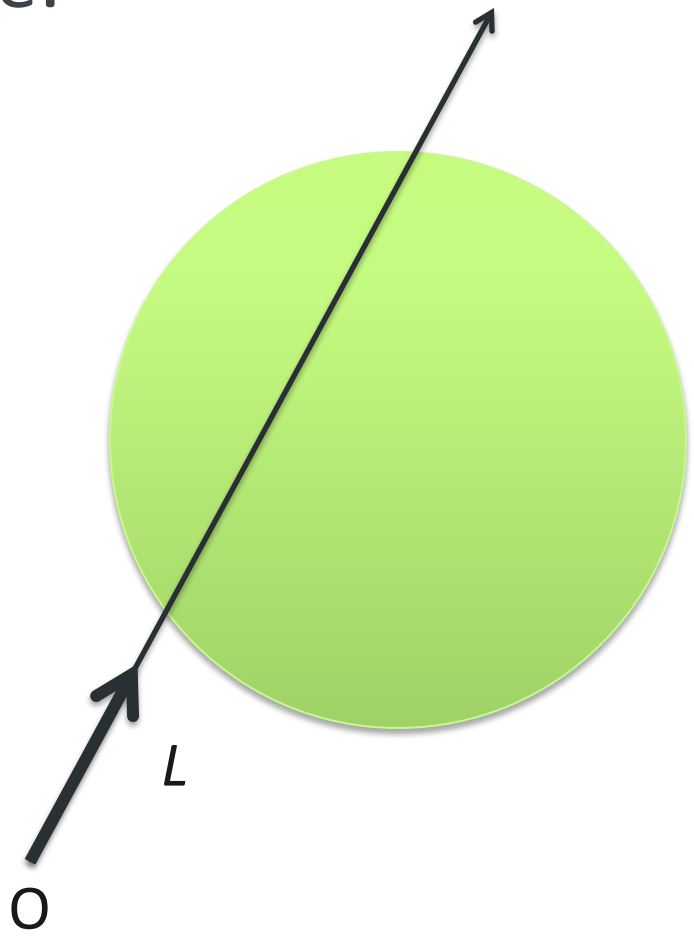


Aside: 3D Picking

- When we click on the screen, what object are we selecting?
  - We can use raycasting to figure this out!

Spheres

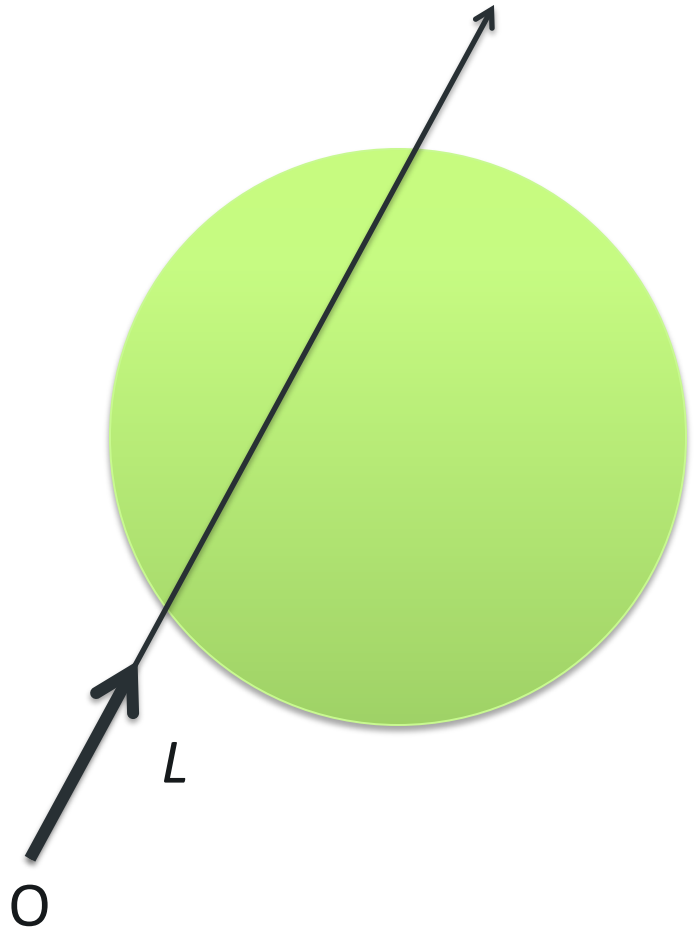
- Representation of a sphere:
  - Center, *C*
  - Radius, *r*
  - $||x - c||^2 = r^2$
- Ray:
  - Origin, *O*
  - Direction, *L*
  - Distance along line, *d*
  - $x = O + dL$



Spheres

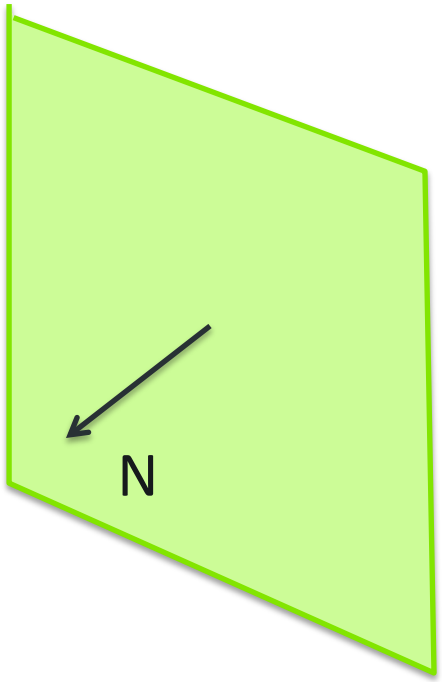
- $||O + dL - C|| ^2 = r^2$

- Solve for *d*….



Planes

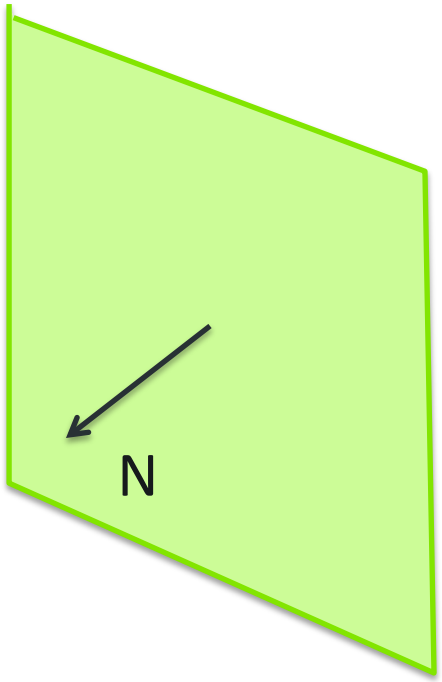
- Plane:
  - $x \cdot N + depth = 0$
- Ray:
  - Origin, O
  - Direction, L
  - Distance along line, d
  - $x = O + dL$



Planes

- $(O + dL) \cdot \mathrm{N} + \mathrm{depth} = 0$

- Solve for d?

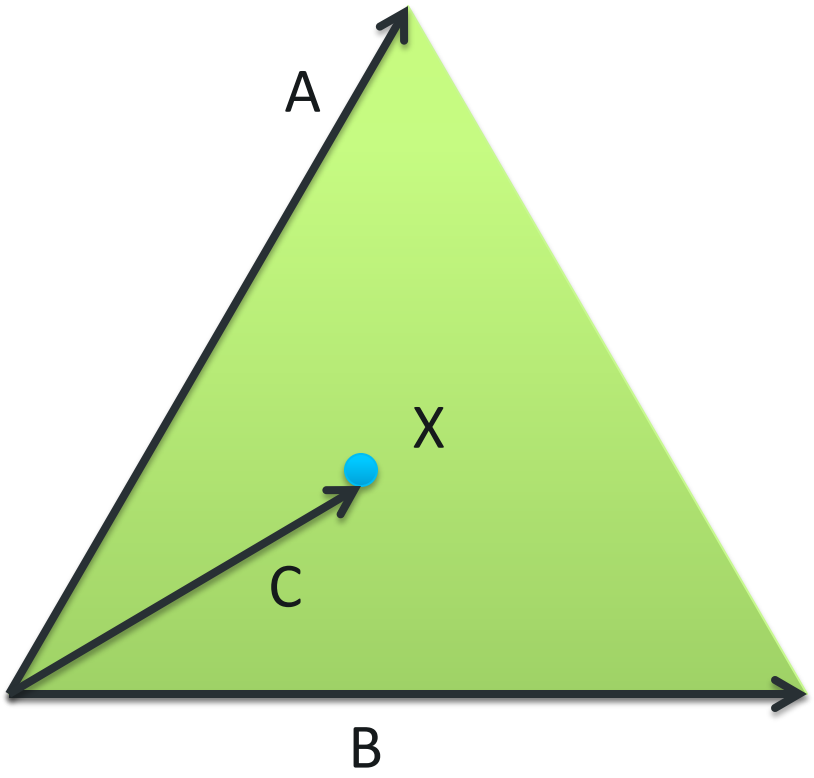


Triangles (Strategy I)

Triangles (Strategy I)

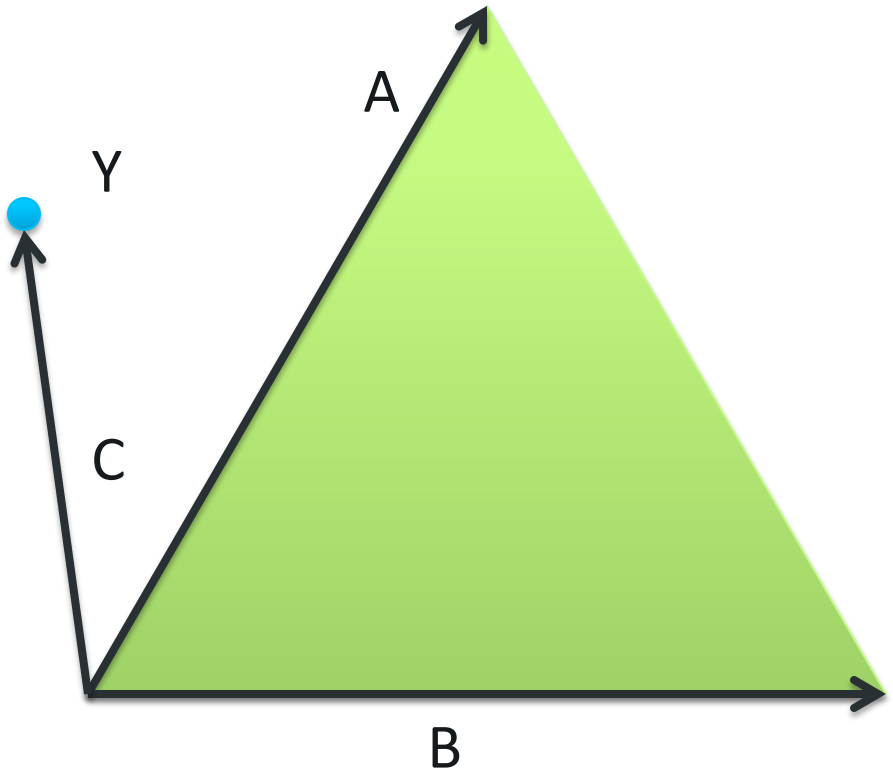
- Check if point on the plane is inside the triangle

- AxC should be same direction as AxB



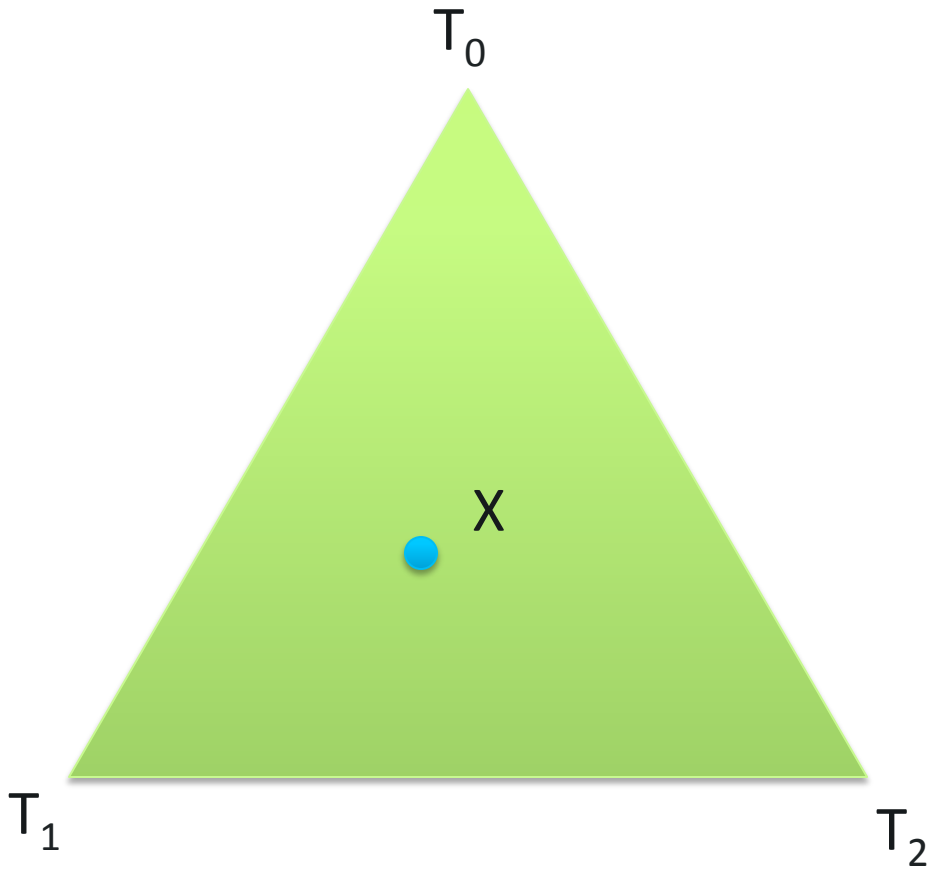
Triangles (Strategy I)

- Check if point on the plane is inside the triangle
- AxC should be same direction as AxB
  - And its not!



Triangles (Strategy II)

- Barycentric coordinates
- $X = \alpha(T_2 - T_1) + \beta(T_0 - T_1)$
- Solve for α and β
  - if $\alpha > 1$ or $\beta > 1$, fail
  - if $\alpha + \beta > 1$, fail



More complicated things?

- Can intersect with any mathematical representation of a 3D structure!
  - But sometimes the math gets kinda complicated
- Examples
  - Cones, cylinders
  - Boxes
  - Polygons (convex, concave)
  - Compound geometry

Optimization Approaches

- Bounding volumes
- Uniform grid
- Octrees
- BSP Trees

- “Embarrassingly parallel”

LIGHTING AND SHADING

Types of Lighting

- Local Illumination
  - Emission from light sources
  - Scattering at surface
- Global Illumination
  - Shadows
  - Reflection
  - Refraction

Types of Lighting

- Local Illumination
  - **Emission from light sources**
  - Scattering at surface
- Global Illumination
  - Shadows
  - Reflection
  - Refraction

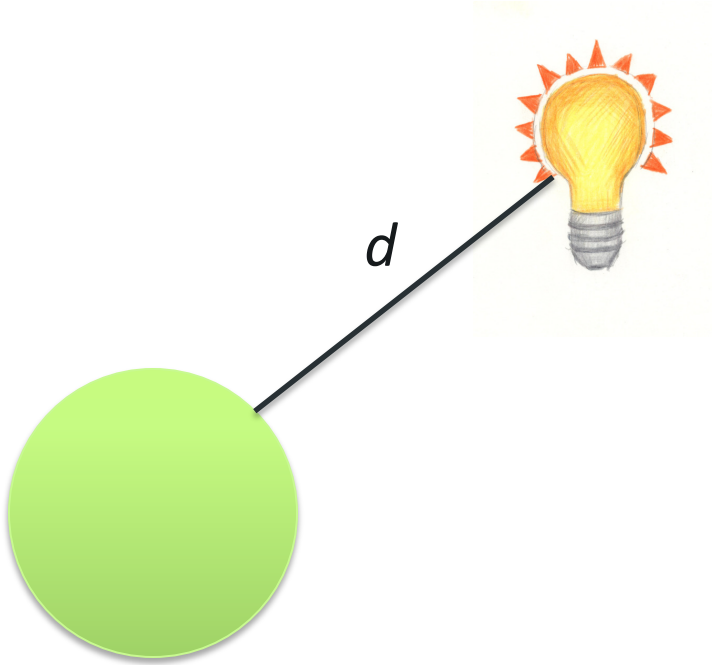
Light Sources

- Point light
- Directional light
- Spot light
- Ambient light

Point Light

- Like a light bulb that radiates light equally in all directions
  - Attenuation: the further an object is from the light, the less light it receives
- Parameters:
  - Position: x, y, z
  - Intensity (I_0): r, g, b
  - Attenuation factor: a, b, c

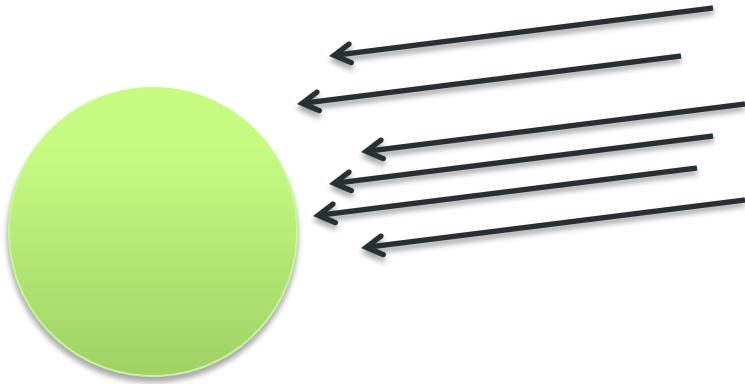
$$I_L = \frac{I_0}{a + b * d + c * d^2}$$



Directional Light

- Like a point light that is infinitely far away with no attenuation
  - Good for modeling “daylight”
- Parameters:
  - Direction: d_x, d_y, d_z
  - Intensity (I_0): r, g, b

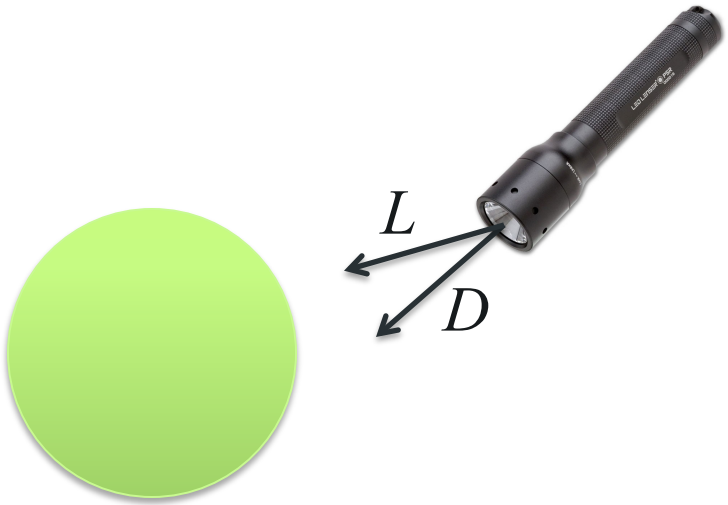
$$I_L = I_0$$



Spot Light

- Directional light that is anchored to a point
  - Desk lamp, flashlight, spotlight
- Parameters:
  - Position: x, y, z
  - Direction: dx, dy, dz
  - Attenuation: a, b, c
  - Intensity (I_0): r, g, b

$$I_L = \frac{I_0(L \bullet D)}{a + bd + cd^2}$$



Ambient Light

- "Hacky" model of light in a scene resulting from it bouncing around everywhere
  - Good for modeling "background" light
- Parameters
  - Intensity: r, g, b

$$I_A = I_0$$

Types of Lighting

- Local Illumination
  - Emission from light sources
  - **Scattering at surface**
- Global Illumination
  - Shadows
  - Reflection
  - Refraction

Surface Material Properties

- M_A: ambient color
- M_D: diffuse color
- M_S: specular color
- n: specular light exponent
- M_E: emissive light color
- M_T: transmissive color
- M_R: index of refraction

Algorithm Overview

for every pixel, *p*, in the image

    let *r* be the ray from camera through pixel

    calculate intersection between *r* and scene objects

    if (!intersection)

        set *p* to “background color”

    else

        ***p* = color calculated from first object it hits**

What color is my pixel?

Amount of ambient light at surface +

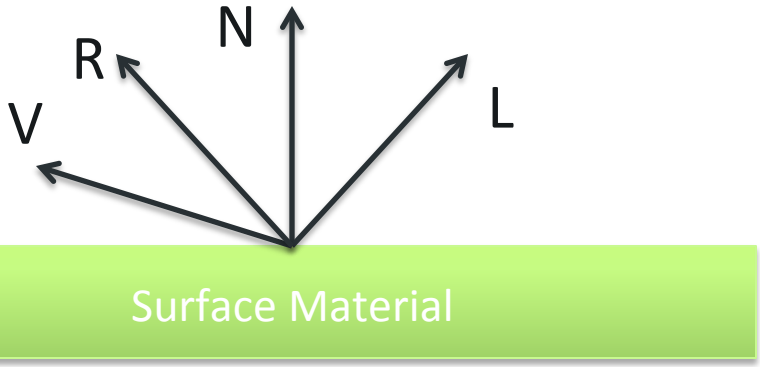
- amount of diffuse light at surface +
- amount of specular light at surface +
- amount of emissive light from material +
- light reflected by the object +
- light due to transparency of object

color = Σ_L …

How much ambient light?

- There is usually only one ambient light in a scene

I

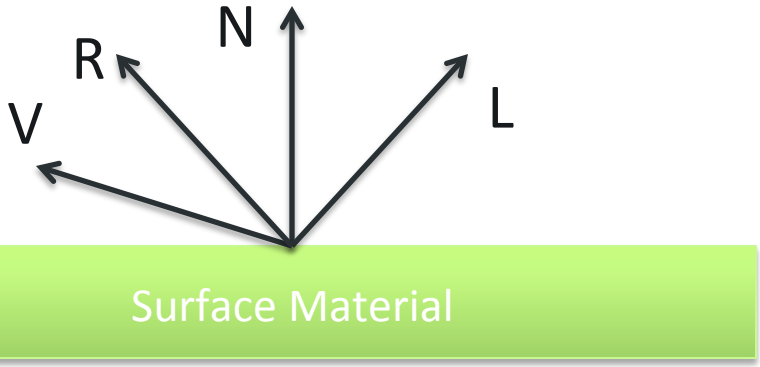


color = $M_A I_A$

How much diffuse light? (Lambertian Shading)

- Diffuse shading: matte color
  - Assume material reflects light evenly in all directions

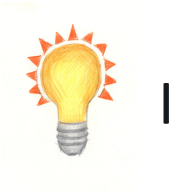
I



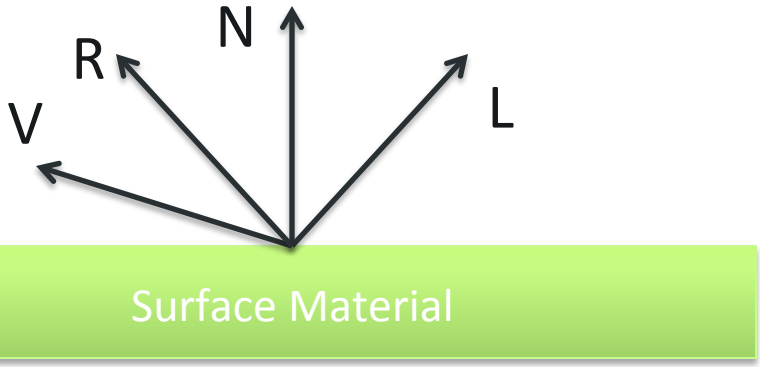
color = $M_A I_A + \Sigma_L M_D (N \bullet L) I_L$

How much specular light? (Phong Shading)

- Specular light: idealized reflection
  - Depends on how much is seen by viewer

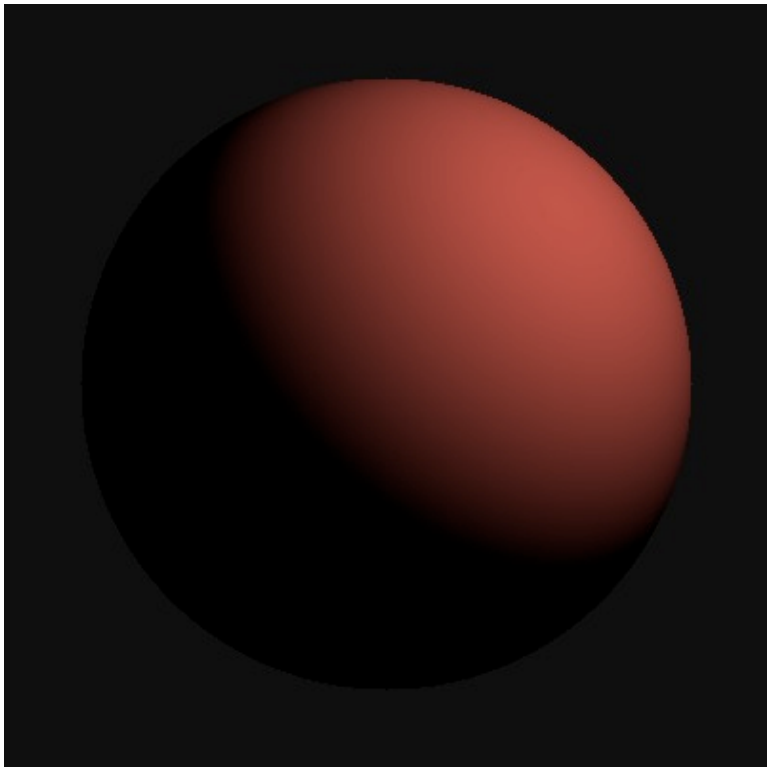


I

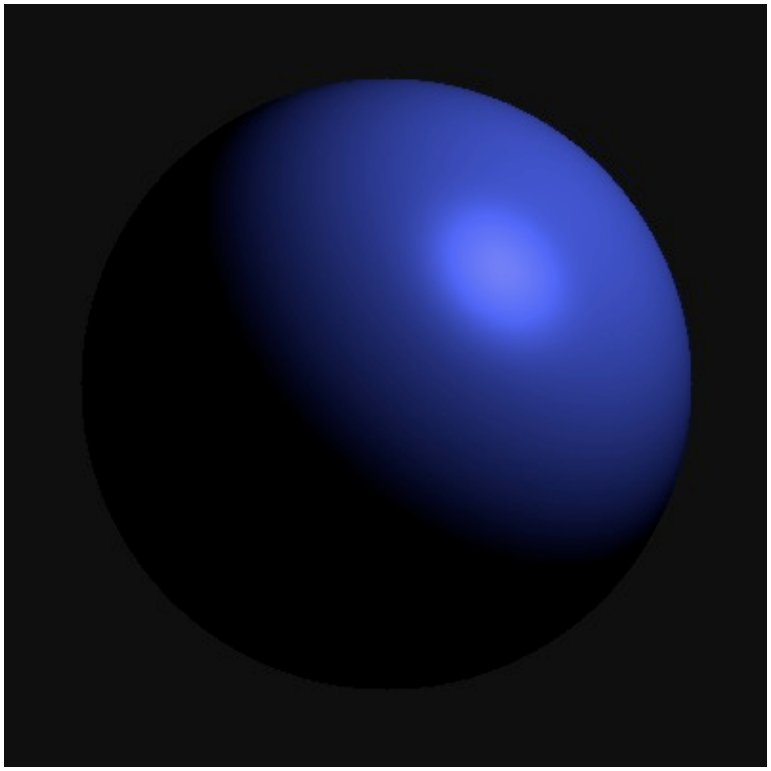


$$\text{color} = M_A I_A + \Sigma_L (M_D (N \bullet L) I_L + \mathbf{M_S (V \bullet R)^n I_L})$$

Lambertian and Phong Shading: Examples



Lambertian (Diffuse)

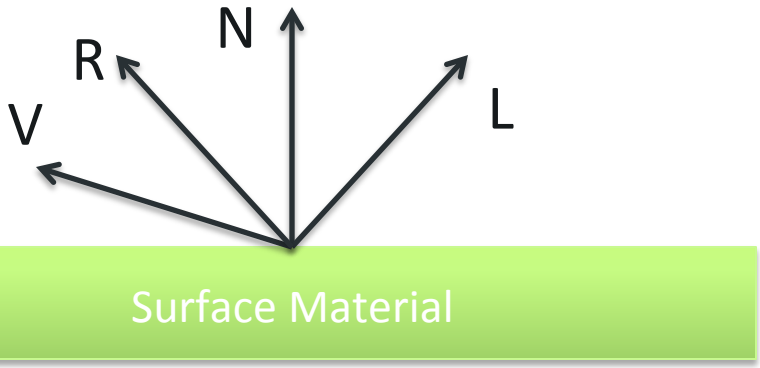


Blinn-Phong (Specular)

How much emissive light?

- Specular light: idealized reflection
  - Depends on how much is seen by viewer

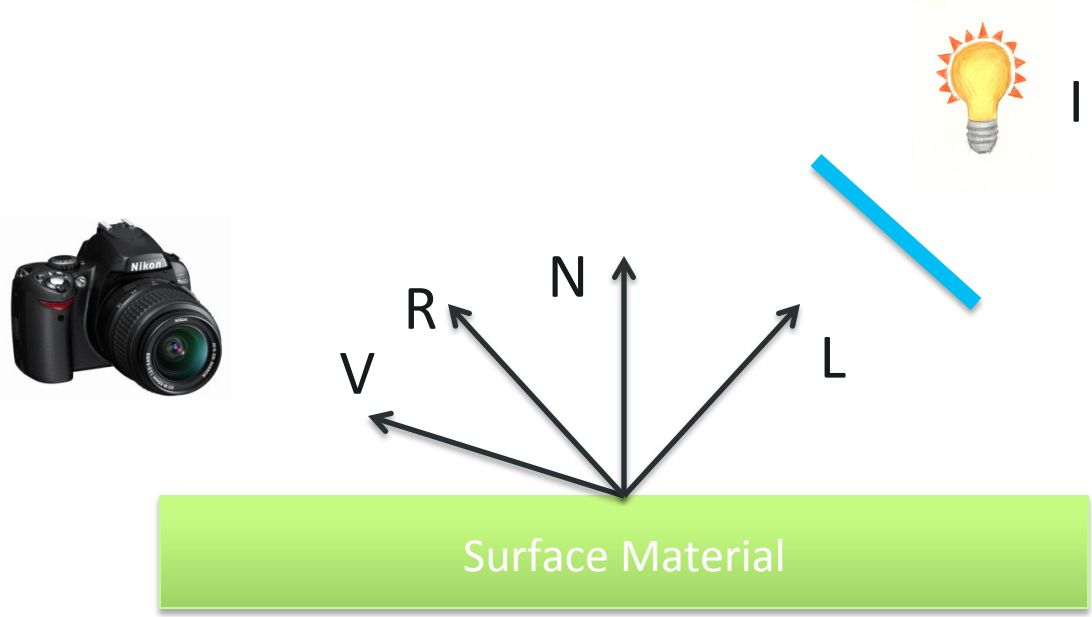
I



$$\text{color} = M_A I_A + \mathbf{M_E} + \Sigma_L (M_D (N \bullet L) I_L + M_S (V \bullet R)^n I_L)$$

Shadows

- If there is an object in the way of the light, it's in shadow for that light



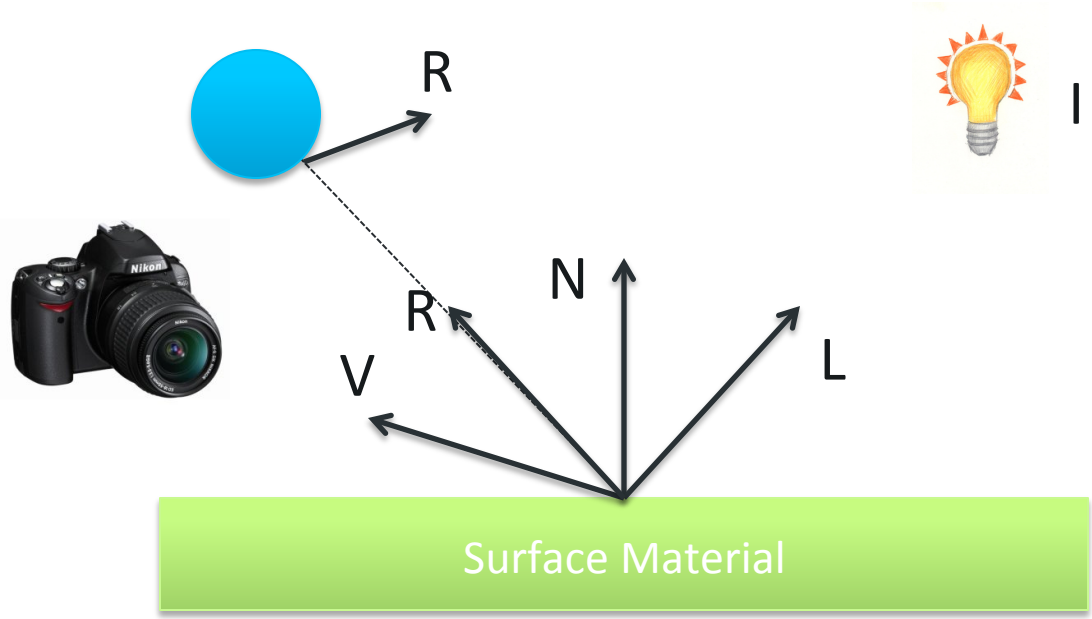
$$\text{color} = M_A I_A + M_E + \Sigma_L \mathbf{S_L} (M_D (N \bullet L) I_L + M_S (V \bullet R)^n I_L)$$

Types of Lighting

- Local Illumination
  - Emission from light sources
  - Scattering at surface
- **Global Illumination**
  - Shadows
  - Reflection
  - Refraction

Reflection

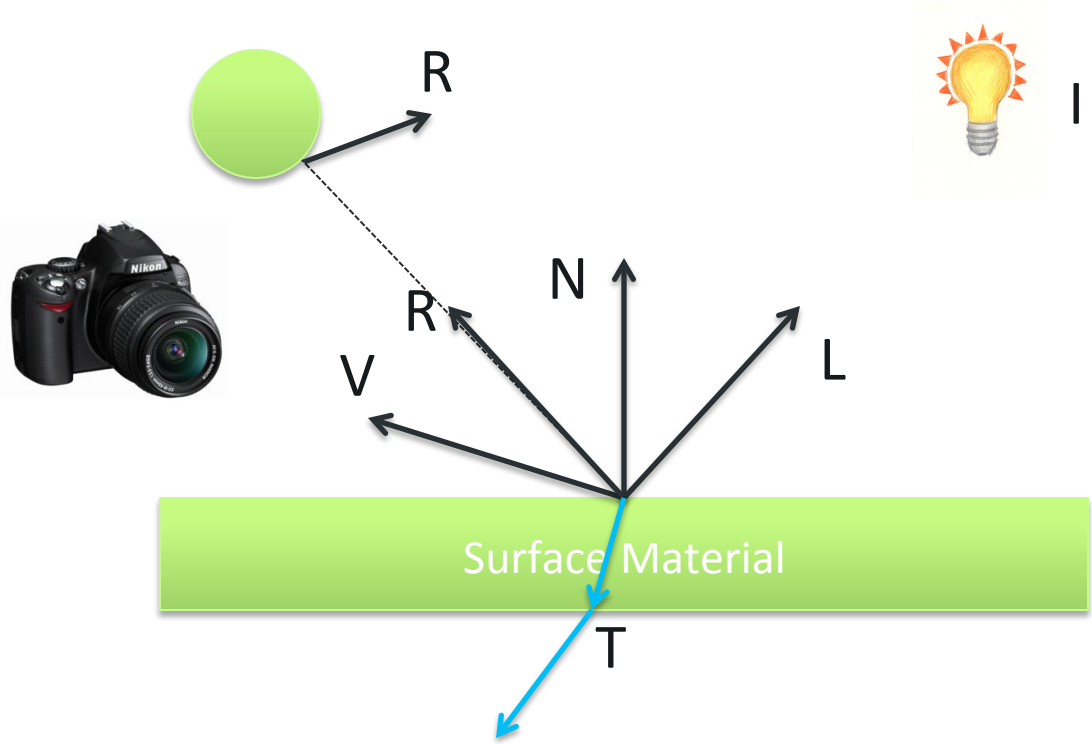
- If there is an object in the way of the light, it's in shadow for that light



$$\text{color} = M_A I_A + M_E + \Sigma_L S_L (M_D (N \bullet L) I_L + M_S (V \bullet R)^n I_L) + \mathbf{M_S I_R}$$

Refraction

- Transmissive light: how does light bend as it goes through the surface? [Book chapter 13.1]



$$\text{color} = M_A I_A + M_E + \Sigma_L S_L (M_D (N \bullet L) I_L + M_S (V \bullet R)^n I_L) + M_S I_R + \mathbf{M_T I_T}$$

Raytracing vs. Raycasting

- Raycasting
  - Non-recursive
  - Only capable of diffuse, specular, emissive lighting
- Raytracing
  - Recursive
  - Send out rays for:
    - Shadow
    - Reflection
    - Refraction
  - Important to have maximum depth of rays to avoid infinite loops (e.g. hall of mirrors)

practice your elevator pitch

2 MINUTE TALKS

Ideal Talk Presentation Format

- 15 seconds
  - Who you are, what your project is called
- 1 minute
  - Summary of what you did
- 45 seconds
  - Video demonstrating what you did

Goals of the Elevator Pitch

- Creating a positive impression

- Opening the door to further conversation

Source: http://graduateschool.nd.edu/assets/32665/elevator_pitch_presentation.pdf