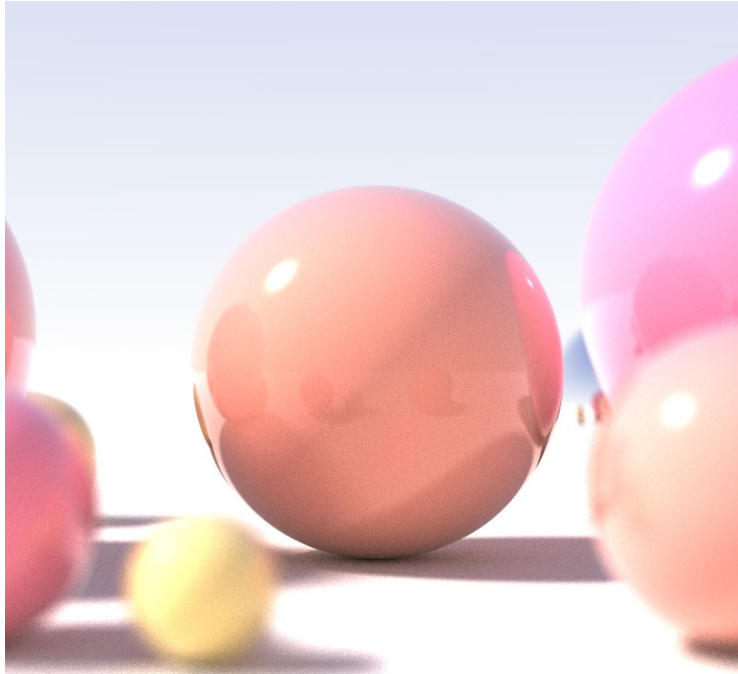


Ray Tracing: Part 2



Some Slides/Images adapted from Marschner and Shirley and David Levin

Announcements

Assignment 3 is due 2 June

Any Questions?

Ray Tracing

(Last week)

Review Ray Casting

Point and Directional Lights

Lambertian Shading Model

Blinn-Phong Shading Model

(Today!)

Shadows

Reflection

Transparency and Refraction

Ray Casting

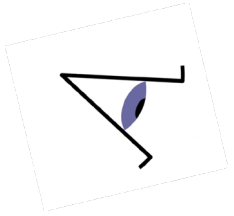
```
for each pixel in the image {  
    Generate a ray  
    for each object in the  
    scene {  
        if (Intersect ray with  
            object) { Set pixel  
                colour  
        }  
    }  
}
```

Review: Blinn-Phong Shading Model

$$L = k_a I_a + \sum_{i=1}^N (k_d I_i \max(0, \mathbf{n} \cdot \mathbf{l}_i) + k_s I_i \max(0, \mathbf{n} \cdot \mathbf{h}_i)^p)$$

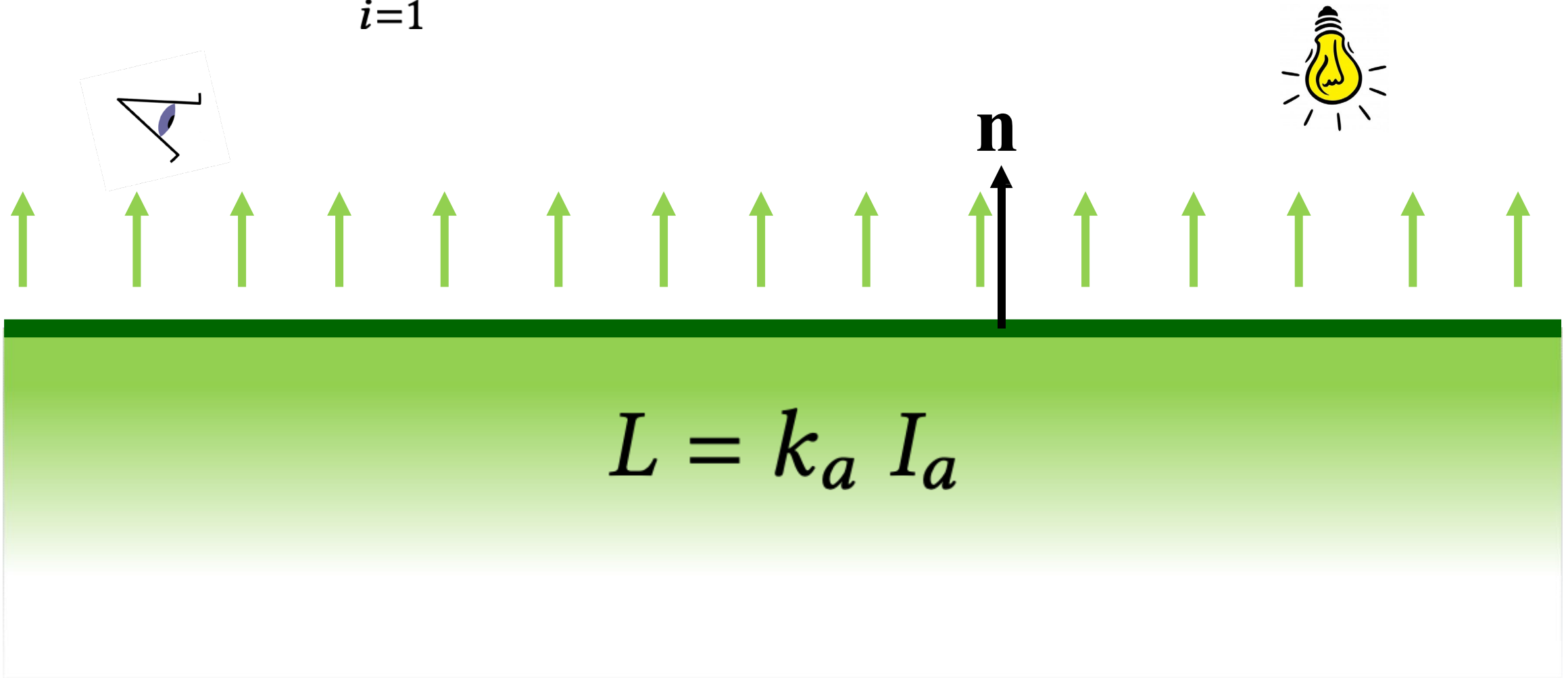
Review: Blinn-Phong Shading Model

$$L = k_a I_a + \sum_{i=1}^N (k_d I_i \max(0, \mathbf{n} \cdot \mathbf{l}_i) + k_s I_i \max(0, \mathbf{n} \cdot \mathbf{h}_i)^p)$$



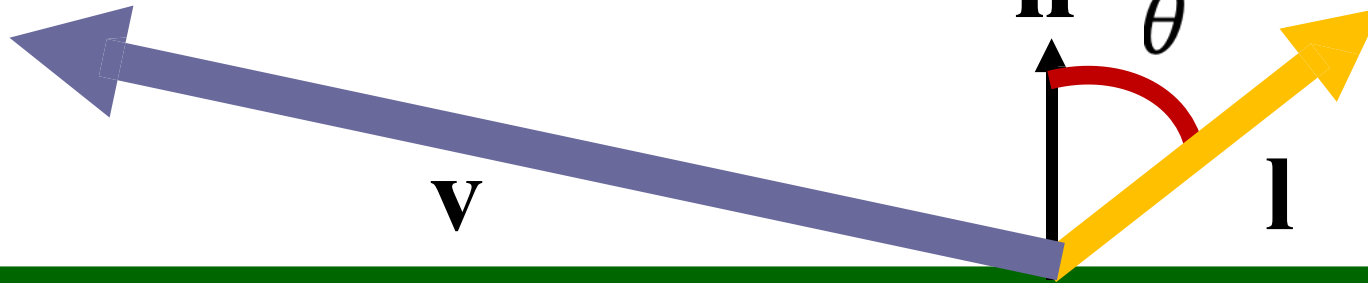
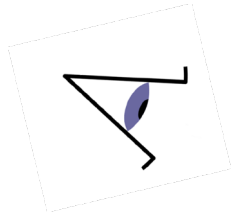
Review: Blinn-Phong Shading Model

$$L = \boxed{k_a I_a} + \sum_{i=1}^N (k_d I_i \max(0, \mathbf{n} \cdot \mathbf{l}_i) + k_s I_i \max(0, \mathbf{n} \cdot \mathbf{h}_i)^p)$$



Review: Blinn-Phong Shading Model

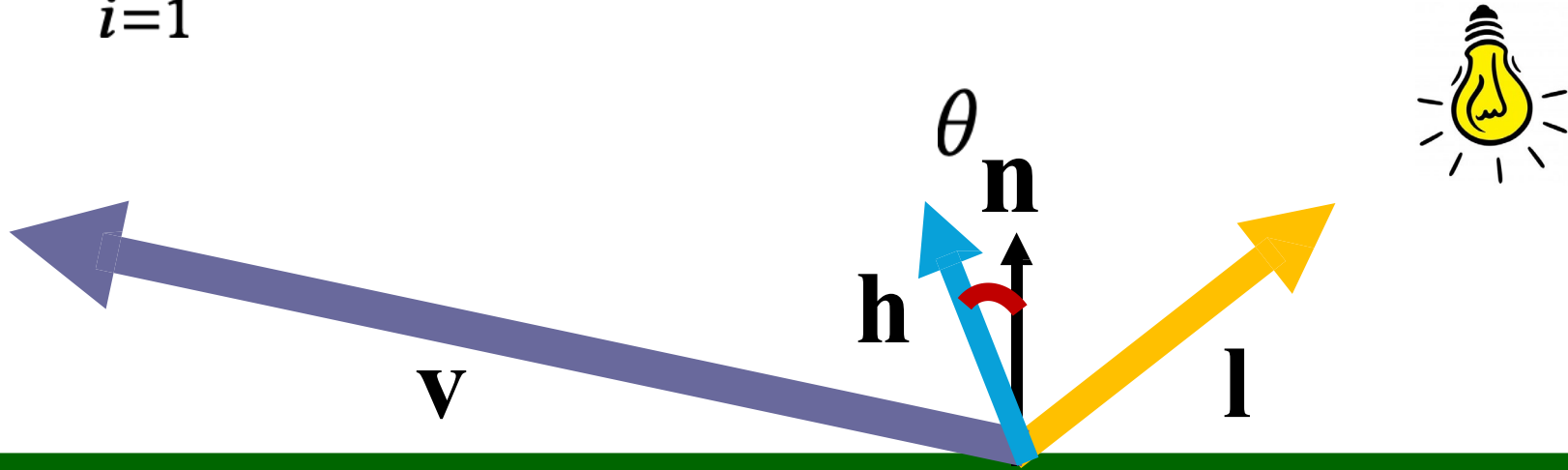
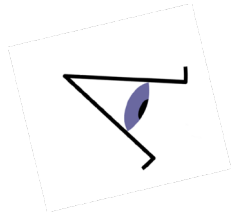
$$L = k_a I_a + \sum_{i=1}^N (k_d I_i \max(0, \mathbf{n} \cdot \mathbf{l}_i) + k_s I_i \max(0, \mathbf{n} \cdot \mathbf{h}_i)^p)$$



$$L = k_d I \max(0, \mathbf{n} \cdot \mathbf{l})$$

Review: Blinn-Phong Shading Model

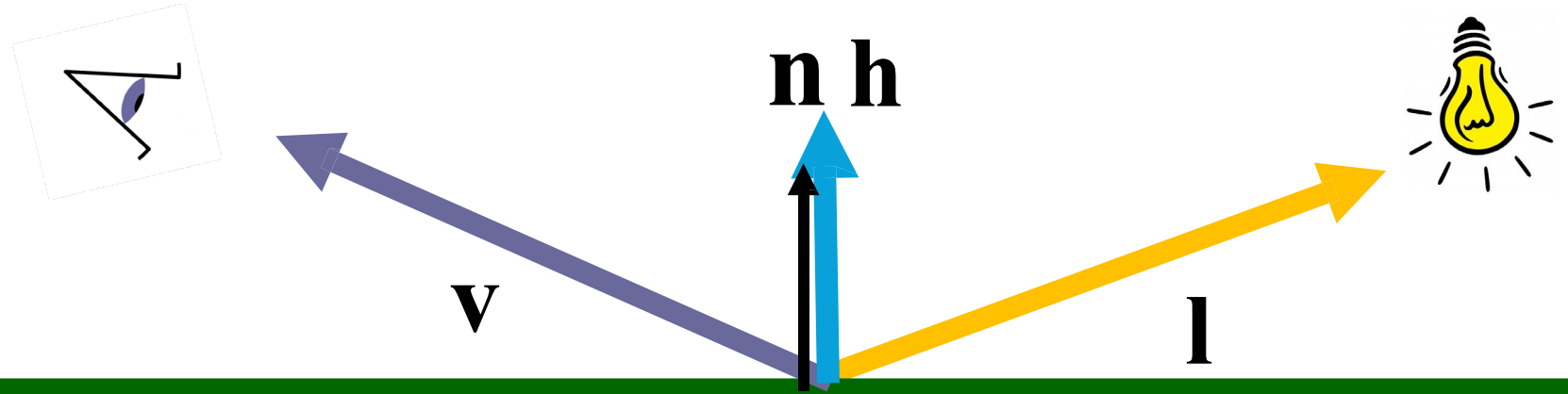
$$L = k_a I_a + \sum_{i=1}^N (k_d I_i \max(0, \mathbf{n} \cdot \mathbf{l}_i) + k_s I_i \max(0, \mathbf{n} \cdot \mathbf{h}_i)^p)$$



$$L = k_s I \max(0, \mathbf{n} \cdot \mathbf{h})^p$$

Review: Blinn-Phong Shading Model

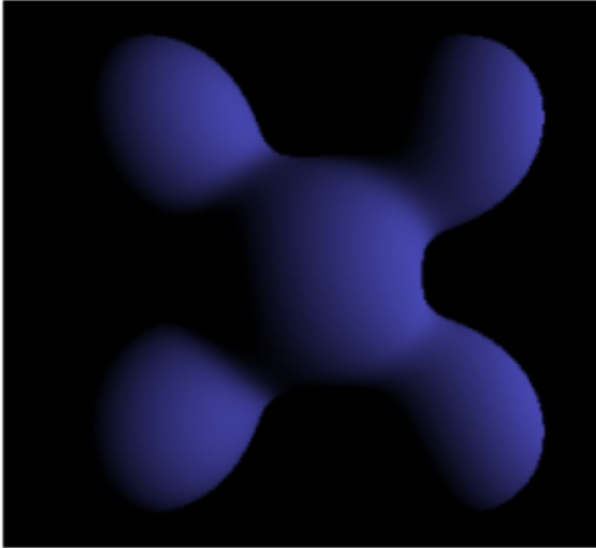
$$L = k_a I_a + \sum_{i=1}^N (k_d I_i \max(0, \mathbf{n} \cdot \mathbf{l}_i) + k_s I_i \max(0, \mathbf{n} \cdot \mathbf{h}_i)^p)$$



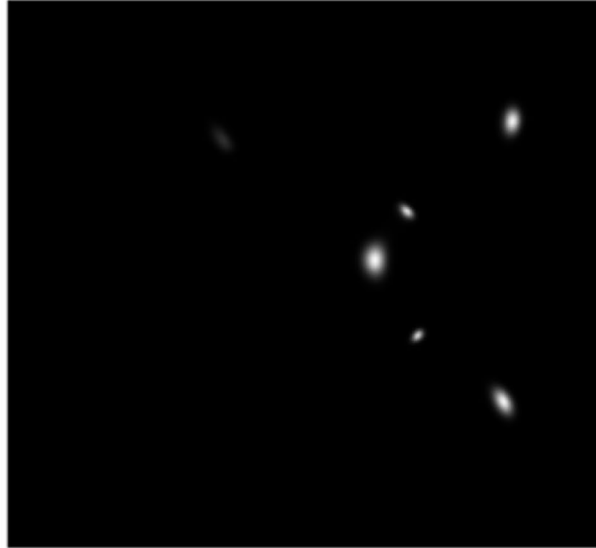
$$L = k_s I \max(0, \mathbf{n} \cdot \mathbf{h})^p$$



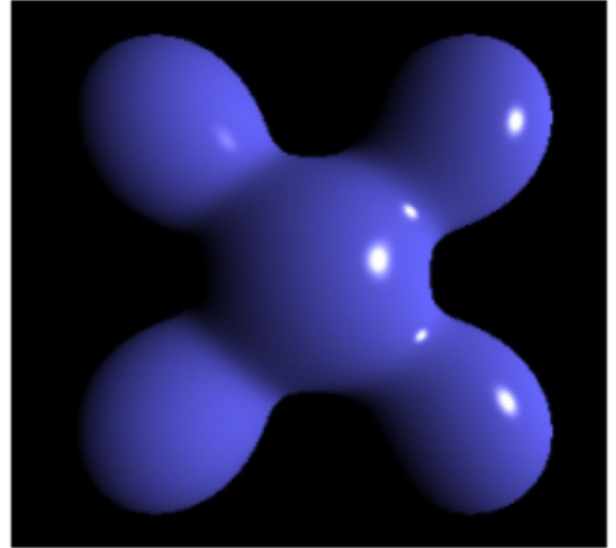
Ambient



Diffuse

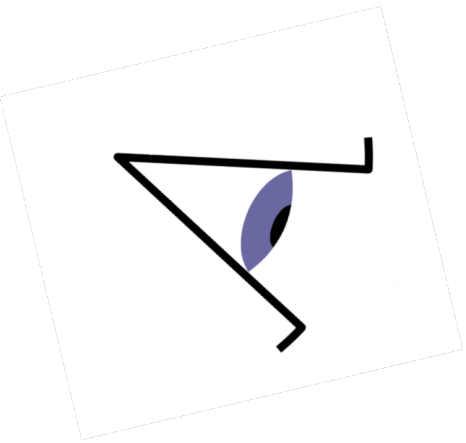


Specular



Final

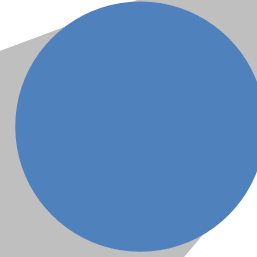
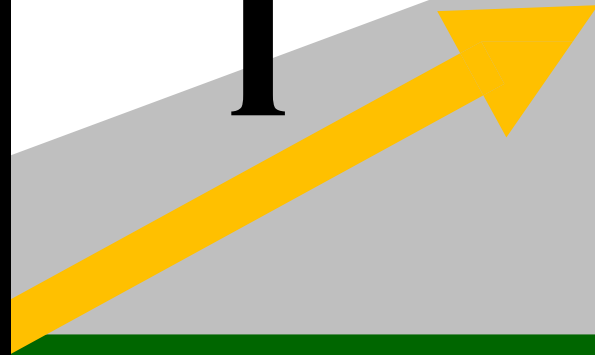
Adding Shadows



n



l



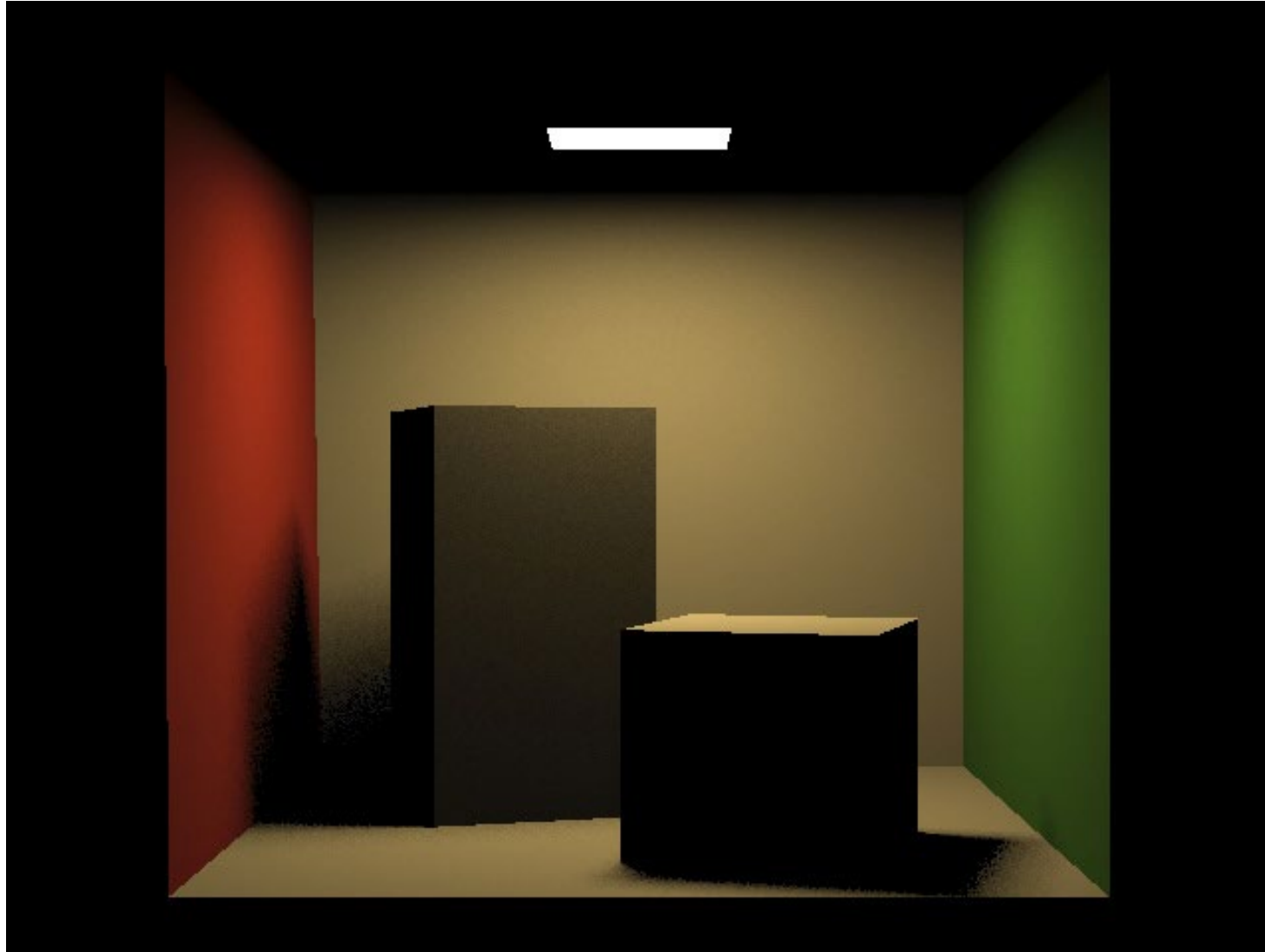
Numerical Precision

What are valid values t for the shadow ray?

Ray Casting

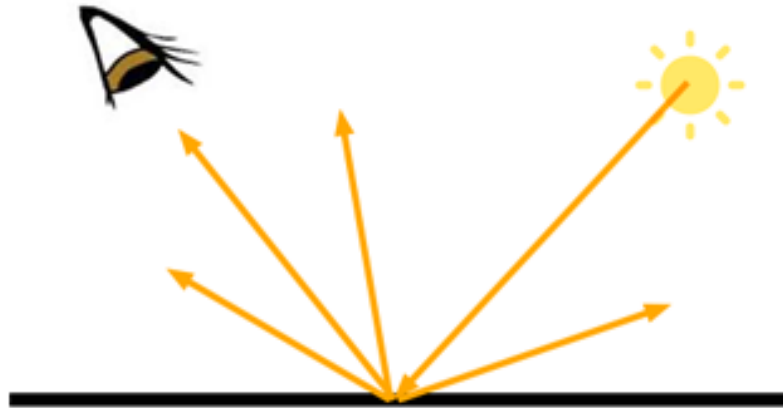
```
for each pixel in the image {  
    Generate a ray  
    for each object in the  
    scene {  
        if (Intersect ray with  
            object) { Set pixel  
                colour  
        }  
    }  
}
```

No Global Effects

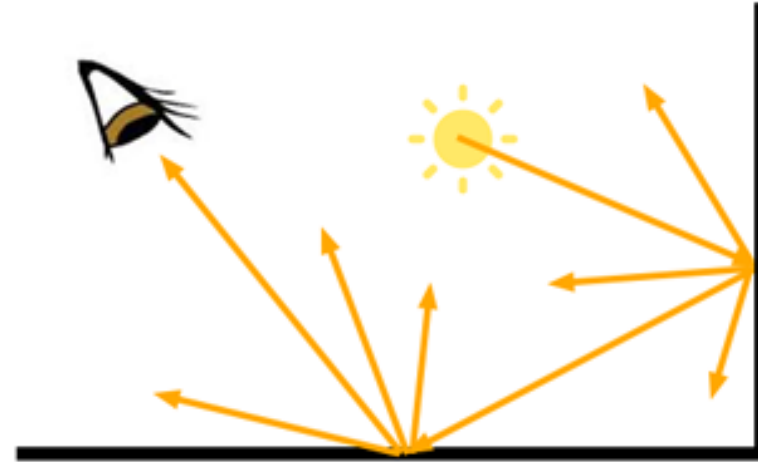


<http://www.deluxerender.com/2017/01/the-cornell-box-a-renderers-rite-of-pathage/>

No Global Effects

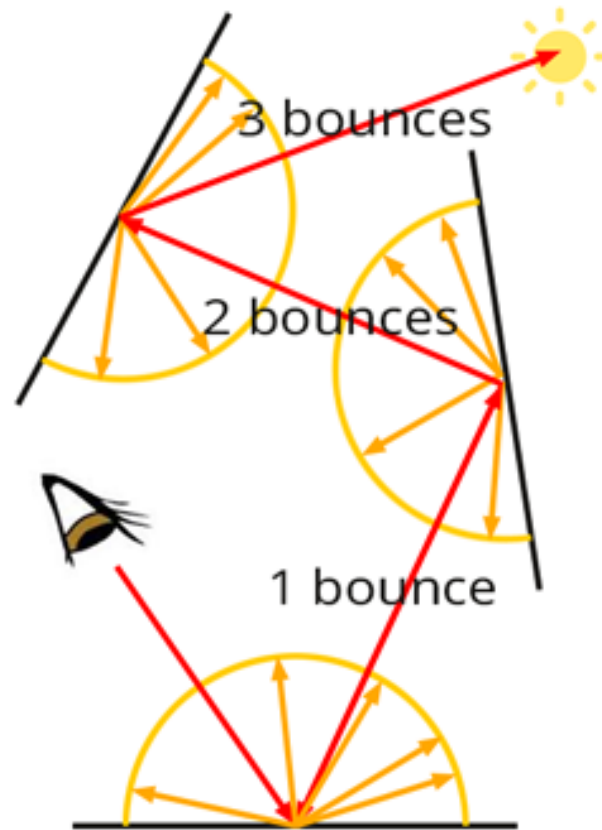


direct illumination

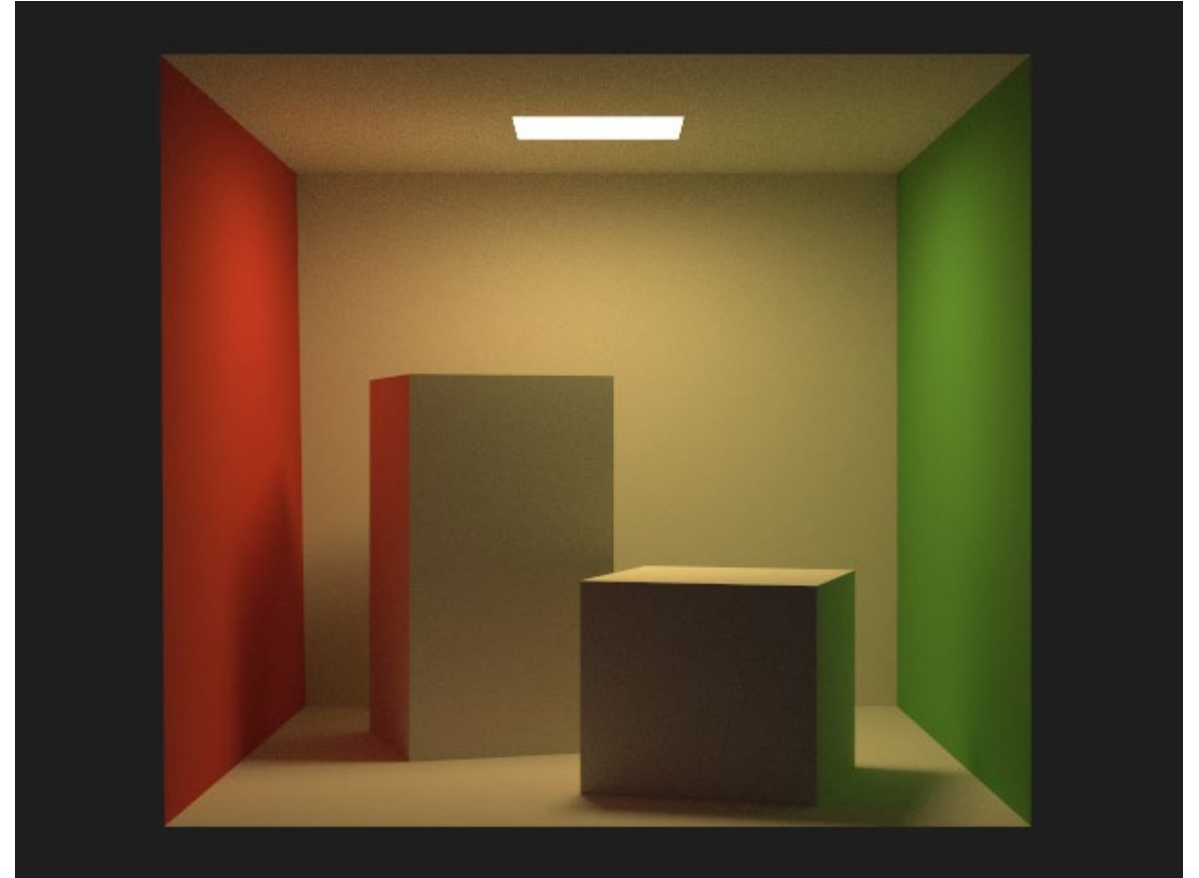
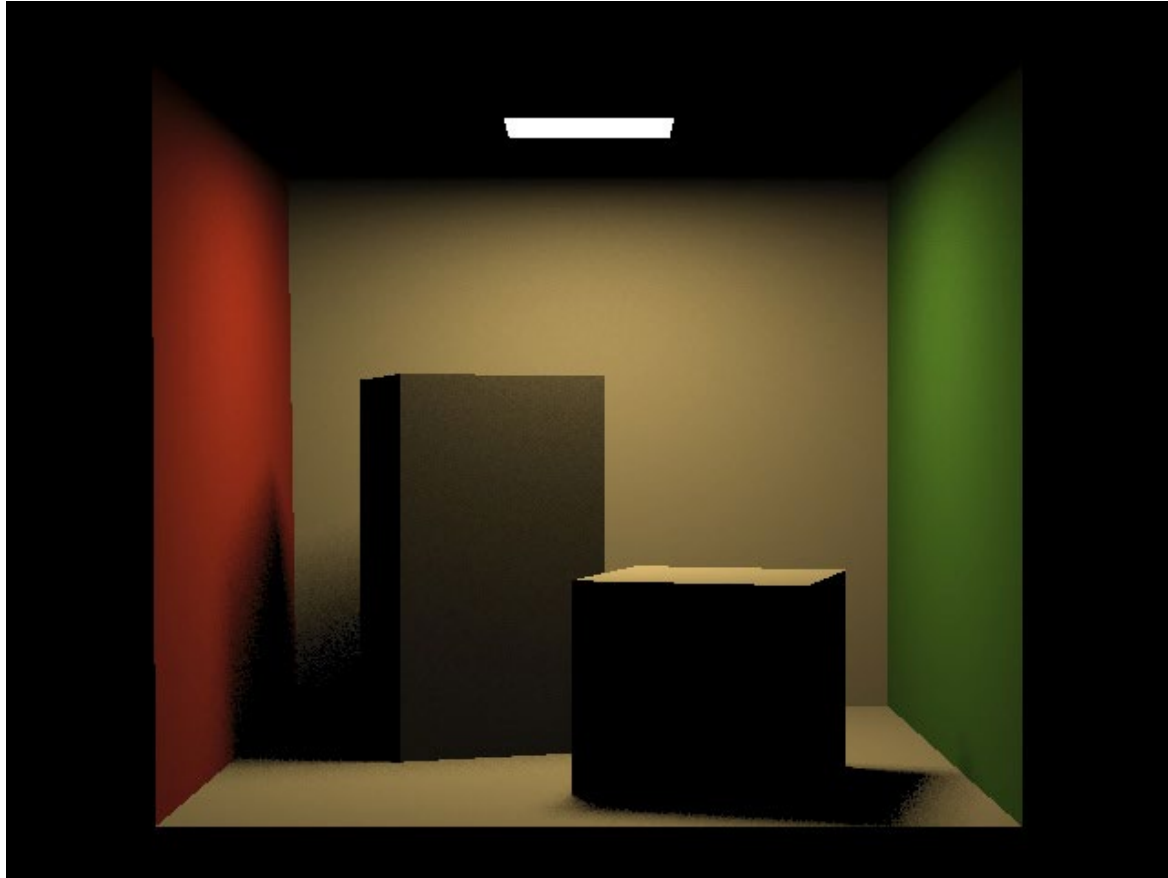


indirect illumination

No Global Effects

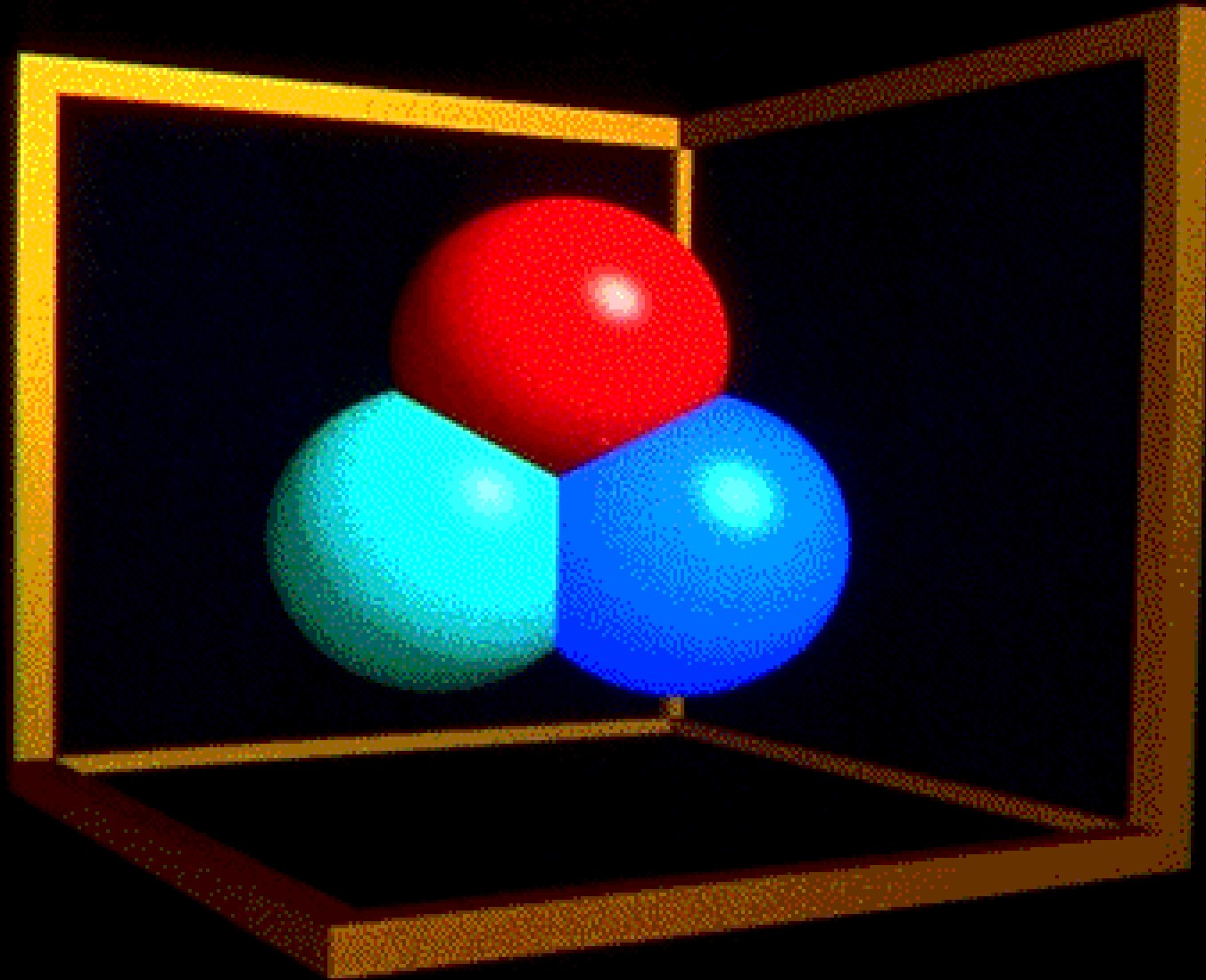


Global Effects

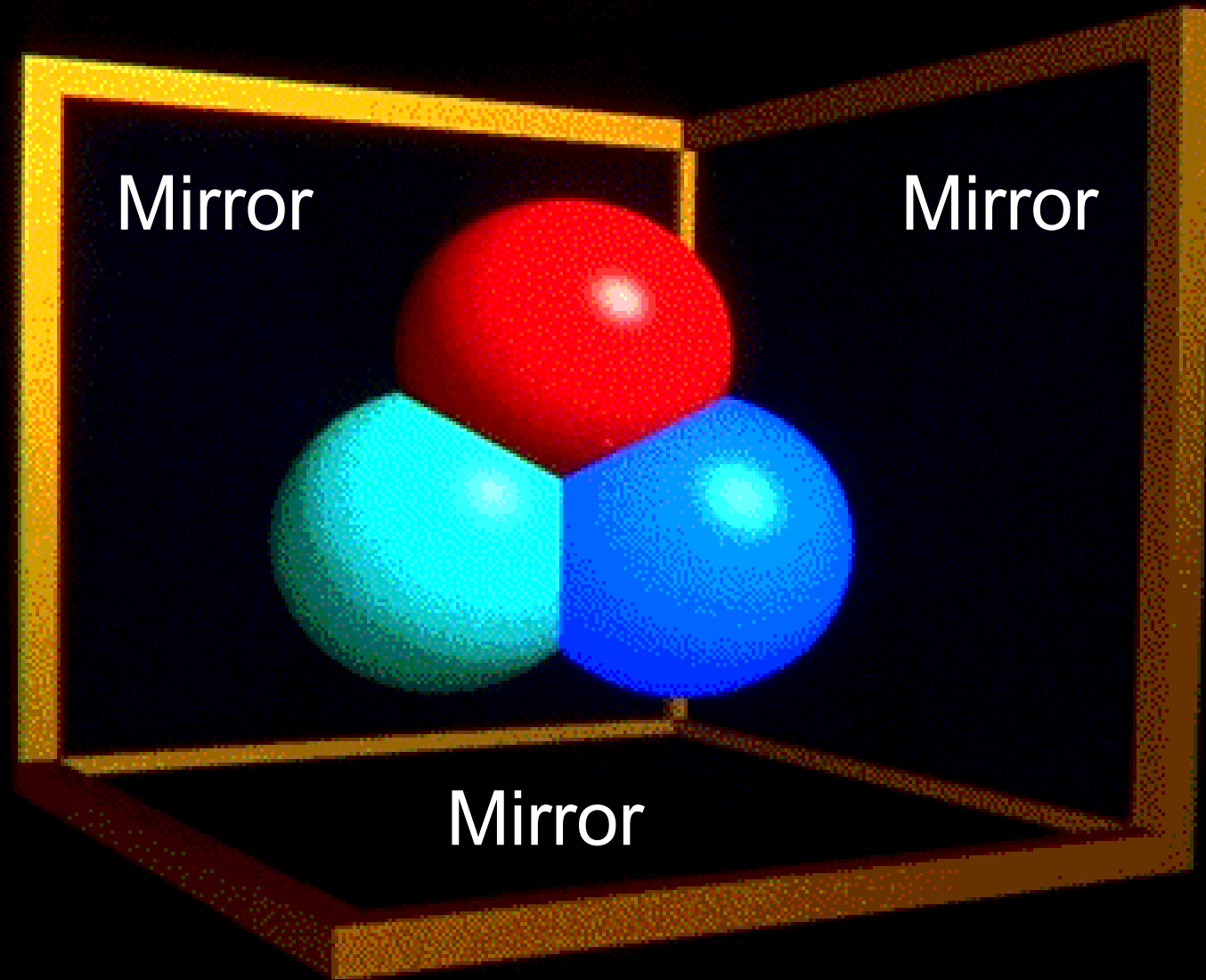


<http://www.deluxerender.com/2017/01/the-cornell-box-a-renderers-rite-of-pathage/>
https://en.wikipedia.org/wiki/Cornell_box

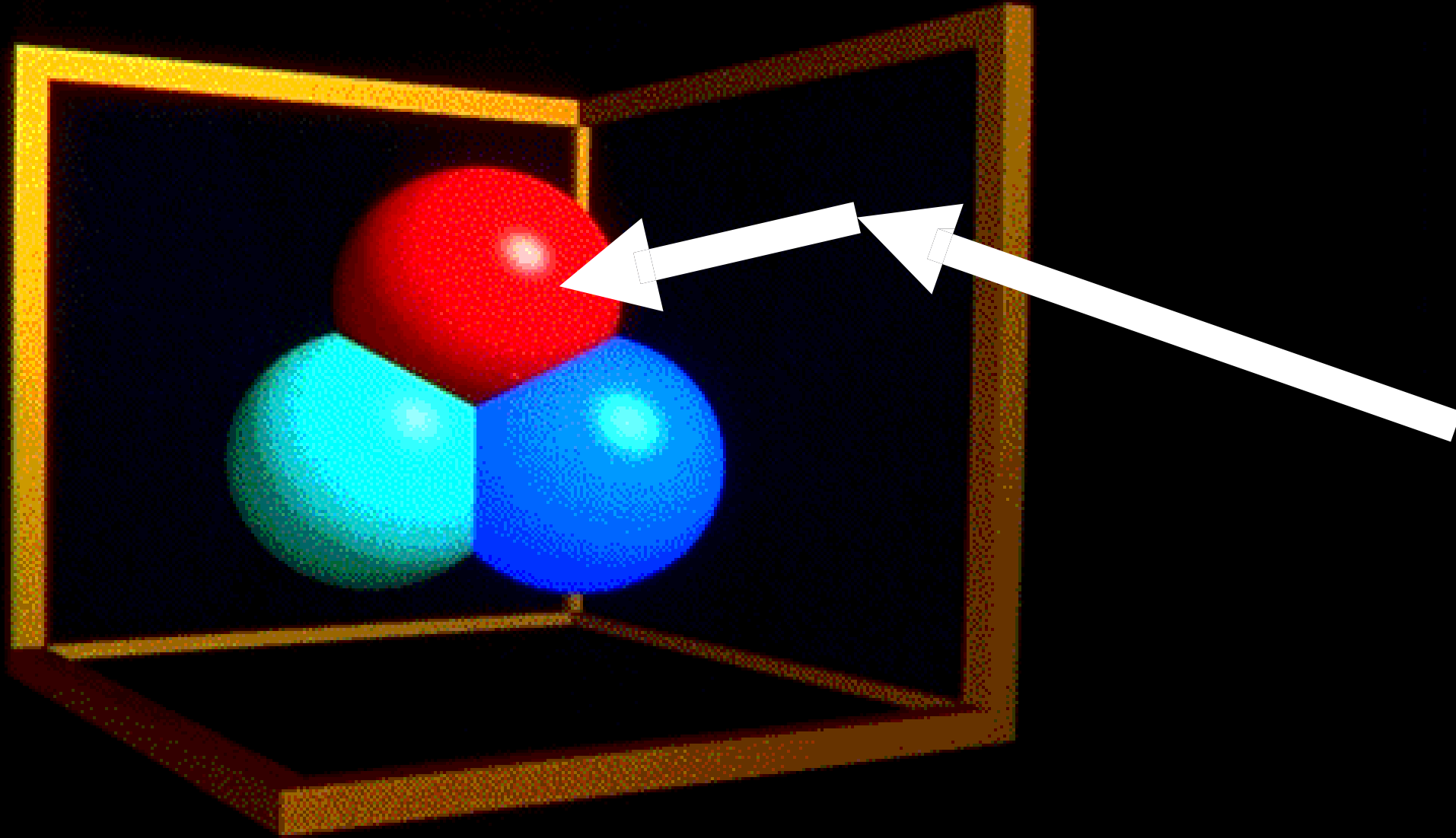
Ray Traced Image



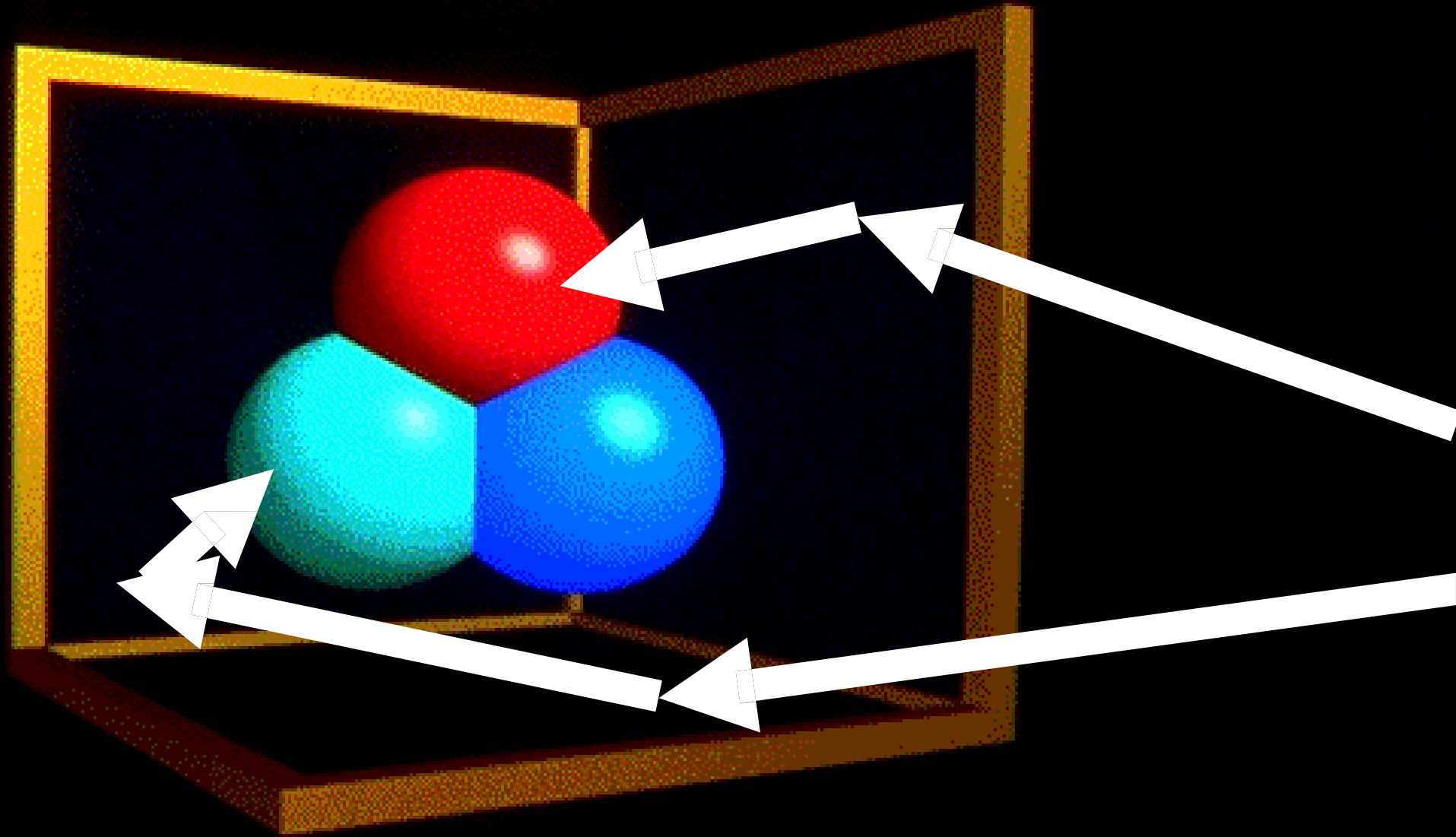
Ray Traced Image

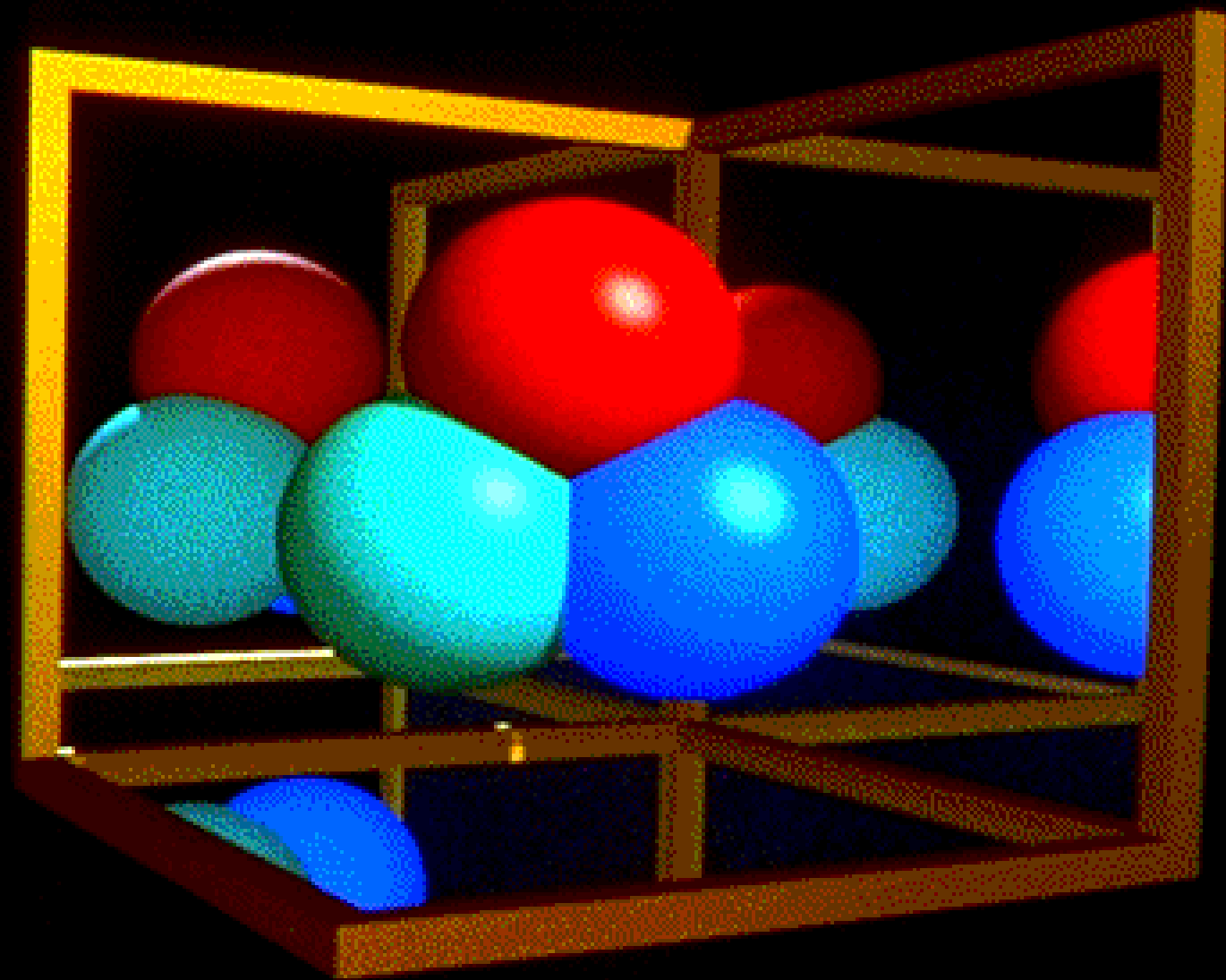


Recursive Ray Tracing

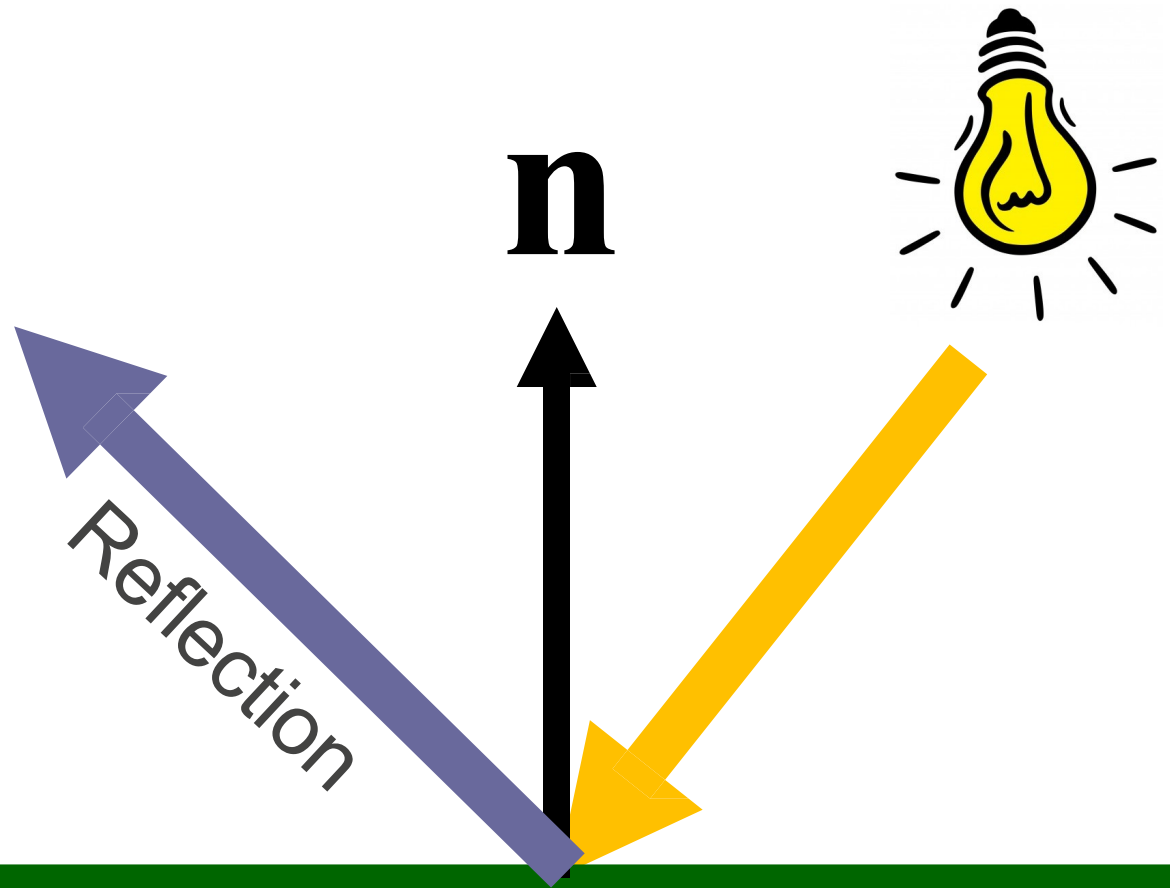


Recursive Ray Tracing

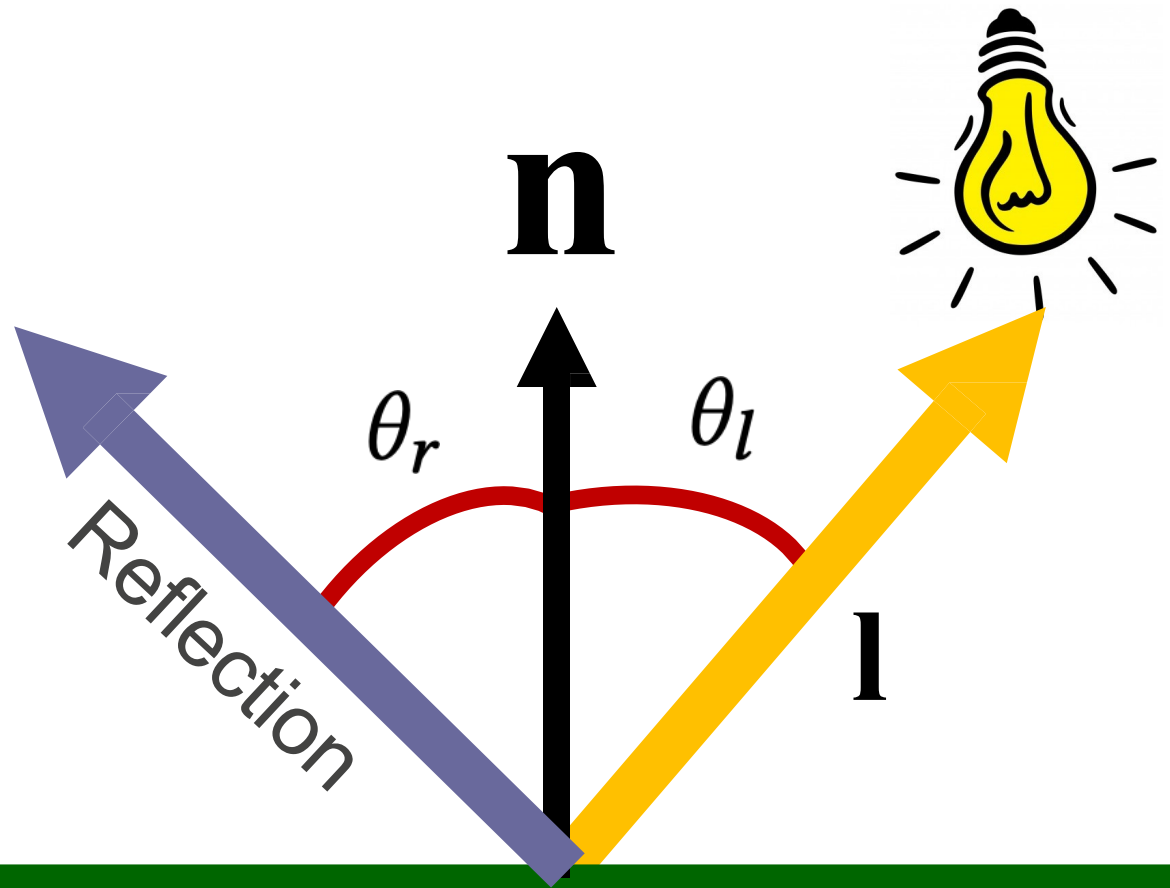




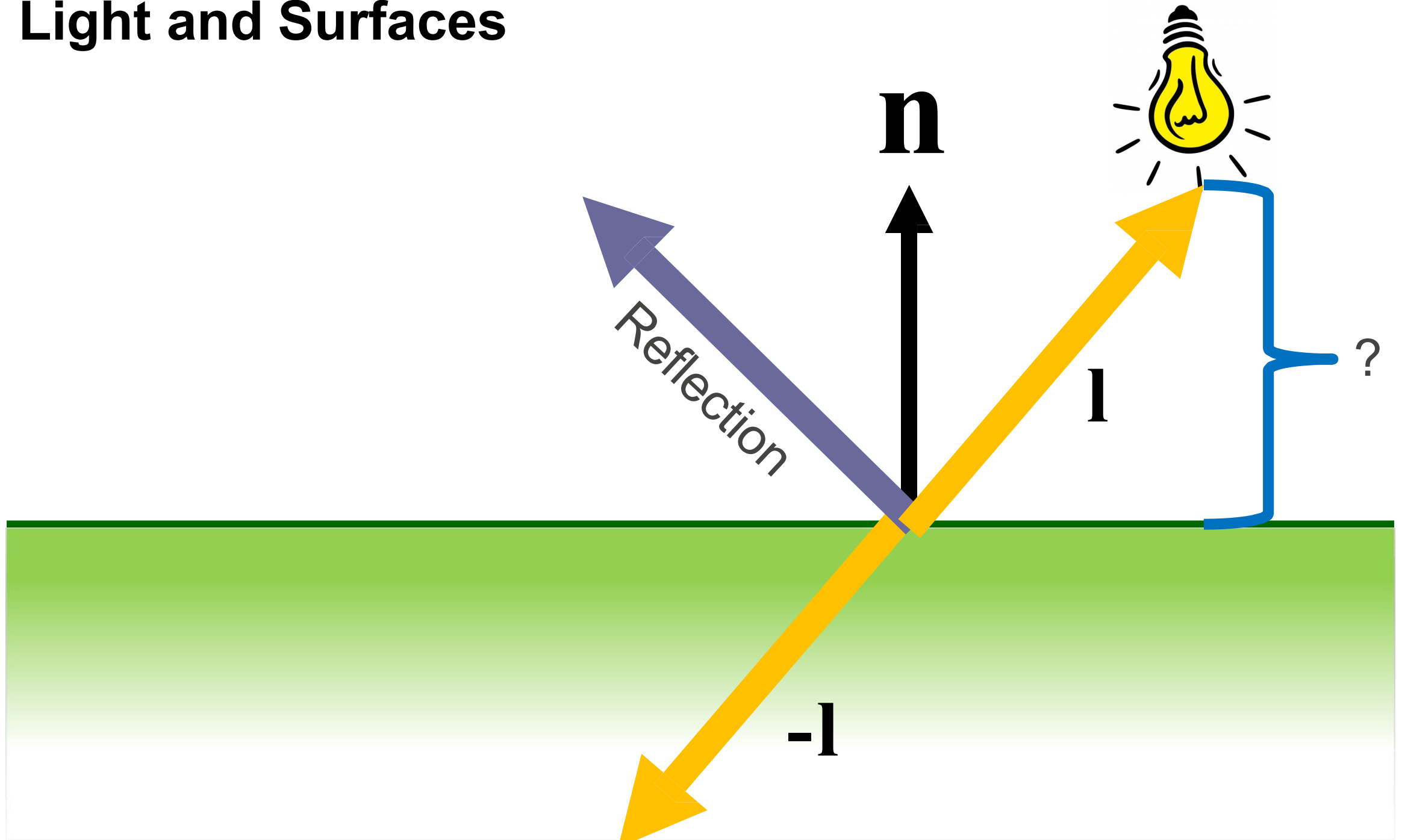
Light and Surfaces



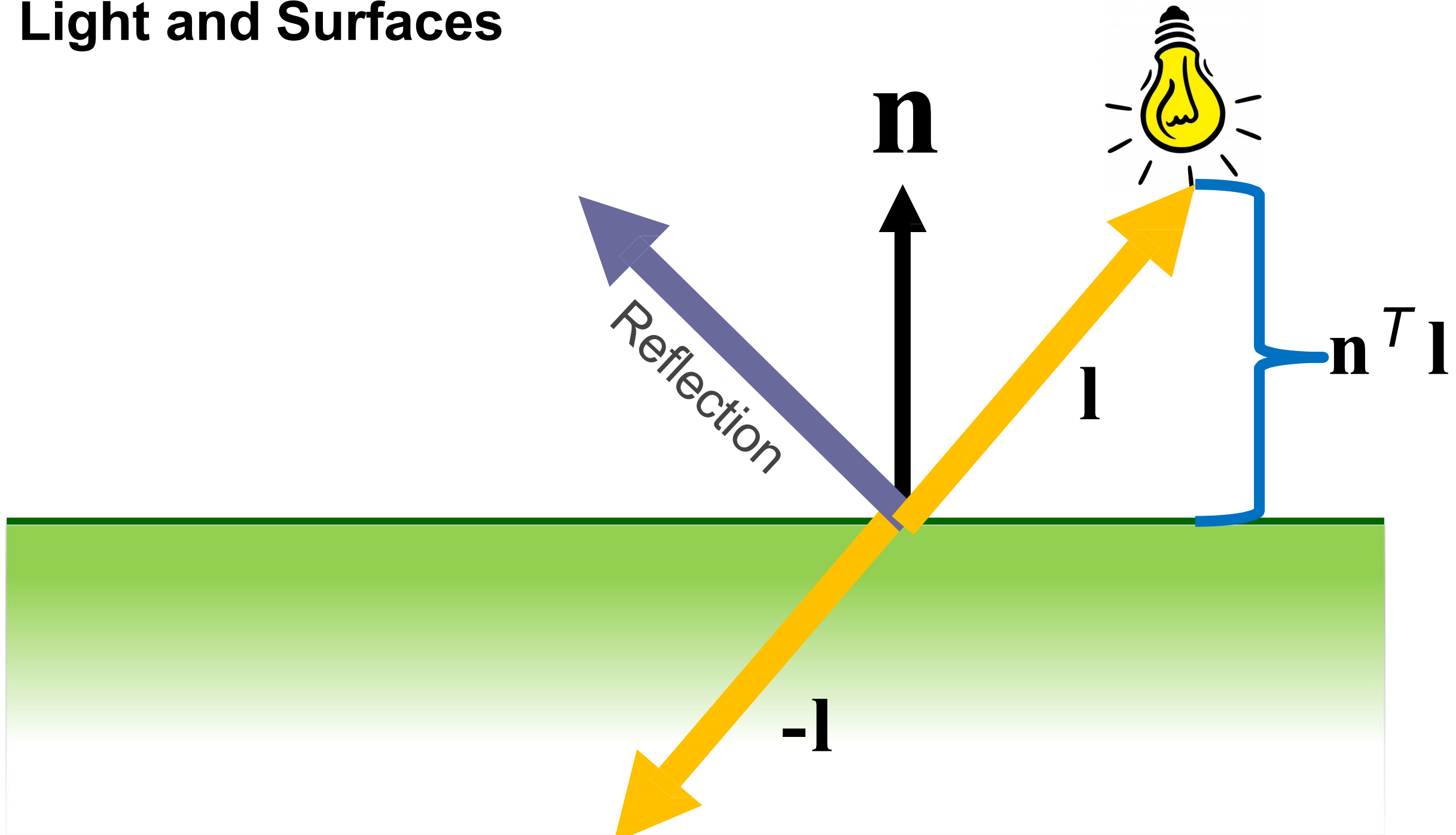
Light and Surfaces



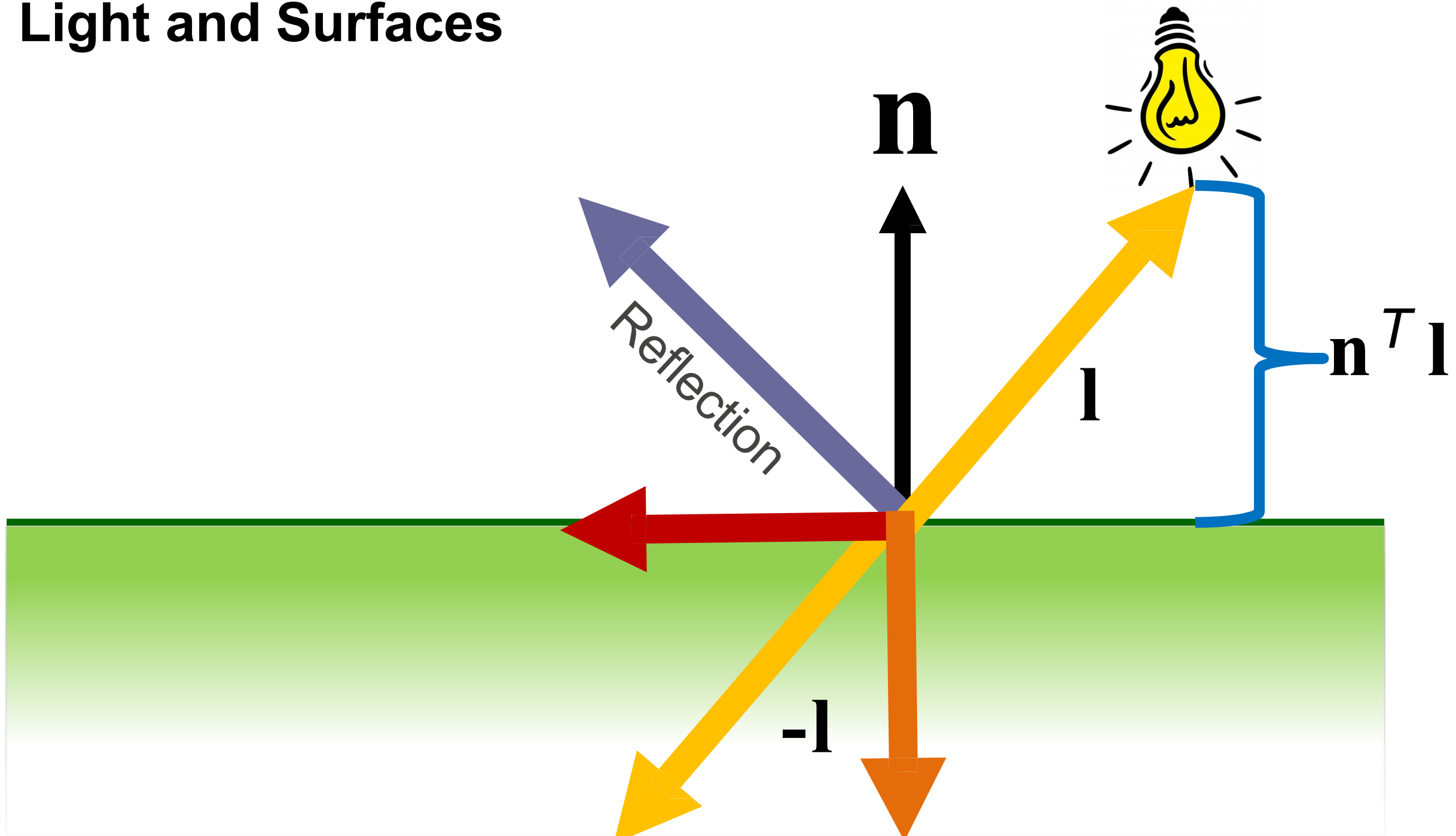
Light and Surfaces



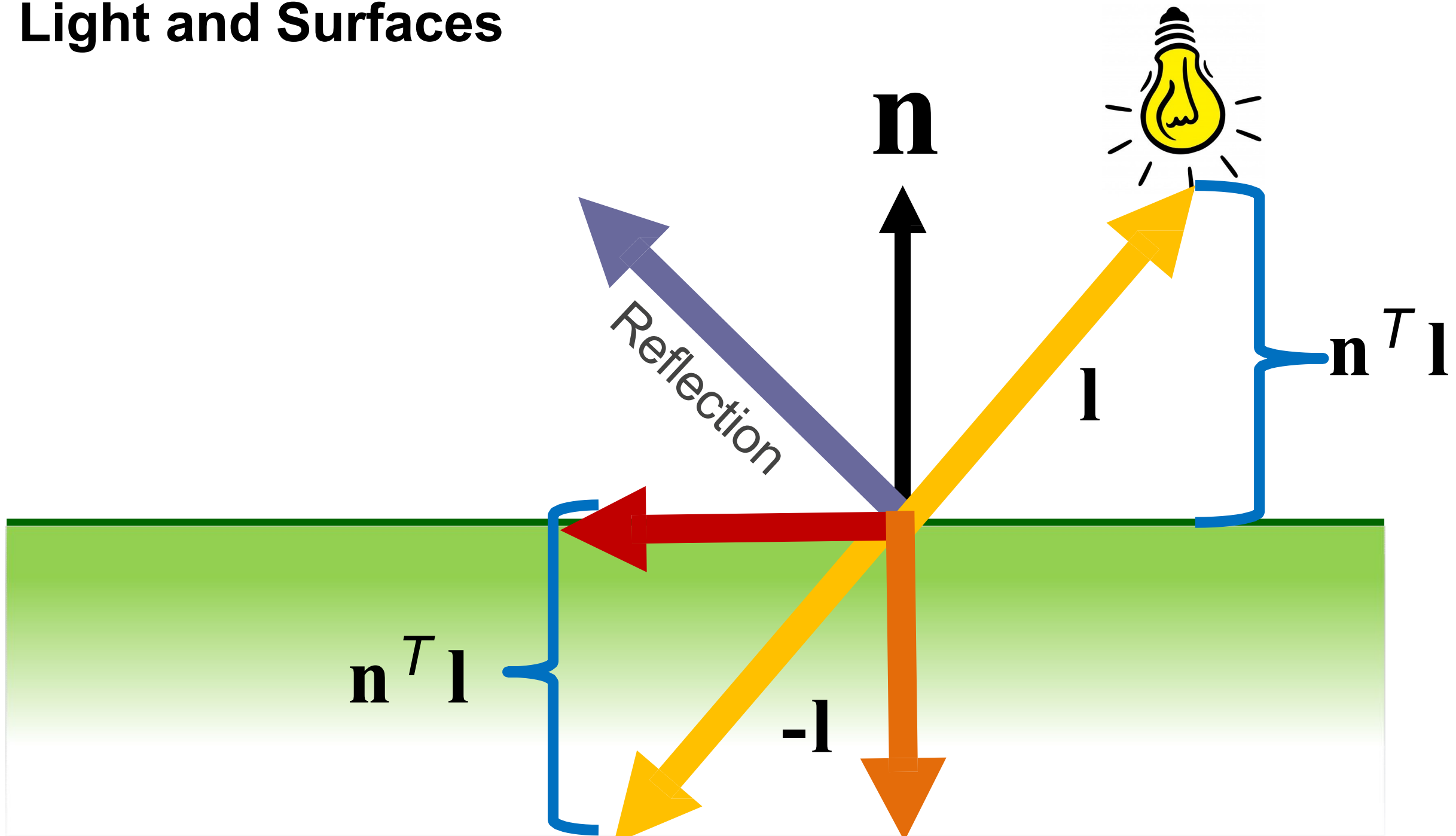
Light and Surfaces



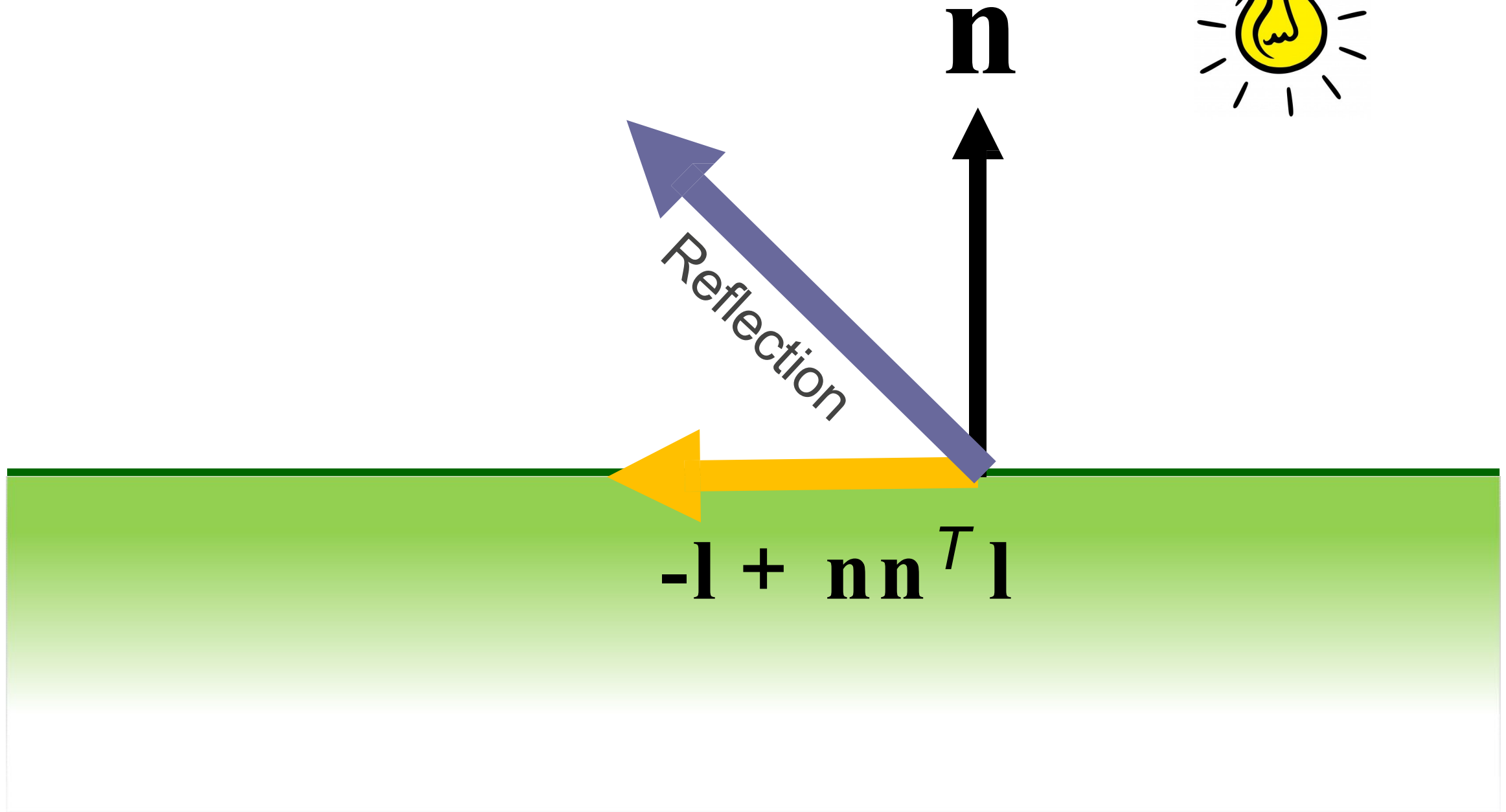
Light and Surfaces



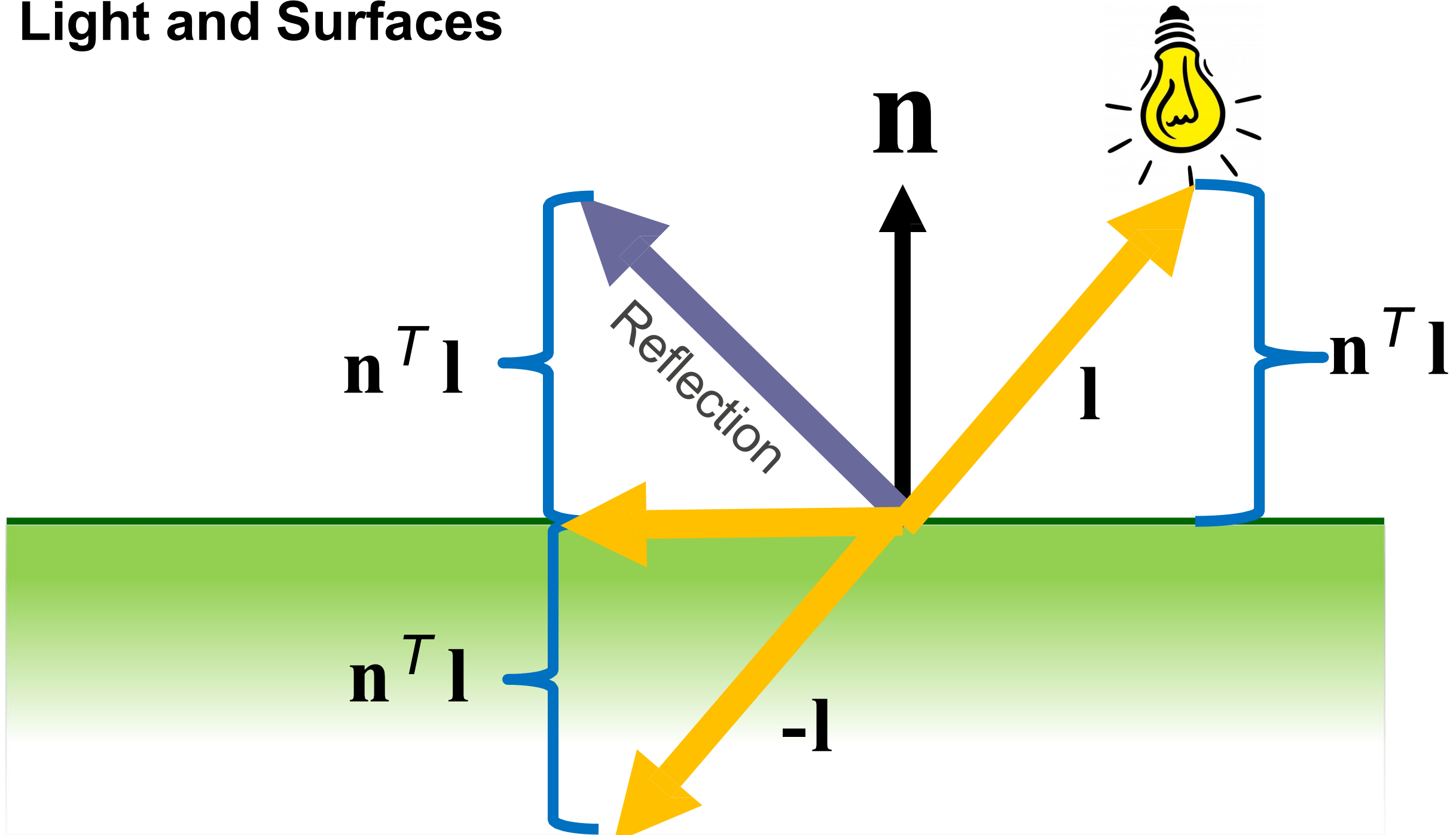
Light and Surfaces



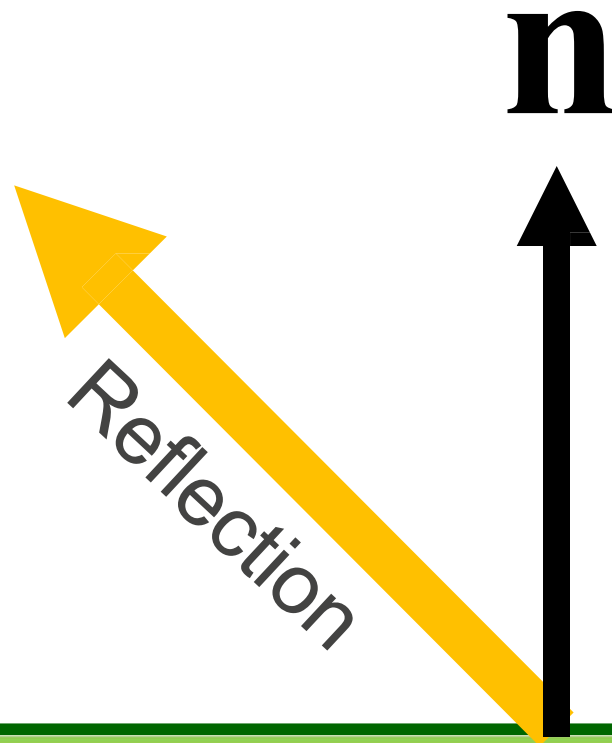
Light and Surfaces



Light and Surfaces



Light and Surfaces

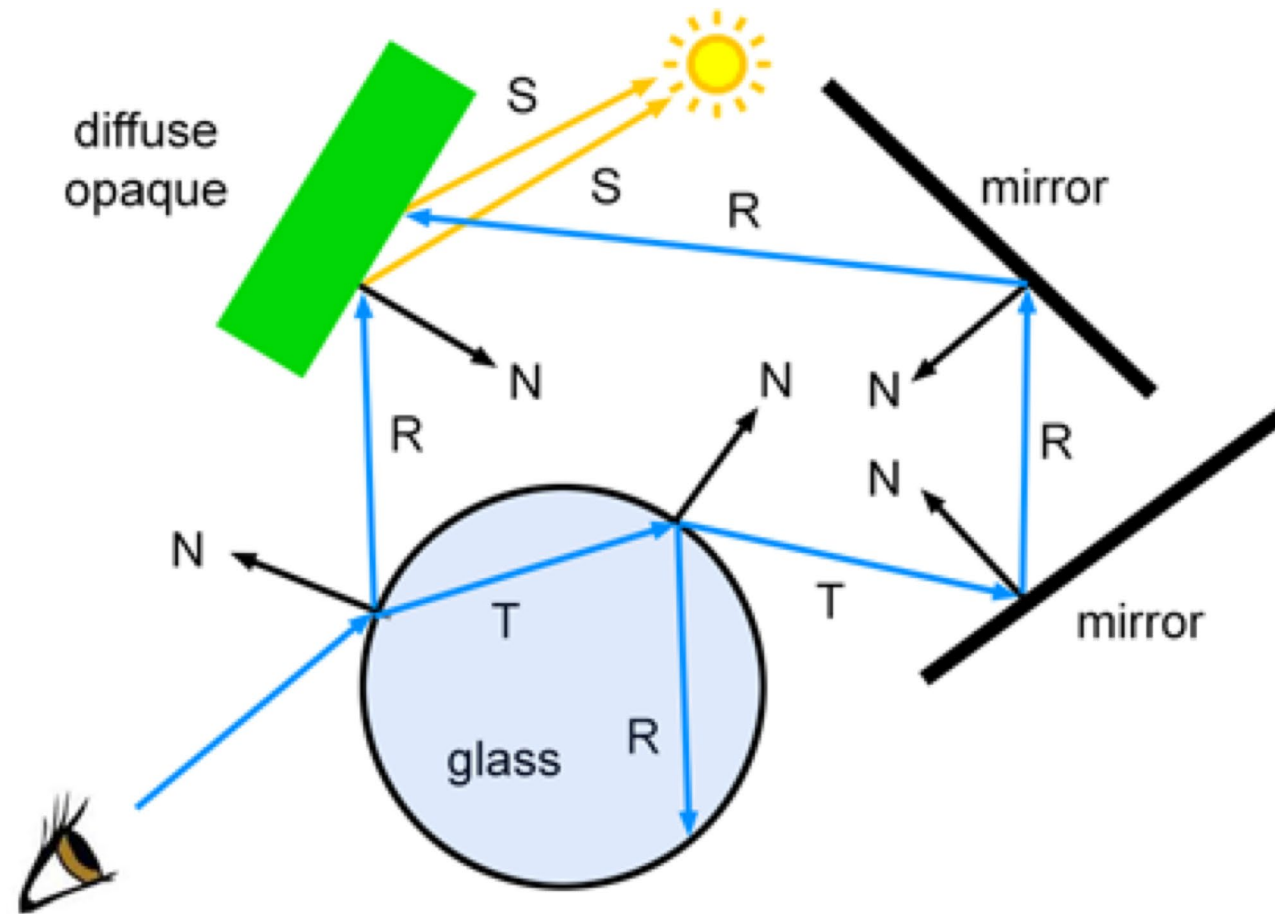


$$-I + 2nn^T I$$

```
for each pixel in the image {  
    pixel colour = rayTrace(viewRay, 0)  
}
```

```
colour rayTrace(Ray, depth) {  
    for each object in the scene {  
        if(Intersect ray with object) {  
            colour = shading model  
            if(depth < maxDepth)  
                colour +=rayTrace(reflectedRay,depth+1)  
        }  
    }  
    return colour  
}
```

Ray Spawning

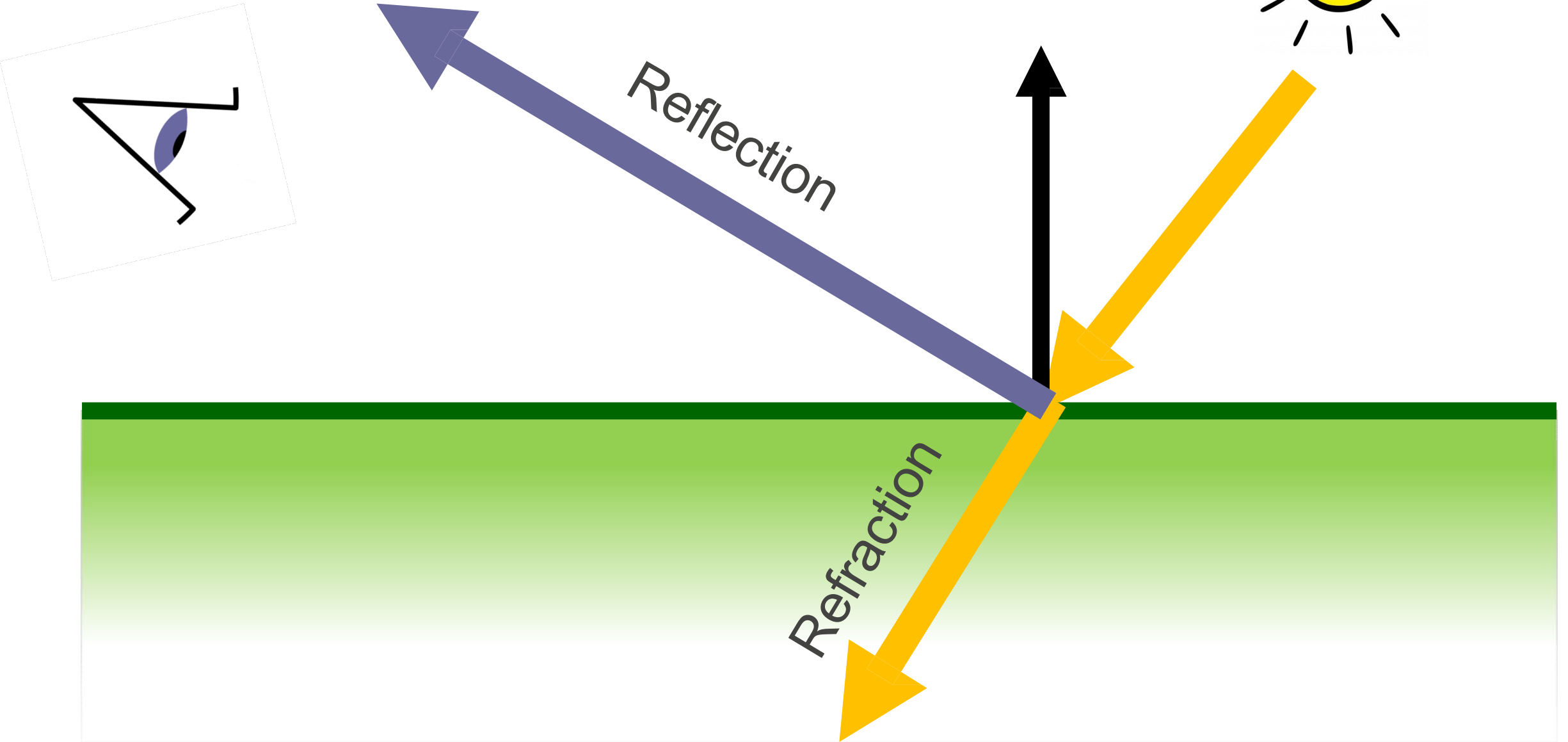


© www.scratchapixel.com

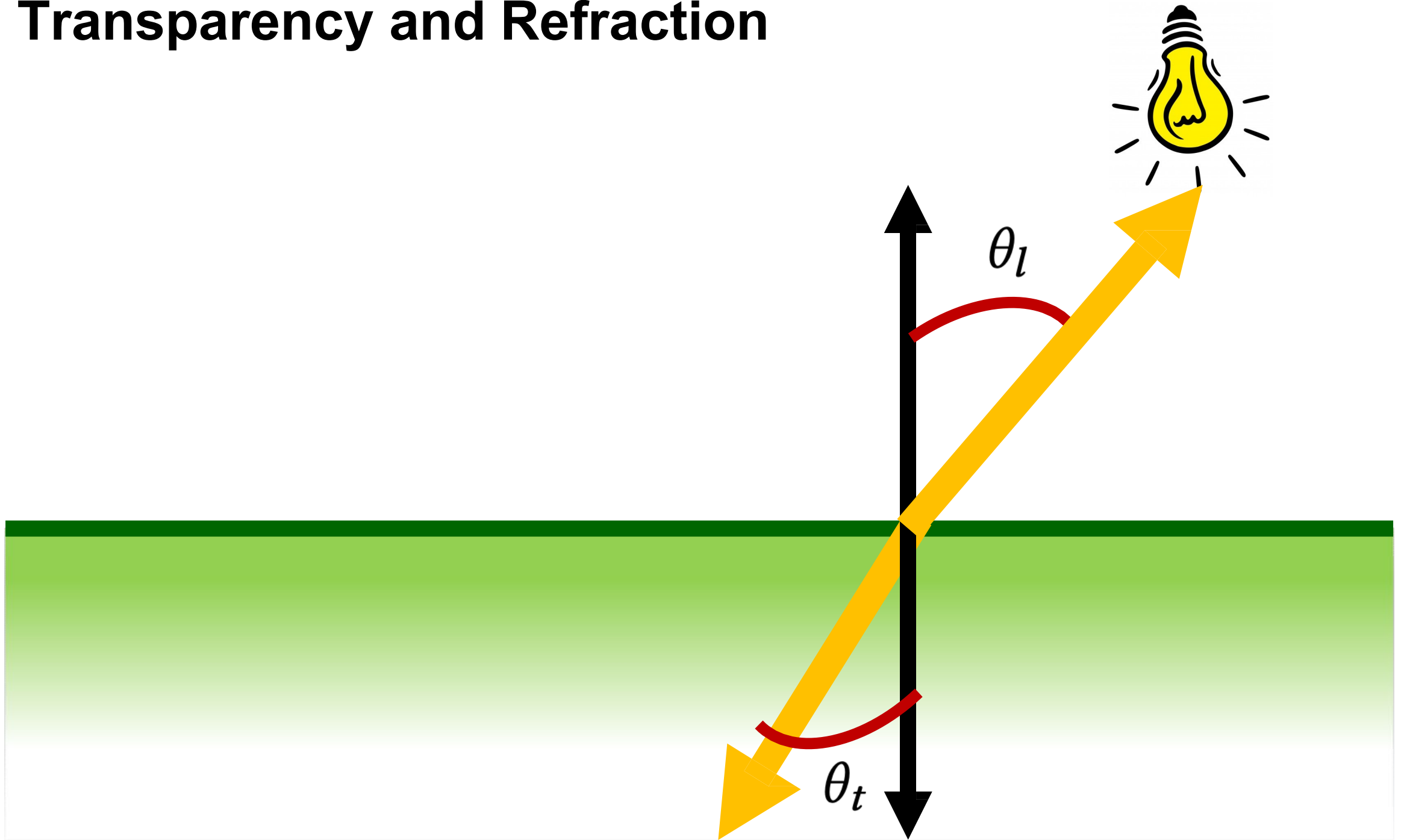
Transparency and Refraction



Transparency and Refraction



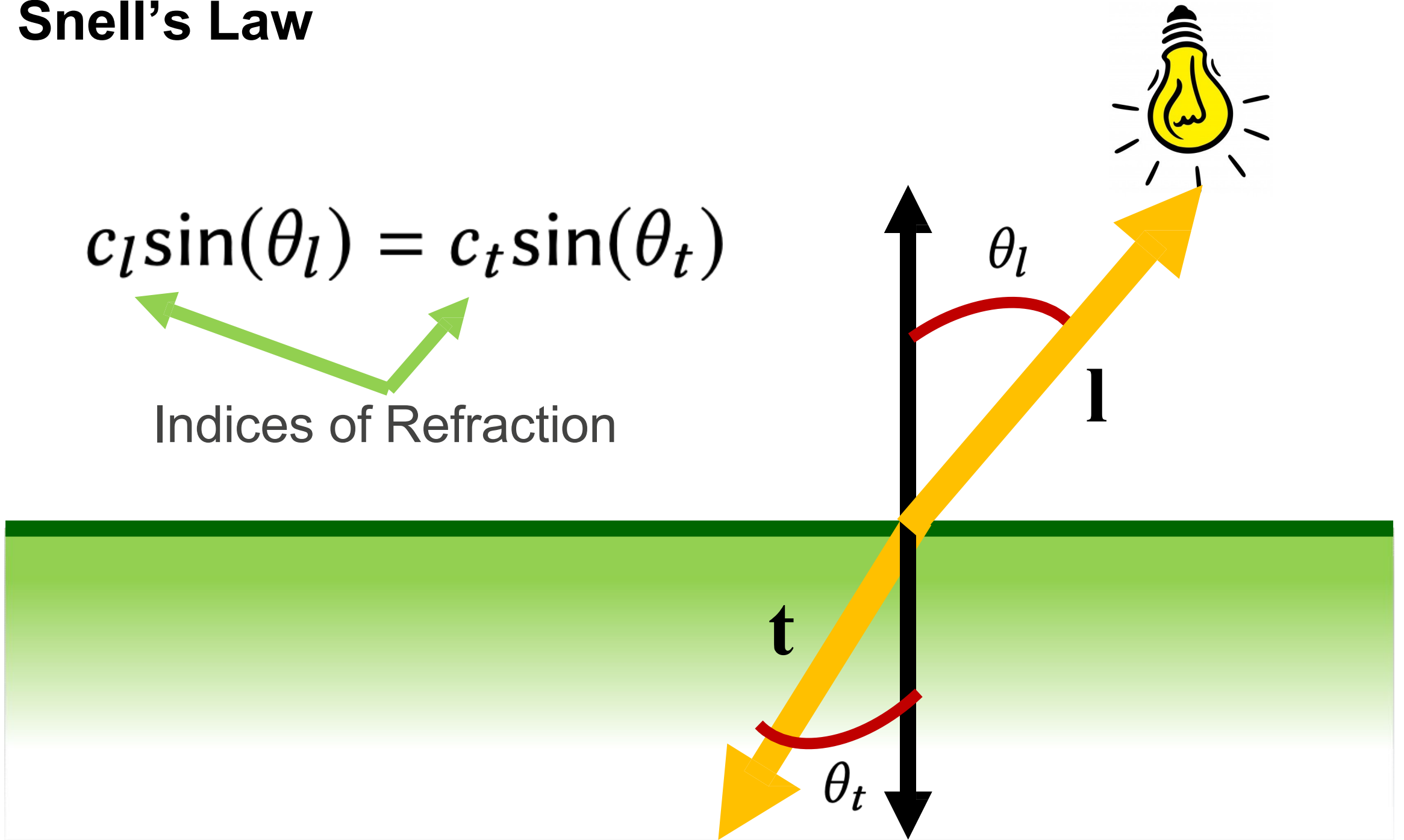
Transparency and Refraction



Snell's Law

$$c_l \sin(\theta_l) = c_t \sin(\theta_t)$$

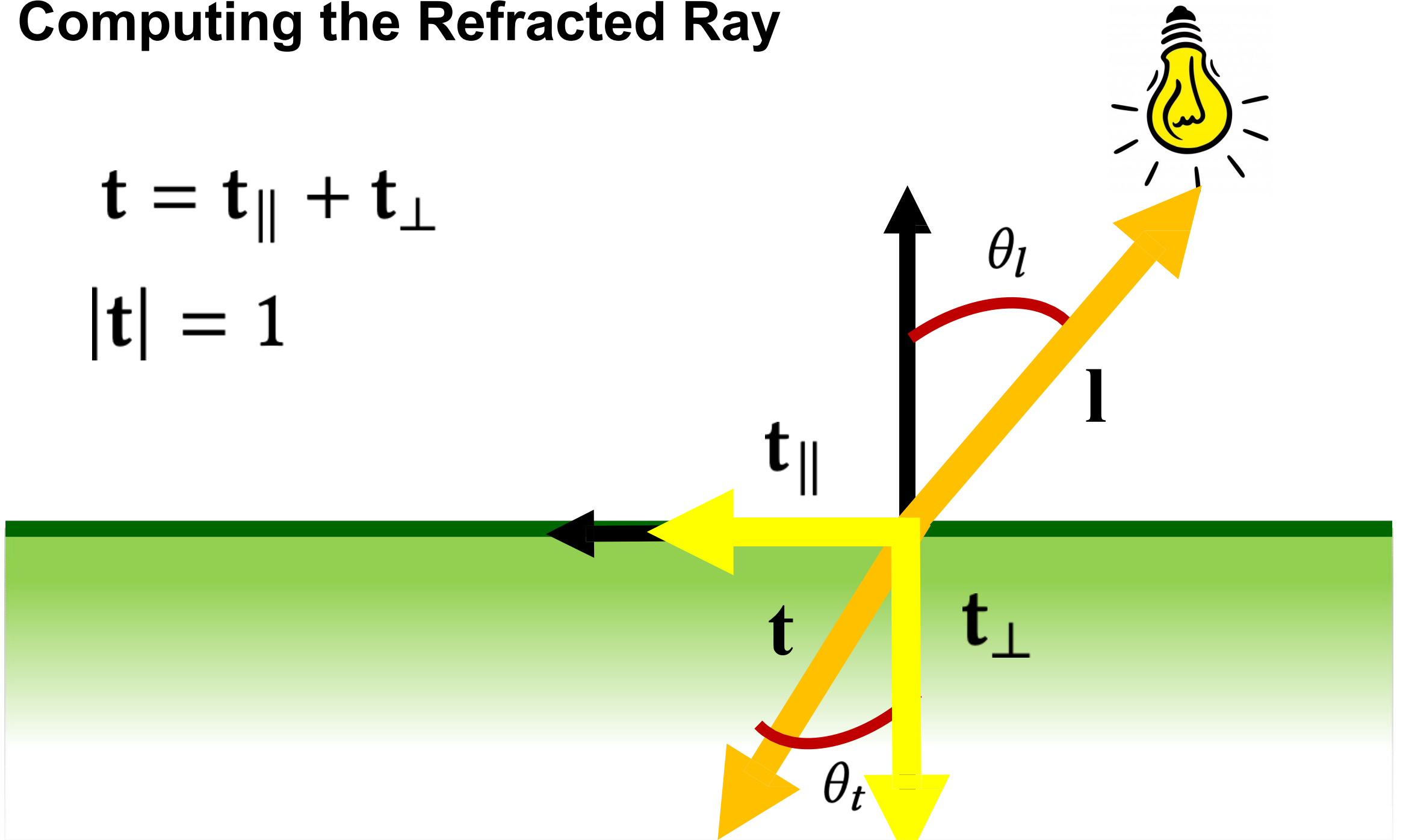
Indices of Refraction



Computing the Refracted Ray

$$\mathbf{t} = \mathbf{t}_{\parallel} + \mathbf{t}_{\perp}$$

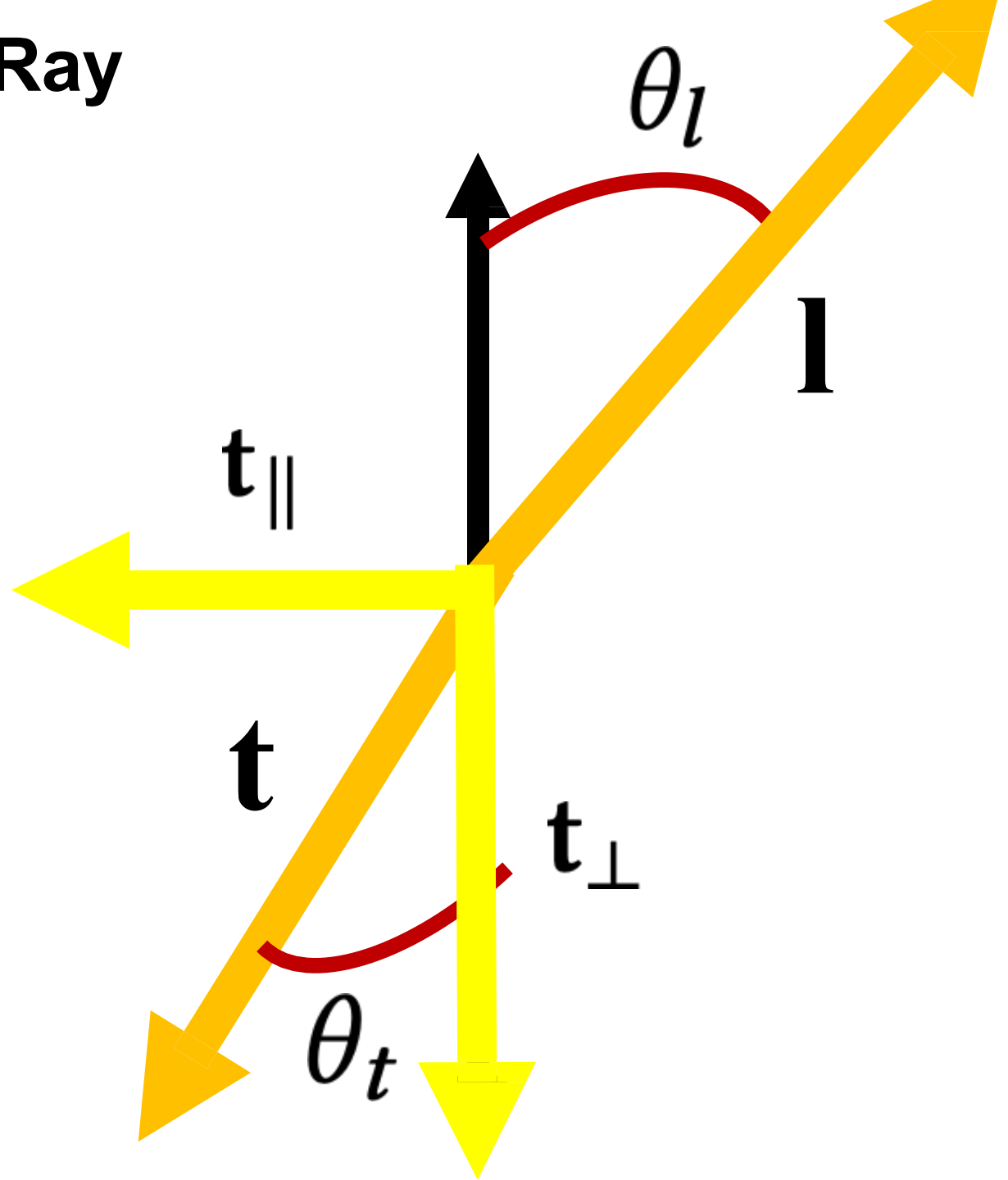
$$|\mathbf{t}| = 1$$



Computing the Refracted Ray

$$|\mathbf{t}_{\parallel}| = \sin(\theta_t)$$

$$|\mathbf{t}_{\parallel}| = \frac{c_l}{c_t} \sin(\theta_l)$$



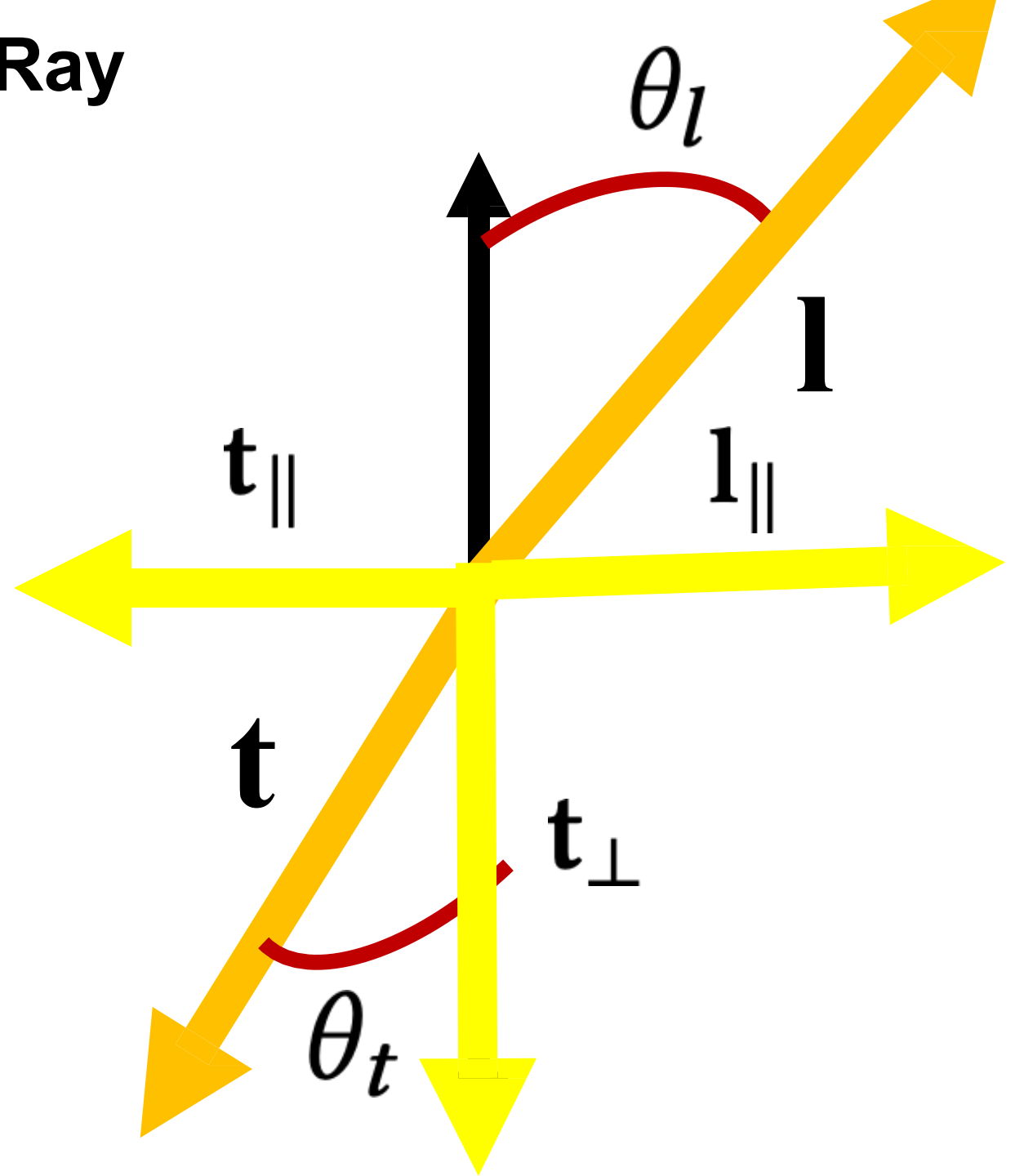
Computing the Refracted Ray

$$|\mathbf{t}_{||}| = \sin(\theta_t)$$

$$|\mathbf{t}_{||}| = \frac{c_l}{c_t} \sin(\theta_l)$$

$$|\mathbf{t}_{||}| = \frac{c_l}{c_t} |\mathbf{l}_{||}|$$

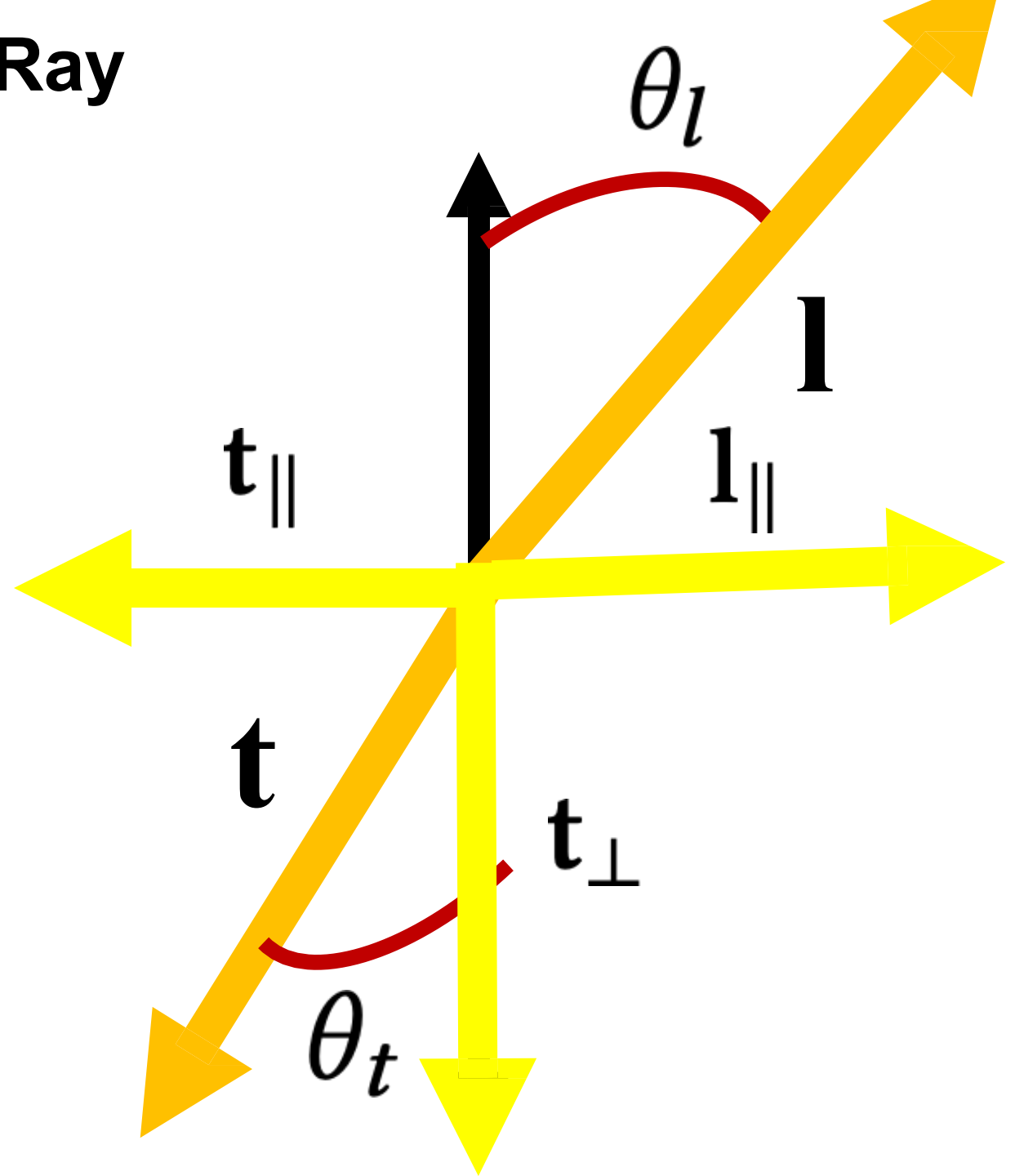
$$\mathbf{t}_{||} = -\frac{c_l}{c_t} \mathbf{l}_{||}$$



Computing the Refracted Ray

$$\mathbf{t}_{\parallel} = -\frac{c_l}{c_t} \mathbf{l}_{\parallel}$$

$$\mathbf{t}_{\parallel} = -\frac{c_l}{c_t} (\mathbf{l} - \mathbf{nn}^T \mathbf{l})$$

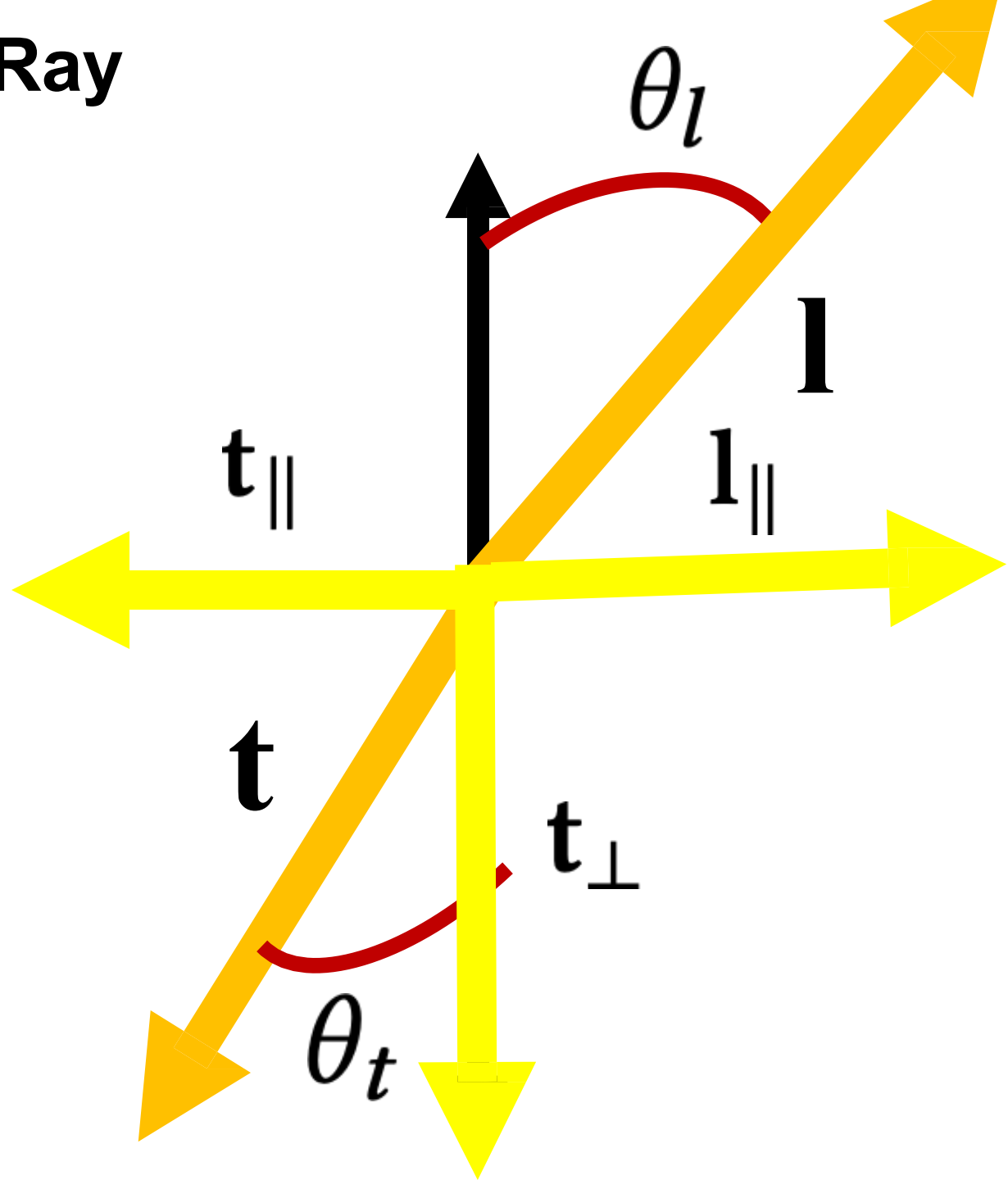


Computing the Refracted Ray

$$\mathbf{t}_{\parallel} = -\frac{c_l}{c_t} \mathbf{l}_{\parallel}$$

$$\mathbf{t}_{\parallel} = -\frac{c_l}{c_t} (\mathbf{l} - \mathbf{n} \mathbf{n}^T \mathbf{l})$$

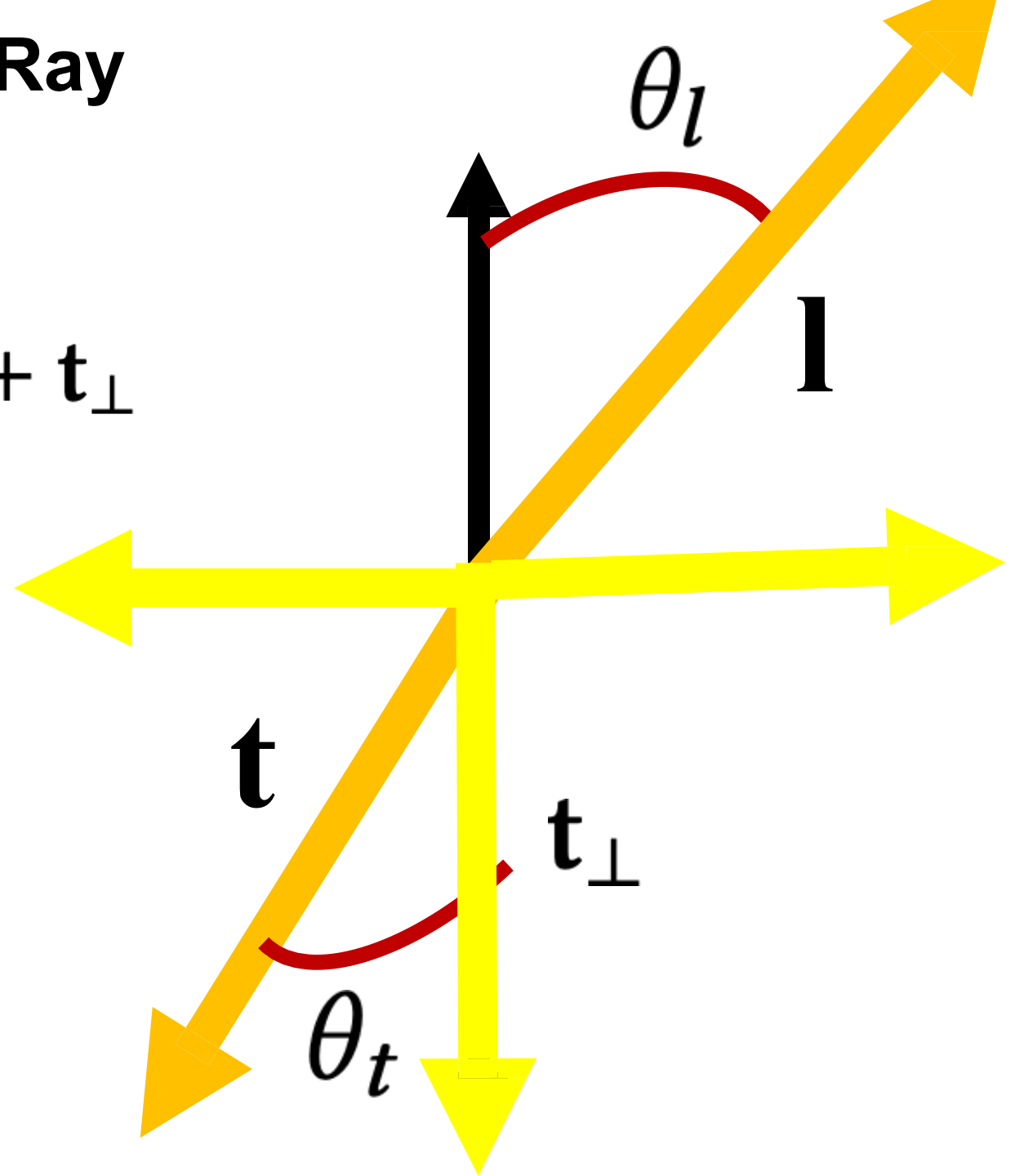
$$\mathbf{t}_{\parallel} = -\frac{c_l}{c_t} (1 - \cos(\theta_l)) \mathbf{n}$$



Computing the Refracted Ray

$$\mathbf{t} = \mathbf{t}_{\parallel} + \mathbf{t}_{\perp}$$

$$\mathbf{t} = -\frac{c_l}{c_t} \mathbf{l} + \frac{c_l}{c_t} \cos(\theta_l) \mathbf{n} + \mathbf{t}_{\perp}$$

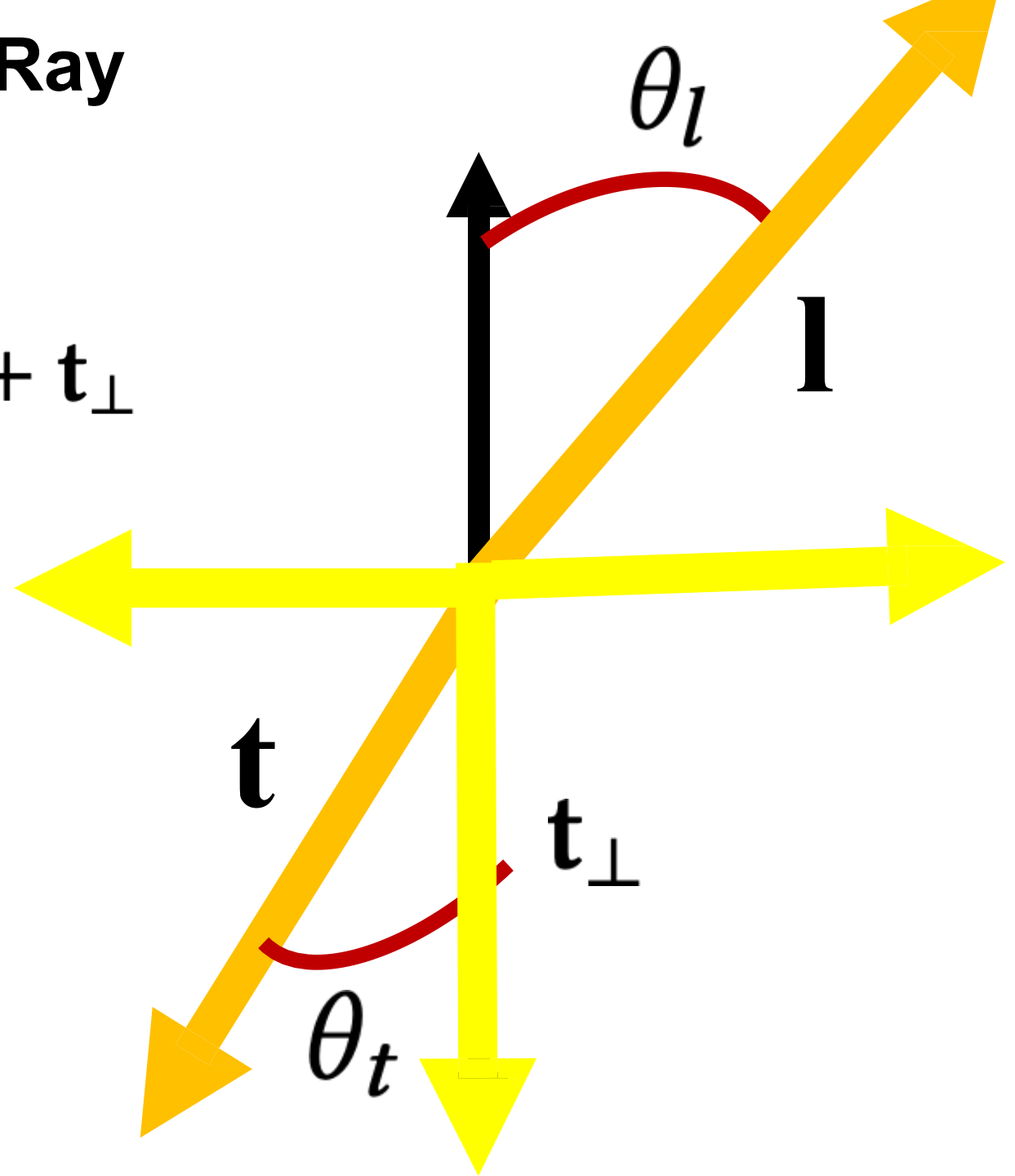


Computing the Refracted Ray

$$\mathbf{t} = \mathbf{t}_{\parallel} + \mathbf{t}_{\perp}$$

$$\mathbf{t} = -\frac{c_l}{c_t} \mathbf{l} + \frac{c_l}{c_t} \cos(\theta_l) \mathbf{n} + \mathbf{t}_{\perp}$$

$$\mathbf{t}_{\perp} = \alpha \mathbf{n}$$



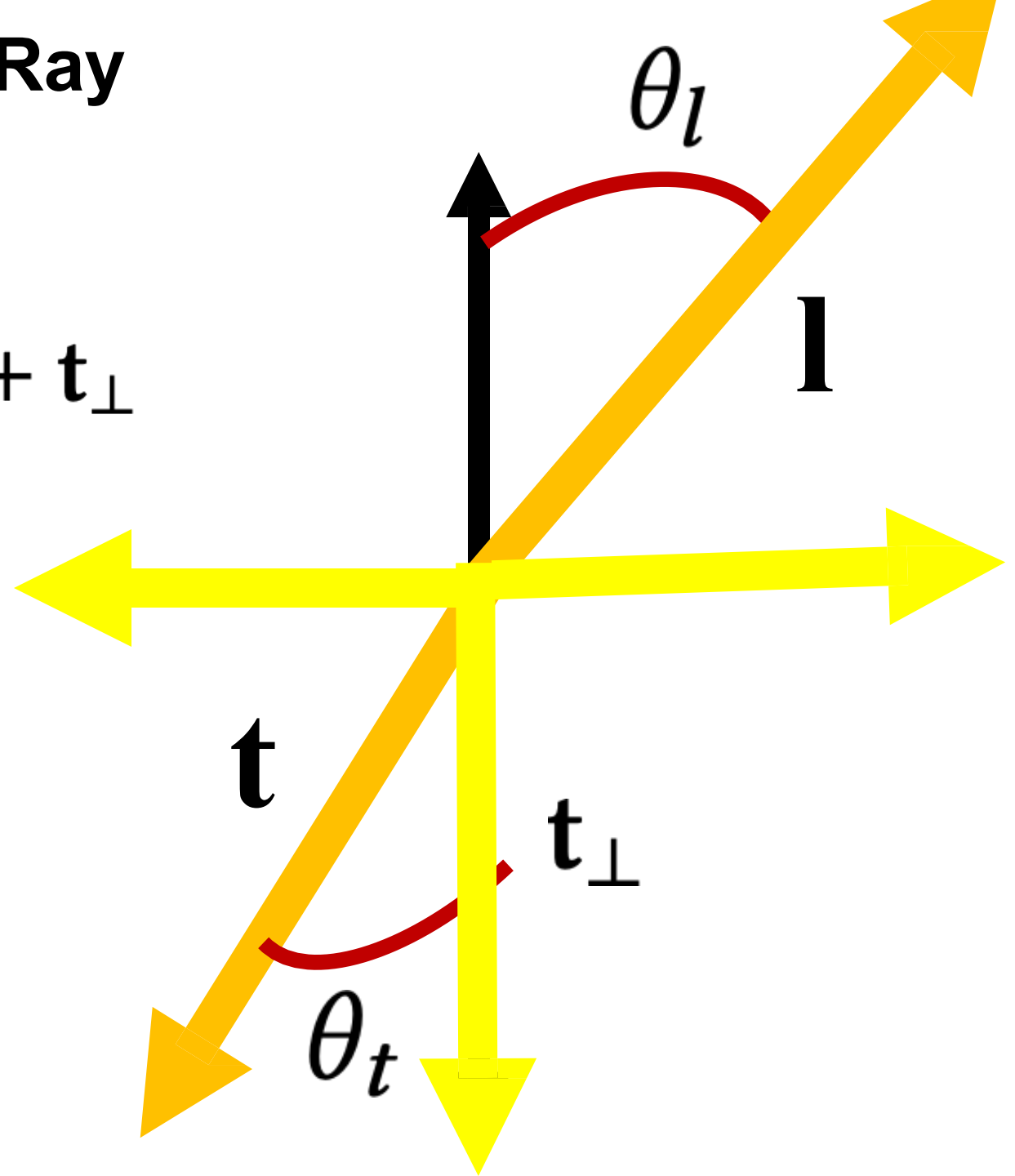
Computing the Refracted Ray

$$\mathbf{t} = \mathbf{t}_{\parallel} + \mathbf{t}_{\perp}$$

$$\mathbf{t} = -\frac{c_l}{c_t} \mathbf{l} + \frac{c_l}{c_t} \cos(\theta_l) \mathbf{n} + \mathbf{t}_{\perp}$$

$$\mathbf{t}_{\perp} = \alpha \mathbf{n}$$

$$\mathbf{t}_{\perp} = -\sqrt{1 - \sin^2 \theta_t} \mathbf{n}$$

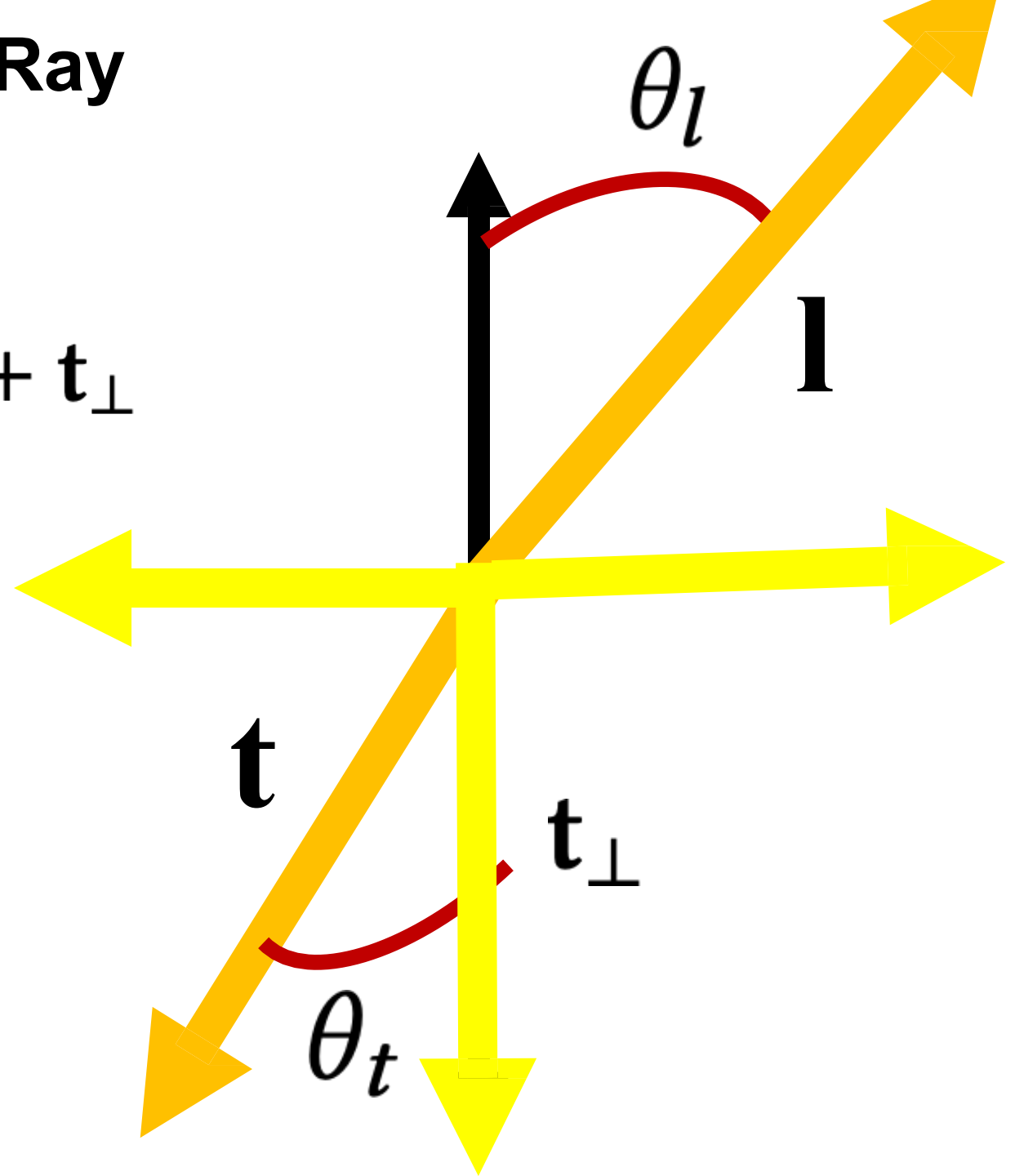


Computing the Refracted Ray

$$\mathbf{t} = \mathbf{t}_{\parallel} + \mathbf{t}_{\perp}$$

$$\mathbf{t} = -\frac{c_l}{c_t}\mathbf{l} + \frac{c_l}{c_t}\cos(\theta_l)\mathbf{n} + \mathbf{t}_{\perp}$$

$$\mathbf{t}_{\perp} = -\sqrt{1 - \sin^2\theta_t}\mathbf{n}$$



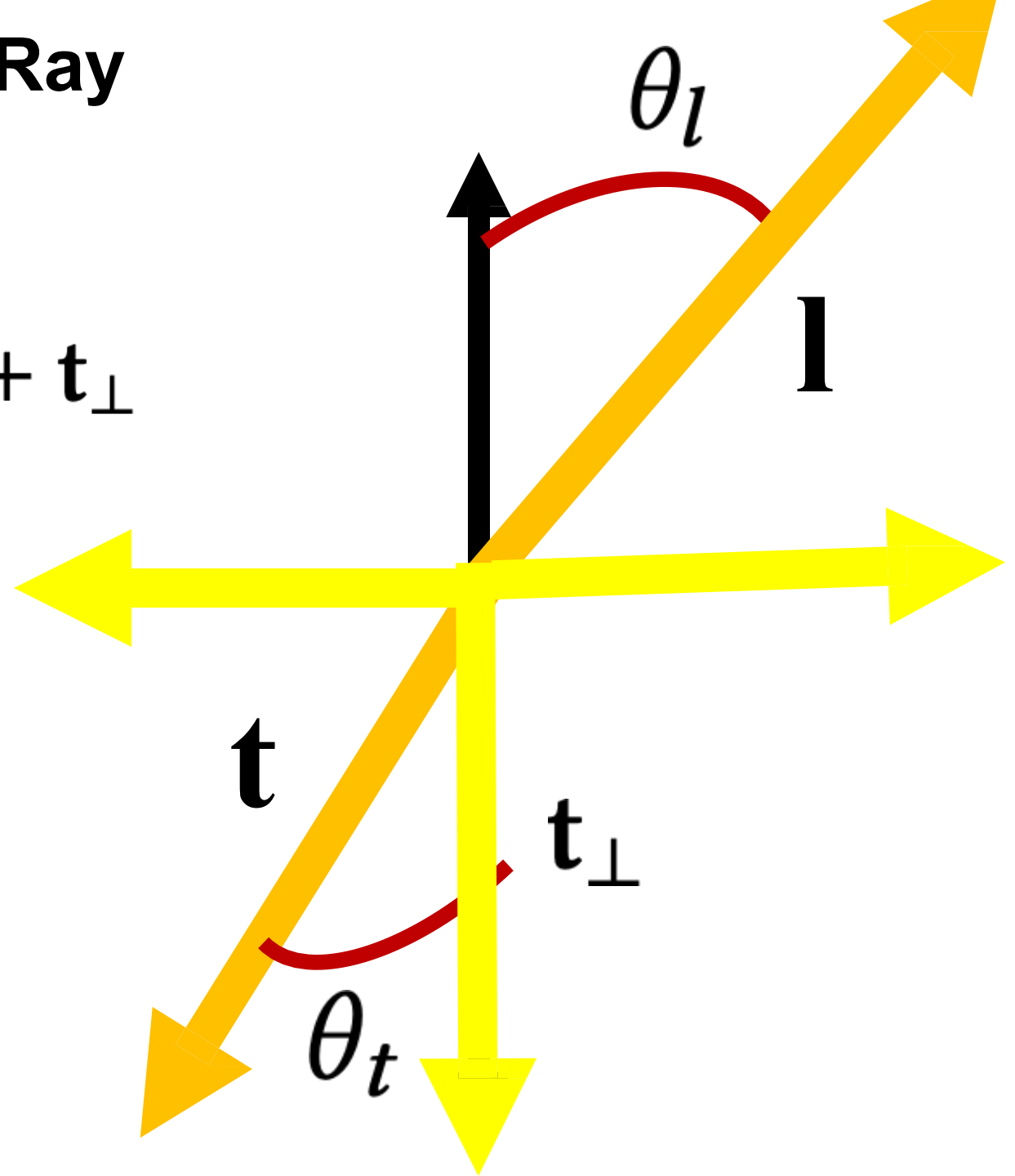
Computing the Refracted Ray

$$\mathbf{t} = \mathbf{t}_{\parallel} + \mathbf{t}_{\perp}$$

$$\mathbf{t} = -\frac{c_l}{c_t} \mathbf{l} + \frac{c_l}{c_t} \cos(\theta_l) \mathbf{n} + \mathbf{t}_{\perp}$$

$$\mathbf{t}_{\perp} = -\sqrt{1 - \sin^2 \theta_t} \mathbf{n}$$

$$\mathbf{t}_{\perp} = -\sqrt{\cos^2 \theta_t} \mathbf{n}$$



Computing the Refracted Ray

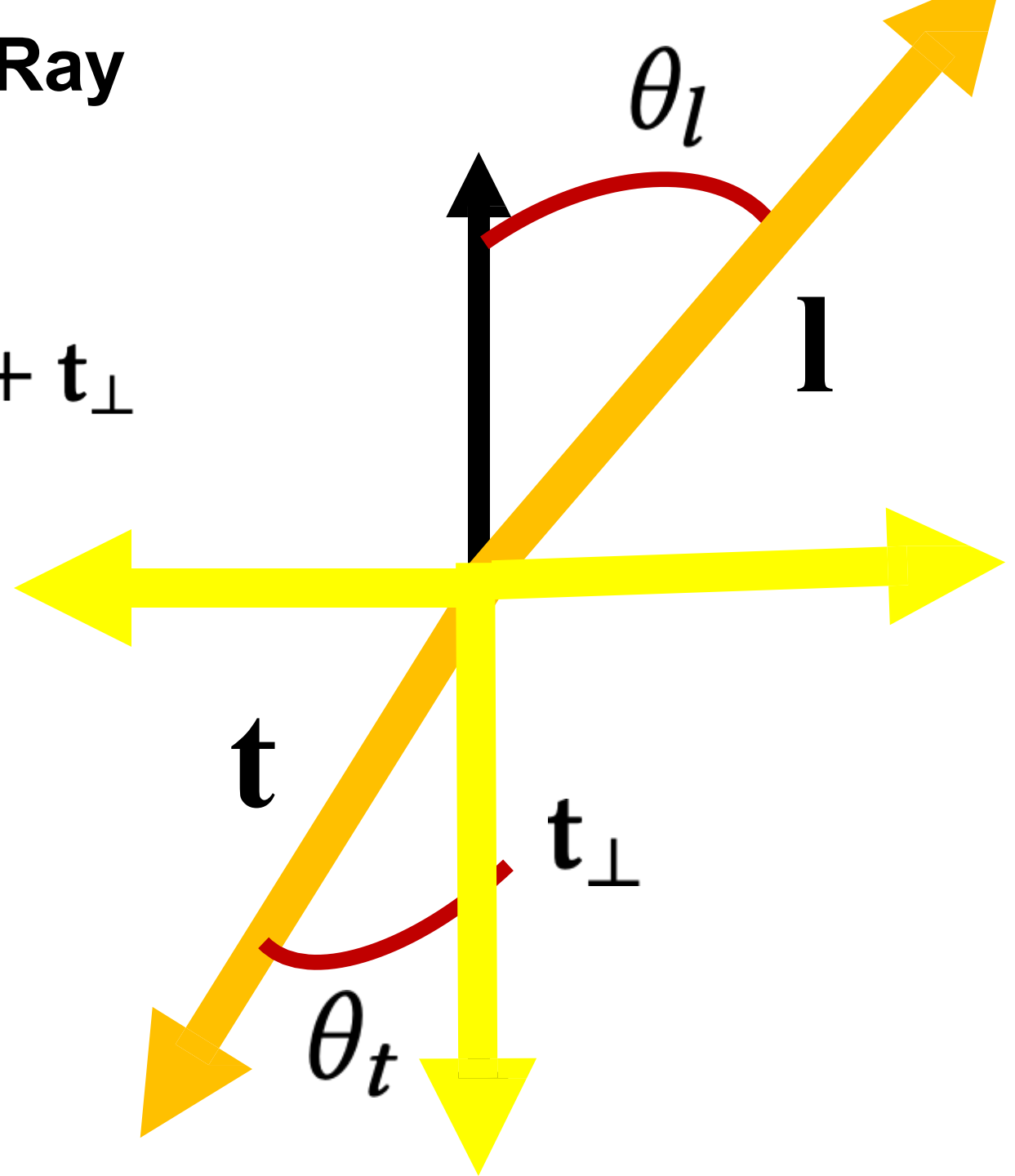
$$\mathbf{t} = \mathbf{t}_{\parallel} + \mathbf{t}_{\perp}$$

$$\mathbf{t} = -\frac{c_l}{c_t}\mathbf{l} + \frac{c_l}{c_t}\cos(\theta_l)\mathbf{n} + \mathbf{t}_{\perp}$$

$$\mathbf{t}_{\perp} = -\sqrt{1 - \sin^2\theta_t}\mathbf{n}$$

$$\mathbf{t}_{\perp} = -\sqrt{\cos^2\theta_t}\mathbf{n}$$

$$\mathbf{t}_{\perp} = -\cos\theta_t\mathbf{n}$$



Computing the Refracted Ray

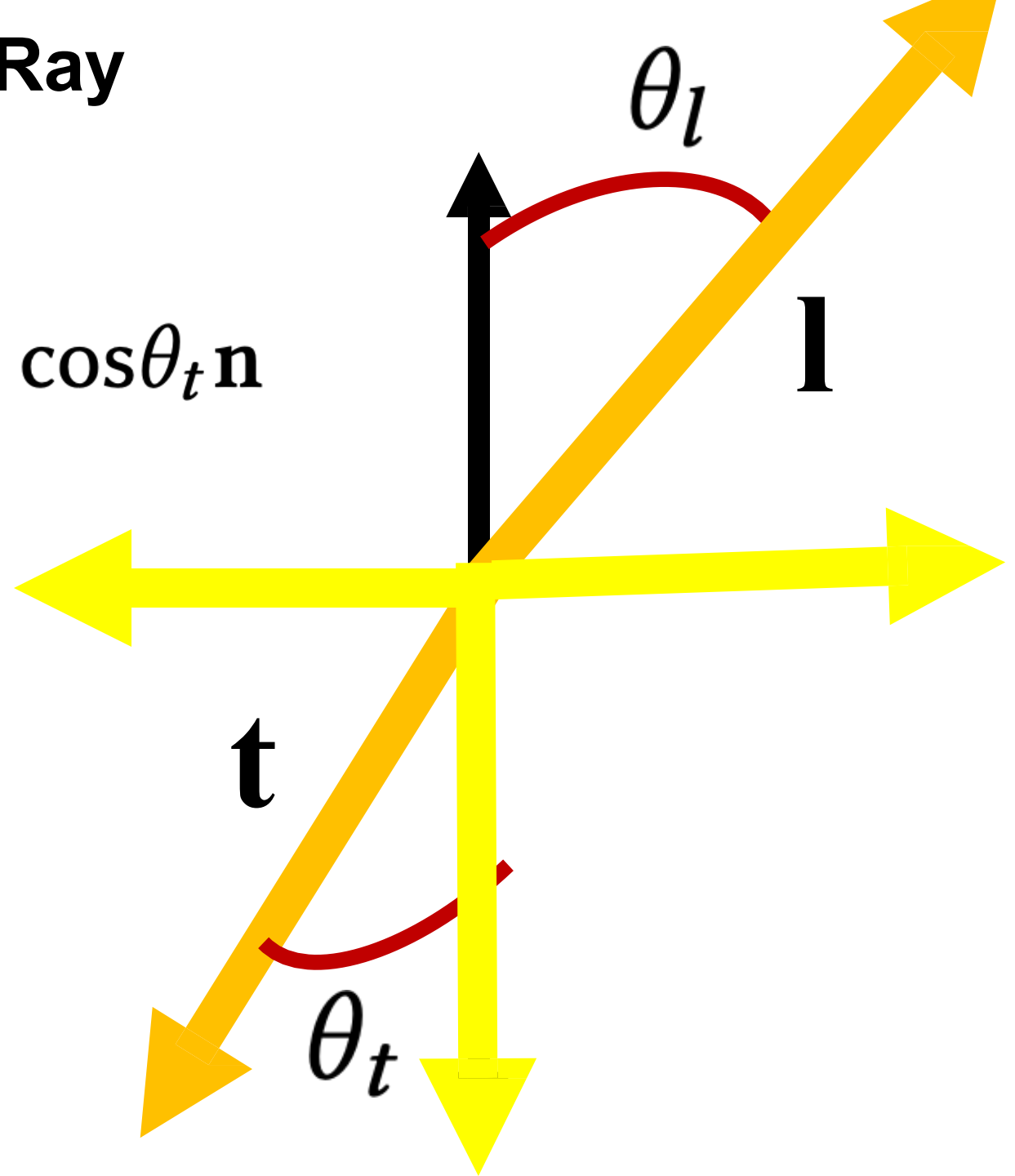
$$\mathbf{t} = \mathbf{t}_{\parallel} + \mathbf{t}_{\perp}$$

$$\mathbf{t} = -\frac{c_l}{c_t} \mathbf{l} + \frac{c_l}{c_t} \cos(\theta_l) \mathbf{n} - \cos \theta_t \mathbf{n}$$

$$\mathbf{t}_{\perp} = -\sqrt{1 - \sin^2 \theta_t} \mathbf{n}$$

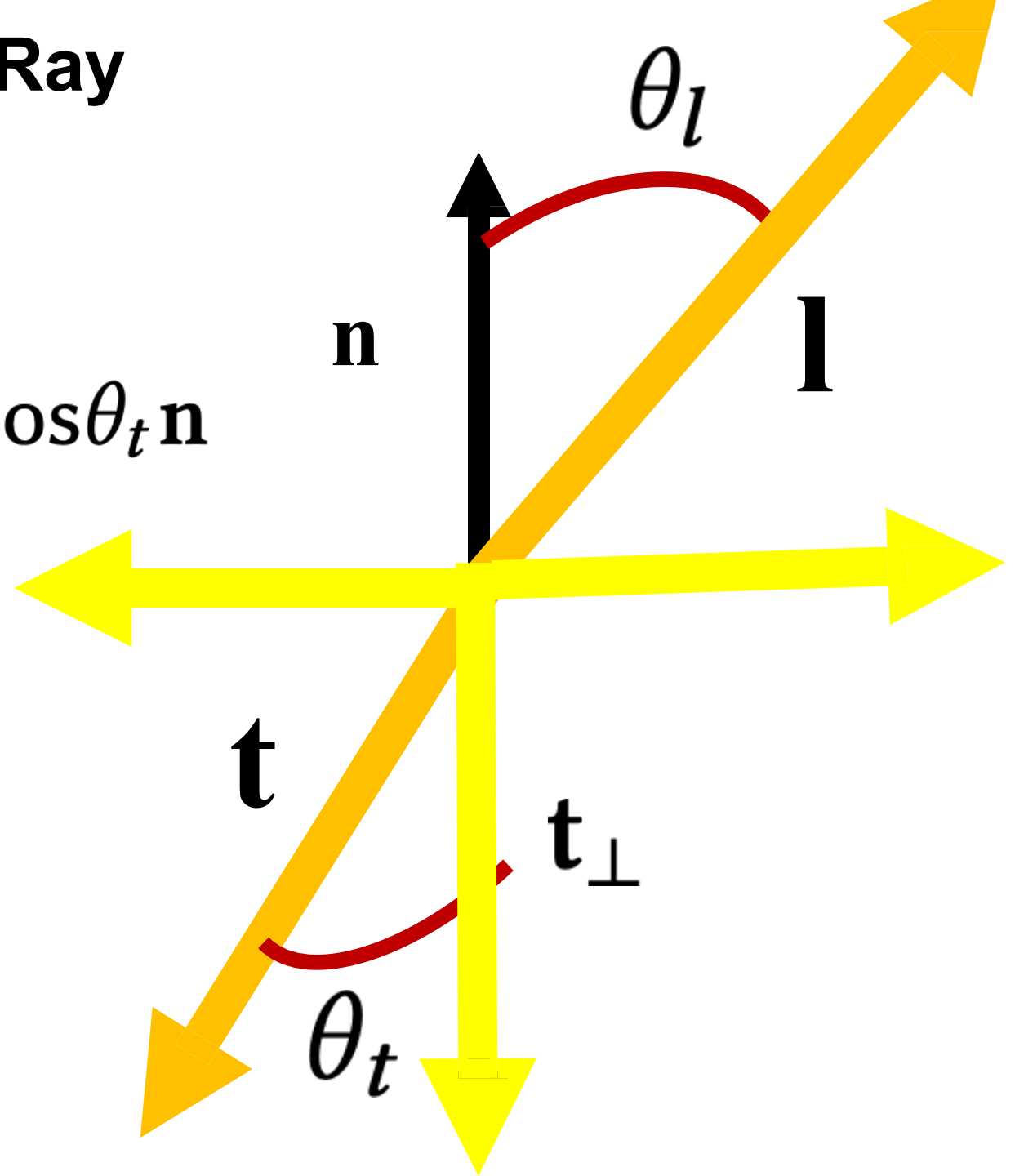
$$\mathbf{t}_{\perp} = -\sqrt{\cos^2 \theta_t} \mathbf{n}$$


$$\mathbf{t}_{\perp} = -\cos \theta_t \mathbf{n}$$



Computing the Refracted Ray

$$\mathbf{t} = -\frac{c_l}{c_t}\mathbf{l} + \frac{c_l}{c_t}\cos(\theta_l)\mathbf{n} - \cos\theta_t\mathbf{n}$$

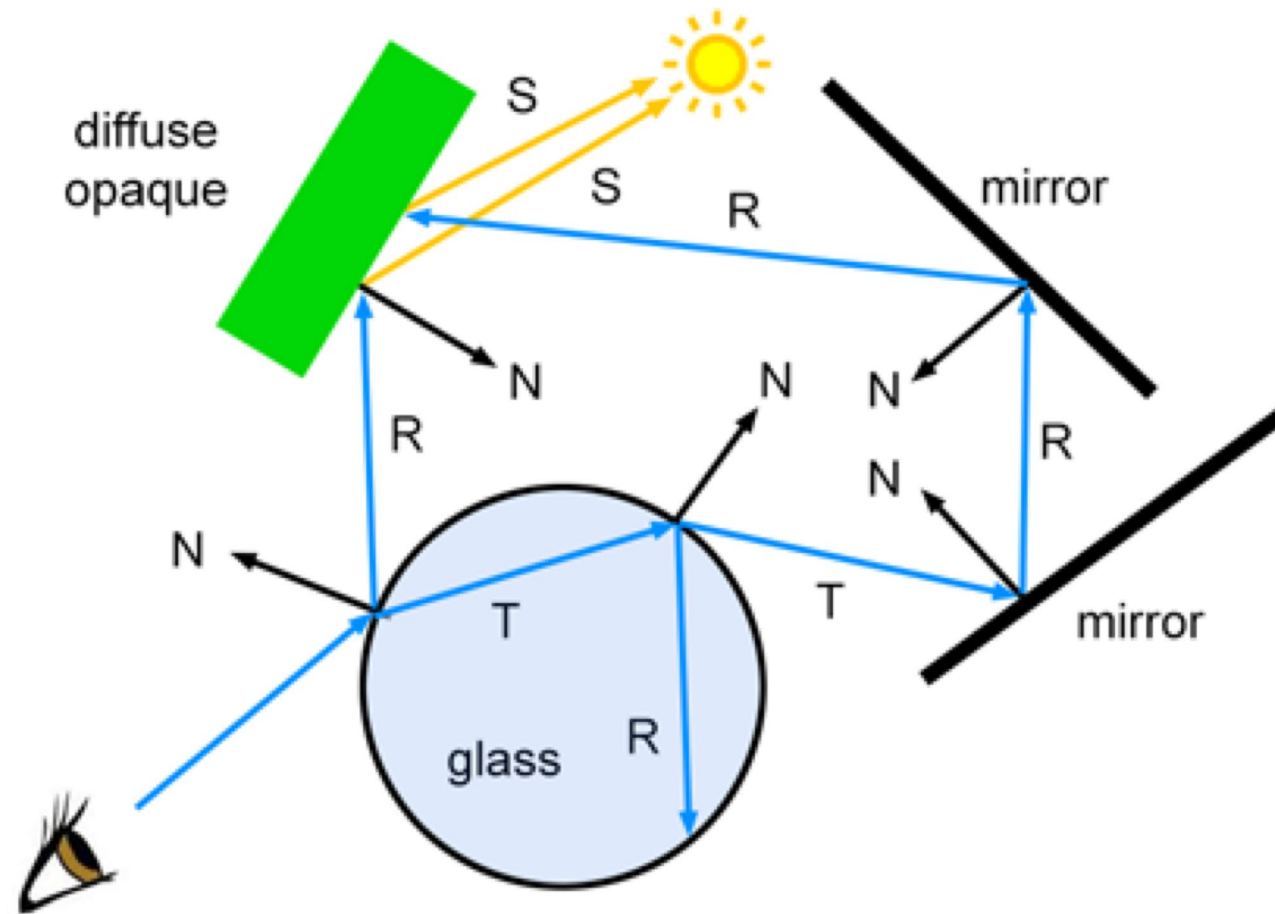


```
colour rayTrace(Ray, depth) {  
    for each object in the scene {  
        if(Intersect ray with object) {  
            colour = shading model  
            if(depth < maxDepth) {  
                colour +=   
rayTrace(reflectedRay, depth+1)  
                colour +=  
rayTrace(refractedRay, depth+1)  
            }  
        }  
    }  
    return colour  
}
```

```
colour rayTrace(Ray, depth) {  
    for each object in the scene {  
        if(Intersect ray with object) {  
            colour = shading model  
            if(depth < maxDepth) {  
                colour +=  
rayTrace(reflectedRay, depth+1)  
                colour +=  
rayTrace(refractedRay, depth+1)  
            }  
        }  
    }  
    return colour  
}
```



Ray Spawning



© www.scratchapixel.com

Any Questions?