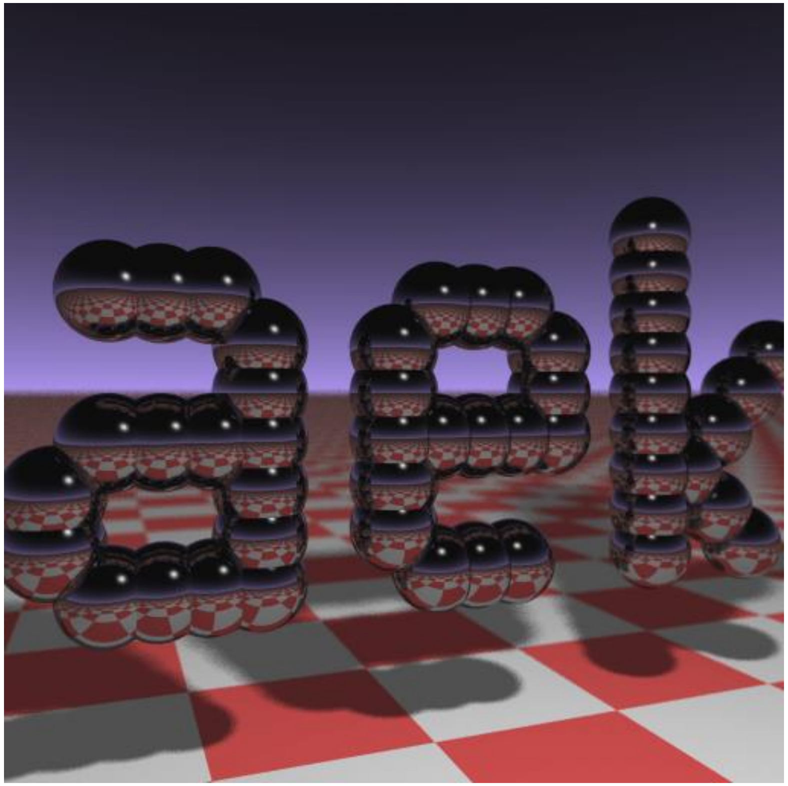
Reflection and Transmission

Requires Minimal Implementation

```
#include <stdlib.h>   // card > aek.ppm
#include <stdio.h>
#include <math.h>
typedef int i;typedef float f;struct v{
f x,y,z;v operator+(v r){return v(x+r.x
,y+r.y,z+r.z);}v operator*(f r){return
v(x*r,y*r,z*r);}f operator%(v r){return
x*r.x+y*r.y+z*r.z;}v(){}v operator^(v r
){return v(y*r.z-z*r.y,z*r.x-x*r.z,x*r.
y-y*r.x);}v(f a,f b,f c){x=a;y=b;z=c;}v
operator!(){return*this*(1/sqrt(*this%*
this));}};i G[]={247570,280596,280600,
249748,18578,18577,231184,16,16};f R(){
return(f)rand()/RAND_MAX;}i T(v o,v d,f
&t,v&n){t=1e9;i m=0;f p=-o.z/d.z;if(.01
<p)t=p,n=v(0,0,1),m=1;for(i k=19;k--;)
for(i j=9;j--;)if(G[j]&1<<k){v p=o+v(-k
,0,-j-4);f b=p%d,c=p%p-1,q=b*b-c;if(q>0
){f s=-b-sqrt(q);if(s<t&&s>.01)t=s,n=!(
p+d*t),m=2;}}return m;}v S(v o,v d){f t
;v n;i m=T(o,d,t,n);if(!m)return v(.7,
.6,1)*pow(1-d.z,4);v h=o+d*t,l=!(v(9+R(
),9+R(),16)+h*-1),r=d+n*(n%d*-2);f b=l%
n;if(b<0||T(h,l,t,n))b=0;f p=pow(l%r*(b
>0),99);if(m&1){h=h*.2;return((i)(ceil(
h.x)+ceil(h.y))&1?v(3,1,1):v(3,3,3))*(b
*.2+.1);}return v(p,p,p)+S(h,r)*.5;}i
main(){printf("P6 512 512 255 ");v g=!v
(-6,-16,0),a=!(v(0,0,1)^g)*.002,b=!(g^a
)*.002,c=(a+b)*-256+g;for(i y=512;y--;)
for(i x=512;x--;){v p(13,13,13);for(i r
=64;r--;){v t=a*(R()-.5)*99+b*(R()-.5)*
99;p=S(v(17,16,8)+t,!(t*-1+(a*(R()+x)+b
*(y+R())+c)*16))*3.5+p;}printf("%c%c%c"
,(i)p.x,(i)p.y,(i)p.z);}}
```



http://fabiensanglard.net/rayTracing_back_of_business_card/

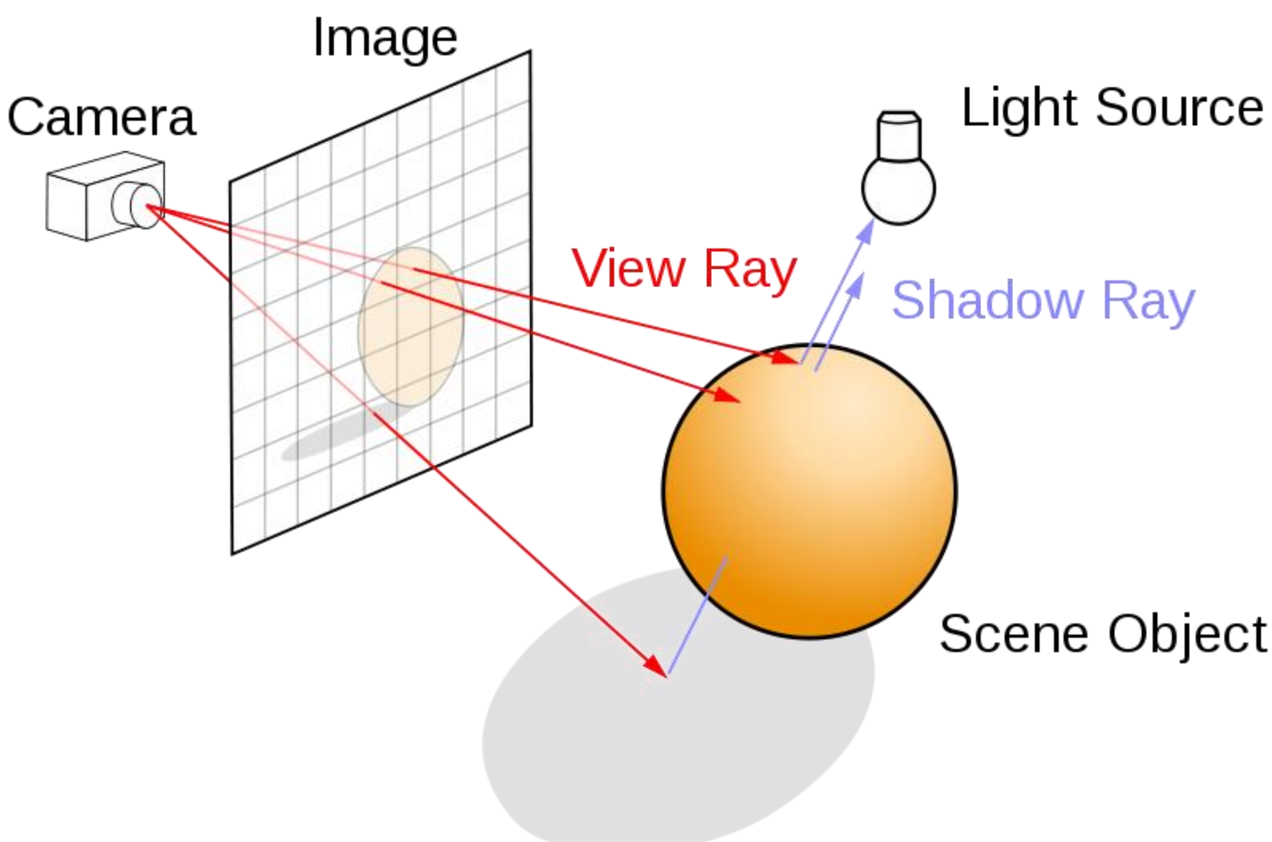
Embarrassingly Parallel

- Can easily implement on a GPU
- Can easily use all the cores on a CPU
- Can easily use many machines across a network, or on the cloud

http://www.youtube.com/watch?v=-P28LKWTzrI

Recall: Ray Tracing

- Generate an image by backwards tracing the path of light through pixels on an image plane
- Simulate the interaction of light with objects



Recall: Basic Ray Tracer

- Trace a primary ray from the eye through each pixel and detect the first intersection point with the objects in the scene
- Trace secondary shadow rays from the intersection point towards light sources and sum the contributions from each (visible) light

- The lighting equation:

$$L = \sum_{j \epsilon lights} L_i^j \left(k_a + k_d(\hat{\mathbf{N}} \cdot \omega_i^j)_+ + k_s(\hat{\mathbf{N}} \cdot \hat{\mathbf{H}}_i^j)_+^s\right)$$

ambient diffuse specular

sum all lights

Recursive Ray Tracing

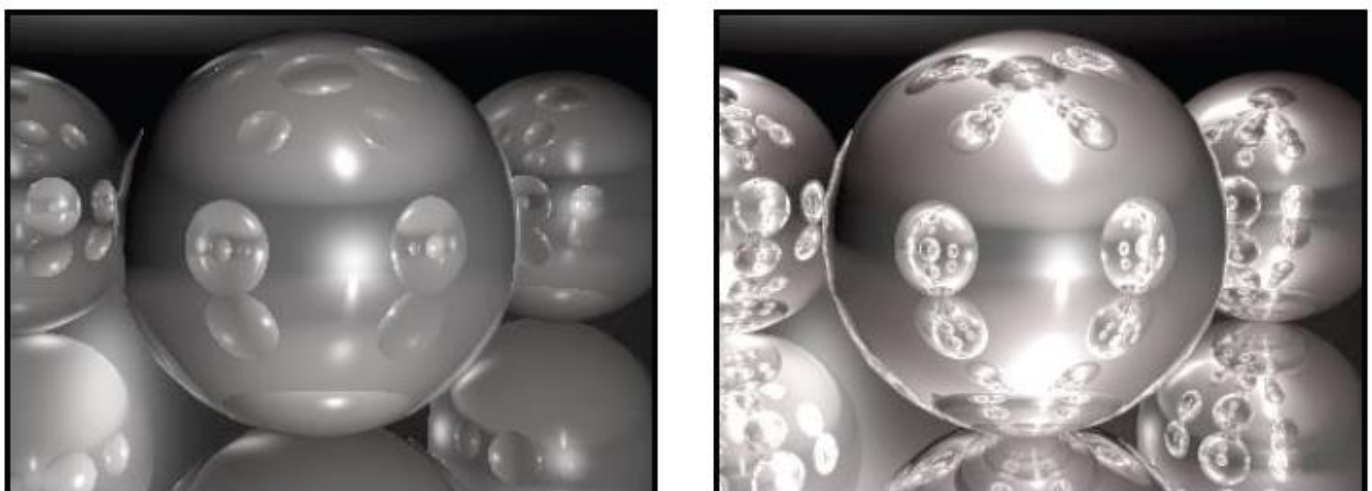
- Light at a point may not just come directly from light sources:
  - Light can bounce off other objects (reflection)
  - Light can pass through objects (transparency/refraction)
- Trace more rays to look for more lighting information
  - Send secondary rays from the intersection point
  - Recursively compute the color for these secondary rays and sum them onto the primary ray
- Lighting equation becomes:

$$L = \underbrace{\sum_{j \epsilon lights} L_i^j (k_a + k_d(\hat{\mathbf{N}} \cdot \omega_i^j)_+ + k_s(\hat{\mathbf{N}} \cdot \hat{\mathbf{H}}_i^j)_+^s)}_{\text{standard direct illumination}} + \underbrace{k_r L_{reflected}}_{\text{recursively computed reflection}} + \underbrace{k_t L_{transmitted}}_{\text{recursively computed refraction}}$$

Reflection/Transmission Adds Light

- Adding additional reflection/transmission terms adds more light to the object making it <u>brighter</u>
- Can offset this by dialing down the object color
- Can offset this by dialing down the reflection/transmission coefficients, k_r/k_r, but this can overly dim the reflections/transmissions

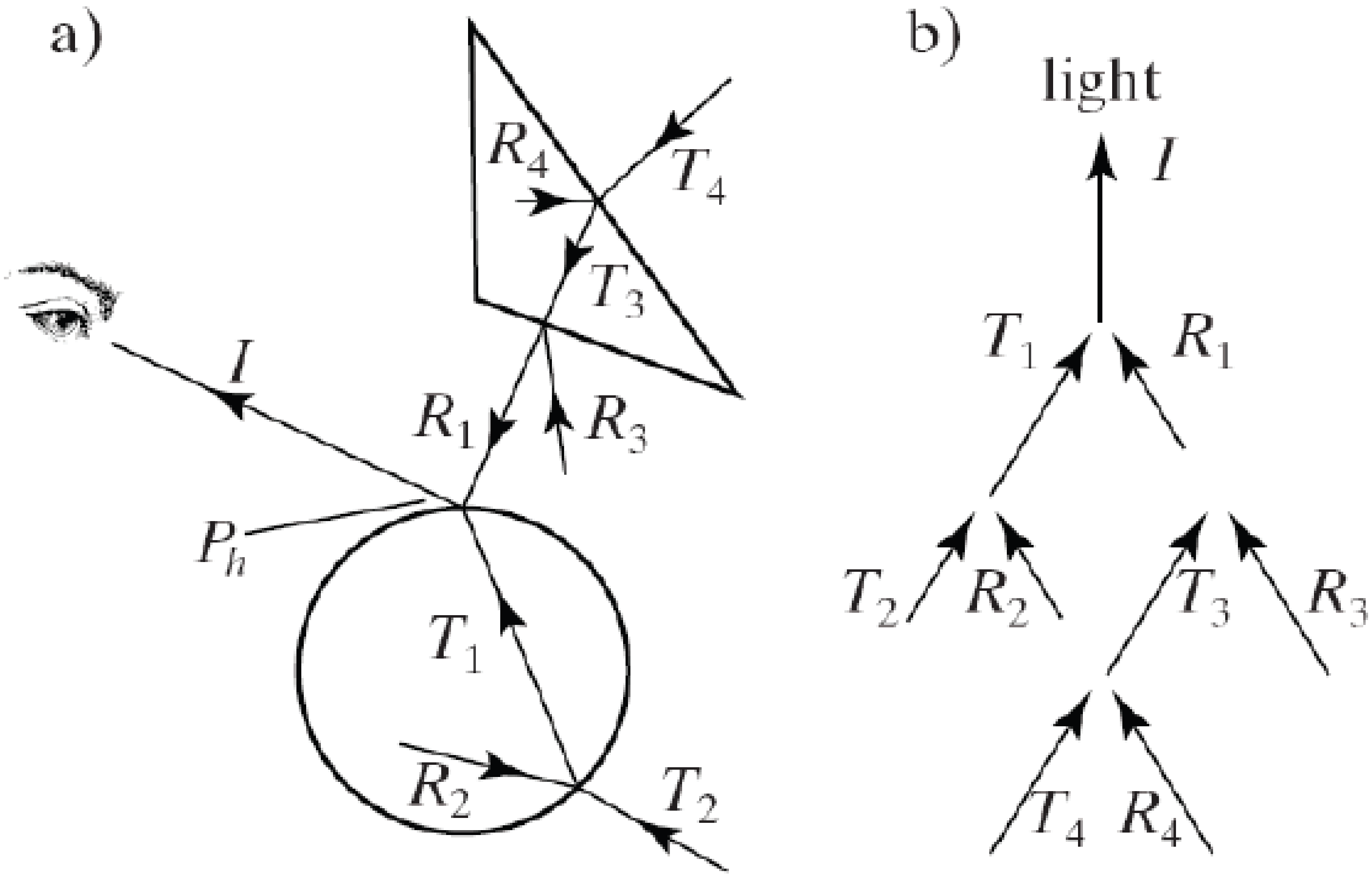
- Recommended approach: adjust the ratio between the object color and the reflected/transmitted light first to get the desired “look,” and then tune everything up and down to adjust brightness while maintaining this ratio



additional reflection makes an object brighter

Ray Tree

- Each reflected or transmitted ray may spawn its own shadow, reflection, and refraction/transmission rays, which is a recursive process



Maximum Depth

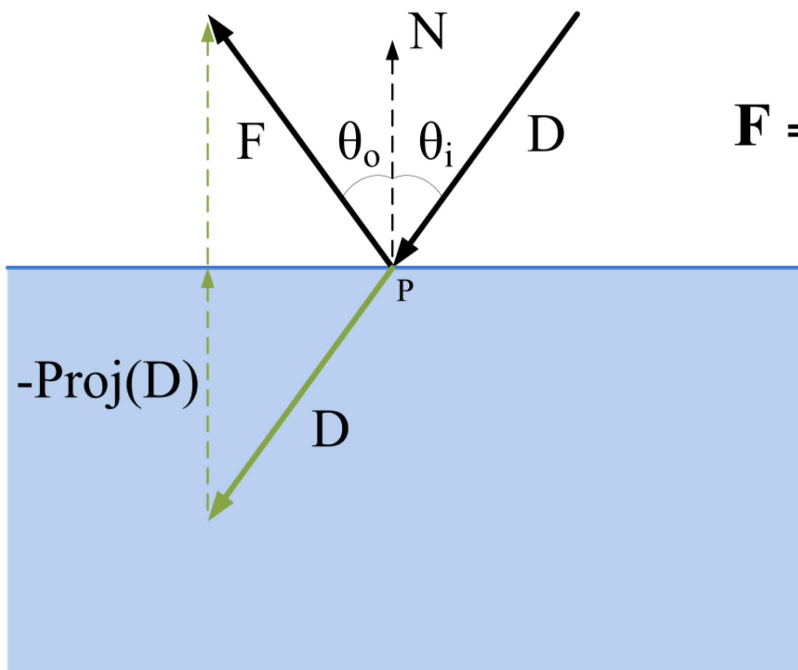
- If a ray tree is too deep (too many rays), it gets too costly/slow
  - N.B. be aware of the size of the recursion stack (which depends on the hardware) to avoid recursive stack overflow
- If each splitting in the tree has a direct lighting component, then rays deeper in the tree make a smaller and smaller contribution to the total light for a pixel
  - So one can truncate when the contribution is below a threshold

- In some cases (mirrors, etc.), there may not be a direct lighting component and 100% of the light at a split comes from reflected/transmitted rays
- In this case premature pruning will adversely affect the image
  - One typically carefully chooses a default color for the early termination point
  - Often this default color is the background color of the scene (used when a ray does not hit anything in the scene)
  - E.g., the background color could be blue if your scene is under the sky
  - If a default color is not set, simply truncating the rays behaves as if a black color was set, i.e. multiplying the discarded rays by 0

Reflection

Reflective Ray

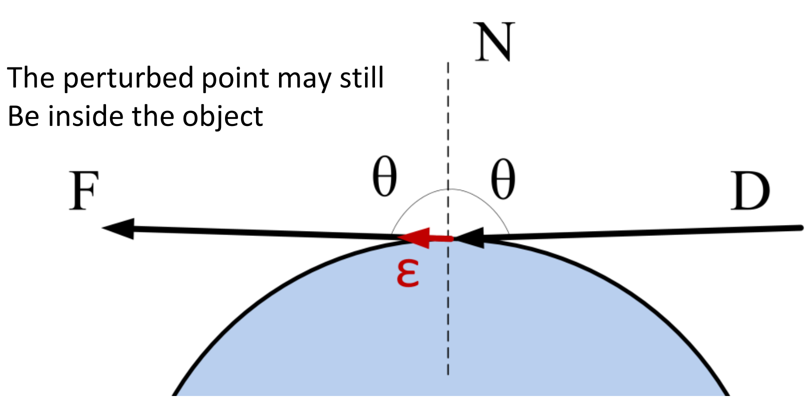
- Given an incident ray $R(t) = A+tD$, the reflective ray $R_{ref}(t) = P + tF$ is computed as follows:
  - **P** is the point where R(t) intersects the surface
  - **F** is the reflected vector of **D** with respect to the surface normal **N**
    - The incident angle θ_i equals the reflection angle θ_o



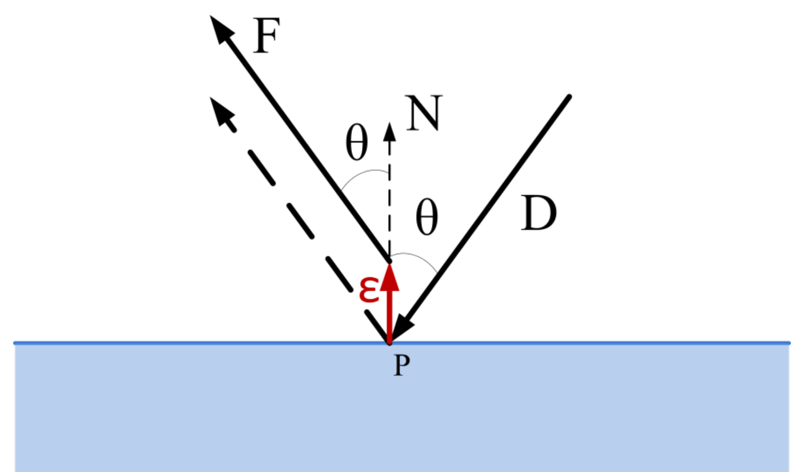
$$\mathbf{F} = \mathbf{D} - 2(\mathbf{D} \bullet \mathbf{N})\mathbf{N}$$

Spurious Self-Occlusion

- Add ε to the starting point of a reflected ray to avoid re-intersection with the original surface
- This often fails for grazing rays near the objects silhouette
- Or offset in the normal direction from the surface
  - Just like shadow rays, except here we preserve the direction of the reflected ray
- Avoid placing the new starting point too close to or inside other nearby objects



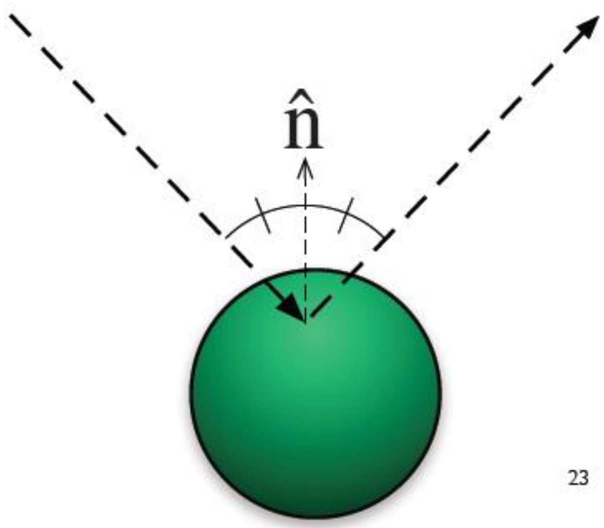
offset in the ray direction

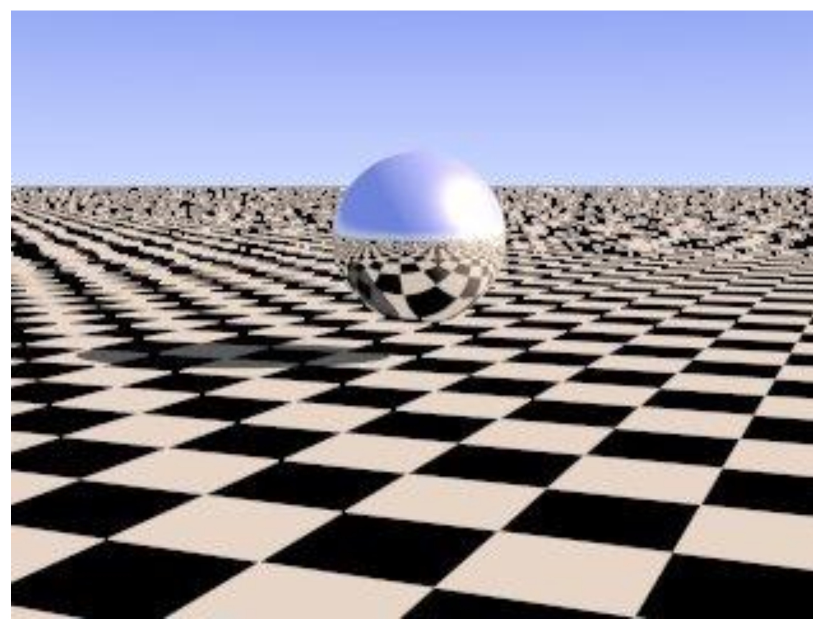


offset in the normal direction

Shading Reflections

- Shade recursively
  - Treat reflected rays like primary rays from the camera
  - Shade the reflected ray and return its color
    - Multiply by the reflection coefficient k_r
  - Add the resulting color to the shading at the original point

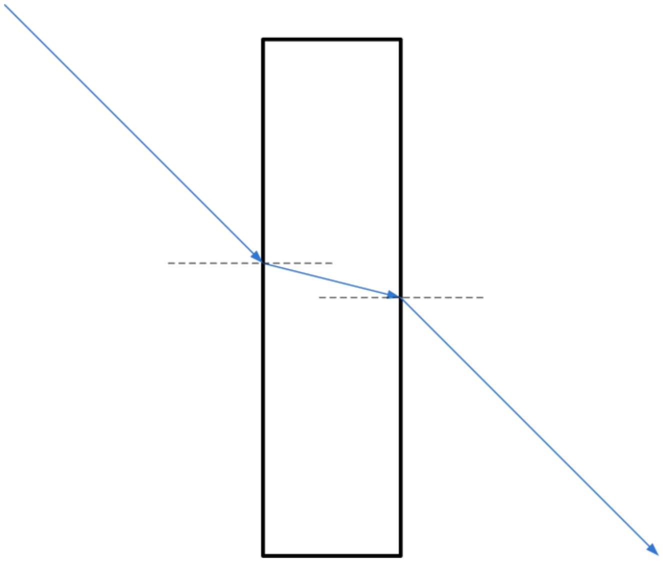




Transmission

Shading Transmission

- If the object is transparent, trace a transmitted ray
  - Treat transmitted rays like primary rays
    - Add ε to avoid self intersection, or offset in the <u>negative</u> normal direction (respecting collisions, other geometry, etc.)
  - Shade the transmitted ray and return its color
  - Multiply by the refraction coefficient k_t
  - Add the resulting color to the shading at the original point

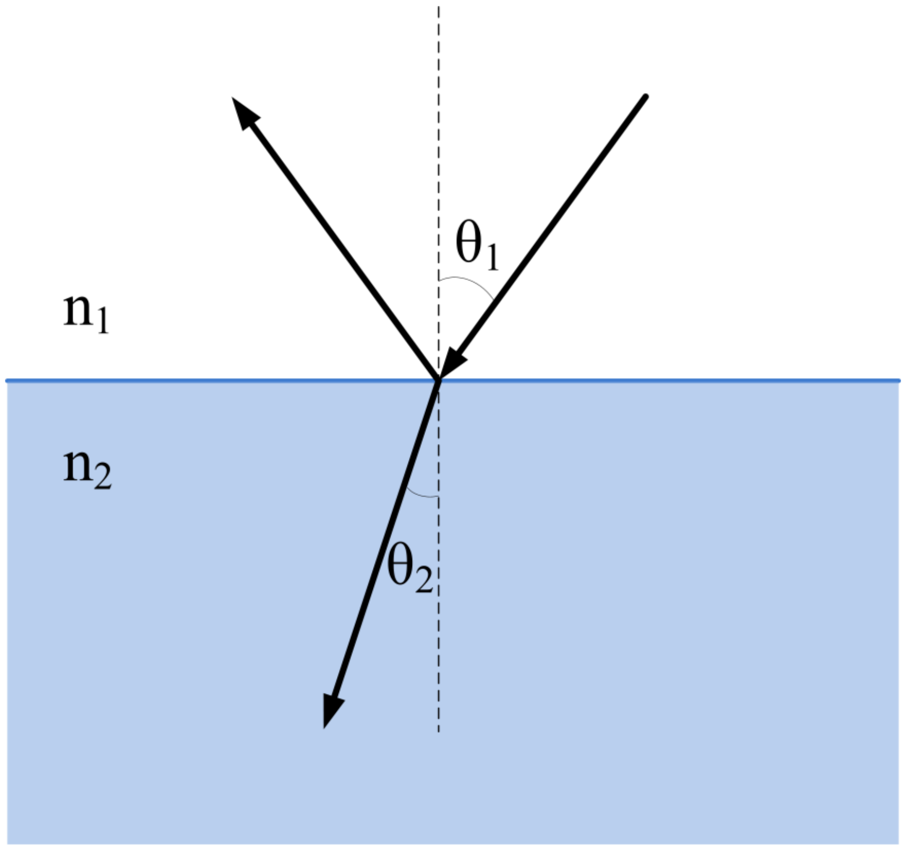


Snell's Law

- The relationship between the angle of incidence and angle of refraction/transmission for light passing through a boundary between two different isotropic media is:

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{v_1}{v_2} = \frac{n_2}{n_1}$$

- θ_1 and θ_2 are incoming/outgoing angles
- v_1 and v_2 are the phase velocities in the two materials
- n_1 and n_2 are the indices of refraction in the two materials



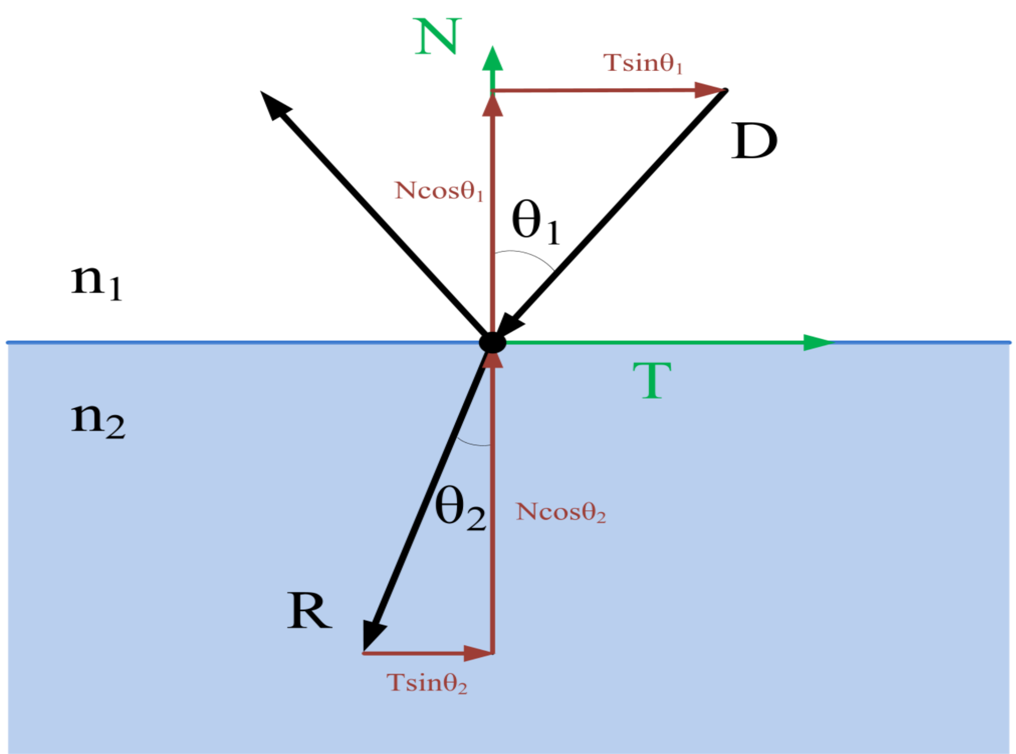
Transmitted Ray

- **D** - incident ray direction
- **N** - unit surface normal
- **T** - unit surface tangent
  (in the plane of **N** and **D**)
- **R** - refracted ray direction
- (N, T, D, and R are all normalized)

$$\boldsymbol{N}\cos\theta_1 + \boldsymbol{T}\sin\theta_1 + \boldsymbol{D} = 0$$

$$\boldsymbol{N}\cos\theta_2 + \boldsymbol{T}\sin\theta_2 + \boldsymbol{R} = 0$$

$$n_1 \sin\theta_1 = n_2 \sin\theta_2 \quad \text{(Snell)}$$



Transmitted Ray

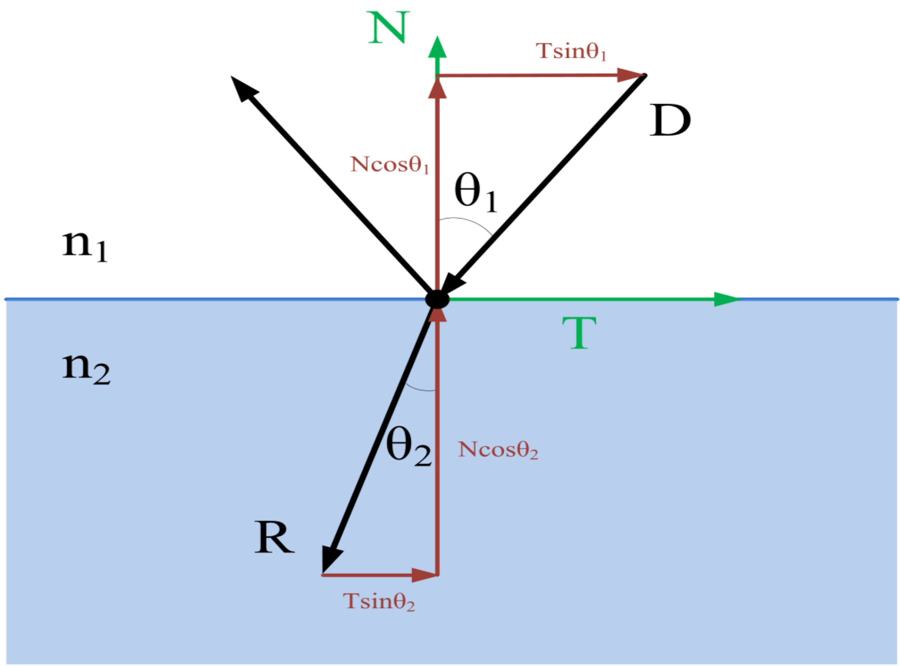
$$\boldsymbol{R} = -\boldsymbol{T}\sin\theta_2 - \boldsymbol{N}\cos\theta_2$$

$$= (\boldsymbol{D} + \boldsymbol{N}\cos\theta_1)\frac{\sin\theta_2}{\sin\theta_1} - \boldsymbol{N}\sqrt{1-\sin^2\theta_2}$$

$$= (\boldsymbol{D} + \boldsymbol{N}\cos\theta_1)\frac{n_1}{n_2} - \boldsymbol{N}\sqrt{1-\left(\frac{n_1}{n_2}\sin\theta_1\right)^2}$$

$$= \boldsymbol{D}\frac{n_1}{n_2} + \boldsymbol{N}\left(\frac{n_1}{n_2}\cos\theta_1 - \sqrt{1-\frac{n_1^2}{n_2^2}(1-\cos^2\theta_1)}\right)$$

$$= \boldsymbol{D}\frac{n_1}{n_2} - \boldsymbol{N}\left(\frac{n_1}{n_2}\boldsymbol{D}\bullet\boldsymbol{N} + \sqrt{1-\frac{n_1^2}{n_2^2}(1-(\boldsymbol{D}\bullet\boldsymbol{N})^2)}\right)$$

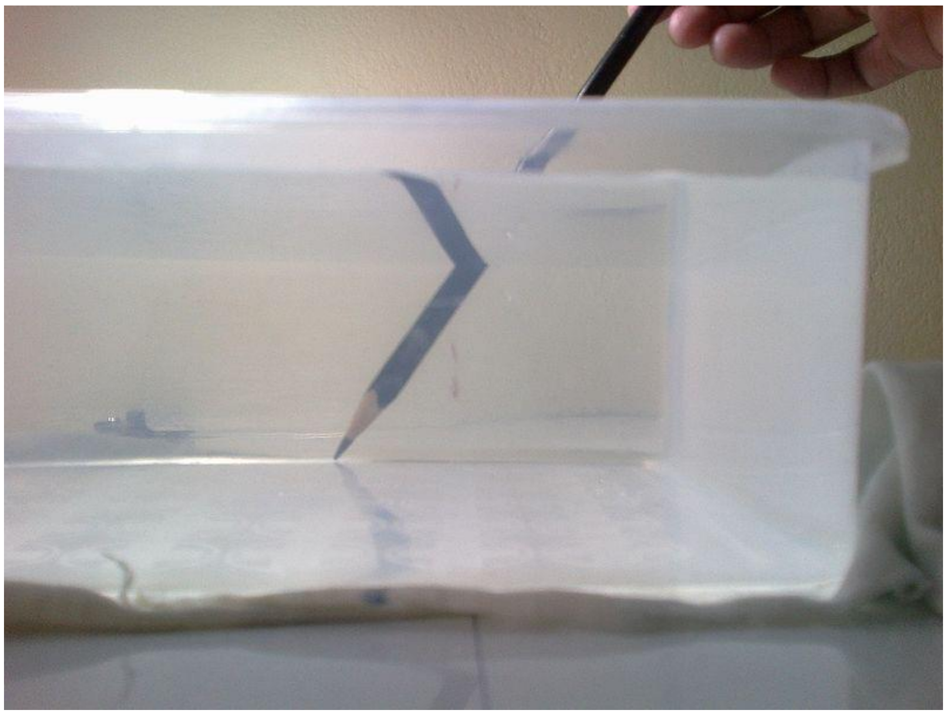


- If the number under the square root is negative, there is no refracted/transmitted ray and all of the light is reflected (total internal reflection)
- This equation works regardless of which of n_1 or n_2 is larger

Total Internal Reflection

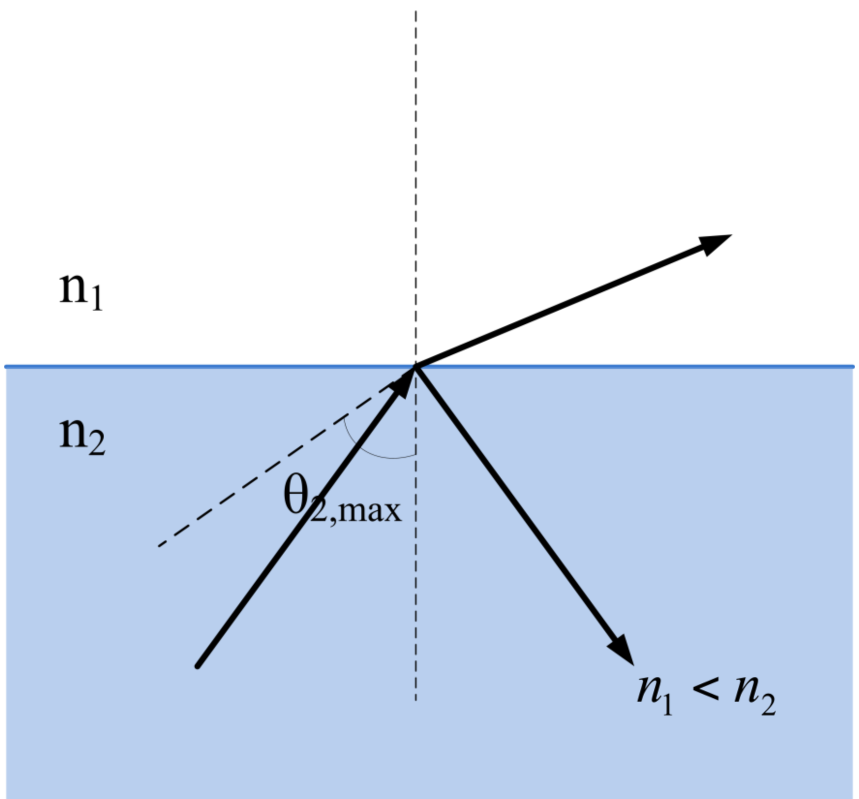
Total Internal Reflection

- Responsible for much of the rich appearance of glass and water

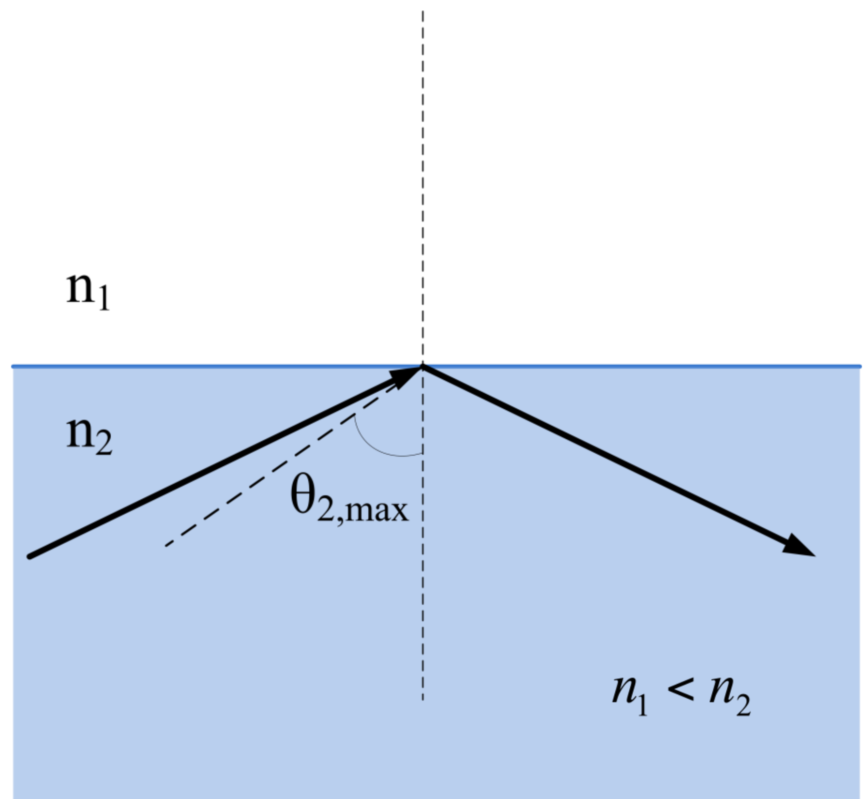


Total Internal Reflection

- When light goes from a material with a higher refraction index (n_2) to a material with a lower refraction index (n_1), there is no refraction/transmission if the incident angle exceeds a critical angle (then, all the light reflects)



when $\theta_2 < \theta_{2,max}$, both refraction/transmission and reflection

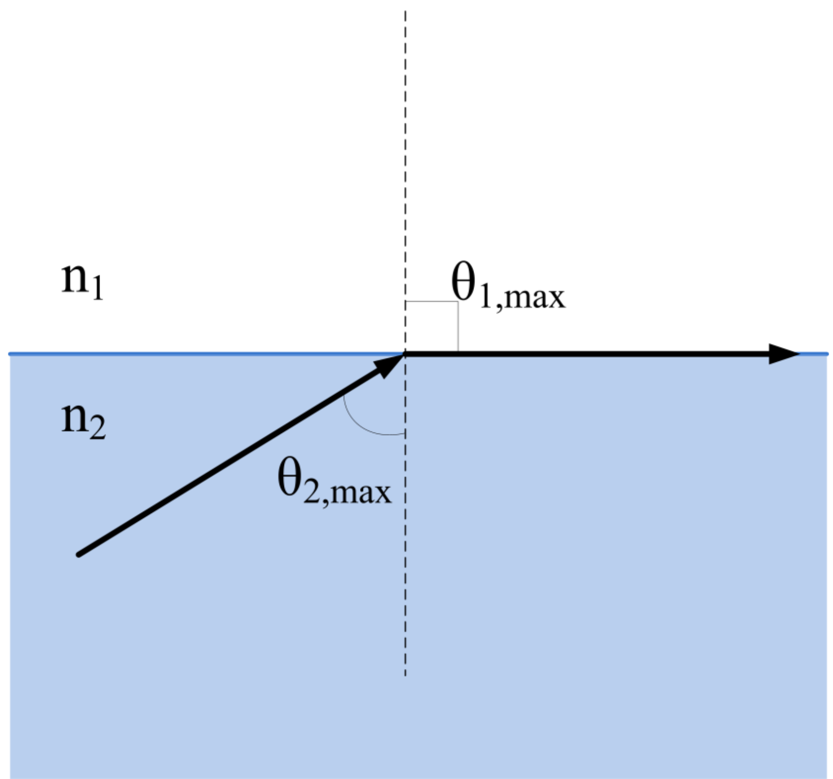


when $\theta_2 > \theta_{2,max}$, only reflection

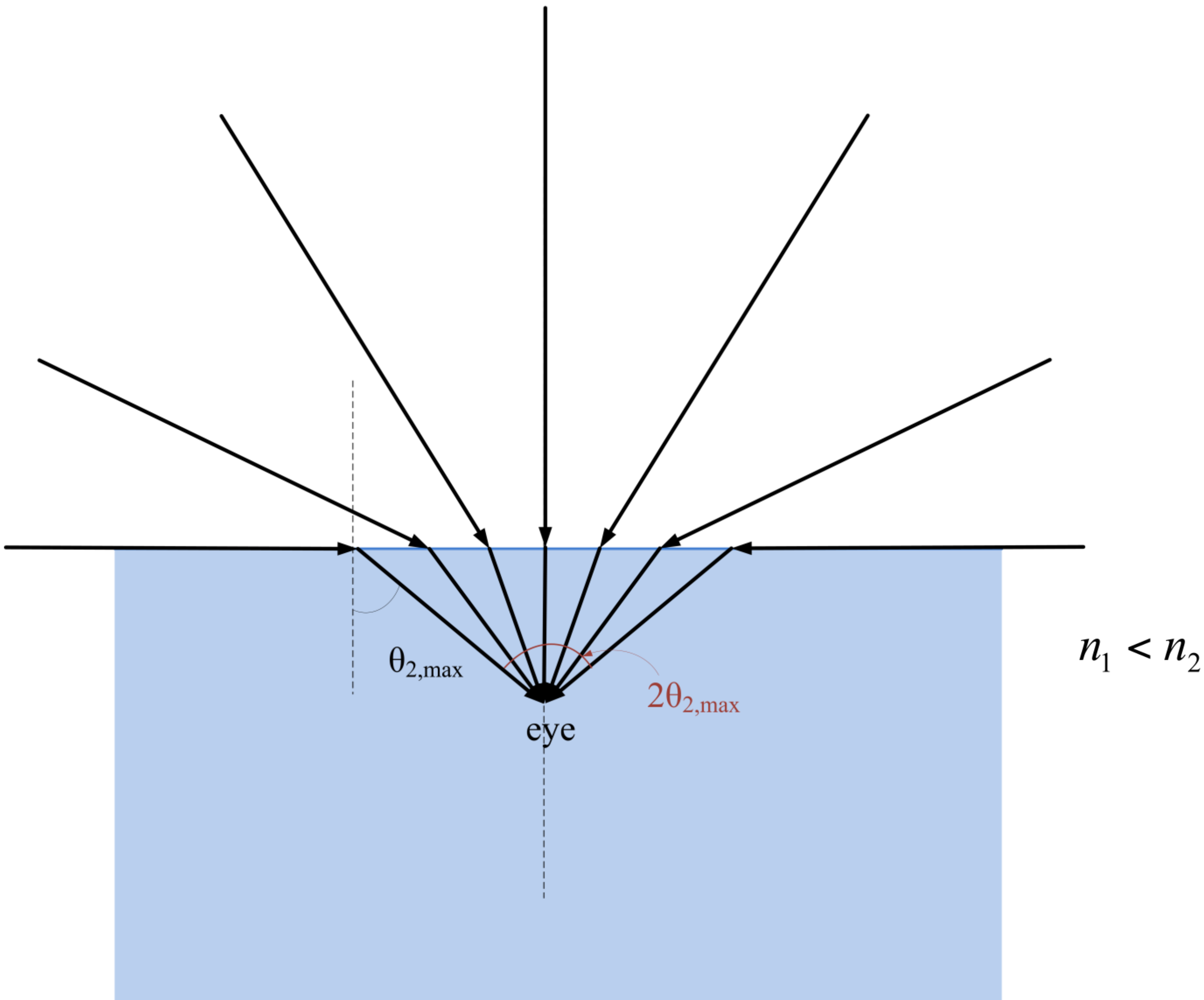
Critical Angle

- Compute the critical angle using Snell's Law: $\dfrac{\sin\theta_1}{\sin\theta_2} = \dfrac{v_1}{v_2} = \dfrac{n_2}{n_1}$
- Assume $n_1 < n_2$, then

$$\theta_1 = \frac{\pi}{2} \Longrightarrow \theta_{2,\max} = \arcsin\left(\frac{n_1}{n_2}\right)$$



Snell’s Window



Snell’s Window

http://en.wikipedia.org/wiki/Snell’s_window

Viewing Angle

Reflection vs. Transmission

- The proportion of reflection versus transmission gradually increases as we go from a perpendicular to a parallel (grazing) viewing direction

Perpendicular view:
high transmission, low reflection

Parallel (grazing) view:
high reflection, low transmission

Reflection vs. Transmission

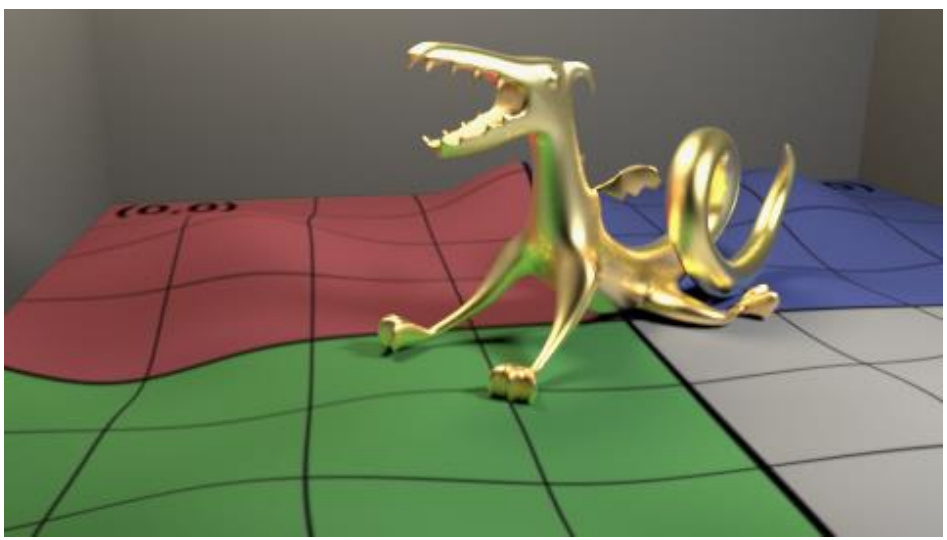
- Reflection similarly increases for non-transparent (opaque) objects as the viewing direction becomes more parallel

- Notice the increase in reflectivity of the table from left to right

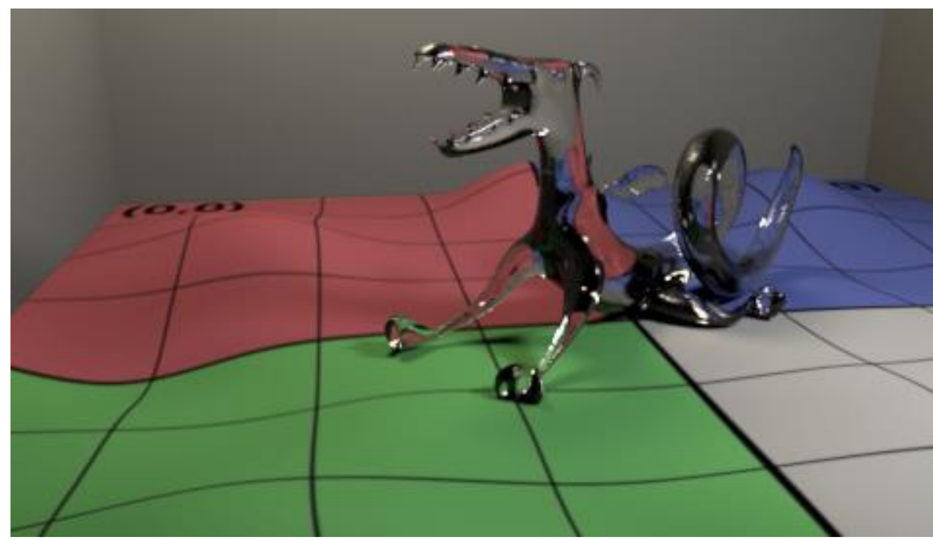
[Lafortune et al. 1997]

Conductors vs. Dielectrics

- Conductors
  - Objects that conduct electricity (e.g. metal)
  - Have reflection and absorption, but no transmission
- Dielectrics
  - Objects that don't conduct electricity (e.g. glass)
  - Have both reflection and transmission



Conductor



Dielectric

http://renderman.pixar.com/resources/current/rms/tutorialRayTracing.html

Conductors

- Conductors don't transmit light
- Most of the incident light is reflected, and some of it is absorbed
  - The reflection term changes relatively slowly with the incident angle (as compared to dielectric materials)
    - E.g., for aluminum, the reflection changes from about 90% to 100% as the incident angle changes from 0^o to 90^o
- Can typically treat k_r as a constant independent of viewing direction (and $k_t = 0$)

Ray traced conductors

http://download.autodesk.com
http://www.raytracing.co.uk/

Dielectrics

- Light can be polarized into 2 parts based on whether the plane containing the incident, reflected, and refracted rays is parallel or perpendicular to the electric field of the light
- Denoted the light as p-polarized if it is parallel and by s-polarized if it is perpendicular (to the electric field)
- The Fresnel equations give the fraction of light reflected off the interface between two materials as:

$$R_p = \left| \frac{n_1 \cos\theta_t - n_2 \cos\theta_i}{n_1 \cos\theta_t + n_2 \cos\theta_i} \right|^2 \qquad R_s = \left| \frac{n_1 \cos\theta_i - n_2 \cos\theta_t}{n_1 \cos\theta_i + n_2 \cos\theta_t} \right|^2$$

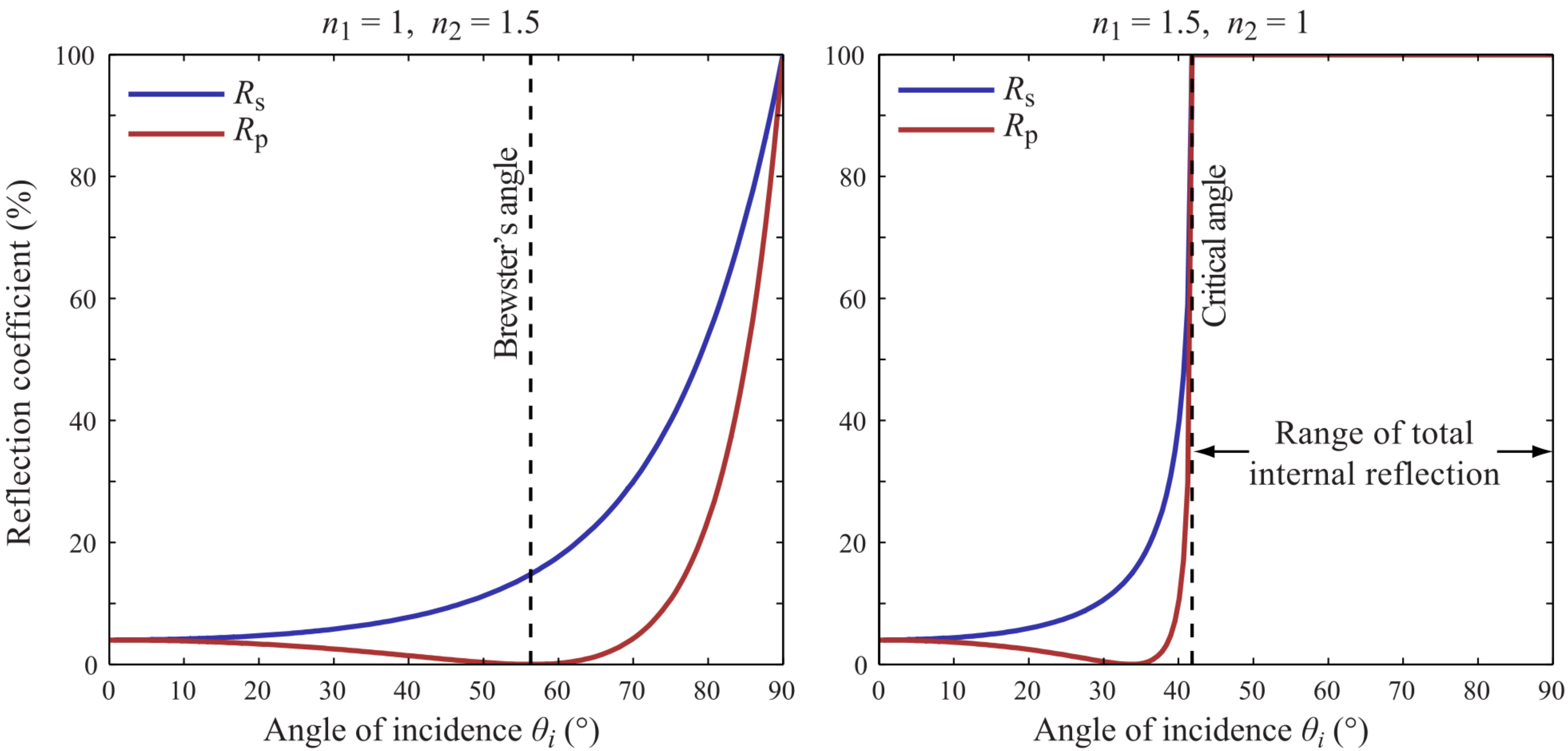
- It is assumed that any light which is not reflected is instead transmitted through the interface from one material to the other, so:

$$T_p = 1 - R_p \qquad T_s = 1 - R_s$$

- For unpolarized light (a typical assumption in ray tracing), the overall reflection and transmission coefficients are assumed to be:

$$R = \frac{R_s + R_p}{2} \qquad T = 1 - R$$

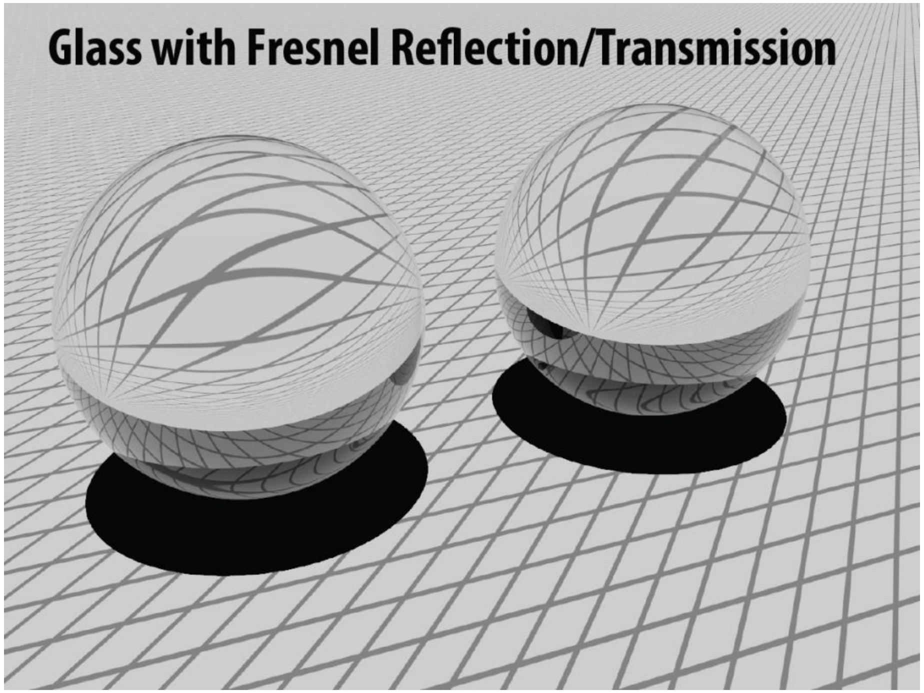
Dielectrics

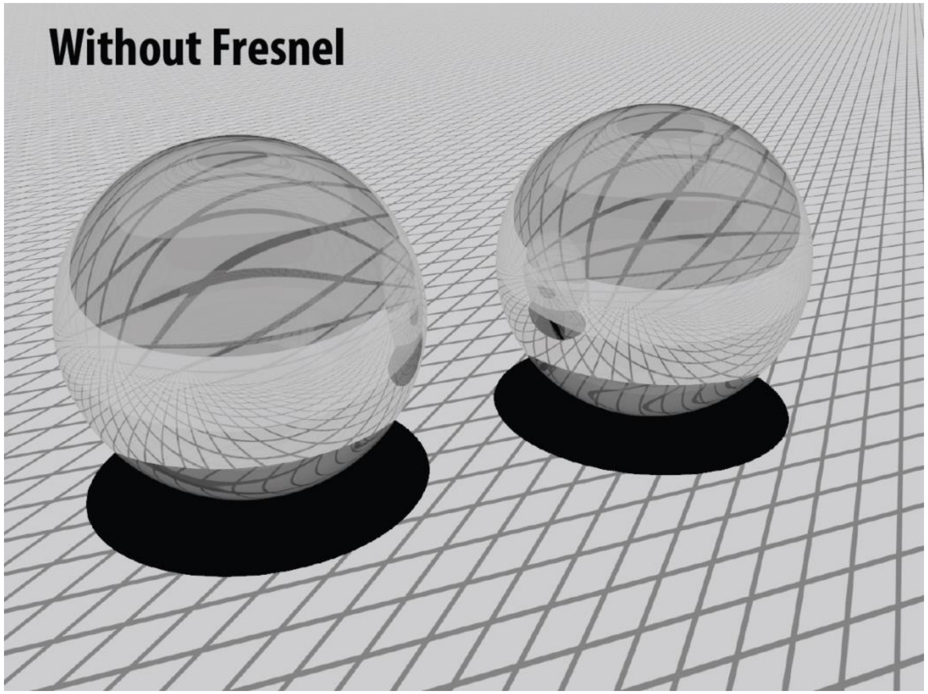


Light entering a denser material, e.g. from air into water

Light leaving a denser material, e.g. from water into air

Dielectrics



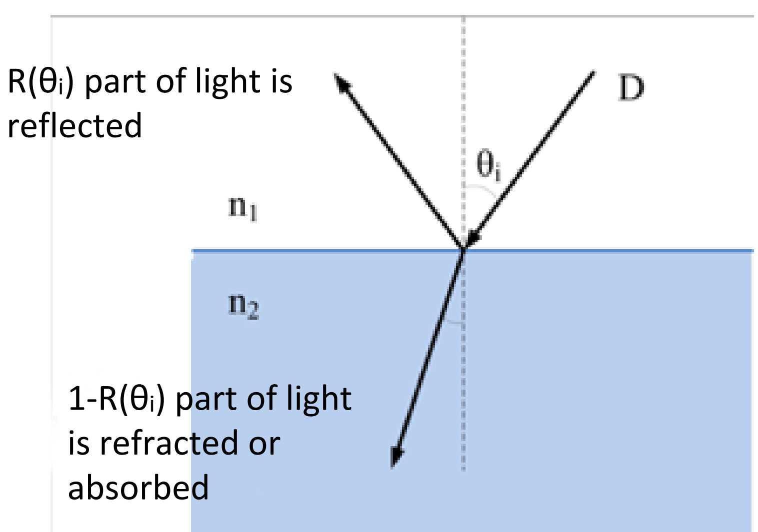


Schlick's Approximation

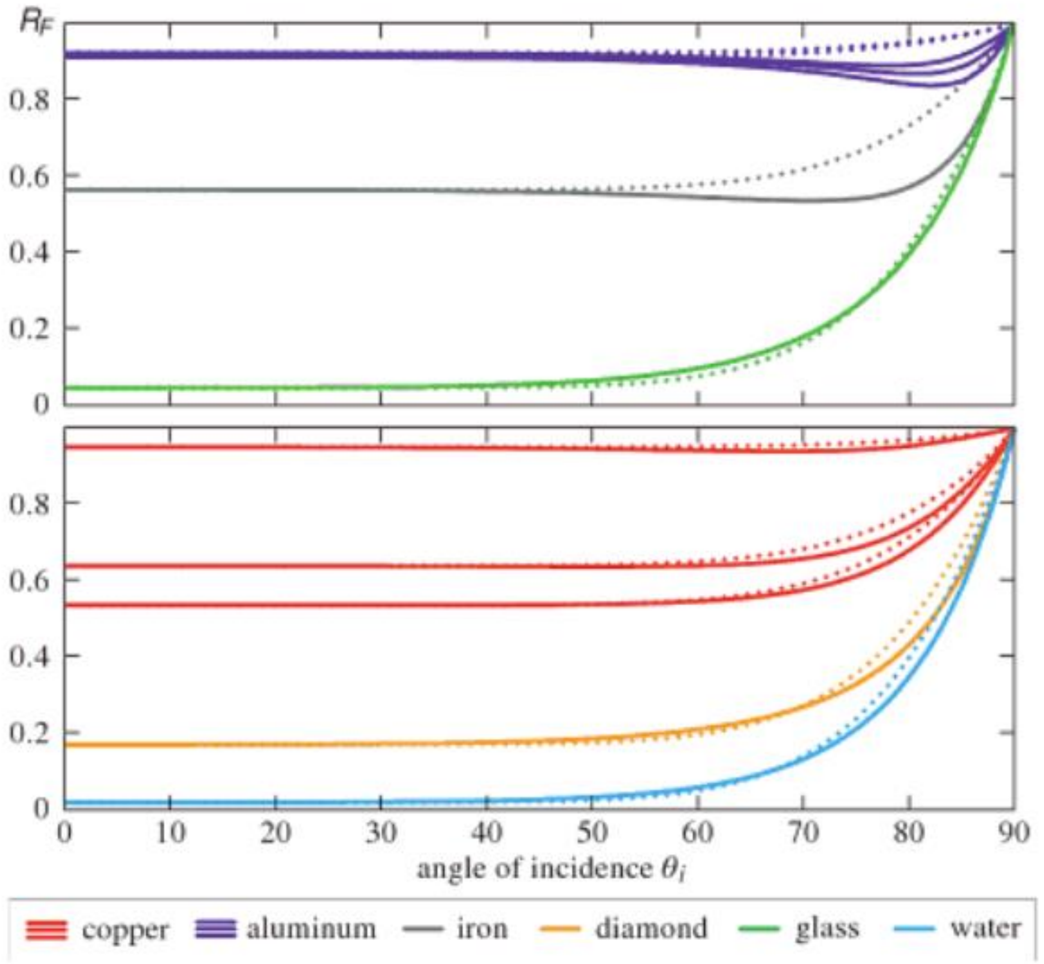
- Approximates the reflection term in the Fresnel Equation as:

$$R(\theta_i) = R_0 + (1 - R_0)(1 - \cos\theta_i)^5 \quad \text{where} \quad R_0 = \left(\frac{n_1 - n_2}{n_1 + n_2}\right)^2$$

- R_0 controls the Fresnel reflections of different materials
  - R_0 ranges from 0 to 1
  - Reference values are available for many real-world materials
  - Same equation can be used for both dielectrics and conductors



Schlick's Approximation



Schlick's approximation (dotted lines) compared to the Fresnel equations (solid lines)

Attenuation

Attenuation

- Light is attenuated as it travels through media
- The attenuation effect from the media is stronger over longer distances
- Different colors attenuate with different rates
  – Shallow water is clear (almost no attenuation)
  – Deeper water attenuates all the red light and looks blue-green
  – Even deeper water also attenuates all the green light and looks blue
  – Eventually all the light attenuates and the color ranges from blackish-blue to black

Beer's Law

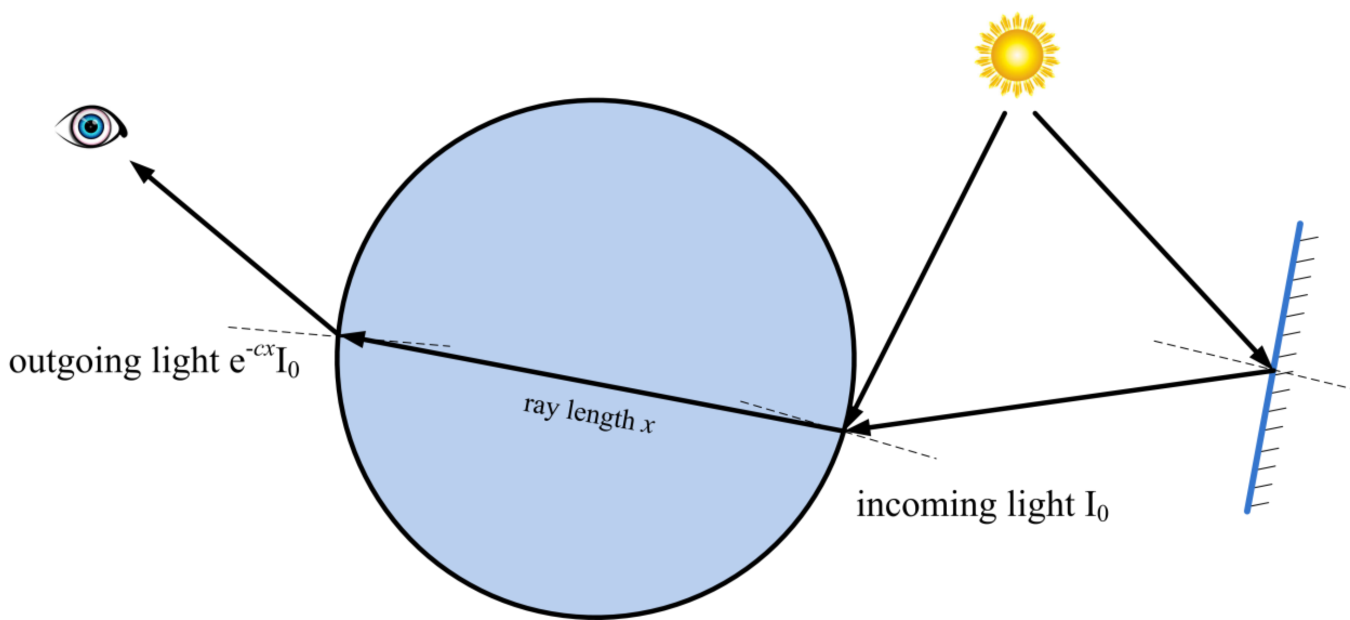
- If the media is homogeneous, the attenuation along the ray can be described using Beer's Law:

$$\frac{dI}{dx} = -cI$$

  where *I* is the light intensity, *x* is the distance along the ray, and *c* is the attenuation constant (which varies based on color/wavelength)

- Solving this Ordinary Differential Equation (ODE) with the initial value $I(0) = I_0$ results in:

$$I(x) = I_0 e^{-cx}$$



Beer's Law

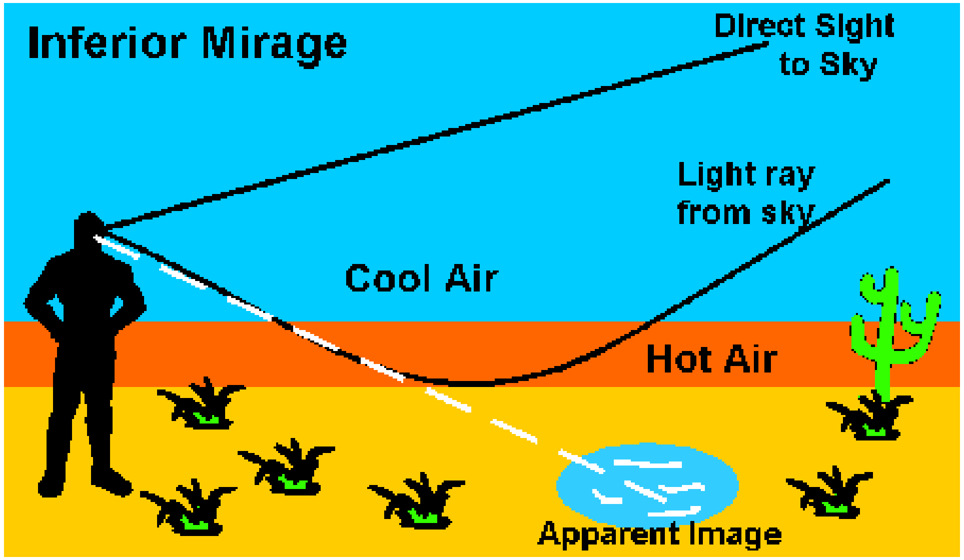
The color of a transparent object can be described by three independent attenuation components for the three color channels (R, G, B)

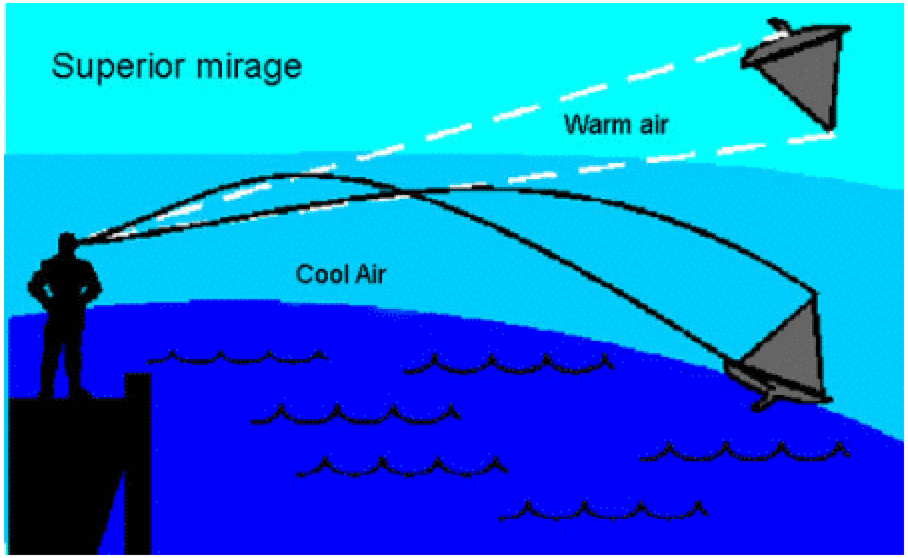
Bending Rays

Atmospheric Refraction

- Light is bent into a curved path when it passes through the atmosphere
- Happens continuously, not just at an interface
- This is due to the continuous variations in air density, and thus continuous variations in the refractive index

- Inferior mirage - the mirage is located under the real object
- Superior mirage - the mirage is located above the real object





http://science-at-home.org/mirages/

Inferior Mirage

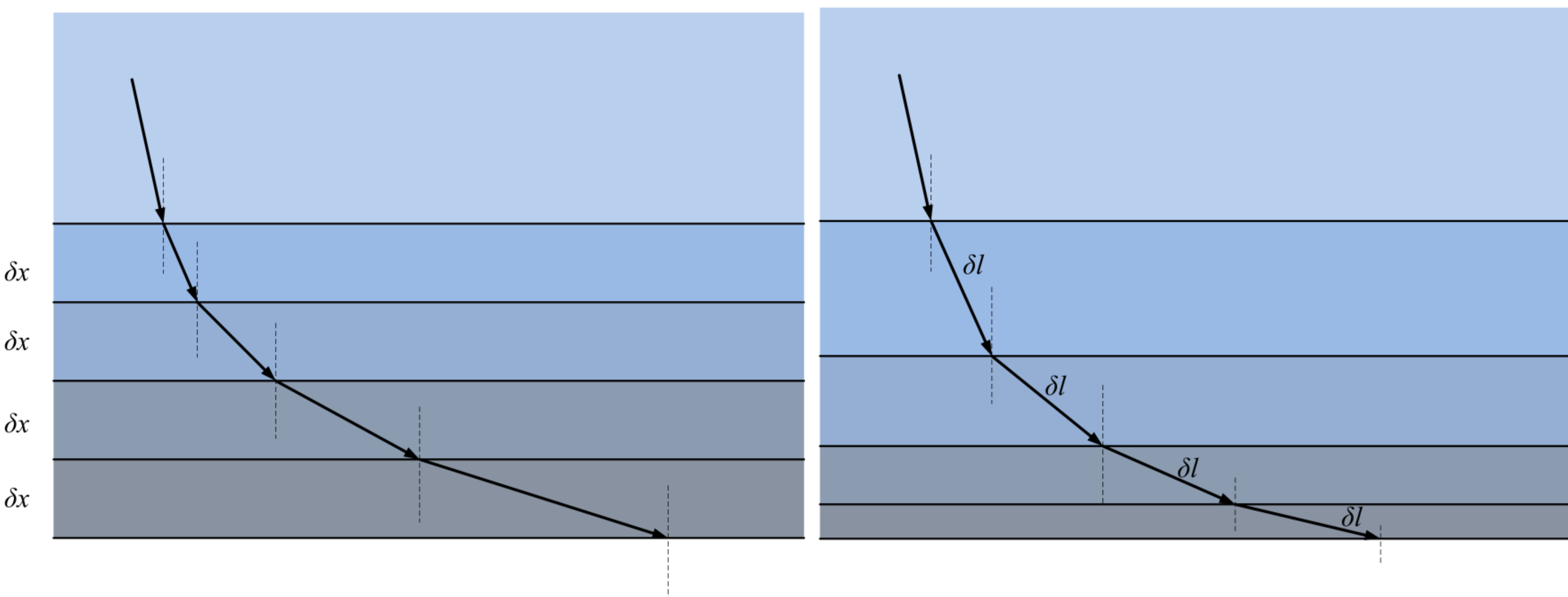
Credit: Brocken Inaglory

Superior Mirage

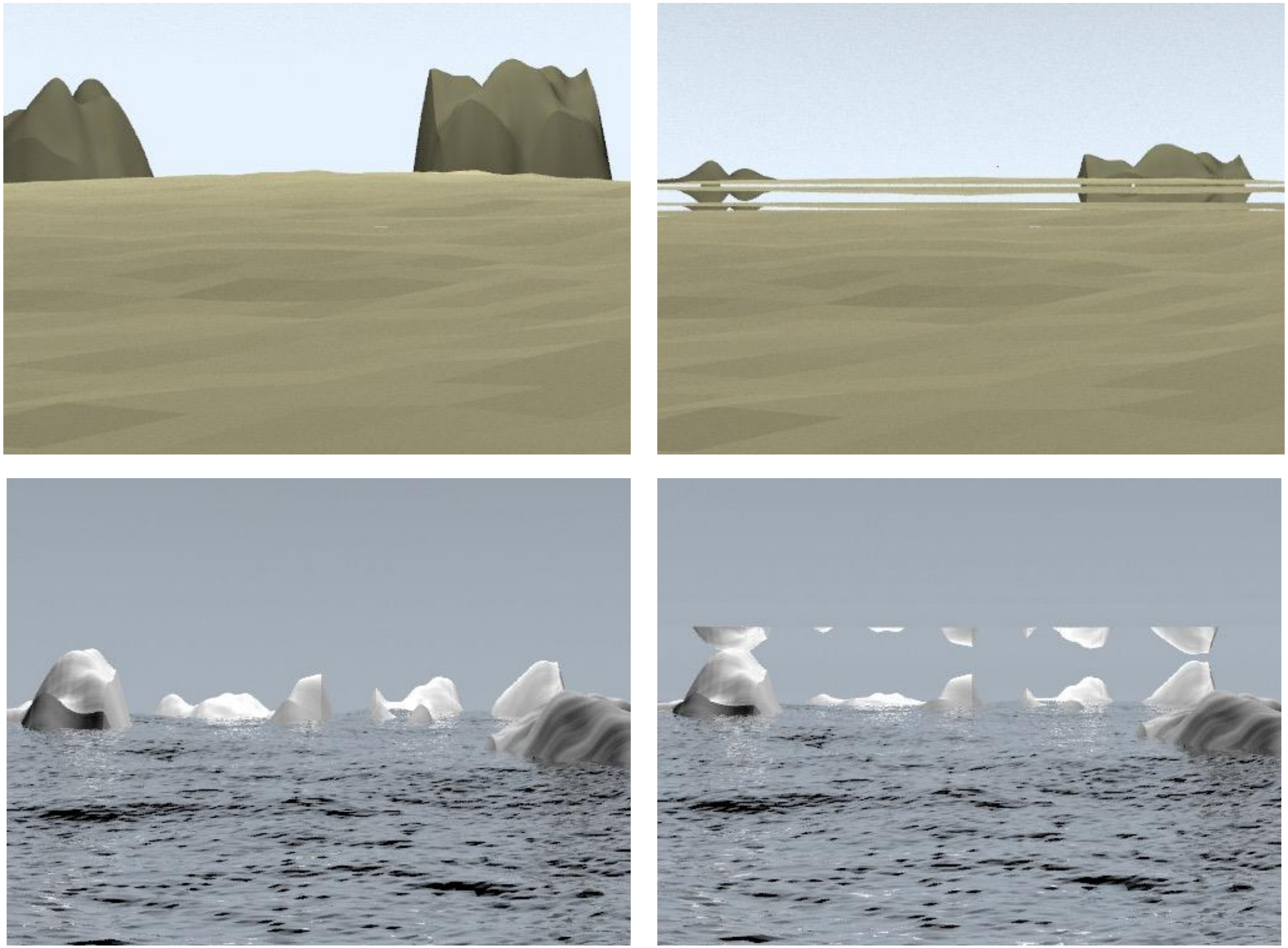
Credit: Timpaananen

Ray Tracing Mirages

- Bend the rays as they go through varying air densities
- Change the light direction between every interval in the vertical direction or along the ray direction

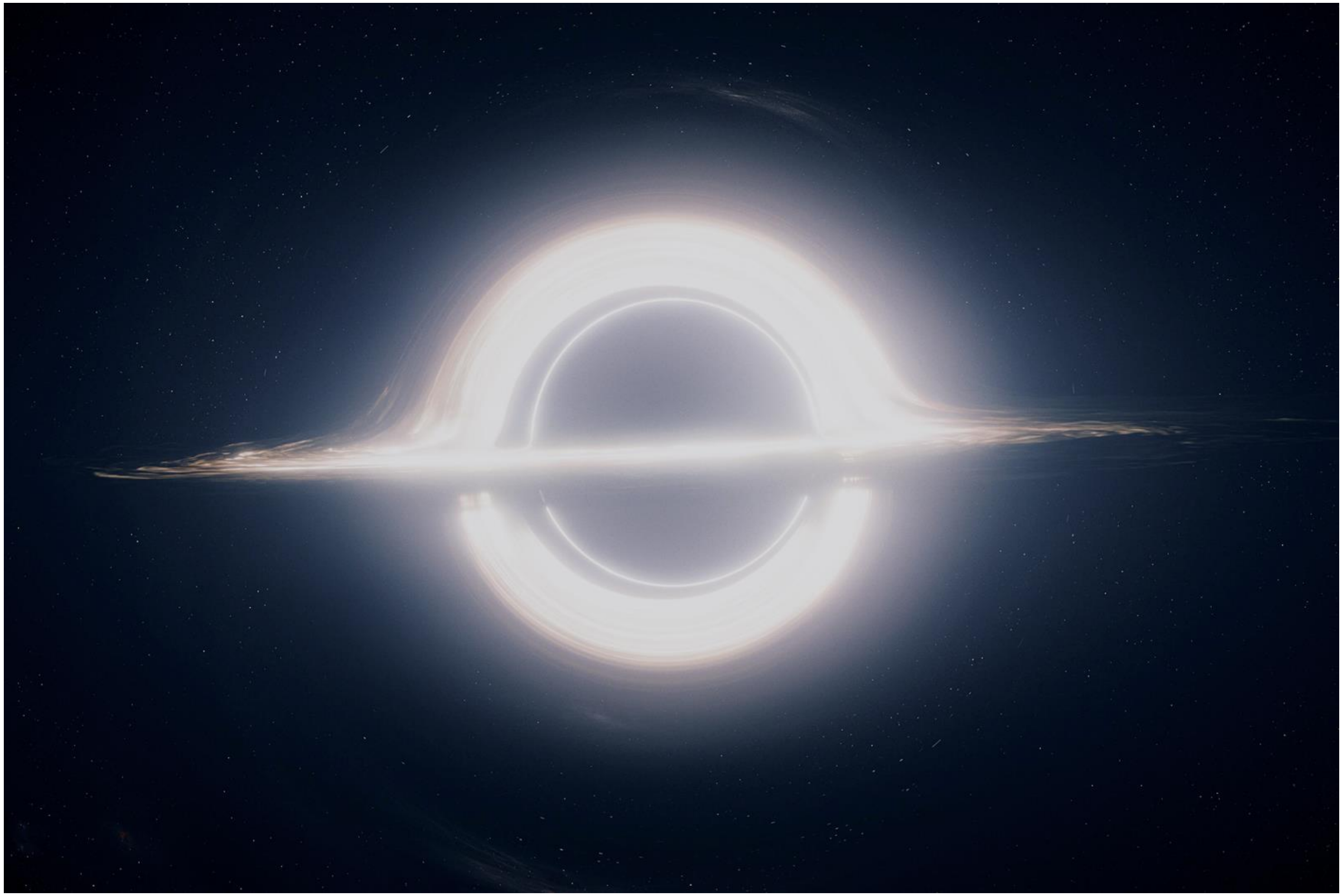


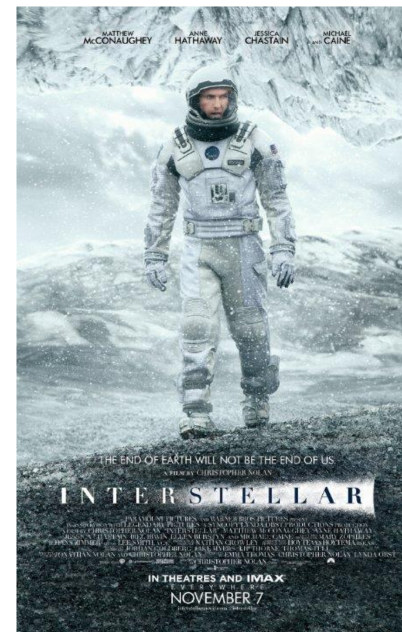
Ray Tracing Mirages



by Itai Katz and Pankhudi

Ray Tracing Curved Paths



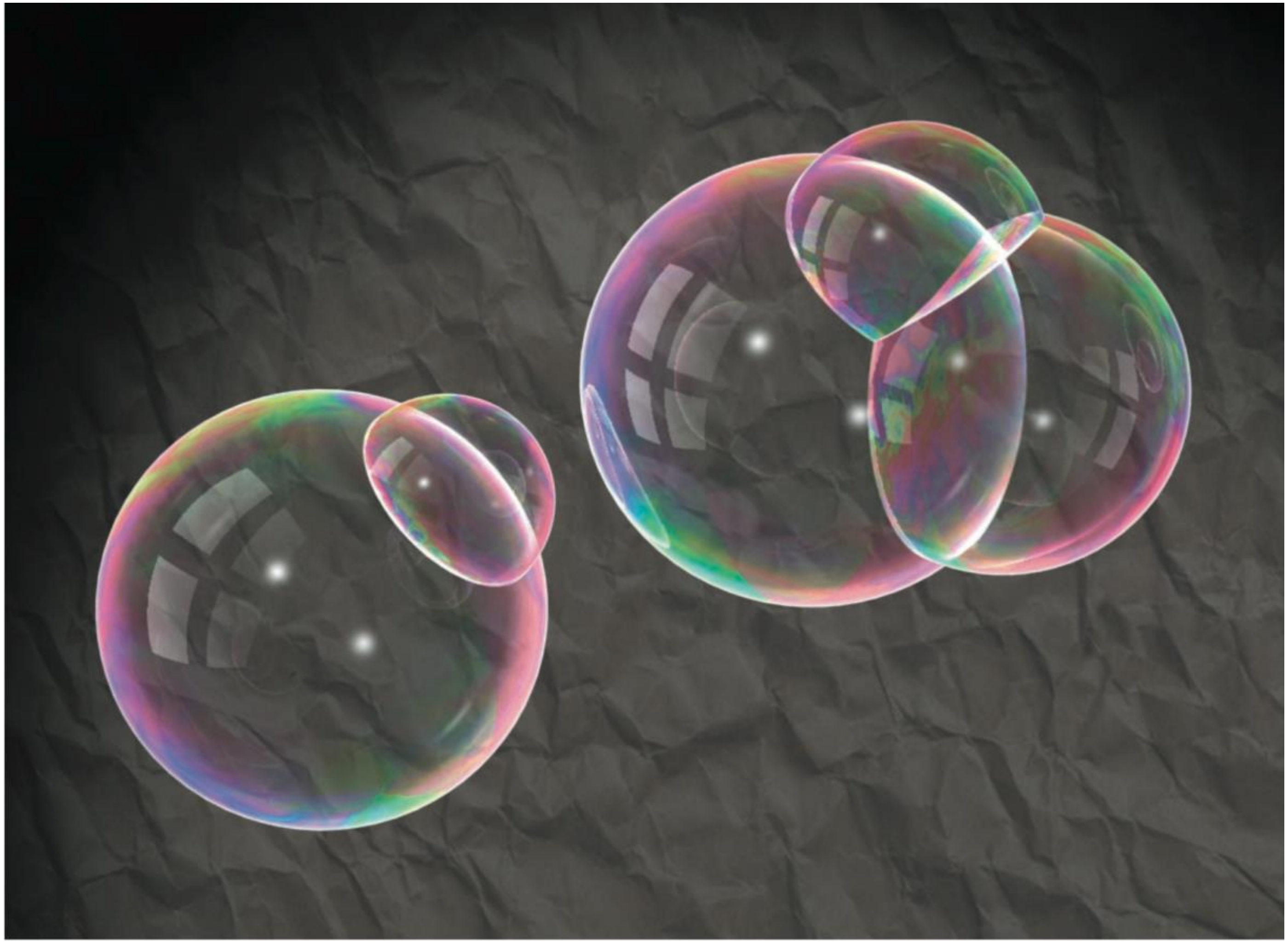


http://www.wired.com/2014/10/astrophysics-interstellar-black-hole/
http://news.discovery.com/space/interstellar-black-hole-is-best-black-hole-in-sci-fi-141029.htm

Iridescence

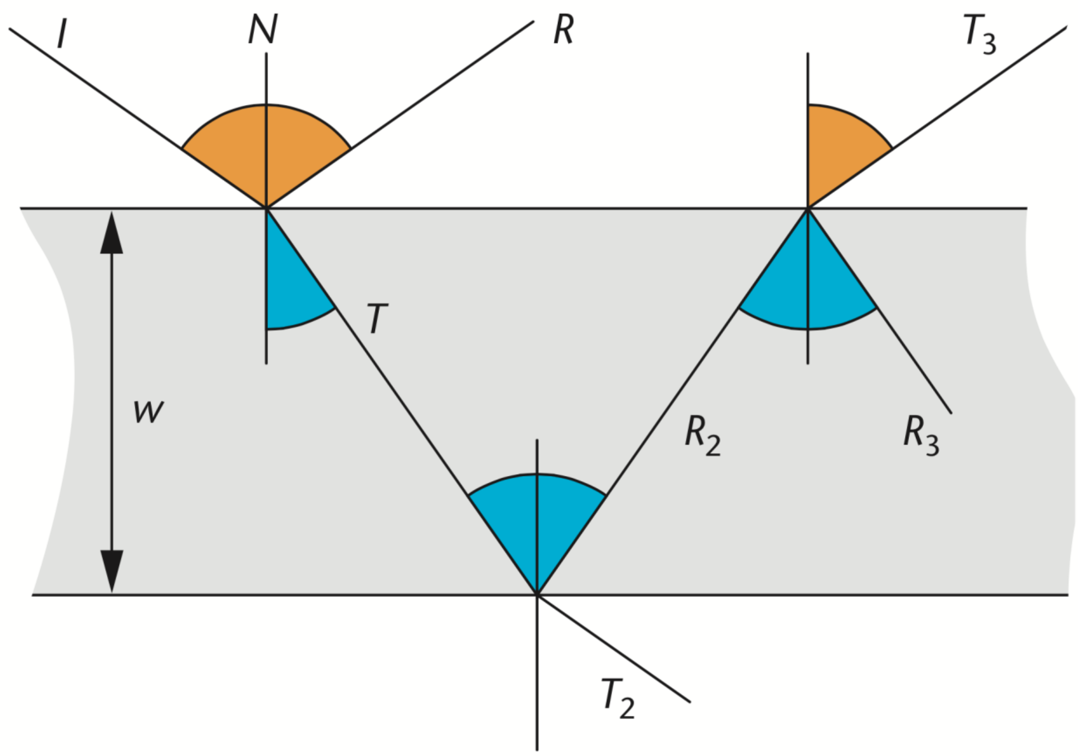
Iridescence

- Color of light changes with the viewing direction



Iridescence

- Various light waves are emitted in the same direction giving constructive and destructive interference



http://www.glassner.com/wp-content/uploads/2014/04/CG-CGA-PDF-00-11-Soap-Bubbles-2-Nov00.pdf