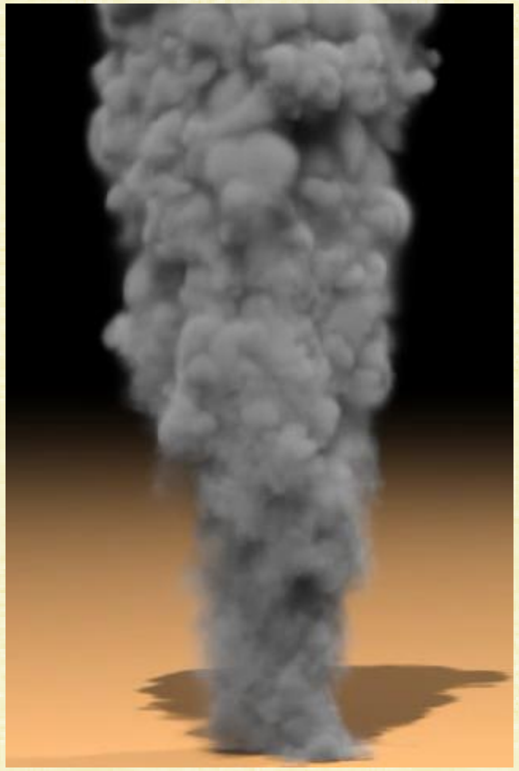
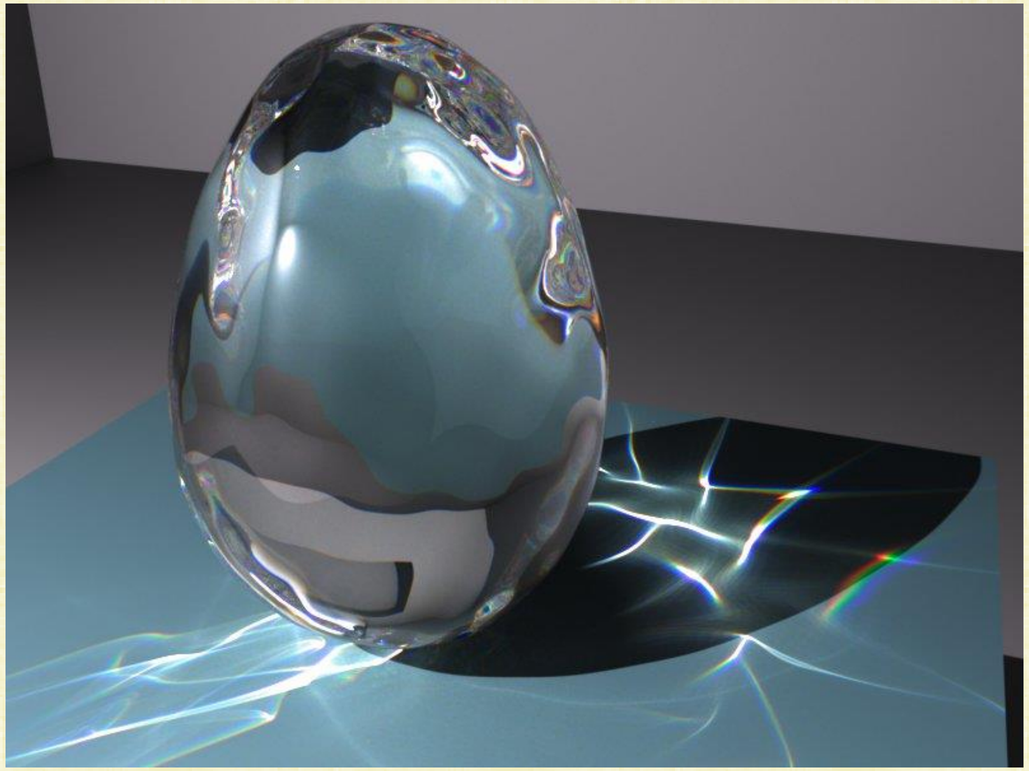
Recursive Ray Tracing

Recursive Ray Tracing

- Pixel color is determined by the camera ray passing through it:

$$(k_R, k_G, k_B)\left(L_{diffuse} + L_{ambient}\right) + L_{reflect} + L_{transmit}$$

- Diffuse shading is computed using all the unobscured lights (as determined via shadow rays):

$$L_{diffuse} = \sum_{lights} V_{light} I_{light} \max\left(0, \cos\theta_{light}\right)$$

- $V_{light} = 1$ for visible light sources, and $V_{light} = 0$ for occluded light sources
- When all V_{light} are identically zero, ambient shading is added:

$$L_{ambient} = I_{ambient} \prod_{lights} \left(1 - V_{light}\right)$$

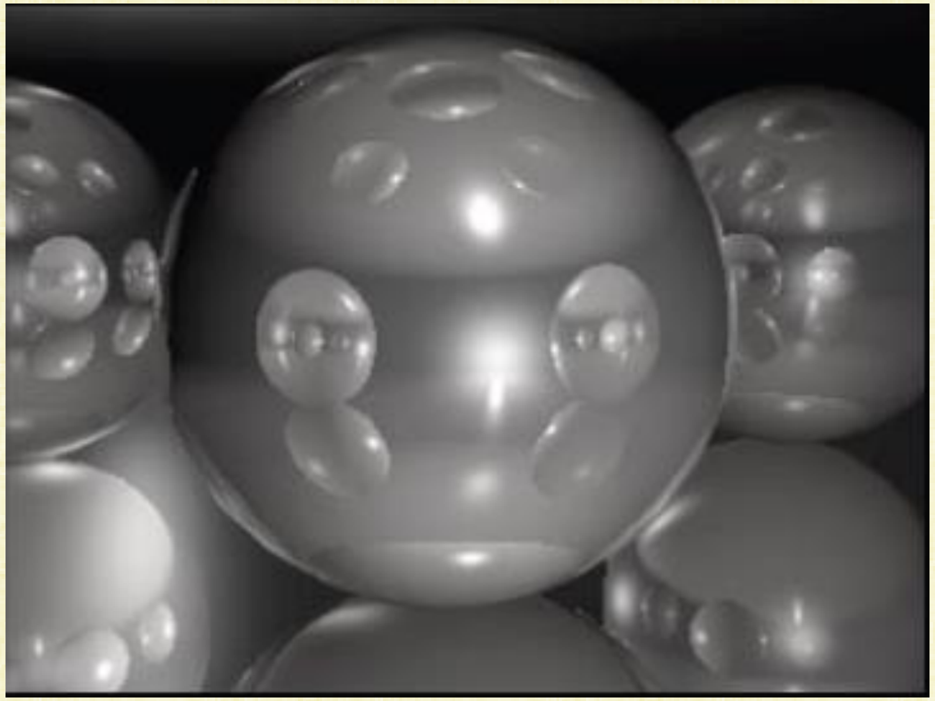
- Color from other objects that shows up due to mirror-like reflections is added via $L_{reflect}$
- Color from other objects that shows up due to transparency is added via $L_{transmit}$

Scaling Coefficients

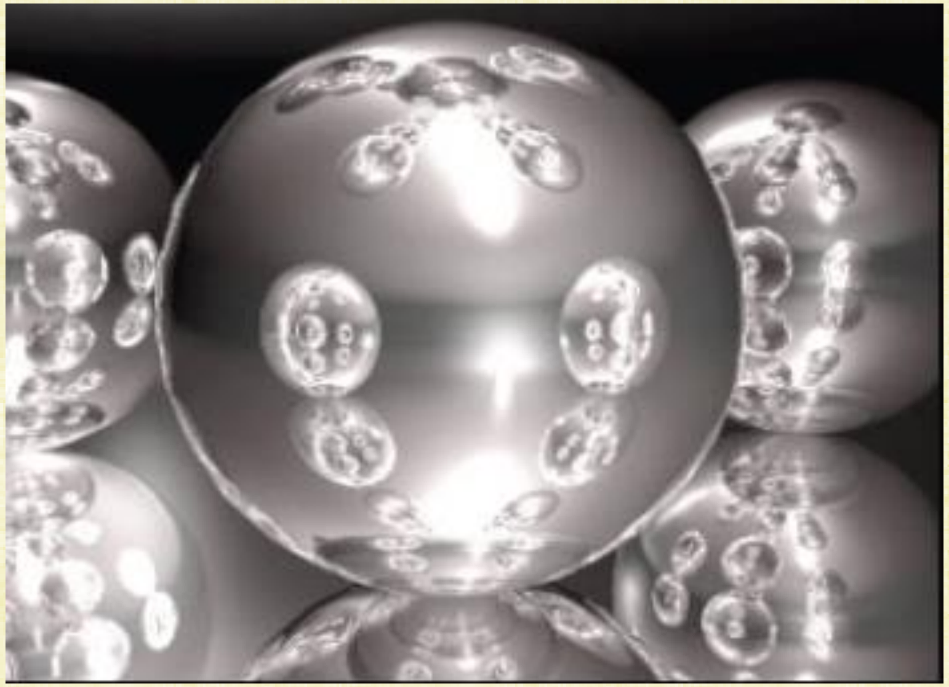
- Scaling coefficients can be added for every contribution:

$$(k_R, k_G, k_B)\left(k_d L_{diffuse} + k_a L_{ambient}\right) + k_r L_{reflect} + k_t L_{transmit}$$

- These coefficients can be adjusted relative to each other to get the desired "look"
- All the coefficients can be scaled together for overall brightness/darkness
  - Be careful to avoid over-saturation, when scaling coefficients up



less reflection (darker)



more reflection (brighter)

Recursion

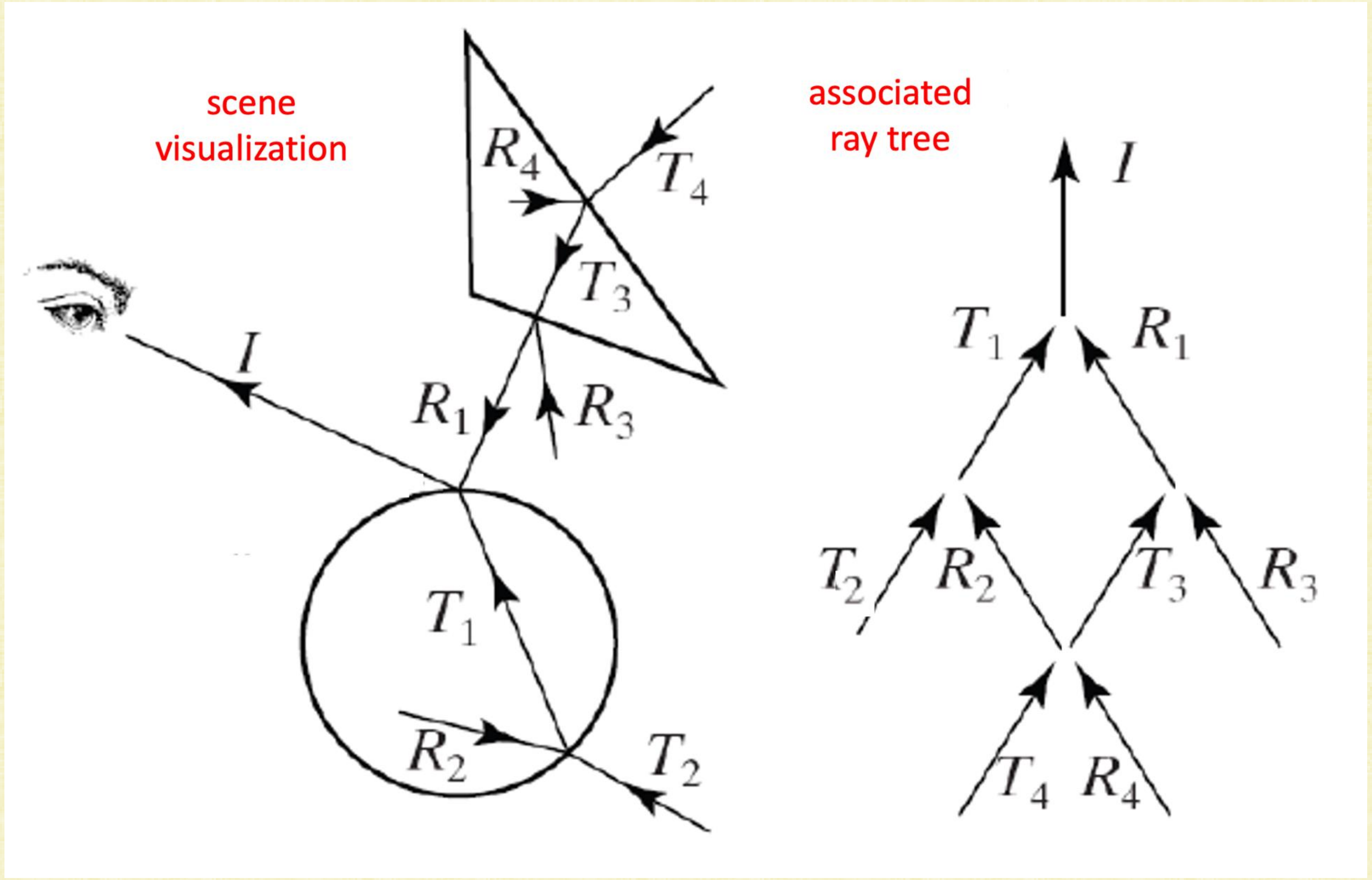
- $L_{reflect}$ and $L_{transmit}$ are computed in the same way that pixel color is computed

- Construct a ray for the <u>reflection direction</u>: intersect it with scene geometry and store the resulting color in $L_{reflect}$
- Construct a ray for the <u>transmission direction</u>: intersect it with scene geometry and store the resulting color in $L_{transmit}$

- Computing color for the points intersected by reflected and transmitted rays requires the evaluation of $(k_R, k_G, k_B)(k_d L_{diffuse} + k_a L_{ambient}) + k_r L_{reflect} + k_t L_{transmit}$ at each intersection point
  - Thus, shadow rays need to be sent out to light sources to compute V_{light} for $L_{diffuse}$ and $L_{ambient}$
  - In addition, reflected and transmitted rays need to be sent out to compute $L_{reflect}$ and $L_{transmit}$
  - Computing the colors for those reflected and transmitted rays requires repeating the process again

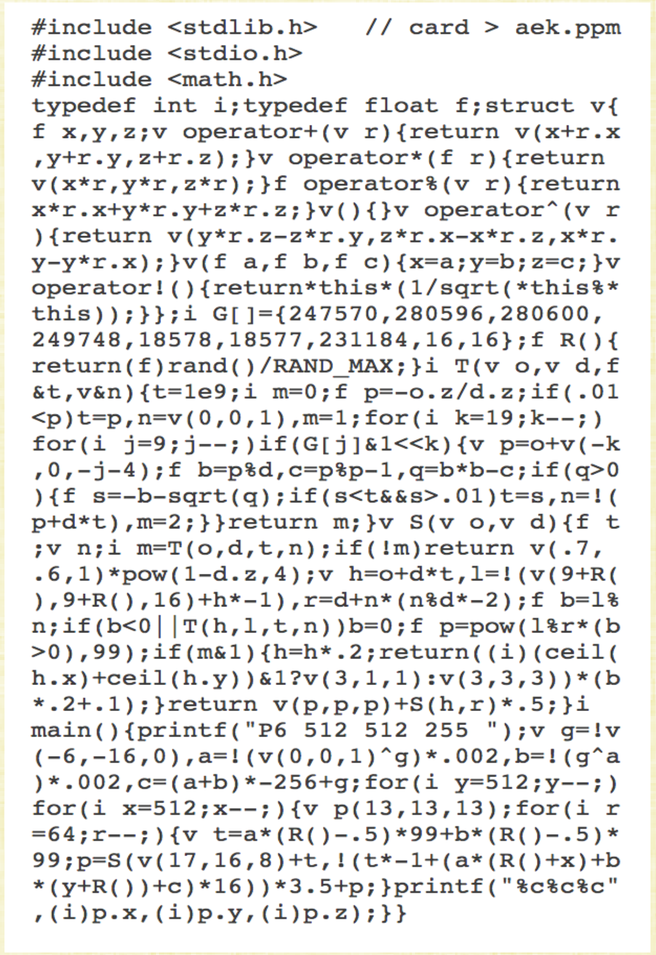
- Recursion occurs whenever $k_r \neq 0$ or $k_t \neq 0$

Ray Trees


scene visualization
associated ray tree
I
R_4
T_4
T_3
R_1
R_3
T_1
R_2
T_2
I
T_1
R_1
T_2
R_2
T_3
R_3
T_4
R_4

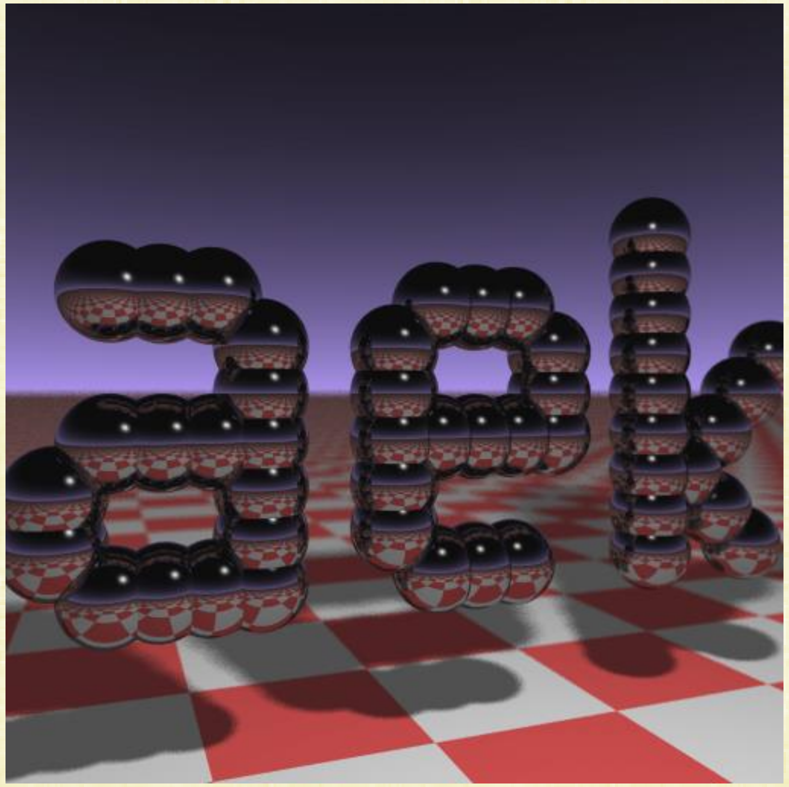
Code Simplicity

- Recursion enables the creation of impressive images with minimal code, as demonstrated by these 1337 characters printed on the back of a business card



```
#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);}}
```

$\longrightarrow$



http://fabiensanglard.net/rayTracing_back_of_business_card/

Termination

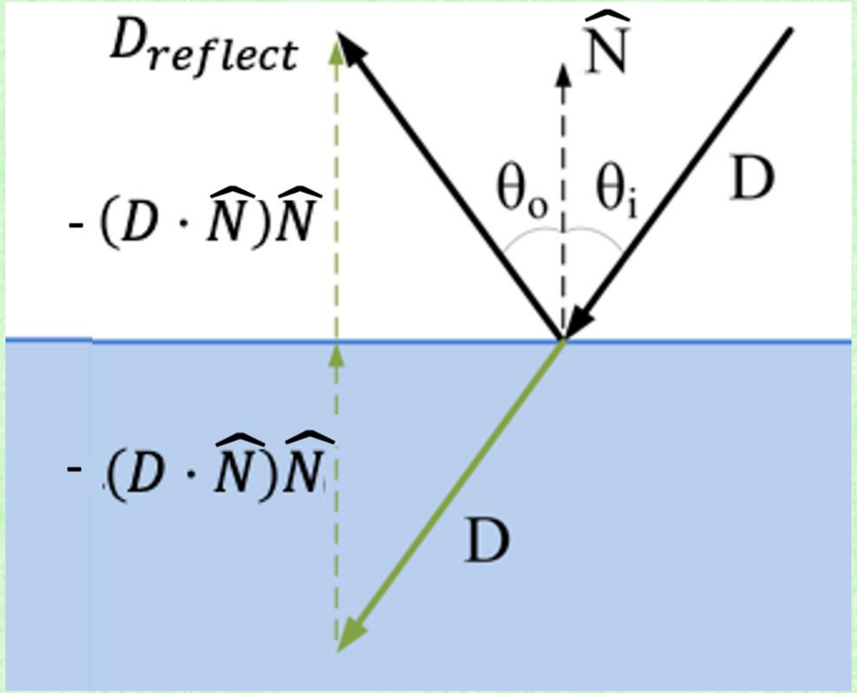
- If every intersected point continued to depend on reflected/transmitted rays, rays would be spawned indefinitely
- Due to limited stack space, a recursion limit is used to prevent stack overflow
  - The available of stack space varies based on hardware, the type of thread, etc.
  - The recursion limit varies based on the stack space, the programming language, etc.

- Terminating the recursion early means ignoring the subsequent values of $L_{reflect}$ and $L_{transmit}$ that would have been computed

- As long as nonzero k_d and k_a were previously found, a reasonable color is still computed
- Otherwise, if all the k_d and k_a were identically zero, termination means that no color is computed (this can happen in a house of mirror, or in a pile of bubbles, etc.)
- This results in black pixels (since the computed color is identically zero)
- Thus, it's preferable to terminate by using <u>nonzero values</u> for $L_{reflect}$ and $L_{transmit}$ (common choices: sky color, background color, etc.)

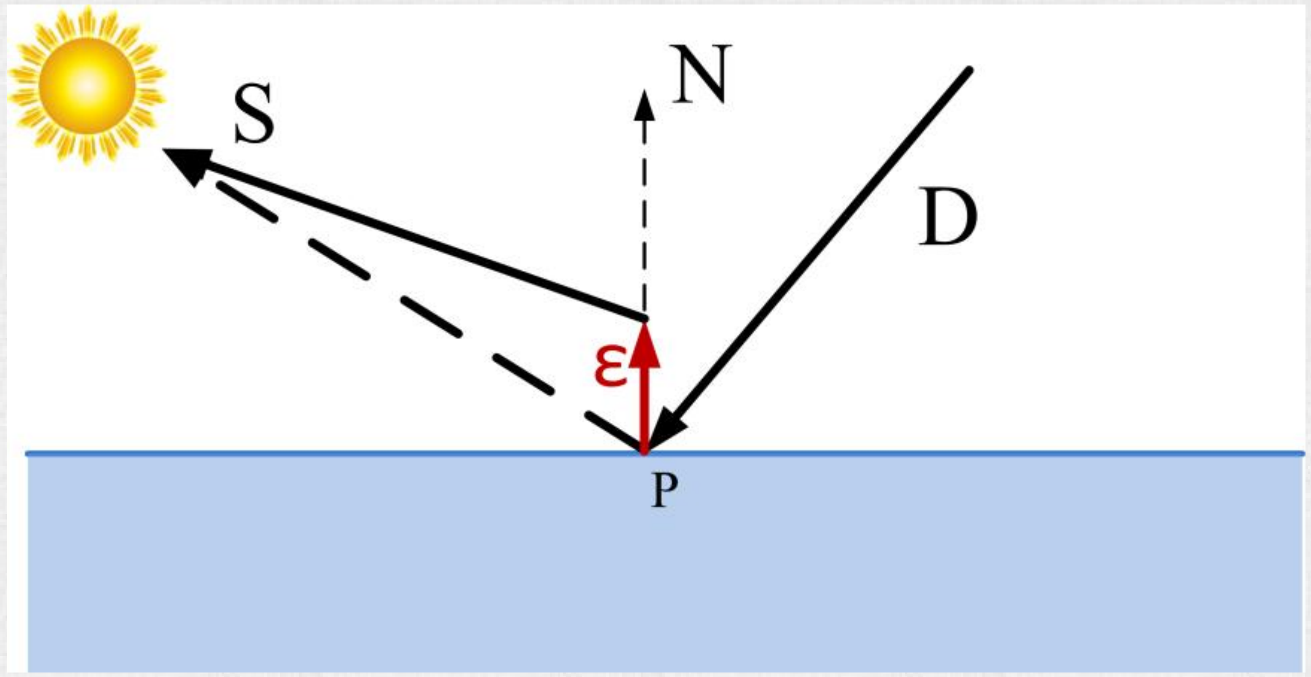
Reflected Ray

- Given an incoming ray $R(t) = A + Dt$, and (outward) unit normal $\widehat{N}$, the angle of incidence is defined via $D \cdot \widehat{N} = -\|D\|_2 \cos\theta_i$
- Mirror reflection: incoming/outgoing rays make the same angle with $\widehat{N}$, i.e. $\theta_o = \theta_i$
  - Note: all the rays and the normal are all coplanar
- Reflected ray direction: $D_{reflect} = D - 2(D \cdot \widehat{N})\widehat{N}$
- Reflected ray: $R_{reflect}(t) = R_o + D_{reflect}t$


$D_{reflect}$
$\widehat{N}$
θ_o
θ_i
D
$-(D \cdot \widehat{N})\widehat{N}$
$-(D \cdot \widehat{N})\widehat{N}$
D

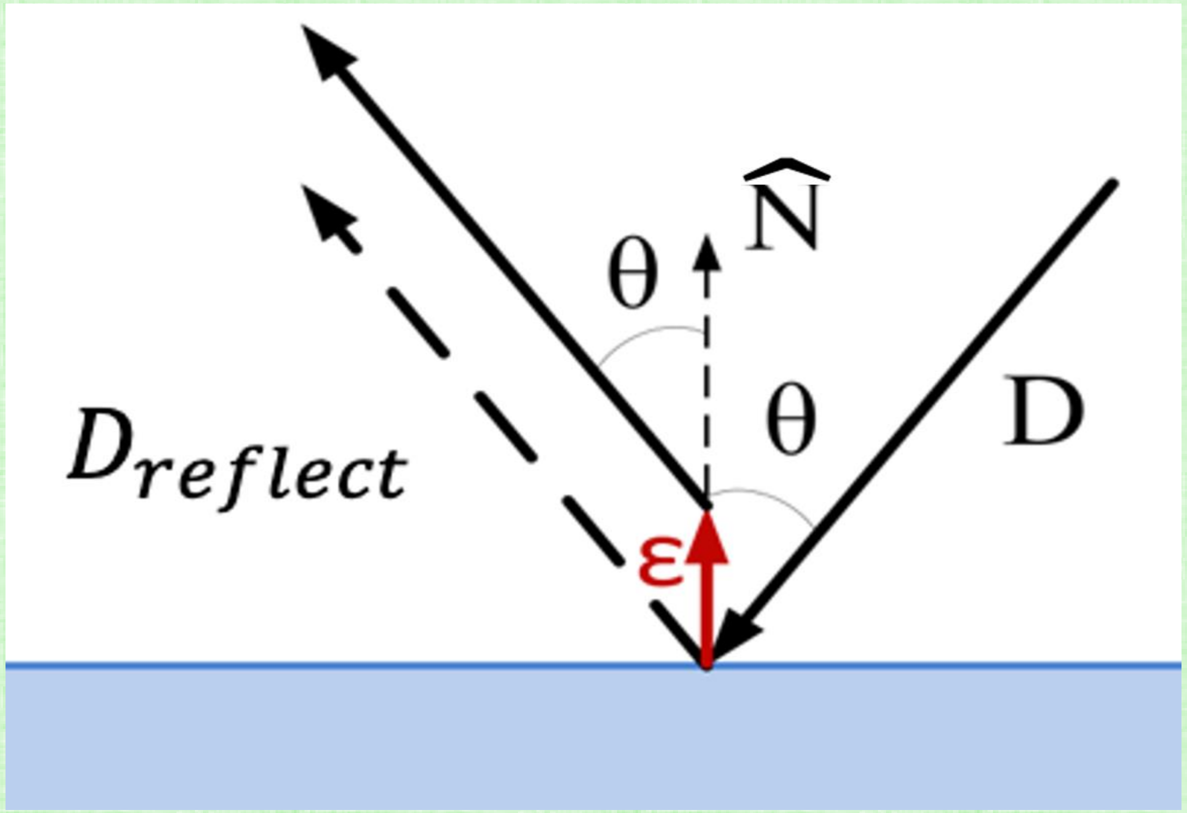
Recall: Fixing Spurious Self-Occlusion

- Perturb the starting point of the shadow ray (typically in the normal direction), i.e. from R_o to $R_o + \epsilon\hat{N}$
- The ray direction needs to be modified too, to go from $R_o + \epsilon\hat{N}$ to the light
- The new shadow ray is $S(t) = R_o + \epsilon\hat{N} - \hat{L}_{mod}t$
- Need to be careful that the new starting point isn't inside (or too close to) any other geometry

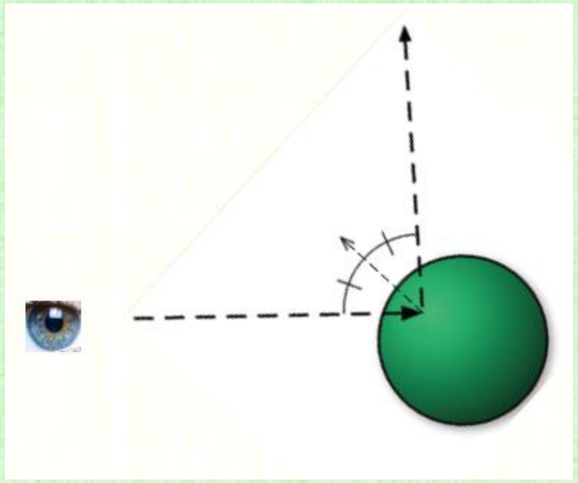


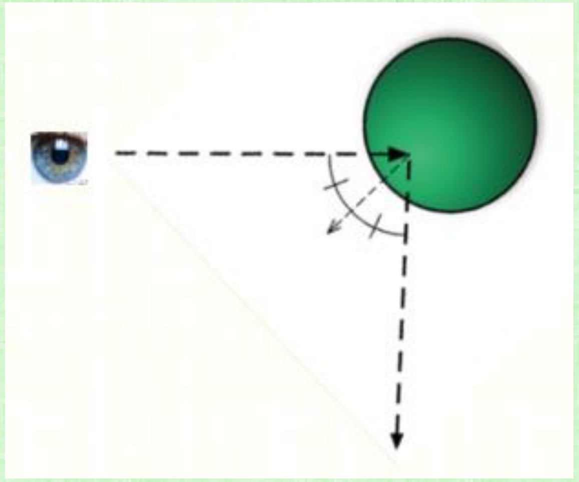
Spurious Self-Occlusion

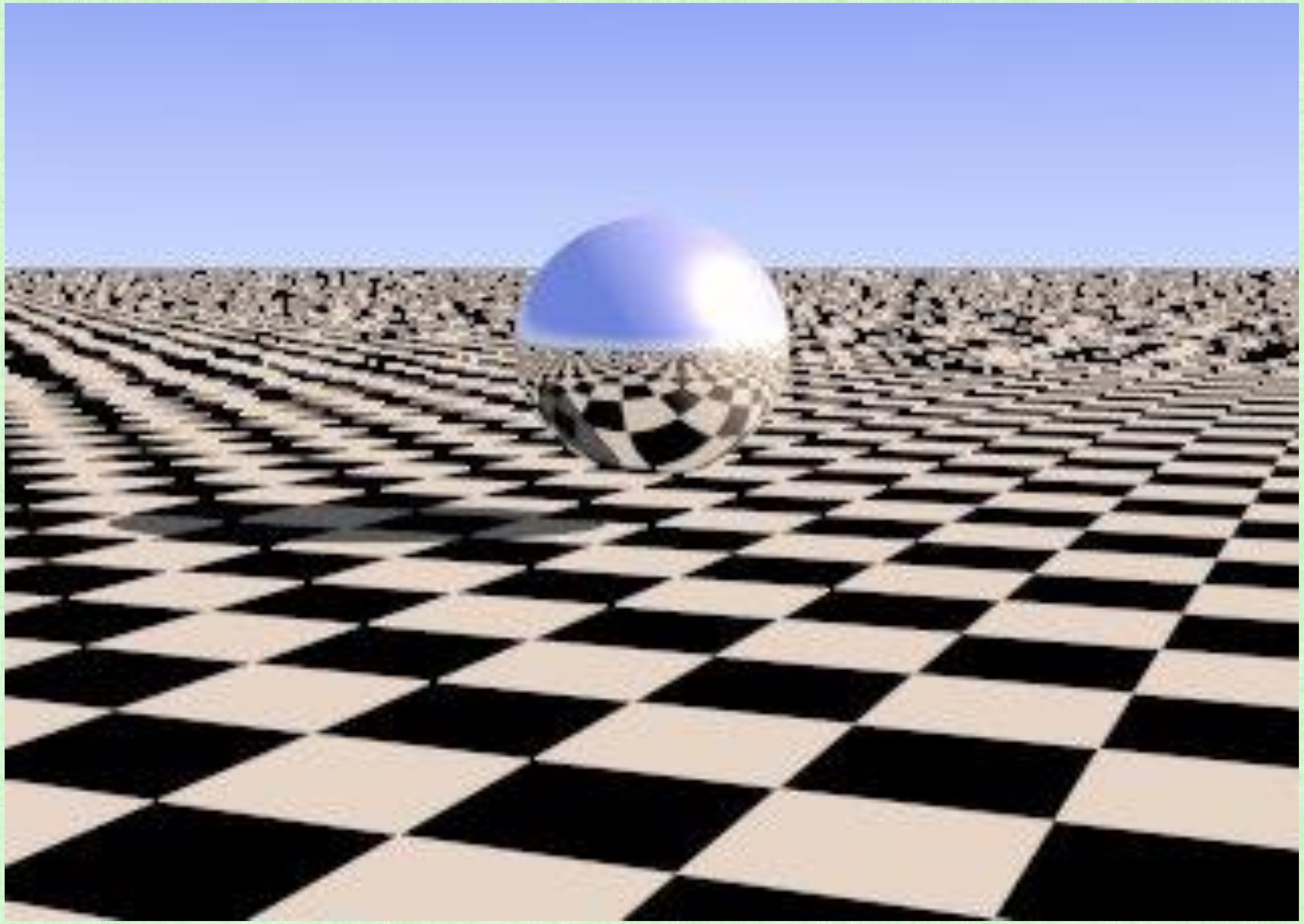
- Perturb the starting point of the reflected ray to $R_o + \epsilon\widehat{N}$
- The ray direction does not need to be modified (dissimilar to shadow rays)
- The new reflected ray is $R_{reflect}(t) = R_o + \epsilon\widehat{N} + D_{reflect}t$
- Need to be careful that the new starting point isn't inside (or too close to) any other geometry



Reflections





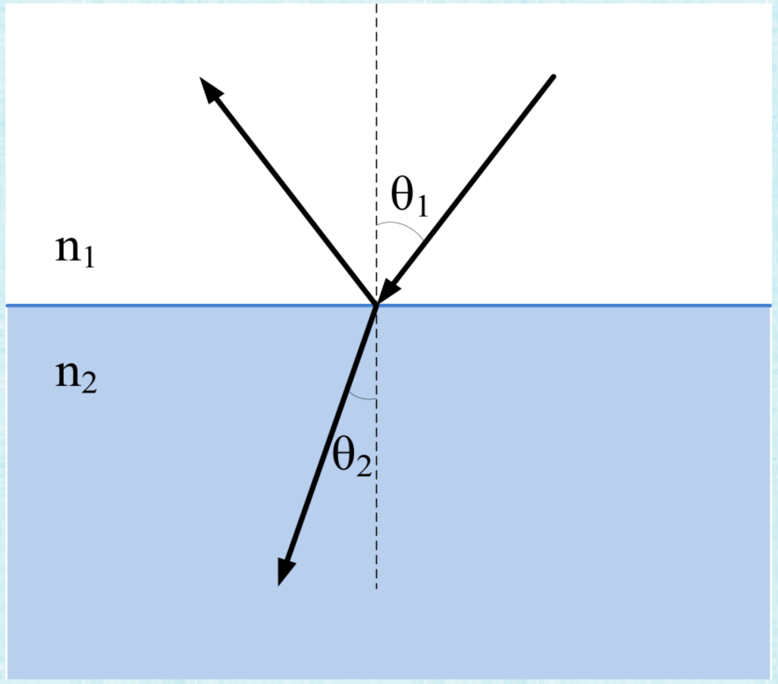


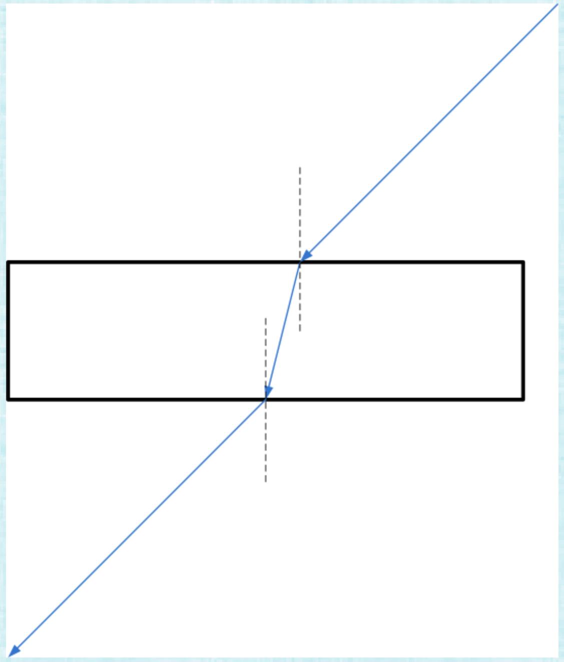
Transmission

- The angle of incidence and angle of transmission (or refraction) are related via Snell's Law:

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

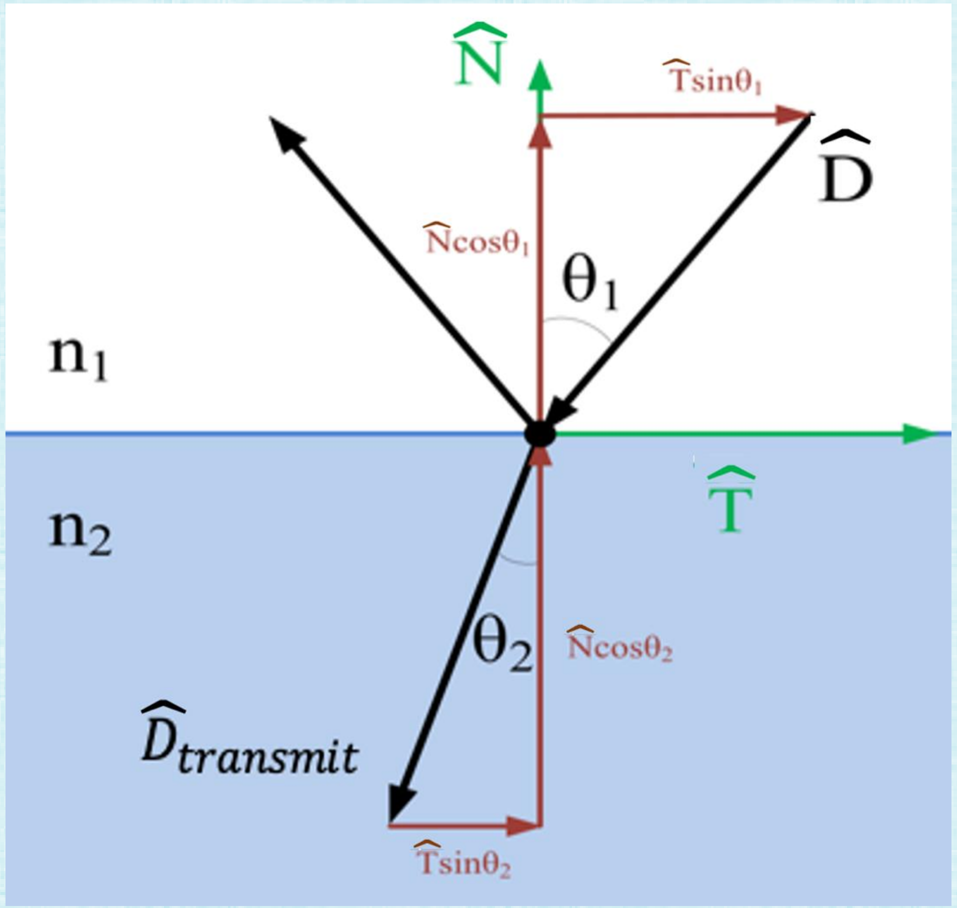
- Incoming/outgoing angles: θ_1, θ_2; phase velocities: v_1, v_2; indices of refraction: n_1, n_2





Geometry for Computing the Transmitted Ray

- Let $\hat{D}$ be the (unit) incoming ray direction, $\hat{N}$ be the (outward) unit normal, and $\hat{T}$ be the unit tangent in the plane of $\hat{D}$ and $\hat{N}$; then, $\hat{D} + \hat{N}cos\theta_1 + \hat{T}sin\theta_1 = 0$
- Let $\hat{D}_{transmit}$ be the (unit) transmitted ray direction; then $\hat{D}_{transmit} + \hat{T}sin\theta_2 + \hat{N}cos\theta_2 = 0$



Algebra for Computing the Transmitted Ray

- $\hat{D}_{transmit} = -\hat{T}sin\theta_2 - \hat{N}cos\theta_2 = (\hat{D} + \hat{N}cos\theta_1)\frac{sin\theta_2}{sin\theta_1} - \hat{N}\sqrt{1 - \sin^2\theta_2}$
- Using Snell's Law: $\hat{D}_{transmit} = (\hat{D} + \hat{N}cos\theta_1)\frac{n_1}{n_2} - \hat{N}\sqrt{1 - \left(\frac{n_1}{n_2}sin\theta_1\right)^2}$
- $\hat{D}_{transmit} = \hat{D}\frac{n_1}{n_2} + \hat{N}\left(\frac{n_1}{n_2}cos\theta_1 - \sqrt{1 - \left(\frac{n_1}{n_2}\right)^2(1 - \cos^2\theta_1)}\right)$
- $cos\theta_1 = -\hat{D}\cdot\hat{N}$ gives $\hat{D}_{transmit} = \hat{D}\frac{n_1}{n_2} - \hat{N}\left(\frac{n_1}{n_2}\hat{D}\cdot\hat{N} + \sqrt{1 - \left(\frac{n_1}{n_2}\right)^2\left(1 - \left(\hat{D}\cdot\hat{N}\right)^2\right)}\right)$

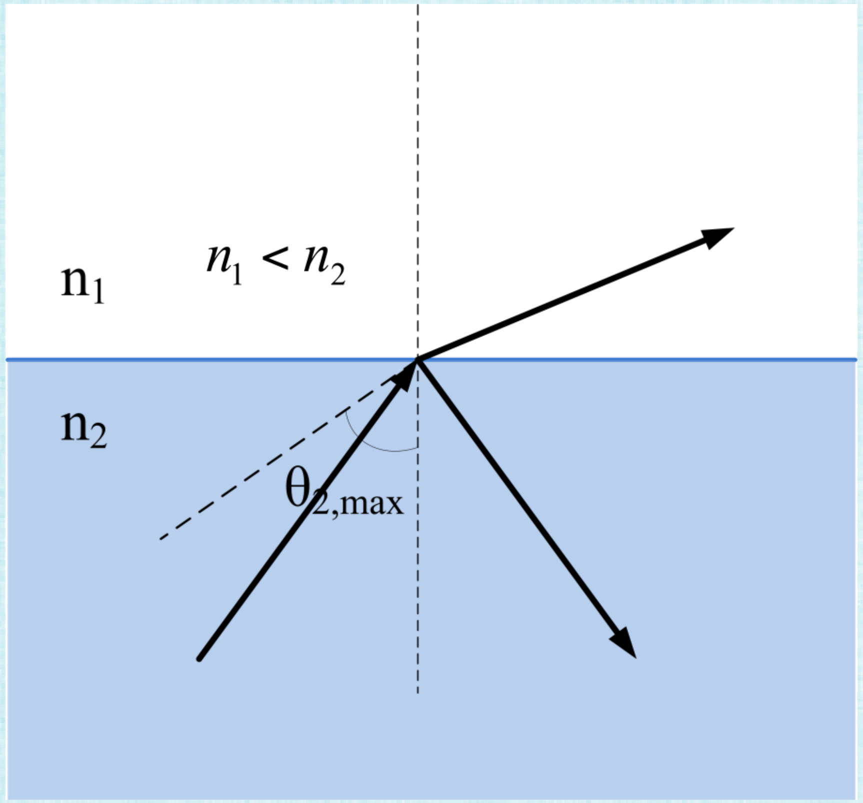
Total internal reflection: when the term under the square root is negative, there is no transmitted ray

Notes:

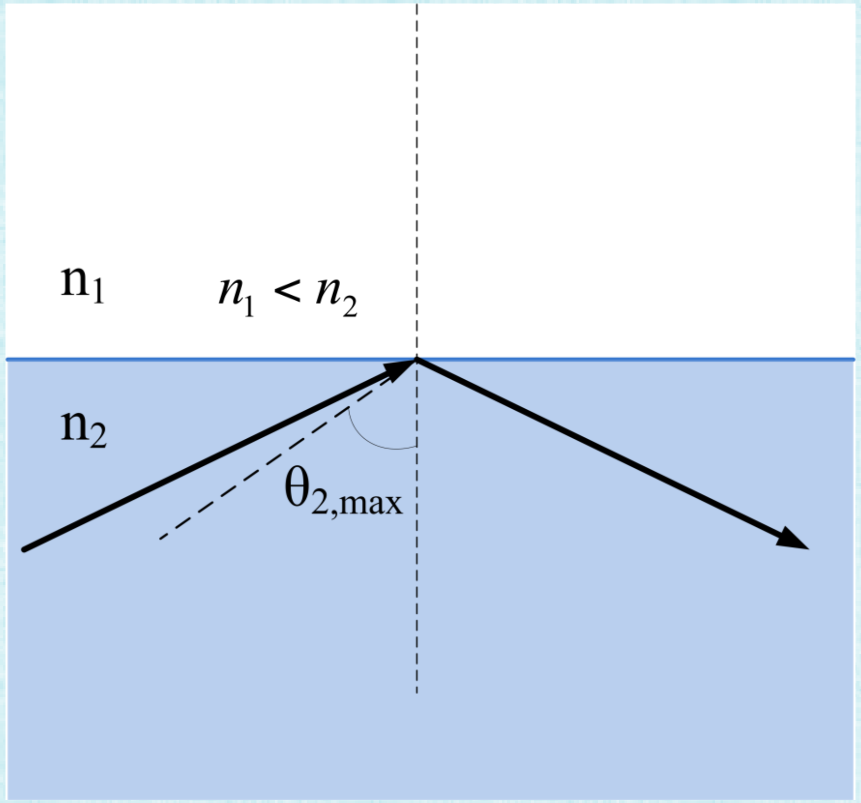
- This equation works regardless of whether n_1 or n_2 is bigger
- Add $\epsilon > 0$ to avoid self intersection, or offset in the negative normal direction (while avoiding other nearby geometry, etc.)

Total Internal Reflection

- When looking from a higher index of refraction material towards a lower index of refraction material, there is no light transmission when the incident angle exceeds a critical angle
- In such a case, all the light comes from the reflected ray



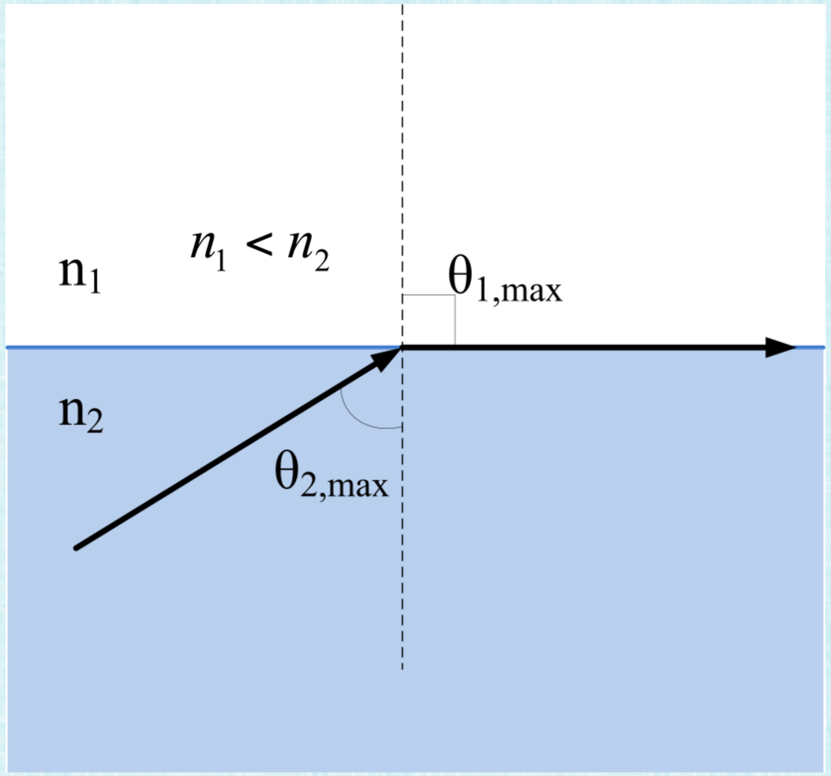
when $\theta_2 < \theta_{2,\max}$, both reflection and transmission occur



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

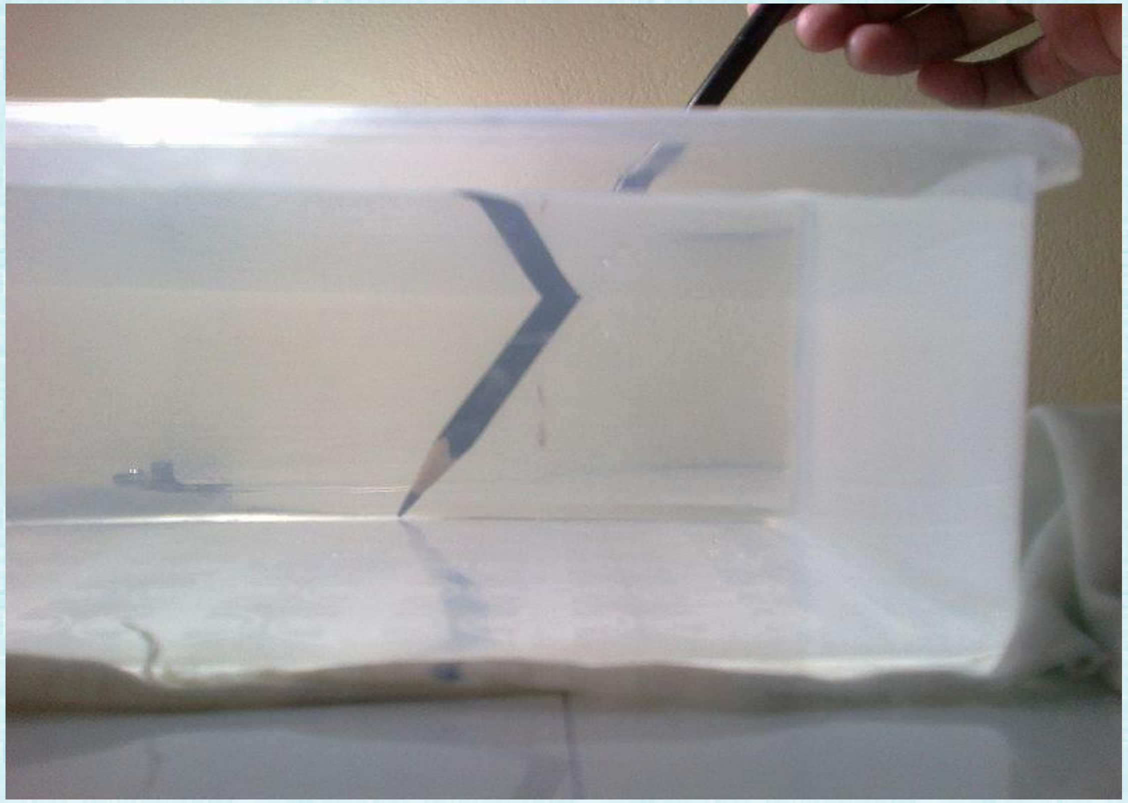
Critical Angle

- $\theta_1 = \frac{\pi}{2}$ is the maximum angle for transmission
- $\sin\left(\frac{\pi}{2}\right) = 1$; so, Snell's Law becomes $\frac{1}{\sin\theta_2} = \frac{n_2}{n_1}$ or $\theta_2 = \arcsin\left(\frac{n_1}{n_2}\right)$
  - This can only occur when $n_1 < n_2$



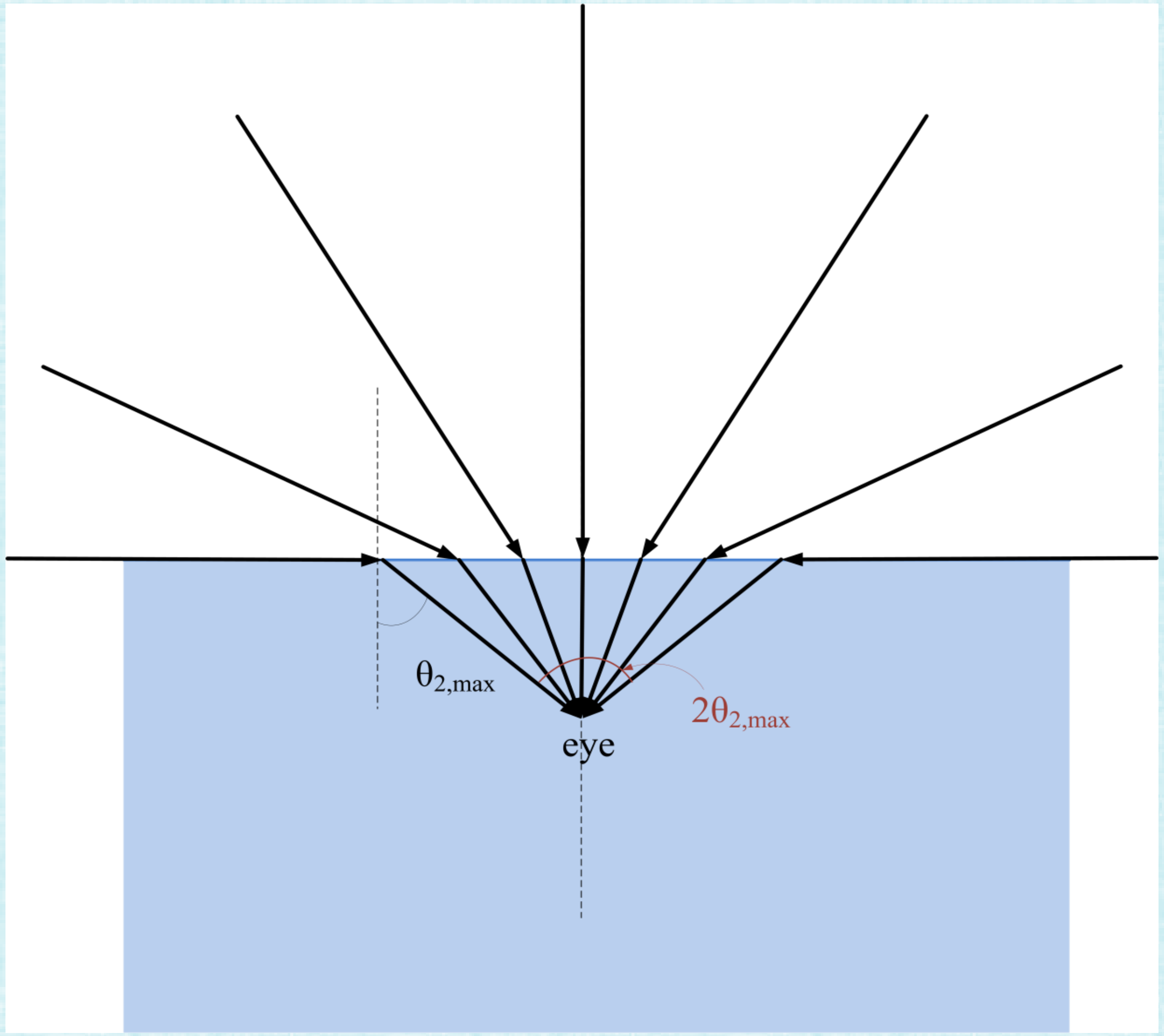
Total Internal Reflection

- Responsible for many interesting and impressive visuals in both glass and water

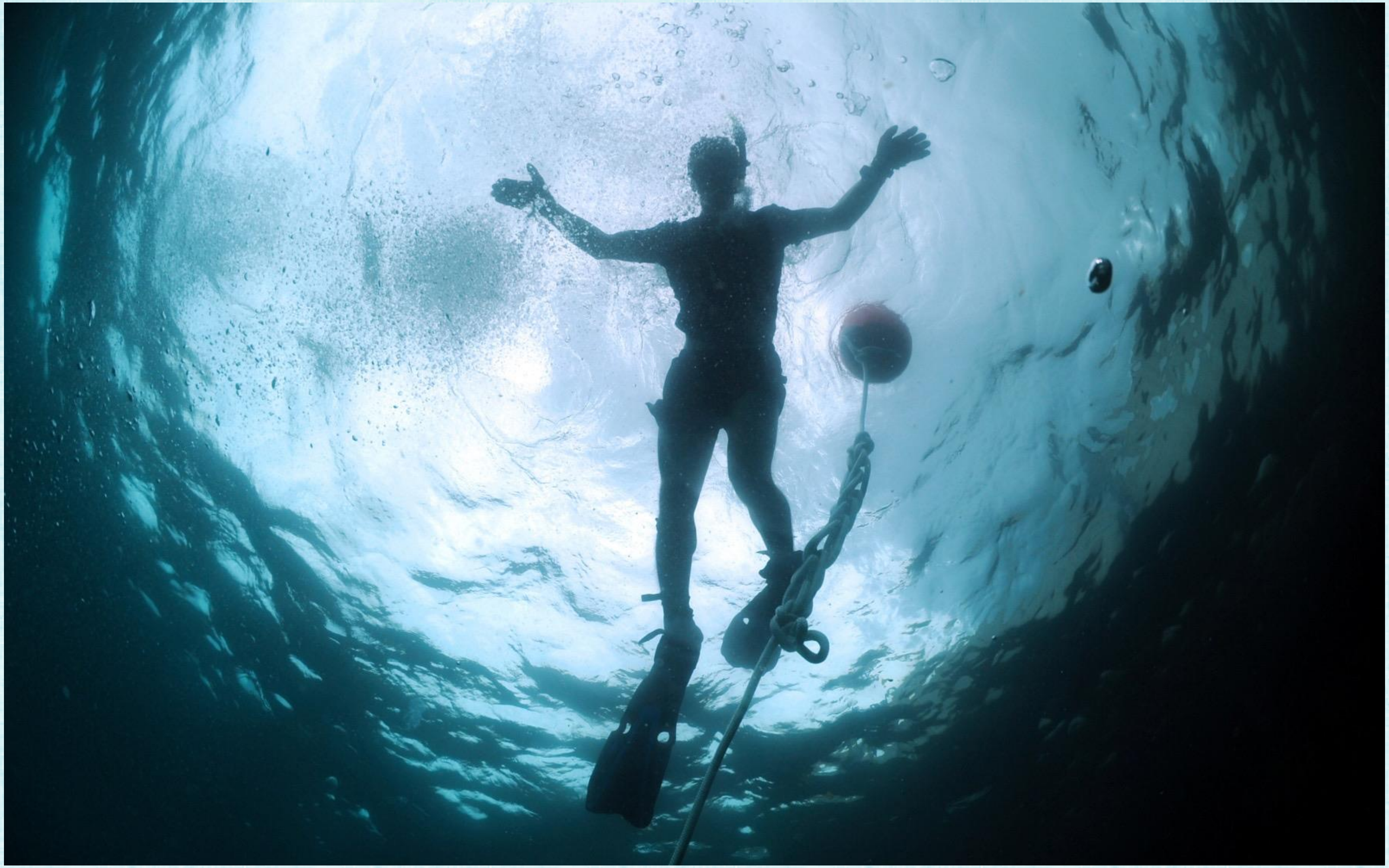


Snell's Window

- Yes, fish can see you standing on the shore!

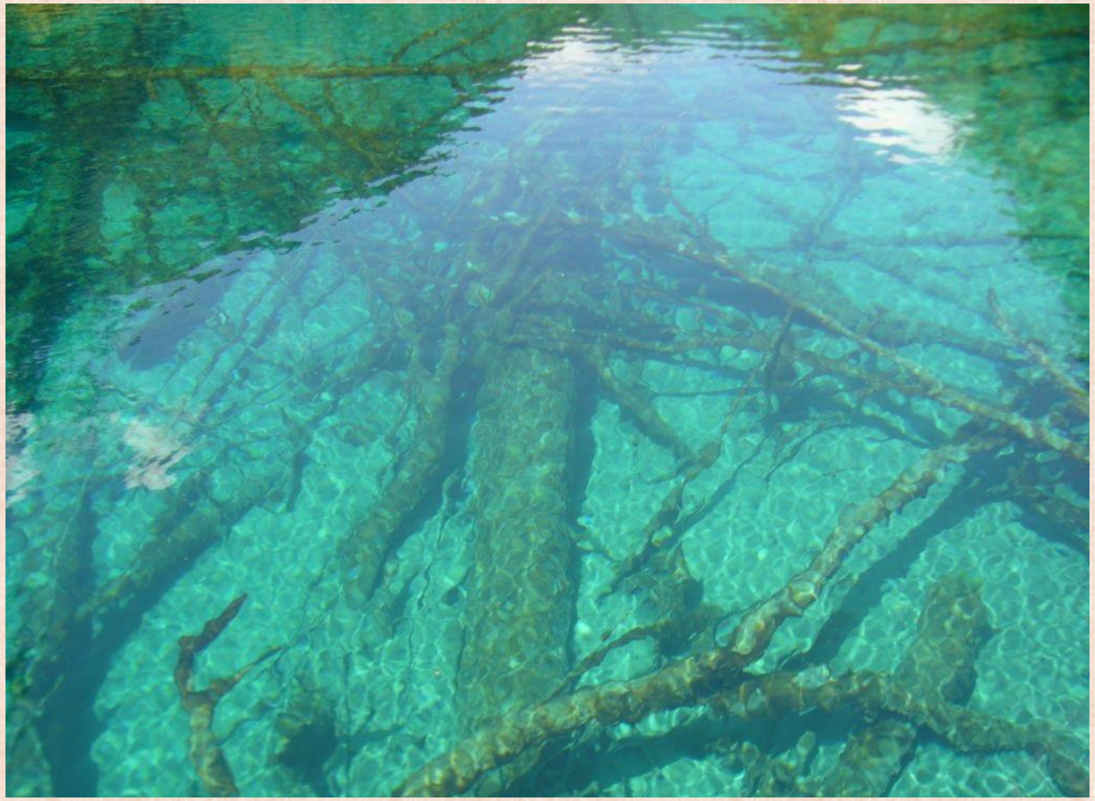


Snell’s Window

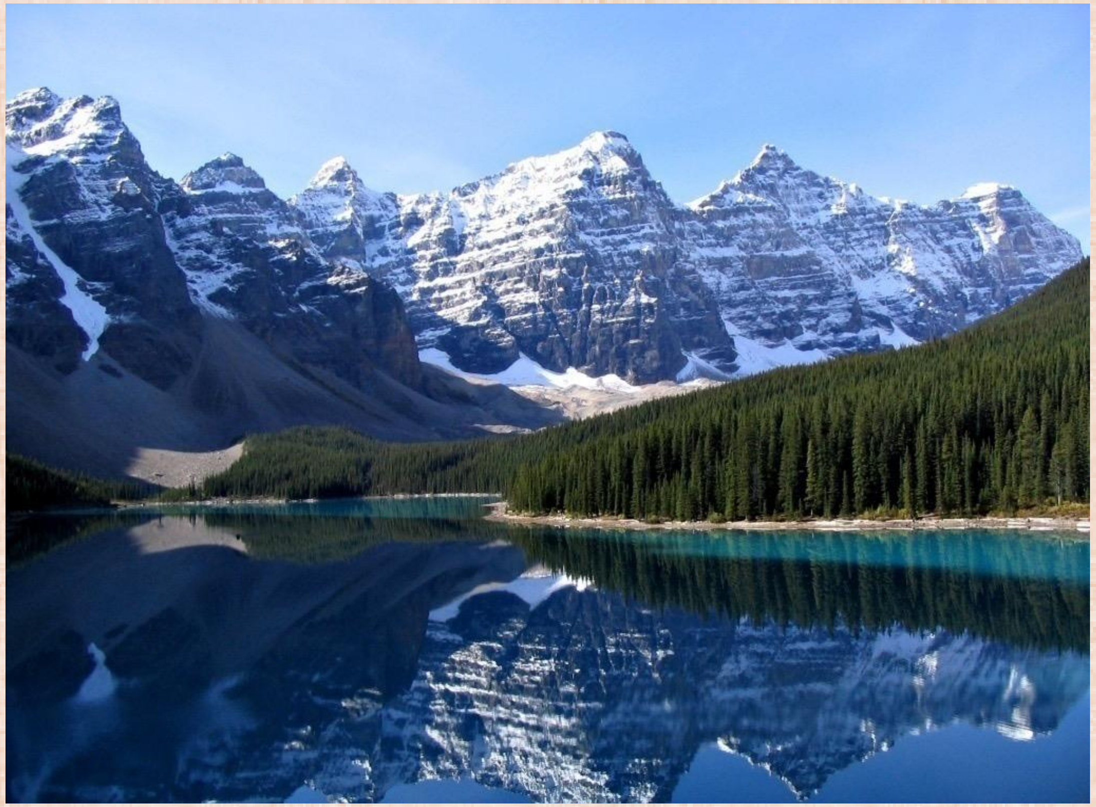


Viewing Angle

- As the viewing angle varies from perpendicular (overhead) to parallel (grazing), transmission decreases and reflection increases



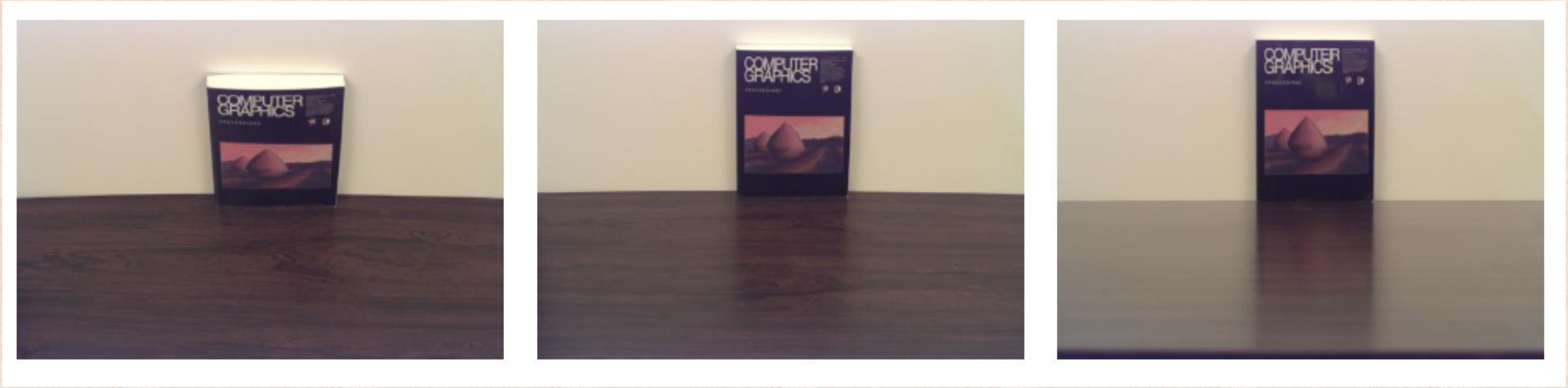
Perpendicular (overhead) view:
more transmission, less reflection



Parallel (grazing) view:
more reflection, less transmission

Viewing Angle (Opaque Surfaces)

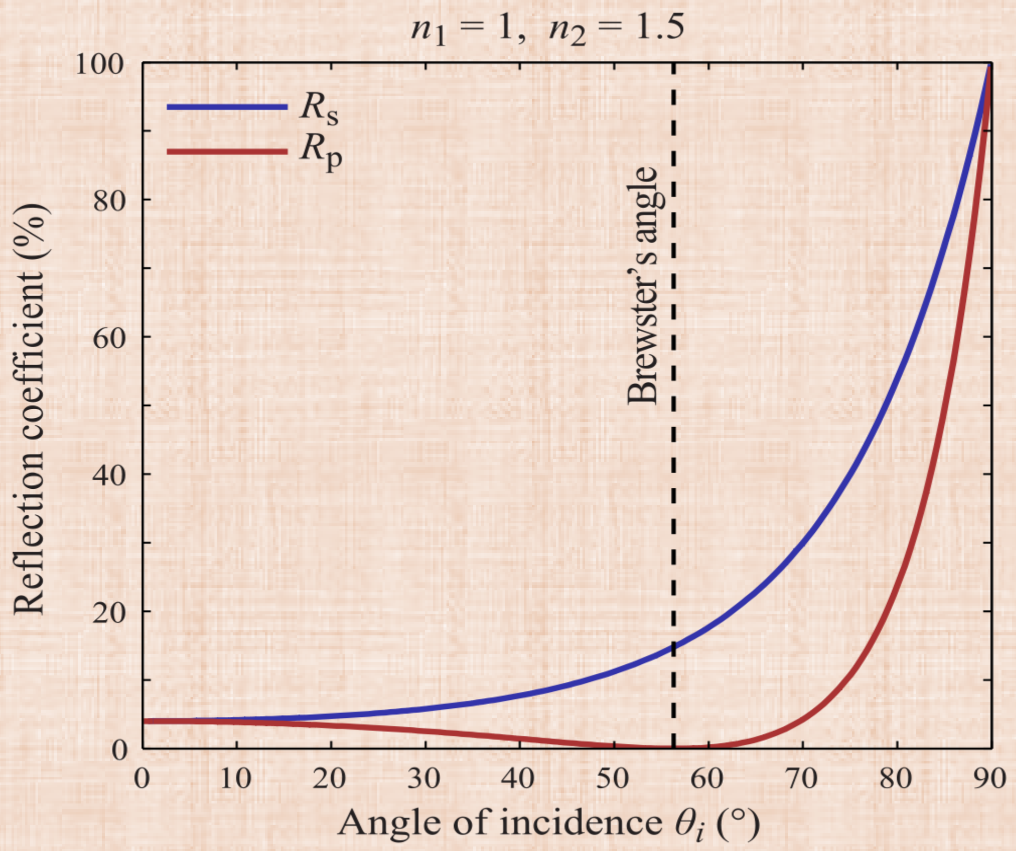
- Even for opaque surfaces that lack transmission, reflection varies based on the viewing angle



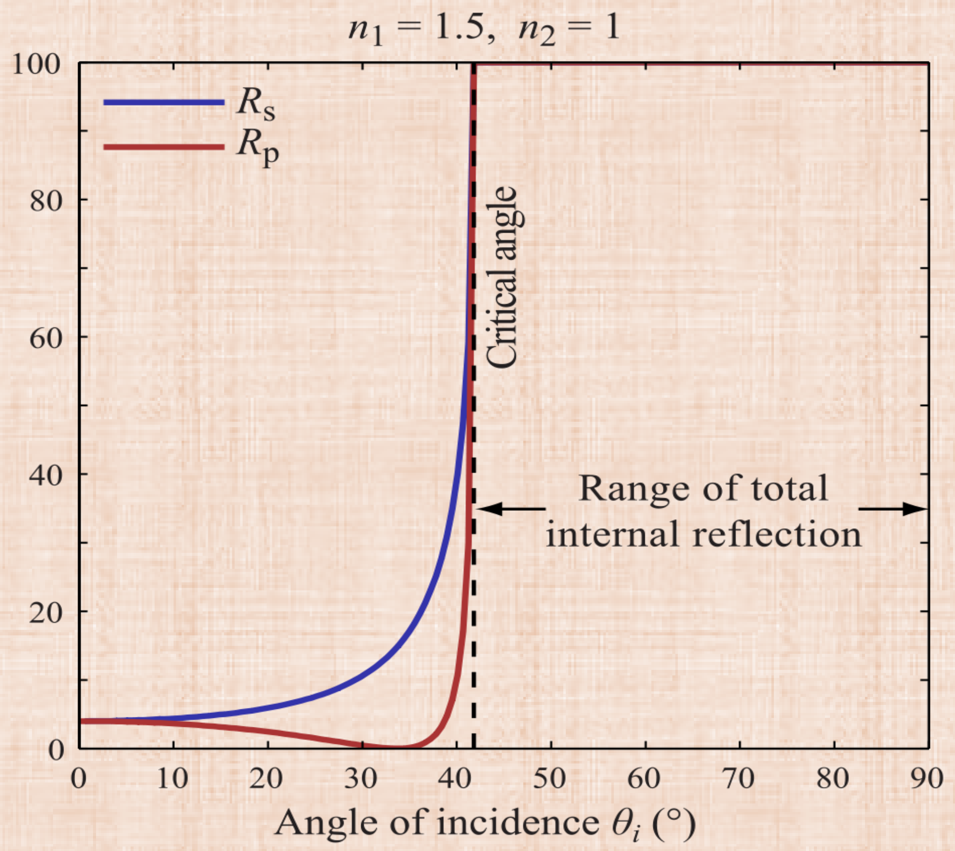
As the viewing angle changes from overhead to grazing (from left to right), the amount of reflection off of the table increases (as illustrated the book's reflection on the table)

Fresnel Equations

- The amount of reflection increases as the viewing angle goes from perpendicular (coincident with the normal) to parallel (orthogonal to the normal)



Looking from air towards water



Looking from water towards air

Light polarization is based on whether the plane containing the incident, reflected, refracted rays is parallel (p-polarized) or perpendicular (s-polarized) to the electric field

Fresnel Equations

- The Fresnel equations approximate the fraction of light reflected as θ_i :

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

p-polarized light

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

s-polarized light

θ_i - incident angle
θ_t - transmission angle

- Transmission (if it occurs) is calculated as the remaining light:

$$T_p = 1 - R_p$$

$$T_s = 1 - R_s$$

- For <u>unpolarized</u> light (a typical assumption in ray tracing), assume:

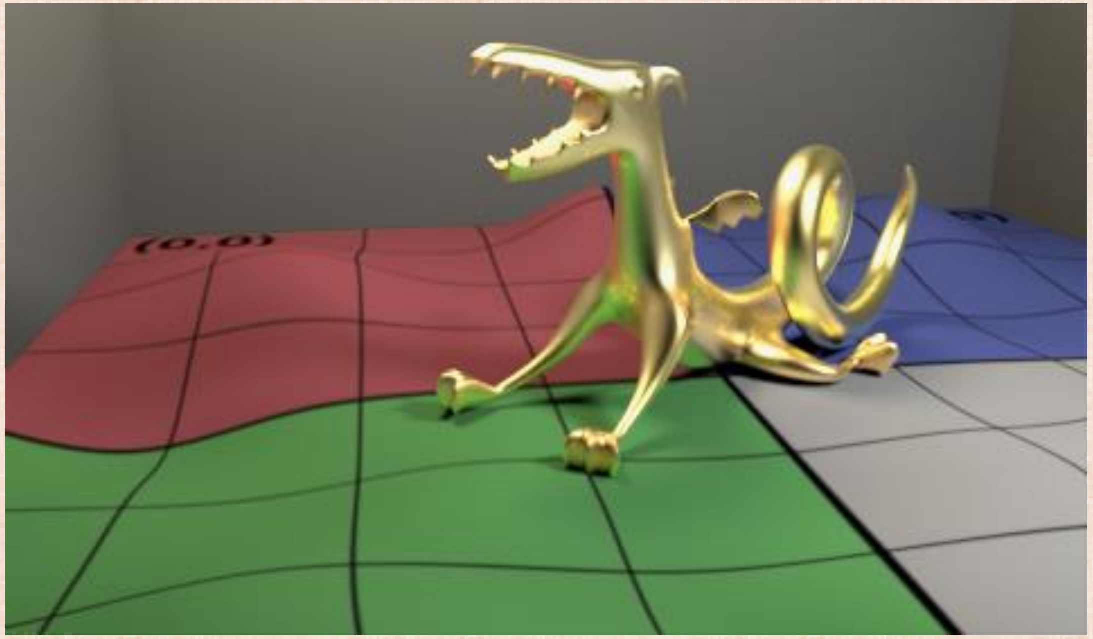
$$R = \frac{R_p + R_s}{2}$$

$$T = 1 - R$$

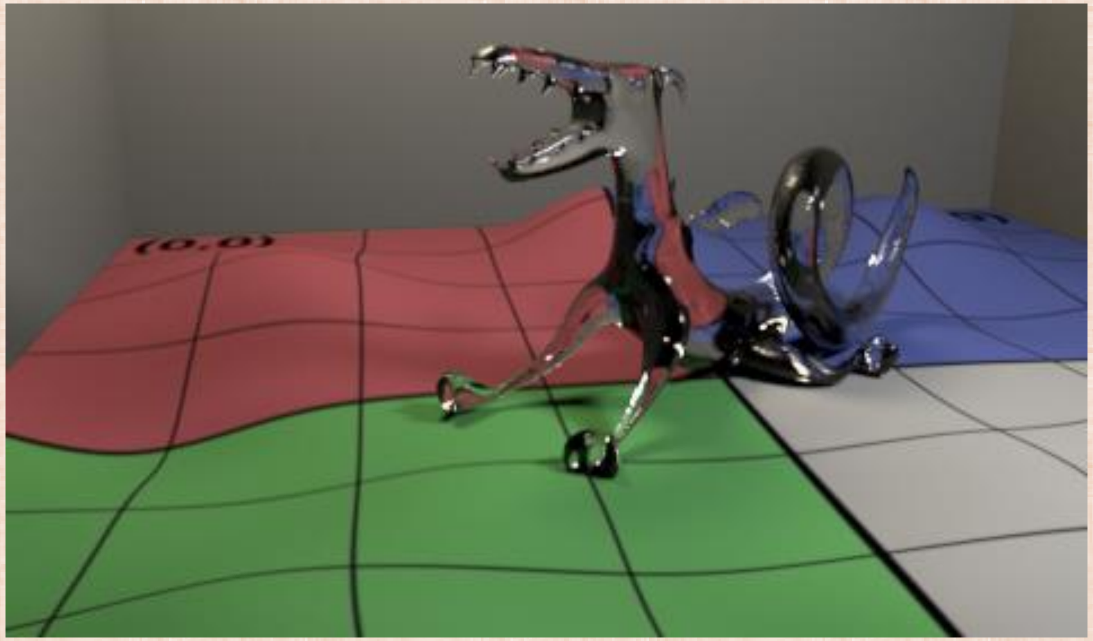
Conductors vs. Dielectrics

- Conductors (of electricity, e.g. metals) mostly reflect light (low absorption, no transmission)
  - Moreover, the amount reflected doesn't change much with viewing angle
  - Thus, k_r can be approximated as a constant (independent of viewing direction)

- Dielectrics (e.g. glass, water) typically have a k_r that varies significantly with viewing angle

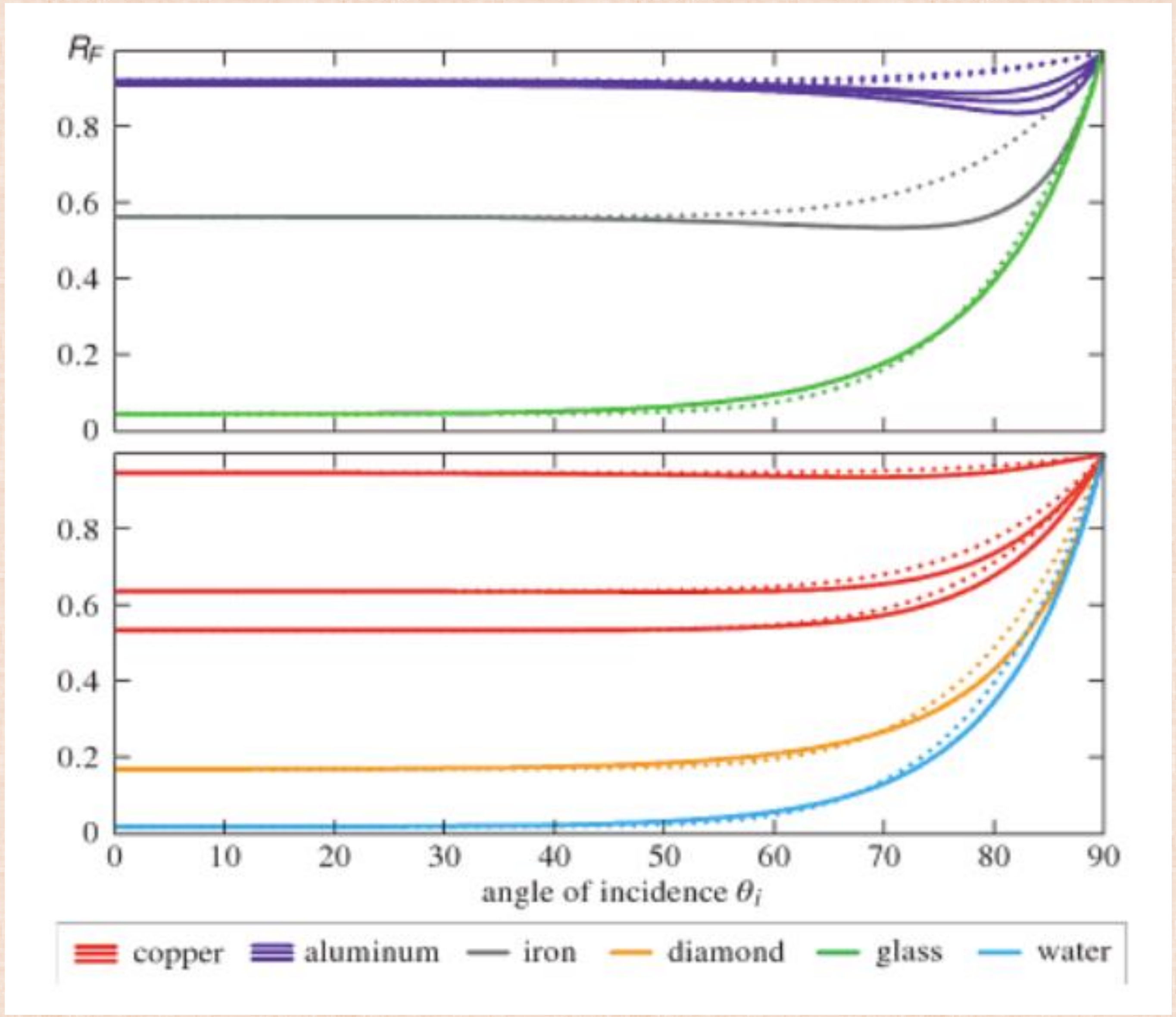


Conductor



Dielectric

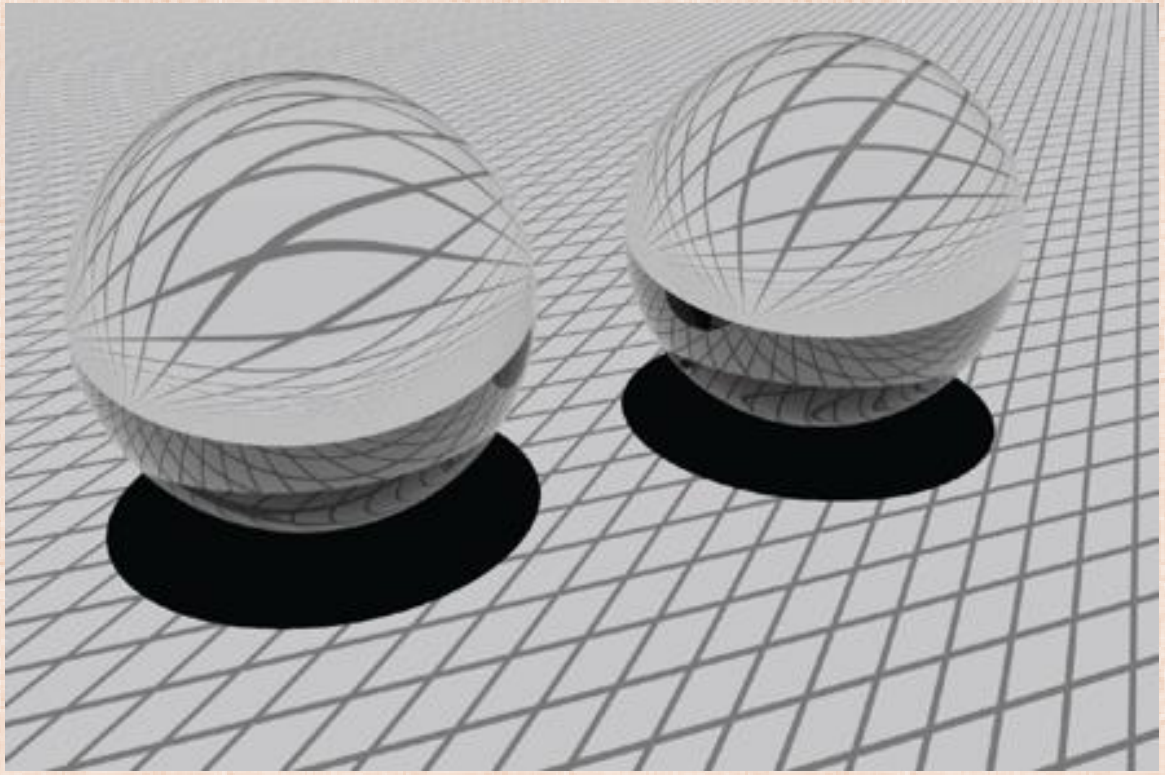
Examples

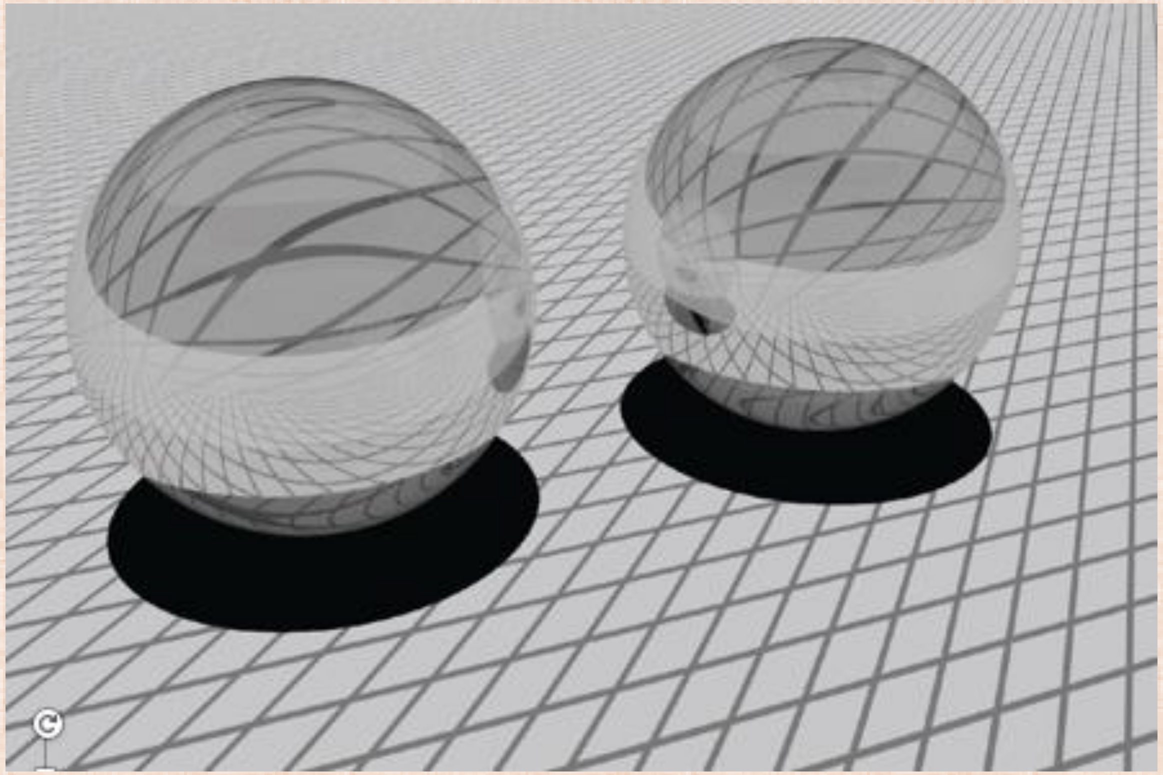


Fresnel (solid lines)
Schlick (dotted lines)

Curved Surfaces

- The viewing angle can vary (from perpendicular to parallel) across the surface of an object
- Thus, the amount of reflection vs. transmission can similarly vary
- Capturing this effect is especially important for dielectrics


Correct reflection vs. transmission
(based on viewing angle)

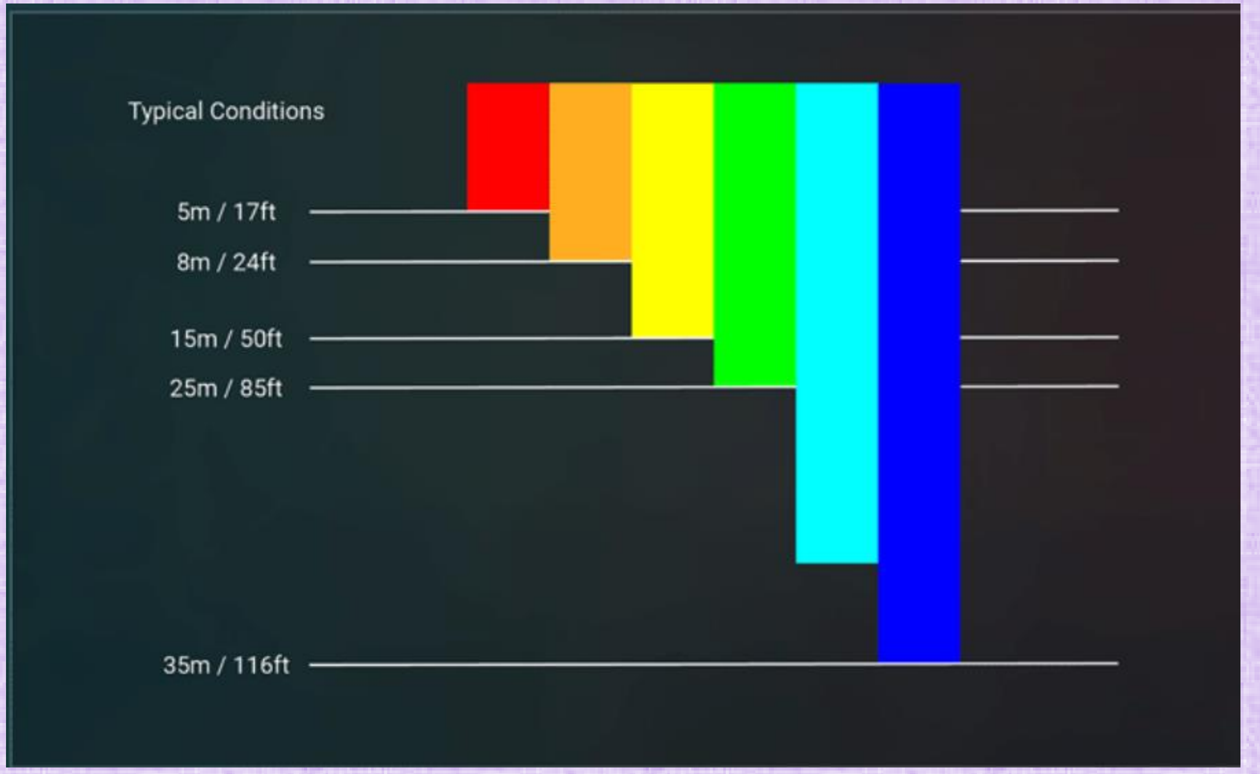

Incorrect reflection vs. transmission
(no dependence on viewing angle)

Attenuation

- Light is absorbed and scattered as it travels through a material
- This attenuates the light traveling along a ray
- The amount of attenuation depends on the distance traveled (through the material)
- Different colors (actually, different wavelengths) are attenuated at different rates

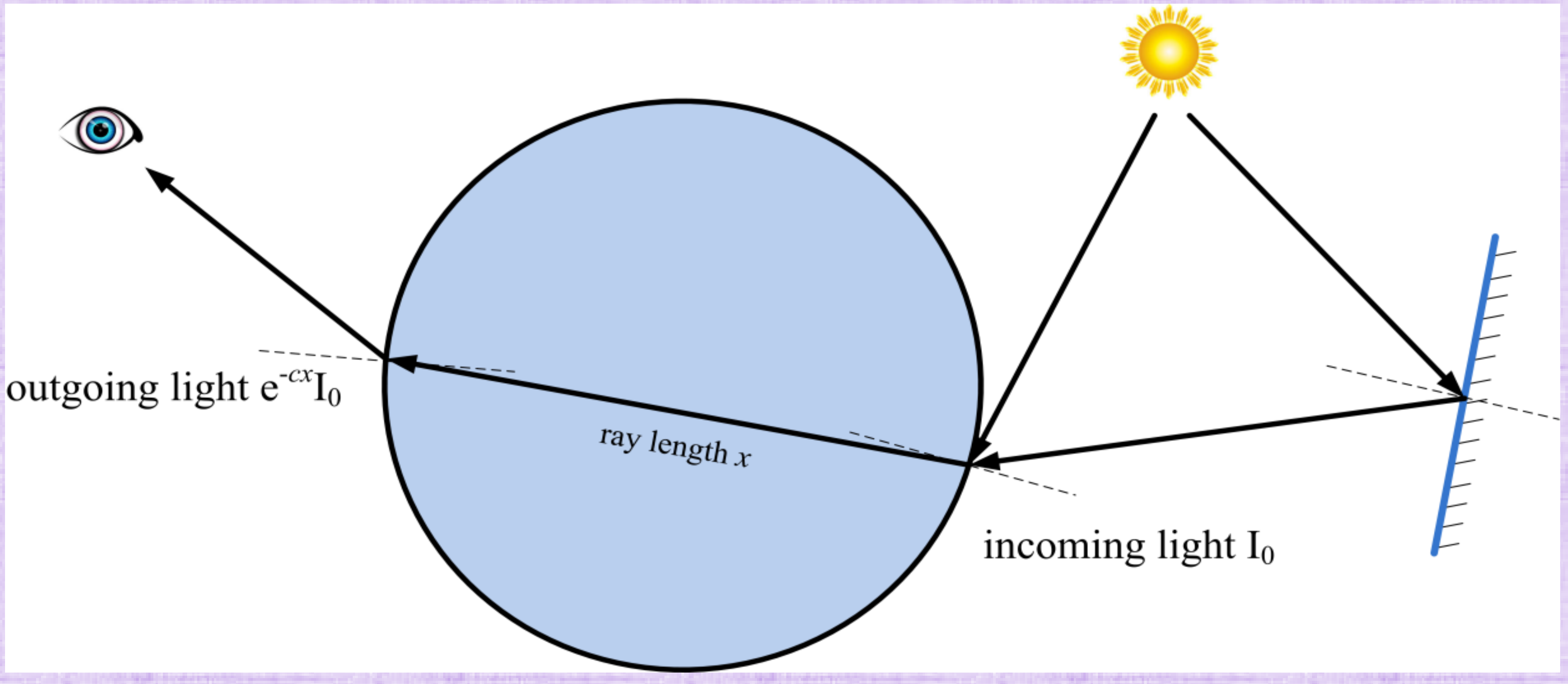
Example:

- Shallow water is clear (almost no attenuation)
- Deeper water attenuates all the red light and looks bluish-green
- Even deeper water attenuates all the green light too, and looks dark blue
- Eventually all the light attenuates, and the color ranges from blackish-blue to black



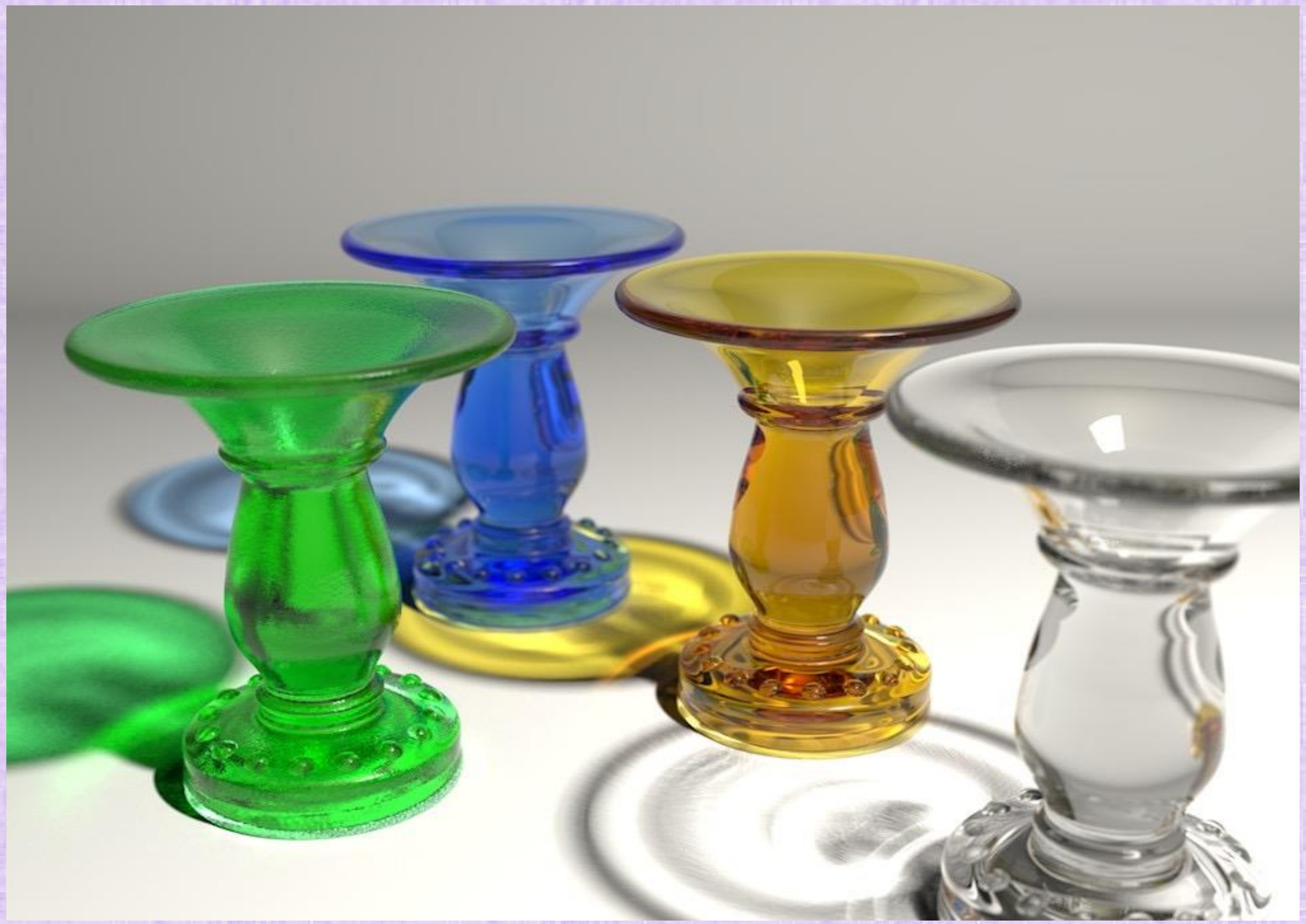
Beer’s Law

- For homogeneous media, attenuation can be approximated by Beer’s Law
- Light with intensity I is attenuated over a distance x via the Ordinary Differential Equation (ODE): $\frac{dI}{dx} = -cI$ where c is the attenuation coefficient
- This ODE has an exact solution: $I(x) = I_o e^{-cx}$ where I_o is the original amount of light



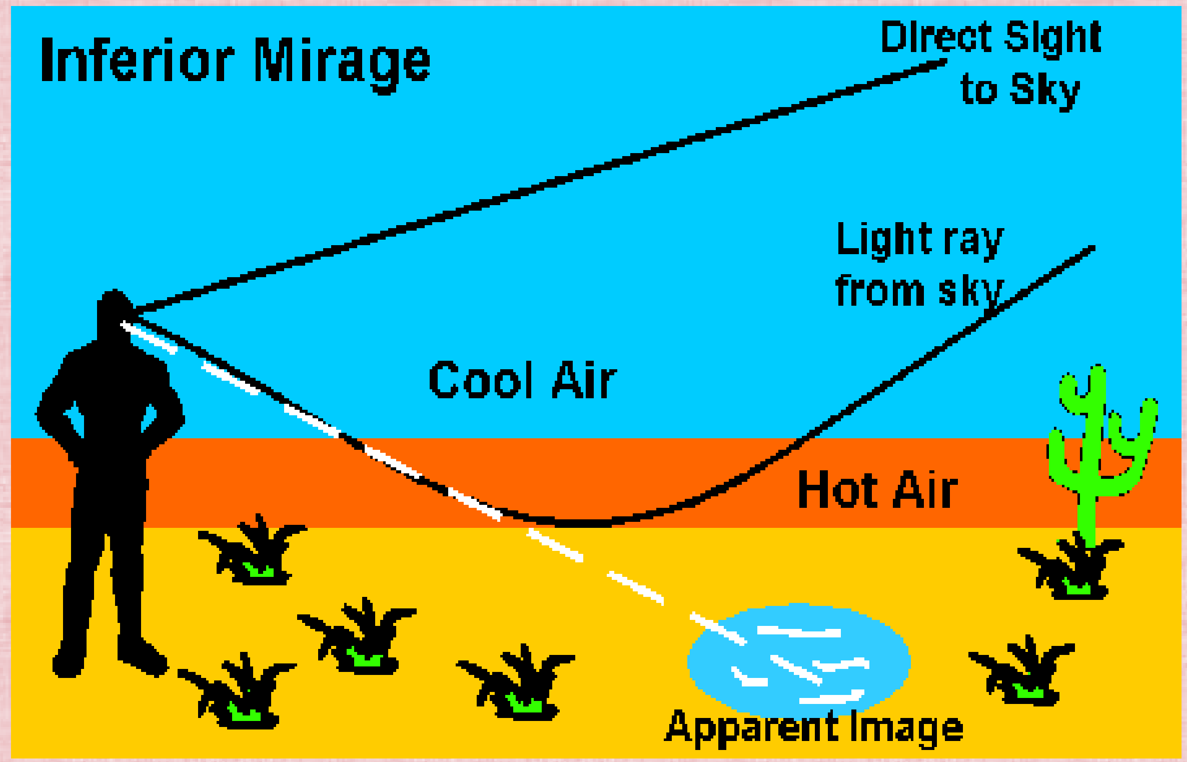
Beer’s Law

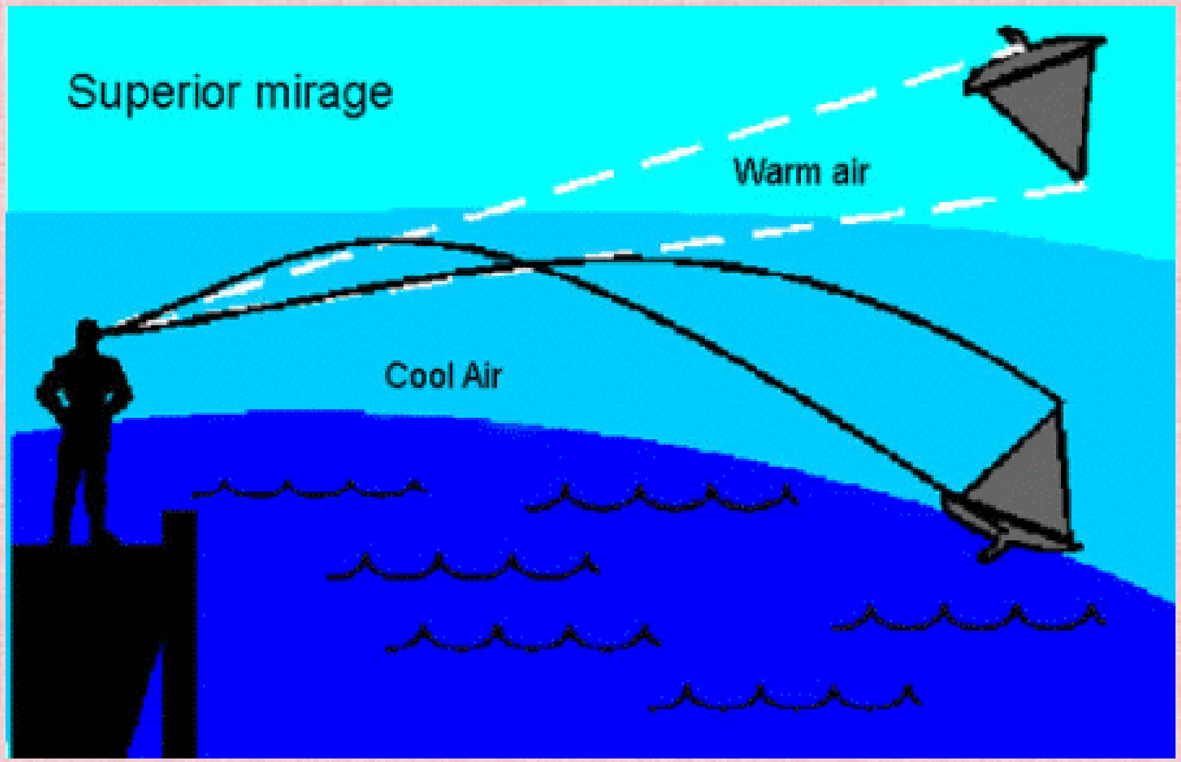
- The color of a transparent object can be described by three attenuation coefficients: c_R, c_G, c_B
- Shadow rays are also attenuated



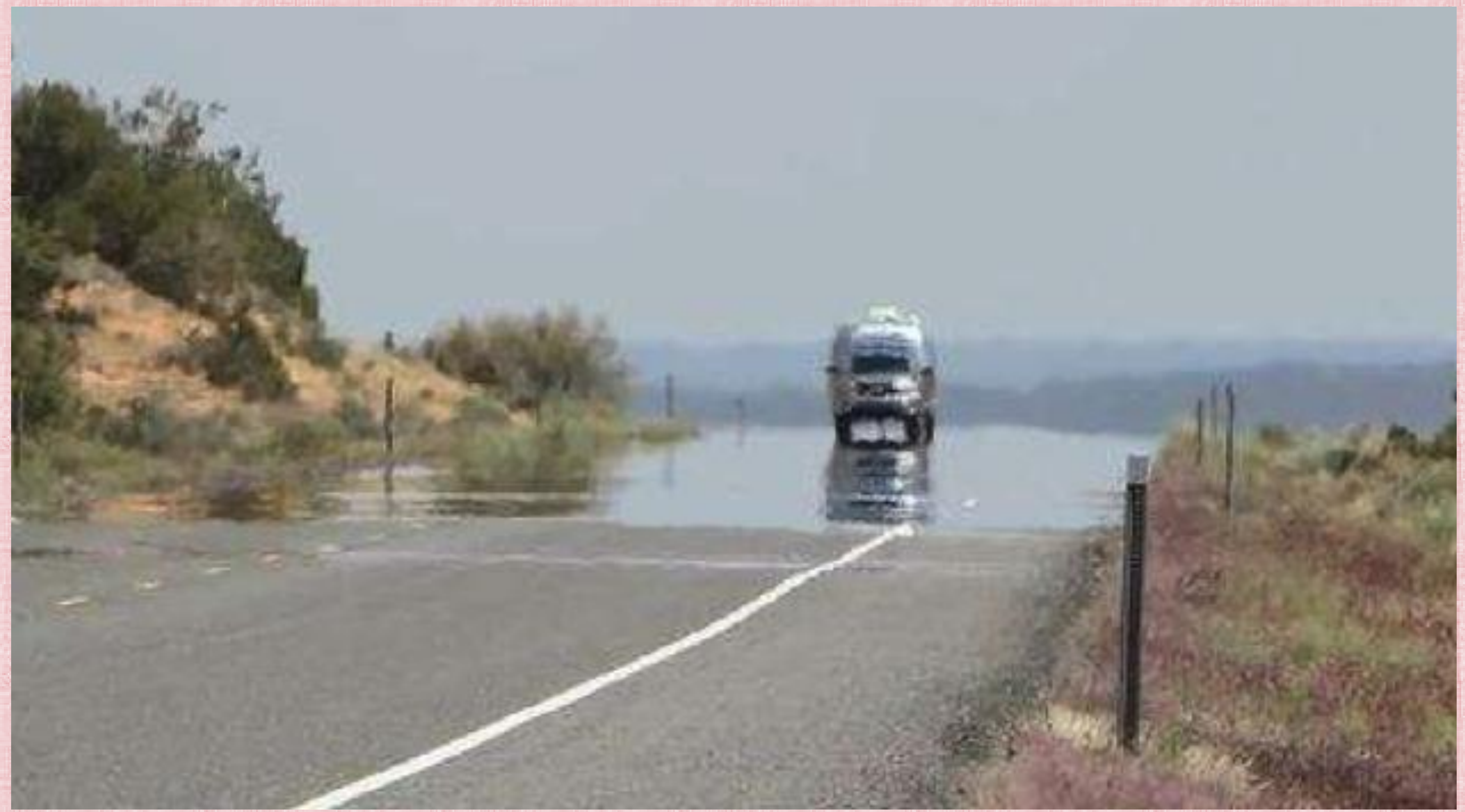
Atmospheric Refraction

- As light passes though air with varying temperature (and thus varying density, and varying index of refraction), it bends continuously
- This causes light to follow a curved path

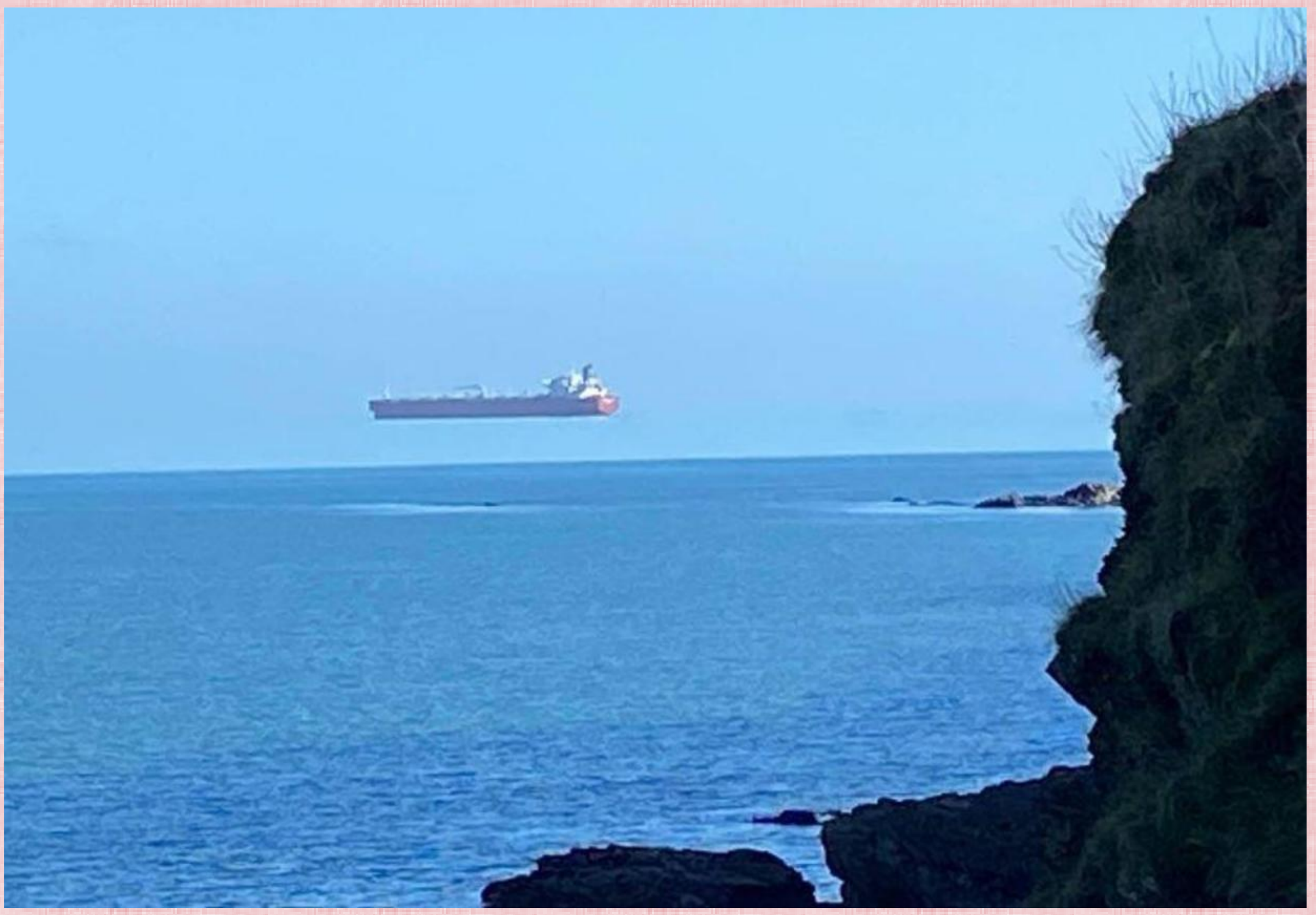




Inferior Mirage

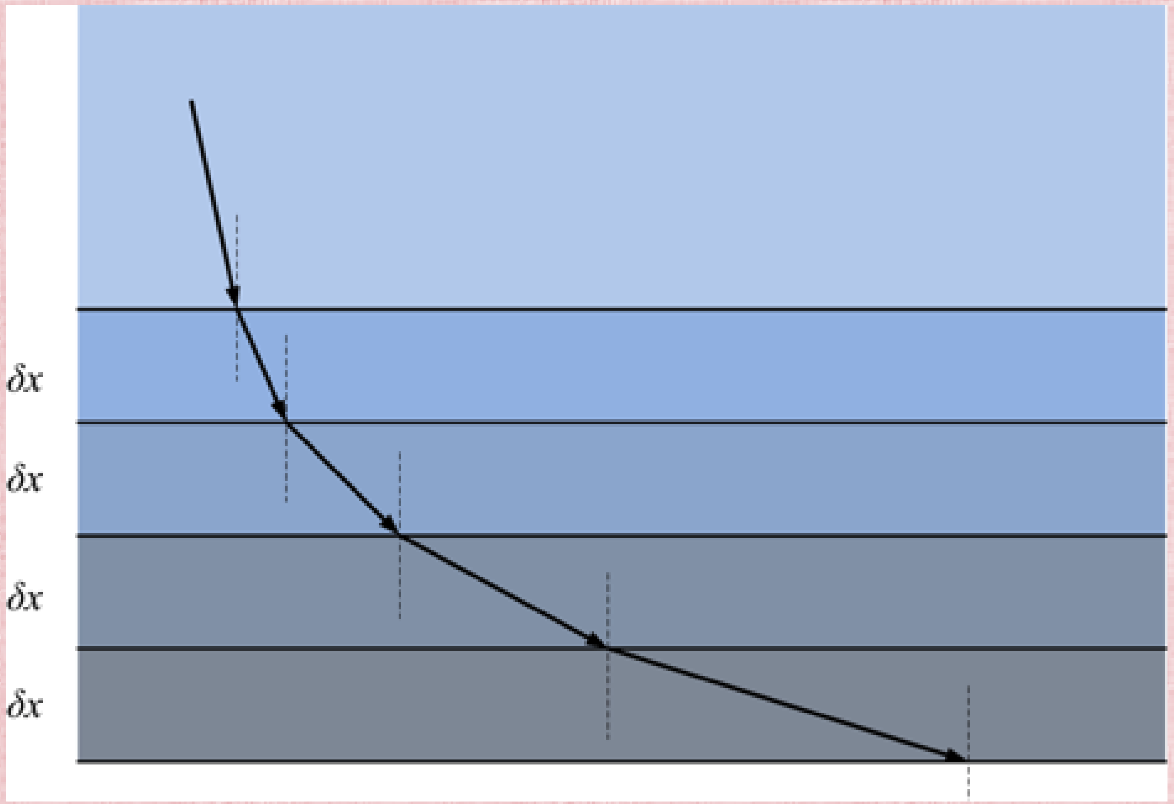


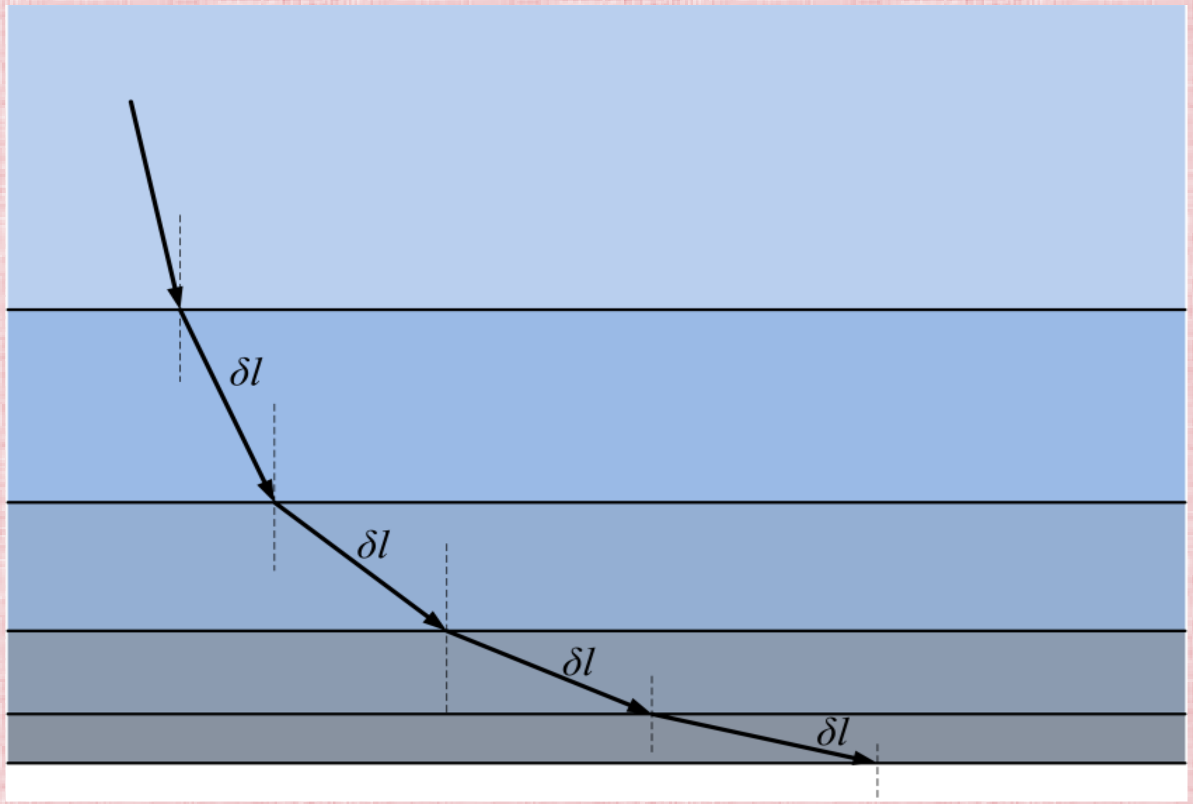
Superior Mirage (March 2021, England)



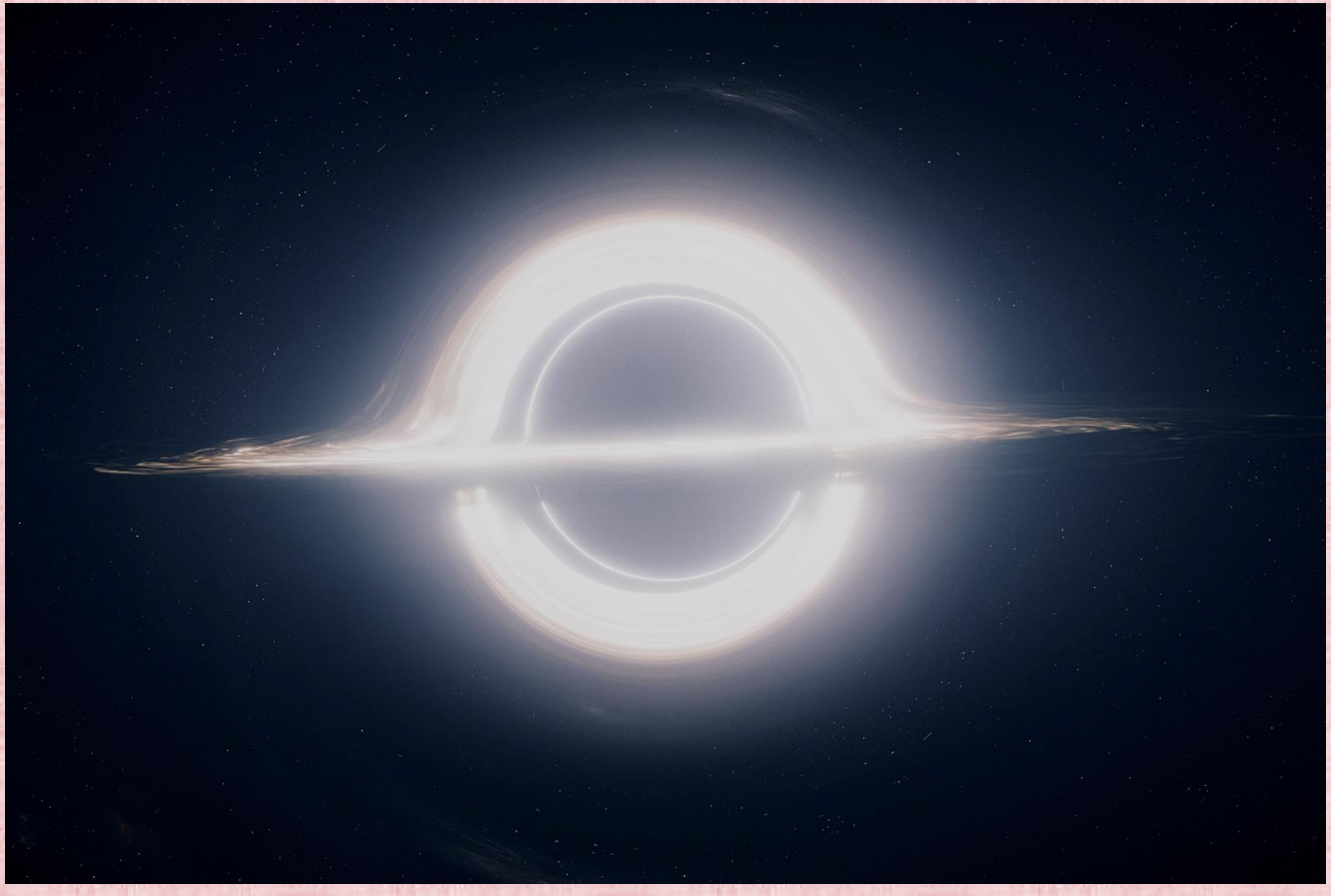
Ray Tracing Atmospheric Refraction

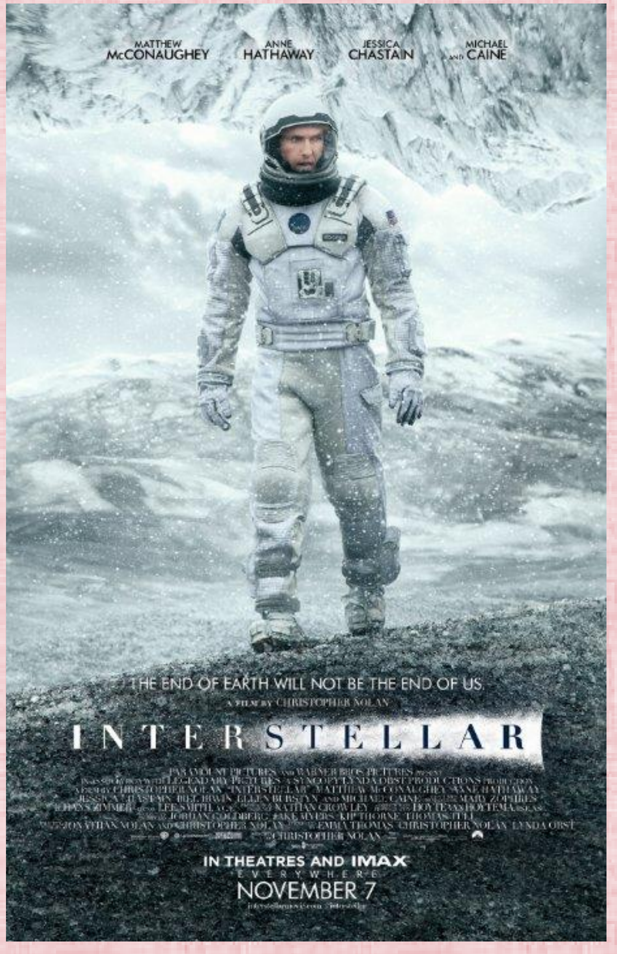
- Bend the rays as they go through air with varying temperature/density
- Change the ray direction between every interval in the vertical direction (left) or along the ray direction (right)





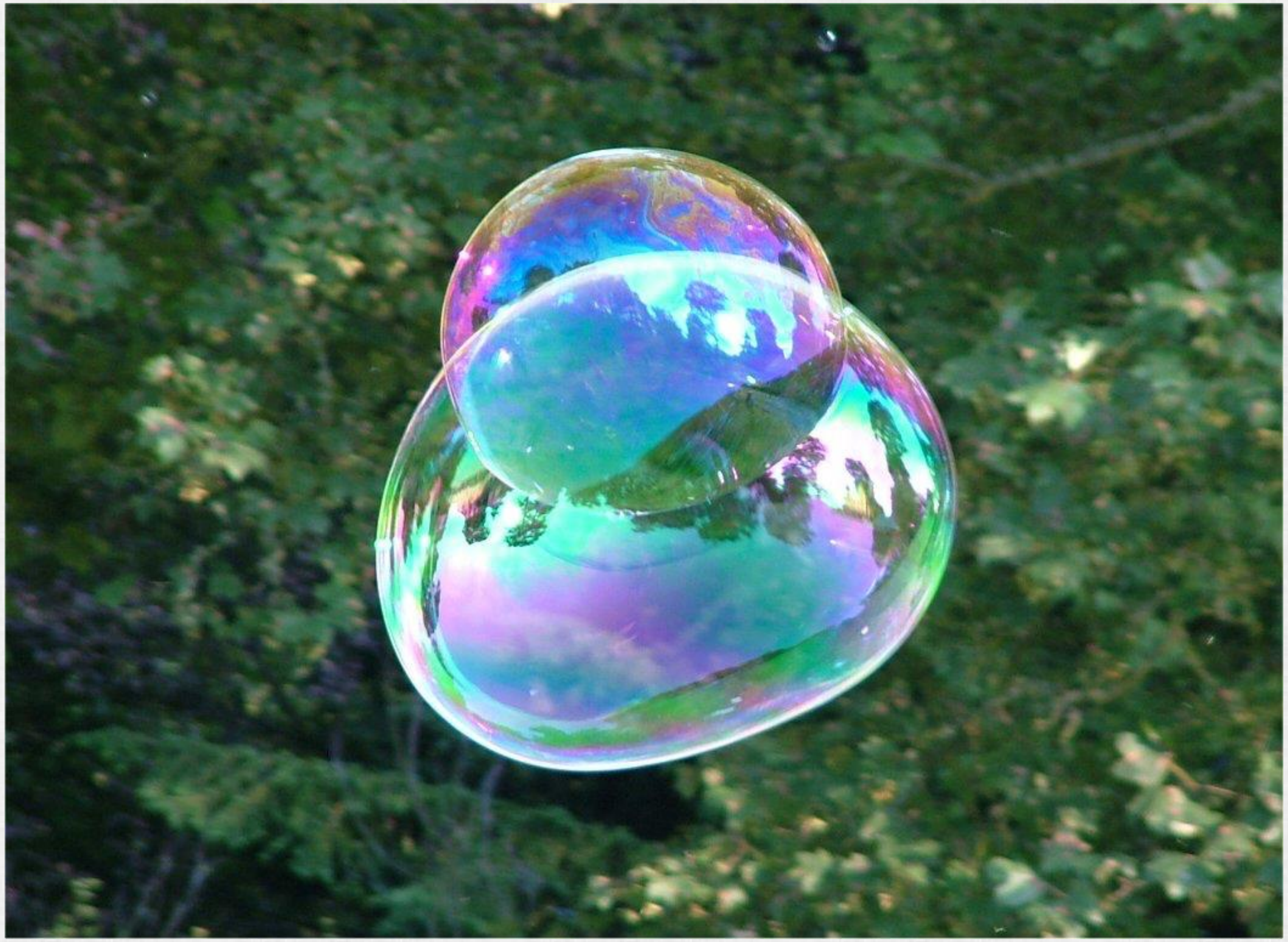
Gravity bends light too!




MATTHEW McCONAUGHEY
ANNE HATHAWAY
JESSICA CHASTAIN
MICHAEL CAINE
THE END OF EARTH WILL NOT BE THE END OF US
A FILM BY CHRISTOPHER NOLAN
INTERSTELLAR
IN THEATRES AND IMAX EVERYWHERE
NOVEMBER 7

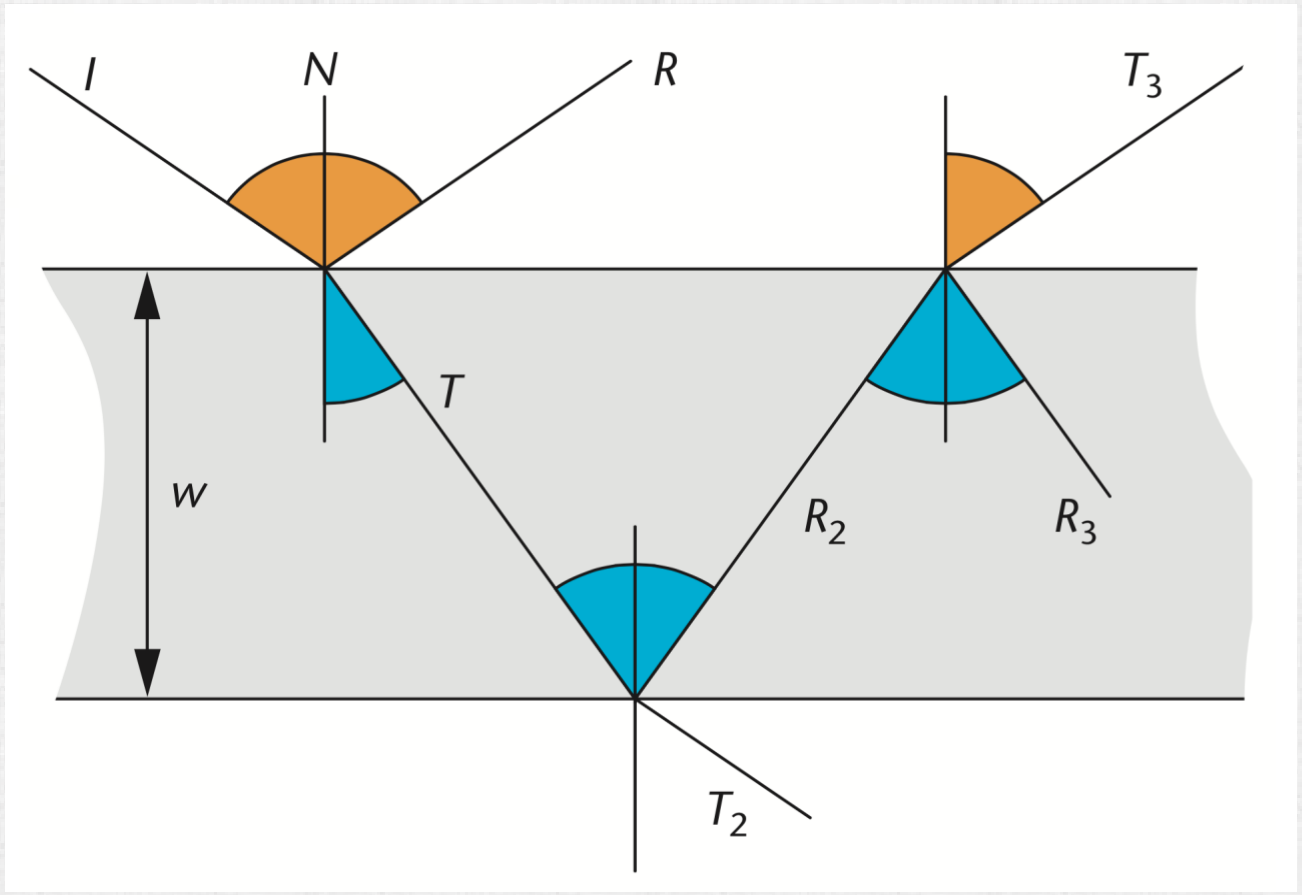
http://www.wired.com/2014/10/astrophysics-interstellar-black-hole/
https://www.businessinsider.com/interstellar-anniversary-learned-about-black-holes-2019-11

Iridescence



Iridescence

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



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