Rendering Algorithms

Spring 2014
Matthias Zwicker

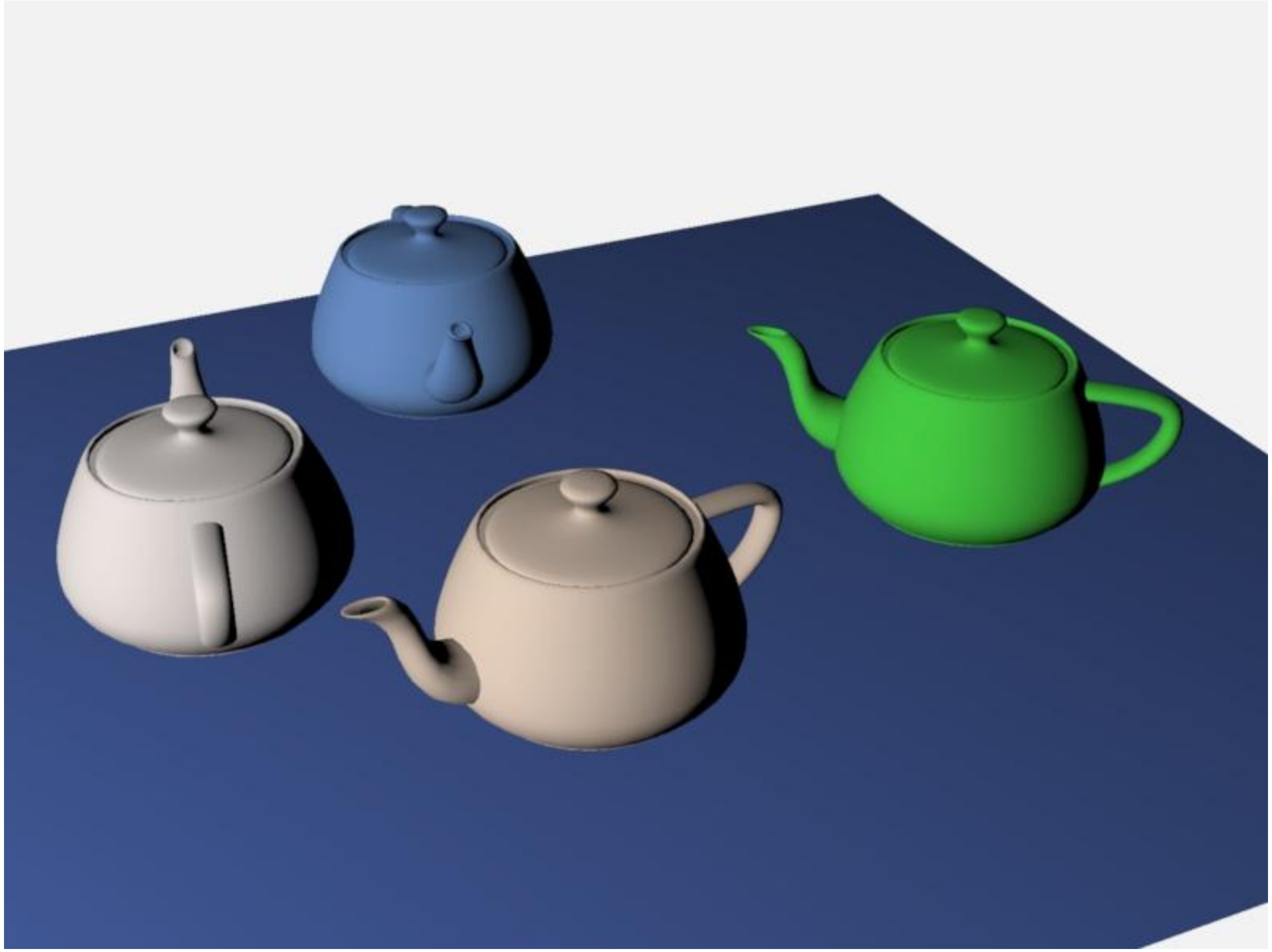
Universität Bern

Today

- Shadows
- Reflection
- Refraction
- Further topics…

Shadows

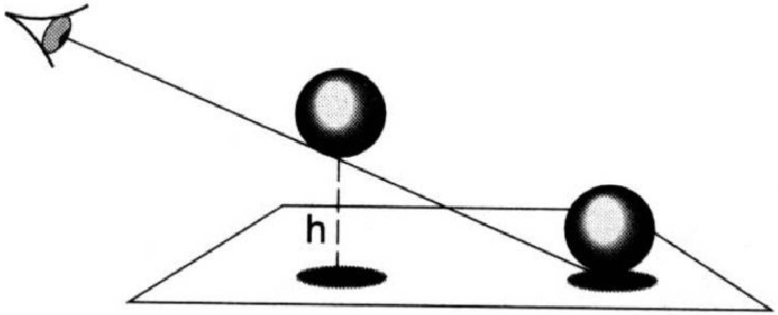
- No shadows, objects appear to float

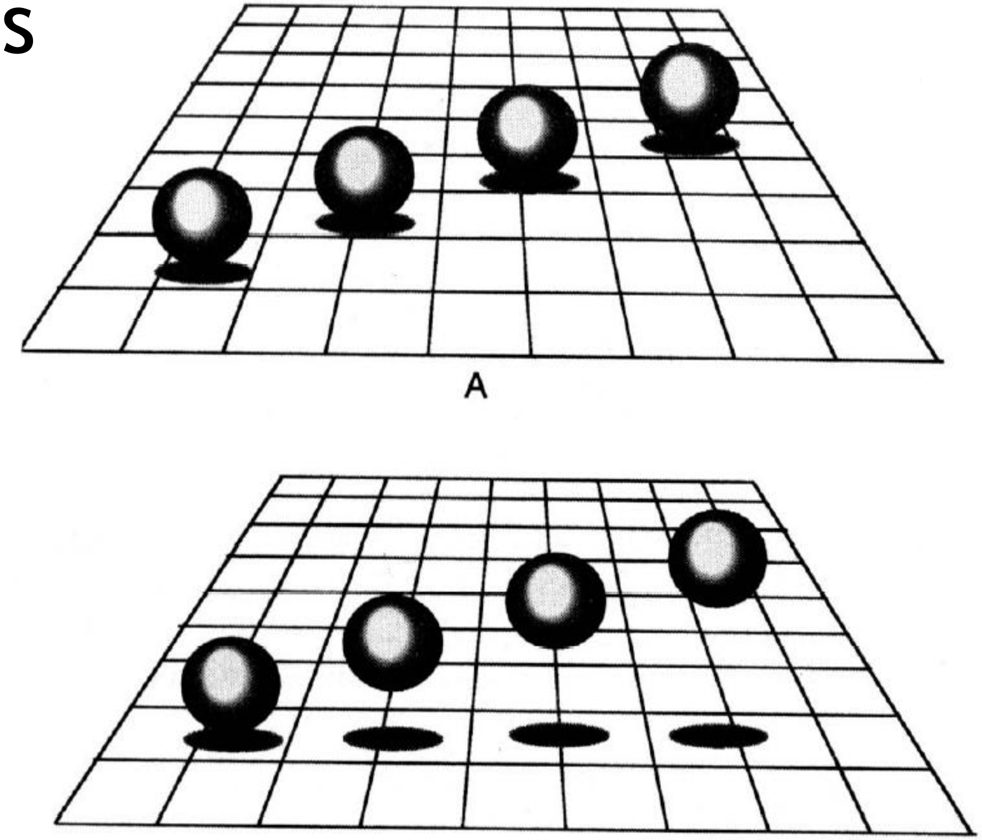


[Jensen]

Shadows

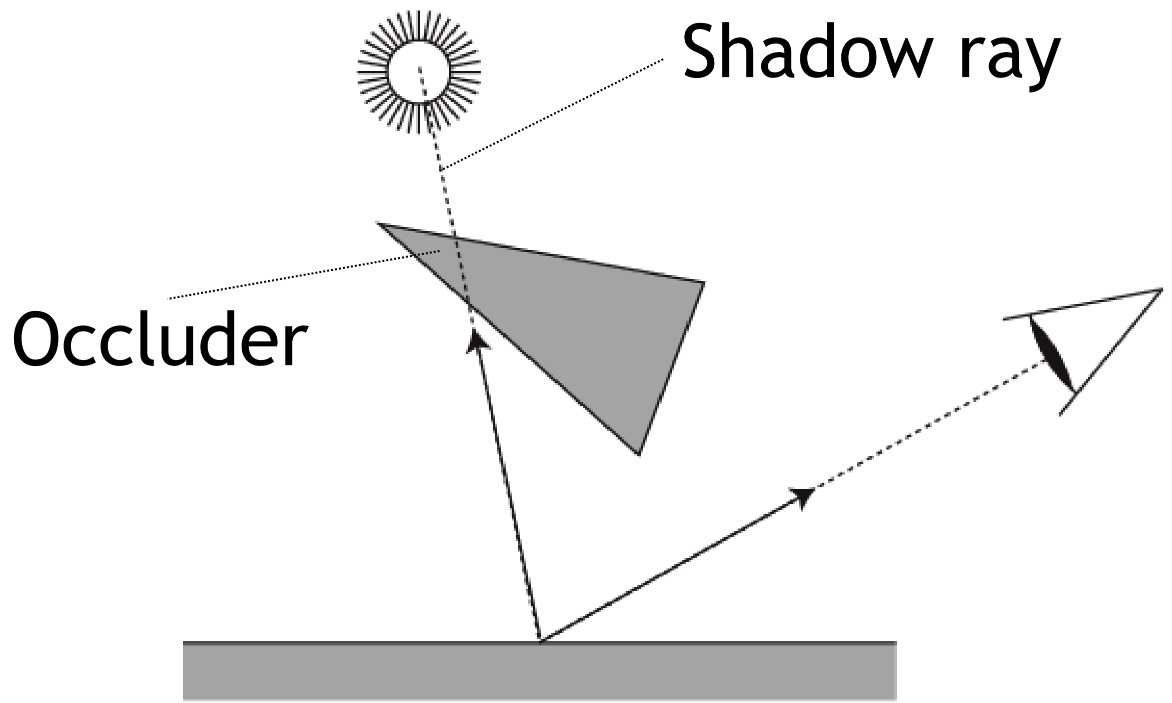
- Indicate contact points
- Provide depth cues





Shadow rays

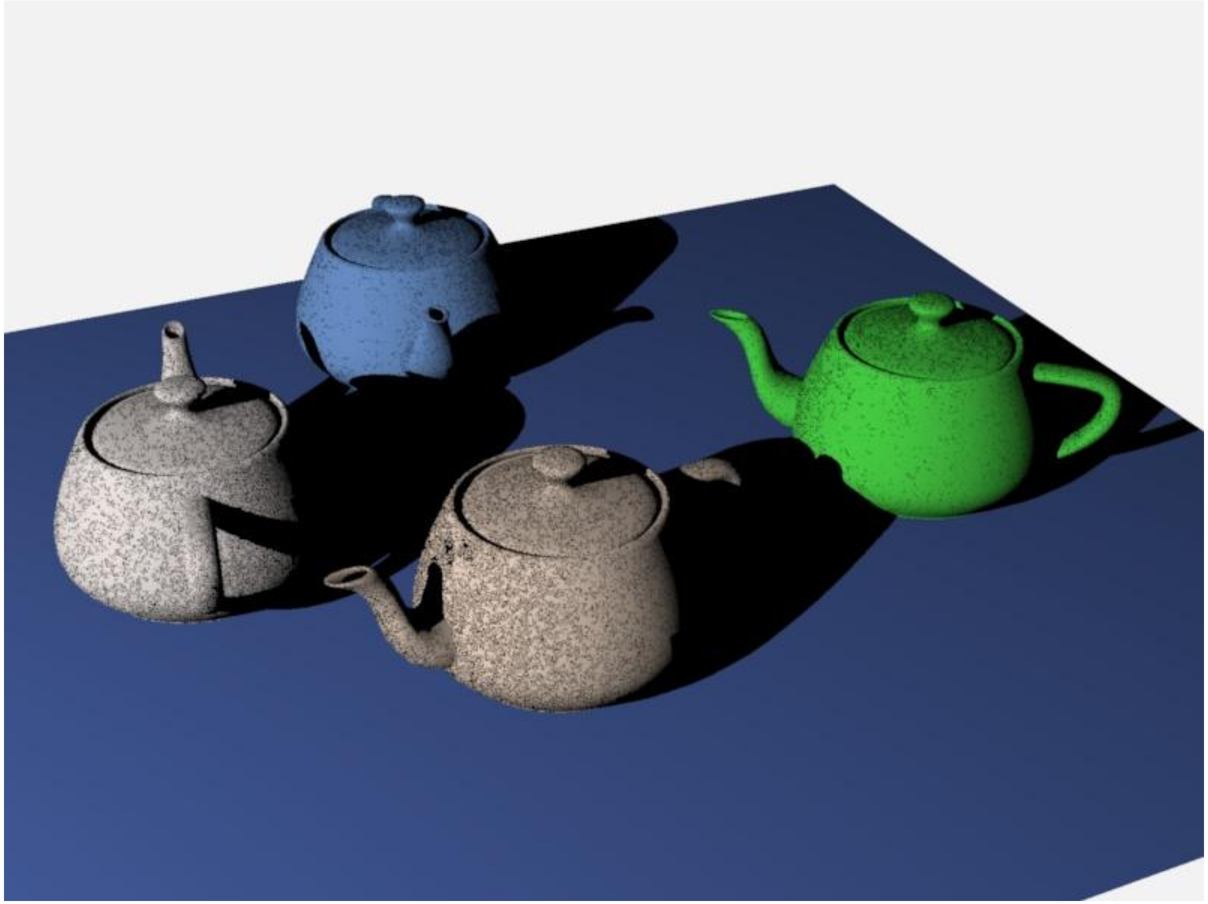
- Shadow ray: ray from hit point towards light
- If shadow ray intersects other geometry (occluders) before it reaches light, light should be ignored



Shadow rays

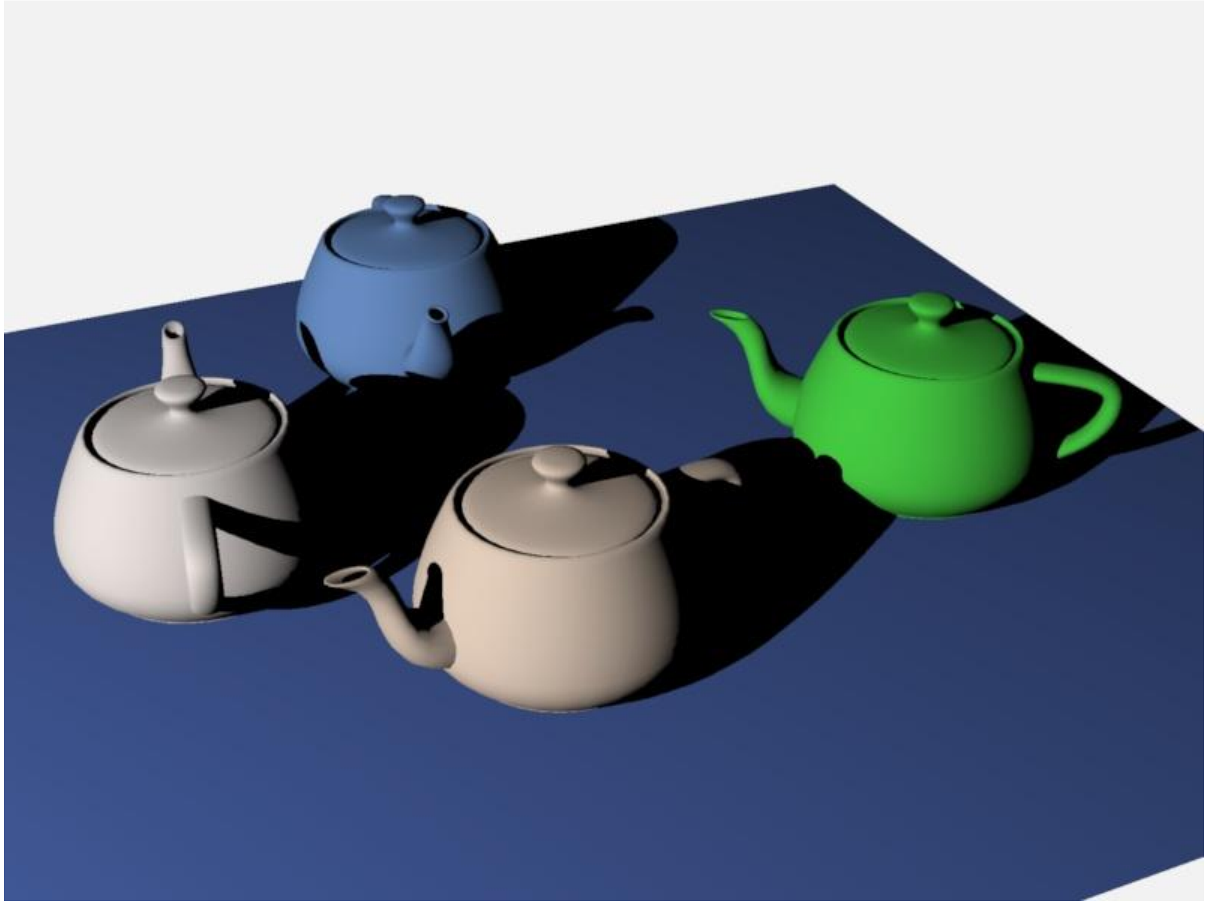
- Diffuse and Phong/Blinn illumination
  - Shoot shadow ray from hit point toward each point light
  - If not occluded, evaluate shading model, accumulate reflected light

“Shadow acne”



[Jensen]

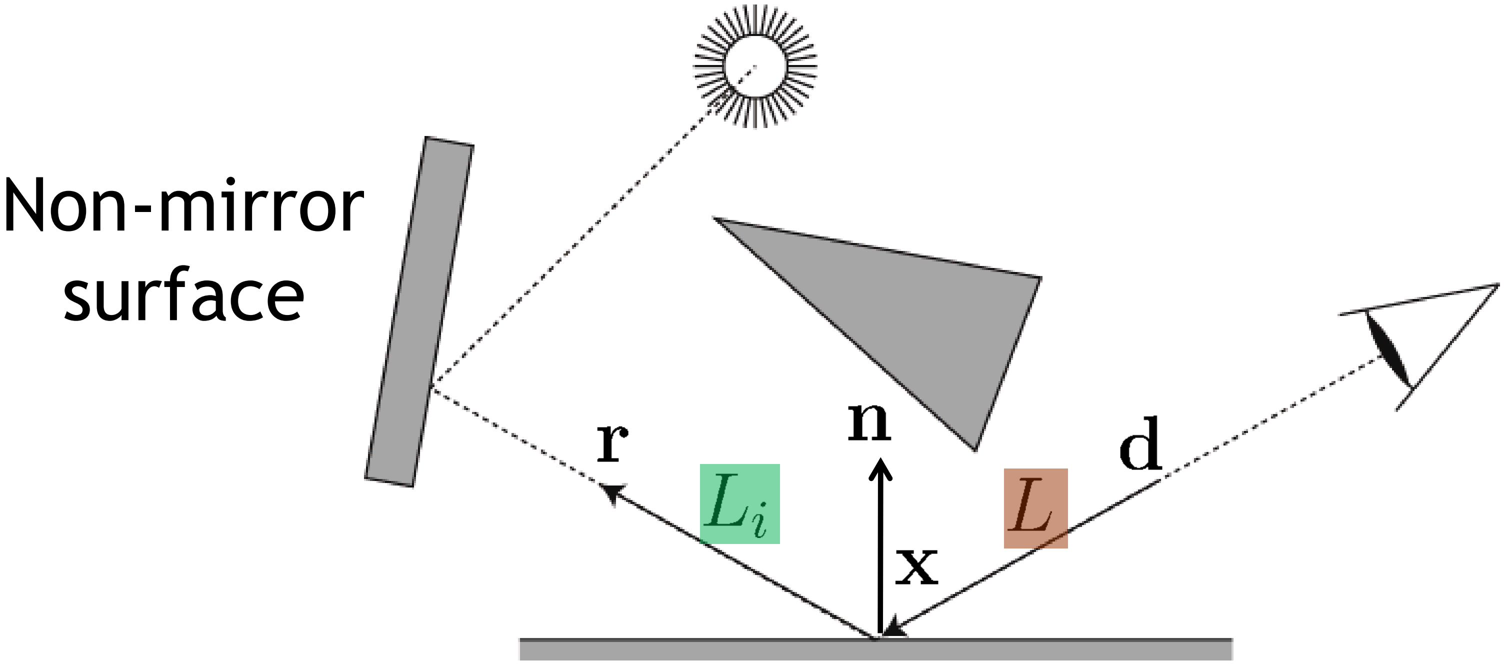
Bias



[Jensen]

Only accept intersections if $t > \epsilon, \quad \epsilon > 0.00001$

Mirror reflection



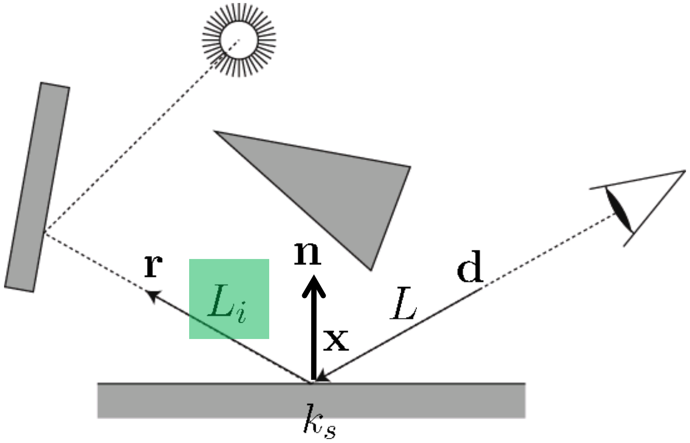
Mirror surface, reflection coefficient k_s

Reflected light $L(\mathbf{x}, \mathbf{d}) = k_s L_i(\mathbf{x}, \mathbf{r})$

Reflection direction $\mathbf{r} = \mathbf{d} - 2(\mathbf{d} \cdot \mathbf{n})\mathbf{n}$

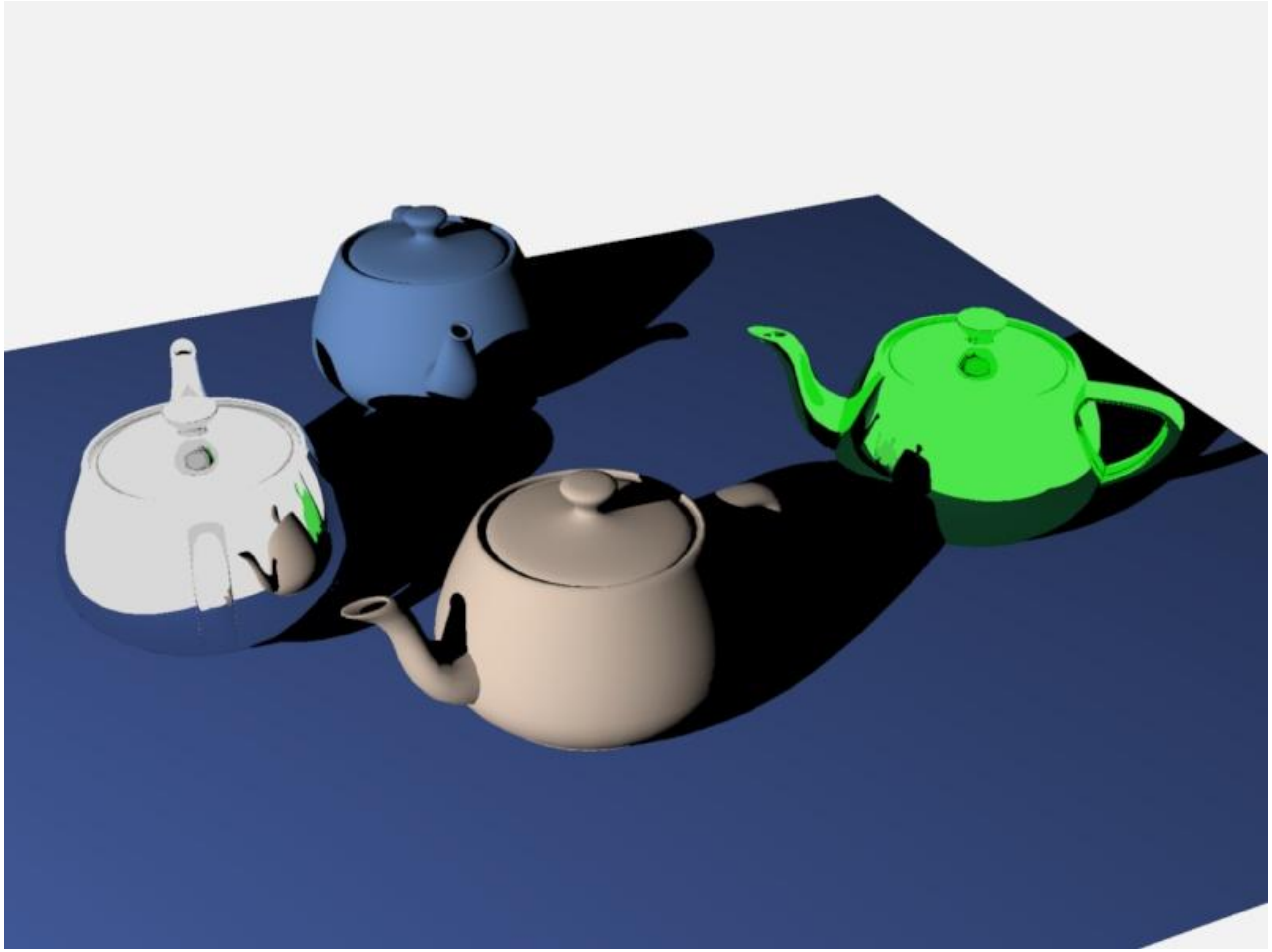
Mirror reflection

- When mirror is hit, trace rays recursively to determine incident light $L_i(\mathbf{x}, \mathbf{r})$
- Limit recursion depth



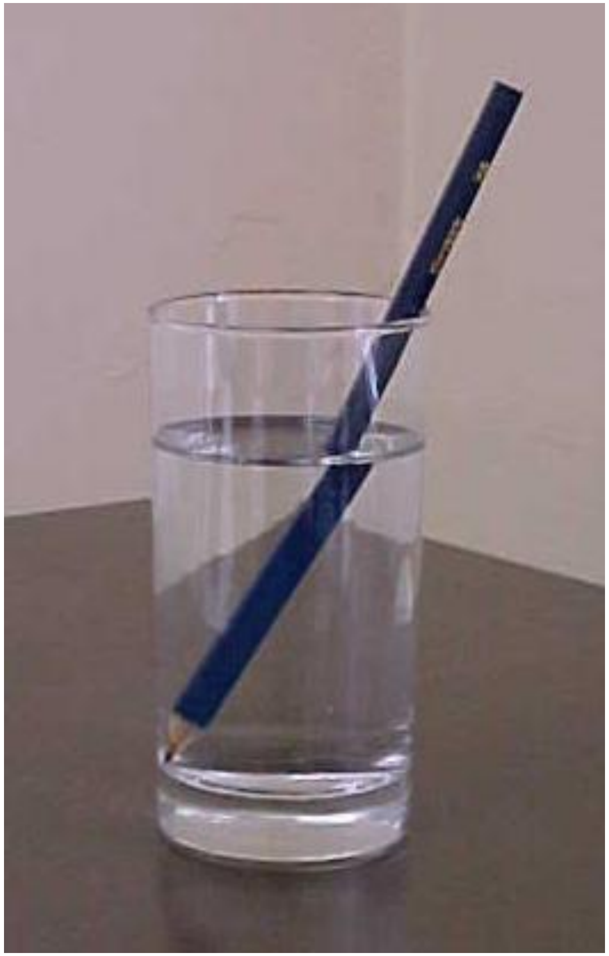
```
spectrum integrate( scene, lights, ray ) {
  hitRecord = intersect( scene, ray )
  material = hitRecord.material
  if( material.isMirror() ) {
    ray = makeReflectedRay( hitRecord )
    return material.ks*integrate( scene, lights, ray )
  } else … }
```

Mirror reflection



Specular refraction

- Light travels at different speeds in different media
- Light is bent when it goes from one medium to another
- Dielectric materials (insulators)
  - Diamond, glass, water, air
  - Etc.

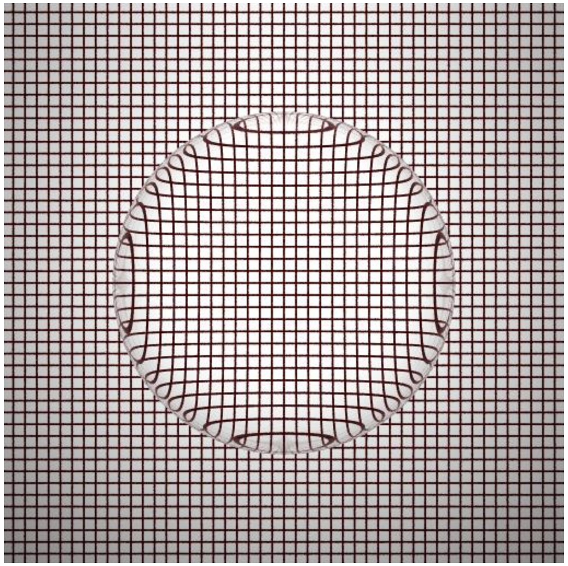


Index of refraction

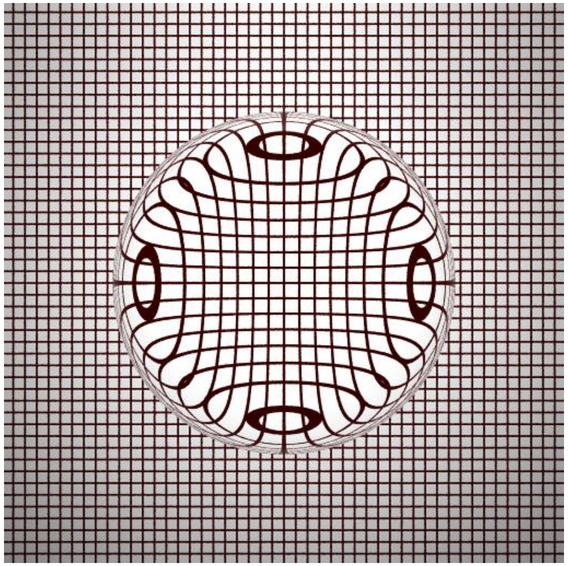
- Speed of light depends on medium
  - Speed of light in vacuum c
  - Speed of light in medium v
- Index of refraction $n = c/v$
  - Air 1.00029
  - Water 1.33
  - Acrylic glass 1.49

Index of refraction

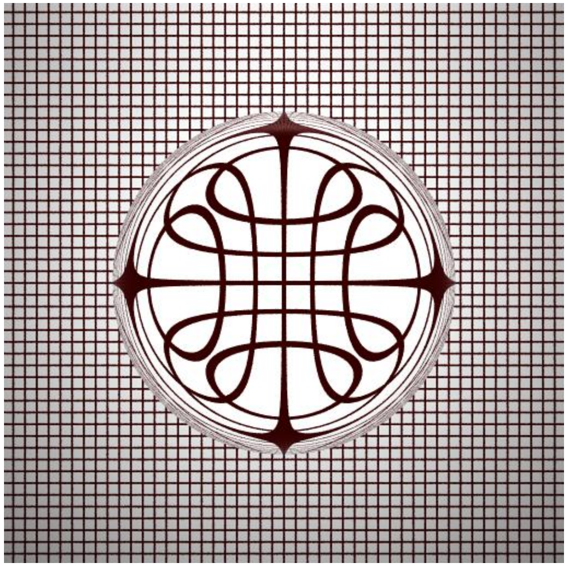
- Examples of refractive spheres with different index of refraction



n=1.01



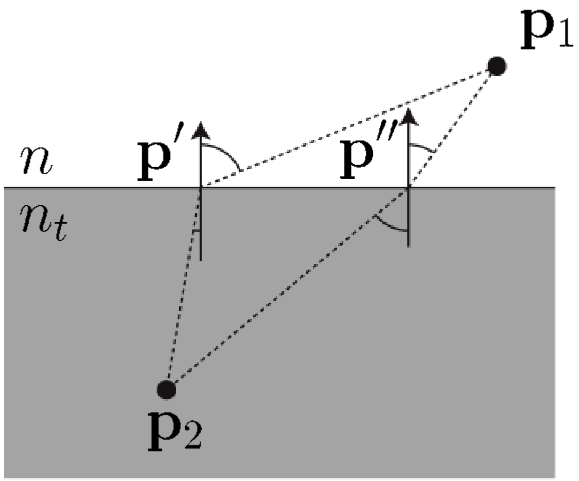
n=1.1



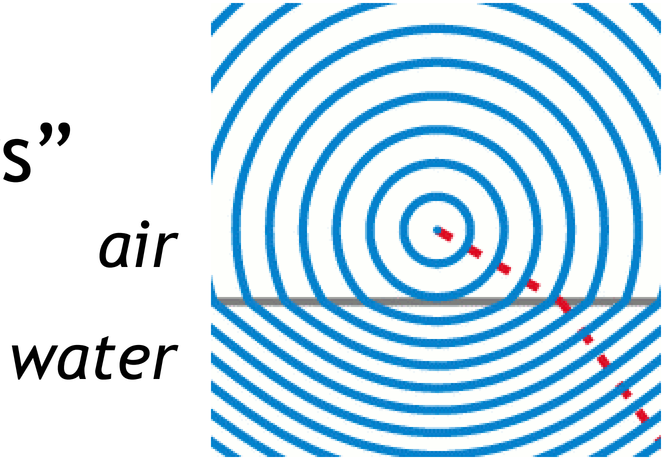
n=1.3

Explanations

- Fermat’s principle
  “The actual path between two points taken by a beam of light is the one which is traversed in the least time“
  http://en.wikipedia.org/wiki/Fermat's_principle



- “Change in phase velocity leads to bending of light rays”
  http://en.wikipedia.org/wiki/Refraction



Snell’s law (http://en.wikipedia.org/wiki/Snell's_law)

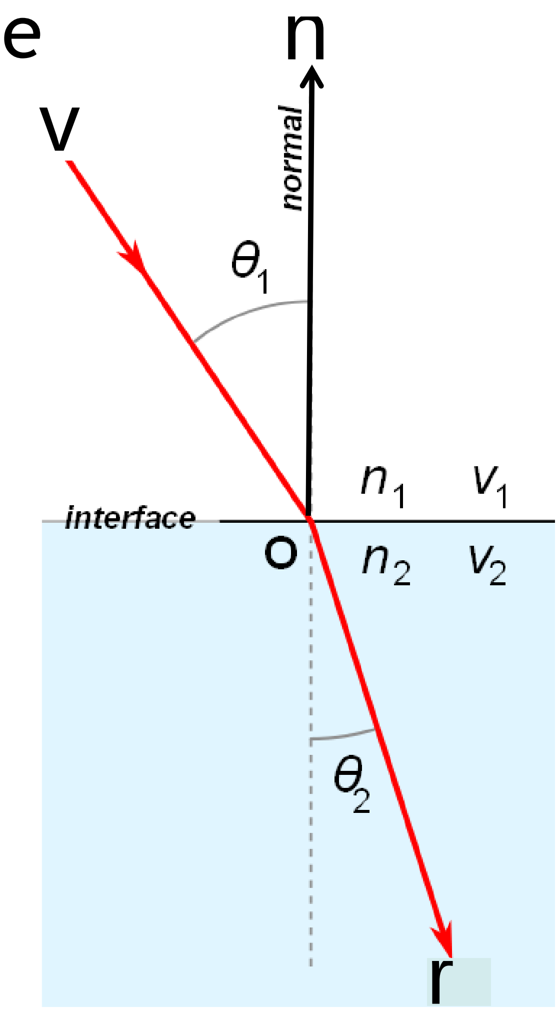
- Ratio of sines of angles of incidence and refraction is equal to opposite ratio of indices of refraction

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

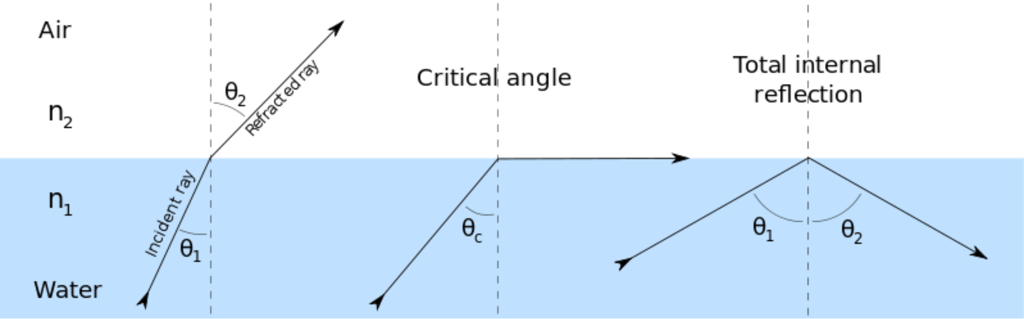
- Vector form to obtain refracted direction $\mathbf{r}$

$$\mathbf{r} = \frac{n_1}{n_2}\mathbf{v} + \left(\frac{n_1}{n_2}\cos\theta_1 - \cos\theta_2\right)\mathbf{n}$$

  - Viewing, refracted direction $\mathbf{v}$, $\mathbf{r}$
  - Normal vector $\mathbf{n}$



Total internal reflection



- Angle of refracted ray

$$\theta_2 = \arcsin\left(\sin(\theta_1)\frac{n_1}{n_2}\right)$$

  - Not defined for argument >1!

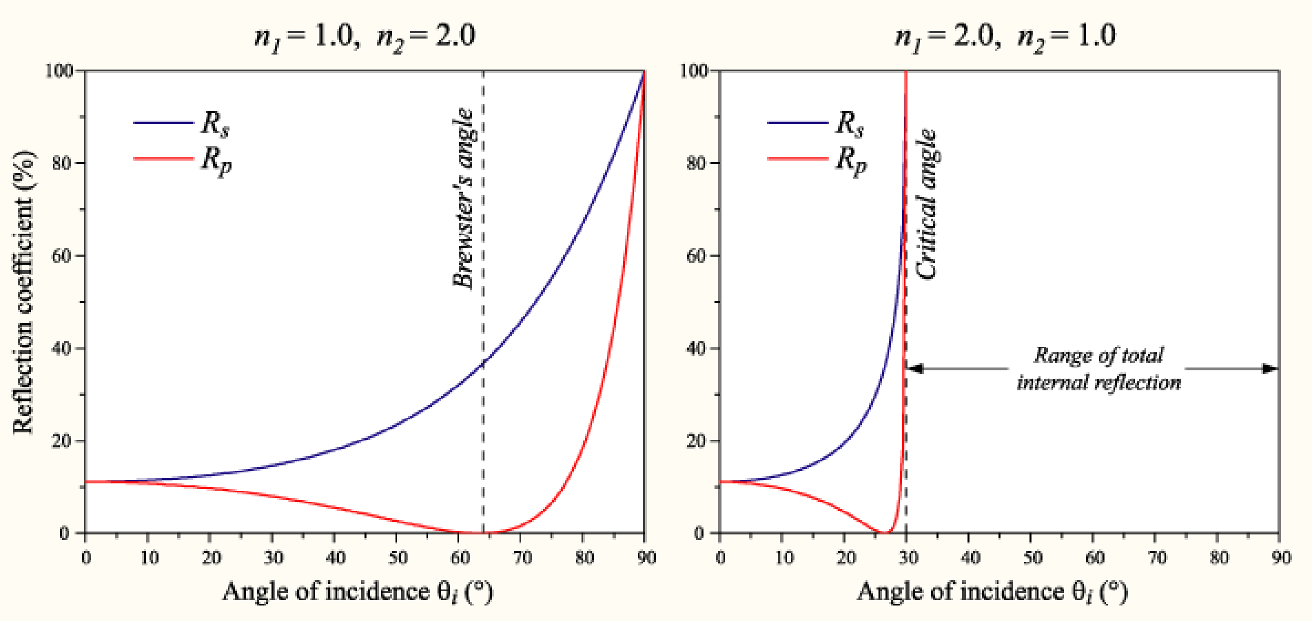
- Critical angle

$$\theta_c = \arcsin\left(\frac{n_2}{n_1}\right)$$

  - If $\theta_1 = \theta_c$ we get $\theta_2 = \pi/2$, refracted ray is parallel to interface
  - If $\theta_1 > \theta_c$ we have total internal reflection (no refracted ray leaving the medium)

Fresnel equations (http://en.wikipedia.org/wiki/Fresnel_equations)

- When light travels from one medium to an other, both reflection and refraction may occur
- Fresnel equations describe fraction of intensity of light that is reflected and refracted
  - Depends on polarization of light



Schlick's approximation

- Fresnel equations are relatively complex to evaluate
- In graphics, often use Schlick's approximation
  - Ratio *F* between reflected and refracted light

$$F = f + (1 - f)(1 - \mathbf{v} \cdot \mathbf{n})^5 \qquad f = \frac{\left(1.0 - \frac{n_1}{n_2}\right)^2}{\left(1.0 + \frac{n_1}{n_2}\right)^2}$$

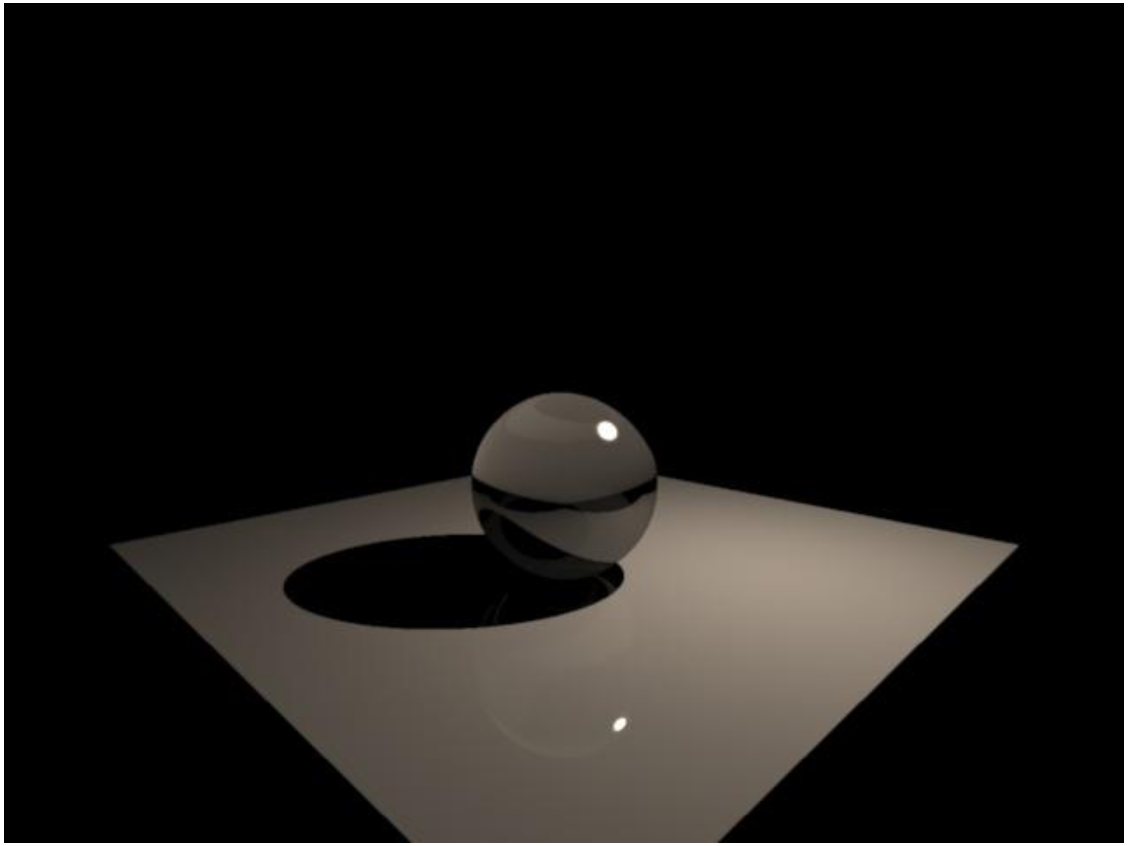
  - Result: *F* * *reflected* + (1-*F*) * *refracted*

Detailed description

- Short but detailed description how to implement reflection and refraction in a ray tracer
  http://graphics.stanford.edu/courses/cs148-10-summer/docs/2006--degreve--reflection_refraction.pdf

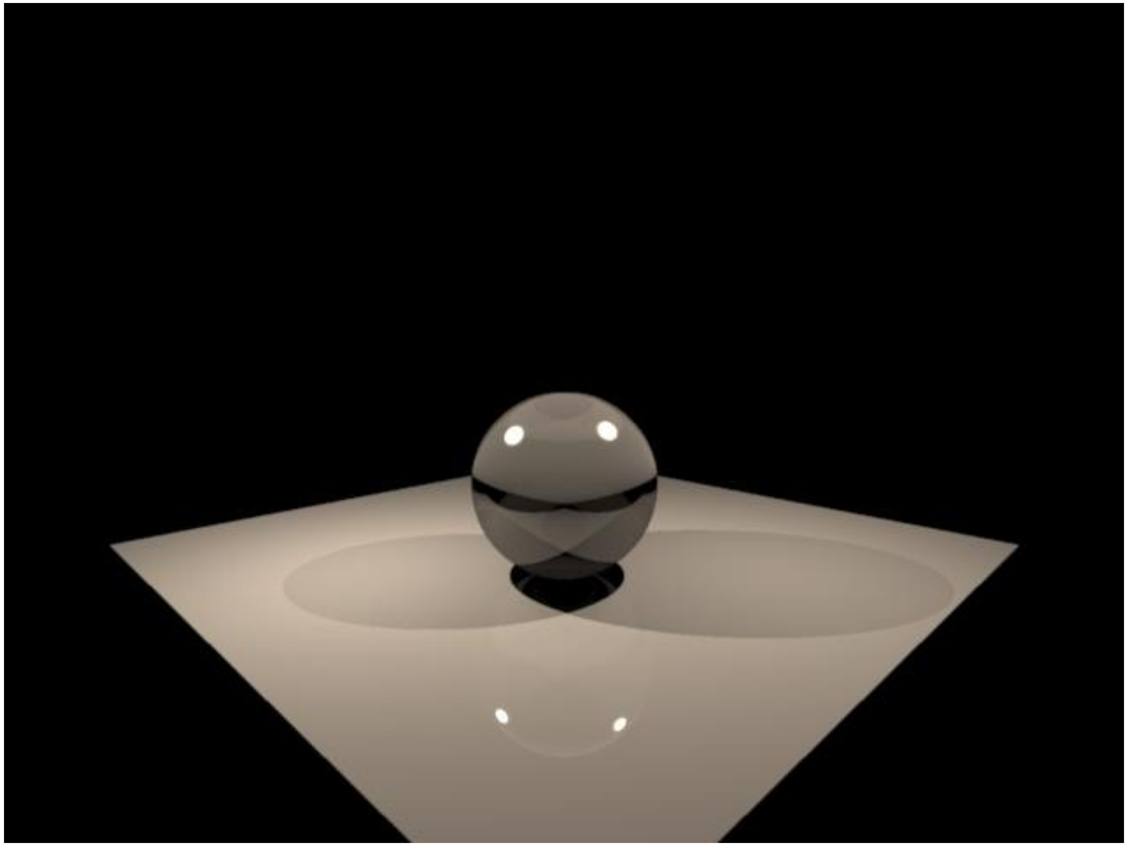
A glass sphere

- One light source



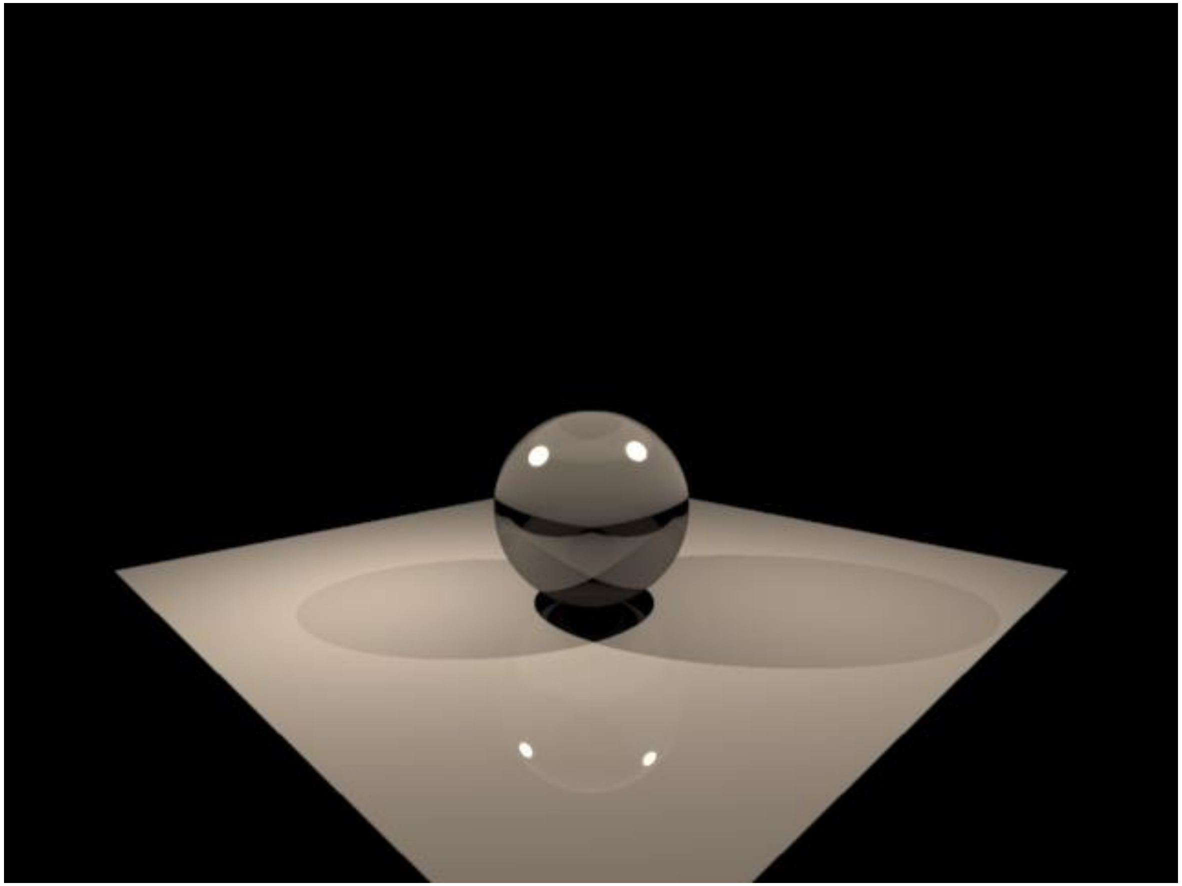
A glass sphere

- Two light sources



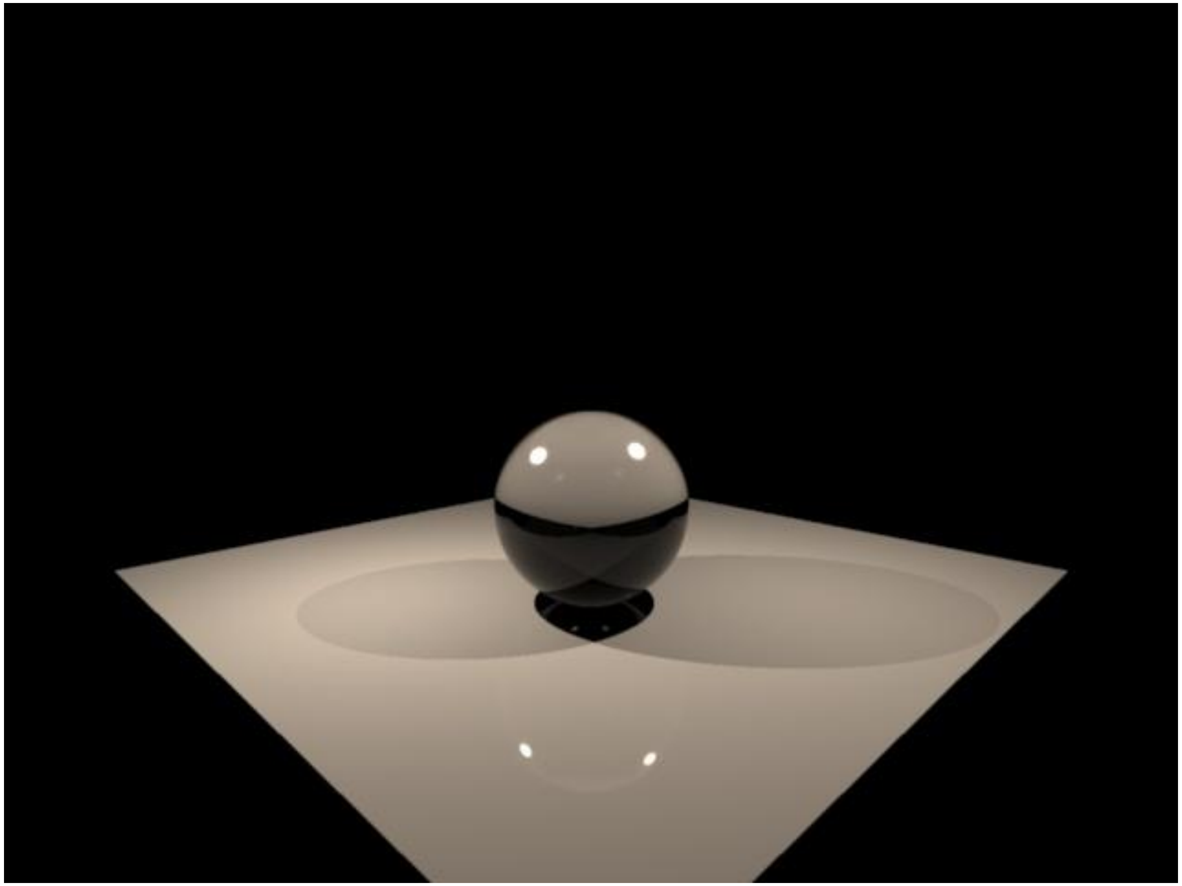
Glass sphere

- No Fresnel reflection



Glass sphere

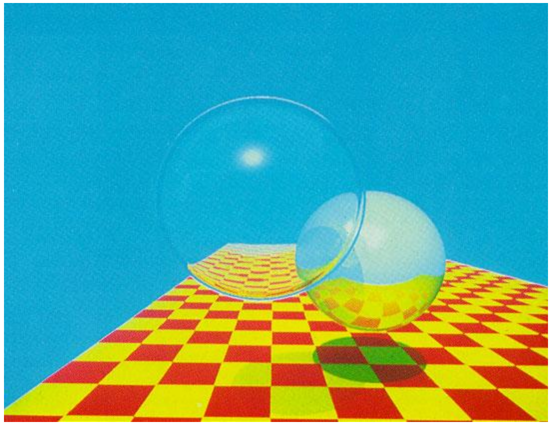
- With Fresnel reflection



Implementation

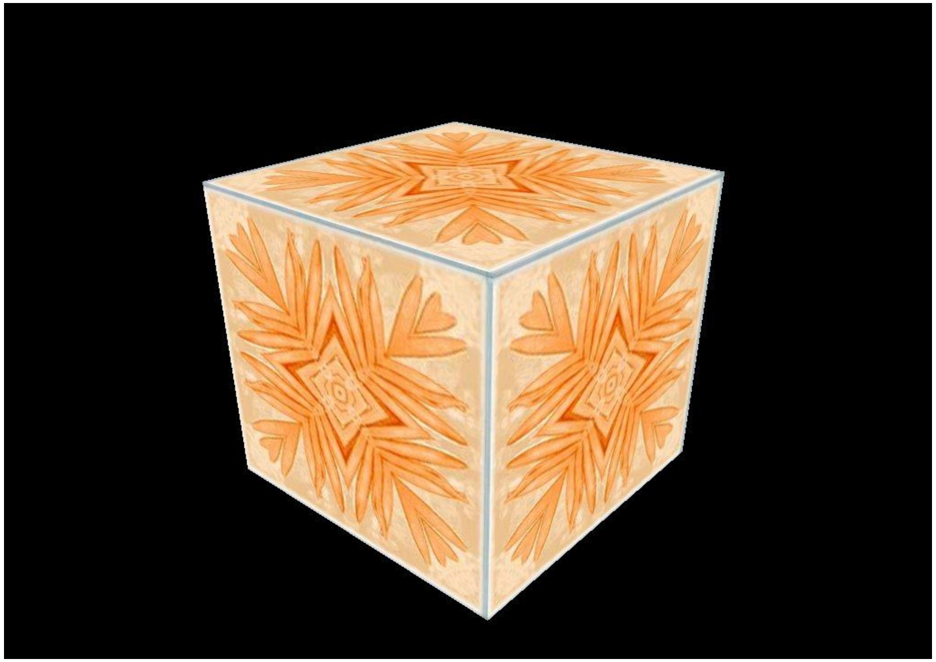
- If refractive material is hit, recursively trace both reflected and refracted rays
- Original idea for recursive ray tracing using reflection and refraction goes back to Turner Whitted, 1980

http://en.wikipedia.org/wiki/Ray_tracing_(graphics)#Ray_tracing_algorithm
http://portal.acm.org/citation.cfm?id=358882



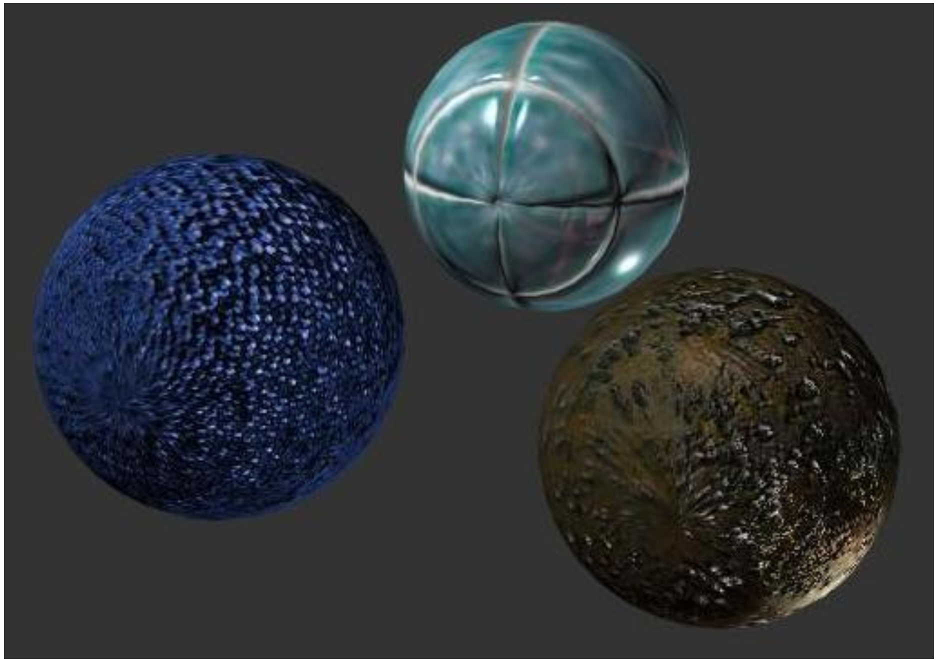
Further Topics...

- Hackers Bonus:
  - **Textures**
  - Bump mapping
  - Procedural textures



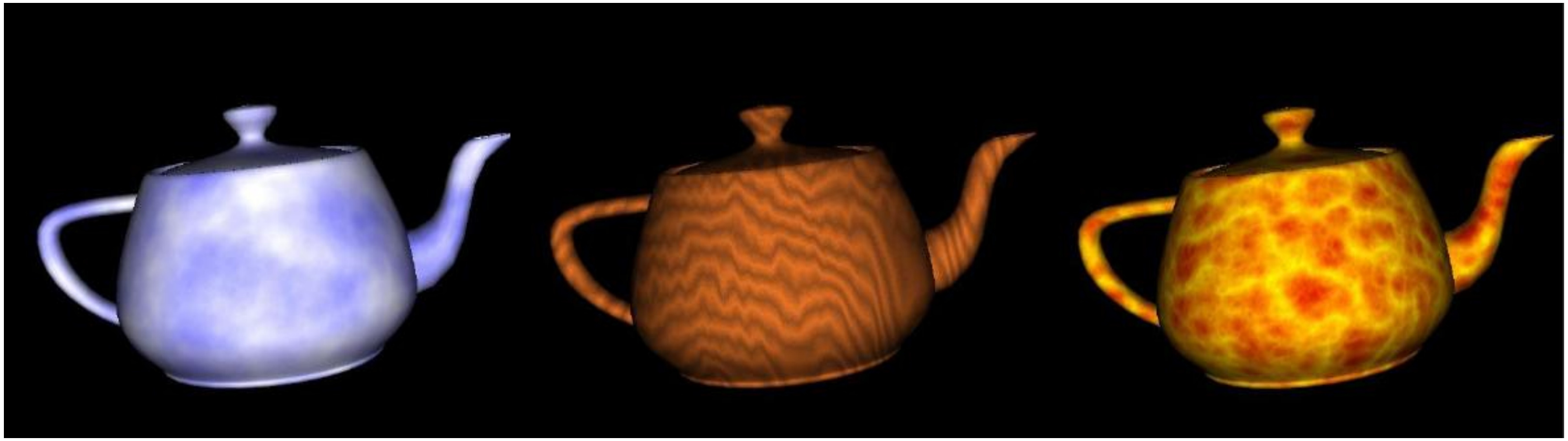
Further Topics...

- Hackers Bonus:
  - Textures
  - **Bump mapping**
  - Procedural textures



Further Topics...

- Hackers Bonus:
  - Textures
  - Bump mapping
  - **Procedural textures**
    - Most famous kind: Perlin noise textures (reference implementation in Java available: http://mrl.nyu.edu/~perlin/noise/)

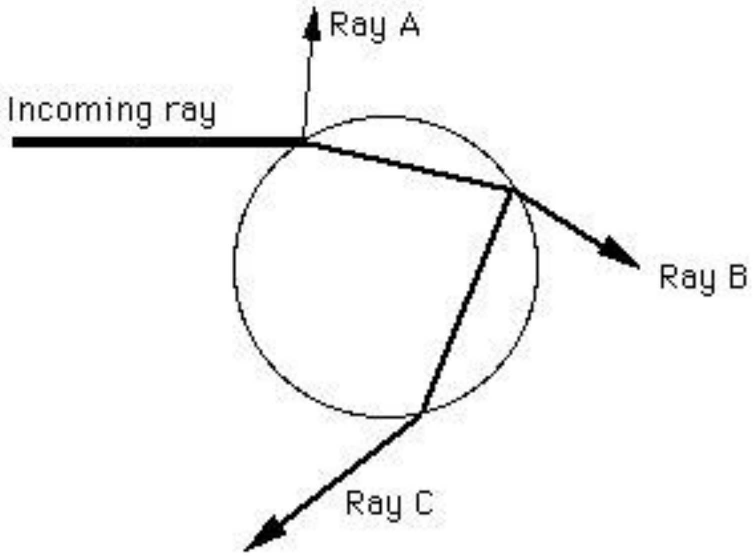


Dispersion (http://en.wikipedia.org/wiki/Dispersion_(optics))

- Index of refraction varies with wavelength

- Need more than three spectral samples (r,g,b) to simulate/render!

Dispersion: rainbows!



Water drop