Ray Tracing

Course web page:

http://goo.gl/EB3aA

Outline

- Recursive rays
  - Reflection
  - Refraction
- Distributed/distribution ray tracing for anti-aliasing
- HW #3 outline

Ray "tracing" for more realism

- Ray casting does not account for two important visual phenomena:
  - Mirror-like surfaces should **reflect** other objects in scene
  - Transparent surfaces should **refract** scene objects behind them

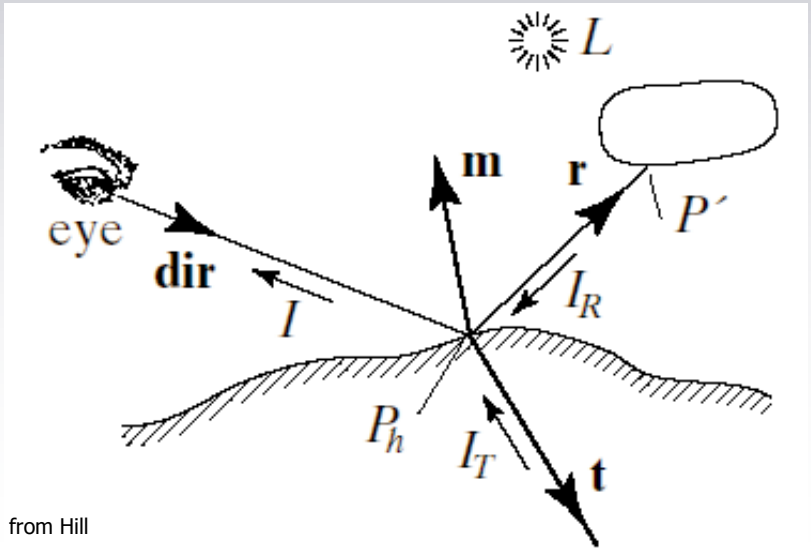
courtesy of J. Arvo

Glossy reflection

Refraction

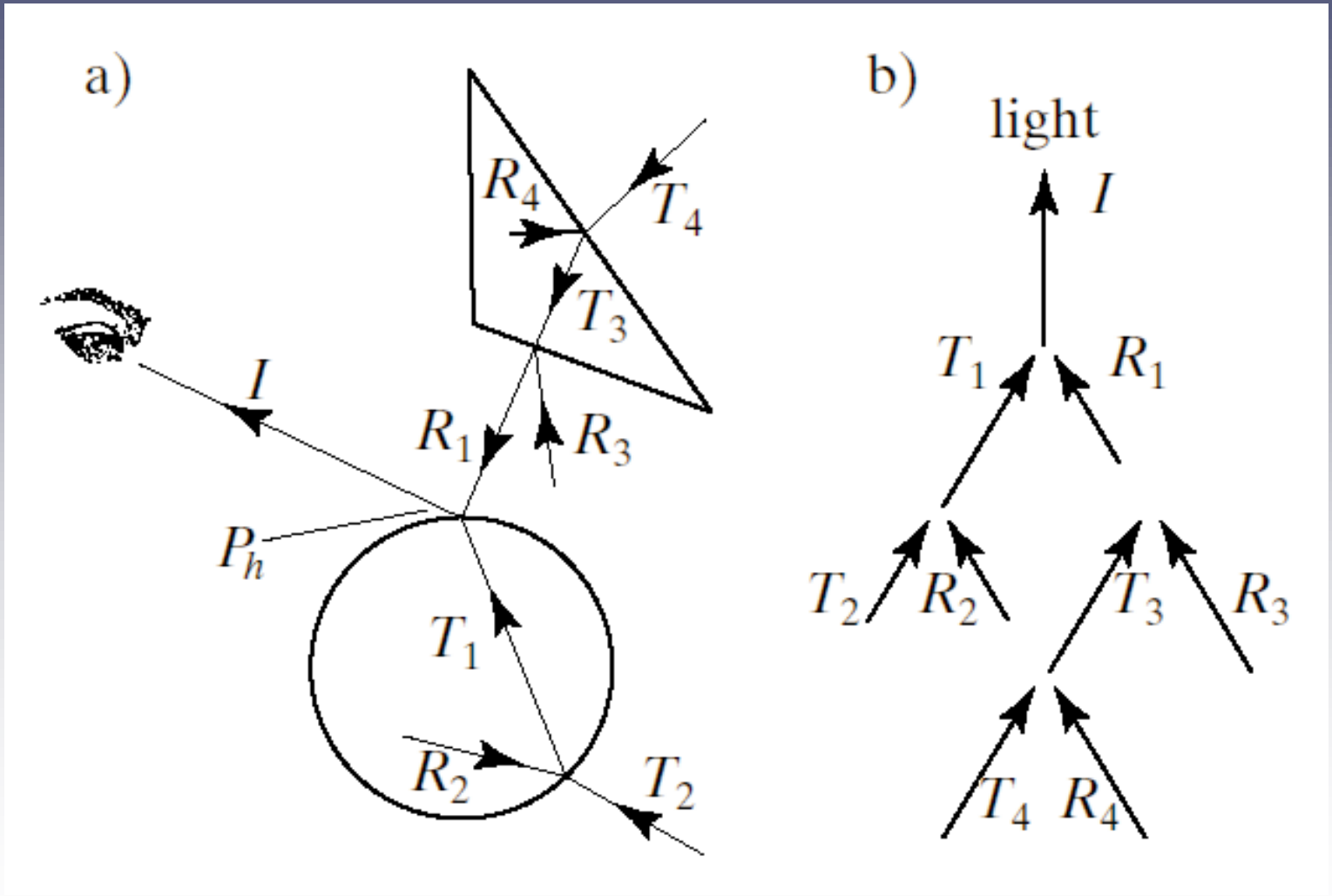
Ray Tracing

- Model: Perceived color at point **p** is an additive combination of local illumination (e.g., Phong), reflection, and refraction effects
  - Weights on last two terms are additional material properties
- Compute reflection, refraction contributions by **tracing** respective rays back from **p** to surfaces they came from and evaluating local illumination at those locations
- Apply operation **recursively** to some maximum depth to get:
  - Reflections of reflections of ...
  - Refractions of refractions of ...
  - And of course mixtures of the two



from Hill

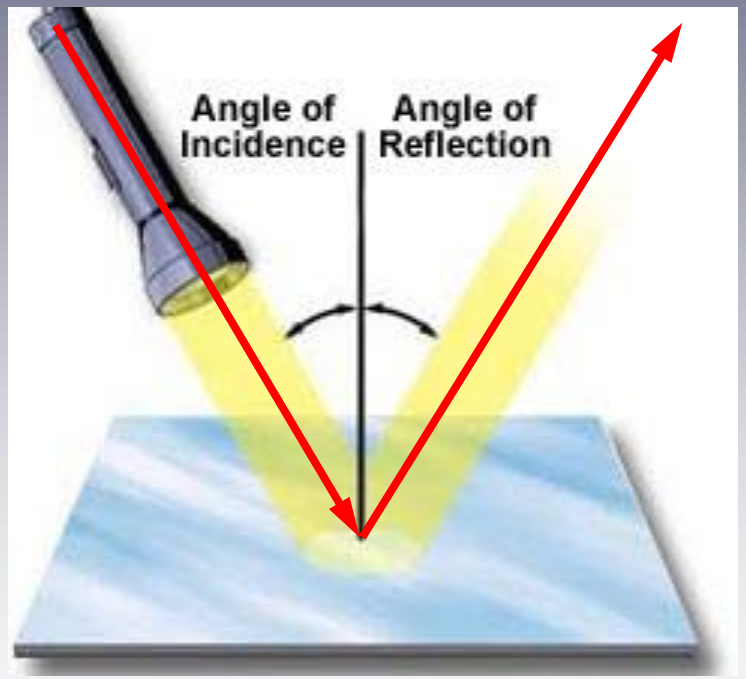
Ray Tracing: Recursion



from Hill

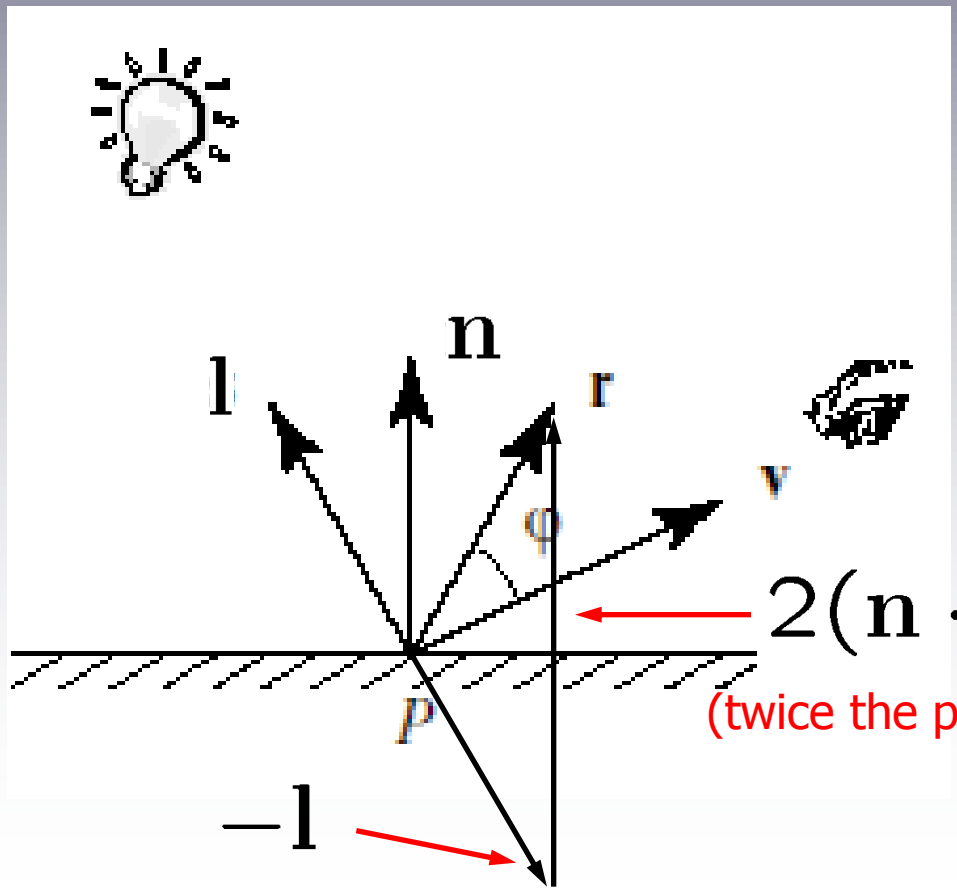
Reflections

incident ray **v** reflected ray **r**



Review: Reflectance direction for Phong model

- We calculated **r** from normal **n**, light direction **l** via: $\mathbf{r} = 2(\mathbf{n} \cdot \mathbf{l})\mathbf{n} - \mathbf{l}$

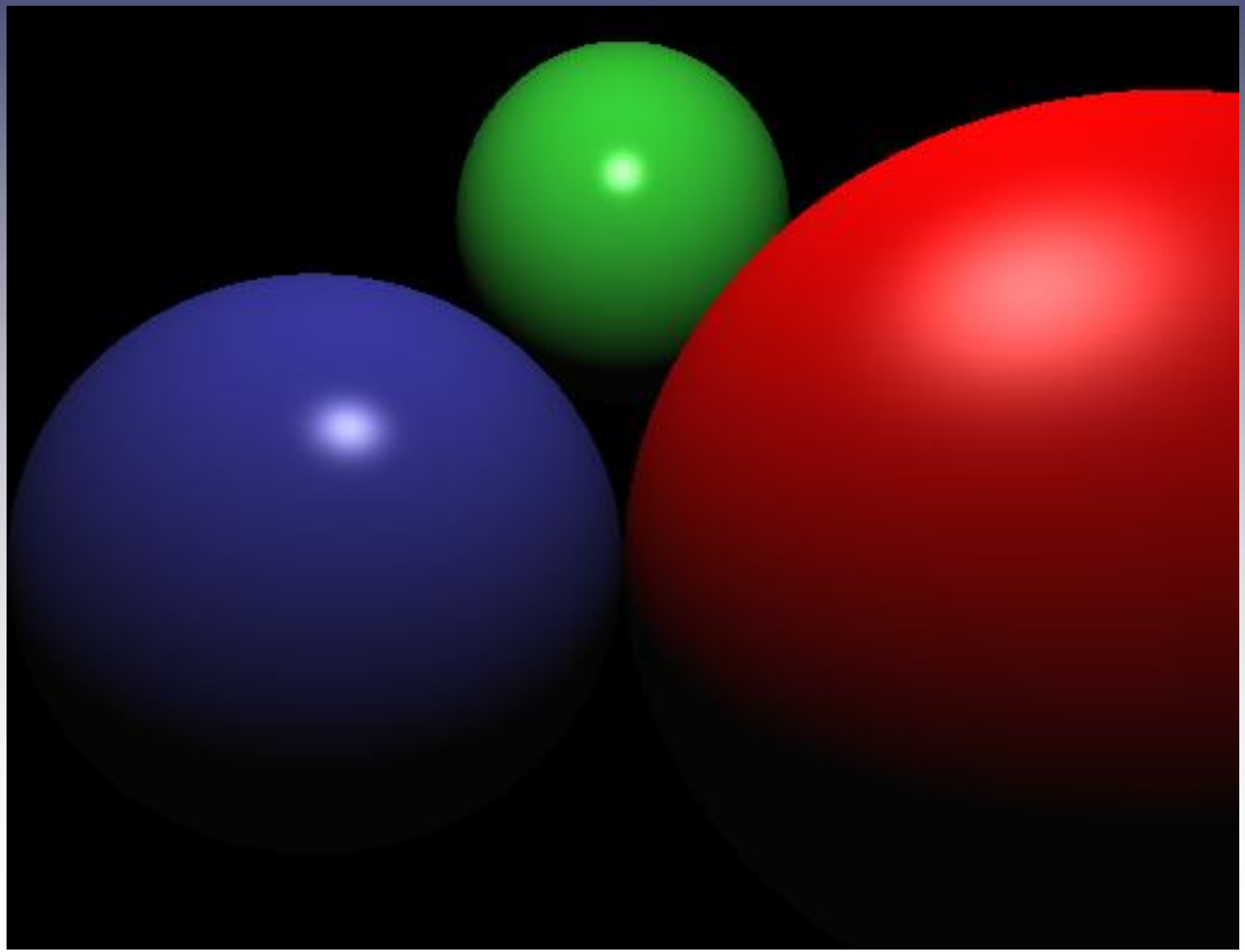


Ray Tracing Reflection Formula

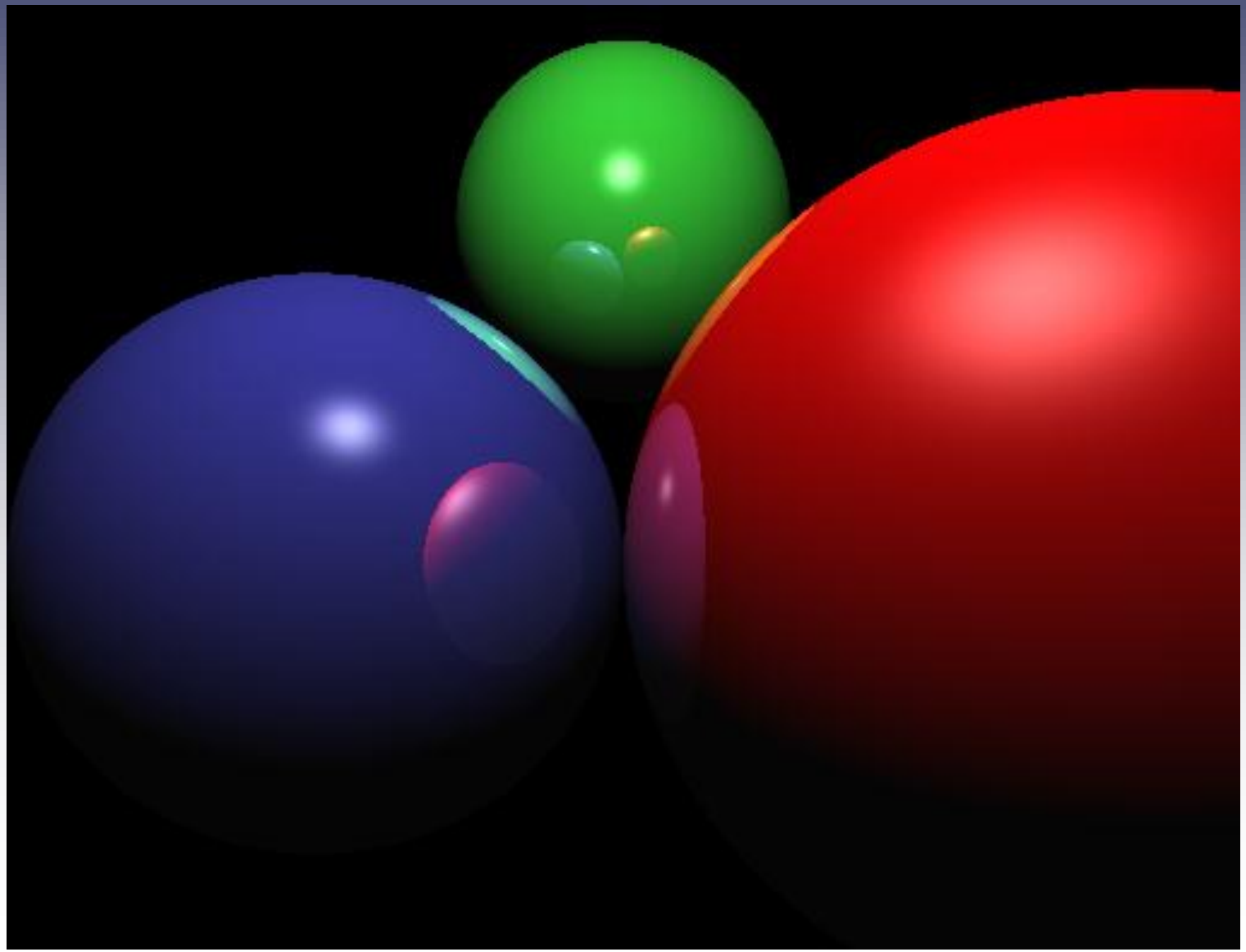
- The formula used for Phong illumination is not what we want here because our incident ray **v** is pointing **in** toward the surface, whereas the light direction **l** was pointed **away** from the surface
- So just negate the formula to get:

$$\mathbf{r} = \mathbf{v} - 2(\mathbf{n} \cdot \mathbf{v})\mathbf{n}$$

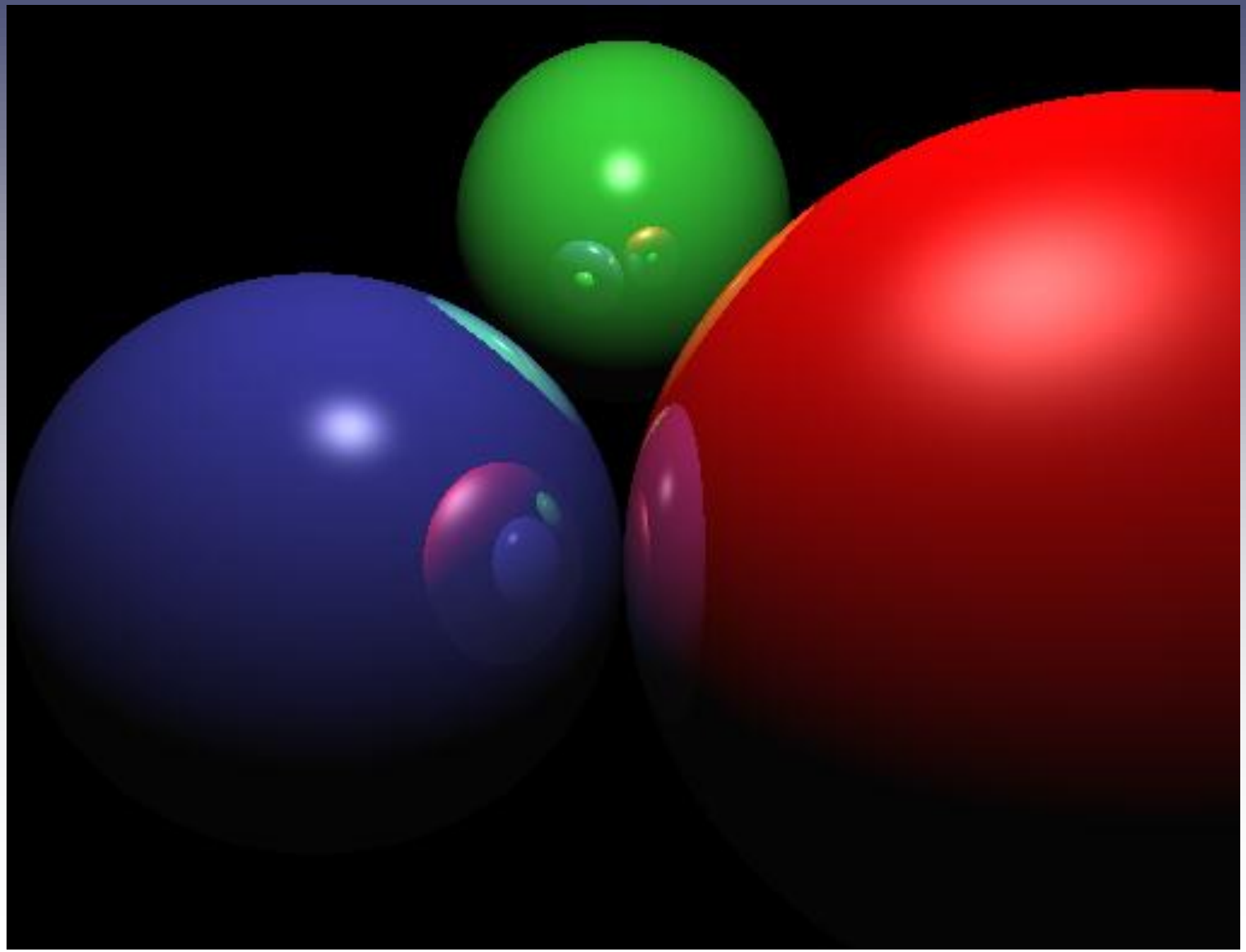
Example: Reflections at depth = 0



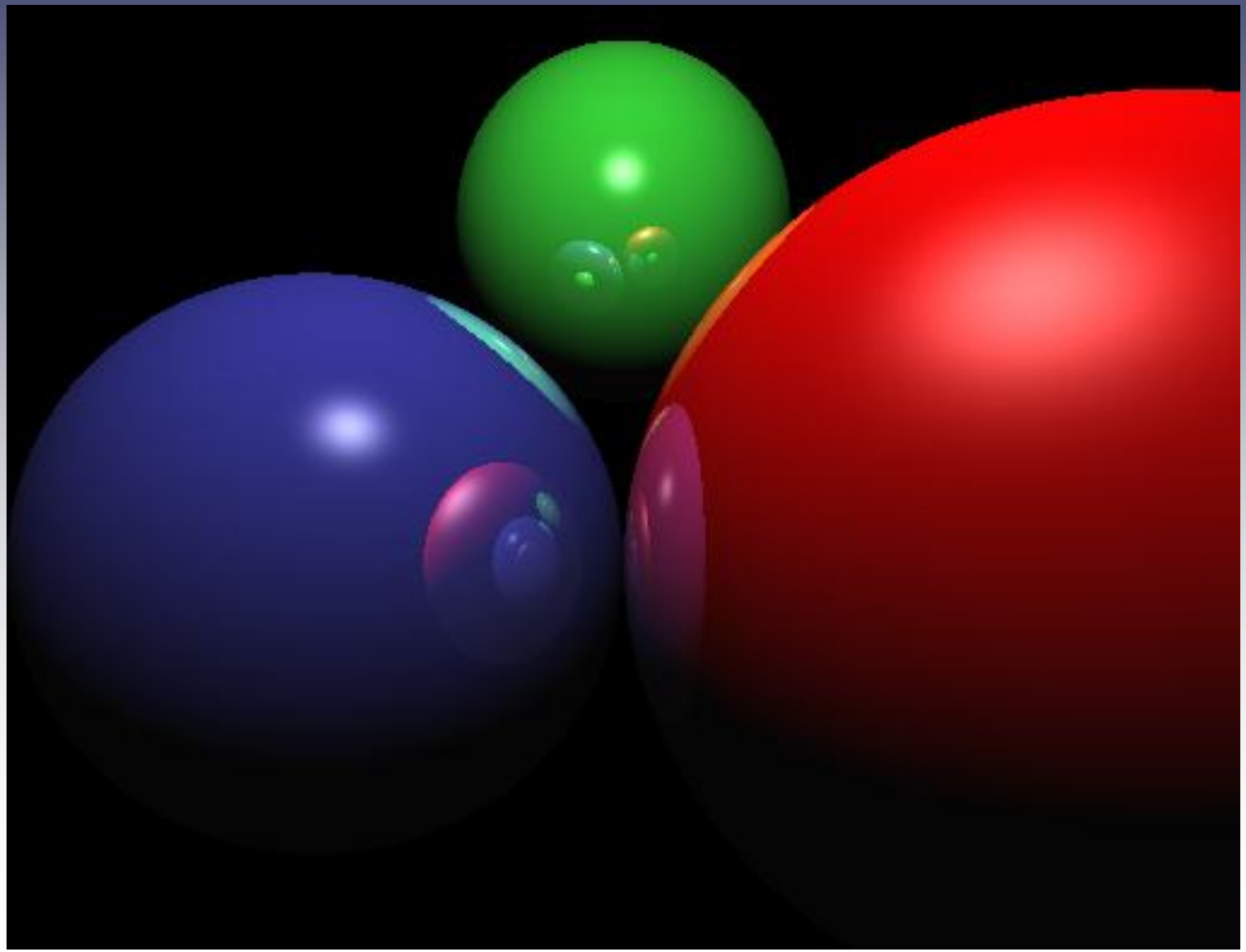
Example: Reflections at depth = 1



Example: Reflections at depth = 2

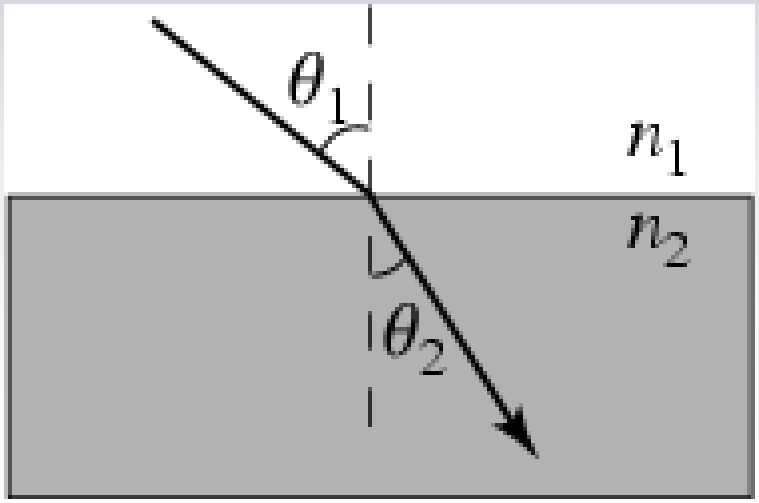


Example: Reflections at depth = 3



Refraction

- Definition: Bending of light ray as it crosses interface between media (e.g., air → glass or vice versa)
- Index of refraction (IOR) *n* for a medium: Ratio of speed of light in vacuum to that in medium (wavelength-dependent ⇒ prisms)
  - By definition, $n \geq 1$
  - Examples: n_{air} (1.00) < n_{water} (1.33) < n_{glass} (1.52)



courtesy of Wolfram

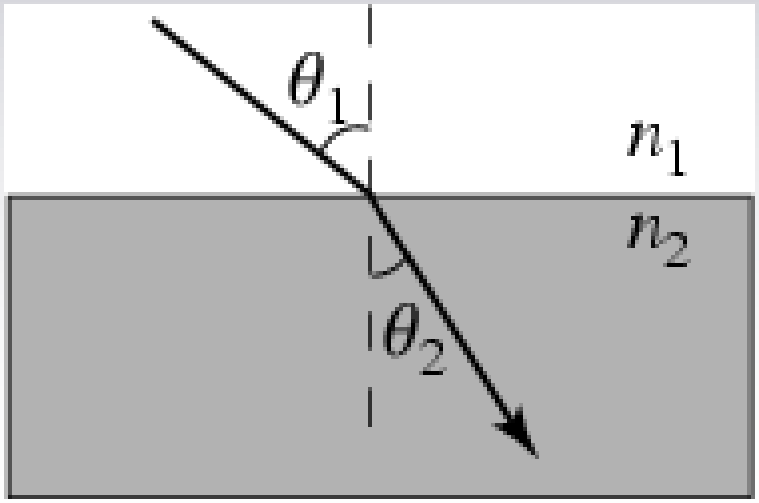
θ_1: Angle of incidence

θ_2: Angle of refraction

Snell's Law

- The relationship between the angle of incidence and the angle of refraction is given by:

$$n_1 \sin\theta_1 = n_2 \sin\theta_2$$



courtesy of Wolfram

Snell's Law: Implications

- Since $\theta \approx \sin\theta$ over the range $[0, \pi/2]$ and the angle of refraction is given by

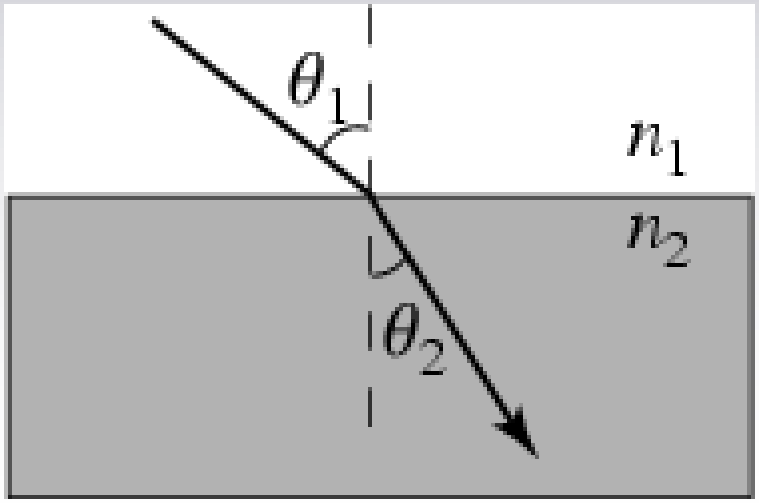
$$\sin\theta_2 = \frac{n_1}{n_2}\sin\theta_1$$

divergence

we can infer the following from their IORs:

$n_1 < n_2 \Rightarrow \theta_2 < \theta_1$ and $n_1 > n_2 \Rightarrow \theta_2 > \theta_1$

convergence



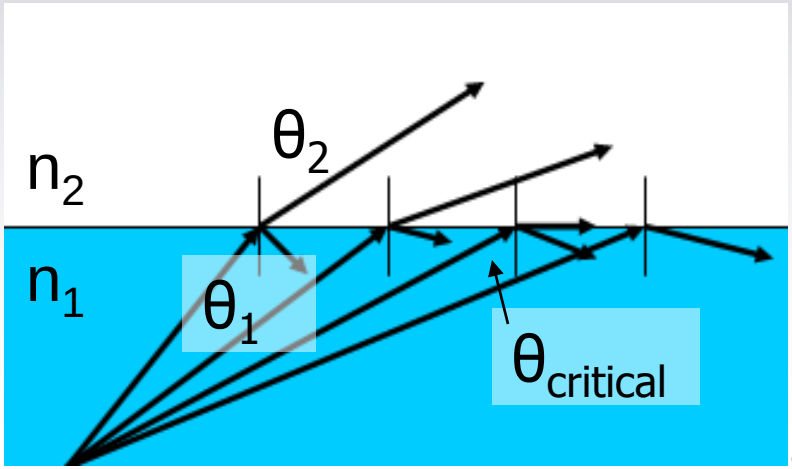
courtesy of Wolfram

So $n_1 < n_2$ in this image (like air to water)

Refraction: Critical Angle

- Snell's law $n_1 \sin\theta_1 = n_2 \sin\theta_2$ says that $n_1 > n_2 \Rightarrow \theta_2 > \theta_1$ (e.g., water to air), but biggest angle θ_2 that exiting ray can be bent is $\pi/2$ (along tangent to the surface)
- Thus, no light escapes—all light is reflected internally—for θ_1 greater than or equal to the **critical angle** of:

$$\theta_{critical} = \sin^{-1}\left(\frac{n_2}{n_1}\right)$$

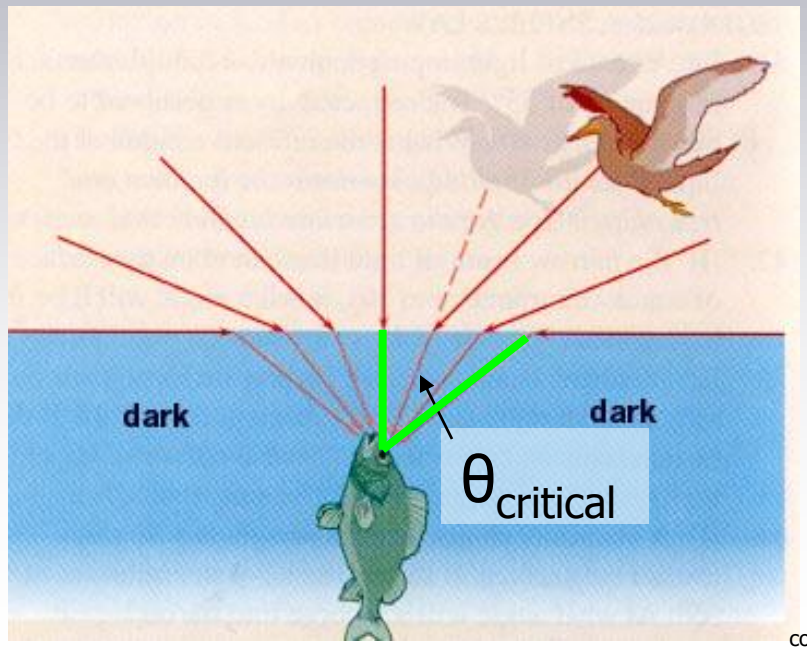


courtesy of G. Kessler

Critical Angle: Example

- Going from water (IOR = 1.33) to air (IOR = 1.00), we have:

$$\theta_{critical} = \sin^{-1}\left(\frac{1.00}{1.33}\right) \approx 48.75^{\circ}$$



courtesy of J. Alward

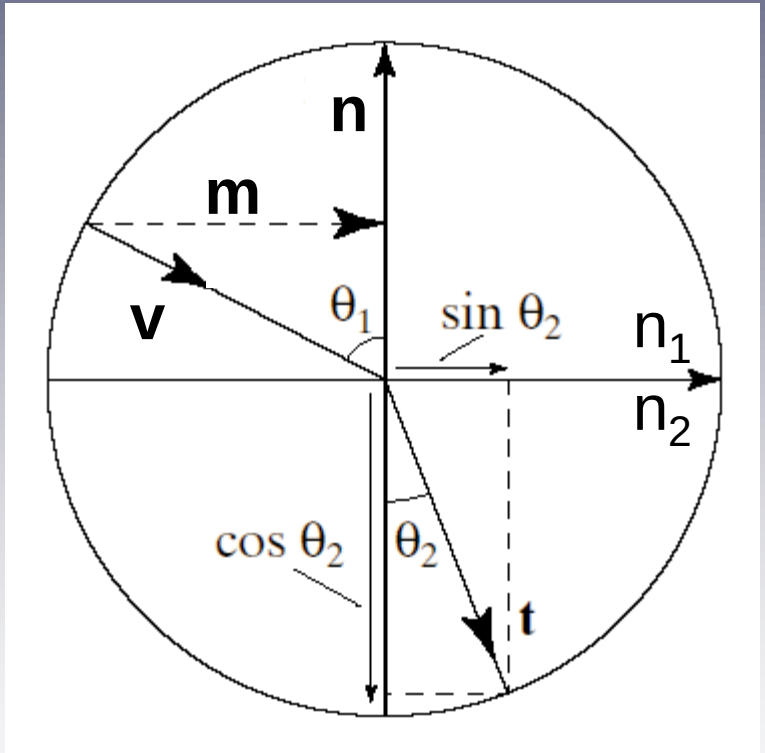
Computing the Transmission Direction **t**

$$n = \frac{n_1}{n_2}$$

$$c_1 = \cos\theta_1 = -\mathbf{v} \cdot \mathbf{n}$$

$$c_2 = \cos\theta_2 = \sqrt{1 - n^2(1 - c_1^2)}$$

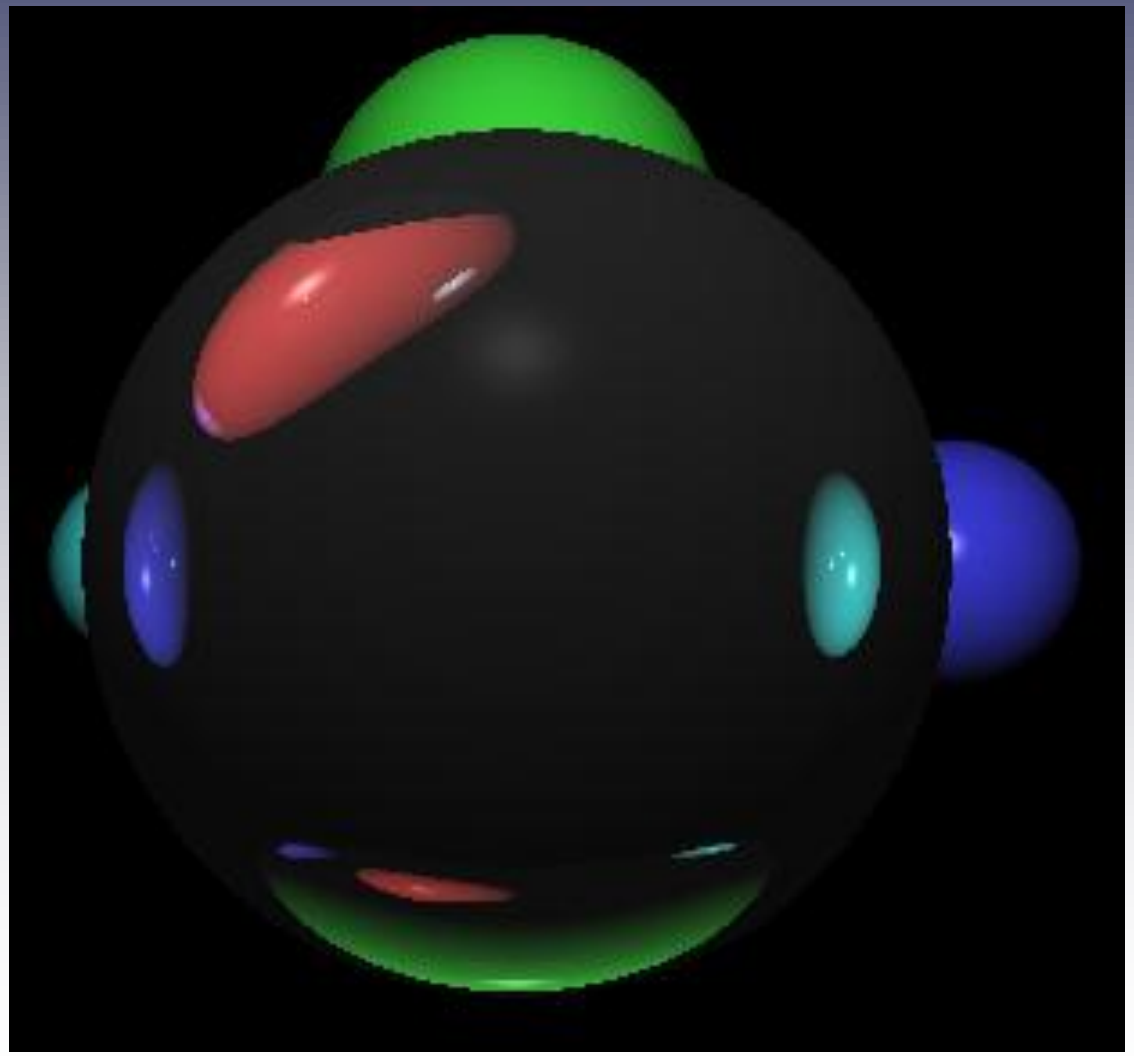
$$\mathbf{t} = n\mathbf{v} + (nc_1 - c_2)\mathbf{n}$$



adapted from Hill

Total internal reflection happens when the term in the square root above isn't positive, which is when $n^2(1 - c_1^2) \geq 1$

Example: Refraction



Ray Tracing Example (with texture mapping)

courtesy of J. Lee

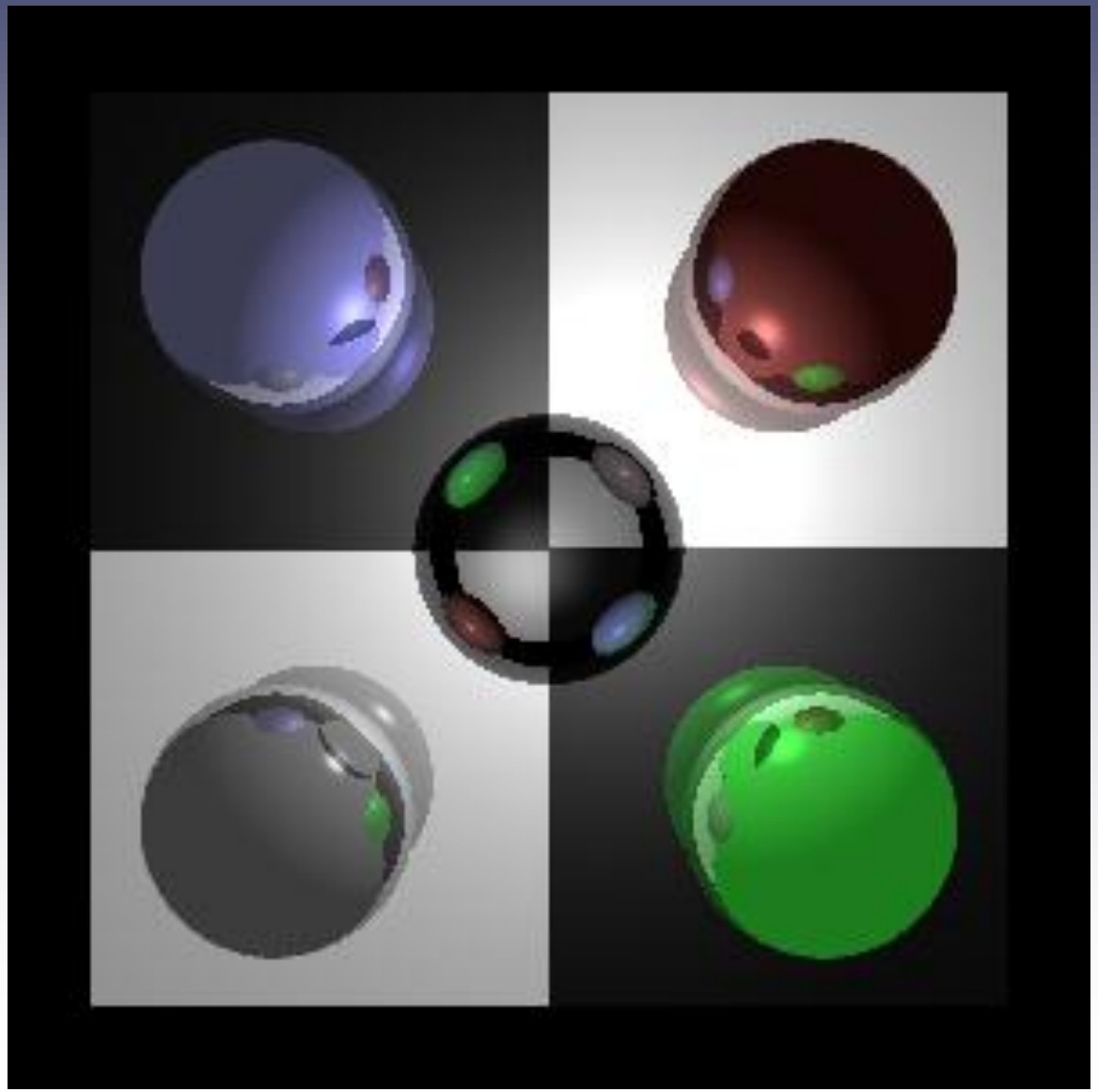
Basic Ray Tracing: Notes

- Global illumination effects simulated by basic algorithm are shadows, purely specular reflection/transmission
- Some outstanding issues
  - Aliasing, aka jaggies
  - Shadows have sharp edges, which is unrealistic
  - No diffuse reflection from other objects
- Intersection calculations are expensive, and even more so for more complex objects
  - Not currently suitable for real-time (i.e., games)

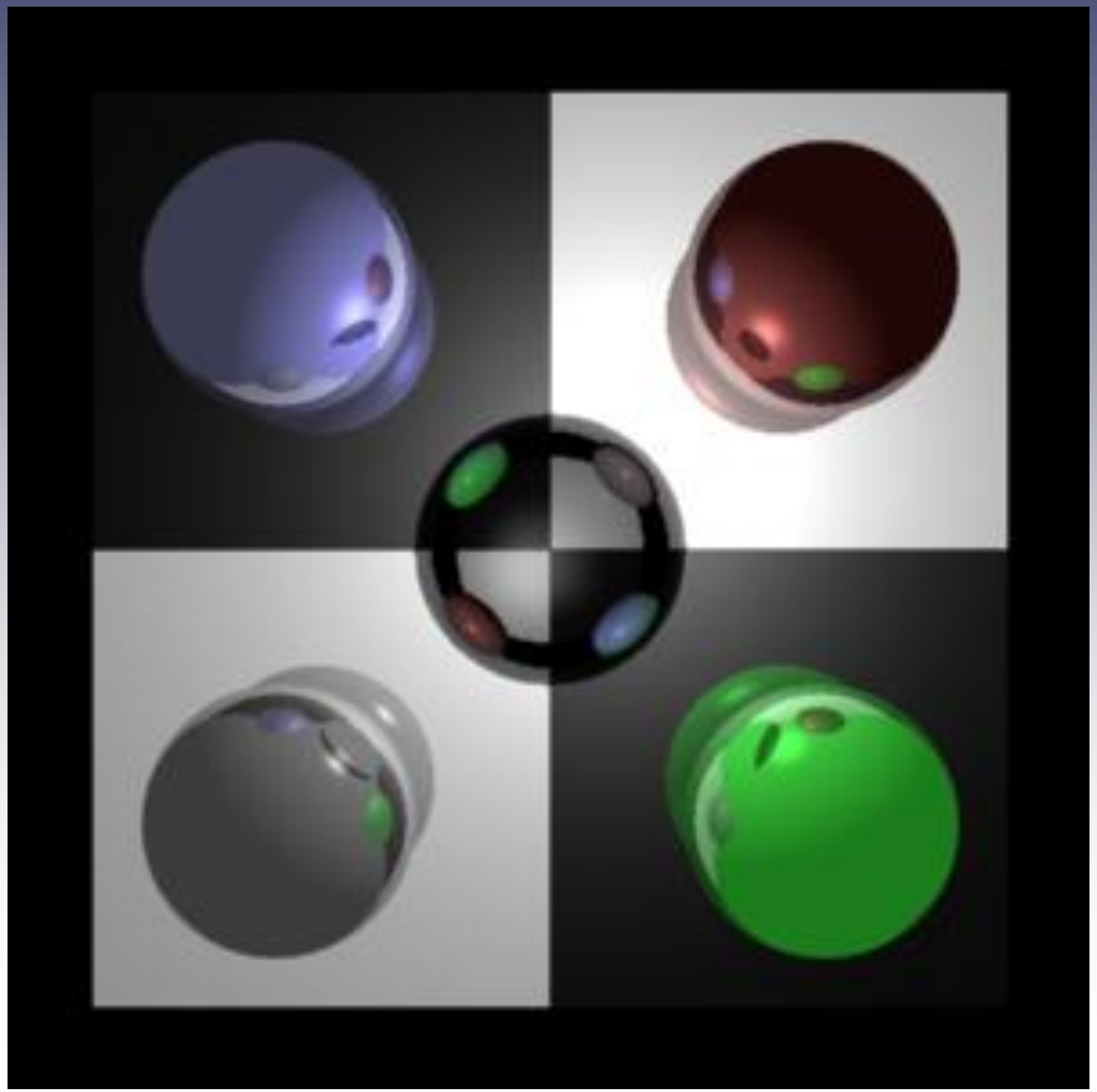
Distributed (aka "distribution") Ray Tracing (DRT)

- Basic idea: Use multiple eye rays for each pixel rendered or multiple recursive rays at intersections
- Application #1: Improving image quality via **anti-aliasing**
  - Supersampling: Shoot multiple nearby eye rays per pixel and combine colors
  - Uniform vs. adaptive: Constant number of rays or change in areas where image is changing more quickly

Aliasing/jaggies

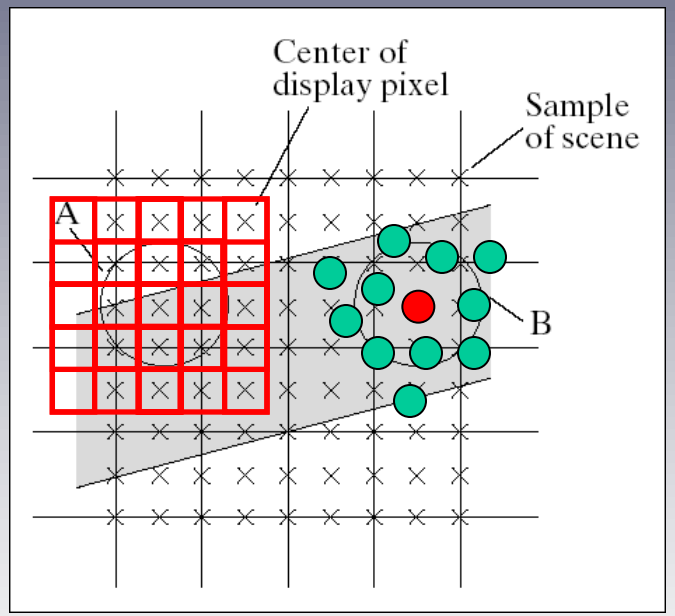


Anti-aliased image



Supersampling

- Rasterize at higher resolution
  - Regular grid pattern around each "normal" image pixel
  - Irregular **jittered** sampling pattern reduces artifacts
- Combine multiple samples into one pixel via **weighted average**
  - "Box" filter: All samples associated with a pixel have equal weight (i.e., directly take their average)
  - Gaussian/cone filter: Sample weights inversely proportional to distance from associated pixel



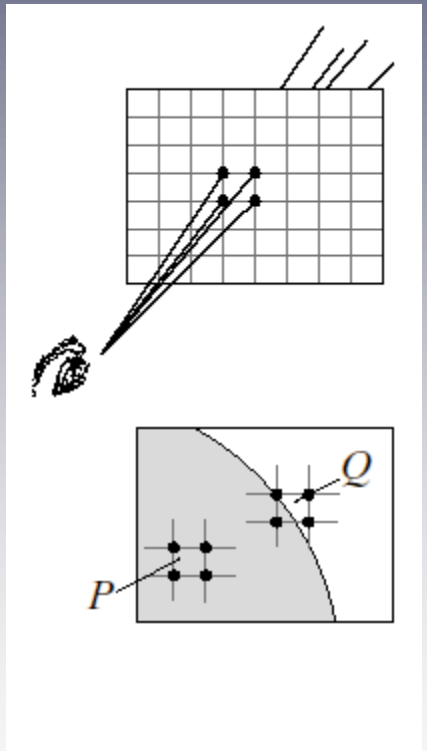
from Hill

Regular supersampling with 2x frequency

Jittered supersampling

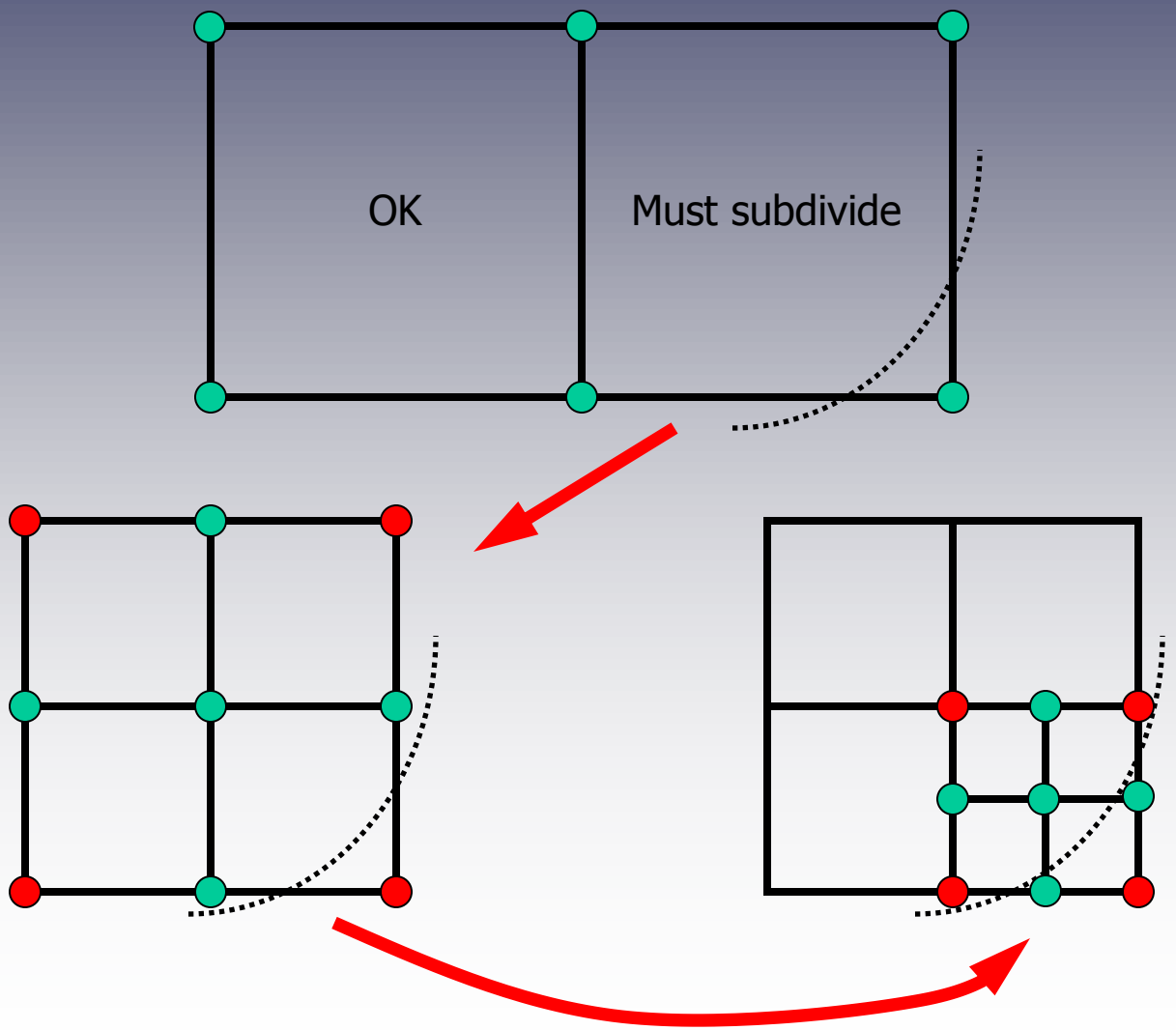
Adaptive Supersampling (Whitted's method)

- Shoot rays through 4 pixel corners and collect colors
- Provisional color for entire pixel is **average** of corner contributions
  - If you stop here, the only overhead vs. center-of-pixel ray-tracing is another row, column of rays
- If any corner's color is too different, **subdivide** pixel into quadrants and recurse on quadrants
- Details
  - Subdivide if any corner is more than 25% different from average (try experimenting with different thresholds here)
  - Maximum depth of 2 subdivisions sufficient



from Hill

Adaptive Supersampling: Details



HW #3

- Basic requirements
  - Complete shade_ray_**diffuse**()
  - Complete shade_ray_local(), which adds **specular and shadow** effects
  - Complete **reflection** component of shade_ray_recursive()
  - Add **sphere intersection testing** in intersect_ray_sphere()
  - Scene complexity and creativity
- Grad student requirements
  - Add support for **refraction** in shade_ray_recursive()
  - Add some version of adaptive supersampling, glossy reflection, ambient occlusion, or another advanced **distributed-ray** technique
  - Implement **bounding spheres** around objects to speed intersection calculations