Ray Tracing

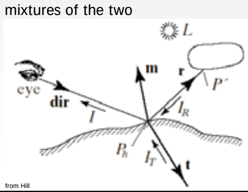
Thanks to UDel and MIT

Outline

- Recursive rays
  - Reflection
  - Refraction

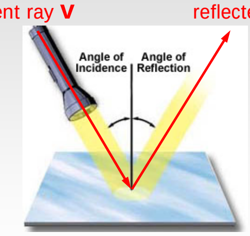
Ray Tracing

- Model: Perceived color at point **p** is an additive combination of local illumination (e.g., Phong), reflection, and refraction effects
- 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



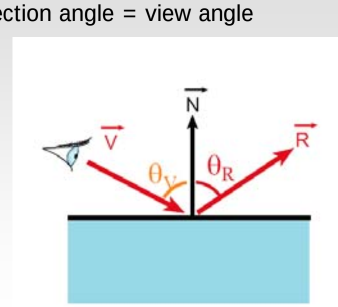
from Hill

Reflections



Reflection

- Reflection angle = view angle

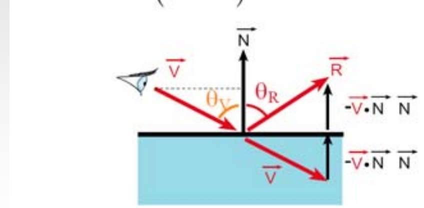


MIT EECS 6.837, Cutler and Durand

Reflection

- Reflection angle = view angle

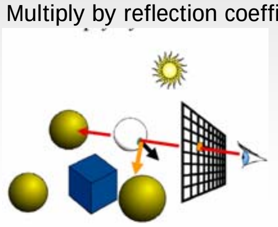
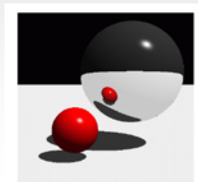
$$\vec{R} = \vec{V} - 2\left(\vec{V} \bullet \vec{N}\right)\vec{N}$$



MIT EECS 6.837, Cutler and Durand

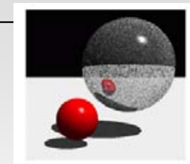
Mirror Reflection

- Compute mirror contribution
- Cast ray
  - –In direction symmetric wrtnormal
- Multiply by reflection coefficient (color)

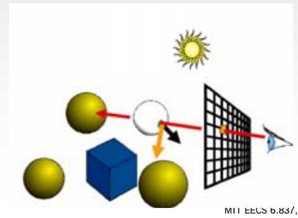
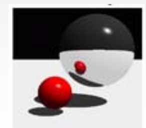



Mirror Reflection

- Cast ray
  - –In direction symmetric wrtnormal
- Don't forget to add epsilon to the ray



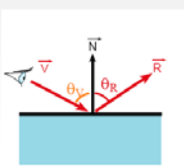
Without epsilon

With epsilon

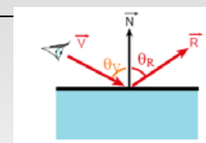
Amount of Reflection

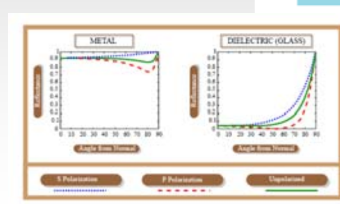
- Traditional (hacky) ray tracing
  - –Constant coefficient reflectionColor
  - –Component per component multiplication

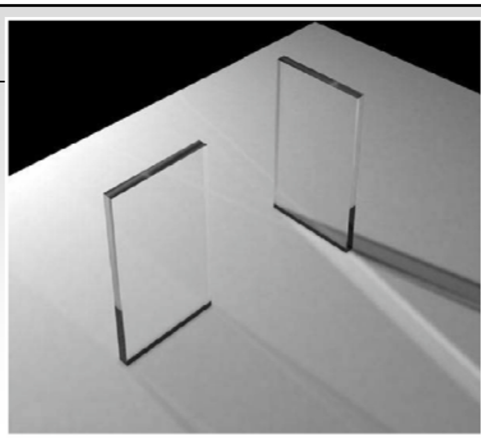


Amount of Reflection

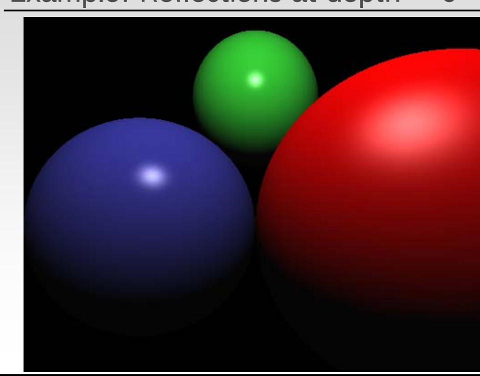
- More realistic:
  - –Fresnelreflection term
  - –More reflection at grazing angle
  - –Schlick'sapproximation: $R(\theta)=R0+(1-R0)(1-\cos\theta)^5$



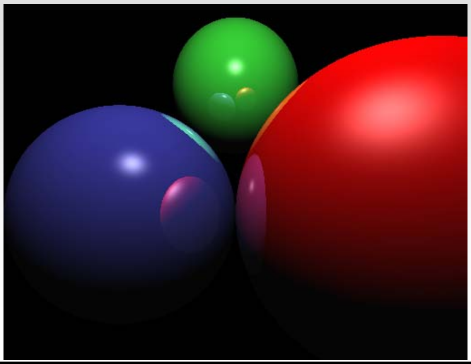




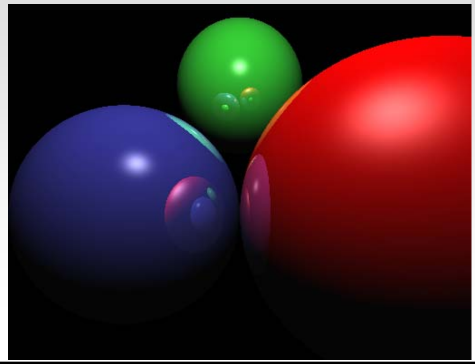
Example: Reflections at depth = 0



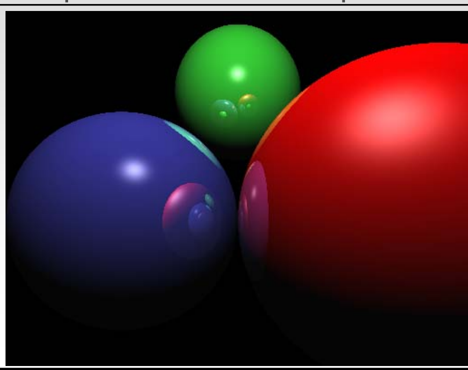
Example: Reflections at depth = 1



Example: Reflections at depth = 2

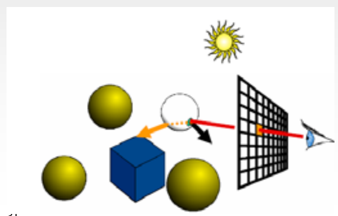


Example: Reflections at depth = 3



Transparency

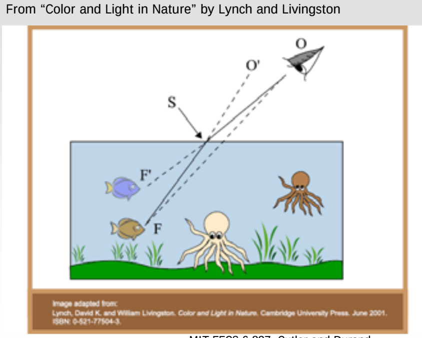
- Compute transmitted contribution
- Cast ray
  - In refracted direction
- Multiply by transparency coefficient (color)



31

Qualitative refraction

- From “Color and Light in Nature” by Lynch and Livingston

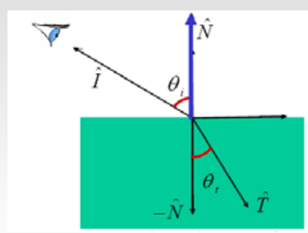


Refraction

Snell-Descartes Law

$n_i \sin\theta_i = n_t \sin\theta_t$

MATERIAL	INDEX OF REFRACTION
air/vacuum	1
water	1.33
glass about	1.5
diamond	2.4



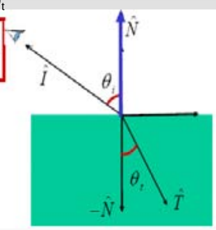
Note that I is the negative of the incoming ray

Refraction

Snell-Descartes Law

$n_i \sin\theta_i = n_t \sin\theta_t$

$$\boxed{\frac{\sin\theta_t}{\sin\theta_i} = \frac{\eta_i}{\eta_t} = \eta_r}$$



Note that I is the negative of the incoming ray

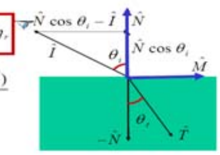
Refraction

Snell-Descartes Law

$$\boxed{\frac{\sin\theta_t}{\sin\theta_i} = \frac{\eta_i}{\eta_t} = \eta_r}$$

$$\hat{T} = \sin\theta_t \hat{M} - \cos\theta_t \hat{N}$$

$$\hat{M} = \frac{(\hat{N}\cos\theta_i - \hat{I})}{\sin\theta_i}$$



Note that I is the negative of the incoming ray

Refraction

Snell-Descartes Law

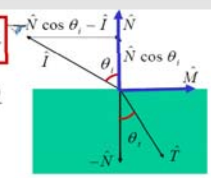
$$\boxed{\frac{\sin\theta_t}{\sin\theta_i} = \frac{\eta_i}{\eta t} = \eta_r}$$

$$\hat{T} = \sin\theta_t \hat{M} - \cos\theta_t \hat{N}$$

$$\hat{M} = \frac{(\hat{N}\cos\theta_i - \hat{I})}{\sin\theta_i}$$

$$\hat{T} = \frac{\sin\theta_t}{\sin\theta_i}(\hat{N}\cos\theta_i - \hat{I}) - \cos\theta_t \hat{N}$$

$$\hat{T} = (\eta_r \cos\theta_i - \cos\theta_t)\hat{N} - \eta_r \hat{I}$$



Note that I is the negative of the incoming ray

Refraction

Snell-Descartes Law

$$\boxed{\frac{\sin\theta_t}{\sin\theta_i} = \frac{\eta_i}{\eta_t} = \eta_r}$$

$$\hat{T} = \sin\theta_t \hat{M} - \cos\theta_t \hat{N}$$

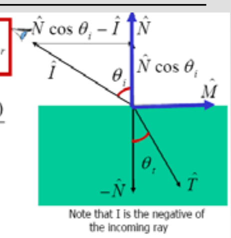
$$\hat{M} = \frac{(\hat{N}\cos\theta_i - \hat{I})}{\sin\theta_i}$$

$$\hat{T} = \frac{\sin\theta_t}{\sin\theta_i}(\hat{N}\cos\theta_i - \hat{I}) - \cos\theta_t \hat{N}$$

$$\hat{T} = (\eta_r \cos\theta_i - \cos\theta_t)\hat{N} - \eta_r \hat{I}$$

$$\cos\theta_i = \hat{N}\cdot\hat{I}$$

$$\cos\theta_t = \sqrt{1-\sin^2\theta_t} = \sqrt{1-\eta_r^2\sin^2\theta_i} = \sqrt{1-\eta_r^2(1-(\hat{N}\cdot\hat{I})^2)}$$



Note that I is the negative of the incoming ray

Use: $\sin^2\theta + \cos^2\theta = 1$

Refraction

Snell-Descartes Law

$$\boxed{\frac{\sin\theta_t}{\sin\theta_i} = \frac{\eta_i}{\eta_t} = \eta_r}$$

$$\hat{T} = \sin\theta_t \hat{M} - \cos\theta_t \hat{N}$$

$$\hat{M} = \frac{(\hat{N}\cos\theta_i - \hat{I})}{\sin\theta_i}$$

$$\hat{T} = \frac{\sin\theta_t}{\sin\theta_i}(\hat{N}\cos\theta_i - \hat{I}) - \cos\theta_t \hat{N}$$

$$\hat{T} = (\eta_r \cos\theta_i - \cos\theta_t)\hat{N} - \eta_r \hat{I}$$

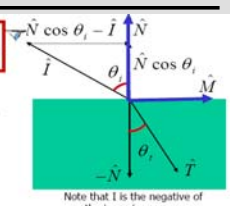
$$\cos\theta_i = \hat{N}\cdot\hat{I}$$

$$\cos\theta_t = \sqrt{1-\sin^2\theta_t} = \sqrt{1-\eta_r^2\sin^2\theta_i} = \sqrt{1-\eta_r^2(1-(\hat{N}\cdot\hat{I})^2)}$$

$$\boxed{\hat{T} = \left(\eta_r(\hat{N}\cdot\hat{I}) - \sqrt{1-\eta_r^2(1-(\hat{N}\cdot\hat{I})^2)}\right)\hat{N} - \eta_r\hat{I}}$$

Total internal reflection when the square root is imaginary

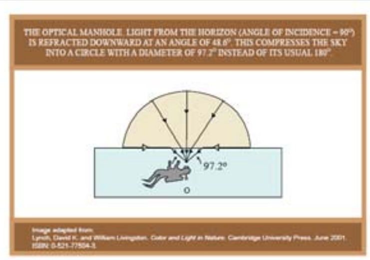
Don't forget to normalize



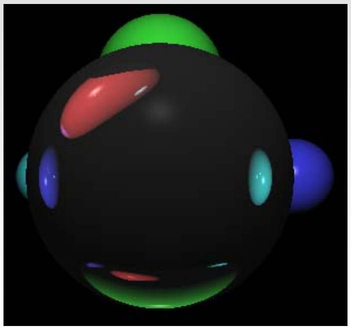
Note that I is the negative of the incoming ray

Total internal reflection

- From “Color and Light in Nature” by Lynch and Livingstone



Example: Refraction

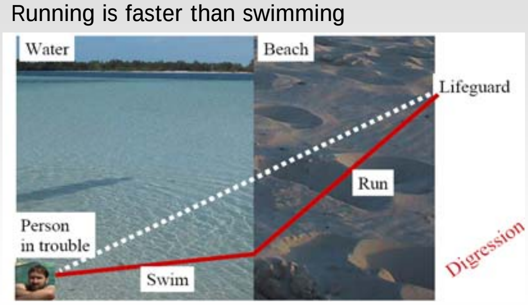


Ray Tracing Example (with texture mapping)

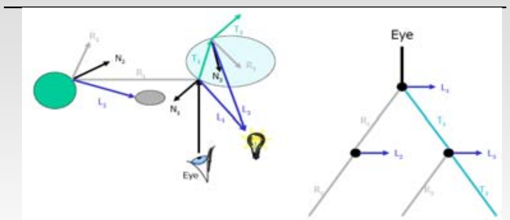
courtesy of J. Lee

Refraction and the lifeguard problem

- Running is faster than swimming



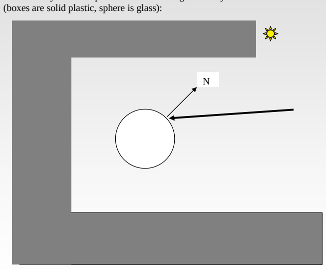
The Ray Tree



Ni surface normal
Ri reflected ray
Li shadow ray
Ti transmitted (refracted) ray

Ray Tree

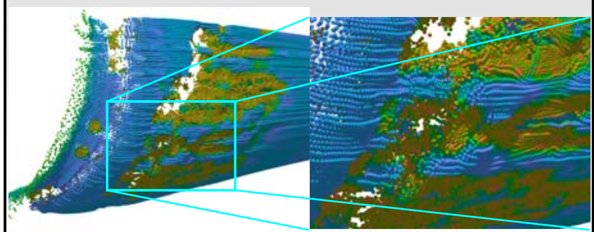
Draw the ray-tree to depth 3 for the following initial ray
(boxes are solid plastic, sphere is glass):



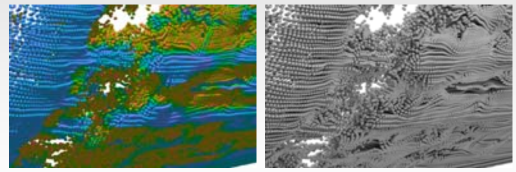
Basic Ray Tracing: Notes

- Intersection calculations are expensive, and even more so for more complex objects
  - Not currently suitable for real-time (i.e., games)
- Only global illumination effect is purely specular reflection/transmission
  - No “diffuse reflection” from other objects ! Still using ambient term
  - One remedy is **radiosity** (slow, offline, precompute
  - Ambient Occlusion
- Shadows have sharp edges, which is unrealistic – next lecture

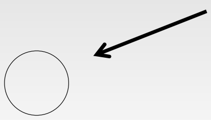
Phong shading



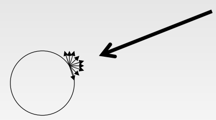
Ambient Occlusion



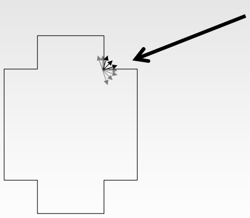
Ambient Occlusion



Ambient Occlusion



Ambient Occlusion

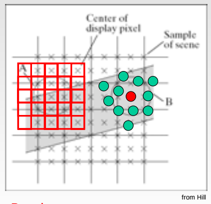


Ray Tracing: Improvements

- Image quality: Anti-aliasing
  - Supersampling: Shoot multiple rays per pixel (grid or jittered)
    - Adaptive: More rays in areas where image is changing more quickly
- Efficiency: Bounding extents
  - Idea: Enclose complex objects in shapes (e.g., sphere, box) that are less expensive to test for intersection
  - Next lecture

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

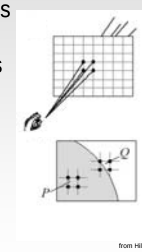


Regular supersampling with 2x frequency

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



Adaptive Supersampling: Details