More Triangles and Reflectance

CS 6965 Fall 2011

Further Reading

- Fast, Minimum Storage Ray/Triangle Intersection
  - Tomas Möller and Ben Trumbore
  - http://people.ksp.sk/~misof/skola/3D%20grafika%20(3ipg%204ipg)/Triangle%20intersection.pdf
- Optimizing Ray-Triangle Intersection via Automated Search
  - Andrew Kensler and Peter Shirley
  - http://www.cs.utah.edu/~aek/research/triangle.pdf

Triangle

- Several ways to do ray-triangle intersections:
  - Similar to disc (Dot products to perform inside/outside test)
  - Similar to disc (Project point to XY, XZ or YZ plane for inside/outside test)
  - Plücker coordinates
  - Barycentric coordinates

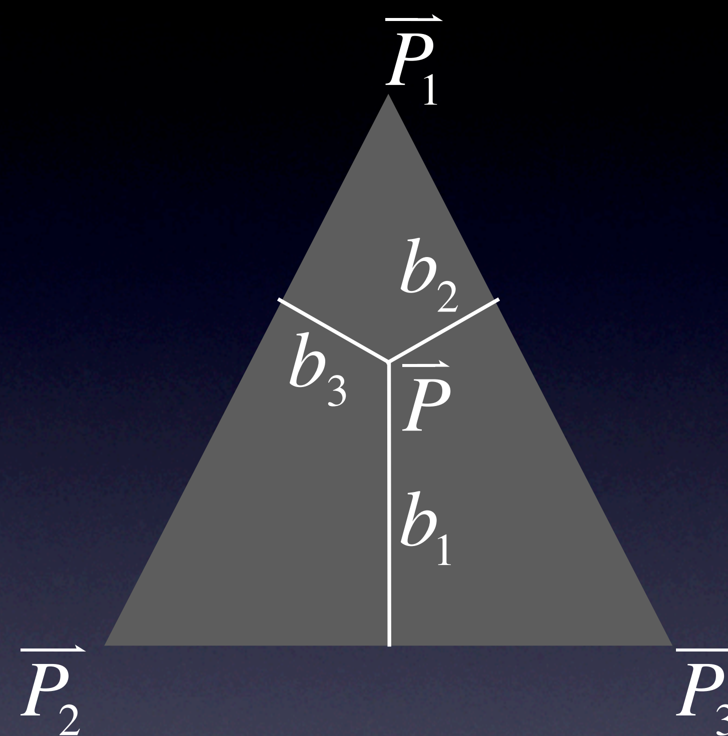
Barycentric coordinates

$$0 \le b_1, b_2, b_3 \le 1$$

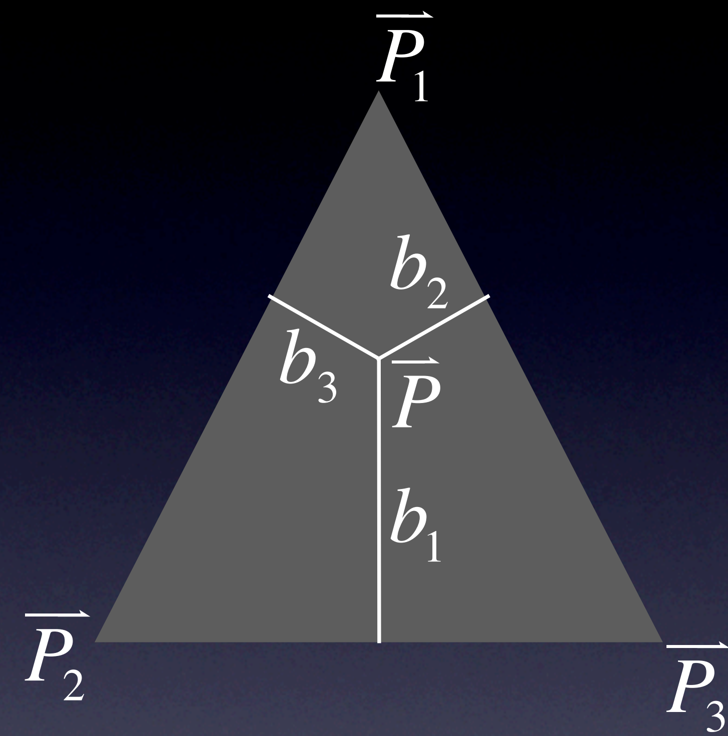
$$b_1 + b_2 + b_3 = 1$$

$$\vec{P} = b_1 \vec{P_1} + b_2 \vec{P_2} + b_3 \vec{P_3}$$

$$= b_1 \vec{P_1} + b_2 \vec{P_2} + \left(1 - b_1 - b_2\right) \vec{P_3}$$



Barycentric coordinates



$$0 \le b_1, b_2, b_3 \le 1$$

$$b_1 + b_2 + b_3 = 1$$

$$\overrightarrow{P} = b_1 \overrightarrow{P_1} + b_2 \overrightarrow{P_2} + b_3 \overrightarrow{P_3}$$

$$= b_1 \overrightarrow{P_1} + b_2 \overrightarrow{P_2} + \left(1 - b_1 - b_2\right) \overrightarrow{P_3}$$

$$\overrightarrow{O} + t\overrightarrow{V} = b_1 \overrightarrow{P_1} + b_2 \overrightarrow{P_2} + \left(1 - b_1 - b_2\right) \overrightarrow{P_3}$$

(3 equations, 3 unknowns)

Ray-triangle intersection

$$\overrightarrow{O} + t\overrightarrow{V} = b_1\overrightarrow{P_1} + b_2\overrightarrow{P_2} + \left(1 - b_1 - b_2\right)\overrightarrow{P_3}$$

$$-t\overrightarrow{V} + b_1\left(\overrightarrow{P_1} - \overrightarrow{P_3}\right) + b_2\left(\overrightarrow{P_2} - \overrightarrow{P_3}\right) = \left(\overrightarrow{O} - \overrightarrow{P_3}\right)$$

$$\overrightarrow{\mathrm{e}_1} = \left(\overrightarrow{P_1} - \overrightarrow{P_3}\right)$$

$$\overrightarrow{e_2} = \left(\overrightarrow{P_2} - \overrightarrow{P_3}\right)$$

$$\vec{s} = \left(\overrightarrow{O} - \overrightarrow{P_3}\right)$$

$$\begin{bmatrix} -V_x & e_{1x} & e_{2x} \\ -V_y & e_{1y} & e_{2y} \\ -V_z & e_{1z} & e_{2z} \end{bmatrix} \begin{bmatrix} t \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix}$$

Use Cramer's rule to solve system

Solution

$$t = \frac{\begin{vmatrix} s_x & e_{1x} & e_{2x} \\ s_y & e_{1y} & e_{2y} \\ s_z & e_{1z} & e_{2z} \end{vmatrix}}{\begin{vmatrix} -V_x & e_{1x} & e_{2x} \\ -V_y & e_{1y} & e_{2y} \\ -V_z & e_{1z} & e_{2z} \end{vmatrix}}, b_1 = \frac{\begin{vmatrix} -V_x & s_x & e_{2x} \\ -V_y & s_y & e_{2y} \\ -V_z & s_z & e_{2z} \end{vmatrix}}{\begin{vmatrix} -V_x & e_{1x} & e_{2x} \\ -V_y & e_{1y} & e_{2y} \\ -V_z & e_{1z} & e_{2z} \end{vmatrix}}, b_2 = \frac{\begin{vmatrix} -V_x & e_{1x} & s_x \\ -V_y & e_{1y} & s_y \\ -V_z & e_{1z} & s_z \end{vmatrix}}{\begin{vmatrix} -V_x & e_{1x} & e_{2x} \\ -V_y & e_{1y} & e_{2y} \\ -V_z & e_{1z} & e_{2z} \end{vmatrix}}$$

We will also use this property:

$$\left|A^T\right| = |A|$$

Improved solution

$$denom = \begin{vmatrix} -V_x & -V_y & -V_z \\ e_{1x} & e_{1y} & e_{1z} \\ e_{2x} & e_{2y} & e_{2z} \end{vmatrix}$$

$$= -\vec{V} \cdot \left(\vec{e_1} \times \vec{e_2} \right) = -\vec{e_1} \cdot \left(\vec{e_2} \times \vec{V} \right) = \vec{e_1} \cdot \left(\vec{V} \times \vec{e_2} \right)$$

$$denom = \vec{e_1} \cdot \left(\vec{V} \times \vec{e_2} \right)$$

$$t = \frac{e_2 \cdot \left(\vec{s} \times \vec{e_1} \right)}{denom}$$

$$b_1 = \frac{\vec{s} \cdot \left(\vec{V} \times \vec{e_2} \right)}{denom}$$

$$b_2 = \frac{\vec{V} \cdot \left(\vec{s} \times \vec{e_1} \right)}{denom}$$

Ray-triangle intersection

$$\overrightarrow{e_1} = \overrightarrow{p_1} - \overrightarrow{p_3}$$
$$\overrightarrow{e_2} = \overrightarrow{p_2} - \overrightarrow{p_3}$$
$$\overrightarrow{r_1} = \overrightarrow{V} \times \overrightarrow{e_2}$$
$$denom = \overrightarrow{e_1} \cdot \overrightarrow{r_1}$$
$$if\,(Abs(denom) < epsilon)\ \text{miss, return;}$$
$$invDenom = \frac{1}{denom}$$
$$\overrightarrow{s} = \overrightarrow{O} - \overrightarrow{p_3}$$
$$b_1 = \left(\overrightarrow{s} \cdot \overrightarrow{r_1}\right) invDenom$$
$$if\,(b_1 < 0 \,\|\, b_1 > 1)\ \text{miss, return;}$$
$$\overrightarrow{\mathrm{r}_2} = \overrightarrow{s} \times \overrightarrow{e_1}$$
$$b_2 = \left(\overrightarrow{V} \cdot \overrightarrow{r_2}\right) invDenom$$
$$if\,(b_2 < 0 \,\|\, \mathrm{b}_1 + b_2 > 1)\ \text{miss, return;}$$
$$t = \left(\overrightarrow{e_2} \cdot \overrightarrow{r_2}\right) invDenom$$
$$hit\text{, save b1/b2 for normal/texture interpolation}$$

	Add / sub / mult	*Compare*	*Divide*	$\sqrt{\ }$
$\overrightarrow{e_1} = \overrightarrow{p_1} - \overrightarrow{p_3}$	3			
$\overrightarrow{e_2} = \overrightarrow{p_2} - \overrightarrow{p_3}$	3			
$\overrightarrow{r_1} = \overrightarrow{V} \times \overrightarrow{e_2}$	9			
$denom = \overrightarrow{e_1} \cdot \overrightarrow{r_1}$	5			
$if\,(Abs(denom) < epsilon)$ miss, return;		2		
$invDenom = \frac{1}{denom}$			1	
$\overrightarrow{s} = \overrightarrow{O} - \overrightarrow{p_3}$	3			
$b_1 = \left(\overrightarrow{s} \cdot \overrightarrow{r_1}\right) invDenom$	6			
$if\,(b_1 < 0 \parallel b_1 > 1)$ miss, return;		2		
$\overrightarrow{\mathrm{r}_2} = \overrightarrow{s} \times \overrightarrow{e_1}$	9			
$b_2 = \left(\overrightarrow{V} \cdot \overrightarrow{r_2}\right) invDenom$	6			
$if\,(b_2 < 0 \parallel \mathrm{b}_1 + b_2 > 1)$ miss, return;	1	2		
$t = \left(\overrightarrow{e_2} \cdot \overrightarrow{r_2}\right) invDenom$	6			
hit, save b1/b2 for normal/texture interpolation		2		
total	20 / 29 / 45 / 51	2 / 4 / 6 / 8	0 / 1 / 1 / 1	0 / 0 / 0 / 0

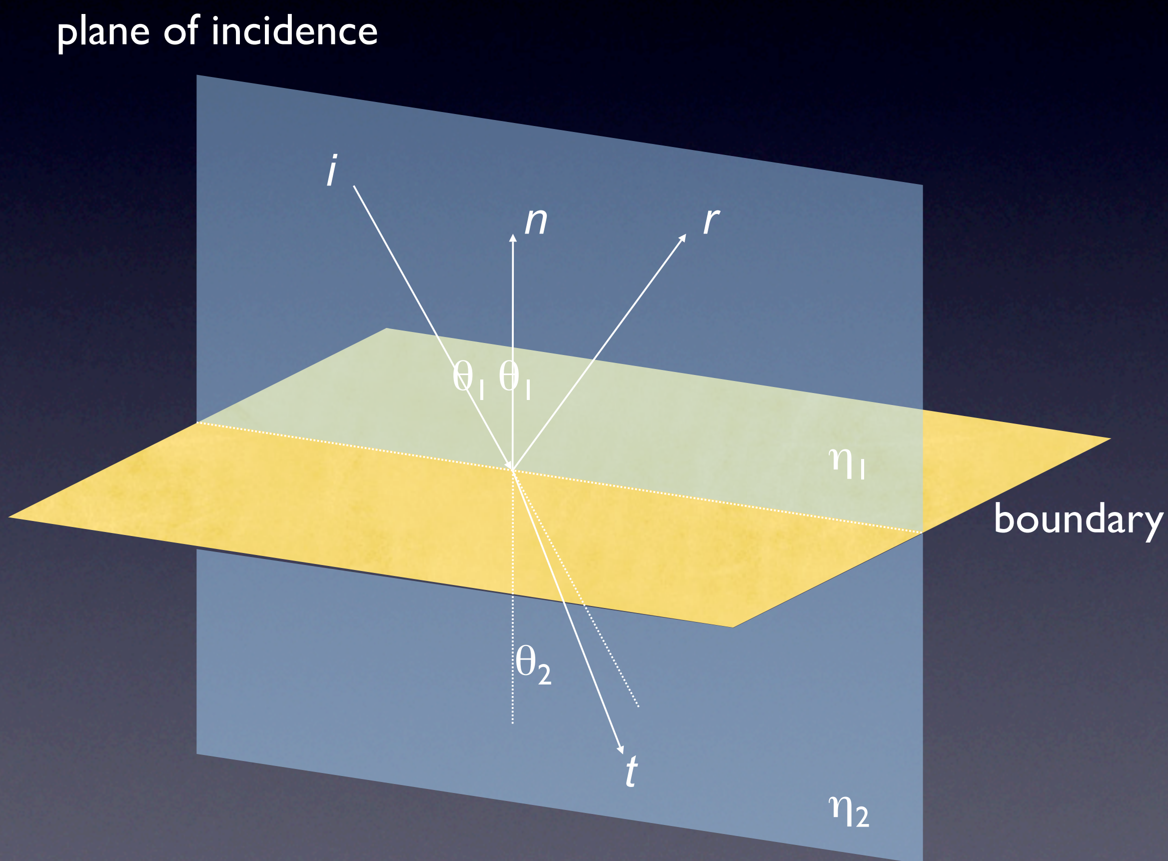
Lambertian shading

- Color at surface: $(\overrightarrow{N} \cdot \overrightarrow{L}) C_L$
- (where N and L are unit vectors)
-

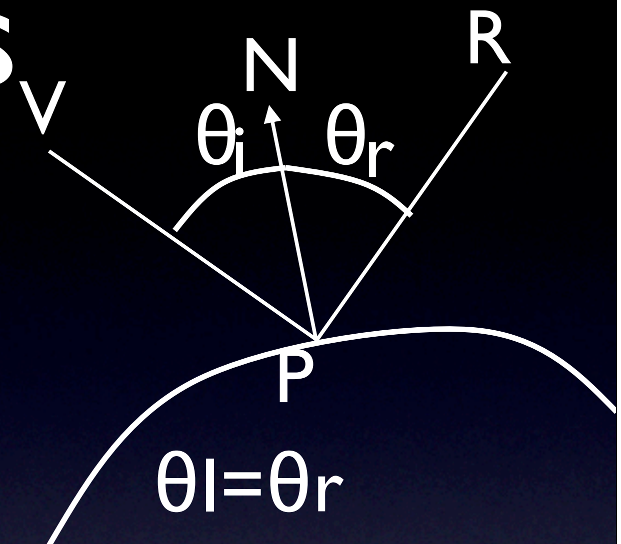
Light transport

- There are 4 primary ways that light interacts with a surface:
- Bounces off (perfect specular reflection)
- Absorbed and retransmitted in an arbitrary direction (perfect diffuse reflection)
- Travels through surface (perfect specular transmission)
- Absorbed and retransmitted on other side (perfect diffuse transmission)

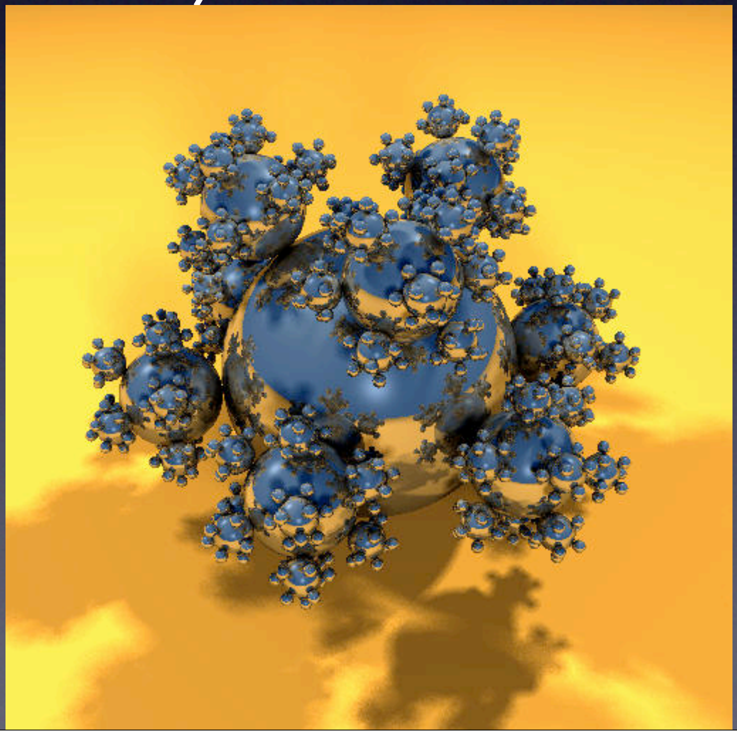
Reflected and transmitted rays



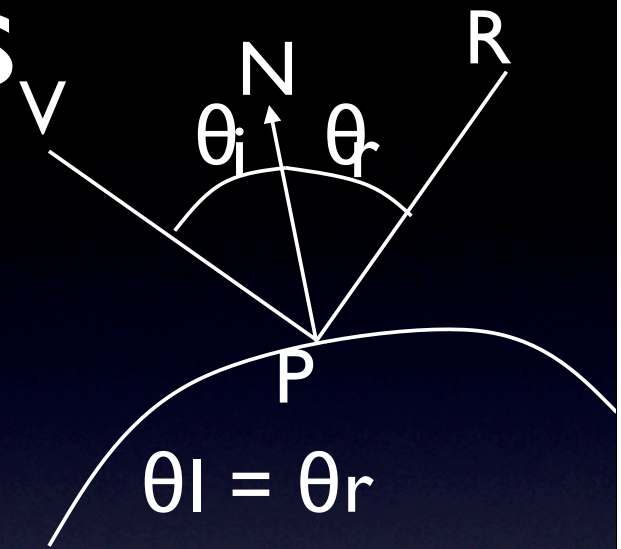
Reflections



- Reflection can be computed by tracing another ray from the intersection point
- Called perfect specular reflection



Reflections



- Perfect "bounce": incoming angle equal to outgoing angle
- θI is called "angle of incidence" or "incident angle"
- V, N and P define a plane
- Trace new ray on the same plane with origin at P and direction R
- Two derivations: algebraic and geometric

Algebraic derivation

Since R is coplanar with N and V:

$$\overline{R} = \alpha\overline{V} + \beta\overline{N}$$

$$\theta_i = \theta_r$$

$$\theta_i = -\overline{N}\cdot\overline{V} = \theta_r = \overline{N}\cdot\overline{R}$$

$$-\overline{N}\cdot\overline{V} = \overline{N}\cdot\left(\alpha\overline{V} + \beta\overline{N}\right)$$

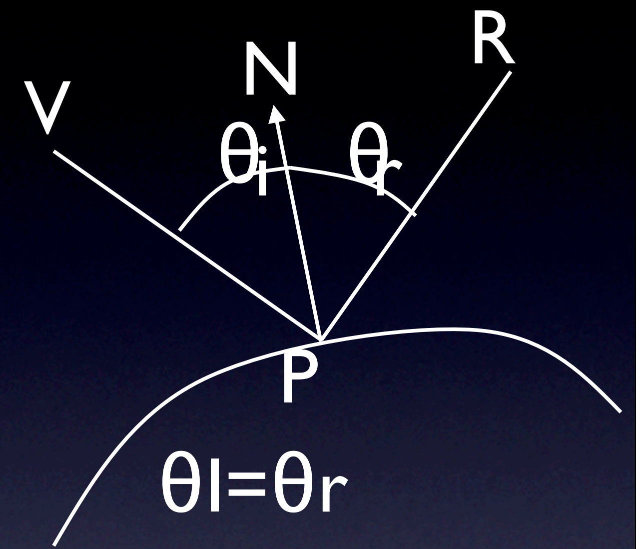
$$= \alpha\overline{N}\cdot\overline{V} + \beta\overline{N}\cdot\overline{N} \qquad (\overline{N}\cdot\overline{N} = 1)$$

$$= \alpha\overline{N}\cdot\overline{V} + \beta$$

Choose $\alpha = 1,$ since length doesn't matter

$$\beta = -2\overline{N}\cdot\overline{V}$$

$$\overline{R} = \overline{V} - 2\left(\overline{N}\cdot\overline{V}\right)\overline{N}$$



Vector length

If $\vec{V}$ and $\vec{N}$ are unit length (must be for the derivation)

$\vec{R}$ will be unit length:

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

$$\begin{aligned}
\vec{R} \cdot \vec{R} &= \left(\vec{V} - 2\left(\vec{N} \cdot \vec{V}\right)\vec{N}\right) \cdot \left(\vec{V} - 2\left(\vec{N} \cdot \vec{V}\right)\vec{N}\right) \\
&= \vec{V} \cdot \vec{V} - 4\left(\vec{N} \cdot \vec{V}\right)\left(\vec{N} \cdot \vec{V}\right) + 4\left(\vec{N} \cdot \vec{V}\right)^2\left(\vec{N} \cdot \vec{N}\right) \\
&= 1 - 4\left(\vec{N} \cdot \vec{V}\right)^2 + 4\left(\vec{N} \cdot \vec{V}\right)^2 \\
&= 1
\end{aligned}$$

Geometric derivation

$$\vec{S} = \vec{V} + \vec{N}\cos\theta_i$$

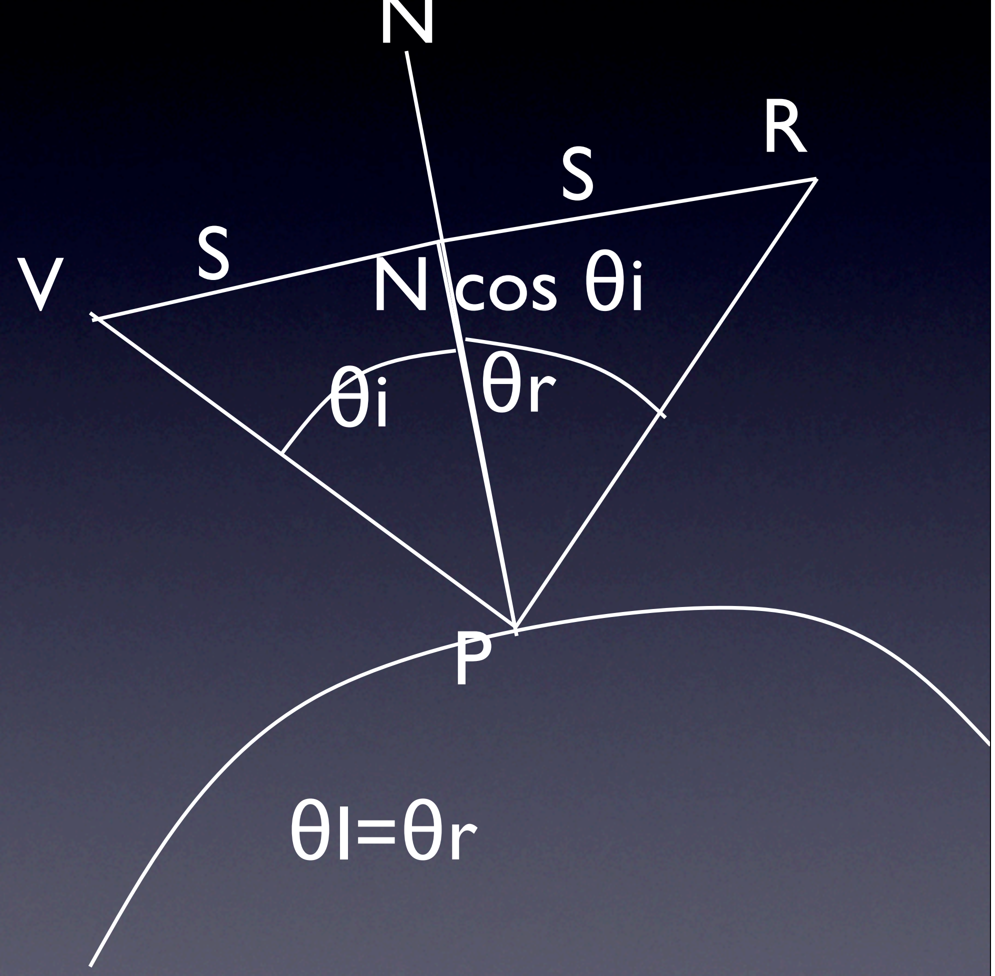
$$\vec{R} = \vec{N}\cos\theta_i + \vec{S}$$

$$\vec{R} = \vec{N}\cos\theta_i + \vec{V} + \vec{N}\cos\theta_i$$

$$\vec{R} = 2\vec{N}\cos\theta_i + \vec{V}$$

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

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



Reflected color

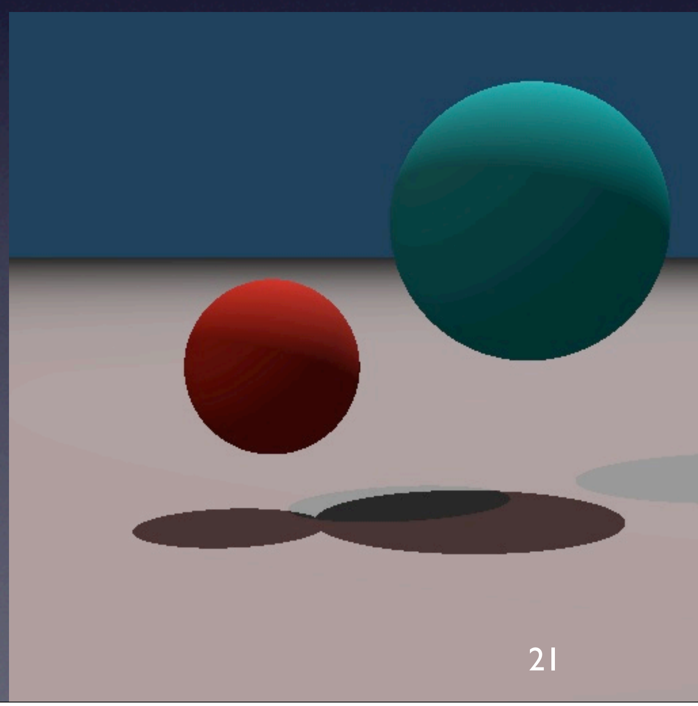
- In Material::shade, trace and shade a ray to get the color along R
- Reflectivity r (range 0-1)
- Result color = r * color along ray
- More about the reflectivity later

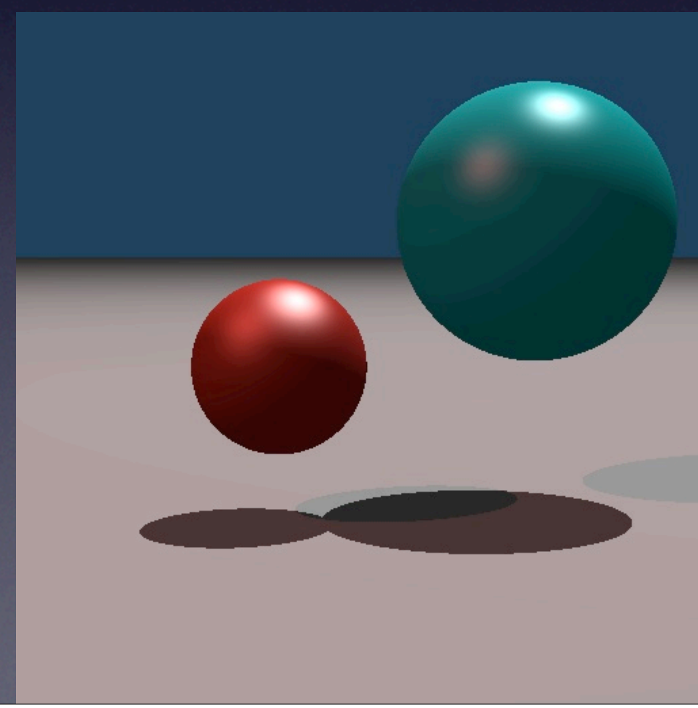
Phong lighting

- Most shiny surfaces reflect the light source
- A point light source has no reflection
- Light source reflections aren't perfect

Phong shading properties

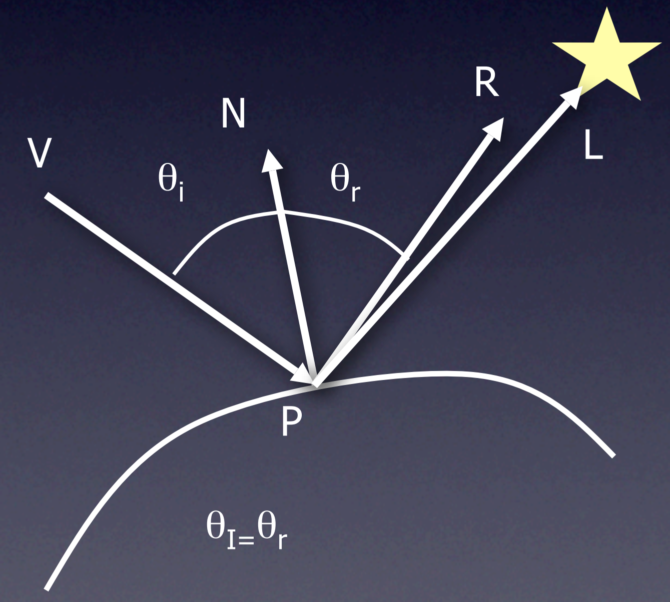
	From light sources	From other surfaces
Diffuse reflection	Lambertian term	-
Specular reflection	Phong term	-
Diffuse transmission	-	-
Specular transmission	-	-





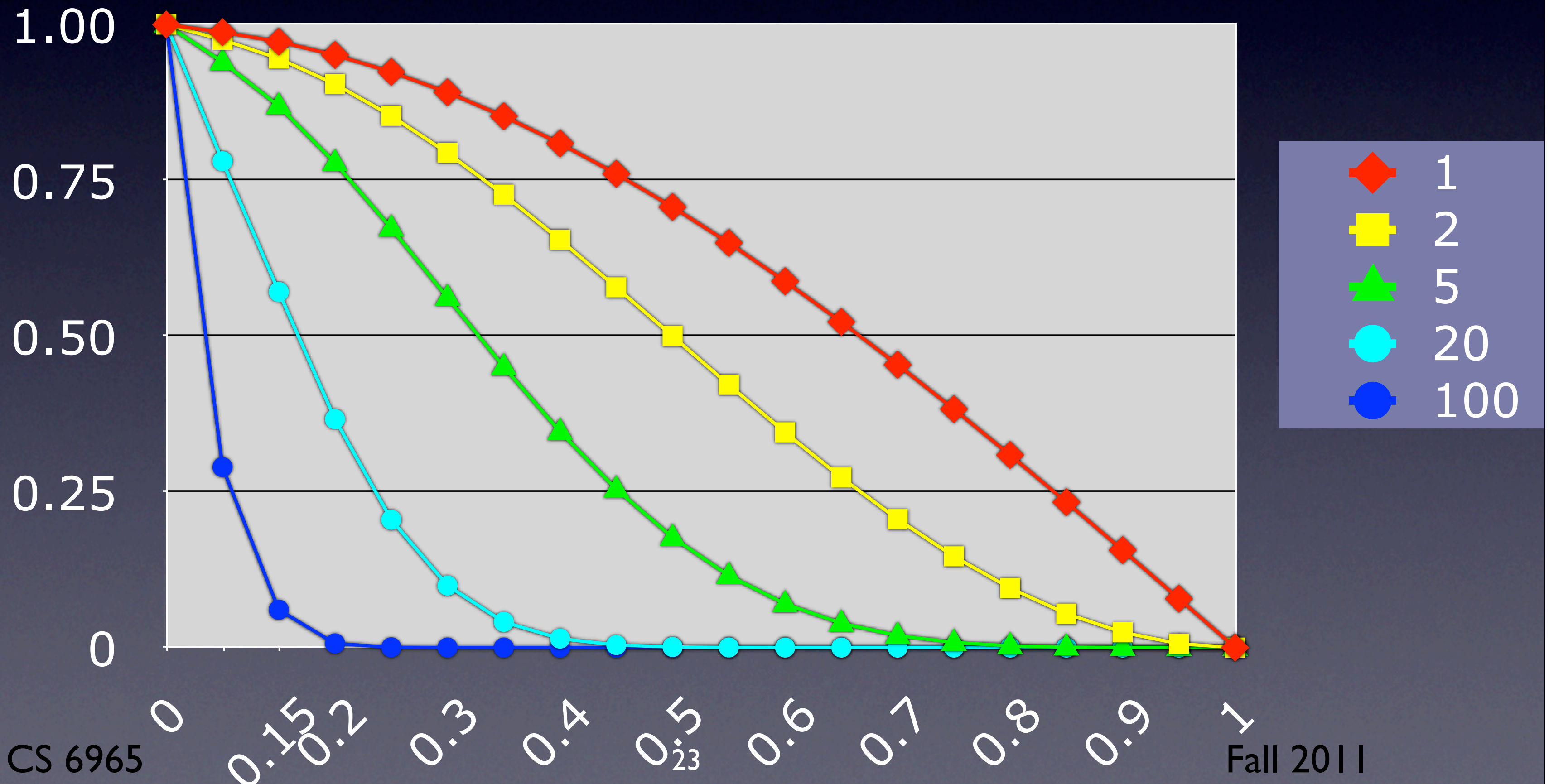
Phong shading

- Observation: light sources near the reflection direction are reflected into the eye
- Or when R•L is near 1
- How much?

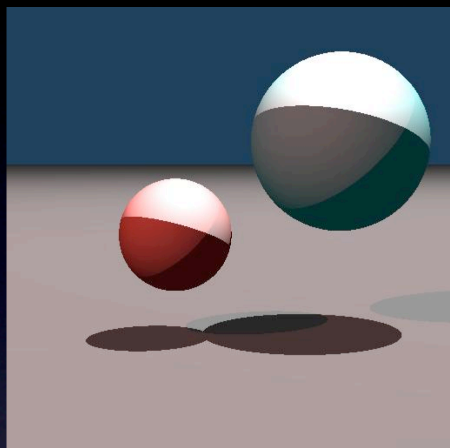


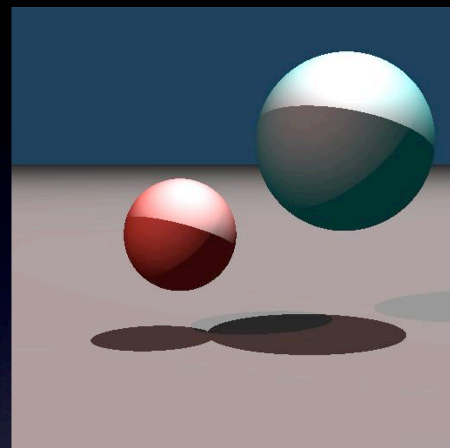
Phong shading

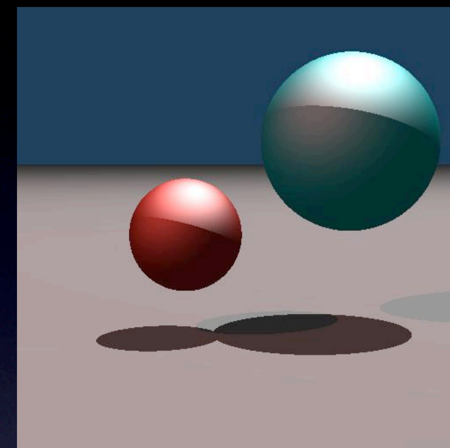
Phong highlights: $Ck_s\left(\vec{R}\cdot\vec{L}\right)^n$

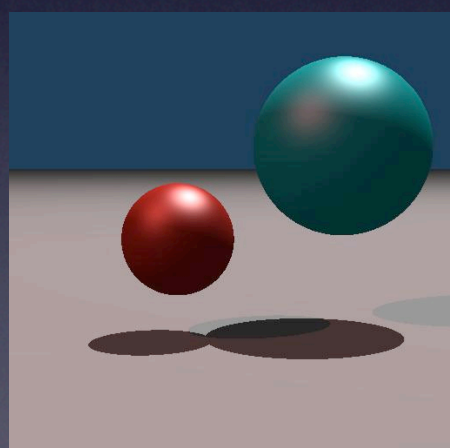


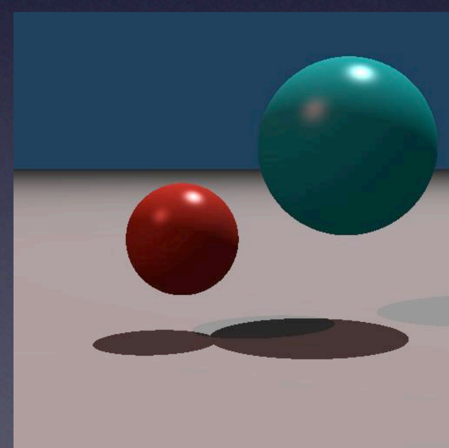
Phong shading


n = 1


n = 2


n = 5

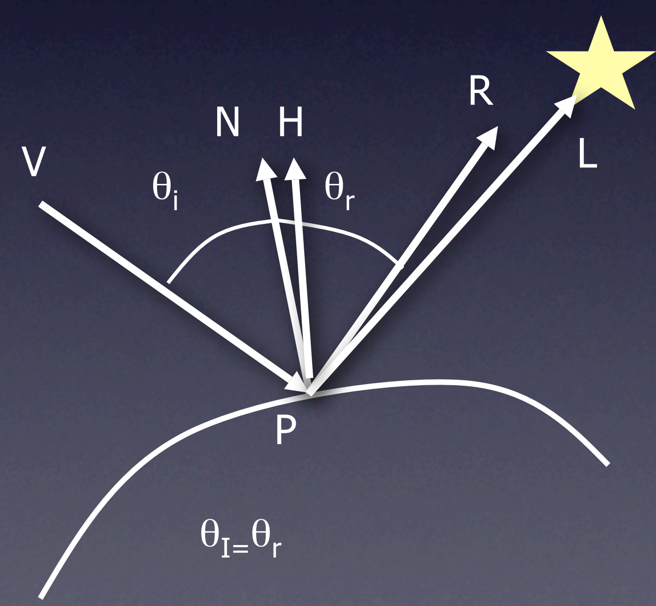

n = 20


n = 100

Better phong shading

$$\vec{H} = \frac{\vec{L} + \left(-\vec{V}\right)}{\left\| \vec{L} + \left(-\vec{V}\right) \right\|}$$

$$C = C_{body}\left(k_a + \sum_{lights} k_d \left(\vec{N} \cdot \vec{L} \right) \right) + C_s \sum_{lights} \left(\vec{N} \cdot \vec{H} \right)^n$$



Lambertian Shading

Compute hit position ($\vec{P} = \vec{O} + t\vec{V}$)

Call primitive to get normal ($\vec{\mathrm{N}}$) (normalized)

$costheta = \vec{N} \cdot \vec{V}$

$if\,(costheta > 0)$

normal =-normal

Color light = scene.ambient*Ka

foreach light source

get $\mathrm{C_L}$ and $\vec{\mathrm{L}}$

dist=$\|\vec{\mathrm{L}}\|, \overrightarrow{L_n} = \dfrac{\vec{L}}{\|\vec{L}\|}$

$cosphi = \vec{N} \cdot \overrightarrow{L_n}$

if($cosphi > 0$)

if(!intersect with $0 < t < dist$)

light += $\mathrm{C_L} * (Kd * cosphi)$

result=light*surface color

Phong Shading

...

Color light = scene.ambient*Ka

foreach light source

  get C_L and $\vec{L}$

  $\text{dist}=\|\vec{L}\|, \vec{L_n} = \frac{\vec{L}}{\|\vec{L}\|}$

  $cosphi = \vec{N} \cdot \vec{L_n}$

  $\text{if}(cosphi > 0)$

    $\text{if(!intersect with } 0 < t < dist)$

      $light \mathrel{+}= C_L * (Kd * \cos phi)$

      $\vec{H} = \frac{\vec{L} - \vec{V}}{\|\vec{L} - \vec{V}\|}$

      $cosalpha = \vec{H} \cdot \vec{N}$

      $if(cosalpha > 0)$

        $speclight \mathrel{+}= C_L cosalpha^n$

result=light*surface color + speclight *phong color