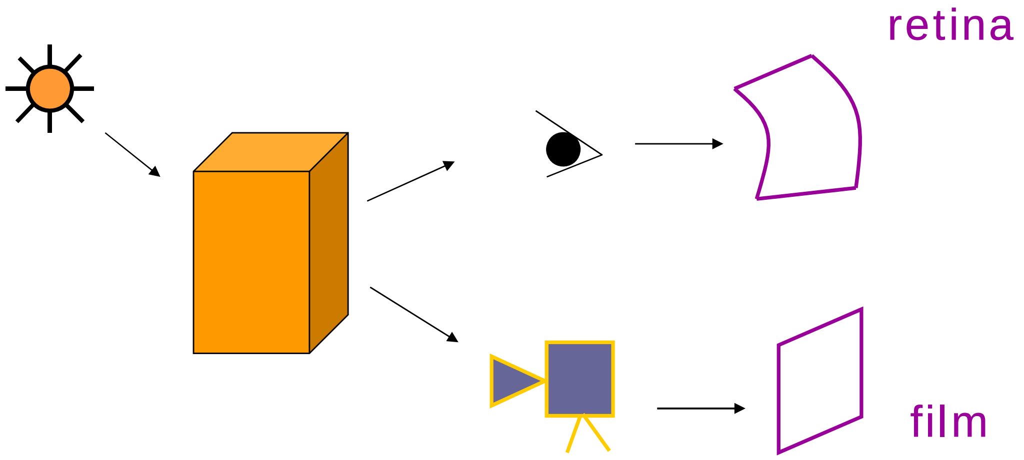
Lecture & Lab

Tue/Thur				Wednesday	
Date	Topic		Assignment		TA (LAB)
Mar 8, 10	Introduction an	Chap 1, Apx A	HW #0	9	OpenGL Intro 1 (Simple 2D)
Mar 15, 17	Linear and Affir	C	HW #1	14	open lab
Mar 22, 24	Frames in Grapl	C Chapters 1~13		23	OpenGL Intro 2 (3D & viewing)
Mar 29, 31	HelloWorld 3D,	C	HW #2	30	open lab
Apr 5, 7	Smooth Rotatic	C	<transformation 1>	6	open lab
Apr 12, 14	Depth, From Ve	Chap 10, 11, 12		13	<Election day>
Apr 19	(Projection), Va	Chap 13	HW #3	20	open lab
Apr 20~26	Midterm Exam		<transformation 2>		
Apr 28, May 3	Color and Shading			4	Lighting setup exercise
May 10, 12	Raytracing		HW #4	11	open lab
May 17, 19	Lighting		Shading/Lighting	18	open lab
May 24, 26	Texture Mapping	Chapters 14~23		25	Texture mapping exercise
May 31, Jun 2	Sampling		HW #5	1	open lab
Jun 7, 9	Resampling		Texture mapping	8	open lab
Jun 14	Animation				
Jun 15~21	Final Exam				

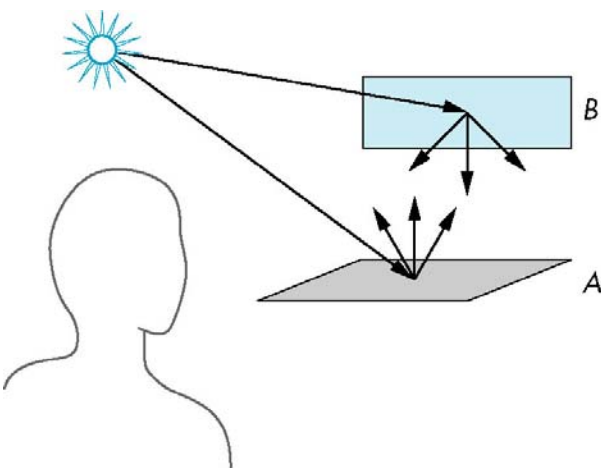
				Readings	Homework
Tue	May	3	Materials	Chap 14	
Wed		4	Lighting setup exercise		HW #3
Thur		5	<Children's Day>		Due (5/6)
		10	Shaders (Review+)	Chap 1~14	HW #4
		11	Open Lab		
		12	Color / Shading	Chap 19/Ext	
		17	Raytracing	Chap 20	
		18	Open Lab		
		19	Light	Chap 21	
		24	Texture Mapping 1	Chap 15	
		25	Texture mapping exercise		HW #5
		26	Lab		
		31	Texture Mapping 2	Chap 15	
7-10PM	June	1	CUDA Special Lab (by NVIDIA)		
		2	Sampling	Chap 16	
		7	Samplling/Reconstruction	Chap 16/17	
		8	Open Lab		
		9	Geometirc modeling	Chap 22	
		14	Animation	Chap 23	
June 16 (Thur) ? ←		21	Final Exam		

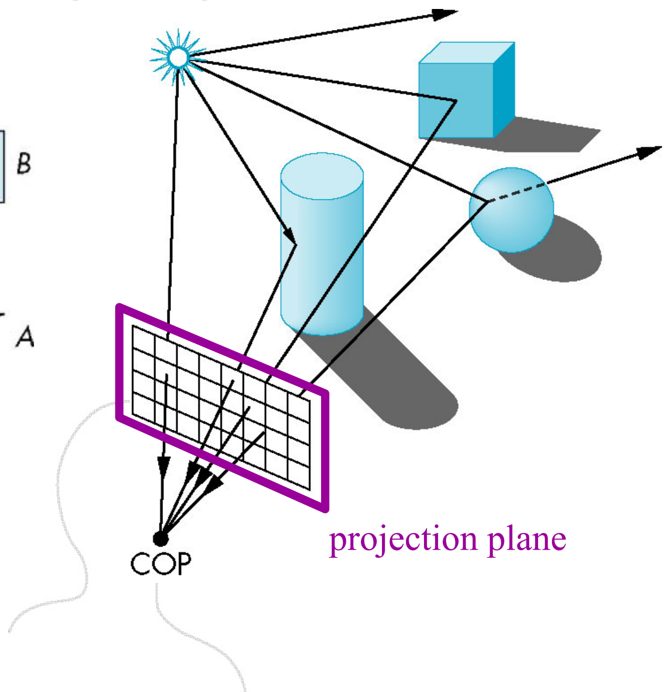
Physical Imaging Process



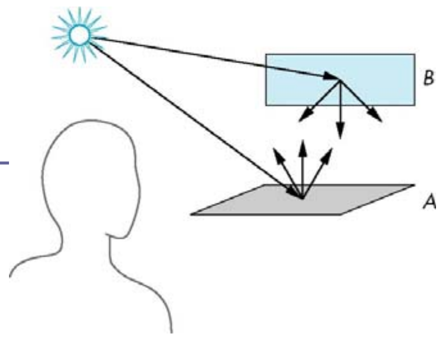
Light and Matter

- Rendering equation — very complex

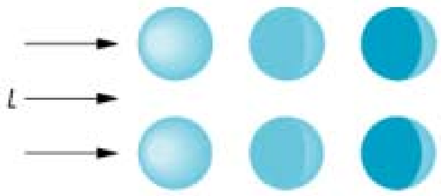




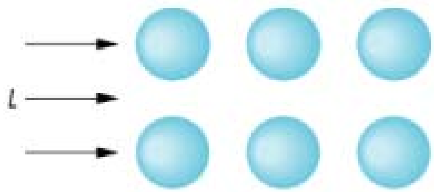
Light and Matter



- Rendering equation
  — very complex
- Ray tracing / Radiocity
  - global model

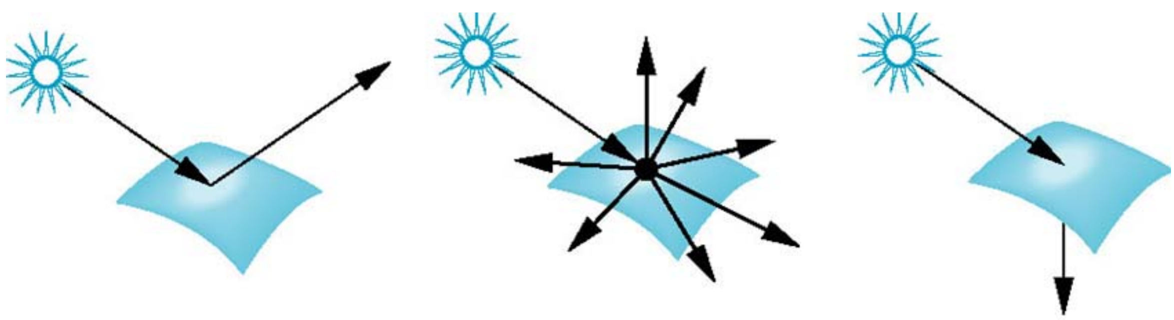


- Phong reflection model
  - local model
  - A point on the surface is independent of the other points

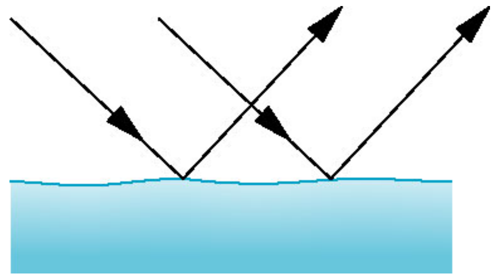


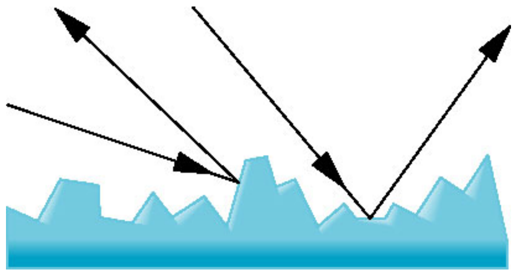
Light-Material Interactions

- Specular surfaces
  - Appear shiny because most of the reflected light is scattered in a narrow range. (mirror)
- Diffuse surfaces
  - Reflected light is scattered in all directions
- Translucent surfaces
  - Allow some light to penetrate the surface



- Perfectly *Specular* Surface
  = very smooth surface
  (mirror)





- Perfectly *Diffuse* Surface
  = very rough surface

- Light (Chapter 21)
- Ray Tracing (Chapter 20)
- Color (Chapter 19)

- Materials (Chapter 14)

Phong Reflection Model

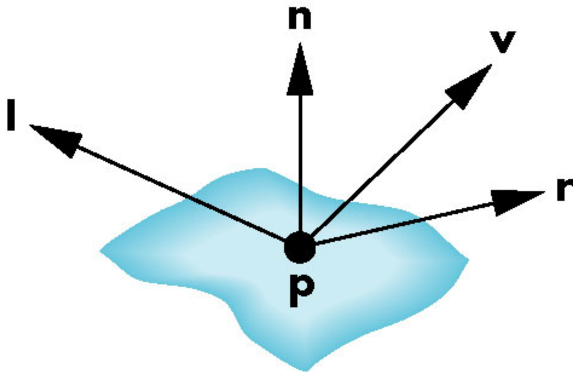
- 3 types of material-light interactions

  Reflection $\otimes$ Illumination $\longrightarrow$ intensity

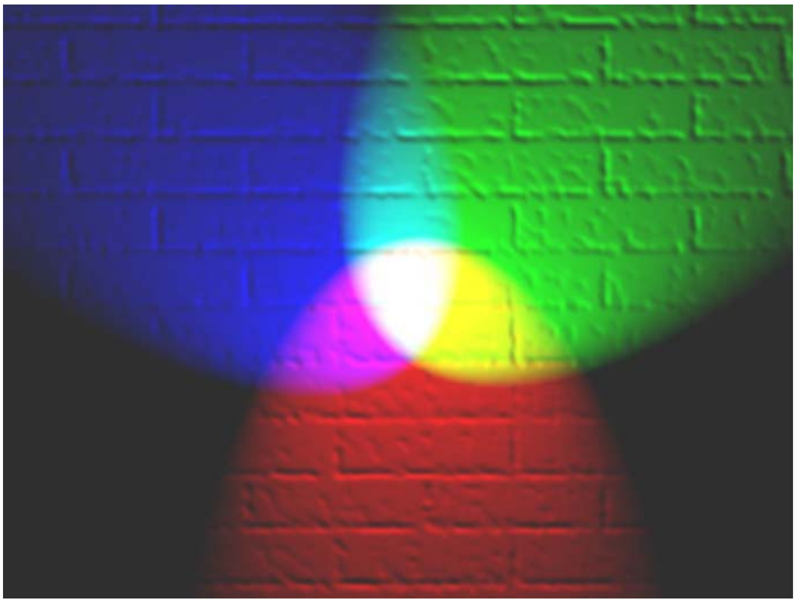
- Reflection Model
  - Ambient : same at every point
  - Diffuse : Lambert's law
  - Specular : shininess

Phong Reflection Model

- Efficient, and close enough to physical reality
- Supports ambient, diffuse and specular
  (material-light interactions)
- To compute a color at a point **p** on the surface, use 4 vectors
  - Surface normal
  - Direction from p to the viewer
  - Direction of a line from p to a light source
  - Direction of reflection



RGB Color Model



Phong Reflection Model

- Light source
  - L : illumination
  - For each light source i

$$L_i = \begin{array}{c c} \begin{array}{ccc} \text{red} & \text{green} & \text{blue} \end{array} & \\ \begin{pmatrix} L_{ira} & L_{iga} & L_{iba} \\ L_{ird} & L_{igd} & L_{ibd} \\ L_{irs} & L_{igs} & L_{ibs} \end{pmatrix} & \begin{array}{l} \textit{ambient} \\ \textit{diffuse} \\ \textit{specular} \end{array} \end{array}$$

In OpenGL,

Each source has separate diffuse, specular and ambient RGB parameters.

A real light source has but one color and cannot be characterized as being both a blue diffuse source and a white ambient source.

Instead of doing global lighting, we can use this added flexibility to give better approximations.

Phong Reflection Model

- Material model
  - R : reflection (how much of each of the incident lights is reflected at the point of interest)
  - At a point, it has the reflection for each light source

$$R_i = \begin{pmatrix} R_{ira} & R_{iga} & R_{iba} \\ R_{ird} & R_{igd} & R_{ibd} \\ R_{irs} & R_{igs} & R_{ibs} \end{pmatrix}$$

  - In OpenGL
    - Materials are modeled in a complementary manner.
    - For each surface, we must give separate ambient, diffuse, and specular components or use default values. These parameters are the fraction of the incoming light of each type that is reflected.

Phong Reflection Model

- Light source
  - L : illumination
  - For each light source i

$$L_i = \begin{array}{cccc} \text{red} & \text{green} & \text{blue} & \\ \left(\begin{matrix} L_{ira} \\ L_{ird} \\ L_{irs} \end{matrix}\right. & \begin{matrix} L_{iga} \\ L_{igd} \\ L_{igs} \end{matrix} & \left.\begin{matrix} L_{iba} \\ L_{ibd} \\ L_{ibs} \end{matrix}\right) & \begin{matrix} \textit{ambient} \\ \textit{diffuse} \\ \textit{specular} \end{matrix} \end{array}$$

- Material model
  - R : reflection (how much of each of the incident lights is reflected at the point of interest)
  - At a point, it has the reflection for each light source

$$R_i = \begin{pmatrix} R_{ira} & R_{iga} & R_{iba} \\ R_{ird} & R_{igd} & R_{ibd} \\ R_{irs} & R_{igs} & R_{ibs} \end{pmatrix}$$

Intensity at point p: **I = Reflection ⊗ Illumination**

Phong Reflection Model

- For each light source
  - For each color component
    - Ambient + Diffuse + Specular
- Illumination for the i^{th} light source L_i
- Reflection term for each color component r,g,b (e.g., R_{ira})
  - depends on
    - the material properties
    - the orientation of the surface
    - the direction of the light source
    - and the distance between the light source and the viewer

Phong Reflection Model

- For each light source
  - For each color component
    - Ambient + Diffuse + Specular
- (Red) intensity that we see at point **p**

$$I_r = \underbrace{\sum_i I_{ir}}_{\textit{for each directional light source}} + \underbrace{I_{ar}}_{\textit{global ambient light}}$$

L: illumination
R: reflection

$$I_{ir} = R_{ira}L_{ira} + R_{ird}L_{ird} + R_{irs}L_{irs}$$
$$= I_{ira} + I_{ird} + I_{irs}$$

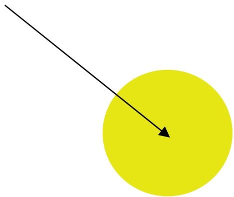
- $I = I_a + I_d + I_s = R_aL_a + R_dL_d + R_sL_s$

Ambient Reflection

- Intensity of ambient light L_a is the same at every point on the surface
- Some of this light is absorbed and some is reflected
- Amount of reflected is given by the ambient reflection coefficients k_{ar} , k_{ag} , k_{ab} $(0 \leq R_a \leq 1)$
- $I_a = k_a L_a$

$$R_a = k_a$$

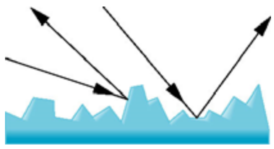
White *ambient* Light $(L_{ar}=L_{ag}=L_{ab}=1.0)$



$k_{ar} = 0.9$
$k_{ag} = 0.9$
$k_{ab} = 0.1$

Diffuse Reflection

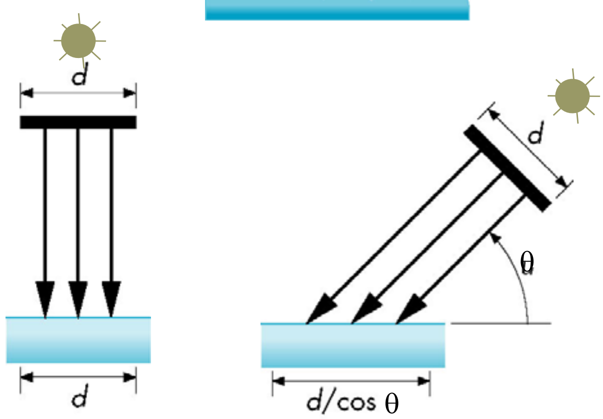
- Perfectly diffuse surface = very rough
  - There is no preferred angle of reflection
  - Lambertian surface
  - Obeys Lambert's law



$$R_d \propto \cos\theta$$

$$\cos\theta = \mathbf{l} \bullet \mathbf{n}$$

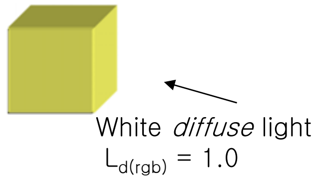
surface normal
direction of light source



Diffuse Reflection

- $R_d \propto \cos\theta = \mathbf{l} \bullet \mathbf{n}$

$k_{dr} = 0.9$
$k_{dg} = 0.9$
$k_{db} = 0.1$

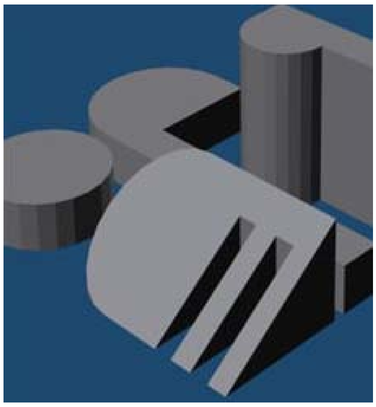


- Let k_d be the fraction of incoming diffuse light that is reflected (reflection coefficient), then

$$I_d = k_d\,(\mathbf{l} \bullet \mathbf{n})\,L_d$$

- Incorporating a distance term d to account for attenuation,

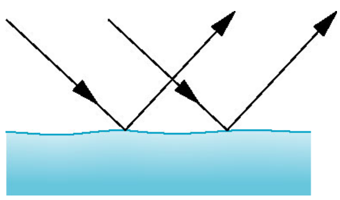
$$I_d = \frac{1}{a + bd + cd^2}\,k_d\,(\mathbf{l} \bullet \mathbf{n})\,L_d$$



Cornell Univ.

Specular Reflection

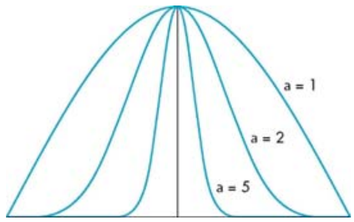
- Whereas a diffuse surface is rough, a specular surface is smooth.
- As the surface gets smoother, the reflected light is concentrated in a smaller range of angles.

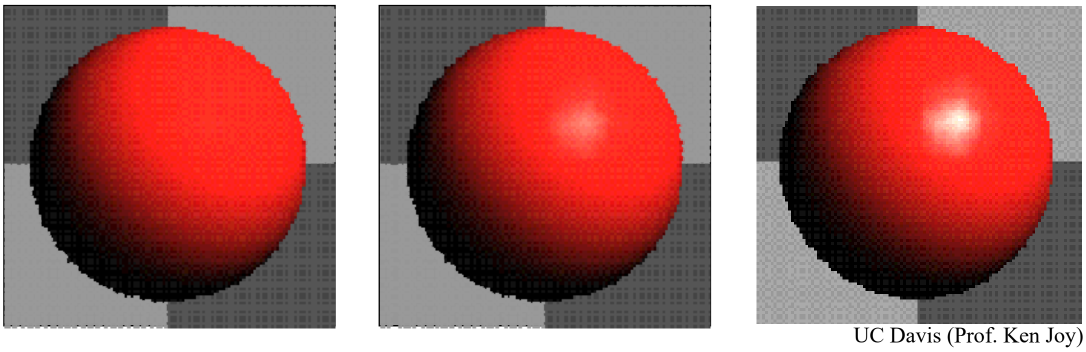


- Phong's model: approximation
  - The amount of light that viewer sees depends on the angle ϕ between the direction of a perfect reflector ($\mathbf{r}$) and the direction of the viewer ($\mathbf{v}$)
  - $$I_s = k_s \cos\phi^{\alpha}\,L_s = k_s\,(\mathbf{r} \bullet \mathbf{v})^{\alpha}\,L_s$$ α : shininess

Specular Reflection

- The shininess coefficient α
  - Infinite: mirror like
  - 100 ~ 500: metalic
  - < 100: broad highlights

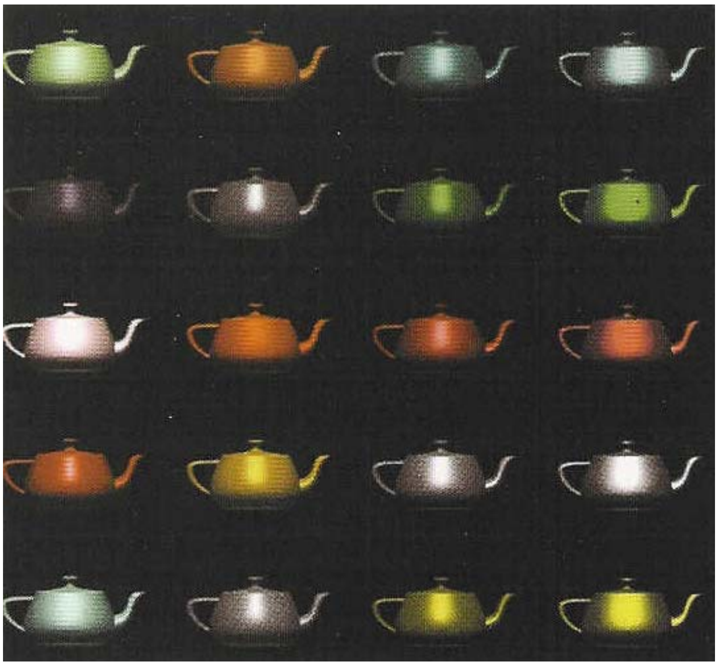




UC Davis (Prof. Ken Joy)

- We can add distance term as we did for diffuse reflection

Phong Reflection Model

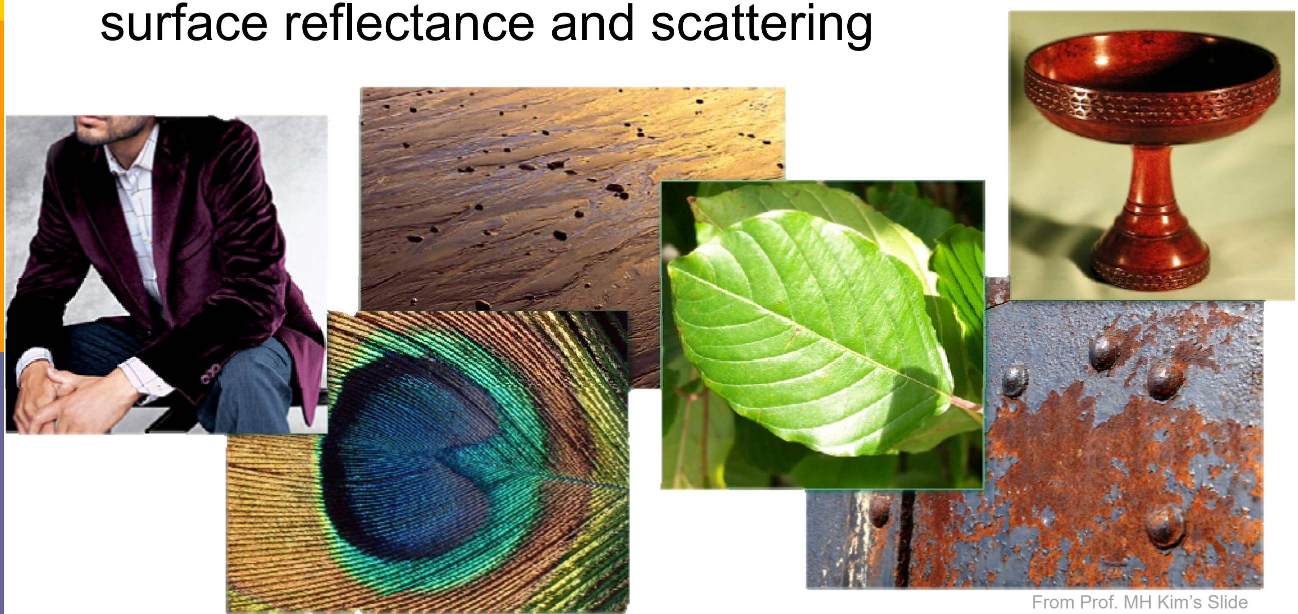


Materials

Chapter 14

Material Appearance

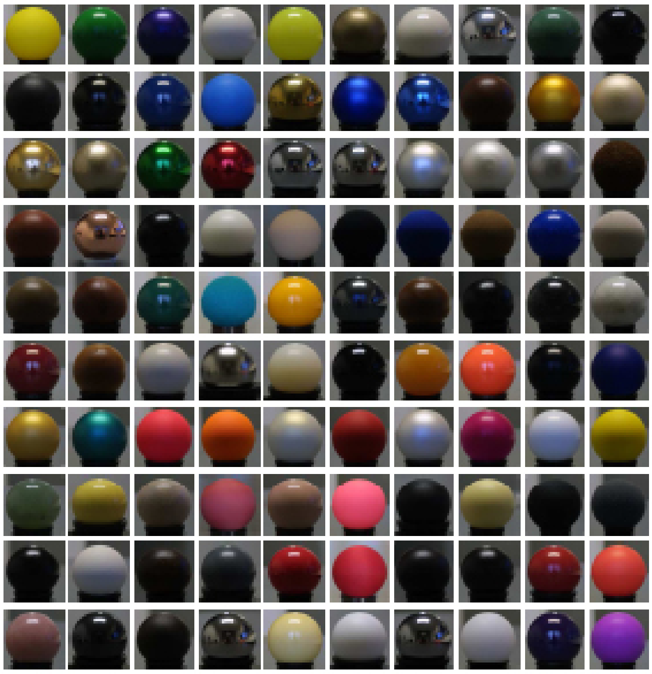
- Rich variety of materials: characterized by surface reflectance and scattering



From Prof. MH Kim's Slide

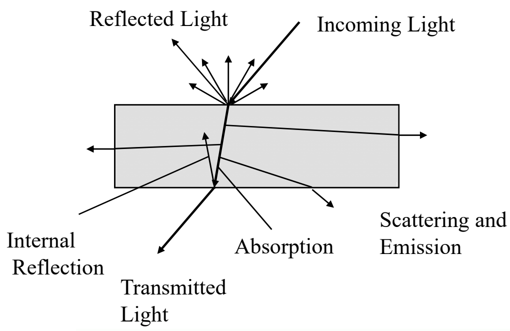
Material Appearance

- Reflection
  - Fig. 21.6
  - BRDF

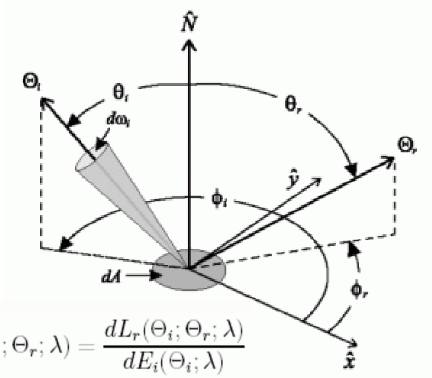


BRDF (Bidirectional Reflectance Distribution Function)

- Light & Matters interaction depends on the physical characteristics of the light as well as the physical composition and characteristics of the matter



- A BRDF describes how much light is reflected when light makes contact with a certain material
- The degree to which light is reflected depends on the viewer and light position relative to the surface normal and tangent
- BRDF is a function of incoming (light) direction and outgoing (view) direction relative to a local orientation at the light interaction point
- BRDFs can be obtained through physical measurement



$$\rho(\Theta_i;\Theta_r;\lambda) = \frac{dL_r(\Theta_i;\Theta_r;\lambda)}{dE_i(\Theta_i;\lambda)}$$

Basic Idea

- When light hits a physical material, it is scattered in various outgoing directions.
- Different kinds of *materials* scatter light in different patterns and this results in different appearances when observed by the eye or a camera.
  - Some materials may appear glossy while others matte.
- By simulating this scattering process, we can give a realistic physical appearance to a 3D rendered object.
  - We will take a pragmatic (practical) approach here instead of a scientific one.

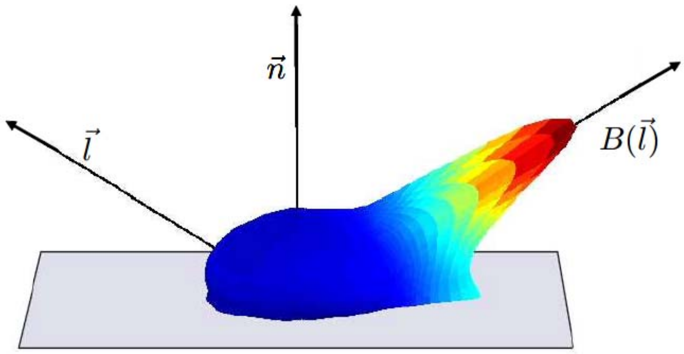
Materials

- We can simulate light bouncing off of a material
- Best done in the ***fragment shader*** if possible.
- Uniform variables are often used to describe thing like the positions of some light sources in the scene, which do not change from vertex to vertex.
- Varying variables are often used to describe the coordinates vector of the point (with respect to, say, the eye frame), a normal for the point.
- Parameters describing the material properties at the point might be uniforms or varying.
- The fragment shader then typically takes this data and simulates how light would bounce off of this material, producing a color in the image.

PVC plastic material

- The figure just describes the result of light that comes in from the specific shown direction $\vec{l}$

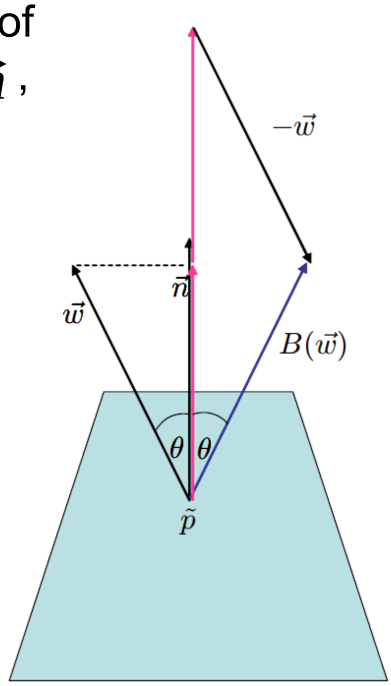
The plastic will appear brightest when observed in the directions clustered about the 'bounce' direction of the light: $B(\vec{l}\,)$

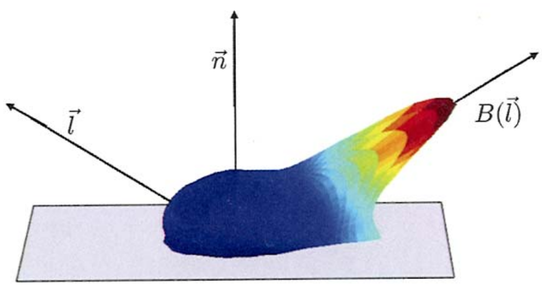


How to compute the reflection vector?

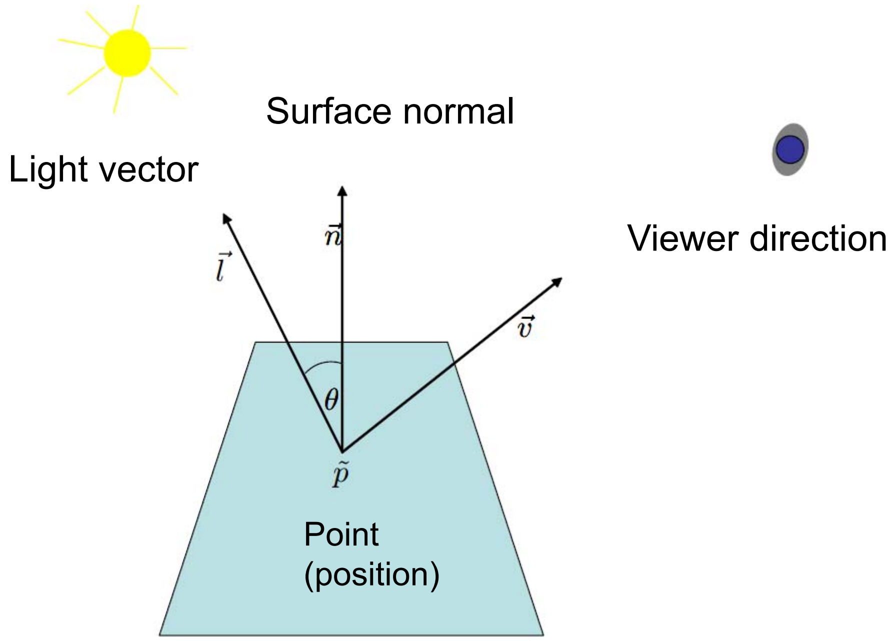
- Given any vector $\vec{w}$ (not necessarily of unit norm) and a unit normal vector $\vec{n}$, we can compute the bounce vector (mirror reflection) of $\vec{w}$ as

$$B(\vec{w}) = 2(\vec{w}\cdot\vec{n})\vec{n} - \vec{w}$$





Basic Setup



Basic setup

- We assume that all light comes from a single point light source that is pointed to by the light vector, $\vec{l}$.
  - In real life, there are some small light sources (sun, candle), there are some big light sources (sky, fluorescent bulbs).
  - Additionally, every surface in the scene reflects light towards the point in question.

Basic setup

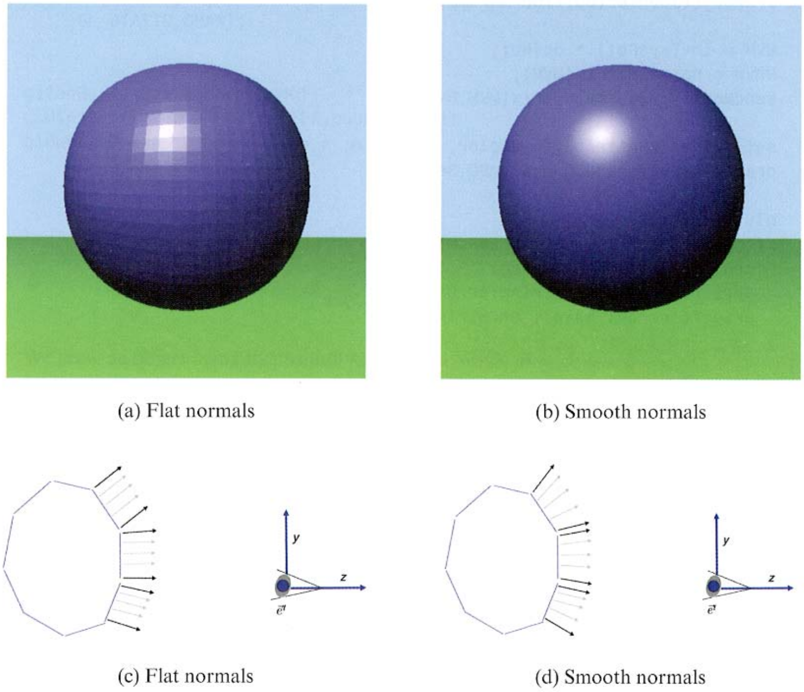
- The angle between the surface normal and the vector pointing towards the light is denoted as θ
- We want to compute the amount of light reflected toward the eye along the view vector, $\vec{v}$
- To handle color, we just compute three reflected quantities for RGB.
- We typically represent the incoming light in RGB (often white).

Basic setup

- And use RGB coefficients for the material property representation.
- And just do 3 independent calculations
  - Note that this use of RGB is not physically accurate.
- These quantities are then directly used as the RGB values for the pixel.
- We will often assume here that all coordinates of points and vectors are written with respect to the eye frame.

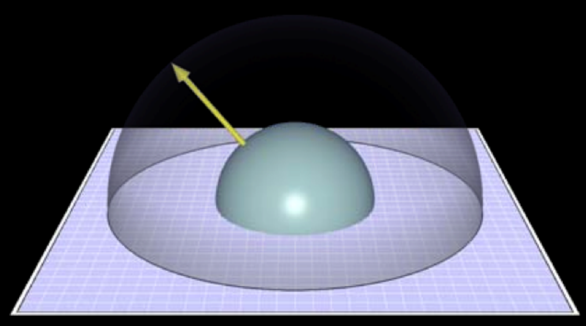
Basic setup

- Note that the normal does not have to be the "true normal" of the flat triangular geometry.

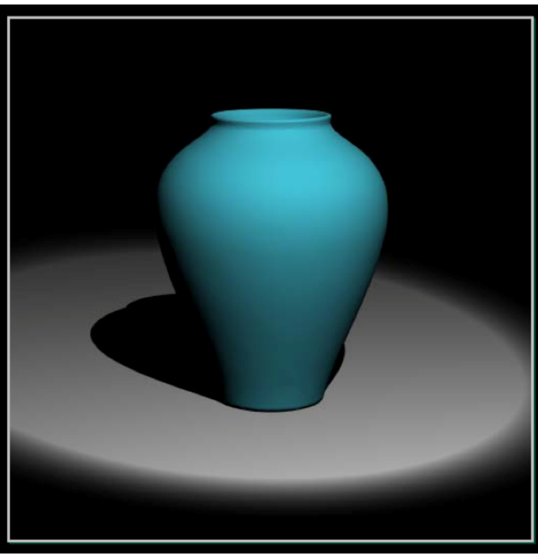


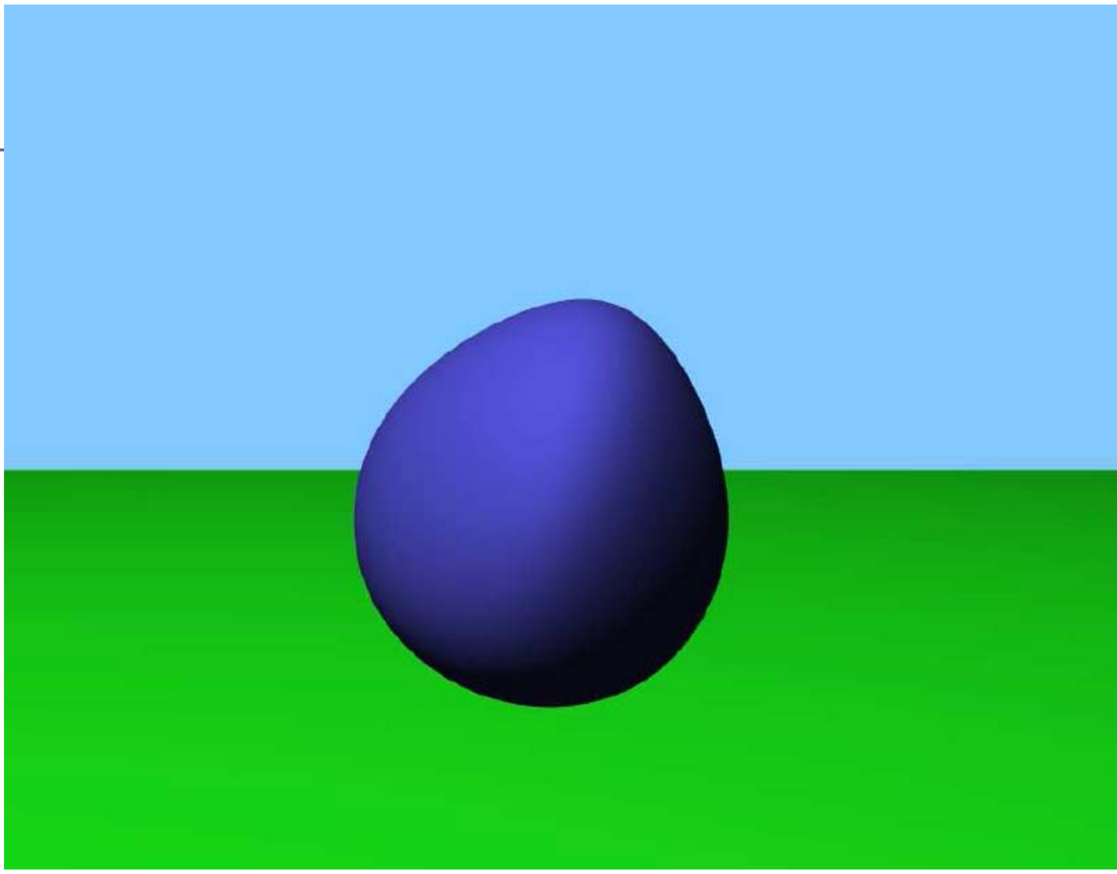
Diffuse reflectance model

- "Diffuse" materials, like rough wood, appear equally bright when observed from all direction



$$f_r = const. = \frac{\rho}{\pi}$$



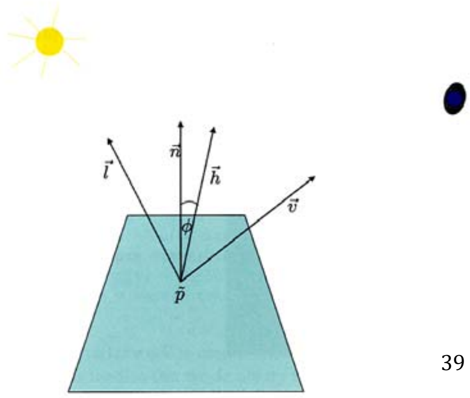


Shiny materials (specularity)

- Many materials, such as plastics, are not diffuse;
- For a fixed incoming light distribution, they appear brighter when observed from some directions and dimmer when observed from others.
- Round objects made up of such materials have bright highlights wherever the surface-normal points “just right”.
- A simple, somewhat plausible computation that is often used in practice, is to simply calculate the light’s bounce vector $B(\vec{l}\,)$, and then compute its angle with $\vec{v}$.

Shiny materials (specularity)

- Halfway vector
  - A simpler way to get the same effect is to compute the halfway vector $\vec{h} = \text{normalize}(\vec{v} + \vec{l})$
  - And then its angle ϕ with $\vec{n}$.
  - The vectors $\vec{h}$ and $\vec{n}$ are well aligned only when $\vec{v}$ and $B(\vec{l})$ are the same.
  - We compute the cosine of using a dot product that falls off as $\vec{h}$ and $\vec{n}$ diverge.



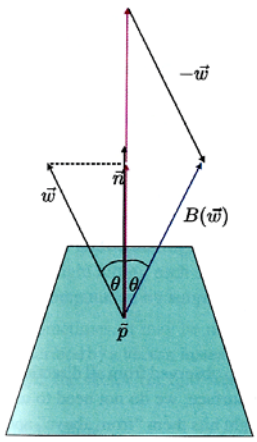
Shiny materials (specularity)

- When the bounce and view vectors are well aligned, we draw a very bright pixel. When they are not well aligned, we draw a dimmer pixel.
  - Done with a dot product.
- To model shiny materials, we want the brightness to fall off very quickly in angle, so we then raise the dot product to a positive power.
- Such specular reflection can be explained by "micro-facet" theory.

Blinn-Phong vs. Phong BRDF

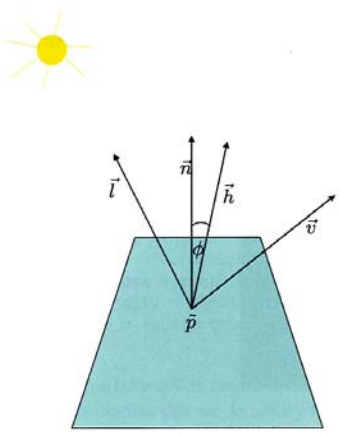
- Phong (1973)

$$B(\vec{l}) = 2(\vec{l} \cdot \vec{n})\vec{n} - \vec{l}$$
$$\cos\theta = B(\vec{l}) \cdot \vec{v}$$



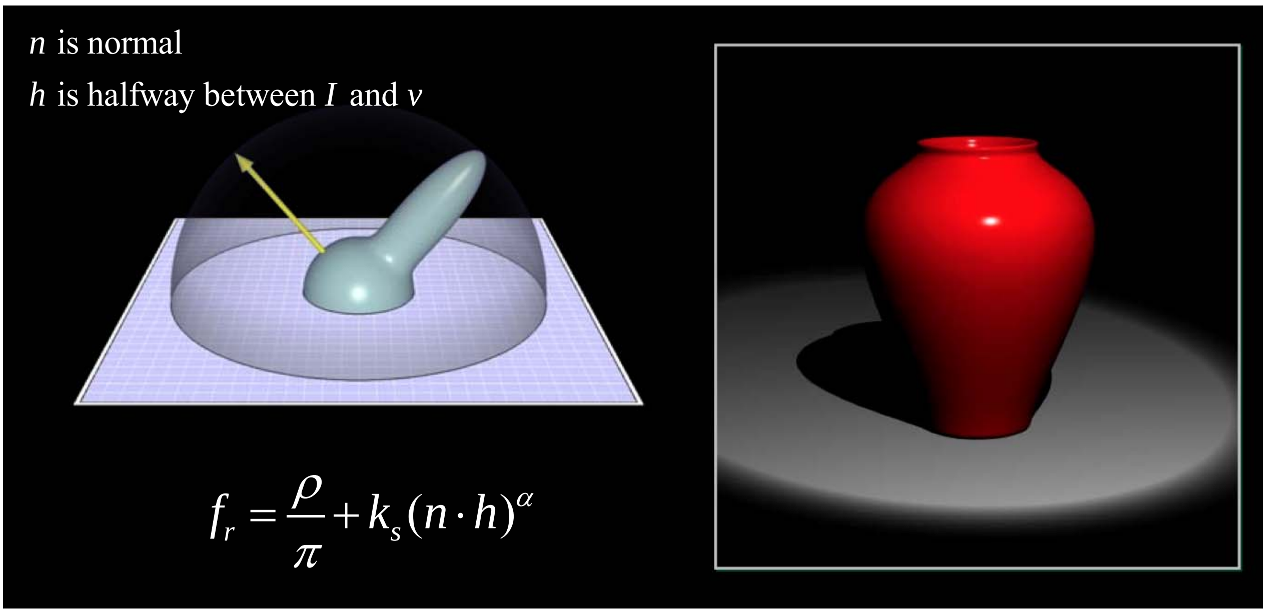
- Blinn-Phong (1977)

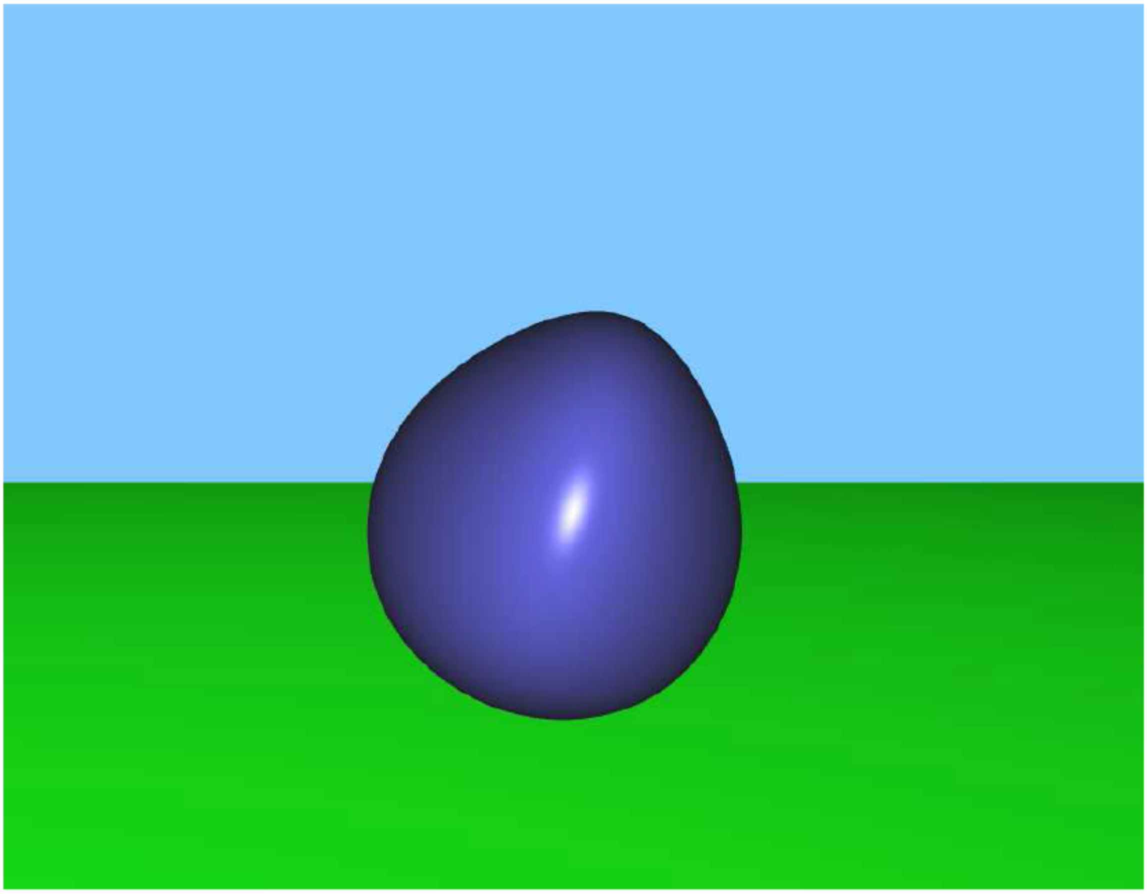
$$\vec{h} = (\vec{l} + \vec{v}) / |\vec{l} + \vec{v}|$$
$$\cos\theta = \vec{h} \cdot \vec{v}$$



Specular reflectance model

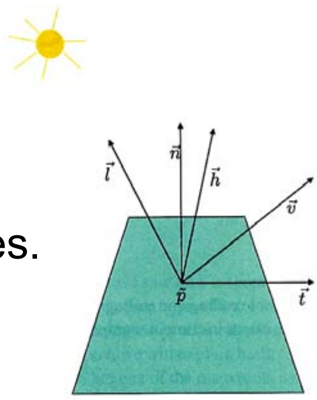
- 'Shiny'

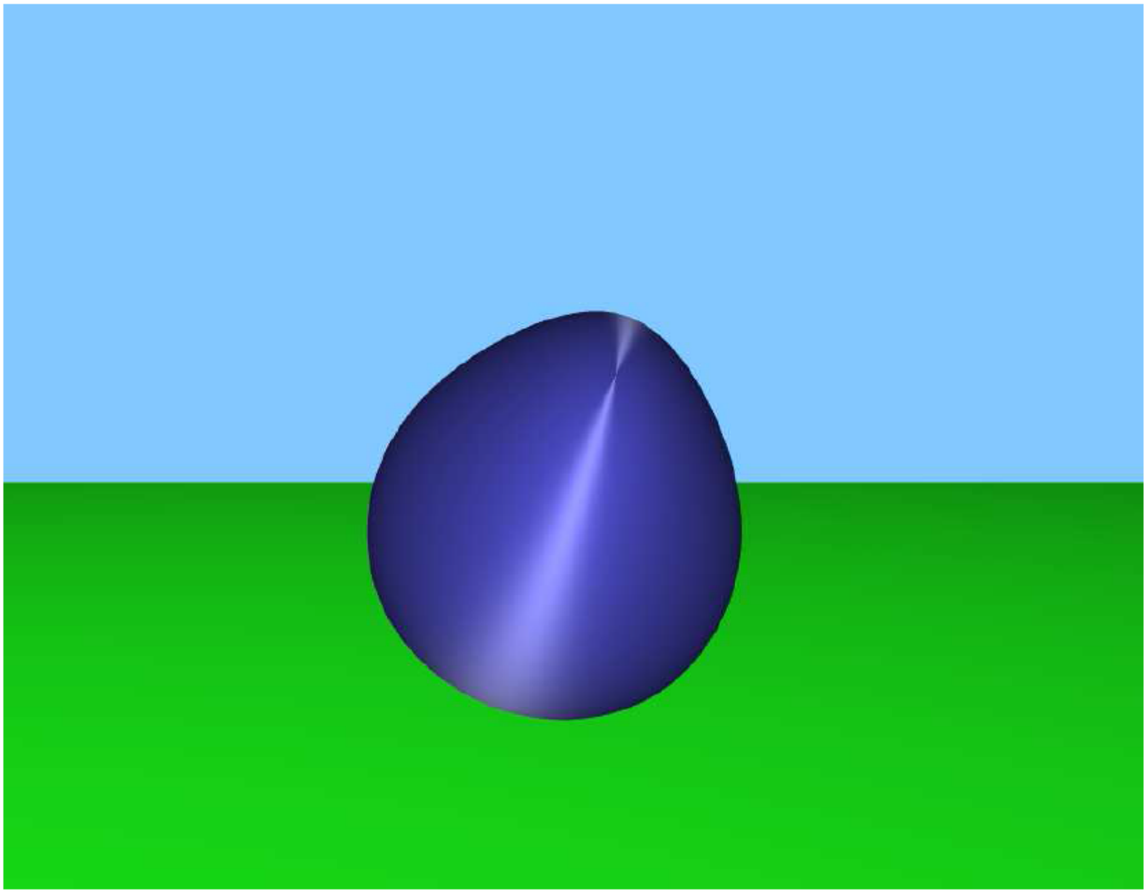




Anisotropy

- The two previous material models, as well as many others, have the property of isotropy.
- This means that there is no preferred “grain” to the surface.
- Some materials, like brushed metal, behave in an anisotropic manner.
  - If one takes a flat sample of such a material and spins it around its normal, the material’s appearance changes.





Summary

- Light – Material Interaction
- Reflectance model
- Ambient, Diffuse, Specular

- We'll learn how to put together in 'shaders'

- **Exercise Tomorrow 7 PM (LAB) !**