1

CS-184: Computer Graphics

Lecture #3: Shading

Prof. James O'Brien
University of California, Berkeley

2

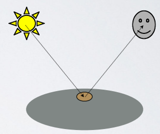
Today

- Local Illumination & Shading
  - The BRDF
  - Simple diffuse and specular approximations
  - Shading interpolation: flat, Gouraud, Phong
  - Some miscellaneous tricks

3

Local Shading

- Local: consider in isolation
  - 1 light
  - 1 surface
  - The viewer
- Recall: lighting is linear
  - Almost always...



Counter example: photochromatic materials

4

Local Shading

- Examples of non-local phenomena
  - Shadows
  - Reflections
  - Refraction
  - Indirect lighting

5

The BRDF

- The **B**i-directional **R**eflectance **D**istribution **F**unction
- Given
  - Surface material
  - Incoming light direction
  - Direction of viewer
  - Orientation of surface
- Return:
  - fraction of light that reaches the viewer
- We'll worry about physical units later...

$$\rho = \rho(\theta_V, \theta_L)$$
$$= \rho(\mathbf{v}, \mathbf{l}, \mathbf{n})$$

5

6

The BRDF

Ideal specular

- **Perfect mirror reflection**

Ideal diffuse

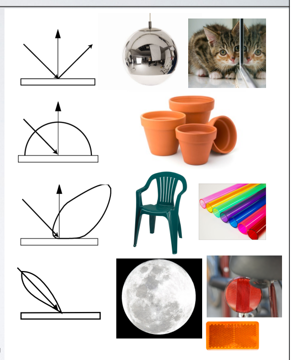
- **Equal reflection in all directions**

Glossy specular

- **Majority of light reflected near mirror direction**

Retro-reflective

- **Light reflected back towards light source**



Diagrams illustrate how light from incoming direction is reflected in various outgoing directions.

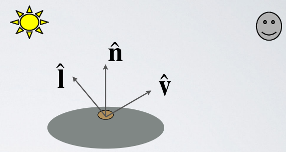
Ren Ng

6

The BRDF

7

$\rho(\mathbf{v}, \mathbf{l}, \mathbf{n})$



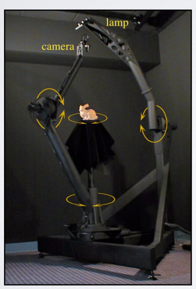
- Spatial variation capture by "the material"
- Frequency dependent
  - Typically use separate RGB functions
  - Does not work perfectly
  - Better:

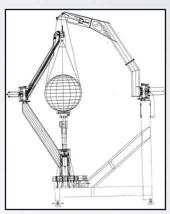
$$\rho = \rho(\theta_V, \theta_L, \lambda_{\text{in}}, \lambda_{\text{out}})$$

Obtaining BRDFs

8

- Measure from real materials





Images from Marc Levoy

9

Obtaining BRDFs

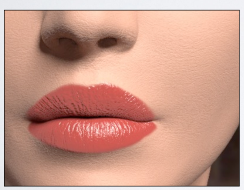
- Measure from real materials
- Computer simulation
  - Simple model + complex geometry
- Derive model by analysis
- Make something up

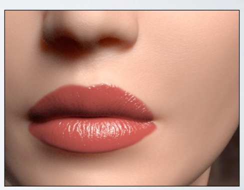
9

10

Beyond BRDFs

- The BRDF model does not capture everything
  - e.g. Subsurface scattering (BSSRDF)





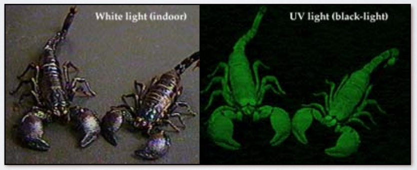
Images from Jensen *et. al, SIGGRAPH 2001*

10

11

Beyond BRDFs

- The BRDF model does not capture everything
  - e.g. Inter-frequency interactions

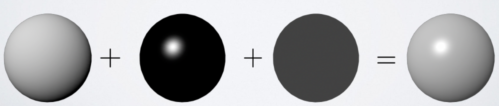


$\rho = \rho(\theta_V, \theta_L, \lambda_{in}, \lambda_{out})$ This version would work....

12

A Simple Model

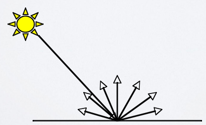
- Approximate BRDF as sum of
  - A diffuse component
  - A specular component
  - A "ambient" term

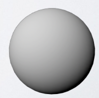


Diffuse Component

13

- Lambert's Law
  - Intensity of reflected light proportional to cosine of angle between surface and incoming light direction
  - Applies to "diffuse," "Lambertian," or "matte" surfaces
  - Independent of viewing angle
- Use as a component of non-Lambertian surfaces

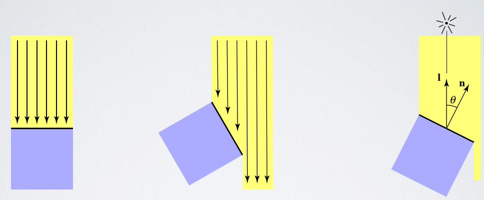




13

Diffuse Component

14



Top face of cube receives a certain amount of light

Top face of 60° rotated cube intercepts half the light

In general, light per unit area is proportional to $\cos\theta = \mathbf{l} \cdot \mathbf{n}$

Steve Marschner

14

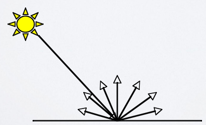
15

Diffuse Component

Comment about two-side lighting in text is wrong...

$$k_d I(\hat{\mathbf{l}} \cdot \hat{\mathbf{n}})$$

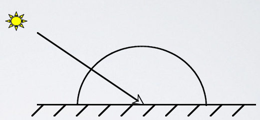
$$\max(k_d I(\hat{\mathbf{l}} \cdot \hat{\mathbf{n}}), 0)$$



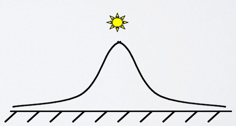
16

Diffuse Component

- Plot light leaving in a given direction:



- Plot light leaving from each point on surface



17

Specular Component

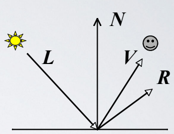
- Specular component is a mirror-like reflection
- Phong Illumination Model
  - A reasonable approximation for some surfaces
  - Fairly cheap to compute
- Depends on view direction

18

Specular Component

$$k_s I (\hat{\mathbf{r}} \cdot \hat{\mathbf{v}})^p$$

$$k_s I \max(\hat{\mathbf{r}} \cdot \hat{\mathbf{v}}, 0)^p$$

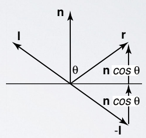


19

Specular Component

- Computing the reflected direction

$$\hat{\mathbf{r}} = -\hat{\mathbf{l}} + 2(\hat{\mathbf{l}} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$$

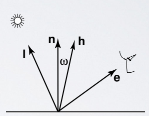


20

Specular Component

- "Half-angle" approximation for specular

$$\hat{\mathbf{h}} = \frac{\hat{\mathbf{l}} + \hat{\mathbf{v}}}{||\hat{\mathbf{l}} + \hat{\mathbf{v}}||}$$



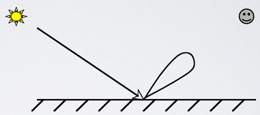
different specular term $k_s I(\hat{\mathbf{h}} \cdot \hat{\mathbf{n}})^p$

Don't use half-angle approximation in your assignments!

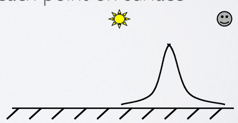
21

Specular Component

- Plot light leaving in a given direction:



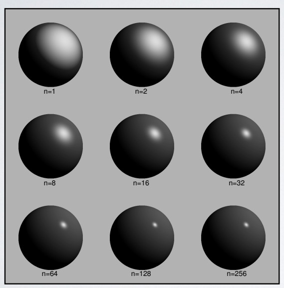
- Plot light leaving from each point on surface

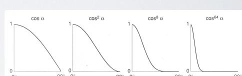


22

Specular Component

- Specular exponent sometimes called "roughness"

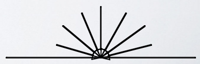




23

Ambient Term

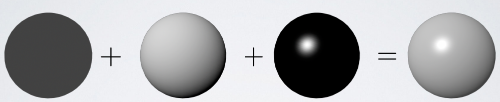
- Really, its a cheap hack
- Accounts for "ambient, omnidirectional light"
- Without it everything looks like it's in space



24

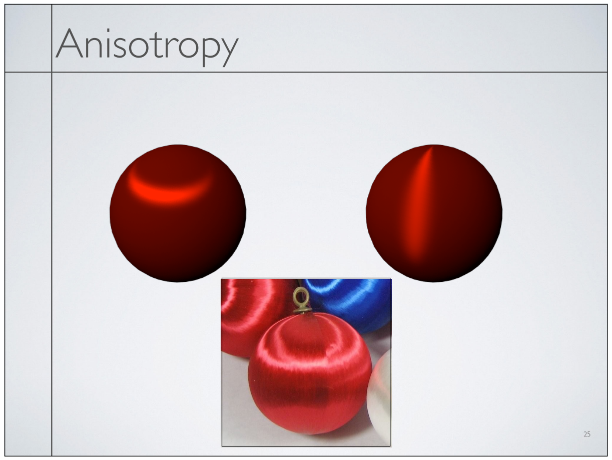
Summing the Parts

$$R = k_a I + k_d I \max(\hat{\mathbf{l}} \cdot \hat{\mathbf{n}}, 0) + k_s I \max(\hat{\mathbf{r}} \cdot \hat{\mathbf{v}}, 0)^p$$

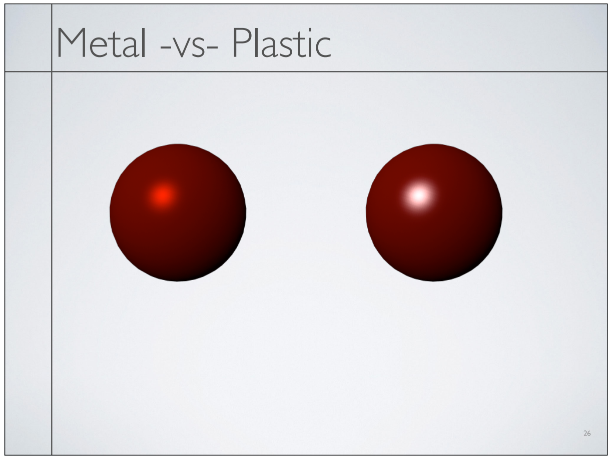


- Recall that the $k_?$ are by wavelength
  - RGB in practice
- Sum over all lights

25

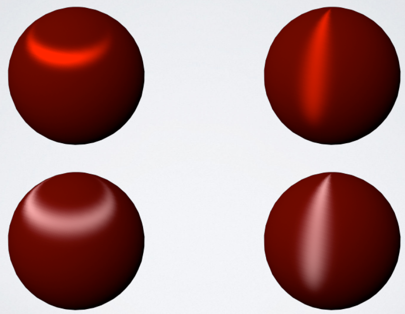


26



27

Metal -vs- Plastic



28

Other Color Effects

29

30

Materials: Paint

31

Materials: Paint

32

33

Materials: Mirror

Ren Ng

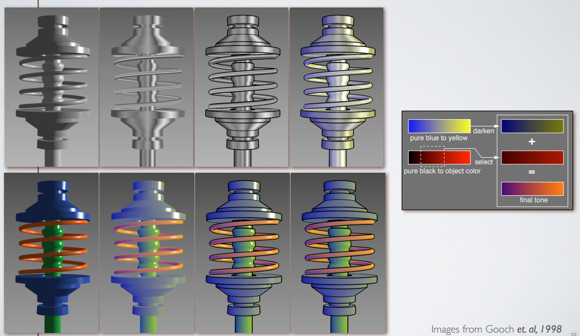
34

Materials: Metallic

Ren Ng

Other Color Effects

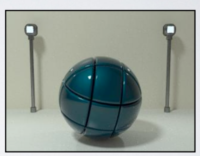
35



Images from Gooch *et. al*, 1998

Measured BRDFs

36



BRDFs for automotive paint

Images from Cornell University Program of Computer Graphics

37

Measured BRDFs


BRDFs for aerosol spray paint

Images from Cornell University Program of Computer Graphics 37

38

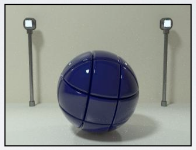
Measured BRDFs

BRDFs for house paint

Images from Cornell University Program of Computer Graphics 38

39

Measured BRDFs



BRDFs for lucite sheet

Images from Cornell University Program of Computer Graphics

39

40

Ashikhmin-Shirley BRDF

- More realistic specular term (for some materials)
- Anisotropic specularities
- Fresnel behavior (grazing angle highlights)
- Energy preserving diffuse term
- Sum of diffuse and specular terms (as before)

$$\rho(\hat{\mathbf{l}}, \hat{\mathbf{v}}) = \rho_d(\hat{\mathbf{l}}, \hat{\mathbf{v}}) + \rho_s(\hat{\mathbf{l}}, \hat{\mathbf{v}})$$

Michael Ashikhmin and Peter Shirley. 2000. An anisotropic phong BRDF model. J. Graph. Tools 5, 2 (February 2000), 25-32.
https://www.cs.utah.edu/~shirley/papers/jgtbrdf.pdf

40

41

Ashikhmin-Shirley BRDF

$$\rho_s(\hat{\mathbf{l}}, \hat{\mathbf{e}}) = \frac{\sqrt{(p_u+1)(p_v+1)}}{8\pi} \frac{(\hat{\mathbf{n}} \cdot \hat{\mathbf{h}})^{p_u \cos^2\phi + p_v \sin^2\phi}}{(\hat{\mathbf{h}} \cdot \hat{\mathbf{e}}) \max\left((\hat{\mathbf{n}} \cdot \hat{\mathbf{e}}), (\hat{\mathbf{n}} \cdot \hat{\mathbf{l}})\right)} F(\hat{\mathbf{h}} \cdot \hat{\mathbf{e}})$$

$$F(\hat{\mathbf{h}} \cdot \hat{\mathbf{e}}) = K_s + (1 - K_s)(1 - (\hat{\mathbf{h}} \cdot \hat{\mathbf{e}}))^5$$

- $\hat{\mathbf{l}}$ Light direction
- $\hat{\mathbf{e}}$ Viewer (eye) direction
- p_u, p_v Specular powers
- $\hat{\mathbf{n}}$ Normal
- $\hat{\mathbf{h}}$ Half angle
- K_s Specular coefficient (color)
- $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ Parametric directions

41

42

Ashikhmin-Shirley BRDF

$$\rho_s(\hat{\mathbf{l}}, \hat{\mathbf{e}}) = \frac{\sqrt{(p_u+1)(p_v+1)}}{8\pi} \frac{(\hat{\mathbf{n}} \cdot \hat{\mathbf{h}})^{\frac{p_u(\hat{\mathbf{h}} \cdot \hat{\mathbf{u}})^2 + p_u(\hat{\mathbf{h}} \cdot \hat{\mathbf{v}})^2}{1 - (\hat{\mathbf{h}} \cdot \hat{\mathbf{n}})^2}}}{(\hat{\mathbf{h}} \cdot \hat{\mathbf{e}}) \max\left((\hat{\mathbf{n}} \cdot \hat{\mathbf{e}}), (\hat{\mathbf{n}} \cdot \hat{\mathbf{l}})\right)} F(\hat{\mathbf{h}} \cdot \hat{\mathbf{e}})$$

$$F(\hat{\mathbf{h}} \cdot \hat{\mathbf{e}}) = K_s + (1 - K_s)(1 - (\hat{\mathbf{h}} \cdot \hat{\mathbf{e}}))^5$$

Approximate Fresnel function

- $\hat{\mathbf{l}}$ Light direction
- $\hat{\mathbf{e}}$ Viewer (eye) direction
- p_u, p_v Specular powers
- $\hat{\mathbf{n}}$ Normal
- $\hat{\mathbf{h}}$ Half angle
- K_s Specular coefficient (color)
- $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ Parametric directions

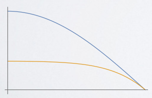
42

Ashikhmin-Shirley BRDF

43

$$\rho_d(\hat{\mathbf{l}},\hat{\mathbf{e}}) = \frac{28K_d}{23\pi}(1-K_s)\left(1-\left(1-\frac{\hat{\mathbf{n}}\cdot\hat{\mathbf{e}}}{2}\right)^5\right)\left(1-\left(1-\frac{\hat{\mathbf{n}}\cdot\hat{\mathbf{l}}}{2}\right)^5\right)$$

Note: The Phong diffuse term (Lambertian) is independent of view. But this term accounts for unavailable light due to specular/Fresnel reflection.

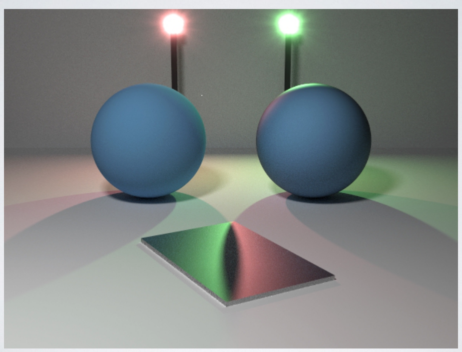


- $\hat{\mathbf{l}}$ Light direction
- $\hat{\mathbf{e}}$ Viewer (eye) direction
- p_u, p_v Specular powers
- $\hat{\mathbf{n}}$ Normal
- $\hat{\mathbf{h}}$ Half angle
- K_s Specular coefficient (color)
- $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ Parametric directions

43

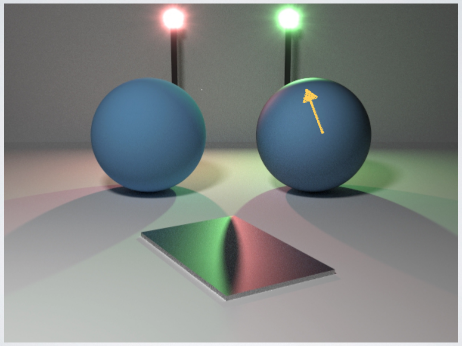
Ashikhmin-Shirley BRDF

44



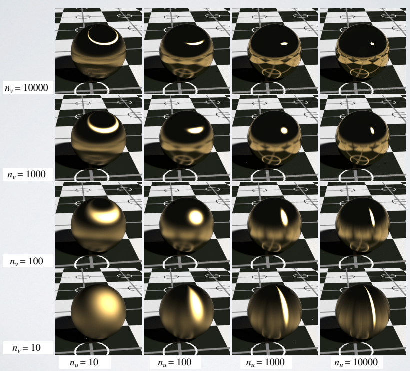
44

Ashikhmin-Shirley BRDF



45

Ashikhmin-Shirley BRDF



46

Details Beget Realism

47

- The "computer generated" look is often due to a lack of fine/subtle details... a lack of richness.

From bustledress.com

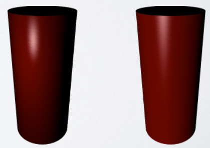
Details Beget Realism

48

49

Direction -vs- Point Lights

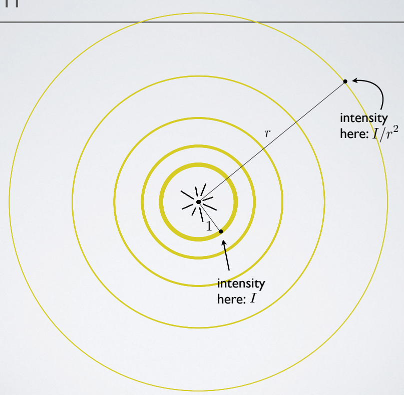
- For a point light, the light direction changes over the surface
- For "distant" light, the direction is constant
- Similar for orthographic/perspective viewer



49

50

Falloff



50

51

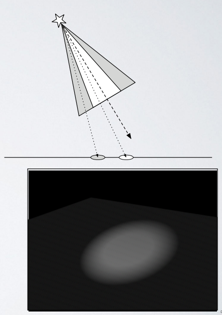
Falloff

- Physically correct: $1/r^2$ light intensify falloff
  - Tends to look bad (why?)
  - Not used in practice
- Sometimes compromise of $1/r$ used

52

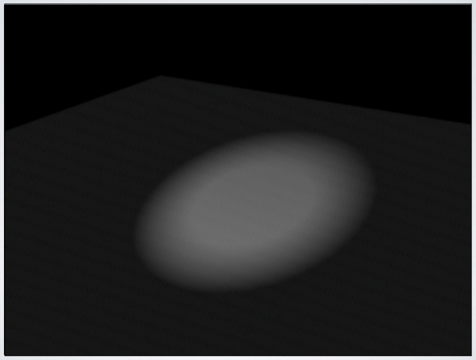
Spot and Other Lights

- Other calculations for useful effects
  - Spot light
  - Only light certain objects
  - Negative lights
  - *etc.*



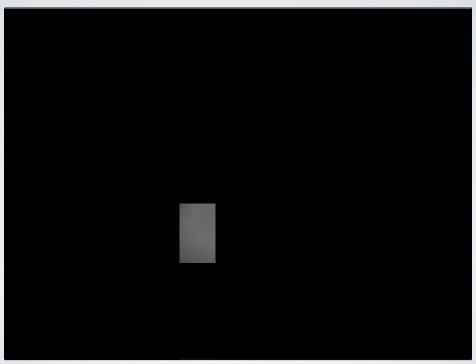
53

Ugly....



54

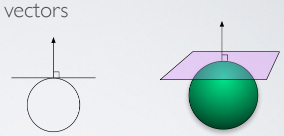
Ugly....



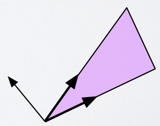
55

Surface Normals

- The normal vector at a point on a surface is perpendicular to all surface tangent vectors



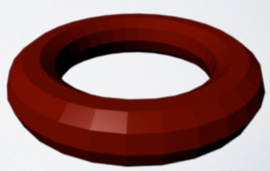
- For triangles normal given by right-handed cross product



56

Flat Shading

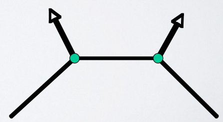
- Use constant normal for each triangle (polygon)
  - Polygon objects don't look smooth
  - Faceted appearance very noticeable, especially at specular highlights
  - Recall mach bands...



57

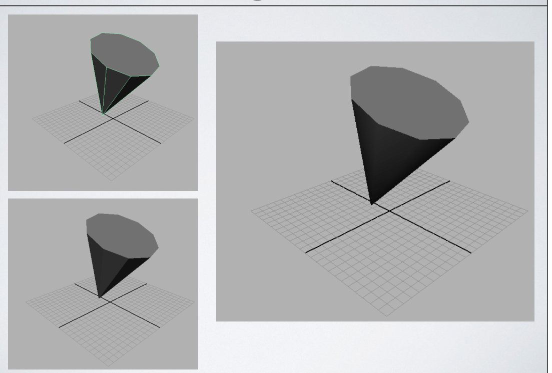
Smooth Shading

- Compute "average" normal at vertices
- Interpolate across polygons
- Use threshold for "sharp" edges
  - Vertex may have different normals for each face



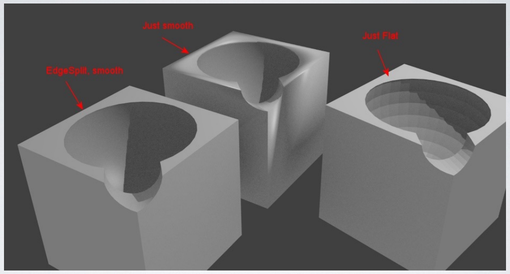
58

Smooth Shading



Smooth Shading

59

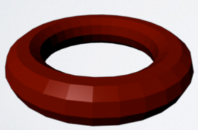


From blender.stackexchange.com

Gouraud Shading

60

- Compute shading at each vertex
  - Interpolate colors from vertices
  - Pros: fast and easy, looks smooth
  - Cons: terrible for specular reflections



Flat

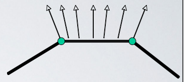
Gouraud

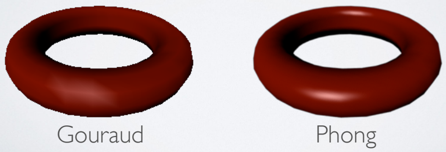
Note: Gouraud was hardware rendered...

61

Phong Shading

- Compute shading at each pixel
  - Interpolate ***normals*** from vertices
  - Pros: looks smooth, better speculars
  - Cons: expensive





Note: Gouraud was hardware rendered...