# CS-184: Computer Graphics

## Lecture #3: Shading

Prof. James O'Brien
University of California, Berkeley

V2014-S-03-1.0

## Announcements

- Assignment 0: due this Friday
- Homework 1: due this Thursday
- **Assignment 1: due February 14th**

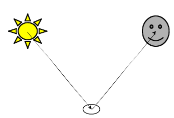
# Today

- 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

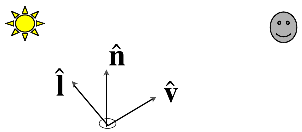
- The **B**i-directional **R**eflectance **D**istribution **F**unction
- Given
  - Surface material
  - Incoming light direction
  - Direction of viewer
  - Orientation of surface
- Return:
  - fraction of light that reaches the viewer
- We'll worry about physical units later...

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

6

# The BRDF

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



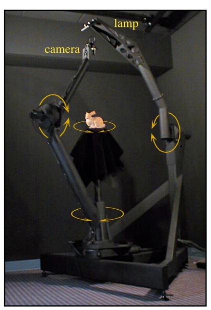
- Spatial variation capture by "the material"
- Frequency dependent
  - Typically use separate RGB functions
  - Does not work perfectly
  - Better:

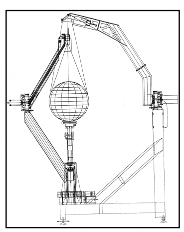
$$\rho = \rho(\theta_V, \theta_L, \lambda_{\text{in}}, \lambda_{\text{out}})$$

7

# Obtaining BRDFs

- Measure from real materials



Images from Marc Levoy

8

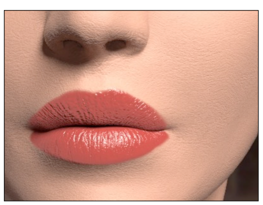
## Obtaining BRDFs

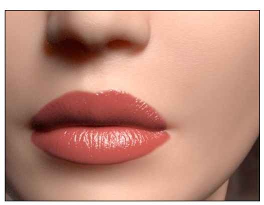
- Measure from real materials
- Computer simulation
  - Simple model + complex geometry
- Derive model by analysis
- Make something up

9

## Beyond BRDFs

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



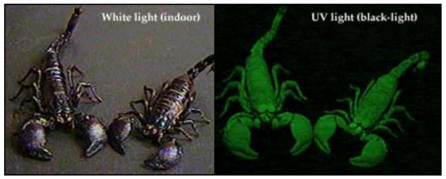


Images from Jensen *et. al, SIGGRAPH 2001*

10

# Beyond BRDFs

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

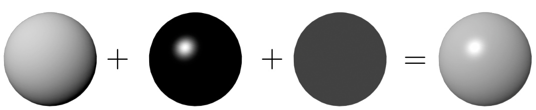



$\rho = \rho(\theta_V, \theta_L, \lambda_{in}, \lambda_{out})$ This version would work....



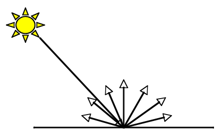
# A Simple Model

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



# Diffuse Component

- 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



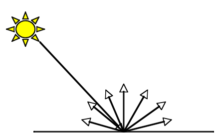
13

# Diffuse Component

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

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

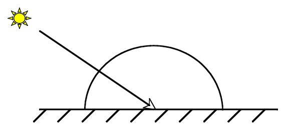
$$\max(k_d I(\hat{\mathbf{l}} \cdot \hat{\mathbf{n}}), 0)$$



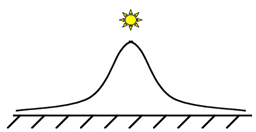
14

## Diffuse Component

- Plot light leaving in a given direction:



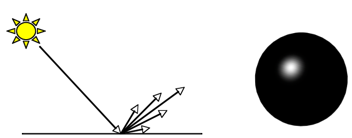
- Plot light leaving from each point on surface



15

## 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

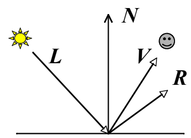


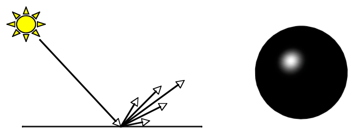
16

## Specular Component

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

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



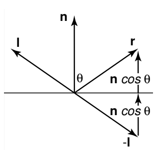



17

## Specular Component

- Computing the reflected direction

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

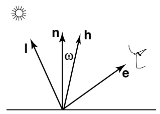



18

# Specular Component

- "Half-angle" approximation for specular

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



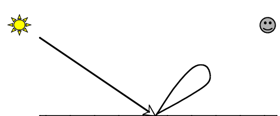

different specular term $k_s I(\hat{\mathbf{h}} \cdot \hat{\mathbf{n}})^p$

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

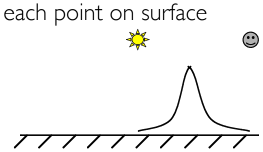


# Specular Component

- Plot light leaving in a given direction:

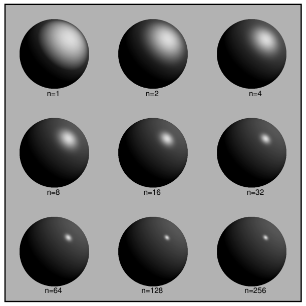


- Plot light leaving from each point on surface



## Specular Component

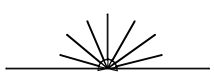
- Specular exponent sometimes called "roughness"




21

## Ambient Term

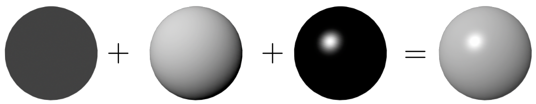
- Really, its a cheap hack
- Accounts for "ambient, omnidirectional light"
- Without it everything looks like it's in space



22

## Summing the Parts

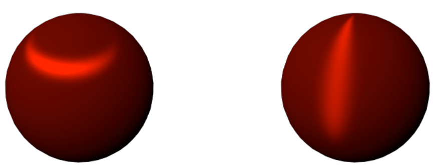
$$R = k_a I + k_d I \max(\hat{\mathbf{l}} \cdot \hat{\mathbf{n}}, 0) + k_s I \max(\hat{\mathbf{r}} \cdot \hat{\mathbf{v}}, 0)^p$$



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

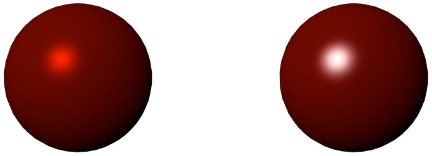
23

## Anisotropy



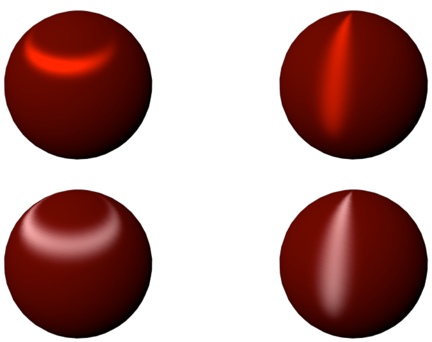
24

# Metal -vs- Plastic





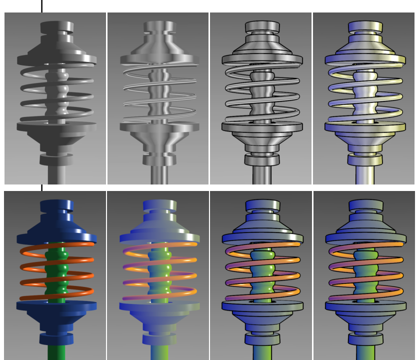
# Metal -vs- Plastic

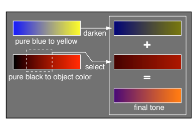


# Other Color Effects

27

# Other Color Effects

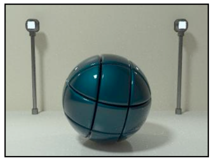





Images from Gooch *et. al, 1998*

28

# Measured BRDFs



BRDFs for automotive paint

Images from Cornell University Program of Computer Graphics

29

# Measured BRDFs

BRDFs for aerosol spray paint

Images from Cornell University Program of Computer Graphics

30

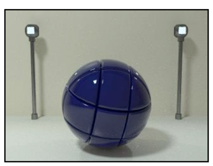
## Measured BRDFs

BRDFs for house paint

Images from Cornell University Program of Computer Graphics



## Measured BRDFs



BRDFs for lucite sheet

Images from Cornell University Program of Computer Graphics

# Details Beget Realism

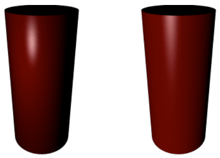
- The "computer generated" look is often due to a lack of fine/subtle details... a lack of richness.

From bustledress.com

33

# 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



34

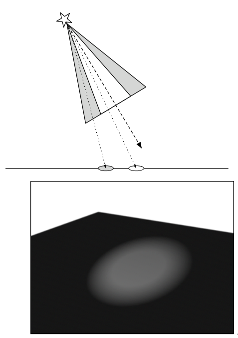
## Falloff

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

35

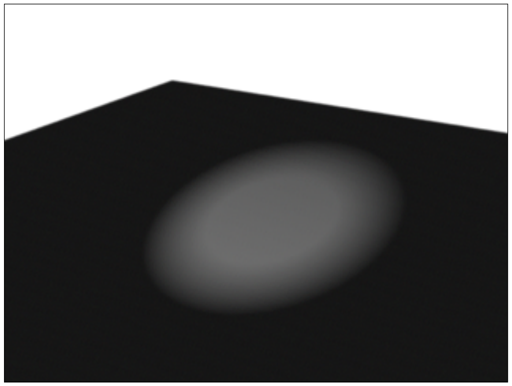
## Spot and Other Lights

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



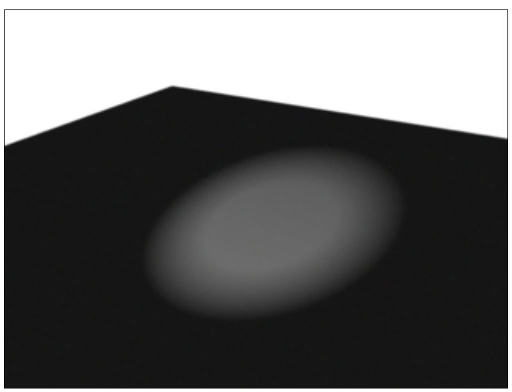
36

## Ugly....



37

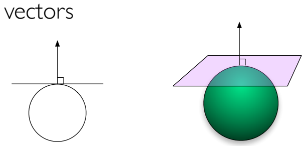
## Ugly....



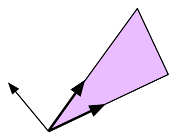
38

# Surface Normals

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



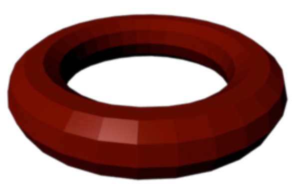
- For triangles normal given by right-handed cross product



39

# Flat Shading

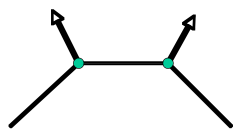
- Use constant normal for each triangle (polygon)
  - Polygon objects don't look smooth
  - Faceted appearance very noticeable, especially at specular highlights
  - Recall mach bands...



40

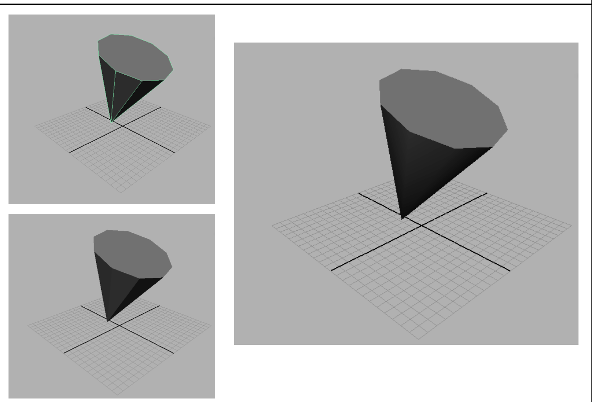
# Smooth Shading

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



41

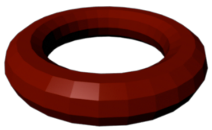
# Smooth Shading



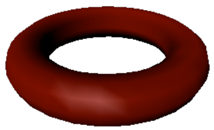
42

# Gouraud Shading

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



Flat



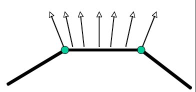
Gouraud

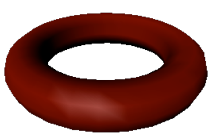
43

Note: Gouraud was hardware rendered...

# Phong Shading

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





Gouraud

Phong

44

Note: Gouraud was hardware rendered...