Week 3

Amy Gooch
CS395: Intro to Animation

Agenda

- Quiz #1
- Critique & review of Project1
- Lecture on Shading & Texturing
- Looking forward to next assignment
  - Bring to class material samples (images or objects)

Quiz #1

Critique and Review of Project 1

Project 2

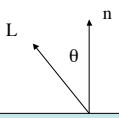
Project 2 Group Assignments

Group1	Che Yusoff, Asrif	Johnson, Rejaie	Schatz, Matthew
Group2	Chin, Ying (YZ)	Krueger, DeBorah	Meor Hamzah, Nurul
Group3	Ku Abdul Rahman, Nizar	Miller, Josh	Pylypczak, Jaroslav
Group4	Md Ishak, Nizam	Nesbitt, Kiel	Teng, Xian Yi
Group5	Edwards, Tennile	Simpson, Alan	

Shading

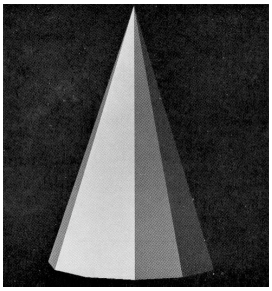
Lambert's law

Light a point in any direction varies as the cosine of the angle between a vector from the point to the light source and the normal vector of the surface at the point



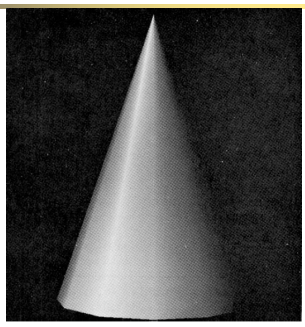
Warnock (Flat) Shading

- Flat shading
- Decrease intensity with distance from light and object
- Highlights



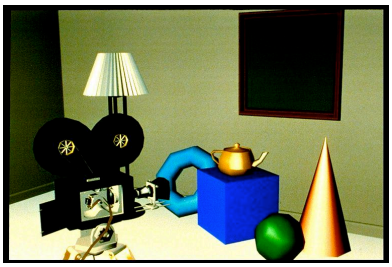
Gouraud Shading

- Compute shading at each vertex
- Interpolate shading



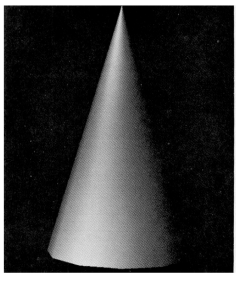
Problem with Gouraud Shading

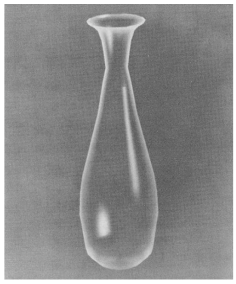
- Highlights across polygons



Phong Shading

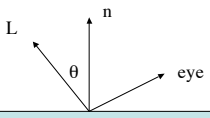
Phong Shading





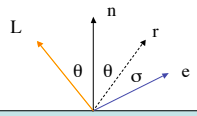
Diffuse Shading

$$I_{diffuse} = k_d I_{light} \cos \theta$$



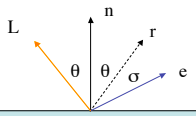
Specular Shading

Add specular by looking at reflection, **r**
Shiny surfaces, such as a mirror

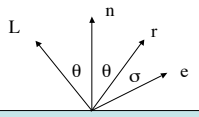


Phong Shading

$$I_{total} = k_a I_{ambient} + \sum_{i=1}^{lights} I_i \left(k_d (N \cdot L) + k_s (V \cdot R)^{n_{shiney}} \right)$$

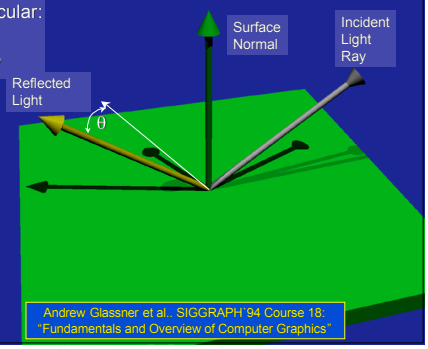


Phong Shading



Review: Surface Properties

Perfectly Specular:
"Mirror"
"infinite gloss"



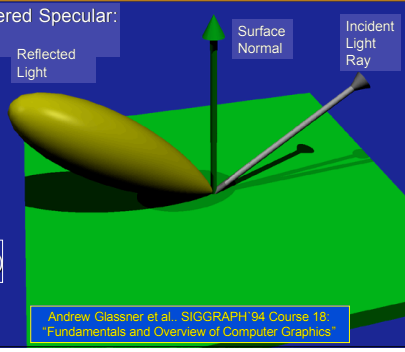
Phong Specular Model:

$L\ R\ \cos^{\infty}(\theta)$

Andrew Glassner et al., SIGGRAPH'94 Course 18: "Fundamentals and Overview of Computer Graphics"

Review: Surface Properties

Slightly scattered Specular:
"high gloss"



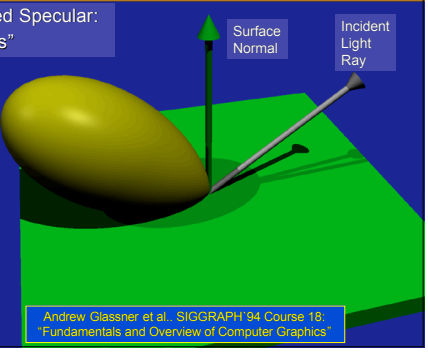
Phong Specular Model:

$L\ R\ \cos^{15}(\theta)$

Andrew Glassner et al., SIGGRAPH'94 Course 18: "Fundamentals and Overview of Computer Graphics"

Review: Surface Properties

More Scattered Specular:
"medium gloss"



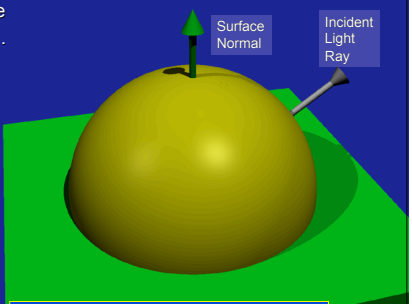
Phong Specular Model:

$L\ R\ \cos^{5}(\theta)$

Andrew Glassner et al., SIGGRAPH'94 Course 18: "Fundamentals and Overview of Computer Graphics"

Review: Surface Properties

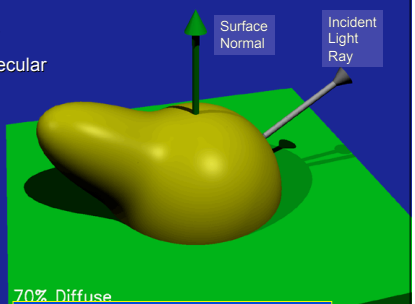
Perfectly Diffuse
"flat", "chalky",…



Andrew Glassner et al., SIGGRAPH'94 Course 18: "Fundamentals and Overview of Computer Graphics"

Review: Surface Properties

Most Materials:
Combination of
Diffuse and Specular



70% Diffuse

Andrew Glassner et al., SIGGRAPH'94 Course 18: "Fundamentals and Overview of Computer Graphics"

OpenGL Lighting Equation

vertex color =

$$\text{emission}_{material} +$$

$$\text{ambient}_{light\ model} * \text{ambient}_{material} +$$

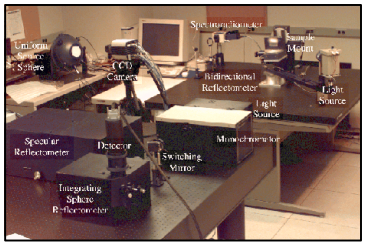
$$\sum_{i=0}^{n-1} (1/(k_c + k_l * d + k_q d^2)) * (\text{spotlight effect})_i *$$

$$[\ \text{ambient}_{light} * \text{ambient}_{material} +$$

$$(\max\{ L \cdot n, 0\}) * \text{diffuse}_{light} * \text{diffuse}_{material} +$$

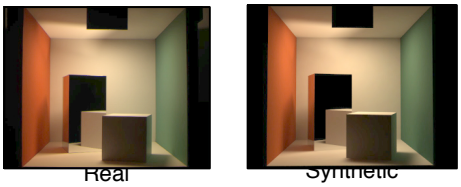
$$(\max\{ s \cdot n, 0\})^{shininess} * \text{specular}_{light} * \text{specular}_{material}\ ]_i$$

Rendering Realism



Cornel Measurement Lab

Rendering Realism

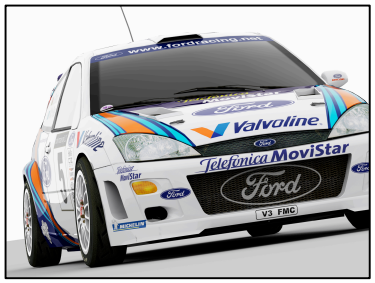


Real

Synthetic

Shirley, et. al., cornell

Is this real?

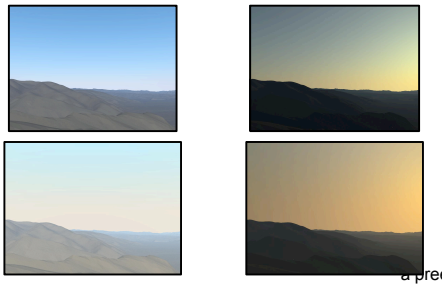


m fajaro, usc

Terrain Modeling: Snow and Trees Added

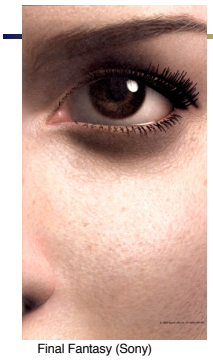
s premoze, et.al., utah

Rendering Realism



a preetham,

Humans

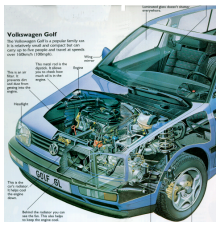


Final Fantasy (Sony)

Jensen et al.

Artistic Shading

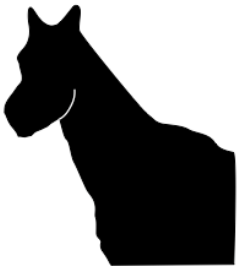
Is Photorealism Everything?



Is Photorealism Everything?

Enough Information…?

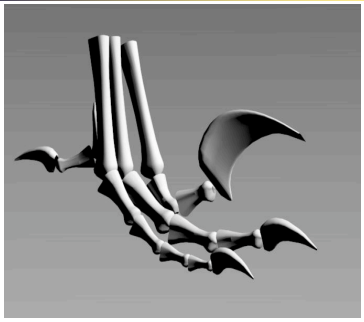
Just a bit more…



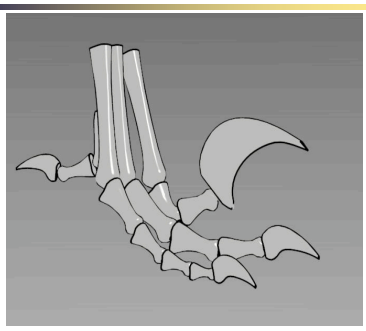
Or did we mean this…?

Diffuse shaded model

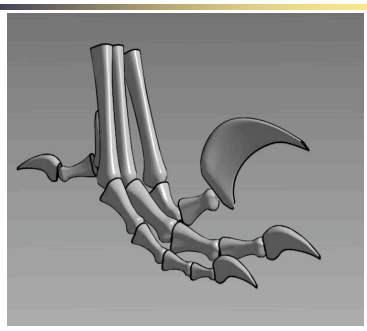
$I = c_r(c_a + c_l \max(0, L \cdot n))$ with $c_r=c_l=1$ and $c_a = 0$.

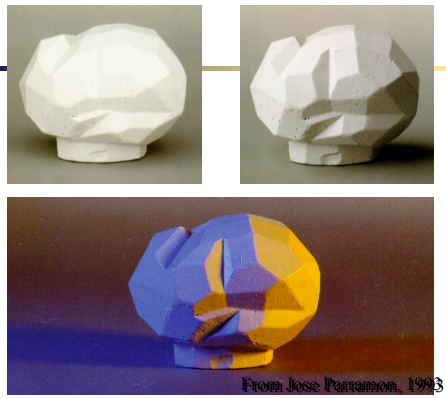


Just Highlights and Edge Lines



Hand-tuned Phong shading





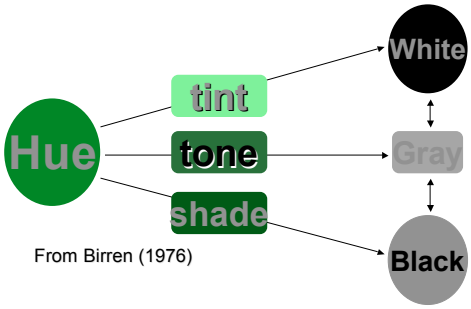
Shading used by Artists

Complementary Shading

Final Image

From "The Book of Color" by Jose Parramon, 1993

Tints, Tones, and Shades

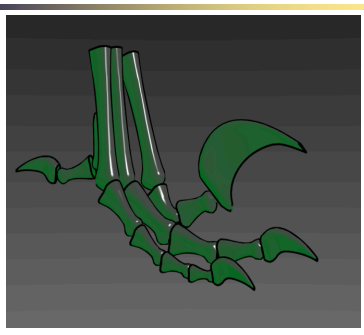


From Birren (1976)

Creating Tones

Green to Gray (tone)

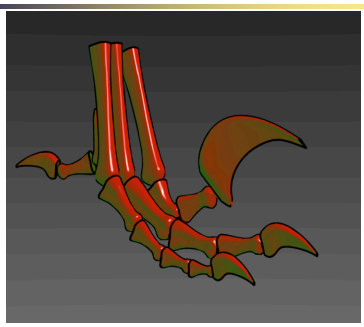
Model Shaded using Tones



Using Color Temperature

Warm to Cool Hue Shift

Constant Luminance Tone Rendering

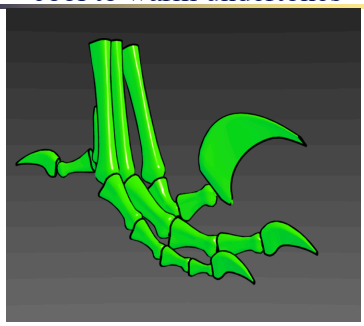


Creating Undertones

Warm to Cool Hue Shift

Green with Warm to Cool Hue Shift

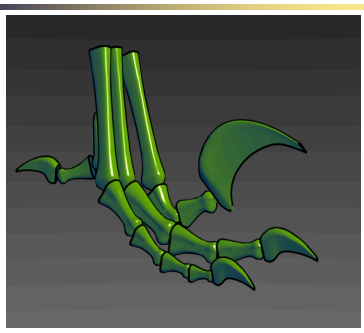
Model tone shaded with cool to warm undertones

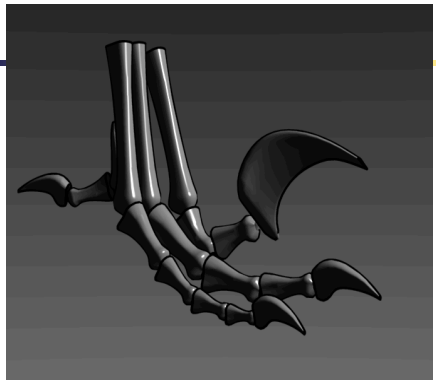


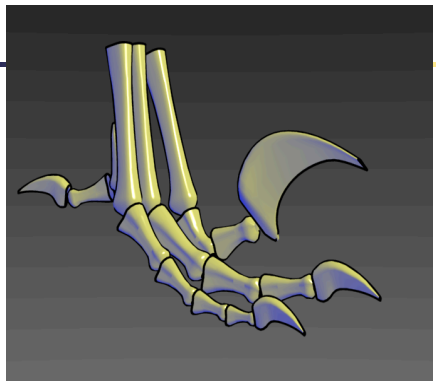
Combining Tones with Undertones

Green with Tone and Undertone

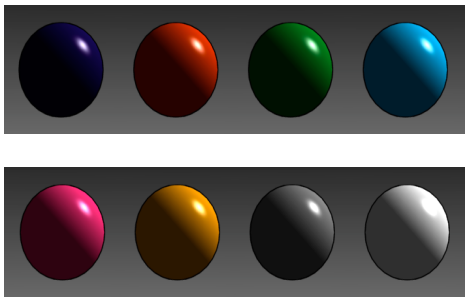
Model shaded with tones and undertones



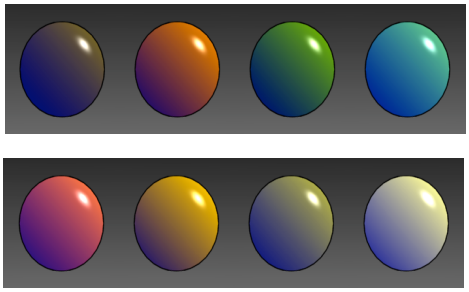




Phong Shaded Spheres



Spheres with New Shading



Phong Shading Formula

$$c = c_r (c_a + c_l \max(0, L \cdot n)) + c_l c_p \cos(h \cdot n)^n$$

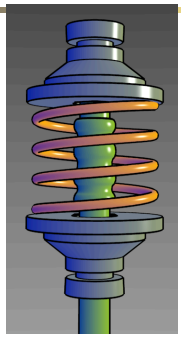
New Shading Formula

$$I = k_w c_{warm} + (1 - k_w) c_{cool}$$

where

$$k_w = (1 + (L \cdot n))*.5)$$

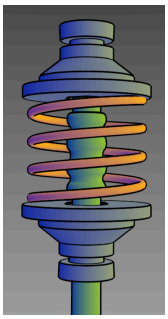
New Shading



OpenGL Approximation

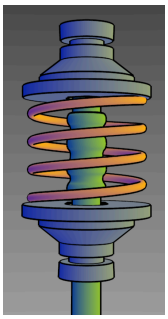
Without Highlights

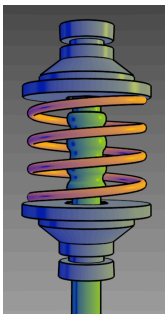
Light RGB Intensities
L_1 = (0.5, 0.5, 0.0)
L_2 = (-0.5, -0.5, 0)



OpenGL Approximation

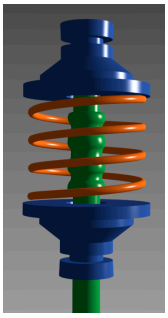
Without Highlights | With highlights

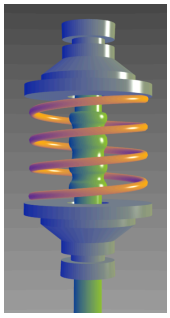


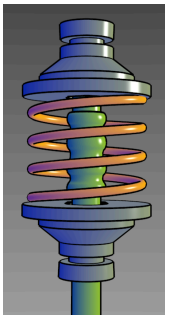


Warm to Cool Shading

Phong Shaded | New Shading Without Edge Lines | New Shading With Edge Lines







Toon Shading

A regular 3D version of Olaf

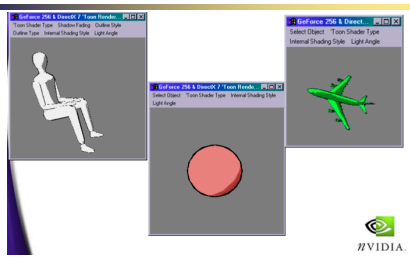
A regular 3D version of a boat

A Toon version of Olaf

A Toon pencil sketch version of the boat (Boat images courtesy of Discreet)

Intel: http://www.intel.com/labs/media/3dsoftware/nonphoto.htm

Toon Shading



Nvidia: developer.nvidia.com/object/Toon_Shading.html

Toon Shading

Blender: w3imagis.imag.fr/Membres/Jean-Dominique.Gascuel/DEAIVR/ Cours2002/17%20janvier/Blender-tutorial80.pdf

Non-Photorealistic Rendering

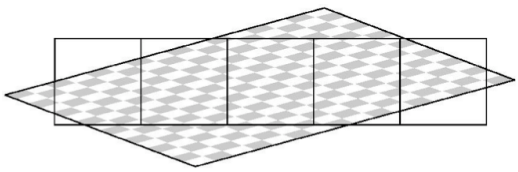
b gooch, et.al., utah

NonPhotorealistic Rendering

Surface mapping

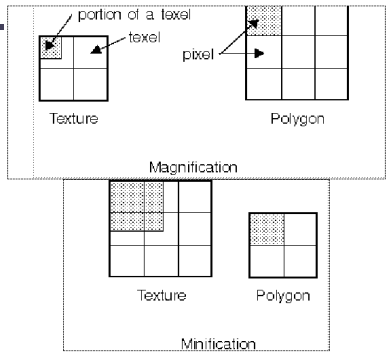
- Texture mapping
- Bump Mapping
- Displacement mapping
  - Actually moving geometry
  - ie Create screw from cylinder, Terrain, etc

What does a pixel see?

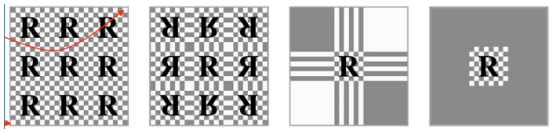


From Tomas Akenine-Moller

Controlling Filtering



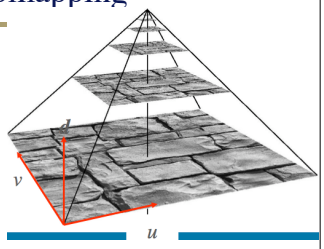
Repeat, Mirror, Clamp, Border



From Tomas Akenine-Moller

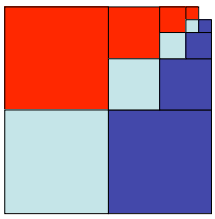
Mipmapping

- Image pyramid
- Half height and width
- Compute d
  - Gives 2 images
- Bilinear Interpolate in each image

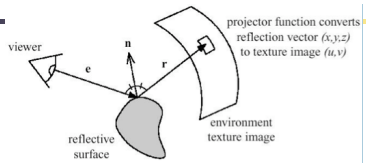


From Tomas Akenine-Moller

MipMapping Memory Requirements



Environment Mapping



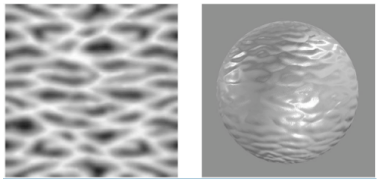
- Assume environment infinitely far away
- Sphere mapping
- Cube mapping (now norm)
  - No singularities
  - Much less distortion
  - Better result
  - Not dependent on view position

Cube Mapping

- Simple math:
  - Compute reflection vector *r*
  - Largest abs-value of component determines which cube face
    - Example: *r* = (5, -1, 2) give POS_X face
    - Divide *r* by 5 gives (u,v) =-1/5, 2/5)
  - Hardware often does all the work

Bump Mapping

Geometry | Bump map | Bump mapped geometry



Bump Mapping Example

Bump Mapping Example