Computer Graphics Lecture 06: Lighting and Shading

DR. ELVIS S. LIU
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY
SPRING 2018

Illumination

- Up to this point, we have considered only the geometry of how objects are transformed and projected to images
- We now discuss the illumination of objects: How the appearance of objects depends,
  - on the lighting that illuminates the scene, and
  - on the interaction of light with the objects in the scene

Illumination in Tom Clancy's The Division

- https://www.youtube.com/watch?v=ZfT0HIFZ37o

Illumination vs. Shading

- Often collectively referred to as shading
- Lighting, or illumination, is the process of computing the intensity and color of a sample point in a scene as seen by a viewer
  - lighting is a function of the geometry of scene (including the model, lights and camera, and their spatial relationships) and material properties (reflection, absorption,…)
  - calculates intensity at a particular point on a surface
- Shading is the process of interpolation of color at points in-between those with known lighting or illumination, typically vertices of triangles in a mesh
  - used in many real time graphics applications (e.g., games) since calculating illumination at a point is usually expensive. In ray-tracing only do lighting for samples (based on pixels or sub-pixel samples for super-sampling), no shading rule
  - uses calculated intensities to shade the whole surface or the whole scene
- On the GPU processing triangles, lighting is calculated by a vertex shader, while shading is done by a fragment or pixel shader
  - the term shader is ambiguous, unfortunately, and also not to be confused with shadows

Basic Lighting and Reflection

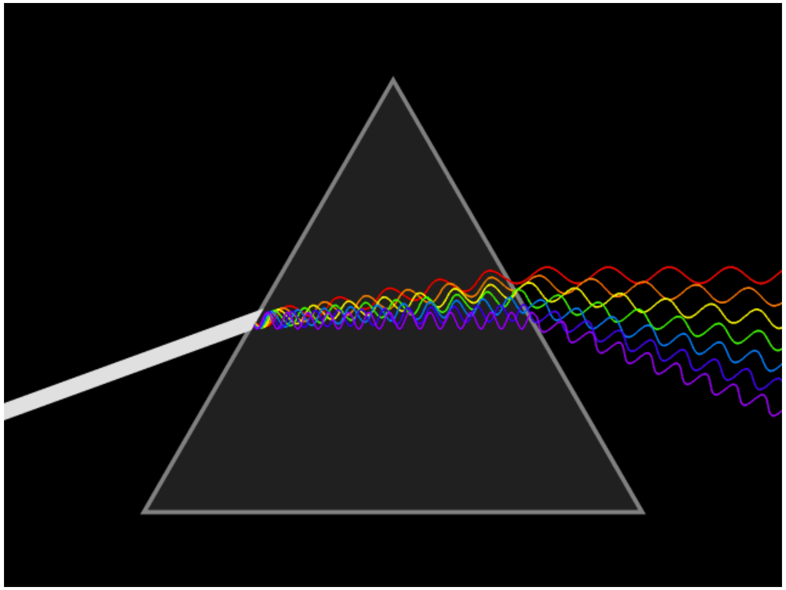
- Some of the basic qualitative properties of lighting and object reflectance that we need to be able to model include
  - Light source – there are different types of sources of light such as point sources (e.g., a small light at a distance), extended sources (e.g. the sky on a cloudy day), and secondary reflections (e.g., light that bounces from one surface to another)
  - Reflectance – different objects reflect light in different ways. For example, diffuse surfaces appear the same when viewed from different directions, whereas a mirror looks very different from different points of view.

Lighting

LECTURE 06: LIGHTING AND SHADING

What is Light?

- Light changes speed as it moves from one medium to another (for example, from air into the glass of the prism)
- This speed change causes the light to be refracted and to enter the new medium at a different angle (Huygens principle)
- The degree of bending of the light's path depends on the angle that the incident beam of light makes with the surface, and on the ratio between the refractive indices of the two media (Snell's law)
- The refractive index of many materials (such as glass) varies with the wavelength or color of the light used, a phenomenon known as dispersion. This causes light of different colors to be refracted differently and to leave the prism at different angles, creating an effect similar to a rainbow.



The Microscopic View of Light

- Light can be thought of as small packets of energy called photons but it exhibits both wave and particle properties
- For sound, wavelength determines pitch, with light it determines color we see: red has longest and violet the shortest wavelength
- Every atom has a nucleus which electrons orbit at various levels
- When a photon strikes an atom and is absorbed, electrons jump from a lower orbital to a higher one, but this is unstable…
- When an electron drops from a higher level orbit back to a lower, energy, photons are emitted (Planck relation)
  - $E=hf$ (frequency of emitted light increases with energy, h is Planck's constant) – the higher the frequency, the more energy (red vs. ultraviolet)
- The wavelength (color) of light emitted or absorbed corresponds to change in orbital; The nature of atoms and their electrons control type of light emission

The Microscopic View of Light (cont.)

- The different energy levels of electrons allow for different wavelengths of light to be reflected
  - Metals have “loose” electrons which can move relatively freely and occupy many different energy levels, which is why they are more reflective
  - Insulators have more constrained electrons which cannot change energy levels as easily, so they convert more absorbed light into heat instead of reflecting it

The Macroscopic View of Light

- Light is a type of electromagnetic radiation and can also be thought of as a continuous wave (as opposed to discrete photons)
- The particle description is best used to determine how energy is exchanged
- The wave nature of light is best used to describe how light propagates or travels, at light speed in space and air
  - Light changes speed in different media as a function of indices of refraction, hence the prism's dispersion – violet w/ highest frequency/energy slows more than red with lowest frequency/energy (Huygen's Principle, Snell's Law…)

The Macroscopic View of Light (cont.)

- Visible light has a wavelength between about 390nm – 700nm, a tiny portion of the entire spectrum
- Different wavelengths correspond to different colors
  - The 3 types of cones in retina react to all wavelengths in visual spectrum but have max response in different points on spectrum (we will discuss about it in a later lecture)

Lighting

- What the human eye sees is a result of light coming off of an object or other light source and striking receptors in the eye
- Trying to recreate reality is difficult
- Lighting in computer graphics refers to the simulation of light
- The techniques used are often heuristics which produce appropriate results
  - They do not work in the same way reality works, but can be computed faster
  - Instead of just specifying a single colour for a polygon we will specify the properties of the material that the polygon is supposed to be made out of
- Illumination models – express components of light reflected from or transmitted through surface

Direct Illumination

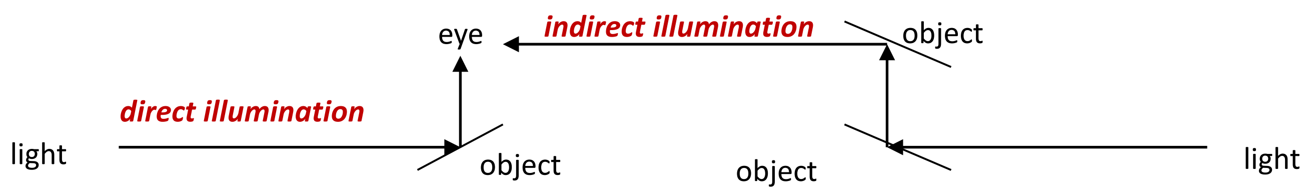
- Crudest hack: take only direct lighting information from light sources into account when computing a sample
- Usually involves an “ambient term” to set a minimum bar for inter-object reflection and thus light up visible surfaces
- Used in OpenGL <3.0, most traditional hardware pipelines (“fixed function” as opposed to programmable shader- driven pipelines on modern GPUs)

Global Illumination

- Simulates how other objects affect light reaching a surface element
- Lights and shadows
  - most light striking a surface element comes directly from emissive light sources
  - sometimes light from a source to a surface element is blocked by other objects; surface element is then in “shadow” from that light source, the blocker/occluder
  - classical h/w pipeline doesn’t account for blockers since each triangle is considered purely by itself

Global Illumination (cont.)

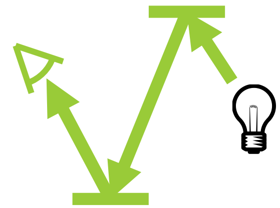
- Inter-object reflection (indirect illumination)
  - light bounces off other objects toward our surface element, to add to direct illumination; diagram shows only a single specular ray out of a potentially infinite number of rays

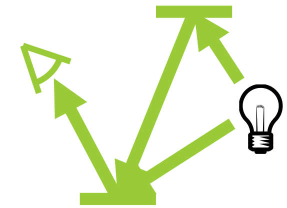


Local vs. Global Rendering

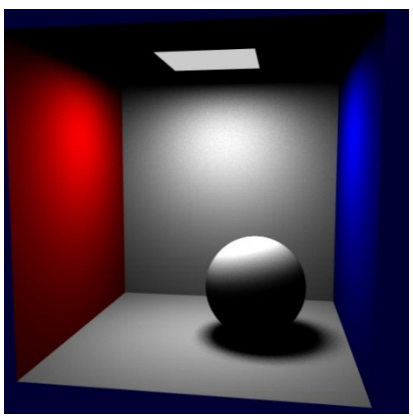
- Correct shading requires a global calculation involving all objects and light sources
  - Incompatible with pipeline model which shades each polygon independently (local rendering)
- However, in computer graphics, especially real tie graphics, we are happy if things look right
  - There exist many techniques for approximating global effects

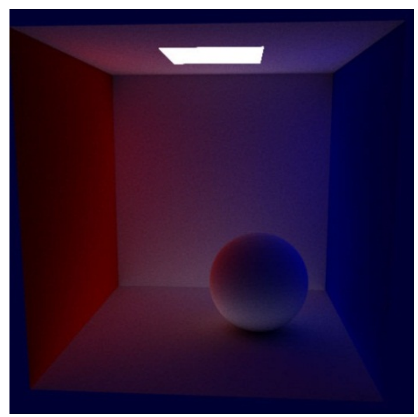
Global Illumination (both diffuse & specular)

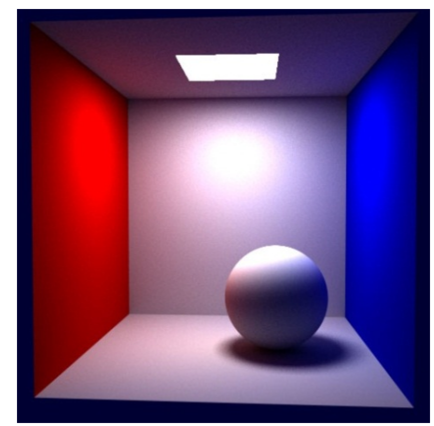




Direct illumination + Indirect illumination = Total illumination







Diffuse Interreflection is Surprisingly Important

Total illumination
(normal image)

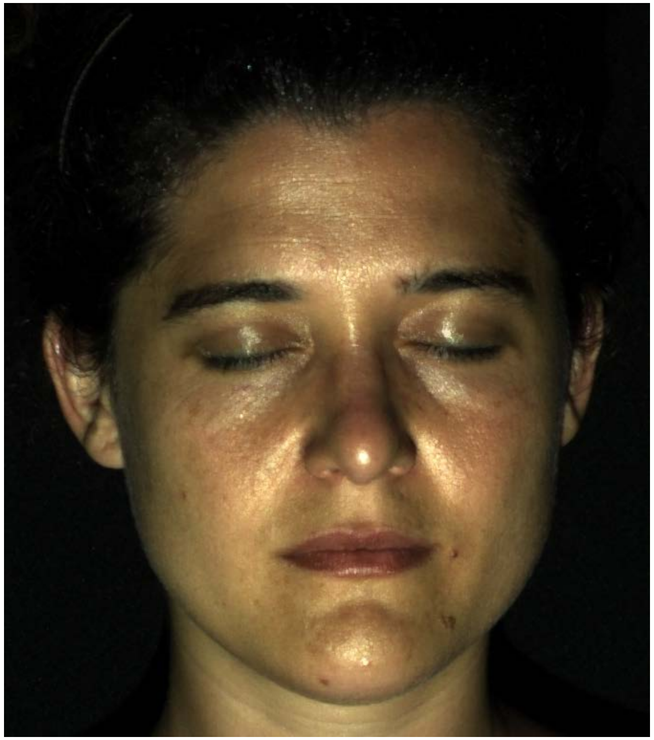
Diffuse Interreflection

Direct illumination

Diffuse Interreflection

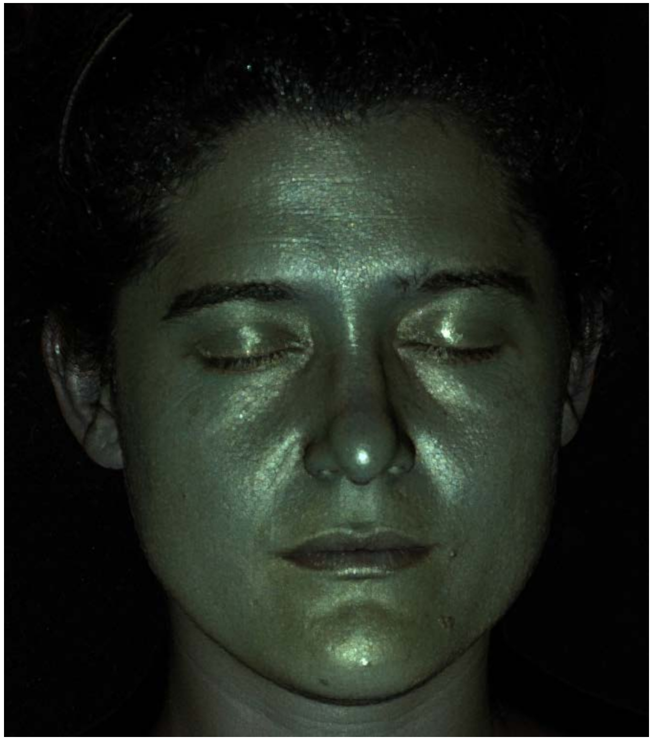
Indirect illumination
(diffuse interreflection)

Human Face



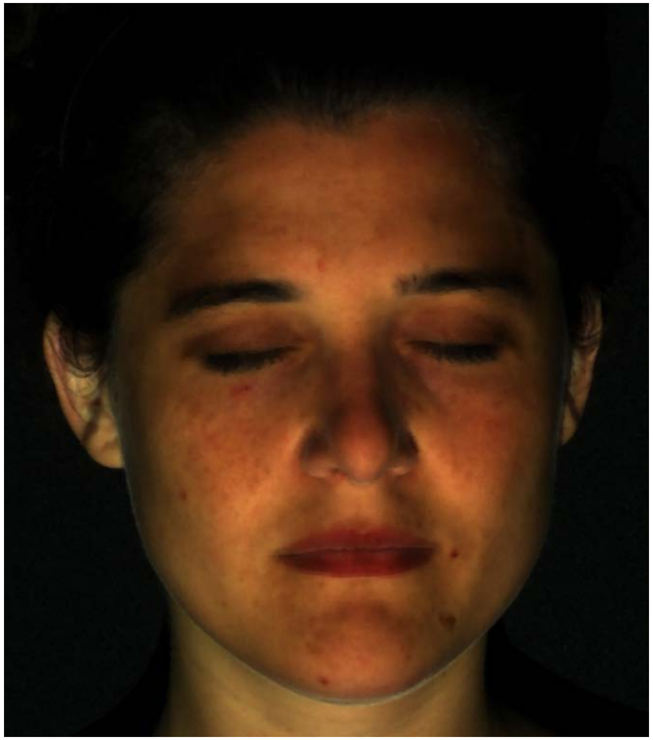
Total illumination (normal image)

Human Face



Direct illumination

Human Face



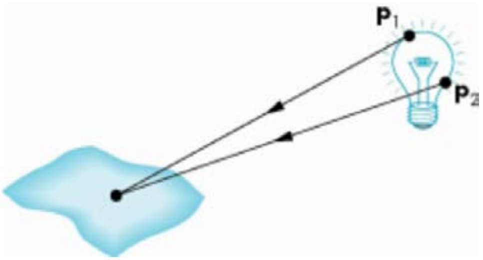
Indirect illumination

Which one is Global?

- https://www.youtube.com/watch?v=MJV55-Buw60

Recall Assumption of Light

- Light travels in a straight line
- Ignore its frequency composition and care about its component-wise intensity (energy) – R, G, B
- Light source: general light sources are difficult to work with because we must integrate light coming from all points on the source

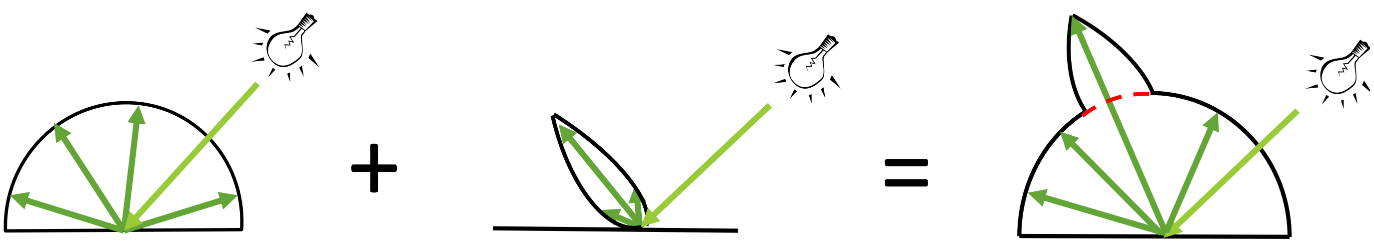


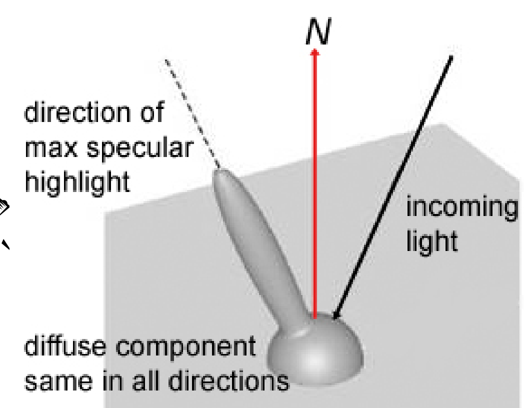
Simplified Light Source Models

- Ideal point light source
  - Modeled by position and colour; shines equally in all directions
  - Can take into account decay with respect to distance
  - Follows perspective projections
  - Parallel projections: distant source = point at infinity
- Spotlight
  - Restrict light from ideal point source
- Global Ambient Light
  - General brightness (same amount of light everywhere)
  - A rough model for inter-surface reflections from all light sources

Modelling Reflectance: The BRDF

- Light arriving at a surface can scatter in many directions
  - The direction of scattering is determined by the material
  - Intensity at a given outgoing direction is dependent on incoming direction and material properties (how much energy it absorbs, how much it reflects diffusely vs. specularly, etc.)
- Model or measurement of reflectance is called the bidirectional reflectance distribution function (BRDF): for any incoming light ray how much energy is reflected for any outgoing light ray





Diffuse Reflections Specular Reflections Total Scattering Distribution (*BRDF*)

Modelling Reflectance: The BRDF (cont.)

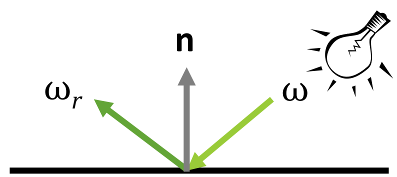
- When light hits a surface, it can be reflected, refracted, absorbed, or otherwise scattered (e.g., subsurface scattering)
- BRDF represents material properties of an object and how it reflects light
- Given an incoming direction, outgoing direction, incident point on object, and properties of lights, BRDF provides fraction of light reflected off surface
  - can be measured with equipment or estimated based on some model
  - family of surfaces: function of 4 variables: incoming and outgoing angles in spherical coordinates, specified by zenith and azimuth angles; frequency dependence adds a 5th.
  - most BRDFs are anisotropic, i.e., not entirely rotationally invariant (e.g., wood grain)
  - conservation of energy (quantity of reflected light <= incident light) and reciprocity (can reverse direction of incoming and outgoing rays – fraction stays the same)

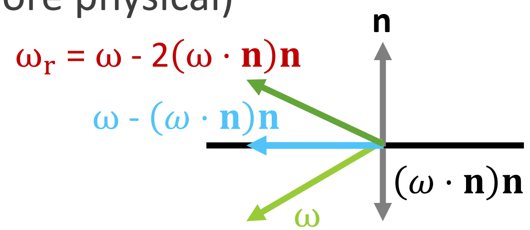
Modelling Reflectance: The BRDF (cont.)

- Note that the BRDF is often implicit in simplest illumination models, and a few parameters (typically diffuse and specular reflection coefficients) are adjusted experimentally until desired visual outcome is achieved
- For this course, primarily concern ourselves with simple reflectance models

Simple Reflectance Models

- Some of these models you have seen before in OpenGL
- ***Lambertian*** light scatters equally (output radiance is equal) in all outgoing directions (i.e. viewer independent) – non-physical
- ***Diffuse*** light scatters (possibly unevenly) in all outgoing directions, e.g., rug, paper, unvarnished wood (more physical)





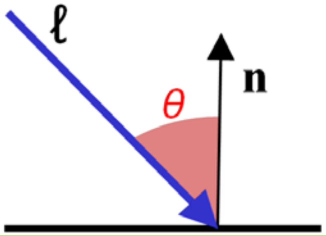
- ***Mirror*** light scatters in a single direction, $\omega_r = \omega - 2(\omega \cdot \mathbf{n})\mathbf{n}$ - non=physical
- ***Specular*** light scatters tightly around a particular direction (shiny objects with sharp highlights) – more physical
- ***Glossy*** light scatters weakly around a particular direction – more physical

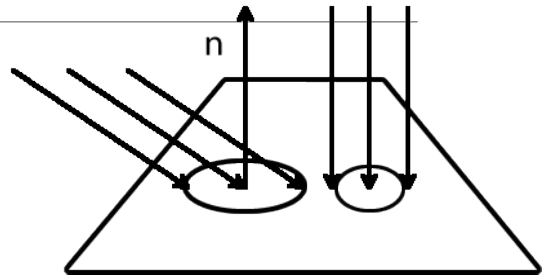
Modelling Reflectance: Lambertian

- Lambertian surfaces have uniform, diffuse-only scattering;
  same apparent brightness independent of view direction and incoming light direction
- Most materials are not perfectly Lambertian; most BRDFs are viewer dependent
- Lambert's cosine law:

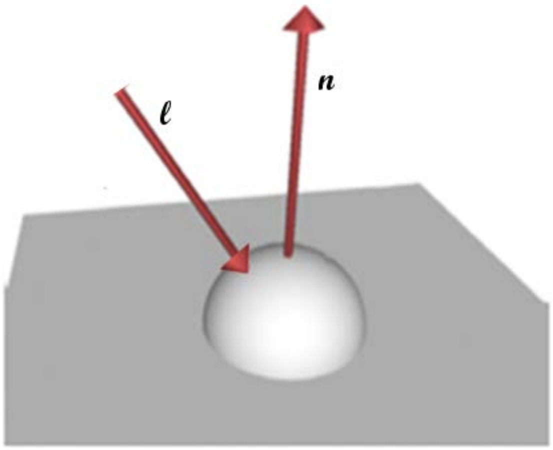
$$I = i_{dir}\cos(\theta) = i_{dir}\,(\boldsymbol{n} \cdot \boldsymbol{\ell})$$

i_{dir} is intensity (light's color) of directional light (rays parallel to $\boldsymbol{\ell}$)





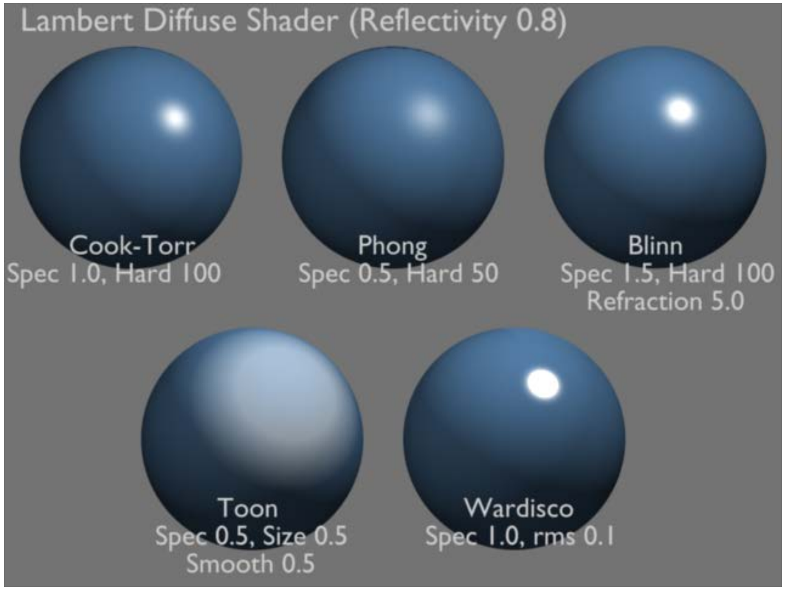
As angle between light and normal increases light's energy spreads across a larger area



3D visualization: hemisphere represents equal magnitude of reflected intensity for any outgoing vector. Real surfaces will have asymmetric BRDFs.

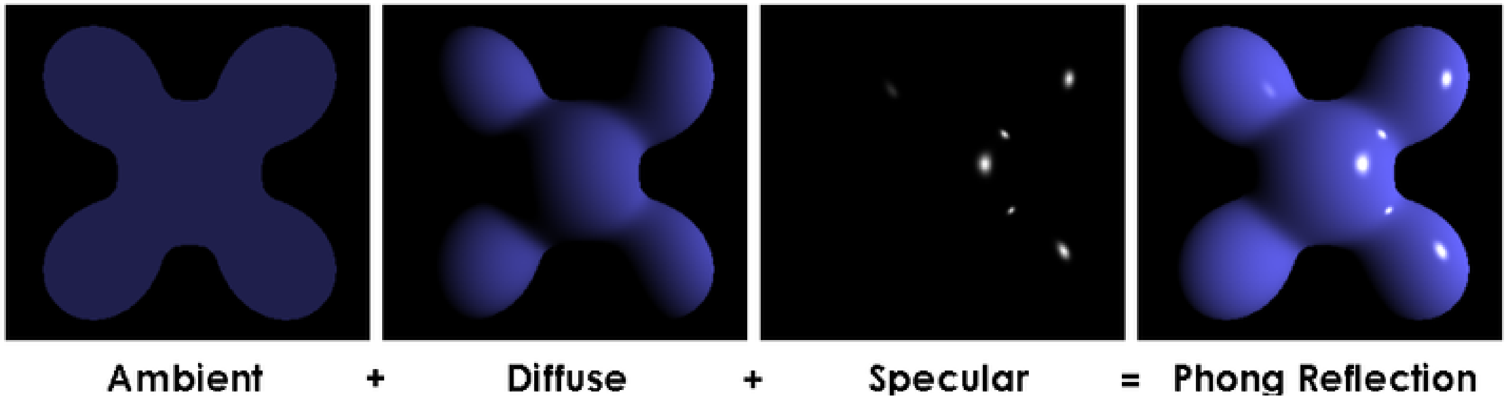
Modelling Reflectance: Lambertian (cont.)

- Several lighting models attempt to approximate these phenomena
- Models you have seen before– no physical foundation, but looks okay
  - Phong Model
  - Blinn-Phong Model
- Some more complicated models are physically based
  - Cook-Torrance Model
  - Oren-Nayer Model
- Which model we use largely depends on our application
  - Usually a performance vs. accuracy tradeoff



Illumination Model: Phong

- Simple model (not physically based)
- Splits illumination at a surface into three components
  - Ambient – Non-specific constant global lighting (hack)
  - Diffuse – Color of object under normal conditions using Lambert's model
  - Specular – Highlights on shiny objects (hack)
    - Proportional to $(\boldsymbol{R}\cdot\mathbf{V})^{\alpha}$ so a larger α results in a more concentrated highlight and glossier object



Lighting – Ambient Light

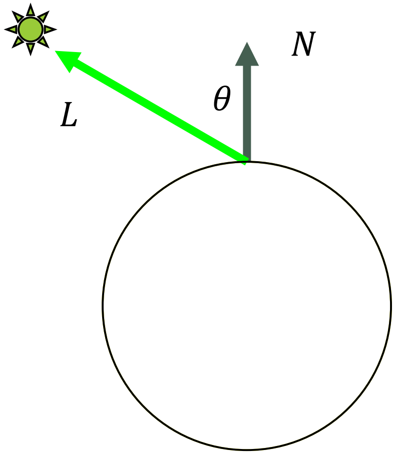
- Non-directional light source that is product of multiple reflections from surrounding environment
- Ambient light model and ambient illumination equation

$$I = k_a I_a$$

  - I - intensity of reflected light from surface
  - k_a - ambient-reflection coefficient
    - ranges from 0 to 1, determines amount of ambient light reflected by object's surface
    - material property
  - I_a - ambient light, constant for all objects

Lighting – Diffuse Light

- Reflected from point with equal intensity in all directions
- Typical for dull, matte surfaces such paper or flat wall paint
- Modeled by Lambert's laws
  - Brightness depends only on angle between light source direction and surface normal



Lighting – Diffuse Illumination Model

- Diffuse illumination equation

$$I_d = k_d \, I_p \, cos \, \theta$$

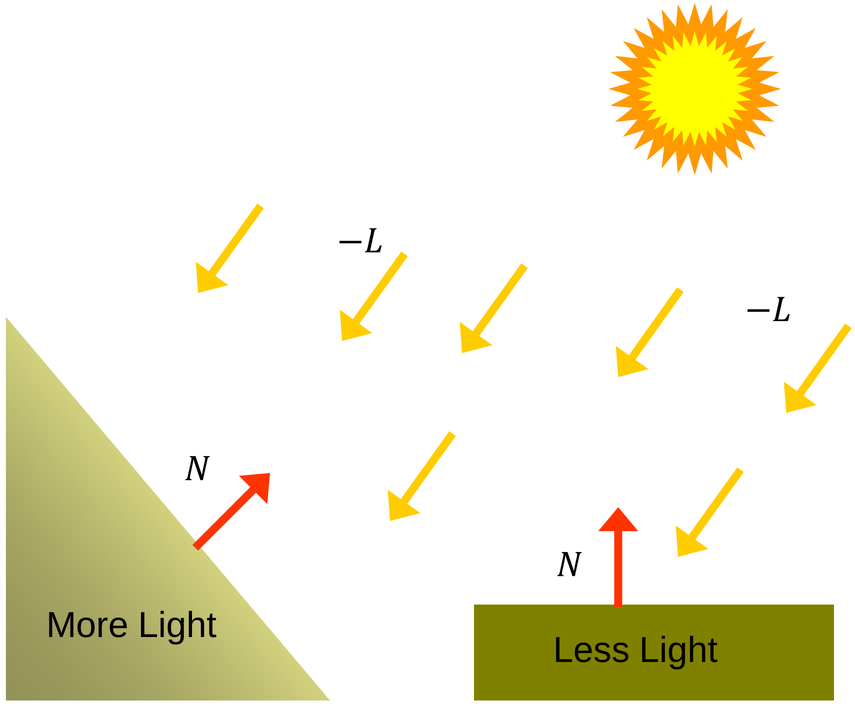
  - k_d - diffuse-reflection coefficient or diffuse reflectivity of surface which varies between 0 and 1 and depends on surface material
  - I_p - intensity of point light source
  - θ - angle between surface normal N and light vector L

- Assuming that N and L are unit length vectors, we can write cosine as simple dot product and diffuse illumination equation as

$$I_d = k_d \, I_p \, (N \cdot L)$$

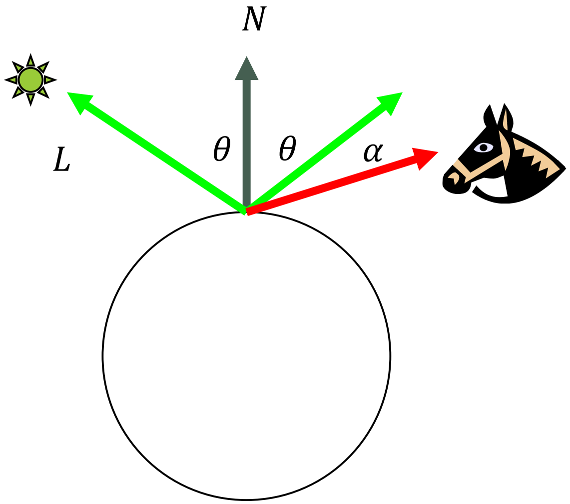
Lighting – Diffuse Light Effect

- Directional
- Depends on $(N \cdot L)$
- Attenuates based on surface orientation relative to light vector



Specular Light

- Specular component of reflection light accounts for highlights caused by light reflecting primarily in one direction
- Specular reflection is mirror-like
  - Gives rise to shiny spots on surfaces
  - Depends on angle between viewer and reflected ray

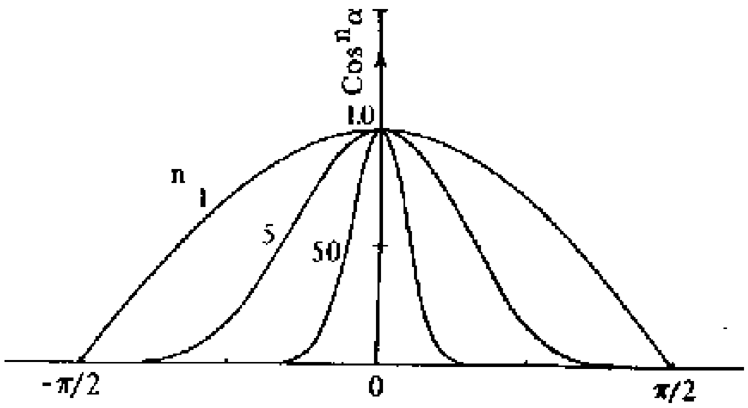


Lighting - Specular Illumination Model

- Bui-Thong Phong developed popular approximation to specular component of light with Phong's model

$$I_s = k_s I_p \cos^n \alpha = k_s I_p (V \cdot R)^n$$

  - k_s - specular-reflection coefficient
  - I_p - intensity of point light source
  - α – angle between eye V and reflected light R
  - n – Phong specular-reflection exponent
    - varies from 1 to several hundred
    - $n = 1$ gives broad gentle falloff to highlight
    - large settings of n give focused highlight

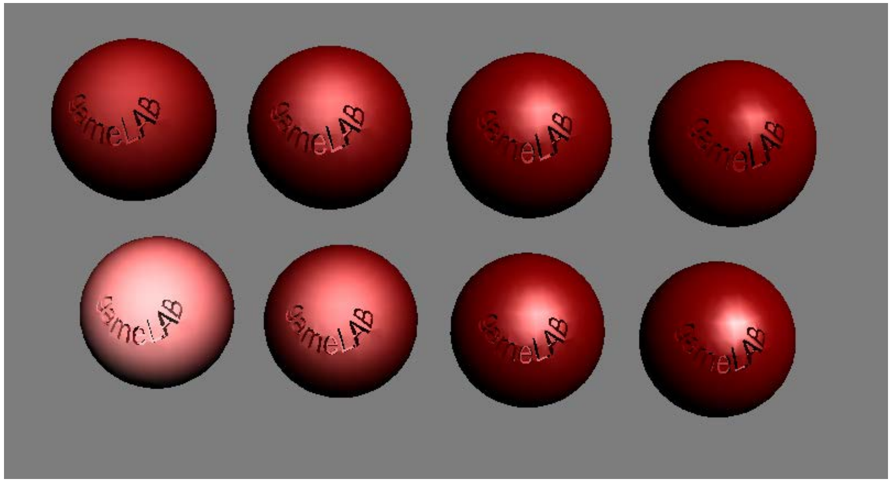


Effects of Specular Coefficients

exponent, $n =$ 5 10 20 40

intensity, $I_p = 0.5$

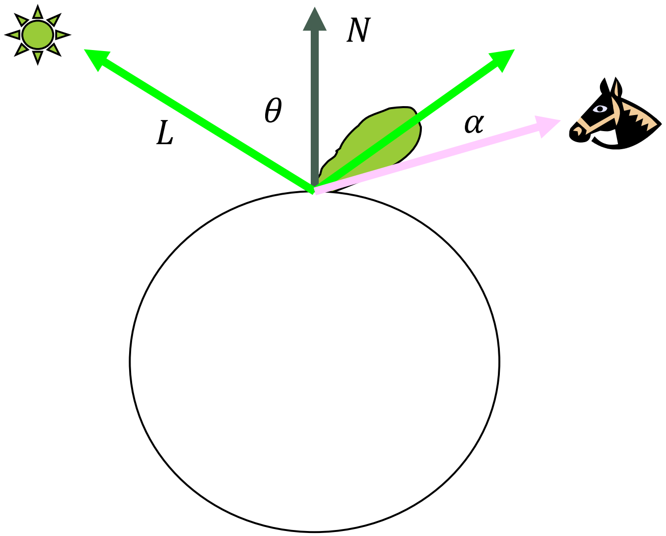
intensity, $I_p = 1.0$



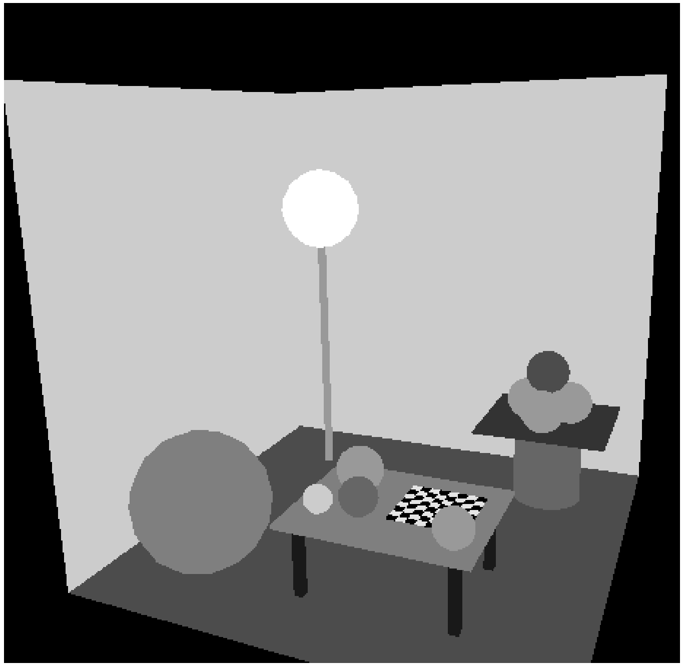
$$I_s = k_s I_p \cos^n \alpha$$

Lighting – Specular Light Effect

- Specularly reflected light
  - Unlike diffusely reflected light, is view dependent
  - Often refer to as specular bump which shows that most light is reflected in particular direction

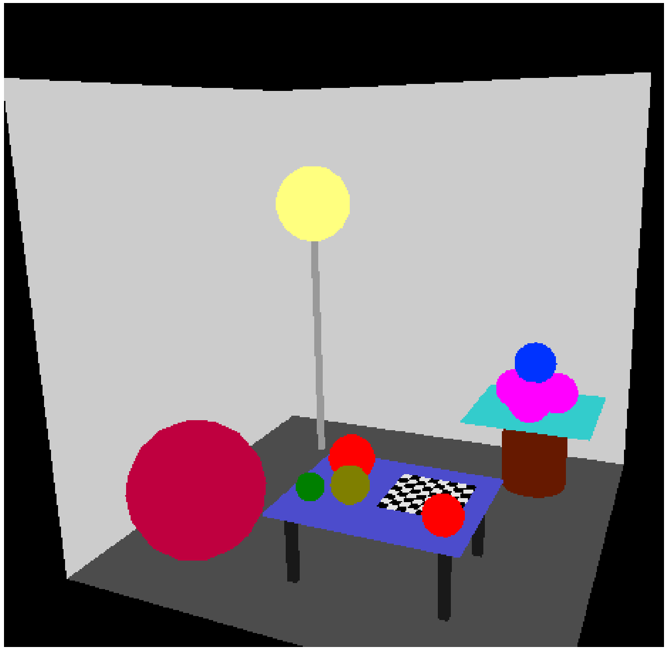


Lighting - Example



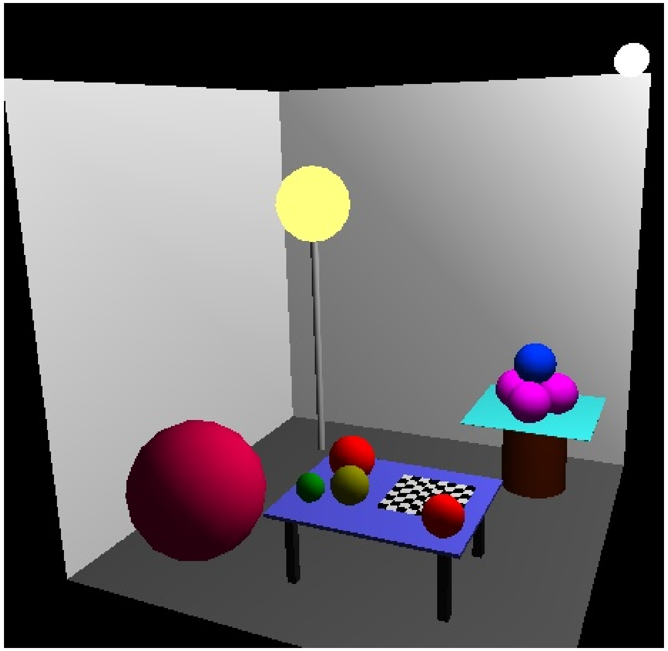
Lighting Example – Ambient Light

- $I = k_a I_a$



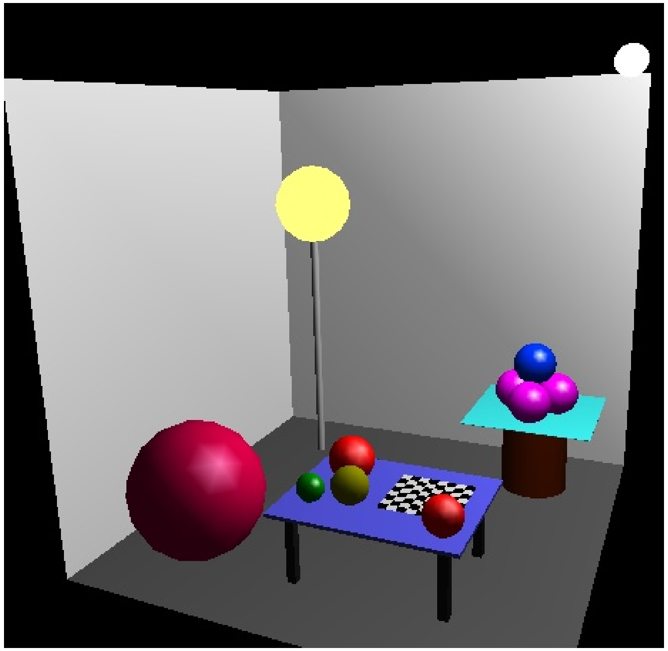
Lighting Example – Diffuse Light

- $I_d = k_d I_p \cos\theta$



Lighting Example – Specular Light

- $I_s = \ k_s \, I_p \, cos^n \alpha$



Total Radiance

- Ambient: $I = k_a I_a$
- Diffuse: $I_d = k_d I_p (N \cdot L)$
- Specular: $I_s = k_s I_p (V \cdot R)^n$
- Ignoring ambient component

$$I_T = \sum_{k=1}^{m} f(I_p, L, V, N, k_d, k_s, n)$$

$\Rightarrow$ Total radiance I_T is equal to a sum over m point / directional light sources

Illumination Models: Blinn-Phong

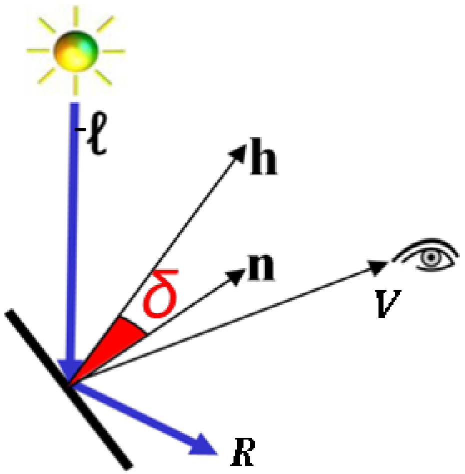
- Variation on Phong model which modifies the specular term to be more computationally efficient
- The new specular term uses the half angle between the viewer and the light (not a function of the orientation/normal) instead of the angle between the reflected light ray and the viewer

$$\sum_{m \in lights} i_{dir} k_s O_s (\mathbf{n} \cdot \mathbf{h})^{\alpha}$$

- $\mathbf{h}$ is the half angle

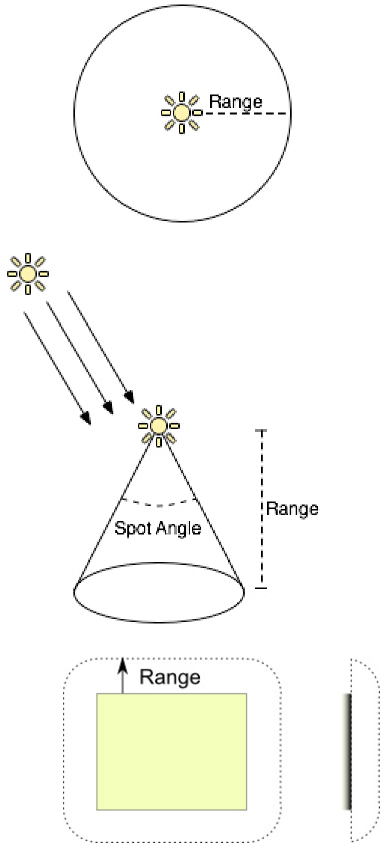
$$h = \frac{\boldsymbol{l} + \boldsymbol{V}}{|\boldsymbol{l} + \boldsymbol{V}|}$$

- $\boldsymbol{\ell}$ is normalized direction from point to light,
  V is normalized vector to viewer
- Use Blinn-Phong or Phong?
  - Doesn't matter, they look slightly different but they're both hacks



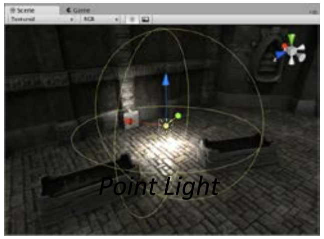
Lights in Unity

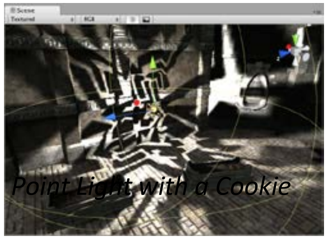
- Point lights
  - Shine from a location equally in all directions, like a light bulb
- Directional lights
  - Placed infinitely far away and affect everything in the scene, like the sun
- Spot lights
  - Shine from a point in a direction and only illuminate objects within a cone, like headlights of a car
- Area lights (only available for lightmap baking)
  - Shine in all directions to one side of a rectangular section of a plane
  - Used as a lightmap



Performance Considerations

- Lights can be rendered in one of two methods
  - Vertex lighting and Pixel lighting
- Vertex lighting
  - Only calculates lighting at vertices of game models and interpolates the lighting over the surfaces of the models
- Pixel lights
  - Calculated at every screen pixel, much more expensive
  - Some older graphics cards only support vertex lighting
- Pixel lighting is slower to render, allows some effects that are not possible with vertex lighting
  - Normal-mapping, light cookies (alpha mask) and realtime shadows (Pro-only feature) are only rendered for pixel lights





Lighting in Unity3D

- https://www.youtube.com/watch?v=u5DNkxkXBeI

Advanced Global Illumination Techniques

- Real Time Global Illumination
- Snowdrop Renderer
  Tom Clancy’s The Division
- Uses Deferred Radiance Transfer Volumes and Volumetric lighting to create real time global illumination , including shadows, light filtering through bullet holes, and changes in environmental light sources due to player actions.

Shading

LECTURE 06: LIGHTING AND SHADING

What is Shading?

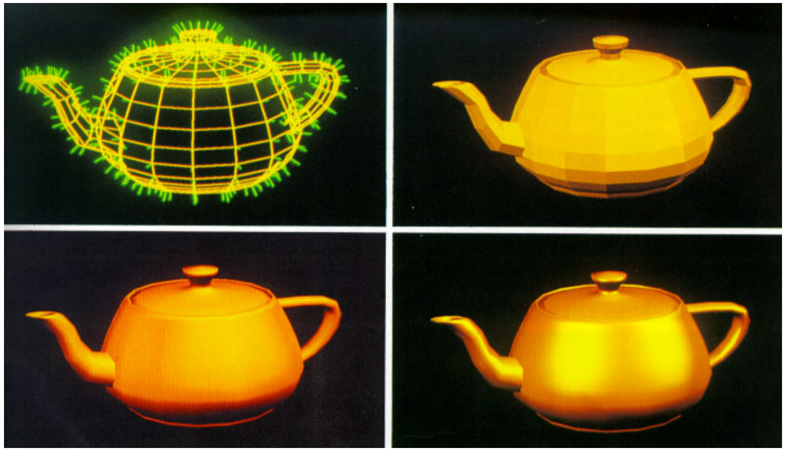
- Shading is the process of altering the colour of an object, surface, or polygon in 3D scene, based on its angle to lights and its distance from lights to create a photorealistic effect.
- Shading is performed during the rendering process by a shader.

Shading of Surfaces

- We can shade a point on a surface using a local illumination model
- How about surface?
  - Flat (planar) polygons are computationally attractive
    - Normals, intersections, visibility, projections, etc., are all easy to compute
    - Hardware acceleration available
  - Curved surfaces are often tessellated into many small flat polygons in graphics (polygon meshes)

Shading Models

- How pixel colour (shading) is determined across geometry surfaces
- Some common shading models
  - Flat
  - Smooth (Gouraud)
  - Phong



Flat Shading Model

- Pixels attributing to each geometry surface receive same intensity
  - Based on calculation using common surface normal and incident light
- Lighting: Evaluate the lighting equation at the center of each polygon using the associated normal
- Shading: Each sample point on the polygon is given the interpolated lighting value at the vertices (i.e., a total hack)

Flat Shading Model - Properties

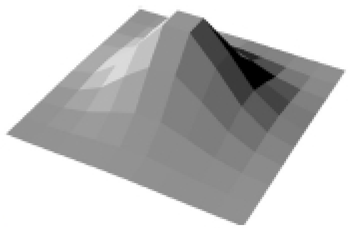
- Fast and simple – A single color across the polygon
- Low quality
- Intensity discontinuity across edges
- Flat shading would be correct if
  - Light source is at infinity, i.e., light vector **l** is constant, so **n·l** is constant across polygon, and
  - Viewer is at infinity, so **r·v** is constant across polygon, and
  - Polygons represent actual surface, not an approximation

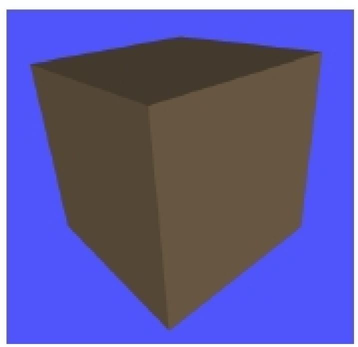
Flat Shading – Example

- Assume we are given the following
  - $\vec{e}$ – centre of projection in world coordinates
  - $\vec{l}$ – point light source location
  - I_a, I_d – intensities of ambient and directional light sources
  - k_a, k_d, k_s – coefficients for ambient, diffuse, and specular reflections
  - n – exponent to control width of highlights

- For a triangle with counterclockwise vertices $\vec{p}_1$, $\vec{p}_2$, and $\vec{p}_3$, as seen from the outside, let the midpoint be $\vec{p}$ = 1/3 ($\vec{p}_1$ + $\vec{p}_2$ + $\vec{p}_3$) with normal $\vec{N} = \frac{(\vec{p}_2-\vec{p}_1)\times(\vec{p}_3-\vec{p}_1)}{||(\vec{p}_2-\vec{p}_1)\times(\vec{p}_3-\vec{p}_1)||}$, then we may find the intensity at $\vec{p}$ and fill the polygon with
  - $E = I_a k_a + k_d I_d \max\left(0, \vec{N}\cdot\vec{s}\right) + k_s I_d \max\left(0, \vec{R}\cdot\vec{c}\right)^{\mathrm{n}}$
  - where $\vec{s} = \frac{\vec{l}-\vec{p}}{||\vec{l}-\vec{p}||}$, $\vec{c} = \frac{\vec{e}-\vec{p}}{||\vec{e}-\vec{p}||}$, and $\vec{R} = -\vec{s} + 2(\vec{s}\cdot\vec{N})\vec{N}$

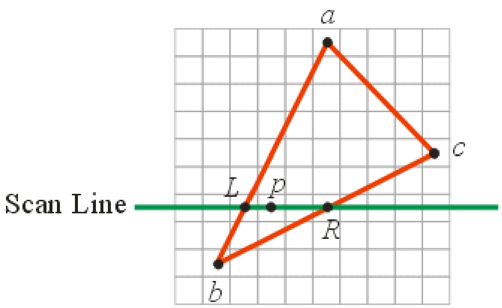
Results of Flat Shading

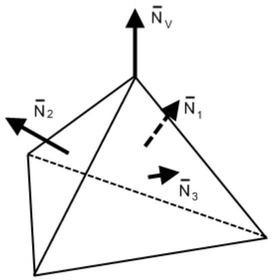


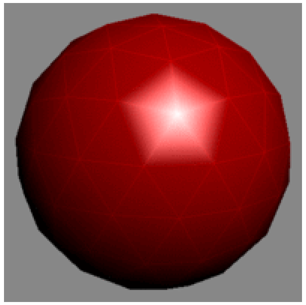


Gouraud Shading Model

- Vertex normal is calculated based on average of adjacent surface normals
- Lighting computation based on Phong illumination model are calculated to produce light intensities at vertices
- Screen pixel intensity/color are linearly interpolated across surface of related vertices
- An issue with Gouraud shading is that specular highlights appear to “jump” as the object rotates. Simple fix is to increase polygon count, at cost of much more computation



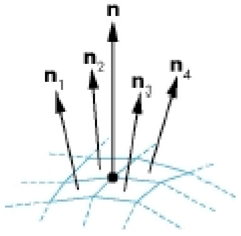




How to compute the normal at a mesh vertex?

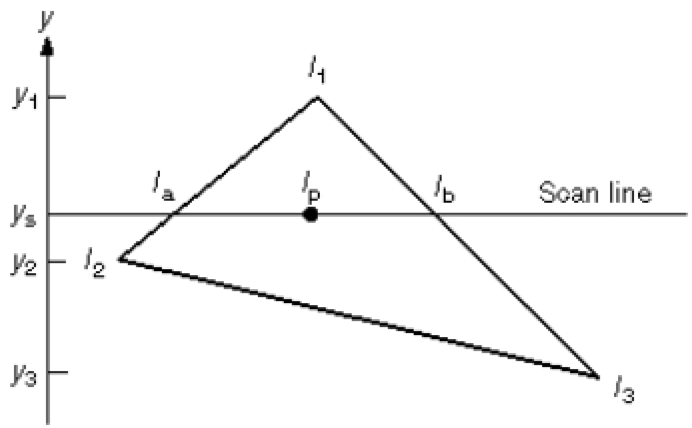
- One method is to take the normalised average of the normal of adjacent faces

$$n = \frac{n_1 + n_2 + n_3 + n_4}{|n_1 + n_2 + n_3 + n_4|}$$



Colour Interpolation

- Interpolate colours along edges and scan-lines
- Can be done incrementally, i.e., via scan-lines

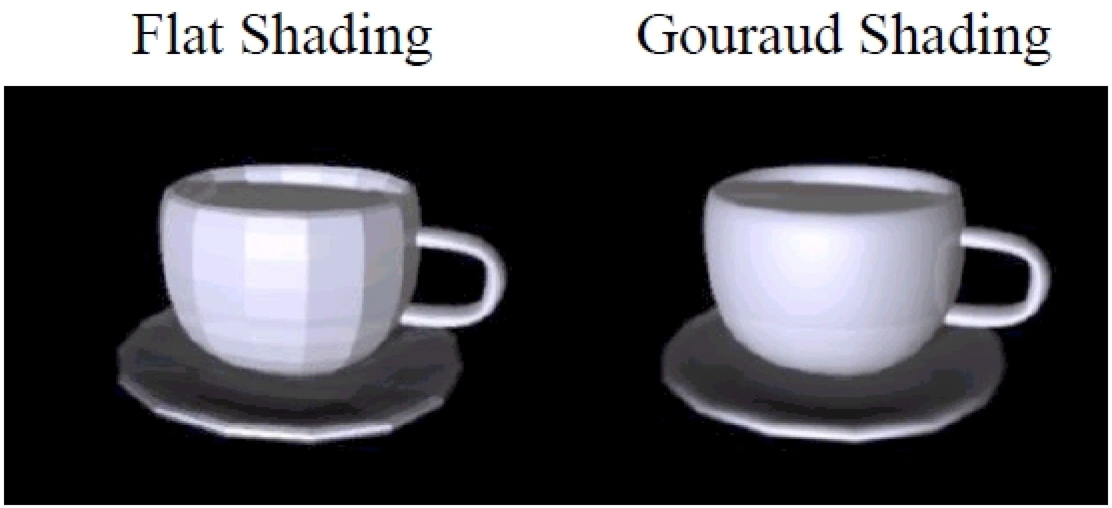


$$I_a = I_1 - (I_1 - I_2)\frac{y_1 - y_s}{y_1 - y_2}$$

$$I_b = I_1 - (I_1 - I_3)\frac{y_1 - y_s}{y_1 - y_3}$$

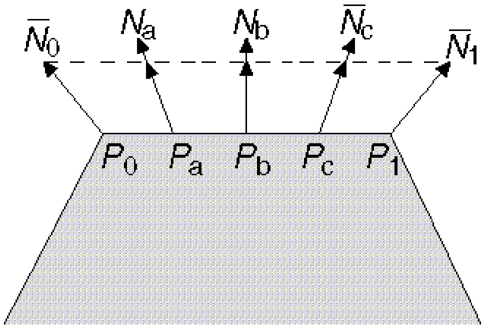
$$I_p = I_b - (I_b - I_a)\frac{x_b - x_p}{x_b - x_a}$$

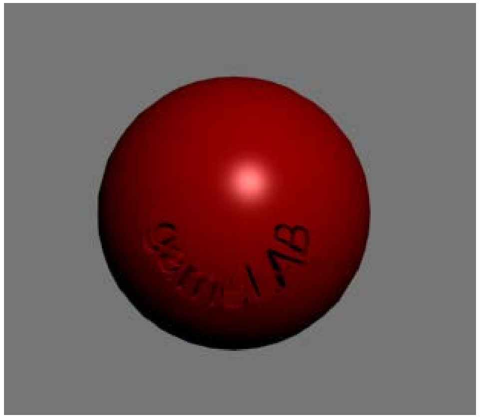
Flat vs. Gouraud Shading



Phong Shading Model

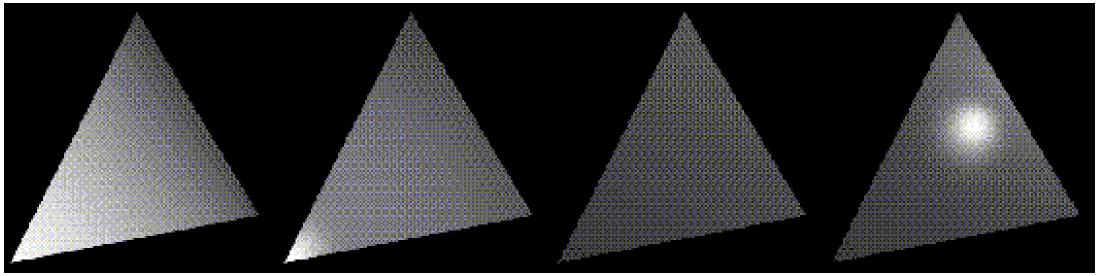
- Improvement of Gouraud shading
- Instead of interpolating illumination intensity across surface among vertices, interpolate vertex normals across surface
- For every sample point on polygon we interpolate normals at vertices of the polygon and compute color using lighting equation with the interpolated normal at each interior pixel
- More accurate representation of true surface normal at each point (and we don’t get moving specular highlights)





Gouraud vs Phong

- (a) Gouraud shading spreads vertex highlight
- (b) Phong shading does not spread vertex highlight
- (c) Gouraud shading misses interior highlight
- (d) Phong captures interior highlight



(a) (b) (c) (d)

Shading in OpenGL

- OpenGL only directly supports Gouraud shading or flat shading
- Gouraud is enabled by default, computing vertex colors, and interpolating colors across triangle faces
- Flat shading can be enabled with glShadeModel(GL FLAT). This renders an entire face with the color of a single vertex, giving a faceted appearance
- With *pixel shaders* on programmable graphics hardware, it is possible to achieve Phong shading by using a small program to compute the illumination at each pixel with interpolated normal
- It is even possible to use a *normal map* to assign arbitrary normals within faces, with a pixel shader using these normals to compute the illumination