Illumination & Shading

- Lighting and Light sources
- Reflection models: Ambient, Diffuse, Specular
- Shading models: Flat, Gouraud, Phong
- Shortcomings of shading

Lighting

- By now, given a complex 3D model and a viewpoint, we can:
  - transform the model in the scene (transformation matrices)
  - project model (polygons) into the projection plane (perspective/parallel projection)
  - determine the visible surfaces and remove hidden surfaces (z-buffer, BSP-Trees, …)
  - set the right pixels (scan converting)
- But what color should those pixels be?

Lighting

- To create a realistic image, we need to simulate the *lighting* of the surfaces in the scene
- Fundamentally we should simulate *physics* and *optics* phenomena
- Perfect simulation is slow, so we use a lot of approximations, also known as *hacks*, to do this simulation fast enough for real-time applications.
- Though many of the hacks employed have little grounding in theory, they work well in practice and are easy to compute.

Definitions

- **Lighting/Illumination models:** express factors related to light, to determine a surface's color at a given point
  Examples:
  - Ambient
  - Diffuse
  - Specular
- **Shading models:** determines when an illumination model will be invoked and with which arguments
  Examples:
  - Flat
  - Gouraud
  - Phong

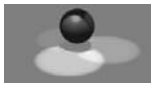
Definitions

Illumination models fall into two categories:

- Physically based:
  - models based on the actual physics of light interacting with matter
  - used in realistic graphics

- Empirical:
  - simple formulations that approximate observed phenomenon
  - mostly used in interactive graphics for simplicity

Components of Illumination



- Two components of illumination:
  - light sources (emitters)
  - surface properties
- Light sources properties:
  - Emmitance spectrum (color of the light)
  - Geometric attributes (Position, Direction, Shape)
  - Directional attenuation
- Surface properties:
  - Reflectance spectrum (color of the surface)
  - Geometric attributes (Position, Orientation, Micro-structure)
  - Absorption

Ambient Light Sources

- Objects not directly lit are typically still visible
  - e.g., the ceiling in this room, undersides of desks
- This is the result of *indirect illumination* from emitters, bouncing off intermediate surfaces
- Too expensive to calculate (in real time), so we use a hack called an *ambient light source*
  - No spatial or directional characteristics; illuminates all surfaces equally
  - Color and intensity of ambient light, depends on the environment
  - Amount reflected depends on surface properties

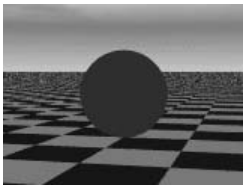
Ambient Light Sources (2)

- For each sampled wavelength, the ambient light reflected from a surface depends on
  - Intensity of the ambient light source which is constant for all points on all surfaces : $I_{a\ (ambient)}$
  - Ambient-reflection coefficient which depends on surfaces properties: $k_{a\ (ambient)}$ *(ranges from 0 to 1)*

$$I = k_a I_a$$

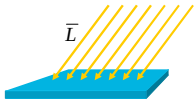
Ambient Light Sources (3)

a scene lit only with an ambient light source



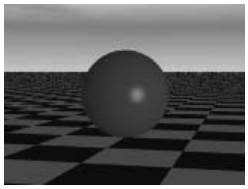
Directional Light Sources

- When the source is far away, we make the simplifying assumption that all rays of light from the source are parallel
  - A good approximation to sunlight
- The direction from a surface to the light source is important in lighting the surface
- With a directional light source, this direction is constant for all surfaces in the scene



Directional Light Sources (2)

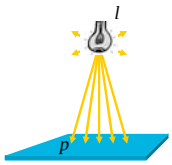
The same scene lit with a directional and an ambient light source



Point Light Sources

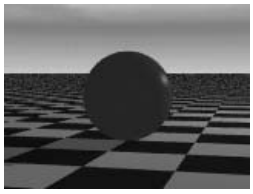
- A *point light source* emits light equally in all directions from a single point
- The direction to the light from a point on a surface thus differs for different points:
  - So we need to calculate a normalized vector to the light source for every point we light:

$$\overline{L} = \frac{p - l}{|p - l|}$$



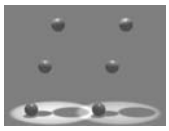
Point Light Sources (2)

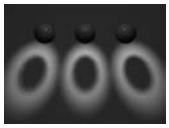
Using an ambient and a point light source



Other Light Sources

- Spotlights
  - has a cone of effect
  - intensity falls off away from a given direction
- Area Light Sources
  - Light source occupies a 2-D area (usually a polygon or disk)
  - Generates soft shadow. Why?





Physics of Diffuse Reflection

- Ideal diffuse reflection
  - Dull or matte surfaces
  - An *ideal diffuse reflector*, at the microscopic level, is a very rough surface (real-world example: chalk)
  - Because of these microscopic variations, an incoming ray of light is equally likely to be reflected in any direction over the hemisphere:

What does the reflected intensity depend on?

Question ?!

Why is North Pole cold?!

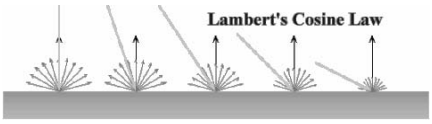
Why is winter cold?

Lambert's Cosine Law

- Ideal diffuse surfaces reflect according to *Lambert's cosine law*:

  The energy reflected by a small portion of a surface from a light source in a given direction is proportional to the *cosine of the angle between that direction and the surface normal*
- These are often called *Lambertian surfaces*
- Note that the reflected intensity is independent of the viewing direction, but does depend on the surface orientation with regard to the light source

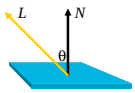
Lambert's Law



Computing Diffuse Reflection

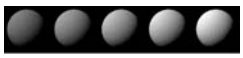
- The angle between the surface normal and the incoming light is the *angle of incidence:*

$$I_{diffuse} = k_d\, I_p\, \cos\theta$$



- In practice we use vector arithmetic:

$$I = k_d\, I_p\, (\boldsymbol{N} \bullet \boldsymbol{L})$$



k_d=0.4 k_d=1.0

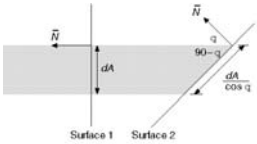
Diffuse Lighting Examples

- A Lambertian sphere seen at several different lighting angles:

- We need only consider angles from 0° to 90°. *Why?*

Physics of Diffuse Reflection

- The beam intercepts more surface area when the surface is at an angle to the beam, so there is less energy per small areas



Light Source Attenuation

- Currently we have come to this illumination equation:

$$Illumination = ambient + diffuse$$

$$I = I_a K_a + I_p K_d\, (N \,.\, L)$$

- Here, diffuse illumination depends on the diffuse-reflection coefficient (K_d) and orientation of surface (N)
- But objects look darker when they are <u>*farther*</u> from a light source
- Energy from a point light source that reaches a given part of a surface falls off by increasing <u>*distance*</u>
- So we introduce a light source attenuation factor f_{att}

$$I = I_a K_a + f_{att} I_p K_d\, (N \,.\, L)$$

Light Source Attenuation (2)

$$I = I_a K_a + f_{att} I_p K_d\, (N \,.\, L)$$

- Energy from a point light source that reaches a surface, falls off as the inverse square of the light distance:

$$f_{att} = \frac{1}{d_L^{\,2}}$$

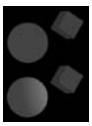
- In practice this often does not work well and the shading does not varies smoothly. so we use:

$$f_{att} = \min\left(\frac{1}{c_1 + c_2 d_L + c_3 d_L^{\,2}}, 1\right)$$

Colored Lights and Surfaces

- So far we have described monochromic lights and surfaces
- To have colored illumination, we should consider color of lights and color of surfaces
- Colored lights: use separate equation for each component of the color model:

$$I = (I_R, I_G, I_B) \quad , \quad I_a = (I_{aR}, I_{aG}, I_{aB}) \quad , \quad I_p = (I_{pR}, I_{pG}, I_{pB})$$



$$I_R = I_{aR} K_a + f_{att} I_{pR} K_d\, (N \,.\, L)$$
$$I_G = I_{aG} K_a + f_{att} I_{pG} K_d\, (N \,.\, L)$$
$$I_B = I_{aB} K_a + f_{att} I_{pB} K_d\, (N \,.\, L)$$

Colored Lights and Surfaces (2)

- Surfaces have different diffuse-reflection proportions for different color components of light. e.g.:
  - a blue surface, reflects blue component of a white light
  - a yellow surface reflects red and green components of a white light (absorbs blue)
- We can represent an object's *diffuse color* by a value O_d for each component: (O_{dR}, O_{dG}, O_{dB})

$$I_R = I_{aR}K_aO_{dR} + f_{att}I_{pR}K_dO_{dR}(N \cdot L)$$
$$I_G = I_{aG}K_aO_{dG} + f_{att}I_{pG}K_dO_{dG}(N \cdot L)$$
$$I_B = I_{aB}K_aO_{dB} + f_{att}I_{pB}K_dO_{dB}(N \cdot L)$$

Colored Lights and Surfaces (3)

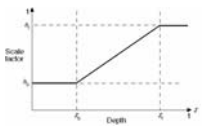
- We have used a single coefficient (K_a *and* K_d) for all equations
- This separation allows the user to be able to control the color of object without altering the coefficients (and vice versa)
- An alternative formulation that is more compact but less convenient to control uses separate coefficient for each component: (K_{aR}, K_{aG}, K_{aB}) and (K_{dR}, K_{dG}, K_{dB})
- For realistic illumination, we should use illumination equations that are wavelength-dependent:

$$I_\lambda = I_{a\lambda}K_aO_{d\lambda} + f_{att}I_{p\lambda}K_dO_{d\lambda}(N \cdot L)$$

Atmospheric Effects

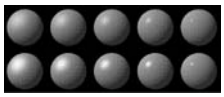
- More distant objects have lower intensity than have closer ones
- This effect depends on the atmosphere (clear, fog) and its color (regular fog, poisonous green gas!, …)
- To simulate atmospheric effects from the object to the viewer, we can employ *depth-cueing*
- We use a front and a back depth-cue reference planes each associated with a scale factor: s_f and s_b
- Scale factor determine the blending of the original intensity with that of a depth-cue color ($I_{dc\lambda}$):
- for a given z_o:

$$I'_\lambda = s_oI_\lambda + (1-s_o)I_{dc\lambda}$$



Specular Reflection

- Shiny surfaces exhibit *specular reflection*
  - Polished metal
  - Glossy car finish
- A light shining on a specular surface causes a bright spot known as a *specular highlight*
- These highlights appear is a function of the viewer's position, so specular reflectance is view-dependent

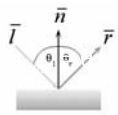


Physics of Specular Reflection

- At the microscopic level a specular reflecting surface is very smooth
- Thus rays of light are likely to bounce off the microgeometry in a mirror-like fashion
- The smoother the surface, the closer it becomes to a perfect mirror

Physics of Specular Reflection (2)

- Reflection follows *Snell's Laws*:
  - The incoming ray and reflected ray lie in a plane with the surface normal
  - The angle that the reflected ray forms with the surface normal equals the angle formed by the incoming ray and the surface normal:



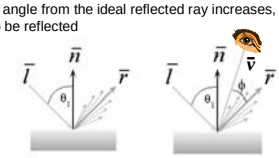
Physics of Specular Reflection (3)

- Snell's law applies to perfect mirror-like surfaces, but aside from mirrors (and chrome) few surfaces exhibit perfect specularity
- How can we capture the "softer" reflections of surfaces that are non-ideal specular reflectance i.e. are glossy rather than mirror-like?
- One option: model the microgeometry of the surface and explicitly bounce rays off of it
- Or…

  An empirical approximation!

Physics of Specular Reflection (4)

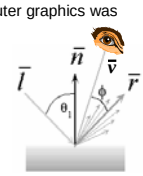
- An empirical approximation for non-ideal specular reflectance?
- Observations:
  - In general, we expect most reflected light to travel in direction predicted by Snell's Law
  - But because of microscopic surface variations, some light may be reflected in a direction slightly off the ideal reflected ray
  - As the angle from the ideal reflected ray increases, we expect less light to be reflected



Specular Reflection: Phong Lighting

- The most common lighting model in computer graphics was suggested by Phong:

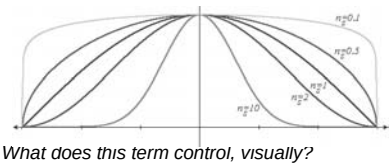
$$I_{specular} = k_s I_{light} (\cos\phi)^{n_{shiny}}$$



- The n_{shiny} term is a purely empirical constant that varies the rate of falloff
- Though this model has no physical basis, it works (sort of) in practice

Phong Lighting: The n_{shiny} Term

- This diagram shows how the Phong reflectance term drops off with divergence of the viewing angle from the ideal reflected ray:



- *What does this term control, visually?*

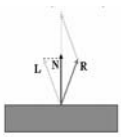
Calculating Phong Lighting

- The **cos** term of Phong lighting can be computed using vector arithmetic:

$$I_{specular} = k_s I_{light} \left(\hat{V} \cdot \hat{R}\right)^{n_{shiny}}$$

  - V is the unit vector towards the viewer
  - R is the ideal reflectance direction

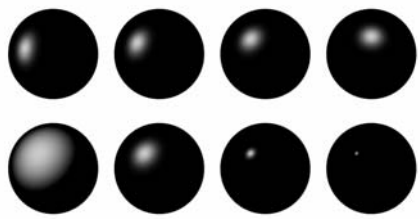
- We can efficiently calculate R



$$\hat{R} = \left(2\left(\hat{N} \cdot \hat{L}\right)\right)\hat{N} - \hat{L}$$

Phong Examples

- These spheres illustrate the Phong model as L and n_{shiny} are varied:



The Phong Lighting Model

- Let's combine ambient, diffuse, and specular components:

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

- Commonly called *Phong lighting*
  - Note: once per light
  - Note: once per color component
  - *Do k_a, k_d, and k_s vary with color component?*

Lighting Review

- Lighting Models
  - Ambient
    - Normals don't matter
  - Lambert/Diffuse
    - Angle between surface normal and light
  - Light attenuation and atmospheric effect
    - Distance matters
  - Phong/Specular
    - Surface normal, light, and viewpoint

Applying Illumination for Shading

- We now have an illumination model for a point on a surface
- Assuming that our surface is defined as a polygon, *which points should we use?*
- Shading involves computing a surface normal and then applying an illumination model to that point
- Doing this for each point on the surface would be very expensive, so there should be more efficient alternatives

Flat Shading

- Also called *constant shading* or *faceted shading*
- Is the simplest approach which calculates illumination at a single point for each polygon
- apply an illumination model once to determine a single intensity and then use it to shade the entire polygon

Question: *is this accurate?*

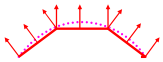
Flat Shading (2)

- *Question:* What If the object is really faceted?
- *Answer:* Again No!
  - For point sources, the direction to light varies across the facet
  - For specular reflectance, direction to eye varies across the facet
- So this approach is valid if:
  - The polygon represents the actual surface being modeled, and is not an approximation to a curved surface
  - The light source is at infinity so N.L is constant across the polygon face
  - The viewer is at infinity so N.V is constant across the polygon face

Flat Shading (3)

- We can refine it a bit by evaluating the Phong lighting model at each pixel of each polygon, but the result is still clearly faceted

- The problem is that if each polygonal facet in the mesh is shaded individually, it is easily distinguished from neighbors whose orientation is different



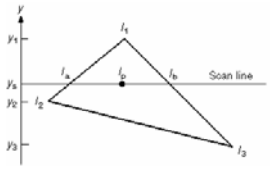
Vertex Normals

- To get smoother-looking surfaces we introduce *vertex normal* at each vertex
- Vertex normals may be
  - provided with the model
  - computed from first principles
  - approximated by averaging the normals of the facets that share the vertex

- So, vertex nomrals are
  - usually different from facet normal
  - used *only* for shading
  - think of as a better approximation of the *real* surface that the polygons approximate

Gouraud Shading

- This is the most common approach
  - Perform Phong lighting at the vertices
  - Linearly interpolate the resulting colors over faces



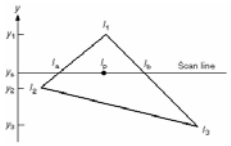
$$I_a = I_1 - (I_1 - I_2)\frac{y_1 - y_a}{y_1 - y_2}$$

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

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

Gouraud Shading (2)

- It requires knowledge of the normal for each vertex
- When this is not known, an averaging process can be used
- From these normals, intensities at the vertices are computed
- These intensities are interpolated along edges and then along scanlines
- just as z values can be interpolated!



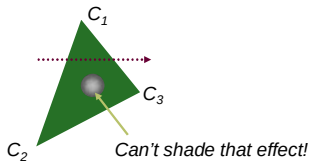
$$I_a = I_1 - (I_1 - I_2)\frac{y_1 - y_a}{y_1 - y_2}$$

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

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

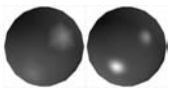
Gouraud Shading (3)

- Artifacts
  - Often appears dull, chalky
  - Lacks accurate specular component
  - Because colors will be averaged over entire polygon



Phong Shading

- Phong shading: linearly *interpolating the surface normal* across the facet, applying the Phong lighting model at every pixel
  - interpolation is done on surface normals not on the resulting colors (Gouraud shading)
  - Same input as Gouraud shading
  - Usually very smooth-looking results
  - But, considerably more expensive

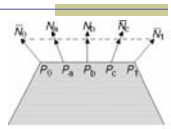


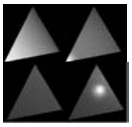
Gouraud Phong

- *NOTE: Phong shading* is not the same as Phong lighting, though they are sometimes mixed up
- Phong lighting is the empirical model we've been discussing to calculate illumination at a point on a surface

Phong Shading (2)

- Linearly interpolate the vertex normals
  - Compute lighting equations at each pixel
  - Can use specular component





Gouraud Phong

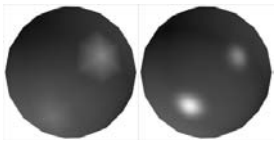
- Notice how highlights are more accurately reflected using the Phong shading model

Shortcomings of Shading

- polygonal silhouette
- perspective distortion
- orientation dependence
- problems at shared vertices
- unrepresentative vertex normals

Shortcomings of Shading (2)

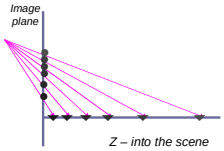
Polygonal silhouettes remain

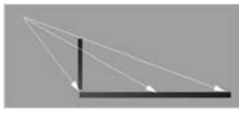


Gouraud *Phong*

Shortcomings of Shading (3)

Perspective Distortion: linear interpolation in screen space does not align with linear interpolation in world space

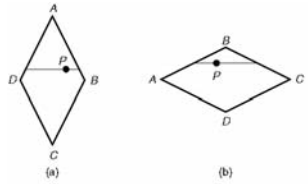




Break up large polygons with many smaller ones

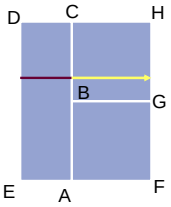
Shortcomings of Shading (4)

Interpolation dependent on polygon orientation



Shortcomings of Shading (5)

Problems at Shared Vertices

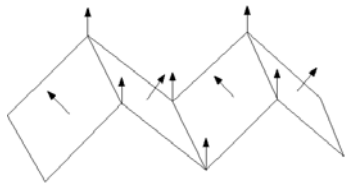


Vertex B is shared by the two rectangles on the right, but not by the one on the left

The first portion of the scanline is interpolated between DE and AC

The second portion of the scanline is interpolated between BC and GH

A large discontinuity could arise

Shortcomings of Shading (6)

unrepresentative vertex normals