Illumination Models & Shading
Lighting vs. Shading

- **Lighting**: Interaction between materials and light sources
  - Physics

- **Shading**: Determining the color of a pixel
  - Computer Graphics

- Shading is typically determined by lighting
  - Could visualizing the lighting function using *Non Photorealistic Rendering.*
Shading Models

- Simulate physical phenomena
  - Real illumination simulation is complicated & expensive
  - Use approximation and heuristics with little physical basis…
  - … that looks surprisingly good:
Local vs. Global Illumination Models

- **Local model** – direct and local interaction of each object location with the light.

- **Global model** – interactions and exchange of light energy between different objects.
Light Sources

- **Point source** (A): All light originates at a point
  - Rays hit a planar surface at different incidence angles

- **Parallel source** (B): All light rays are parallel
  - Rays hit a planar surface at identical incidence angles
  - May be modeled as a point source at infinity
  - Also denoted *Directional light source*

- **Area source** (C): Light originates at finite area in space.
  - An object of finite area that emits light
  - Also denoted *Distributed source*

**Question**: One of these lights Sources is far more difficult to Handle. Which one and why?
The Shading’s components

- **Specular reflection**
  - Metallic (smooth) surface
  - Reflects light at defined angle

- **Diffuse reflection**
  - Plastic (rough) surface
  - Reflects light in all directions

- **Ambient light**
  - Light reflected many times, comes equally from all directions
**Ambient Light**

- Assume non-directional light in the environment
- Object illuminated with same light everywhere
  - Looks like silhouette

- **The Illumination equation** \( I = I_a k_a \)
  - \( I_a \) - ambient light intensity
  - \( k_a \) - fraction of ambient light reflected from surface
  - As a vector, also defines object color
Diffuse Light

- Dull surfaces such as solid matte plastic reflects incoming light uniformly in all directions.
- This is called diffuse or Lambertian reflection.
- For light source in a normalized direction $L$ and a surface with normal $N$, the illumination of the surface is proportional to $\langle N, L \rangle$.
Diffuse Reflection

- Illumination equation is now:

\[ I = I_a k_a + I_p k_d \langle N, L \rangle = I_a k_a + I_p k_d \cos \theta \]

- \( I_p \) - point light source’s intensity
- \( k_d \) - surface diffuse reflection coefficient

**Question**: Can we locate the light source from the shading?
Specular Reflection

- Shiny objects (e.g. metallic) reflect light in a preferred direction \( R \) determined by the surface normal \( N \).

- Most objects are not ideal mirrors – also reflect in the immediate vicinity of \( R \).

- **Phong Model** – approximate attenuation by the form of \( \cos^n \alpha \) (no real physical basis).

**Question**: What is the color of the reflected component?
Specular Reflection (Phong Model)

- Illumination equation:

\[ I = I_a k_a + I_p \left( k_d (N \cdot L) + k_s (R \cdot V)^n \right) \]

- \( k_s \) - Specular reflection coefficient
- \( n \) - Specularity exponent
Specular Reflection (cont’d)

- Exponent $n$ of cosine controls the decay factor of the attenuation function:

- Again, no physical basis but it does look good:
The complete illumination model is hence:

\[ I = I_a k_a + I_p \left( k_d (N \cdot L) + k_s (R \cdot V)^n \right) \]
More on Illumination Equation

- For multiple light sources:

\[ I_p = \sum I_{p_i} \left( k_d (N \cdot I_{p_i}) + k_s (R_p \cdot V)^n \right) \]

- \( I_p \) of all light sources are added together
- Precautions should be taken from overflows

**Question**: How can we achieve atmospheric attenuation effects?
Even More on Illumination Equation

For distance/atmospheric attenuation sources:

\[ I = I_a k_a + \int \frac{I_p}{d_p} \left( k_d (N \cdot L_p) + k_s (R_p \cdot V)^n \right) \]

\(d_p\) - distance between surface and light source and/or distance between surface and viewer (Heuristic atmospheric attenuation)

**Question**: why an attenuation of \(1/d\) and not the physically correct decay (which is!?)
Flat Shading

- Applied to piecewise linear polygonal models
- Simple surface lighting approximated over polygons
- Illumination value depends only on polygon normal $\Rightarrow$ each polygon is colored with a uniform intensity
- Looks non-smooth (worsened by “Mach bands” effects)
Flat Shading
Gouraud Shading

- If a polyhedron is an approximation of smooth surface:
  - assign to each vertex the normal of original surface at that point
  - If surface is not available, use estimated normal (how?)

- Compute illumination intensity at vertices using those normals

Question: And then what?
Gouraud Shading

Linearly interpolate lighting intensities at the vertices over interior pixels of the polygon, in the image plane.

**Question:** Can Gouraud shading support specular lighting?
Gouraud Shading
Phong Shading

- Interpolate (at the vertices in image space) normal vectors instead of illumination intensities
- Apply the illumination equation for each interior pixel with its own (interpolated) normal
Gouraud Shading a Triangle
Comments on Shading

- Phong shading is clearly more expensive (why?) but well worth the effort (yet, with no Open GL support)

- Can achieve good looking specular highlight effects

- Both the Gouraud and Phong shading schemes are performed in the image plane and fit well into our polygonal scan-conversion fill scheme

- Both the Gouraud and Phong are view dependent

- Can cause artifacts during animation as they are transformation dependent
More Examples

Flat  Gouraud  Phong