Light Sources

- ambient lights
  - no identifiable source or direction
  - hack for replacing true global illumination
  - (diffuse interreflection: light bouncing off from other objects)

Diffuse Interreflection

- scene lit only with an ambient light source

Ambient Light Sources

- scene lit with directional and ambient light

Directional Light Sources

- area lights
  - light sources with a finite area
  - more realistic model of many light sources
  - not available with projective rendering pipeline (i.e., not available with OpenGL)
Point Light Sources
• scene lit with ambient and point light source

Light Sources
• geometry: positions and directions
• standard: world coordinate system
• effect: lights fixed wrt world geometry
• alternative: camera coordinate system
• effect: lights attached to camera (car headlights)
• points and directions undergo normal model/view transformation
• illumination calculations: camera coords

Types of Reflection
• specular (a.k.a. mirror or regular) reflection causes light to propagate without scattering.
• diffuse reflection sends light in all directions with equal energy.
• mixed reflection is a weighted combination of specular and diffuse.

Diffuse Lighting Examples
• Lambertian sphere from several lighting angles:
  - need only consider angles from 0° to 90°
  - why?
  - demo: Brown exploratory on reflection

Surface Roughness
• at a microscopic scale, all real surfaces are rough
• cast shadows on themselves
• “mask” reflected light:

Optics of Reflection
• reflection follows Snell’s Law: incoming ray and reflected ray lie in a plane with the surface normal
• angle the reflected ray forms with surface normal equals angle formed by incoming ray and surface normal

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

Lambert’s Cosine Law
• ideal diffuse surface reflection
  - 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
  - reflected intensity
  - independent of viewing direction
  - depends on surface orientation wrt light
• often called Lambertian surfaces

Reflectance Distribution Model
• most surfaces exhibit complex reflectances
• vary with incident and reflected directions
• model with combination

Surface Roughness
• notice another effect of roughness:
  - each “microfacet” is treated as a perfect mirror.
  - incident light reflected in different directions by different facets.
  - end result is mixed reflectance.
  - smoother surfaces are more specular or glossy.
  - random distribution of facet normals results in diffuse reflectance.

Physics of Diffuse Reflection
• ideal diffuse reflection
  - very rough surface at the microscopic level
  - real-world example: chalk
  - microscopic variations mean incoming ray of light equally likely to be reflected in any direction over the hemisphere.
• what does the reflected intensity depend on?

Lambert’s Law
• depends on angle of incidence: angle between surface normal and incoming light

Specular Highlights
• 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

Types of Reflection
• retro-reflection occurs when incident energy reflects in directions close to the incident direction, for a wide range of incident directions.
• gloss is the property of a material surface that involves mixed reflection and is responsible for the mirror like appearance of rough surfaces.

Optics of Reflection
• reflection follows Snell’s Law: incoming ray and reflected ray lie in a plane with the surface normal
• angle the reflected ray forms with surface normal equals angle formed by incoming ray and surface normal

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

Lambert’s Cosine Law
• ideal diffuse surface reflection
  - 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
  - reflected intensity
  - independent of viewing direction
  - depends on surface orientation wrt light
• often called Lambertian surfaces

Reflectance Distribution Model
• most surfaces exhibit complex reflectances
• vary with incident and reflected directions
• model with combination

Surface Roughness
• at a microscopic scale, all real surfaces are rough
• cast shadows on themselves
• “mask” reflected light:

Optics of Reflection
• reflection follows Snell’s Law: incoming ray and reflected ray lie in a plane with the surface normal
• angle the reflected ray forms with surface normal equals angle formed by incoming ray and surface normal

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

Lambert’s Cosine Law
• ideal diffuse surface reflection
  - 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
  - reflected intensity
  - independent of viewing direction
  - depends on surface orientation wrt light
• often called Lambertian surfaces

Reflectance Distribution Model
• most surfaces exhibit complex reflectances
• vary with incident and reflected directions
• model with combination

Surface Roughness
• at a microscopic scale, all real surfaces are rough
• cast shadows on themselves
• “mask” reflected light:
Non-Ideal Specular Reflectance
- Snell’s law applies to perfect mirror-like surfaces, but aside from mirrors (and chrome) few surfaces exhibit perfect specularly.
- how can we capture the “softer” reflections of surfaces that are glossy, not mirror-like?
- one option: model the microgeometry of the surface and explicitly bounce rays off of it
- or…

Light Source Falloff
- quadratic falloff
- brightness of objects depends on power per unit area that hits the object
- the power per unit area for a point or spot light decreases quadratically with distance

Phong Lighting
- most common lighting model in computer graphics
- (Phong Bui-Tuong, 1975)
- \( I_{\text{specular}} = k_s I_{\text{light}} (\cos \phi)^{n_{\text{shiny}}} \)
- \( n_{\text{shiny}} \): purely empirical constant, varies rate of falloff
- \( k_s \): specular coefficient, highlight color
- no physical basis, works ok in practice

Lighting Review
- lighting models
  - ambient
  - normals don’t matter
  - Lambert/diffuse
  - angle between surface normal and light
  - Phong/specular
  - surface normal, light, and viewpoint

Lighting in OpenGL
- light source: amount of RGB light emitted
- value represents percentage of full intensity e.g., \((0.1,0.5,0.0)\)
- every light source emits ambient, diffuse, and specular light
- materials: amount of RGB light reflected
- value represents percentage reflected e.g., \((0.2,0.3,0.5)\)
- interaction: multiply components
  - red light \((1,0,0)\) x green surface \((0,1,0)\) = black \((0,0,0)\)
Lighting in OpenGL

```c
glLightfv(GL_LIGHT0, GL_AMBIENT, amb_light_rgba);
glLightfv(GL_LIGHT0, GL_DIFFUSE, dif_light_rgba);
glLightfv(GL_LIGHT0, GL_SPECULAR, spec_light_rgba);
glLightfv(GL_LIGHT0, GL_POSITION, position);
glMaterialfv(GL_FRONT, GL_SHININESS, n);
glEnable(GL_LIGHT0);
glLightfv(GL_LIGHT0, GL_SPECULAR, spec_light_rgba);
glLightfv(GL_LIGHT0, GL_DIFFUSE, dif_light_rgba);
glLightfv(GL_LIGHT0, GL_AMBIENT, amb_light_rgba);
```

Lighting vs. Shading

- **lighting**
  - process of computing the luminous intensity (i.e., outgoing light) at a particular 3-D point, usually on a surface
- **shading**
  - the process of assigning colors to pixels

... (Continues with diagrams and explanations)

Applying Illumination

- polygonal/triangular models
  - each facet has a constant surface normal
  - if light is directional, diffuse reflectance is constant across the facet
  - why?

Flat Shading

- simplest approach calculates illumination at a single point for each polygon
- obviously inaccurate for smooth surfaces

Flat Shading Approximations

- if an object really is faceted, is this accurate?
- no!
  - for point sources, the direction to light varies across the facet
  - for specular reflectance, direction to eye varies across the facet

Improving Flat Shading

- what if evaluate Phong lighting model at each pixel of the polygon?
  - better, but result still clearly faceted
  - for smoother-looking surfaces we introduce *vertex normals* at each vertex
  - usually different from facet normal
  - used *only* for shading
  - think of as a better approximation of the real surface that the polygons approximate

Vertex Normals

- vertex normals may be
  - provided with the model
  - computed from first principles
  - approximated by averaging the normals of the facets that share the vertex

Gouraud Shading

- most common approach, and what OpenGL does
  - perform Phong lighting at the vertices
  - linearly interpolate the resulting colors over faces
  - along edges
  - along scanlines
  - does this eliminate the facets?

Gouraud Shading Artifacts

- often appears dull, chalky
- lacks accurate specular component
  - if included, will be averaged over entire polygon

Phong Shading

- perspective transformation problem
  - colors slightly "swim" on the surface as objects move relative to the camera
  - usually ignored since often only small difference
  - usually smaller than changes from lighting variations
  - to do it right
    - either shading in object space
    - or correction for perspective foreshortening
  - expensive – thus hardly ever done for colors

... (Continues with diagrams and explanations)
Phong Shading
- linearly interpolate the vertex normals
- compute lighting equations at each pixel
- can use specular component
\[
I_{\text{total}} = k_dI_{\text{diffuse}} + (k_l + k_r)I_{\text{specular}}
\]
where normals used in diffuse and specular terms

Phong Shading Difficulties
- computationally expensive
- per-pixel vector normalization and lighting computation!
- floating point operations required
- lighting after perspective projection
- messes up the angles between vectors
- have to keep eye-space vectors around
- no direct support in pipeline hardware
- but can be simulated with texture mapping
- stay tuned for modern hardware: shaders

Shading Models Summary
- flat shading
  - compute Phong lighting once for entire polygon
- Gouraud shading
  - compute Phong lighting at the vertices and interpolate lighting values across polygon
- Phong shading
  - compute averaged vertex normals
  - interpolate normals across polygon and perform Phong lighting across polygon

Shading Artifacts: Shared Vertices
- first portion of the scanline is interpolated between DE and AC
- second portion of the scanline is interpolated between BC and GH
- a large discontinuity could arise

Shading Artifacts: Silhouettes
- polygonal silhouettes remain

Shading Artifacts: Orientation
- interpolation dependent on polygon orientation
- view dependence!

Computing Normals
- per-vertex normals by interpolating per-facet normals
- OpenGL supports both
- computing normal for a polygon
  - three points form two vectors
  - cross: normal of plane gives direction
  - normalize to unit length!
  - which side is up?
    - convention: points in counterclockwise order

Computing Normals
- OpenGL state machine
  - uses last normal specified
  - if no normals specified, assumes all identical
- per-vertex normals
  - per-face normals
    - OpenGL supports both