Topic 8:

Lighting & Reflection models

- Lighting & reflection
- The Phong reflection model
  - diffuse component
  - ambient component
  - specular component
Logistics

• Welcome back
• Professor Singh is away for the next 3 lectures (including this one).
  • If you need something desperately contact me
  • diwlevin@cs.toronto.edu
• You should have your midterm marks (emailed to UT email)
• We will release solutions to the midterm
• Assignment 2 due March 9th
• Assignment 3 will be available roughly the same time
• Midterm, A1, A2 TA office hours
  • Thursday March 1st 2pm-3pm
  • Friday March 2nd 3pm-4pm
Spot the differences
Illumination

• The transport of luminous flux from light sources between points via direct and indirect paths

Lighting

• The process of computing the luminous intensity reflected from a specified 3-D point

Shading

• The process of assigning a color to a pixel

Illumination Models

• Simple approximations of light transport
• Physical models of light transport
Two Components of Illumination

**Light Sources**

- Emission Spectrum (color)
- Geometry (position and direction)
- Directional Attenuation

**Surface Properties** (Reflectors)

- Reflectance Spectrum (color)
- Geometry (position, orientation, and micro-structure)
- Absorption
- Transmission
Main sources of light:

- Point source
- Directional Light
- Spotlight
Light Source Types

Point Light
- light originates at a point

Directional Light (point light at infinity)
- light rays are parallel
- Rays hit a planar surface at identical angles

Spot Light
- point light with limited angles

Bessmeltsev et al.
Point Light

- light originates at a point
- defined by location only

Directional Light (point light at infinity)

- light rays are parallel
- Rays hit a planar surface at identical angles
- defined by direction only

Spot Light

- point light with limited angles
- defined by location, direction, and angle range
Point Light Sources

The point light source emits rays in radial directions from its source. A point light source is a fair approximation to a local light source such as a light bulb.

The direction of the light to each point on a surface changes when a point light source is used. Thus, a normalized vector to the light emitter must be computed for each point that is illuminated.
Directional Light Sources

All of the rays from a directional light source have a common direction, and no point of origin. It is as if the light source was infinitely far away from the surface that it is illuminating. Sunlight is an example of an infinite light source.

The direction from a surface to a light source is important for computing the light reflected from the surface. With a directional light source this direction is a constant for every surface. A directional light source can be colored.
Other Light Sources

Spotlights

• Point source whose intensity falls off away from a given direction
• Requires a color, a point, a direction, parameters that control the rate of fall off

Area Light Sources

• Light source occupies a 2-D area (usually a polygon or disk)
• Generates *soft* shadows

Extended Light Sources

• Spherical Light Source
• Generates *soft* shadows
Area Light Source: Direct Lighting
Area Light Source: Indirect Lighting
Even though an object in a scene is not directly lit it will still be visible. This is because light is reflected indirectly from nearby objects. A simple hack that is commonly used to model this indirect illumination is to use of an ambient light source. Ambient light has no spatial or directional characteristics. The amount of ambient light incident on each object is a constant for all surfaces in the scene. An ambient light can have a color.

The amount of ambient light that is reflected by an object is independent of the object's position or orientation. Surface properties are used to determine how much ambient light is reflected.
The Common Modes of “Light Transport”

- Specular reflection
- Surface scattering
- Transmission
- Incident light
- Light source

Material
Two Types of Surface Reflection

1. Diffuse Reflection
2. Specular Reflection
Diffuse reflection:
- Represents "matte" component of reflected light
- Usually cause by "rough" surfaces (clay, eggshell, etc)
Specular reflection:
• Represents shiny component of reflected light
• Caused by mirror like reflection off of smooth or polished surfaces (plastics, polished metals, etc)
Specular reflection:
• Represents shiny component of reflected light
• Caused by mirror like reflection off of smooth or polished surfaces (plastics, polished metals, etc)
Specular reflection:
• Represents shiny component of reflected light
• Caused by mirror like reflection off of smooth or polished surfaces (plastics, polished metals, etc)
Specular reflection:
• Represents shiny component of reflected light
• Caused by mirror like reflection off of smooth or polished surfaces (plastics, polished metals, etc)
Specular reflection:
- Represents shiny component of reflected light
- Caused by mirror like reflection off of smooth or polished surfaces (plastics, polished metals, etc)
Transmission:
• Caused by materials that are not perfectly opaque
• Examples include glass, water and translucent materials such as skin
Subsurface scattering:

- Represents the component of reflected light that scatters in the material's interior (after transmission) before exiting again.
- Examples include skin, milk, fog, etc.
Rendering with no subsurface scattering (opaque skin)
Rendering with subsurface scattering (translucent skin)
Rendering with no subsurface scattering (opaque milk)
Rendering with subsurface scattering (full milk)

Jensen et al., SIGGRAPH'01
Rendering with subsurface scattering (skim milk)

Jensen et al., SIGGRAPH '01
The Common Modes of “Light Transport”

- Light source
- Incident light
- Specular reflection
- Surface scattering
- Subsurface scattering
- Transmission
The Phong Reflectance Model

Phong model: A simple computationally efficient model that has 3 components:
- Diffuse
- Ambient
- Specular
The Phong Reflectance Model

Phong model: A simple computationally efficient model that has 3 components:
- Diffuse
- Ambient
- Specular
Phong Reflection: The Diffuse Component

- A diffuse point looks the same from all viewing positions
- Simplest case: a single, point light source
A diffuse point looks the same from all viewing positions.
Simplest case: a single, point light source.
Ideal diffuse reflectors reflect light according to *Lambert's cosine law*, Lambert's law states that the reflected energy from a small surface area in a particular direction is proportional to cosine of the angle between that direction and the surface normal.
The Diffuse Component: Basic Equation

- A diffuse point looks the same from all viewing positions
- Simplest case: a single, point light source

\[
I_P = r_d \cdot I \cdot \max \left( 0, \frac{\mathbf{S} \cdot \mathbf{n}}{\| \mathbf{l} - \mathbf{P} \|} \right)
\]

- \( r_d \) is the fraction of light reflected
- \( I \) is the intensity of the light source
- \( \mathbf{S} \) is the direction of the light source
- \( \mathbf{P} \) is the projection of the point \( P \)
- \( \mathbf{n} \) is the outward unit surface normal
- \( \| \mathbf{l} - \mathbf{P} \| \) is the distance from the light source to the projection of \( P \)
The Diffuse Component: Basic Equation

- A diffuse point looks the same from all viewing positions

\[
\mathbb{I}_{\overrightarrow{p}} = r_d \cdot I \cdot \max \left( \mathbf{0}, \overrightarrow{S} \cdot \overrightarrow{n} \right)
\]

- independent of \( \overrightarrow{C} \)
- outward unit surface normal
- intensity at projection of \( \overrightarrow{p} \)
- direction of light source
- \( \overrightarrow{S} = \frac{\overrightarrow{l} - \overrightarrow{p}}{||\overrightarrow{l} - \overrightarrow{p}||} \)
The Diffuse Component: Foreshortening

As the angle $\theta_i$ between $\vec{s}$ and $\vec{n}$ increases, the area of the surface around $\vec{p}$ receiving light increases. Therefore, the light intensity received per unit area decreases. This is called foreshortening. Consequently, point $\vec{p}$ will appear dimmer.

\[ I_p = r_d \cdot I \cdot \max \left( 0, \vec{s} \cdot \vec{n} \right) \]

accounts for dimming due to foreshortening.
The Diffuse Component: Foreshortening

As the angle $\theta_i$ between $\vec{s}$ and $\vec{n}$ increases, the area of the surface around $\vec{p}$ receiving light increases.

$\Rightarrow$ the light intensity received per unit area decreases.

This is called foreshortening.

$\Rightarrow$ point $\vec{p}$ will appear dimmer.

Q: What is the intensity at $\vec{p}$'s projection?

$I_{\vec{p}} = r_d \cdot I \cdot \max(0, \vec{s} \cdot \vec{n})$

accounts for dimming due to foreshortening.
The Diffuse Component: Foreshortening

As the angle $\theta$ between $\vec{s}$ and $\vec{n}$ increases, the area of the surface around $\vec{p}$ receiving light increases. Therefore, the light intensity received per unit area decreases. This is called foreshortening. Hence, point $\vec{p}$ will appear dimmer.

$\vec{p}$: What is the intensity at $\vec{p}$'s projection?

$I_\vec{p} = r_d \cdot I \cdot \max \left( 0, \frac{\vec{s} \cdot \vec{n}}{\| \vec{s} \| \cdot \| \vec{n} \|} \right)$

accounts for dimming due to foreshortening.
The Diffuse Component: Foreshortening

As the angle $\theta_i$ between $\mathbf{s}$ and $\hat{n}$ increases, the area of the surface around $\bar{p}$ receiving light increases. 

$\Rightarrow$ the light intensity received per unit area decreases. 

this is called foreshortening.

$\Rightarrow$ point $\bar{p}$ will appear dimmer.

**Q:** What is the intensity at $\bar{p}$'s projection?

$$I_{\bar{p}} = r_d \cdot I \cdot \max \left( 0, \mathbf{s} \cdot \hat{n} \right)$$

accounts for dimming due to foreshortening.
The Diffuse Component: Self-Shadowing

As the angle $\theta_i$ between $\mathbf{S}$ and $\mathbf{n}$ increases, the area of the surface around $\mathbf{P}$ receiving light increases. 

$\Rightarrow$ the light intensity received per unit area decreases. 

this is called foreshortening.

$\Rightarrow$ point $\mathbf{P}$ will appear dimmer.

Question: What is the intensity at $\mathbf{P}$'s projection?

$I_p = 0$

$I_p = r_d \cdot I \cdot \max\left(0, \mathbf{S} \cdot \mathbf{n}\right)$

accounts for cases where light source not visible.
A diffuse point looks the same from all viewing positions.

When the scene is illuminated by many point sources, we just sum up their contributions to the diffuse component.

\[ I_{\vec{p}} = r_d \sum_i I_i \max(0, \vec{s}_i \cdot \vec{n}) \]

- intensity at projection of \( \vec{p} \)
- intensity of source \( i \)
The Diffuse Component: Incorporating Color

- A diffuse point looks the same from all viewing positions
- Coloured sources and coloured objects are handled by considering the RGB components of each colour separately

\[ I_{\bar{p}, q} = r_{d, q} \sum_i I_{i, q} \max(0, \bar{s}_i \cdot \bar{n}) \quad q = R, G, B \]
Putting it all together:

\[ I_{P,q} = r_{d,q} \sum_i I_{i,q} \max(0, \mathbf{s}_i \cdot \mathbf{n}) \]
When we look at a shiny surface, such as polished metal, we see a highlight, or bright spot. Where this bright spot appears on the surface is a function of where the surface is seen from. The reflectance is view dependent.
The Ideal Specular Component

- Idea: For each incident reflection direction, there is one emittent direction.
- It is an idealization of a mirror:

\[
\text{angle}(\vec{n}, \vec{s}) = \text{angle}(\vec{n}, \vec{r})
\]

\[
\theta_i \quad \theta_r
\]
The Ideal Specular Component

Romeiro et al, ECCV’08

- Idea: For each incident reflection direction, there is one emittent direction
- It is an idealization of a mirror:

\[ \text{angle}(\vec{n}, \vec{s}) = \text{angle}(\vec{n}, \vec{r}) \]

\[ \theta_i \quad \theta_r \]

Q: How can we express \( \vec{r} \) in terms of \( \vec{n}, \vec{s} \)?
The Ideal Specular Component

\[ \vec{r} = -\vec{s} + 2\vec{t} \]

\[ \vec{t} = \text{projection of vector } \vec{s} \text{ onto vector } \vec{n} \]

\[ = (\vec{n} \cdot \vec{s}) \vec{n} \]

\[ \Rightarrow \vec{r} = -\vec{s} + 2(\vec{n} \cdot \vec{s}) \vec{n} \]

Q: How can we express \( \vec{r} \) in terms of \( \vec{n}, \vec{s} \)?
The Ideal Specular Component

Ideal specular reflection term:

\[ I = I_s \ I_s \ \delta \left( \vec{r} \cdot \vec{b} \cdot \hat{c} - 1 \right) \]

where \( \delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases} \)
The Ideal Specular Component

Ideal specular reflection term:

\[ I = \mathcal{I}_S \mathcal{I}_L \delta \left( \vec{r} \cdot \vec{b} - 1 \right) \]

where \( \delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases} \)
The Ideal Specular Component

Ideal specular reflection term:

\[ I = I_s \ I_s \ \delta \left( \frac{\vec{r} - \vec{b} - \vec{1}}{2} \right) \]

where \( \delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases} \)

- \( I_s \): Specular reflection coefficient
- \( I_s \): Intensity of specular light source
- \( \vec{1} \): Unit vector in camera direction
- \( \vec{b} \): Unit vector in light source direction
- \( \vec{c} \): Surface normal
- \( \vec{r} \): Ray direction

\( \Rightarrow \vec{b} = \frac{\vec{c} - \vec{p}}{||\vec{c} - \vec{p}||} \)
Phong Reflection: Off-Specular Reflection

Brad Smith, Wikipedia

Panjasan, Wikipedia
In reality, most specular surfaces reflect light into directions near the perfect direction (e.g. highlights in plastics, metals)

→ Introduce cosine power

\[ I = \nu_s I_s \max \left( 0, \vec{r} \cdot \vec{b} \right)^\alpha \]

\[ \approx 1 \text{ when } \alpha \to \infty \text{ term approaches ideal specular reflection term} \]
The length of vector \((\vec{r} \cdot \vec{s})^\alpha \vec{b}\) represents the contribution of the specular term when the camera is along \(\vec{b}\).

\[
I = r_s \ I_s \ \max (0, \ (\vec{r} \cdot \vec{s})^\alpha) = 1 \ \text{when} \ \vec{r} = \vec{b}
\]
"Area Light Source, Direct Lighting"

"hard" shadow: points not visible from light source

"soft" shadows: shadows created because points visible from part of area light source
Phong Reflection: Ambient Component

- Solution#2: (simpler) Use an "ambient" term that is independent of any light source or surface normal.
- This term is not meaningful in terms of physics but improves appearance over pure diffuse reflection.

\[
\mathbf{I}_p = \sqrt{a} \cdot \mathbf{I}_a
\]

- Intensity of ambient illumination
- Ambient reflection coefficient (often \( r_a = r_d \))
- Can also have 3 such eqs for \( R, G, B \) components

- Diffuse reflectance with a single point light source produces strong shadows
- Surface patches with are perfectly black
  - \( \mathbf{s} \cdot \mathbf{n} < 0 \)
  - Looks unnatural
Phong Reflection: The General Equation

\[
L(b, n, s) = r_a I_a + r_d I_d \max(0, n \cdot s) + r_s I_s \max(0, n \cdot b)^p
\]

- **ambient** intensity at projection of point \(P\)
- **diffuse**
- **specular**
Phong Reflection: The General Equation

\[
L(\vec{b}, \vec{n}, \vec{s}) = r_a I_a + r_d I_d \max(0, \vec{n} \cdot \vec{s}) + r_s I_s \max(0, \vec{n} \cdot \vec{b})^\alpha
\]

- **Ambient**
- **Diffuse**
- **Specular**

Brad Smith, Wikipedia
Computing Diffuse Reflection

The angle between the surface normal and the incoming light ray is called the angle of incidence.

$I_{\text{light}}$ : intensity of the incoming light.

$k_d$ : represents the diffuse reflectivity of the surface at that wavelength.

What is the range of $k_d$

$$I_{\text{diffuse}} = k_d I_{\text{light}} \cos \theta$$
Where do we Illuminate?

To this point we have discussed how to compute an illumination model at a point on a surface.

Which points on the surface is the illumination model applied?

Illumination can be costly...
Topic 10: Shading

- Introduction to Shading
- Flat Shading
- Interpolative Shading
  - Gouraud shading
  - Phong shading
- Triangle scan-conversion with shading
Shading: Motivation

- Suppose we know how to compute the appearance of a point.
- How do we shade a whole polygon mesh?

Answer: Assign intensities to every pixel at the mesh's projection in accordance with Phong reflection model.
Shading: Motivation

• Suppose we know how to compute the appearance of a point.
• How do we shade a whole polygon mesh?

Answer:
Assign intensities to every pixel at the mesh's projection in accordance with Phong reflection model.

\[ L(\vec{b}, \vec{n}, \vec{s}) = r_a I_a + r_d I_d \max(0, \vec{n} \cdot \vec{s}) + r_s I_s \max(0, \vec{s} \cdot \vec{b}) \]

- \( I_a \): ambient intensity at point \( \vec{p} \)
- \( I_d \): diffuse intensity
- \( I_s \): specular intensity
- \( \vec{b} \): light source direction
- \( \vec{n} \): surface normal
- \( \vec{s} \): viewing direction
Shading: Problem Definition

Given

- camera center, \( \vec{C} \)
- light source position, \( \vec{I} \)
- intensity of ambient, diffuse and specular sources,
- reflection coefficients, \( r_{\alpha}, r_{d}, r_{s} \)
- specular exponent, \( \alpha \)
- normals at \( \vec{p}_{1}, \vec{p}_{2}, \vec{p}_{3} \)

Goal

Computer colour/intensity at an interior pixel

\[
L(\vec{b}, \vec{n}, \vec{s}) = r_{a} I_{\alpha} + r_{d} I_{d} \max(0, \vec{n} \cdot \vec{s}) + r_{s} I_{s} \max(0, \vec{r} \cdot \vec{b})^{\alpha}
\]

intensity at projection of point \( \vec{p} \)
ambient
diffuse
specular
Flat Shading: Main Idea

Flat shading
Draw all triangle points $\vec{P}$ with identical colour/intensity

Sphere with flat shading

Jalo, Wikipedia

\[
L(b, n, s) = r_a I_a + r_d I_d \max(0, n \cdot s) + r_s I_s \max(0, n \cdot b)^x
\]
Flat Shading: Key Issues

Flat shading
Draw all triangle points \( \vec{P} \) with identical colour/intensity

Issues:
- For large triangles:
  - Specular term is poor approximation because highlight should be sharp (often better to drop this term)
  - flat shading essentially assumes a distant light source
- Triangle boundaries are usually visible (people very sensitive to intensity steps)

\[
L(\vec{b}, \vec{n}, \vec{s}) = r_a I_a + r_d I_d \max(0, \vec{n} \cdot \vec{s}) + r_s I_s \max(0, \vec{s} \cdot \vec{b})^\alpha
\]

- intensity at projection of point \( \vec{P} \)
- ambient
- diffuse
- specular
Flat Shading: Key Issues

Flat shading
Draw all triangle points $\vec{P}$ with identical colour/intensity

Issues:
- For large triangles:
  - Specular term is poor approximation because highlight should be sharp (often better to drop this term)
  - flat shading essentially assumes a distant light source
- Triangle boundaries are usually visible (people very sensitive to intensity steps)

One solution
- Since flat shading treats a triangle as a point, use small triangles!
Interpolated Shading

FLAT SHADING

PHONG SHADING

Jalo, Wikipedia
Interpolative Shading: Basic Approaches

Gouraud shading
1. Compute $L_i = L(b_i, n_i, s_i)$ for each vertex
2. Interpolate the $L_i$'s to get the value at $\vec{p}$

Phong shading
1. Interpolate $\vec{b}_i, \vec{n}_i, \vec{s}_i$ to get $\vec{b}, \vec{n}, \vec{s}$ at $\vec{p}$
2. Compute $L(\vec{b}, \vec{n}, \vec{s})$

$$L(b, n, s) = r_a I_a + r_d I_d \max(0, n, s) + r_s I_s \max(0, n, b)^\alpha$$

- $I_a$: Ambient intensity at the projection of point $\vec{p}$
- $I_d$: Diffuse intensity
- $I_s$: Specular intensity
Gouraud Shading: Computation at Vertices

1. Compute $L_i = L(b_i, n_i, s_i)$ for each vertex
2. Interpolate the $L_i$’s to get the value at $\bar{p}$

Notes
- Vectors $b_i, s_i$ computed directly from $\bar{p}_i, \bar{c}$ and $\bar{l}$
- Many possible ways to assign a normal to a vertex

\[
L(b_i, n_i, s_i) = r_a I_a + r_d I_d \max(0, n_i \cdot s_i) + r_s I_s \max(0, \bar{p} \cdot \bar{b}_i)^\alpha
\]
Gouraud Shading: Computation at Vertices

1. Compute $L_i = L(b_i, \vec{n}_i, \vec{s}_i)$ for each vertex
2. Interpolate the $L_i$’s to get the value at $\vec{p}$

Notes
- Vectors $\vec{b}_i, \vec{s}_i$ computed directly from $\vec{p}_i, \vec{c}$ and $\vec{l}$
- Many possible ways to assign a normal to a vertex

1. $\vec{n}_j$ is the average of the normals of all faces that contain vertex $\vec{p}_j$
Gouraud Shading: Computation at Vertices

1. Compute $L_i = L(\vec{b}_i, \vec{n}_i, \vec{s}_i)$ for each vertex
2. Interpolate the $L_i$’s to get the value at $\vec{p}$

Notes
- Vectors $\vec{b}_i, \vec{s}_i$ computed directly from $\vec{p}_i, \vec{c}$ and $\vec{l}$
- Many possible ways to assign a normal to a vertex

$\vec{n}_j$ is the normal of a point sample on a parametric surface computed when sampling points to create the original mesh
Gouraud shading

1. Compute \( L_i = L(b_i, n_i, s_i) \) for each vertex

2. Interpolate the \( L_i \)’s to get the value at \( \vec{p} \)

This step is integrated into the standard triangle-filling algorithm
Gouraud Shading: Computation at Pixels

**Gouraud shading**

1. Compute $L_i = L(\vec{b}_i, \vec{n}_i, \vec{s}_i)$ for each vertex
2. Interpolate the $L_i$‘s to get the value at $\vec{p}$

This step is integrated into the standard triangle-filling algorithm
Gouraud Shading: Comparisons

Gouraud shading
1. Compute $L_i = L\left(\vec{b}_i, \vec{n}_i, \vec{s}_i\right)$ for each vertex
2. Interpolate the $L_i$’s to get the value at $\vec{p}$

Comparison to flat shading

+ No visible seams between mesh triangles
+ Smooth, visually pleasing intensity variation that “mask” coarse geometry
- Specular highlights still a problem for large triangles (why?)
Gouraud shading

1. Compute \( L_i = L(b_i, \vec{n}_i, \vec{s}_i) \) for each vertex
2. Interpolate the \( L_i \)’s to get the value at \( \vec{p} \)
Gouraud shading
1. Compute \( L_i = L(\vec{b}_i, \vec{n}_i, \vec{s}_i) \) for each vertex
2. Interpolate the \( L_i \)'s to get the value at \( \vec{p} \)

\[
L = \beta L_1 + \gamma L_2 + \epsilon L_3
\]

Comparison to flat shading

+ No visible seams between mesh triangles
+ Smooth, visually pleasing intensity variation that “mask” coarse geometry
- Specular highlights still a problem for large triangles (why?)
Gouraud shading
1. Compute $L_i = L(\vec{b}_i, \vec{n}_i, \vec{s}_i)$ for each vertex
2. Interpolate the $L_i$'s to get the value at $\vec{p}$

\[ L = \beta L_1 + \gamma L_2 + \epsilon L_3 \]
Topic 10: Shading

- Introduction to Shading
- Flat Shading
- Interpolative Shading
  - Gouraud shading
  - Phong shading
Phong Shading: Main Idea

Phong shading:
1. Interpolate to get at
   \[ \bar{b}_i, \bar{n}_i, \bar{s}_i \]
   \[ \bar{b}, \bar{n}, \bar{s} \]
   \[ L(\bar{b}, \bar{n}, \bar{s}) \]

Comparison to Gouraud shading

+ Smooth intensity variations as in Gouraud shading

+ Handles specular highlights correctly even for large triangles (Why?)

\[ L(\bar{b}, \bar{n}, \bar{s}) = r_a I_a + r_d I_d \max(0, \bar{n} \cdot \bar{s}) + r_s I_s \max(0, \bar{n} \cdot \bar{b})^\alpha \]

- intensity at projection of point \( \bar{p} \)
- ambient
- diffuse
- specular
Phong Shading: Comparisons

Phong shading:
1. Interpolate to get \( \bar{b}, \bar{n}, \bar{s} \) at \( \bar{p} \)
2. Compute
   \[
   L(\bar{b}, \bar{n}, \bar{s})
   \]

Comparison to Gouraud shading

+ Smooth intensity variations as in Gouraud shading

+ Handles specular highlights correctly even for large triangles (Why?)

it is possible to have a significant specular component at \( \bar{p} \) even when all vertices have a negligible specular component
Phong shading:
1. Interpolate to get \( \vec{b}, \vec{n}, \vec{s} \) at \( \vec{b}_i, \vec{n}_i, \vec{s}_i \)
2. Compute \( L(\vec{b}, \vec{n}, \vec{s}) \)
Phong Shading: Comparisons

Phong shading:
1. Interpolate to get \( \vec{b}_i, \vec{n}_i, \vec{s}_i \) at \( \vec{b}, \vec{n}, \vec{s} \)
2. Compute

\[
L(\vec{b}, \vec{n}, \vec{s})
\]

Comparison to Gouraud shading

+ Smooth intensity variations as in Gouraud shading

+ Handles specular highlights correctly even for large triangles (Why?)

- Computationally less efficient (but okay in today's hardware!) (Must interpolate 3 vectors & evaluate Phong reflection model at each triangle pixel)