Shading

Computer Graphics
Instructor: Sungkil Lee

Today

- Overview
- Light source and materials
- Phong illumination model
- Blinn-Phong illumination model
- Polygonal shading

Shading

- **Shading:**
  - is a process of altering the color of an object/surface/polygon in a 3D scene, based on how lights and materials interact.

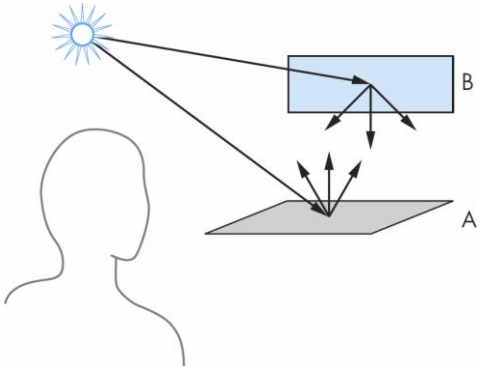
Color or albedo

vs.

Shading

Light-Material Interaction

- **Light-material interactions**
  - cause each point to have a different color or shade.
  - Light reaching surfaces are partially absorbed, and some reflected.



- **The color (or brightness) of an object**
  - The amounts of reflected lights
  - For example, a surface appears red under white light, because the red component of the light is reflected and the rest absorbed.

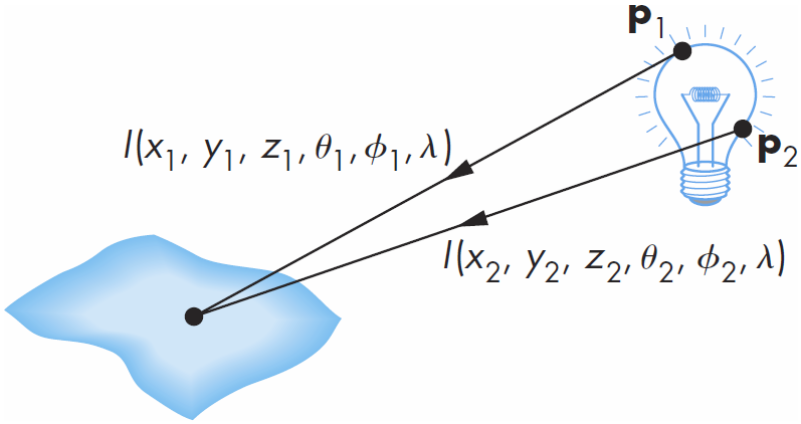
Light Sources and Materials

Elements of Light-Material Interaction

- **Four elements on which light reflection depends:**
  - Light sources
  - Surface materials (e.g., smoothness) of the object surfaces
    - Material indicates actually an internal atomic/molecular structure.
  - Surface orientation: indicated by normal vectors
  - Location of viewer: eye/camera position in our viewing

Abstract Modeling

- **Abstract modeling**
  - Light sources and surface materials are modeled as abstract forms.
  - Location of viewer and surface orientation are used as they are.
- **Example: general lights**
  - are hard to work with, because we must integrate light coming from all points on the source.
  - Instead, we consider simpler models, which effectively abstracts general light sources.

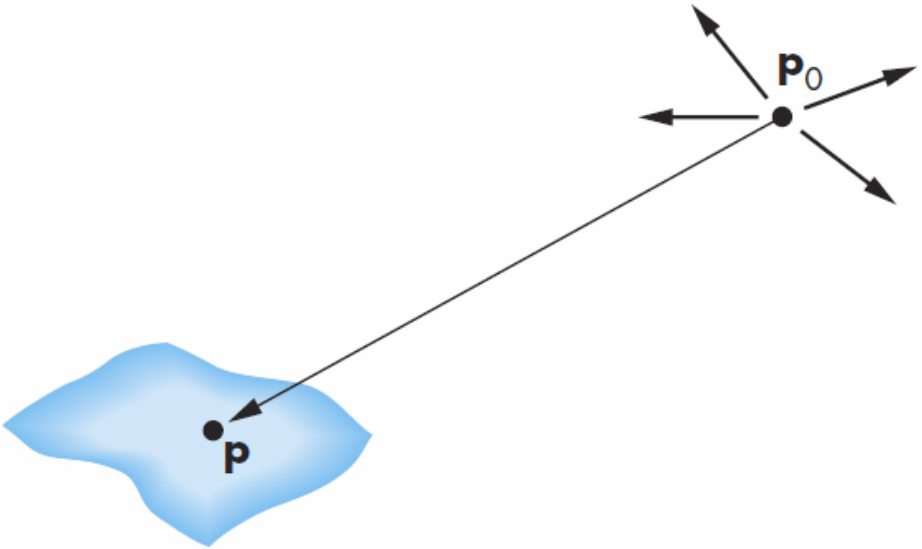


Light Source Models

- **Three simplified light source models**
  - Point light source
  - Distant (or directional) light source
  - Ambient light

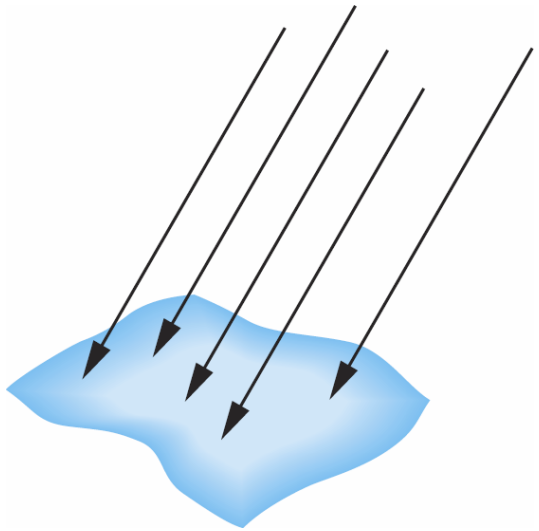
Light Source Models

- **Point light source**
  - Model with position and color.
  - The light source position is a point.
  - We may apply distance attenuation, inversely scaling with the square of distances from a light source to surface points.



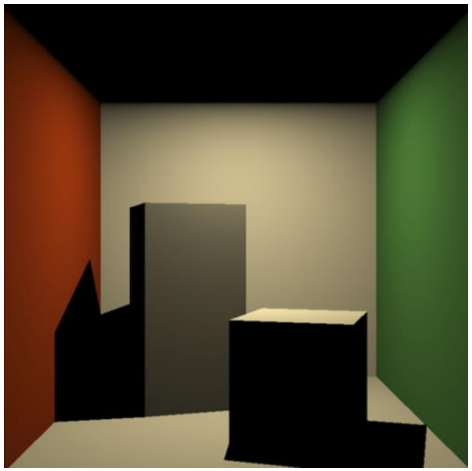
Light Source Models

- **Distant (directional) light source**
  - Infinite distance away (e.g., parallel sunlight)
  - The light source has only a direction vector.
    - In case of HC (i.e., vec4), the last element w is 0.



Light Source Models

- **Ambient light**
  - Same amount of light everywhere in a scene, like a shading offset
  - Can model contribution of many sources and indirect reflections.



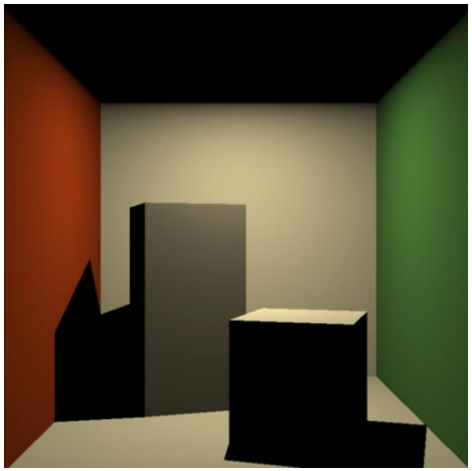
Direct

Indirect

Direct + Indirect

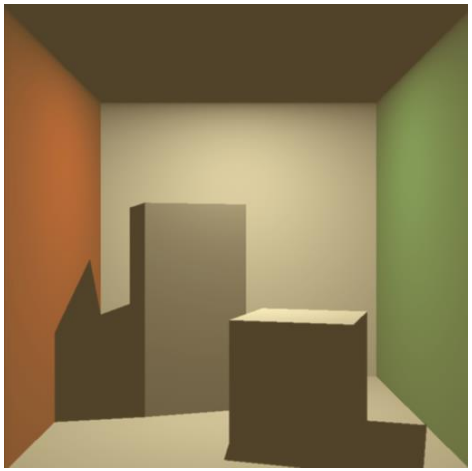
Light Source Models

- **Ambient light**
  - Same amount of light everywhere in a scene, like a shading offset
  - Can model contribution of many sources and indirect reflections.
  - This, however, is significant approximation, and limited in practice.



Direct

Ambient



Direct + Ambient

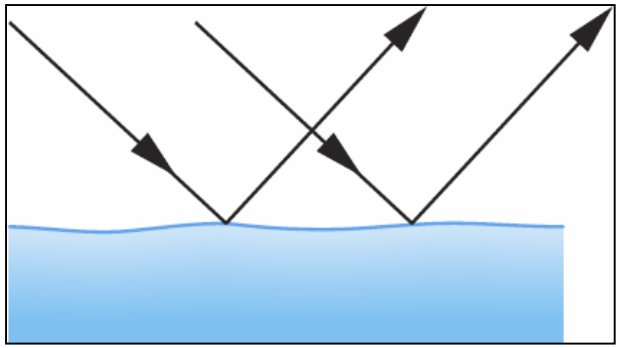
Surface Materials

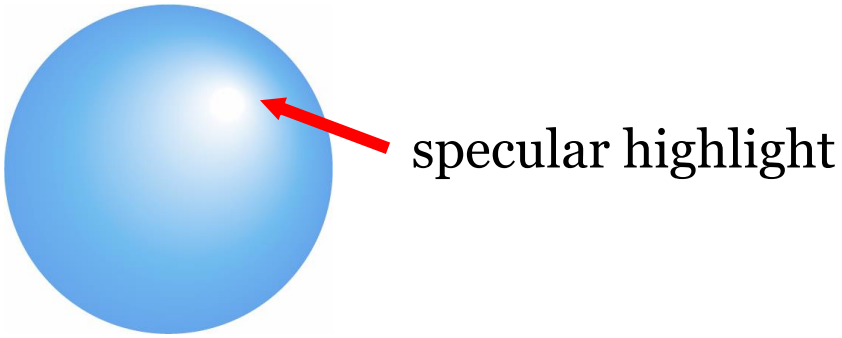
- **Microfacet model**
  - Surface are composed of many very tiny facets, each of which is a perfect specular reflector (i.e., mirror).
  - The distribution of the microfacets or their normals determine the characteristics of surface reflection.

Smooth and rough microfacets

Types of Surface Materials

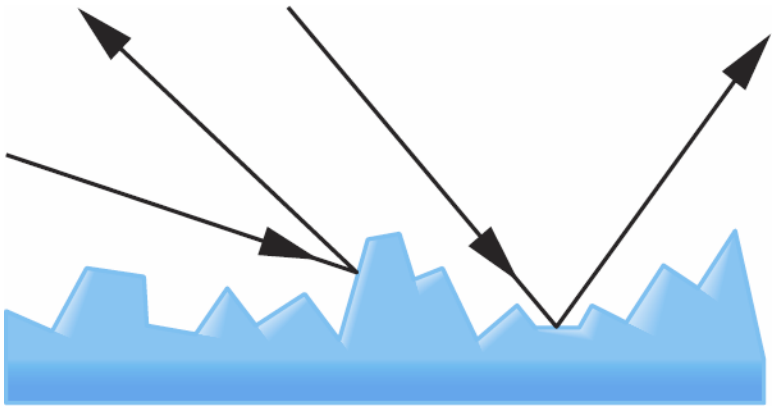
- **Specular/glossy surfaces**
  - Smooth surfaces show specular highlights.
  - The smoother a surface is, the more reflected light is concentrated into the direction of a perfect mirror reflection.





Types of Surface Materials

- **Diffuse reflector**
  - A very rough surface scatters light equally in all directions.
  - Amount of light reflected is proportional to the vertical component of incoming light.
  - Ideal (or perfect) diffuse surfaces are called ***"Lambertian"*** surfaces.



Surface Materials

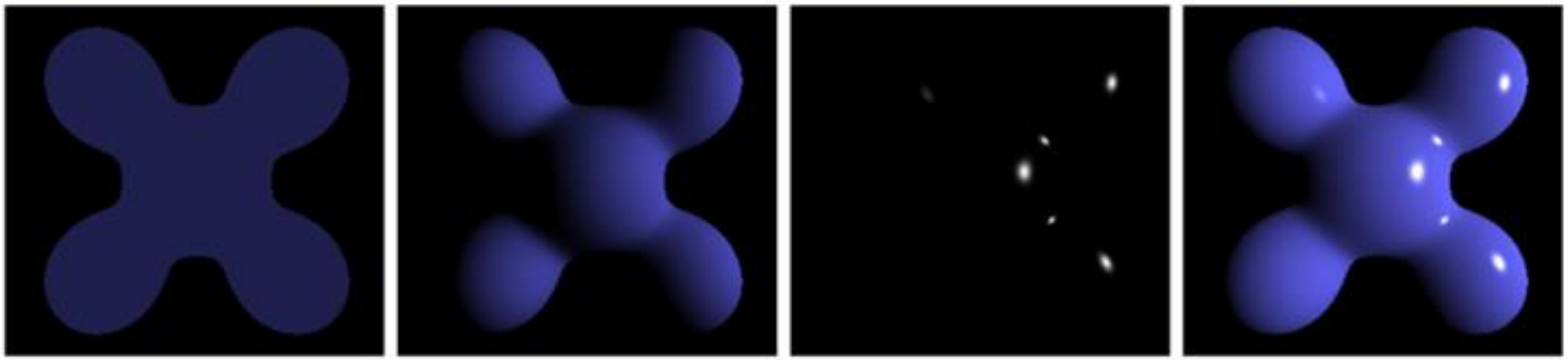
- **The types of surface materials**
  - Most surfaces are neither ideally rough or perfectly smooth.
  - Hence, we decompose light source and surface materials, and combine them for the final shading.
  - Phong illumination model implements this idea.

Phong Illumination Model

Phong Illumination Model

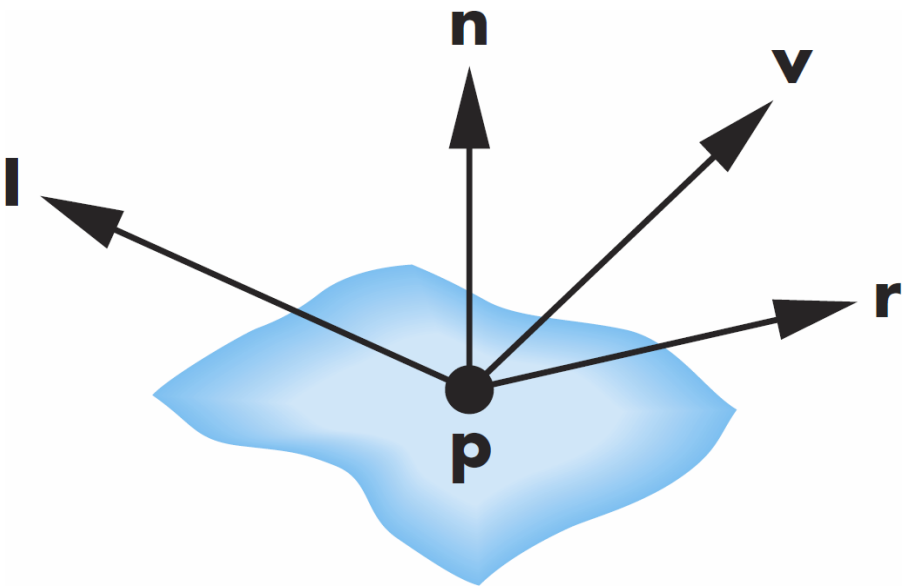
- **A simple empirical model that**
  - can be computed rapidly in the pipeline approach.
  - the form does not have perfect physical justification.

- **Three illumination components:**
  - Ambient, diffuse, and specular



Phong Illumination Model

- **Uses four vectors**
  - To light source: **l** (a vector ***not from light source but to light source***)
  - To viewer: **v**
  - Normal vector: **n** (will be discussed later in more details)
  - Ideal perfect reflector: **r**

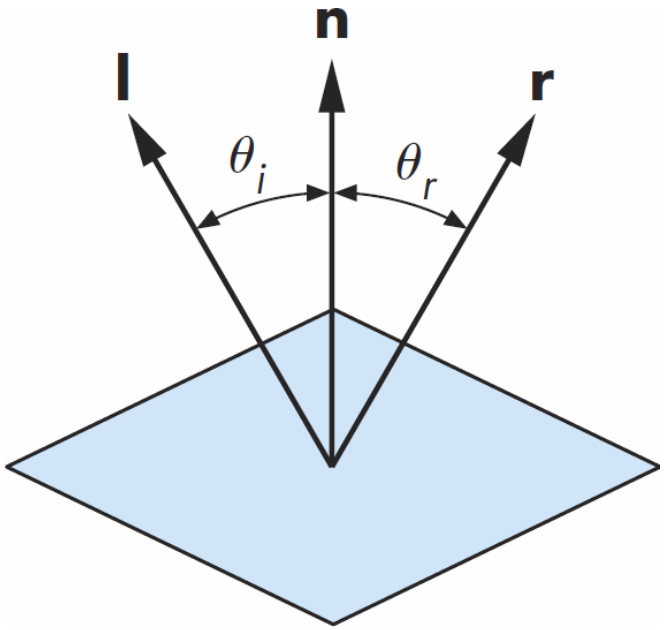


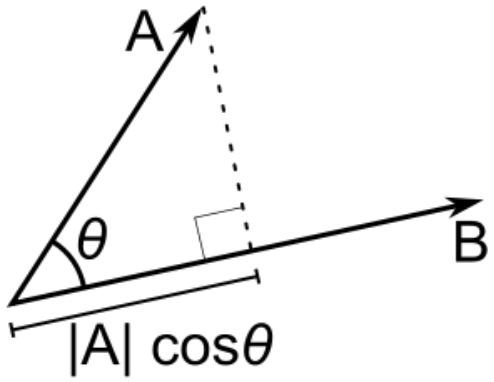
Perfect Reflector $\mathbf{r}$

- **While $\mathbf{l}$ and $\mathbf{n}$ are given, $\mathbf{r}$ needs to be computed**
  - Angle of incidence (θ_i) = angle of reflection (θ_r)
  - The three vectors must be coplanar.
  - The reflector is given as:

$$\mathbf{r} = 2(\mathbf{l} \cdot \mathbf{n})\mathbf{n} - \mathbf{l}$$

  - Note: projection of l onto n is $(\mathbf{l} \cdot \mathbf{n})\mathbf{n}$



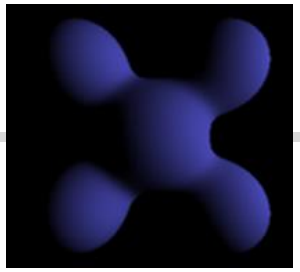


Ambient Reflections

- **Ambient reflections**
  - Amounts of reflected lights depend on both the color of the lights and the material properties (i.e., reflectance; $\mathbf{k}_a$) of the object.
  - This is a coarse approximation to the secondary (indirect) reflections among the surfaces.

$$\mathbf{I}_{ra} = \mathbf{k}_a \mathbf{I}_a$$

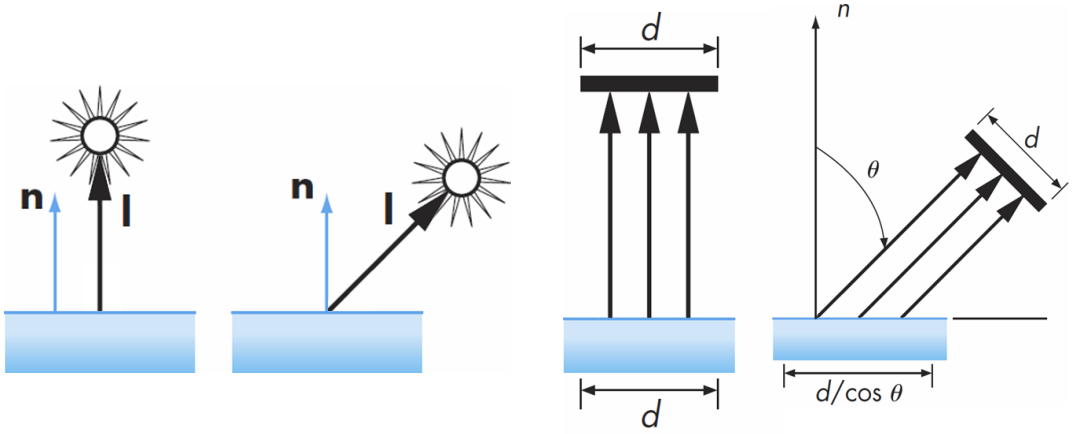
Diffuse Reflections



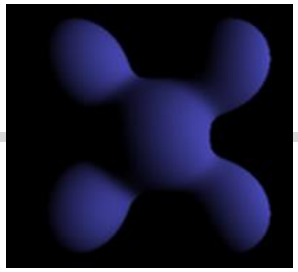
- **Diffuse reflections**
  - Similar to ambient reflection, but the angular attenuation is included here:

$$\mathbf{I}_{rd} = \max((\mathbf{l} \cdot \mathbf{n})\mathbf{k}_d \mathbf{I}_d, 0)$$

- **Angular attenuation $(\mathbf{l} \cdot \mathbf{n})$:**
  - amount of incoming light scales with the angle of incoming light.
  - reflected light $\approx \cos\theta_i = \mathbf{l} \cdot \mathbf{n}$



Diffuse Reflections



- **We can further incorporate distance attenuation,**

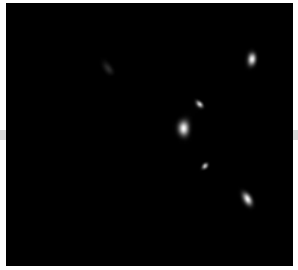
$$\mathbf{I}_{rd} = \text{att}(d) \cdot \max(\mathbf{k}_d(\mathbf{l} \cdot \mathbf{n})\mathbf{I}_d, 0)$$

- **Distant attenuation**
  - The light from a point source that reaches a surface is inversely proportional to the square of the distance (d)

$$\text{att}(d) = \frac{1}{a + bd + cd^2},$$

  - $\mathrm{a}, \mathrm{b}, \mathrm{c}$ are user-defined coefficients.
  - The constant and linear terms (a and b) soften the effect of the point source

Specular Reflections

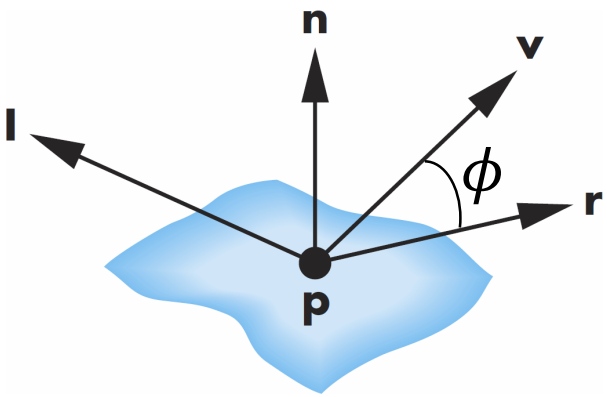


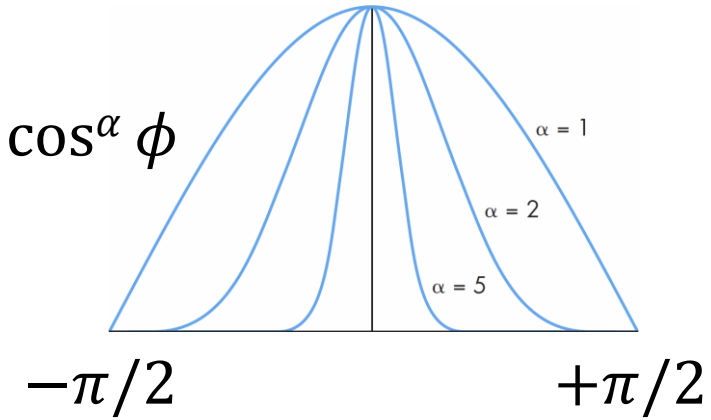
- **Specular reflections (with distance attenuation):**

$$\mathbf{I}_{rs} = \text{att}(\text{d}) \cdot \max(\mathbf{k}_s \mathbf{I}_s (\mathbf{r} \cdot \mathbf{v})^{\alpha}, 0)$$

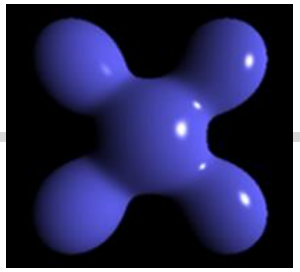
  - The ideal reflector $\mathbf{r}$ closer to the viewer direction $\mathbf{v}$ shines more.
  - Hence, the angle between $\mathbf{v}$ the $\mathbf{r}$ can be a drop-off term.

$$\cos^{\alpha}\phi = (\mathbf{r} \cdot \mathbf{v})^{\alpha}$$





Putting them altogether



- **For each light source and each color component,**
  - the Phong model can be written (without distance terms) as:

$$\mathbf{I}_r = \mathbf{I}_{ra} + \mathbf{I}_{rd} + \mathbf{I}_{rs}$$
$$= \mathbf{k}_a \mathbf{I}_a + \max(\mathbf{k}_d(\mathbf{l} \cdot \mathbf{n})\mathbf{I}_d, 0) + \max(\mathbf{k}_s(\mathbf{r} \cdot \mathbf{v})^{\alpha}\mathbf{I}_s, 0)$$

  - You can also include distance attenuation term to the model, if necessary.

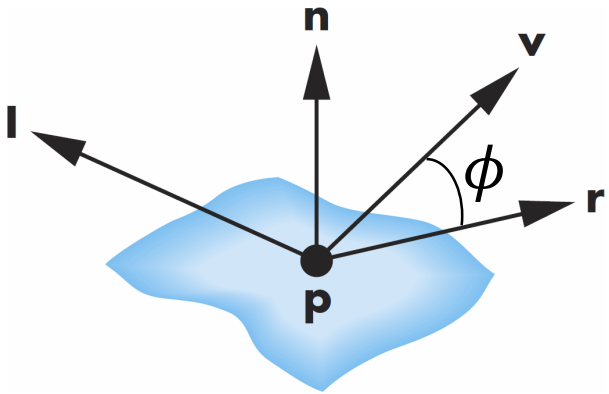
Blinn-Phong Model:
Efficient approximation to Phong Model

Modified Phong Model

- **The specular term in Phong model**
  - calculates $\mathbf{r}$ and $\mathbf{v}$ for each vertex

$$\mathbf{k}_s(\mathbf{r} \cdot \mathbf{v})^{\alpha}\mathbf{I}_s$$

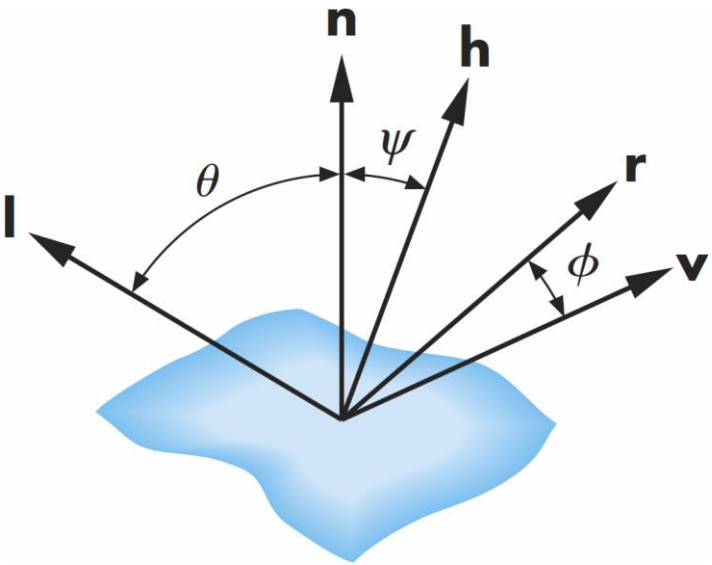
  - recall that $\mathbf{r} = 2(\mathbf{l} \cdot \mathbf{n})\mathbf{n} - \mathbf{l}$
  - Blinn suggested an efficient approximation using the halfway vector.



The Halfway Vector

- **The halfway vector $\mathbf{h}$:**
  - A normalized vector halfway between $\mathbf{l}$ and $\mathbf{v}$.
  - When $\mathbf{v}$ is coplanar with $\mathbf{l}$, $\mathbf{n}$, and, $\mathbf{r}$, we can show that $2\psi = \phi$.

$$\mathbf{h} = \frac{\mathbf{l}+\mathbf{v}}{|\mathbf{l}+\mathbf{v}|}$$



Blinn-Phong Model

- **The Blinn-Phong model can be written:**

$$\mathbf{I}'_r = \mathbf{I}_{ra} + \mathbf{I}_{rd} + \mathbf{I}'_{rs}$$
$$= \mathbf{k}_a\mathbf{I}_a + \max(\mathbf{k}_d(\mathbf{l}\cdot\mathbf{n})\mathbf{I}_d, 0) + \max(\mathbf{k}_s(\mathbf{n}\cdot\mathbf{h})^{\beta}\mathbf{I}_s, 0)$$

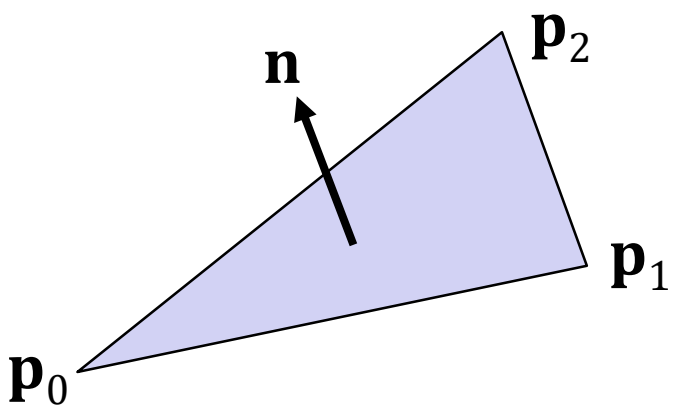
  - Replace $(\boldsymbol{v}\cdot\boldsymbol{r})^{\alpha}$ with $(\boldsymbol{n}\cdot\boldsymbol{h})^{\beta}$
  - β is chosen to match shininess α.
  - This model has been a standard in the old-style OpenGL.

Polygonal Shading: Effects of Normal Vectors

Per-Face Normals

- **Normal vectors usually indicate face (or per-face) normals.**
  - Given 3 vertices, $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$, the *surface normal* is given as:

$$\mathbf{n} = (\mathbf{p}_1 - \mathbf{p}_0) \times (\mathbf{p}_2 - \mathbf{p}_0)$$



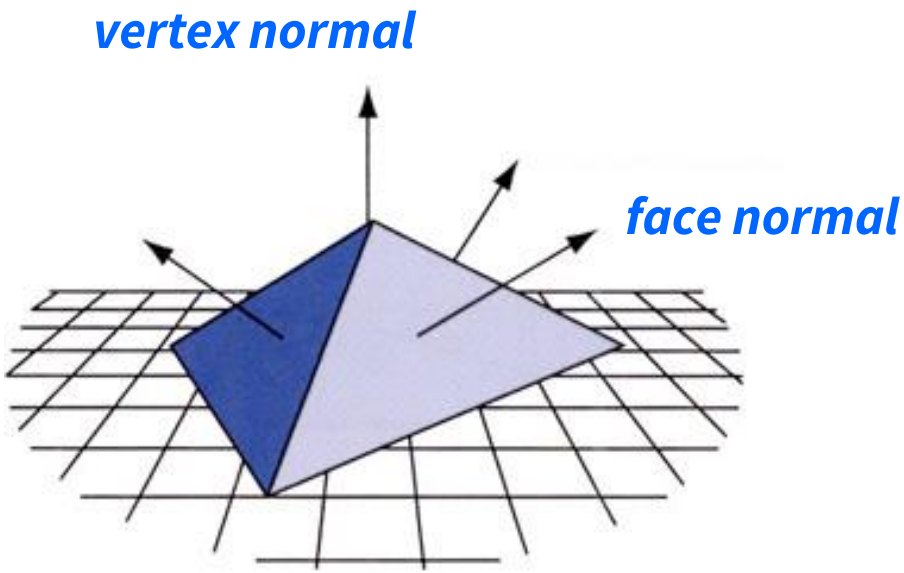
Discontinuity with Face Normals

- **However, polygonal meshes is composed of discrete polygons (or triangles).**
  - The discontinuity of per-face normals causes the discontinuity of shadings around the face boundaries. For example,

- **So, it is better to use different normals:**
  - **which can be interpolated**
  - either of per-vertex normals or per-fragment normals

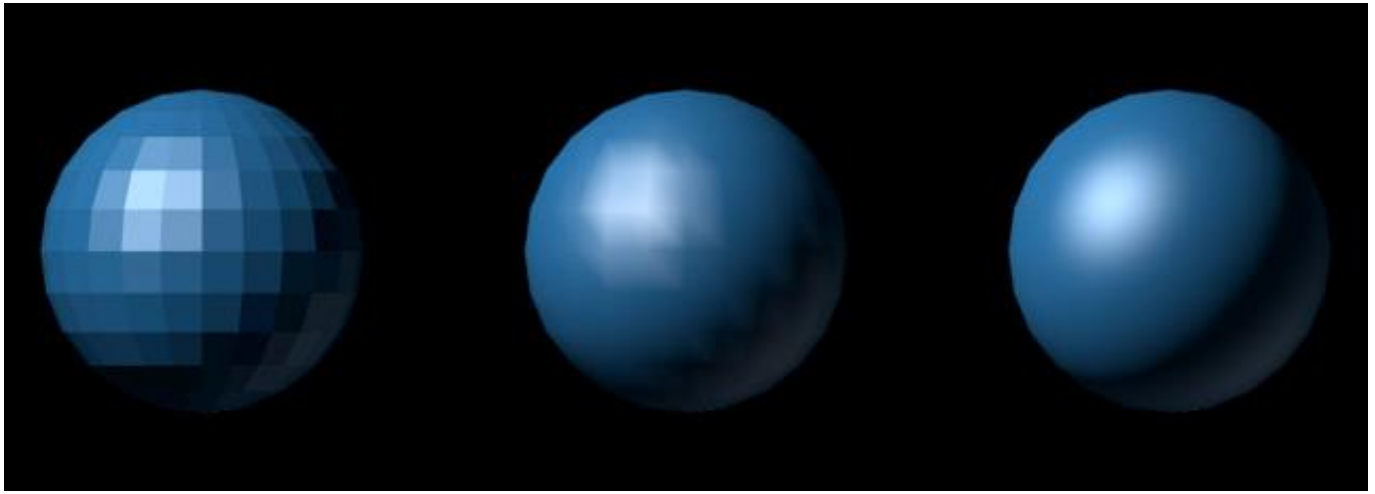
(Per-) Vertex Normals

- **A *vertex normal* is the average of adjacent face normals.**
  - Physically, this does not make sense, but provides smoother shading by **interpolating** the normals of adjacent vertices.
  - Hence, this is the default choice for normals in CG.
    - Saying just "normals" means vertex normals in practice in CG.



Three Polygonal Shading Methods

- **Shading implementation relies on:**
  - normals: either per-face or per-vertex normals
  - interpolation: either colors or normals
- **There are three techniques**
  - Flat shading
  - Gouraud shading
  - Phong shading



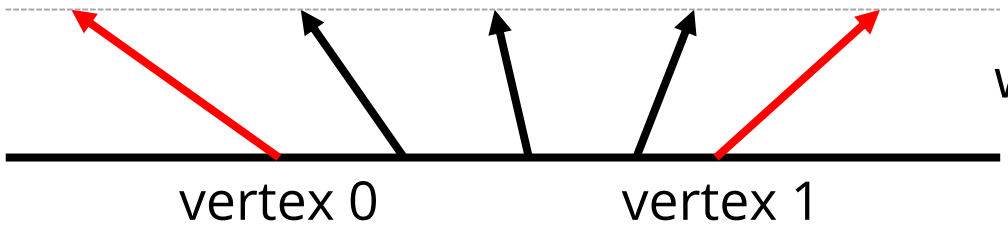
Three Polygonal Shading Methods

- **Where shading is applied for each method with normals:**
  - Only Phong shading computes in FS, while the others in VS.
  - Now, we see why we call vertex "shader."
    - Gouraud shading is computed in vertex shader.
    - It was a standard for fixed OpenGL pipeline (in old-style OpenGL).

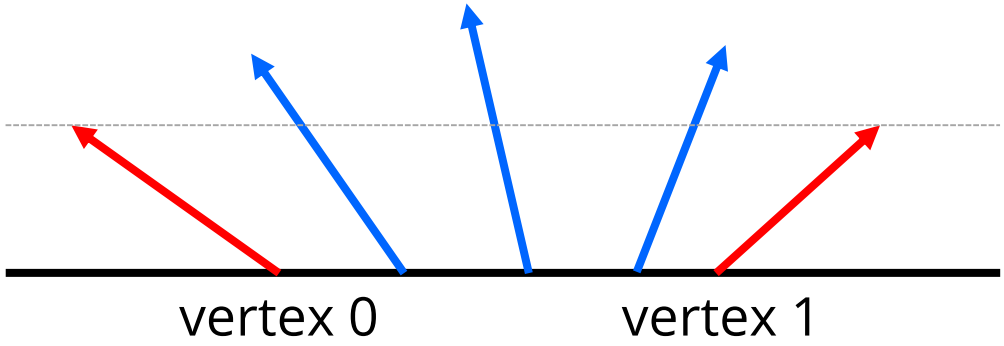
Method	Input normals	Vertex shader	Interpolation in Rasterizer		Fragment shader
			from	to	
Flat	face normals	**shading**			
Gouraud	vertex normals	**shading**	vertex colors	fragment colors	
Phong	vertex normals		vertex normals	fragment normals	**- renormalize** fragment normals **- shading** with renormalized normals

Normals in Phong Shading

- **Phong shading requires to re-normalize per-fragment normals.**
  - Per-fragment normals are linearly interpolated by rasterizer, instead of spherical interpolation.
  - So, we need to make them have unit lengths.



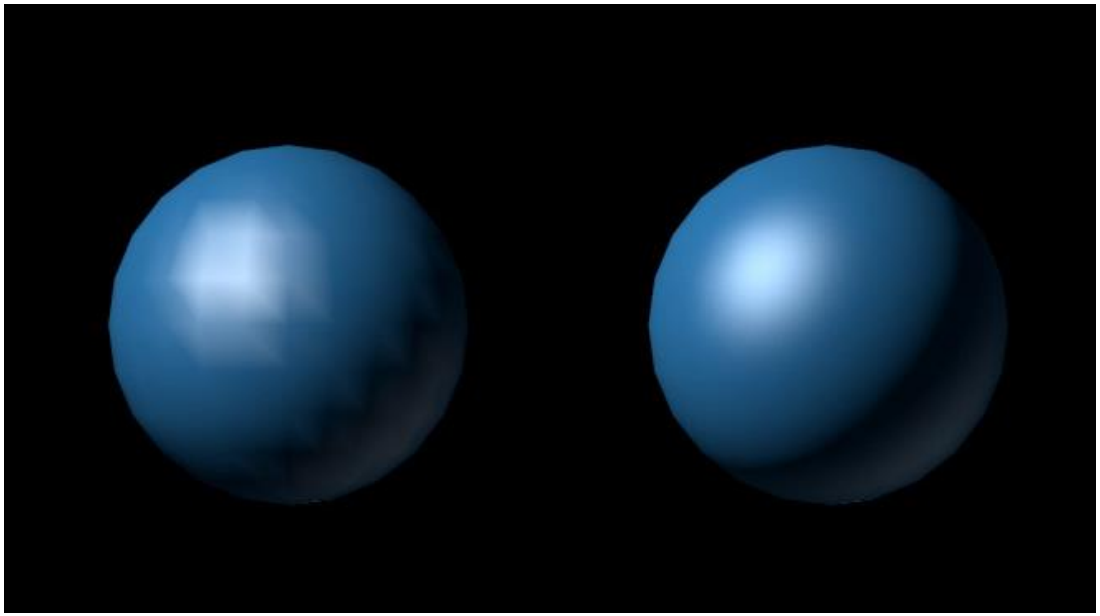
without re-normalization



with re-normalization

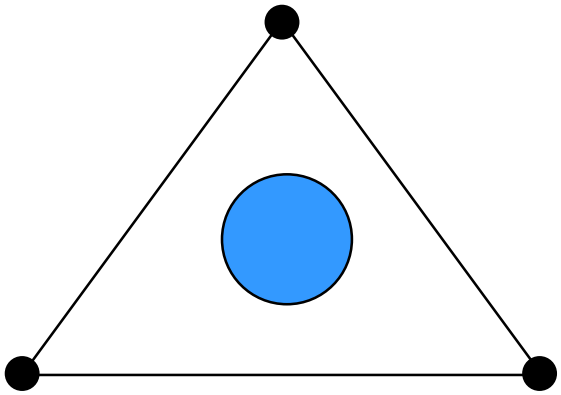
Gouraud vs. Phong Shading

- **Phong shading better preserves details, and much smoother than Gouraud shading.**
  - This difference becomes larger with sharper specular reflections.



Gouraud vs. Phong Shading

- **Phong shading better preserves details, and much smoother than Gouraud shading.**
  - If highlights reside inside triangles (see the figure below), that would be missing in Gouraud shading.
  - Example: Blue highlight will be missing in shading of vertices.
  - On the other hand, Phong shading interpolates normals, and the per-fragment highlights are captured well.



Phong Shading vs. Phong Illumination Model

- **Phong shading and Phong illumination model are different.**
  - Phong shading: shading using per-fragment normals
  - Phong illumination model: shading model with ambient, diffuse, specular components.