Topic 8:

Lighting & Reflection models

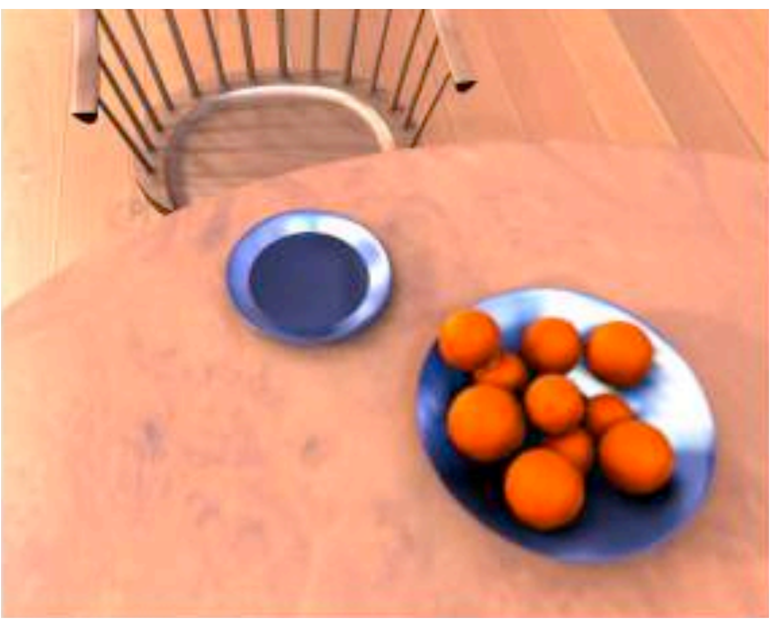
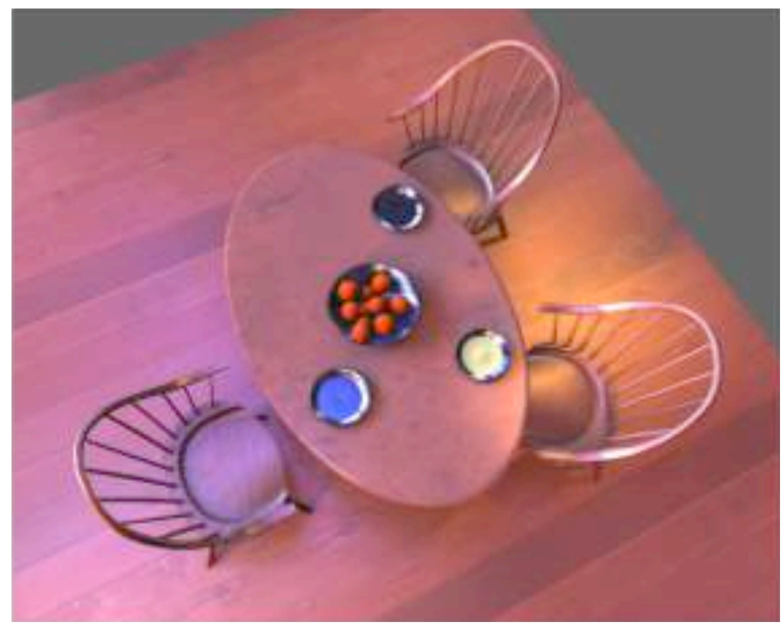
- Lighting & reflection
- The Phong reflection model
    - diffuse component
    - ambient component
    - specular component

Showtime

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

Terminology

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

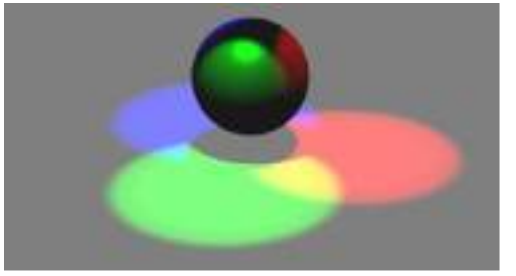
Illumination Models

- Simple approximations of light transport
- Physical models of light transport

Two Components of Illumination

Light Sources

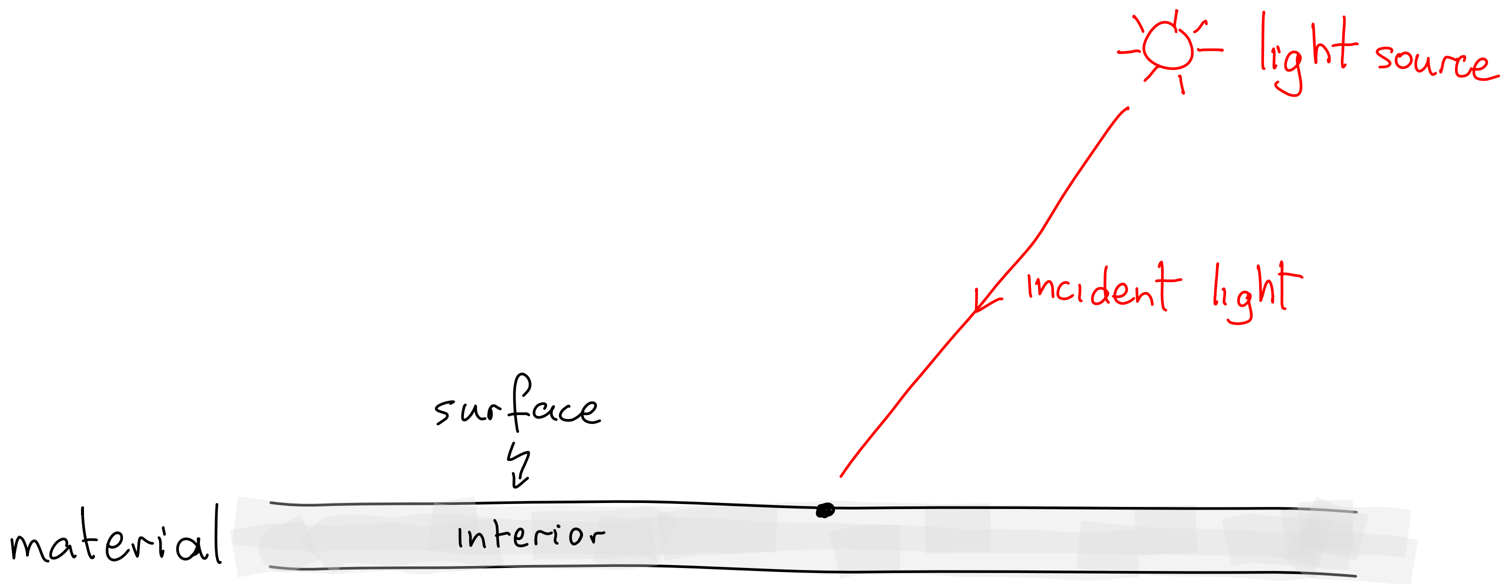
- Emission Spectrum (color)
- Geometry (position and direction)
- Directional Attenuation



Surface Properties (Reflectors)

- Reflectance Spectrum (color)
- Geometry (position, orientation, and micro-structure)
- Absorption
- Transmission

Light Sources



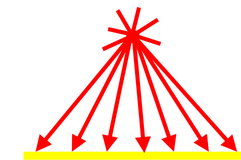
Main sources of light:

- Point source
- Directional Light
- Spotlight

Light Source Types

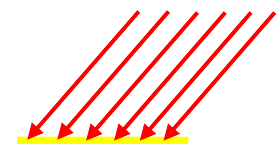
Point Light

- light originates at a point



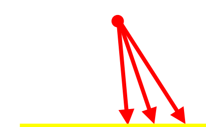
Directional Light (point light at infinity)

- light rays are parallel
- Rays hit a planar surface at identical angles



Spot Light

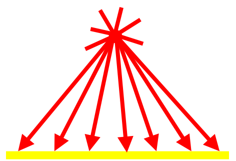
- point light with limited angles



Bessmeltsev et al.

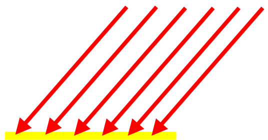
Light Source Types

Point Light



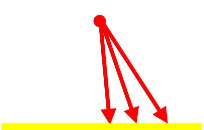
- light originates at a point
- defined by **location** only

Directional Light (point light at infinity)



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

Spot Light

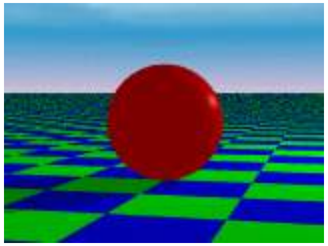


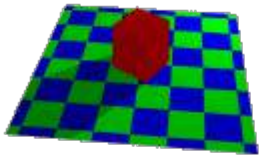
- point light with limited angles
- defined by **location, direction, and angle range**

Bessmeltsev et al.

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.

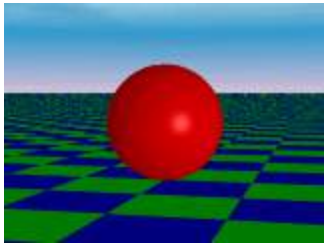




The direction of the light to each point on a surface changes when a point light source is used. Thus, a normalized vector to the light emitter must be computed for each point that is illuminated.

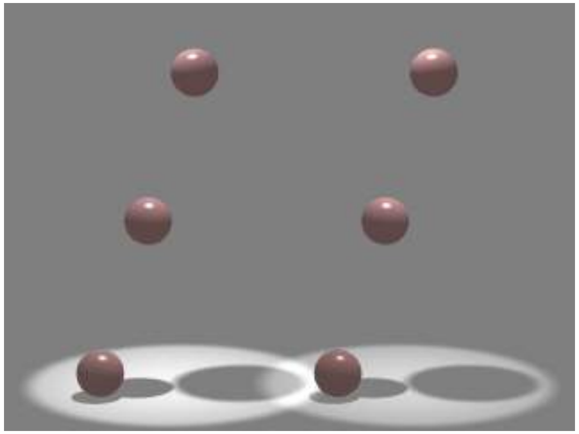
Directional Light Sources

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

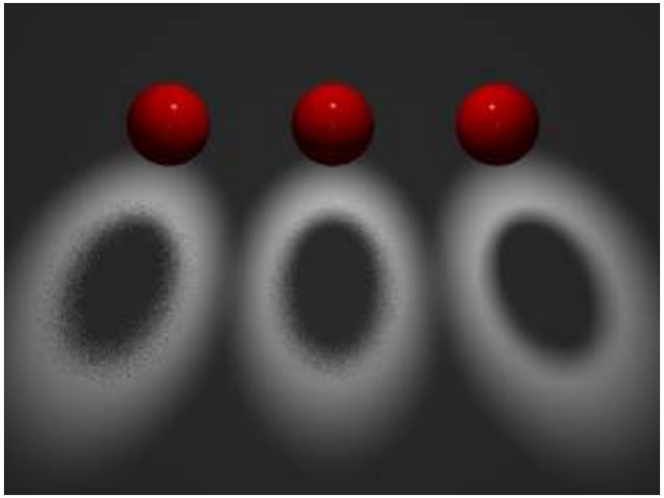


Spotlights

- 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

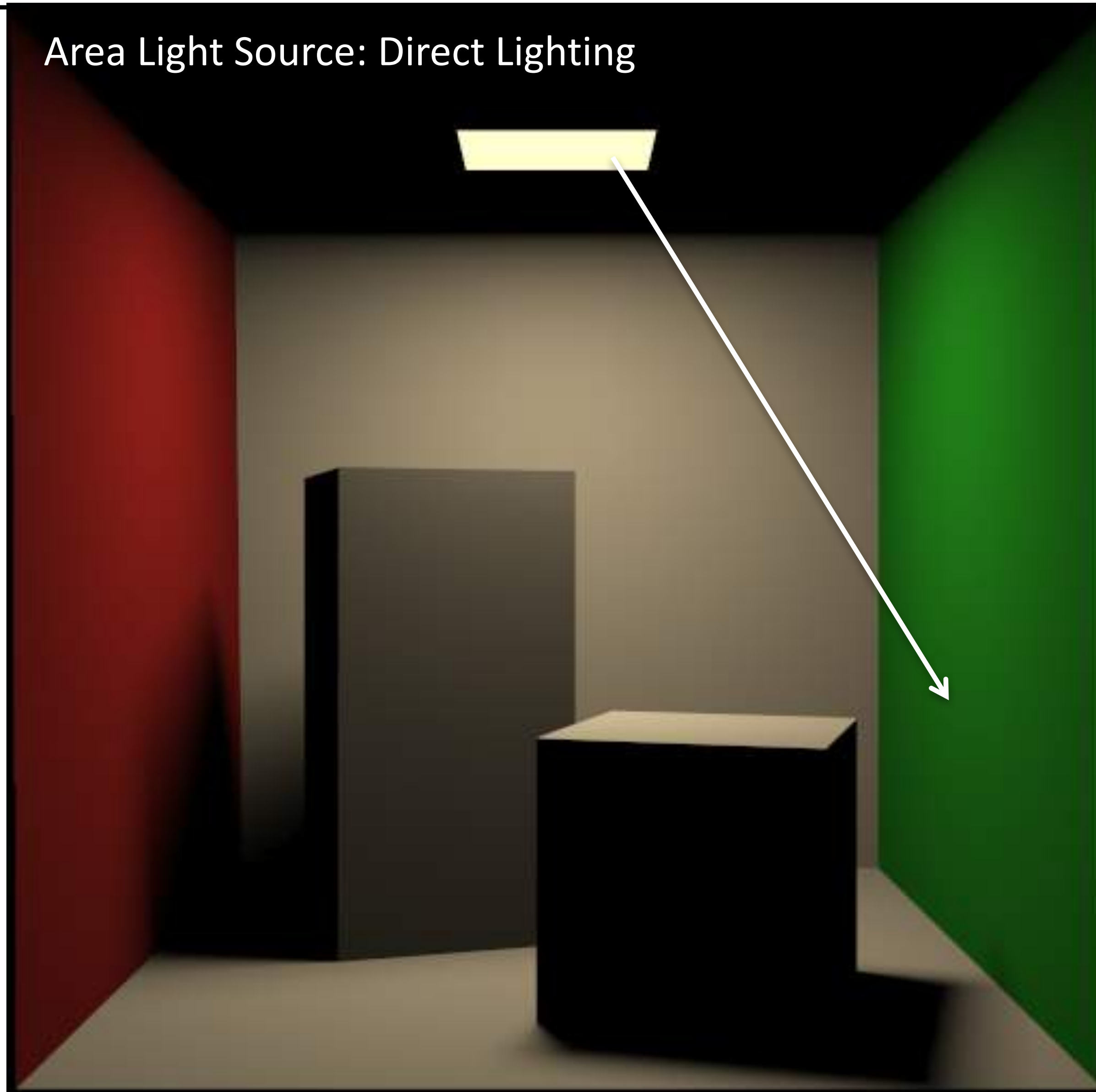
Area Light Sources

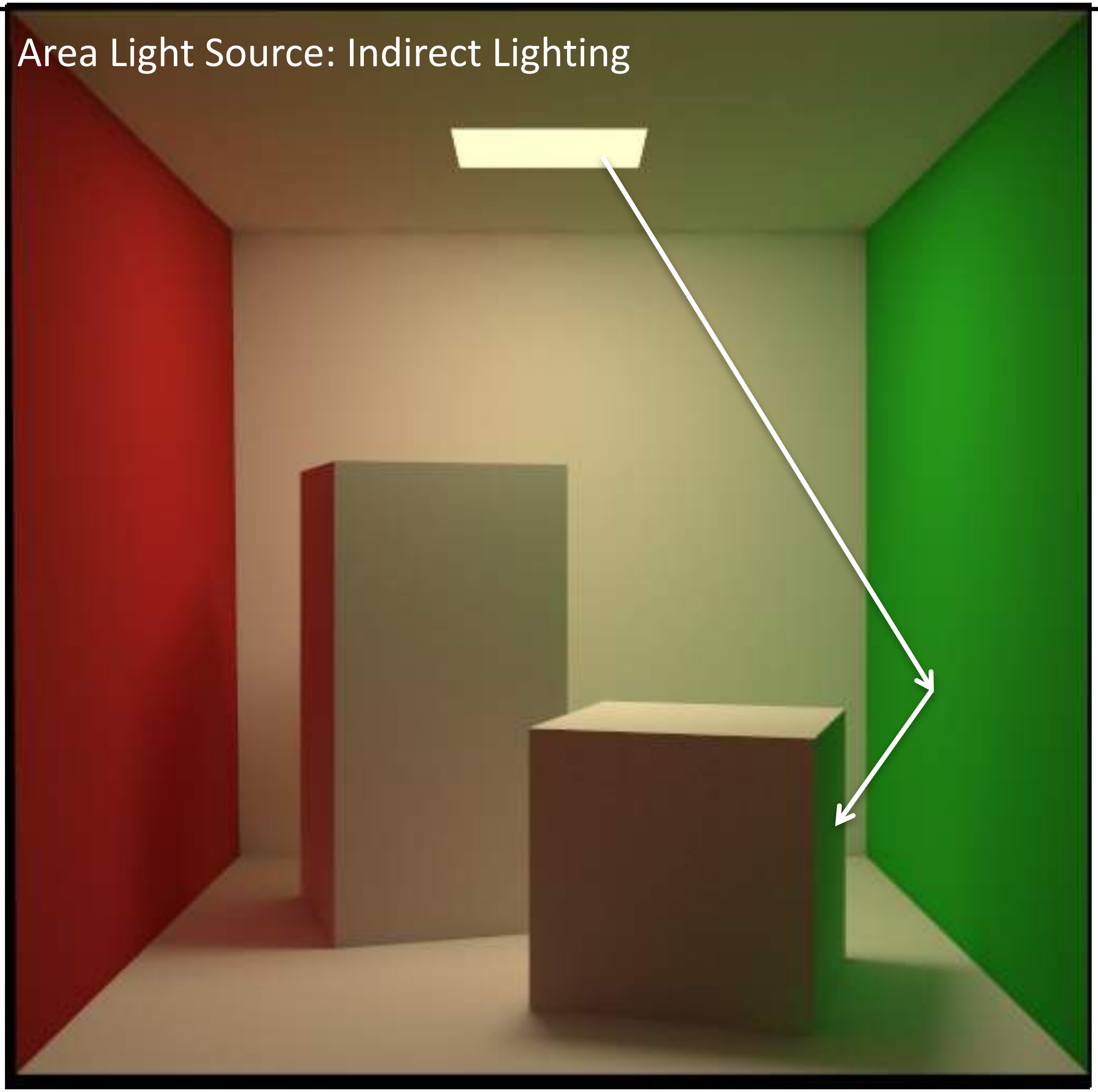
- Light source occupies a 2-D area (usually a polygon or disk)
- Generates *soft* shadows



Extended Light Sources

- Spherical Light Source
- Generates *soft* shadows

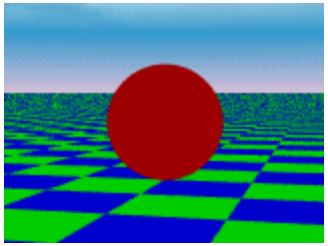

Area Light Source: Direct Lighting


Area Light Source: Indirect Lighting

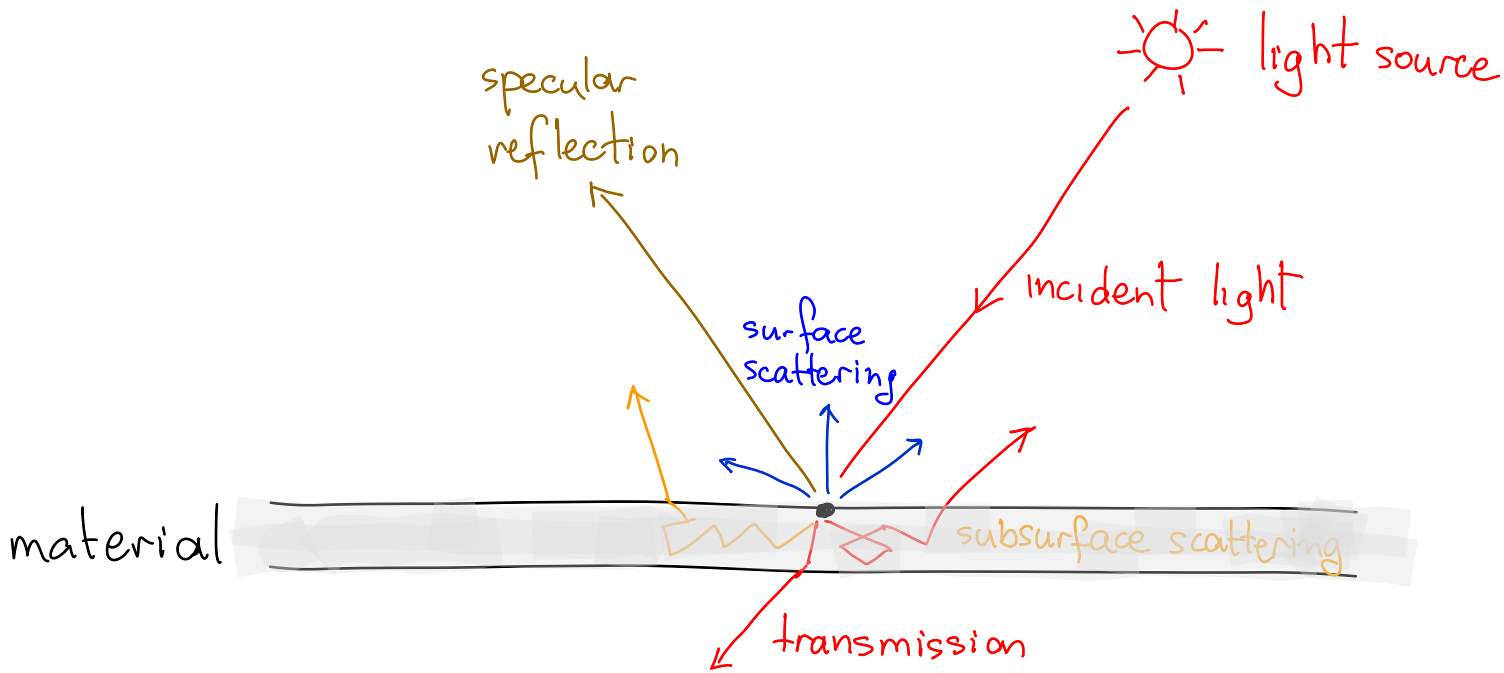
Ambient Light Source

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”

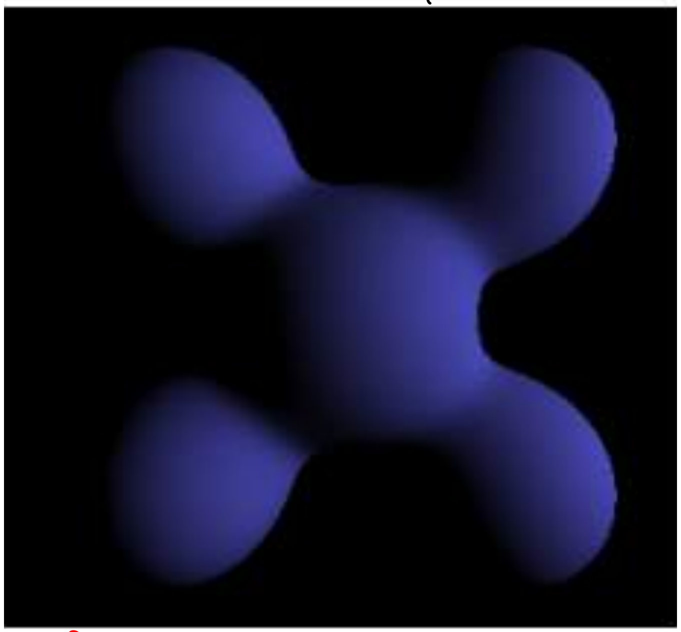


Two Types of SurfaceReflection

1. Diffuse Reflection
2. Specular Reflection

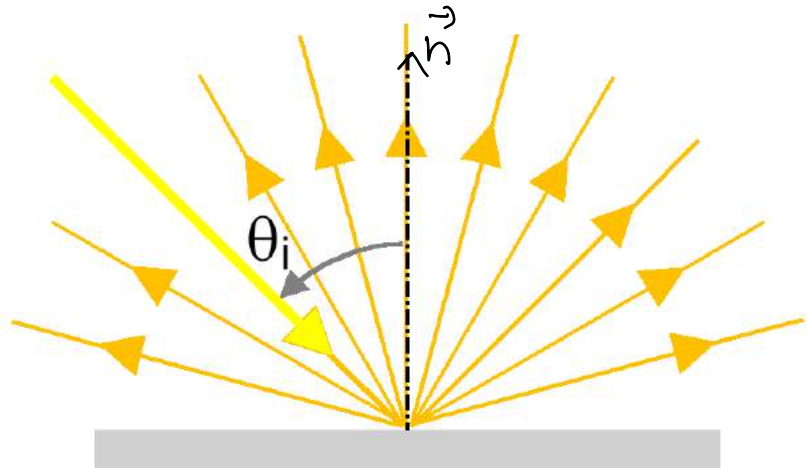
Modeling Reflection: Diffuse Reflection

Brad Smith, Wikipedia



Diffusely-shaded object

θ_i = angle of incidence

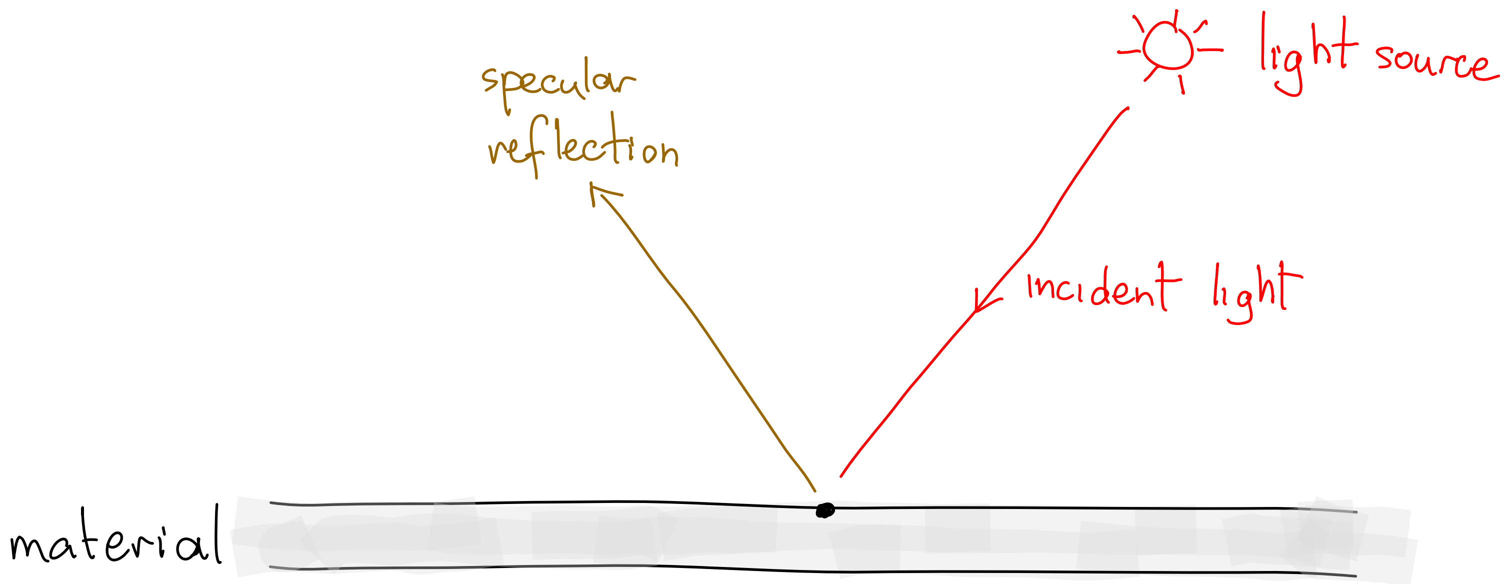


Panjasan, Wikipedia

Diffuse reflection:

- Represents "matte" component of reflected light
- Usually cause by "rough" surfaces (clay, eggshell, etc)

Modeling Reflection: Specular Reflection



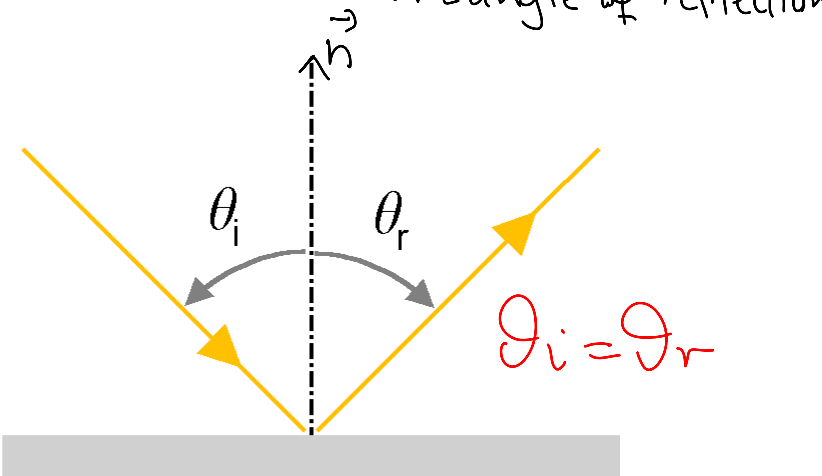
Specular reflection:

- Represents shiny component of reflected light
- Caused by mirror like reflection off of smooth or polished surfaces (plastics, polished metals, etc)

Modeling Reflection: Specular Reflection

Romeiro et al, ECCV '08

mirror-like sphere



Panjasan, Wikipedia

Specular reflection:

- Represents shiny component of reflected light
- Caused by mirror like reflection off of smooth or polished surfaces (plastics, polished metals, etc)

Modeling Reflection: Specular Reflection

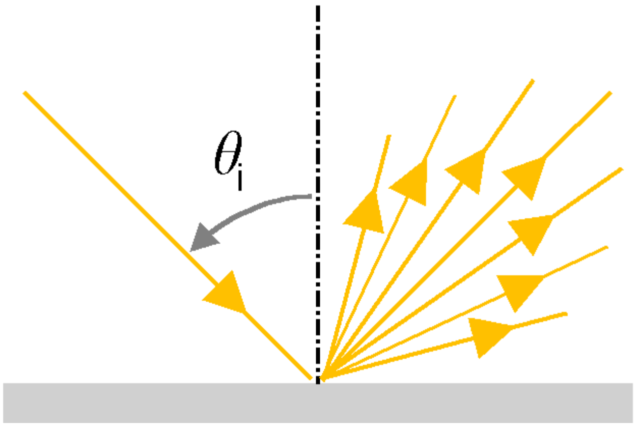
Romeiro et al, ECCV'08

Specular reflection:

- Represents shiny component of reflected light
- Caused by mirror like reflection off of smooth or polished surfaces (plastics, polished metals, etc)

Modeling Reflection: Specular Reflection

Romeiro et al, ECCV '08



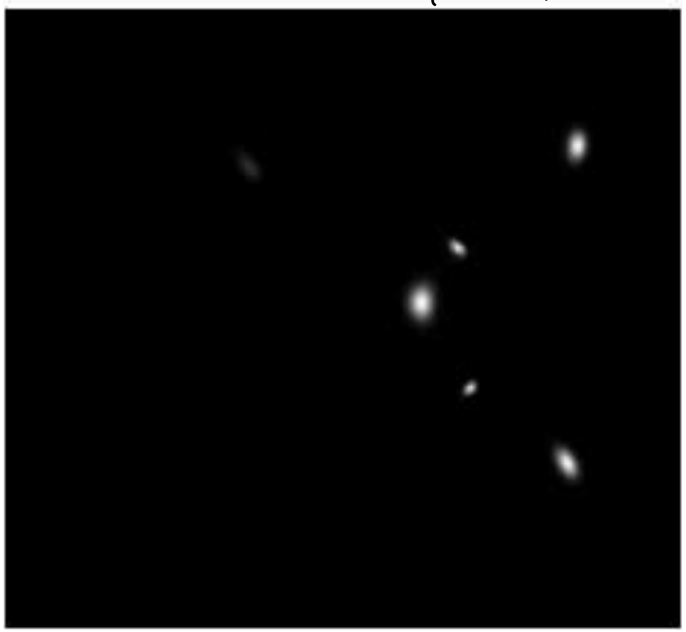
Panjasan, Wikipedia

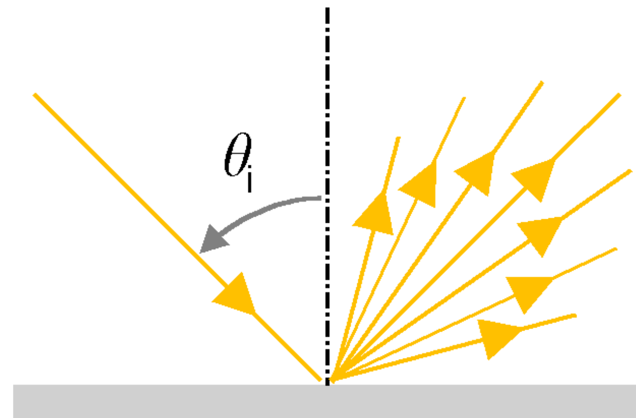
Specular reflection:

- Represents shiny component of reflected light
- Caused by mirror like reflection off of smooth or polished surfaces (plastics, polished metals, etc)

Modeling Reflection: Specular Reflection

Brad Smith, Wikipedia



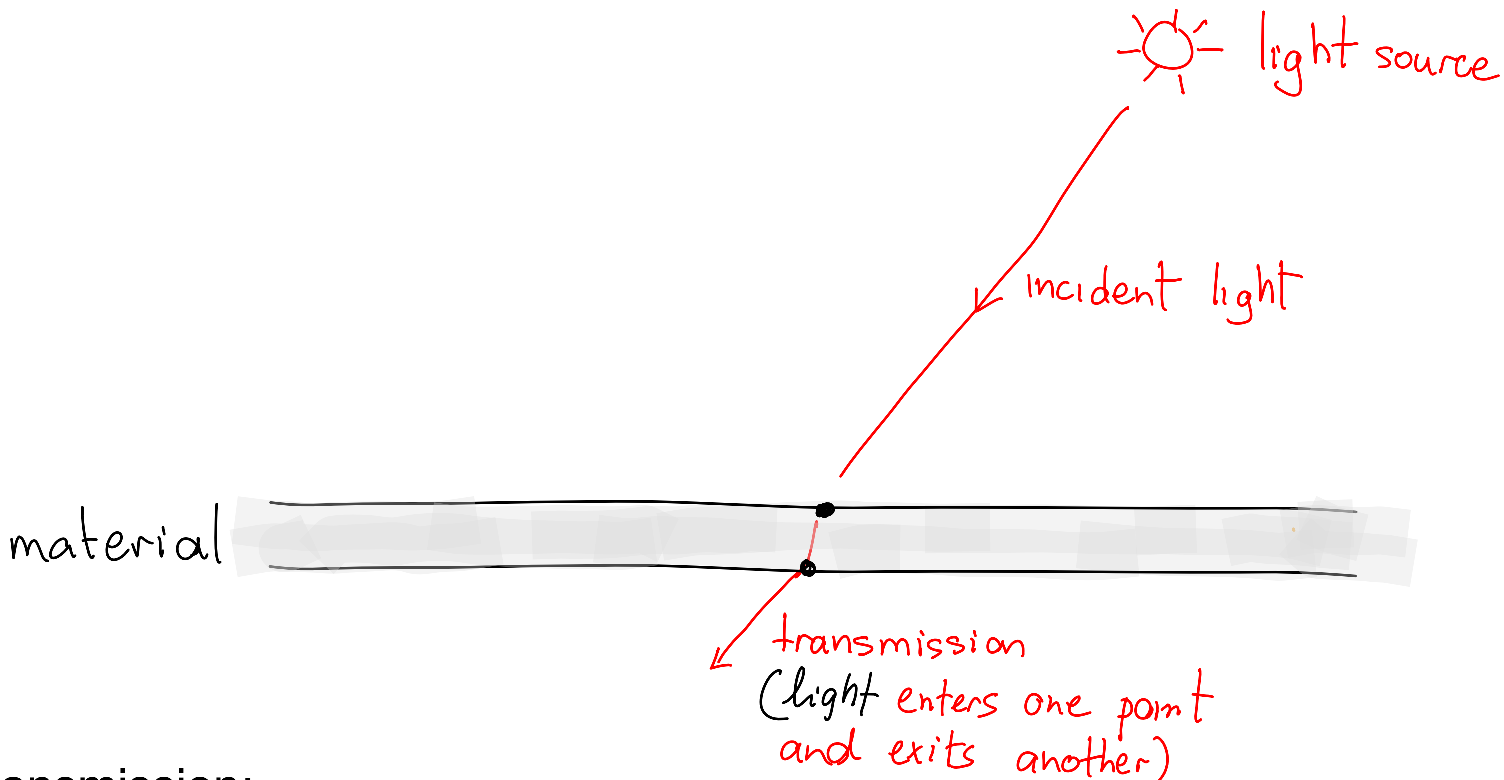


Panjasan, Wikipedia

Specular reflection:

- Represents shiny component of reflected light
- Caused by mirror like reflection off of smooth or polished surfaces (plastics, polished metals, etc)

Modeling Reflection: Transmission

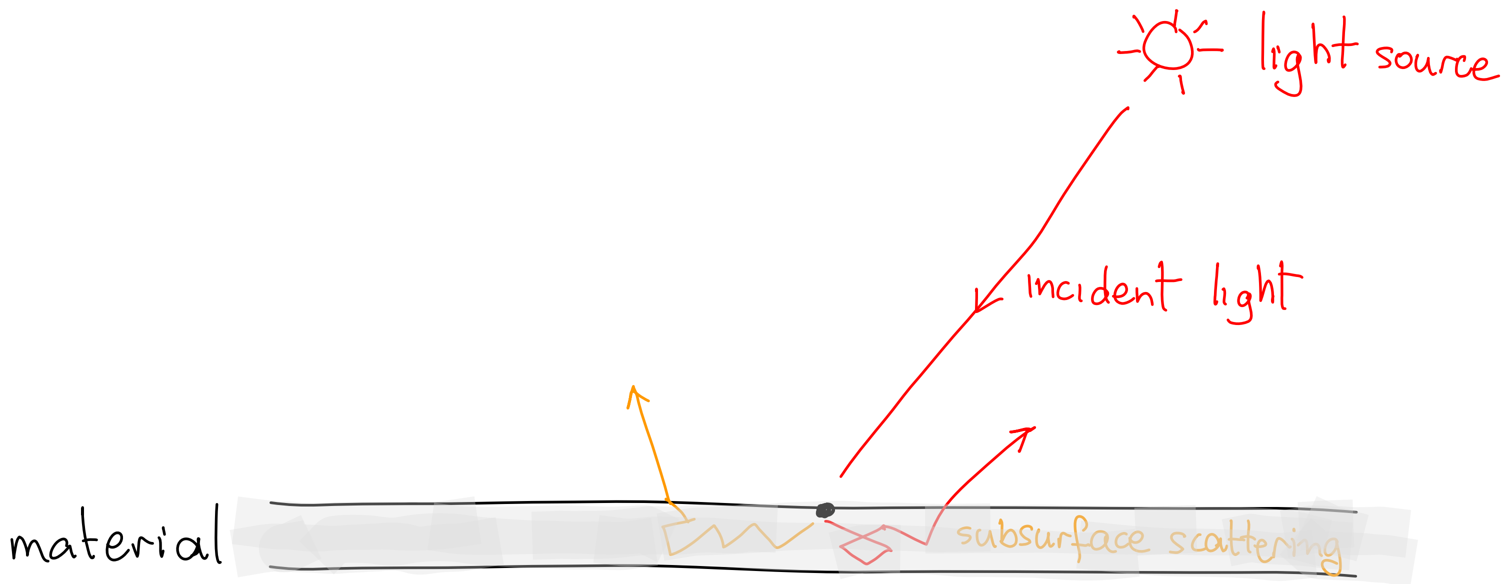


Transmission:

- Caused by materials that are not perfectly opaque
- Examples include glass, water and translucent materials such as skin

Gu et al, EGSR'07

Modeling Reflection: Sub-surface Scattering



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)

Jensen et al, SIGGRAPH'01

Rendering with subsurface scattering (translucent skin)

Jensen et al, SIGGRAPH'01

Rendering with no subsurface scattering (opaque milk)

Jensen et al, SIGGRAPH'01

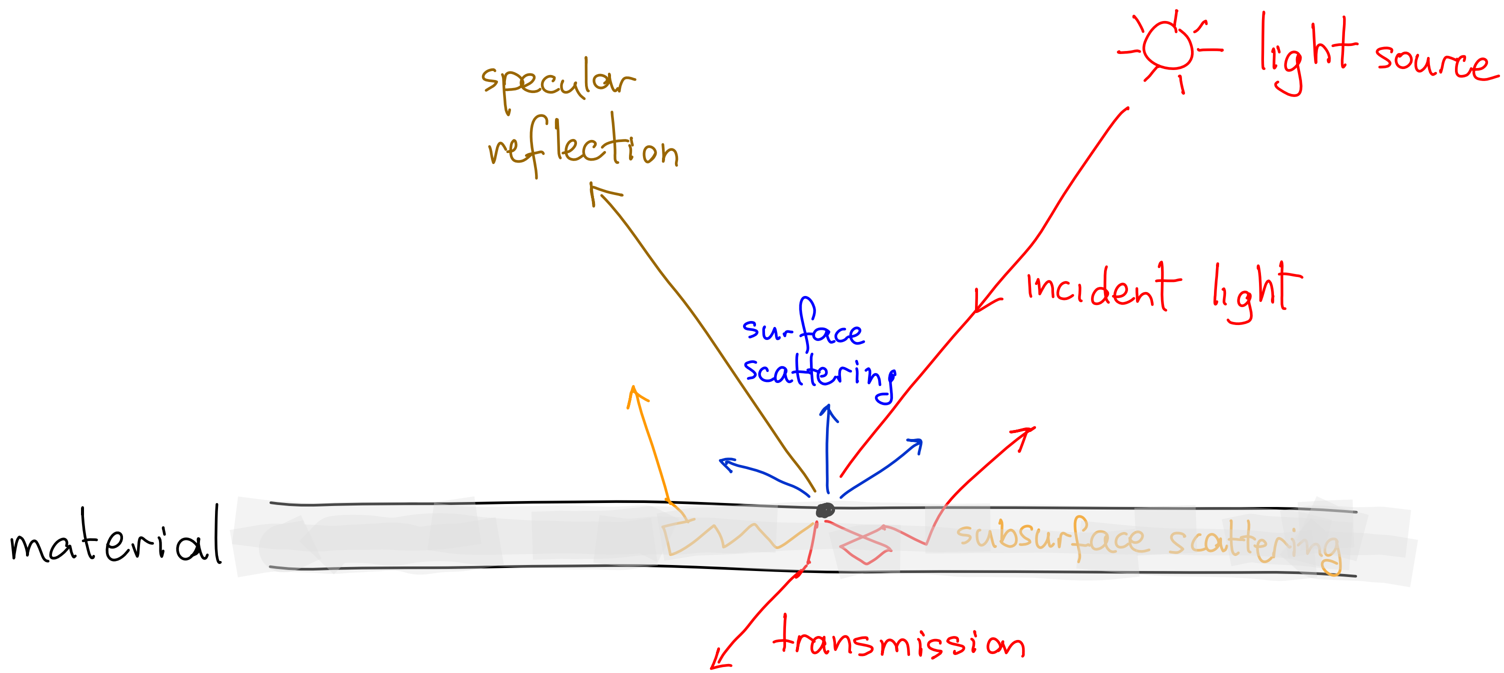
Rendering with subsurface scattering (full milk)

Jensen et al, SIGGRAPH'01

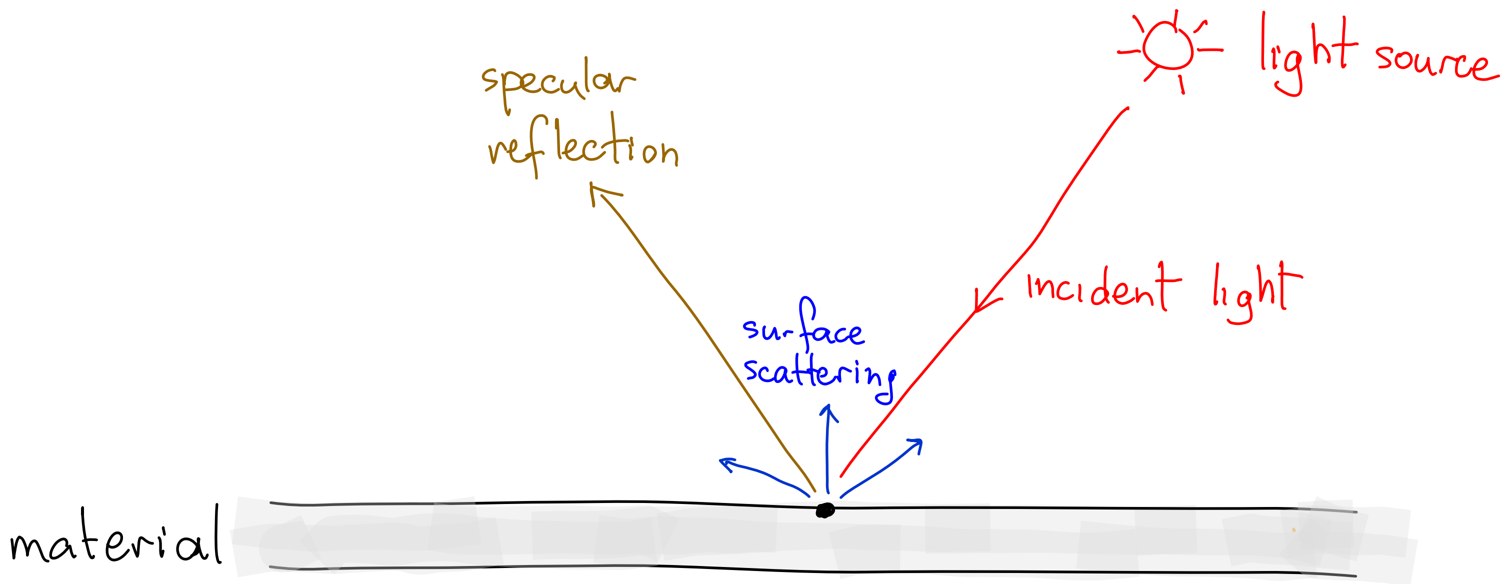
Rendering with subsurface scattering (skim milk)

Jensen et al, SIGGRAPH'01

The Common Modes of "Light Transport"



The Phong Reflectance Model

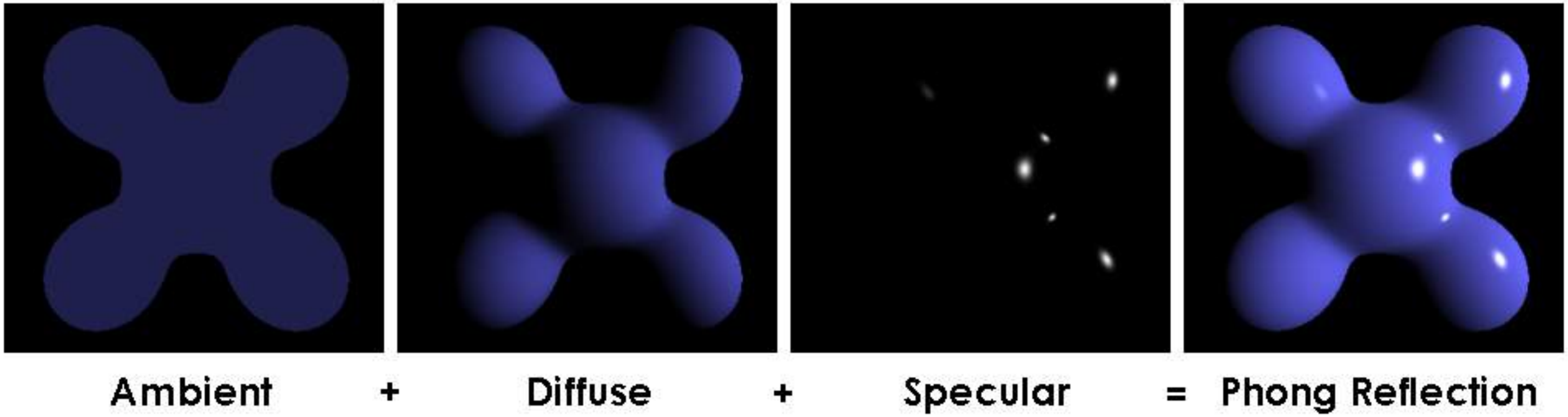


Phong model: A simple computationally efficient model that has 3 components:

- Diffuse
- Ambient
- Specular

The Phong Reflectance Model

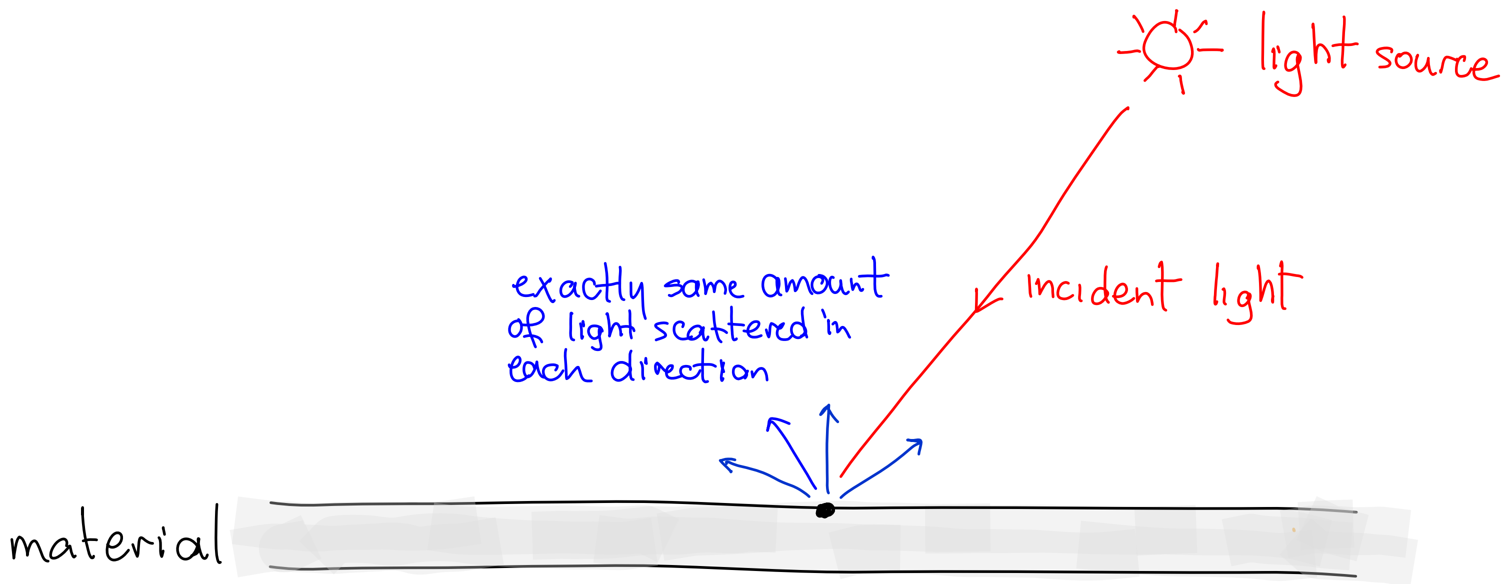
Brad Smith, Wikipedia



Phong model: A simple computationally efficient model that has 3 components:

- Diffuse
- Ambient
- Specular

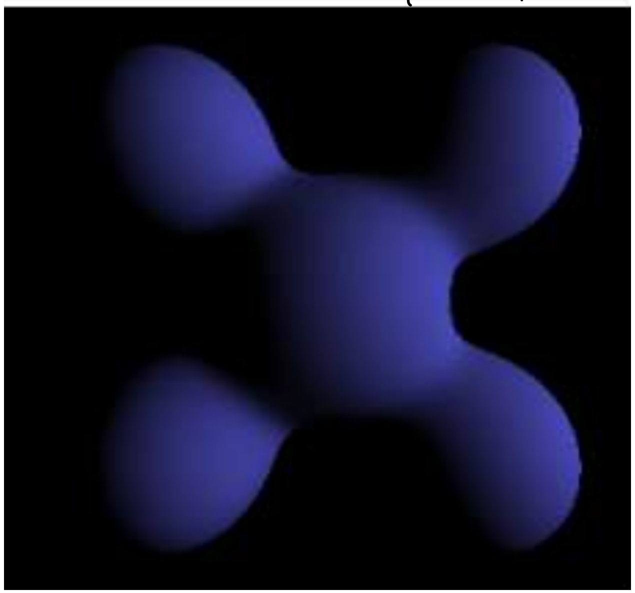
Phong Reflection: The Diffuse Component



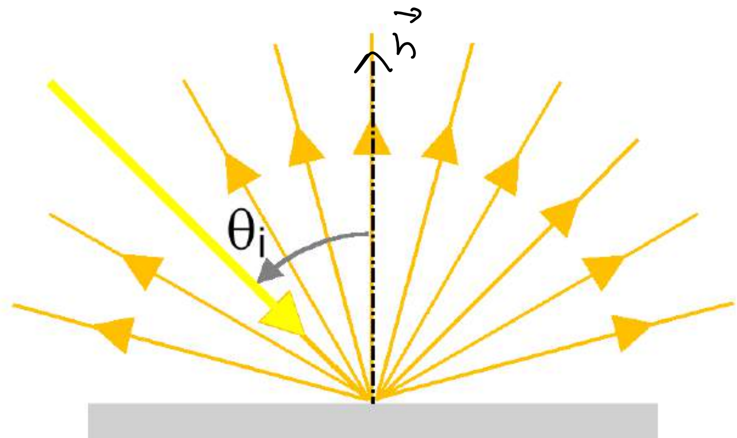
- A diffuse point looks the same from all viewing positions
- Simplest case: a single, point light source

Phong Reflection: The Diffuse Component

Brad Smith, Wikipedia



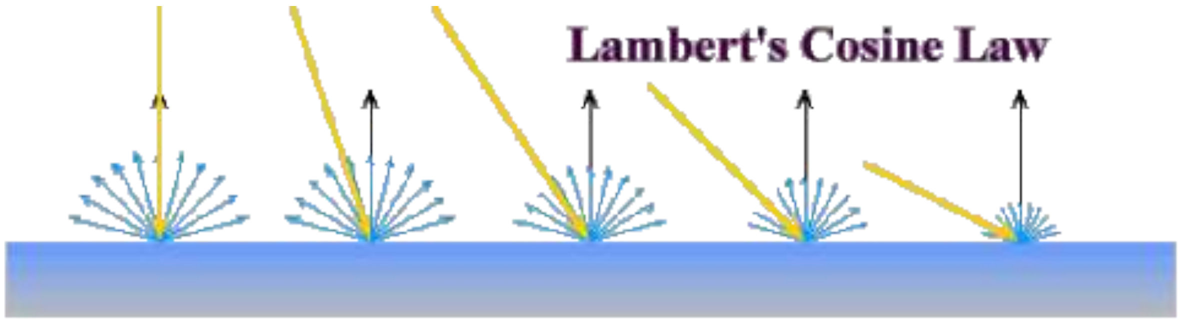
θ_i: angle of incidence



Panjasan, Wikipedia

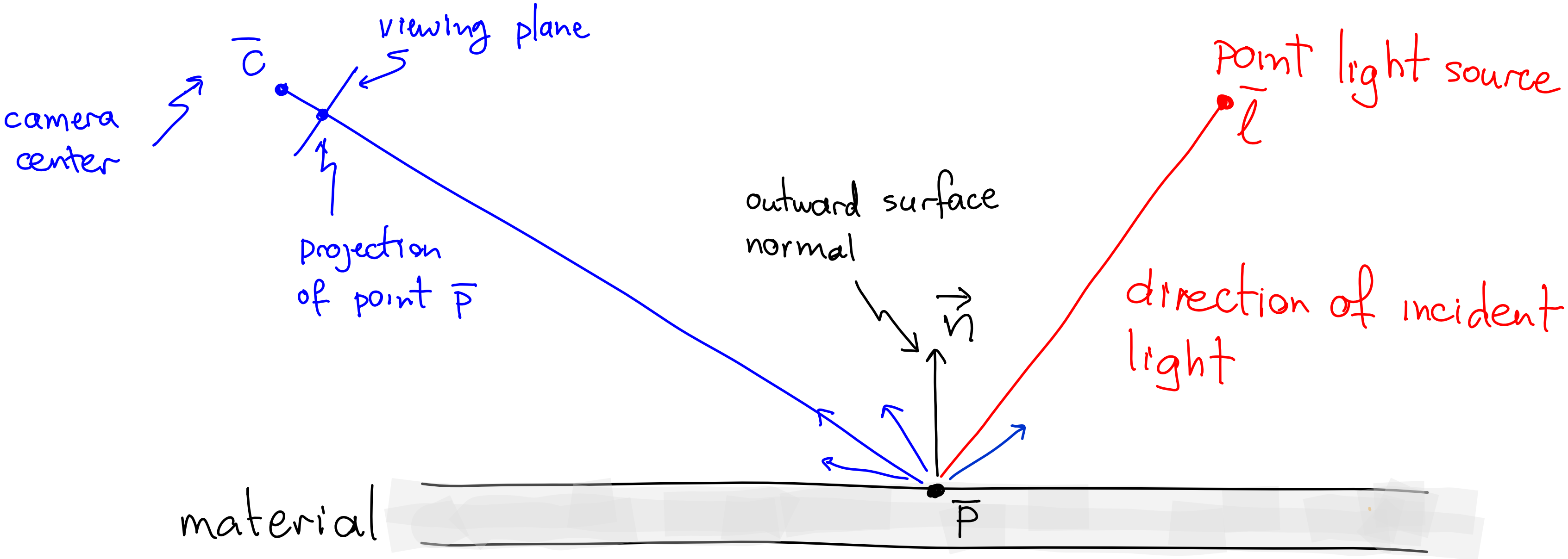
- A diffuse point looks the same from all viewing positions
- Simplest case: a single, point light source

Lambert's Cosine Law



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

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

fraction of light reflected (r_d)

outward unit surface normal ($\vec{n}$)

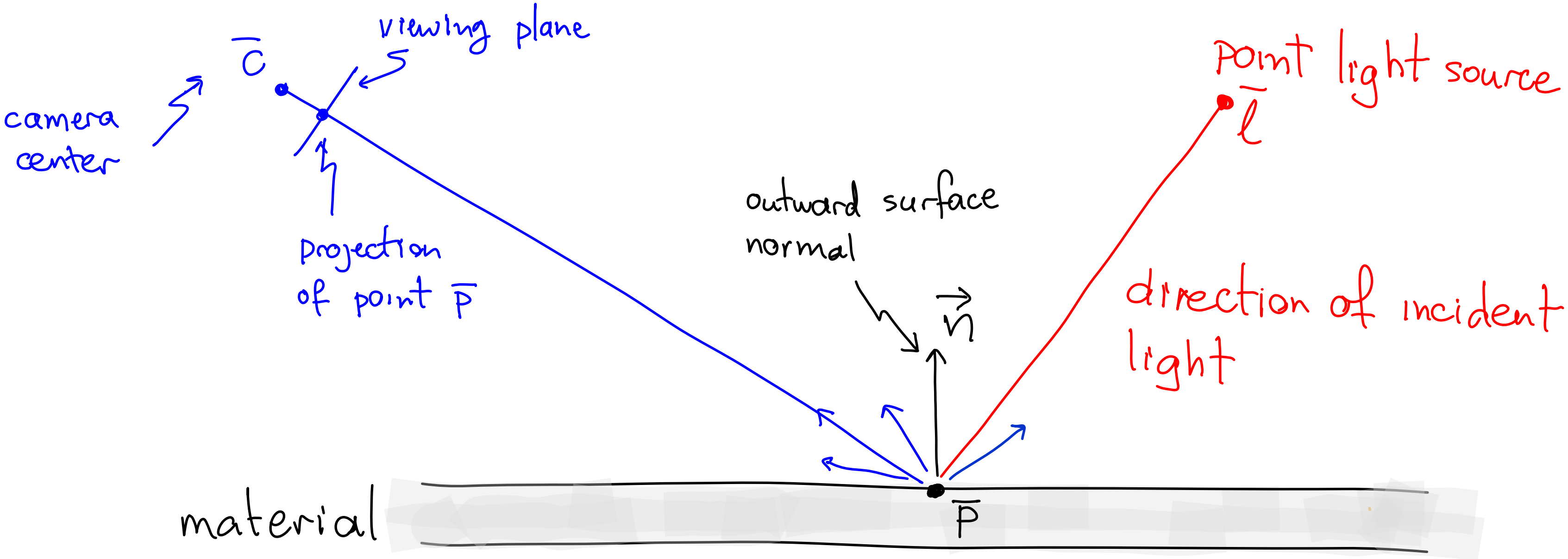
intensity at projection of $\bar{p}$ ($I_{\bar{p}}$)

intensity of source (I)

direction of light source ($\vec{s}$)

$$\vec{s} = \frac{\bar{\ell} - \bar{p}}{\|\bar{\ell} - \bar{p}\|}$$

The Diffuse Component: Basic Equation



- A diffuse point looks the same from all viewing positions

independent of $\bar{c}$

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

intensity at projection of $\bar{p}$

direction of light source $\vec{s} = \dfrac{\bar{\ell} - \bar{p}}{\|\bar{\ell} - \bar{p}\|}$

outward unit surface normal

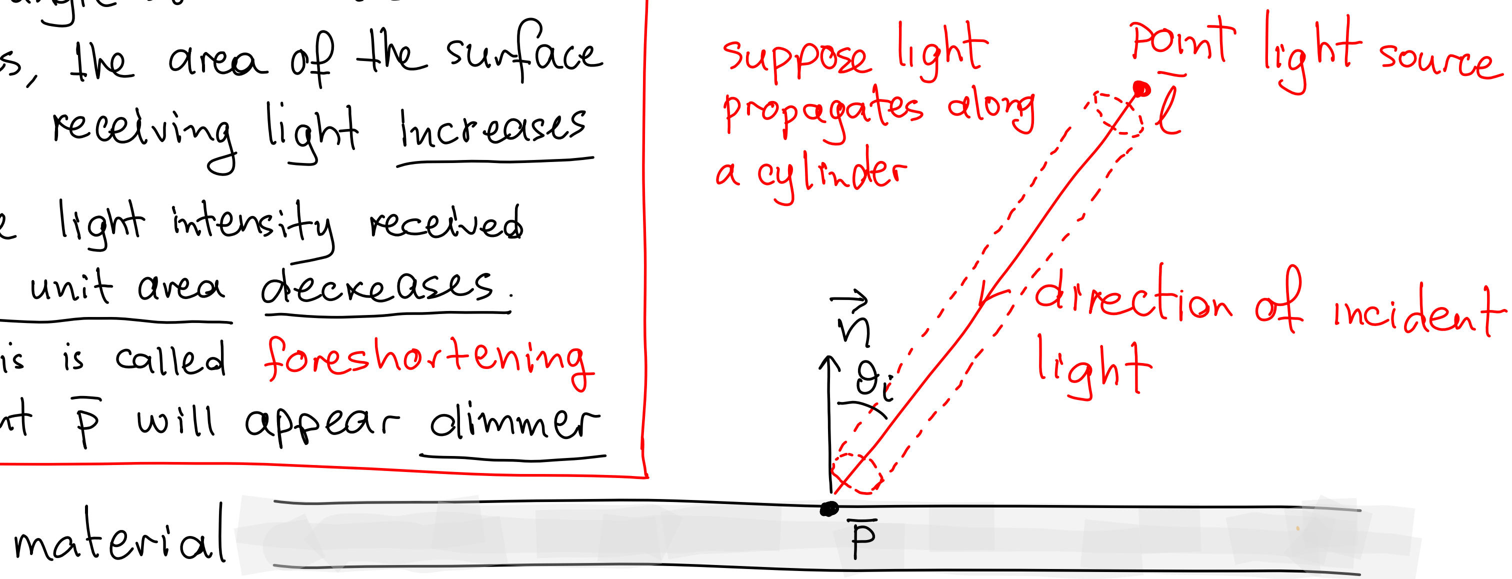
The Diffuse Component: Foreshortening

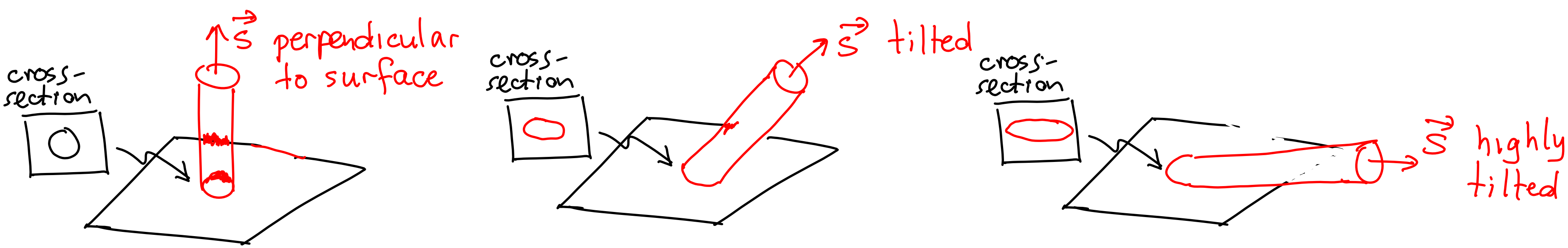
As the angle θ_i between $\vec{s}$ and $\vec{n}$ increases, the area of the surface around $\bar{p}$ receiving light increases

⇒ the light intensity received per unit area decreases.

this is called foreshortening

⇒ point $\bar{p}$ will appear dimmer





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

accounts for dimming due to foreshortening

The Diffuse Component: Foreshortening

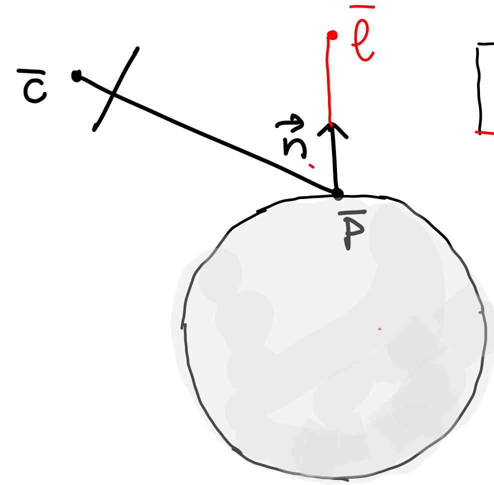
As the angle θ_i between $\vec{s}$ and $\vec{n}$ increases, the area of the surface around $\bar{p}$ receiving light increases

⇒ the light intensity received per unit area decreases.

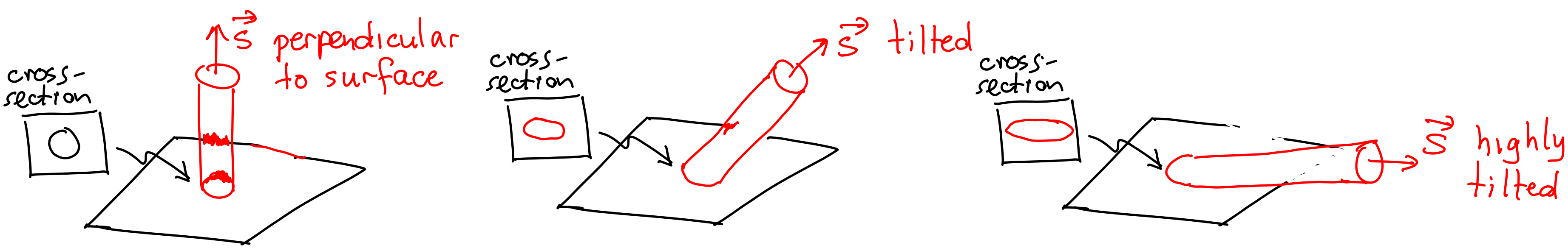
this is called foreshortening

⇒ point $\bar{p}$ will appear dimmer

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



$$I_p = r_d \cdot I$$



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

($\vec{s} \cdot \vec{n}$) accounts for dimming due to foreshortening

The Diffuse Component: Foreshortening

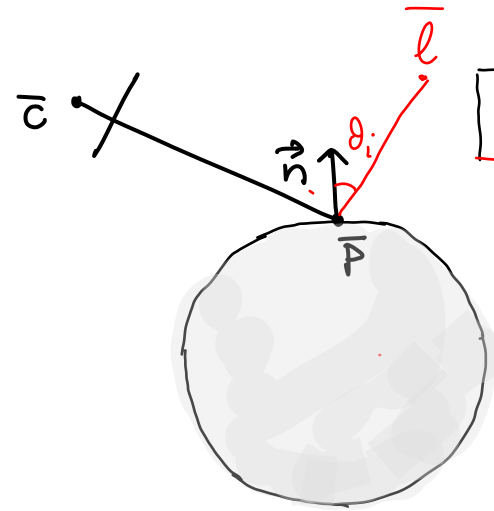
As the angle θ_i between $\vec{s}$ and $\vec{n}$ increases, the area of the surface around $\bar{p}$ receiving light increases

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

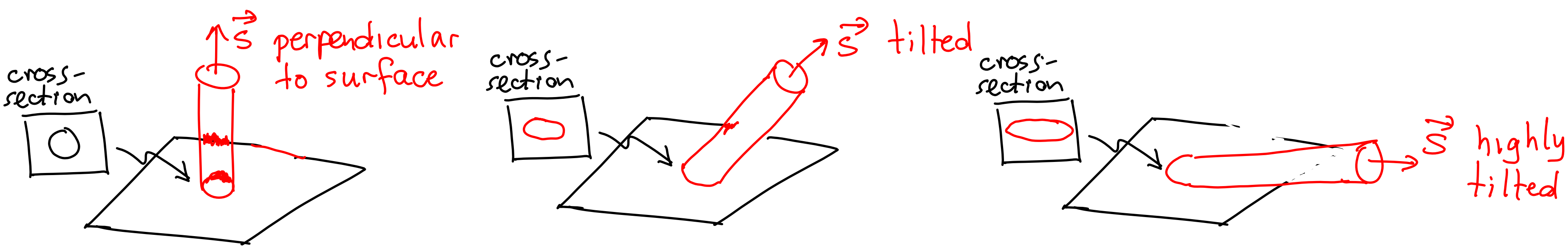
this is called foreshortening

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

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



$$I_p = r_d I \cos\theta_i$$



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

accounts for dimming due to foreshortening

The Diffuse Component: Foreshortening

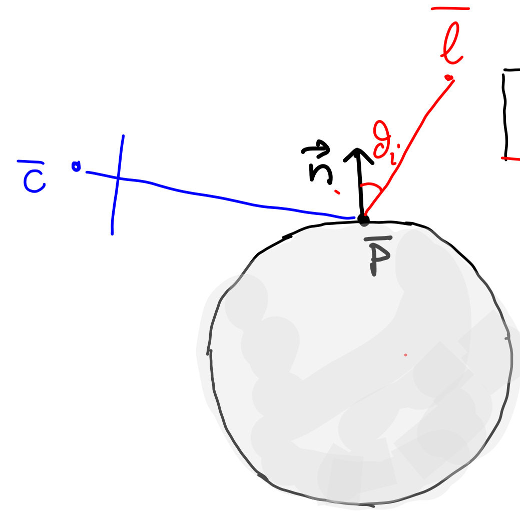
As the angle θ_i between $\vec{s}$ and $\vec{n}$ increases, the area of the surface around $\bar{p}$ receiving light increases

⇒ the light intensity received per unit area decreases.

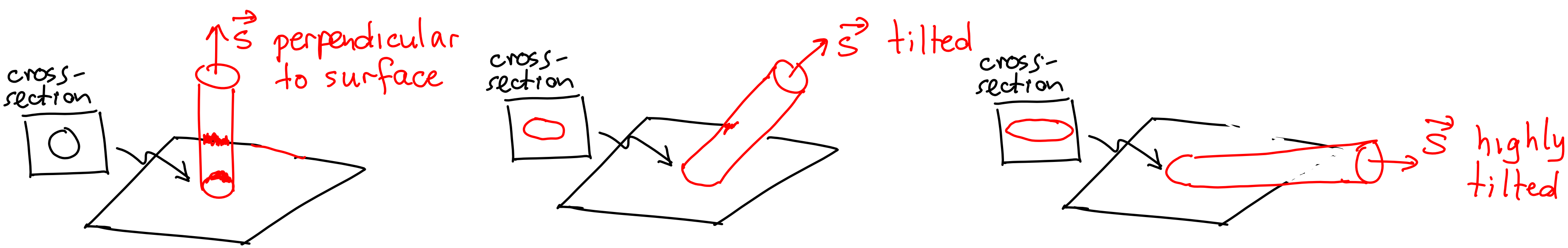
this is called foreshortening

⇒ point $\bar{p}$ will appear dimmer

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



$$I_p = r_d \, I \cos\theta_i$$



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

$\vec{s}\cdot\vec{n}$ accounts for dimming due to foreshortening

The Diffuse Component: Self-Shadowing

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

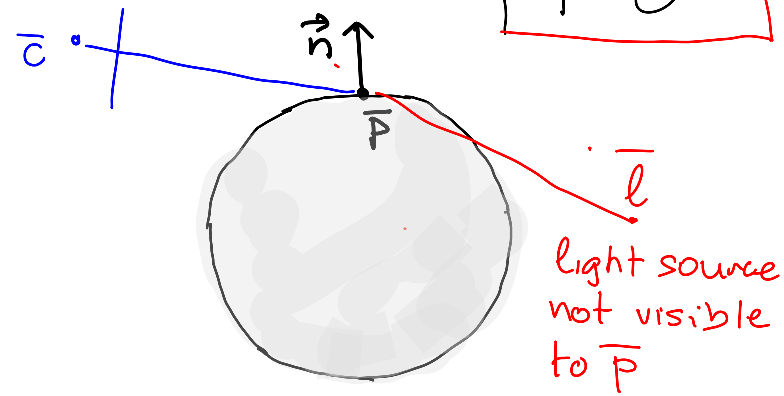
$\Rightarrow$ the light intensity received per unit area decreases.

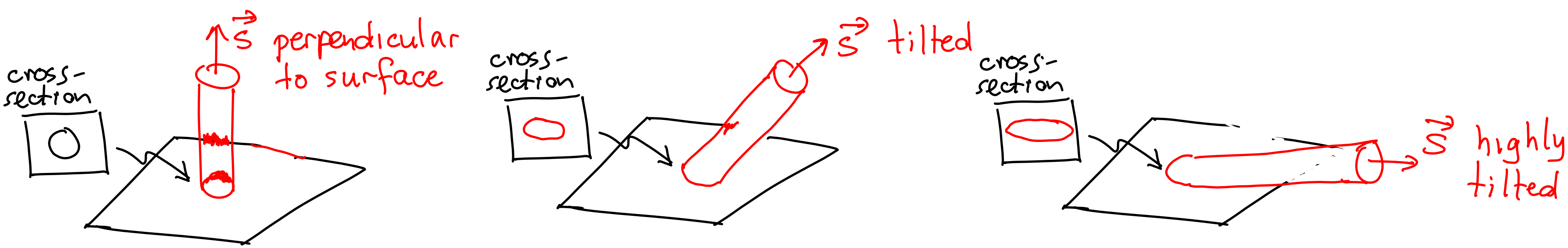
this is called foreshortening

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

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

$I_p = 0$

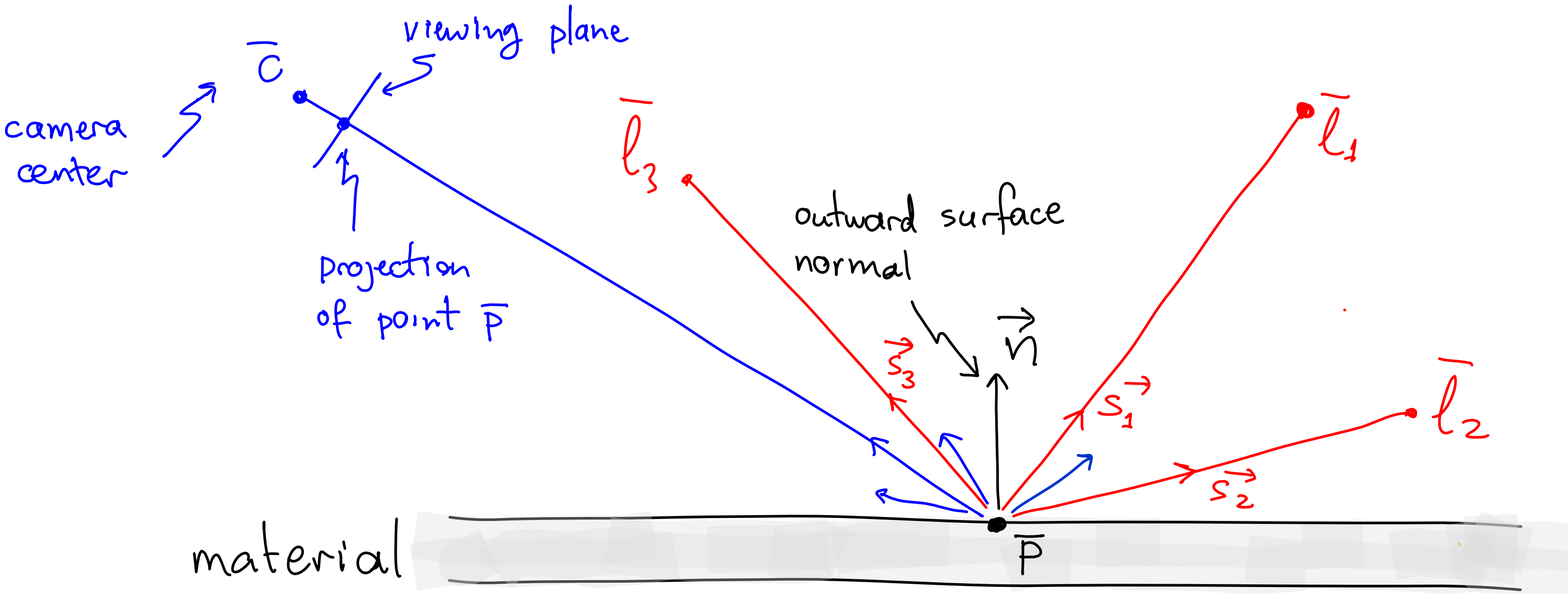




$$I_{\bar{p}} = r_d \cdot I \cdot \max(0, \vec{s} \cdot \vec{n})$$

max: accounts for cases where light source not visible

The Diffuse Component: Multiple Lights



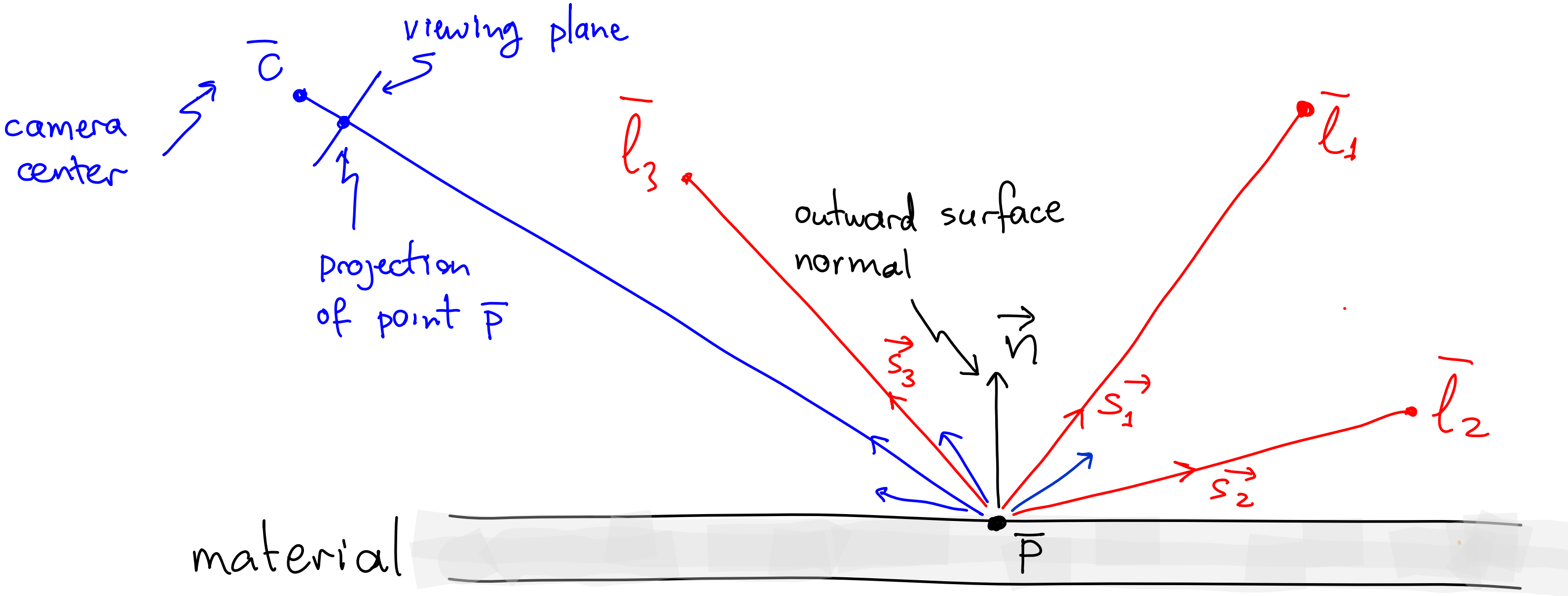
- A diffuse point looks the same from all viewing positions
- When the scene is illuminated by many point sources, we just sum up their contributions to the diffuse component

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

intensity at projection of $\bar{p}$

intensity of source i

The Diffuse Component: Incorporating Color



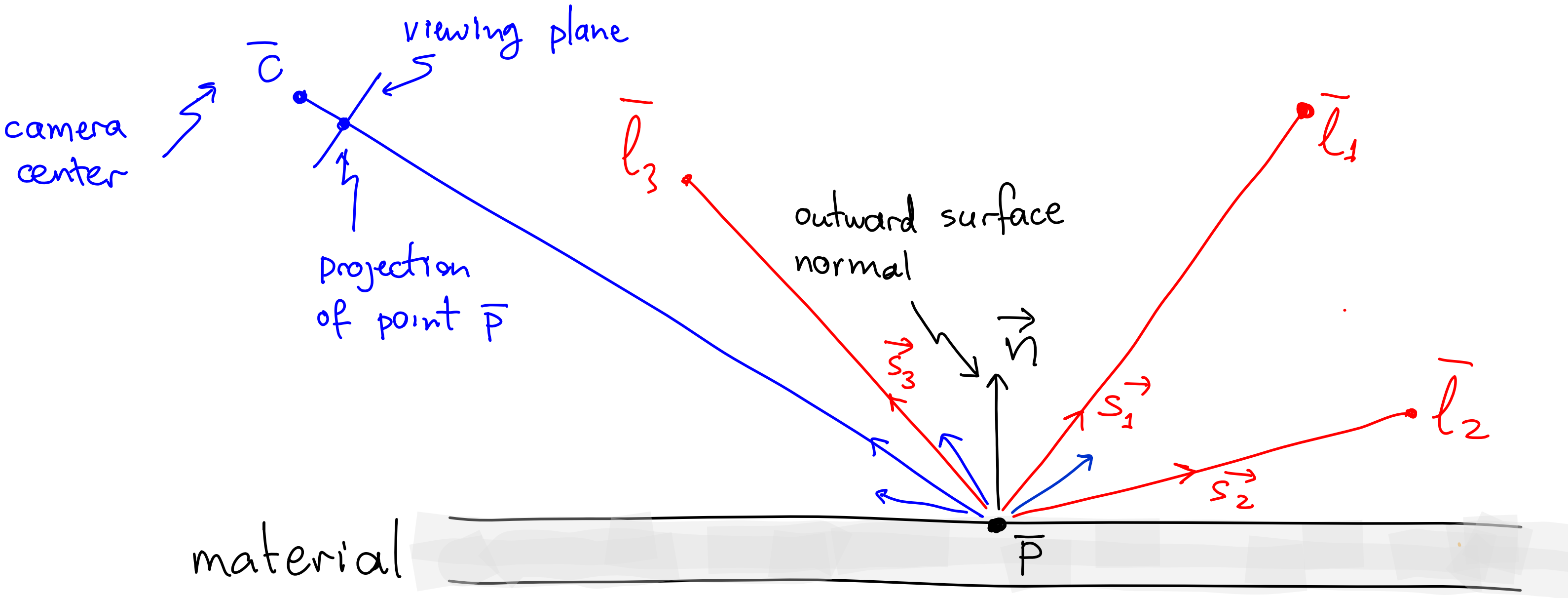
- A diffuse point looks the same from all viewing positions
- Coloured sources and coloured objects are handled by considering the RGB components of each colour separately

$$I_{\bar{p},q} = r_{d,q} \sum_i I_{i,q} \max(0, \vec{s}_i \cdot \vec{n}) \qquad q = R, G, B$$

$I_{\bar{p},q}$: intensity of color component q at projection of $\bar{p}$

$I_{i,q}$: intensity of color component q for light source i

The Diffuse Component: General Equation



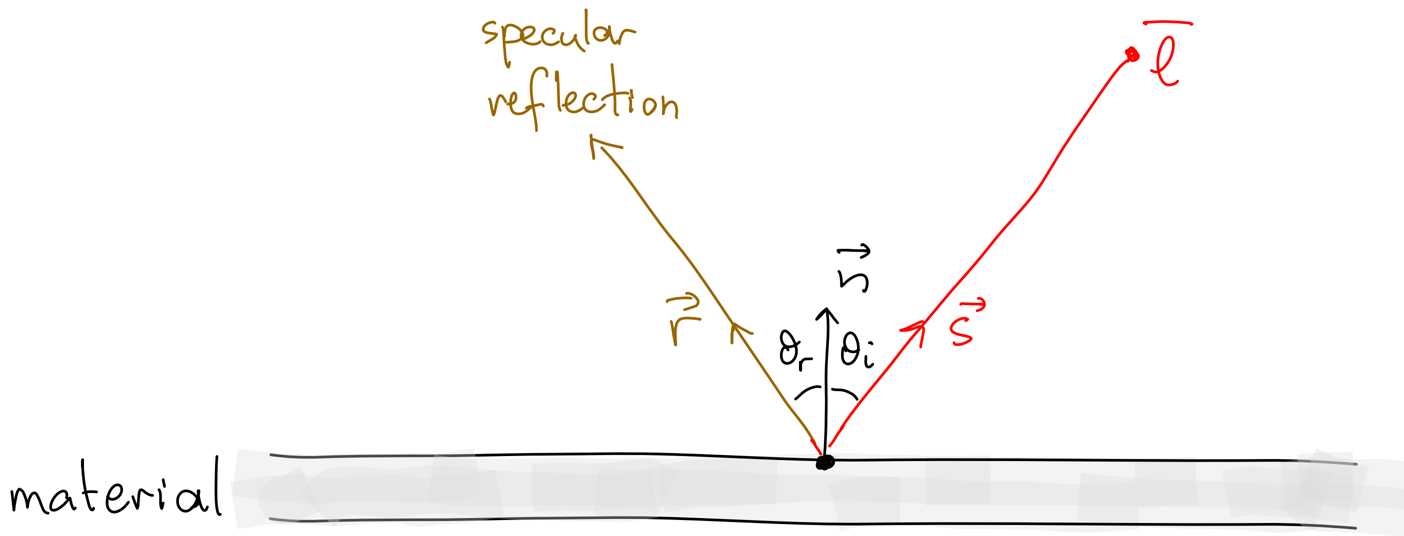
Putting it all together:

$$I_{\bar{p},q} = r_{d,q} \sum_i I_{i,q} \max(0, \vec{s}_i \cdot \vec{n})$$

Specular Reflection

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:

$\vec{s}$

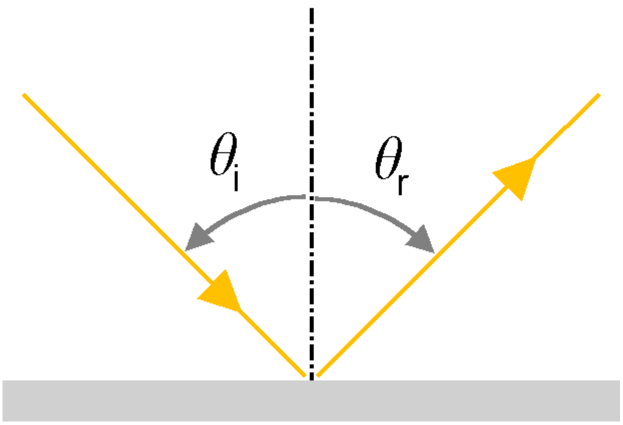
$\vec{r}$

$$angle(\vec{n},\vec{s}) = angle(\vec{n},\vec{r})$$

θ_i θ_r

The Ideal Specular Component

Romeiro et al, ECCV'08



Panjasan, Wikipedia

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

$\vec{s}$

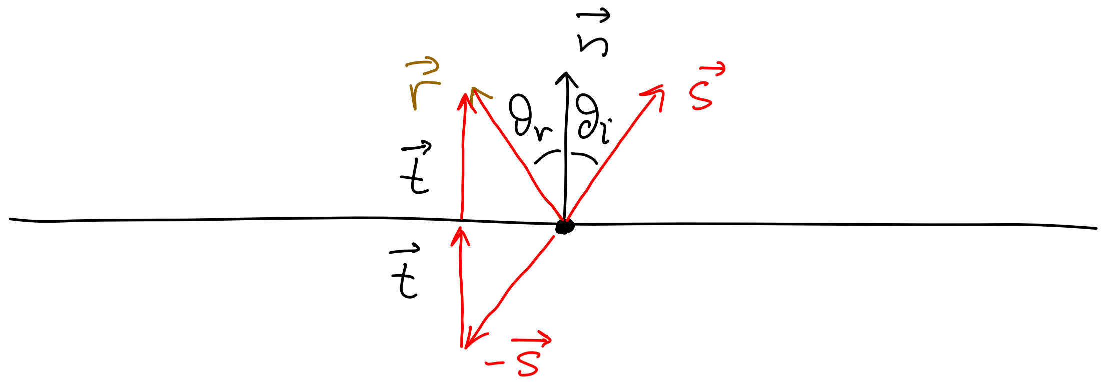
$\vec{r}$

$$angle(\vec{n}, \vec{s}) = angle(\vec{n}, \vec{r})$$

$$\theta_i \qquad\qquad \theta_r$$

Q: How can we express $\vec{r}$ in terms of $\vec{n}, \vec{s}$?

The Ideal Specular Component



$$\vec{r} = -\vec{s} + 2\vec{t}$$

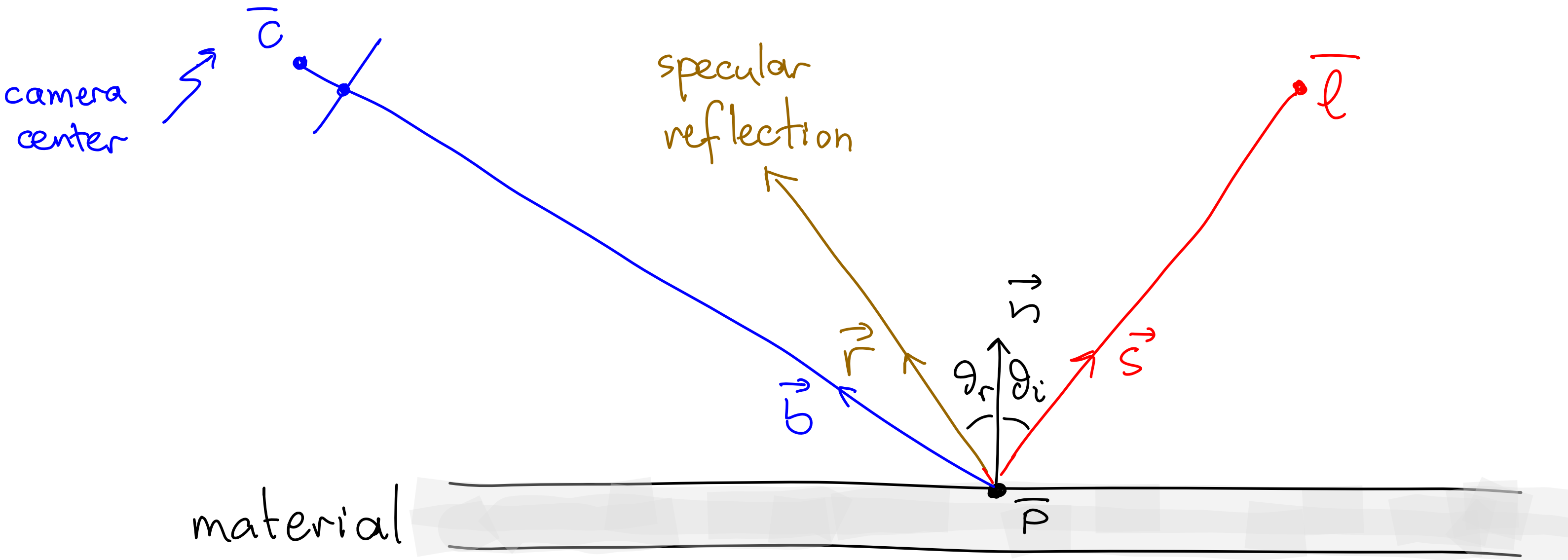
$\vec{t}$ = projection of vector $\vec{s}$ onto vector $\vec{n}$

$$= (\vec{n} \cdot \vec{s})\,\vec{n}$$

$$\Rightarrow \quad \boxed{\vec{r} = -\vec{s} + 2(\vec{n} \cdot \vec{s})\,\vec{n}}$$

Q: How can we express $\vec{r}$ in terms of $\vec{n}, \vec{s}$?

The Ideal Specular Component



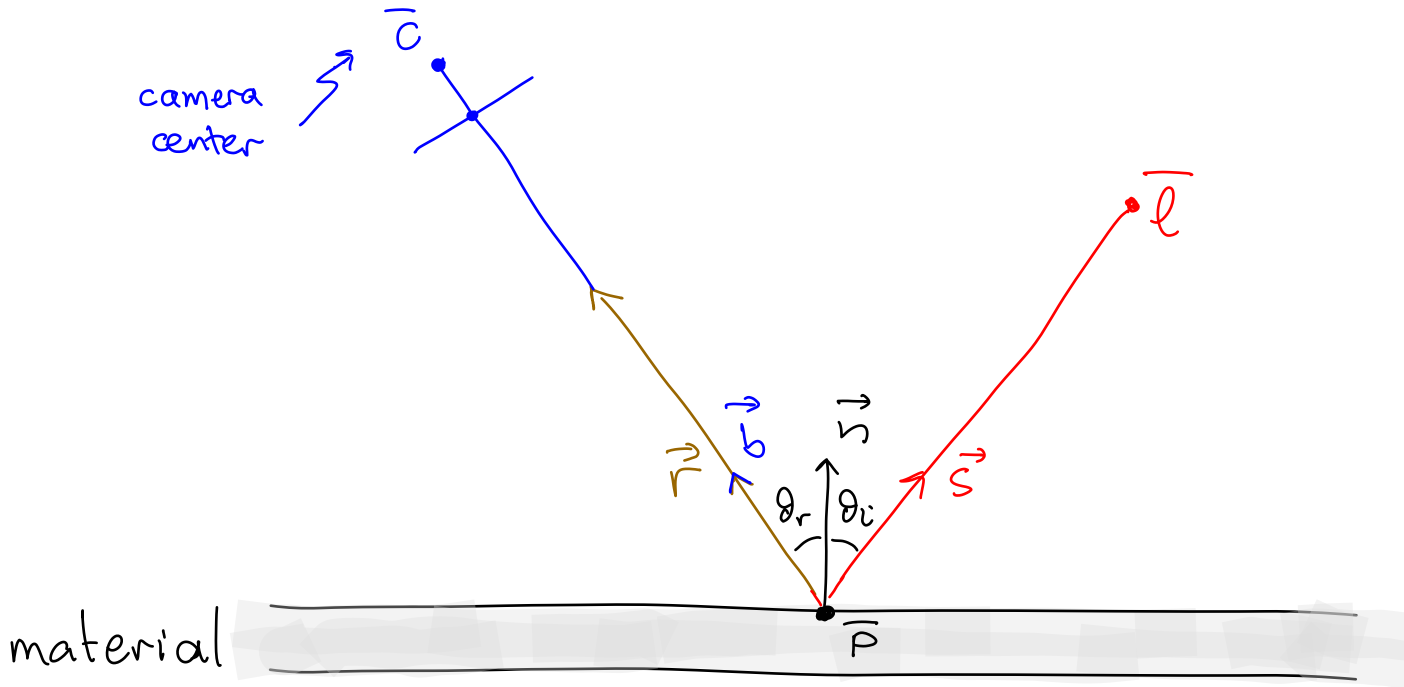
Ideal specular reflection term:

is 1 if and only if camera is along vector $\vec{r}$

$$I = r_s I_s \delta\left(\vec{r} \cdot \vec{b} - 1\right) \quad \text{where} \quad \delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

- r_s: specular reflection coefficient
- I_s: intensity of specular light source
- $\vec{b}$: unit vector in camera direction, $\vec{b} = \dfrac{\bar{c} - \bar{p}}{\|\bar{c} - \bar{p}\|}$

The Ideal Specular Component



Ideal specular reflection term:

is 1 if and only if camera is along vector $\vec{r}$

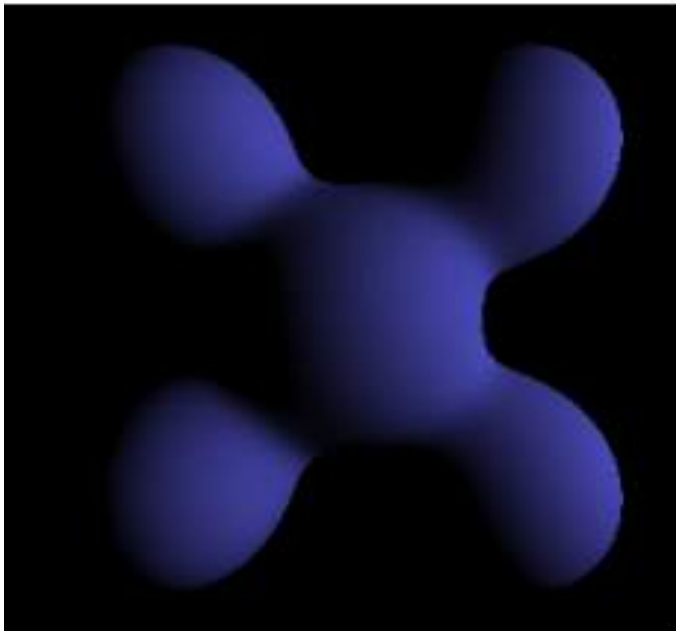
$$I = r_s I_s \delta(\vec{r} \cdot \vec{b} - 1) \quad \text{where} \quad \delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

- r_s: specular reflection coefficient
- I_s: intensity of specular light source
- $\vec{b}$: unit vector in camera direction

$$\vec{b} = \frac{\bar{c} - \bar{p}}{\|\bar{c} - \bar{p}\|}$$

The Ideal Specular Component

Brad Smith, Wikipedia



Ideal specular reflection term:

$$I = r_s \, I_s \, \delta\left(\vec{r} \cdot \vec{b} - 1\right) \quad \text{where} \quad \delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

$\delta(\vec{r} \cdot \vec{b} - 1)$ is 1 if and only if camera is along vector $\vec{r}$

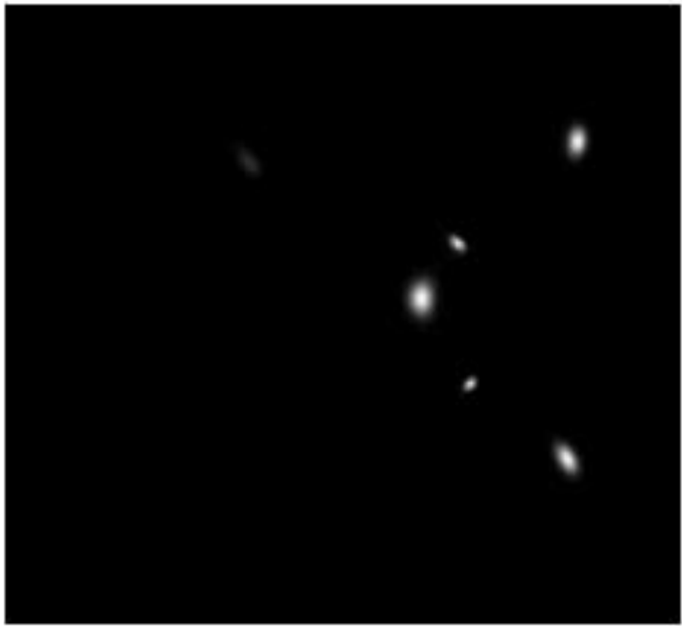
r_s: specular reflection coefficient

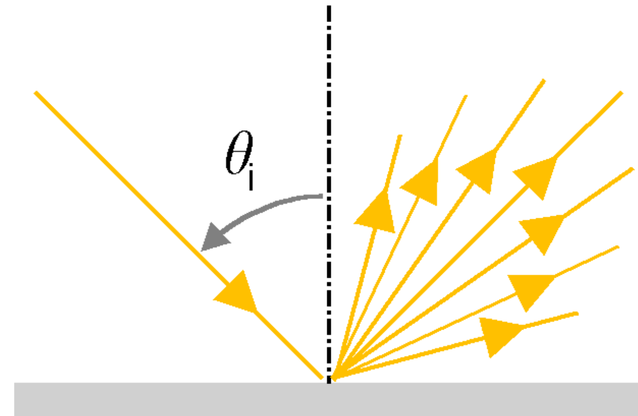
I_s: intensity of specular light source

$\vec{b}$: unit vector in camera direction, $\vec{b} = \frac{\bar{c} - \bar{p}}{\|\bar{c} - \bar{p}\|}$

Phong Reflection: Off-Specular Reflection

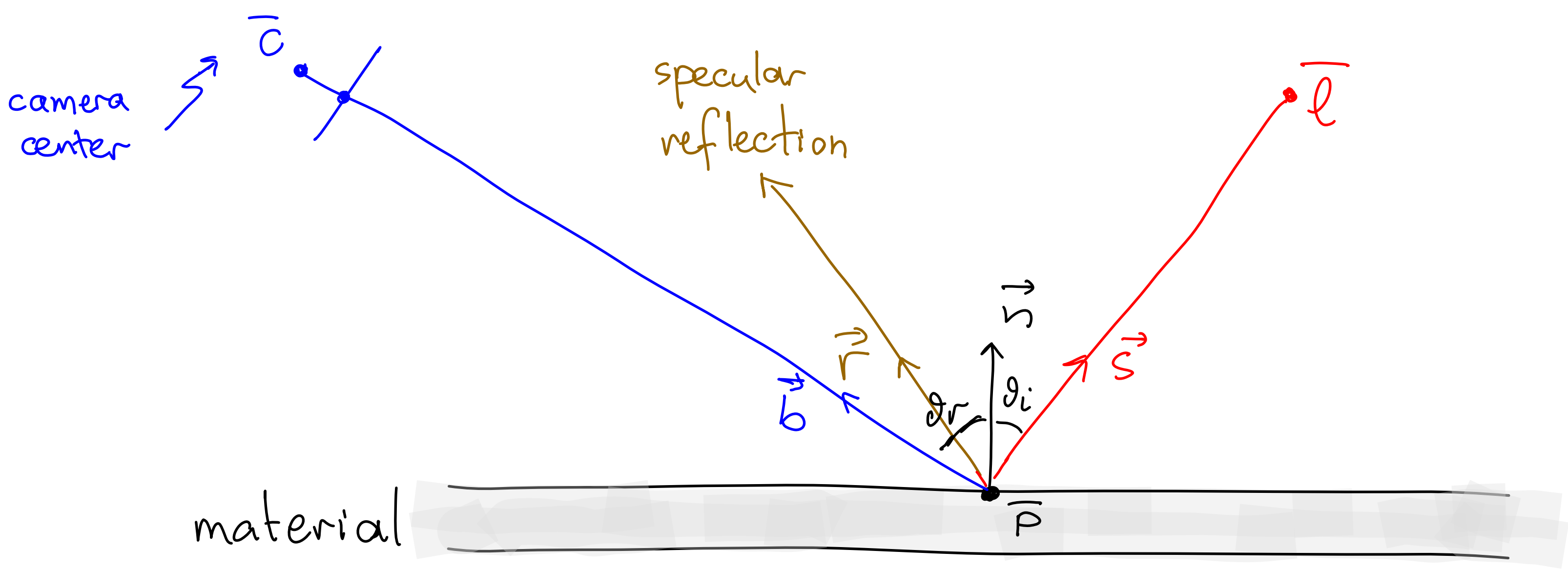
Brad Smith, Wikipedia





Panjasan, Wikipedia

The Specular Component: Basic Equation



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

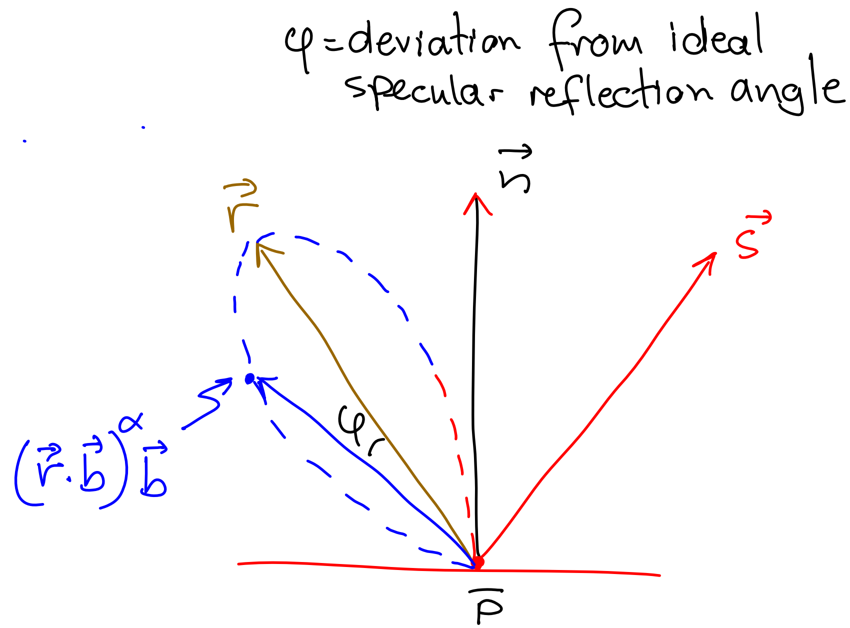
→ Introduce cosine power

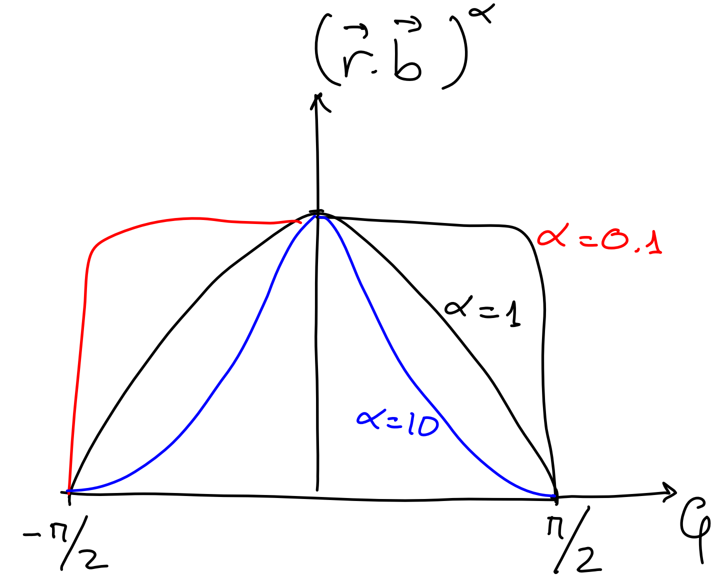
$$I = r_s I_s \max(0, \vec{r} \cdot \vec{b})^{\alpha}$$

$\vec{r} \cdot \vec{b} = 1$ when $\vec{r} = \vec{b}$

α ← when $\alpha \to \infty$ term approaches ideal specular reflection term

The Specular Component: Visualization





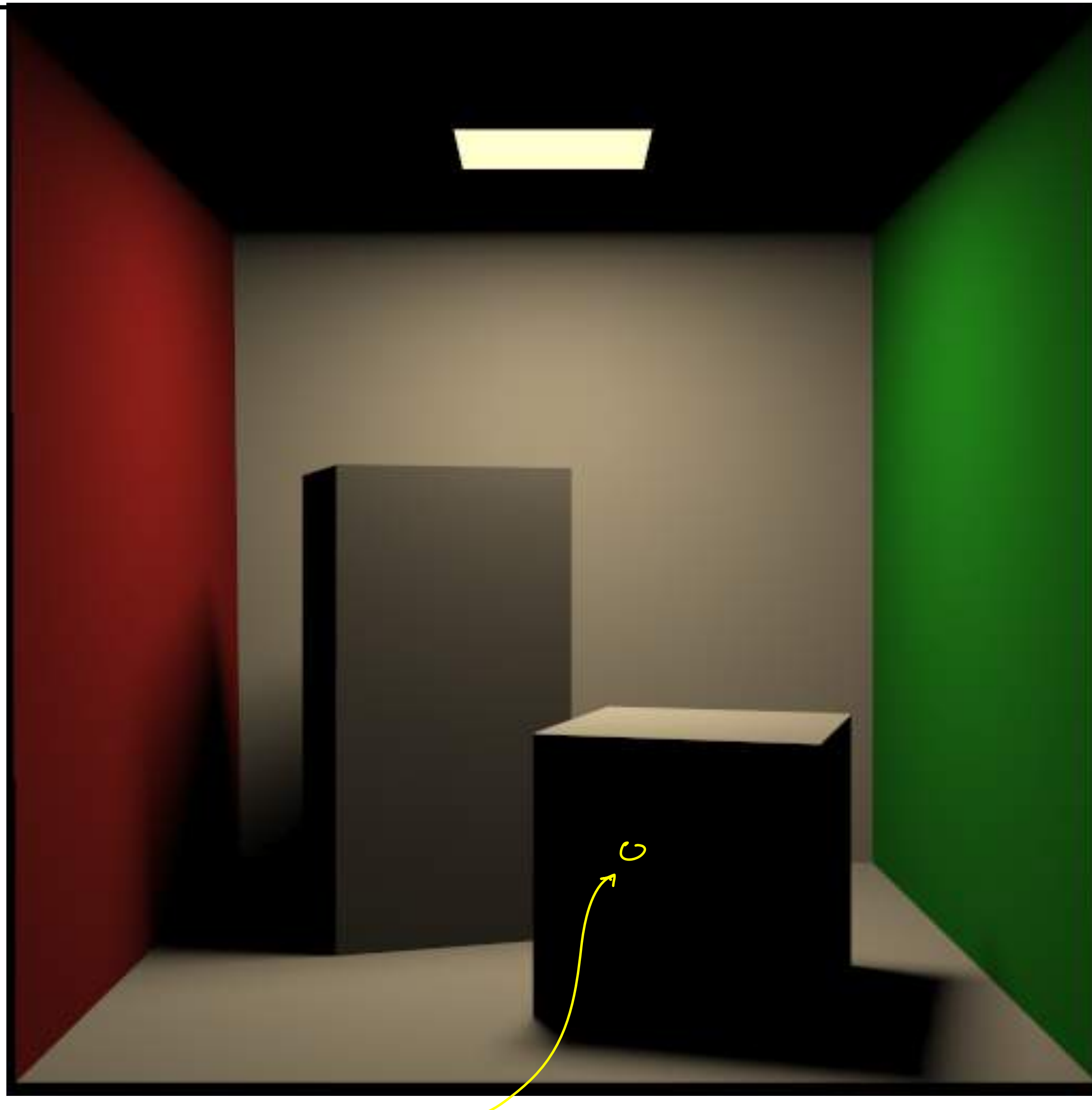
The length of vector $(\vec{r}\cdot\vec{s})^{\alpha}\vec{b}$ represents the contribution of the specular term when the camera is along $\vec{b}$

$$I = r_s I_s \max(0, \vec{r}\cdot\vec{b})^{\alpha}$$

← when α → ∞ term approaches ideal specular reflection term

$\vec{r}\cdot\vec{b}$ = 1 when $\vec{r} = \vec{b}$

Area Light Source, Direct Lighting



"soft" shadows: shadows created because points visible from part of area light source

"hard" shadow: points not visible from light source

Phong Reflection: Ambient Component

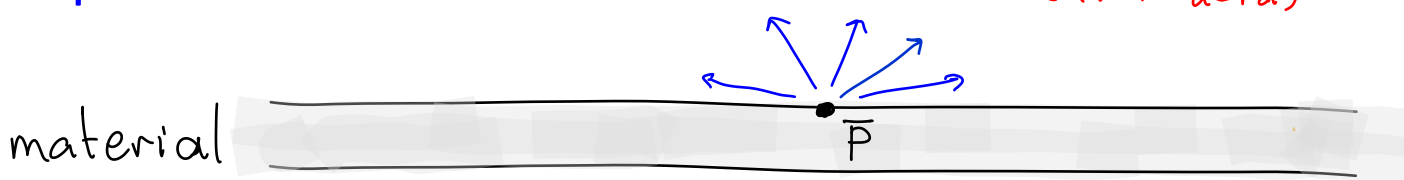
- Solution#2: (simpler) Use an "ambient" term that is independent of any light source or surface normal.
- This term is not meaningful in terms of physics but improves appearance over pure diffuse reflection.

can also have 3 such eqs for R, G, B components

$$I_{\bar{p}} = r_a \cdot I_a$$

r_a: ambient reflection coefficient (often $r_a = r_d$)

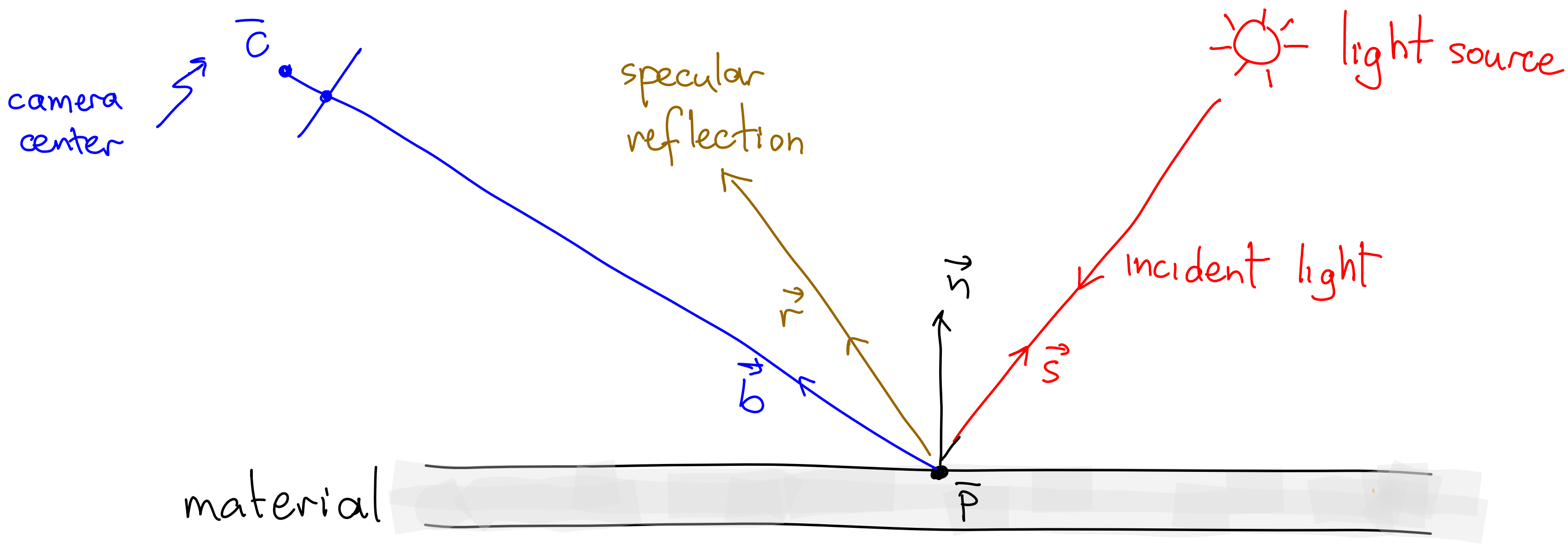
I_a: intensity of ambient illumination



- Diffuse reflectance with a single point light source produces strong shadows
- Surface patches with $\vec{s} \cdot \vec{n} < 0$ are perfectly black

  ➔Looks unnatural

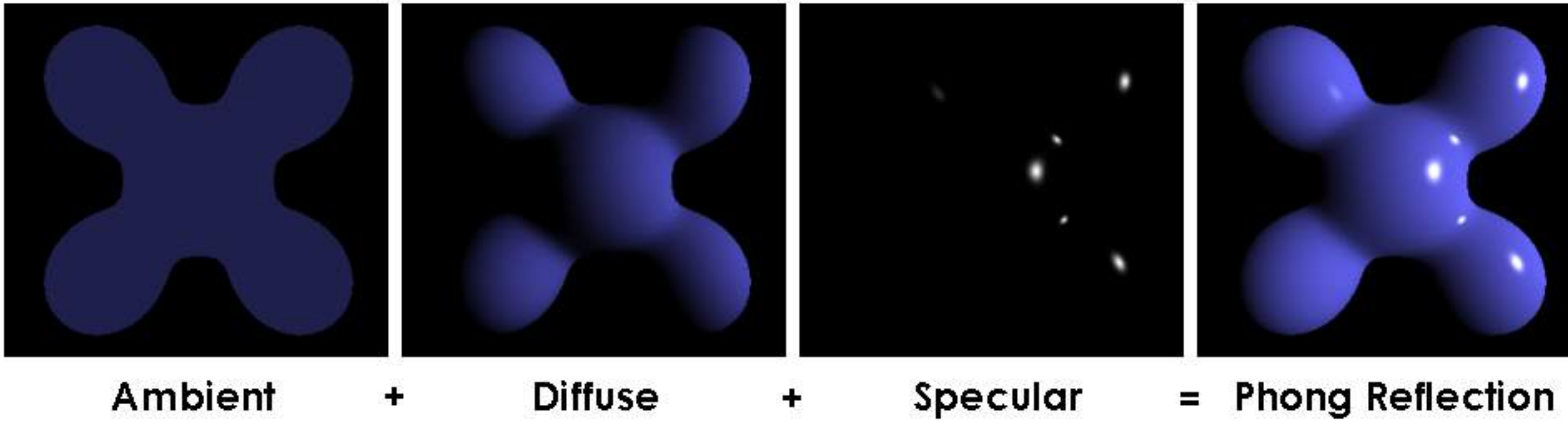
Phong Reflection: The General Equation



$$\underbrace{L(\vec{b}, \vec{n}, \vec{s})}_{\text{intensity at projection of point } \bar{p}} = \underbrace{r_a I_\alpha}_{\text{ambient}} + \underbrace{r_d I_d \max(0, \vec{n} \cdot \vec{s})}_{\text{diffuse}} + \underbrace{r_s I_s \max(0, \vec{r} \cdot \vec{b})^\alpha}_{\text{specular}}$$

Phong Reflection: The General Equation

Brad Smith, Wikipedia



$$\underbrace{L(\vec{b},\vec{n},\vec{s})}_{\text{intensity at projection of point } \bar{p}} = \underbrace{r_a I_a}_{\text{ambient}} + \underbrace{r_d I_d \max(0, \vec{n}\cdot\vec{s})}_{\text{diffuse}} + \underbrace{r_s I_s \max(0, \vec{r}\cdot\vec{b})^{\alpha}}_{\text{specular}}$$

Computing Diffuse Reflection

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

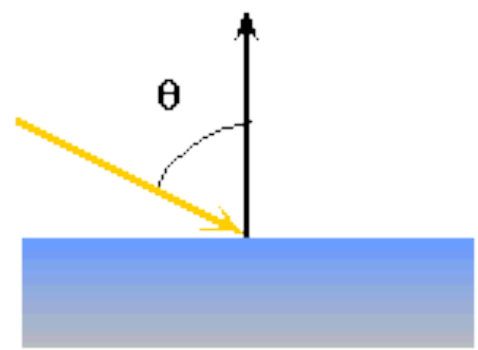
I_{light} : intensity of the incoming light.

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

What is the range of k_d

$$I_{diffuse} = k_d I_{light} \cos\theta$$

$$I_{diffuse} = k_d I_{light} \left(\bar{n} \cdot \bar{l}\right)$$



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…

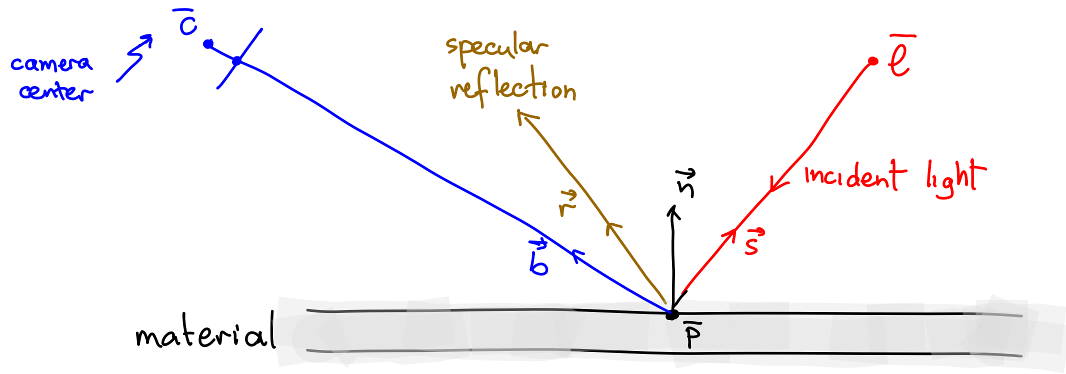
Topic 10:

Shading

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

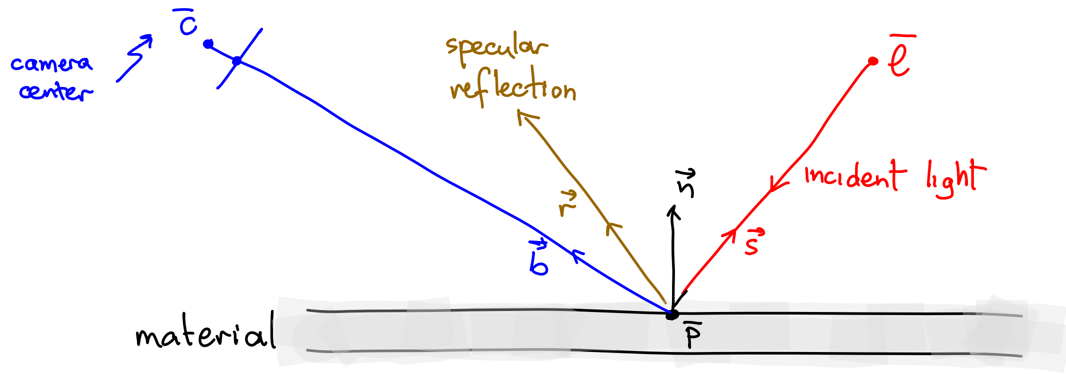
Shading: Motivation

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



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 meshe's projection in accordance with Phong reflection model.

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

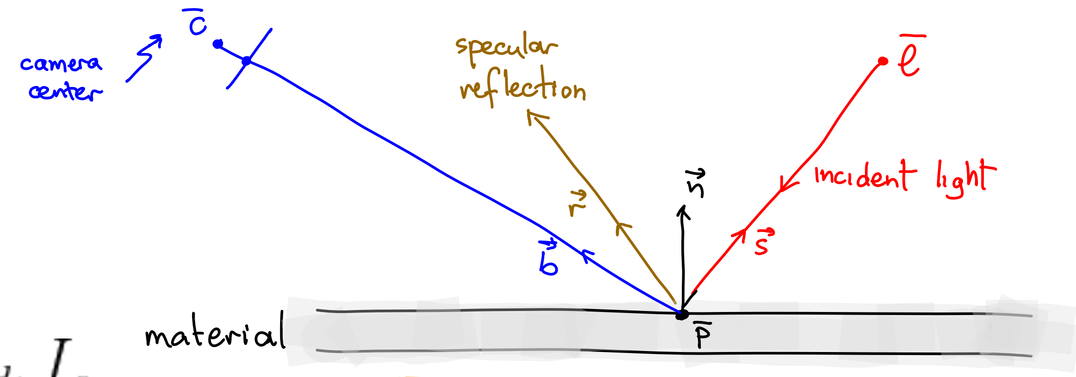
$L(\vec{b},\vec{n},\vec{s})$: intensity at projection of point $\bar{p}$

$r_a I_a$: ambient

$r_d I_d \max(0, \vec{n}\cdot\vec{s})$: diffuse

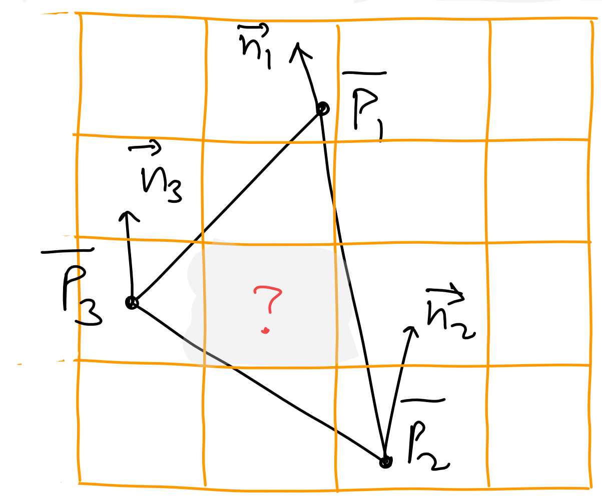
$r_s I_s \max(0, \vec{r}\cdot\vec{b})^{\alpha}$: specular

Shading: Problem Definition



<u>Given</u>

- camera center, $\bar{c}$
- light source position $\bar{l}$
- intensity of ambient, diffuse and specular sources, I_α, I_d, I_s
- reflection coefficients, r_α, r_d, r_s
- specular exponent, α
- normals at $\bar{p}_1, \bar{p}_2, \bar{p}_3$



<u>Goal</u>

Computer colour/intensity at an interior pixel

$$L(\vec{b},\vec{n},\vec{s}) = r_a I_\alpha + r_d I_d \max(0, \vec{n}\cdot\vec{s}) + r_s I_s \max(0, \vec{r}\cdot\vec{b})^\alpha$$

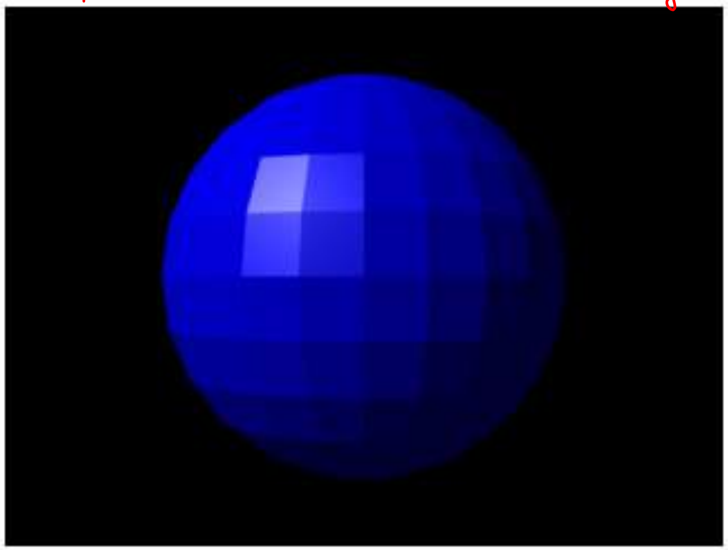
intensity at projection of point $\bar{p}$ — ambient — diffuse — specular

Flat Shading: Main Idea

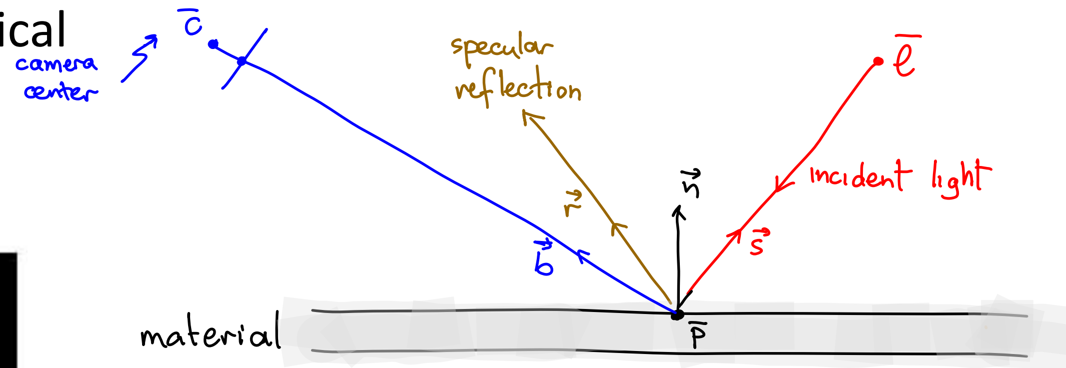
Flat shading

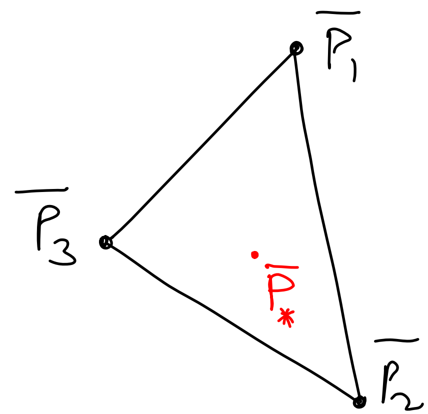
Draw all triangle points $\bar{p}$ with identical colour/intensity

Sphere with flat shading



Jalo, wikipedia





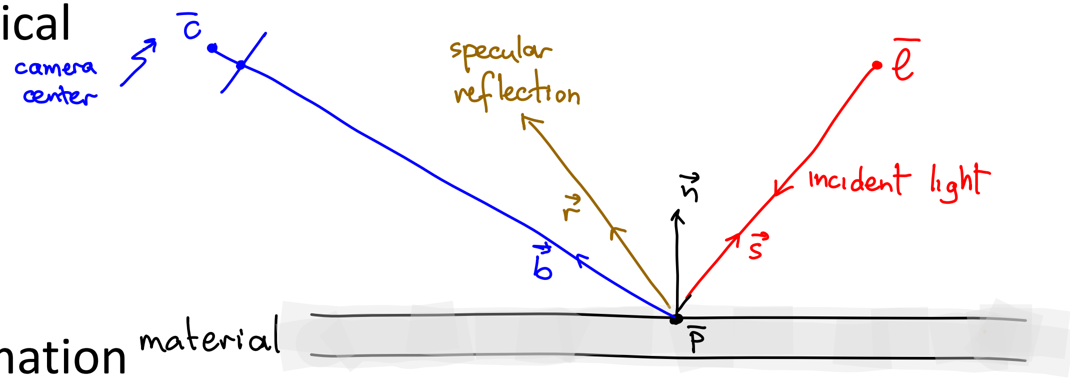
$$L(\vec{b}, \vec{n}, \vec{s}) = r_a I_a + r_d I_d \max(0, \vec{n} \cdot \vec{s}) + r_s I_s \max(0, \vec{r} \cdot \vec{b})^{\alpha}$$

- $L(\vec{b}, \vec{n}, \vec{s})$: intensity at projection of point $\bar{p}$
- $r_a I_a$: ambient
- $r_d I_d \max(0, \vec{n} \cdot \vec{s})$: diffuse
- $r_s I_s \max(0, \vec{r} \cdot \vec{b})^{\alpha}$: specular

Flat Shading: Key Issues

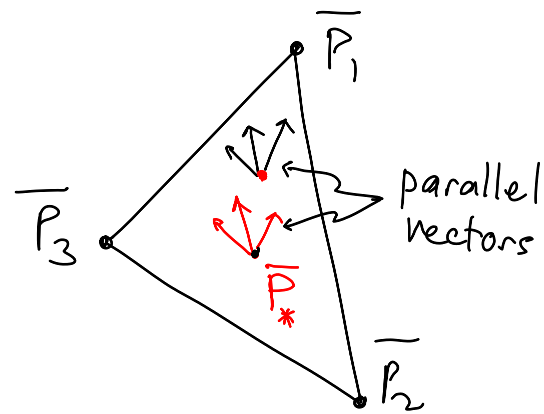
Flat shading

Draw all triangle points $\bar{p}$ 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)



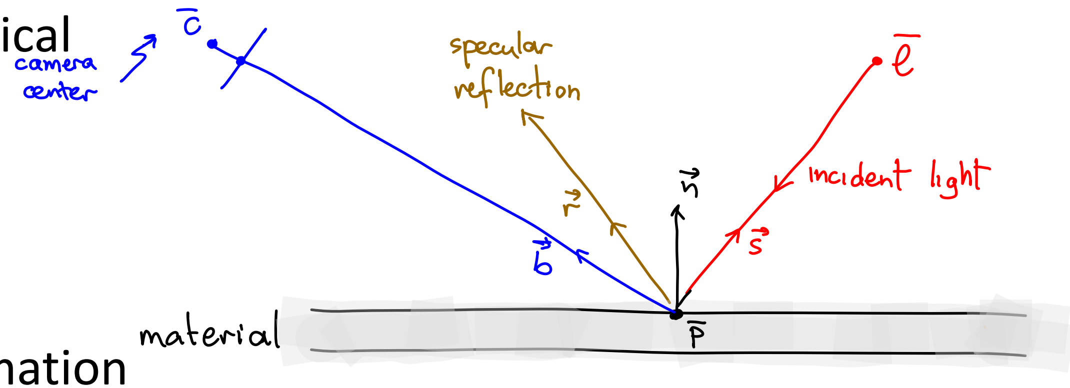
$$L(\vec{b},\vec{n},\vec{s}) = r_a I_a + r_d I_d \max(0,\vec{n}\cdot\vec{s}) + r_s I_s \max(0,\vec{r}\cdot\vec{b})^{\alpha}$$

intensity at projection of point $\bar{p}$ | ambient | diffuse | specular

Flat Shading: Key Issues

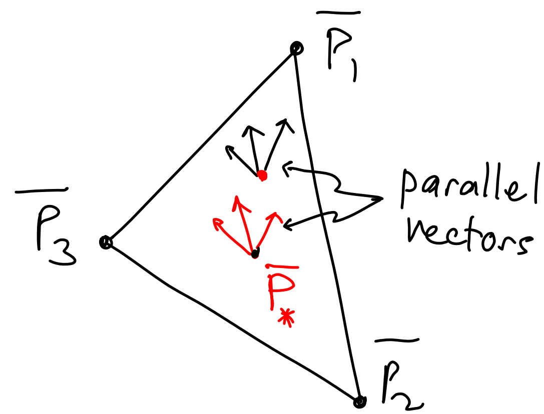
<u>Flat shading</u>

Draw all triangle points $\bar{p}$ 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)

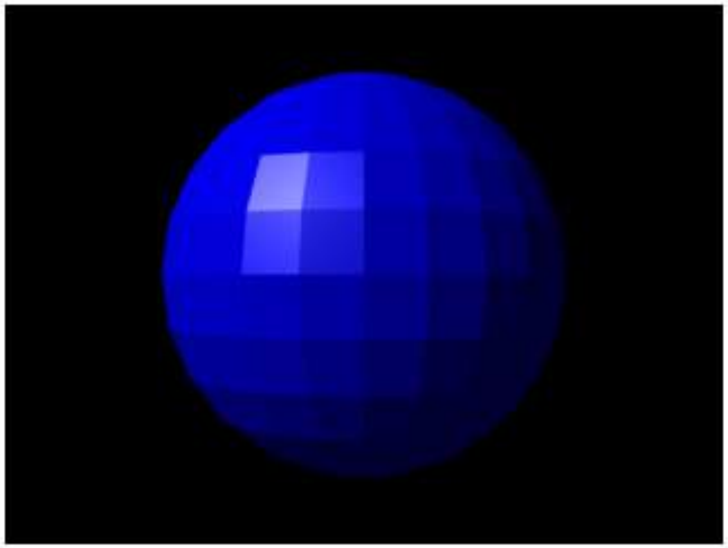


<u>One solution</u>

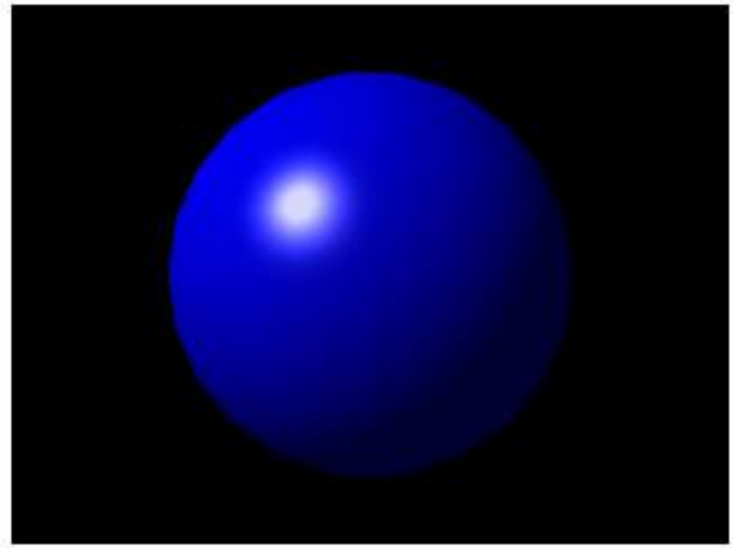
- Since flat shading treats a triangle as a point, use small triangles!

Interpolated Shading

Jalo, Wikipedia

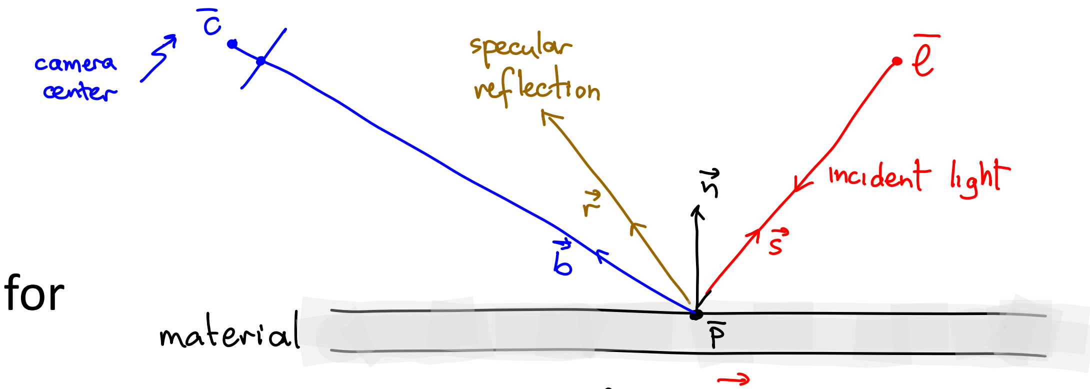


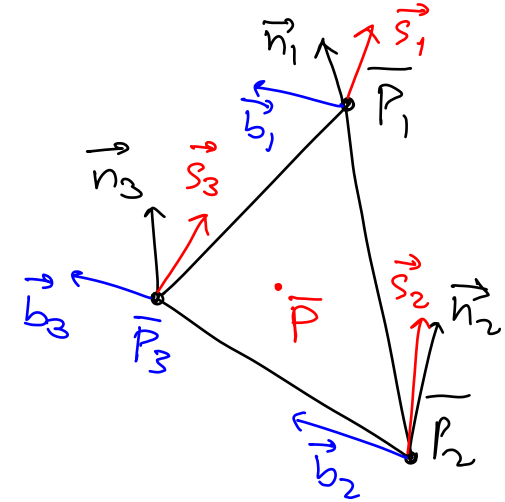
FLAT SHADING



PHONG SHADING

Interpolative Shading: Basic Approaches





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 $\bar{p}$

Phong shading

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

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

intensity at projection of point $\bar{p}$ — ambient — diffuse — specular

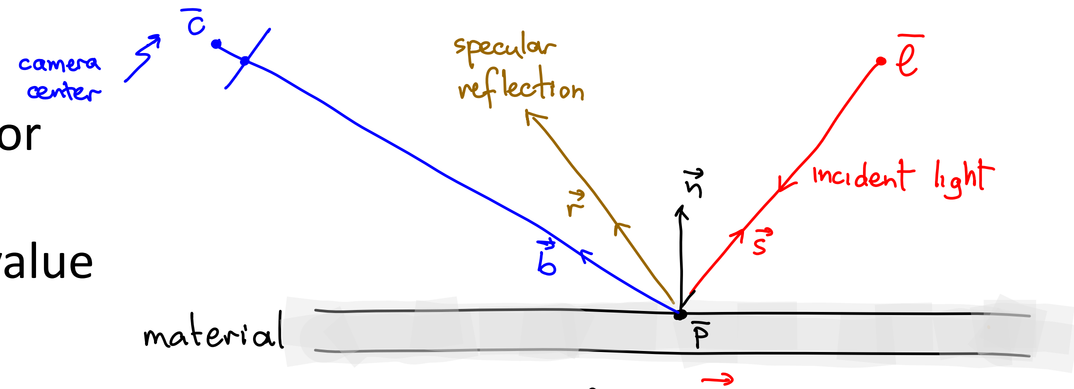
Gouraud Shading: Computation at Vertices

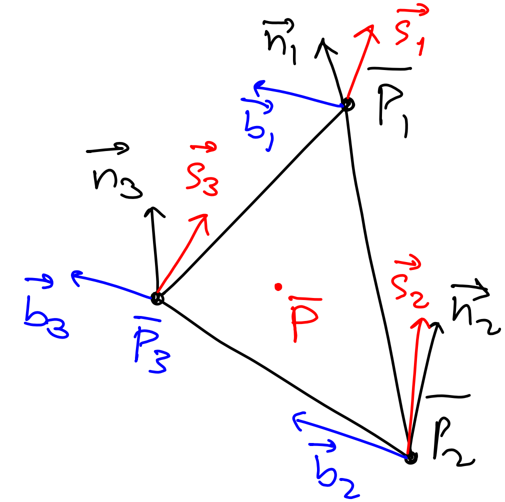
<u>Gouraud shading</u>

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 $\bar{p}$

Notes

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





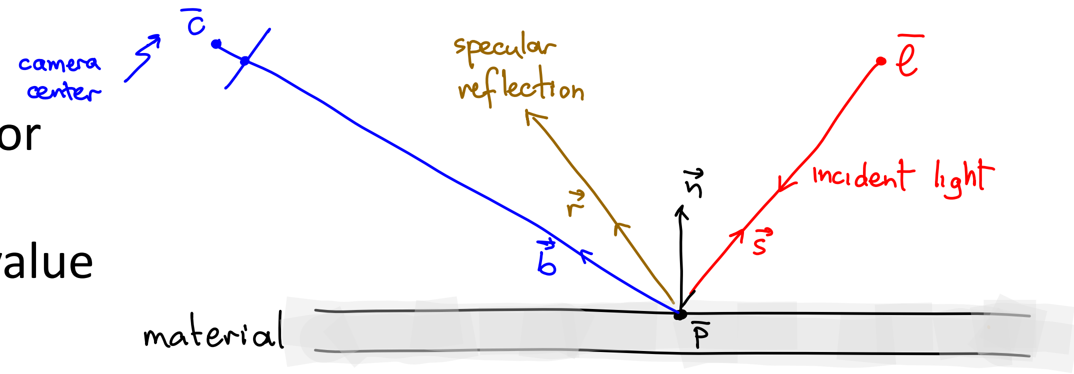
$$L(\vec{b}_i, \vec{n}_i, \vec{s}_i) = r_a I_a + r_d I_d \max(0, \vec{n}_i \cdot \vec{s}_i) + r_s I_s \max(0, \vec{r}_i \cdot \vec{b}_i)^{\alpha}$$

intensity at projection of point $\bar{p}$ — ambient — diffuse — specular

Gouraud Shading: Computation at Vertices

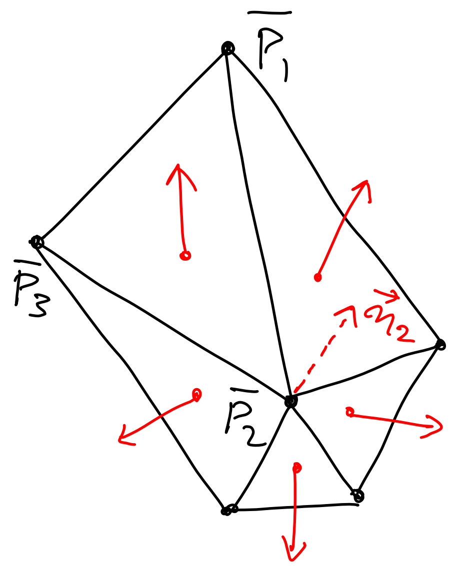
Gouraud shading

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 $\bar{p}$



Notes

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

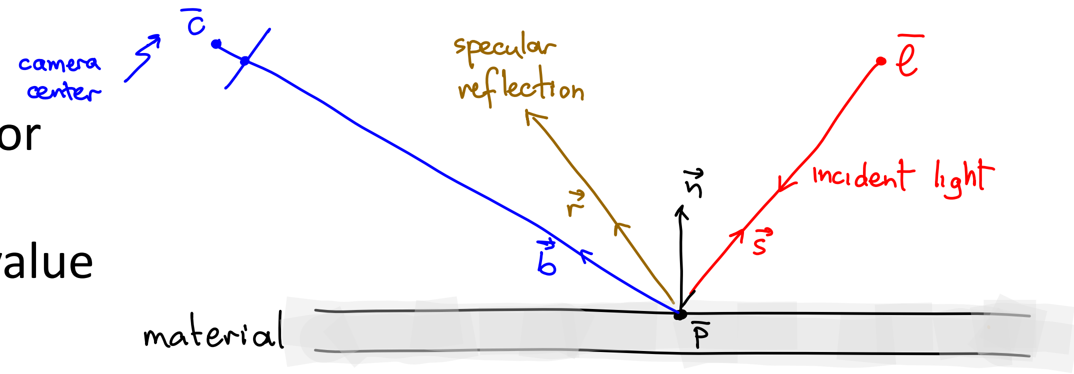


1. $\vec{n}_j$ is the average of the normals of all faces that contain vertex $\bar{p}_j$

Gouraud Shading: Computation at Vertices

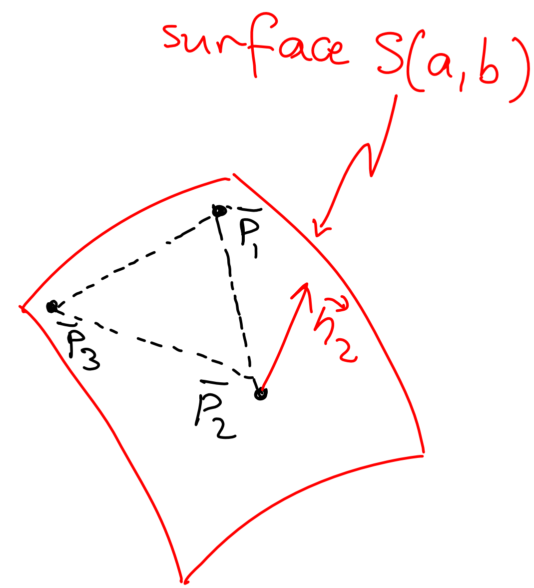
<u>Gouraud shading</u>

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 $\bar{p}$



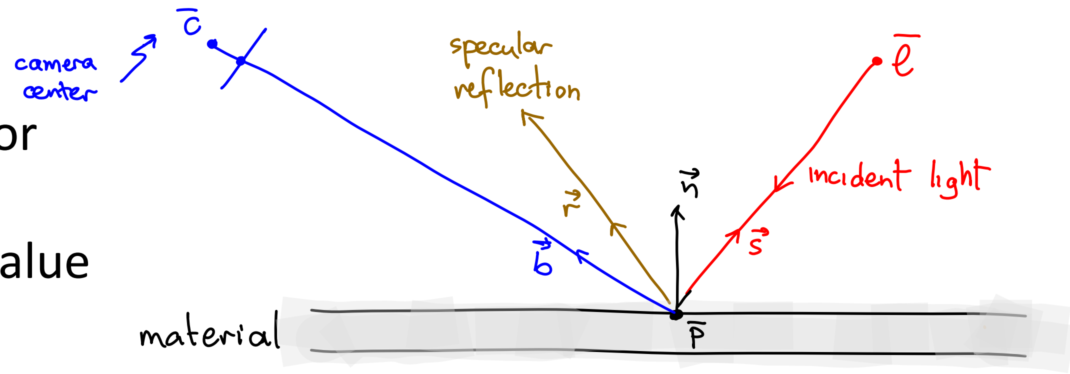
Notes

- Vectors $\vec{b}_i, \vec{s}_i$ computed directly from $\bar{p}_i, \bar{c}$ and $\bar{l}$
- 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

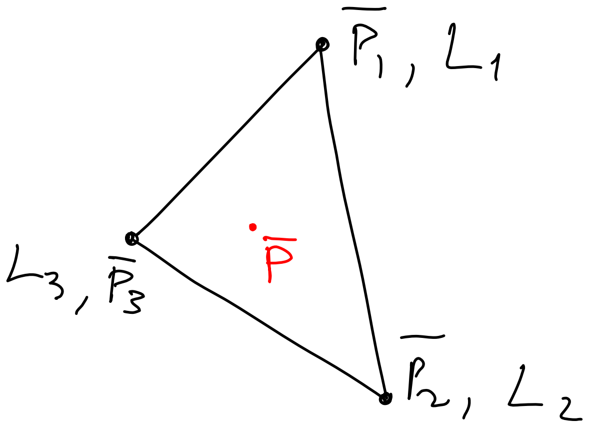
Gouraud Shading: Computation at Pixels



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 $\bar{p}$

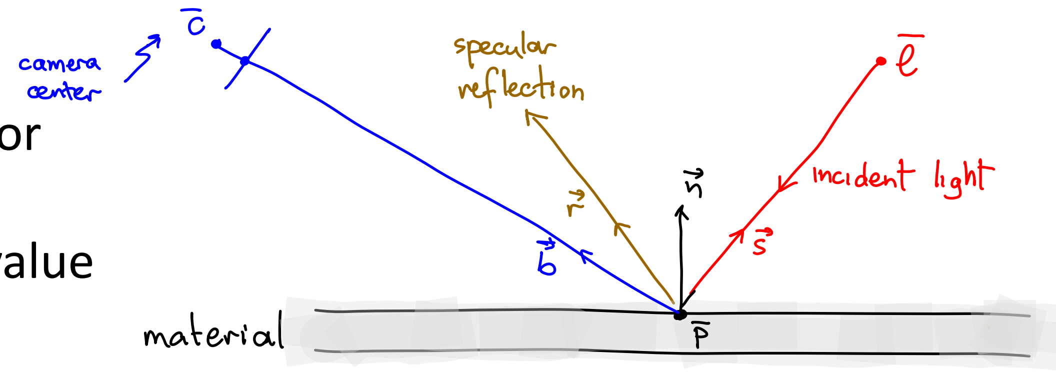
This step is integrated into the standard triangle-filling algorithm



Gouraud Shading: Computation at Pixels

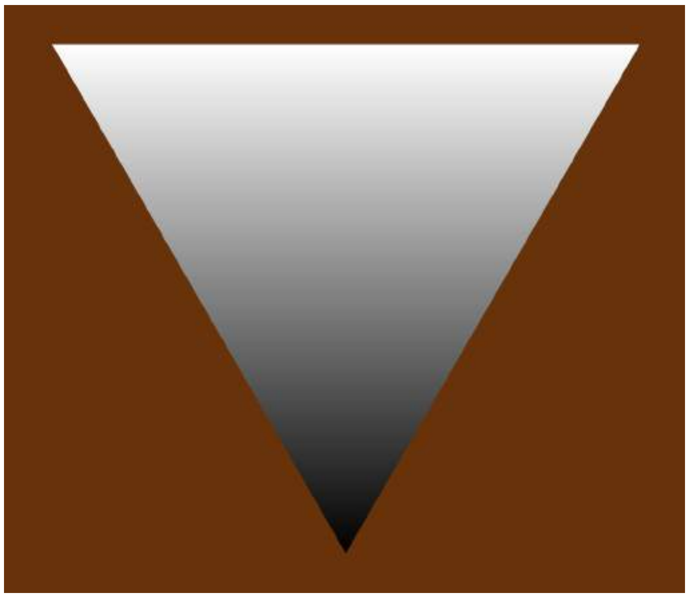
<u>Gouraud shading</u>

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 $\bar{p}$



This step is integrated into the standard triangle-filling algorithm

Gouraud-shaded triangle

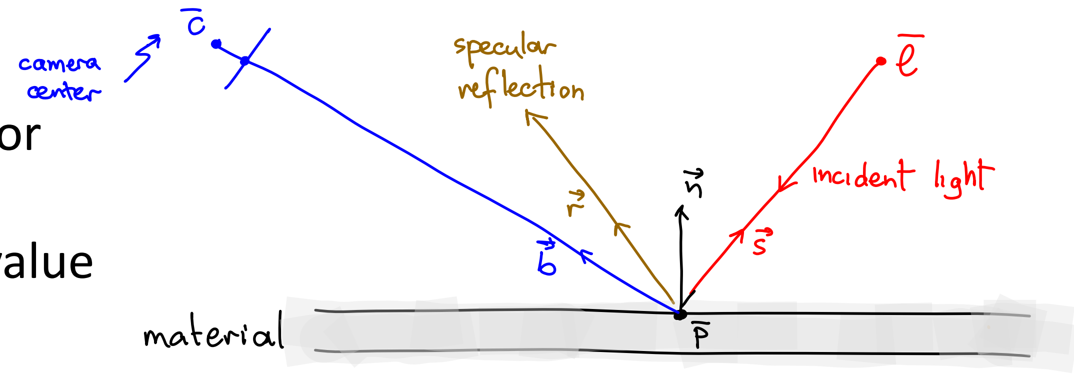


Yzmo, Wikipedia

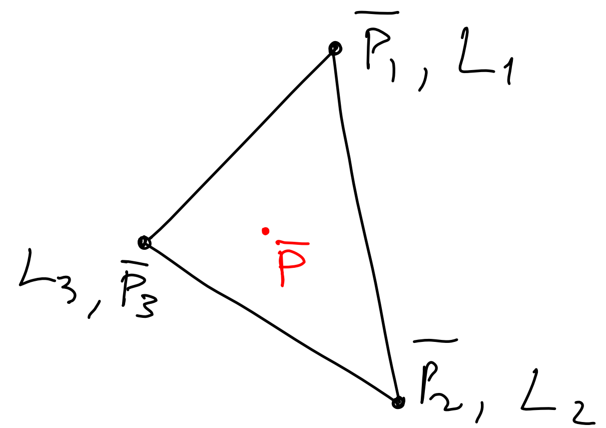
Gouraud Shading: Comparisons

<u>Gouraud shading</u>

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 $\bar{p}$



<u>Comparison to flat shading</u>

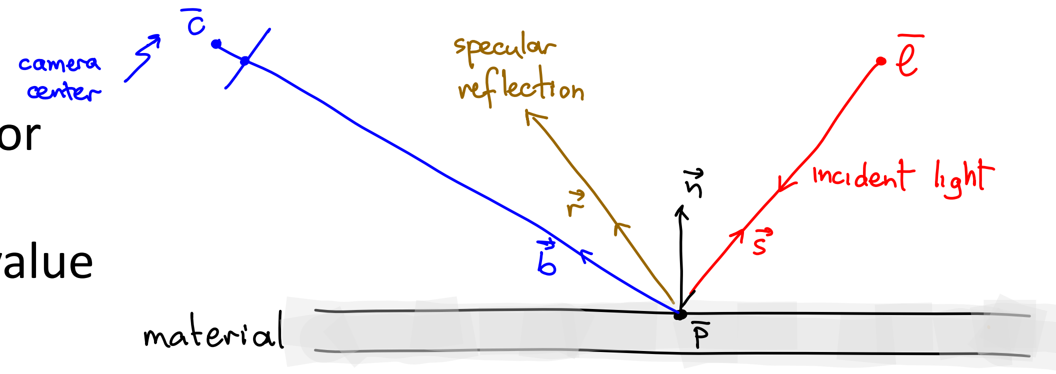


\+ No visible seams between mesh triangles
\+ Smooth, visually pleasing intensity variation that "mask" coarse geometry
\- Specular highlights still a problem for large triangles (why?)

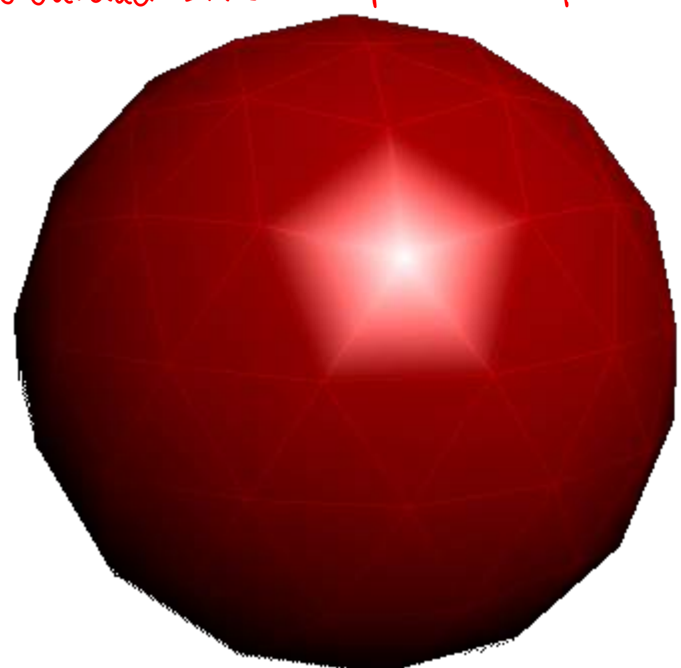
Gouraud Shading: Comparisons

<u>Gouraud shading</u>

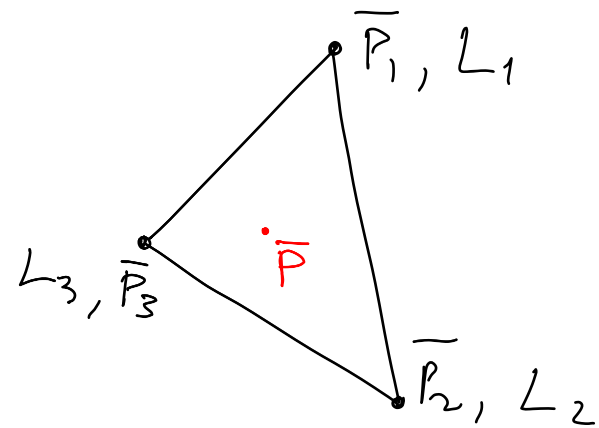
1. Compute $L_i = L(\vec{b}_i, \vec{n}_i, \vec{s}_i)$ for each vertex
2. Interpolate the L_i's to get the value at $\bar{p}$



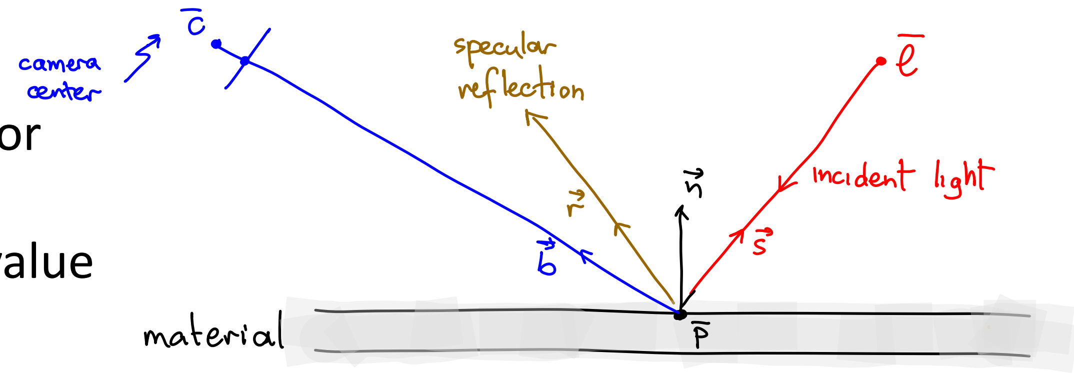
Gouraud-shaded specular sphere



Jalo, Wikipedia



Gouraud Shading: Comparisons



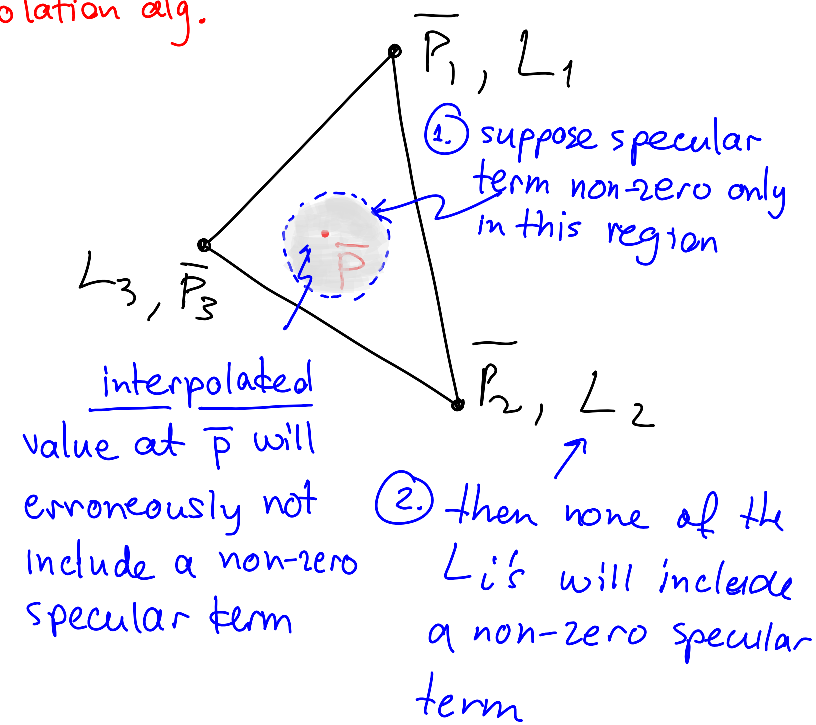
__Gouraud shading__

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 $\bar{p}$

$$L = \beta L_1 + \gamma L_2 + \epsilon L_3$$

constants determined by interpolation alg.

__Comparison to flat shading__



\+ No visible seams between mesh triangles

\+ Smooth, visually pleasing intensity variation that "mask" coarse geometry

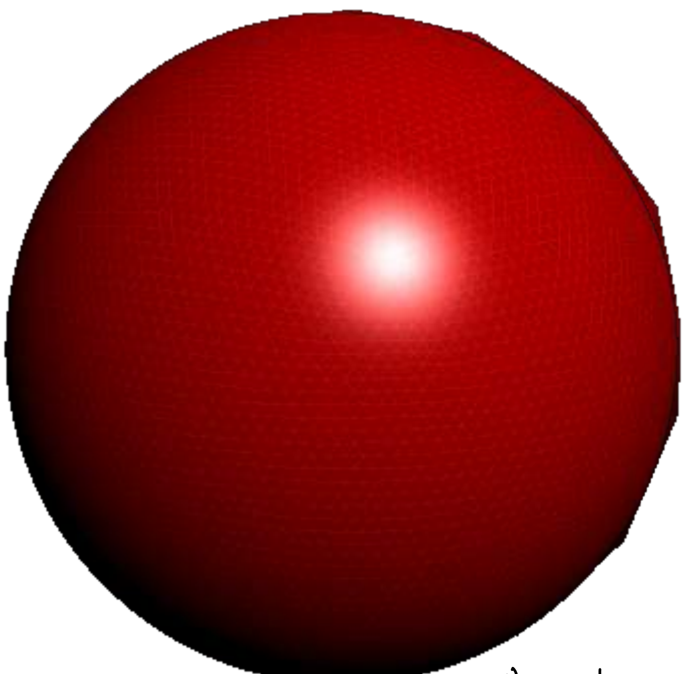
\- Specular highlights still a problem for large triangles (why?)

Gouraud Shading: Comparisons

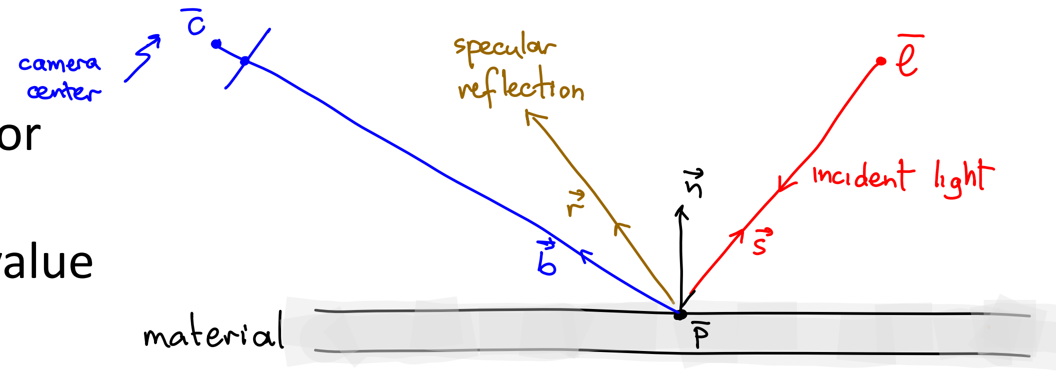
<u>Gouraud shading</u>

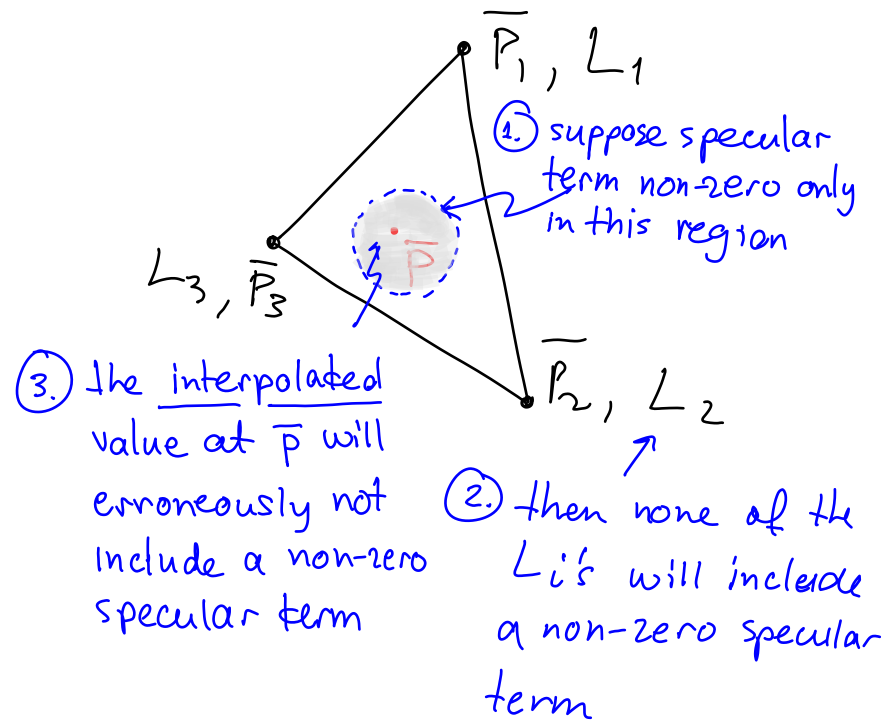
1. Compute $L_i = L(\vec{b}_i, \vec{n}_i, \vec{s}_i)$ for each vertex
2. Interpolate the L_i's to get the value at $\bar{p}$

$$L = \beta L_1 + \gamma L_2 + \epsilon L_3$$



Jalo, Wikipedia





Topic 10:

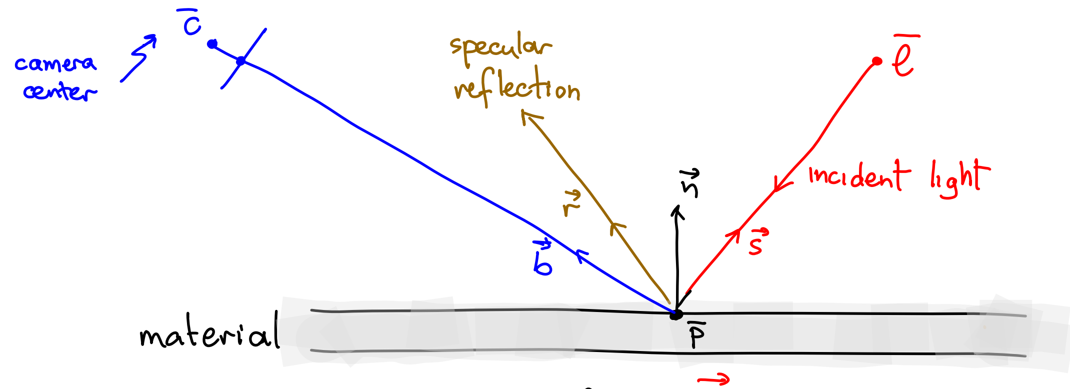
Shading

- Introduction to Shading
- Flat Shading
- Interpolative Shading
  - Gouraud shading
  - Phong shading

Phong Shading: Main Idea

<u>Phong shading:</u>

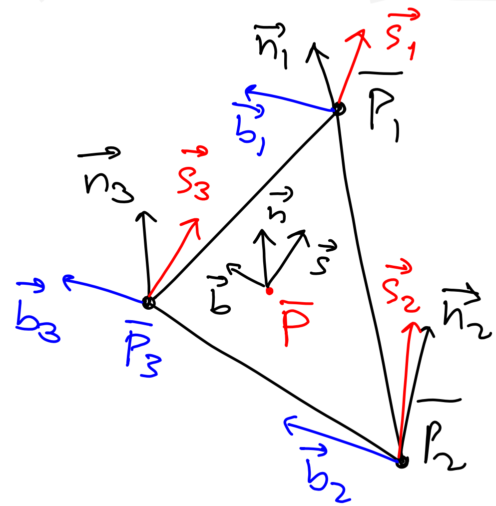
1. Interpolate $\vec{b}_i, \vec{n}_i, \vec{s}_i$ to get $\vec{b}, \vec{n}, \vec{s}$ at $\bar{p}$
2. Compute $L(\vec{b}, \vec{n}, \vec{s})$



<u>Comparison to Gouraud shading</u>

\+ Smooth intensity variations as in Gouraud shading

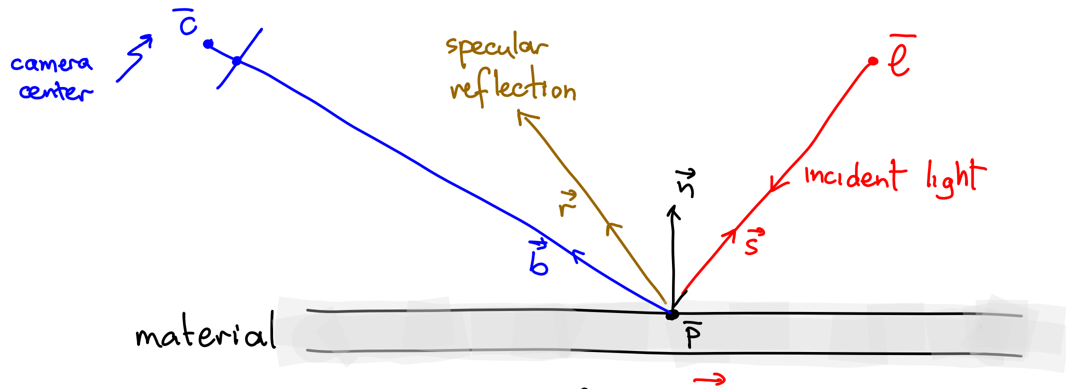
\+ Handles specular highlights correctly even for large triangles (Why?)



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

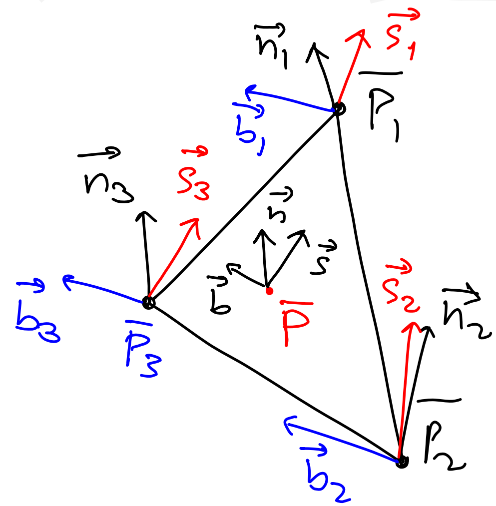
$L(\vec{b}, \vec{n}, \vec{s})$: intensity at projection of point $\bar{p}$; $r_a I_a$: ambient; $r_d I_d \max(0, \vec{n} \cdot \vec{s})$: diffuse; $r_s I_s \max(0, \vec{r} \cdot \vec{b})^{\alpha}$: specular

Phong Shading: Comparisons



Phong shading:

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



Comparison to Gouraud shading

+ Smooth intensity variations as in Gouraud shading

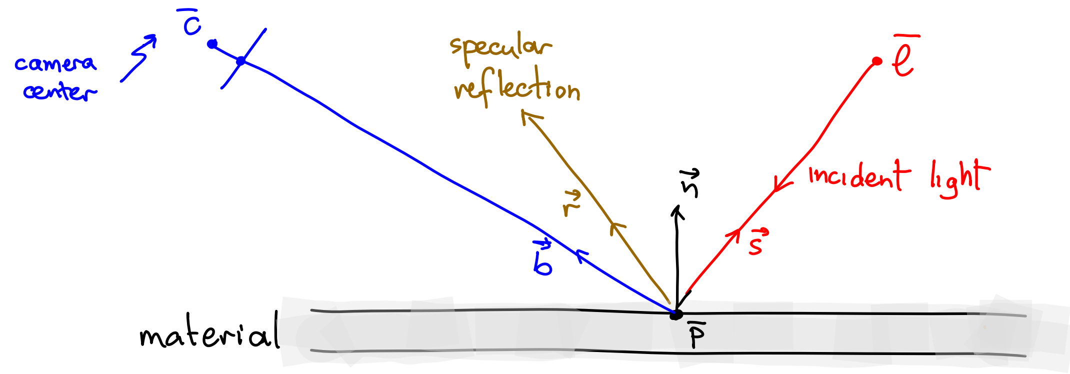
+ Handles specular highlights correctly even for large triangles (Why?)

it is possible to have a significant specular component at $\bar{p}$ even when all vertices have a negligible specular component

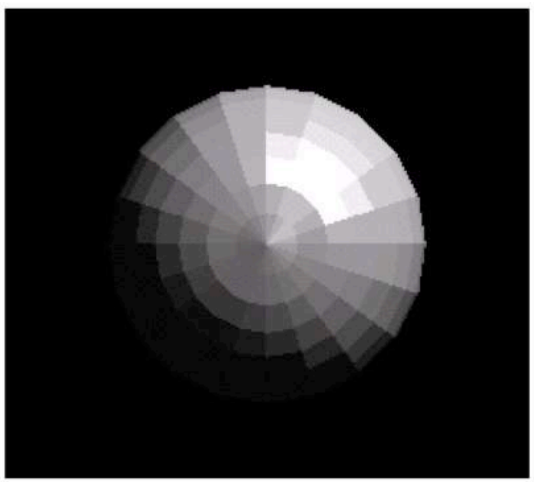
Phong Shading: Comparisons

<u>Phong shading:</u>

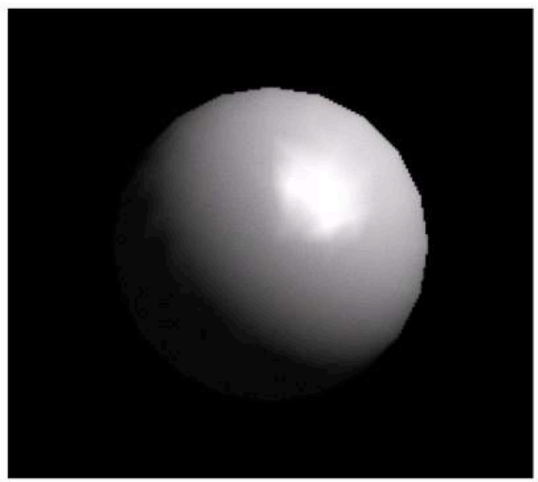
1. Interpolate $\vec{b}_i, \vec{n}_i, \vec{s}_i$ to get $\vec{b}, \vec{n}, \vec{s}$ at $\bar{p}$
2. Compute $L(\vec{b}, \vec{n}, \vec{s})$



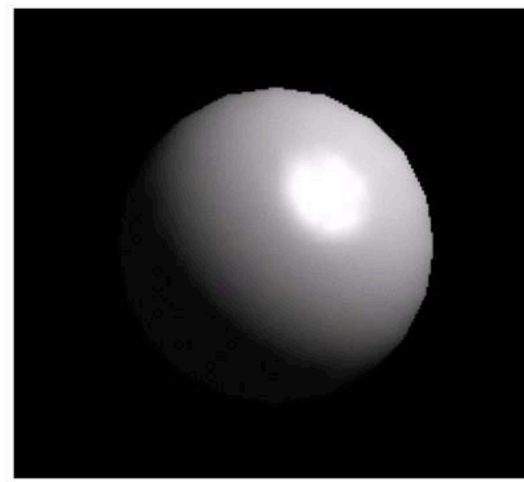
Hsien-Hsin Sean Lee, GaTech



Flat shading



Gouraud shading

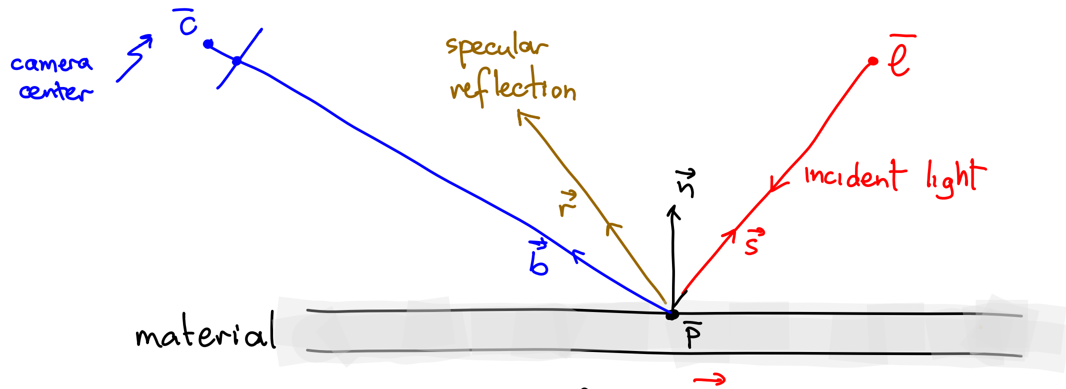


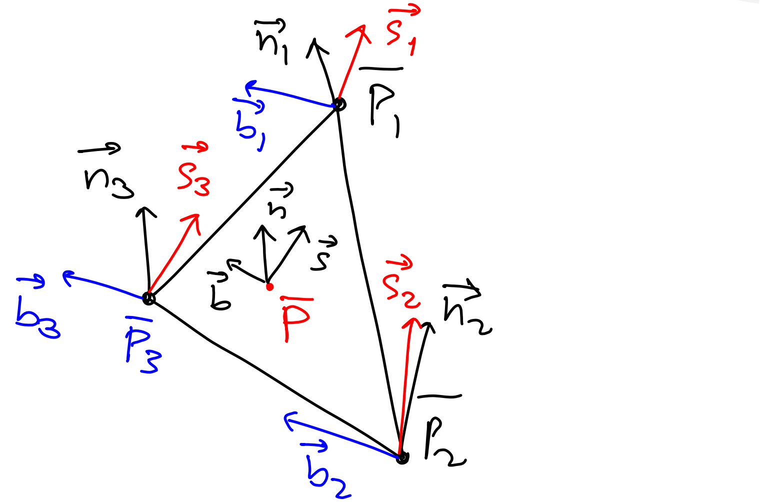
Phong shading

Phong Shading: Comparisons

Phong shading:

1. Interpolate $\vec{b}_i, \vec{n}_i, \vec{s}_i$ to get $\vec{b}, \vec{n}, \vec{s}$ at $\bar{p}$
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)