Today's Topics

9. Lighting & Reflection models
10. Shading

Topic 9:

Lighting & Reflection models

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

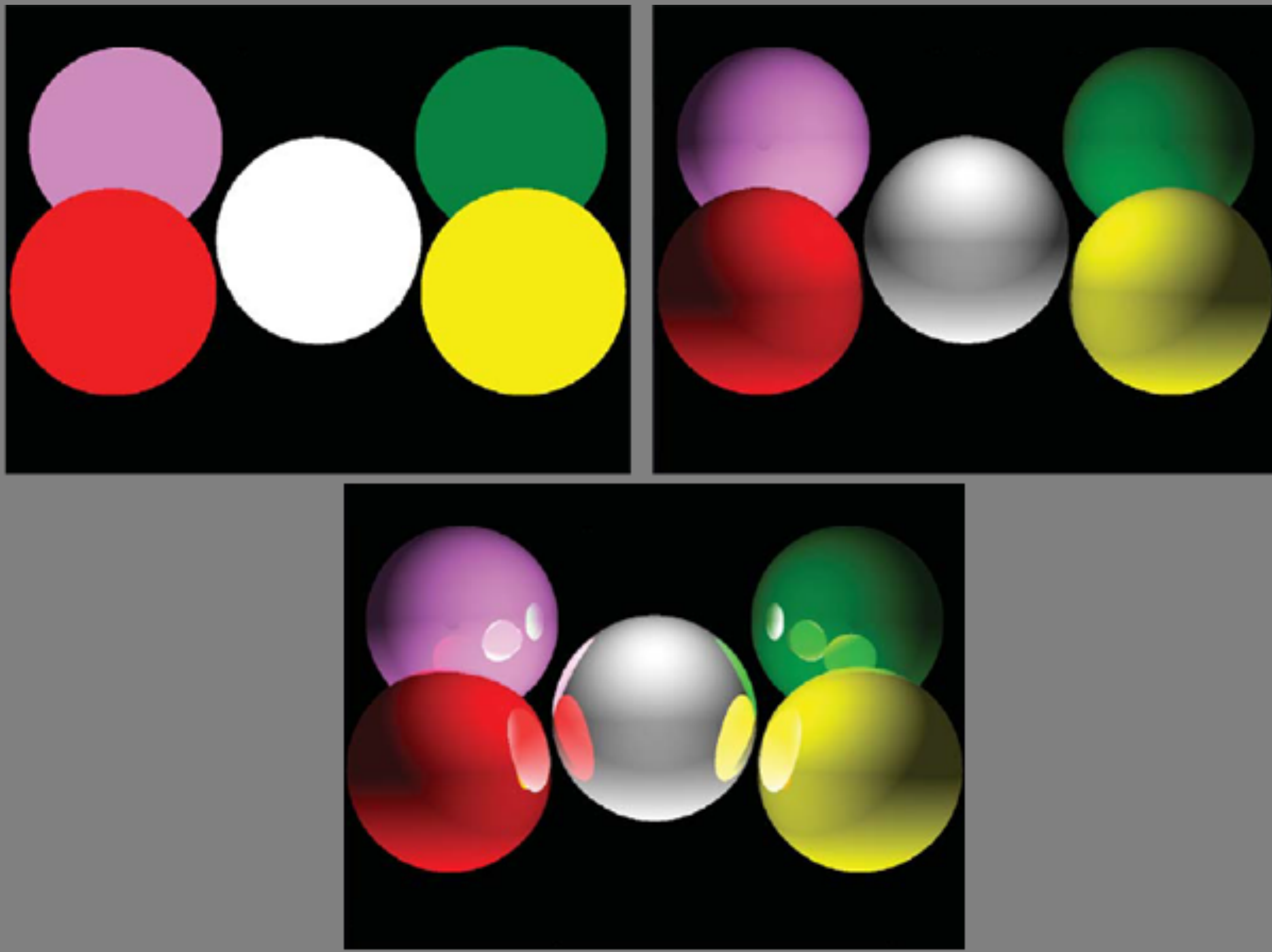
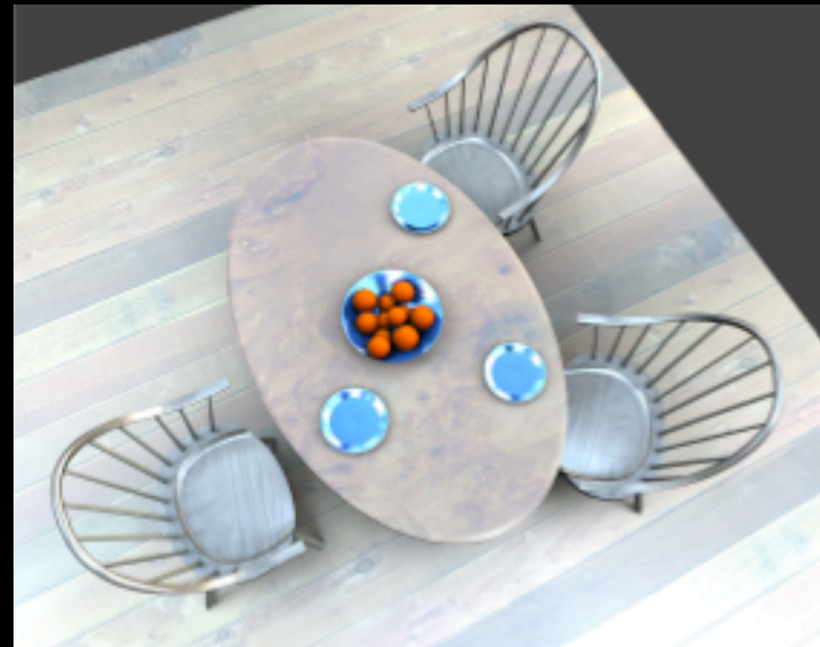


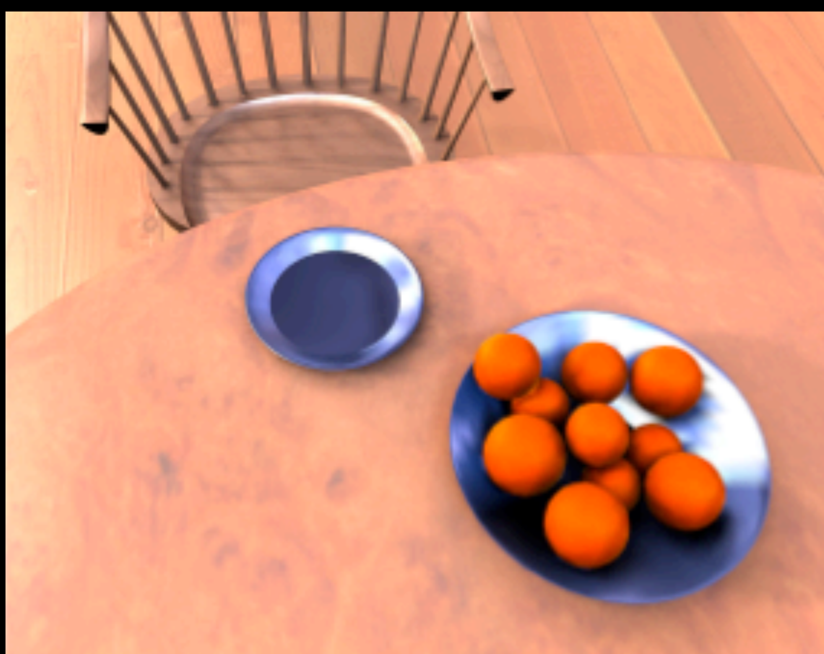
Image credit: http://www.doc.gold.ac.uk/~ma901na/Portfolio.html

Ng et al, SIGGRAPH'04

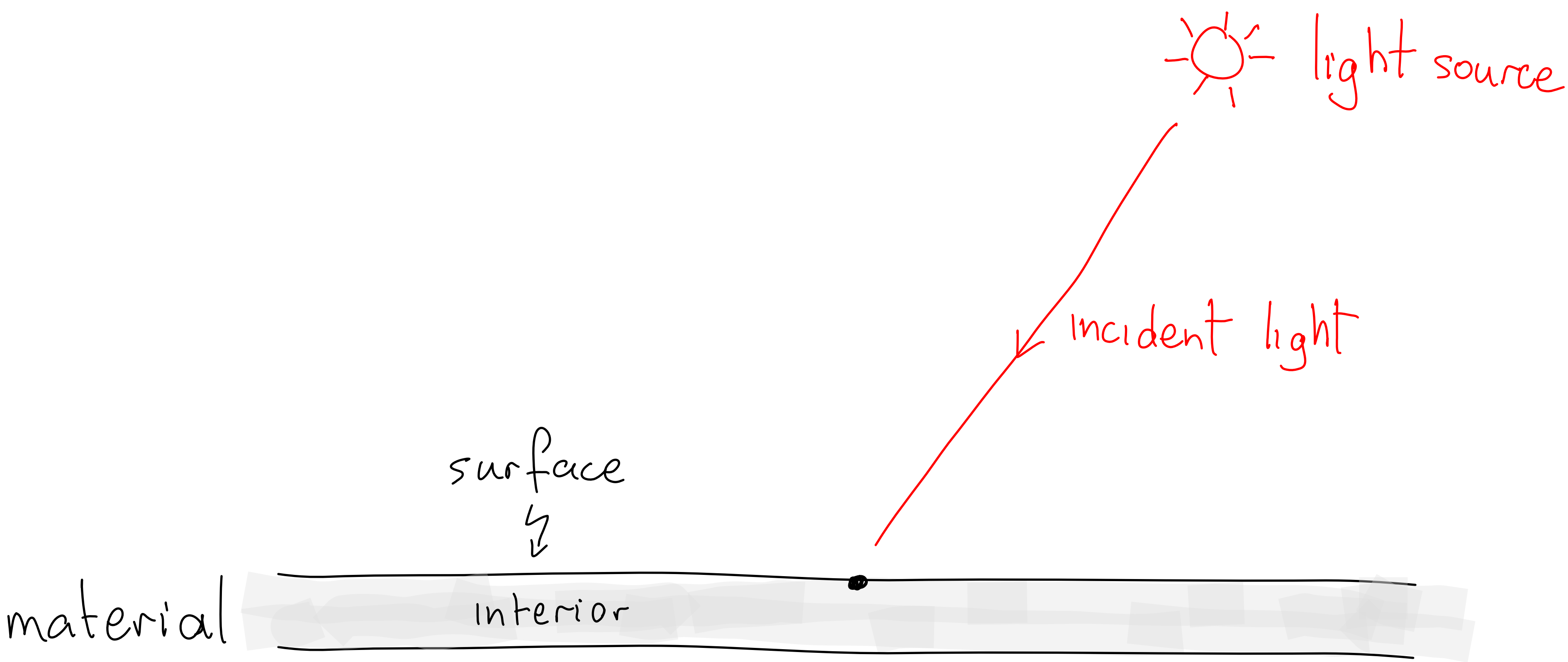
Ng et al, SIGGRAPH'04

Ng et al, SIGGRAPH'04

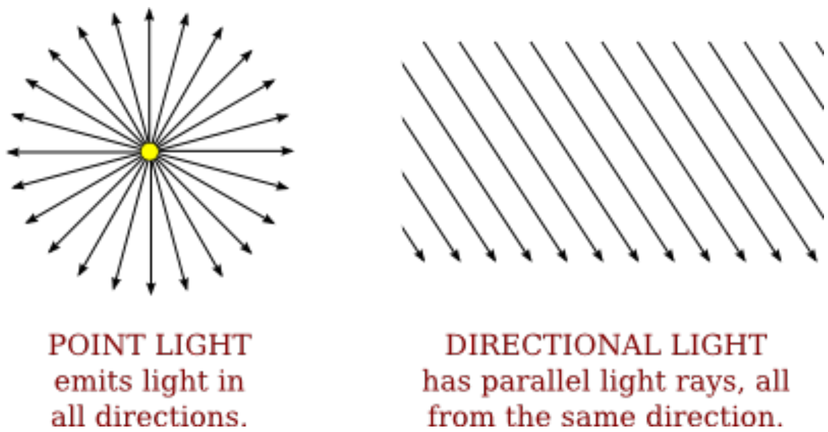




Light Sources



Light Sources



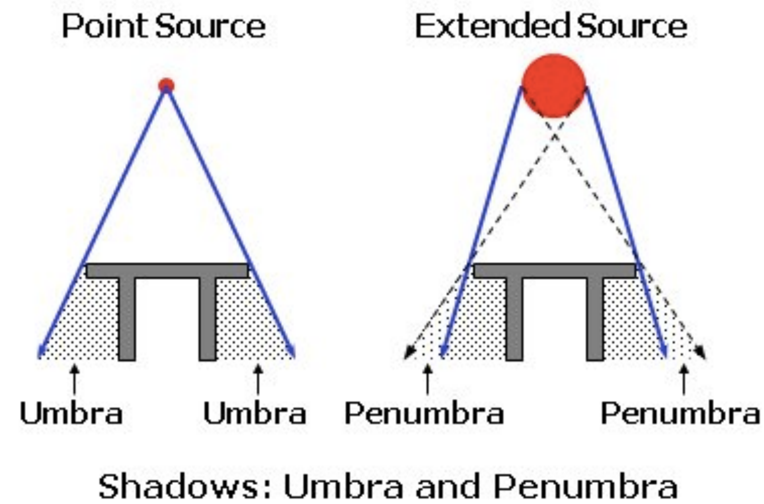
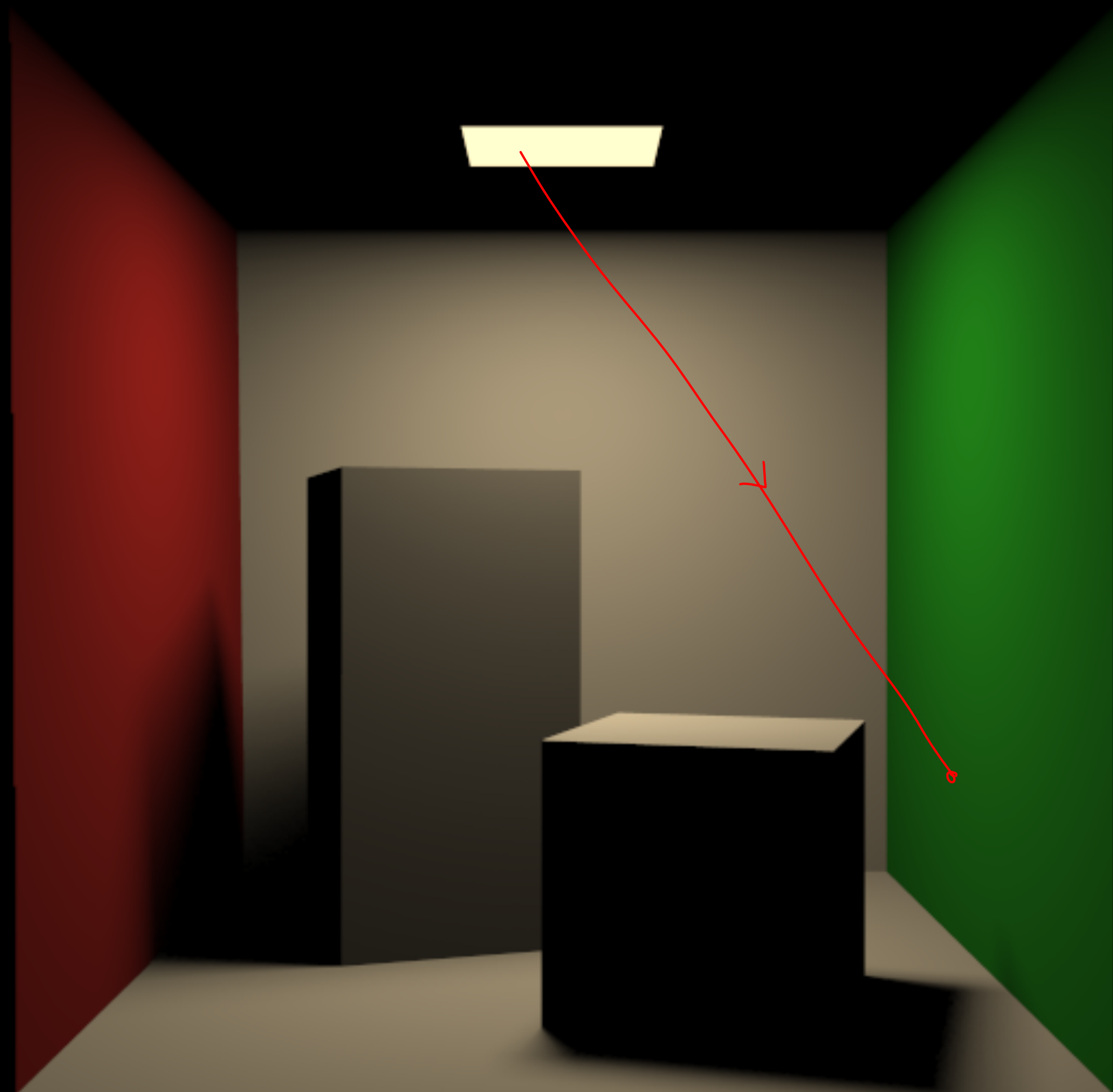
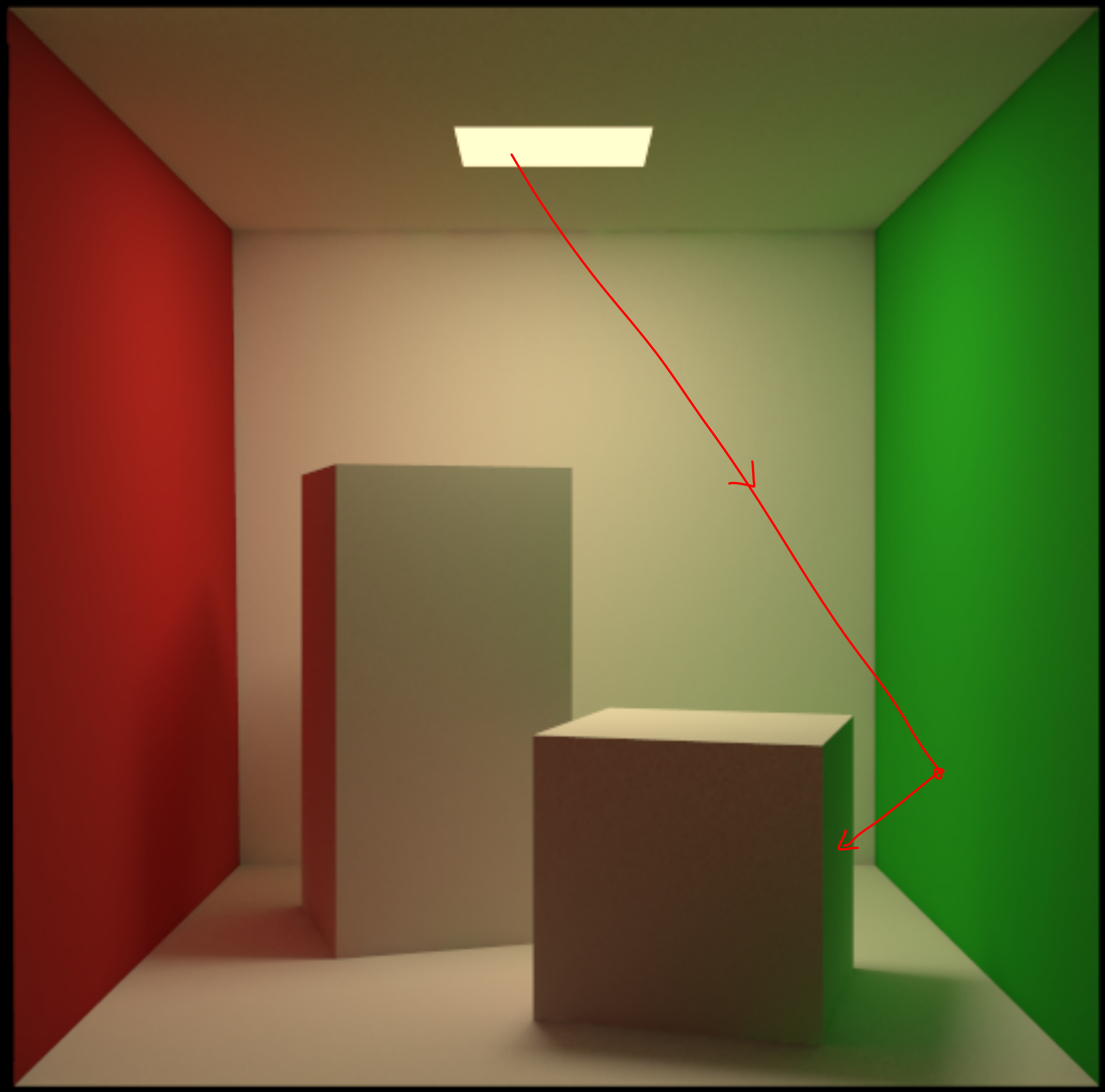


Image credit: Google images search

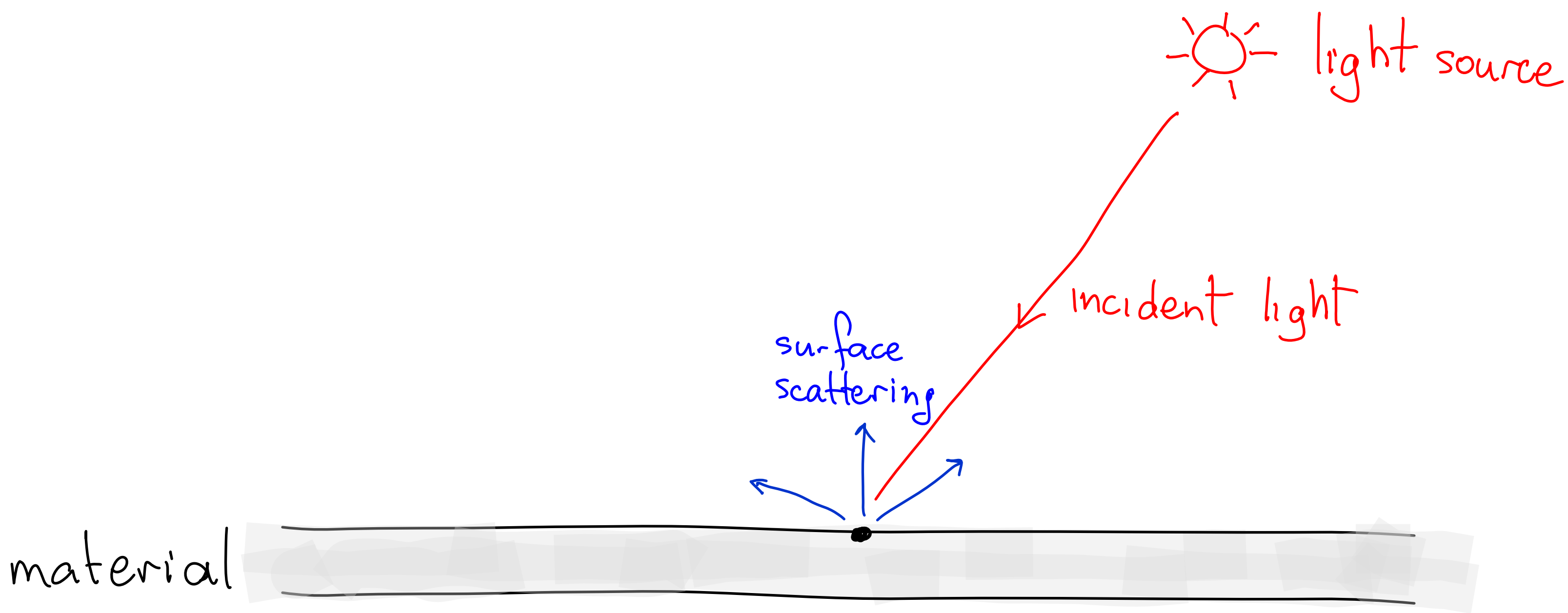
Area Light Source, Direct Lighting



Area Light Source, Indirect Lighting



Modeling Reflection: Diffuse Reflection

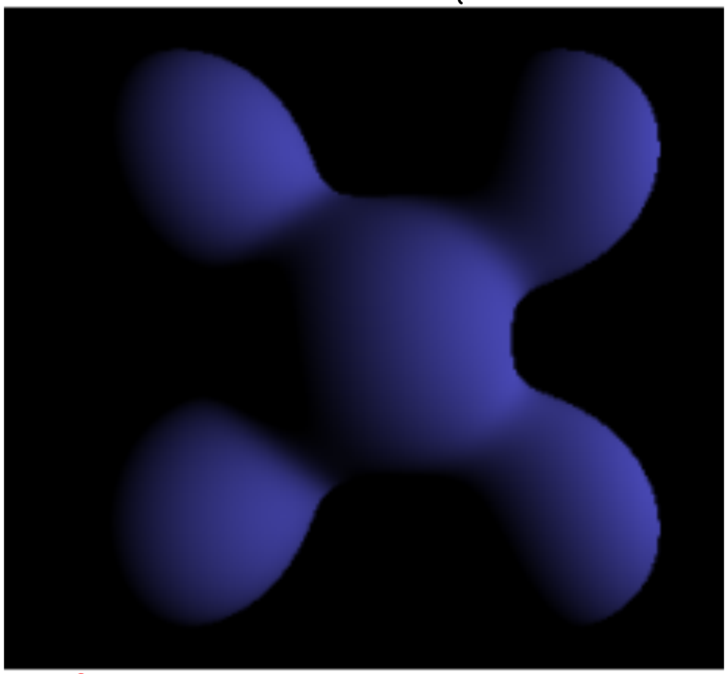


Diffuse reflection:

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

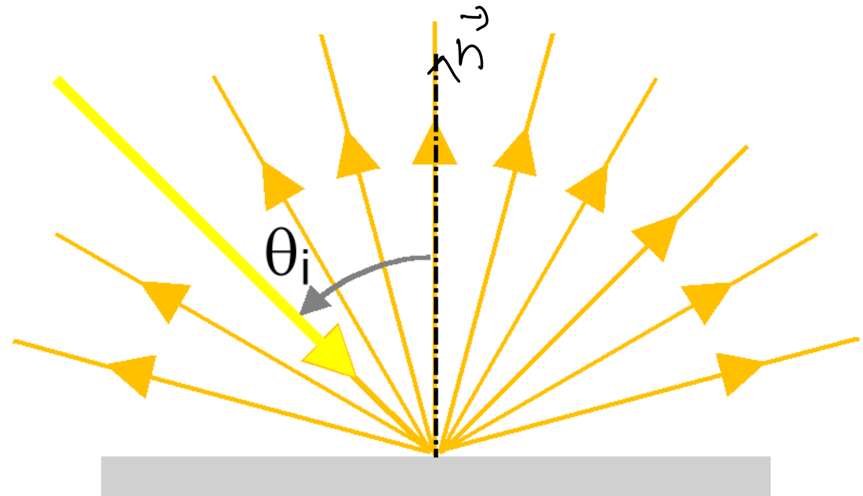
Modeling Reflection: Diffuse Reflection

Brad Smith, Wikipedia



Diffusely-shaded object

θ_i = angle of incidence

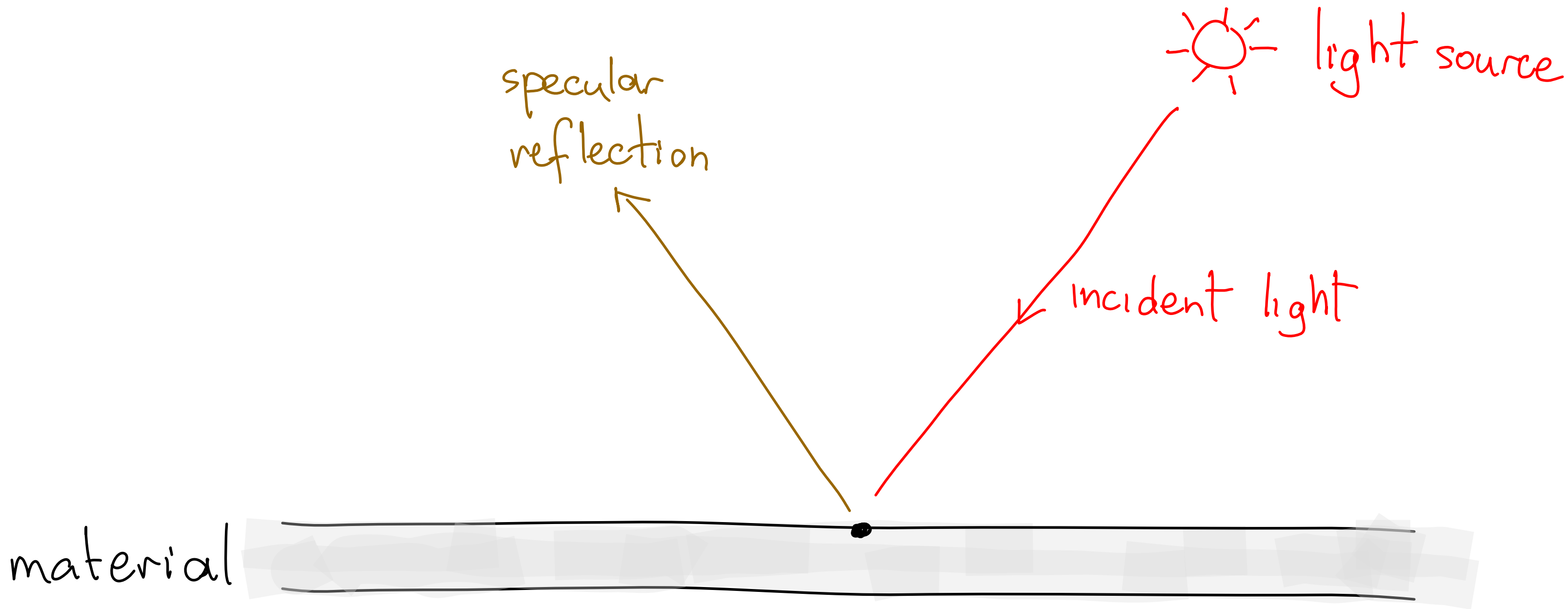


Panjasan, Wikipedia

Diffuse reflection:

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

Modeling Reflection: Specular Reflection



Specular reflection:

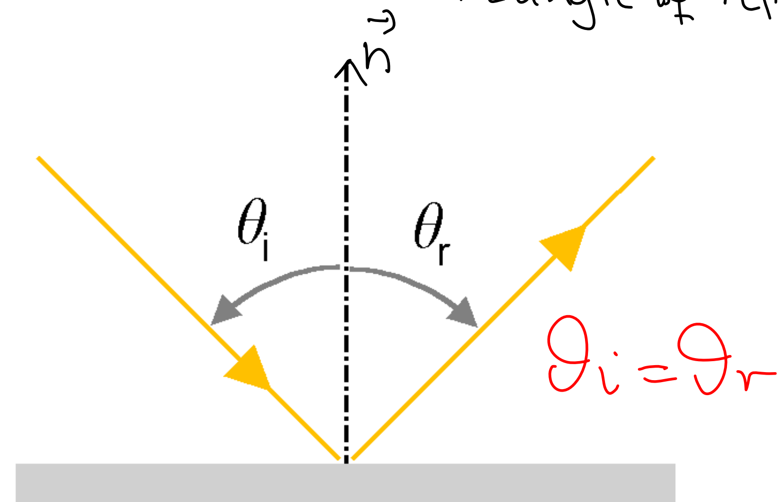
- Represents shiny component of reflected light
- Caused by mirror like reflection off of smooth or polished surfaces (plastics, polished metals, etc)

Modeling Reflection: Specular Reflection

Romeiro et al, ECCV'08

mirror-like sphere

ϑ_i = angle of incidence
ϑ_r = angle of reflection



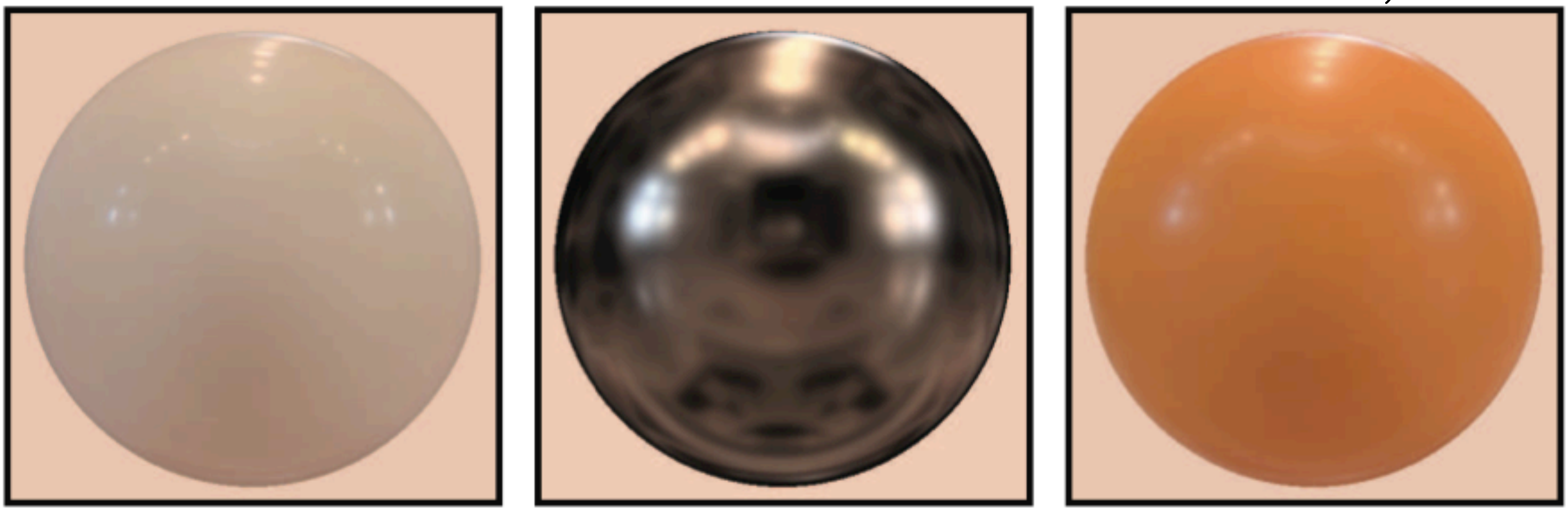
Panjasan, Wikipedia

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

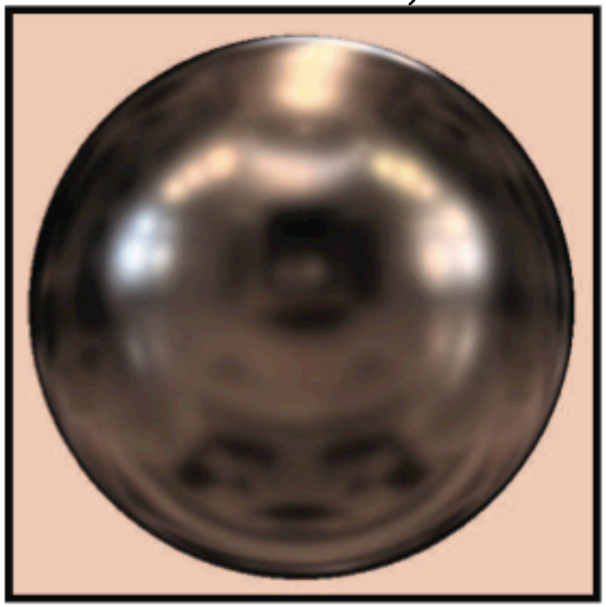


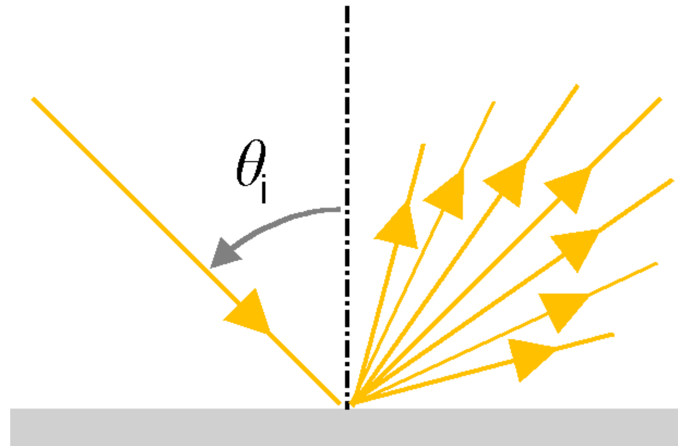
Specular reflection:

- 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





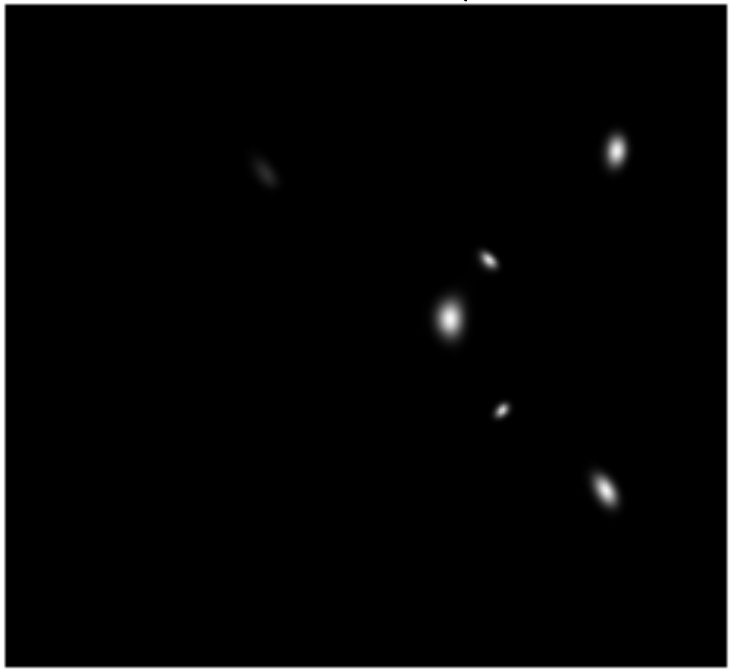
Panjasan, Wikipedia

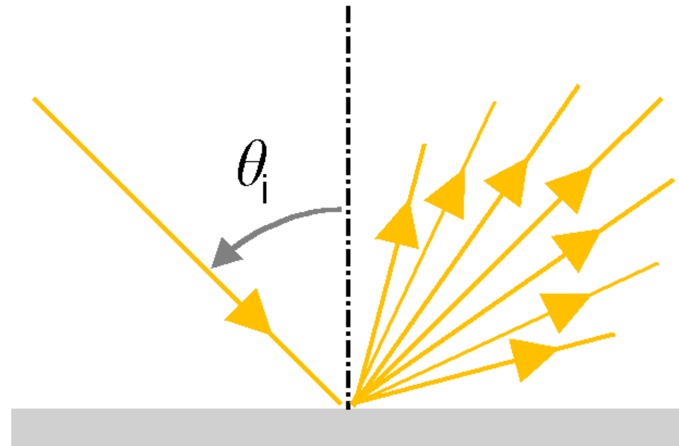
Specular reflection:

- 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



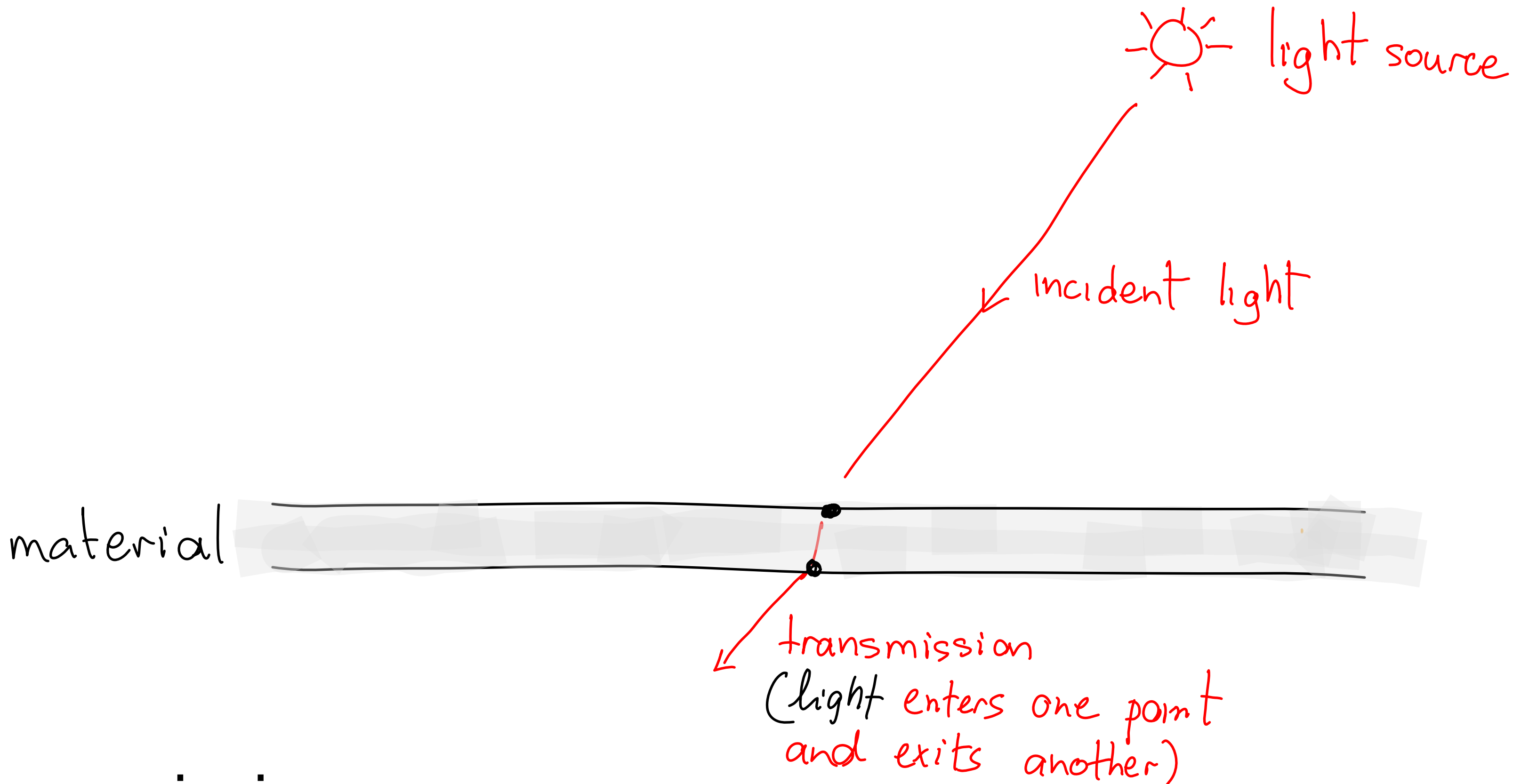


Panjasan, Wikipedia

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

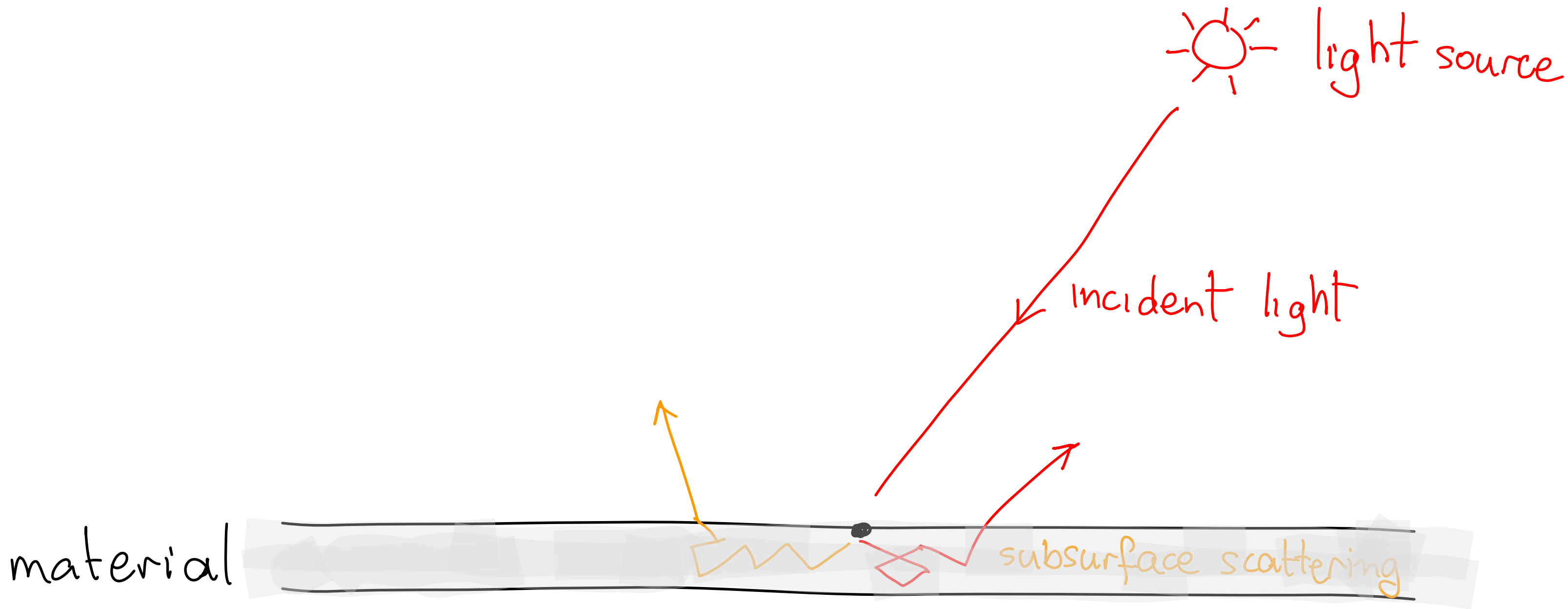


Transmission:

- 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

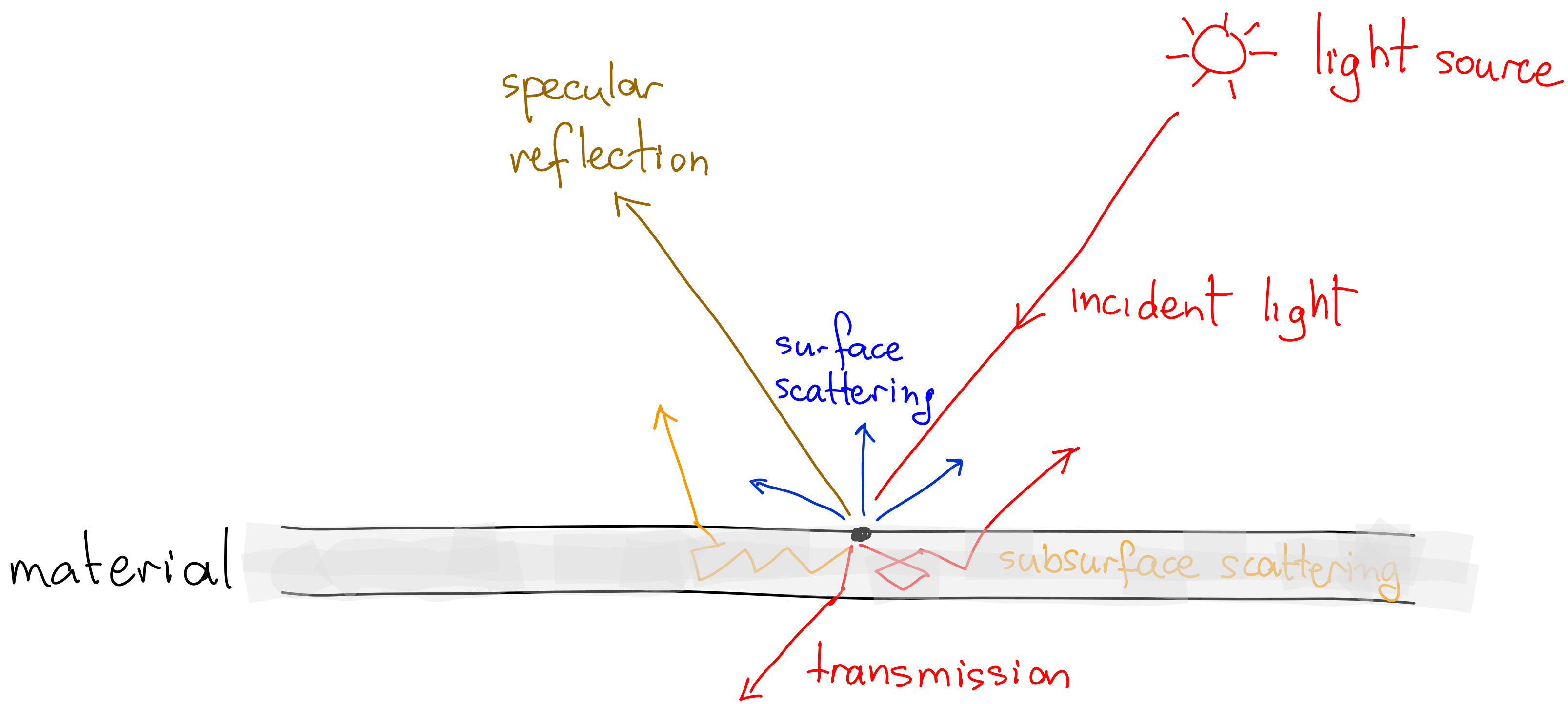
Rendering with subsurface scattering (full milk)

Jensen et al, SIGGRAPH'01

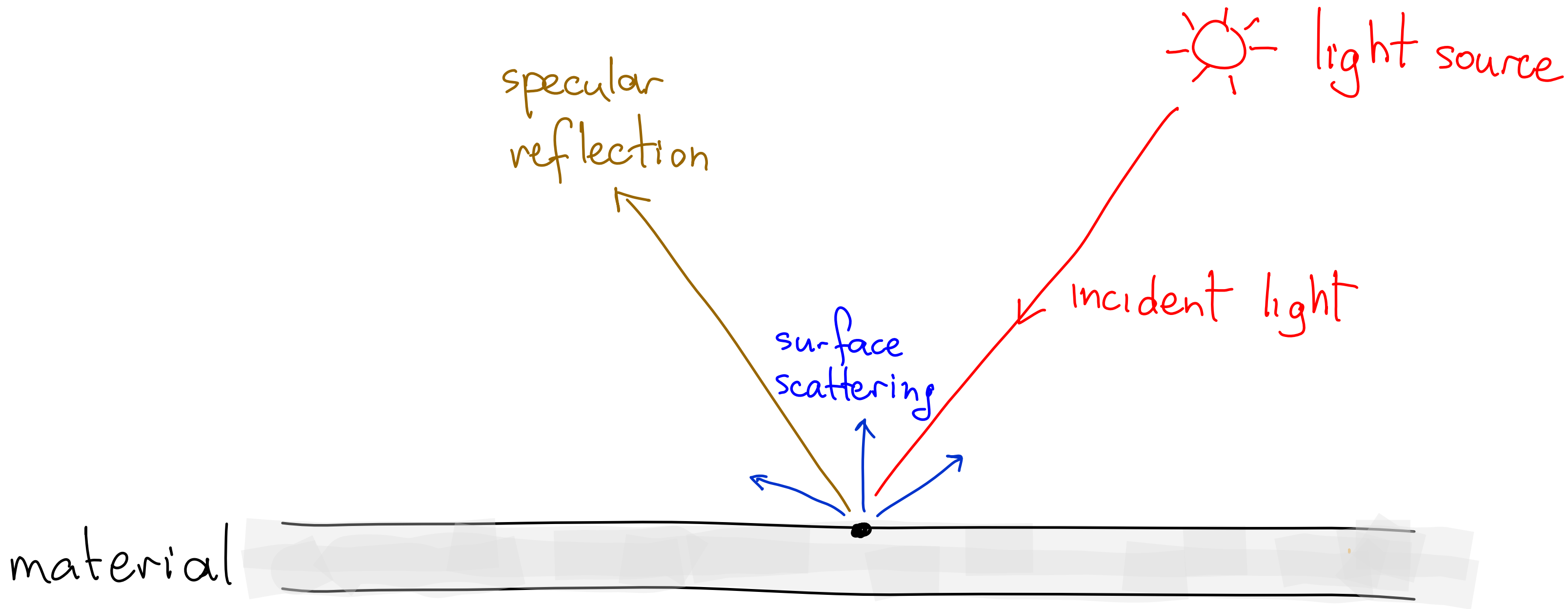
Rendering with subsurface scattering (skim milk)

Jensen et al, SIGGRAPH'01

The Common Modes of “Light Transport”



The Phong Reflectance Model

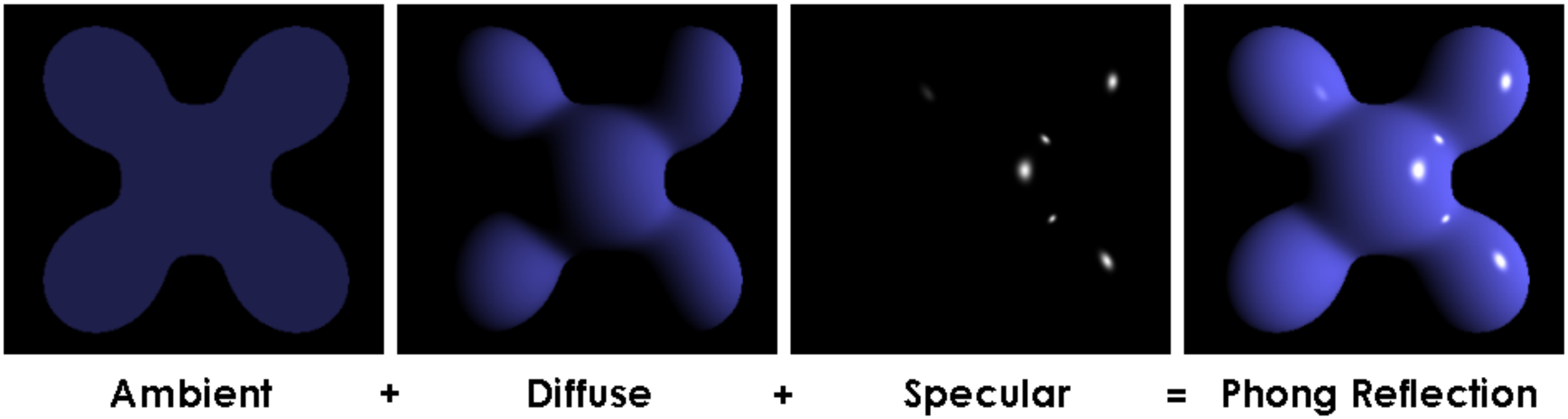


Phong model: A simple computationally efficient model that has 3 components:

- Diffuse
- Ambient
- Specular

The Phong Reflectance Model

Brad Smith, Wikipedia



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

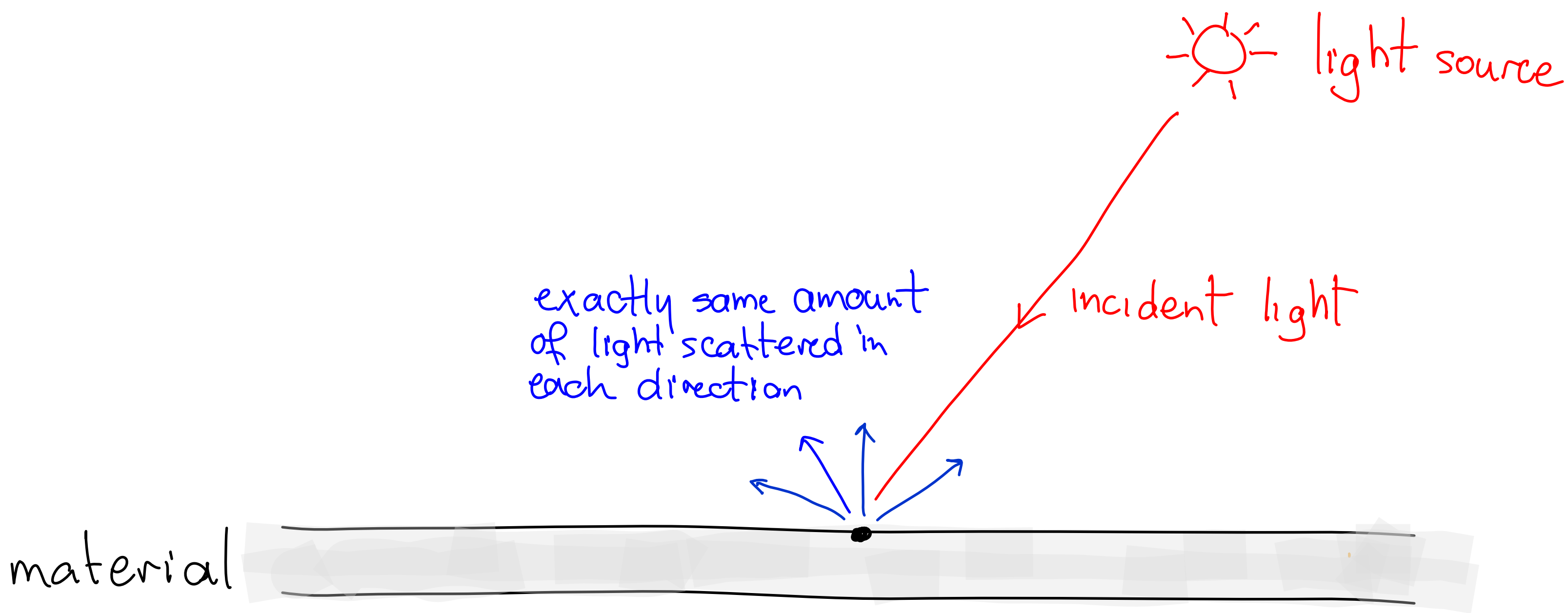
- Diffuse
- Ambient
- Specular

Topic 9:

Lighting & Reflection models

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

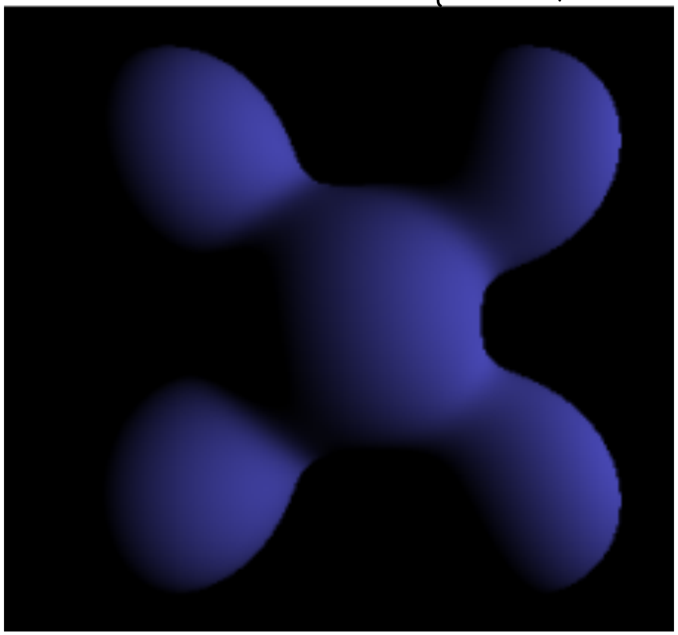
Phong Reflection: The Diffuse Component



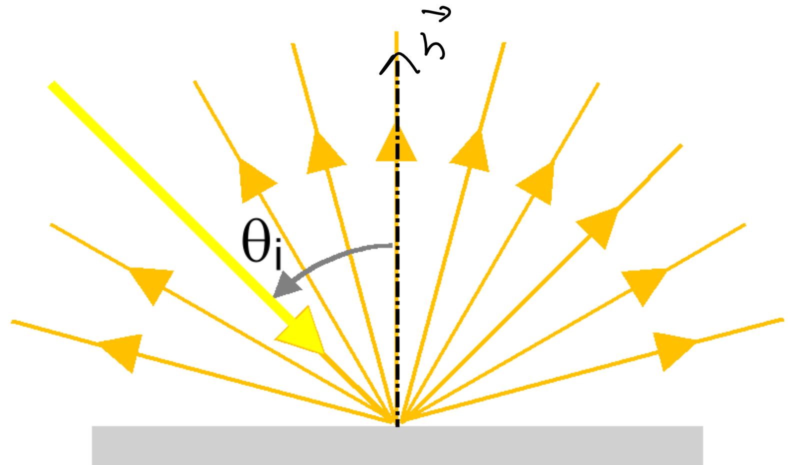
- A diffuse point looks the same from all viewing positions
- Simplest case: a single, point light source

Phong Reflection: The Diffuse Component

Brad Smith, Wikipedia



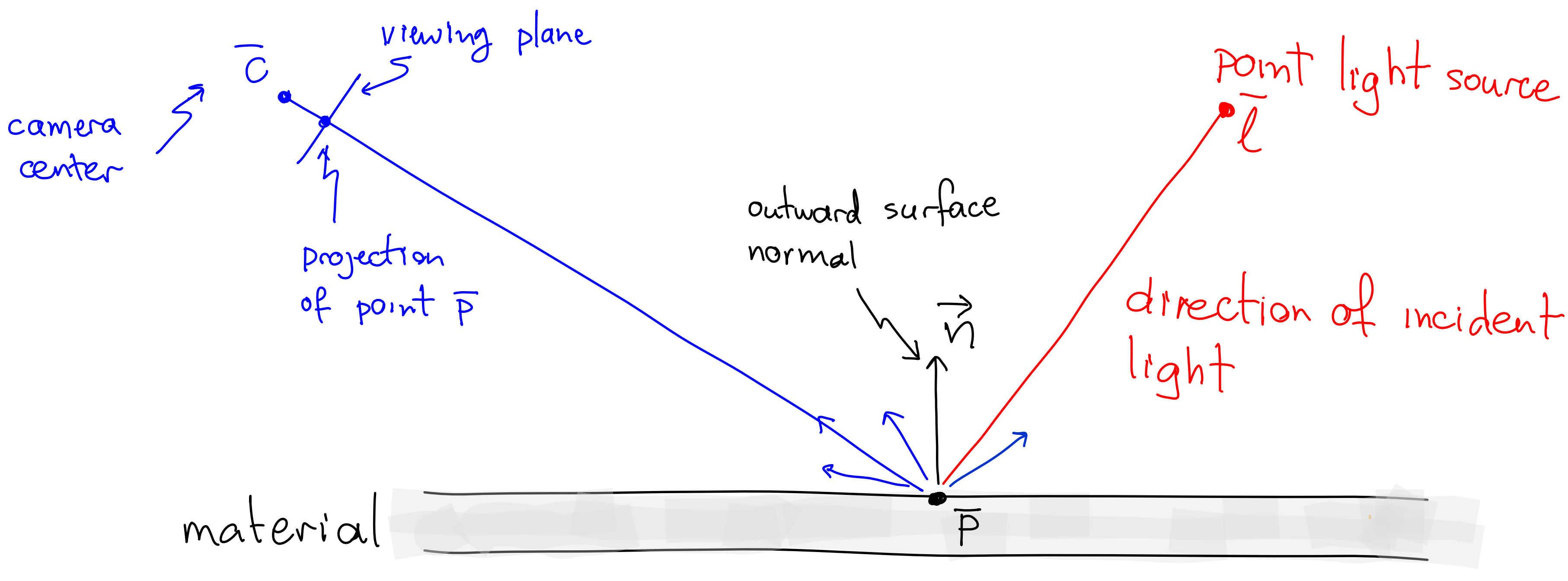
θ_i = angle of incidence



Panjasan, Wikipedia

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

The Diffuse Component: Basic Equation



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

fraction of light reflected

outward unit surface normal

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

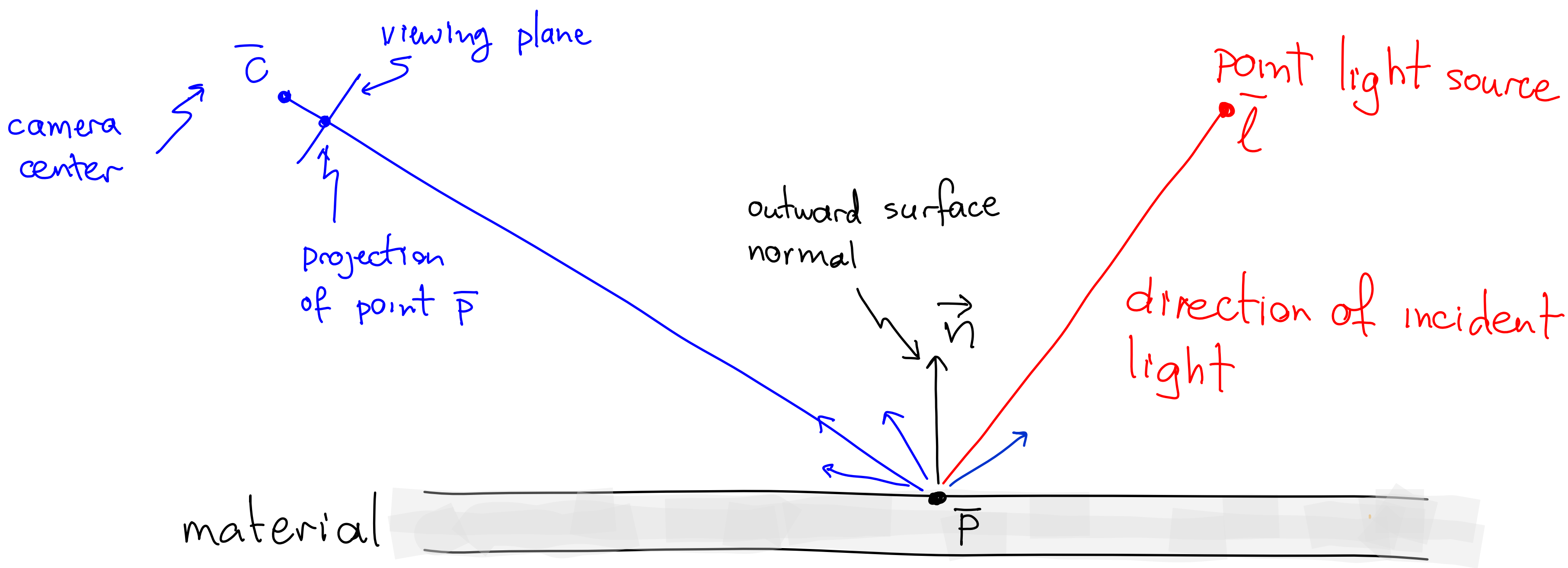
intensity at projection of $\bar{p}$

intensity of source

direction of light source

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

The Diffuse Component: Basic Equation



- A diffuse point looks the same from all viewing positions

outward unit surface normal

$$I_{\bar{p}} = r_d \cdot \overbrace{I \cdot \max\left(0, \vec{s} \cdot \vec{n}\right)}^{\text{independent of } \bar{c}}$$

intensity at projection of $\bar{p}$

direction of light source

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

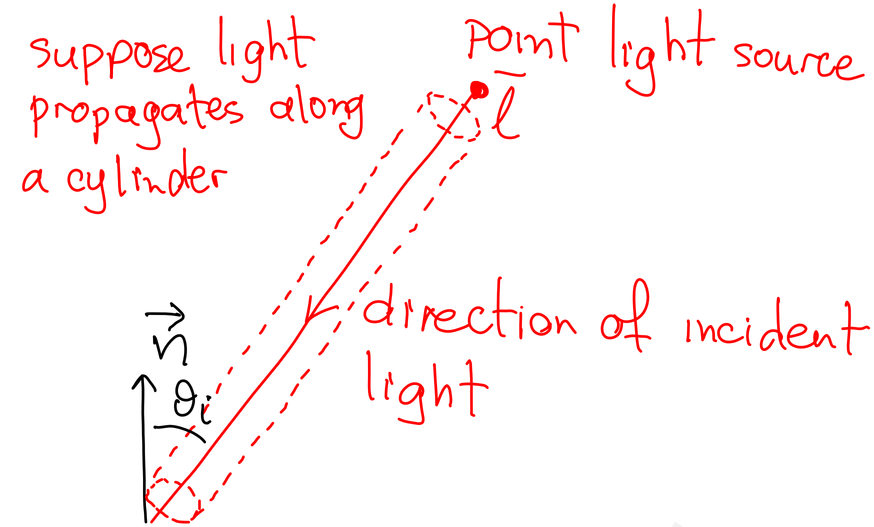
The Diffuse Component: Foreshortening

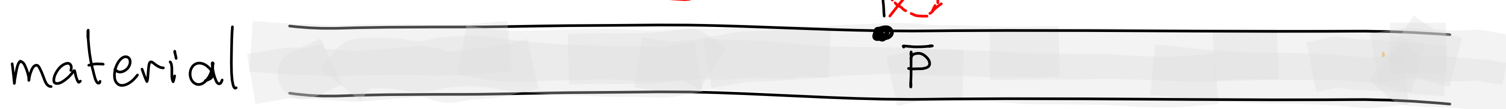
As the angle θ_i between $\vec{s}$ and $\vec{n}$ increases, the area of the surface around $\bar{p}$ receiving light increases

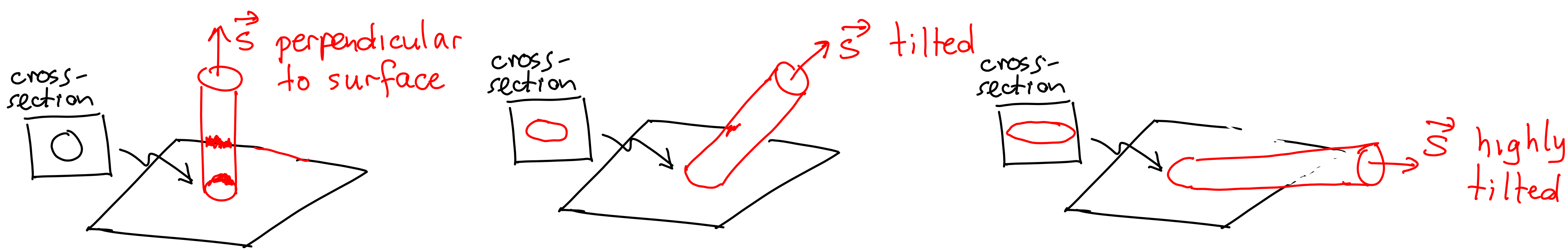
$\Rightarrow$ the light intensity received per unit area decreases.

this is called foreshortening

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







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

accounts for dimming due to foreshortening

The Diffuse Component: Foreshortening

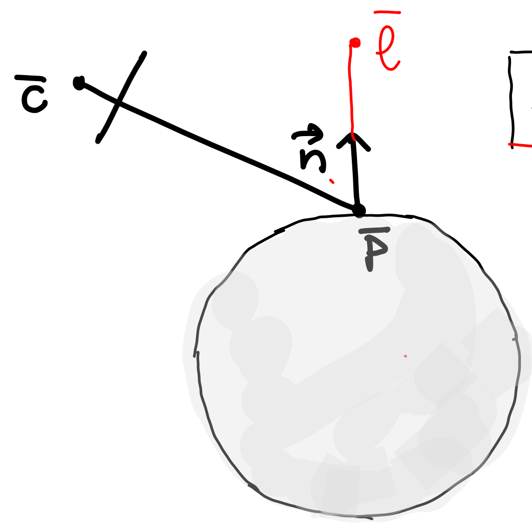
As the angle θ_i between $\vec{s}$ and $\vec{n}$ increases, the area of the surface around $\bar{p}$ receiving light increases

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

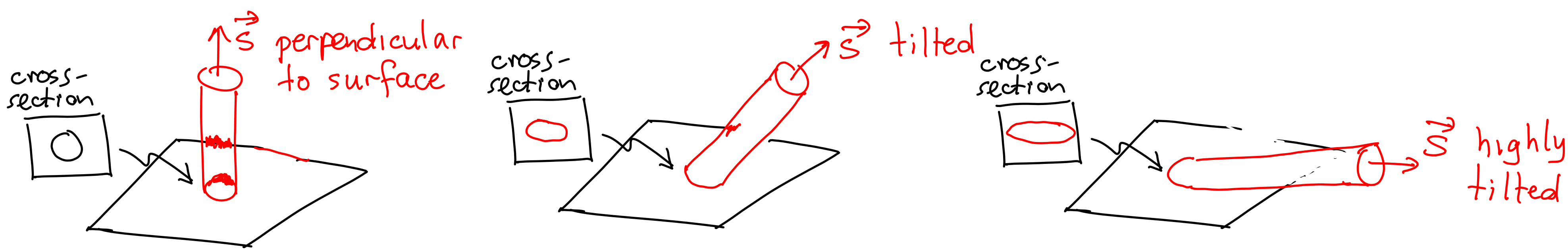
this is called foreshortening

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

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



$$\boxed{I_p = r_d \cdot I}$$



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

accounts for dimming due to foreshortening

The Diffuse Component: Foreshortening

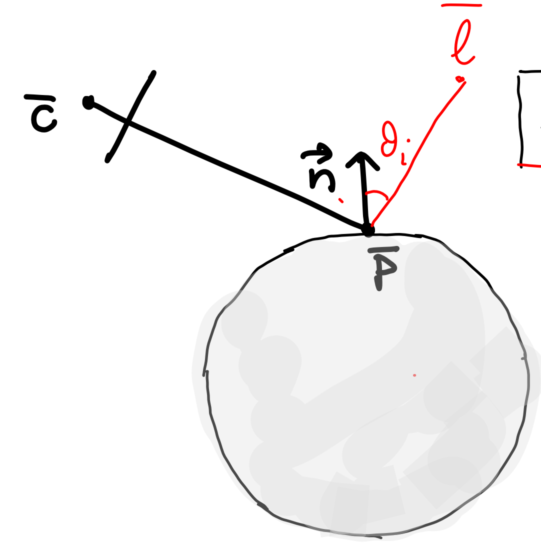
As the angle θ_i between $\vec{s}$ and $\vec{n}$ increases, the area of the surface around $\bar{p}$ receiving light increases

⇒ the light intensity received per unit area decreases.

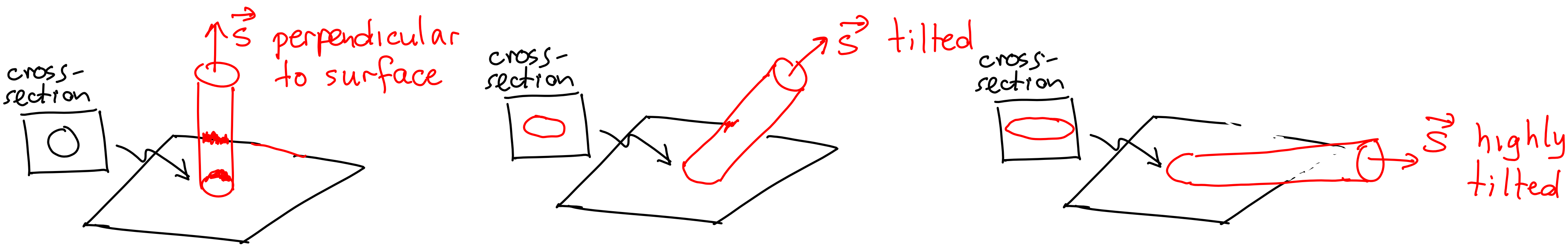
this is called foreshortening

⇒ point $\bar{p}$ will appear dimmer

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



$$\boxed{I_p = r_d\, I \cdot \cos\theta_i}$$



$$I_{\bar{p}} = r_d \cdot I \cdot \max(0, \underbrace{\vec{s}\cdot\vec{n}}_{\text{accounts for dimming due to foreshortening}})$$

The Diffuse Component: Foreshortening

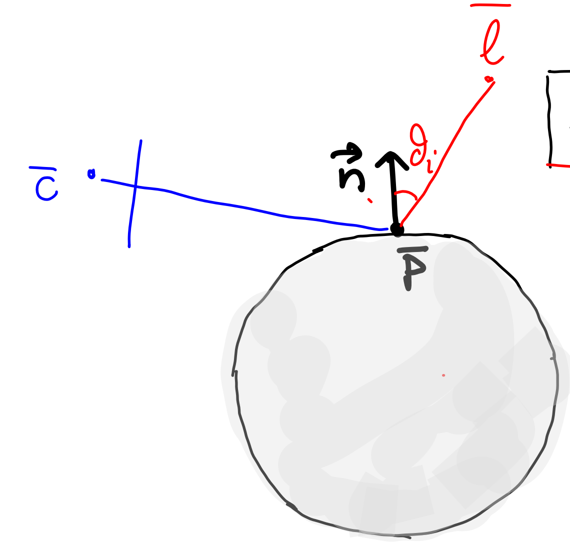
As the angle θ_i between $\vec{s}$ and $\vec{n}$ increases, the area of the surface around $\bar{p}$ receiving light increases

⇒ the light intensity received per unit area decreases.

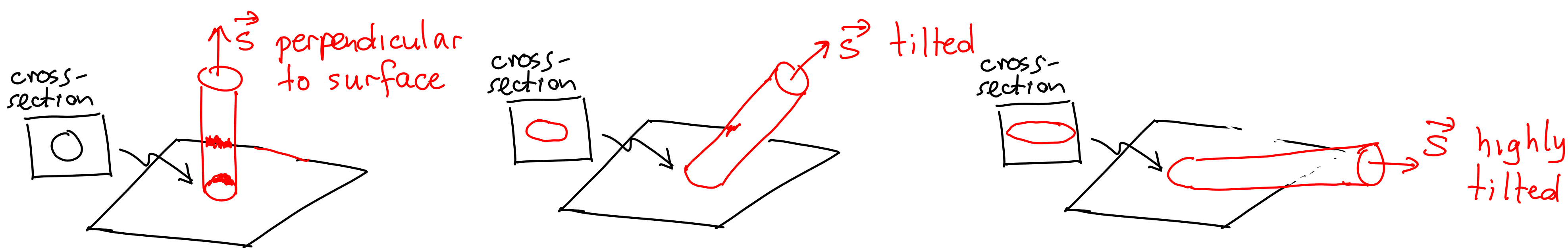
this is called foreshortening

⇒ point $\bar{p}$ will appear dimmer

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



$$\boxed{I_p = r_d I \cdot \cos\theta_i}$$



$$I_{\bar{p}} = r_d \cdot I \cdot \max\Big(0, \underbrace{\vec{s}\cdot\vec{n}}_{\text{accounts for dimming due to foreshortening}}\Big)$$

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

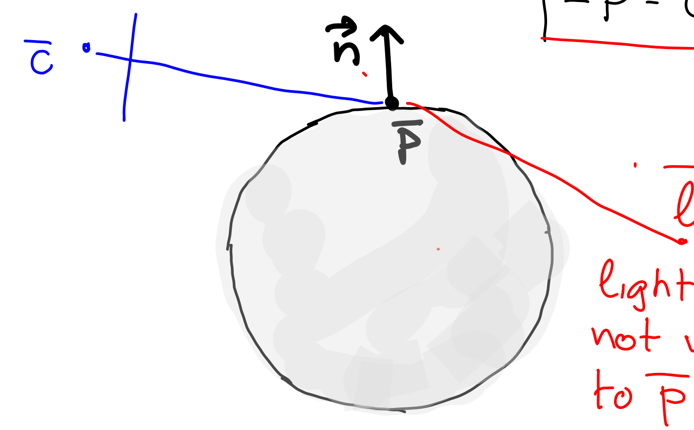
⇒ the light intensity received per unit area decreases.

this is called foreshortening

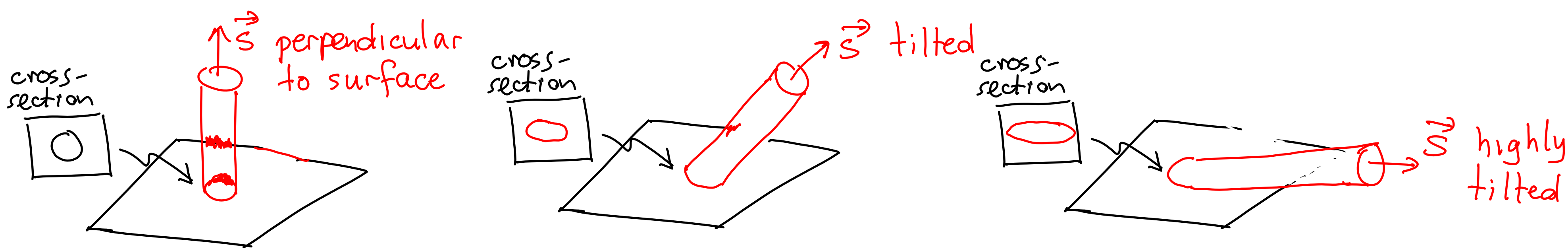
⇒ point $\bar{p}$ will appear dimmer

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

$\boxed{I_p = 0}$

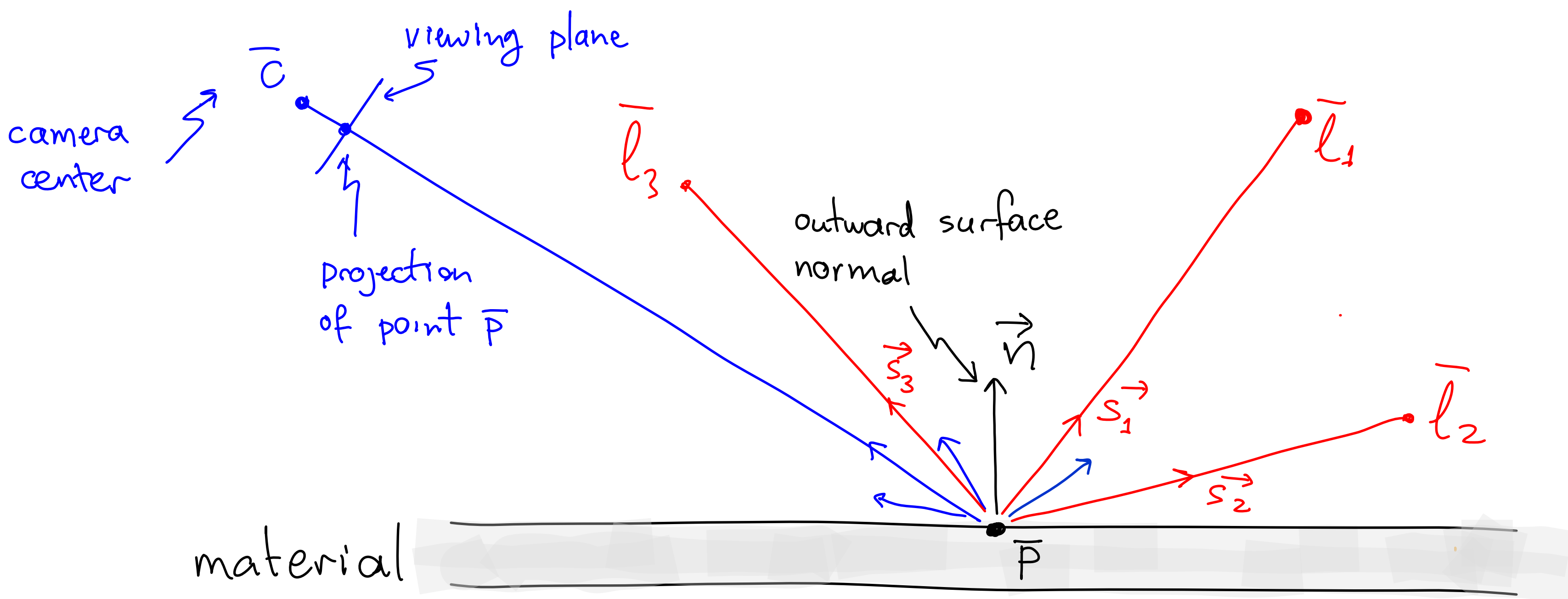


light source not visible to $\bar{p}$



$$I_{\bar{p}} = r_d \cdot I \cdot \underbrace{\max}_{\text{accounts for cases where light source not visible}}\left(0,\ \vec{s}\cdot\vec{n}\right)$$

The Diffuse Component: Multiple Lights



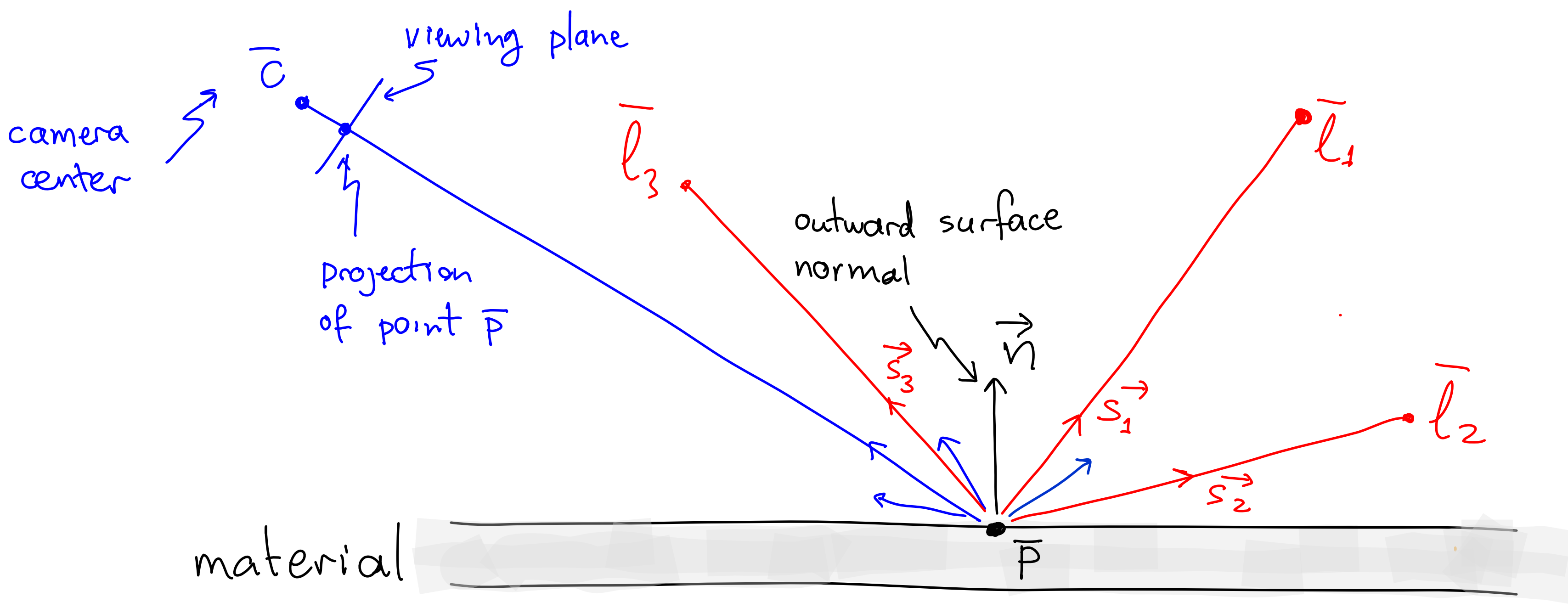
- A diffuse point looks the same from all viewing positions
- When the scene is illuminated by many point sources, we just sum up their contributions to the diffuse component

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

intensity at projection of $\bar{p}$

intensity of source i

The Diffuse Component: Incorporating Color



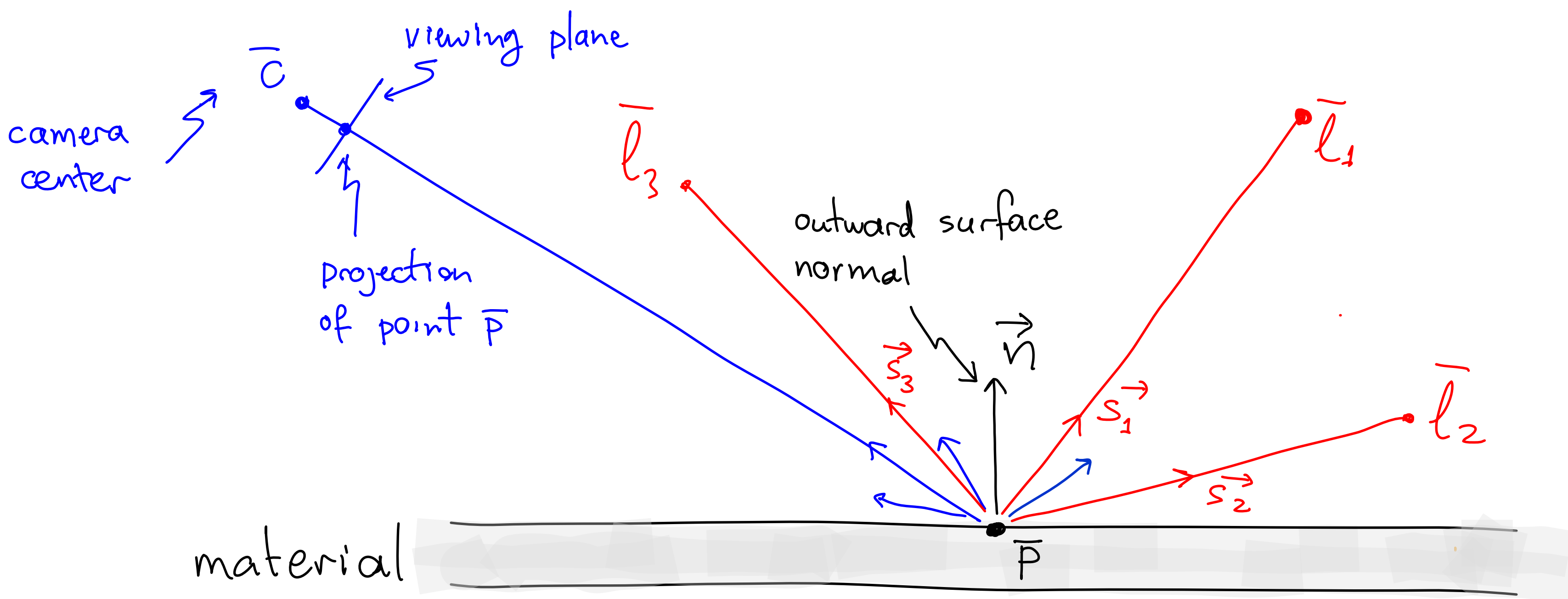
- A diffuse point looks the same from all viewing positions
- Coloured sources and coloured objects are handled by considering the RGB components of each colour separately

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

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

intensity of color component q for light source i

The Diffuse Component: General Equation



Putting it all together:

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

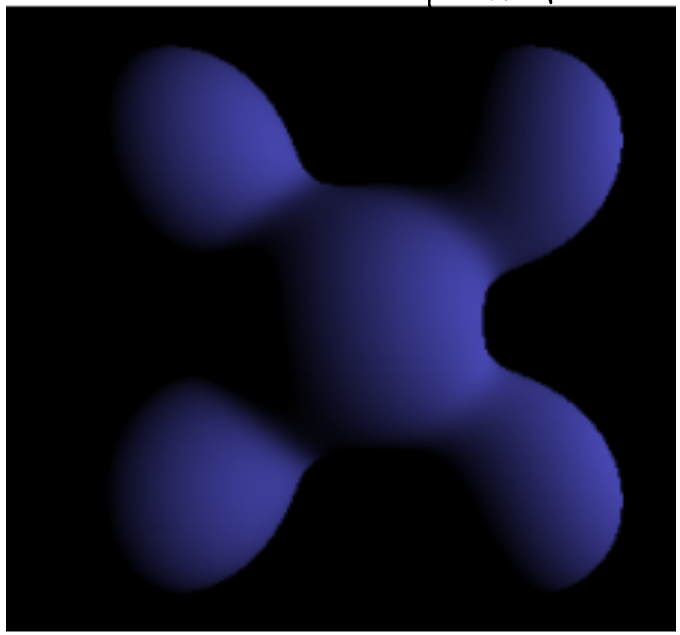
Topic 9:

Lighting & Reflection models

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

Phong Reflection: Ambient Component

Brad Smith, Wikipedia

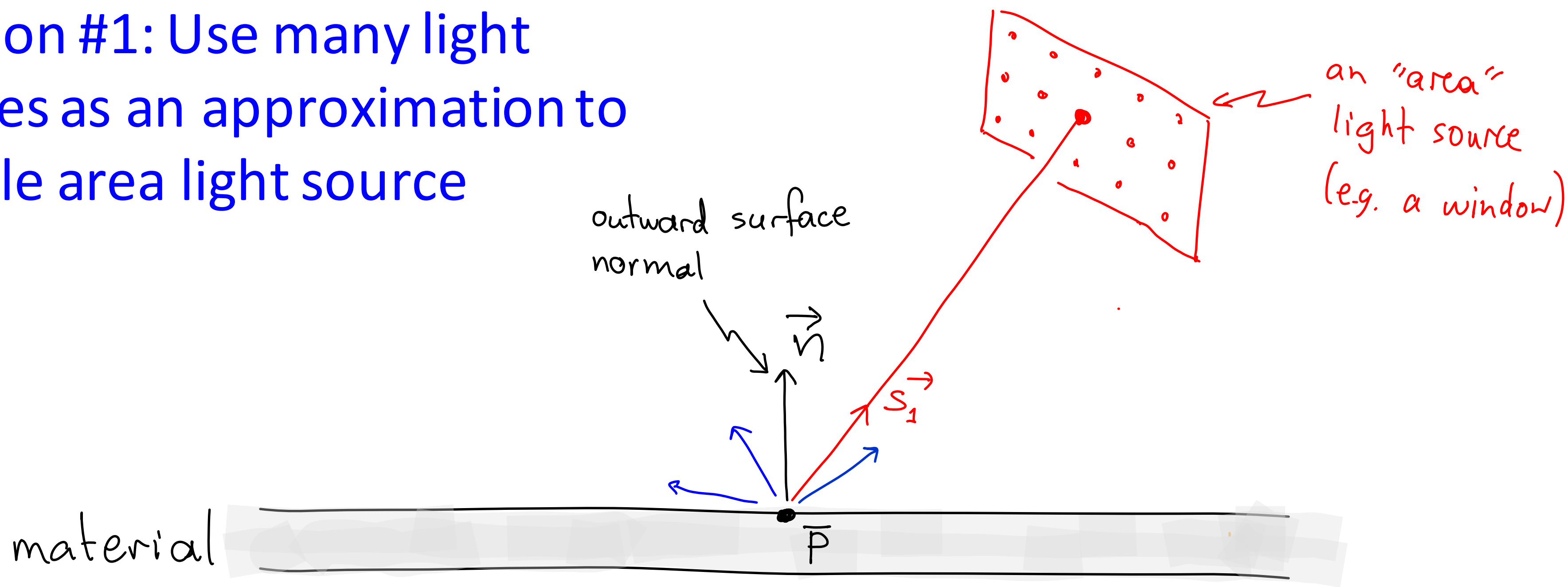


- Diffuse reflectance with a single point light source produces strong shadows
- Surface patches with $\vec{s} \cdot \vec{n} < 0$ are perfectly black

  ➔Looks unnatural

Phong Reflection: Ambient Component

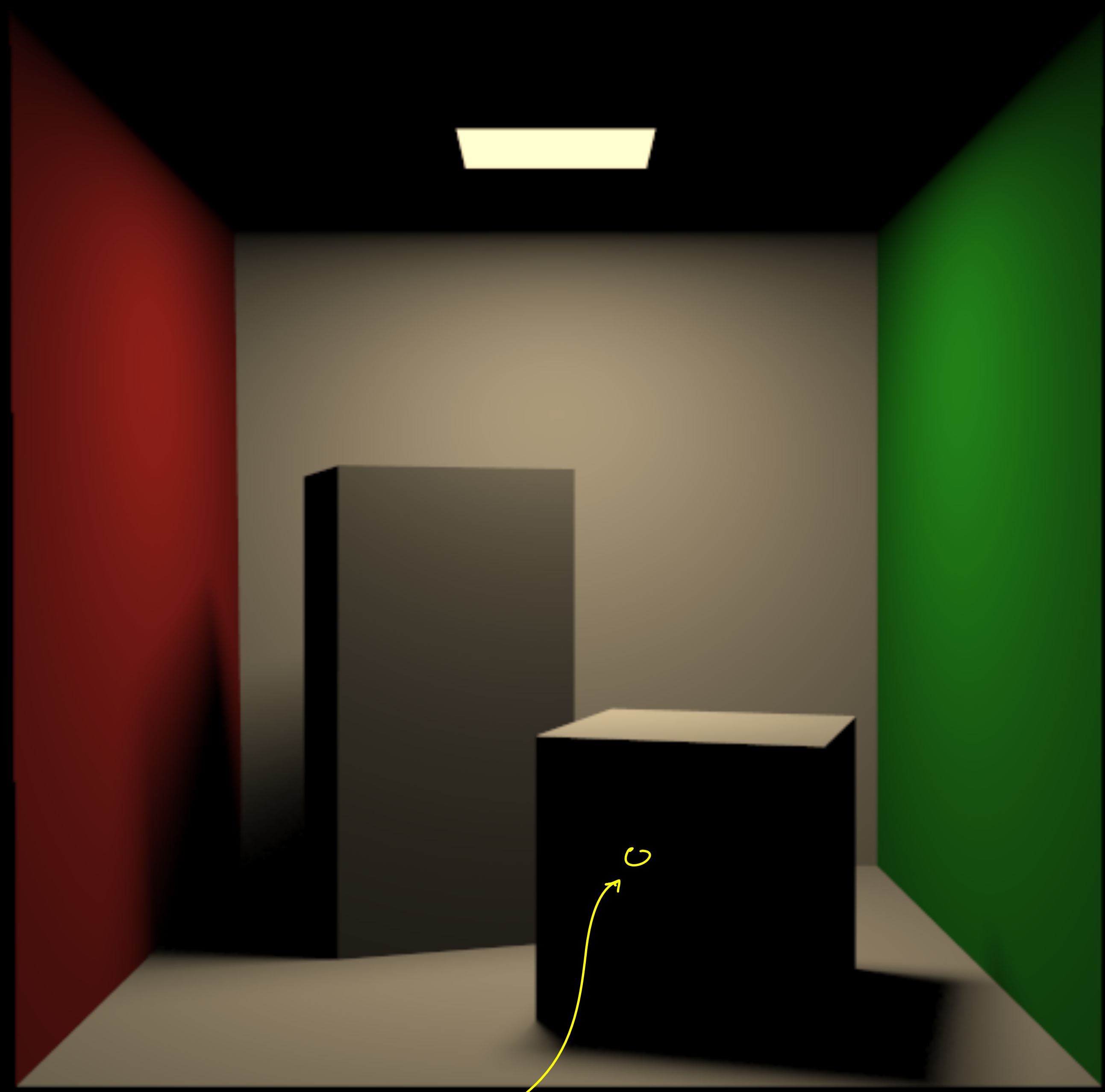
Solution #1: Use many light sources as an approximation to a single area light source



- 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

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

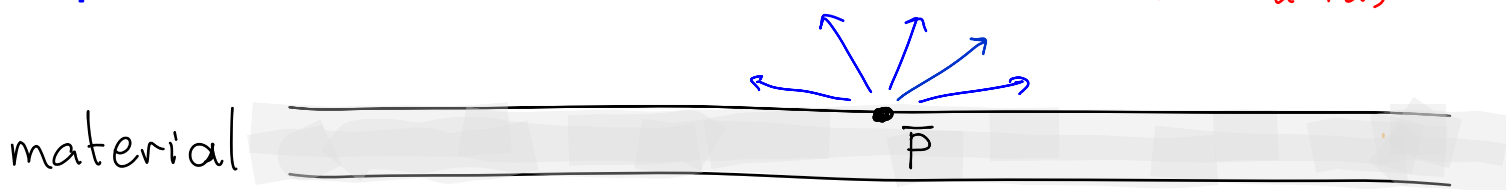
- Solution#2: (simpler) Use an "ambient" term that is independent of any light source or surface normal.
- This term is not meaningful in terms of physics but improves appearance over pure diffuse reflection.

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

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

ambient reflection coefficient (often $r_a = r_d$)

intensity of ambient illumination

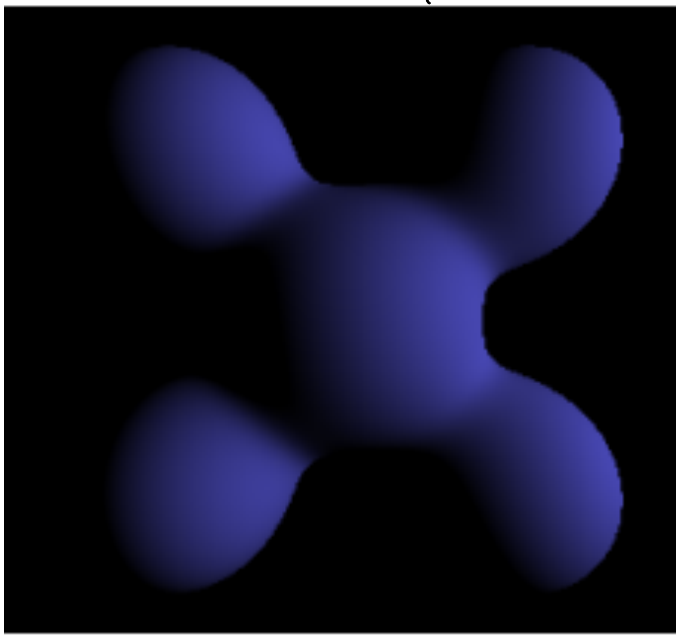


- Diffuse reflectance with a single point light source produces strong shadows
- Surface patches with $\vec{s} \cdot \vec{n} < 0$ are perfectly black

  ➔Looks unnatural

Phong Reflection: Ambient Component

Brad Smith, Wikipedia



Diffuse

Brad Smith, Wikipedia

Ambient

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

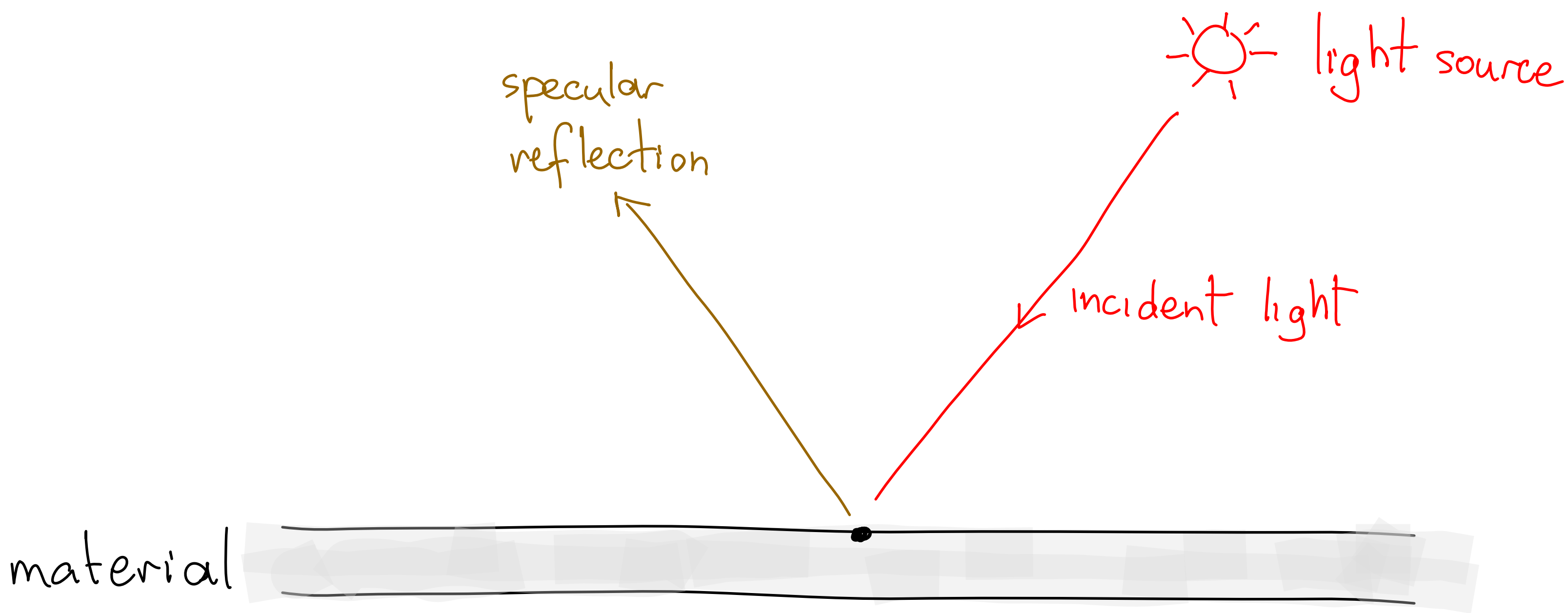
  ➔Looks unnatural

Topic 9:

Lighting & Reflection models

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

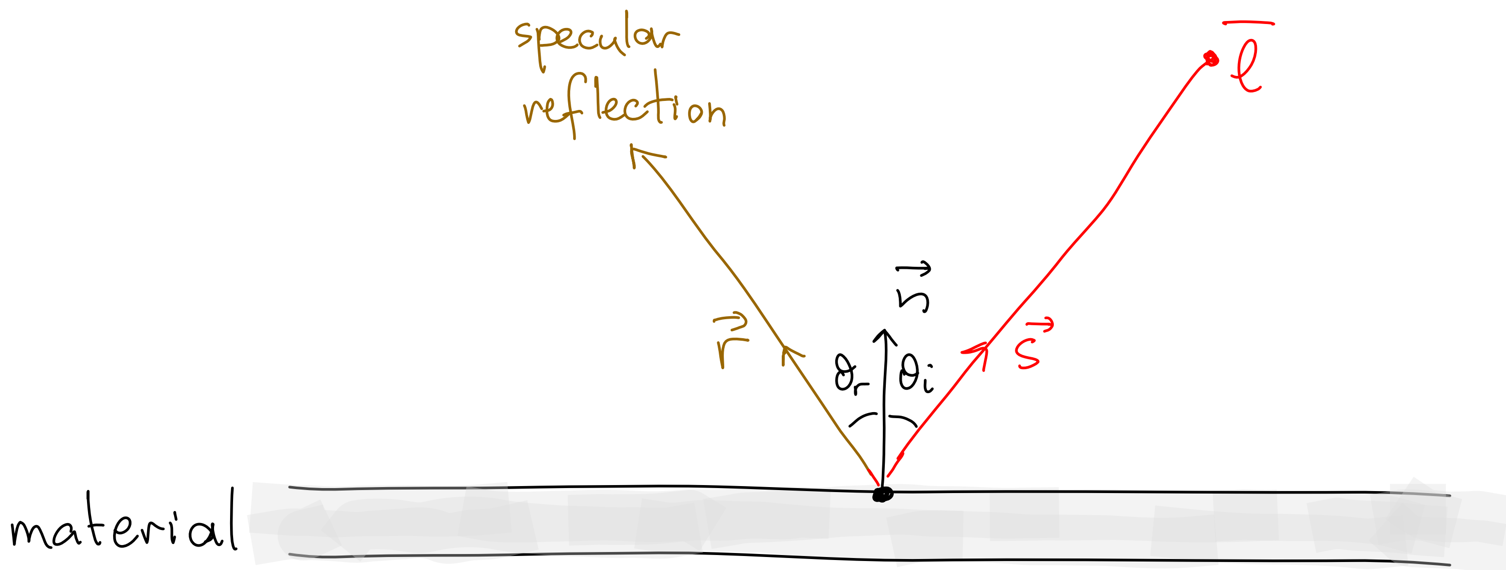
Phong Reflection: The Specular Component



Specular Reflection:

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

The Ideal Specular Component

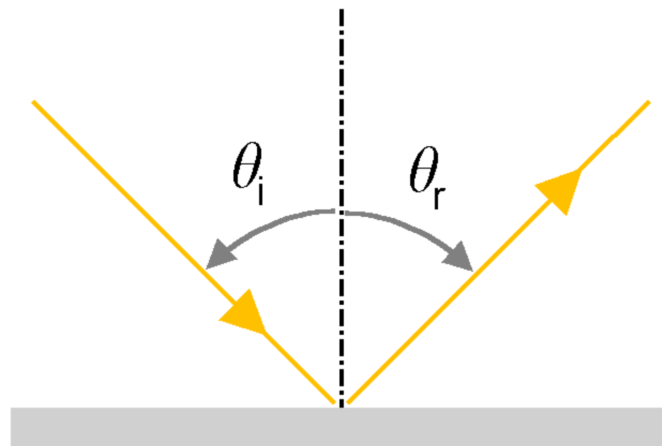


- Idea: For each incident reflection direction, $\vec{s}$
  there is one emittent direction $\vec{r}$
- It is an idealization of a mirror:

$$\underbrace{angle(\vec{n}, \vec{s})}_{\theta_i} = \underbrace{angle(\vec{n}, \vec{r})}_{\theta_r}$$

The Ideal Specular Component

Romeiro et al, ECCV'08



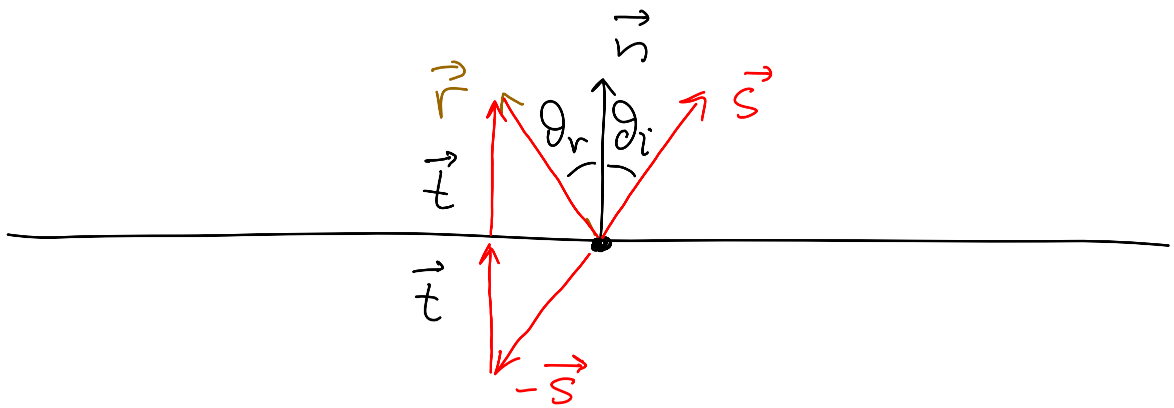
Panjasan, Wikipedia

- Idea: For each incident reflection direction, $\vec{s}$
  there is one emittent direction $\vec{r}$
- It is an idealization of a mirror:

$$\underbrace{angle(\vec{n}, \vec{s})}_{\theta_i} = \underbrace{angle(\vec{n}, \vec{r})}_{\theta_r}$$

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

The Ideal Specular Component



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

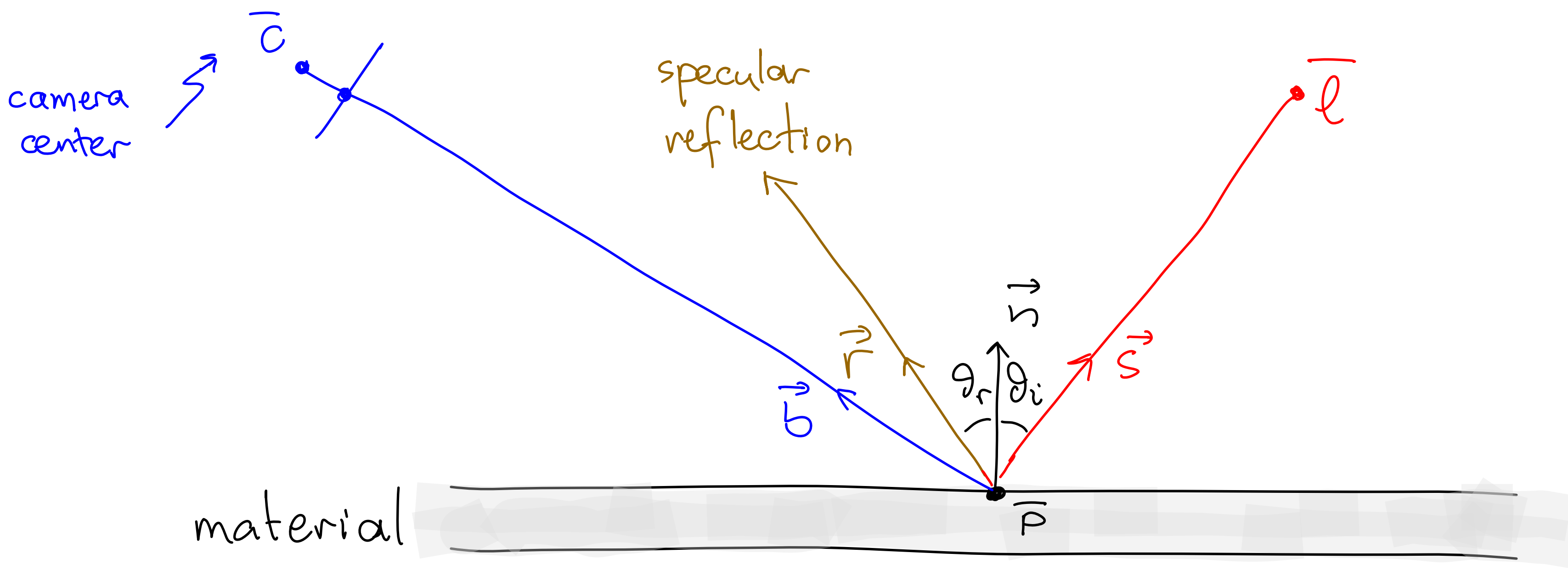
$\vec{t}$ = projection of vector $\vec{s}$ onto vector $\vec{n}$

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

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

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

The Ideal Specular Component



Ideal specular reflection term:

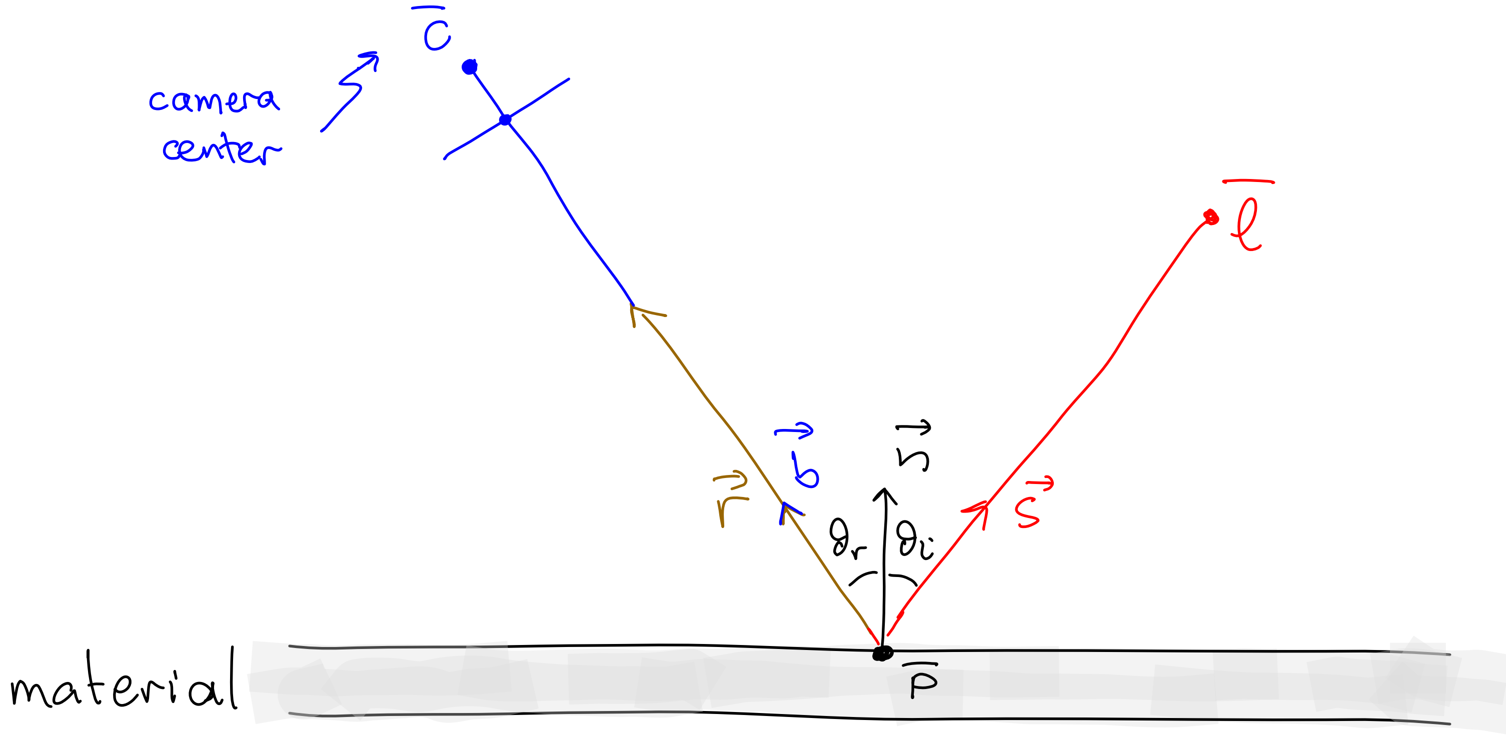
$$I = r_s \, I_s \, \delta\overbrace{\left(\vec{r}\cdot\vec{b} - 1\right)}^{\text{is 1 if and only if camera is along vector } \vec{r}} \quad \text{where} \quad \delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

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:

$$I = r_s \, I_s \, \delta\big(\overbrace{\vec{r}\cdot\vec{b} - 1}^{\text{is 1 if and only if camera is along vector } \vec{r}}\big) \quad \text{where} \quad \delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

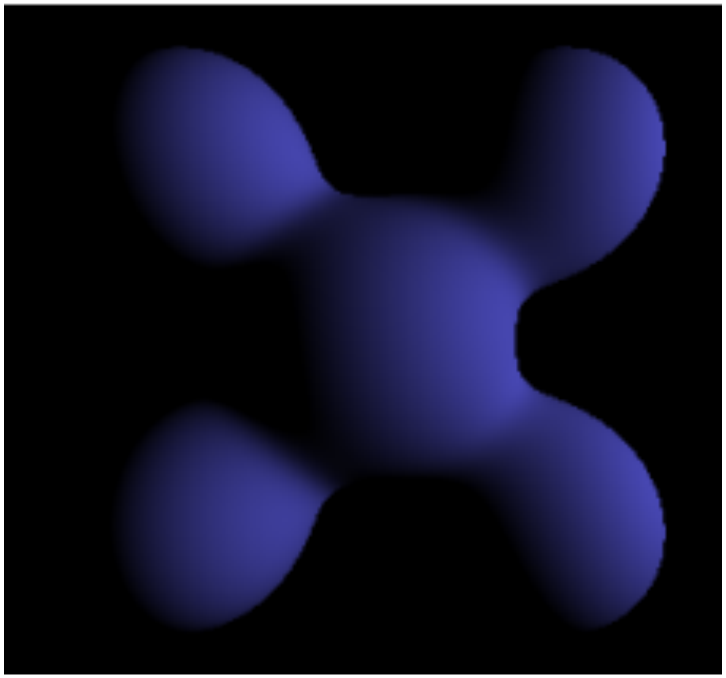
r_s: specular reflection coefficient

I_s: intensity of specular light source

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

The Ideal Specular Component

Brad Smith, Wikipedia



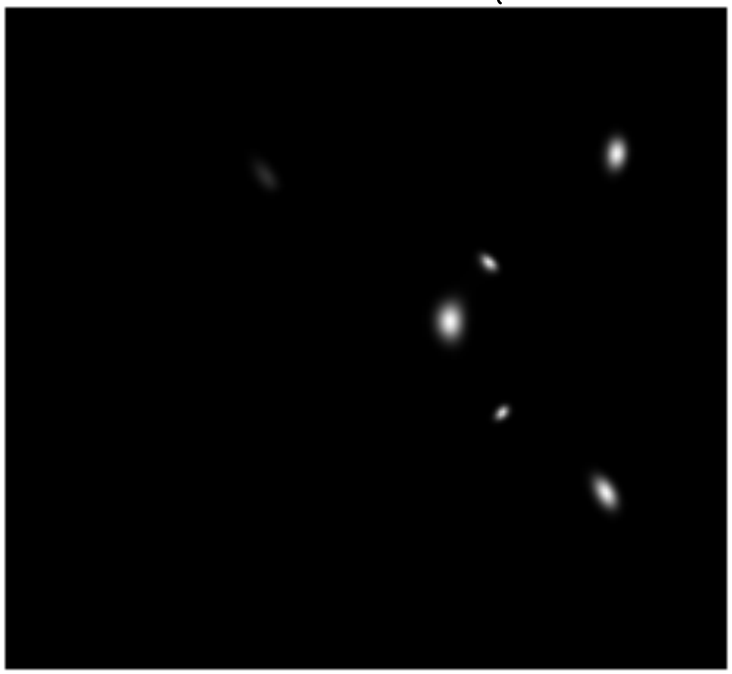
Ideal specular reflection term:

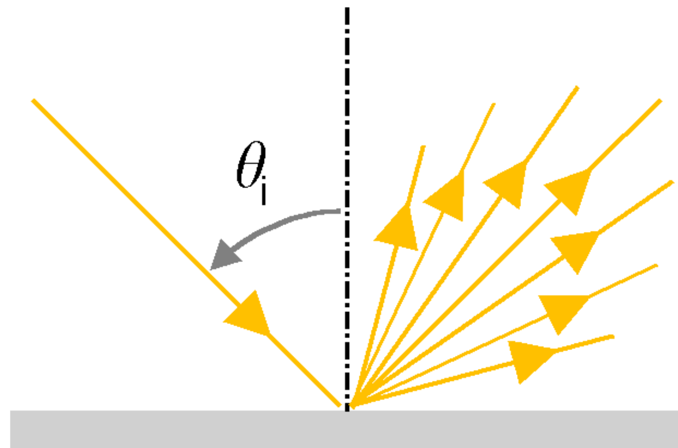
$$I = \underset{\text{specular reflection coefficient}}{r_s}\ \underset{\text{intensity of specular light source}}{I_s}\ \delta\big(\overbrace{\vec{r}\cdot\underset{\text{unit vector in camera direction}}{\vec{b}} - 1}^{\text{is 1 if and only if camera is along vector } \vec{r}}\big) \quad \text{where} \quad \delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

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

Phong Reflection: Off-Specular Reflection

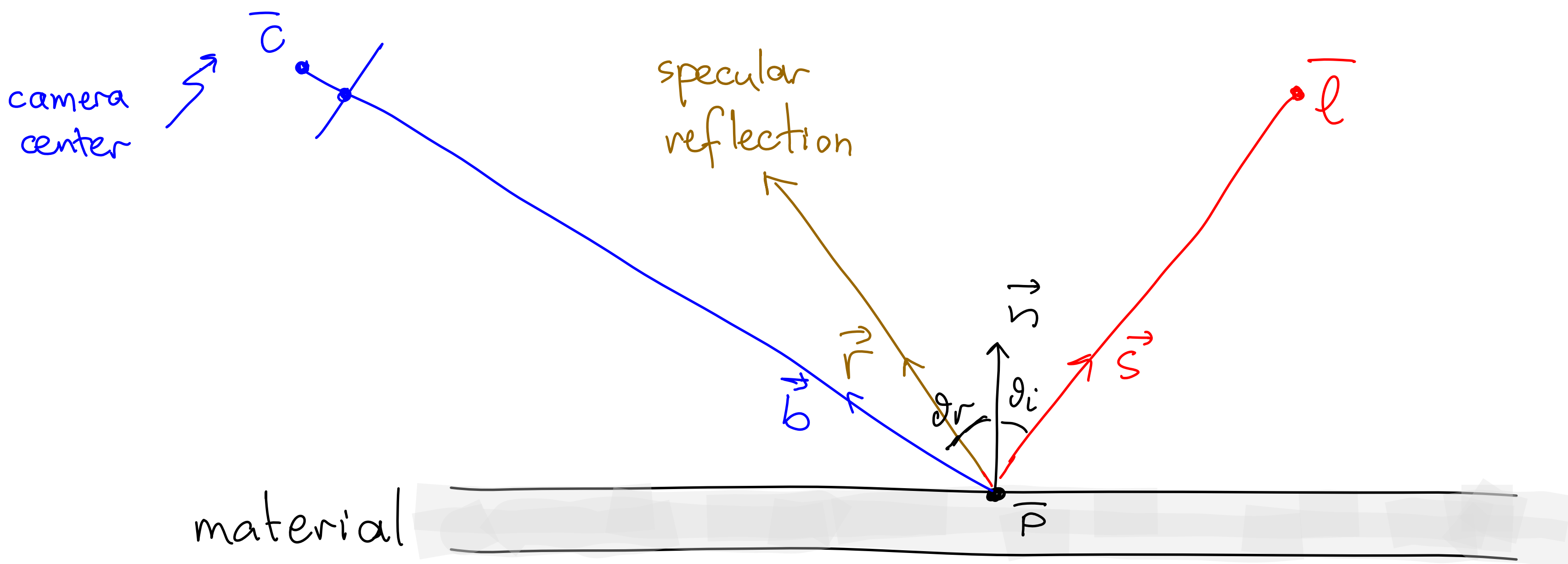
Brad Smith, Wikipedia





Panjasan, Wikipedia

The Specular Component: Basic Equation



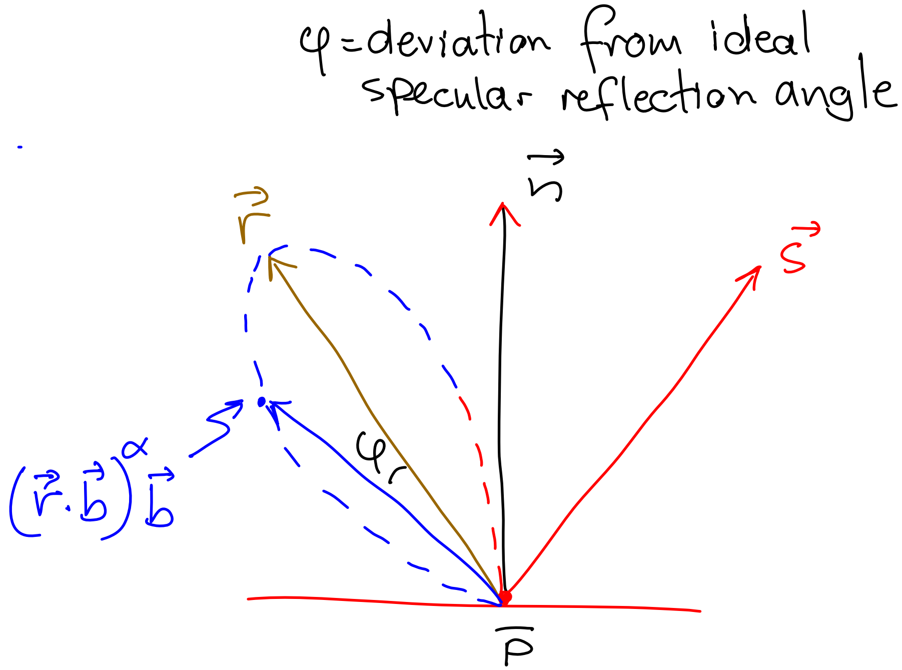
In reality, most specular surfaces reflect light into directions near the perfect direction (e.g. highlights in plastics, metals)

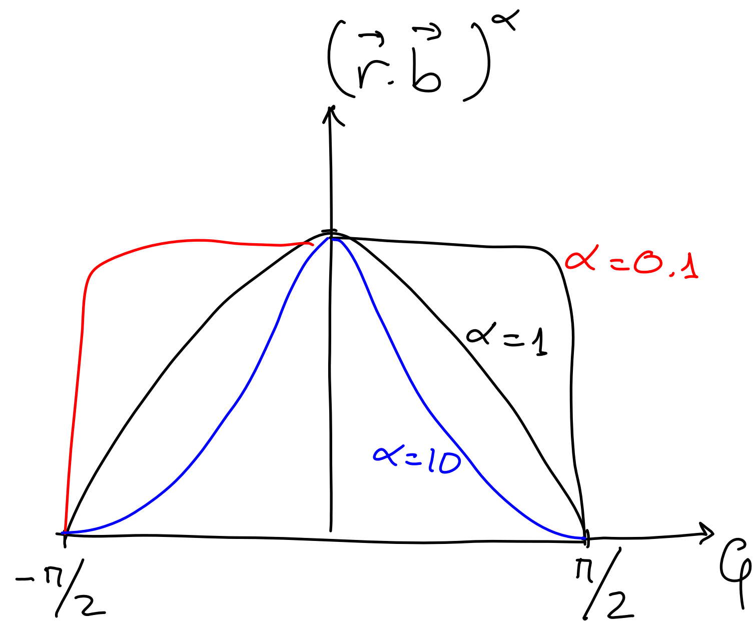
→ Introduce cosine power

$$I = r_s I_s \max\left(0, \underbrace{\vec{r}\,\vec{b}}_{=1 \text{ when } \vec{r}=\vec{b}}\right)^{\alpha}$$

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

The Specular Component: Visualization



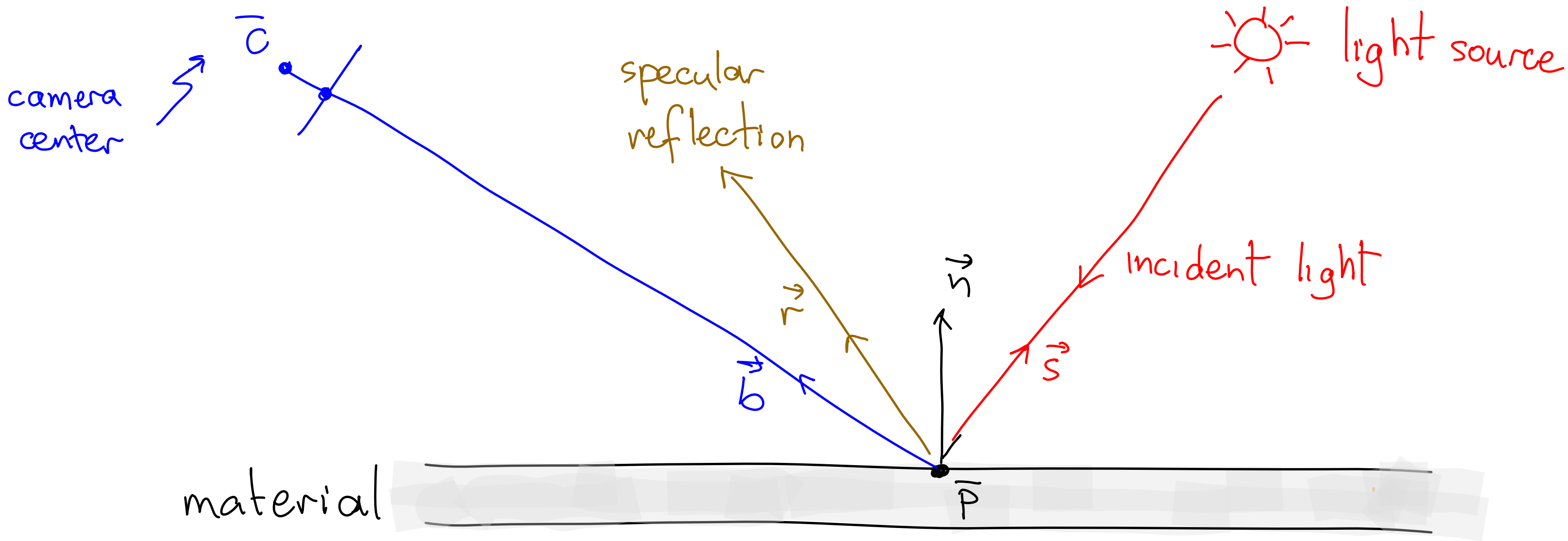


The length of vector $(\vec{r}\cdot\vec{s})^{\alpha}\vec{b}$ represents the contribution of the specular term when the camera is along $\vec{b}$

$$I = r_s I_s \max\left(0, \underbrace{\vec{r}\cdot\vec{b}}_{=1 \text{ when } \vec{r}=\vec{b}}\right)^{\alpha}$$

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

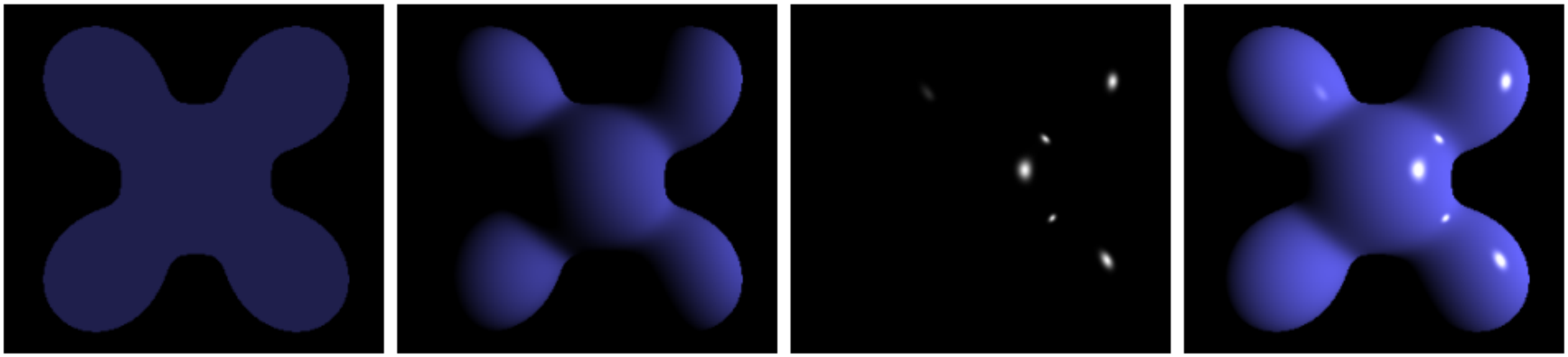
Phong Reflection: The General Equation



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

Phong Reflection: The General Equation

Brad Smith, Wikipedia



Ambient + Diffuse + Specular = Phong Reflection

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

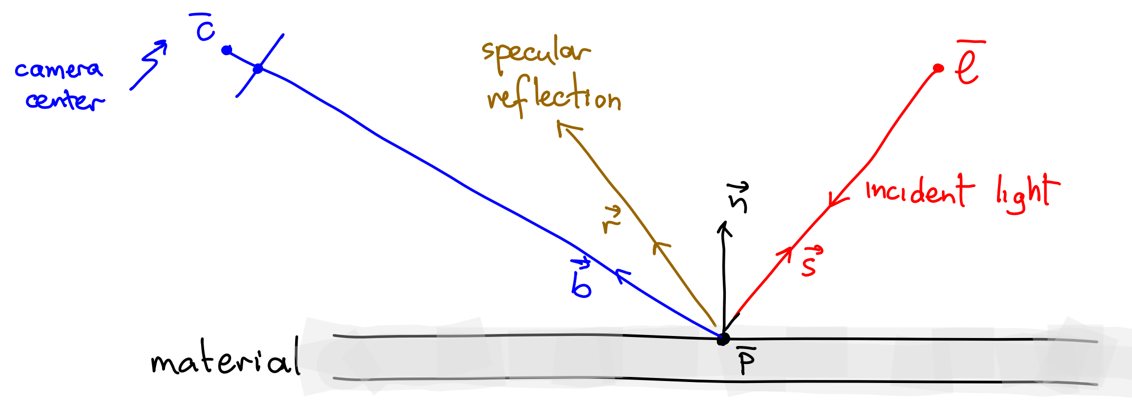
Topic 10:

Shading

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

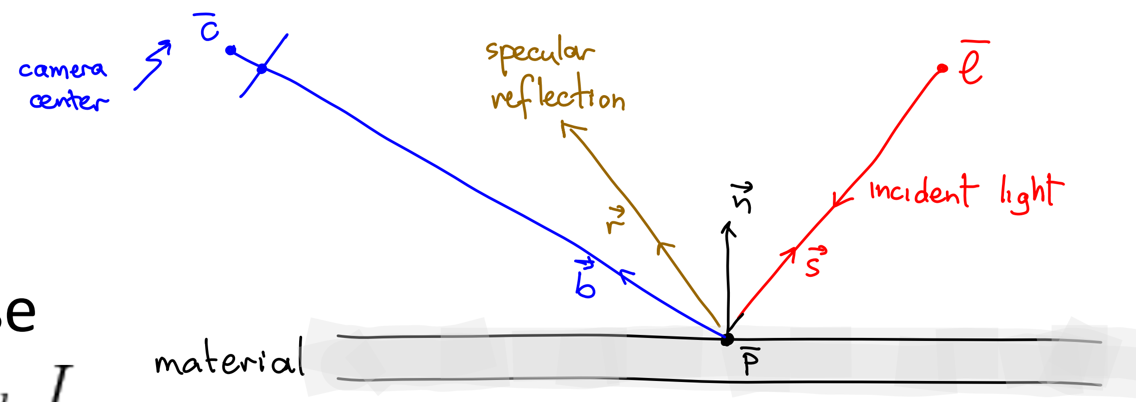
$$\underbrace{L(\vec{b},\vec{n},\vec{s})}_{\text{intensity at projection of point } \bar{p}} = \underbrace{r_a I_a}_{\text{ambient}} + \underbrace{r_d I_d \max(0, \vec{n}\cdot\vec{s})}_{\text{diffuse}} + \underbrace{r_s I_s \max(0, \vec{r}\cdot\vec{b})^{\alpha}}_{\text{specular}}$$

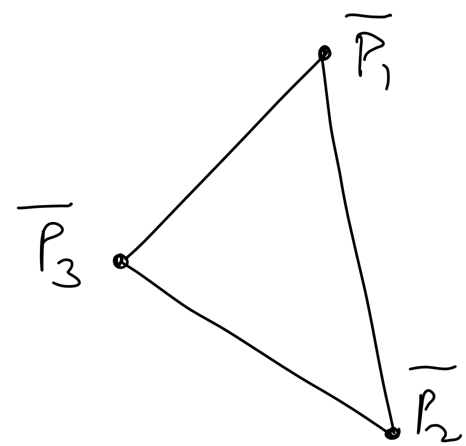
Shading: Motivation

<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

<u>Shade</u> every pixel in triangle's projection.



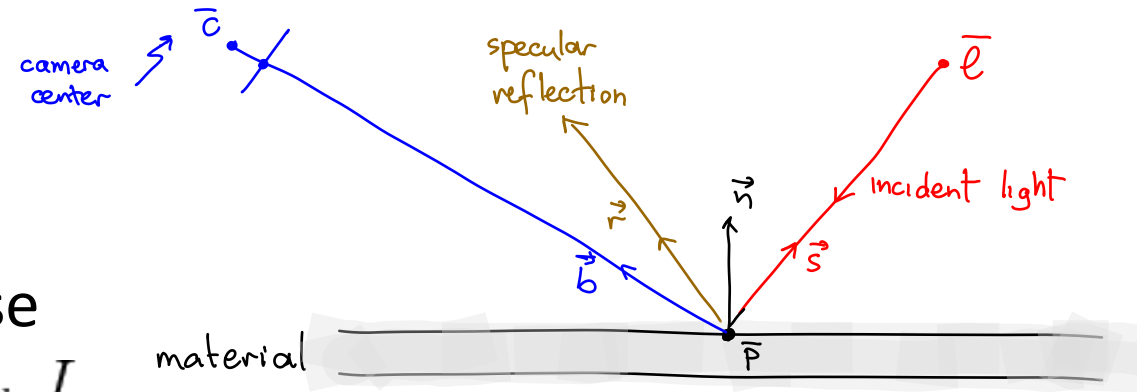


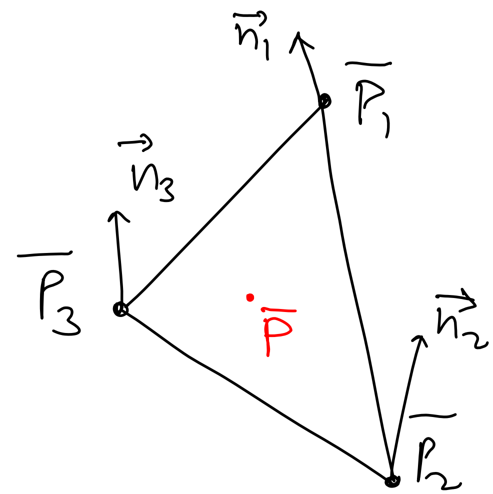
$$\underbrace{L(\vec{b},\vec{n},\vec{s})}_{\text{intensity at projection of point } \bar{p}} = \underbrace{r_a I_\alpha}_{\text{ambient}} + \underbrace{r_d I_d \max(0,\vec{n}\cdot\vec{s})}_{\text{diffuse}} + \underbrace{r_s I_s \max(0,\vec{r}\cdot\vec{b})^\alpha}_{\text{specular}}$$

Shading: Problem Definition

Given

- 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 $\bar{p}_1, \bar{p}_2, \bar{p}_3$





Goal

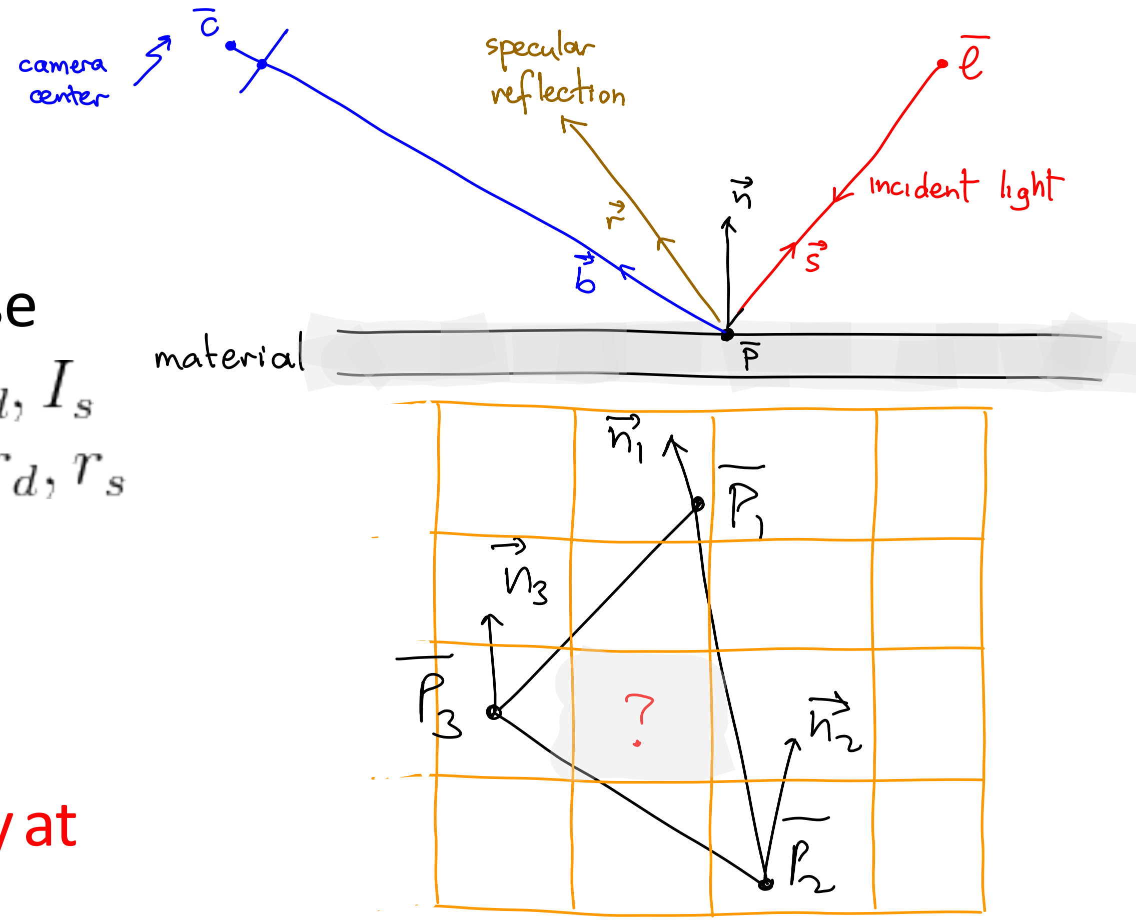
Computer colour/intensity at an interior point

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

Shading: Problem Definition

Given

- 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 $\bar{p}_1, \bar{p}_2, \bar{p}_3$



Goal

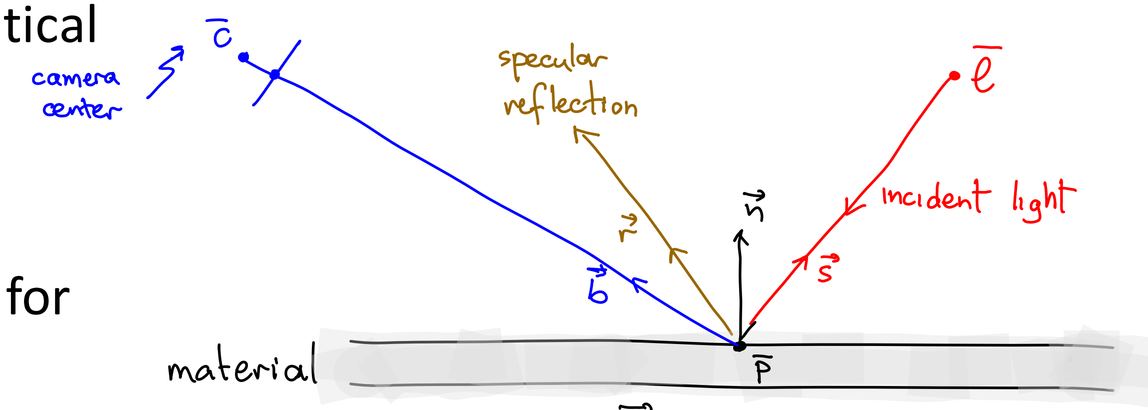
Computer colour/intensity at an interior pixel

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

Basic Approaches to Shading

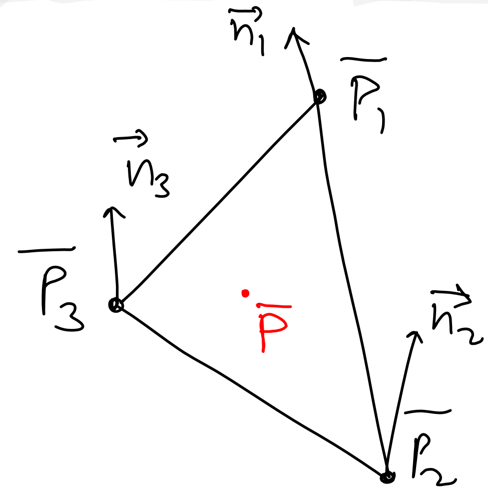
<u>Flat shading</u>

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



<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>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 $L(\vec{b}, \vec{n}, \vec{s})$

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

Topic 10:

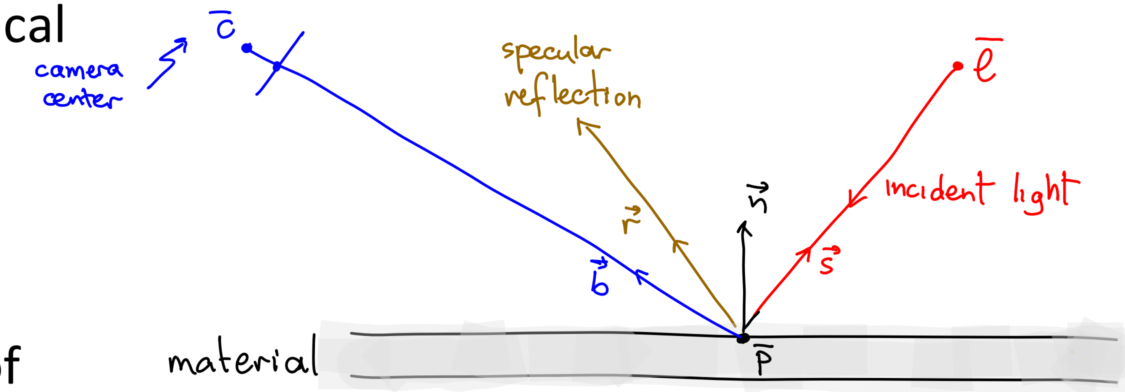
Shading

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

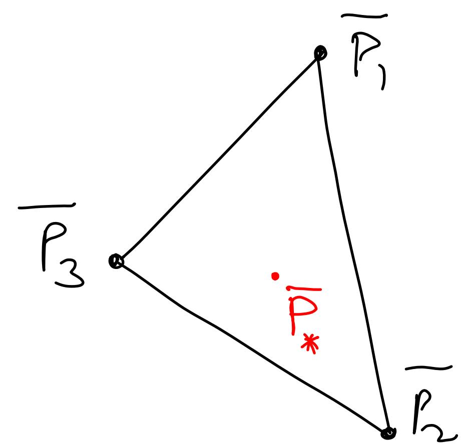
Flat Shading: Main Idea

Flat shading

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



- All points have same normal $\vec{n}$ (i.e. triangle is "flat")
- Phong model applied to center of triangle: $\bar{p}_* = \frac{1}{3}(\bar{p_1} + \bar{p_2} + \bar{p_3})$



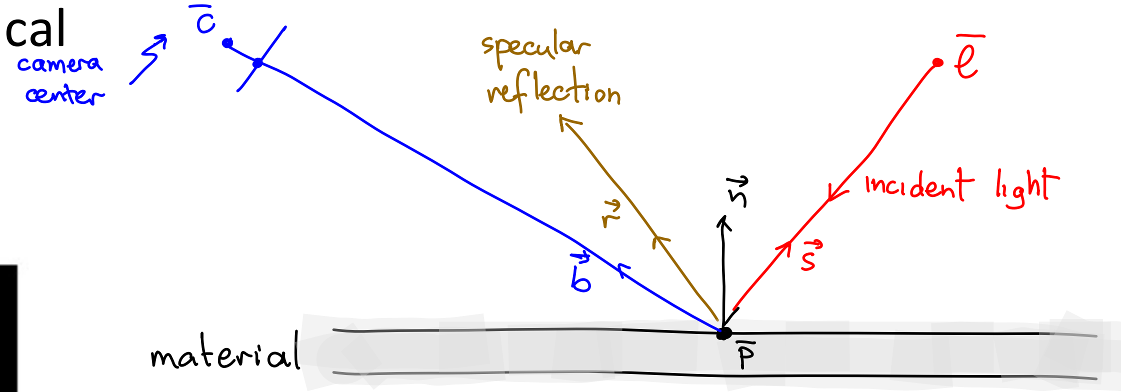
- (i.e. $\vec{b}, \vec{s}$ computed for $\bar{p}_*$)
- Triangle filled with colour/intensity $L(\vec{b}, \vec{n}, \vec{s})$

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

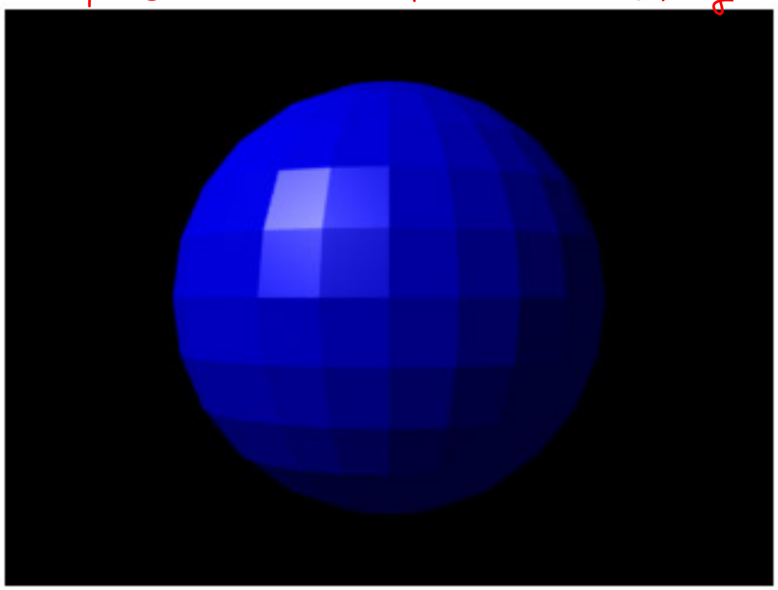
Flat Shading: Main Idea

Flat shading

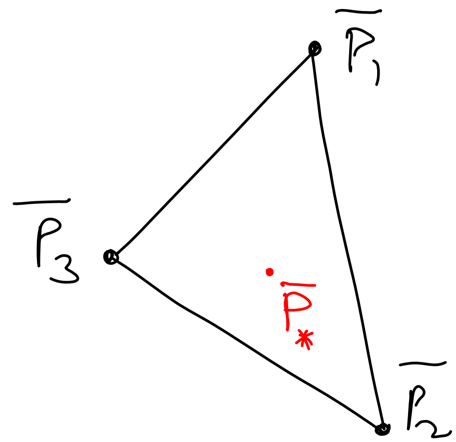
Draw all triangle points $\bar{p}$ with identical colour/intensity



Sphere with flat shading



Jalo, Wikipedia

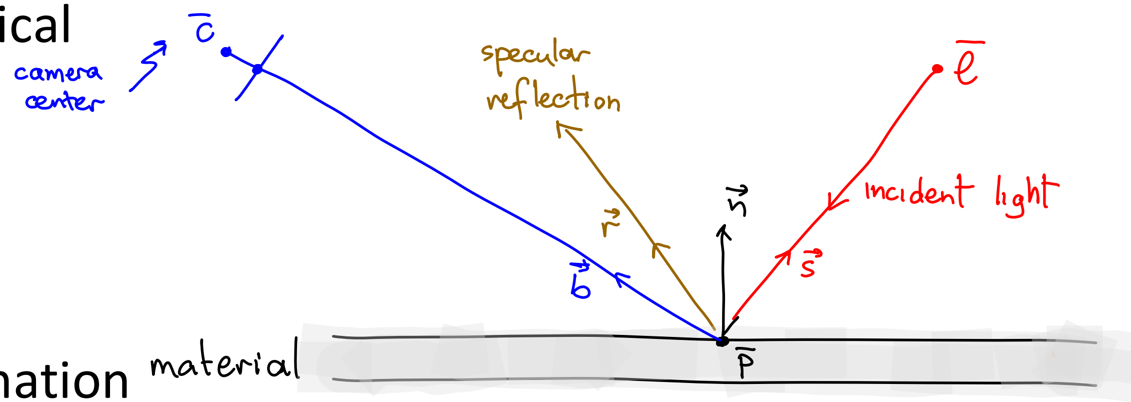


$$\underbrace{L(\vec{b},\vec{n},\vec{s})}_{\substack{\text{intensity at}\\ \text{projection of}\\ \text{point } \bar{p}}} = \underbrace{r_a I_a}_{\text{ambient}} + \underbrace{r_d I_d \max(0,\vec{n}\cdot\vec{s})}_{\text{diffuse}} + \underbrace{r_s I_s \max(0,\vec{r}\cdot\vec{b})^{\alpha}}_{\text{specular}}$$

Flat Shading: Key Issues

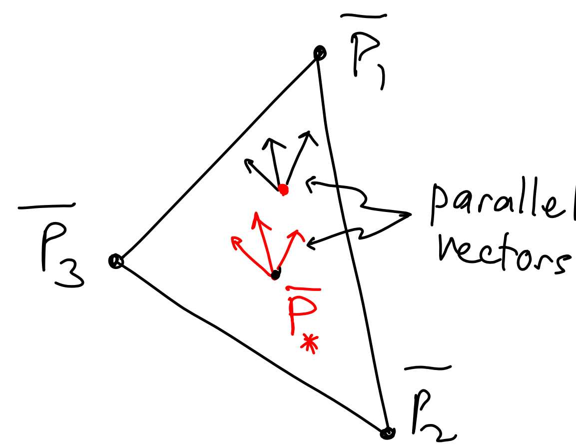
<u>Flat shading</u>

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)

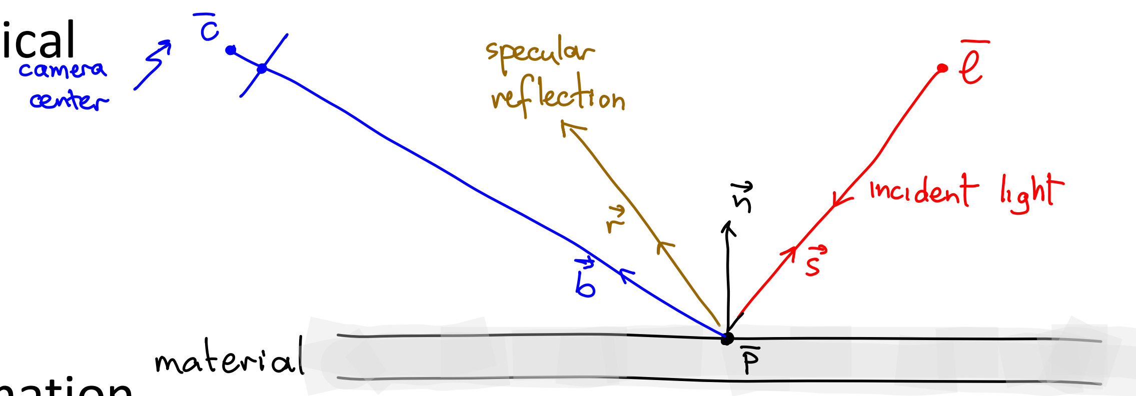


$$\underbrace{L(\vec b,\vec n,\vec s)}_{\text{intensity at projection of point } \bar p} = \underbrace{r_a I_a}_{\text{ambient}} + \underbrace{r_d I_d \max(0,\vec n\cdot\vec s)}_{\text{diffuse}} + \underbrace{r_s I_s \max(0,\vec r\cdot\vec b)^{\alpha}}_{\text{specular}}$$

Flat Shading: Key Issues

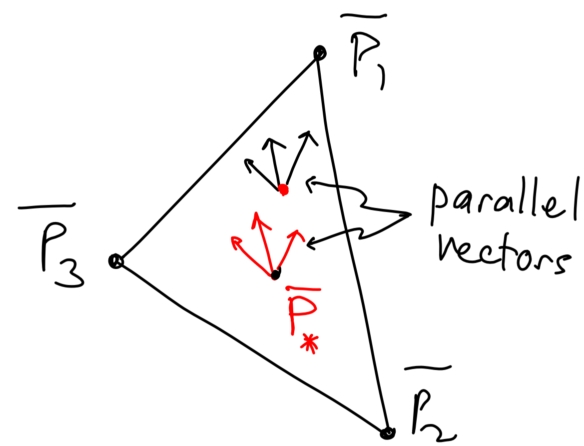
<u>Flat shading</u>

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!

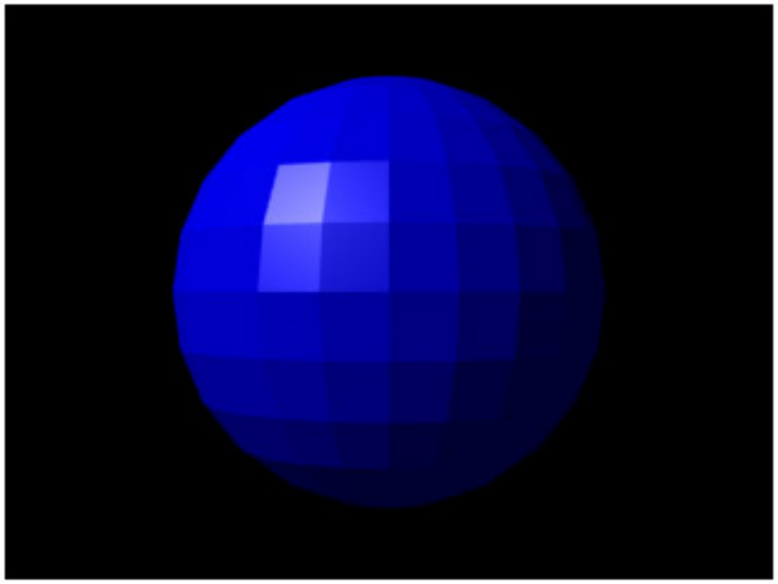
Topic 10:

Shading

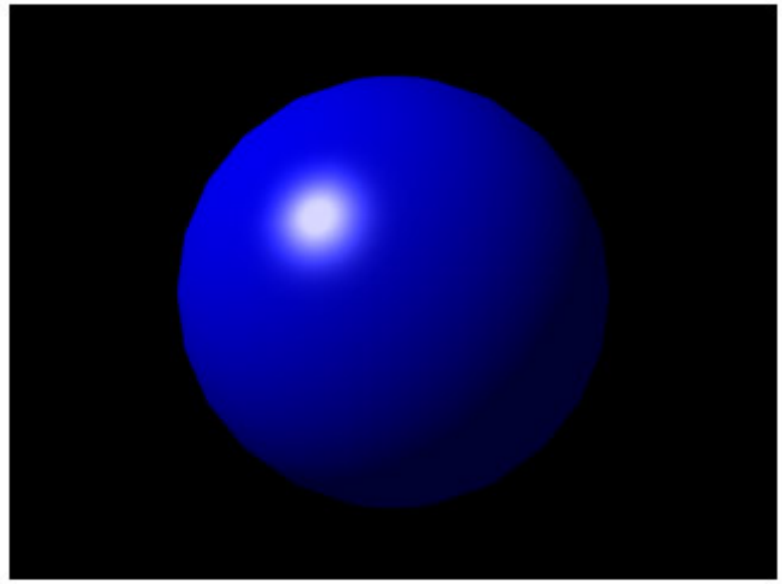
- Introduction to Shading
- Flat Shading
- Interpolative Shading
  - Gouraud shading
  - Phong shading

Interpolated Shading

Jalo, Wikipedia

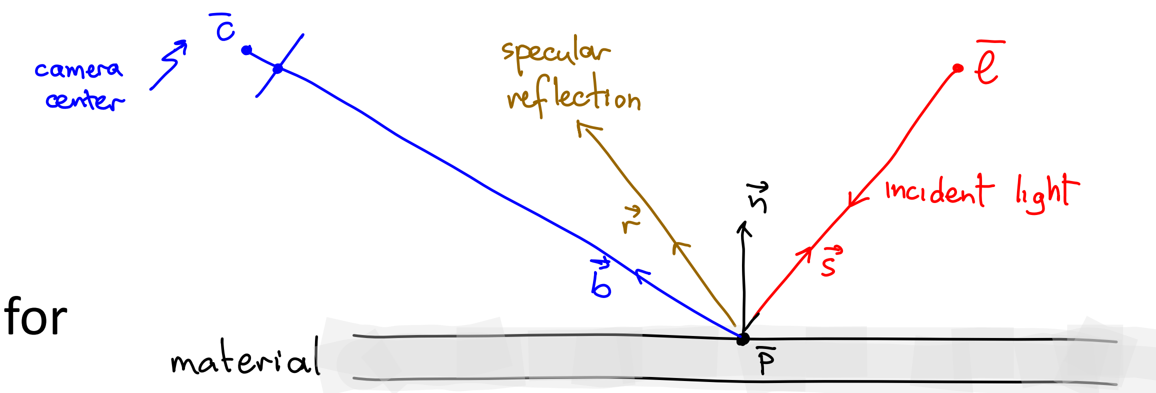


FLAT SHADING



PHONG SHADING

Interpolative Shading: Basic Approaches

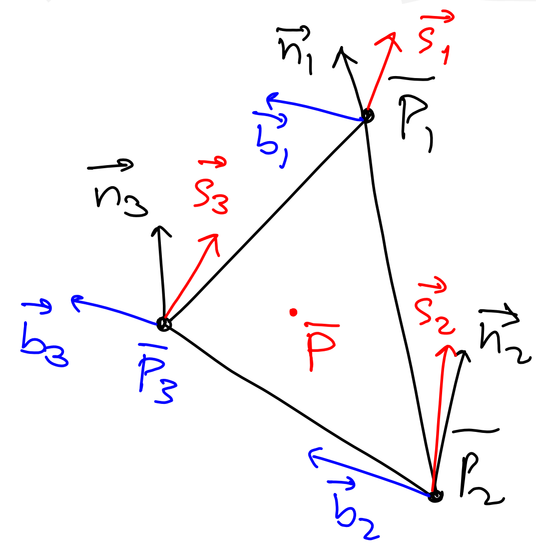


Gouraud shading

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 $L(\vec{b}, \vec{n}, \vec{s})$



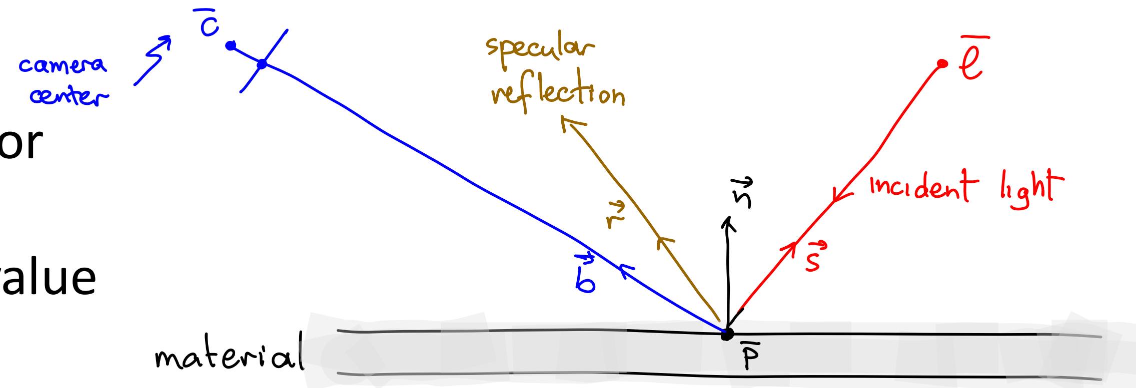
$$\underbrace{L(\vec{b}_i, \vec{n}_i, \vec{s}_i)}_{\text{intensity at projection of point } \bar{p}} = \underbrace{r_a I_\alpha}_{\text{ambient}} + \underbrace{r_d I_d \max(0, \vec{n}_i \cdot \vec{s}_i)}_{\text{diffuse}} + \underbrace{r_s I_s \max(0, \vec{r}_i \cdot \vec{b}_i)^\alpha}_{\text{specular}}$$

Topic 10:

Shading

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

Gouraud Shading: Computation at Vertices

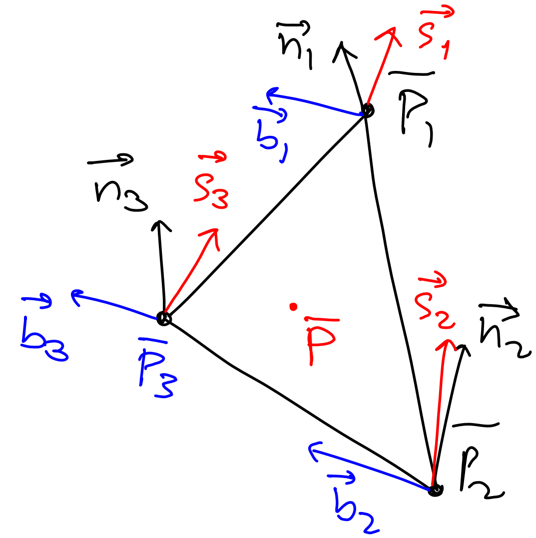


<u>Gouraud shading</u>

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 $\bar{p}$

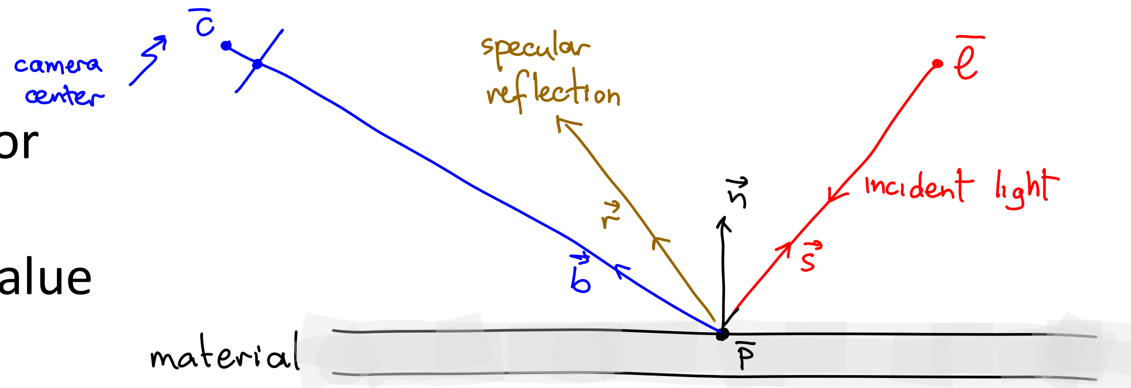
Notes

- 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



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

Gouraud Shading: Computation at Vertices



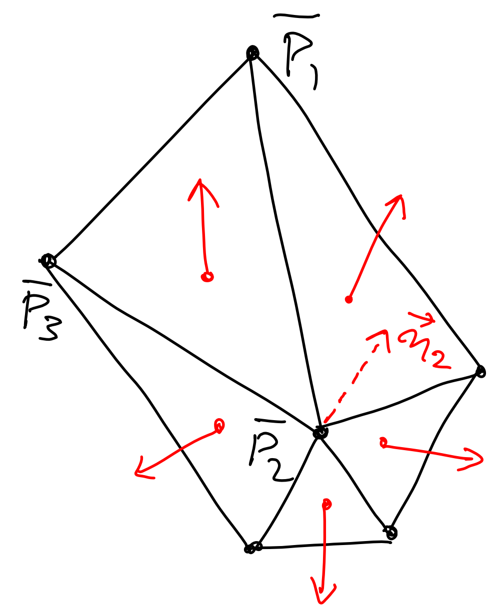
<u>Gouraud shading</u>

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

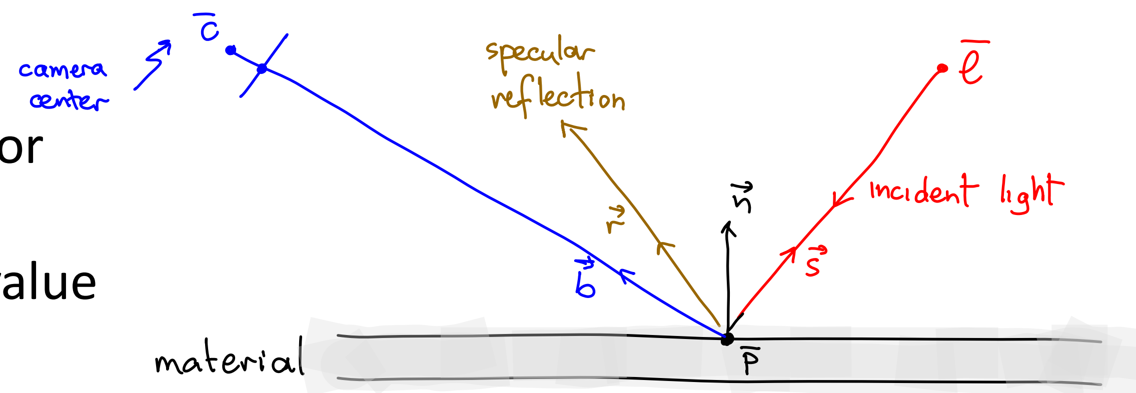
1. $\vec{n}_j$ is the average of the normals of all faces that contain vertex $\bar{p}_j$



Gouraud Shading: Computation at Vertices

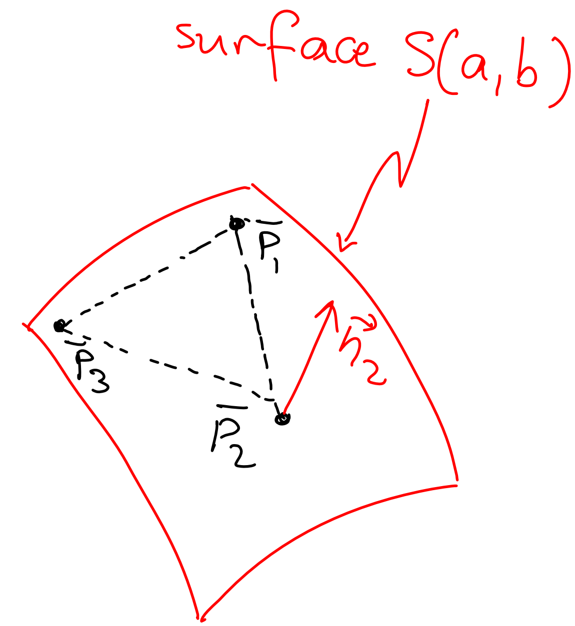
<u>Gouraud shading</u>

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

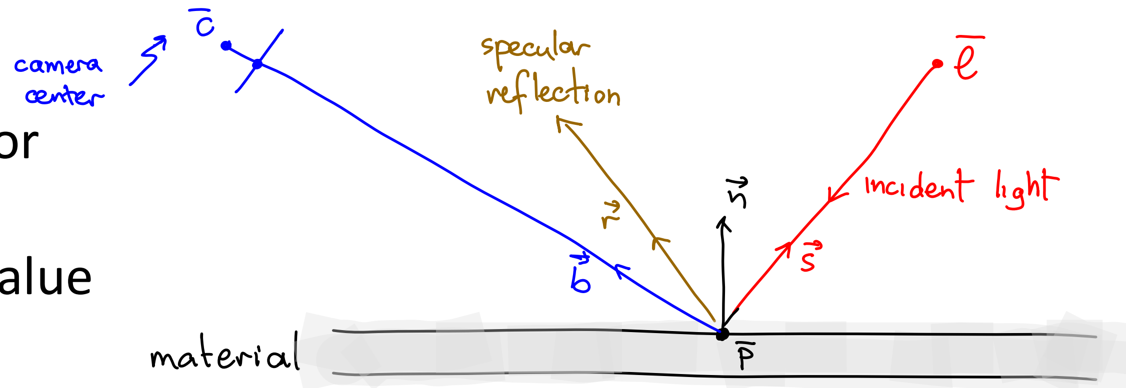


$\vec{n}_j$ is the normal of a point sample on a parametric surface computed when sampling points to create the original mesh

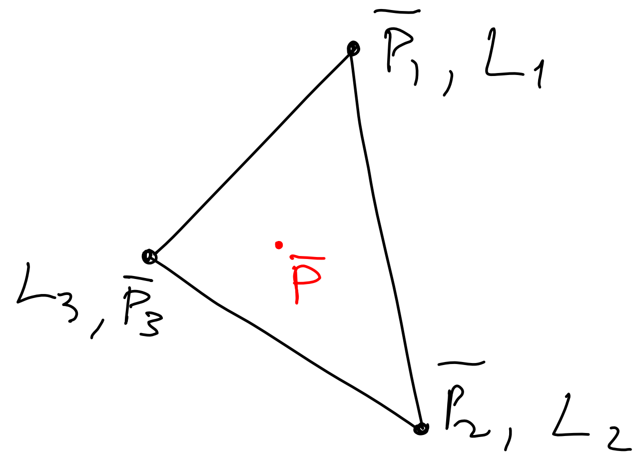
Gouraud Shading: Computation at Pixels

<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

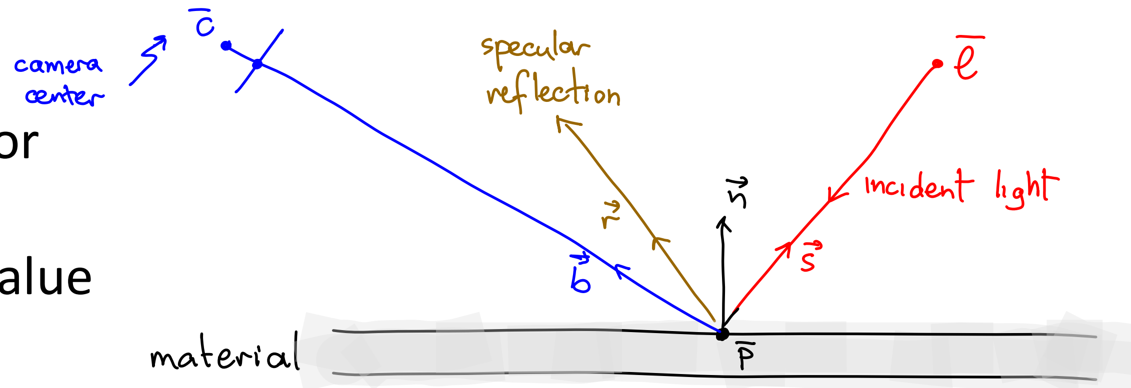


Gouraud Shading: Computation at Pixels

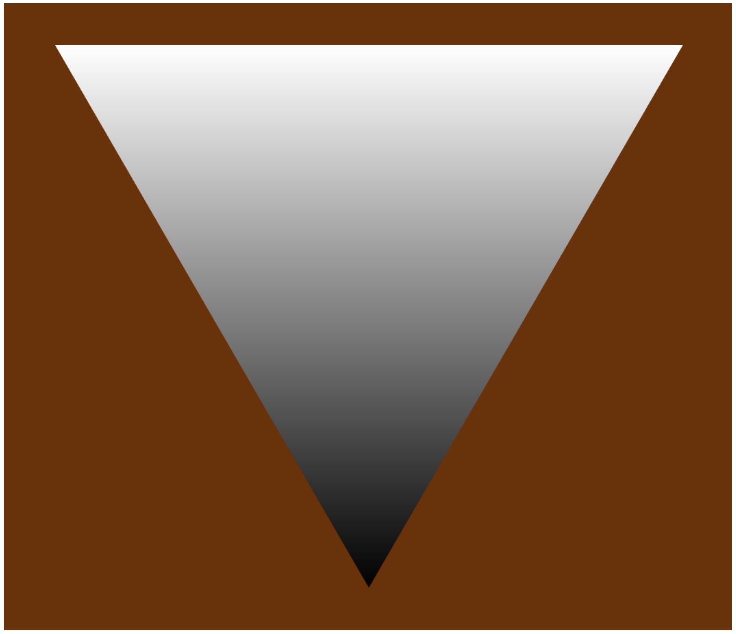
Gouraud shading

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-shaded triangle

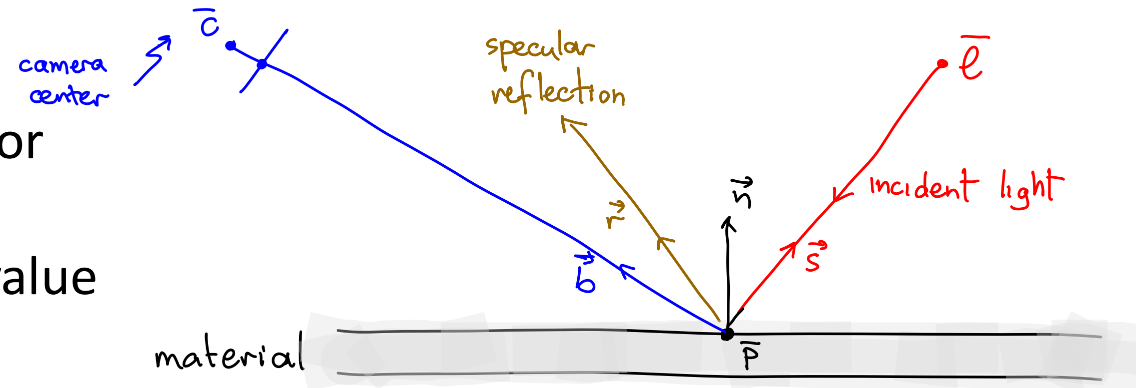


Yzmo, Wikipedia

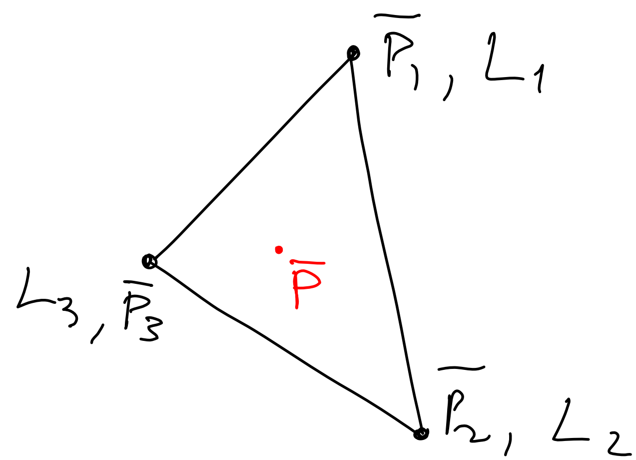
Gouraud Shading: Comparisons

<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

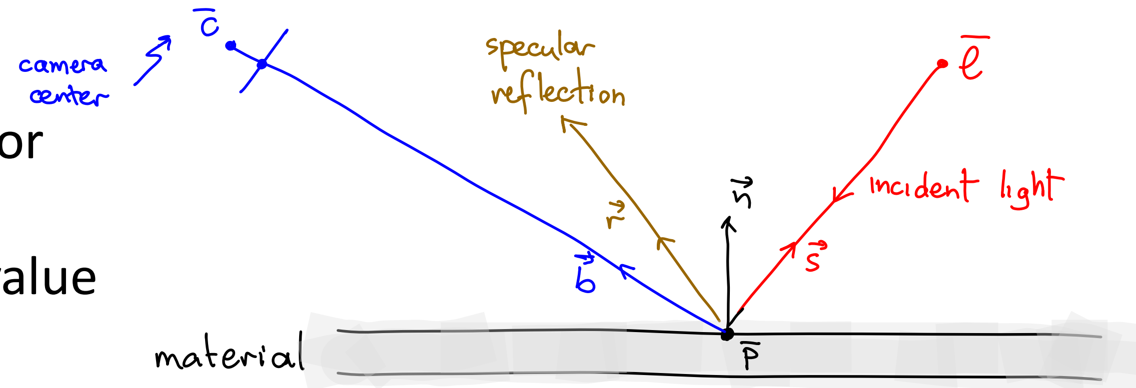
\+ Smooth, visually pleasing intensity variation that "mask" coarse geometry

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

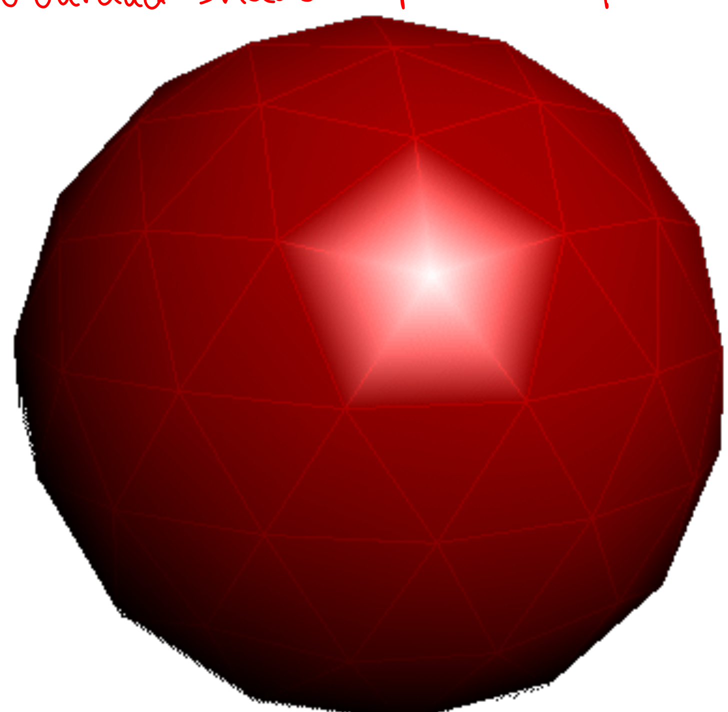
Gouraud Shading: Comparisons

<u>Gouraud shading</u>

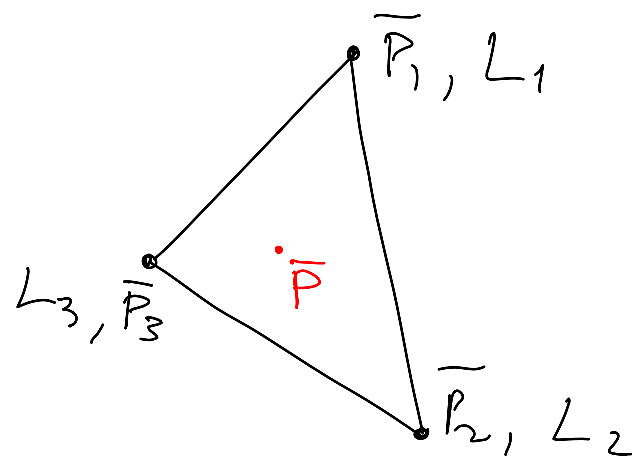
1. Compute $L_i = L(\vec{b}_i, \vec{n}_i, \vec{s}_i)$ for each vertex
2. Interpolate the L_i's to get the value at $\bar{p}$



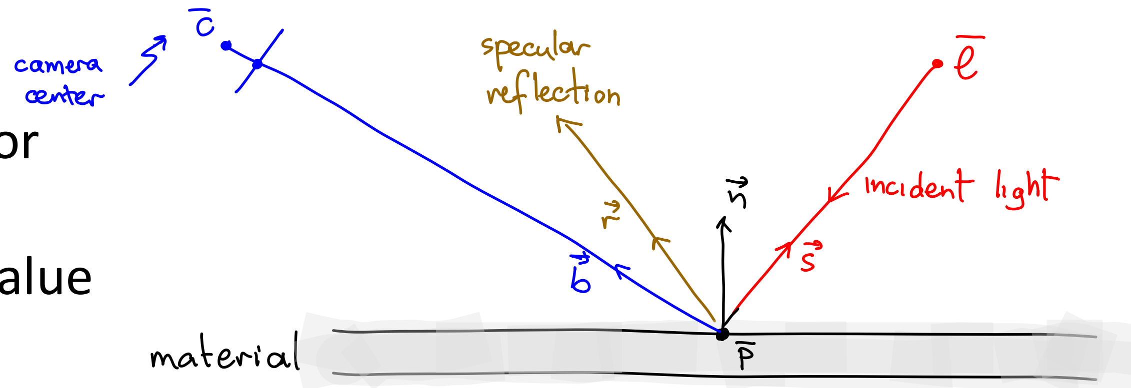
Gouraud-shaded specular sphere



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 $\bar{p}$

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

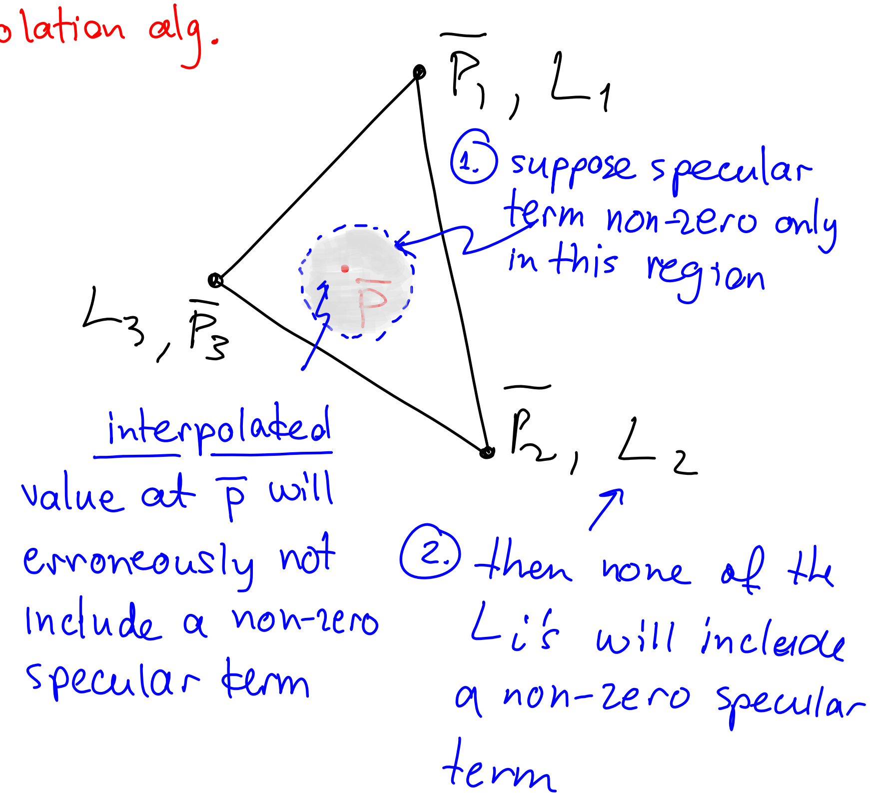
constants determined by interpolation alg.

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?)

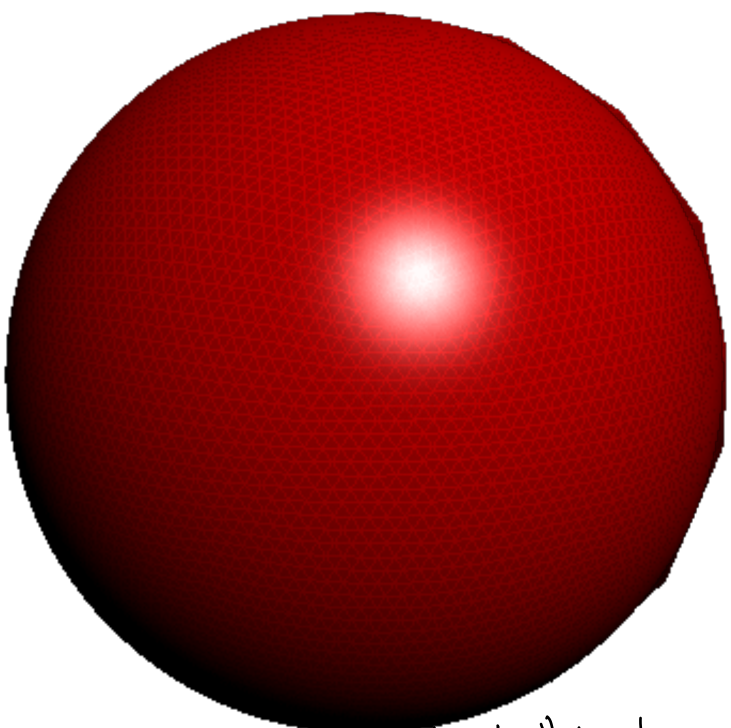


Gouraud Shading: Comparisons

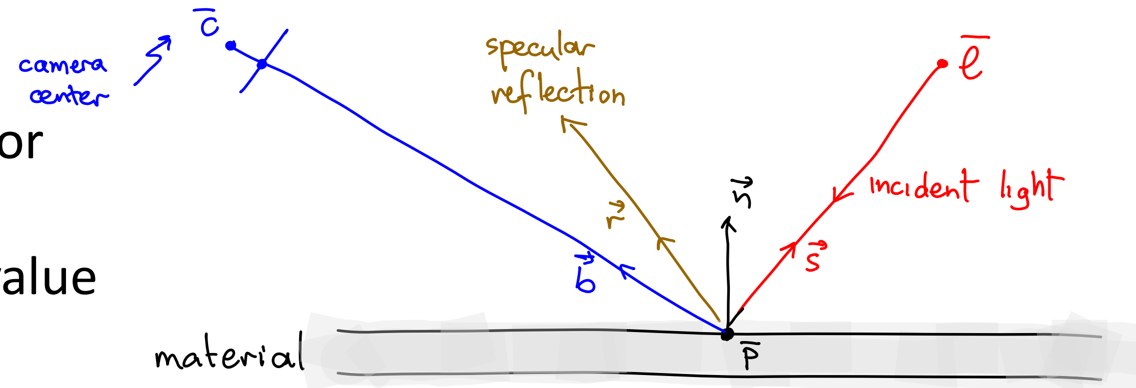
Gouraud shading

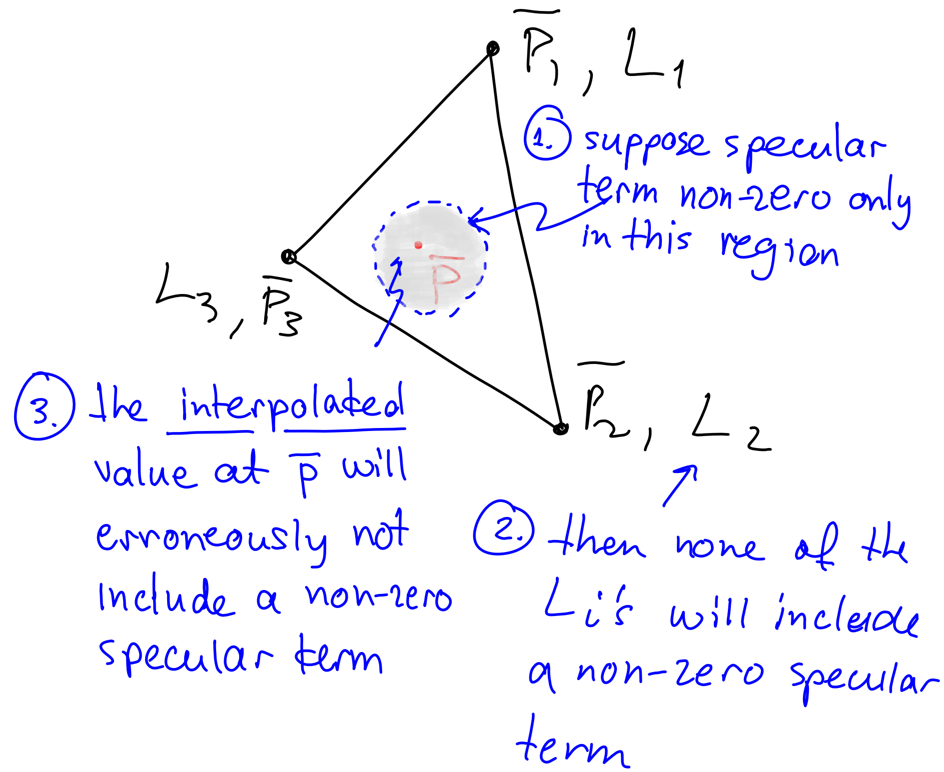
1. Compute $L_i = L(\vec{b}_i, \vec{n}_i, \vec{s}_i)$ for each vertex
2. Interpolate the L_i's to get the value at $\bar{p}$

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



Jalo, Wikipedia



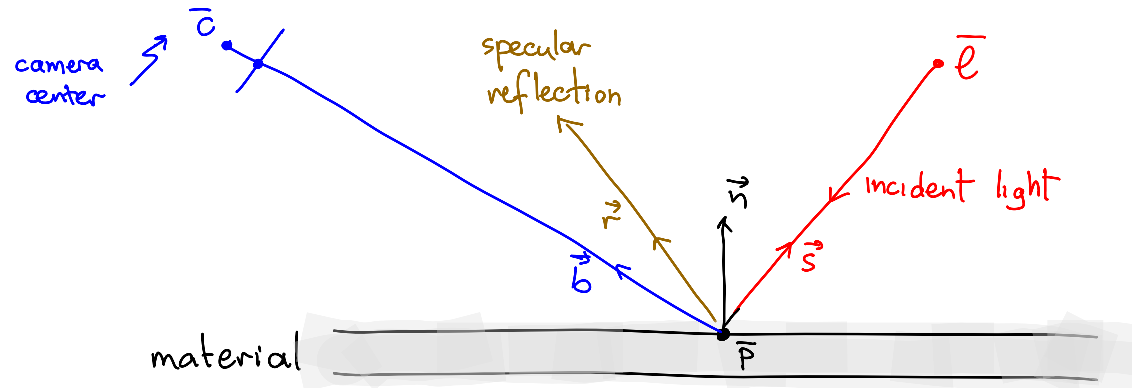


Topic 10:

Shading

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

Phong Shading: Main Idea



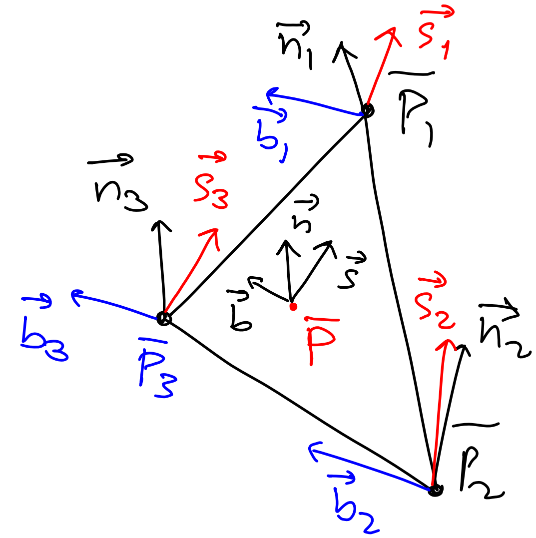
Phong shading:

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?)

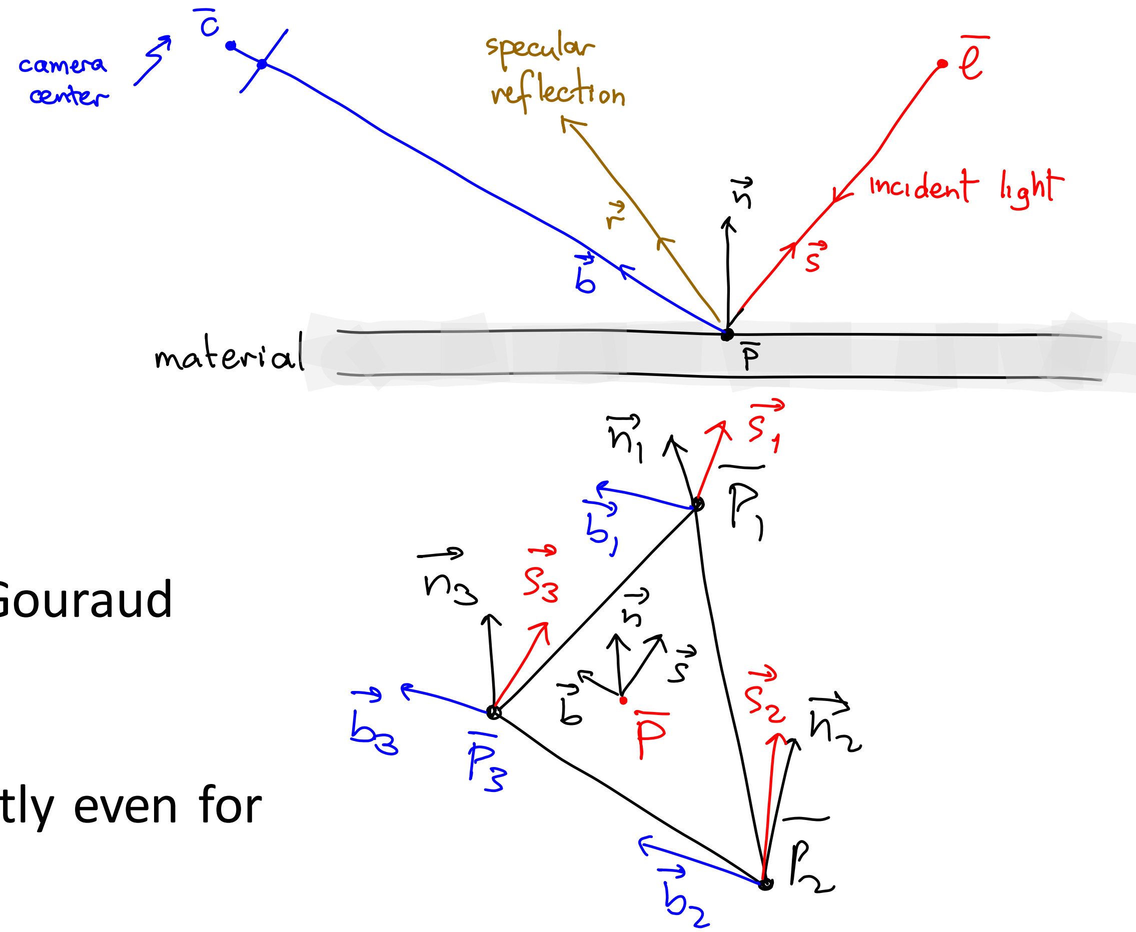


$$\boxed{\underbrace{L(\vec{b},\vec{n},\vec{s})}_{\text{intensity at projection of point } \bar{p}} = \underbrace{r_a I_a}_{\text{ambient}} + \underbrace{r_d I_d \max(0, \vec{n}\cdot\vec{s})}_{\text{diffuse}} + \underbrace{r_s I_s \max(0, \vec{r}\cdot\vec{b})^{\alpha}}_{\text{specular}}}$$

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 $L(\vec{b}, \vec{n}, \vec{s})$



<u>Comparison to Gouraud shading</u>

\+ Smooth intensity variations as in Gouraud shading

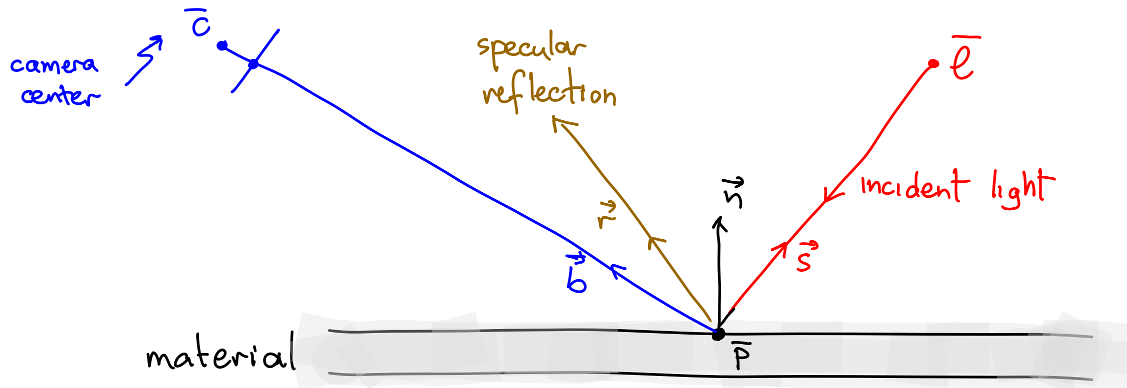
\+ Handles specular highlights correctly even for large triangles (Why?)

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

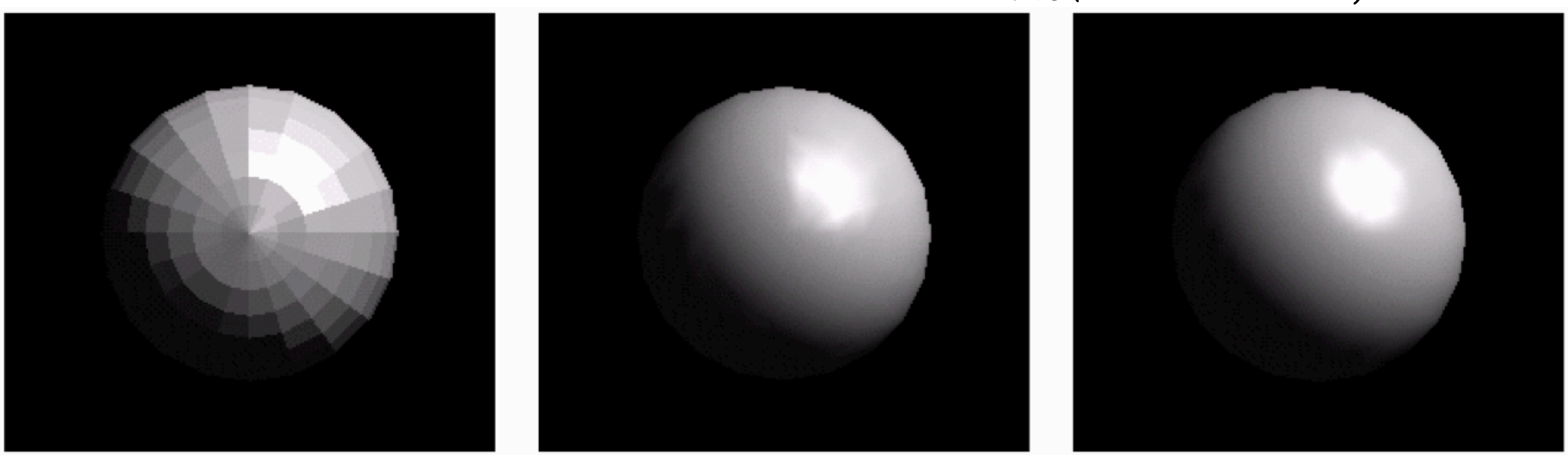
Phong Shading: Comparisons

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 $L(\vec{b}, \vec{n}, \vec{s})$



Hsien-Hsin Sean Lee, GaTech

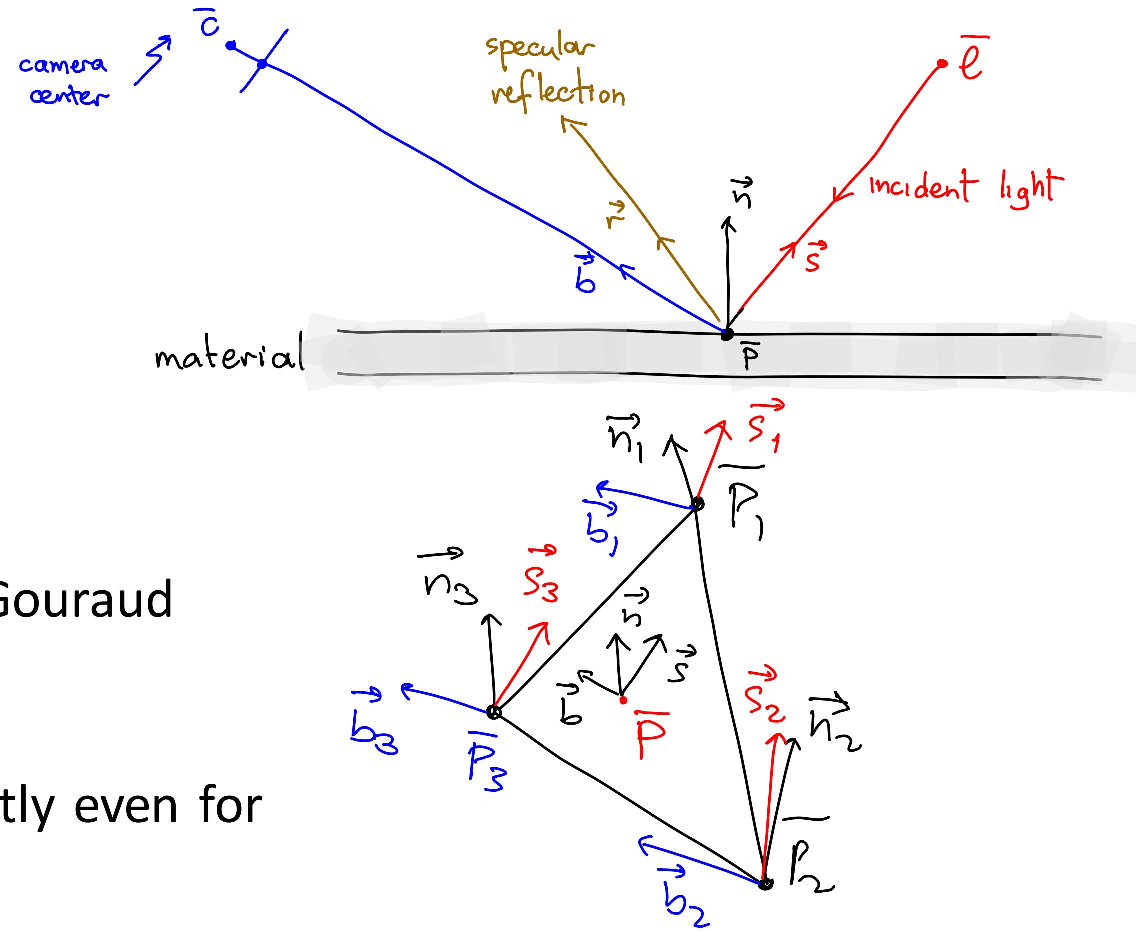


Flat shading | Gouraud shading | Phong shading

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)