Computer Graphics
Shading

Matthias Teschner

UNI FREIBURG

Outline

- Context
- Phong illumination model
- Extensions
- Shading models

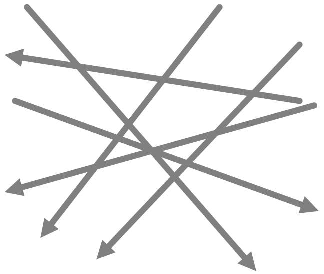
Rendering

- What is visible by the sensor?
  - Rasterization
  - Ray casting
- Which color / intensity does it have?
  - Shading
  - Evaluation of governing equations for light interaction at surfaces (rendering equation) and in participating media (volume rendering equation)
  - Local illumination models / Phong illumination model

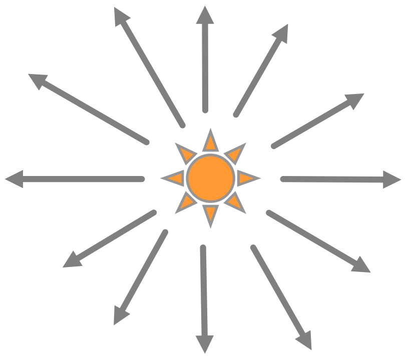
Light

- Modeled as energy parcels / photons that travel
  - Along geometric rays
  - At infinite speed
  - Radiance
- Emitted by light sources
- Scattered / absorbed at surfaces
- Scattered / absorbed by participating media
- Absorbed / measured by sensors

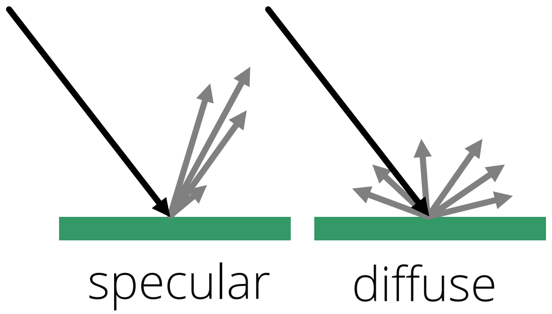
Light



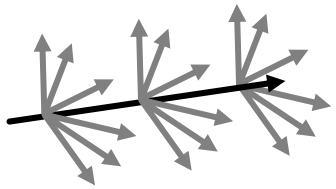
Light travels along rays



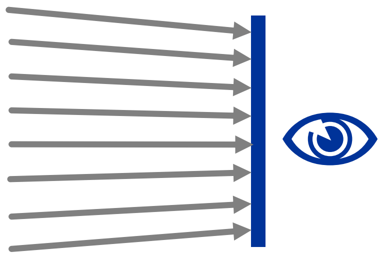
Light is generated at light sources



Incoming light is scattered and absorbed at surfaces



Participating media scatters and absorbs light

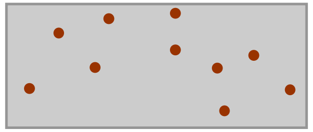


Sensors absorb light

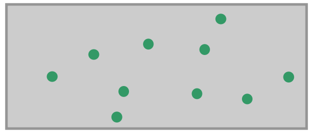
Color

– Photons are characterized by a wavelength within the visible spectrum

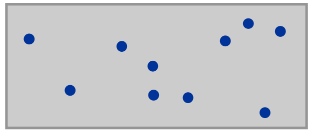
– Distribution of wavelength $\Rightarrow$ spectrum $\Rightarrow$ color



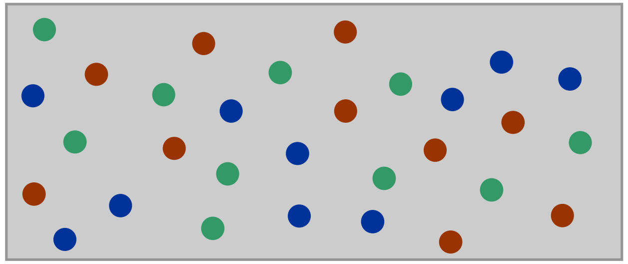
$\Phi_\lambda(\lambda_1)$



$\Phi_\lambda(\lambda_2)$



$\Phi_\lambda(\lambda_3)$



$$\begin{aligned}\Phi &= \int_{\text{VisibleSpectrum}} \Phi_\lambda(\lambda)d\lambda \\ &\approx \sum_i \Phi_\lambda(\lambda_i)\Delta\lambda_i \\ &\approx \Phi_{\text{red}}\Delta\lambda + \Phi_{\text{green}}\Delta\lambda + \Phi_{\text{blue}}\Delta\lambda\end{aligned}$$

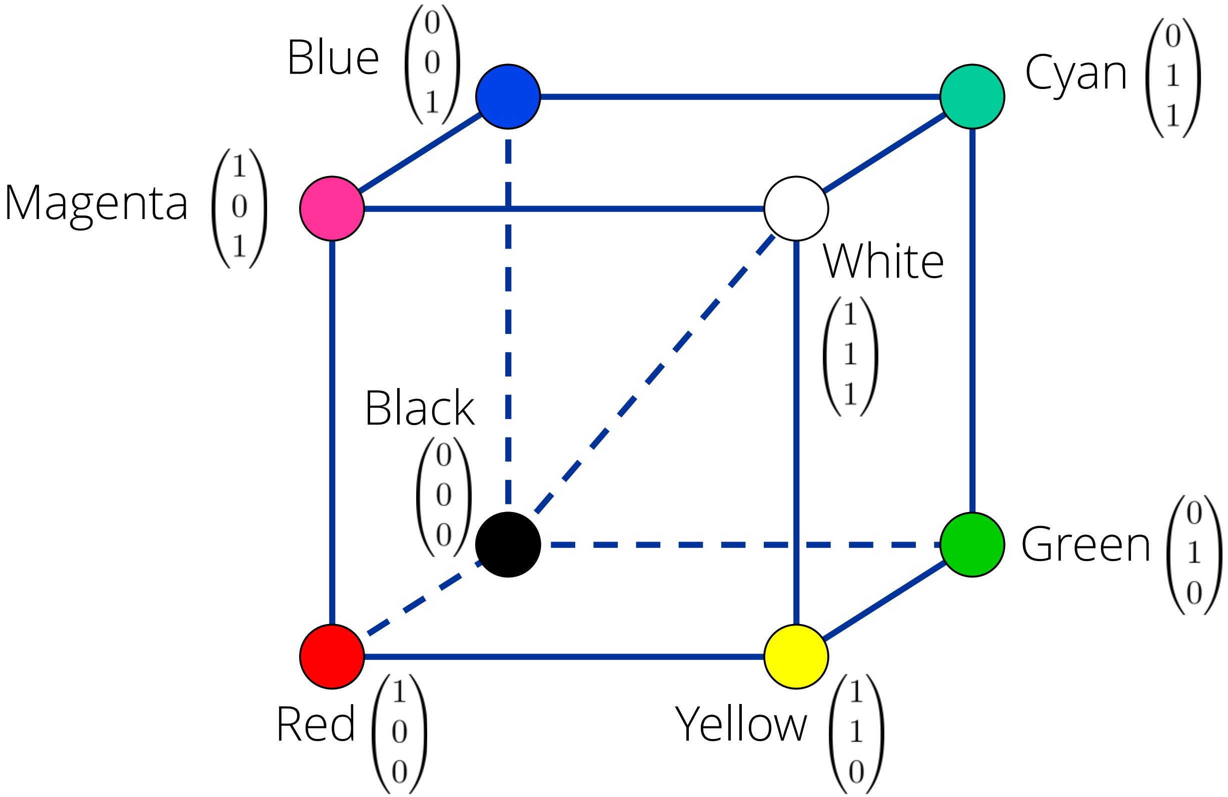
$\Phi_\lambda(\lambda)$: number of photons per time with a wavelength in a range $\Delta\lambda_i$ around λ_i.

Colored Light

- Colored light / radiance travelling along a line / ray is typically represented as a 3D vector

$$\boldsymbol{L} = \begin{pmatrix} L_{\text{red}} \\ L_{\text{green}} \\ L_{\text{blue}} \end{pmatrix}$$

- RGB color space



Colored Objects

- Surfaces are characterized by a reflectance coefficient

$$\boldsymbol{\rho} = \begin{pmatrix} \rho_{\text{red}} \\ \rho_{\text{green}} \\ \rho_{\text{blue}} \end{pmatrix}$$

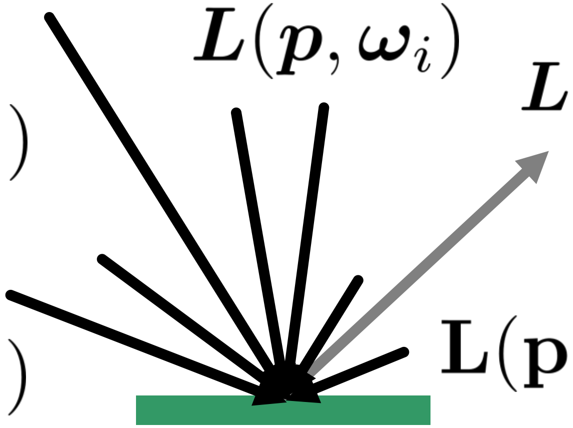
- Which components of the incoming light are reflected and which are absorbed?
- E.g., a yellow surface is described by $\boldsymbol{\rho} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$
  - Red and green are reflected
  - Blue is absorbed

Governing Equations

- Light is affected by surfaces and by participating media
- Processes described by governing equations
  - Rendering equation
  - Volume rendering equation

Rendering Equation

– Governing equation for reflected light at surfaces into a particular direction given incident light from all directions

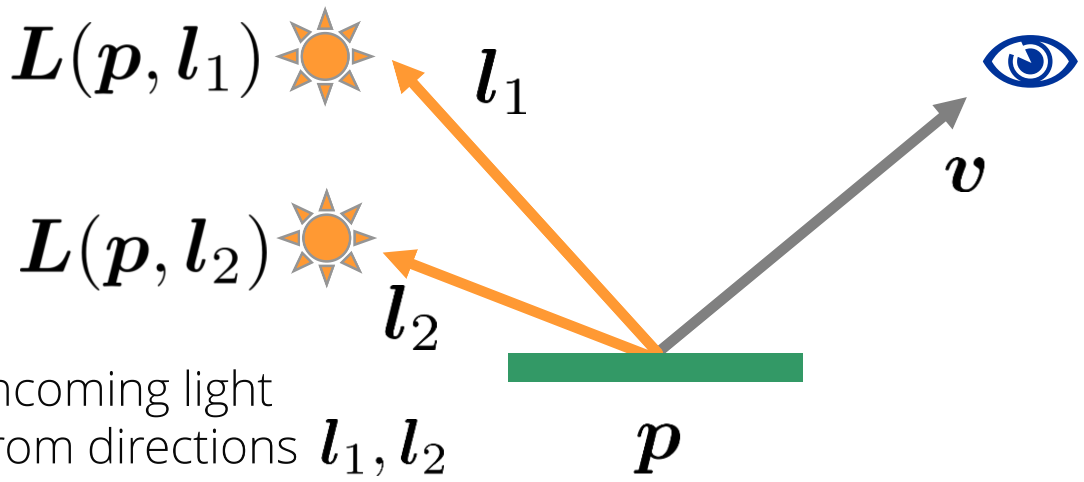


$$\boldsymbol{L}(\boldsymbol{p}, \boldsymbol{\omega}_o) = \int_\Omega \mathrm{mat}(\boldsymbol{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \boldsymbol{L}(\boldsymbol{p}, \boldsymbol{\omega}_i) \cos\theta_i \mathrm{d}\omega_i$$

Outgoing light into direction $\boldsymbol{\omega}_o$ is a sum of incident light from all directions weighted with material properties $\mathrm{mat}(\boldsymbol{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$

Local Illumination Models, e.g. Phong

- Approximately solve the rendering equation
  - Considering direct illumination from point light sources and parallel light
  - Indirect illumination from other surfaces mostly ignored



Incoming light from directions l_1, l_2 from two light sources

$$L(p, v) = \sum_i f_{\text{Phong}}\left(L(p, l_i)\right)$$

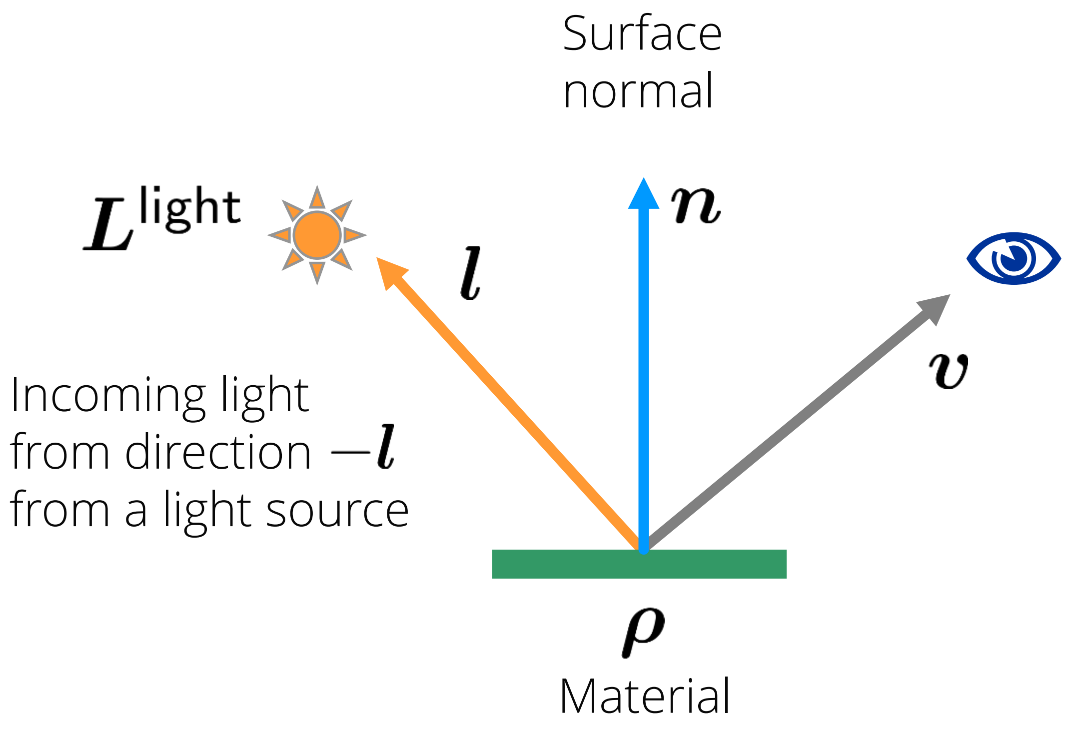
f_{Phong} computes reflected light into direction v towards the sensor considering illumination from directions l_i.

f_{Phong} is a local illumination model, e.g. the Phong illumination model.

Outline

- Context
- Phong illumination model
- Extensions
- Shading models

Setting



$$\boldsymbol{L}^{\text{cam}} = \boldsymbol{f}_{\text{Phong}}\left(\boldsymbol{L}^{\text{light}}, \boldsymbol{\rho}, \boldsymbol{l}, \boldsymbol{v}, \boldsymbol{n}, \ldots\right)$$

$\boldsymbol{f}_{\text{Phong}}$ computes reflected light into direction $\boldsymbol{v}$ towards the sensor.

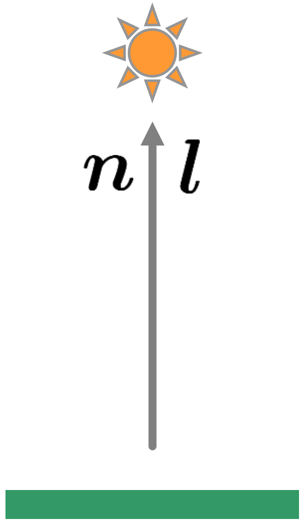
It is typically assumed that all direction vectors are normalized.

How to Compute Shading?

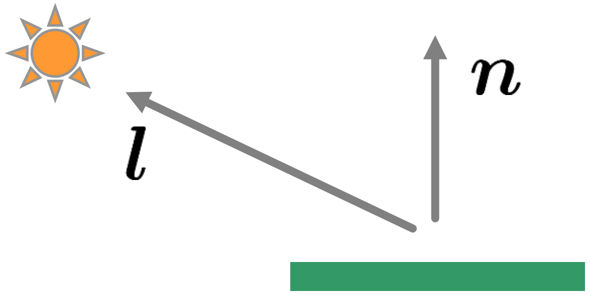
- Light $\boldsymbol{L}^{\text{light}}$ is emitted by a source (color and intensity)
- Light $\boldsymbol{L}^{\text{surf}}$ is the surface illumination caused by $\boldsymbol{L}^{\text{light}}$
  - Depends on angle between $\boldsymbol{l}$ and $\boldsymbol{n}$
- How much light $\boldsymbol{L}^{\text{refl}}$ is reflected?
  - Governed by object color $\boldsymbol{\rho}$
- Which portion $\boldsymbol{L}^{\text{cam}}$ from $\boldsymbol{L}^{\text{refl}}$ is transported towards the sensor / camera
  - Governed by materials, e.g. matte or shiny

Surface Illumination

– Angle between surface normal $\boldsymbol{n}$ and light source $\boldsymbol{l}$ direction influences the surface brightness



The same light source illuminates
a surface at different angles.

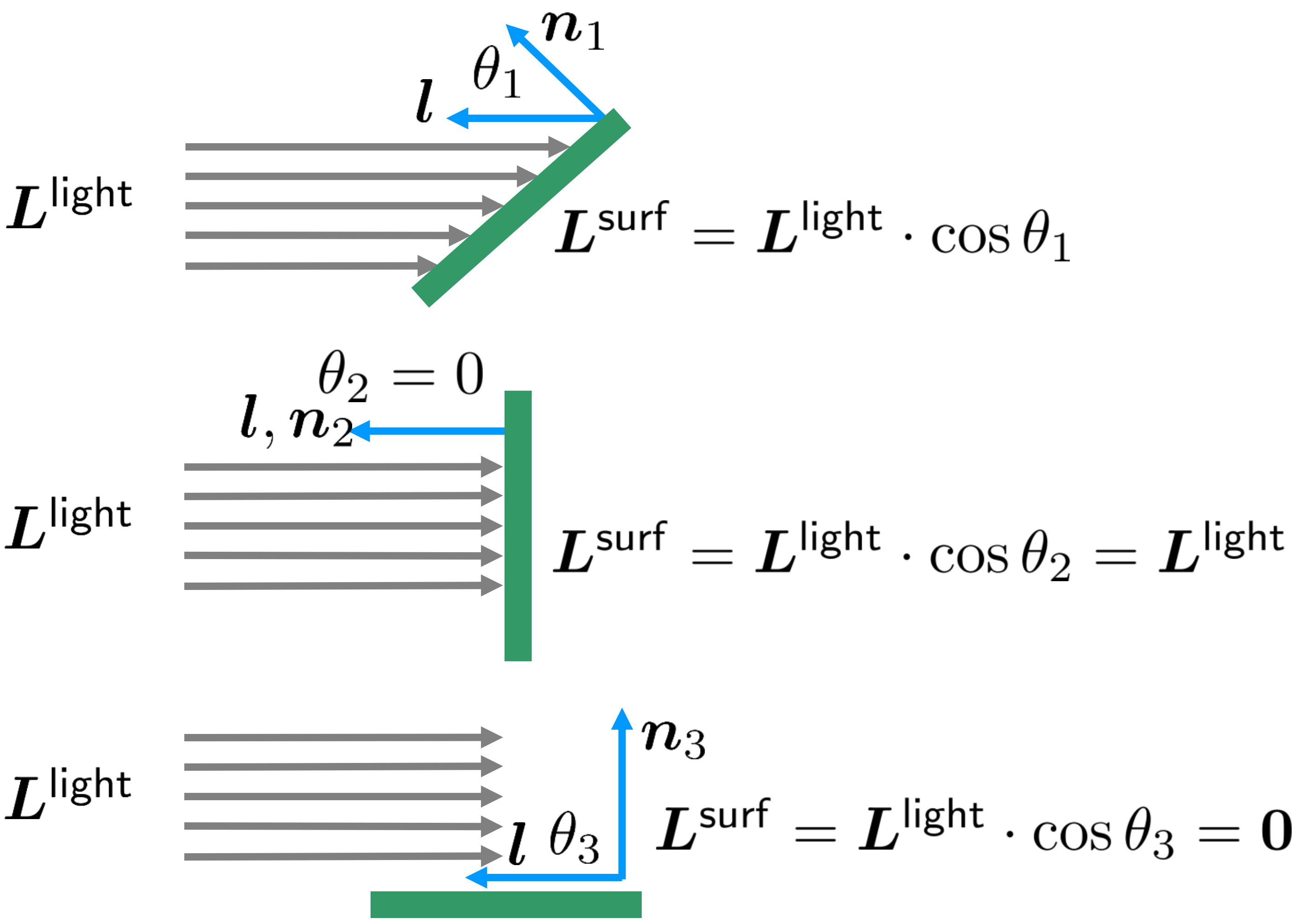


Surface receives more light
per area. Appears brighter.

Surface receives less light
per area. Appears darker.

Lambert's Cosine Law

- Illumination strength at a surface is proportional to the cosine of the angle between $\boldsymbol{l}$ and $\boldsymbol{n}$



Overall Reflected Light

- Incoming light $\boldsymbol{L}^{\text{surf}}$ at a surface patch can be reflected or absorbed
- Governed by the surface reflectance, i.e. color $\boldsymbol{\rho}$
- Overall reflected light is

$$\boldsymbol{L}^{\text{refl}} = \boldsymbol{\rho} \otimes \boldsymbol{L}^{\text{surf}} = \begin{pmatrix} \rho_{\text{red}} \cdot L^{\text{surf}}_{\text{red}} \\ \rho_{\text{green}} \cdot L^{\text{surf}}_{\text{green}} \\ \rho_{\text{blue}} \cdot L^{\text{surf}}_{\text{blue}} \end{pmatrix} = \boldsymbol{\rho} \otimes \boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l})$$

n and *l* have to be normalized. *n*·*l* has to be non-negative.

- Amount of light that leaves the surface without knowing its direction

Overall Reflected Light

– Yellow surface under white illumination

$$\boldsymbol{L}^{\text{refl}} = \boldsymbol{\rho} \otimes \boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) = \begin{pmatrix} 1 \cdot 1 \\ 1 \cdot 1 \\ 0 \cdot 1 \end{pmatrix} \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \cdot (\boldsymbol{n} \cdot \boldsymbol{l})$$

Reflects yellow light

– Yellow surface under red illumination

$$\boldsymbol{L}^{\text{refl}} = \boldsymbol{\rho} \otimes \boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) = \begin{pmatrix} 1 \cdot 1 \\ 1 \cdot 0 \\ 0 \cdot 0 \end{pmatrix} \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \cdot (\boldsymbol{n} \cdot \boldsymbol{l})$$

Reflects red light

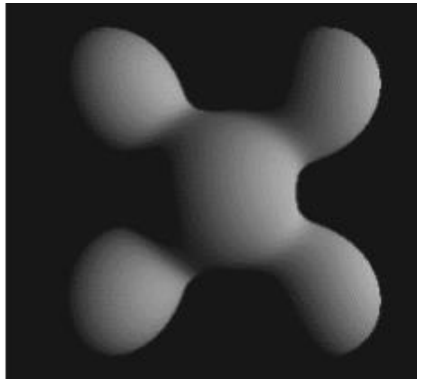
– Yellow surface under blue illumination

$$\boldsymbol{L}^{\text{refl}} = \boldsymbol{\rho} \otimes \boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) = \begin{pmatrix} 1 \cdot 0 \\ 1 \cdot 0 \\ 0 \cdot 1 \end{pmatrix} \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \cdot (\boldsymbol{n} \cdot \boldsymbol{l})$$

Does not reflect light. Blue is absorbed. Red and green could be reflected, but are not in the light.

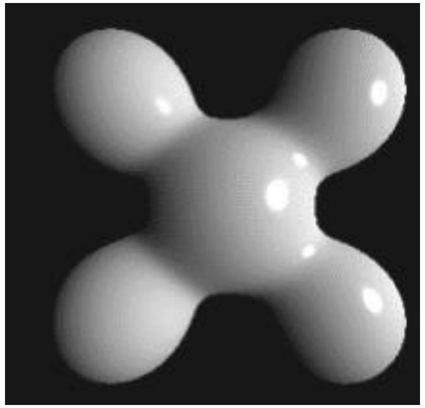
Material

- Matte
  - Diffuse reflection
  - Incident light is reflected into many different directions
- Shiny
  - Specular reflection
  - Incident light is reflected into a small set of dominant directions
  - Perceived as specular highlight



[Wikipedia: Phong Shading]

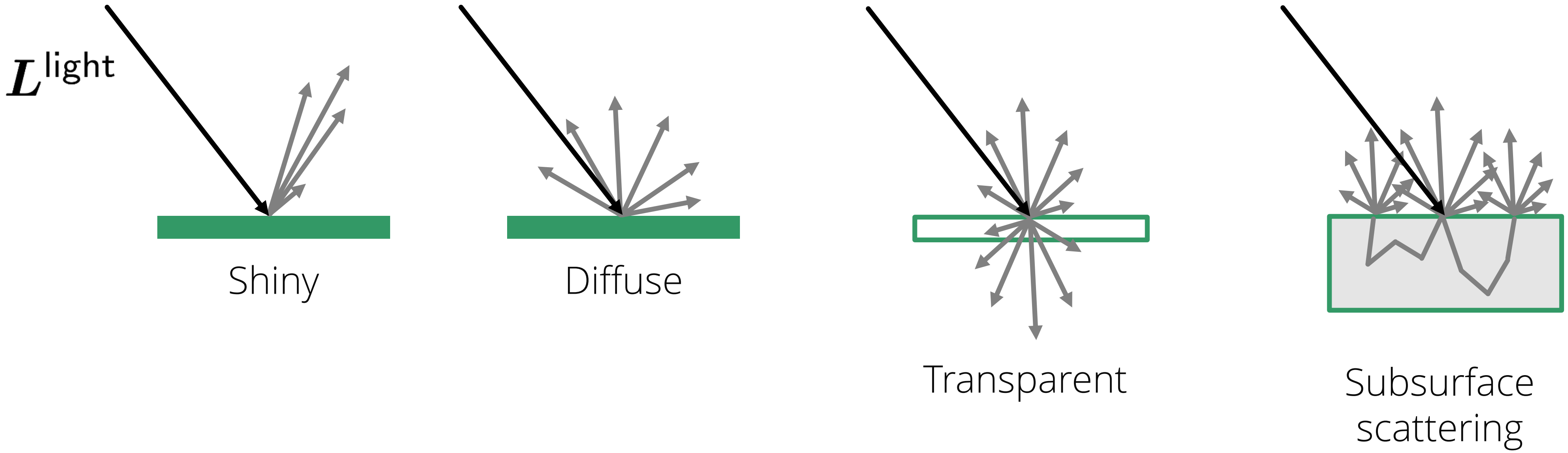
Ideal diffuse reflecting surface



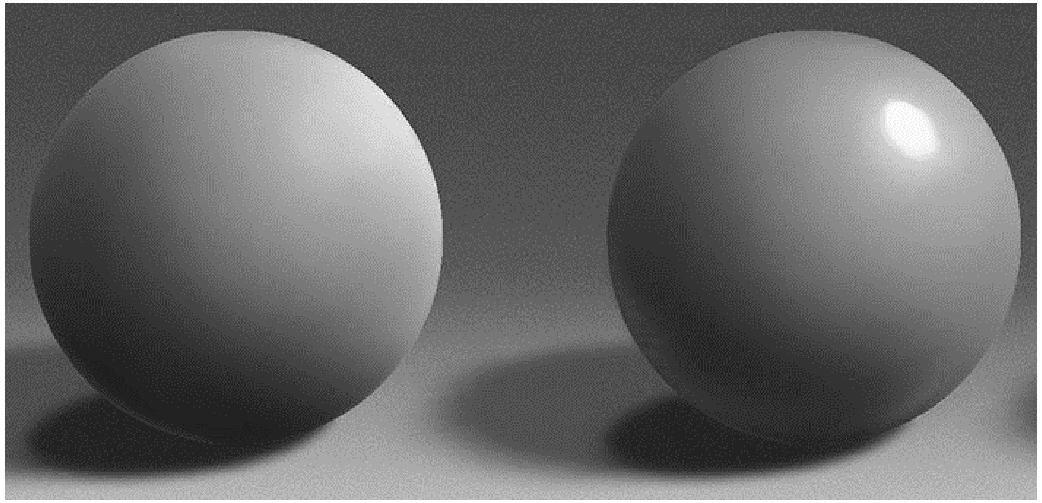
Diffuse and specular reflecting surface

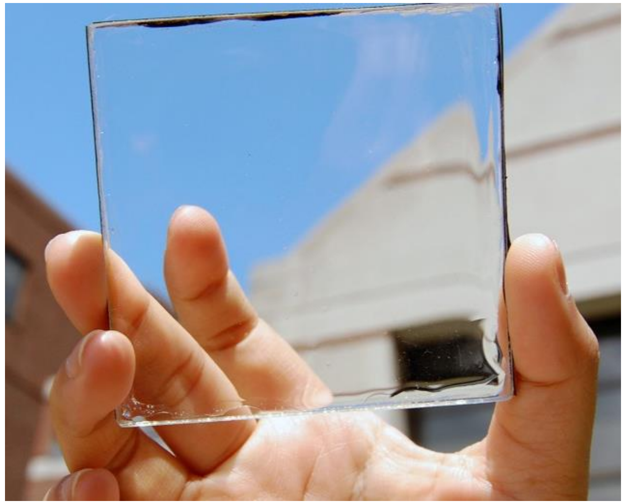
Material

– Describes how reflected light L^{refl} is distributed within the hemisphere above a surface patch



Material





Diffuse Shiny Transparent Subsurface scattering

[Oliver Wetter]

[David Turesson]

[https://cgiknowledge.wordpress.com/]

Diffuse Reflection

– Matte surfaces reflect light equally into all directions

– Light $\boldsymbol{L}^{\text{cam}}$ towards sensor

$$\boldsymbol{L}^{\text{refl}} = \int_{2\pi} \boldsymbol{L}(\boldsymbol{\omega}_o) \cos\theta_o \mathrm{d}\boldsymbol{\omega}_o$$

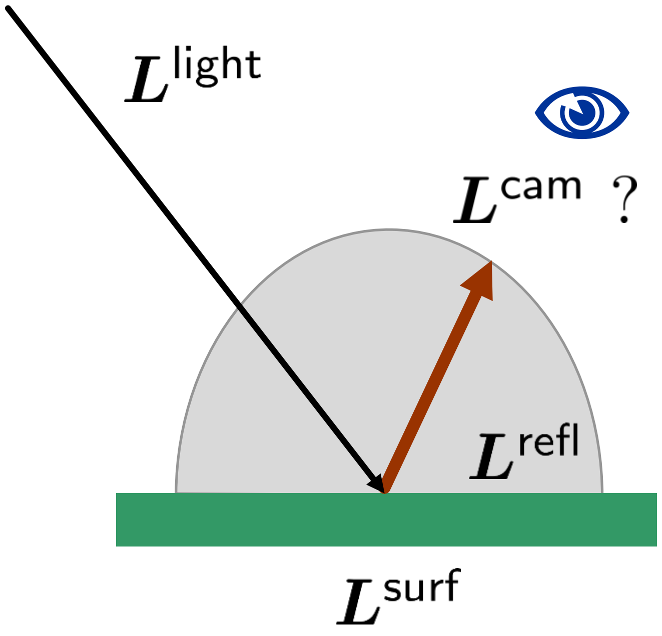
Overall reflected light equals reflected light into a direction integrated over all directions

$$\boldsymbol{L}(\boldsymbol{\omega}_o) = \text{const} = \boldsymbol{L}^{\text{cam}}$$

Definition of diffuse reflection

$$\boldsymbol{L}^{\text{refl}} = \boldsymbol{L}^{\text{cam}} \int_{2\pi} \cos\theta_o \mathrm{d}\boldsymbol{\omega}_o = \boldsymbol{L}^{\text{cam}} \cdot \pi$$

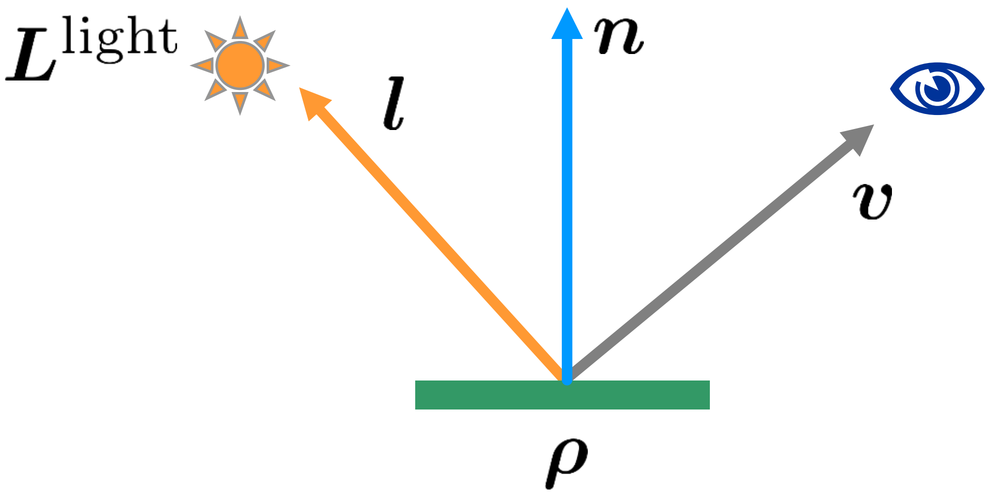
$$\boldsymbol{L}^{\text{cam}} = \frac{1}{\pi}\boldsymbol{L}^{\text{refl}} = \frac{1}{\pi} \cdot \boldsymbol{\rho} \otimes \boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l})$$



The cosine term in the integral is related to Lambert's cosine law.
More insights in Advanced Computer Graphics.

Diffuse Reflection

– Reflected light from a matte surface according to the Phong illumination model



$$\boldsymbol{L}^{\text{cam}} = \frac{1}{\pi} \cdot \boldsymbol{\rho} \otimes \boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l})$$

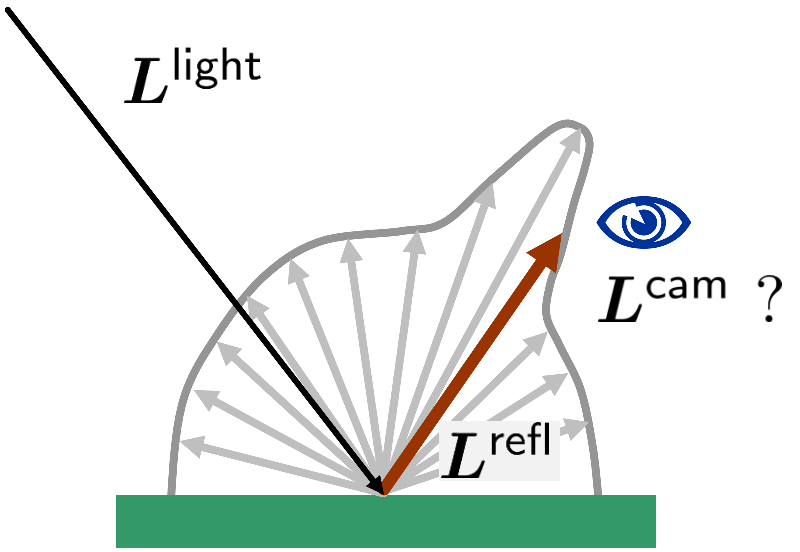
Diffuse Reflection - Discussion

- Light from light source $\boldsymbol{L}^{\text{light}}$
- Illumination at the surface $\boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l})$
- Overall reflected light $\boldsymbol{\rho} \otimes \boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l})$
- Reflected light towards viewer $\frac{1}{\pi} \cdot \boldsymbol{\rho} \otimes \boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l})$
- View-independent
  - Same reflection into all directions
  - Computation does not require $\boldsymbol{v}$
  - If the view changes, the reflected light does not change

Specular Reflection

– Shiny surfaces reflect light into a small set of dominant directions

– Light $\boldsymbol{L}^{\text{cam}}$ towards sensor



$$\boldsymbol{L}^{\text{refl}} = \int_{2\pi} \boldsymbol{L}(\boldsymbol{\omega}_o) \cos\theta_o \mathrm{d}\boldsymbol{\omega}_o$$

Overall reflected light equals reflected light into a direction integrated over all directions

$$\boldsymbol{L}(\boldsymbol{\omega}_o) \sim (\boldsymbol{r} \cdot \boldsymbol{\omega}_o)^m$$

Definition of specular reflection. **r** is the reflection vector of light direction **l** with respect to normal **n**.

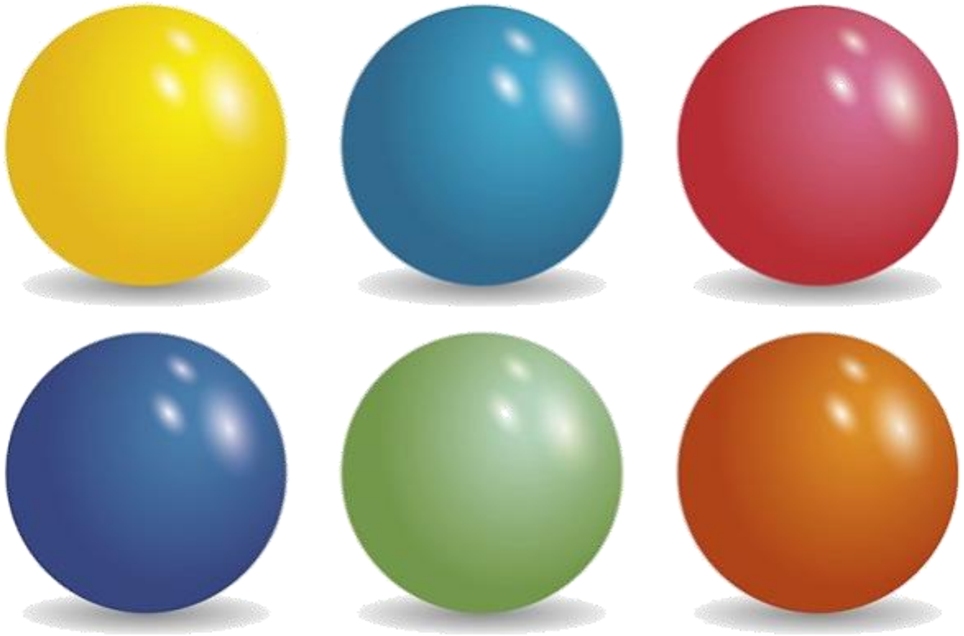
$$\boldsymbol{L}^{\text{refl}} = \int_{2\pi} k(\boldsymbol{r} \cdot \boldsymbol{\omega}_o)^m \cos\theta_o \mathrm{d}\boldsymbol{\omega}_o$$

$$\boldsymbol{L}^{\text{cam}} = \boldsymbol{\rho}^{\text{white}} \otimes \boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) \cdot (\boldsymbol{r} \cdot \boldsymbol{v})^m$$

k is not analyzed in Phong's model. White surface color accounts for the fact that shiny surfaces reflect the entire light spectrum.

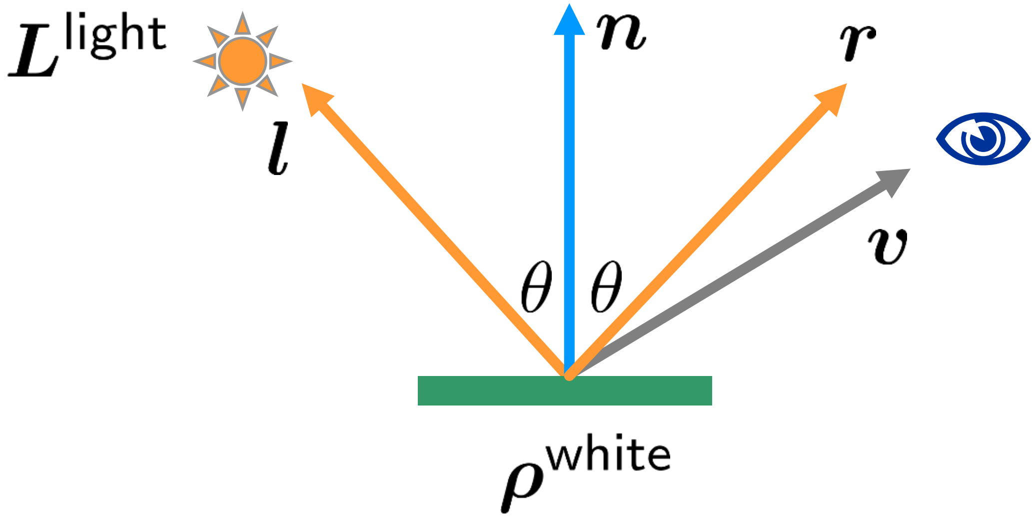
Example

– Shiny surfaces reflect all color components of the incoming light independent from the surface color



Specular Reflection

– Reflected light from a shiny surface according to the Phong illumination model



$$\boldsymbol{L}^{\text{cam}} = \boldsymbol{\rho}^{\text{white}} \otimes \boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) \cdot (\boldsymbol{r} \cdot \boldsymbol{v})^{m}$$

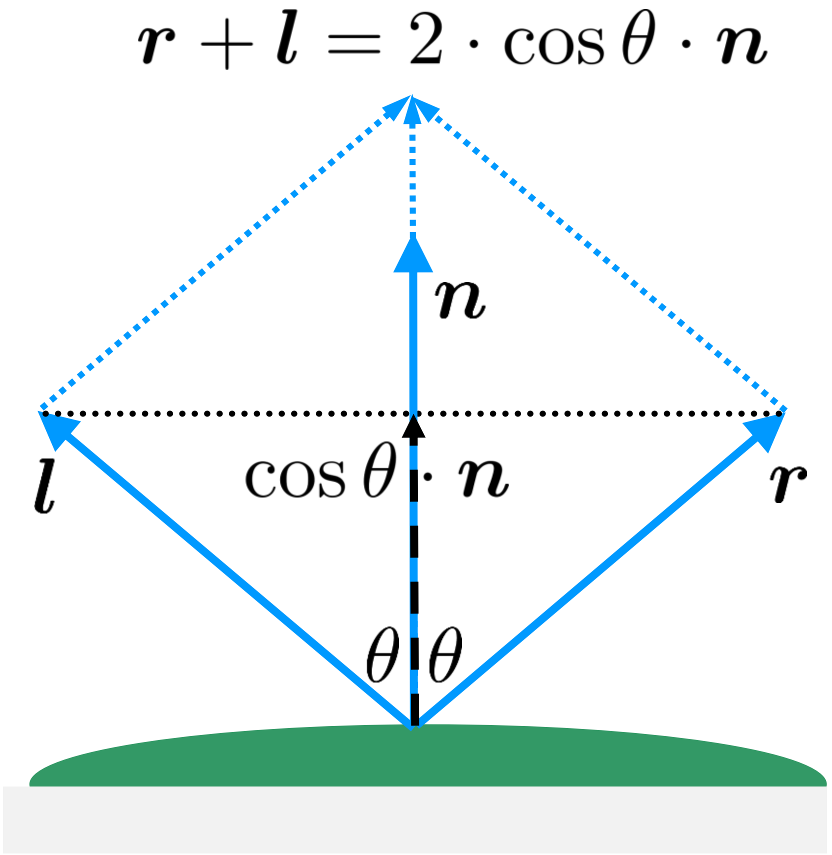
Reflected light is maximal, if the viewer direction equals the reflection direction of the illumination. *m* governs the size of the shiny area. The color of highlights converges to the color of the light source. That's why, the surface reflectance should not change the color of incoming light.

Reflection Vector

- Computed with light source direction $\boldsymbol{l}$ and surface normal $\boldsymbol{n}$

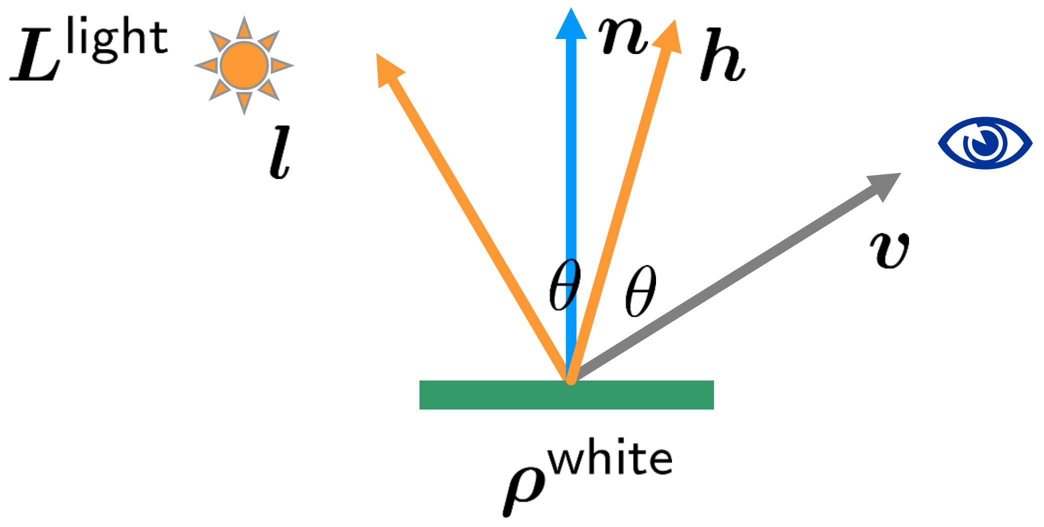
$$\boldsymbol{r} + \boldsymbol{l} = 2 \cdot \cos\theta \cdot \boldsymbol{n}$$
$$\cos\theta = \boldsymbol{l} \cdot \boldsymbol{n}$$
$$\boldsymbol{r} = 2 \cdot (\boldsymbol{l} \cdot \boldsymbol{n}) \cdot \boldsymbol{n} - \boldsymbol{l}$$

- Vectors $\boldsymbol{l}$ and $\boldsymbol{n}$ have to be normalized
- Vector $\boldsymbol{r}$ is normalized



Specular Reflection

– Reflected light from a shiny surface according to the Blinn-Phong illumination model



$$\boldsymbol{L}^{\text{cam}} = \boldsymbol{\rho}^{\text{white}} \otimes \boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) \cdot (\boldsymbol{n} \cdot \boldsymbol{h})^m$$

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

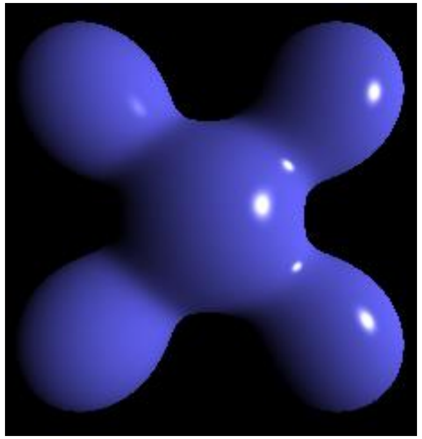
No considerable difference to Phong.

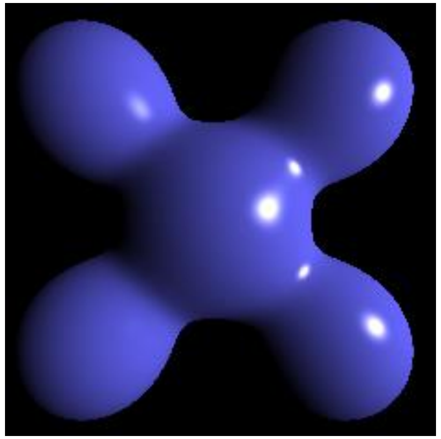
Specular Reflection - Discussion

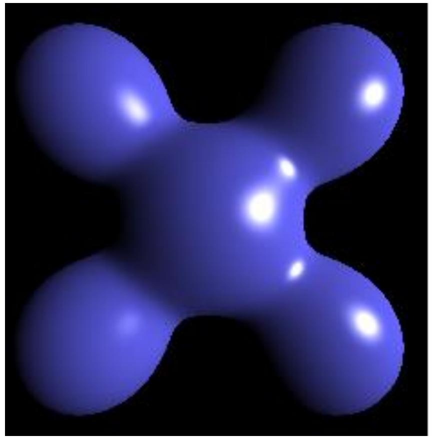
- Light from light source $\boldsymbol{L}^{\text{light}}$
- Illumination at the surface $\boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l})$
- Overall reflected light $\boldsymbol{\rho}^{\text{white}} \otimes \boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l})$
- Reflected light towards viewer $\boldsymbol{\rho}^{\text{white}} \otimes \boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) \cdot (\boldsymbol{r} \cdot \boldsymbol{v})^m$
- Models specular highlights on shiny surfaces

Specular Reflection - Discussion

– Maximal, if viewer and reflection direction coincide
– Entire light spectrum is reflected
– Color converges to light source color


Phong.


Blinn-Phong. Larger *m*.


Blinn-Phong. Smaller *m*.

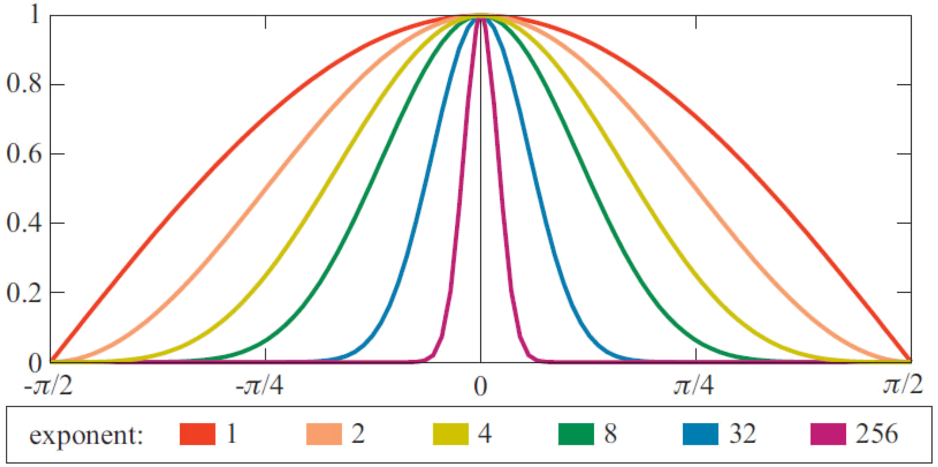
Exponent *m* governs the size of the highlight area. *M* does not influence the maximal intensity.

[Wikipedia: Blinn-Phong shading model]

Specular Reflection - Discussion

- Phong and Blinn-Phong do not account for energy preservation
- Reflected light depends on angle θ and exponent m
- Overall reflected light depends on m

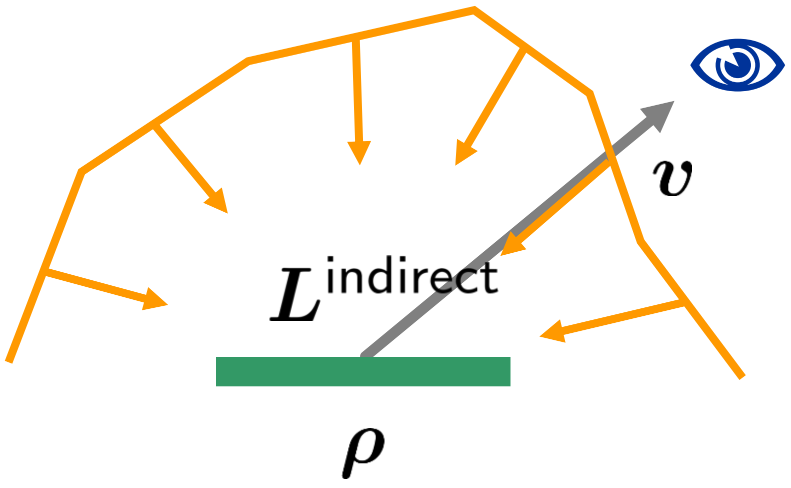
$\cos^m \theta$ [Akenine-Möller et al.]



Angle θ between **v** and **r** (Phong) or **n** and **h** (Blinn-Phong)

Reflection From Ambient Illumination

- Accounts for indirect illumination from other surfaces
- Indirect illumination at the surface $\boldsymbol{L}^{\text{indirect}}$
- Overall reflected light $\boldsymbol{\rho} \otimes \boldsymbol{L}^{\text{indirect}}$
- Diffuse reflection towards viewer $\boldsymbol{L}^{\text{cam}} = \frac{1}{\pi} \cdot \boldsymbol{\rho} \otimes \boldsymbol{L}^{\text{indirect}}$



Ambient Reflection - Discussion

- Appropriate if surfaces illuminate each other
- E.g., red cube in an illuminated room with yellow walls:

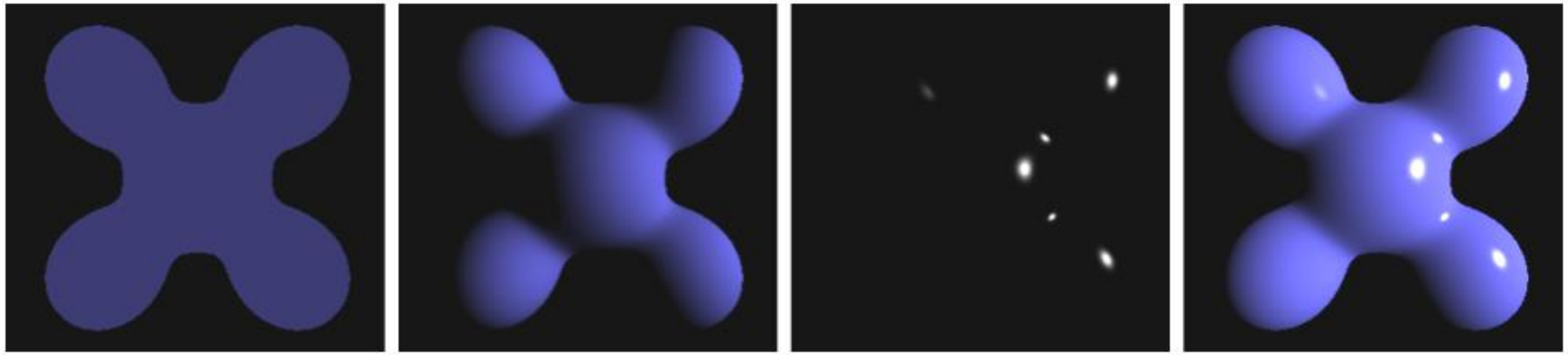
$$\boldsymbol{L}^{\text{indirect}} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \qquad \boldsymbol{\rho}^{\text{cube}} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\boldsymbol{L}^{\text{cam}} = \frac{1}{\pi} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \frac{1}{\pi} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Phong Illumination Model

– Combination of ambient, diffuse and specular reflection

[Wikipedia: Blinn-Phong shading model]



Ambient Reflection	Diffuse reflection	Specular reflection	Phong model

Ambient Reflection

(not physically motivated in a black environment)

Diffuse reflection

Specular reflection

Phong model

Phong Illumination Model

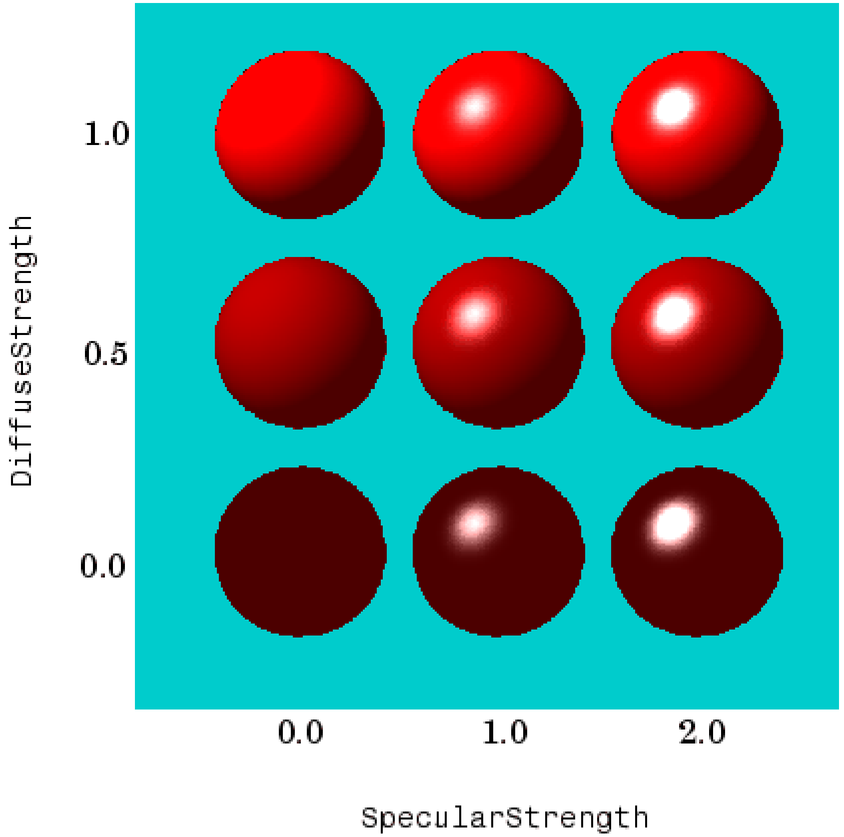
$$\boldsymbol{L}^{\text{cam}} = \frac{1}{\pi} \cdot \boldsymbol{\rho} \otimes \boldsymbol{L}^{\text{indirect}} + \sum_i \boldsymbol{L}_i^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l}_i) \otimes \left(\frac{1}{\pi} \cdot \boldsymbol{\rho} + \boldsymbol{\rho}^{\text{white}} \cdot (\boldsymbol{r}_i \cdot \boldsymbol{v})^m\right)$$

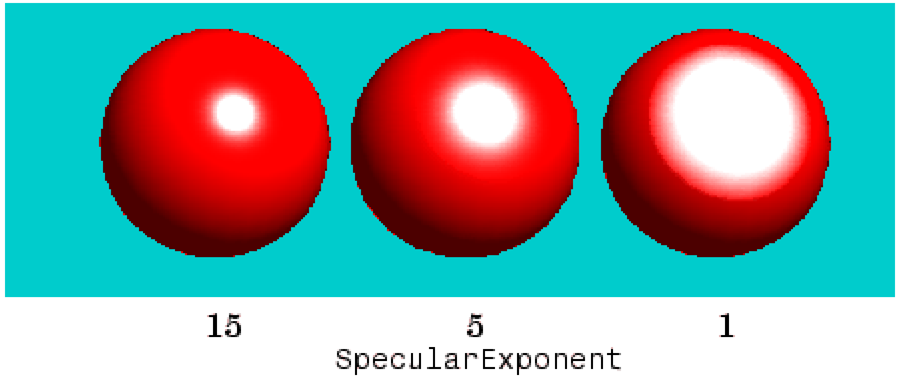
Ambient reflection | Multiple light sources | Surface illumination | Diffuse reflection | Specular reflection

– As the specular term does not account for energy preservation, ambient, diffuse and specular reflection are weighted by user-defined scalar coefficients α, β, γ

$$\boldsymbol{L}^{\text{cam}} = \alpha \cdot \boldsymbol{\rho} \otimes \boldsymbol{L}^{\text{indirect}} + \sum_i \boldsymbol{L}_i^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l}_i) \otimes \left(\beta \cdot \boldsymbol{\rho} + \gamma \cdot \boldsymbol{\rho}^{\text{white}} \cdot (\boldsymbol{r}_i \cdot \boldsymbol{v})^m\right)$$

Phong Illumination Model





[http://www.mathworks.com/help/techdoc/visualize/f1-21818.html]

Phong Illumination Model - Variants

– Physical motivations are sometimes weakened, e.g. by introducing separate illuminations and reflectance coefficients for ambient, diffuse and specular reflection, e.g. $\boldsymbol{L}^{\text{cam}} = \boldsymbol{k}^{\text{amb}} \otimes \boldsymbol{L}^{\text{amb}} + \sum_i \boldsymbol{L}_i^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l}_i) \otimes \left(\boldsymbol{k}^{\text{diff}} + \boldsymbol{k}^{\text{spec}} \cdot (\boldsymbol{r}_i \cdot \boldsymbol{v})^m\right)$
or $\boldsymbol{L}^{\text{cam}} = \boldsymbol{k}^{\text{amb}} \otimes \boldsymbol{L}^{\text{amb}} + \boldsymbol{k}^{\text{diff}} \otimes \boldsymbol{L}^{\text{diff}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) + \boldsymbol{k}^{\text{spec}} \otimes \boldsymbol{L}^{\text{spec}} \cdot (\boldsymbol{r} \cdot \boldsymbol{v})^m$

Weighting coefficient are incorporated into the *k*-values. *K*-values encode the reflectance and the scaling coefficients. Such variants compute some color, but depart from physical motivations.

Phong Illumination Model - Discussion

- Considers reflections from matte and shiny surfaces due to direct illumination
  - Diffuse and specular reflection
- Considers reflection from matte surfaces due to indirect illumination
  - Ambient reflection

Phong Illumination Model - Discussion

- Physically motivated
- Approximate
- Limited to opaque surfaces
- Efficient local computation using
  - Light direction, camera direction, surface normal, surface color, light color

Phong Illumination Model - Discussion

- Resulting images tend to look less realistic
  1. Realistic scenes have complex illuminations
     - Area light sources and dominant indirect illumination would have to be represented with numerous point light sources in the computation
  2. Realistic scenes have complex materials
     - Spatially varying reflectance values would have to be modeled
  3. Non-physical Phong parameters cause issues

Phong Illumination Model - Derivation

- This slide set focuses on the general ideas with simplified derivations
- For physical quantities that characterize light, see Advanced Computer Graphics
  - Flux
  - Irradiance (e.g. illumination of a surface)
  - Radiosity (e.g. overall light that leaves a surface)
  - Radiance (e.g. light transported along rays, light that arrives at a sensor element)

Outline

- Context
- Phong illumination model
- Extensions
- Shading models

Considering Distances

- Between object surface and light source
  - Surface illumination is inversely proportional to the squared distance between surface and light source
  - Light source attenuation
- Between object surface and viewer
  - Volumetric effects, e.g. fog, influence the light transport
    - If air is transparent, objects are clearly visible
    - In less transparent air, fog particles absorb some light and scatter additional light towards the viewer
    - In low visibility, light at the sensor converges to a "fog color"

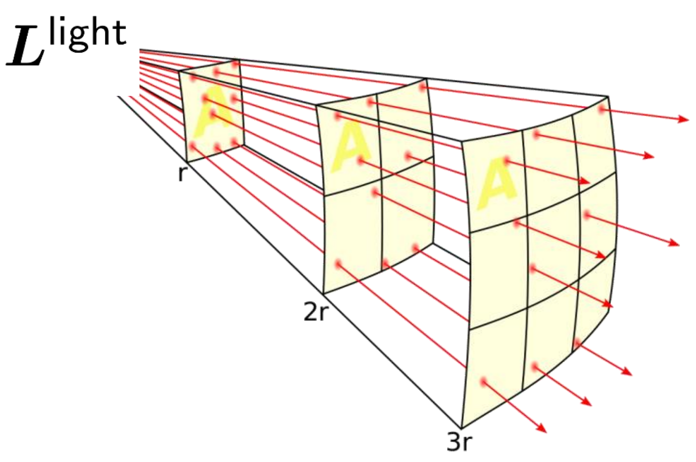
Light Source Attenuation

- Inverse Square Law
  - Illumination of a surface decreases quadratically with the distance from a light source
  - Light position $\boldsymbol{p}^{\text{light}}$, surface $\boldsymbol{p}$, distance $r = \|\boldsymbol{p}^{\text{light}} - \boldsymbol{p}\|$
  - Illumination at surface

    $$\boldsymbol{L}^{\text{surf}} = \frac{1}{r^2} \cdot \boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l})$$
  - Variant

    $$\boldsymbol{L}^{\text{surf}} = \frac{1}{k_c + k_l \cdot r + k_q \cdot r^2} \cdot \boldsymbol{L}^{\text{light}} \cdot (\boldsymbol{n} \cdot \boldsymbol{l})$$

[Wikipedia: Inverse Square Law]



Same light at all surfaces whose area grows quadratically with distance *r*. Therefore, illumination at the surfaces decreases quadratically with distance *r*.

Fog

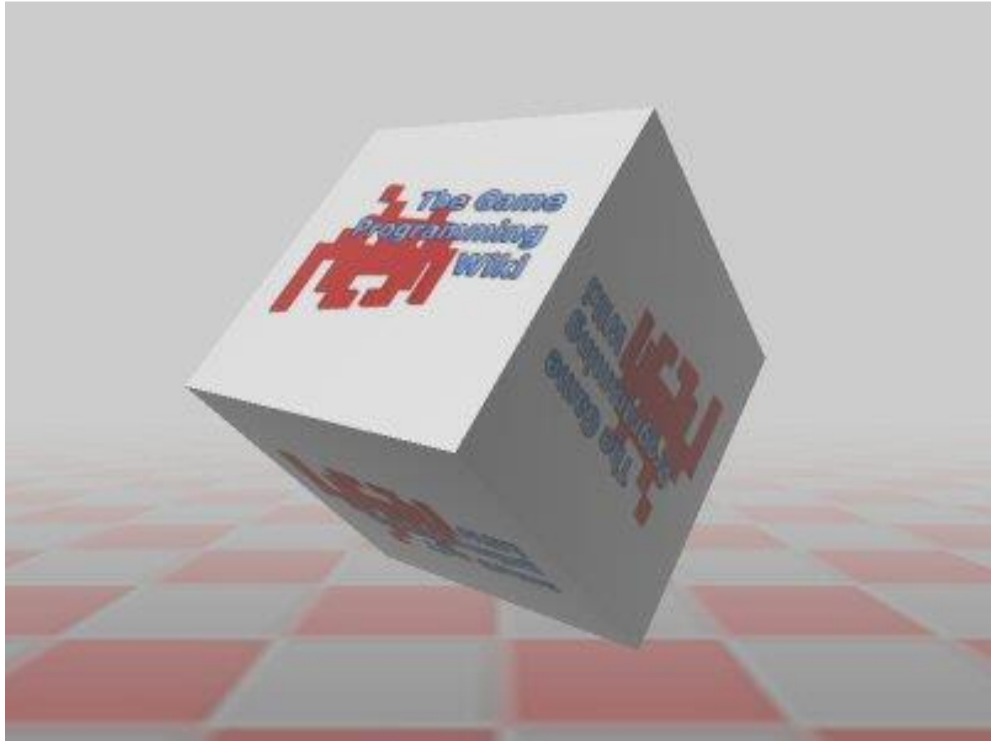
- Fog is approximated by a linear combination of the computed light $\boldsymbol{L}^{\mathsf{cam}}$ and a fog color $\boldsymbol{c}^{\mathsf{fog}}$
- Distance d from the surface to the viewer
- Light $\boldsymbol{L}^{\mathsf{cam,fog}}$ towards sensor considering fog

  $$\boldsymbol{L}^{\mathsf{cam,fog}} = f(d) \cdot \boldsymbol{L}^{\mathsf{cam}} + (1 - f(d)) \cdot \boldsymbol{c}^{\mathsf{fog}}$$
- $0 \leq f(d) \leq 1$ describes the visibility depending on d
  - $f(d) = 1$: max visibility ($\boldsymbol{L}^{\mathsf{cam}}$ is unaffected)
  - $f(d) = 0$: min visibility ($\boldsymbol{L}^{\mathsf{cam}}$ is changed to fog color $\boldsymbol{c}^{\mathsf{fog}}$)
  - E. g.: $f(d) = \frac{d^{\text{end}} - d}{d^{\text{end}} - d^{\text{start}}}$

UNI FREIBURG

Attenuation and Fog

[http://www.gamedev.net/topic/
541383-typical-light-attenuation-coefficients/]



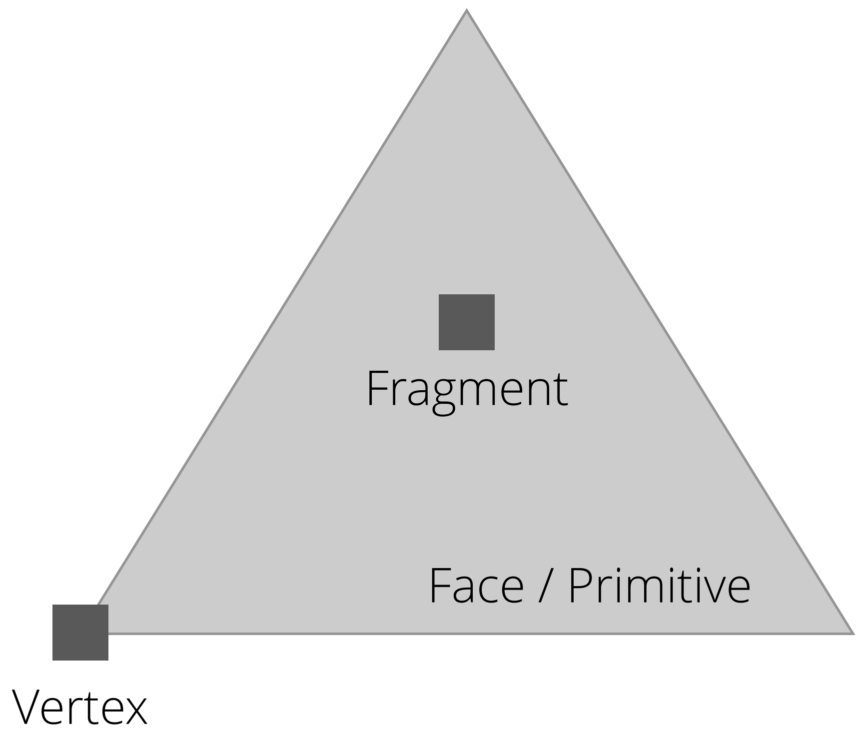
[The Game Programming Wiki:
OpenGL:Tutorials:Tutorial
Framework:Light and Fog]

Outline

- Context
- Phong illumination model
- Extensions
- Shading models

Introduction

- Illumination models can be evaluated per vertex or per fragment
- Faces / primitives, e.g. triangles, are characterized by vertices
- The projected area of a face onto the sensor is subdivided into fragments (one fragment per image pixel)



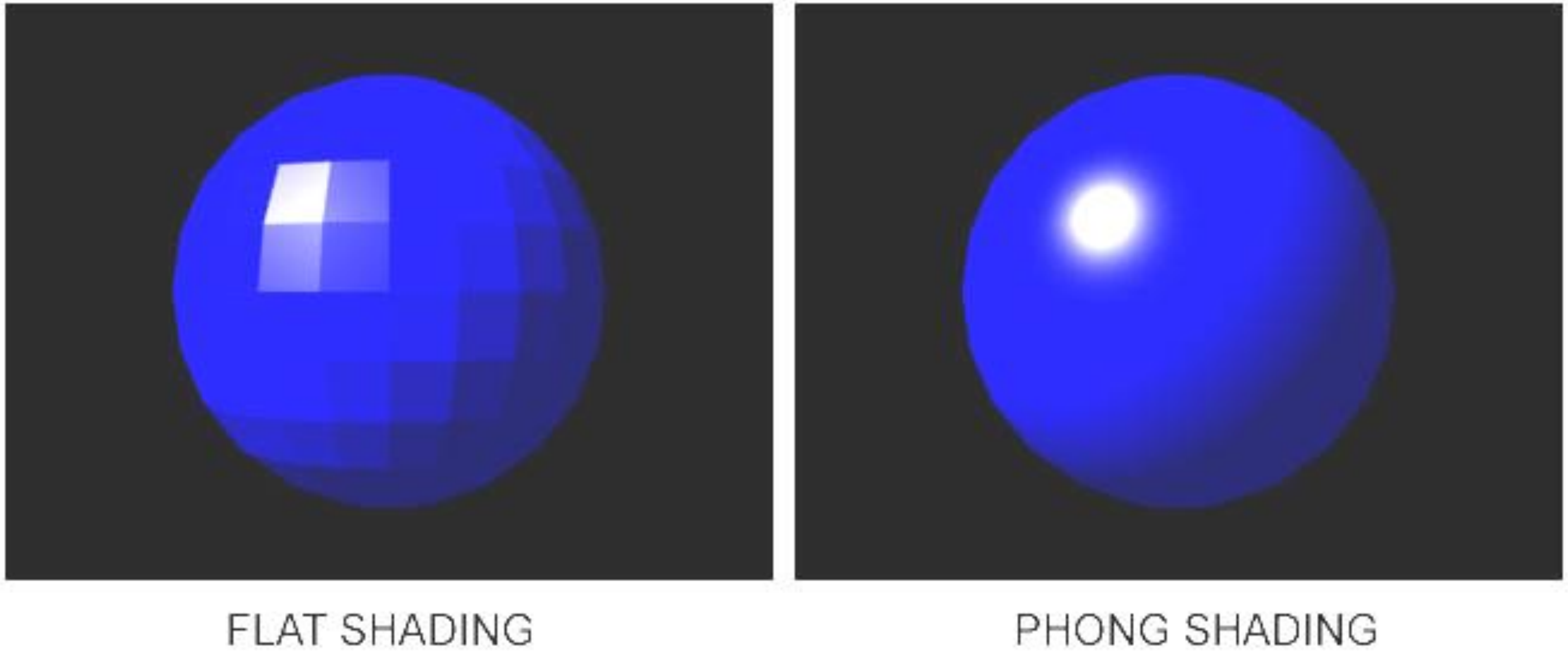
Introduction

- Shading models specify whether the illumination model is evaluated per vertex or per fragment
- If evaluated per vertex, the shading model specifies whether the resulting vertex colors are interpolated across a primitive or not
- If evaluated per fragment, surface normals are interpolated across a primitive

Shading Models

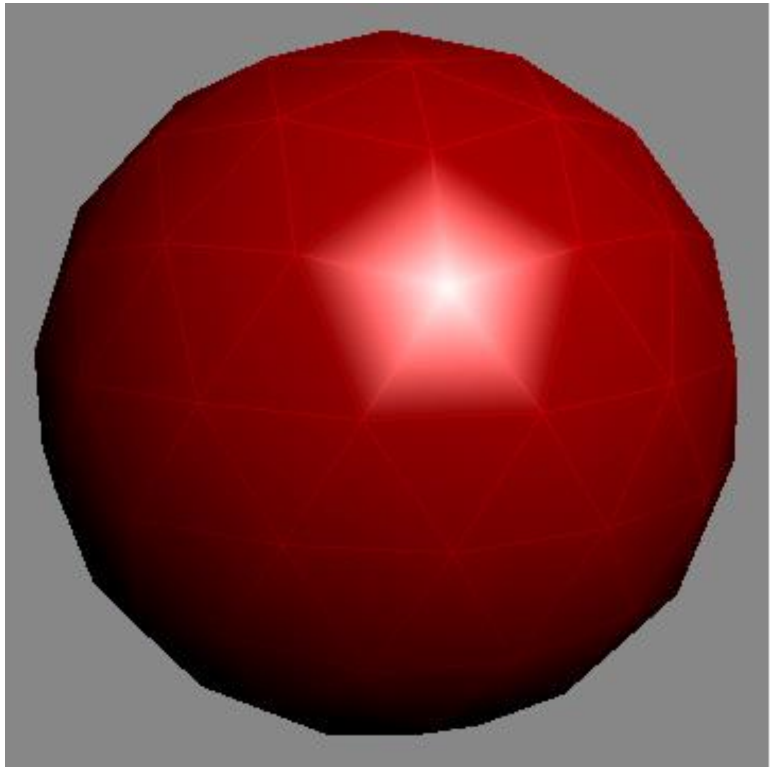
- Flat shading (constant shading)
  - Evaluation per vertex
  - Fragments are colored with the color of one specific vertex
- Gouraud shading
  - Evaluation per vertex
  - Fragment colors are interpolated from vertex colors
- Phong shading
  - Evaluation per fragment
  - Normals have to be interpolated from vertices to fragments

Flat vs. Phong

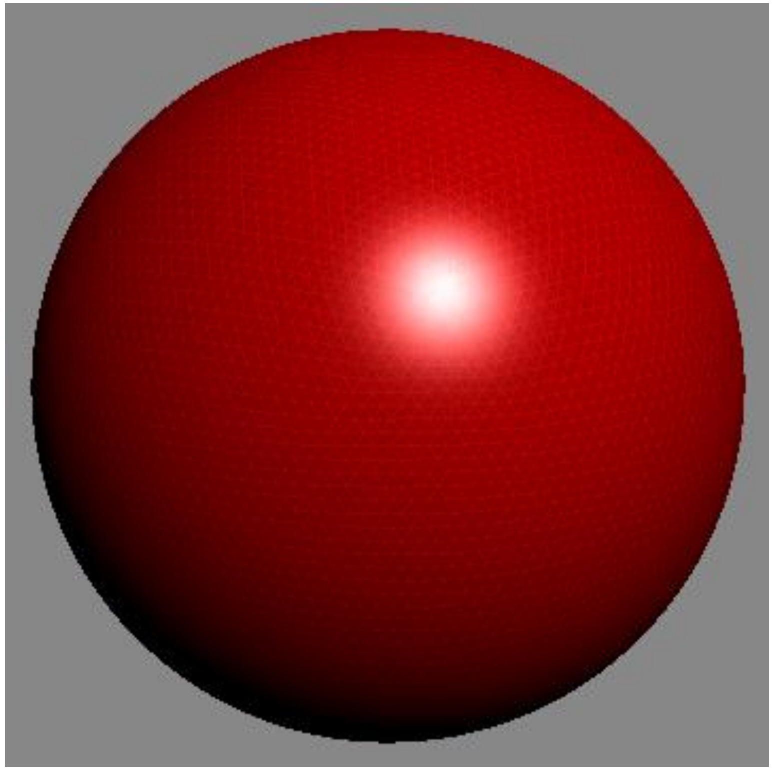


[Wikipedia: Phong shading]

Gouraud Shading



Low primitive count
Highlight is poorly resolved.
Mach band effect.

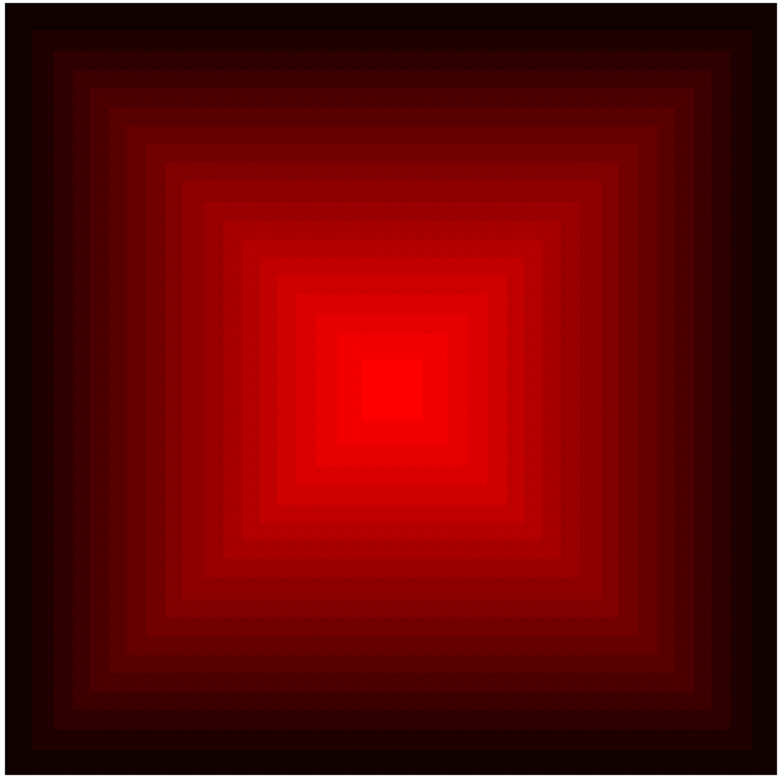


High primitive count

[Wikipedia: Gouraud shading]

Mach Band Effect

– Mach bands are illusions due to our neural processing



The intensity inside each
square is the same.
The bright bands at 45 degrees
and 135 degrees are illusory.

Summary

- Flat shading (constant shading)
  - Efficient
- Phong shading
  - Expensive
- Gouraud shading
  - Mach band effect
  - Local highlights are not resolved, if the highlight is not captured by a vertex