Computer Graphics
Shading

Matthias Teschner
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
  - Local illumination models / Phong illumination model
  - Evaluation of governing equations for light interaction at surfaces (rendering equation) and in participating media (volume rendering equation)
Light

- Modeled as energy parcels / **photons** that travel
  - Along geometric rays (**radiance**)
  - At infinite speed
- Emitted by light sources
- Scattered / absorbed at surfaces
- Scattered / absorbed by participating media
- Absorbed / measured by sensors
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 $\xrightarrow{}$ spectrum $\xrightarrow{}$ color

$$
\Phi_\lambda(\lambda_1) \quad \Phi_\lambda(\lambda_2) \quad \Phi_\lambda(\lambda_3)
$$

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

$$
\Phi = \int_{\text{Visible Spectrum}} \Phi_\lambda(\lambda) d\lambda \\
\approx \sum_i \Phi_\lambda(\lambda_i) \Delta\lambda_i \\
\approx \Phi_{\text{red}} \Delta\lambda + \Phi_{\text{green}} \Delta\lambda + \Phi_{\text{blue}} \Delta\lambda
$$
Colored Light

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

\[ L = \begin{pmatrix} L_{\text{red}} \\ L_{\text{green}} \\ L_{\text{blue}} \end{pmatrix} \]

- RGB color space
Colored Objects

- Surfaces are characterized by a reflectance coefficient
  \[ \rho = \begin{pmatrix} \rho_{\text{red}} \\ \rho_{\text{green}} \\ \rho_{\text{blue}} \end{pmatrix} \]
- Which components of the incoming light are reflected and which are absorbed?
- E.g., a yellow surface is described by \( \rho = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \)
  - 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

\[ L(p, \omega_i) \]

\[ L(p, \omega_o) = \int_{\Omega} \text{mat}(p, \omega_i, \omega_o) L(p, \omega_i) \cos \theta_i \, d\omega_i \]

Outgoing light into direction \( \omega_o \) is a sum of incident light from all directions weighted with material properties \( \text{mat}(p, \omega_i, \omega_o) \)
Local Illumination Models

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

\[ L(p, l_1) \rightarrow l_1 \]
\[ L(p, l_2) \rightarrow l_2 \]
\[ L(p, v) = \sum_i f_{\text{Phong}} (L(p, l_i)) \]

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

\( f_{\text{Phong}} \) is a local illumination model, e.g. the Phong illumination model.
Outline

- Context
- Phong illumination model
- Extensions
- Shading models
Setting

- Light is represented as 3D vector $L = \begin{pmatrix} L_{\text{red}} \\ L_{\text{green}} \\ L_{\text{blue}} \end{pmatrix}$
- Illumination $L^{\text{light}}$ from a point source into direction $-l$
- Viewer / sensor in direction $\nu$ from surface
- Normal direction at surface $n$
- Surface color / material $\rho = \begin{pmatrix} \rho_{\text{red}} \\ \rho_{\text{green}} \\ \rho_{\text{blue}} \end{pmatrix}$

- It is typically assumed, that all vectors are normalized
Setting

Incoming light from direction \( -\mathbf{l} \) from a light source computes reflected light into direction \( \mathbf{v} \) towards the sensor.

\[
L_{\text{cam}} = f_{\text{Phong}} \left( L_{\text{light}}, \rho, l, v, n, \ldots \right)
\]

\( f_{\text{Phong}} \) computes reflected light into direction \( \mathbf{v} \) towards the sensor.
How to Compute Shading?

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

- Angle between surface normal $n$ and light source $l$ 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 \( l \) and \( n \).
Overall Reflected Light

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

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

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

$n$ and $l$ have to be normalized. $n \cdot l$ has to be non-negative.
Overall Reflected Light

- Yellow surface under white illumination

\[ L_{\text{refl}} = \rho \otimes L_{\text{light}} \cdot (n \cdot l) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot (n \cdot l) \]

Reflects yellow light

- Yellow surface under red illumination

\[ L_{\text{refl}} = \rho \otimes L_{\text{light}} \cdot (n \cdot l) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot (n \cdot l) \]

Reflects red light

- Yellow surface under blue illumination

\[ L_{\text{refl}} = \rho \otimes L_{\text{light}} \cdot (n \cdot l) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \cdot (n \cdot l) \]

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 $L^{\text{cam}}$ towards sensor

$$L^{\text{refl}} = \int_{2\pi} L(\omega_o) \cos \theta_o d\omega_o$$  \hspace{1cm} \text{Overall reflected light equals reflected light into a direction integrated over all directions}

$$L(\omega_o) = \text{const} = L^{\text{cam}}$$  \hspace{1cm} \text{Definition of diffuse reflection}

$$L^{\text{refl}} = L^{\text{cam}} \int_{2\pi} \cos \theta_o d\omega_o = L^{\text{cam}} \cdot \pi$$

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

The cosine term in the integral is related to Lambert’s cosine law.
Diffuse Reflection

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

\[ L_{\text{cam}} = \frac{1}{\pi} \cdot \rho \otimes L_{\text{light}} \cdot (n \cdot l) \]
Diffuse Reflection - Discussion

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

- Shiny surfaces reflect light into a small set of dominant directions
- Light $L_{\text{cam}}$ towards sensor

\[
L_{\text{refl}} = \int_{2\pi} L(\omega_o) \cos \theta_o d\omega_o
\]

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

\[
L(\omega_o) \sim (r \cdot \omega_o)^m
\]

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

\[
L_{\text{refl}} = \int_{2\pi} k(r \cdot \omega_o)^m \cos \theta_o d\omega_o
\]

$k$ is not analyzed in Phong's model.

\[
L_{\text{cam}} = \rho^{\text{white}} \otimes L_{\text{light}} \cdot (n \cdot l) \cdot (r \cdot v)^m
\]

White surface color accounts for the fact that shiny surfaces reflect the entire light spectrum.
Specular Reflection

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

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

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 absorb incoming light.
Reflection Vector

- Computed with light source direction $l$ and surface normal $n$
  \[ r + l = 2 \cdot \cos \theta \cdot n \]
  \[
  \cos \theta = l \cdot n
  \]
  \[
  r = 2 \cdot (l \cdot n) \cdot n - l
  \]

- Vectors $l$ and $n$ have to be normalized
- Vector $r$ is normalized

\[ r + l = 2 \cdot \cos \theta \cdot n \]
Specular Reflection

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

\[
L_{\text{cam}} = \rho_{\text{white}} \odot L_{\text{light}} \cdot (n \cdot l) \cdot (n \cdot h)^m
\]

\[
h = \frac{l + v}{||l + v||}
\]

No considerable difference to Phong.
Specular Reflection - Discussion

– Light from light source \( L \text{\scriptsize{light}} \)
– Illumination at the surface \( L \text{\scriptsize{light}} \cdot (n \cdot l) \)
– Overall reflected light \( \rho \text{\scriptsize{white}} \otimes L \text{\scriptsize{light}} \cdot (n \cdot l) \)
– Reflected light towards viewer \( \rho \text{\scriptsize{white}} \otimes L \text{\scriptsize{light}} \cdot (n \cdot l) \cdot (r \cdot v)^m \)
– 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

Exponent $k$ governs the size of the highlight area. $K$ 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 $\theta$ and exponent $m$
- Overall reflected light depends on $m$

Angle $\theta$ between $\mathbf{v}$ and $\mathbf{r}$ (Phong) or $\mathbf{n}$ and $\mathbf{h}$ (Blinn-Phong)

$\cos^m \theta$
Reflection From Ambient Illumination

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

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

\[ L^{\text{indirect}} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \quad \rho^{\text{cube}} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \]

\[ L^{\text{cam}} = \frac{1}{\pi} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \frac{1}{\pi} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \]
Phong Illumination Model

- Combination of ambient, diffuse and specular reflection

[Wikipedia: Blinn-Phong shading model]
Phong Illumination Model

\[
L_{\text{cam}} = \frac{1}{\pi} \cdot \rho \otimes L_{\text{indirect}} + \sum_i L_{i}^{\text{light}} \cdot (n \cdot l_i) \otimes \left( \frac{1}{\pi} \cdot \rho + \rho^{\text{white}} \cdot (r_i \cdot v)^m \right)
\]

- As the specular term does not account for energy preservation, ambient, diffuse and specular reflection are weighted by user-defined scalar coefficients \(\alpha, \beta, \gamma\)

\[
L_{\text{cam}} = \alpha \cdot \rho \otimes L_{\text{indirect}} + \sum_i L_{i}^{\text{light}} \cdot (n \cdot l_i) \otimes \left( \beta \cdot \rho + \gamma \cdot \rho^{\text{white}} \cdot (r_i \cdot v)^m \right)
\]
Phong Illumination Model

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.

\[ L_{\text{cam}} = k_{\text{amb}} \times L_{\text{amb}} + \sum_i L_{i}^{\text{light}} \cdot (n \cdot l_i) \times (k_{\text{diff}} + k_{\text{spec}} \cdot (r_i \cdot v)^m) \]

Or

\[ L_{\text{cam}} = k_{\text{amb}} \times L_{\text{amb}} + k_{\text{diff}} \times L_{\text{diff}} \cdot (n \cdot l) + k_{\text{spec}} \times L_{\text{spec}} \cdot (r \cdot v)^m \]

Weighting coefficients 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 computation from local information
  – 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 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 $p_{\text{light}}$, surface $p$, distance $r = \|p_{\text{light}} - p\|
  - Illumination at surface
    \[ L_{\text{surf}} = \frac{1}{r^2} \cdot L_{\text{light}} \cdot (n \cdot l) \]
  - Variant
    \[ L_{\text{surf}} = \frac{1}{k_e + k_l \cdot r + k_q \cdot r^2} \cdot L_{\text{light}} \cdot (n \cdot l) \]

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

[Wikipedia: Inverse Square Law]
Fog

- Fog is approximated by a linear combination of the computed light $L_{\text{cam}}$ and a fog color $c_{\text{fog}}$
- Distance $d$ from the surface to the viewer
- Light $L_{\text{cam, fog}}$ towards sensor considering fog

$$L_{\text{cam, fog}} = f(d) \cdot L_{\text{cam}} + (1 - f(d)) \cdot c_{\text{fog}}$$

- $0 \leq f(d) \leq 1$ describes the visibility depending on $d$
  - $f(d) = 1$: max visibility ( $L_{\text{cam}}$ is unaffected)
  - $f(d) = 0$: min visibility ( $L_{\text{cam}}$ is changed to fog color $c_{\text{fog}}$)
  - E. g.: $f(d) = \frac{d_{\text{end}} - d}{d_{\text{end}} - d_{\text{start}}}$
Attenuation and Fog

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

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