Computer Graphics

8 - Lighting

Yoonsang Lee
Hanyang University

Spring 2023

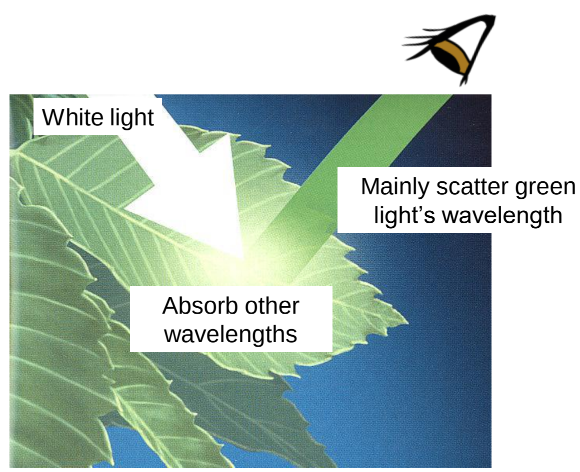
Outline

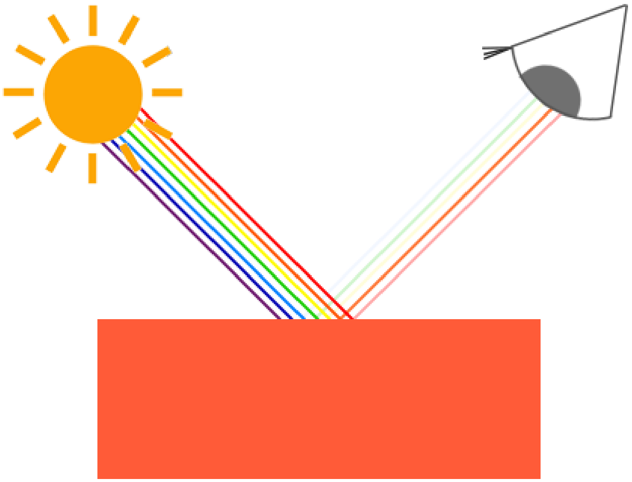
- Visible Color of Objects
- Reflection of Light
- Phong Illumination Model
- Polygon Shading
  - Face / Vertex Normal
  - Flat / Goraud / Phong Shading

Visible Color of Objects

Visible Color of Objects

- When light strikes an object, some of the light is absorbed by the object, and some is reflected.
- The color of the object is determined by the **wavelengths of light that are reflected.**
  - For example, a red object appears red because it reflects primarily red light and absorbs other wavelengths.
- Which color is absorbed or reflected is an **inherent property of a surface.**

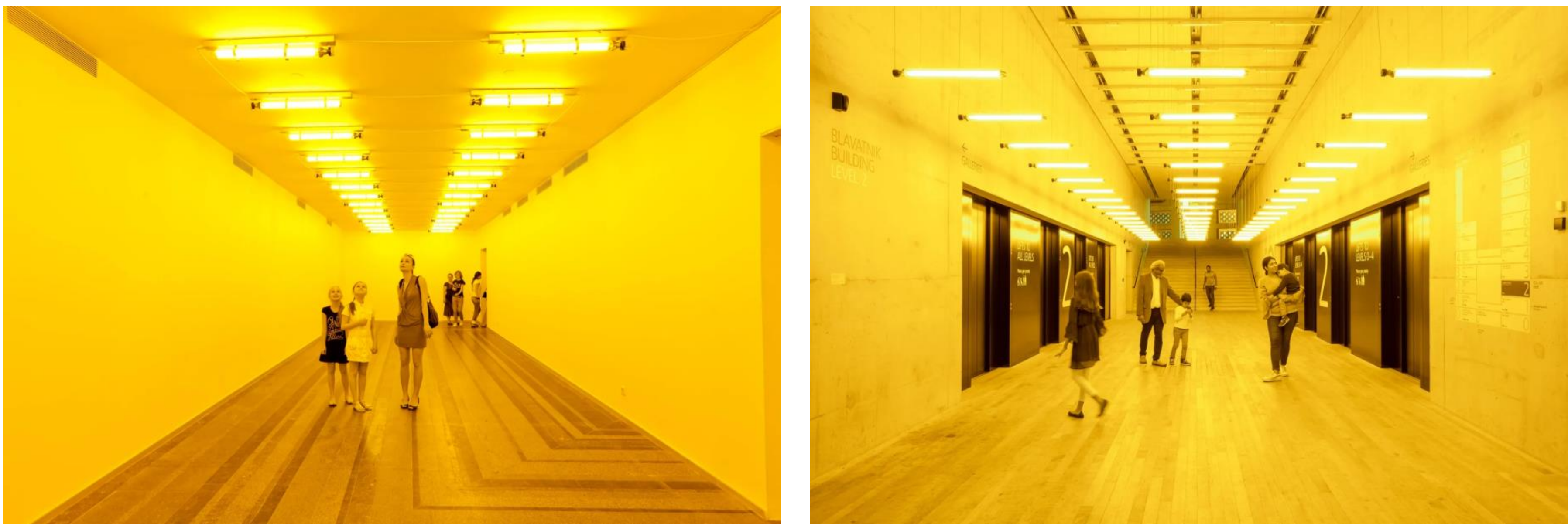




* This image is from https://learnopengl.com/Lighting/Colors

Visible Color of Objects

- So, quite obviously, the visible color of an object is affected by the color of the light source.



Room for one colour, Olafur Eliasson

* This image is from https://olafureliasson.net/artwork/room-for-one-colour-1997/

Computing Visible Color of Objects

- In CG, color is usually represented by R, G, B components.

- Light color: The intensity of each color component emitted by a light source.
  - e.g., (1, 1, 1) → white light source

- Material color: The percentage of each color component reflected in incident light.
  - e.g., (0.5, 0, 0) → half red is reflected, green and blue are all absorbed

- **Element-wise multiplication** of the light and material RGB color values is a good approximation of the surface's light reflection.

Computing Color of Objects: Examples

- For example,
- Material color of a surface is (0.5, 0.8, 0.2).
  - This surface reflects 50% of red, 80% of green, and 20% of blue in incident light.

- If light color is (1.0, 1.0, 1.0),
- Visible surface color is (0.5, 0.8, 0.2). (element-wise multiplication)

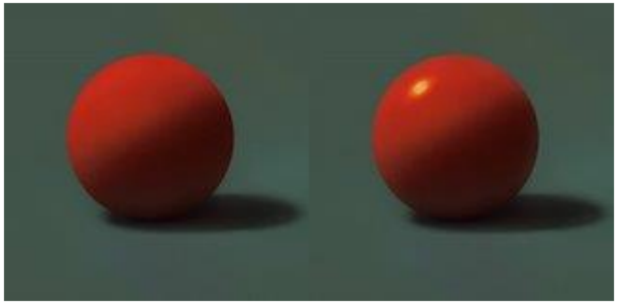
- If light color is (1.0, 0.0, 0.0),
- Visible surface color is (0.5, 0.0, 0.0). → Darker red surface.

Reflection of Light

Reflection of Light

- Light can be absorbed(흡수), emitted(발산), scattered(산란), reflected(반사), or refracted(굴절) by objects.

- Scattering and reflection are the main factors in the visual characteristics of an opaque object surface.
  - such as surface color, highlight on surface



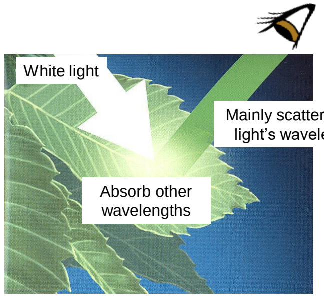
- Types of reflection:
  - Diffuse reflection
  - Specular reflection
    - Ideal specular reflection
    - Non-ideal specular reflection (a.k.a. Glossy reflection)

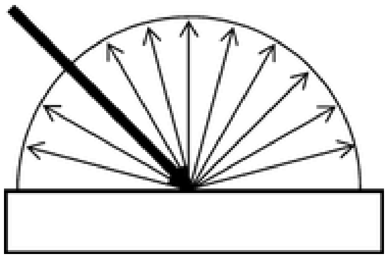
* In computer graphics, both scattering and reflection are often referred to as "reflection"

Diffuse Reflection

- : Scattering specific light spectrum in all direction
- → Determines surface color

- **View-independent**





strongly scatters magenta wavelengths

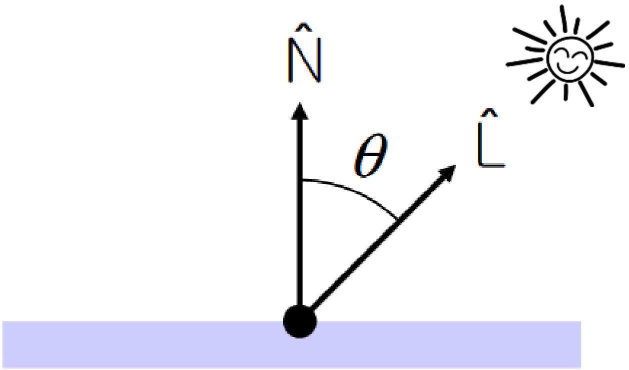
scatter all wavelengths with roughly equal strength

absorb all wavelengths (scatters a little)

Diffuse Reflection - Lambert's Cosine Law

- The **reflected energy** from a small surface area is proportional to the **cosine of the angle** between **incident light direction** and the **surface normal**

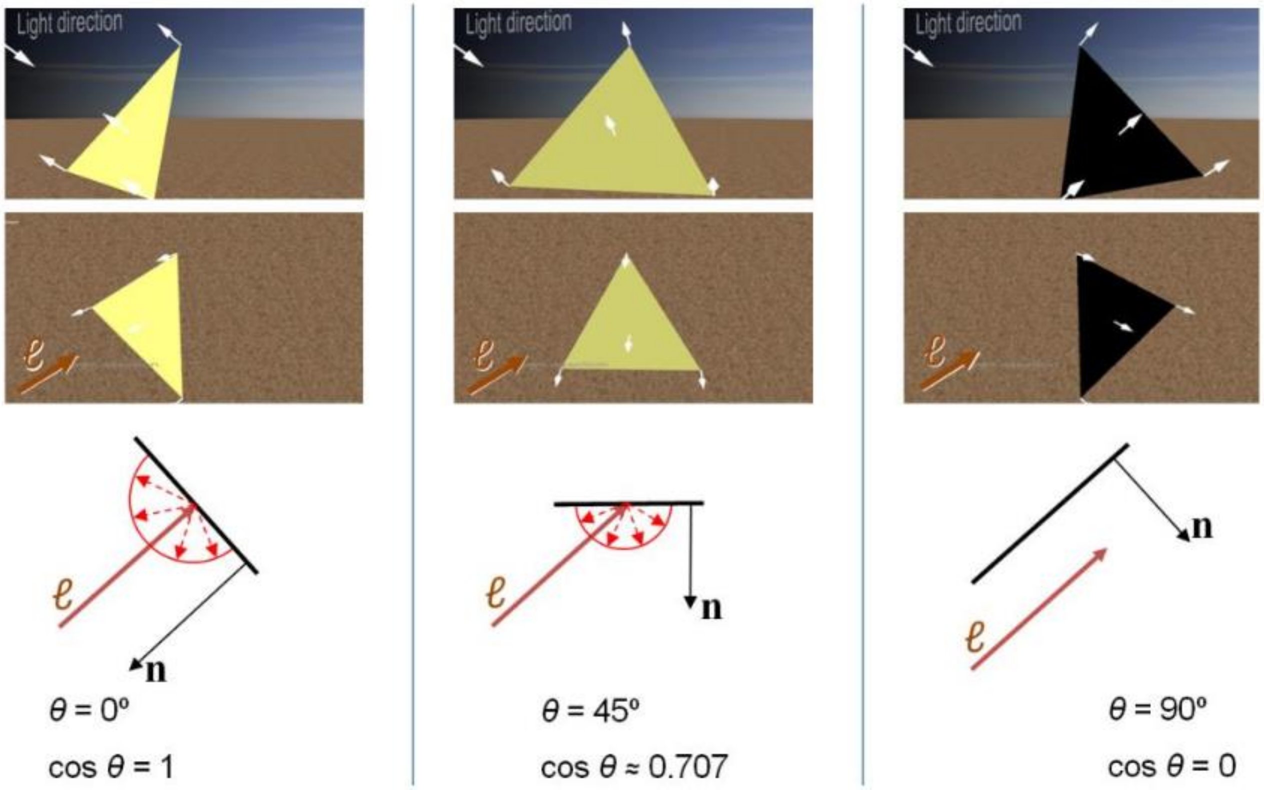
$$I_{reflected} = I_{incident} cos\theta$$
$$= I_{incident}(\hat{\mathbf{N}} \cdot \hat{\mathbf{L}})$$



$I_{reflected}$	intensity of reflected ray
$I_{incident}$	intensity of incident ray
$\hat{\mathbf{N}}$	normal to the reflection surface at the point of the incidence
$\hat{\mathbf{L}}$	normalized light direction vector

Diffuse Reflection - Lambert's Cosine Law

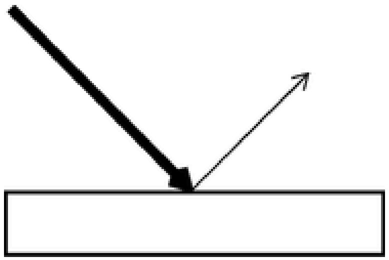
- Visualization of Lambert's law in 2D
-



* This slide is from the slides of Prof. Andy van Dam (Brown Univ.)
http://cs.brown.edu/courses/csci1230/lectures.shtml

Ideal Specular Reflection

- : Mirror-like reflection of light from smooth, polished surface
- → Generate mirrored images



- **View-dependent**

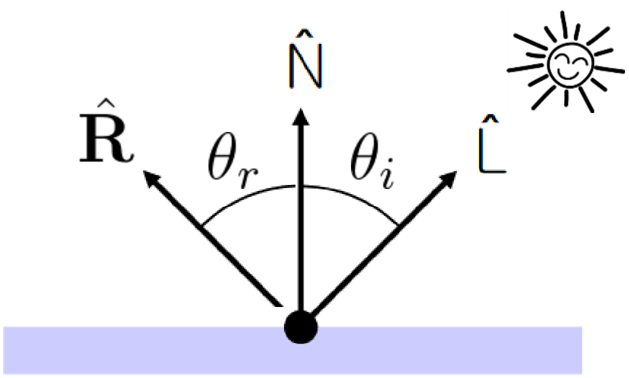
Ideal Specular Reflection - Laws of Reflection

- $\hat{\mathbf{N}}$, $\hat{\mathbf{L}}$, $\hat{\mathbf{R}}$ lie in the same plane
- $\theta_r = \theta_i$
- $\hat{\mathbf{L}}$ and $\hat{\mathbf{R}}$ are on the opposite sides of $\hat{\mathbf{N}}$

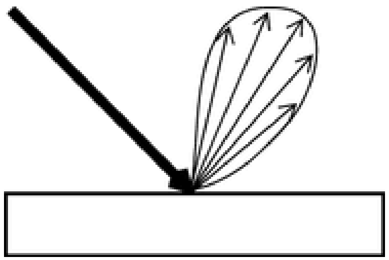
$\hat{\mathbf{N}}$ normal to the reflection surface at the point of the incidence

$\hat{\mathbf{L}}$ normalized indicent ray direction vector

$\hat{\mathbf{R}}$ normalized reflected ray direction vector



Non-Ideal Specular Reflection (a.k.a. Glossy Reflection)

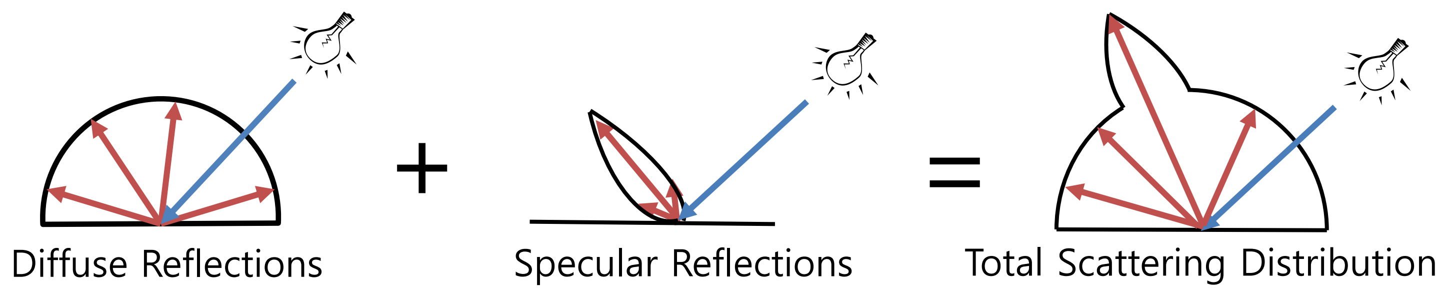


- : Reflection on shiny & glossy surface, but not as smooth as a mirror
- Reflected rays are “spread out” due to surface roughness
- → Generate bright highlights

- **View-dependent**

Reflection of General Materials

- Many materials' surface have both diffuse reflection and (non-ideal) specular reflection.



* This image is from the slides of Prof. Andy van Dam (Brown Univ.)
http://cs.brown.edu/courses/csci1230/lectures.shtml

Quiz 1

- Go to https://www.slido.com/
- Join #**cg-ys**
- Click "Polls"

- Submit your answer in the following format:
  - **Student ID: Your answer**
  - **e.g. 2021123456: 4.0**

- Note that your quiz answer must be submitted **in the above format** to receive a quiz score!

Phong Illumination Model

Lighting (or Illumination)

- In computer graphics, **lighting** (or **illumination**) refers to the process of computing the effects of lights.

- → Computing surface color and highlights of objects.

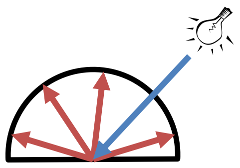
Phong Illumination Model

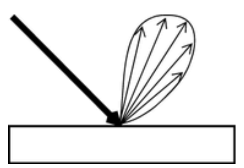
- One of the most commonly used “classical” illumination models in computer graphics
  - Empirical model, not physically based

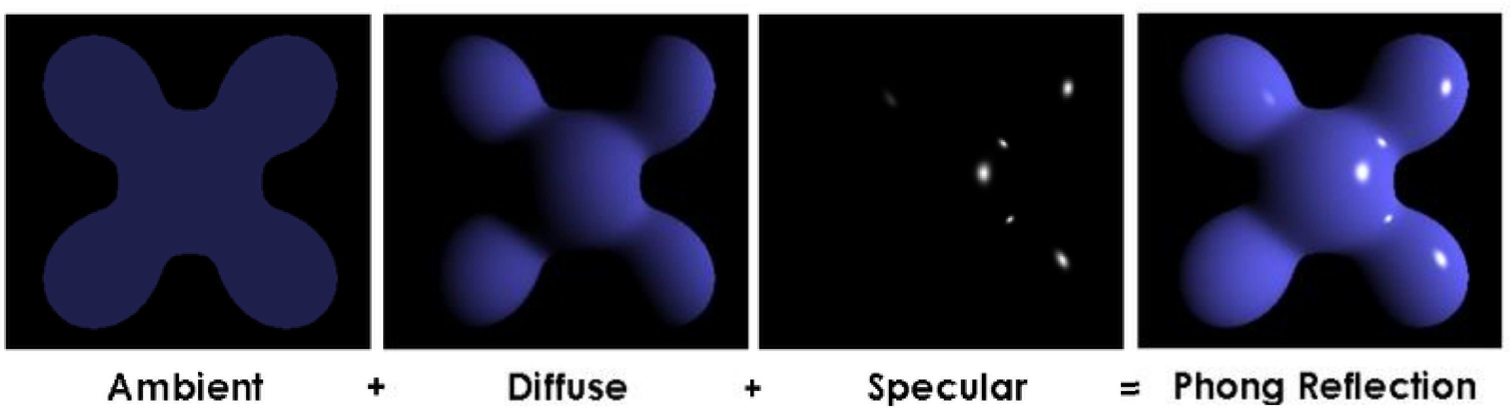
Bùi Tường Phong
(1942 – 1975)

Phong Illumination Model

- Three components:
- **Ambient**
  - Non-specific constant global lighting.
  - Crudest approximation for indirect lighting.
- **Diffuse**
  - Models diffuse reflection using Lambert's law.
  - Determine the surface color.
- **Specular**
  - Approximation for glossy reflection using $\cos^n(\alpha)$.
  - Computes highlights on shiny objects.







Phong Illumination Model

- Now we will look at how to calculate each component of Phong illumination model - ambient, diffuse, specular color - at a specific location on an object's surface.

- The location might be
  – a polygon vertex
  – or an interior point in a polygon (corresponds to a pixel in the film space).

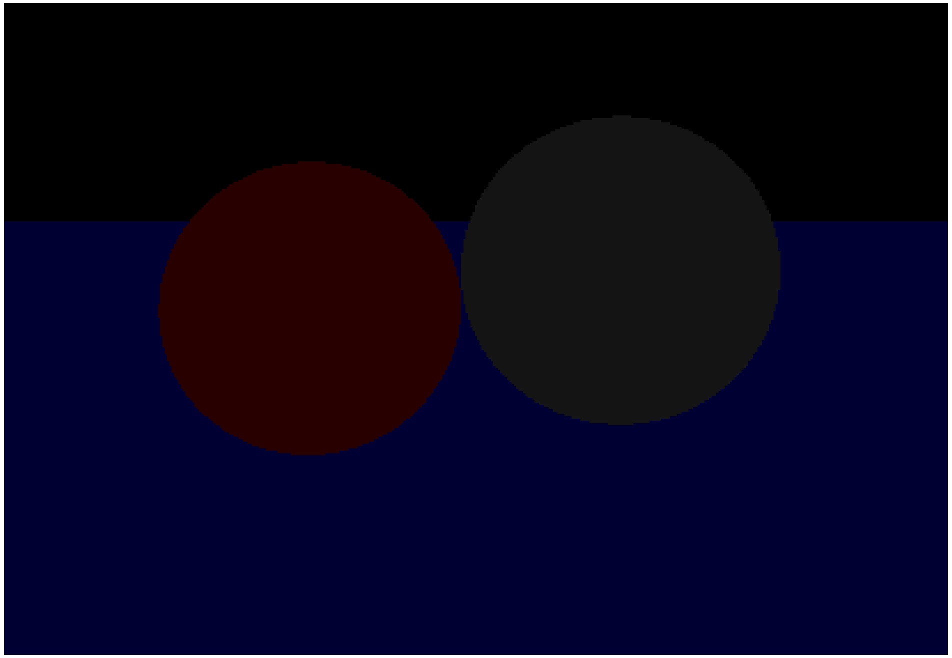
Ambient Component

- $\mathbf{I}_a = \mathbf{l}_a * \mathbf{m}_a$

- $\mathbf{l}_a$: light ambient color
- $\mathbf{m}_a$: material ambient color
- $\mathbf{I}_a$: final ambient color of a surface point
- * : element-wise multiplication

Result

- $\mathbf{I = l_a * m_a}$

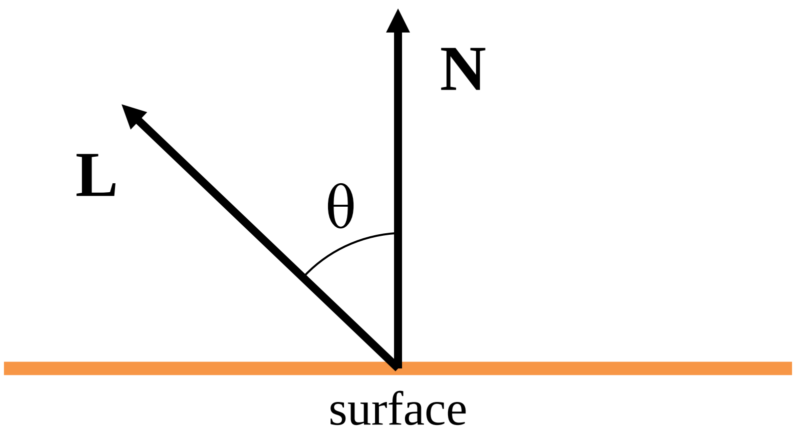


* The images are from the slides of Prof. Jinxiang Chai (Texas A&M University): http://faculty.cs.tamu.edu/jchai/csce441_2016spring/lectures.html

Diffuse Component

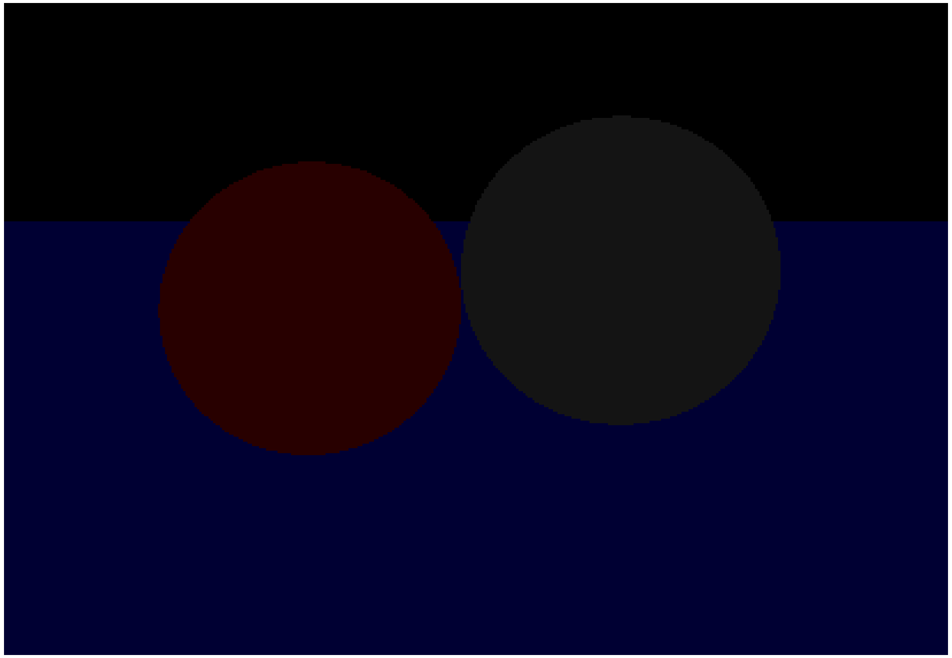
- $\mathbf{I}_d = \mathbf{l}_d * \mathbf{m}_d \cos(\theta) = \mathbf{l}_d * \mathbf{m}_d (\mathbf{L} \cdot \mathbf{N})$

- **L** : light direction
- **N** : normal
  - **L** and **N** are unit vectors.
- · : dot (inner) product
- $\mathbf{l}_d$: light diffuse color
- $\mathbf{m}_d$: material diffuse color
- $\mathbf{I}_d$: final diffuse color of a surface point



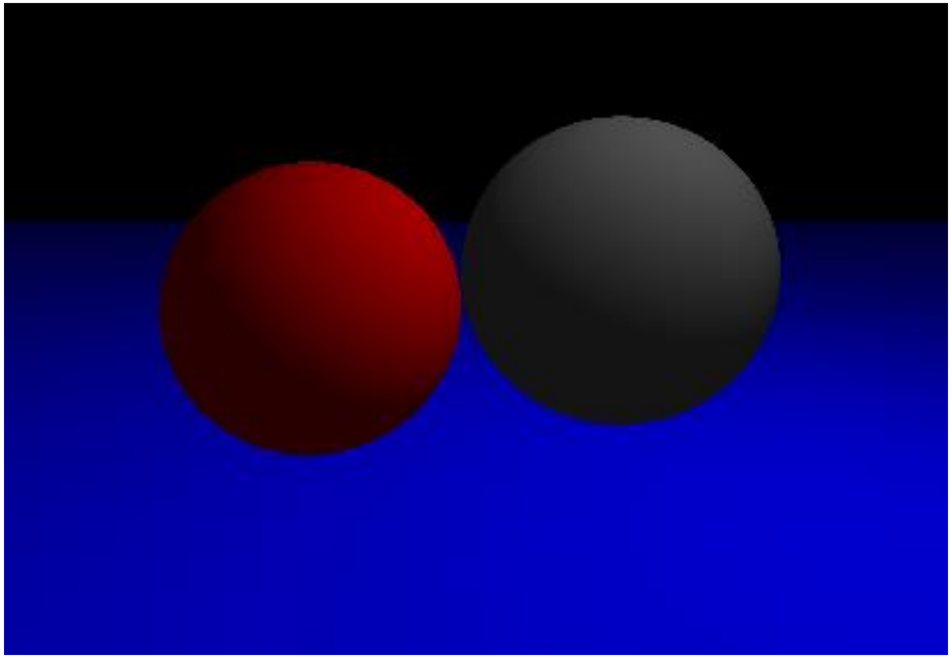
Result

- $\mathbf{I = l_a * m_a}$



Result

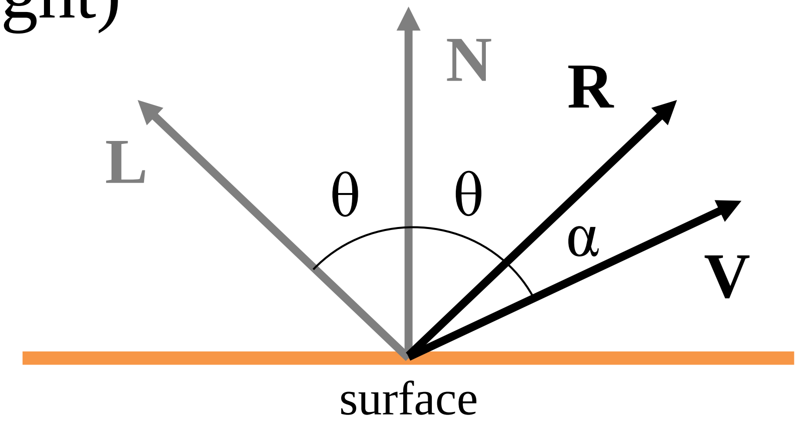
- $\mathbf{I = l_a * m_a + l_d * m_d (L \cdot N)}$



Specular Component

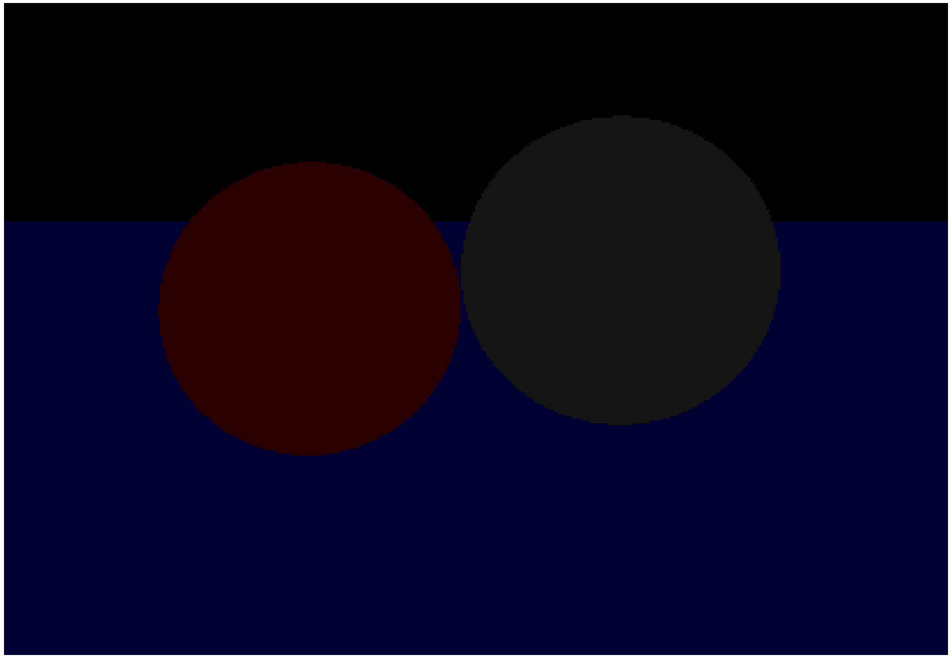
- $\mathbf{I}_s = \mathbf{l}_s * \mathbf{m}_s \cos^n(\alpha) = \mathbf{l}_s * \mathbf{m}_s (\mathbf{V} \cdot \mathbf{R})^n$

- **V** : view direction
- **R** : reflection direction (of light)
  - **V** and **R** are unit vectors.
- n : shininess coefficient
- $\mathbf{l}_s$: light specular color
- $\mathbf{m}_s$: material specular color
- $\mathbf{I}_s$: final specular color of a surface point



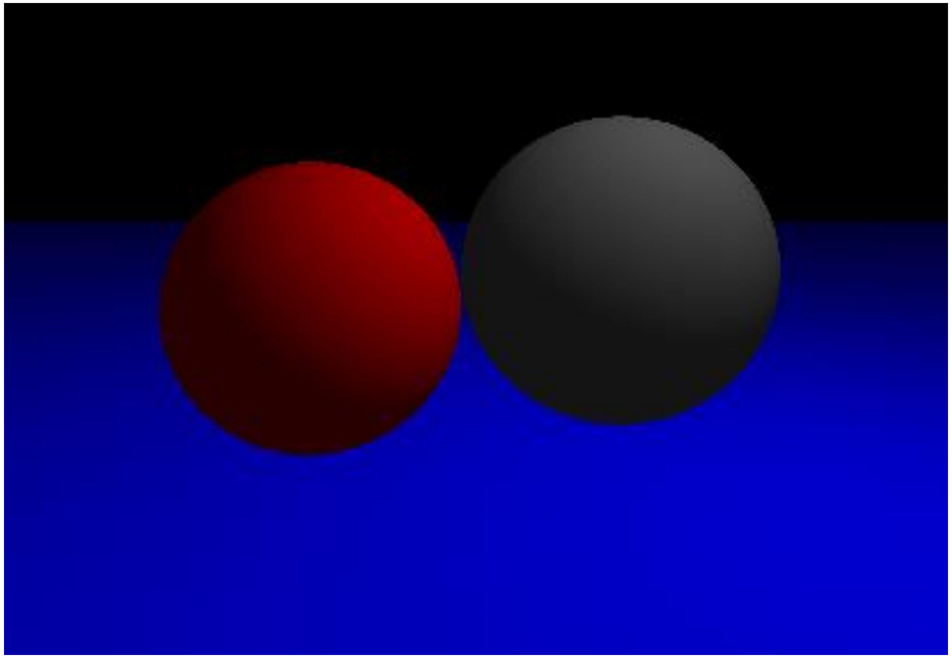
Result

- $\mathbf{I = l_a * m_a}$



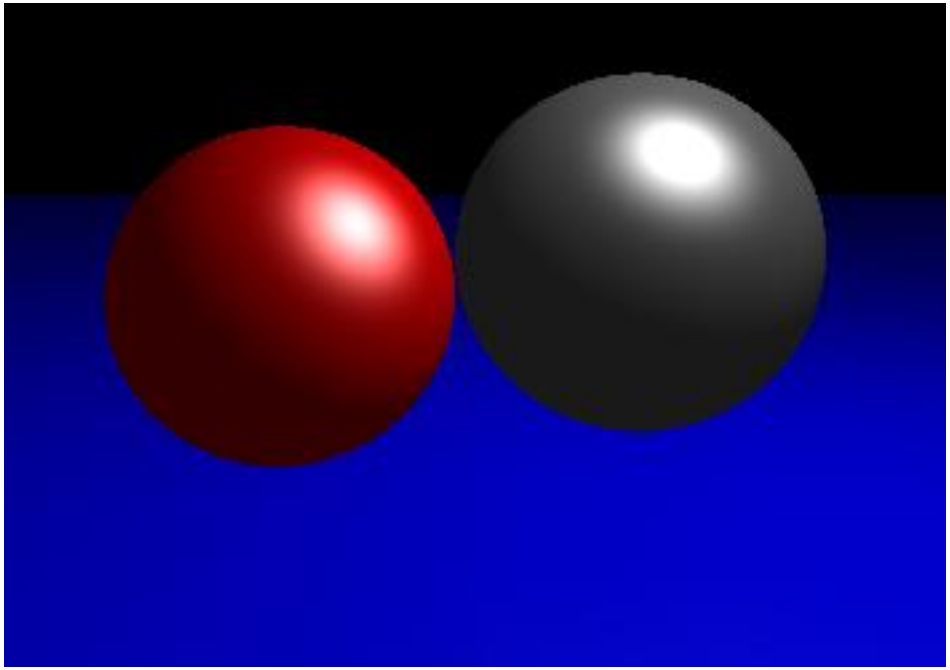
Result

- $\mathbf{I = l_a * m_a + l_d * m_d (L \cdot N)}$



Result

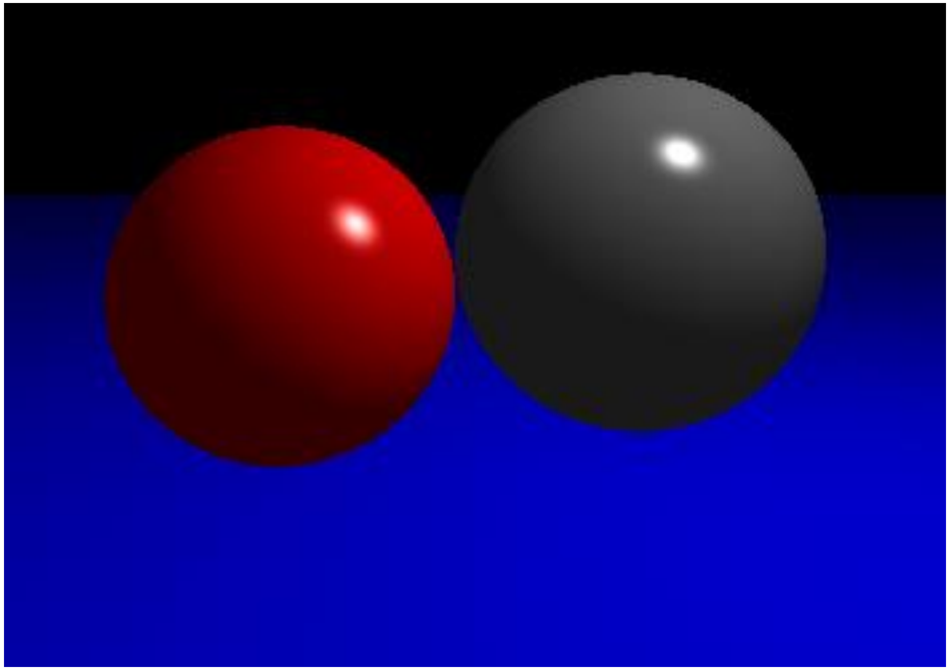
- $\mathbf{I = l_a * m_a + l_d * m_d (L \cdot N) + l_s * m_s (V \cdot R)^n}$



n = 5

Result

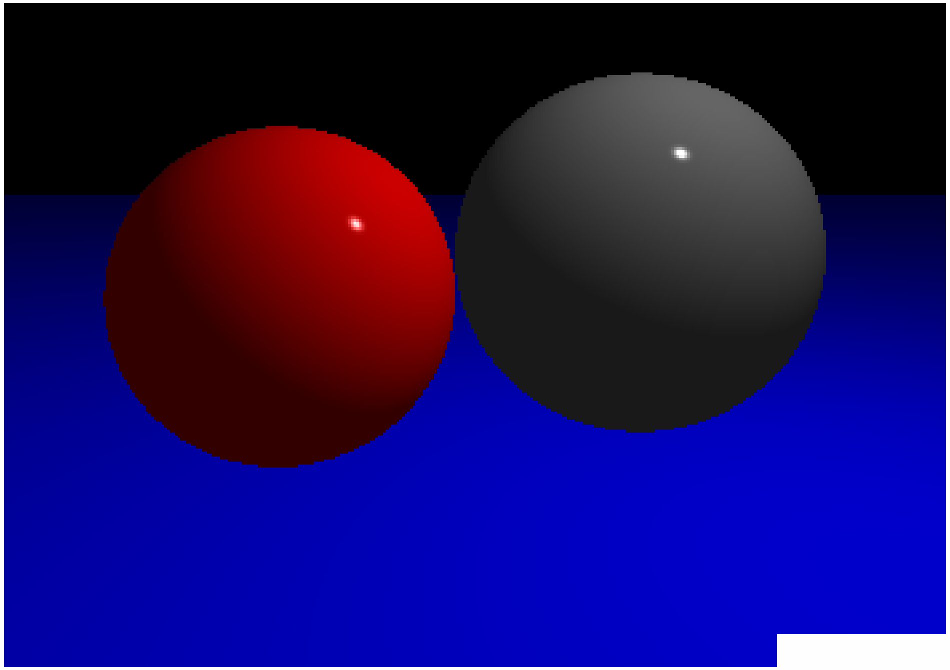
- $\mathbf{I = l_a * m_a + l_d * m_d (L \cdot N) + l_s * m_s (V \cdot R)^n}$



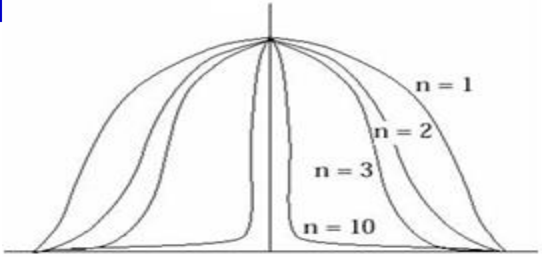
n = 50

Result

- $\mathbf{I} = \mathbf{l}_a * \mathbf{m}_a + \mathbf{l}_d * \mathbf{m}_d\,(\mathbf{L} \cdot \mathbf{N}) + \mathbf{l}_s * \mathbf{m}_s\,(\mathbf{V} \cdot \mathbf{R})^n$

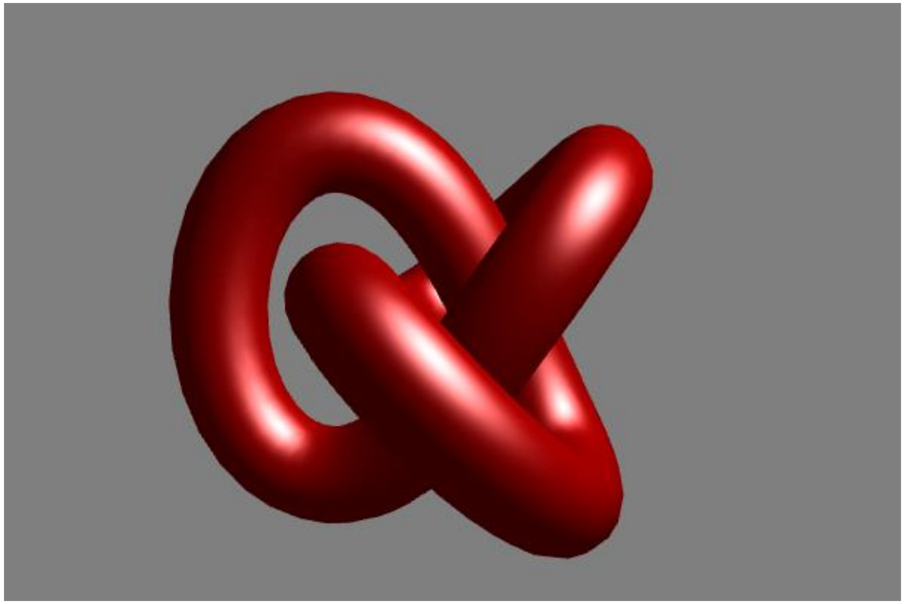


n = 500



Specular falloff of $(\cos \alpha)^n$

[Demo] Phong Illumination



http://www.cs.toronto.edu/~jacobson/phong-demo/

- Set the value of the first drop down box to “Phong Shading”
- Try changing
  - reflection coefficient and color of ambient, diffuse, and specular
  - specular shininess
  - you can also change object type, light position and background color

Quiz 2

- Go to https://www.slido.com/
- Join #**cg-ys**
- Click "Polls"

- Submit your answer in the following format:
  - **Student ID: Your answer**
  - **e.g. 2021123456: 4.0**

- Note that your quiz answer must be submitted **in the above format** to receive a quiz score!

Polygon Shading

Shading

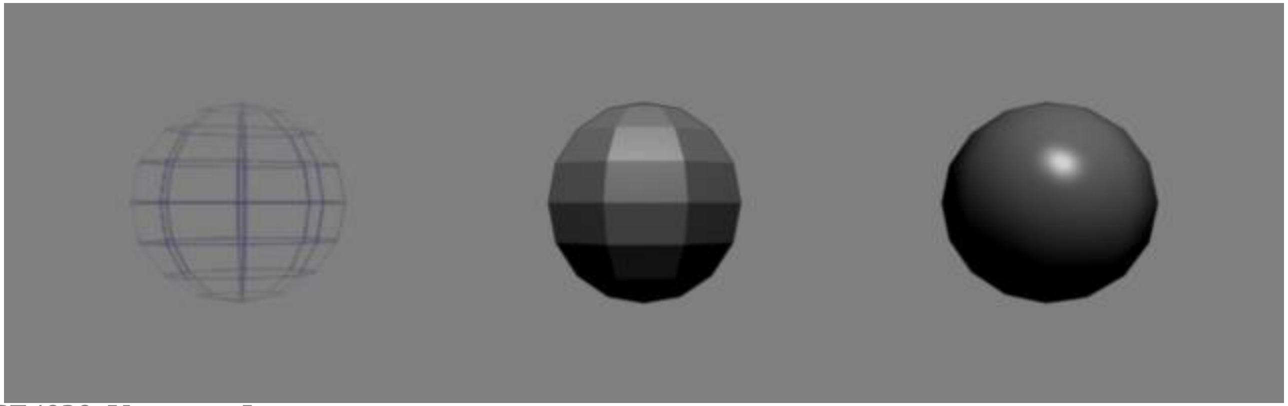
- Variation in observed color across an object
  - Strongly affected by lighting

Polygon Shading

- In computer graphics, the term *shading* describes…
  - Variation in surface color due to the illumination model
  - or Variation in pixel color inside a polygon

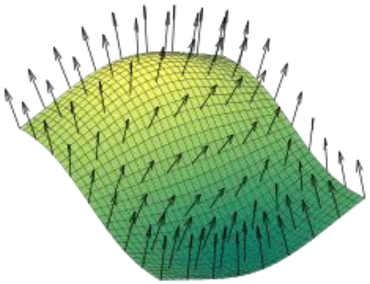
- The second meaning is what we're dealing with now. I'll use the term *polygon shading* to avoid confusion.

- Polygon shading: The process of determining **each pixel color in a polygon** based on an illumination model



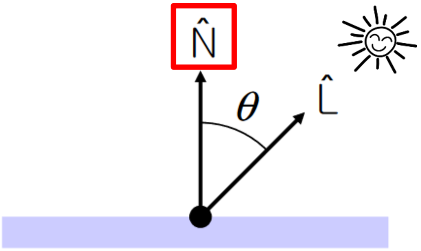
Surface Normal

- A vector that is perpendicular to the surface at a given point
  - A unit normal vector (of length 1) is generally used



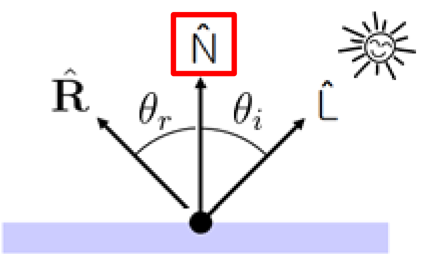
- Plays a key role in shading & illumination process

- Diffuse reflection
  - Lambert's Cosine Law



$$I_{reflected} = I_{incident} cos\theta$$

- Specular reflection
  - Laws of Reflection



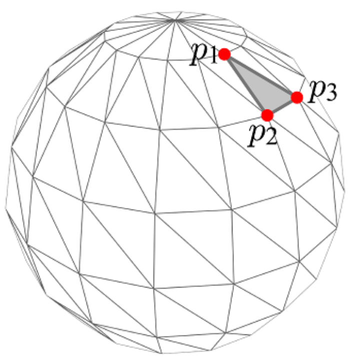
$$\theta_r = \theta_i$$

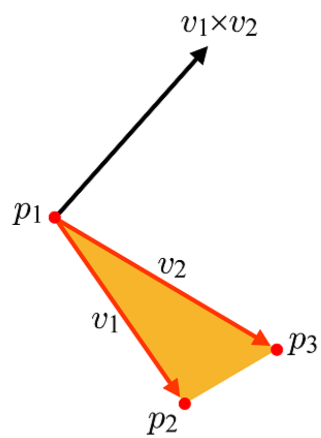
Face Normal

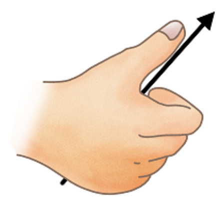
- How to get the face normal - the surface normal of a polygonal face?

The order does matter!

- The normal of a triangle **<p1, p2, p3>** is computed as v1×v2
  - v1 is the vector connecting p1 and p2, v2 connects p1 and p3







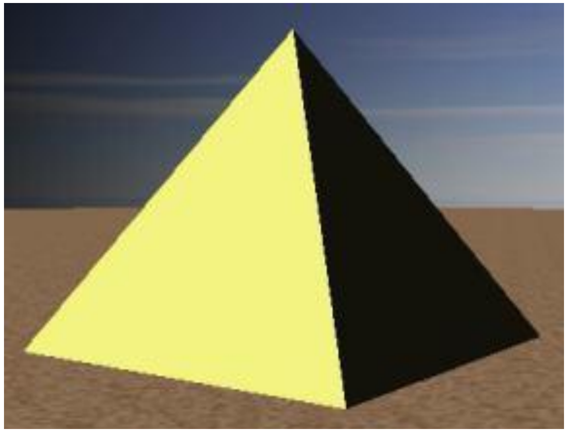
$$\frac{v_1 \times v_2}{||v_1 \times v_2||}$$

- That’s why we need **counterclockwise** vertex ordering
  - The direction of a face normal determines “outside” of the face

* This image is from the slides of Prof. JungHyun Han (Korea Univ.)
http://media.korea.ac.kr/book/

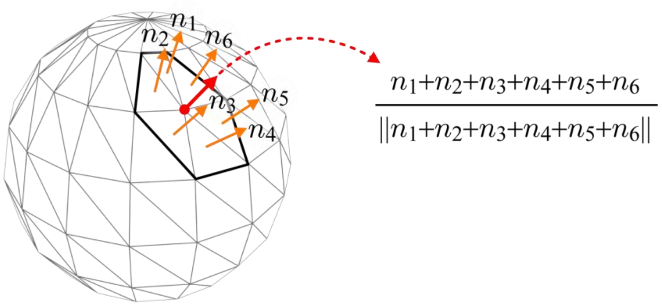
Flat Shading

- Use a single normal per polygon
- Calculate color once per polygon
- Fast, but not very desirable for curved shapes
  - Even if we increase the number of polygons, it's still "faceted"



Smooth Shading

- Use a single "averaged" normal per vertex

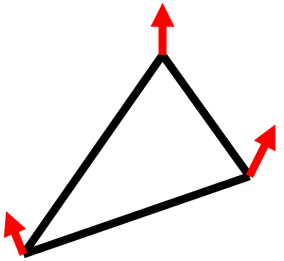


- Smooth color transition between two adjacent polygons

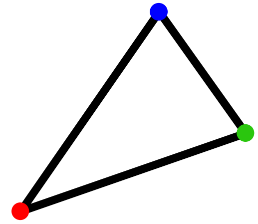
- Two methods:
  - Gouraud shading
  - Phong shading

Gouraud Shading

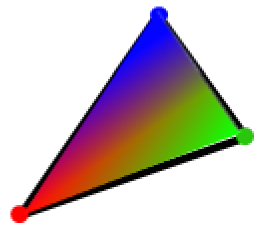
Henri Gouraud (1944~)



- Use a single vertex normal for each vertex

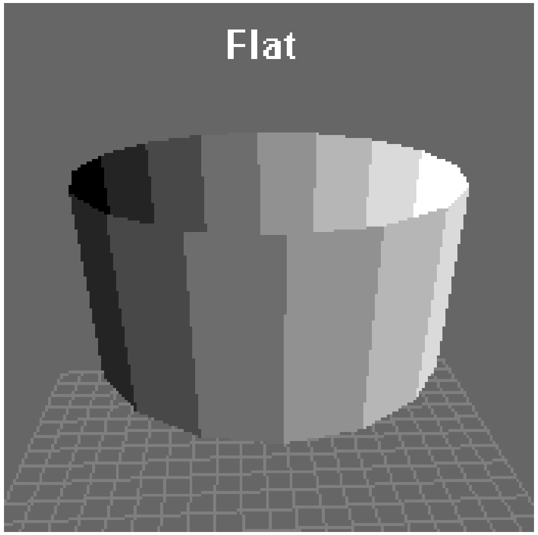


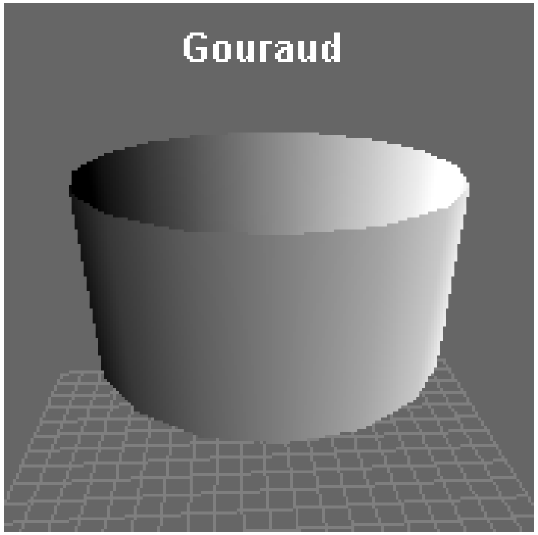
- Calculate color (by illumination) at each vertex



- Interpolate vertex colors across polygon
  - Barycentric interpolation

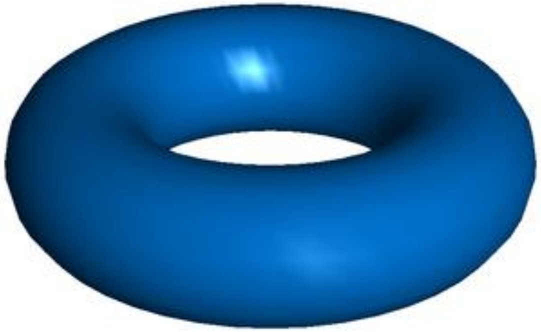
Gouraud Shading

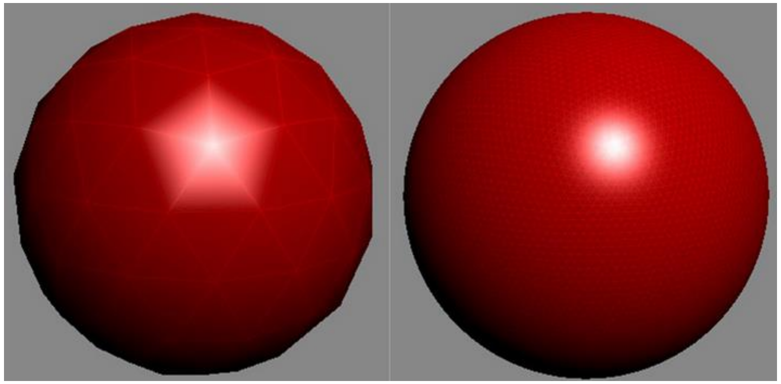




Gouraud Shading

- Problem: poor specular highlight
  - Specular highlights may be distorted or averaged away altogether



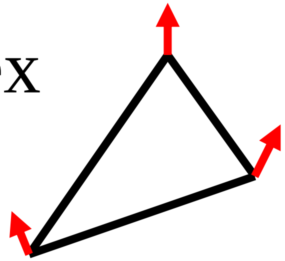


Higher polygon count reduces this artifact

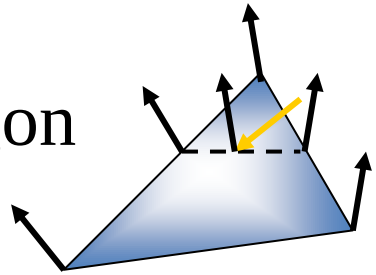
Phong Shading

Bùi Tường Phong
(1942 – 1975)

- Use a single vertex normal for each vertex

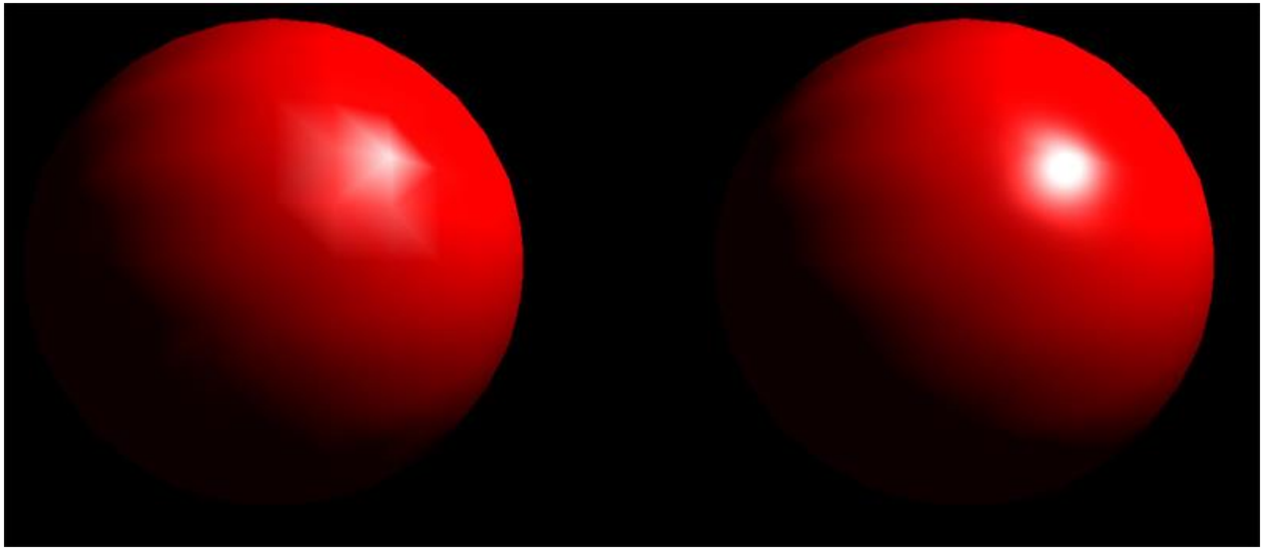


- Interpolate vertex normals across polygon



- Calculate color (by illumination) at each pixel in polygon using the interpolated normal

Phong Shading



Gouraud shading Phong shading

Phong Shading

- Captures highlights much better
  - The interpolated normal at each interior pixel is more accurate representation of true surface normal at each point
  - Higher quality, but needs more computation

- Not to be confused with Phong's illumination model (developed by the same person)

[Demo] Polygon Shading

- Flat & Gouraud shading
  - http://math.hws.edu/graphicsbook/demos/c4/smooth-vs-flat.html

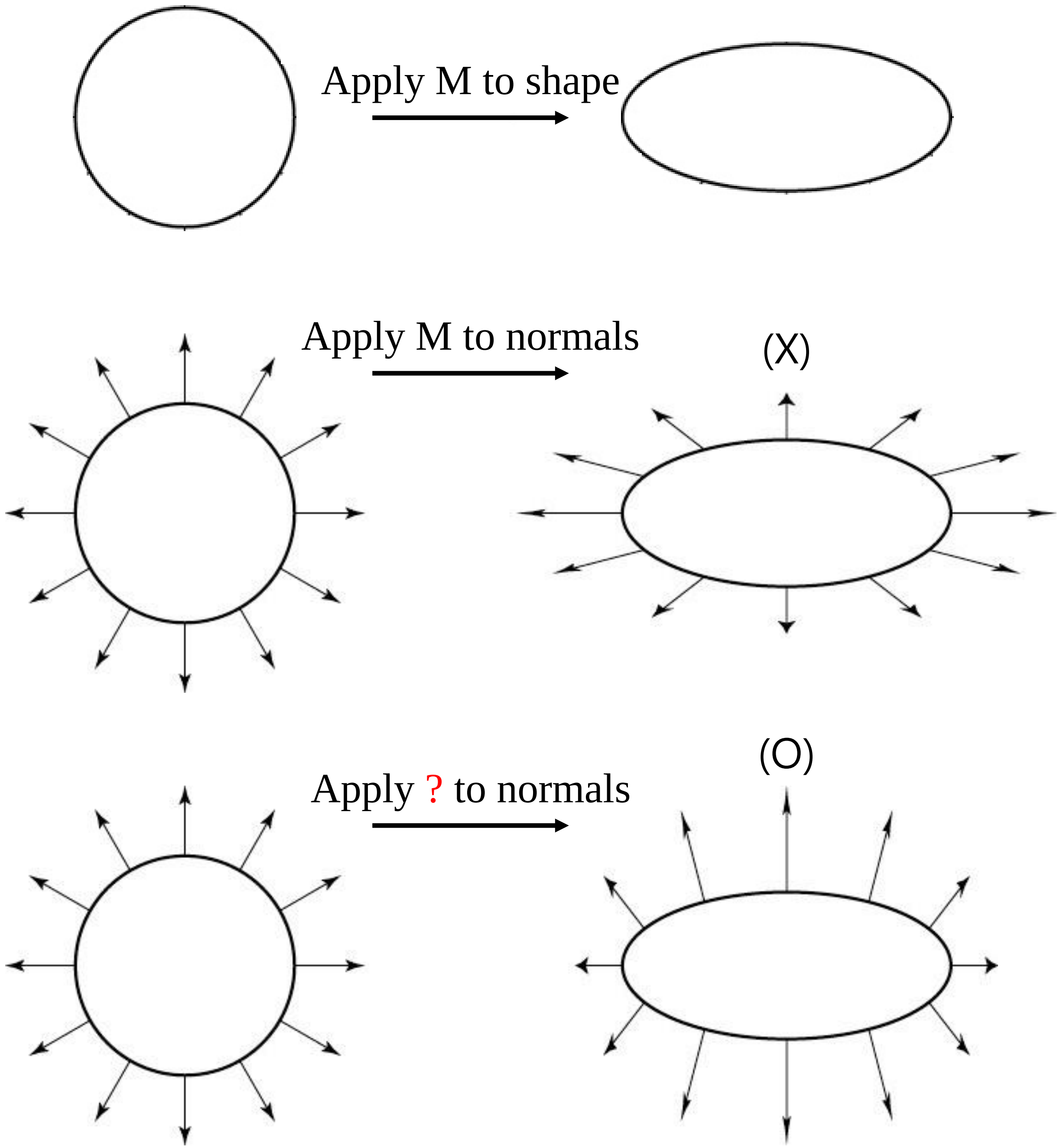
- Gouraud & Phong shading
  - http://www.cs.toronto.edu/~jacobson/phong-demo/

Normal Vector Transformation

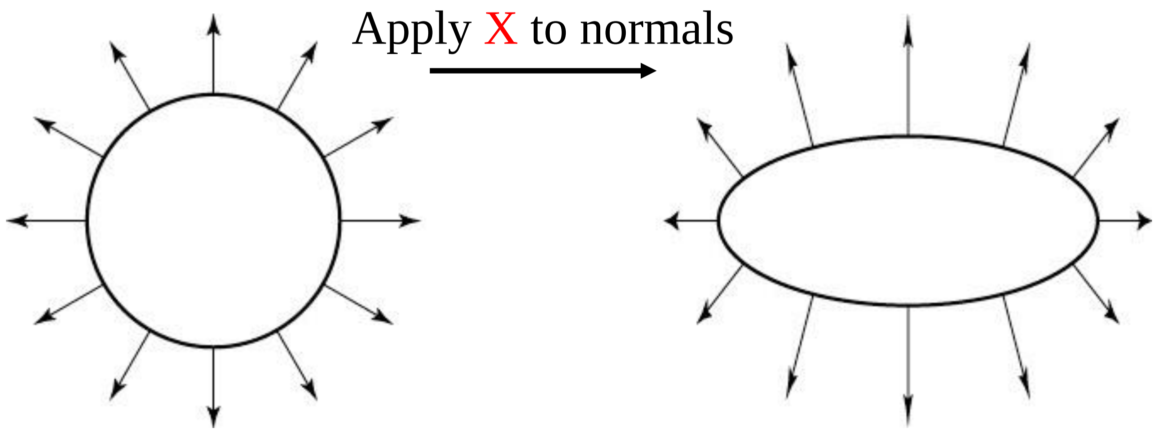
- If a set of points on a surface is transformed by an affine transformation M,

- Tangents are transformed by M.
  - Because the differences of points are transformed by M.

- However, normals should not be transformed by M.
  - Because normals should be perpendicular to tangents.


Apply M to shape
Apply M to normals
(X)
(O)
Apply ? to normals

Normal Vector Transformation



t: tangent vector

n: normal vector

have: $\mathbf{t} \cdot \mathbf{n} = \mathbf{t}^T \mathbf{n} = 0$

want: $M\mathbf{t} \cdot X\mathbf{n} = \mathbf{t}^T M^T X \mathbf{n} = 0$

so set $X = (M^T)^{-1}$

then: $M\mathbf{t} \cdot X\mathbf{n} = \mathbf{t}^T M^T (M^T)^{-1} \mathbf{n} = \mathbf{t}^T \mathbf{n} = 0$

Solution: $X = (M^T)^{-1}$

Lab Session

- Now let's start the lab session.