CSE 167:
Introduction to Computer Graphics
Lecture #7: Illumination

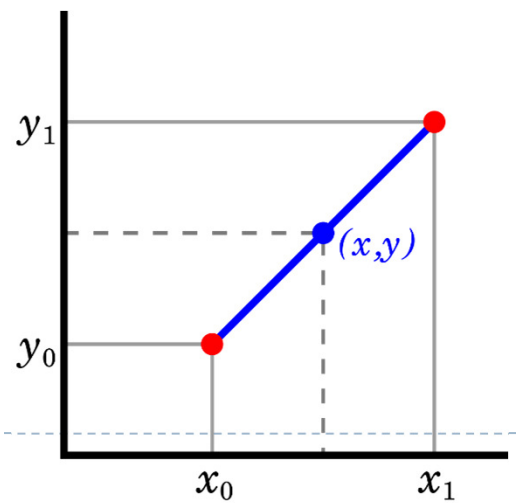
Jürgen P. Schulze, Ph.D.
University of California, San Diego
Fall Quarter 2020

Announcements

- Sunday, October 25 at 11:59pm:
  - Homework Project 1 due
  - Submission on Canvas
  - Follow instructions at end of project write-up
- Next Wednesday at 1pm:
  - Discussion Project 2
- Sunday, November 8th at 11:59pm:
  - Homework Project 2 due
- Extra extra credit option for project 2:
  - Degree completion plan
  - Submission independent of homework project

LERP Function

- LERP = Linear intERPolation
- Can be used with scalars or vectors, and even points
- Useful for smooth transitions

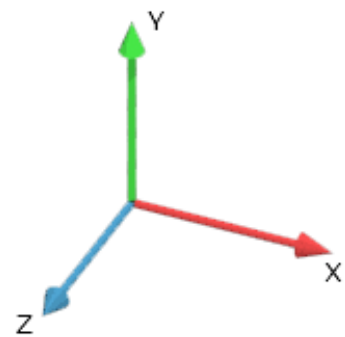


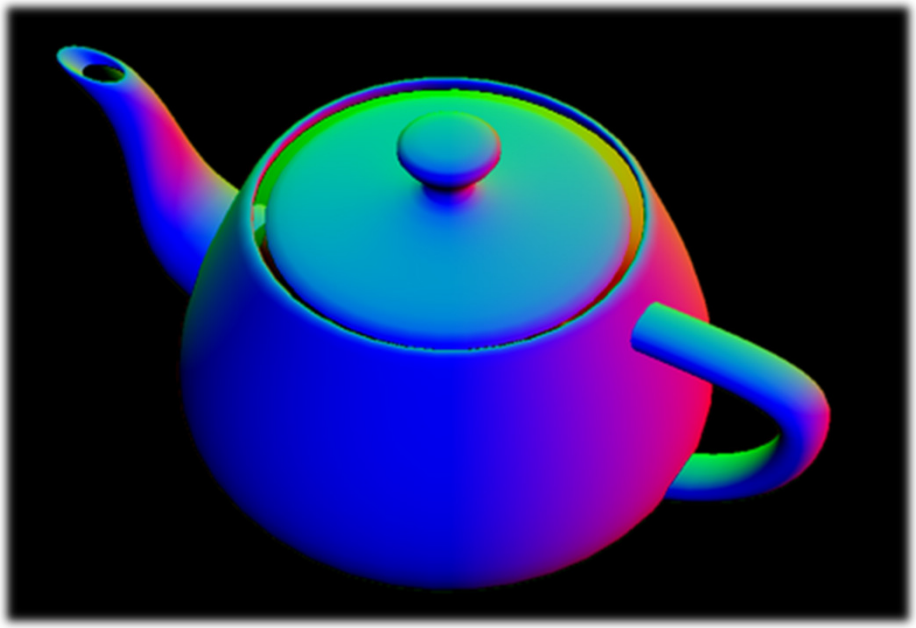

```
// Linear Interpolation
// Also known as "lerp" or "mix"
function lerp (start, end, t) {
  return start * (1 - t) + end * t;
}

// Examples:
lerp(0, 100, 0.5); // 50
lerp(20, 80, 0); // 20
lerp(30, 5, 1); // 5
lerp(-1, 1, 0.5); // 0
lerp(0.5, 1, 0.5); // 0.75
```

Shading

Normal Shading

- Coloring based on surface normal
  - X coordinate maps to Red
  - Y coordinate maps to Green
  - Z coordinate maps to Blue
- Need to map normal range of -1 to +1 to color range of 0.0 to 1.0

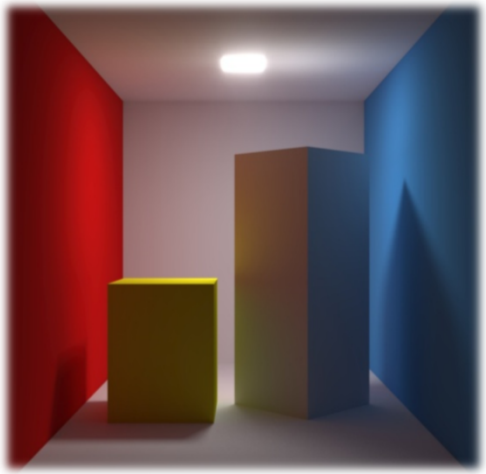




Realistic Shading

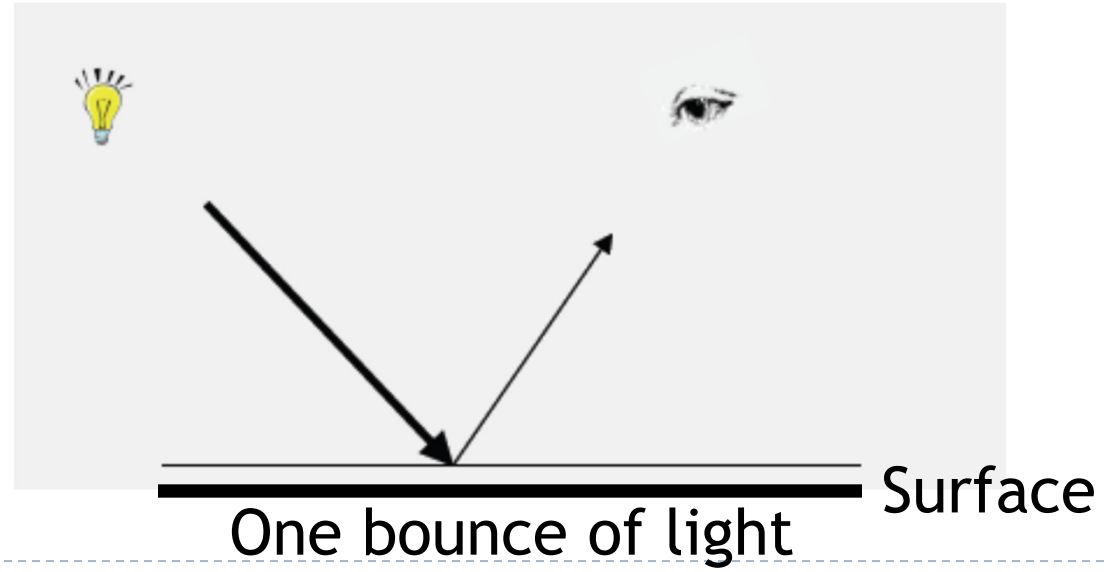
- Compute interaction of light with surfaces
- Requires simulation of physics
- "Global illumination"
  - Multiple bounces of light
  - Computationally expensive, minutes per image
  - Used in movies, architectural design, etc.
- Appearance = Material Definition + Light Sources

Global Illumination



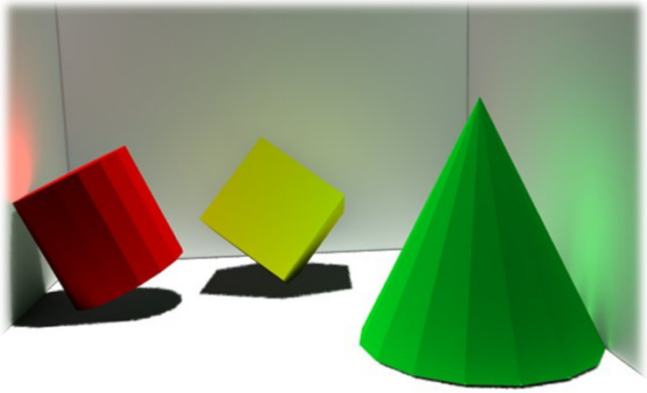
Interactive Applications

- No physics-based simulation
- Simplified models
- Reproduce perceptually most important effects
- Local illumination
  - Only one bounce of light between light source and viewer



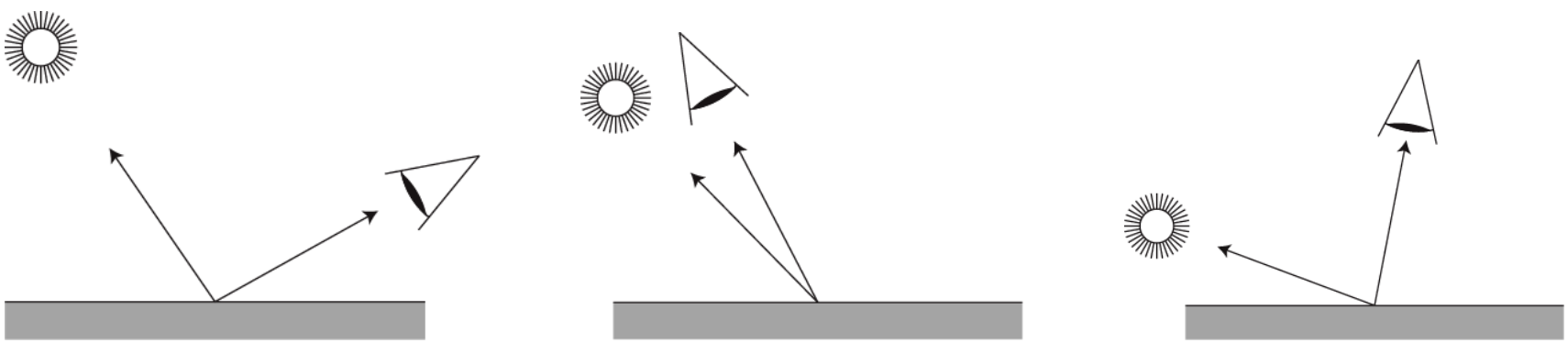
Local Illumination

- Gives material its color
- Light can be reflected by
  - Mirror
  - White wall
  - Glossy metal
  - etc.



Local Illumination

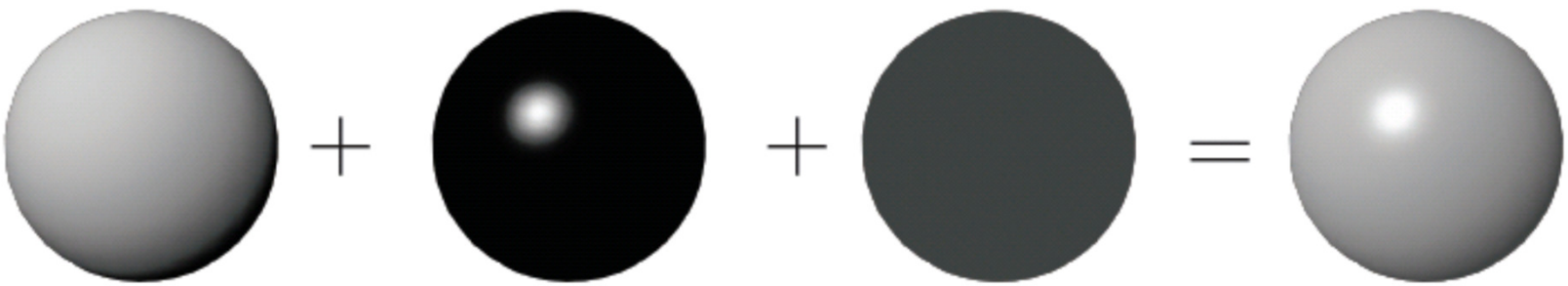
- Model reflection of light at surfaces
  - Assumption: no subsurface scattering
- Bidirectional reflectance distribution function (BRDF)
  - Given light direction, viewing direction, how much light is reflected towards the viewer
  - For any pair of light/viewing directions!



Local Illumination

Simplified model

- Sum of 3 components
- Covers a large class of real surfaces

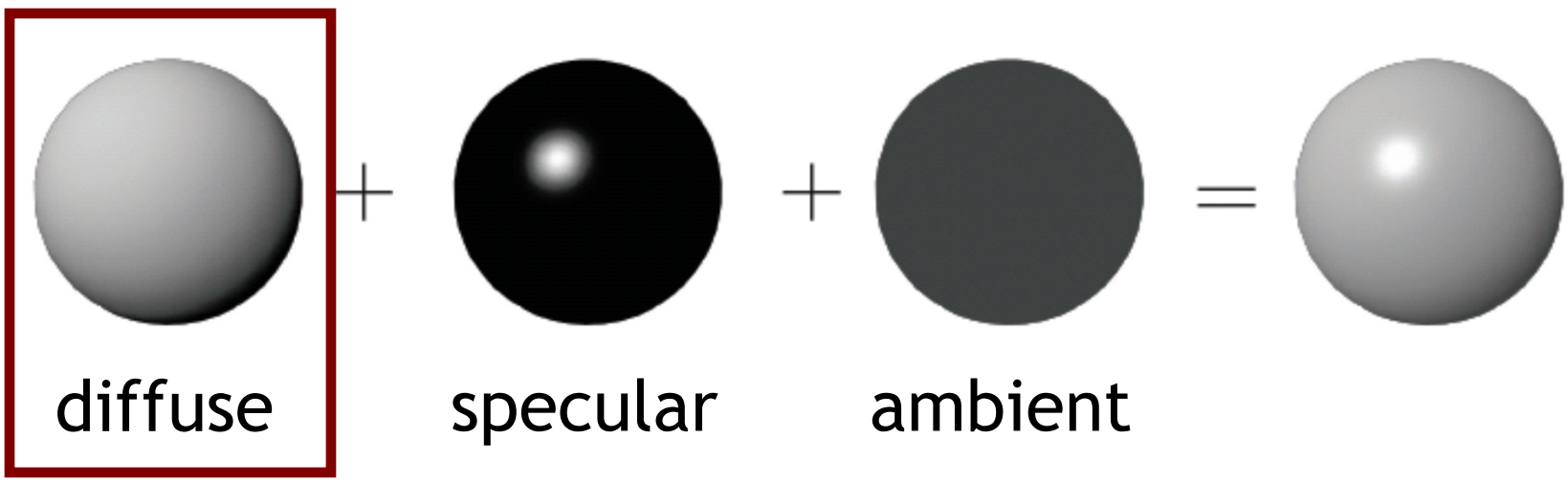


diffuse specular ambient

Local Illumination

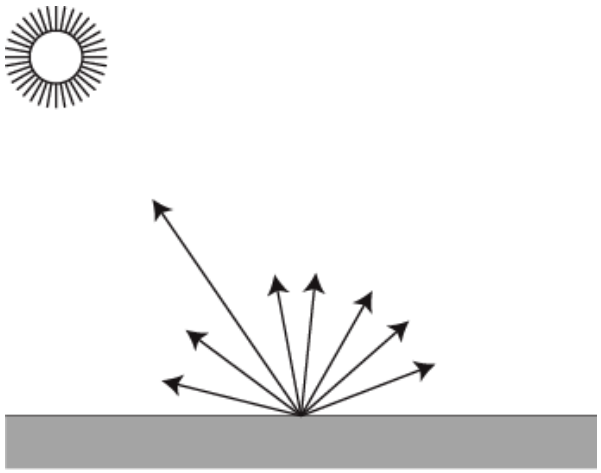
Simplified model

- Sum of 3 components
- Covers a large class of real surfaces



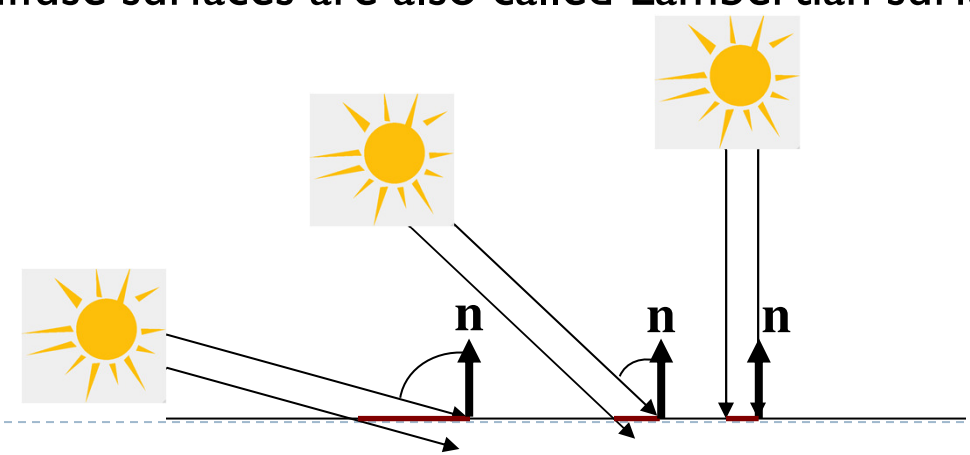
Diffuse Reflection

- Ideal diffuse material reflects light equally in all directions
- View-independent
- Matte, not shiny materials
  - Paper
  - Unfinished wood
  - Unpolished stone



Diffuse Reflection

- Beam of parallel rays shining on a surface
  - Area covered by beam varies with the angle between the beam and the normal
  - The larger the area, the less incident light per area
  - Incident light per unit area is proportional to the cosine of the angle between the normal and the light rays
- Object darkens as normal turns away from light
- Lambert's cosine law (Johann Heinrich Lambert, 1760)
- Diffuse surfaces are also called Lambertian surfaces



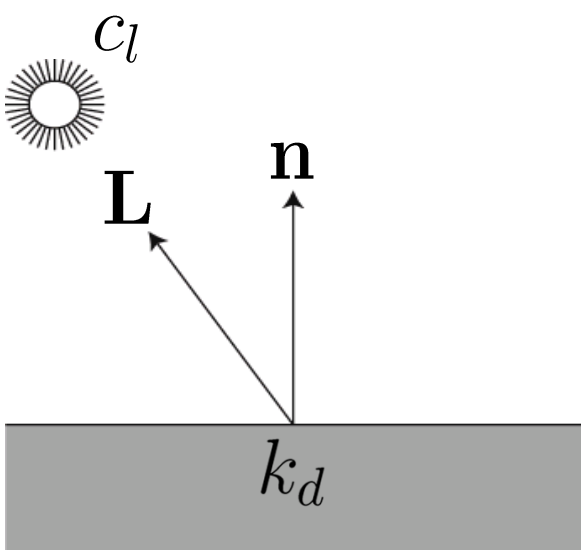
Diffuse Reflection

- Given
  - Unit (normalized!) surface normal **n**
  - Unit (normalized!) light direction **L**
  - Material diffuse reflectance (material color) k_d
  - Light color (intensity) c_l

- Diffuse color c_d is:

$$c_d = c_l k_d \underbrace{(\mathbf{n} \cdot \mathbf{L})}$$

Proportional to cosine between normal and light



Diffuse Reflection

Notes

- Parameters k_d, c_l are r,g,b vectors (colors)
- Need to compute r,g,b values of diffuse color c_d separately
- Parameters in this model have no precise physical meaning
  - c_l: intensity and color of light source
  - k_d: fraction of reflected light, material color

Diffuse Reflection

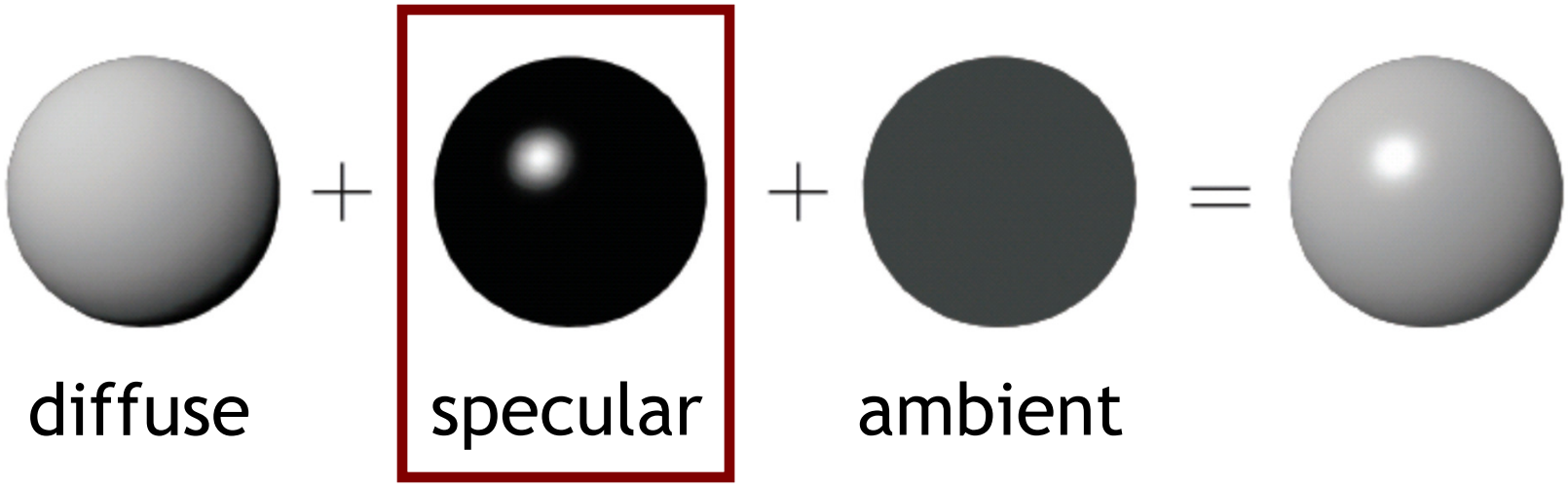
- Provides visual cues
  - Surface curvature
  - Depth variation

Lambertian (diffuse) sphere under different lighting directions

Local Illumination

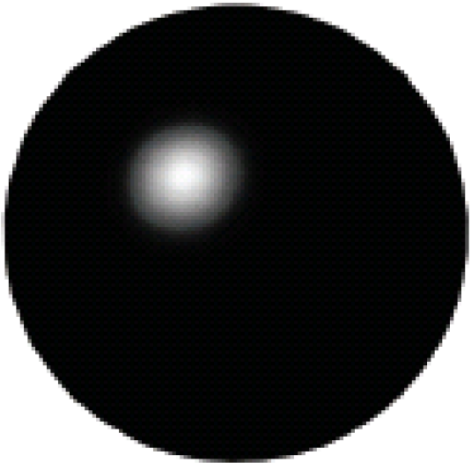
Simplified model

- Sum of 3 components
- Covers a large class of real surfaces



Specular Reflection

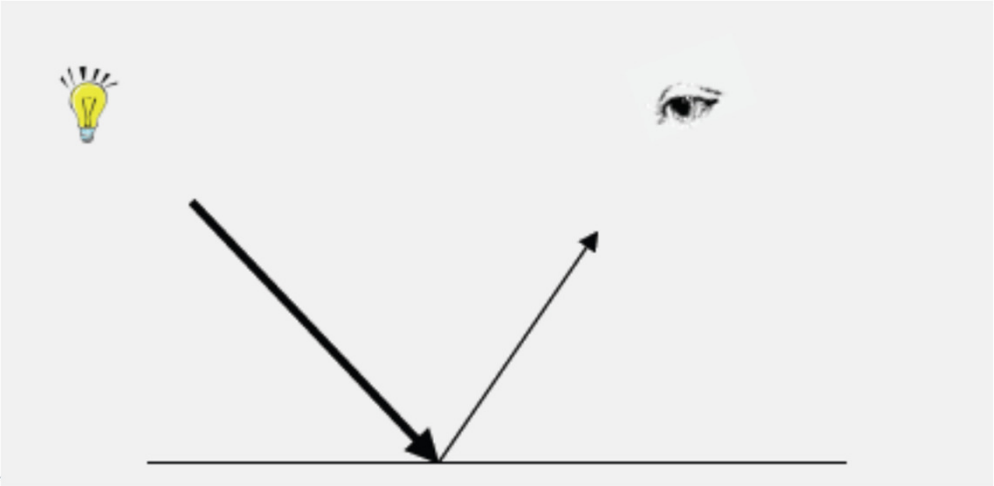
- Shiny surfaces
  - Polished metal
  - Glossy car finish
  - Plastics
- Specular highlight
  - Blurred reflection of the light source
  - Position of highlight depends on viewing direction



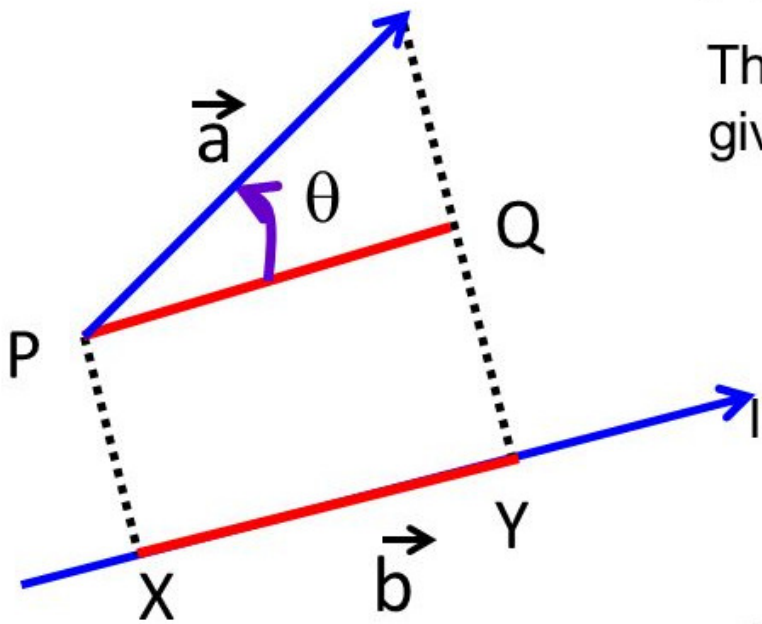
Specular highlight

Specular Reflection

- Ideal specular reflection is mirror reflection
  - Perfectly smooth surface
  - Incoming light ray is bounced in single direction
  - Angle of incidence equals angle of reflection



Projection of vector on another vector



Projection of $\vec{a}$ on $\vec{b}$ is XY

The projection of **a** onto **b** will be given by:

$$proj_b a = |a| \cos\theta \frac{\mathrm{b}}{|\mathrm{b}|}$$

In summary, the proj_a b has length $|a|\cos\theta$, and direction $\frac{b}{|b|}$

It is called the scalar component of **a** in the direction of **b**

Law of Reflection

- Angle of incidence equals angle of reflection

We can derive the reflection vector R:

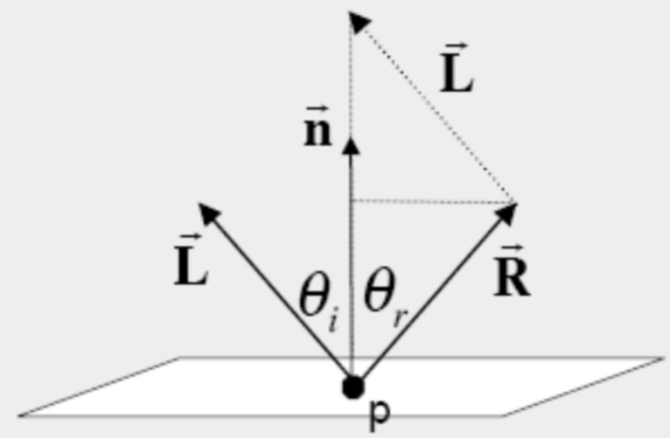
Using these equations:

$$proj_b a = |a| \cos\theta \frac{\mathrm{b}}{|\mathrm{b}|}$$

$$\cos\theta = \left(\frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}||\mathbf{b}|} \right)$$

$$\vec{\mathbf{R}} + \vec{\mathbf{L}} = 2\cos\theta\ \vec{\mathbf{n}} = 2\left(\vec{\mathbf{L}} \cdot \vec{\mathbf{n}}\right)\vec{\mathbf{n}}$$

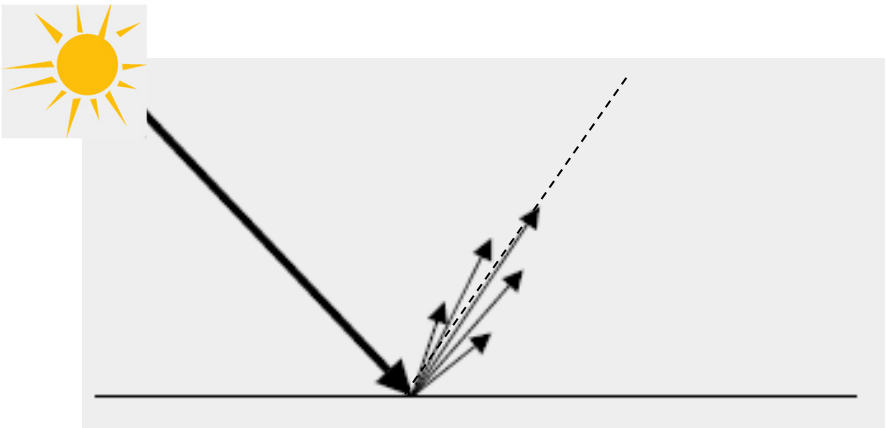
$$\vec{\mathbf{R}} = 2\left(\vec{\mathbf{L}} \cdot \vec{\mathbf{n}}\right)\vec{\mathbf{n}} - \vec{\mathbf{L}}$$



$$\theta_r = \theta_i$$

Specular Reflection

- Many materials are not perfect mirrors
  - Glossy materials



Glossy teapot

Glossy Materials

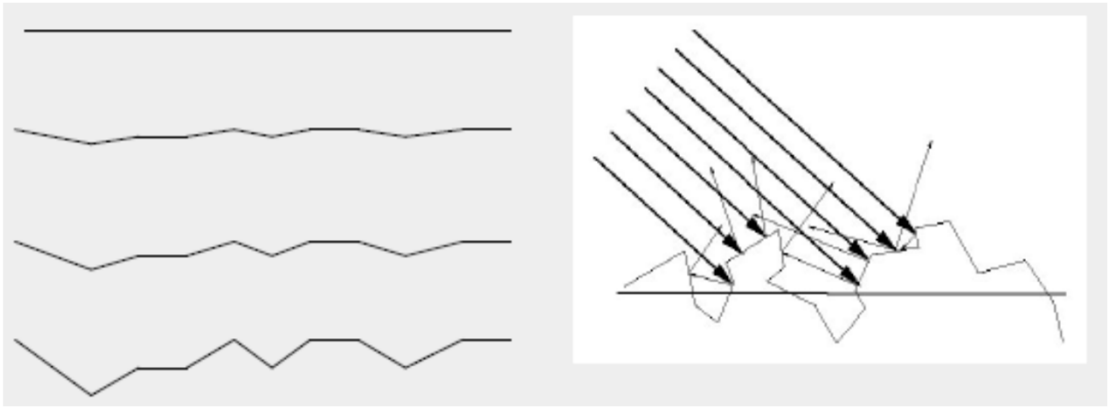
- Assume surface composed of small mirrors with random orientation (micro-facets)
- Smooth surfaces
  - Micro-facet normals close to surface normal
  - Sharp highlights
- Rough surfaces
  - Micro-facet normals vary strongly
  - Blurry highlight

Polished

Smooth

Rough

Very rough

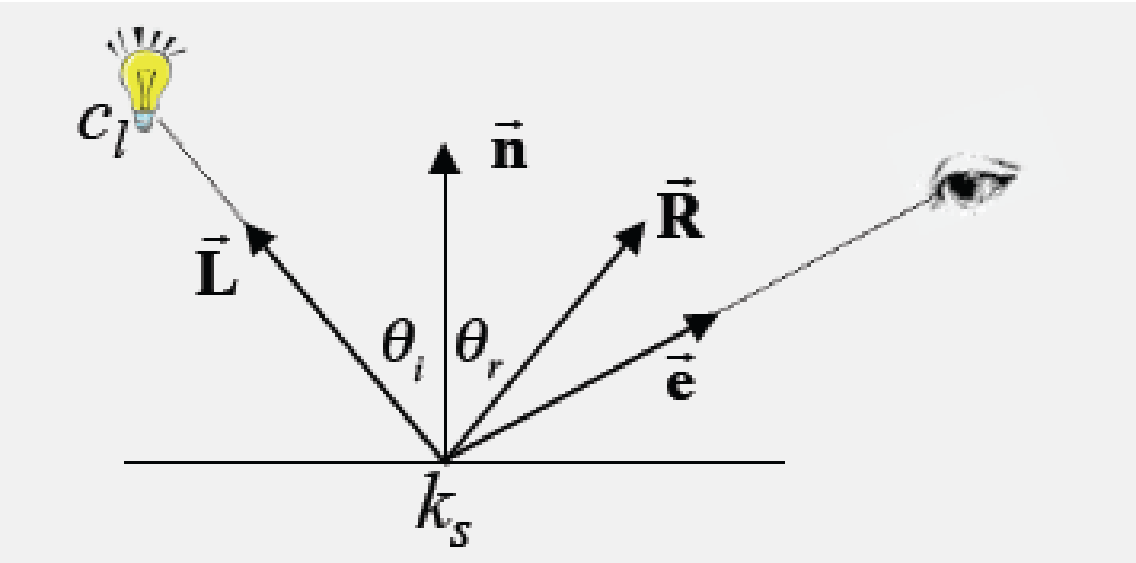


Glossy Surfaces

- Expect most light to be reflected in mirror direction
- Because of micro-facets, some light is reflected slightly off ideal reflection direction
- Reflection
  - Brightest when view vector is aligned with reflection
  - Decreases as angle between view vector and reflection direction increases

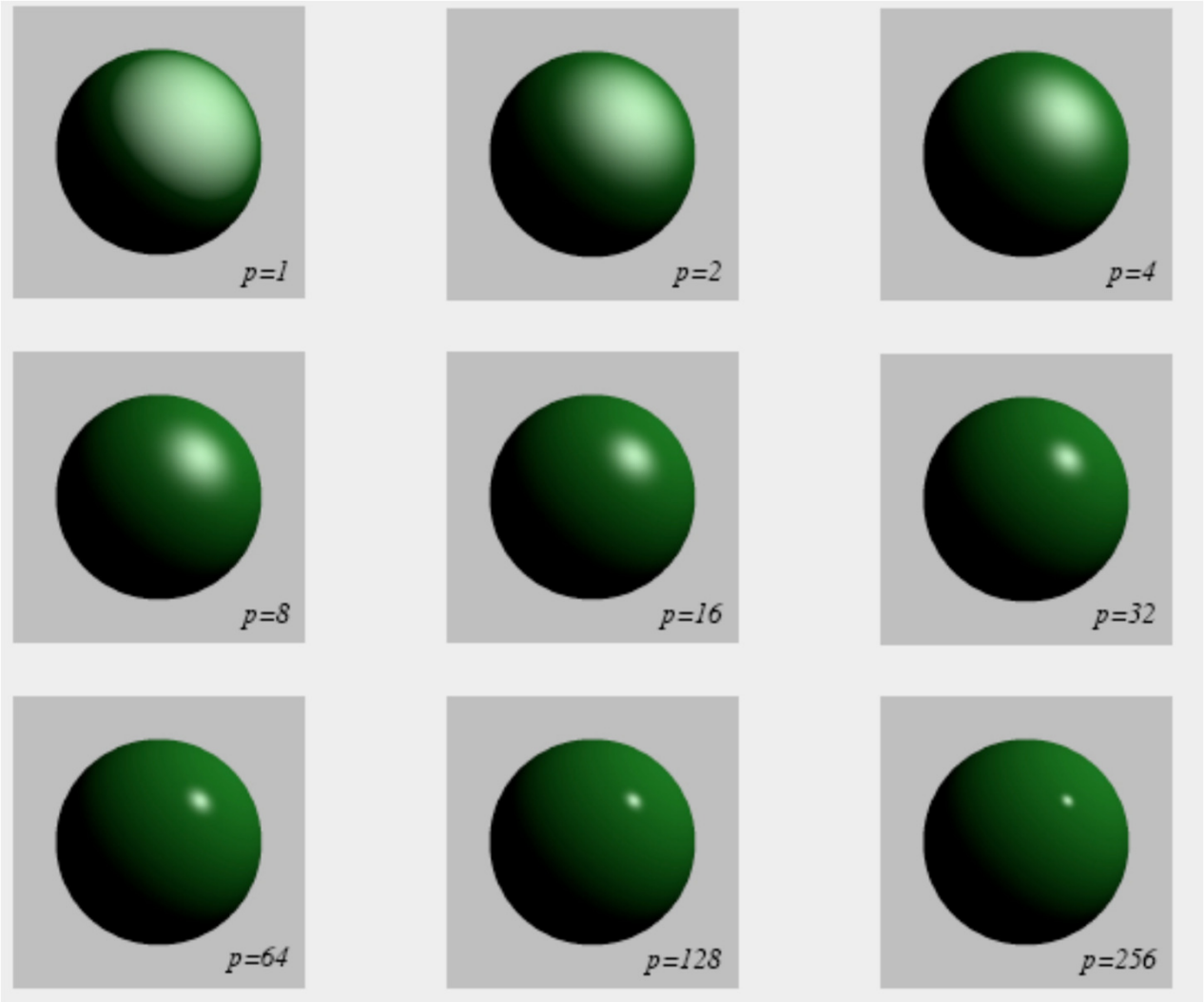
Phong Shading Model

- Developed by Bui Tuong Phong in1973
- Specular reflectance coefficient (color) k_s
- Phong exponent p
  - Greater p means smaller (sharper) highlight



$$c = k_s c_l \left(\mathbf{R} \cdot \mathbf{e}\right)^p$$

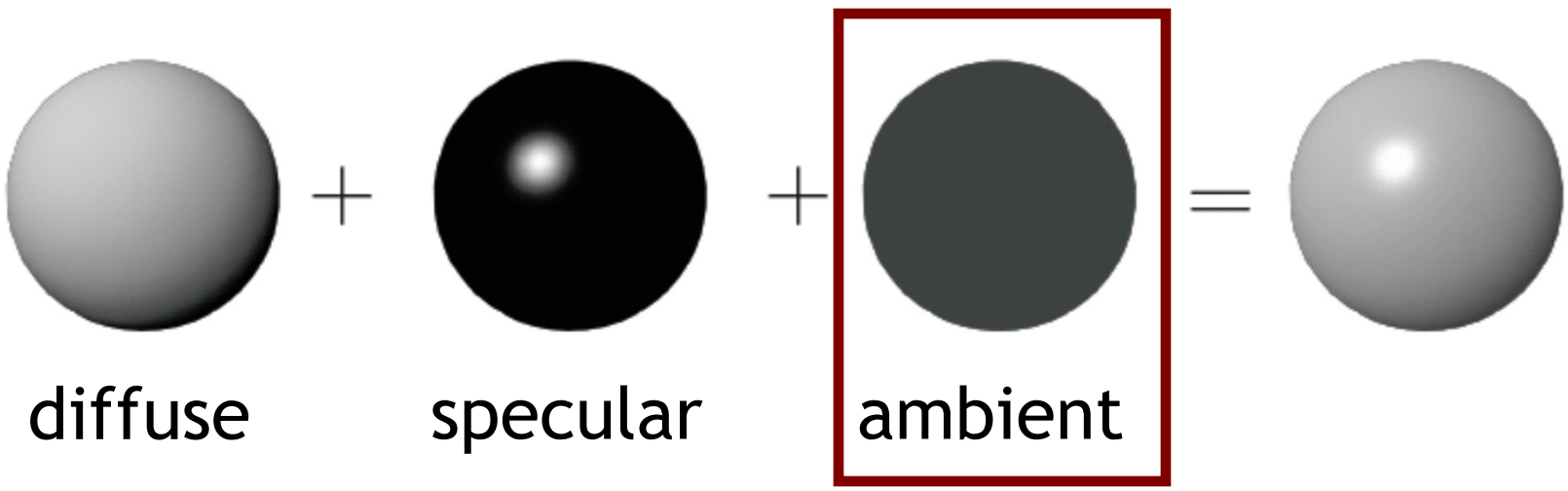
Phong Shading Model



Local Illumination

Simplified model

- Sum of 3 components
- Covers a large class of real surfaces



Ambient Light

- In real world, light is bounced all around scene
- Could use global illumination techniques to simulate
- Simple approximation
  - Add constant ambient light at each point: $k_a c_a$
  - Ambient light color: c_a
  - Ambient reflection coefficient: k_a
- Areas with no direct illumination are not completely dark

Complete Phong Shading Model

- Phong model supports multiple light sources
- All light colors *c* and material coefficients *k* are 3-component vectors for red, green, blue

$$c = \sum_i c_{l_i}(k_d(L_i \cdot n) + k_s(R \cdot e)^p + k_a)$$

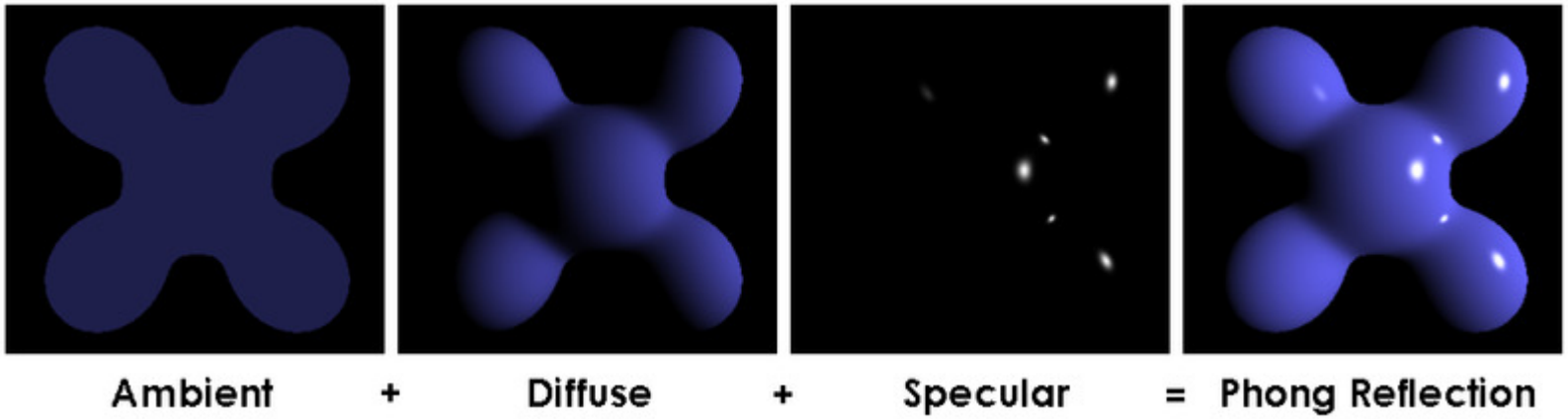


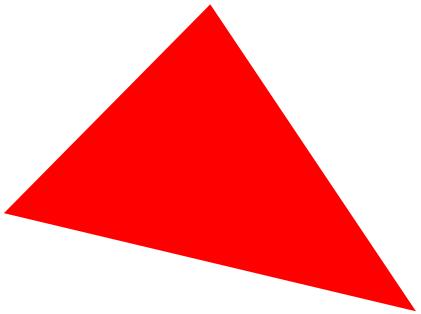
Image by Brad Smith

Types of Shading

- Per-triangle
- Per-vertex
- Per-pixel

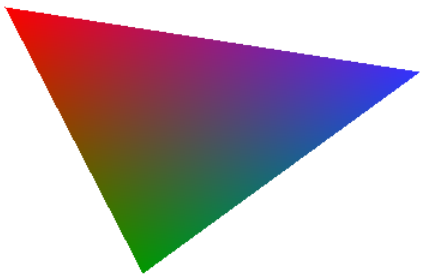
Per-Triangle Shading

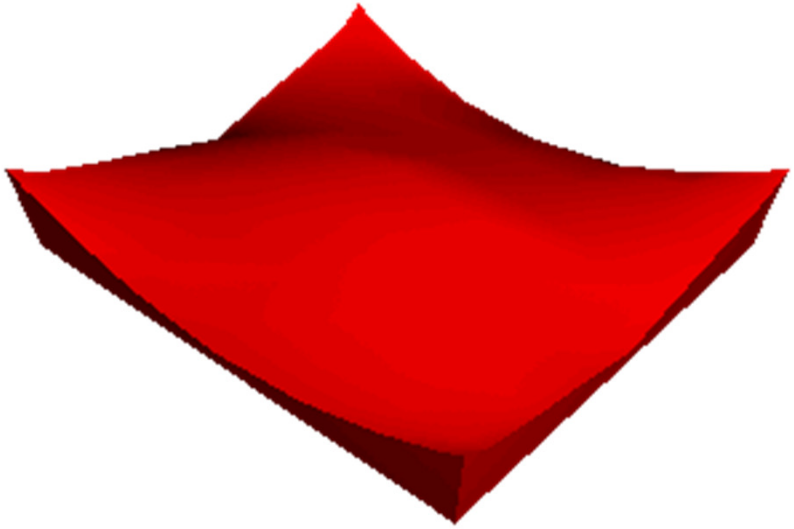
- A.k.a. *flat shading*
- Evaluate shading once per triangle, based on normal vector
- Advantage
  - Fast
- Disadvantage
  - Faceted appearance



Per-Vertex Shading

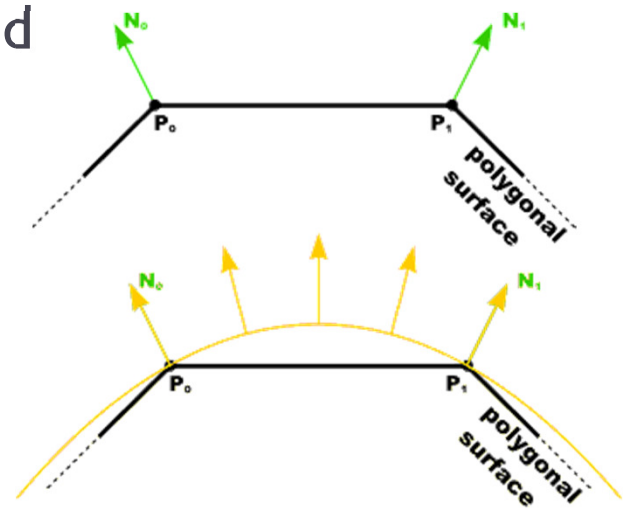
- Known as *Gouraud shading* (→ Henri Gouraud, 1971)
- Interpolates vertex colors across triangles
- Advantages
  - Fast (no less work in fragment shader)
  - Smoother surface appearance than with flat shading
- Disadvantage
  - Problems with small highlights





Per-Pixel Shading

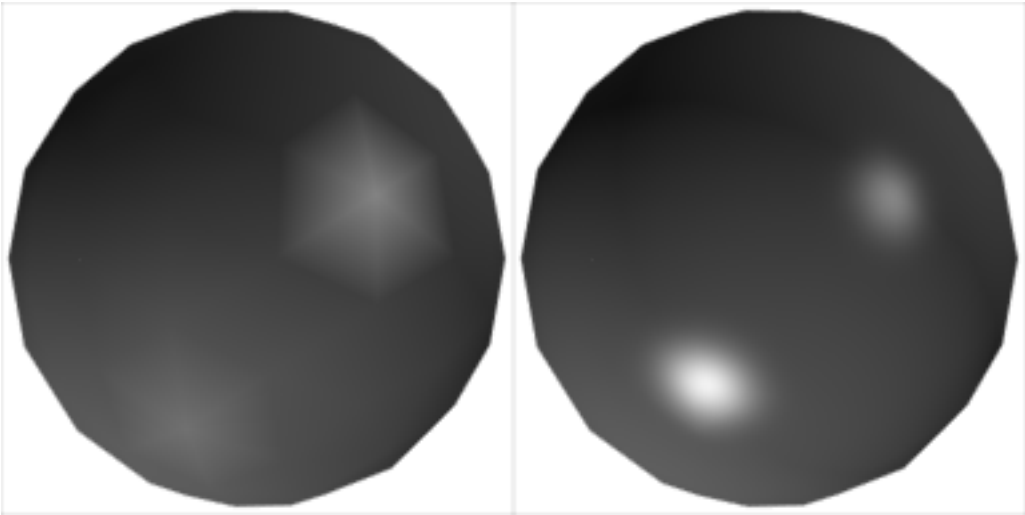
- A.k.a. *Phong Interpolation* (not to be confused with *Phong Illumination Model*)
  - Rasterizer interpolates normals (instead of colors) across triangles
  - Illumination model is evaluated at each pixel
  - Simulates shading with normals of a curved surface
- Advantage
  - Highest rendering quality
- Disadvantage
  - Slow



Source: Penny Rheingans, UMBC

Gouraud vs. Per-Pixel Shading

- Gouraud shading has problems with highlights when polygons are large
- More triangles improve the result, but reduce frame rate
- Video: https://www.youtube.com/watch?v=Fl5i-UnlQps&feature=youtu.be



Per-Vertex (Gouraud) Per-Pixel

Summary

- Per-pixel shading looks best and is only slightly more computationally expensive
- On slower GPUs Gouraud shading may make sense (e.g., in OpenGL ES on older mobile devices)
- In CSE 167 we always use per-pixel shading