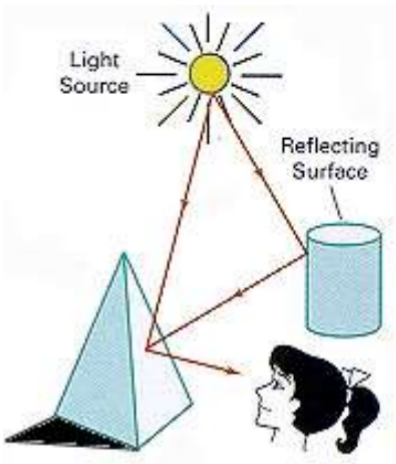
Lecture7 Illumination

1. Introduction

- Illumination (lighting) is one component for creating graphics images
- An illumination model is used to determines the color of a surface point by simulating some light attributes
- Using illumination models, we can simulate **shading**, **reflection**, and **refraction** of light, comparable to what we see in the real world.

Illumination Model



- Light Source
  - Color
  - Position
  - Direction
- Object Properties
  - Geometry
  - Material
- Observer (camera)

2. Basic Light Sources

Ambient light source

- simulates the effect that even in a dark environment there is usually still some light somewhere giving objects some color
- a very simple model of global illumination, which uses a small **constant light (color)** added to the final color of objects
- Property: no spatial or directional characteristics

Specification

- constant color $[R, G, B]$
- intensity I_a

Directional light source

- emits light from an **infinite distance**
- Example: the sun
- Property: All the rays emitted are parallel, and thus can be defined by a vector

Specification

- constant color $[R, G, B]$
- intensity I_{source}
- direction vector v_d

Point(positional) light source

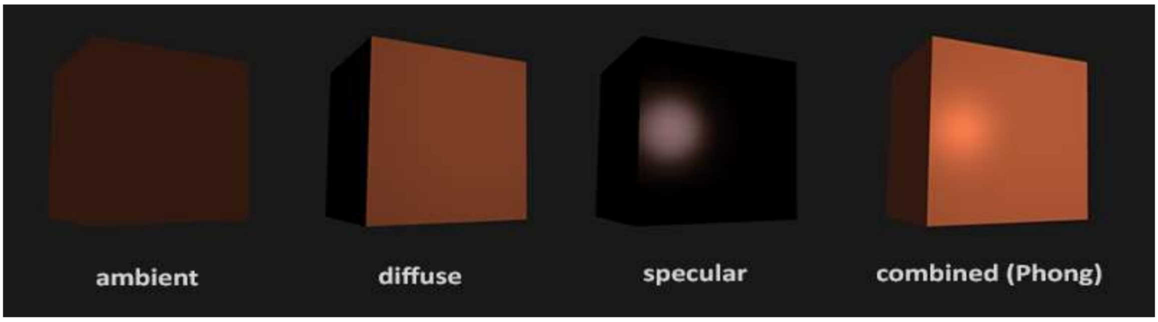
- emits light from a particular location
- Property: emitting rays in all directions

Specification

- constant color $[R, G, B]$
- intensity I_{source}
- location (x_0, y_0, z_0)

> The following pseudo-codes define 3 light sources:
>
> ```
> struct {
>     float     intensity;
>     Vector3d  color;
>     Vector3d  dir;
> } lightA;
> ```
>
> ```
> struct {
>     float     intensity;
> } lightB;
> ```
>
> ```
> struct {
>     float     intensity;
>     Vector3d  color;
>     Vector3d  loc;
> } lightC;
> ```
>
> Which one defines an ambient light, a point light, and a directional light?
>
> - light A: directional light
> - light B: ambient light
> - light C: point light

3. Phone Illumination Model



$$I = k_a I_a + \sum_{\substack{\text{for each} \\ \text{light } s}} k_d I_s cos\theta + \sum_{\substack{\text{for each} \\ \text{light } s}} k_s I_s cos^n \phi$$

It contains three parts

- ambient reflection
- diffuse reflection
- specular reflection

Ambient reflection

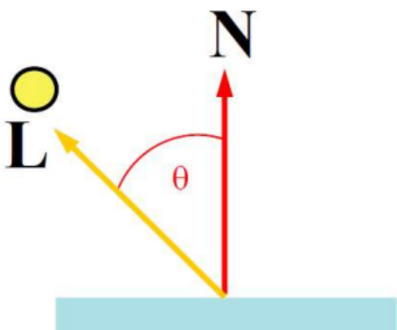
Ambient reflection roughly tells how much of the ambient light is reflected

$$A = k_a I_a$$

- k_a: the ambient reflection coefficient $k_a \in [0, 1]$
- I_a: the ambient light color or intensity

Diffuse reflection

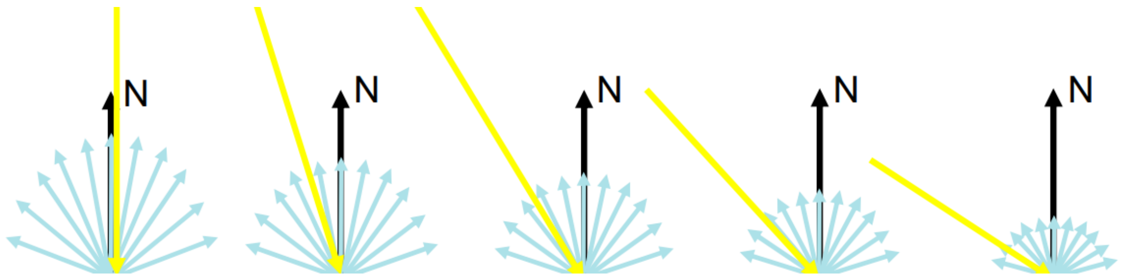
Diffuse reflection is based on Lambert's (cosine) law



$$D = \sum_{\substack{\text{for each} \\ \text{light } s}} k_d I_s cos\theta$$

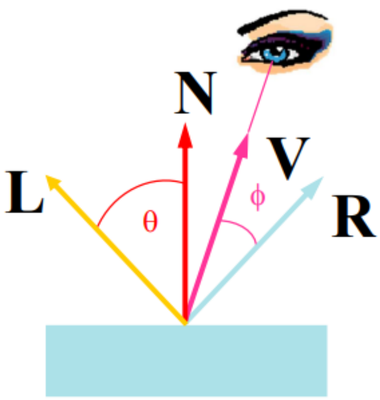
- k_d: the diffuse reflection coefficient $k_d \in [0, 1]$

- I_s: the directional/point light color or intensity
- θ: the angle of incidence(入射角), i.e., the angle between the **surface normal** and the **vector to the light source**



Specular reflection

Specular reflection accounts for the highlight in a shiny, glossy surface (very smooth surface)



$$S = \sum_{\substack{\text{for each} \\ \text{light } s}} k_s I_s cos^n \phi$$

- k_s: the specular reflection coefficient $k_s \in [0, 1]$
- I_s: the directional/point light color or intensity
- n: specular exponent (specular-reflection parameter)
    - the bigger the n, the faster the highlight falls off
- ϕ: the angle between the viewing direction V and the reflected vector R

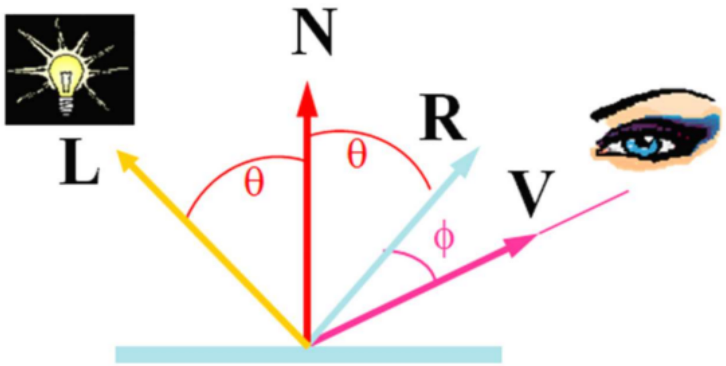
4. Computation with Phone Illumination Model

Calculation Steps

Given information

- Position of the point light source: P
- Location of viewer: Q
- Point on the surface: S

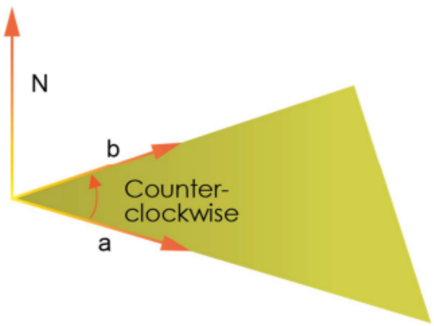
We need to calculate four vectors (unit vector)



- Lighting vector $\vec{L}$: a vector from a point on the surface towards a light source
  - $\vec{L} = \frac{P-S}{|P-S|}$
- Viewing vector $\vec{V}$: a vector from a point on the surface towards the viewer
  - $\vec{V} = \frac{Q-S}{|Q-S|}$
- Normal vector $\vec{N}$: a vector perpendicular to the surface
  - The calculation of normal vector is depending on the geometry shape
- Reflected vector $\vec{R}$: the image of the lighting vector L reflected off the surface
  - $R_{vec} = 2(\vec{N} \cdot \vec{L}) \cdot \vec{N} - \vec{L}$
  - $\vec{R} = \frac{R_{vec}}{|R_{vec}|}$

Computation of surface normal vectors

Polygonal surfaces



- use cross product of 2 vectors lying on a facet (right-hand rule)

Implicit surface

$$f(x,y,z) = 0 \rightarrow N(x,y,z) = \pm \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial z} \end{bmatrix}$$

Parametric surface

$$N(u,v) = \frac{\partial P}{\partial u} \times \frac{\partial P}{\partial v} = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial y}{\partial u} & \frac{\partial z}{\partial u} \end{bmatrix} \times \begin{bmatrix} \frac{\partial x}{\partial v} & \frac{\partial y}{\partial v} & \frac{\partial z}{\partial v} \end{bmatrix}$$

Plane

$$f(x, y, z) = Ax + By + Cz + D = 0$$

One normal vector is $\vec{N} = \pm[A \;\; B \;\; C]$ (unified it)

Sphere

Sphere centered at $C = (x_0, y_0, z_0)$ with radius r:

$$f(x, y, z) = r^2 - (x - x_0)^2 - (y - y_0)^2 - (z - z_0)^2$$

The outward normal vector is $\vec{N} = (x, y, z) - (x_0, y_0, z_0)$