Lecture 07 – Lighting (Chapter 17)

I. Basic definitions

A. Illumination model (lighting model) – calculations of colors on surfaces of objects based on illumination.

B. Surface rendering method – color calculations (procedure) from an illumination model to determine all projected pixel colors of a surface. (interpolation w/ scanline)

C. Photorealism involves light reflections, transparency, texture, shadowing, etc. We will

i. ~~see~~ very basic models of physics

a. Light is an interaction of electromagnetic energy with an object.

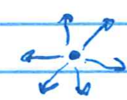
b. Involves properties of materials, positions of objects, and features of light sources
(shiny/dull/transparent) (shape, color, positions)

ii. Goal approximate physics realistically + efficiently.

II. Light Sources – objects that emit light (radiant energy)

A. Need position, color, direction, shape

B. Point light source (simplest model)

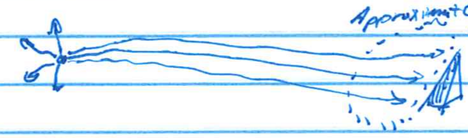


i. Single Color (RGB)

ii. Position (world space)

iii. Approximation for lightsource that is much smaller than objects in scene. Or large far away object.

C. Infinitely-Distant Light Sources (Directional light)



i. Single Color

ii. Direction (parallel light rays)

iii. Large far away light source

D. Radial Intensity Attenuation

i. Idea – close objects should be affected more by light vs. farther light sources

ii. Proportional intensity to $1/d_l$ or $1/d_l^2$ where d_l is distance from light source.

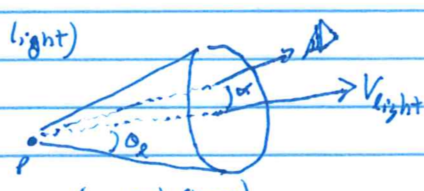
iii. Attenuation function:

$$f_{radatten}(d_l) = \begin{cases} \frac{1}{a_0 + a_1 d_l + a_2 d_l^2} & \text{for point light} \\ 1.0 & \text{for directional light.} \end{cases}$$

where a_0, a_1, a_2 are coefficients

Discuss setting a_0, a_1, a_2

E. Spot light (point + directional light)



i. Single Color

ii. Position

iii. Direction V_{light}

iv. Angular limit θ_l

v. $V_{obj} \cdot V_{light} = \cos\alpha$, object in the spotlight if $\cos\alpha \geq \cos\theta_l$

F. Angular Attenuation – away from center of spotlight

$$f_{angatten} = \begin{cases} 1.0 & \text{if not a spotlight} \\ 0.0 & \text{if } V_{obj} \cdot V_{light} = \cos\alpha < \cos\theta_l \text{ (outside cone)} \\ (V_{obj} \cdot V_{light})^{a_l} & \text{otherwise} \quad a_l \text{ is constant.} \end{cases}$$

Ask how to simulate light strip/neon lights shaped nonuniformly.

III. Surface lighting effects

A. Part of light is reflected from material, other part is absorbed (transparency, part is transmitted)

Material properties are vital - shiny vs. dull etc.

B. Scattered light is called diffuse reflection (equally bright from any direction), ex. carpet.

"color" is the diffuse reflection w/ white light

C. Concentrated reflections are called specular reflections - based on ~~lighting~~ viewing direction

"highlighting" effect, eg. whiteboards.

D. Background light or ambient light - illumination from reflected light in scene

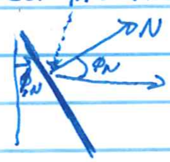
rays off cylinder are part of ambient light.

IV. Basic Illumination Models

A. Ambient light - approximated by global brightness level

i. Designated by intensity I_a

B. Diffuse Reflection - based on Lambert's cosine law - amount of reflection from any small surface area in a direction ϕ_N is proportional to $\cos \phi_N$



$$I = \frac{\cos \phi_N}{dA \cos \phi_N} \Rightarrow \text{constant.} \quad \text{(ideal model)}$$

i. Let K_d be constant (diffuse-reflection coefficient) in [0, 1]

K_d close to 1 yields more reflectivity.

ii. Ambient contribution is $K_d I_a = I_{amb\,diff}$

iii. Light Intensity is based on orientation of surface

brighter → less bright

angle of incidence θ is angle between light direction and normal

$$I_{l,\,incident} = I_l \cos\theta$$

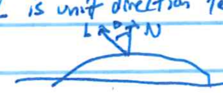
and

$$I_{l,\,diff} = k_d I_l \cos\theta$$

iv. Also can incorporate faces not facing light direction

$$I_{l,\,diff} = \begin{cases} k_d I_l (N \cdot L) & \text{if } N \cdot L > 0 \\ 0.0 & \text{else} \end{cases}$$

where L is unit direction to light source.

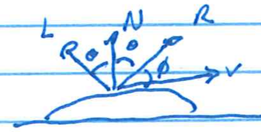


v. Total:

$$I_{diff} = \begin{cases} k_a I_a + k_d I_l (N \cdot L) & \text{if } N \cdot L > 0 \\ k_a I_a & \text{if } N \cdot L \le 0 \end{cases}$$

where k_a is ambient reflection coefficient (per surface)

C. Specular reflection and the Phong Model



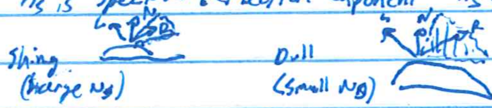
i. R is ideal specular reflection direction

V is view direction

ϕ is viewing angle relative to R

ii. Phong Model by (Phong Bui Thuong) has intensity relative to $\cos^{n_s}\phi$ where

n_s is specular reflection exponent, $n_s \geq 1$



iii. Specular reflection coefficient $w(\theta)$ is based on material and Fresnel's law of reflection

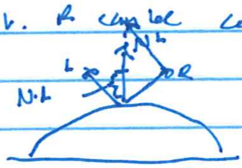
$$I_{l_{spec}} = W(\theta) I_l \cos^{n_s} \phi$$

however $w(\theta)$ is practically constant, k_s between 0 and 1

because V and R are unit vectors $\cos\phi$ is $V \cdot R$

so:
$$I_{l_{spec}} = \begin{cases} k_s I_l (V \cdot R)^{n_s} & \text{if } V \cdot R > 0 \text{ and } N \cdot L > 0 \\ 0.0 & \text{if } V \cdot R \leq 0 \text{ or } N \cdot L \leq 0 \end{cases}$$

iv. R can be computed from N and L



$R + L = 2(N \cdot L) N$ so $R = 2(N \cdot L) N - L$

v. In simplified Phong model you use halfway vector H between L and V as approximation of R $\quad H = \frac{L+V}{|L+V|}$. Not as realistic but faster.

D. Combined:

$$I = I_{diff} + I_{spec} = k_a I_a + k_d I_l (N \cdot L) + k_s I_l (N \cdot H)^{n_s} + I_{emiss}$$

(emission color)

E. Multiple light sources

$$I = k_a I_a + \sum_{l=1}^{n} I_l [k_d (N \cdot L) + k_s (N \cdot H)^{n_s}] + I_{emiss}$$

(sum of diffuse + specular of each light source.)

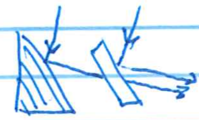
F. With attenuation:

$$I = I_{ambdiff} + \sum_{l=1}^{n} f_{l_{radatten}} f_{l_{ang\,atten}} (I_{l_{diff}} + I_{l_{spec}}) + I_{emiss}$$

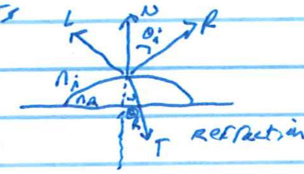
G. With color: I_a, I_l intensity values and coefficients can be separated into RGB components also can have base surface color included. Can be applied to all color models as well.

V Transparent Surfaces - can see behind

A. Translucent objects transmit light through object



Diffuse component is important. but extremely complex. So basic techniques are limited to specular effects



B. Light Refraction - r

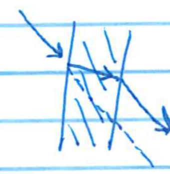
i. Refraction path is how light travels through materials different because speed of light is different between materials

ii. Angle of refraction θ_r w/ respect to Normal

Index of refraction material property

iii. Snell's law to determine angle of refraction

$$\sin \theta_r = \frac{n_i}{n_r} \sin \theta_i$$



iv. Refraction is complex

a. In anisotropic materials - double refraction (2 rays)

b. Refraction occurs twice

v. Transmission vector $T = \left(\frac{n_i}{n_r} \cos\theta_i - \cos\theta_r\right) N - \frac{n_i}{n_r} L$

however time consuming!

vi. Basic transparency model - ignore ~~transmission~~ refraction

$$I = (1-K_t) I_{refl} + K_t I_{trans}$$

where I_{trans} is transmitted light, ~~and~~ I_{refl} is reflected light and K_t is transparency coefficient.

$(1-K_t)$ is opacity.

VI. Atmospheric Effects, eg fog

i. $f_{atmo}(d) = e^{-\rho d}$ or $e^{-(\rho d)^2}$ where d is distance to view and ρ is constant. higher value mutes brightness.

ii. w/ color in atmosphere

$$I = f_{atmo}(d) I_{obj} + \{1 - f_{atmo}(d)\} I_{atmo}$$

VII. Shadows → beyond scope. ~~Use~~ will learn global illumination techniques in chapter 21.

VIII. Halftone/Dithering - producing color tones with limited color intensities (fooling eyes)

example newspapers - small black circles of various radius to produce grays.

outside scope of course.

IX. Polygon Rendering Methods (surface rendering) - goal limit expensive computations of illuminations

A. Constant-intensity surface rendering (flat)

i. Use face normal for determining single color for whole face.

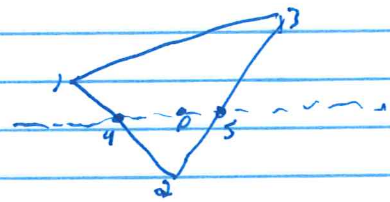
ii. Not realistic unless very far from camera or is truly flat object.

B. Gouraud Surface Rendering (intensity interpolation)

i. Idea - linear interpolation across polygon

ii. Compute intensity values at vertices w/ vertex normals then interpolate

iii. How - use scan lines



$$I_4 = \frac{y_4 - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_4}{y_1 - y_2} I_2$$

$$I_p = \frac{x_5 - x_p}{x_5 - x_4} I_4 + \frac{x_p - x_4}{x_5 - x_4} I_5$$

Can be incrementally updated: example

I_4 can be updated by $\frac{I_2 - I_1}{y_1 - y_2}$ each scan line.

iiii. Not perfect but quite sufficient for ~~realistic~~ real-time rendering.

C. Phong Surface Rendering - much more realistic

i. Idea - interpolate normals instead of intensities. Compute intensity values fresh

ii. Can be made "fast" by interpolating some of items, eg light direction as well as normal.

iii. Fast-Phong is 2x time of Gouraud and Phong is ~7x time of Gouraud