Computer Graphics

P. Healy

CS1-08
Computer Science Bldg.
tel: 202727
patrick.healy@ul.ie

Spring 2021–2022

Outline

Outline

Introduction to Lighting

- Lighting is an important part of creating convincing 3-D scenes

- Discussion of graphic on left
- Some confusion exists between the related topics of **lighting** and **shading** (or **surface rendering**)
- We refer to the model for calculating the light intensity at a single surface point as the
- We use the term to mean a procedure for applying a lighting model to obtain pixel colours for all surface positions

Outline

1. Lighting Models: §10
   - Introduction
   - Light Sources: §10.1
   - Surface Lighting and Reflection

Light Sources

- It is possible to model light sources that have a variety of and characteristics
- In some situations we may want to model an object that is both a *source* and *reflector* of light
  - A with reflector behind
  - A around a light bulb both emits (some) and reflects (remainder) – possibly model this as a semi-transparent surface
- A light source can be defined with certain properties:
  - position
  - colour of light emitted
  - emission direction
  - shape
- Some examples...

Light Source Examples

- Point Light Source
  - Simplest model of light emitting object
  - Defined by giving its position and the color of the light emitted
  - Light rays are generated along paths
  - Reasonable approximation for light sources whose dimensions are small compared to the size of objects in scene
- Infinitely Distant Light Source
  - Objects such as the sun can be approximated by a point emitter
  - But now there is little variation in its effects...
  - Rays are effectively to each other
  - Distance is not important here, just direction

Distance Effects on Light Sources

Radial Intensity Attenuation

- As radiant energy from a light source travels through space its strength at distance d_l from source is attenuated by the factor $1/d_l^2$
- So surfaces closer to light source receive higher light intensity than those far away
- This needs to be factored in for realistic lighting effects
- In practice a factor of $1/d_l^2$ doesn't work so well as it "over lights" surfaces close to the light source
- Instead we use

$$f_{\text{radatten}}(d_l) = \frac{1}{a_0 + a_1 d_l + a_2 d_l^2}$$

with coefficients a_0, a_1, a_2 chosen by trial and error

Distance Effects on Light Sources (contd.)

- For infinite-distance point sources this doesn't work so it is modified to

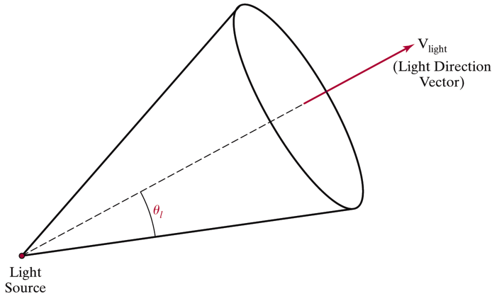
$$f_{\text{radatten}}(d_l) = \begin{cases} 1.0 & \text{if source is at infinity} \\ \frac{1}{a_0 + a_1 d_l + a_2 d_l^2} & \text{if source is local} \end{cases}$$

Directional Light Sources and Spotlight Effects

- Easy to modify a local light source to produce a (**spotlight**) beam of light
- Easy to duplicate this so that we have vectors corresponding to different colours for multi-colour light
- Now we can model light within this cone with

Directional Light Sources and Spotlight Effects

- Easy to modify a local light source to produce a (**spotlight**) beam of light



V_{light} is a *unit* vector giving direction of light

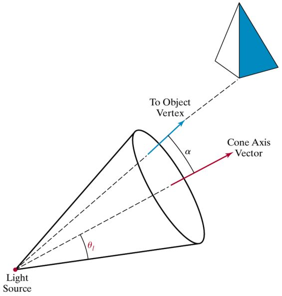
- Easy to duplicate this so that we have vectors corresponding to different colours for multi-colour light
- Now we can model light within this cone with

Directional Light Sources and Spotlight Effects

- Easy to modify a local light source to produce a (**spotlight**) beam of light
- Easy to duplicate this so that we have vectors corresponding to different colours for multi-colour light
- Now we can model light within this cone with

Directional Light Sources and Spotlight Effects

- Easy to modify a local light source to produce a (**spotlight**) beam of light
- Easy to duplicate this so that we have vectors corresponding to different colours for multi-colour light
- Now we can model light within this cone with



Directional Light Sources and Spotlight Effects

- An object is within light cone when $\alpha < \theta_l$
- This can be tested with the **scalar** or **dot product** of V_{light} and V_{obj}

$$V_{\text{light}} \cdot V_{\text{obj}} = |V_{\text{light}}| \cdot |V_{\text{obj}}| \cos \alpha$$

- If the two vectors are unit vectors (normalised so that $|V_{\text{light}}| = |V_{\text{obj}}| = 1$) we get

$$V_{\text{light}} \cdot V_{\text{obj}} = \cos \alpha$$

- Then

$$\cos \alpha \geq \cos \theta_l \implies \alpha \leq \theta_l$$

- In OpenGL's light model we can specify the light direction, cone angle and **angular intensity attenuation**

Angular Intensity Attenuation

- In addition to light decaying distance-wise, we can make the light decay as the angular distance from the cone axis increases
- Since $\alpha < \alpha' \Rightarrow \mathit{cos}\alpha > \mathit{cos}\alpha'$ for a directional light source we can model the intensity factor for the light at angle α with

$$f_{\text{angatten}}(\alpha) = \cos^{a_l} \alpha, \quad 0 \leq \alpha \leq \theta_l$$

  where the angular attenuation $a_l \geq 1$
- So, three cases for a light source l (need normalized vectors below)

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

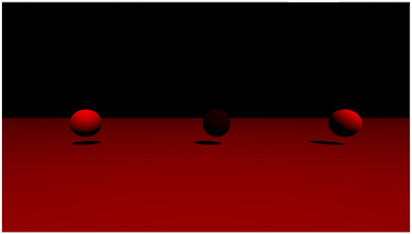
Outline

Surface Lighting Effects

- When light falls on an opaque surface some gets absorbed and some gets reflected
- The amount of incident light that gets reflected is depends on the type of material
- If the material is *transparent* then some (most?) light is also passed through
- Rough surfaces tend to scatter reflected light in all directions; this is called *diffuse* reflection
  - A rough, *matte* surface produces mainly diffuse reflection
  - The colour of an object is determined by the colour of the diffuse light reflected:
  - A red object reflects the red component of white light and absorbs all others; what happens with green light on red object?
  - See also this and this

Surface Lighting Effects (contd.)

With thanks to Vilius Drumsta we see the following effects of diffuse reflection on surfaces of different colours. In his own words:



Hi Paddy,

I set up this scene in Unity to check out how red light is reflected off of different coloured surfaces.

It looks like for a white surface (left sphere) it reflects all the red light, as expected. Same with a red surface (right sphere).

Blue surface (middle sphere) absorbs most of the red light except for a tiny percentage. In the real world, no surface is ever perfectly blue so what most game engines do is reflect a tiny percentages of other colours as well.

Surface Lighting Effects (contd.)

- As well as diffuse light scattering, reflected light concentrated in a narrow focus, or bright spot, is called **specular** reflection
  - In an ideal reflector (a mirror, say) all reflection is specular
  - Specular reflection only appears a certain viewing angle
  - See also here
- Light that is difficult to attribute to a specific source is called **ambient** light
  - This is light that occurs from light reflected from other light sources
  - Ambient light is usually constant over the entire scene
  - It can be thought of as "background" lighting
  - Reflection caused by ambient light is a form of diffuse reflection