COS 344: L4 Chapter 4.5 and Chapter 5

Cobus Redelinghuys

University of Pretoria

29/02/2024

Shading

- Recall from L3 that shading is applied where the ray intersects the surface.
- Chapter 5 is basic shading models.
- Chapter 14 is advanced shading models.

Section 4.5.1: Light sources

- Three most basic forms of lights:

Point light

A light that illuminates the scene from a **single point in space**.

Directional light

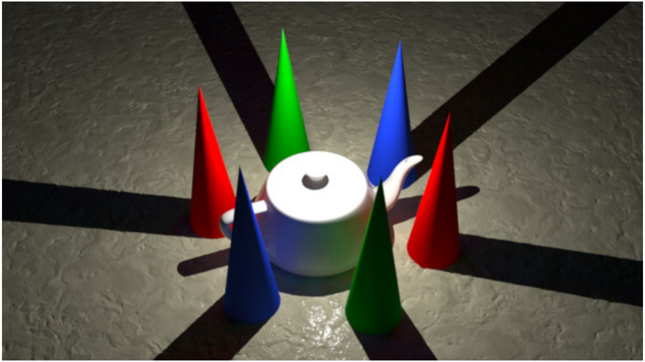
A light that illuminates the scene from a **single direction**.

Ambient light

A light that provides a **global light** that illuminates all the shadows in the scene.

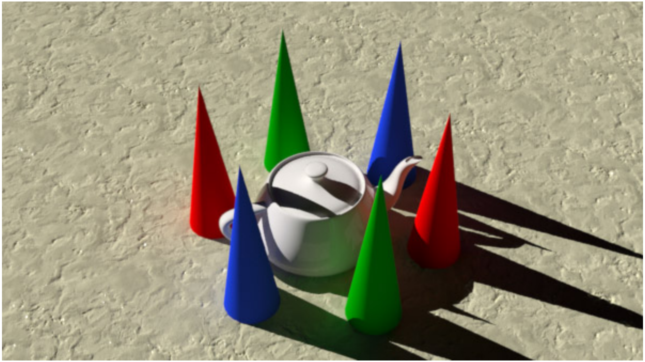
- Other fancier light systems do exist, but will be discussed later.

Point light



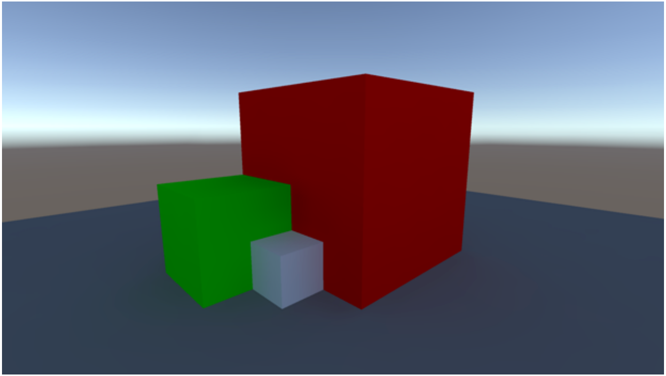
https://learn.foundry.com/modo/content/resources/images/shade_render/point_light.jpg

Directional light



https://learn.foundry.com/modo/content/resources/images/shade_render/distant_light.jpg

Ambient light



https://arbentapia.gitbooks.io/intro-lighting-and-rendering-in-unity/content/assets/ambient_light.png

- In order to render a point or directional light, the following is needed:
  1. $\mathbf{x}$: The shading point where the viewing ray intersects the surface for a value of t.
  2. $\mathbf{n}$: The normal of the surface at point $\mathbf{x}$.
  3. $\mathbf{l}$: The light direction which is computed depending on the light source.
     - Point light: light position
     - Directional light: light direction
  4. $\mathbf{v}$: The viewing direction which is the opposite of the direction of the ray:

$$\mathbf{v} = \frac{-\mathbf{d}}{||\mathbf{d}||}$$

- Ambient lighting is easier to calculate as there is no $\mathbf{l}$ and is independent of $\mathbf{v}$.
  - Why?

Section 4.5.2: Shading in software

- Textbook approach:
  - Light is responsible for overall illumination computations.
  - Material is responsible for computing BRDF values:
    - Bidirectional Reflectance Distribution Function
    - This function defines how light from a source is reflected off an opaque surface

Algorithm 1 Pseudo-code for point light illumination

Require: Color i, Point $\mathbf{p}$

$\mathbf{x} :=$ calculate where the ray intersects the surface.

$r = ||\mathbf{p} - \mathbf{x}||$

$\mathbf{l} = \frac{\mathbf{p}-\mathbf{x}}{r}$

$\mathbf{n} =$ normal of the surface at position $\mathbf{x}$

$\mathtt{E} = \frac{max(0, \mathbf{n} \cdot \mathbf{l})\mathtt{i}}{r^2}$

$\mathtt{k} =$ calculate the resulting color using the BRDF of the material of the surface.

return kE

- **p, x, l , n** are all vectors.
- i, E, k are colors.
  - Colors can be represented as a vector which depends on the number of channels.

Algorithm 2 Pseudo-code for ambient light illumination

Require: Color `i`

k = obtain the surface material of the surface.

return `ki`

- Algorithm 1 is repeated for each light ray.
- Algorithm 2 is repeated for each surface.
- If multiple light sources exist, the resulting color will be the summation of each light calculations- .

Section 4.5.3: Shadows

Shadow ray

A ray that determines if the light can reach a position on a surface.

- Shadows can be incorporated into Algorithm 1 with a simple shadow test:
  - If $\mathbf{x}$ is the first object that the shadow ray intersects, use the normal lighting effect.
  - Else shade the point $\mathbf{x}$.
- Shadows need to be computed for each light source.
- Do we compute shadows for ambient light sources?

Section 4.5.4: Mirror reflection

Mirror or ideal specular reflection

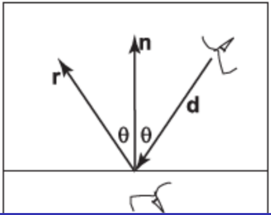
The angle of the incoming view ray is the same as the angle of the reflection.

- The reflection direction can be calculated as follows:

$$\mathbf{r} = \mathbf{d} - 2(\mathbf{d} \cdot \mathbf{n})\mathbf{n}$$

where:

- $\mathbf{r}$: reflection direction
- $\mathbf{d}$: viewing direction
- $\mathbf{n}$: normal of the surface



- Mirror reflection is not always usable as not all materials are perfect reflectors.
- Some material shift or alter the color:
  - Gold reflects yellow better and thus shifts the reflected color.
- The reflected color can thus be calculated as:

$$\text{Color } c = c + k_m shadeRayCalculation(...)$$

  - Where k_m is the mirror reflection and is also a specular RGB color.
- In the real world, k_m is non constant and will be discussed in Chapter 14.

Introduction

- Why is there a need for shadows in computer graphics?
  - Creates a more realistic depiction of 3D shapes.

Shading model

Equations used to compute shading.

- Shading models are independent of the rest of the rendering system and can be swapped in and out.
- Chapter 5 focuses on a point light source on an opaque surface.
- Chapter 14 will discuss more advanced topics.

Section 5.1: Point-like light sources

- Point-like light sources come in two categories:

Point source

A small light that is:

- Treated as a point
- Close to the scene that is being illuminated
- Can illuminate different surfaces differently

Directional source

A light that is:

- Small and relative to the distance from the scene.
- Illuminates all the surfaces equally.
- Only keeps track of the direction and not its position.

- Examples of each:
  - Flashlight - Point source
  - Sun - Direction source

Section 5.1.1: Point source illumination

- Point light sources are defined by two properties:
  - Position
  - Intensity
    - Describes the amount of light it produces.
- Types of point light sources:
  - Isotropic:
    - The light shines evenly in all directions
  - Spotlights:
    - Light only shines in certain directions.

Irradiance

Density of radiant power per unit area, for light falling on a surface with the purpose of light reflection.

- Given a source has a power of P and a receiving sphere of r, the irradiance E is:

$$E = \frac{P}{4\pi}\frac{1}{r^2} = \frac{I}{r^2}$$

- The quantity $I = \frac{P}{4\pi}$ is the intensity of the light source.
- r^2 shows that radiance is dependant on the distance, but not on the surface it is illuminating.

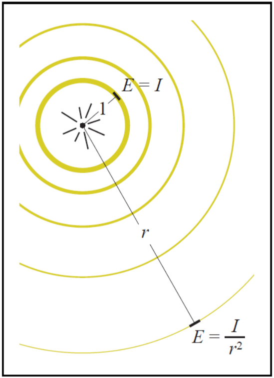
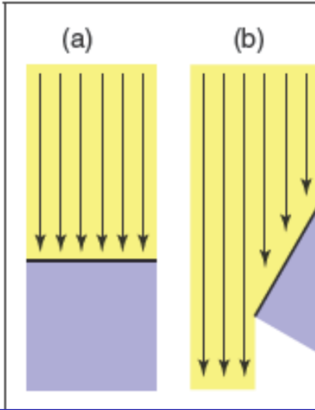


Figure 5.1. Irradiance from

Angle of incidence

The angle between the surface normal and the direction the light is traveling.

- Consider a small surface with a point light whose far away distance is relative to the size of the surface.
- The light rays that intersect the surface are parallel to each other and perpendicular to the surface.
- If the surface is rotated, such that only half the light falls on the surface, the light's intensity on the surface is halved.
- Lambert's cosine law describes this phenomena.



- Using the Lambert's cosine law, the irradiation equation can be expanded:

$$E = I\frac{cos\theta}{r^2}$$

- $\frac{cos\theta}{r^2}$ is known as the geometry factor for a point source.
- $cos\theta$ can be replaced with $\mathbf{n} \cdot \mathbf{l}$.
  - $\mathbf{n}$ is the normal of the surface.
  - $\mathbf{l}$ points *towards* the light.
  - Both $\mathbf{n}$ and $\mathbf{l}$ are unit length

$$E = I\frac{\mathbf{n} \cdot \mathbf{l}}{r^2}$$

Section 5.1.2: Directional illumination

- Directional light is a light whose source is very, far far away.
- The effects of $\frac{I}{r^2}$ are less and less visible.
- Thus replace $\frac{I}{r^2}$ with constant normal irradiance constant H.

$$E = Hcos\theta$$

Section 5.2.1: Lambertian reflection

Ideal diffuse surface

Surface that reflects the light equally to all directions regardless of light origin.

$$L_r = kE$$

- Ideal diffuse surfaces have the following properties:
  - Brightness is same in all directions.
  - Color is independent of viewing direction and described by reflectance R.

$$L_r = \frac{R}{\pi}E$$

- Provides a flat, chalky appearance and effective for modelling paper, flat paint, dirt, tree etc.

Section 5.2.2: Specular reflection

Specular reflection

Reflection that is view-dependant

Ideal specular reflection

Reflection on a perfectly smooth surface like water or mirror.

Glossy reflection

Reflection on surfaces that are not perfectly smooth.

- Most common glossy reflection model: Modified Binn-Phong model.

Idea: When the view (**v**) and light direction (**l**) are symmetrically positioned across the surface normal (**n**), the reflection is at its brightest and decreases smoothy as the vectors move away.

- Use a half vector (**h**) that is a vector that is halfway between **l** and **v**.

$$h = \frac{\mathbf{l} + \mathbf{v}}{||\mathbf{l} + \mathbf{v}||}$$

- The closer **h** is to **n** the shinier and is measured using $\mathbf{n} \cdot \mathbf{h}$.
- Using the Phong exponent (p) we can alter the decay of shininess.

$$(\mathbf{n} \cdot \mathbf{h})^p$$

- Incorporating Blinn-Phong into Lambertian shading we can use:

$$L_r = \left(\frac{R}{\pi} + k_s max(0, n \cdot h)^p \right) E$$

- k_s is the specteral coefficient.
- Section 5.2.3 is left to read while implementing lighting.

Section 5.3: Ambient illumination

- Ambient illumination is the easiest form of lighting as it is just a constant.
  - Due to no light direction vectors.

$$L_r = k_a I_a$$

where:

  - k_a is the ambient reflection coefficient.
  - I_a is the ambient intensity.
- Allows for easy fine tuning for different objects and the scene as a whole.

Joke of the day - By ChatGPT

Why did the surface shader refuse to work?

Joke of the day - By ChatGPT

Why did the surface shader refuse to work?

It had too many "normal" issues!