CMT107 Visual Computing

IV.1 Illumination Models

Xianfang Sun

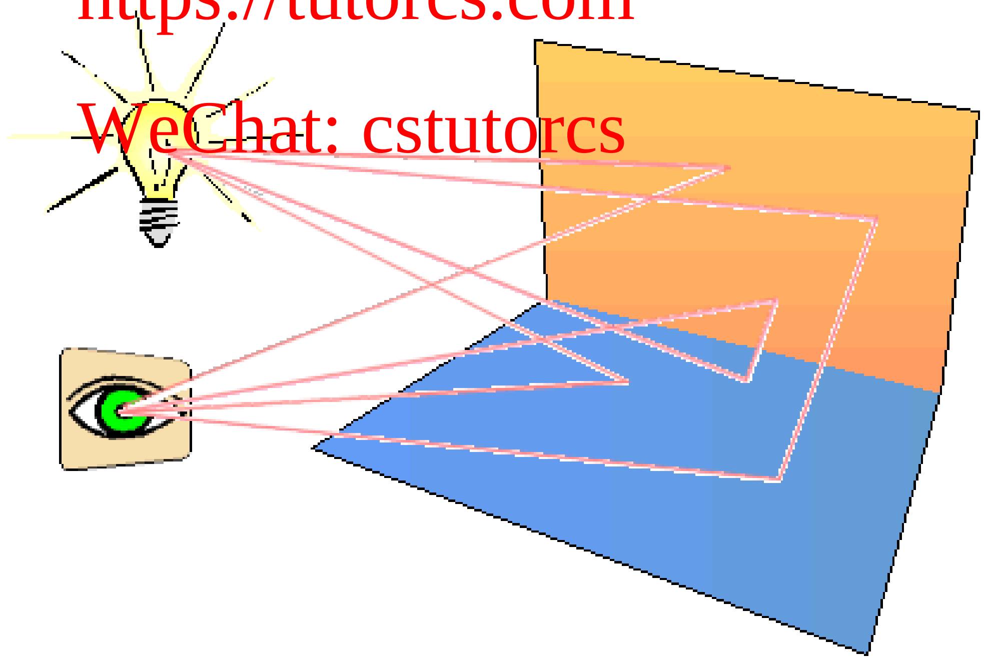
School of Computer Science & Informatics
Cardiff University

Overview

- Illumination Concepts
- Light Reflection model
  - Phong illumination model
- Light source types
- OpenGL lighting

Illumination Concepts

- *Illumination:* transport of luminous flux from light sources between points via direct and indirect paths
- *Lighting:* computing luminous intensity reflected from a specific 3D point
- *Shading:* assigning colours to a pixel
- Illumination Models: Simple approximations of light transport

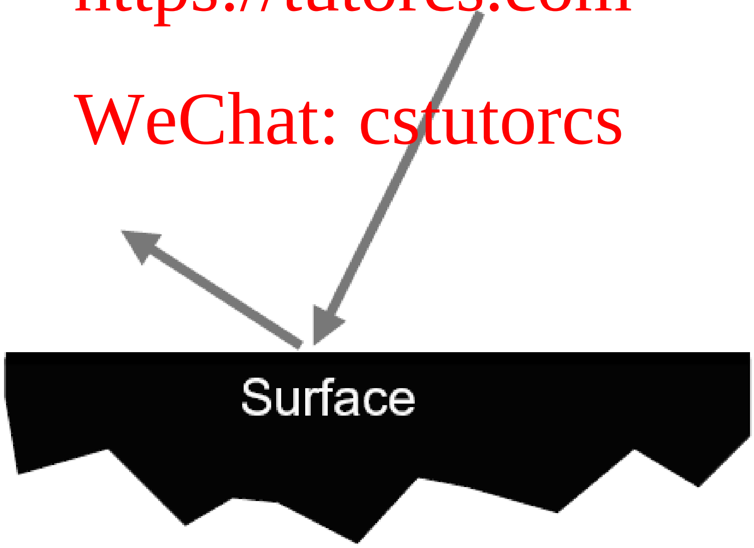


Light-Surface Interaction

- Light and surface properties determine the illumination
- Light that strikes an object is partially absorbed and partially reflected
- The amount reflected determines the colour and brightness of the object (subtractive colours)
- Reflected light is scattered depending on the smoothness and orientation of the surface

Modelling Surface Reflectance

- Compute light *reflected* by surface as *observed by viewer*
- Surface material tells *how much* of the incoming light is reflected
  - Type of light determines reflection model
- Intensity of observed light depends on *direction to light source* and *direction to viewer*



Light Reflection Types

- *Ambient* light: comes from all directions, is scattered in all directions
- *Diffuse* light: comes from one direction, is scattered in all directions
- *Specular* light: comes from one direction, reflected in preferred direction (highlights)

Ambient Reflection

- Ambient light is the same everywhere
  - Amount of reflected light of incoming intensity $\mathbb{I}_{\text{ambient},c}$ is *independent* of direction to light source and viewer
- Intensity of reflected light observed by a viewer:

$$L_{\text{ambient},c} = R_{\text{ambient},c} \mathbb{I}_{\text{ambient},c}$$

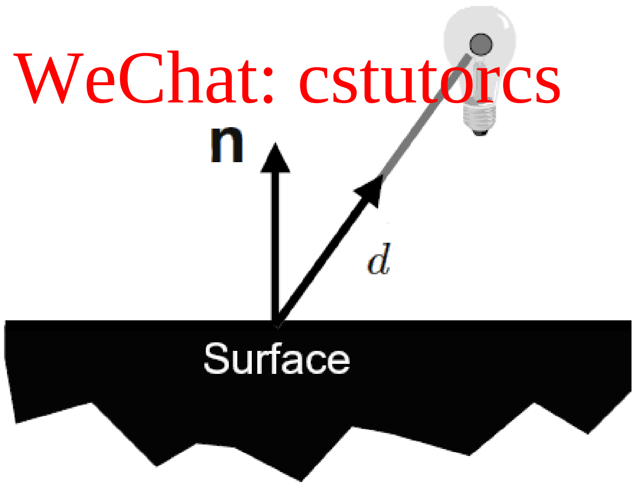
  - $R_{\text{ambient},c}$ is ambient material property for colour c (percentage of red, green or blue ambient light reflected by surface)

Diffuse Reflection

- Light is reflected in all directions
  - Amount of reflected light of incoming intensity *depends only on direction to light source*
- *Lambertian model* (use cosine law / scalar product):

$$L_{\text{diffuse},c} = R_{\text{diffuse},c}(n^t d) I_{\text{diffuse},c}$$

  - *d*: unit direction from surface point to light source
  - *n*: unit surface normal

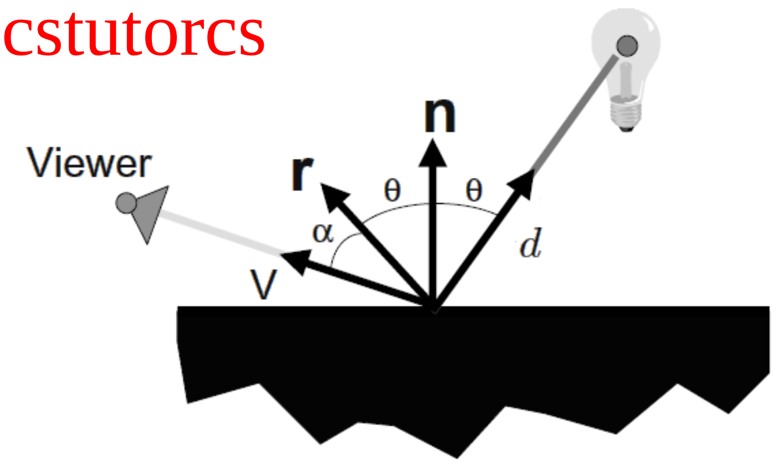


Specular Reflection

- Light is reflected preferable in *direction of perfect reflection*
  - Amount of reflected light of incoming intensity depends on direction to light source and to viewer
- Observed light intensity:

$$L_{\text{specular},c} = R_{\text{specular},c}(r^t v)^{\sigma} I_{\text{specular},c}$$

  - *r*: unit direction of perfect reflection of d
  - *v*: unit direction towards viewer position
  - σ is shininess exponent



Surface Light Emissions

- Can make surface *emit* light, not just reflect light
- Simple model:
  - Add emissive light intensities $E_{t,c}$ to light intensities for each light type t and colour c
  - Does not illuminate other surfaces (but can add a multiple point light sources behind surface or a directional light source for larger light emitting surfaces)

Phong Illumination Model

- Putting everything together gives the *Phong Illumination Model*
- Consider monochromatic light (e.g. red, green or blue) and a single light source:
  - Depending on light source type, at a surface point the incoming intensity of different light types is $\mathbb{I}_a, \mathbb{I}_d, \mathbb{I}_s$
  - The *intensity of reflected light* is:

$$k_a \mathbb{I}_a + k_d (n^t d) \mathbb{I}_d + k_s (r^t v)^\sigma \mathbb{I}_s$$

  - *Summation* over all light sources for red, green, blue gives total intensity for all colours
- Note, Phong's illumination model is *not* physically accurate

Light Source Types

- Ambient light source: light from the environment
- Directional light source: light from infinite distance in a specified direction
- Point light source: light from single point
- Spot light source: light emitted in a cone
- other light source: area light source, extended light source etc.

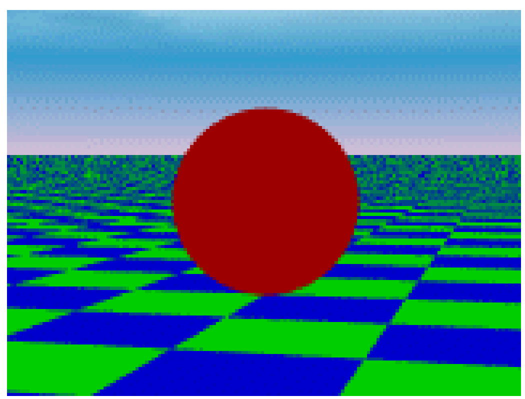
Ambient Light Source

- An object not directly lit is still visible
  - Caused by light reflected from other surfaces
- Modelled by a single ambient light source
  - Instead of computing surface reflections, specify *constant ambient light* for all surfaces

  - Defined solely by ambient RGB light *intensities*

- Intensity arriving at point p from an ambient light of intensity $L_{ambient,c}$ and colour c :

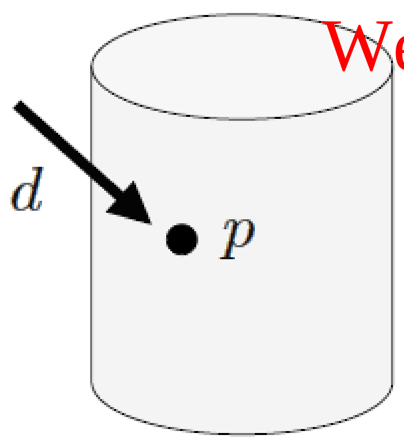
$$I_{ambient}(p, L_{ambient,c}) = L_{ambient,c}$$

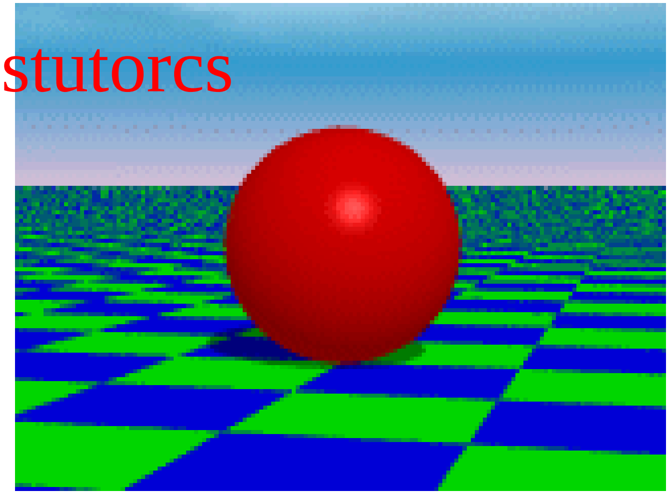


Directional Light Source

- Light from a source *infinitely far away*
  - Defined by *intensities* of emitted RGB light of all types,
  - *direction* $\boldsymbol{d}, \|\boldsymbol{d}\| = 1$ (and no position)
- Intensity arriving at point p from a directional light of intensity $L_{t,c}$:

$$I_{\text{directional:d}}(p, L_{t,c}) = L_{t,c}$$

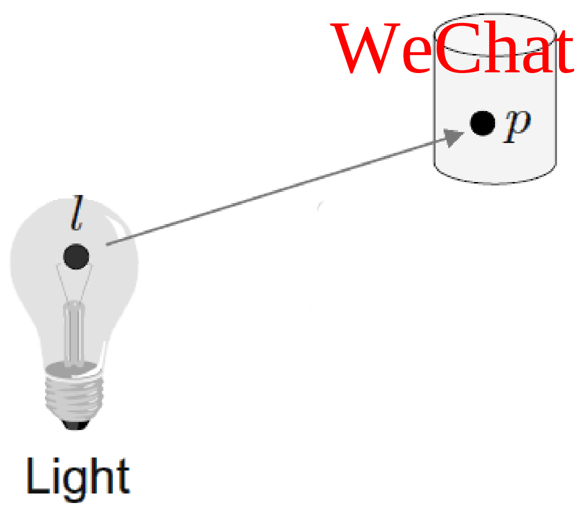


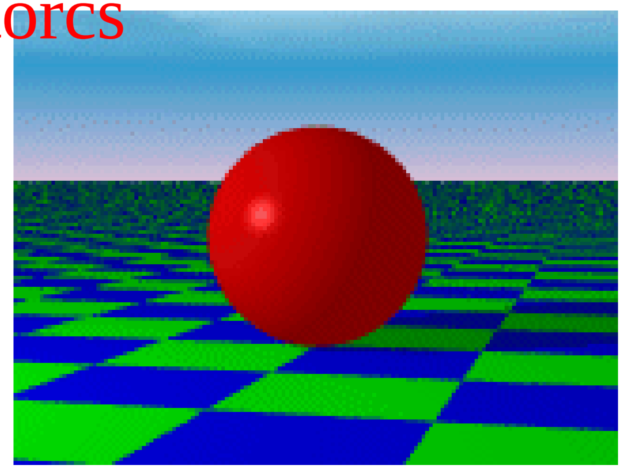


Point Light Source

- Light emitted *radially* from single point *in all directions*
  - Defined by *intensities* of emitted RGB light for all types,
  - *position* l (and no direction),
  - constant, linear and quadratic *attenuation* (k_c, k_l, k_q)
- Intensity arriving at point p from a point light of intensity L_c :

$$I_{\text{point}:l,k_c,k_l,k_q}(p, L_{t,c}) = \frac{1}{k_c + k_l\|p - l\| + k_q\|p - l\|^2} L_{t,c}$$

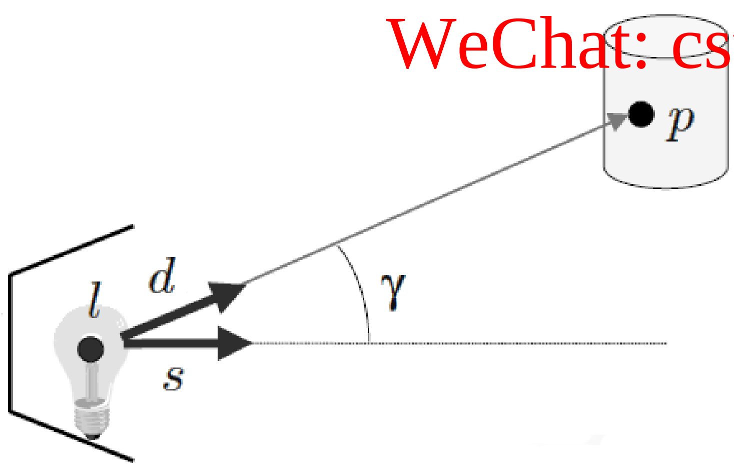


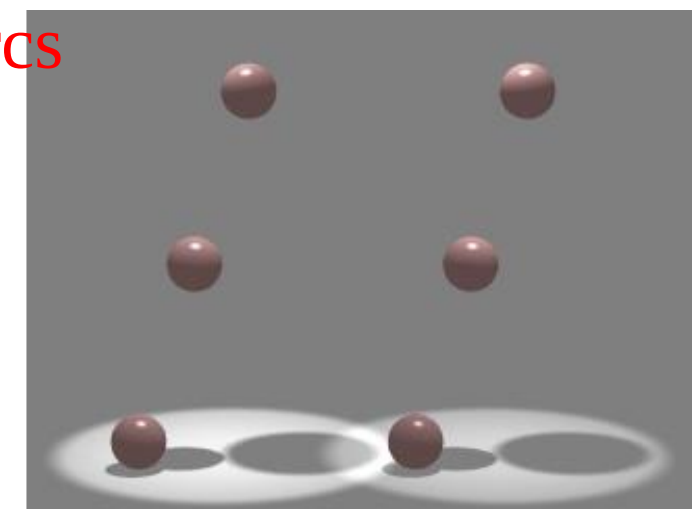


Spot Light Source

- Light emitted in a *cone*
  - Defined by *intensities* of emitted RGB light for all types,
  - *position* l, *unit cone direction* s, *spot cut-off exponent* τ,
  - constant, linear and quadratic *attenuation* (k_c, k_l, k_q)
- Intensity arriving at point p from an point light of intensity $L_{t,c}$:

$$I_{spot:l,s,\tau,k_c,k_l,k_q}(p, L_{t,c}) = \frac{(s^t((p-l)/\|p-l\|))^\tau}{k_c + k_l\|p-l\| + k_q\|p-l\|^2} L_{t,c}$$





Light Source "Visibility"

- *Angle cut-off* for spot lights:
  - If position p is outside light cone ($s^{\mathrm{T}}d = \cos\gamma < \cos\delta$ with $\boldsymbol{d} = (\boldsymbol{p}-\boldsymbol{l})/\|\boldsymbol{p}-\boldsymbol{l}\|$ and cone semi-angle δ), set I to 0
- Light source *behind* surface:
  - Diffuse and specular light only reflected if light source is in front of surface at p
  - Set diffuse and specular light intensities from light sources to 0 if $n^{\mathrm{T}}d \leq 0$
    - n: unit surface normal at p
    - d: unit direction from p to light source
  - This distinguishes between front and back of surfaces / polygons (also see two-sidedness)

OpenGL lighting

- Fixed-function pipeline version of OpenGL (old version) uses specific functions to define lighting and material properties. And lighting effects are realised inside the OpenGL pipeline
- Shader version of OpenGL (new version) needs the programmer to write code in the main program and/or the shaders to implement lighting effects
- More details in the labs…

Surface Normal Vectors

- For lighting computations OpenGL requires *normal vectors* of polygonal primitives
  - Orthogonal to surface pointing outwards
  - Used to compute reflection angle
- Normals are sent to the vertex shader together with vertex coordinates
- Normals should be unit vectors
  - The function normalize() in shaders can be used to convert a vector to a unit vector:

```
Vn = normalize(V);
```

Summary

- What is ambient, diffuse and specular light? How is the amount of reflected light for each light type computed?
- What is the Phong illumination model?
- What are ambient, directional, point and spot light sources? How is the light intensity arriving from one of these light sources at a surface point computed?
- Distinguish light reflection types and light source types.