Lighting

Measuring light

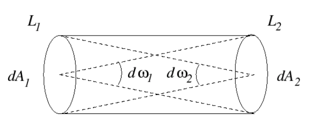
For any point in space, we can consider directional distribution of photons going through a differential area at this point.

Radiance: energy per unit time, per unit differential area perpendicular to the ray, per unit solid angle in the direction of the ray.
Measured in watts/meter^2 /steradian

If $\phi(x,\omega) = \frac{dN}{d\Omega}$ is directional distribution of photons of wavelength λ going through an area , then the radiance is $L(x,\omega,\lambda) = \frac{hc}{\lambda}\phi(x,\omega)$

energy of a photon

Constancy of Radiance



radiance is constant along a ray: consider the flow of photons in a a thin pencil; the number of photons entering onthe right with the direction inside $d\omega_1$ exit through the other side; equating the expressions for entering and exiting diff. flows we get

$$d\Phi_1 = L_1\ d\omega_1 dA_1 = L_2\ d\omega_2 dA_2 = d\Phi_2$$

but $dA_1 d\omega_1 = dA_2 d\omega_2$ so $L_1 = L_2$

Local illumination model

Describes interaction of the light and the surface

- light can be reflected, absorbed and transmitted
- most important: reflection
- reflection can be ideal specular, diffuse and and anything between
- reflection equation relates the outgoing radiance in some direction to the incoming radiance

BRDF

<u>irradiance</u>: light flow per unit area of surface

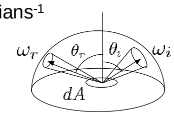
flow of radiance L spanning solid angle $d\omega_i$ creates differential irradiance $L d\omega_i \cos\theta_i$

bidirectional reflectance distribution function:

the ratio of reflected radiance in direction r to the differtial irradiance in the direction i

units: steradians^{-1}

$$f(\omega_i,\omega_r) = \frac{dL_r(\omega_i,\omega_r)}{L_i \cos\theta_i d\omega_i}$$

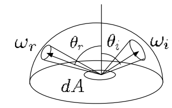


Reflection equation

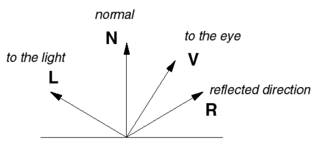
the outgoing radiance in direction r is the sum of the radiances due to radiance from all incoming directions:

$$L_r(\omega_r) = \int f_r(\omega_i,\omega_r)\ L_i(\omega_i)\cos\theta_i d\omega_i$$

the integral is over the upper hemisphere



Reflection geometry



$$\mathbf{V} = \omega_r,\ \mathbf{L} = \omega_i$$

$$\mathbf{R} = \mathbf{L} + 2(\mathbf{L}\cdot\mathbf{N})\mathbf{N}$$

Phong model

1."BRDF"

$$f_r(\omega_i,\omega_r) = K_{diff} + K_{spec}(\omega_i\cdot\omega_r)^p$$

Point light source intensity: power per unit solid angle

intensity in a direction ω**:** $I(\omega) = \dfrac{d\Phi}{d\omega}$

radiance created by light source at distance r **in the direction of the source:** $L(\omega, r) = \dfrac{I(\omega)}{r^2}$

To avoid integration in the reflection equation, ignore radiance from all directions except a finite number (e.g. direction to the light sources).

Phong model

Phong model and Z-buffer rendering:

- **assume point light sources; ignore irradiance from all directions except the directions to the lights;**
- **ignore occlusions, that is, no shadows).**

Phong model and (classical) ray tracing:

- **consider reflection and transmission;**
- **take occlusions into account.**

Phong model

$$L(\mathbf{V}) = K_{amb}L_{amb} + \sum_i L_i\left(K_{diff}(\mathbf{L}_i\cdot\mathbf{N}) + K_{spec}(\mathbf{R}_i\cdot\mathbf{V})^p\right)$$

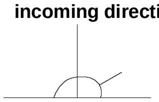
summation is over all light sources.

Ambient term: a hack. Because we ignore diffuse reflected light from objects (e.g. walls) the resulting images are often too dark.

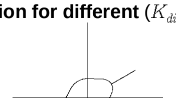
Another hack: replace $L_i = \dfrac{I_i}{r^2}$ **with** $\dfrac{I_i}{d_c + d_l r + d_q r^2}$

Phong model

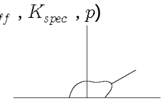
Directional plots of BRDF for a fixed incoming direction for different (K_{diff} , K_{spec} , p)



(0.8,0.2,4)

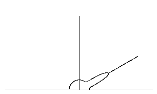


(0.7,0.3,8)

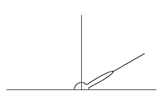


(0.6,0.4,16)

(0.4,0.6,32)



(0.3,0.7,64)



(0.2,0.8,128)

Constants and units

K_{diff} , K_{spec} **reflection coefficients, 3 color components**

p **Phong exponent, nondimensional, same for all colors**

L, L_{amb} **watts/meter²/steradian, 3 color components**

I_i **light source intensity, 3 color components**

OpenGL model

Several additions:

ambient term per object;
emmision;
ambient, diffuse, specular light “intensities”

Setting material parameters ($K_{diff}, K_{spec}, K_{amb}, p$)

GLfoat mat_diffuse[3], mat_spec[3], mat_amb[3];
GLfloat shininess;

```
...
glMaterialfv(GL_FRONT,GL_DIFFUSE,mat_diffuse);
glMaterialfv(GL_FRONT,GL_SPECULAR,mat_spec);
glMaterialfv(GL_FRONT, GL_AMBIENT,mat_amb);
```


Ray tracing

For each pixel shoot a ray R from camera;

```
pixel = TraceRay(R)
```

The recursive ray tracing procedure:

`RGBvalue TraceRay(Ray R)`
shoot rays to all light sources;
for all visible sources, compute RGB values r_i
shoot reflected ray R_{refl}; r_{refl}=TraceRay(R_{refl})
shoot refracted ray R_{trans}; r_{trans}= TraceRay(R_{trans})
compute resulting RGB value from
r_i, r_{refl}, r_{trans} using the lighting model

Lighting model for ray tracing

New effects: reflection, refraction; need more terms

reflection part:

$$L_1(\mathbf{V}) = \sum_{\substack{\text{visible sources}\\ \text{in front}}} L_i\left(K_{diff}(\mathbf{L}_i \cdot \mathbf{N}) + K_{spec}(\mathbf{R}_i \cdot \mathbf{V})^p\right) + k_{refl} L_{refl}$$

radiance from the reflected ray

$$L_2(\mathbf{V}) = \sum_{\substack{\text{visible sources}\\ \text{behind}}} L_i\left(K_{diff}(\mathbf{L}_i \cdot \mathbf{N}) + K_{spec}(\mathbf{T}_i \cdot \mathbf{V})^p\right) + k_{trans} L_{trans}$$

radiance from the refracted ray

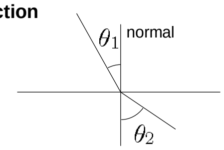
$$L(\mathbf{V}) = K_{amb} L_{amb} + (1-t) L_1(\mathbf{V}) + t L_2(\mathbf{V})$$

t **is transparency**

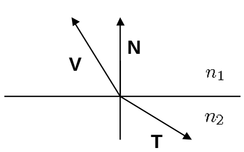
Snell’s law

If a surface separates two media with different refraction indices (e.g. air and water) the light rays change direction when they go through.

Snell’s law: the refracted ray is stays in the plane spanned by the normal and the direction of the original ray. The angles between the normal and the rays are related by $n_1 \sin\theta_1 = n_2 \sin\theta_2$ **where n_1 and n_2 are refraction indices.**



Refracted ray direction



$$n = \frac{n_1}{n_2}$$

$$\mathbf{T} = -n\mathbf{V} + \left(n(\mathbf{V},\mathbf{N}) - \sqrt{1 - n^2(1 - (\mathbf{V},\mathbf{N})^2)}\right)\mathbf{N}$$