



CS 354

Lighting

Mark Kilgard
University of Texas
February 28, 2012

Fixed-function OpenGL

Per-vertex Lighting

- OpenGL API includes lighting model
 - Operates at the per-vertex level
 - Implements a Blinn-Phong lighting model
- Example usage
 - `glEnable(GL_LIGHTING);`
 - `glEnable(GL_LIGHT0);`
 - `glLightfv(GL_LIGHT0, GL_POSITION, position);`
 - `glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse);`
 - `glLightfv(GL_LIGHT0, GL_SPECULAR, specular);`
- Relies on
 - Per-vertex normal value sent with `glNormal3f`
 - And per-vertex material properties sent with `glMaterialfv`

OpenGL Light Parameters

- Per-light state set with `glLightfv` command

Token Name	Meaning	Type	Initial Value
<code>GL_AMBIENT</code>	Ambient light color	RGBA, 4 floats	(0,0,0,1)
<code>GL_DIFFUSE</code>	Diffuse light color	RGBA, 4 floats	(1,1,1,1) for light 0, (0,0,0,1) for other lights
<code>GL_SPECULAR</code>	Specular light color	RGBA, 4 floats	(1,1,1,1) for light 0, (0,0,0,1) for other lights
<code>GL_POSITION</code>	Light location	Homogeneous position, 4 floats	(0,0,1,0), +Z axis direction
<code>GL_SPOT_DIRECTION</code>	Spot light direction	Direction vector, 3 floats	(0,0,-1)
<code>GL_SPOT_EXPONENT</code>	Spot light exponential fall-off	Non-negative scalar exponent	0
<code>GL_SPOT_CUTOFF</code>	Angle of spot light cut off	Angle in degrees	180, uniform distribution
<code>GL_CONSTANT_ATTENUATION</code>	Light attenuation inverse constant	Scalar float	1
<code>GL_LINEAR_ATTENUATION</code>	Inverse linear attenuation	Scalar float	0
<code>GL_QUADRATIC_ATTENUATION</code>	Inverse quadratic attenuation	Scalar float	0

OpenGL Material Parameters

- Per-light state set with `glLightfv` command
 - Surface has `GL_FRONT` and `GL_BACK` versions of these materials
 - For two-sided lighting, depending on how polygon faces

Token Name	Meaning	Type	Initial Value
<code>GL_AMBIENT</code>	Ambient material color	RGBA, 4 floats	(0.2, 0.2, 0.2, 1)
<code>GL_DIFFUSE</code>	Diffuse material color	RGBA, 4 floats	(0.8, 0.8, 0.8, 1)
<code>GL_SPECULAR</code>	Specular material color	RGBA, 4 floats	(0, 0, 0, 1)
<code>GL_EMISSION</code>	Emissive material color	RGBA, 4 floats	(0, 0, 0, 1)
<code>GL_SHININESS</code>	Specular exponent of material	Scalar float	0

OpenGL Fixed-function Lighting Equation

surface
result
color

$$\begin{aligned}
 \mathbf{c} = & \mathbf{e}_{cm} \quad \leftarrow \text{emissive} \\
 & + \mathbf{a}_{cm} * \mathbf{a}_{cs} \quad \leftarrow \text{global ambient} \\
 & + \sum_{i=0}^{n-1} (att_i)(spot_i) [\mathbf{a}_{cm} * \mathbf{a}_{cli} \quad \leftarrow \text{per-light ambient} \\
 & \quad + (\mathbf{n} \odot \overrightarrow{\mathbf{VP}}_{pli}) \mathbf{d}_{cm} * \mathbf{d}_{cli} \\
 & \quad + (f_i)(\mathbf{n} \odot \hat{\mathbf{h}}_i)^{s_{rm}} \mathbf{s}_{cm} * \mathbf{s}_{cli}] \\
 & \quad \leftarrow \text{diffuse} \quad \quad \quad \leftarrow \text{specular}
 \end{aligned}$$

for each light source

$$f_i = \begin{cases} 1, & \mathbf{n} \odot \overrightarrow{\mathbf{VP}}_{pli} \neq 0, \\ 0, & \text{otherwise,} \end{cases} \quad \leftarrow \text{diffuse squashes specular}$$

OpenGL Lighting Equation Terms

half-angle

$$\mathbf{h}_i = \begin{cases} \overrightarrow{\mathbf{VP}}_{pli} + \overrightarrow{\mathbf{VP}}_e, & v_{bs} = \text{TRUE}, \\ \overrightarrow{\mathbf{VP}}_{pli} + (0 \ 0 \ 1)^T, & v_{bs} = \text{FALSE}, \end{cases}$$

local viewer assumption

infinite viewer assumption

distance attenuation

$$att_i = \begin{cases} \frac{1}{k_{0i} + k_{1i}\|\mathbf{VP}_{pli}\| + k_{2i}\|\mathbf{VP}_{pli}\|^2}, & \text{if } \mathbf{P}_{pli}'s \ w \neq 0, \\ 1.0, & \text{otherwise,} \end{cases}$$

inverse square fall-off

spotlight attenuation

$$spot_i = \begin{cases} (\overrightarrow{\mathbf{P}}_{pli} \odot \hat{\mathbf{s}}_{dli})^{s_{rli}}, & c_{rli} \neq 180.0, \overrightarrow{\mathbf{P}}_{pli} \odot \hat{\mathbf{s}}_{dli} \geq \cos(c_{rli}), \\ 0.0, & c_{rli} \neq 180.0, \overrightarrow{\mathbf{P}}_{pli} \odot \hat{\mathbf{s}}_{dli} < \cos(c_{rli}), \\ 1.0, & c_{rli} = 180.0. \end{cases}$$