

11/11/2011

FACULTATEA DE
AUTOMATICA SI
CALCULATOARE

ELEMENTE DE GRAFICA PE CALCULATOR



Laborator 1

Iluminare in OpenGL

1. Modelul de iluminare de baza

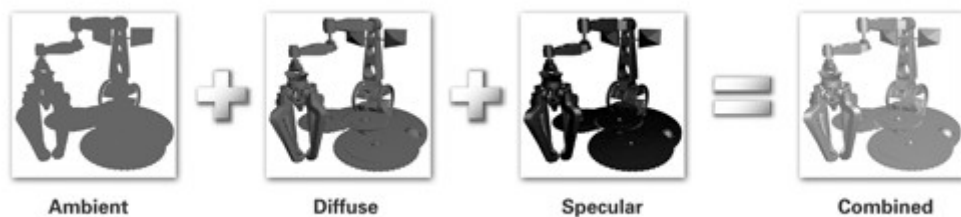
// modelul de iluminare folosit (normale per vertex sau per fata)

glShadeModel(GL_SMOOTH / GL_FLAT);

Un prim pas catre realismul unei scene 3D este introducerea de lumini. Desi fenomenul natural are o complexitate mult mai mare in OpenGL acesta poate fi simplificat pana in punctul in care poate fi descompus in urmatoarele componente :

Impreuna, iluminarea si proprietatile de material ale unui obiect ii determina aspectul. Un model de iluminare descrie felul in care lumina interactioneaza cu un obiect pe baza caracteristicilor luminii si amaterialului obiectului.

In continuare vom prezenta o versiune simplificata a modelului implementat in OpenGL. Acest model extinde si modifica modelul Phong de baza. In acest model, culoarea suprafetei unui obiect este data de suma componentelor emisiva, ambientala, difuza si speculara. Fiecare contributie depinde de combinatia dintre proprietatile materiale ale suprafetei si proprietatile sursei de lumina.

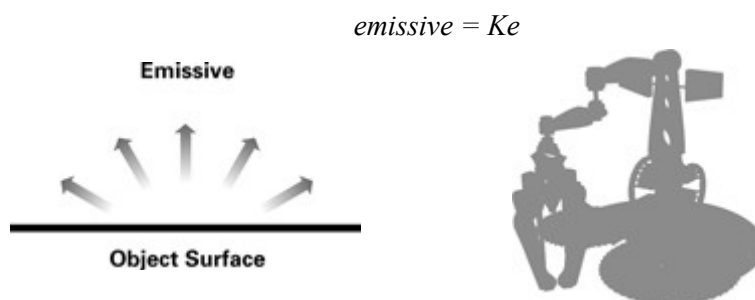


Pentru setarea unei proprietati de material se foloseste functia

glMaterialfv(GL_FRONT_AND_BACK, channel, color4f);

1.1. Lumina emisiva

Termenul emisiv reprezinta lumina emisa de o suprafata. Aceasta contributie este independenta de sursele de lumina. In OpenGL, aceasta este doar simulata, obiectul neluminand la randul lui alte obiecte.



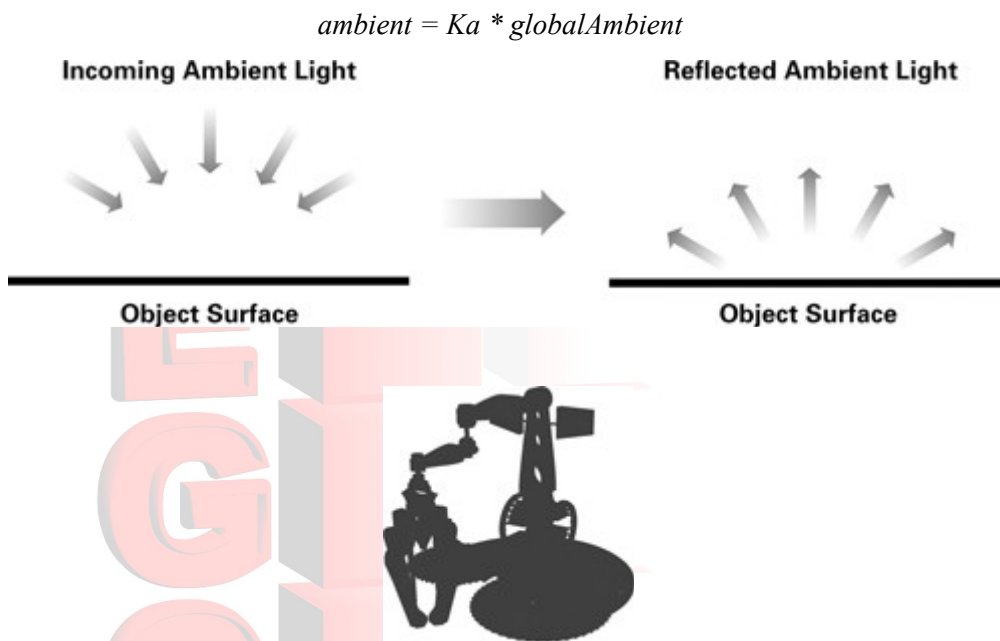
$$emissive = Ke$$

Setare componenta emisiva:

```
glMaterialfv(GL_FRONT_AND_BACK, GL_EMISSION, color4f);
```

1.2. Lumina ambientală

Lumina ambientală reprezintă lumina ce s-a reflectat în mediu atât de mult încât pare să vină de pretutindeni. Datorită acestui lucru, componenta ambientală este independentă de poziția sursei de lumină. Termenul ambiental depinde de reflectanța ambientală a materialului și de culoarea luminii ambientale incidente.

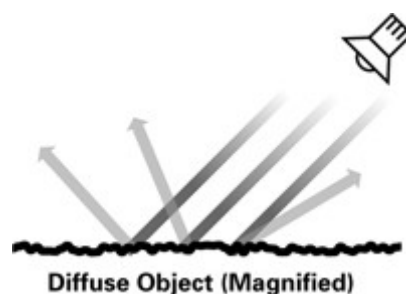


Setare K_a și culoare globală ambientală:

```
glLightModelfv(GL_LIGHT_MODEL_AMBIENT, globalAmbientColor4f); \
glMaterialfv(GL_FRONT_AND_BACK, GL_AMBIENT, color4f);
```

1.3. Lumina difuză

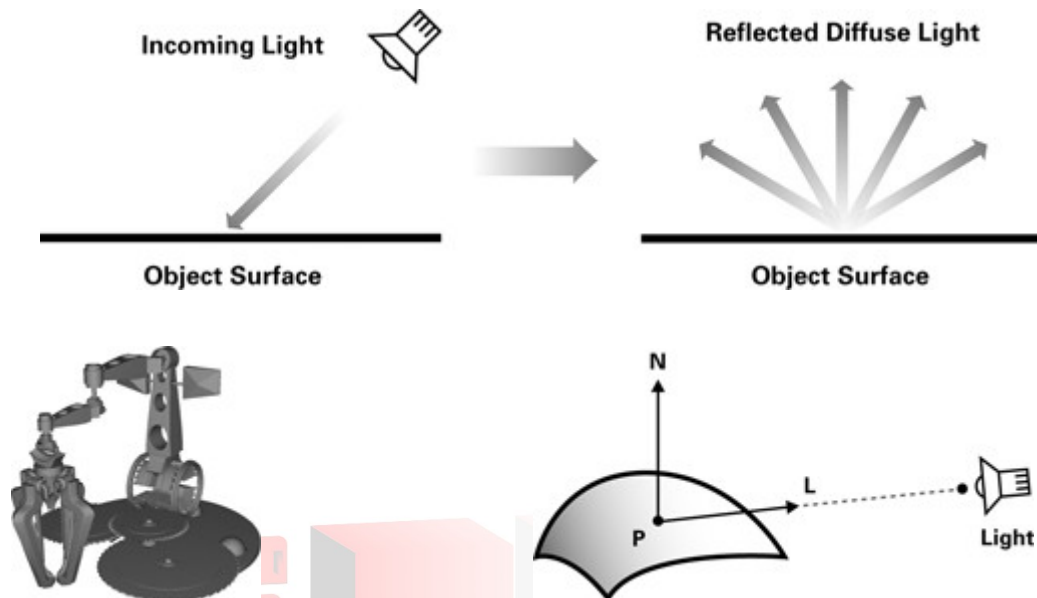
Lumina difuză reprezintă lumina provenită de la o sursă clar definită și reflectată în toate direcțiile de către suprafața.



Cantitatea de lumina reflectata este direct proportionala cu unghiul de incidenta al luminii.

$$diffuse = K_d * lightColor_d * \max(\text{dot}(N, L), 0)$$

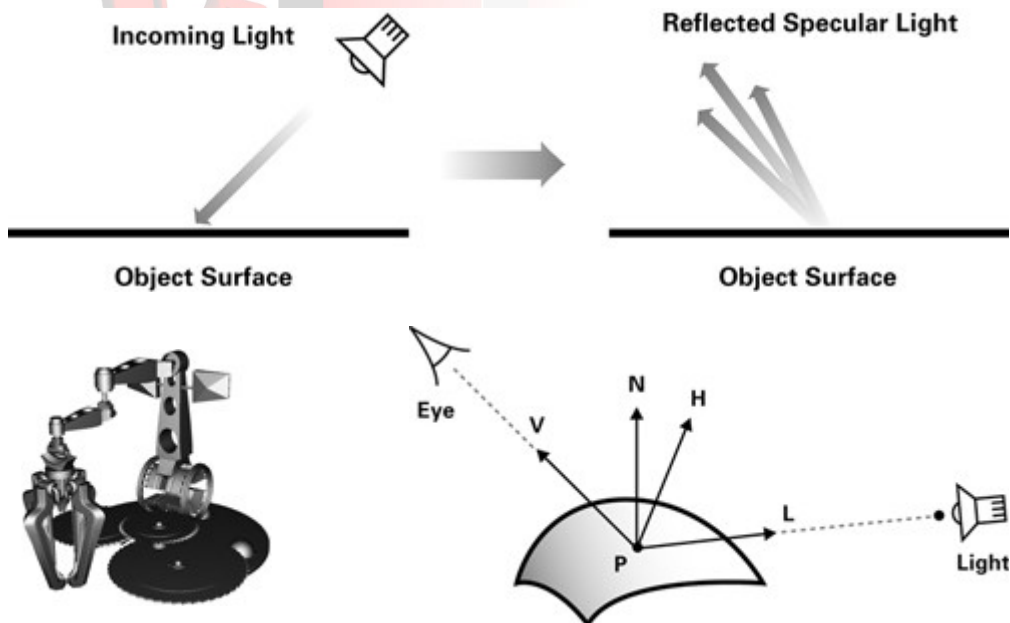
```
glMaterialfv(GL_FRONT_AND_BACK, GL_DIFFUSE, color4f);
```



1.4. Lumina speculara

Termenul specular reprezinta lumina reflectata de o suprafata predominant in jurul axei de oglindire. Aceasta este componenta care da stralucire obiectelor si este foarte proeminenta pe suprafete foarte netede si lucioase. Spre deosebire de celelalte, contributia speculara depinde de pozitia observatorului.

$$specular = K_s * lightColor_s * facing * \text{pow}(\max(\text{dot}(H, N), 0), shininess)$$



Setare componenta speculara

```
glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, color4f);
```

Pentru shininess se foloseste

```
glMaterialf(GL_FRONT_AND_BACK, GL_SHININESS, value);
```

1.5. Crearea unei lumini

Iluminarea in OpenGL este initializata (oprita) folosind comenzile:

```
glEnable(GL_LIGHTING);
```

```
glDisable(GL_LIGHTING);
```

Dupa aceasta, lumini individuale pot fi initializate / oprite folosind glEnable / glDisable si primind ca parametru o constanta simbolica.

```
glEnable(GL_LIGHT0); // porneste prima lumina
```

...

```
glEnable(GL_LIGHT7); // porneste a opta lumina
```

Pentru a specifica proprietatile unei surse de lumina se folosesc functiile

```
void glLightf(GLenum light, GLenum pname, GLfloat param);
```

```
void glLighti(GLenum light, GLenum pname, GLint param);
```

```
glLightfv(GL_LIGHTn, GL_POSITION, vector4f); // pozitie
```

De exemplu pentru culoarea ambientala a luminii 0:

```
GLfloat light_ambient[] = { 0.5, 0.5, 0.5, 1.0 };
```

```
glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
```

2. Extensii ale modelului de baza

OpenGL implementeaza si extensii ale acestui model de baza pentru a-l face mai util.

2.1. Atenuarea

Pentru majoritatea luminilor, intensitatea acestora scade o data cu cresterea distantei fata de sursa acestora. Atenuarea in OpenGL este calculata in felul urmatoar:

$$attenuationFactor = \frac{1}{k_C + k_L d + k_Q d^2}$$

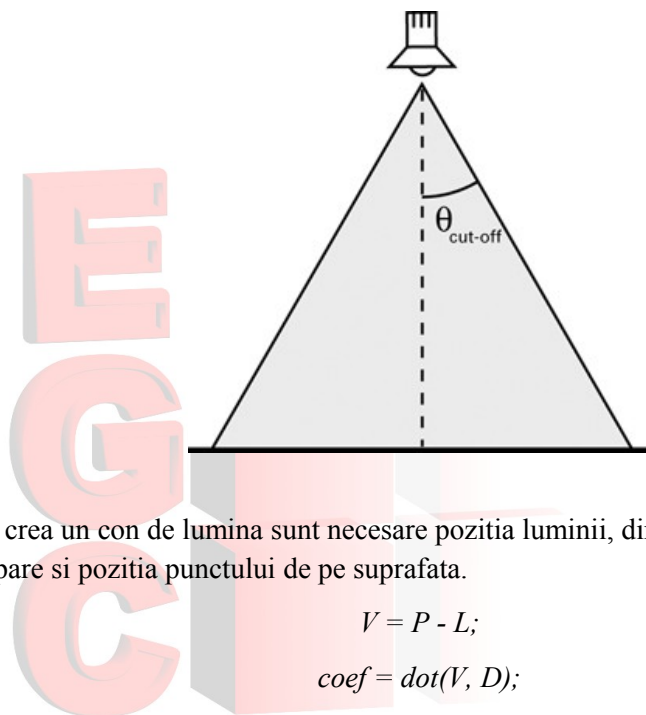
Apoi, aceasta atenuare este aplicata componentelor difuza si speculara a obiectului:

*diffuse *= attenuation; specular *= attenuation;*

```
glLightf(light, GL_CONSTANT_ATTENUATION, value);
glLightf(light, GL_LINEAR_ATTENUATION, value);
glLightf(light, GL_QUADRATIC_ATTENUATION, value);
```

2.2. Efectul de spot luminos

O alta extensie a modelului de baza implementata in OpenGL este crearea unui con de lumina in locul unei lumini omnidirectionale. Un unghi de decupare controleaza volumul in care obiectele sunt luminate.



Pentru a crea un con de lumina sunt necesare pozitia luminii, directia conului (axa acestuia), unghiul de decupare si pozitia punctului de pe suprafata.

```
V = P - L;
coef = dot(V, D);
if (cos(theta) <= coef)
    lightColor = 0;
else lightColor = C;
```

unde P = pozitia punctului de pe suprafata, L = pozitia luminii, D = directia spotului

Specificarea componentelor spotului:

```
glLightfv(light, GL_POSITION, vector4f);
glLightfv(light, GL_SPOT_DIRECTION, vector3f);
glLightf(light, GL_SPOT_CUTOFF, angle); // angle is 0 to 180
```

Observatie:

Iluminarea este dependenta de normalele la suprafata, drept urmare, pentru obiecte custom, este necesara specificarea normalelor la suprafata. Aceasta poate fi defnita per vertex sau per suprafata. In ambele cazuri, se foloseste functia

glNormal3f(x, y, z); // (x,y,z) = directia normalei

