

Argitasuna

<a.soraa@si.ehu.es>

EHU

2006ko aprilak 11

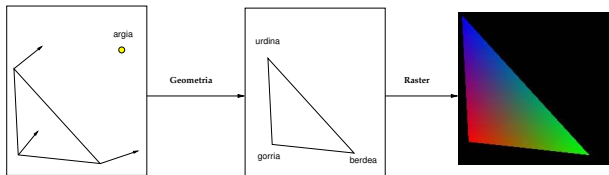
- 1 Argizatzea
- 2 Poligonoak argitatzeko teknikak. Itzaldura (shading).
- 3 Argizatze eredia
- 4 Lanbroa

Argiztatzea

- Errealitate itxura lortzeko argiaren eragina kontuan hartu behar da
- Hori lortzeko behar da:
 - Objektuaren zehaztapen zehatza
 - Argiaren efektu optikoaren **eredua**
- Lortu beharreko efektuak:
 - Argi islak
 - Gardentasuna
 - Ehundura (testurak)
 - Itzalak

Nola lortu argitasuna?

- Erpin bakoitzak kolore balio bat jasotzen du
 - Poligonoa interpolatuz irudikatu: itzaldura
- Erpinen koloreak zuzenean ezar ditzakegu, eskuz
- baina hobe da kontuan hartzea
 - Argi iturriak
 - Material ezaugarriak
 - Geometria



Itzaldura (shading)

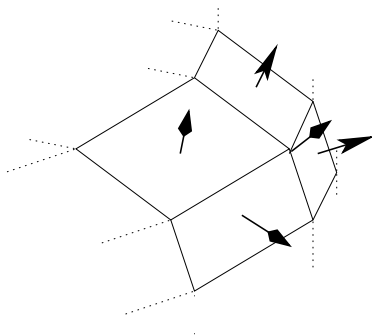
- Poligonoez osatutako objektetan koloreak esleitu
- Esan bezala, aurpegien erpinek dute koloreari inguruko informazioa
 - Barne-pixelen kolorea interpolatu.
- Aurpegi bakoitza independenteki koloreztatzen bada (*flat shading*): objektua ez da leuna.
- Aurpegien arteko interpolazioa!
 - *Gouraud*
 - *Phong*

Itzaldura konstantea (flat shading)

- Poligonoa osatzen duten pixelek kolore konstantea dute
- Kolorea kalkula daiteke jakinda:
 - I argiaren norabidea
 - n poligonoaren normala
 - v ikuslearen kokapena
- Kolorea kalkulatu, eta poligonoko pixel guztiei esleitu
- Era azkarragoak:
 - OpenGL: poligonoko erpin baten balioa hartu
- Arazoa: gainazal kurboak ezin dira irudikatu

Itzaldura interpolatua (interpolated shading).

- Poligono barneko puntu bakoitzean argiaren kalkuluak egin behar dira
- Poligono bakoitza leundu nahi da, ingurukoekin bat eginaz
- Bektore normala erpinetan da ezaguna



Gouraud

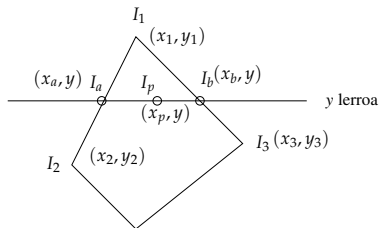
- Intentsitatea erpinetan kalkulatu eta barneko pixeletan interpolatu

$$I_a = \frac{1}{y_1 - y_2} (I_1(y - y_2) + I_2(y_1 - y))$$

$$I_b = \frac{1}{y_1 - y_3} (I_1(y - y_3) + I_3(y_1 - y))$$

$$I_p = \frac{1}{x_b - x_a} (I_a(x_b - x) + I_b(x - x_a))$$

- Eraginkortasuna handitzeko: $I_{x+1} = I_x + \Delta I_u = I_x + \frac{I_b - I_a}{x_b - x_a}$



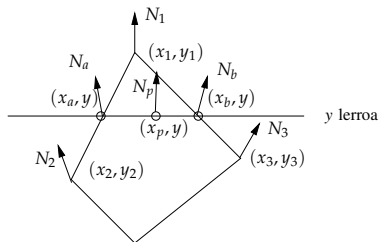
Phong

- Erpinetako bektore normala kalkulatu eta barneko pixeletan bektorea interpolatu, ondoren intentsitatea kalkulatzeko.

$$N_a = \frac{1}{y_1 - y_2} (N_1(y - y_2) + N_2(y_1 - y))$$

$$N_b = \frac{1}{y_1 - y_3} (N_1(y - y_3) + N_3(y_1 - y))$$

$$N_p = \frac{1}{x_b - x_a} (N_a(x_b - x) + N_b(x - x_a))$$



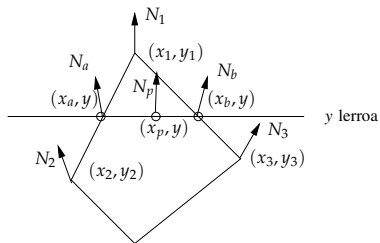
Phong

- Beste modu batean:

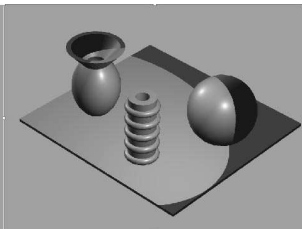
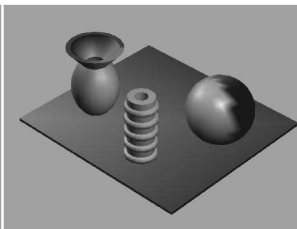
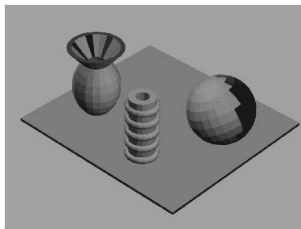
$$N_{px,x+1} = N_{px,x} + \Delta N_{ux}$$

$$N_{py,x+1} = N_{py,x} + \Delta N_{uy}$$

$$N_{pz,x+1} = N_{pz,x} + \Delta N_{uz}$$



Itzaldura ereduak. Flat, gouraud, phong



Itzaldura ereduak eta OpenGL

- OpenGL-k beti du itzaldura eredu bat
- Erpin bakoitzak normala bat behar du: `glNormal*` ()
- `glShadeModel()` funtzioa
 - `GL_FLAT`: interpolaziorik ez
 - Poligonoaren erpin baten normala hartzen du
 - `GL_SMOOTH`: gouraud (lehenetsia)
 - Poligonoko erpin orok du normala bat
- *Phong*-en ereduak garestia da *hw*-ean implementatzeko

Argiztatze eredua (*Phong*)

- Erpin bakoitzak jasotzen duen argia: hiru terminoen batura
- Ingurune argia
 - i_{amb} intentsitatea
 - Argi orokorra: Toki guztietara berdin heltzen da
 - Ad-hoc sortutakoa
- Isla barreiatua
 - i_{diff} intentsitatea
 - Objektuen kolorea ezartzen du
 - Ikuslearekiko independentea
- Ispilu-isa
 - i_{spec} intentsitatea
 - Zenbait puntutan, objektuek argia islatzen dute
 - Ikuslearekiko dependentea
- $i = i_{amb} + i_{diff} + i_{spec}$

Ingurune argia

- Esan bezala, *ad hoc* asmatutako intentsitatea
- Erpin bakoitzak kolore konstantea jasotzen du:

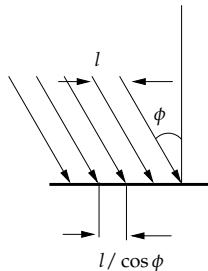
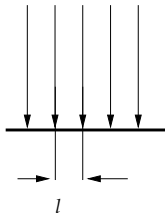
$$\mathbf{i}_{\text{amb}} = \mathbf{m}_{\text{amb}} \otimes \mathbf{s}_{\text{amb}}$$

- \mathbf{m}_{amb} materialaren ingurune-koefizientea (objektuarena)
- \mathbf{s}_{amb} eszenaren ingurune-koefizientea
- \otimes eragileak osagai bakoitza biderkatzen du, hots $\mathbf{a} = (a_x, a_y)$ eta $\mathbf{b} = (b_x, b_y)$, orduan $\mathbf{a} \otimes \mathbf{b} = (a_x b_x, a_y b_y)$

Isla barreiatua

- Lambert legea jarraitzen du: $\mathbf{i}_{\text{diff}} = n \cdot l = \cos\phi$
- Fotoiak norabide guztietan hedatzen dira (intentsitate desberdinekin)
- Ikuspuntuarekiko independentea

$$\mathbf{i}_{\text{diff}} = (n \cdot l) \mathbf{m}_{\text{diff}} \otimes \mathbf{s}_{\text{diff}}$$



Ispilu-isa

- Ispilu-islak objektuek argiaren islak adierazten ditu
 - Objektuaren errealitate-itxura
 - Material mota anitz
 - Ikuspuntuarekiko dependentea

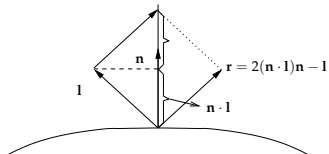
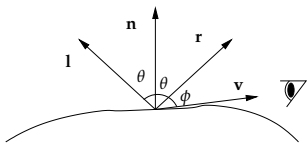


Ispilu-isola

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

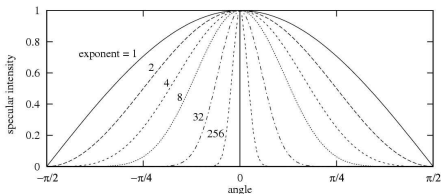
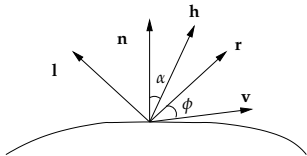
$$\mathbf{i}_{\text{spec}} = (\mathbf{r} \cdot \mathbf{v})^m = \cos(\phi + \theta)^m$$

$$\mathbf{i}_{\text{spec}} = \max(0, (\mathbf{r} \cdot \mathbf{v})^m) \mathbf{m}_{\text{spec}} \otimes \mathbf{s}_{\text{spec}}$$



Ispilu-isola

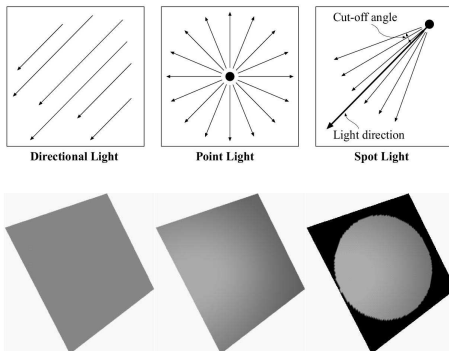
- \mathbf{r} kalkulatzea garestia izan daiteke.
- $\mathbf{h} = \frac{\mathbf{l} + \mathbf{v}}{\|\mathbf{l} + \mathbf{v}\|}$ bektorea: \mathbf{v} eta \mathbf{l} en arteko erdikaria
- $\mathbf{i}_{\text{spec}} = \max(0, (\mathbf{h} \cdot \mathbf{v})^m) \mathbf{m}_{\text{spec}} \otimes \mathbf{s}_{\text{spec}} = \max(0, \cos \alpha^m) \mathbf{m}_{\text{spec}} \otimes \mathbf{s}_{\text{spec}}$
- m k islako intentsitatearen ahuldura neurtzen du



Argi motak

Hiru argi mota

- Direkzioa soilik dutenak (*directional*)
- Posizioa soilik dutenak (*local*)
- Direkzio eta posizioa dutenak (*spotlight*)



Terminoak batuz

- $\mathbf{i}_{\text{tot}} = \mathbf{i}_{\text{amb}} + \mathbf{i}_{\text{diff}} + \mathbf{i}_{\text{spec}}$
- Intentsitatearen ahuldura:

$$d = \frac{1}{s_c + s_l \|\mathbf{s}_{\text{pos}} - \mathbf{p}\| + s_q \|\mathbf{s}_{\text{pos}} - \mathbf{p}\|^2}$$

non

- $\|\mathbf{s}_{\text{pos}} - \mathbf{p}\|$ argiaren \mathbf{s}_{pos} posizioa eta \mathbf{p} puntuaren arteko distantzia
- s_c : ahuldura konstantea
- s_l : ahuldura lineala
- s_q : ahuldura koadratikoa (errealitatean ematen dena)
- $s_c + s_l + s_q \leq 1$

orduan:

$$\mathbf{i}_{\text{tot}} = \mathbf{i}_{\text{amb}} + d(\mathbf{i}_{\text{diff}} + \mathbf{i}_{\text{spec}})$$

- Errealitatean $s_c = s_l = 0$, eta $s_q = 1$

Terminoak batuz

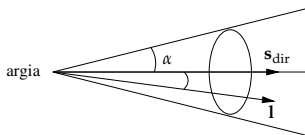
- *spotlight* motako argiak

$$c_{\text{spot}} = \max(-\mathbf{l} \cdot \mathbf{s}_{\text{dir}}, 0)^{s_{\text{exp}}}$$

$$\mathbf{i}_{\text{tot}} = c_{\text{spot}}(\mathbf{i}_{\text{amb}} + d(\mathbf{i}_{\text{diff}} + \mathbf{i}_{\text{spec}}))$$

non

- \mathbf{l} : argiaren bektorea
- \mathbf{s}_{dir} : konoaren bektorea
- s_{exp} : ahuldura-faktorea
- Argia *spotlight* motakoa ez bada, $c_{\text{spot}} = 1$



Terminoak batuz

- Ekuazioari gehituko dugu:
 - \mathbf{a}_{glob} : “global ambient light source”
 - \mathbf{m}_{emi} : objektuek igortzen duten argi intentsitatea

$$\mathbf{i}_{\text{tot}} = \mathbf{a}_{\text{glob}} \otimes \mathbf{m}_{\text{amb}} + \mathbf{m}_{\text{emi}} + c_{\text{spot}}(\mathbf{i}_{\text{amb}} + d(\mathbf{i}_{\text{diff}} + \mathbf{i}_{\text{spec}}))$$

Argiak eta OpenGL

- Argiztapen-eredua `glLightModel*()`
- 4 osagai
 - “global ambient light” \mathbf{a}_{amb}
 - Ikusleak kokapen bat duen edo infinituan kokatuta dagoen (ispilu-islentzat). Normalean infinitua (objektuen ertzen eta ikuslearen kokapearen arteko angelua konstantea)
 - Argia desberdina ote den aurreko/atzeko aurpegientzat
 - Ispilu-ista era desberdin batean tratatu behar ote den, ista barreiatua eta ingurune argietatik (testurentzat)

Argiak eta OpenGL

- `glLight*()` funtzioak ezartzen ditu argien parametroak. Besteak beste:
 - Ingurune intentsitatea: s_{amb}
 - Isla barreiatuaren intentsitatea: s_{diff}
 - Ispilu-islaren intentsitatea: s_{spec}
 - Posizioa
 - “spotlight” norabidea: s_{dir}
- 8 argi defini daitezke (`GL_LIGHT0...GL_LIGHT7`)
- Argiak “pizteko”: `glEnable(GL_LIGHTING)`
- Argiak “itzaltzeko”: `glDisable(GL_LIGHTING)`

Materialak eta OpenGL

- Modu erraza: kolore bakarra erabili kanal bakoitzean
 - Kanala aukeratu: `glColorMaterial()`
 - Kolorea aukeratu: `glColor*()`
 - OpenGL modu “errazan” jarri: `glEnable(GL_COLOR_MATERIAL)`
- Adibidez:

```
glColorMaterial(GL_FRONT, GL_DIFFUSE) // kanal barreiatua  
glColor3f(0.2, 0.5, 0.8) // Kolorea ezarri  
glEnable(GL_COLOR_MATERIAL) // Modu hau ezarri
```

Materialak eta OpenGL

- Modu orokorra: `glMaterial*()`
 - Ingurunea: \mathbf{m}_{amb}
 - Isla barreiatua: \mathbf{m}_{diff}
 - Ispilu-isa: \mathbf{m}_{spec}
 - “Shininess”: m
 - Igorpena: \mathbf{m}_{emi}
- Objektuek material-parametro desberdina izan dezakete (objektuak marraztu aurretik beren material parametroak ezarri)

Materialak eta OpenGL

Material tipikoen balioak:

Materiala		GL_AMBIENT	GL_DIFFUSE	GL_SPECULAR	GL_SHININESS
Brass	R	0.329412	0.780392	0.992157	27.8974
	G	0.223529	0.568627	0.941176	
	B	0.027451	0.113725	0.807843	
	A	1.0	1.0	1.0	
Bronze	R	0.2125	0.714	0.393548	25.6
	G	0.1275	0.4284	0.271906	
	B	0.054	0.18144	0.166721	
	A	1.0	1.0	1.0	
Gold	R	0.24725	0.75164	0.628281	51.2
	G	0.1995	0.60648	0.555802	
	B	0.0745	0.22648	0.366065	
	A	1.0	1.0	1.0	

Gehiago hemen: http://www.cs.utk.edu/kuck/materials_ogl.htm

Alfa balioa

- Alfa balioak kolore baten gardentasuna neurtzen du
- Objektu gardenak irudikatzeko, zatien koloreak “nahastu” egin behar dira (*blending*)
- Izan bitez
 - Iturriaren kolorea (*source*) (R_s, G_s, B_s, A_s) , kolore berria
 - Xede kolorea (*destination*) (R_d, G_d, B_d, A_d) , dagoeneko pantailan dagoen kolorea
- Bi urrats:
 - 1 Bai iturriko zein xedeko “nahasmen” faktoreak kalkulatu
 - Iturriko nahasmen-faktoreak: $(\sigma_r, \sigma_g, \sigma_b, \sigma_a)$
 - Xedeko nahasmen-faktoreak: $(\delta_r, \delta_g, \delta_b, \delta_a)$
 - 2 Margotuko den kolore berria kalkulatu
 - $(\sigma_r R_s + \delta_r R_d, \sigma_g G_s + \delta_g G_d, \sigma_b B_s + \delta_b B_d, \sigma_a A_s + \delta_a A_d)$

Nahasketa OpenGLn

- `glBlendFunc(GLenum sfactor, GLenum dfactor)`

`sfactor` iturriaren nahasmen-faktorea

`dfactor` xedearen nahasmen-faktorea

Konstantea	F. esanguratsua	Nahasmen-faktorea
GL_ZERO	iturria edo xedea	$(0, 0, 0, 0)$
GL_ONE	iturria edo xedea	$(1, 1, 1, 1)$
GL_DST_COLOR	iturria	(R_d, G_d, B_d, A_d)
GL_SRC_COLOR	xedea	(R_s, G_s, B_s, A_s)
GL_ONE_MINUS_DST_COLOR	iturria	$(1, 1, 1, 1) - (R_d, G_d, B_d, A_d)$
GL_ONE_MINUS_SRC_COLOR	xedea	$(1, 1, 1, 1) - (R_s, G_s, B_s, A_s)$
GL_SRC_ALPHA	iturria edo xedea	(A_s, A_s, A_s, A_s)
GL_ONE_MINUS_SRC_ALPHA	iturria edo xedea	$(1, 1, 1, 1) - (A_s, A_s, A_s, A_s)$
GL_DST_ALPHA	iturria edo xedea	(A_d, A_d, A_d, A_d)
GL_ONE_MINUS_DST_ALPHA	iturria edo xedea	$(1, 1, 1, 1) - (A_d, A_d, A_d, A_d)$
GL_SRC_ALPHA_SATURATE	iturria	$(f, f, f, 1); f = \min(A_s, 1 - A_d)$

Nahasketa OpenGLn

- Nahasketa aktibatu/desaktibatu
 - `glEnable (GL_BLEND) / glDisable (GL_BLEND)`
- Adib: Bi irudi nahastu %50
 - 1 Marraztu lehenengo irudia (nahasketarik gabe)
 - 2 `glEnable (GL_BLEND)`
 - 3 `glBlendFunc (GL_SRC_ALPHA, GL_SRC_ALPHA)`
 - 4 Marraztu bigarren irudia, alfa balioa 0.5 dela
- Adib: Lehenengo irudia %25, bigarrena %75
 - 1 Marraztu lehenengo irudia lehen bezala
 - 2 `glEnable (GL_BLEND)`
 - 3 `glBlendFunc (GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)`
 - 4 Marraztu bigarren irudia, alfa balioa 0.75 dela

Nahasketa OpenGLn

- Maiz erabilitako faktoreak

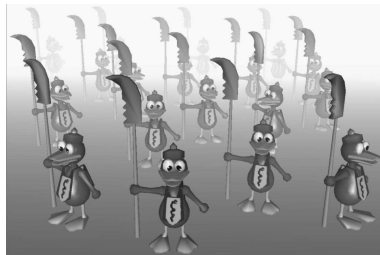
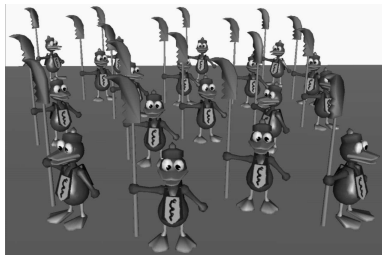
- $A_s(R_s, G_s, B_s, A_s) + (1 - A_s)(R_d, G_d, B_d, A_d)$
`glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)`
- $A_s(R_s, G_s, B_s, A_s) + (R_d, G_d, B_d, A_d)$
`glBlendFunc(GL_SRC_ALPHA, GL_ONE)`
- $((1 - R_d)R_s, (1 - G_d)G_s, (1 - B_d)B_s, (1 - A_d)A_s) + (R_d, G_d, B_d, A_d)$
- Zein da ?? (ariketa)

Nahasketa eta 3D irudiak

- Objektu gardenak marrazteko ordena oso garrantzitsua da
 - 1 `glEnable(GL_DEPTH_TEST);` //Z-buffer aktibatu
 - 2 Objektu opakoak marraztu
 - 3 `glEnable(GL_BLEND);`
 - 4 *nahi den nahasmen funtzioa ezarri*
 - 5 `glDepthMask(0);` //Z-buffer irakurtzeko soilik
 - 6 Objektu gardenak marraztu
 - 7 `glDepthMask(1);` //Z-buffer irakurri/idatzi
 - 8 `glDisable(GL_BLEND);`

Lanbroa

- Efektua: ikuspuntearekiko distantzia zenbat eta handiagoa izan, orduan eta gutxiago ikusiko da
- Eszenari errealismoa emateko
- Urruneko objektuak era atseginean desagertzeko (*culling*)



Lanbroa

- Izan bitez:
 - \mathbf{c}_p pixelaren kolorea
 - \mathbf{c}_{fog} lanbroaren kolorea (grisa normalean)
 - $\phi \in [0, 1]$ lanbroaren koefizientea, ikuspuntuarekiko proportzionala

Pixelaren kolorea lanbroa kontuan hartuta:

$$\mathbf{c}'_p = (1 - \phi)\mathbf{c}_p + \phi\mathbf{c}_{\text{fog}}$$

- ϕ kalkulatzeko

$\phi = e^{-\lambda z}$	GL_EXP
$\phi = e^{-(\lambda z)^2}$	GL_EXP2
$\phi = \frac{z-z_0}{z_1-z_0}$	GL_LINEAR

Lanbroa eta OpenGL

- `glFog* ()`
 - Lanboraren kolorea aukeratu (`glFogfv (GL_FOG_COLOR, GLfloat *)`)
 - Lanboraren ereduak aukeratu (`glFogi (GL_FOG_MODE, GL_EXP | GL_EXP2 | _GL_LINEAR)`)
- **aktibatu:** `glEnable (GL_FOG)`
- **desgaitu:** `glDisable (GL_FOG)`