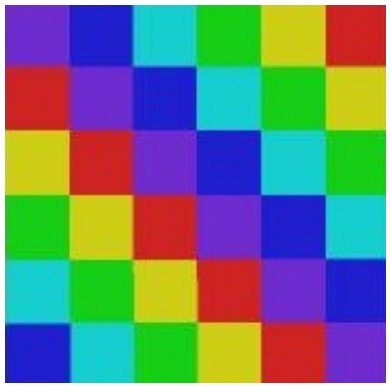
Aurkezpenaren eskema

Testurak

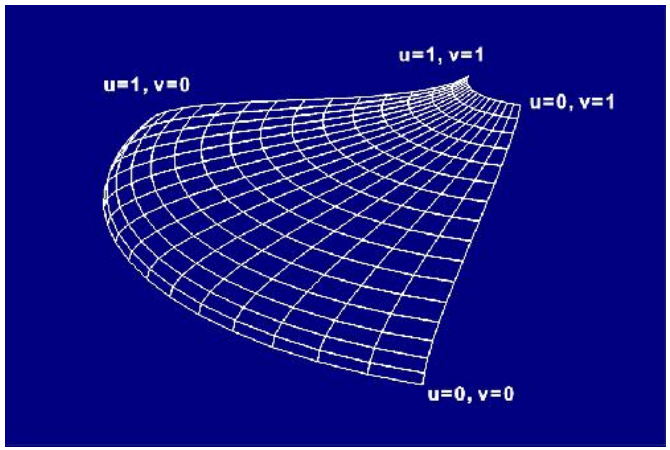
- Helburua: objektuen itxura errealagoa lortzea
  - eta garestia ez izatea
- Objektu geometrikoei 2Dko irudiak esleitu.
  - Obj. geometriko usuena hirukia da: interpolazioa biraketarekiko inbariantea da.
- Testurak erabil daitezke hamaika gauza egiteko:
  - Erabilpen *arrunta*
  - *Bump mapping*
  - *Environment mapping*
  - *Lightmaps*
  - etab . . .

Testuren mapaketa



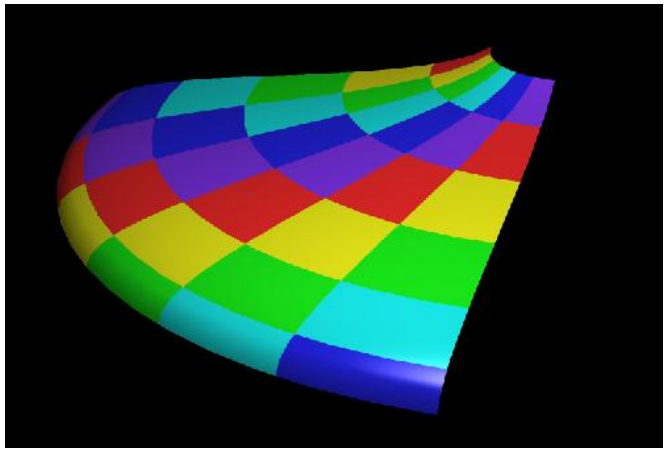
Irudia

+



Gainazal parametrikoa

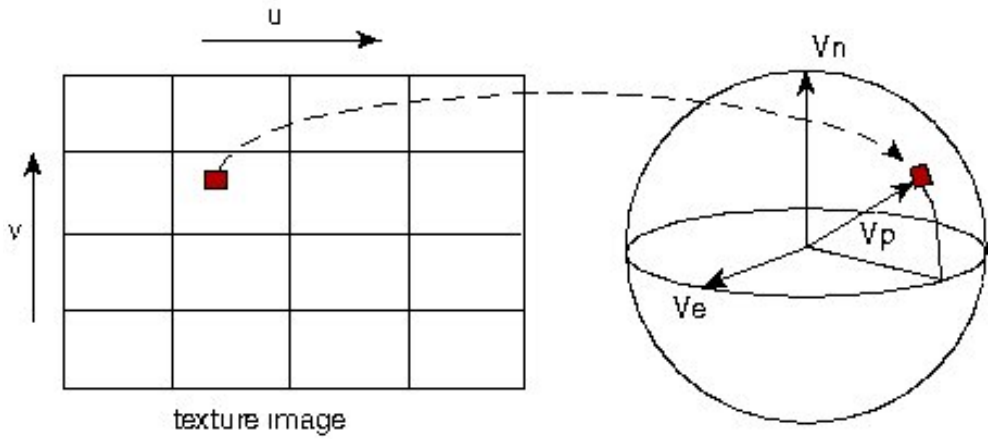
=



Gainazal testurekin

Testuren mapaketa. Proiekzioa.

Irudiaren zati bat gainazalaren zati batean “mapatzen” da

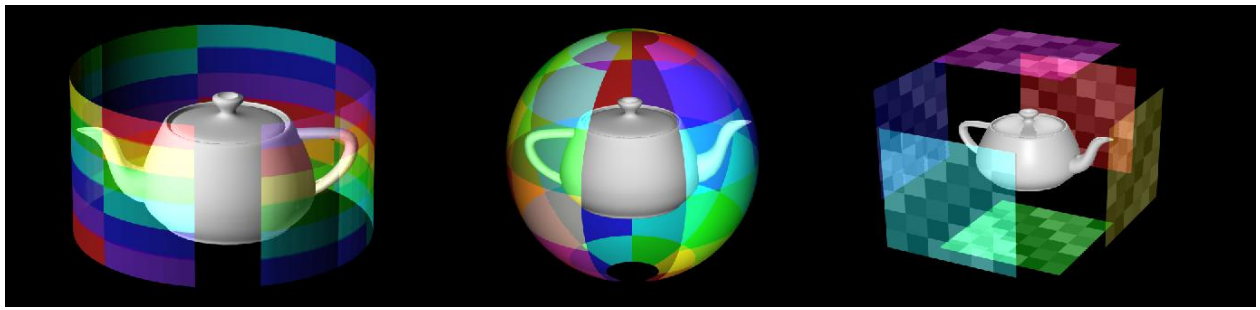


Paul Rademacher http://www.cs.unc.edu/ rademach/xroads-RT/RTarticle.html

Testuren mapaketa. Proiekzioa.

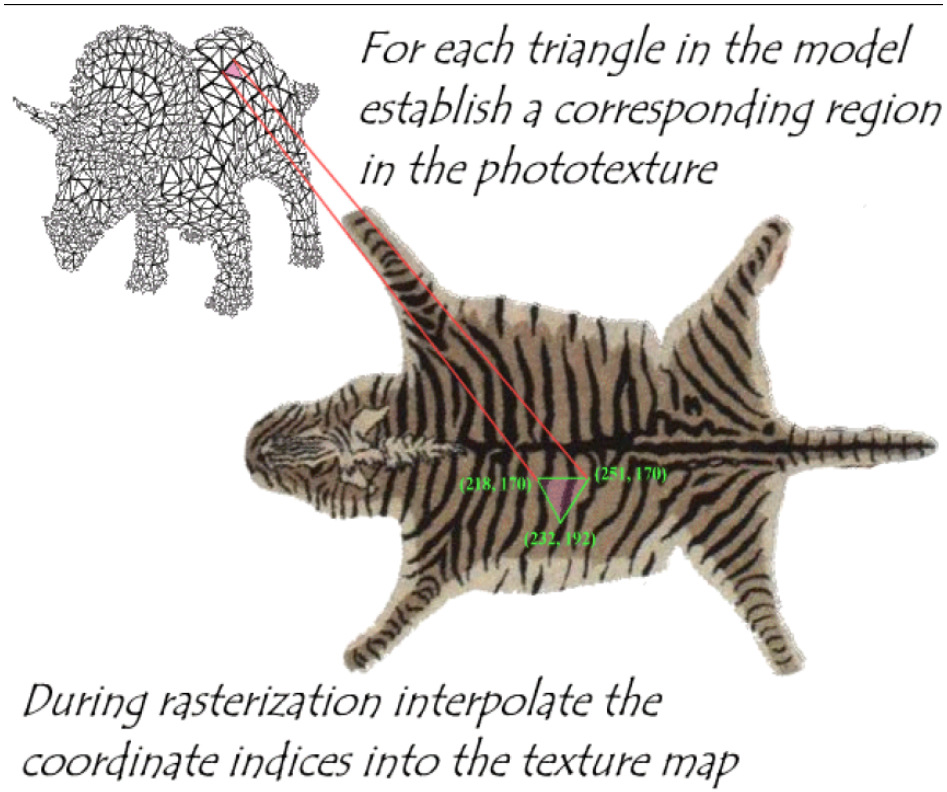
- Mapaketa erraza da gainazal parametriko bat badugu. Ez, ordea, poligonoak baditugu!
- Irtenbidea: proiekzioa. Lau proiekzio erabiltzen dira usuen: planarra. zilindrikoa, esferikoa eta kubikoa
- Proiekzioa preprozesatu ohi da, eta geometriarekin batera gorde.
  - Erpin bakoitzari zein testura-koordenatu dagokion gorde

Zilindrikoa Esferikoa Kunikoa

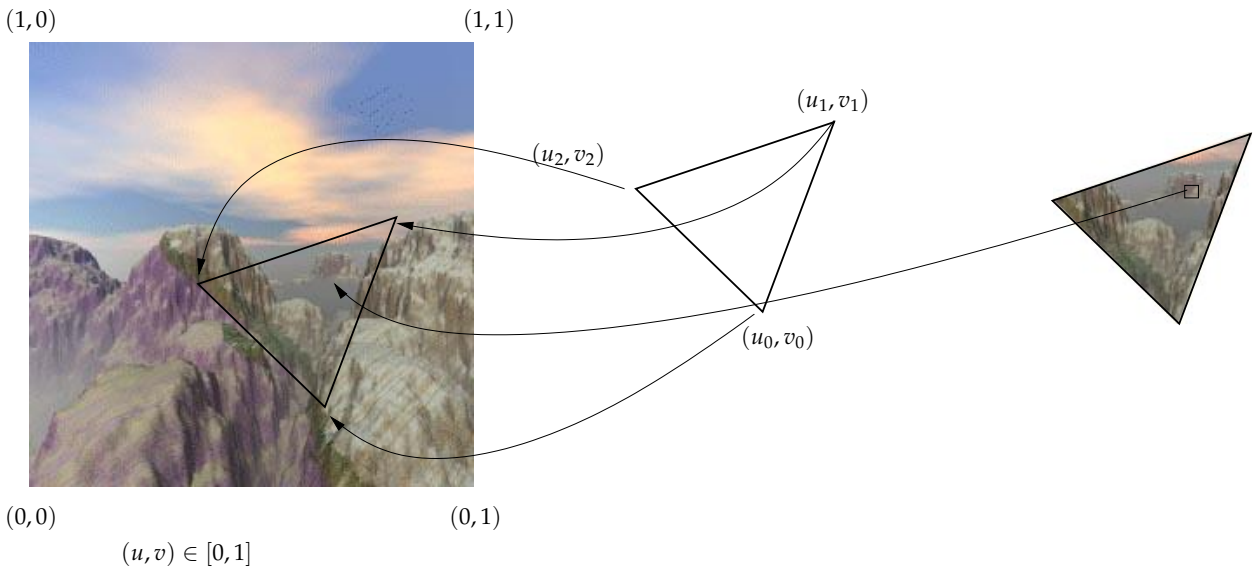


Testuren mapaketa. Proiekzioa.

- Edo proiekzioa eskuz zehaz daiteke
  - erpin bakoitzaren (s,t) testura-koordenatuak esplizituki adierazi.



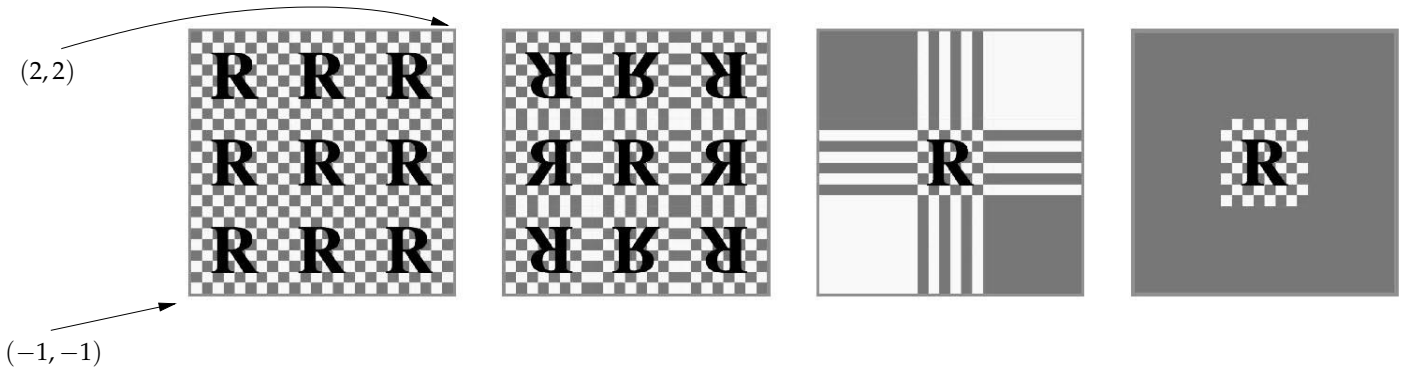
Testuren mapaketa



- Erpin bakoitzak (u, v) testura-koordenatuak ditu
  - Barneko pixelak interpolatu egiten dira

Testuren koordenatuak

- Testuren koordenatuak $\in [0,1]^2$
  - Testura-pixelei *texel* esaten zaie
- Zer gertatuko litzateke $(u,v) > 1.0$ edo < 1.0?
- Testurak errepikatzeko zati frakzionala soilik erabili
- Lau modu nagusi: *repeat, mirror, clamp, border*



Testurak esleitzeko moduak

- Poligonoko pixelek jasoko duten intentsitatea bi faktoreen arabera dira:
  - Argi zein materialen arabera (argitasuna): isla barreiatua eta ispili-isla
  - Testuraren kolorea
- Faktore bakoitzak α balio bat izan dezake
- Testura-kolorea (gehi alpha) gainazalean ezartzeko modu anitz. Usuenak:
  - *replace* (ordezkatu): gainazalak zuen kolorea testurak duenarekin ordezkatu. Ispilu-isla desagertzen da.
  - *decal*: testurak alpha balioa izan behar du (maskara). Testura gainazalak duen kolorearekin nahasten da, testuraren alpha balioaren arabera. Gainazalaren alpha balioa ez da ukitzen.
  - *modulate*: testura kolorea eta gainazalarena biderkatu.

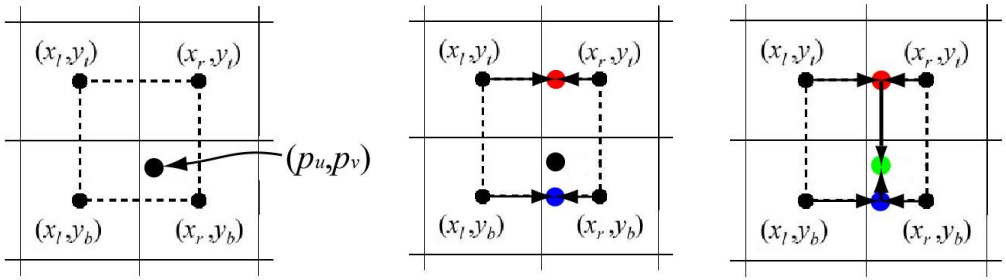
Aliasing. Magnification.

- Testura-irudia poligonoak duen azalera (pantailan) baino *txikiagoa* denean.
  - Pixelak handiak ikusiko dira
- Bi teknika:
  - *Nearest neighbor*: texel hurbilena erabili
  - *Bilinear interpolation*: 2Dko texel-en arteko batazbestekoa

Bilinear interpolation

2Dtan gauzatutako interpolazio lineala

- Testura-koordenatuak: $(p_u, p_v) \in [0,1]$
- $(m \times n)$ texel
- Texel hurbilena: $(\lfloor m \cdot p_u \rfloor, \lfloor n \cdot p_v \rfloor)$
- 1D interpolazioa x eta yn



Bilinear interpolation

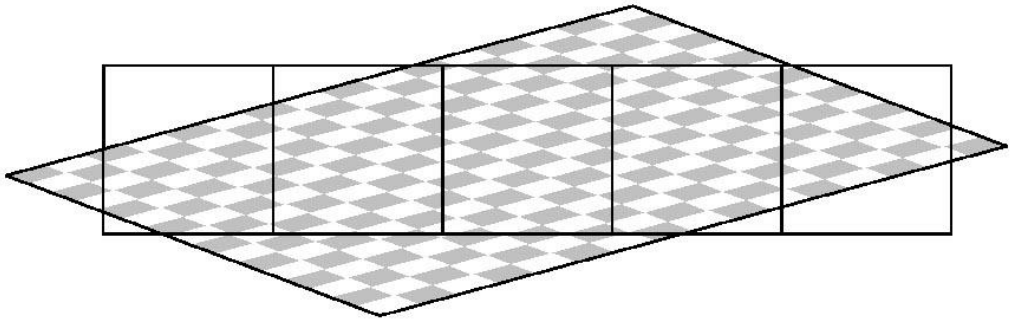
Formulazioa:

- $\mathbf{t}(u,v)$ testura eskuratzeko
- $\mathbf{b}(u,v)$ testura filtratua
- $x_l = \lfloor p_u \rfloor$; $x_r = \lfloor p_u + 1 \rfloor$;
- $y_b = \lfloor p_v \rfloor$; $y_y = \lfloor p_v + 1 \rfloor$;
- $(u', v') = (p_u - \lfloor p_u \rfloor, p_v - \lfloor p_v \rfloor)$

$$\begin{aligned}\mathbf{b}(u,v) \quad = \quad & (1-u')(1-v')\mathbf{t}(x_l, y_b) + \\ & u'(1-v')\mathbf{t}(x_r, y_b) + \\ & (1-u')v'\mathbf{t}(x_l, y_t) + \\ & u'v'\mathbf{t}(x_r, y_t)\end{aligned}$$

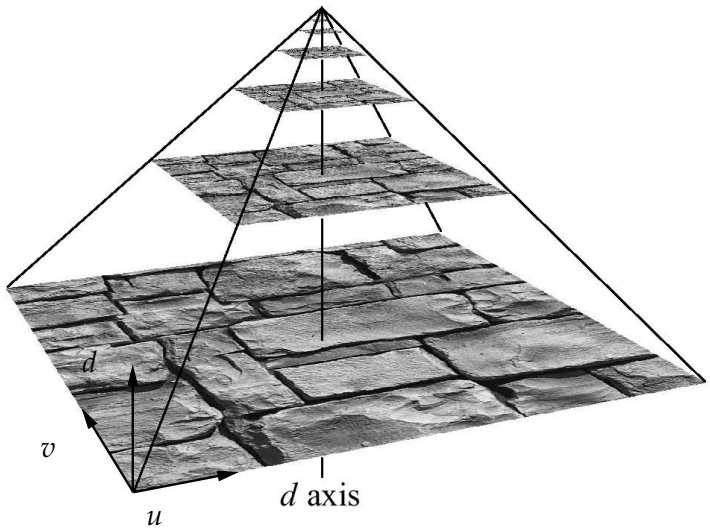
Aliasing. Minification

- Testura-irudia poligonoak duen azalera (pantailan) baino *handiagoa* denean.
- Magnification baino arazo zailagoa da hau
  - *Nearest neighbor*: oso emaitza kaxkarrak
  - *Bilinear interpolation*: emaitzak ez dira onak
- Usuen erabilitako teknika: *MipMapping*



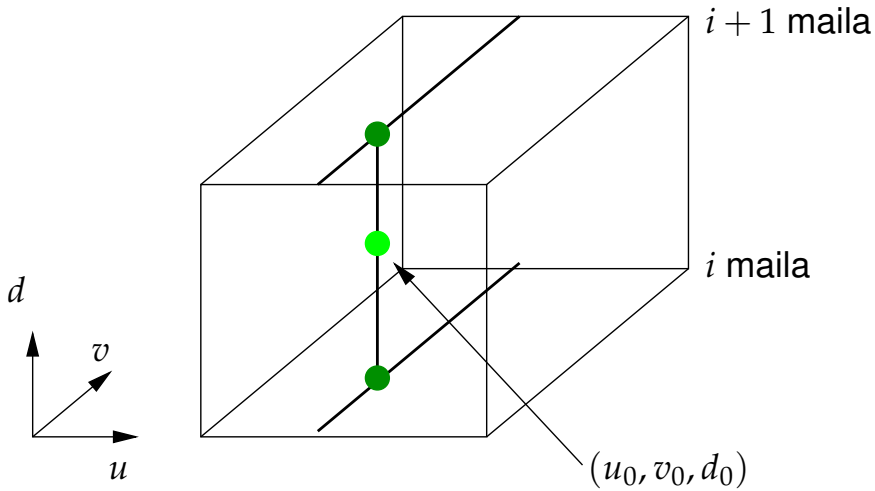
Mipmapping

- Irudiaren piramide bat egiten da
  - 0. mailan ($d = 0$) irudia bera dago
  - $i + 1$ mailako irudiak i.-aren tamaina erdia du
  - gurasoaren 4 texelak konbinatuz lortzen dire umeen texel-ak
  - testura zenbat eta urrunago egon, orduan eta maila handiago erabili
  - d altueraren kalkuluak bi irudi emango dizkigu
    - Bakoitzean “bilinear interpolation” gauzatu



Mipmapping. Trilinear interpolation.

- Lortutako bi balioen arteko interpolazioa egin: trilinear interpolation
- Denbora konstantean gauzatzen da.
- 8 texelen arteko interpolazioa
- Arazoa: nola kalkulatu d?



Multitexturing

- Objektu bati testura bat baino gehiago esleitzea
- Abantaila handiak ditu
  - Testura-mota anitz: arrunta, *bump*, *lightmap*
- Adib: *quake III* jokoaren motorrak 10 testura eslei dakieke objektuei

  1-4 *Bump mapping*erako
  5 isla barreiatua
  6 testura
  7-10 isla-ispilua, igorpena, efektu atmosferikoak, *screen slashes*
- Abiadiraren arabera, zenbait urrats ez da egingo (1-4 eta 7-10 hautazkoak dira)

Testurak eta OpenGL

- Hainbat dimentsiotako testurak: 1D, 2D eta 3D
- Zenbait urrats bete behar dira:
  - Testura-objektu bat sortu, eta objektuari testura bat esleitu
  - Ezarri testura aplikatzeko era
    - Testura-koordenatuak $[0, 1]$etik at daudenean zer egin
    - Antialiasing-a nola gauzatu
    - Testurak nola nahastu
  - Testurak aktibatu
  - Objektua marraztu, erpinen geometria zein testura koordenatuak zehaztuz

Testurak sortu eta zehaztu

- Testura-identifikatzaile bat lortu

  `glGenTextures(int n, GLuint *id)`
  `id` aldagaian gordeko da testuraren identifikatzailea

- Uneko testura aukeratu
  - `glBindTexture( GL_TEXTURE_2D, id)`
  - `id` berria bada, testura sortu egiten du. Aurretik sortua bazegoen, uneko testura bihurtzen du.

Testura-irudia

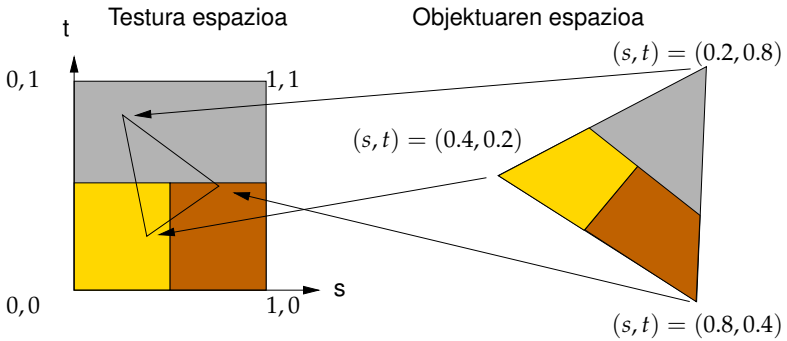
`glTexImage2D(target, level, components, w, h, border, format, type, texels`

- `target`: testura-mota. Adib: `GL_TEXTURE_2D`
- `level`: *mipmapping*-erako erabilia (bestela 0)
- `components`: zenbat osagai *texel* bakoitzeko (RGB:3, RGBA:4)
- `w, h`: zabalera eta altuera. Bakoitzak 2^n izan behar du (minimo 64). Erresoluzioa ez bada horrela, `gluScaleImage` funtzioa erabil daiteke.
- `border`: testurako borneen tamaina (besterik ezean 0)
- `format, type`: *texel*ak deskribatzen dituzte
- `texels`: irudia gordeta dagoen tokia

- Adibidez:
  `glTexImage2D( GL_TEXTURE_2D, 0, 3, 64, 64, 0, GL_RGB, GL_UNSIGNED_BYTE, irudia)`

Testura-koordenatuak

- Testurak marrazteko, aktibatu egin behar dira:
  `glEnable(GL_TEXTURE_2D)`
- Erpin bakoitzak (u, v) testura-koordenatu bat jasotzen du:
  `glTexCoord2f`
- Nahi bada, koordenatuen gainean matrize-aldaketa bat ezar daiteke:
  `glMatrixMode(GL_TEXTURE);`
  - Mugitu, biratu, tamaina aldatu
  - Testurak proiektatu.



Testuraren zehaztapena: errepikapenak

`glTexParameteri(GLenum target, GLenum pname, TYPE param)` funtzioa

Parametroa	Balio posibleak
GL_TEXTURE_WRAP_S	`GL_CLAMP`, `GL_REPEAT`
GL_TEXTURE_WRAP_T	`GL_CLAMP`, `GL_REPEAT`

Testuraren zehaztapena: aliasing

`glTexParameteri(GLenum target, GLenum pname, TYPE param)` funtzioa

Parametroa	Balio posibleak
`GL_TEXTURE_MAG_FILTER`	`GL_NEAREST`, `GL_LINEAR`
GL_TEXTURE_MIN_FILTER	`GL_NEAREST`, `GL_LINEAR`

- `NEAREST`: pixel hurbilena
- `LINEAR`: *bilinear interpolation*

Mipmapping

- Maila bakoitzeko irudia definitu behar da `glTexImage2D` funtzioaren bidez
  - *level, width, heigth* eta *image* parametroak zehaztu behar dira
- OpenGl-k kudeatuko du zein maila erabili, testura eta poligionoaren erresoluzioaren arabera
- `glTexParameteri` funtzioaren bidez zehaztuko dugu zein filtro erabili
- Irudi bat emanda piramidea automatikoki sortzen duen funtzio lagungarria: `gluBuild2DMipmaps`

Mipmapping filtroak

Parametroa	Balio posibleak
`GL_TEXTURE_MIN_FILTER`	`GL_NEAREST_MIPMAP_NEAREST`, `GL_NEAREST_MIPMAP_LINEAR`, `GL_LINEAR_MIPMAP_NEAREST`, `GL_LINEAR_MIPMAP_LINEAR`

Balioa	maila	pixel
`GL_NEAREST_MIPMAP_LINEAR`	bilinear	hurbilena
`GL_LINEAR_MIPMAP_LINEAR`	bilinear	bilinear
`GL_LINEAR_MIPMAP_NEAREST`	hurbilena	hurbilena
`GL_NEAREST_MIPMAP_NEAREST`	hurbilena	bilinear

- Modu garestiena: `GL_LINEAR_MIPMAP_LINEAR`
  - *Trilinear interpolation*

Nahasketa

`glTexEnv{if}{v}(GLenum target, GLenum pname, TYPE param)` funtzioa

- `target` izan behar du `GL_TEXTURE_ENV`
- `pname` aldagaiak `GL_TEXTURE_ENV_MODE` balio badu, nahasketa-funtzioa ezarriko da
  - `param` balio posibleak `GL_DECAL`, `GL_MODULATE`, `GL_BLEND`, `GL_REPLACE`
- `pname` aldagaiak `GL_TEXTURE_ENV_COLOR` balio badu
  - `param` $\mathbf{C}_c = (R_C, G_C, B_C, A_C)$ 4-kote batera apuntatu behar du
  - $\mathbf{C}_C$ bakarrik erabiliko da nahasketa-funtzioa `GL_BLEND` bada

Nahasketa

- Hiru kolore iturri:

Testura ($\mathbf{C}_t$)	os. 1	2 os.	3 os.	4 os.
	L_t	L_t, A_t	(R_t, G_t, B_t)	(R_t, G_t, B_t, A_t)
Pantaila ($\mathbf{C}_f$)	(R_f, G_f, B_f, A_f)			
Finkoa ($\mathbf{C}_c$)	(R_C, G_C, B_C, A_C)			

- Nahasketa posibleak

Osagai. kop.	*Decal*	*Modulate*	*Blend*	*Repace*
1	–	$\mathbf{C} = L_t\mathbf{C}_f$ $A = A_f$	$\mathbf{C} = (1 - L_t)\mathbf{C_f} + L_t\mathbf{C}_c$ $A = A_f$	$\mathbf{C} = L_t$ $A = A_f$
2	–	$\mathbf{C} = L_t\mathbf{C}_f$ $A = A_tA_f$	$\mathbf{C} = (1 - L_t)\mathbf{C_f} + L_t\mathbf{C}_c$ $A = A_tA_f$	$\mathbf{C} = \mathbf{C}_f$ $A = A_t$
3	$\mathbf{C} = \mathbf{C}_t$ $A = A_f$	$\mathbf{C} = \mathbf{C}_t\mathbf{C}_f$ $A = A_f$	$\mathbf{C} = (1 - \mathbf{C}_t)\mathbf{C}_f + \mathbf{C}_c\mathbf{C}_t$ $A = A_f$	$\mathbf{C} = \mathbf{C}_t$ $A = A_f$
4	$\mathbf{C} = \mathbf{C}_t$ $A = A_f$	$\mathbf{C} = \mathbf{C}_t\mathbf{C}_f$ $A = A_fA_t$	$\mathbf{C} = (1 - \mathbf{C}_t)\mathbf{C}_f + \mathbf{C}_c\mathbf{C}_t$ $A = A_fA_t$	$\mathbf{C} = \mathbf{C}_t$ $A = A_f$