Számítógépes Grafika

Klár Gergely
tremere@elte.hu

Eötvös Loránd Tudományegyetem
Informatikai Kar

2010/2011. őszi félév

Tartalom

1. Textúra leképezések
   - Paraméterezés
   - Sugárkövetés
   - Inkrementális képszintézis
2. Textúrák szűrése
3. Procedurális textúrák
4. Nem-szín textúrák

Mi az a textúrázás?

- Eddig: *egyszínű* anyag modellek, azaz a teljes felület azonos színű.
- Kevés ilyen van a valóságban.
- Finom részleteket szeretnénk megadni.
- Változó paraméterekre van szükségünk a BRDF-ben.
- Ezeket a paramétereket, elsősorban színt, adjuk meg a *textúrákban*.

Textúra megadási módok

- ”Tömbbel”: valamilyen 1/2/3 dimenziós tömbből olvasunk, elemei a *texel*-ek.
- Függvénnyel adjuk meg: ezt nevezzük *procedurális textúrázásnak*.

Leképezési módok

- Két terünk van: *képtér* és *textúratér*.
- Honnan hova képzünk?
- Textúratér → képtér: *textúra alapú leképezés*
  - » Minden *texelhez* keresünk a neki megfelelő *pixelt*.
  - ⊕ Hatékony.
  - ⊖ Nem biztos, hogy minden pixel sorra kerül, és ami mégis, nem biztos, hogy csak egyszer.
- Képtér → textúratér: *képtér alapú leképezés*
  - » Minden *pixelhez* keresünk a neki megfelelő *texelt*.
  - ⊕ Inkrementális képszintézishez jól passzol.
  - ⊖ Szükség van hozzá a paraméterezési és vetítési transzformációk inverzére.

Leképezési módok

- Két terünk van: *képtér* és *textúratér*.
- Honnan hova képzünk?
- Textúratér → képtér: *textúra alapú leképezés*
  - » Minden *texelhez* keresünk a neki megfelelő *pixelt*.
  - ⊕ Hatékony.
  - ⊖ Nem biztos, hogy minden pixel sorra kerül, és ami mégis, nem biztos, hogy csak egyszer.
- Képtér → textúratér: *képtér alapú leképezés*
  - » Minden *pixelhez* keresünk a neki megfelelő *texelt*.
  - ⊕ Inkrementális képszintézishez jól passzol.
  - ⊖ Szükség van hozzá a paraméterezési és vetítési transzformációk inverzére.

Leképezési módok

- Két terünk van: *képtér* és *textúratér*.
- Honnan hova képzünk?
- Textúratér → képtér: *textúra alapú leképezés*
  - » Minden *texelhez* keresünk a neki megfelelő *pixelt*.
  - ⊕ Hatékony.
  - ⊖ Nem biztos, hogy minden pixel sorra kerül, és ami mégis, nem biztos, hogy csak egyszer.
- Képtér → textúratér: *képtér alapú leképezés*
  - » Minden *pixelhez* keresünk a neki megfelelő *texelt*.
  - ⊕ Inkrementális képszintézishez jól passzol.
  - ⊖ Szükség van hozzá a paraméterezési és vetítési transzformációk inverzére.

Paraméterezés

- Hogyan tudjuk eldönteni, hogy az egyes felületi pontokhoz a tömb melyik elemét választjuk, vagy a függvényt milyen paraméterekkel értékeljük ki?
- A felület minden pontjához *textúra koordinátákat* rendelünk.
- A textúra koordináták hozzárendelését a felülethethez nevezzük *paraméterezésnek*.
- A továbbiakban 2D textúrákról beszélünk.

Parametrikus felületek paraméterezése

- Parametrikus felület:

$$F \in \mathbb{R}^2 \to \mathbb{R}^3, \quad F(u,v) := (x,y,z)$$

- u, v paraméterek természetes módon használhatók textúra koordinátáknak.
- Ha $\mathcal{D}_F \neq \mathcal{D}_{tex}$, akkor u, v transzformálni kell.
- PI:
  - Henger palást
  - $\mathcal{D}_F = [0, 2\pi] \times [0, h]$, $F(u,v) := (\cos u, h, \sin u)$
  - Textúra tér: $(\bar{u}, \bar{v}) \in [0,1] \times [0,1]$
  - Transzformáció: $\bar{u} := u/2\pi$, $\bar{v} := v/h$

Parametrikus felületek paraméterezése

- Parametrikus felület:

$$F \in \mathbb{R}^2 \to \mathbb{R}^3, \quad F(u,v) := (x,y,z)$$

- u, v paraméterek természetes módon használhatók textúra koordinátáknak.
- Ha $\mathcal{D}_F \neq \mathcal{D}_{tex}$, akkor u, v transzformálni kell.
- Pl:
  - Henger palást
  - $\mathcal{D}_F = [0, 2\pi] \times [0, h]$, $F(u,v) := (\cos u, h, \sin u)$
  - Textúra tér: $(\bar{u}, \bar{v}) \in [0,1] \times [0,1]$
  - Transzformáció: $\bar{u} := u/2\pi$, $\bar{v} := v/h$

Parametrikus felületek paraméterezése

- Parametrikus felület:

$$F \in \mathbb{R}^2 \to \mathbb{R}^3, \quad F(u,v) := (x,y,z)$$

- u, v paraméterek természetes módon használhatók textúra koordinátáknak.
- Ha $\mathcal{D}_F \neq \mathcal{D}_{tex}$, akkor u, v transzformálni kell.
- Pl:
  - Henger palást
  - $\mathcal{D}_F = [0, 2\pi] \times [0, h]$, $F(u,v) := (\cos u, h, \sin u)$
  - Textúra tér: $(\bar{u}, \bar{v}) \in [0,1] \times [0,1]$
  - Transzformáció: $\bar{u} := u/2\pi$, $\bar{v} := v/h$

Parametrikus felületek paraméterezése

- Parametrikus felület:

$$F \in \mathbb{R}^2 \to \mathbb{R}^3, \quad F(u,v) := (x,y,z)$$

- u, v paraméterek természetes módon használhatók textúra koordinátáknak.
- Ha $\mathcal{D}_F \neq \mathcal{D}_{tex}$, akkor u, v transzformálni kell.
- Pl:
  - Henger palást
  - $\mathcal{D}_F = [0, 2\pi] \times [0, h]$, $F(u,v) := (\cos u, h, \sin u)$
  - Textúra tér: $(\bar{u}, \bar{v}) \in [0,1] \times [0,1]$
  - Transzformáció: $\bar{u} := u/2\pi$, $\bar{v} := v/h$

Háromszögek paraméterezése

- Legyen adott a háromszög három csúcsa: $p_i = (x_i, y_i, z_i) \in \mathbb{R}^3$, $i \in \{1, 2, 3\}$, valamint az ezeknek megfelelő csúcsok textúra térben: $t_i = (u_i, v_i) \in \mathbb{R}^2$, $\in \{1, 2, 3\}$.
- Olyan $\mathbb{R}^3 \to \mathbb{R}^2$ leképezést keresünk, amire $p_i \mapsto u_i$, és háromszöget háromszögbe visz át.
- A legegyszerűbb ilyen leképezés a lineáris leképezés, ami megadható egy 3×3-as mátrixszal.

$$(x, y, z) = (u, v, 1) \cdot \begin{bmatrix} A_x & A_y & A_z \\ B_x & B_y & B_z \\ C_x & C_y & C_z \end{bmatrix} = (u, v, 1) \cdot \mathbf{P}$$

Háromszögek paraméterezése

- Legyen adott a háromszög három csúcsa: $p_i = (x_i, y_i, z_i) \in \mathbb{R}^3$, $i \in \{1, 2, 3\}$, valamint az ezeknek megfelelő csúcsok textúra térben: $t_i = (u_i, v_i) \in \mathbb{R}^2$, $\in \{1, 2, 3\}$.
- Olyan $\mathbb{R}^3 \to \mathbb{R}^2$ leképezést keresünk, amire $p_i \mapsto u_i$, és háromszöget háromszögbe visz át.
- A legegyszerűbb ilyen leképezés a lineáris leképezés, ami megadható egy 3×3-as mátrixszal.

$$(x, y, z) = (u, v, 1) \cdot \begin{bmatrix} A_x & A_y & A_z \\ B_x & B_y & B_z \\ C_x & C_y & C_z \end{bmatrix} = (u, v, 1) \cdot \mathbf{P}$$

Háromszögek paraméterezése

- Legyen adott a háromszög három csúcsa: $p_i = (x_i, y_i, z_i) \in \mathbb{R}^3$, $i \in \{1, 2, 3\}$, valamint az ezeknek megfelelő csúcsok textúra térben: $t_i = (u_i, v_i) \in \mathbb{R}^2$, $\in \{1, 2, 3\}$.
- Olyan $\mathbb{R}^3 \to \mathbb{R}^2$ leképezést keresünk, amire $p_i \mapsto u_i$, és háromszöget háromszögbe visz át.
- A legegyszerűbb ilyen leképezés a lineáris leképezés, ami megadható egy 3×3-as mátrixszal.

$$(x, y, z) = (u, v, 1) \cdot \begin{bmatrix} A_x & A_y & A_z \\ B_x & B_y & B_z \\ C_x & C_y & C_z \end{bmatrix} = (u, v, 1) \cdot \mathbf{P}$$

Háromszögek paraméterezése

$$(x, y, z) = (u, v, 1) \cdot \begin{bmatrix} A_x & A_y & A_z \\ B_x & B_y & B_z \\ C_x & C_y & C_z \end{bmatrix} = (u, v, 1) \cdot \mathbf{P}$$

- Kilenc ismeretlen, kilenc egyenlet $\Rightarrow$ számolható.
- Ez textúra alapú leképezés.
- Képtér alapú leképezéshez inverz trafó kell:

$$(u, v, 1) = (x, y, z) \cdot \mathbf{P}^{-1}$$

Háromszögek paraméterezése

$$(x, y, z) = (u, v, 1) \cdot \begin{bmatrix} A_x & A_y & A_z \\ B_x & B_y & B_z \\ C_x & C_y & C_z \end{bmatrix} = (u, v, 1) \cdot \mathbf{P}$$

- Kilenc ismeretlen, kilenc egyenlet $\Rightarrow$ számolható.
- Ez textúra alapú leképezés.
- Képtér alapú leképezéshez inverz trafó kell:

$$(u, v, 1) = (x, y, z) \cdot \mathbf{P}^{-1}$$

Háromszögek paraméterezése

$$(x, y, z) = (u, v, 1) \cdot \begin{bmatrix} A_x & A_y & A_z \\ B_x & B_y & B_z \\ C_x & C_y & C_z \end{bmatrix} = (u, v, 1) \cdot \mathbf{P}$$

- Kilenc ismeretlen, kilenc egyenlet $\Rightarrow$ számolható.
- Ez textúra alapú leképezés.
- Képtér alapú leképezéshez inverz trafó kell:

$$(u, v, 1) = (x, y, z) \cdot \mathbf{P}^{-1}$$

Textúrázás és sugárkövetés

- Felület-sugár metszés: világ-koordináta rendszerben
- Inverz transzformációk:
  - Világ KR → Model KR
  - Model KR → textúra tér

Model KR → textúra tér

- Parametrikus felületek
  A metszéspont számítás során adódik u, v, nem kell külön számítani.
- Háromszögek
  Az előbb levezetett $\mathbf{P}^{-1}$ szükséges.
  Gyorsítási lehetőség: ha sem maga a háromszög, sem a textúra koordináták nem változnak a csúcsokban, akkor $\mathbf{P}^{-1}$ állandó.

Háromszögek paraméterezése

- Gyakorlatban ez gyorsan számítható inkrementális algoritmussal.
- Ha a három csúcsban adottak a textúra koordináták, akkor ezeket a háromszögek kitöltésénél használt algoritmussal kiszámíthatjuk minden pixelre.
- Minden pontra legyen a felület egy pontja $p = \alpha p_1 + \beta p_2 + \gamma p_3$, az α, β, γ baricentrikus koordinátákkal adott.
- Ekkor a p-hez tartozó textúra koordináta is megkaphtó $t = \alpha t_1 + \beta t_2 + \gamma t_3$ alapján

Háromszögek paraméterezése

- Gyakorlatban ez gyorsan számítható inkrementális algoritmussal.
- Ha a három csúcsban adottak a textúra koordináták, akkor ezeket a háromszögek kitöltésénél használt algoritmussal kiszámíthatjuk minden pixelre.
- Minden pontra legyen a felület egy pontja $p = \alpha p_1 + \beta p_2 + \gamma p_3$, az α, β, γ baricentrikus koordinátákkal adott.
- Ekkor a p-hez tartozó textúra koordináta is megkaphtó $t = \alpha t_1 + \beta t_2 + \gamma t_3$ alapján

Háromszögek paraméterezése

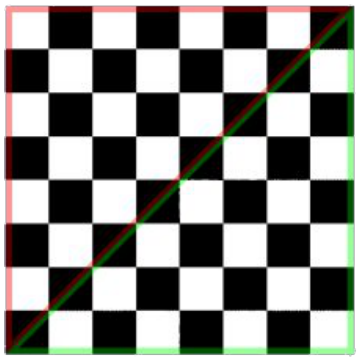
- Gyakorlatban ez gyorsan számítható inkrementális algoritmussal.
- Ha a három csúcsban adottak a textúra koordináták, akkor ezeket a háromszögek kitöltésénél használt algoritmussal kiszámíthatjuk minden pixelre.
- Minden pontra legyen a felület egy pontja $p = \alpha p_1 + \beta p_2 + \gamma p_3$, az α, β, γ baricentrikus koordinátákkal adott.
- Ekkor a p-hez tartozó textúra koordináta is megkaphtó $t = \alpha t_1 + \beta t_2 + \gamma t_3$ alapján

Háromszögek paraméterezése

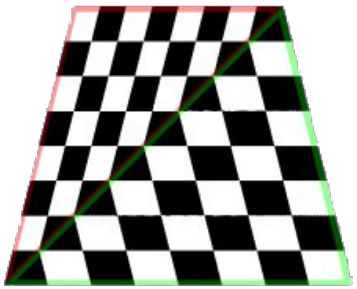
- Gyakorlatban ez gyorsan számítható inkrementális algoritmussal.
- Ha a három csúcsban adottak a textúra koordináták, akkor ezeket a háromszögek kitöltésénél használt algoritmussal kiszámíthatjuk minden pixelre.
- Minden pontra legyen a felület egy pontja $p = \alpha p_1 + \beta p_2 + \gamma p_3$, az α, β, γ baricentrikus koordinátákkal adott.
- Ekkor a p-hez tartozó textúra koordináta is megkaphtó $t = \alpha t_1 + \beta t_2 + \gamma t_3$ alapján

Perspektívikusan korrekt textúrázás

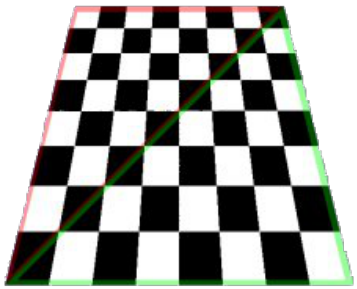
- A textúra koordináták lineáris interpolációja hibás képet ad, ha a megjelenítendő háromszögön nem csak affin transzformációt végzünk.
- Értsd: esetek 99%-a.



Sík



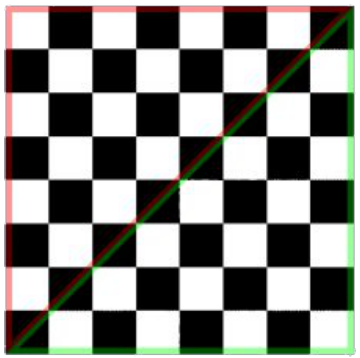
Affin



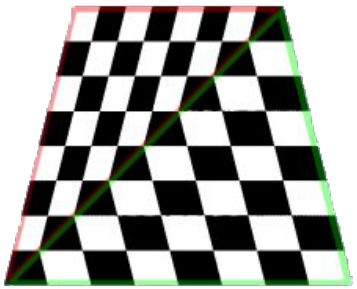
Helyes

Perspektívikusan korrekt textúrázás

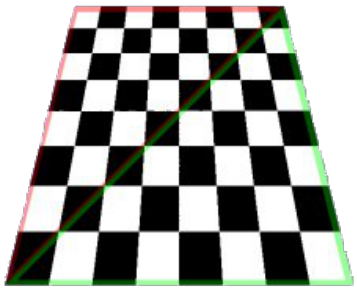
- A textúra koordináták lineáris interpolációja hibás képet ad, ha a megjelenítendő háromszögön nem csak affin transzformációt végzünk.
- Értsd: esetek 99%-a.



Sík



Affin



Helyes

Perspektívikusan korrekt textúrázás

- Eredeti csúcsok: $[x, y, z, 1]$
- Transzformálva, homogén osztás előtt: $[x_t, y_t, z_t, w_t]$
- Homogén osztás után: $[x_s, y_s, z_s, 1] = [\frac{x_t}{w_t}, \frac{y_t}{w_t}, \frac{z_t}{w_t}, 1]$
- Ha $[x_s, y_s, z_s, 1]$ szerint interpoláljuk a textúra koordinátákat, akkor az nem lesz jó.
- Ezzel szemben: ha $\frac{1}{w}$-t interpoláljuk, az helyes marad! Sőt, bármilyen $\frac{q}{w}$ érték jól interpolálódik!
- α, β, γ helyett használjuk a világ koordináta-rendszerbeli $\alpha_w, \beta_w, \gamma_w$ koordinátákat, és
- $\frac{\alpha_w}{w}, \frac{\beta_w}{w}, \frac{\gamma_w}{w}$ interpolálva helyes textúrázást kapunk.

Perspektívikusan korrekt textúrázás

- Eredeti csúcsok: $[x, y, z, 1]$
- Transzformálva, homogén osztás előtt: $[x_t, y_t, z_t, w_t]$
- Homogén osztás után: $[x_s, y_s, z_s, 1] = [\frac{x_t}{w_t}, \frac{y_t}{w_t}, \frac{z_t}{w_t}, 1]$
- Ha $[x_s, y_s, z_s, 1]$ szerint interpoláljuk a textúra koordinátákat, akkor az nem lesz jó.
- Ezzel szemben: ha $\frac{1}{w}$-t interpoláljuk, az helyes marad! Sőt, bármilyen $\frac{q}{w}$ érték jól interpolálódik!
- α, β, γ helyett használjuk a világ koordináta-rendszerbeli $\alpha_w, \beta_w, \gamma_w$ koordinátákat, és
- $\frac{\alpha_w}{w}, \frac{\beta_w}{w}, \frac{\gamma_w}{w}$ interpolálva helyes textúrázást kapunk.

Perspektívikusan korrekt textúrázás

- Eredeti csúcsok: $[x, y, z, 1]$
- Transzformálva, homogén osztás előtt: $[x_t, y_t, z_t, w_t]$
- Homogén osztás után: $[x_s, y_s, z_s, 1] = [\frac{x_t}{w_t}, \frac{y_t}{w_t}, \frac{z_t}{w_t}, 1]$
- Ha $[x_s, y_s, z_s, 1]$ szerint interpoláljuk a textúra koordinátákat, akkor az nem lesz jó.
- Ezzel szemben: ha $\frac{1}{w}$-t interpoláljuk, az helyes marad! Sőt, bármilyen $\frac{q}{w}$ érték jól interpolálódik!
- α, β, γ helyett használjuk a világ koordináta-rendszerbeli $\alpha_w, \beta_w, \gamma_w$ koordinátákat, és
- $\frac{\alpha_w}{w}, \frac{\beta_w}{w}, \frac{\gamma_w}{w}$ interpolálva helyes textúrázást kapunk.

Perspektívikusan korrekt textúrázás

- Eredeti csúcsok: $[x, y, z, 1]$
- Transzformálva, homogén osztás előtt: $[x_t, y_t, z_t, w_t]$
- Homogén osztás után: $[x_s, y_s, z_s, 1] = [\frac{x_t}{w_t}, \frac{y_t}{w_t}, \frac{z_t}{w_t}, 1]$
- Ha $[x_s, y_s, z_s, 1]$ szerint interpoláljuk a textúra koordinátákat, akkor az nem lesz jó.
- Ezzel szemben: ha $\frac{1}{w}$-t interpoláljuk, az helyes marad! Sőt, bármilyen $\frac{q}{w}$ érték jól interpolálódik!
- α, β, γ helyett használjuk a világ koordináta-rendszerbeli $\alpha_w, \beta_w, \gamma_w$ koordinátákat, és
- $\frac{\alpha_w}{w}, \frac{\beta_w}{w}, \frac{\gamma_w}{w}$ interpolálva helyes textúrázást kapunk.

Perspektívikusan korrekt textúrázás

- Eredeti csúcsok: $[x, y, z, 1]$
- Transzformálva, homogén osztás előtt: $[x_t, y_t, z_t, w_t]$
- Homogén osztás után: $[x_s, y_s, z_s, 1] = [\frac{x_t}{w_t}, \frac{y_t}{w_t}, \frac{z_t}{w_t}, 1]$
- Ha $[x_s, y_s, z_s, 1]$ szerint interpoláljuk a textúra koordinátákat, akkor az nem lesz jó.
- Ezzel szemben: ha $\frac{1}{w}$-t interpoláljuk, az helyes marad! Sőt, bármilyen $\frac{q}{w}$ érték jól interpolálódik!
- α, β, γ helyett használjuk a világ koordináta-rendszerbeli $\alpha_w, \beta_w, \gamma_w$ koordinátákat, és
- $\frac{\alpha_w}{w}, \frac{\beta_w}{w}, \frac{\gamma_w}{w}$ interpolálva helyes textúrázást kapunk.

Perspektívikusan korrekt textúrázás

- Eredeti csúcsok: $[x, y, z, 1]$
- Transzformálva, homogén osztás előtt: $[x_t, y_t, z_t, w_t]$
- Homogén osztás után: $[x_s, y_s, z_s, 1] = [\frac{x_t}{w_t}, \frac{y_t}{w_t}, \frac{z_t}{w_t}, 1]$
- Ha $[x_s, y_s, z_s, 1]$ szerint interpoláljuk a textúra koordinátákat, akkor az nem lesz jó.
- Ezzel szemben: ha $\frac{1}{w}$-t interpoláljuk, az helyes marad! Sőt, bármilyen $\frac{q}{w}$ érték jól interpolálódik!
- α, β, γ helyett használjuk a világ koordináta-rendszerbeli $\alpha_w, \beta_w, \gamma_w$ koordinátákat, és
- $\frac{\alpha_w}{w}, \frac{\beta_w}{w}, \frac{\gamma_w}{w}$ interpolálva helyes textúrázást kapunk.

Perspektívikusan korrekt textúrázás

- Eredeti csúcsok: $[x, y, z, 1]$
- Transzformálva, homogén osztás előtt: $[x_t, y_t, z_t, w_t]$
- Homogén osztás után: $[x_s, y_s, z_s, 1] = [\frac{x_t}{w_t}, \frac{y_t}{w_t}, \frac{z_t}{w_t}, 1]$
- Ha $[x_s, y_s, z_s, 1]$ szerint interpoláljuk a textúra koordinátákat, akkor az nem lesz jó.
- Ezzel szemben: ha $\frac{1}{w}$-t interpoláljuk, az helyes marad! Sőt, bármilyen $\frac{q}{w}$ érték jól interpolálódik!
- α, β, γ helyett használjuk a világ koordináta-rendszerbeli $\alpha_w, \beta_w, \gamma_w$ koordinátákat, és
- $\frac{\alpha_w}{w}, \frac{\beta_w}{w}, \frac{\gamma_w}{w}$ interpolálva helyes textúrázást kapunk.

Tartalom

1 Textúra leképezések

2 Textúrák szűrése
- Nagyítás
- Kicsinyítés

3 Procedurális textúrák

4 Nem-szín textúrák

Textúrák szűrése

- Ritkán esik pontosan egy texel egy pixelre.
- Nagyítás: a pixel mérete kisebb a texelénél – több pixel jut egyetlen texelre. OpenGL: `GL_TEXTURE_MAG_FILTER`
- Kicsinyítés: a pixel mérete nagyobb a texelénél – több texel jut egyetlen pixelre. OpenGL: `GL_TEXTURE_MIN_FILTER`
- További probléma: ami a textúra térben lineáris, az nem az a képtérben a perspektíva miatt.

Textúrák szűrése

- Ritkán esik pontosan egy texel egy pixelre.
- Nagyítás: a pixel mérete kisebb a texelénél – több pixel jut egyetlen texelre. OpenGL: `GL_TEXTURE_MAG_FILTER`
- Kicsinyítés: a pixel mérete nagyobb a texelénél – több texel jut egyetlen pixelre. OpenGL: `GL_TEXTURE_MIN_FILTER`
- További probléma: ami a textúra térben lineáris, az nem az a képtérben a perspektíva miatt.

Textúrák szűrése

- Ritkán esik pontosan egy texel egy pixelre.
- Nagyítás: a pixel mérete kisebb a texelénél – több pixel jut egyetlen texelre. OpenGL: `GL_TEXTURE_MAG_FILTER`
- Kicsinyítés: a pixel mérete nagyobb a texelénél – több texel jut egyetlen pixelre. OpenGL: `GL_TEXTURE_MIN_FILTER`
- További probléma: ami a textúra térben lineáris, az nem az a képtérben a perspektíva miatt.

Textúrák szűrése

- Ritkán esik pontosan egy texel egy pixelre.
- Nagyítás: a pixel mérete kisebb a texelénél – több pixel jut egyetlen texelre. OpenGL: `GL_TEXTURE_MAG_FILTER`
- Kicsinyítés: a pixel mérete nagyobb a texelénél – több texel jut egyetlen pixelre. OpenGL: `GL_TEXTURE_MIN_FILTER`
- További probléma: ami a textúra térben lineáris, az nem az a képtérben a perspektíva miatt.

Nagyítás

Egy pixel mérete kisebb egy texelénél – több pixel jut egyetlen texelre

- Szűrés nélkül: a pixel középpontjához legközelebb eső texel értékét használjuk.
- Bilineáris szűrés: a legközelebbi négy texel súlyozott átlagát vesszük.

Kicsinyítés

Egy pixel mérete nagyobb egy texelénél – több texel jut egyetlen pixelre

- Pontos mintavételezés lenne: a pixel négyzetét transzformáljuk el textúra térbe, és az ott kijelölt texelek átlagát vegyük.
- Helyette: a textúra térben is négyzetet veszünk.
- Gyakorlatban: a textúra egy tetszőleges négyzetének a legátlagolás túl erőforrás igényes, helyette vagy használjunk kevesebb texelt, vagy *MIP-map*-eket
- Kevesebb pixel:
  - Szűrés nélkül: a pixel középpontjához legközelebb eső texel értékét használjuk.
  - Bilineáris szűrés: a legközelebbi négy texel súlyozott átlagát vesszük.

Kicsinyítés

Egy pixel mérete nagyobb egy texelénél – több texel jut egyetlen pixelre

- Pontos mintavételezés lenne: a pixel négyzetét transzformáljuk el textúra térbe, és az ott kijelölt texelek átlagát vegyük.
- Helyette: a textúra térben is négyzetet veszünk.
- Gyakorlatban: a textúra egy tetszőleges négyzetének a legátlagolás túl erőforrás igényes, helyette vagy használjunk kevesebb texelt, vagy *MIP-map*-eket
- Kevesebb pixel:
  - Szűrés nélkül: a pixel középpontjához legközelebb eső texel értékét használjuk.
  - Bilineáris szűrés: a legközelebbi négy texel súlyozott átlagát vesszük.

Kicsinyítés

Egy pixel mérete nagyobb egy texelénél – több texel jut egyetlen pixelre

- Pontos mintavételezés lenne: a pixel négyzetét transzformáljuk el textúra térbe, és az ott kijelölt texelek átlagát vegyük.
- Helyette: a textúra térben is négyzetet veszünk.
- Gyakorlatban: a textúra egy tetszőleges négyzetének a legátlagolás túl erőforrás igényes, helyette vagy használjunk kevesebb texelt, vagy *MIP-map*-eket
- Kevesebb pixel:
  - Szűrés nélkül: a pixel középpontjához legközelebb eső texel értékét használjuk.
  - Bilineáris szűrés: a legközelebbi négy texel súlyozott átlagát vesszük.

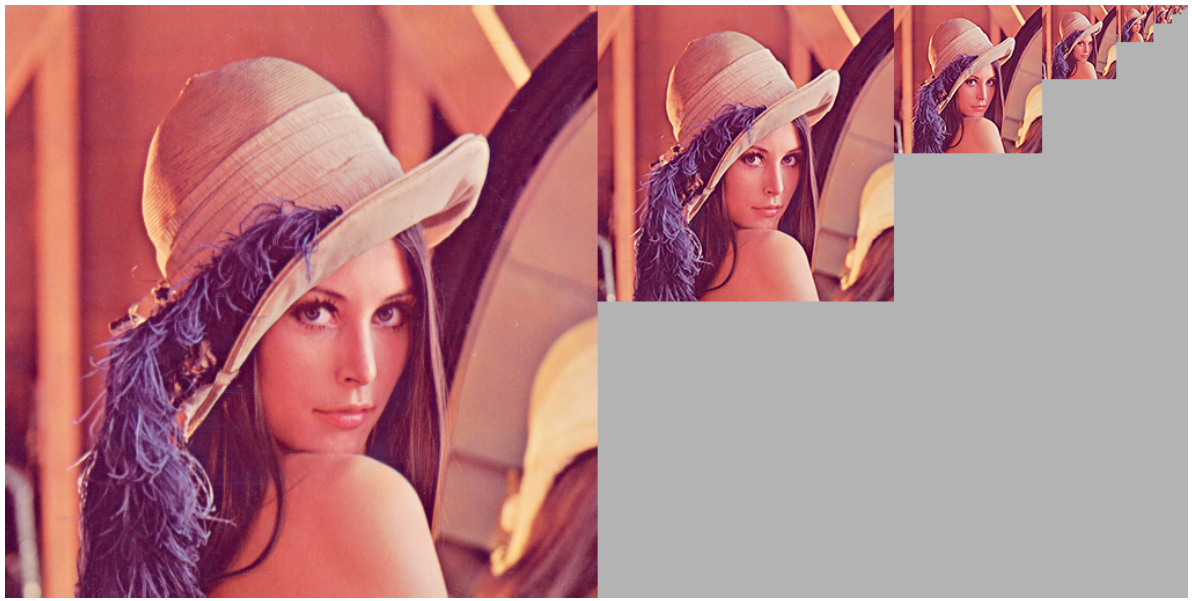
Kicsinyítés

Egy pixel mérete nagyobb egy texelénél – több texel jut egyetlen pixelre

- Pontos mintavételezés lenne: a pixel négyzetét transzformáljuk el textúra térbe, és az ott kijelölt texelek átlagát vegyük.
- Helyette: a textúra térben is négyzetet veszünk.
- Gyakorlatban: a textúra egy tetszőleges négyzetének a legátlagolás túl erőforrás igényes, helyette vagy használjunk kevesebb texelt, vagy *MIP-map*-eket
- Kevesebb pixel:
  - Szűrés nélkül: a pixel középpontjához legközelebb eső texel értékét használjuk.
  - Bilineáris szűrés: a legközelebbi négy texel súlyozott átlagát vesszük.

MIP-map-ek

- MIP: *multum in parvo* – sok, kis helyen
- Ú.n. pirasmist generálunk a textúrábólből, minden szinten felezve a méretét, egészen az 1×1-es méretig.



MIP-map-ek

- Szűrés során kiválasztjuk a pixel/texel terület aránynak megfelelő szintet és onnan olvasunk.
- Az adott MIP-map-en belül is lehet az olvasás szűrés nélküli, vagy lehet bilienárisan szűrt.
- *Trilineáris szűrés*: két szomszédos szintet használunk, azokon belül bilineáris szűréssel, és ezeknek vesszük a súlyoztott átlagát.

Tartalom

1 Textúra leképezések

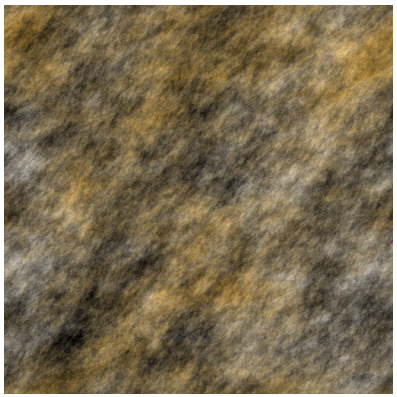
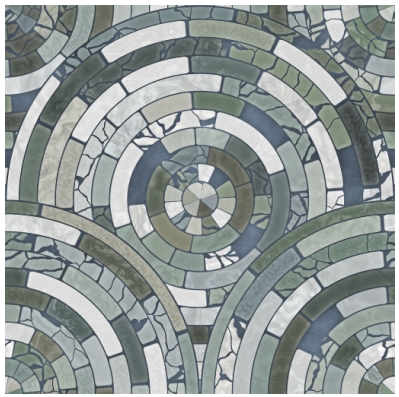
2 Textúrák szűrése

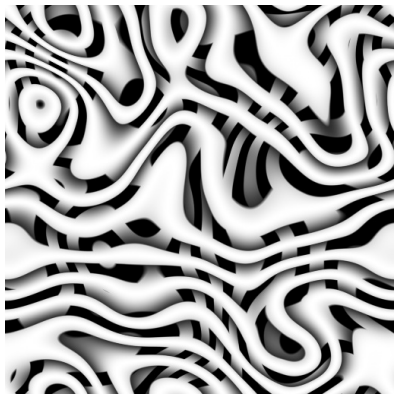
3 Procedurális textúrák

4 Nem-szín textúrák

Procedurális textúrák

- A textúrákat ”tömb” helyett megadhatjuk függvénnyel is.
- Textúra koordináták → a függvény paraméterei.



generated with Filter Forge

Tulajdonságai

- Előnyök:
  - Sokkal kevesebb tárhely
  - Tetszőleges felbontás – csak a numerikus pontosság a korlát

  $\Rightarrow$ Nagyításnál nem kell szűrés. (Kicsinyítést sajnos nem oldja meg.)
- Hátrányok:
  - Nagy számítás igény.
  - Nem lehet vele bármit leírni.
  - Nem módosítható tetszőlegesen.

Tulajdonságai

- Előnyök:
  - Sokkal kevesebb tárhely
  - Tetszőleges felbontás – csak a numerikus pontosság a korlát

  $\Rightarrow$ Nagyításnál nem kell szűrés. (Kicsinyítést sajnos nem oldja meg.)
- Hátrányok:
  - Nagy számítás igény.
  - Nem lehet vele bármit leírni.
  - Nem módosítható tetszőlegesen.

Tulajdonságai

- Előnyök:
  - Sokkal kevesebb tárhely
  - Tetszőleges felbontás – csak a numerikus pontosság a korlát

  $\Rightarrow$ Nagyításnál nem kell szűrés. (Kicsinyítést sajnos nem oldja meg.)
- Hátrányok:
  - Nagy számítás igény.
  - Nem lehet vele bármit leírni.
  - Nem módosítható tetszőlegesen.

Tulajdonságai

- Előnyök:
  - Sokkal kevesebb tárhely
  - Tetszőleges felbontás – csak a numerikus pontosság a korlát

  $\Rightarrow$ Nagyításnál nem kell szűrés. (Kicsinyítést sajnos nem oldja meg.)
- Hátrányok:
  - Nagy számítás igény.
  - Nem lehet vele bármit leírni.
  - Nem módosítható tetszőlegesen.

Tulajdonságai

- Előnyök:
  - Sokkal kevesebb tárhely
  - Tetszőleges felbontás – csak a numerikus pontosság a korlát

  $\Rightarrow$ Nagyításnál nem kell szűrés. (Kicsinyítést sajnos nem oldja meg.)
- Hátrányok:
  - Nagy számítás igény.
  - Nem lehet vele bármit leírni.
  - Nem módosítható tetszőlegesen.

Tulajdonságai

- Előnyök:
  - Sokkal kevesebb tárhely
  - Tetszőleges felbontás – csak a numerikus pontosság a korlát

  $\Rightarrow$ Nagyításnál nem kell szűrés. (Kicsinyítést sajnos nem oldja meg.)
- Hátrányok:
  - Nagy számítás igény.
  - Nem lehet vele bármit leírni.
  - Nem módosítható tetszőlegesen.

Tulajdonságai

- Előnyök:
  - Sokkal kevesebb tárhely
  - Tetszőleges felbontás – csak a numerikus pontosság a korlát

  $\Rightarrow$ Nagyításnál nem kell szűrés. (Kicsinyítést sajnos nem oldja meg.)
- Hátrányok:
  - Nagy számítás igény.
  - Nem lehet vele bármit leírni.
  - Nem módosítható tetszőlegesen.

Tartalom

1. Textúra leképezések
2. Textúrák szűrése
3. Procedurális textúrák
4. Nem-szín textúrák
   - Bucka vagy normál leképezés
   - Displacement leképezés
   - Árnyék térképek
   - Tükrözédések

Nem-szín textúrák

- A textúrák segítségével a felületi pont bármilyen tulasdonságát leírhatjuk.
- Ezek a tulajdonságok lehetnek pl.:
  - felületi normális – *Bucka ill. normál leképezés*
  - elmozdulás – Displacement *leképezés*
  - fényforrás láthatósága – *Árnyék térképek*
  - tükörként visszavert fény – *Visszaverődés leképezés* ill. *Környezet leképezés*

Nem-szín textúrák

- A textúrák segítségével a felületi pont bármilyen tulasdonságát leírhatjuk.
- Ezek a tulajdonságok lehetnek pl.:
  - felületi normális – *Bucka ill. normál leképezés*
  - elmozdulás – Displacement *leképezés*
  - fényforrás láthatósága – *Árnyék térképek*
  - tükörként visszavert fény – *Visszaverődés leképezés* ill. *Környezet leképezés*

Nem-szín textúrák

- A textúrák segítségével a felületi pont bármilyen tulasdonságát leírhatjuk.
- Ezek a tulajdonságok lehetnek pl.:
  - felületi normális – *Bucka ill. normál leképezés*
  - elmozdulás – Displacement *leképezés*
  - fényforrás láthatósága – *Árnyék térképek*
  - tükörként visszavert fény – *Visszaverődés leképezés* ill. *Környezet leképezés*

Nem-szín textúrák

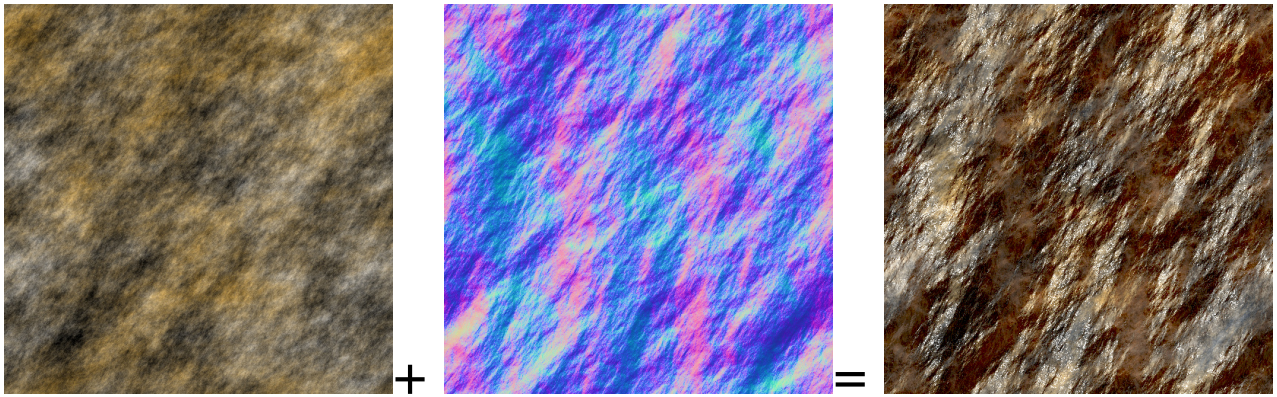
- A textúrák segítségével a felületi pont bármilyen tulasdonságát leírhatjuk.
- Ezek a tulajdonságok lehetnek pl.:
  - felületi normális – *Bucka ill. normál leképezés*
  - elmozdulás – Displacement *leképezés*
  - fényforrás láthatósága – *Árnyék térképek*
  - tükörként visszavert fény – *Visszaverődés leképezés* ill. *Környezet leképezés*

Nem-szín textúrák

- A textúrák segítségével a felületi pont bármilyen tulasdonságát leírhatjuk.
- Ezek a tulajdonságok lehetnek pl.:
  - felületi normális – *Bucka ill. normál leképezés*
  - elmozdulás – Displacement *leképezés*
  - fényforrás láthatósága – *Árnyék térképek*
  - tükörként visszavert fény – *Visszaverődés leképezés* ill. *Környezet leképezés*

Bucka vagy normál leképezés

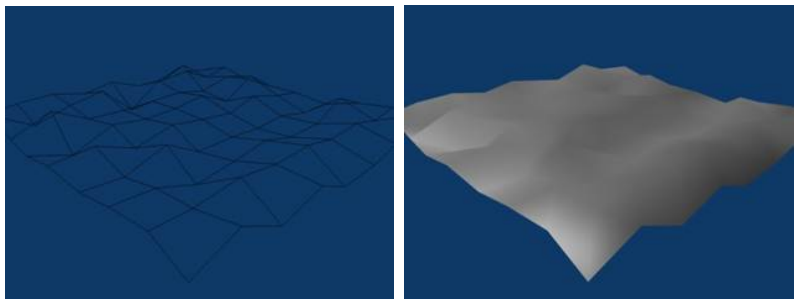
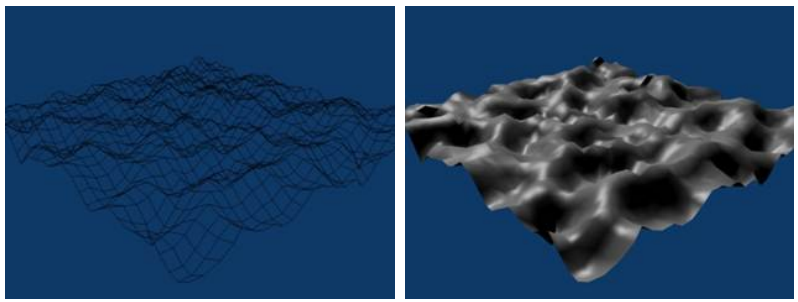
- A textúrával normál vektorokat adunk meg.
- A felület eredeti normálisai helyett ezeket használjuk a megvilágítás számításakor.
- Rücskös/érdes felület látszatát adja, amíg nem nézünk rá túl lapos szögben.



generated with Filter Forge

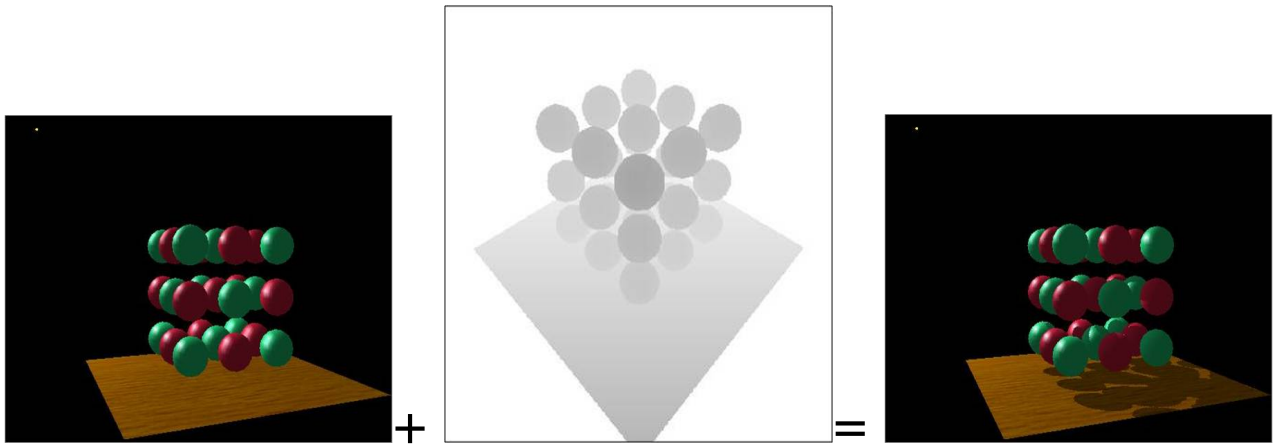
Displacement leképezés

- A feületi pontot tényleges arrébb mozdítjuk.
- A csak háromszögek csúcsait tudjuk elmozdítani ⇒ függ a geometria felbontásától.
- Lapos szögben is helyes látványt kapunk.

Árnyék térképek

- A fényforrás szemszögéből készítünk egy textúrát, amiben a fényforrástól vett távolságokat tároljuk el.
- A tényleges rajzolás során ez alapján döntjük el minden pontra, hogy azt éri-e közvetlenül a fény vagy sem.



Visszaverődés leképezés

- Sík tükrök esetén használható.
- Külön képet készítünk, textúrába mentve, arról, hogy mi látszik tükörirányban.
- Ezt textúraként ráfeszítjük a felületre a végső kirajzoláskor.
- Csak egyszeres tükröződést ad.
- Tükrönként külön el kell végezni.

Környezet leképezés

- A színterünk környezetét végtelenül távolinak tekintjük.
- Speciális textúrában tároljuk.
- Rajzolás a tükröződő felületekhez ebből olvasunk a tükörirány szerint.



