Tekstuurimine

Mark Fishel (fishel@ut.ee)

17. oktoober 2005

Eel ine ord

- Valgus, värv
- Valgusallikad
- Punkti valgustus (illumination)
- Polügonide värvimine (shading)

Valgusalli a

TODO 4 pilti

Pun ti valgustuse o ponen i

- Keskkond (ambient)
- Diffusne värv
- Läikimine (specular highlighting)
- Säramine

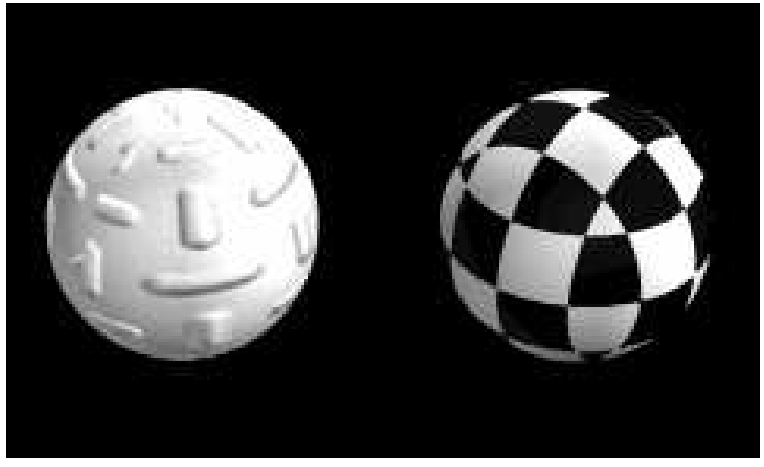
DEMO...

olügoni e värvi ine

- Konstantne
- Gouraud (interpoleeritud tipude värvus)
- Phong (interpoleeritud tipude normaalid)

See or

- Tekstuurid
- Filtreerimine
- Bump mapping
- Environment mapping



e stuuri ine

ilt obje ti pinnal

$$I_{diffuse} = k_d L \cdot \mathbf{n}^T \mathbf{l}$$

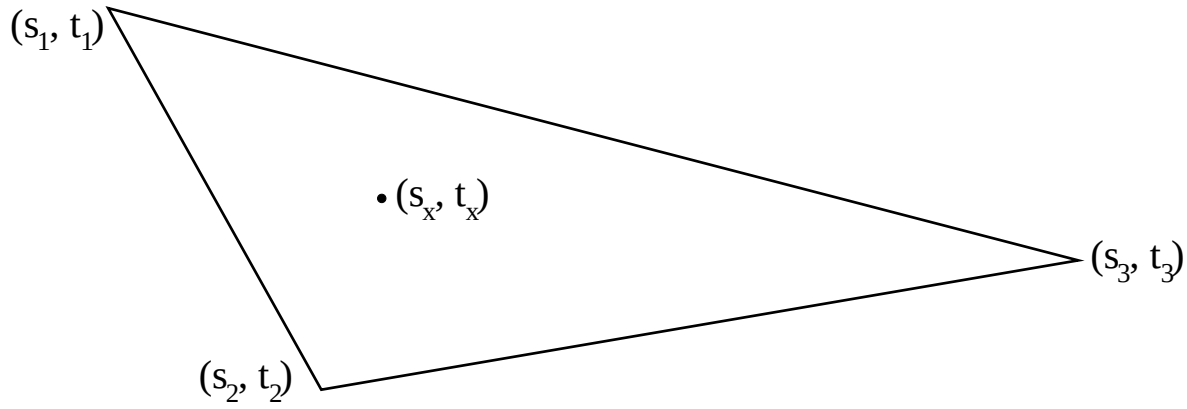
- Diffusse värvi (k_d) asemel korrutame pildi vastava piksliga
- Korrates seda iga piksli jaoks, saame pildi objekti pinna peale
- Põhiprobleem – kuidas leida *vastava piksli*

e stuur

- *Tekstuur* on informatsioon pinna “välimuse” kohta
- Võimalikud informatsiooni variandid:
  - Asendusvärv
  - Lisa-atribuutid
    - Värv, läbipaistvus jms
    - Eelarvutatud valgustusmudelite suurused
    - Normaali nihked, jms
- Harilikult kahemõõtmeline massiiv
  - Koordinaadid $s \in [0, 1]$ ja $t \in [0, 1]$

e stuuri oor inaa i

- Iga polügoni tipuga seostatakse tema tekstuuri koordinaadid
- Interpoleerides tipude koordinaate, saame muude punktide tekstuuri koordinaadid



$$\langle s_x, t_x \rangle = \alpha_1 \langle s_1, t_1 \rangle + \alpha_2 \langle s_2, t_2 \rangle + \alpha_3 \langle s_3, t_3 \rangle$$

Interpoleeri ine

- Perspektiivse projektsiooniga ei saa interpoleerida lineaarselt
- Kasutame hüperboolset interpoleerimist

$$\frac{a_t}{z_t} = t\frac{a_1}{z_1} + (1-t)\frac{a_2}{z_2}$$

e stuuri oor inaati e äära ine

Olgu pind määratud funktsiooniga $\mathbf{p}(u, v)$ kus $\langle u, v \rangle$ kuulub mingi $\mathbb{R}^2$ alampiirkonda. Vaatleme punktide tekstuuri koordinaate kui funktsioone paarist $\langle u, v \rangle$:

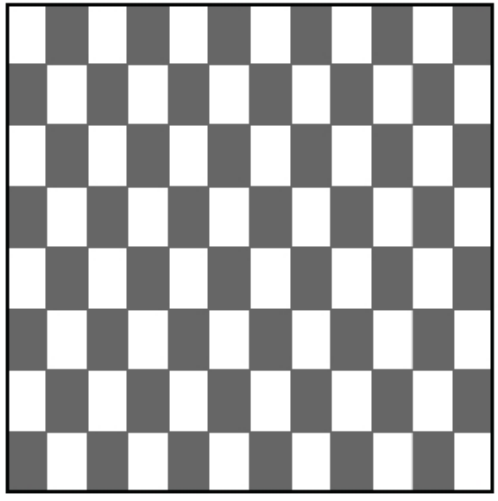
$$s(u, v), \quad t(u, v)$$

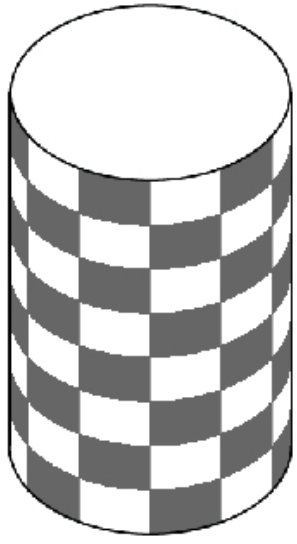
e stuuri oor inaati e äära ine

Silinder (ilma alusteta):

$$\mathbf{p}(\theta, y) = \langle r \sin\theta,\ y,\ r\cos\theta \rangle,\ y \in [0, h],\ \theta \in [0, 2\pi]$$

Selle tekstuuri koordinaadid: $s = \frac{\theta}{2\pi}$ ja $t = \frac{y}{h}$:

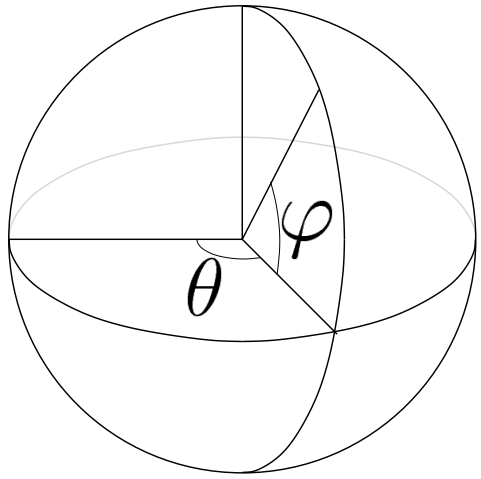




e stuuri oor inaati e äära ine

Sfäär:

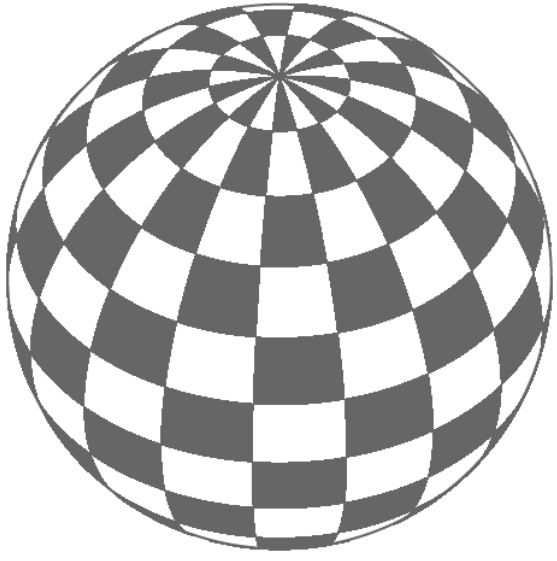
$$\mathbf{p}(\theta, \varphi) = \langle r \sin\theta \cos\varphi,\ r sin\varphi,\ r \cos\theta \cos\varphi \rangle,$$
$$\theta \in [0, 2\pi],\ \varphi \in [-\tfrac{\pi}{2}, \tfrac{\pi}{2}]$$



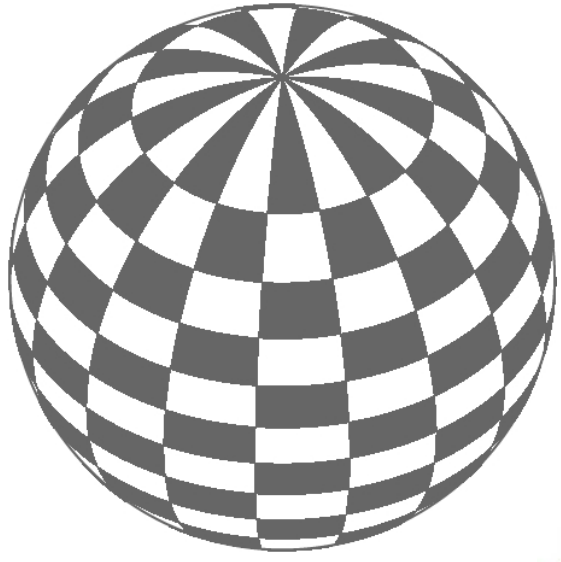
e stuuri oor inaati e äära ine

Sfääri tekstuuri koordinaadid: $s = \frac{\theta}{2\pi}$ ja

$t = \frac{\varphi}{\pi} + \frac{1}{2}$:



$t = \frac{\sin\varphi}{2} + \frac{1}{2}$:

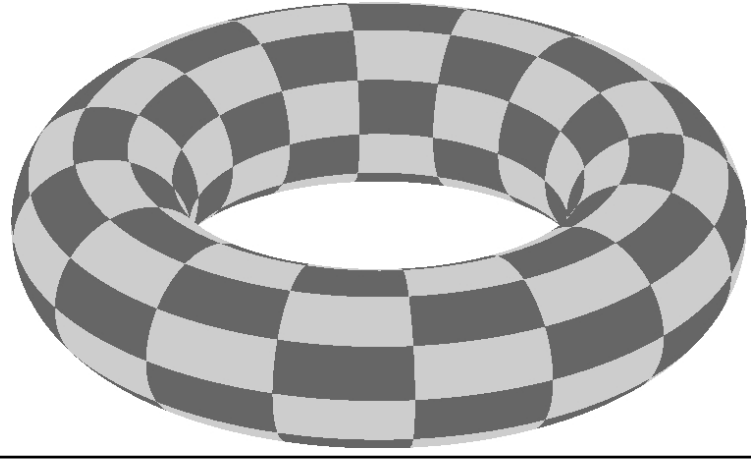


e stuuri oor inaati e äära ine

Toor:

$$\mathbf{p}(\theta,\varphi) = \langle (R + r\cos\varphi)\sin\theta,\ r sin\varphi,\ (R + r\cos\varphi)\cos\theta \rangle,$$
$$\theta \in [0, 2\pi],\ \varphi \in [0, 2\pi]$$

Tekstuuri koordinaadid: $s = \frac{\theta}{2\pi}$ ja $t = \frac{\varphi}{2\pi}$:



e stuuri oor inaati e äära ine

. . .

Laien atu te stuuri oor inaa i

Kuidas lahendada olukorda kui interpoleerimisel tekstuuri koordinaat läheb lubatud piirkonnast $[0, 1]$ välja?

- Väljaspool tekstuuri kõik on mingit ühte värvi
- Väljaspool tekstuuri kõik on sama värvi kui tekstuuri äärmine piksel
- Tekstuuri korratakse:

$$s^* = s - \lfloor s \rfloor$$
$$t^* = t - \lfloor t \rfloor$$

Filtreeri ine

iltreeri ine

- *Tekstuuri filtreerimine – tekslite* (tekstuuri pikslite) seostamine objekti punktidega (reeglina see pole üksühene vastavus).
- Tekstuur on tavaliselt diskreetne, objekti punktile vastavad tekstuuri koordinaadid ei pruugi sattuda täpselt teksli peale.
- *Aliasing* (sakilisus), laias mõtes – kõik probleemid mida põhjustab teisendamine analoog- ja digitaalsignaali vahel, või erinevalt diskreeditud digitaalsete formaatide vahel.

ähi a naa ri eeto

- Punktile vastab lähim teksel.
  - kui tekstuuri resolutsioon on ekraani omast väiksem, probleemiks on aliasing (sakilisus) kitsas mõtes
  - kui on suurem, probleeme tekib see et igale punktile vastab ainult üks teksel:
    - mõned tekslid ei kajastu, algpilt on moondunud
    - liikuva tekstuuri puhul erinevad tekslid vastavad punktile erinevate kaadrite renderdamisel, pilt vilkub

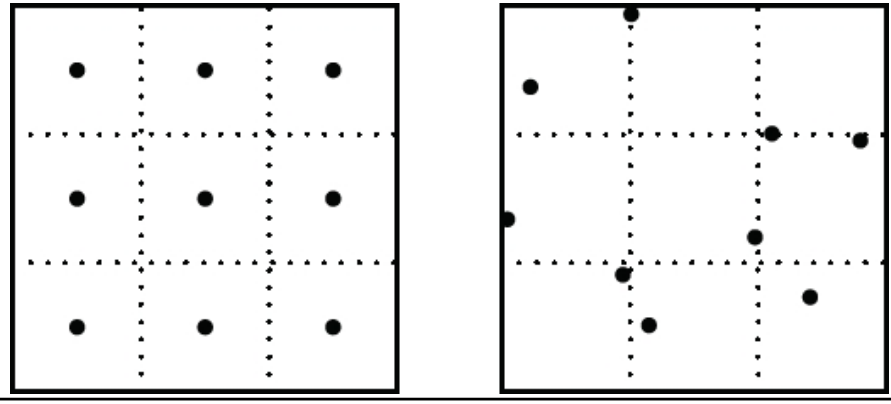
Supersa pling

- Jagame pikslit $N \times N$ alampiksliteks
- Leiame teksli iga alampiksli jaoks (lähima naabri meetodil)
- Piksli punktile vastab alampikslite tekslite aritmeetiline keskmine

Lükab edasi selle hetke kui lähima naabri meetodi probleeme on näha (kui tekstuuri resolutsioon on rohkem kui N korda suurem ekraani omast).

Stoc astic supersa pling

- Selle asemel et alampiksleid hoida ruudulises võrgustikus, paigutame alampikslite punkte suvaliselt ümber piksli ala.
- Kuna täiesti suvaliselt paigutatud punktid ei pruugi alati olema võrdselt jagatud piksli alal, kasutame *jitter* meetodit:



Bilineaarfiltreeri ine

- Puntile vastab lähima nelja teksli bilineaarne kumer kombinatsioon
- Kui tekstuuri resolutsioon on rohkem kui kahekordselt suurem ekraani resolutsioonist
  - Võib võtta rohkem kui 2×2 tekslit
  - Suhteliselt kallis sooritada
  - Eelistatakse meetodit *mipmapping*

Mip apping

- Genereeritakse tekstuuri vähendatud versioone: 2 korda väiksem, 4 korda, jne
- Sõltuvalt vajadusest kajastatakse vastava detailsusega versiooni
- Mälu kasutus suureneb maksimaalselt 33% võrra

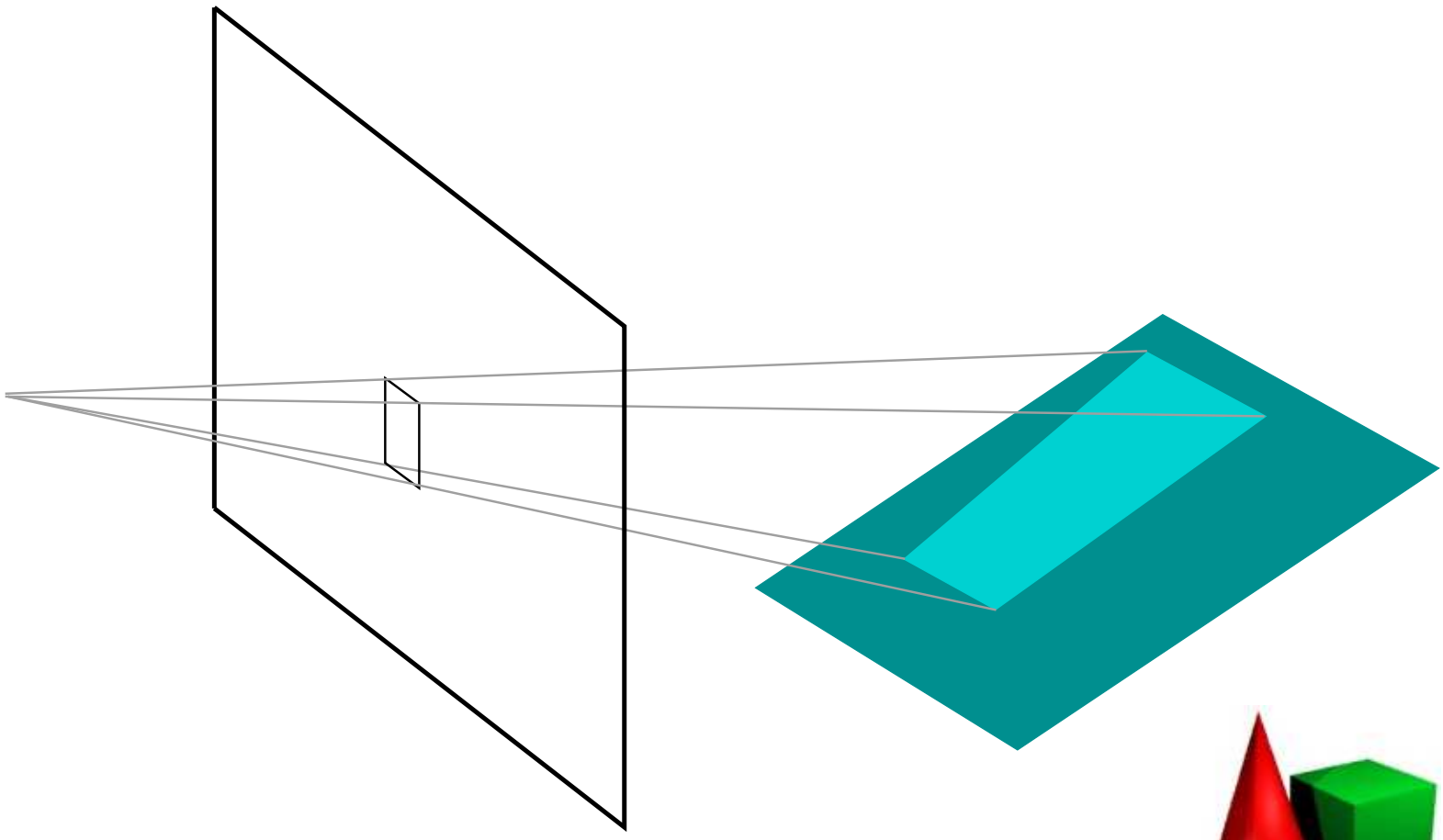
rilineaar ltreeri ine

- Kui pinna üks osa asub lähedal ja teine kaugel, on vaja kahte mipmapi kokku segada.
- Selleks kasutame kahe vastava teksli kumera kombinatsiooni
- Kaalud sõltuvad ekraani ja tekstuuride resolutsioonide suhest

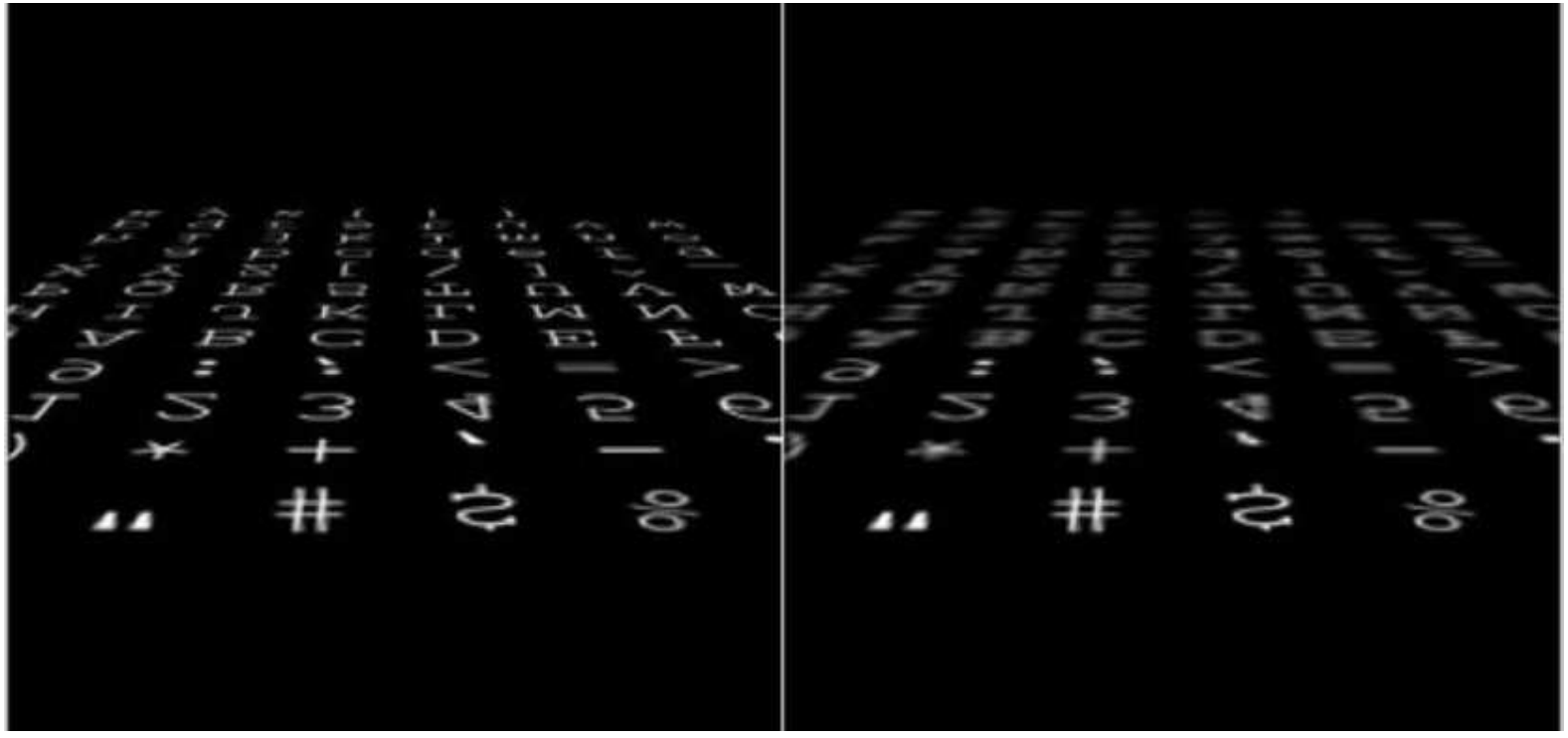
Anisotroopne ltreeri ine

- Siiani vaadeldud meetodid on *isotroopsed*
- Peamine probleem selle lähenemisega tekib kui pinna normaali ja vaatamissuuna vaheline nurk on suur:
  - Tekstuuri koordinaatide s ja t skaleerimine on erinev
  - Tulemuseks pilt on hägus
- Anisotroopsel filtreerimisel piksliga seostatakse sellega kaetud tekstuuri ala (mis pole isotroopne)

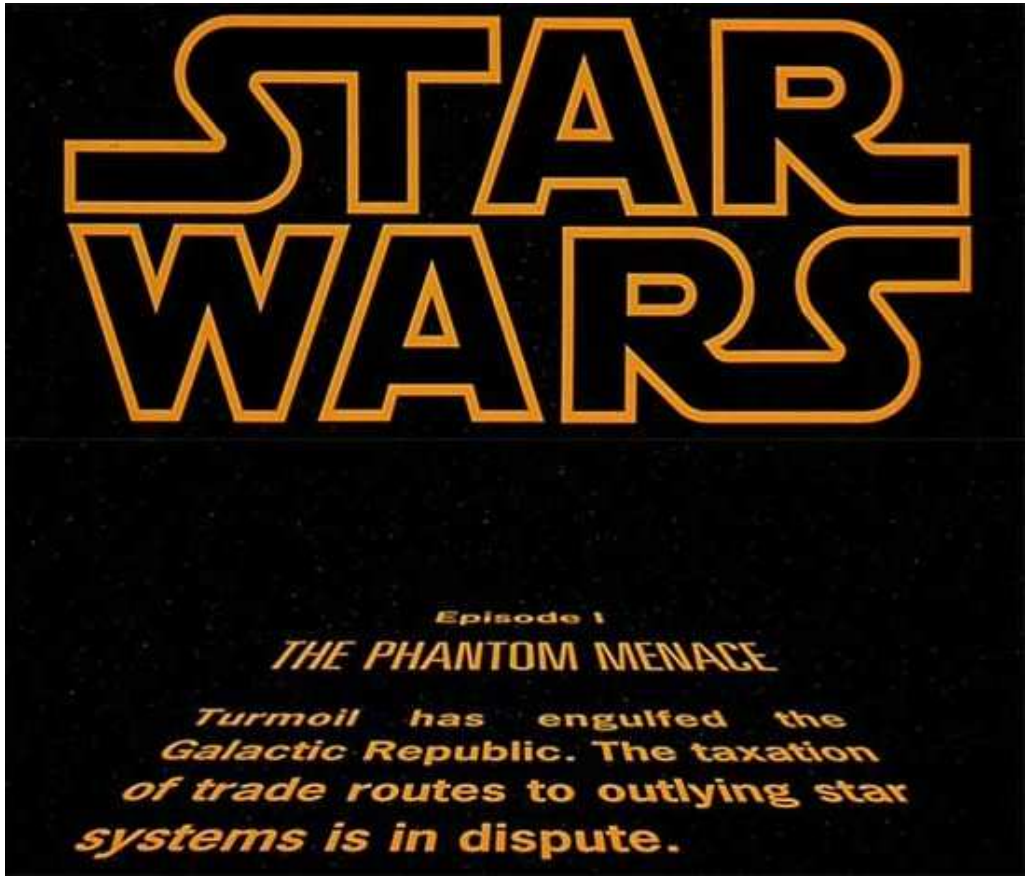
nisotroopne ltreeri ine



nisotroopne ltreeri ine



. . .


STAR WARS
Episode I
THE PHANTOM MENACE
Turmoil has engulfed the Galactic Republic. The taxation of trade routes to outlying star systems is in dispute.

u p apping

u p apping

Põhimõte:

- Tekstuuris salvestatav info – skalaar $d(s,t)$, mis tähendab väikest pinna tõusu
- Bump map mõjutab ainult valgustusmudelit (objekti kuju säilib)

Võrdlemiseks – *displacement mapping*

Normaali arvutamine

Olgu pind kirjeldatud funktsiooniga $\mathbf{p}(u, v)$. Uue punkti valem:

$$\mathbf{p}^*(u, v) = \mathbf{p} + d\,\mathbf{n}$$

or aali arvuta ine

Olgu pind kirjeldatud funktsiooniga $\mathbf{p}(u, v)$. Uue punkti valem:

$$\mathbf{p}^*(u, v) = \mathbf{p} + d\,\mathbf{n}$$

Uus pinna normaal:

$$\mathbf{m} \approx \left(\frac{\partial \mathbf{p}}{\partial u} \times \frac{\partial \mathbf{p}}{\partial v}\right) + \left(\frac{\partial d}{\partial u}\mathbf{n} \times \frac{\partial \mathbf{p}}{\partial v}\right) - \left(\frac{\partial d}{\partial u}\mathbf{n} \times \frac{\partial \mathbf{p}}{\partial v}\right)$$

$$\mathbf{n}^* = \frac{1}{\|\mathbf{m}\|}\mathbf{m}$$

nviron ent apping

nviron ent apping

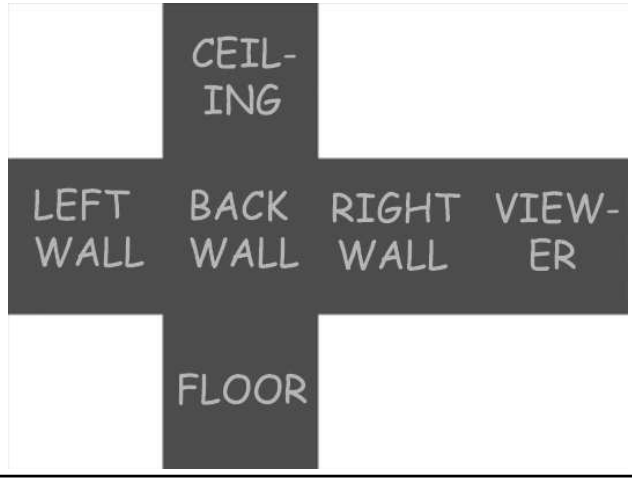
Kuidas modelleerida väikest objekti mis peegeldab ümbritsevat keskkonda?

Sfääri eeto

- Renderdame ideaalselt peegeldavat sfääri ortogonaalses projektsioonis (eraldi puhvris)
- Objekti piksli värvimisel teame vaataja suuna ja objekti pinna nurga, vastavalt sellele võtame sfääri pealt piksli

Kar i eeto

- Paigaldades kaamerat objekti asukohta renderdame ümbritsevat maailma kuubikujulise karbi sisemiste tahkude peale
- Objekti piksli värvimisel leiame peegeldusvektori ja karbi tahku lõikepunkti, sealt saame piksli värvi



ar i eeto

- Eelis sfääri meetodi suhtes – ei sõltu vaataja asukohast

e stuuri ine OpenG 'is

e stuuri ra en a ine

```
glPixelStorei(GL_UNPACK_ALIGNMENT, 1);

glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA,
    width, height, 0, GL_RGBA, GL_FLOAT,
    pixelArray);
```

ip apping

```
gluBuild2DMipmaps(GL_TEXTURE_2D, GL_RGBA,
    width, height, GL_RGBA, GL_FLOAT,
    pixelArray);
```

oor inaati e ontroll

```
glTexParameteri(GL_TEXTURE_2D,
    GL_TEXTURE_WRAP_S | GL_TEXTURE_WRAP_T,
    GL_CLAMP | GL_CLAMP_TO_EDGE | GL_REPEAT);
```

iltreeri ine

```
glTexParameteri(GL_TEXTURE_2D,

    GL_TEXTURE_MAG_FILTER |
    GL_TEXTURE_MIN_FILTER,

    GL_NEAREST | GL_LINEAR);
```

ip apping

```
glTexParameteri(GL_TEXTURE_2D,
    GL_TEXTURE_MIN_FILTER,

    GL_NEAREST_MIPMAP_NEAREST |
    GL_LINEAR_MIPMAP_NEAREST |
    GL_NEAREST_MIPMAP_LINEAR |
    GL_LINEAR__MIPMAP_LINEAR);
```

nisotroopne ltreeri ine

- Genereeritakse mipmappe erinevate kõrguse ja laiuse suhetega
- Kaadri renderdamisel saab vaataja suuna ja pinna normaali vahelisest nurgast arvutada milline mipmap sobib paremini antud suhe jaoks
- Piksli poolt kaetud tekstuuri ala on ristkülik

Ra en a ise viisi

```
glTexEnvi(GL_TEXTURE_ENV,
    GL_TEXTURE_ENV_MODE,
    GL_DECAL | GL_MODULATE);
```

äi i ise säili ine

```
glLightModeli(GL_LIGHT_MODEL_COLOR_CONTROL,
    GL_SINGLE_COLOR |
    GL_SEPARATE_SPECULAR_COLOR);
```

nviron ent apping

```
glTexGeni(GL_S | GL_T,
    GL_TEXTURE_GEN_MODE,
    GL_SPHERE_MAP);

glEnable(GL_TEXTURE_GEN_S |
    GL_TEXTURE_GEN_T);
```

si use ?