

Texturmappning

I TSBK05: 2D-texturer för materialytor, koordinater gavs manuellt eller med `glTexGen`.

Låt oss gå lite längre:

- Fler dimensioner
- Optimering
- Multipass
- Texturplacering
- Multitexturering

Ingemar
Ragnemalm
ingis@isy.liu.se

Texturer i 1, 2, 3 dimensioner

1D-texturer: rad av texlar

2D-texturer: som vanligt

3D-texturer: volymdata

Ingemar
Ragnemalm
ingis@isy.liu.se

Texturer i 1 och 3 dimensioner

1D-texturer:

- Höjdfärgkodning
- Ljusdämpning

3D-texturer:

- 3D-visualisering

Ingemar
Ragnemalm
ingis@isy.liu.se

Vi kan väl redan:

Generera texturkoordinater:

- linjär mappning
- cylindrisk mappning
- sfärisk mappning

Texturinställningar: repeat/clamp

Texturfilter, near/far, närmaste granne, linjär interpolation...

Mipmappning

Texturcachning

VRAM är begränsat. Texturer tar plats (även komprimerade)!

Scenberoende lista?

Återanvänd texturer genom att skriva över samma minnesutrymme med `glTexSubImage`.

Blir VRAM fullt så måste vissa texturer ”swappas” ut. Textas med `glAreTexturesResident`.

Ingemar
Ragnemalm
ingis@isy.liu.se

Texturcachning

System för cachning av texturer:

Märk varje textur med en tidsstämpel, senaste tiden den användes. När mer plats krävs, tag bort den textur med äldst stämpel.

LRU-policy, “least recently used”.

FIFO alternativ, men oftast ineffektivt.

**OpenGL har visst stöd inbyggt!
(`glPrioritizeTextures`)**

Ingemar
Ragnemalm
ingis@isy.liu.se

Multipass-texturering

Man kan rita samma yta flera gånger - och det kan vara meningsfullt!

Görs enklast - men inte snabbast - genom att helt enkelt rita geometrin flera gånger, med olika texturer.

Kräver att man ändrar Z-buffern till att rita på lika, LEQUAL i stället för LESS:

```
drawMyShape();
glDepthFunc(GL_LEQUAL); // Rita även på lika
// (ändra texturinformation)
drawMyShape();
glDepthFunc(GL_LESS); // Ställ tillbaka
```

Ingemar
Ragnemalm
ingis@isy.liu.se

Texturplacering

När vi använder glTexGen, hur får vi rätt täthet och rätt placering?

Placeringen justeras med spec av objektplan (en per texturkoordinat):

glTexGenfv(GL_S, GL_OBJECT_PLANE, plane);

GLfloat plane[] = {1, 0, 0, 0.5}

Ger möjlighet att sätta rotation, translation skalning, till och med skevning!

Ingemar
Ragnemalm
ingis@isy.liu.se

Texturmatrisen

Men objektplanen är ändå inte helt generella!

För ännu större flexibilitet: texturmatris!

```
glMatrixMode(GL_TEXTURE);
glRotatef();
glTranslatef();

-
glMatrixMode(GL_MODELVIEW); // Ställ tillbaka
```

Tillåter alla transformationer som en 4x4-matris kan representera - till och med projektion!

Kan använda de vanliga transformanropen - smidigt!