

8. Texturarea suprafețelor

8.1. Suport teoretic

Texturarea suprafețelor reprezintă un subiect complex în OpenGL și oferă o serie de posibilități de utilizare. De exemplu, poți textura suprafețe formate din mai multe poligoane sau suprafețe curbate, texturând un singur poligon și apoi repetând acest lucru într-una sau mai multe direcții.

Există trei tipuri de texturare:

- texturare 1D
- texturare 2D
- texturare 3D.

Texturile sunt tablouri de date- de exemplu, culoare (RGB), intensitate (în cazul imaginilor monocrome), culoare și transparență (RGBA). Aceste date sunt elemente de textură denumite *texeli*.

Un *texel* poate fi format din unul, două, trei, patru elemente în funcție de textură (exemplu: pentru textura de tip RGB *texel*-ul are 3 elemente, pentru textura de tip RGBA sunt necesare 4 elemente, ultimul reprezentând transparența - alfa).

Un element de textură îl putem repera pe baza coordonatelor de texturare (*s*, *t*, *r*, *q*).

Pentru texturarea unei suprafețe se parcurg următorii pași:

1. Se încarcă un tablou cu imaginea texturii, unidimensional sau bidimensional; acesta este fie generat intern, fie încărcat dintr-un fișier BMP. Componentele se memorează după regulile de la bitmap-uri.
2. Pe baza tabloului se construiește textura în memoria OpenGL. În memoria OpenGL pot fi salvate mai multe texturei.

```
glTexImage2D();  
glTexImage3D(), având ca parametrii dimensiunea texturii, tabloul cu  
texelii, nivelul de texturare, tipul texturii.
```

3. Caracterizarea texturii:

- tipul de wrap-are (înfășurare);
- moduri de filtrare ale texturii;
- mipmapping;
- modurile de texturare.

4. Activarea texturării:

```
glEnable(GL_TEXTURE_1D);  
glEnable(GL_TEXTURE_2D);  
glEnable(GL_TEXTURE_3D);
```

Dacă avem o singură textură, aceasta este realizată în *myinit()*. Dacă sunt mai multe texturei, acestea sunt utilizate înaintea fiecărui obiect desenat (astfel se leagă textura cu obiectul).

5. Reprezentarea obiectelor:

- dacă obiectele sunt modelate intern, atunci trebuie date coordonate de texturare asociate cu vertex-urile respective;
- dacă obiectele sunt luate dintr-o bibliotecă (GLAUX, GLUT, GLU), atunci coordonatele de texturare sunt generate automat.

Este obligatoriu ca dimensiunile texturilor să fie puteri ale lui 2.:

$$width = 2^n + 2 \cdot b$$

$$height = 2^n$$

8.2. Wrapping

Wrapping-ul (înfășurare) se referă la felul cum se plasează textura pe poligon, adică este întinsă pe tot poligonul, sau o aplicăm pe fiecare direcție de 2, 3...n ori.

GL_CLAMP - texelii din marginea texturii sunt estinși;

GL_REPEAT – se repetă texturarea.

Nu este suficient a se furniza modul de wrapping ci mai este necesară și furnizarea în mod corespunzător a coordonatelor de texturare.

8.3. Modele de filtrare

În OpenGL există două filtre: filtrul de maximizare și filtrul de minimizare.

Filtrele pot fi:

GL_NEAREST- se alege texel-ul cel mai apropiat pentru un anumit pixel;

GL_LINEAR- se face o interpolare liniară a texel-ilor.

Filtrele următoare se folosesc doar ca filtre de minimizare în modul mipmap:

GL_LINEAR_MIPMAP_NEAREST

GL_LINEAR_MIPMAP_LINEAR

GL_NEAREST_MIPMAP_LINEAR

GL_NEAREST_MIPMAP_NEAREST

Filtrele se specifică în felul următor:

```
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER,  
GL_NEAREST) - filtru de minimizare NEAREST;
```

```
glTexParameterf(GL_TEXTURE_1D, GL_TEXTURE_MAG_FILTER,  
GL_LINEAR) – filtru de maximizare LINEAR.
```

8.4. Mipmap

Numărul de texeli nu este întotdeauna egal cu numărul de pixeli ai figurii care se texturează și atunci furnizăm texturi cu toate nivelele inferioare:

Exemplu:

level 0	64x64
level 1	32x32
level 2	16x16

```

level 3  8x8
level 4  4x4
level 5  2x2
level 6  1x1

```

În **figura 1** este texturat un poligon folosind tehnica mipmapping. OpenGL delimitează zonele pentru aplicarea fiecărui nivel de texturare. Texturile pe fiecare nivel au în acest caz culori solide.

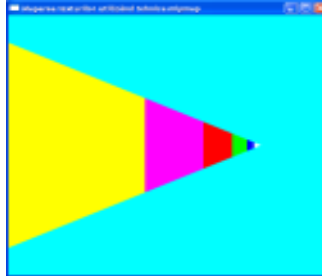


Fig. 1

8.5. Moduri de texturare

OpenGL dispune de trei moduri de texturare:

GL_DECAL – nu se ține seama de iluminare și de culoarea existentă în buffer-ul de culoare;

GL_MODULATE – se ține cont de intensitate, culorile texelilor sunt alterate;

GL_BLEND – se amestecă culoarea din buffer cu cea a texturii.

8.6. Desfășurarea lucrării

Aplicația 1. Aplicațiile GL_CLAMP, GL_CLAMP1, GL_CLAMP2, REPEAT demonstrează modul de folosire al wrappingului. Să se urmărească în fiecare aplicație felul în care se setează modul de wrapping și felul în care se dau coordonatele de texturare:

-checker.c.- mapează o imagine -tabla de șah- pe două dreptunghiuri. Acest program blochează (GL_CLAMP) texturarea dacă coordonatele texturii (s, t) depășesc valorile 0 respectiv 1.

-chess.c.- reia programul checker.c dar cu valori pentru coordonatele de texturare cuprinse între 0 și 3 în condițiile în care se păstrează parametrul GL_CLAMP.

- chess.c- reia programul "checker.c", pe o direcție se repetă texturarea iar pe cealaltă nu.

-checker2.c.- texturează cu imaginea unei table de șah două dreptunghiuri. Este repetată textura dacă coordonatele de texturare depășesc domeniul 0 respectiv 1.

Aplicația 2. Programul mipmap.c demonstrează folosirea tehnicii mipmap. Să se urmărească modificările care apar în acest caz.

Aplicația 3. Programul `texgen.c`. mapează o textură pe un ceainic cu generarea automată a coordonatelor de texturare. Este construită o textură 1D cu 3 octeți pentru fiecare texel, din 32 texeli 5 sunt roșii și 27 sunt verzi. Textura este redată ca hașuri pe ceainic.

Se folosește modul de texturare `GL_MODULATE` astfel că ceainicul este și iluminat și texturat. Să se schimbe orientarea dungiilor pe ceainic schimbând planul față de care se direcționează textura. Să se modifice generarea texturii astfel încât aceasta să nu mai fie solidară cu obiectul. Să se modifice culorile ceainicului.

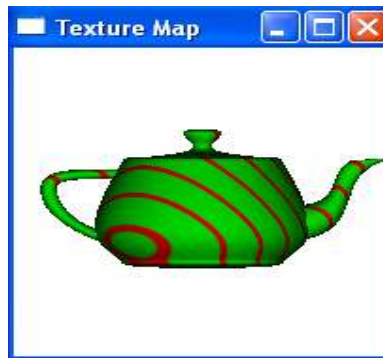


Fig. 2