Redarea suprafetelor 3D folosind texturi- 2

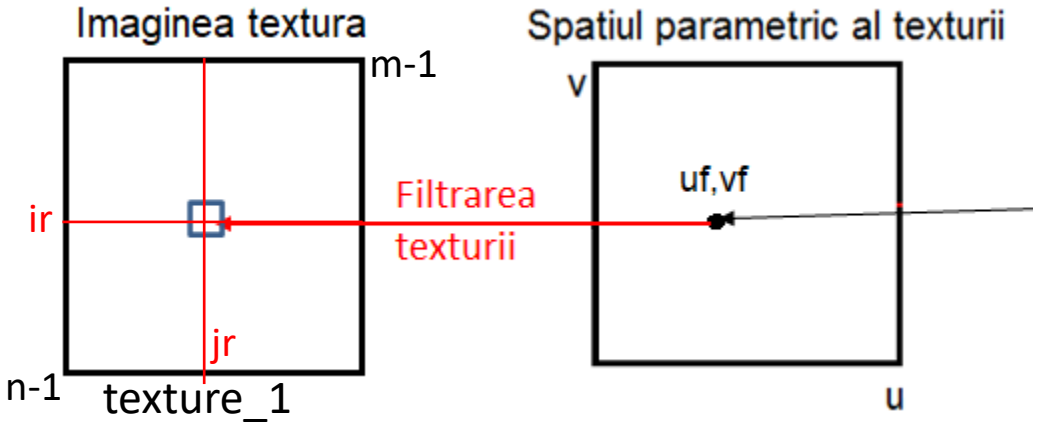
Prof. univ. dr. ing. Florica Moldoveanu

Curs Elemente de grafica pe calculator – UPB, Automatică şi Calculatoare
2020-2021

Filtrarea texturii(1)

```
//Fragment shader
uniform sampler2D texture_1;   // imaginea textura
in vec2 texcoord; //coordonatele textura – intrare pentru Fragment shader

layout(location = 0) out vec4 out_color;
void main()
{ vec4 color = texture2D(texture_1, texcoord);
out_color = color;
}
```



texture2D(texture_1, texcoord)

uf = texcoord.x
vf = texcoord.y

Adresa in spatiul texturii corespunzatoare coordonatelor (uf,vf) sunt 2 numere reale:

jr = uf*(m-1) ir = (1-vf)*(n-1)

Filtrarea texturii(2)

(ir, jr) – numere reale: i1 < ir < i2, j1 < jr < j2

[i1][j1], [i1][j2], [i2][j1], [i2][j2] sunt adrese in spatiul discret al texturii (adrese de texeli)

- Accesarea imaginii textura – prin numere intregi

Obtinerea unei culori din imaginea textura: Filtrarea texturii

vec4 color = texture2D(texture_1, texcoord);

2 posibilitati (Cele 2 moduri de filtrare s-au discutat în cursul 12):

1) color = textura [i][j], unde [i][j] este adresa cea mai apropiata de (ir,jr) in spatiul discret al texturii; în OpenGL:

glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER/GL_TEXTURE_MIN_FILTER, GL_NEAREST);

2) color = interpolare biliniara intre culorile texelilor de adrese [i1][j1], [i1][j2], [i2][j1], [i2][j2]

In OpenGL:

glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER/GL_TEXTURE_MIN_FILTER, GL_LINEAR);

Filtrarea texturii(3)

- Dimensiunea in imagine a suprafetei care se aplica textura variaza in functie de distanta de la suprafata la observator.
  - Numarul de texeli folositi la colorarea unui pixel variaza.
- Textura trebuie sa fie mărită sau micşorată în funcție de dimensiunea în spațiul ecran a suprafeței pe care se aplica.
- **Mărire (magnification)**. Exemplu: textura 256x256 texeli trebuie aplicata pe o suprafata de 512x512 pixeli.
  - un texel contribuie la culoarea mai multor pixeli adiacenți
- **Micşorare (minification)**. Exemplu:

  textura este de 256x256 texeli şi trebuie afişată pe o suprafață de 8x8 pixeli.
  - mai mulți texeli ar trebui să fie comprimați pentru a se obține culoarea unui pixel, de exemplu, calculând media culorilor texelilor

Filtrarea texturii(4)

- OpenGL dă posibilitatea să se specifice modul de accesare a texturii în cele 2 cazuri:
- glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER /GL_TEXTURE_MIN_FILTER, GL_NEAREST/ GL_LINEAR);
- **În cazul măririi**, filtrarea NEAREST nu dă rezultate bune, dar filtrarea LINEAR dă rezultate satisfacatoare.

În cazul micşorării:

- Presupunând că textura este de 256x256 texeli şi trebuie afişată pe o suprafață de 8x8 pixeli, ar trebui ca 1024 texeli sa fie „comprimați” pentru a se obține culoarea unui pixel.
- Filtrarea NEAREST are ca efect colorarea a 2 pixeli adiacenți folosind culorile a 2 texeli (foarte) îndepărtați între ei în spațiul texturii – eşantionare rară a texturii. Numai 64 de texeli din cei 256x256 sunt folositi la afişarea suprafeței.
- Filtrarea LINEAR nu are sens, deoarece conduce la interpolarea între culorile a 4 texeli îndepărtați între ei în spațiul texturii (eşantionare rară a texturii)

Metoda "mip-mapping" (1)

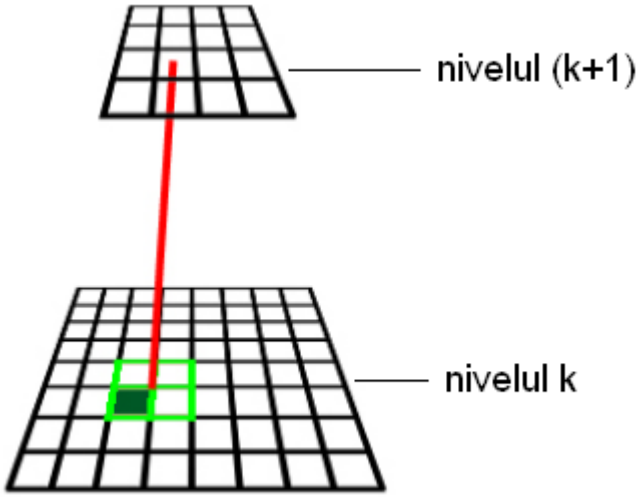
- Ambele moduri de filtrare produc imagini cu defecte, iar la schimbarea direcției observatorului apar pâlpâiri ale imaginii din cauza adresării aleatorii a texturii → Exemplu curs

- Totodată, accesarea aleatorie a texturii pentru pixeli adiacenți este ineficientă: creşte timpul necesar producerii imaginii.

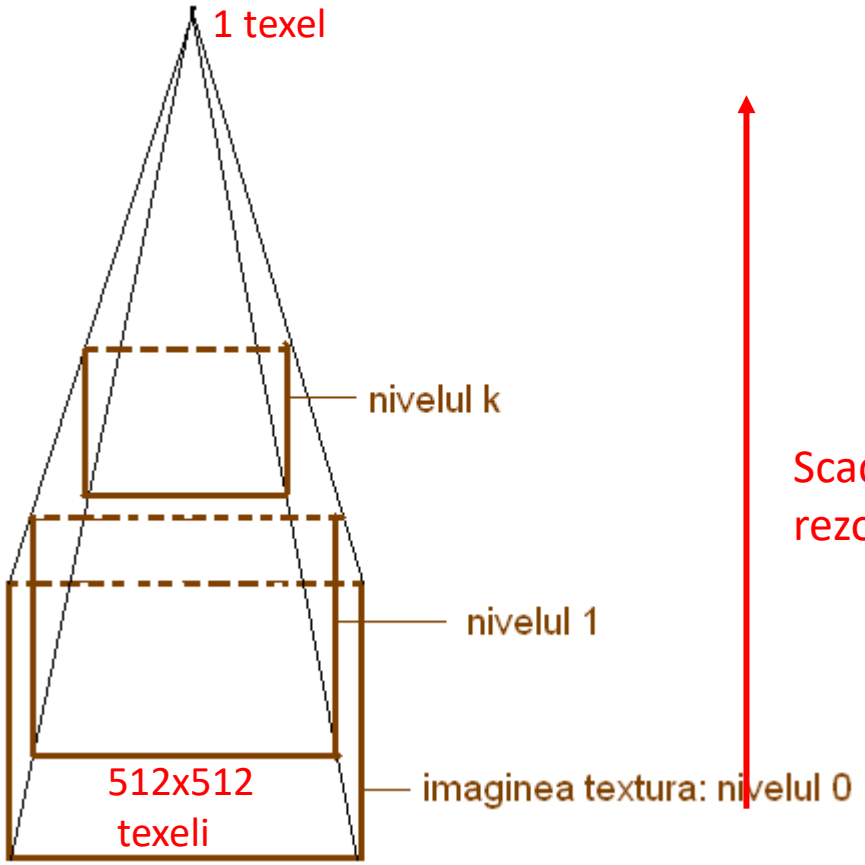
Soluția: metoda mip-mapping, care foloseste pentru filtrarea texturii un set de texturi pre-comprimate.

- Ideea mip-mapping: **utilizarea de imagini textura de rezolutii diferite în funcție de mărimea în imagine a suprafeței texturate**.
- Pornind de la o imagine textură, se pot obtine texturi de rezolutie mai joasa prin medierea culorilor din textura initiala.
- "MIP": acronim al frazei din limba latina: *multum in parvo*, insemnand "mult in puțin"

Metoda "mip-mapping" (2)
- Piramida mip-map -

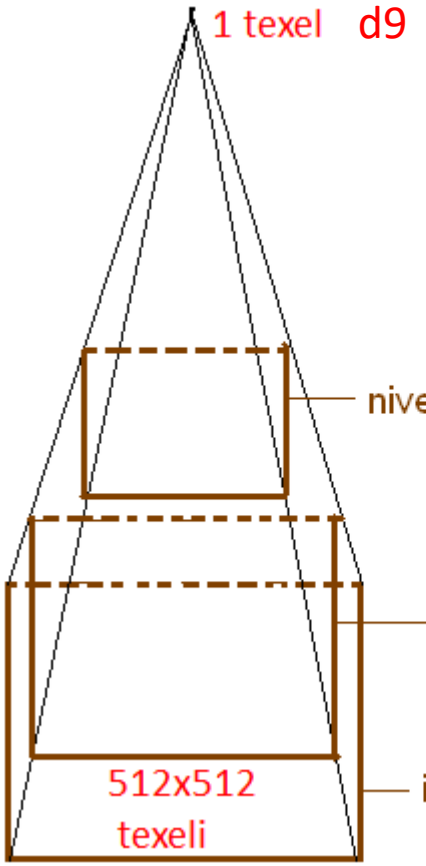


Fiecare nivel corespunde unui factor de compresie a texturii



La momentul rasterizarii unei primitive se alege imaginea textura de pe nivelul cu rezolutia cea mai apropiata de dimensiunea in imagine a primitivei.

Metoda "mip-mapping" (3)
- Piramida mip-map -



aria (u-v) coresp unui texel = $(1/2^9)x(1/2^9) = \mathbf{2^{2(9-9)}}$

d9 = **1:** compresie maxima – intreaga txtura este comprimata intr-un texel

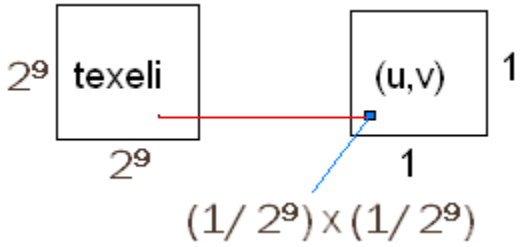
Fiecare nivel al piramidei corespunde unui factor de compresie a texturii:

aria (u-v) coresp unui texel = $(1/2^7)x(1/2^7) = \mathbf{2^{2(2-9)}}$

d2 = $\mathbf{1/\ 2^{(2-9)}}$

d1 aria (u-v) coresp unui texel = $(1/2^8)x(1/2^8) = \mathbf{2^{2(1-9)}}$

d1 = $\mathbf{1/\ 2^{(1-9)}}$



aria (u-v) coresp unui texel =

$\mathbf{(1/\ 2^9)\ x\ (1/\ 2^9)} = 1/2^{18} = \mathbf{2^{2(0-9)}}$

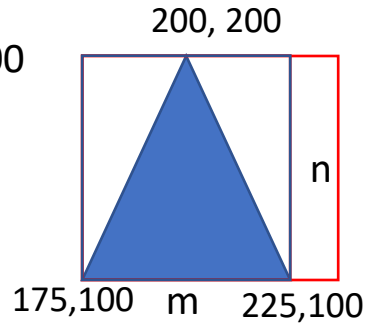
d0 = $\mathbf{1/\ 2^9}$

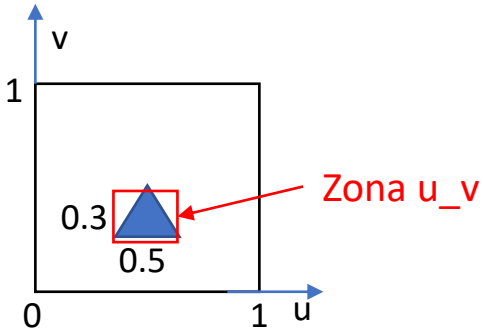
Metoda "mip-mapping" (4)

- Factorul de compresie a texturii pentru fiecare primitiva grafica se calculeaza pe GPU dupa transformarea varfurilor in spatiul ecran, tinand cont de suprafata primitivei in spatiul ecran si zona u-v folosita la texturarea sa.
- Suprafata primitivei se aproximeaza printr-un patrat, la fel si zona u-v (→ aceeasi compresie pe axele u si v)

Exemplu: m=50, n= 100

se aprox printr-un patrat cu latura de 100





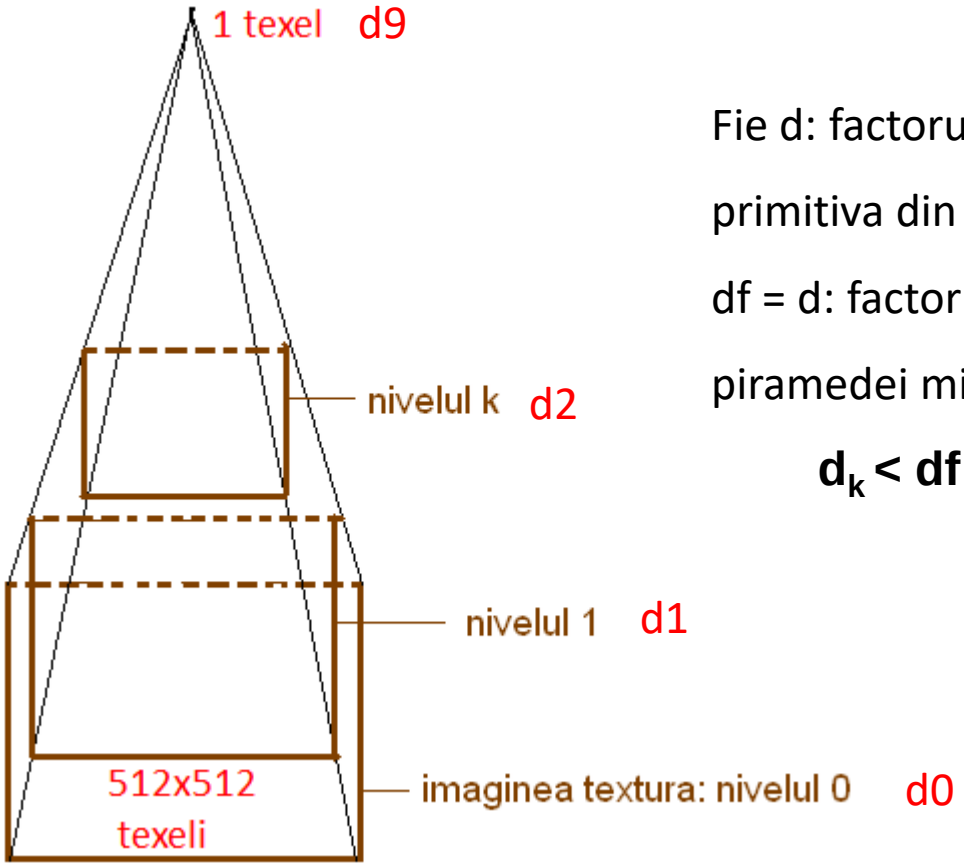
Zona u_v se aproximeaza prin patratul incadrator cu latura de 0.5

Factorul de compresie a texturii: f=0.5/100 = 1/200

Daca rezolutia texturii este de 512x512 texeli → 512/200 = 2.56

→ 2.56x2.56 = 6.55 texeli se comprima într-un pixel.

Metoda "mip-mapping" (5)
– filtrarea texturii -



Fie d: factorul de compresie a texturii pentru primitiva din care face parte fragmentul curent.

df = d: factorul de compresie folosit pt accesarea piramedei mip-map

$$d_k < df < d_{k+1}$$

Metoda "mip-mapping" (6)
– filtrarea texturii -

Piramida mip-map este adresata prin 3 coordonate: (u, v, d):

(u,v) prin care se selecteaza un texel/texelii de pe un nivel al piramidei, pe baza carora se calculeaza o culoare;

d, care este o masura a compresiei texturii intr-un pixel ecran, se foloseste pentru a selecta nivelul piramidei din care se calculeaza culoarea sau nivelurile intre care se va interpola pentru calculul culorii; **d poate fi considerata ca fiind coordonata verticala a piramidei**.

1) Fie uf si vf coordonatele folosite pentru calculul culorii unui fragment

2) Fie df factorul de compresie al texturii pentru fragmentul curent

Daca nu s-a cerut interpolare intre nivelurile mip-map

glTexParameteri(….GL_NEAREST_**MIPMAP**_**NEAREST** sau GL_LINEAR_**MIPMAP**_**NEAREST**),

atunci se determina nivelul piramidei pentru care factorul de compresie este cel mai apropiat de df.

Se calculeaza o culoare din textura corespunzatoare nivelului folosind (uf,vf)

Metoda "mip-mapping" (7)
– filtrarea texturii -

Daca s-a cerut prin OpenGL interpolare între nivelurile mip-map

glTexParameteri(….GL_NEAREST_**MIPMAP_LINEAR** sau GL_LINEAR_**MIPMAP_LINEAR**)

Fie: $d_k < df < d_{k+1}$

3) Folosind (uf, vf)

- se determina o **culoare textura de pe nivelul k**, C_k
- se determina o **culoare textura de pe nivelul k+1**, C_{k+1}

4) Se determina culoarea de afisare a fragmentului prin interpolare liniara intre C_k si C_{k+1}**:**

$C = C_k + dInterp\ (C_{k+1} - C_k)$

unde **dInterp** se determina din conditia: $df = d_k + dInterp\ (d_{k+1} - d_k) = 2^{k-9} + dInterp\ (2^{k-8} - 2^{k-9})$

Rezulta: **dInterp** $= 2^{9-k} * df - 1$

Exemplu:

Factorul de compresie: df=1/40 → d3 (1/64) < df < d4 (1/32)

Se va interpola intre culorile calculate pe nivelurile d3 si d4.

Metoda "mip-mapping" (8)
- Aprecieri -

- **Tehnica mip-mapping:**

  - da rezultate bune, chiar daca se bazeaza pe o serie de aproximari

  - este implementata de GPU: trebuie sa i se transmita imaginile textura ale piramidei, care poate fi obtinuta apeland functia OpenGL: glGenerateMipmap(GL_TEXTURE_2D);

- **Existenta piramidei de imagini simplifica filtrarea texturii in cazul micsorarii**: pe masura ce suprafata se indeparteaza de observator, culoarea se calculeaza din nivelurile din ce in ce mai mari ale piramidei.

- **Avantajelei metodei:**

  - Elimina artefactele vizuale produse de accesarea aleatorie a texturii la micsorarea suprafetei texturate.
  - Creste performanta procesului de extragere a culorilor din textura, atunci cand un numar mare de texeli trebuie folositi pentru un pixel.
  - Consumul de memorie suplimentar pentru memorarea piramidei este de 33%.