Redarea suprafetelor 3D folosind texturi

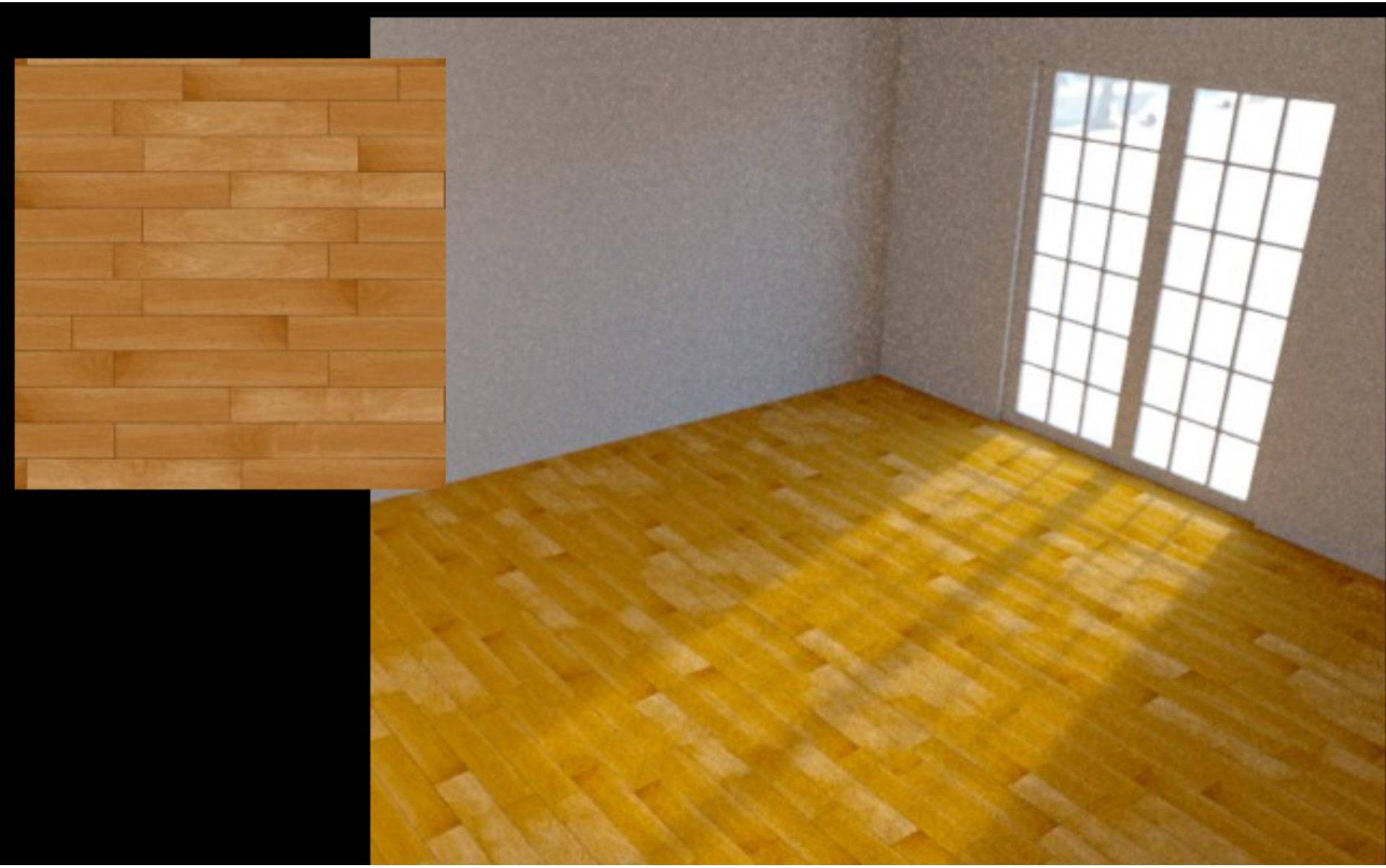
Prof. univ. dr. ing. Florica Moldoveanu

Texturi (1)

- Se folosesc in diferite scopuri:
  - Redarea unei imagini pe o suprafata

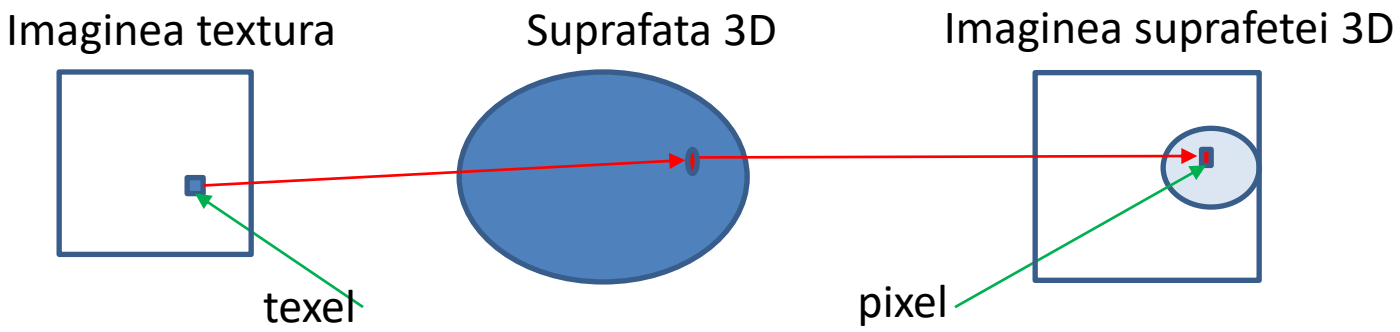
  - Simularea reflexiei/refractiei mediului inconjurator de catre o suprafata
  - Simularea detaliilor geometrice ale unei suprafete

s.a.

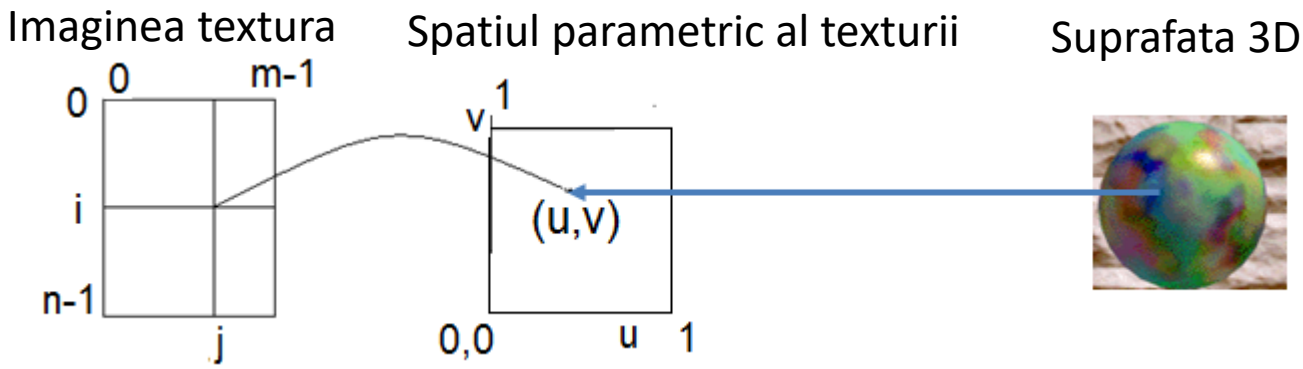


Aplicarea unei imagini textura pe o suprafata 3D

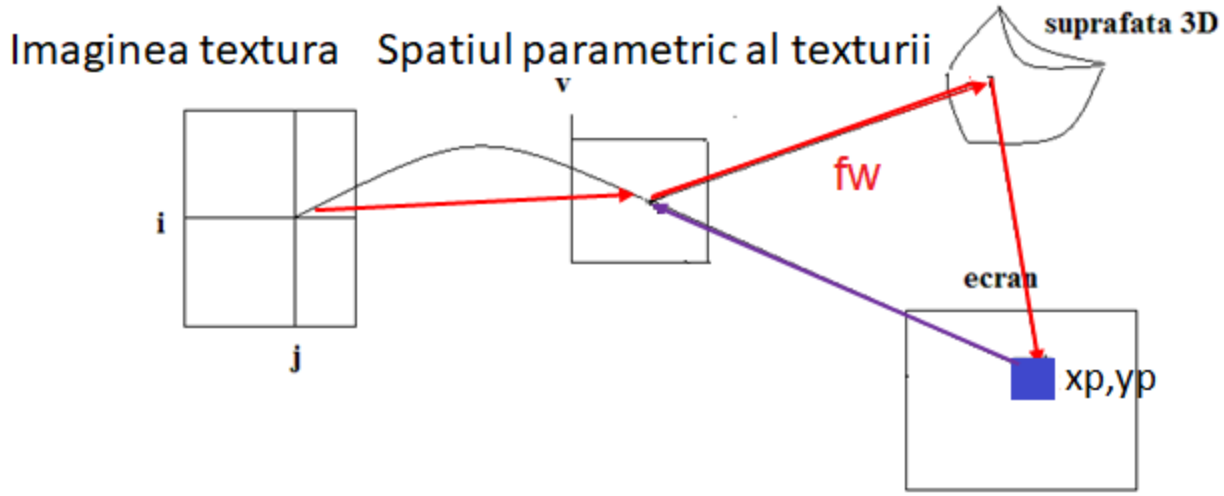
- Elementele imaginii textura sunt culori si se numesc « texeli ».



- Corespondenta dintre punctele suprafetei 3D si spatiul discret al texturii este realizata prin intermediul unui spatiu continuu, (u,v), numit *spatiul parametric al texturii,* 0 <= u,v =< 1



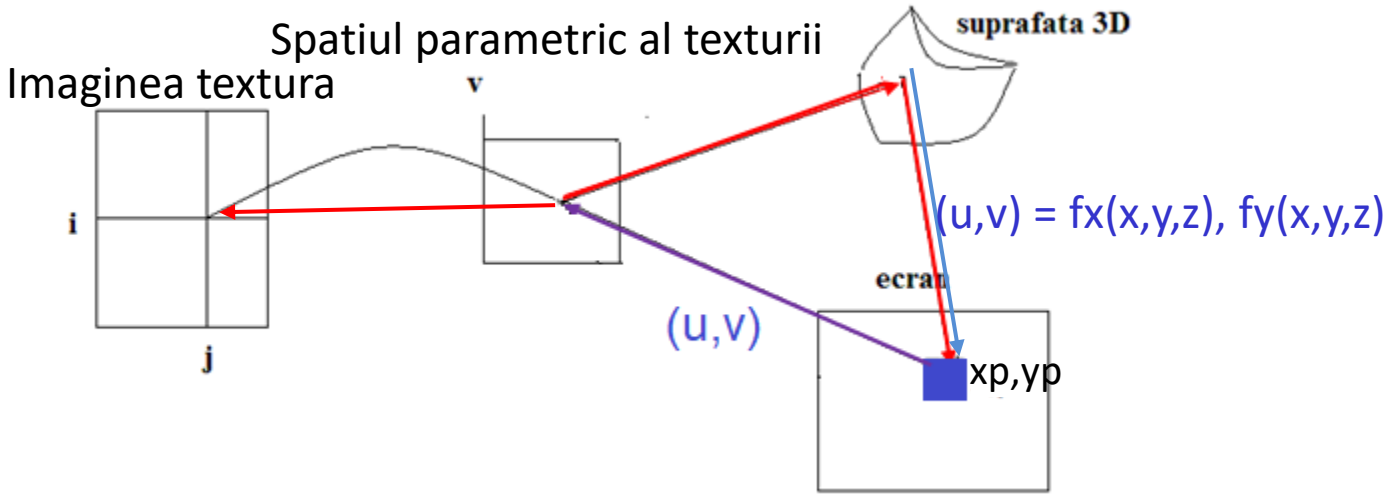
Maparea directa (forward mapping)



Se parcurge imaginea textura linie cu linie si se calculeaza pentru fiecare texel pixelul in care se afiseaza:

- adresa texel, (i,j) → adresa (u,v) → punct de pe suprafaţa 3D (x,y,z) → adresa pixel, (xp, yp)
- Se afiseaza pixelul (xp, yp) in culoarea texelului (i, j)
- Metoda nu este adecvata hardware-ului grafic actual (in plus, poate produce defecte)

Maparea inversa (inverse mapping)

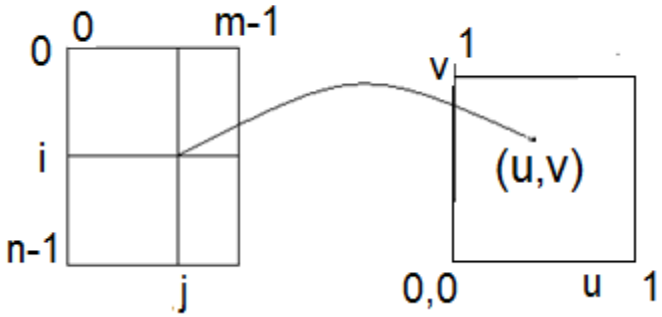


Se calculeaza pentru fiecare fragment rezultat din rasterizarea suprafetei 3D texelul/texelii din care se obtine culoarea textură.

- Pentru fiecare fragment (xp,yp) se calculeaza adresa (u,v) corespunzatoare punctului de pe suprafaţa 3D (x,y,z) care se afisaza în (xp,yp); adresa (u,v) → adresa (i,j) in spatiul discret al texturii de unde se va calcula culoarea fragmentului.
- Afiseaza fragmentul folosind culoarea obtinuta de la adresa (i, j) prin filtrarea texturii.

Obtinerea unei culori din imaginea textura in maparea inversa

Imaginea textura



Accesarea matricei textura din spatiul (u,v)

jr = u * (m-1)

ir = (1 - v) * (n-1)

⇐ (u, v)

(ir, jr) – numere reale

- Accesarea imaginii textura – prin numere intregi

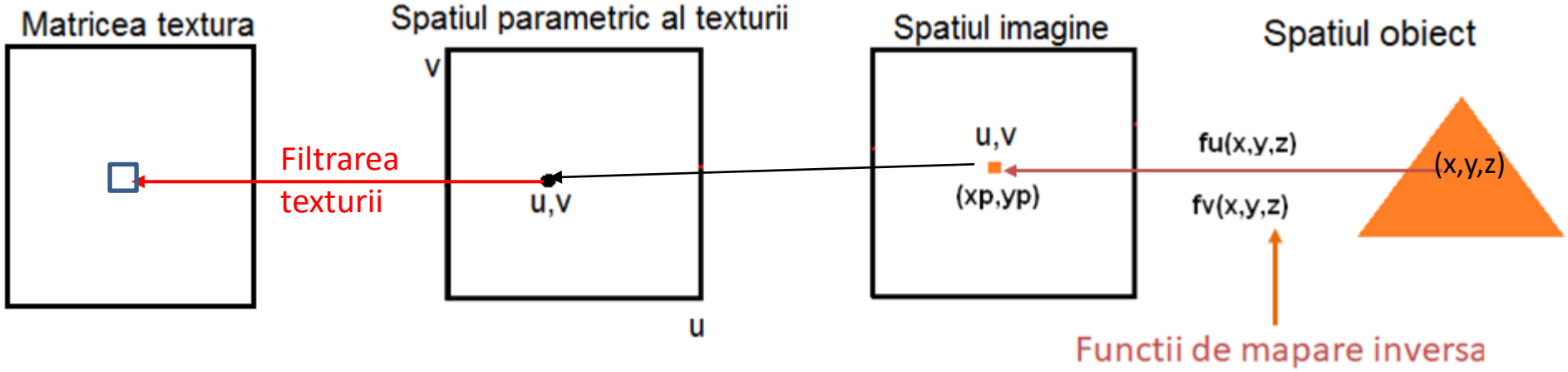
Obtinerea unei culori din imaginea textura: Filtrarea texturii

2 posibilitati:

1) Culoare = textura (i,j), unde (i,j) adresa cea mai apropiata de (ir,jr) in spatiul discret al texturii; în OpenGL: glTexParameteri(..,.., GL_NEAREST)
2) Culoare = interpolare_biliniara (textura, ir, jr); în OpenGL: glTexParameteri(..,.., GL_LINEAR)

Maparea inversa

- Este metoda uzuala de mapare a texturilor pe suprafetele 3D in banda grafica actuala.
- Se pleaca de la spatiul imagine, aplicarea texturii fiind inglobata în calculul culorii fragmentelor.
- Fiecare fragment rezultat din rasterizarea suprafatei 3D pe care se aplica o textură are asociata o adresa (u,v), obtinuta cu ajutorul unor functii de mapare inversa.
- Pe baza adresei (u,v) a fragmentului se extrage din imaginea textura o culoare, prin operatia de filtrare a texturii.



Functii de mapare inversa

Definesc o corespondenta intre punctele unei suprafete 3D si punctele spatiului (u,v) :

$u = f_u(x,y,z)$; $v = f_v(x,y,z)$

In general, aceasta corespondenta se stabileste pentru varfurile primitivelor rasterizate, adresele (uf,vf) pentru fragmentele rezultate din rasterizare obtinandu-se prin interpolare intre adresele (u,v) asociate varfurilor.

$f_u(x,y,z)$, $f_v(x,y,z)$ Se numesc **functii de mapare inversa**

- E pot defini functii de mapare inversa numai pentru suprafeţe definite intr-un spatiu parametric, (s,t):

  s= Fs(x,y,z), t=Ft(x,y,z)

- a.î. se poate stabili o corespondenta intre spatiul parametric al suprafetei şi spatiul parametric al texturii: **(s,t) → (u,v)**

Functii de mapare standard(1)

1. Maparea plana (maparea pe un plan)

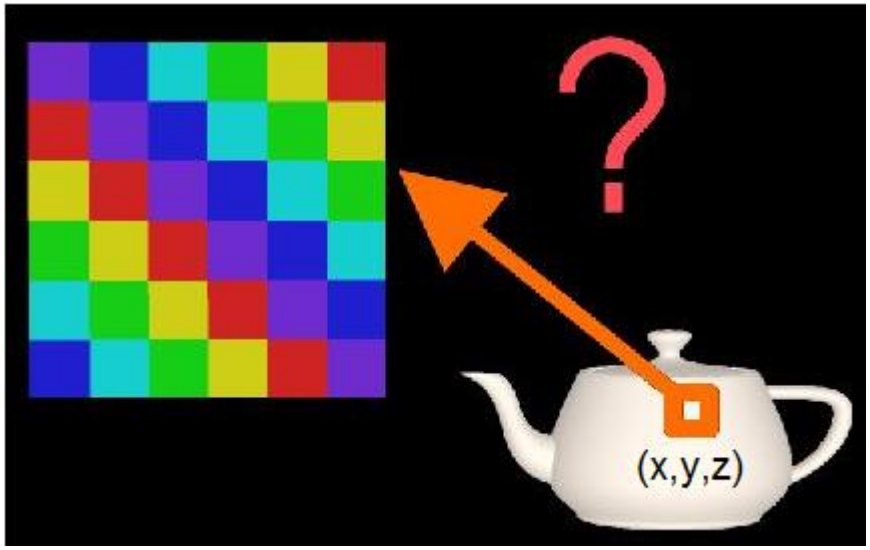
Ecuatiile care descriu aceasta mapare sunt:

$u = su*x - ou$ $\quad x \in X,\ y \in Y$

$v = sv*y - ov$ $\quad 0 <= u, v <= 1$

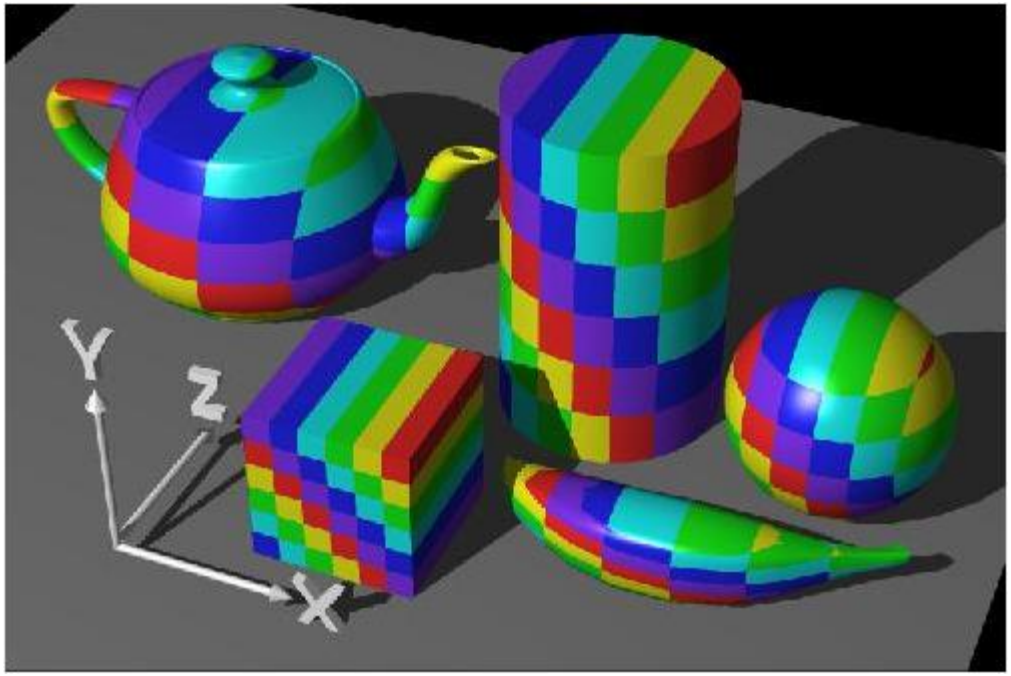
unde : su=1/X, sv=1/Y sunt factorii de scalare pentru transformarea spatiului dreptunghiular [X, Y] in [0,1] iar (ou,ov) este originea texturii: determina modul în care se aşază textura pe suprafaţă.

(u, v) ← (x, y) ← (x, y, z) - se renunta la coordonata z: suprafata 3D devine o suprafata plana



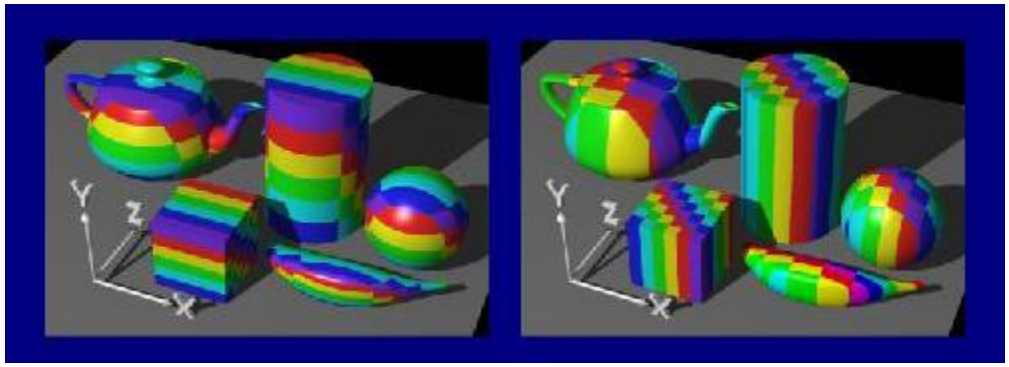
http://www.siggraph.org/education/materials/HyperGraph/mapping/r_wolfe/r_wolfe_mapping_2.htm

Exemple de mapari plane



Maparea plana prin renuntarea la coordonata z: toate punctele suprafetei cu aceleasi coordonate (x,y) au aceeasi culoare.

http://www.siggraph.org/education/materials/HyperGraph/mapping/r_wolfe/r_wolfe_mapping_2.htm

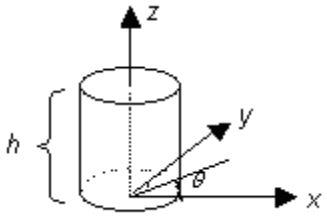


Maparea plana prin renuntarea la coordonata x, respectiv y.

Functii de mapare standard(2)

2. Maparea cilindrica

- Stabileste corespondenta dintre coordonatele (x,y,z) ale unui punct de pe o suprafata cilindrica si spatiul parametric al texturii, (u, v)



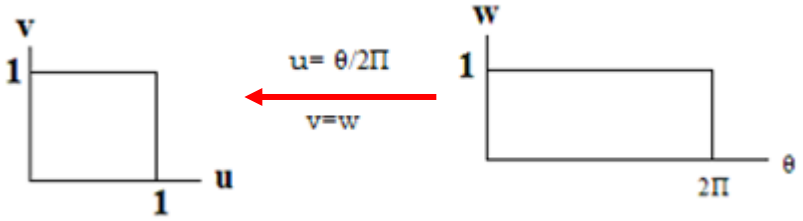
Ecuatiile parametrice ale unei suprafete cilindrice:

x=r*cos(θ)
y=r*sin(θ) 0<= θ< 2π, 0 < = w <= 1
z=w*h

- Coordonatele cilindrice (*θ, w*) ale punctului (x, y, z)

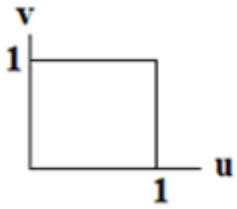
θ = arctg(y/x)
w = z/h

-Se pune in corespondenţă spatiul parametric al texturii, (u,v), spatiului parametric al suprafetei cilindrice:

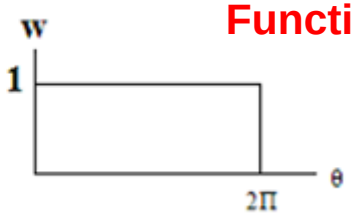


Functii de mapare standard (3)

θ = arctg(y/x)
w = z/h



u = θ/2Π ; v=w

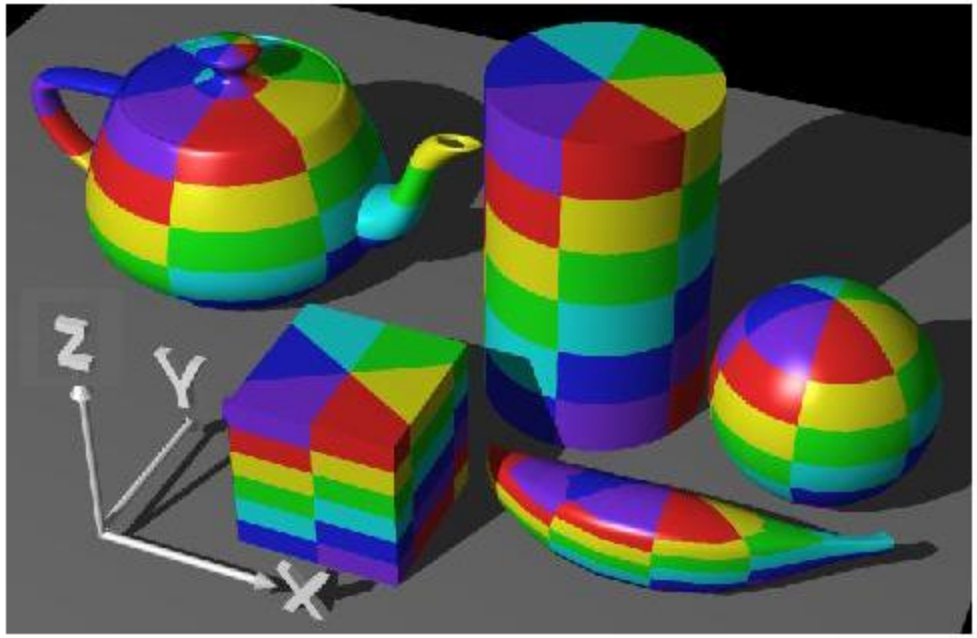


Functiile de mapare cilindrica

u = (1/2π)arctg(y/x)
v = z/h

Textura este “înfaşurată” în jurul obiectului

Transformarea patratelor texturii in punctele suprafetei de z= zmin si z= zmax



Redarea supratelelor folosind texturi

Functii de mapare standard(4)

3. Maparea sferica

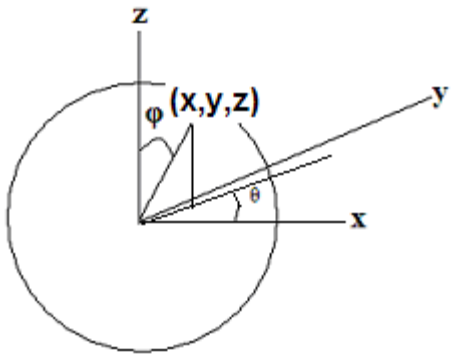
❖ Stabileste corespondenta dintre coordonatele (x,y,z) ale unui punct de pe o suprafata sferica si spatiul parametric al texturii, (u, v)

- Ecuatiile parametrice ale sferei:

$x = r*\cos(\theta)*\sin(\varphi) \quad 0 <= \theta < 2\pi$

$y = r*\sin(\theta)*\sin(\varphi) \quad 0 <= \varphi <= \pi$

$z = r*\cos(\varphi)$



- Coordonatele sferice ale punctului (x,y,z):

$\theta = \text{arctg}(y/x)$

$\varphi = \arccos(z/r)$

Punand in corespondenta spatiul (u,v) spatiului parametric al sferei rezulta:

$u = \theta/2\pi$

$v = \varphi/\pi$

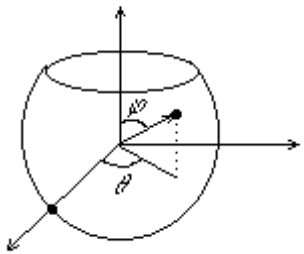
Functii de mapare standard (5)

u = θ/2π
v = φ/ π

$$\begin{cases} u = \dfrac{1}{2\pi} arctg\left(\dfrac{y}{x}\right) \\ v = \dfrac{1}{\pi}\arccos\left(\dfrac{z}{r}\right) = \dfrac{1}{\pi}\arccos\dfrac{z}{\sqrt{x^2+y^2+z^2}} \end{cases}$$

Functiile de mapare sferica

Maparea pe un sector de sfera



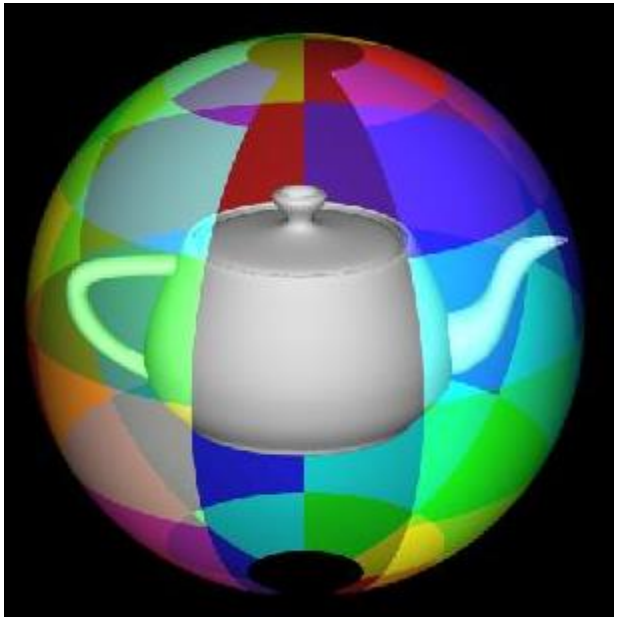
0 <= θ <= π/2
π /4 <= φ <= π/2

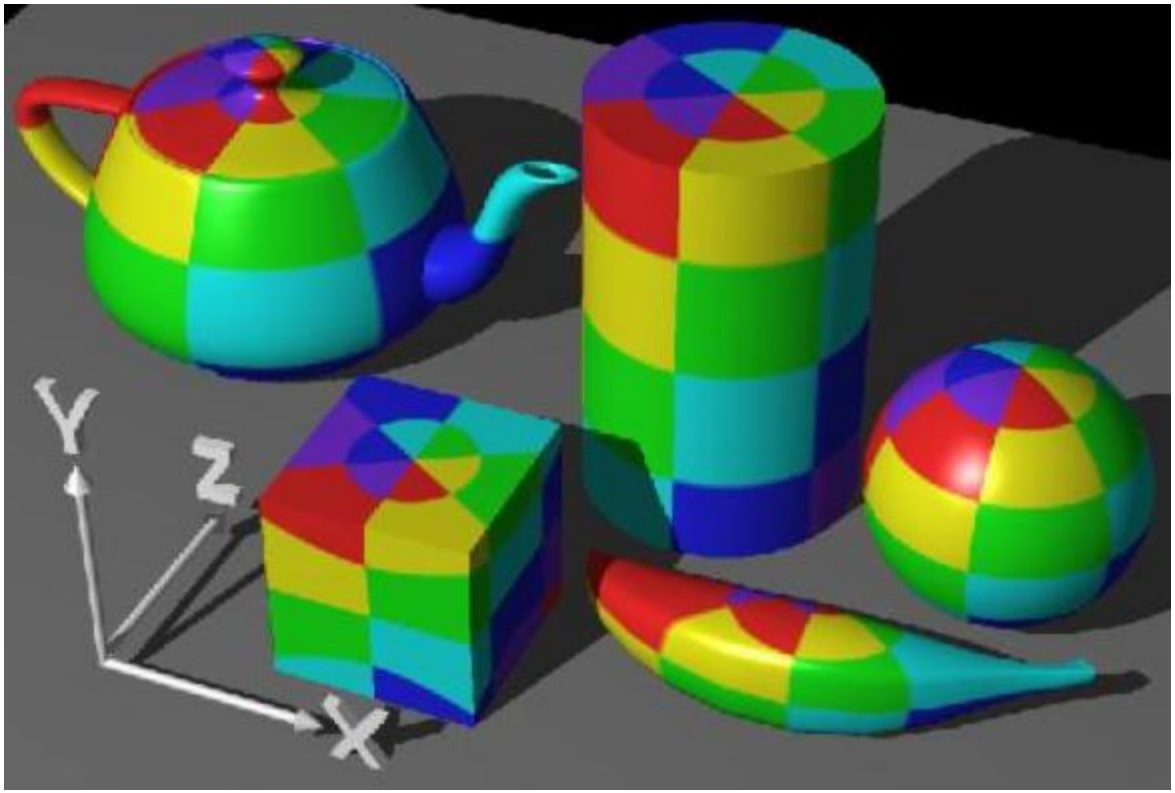
Functiile de mapare

u = θ/(π/2)
v = ((π /2) – φ) / (π /4)

Exemple de mapari sferice

Maparea sferica “întinde” patratele texturii la nivelul ecuatorului si le “comprimă” proportional cu aproprierea de poli.

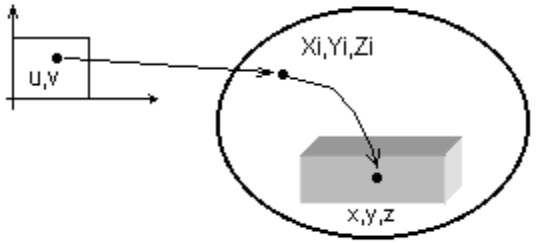




http://www.siggraph.org/education/materials/HyperGraph/mapping/r_wolfe/r_wolfe_mapping_2.htm

Maparea în 2 pasi (1)

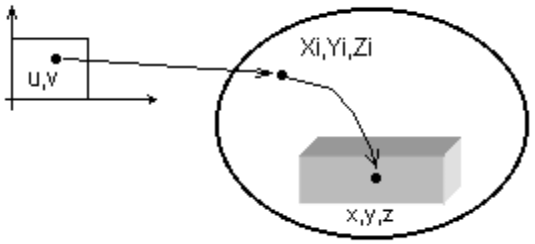
- Este o tehnica de mapare independenta de forma obiectului pe care se aplica textura. Metoda a fost definita de Bier si Sloan (1986).
- Daca pentru suprafata texturata nu exista o functie de mapare inversa, atunci, se stabileste o corespondenta între punctele suprafetei texturate si punctele unei suprafete 3D pentru care exista o functie de mapare inversa, numita in continuare « suprafata intermediara ».
- Aplicarea texturii este descompusa in doua operatii,



al caror efect este :

  - **mularea texturii pe suprafata intermediara;**
  - **mularea suprafetei intermediare pe suprafata de texturat.**

Maparea în 2 pași(2)



a) **maparea S (mapare inversa):**
u = fu(xi, yi, zi)
v = fv(xi, yi, zi)

b) (xi, yi, zi) ← **maparea O** ← (x, y, z)

a) Mularea texturii pe suprafata intermediara se realizeaza prin functia de mapare inversa. Aceasta este numita « **mapare S** », adica mapare pe *suprafata* intermediara.

b) Mularea suprafetei intermediare pe suprafata de texturat este numita « **mapare O** ». Ea trebuie sa puna in corespondenta fiecarui punct (x,y,z) al suprafetei (*obiectului*) de texturat, un punct (xi, yi, zi) al suprafetei intermediare

Maparea în 2 pași(3)

Maparea S

Se folosesc 4 tipuri de suprafete intermediare :

1. un plan cu o orientare oarecare; →maparea S: mapare plana
2. sfera; →maparea S: mapare sferica
3. cilindrul; →maparea S: mapare cilndrica
4. fețele unui cub; →maparea S: mapare plana

Alegerea tipului suprafetei intermediare este dependenta de forma obiectului pe care trebuie sa fie aplicata textura.

Maparea O

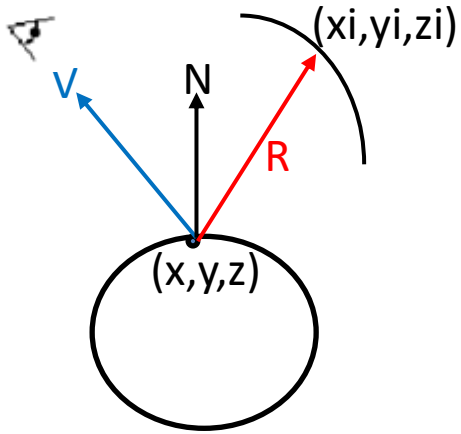
Autorii metodei au definit 4 tipuri de mapari O:

Maparea în 2 pași(4)

Mapari O

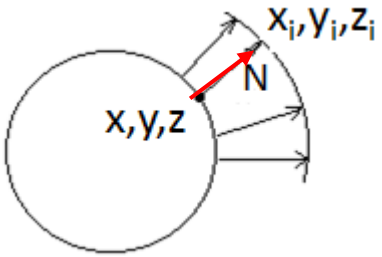
(x,y,z) – un punct al suprafetei , pentru care se doreste calculul coordonatelor (u,v)

(xi, yi, zi) – punctul corespunzator lui (x,y,z) pe suprafata intermediara



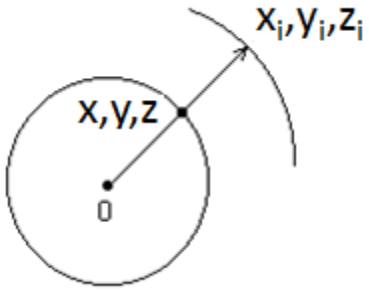
1. (xi, yi, zi), se obtine intersectand suprafata intermediara cu vectorul reflectat al vederii in punctul (x,y,z). R este simetric cu V fata de N.

Metoda este dependenta de pozitia observatorului (se foloseste la « maparea mediului inconjurator »)

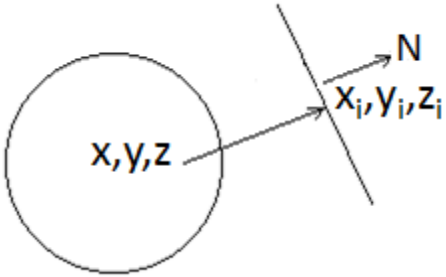


2. (xi,yi,zi) este intersectia suprafetei intermediare cu normala la obiect în (x,y,z).

Maparea în 2 pași(5)



3. (xi,yi,zi) este intersectia suprafetei intermediare cu dreapta care trece prin (x,y,z) si centroidul obiectului.



4. (xi,yi,zi) este intersectia suprafetei intermediare cu dreapta care trece prin (x,y,z) si are directia normalei la suprafata intermediara.

Acest tip de mapare O este corelat cu o mapare S pe un plan sau pe fetele unui cub.

Combinatia « mapare S pe cilindru » - « mapare O de tip 4 » este numita « shrinkwrap ».

Aplicarea texturilor pe fetele unei retele poligonale 3D (1)

In sistemele grafice actuale, orice suprafata 3D este descompusa, in vederea redarii sale, într-o rețea de fațete triunghiulare: textura se aplica pe suprafetele triunghiulare la momentul rasterizarii lor.

Pentru fiecare fragment rezultat din rasterizare se poate obtine o culoare din textură in 2 moduri:

1. In fragment shader, se calculeaza coordonatele (u,v) corespunzatoare adresei fragmentului (xf,yf,zf), folosind functii de mapare standard sau maparea in 2 pasi:

   (uf,vf) ← functii de mapare (xf,yf,zf)

2. In fragment shader, se utilizeaza coordonatele (uf,vf) calculate de rasterizator prin interpolarea coordonatelor (ui,vi) atasate varfurilor primitivei din care face parte fragmentul.

   - Pentru rezultate corecte este necesara efectuarea unei interpolari perspectiva. Interpolarea liniara este mai rapida dar poate produce defecte.
   - Metode mai exacte (ex. Mip-mapping) calculeaza coordonatele textura ale unui fragment tinand cont de aria suprafetei textură comprimată in fragment.

Aplicarea texturilor pe fetele unei retele poligonale 3D(2)

- In cazul utilizarii mai multor texturi la redarea unei primitive, in fragment shader se extrage cate o culoare din fiecare textura, fie calculand adresele (uf,vf) in fragment shader, fie folosind coordonatele textura interpolate. Culorile extrase din diferite texturi pot fi combinate in diferite moduri pentru a se obtine culoarea fragmentului.

- Daca o imagine textura trebuie mulata pe intreaga retea de fatete care aproximeaza o suprafata, atunci varfurile retelei trebuie sa se mapeze pe intreg spatiul 0 <= (u,v) <= 1.
- Coordonatele (ui,vi) atasate varfurilor unei primitive se pot obtine intr-o aplicatie OpenGL printr-o mapare in 2 pasi.
- Aplicatiile de modelare a suprafetelor 3D permit atasarea de coordonate textura varfurilor suprafetei. Obiectele importate contin coordonatele (u,v) atasate varfurilor obiectului.
- Coordonatele textura ale varfurilor se transmit ca intrari la Vertex shader.

Transparenta texturii (1)

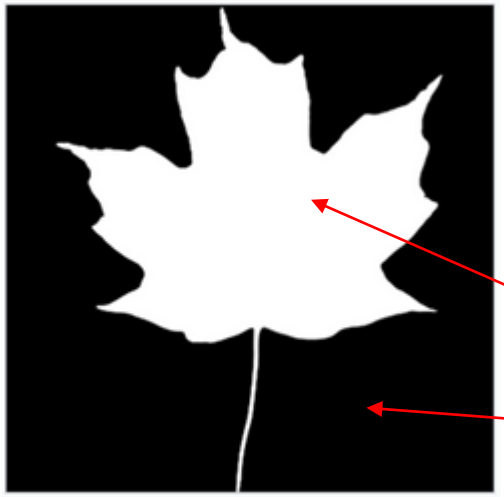
- Imaginea textura poate avea o componenta aditionala, numita “canalul alfa”, prin care se poate modela transparenta texturii.
- Fiecare texel este in acest caz reprezentat prin 4 componente: (R,G,B,A), unde A reprezinta opacitatea.
  - daca A = 1, atunci texelul este complet opac
  - daca A=0, texelul este complet transparent
- Culoarea unui fragment se poate obtine: dintr-o textura, combinand culori din mai multe texturi, combinand culoarea fragmentului calculata pe baza geometriei (modelul Gouraud sau Phong) cu o culoare extrasa dintr-o textura (sau mai multe).
- Atunci cand culoarea fragmentului are opacitatea $0< A < 1$, se poate combina culoarea sa cu aceea a pixelului destinatie, printr-o operatie de “amestec” între cele 2 culori, specificata prin functia glBlendFunc(..) (vezi curs “Transparenta în modelul de iluminare locala”). Detalii la: https://open.gl/textures.

Transparenta texturii (2)

- Atunci cand culoarea fragmentului are opacitatea A=0 sau sub o anumita valoare, se poate cere eliminarea sa (suprascrierea unui pixel cu ceva transparent nu are efect).

```
uniform sampler2D texture1;
in vec2 TexCoords;
out vec4 FragColor;
//Fragment shader
void main() {
    vec4 texColor = texture(texture1, TexCoords);
    if(texColor.a < 0.1) discard; // fragmentul este eliminat – nu va fi procesat in continuare
    FragColor = texColor;
}
```

Transparenta texturii (3)



Imaginea textura

Desenul frunzei este memorat pe canalul alfa al texturii.

Toti texelii texturii au culoarea albastru.

(0,0,1,1) – texeli opaci

(0,0,1,0) – texeli transparenti

Imaginea texturii aplicată pe o suprafață albă.

```
void main() {
  vec4 texColor = texture(texture1, TexCoords);
 // pentru eficiență:
   if(texColor.a == 0) discard;
   FragColor = texColor;
 }
```

Filtrarea texturii (1)

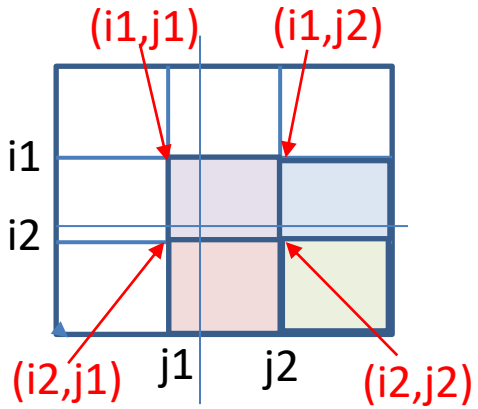
- O textura este definita pe un spatiu discret in timp ce coordonatele textura sunt valori reale. In general, o pereche (uf,vf) nu corespunde unui texel al texturii.
- Calculul culorii dintr-o textura folosind coordonatele (uf,vf), se numeste “filtrarea texturii”.

Cazul 0<= uf, vf <= 1

Fie (iv,ju) coordonatele reale textura corespunzatoare unei adrese (uf,vf).

Sunt doua moduri standard de filtrare:

1. Se considera culoarea texelului cu adresa cea mai apropiata de (iv,ju). In OpenGL: GL_NEAREST.

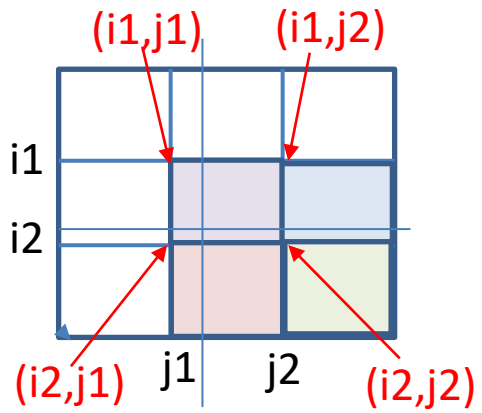


i1 = 12, i2= 13,
j1 = 4, j2 = 5
iv = 12.8, ju = 4.3
Culoare_textura ← textura[i2][j1]

- rapida dar poate produce defecte in imagine
- este metoda de filtrare implicita

Filtrarea texturii (2)

2. Se efectueaza o **interpolare biliniara** intre culorile celor mai apropiati texeli din spatiul texturii (în OpenGL: GL_LINEAR):



Culoare_textura ← interpolare intre culorile texelilor de adrese [i1][j1], [i1][j2], [i2][j1], [i2][j2]

- scade viteza de calcul a culorii fragmentului
- poate produce un efect de înceţoşare

- Cele 2 moduri de filtrare se pot specifica separat pentru cele 2 cazuri:

glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER/GL_TEXTURE_MIN_FILTER, GL_NEAREST/GL_LINEAR).

➢ Mărirea texturii – suprafața unui texel se mapează pe mai mulți pixeli adiacenți; se pot folosi ambele moduri de filtrare, dar filtrarea prin interpolare produce imagini cu un aspect mai placut.

➢ Micsorarea texturii – suprafata unui pixel se mapeaza pe o zona mai mare decat cea a unui texel. Ar trebui ca mai multi texeli sa fie comprimati pentru a se obtine culoarea pixelului. Nici unul dintre cele 2 moduri de filtrare nu da rezultate bune. **Soluția: metoda mip-mapping.**

Filtrarea texturii (3)

Interpolarea biliniara intre culorile din matricea textura

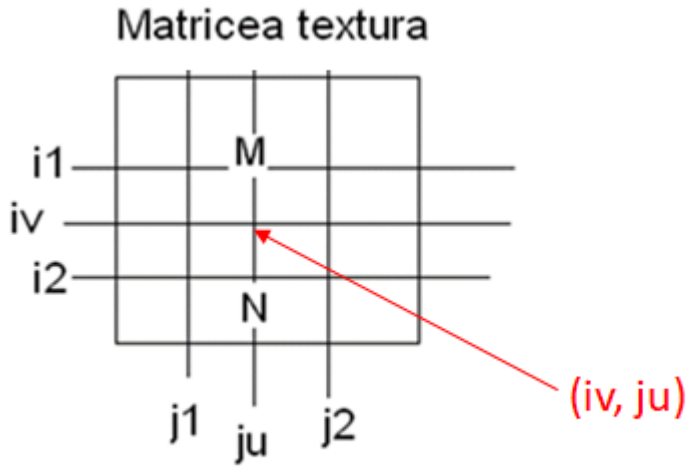
- Daca (ju, iv) sunt adresele reale corespunzatoare valorilor (uf,vf) in spatiul textura,

  iv = (1-vf)*(n-1); ju = uf*(m-1)

  n- numarul de linii ale matricei textura; m – numarul de coloane

  i1< iv < i2, i2 = i1 + 1 j1< ju < j2, j2 = j1 + 1

- Culoarea extrasa din matricea textura se calculeaza astfel:



M(i1,ju); N(i2,ju);

C(i,j) – culoarea texelului de indici (i,j)

Cul(M) = (j2-ju)*C(i1,j1) + (ju-j1)*C(i1,j2)

Cul(N) = (j2-ju)*C(i2,j1) + (ju-j1)*C(i2,j2)

Culoare_textura = (i2-iv)*Cul(M) + (iv-i1)*Cul(N)

i1, i2, j1, j2 sunt indici de linie/coloana in matricea textura

Filtrarea texturii (4)

Cazul uf, vf in afara domeniului [0,1]

- Coordonatele (ui, vi) asociate varfurilor pot fi in afara domeniului [0,1]. Rezulta:

  coordonatele (uf, vf) ale unui fragment pot fi in afara domeniului [0,1].

- Coordonatele (uf, vf) rezultate in procesul de rasterizare sunt transformate în coordonate 0<= u, v <=1.

- Modul de transformare a coordonatelor (u,v) din afara domeniului [0,1] se specifica separat pentru fiecare axa: u/v.

- Exista cateva moduri standard de transformare a coordonatelor textura in domeniul [0,1]. Doua dintre acestea sunt: “clamping” si “wrapping (sau “repeating).

Clamping (Comprimare)

Coordonatele uf/vf sunt transformate in u/ v astfel:

u = min(max(0,uf),1) v = min(max(0,vf),1)

Deci: daca uf / vf <0, atunci u / v =0; daca uf / vf >1, atunci u / v=1

daca 0<= uf, vf <=1, atunci u, v = uf, vf.

Filtrarea texturii (5)

Clamping (Comprimare)

Aplicarea acestei transformari are ca efect propagarea culorii de la marginea texturii de-a lungul directiei in care coordonata textura este < 0 sau > 1.

Filtrarea texturii (6)

Wrapping (Repetare)

- Pentru accesarea texturii se calculeaza (u,v) cu formula:

  u = uf- [uf] v= vf-[vf]

  unde [w] reprezinta cel mai mare intreg mai mic sau egal cu w.

  De exemplu, (-0.1, 1.5) se transforma in (-0.1 – (-1), 1.5 – (1)) = (0.9, 0.5).

- Acest mod de transformare permite repetarea unei texturi, producand un efect periodic.
- La utilizarea acestui mod este necesar ca laturile stanga/dreapta, respectiv sus/jos ale imaginii textura se se potriveasca, astfel incat sa nu se observe marginile texturii.

- Daca pentru una dintre coordonate se utilizeaza transformarea “comprimare” iar pentru cealalta “repetare”, textura se numeste “cilindrica”.
- Daca pentru ambele coordonate se utilizeaza modul “repetare”, textura se numeste “toroidala”.

Filtrarea texturii (7)

Wrapping (Repetare)