UNIVERSITATEA TEHNICĂ A MOLDOVEI
Facultatea Calculatoare, Informatică şi Microelectronică
Departamentul Informatică şi Ingineria Sistemelor

GRAFICA PE CALCULATOR

TEMA 10. TEXTURI

l. u., dr. NASTAS Andrei

CONȚINUT

10.1. Generalități privind texturile
10.2. Generarea de texturi prin metoda fractală
10.3. Filtrarea texturilor

10.1. Generalități privind texturile

Textura se definește ca o structură compusă din elemente primitive (tipare) care se repetă mai mult sau mai puțin regulat, fără ca unul sau altul din ele să atragă atenția în mod special.

Texturile pot fi studiate din punctul de vedere al:

– analizei de imagini (fiind utile în domenii: segmentarea imaginilor, clasificarea imaginilor sau codificarea imaginilor),

– sau al sintezei de imagini.

– în aplicațiile din domeniul analizei de imagini, textura este de fapt o caracteristică a unei imagini, ea cuantifică prin intermediul unor indicatori cantitativi, aspectul unei regiuni a imaginii.

– din punctul de vedere al sintezei de imagini textura este ea însăși o imagine care poate fi aplicată pe o anumită suprafață (plană sau nu) dintr-o scenă de vizualizat.

Caracteristici de tip textură ale unei imagini pot fi determinate și prin calculul transformatei Fourier a imaginii de analizat sau prin calculul distribuției nivelelor de gri (sau a culorilor) din punctele imaginii.

Dintre domeniile tipice de aplicație a texturilor (în domeniul analizei imaginilor) menționăm – clasificarea, segmentarea, codificarea.

10.1. Generalități privind texturile

1. Clasificarea texturilor cuprinde aplicații ca:

– recunoașterea automată a țesuturilor în imagini obținute prin tomografie computerizată (CT - computer tomography) sau rezonanță magnetică (MR – magnetic rezonancy).

– controlul calității diferitelor produse (hârtie, timbre, plăci cu circuite imprimate).

2. Segmentarea este un proces de transformare a unor imagini, prin care se identifică regiunile imaginii în interiorul cărora valorile unor parametrii caracteristici sunt aproximativ aceleași.

Textura suprafețelor poate constitui un criteriu de segmentare a imaginilor, mai ales dacă alte criterii (cum ar fi culoarea sau intensitatea) nu sunt suficiente pentru a distinge între ele regiunile unei imagini.

3. Codificarea imaginilor se utilizează pentru ca acestea să ocupe cât mai puțin spațiu, păstrând totuși întreaga informație conținută în imagine, gradul de compresie al imaginii poate fi îmbunătățit prin analiza imaginii.

10.1. Generalități privind texturile

În domeniul sintezei de imagini, pot fi modelate obiecte naturale folosind diferite tehnici de generare ale texturilor cum ar fi:

– obiecte biologice (suprafețe de țesuturi, plante);
– suprafețe caracteristice formelor de relief (munți, mare);
– suprafețele corpurilor naturale (de exemplu cristale);
– universal galactic;
– obiecte construite de om (covoare, suprafețele pereților clădirilor).

În cele ce urmează ne vom referi în special la acest domeniu de aplicație al texturilor.

10.2. Generarea de texturi prin metoda fractală
10.2.1. Noțiunea de fractal

Noțiunea de fractal se definește în cadrul unui spațiu metric.

Un spațiu metric este cuplul (*X*, *d*), unde *X* este o mulțime nevidă (ale cărei elemente se numesc punctele spațiului metric), iar *d*: $X^2 \geq R$ este o funcție (distanță) ce îndeplinește următoarele proprietăți:

1. (pozitivitate) orice $x, y \in X \implies d(x, y) = 0$,
   d(*x*, *y*) = 0 dacă și numai dacă *x* coincide cu *y*,
2. (simetrie) orice $x, y \in X \implies d(x, y) = d(y, x)$,
3. (inegalitatea triunghiului) orice $x, y, z \in X \implies d(x, z) = d(x, y) + d(y, z)$,

Definiție: ***Un fractal*** *este o submulțime a unui spațiu Euclidian pentru care dimensiunea Hausdorff-Besicovitch este strict mai mare decât dimensiunea topologica.*

Dimensiunea topologica este un număr întreg:

1. în cazul unei curbe,
2. în cazul unei suprafețe,
3. în cazul unui volum.

10.2.2. Curbe autoasemenea

Multe astfel de curbe (fractali de dimensiune topologica 1) poarta numele unor matematicieni celebri:
– Peano,
– Sierpinsky,
– Hilbert.
În continuare vom descrie modul de construcție a unor astfel de curbe.

10.2.3. Curba lui Koch

– Curba lui Koch de ordinul 0 este un segment de dreapta (de lungime *l*).

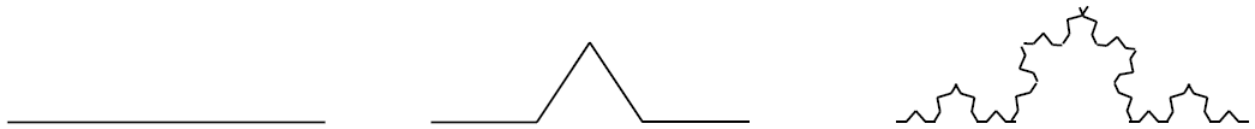
– Curba lui Koch de ordinul *k*+1 se obține împărțind fiecare din segmentele componente ale curbei Koch de ordin *k* in 3 părți egale si înlocuind segmentul din mijloc (fie el *t*) cu celelalte doua laturi ale unui triunghi echilateral care are ca baza pe *t*.

Următoarea funcție recursiva de generare in grafica ”turtle” a curbei lui Koch de ordinul *k*.

În grafica ”turtle” o curba se considera generata prin deplasarea unui creion virtual.

Direcția de deplasare curenta (unghiul *alfa* format de aceasta cu axa *Ox*) poate fi schimbata prin apelurile *stânga*(), *dreapta*() (argumentul fiind exprimat în grade), iar trasarea pe direcția curenta a unui segment de lungime *l* se face cu apelul *draw*().

Poziția curenta a creionului virtual este reținută in variabilele *xc*, *yc*, iar direcția de deplasare este conținută in variabila *alfa* (unghi in radiani).



Curba de ordinul 0 (initiator) **Curba de ordinul 1 (generator)** **Curba de ordinul 3**

Fig. 10.1. Exemple de curbe Koch

10.2.4. Curba lui Peano-Gosper

In cazul curbei Peano-Gosper daca presupunem ca inițiatorul (curba de ordinul 0) are lungimea 1, generatorul (curba de ordinul 1) va cuprinde *n* = 3 segmente, fiecare având o lungime de $(1/7)^{1/2}$ (raportul de similitudine 1/*s*).

Funcția primește ca parametri coordonatele punctelor A si B.

Daca notam cu alfa unghiul CAB (știind ca unghiurile ACD si CBD au 120 grade), obținem:

$$\cos(alfa) = \frac{2{,}5}{\sqrt{7}},$$

$$\sin(alfa) = 0{,}5\sqrt{\frac{3}{7}}.$$

Se vor calcula coordonatele punctelor C, respectiv D (rotații in jurul lui A, respectiv B, cu unghiul alfa, ale unor puncte (M,N) de pe segmentul AB). Curba de ordinul 4 din figura 10.2 are ca inițiator un hexagon regulat.

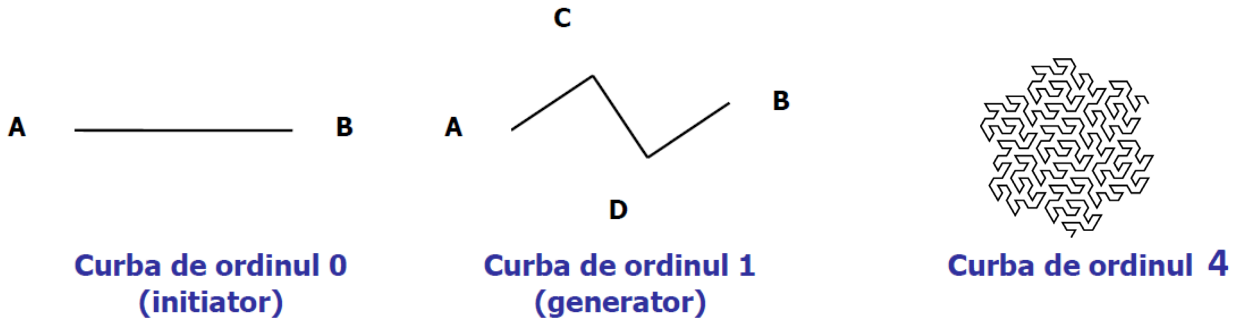


Fig. 10.2. Exemple de curbe Peano-Gosper

10.2.5. Curba dragonului

In cazul curbei dragonului (Knuth) generatorul este construit astfel încât direcția de expandare sa alterneze de o parte si de cealaltă a laturilor poligonului inițiator.

O astfel de curba are proprietatea ca poate umple (pava) o suprafață plană.

Un caz de curba care – ca si curba dragon – acoperă o porțiune de suprafață plana este curba (de tip Peano-Gosper) al cărei generator este format din 7 copii ale inițiatorului, fiecare dintre ele scalate cu raportul $(1/7)^{1/2}$.

Funcția care o generează este dezvoltata folosind o tehnica asemănătoare cu cea utilizata la scrierea funcției Gosper.

Fig. 10.3. Construirea curbei dragonului

10.3. Filtrarea texturilor

În grafica computerizată, filtrarea texturii sau netezirea texturii este metoda utilizată pentru a determina culoarea texturii pentru un pixel mapat textură, utilizând culorile texelurilor din apropiere (pixeli ai texturii). Există două categorii principale de filtrare a texturii, de mărire și de minificare. În funcție de situație, filtrarea texturii este fie un tip de filtru de reconstrucție în care datele rare sunt interpolate pentru a umple golurile (mărire), fie un tip de anti-aliasing (AA), unde probele de textură există la o frecvență mai mare decât cea necesară pentru frecvența eșantionului necesar pentru umplerea texturii (minificare). Filtrarea descrie modul în care se aplică o textură la: forme, dimensiuni, unghiuri și scalări. În funcție de algoritmul de filtrare ales, rezultatul va prezenta diferite grade de neclaritate, detaliu, aliasare spațială, aliasare temporală și blocare. În funcție de circumstanțe, filtrarea poate fi efectuată în software (cum ar fi un pachet de redare software) sau în hardware pentru redare accelerată în timp real sau GPU sau într-un amestec al ambelor. Pentru majoritatea aplicațiilor grafice interactive comune, filtrarea modernă a texturilor este realizată de un hardware dedicat care optimizează accesul la memorie prin memorarea în cache și pre-preluare și implementează o selecție de algoritmi disponibili utilizatorului și dezvoltatorului.

Există mai multe metode de filtrare a texturilor, care fac diferențe între complexitatea calculațională, lățimea de bandă a memoriei și calitatea imaginii.

10.3. Filtrarea texturilor

În timpul procesului de mapare a texturii pentru orice suprafață 3D arbitrară, are loc o căutare a texturii pentru a afla unde pe textură cade fiecare centru de pixeli. Pentru suprafețele poligonale texturate, compuse din triunghiuri tipice pentru majoritatea suprafețelor din jocurile și filmele 3D, fiecare pixel (sau eșantion de pixeli subordonat) al acelei suprafețe va fi asociat cu un anumit triunghi și un set de coordonate barientrice, care sunt utilizate pentru a oferi o poziție în interiorul unei texturi. O astfel de poziție poate să nu se afle perfect pe „grila pixelilor”, necesitând o anumită funcție pentru a explica aceste cazuri. Cu alte cuvinte, deoarece suprafața texturată poate fi la o distanță și orientare arbitrare în raport cu vizualizatorul, un pixel nu corespunde de obicei direct unui singur texel. O anumită formă de filtrare trebuie aplicată pentru a determina cea mai bună culoare pentru pixel. Filtrarea insuficientă sau incorectă va apărea în imagine ca artefacte (erori în imagine), cum ar fi „blocaj”, jaggies sau sclipire.

10.3. Filtrarea texturilor

Pot exista diferite tipuri de corespondență între un pixel și texel / texels pe care îl reprezintă pe ecran. Acestea depind de poziția suprafeței texturate față de vizualizator și sunt necesare diferite forme de filtrare în fiecare caz. Poate fi cazul unei texturi pătrate mapată pe o suprafață pătrată, plasată la o anumită distanță de vizualizare, în acest caz dimensiunea unui pixel de ecran este aceeași ca a unui texel. Alt caz, cînd texelii sunt mai mari decât pixelii ecranului și trebuie măriți în mod corespunzător - un proces cunoscut sub numele de mărire a texturii. Și al treilea caz, cînd fiecare texel este mai mic decât un pixel, astfel un pixel acoperă mai mulți texeli. În acest caz, trebuie aleasă o culoare adecvată pe baza texelilor acoperiți, prin reducerea texturii. API-urile (Application Programming Interface) grafice precum OpenGL permit programatorului să seteze diferite opțiuni pentru filtrele de mărire și minificare. Chiar și în cazul în care pixelii și texelii au exact aceeași dimensiune, un pixel nu se va potrivi neapărat exact cu un texel. Poate fi aliniat greșit sau rotit și poate acoperi părți a patru texeli învecinați. Prin urmare, este nevoie de o filtrare a texturii.

10.3.1. Mipmapping

Mipmapping este o tehnică standard utilizată pentru a salva o parte din activitatea de filtrare necesară în timpul minimizării texturii. De asemenea, este extrem de benefic pentru coerența cache-ului - fără acesta, modelul de acces la memorie în timpul eșantionării de la texturi îndepărtate va prezenta o aparență extrem de slabă, afectând în mod negativ performanța chiar dacă nu se efectuează nici o filtrare. În timpul măririi texturii, numărul de texturi care trebuie căutate pentru orice pixel este întotdeauna de patru sau mai mică; în timpul minificării, totuși, pe măsură ce poligonul texturat se îndepărtează mai mult, întreaga textură ar putea ajunge într-un singur pixel. Acest lucru ar necesita citirea tuturor texturilor sale și combinarea valorilor acestora pentru a determina corect culoarea pixelilor, o operațiune costisitoare.

Mipmapping evită acest lucru prin prefiltrarea texturii și stocarea acesteia în dimensiuni mai mici până la un singur pixel. Pe măsură ce suprafața texturată se îndepărtează mai mult, textura aplicată trece la dimensiunea mai mică prefiltrată. Diferitele dimensiuni ale mipmap-ului sunt denumite „niveluri”, nivelul 0 fiind cea mai mare dimensiune (utilizat cel mai aproape de vizualizator) și nivelurile crescânde utilizate la distanțe crescânde.

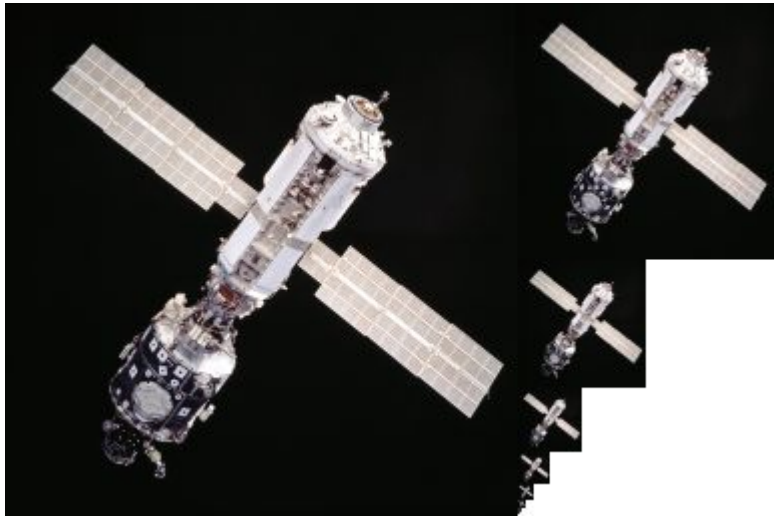


Fig. 10.4. Exemplu de Mipmapping

10.3.2. Metode de filtrare a texturilor

Pentru filtrarea texturiise utilizează : interpolarea celui mai apropiat vecin, filtrarea Bilinear, filtrarea trilineară, filtrarea anizotropică, hărțile MIP.

Interpolarea celui mai apropiat vecin

Interpolarea celui mai apropiat vecin este cea mai simplă și mai brută metodă de filtrare - folosește pur și simplu culoarea texelului cel mai apropiat de centrul pixelului pentru culoarea pixelilor. Deși este simplu, acest lucru are ca rezultat un număr mare de artefacte - „blocarea” texturii în timpul măririi, și aliasing și sclipire în timpul minificării.

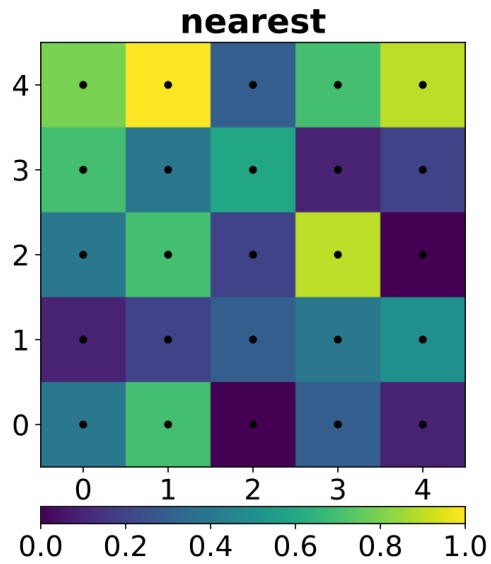


Fig. 10.5. Interpolarea celui mai apropiat vecin

10.3.2. Metode de filtrare a texturilor

Filtrare liniară mipmap

Acceptă eșantionarea celui mai apropiat vecin din mipmap-uri individuale, în timp ce interpolează liniar cele două cele mai apropiate mipmaps relevante pentru eșantion.

Filtrare biliniară

Filtrarea biliniară este următorul pas. În această metodă sunt eșantionate cele mai apropiate patru texturi de centrul pixelului (la cel mai apropiat nivel de mipmap), iar culorile lor sunt combinate cu media ponderată în funcție de distanță.

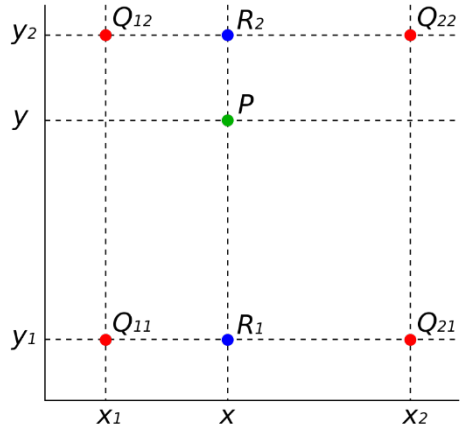


Fig. 10.6. Filtrare biliniară

10.3.2. Metode de filtrare a texturilor

Filtrarea triliniară execută căutarea texturii și filtrarea biliniară pe cele două cele mai apropiate niveluri de mipmap (unul de calitate superioară și unul mai scăzut) și apoi interpolează liniar rezultatele. Acest lucru are ca rezultat o degradare lină a calității texturii pe măsură ce distanța față de vizualizator crește. Mai aproape de nivelul 0 există un singur nivel de mipmap disponibil, iar algoritmul revine la filtrarea biliniară.

10.3.2. Metode de filtrare a texturilor

Filtrarea anisotropică

Filtrarea anizotropă este filtrarea de cea mai înaltă calitate disponibilă pe plăcile grafice 3D pentru consumatori actuali. Altfel vorbind, tehnicile "izotrope" folosesc doar mipmap-uri pătrate care sunt apoi interpolate folosind filtrarea bi- sau triliniară. (Izotropic înseamnă același în toate direcțiile și, prin urmare, este folosit pentru a descrie un sistem în care toate hărțile sunt mai degrabă pătrate decât dreptunghiuri sau alte patrulatere.) Când o suprafață este la un unghi ridicat față de cameră, zona de umplere pentru o textură nu va fi aproximativ pătrată.

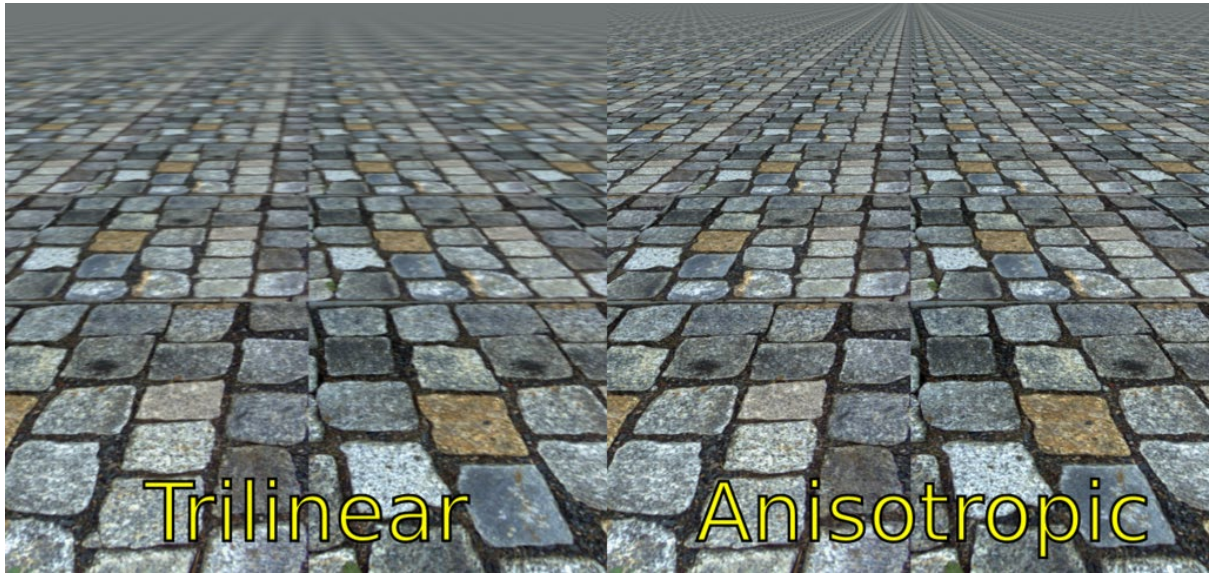


Fig. 10.7. Comparație dintre filtrarea trilineară și anisotropică

ÎNTREBĂRI