

24/11/2013

FACULTATEA
DE
AUTOMATICA SI
CALCULATOARE

SISTEME DE PRELUCRARE
GRAFICA



Tema 3

Shadow Mapping

NOTA: orice nu este specificat este la latitudinea voastra, daca aveti dubii nu ezitati sa folositi forumul!

In aceasta tema va trebui sa implementati in shadere algoritmul shadow mapping (**Atentie!** NU confundati cu shadow volumes).

Algoritmul shadow mapping este unul impartit in 2 etape, rulate impreuna pentru fiecare cadru afisat pe ecran (se mai numesc si „passuri”).

In **prima etapa** se deseneaza scena din **perspectiva luminii cu o proiectie ortografica**. Pentru a putea retine eficient datele se foloseste un framebuffer la care e atasata o textura de adancime. Un exemplu de imagine retinuta in acest buffer este:



Cu cat obiectele sunt mai apropiate (adancime mica, ex: planta) cu atat sunt mai inchise la culoare. Astfel transformarea pentru fiecare pixel este urmatoarea:

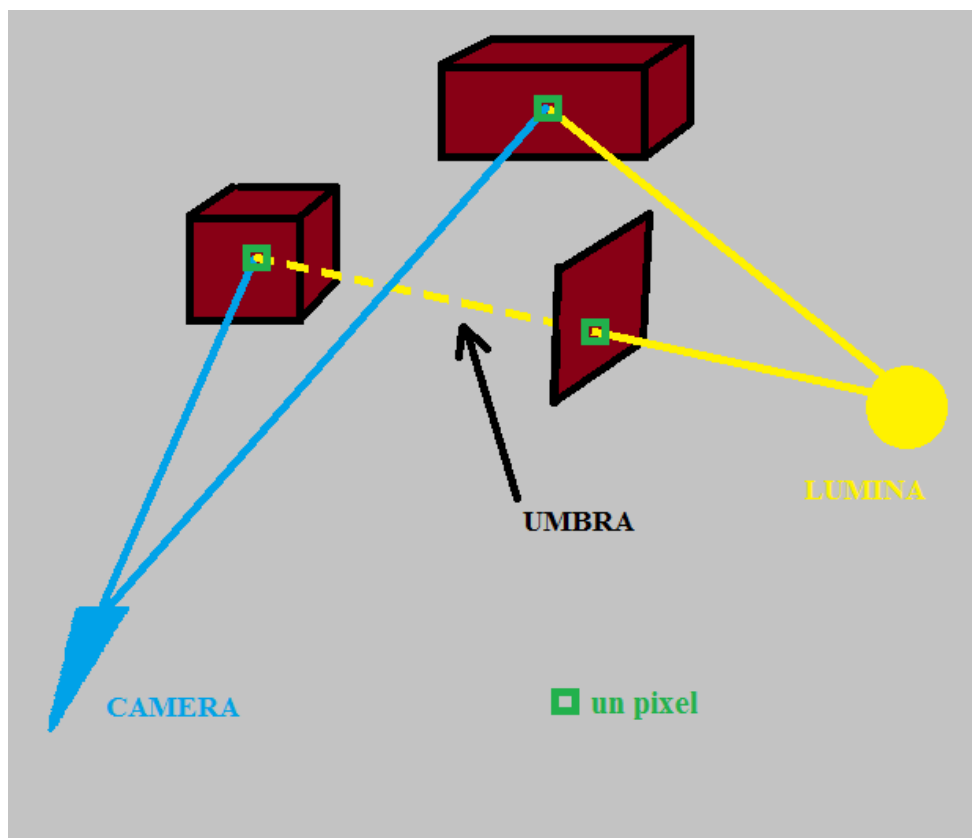
$$Shadowcoord = \begin{pmatrix} 0.5 & 0 & 0 & 0.5 \\ 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \left(\frac{1}{w}\right) * M_{proiectie_{lumina}} * M_{view_{lumina}} * M_{modelare}$$

1/w reprezinta impartirea perspectiva iar prima matrice se numeste matricea de bias si transforma coordonatele din NDClumina intr-un spatiu [0,1]x[0,1]x[0,1] ale carui coordonate pot fi apoi folosite pentru a adresa textura de adancime. **Shadowcoord.z reprezinta adancimea din perspectiva luminii trecuta din [-1,1] (NDC) in [0,1].**

In a **doua etapa** se deseneaza scena din **perspectiva camerei** si pozitia fiecarui fragment de pe ecran este calculata atat din perspectiva camerei (pentru afisare pe ecran) cat si din perspectiva luminii (Shadowcoord-ul fragmentului). **Cum din perspectiva luminii am folosit o proiectie ortografica, indiferent de adancime, ShadowCoord.xy va fi proiectat tot timpul exact pe acelasi texel/pixel.** Deci ne ramane doar ca sa folosim ShadowCoord.xy pe post de coordonata de texturare in harta de adancime de la etapa 1 si sa comparam adancimea din harta cu adancimea pixelului curent. Daca

adancimea din harta e mai mica decat adancimea pixelului curent inseamna ca lumina vede ceva mai apropiat decat pixelul curent, deci pixelul curent nu primeste lumina, deci este in umbra.

O schema simpla: doua fragmente, amandoua vazute atat de camera cat si de lumina, doar ca unul dintre ei este ocludat pentru ca adancimea sa este mai mare decat adancimea primului fragment vazut de lumina pe acel pixel. **Daca valoarea din harta de adancime e mai mica decat valoarea reconstruita => fragmentul e in umbra.**



Cerinte efective, si barem orientativ (din 100)

- Implementarea algoritmului shadow mapping in shadere, capacitate de a explica perfect codul (60)
- Capacitatea de a afisa pe ecran harta de adancime din perspectiva luminii (10)
- Lumina dinamica (5)
- Aspect placut (scena complexa, camera functionala) 10
- Buna intelegere teoretica a algoritmului (15)

Potentiale bonusuri:

- Percentage closer filtering – PCF (10)
- Variance Shadow Maps/ Exponential Shadow Maps (20)
- Algoritmul imbunatatit: Cascaded shadow maps / Light Space Perspective Shadow Maps / Parallel Split Shadow Maps / etc (40)

Responsabil: Lucian Petrescu. **Termen: 8 decembrie 2013, ora 25:55.**

ANEXA 1

Scena cu umbra



Scena fara umbra:



ANEXA 2

Shadow mapping fara filtrare (a se observa trecerea brusca intre lumina si umbra)



Shadow mapping cu PCF (a se observa trecerea graduala intre lumina si umbra).

