Redarea umbrelor în imagini -2

Prof. univ. dr. ing. Florica Moldoveanu

Curs *Elemente de Grafică pe Calculator* – UPB, Automatică şi Calculatoare
2021-2022

SHADOW MAPPING (1)

Williams [WiLL78] a propus o metodă de generare a umbrelor bazată pe rasterizarea de 2 ori a scenei, cu execuţia algoritmului Z-Buffer:

- In prima etapă se construieşte buffer-ul Z al scenei văzută din poziţia sursei de lumina (în transformarea de vizualizare a scenei – funcția lookAt – pozitia observatorului este pozitia sursei).

  Vom nota acest buffer cu ZS. El contine coordonatele z ale fragmentelor care primesc lumina de la sursa. Mai este numit si « **shadow map** », de unde si denumirea sub care este cunoscuta metoda in prezent.

- In etapa a doua se construieşte imaginea văzută din poziţia observatorului, utilizand buffer-ul ZS pentru determinarea punctelor aflate in umbra.

In cazul mai multor surse de lumină, se utilizează câte un buffer ZS pentru fiecare sursă.

Implementarea moderna a metodei este numita “shadow mapping”.

SHADOW MAPPING (2)

Etapa I.

Se calculeaza matricea de vizualizare, ViewS, cu observatorul in pozitia sursei de lumina

Se transmite ViewS la vertex shader (ca variabila uniforma).

Se cere folosirea unui buffer ZS ca buffer de adancime, in locul buffer-ului Z implicit
(In OpenGL, ZS este implementat ca un buffer textura)

Se cere rasterizarea scenei fara scriere in frame-buffer:

1. In vertex shader se transforma varfurile primitivelor folosind ViewS
2. Se rasterizeaza fiecare primitiva a scenei, cu testul de adancime:

pentru fiecare fragment f(x,y,z) rezultat din rasterizarea unei primitive

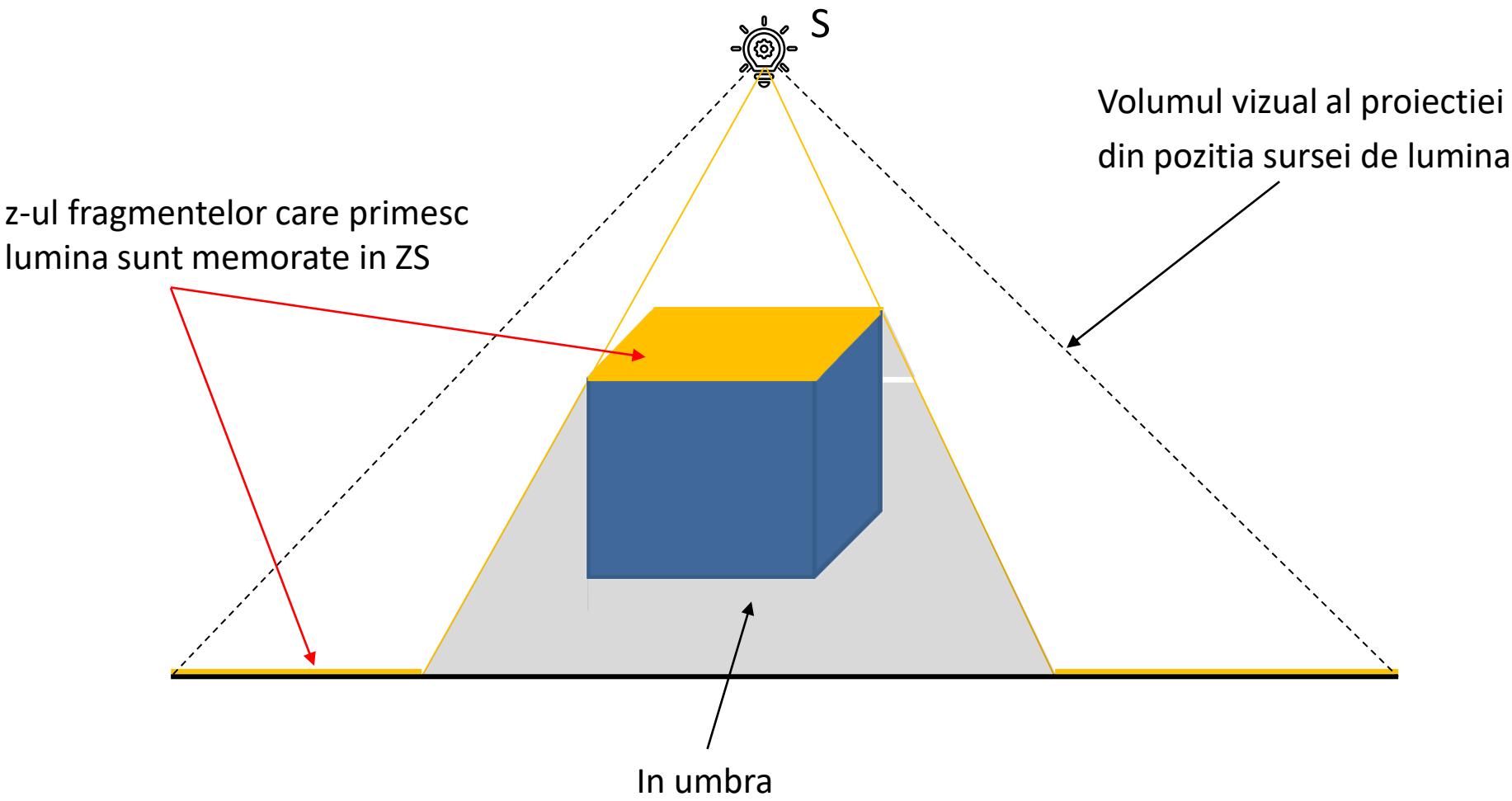
dacă z < ZS[y][x] atunci

ZS[y][x] = z //fragmentul este vizibil în pixelul (x,y) din poziţia sursei

La sfarsitul acestei etape, in ZS sunt memorate coordonatele z ale fragmentelor care primesc lumina de la sursa.

SHADOW MAPPING (3)

Etapa I



SHADOW MAPPING (4)

Etapa II.

Se calculeaza matricea de matricea de vizualizare, ViewO, cu observatorul in pozitia sa.

Se transmit la vertex shader (ca variabile uniforme) ViewO si ViewS.

Se cere rasterizarea scenei:

1. In vertex shader se calculeaza varful transformat cu ViewO (gl_Position) si ViewS (poz_sursa)

poz_sursa - pozitia varfului transformata cu ViewS este variabila de iesire din vertex shader

2. Se rasterizeaza fiecare primitiva a scenei, cu testul de adancime:

pentru fiecare fragment f(x,y,z) rezultat din rasterizarea unei primitive

dacă z < Z-Buffer[y][x] atunci //fragment vizibil în pixelul (x,y), din poziţia observatorului

{Z-Buffer[y][x] = z; frame-Buffer[y][x] = Culoare fragment (întoarsa de fragment shader);}

In fragment shader

poz_sursa - variabilă de intrare pt fragm shader (pozitie fragment in vederea din pozitia sursei)

Culoare_fragment = culoare calculata folosind modelul de iluminare locală;

dacă poz_sursa.z > ZS[poz_sursa.y][poz_sursa.x] atunci //fragmentul nu primeste lumina

Culoare_fragment = factor_umbra* Culoare_fragment; (0< factor_umbra<1)

SHADOW MAPPING (5)

Etapa II.