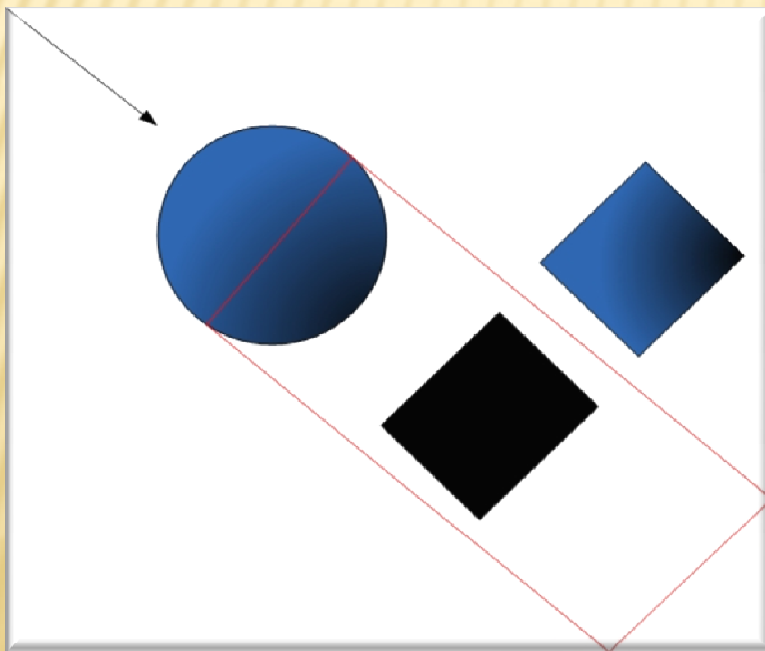


DINAMIČKO OSVIJETLJENJE POMOĆU VOLUMENA SJENA

IDEJA

- Ideja za simuliranje sjena dolazi iz promatranja sjena u prirodi
- Promatrajući osvijetljeni objekt i sjenu koju proizvodi možemo doći do spoznaje da sjena posjeduje nekakav nevidljivi volumen.
- Razlog zbog kojeg sjenu vidimo je njena projekcija na neki drugi objekt ili možemo reći da se objekt (ili dio objekta) nalazi unutar volumena sjene



- Jedna od osnovne problematike simulacije sjena je jednoznačno određivanje takvog volumena...

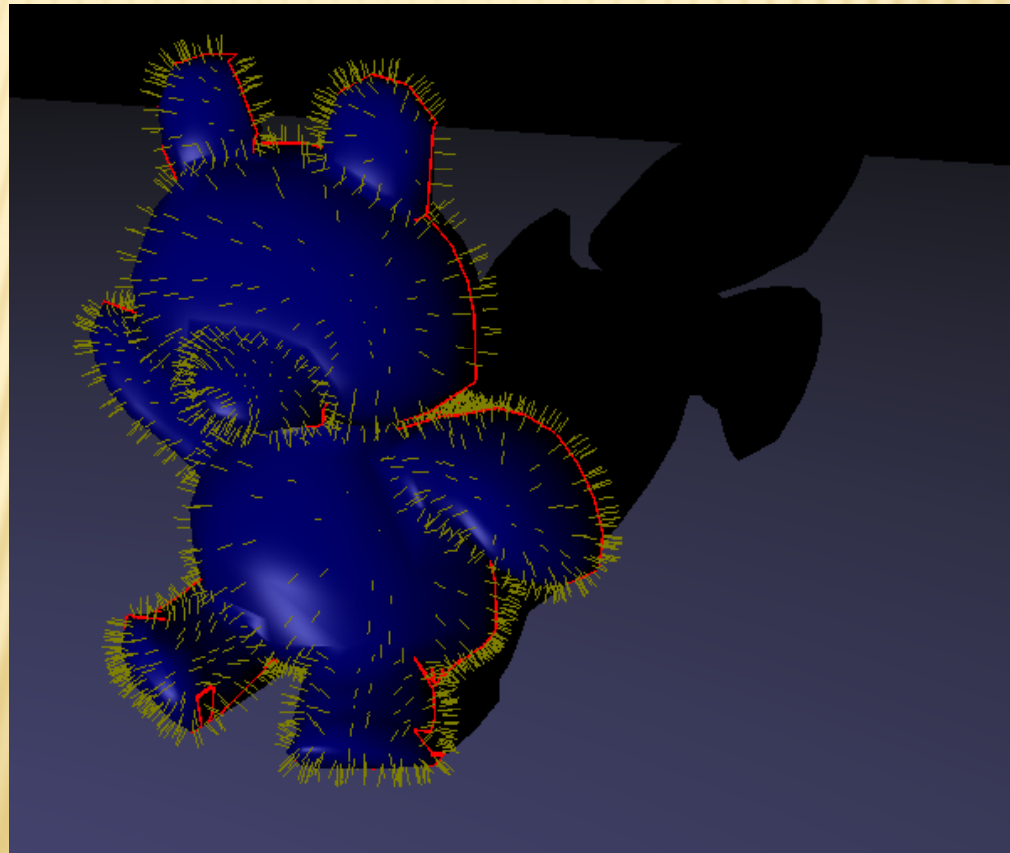
KONSTRUKCIJA VOLUMENA SJENE

Konstrukciju volumena sjene ćemo promatrati na modelima koji su aproksimirani trokutima. Pretpostavljamo da znamo sljedeće informacije o modelu:

- Niz njegovih vrhova (vertex-a)
- Niz indeksa, tj. načine povezivanja vrhova
- Za svaki trokut znamo koji su njegova 3 susjedna trokuta

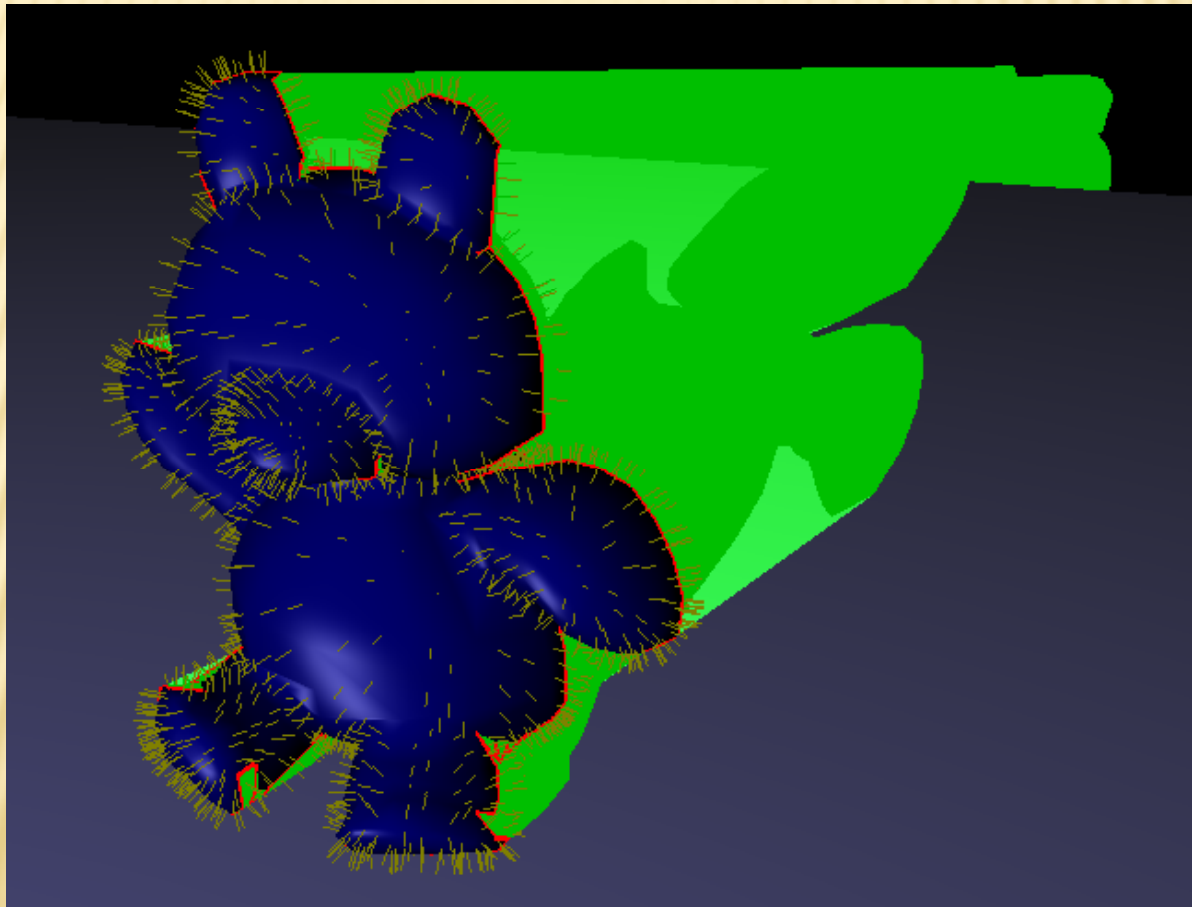
KONSTRUKCIJA VOLUMENA SJENE

Prvi korak pri konstrukciji volumena sjene za neki model je pronaći tzv. bridove siluete s obzirom na neko svjetlo. Bridovi siluete su oni bridovi koji razdjeljuju poligone koji su okrenuti prema svjetlu i njima susjedne poligone koji nisu okrenuti prema svjetlu.



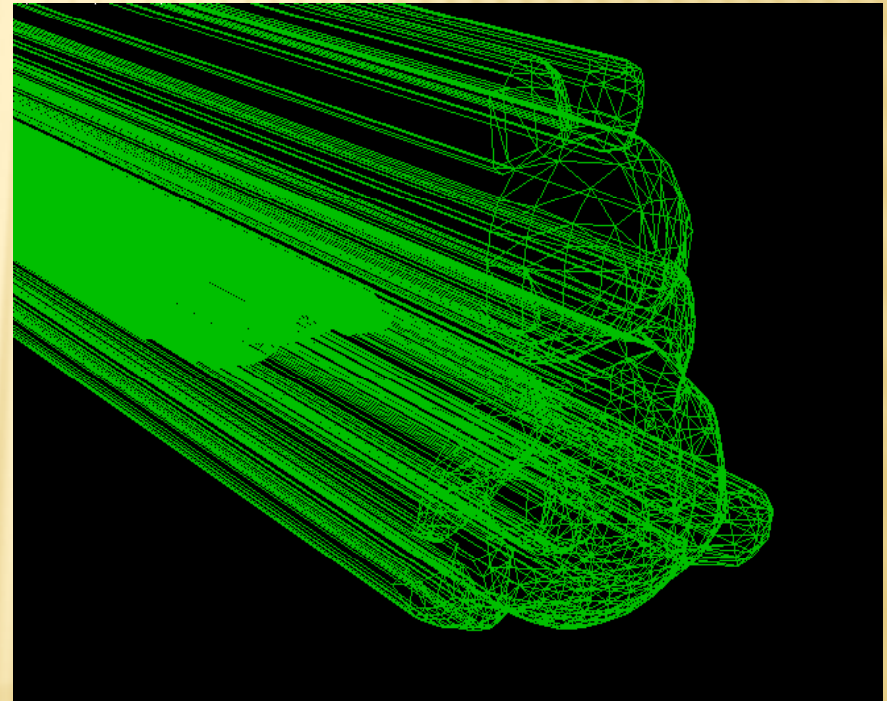
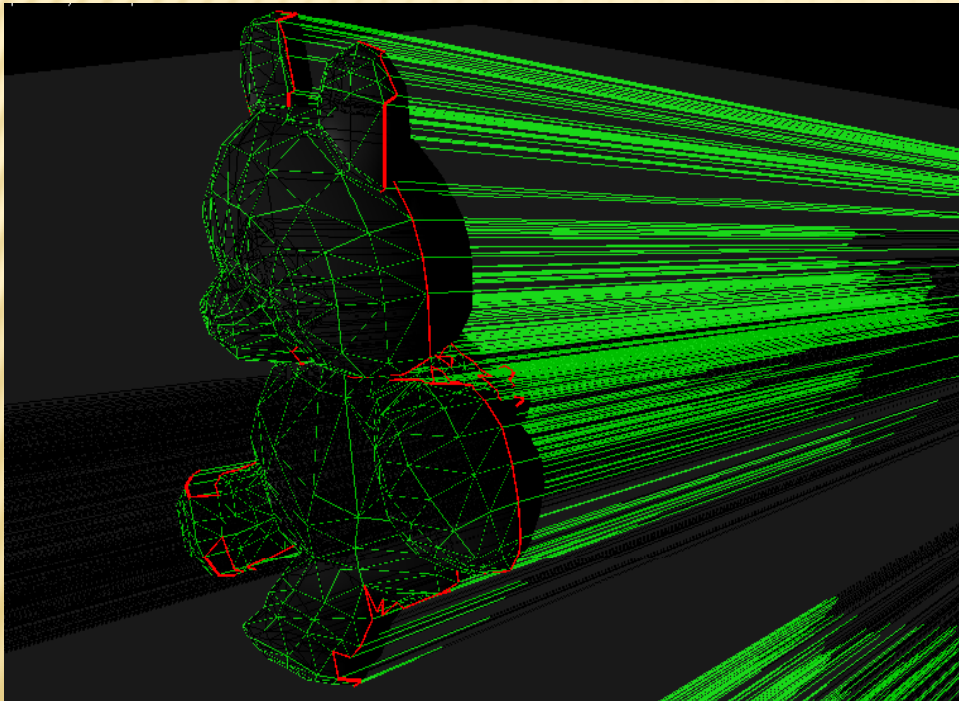
KONSTRUKCIJA VOLUMENA SJENE

Ako znamo bridove siluete i poziciju svjetla, strane volumena sjene možemo izgraditi tako da izdužimo bridove siluete u smjeru svjetla.



KONSTRUKCIJA VOLUMENA SJENE

Kako imamo samo strane volumena, a nama treba volumen, potrebno je dodati tzv. *Light Cap* i *Dark Cap*. *Light Cap* je dio poligona modela koji su okrenuti svijetlu a *Dark Cap* je ostatak.



KONSTRUKCIJA VOLUMENA SJENE

Radit ćemo tri 'spremnika' za geometriju volumena sjene: svijetli, tamni, strane.

ALGORITAM:

```
za (sve trokute modela) {  
  ako (trokut orijentiran prema svijetlu) {  
    dodaj ga u spremnik svijetli  
  
    za (njegova tri susjedna trokuta) {  
      ako (susjedni trokut nije orijentiran od svijetla) {  
        izduži vrhove dijeljenog brida s obzirom na poziciju/smjer svijetla  
        dobijeni četverokut spremi u spremnik strane  
      }  
    }  
  }  
  inače {  
    dodaj ga u spremnik tamni  
  }  
}
```

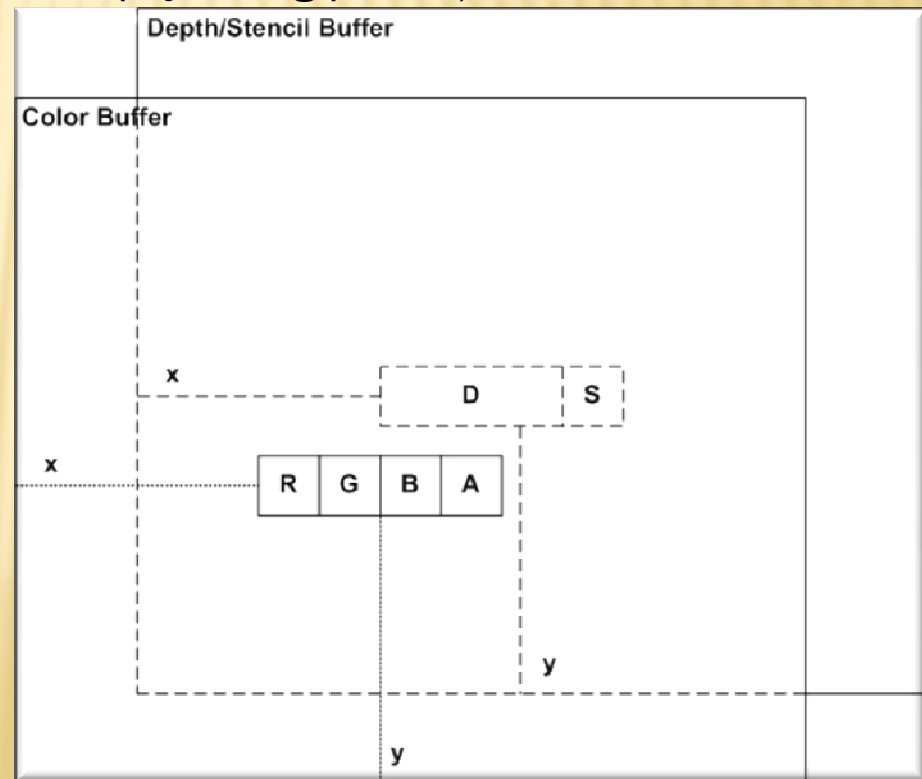
RENDERIRANJE SCENE SA SJENAMA

OpenGL Frame Buffer Ukratko

Zanimaju nas tri buffer-a:

- Color Buffer (informacije o boji pojedinog pixel-a - ono što vidimo)
- Depth Buffer (informacija o z-koordinati pojedinog pixel-a)
- Stencil Buffer (generični buffer)

Svaki pixel koji pripada OpenGL prozoru ima svoje mjesto rezervirano u svim buffer-ima:



RENDERIRANJE SCENE SA SJENAMA

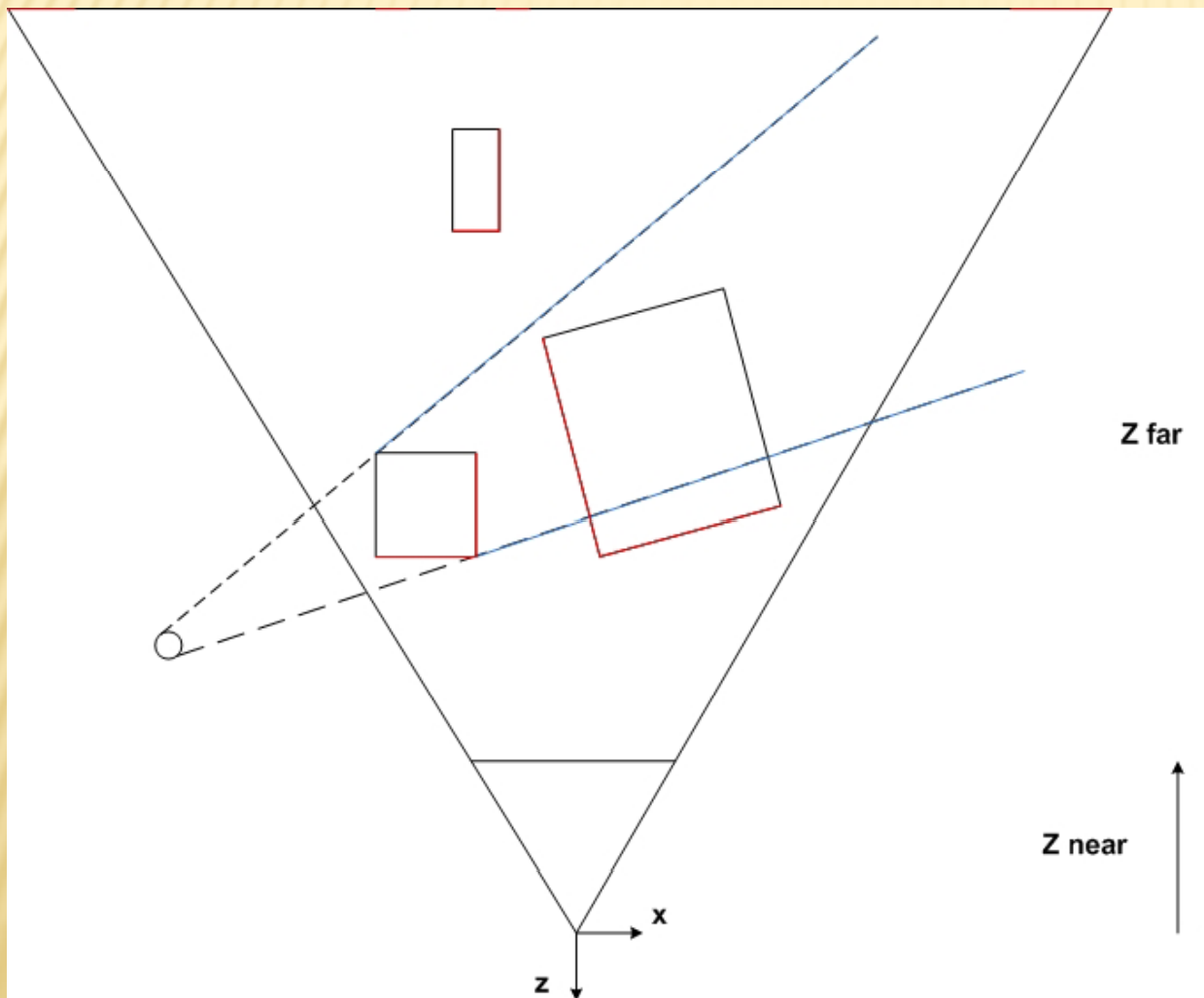
OpenGL Frame Buffer Ukratko

Prije nego je pixel zapisan u frame buffer, on prolazi niz 'testova'. Ako fragment prođe 'testove', postoji mogućnost konfiguriranja na koji način će se boja, alpha, depth kombinirati sa trenutnim vrijednostima u frame buffer-u. Stencil buffer je nešto posebniji od ostalih buffer-a jer se način izmjene stencil vrijednosti može vrlo fleksibilno manipulirati i ako fragment ne prođe sve testove.

Slika: OpenGL State Machine

RENDERIRANJE SCENE SA SJENAMA

Z-Pass Varijanta



Postavimo vrijednosti depth buffer-a
Za sve pixele na maksimalnu
Vrijednost.

Postavimo vrijednosti stencil buffer-a
Za sve pixele na 128.

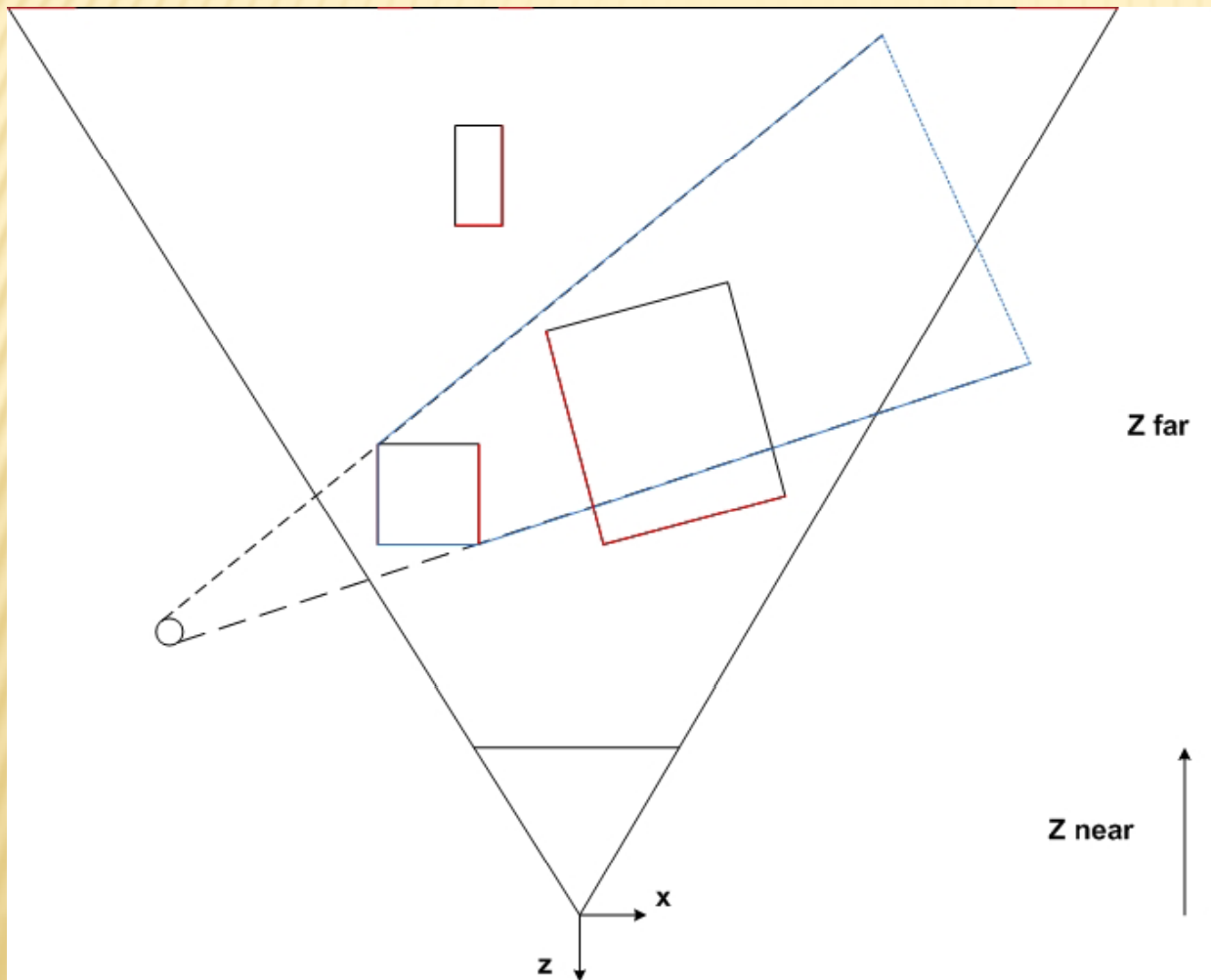
Renderiramo scenu samo s ambijent
Utjecajem svjetla.

Sada u depth buffer-u imamo
Informacije o dubini scene (crveno)

Zabranimo pisanje u color i depth
Buffer, omogućimo pisanje u stencil
Buffer.

RENDERIRANJE SCENE SA SJENAMA

Z-Fail Varijanta



Postavimo vrijednosti depth buffer-a
Za sve pixele na maksimalnu
Vrijednost.

Postavimo vrijednosti stencil buffer-a
Za sve pixele na 128.

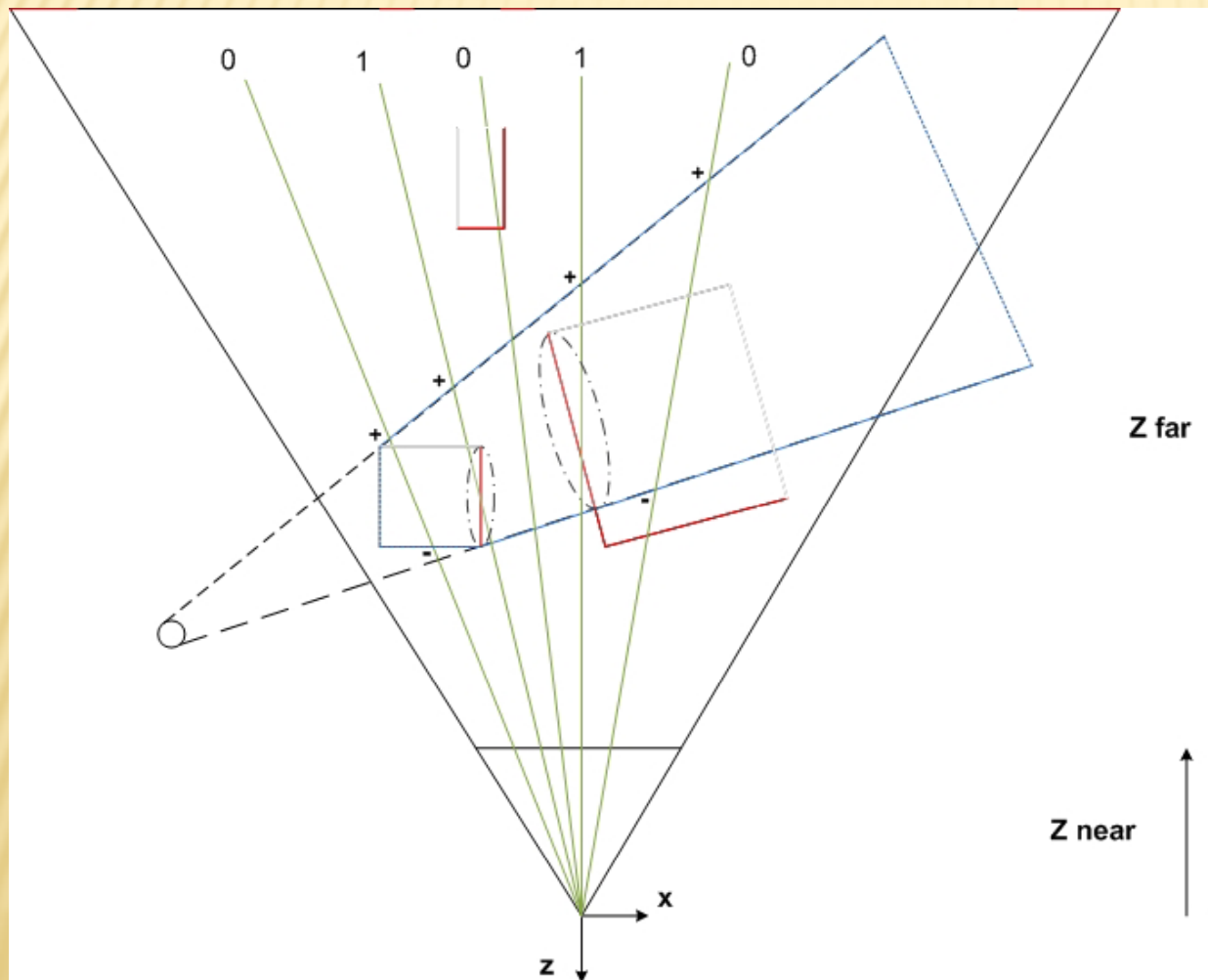
Renderiramo scenu samo s ambijent
Utjecajem svjetla.

Sada u depth buffer-u imamo
Informacije o dubini scene (crveno)

Zabranimo pisanje u color i depth
Buffer, omogućimo pisanje u stencil
Buffer.

RENDERIRANJE SCENE SA SJENAMA

Z-Fail Varijanta



Renderiramo stražnje poligone
Volumena sjene tako da u slučaju
Da je fragment volumena iza
Fragmenta objekta, vrijednost u
Stencil buffer-u uvećamo za 1.

Isto ponovimo za prednje poligone,
Samo što ovaj put smanjimo
Vrijednost u stencil buffer-u za 1.

Uključimo pisanje u color buffer.

Renderiramo scenu sa svjetlima,
Samo za one fragmente za koje je
Stencil vrijednost jednaka 0.

RENDERIRANJE SCENE SA SJENAMA

Z-Fail Varijanta – RIJEŠENJE PROBLEMA SA *FRUSTRUM CULLING*-om

$$\mathbf{P} = \begin{bmatrix} \frac{2 \times \text{Near}}{\text{Right} - \text{Left}} & 0 & \frac{\text{Right} + \text{Left}}{\text{Right} - \text{Left}} & 0 \\ 0 & \frac{2 \times \text{Near}}{\text{Top} - \text{Bottom}} & \frac{\text{Top} + \text{Bottom}}{\text{Top} - \text{Bottom}} & 0 \\ 0 & 0 & \frac{\text{Far} + \text{Near}}{\text{Far} - \text{Near}} & -\frac{2 \times \text{Far} \times \text{Near}}{\text{Far} - \text{Near}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$\lim_{\text{Far} \rightarrow \infty} \mathbf{P} = \mathbf{P}_{\text{inf}} = \begin{bmatrix} \frac{2 \times \text{Near}}{\text{Right} - \text{Left}} & 0 & \frac{\text{Right} + \text{Left}}{\text{Right} - \text{Left}} & 0 \\ 0 & \frac{2 \times \text{Near}}{\text{Top} - \text{Bottom}} & \frac{\text{Top} + \text{Bottom}}{\text{Top} - \text{Bottom}} & 0 \\ 0 & 0 & -1 & -2 \times \text{Near} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

RENDERIRANJE SCENE SA SJENAMA

Z-Fail Varijanta – RIJEŠENJE PROBLEMA SA *FRUSTRUM CULLING*-om

Što se dogodi kad se neki vrh projicira ovakvom matricom?

$$\begin{bmatrix} \frac{2 \times \text{Near}}{\text{Right} - \text{Left}} & 0 & \frac{\text{Right} + \text{Left}}{\text{Right} - \text{Left}} & 0 \\ 0 & \frac{2 \times \text{Near}}{\text{Top} - \text{Bottom}} & \frac{\text{Top} + \text{Bottom}}{\text{Top} - \text{Bottom}} & 0 \\ 0 & 0 & -1 & -2 \times \text{Near} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ -z \\ -z \end{bmatrix}$$

Nakon djeljenja sa homogenom koordinatom, dobijemo da je dubina uvijek 1. Neće doći do *frustrum culling* jer je dubina uvijek 1, tj. u dozvoljenom rasponu (raspon dubina obično [0, 1]).

RENDERIRANJE SCENE SA SJENAMA

Z-Fail Varijanta – ALGORITAM

Postaviti sve vrijednosti depth buffer-a na 1

Omogućiti pisanje u color i depth buffer-a. Zabraniti pisanje u stencil buffer.

```
Za (svaki objekt scene) {  
    Renderirati samo uz ambijent utjecaj svjetla  
}
```

```
Za (svako svjetlo u sceni) {  
    Zabraniti pisanje u color i depth buffer-a. Omogućiti pisanje u stencil buffer.  
    Postaviti sve vrijednosti stencil buffer-a na 128.
```

```
    za (svaki objekt scene koji proizvodi scenu) {  
        Izračunati volumen sjene i renderirati ga određenom tehnikom (z-pass ili z-fail)  
    }
```

Omogućiti pisanje u color buffer.

```
    za (svaki objekt scene) {  
        Aditivno renderirati uz utjecaj osvjetljenja  
    }
```

```
Zabraniti pisanje u color buffer  
}
```

REFERENCE

U google upisati 'shadow volumes'