

1 Forside

2 Problemformulering

3 Indledning

4 Intruduktion

Før jeg gik igang med dette projekt lagde jeg sjælden mærke til skyggeren i et rum. Selv om vi ikke ligger ret meget mærke til skygger ville verden se helt anderledes ud uden. Derfor er det også umuligt at lave en grafisk realistisk gengivelse af virkeligheden på en computer uden at på en eller anden måde at ligge skygger på.

5 Hårde skygger

Beskriver forskellige måder at ligge hårde skygger på en scene. De 2 metoder der i dag bruges til at ligge.

shadow maps stencil shadow volumes

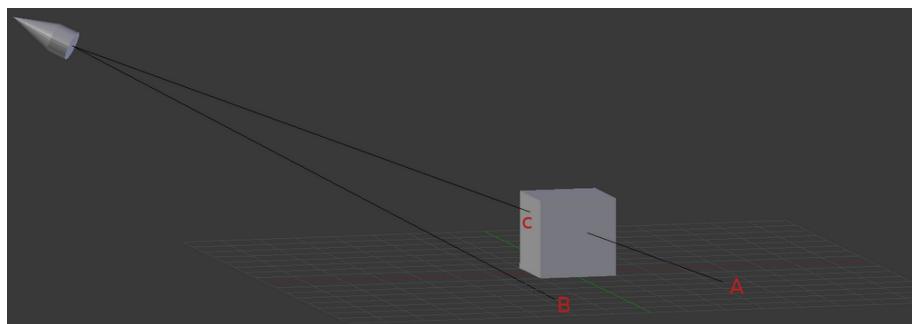
6 Shadowmap

Shadow maps blev indreduceret i 1987 af Lance Williams i artiklen ”Casting curved shadows on curved surfaces”. Siden er teknikken blevet brugt i film og computer grafik til at skyggelægge objekter.

Shadow mapping algoritmen arbejder kun i billedopløsninger og stiller derfor ingen særlige krav scenen anden end at den skal kunne tegnes, dette gør at algoritmen hurtig samt meget fleksibel i forhold til valg af grafiske primitivere.

6.1 Teori

En intuitiv måde at finde ud af om et punkt er i skygge eller ej er forstille sig at man tegner en lige linje fra lyskildes origon til punktet A. Hvis denne linje rammer et eller flere punkter før den når punktet A, vil A være i skygge ellers ikke. Dette kan ses på figur 1.



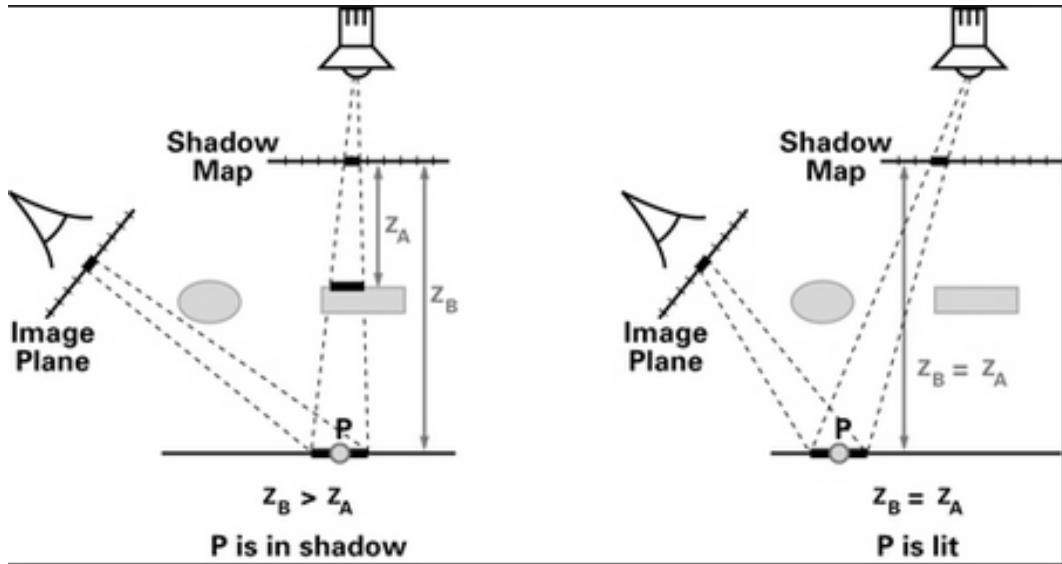
Figur 1: Intuitiv forklaring af skygger

Det er netop denne tanke gang og en dybde buffer som shadow mapping udnytter. At skulle tegne linjer fra lyskilden til hvert punkt vil blive en meget kostelig affære, og det er her dybde bufferen bliver udnyttet. For når en 3D scenen projekteres til 2D billede, ses netop alle de alle punkter der er synlige fra kameraet, og hvis kameraet er i samme punkt som lyset vil disse også være de punkter der er synlige fra lyskilden. (Disse punkter og deres dybder gemmes i en dybde buffer beregnet til dette og kan senere bruges.)

Men dette i tankerne kan to steps algoritmen for shadow mapping nu beskrives således:

1. Scenen renderes set fra lyset men kun dybderne i scenen bliver gemt i dybde bufferen(og det er denne der fungere som shadow mappet).
2. kameraet flyttes tilbage til øjet og scenen renderes. Hvert fragment projekteres over i dybde bufferen og de to z-værdier sammenlignes. Er z-værdien i dybde buffere mindre er fragmentet i skygge.

I step 1 er det kun dybder der er interessante og der kan derfor spares regne kraft på at udlade at rendere farver. Shadow mapping algoritmen er visualiseret på figur 1.



Figur 2: Shadow mapping teknikken

At kamераet bruges for at finde ud af hvad der er lyst op gør at denne metode egner sig godt ved brug af spot light som lys kilde, dette kan dog udvides til andre lyskilder som fx. punkt lyskilder. Denne opgave vil kun arbejde med spot light som lyskilder og i stedet fokusere på teknikker til at forbedre skyggerne der bliver kastet.

6.2 Praktisk

Shadow mapping algoritmen er i teorien er meget simpel, men når algoritmen bruges praktisk kan der opstå mange fejl der gør at algoritmen ikke giver et godt resultat. Disse problemer opstår især fordi dybde kortet er en diskret repræsentation af scene

Et af de første problemer med shadow mapping der skal tages hånd om er clipping planerne i visningstuben (view frustum). Hvis alle objekter ikke ligger inden for "front" og "back" clipping planerne vil disse fejlagtigt komme i skygge selv om de skulle være lyst op. se figur 3. Det er derfor vigtigt at sørge for at alle objekter ligger inden for disse 2 planer for at få et godt resultat.

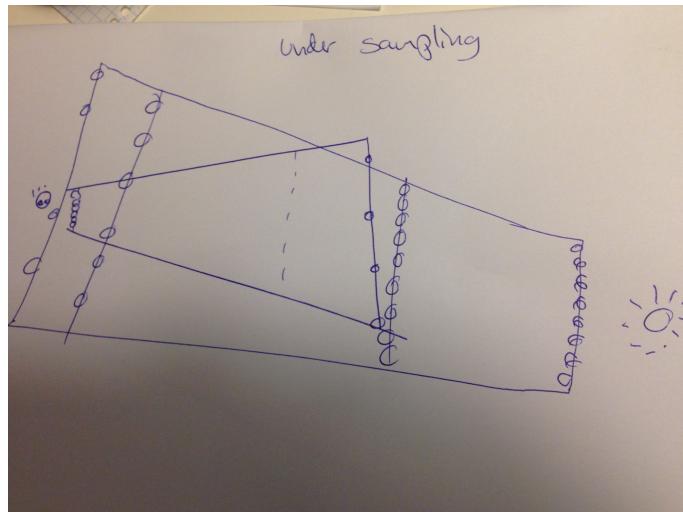
Disse problemer er:

1. Numerisk unøjagtighed ved dybte sammen ligning.
2. Geometrisk unøjagtighed ved projektion.

3. Oversampling
4. Undersampling
5. Front og back clipping plan.

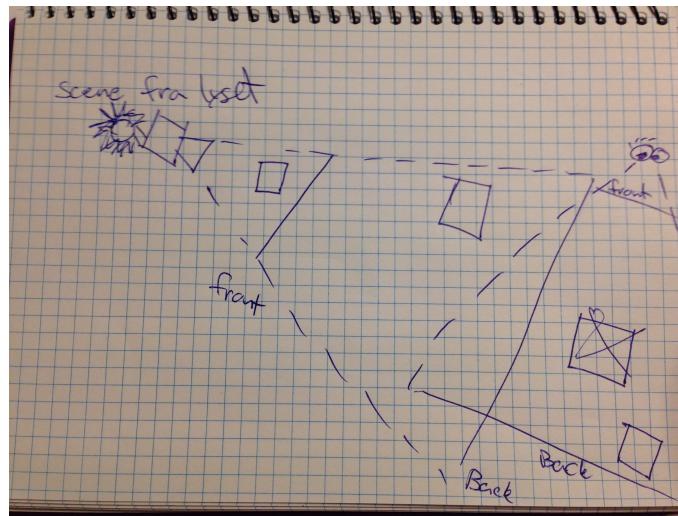
Problem 1 og 2 kan løses ved at indføre en tolerance (bias) ved sammenligning af z-værdierne. Undersampling kan løses forbedres ved at bruge PCF filter sammen med shadow mapping algoritmen.

Undersampling opstår når shadow mappet ikke indeholder tilstrækkelig information. Dette opstår fordi der bliver brugt en perspektiv projktion, hvilket betyder at der er meget information tæt på front clipping planet og meget mindre ude ved back-clipping planet. Hvis kameraet er placeret i den anden enden af scenen vil dette give undersampling da mange pixel tæt på kameraet skal projekteres til meget få pixel i sahdow mappet. Problemet er illustreret i 2D på figur 3



Figur 3: Problem med back-clipping planet ved optegning af shadow mappet.

For at algoritmen skal fungere korret er det vigtigt at alle objekter er inden for back clipping planet når scenen optegnes fra lyse, ellers vil objekterne fejlagtigt komme i skygge selv om lyskilden kaster lys uendeligt langt væk og der ikke er noget der blokere lyset. Dette skyldes at ved opslaget i shadow mappet for objekter der ligger udenfor back-clipping plantet vil denne dybde værdi altid være større end den for shadow mappet. Løsning på problem 5, er at sørge for at back-clipping planet bag det bagerste objekt i scene, dette virker dog kun for afgrænsede scener. En anden løsning er helt at undlade back-clipping planet(Dette skal jeg undersøgt betydningen af). Det samme problem opstår med front-clipping planet og også her er løsningen at dette opsættes så ingen objekter ligger melle dette og kameraet. Problem 5 er illustreret på figur 4.

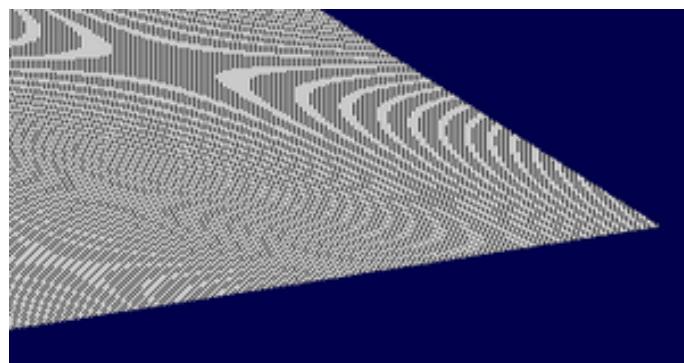


Figur 4: Problem med back-clipping planet ved optegning af shadow mappet.

6.2.1 Shadow acne

//komments Når punkter fra kameraet projekteres ind i dybde kortet kan forskellige problemer opstå. Et dybde buffer eller kort?

Det mest synlige problem med shadow mapping algoritmen er *self-shadowing aliasing* bedre kendt som *shadow acne*, hvor et polygon fejlagtigt kaster skygge på sig selv. Der er to grunde til at dette problem opstår, den første er at decimaler smides væk fordi der dybde buffer indeholder flydende tal. Dybde buffer har et vis antal bit til at repræsentere det flydende tal for z-værdien, og som altid når der regnes flyenden tal kan der opstå afrundings fejl. Denne afrundings fejl kan betyde at når de 2 z-værdier sammenlignes er disse ikke helt ens og polygonet kan derfor fejlagtigt kaste skygge på sig selv. Se figur 5.

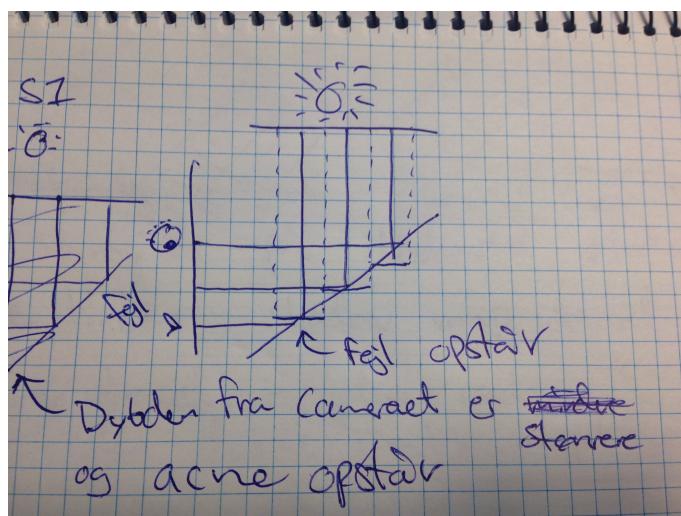


Figur 5: Shadow acne

En af de ting der skyldes shadow acne at der ikke er præcision nok i dybde kortet, denne præcision kan forbedres ved at rykke ”front” og ”back” clipping planerne så de tætter indkapsler alle objekterne i scenen. Dette vil dog ikke kun fjerne shadow acne helt men forbedre resultatet samt forbedre resultatet når andre metoder tages i brug. Denne opgave vil ikke forklare hvordan denne tætning opnås men blot forklare og vise at dette(OVERVEJ OM JEG OGSÅ VIL DET) vil give et bedre resultat.

6.2.2 Biasing

Den anden grund til at *shadow acne* opstår er geometriske fejl. Disse geometriske fejl forekommer fordi dybde værdien for et punkt sample bliver brugt til at beskrive dybden for et areal. Når et punkt fra kameraet projekteres ind i shadow mappet vil disse to punkt sample sjældent være helt det sammen og derfor vil dybde værdierne heller ikke være helt den sammen, dette problem er illustreret på figur 6(a).

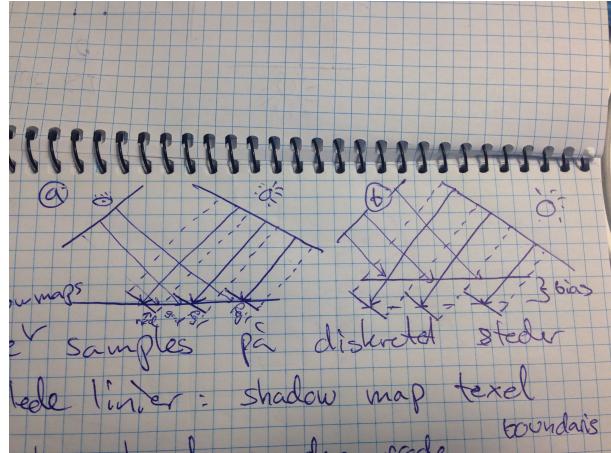


Figur 6: Shadow acne opstår fordi punkt samplet ikke er helt ens set fra lyset og øjet.

Det ses på 6(a) at hvis punkt samplet ikke er helt ens set fra lyset og øjet vil det nogle gange gå godt hvor z-værdien set fra øjet vil være mindre eller lig den set fra lyset mens det andre gang vil opstå selv skygning fordi z-værdien er større set fra øjet. *shadow acne* er så stort problem at der skal tages hånd om det for at shadow mapping algoritmen kan give et acceptabelt resultat.

En måde at håndtere dette problem er at indføre en bias, biasing løser shadow acne problemet ved at sørge for at dybde fra øjet(kameratet) = i dybde i shadowmappet for alle fragtmenter fra det samme polygonum.

Bias er et lille offset der adderes til z-værdien inden dybde testen i opengl foretages og derved bliver alle fragtmenter flyttet lidt tilbage og vil derfor ikke kunne skygge for sig selv se figur 7(b). Dette vil i mange tilfælde løse shadow arne.



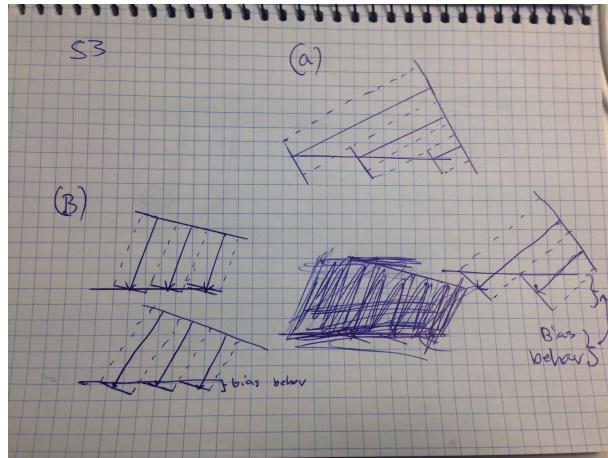
Figur 7: Pile er shadow map samples på diskrete steder. Stippled linier er shadow map texel afgrænsninger. For hver texel er den røde linie den dybde shadow mappet gemmer. Figur (b) ses betydningen af biasen, dybde værdierne fra øjet vil nu altid være større for samme polygon og der opstår derfor ikke geometrisk eller numerisk fejl.

I den simpleste form er biasing en konstant offset værdi der tilføjes til fragmentet, denne værdi er dog ikke så lige til at vælge fordi størrelsen af nødvendige værdi ændre sig i forhold til vinklen mellem projektions planet for lyset og fladen på det polygonie der kastes lys på. Desto mere vinkelret de to planer er på hinanden jo større bias er der behov for, dette ses på figur 8.

Som det ses på figur 8 er det ikke altid at en konstant bias værdi kan løse shadow acne problemet, hertil kan bruges en *slop-scale bias* der har til formål at blive større i takt med vinklen mellem projektions planet for lyset og fladen på det polygonie vokser, derved sørge for at bias værdien altid er stor nok.

hvor r er den mindets værdi garanteret at give en beregnetlig forskel i dybde værdien. m er den største dybde hældning og *factor* og *unit* tilpasser implantationen.

Både konstant bias og slop-scale bias løser løser de to problemer der kan opstå når der foretages projektion ind i shadow mappet (geometrisk og numerisk fejl). Grunde til der ikke bare kan vælge en konstant bias værdi der altid vil være stor nok er at dette vil skabe et nyt problem kendt som Peter panning eller lys lekagese. Problemet opstår når bias værdi bliver for stor og skyggerne derved bliver løsrevet fra objekterne. Dette er et stort



Figur 8: Shadow acne opstår fordi punkt samplet ikke er helt ens set fra lyset og øjet.

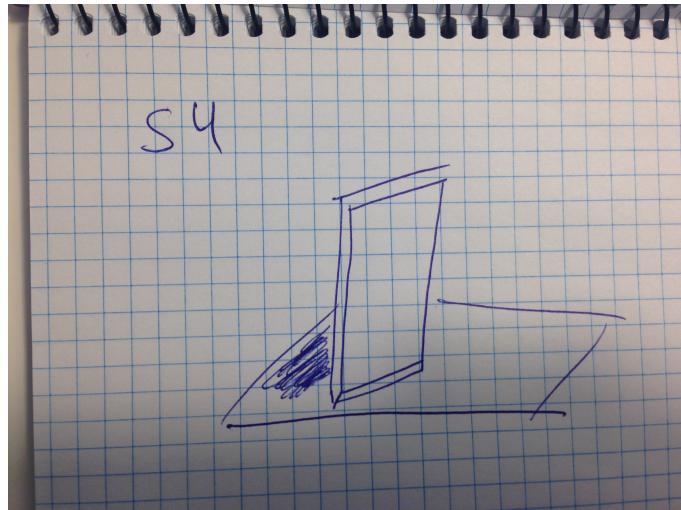
$$offset = m * factor + r * units$$

$$m = \max\left(\frac{aZ}{ax}, \frac{aZ}{aY}\right)$$

Figur 9: hvor r er den mindets værdi garanteret at give en beregnetlig forskel i dybde værdien. m er den største dybde hældning og $factor$ og $unit$ tilpasses implantationen.

problem visuelt fordi objekter kommer til at se ud som om de svæver figur 10.

Derfor bruges en *slop-scale bias* der sørge for der er bias nok til ikke at skabe shadow acne men ikke så meget at der opstår peter panning.



Figur 10: Shadow acne opstår fordi punkt samplet ikke er helt ens set fra lysset og øjet.

6.3 Forbedringer/aliasing

Oversampling er skyld i en del støj der kan

En anden grund til problemet opstår er fordi dybde bufferen ikke er lineær (er den kun for orthogonale projektion), dvs. største delen af præcisionen ligger tæt på ”front clipping” planet. Dette betyder at jo længere væk objekterne er fra lyskilden jo mindre præcise vil skyggerne blive, dette bliver dog kun et problem hvis kameraet er tæt på disse objekter.

Kigger på mulighederne for at forbedre shadow mapping teknikken vha. aliasing:

forskellige metoder: Interpolated shadowing Percentage Closer Filtering (PCF) Variance Shadow Mapping (VSM)

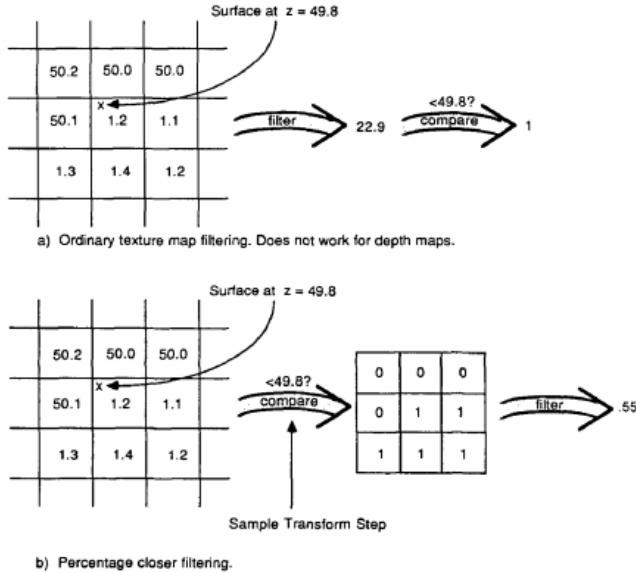
6.3.1 Percentage Closer Filtering (PCF)

Percentage Closer Filtering(herefter referet til som PCF) er en anti-aliasing teknik der løser det problem der opstår ved oversampling. Problemets er: PCF kan også til dels også hjælpe på undersamplings problemet.

PCF teknikken blev udviklet i 1987 af Reeves, Salesin og Cook (referance).

PCF teknikken fungere næsten efter samme princip som ordinære texture maps i opengl, nemlig ved at sample nærliggende værdier i mappet til en middel værdi af disse. Ordinære texture maps i opengl fungere ved at først sample værdierne til en ny samplet værdi og derefter sammenligne denne. Denne tilgang vil ikke virke da der sammen lignes med dybde værdier og derfor vil give en middel dybde. I stedet byttes om på rækkefølgen så der

først sammenlignes nærliggende punkter om disse er i skygge eller ej og derefter tages et gennemsnit af værdierne og bruges som skygge værdien. figur 11 viser forskellen på de to metoder.



Figur 11: Shadow acne opstår fordi punkt samplet ikke er helt ens set fra lysset og øjet.

7 Stencil Shadow Volumes

7.1 Teori

Beskrivelse af Stencil shadow Volumes teknikken. heriblandt fordele og ulemper.

løser problemet med 1-1 fra cam til lys

7.2 Depth-Pass Vs. Depth-Fail

Sammen ligner de 2 metoder Depth-Pass og Depth-Fail for at bestemme om en vertex er i skygge eller ej.

8 Afprøvning

De 2 metoder afprøves, og det vises hvordan de forskellige metoder og filtre påvirker resultaterne.

8.1 Køretider

Køretiderner for de 2 metoder afprøves.

9 Konklusion