

## 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

### 5.1 Forskellige shadow mapping metoder

Der findes rigtig mange måde at forbedre shadow mapping teknikken nogle af forskellige metoder er:

- Interpolated shadowing
- Shadow mapping med Percentage Closer Filtering (PCF)
- Perspective Shadow Mapping
- Cascaded shadow maps (CSMs)

#### 5.1.1 Standart shadow maps

#### 5.1.2 Perspective Shadow maps

Perspective Shadow Maps videre bygger på videre på den orginale shadow mapping ide [?]. Teknikken prøver at løse de alaising problemer der opstår i Shadow mapping, ved at justere den uniforme sampling fordeling i shadow mappet. Forbedring opnås ved at generere shadow mappet i *Post-perspective Space* for kameraet, inden det perspective Shadow map bliver genereret bliver scenen og lyset transformert med kamerates matrix. Dette betyder at objekter tæt på kameraet vil have en højere sampling, til forskel fra shadow mapping hvor samplingen faktoren i forhold til objekter kun afhænger af deres placering i forhold til lyset og ikke kameraet.

## Fordele

- Løser i de fleste tilfælde aliasing problemer med shadow mapping
- Kræver ikke ekstra arbejde ved statiske scener og kamera.

## Ulemper:

- Shadow mappet skal generes hvert gang kameraet flyttes. Dette bevirker at algoritmen kræver en del mere arbejde end shadow mapping
- Objekter bagved kameraet der burde kaste skygge, kaster ikke skygge. Dette kan dog løses men giver andre problemer.
- Større problemer med Shadow acne.
- Sensitive til the dueling frustum problemet

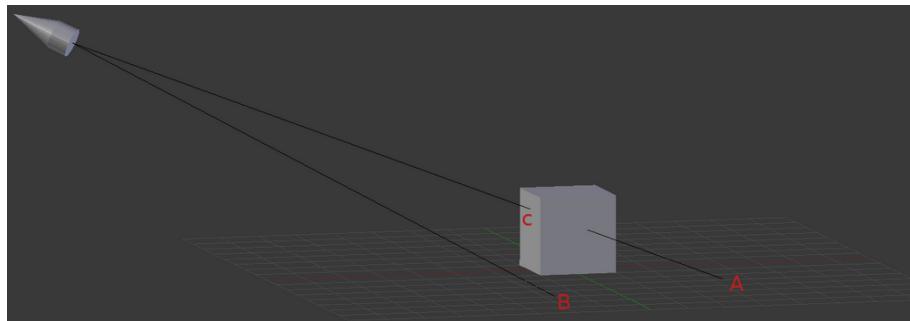
## 6 Shadowmap

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. Algoritmens styrke er dog også dens svaghed, dette skaber en del fejl skygninger og aliasing problemer som vil blive diskuteret i dette afsnit

### 6.1 Teori

En intuitive 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. Figur 1 illustrerer dette princip.



Figur 1: Intuitiv forklaring af skygger. Punktet A ligger i skygge fordi den rette linie fra lyskilden til A går igennem C

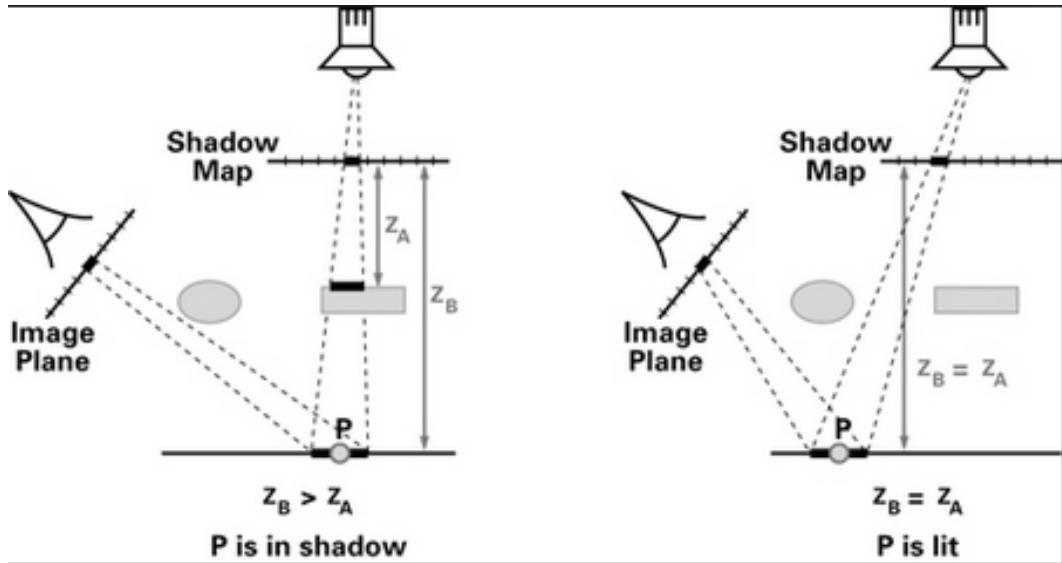
Det er netop denne tanke gang og en dybde buffer som shadow mapping udnytter. At skulle tegne linjer fra lyskilden til hvert punkt vil blive en meget kostelig affære, og det er her dybte bufferen bliver udnyttet. For når en 3D scenen projekteres til 2D billede, ses netop alle de alle punkter der er synlige 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, det er denne der fungere som selve shadow mappet.
2. kameraet flyttes tilbage til øjet og scenen renderes. Hvert fragment

projekteres over i lyset koordinatsystem 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 algorithmen er visualiseret på figur 2.



Figur 2: Shadow mapping teknikken

At kameraet bruges for at finde ud af hvad der er lyst op gør at denne metode egner sig godt ved brug af spotlight som lys kilde, dette kan dog udvides til andre lyskilder som fx. punkt lyskilder ved at tage billeder i alle retninger set fra lyste. Dette giver 6 shadow maps der sættes sammen i en kvadrat og derved få dybden for punkter set i alle retninger. Denne opgave vil kun arbejde med spotlight som lyskilder og i stedet fokusere på teknikker til at forbedre skyggerne der bliver kastet.

## 6.2 Praktisk

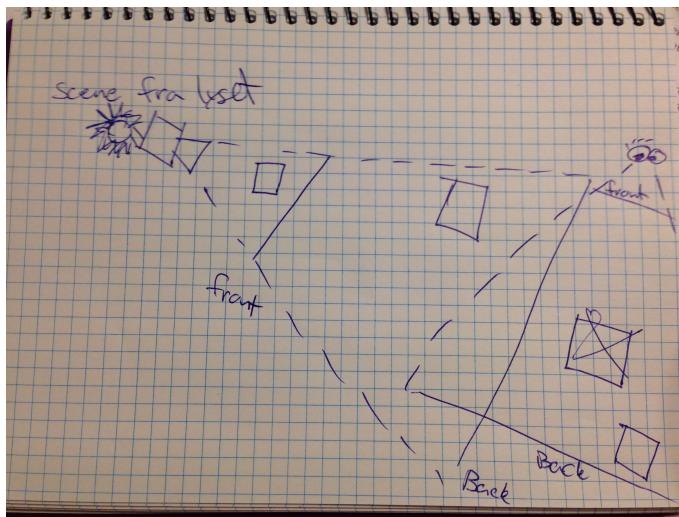
Shadow mapping algoritmen er i teorien er meget simpel, men når algoritmen bruges praktisk kan der opstå mange fejl der gør at algoritmen ikke giver et godt resultat. Disse problemer opstår især fordi shadow mappet er en diskret repræsentation af scene

Disse problemer er:

1. Front og back clipping plan.
2. Numerisk unøjagtighed ved dybte sammen ligning.

3. Geometrisk unøjagtighed ved projktion(opstår bla. pga. undersampling).
4. Oversampling (Opstår når et fragtment i kamera space mappes til mange punkter i shadow mappet)
5. Undersampling (Opstår når mange fragtment i kamera space mappes til et eller få punkter i shadow mappet)

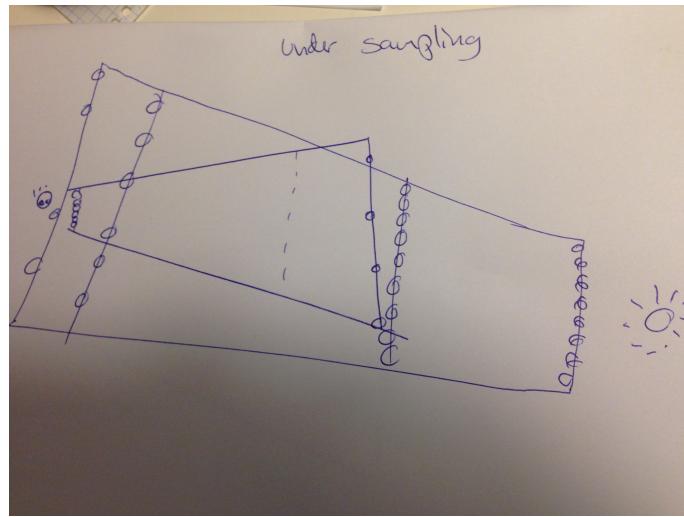
Det første problem med shadow mapping der skal tages hånd om er clipping planerne i visningstuben (view frustum). For at algoritmen skal fungere korret er det vigtigt at alle objekter er inden for back clipping planet når scenen optegnes fra lyse, ellers vil objekterne fejlagtigt komme i skygge selv 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 planet vil denne dybde værdi altid være større end den for shadow mappet. Løsning på problem 1, 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 3.



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

Problem 2 og 3 ses ved at objekter i scenen fejlagtigt kaster skygge på sig selv, fænomenet er også kendt som shadow acne og er beskrevet i afsnitte af samme navn i rapporten. Shadow acne kan løses ved at indføre en tolerance (bias) ved sammenligning af z-værdierne. Hvilket beskrives i Biasing afsnittet.

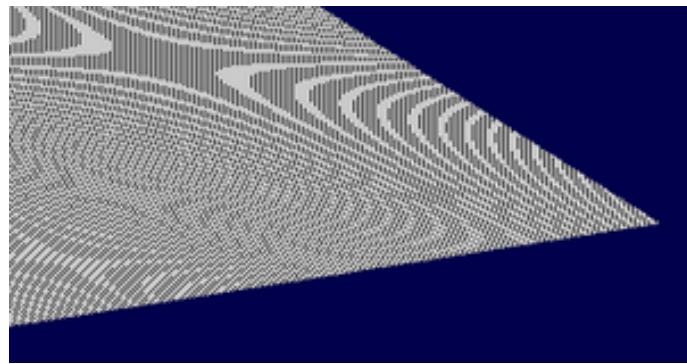
Oversampling opstår når et fragment i kamera space mappes til mange punkter i shadow mappet, dette kan være et problem i realtids programmer da dette skaber unødvendigt meget arbejde, samtidig kan skygger også forekomme rystelser i skyggerne. Undersampling opstår når shadow mappet ikke indeholder tilstrækkelig information, dette er en fordel i realtids programmer, dog kan skygger forekomme meget kantede og nogle steder forkerte. Dette opstår fordi der bliver brugt en perspektiv projktion, hvilket betyder at der er meget information tæt på front clipping plane og meget mindre ude ved back-clipping plane. Hvis kameraet er placeret i den anden ende af scenen vil dette give undersampling da mange pixel tæt på kameraet skal projekteres til meget få pixel i shadow mappet. Problemet er illustreret i 2D på figur 4. Både over og under-samplings problemet vil blive forsøgt forbedret i Percentage Closer Filtering (PCF) afsnittet.



Figur 4: Undersampling. Et meget stort antal pixel fra kameraet skal mappes til kun 3 pixel i shadow mappet.

### 6.2.1 Shadow acne

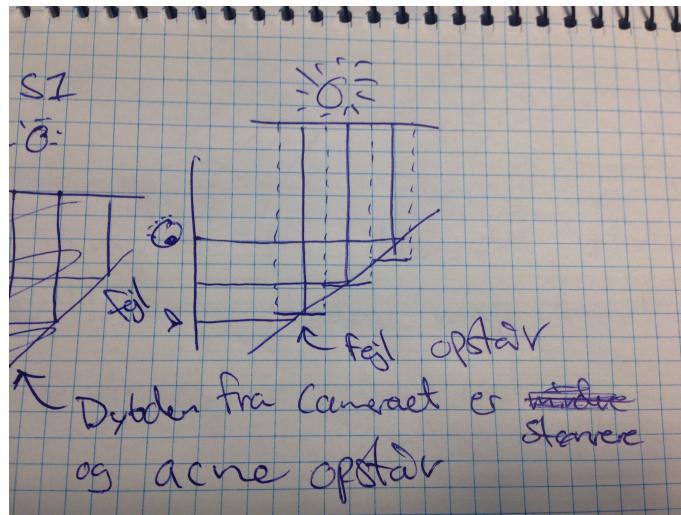
Det mest synlige problem med shadow mapping algoritmen er *self-shadowing aliasing* bedre kendt som *shadow acne*, hvor et polygon fejlagtigt kaster skygge på sig selv. Der er to grunde til at dette problem opstår, den første er at decimaler smides væk fordi der dybde buffer indeholder flydende tal. 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 afrundingsfejl. 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 resultat samt forbedre resultat når andre metoder tages i brug.

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).



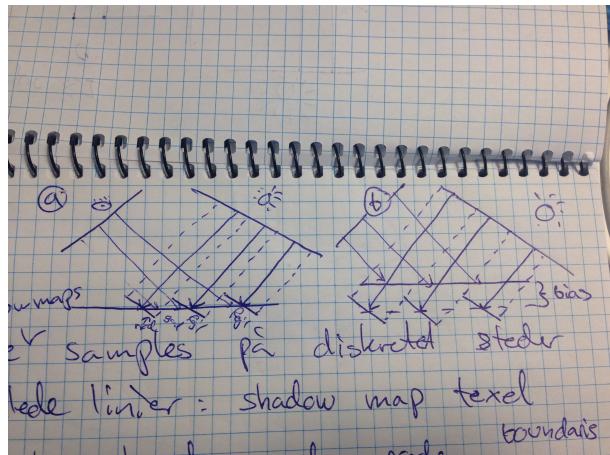
Figur 6: Shadow acne opstår fordi punkt samplet ikke er helt ens set fra lyset og øjet.

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. Løsning på dette problem er at indføre en bias værdi der flytter alle polygonsers z-værdi en lille smule når dybde kortet optegnes. Det næste afsnit vil forklare hvordan denne bias værdi vælges og anvendes.

### 6.2.2 Biasing

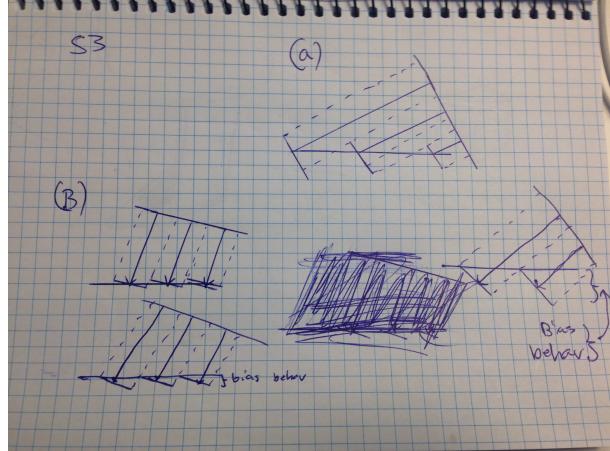
Biasing er en måde at håndtere shadow acne problemet på, biasing løser shadow acne problemet ved at sørge for at dybde fra øjet(kameratet) = i dybde i shadowmappet for alle fragtmenter fra det samme polygonum. Bias er et lille offset der adderes til z-værdien inden dybde testen i opengl foretages og 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 værdi bliver nød til at ændre sige i forhold til vinklen mellem projektionsplanet 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. 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.



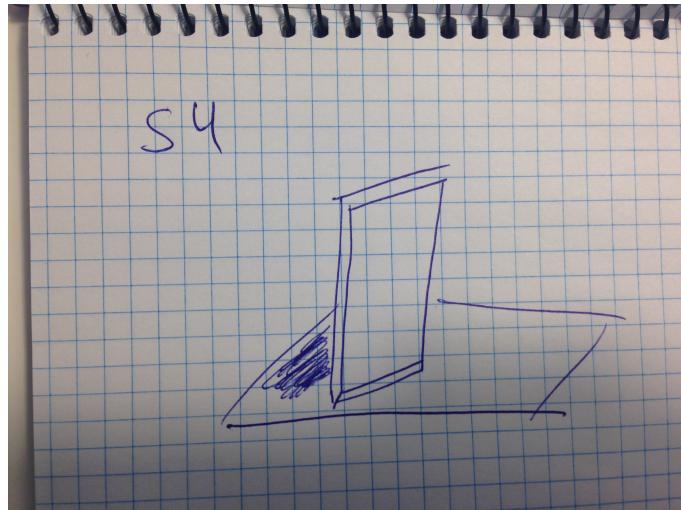
Figur 8: Shadow acne opstår fordi punkt samplet ikke er helt ens set fra lyset og øjet.

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 er at dette vil skabe et nyt problem kendt som Peter panning eller lys lekagelse. Problemet opstår når bias værdi bliver for stor og skyggerne derved bliver løsrevet fra objekterne. Dette er et stort 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.

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

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

Figur 9: Formlen for *slop-scale bias*.  $r$  er den mindets værdi garanteret at give en beregnelig forskel i dybde værdien.  $m$  er den største dybde hældning og  $factor$  og  $unit$  tilpasses implantationen.



Figur 10: Effekt af for stor bias værdi. Dette skaber peter panning, skygger ser ud som om de er løsrevet fra figuren. Dette kan give illusion om at figure svæver selv om dette ikke er tilfældet.

### 6.3 Forbedringer/aliasing

Shadow map er utsat for mange aliasing problemer fordi algoritmen arbejder kun i billedopløsninger. Det kritisk at fragtmenter i kamera space bliver mappet korrekt eller så tæt som muligt på de tilsvarende pixels i shadow mappet, ellers kan skygger forekomme: kantede, jitteret eller (Selv skygge kastning er løst med bias teknikken).

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.

Jeg vil her kort forklare de forskellige metode og vælge en at arbejde videre med.

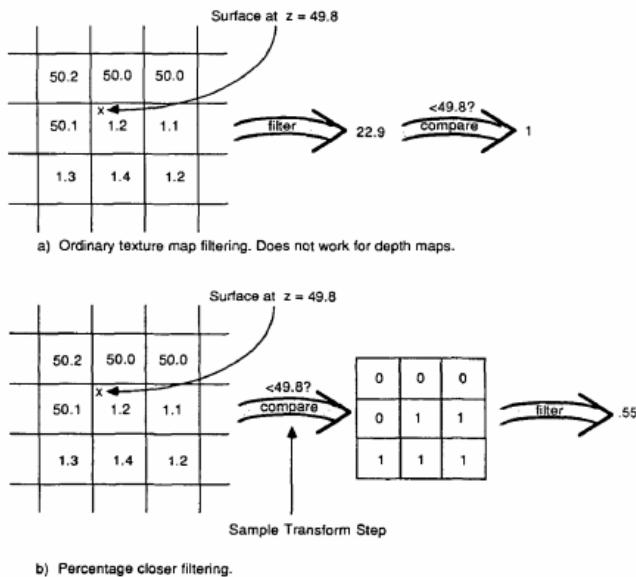
#### 6.3.1 Percentage Closer Filtering (PCF)

Percentage Closer Filtering(herefter referet til som PCF) er en anti-aliasing teknik der løser det problem der opstår ved oversampling. PCF teknikken blev udviklet i 1987 af Reeves, Salesin og Cook (referance). PCF kan også til dels også hjælpe på undersamplings problemet, da den udglatter skyggen og derved vil steder med undersampling ikke fremstå så kantede som uden PCF.

PCF kan også tildels bruges til at simulere penumbra (bløde skygger) men da PCF ikke tager højde for afstande mellem lyskilden og objekt er dette

ikke en penumbra algoritme. Algoritme vil dog alligevel i mange tilfælde virke overbevisende som penumbra.

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: (a) ordinære texture maps i opengl. (b) PCF metoden brugt på dybde kortet.

På figur 11(b) ses hvordan PCF filteret først sammenligner dybde værdien med alle punkterne og bestemmer om punktet er i skygge eller ej, herefter tages et gennemsnit af de binære resultater hvilket bliver shadow værdien og herved får en skygger der bliver udglattet og ikke binær.

Det næste er at bestemme det område i shadow mappet der skal filtres over. Artiklen [RSC84] forslå to forskellige metoder at bestemme dette sampling område på.

1. Der anvendes en fast størrelse på omegnen der medtages af nærliggende punkter i shadow mappet
2. Nærliggende punkter på polygonet projekteres over i shadow mappet og disse punkter bruges til at filtrere over

Den første metode er hurtig og nem at implementere fordi der ikke skal lave nogle ekstra projektioner ind, men kun laves et antal ekstra opslag i shadow mappet følgende det størrelsen af det område der ønske. Problemet er dog at de pixel der ligger i område der filteres over ikke har nogen relation til den pixel der skal skygge farves.

Metode to kræver en del mere arbejde fordi udover punkterne skal vælges skal disse også projekteres over i shadow mappet før der kan sammenlignes og filtres over disse. Metoden burde tilgengæld i teorien give et bedre resultat da alle punkterne der filteres over ret faktisk ligger på polygonet. Denne metode har artiklen [RSC84] dog ikke testet.

FORKLAR HVILKEN METODE JEG VIL BRUGE.

## 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

Køretidernes for de 2 metoder afprøves.

## 9 Konklusion

## Litteratur

- [RSC84] Rendering Antialiased Shadows with Depth Maps. William T. Reeves , David H. Salesin, Robert L. Cook *Computer Graphics, Volume 21*, 283-291, 1987.
- [LW78] CASTING CURVED SHADOWS ON CURVED SURFACES Lance Williams *Computer Graphics Lab New York Institute of Technology* , 1978.