



Coherente Schaduwmappen

door

Xavier TALPE

Promotor: Prof. dr. ir. Ph. Dutré

Proefschrift ingediend tot het
 behalen van de graad van
 Master in de informatica

Academiejaar 2009-2010

© Copyright by K.U.Leuven

Zonder voorafgaande schriftelijke toestemming van zowel de promotor(en) als de auteur(s) is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen tot of informatie i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, wendt u tot het Departement Computerwetenschappen, Celestijnlaan 200A, 3001 Leuven, (016) 32 77 00 of via email: info@cs.kuleuven.be.

Voorafgaande schriftelijke toestemming van de promotor(en) is eveneens vereist voor het aanwenden van de in dit afstudeerwerk beschreven (originele) methoden, producten en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

Faculteit Wetenschappen
Departement Computerwetenschappen
Celestijnenslaan 200A
3001 Leuven
(016) 32 77 00

K.U. Leuven
Academiejaar 2009–2010

Voornaam en naam student : Xavier TALPE

Titel :

Coherente Schaduwassen

Engelse vertaling :

Coherent Shadow Maps

Korte inhoud :

Schaduwen vormen een essentieel onderdeel bij het visualiseren van virtuele scènes. Ze helpen bij het beter begrijpen van de vorm en de relatieve positie van objecten. Voor het berekenen van schaduwen in interactieve toepassingen wordt zeer vaak gebruik gemaakt van schaduwassen. Schaduwassen zijn echter beperkt tot het weergeven van harde, onrealistische schaduwen. Om realistische schaduwen te visualiseren kunnen de resultaten van meerdere schaduwassen worden gecombineerd. Een mogelijke manier om dit te doen is met behulp van coherente schaduwassen. Daarbij worden op voorhand schaduwassen geconstrueerd van een specifiek object in de scène. Deze schaduwassen worden aangemaakt vanop verscheidene posities rondom dit object. Tijdens het visualiseren van een scène kunnen deze schaduwassen worden geraadpleegd om de schaduwen van dit object te berekenen. Door de coherentie tussen deze schaduwassen uit te buiten is een compacte en snel toegankelijke representatie mogelijk. Daardoor kunnen coherente schaduwassen worden gebruikt om realistische schaduwen weer te geven voor interactieve toepassingen.

In dit eindwerk wordt bestudeerd hoe nauwkeurig de schaduwen zijn die met coherente schaduwassen worden berekend. Daarbij wordt de invloed van verscheidene parameters op de kwaliteit van de schaduwen onderzocht. Tevens wordt onderzocht of een meer compacte voorstelling van de schaduwassen mogelijk is door de coherentie tussen deze schaduwassen nog verder uit te buiten.

Proefschrift ingediend tot het behalen van de graad van
Master in de informatica

Promotor : Prof. dr. ir. Ph. Dutré

Assessoren : Prof. dr. B. Demoen
Dr. B. Brown

Begeleider : ir. J. Laurijssen

Dankwoord

Dit eindwerk is het resultaat van een jaar noeste arbeid. Zonder de steun en hulp van vele mensen zou dit eindwerk niet tot een goed einde zijn gebracht. Dit is dan ook het ideale moment om enkele mensen in het bijzonder te bedanken.

In de eerste plaats wens ik mijn ouders te bedanken. Zij boden mij vijf jaar geleden de kans om aan deze studies te beginnen. Zonder hun steun zou ik deze nooit tot een goed einde hebben gebracht.

Bijzondere dank gaat uit naar mijn begeleider Jurgen Laurijssen. Zijn ervaring en kennis waren een grote meerwaarde bij het maken van dit eindwerk. De vele contactmomenten gedurende het jaar waren steeds een verrijking en motiveerden me elke keer opnieuw om het beste van mezelf te geven. Ik wens hem dan ook alle succes toe met zijn aankomende doctoraatsverdediging.

Daarnaast wens ik ook mijn promotor Prof. dr. ir. Ph. Dutré te bedanken om mijn interesse in de wereld van computergrafieken op te wekken. Ik ben hem dankbaar voor de kans die hij mij heeft gegeven om een eindwerk te maken in dit vakgebied.

Tot slot wens ik ook mijn vriendin Lieselot te bedanken, voor de steun en motivatie die je me de voorbije jaren hebt gegeven. Ook op de moeilijkere momenten stond je steeds voor me klaar.

Inhoudsopgave

1 Inleiding	1
1.1 Situering	1
1.2 Doelstellingen	2
1.3 Overzicht	2
2 Achtergrond	3
2.1 Terminologie	3
2.1.1 Geometrie	3
2.1.2 Licht	4
2.1.3 Lichtbronnen	5
2.1.4 Materiaaleigenschappen	6
2.1.5 Materialen	8
2.2 Grafische Pijplijn	9
2.2.1 Rasterisatie	9
2.2.2 Grafische Verwerkingsseenheid	11
2.2.3 Ray Tracing	11
2.3 Camerasysteem	12
2.3.1 Homogene Coördinaten	12
2.3.2 Modeltransformatie	12
2.3.3 Weergavetransformatie	13
2.3.4 Orthografische Projectie	13
2.3.5 Perspectief-projectie	14
2.3.6 Canoniek Weergavevolume	16
2.4 Ruimtevullende Curves	17
3 Schaduwen	20
3.1 Schaduwmallen	21
3.1.1 Dieptemallen	21
3.1.2 Visibiliteitstest	23
3.2 Problemen	24
3.2.1 Vouwvervorming	25
3.2.2 Zelfbeschaduwing	26
3.3 Conclusie	28

4 Coherente Schaduwmannen	29
4.1 Introductie	30
4.2 Berekenen van de Visibiliteitsinformatie	31
4.2.1 Sferische Coördinaten	31
4.2.2 Bemonstering	32
4.2.3 Uniforme Verdeling	32
4.2.4 Conclusie	33
4.3 Visualisatie	34
4.4 Resultaten	36
4.4.1 Percentage Closer Filtering	39
4.5 Conclusie	42
5 Compressie	43
5.1 Introductie	44
5.1.1 Foutloze Compressie	44
5.1.2 Duale Dieptemappen	45
5.2 Per-Pixel Compressie	46
5.3 Resultaten Per-Pixel Compressie	50
5.3.1 Invloed van Geometrie	51
5.3.2 Invloed van Ruimtevullende Curves	53
5.3.3 Invloed van Resolutie en Aantal Dieptemappen	56
5.3.4 Invloed van Uniforme Positiebepaling	56
5.3.5 Invloed van Oriëntatie	59
5.3.6 Coherentiemappen	59
5.3.7 Conclusie	61
5.4 Per-Map Compressie	62
5.5 K-Means Clustering	65
5.6 Compressie aan de hand van een Hilbert Curve	67
5.6.1 Driedimensionale Hilbert Compressie	67
5.6.2 Vierdimensionale Hilbert Compressie	68
5.7 Wavelet Compressie	70
5.8 Conclusie	73
6 Kwaliteitsstudie	75
6.1 Opstelling	75
6.2 Invloed van Geometrie	76
6.3 Invloed van Resolutie en Aantal Dieptemappen	77
6.4 Invloed van Uniforme Positiebepaling	80
6.5 Invloed van Oriëntatie	80
6.6 Conclusie	83

INHOUDSOPGAVE	viii
7 Besluit	85
7.1 Overzicht	85
7.1.1 Compressie	85
7.1.2 Kwaliteit	86
7.2 Toekomstig Werk	86
A Resultaten Per-Pixel Compressie	87
B Resultaten Kwaliteitsstudie	90
Bibliografie	95
Lijst van Figuren	103
Lijst van Tabellen	105

Hoofdstuk 1

Inleiding

1.1 Situering

Computergrafieken heeft als doelstelling het visualiseren van virtuele scènes. Met behulp van wiskundige modellen kan onder andere de geometrie en belichting van een scène worden beschreven. Computergrafieken kent vele toepassingen. Zowel in de entertainmentindustrie als in het domein van medische beeldvorming en architectuur worden computergrafieken veelvuldig gebruikt.

Schaduwen vormen een essentieel onderdeel bij het visualiseren van virtuele scènes. Zonder schaduwen zien scènes er vaak onrealistisch en vlak uit. Daarnaast helpen schaduwen in het beter begrijpen van de geometrische structuur van een scène.

Een veelgebruikte techniek om schaduwen te berekenen zijn schaduwassen. Daarbij wordt een scène eerst vanuit het standpunt van de lichtbron gevisualiseerd en daarna vanuit het gewenste camerastandpunt. Enkel de punten die vanuit het standpunt van de lichtbron zichtbaar zijn, worden belicht. Punten die niet zichtbaar zijn bevinden zich dan in de schaduw van een ander object.

Schaduwassen zijn echter beperkt tot het weergeven van harde schaduwen. Een punt ligt dan ofwel volledig in schaduw ofwel volledig buiten schaduw. Dit leidt tot een abrupte overgang tussen punten in schaduw en punten daarbuiten. In realiteit komen harde schaduwen echter niet voor aangezien elke lichtbron over een oppervlak beschikt. Hierdoor kan een punt zichtbaar zijn voor bijvoorbeeld 80% van het lichtoppervlak en dus voor slechts 20% in schaduw liggen. Dit worden zachte schaduwen genoemd.

Om zachte schaduwen te visualiseren kunnen bijvoorbeeld meerdere schaduwassen worden geconstrueerd op het oppervlak van een lichtbron. Voor elk punt in de scène wordt dan nagegaan vanuit welke posities op de lichtbron dit punt zichtbaar is. Het gemiddelde resultaat geeft een indicatie van het percentage aan invallend licht dat wordt geblokkeerd.

Voor interactieve toepassingen (met dynamische scènes) is een dergelijke aanpak echter niet haalbaar. Telkens een object of lichtbron van positie wijzigt, dienen de schaduw-mappen opnieuw te worden gegenereerd. Hierdoor wordt het visualisatieproces sterk vertraagd. Om realistische schaduwen te visualiseren, zijn bovendien ook veel verschillende schaduw-mappen nodig. Deze schaduw-mappen nemen veel ruimte in beslag. Om deze problemen te vermijden kunnen de schaduw-mappen vooraf worden berekend, om daarna te worden gecomprimeerd. Dit is het idee achter coherente schaduw-mappen.

Bij coherente schaduw-mappen worden op voorhand schaduw-mappen geconstrueerd van een specifiek object in de scène. Deze schaduw-mappen worden geconstrueerd op allerhande posities rondom dit object. Tijdens het visualiseren van een scène kunnen deze schaduw-mappen worden geraadpleegd om de schaduwen van dit object te berekenen. Door de coherentie tussen deze schaduw-mappen uit te buiten is een compacte en snel toegankelijke representatie mogelijk. Hierdoor kunnen realistische schaduwen worden weergegeven voor interactieve toepassingen.

1.2 Doelstellingen

In dit eindwerk wordt de techniek van coherente schaduw-mappen onderzocht. Er wordt bestudeerd hoe nauwkeurig de schaduwen zijn die met deze techniek worden berekend. Daarbij wordt onderzocht wat de invloed is van verscheidene parameters op de kwaliteit van de schaduwen, zoals het aantal of de resolutie van deze schaduw-mappen.

Om de schaduw-mappen op een compacte en foutloze manier voor te stellen, wordt de coherentie tussen de verschillende schaduw-mappen uitgebuit. Er wordt onderzocht wat de invloed is van verschillende parameters op de efficiëntie van deze compressietechniek. Daarnaast wordt onderzocht of een meer compacte voorstelling van de schaduw-mappen mogelijk is.

1.3 Overzicht

In hoofdstuk 2 wordt een algemene introductie gegeven tot het domein van computergrafieken. Daarbij worden de verschillende onderdelen van een scène besproken en wordt uitgelegd hoe een scène wordt gevisualiseerd. In hoofdstuk 3 wordt uitgelegd hoe schaduwen ontstaan en hoe ze kunnen worden berekend met behulp van schaduw-mappen. Hoofdstuk 4 beschrijft de techniek van coherente schaduw-mappen. Er wordt uitgelegd wat coherente schaduw-mappen zijn, hoe ze worden geconstrueerd en op welke manier ze worden gebruikt. In hoofdstuk 5 wordt onderzocht hoe schaduw-mappen op een compacte en foutloze manier kunnen worden opgeslagen. In hoofdstuk 6 wordt de kwaliteit van coherente schaduw-mappen nagegaan. In hoofdstuk 7 ten slotte wordt een overzicht gegeven van het geleverde werk samen met enkele bedenkingen bij de bekomen resultaten.

Hoofdstuk 2

Achtergrond

Een virtuele scène is opgebouwd uit vier onderdelen: geometrie, materialen, lichtbronnen en een camera. De geometrie beschrijft de algemene vorm van objecten in een scène. Aan de hand van materialen wordt voor elk object beschreven hoe het invallend licht wordt gereflecteerd. Om dit licht te simuleren worden lichtbronnen gebruikt. De camera ten slotte bepaalt hoe en welk deel van een virtuele scène wordt gevisualiseerd. Om deze onderdelen te beschrijven wordt telkens gebruik gemaakt van wiskundige modellen.

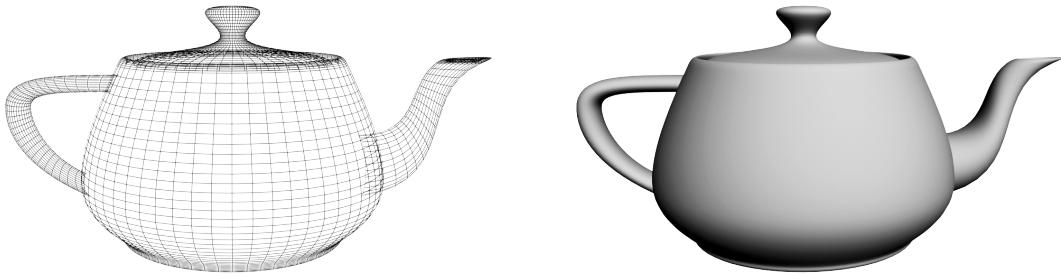
In sectie 2.1 wordt een inleiding gegeven tot het domein van computergrafieken. Verschillende begrippen met betrekking tot geometrie, materialen en lichtbronnen worden verklaard. In sectie 2.2 wordt uitgelegd welke stappen nodig zijn om een virtuele scène om te zetten naar een tweedimensionaal beeld. Dit wordt de grafische pijplijn genoemd. Het camerasysteem dat daarbij wordt gebruikt, wordt toegelicht in sectie 2.3. Ten slotte worden in sectie 2.4 ruimtevullende curves besproken. Deze curves worden gebruikt om een coherente volgorde te bepalen voor punten in een ruimte.

2.1 Terminologie

Eerst wordt uitgelegd hoe de geometrie van een object wordt beschreven. Daarna worden de fysische eigenschappen van licht behandeld samen met enkele modellen om lichtbronnen te simuleren. Tot slot worden materialen en hun reflectie-eigenschappen besproken.

2.1.1 Geometrie

Een driedimensionaal model is opgebouwd uit vertices en veelhoeken. Deze beschrijven het oppervlak van een object. Hoe meer vertices en veelhoeken worden gebruikt, hoe nauwkeuriger en zachter verlopend het oppervlak wordt. Figuur 2.1 toont een voorbeeld van een driedimensionaal model.



Figuur 2.1: Een driedimensionaal model van de theepot van Utah. Het volledige model is opgebouwd uit vierhoeken (links) en kan worden gevisualiseerd (rechts).

2.1.2 Licht

Licht kan worden gedefinieerd als elektromagnetische straling die bestaat uit verschillende golven en hun golflengte λ . Van het volledige spectrum is het menselijk oog slechts gevoelig voor het gebied met een golflengte tussen 380 en 800nm. Dit gebied komt overeen met het Rood-Groen-Blauw (RGB) kleurenspectrum [Shi05].

Om de belichting in een scène te berekenen, is het interessant om de fysische eigenschappen van licht beter te begrijpen. Daarom wordt gebruik gemaakt van radiometrische grootheden. Het is aan de hand van deze grootheden dat de interactie van licht met materialen wordt beschreven [DBB06, Suf07].

Radiometrische Grootheden

Stralingsenergie Q vormt de basiseenheid voor elektromagnetische energie. De totale bijdrage van alle fotonen vormt de stralingsenergie. De eenheid ervan is de joule [J].

Stralingsflux Φ is de hoeveelheid stralingsenergie per eenheid van tijd:

$$\Phi = \frac{dQ}{dt}.$$

Stralingsflux wordt ook wel **vermogen** genoemd. De eenheid is de watt [$W = J/s$].

Irradiantie E en **exitantie** M stellen de stralingsflux voor die respectievelijk op een oppervlak invalt en een oppervlak verlaat, per eenheid van oppervlak. Exitantie wordt ook wel **radiositeit** B genoemd.

$$E = \frac{d\Phi_i}{dA} \quad , \quad M = B = \frac{d\Phi_o}{dA}.$$

De eenheid voor beide is watt per vierkante meter [W/m^2].

Stralingsintensiteit I is stralingsflux per eenheid van ruimtehoek:

$$\Phi = \frac{d\Phi}{d\omega}.$$

De eenheid is watt per vierkante meter per sterradiaal [$W/(m^2 sr)$].

De ruimtehoek is een maat voor oppervlakte op een (hemi)sfeer. De oppervlakte die wordt gemeten is de projectie van een object op de (hemi)sfeer rond een punt. De ruimtehoek beschrijft dus de grootte van een object gezien vanuit een punt. Een klein nabijgelegen object kan bijvoorbeeld dezelfde ruimtehoek hebben als een groot object dat veraf gelegen is. De ruimtehoek wordt gemeten in sterradielen [sr].

Radiantie L is stralingsflux per eenheid van loodrecht oppervlak en per eenheid van ruimtehoek:

$$L(x, \Theta) = \frac{d^2\Phi}{d\omega dA^\perp} = \frac{d^2\Phi}{d\omega dA \cos \theta}.$$

Radiantie meet de stralingsflux voor een punt x afkomstig uit de richting Θ per eenheid van loodrecht oppervlak en per eenheid van ruimtehoek. De eenheid is dezelfde als bij stralingsintensiteit, namelijk watt per vierkante meter per sterradiaal [$W/(m^2 sr)$].

Radiantie heeft twee interessante eigenschappen. De eerste eigenschap is dat radiantie invariant is langs een recht pad: $L(x \rightarrow y) = L(y \leftarrow x)$. De radiantie die vertrekt uit het punt x in de richting van het punt y , is dus gelijk aan de radiantie in het punt y afkomstig van het punt x . Hierbij wordt verondersteld dat er geen participerende media (gas, smog) aanwezig is die het licht kunnen verstrooien. Het licht beweegt dus door een vacuüm. De tweede eigenschap is dat het menselijke oog gevoelig is voor radiantie. De combinatie van beide eigenschappen heeft als gevolg dat de radiantie van een punt op een oppervlak overeenstemt met de radiantie die op ons oog invalt. Dit maakt radiantie een belangrijke grootheid binnen computergrafieken. Het is deze grootheid die we voor elk punt in een scène wensen te berekenen.

2.1.3 Lichtbronnen

Om licht in een scène te simuleren, worden lichtbronnen gebruikt. Hieronder worden de meest gebruikte modellen besproken. Figuur 2.2 illustreert de eigenschappen van elk model.

Directionele Lichtbron

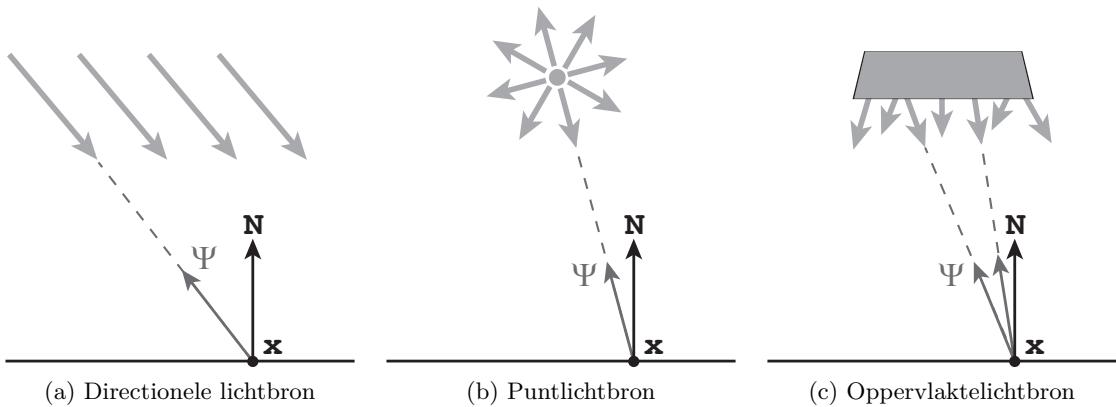
Een directionele lichtbron bevindt zich op een oneindige afstand. Daardoor worden de invallende stralen als parallel beschouwd. De lichtintensiteit is tevens constant voor alle punten in de scène. Directionele lichtbronnen worden meestal gebruikt om de zon te simuleren.

Puntlichtbron

Een puntlichtbron bevindt zich op een specifieke positie in de ruimte. Het licht wordt in alle richtingen even sterk uitgestraald. Aangezien een puntlichtbron zich op een welbepaalde positie bevindt, wordt elk punt vanuit een andere richting belicht. Hierdoor is ook de lichtintensiteit verschillend voor elk punt in de scène.

Oppervlaktelichtbron

Een oppervlaktelichtbron beschrijft een oppervlak dat licht uitstraalt. Het uitgestraalde licht is uniform verdeeld over dit oppervlak. Elk punt in de scène wordt door het volledige oppervlak (en dus vanuit verschillende posities) belicht. Oppervlaktelichtbronnen geven tevens aanleiding tot zachte schaduwen. Zachte schaduwen worden besproken in hoofdstuk 3.



Figuur 2.2: Vaak gebruikte modellen om lichtbronnen te simuleren.

2.1.4 Materiaaleigenschappen

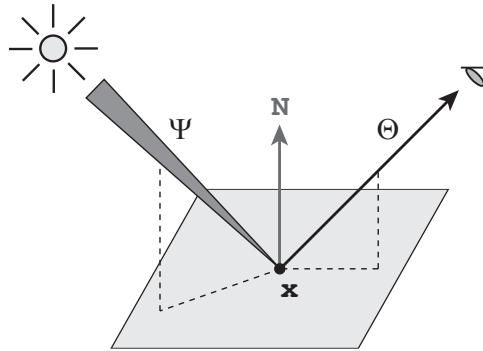
Materialen beschrijven de interactie van licht met een oppervlak. Hiervoor wordt de Bidirectionele Reflectantie Distributie Functie (BRDF) gebruikt. De BRDF is een vier-dimensionale functie die beschrijft in welke mate het invallend licht wordt gereflecteerd. Meer formeel is het de verhouding tussen de gereflecteerde radiantie vanuit het punt x in de richting Θ ($L_r(x \rightarrow \Theta)$) en de invallende irradiantie in datzelfde punt x afkomstig uit de richting Ψ ($E(x \leftarrow \Psi)$) [DBB06]. Zie figuur 2.3 voor een grafische voorstelling.

De BRDF f_r kan als volgt worden geconstrueerd:

$$\begin{aligned} f_r(x, \Psi \rightarrow \Theta) &= \frac{dL_r(x \rightarrow \Theta)}{dE(x \leftarrow \Psi)} \\ &= \frac{dL_r(x \rightarrow \Theta)}{L(x \leftarrow \Psi) \cos(N, \Psi) d\omega_\Psi}, \end{aligned}$$

met N de normaal in het punt x . Daaruit kan tevens de gereflecteerde radiantie L_r worden afgeleid:

$$\begin{aligned} dL_r(x \rightarrow \Theta) &= f_r(x, \Psi \rightarrow \Theta)L(x \leftarrow \Psi) \cos(N, \Psi)d\omega_\Psi \\ L_r(x \rightarrow \Theta) &= \int_{\Omega_x} f_r(x, \Psi \rightarrow \Theta)L(x \leftarrow \Psi) \cos(N, \Psi)d\omega_\Psi. \end{aligned}$$



Figuur 2.3: De Bidirectionele Reflectantie Distributie Functie $f_r(x, \Psi \rightarrow \Theta)$ beschrijft de interactie van licht met een materiaal in het punt x . Licht afkomstig uit richting Ψ wordt gereflecteerd in richting Θ .

Vervolgens kan de *rendering vergelijking* worden opgesteld. De rendering vergelijking beschrijft voor elk punt x en elke richting Θ wat de uitgaande radiantie $L(x \rightarrow \Theta)$ is:

$$\begin{aligned} L(x \rightarrow \Theta) &= L_e(x \rightarrow \Theta) + L_r(x \rightarrow \Theta) \\ L(x \rightarrow \Theta) &= L_e(x \rightarrow \Theta) + \int_{\Omega_x} f_r(x, \Psi \rightarrow \Theta)L(x \leftarrow \Psi) \cos(N, \Psi)d\omega_\Psi. \end{aligned}$$

Hierbij staat $L_e(x \rightarrow \Theta)$ voor de zelf-uitgestraalde radiantie. Deze term is enkel positief voor lichtbronnen. Voor andere objecten daarentegen is deze term nul. De radiantie L speelt een belangrijke rol. Het is immers deze term die voor elk punt in een scène wordt berekend.

In zijn huidige vorm omvat de rendering vergelijking zowel directe als indirecte belichting. Men spreekt over directe belichting als het licht, afkomstig van een lichtbron, rechtstreeks op een oppervlak invalt. Bij indirecte belichting is het licht niet rechtstreeks afkomstig van een lichtbron, maar van het licht dat door andere oppervlakken wordt gereflecteerd. In deze thesis wordt abstractie gemaakt van de indirecte belichting in een scène.

De rendering vergelijking kan ook worden geschreven in oppervlaktevorm. Hierbij wordt geïntegreerd over alle oppervlakken in de scène. In het geval van directe belichting worden enkel de oppervlakken van de lichtbronnen in beschouwing genomen:

$$L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_A f_r(x, \Psi \rightarrow \Theta)L(y \leftarrow -\Psi)V(x, y)G(x, y)dA_y.$$

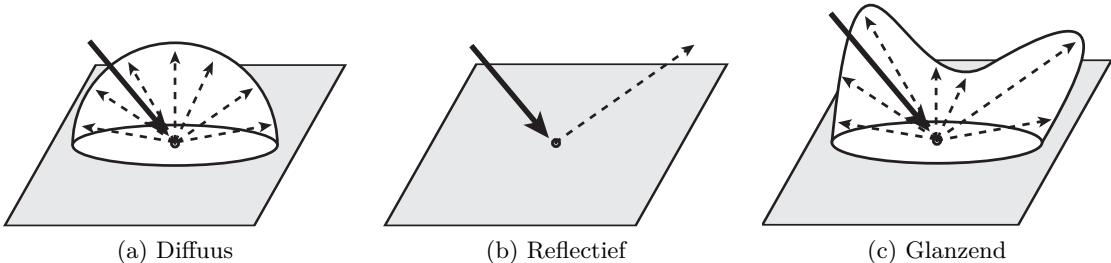
De term $G(x, y)$ wordt de geometrische term genoemd. De waarde ervan is afhankelijk van de geometrie in het punt x en y :

$$G = \frac{\cos(N_x, \Psi) \cos(N_y, -\Psi)}{r_{xy}^2}.$$

De term $V(x, y)$ wordt de visibiliteitsterm genoemd. Deze term drukt uit of twee punten x en y zichtbaar zijn voor elkaar. Indien er zich een object tussen de punten x en y bevindt dan wordt het invallend licht uit y geblokkeerd en is $V(x, y) = 0$. In dat geval ligt het punt x in de schaduw van een object. De visibiliteitsterm speelt dus een belangrijke rol bij het berekenen van schaduwen in een scène. In hoofdstuk 3 wordt in meer detail uitgelegd hoe deze visibiliteitsterm kan worden berekend.

2.1.5 Materialen

Een materiaal kan zowel diffuus, glanzend als reflectief zijn. Een diffuus materiaal reflecteert het licht uniform in alle richtingen. De gereflecteerde radiantie is dus onafhankelijk van de kijkrichting. Een voorbeeld van een diffuus materiaal is papier. Bij een reflectief materiaal wordt het invallend licht perfect gereflecteerd. Een spiegel is hiervan het beste voorbeeld. De meeste materialen zijn noch perfect diffuus, noch perfect reflectief. Veel eer zijn ze een combinatie van beide. Dergelijke materialen worden glanzend genoemd. Een voorbeeld van zo'n materiaal is gelakt hout. Figuur 2.4 illustreert de verschillende materiaaltypes [DBB06].



Figuur 2.4: Drie types van materialen: diffuus, reflectief en glanzend.

Er bestaan verscheidene modellen om de BRDF van een materiaal te beschrijven. Als voorbeeld worden twee modellen besproken: het Lambertiaans model en het Phong model.

Lambertiaans Model

Een Lambertiaans model wordt gebruikt voor diffuse materialen. Aangezien een diffuus materiaal het licht uniform reflecteert, ongeacht de kijkrichting Θ , is de BRDF voor een diffuus oppervlak een constante:

$$f_r(x, \Psi \rightarrow \Theta) = \frac{\rho_d}{\pi} = k_d.$$

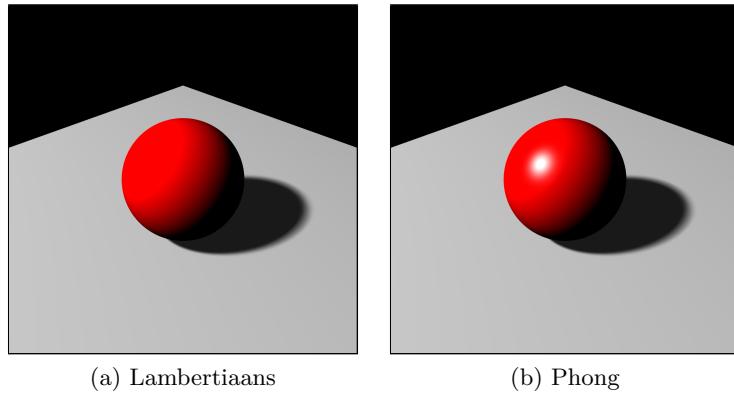
Diffuse materialen komen in realiteit niet voor. Toch worden deze frequent gebruikt voor het modelleren van materialen zoals papier, muren, enz. Figuur 2.5a toont een voorbeeld van een materiaal dat wordt beschreven met het Lambertiaans model.

Phong Model

Het Phong model beschrijft een combinatie van een diffuus en glanzend materiaal. Het reflectieve gedeelte is afhankelijk van de kijkrichting:

$$f_r(x, \Psi \rightarrow \Theta) = k_s \frac{(R \cdot \Theta)^n}{N \cdot \Psi} + k_d.$$

Daarbij beschrijft R de richting van de perfecte reflectie van het invallend licht. Figuur 2.5b toont een voorbeeld van een materiaal dat wordt beschreven met het Phong model.



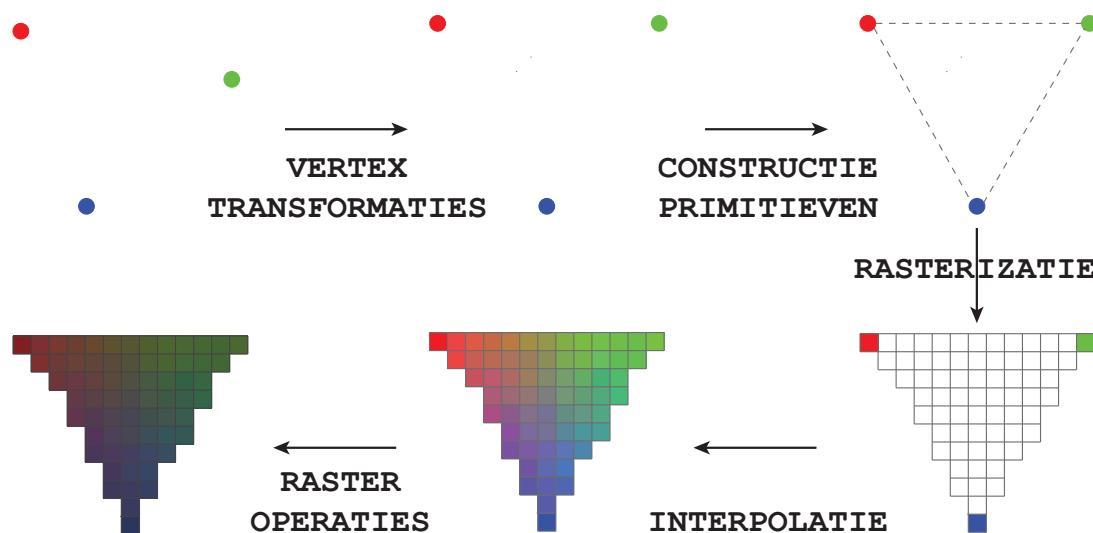
Figuur 2.5: Twee verschillende modellen om materialen te beschrijven.

2.2 Grafische Pijplijn

De grafische pijplijn beschrijft het volledige proces om een virtuele scène te visualiseren. Aan de hand van een beschrijving van de scène (geometrie, materialen, enz.) wordt een tweedimensionaal beeld opgebouwd. Dit beeld is opgebouwd uit pixels. Pixels, afgeleid van de woorden *picture elements*, zijn kleine punten die elk een eigen kleurwaarde hebben.

2.2.1 Rasterisatie

De grafische pijplijn kan worden opgesplitst in drie stappen: vertex transformaties, rasterisatie en raster operaties [SWND07, FK03]. Zie figuur 2.6 voor een grafische weergave van de verschillende stappen.



Figuur 2.6: De grafische pijplijn. Vertices worden eerst getransformeerd naar de gewenste positie en oriëntatie. Daarna worden polygonen gebouwd en gerasterd zodat fragmenten ontstaan. De kleurwaarde voor elk fragment wordt berekend door interpolatie van de hoekpunten. Ten slotte worden raster operaties toegepast om de uiteindelijke kleurwaarde te bekomen.

Vertex Transformaties

In de eerste stap worden de vertices getransformeerd naar posities op het scherm. Dit gebeurt met behulp van transformatiematrizes. De waardes in deze matrices zijn afhankelijk van het object, camera en beeldscherm waarop wordt geprojecteerd. In sectie 2.3 wordt deze stap in meer detail besproken.

Rasterisatie

Geometrische primitieven (bijvoorbeeld driehoeken) worden geconstrueerd aan de hand van de getransformeerde vertices. Deze primitieven worden vervolgens uitgezet op een raster en omgezet naar fragmenten. Elk fragment houdt waardes bij voor positie, kleur, diepte ... Voor fragmenten binnen een geometrische primitief worden deze waardes geïnterpoleerd aan de hand van de vertices.

Raster Operaties

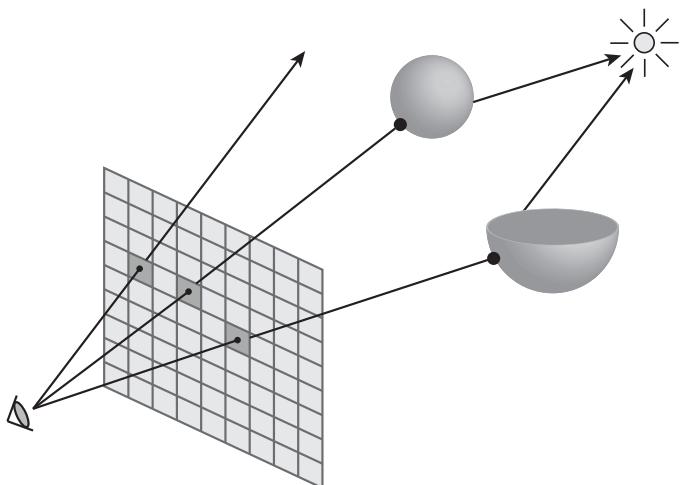
Elk fragment komt overeen met een pixel in de *framebuffer*. Een framebuffer houdt het beeld bij dat in een volgende stap op het beeldscherm wordt geprojecteerd. Vooraleer alle fragmenten worden omgezet naar pixels, worden nog enkele operaties uitgevoerd. Zo wordt onder andere voor elk fragment de finale kleur berekend afhankelijk van licht- en materiaaleigenschappen.

2.2.2 Grafische Verwerkingsseenheid

De grafische verwerkingsseenheid (GVE) is een hardware onderdeel van een computer. Dit onderdeel is speciaal ontworpen om de grafische pijplijn na te bootsen en beperkt zich dan ook tot instructies van dit type. Dit staat in contrast met de processor of centrale verwerkingsseenheid (CVE) van een computer. De processor is namelijk ontworpen om een grote variëteit aan instructies te kunnen verwerken (tekstverwerking, beeldverwerking ...). Doordat de GVE geoptimaliseerd is voor het uitvoeren van instructies met betrekking tot de grafische pijplijn, kunnen dergelijk instructies veel sneller worden uitgevoerd op de GVE dan op de CVE. Het is dankzij de GVE dat interactieve toepassingen, zoals videospelletjes, mogelijk zijn.

2.2.3 Ray Tracing

Een andere veelgebruikte visualisatietechniek is *ray tracing*. Vanuit het kijkpunt worden zichtstralen geschoten door het beeldvlak. Elk element op dit beeldvlak komt overeen met een pixel op het uiteindelijke beeld. De zichtstralen worden getraceerd doorheen de scène en getest op intersectie met objecten. Het dichtste intersectiepunt is het punt dat zichtbaar is vanuit het kijkpunt. Vervolgens wordt voor dit snipunt nagegaan wat de radiantie is. Figuur 2.7 illustreert het principe van ray tracing.



Figuur 2.7: Ray tracing. Vanuit het kijkpunt worden zichtstralen afgeschoten. Bij intersectie met een object wordt nagegaan wat de radiantie is voor dit snipunt.

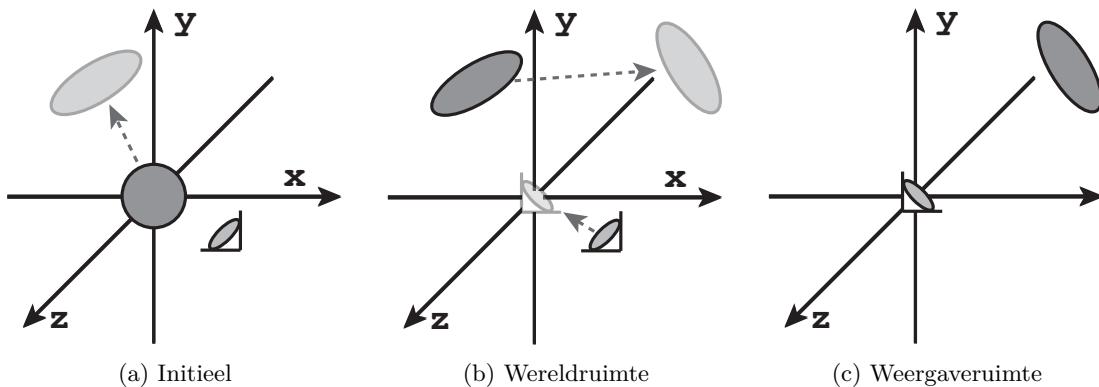
Ray tracing laat toe om het pad dat het licht aflegt, op een nauwkeurige manier te berekenen. Belichtingseffecten zoals schaduwen en reflecties kunnen dan ook op een eenvoudige en nauwkeurige manier worden berekend. Deze nauwkeurigheid brengt echter ook een grote computationele kost met zich waardoor interactieve toepassingen met ray tracing moeilijker haalbaar zijn. Visualisaties met behulp van ray tracing worden dan ook voornamelijk gebruikt als referentie.

2.3 Camerasysteem

Het camerasysteem beschrijft de ruimtelijke transformatie van driedimensionale punten naar tweedimensionale posities op het beeldvlak. Eerst wordt elk object geplaatst in de ruimte door een modeltransformatie toe te passen. Op deze manier wordt de volledige structuur van een scène opgebouwd. Daarna wordt elk punt getransformeerd in functie van de camera. Vervolgens kan een orthografische of perspectief-projectie worden toegepast. Deze projecties transformeren het zichtbare gedeelte van een scène naar het canoniek weergavevolume. Het is aan de hand van dit volume dat voor elk punt in een scène de positie op een tweedimensionaal beeldvlak wordt bepaald. Om al deze transformaties uit te voeren, worden homogene coördinaten gebruikt [Shi05].

2.3.1 Homogene Coördinaten

De transformatiematrices die in de volgende secties worden besproken, maken allemaal gebruik van 4×4 matrices. Dit komt omdat bij de projectie van een scène naar het beeldvlak, een homogeen coördinaat h wordt gebruikt. Dit is een extra coördinaat die wordt toegevoegd aan elk punt of vector. Deze coördinaat drukt uit in welke mate de andere coördinaten geschaald zijn. Voor een punt is $h = 1$ en voor een vector is $h = 0$. Het belang van deze coördinaten zal duidelijk worden bij het bespreken van de perspectief projectie.



Figuur 2.8: Eerst wordt de modeltransformatie toegepast op objecten in de scène (links). Daarna wordt de weergavetransformatie toegepast waardoor de camera wordt geplaatst op de oorsprong. Alle objecten worden bovendien mee getransformeerd (midden). De volledige scène na het toepassen van model- en weergave transformatie (rechts).

2.3.2 Modeltransformatie

De eerste stap bestaat uit het toepassen van de modeltransformatie. Daarbij wordt elk model in de scène getransformeerd met behulp van een transformatiematrix. Met

deze matrix kan onder andere de positie, oriëntatie en grootte van het model worden gewijzigd. Door elk model te transformeren naar de gewenste positie wordt de volledige structuur van een scène opgebouwd. De volledige ruimte van vertices na modeltransformatie wordt de wereldruimte genoemd. Figuur 2.8a illustreert deze stap.

Een voorbeeld van een transformatiematrix waarbij de x -, y - en z -coördinaten van een model worden gescaleerd, is de volgende:

$$\mathbf{M}_m = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.1)$$

2.3.3 Weergavetransformatie

In de volgende stap worden alle vertices in wereldruimte getransformeerd naar weergaveruimte. Daarvoor wordt gebruik gemaakt van een camera. De camera bepaalt, net als bij het nemen van een foto, welke delen van een scène worden gevisualiseerd en welke niet. Daarom beschikt een camera over een positie (ook wel kijkpunt of oogpunt genoemd) en een oriëntatie uvw . De weergave transformatie heeft als doel de positie en oriëntatie van de camera te herleiden tot een standaardpositie en -oriëntatie. Daarbij wordt de volledige scène mee getransformeerd. De weergavematrix \mathbf{M}_w beschrijft de transformaties die hiervoor nodig zijn:

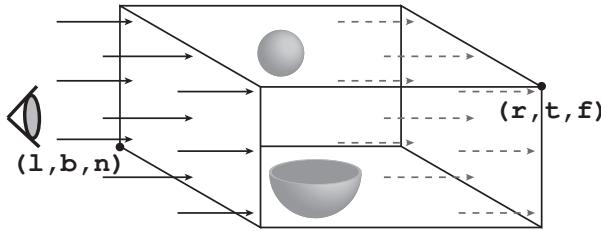
$$\mathbf{M}_w = \begin{bmatrix} x_u & y_u & z_u & 0 \\ x_v & y_v & z_v & 0 \\ x_w & y_w & z_w & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_c \\ 0 & 1 & 0 & -y_c \\ 0 & 0 & 1 & -z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

De ruimte na het uitvoeren van de weergavetransformatie wordt weergave- of oogruimte genoemd. Figuur 2.8b illustreert deze stap. Na het toepassen van de model- en weergavetransformatie wordt de scène geprojecteerd op het scherm. Daarvoor wordt een orthografische of perspectief-projectie gebruikt.

2.3.4 Orthografische Projectie

Orthografische projectie heeft enkele opvallende kenmerken. Lijnen die in een virtuele scène parallel zijn, worden ook parallel geprojecteerd op het beeldvlak. Daarnaast is de lengte voor elke lijn dezelfde, ongeacht of een object zich ver of dicht bij de camera bevindt. Figuur 2.9 toont het weergavevolume voor de orthografische projectie.

Het balkvormig weergavevolume bestaat uit zes vlakken. Deze worden bepaald door de hoekpunten (l, b, n) en (r, t, f) . Vanuit het standpunt van de camera bepalen l en r het linker- en rechtervlak, b en t het boven- en ondervlak en n en f het dichtste en verste vlak. De orthografische projectie transformeert dit weergavevolume naar het



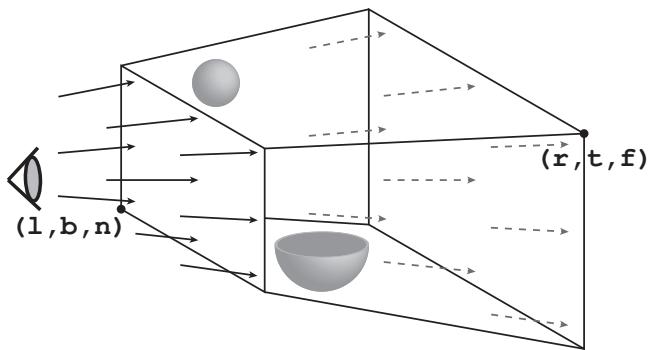
Figuur 2.9: Bij orthografische projectie wordt een balk als weergavevolume gebruikt. Alle projectielijnen zijn parallel en staan loodrecht op het projectievlak.

canoniek weergavevolume. Het canoniek weergavevolume is een kubus gecentreerd rond de oorsprong waarbij alle punten binnen het domein $[-1, 1]^3$ liggen. Dit weergavevolume wordt verderop besproken. Punten die buiten het orthografisch weergavevolume liggen worden niet getransformeerd aangezien ze buiten het zichtbare gedeelte van de scène vallen. De projectiematrix \mathbf{M}_o beschrijft de transformaties die worden uitgevoerd bij de orthografische projectie van een scène:

$$\mathbf{M}_o = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & \frac{2}{n-f} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -\frac{l+r}{2} \\ 0 & 1 & 0 & -\frac{b+t}{2} \\ 0 & 0 & 1 & -\frac{n+f}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

2.3.5 Perspectief-projectie

Bij perspectief-projectie speelt de diepte van een punt een belangrijke rol. Hoe verder een object gelegen is van de camera, hoe kleiner het wordt weergegeven. De zichtstralen zijn ook niet parallel zoals bij orthografische projectie. Vanuit alle richtingen worden punten in een scène op het kijkpunt geprojecteerd. Om dit te simuleren wordt een afgeknotte piramide als weergavevolume gebruikt (figuur 2.10). Punten binnen dit volume worden geprojecteerd op de top van de piramide, het beeldvlak.



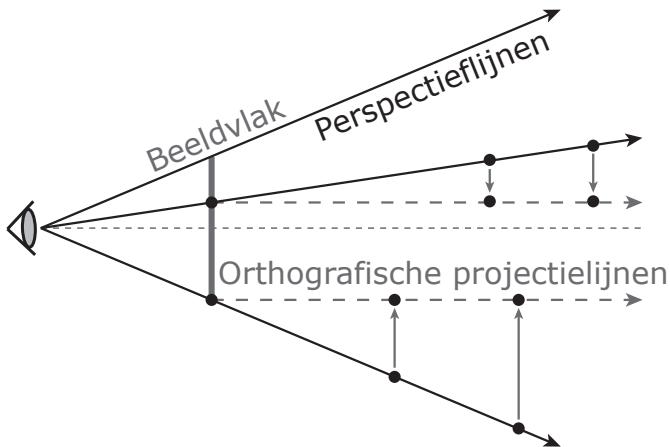
Figuur 2.10: Perspectief: het weergavevolume is een afgeknotte piramide.

Om perspectief te bekomen, wordt eerst een perspectief-transformatie toegepast. Daarbij worden de zichtstralen van de afgeknotte piramide getransformeerd naar de parallelle zichtstralen van een orthografische projectie. De afgeknotte piramide wordt dus getransformeerd naar een balk. Na het uitvoeren van deze perspectief-transformatie wordt een orthografische projectie toegepast. Dit vormt de volledige perspectief-projectie. Op deze manier worden alle punten uit het perspectief weergavevolume getransformeerd naar punten binnen het canoniek weergavevolume.

De transformatie om een perspectief weergavevolume om te zetten naar een orthografisch weergavevolume wordt beschreven door de matrix \mathbf{M}_p :

$$\mathbf{M}_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{n+f}{n} & -f \\ 0 & 0 & 1/n & 0 \end{bmatrix}.$$

Figuur 2.11 illustreert het effect van deze transformatie. De perspectieflijnen worden omgezet naar orthografische projectielijnen.



Figuur 2.11: Punten worden geprojecteerd in de richting van het kijkpunt, namelijk waar de perspectieflijnen snijden met het beeldvlak. De projectie van diezelfde punten kan ook worden bekomen met een orthografische projectie.

Na het uitvoeren van de perspectief-transformatie is echter nog altijd geen rekening gehouden met de diepte van een punt. Om het effect van perspectief te bekomen is het belangrijk dat objecten die ver van de camera gelegen zijn, kleiner worden weergegeven en vice versa. Hiervoor worden de eerder vermelde homogeenen coördinaten gebruikt. Na het uitvoeren van verschillende transformaties zal de homogene coördinaat van een vertex niet meer gelijk zijn aan één. Om deze vertex terug te transformeren naar cartesische coördinaten worden alle coördinaten gedeeld door de homogene coördinaat. Het homogene punt $[6 \ -2 \ 8 \ 2]$ bijvoorbeeld, is gelijk aan het cartesisch punt $[3 \ -1 \ 4 \ 1]$. Het

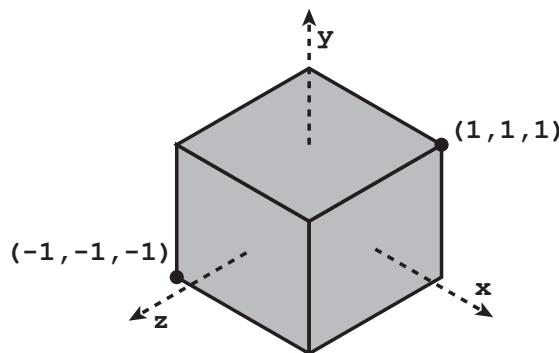
is deze homogenisatie of perspectief divisie die ervoor zorgt dat de diepte in rekening wordt gebracht. De volledige perspectief-projectie van een punt in wereldruimte naar een punt binnen het canoniek weergavevolume gaat dan als volgt:

$$\mathbf{M}_o \mathbf{M}_p \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} \longrightarrow \begin{bmatrix} x'/w' \\ y'/w' \\ z'/w' \\ 1 \end{bmatrix}.$$

Het uitvoeren van de perspectief divisie heeft gevolgen voor de nauwkeurigheid waarmee transformaties worden uitgevoerd. Naarmate een punt verder weg ligt van de camera, daalt de precisie voor dit punt. Voor een perspectief projectie komt dit overeen met een logaritmische verdeling van de z -coördinaten over het weergavevolume. Twee punten die op een bepaalde afstand van elkaar liggen, kunnen bijgevolg dezelfde z -coördinaat hebben. Dit heeft gevolgen voor de visibiliteitstest die in het volgende onderdeel wordt besproken. Bij een orthografische projectie is dit niet het geval, daar worden de z -coördinaten lineair verdeeld over de volledige diepte van het orthografisch weergavevolume.

2.3.6 Canoniek Weergavevolume

Het canoniek weergavevolume is een kubus die gecentreerd is rond de oorsprong en ribben van lengte twee heeft. Alle driedimensionale punten binnen deze kubus hebben cartesische coördinaten tussen -1 en +1. Figuur 2.12 illustreert hoe het canoniek weergavevolume eruitziet.



Figuur 2.12: Het canoniek weergavevolume.

De laatste stap omvat het projecteren van het canoniek weergavevolume naar het beeldvlak. Het beeldvlak is opgebouwd uit pixels en heeft een resolutie van $n_x \times n_y$. Voor elk punt in het canoniek weergavevolume wordt berekend op welke positie (pixel) dit punt

zich bevindt op het beeldvlak. Hiervoor wordt de volgende transformatie gebruikt:

$$\mathbf{M}_{cuv} = \begin{bmatrix} \frac{n_x}{2} & 0 & 0 & \frac{n_x-1}{2} \\ 0 & \frac{n_y}{2} & 0 & \frac{n_y-1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

De volledige transformatie van een driedimensionaal punt p naar een tweedimensionale positie op het scherm wordt dan als volgt berekend:

$$\mathbf{Mp} = \mathbf{M}_{cuv}\mathbf{M}_o\mathbf{M}_p\mathbf{M}_w\mathbf{M}_m p.$$

De x - en y -coördinaten bepalen de pixel waarop het punt wordt geprojecteerd. De z -coördinaat van elk punt wordt afgebeeld op het interval $[0, 1]$ en stelt de diepte van een punt voor. De z -coördinaten spelen nog een belangrijk rol bij het visualiseren van de scène. Aan de hand van deze coördinaten wordt immers beslist welke punten in de scène zichtbaar zijn en welke niet. Het identificeren van deze punten wordt het visibiliteitsprobleem genoemd.

Visibiliteitsprobleem

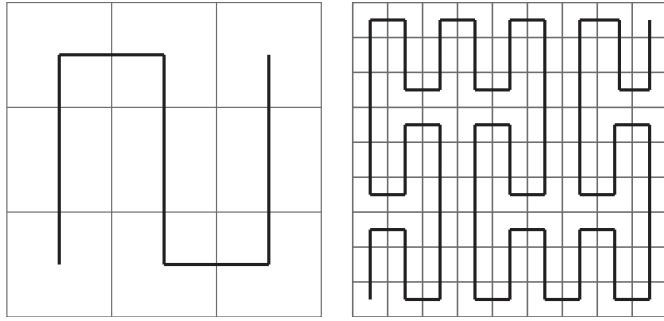
Om dit probleem aan te pakken wordt gebruik gemaakt van de Z-buffer. De Z-buffer is een tweedimensionale buffer van dezelfde resolutie als het beeldvlak. Elke pixel bevat een z -waarde of dieptewaarde binnen het interval $[0, 1]$. Hierbij komen lage waarden overeen met punten die zich dichtbij bevinden en komen hoge waarden overeen met punten die zich veraf bevinden. Initieel worden alle pixels in de Z-buffer op 1 geplaatst. Een waarde van 1 wil zeggen dat er op die positie geen object voorkomt.

Tijdens het visualiseren wordt de z -coördinaat van elk punt vergeleken met de overeenkomstige dieptewaarde in de Z-buffer. Dit wordt de dieptetest genoemd. Indien de z -coördinaat kleiner is dan de opgeslagen dieptewaarde, dan wil dit zeggen dat het punt zichtbaar is. De dieptewaarde in de Z-buffer wordt dan vervangen door de z -coördinaat van het punt. Omgekeerd, indien de z -coördinaat van het punt groter is dan de opgeslagen dieptewaarde, dan is het punt niet zichtbaar. Er is immers al een ander punt dat dichter ligt. Op deze manier wordt voor elk punt nagegaan of het zichtbaar is of niet.

2.4 Ruimtevullende Curves

Ruimtevullende curves beschikken over enkele interessante eigenschappen voor het comprimeren en terugvinden van data. Omwille van deze eigenschappen spelen ze een belangrijke rol bij de compressie van coherente schaduwmannen (zie hoofdstuk 5).

Ruimtevullende curves bepalen een ordening voor alle punten in een n -dimensionale hyperkubus. Voorbeelden van een hyperkubus zijn een vierkant ($n = 2$) en een kubus ($n = 3$). Meer wiskundig uitgedrukt zijn ruimtevullende curves continue functies

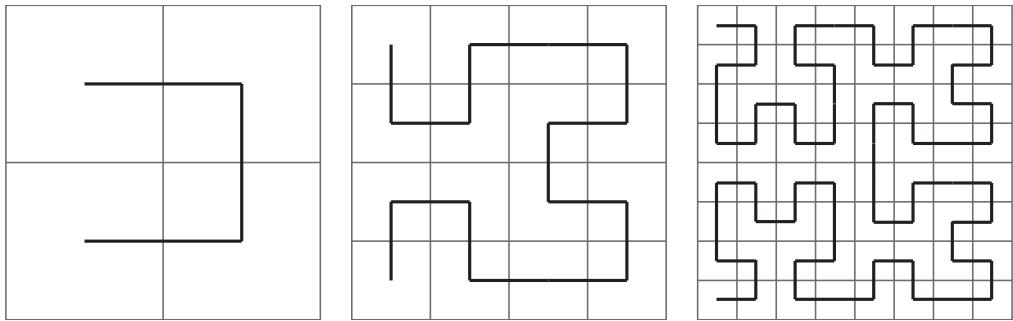


Figuur 2.13: Peano curve van eerste en tweede orde. Deze curves bepalen een ordening voor respectievelijk 3^2 en 3^4 punten.

die een, weliswaar niet-intuïtieve, mapping voorzien tussen $[0, 1]$ en $[0, 1]^n$ [Pea90, Hil91].

Figuur 2.13 toont de Peano curve. Dit is de eerste ruimtevullende curve die werd ontdekt. De Peano curve is beperkt tot twee dimensies en voorziet een ordening voor ruimtes met 3^{2k} punten, met $k \in \mathbb{R}_0^+$. In praktijk wil dit zeggen dat aan alle punten een getal $i \in [0, 3^{2k} - 1]$ kan worden toegekend. Dit getal bepaalt de volgorde waarin elk punt wordt bezocht door de Peano curve. Het omgekeerde is ook mogelijk. Aan de hand van een getal $i \in [0, 3^{2k} - 1]$ kunnen de x - en y -coördinaten van een punt worden bepaald. Doordat de Peano curve recursief is gedefinieerd, wordt deze ook aanzien als een speciale vorm van fractaal.

Een andere ruimtevullende curve is de Hilbert curve. Deze curve is gebaseerd op de Peano curve, maar heeft een andere vorm en is bovendien multidimensionaal. De curve definieert een ordening voor 2^{nk} punten met $n \in \mathbb{R}_0^+$ het aantal dimensies en $k \in \mathbb{R}_0^+$ de orde van de curve. Figuur 2.14 toont de geconstrueerde Hilbert curve van tweede en derde orde.



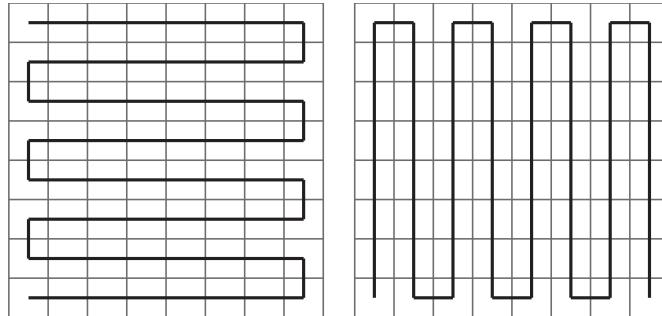
Figuur 2.14: Tweedimensionale Hilbert curve van eerste, tweede en derde orde. Deze curves bepalen een ordening voor respectievelijk 2^2 , 2^4 en 2^6 punten.

De Hilbert curve kan op een geometrische manier worden geconstrueerd [Ham06]. Er

wordt vertrokken van een curve van orde één. Gegeven een curve van orde k , dan wordt een curve van orde $k + 1$ als volgt geconstrueerd:

1. Roteer een kopie van de originele curve 90° tegen de klok in. Plaats deze vervolgens in het deel links onderaan.
2. Roteer een kopie van de originele curve 90° met de klok mee. Plaats deze vervolgens in het deel links bovenaan.
3. Plaats een kopie van de originele curve in elk van de twee rechterdelen.
4. Verbind de vier aparte curves met elkaar.

De Hilbert curve beschikt over een interessante eigenschap. Punten die dicht bij elkaar liggen in een n -dimensionale ruimte, liggen namelijk ook dicht bij elkaar in de Hilbert ordening. Het gevolg is dat de Hilbert curve zeer goed is voor het bewaren van de lokale eigenschappen van punten. Een voorbeeld hiervan zijn nabijgelegen pixels op een kleurenfoto. Deze hebben vaak gelijkaardige kleurwaarden. De Hilbert curve zorgt ervoor dat deze pixels dicht bij elkaar liggen in de Hilbert ordening. Hierdoor zijn de verschillen voor opeenvolgende punten klein. Er wordt met andere woorden een coherente volgorde gedefinieerd doorheen alle punten in een ruimte. De Hilbert curve is tevens de ruimtevullende curve die het best de coherentie tussen nabijgelegen punten bewaart [Voo91, MJFS01].



Figuur 2.15: Tweedimensionale zigzag curve. Links: horizontaal. Rechts: verticaal.

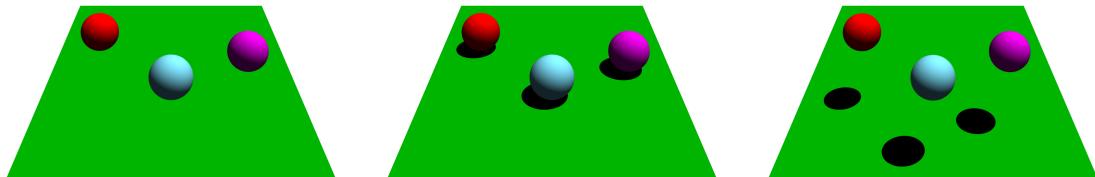
Een meer eenvoudige ruimtevullende curve is de zigzag curve. Deze curve doorkruist al zigzaggend alle punten in een ruimte. Een verschil met de Peano curve is dat de zigzag curve niet beperkt is tot punten in een hyperkubus. De ruimte kan dus een willekeurige grootte hebben. Voor het tweedimensionale geval zijn twee zigzag curves mogelijk: een horizontale en een verticale. Zie figuur 2.15 voor een grafische voorstelling van deze curves.

Hoofdstuk 3

Schaduwen

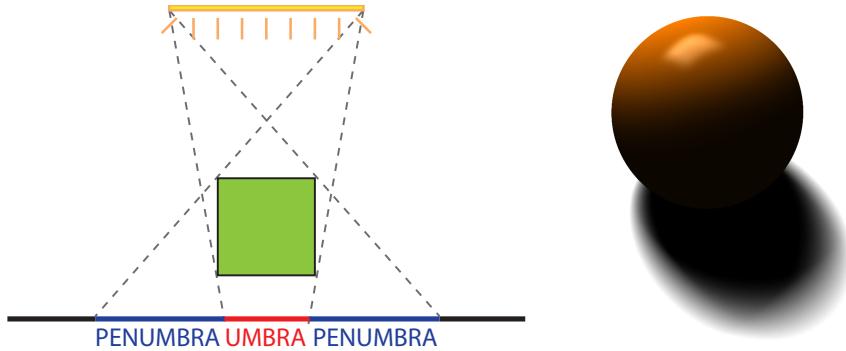
Schaduwen zijn het gevolg van objecten die invallend licht volledig of gedeeltelijk blokkeren. Hierdoor ontstaat een gebied dat relatief donkerder is dan de rest van de omgeving.

Schaduwen spelen een belangrijke rol bij de weergave van virtuele scènes. Ze helpen bij het beter begrijpen van de vorm en relatieve positie van objecten. Daarnaast geven ze een schatting voor de grootte en intensiteit van een lichtbron. Scènes zonder schaduwen voelen onrealistisch aan en zijn bovendien voor misinterpretatie vatbaar. Figuur 3.1 illustreert dat het moeilijk is om de relatieve positie van de bollen te kennen indien er geen schaduwen aanwezig zijn.



Figuur 3.1: Een scène zonder schaduwen (links) is voor misinterpretatie vatbaar. De bollen kunnen op het vlak liggen (midden) of boven het vlak zweven (rechts).

Schaduwen bestaan uit twee gebieden: umbra en penumbra. De umbra is het deel van een schaduw dat volledig wordt verduisterd en van waaruit de lichtbron helemaal niet zichtbaar is. De penumbra is het deel van een schaduw van waaruit de lichtbron wel nog gedeeltelijk zichtbaar is. Naargelang de positie binnen de penumbra is een verschillende intensiteit mogelijk. Figuur 3.2 illustreert het verschil tussen beide termen.



Figuur 3.2: Links: meetkundige voorstelling van umbra en penumbra. Rechts: umbra en penumbra voor een oppervlaktelichtbron.

Doorheen de jaren zijn verschillende technieken ontstaan om schaduwen in virtuele scènes te berekenen en te visualiseren. In dit hoofdstuk wordt de techniek van schaduwassen besproken. In sectie 3.1 wordt uitgelegd wat schaduwassen zijn en hoe ze worden gebruikt om schaduwen te visualiseren. In sectie 3.2 worden de typische problemen van schaduwassen besproken zoals vouwvervorming en zelfbeschaduwing.

3.1 Schaduwassen

Voor het berekenen van schaduwen in interactieve toepassingen worden zeer vaak schaduw- of dieptemappen gebruikt. Het volledige schaduwmap algoritme kan worden opgedeeld in twee stappen [Wil78]:

1. Construeer een dieptemap van de scène vanuit het standpunt van de lichtbron. Deze dieptemap bevat informatie over welke punten in een scène worden belicht en welke niet (sectie 3.1.1).
2. Bereken de scène opnieuw, maar dit keer vanuit het standpunt van de camera. Voor elk punt wordt nagegaan of het zichtbaar is vanaf de lichtbron of niet, gebruik makende van de dieptemap uit stap één (sectie 3.1.2).

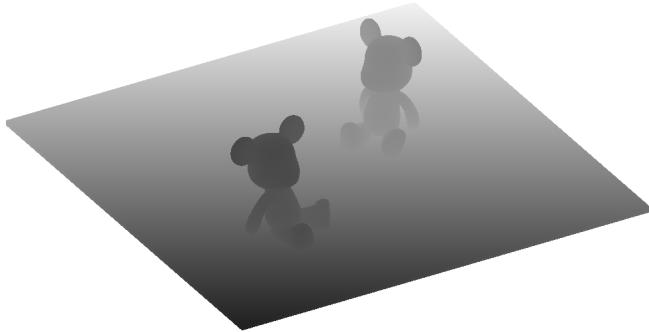
3.1.1 Dieptemappen

Een dieptemap is een visualisatie van een scène waarbij in plaats van kleurwaarden, dieptewaarden worden weergegeven. Deze dieptewaarden geven aan op welke afstand het dichtste punt zich bevindt ten opzichte van de camera. Dieptemappen worden ook wel schaduwassen genoemd.

Constructie

Het construeren van een dieptemap verschilt amper van het normale proces om een scène te visualiseren. Meer nog, tijdens het visualiseren van een scène wordt al een dieptemap

aangemaakt, namelijk op de Z-buffer (zie sectie 2.3.6). De Z-buffer houdt voor elke pixel bij wat de kleinste z -waarde of dieptewaarde is. Een dieptemap van een scène is dus niets anders dan een visualisatie van de Z-buffer. Figuur 3.3 toont een voorbeeld van een dieptemap.



Figuur 3.3: Een dieptemap van een scène. Zwarte pixels geven punten aan die dichtbij zijn gelegen. Witte pixels zijn punten veraf.

Net als bij de visualisatie van een scène wordt bij de constructie van een dieptemap gebruik gemaakt van een cameramodel. Het is dit cameramodel dat in de eerste stap van het schaduwmap algoritme van belang is. Het model wordt aangepast zodat de scène wordt gevisualiseerd vanuit het standpunt van de lichtbron. Het is vanuit dit standpunt dat een dieptemap wordt gegenereerd. De volledige transformatie om een scène te visualiseren vanuit het standpunt van de lichtbron kan worden voorgesteld door één enkele transformatiematrix \mathbf{M}_L . Deze transformatie is van groot belang voor de volgende stap van het algoritme.

Een dieptemap die wordt aangemaakt vanuit het standpunt van de lichtbron, bevat interessante informatie voor het berekenen van schaduwen. Aan de hand van deze dieptemap kan namelijk worden bepaald welke punten worden belicht en welke niet. Van een punt dat niet zichtbaar is op deze dieptemap wordt immers verondersteld dat het geen licht ontvangt. Het invallend licht op dit punt wordt dan geblokkeerd door een ander punt dat zich dichter bij de lichtbron bevindt. Het verste punt bevindt zich met andere woorden in de schaduw van het dichtste punt.

Precisie

Een belangrijk technisch aspect van dieptemappen is de gebruikte numerieke precisie. Aangezien een computer een discrete machine is kunnen slechts een eindig aantal z -waardes worden voorgesteld binnen het interval $[0, 1]$. De numerieke precisie bepaalt dan ook hoeveel verschillende waardes binnen dit interval mogelijk zijn. Stel dat er 8 bits worden gebruikt om de dieptewaardes voor te stellen. Dit betekent dat er slechts $2^8 = 256$ verschillende waardes mogelijk zijn. Punten die relatief dicht bij elkaar gelegen zijn, kunnen bijgevolg eenzelfde dieptewaarde hebben. Daardoor is het mogelijk dat

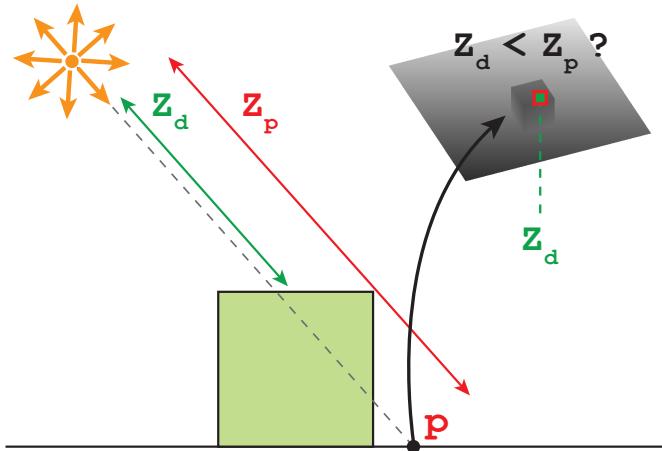
beide punten niet meer van elkaar te onderscheiden vallen, ook al liggen ze in werkelijkheid op een bepaalde afstand van elkaar. Om precisieproblemen te vermijden, worden standaard 32 bits vloottende komma getallen gebruikt.

De precisie wordt ook beïnvloed door de projectie die wordt gebruikt. Indien een orthografische projectie wordt toegepast dan worden de z -coördinaten lineair verdeeld over de volledige diepte van het orthografisch weergavevolume. De precisie is dan overal in de scène gelijk verdeeld. Indien een perspectief-projectie wordt toegepast dan zijn de z -coördinaten logaritmisch verdeeld over het weergavevolume. Daarbij daalt de precisie naarmate punten verder weg liggen van de camera. Bijgevolg is het mogelijk dat de dieptetest fout verloopt voor punten veraf.

3.1.2 Visibiliteitstest

Na het construeren van de dieptemap wordt de scène gevisualiseerd vanuit het standpunt van de camera. Tijdens de visualisatie wordt elk punt p aan een visibiliteitstest $V(p, l)$ onderworpen. Deze visibiliteitstest bepaalt of het punt p wordt belicht door de lichtbron op positie l . Als $V(p, l) = 0$ dan wordt p niet belicht en ligt het punt in schaduw. In het andere geval, $V(p, l) = 1$, wordt p wel belicht.

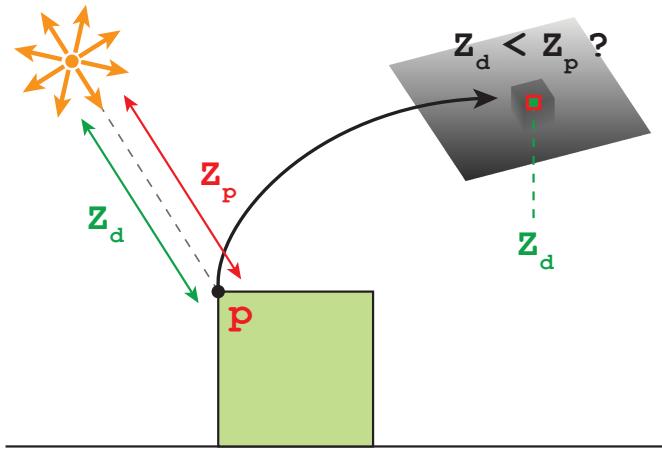
Figuur 3.4 en 3.5 illustreren de visibiliteitstest. Met behulp van een dieptemap wordt nagegaan of er een object is dat het invallend licht op het punt p blokkeert. Eerst wordt het punt p getransformeerd met de matrix \mathbf{M}_L : $\mathbf{M}_L p = p_L$. Daarbij stelt het punt p_L de positie voor van het punt p vanuit het standpunt van de lichtbron.



Figuur 3.4: Visibiliteitstest voor een punt in schaduw. Het punt p wordt geprojecteerd naar het standpunt van de lichtbron. De dieptewaarde z_p wordt dan vergeleken met de dieptewaarde z_d die is opgeslagen in de dieptemap. Aangezien $z_d < z_p$ ligt het punt p in schaduw.

De x - en y -coördinaten van p_L bepalen de overeenkomstige positie (pixel) van dit punt op de dieptemap. Aan de hand van deze positie kan de dieptewaarde z_d uit de dieptemap worden opgehaald. Voor een punt dat buiten het bereik van de dieptemap valt, wordt aangenomen dat het punt wordt belicht en dus niet in schaduw ligt.

In de volgende stap wordt z_d vergeleken met de z -coördinaat van het punt p_L (z_p). De waarde z_p drukt de afstand uit tussen de lichtbron l en het punt p . Als z_d kleiner is dan z_p , dan ligt het punt p in schaduw. In dit geval wordt het invallend licht namelijk geblokkeerd door een object dat zich dichter bij de lichtbron bevindt. Als z_d groter is dan z_p , dan betekent dit dat het invallend licht op p niet wordt geblokkeerd. Het punt ligt bijgevolg niet in schaduw.



Figuur 3.5: Visibiliteitstest voor een punt dat wordt belicht. Het punt p wordt geprojecteerd naar het standpunt van de lichtbron. De dieptewaarde z_p wordt dan vergeleken met de dieptewaarde z_d die is opgeslagen in de dieptemap. Aangezien $z_d = z_p$ wordt het punt p belicht.

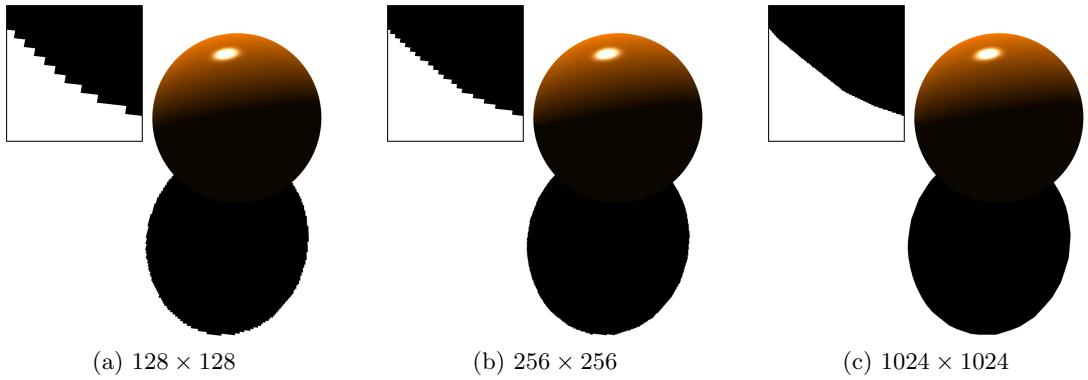
Merk op dat elk punt ofwel volledig in schaduw ligt ofwel volledig daarbuiten. Dit is het gevolg van het binaire resultaat van de visibiliteitstest. Schaduwassen zijn dus beperkt tot het weergeven van harde schaduwen.

3.2 Problemen

Schaduwassen introduceren enkele kenmerkende problemen zoals vouwvervorming en zelfbeschaduwing. Er bestaan verschillende methodes om deze problemen op te lossen.

3.2.1 Vouwvervorming

Vouwvervorming of aliasing is het gevolg van een signaal te weinig te bemonsteren. Het originele signaal kan dan niet worden gereconstrueerd waardoor fouten worden geïntroduceerd. Schaduwassen zijn zeer gevoelig voor vouwvervorming en dan vooral aan de rand van de schaduwen. Mogelijke oorzaken hiervan zijn een te lage resolutie van de dieptemap of het nemen van te weinig monsters bij de visibiliteitstest.



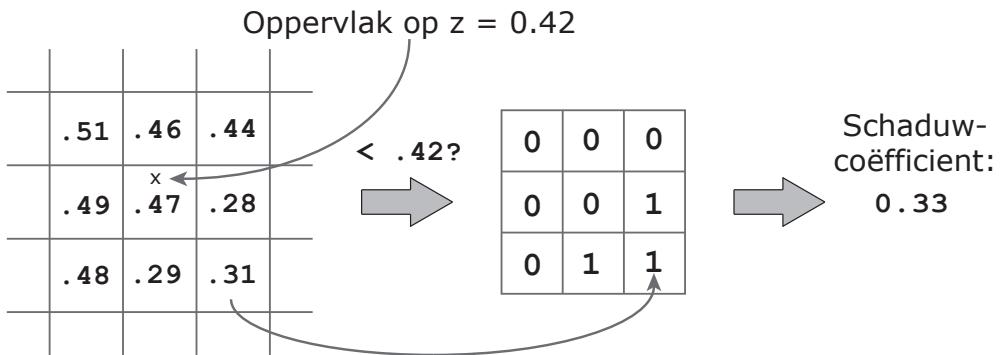
Figuur 3.6: Naarmate de resolutie van de dieptemappen toeneemt, stijgt ook de kwaliteit van de schaduwen.

Resolutie

Lage resoluties leiden tot dieptemappen met weinig en onnauwkeurige informatie. De gegenereerde schaduwen zijn blokkerig en ongedetailleerd. Figuur 3.6a illustreert dit effect voor een dieptemap van 128×128 pixels. Naarmate de resolutie toeneemt, daalt de vouwvervorming van de schaduwen. Dit komt omdat elke dieptemap over nauwkeurigere informatie beschikt. Figuur 3.6 toont het verschil in kwaliteit voor dieptemappen met toenemende grootte. Een hogere kwaliteit van schaduwen brengt echter ook een kost met zich mee, het berekenen van de dieptemap vraagt immers meer tijd.

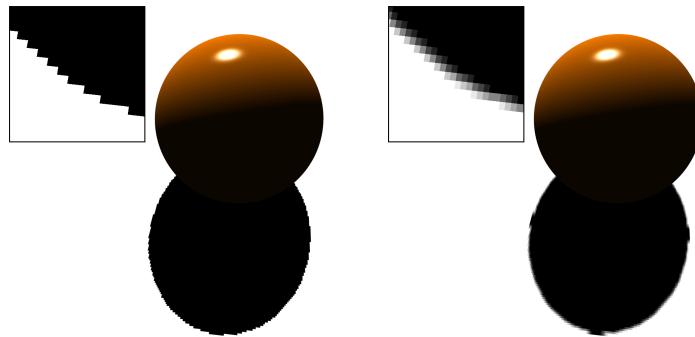
Percentage Closer Filtering

Een ander manier om vouwvervorming tegen te gaan is met *Percentage Closer Filtering* (PCF) [RSC87]. Met PCF worden meerdere visibiliteitstesten uitgevoerd voor een punt. Daarbij wordt gebruik gemaakt van de naburige dieptewaarden van een pixel op de dieptemap. Het gemiddelde resultaat van de visibiliteitstesten levert de schaduwcoëfficiënt op. Deze coëfficiënt drukt uit voor hoeveel percent een punt in schaduw ligt. Figuur 3.7 illustreert de bewerkingen die worden uitgevoerd met PCF.



Figuur 3.7: Percentage Closer Filtering (PCF). Waardes uit de dieptemap worden vergeleken met de z -waarde op het oppervlak (links). Het slaagpercentage van de visibiliteitstesten (midden) bepaalt de schaduwcoëfficient (rechts) [RSC87].

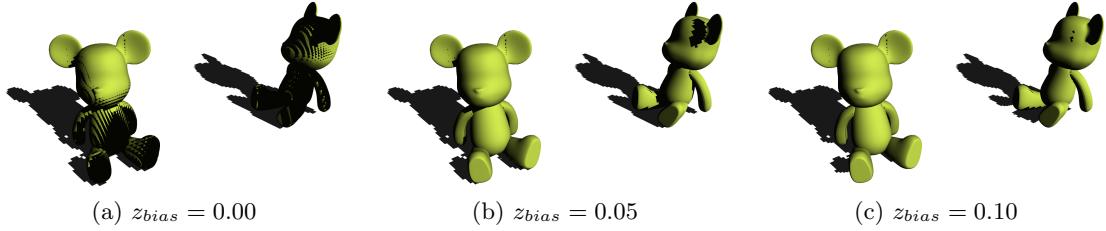
Doordat de visibiliteitstest niet langer beperkt is tot een binair resultaat, wordt ook de indruk van zachte schaduwen gewekt. Figuur 3.8 illustreert dit effect. Bemerk wel dat deze zachte schaduwen geen benadering zijn voor de échte penumbra. PCF zorgt enkel voor een zachtere overgang aan de randen en vermindert de vouwvervorming.



Figuur 3.8: Links: zonder PCF. Rechts: met PCF.

3.2.2 Zelfbeschaduwing

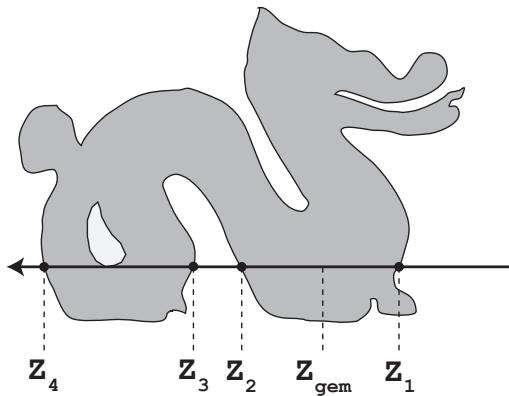
Zelfbeschaduwing is het gevolg van machine-imprecisie en kwantisatie van de Z-buffer. Om dit probleem op te lossen wordt gebruik gemaakt van een bias. Hierbij wordt aan elke dieptewaarde in de dieptemap een (kleine) constante factor toegevoegd. Daardoor schuift de schaduw lichtjes op naar achter en verdwijnt de zelfbeschaduwing. Indien de bias echter te groot wordt gekozen, dan is de verschuiving van de schaduw duidelijk zichtbaar. De keuze voor de bias is dus belangrijk voor het resultaat. Figuur 3.9 illustreert het effect van verschillende waarden als bias.



Figuur 3.9: Indien geen bias wordt gekozen treedt er zelfbeschaduwing op (links). Het toevoegen van een kleine factor aan de dieptewaarden verhindert dit (midden). Indien de bias echter te groot wordt gekozen, dan zijn de verschuivingen van de schaduwen duidelijk zichtbaar (rechts).

Duale Dieptelagen

Een andere manier om het probleem van zelfbeschaduwing op te lossen is met behulp van duale dieptelagen [WE03]. Duale dieptelagen maken gebruik van de dieptewaarde van het tweede dichtste punt om een adaptieve bias te kiezen. Figuur 3.10 illustreert het verschil tussen de dichtste (z_1) en tweede dichtste (z_2) dieptewaarden.



Figuur 3.10: Duale dieptelagen. Voor het correct verlopen van de visibiliteitstest mag elke waarde tussen z_1 en z_2 worden gekozen, bijvoorbeeld z_{gem} . Alle punten in schaduw ($z_{2,3,4}$) bevinden zich nog altijd verder dan z_{gem} . Ook voor andere punten (z_1) wijzigt het resultaat van de visibiliteitstest niet [RGKM07].

Aan de hand van de z_1 - en z_2 -waarden kan voor elke pixel een bias worden gekozen in functie van de geometrie:

$$z_{bias} = \min\left(\frac{z_2 - z_1}{2}, z_{max}\right)$$

Daarbij stelt z_{max} een maximum waarde van de bias voor. Door een adaptieve bias te gebruiken, wordt vermeden dat de bias te groot wordt gekozen en zo een zichtbare verschuiving van de schaduw veroorzaakt. Dit kan voorkomen indien twee oppervlakken

op een afstand kleiner dan z_{max} uit elkaar liggen. Merk op dat het resultaat van de visibiliteitstest niet wordt beïnvloed door deze techniek. Voor het correct verlopen van de visibiliteitstest mag immers elke waarde tussen z_1 en z_2 worden gebruikt. Figuur 3.10 illustreert dit.

3.3 Conclusie

In dit hoofdstuk werd de techniek van schaduwmappen besproken. Deze techniek laat toe om op een eenvoudige manier schaduwen te berekenen voor virtuele scènes.

Om een schaduw- of dieptemap te construeren kan gebruik worden gemaakt van de grafische pijplijn. Hierdoor is de kost om een dieptemap te genereren lineair met het aantal objecten in de scène. Daarnaast wordt bij constructie enkel de Z-buffer gebruikt. Het is bijvoorbeeld niet nodig om van elk fragment ook de kleur te berekenen. Er zijn dus minder berekeningen nodig om een dieptemap te construeren dan voor een volledige visualisatie van een scène. De voornaamste kost bij de constructie van een dieptemap is dan ook de resolutie.

Schaduwmappen hebben echter ook enkele nadelen. Zo kan er vouwvervorming optreden aan de rand van de schaduwen. Dit is het gevolg van de discretisatie die aanwezig is bij de constructie van een dieptemap. Een mogelijke oplossing hiervoor is het verhogen van de resolutie of het nemen van meerdere monsters voor de visibiliteitstest, bijvoorbeeld met percentage closer filtering. Een ander nadeel van schaduwmappen zijn zelfbeschaduwing. Om dit probleem te verhelpen kan een bias worden gebruikt. De techniek van duale dieptelagen laat toe om een bias te kiezen in functie van de geometrie.

Schaduwmappen dienen regelmatig te worden vernieuwd. Telkens de positie van een lichtbron of een ander object in de scène wijzigt, dient een nieuwe dieptemap te worden geconstrueerd. Schaduwen zijn immers afhankelijk van de lichtbronnen en geometrie in een scène. In praktijk is het meestal zo dat een nieuwe dieptemap wordt geconstrueerd telkens de scène wordt gevisualiseerd. Voor interactieve toepassingen betekent dit dat er minstens 24 dieptemappen per seconde worden gegenereerd.

Ten slotte zijn schaduwmappen beperkt tot het weergeven van harde schaduwen. Een mogelijke manier om toch zachte schaduwen te visualiseren is met de techniek van coherente schaduwmappen. Deze techniek wordt in het volgende hoofdstuk besproken.

Hoofdstuk 4

Coherente Schaduwassen

De klassieke schaduwmap is beperkt tot het weergeven van harde schaduwen. Om zachte schaduwen te genereren kunnen bijvoorbeeld meerder schaduwassen worden geconstrueerd op het oppervlak van een lichtbron. Voor elk punt in de scène wordt dan nagegaan vanuit welke posities op de lichtbron dit punt zichtbaar is. Het gemiddelde resultaat geeft een indicatie van het percentage aan invallend licht dat wordt geblokkeerd. Dit percentage geeft aanleiding tot zachte schaduwen.

Voor interactieve toepassingen (met dynamische scènes) is een dergelijke aanpak echter niet haalbaar. Doordat bij elke visualisatie van een scène alle schaduwassen opnieuw dienen te worden geconstrueerd, wordt het visualisatieproces sterk vertraagd. Een betere oplossing zou erin bestaan om deze dieptemappen vooraf te construeren. Op deze manier wordt vermeden dat de dieptemappen telkens opnieuw moeten worden gegenereerd. Door deze dieptemappen ook nog te comprimeren, kan een compacte representatie worden bekomen. Dit is het idee achter coherente schaduwassen.

Coherente schaduwassen vormen een uitbreiding op de klassieke schaduwmap. Bij coherente schaduwassen worden op voorhand dieptemappen geconstrueerd van een specifiek driedimensionaal object. Deze dieptemappen kunnen tijdens de visualisatie van een scène worden gebruikt om zachte schaduwen te genereren. Door de coherentie tussen de dieptemappen uit te buiten kan een compacte en snel toegankelijke datastructuur worden opgesteld. Daardoor kunnen coherente schaduwassen worden gebruikt voor interactieve toepassingen.

In sectie 4.1 wordt een inleiding gegeven tot de techniek van coherente schaduwassen. Daarbij worden de verschillende onderdelen van het algoritme op een hoog niveau beschreven. In sectie 4.2 wordt vervolgens dieper ingegaan op de constructie van dieptemappen voor een driedimensionaal object. Met behulp van deze dieptemappen kunnen tenslotte schaduwen worden gevisualiseerd. De verschillende stappen die hiervoor nodig zijn worden beschreven in sectie 4.3. De resultaten van deze techniek worden besproken in sectie 4.4.

4.1 Introductie

Bij het klassieke schaduwmap algoritme wordt een dieptemap geconstrueerd vanuit het standpunt van de lichtbron. Deze dieptemap bevat de visibiliteitsinformatie van de scène vanuit een specifieke positie. De visibiliteitsinformatie vanuit andere standpunten is echter niet gekend. Doordat het schaduwmap algoritme beperkt is tot één standpunt kunnen enkel harde schaduwen worden weergegeven.

Bij coherente schaduwmappen wordt de volledige visibiliteitsinformatie van een driedimensionaal object op voorhand berekend. Hiervoor worden orthografische dieptemappen geconstrueerd op verscheidene posities rondom dit object. Doordat de dieptemappen worden geconstrueerd vanuit alle richtingen, wordt de volledige visibiliteitsinformatie van dit object bekomen. Figuur 4.1 illustreert dit. De manier waarop deze posities worden bepaald, wordt in meer detail besproken in sectie 4.2. Aangezien de dieptemappen vooraf worden berekend, mag de geometrie van dit object niet wijzigen over tijd. Dit worden rigide objecten genoemd.

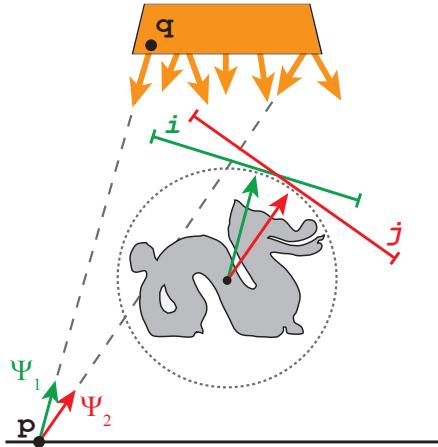


Figuur 4.1: Om de visibiliteitsinformatie van een object te verzamelen worden dieptemappen geconstrueerd op allerhande posities rondom dit object [RGKM07].

Na het construeren van de dieptemappen worden ze op een foutloze manier gecomprimeerd tot coherente schaduwmappen. De resulterende datastructuur is compact en snel toegankelijk. Daardoor kunnen coherente schaduwmappen worden gebruikt voor interactieve toepassingen. De verschillende stappen voor de compressie en decompressie van de dieptemappen worden besproken in hoofdstuk 5. Voor het verdere verloop van dit hoofdstuk is de gebruikte compressietechniek echter niet belangrijk.

Tijdens het visualiseren van de scène worden de coherente schaduwmappen gebruikt om het resultaat van de visibiliteitstest te bepalen. Daarvoor wordt telkens de coherente schaduwmap gebruikt die het best overeenkomt met de richting van het invallend licht op een punt. Figuur 4.2 illustreert dit. Eenmaal deze dieptemap is gevonden kan de eigenlijke visibiliteitstest worden uitgevoerd, deze test is dezelfde als bij het klassieke schaduwmap algoritme.

Doordat de opgemeten visibiliteitsinformatie van het object onafhankelijk is van de positie of oriëntatie van de lichtbron, kunnen de coherente schaduwassen worden gecombineerd met allerhande modellen van lichtbronnen. Voor een oppervlaktelichtbron wil dit zeggen dat zachte schaduwen kunnen worden berekend. Daarvoor worden meerdere visibiliteitstesten uitgevoerd, elk voor een andere positie op het oppervlak van de lichtbron (figuur 4.2). Deze visualisatiestap wordt in meer detail besproken in sectie 4.3.



Figuur 4.2: Tijdens het visualiseren van de scène worden voor elk punt meerdere visibiliteitstesten uitgevoerd. Telkens wordt de dieptemap gebruikt die het best overeenkomt met de richting van het invallend licht.

4.2 Berekenen van de Visibiliteitsinformatie

Om de visibiliteitsinformatie van een object te berekenen worden dieptemappen gebruikt. Deze dieptemappen worden geconstrueerd op verscheidene posities rondom het object. Daarvoor wordt gebruik gemaakt van de omhullende sfeer. De omhullende sfeer omvat de volledige geometrie van dit object en bij voorkeur is de oppervlakte van deze sfeer minimaal. Elke dieptemap bevindt zich op een welbepaalde positie op deze sfeer. Daarbij is het uiteraard belangrijk dat de visibiliteitsinformatie vanuit alle richtingen wordt berekend. Om deze posities te bepalen worden sferische coördinaten gebruikt.

4.2.1 Sferische Coördinaten

Sferische coördinaten (r, θ, ϕ) bepalen de positie van een punt in een driedimensionale ruimte. Hierbij is $r \geq 0$, $\theta \in [0, \pi]$ en $\phi \in [0, 2\pi]$. Sferische coördinaten kunnen worden omgezet naar driedimensionale cartesische coördinaten met de volgende transformatie:

$$\begin{aligned} x &= r \sin \theta \sin \phi, \\ y &= r \cos \theta, \\ z &= r \sin \theta \cos \phi. \end{aligned}$$

Het omgekeerde is ook mogelijk, namelijk van driedimensionale cartesische coördinaten naar sferische coördinaten:

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2}, \\ \theta &= \arccos \frac{y}{r}, \\ \phi &= \text{atan2}(x, z). \end{aligned}$$

4.2.2 Bemonstering

Met behulp van sferische coördinaten worden de posities van de dieptemappen op de omhullende sfeer bepaald. Zoals eerder werd vermeld omvat de omhullende sfeer de volledige geometrie van een object. De afstand tussen de sfeer en het centrum van het object bepaalt dus de afstand r waarop de dieptemappen worden geconstrueerd. Voor het bepalen van de θ - en ϕ -coördinaten worden reguliere monsters genomen in het domein $[0, \pi] \times [0, 2\pi]$. Samen met de afstand r stellen ze sferische coördinaten voor. Bijgevolg kunnen ze worden omgezet naar driedimensionale cartesische coördinaten. Dit zijn de uiteindelijke posities waarop de dieptemappen worden geconstrueerd. Doordat reguliere monsters worden genomen, wordt de visibiliteitsinformatie van een object vanuit alle richtingen opgemeten. Figuur 4.3 illustreert de verschillende stappen die nodig zijn om de posities van de dieptemappen te bepalen.

De posities die op bovenstaande manier worden gegenereerd, zijn echter niet uniform verdeeld over het oppervlak van de omhullende sfeer. De polen worden namelijk meer bemonsterd dan andere gebieden, bijvoorbeeld het gebied rond de evenaar. Om dit te vermijden kan worden gebruik gemaakt van een uniforme verdeling.

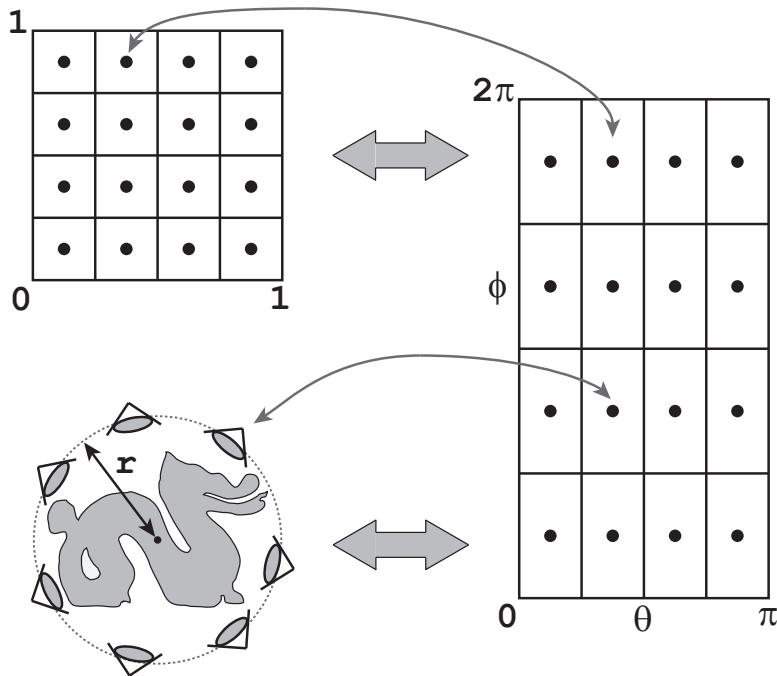
4.2.3 Uniforme Verdeling

Om tot een uniforme verdeling te komen, worden reguliere monsters genomen binnen het domein $[0, 1]^2$. Op deze monsters wordt vervolgens onderstaande transformatie toegepast:

$$\begin{aligned} \theta &= \cos^{-1}(1 - 2s), \\ \phi &= 2\pi t, \end{aligned}$$

met $s, t \in [0, 1]$, $\theta \in [0, \pi]$ en $\phi \in [0, 2\pi]$. De getransformeerde monsters vormen samen met de afstand r sferische coördinaten (r, θ, ϕ) . Net als bij de niet-uniforme verdeling kunnen deze punten worden omgezet naar driedimensionale cartesische coördinaten. Het resultaat is een uniforme verdeling van punten op het oppervlak van de omhullende sfeer.

De bekomen punten kunnen ook worden terug getransformeerd naar punten binnen $[0, 1]^2$. Daarvoor worden de punten eerst omgezet naar sferische coördinaten en daarna



Figuur 4.3: Positiebepaling op de omhullende sfeer voor niet-uniforme bemonstering. Er wordt vertrokken van N reguliere monsters binnen het domein $[0, 1]^2$. Deze worden vervolgens gescaleerd naar $[0, \pi] \times [0, 2\pi]$. Samen met de afstand r stellen ze sferische coördinaten voor. Transformatie naar driedimensionale cartesische coördinaten levert de positie op voor elke dieptemap op de omhullende sfeer van het object. Elk punt op de omhullende sfeer kan bovendien worden terug getransformeerd naar een punt binnen het domein $[0, 1]^2$.

wordt onderstaande transformatie toegepast:

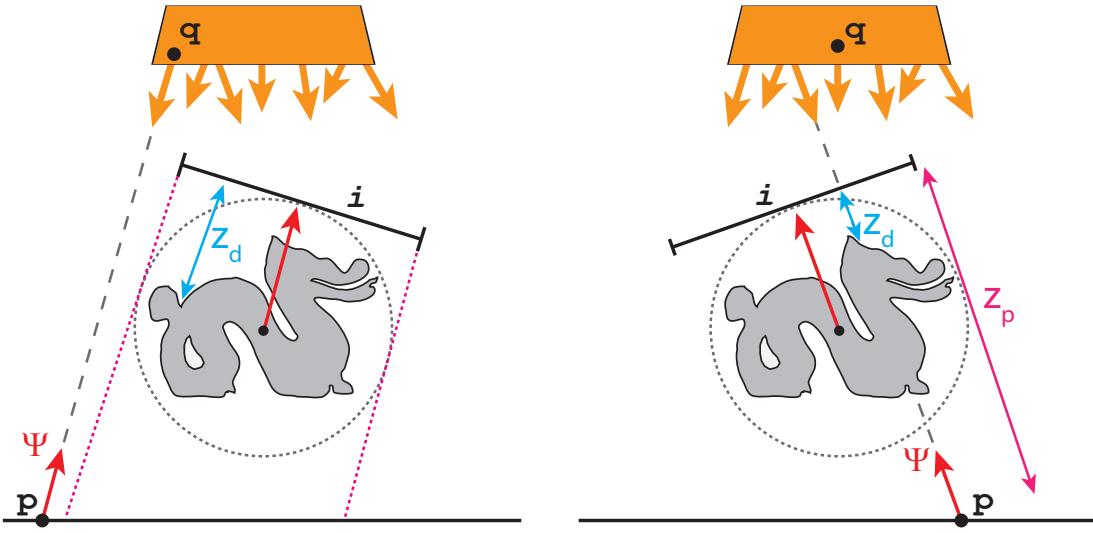
$$\begin{aligned} s &= \frac{\cos \theta - 1}{-2}, \\ t &= \frac{\phi}{2\pi}. \end{aligned}$$

4.2.4 Conclusie

In deze sectie werden de stappen besproken om de posities van de dieptemappen op de omhullende sfeer te bepalen. Merk op dat elke dieptemap op een unieke wijze geïdentificeerd wordt aan de hand van zijn positie. Bovendien kan elke positie op de omhullende sfeer terug worden getransformeerd naar een punt in de tweedimensionale ruimte $[0, 1]^2$. Dit verband is van groot belang bij het berekenen van schaduwen in een scène.

4.3 Visualisatie

Doordat de coherente schaduwassen de visibiliteitsinformatie van een object bevatten, kunnen ze, net als bij het klassieke schaduwmap algoritme, worden gebruikt om na te gaan of dit object het invallend licht op een punt blokkeert. Om deze visibiliteitstest uit te voeren wordt telkens de dieptemap gebruikt die het best overeenkomt met de richting van het invallend licht. Figuur 4.4 en 4.5 illustreren de verschillende stappen die hiervoor nodig zijn.

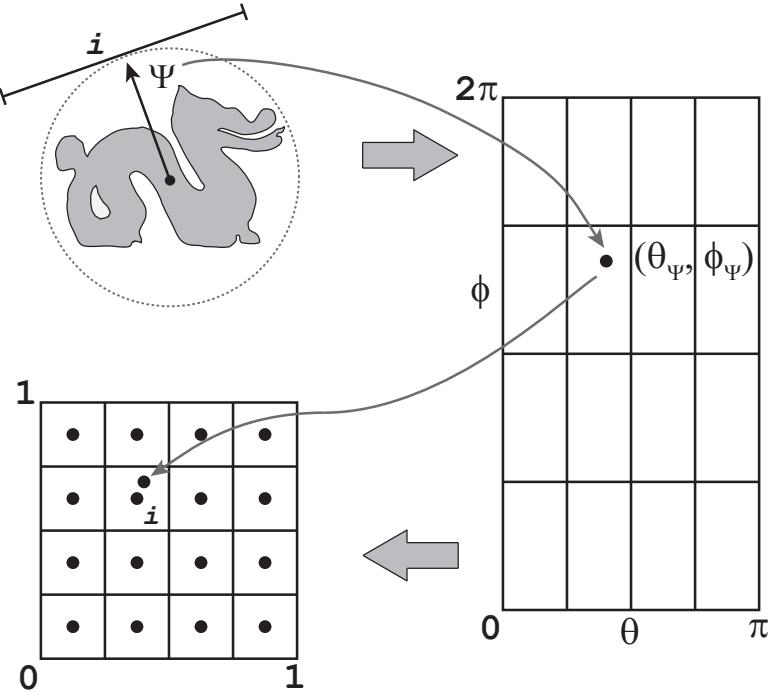


Figuur 4.4: De overeenkomstige dieptemap i wordt opgezocht aan de hand de lichtvector Ψ . Deze dieptemap wordt vervolgens gebruikt om de visibiliteitstest uit te voeren. Links: het punt p ligt buiten het bereik van de dieptemap en wordt dus belicht. Rechts: het punt p ligt binnen het bereik van de dieptemap. Net als bij het klassieke schaduwmap algoritme, wordt de dieptewaarde voor het punt p berekend en vergeleken met de overeenkomstige waarde z_d in de dieptemap. Aangezien $z_d < z_p$ ligt het punt p in de schaduw van het object.

Gegeven een willekeurig punt p in de scène en een willekeurig punt q op het oppervlak van de lichtbron. De lichtvector $\Psi = q - p$ beschrijft de richting waar het licht vandaan komt. Aan de hand van deze vector wordt de overeenkomstige dieptemap i opgezocht. De dieptemap die wordt gezocht is namelijk diegene waarvan de kijkrichting gelijk is aan $-\Psi$. Anders gezegd, dit is de dieptemap die in de richting Ψ ligt, ten opzichte van het middelpunt van de omhullende sfeer. Dit wordt geïllustreerd op figuur 4.4.

De lichtvector Ψ wordt vervolgens getransformeerd naar sferische coördinaten $(r_\Psi, \theta_\Psi, \phi_\Psi)$. Door θ_Ψ en ϕ_Ψ te delen door respectievelijk π en 2π wordt een punt binnen het domein $[0, 1]^2$ bekomen. Aangezien elke dieptemap overeenkomt met een specifiek punt binnen deze ruimte, moet er enkel nog worden gezocht naar de dieptemap die het dichtst bij dit

punt ligt. Figuur 4.5 illustreert dit. Merk op dat deze procedure het omgekeerde is van wat in sectie 4.2 werd gedaan om de positie van een dieptemap te bepalen.



Figuur 4.5: De lichtvector Ψ beschrijft de dieptemap die wordt gezocht. De lichtvector Ψ wordt eerst getransformeerd naar sferische coördinaten($r_\Psi, \theta_\Psi, \phi_\Psi$). Met behulp van θ_Ψ en ϕ_Ψ kan de overeenkomstige dieptemap worden gevonden (zie tekst).

Eenmaal de dichtste dieptemap werd gekozen kan de visibiliteitstest worden uitgevoerd. Deze test is identiek als bij het klassieke schaduwmap algoritme. Voor elke dieptemap wordt ook de bijhorende transformatiematrix \mathbf{M}_i bijgehouden. Aan de hand van deze matrix kan de positie van het punt p worden bepaald vanuit het standpunt van de geselecteerde dieptemap: $\mathbf{M}_i p = p_i$. De x - en y -coördinaten van het punt p_i beschrijven de positie (pixel) op de dieptemap waarmee het punt p overeenkomt. De dieptewaarde z_d op deze positie wordt ten slotte vergeleken met de z -waarde van het punt p_i (z_p). Indien $z_d < z_p$ dan ligt het punt p in schaduw. In het andere geval wordt het punt p belicht.

Na het berekenen van de visibiliteitstest wordt de radiantie in het punt p bepaald. Om tot zachte schaduwen te komen, dienen uiteraard meerdere visibiliteitstesten te worden uitgevoerd. Daarbij wordt telkens een willekeurig punt q op de lichtbron genomen. Voor elke combinatie (p, q) wordt dan de radiantie opnieuw berekend. Het gemiddelde van alle radianties vormt dan de uiteindelijke radiantie in het punt p .

Rotaties en Translaties

Tot nu toe werd telkens verondersteld dat bij het berekenen van de schaduwen de positie en oriëntatie van het object dezelfde is als bij het berekenen van de visibiliteitsinformatie. Voor dynamische scènes is dit echter niet praktisch. Coherente schaduwmappen kunnen daarom worden gebruikt in combinatie met een translatie en/of rotatie van het object. Om ervoor te zorgen dat de schaduwen mee worden getransformeerd, moeten de transformatiematrices van de dieptemappen worden aangepast. Voor elke dieptemap i wordt de nieuwe matrix als volgt berekend: $\mathbf{M}'_i = \mathbf{M}_i \mathbf{M}_t^{-1}$ met $i \in [0, N - 1]$ en \mathbf{M}_t de matrix die een translatie en/of rotatie van het object beschrijft.

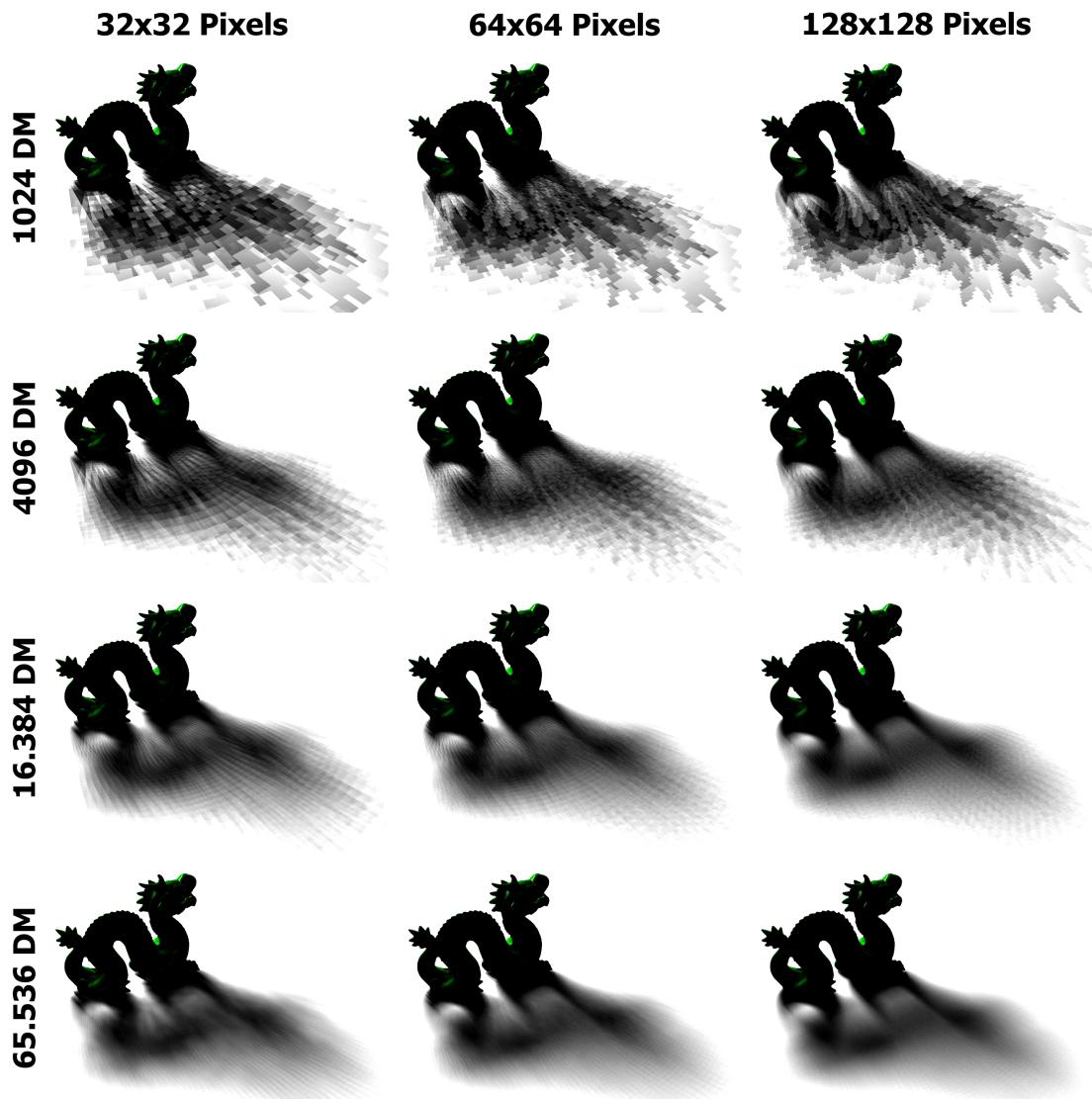
Meerdere Objecten

De coherente schaduwmappen bevatten telkens slechts de visibiliteitsinformatie van een specifiek object. Het resultaat van de visibiliteitstesten slaat dan ook enkel op het feit of dit object het invallend licht op een punt blokkeert of niet. Een typische scène bevat echter meerdere objecten die elk op zich het invallend licht op een punt kunnen blokkeren. Er dienen dus meerdere coherente schaduwmappen te worden geconstrueerd, namelijk één voor elk object. Om te bepalen of een punt dan in schaduw ligt of niet, volstaat het om de visibiliteitstest uit te voeren voor elk object.

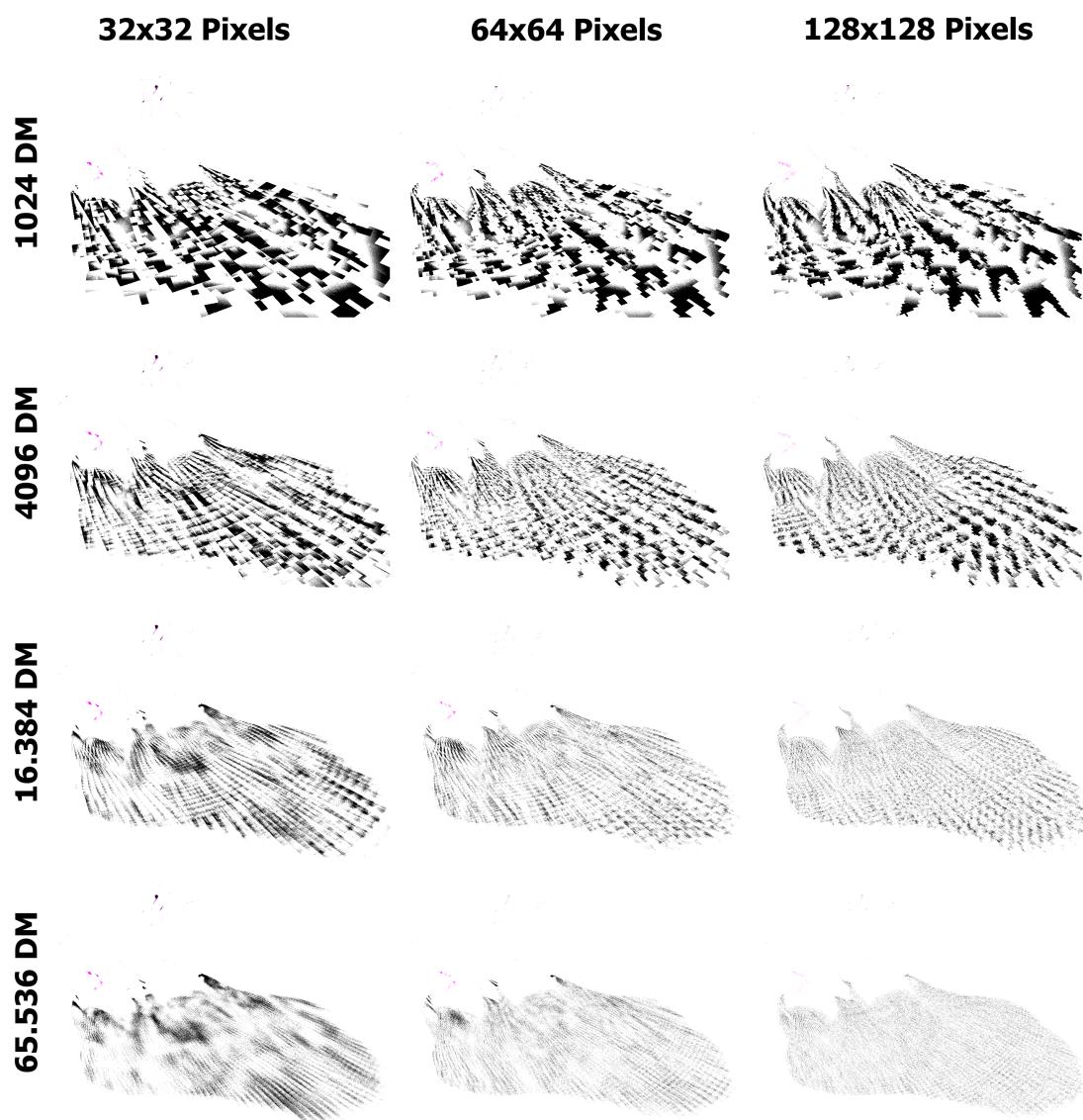
4.4 Resultaten

In deze sectie worden de resultaten van coherente schaduwmappen besproken. Figuur 4.6 toont enkele schaduwen die werden berekend met behulp van coherente schaduwmappen. Daarbij werden verscheidene configuraties van resolutie en aantal dieptemappen tegenover elkaar geplaatst. Op deze manier is de invloed van elke parameter duidelijk zichtbaar. Figuur 4.7 toont voor diezelfde configuraties de fouten die worden gemaakt ten opzichte van een referentie visualisatie. Deze referentie werd berekend met behulp van ray tracing.

Uit de resultaten kan worden afgeleid dat de resolutie belangrijk is om de geometrische details van een object te beschrijven. Voor lage resoluties zijn de dieptemappen onnauwkeurig en bevatten ze enkel een ruwe beschrijving van het oppervlak. De keuze voor een bepaalde resolutie van de dieptemappen wordt dus bepaald door de geometrische structuur van het object. Naargelang het object minder detailinformatie bevat, kan een lagere resolutie worden gebruikt. Naast de resolutie speelt ook het aantal dieptemappen een rol. Uit de figuren is af te leiden dat de kwaliteit van de schaduwen voornamelijk wordt bepaald door het aantal dieptemappen. Zo kunnen zelfs voor lage resoluties aanvaardbare resultaten worden bekomen, zolang het aantal dieptemappen hoog ligt. Indien te weinig dieptemappen worden geconstrueerd, dan zijn de overgangen tussen de verschillende dieptemappen zichtbaar op de geprojecteerde schaduwen. Dit komt omdat voor gelijkaardige richtingen van invallend licht, steeds dezelfde dieptemap wordt gekozen. Daardoor ontstaan gebieden waarvoor het resultaat van de visibiliteitstest telkens



Figuur 4.6: Visualisaties voor verschillende configuraties van coherente schaduwmappen. Van links naar rechts: Een toenemende resolutie betekent dat meer details kunnen worden weergegeven. Dit effect is vooral zichtbaar voor een laag aantal dieptemappen. De stekels op het hoofd van de Stanford draak bijvoorbeeld zijn niet herkenbaar op dieptemappen met een resolutie van 32×32 pixels. Van boven naar onder: Het aantal dieptemappen heeft een sterke invloed op de kwaliteit van de schaduwen. Voor een te laag aantal zijn de individuele dieptemappen nog zichtbaar. Het is pas vanaf 16384 dieptemappen dat dit effect langzaamaan verdwijnt.



Figuur 4.7: Visualisaties van de fouten ($\times 10$) voor verschillende configuraties van coherente schaduwmappen. Naarmate het aantal dieptemappen en de resolutie toenemen, daalt het aantal fouten. Een hoger aantal dieptemappen heeft als voornaamste gevolg dat de overgangen tussen de individuele dieptemappen minder zichtbaar worden. Een hogere resolutie doet het aantal fouten over het gehele oppervlak van de schaduw dalen. Voor lage resoluties blijken er gebieden te zijn waar opvallend meer fouten voorkomen. Voor 32×32 pixels is dit onder andere het gebied waar de schaduw van het hoofd van de draak zich bevindt. Deze fouten zijn het gevolg van visibiliteitsinformatie die niet nauwkeurig genoeg is.

wordt bepaald door dezelfde dieptemap. Deze gebieden zijn ook te zien op de figuur. Naarmate het aantal dieptemappen toeneemt, worden deze gebieden kleiner. Voor de Stanford draak [CL96] zijn bijvoorbeeld minstens 16384 dieptemappen om de zichtbare overgangen tussen de individuele schaduwmappen te vermijden.

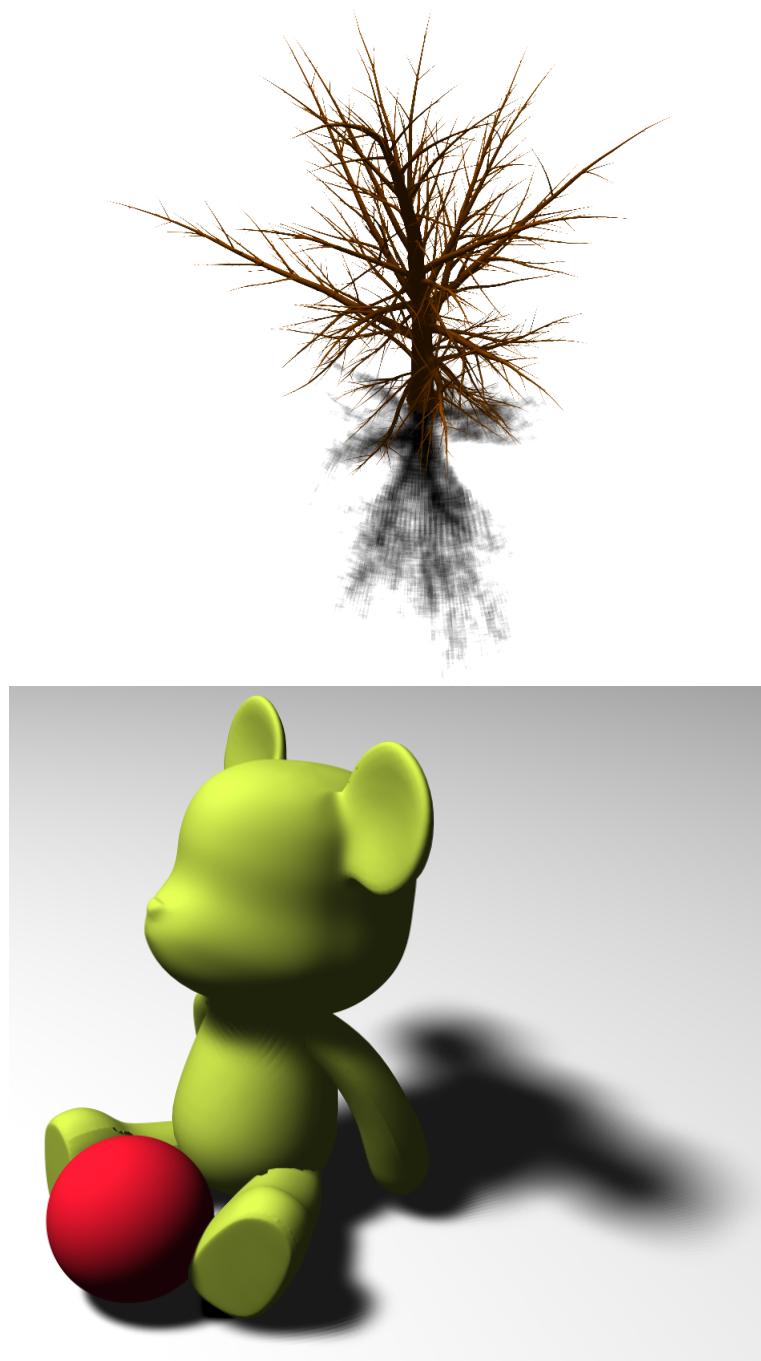
Figuur 4.8 toont het resultaat van coherente schaduwmappen voor twee verschillende scènes. In de eerste scène wordt een boom weergegeven. Een boom bevat zeer veel fijne structuren, waardoor de resolutie een belangrijke rol speelt op de kwaliteit. Op de figuur is te zien dat de schaduwen die worden bekomen, blokkerig en onnauwkeurig zijn. Dit is het gevolg van een te lage resolutie voor dit object. De dieptemappen bevatten te weinig informatie om de schaduwen gedetailleerd te kunnen weergeven. Ditzelfde effect is ook te zien op figuur 4.6 voor een resolutie van 32×32 pixels en 4096 dieptemappen. De tweede scène toont een situatie waarin met coherente schaduwmappen, realistische schaduwen worden bekomen.

4.4.1 Percentage Closer Filtering

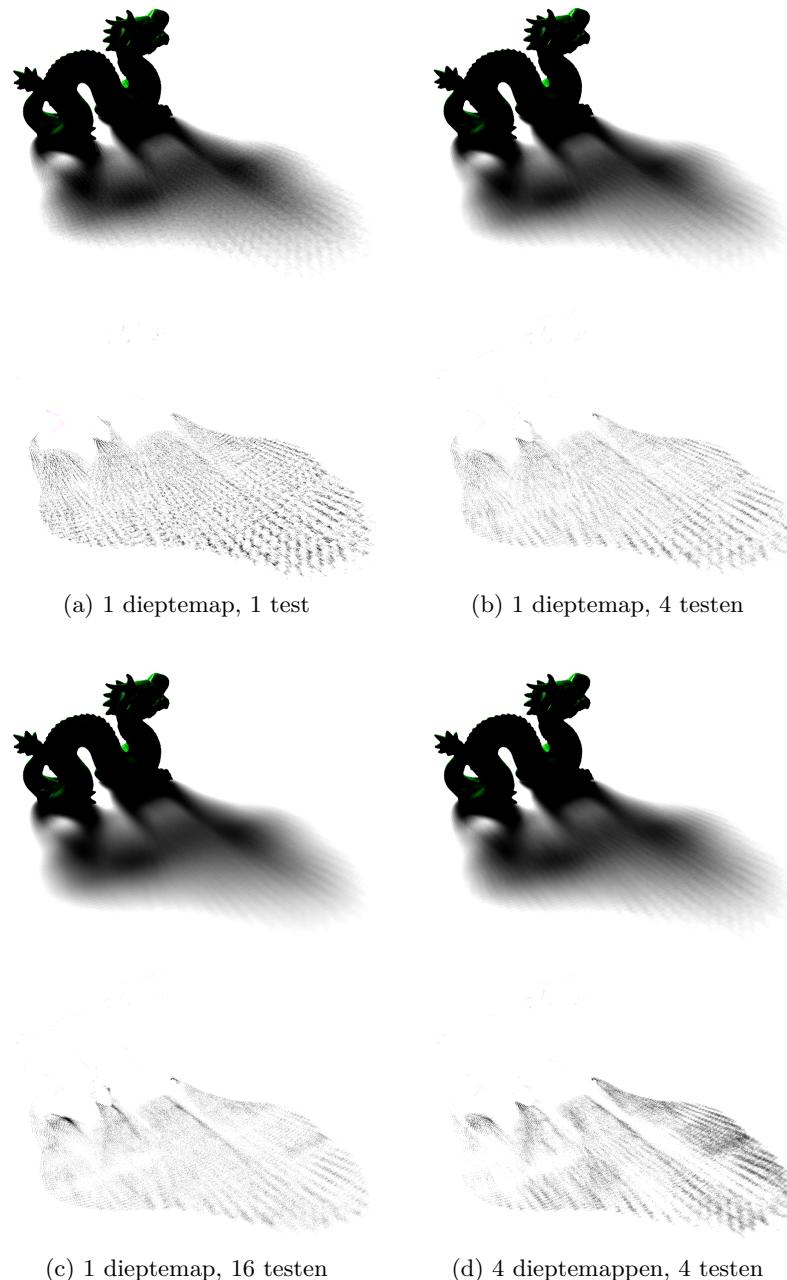
Coherente schaduwmappen kunnen ook worden gecombineerd met percentage closer filtering. Deze techniek werd besproken in hoofdstuk 3. Percentage closer filtering wordt gebruikt om het resultaat van meerdere visibiliteitstesten te combineren. Op deze manier wordt een betere benadering bekomen van de visibiliteitsinformatie in een punt.

Figuur 4.9 toont enkele resultaten voor deze techniek. In het geval dat geen percentage closer filtering wordt gebruikt, wordt slechts één enkele visibiliteitstest uitgevoerd. Dit komt overeen met de visibiliteitstest die werd besproken in de vorige sectie. Bij percentage closer filtering worden meerdere visibiliteitstesten uitgevoerd. Om dit te doen worden de naburige dieptewaarden van een pixel op de dieptemap gebruikt. Naast het uitvoeren van visibiliteitstesten met dieptewaarden uit één en dezelfde dieptemap, kunnen ook visibiliteitstesten met naburige dieptemappen worden uitgevoerd. De dieptemappen die hiervoor worden gebruikt, zijn deze die het dichtst aansluiten bij de richting van het invallend licht. Voor elke dieptemap kan vervolgens één of meerdere visibiliteitstesten worden uitgevoerd.

Op de figuren is te zien dat het gebruik van percentage closer filtering een daling van het aantal fouten veroorzaakt. Naarmate meer testen worden genomen binnen dezelfde dieptemap, daalt het aantal fouten. Aangezien over meerdere dieptemappen wordt beschikt, kunnen ook visibiliteitstesten worden uitgevoerd op andere, dichtbij gelegen, dieptemappen. Zo kunnen bijvoorbeeld vier visibiliteitstesten worden uitgevoerd voor elk van de vier meest nabij gelegen dieptemappen. Er worden dus zestien visibiliteitstesten uitgevoerd. Uit de resultaten blijkt echter dat deze methode minder efficiënt is dan het nemen van zestien visibiliteitstesten binnen dezelfde dieptemap.



Figuur 4.8: Coherente schaduwassen voor twee verschillende scènes. Telkens werden voor elk object in de scène 36864 dieptemappen geconstrueerd met een resolutie van 128×128 pixels. Boven: visualisatie van een boom. De schaduwen die worden bekomen zijn blokkerig en ongedetailleerd. Dit is het gevolg van een te lage resolutie voor dit object. Ditzelfde effect is ook te zien op figuur 4.6 voor een lage resolutie van de dieptemappen. Onder: realistische schaduwen voor een knuffelbeer met bal. Voor beide objecten werden coherente schaduwassen geconstrueerd. De objecten hebben een zachtverlopend oppervlak, waardoor de kwaliteit van de schaduwen hoger ligt dan bij de boom.



Figuur 4.9: Coherente schaduwassen kunnen worden gecombineerd met percentage closer filtering. Door meerdere visibiliteitstesten uit te voeren, daalt het aantal fouten. Deze resultaten werden bekomen voor 16384 dieptemappen met een resolutie van 128×128 pixels. De fouten zijn het laagst indien 16 visibiliteitstesten worden uitgevoerd voor één dieptemap.

4.5 Conclusie

Coherente schaduwmappen laten toe om realistische schaduwen te visualiseren, door enkel gebruik te maken van dieptemappen. Door dieptemappen te construeren op de omhullende sfeer van een object, wordt de volledige visibiliteitsinformatie van een object beschreven. Tijdens het visualiseren van de scène kunnen de coherente schaduwmappen worden gebruikt om de visibiliteitstest op een performante manier uit te voeren.

In sectie 4.4 werden de resultaten van coherente schaduwmappen besproken. Daarbij werd gezien dat de kwaliteit van de schaduwen afhankelijk is van de resolutie en het aantal dieptemappen dat wordt gebruikt. De resolutie is belangrijk om de detailinformatie van een object te kunnen weergeven. Voor een object zoals een boom, waarvan de geometrie uit fijne structuren bestaat, is deze resolutie zeer belangrijk. Naarmate een object over een zachter verlopend oppervlak beschikt, daalt de nood aan een hoge resolutie van de dieptemappen.

Ook belangrijk is het aantal dieptemappen waarvoor de coherente schaduwmappen worden opgesteld. Deze dieptemappen zijn belangrijk om de visibiliteitsinformatie van een object vanuit alle mogelijke richtingen te kunnen beschrijven. Indien te weinig dieptemappen worden gebruikt, dan zijn de individuele dieptemappen zichtbaar in de schaduwen.

Om de kwaliteit van de schaduwen te verbeteren kan ook nog percentage closer filtering worden toegepast. Daardoor wordt een nauwkeuriger resultaat bekomen.

Coherente schaduwmappen hebben echter ook enkele beperkingen. Vooreerst is deze techniek enkel toepasbaar voor rigide objecten. De animaties van elk object zijn dan ook beperkt tot translaties en rotaties. Coherente schaduwmappen bevatten ook telkens slechts de visibiliteitsinformatie van één object in een scène. Indien rekening moet worden gehouden met schaduwen die afkomstig zijn van andere objecten, dan moeten voor meerdere objecten in de scène coherente schaduwmappen worden geconstrueerd. Aangezien uit de resultaten blijkt dat er veel dieptemappen nodig zijn om tot realistische schaduwen te komen, is het belangrijk om de coherente schaduwmappen zo compact mogelijk te kunnen voorstellen. In het volgende hoofdstuk worden enkele compressietechnieken besproken die hiervoor zorgen.

Hoofdstuk 5

Compressie

De dieptemappen die in het vorige hoofdstuk werden geconstrueerd, nemen veel ruimte in beslag. Voor praktische toepassingen zou het handig zijn mochten deze dieptemappen kunnen worden gecomprimeerd. Een mogelijke manier om dit te doen is door de coherentie tussen deze dieptemappen uit te buiten. Daardoor kan een compacte representatie worden bekomen. Dit is het idee achter coherente schaduwmappen.

In dit hoofdstuk worden verschillende compressietechnieken bestudeerd. Deze technieken zijn allemaal gebaseerd op een eigenschap die toelaat om dieptemappen op een foutloze manier te comprimeren. Deze compressie-eigenschap wordt besproken in sectie 5.1. Daarna wordt in sectie 5.2 de standaard compressietechniek van coherente schaduwmappen uitgelegd, namelijk per-pixel compressie. De resultaten van deze compressie worden besproken in sectie 5.3. Daarbij wordt de invloed van verschillende parameters op de compressiefactor onderzocht.

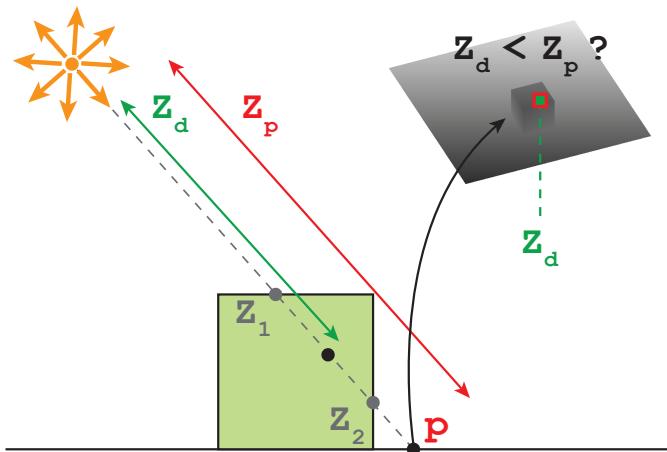
Naast per-pixel compressie zijn ook andere compressietechnieken mogelijk. Een eerste alternatief dat wordt onderzocht is per-map compressie. Daarbij wordt elke dieptemap afzonderlijk gecomprimeerd. Dit wordt besproken in sectie 5.4. Een tweede techniek maakt gebruik van clustering. Daarbij worden dieptemappen die een sterke gelijkenis hebben samengenomen en voorgesteld door een gemeenschappelijke dieptemap. Het achterliggende idee is dat daardoor de coherentie tussen de verschillende dieptemappen wordt vergroot. Dit wordt besproken in sectie 5.5. In sectie 5.6 worden twee compressietechnieken onderzocht die gebaseerd zijn op de multidimensionale eigenschappen van de Hilbert curve. Met behulp van deze curve kunnen individuele pixels van verschillende dieptemappen in een coherente volgorde worden gecomprimeerd. Ten slotte wordt in sectie 5.7 onderzocht of per-pixel compressie kan worden gecombineerd met wavelet compressie.

5.1 Introductie

Bij het comprimeren van dieptemappen is het van groot belang dat de compressietechniek foutloos is. Indien dit niet het geval is, betekent dit dat er wijzigingen worden aangebracht aan de dieptewaarden. Bijgevolg wordt het resultaat van de visibiliteitstest beïnvloed, waardoor fouten worden geïntroduceerd in de gegenereerde schaduwen. Om dit te vermijden wordt beroep gedaan op een eigenschap van duale dieptelagen. Met behulp van deze eigenschap is het mogelijk om dieptemappen te comprimeren, zonder dat het resultaat van de visibiliteitstesten wordt beïnvloed.

5.1.1 Foutloze Compressie

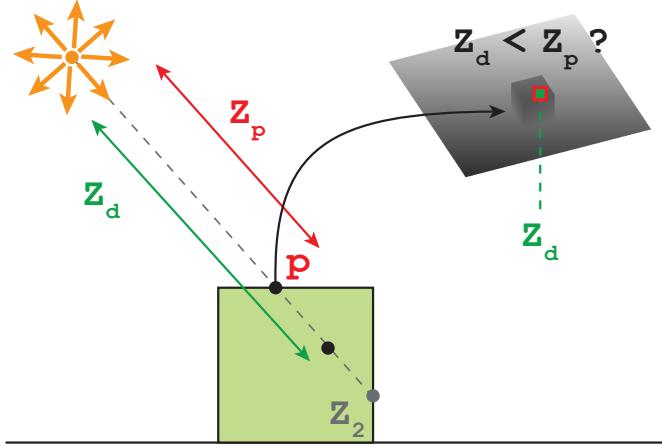
Duale dieptelagen vormen de basis voor de meeste compressietechnieken die in dit hoofdstuk aan bod komen. Duale dieptelagen werden in hoofdstuk 3 besproken om het probleem van zelfbeschaduwing op te lossen. Daarbij werd voor elke dieptewaarde een bias gekozen in functie van de geometrie. De eigenschap waarop deze techniek is gebaseerd, wordt geïllustreerd op figuur 5.1 en 5.2.



Figuur 5.1: Visibiliteitstest voor een punt in schaduw. Het punt p wordt geprojecteerd naar het standpunt van de lichtbron. De dieptewaarde z_p wordt dan vergeleken met de dieptewaarde z_d die is opgeslagen in de dieptemap. De waarde van z_d kan vrij worden gekozen tussen z_1 en z_2 . Zolang $z_d \in [z_1, z_2]$ blijft de visibiliteitstest correct, aangezien z_d altijd kleiner is dan z_p . Bijgevolg ligt het punt p in schaduw.

Voor het correct uitvoeren van de visibiliteitstest mag elke dieptewaarde tussen het dichtste punt z_1 en het tweede dichtste punt z_2 worden gebruikt. Met behulp van deze eigenschap kunnen dieptemappen worden gecomprimeerd. Neem als voorbeeld twee pixels i en j , waarvoor de z_1 - en z_2 -waarden zijn gekend. In plaats van voor elke pixel een aparte dieptewaarde op te slaan, kan een gemeenschappelijke dieptewaarde z_g worden bepaald. Deze waarde kan vrij worden gekozen uit de doorsnede van beide pixels: $z_g \in [z_{1,i}, z_{2,i}] \cup [z_{1,j}, z_{2,j}]$. Indien de doorsnede niet leeg is, kan dus een gemeenschap-

pelijke dieptewaarde z_g worden bepaald, die voor beide pixels kan worden gebruikt. Daardoor moet maar één dieptewaarde worden opgeslagen in plaats van twee dieptewaarden. Het spreekt voor zich dat deze compressietechniek ook kan worden toegepast op meer dan twee pixels. Op deze manier kunnen dus volledige dieptemappen worden gecomprimeerd, zonder dat er fouten worden geïntroduceerd.

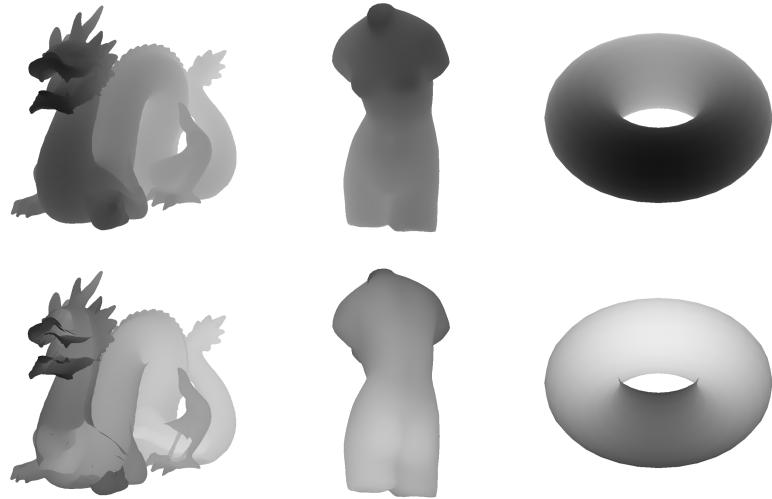


Figuur 5.2: Visibiliteitstest voor een punt dat niet in schaduw ligt. Het punt p wordt geprojecteerd naar het standpunt van de lichtbron. De dieptewaarde z_p wordt dan vergeleken met de dieptewaarde z_d die is opgeslagen in de dieptemap. De waarde van z_d kan vrij worden gekozen tussen $z_1 = z_p$ en z_2 . Zolang $z_d \in [z_1, z_2]$ blijft de visibiliteitstest correct, aangezien z_d altijd groter is dan z_p of gelijk is aan z_p . Bijgevolg ligt het punt p niet in schaduw.

5.1.2 Duale Dieptemappen

Om gebruik te kunnen maken van deze compressie-eigenschap moeten de z_2 -waarden natuurlijk gekend zijn. Daarvoor worden duale dieptemappen gebruikt. Een klassieke dieptemap bevat enkel de dieptewaarden van de dichtste punten in de scène, de z_1 -waarden. Een duale dieptemap bevat de dieptewaarden van de tweede dichtste punten, de z_2 -waarden. Figuur 5.3 illustreert de verschillen tussen deze dieptemappen.

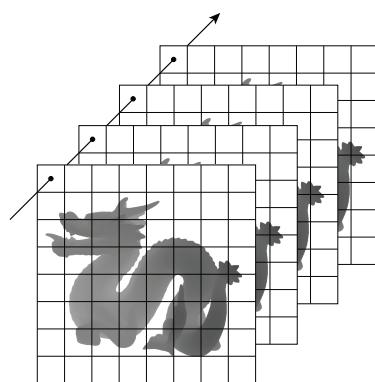
In het vorige hoofdstuk werd al vermeld dat de dieptemappen worden geconstrueerd op verschillende posities rondom een object. Tot nu toe werd ervan uitgegaan dat op elke positie slechts één klassieke dieptemap van het object werd geconstrueerd. Aangezien er voor de compressie van dieptemappen over zowel z_1 - als z_2 -waarden moet worden beschikt, zullen voor elke positie op de omhullende sfeer twee orthografische dieptemappen van het object worden geconstrueerd, namelijk een klassieke dieptemap en een duale dieptemap. In het verdere verloop van deze tekst wordt met de termen “dieptemappen” en “dieptewaarden” telkens verwezen naar de combinatie van respectievelijk beide dieptemappen en beide dieptewaarden.



Figuur 5.3: Dieptemappen voor drie objecten: De Stanford draak [CL96], een standbeeld van Venus en een torus. Boven: de klassieke dieptemap bevat de z_1 -waardes. Onder: de duale dieptemap bevat de z_2 -waardes.

5.2 Per-Pixel Compressie

Bij per-pixel compressie worden alle dieptemappen eerst na elkaar geplaatst in een balkvormige structuur. Daarna worden voor een specifieke pixel, alle dieptewaarden op deze positie gecomprimeerd. Dit wordt herhaald voor alle pixels van een dieptemap. De volgorde waarin de dieptewaarden worden gecomprimeerd, wordt bepaald door de volgorde waarin de dieptemappen zijn geordend. Per-pixel compressie is gebaseerd op de eerder vermelde eigenschap van duale dieptelagen. Daardoor kunnen de gecomprimeerde dieptewaarden op een compacte en snel toegankelijke manier worden opgeslagen [RGKM07]. Figuur 5.4 illustreert het algemene idee van deze compressietechniek.

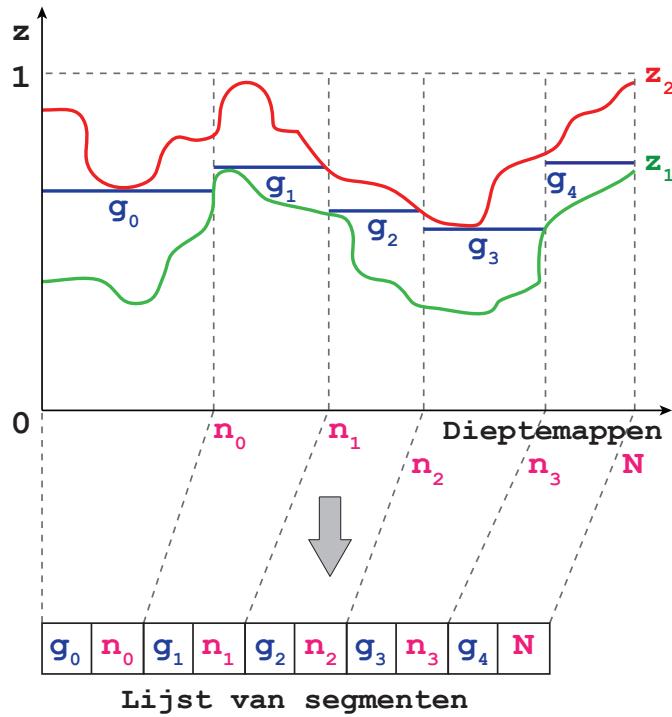


Figuur 5.4: Per-pixel compressie. Dieptemappen worden na elkaar geplaatst en alle dieptewaarden voor een specifieke pixel worden gecomprimeerd.

Bij per-pixel compressie worden telkens de dieptewaarden gecomprimeerd waarvoor hun pixelposities gelijk zijn. Aangezien voor elke pixel de z_1 - en z_2 -waardes gekend zijn, kunnen alle dieptewaarden worden voorgesteld door stuksgewijze constante functies. Daarbij wordt telkens op zoek gegaan naar een index j zodat:

$$\max(z_{1,j-k}, z_{1,j-k+1}, \dots, z_{1,j}) < \min(z_{2,j-k}, z_{2,j-k+1}, \dots, z_{2,j}).$$

Er wordt met andere woorden gezocht naar opeenvolgende groepen van pixels waarvoor het maximum van de z_1 -waardes kleiner is dan het minimum van de z_2 -waardes. Het gemiddelde tussen deze twee grenzen wordt dan als gemeenschappelijke dieptewaarde g_i genomen. Deze dieptewaarden worden samen met de bijhorende index n_i opgeslagen in segmenten (g_i, n_i) . De index n_i houdt bij hoeveel dieptemappen gebruik maken van de overeenkomstige dieptewaarde g_i . Na het comprimeren van alle dieptewaarden op eenzelfde pixelpositie wordt een lijst van segmenten bekomen. Dit wordt geïllustreerd op figuur 5.5. Het volledige proces wordt herhaald voor elke pixelpositie. Voor een resolutie van $M \times M$ pixels resulteert dit in $M \times M$ lijsten van segmenten.

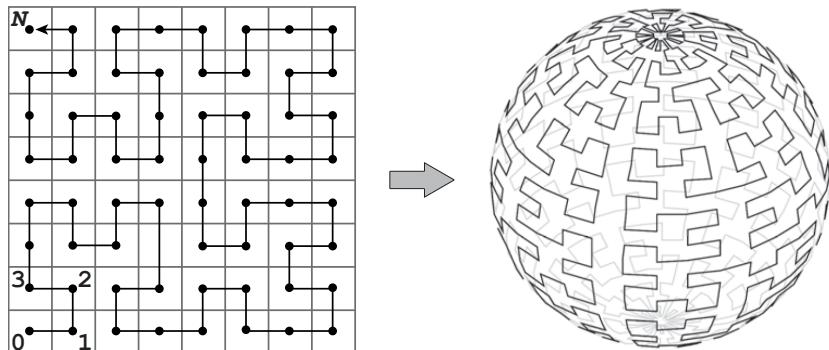


Figuur 5.5: Per-pixel compressie voor een specifieke pixel. Door de z_1 - en z_2 -waardes te vergelijken worden gemeenschappelijke dieptewaarden g_i gevonden. Deze stellen stuksgewijze constante functies voor. De gemeenschappelijke dieptewaarden worden opgeslagen in segmenten (g_i, n_i) . De index n_i houdt bij hoeveel dieptemappen gebruik maken van de overeenkomstige dieptewaarde g_i .

Om de dieptemappen van een object te comprimeren wordt gebruik gemaakt van z_2 -waardes. Voor sommige objecten zijn deze dieptewaardes echter niet gekend omdat er eenvoudigweg geen tweede dichtste punten zijn. Een voorbeeld van zo'n object is een vlak. Daarvan kunnen enkel de z_1 -waardes worden berekend. Dit wil dus zeggen dat per-pixel compressie enkel kan worden toegepast indien het om een gesloten object gaat. Voor niet-gesloten objecten bestaat echter een oplossing. Om de z_2 -waardes te berekenen kan bijvoorbeeld een constante factor bij de z_1 -waardes worden opgeteld. Een andere oplossing kan zijn om deze onbekende z_2 -waardes te beperken tot de omhullende sfeer. Voor beide oplossingen geldt dat er tijdens het visualiseren geen andere objecten binnen de omhullende sfeer mogen liggen. In dat geval is het immers mogelijk dat de visibiliteitstest niet meer correct verloopt.

Bepalen van een Coherente Sequentie

De volgorde waarin de dieptemappen worden gecomprimeerd is van groot belang voor de efficiëntie van de compressie. Daarom is een coherente sequentie van de dieptemappen aangeraden. Met coherent wordt bedoeld dat het verschil in dieptewaardes tussen twee opeenvolgende dieptemappen minimaal is. Dieptemappen die naast elkaar liggen in deze sequentie bevatten dus gelijkaardige z_1 - en z_2 -waardes.

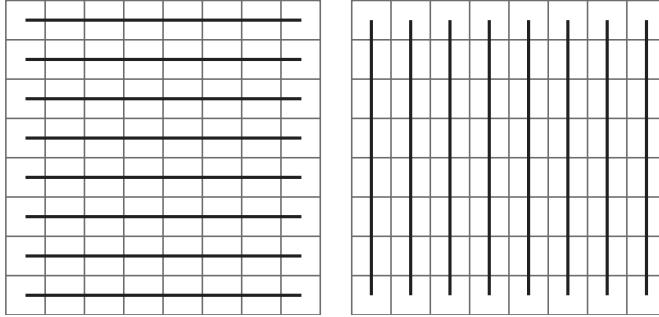


Figuur 5.6: Aangezien elke dieptemap op een unieke wijze wordt voorgesteld door een punt in de tweedimensionale ruimte, kan een tweedimensionale Hilbert curve worden gebruikt om een coherente sequentie van de dieptemappen te bepalen. De coherentie tussen de punten wordt bewaard bij het transformeren van de punten naar posities op de omhullende sfeer. Punten die dicht bij elkaar liggen in de tweedimensionale ruimte, liggen dus ook dicht bij elkaar op de omhullende sfeer [RGKM07].

Om een coherente volgorde te bepalen worden ruimtevullende curves gebruikt. In het vorige hoofdstuk werd gezien dat elke dieptemap op een unieke wijze kan worden voorgesteld door een punt in de tweedimensionale ruimte $[0, 1]^2$. Met behulp van een ruimtevullende curve kan een coherente sequentie worden opgesteld voor alle punten binnen deze ruimte. Een belangrijke opmerking hierbij is dat de coherentie van deze sequentie wordt bewaard bij het transformeren van de punten naar driedimensionale punten op

de omhullende sfeer. Punten die dicht bij elkaar liggen in de tweedimensionale ruimte, liggen namelijk ook dicht bij elkaar op de omhullende sfeer. Voor twee opeenvolgende dieptemappen in de sequentie heeft dit als gevolg dat de z_1 - en z_2 -waardes van beide dieptemappen zeer gelijkaardig zullen zijn. Figuur 5.6 illustreert dit voor een Hilbert curve.

Naast de Hilbert curve en de zigzagende curves die werden besproken in hoofdstuk 2, kunnen ook andere curves worden gebruikt, zoals bijvoorbeeld een lineaire curve. Figuur 5.7 toont deze curve.



Figuur 5.7: Een tweedimensionale lineaire curve. Links: horizontaal. Rechts: verticaal.

De volgorde waarin de dieptemappen worden geordend in de coherente sequentie, heeft een rechtstreeks invloed op de efficiëntie van per-pixel compressie. Hoe coherenter de sequentie, hoe groter de kans dat een gemeenschappelijke dieptewaarde kan worden gevonden voor meerdere pixels. In dat geval zullen er minder segmenten nodig zijn om alle pixels voor te stellen, wat een positief effect heeft op de compressiefactor. In het geval dat de sequentie niet coheren is, zullen er sterke verschillen zijn tussen de dieptewaarden van opeenvolgende dieptemappen. Daardoor wordt het moeilijker om gemeenschappelijke dieptewaarden te vinden. Er zullen dan meer segmenten nodig zijn, waardoor de compressiefactor zal dalen.

Decompressie

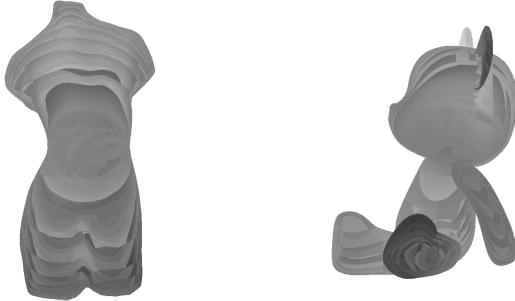
Bij het visualiseren van een scène wordt gebruikt gemaakt van de gecomprimeerde dieptewaarden om het resultaat van de visibiliteitstest te bepalen. Daarvoor moeten de dieptewaarden echter eerst worden gedecompresseerd.

In hoofdstuk 4 werd al uitgelegd hoe de dieptemap i wordt gevonden die het best overeenkomt met de richting van het invallend licht op een punt p . Aan de hand van deze dieptemap en de bijhorende transformatiematrix \mathbf{M}_i kan de gewenste dieptewaarde worden opgehaald. Bij de gecomprimeerde dieptemappen is deze dieptewaarde echter niet rechtstreeks beschikbaar. Eerst moet de lijst met segmenten worden teruggevonden die deze dieptewaarde bevat. Daarvoor wordt een opzoekingstabell gebruikt. Aan de hand van een opgegeven pixelpositie kan dan de overeenkomstige lijst met segmenten

worden opgezocht. De pixelpositie van de dieptewaarde wordt bepaald door de x - en y -coördinaten van het getransformeerde punt $p' = \mathbf{M}_i p$.

Daarna moet in de lijst met segmenten de gewenste dieptewaarde worden opgezocht. Om het segment dat deze dieptewaarde bevat terug te vinden, wordt gebruik gemaakt van de index i van de dieptemap. Deze index is dezelfde als de positie van de dieptemap in de coherente sequentie. Het segment dat wordt gezocht is datgene met de kleinste index n_j en $n_j \geq i$. Om dit segment te vinden wordt een binair zoekalgoritme gebruikt. Op deze manier kan op een performante manier de gewenste dieptewaarde worden opgehaald.

Figuur 5.8 toont enkele gedecomprimeerde dieptemappen. Op deze dieptemappen is duidelijk de invloed van andere dieptemappen te zien. Dit is het gevolg van gemeenschappelijke dieptewaardes te gebruiken.



Figuur 5.8: Voorbeelden van een gedecomprimeerde dieptemap. Voor elke pixel wordt de overeenkomstige dieptewaarde opgezocht in de coherente schaduwassen. Als resultaat wordt een dieptemap bekomen waarop de invloed van andere dieptemappen zichtbaar is. Dit vertaalt zich in herhalingen van andere dieptemappen. Links: een standbeeld van Venus. Rechts: een knuffelbeer.

5.3 Resultaten Per-Pixel Compressie

Bij het construeren van de coherente schaduwassen kunnen verschillende parameters worden ingesteld, zoals de resolutie en het aantal dieptemappen. De invloed van deze parameters op de compressiefactor is echter niet altijd even duidelijk.

In deze sectie worden de resultaten van per-pixel compressie besproken. Er wordt onderzocht wat de invloed is van volgende parameters op de compressiefactor:

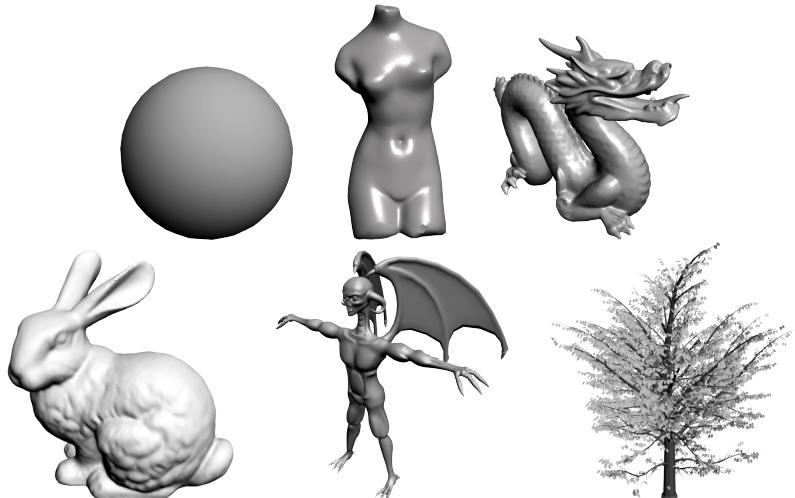
- de geometrie van het object (sectie 5.3.1);
- de ruimtevullende curve die de coherente sequentie bepaalt (sectie 5.3.2);
- de resolutie van de coherente schaduwassen (sectie 5.3.3);

- een (niet-)uniforme verdeling van de dieptemappen (sectie 5.3.4);
- de oriëntatie van de omhullende sfeer (sectie 5.3.4).

5.3.1 Invloed van Geometrie

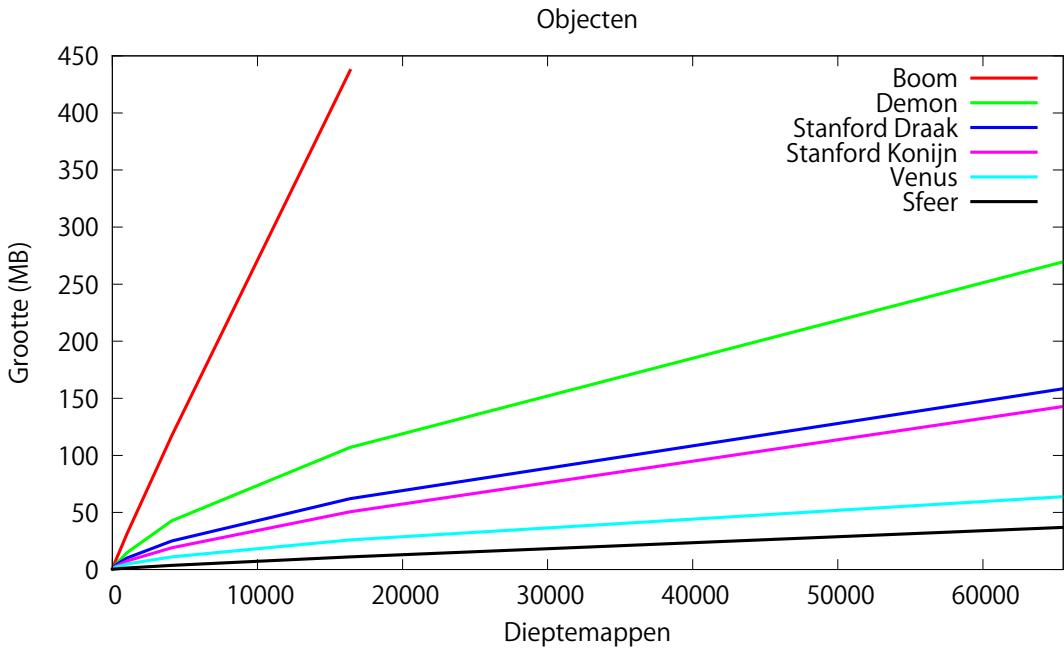
Coherente schaduwmannen maken gebruik van coherentie om tot een compacte representatie te komen. Om deze coherentie uit te buiten wordt een coherente sequentie van de dieptemappen opgesteld. Deze dieptemappen zijn eigenlijk visualisaties van een object. Bijgevolg wordt de efficiëntie van de per-pixel compressie ook beïnvloed door de geometrie van dit object. Aangezien er een grote variëteit aan objecten bestaat is het interessant om na te gaan wat de invloed is van geometrie op de compressiefactor.

Voor het berekenen van de resultaten worden verschillende objecten bestudeerd. Figuur 5.9 toont deze objecten. De coherente schaduwmannen hebben telkens een resolutie van 128×128 pixels en werden uniform verdeeld over het oppervlak van de omhullende sfeer. Voor het bepalen van de coherente sequentie werd een horizontale zigzag curve gebruikt. De resultaten van de per-pixel compressie zijn terug te vinden op figuur 5.10 en in de tabel 5.1. De resultaten voor andere objecten zijn terug te vinden in appendix A.



Figuur 5.9: Voor het construeren van coherente schaduwmannen werden zes objecten gebruikt. Van links naar rechts en van boven naar onder: een sfeer, een standbeeld van Venus, de Stanford draak [CL96], het Stanford konijn [TL94], een demon en een boom. Alle objecten zijn opgebouwd uit driehoeken, voor de sfeer wil dit zeggen dat slechts een benadering van het echte oppervlak wordt bekomen.

Uit de resultaten kan worden afgeleid dat met een sfeer de hoogste compressiefactor wordt bekomen. Dit is niet verwonderlijk. De dieptemappen die voor een sfeer worden geconstrueerd zijn namelijk identiek aan elkaar. Ongeacht de richting vanwaar je



Figuur 5.10: Resultaten van per-pixel compressie voor verschillende objecten. Op de grafiek is te zien dat met per-pixel compressie een hoge compressiefactor wordt bekomen. Voor objecten met een sterk variërende geometrie of een geometrie met veel fijne structuren, blijkt per-pixel compressie minder efficiënt te zijn. De geometrie van een object heeft dus een invloed op de compressie. De compressiegraad stijgt ook toe naarmate de resolutie of het aantal dieptemappen toeneemt. De reden waarom er geen perfecte coherentie is voor de sfeer, is te wijten aan de geometrische benadering die voor dit object werd gebruikt.

Dieptemappen	Ongecomprimeerd	Gecomprimeerd	Ratio	Tijd
16	2 MB	0,93 MB	2,1:1	1s
64	8 MB	1,7 MB	2,4:1	1s
256	32 MB	4,1 MB	7,7:1	2s
1.024	128 MB	10 MB	12:1	5s
4.096	512 MB	25 MB	20:1	14s
16.384	2.048 MB	62 MB	33:1	42s
65.536	8.192 MB	158 MB	52:1	183s

Tabel 5.1: Resultaten van per-pixel compressie voor de Stanford draak. Hieruit kan worden afgeleid dat de compressiegraad toeneemt naarmate meer dieptemappen worden gebruikt. In het geval van 65536 dieptemappen betekent dit dat 8 GB aan data wordt voorgesteld door amper 158 MB. Per-pixel compressie is dus een behoorlijk efficiënte compressietechniek.

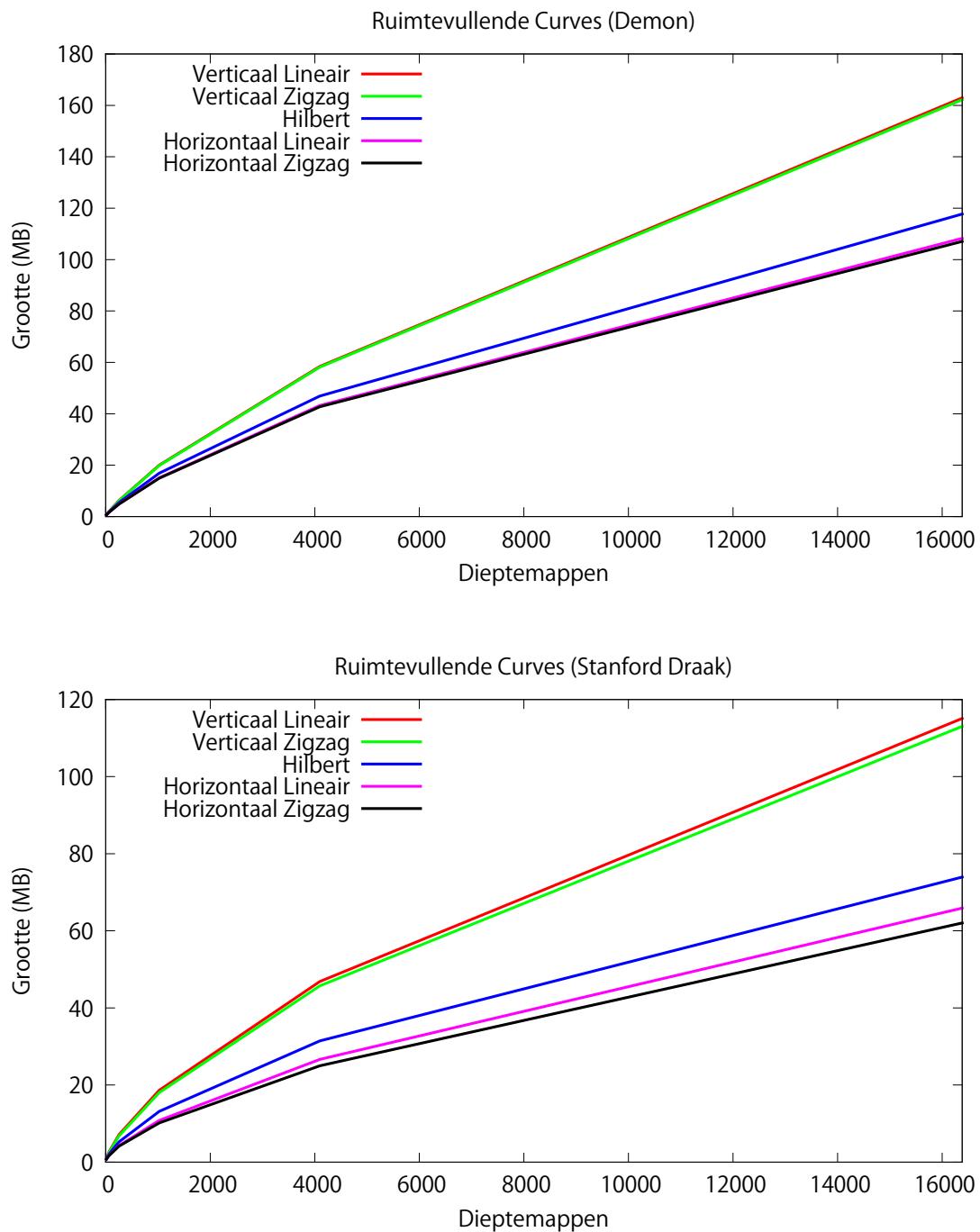
naar een sfeer kijkt, het resulterende beeld is identiek. Dit wil dus zeggen dat de volledige visibiliteitsinformatie van een sfeer kan worden voorgesteld met één enkele dieptemap. Bij het construeren van coherente schaduwassen is dit uiteindelijk niet het geval, aangezien slechts een benadering van het oppervlak van een echte sfeer werd gebruikt. Door deze discretisatie worden voor pixels aan de rand van de sfeer incoherenties geïntroduceerd. Daardoor is de compressie voor de sfeer niet perfect. Een ander object dat blijkbaar veel coherentie bevat is de Venus. Dit is te wijten aan het zachtverlopend oppervlak van het beeld. De Stanford draak en het Stanford konijn kennen daarentegen een iets grilliger oppervlak, daardoor ligt de compressiefactor ook iets lager dan bij de Venus.

De demon en de boom zijn twee voorbeelden van objecten waarvoor de per-pixel compressie minder efficiënt is. Vooral voor de boom ligt de compressiefactor zeer laag. Dit is te wijten aan de vele takken en bladeren die ervoor zorgen dat de geometrie van het object zeer sterk varieert. Daardoor is er weinig coherentie tussen de dieptemappen aanwezig. Daarnaast speelt ook nog een andere factor een rol. Voor zowel de boom als de demon geldt dat de geometrische structuren van het object dun zijn (takken bijvoorbeeld). Bijgevolg is het gebied tussen de z_1 - en z_2 -waarden ook klein. Voor het bepalen van gemeenschappelijke dieptewaarden heeft dit als gevolg dat er minder keuze mogelijk is en dat er dus ook minder pixels worden voorgesteld door eenzelfde gemeenschappelijke waarde. Daardoor stijgt het aantal segmenten en neemt de compressiefactor af.

In tabel 5.1 wordt ook de tijd vermeld die nodig is om de coherente schaduwassen te construeren. Deze tijd omvat zowel het genereren als het comprimeren van de dieptemappen. Deze tijd wordt onder andere beïnvloed door het aantal polygonen van het object en het aantal geconstrueerde dieptemappen. Daarnaast speelt ook de coherentie van het object zelf een rol. Voor een boom zullen bijvoorbeeld veel meer segmenten worden aangemaakt dan voor een sfeer, waardoor ook de constructietijd stijgt.

5.3.2 Invloed van Ruimtevullende Curves

Voor het bepalen van een coherente sequentie van de dieptemappen wordt gebruik gemaakt van een ruimtevullende curve. Deze sequentie speelt een belangrijke rol voor de efficiëntie van de per-pixel compressie. Hoe coherenter de sequentie, hoe hoger de compressiefactor. Voor het bepalen van deze sequentie kunnen verschillende ruimtevullende curves worden gebruikt. In totaal worden vijf ruimtevullende curves bestudeerd: een Hilbert curve, een horizontale en verticale zigzag curve en een horizontale en verticale lineaire curve. De invloed van deze curves werd onderzocht voor twee objecten, namelijk een demon en de Stanford draak. De coherente schaduwassen die werden geconstrueerd, hebben een resolutie van 128×128 pixels en werden uniform verdeeld over het oppervlak van de omhullende sfeer. Figuur 5.11 toont de resultaten voor beide objecten.



Figuur 5.11: Invloed op de compressiefactor voor verschillende ruimtevullende curves. Boven: de resultaten van de demon. Onder: de resultaten van de Stanford draak. Voor beide wordt een gelijkaardig resultaat bekomen. De horizontale zigzag curve geeft telkens het beste resultaat.

Voor beide objecten zijn de resultaten hetzelfde. Met de horizontale zigzag curve wordt de hoogste compressiefactor bekomen, gevolgd door de horizontale lineaire curve. Daarna volgt de Hilbert curve. De resterende (verticale) curves bevinden zich helemaal bovenaan de grafiek en leveren de laagste compressiefactor op.

Een eerste observatie is dat met een horizontaal georiënteerde curve een hogere compressiefactor wordt bekomen dan met een verticaal georiënteerde curve. Dit wil zeggen dat de volgorde waarin de dieptemappen worden gecomprimeerd, minder coherent is voor een curve die verticaal is georiënteerd. De oorzaak hiervan ligt bij de positiebepaling van de dieptemappen op de omhullende sfeer. Daarbij wordt vertrokken van reguliere punten binnen het domein $[0, 1]^2$. Deze worden eerst gescaleerd (al dan niet via een uniforme verdeling) naar punten binnen het domein $[0, \pi] \times [0, 2\pi]$ vooraleer ze worden omgezet naar posities op de omhullende sfeer. Binnen het domein $[0, 1]^2$ is de horizontale en verticale afstand tussen alle punten identiek. Dit is niet langer het geval wanneer de punten worden gescaleerd naar $[0, \pi] \times [0, 2\pi]$. Binnen dit domein is de verticale afstand tussen punten namelijk groter dan de horizontale afstand (zie figuur 4.3 voor een visuele voorstelling hiervan). De relatie tussen deze punten wordt bovendien bewaard bij het transformeren van de punten naar posities op de omhullende sfeer. In het geval van een verticaal georiënteerde curve wil dit zeggen dat de afstand op de omhullende sfeer tussen twee opeenvolgende dieptemappen in de sequentie, groter is dan bij een horizontale curve. Doordat de afstand tussen de dieptemappen groter is, is er ook minder coherentie tussen de dieptewaarden van beide dieptemappen. Dit heeft uiteraard een negatief effect op de compressiefactor.

Een tweede observatie is dat de resultaten voor de lineaire curves niet zo sterk verschillen van de overeenkomstige zigzag curves. Dit heeft te maken met het oppervlak waarop de volgorde van de dieptemappen wordt bepaald. Bij een sfeer komen het begin- en eindpunt van een dimensie namelijk overeen. Daardoor is er slechts een klein verschil tussen een zigzag curve en een lineaire curve op de omhullende sfeer.

Op de grafiek is ook te zien dat de Hilbert curve tussen de horizontale en verticale curves ligt. In sectie 2.4 werd nochtans gezien dat de Hilbert curve de coherentie het best bewaart. Dit blijkt hier echter niet het geval te zijn. De oorzaak ligt opnieuw bij de niet-uniforme scalering van de punten in het domein $[0, 1]^2$ naar de punten in het domein $[0, \pi] \times [0, 2\pi]$. Doordat een Hilbert curve zowel horizontale als verticale bewegingen maakt, wordt op de omhullende sfeer niet telkens de dieptemap gekozen die het best de coherentie bewaart. De horizontale zigzag curve daarentegen maakt optimaal gebruik van de niet-uniforme scalering van de punten. Door verticale bewegingen zoveel mogelijk te beperken wordt telkens de dichtste en meest gelijkaardige dieptemap gekozen. Daardoor wordt met een horizontale zigzag curve de meest coherente sequentie bekomen.

5.3.3 Invloed van Resolutie en Aantal Dieptemappen

De coherente schaduwmappen worden geconstrueerd op een bepaalde resolutie. In sectie 3.2 werd al gezien dat de visibiliteitsinformatie nauwkeuriger wordt naarmate de resolutie toeneemt. Voor de per-pixel compressie worden vier resoluties bestudeerd: 32×32 , 64×64 , 128×128 en 256×256 pixels. De resultaten werden gegenereerd voor de demon en het Stanford konijn. Daarbij werd telkens een uniforme verdeling van de dieptemappen gebruikt. Voor het bepalen van de coherente sequentie werd een horizontale zigzag curve gebruikt. De resultaten voor beide objecten zijn te zien op figuur 5.12. De tabellen A.6 en A.7 bevatten een meer gedetailleerd beeld van deze resultaten.

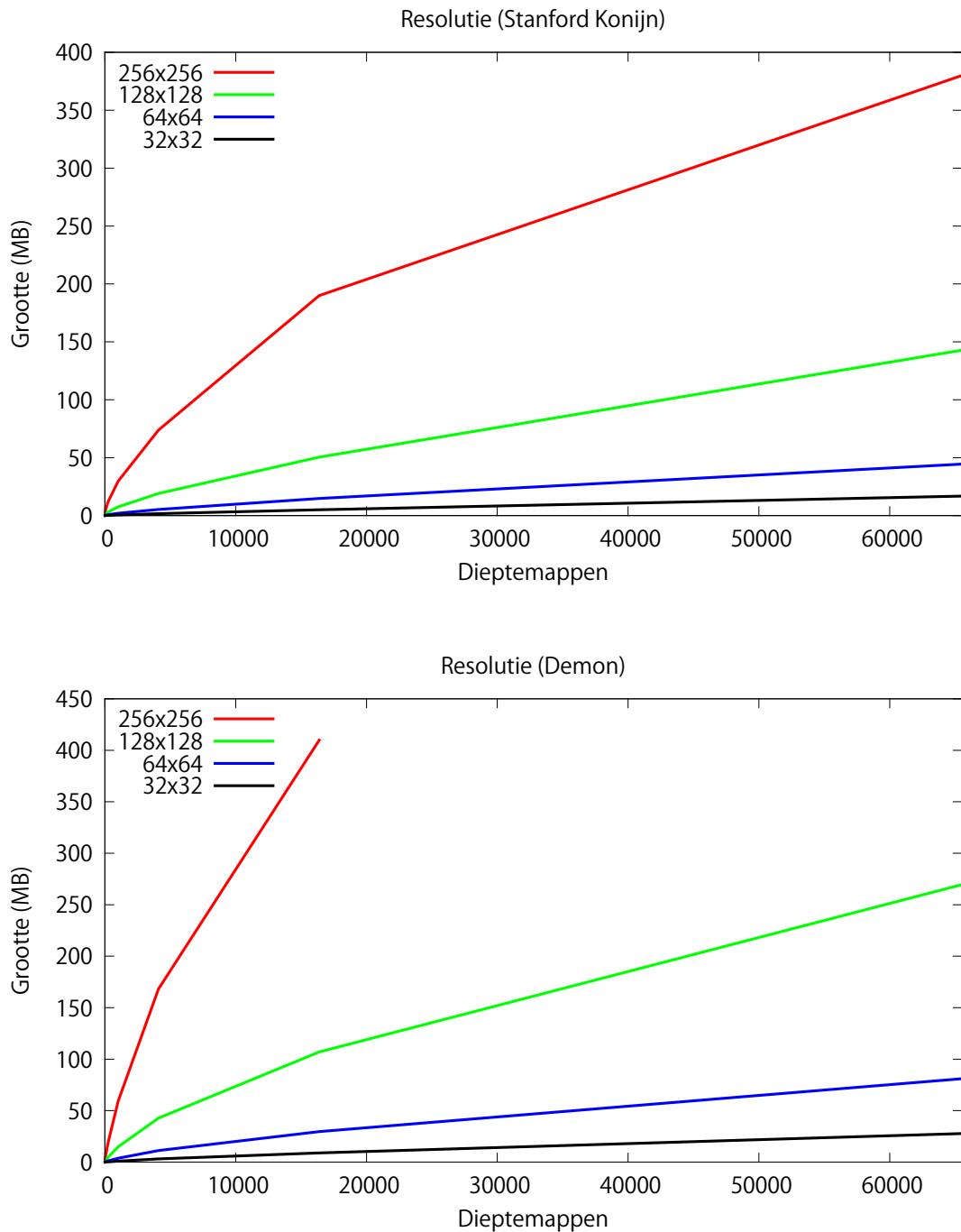
Op de grafieken is te zien dat een hogere resolutie leidt tot een sterke stijging van de bestands grootte. Uit de bijhorende tabellen kan tevens worden afgeleid dat de compressiefactor toeneemt naarmate de resolutie stijgt. De reden waarom de compressiefactor toeneemt naarmate de resolutie stijgt, heeft te maken met de extra informatie waarover elke dieptemap beschikt. Doordat er meer dieptewaarden beschikbaar zijn, is er een zachte overgang van dieptewaarden tussen naburige pixels. De visibiliteitsinformatie van een object wordt dus nauwkeuriger beschreven.

Daarnaast neemt de compressiefactor voor elk object toe naarmate er meer dieptemappen worden geconstrueerd. Dit is te zien in de tabellen en aan het logaritmisch gedrag van de curves op de grafieken. De verklaring hiervoor is dat een toenemend aantal dieptemappen zorgt voor een grotere coherentie in de sequentie dieptemappen. Doordat er meer dieptemappen worden geconstrueerd, daalt immers de afstand tussen de posities van deze dieptemappen op de omhullende sfeer. Voor naburige dieptemappen heeft dit als gevolg dat de verschillen in dieptewaarden kleiner worden, waardoor de coherentie tussen deze dieptemappen stijgt.

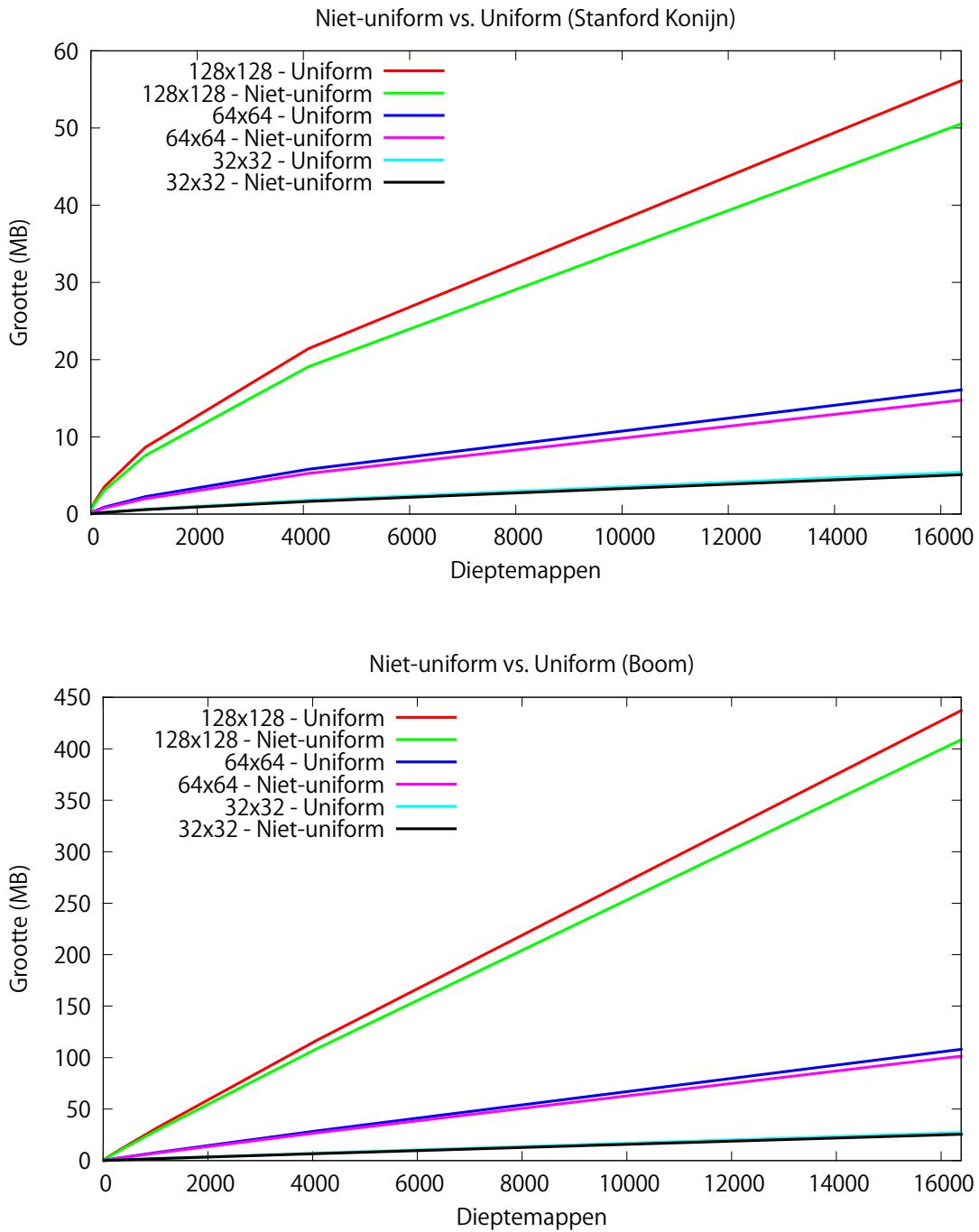
Op de grafiek is ook te zien dat de compressiefactor voor bepaalde objecten sneller toeneemt (het Stanford konijn) dan voor andere objecten (de demon). Dit heeft te maken met oppervlak van beide objecten. Voor de demon zorgt de dunne geometrie van het object ervoor dat de compressiefactor minder snel toeneemt dan bij het Stanford konijn.

5.3.4 Invloed van Uniforme Positiebepaling

Voor het bepalen van de posities van de dieptemappen op de omhullende sfeer kan zowel een uniforme als niet-uniforme verdeling worden gebruikt. Figuur 5.13 toont de resultaten van per-pixel compressie voor beide verdelingen. De coherente schaduw mappen die werden gebruikt hebben een resolutie van 128×128 pixels. Voor het bepalen van de coherente sequentie werd een horizontale zigzag curve gebruikt.



Figuur 5.12: Invloed op de compressiefactor voor verschillende resoluties. Boven: de resultaten van de demon. Onder: de resultaten van het Stanford konijn. Naarmate de resolutie en/of het aantal dieptemappen toeneemt, stijgt ook de compressiefactor.



Figuur 5.13: Invloed op de compressiefactor voor een (niet-)uniforme verdeling van de dieptemappen op de omhullende sfeer. Boven: de resultaten van het Stanford konijn. Onder: de resultaten van de boom. Op de grafieken is te zien dat een uniforme verdeling tot een lagere compressiefactor leidt. Dit is logisch, aangezien er bij een niet-uniforme verdeling meer dieptemappen worden geconstrueerd rond de polen, is er ook meer coherente tussen deze dieptemappen aanwezig.

Uit de grafiek kan worden afgeleid dat een uniforme verdeling tot een lagere compressiefactor leidt. Bij de niet-uniforme verdeling worden er immers meer dieptemappen geconstrueerd rond de polen van de omhullende sfeer dan rond de evenaar. Doordat er meer dieptemappen worden geconstrueerd rond de polen, is ook de afstand tussen deze dieptemappen kleiner. Er is dus meer coherentie tussen deze dieptemappen en bijgevolg stijgt de compressiefactor. Bij een uniforme verdeling is dit niet het geval. Daardoor ligt de compressiefactor lager.

5.3.5 Invloed van Oriëntatie

Een laatste parameter die wordt bestudeerd is de oriëntatie van de omhullende sfeer. In totaal worden zes oriëntaties onderzocht. Daarbij worden de polen van de omhullende sfeer zo geroteerd dat ze samenvallen met de x -, y - of z -as. Voor elk van deze drie opstellingen wordt ook nog de variant getest waarbij de polen 180° zijn geroteerd. De coherente schaduwassen die werden getest hebben een resolutie van 128×128 pixels en werden uniform verdeeld over het oppervlak van de omhullende sfeer. Voor het bepalen van de coherente sequentie werd een horizontale zigzag curve gebruikt. Figuur 5.14 toont de resultaten voor verschillende oriëntaties van de omhullende sfeer.

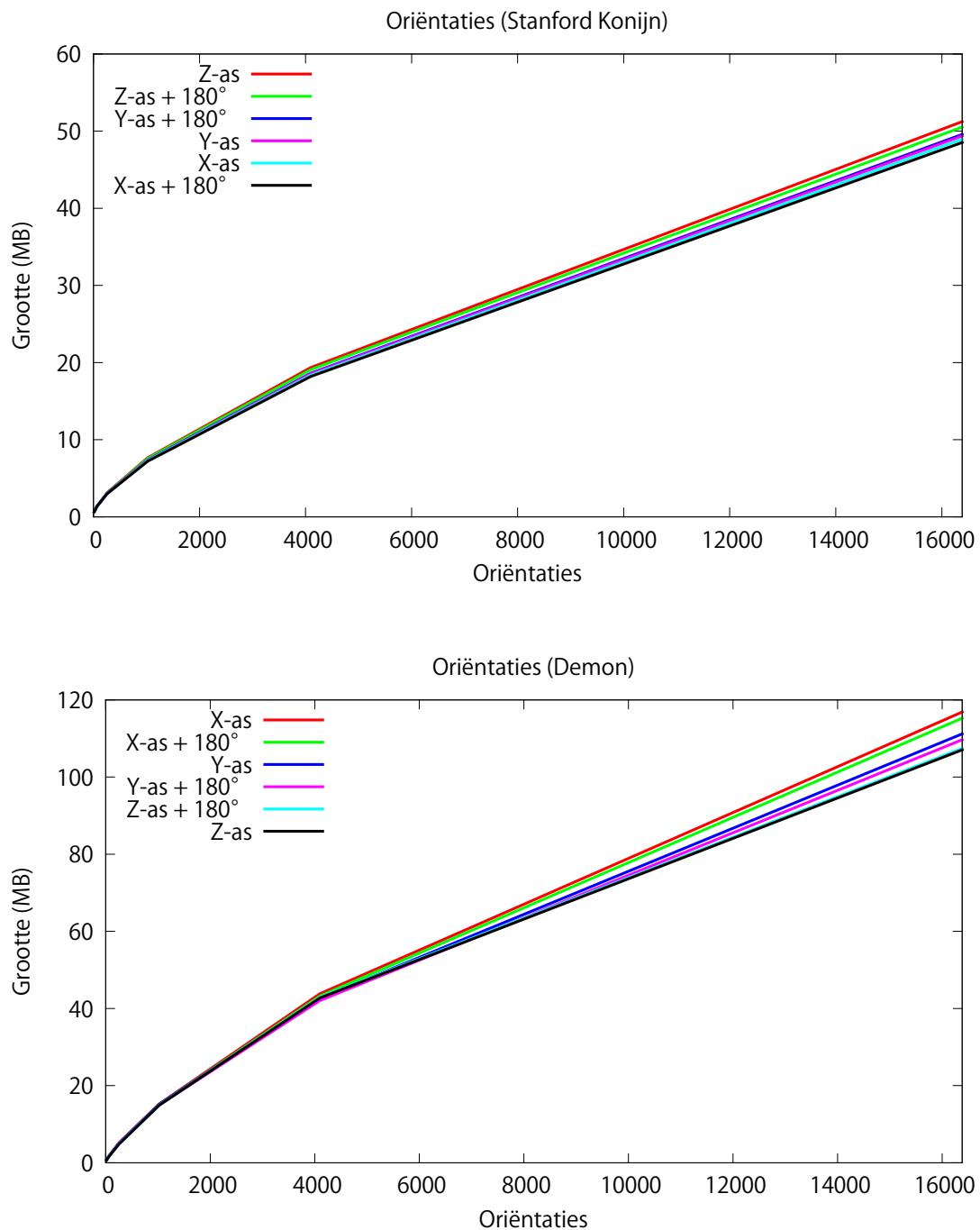
Voor beide objecten wordt een verschillend resultaat bekomen. Daaruit kan worden afgeleid dat de oriëntatie afhankelijk is van de geometrie van het object. Een mogelijke verklaring is dat dit te maken heeft met de gemeenschappelijke dieptewaarden die worden berekend. Doordat deze worden berekend voor verschillende dieptemappen kan het voorvallen dat er soms minder segmenten nodig zijn om alle dieptewaarden voor te stellen.

5.3.6 Coherentiemappen

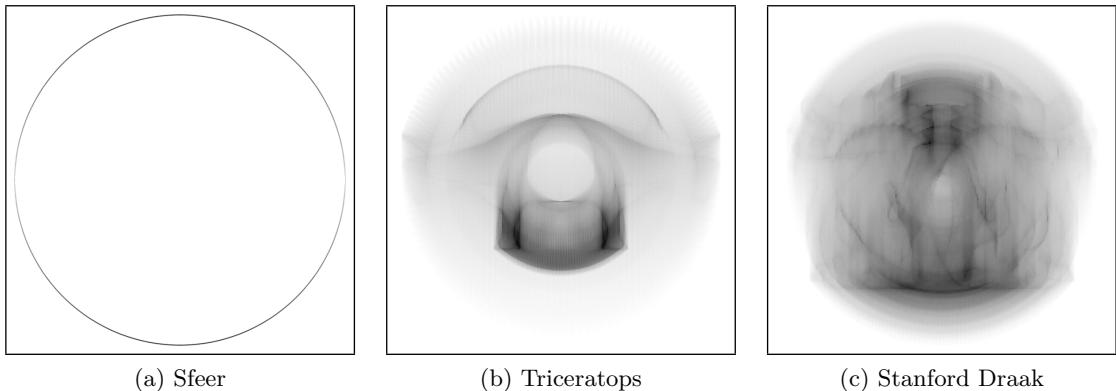
Een coherentiemap beschrijft op een grafische manier hoe coheret de dieptemappen van een bepaald object zijn. Anders gezegd, ze beschrijven hoe sterk de dieptemappen kunnen worden gecomprimeerd met per-pixel compressie.

Een coherentiemap is van dezelfde resolutie als de dieptemappen waarop deze betrekking heeft. Elke pixel van een coherentiemap drukt de efficiëntie uit voor het comprimeren van alle dieptewaarden met een gelijke pixelpositie. De waarde voor elke pixel van een coherentiemap is een getal tussen nul en één. Een lage waarde wil zeggen dat er weinig coherente is voor die pixel terwijl een hoge waarde op veel coherente duidt. Grafisch komt een waarde van nul overeen met een zwarte pixel en een waarde van één met een witte pixel.

De constructie van een coherentiemap gebeurt aan de hand van de coherente schaduwassen. De waarde voor elke pixel in een coherentiemap wordt namelijk berekend op basis van het aantal segmenten die nodig zijn om deze specifieke pixel te comprimeren. Hoe meer segmenten, hoe lager de coherente en hoe lager de waarde voor deze pixel.



Figuur 5.14: Invloed op de compressiefactor voor verschillende oriëntaties van de omhullende sfeer. Boven: de resultaten van het Stanford konijn. Onder: de resultaten van de demon. Voor elk object wordt een verschillend resultaat bekomen, dit doet vermoeden dat de oriëntatie afhankelijk is van de geometrie van het object.



Figuur 5.15: Drie coherentiemappen gesorteerd van veel coherente (links) naar weinig coherente (rechts). Voor de coherentiemap van de sfeer is duidelijk te zien dat er discontinuïteiten zijn aan de randen. Deze zijn het gevolg van de geometrische benadering die voor het oppervlak van de sfeer werd gebruikt.

Figuur 5.15 toont een coherentiemap voor een sfeer, een triceratops en de Stanford draak. De zwarte pixels duiden de gebieden aan waar er weinig coherente is. Voor de sfeer levert deze coherentiemap een interessant resultaat op. De dieptemappen zijn blijkbaar perfect coherente, op de eerder vermelde discretisatieproblemen aan de randen na.

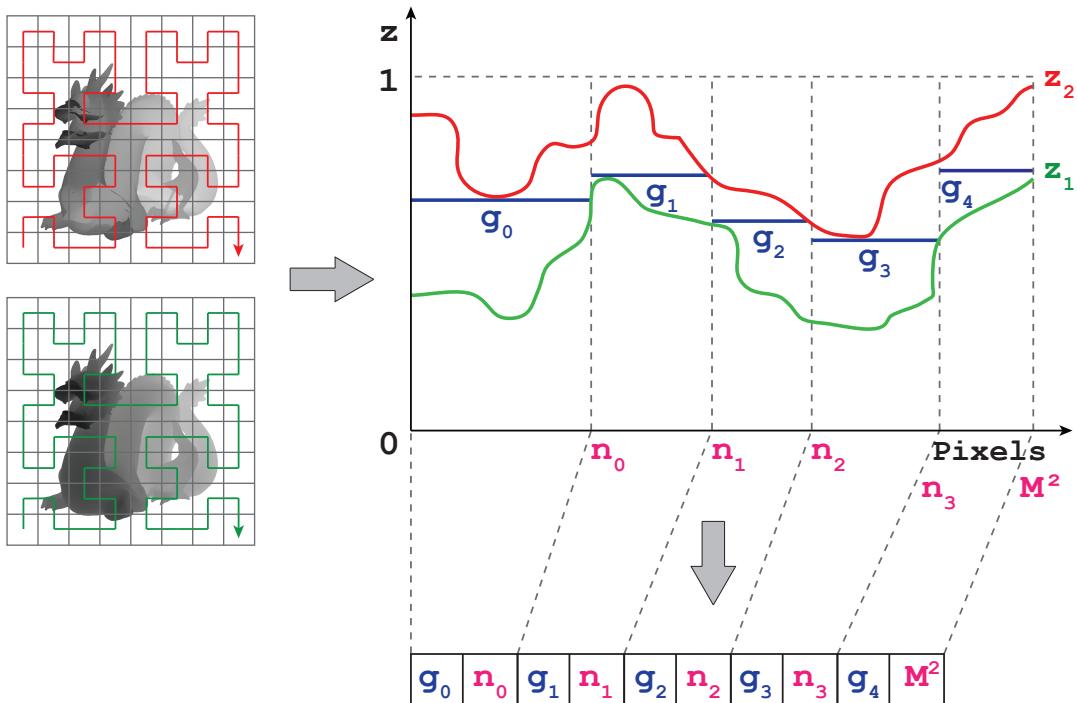
5.3.7 Conclusie

Per-pixel compressie blijkt een efficiënte compressietechniek te zijn om dieptemappen te comprimeren. Door de coherente tussen de dieptemappen uit te buiten kan een hoge compressiefactor worden bekomen. Bovendien neemt deze compressiefactor nog toe naarmate de resolutie en/of het aantal dieptemappen stijgt. De compressiefactor wordt ook nog beïnvloed door andere parameters. Om de coherente maximaal uit te buiten, moet bijvoorbeeld een horizontaal zigzagende curve worden gebruikt. Een uniforme verdeling van de dieptemappen op de omhullende sfeer zorgt dan weer voor een daling van de compressiefactor. De invloed van oriëntaties is echter moeilijker te voorspellen.

Voor objecten met sterke variaties in geometrie of zeer kleine verschillen tussen de z_1 - en z_2 -waarden, is per-pixel compressie minder efficiënt. In deze gevallen is er minder coherente aanwezig en zal de compressiefactor opvallend lager liggen.

5.4 Per-Map Compressie

Per-map compressie werkt op een gelijkaardige manier als per-pixel compressie, maar in plaats van te comprimeren op basis van pixelpositie, wordt elke dieptemap apart gecomprimeerd. Daarbij wordt gebruik gemaakt van dezelfde compressie-eigenschap als bij per-pixel compressie. Door gemeenschappelijke dieptewaarden te gebruiken, kan elke dieptemap op een compacte manier worden voorgesteld. De efficiëntie van deze compressie hangt af van de volgorde waarin de dieptewaarden van een dieptemap worden gecomprimeerd. Voor een optimaal resultaat dient dus een coherente volgorde te worden bepaald. Daarvoor worden opnieuw ruimtevullende curves gebruikt. Figuur 5.16 toont een grafische weergave van per-map compressie met behulp van een Hilbert curve. Aangezien elke dieptemap apart wordt gecomprimeerd, speelt de onderlinge volgorde van de dieptemappen geen enkele rol op het resultaat van de compressie. De gecomprimeerde data wordt opnieuw voorgesteld door segmenten (g_i, n_i) . De index n_i is gebaseerd op de ordening van de dieptewaarden in de coherente sequentie.

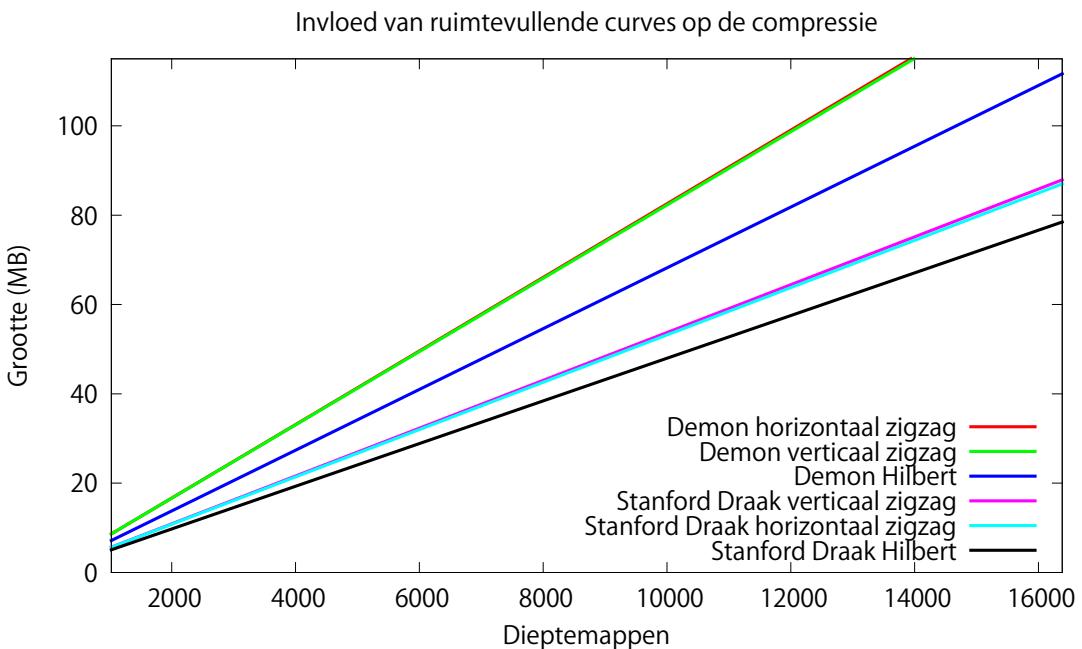


Figuur 5.16: Per-map compressie aan de hand van een Hilbert curve. Door op zoek te gaan naar de gemeenschappelijke dieptewaarden (blauw) kan een dieptemap worden gecomprimeerd. Deze dieptewaarden worden vervolgens samen met een index opgeslagen in segmenten.

Resultaten

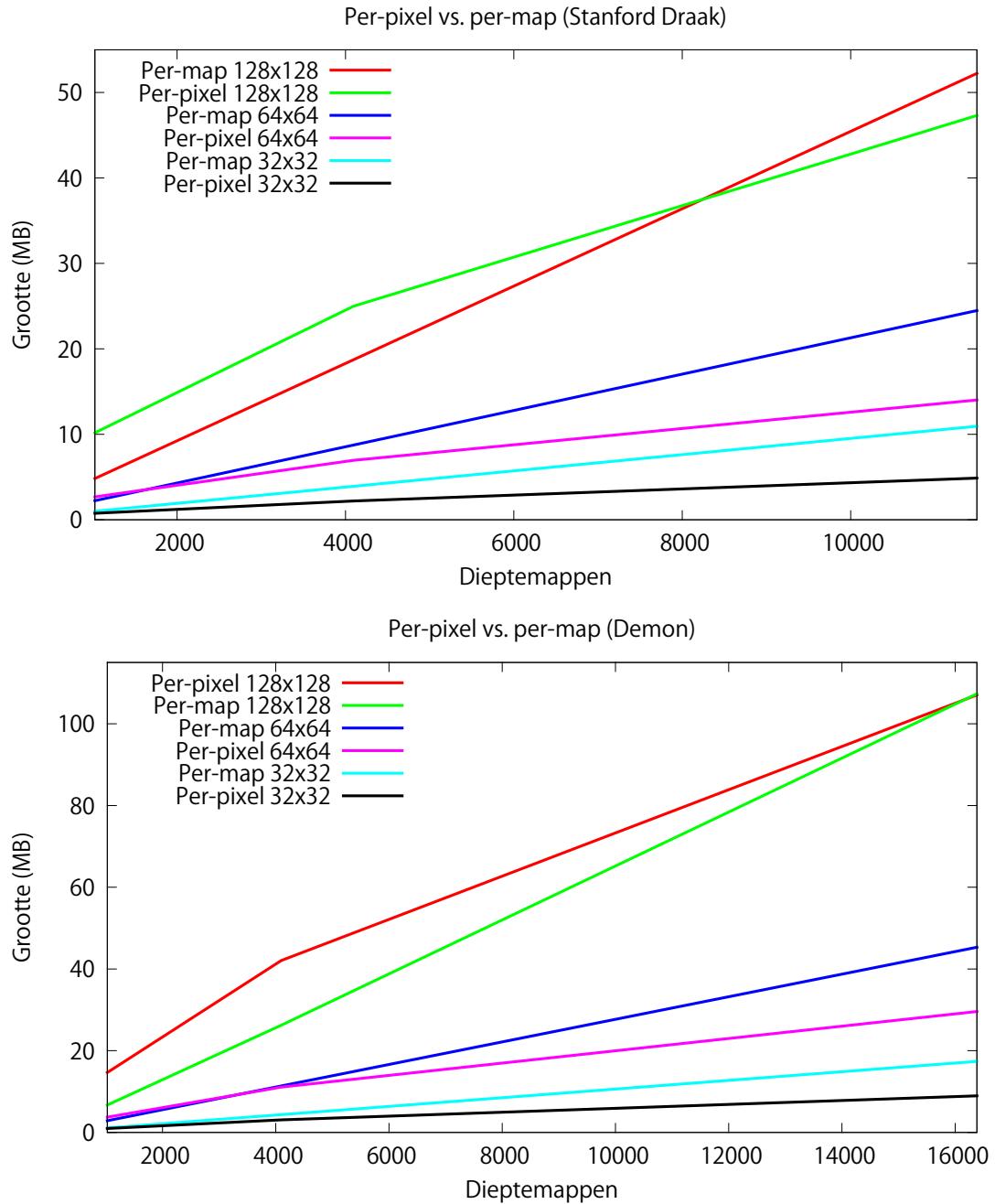
In eerste instantie wordt onderzocht voor welke ruimtevullende curve de hoogste compressiefactor wordt bekomen. De ruimtevullende curves die worden bestudeerd zijn de horizontale en verticale zigzag curve, en de tweedimensionale Hilbert curve. Deze curves werden getest voor de Stanford draak en de demon. De geconstrueerde dieptemappen hadden een resolutie van 128×128 pixels. Figuur 5.17 toont de resultaten voor verschillende ruimtevullende curves.

Uit deze resultaten blijkt dat de Hilbert curve voor zowel de Stanford draak als de demon tot de hoogste compressiefactor leidt. Dit is een gevolg van de coherente eigenschappen van de Hilbert curve. De resultaten voor de horizontale en verticale curve zijn bijna identiek. Dit is niet verwonderlijk aangezien beide curves, op een rotatie na, ook identiek zijn. Op de grafiek valt tevens te zien dat per-map compressie een lineair verloop kent in functie van het aantal dieptemappen.



Figuur 5.17: Resultaten voor verschillende ruimtevullende curves. Voor beide objecten wordt met de tweedimensionale Hilbert curve de hoogste compressiefactor bekomen.

Om de efficiëntie van per-map compressie na te gaan worden de resultaten vergeleken met per-pixel compressie. Voor de per-map compressie werd telkens gebruik gemaakt van de Hilbert curve. Figuur 5.18 toont de resultaten voor beide compressietechnieken.



Figuur 5.18: Per-map compressie ten opzichte van per-pixel compressie. De efficiëntie van per-map compressie wordt bepaald door de resolutie van de dieptemappen. Afhankelijk van deze resolutie zal met per-pixel compressie sneller een hogere compressiefactor worden bekomen. De overgangspunten tussen beide compressietechnieken zijn ook zichtbaar op de grafieken.

Op de grafiek is te zien dat de resolutie een belangrijke invloed heeft op de efficiëntie van per-map compressie. Voor hogere resoluties wordt, afhankelijk van het aantal dieptemappen, een hogere compressiefactor bekomen dan voor per-pixel compressie. Bij per-pixel compressie stijgt de compressiefactor echter naarmate het aantal dieptemappen toeneemt, terwijl bij per-map compressie de compressiefactor constant blijft. Daardoor zal met per-pixel compressie sowieso een beter resultaat worden bekomen naarmate het aantal dieptemappen toeneemt.

5.5 K-Means Clustering

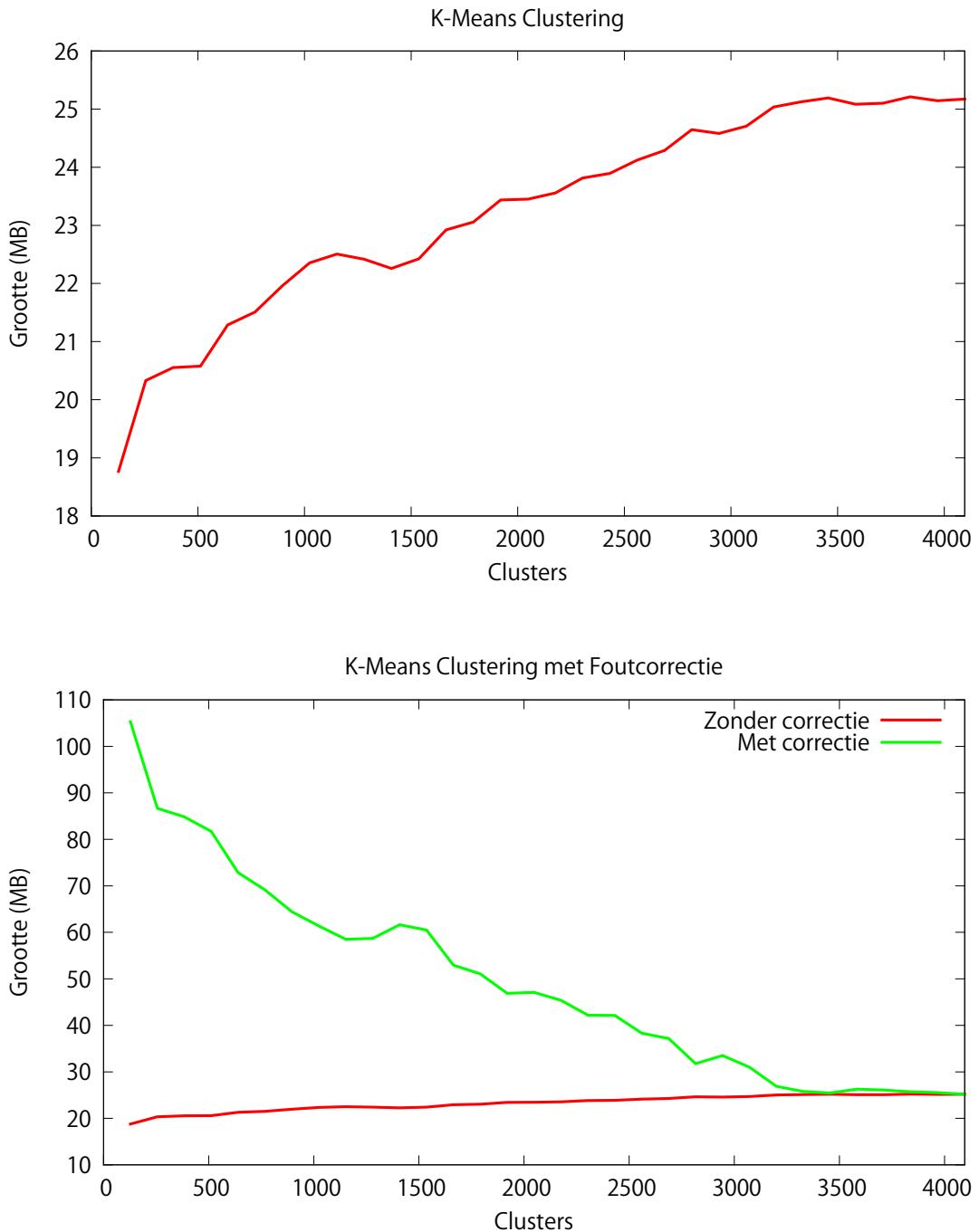
Bij per-pixel compressie wordt de compressiefactor in grote mate beïnvloed door de coherentie tussen de dieptemappen. Om tot een hogere compressiefactor te komen, kan bijvoorbeeld de coherentie tussen deze dieptemappen worden verhoogd. Een mogelijke manier om dit te bereiken is door gebruik te maken van *k-means clustering* [Llo82].

K-means clustering is een iteratieve methode om N punten te verdelen over k clusters. Initieel worden k willekeurige punten geselecteerd. Deze punten vormen het gemiddelde voor elke cluster. Daarna kan het clusteringsproces beginnen. Elk punt wordt toegekend aan de meest nabije cluster. Hiervoor wordt de afstand gemeten tot het gemiddelde van elke cluster. Na afloop beschikken alle clusters over een verzameling van punten. Aan de hand van deze verzamelingen wordt vervolgens voor elke cluster een nieuw gemiddelde berekend. Door deze procedure te herhalen tot convergentie optreedt, bekomen we een partitionering van N punten over k clusters.

Voor het verhogen van de coherentie tussen de dieptemappen worden de N dieptemappen eerst geclusterd. Elke dieptemap is dan toegekend aan een cluster. Vervolgens wordt elke dieptemap vervangen door het gemiddelde van de cluster waartoe deze dieptemap behoort. Het resultaat is dan een nieuwe sequentie van N dieptemappen. Voor k clusters vertaalt dit zich in k unieke dieptemappen die meerdere keren kunnen voorkomen in deze sequentie. De nieuwe sequentie is daardoor coherenter dan de originele sequentie. Waar voordien nog kleine verschillen mogelijk waren tussen gelijkaardige dieptemappen, zijn deze nu identiek aan elkaar. Om deze coherentie uit te buiten, kan ten slotte per-pixel compressie worden toegepast.

Resultaten

Bij het clusteren van de dieptemappen is uiteraard het aantal clusters belangrijk. De grafiek bovenaan figuur 5.19 toont het verband tussen het aantal clusters en de grootte van de coherente schaduwmannen na per-pixel compressie. Op de grafiek is te zien dat de bestandsgrootte toeneemt naarmate het aantal clusters stijgt. De verklaring daarvoor is dat voor een stijgend aantal clusters, ook het aantal unieke dieptemappen toeneemt. Daardoor neemt de coherentie af en daalt de compressiefactor. Merk op dat indien het aantal clusters gelijk is aan het aantal dieptemappen, hetzelfde resultaat wordt bekomen



Figuur 5.19: K-means clustering voor de Stanford draak en 4096 dieptemappen. Boven: resultaten van k-means clustering. De originele grootte van de coherente schaduwmappen komt overeen met het geval waarbij 4096 clusters worden gebruikt. Het aantal clusters bepaalt dus hoe sterk de coherente schaduwmappen nog kunnen worden geocomprimeerd. Onder: om foutloze compressie te bekomen moet een correctie worden uitgevoerd. Daarbij wordt voor elke foute pixel ook nog de correcte dieptewaarde opgeslagen. Aangezien het aantal fouten zeer hoog ligt, is k-means clustering geen praktisch alternatief voor per-pixel compressie.

als wanneer er enkel per-pixel compressie wordt toegepast.

Hoewel het met clustering mogelijk is om een tot een hogere compressiefactor te komen, heeft deze manier van werken één groot nadeel. Doordat de dieptemappen worden vervangen door het gemiddelde van de cluster waar toe ze behoren, worden immers fouten geïntroduceerd. Naarmate het aantal clusters daalt, stijgt het aantal pixels waarvoor de visibiliteitstest niet meer correct is. Het is het dan ook belangrijk om met deze fouten rekening te houden bij het vergelijken van de resultaten met en zonder clustering. Daarom kan voor elke pixel met een foutieve dieptewaarde, de (ongecomprimeerde) correcte dieptewaarde worden opgeslagen. De grafiek onderaan figuur 5.19 toont het resultaat van deze correctie. Op de grafiek is te zien dat voor een laag aantal clusters, zeer veel bijkomende dieptewaarden moeten worden opgeslagen om een foutloze compressie te bekomen. Daardoor is k-means clustering niet geschikt om dieptemappen op een foutloze manier te comprimeren.

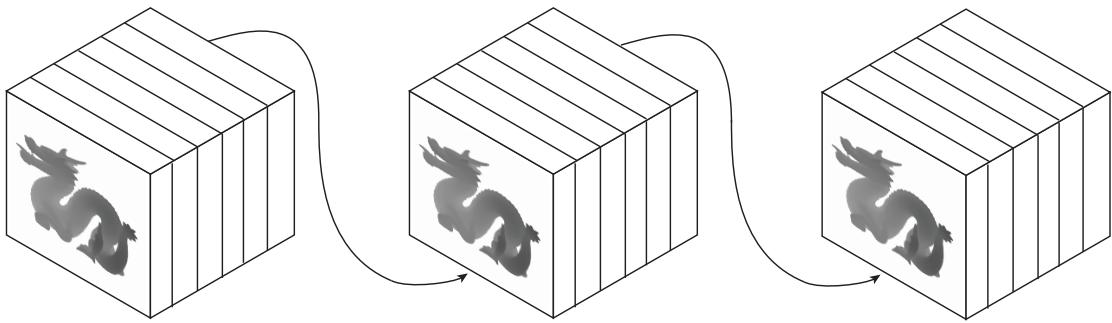
5.6 Compressie aan de hand van een Hilbert Curve

Bij het comprimeren van dieptemappen met per-pixel compressie speelt de volgorde waarin de pixels worden gecomprimeerd een belangrijke rol. Hoe coherenter deze volgorde, hoe efficiënter de compressie werkt. Bij per-pixel compressie worden telkens de dieptewaarden met dezelfde pixelpositie gecomprimeerd. Er wordt dus enkel rekening gehouden met de coherentie tussen dieptewaarden in opeenvolgende dieptemappen. Er is echter ook coherentie aanwezig tussen nabijgelegen pixels in dezelfde dieptemap, net zoals nabijgelegen pixels op een kleurenfoto meestal een gelijkaardige kleur hebben. Om de coherentie in alle dimensies uit te buiten kan een Hilbert curve worden gebruikt. Zoals al werd aangehaald in sectie 2.4, is de Hilbert curve zeer geschikt voor het bewaren van de coherentie van naburige punten. Bovendien is deze curve multidimensionaal. Daardoor kan een coherente volgorde voor de dieptewaarden worden bepaald.

Net als bij de voorgaande technieken is de compressie gebaseerd op de eigenschap van duale dieptelagen. De gemeenschappelijke dieptewaarden worden, samen met een index n_i , opgeslagen in segmenten. De index n_i is gebaseerd op de Hilbert curve, aangezien deze een ordening bepaald voor alle dieptewaarden die worden gecomprimeerd.

5.6.1 Driedimensionale Hilbert Compressie

Met behulp van een driedimensionale Hilbert curve kan de coherentie tussen pixels in meerdere dimensies worden uitgebuit. Aangezien de Hilbert curve enkel gedefinieerd is voor een hyperkubus, dienen in dit geval de dieptemappen te worden georganiseerd in kubussen.



Figuur 5.20: Voor de compressie gebaseerd op een driedimensionale Hilbert curve worden de dieptemappen verdeeld in kubussen. Elke kubus wordt vervolgens apart gecomprimeerd.

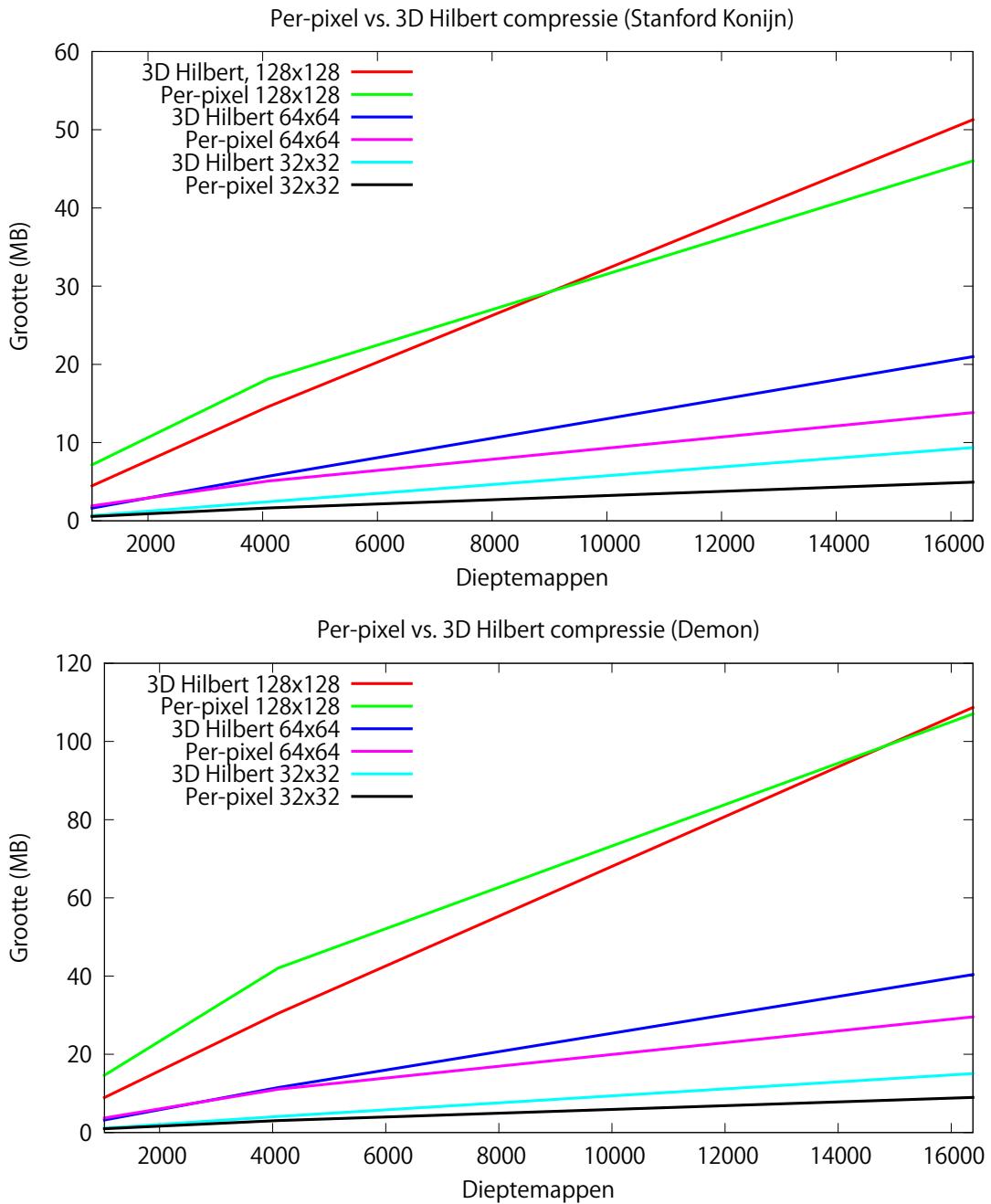
De grootte van elke kubus wordt bepaald door de resolutie van de dieptemap. Voor een resolutie van bijvoorbeeld $2^7 \times 2^7 = 128 \times 128$ pixels wordt een Hilbert curve van orde 7 gebruikt. Dit komt overeen met een kubus van $2^{7 \times 3} = 128^3$ elementen. Bijgevolg zullen dus 128 dieptemappen worden gebruikt om de kubus voor te stellen. Aangezien het aantal dieptemappen in praktijk hoger ligt dan 128 dienen dus meerdere kubussen te worden gebruikt. Daarom zal de volledige sequentie aan dieptemappen worden opgesplitst in kubussen van gelijke grootte. Elk van deze kubussen wordt dan apart gecomprimeerd met behulp van de driedimensionale Hilbert curve. Figuur 5.20 illustreert deze werkwijze.

Resultaten

Figuur 5.21 toont de resultaten van de driedimensionale Hilbert compressie voor respectievelijk het Stanford konijn en de demon. Net als bij per-map compressie wordt de efficiëntie van de driedimensionale Hilbert compressie voornamelijk bepaald door de resolutie van de dieptemappen. Naarmate de resolutie toeneemt is er een duidelijke verbetering ten opzichte van per-pixel compressie, zolang het aantal dieptemappen relatief laag ligt. Wanneer het aantal dieptemappen groter wordt stijgt de compressiefactor voor per-pixel compressie echter sneller dan voor de driedimensionale Hilbert compressie. Daardoor levert ook deze compressie slechts in enkele gevallen betere resultaten op.

5.6.2 Vierdimensionale Hilbert Compressie

De volledige verzameling van dieptemappen kan ook op een vierdimensionale manier worden voorgesteld. Elke dieptemap wordt immers op een unieke wijze geïdentificeerd aan de hand van zijn positie op de omhullende sfeer. Deze positie kan worden voorgesteld door sferische coördinaten (θ, ψ) . Daarnaast bestaat elke dieptemap uit twee dimensies (x, y) . Samen vormt dit een vierdimensionale ruimte. Met behulp van een vierdimensionale Hilbert curve kan een coherente volgorde worden bepaald voor alle elementen in deze ruimte.



Figuur 5.21: Compressie gebaseerd op een driedimensionale Hilbert curve. De efficiëntie van de driedimensionale Hilbert compressie wordt bepaald door de resolutie van de dieptemappen. Afhankelijk van deze resolutie zal met per-pixel compressie sneller een hogere compressiefactor worden bekomen. De overgangen tussen beide compressietechnieken zijn ook zichtbaar op de grafieken.

Resultaten

Tabel 5.2 geeft de resultaten terug van de vierdimensionale Hilbert compressie voor een hyperkubus met $2^{7 \times 4}$ elementen. Dit komt overeen met een resolutie van 128×128 pixels en 16384 dieptemappen. De volgorde waarin de dieptemappen geordend zijn, is opnieuw afhankelijk van de ruimtevullende curve die wordt gebruikt. Beide zigzag curves en de tweedimensionale Hilbert curve werden getest. In alle drie de gevallen blijkt de vierdimensionale Hilbert compressie minder efficiënt te zijn dan de per-pixel compressie.

Volgorde	Per-pixel	4D Hilbert	Δ
Zigzag (horizontaal)	62 MB	164 MB	+266%
Zigzag (verticaal)	116 MB	188 MB	+162%
2D Hilbert	76 MB	161 MB	+212%

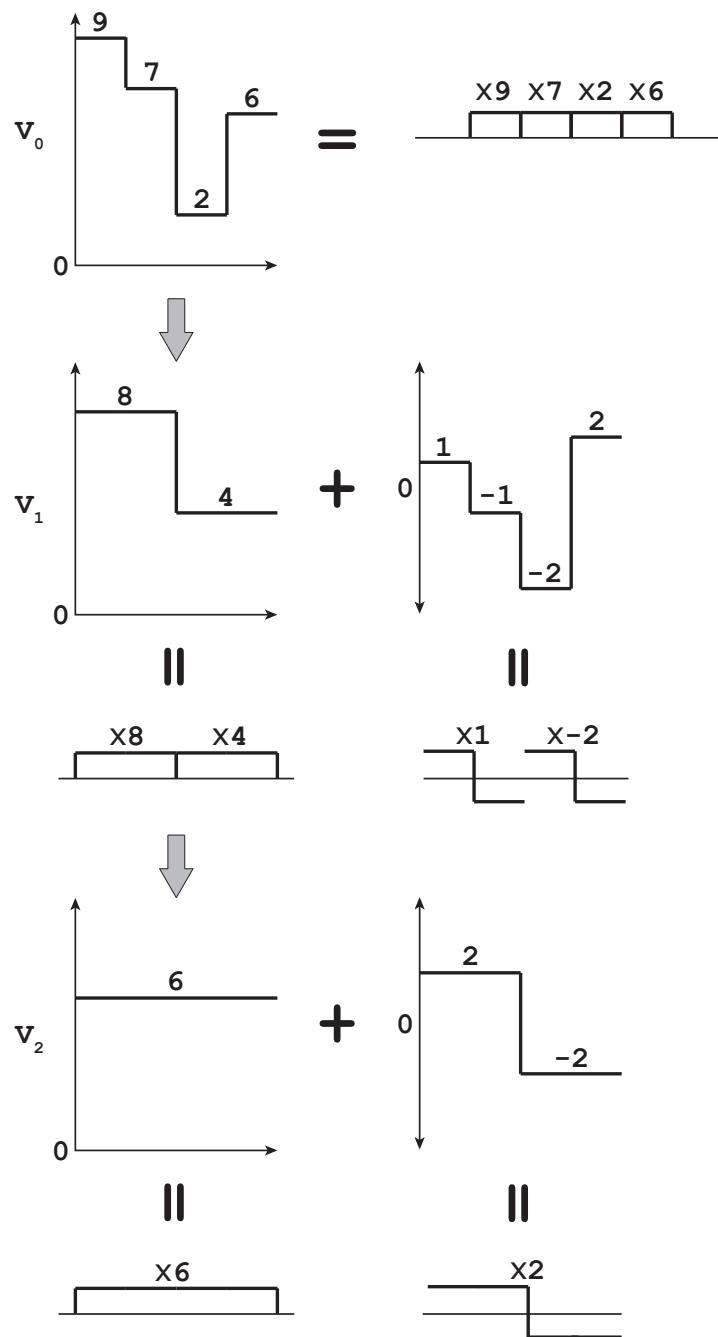
Tabel 5.2: Resultaten van de vierdimensionale Hilbert compressie voor een resolutie van 128×128 pixels, 16384 dieptemappen en de Stanford draak als model. Uit de resultaten blijkt dat een compressie gebaseerd op de vierdimensionale Hilbert curve niet tot een hogere compressiefactor zal leiden.

5.7 Wavelet Compressie

De laatste compressietechniek die wordt onderzocht maakt gebruik van wavelets. Wavelets zijn korte golven of signalen waarin andere signalen kunnen worden ontbonden. Wavelets laten toe om een signaal te herschrijven als een combinatie van ruwe benaderingen en details. Deze signalen kunnen van alle aard zijn en hoeven dus niet beperkt te zijn tot audiovisuele signalen. Wavelets vormen de basis voor verscheidene bestandsformaten zoals JPEG en PNG [SDS95].

De Haar wavelet is de meest eenvoudige wavelet. De eendimensionale versie van deze wavelet kan worden voorgesteld als het signaal $[1 -1]$. Met behulp van deze wavelet kan een signaal worden ontbonden. Neem als voorbeeld de vector $[9 7 2 6]$. De wavelet transformatie van deze vector wordt geïllustreerd in figuur 5.22.

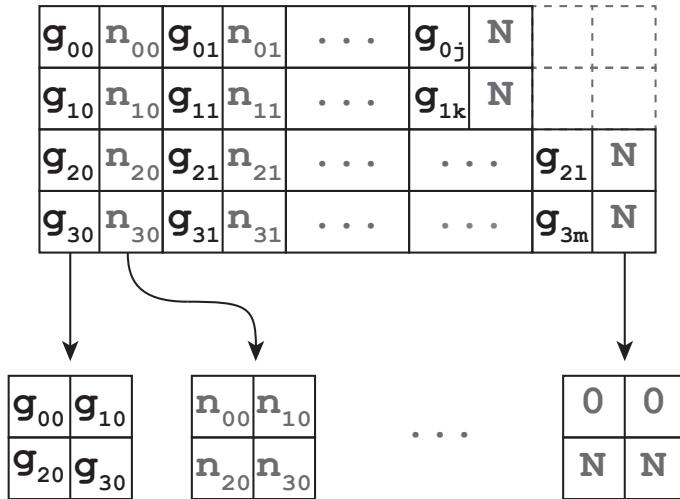
De Haar wavelet transformatie ontbindt een signaal dus in zijn gemiddelde met bijhorende detailcoëfficiënten. Zowel de originele vector als de wavelet transformatie beschrijven eenzelfde signaal en bestaan uit evenveel coëfficiënten. Een voordeel van wavelet transformaties is dat een groot deel van de coëfficiënten na transformatie zeer klein is. Deze kleine coëfficiënten kunnen vervolgens worden weggelaten om tot een meer compacte representatie te komen. Doordat deze coëfficiënten details bevatten, introduceren ze dan ook maar een kleine fout bij de reconstructie van het originele signaal.



Figuur 5.22: Wavelet transformatie van een vector. Het originele signaal wordt voorgesteld door het gemiddelde van elk paar coëfficiënten: $[8 \ 4]$ en detailcoëfficiënten $[1 \ -2]$. Deze detailcoëfficiënten stellen veelvouden van de Haar wavelet voor. De wavelet transformatie wordt vervolgens toegepast op de vector $[8 \ 4]$. De volledige wavelet transformatie van een vector is dan $[6 \ 2 \ 1 \ -2]$. Dit resultaat beschrijft een multi-resolutie analyse van de originele vector.

Compressie van Dieptemappen

Voor het comprimeren van de dieptemappen kan een tweedimensionale Haar wavelet worden gebruikt. De compressie wordt toegepast na het uitvoeren van de per-pixel compressie. Aangezien deze data in gecomprimeerde vorm wordt opgeslagen, dient de data eerst op een andere manier te worden voorgesteld vooraleer de wavelet transformatie kan worden gebruikt.



Figuur 5.23: De per-pixel gecomprimeerde data wordt op een andere manier voorgesteld vooraleer wavelet compressie wordt toegepast. Daarbij worden alle segmenten verdeeld over tweedimensionale mappen.

Daarom worden nieuwe mappen aangemaakt met dezelfde resolutie als de gecomprimeerde dieptemappen. Deze mappen zullen de segmenten opslaan die werden bekomen na het toepassen van de per-pixel compressie. De eerste map met dieptewaarden slaat voor elke pixel de gemeenschappelijk dieptewaarden g_i op van het eerste segment. De tweede map de dieptewaarden van het tweede segment, enz. De mappen die de breuklijnen opslaan worden op een analoge manier geconstrueerd, alleen worden in plaats van dieptewaarden, de indices n_i opgeslagen. Er worden dus evenveel mappen geconstrueerd als de maximale lengte van alle lijsten van segmenten. Figuur 5.23 illustreert dit proces.

Op elk van de geconstrueerde mappen wordt dan de tweedimensionale Haar wavelet transformatie uitgevoerd. Het aantal nulcoëfficiënten in de resulterende mappen bepaalt hoe sterk de data kan worden gecomprimeerd met behulp van wavelet compressie.

Resultaten

Er werd onderzocht in welke mate wavelet compressie tot een hogere compressiefactor zou kunnen leiden. De resultaten hiervan worden beschreven in tabel 5.3. De kolom voor wavelet transformatie beschrijft de grootte van de data nadat de gecomprimeerde diep-

temappen werden omgezet naar tweedimensionale mappen. Het aantal nulcoëfficiënten dat in deze data voorkomt wordt beschreven in de vierde kolom. Door de nulcoëfficiënten te verwijderen wordt een wavelet gecomprimeerde voorstelling van de data bekomen. De resultaten hiervan worden beschreven in kolom vijf.

Diepte-mappen	Per-pixel	Wavelet transformatie	Nul-coëfficiënten	Wavelet compressie
1.024	10 MB	35 MB	60%	14 MB
4.096	24 MB	81 MB	60%	32 MB
16.384	60 MB	192 MB	59%	79 MB

Tabel 5.3: Resultaten na wavelet transformatie voor een gordeldier object. Het aantal nulcoëfficiënten ligt te laag om met wavelet compressie tot een hogere compressiefactor te komen.

Uit deze resultaten blijkt dat ongeveer 60% van alle coëfficiënten gelijk zijn aan nul. Het zijn deze nullen die belangrijk zijn voor de efficiëntie van de wavelet compressie. Deze nullen hebben namelijk geen invloed bij de reconstructie van het gereconstrueerde signaal. Door deze nullen weg te laten wordt dus een foutloze compressie bekomen. In de laatste kolom wordt een schatting gegeven voor de grootte van de data na wavelet compressie. Daaruit blijkt dat met wavelet compressie geen hogere compressiefactor zal worden bekomen. Daarvoor ligt het aantal nulcoëfficiënten immers te laag.

5.8 Conclusie

In dit hoofdstuk werd onderzocht of een compacte representatie van de dieptemappen mogelijk is. De technieken die daarbij werden bestudeerd zijn per-pixel compressie, per-map compressie, per-pixel compressie in combinatie met k-means clustering, drie- of vierdimensionale Hilbert compressie en ten slotte wavelet compressie.

In sectie 5.3 werd geconcludeerd dat per-pixel compressie een efficiëntie techniek is om dieptemappen te comprimeren. Door de coherentie tussen de dieptemappen uit te buiten kan een hoge compressiefactor worden bekomen. Bovendien neemt deze compressiefactor nog toe naarmate de resolutie en/of het aantal dieptemappen stijgt. Enkel voor objecten met sterke variaties in geometrie of zeer kleine verschillen tussen de z_1 - en z_2 -waarden, is per-pixel compressie minder efficiënt.

Met per-map compressie wordt elke dieptemap afzonderlijk gecomprimeerd. De resultaten daarvan komen in de buurt van per-pixel compressie. Voor specifieke gevallen wordt zelfs een hogere compressiefactor bekomen dan bij per-pixel compressie. Naarmate het aantal dieptemappen toeneemt, wordt met per-pixel compressie echter een beter resultaat bekomen.

Met k-means clustering is het mogelijk om, ongeacht resolutie of aantal dieptemappen, een hogere compressiefactor te bekomen. Daarvoor worden weliswaar veel fouten geïntroduceerd in de gecomprimeerde data. Voor het visualiseren van realistische schaduwen zijn deze fouten niet gewenst. Indien voor de compressie van dieptemappen een niet-foutloze compressietechniek is gewenst, kan k-means clustering worden gebruikt. Het resultaat op de kwaliteit van de schaduwen valt echter niet te voorspellen.

De driedimensionale en vierdimensionale Hilbert compressie maken gebruik van een Hilbert curve om een coherente sequentie te bepalen voor alle pixels en alle dieptemappen. Net als bij per-map compressie benadert de driedimensionale Hilbert compressie de resultaten van per-pixel compressie. Naarmate het aantal dieptemappen toeneemt, wordt met per-pixel compressie echter een beter resultaat bekomen. De vierdimensionale Hilbert compressie en de wavelet compressie bleken beide niet geschikt om dieptemappen te comprimeren.

Aangezien het niet eenvoudig is om een compressietechniek te vinden die in elke situatie een hogere compressiefactor oplevert dan per-pixel compressie, kan worden besloten dat per-pixel compressie een zeer efficiënte techniek is om dieptemappen te comprimeren.

Hoofdstuk 6

Kwaliteitsstudie

Bij het construeren van de coherente schaduwmannen kunnen verschillende parameters worden ingesteld. In het vorige hoofdstuk werd al gezien dat elk van deze parameters een invloed heeft op de compressiefactor. De invloed van deze parameters op de kwaliteit van de gegenereerde schaduwen is echter niet altijd gekend.

In dit hoofdstuk wordt onderzocht welke invloed de verschillende parameters hebben op de nauwkeurigheid van de gegenereerde schaduwen. In sectie 6.1 wordt de procedure beschreven waarmee de kwaliteit van de coherente schaduwmannen werd gemeten. In de daarop volgende secties wordt de invloed van volgende parameters bestudeerd:

- de geometrie van het object (sectie 6.2);
- de resolutie en het aantal dieptemappen (sectie 6.3);
- een (niet-)uniforme verdeling van de dieptemappen (sectie 6.4);
- de oriëntatie van de omhullende sfeer (sectie 6.5).

6.1 Opstelling

Om de nauwkeurigheid van coherente schaduwmannen op te meten werden een groot aantal visibiliteitstesten uitgevoerd. Het resultaat van deze visibiliteitstesten werd ook berekend met behulp van ray tracing. Met ray tracing wordt het correcte resultaat van de visibiliteitstesten bekomen, deze resultaten dienen dus als referentie. Door de resultaten van de coherente schaduwmannen te vergelijken met de resultaten van ray tracing, kan een indicatie worden bekomen van de nauwkeurigheid van coherente schaduwmannen.

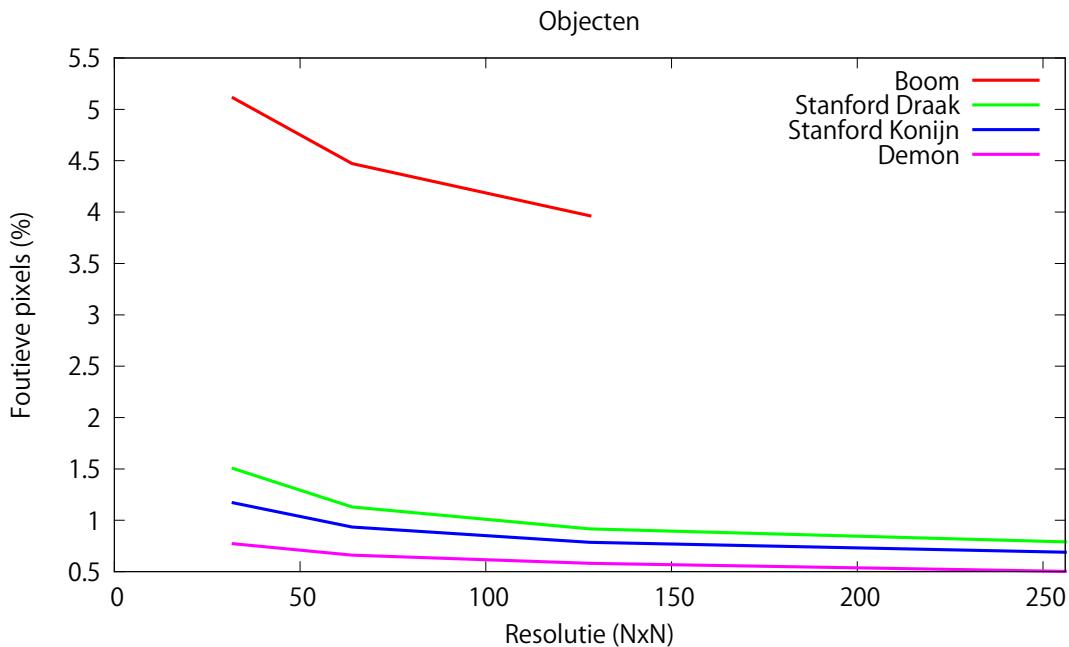
Om de visibiliteitstesten op te stellen werden meerdere visualisaties gemaakt van het object waarvoor coherente schaduwmannen werden geconstrueerd. Daarbij werd telkens een andere positie gekozen voor camera en lichtbron. Voor de camera werden willekeurige posities genomen op de omhullende sfeer van het object. De lichtbron bevond zich

telkens op een willekeurig positie buiten deze omhullende sfeer, maar binnen een maximale afstand van het object. Het type lichtbron dat werd gebruikt was een puntlichtbron.

Voor elke combinatie van posities voor camera en lichtbron werd de scène opnieuw gevisualiseerd. De ene keer werden coherente schaduwassen gebruikt, de andere keer ray tracing. Door beide visualisaties met elkaar te vergelijken werd het aantal pixels berekend waarvoor coherente schaduwassen een verschillend (en dus foutief) resultaat opleveren. Dit geeft een indicatie van de nauwkeurigheid van coherente schaduwassen.

6.2 Invloed van Geometrie

Om de invloed van geometrie op de kwaliteit van coherente schaduwassen te bestuderen, werden vier objecten getest: het Stanford konijn [TL94], de Stanford draak [CL96], een demon en een boom. Een visualisatie van elk object is te zien op figuur 5.9. De coherente schaduwassen die voor elk object werden geconstrueerd, waren opgebouwd uit 1024 dieptemappen die uniform verdeeld waren over de omhullende sfeer. De dieptemappen beschikten over een resolutie van 128×128 pixels. Figuur 6.1 toont de resultaten voor de verschillende objecten. Voor de boom zijn geen resultaten beschikbaar voor resoluties hoger dan 128×128 pixels. Dit komt omdat de coherente schaduwassen voor dit object zo groot zijn, dat ze nog moeilijk kunnen worden geconstrueerd.



Figuur 6.1: De invloed van geometrie op het aantal foutieve pixels. Objecten die veel detailinformatie bevatten introduceren opvallend meer fouten.

Op de grafiek is een duidelijk verband te zien tussen de geometrie van een object en het aantal foutieve pixels. Voor de boom ligt het aantal foutieve pixels opvallend hoger dan bij de andere objecten. Dit is te wijten aan de fijne structuur van takken en bladeren. Om deze details nauwkeurig te kunnen visualiseren, moeten de dieptemappen op een hogere resolutie worden geconstrueerd. Voor de drie andere objecten ligt het percentage van foutieve pixels veel lager. Deze objecten hebben dan ook een relatief zachtverlopend oppervlak en bevatten weinig echte detailinformatie. Coherente schaduwmannen geven dus een nauwkeurige benadering van de visibiliteitsinformatie voor deze objecten.

Merk op dat in hoofdstuk 5 een lage compressiefactor werd bekomen, bij de constructie van coherente schaduwmannen voor de boom en de demon. Voor beide objecten is er echter een groot verschil in de nauwkeurigheid van de schaduwen. Daaruit kan worden besloten dat de compressiefactor geen rechtstreeks invloed heeft op de nauwkeurigheid van de gegenereerde schaduwen.

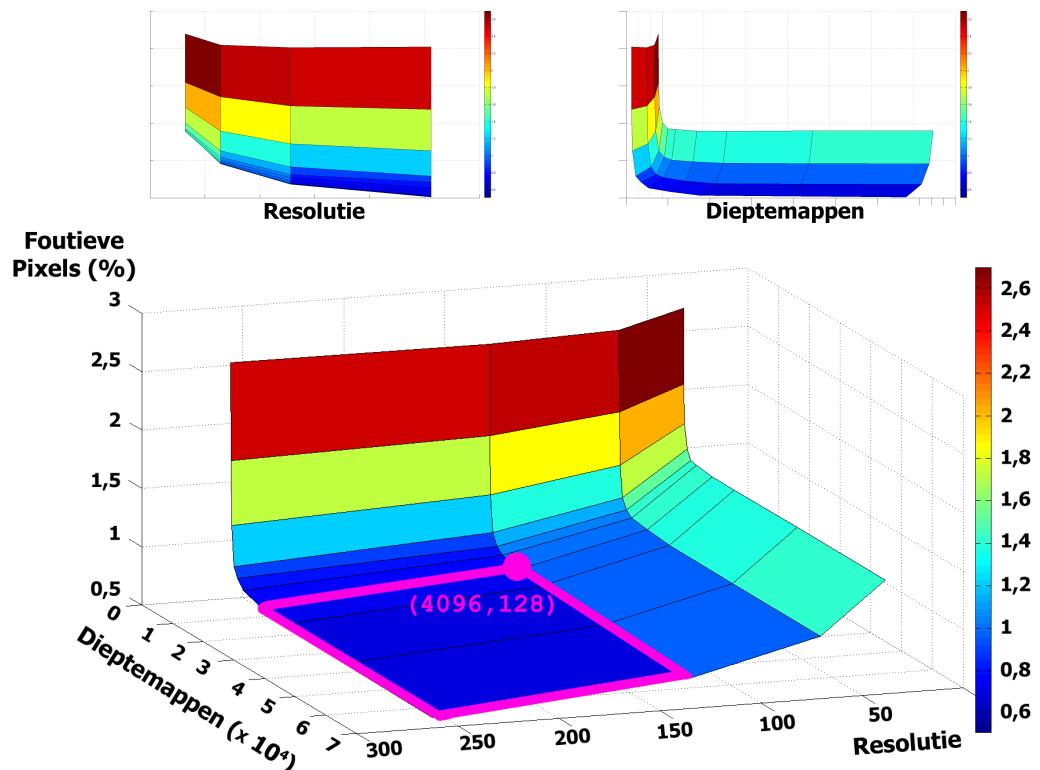
6.3 Invloed van Resolutie en Aantal Dieptemappen

De parameters die de grootste invloed hebben op de nauwkeurigheid van coherente schaduwmannen zijn de resolutie en het aantal dieptemappen. De invloed van beide parameters wordt bestudeerd voor de Stanford draak en de boom. Figuur 6.2 en 6.3 tonen een driedimensionale weergave van het verband tussen de resolutie en het aantal dieptemappen. In appendix B worden nog bijkomende grafieken en tabellen ter beschikking gesteld.

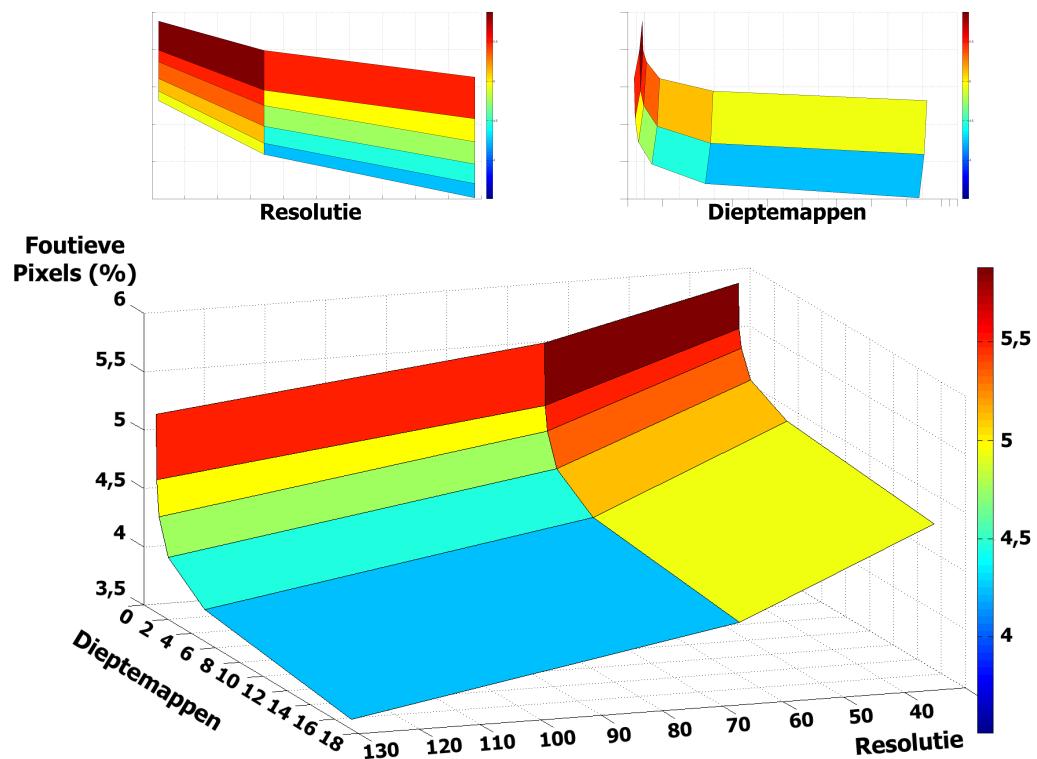
Een eerste observatie is dat de nauwkeurigheid stijgt naarmate de resolutie van de dieptemappen stijgt. Dit is logisch, aangezien voor een hogere resolutie een meer gedetailleerd beeld van de visibiliteitsinformatie wordt bekomen. Een analoog effect is zichtbaar voor het aantal dieptemappen. Voor een toenemend aantal dieptemappen daalt het aantal foutieve pixels. Dit komt omdat voor een groter aantal dieptemappen de kans stijgt dat een dieptemap kan worden gevonden die aansluit bij de richting van het invallend licht.

Voor een laag aantal dieptemappen heeft de resolutie echter geen of amper invloed op de nauwkeurigheid. Dit komt omdat bij het visualiseren geen dieptemap wordt gevonden die aansluit met de richting van het invallend licht. In dat geval maakt de resolutie van de dieptemap weinig uit aangezien toch al een fout wordt gemaakt ten opzichte van de richting van het licht. Dit effect verdwijnt naarmate het aantal dieptemappen toeneemt.

De positieve invloed voor een bijkomend aantal dieptemappen neemt ook af naarmate er over meer dieptemappen wordt beschikt. Voor een groot aantal dieptemappen is de afstand tussen twee posities op de omhullende sfeer zo klein geworden, dat er nog weinig meerwaarde te halen valt in het toevoegen van extra dieptemappen.



Figuur 6.2: Verband tussen de resolutie en het aantal dieptemappen voor de Stanford draak. Voor een lage resolutie of een laag aantal dieptemappen, ligt het aantal foutieve pixels hoog. Naarmate het aantal dieptemappen toeneemt en de resolutie stijgt, zal het aantal foutieve pixels stagneren. Dit leidt tot een gebied waarvoor de nauwkeurigheid nagenoeg constant zal blijven. Dit gebied is op de grafiek aangeduid met een roze rand. Aan de hand van dit gebied kan tevens een punt worden bepaald, waarvoor de bestandsgrootte van de coherente schaduw mappen minimaal is. Voor de Stanford draak komt dit punt overeen met 4096 dieptemappen en een resolutie van 128×128 pixels.



Figuur 6.3: Verband tussen de resolutie en het aantal dieptemappen voor de boom. Wegens de grote bestandsgrootte voor dit object, zijn slechts een beperkt aantal resultaten beschikbaar. Net als bij de Stanford draak lijkt het aantal foutieve pixels te stagneren naarmate het aantal dieptemappen en de resolutie toeneemt.

Net als bij de dieptemappen neemt ook de positieve invloed van de resolutie af naarmate er over een hogere resolutie wordt beschikt. De verklaring hiervoor is dat voor de meeste objecten al een goede benadering van de visibiliteitsinformatie wordt bekomen voor lage resoluties. De hogere resoluties dienen dan voornamelijk om detailinformatie te bewaren.

Aangezien de invloed van zowel de resolutie als het aantal dieptemappen stagneert, kan een gebied worden gedefinieerd waarvoor het aantal foutieve pixels nagenoeg constant blijft. De nauwkeurigheid van de coherente schaduwmappen wordt dus beïnvloed door een combinatie van de resolutie en het aantal dieptemappen. Het verband tussen de resolutie en het aantal dieptemappen is te zien op de driedimensionale grafieken van figuur 6.2 en 6.3. Op de grafiek is te zien dat naarmate de resolutie en het aantal dieptemappen toenemen, er een gebied (een vlak) ontstaat waarvoor het aantal foutieve pixels stagneert. Dit gebied is op de grafiek aangeduid met een roze rand.

Aangezien alle punten binnen dit gebied nagenoeg over dezelfde nauwkeurigheid beschikken, kan dus op zoek worden gegaan naar het punt waarvoor de bestandsgrootte van de coherente schaduwmappen minimaal is. Voor de Stanford draak komt dit punt overeen met 4096 dieptemappen en een resolutie van 128×128 pixels. Deze redenering kan worden opgebouwd voor elk object.

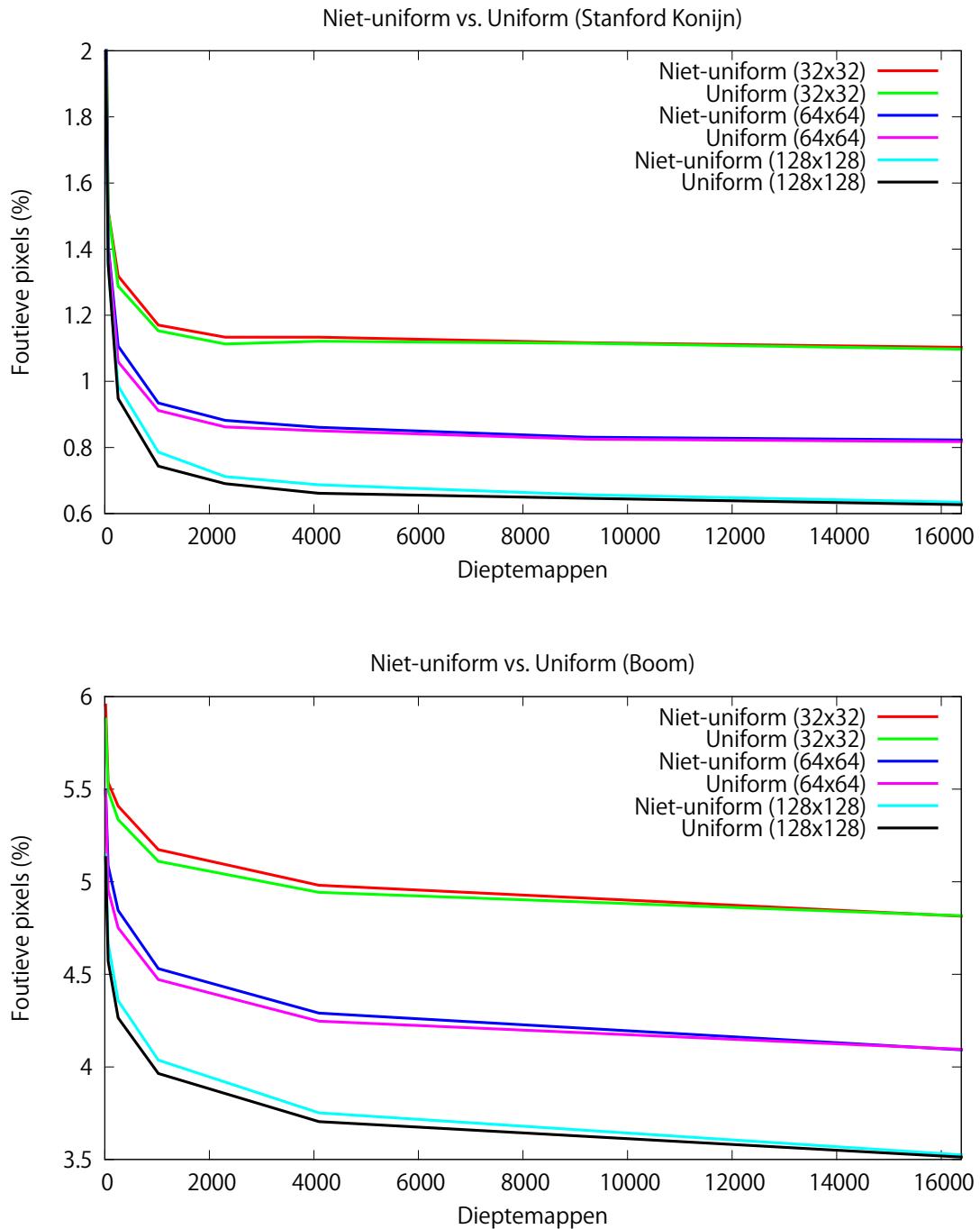
6.4 Invloed van Uniforme Positiebepaling

Voor het bepalen van de posities van de dieptemappen op de omhullende sfeer, kan zowel een uniforme als niet-uniforme verdeling worden gebruikt. De objecten die werden getest zijn het Stanford konijn en de boom. Figuur 6.4 toont de resultaten voor beide objecten.

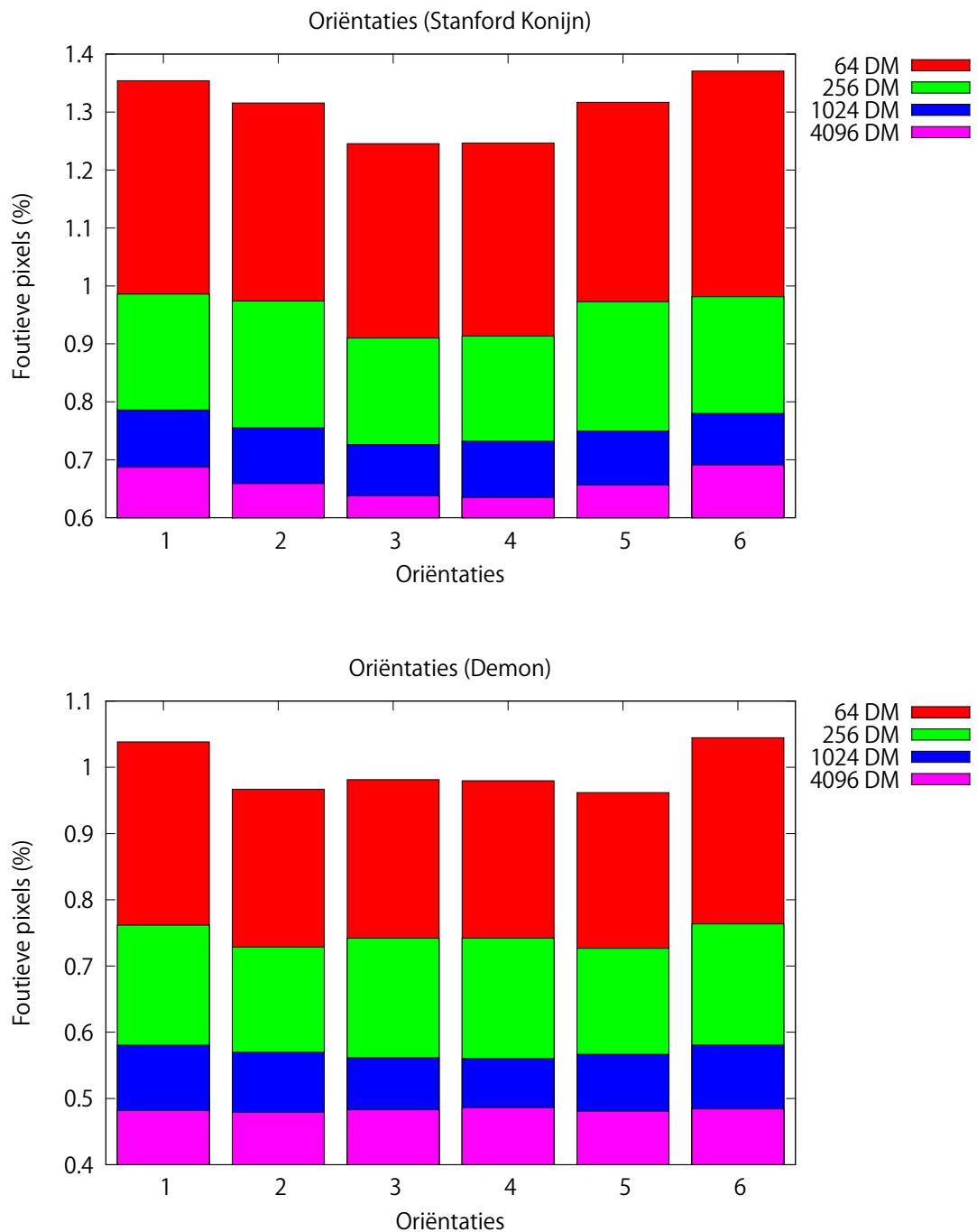
Uit de grafieken kan worden afgeleid dat voor een uniforme verdeling een hogere nauwkeurigheid wordt bekomen. Dit komt omdat bij een uniforme verdeling de dieptemappen beter gespreid zijn over het oppervlak van de omhullende sfeer. Daardoor wordt een betere benadering van de visibiliteitsinformatie opgesteld. Dit effect neemt echter af naarmate het aantal dieptemappen toeneemt. Voor een groot aantal dieptemappen maakt het immers weinig uit welke verdeling wordt gebruikt. De afstand tussen de posities op de omhullende sfeer wordt dan zo klein dat zowel bij een uniforme als niet-uniforme verdeling de visibiliteitsinformatie vanuit alle mogelijke richtingen wordt bekomen. Op de grafiek is dit duidelijk te zien. Naarmate het aantal dieptemappen toeneemt wordt het verschil tussen beide delingen kleiner.

6.5 Invloed van Oriëntatie

Ten slotte wordt nagegaan welke invloed de oriëntatie van de omhullende sfeer heeft op de nauwkeurigheid van coherente schaduwmappen. De zes oriëntaties die werden getest zijn dezelfde als bij de compressie. Daarbij werden de polen van de omhullende sfeer zo geroteerd dat ze werden gealigneerd met de x -, y - of z -as. Voor elk van deze



Figuur 6.4: Invloed van een (niet-)uniforme verdeling op het aantal foutieve pixels. Naarmate het aantal dieptemappen toeneemt daalt het kwaliteitsverschil tussen beide verdelingen. Dit komt omdat de afstand tussen de dieptemappen daalt naarmate meer dieptemappen worden geconstrueerd. Bijgevolg wordt voor elke mogelijke richting wel een dieptemap geconstrueerd, ongeacht de verdeling.



Figuur 6.5: Invloed van oriëntatie van de omhullende sfeer op het aantal foutieve pixels. De oriëntaties die worden bestudeerd zijn (van links naar rechts): polen van de sfeer gealigneerd op de z -as, de y -as, de x -as, de x -as + 180° , de y -as + 180° en de z -as + 180° .

opstellingen werd ook nog de variant getest waarbij de polen 180° waren geroteerd. De objecten die werden getest zijn het Stanford konijn en de demon. De resultaten werden bekomen voor een resolutie van 128×128 pixels en een uniforme verdeling van de dieptemappen. Deze resultaten zijn terug te vinden op figuur 6.5.

Voor zowel het Stanford konijn als de demon blijken de oriëntaties een invloed te hebben op de nauwkeurigheid. Dit effect is echter afhankelijk van zowel het object als het aantal dieptemappen waarvoor coherente schaduwmappen worden geconstrueerd. Voor het Stanford konijn leiden de oriëntaties rond de x -as tot de hoogste nauwkeurigheid, voor de demon zijn dit de oriëntaties rond de y -as. Voor het Stanford konijn blijft dit effect ook aanwezig naarmate het aantal dieptemappen toeneemt terwijl dit niet het geval is voor de demon. Het is met andere woorden moeilijk om de invloed van oriëntaties op een logische manier te beschrijven. Merk ook op dat er voor de verschillende oriëntaties geen verband bestaat tussen kwaliteit en compressiefactor (sectie 5.3.5).

6.6 Conclusie

In dit hoofdstuk werd de nauwkeurigheid van coherente schaduwmappen onderzocht. Er werd nagegaan wat de invloed is van verschillende parameters op deze nauwkeurigheid.

Uit de resultaten blijkt dat de resolutie en het aantal dieptemappen een belangrijk rol spelen. Met een hogere resolutie wordt immers een gedetailleerder beeld van de visibiliteitsinformatie bekomen. Een hoger aantal dieptemappen zorgt er dan weer voor dat er altijd wel een dieptemap kan worden gevonden die overeenkomt met de richting van het invallend licht. Beide parameters oefenen ook een invloed uit op elkaar. Voor een laag aantal dieptemappen speelt de resolutie bijvoorbeeld geen rol en vice versa. Daardoor kan voor beide parameters een minimum worden bepaald waarvoor een nauwkeurige benadering van de visibiliteitsinformatie wordt gegarandeerd. Dit punt komt tevens overeen met het punt waarvoor de bestandsgrootte van de coherente schaduwmappen minimaal is.

Voor het bepalen van de posities van de dieptemappen op de omhullende sfeer kan zowel een uniforme als niet-uniforme verdeling worden gebruikt. Voor een laag aantal dieptemappen wordt een hogere nauwkeurigheid bekomen met een uniforme verdeling. Naarmate het aantal dieptemappen stijgt, verdwijnt echter het verschil tussen een uniforme en niet-uniforme verdeling. Voor een groot aantal dieptemappen maakt het bijgevolg weinig uit welke verdeling wordt gebruikt.

Er werd ook onderzocht of een hogere nauwkeurigheid kan worden bereikt door de omhullende sfeer te roteren. De invloed hiervan is echter niet altijd even duidelijk. Aangezien de verschillen tussen de oriëntaties ook klein zijn, kan worden besloten dat de oriëntatie van weinig belang is voor de nauwkeurigheid van coherente schaduwmappen.

Ook de geometrie van een object is bepalend voor de nauwkeurigheid van de coherente schaduw mappen. Voor objecten die zeer veel details bevatten of over een sterk variërend oppervlak beschikken, zullen minder nauwkeurige resultaten worden bekomen. Om de detailinformatie van deze objecten te beschrijven moeten hogere resoluties van de dieptemappen worden gebruikt. Voor objecten met een eerder zachtverlopend oppervlak of zelfs met een zeer dun oppervlak (demon) kan de visibiliteitsinformatie echter wel op een nauwkeurige manier worden bepaald. De mate waarin een object kan worden ge-comprimeerd, is dus niet altijd bepalend voor de nauwkeurigheid van de gegenereerde schaduwen.

Hoofdstuk 7

Besluit

7.1 Overzicht

Schaduwen vormen een essentieel onderdeel bij het visualiseren van virtuele scènes. In dit eindwerk werd de techniek van coherente schaduwassen onderzocht. Met behulp van coherente schaduwassen kunnen realistische schaduwen worden weergegeven voor interactieve toepassingen.

7.1.1 Compressie

Coherente schaduwassen maken gebruik van de coherentie tussen schaduwassen, om deze op een compacte en foutloze manier voor te stellen. In hoofdstuk 5 werd onderzocht wat de invloed is van verschillende parameters op de efficiëntie van deze compressietechniek.

Zo werd nagegaan wat de invloed is op de compressiefactor, voor de resolutie of het aantal dieptemappen dat wordt geconstrueerd. Er werd vastgesteld dat de compressiefactor toeneemt, wanneer een hogere resolutie of een groter aantal dieptemappen wordt gebruikt. Om een maximale compressie te bekomen werden bovendien verschillende ruimtevullende curves bestudeerd die kunnen worden gebruikt om een coherente sequentie op te stellen. Uit de resultaten die werden bekomen, kon worden besloten dat per-pixel compressie een zeer efficiënte compressietechniek is om dieptemappen te comprimeren. De visibiliteitsinformatie van verscheidene objecten kan met deze compressietechniek namelijk op een compacte manier worden voorgesteld. Enkel voor objecten met een sterk variërend of zeer dun oppervlak, blijkt per-pixel compressie minder efficiënt te zijn.

In hoofdstuk 5 werd ook onderzocht of andere, meer compacte, voorstellingen van schaduwassen mogelijk zijn. Daarbij werden verschillende technieken bedacht die de coherentie tussen schaduwassen op een andere manier uitbuiten dan bij per-pixel compressie. Voor twee van deze technieken, per-map compressie en driedimensionale Hilbert

compressie, werden resultaten bekomen die in de buurt liggen van per-pixel compressie. In specifieke gevallen werd zelfs een hogere compressiefactor bekomen dan met per-pixel compressie.

7.1.2 Kwaliteit

In hoofdstuk 6 werd bestudeerd hoe nauwkeurig de schaduwen zijn die met coherente schaduwassen worden berekend. Daarbij werd onderzocht wat de invloed is van verscheidene parameters op de kwaliteit van de schaduwen, zoals het aantal of de resolutie van de dieptemappen.

Verscheidene resultaten werden bekomen. Zo bleek een uniforme verdeling van de dieptemappen op de omhullende sfeer, nauwkeurigere resultaten op te leveren dan bij een niet-uniforme verdeling. Ook het verband tussen resolutie en het aantal dieptemappen werd onderzocht. Daaruit bleek dat beide een invloed op elkaar uitoefenen. Naarmate de resolutie en het aantal dieptemappen toeneemt, kan een gebied worden bepaald waarvoor de nauwkeurigheid nagenoeg constant blijft. Met behulp van dit gebied kan tevens een punt worden bepaald waarvoor de bestandsgrootte van de coherente schaduwassen minimaal is. Op deze manier kunnen de waardes voor resolutie en het aantal dieptemappen worden bepaald, waarvoor nauwkeurige schaduwen worden gegarandeerd, en de bestandsgrootte van de coherente schaduwassen minimaal is.

7.2 Toekomstig Werk

Een mogelijke uitbreiding zou zijn om de techniek van coherente schaduwassen, te vergelijken met andere technieken om realistische schaduwen te berekenen. Een beperking voor interactieve toepassingen is dat coherente schaduwassen enkel kunnen worden gebruikt voor rigide objecten. Daarnaast nemen de coherente schaduwassen ook veel ruimte in beslag. Voor een scène met meerdere objecten betekent dit dat er veel geheugenruimte moet worden voorzien. Daarom kan het interessant zijn om na te gaan op welke manier andere technieken met deze problemen omgaan.

Bijlage A

Resultaten Per-Pixel Compressie

Dieptemappen	Ongecomprimeerd	Gecomprimeerd	Ratio
16	2 MB	0,33 MB	6.0:1
64	8 MB	0,38 MB	11:1
256	32 MB	0,58 MB	55:1
1.024	128 MB	1,3 MB	98:1
4.096	512 MB	3,6 MB	143:1
16.384	2.048 MB	11 MB	184:1
65.536	8.192 MB	37 MB	221:1

Tabel A.1: Resultaten van per-pixel compressie voor de sfeer.

Dieptemappen	Ongecomprimeerd	Gecomprimeerd	Ratio
16	2 MB	0,56 MB	3,5:1
64	8 MB	1,0 MB	3,8:1
256	32 MB	2,2 MB	15:1
1.024	128 MB	4,8 MB	27:1
4.096	512 MB	11 MB	46:1
16.384	2.048 MB	26 MB	79:1
65.536	8.192 MB	64 MB	128:1

Tabel A.2: Resultaten van per-pixel compressie voor de Venus.

Dieptemappen	Ongecomprimeerd	Gecomprimeerd	Ratio
16	2 MB	0,63 MB	3,2:1
64	8 MB	1,31 MB	6,18:1
256	32 MB	3,09 MB	10:1
1.024	128 MB	7,57 MB	17:1
4.096	512 MB	19,1 MB	266:1
16.384	2.048 MB	50,6 MB	41:1
65.536	8.192 MB	142 MB	57:1

Tabel A.3: Resultaten van per-pixel compressie voor het Stanford konijn.

Dieptemappen	Ongecomprimeerd	Gecomprimeerd	Ratio
16	2 MB	0.92 MB	2,1:1
64	8 MB	1.7 MB	2,4:1
256	32 MB	4.9 MB	6,6:1
1.024	128 MB	15 MB	8,6:1
4.096	512 MB	43 MB	12:1
16.384	2.048 MB	107 MB	19:1
65.536	8.192 MB	269 MB	30:1

Tabel A.4: Resultaten van per-pixel compressie voor de demon.

Dieptemappen	Ongecomprimeerd	Gecomprimeerd	Ratio
16	2 MB	1,3 MB	1,5:1
64	8 MB	2,5 MB	1,6:1
256	32 MB	8,6 MB	3,7:1
1.024	128 MB	32 MB	4,0:1
4.096	512 MB	117 MB	4,3:1
16.384	2.048 MB	437 MB	4,7:1

Tabel A.5: Resultaten van per-pixel compressie voor de boom.

Dieptemappen	Resolutie	Data	Per-pixel	Ratio
1.024	32×32	8 MB	0,57 MB	14:1
1.024	64×64	32 MB	1,99 MB	16:1
1.024	128×128	128 MB	7,57 MB	17:1
1.024	256×256	512 MB	30,0 MB	17:1
4.096	32×32	32 MB	1,65 MB	19:1
4.096	64×64	128 MB	5,26 MB	24:1
4.096	128×128	512 MB	19,1 MB	27:1
4.096	256×256	2.048 MB	73,9 MB	28:1
16.384	32×32	128 MB	5,10 MB	25:1
16.384	64×64	512 MB	14,8 MB	35:1
16.384	128×128	2.048 MB	50,6 MB	41:1
16.384	256×256	8.192 MB	190 MB	43:1

Tabel A.6: Invloed op de compressiefactor voor verschillende resoluties en een verschillend aantal dieptemappen. Dit zijn de resultaten die werden bekomen voor het Stanford konijn.

Dieptemappen	Resolutie	Data	Per-pixel	Ratio
1.024	32×32	8 MB	1,0 MB	8,0:1
1.024	64×64	32 MB	3,8 MB	8,4:1
1.024	128×128	128 MB	15 MB	8,6:1
1.024	256×256	512 MB	60 MB	8,6:1
4.096	32×32	32 MB	3,1 MB	10:1
4.096	64×64	128 MB	11 MB	11:1
4.096	128×128	512 MB	43 MB	12:1
4.096	256×256	2.048 MB	168 MB	12:1
16.384	32×32	128 MB	9,0 MB	14:1
16.384	64×64	512 MB	30 MB	17:1
16.384	128×128	2.048 MB	107 MB	19:1
16.384	256×256	8.192 MB	410 MB	20,0:1

Tabel A.7: Invloed op de compressiefactor voor verschillende resoluties en een verschillend aantal dieptemappen. Dit zijn de resultaten die werden bekomen voor de demon.

Bijlage B

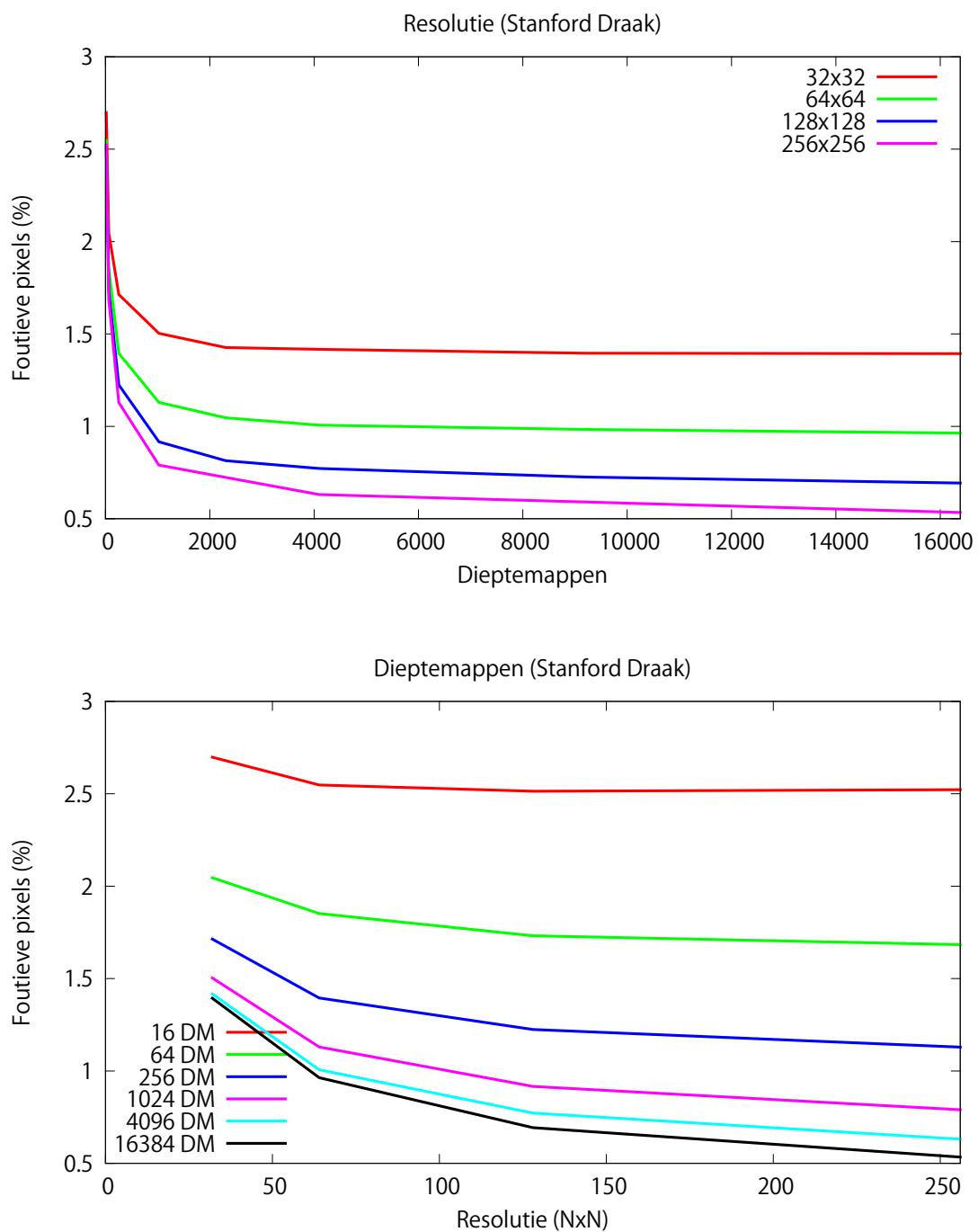
Resultaten Kwaliteitsstudie

Dieptemappen	Resolutie	Foutieve Pixels	$\Delta_{dieptemappen}$	$\Delta_{resolutie}$
256	32×32	1,71%	n.v.t.	n.v.t.
1.024	32×32	1,50%	-12,3%	n.v.t.
4.096	32×32	1,42%	-5,74%	n.v.t.
16.384	32×32	1,39%	-1,66%	n.v.t.
256	64×64	1,39%	n.v.t.	-18,6%
1.024	64×64	1,13%	-19,0%	-24,9%
4.096	64×64	1,01%	-10,9%	-29,0%
16.384	64×64	0,96%	-4,22%	-30,8%
256	128×128	1,22%	n.v.t.	-12,2%
1.024	128×128	0,92%	-25,1%	-18,9%
4.096	128×128	0,77%	-15,8%	-23,3%
16.384	128×128	0,69%	-10,2%	-28,1%
256	256×256	1,13%	n.v.t.	-7,78%
1.024	256×256	0,79%	-30,0%	-13,8%
4.096	256×256	0,63%	-20,1%	-18,2%
16.384	256×256	0,53%	-15,4%	-23,0%

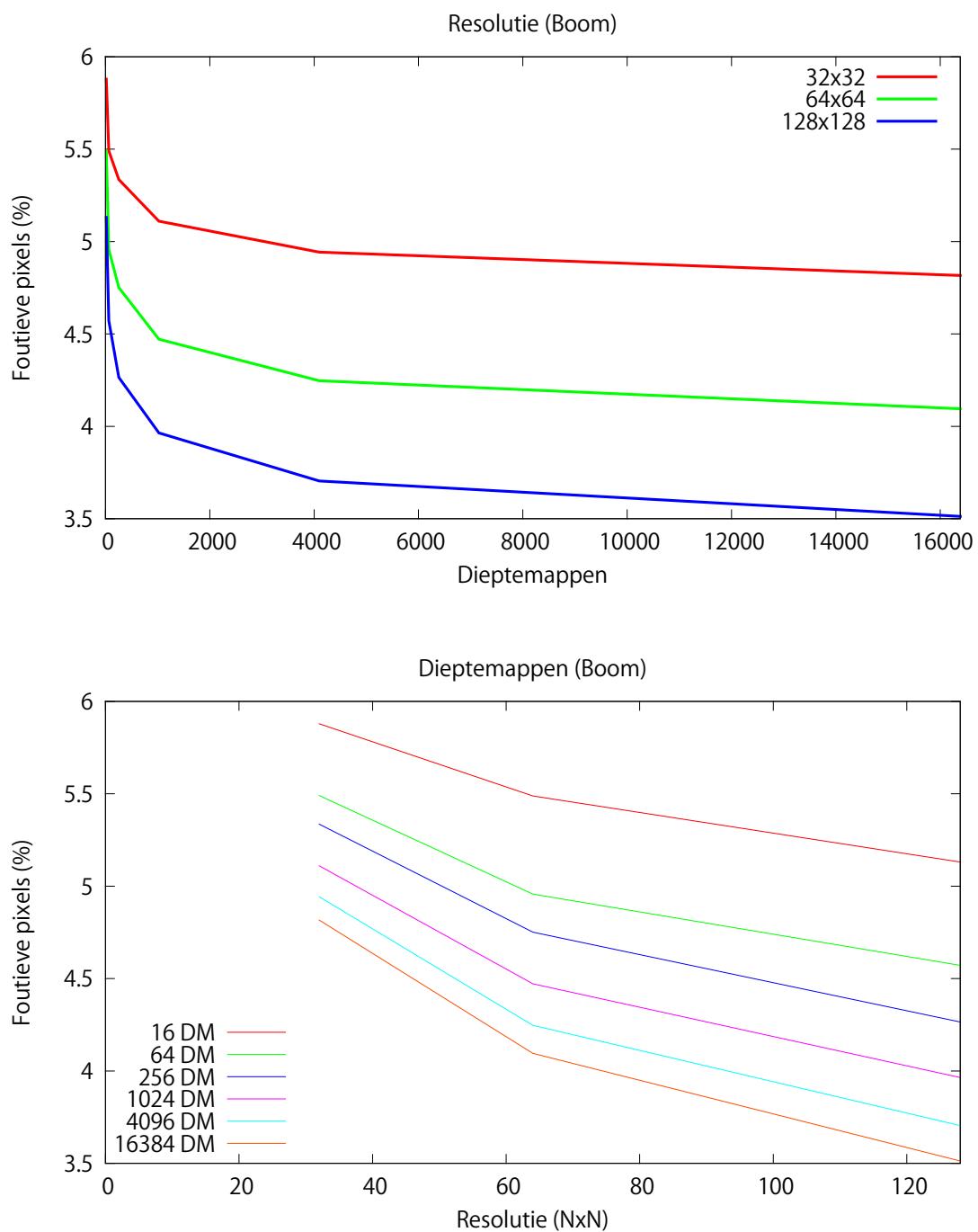
Tabel B.1: Verband tussen resolutie en dieptemappen voor de Stanford draak. $\Delta_{dieptemappen}$ beschrijft de relatieve wijziging van het aantal foutieve pixels voor een toenemend aantal dieptemappen en een vaste resolutie. $\Delta_{resolutie}$ beschrijft de relatieve wijziging voor een toenemende resolutie en een vast aantal dieptemappen.

Dieptemappen	Resolutie	Foutieve Pixels	$\Delta_{dieptemappen}$	$\Delta_{resolutie}$
256	32×32	5,34%	n.v.t.	n.v.t.
1.024	32×32	5,11%	-4,24%	n.v.t.
4.096	32×32	4,94%	-3,28%	n.v.t.
16.384	32×32	4,82%	-2,55%	n.v.t.
256	64×64	4,75%	n.v.t.	-11,0%
1.024	64×64	4,47%	-5,87%	-12,5%
4.096	64×64	4,25%	-5,04%	-14,0%
16.384	64×64	4,09%	-3,57%	-15,0%
256	128×128	4,27%	n.v.t.	-10,2%
1.024	128×128	3,96%	-7,05%	-11,4%
4.096	128×128	3,70%	-6,55%	-12,8%
16.384	128×128	3,51%	-5,18%	-14,2%

Tabel B.2: Verband tussen resolutie en dieptemappen voor de boom. $\Delta_{dieptemappen}$ beschrijft de relatieve wijziging van het aantal foutieve pixels voor een toenemend aantal dieptemappen en een vaste resolutie. $\Delta_{resolutie}$ beschrijft de relatieve wijziging voor een toenemende resolutie en een vast aantal dieptemappen.



Figuur B.1: Invloed van de resolutie en het aantal dieptemappen voor de Stanford draak.



Figuur B.2: Invloed van de resolutie en het aantal dieptemappen voor de boom.

Bibliografie

- [CL96] Curless B. and Levoy M. A Volumetric Method for Building Complex Models from Range Images. In *Proceedings of ACM SIGGRAPH*, pages 303–312. 1996.
- [DBB06] Dutré P., Bala K. and Bekaert P. *Advanced Global Illumination*. A K Peters, September 2006.
- [FK03] Fernando R. and Kilgard M.J. *The Cg Tutorial*. Addison-Wesley, March 2003.
- [Ham06] Hamilton C. Compact Hilbert Indices. Technical report, Dalhousie University, July 2006.
- [Hil91] Hilbert D. Über die stetige Abbildung einer Linie auf ein Flächenstück. *Mathematische Annalen* 38, page 459, 1891.
- [Llo82] Lloyd S.P. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28:129–137, 1982.
- [MJFS01] Moon B., Jagadish H., Faloutsos C. and Saltz. J.H. Analysis of the Clustering Properties of the Hilbert Space-Filling Curve. *IEEE Transactions on Knowledge and Data Engineering*, pages 124–141, January 2001.
- [Pea90] Peano G. Sur une Courbe qui Remplit toute une Aire Plaine. *Mathematische Annalen* 36, page 157, 1890.
- [RGKM07] Ritschel T., Grosch T., Kautz J. and Müller S. Interactive Illumination with Coherent Shadow Maps. In *Proceedings of Eurographics Symposium on Rendering*, pages 61–72. June 2007.
- [RSC87] Reeves W.T., Salesin D.H. and Cook R.L. Antialiased Shadows with Depth Maps. In *Proceedings of ACM SIGGRAPH*, pages 283–291. July 1987.
- [SDS95] Stollnitz E., DeRose T. and Salesin D. Wavelets for Computer Graphics: A Primer, Part 1. *IEEE Computer Graphics and Applications*, pages 76–84, May 1995.
- [Shi05] Shirley P. *Fundamentals of Computer Graphics*. A K Peters, July 2005.

- [Suf07] Suffern K. *Ray Tracing from the Ground Up*. A K Peters, September 2007.
- [SWND07] Shreiner D., Woo M., Neider J. and Davis T. *OpenGL Programming Guide*. Addison-Wesley, August 2007.
- [TL94] Turk G. and Levoy M. Zippered Polygon Meshes from Range Images. In *Proceedings of ACM SIGGRAPH*, pages 311–318. 1994.
- [Voo91] Voorhies D. Space-Filling Curves and a Measure of Coherence. In J. Arvo, editor, *Graphics Gems II*, pages 26–30. Academic Press, 1991.
- [WE03] Weiskopf D. and Ertl T. Shadow Mapping Based on Dual Depth Layers. In *Eurographics 2003 Short Papers*, pages 53–60. July 2003.
- [Wil78] Williams L. Casting Curved Shadows on Curved Surfaces. In *Proceedings of ACM SIGGRAPH*, pages 270–274. August 1978.

Lijst van figuren

2.1	Een driedimensionaal model van de theepot van Utah. Het volledige model is opgebouwd uit vierhoeken (links) en kan worden gevisualiseerd (rechts).	4
2.2	Vaak gebruikte modellen om lichtbronnen te simuleren.	6
2.3	De Bidirectionele Reflectantie Distributie Functie $f_r(x, \Psi \rightarrow \Theta)$ beschrijft de interactie van licht met een materiaal in het punt x . Licht afkomstig uit richting Ψ wordt gereflecteerd in richting Θ	7
2.4	Drie types van materialen: diffuus, reflectief en glanzend.	8
2.5	Twee verschillende modellen om materialen te beschrijven.	9
2.6	De grafische pijplijn. Vertices worden eerst getransformeerd naar de gewenste positie en oriëntatie. Daarna worden polygonen gebouwd en gerasterd zodat fragmenten ontstaan. De kleurwaarde voor elk fragment wordt berekend door interpolatie van de hoekpunten. Ten slotte worden raster operaties toegepast om de uiteindelijke kleurwaarde te bekomen.	10
2.7	Ray tracing. Vanuit het kijkpunt worden zichtstralen afgeschoten. Bij intersectie met een object wordt nagegaan wat de radiantie is voor dit snijpunt.	11
2.8	Eerst wordt de modeltransformatie toegepast op objecten in de scène (links). Daarna wordt de weergavetransformatie toegepast waardoor de camera wordt gepositioneerd op de oorsprong. Alle objecten worden bovendien mee getransformeerd (midden). De volledige scène na het toepassen van model- en weergave transformatie (rechts).	12
2.9	Bij orthografische projectie wordt een balk als weergavevolume gebruikt. Alle projectielijnen zijn parallel en staan loodrecht op het projectievak.	14
2.10	Perspectief: het weergavevolume is een afgeknotte piramide.	14
2.11	Punten worden geprojecteerd in de richting van het kijkpunt, namelijk waar de perspectieflijnen snijden met het beeldvlak. De projectie van diezelfde punten kan ook worden bekomen met een orthografische projectie.	15
2.12	Het canoniek weergavevolume.	16
2.13	Peano curve van eerste en tweede orde. Deze curves bepalen een ordening voor respectievelijk 3^2 en 3^4 punten.	18
2.14	Tweedimensionale Hilbert curve van eerste, tweede en derde orde. Deze curves bepalen een ordening voor respectievelijk 2^2 , 2^4 en 2^6 punten.	18

2.15 Tweedimensionale zigzag curve. Links: horizontaal. Rechts: verticaal.	19
3.1 Een scène zonder schaduwen (links) is voor misinterpretatie vatbaar. De bollen kunnen op het vlak liggen (midden) of boven het vlak zweven (rechts).	20
3.2 Links: meetkundige voorstelling van umbra en penumbra. Rechts: umbra en penumbra voor een oppervlaktelichtbron.	21
3.3 Een dieptemap van een scène. Zwarde pixels geven punten aan die dichtbij zijn gelegen. Witte pixels zijn punten veraf.	22
3.4 Visibiliteitstest voor een punt in schaduw. Het punt p wordt geprojecteerd naar het standpunt van de lichtbron. De dieptewaarde z_p wordt dan vergeleken met de dieptewaarde z_d die is opgeslagen in de dieptemap. Aangezien $z_d < z_p$ ligt het punt p in schaduw.	23
3.5 Visibiliteitstest voor een punt dat wordt belicht. Het punt p wordt geprojecteerd naar het standpunt van de lichtbron. De dieptewaarde z_p wordt dan vergeleken met de dieptewaarde z_d die is opgeslagen in de dieptemap. Aangezien $z_d = z_p$ wordt het punt p belicht.	24
3.6 Naarmate de resolutie van de dieptemappen toeneemt, stijgt ook de kwaliteit van de schaduwen.	25
3.7 Percentage Closer Filtering (PCF). Waardes uit de dieptemap worden vergeleken met de z -waarde op het oppervlak (links). Het slaagpercentage van de visibiliteitstesten (midden) bepaalt de schaduwcoëfficient (rechts) [RSC87].	26
3.8 Links: zonder PCF. Rechts: met PCF.	26
3.9 Indien geen bias wordt gekozen treedt er zelfbeschaduwing op (links). Het toevoegen van een kleine factor aan de dieptewaarden verhindert dit (midden). Indien de bias echter te groot wordt gekozen, dan zijn de verschuivingen van de schaduwen duidelijk zichtbaar (rechts).	27
3.10 Duale dieptelagen. Voor het correct verlopen van de visibiliteitstest mag elke waarde tussen z_1 en z_2 worden gekozen, bijvoorbeeld z_{gem} . Alle punten in schaduw ($z_{2,3,4}$) bevinden zich nog altijd verder dan z_{gem} . Ook voor andere punten (z_1) wijzigt het resultaat van de visibiliteitstest niet [RGKM07].	27
4.1 Om de visibiliteitsinformatie van een object te verzamelen worden dieptemappen geconstrueerd op allerhande posities rondom dit object [RGKM07].	30
4.2 Tijdens het visualiseren van de scène worden voor elk punt meerdere visibiliteitstesten uitgevoerd. Telkens wordt de dieptemap gebruikt die het best overeenkomt met de richting van het invallend licht.	31

4.3	Positiebepaling op de omhullende sfeer voor niet-uniforme bemonstering. Er wordt vertrokken van N reguliere monsters binnen het domein $[0, 1]^2$. Deze worden vervolgens gescaleerd naar $[0, \pi] \times [0, 2\pi]$. Samen met de afstand r stellen ze sferische coördinaten voor. Transformatie naar driedimensionale cartesische coördinaten levert de positie op voor elke dieptemap op de omhullende sfeer van het object. Elk punt op de omhullende sfeer kan bovendien worden terug getransformeerd naar een punt binnen het domein $[0, 1]^2$	33
4.4	De overeenkomstige dieptemap i wordt opgezocht aan de hand de lichtvector Ψ . Deze dieptemap wordt vervolgens gebruikt om de visibiliteitstest uit te voeren. Links: het punt p ligt buiten het bereik van de dieptemap en wordt dus belicht. Rechts: het punt p ligt binnen het bereik van de dieptemap. Net als bij het klassieke schaduwmap algoritme, wordt de dieptewaarde voor het punt p berekend en vergeleken met de overeenkomstige waarde z_d in de dieptemap. Aangezien $z_d < z_p$ ligt het punt p in de schaduw van het object.	34
4.5	De lichtvector Ψ beschrijft de dieptemap die wordt gezocht. De lichtvector Ψ wordt eerst getransformeerd naar sferische coördinaten $(r_\Psi, \theta_\Psi, \phi_\Psi)$. Met behulp van θ_Ψ en ϕ_Ψ kan de overeenkomstige dieptemap worden gevonden (zie tekst).	35
4.6	Visualisaties voor verschillende configuraties van coherente schaduwmaps. Van links naar rechts: Een toenemende resolutie betekent dat meer details kunnen worden weergegeven. Dit effect is vooral zichtbaar voor een laag aantal dieptemappen. De stekels op het hoofd van de Stanford draak bijvoorbeeld zijn niet herkenbaar op dieptemappen met een resolutie van 32×32 pixels. Van boven naar onder: Het aantal dieptemappen heeft een sterke invloed op de kwaliteit van de schaduwen. Voor een te laag aantal zijn de individuele dieptemappen nog zichtbaar. Het is pas vanaf 16384 dieptemappen dat dit effect langzaamaan verdwijnt.	37
4.7	Visualisaties van de fouten ($\times 10$) voor verschillende configuraties van coherente schaduwmaps. Naarmate het aantal dieptemappen en de resolutie toenemen, daalt het aantal fouten. Een hoger aantal dieptemappen heeft als voornaamste gevolg dat de overgangen tussen de individuele dieptemappen minder zichtbaar worden. Een hogere resolutie doet het aantal fouten over het gehele oppervlak van de schaduw dalen. Voor lage resoluties blijken er gebieden te zijn waar opvallend meer fouten voorkomen. Voor 32×32 pixels is dit onder andere het gebied waar de schaduw van het hoofd van de draak zich bevindt. Deze fouten zijn het gevolg van visibiliteitsinformatie die niet nauwkeurig genoeg is.	38

4.8 Coherente schaduwmappen voor twee verschillende scènes. Telkens worden voor elk object in de scène 36864 dieptemappen geconstrueerd met een resolutie van 128×128 pixels. Boven: visualisatie van een boom. De schaduwen die worden bekomen zijn blokkerig en ongedetailleerd. Dit is het gevolg van een te lage resolutie voor dit object. Ditzelfde effect is ook te zien op figuur 4.6 voor een lage resolutie van de dieptemappen. Onder: realistische schaduwen voor een knuffelbeer met bal. Voor beide objecten werden coherente schaduw mappen geconstrueerd. De objecten hebben een zachtverlopend oppervlak, waardoor de kwaliteit van de schaduwen hoger ligt dan bij de boom.	40
4.9 Coherente schaduw mappen kunnen worden gecombineerd met percentage closer filtering. Door meerdere visibiliteitstesten uit te voeren, daalt het aantal fouten. Deze resultaten werden bekomen voor 16384 dieptemappen met een resolutie van 128×128 pixels. De fouten zijn het laagst indien 16 visibiliteitstesten worden uitgevoerd voor één dieptemap.	41
5.1 Visibiliteitstest voor een punt in schaduw. Het punt p wordt geprojecteerd naar het standpunt van de lichtbron. De dieptewaarde z_p wordt dan vergeleken met de dieptewaarde z_d die is opgeslagen in de dieptemap. De waarde van z_d kan vrij worden gekozen tussen z_1 en z_2 . Zolang $z_d \in [z_1, z_2]$ blijft de visibiliteitstest correct, aangezien z_d altijd kleiner is dan z_p . Bijgevolg ligt het punt p in schaduw.	44
5.2 Visibiliteitstest voor een punt dat niet in schaduw ligt. Het punt p wordt geprojecteerd naar het standpunt van de lichtbron. De dieptewaarde z_p wordt dan vergeleken met de dieptewaarde z_d die is opgeslagen in de dieptemap. De waarde van z_d kan vrij worden gekozen tussen $z_1 = z_p$ en z_2 . Zolang $z_d \in [z_1, z_2]$ blijft de visibiliteitstest correct, aangezien z_d altijd groter is dan z_p of gelijk is aan z_p . Bijgevolg ligt het punt p niet in schaduw.	45
5.3 Dieptemappen voor drie objecten: De Stanford draak [CL96], een standbeeld van Venus en een torus. Boven: de klassieke dieptemap bevat de z_1 -waardes. Onder: de duale dieptemap bevat de z_2 -waardes.	46
5.4 Per-pixel compressie. Dieptemappen worden na elkaar geplaatst en alle dieptewaarden voor een specifieke pixel worden gecomprimeerd.	46
5.5 Per-pixel compressie voor een specifieke pixel. Door de z_1 - en z_2 -waardes te vergelijken worden gemeenschappelijke dieptewaarden g_i gevonden. Deze stellen stuksgewijze constante functies voor. De gemeenschappelijke dieptewaarden worden opgeslagen in segmenten (g_i, n_i) . De index n_i houdt bij hoeveel dieptemappen gebruik maken van de overeenkomstige dieptewaarde g_i	47

5.6	Aangezien elke dieptemap op een unieke wijze wordt voorgesteld door een punt in de tweedimensionale ruimte, kan een tweedimensionale Hilbert curve worden gebruikt om een coherente sequentie van de dieptemappen te bepalen. De coherente tussen de punten wordt bewaard bij het transformeren van de punten naar posities op de omhullende sfeer. Punten die dicht bij elkaar liggen in de tweedimensionale ruimte, liggen dus ook dicht bij elkaar op de omhullende sfeer [RGKM07].	48
5.7	Een tweedimensionale lineaire curve. Links: horizontaal. Rechts: verticaal.	49
5.8	Voorbeelden van een gedecomprimeerde dieptewaarde. Voor elke pixel wordt de overeenkomstige dieptewaarde opgezocht in de coherente schaduwmappen. Als resultaat wordt een dieptemap bekomen waarop de invloed van andere dieptemappen zichtbaar is. Dit vertaalt zich in herhalingen van andere dieptemappen. Links: een standbeeld van Venus. Rechts: een knuffelbeer.	50
5.9	Voor het construeren van coherente schaduwmappen werden zes objecten gebruikt. Van links naar rechts en van boven naar onder: een sfeer, een standbeeld van Venus, de Stanford draak [CL96], het Stanford konijn [TL94], een demon en een boom. Alle objecten zijn opgebouwd uit driehoeken, voor de sfeer wil dit zeggen dat slechts een benadering van het echte oppervlak wordt bekomen.	51
5.10	Resultaten van per-pixel compressie voor verschillende objecten. Op de grafiek is te zien dat met per-pixel compressie een hoge compressiefactor wordt bekomen. Voor objecten met een sterk variërende geometrie of een geometrie met veel fijne structuren, blijkt per-pixel compressie minder efficiënt te zijn. De geometrie van een object heeft dus een invloed op de compressie. De compressiegraad stijgt ook toe naarmate de resolutie of het aantal dieptemappen toeneemt. De reden waarom er geen perfecte coherente is voor de sfeer, is te wijten aan de geometrische benadering die voor dit object werd gebruikt.	52
5.11	Invloed op de compressiefactor voor verschillende ruimtevullende curves. Boven: de resultaten van de demon. Onder: de resultaten van de Stanford draak. Voor beide wordt een gelijkaardig resultaat bekomen. De horizontale zigzag curve geeft telkens het beste resultaat.	54
5.12	Invloed op de compressiefactor voor verschillende resoluties. Boven: de resultaten van de demon. Onder: de resultaten van het Stanford konijn. Naarmate de resolutie en/of het aantal dieptemappen toeneemt, stijgt ook de compressiefactor.	57

5.13 Invloed op de compressiefactor voor een (niet-)uniforme verdeling van de dieptemappen op de omhullende sfeer. Boven: de resultaten van het Stanford konijn. Onder: de resultaten van de boom. Op de grafieken is te zien dat een uniforme verdeling tot een lagere compressiefactor leidt. Dit is logisch, aangezien er bij een niet-uniforme verdeling meer dieptemappen worden geconstrueerd rond de polen, is er ook meer coherentie tussen deze dieptemappen aanwezig.	58
5.14 Invloed op de compressiefactor voor verschillende oriëntaties van de omhullende sfeer. Boven: de resultaten van het Stanford konijn. Onder: de resultaten van de demon. Voor elk object wordt een verschillend resultaat bekomen, dit doet vermoeden dat de oriëntatie afhankelijk is van de geometrie van het object.	60
5.15 Drie coherentiemappen gesorteerd van veel coherentie (links) naar weinig coherentie (rechts). Voor de coherentiemap van de sfeer is duidelijk te zien dat er discontinuïteiten zijn aan de randen. Deze zijn het gevolg van de geometrische benadering die voor het oppervlak van de sfeer werd gebruikt.	61
5.16 Per-map compressie aan de hand van een Hilbert curve. Door op zoek te gaan naar de gemeenschappelijke dieptewaarden (blauw) kan een dieptemap worden gecomprimeerd. Deze dieptewaarden worden vervolgens samen met een index opgeslagen in segmenten.	62
5.17 Resultaten voor verschillende ruimtevullende curves. Voor beide objecten wordt met de tweedimensionale Hilbert curve de hoogste compressiefactor bekomen.	63
5.18 Per-map compressie ten opzichte van per-pixel compressie. De efficiëntie van per-map compressie wordt bepaald door de resolutie van de dieptemappen. Afhankelijk van deze resolutie zal met per-pixel compressie sneller een hogere compressiefactor worden bekomen. De overgangspunten tussen beide compressietechnieken zijn ook zichtbaar op de grafieken.	64
5.19 K-means clustering voor de Stanford draak en 4096 dieptemappen. Boven: resultaten van k-means clustering. De originele grootte van de coherente schaduwmannen komt overeen met het geval waarbij 4096 clusters worden gebruikt. Het aantal clusters bepaalt dus hoe sterk de coherente schaduwmannen nog kunnen worden gecomprimeerd. Onder: om foutloze compressie te bekomen moet een correctie worden uitgevoerd. Daarbij wordt voor elke foute pixel ook nog de correcte dieptewaarde opgeslagen. Aangezien het aantal fouten zeer hoog ligt, is k-means clustering geen praktisch alternatief voor per-pixel compressie.	66
5.20 Voor de compressie gebaseerd op een driedimensionale Hilbert curve worden de dieptemappen verdeeld in kubussen. Elke kubus wordt vervolgens apart gecomprimeerd.	68

5.21 Compressie gebaseerd op een driedimensionale Hilbert curve. De efficiëntie van de driedimensionale Hilbert compressie wordt bepaald door de resolutie van de dieptemappen. Afhankelijk van deze resolutie zal met per-pixel compressie sneller een hogere compressiefactor worden bekomen. De overgangen tussen beide compressietechnieken zijn ook zichtbaar op de grafieken.	69
5.22 Wavelet transformatie van een vector. Het originele signaal wordt voorgesteld door het gemiddelde van elk paar coëfficiënten: [8 4] en detailcoëfficiënten [1 -2]. Deze detailcoëfficiënten stellen veelvouden van de Haar wavelet voor. De wavelet transformatie wordt vervolgens toegepast op de vector [8 4]. De volledige wavelet transformatie van een vector is dan [6 2 1 -2]. Dit resultaat beschrijft een multi-resolutie analyse van de originele vector.	71
5.23 De per-pixel gecomprimeerde data wordt op een andere manier voorgesteld vooraleer wavelet compressie wordt toegepast. Daarbij worden alle segmenten verdeeld over tweedimensionale mappen.	72
6.1 De invloed van geometrie op het aantal foutieve pixels. Objecten die veel detailinformatie bevatten introduceren opvallend meer fouten.	76
6.2 Verband tussen de resolutie en het aantal dieptemappen voor de Stanford draak. Voor een lage resolutie of een laag aantal dieptemappen, ligt het aantal foutieve pixels hoog. Naarmate het aantal dieptemappen toeneemt en de resolutie stijgt, zal het aantal foutieve pixels stagneren. Dit leidt tot een gebied waarvoor de nauwkeurigheid nagenoeg constant zal blijven. Dit gebied is op de grafiek aangeduid met een roze rand. Aan de hand van dit gebied kan tevens een punt worden bepaald, waarvoor de bestandsgrootte van de coherente schaduwmannen minimaal is. Voor de Stanford draak komt dit punt overeen met 4096 dieptemappen en een resolutie van 128×128 pixels.	78
6.3 Verband tussen de resolutie en het aantal dieptemappen voor de boom. Wegens de grote bestandsgrootte voor dit object, zijn slechts een beperkt aantal resultaten beschikbaar. Net als bij de Stanford draak lijkt het aantal foutieve pixels te stagneren naarmate het aantal dieptemappen en de resolutie toeneemt.	79
6.4 Invloed van een (niet-)uniforme verdeling op het aantal foutieve pixels. Naarmate het aantal dieptemappen toeneemt daalt het kwaliteitsverschil tussen beide verdelen. Dit komt omdat de afstand tussen de dieptemappen daalt naarmate meer dieptemappen worden geconstrueerd. Bijgevolg wordt voor elke mogelijke richting wel een dieptemap geconstrueerd, ongeacht de verdeling.	81

6.5 Invloed van oriëntatie van de omhullende sfeer op het aantal foutieve pixels. De oriëntaties die worden bestudeerd zijn (van links naar rechts): polen van de sfeer gealigneerd op de z -as, de y -as, de x -as, de x -as + 180°, de y -as + 180° en de z -as + 180°.	82
B.1 Invloed van de resolutie en het aantal dieptemappen voor de Stanford draak.	92
B.2 Invloed van de resolutie en het aantal dieptemappen voor de boom.	93

Lijst van tabellen

5.1	Resultaten van per-pixel compressie voor de Stanford draak. Hieruit kan worden afgeleid dat de compressiegraad toeneemt naarmate meer dieptemappen worden gebruikt. In het geval van 65536 dieptemappen betekent dit dat 8 GB aan data wordt voorgesteld door amper 158 MB. Per-pixel compressie is dus een behoorlijk efficiënte compressietechniek.	52
5.2	Resultaten van de vierdimensionale Hilbert compressie voor een resolutie van 128×128 pixels, 16384 dieptemappen en de Stanford draak als model. Uit de resultaten blijkt dat een compressie gebaseerd op de vierdimensionale Hilbert curve niet tot een hogere compressiefactor zal leiden.	70
5.3	Resultaten na wavelet transformatie voor een gordeldier object. Het aantal nulcoëfficiënten ligt te laag om met wavelet compressie tot een hogere compressiefactor te komen.	73
A.1	Resultaten van per-pixel compressie voor de sfeer.	87
A.2	Resultaten van per-pixel compressie voor de Venus.	87
A.3	Resultaten van per-pixel compressie voor het Stanford konijn.	88
A.4	Resultaten van per-pixel compressie voor de demon.	88
A.5	Resultaten van per-pixel compressie voor de boom.	88
A.6	Invloed op de compressiefactor voor verschillende resoluties en een verschillend aantal dieptemappen. Dit zijn de resultaten die werden bekomen voor het Stanford konijn.	89
A.7	Invloed op de compressiefactor voor verschillende resoluties en een verschillend aantal dieptemappen. Dit zijn de resultaten die werden bekomen voor de demon.	89
B.1	Verband tussen resolutie en dieptemappen voor de Stanford draak. $\Delta_{dieptemappen}$ beschrijft de relatieve wijziging van het aantal foutieve pixels voor een toenemend aantal dieptemappen en een vaste resolutie. $\Delta_{resolutie}$ beschrijft de relatieve wijziging voor een toenemende resolutie en een vast aantal dieptemappen.	90

B.2 Verband tussen resolutie en dieptemappen voor de boom. $\Delta_{dieptemappen}$ beschrijft de relatieve wijziging van het aantal foutieve pixels voor een toenemend aantal dieptemappen en een vaste resolutie. $\Delta_{resolutie}$ beschrijft de relatieve wijziging voor een toenemende resolutie en een vast aantal dieptemappen.	91
--	----

Compression Techniques for Coherent Shadow Maps

Xavier Talpe*

Katholieke Universiteit Leuven

Abstract

Shadows are essential in the visualization of realistic looking scenes. A commonly used technique to visualize shadows for interactive applications are shadow maps. For rendering realistic looking shadows, multiple shadow maps have to be generated and stored in memory. Unfortunately these shadow maps require a prohibitive amount of storage. By exploiting the coherence between these shadow maps, a compact data structure can be obtained. This is the idea behind coherent shadow maps. This paper investigates the compression technique used in the construction of coherent shadow maps. Other compression techniques that further exploit the coherence between shadow maps are discussed. Our results show that per-pixel compression is limited to objects with a relative smooth surface, but overall achieves a high compression rate that is hard to achieve with other techniques.

Keywords: shadow mapping, coherent, precomputed visibility, compression.

1 Introduction

Shadows are an important element in the production of realistic looking images. Shadows provide visual cues to aid the user in understanding the shape and relative position of objects in a scene. Without shadows, scenes often look flat and unrealistic.

A commonly used technique to visualize shadows for interactive applications are shadow maps. Unfortunately, shadow maps are limited to rendering hard, non-realistic shadows. To overcome this problem, one could combine the visibility information of multiple shadow maps. However, in order to produce realistic looking shadows, many shadow maps would have to be constructed.

For interactive applications, reconstructing multiple shadow maps during rendering is a time consuming process. Instead, these shadow maps could be precomputed as to prevent them from being reconstructed every frame. One possible way to do this is by precomputing the visibility information of a specific object in the scene by using shadow maps. To capture the full visibility information, shadow maps are constructed from all possible directions. During rendering, these shadow maps can be queried to decide whether a point lies in shadow or not for a given direction of incoming light.

To achieve realistic looking shadows, many shadow maps have to be constructed. Because these shadow maps require prohibitive amounts of storage, compression is vital. Coherent shadow maps [Ritschel et al. 2007] exploit the coherence between the shadow maps to construct a compact and easily accessible data structure that can be used for interactive applications.

This paper investigates the effectiveness of the compression technique used in the construction of coherent shadow maps. The influence of several parameters on the compression rate is checked. In an attempt to further compress the shadow maps, other compression techniques are discussed. These techniques all rely on exploiting the coherence between shadow maps. Our results show that per-pixel compression is limited to objects with a relative smooth

surface, but overall achieves a high compression rate that is hard to obtain with other techniques.

In section 2 the previous work is discussed. Section 2.1 explains the standard compression technique that is used to construct coherent shadow maps. Section 3 introduces several alternative compression techniques that can be used to further exploit the coherence between shadow maps. The results of these compression techniques are discussed in section 4.

2 Previous Work

Shadows have been a popular research topic in computer graphics. During the last decades several techniques have been developed. The most popular and widely known technique uses a shadow map (also called depth map) [Williams 1978]. To render shadows, the scene is first rendered from the viewpoint of the light source, creating a depth map of the scene. This depth map gives an idea on the surfaces that are lighted. In the next step the scene is rendered from the viewpoint of the camera. Each visible point in the scene is then transformed to the coordinate space of the light. By comparing the depth value of the transformed point to the depth value stored in the depth map, a decision is made on whether this point lies in shadow or not.

Shadow maps have been extensively used in interactive applications because of their speed and ease of use. Shadow maps however, suffer from severe aliasing and self-shadowing. [Reeves et al. 1987] introduced a method called percentage closer filtering which uses multi-sampling to produce anti-aliased shadows. To prevent self-shadowing one typically introduces a small bias that is added to the depth values in the depth map. [Weiskopf and Ertl 2003] introduced the idea of dual depth maps. Dual depth maps capture the depth values of the second closest points in a scene. Using these, an adaptive bias can be chosen that prevents self-shadowing from occurring.

Space-filling curves define an ordering for all points in n -dimensional hypercube. An example of such a curve is the Hilbert curve [Hilbert 1891]. This curve defines an ordering for 2^{nk} points with $n \in \mathbb{R}_0^+$ the number of dimensions and $k \in \mathbb{R}_0^+$ the order of the curve. An illustration of this curve is shown in figure 1.

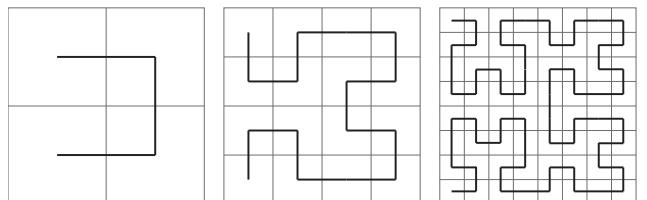


Figure 1: Two dimensional Hilbert curve of first, second and third order. These curves define an ordering for respectively 2^2 , 2^4 and 2^6 points.

The Hilbert curve has some unique properties. The most important one is that it preserves data locality very well. In fact, it has been found to be the most coherent space-filling curve [Voorhies 1991] [Moon et al. 2001]. Because of these properties, the Hilbert curve has been used in variety of domains such as: cryptology and database systems [Hamilton 2006].

*e-mail: xavier.talpe@student.kuleuven.be

2.1 Coherent Shadow Maps

Coherent shadow maps describe the full visibility information of a specific object in a scene [Ritschel et al. 2007]. To capture this visibility information, orthographic shadow maps are constructed on the bounding sphere of the object, capturing the visibility information from all directions. This is illustrated in figure 2. Once these shadow maps are constructed, they have to be compressed. To ensure a high compression rate, the shadow maps are first ordered in a coherent sequence. Coherent means that the depth values of consecutive shadow maps are as similar as possible. A space-filling curve such as the Hilbert curve or a zigzag curve can be used to define this ordering. The resulting sequence can then be compressed using per-pixel compression.

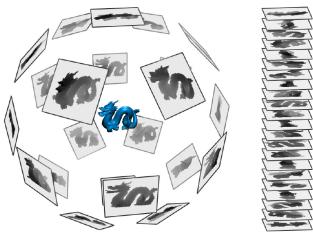


Figure 2: Coherent shadow maps. Shadow maps are constructed from all directions around an object (left). In the next step they are stored in a coherent sequence (right) [Ritschel et al. 2007].

2.2 Per-Pixel Compression

Per-pixel compression is based on a property of dual depth maps. Dual depth maps describe the depth values of the second closest objects in a scene. These dual depth values lead to an interesting property for shadow maps. For the visibility test to succeed, any depth value between the first (z_1) and second (z_2) closest point can be used. As long as a depth value is chosen between these bounds, the visibility test will remain correct for all points in the scene.

This means that for two or more pixels with similar z_1 - and z_2 -values, a common depth value z^* can be chosen. As a result, only one depth value has to be stored instead of two or more. Most importantly, a compression technique based on this property does not introduce any errors at all as long as $z^* \in [z_{1,i}, z_{2,i}]$ for all pixels i . This is illustrated in figure 3.

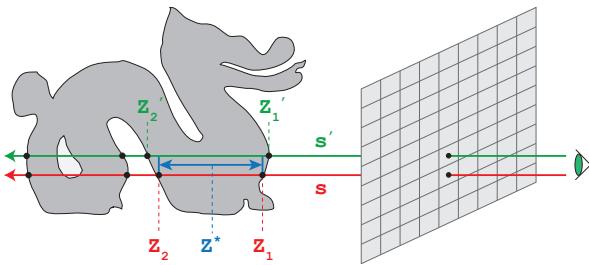


Figure 3: Dual depth values. A common depth value z^* can be used. The result of the visibility test is not influenced as long as $z^* \in [z_1, z_2]$, this is illustrated by the blue line segment.

Per-pixel compression uses this property to compress multiple shadow maps into a compact and easily accessible data structure. Therefore depth values are compressed in separate groups based on their pixel position. Each group thus contains as many depth values as there are shadow maps. The order in which the individual

depth values are compressed is the same as the order defined for each depth map in the coherent sequence. Each group is then compressed in the following way. Starting from the first till the last depth value, a common depth value is searched that can be used for as many pixels as possible. To do this, consecutive groups of pixels are found for which the maximum of their z_1 -values is smaller than the minimum of their z_2 -values. The average between these bounds is then chosen as the common depth value g_i . Figure 4 illustrates this for a specific pixel.

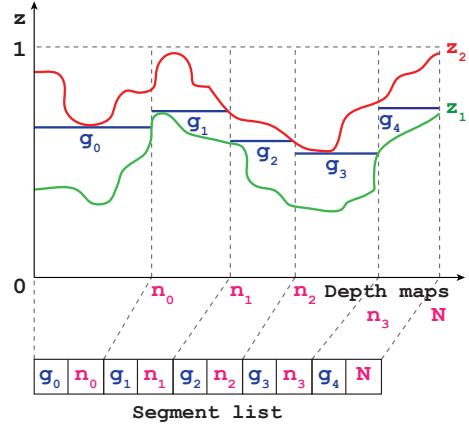


Figure 4: Per-pixel compression for N depth maps. All depth values at a specific pixel are compressed by constructing piecewise constant line segments g_i . These common depth values can then be stored in segments (g_i, n_i) .

The common depth values g_i are stored in segments together with an index n_i . This index describes the number of depth maps for which g_i can be used. All segments are stored in a segment list. This compression step is repeated for every pixel position, resulting in $M \times M$ segment lists, with M the horizontal and vertical dimension of the depth maps.

The order in which the depth maps are compressed are of great importance for the efficiency of per-pixel compression. If consecutive shadow maps show a big variation in their depth values, a common depth value for multiple pixels will be hard to find. Instead, if the depth values are more or less equal, there is a large chance a common depth value can be found for several pixels.

3 Alternative Compression Techniques

Per-pixel compression is limited to exploiting the coherence between depth maps in one dimension, i.e. over multiple depth maps. There is however a lot more coherence to be exploited. Take the example of a color image. Neighbouring pixels often have a similar color. The same is true for the depth maps. Pixels that are close to each other, generally have similar depth values. This implies that a common depth value can be found and presumably it can be used for multiple pixels.

In this section some alternative compression techniques are introduced. These techniques try to further exploit the coherence between depth maps to achieve a high compression rate. The results of each technique are discussed at the end of this paper.

3.1 Per-Map Compression

With per-map compression each depth map is compressed separately. The compression itself is based on the same property as

per-pixel compression. For each pixel and its z_1 - and z_2 -values, a common depth value is found such that multiple pixels can be represented by this depth value. As with per-pixel compression, these depth values are stored in segments (g_i, n_i) . Compressing one shadow map thus results in a list of segments.

The index n_i is based on the order in which pixels of a depth map are compressed. As with per-pixel compression, this order has a high influence on the efficiency of per-map compression. Therefore a space-filling curve such as the Hilbert curve can be used to define a coherent order of the pixels. Because of the unique properties of this curve, a high compression rate can be achieved.

3.2 3D Hilbert Compression

Per-map compression is limited to exploiting coherence between pixels in one depth map. Per-pixel compression on the other hand shows that there is also coherence between depth maps. Thus, in order to further exploit the coherence, these two techniques could be combined. One way to do this is with a three dimensional Hilbert curve.

A three dimensional Hilbert curve defines a coherent ordering for points in a cube. Due to this limitation, the depth maps have to be organised in cubes. The size of a cube will generally be determined by the resolution of the depth maps. If one has a resolution of 128×128 pixels, one cube will consist of 128 depth maps. In practice there will be many more depth maps than 128 so the entire sequence of depth maps is split into cubes of equal size. This also implies that the coherent sequence of depth maps is still important and should be as coherent as possible.

Similar to the previously mentioned compression techniques, the common depth values are stored in segments. This time the index n_i is defined by the order in which pixels are compressed.

3.3 4D Hilbert Compression

Since the Hilbert curve is multidimensional, one could also try compressing the depth maps based on a four dimensional Hilbert curve. Of course a four dimensional representation is needed in order to use a compression like this.

The depth maps are known to be two dimensional. The only requirement left is a way to identify what depth value belongs to what depth map. Since all depth maps are constructed on an equal distance around an object, each depth map can be uniquely identified by its position. This three dimensional position can be transformed into two dimensional spherical coordinates (θ, ϕ) . This way a four dimensional space of depth values is constructed. With a Hilbert curve, this space can be compressed in a coherent way.

4 Results

Figure 5 shows the results of per-pixel compression for the Stanford dragon, the Stanford bunny, a simple sphere and a tree consisting of many leafs and branches. From these results we see a sphere leads to the highest compression rate. This is no surprise. When constructing depth maps of a sphere from a fixed distance, all these depth maps will look identical. In theory this means that only one coherent shadow map is constructed, in practice multiple maps are required because of discretization at the edges. The Stanford dragon and bunny give similar results. Because of their relative smooth surfaces, high compression rates can be achieved. The tree on the other hand is a typical example of an object where per-pixel compression fails horribly. This is due to the many fine leafs and branches, and the resulting highly variable geometry of the object.

If the distance between the first and second closest point is very small, there is very little room to pick a common depth value. This results in a lower amount of pixels that can be compressed into one common depth value.

Figure 6 and table 1 show the detailed results of per-pixel compression for the Stanford bunny. The first observation is that resolution has a huge impact on the size of the coherent shadow maps. However, as the resolution increases, so does the compression rate. This is due to more detailed information on the visibility of the object. Variations between depth values are smaller, leading to a higher compression rate. Another important factor are the number of depth maps. As this number increases, so does the compression factor. This is because an increase in depth maps also results in a smaller distance between depth maps. This leads to a depth values that are less diversified, which in turn increases the coherence between depth maps.

Figure 7 shows the results of per-map compression compared to the results of per-pixel compressions. Apparently the resolution is the main factor in the effectiveness of per-map compression. Only for higher resolutions is a compression rate obtained that is higher than with per-pixel compression. As the number of depth maps goes up, a higher compression rate is achieved with per-pixel compression instead. This is due to the linear behaviour of the curves for per-map compression, this indicates that it has a constant compression rate. As a result, no matter what resolution is used, per-pixel compression will always be the better choice if the number of depth maps is high.

Figure 8 shows the results of compression based on the three dimensional Hilbert curve for the Stanford bunny. These results are quite similar to the ones for per-map compression. Only for higher resolutions will a better result be obtained than for per-pixel compression. Once the number of depth maps starts to climb, per-pixel compression will give the best result.

For the compression based on a four dimensional Hilbert curve, coherent shadow maps were constructed for the Stanford dragon. For 16.384 depth maps with a resolution of 128×128 pixels, the resulting coherent shadow maps were 164 MB in size. The original coherent shadow maps only took 62 MB. Similar test cases gave the same results.

5 Conclusion

In this paper the results of the standard per-pixel compression were analysed. From these results we can conclude that with per-pixel compression high compression rates can be obtained, as long as the object has a relative smooth surface. Tree-like objects with a high variation in geometry or fine structures such as leafs or branches, will typically result in a very low compression factor.

This paper also introduced several alternative compression techniques for the construction of coherent shadow maps. By exploiting the coherence in different dimensions than per-pixel compression, higher compression rates can be obtained for very specific settings. For everyday use however, per-pixel compression is still the best choice, as it scales a lot better with an increase in resolution or an increase in the number of depth maps.

References

- HAMILTON, C. 2006. Compact hilbert indices. Tech. rep., Dalhousie University, July.
- HILBERT, D. 1891. ber die stetige abbildung einer linie auf ein flchenstck. *Mathematische Annalen* 38, 459.

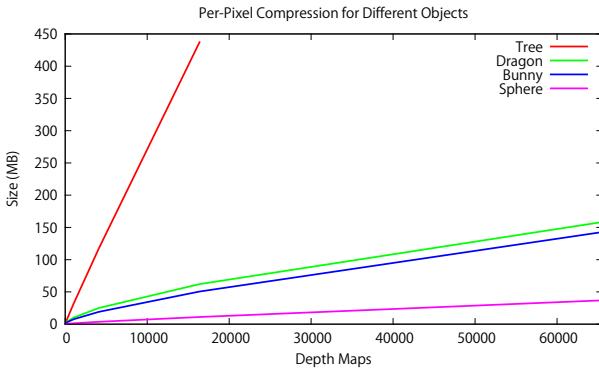


Figure 5: Per-pixel compression for different objects. The constructed depth maps had a resolution of 128×128 pixels. There is a clear difference between the compression rate of different objects. The less smooth the surface of the object, the lower its compression rate.

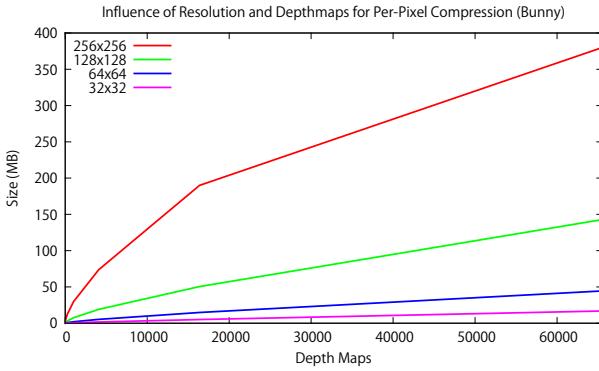


Figure 6: Per-pixel compression for the Stanford Bunny. Even though a higher resolution results in a higher compression rate, resolution has a huge impact on the size of the coherent shadow maps. The compression rate also increases when a higher number of depth maps is used.

MOON, B., JAGADISH, H., FALOUTSOS, C., AND SALTZ., J. H. 2001. Analysis of the clustering properties of the hilbert space-filling curve. *IEEE Transactions on Knowledge and Data Engineering* (January), 124–141.

REEVES, W. T., SALESIN, D. H., AND COOK, R. L. 1987. Antialiased shadows with depth maps. In *Proceedings of ACM SIGGRAPH*, 283–291.

RITSCHEL, T., GROSCH, T., KAUTZ, J., AND MLLER, S. 2007. Interactive illumination with coherent shadow maps. In *Proceedings of Eurographics Symposium on Rendering*, 61–72.

VOORHIES, D. 1991. Space-filling curves and a measure of coherence. In *Graphics Gems II*, J. Arvo, Ed. Academic Press, 26–30.

WEISKOPF, D., AND ERTL, T. 2003. Shadow mapping based on dual depth layers. In *Eurographics 2003 Short Papers*, 53–60.

WILLIAMS, L. 1978. Casting curved shadows on curved surfaces. In *Proceedings of ACM SIGGRAPH*, 270–274.

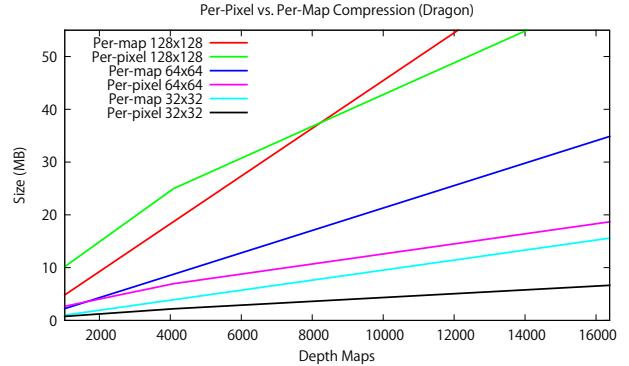


Figure 7: Per-map compression for the Stanford dragon. The linear curves for per-map compression indicate that it has a constant compression rate. As the resolution increases, per-map compression becomes a possible alternative for per-pixel compression. As the number of depth maps start to climb however, a higher compression rate is achieved with per-pixel compression instead.

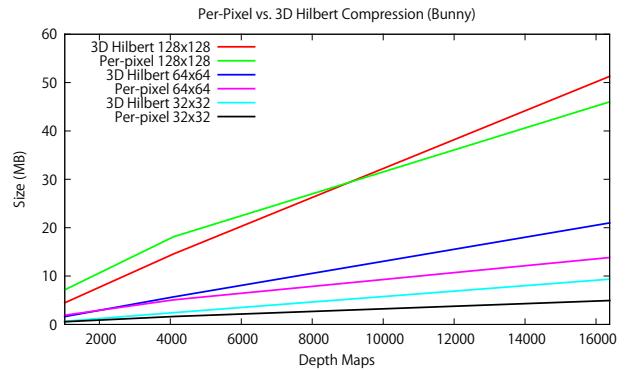


Figure 8: Compression based on a three dimensional Hilbert curve. The compression rate increases as the resolution gets higher. As the number of depth maps start to climb however, a higher compression rate is achieved with per-pixel compression instead.

Depth Maps	Resolution	Uncompressed	Per-pixel	Ratio
1.024	32×32	8 MB	0,57 MB	14:1
1.024	64×64	32 MB	1,99 MB	16:1
1.024	128×128	128 MB	7,57 MB	17:1
1.024	256×256	512 MB	29,8 MB	17:1
4.096	32×32	32 MB	1,65 MB	19:1
4.096	64×64	128 MB	5,26 MB	24:1
4.096	128×128	512 MB	19,09 MB	27:1
4.096	256×256	2.048 MB	73,85 MB	28:1
16.384	32×32	128 MB	5,10 MB	25:1
16.384	64×64	512 MB	14,75 MB	35:1
16.384	128×128	2.048 MB	50,54 MB	41:1
16.384	256×256	8.192 MB	190,02 MB	43:1

Table 1: Subset of the results of per-pixel compression for the Stanford bunny. As the resolution and/or number of depth maps increases, a higher compression rate is obtained.



Coherente Schaduwassen

Situering

Schaduwen vormen een essentieel onderdeel bij het visualiseren van virtuele scènes.

Met behulp van coherente schaduwassen kunnen realistische schaduwen worden gevisualiseerd voor interactieve toepassingen.

Door de visibiliteitsinformatie van een object op voorhand te berekenen en deze informatie te comprimeren, wordt een compacte en snel toegankelijke datastructuur bekomen. Deze structuur kan tijdens het visualiseren van een scène worden geraadpleegd om te bepalen of een punt in schaduw ligt of niet.

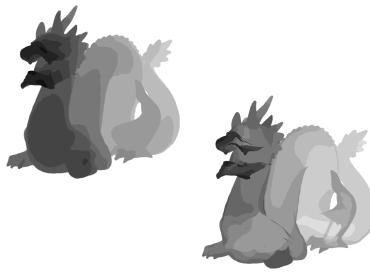
Opslaan Visibiliteitsinformatie

Met behulp van schaduwassen wordt de visibiliteitsinformatie van een object opgeslagen.



Compressie

Door een coherente sequentie van de schaduwassen te bepalen, kunnen de schaduwassen op een compacte manier worden voorgesteld. Daarvoor worden duale dieptemappen gebruikt.



Kwaliteit

De kwaliteit van de schaduwen wordt o.a. bepaald door het aantal schaduwassen die worden geconstrueerd. Van boven naar onder: 1024, 16384 en 65536 schaduwassen.

Resultaten

