

Multispectrale rendering

Peter Vangorp Koen Yskout

17 mei 2005



Dankwoord

In de eerste plaats willen we onze promotor, Prof. Dr. ir. Philip Dutré, bedanken om onze interesse in de wereld van computer graphics op te wekken. We zijn hem zeer dankbaar voor de kans die we krijgen om een eindwerk te maken in dit vakgebied en voor zijn opbouwende kritiek hierbij.

Ook bedanken we onze begeleider Karl vom Berge voor zijn begeleiding en opvolging van dit eindwerk en voor zijn suggesties bij het nalezen van de tekst.

De auteurs betuigen ook dank aan hun ouders, broers en zus, vrienden en familie, voor de steun en hulpvaardigheid die ze geboden hebben in de stressvolle periodes.

We zijn verder Jan-Willem Bijl zeer dankbaar, voor zijn mooie foto van een bijna-ideale zeepbel en de toestemming om deze te mogen gebruiken.

Onze dank gaat tevens uit naar Pieter Peers, voor zijn deskundige uitleg over de apparatuur om HDR-foto's te nemen in het graphics lab.

Koen zou in het bijzonder zijn vriendin Sophie willen bedanken, voor de steun, interesse, hulp en suggesties die ze geboden heeft bij het maken van dit eindwerk. Talloze tik- en taalfouten die nu ontbreken in de tekst, werden door haar aangeduid.



Inhoudsopgave

Lijst van symbolen	iv
Lijst van figuren	vi
Listings	ix
Lijst van afkortingen	x
1 Inleiding	1
1.1 Situering	1
1.2 Historisch overzicht	2
1.3 Bestaande oplossingen en systemen	3
1.4 Doelstellingen	3
1.5 Overzicht	4
2 Kleur	6
2.1 Licht	6
2.2 Spectra	8
2.3 Kleurwaarneming	11
2.4 Colorimetrie	13
2.5 Kleurenruimtes	16
2.5.1 RGB	16
2.5.2 XYZ	17
2.6 RGB en spectra	20
2.6.1 Beperkingen van RGB	20
2.6.2 Omzetting van RGB naar spectrum	24
3 Spectraal raamwerk	28
3.1 Spectrale voorstellingen	28
3.1.1 Overzicht	28
3.1.2 Bemonsterd spectrum	29

3.1.3	Getabuleerd spectrum	30
3.1.4	Fourierspectrum	30
3.1.5	Composietspectrum	32
3.2	Implementatie	33
3.2.1	Spectrum	33
3.2.2	Conversie	33
3.2.3	Invoerformaat	35
4	Optica	37
4.1	Breking	38
4.2	Interferentie	41
5	Spectraal lichttransport	44
5.1	Radiometrische grootheden	45
5.2	Klassieke ray tracing	47
5.3	Intersectie	47
5.4	Versnellingsstructuren	48
5.5	Zichtbaarheid	48
5.6	Kleuring	49
5.7	Textuur	52
5.8	Recursie	54
5.9	Globale belichting	55
5.10	Stochastische ray tracing	57
5.11	Foton mapping	60
5.12	Participerende media	61
6	Interferentie in een dunne film	65
6.1	Voorgaand werk	65
6.2	Theoretisch model	66
6.3	Implementatie	70
6.4	Resultaten	75
7	DiffRACTIE aan gegroefde materialen	77
7.1	Voorgaand werk	77
7.2	Beschrijving van de microstructuur	78
7.3	Wiskundig model	80
7.4	Implementatie	84
7.5	Resultaten	84

8	Dispersie	90
8.1	Voorgaand werk	90
8.2	Directe en indirecte dispersie	91
8.3	Technieken	92
8.3.1	Directe dispersie	94
8.3.2	Indirecte dispersie	94
8.3.3	Combinatie	95
8.4	Implementatie	95
8.5	Resultaten	95
9	Absorptie in volumes	99
9.1	Voorgaand werk	99
9.2	Theorie	100
9.3	Implementatie	102
9.4	Resultaten	103
10	Opgemeten BRDF's	106
10.1	Voorgaand werk	106
10.2	Meetgegevens	107
10.3	Implementatie	108
10.4	Resultaten	108
11	Implementatie	112
11.1	Overzicht	112
11.2	Opmerkingen	113
11.3	Performantie	114
11.4	Beslissingen	115
12	Besluit	116
12.1	Overzicht	116
12.2	Spectraal raamwerk	116
12.3	Spectrale effecten	117
12.4	Slot	119
	Bibliografie	120
A	Aanpassingen in het .pbrt invoerformaat	126
A.1	Lijsten van parameters	126
A.2	Uitdrukkingen	127
A.3	Plug-Ins	128



Lijst van symbolen

\vec{v}	vector
\hat{v}	eenheidsvector of genormaliseerde richting
\hat{n}	normaalvector
θ	hoek met de normaalvector
i	imaginaire eenheid
λ	golflengte
k	golftal
\vec{k}	golfvector
c	lichtsnelheid in vacuüm of lucht
ϕ	fase
\hat{r}	propagatierichting
\vec{E}	elektrische golf
\vec{B}	magnetische golf
x	afgelegde weg
ω	cirkelfrequentie of pulsatie
t	tijd
$n(\lambda)$	brekingsindex
θ_i	inkomende hoek
θ_r	gebroken hoek
$\hat{\omega}_i$ of Ψ	inkomende richting
$\hat{\omega}_o$ of Θ	uitgaande richting
$\hat{\omega}_h$	<i>halve-hoek-vector</i> , de bissectrice van de inkomende en uitgaande richting

$S(\lambda)$	spectrum of SPD
$\delta(\lambda)$	Dirac-impuls
$Q(\lambda)$	energie
$\Phi(\lambda)$	vermogen
P	totaal vermogen van het spectrum
$E(\lambda)$	irradiantie
$M(\lambda)$	exitantie
$B(\lambda)$	radiositeit
$I(\lambda)$	intensiteit
$L(\lambda)$	radiantie
$L_e(\lambda)$	radiantie uitgestraald door een lichtbron
f_r	bidirectionele reflectantie distributie functie
ρ	reflectantie
$\cdot (x \rightarrow \hat{\omega})$	een grootheid reist vanuit het punt x naar de richting $\hat{\omega}$
$\cdot (x \leftarrow \hat{\omega})$	een grootheid reist vanuit de richting $\hat{\omega}$ naar het punt x
$\cdot (x, \hat{\omega}_i \rightarrow \hat{\omega}_o)$	een grootheid verandert in het punt x van richting $\hat{\omega}_i$ naar richting $\hat{\omega}_o$
$\cdot (x, \hat{\omega}_i \leftrightarrow \hat{\omega}_o)$	de richtingsverandering is wederkerig
$S_1(\lambda) \cong S_2(\lambda)$	$S_1(\lambda)$ en $S_2(\lambda)$ zijn metamere spectra



Lijst van figuren

2.1	Voorbeelden van SPD's van een gloeilamp, een laserlichtje en een TL-lamp.	9
2.2	Het reflectantiespectrum van groen karton en het transmissiespectrum van een oranje fotografische filter.	10
2.3	Gevoeligheid van de drie soorten kegeltjes voor elke golflengte.	12
2.4	Schematische voorstelling van het color matching experiment.	13
2.5	Color matching functies voor een rood, groen en blauw testlicht.	14
2.6	Gemiddelde (gescaleerde) SPD's voor de rode, groene en blauwe fosfors van een CRT-monitor.	17
2.7	Color matching functies $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ en $\bar{z}(\lambda)$ voor XYZ.	18
2.8	Chromaticiteitsdiagram en gamut van een RGB-monitor.	19
2.9	Vergelijking tussen RGB en spectraal	22
2.10	Vergelijking tussen een echte scène en een overeenkomstige virtuele scène, gerenderd met een RGB en een spectraal kleurmodel.	23
2.11	Basisspectra voor de methode van Smits.	25
2.12	De som van het rode, groene en blauwe basisspectrum verschilt van het witte basisspectrum.	26
3.1	Fourierbenadering van een rechthoekig spectrum.	31
4.1	Breking van een straal aan de rand van een medium.	39
4.2	Variatie van de brekingsindex van glas en diamant in functie van de golflengte.	41
4.3	Fasordiagram van een golf.	42
4.4	Resulterende golf van interferentie, grafisch afgeleid.	43
5.1	Belang van de invallende hoek bij irradiantie.	45
5.2	Radiantie.	46
5.3	Basisalgoritme van ray tracing.	47
5.4	De bidirectionele reflectantie distributie functie.	49

5.5	Een lichtbundel weerkaatst aan een oppervlak met een micro-facetstructuur.	51
5.6	De BRDF van een mat Lambertiaans materiaal en een glanzend Phong-materiaal.	51
5.7	Een Lambertiaans diffuse bol en het typisch plastic-achtige uitzicht van Phong-kleuring.	51
5.8	Subsurface scattering in de huid van het gelaat.	52
5.9	De textuurprojectie T	53
5.10	De Utah theepot, één van de populairste voorwerpen om te renderen, met een textuur van marmer en met een bump map.	54
5.11	De <i>Cornell box</i> testscène met zachte schaduwen en <i>color bleeding</i> , en een scène die grotendeels met indirecte belichting is verlicht.	56
5.12	Iteratieve verbetering in fotorealistische rendering.	57
5.13	Een brandvlek van licht dat door een glas cognac schijnt.	61
5.14	Een volumetrische brandvlek in een participerend medium, en vuur als participerend medium.	62
5.15	Schematische voorstelling van de interactie met een participerend medium op verschillende plaatsen z tussen y en x	63
5.16	Schematische voorstelling van <i>ray marching</i>	64
6.1	Ringenvan Newton rond het raakpunt van een bolle lens aan een ander oppervlak.	66
6.2	Fysisch model voor interferentie in een dunne film.	67
6.3	Verschil in afgelegde weg.	68
6.4	Fasorvoorstelling van de superpositie van twee golven.	69
6.5	Een foto van een zeepbel en een gerenderde gelijkaardige zeepbel.	75
6.6	Ringenvan Newton in een zeepbel met een groter dikte-verloop (van 200 nm tot 1500 nm).	76
7.1	Macroscopische dimensies van een compact disc.	79
7.2	Microscopische dimensies van een compact disc.	79
7.3	Foto van een compact disc onder een puntlichtbron.	80
7.4	Model voor diffractie aan een compact disc.	80
7.5	Een grammfoonplaat met in het midden een compact disc.	82
7.6	Oppervlak met cilindervormige groeven.	83
7.7	Foto's van een cd en gerenderde cd's met de opgemeten posities van camera en lichtbron.	86
7.8	Foto van een cd en gerenderde cd met de opgemeten positie van camera en lichtbron, bijna onder de perfect gereflecteerde richting.	87

7.9	Foto's van een cd en een dvd, en de gerenderde cd en dvd met de opgemeten posities van camera en lichtbron.	88
8.1	Dispersie bij het kijken doorheen een object (directe dispersie).	92
8.2	Dispersie en weerkaatsing op een oppervlak (indirecte dispersie), met één lichtstraal en met twee lichtstralen.	93
8.3	Glazen bol zonder en met dispersie.	96
8.4	Dispersieve caustics van een smalle en een bredere lichtbundel.	97
8.5	Roterende diamant zonder en met dispersie.	98
9.1	Verschillende situaties bij het bepalen van de lengte van de overlapping van een deel van een straal met een volume.	103
9.2	Absorptie in balken met verschillende dikte, met het RGB-kleurmodel en met een spectraal model.	104
9.3	Absorptie in een glazen piramide, waardoor de top veel lichter gekleurd is dan de basis.	105
10.1	Posities van de inkomende en uitgaande richtingen op de hemisfeer.	108
10.2	Drie gegroefde bollen van blauwe acryl (plastic), Mystic autolak en Cayman autolak.	110
10.3	De compact disc gerenderd met de analytische BRDF en met de virtueel opgemeten BRDF.	111



Listings

3.1	BaseSpectrum interface.	34
6.1	ThinFilm BRDF.	72
6.2	ThinFilm BTDF.	73
6.3	ThinFilm functie X_k (formule 6.12).	74
7.1	Diffraction BRDF.	85
10.1	SampledBDF.	109



Lijst van afkortingen

BRDF	bidirectionele reflectantie distributie functie
BSDF	bidirectionele <i>scattering</i> distributie functie
BSSRDF	bidirectionele <i>scattering surface</i> reflectantie distributie functie
BTDF	bidirectionele transmittantie distributie functie
CIE	Commission Internationale de l'Eclairage
PBR	Physically Based Rendering: From Theory to Implementation
PBRT	Physically Based Ray Tracer
SPD	Spectral Power Distribution

Inleiding

De mens heeft altijd al afbeeldingen gemaakt van zijn omgeving. Soms is het mogelijk om een object in drie dimensies af te beelden (beeldhouwwerken), maar meestal is men beperkt tot slechts twee dimensies (tekeningen, schilderijen, foto's).

Sinds de jaren 1960 worden computers gebruikt om deze afbeelding van de driedimensionale wereld naar een tweedimensionale figuur uit te voeren. Deze transformatie, *rendering* genaamd, is door de jaren geëvolueerd naar een complexe verzameling van algoritmes.

Traditioneel wordt op computers met de bekende RGB kleuren gewerkt. Er zal aangetoond worden dat dit een onnauwkeurige voorstelling is. *Multispectrale* rendering werkt op een fysisch correcte manier met kleuren. Het volledige spectrum van licht wordt gebruikt, net zoals in de realiteit.

1.1 Situering

Sinds de uitvinding van de computer werd deze al voor heel veel verschillende zaken gebruikt. Zo voeren computers wetenschappelijke berekeningen uit en worden ze ingezet voor de automatisering van productieprocessen. Kantoor-toepassingen zoals tekstverwerkers, rekenbladen en databanken zijn noodzakelijk geworden voor de goede werking van vele bedrijven. Ter ontspanning bestaat er ook een hele reeks al dan niet vredelievende en realistische spellen op de computer. Dit eindwerk bevindt zich daarentegen in nog een ander domein: simulatie, en meer bepaald de simulatie van lichttransport met als uiteindelijk doel een afbeelding van een scène te kunnen genereren die niet te onderscheiden is van een foto van diezelfde scène in realiteit. Het onderzoek hiernaar gebeurt binnen computerwetenschappen in de tak *computer graphics*.

Eén van de belangrijke aspecten voor deze simulatie is het gedrag van

het licht en de principes van kleur. Gewoonlijk worden er vereenvoudigingen gemaakt van de fysische beschrijving van licht en kleur om een eenvoudigere manier van werken mogelijk te maken. Hierdoor worden echter fouten geïntroduceerd en sommige fenomenen kunnen niet meer accuraat beschreven worden. In dit eindwerk zal teruggegrepen worden naar de fysische achtergrond om nauwkeurige berekeningen met kleur uit te kunnen voeren. Hiermee kunnen dan enkele effecten gesimuleerd worden, die met de traditionele methodes niet mogelijk waren.

Het toepassingsgebied van dit eindwerk is zowat het gehele domein van fotorealistische beeldgeneratie waarin kleur belangrijk is. Ook in omgevingen waar de behandelde effecten voorkomen, kan het nuttig zijn om realistische simulaties uit te voeren. Een voorbeeld hiervan is het maken van lenzen of het slijpen van diamanten, waarbij effecten zoals dispersie een belangrijke rol spelen. Een computersimulatie zou hier het gedrag van de lens of de diamant kunnen voorspellen in verschillende configuraties.

1.2 Historisch overzicht

Vóór het ontstaan van raytracing-algoritmes werden voornamelijk scanline-algoritmes gebruikt. Deze algoritmes zijn in staat om snel een verzameling veelhoeken (polygonen) op het scherm weer te geven, maar hebben quasi geen voorzieningen voor andere geometrische vormen, schaduwen, reflectie en refractie. Deze fenomenen zijn echter cruciaal om fotorealistische beelden te kunnen genereren.

In 1979 stelde Whitted de eerste recursieve raytracer voor. Deze raytracer was in staat om perfecte spiegelingen, brekingen en schaduwen te genereren. Dit leidde, ten koste van de rekentijd, tot meer realistische resultaten dan een scanline-algoritme. De materiaalmodellen die gebruikt werden, waren echter verre van realistisch. Men zocht steeds naar eenvoudige en snel te evalueren uitdrukkingen voor het uitzicht van materialen, zodanig dat het resultaat er “goed” uitzag (maar daarom niet fysisch correct was).

Met de toenemende rekenkracht van de computers wordt het mogelijk de ad hoc-benadering die toegepast werd, te vervangen door fysisch gebaseerde modellen. Zo is het nu mogelijk om een virtuele wereld onderhevig te maken aan bijvoorbeeld zwaartekracht en wind. Het gedrag van de bewegende objecten en de interacties ertussen kan automatisch berekend worden, waar dit vroeger handmatig moest gebeuren door een animator (en dus ook niet steeds de fysisch correct was). Ook de manier waarop bijvoorbeeld vloeistoffen zich gedragen kan steeds efficiënter gesimuleerd worden.

Naast de geometrie en modellering worden ook de materiaalmodellen alsmear meer gebaseerd op fysische principes. Men bestudeert bijvoorbeeld de microstructuur van het materiaal en bepaalt daaruit een model dat voldoet aan de relevante fysische wetten. Ook bij het bepalen van de kleur van elke pixel in een afbeelding wordt tegenwoordig meer en meer gebruik gemaakt van de fysische principes van lichttransport en radiometrie. Men berekent ook niet meer enkel het licht dat rechtstreeks op een object invalt: ook het licht dat via weerkaatsing op één of meerdere oppervlakken een punt bereikt, wordt nu gesimuleerd (globale belichting).

1.3 Bestaande oplossingen en systemen

Voor de meeste commerciële 3D-pakketten bestaan er plug-ins die het berekenen van sommige spectrale effecten toelaten. Het is echter niet altijd duidelijk of deze plug-ins gebaseerd zijn op de werkelijke fysica of gebruik maken van een hele reeks benaderingen.

Er zijn ook een aantal (open source en commerciële) volledig fysisch gebaseerde raytracers. Enkele voorbeelden hiervan zijn:

- Physically Based Ray Tracer (PBRT) (<http://www.pbrt.org>)
- Realistic Image Synthesis Engine (R.I.S.E.) (<http://rise.sourceforge.net>)
- RenderPark (<http://www.renderpark.be>)
- Maxwell (<http://www.maxwellrender.com>)

Op dit ogenblik is R.I.S.E. hiervan de enige beschikbare raytracer die spectraal werkt.

1.4 Doelstellingen

In het kader van dit eindwerk zullen we proberen een goede en fysisch gebaseerde definitie van kleur te bekijken en te gebruiken om fotorealistische afbeeldingen te genereren. Hiertoe zullen we een spectraal raamwerk ontwikkelen waarin verschillende voorstellingswijzen voor een kleur ondergebracht worden. Dit raamwerk zal ook gekoppeld worden aan een bestaande, fysisch gebaseerde raytracer.

Hiernaast zullen we ook enkele effecten beschrijven waarbij het gebruik van de fysische eigenschappen van licht en kleur noodzakelijk zijn. Meer bepaald zal besproken worden hoe bijvoorbeeld de kleuren in een zeepbel, een compact disc en een diamant ontstaan en hoe licht geabsorbeerd wordt door gekleurd glas. Naast de fysische achtergrond van deze effecten zullen ze ook geïmplementeerd worden gebruik makende van het spectraal raamwerk en de raytracer.

1.5 Overzicht

In dit hoofdstuk werd het onderwerp van dit eindwerk in een ruimer kader geplaatst. De doelstellingen werden kort aangehaald.

In hoofdstuk 2 wordt het fenomeen kleur uitgewerkt. Er zal meer bepaald een definitie van kleur gegeven worden aan de hand van het spectrum van licht en de menselijke waarneming. De veel gebruikte RGB-kleurenruimte zal hierin geplaatst worden. Het werken in deze kleurenruimte zal vergeleken worden met de fysische werkelijkheid. Een grondige kennis van dit hoofdstuk is fundamenteel voor de verdere uiteenzetting.

Hoofdstuk 3 bespreekt de belangrijkste manieren om een spectrum voor te stellen zodat ermee gewerkt kan worden in een computerprogramma. Er zal ook aangegeven worden hoe dit in het kader van dit eindwerk geïmplementeerd werd in een spectraal raamwerk. Dit raamwerk zal de basis vormen voor de implementatie.

Vervolgens wordt in hoofdstuk 4 een overzicht gegeven van de basiseigenschappen van licht en enkele optische verschijnselen die later zullen toegepast worden, zoals breking en interferentie.

In hoofdstuk 5 wordt raytracing, de belangrijkste techniek om fotorealistische afbeeldingen te genereren, uitvoerig behandeld. Er wordt daarbij expliciet aangegeven waar het werken met het spectrum van licht invloed zal hebben. Ook meer geavanceerde technieken voor globale belichting komen aan bod. Er wordt een kader geschetst voor de spectrale effecten uit de volgende hoofdstukken 6-10.

De verschillende effecten worden grondig bestudeerd en de implementatie wordt kort toegelicht. In hoofdstuk 6 komt de interferentie in een dunne film aan bod, met als toepassing een zeepbel. Hoofdstuk 7 geeft een model voor diffractie aan een gegroefd materiaal, zoals een compact disc. In hoofdstuk 8 worden enkele technieken beschreven om dispersie te visualiseren. Hoofdstuk 9 bespreekt een fysisch gebaseerde aanpak voor absorptie in volumes.

De manier om een nauwkeurig opgemeten complex materiaal te gebruiken bij het genereren van fotorealistische afbeeldingen, wordt besproken in hoofdstuk 10.

Tenslotte zal in hoofdstuk 11 een overzicht gegeven worden van de implementatie en enkele bemerkingen hierbij.

In hoofdstuk 12 geven we een kritisch overzicht van het eigen werk en de bereikte doelstellingen. Er worden ook enkele richtingen voor verder onderzoek gegeven.

Bijlage A geeft een kort overzicht van de aanpassingen in het invoerformaat van PBRT om de beschreven uitbreidingen te gebruiken.

Mensen zijn vertrouwd met kleur. In het dagelijks leven wordt er onnoemelijk veel gebruik gemaakt van kleur. Toch heerst er soms verwarring over wat kleur nu juist wel of niet is. Ook in het domein van computer graphics wordt quasi-voortdurend met kleur gewerkt.

In dit hoofdstuk gaan we wat dieper in op de fysische en digitale achtergrond van kleur, kleurwaarneming door de mens en het gebruik van kleur in computer graphics. Er wordt meer bepaald besproken wat licht juist is, hoe licht ontstaat en hoe licht kan worden voorgesteld als een spectrum. Vervolgens wordt verklaard hoe een object een bepaalde kleur krijgt. Nadien wordt behandeld hoe mensen kleuren waarnemen. Het zal blijken dat licht dat op onze ogen invalt, wordt omgezet in exact drie signalen. Die signalen worden dan door onze hersenen geïnterpreteerd als kleur. Dit principe van drie signalen ligt aan de basis voor het gebruik van RGB om met kleur te werken in computer graphics.

2.1 Licht

Licht is samengesteld uit meerdere elektromagnetische golven, elk met hun eigen golflengte λ . Het menselijk oog is gevoelig voor golven met een golflengte van 400 tot 700 nm. Dit is dus het gebied van het elektromagnetisch spectrum waarin we kleur kunnen onderscheiden. Een golf met een golflengte van ± 400 nm geeft een blauwe kleur, ± 550 nm een groene en ± 700 nm een rode kleur. Golven met λ kleiner dan 400 nm zijn ultraviolet (UV). Een golflengte groter dan 700 nm geeft infrarode (IR) straling, soms ook warmtestraling genoemd.

Een andere manier waarop licht beschreven kan worden, is als deeltjes. Elk deeltje (of foton) heeft een bepaalde energie $E = hc/\lambda$, waarbij c de lichtsnelheid is. Licht met een lange golflengte (rood) bevat dus minder energie dan licht met een korte golflengte (blauw). De beschrijving van licht

als deeltjes is complementair met de beschrijving van licht als golf; dit is de golf-deeltjesdualiteit. De beschouwing van licht als deeltje is echter niet van belang voor het vervolg van de uiteenzetting.

Licht kan ontstaan uit verschillende fysico-chemische processen. Ten eerste komt er licht vrij van een lichaam op een bepaalde temperatuur, ook wel een zwarte straler genoemd. De verdeling van de intensiteit over de verschillende golflengtes wordt beschreven door de wet van Planck:

$$I(\lambda) = \frac{2hc^2}{\lambda^5} \frac{1}{\exp\left(\frac{hc}{\lambda kT}\right) - 1} \quad (2.1)$$

met

- $I(\lambda)$ de intensiteit in functie van de golflengte λ ,
- h de constante van Planck (6.626069×10^{-34} J·s),
- c de lichtsnelheid (299 792 458 m/s),
- k de constante van Boltzmann ($1.3806505 \times 10^{-23}$ J/K),
- T de temperatuur in Kelvin.

Voorbeelden van lichtbronnen die goed beschreven kunnen worden op deze manier, zijn de zon en gloeilampen. Het licht dat door deze bronnen wordt uitgestraald, bevat alle golflengtes.

Een tweede manier waarop licht kan ontstaan, is door de absorptie en emissie van energie door de elektronen van atomen. Een elektron kan een hoeveelheid energie opnemen en daardoor geëxciteerd worden. Wanneer het elektron terug vervalt naar zijn grondtoestand, komt er een foton vrij. Als dat foton een frequentie heeft die in het zichtbare gebied valt, ontstaat er licht. Lichtbronnen die op deze manier werken zijn bijvoorbeeld gasontladingslampen (zoals een neonlamp) en lasers. Licht van deze lichtbronnen bevindt zich steeds sterk geconcentreerd rond bepaalde golflengtes.

De derde manier waarop zichtbaar licht kan ontstaan is door fluorescentie. Hierbij wordt licht met een hogere energie door een fosfor omgezet in licht met een lagere energie en warmte. Zo kan ultraviolet licht omgezet worden in zichtbaar licht. Dit wordt bijvoorbeeld gebruikt in een TL-lamp. Een TL-lamp is een gasontladingslamp bedekt met fosfors. De gasontlading produceert ultraviolet licht, die door de fosfors omgezet wordt in zichtbaar licht. De kleur van het uitgestraalde licht wordt bepaald door de verschillende fosfors die gebruikt worden.

In plaats van door ultraviolet licht, kunnen sommige fosfors door elektronen gestimuleerd worden om zichtbaar licht uit te stralen. Dit principe wordt fosforescentie genoemd, en wordt gebruikt voor bijvoorbeeld televisies en CRT-beeldschermen. Een elektronenstraal wordt door een magnetisch veld afgebogen en komt terecht op een fosfor. Hierbij straalt de fosfor zichtbaar licht uit. Een fosfor blijft meestal nog even nalichten nadat de elektronen ingevallen zijn. De duur van dat nalichten varieert afhankelijk van de fosfor. De meeste fosfors blijven slechts enkele milliseconden nalichten (zoals bij een beeldscherm), maar sommige kunnen nog seconden, minuten of zelfs uren nalichten. Die laatste worden dan gebruikt voor materialen die licht geven in het donker.

2.2 Spectra

Met de term ‘spectrum’ kunnen meerdere dingen aangeduid worden. Het kan gebruikt worden voor het spectrum van de lichtbron, of voor het reflectantiespectrum van een materiaal. Beide soorten spectra zullen nu gedefiniëerd en besproken worden.

Spectrum van lichtbronnen

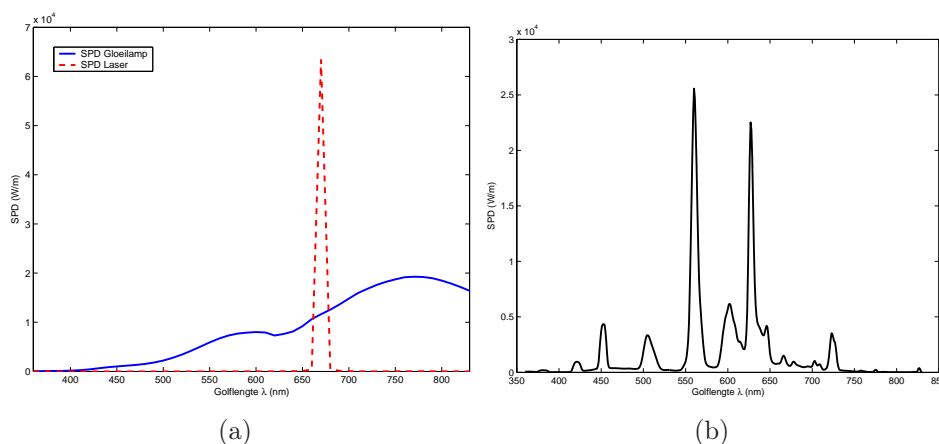
Niet elke golflengte is in een lichtstraal even sterk aanwezig. Het spectrum van licht geeft weer hoe het vermogen van een lichtstraal verdeeld is over de golflengtes. Daarom wordt het spectrum $S(\lambda)$ ook veelal de Spectral Power Distribution (SPD) of spectrale vermogensdistributie genoemd. $S(\lambda)$ wordt uitgedrukt in Watt per meter (W/m). Het is een distributie, wat wil zeggen dat de waarde $S(\lambda)$ het vermogen geeft dat aanwezig is in het licht voor alle golflengtes in een interval $[\lambda, \lambda + d\lambda]$. Het vermogen van één bepaalde golflengte λ is gelijk aan nul¹. Het vermogen van het licht tussen twee golflengtes λ_1 en λ_2 is gelijk aan

$$P_{\lambda_1, \lambda_2} = \int_{\lambda_1}^{\lambda_2} S(\lambda) d\lambda. \quad (2.2)$$

Het totale vermogen van het beschouwde licht is dan de integraal over alle golflengtes van de SPD:

$$P = \int_{-\infty}^{\infty} S(\lambda) d\lambda. \quad (2.3)$$

¹Wanneer dit niet zou zijn, zou men een oneindig vermogen bekommen. Er zijn immers oneindig veel golflengtes.



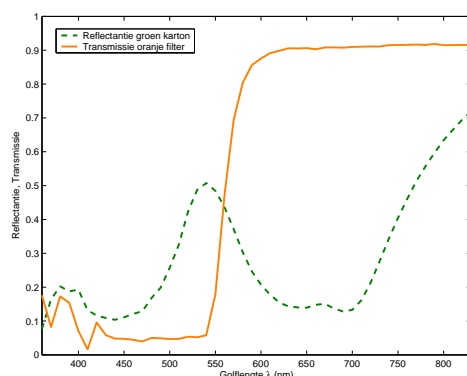
Figuur 2.1: Voorbeelden van SPD's. (a) De SPD van een gloeilamp (volle lijn) en een rood laserlichtje (streeplijn). (b) Opgemeten spectrum van een TL-lamp.

Een SPD kan zacht verlopen of pieken bevatten. Het spectrum dat gevormd wordt door een zwarte straler (de zon of een gloeilamp bijvoorbeeld) is zacht verlopend. Spectra van lichtbronnen waar gebruik gemaakt wordt van gasontlading of fosfors bevat meestal enkele duidelijk afgelijnde pieken. Voorbeelden van onze metingen van de SPD van een gloeilamp, een rood laserlichtje en een TL-lamp staan in figuur 2.1.

Kleur van objecten

We zullen ook nog veelvuldig een ander soort spectrum tegenkomen, namelijk een *reflectantiespectrum*. Dat spectrum geeft niet de vermogensverdeling weer, maar wel de verhouding van de hoeveelheid licht die weerkaatst wordt en de inkomende hoeveelheid licht voor elke golflengte. Dit veronderstelt dat licht gereflecteerd wordt met dezelfde golflengte als waarmee het invalt (en er dus geen verschijnselen zoals fluorescentie optreden). Ook wordt er verondersteld dat het reflectantiespectrum onafhankelijk is van in- en uitgaande richting. Dat laatste is echter niet realistisch. Later zal dit daarom veralgemeend worden naar een bidirectionele reflectantie distributie functie (BRDF), waar de richtingafhankelijkheid wel in rekening gebracht wordt.

Een reflectantiespectrum gaat in principe van 0 (geen weerkaatsing) naar 1 (volledige weerkaatsing) voor een bepaalde golflengte. Analoog aan een reflectantiespectrum kunnen we voor doorschijnende materialen ook een *transmissiespectrum* beschouwen. Dat geeft dan aan hoeveel licht er per golflengte



Figuur 2.2: Het reflectantiespectrum van groen karton (streeplijn) en het transmissiespectrum van een oranje fotografische filter (volle lijn).

doorgelaten wordt. Een voorbeeld van een reflectantie- en transmissiespectrum dat we opgemeten hebben is afgebeeld in figuur 2.2.

Het spectrum van het licht dat weerkaatst wordt door een object, is nu eenvoudig te bepalen. Het is de vermenigvuldiging van de SPD van het invallend licht en het reflectantiespectrum van het materiaal. Dat geeft opnieuw een SPD: de SPD van het weerkaatste licht. In deze SPD is de bijdrage van elke golflengte uit het invallend licht gescaleerd met de waarde van het reflectantiespectrum voor die golflengte. Dat betekent dat een object op zich dus geen kleur heeft: de kleur van een object wordt bepaald door zowel de SPD van het invallend licht als door het reflectantiespectrum van het materiaal van het object. Eenzelfde object kan er totaal verschillend uitzien onder een andere belichting. Wanneer men in het dagelijks leven spreekt over de kleur van een object, bedoelt men meestal de kleur die het object krijgt wanneer het belicht wordt met daglicht.

Implementatie

De verschillende soorten spectra (SPD's, reflectantiespectra en eventueel transmissiespectra) hebben dus elk een andere betekenis. In principe moet dit onderscheid ook in de implementatie behouden blijven. Zo zal later blijken dat enkel SPD's een RGB-waarde hebben. Spreken over de RGB-waarde van een reflectantiespectrum is eigenlijk zinloos. Toch worden deze spectra traditioneel ook met een RGB-waarde voorgesteld, omwille van de eenvoud. Het zou dus handig zijn om de verschillende soorten spectra toch op een uniforme manier te behandelen.

Daarom wordt in onze implementatie elk spectrum voorgesteld alsof het

een SPD is. Voor een reflectantiespectrum betekent dit dat we eigenlijk het spectrum bekijken dat ontstaat uit de reflectie van de CIE Illuminant E door het reflectantiespectrum. De CIE Illuminant E is een niet bestaand referentiespectrum (SPD), met voor alle golflengtes een waarde gelijk aan 1 W/m. Vermenigvuldigen met deze SPD heeft dus geen merkbaar numeriek effect. Conceptueel gaan we echter over van een reflectantiespectrum naar een SPD.

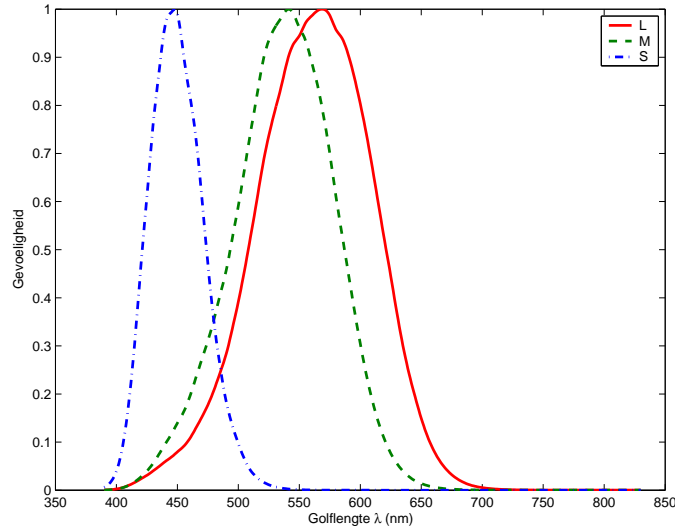
2.3 Kleurwaarneming

Het netvlies in het oog van de mens bestaat voor een groot gedeelte uit fotoreceptoren. Fotoreceptoren reageren op inkomend licht door een signaal naar de hersenen te sturen via de zenuwen. De mens heeft twee verschillende soorten fotoreceptoren, namelijk staafjes en kegeltjes. De staafjes werken vooral goed bij een lage lichtintensiteit. Ze geven ons echter geen kleursensatie. Wanneer er weinig licht aanwezig is, en de staafjes gebruikt worden voor het zicht, wordt dit *scotopische* waarneming genoemd.

Interessanter zijn de kegeltjes. Deze werken alleen bij voldoende lichtintensiteit, maar geven ons dan wel meer informatie, namelijk kleur. Wanneer de kegeltjes actief zijn, spreekt men over *fotopische* waarneming. De overgang tussen scotopische en fotopische waarneming wordt *mesopische* waarneming genoemd. Hier zijn zowel de staafjes als de kegeltjes actief. Aangezien voor kleur enkel de kegeltjes van belang zijn, worden deze verder behandeld.

We hebben drie soorten kegeltjes, namelijk L-, M- en S-kegeltjes. Elk van deze soorten is gevoelig in een specifiek gebied van het zichtbaar spectrum. In figuur 2.3 zijn deze gevoeligheden weergegeven. De S-kegeltjes zijn voornamelijk gevoelig voor de korte golflengtes, in het blauwe gebied dus (400–500 nm). Vandaar ook de letter S (Short). De M- en L-kegeltjes (Medium en Long) overlappen voor een groot deel in de golflengtes waarvoor ze gevoelig zijn. De L-kegeltjes zijn iets gevoeliger naar de rode golflengtes toe, terwijl de gevoeligheid van de M-kegeltjes zich meer in het groene gedeelte van het spectrum bevindt. Merk op dat de gevoeligheden voor elk type kegeltje gescaleerd zijn zodat het maximum zich telkens op 1 bevindt. De grafiek geeft dus niet de relatieve gevoeligheden van de verschillende soorten kegeltjes weer. In werkelijkheid zijn de S-kegeltjes bijvoorbeeld veel minder gevoelig dan de M-kegeltjes.

De kegeltjes vangen licht op, genereren een signaal en sturen dat door naar de hersenen. Vermits de mens drie soorten kegeltjes heeft, krijgen de hersenen drie afzonderlijke signalen, die geïnterpreteerd worden als kleur. Dit



Figuur 2.3: Gevoeligheid van de drie soorten kegeltjes voor elke golflengte.

wordt *trichromaticiteit* genoemd. De meeste primaten zijn net zoals de mens trichromaten. Vogels en vissen hebben echter vaak vier soorten sensors. Van katten en honden wordt dan weer aangenomen dat ze slechts twee soorten sensors hebben, in het blauw-groene gebied. De gevoeligheid van de kegeltjes bij de verschillende soorten dieren is ook niet hetzelfde. Het menselijk model voor kleur is dus niet zonder meer toepasbaar op andere dieren; elk dier vereist een eigen model.

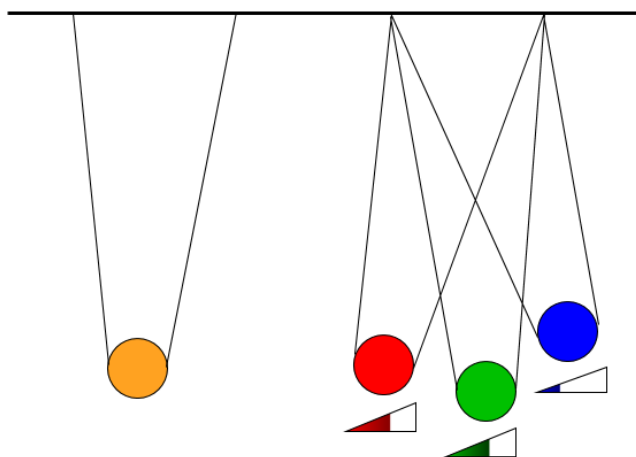
Vermits de mens maar drie soorten receptoren heeft waarmee kleur kan worden waargenomen, kan er geen onderscheid gemaakt worden tussen elk verschillend spectrum. Er zullen dus meerdere spectra zijn, die dezelfde kleursensatie oproepen. Zulke spectra worden *metameren* genoemd. Een zeer belangrijke eigenschap van metameren is dat ze metameer blijven door het toepassen van lineaire operaties. Noteren we het metameer zijn van twee spectra $S_1(\lambda)$ en $S_2(\lambda)$ met \cong , dan geldt dus meer specifiek:

$$\begin{aligned} S_1(\lambda) &\cong S_2(\lambda) \\ \Rightarrow a \cdot S_1(\lambda) &\cong a \cdot S_2(\lambda) \end{aligned} \quad (2.4)$$

$$\Rightarrow S_1(\lambda) + V(\lambda) \cong S_2(\lambda) + V(\lambda) \quad (2.5)$$

met $V(\lambda)$ een willekeurig spectrum. Het metameer zijn van twee spectra is ook transitief, dus:

$$\begin{aligned} S_1(\lambda) &\cong S_2(\lambda) \quad \text{en} \quad S_2(\lambda) \cong S_3(\lambda) \\ \Rightarrow S_1(\lambda) &\cong S_3(\lambda) \end{aligned} \quad (2.6)$$



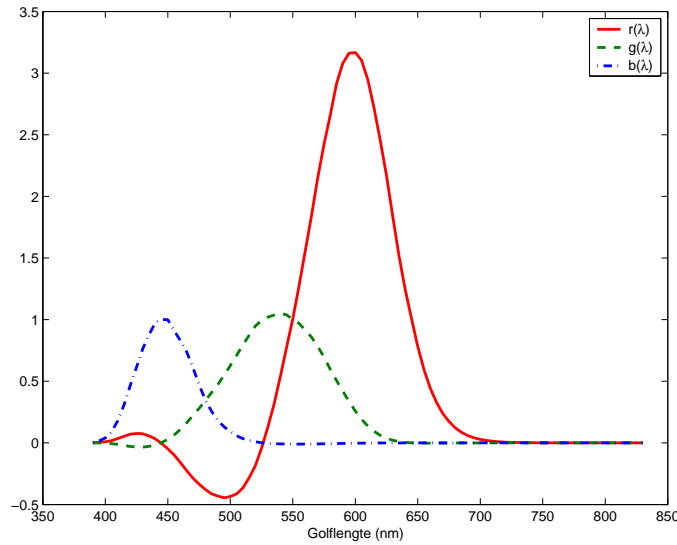
Figuur 2.4: Schematische voorstelling van het color matching experiment.

Merk op dat het metameer zijn niet behouden blijft door beide spectra te *vermenigvuldigen* met eenzelfde spectrum! Dus $S_1(\lambda) \cdot V(\lambda) \not\cong S_2(\lambda) \cdot V(\lambda)$ voor een willekeurig niet-constant spectrum $V(\lambda)$.

2.4 Colorimetrie

Vanwege de drie verschillende receptoren die een mens bezit, kan elke kleur gekarakteriseerd worden door drie getallen. Deze drie getallen worden de *tristimuluswaarden* genoemd. Metamere spectra hebben dezelfde tristimuluswaarden en bij een set tristimuluswaarden horen oneindig veel metamere spectra. Een belangrijk hulpmiddel voor het bepalen van tristimuluswaarden voor een kleur zijn de color matching functies. Wat nu volgt, is een beschrijving van het experiment dat gebruikt kan worden om deze functies te bekomen ((Wright 1928) en (Guild 1931)). Een schematische voorstelling hiervan wordt gegeven in figuur 2.4.

In het experiment wordt een proefpersoon voor een scherm geplaatst waarop twee kleuren geprojecteerd worden. Eén kleur komt van een monochromatische lichtbron. De andere kleur wordt gevormd door de combinatie van drie testlichtbronnen. Er wordt dan aan de proefpersoon gevraagd om de gewichten (intensiteiten) van elk van de testlichten aan te passen om zo een kleur te verkrijgen die dezelfde indruk geeft als de monochromatische kleur. Dit wordt dan herhaald voor andere monochromatische lichtbronnen. De drie gewichten (één voor elke lamp) worden uitgezet in functie van de golflengte van elke monochromatische lichtbron, en zo bekomt men de color



Figuur 2.5: Color matching functies voor een rood, groen en blauw testlicht.

matching functies. De color matching functies die horen bij de testlichten met spectra $X(\lambda)$, $Y(\lambda)$ en $Z(\lambda)$ worden veelal aangeduid met $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ en $\bar{z}(\lambda)$ ². Voor een rood, groen en blauw testlicht zien die er bijvoorbeeld uit zoals in figuur 2.5. Merk op dat er soms negatieve gewichten voorkomen. De fysische betekenis is eenvoudig: het licht met een negatieve coëfficiënt moet bij de monochromatische lichtbron toegevoegd worden, en niet bij de andere lichtbronnen.

Met behulp van de color matching functies kunnen nu aan elke kleursensatie (en dus alle spectra die metameer zijn en die kleursensatie geven) drie getallen toegekend worden. Stel dat we een monochromatisch spectrum $M_1(\lambda)$ hebben met golflengte λ_1 . De kleursensatie die dat spectrum oplevert, kan ook gegenereerd worden met behulp van de testlichten. Hiertoe wordt de waarde van de drie color matching functies op de golflengte λ_1 gebruikt als gewicht voor de drie testlichten, met spectra die we bijvoorbeeld $X(\lambda)$, $Y(\lambda)$ en $Z(\lambda)$ noemen. Zo krijgen we, met $x_1 = \bar{x}(\lambda_1)$, $y_1 = \bar{y}(\lambda_1)$ en $z_1 = \bar{z}(\lambda_1)$, dat

$$M_1(\lambda) \cong x_1 X(\lambda) + y_1 Y(\lambda) + z_1 Z(\lambda). \quad (2.7)$$

Voor een tweede monochromatisch spectrum $M_2(\lambda)$ kan hetzelfde gedaan

²We gebruiken hier $X(\lambda)$, $Y(\lambda)$ en $Z(\lambda)$ en $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ en $\bar{z}(\lambda)$ als algemene voorstelling voor de testlichten en de bijhorende color matching functies. De afleidingen zijn dus niet enkel voor de CIE XYZ-standaard, zoals uit de notatie misschien zou kunnen afgeleid worden. De XYZ-standaard wordt later behandeld.

worden, en met $x_2 = \bar{x}(\lambda_2)$, $y_2 = \bar{y}(\lambda_2)$ en $z_2 = \bar{z}(\lambda_2)$ bekomen we dat

$$M_2(\lambda) \cong x_2 X(\lambda) + y_2 Y(\lambda) + z_2 Z(\lambda). \quad (2.8)$$

Wegens de lineariteit en transitiviteit van metamere spectra geldt dan ook dat

$$M_1(\lambda) + M_2(\lambda) \cong (x_1 + x_2)X(\lambda) + (y_1 + y_2)Y(\lambda) + (z_1 + z_2)Z(\lambda). \quad (2.9)$$

Vergelijking (2.9) is ook gekend als de wet van Grassmann.

Beschouwen we nu een willekeurig spectrum $S(\lambda)$. Met $\delta(\lambda)$ een Dirac-impuls³ kan $S(\lambda)$ ook geschreven worden als

$$S(\lambda) = \int_{-\infty}^{\infty} S(\mu) \cdot \delta(\mu - \lambda) d\mu \quad (2.10)$$

$M_\mu(\lambda) = \delta(\mu - \lambda)$ kan gezien worden als het spectrum van een monochromatische lichtbron met golflengte μ en met vermogen gelijk aan 1 W. Analoog aan 2.7 kan $M_\mu(\lambda)$ dan geschreven worden als

$$M_\mu(\lambda) = \delta(\mu - \lambda) \cong \bar{x}(\mu)X(\lambda) + \bar{y}(\mu)Y(\lambda) + \bar{z}(\mu)Z(\lambda). \quad (2.11)$$

Door dit te gebruiken in 2.10, bekomen we

$$\begin{aligned} S(\lambda) &= \int_{-\infty}^{\infty} S(\mu) \cdot \delta(\mu - \lambda) d\mu \\ &\cong \int_{-\infty}^{\infty} S(\mu) \cdot (\bar{x}(\mu)X(\lambda) + \bar{y}(\mu)Y(\lambda) + \bar{z}(\mu)Z(\lambda)) d\mu \\ &\cong X(\lambda) \int_{-\infty}^{\infty} S(\mu) \bar{x}(\mu) d\mu \\ &\quad + Y(\lambda) \int_{-\infty}^{\infty} S(\mu) \bar{y}(\mu) d\mu \\ &\quad + Z(\lambda) \int_{-\infty}^{\infty} S(\mu) \bar{z}(\mu) d\mu. \end{aligned} \quad (2.12)$$

Met

$$\begin{aligned} x &= \int_{-\infty}^{\infty} S(\lambda) \bar{x}(\lambda) d\lambda \\ y &= \int_{-\infty}^{\infty} S(\lambda) \bar{y}(\lambda) d\lambda \\ z &= \int_{-\infty}^{\infty} S(\lambda) \bar{z}(\lambda) d\lambda \end{aligned} \quad (2.13)$$

³Een Dirac-impuls is een distributie $\delta(x)$ waarvoor geldt dat $\delta(x) = 0$ voor $x \neq 0$ en waarvoor $\int \delta(x) dx = 1$. Men kan dan schrijven dat $f(x) = \int f(t) \delta(t - x) dt$.

wordt (2.12) dus $S(\lambda) \cong xX(\lambda) + yY(\lambda) + zZ(\lambda)$. x , y en z zijn dus de tristimuluswaarden van het spectrum $S(\lambda)$ en kunnen berekend worden via (2.13) voor testlichten $X(\lambda)$, $Y(\lambda)$ en $Z(\lambda)$.

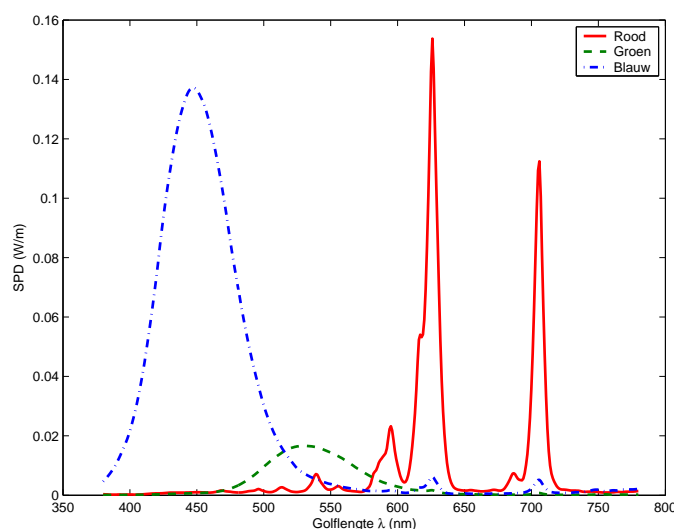
2.5 Kleurenruimtes

Het is al meermaals aangehaald dat elke kleur beschreven kan worden met drie getallen. Het is echter niet vastgelegd hoe deze drie getallen gekozen of berekend moeten worden. Hiervoor zijn dan ook verschillende methodes, die elk een eigen *kleurenruimte* opleveren. De twee belangrijkste kleurenruimtes voor computer graphics, namelijk RGB en XYZ, worden besproken. De RGB-kleurenruimte wordt voornamelijk gebruikt voor de weergave op een beeldscherm, terwijl XYZ vooral bij beeldschermonafhankelijke kleurbeschrijvingen toegepast wordt.

2.5.1 RGB

De tristimuluswaarden van een spectrum $S(\lambda)$ kunnen gebruikt worden om een kleursensatie die hoort bij dat spectrum weer te geven op een scherm. Daarvoor zijn de basisspectra nodig van het scherm, bijvoorbeeld de fosfors van een CRT-scherm. Een CRT-scherm bevat meestal drie soorten fosfors: rood, groen en blauw. De spectra $R(\lambda)$, $G(\lambda)$ en $B(\lambda)$ van die fosfors zouden er kunnen uitzien als in figuur 2.6. Als de spectra van het scherm gekend zijn, kan daarmee een spectrum gegenereerd worden dat metameer is met het oorspronkelijke spectrum $S(\lambda)$. Dat gebeurt door r , g en b uit te rekenen zoals dat in (2.13) gebeurt voor x , y en z . Als de intensiteit waarmee een fosfor oplicht, evenredig genomen wordt met r , g en b , dan vormt de combinatie van drie fosfors een metameer spectrum met het oorspronkelijke spectrum. Alle spectra kunnen zo in principe weergegeven worden.

Er is echter een probleem: de waarden van r , g en b kunnen negatief zijn, of groter dan de maximale intensiteit die met behulp van een scherm gegenereerd kan worden. Het is dus onmogelijk om alle kleursensaties weer te geven op een CRT-monitor. De waarden van r , g en b worden meestal beperkt tot een geheel getal tussen 0 en 255, of tot een floating point-getal tussen 0 en 1. Verder zijn de waarden van r , g en b voor een bepaald spectrum ook afhankelijk van het spectrum van de fosfors die gebruikt worden in de monitor. Een kleur met waarde $(r, g, b) = (1, 0, 0)$ geeft dus niet op elk beeldscherm hetzelfde rood.



Figuur 2.6: Gemiddelde (gescaleerde) SPD's voor de rode, groene en blauwe fosfors van een CRT-monitor (Golz en MacLeod 2003).

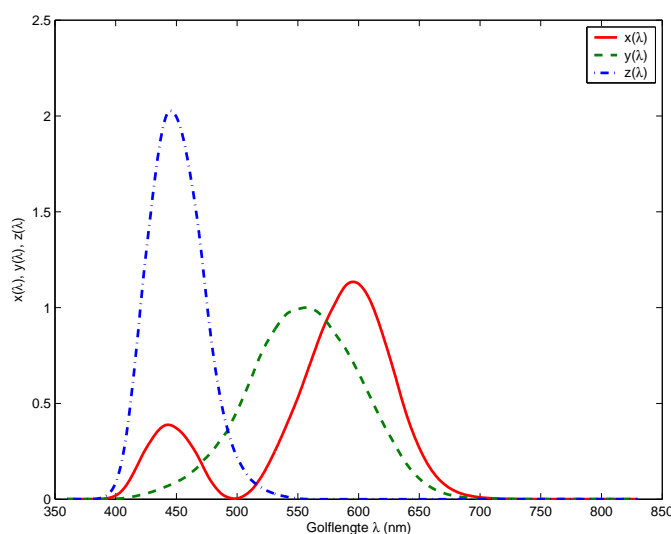
2.5.2 XYZ

Om de problemen van RGB te verhelpen, heeft het Commission Internationale de l'Éclairage (CIE) in 1931 de XYZ color matching functies en dus ook de XYZ-kleurenruimte gedefiniëerd. De color matching functies $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ en $\bar{z}(\lambda)$ zijn zo gedefiniëerd dat ze overal positief zijn (figuur 2.7). Dat betekent dus dat alle kleuren in de XYZ-kleurenruimte met positieve tristimuluswaarden voorgesteld kunnen worden. Dat heeft wel tot gevolg dat de testlichten geen fysisch realiseerbaar spectrum hebben, en in werkelijkheid dus niet bestaan. Dat belet natuurlijk niet dat er mathematisch nog met de spectra $X(\lambda)$, $Y(\lambda)$ en $Z(\lambda)$ gewerkt kan worden.

Er kan een scalering doorgevoerd worden op de x -, y - en z -waarden zodat $x + y + z = 1$. Alle kleuren waarvoor $x + y + z$ gelijk is, hebben eenzelfde intensiteit. Om enkel de kleurtint te bepalen, zijn de x - en y -waarde voldoende. De intensiteit kan hieruit niet meer afgeleid worden. Als ook de intensiteit bepaald moet zijn, kan naast de gescaleerde x en y ook bijvoorbeeld de originele y -waarde⁴ vermeld worden. Dit is dan het xyY-model.

De kleuren kunnen in een diagram gezet worden in functie van de x - en y -waarde van het spectrum. Dit geeft het chromaticiteitsdiagram uit figuur 2.8(a). Hierbij moet gezegd dat niet alle kleuren correct afgedrukt

⁴Men kiest traditioneel voor de y -waarde, omdat deze sterk gerelateerd is aan de luminantie van een kleur.

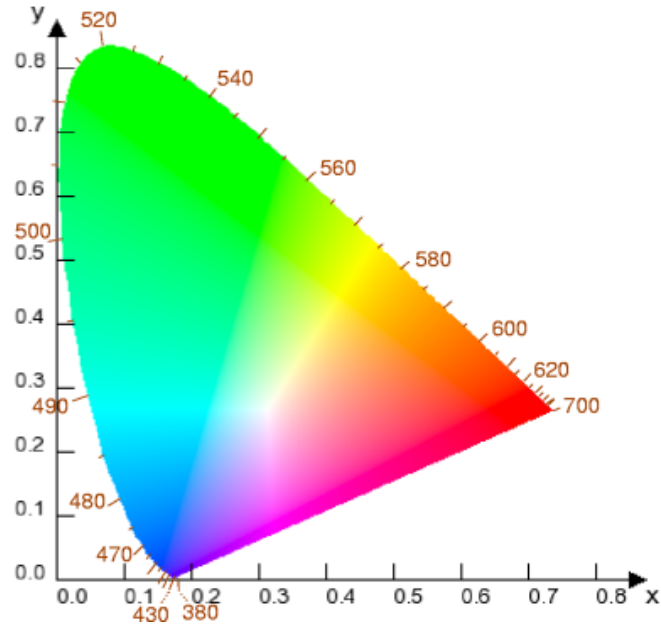


Figuur 2.7: Color matching functies $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ en $\bar{z}(\lambda)$ voor XYZ.

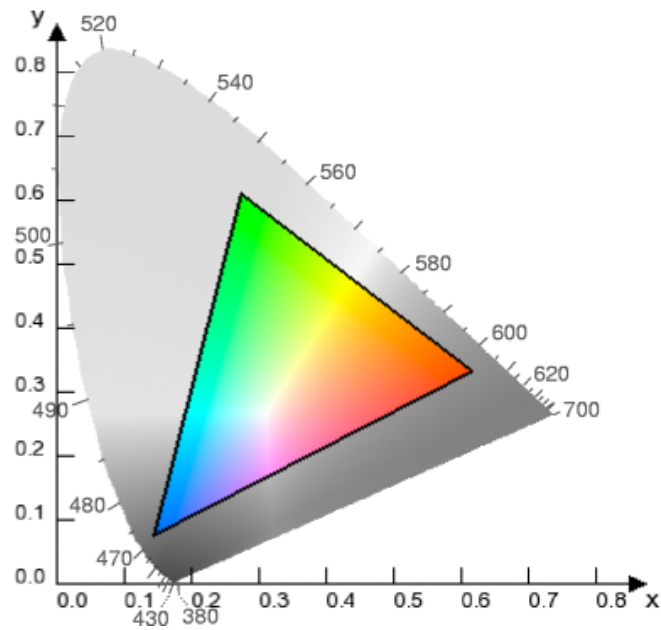
kunnen worden. Het diagram geeft dus slechts een idee van hoe de kleuren verdeeld zijn. Aan de gebogen rand van de hoefijzervorm staan de puur spectrale kleuren. Die gaan van blauw linksonder over groen bovenaan naar rood rechtsonder. De rechte die de blauwe met de rode golflengtes verbindt, noemt men traditioneel de purperlijn. De xy -waarden die buiten het diagram vallen, zijn niet fysisch realiseerbaar.

Neemt men twee spectra $S_1(\lambda)$ en $S_2(\lambda)$ en hun overeenkomstige xy -waarden x_1, y_1, x_2 en y_2 , dan liggen alle spectra die gevormd worden door S_1 en S_2 te mengen, op de rechte tussen (x_1, y_1) en (x_2, y_2) in het diagram. Voor drie spectra liggen alle punten in het diagram binnen de driehoek gevormd door de xy -coördinaten van de drie spectra. Voor een scherm liggen alle kleuren die getoond kunnen worden in de driehoek gevormd door de xy -coördinaten van de fosfors van het scherm. Dit wordt de ‘gamut’ van het scherm genoemd. Een voorbeeld is gegeven in figuur 2.8(b).

RGB en XYZ kunnen naar elkaar geconverteerd worden door middel van een lineaire transformatie. We kunnen immers de spectra van de testlichten



(a)



(b)

Figuur 2.8: (a) Chromaticiteitsdiagram. (b) Gamut van een RGB-monitor.

van RGB ook omzetten naar XYZ. Bijvoorbeeld, voor het rode testlicht is

$$\begin{aligned} x_r &= \int_{-\infty}^{\infty} R(\lambda)\bar{x}(\lambda)d\lambda \\ y_r &= \int_{-\infty}^{\infty} R(\lambda)\bar{y}(\lambda)d\lambda \\ z_r &= \int_{-\infty}^{\infty} R(\lambda)\bar{z}(\lambda)d\lambda. \end{aligned} \quad (2.14)$$

Analoge relaties gelden voor het groene en het blauwe testlicht. Een spectrum $S(\lambda)$ met RGB-waarde (r, g, b) kan dan beschreven worden als

$$\begin{aligned} S(\lambda) &\cong rR(\lambda) + gG(\lambda) + bB(\lambda) \\ &\cong r(x_r X(\lambda) + y_r Y(\lambda) + z_r Z(\lambda)) \\ &\quad + g(x_g X(\lambda) + y_g Y(\lambda) + z_g Z(\lambda)) \\ &\quad + b(x_b X(\lambda) + y_b Y(\lambda) + z_b Z(\lambda)) \\ &= (rx_r + gx_g + bx_b)X(\lambda) + (ry_r + gy_g + by_b)Y(\lambda) \\ &\quad + (rz_r + gz_g + bz_b)Z(\lambda) \\ &= xX(\lambda) + yY(\lambda) + zZ(\lambda) \end{aligned} \quad (2.15)$$

of dus:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_r & x_g & x_b \\ y_r & y_g & y_b \\ z_r & z_g & z_b \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix} \quad (2.16)$$

2.6 RGB en spectra

Er werden tot hiertoe twee manieren gegeven waarmee kleur beschreven kan worden. Er werd gesteld dat een kleur bepaald wordt door een spectrum en dat elk spectrum voor het menselijk visueel systeem gekarakteriseerd kan worden door een set van drie waarden zoals in RGB. Deze manieren van beschrijven zijn echter niet equivalent wanneer het gaat over het rekenen met kleur. Deze sectie behandelt de beperkingen van het rekenen met RGB tegenover rekenen met de volledige spectrale informatie. Ook worden enkele technieken besproken om, uitgaande van een RGB-waarde, een spectrum te construeren met deze RGB-waarde.

2.6.1 Beperkingen van RGB

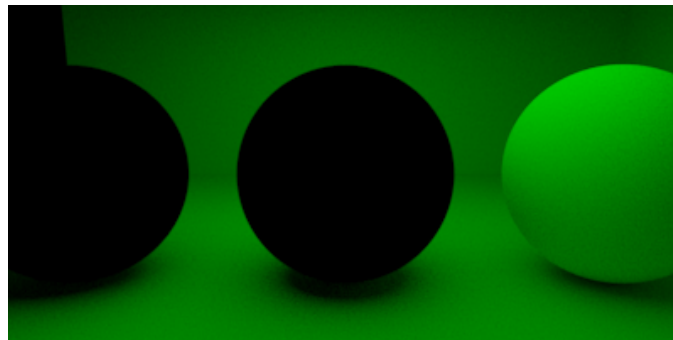
Volgens de wet van Grassmann (vergelijking (2.9)) kan er voor lineaire operaties op spectra met de tristimuluswaarden gewerkt worden. De som van twee

spectra zal dus als RGB-waarde de som van de individuele RGB-waarden van de spectra hebben. Hetzelfde geldt voor de scalering van een spectrum met een constante. Veel operaties kunnen dus evengoed in RGB of XYZ uitgevoerd worden als met de volledige spectrale informatie.

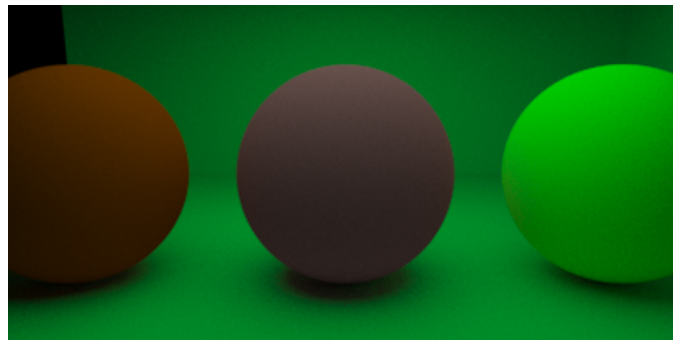
Dit kan echter niet altijd! Zo kan de vermenigvuldiging van twee spectra (bijvoorbeeld een SPD van een lichtbron en de reflectantie van een materiaal) niet correct berekend worden door enkel gebruik te maken van de RGB-waarden. Het is jammer genoeg juist deze operatie die zeer vaak voorkomt bij het genereren van fotorealistische beelden. Het gebruik van RGB (of XYZ) hiervoor is slechts een benadering, die geen garantie biedt op een juist resultaat.

Dit wordt geïllustreerd met twee voorbeelden. Het eerste voorbeeld (figuur 2.9) toont drie bollen. Een rode bol met als RGB-waarde $(1, 0, 0)$, een paarse met RGB-waarde $(1, 0, 1)$ en een gele bol met RGB-waarde $(1, 1, 0)$. De bollen worden belicht met een groene lamp (RGB-waarde $(0, 1, 0)$). Het resultaat gebruik makende van RGB (figuur 2.9(a)) is eenvoudig te berekenen als het product van de reflectantie van de bollen met het licht: $(0, 0, 0)$, $(0, 0, 0)$ en $(0, 1, 0)$. De twee linkse bollen zijn zwart geworden omdat ze hun reflectantie geen groene component bevat. De gele bol is groen, omdat het groene licht geen rode component bevat. De scène werd ook spectraal gerenderd. Hiertoe werden alle RGB-waarden vervangen door een uniform bemonsterd spectrum. De gebruikte spectra werden afgeleid van de RGB-waarden via de methode van Smits, die later toegelicht wordt. De gebruikte spectra hebben als bijhorende RGB-waarde de hierboven gegeven waarden. Het resultaat staat in figuur 2.9(b). De originele kleur van de rode en paarse bol is nog duidelijk herkenbaar. Dat komt omdat de gegenereerde spectra over het gehele zichtbare gebied gedefinieerd zijn. De groene lichtbron bevat dus ook nog een (zwakker) gedeelte in het rood en blauw gebied. Dit is ook zo in realiteit: de meeste lichtbronnen hebben een breed spectrum.

Voor het tweede voorbeeld werd een scène geconstrueerd en gefotografeerd, en vervolgens een overeenkomstige scène gemodelleerd en gerenderd. De spectraal gerenderde scènes gebruiken zelf opgemeten waarden voor de spectra, de RGB-scènes de RGB-waarden van die opgemeten spectra. De resultaten van dit experiment vindt men terug in figuur 2.10. De linkerkolom bevat steeds de originele scène. Voor de afbeeldingen in de rechterkolom werd een oranje fotografische filter geplaatst voor de cameralens. (We gebruikten een echte filter voor het nemen van de foto en de opgemeten waarden ervan voor het maken van de afbeeldingen). Uit de afbeeldingen blijkt dat de RGB-versie een aanvaardbaar resultaat oplevert, zeker wanneer er geen oranje filter gebruikt wordt. Bij de oranje filter geeft RGB hier en daar ech-

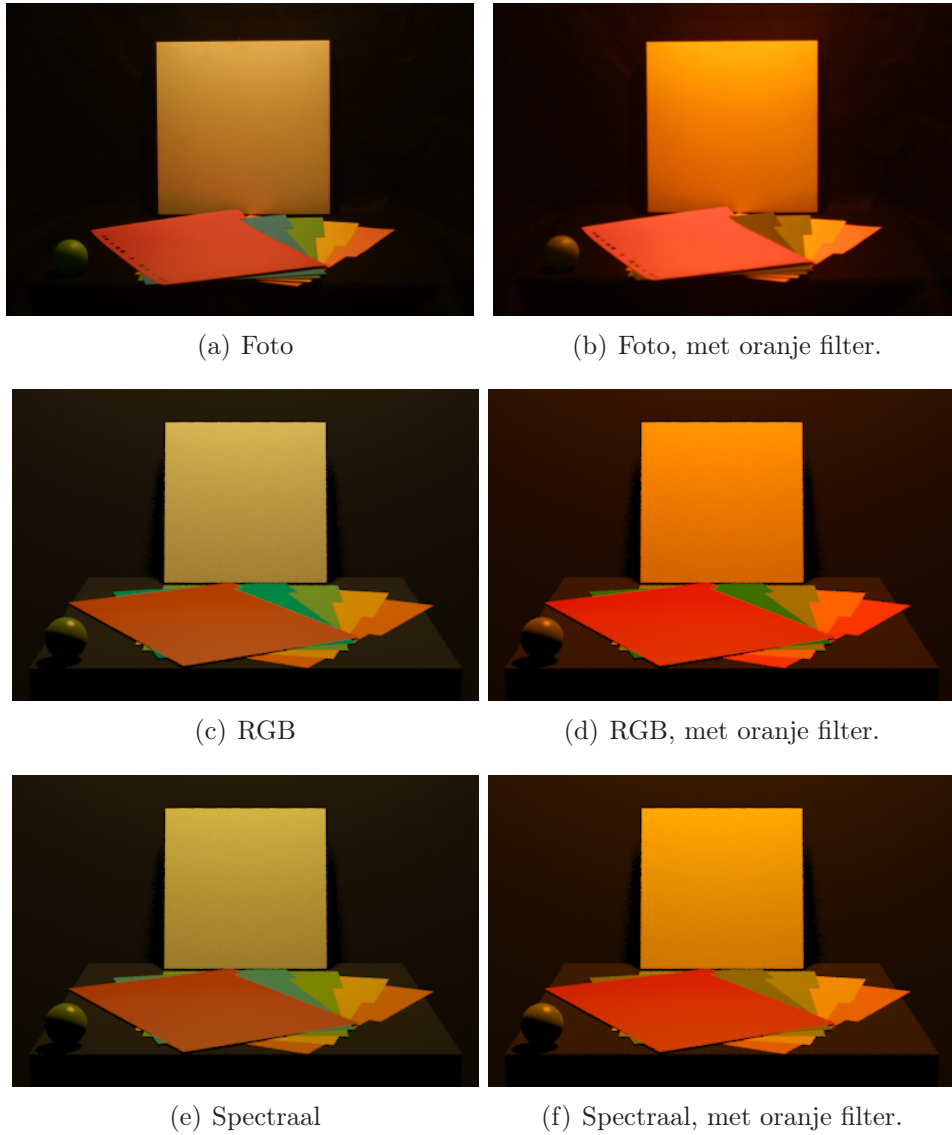


(a) RGB (256 stralen per pixel)



(b) Spectraal (uniform bemonsterd, 31 monsters van 360 tot 830 nm, 256 stralen per pixel)

Figuur 2.9: Vergelijking tussen RGB en spectraal; v.l.n.r. telkens een rode, paarse en gele diffuse bol, belicht met een groene lamp.



Figuur 2.10: Vergelijking tussen een echte scène en een overeenkomstige virtuele scène, gerenderd met een RGB en een spectraal kleurmodel (uniform bemonsterd, 48 monsters van 360 tot 830 nm). Links zonder filter, rechts met een oranje filter.

ter sterk afwijkende kleuren. Een voorbeeld daarvan is het blauwe tabblad (tweede van links), dat volledig groen geworden is. Ook het gele en oranje tabblad (de twee rechtse) met de oranje filter zijn sterk van kleur veranderd, naar oranje en rood toe. De spectrale versie benadert de foto beter dan het RGB-model.

Het is echter niet zo dat de spectrale versie volledig overeenstemt met de foto. Hiervoor zijn meerdere oorzaken. Ten eerste was er geen kalibrering van het gebruikte fototoestel beschikbaar. Vervolgens blijken de metingen van de spectra in het blauwe gebied minder nauwkeurig te zijn dan in de andere gebieden. Ten derde is van de materialen ook telkens maar één spectrale meting gedaan en werd de veronderstelling gemaakt dat alle materialen diffuus zijn. Er is dus geen volledig spectrale BRDF opgemeten en gebruikt. Ten slotte komt het 3D-model natuurlijk niet volledig overeen met de gefotografeerde scène.

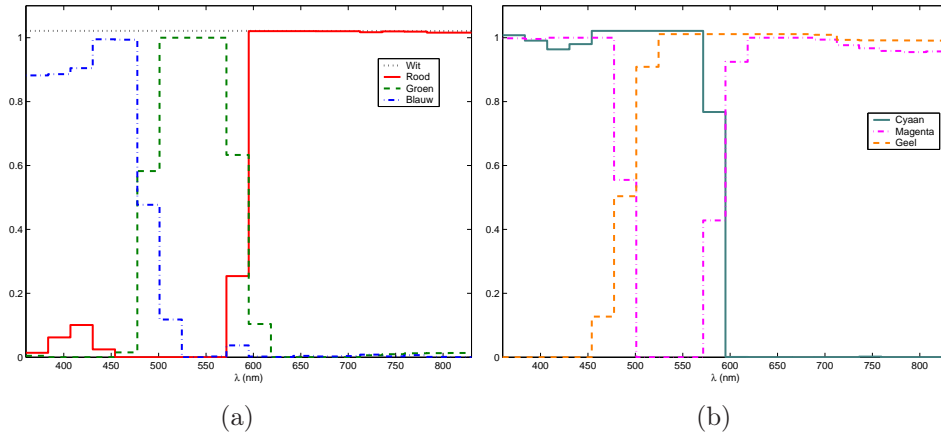
2.6.2 Omzetting van RGB naar spectrum

In veel gevallen zou het eenvoudiger zijn om een kleur nog steeds te specificeren aan de hand van zijn RGB-waarde, omdat dat traditioneel zo gebeurt, maar om vanaf dan voort te werken met een volledig spectrum. We hebben dan een omzetting nodig van RGB naar een spectrum. Aangezien met één RGB-waarde oneindig veel metamere spectra overeenkomen, hebben we dus in principe ook oneindig veel manieren om deze omzetting te doen. We bespreken er twee, namelijk de methode van Smits en die van Sun.

Methode van Smits

De methode van Smits wordt beschreven in (Smits 1999). Men bepaalt zeven basisspectra. De spectra worden zodanig gekozen dat hun RGB-waarden als volgt zijn: $(1, 0, 0)$ voor rood, $(0, 1, 0)$ voor groen, $(0, 0, 1)$ voor blauw, $(0, 1, 1)$ voor cyaan, $(1, 0, 1)$ voor magenta, $(1, 1, 0)$ voor geel en $(1, 1, 1)$ voor wit. Verder wordt getracht om de spectra zacht te laten verlopen, omdat de meeste fysische spectra dat ook doen. Ten slotte wil men de spectra ook bruikbaar maken als reflectantiespectrum. De waarden zouden daarom in het ideale geval tussen 0 en 1 moeten liggen. Voor de gedetailleerde bepaling van de basisspectra verwijzen we naar (Smits 1999). In figuur 2.11 zijn de basisspectra afgebeeld.

In principe zou men ook kunnen werken met slechts drie basisfuncties, bijvoorbeeld rood, groen en blauw. Dat geeft echter minder goede resultaten, omdat een apart wit spectrum beter is dan de som van het rood, groen en



Figuur 2.11: Basisspectra voor de methode van Smits. (a) Rood, groen, blauw en wit. (b) Cyaan, magenta, geel.

blauw. Als resultaat wordt immers een zacht verlopend spectrum verwacht en dat kan beter bereikt worden door een nieuw basisspectrum te bepalen. In figuur 2.12 wordt dit geïllustreerd. Hetzelfde geldt voor magenta als de som van rood en blauw, cyaan als groen en blauw en voor geel als rood en groen.

De eigenlijke omzetting van RGB naar een spectrum gebeurt nu als volgt. Stel dat een kleur (r, g, b) omgezet moet worden naar een spectrum. Eerst wordt de hoeveelheid wit w die in de kleur aanwezig is, bepaald. Deze kan gevonden worden als $w = \min(r, g, b)$. De kleur kan dus ook geschreven worden als:

$$(r, g, b) = w \cdot (1, 1, 1) + (r - w, g - w, b - w) = w \cdot (1, 1, 1) + (r', g', b').$$

Minstens één van de componenten van de resterende kleur is nu gelijk aan 0. Stel dat dit $r' = r - w$ is (de rode component was dus het kleinst). De overblijvende kleur is dan $(0, g', b')$. Hiervan kan de hoeveelheid cyaan (groen+blauw) bepaald worden als $c = \min(g', b')$. De kleur wordt dan:

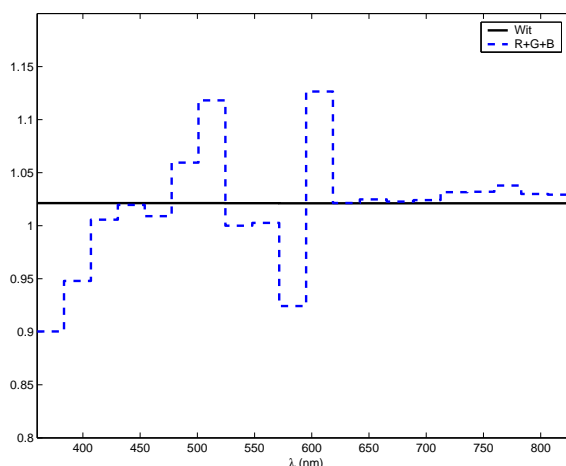
$$(r, g, b) = w \cdot (1, 1, 1) + c \cdot (0, 1, 1) + (0, g' - c, b' - c)$$

Stel dat deze keer de groene component de kleinste was ($c = g'$), dan blijft er nog $(0, 0, b' - c) = (0, 0, b'')$ over. Daarvan kan dus nog b'' keer blauw weggenomen worden. Het uiteindelijke resultaat wordt dan dat:

$$(r, g, b) = w \cdot (1, 1, 1) + c \cdot (0, 1, 1) + b'' \cdot (0, 0, 1)$$

en er kan dus een spectrum gegenereerd worden met RGB-waarde (r, g, b) door

$$S(\lambda) = wW(\lambda) + cC(\lambda) + b''B(\lambda)$$



Figuur 2.12: De som van het rode, groene en blauwe basisspectrum (streep-lijn) en het witte basisspectrum (volle lijn).

waarbij $W(\lambda)$, $C(\lambda)$ en $B(\lambda)$ de witte, cyaan en blauwe basisspectra zijn, en w , c en b'' de coëfficiënten zoals hierboven bepaald. Analoge redeneringen gelden wanneer andere componenten het minimum zijn.

Methode van Sun

Sun beschrijft een andere manier van omzetten in (Sun *et al.* 1999). Het basisidee is om een spectrum voor te stellen als de lineaire combinatie van drie Gauss-curves. De gewichten van elke Gauss-curve worden bekomen door de Gauss-curves te gebruiken als basis voor de kleurenruimte en vervolgens een transformatie van de RGB-ruimte naar die ruimte door te voeren (analoog aan (2.15)-(2.16)). De breedte van de Gauss-curves wordt gekozen in functie van de lokale saturatie van de kleur. Een gesatureerde kleur (een kleur die dicht aanleunt bij een puur spectrale kleur) zal zo voorgesteld worden door een smalle curve, terwijl een eerder ongesatureerde kleur een bredere curve zal opleveren.

De basisfuncties die gebruikt worden, hangen dus af van de kleur die omgezet zal worden. Het is dus niet mogelijk op voorhand de coëfficiënten te berekenen zoals in (2.16). Elke keer een omzetting gebeurt, dienen deze coëfficiënten berekend te worden. De berekening hiervan volgt (2.13) en bevat dus een integratie van de basisfuncties. De basisfuncties zijn exponentiële functies, die daarenboven wat meer tijd kosten om te evalueren. Bovendien blijkt dat, om nauwkeurige resultaten te behalen door de omzetting, een spectrum nodig is dat minstens op een 50-tal plaatsen bemonsterd is. De

methode van Smits heeft ter vergelijking slechts een 20-tal monsters nodig. Om deze redenen gebruiken we steeds de methode van Smits. De methode van Sun is wel geïmplementeerd en zou dus gebruikt kunnen worden indien gewenst.

Spectraal raamwerk

Om te kunnen werken met de volledige spectrale informatie, moeten de spectra op een bepaalde wijze voorgesteld worden. Hiervoor bestaan verschillende manieren, elk met hun voor- en nadelen. In dit hoofdstuk worden enkele manieren beschreven en geëvalueerd. Er wordt ook wat dieper ingegaan op de conversie tussen de verschillende voorstellingswijzen. Tenslotte wordt de manier waarop dit uiteindelijk geïmplementeerd wordt, kort toegelicht.

3.1 Spectrale voorstellingen

3.1.1 Overzicht

Men kan verschillende manieren bedenken om een spectrum voor te stellen. Al deze manieren kunnen onderverdeeld worden in 4 grote types (Sun 2000), namelijk puntbemonstering (*point sampling*), veeltermvoorstelling (*polynomial representation*), lineair model (*linear model*) en kleurmodel (*color model*). In (Sun 2000) wordt aangegeven dat veeltermvoorstellingen veel nadelen hebben voor het voorstellen van spectra. Zo kan de berekening instabiel worden als een hoge nauwkeurigheid gevraagd wordt. Verder is de methode niet flexibel en niet efficiënt voor vermenigvuldiging. Deze manier zal dus verder buiten beschouwing gelaten worden. De andere methoden worden kort toegelicht.

- **Puntbemonstering:** Bij puntbemonstering wordt een spectrum voorgesteld door een aantal discrete punten. Tussen de punten kan er geïnterpoleerd worden op verschillende manieren. Men kan bijvoorbeeld voor elke golflengte de waarde nemen van het dichtstbijzijnde meetpunt, of lineair interpoleren tussen de meetpunten. Het voordeel van deze methode is de grote nauwkeurigheid die bereikt kan worden, ten koste van het aantal punten. Ook kan er zeer efficiënt mee gerekend

worden. Er zullen later twee puntbemonsteringsmethodes van naderbij bekeken worden, namelijk het uniform bemonsterd spectrum en het getabuleerd spectrum.

- **Lineair model:** Een spectrum kan ook voorgesteld worden als de lineaire combinatie van een aantal lineair onafhankelijke basisfuncties. Meer bepaald kan een spectrum $S(\lambda)$ voorgesteld worden als

$$S(\lambda) = \sum_{i=0}^{\infty} a_i B_i(\lambda)$$

met a_i de coëfficiënt van de i -de basisfunctie $B_i(\lambda)$. Voor praktisch gebruik wordt natuurlijk slechts een eindig aantal basisfuncties en coëfficiënten gebruikt. Als de basisfuncties gekend zijn, hoeven enkel de coëfficiënten bewaard te worden. Dit model is zeer compact en flexibel. Het vraagt echter meer rekenwerk. Het Fourierspectrum dat later behandeld wordt, is een voorbeeld van een spectrum volgens het lineair model.

- **Kleurmodel:** Zoals besproken in secties 2.4 en 2.5 kan een kleur ook voorgesteld worden door 3 waarden. Een spectrum zou dus ook gekarakteriseerd kunnen worden door zijn RGB- of XYZ-waarden. Dit is een zeer compacte, snelle en intuïtieve voorstelling om mee te werken. Er werd echter al aangegeven dat deze manier niet voldoende accuraat is om berekeningen mee uit te voeren.

3.1.2 Bemonsterd spectrum

De uniform bemonsterde voorstellingswijze is veruit de eenvoudigste en meest voor de hand liggende manier om een spectrum voor te stellen. Op n equidistante punten tussen een minimumgolflengte λ_{\min} en een maximumgolflengte λ_{\max} wordt de waarde van het spectrum bepaald (bemonsterd). Het spectrum wordt gelijk gesteld aan 0 buiten het interval $[\lambda_{\min}, \lambda_{\max}]$. De bemonsterde waarden worden samen met n , λ_{\min} en λ_{\max} bijgehouden.

Het voordeel van deze methode is dat ze eenvoudig en snel is. Ze kan zo nauwkeurig gemaakt worden als nodig, door het aantal bemonsteringen te verhogen. Verder is dit ook een intuïtieve manier van werken om spectra te definiëren. De nadelen zijn de tamelijk grote hoeveelheid data, vooral bij een grote nauwkeurigheid. Bij een te lage bemonsteringsfrequentie kan het zijn dat belangrijke variaties gemist worden. Een te hoge frequentie leidt dan weer tot teveel data.

Bewerkingen met bemonsterde spectra zijn eenvoudig wanneer de monsters op dezelfde plaats genomen zijn (met andere woorden, wanneer λ_{\min} , λ_{\max} en n gelijk zijn voor beide spectra). De bewerking kan dan gewoon uitgevoerd worden op de overeenkomende monsters. In het andere geval wordt de bewerking uitgevoerd op de waarden van het spectrum met de kleinste afstand tussen de monsters, om geen informatie verloren te laten gaan.

3.1.3 Getabuleerd spectrum

Het getabuleerd spectrum lijkt sterk op het bemonsterd spectrum. Bij het getabuleerd spectrum wordt echter niet meer gewerkt met equidistante monsters, maar met koppels $(\lambda_i, S(\lambda_i))$. Er wordt dus niet enkel de bemonsterde waarde bijgehouden, maar ook de golflengte waarbij het monster hoort. Tussen de waarden kan opnieuw geïnterpoleerd worden volgens een methode naar keuze.

Het grootste voordeel van deze manier van voorstellen is dat de plaats van de monsters gekozen kan worden. Zo kan een sterk variërend gebied in een spectrum meer monsters krijgen dan een eerder vlak deel. Dat gaat wel ten koste van het geheugengebruik: er is nu twee keer zoveel data nodig als bij een bemonsterd spectrum met hetzelfde aantal monsters. Ook deze manier van voorstellen is intuïtief om te gebruiken.

3.1.4 Fourierspectrum

Een derde voorstellingswijze maakt gebruik van de Fourier-voorstelling van het spectrum. Een spectrum $S(\lambda)$ gedefinieerd over een gebied $[\lambda_{\min}, \lambda_{\max}]$ kan geschreven worden als

$$S(\lambda) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{2\pi n(\lambda - \lambda_{\min})}{\Delta\lambda}\right) + b_n \sin\left(\frac{2\pi n(\lambda - \lambda_{\min})}{\Delta\lambda}\right) \quad (3.1)$$

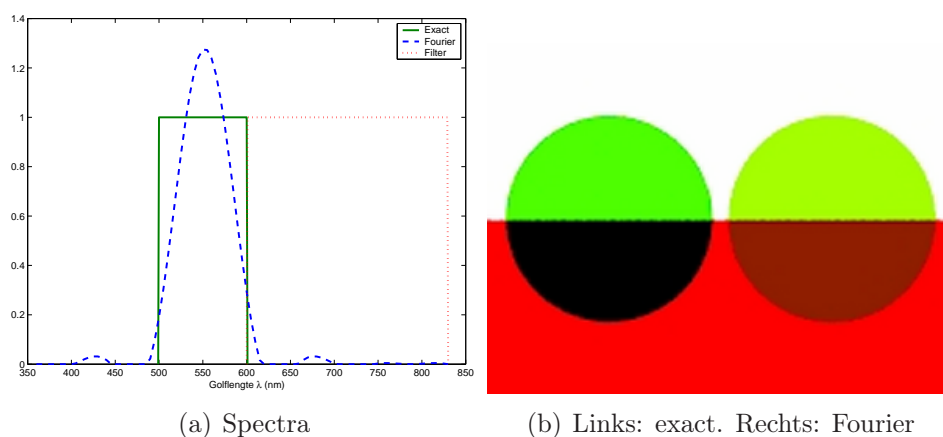
waarbij $\Delta\lambda = \lambda_{\max} - \lambda_{\min}$ en de coëfficiënten a_n en b_n bepaald worden door

$$a_n = \frac{2}{\Delta\lambda} \int_{\lambda_{\min}}^{\lambda_{\max}} S(\lambda) \cos\left(\frac{2\pi n(\lambda - \lambda_{\min})}{\Delta\lambda}\right) d\lambda, \quad n = 0 \dots \infty \quad (3.2)$$

en

$$b_n = \frac{2}{\Delta\lambda} \int_{\lambda_{\min}}^{\lambda_{\max}} S(\lambda) \sin\left(\frac{2\pi n(\lambda - \lambda_{\min})}{\Delta\lambda}\right) d\lambda, \quad n = 1 \dots \infty \quad (3.3)$$

Om het spectrum te benaderen, worden enkel de eerste N termen van 3.1 behouden: dit zijn de laagste N frequenties. Hogere frequenties worden



Figuur 3.1: Fourierbenadering van een rechthoekig spectrum.

verwaarloosd. Wanneer het spectrum zacht verloopt, bekomt men zo een goede benadering. In (Sun 2000) wordt aangetoond dat 9 tot 11 coëfficiënten volstaan voor de meeste zacht verlopende spectra.

Een gewenste nauwkeurigheid kan bovendien eenvoudig bereikt worden door het aantal coëfficiënten te kiezen (en de hogere frequenties weg te laten vallen). Door de hogere frequenties te verwaarlozen, worden minder fouten geïntroduceerd dan bij het verwaarlozen van lagere frequenties. Dit zou bijvoorbeeld niet kunnen bij een bemonsterd of getabuleerd spectrum. Daar bestaat geen eenvoudige manier om monsters weg te laten. Om met minder monsters te werken, zal in dat geval het spectrum herbemonsterd moeten worden.

Een Fourier-spectrum is echter niet geschikt voor spectra met steil verlopende flanken. Dergelijke spectra bevatten immers zeer veel hogere frequenties, die niet in rekening gebracht worden. Dit wordt geïllustreerd in figuur 3.1. Er wordt een rechthoekig basisspectrum van 500 nm tot 600 nm gebruikt. In figuur 3.1(a) wordt dit basisspectrum en zijn Fourierbenadering (met 9 coëfficiënten) afgebeeld. Figuur 3.1(b) geeft het resultaat bij het gebruik van deze twee spectra. Bovenaan zonder filter, onderaan gebruik makend van een rode filter (van 600 nm tot 830 nm). Het is duidelijk dat het Fourier-spectrum niet-verwaarloosbare componenten bevat in het gebied buiten dat van het oorspronkelijke spectrum. Bij reële spectra komen dergelijke steile overgangen nauwelijks voor en zal dit fenomeen wat afgezwakt worden.

De Fourier-voorstelling gebruikt weinig data en laat dus een compacte voorstelling van een spectrum toe. Daartegenover staat dat deze manier niet erg intuïtief werkt. Verder is de vermenigvuldiging van twee Fourier-spectra

een complexe operatie die $\mathcal{O}(N^2)$ bewerkingen vraagt. Bovendien is voor de evaluatie van een spectrum dat op deze manier voorgesteld wordt een reeks evaluaties van sinus- en cosinusfuncties nodig. Dit alles maakt rekenen met de Fourier-voorstelling minder aantrekkelijk.

Daarom wordt in (Sun 2000) voorgesteld om de Fourier-voorstelling enkel te gebruiken om zacht verlopende spectra met een hoge graad van nauwkeurigheid op te slaan, bijvoorbeeld in een databank met spectra. Bij het gebruik van een spectrum wordt dan de gewenste nauwkeurigheid bepaald door de eerste N coëfficiënten te gebruiken. Het spectrum dat zo bekomen wordt, kan dan bemonsterd worden. Vanaf dan wordt het bemonsterde spectrum gebruikt om berekeningen uit te voeren.

3.1.5 Composietspectrum

Het composietmodel werd voorgesteld in (Sun *et al.* 1998). Bij deze manier van beschrijven wordt het spectrum opgesplitst in twee delen: een zacht verlopend spectrum (S_{smooth}) en een deel dat bestaat uit pieken (S_{spikes}). Dus,

$$S(\lambda) = S_{\text{smooth}}(\lambda) + S_{\text{spikes}}(\lambda). \quad (3.4)$$

Voor S_{smooth} wordt in (Sun *et al.* 1998) een Fourier-spectrum gebruikt. In onze implementatie hebben we dit uitgebreid zodat hier een willekeurige voorstelling gebruikt kan worden. Zo kan men het type spectrum gebruiken dat het best geschikt lijkt voor de data.

S_{spikes} bevat de eventuele pieken van het spectrum. Een piek wordt voorgesteld als een Dirac-impuls op een bepaalde golflengte λ_i en met een bepaald gewicht w_i . Voor m pieken geeft dit

$$S_{\text{spikes}}(\lambda) = \sum_{i=1}^m w_i \delta(\lambda - \lambda_i). \quad (3.5)$$

Het grote voordeel van composietspectra is dat ze eenvoudig spectra met pieken kunnen voorstellen. Dergelijke spectra zijn belangrijk wanneer we bijvoorbeeld TL-verlichting willen gebruiken in onze scène. Een voorbeeld van het spectrum van een TL-lamp werd gegeven in figuur 2.1(b). Dergelijke spectra zijn zeer moeilijk voor te stellen met de andere besproken manieren, omdat dat zeer veel data zou vergen. Het composietmodel is hier daarentegen uiterst geschikt voor.

3.2 Implementatie

3.2.1 Spectrum

De belangrijkste methodes uit de interface waaraan elk type spectrum moet voldoen, worden gegeven in listing 3.1. Een spectrum moet methodes aanbieden om de waarde op bepaalde golflengtes op te vragen, er moeten bewerkingen uitgevoerd kunnen worden op spectra en een spectrum moet geconverteerd kunnen worden naar andere types. De implementatie van deze methodes is voor de meeste spectra vrij vanzelfsprekend. Speciale aandacht moet echter besteed worden aan de **value**-methodes voor RGB en XYZ. Aangezien deze voorstellingswijzen een volledige verzameling van (metamere) spectra beschrijft, is het moeilijk te bepalen wat het resultaat van deze methode moet zijn. Onze implementatie geeft dan ook steeds 0 terug, en er werd voor gezorgd dat deze methode in de rest van het systeem nooit gebruikt zou worden voor RGB- en XYZ-spectra. Tenslotte willen we opmerken dat het achteraf beschouwd nuttig had geweest om ook een methode te voorzien om een spectrum te bemonsteren. Aangezien de nood hiervoor zich slechts laat in de ontwikkeling van het systeem voordeed en niet absoluut noodzakelijk was, hebben we dit niet meer gedaan.

3.2.2 Conversie

In principe zou elk spectrum in zijn eigen implementatie kunnen bepalen hoe het geconverteerd wordt naar alle andere types. Dit heeft echter twee nadelen: ten eerste zou de toevoeging van een nieuw type spectrum dan vereisen dat alle bestaande spectra aangepast worden, om ook te kunnen converteren naar het nieuwe spectrum. Ten tweede zou dit leiden tot veelvuldige codeduplicatie. De code voor het converteren naar XYZ is bijvoorbeeld identiek voor alle spectra (behalve RGB). Het is daarom beter dat die code zich slechts op één plaats bevindt.

We hebben daarom gekozen voor een ontwerp met zogenaamde convertoren. Er wordt één standaardconvector voorzien die zo goed mogelijk naar alle types kan converteren. Het is verder mogelijk om, afhankelijk van het type van het bronspectrum, een nieuwe convector te voorzien. Die wordt dan automatisch gebruikt in plaats van de standaardconvector en kan gebruikt worden om rekening te houden met de specifieke eigenschappen van het bronspectrum. Zo is er bijvoorbeeld een convector voorzien voor composietspectra, die rekening houdt met de pieken in composietspectra. De standaardconvector kan immers enkel de basisinterface van een spectrum gebruiken en daarin

Listing 3.1: BaseSpectrum interface.

```

class BaseSpectrum {
public:
    /* Opvragen van waarde op 1 golflengte of meerdere golflengtes tegelijk */
    virtual float value (const float wavelength) const = 0;
    virtual float* values(const float *const wavelengths, const unsigned int len) const = 0;

    /* Operaties
     * Het argument (<s>) zal geconverteerd worden naar the type van het spectrum
     *   waarop de operatie uitgevoerd wordt (<this>) voor de operatie.
     * Het resultaat heeft hetzelfde type als het spectrum waarop de operatie
     *   uitgevoerd wordt.
     */
    /* basisbewerkingen */
    virtual BaseSpectrum* operator+(const BaseSpectrum& s) const = 0;
    virtual BaseSpectrum* operator-(const BaseSpectrum& s) const = 0;
    virtual BaseSpectrum* operator*(const BaseSpectrum& s) const = 0;
    virtual BaseSpectrum* operator/(const BaseSpectrum& s) const = 0;
    virtual BaseSpectrum* operator^(const BaseSpectrum& s) const = 0; // macht

    virtual BaseSpectrum& operator+=(const BaseSpectrum& s) = 0;
    virtual BaseSpectrum& operator-=(const BaseSpectrum& s) = 0;
    virtual BaseSpectrum& operator*=(const BaseSpectrum& s) = 0;
    virtual BaseSpectrum& operator/=(const BaseSpectrum& s) = 0;
    virtual BaseSpectrum& operator^=(const BaseSpectrum& s) = 0;

    /* vermenigvuldiging/deling/machtsverheffing met constante */
    virtual BaseSpectrum* operator*(const float t) const = 0;
    virtual BaseSpectrum& operator*=(const float t) = 0;
    virtual BaseSpectrum* operator/(const float t) const = 0;
    virtual BaseSpectrum& operator/=(const float t) = 0;
    virtual BaseSpectrum* operator^(const float t) const = 0;
    virtual BaseSpectrum& operator^=(const float t) = 0;

    /* negatie */
    virtual BaseSpectrum* operator-() const = 0;
    /* exponentiele */
    virtual BaseSpectrum* _exp() const = 0;
    /* vierkantswortel */
    virtual BaseSpectrum* _sqrt() const = 0;

    /* beperk waarden tot [min, max] */
    virtual BaseSpectrum* clamp(const float min, const float max) const = 0;
    /* maximale waarde */
    virtual float maxcomponent() const = 0;
    /* minimale waarde */
    virtual float mincomponent() const = 0;

    /* alle waarden 0? */
    virtual bool isBlack() const = 0;

    /* Conversies naar andere types (gebruikt convertoren) */
    virtual operator XYZSpectrum() const;
    virtual operator RGBSpectrum() const;
    virtual operator SampledSpectrum() const;
    virtual operator TabulatedSpectrum() const;
    virtual operator FourierSpectrum() const;
    virtual operator CompositeSpectrum() const;

    /* andere methodes weggelaten */
};

```

zijn geen mogelijkheden voorzien om informatie over pieken op te vragen.

Vermits het gebruik van verschillende voorstellingswijzen van spectra in dezelfde scene toegelaten zijn, moet er ook beslist worden wat het type het resultaat van een bewerking op twee spectra is. Zo kan het voorkomen dat een RGB-spectrum vermenigvuldigd dient te worden met een bemonsterd spectrum. Het resultaat daarvan kan zowel een RGB-spectrum als een bemonsterd spectrum zijn. Merk op dat in de interface van het basisspectrum in listing 3.1 bepaald wordt dat het resultaat het type heeft van de linkeroperand. Dat zou echter betekenen dat het resultaat afhangt van hoe de operatie geschreven wordt. Dit zou snel leiden tot fouten, en zou ook een aanpassing vereisen doorheen heel PBRT om alle bewerkingen in de juiste volgorde te zetten. In ons raamwerk wordt rekening gehouden met deze conventie, maar in PBRT kunnen we daar niet op rekenen. Daarom wordt bij elke bewerking op een **Spectrum** uit PBRT eerst het resultaattype bepaald, door te gaan kijken in een op voorhand gedefiniëerde lijst. In die lijst staat dan bijvoorbeeld dat een bewerking op een RGB-spectrum en een bemonsterd spectrum een RGB-spectrum teruggeeft. De keuze is arbitrair, en kan aangepast worden indien nodig. Wij hebben ervoor gekozen om een RGB-spectrum steeds te behouden als RGB-spectrum.

3.2.3 Invoerformaat

In de originele implementatie van PBRT is slechts één mogelijkheid voorzien om een kleur op te geven. Dat gebeurt aan de hand van het sleutelwoord **color** en verwacht 3 RGB-waarden. Om met meerdere verschillende soorten spectra te werken, volstaat dat niet. Daarom hebben we het invoerformaat lichtjes uitgebreid. Alle wijzigingen zijn gedetailleerd beschreven in appendix A. Kort samengevat komt het erop neer dat we aan de kleurspecificatie volgende mogelijkheden hebben toegevoegd:

- **samplespectrum**: bemonsterd spectrum,
- **tabulatedspectrum**: getabuleerd spectrum,
- **fourierspectrum**: Fourierspectrum,
- **compositespectrumsampled**: composietspectrum, met een bemonsterd spectrum als zacht verlopend gedeelte,
- **compositespectrumfourier**: composietspectrum, met een Fourierspectrum als zacht verlopend gedeelte,

- **rgbspectrum**: RGB-spectrum,
- **rgbfromsampled**: RGB-spectrum afgeleid van een regulier bemonsterd spectrum,
- **rgbfromtabulated**: RGB-spectrum afgeleid van een getabuleerd spectrum.

De oorspronkelijke **color** wordt omgezet naar een bemonsterd spectrum via de methode van Smits (2.6.2 p. 24). **rgbfromsampled** is hiervan het omgekeerde. Met **rgbspectrum** kan een RGB-spectrum opgegeven worden dat als RGB-spectrum behouden blijft.

Dit hoofdstuk is een korte inleiding over de fysische eigenschappen van elektromagnetische golven, zoals licht. Voor een ruimere inleiding wordt verwezen naar (Huyse 2001) en voor een meer theoretische uitdieping naar (Born en Wolf 1999).

Er zijn verschillende niveaus van nauwkeurigheid om de wetten van de optica te beschrijven:

- De geometrische optica is de eenvoudigste en wordt in de meeste grafische toepassingen gebruikt. De verklaring van vele fenomenen vloeit niet voort uit de theorie, maar de meeste effecten kunnen wel omschreven worden. De basisveronderstelling is dat licht als een stroom van deeltjes in een lichtbron ontstaat en altijd rechtdoor reist tot er een weerkaatsing of breking optreedt.
- De golfoptica, beschreven door de wetten van Maxwell, geeft wel een verklaring voor vele effecten, zoals interferentie, dispersie en polarisatie. In grafische toepassingen wordt het golfkarakter van licht zelden beschouwd, maar onze toepassing is er net op gebaseerd.
- De kwantumoptica is een unificatie tussen licht als deeltjes en als golven. Deze beschrijving van licht is enkel van theoretisch nut. Ze geeft enkel bijkomende verklaringen voor effecten die men beter op de andere niveaus kan simuleren.

Een lichtstraal is zoals gezegd een elektromagnetische golf die zich met een snelheid c voortplant. Het is een sinusoïdale golf met golflengte λ en frequentie ν :

$$c = \lambda\nu. \quad (4.1)$$

Meestal is c de lichtsnelheid in vacuüm of lucht: 299 792 458 m/s. De golflengte en de daarbij horende frequentie bepalen de kleur van het licht. Andere

belangrijke grootheden zijn de periode T en de cirkelfrequentie of pulsatie ω :

$$T = \frac{1}{\nu}, \quad \omega = 2\pi\nu. \quad (4.2)$$

Het (circulaire) golfgetal k is van belang omdat het de fase van de golf uitdrukt in functie van de afgelegde weg:

$$k = \frac{2\pi}{\lambda}. \quad (4.3)$$

Vaak gebruikt men de golfvector of voortplantingsvector \vec{k} met lengte k in de voortplantingsrichting \hat{r} van de golf:

$$\vec{k} = k\hat{r}. \quad (4.4)$$

De golf in het elektrisch veld \vec{E} wordt verder besproken; de golf in het magnetisch veld \vec{B} kan analoog beschreven worden. De algemene vergelijking van een sinusoidale golf met amplitude \vec{E}_0 is:

$$\vec{E}(x, t) = \vec{E}_0 \sin(kx - \omega t), \quad (4.5)$$

waarbij $\vec{E}(x, t)$ de elektrische veldsterkte aangeeft na een weglengte x op tijd t .

Meestal werkt men met vlakke golven: (bijna) evenwijdige lichtbundels waarin iedere straal dezelfde weglengte aflegt. De maxima van de stralen vormen telkens een vlak golffront. De vergelijking voor de elektrische veldsterkte van een vlakke golf in de driedimensionale ruimte is:

$$\vec{E}(\vec{x}, t) = \vec{E}_0 \sin(\vec{k} \cdot \vec{x} - \omega t), \quad (4.6)$$

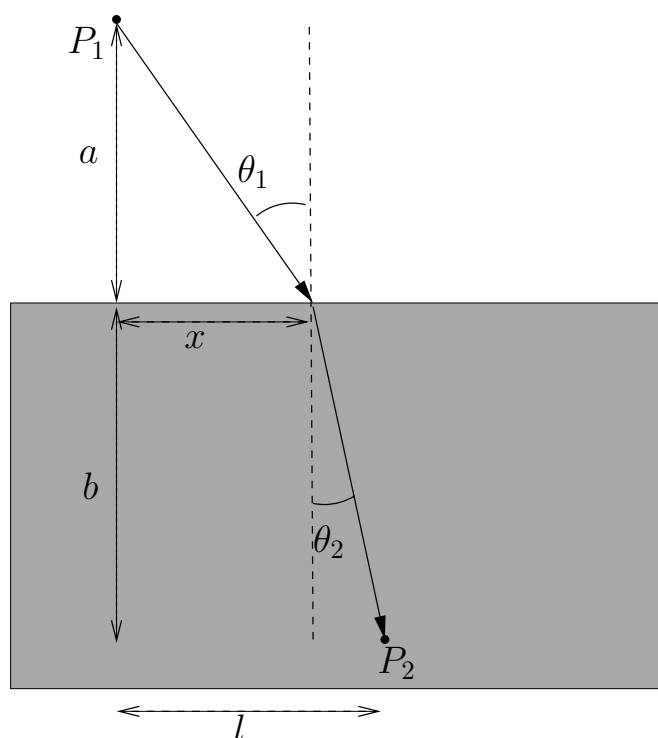
waarbij \vec{x} een punt in de ruimte is. Voor vele toepassingen is de exponentiële schrijfwijze vaak eenvoudiger:

$$\vec{E}(\vec{x}, t) = \vec{E}_0 \exp\left(i(\vec{k} \cdot \vec{x} - \omega t)\right). \quad (4.7)$$

Deze vorm is triviaal af te leiden via de formule van Euler.

Gewoonlijk houdt men geen rekening met het tijdsgebonden aspect van lichtgolven. Door de vergelijkingen uit te middelen over de tijd t zal de term ωt overal verdwijnen. De formule die het vaakst zal gebruikt worden is dan:

$$\vec{E}(\vec{x}) = \vec{E}_0 \exp\left(i(\vec{k} \cdot \vec{x})\right). \quad (4.8)$$



Figuur 4.1: Breking van een straal aan de rand van een medium.

4.1 Breking

Volgens het principe van Fermat zal een lichtstraal tussen twee punten steeds de weg kiezen die het minste tijd vraagt. In vacuüm of in één en hetzelfde (homogene) medium gaat een lichtstraal dus steeds via de rechte tussen die twee punten. Wat er gebeurt bij de grens van twee verschillende media, is interessanter. Dit wordt nu verder uitgewerkt (zie ook figuur 4.1).

Zowel in het eerste als in het tweede medium zal het licht zich volgens een rechte voortplanten. Aan de grens tussen de twee media kan het licht eventueel gebroken worden. Nemen we nu θ_1 en θ_2 de hoek van de lichtstraal met de normaal in medium 1 en medium 2, dan is de totale tijd t die de lichtstraal onderweg is tussen punt 1 en 2 gelijk aan $t = d_1/v_1(\lambda) + d_2/v_2(\lambda)$ met $v_1(\lambda)$ de lichtsnelheid voor licht met golflengte λ in medium 1, d_1 de afgelegde weg in medium 1, $v_2(\lambda)$ de lichtsnelheid voor licht met golflengte λ in medium 2 en d_2 de afgelegde weg in medium 2.

Met behulp van Pythagoras kan men d_1 en d_2 schrijven als respectievelijk $\sqrt{a^2 + x^2}$ en $\sqrt{b^2 + (l - x)^2}$. Ingevuld in de vergelijking voor de totale tijd,

verkrijgt men dat

$$t = \frac{\sqrt{a^2 + x^2}}{v_1(\lambda)} + \frac{\sqrt{b^2 + (l-x)^2}}{v_2(\lambda)} \quad (4.9)$$

De afstand x kan nu bepaald worden waarbij de tijd minimaal wordt, of dus

$$\begin{aligned} \frac{dt}{dx} = 0 &\Leftrightarrow \frac{x}{v_1(\lambda)\sqrt{a^2 + x^2}} = \frac{l-x}{v_2(\lambda)\sqrt{b^2 + (l-x)^2}} \\ &\Leftrightarrow \left(v_1(\lambda)\sqrt{1 + \left(\frac{a}{x}\right)^2} \right)^{-1} = \left(v_2(\lambda)\sqrt{1 + \left(\frac{b}{l-x}\right)^2} \right)^{-1}. \end{aligned} \quad (4.10)$$

Gebruiken we vervolgens dat $\tan \theta_1 = x/a$ en $\tan \theta_2 = (l-x)/b$ in (4.10), dan krijgt men als resultaat dat

$$\left(v_1(\lambda)\sqrt{1 + \frac{1}{\tan^2 \theta_1}} \right)^{-1} = \left(v_2(\lambda)\sqrt{1 + \frac{1}{\tan^2 \theta_2}} \right)^{-1} \quad (4.11)$$

en vermits $\sqrt{1 + 1/\tan^2 x} = 1/\sin x$,

$$\frac{\sin \theta_1}{v_1(\lambda)} = \frac{\sin \theta_2}{v_2(\lambda)}. \quad (4.12)$$

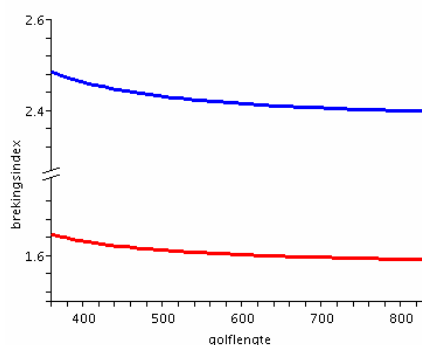
Definiëren we nu de brekingsindex $n(\lambda)$ van een materiaal als $n(\lambda) = c/v(\lambda)$ met c de lichtsnelheid in vacuüm, dan bekomt men de bekende wet van Snellius:

$$\frac{\sin \theta_1}{\sin \theta_2} = \frac{n_2(\lambda)}{n_1(\lambda)}. \quad (4.13)$$

Merk op dat de lichtsnelheid in een medium afhankelijk is van de golflengte van het beschouwde licht. Omdat blauw licht bijvoorbeeld trager door glas gaat dan rood licht, wordt blauw licht ook onder een iets andere hoek gebroken als rood licht (namelijk dichter naar de normaal toe). Om deze hoeken te berekenen, volstaat het te weten hoe de lichtsnelheid of de brekingsindex van een materiaal verandert met de golflengte. Deze relatie wordt vaak beschreven door middel van de Sellmeier-vergelijking, die verder zal gebruikt worden. De Sellmeier-vergelijking is een empirische formule, waarvan de coëfficiënten experimenteel bepaald worden:

$$n(\lambda) = \sqrt{1 + \frac{B_1\lambda^2}{\lambda^2 - C_1} + \frac{B_2\lambda^2}{\lambda^2 - C_2} + \frac{B_3\lambda^2}{\lambda^2 - C_3}} \quad (4.14)$$

De coëfficiënten B_1 , B_2 , B_3 , C_1 , C_2 en C_3 kunnen voor glas zeer vaak teruggevonden worden in de catalogus van de producent ervan. In figuur 4.2 is een voorbeeld van $n(\lambda)$ afgebeeld voor glas en diamant.



Figuur 4.2: Variatie van de brekingsindex van glas (onder) en diamant (boven) in functie van de golflengte.

4.2 Interferentie

Verschillende golven in hetzelfde punt in de ruimte kunnen gesuperponeerd worden. De vectoren mogen gewoon opgeteld worden. Vaak kan men veronderstellen dat deze golven dezelfde amplitude en frequentie hebben, maar een verschillende weglengte hebben afgelegd. Dit betekent dat er een nieuwe sinusoïdale golf ontstaat met dezelfde frequentie, maar een andere amplitude:

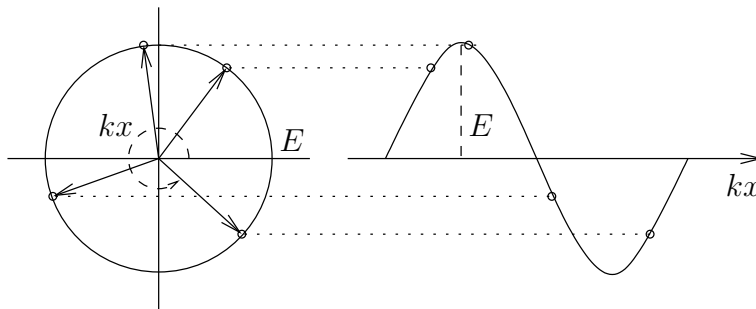
$$\begin{aligned} \vec{E}_1(x, t) + \vec{E}_2(x, t) &= \vec{E}_0 \sin(kx_1 - \omega t) + \vec{E}_0 \sin(kx_2 - \omega t) \\ &= \underbrace{2\vec{E}_0 \cos\left(k \frac{x_1 - x_2}{2}\right)}_{\text{nieuwe amplitude}} \cdot \sin\left(k \frac{x_1 + x_2}{2} - \omega t\right). \end{aligned} \quad (4.15)$$

Merk op dat de nieuwe amplitude groter of kleiner kan zijn dan de oorspronkelijke amplitude \vec{E}_0 , afhankelijk van het verschil in weglengte. Dit verschijnsel wordt interferentie genoemd. Als het verschil in weglengte een oneven aantal halve golflengten bedraagt (of het faseverschil een oneven aantal keer π), is er ongunstige of negatieve of destructieve interferentie. Als het verschil een geheel aantal golflengten bedraagt (of het faseverschil een even aantal keer π), spreekt men van gunstige of positieve of constructieve interferentie.

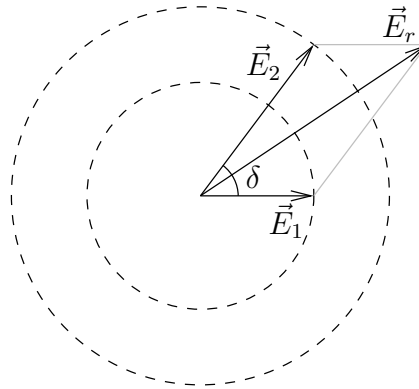
$$k(x_1 - x_2) = (2n + 1)\pi \quad \text{of} \quad x_1 - x_2 = (2n + 1)\frac{\lambda}{2} \quad (\text{destructief}) \quad (4.16)$$

$$k(x_1 - x_2) = 2n\pi \quad \text{of} \quad x_1 - x_2 = n\lambda \quad (\text{constructief}) \quad (4.17)$$

Een grafische manier om de interferentie van twee golven te berekenen is de fasorvoorstelling. Een fasor is een vector met als lengte de amplitude van



Figuur 4.3: Fasordiagram van een golf.



Figuur 4.4: Resulterende golf van interferentie, grafisch afgeleid.

een golf. De fasor van een golf wordt in een tweedimensionaal assenstelsel getekend, vertrekkende in de oorsprong en in een willekeurige richting. Als men de fasor laat ronddraaien rond de oorsprong en ondertussen het andere uiteinde projecteert op een as, kan de golfvorm terug worden afgeleid (figuur 4.3).

De fasor van de tweede golf wordt op hetzelfde diagram getekend zodat de hoek δ tussen de fasoren gelijk is aan het faseverschil (figuur 4.4). De som van de golven is dan gewoon de vectorsom. Men kan grafisch de amplitude en de fase aflezen.

Spectraal lichttransport

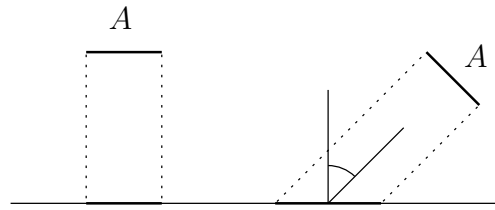
Lichttransport, de manier waarop licht zich voortplant en interageert met materialen en media, kan op verschillende manieren gesimuleerd worden. Eén van de algoritmes, en veruit het belangrijkste, is *ray tracing*.¹ Het ligt aan de basis van een grote familie van rendering algoritmes.

Zoals de naam al doet vermoeden, is ray tracing niet meer dan het volgen van lichtstralen door een driedimensionale ruimte. Licht heeft de handige eigenschap dat het in vele omstandigheden gewoon rechtdoor gaat, tot het op een voorwerp botst. Deze eigenschap werd door schilders in de renaissance al gebruikt om realistisch in perspectief te tekenen en om correcte schaduwen te werpen. Lenzenmakers in de 19de eeuw gebruikten een wiskundige vorm van ray tracing om de convergentie van lenzen te berekenen. Met een computer kunnen die berekeningen veel sneller uitgevoerd worden, zodat het haalbaar wordt om vele miljoenen stralen te volgen doorheen een scène.

De belangrijkste algoritmes die geïmplementeerd zijn in PBRT worden even toegelicht. Voor een grondige beschrijving verwijzen we naar het bijbehorende boek *Physically Based Rendering (PBR)* (Pharr en Humphreys 2003).

De meeste publicaties vermelden in een inleidend hoofdstuk dat men eigenlijk spectra zou moeten gebruiken. Men kondigt dan aan dat in de rest van de tekst enkel over RGB zal gesproken worden en dat de lezer er iedere keer aan moet denken dat er spectra hadden moeten staan. Wij zullen in ieder onderdeel van ray tracing duidelijk aangeven waar er spectra opduiken.

¹Hierover is discussie mogelijk. In real time rendering wordt er nog veel gewerkt met het scanline algoritme, omdat het efficiënt in hardware kan geïmplementeerd worden. Ray tracing begint ook in deze markt aan populariteit te winnen door de ruimere mogelijkheden die deze aanpak biedt.



Figuur 5.1: Belang van de invallende hoek bij irradiantie.

5.1 Radiometrische grootheden

In de verdere bespreking worden een aantal fysische grootheden i.v.m. de lichtsterkte gebruikt. Al deze grootheden variëren over de golflengte λ .

De (stralings-)flux of het vermogen $\Phi(\lambda)$, uitgedrukt in joule per seconde (J/s) of watt (W), duidt de energiestroom per tijdseenheid van een lichtbron aan:

$$\Phi(\lambda) = \frac{dQ(\lambda)}{dt}. \quad (5.1)$$

Een gloeilamp van 60 W met een rendement van 5% heeft een flux van 3 W. Het grootste deel van die flux bevindt zich in het gele en rode licht.

De irradiantie $E(\lambda)$ is de inkomende flux per eenheid van oppervlakte, uitgedrukt in W/m^2 :

$$E(\lambda) = \frac{d\Phi(\lambda)}{dA}. \quad (5.2)$$

De irradiantie is afhankelijk van de hoek waaronder het licht invalt op het oppervlak. Als het licht loodrecht invalt, is de irradiantie groter dan wanneer het schuin invalt. Dat komt omdat dezelfde flux uitgesmeerd wordt over een groter oppervlak (figuur 5.1).

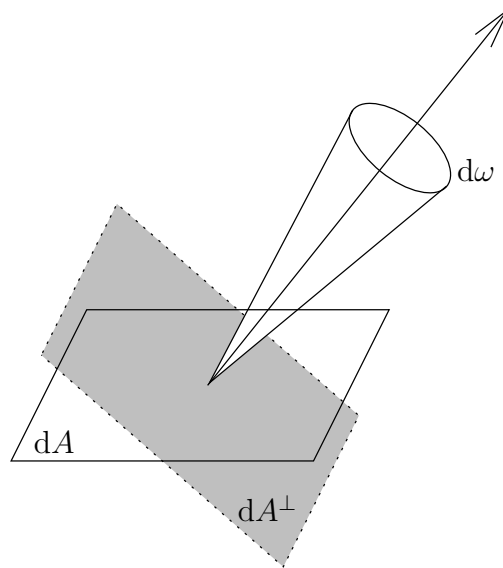
De exitantie $M(\lambda)$ of radiositeit $B(\lambda)$ is hetzelfde als de irradiantie, maar dan voor het uitgaande licht. Het is de uitgestraalde energie per oppervlak, bijvoorbeeld voor een oppervlaktelichtbron.

De intensiteit $I(\lambda)$ is de flux per ruimtehoek. De ruimtehoek is de oppervlakte op de eenheidsbol rond het beschouwde punt. De eenheid is watt per steradiaal (W/sr).

$$I(\lambda) = \frac{d\Phi(\lambda)}{d\omega} \quad (5.3)$$

Deze grootheid kan de sterkte van een puntlichtbron aanduiden.

De belangrijkste grootheid in ray tracing is de radiantie $L(\lambda)$. Het is de



Figuur 5.2: Radiantie.

flux per oppervlakte en per ruimtehoek, uitgedrukt in $\text{W}/\text{m}^2 \text{sr}$:

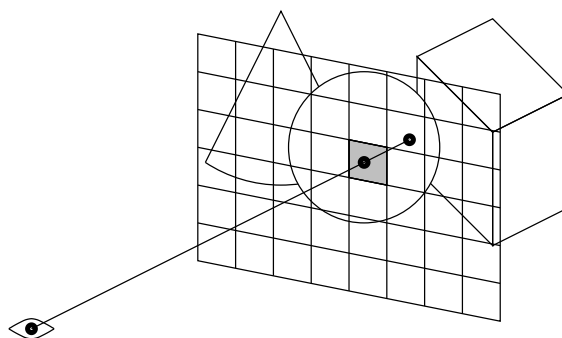
$$\begin{aligned} L(\lambda) &= \frac{d^2\Phi(\lambda)}{d\omega dA^\perp} \\ &= \frac{d^2\Phi(\lambda)}{d\omega dA \cos\theta}. \end{aligned} \quad (5.4)$$

De oppervlakte dA^\perp is de oppervlakte dA geprojecteerd loodrecht op de richting ω (figuur 5.2).

Radiantie is de belangrijkste radiometrische grootte om vele redenen:

- Alle andere grootheden kunnen eruit afgeleid worden door een eenvoudige integratie.
- Radiantie is invariant langs een straal in vacuüm of zuivere lucht. Dit is bijzonder handig in ray tracing omdat er enkel bij de oppervlakken iets kan veranderen aan de radiantie.
- Het menselijk oog is gevoelig voor de luminantie of helderheid L_v . Die kan afgeleid worden uit de radiantie door een gewogen integratie met de color matching functie $\bar{y}(\lambda)$ uit het CIE XYZ kleurmodel (figuur 2.7):

$$L_v = K_m \int L(\lambda) \bar{y}(\lambda) d\lambda \quad \text{met } K_m = 683 \text{ lm/W}. \quad (5.5)$$



Figuur 5.3: Basisalgoritme van ray tracing.

5.2 Klassieke ray tracing

De kern van ieder ray tracing programma is een implementatie van het basisalgoritme dat door Whitted werd gepubliceerd (Whitted 1980). Een goed inleidend werk dat ook kort enkele geavanceerde onderwerpen aanraakt, is (Shirley en Morley 2003).

Figuur 5.3 toont de basisopstelling. Er wordt een rooster van pixels tussen het oog (of de camera) en de scène geplaatst en er wordt een straal getrokken van het oog door een pixel. De kleur die men door die pixel ziet, is de kleur van het eerste voorwerp in de scène waar de straal op invalt, en die kleur wordt bepaald door het invallende licht in dat punt.

De stralen hebben de vorm:

$$\vec{r}(t) = \vec{o} + t\hat{d}. \quad (5.6)$$

Ieder punt $\vec{r}(t)$ ligt op de straal die vertrekt in \vec{o} (de camera) en loopt in de richting \hat{d} . De parameter t is positief.

Merk op dat de straal in de tegengestelde zin van het licht wordt gevolgd: vanuit het oog naar de scène en uiteindelijk naar een lichtbron.

5.3 Intersectie

De kleur die men in een pixel ziet, is bepaald door het eerste voorwerp waarop de straal in de scène botst. Die voorwerpen zijn vaak gedefiniëerd door een impliciete functie:

$$F(\vec{x}) = 0. \quad (5.7)$$

Als men daarin de vergelijking van de straal (5.6) substitueert, kunnen de positieve reële oplossingen voor t gevonden worden:

$$\begin{aligned} F(\vec{r}(t)) &= 0 \\ F(\vec{o} + t\hat{d}) &= 0 \quad \text{met } t \in [0, \infty) \end{aligned} \tag{5.8}$$

Vaak is dit een vergelijking van hoge graad (kwadratisch, kubisch, vierdegraads of zelfs hoger) en kan de berekening van de intersectie enig rekenwerk vergen.

5.4 Versnellingsstructuren

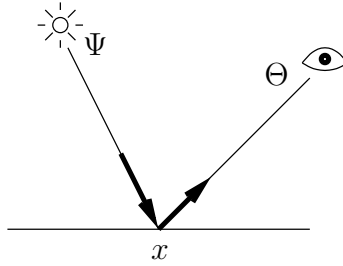
Als er veel voorwerpen aanwezig zijn in de scène, moet in principe deze intersectievergelijking opgelost worden voor ieder voorwerp en voor iedere straal door een pixel. Dat zou een complexiteit $\mathcal{O}(IN)$ opleveren met I het aantal pixels en N het aantal voorwerpen. Bovendien zijn de meeste complexe voorwerpen opgebouwd uit een grote *mesh* van aaneensluitende driehoeken, waarvan elke driehoek apart moet onderzocht worden op snijpunten.

Als men alle voorwerpen en driehoeken omvat in volumes met een heel eenvoudige impliciete functie F , dan kan men snel testen of er kans is op een snijpunt. Dit noemt men een omhullend volume (*bounding volume*), en praktisch gebruikt men meestal een omhullende balk (*bounding box*). Als bovendien verschillende naburige omhullende volumes omvat worden in grotere omhullende volumes, krijgt men een hiërarchie van volumes, concreet een *kd-boom*. In zo'n boom kunnen heel efficiënt de volumes gesnoeid worden die zeker geen snijpunt hebben met een gegeven straal. Vaak beperkt men zo de complexiteit tot $\mathcal{O}(I \log N)$.

Het spreekt voor zich dat de snelheid van een ray tracer in grote mate zal afhangen van de efficiëntie van de intersecties en de versnellingsstructuur. Het snelheidsverschil tussen 3 RGB-waarden en een volledig spectrale behandeling is verwaarloosbaar, omdat er relatief weinig kleurberekeningen gebeuren t.o.v. intersecties. De complexiteit zal $\mathcal{O}(I(\log N + S))$ worden voor een bemonsterd spectrum met S waarden. S is typisch veel kleiner dan I of $\log N$.

5.5 Zichtbaarheid

Het gevonden snijpunt kan in de schaduw van een lichtbron liggen, zodat de kleurbijdrage van die lichtbron zwart is. Om dit te weten te komen volgt



Figuur 5.4: De bidirectionele reflectantie distributie functie.

men een straal van het punt naar de lichtbron, en men controleert of er een ander voorwerp tussen ligt. Deze straal noemt men een schaduwstraal. De snijpunten van een schaduwstraal kunnen efficiënter berekend worden omdat de positie van het dichtste snijpunt niet belangrijk is, enkel het feit of er minstens één snijpunt is.

5.6 Kleuring

Zodra het snijpunt het dichtst bij de camera gevonden is, moet men de kleur van dat punt (en van de pixel) bepalen. Daarvoor gebruikt men al het licht dat rechtstreeks of onrechtstreeks invalt op het punt in kwestie. Men noemt dit de lokale belichting.

De uiteindelijke kleur van het punt zal ook afhangen van de kleur en de aard van het materiaal. Hiervoor wordt een bidirectionele reflectantie distributie functie (BRDF) gebruikt. Dat is een materiaalmodel dat o.a. rekening kan houden met de ruwheid en de glans van een materiaal.

De BRDF f_r wordt gedefinieerd als:

$$\begin{aligned}
 f_r(x, \Psi \leftrightarrow \Theta, \lambda_{\text{in}}, \lambda_{\text{uit}}) &= \frac{dL(x \rightarrow \Theta, \lambda_{\text{uit}})}{dE(x \leftarrow \Psi, \lambda_{\text{in}})} \\
 &= \frac{d^2L(x \rightarrow \Theta, \lambda_{\text{uit}})}{L(x \leftarrow \Psi, \lambda_{\text{in}}) \cos(\hat{n}_x, \Psi) d\omega_\Psi d\lambda_{\text{in}}}.
 \end{aligned} \tag{5.9}$$

In woorden: de BRDF is de verhouding tussen de uitgaande radiantie en de inkomende irradiantie (figuur 5.4). Als er geen uitzonderlijke effecten zoals fluorescentie of fosforescentie optreden, dan is λ_{in} gelijk aan λ_{uit} .

Ook de naam verdient wat uitleg:

bidirectioneel: of wederkerig. De waarde verandert niet als de straal in de omgekeerde richting wordt gevolgd. Dit wordt aangeduid door de

dubbele pijl tussen de richtingen Ψ en Θ . Dit wordt ook de Helmholtz-wederkerigheid genoemd.

reflectantie: De waarde is, op een normalisatie na, gelijk aan de reflectantie ρ .

distributie-functie: Deze functie bepaalt de energieverdeling van het uitgaande licht over de hemisfeer. Die is steeds positief, en ze integreert tot 1 als het inkomende licht niet in het materiaal geabsorbeerd wordt.

Er zijn enkele typische voorbeelden van veel gebruikte BRDF's. De eenvoudigste is een Lambertiaans diffuus (mat) materiaal (figuur 5.6(a) en 5.7(a)):

$$f_{r,\text{diffuus}}(x, \lambda) = \frac{\rho_{\text{diffuus}}(x, \lambda)}{\pi}, \quad (5.10)$$

constant over Ψ en Θ . Vaak hebben alle punten van een oppervlak dezelfde kleur, waardoor de formule nog vereenvoudigt tot:

$$f_{r,\text{diffuus}}(\lambda) = \frac{\rho_{\text{diffuus}}(\lambda)}{\pi}. \quad (5.11)$$

Een andere bekende BRDF is de aangepaste Blinn-Phong-BRDF (figuur 5.6(b) en 5.7(b)), gebaseerd op de Phong-kleuring (Phong 1975, Blinn 1977):

$$f_{r,\text{Phong}}(x, \Psi \leftrightarrow \Theta, \lambda) = \frac{\rho_{\text{speculair}}(n+2)}{2\pi} \cos^n(\hat{n}_x, \hat{\omega}_h) + f_{r,\text{diffuus}}(\lambda). \quad (5.12)$$

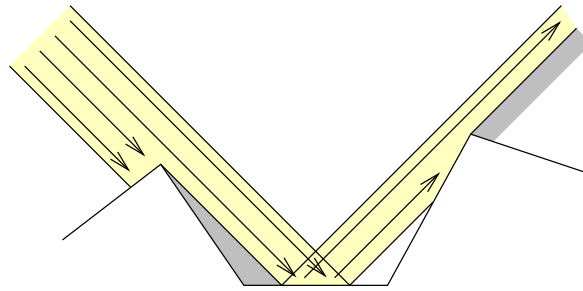
De eenheidsvector $\hat{\omega}_h$ is de *halve-hoek-vector*, de bissectrice van de richtingen Ψ en Θ :

$$\hat{\omega}_h = \frac{\Psi + \Theta}{\|\Psi + \Theta\|}. \quad (5.13)$$

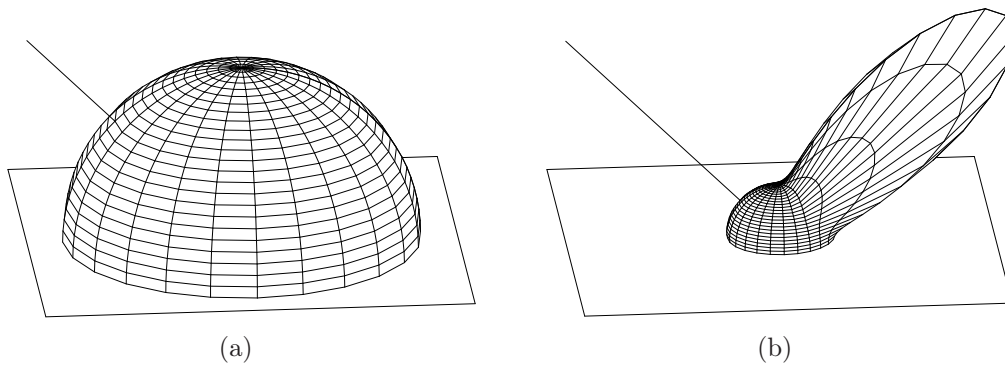
Dit model is gebaseerd op een oppervlak dat macroscopisch wel glad en glanzend is, maar microscopisch een ruwe structuur van microfacetten heeft (figuur 5.5).

Andere interessante BRDF's zijn o.a. Cook-Torrance (Cook en Torrance 1981), het anisotrope model van Ward (Ward 1992), en de eenvoudig op meetwaarden te fitten BRDF van Lafortune (Lafortune *et al.* 1997).

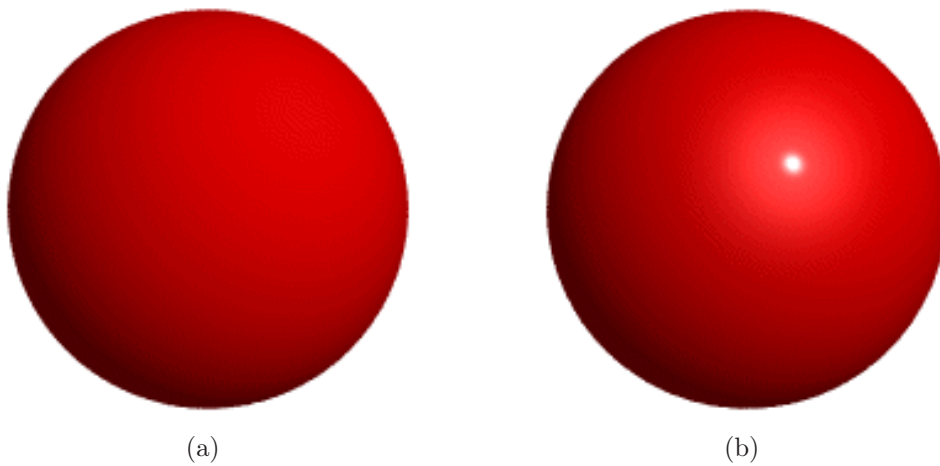
Voor doorschijnende materialen is er een analoog concept voor het indringende licht: de bidirectionele transmittantie distributie functie (BTDF). De BRDF en BTDF vormen samen de bidirectionele *scattering* distributie functie (BSDF).



Figuur 5.5: Een lichtbundel weerkaatst aan een oppervlak met een microfacetstructuur.



Figuur 5.6: De BRDF van (a) een mat Lambertiaans materiaal en (b) een glanzend Phong-materiaal.



Figuur 5.7: (a) Een Lambertiaans diffuse bol; (b) het typisch plastic-achtige uitzicht van Phong-kleuring.



Figuur 5.8: *Subsurface scattering* in de huid van het gelaat (Jensen *et al.* 2001): (a) een BRDF geeft een harde, onnatuurlijke indruk; (b) een BSSRDF geeft overtuigend de zachte schaduw rond de neus weer (afbeeldingen © Henrik Wann Jensen, model © Steven Stahlberg).

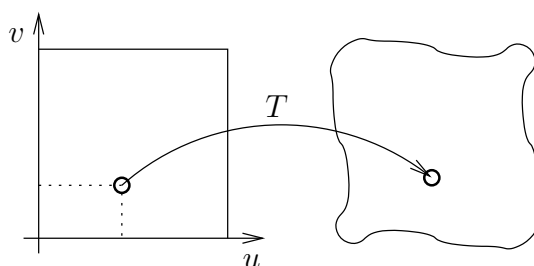
In sommige complexe materialen kan het licht een klein stukje in het oppervlak indringen en er wat verder weer uit ontsnappen. Ze zijn doorschijnend, maar niet doorzichtig², zoals bijvoorbeeld marmer, melk, huid, ... Voor zulke materialen bestaat er een bidirectionele *scattering surface* reflectantie distributie functie (BSSRDF) (figuur 5.8).

5.7 Textuur

Een BRDF is meestal onafhankelijk van het beschouwde punt x . De kleur en de microstructuur verandert niet over het oppervlak. Een voorwerp dat gekleurd wordt aan de hand van een BRDF zal dus meestal een effen kleur hebben. Dat is natuurlijk niet aanvaardbaar voor vele realistische materialen en voorwerpen. Een materiaal als hout heeft duidelijke kleurverschillen in de structuur van de jaarringen. Voorwerpen kunnen geschilderd of bedrukt zijn in verschillende kleuren.

Om zulke voorwerpen te kunnen weergeven zonder daarvoor verschillende gebieden op het oppervlak een andere BRDF te geven, gebruikt men texturen (Catmull 1974). Dat zijn afbeeldingen die als het ware op het voorwerp geplakt worden (figuur 5.10(a)). Men moet natuurlijk rekening houden met

²Er is een belangrijk onderscheid tussen “doorschijnend” en “doorzichtig”. Men kan door een doorzichtig materiaal zien wat zich erachter bevindt, omdat er voldoende licht zonder verstrooiing door kan. Een doorschijnend voorwerp laat enkel verstrooid licht door, zodat er geen beeld kan gevormd worden van wat zich erachter bevindt.

Figuur 5.9: De textuurprojectie T .

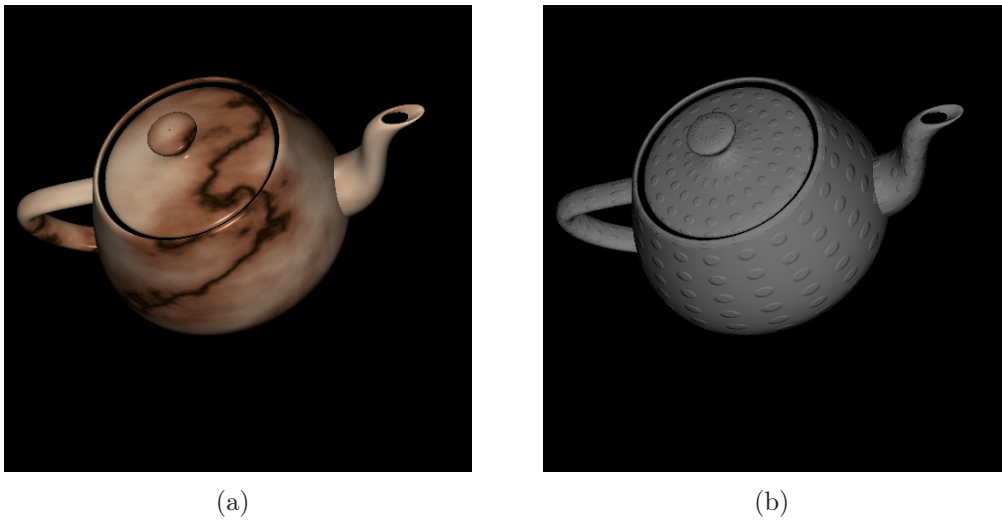
vervormingen als men een vlakke afbeelding op een gekromd voorwerp plakt. Er zijn verschillende textuurprojecties T mogelijk om de vervormingen te beperken. Door die projectie worden de pixelcoördinaten van de afbeelding getransformeerd naar een parametrisatie van het oppervlak. Dit noemt men de textuurcoördinaten (figuur 5.9).

Voor sommige materialen is een afbeelding niet aangewezen. Een dambordpatroon kan eenvoudig gegenereerd worden met een procedurele textuurfunctie afhankelijk van de textuurcoördinaten. Vele oppervlakken zoals marmer kunnen gesimuleerd worden met een ruisfunctie.

Er bestaan ook driedimensionale texturen die voor ieder driedimensionaal punt een kleur bepalen. Op die manier ziet een voorwerp eruit alsof het uit een echt blok hout of marmer gebeeldhouwd is.

Met een tweedimensionale functie of afbeelding kan men ook nog andere materiaaleigenschappen modelleren. Het is bijvoorbeeld mogelijk om de mesostructuur van een oppervlak uit te drukken. Dat is een structuurniveau dat zich tussen de macrostructuur (geometrie) en microstructuur (microfacetten) bevindt. De bekendste toepassing is *bump mapping* (figuur 5.10(b)), uitgevonden door Blinn (Blinn 1978). Men gebruikt voor de kleuring een normaal die een kleine afwijking heeft ten opzichte van de echte geometrische normaal. De afwijking wordt gegeven door een textuur. Door deze afwijkende normaal geeft de kleuring de indruk van bulten en deuken in het oppervlak, hoewel de complexiteit van de geometrie niet verandert.

PBRT veralgemeent het gebruik van texturen voor alle materiaalparameters die zouden kunnen veranderen over een oppervlak, bijvoorbeeld de ruwheid of transparantie.



Figuur 5.10: De Utah theepot, één van de populairste voorwerpen om te renderen: (a) met een textuur van marmer; (b) met een *bump map* (© Michael Bax).

5.8 Recursie

Als het materiaal glanzend of doorschijnend is, moet ook het weerkaatste of gebroken licht in rekening gebracht worden. Daarvoor trekt men nieuwe stralen alsof de camera in het punt op het oppervlak staat, en in de richting van de reflectie of refractie.

De kleur van een punt wordt in het basisalgoritme van Whitted traditioneel aangeduid met de volgende formule:

$$\begin{aligned}
 I(\lambda) = \sum_{\text{lichtbronnen}} I_{\text{bron}}(\lambda) & (\rho_{\text{diffuus}}(\lambda) \cos \theta + \rho_{\text{speculair}}(\lambda) \cos^n \alpha) \\
 & + I_{\text{reflectie}}(\lambda) \rho_{\text{reflectie}}(\lambda) + I_{\text{transmissie}}(\lambda) \rho_{\text{transmissie}}(\lambda) \\
 & + I_{\text{ambiënt}}(\lambda) \rho_{\text{ambiënt}}(\lambda).
 \end{aligned} \tag{5.14}$$

De verschillende componenten zijn:

diffuus voor matte oppervlakken, afhankelijk van de hoek θ waaronder het licht invalt.

speculair voor een glanzende indruk, met een exponent n die aanduidt hoe goed het oppervlak spiegelt, en afhankelijk van de hoek α tussen de perfect gereflecteerde richting en de beschouwde richting. Dit is de Phong-kleuring die al eerder vermeld werd.

reflectie voor spiegellende oppervlakken, berekend met een recursieve straal.

transmissie voor doorschijnende oppervlakken, berekend met een recursieve straal.

ambiënt, een lapmiddel voor het lichttransport dat niet beschouwd werd, een virtuele lichtbron die overal in de scène een constante intensiteit uitstraalt.

Formule 5.14 is een recursieve som. $I_{\text{reflectie}}$ en $I_{\text{transmissie}}$ worden zelf ook berekend door die formule.

Het eenvoudige concept van stralen kan dus op heel wat manieren hergebruikt worden. Vele effecten die in andere algoritmes heel wat werk vragen, komen in ray tracing bijna als vanzelf. Typische afbeeldingen van ray tracing trekken de aandacht met metalen en glazen bollen, iets wat in andere algoritmes vaak moeilijk of onmogelijk is.

5.9 Globale belichting

Klassieke ray tracing beschouwt maar een stuk van het lichttransport in een scène, namelijk het licht dat langs de gevolgde stralen reist. Waarom zou de recursie op een glanzend voorwerp moeten doorgaan en op een mat voorwerp stoppen? De natuur maakt dit onderscheid niet. In de natuur blijft het licht botsen tot het geabsorbeerd wordt.

Sinds de late jaren 1980 is men op zoek naar methoden om het volledige lichttransport in een scène te berekenen. Dat wil zeggen, niet alleen het licht dat langs de traditioneel getrokken stralen reist, maar al het licht dat langs alle mogelijke stralen door de scène heen en weer botst. Deze aanpak noemt men globale belichting. Op deze manier kunnen alle mogelijke lichteffecten, indirecte belichting en zachte schaduwranden gevisualiseerd worden (figuur 5.11). Voor een goed overzicht van globale belichtingsalgoritmes raden we (Dutr e *et al.* 2003) aan.

Om het volledige lichttransport te simuleren, moet men de fenomenologische formule 5.14 vervangen door de *rendering vergelijking*. Kajiya was de eerste die deze vergelijking opstelde (Kajiya 1986). Een mogelijke formulering is:

$$L(x \rightarrow \Theta, \lambda_{\text{uit}}) = L_e(x \rightarrow \Theta, \lambda_{\text{uit}}) + \iint_{\Omega_x} L(x \leftarrow \Psi, \lambda_{\text{in}}) f_r(x, \Psi \leftrightarrow \Theta, \lambda_{\text{in}}, \lambda_{\text{uit}}) \cos(\vec{n}_x, \Psi) d\omega_\Psi d\lambda_{\text{in}}. \quad (5.15)$$



(a)



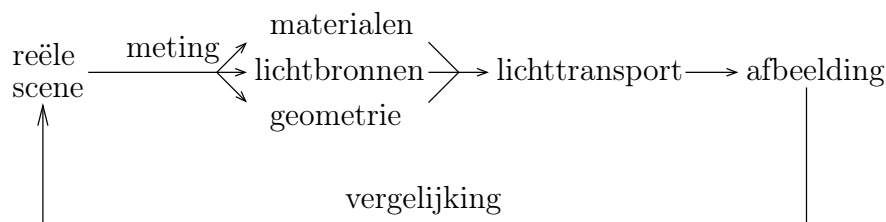
(b) afbeelding © Henrik Wann Jensen, model © Christophe Valtin

Figuur 5.11: (a) De *Cornell box* testscène met zachte schaduwen en *color bleeding* (de kleur van de muren loopt als het ware over in het plafond en de zijwanden van de dozen); (b) een scène die grotendeels met indirecte belichting is verlicht.

Deze formule beschrijft de radiantie met golflengte λ_{uit} die in het punt x vertrekt in de richting Θ . Die bestaat enerzijds uit de zelf uitgestraalde radiantie van het materiaal op punt x , in de richting Θ en met golflengte λ_{uit} . Anderzijds is er ook de radiantie met golflengte λ_{in} die in het punt x aankomt uit alle mogelijke richtingen Ψ . Die lichtenergie interageert in x met het materiaal, wat uitgedrukt wordt door de BRDF f_r . Enkel de radiantie die in de richting Θ weerkaatst met golflengte λ_{uit} draagt bij tot de berekende radiantie. Als uitzonderlijke fenomenen zoals fluorescentie en fosforescentie buiten beschouwing mogen blijven, dan zal de BRDF de golflengte niet veranderen: $\lambda_{\text{in}} = \lambda_{\text{uit}}$.

Om deze recursieve integraal te berekenen, gebruikt men Monte Carlo integratie. Dit is niet de snelst convergerende integratiemethode, maar ze is wel algemeen toepasbaar op alle mogelijke vormen van integralen en ze is eenvoudig te implementeren. Verschillende oplossingsmethoden kunnen als volgt geclassificeerd worden:

objectgebaseerd: Men beschouwt het lichttransport tussen oppervlakken van een mesh i.p.v. tussen afzonderlijke punten. De belangrijkste voorbeelden hiervan zijn de radiositeitsalgoritmes. Die slaan de oplossing van het lichttransport op in de mesh van de scène om ze later vaak interactief te kunnen visualiseren.



Figuur 5.12: Iteratieve verbetering in fotorealistische rendering.

beeldgebaseerd: Men volgt naast de traditionele stralen ook stralen in willekeurige richtingen. Er kunnen ook stralen vanuit de lichtbron vertrekken, wat men achterwaartse ray tracing of *light tracing* noemt. Deze algoritmes zijn voorbeelden van *stochastische ray tracing*.

hybride: Men verdeelt het licht over de scène, vaak met een objectgebaseerde techniek. Daarna wordt de afbeelding gemaakt met een voorwaartse stochastische ray tracing die gebruik maakt van de lichtverdeling. Als voorbeeld zal *foton mapping* besproken worden.

Er is samen met globale belichting ook een groeiende interesse in fysisch gebaseerde rendering. Men simuleert de wetten van de fysica om realistische beelden te genereren. Daarbij is het van belang om de correcte formules en eenheden te respecteren. Het doel is een fotorealistische afbeelding, niet te onderscheiden van een echte foto. Door de afbeelding te vergelijken met de realiteit kan men de modellen en algoritmes valideren (figuur 5.12).

Als men zegt dat een afbeelding “niet te onderscheiden” is van de realiteit, dan bedoelt men dat de visuele indruk voor het menselijk oog dezelfde moet zijn. Dat houdt verband met perceptuele rendering, een domein dat zich eerder richt op de psychologie van de waarneming dan op de wiskundige perfectie. De zwakheden van het menselijk visueel systeem, zowel fysiologisch als psychologisch, worden uitgebuit om de rendering te verbeteren. Als men bijvoorbeeld weet dat de mens een grote tolerantie heeft voor een bepaald soort onnauwkeurigheid in een beeld, dan kan men minder tijd besteden aan het berekenen van dat aspect.

5.10 Stochastische ray tracing

Op een glanzend voorwerp is het eenvoudig te berekenen in welke richting de volgende straal moet lopen. Op een mat voorwerp weerkaatst het licht eigenlijk in alle richtingen. Natuurlijk kan men niet oneindig veel stralen

trekken, maar men kan wel een goede schatting maken van het licht in alle richtingen.

Men past Monte Carlo integratie toe door een beperkt aantal stralen te trekken in willekeurige richtingen. De integrand van de rendering vergelijking wordt op die manier herhaaldelijk berekend voor een willekeurige waarde in het integratiegebied (Ω_x , de richtingen rond het punt x). Daarvan neemt men het gemiddelde als schatting van de echte waarde van de integraal.

Al die stralen volgen kost enorm veel rekenwerk. Het aantal stralen dat men bij iedere botsing zal volgen, noemt men de vertakkingsfactor (*branching factor*). Men wil zeker niet bij iedere botsing een brede waaier van stralen volgen.

De stralen kunnen ook niet tot in het oneindige gevolgd worden bij hun reflecties. Om de recursie te stoppen wordt Russische roulette gebruikt. Net zoals bij het beruchte gokspel is er bij iedere botsing van de straal slechts een bepaalde kans α dat de recursie doorgaat. Een schatting van de bijdrage van de afgebroken recursie wordt toegevoegd met een factor $1/\alpha$.

Om de convergentie te versnellen, kan men de willekeurige waarden op een intelligente manier kiezen:

stratificatie: een goede spreiding van monsters binnen het domein van de integratie. Verschillende monsters in een *cluster* bij elkaar zorgen voor een vertekend gemiddelde en bijgevolg een grotere fout op het resultaat. Een mogelijke manier om een goede spreiding te bekomen, is het domein verdelen in deelgebieden en uit ieder deelgebied een monster nemen. Een andere manier is enkel monsters aanvaarden die op een minimum afstand van elkaar liggen.

belang (*importance sampling*): meer waarden in gebieden die een grote bijdrage leveren. Voor radiantie is de loodrecht invallende richting vaak belangrijker dan schuin invallende richtingen (zie vergelijking 5.4). Voor glanzende materialen zijn de richtingen nabij de perfecte reflectie belangrijk.

Als men werkt in het RGB kleurmodel, dan werkt men eigenlijk met drie discrete golflengtes. Als men spectraal werkt, moet de golflengte als een volledige extra dimensie beschouwd worden. Het kan interessant zijn om ook efficiënte bemonsteringstechnieken te bedenken voor de integratie over de golflengte. In een later hoofdstuk geven we een aanzet met een eenvoudige adaptieve techniek.

Lichttransport langs verschillende soorten paden

In de verschillende algoritmes kunnen telkens maar bepaalde soorten paden gevolgd worden. De notatie van paden bestaat uit een opeenvolging van letters, die eventueel als een reguliere uitdrukking worden samengenomen:

letter	gebeurtenis	uitdrukking	betekenis
L	lichtbron	$(k)^+$	één of meer gebeurtenissen k
E	oog (<i>eye</i>)	$(k)^*$	nul of meer gebeurtenissen k
S	speculaire reflectie	$(k)^?$	nul of één gebeurtenis k
D	diffuse reflectie	$(k k')$	een gebeurtenis k of k'

Klassieke ray tracing kan slechts een heel specifieke soort van paden volgen: $LD?S^*E$. Het volledige lichttransport is $L(S|D)^*E$.

Als men in stochastische ray tracing willekeurige paden doorheen de scène volgt, is de kans klein dat er toevallig een interessant pad van de camera naar een lichtbron gevonden wordt. Daarom wordt de directe belichting apart berekend door naast willekeurige stralen ook stralen expliciet naar de lichtbronnen te trekken ($LD(S|D)^*E$). Men moet goed opletten dat deze energie niet dubbel geteld wordt, want met de andere willekeurige paden zou men ook zo'n pad kunnen vinden.

Zoals vermeld bepaalt de vertakkingsfactor in grote mate de hoeveelheid rekenwerk. Men verkleint de vertakkingsfactor vaak zelfs tot 1. Een straal wordt dan bij iedere botsing maar in één richting voortgezet. Dit noemt men *path tracing*. Als men vanuit het oog vele stralen willekeurig door een pixel trekt, en telkens zo één pad laat volgen, lost men twee problemen tegelijk op:

1. Harde gekartelde randen van voorwerpen op de afbeelding (*aliasing*) worden zachte vloeiende lijnen.
2. De rekentijd voor het lichttransport blijft binnen de perken.

De mogelijke paden zijn in principe $L(S|D)^*E$, maar de convergentie voor zeldzame paden zoals brandvlekken ($LS+DS^*E$) is extreem traag. In de praktijk zijn de haalbare paden van de vorm $LD(S|D)^*E$.

Het is soms interessant om ook in omgekeerde richting te werken. Vertrekkende van de lichtbron volgt men het licht in de natuurlijke richting. Dit noemt men *light tracing*. De mogelijke paden zijn $L(S|D)^*DE$.

De kans om met path tracing een kleine lichtbron te bereiken, of om met light tracing de camera te bereiken, is natuurlijk miniem. Men kan een lichtpad en een zichtpad aan elkaar koppelen om betere resultaten te krijgen.

Dit noemt men bidirectionele path tracing. De verbindingsstraal kan echter enkel tussen twee diffuse oppervlakken lopen, dus de volledige paden zijn van de vorm $L(S|D)*DD(S|D)*E$.

5.11 Foton mapping

Als in bidirectionele path tracing telkens een nieuw lichtpad en zichtpad wordt berekend, doen men onnodig veel werk. De informatie uit de lichtpaden kan opgeslagen en hergebruikt worden voor ieder zichtpad. Dit noemt men foton mapping en wordt uitvoerig beschreven in (Jensen 2001).

Deze methode werkt in twee stappen:

1. Er worden lichtpaden berekend en voor iedere botsing op een diffuus oppervlak wordt er een *foton* (een hoeveelheid lichtenergie op het oppervlak) opgeslagen in een foton map.
2. Er worden zichtpaden berekend, gebruik makend van de foton map voor de indirecte belichting. Typisch gebruikt men hiervoor een aangepaste vorm van stochastische ray tracing.

De foton map is een datastructuur die noch objectgebaseerd noch beeldgebaseerd is. De fotonen worden opgeslagen op vrije driedimensionale coördinaten, ook al liggen ze allemaal op diffuse oppervlakken uit de scène. In dit opzicht verschilt de foton map fundamenteel van een radiositeitsmesh.

Er kunnen specifieke zeldzame lichtpaden apart behandeld worden. Deze methode is bijzonder geschikt voor lichteffecten zoals brandvlekken (*caustics*): licht dat door een lens op een oppervlak gebundeld wordt, bijna tot een brandpunt (figuur 5.13). Er worden extra veel lichtpaden van de vorm $LS+D$ getrokken, en de fotonen worden in een aparte caustic foton map opgeslagen.

De algemene foton map bevat informatie van paden van de vorm $L(S|D)*D$. Hierin is directe en indirecte belichting vervat. Merk op dat paden uit de caustic foton map aan een specifiekere vorm van deze notatie voldoen. Om de lichtenergie niet dubbel te tellen wordt de algemene foton map enkel gebruikt op plaatsen waar de caustic map geen informatie heeft.

In de visualisatiestap worden paden berekend van de vorm $LS*E$ en $DS*E$. Het gebruikte algoritme is een klassieke ray tracer die voor indirecte belichting nog één niveau stochastische ray tracing doet. Als men de padnotaties van de twee stappen achter elkaar zet, krijgt men inderdaad het volledige lichttransport $L(S|D)*E$, dat bovendien zeer efficiënt wordt berekend.



Figuur 5.13: Een brandvlek van licht dat door een glas cognac schijnt (© Henrik Wann Jensen).

5.12 Participerende media

Tot nu toe werd altijd aangenomen dat licht langs een straal geen energie verliest. Dat is inderdaad een redelijke veronderstelling in vacuüm of zuivere lucht. Als men echter participerende media zoals wolken, vuur of gekleurd glas wil visualiseren (figuur 5.14), moet men er wel rekening mee houden dat de energie kan afnemen of toenemen. De rendering vergelijking (5.15) wordt dan een stuk ingewikkelder:

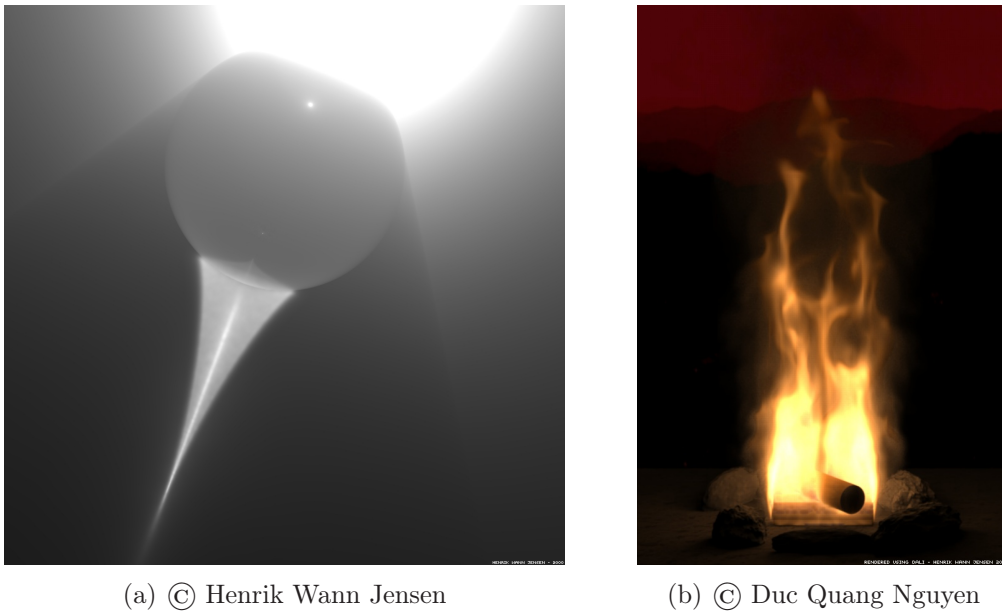
$$\begin{aligned}
 L(x \rightarrow \Theta, \lambda_{\text{uit}}) = & L_e(x \rightarrow \Theta, \lambda_{\text{uit}}) \\
 & + \iint_{\Omega_x} L(y \rightarrow -\Psi, \lambda_{\text{in}}) \tau(x, y, \lambda_{\text{in}}) \\
 & \quad \cdot f_r(x, \Psi \leftrightarrow \Theta, \lambda_{\text{in}}, \lambda_{\text{uit}}) \cos(\hat{n}_x, \Psi) d\omega_\Psi d\lambda_{\text{in}} \\
 & + \iint_{\Omega_x} \left(\int_0^{r_{xy}} L^+(z \rightarrow -\Psi, \lambda_{\text{in}}) \tau(z, x, \lambda_{\text{in}}) dr \right) \\
 & \quad \cdot f_r(x, \Psi \leftrightarrow \Theta, \lambda_{\text{in}}, \lambda_{\text{uit}}) \cos(\hat{n}_x, \Psi) d\omega_\Psi d\lambda_{\text{in}},
 \end{aligned} \tag{5.16}$$

waarin de emissie (ϵ) en inkomende verstrooiing (σ_s) gegeven worden door:

$$L^+(z \rightarrow \Theta, \lambda) = \frac{\epsilon(z, \lambda)}{4\pi} + \int_{\Omega_z} p(z, \Psi \leftrightarrow \Theta, \lambda) \sigma_s(z, \lambda) L(z \rightarrow \Psi, \lambda) d\omega_\Psi, \tag{5.17}$$

en waarin de uitdoving door uitgaande verstrooiing (σ_t) gegeven wordt door:

$$\tau(z, x, \lambda) = \exp \left(- \int_0^{r_{zx}} \sigma_t(z + s\Theta, \lambda) ds \right). \tag{5.18}$$



Figuur 5.14: (a) Een volumetrische brandvlek in een participerend medium; (b) vuur als participerend medium (Nguyen *et al.* 2002).

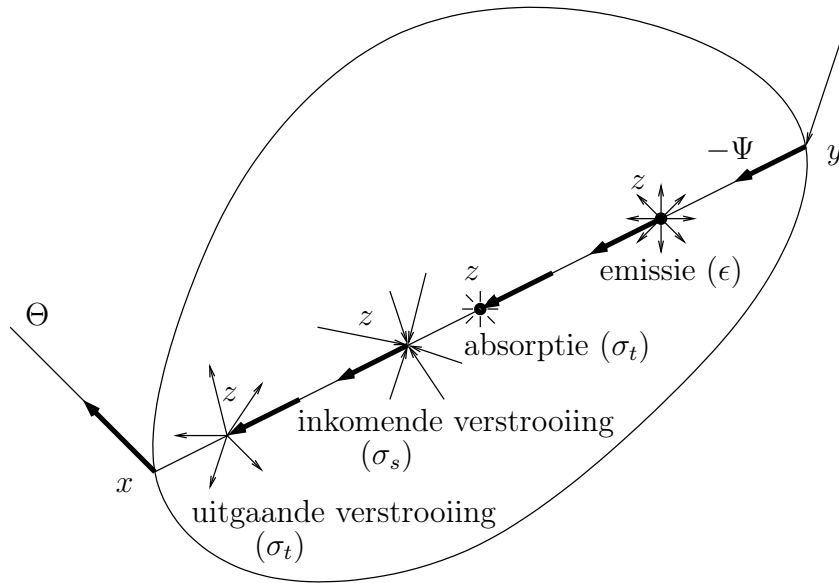
De fase-functie $p(z, \Psi \leftrightarrow \Theta, \lambda)$ beschrijft de kans dat een lichtstraal met golflengte λ op het punt z van richting Ψ naar richting Θ wordt verstrooid. Er bestaan empirische formules die met enkele parameters kunnen gefit worden op meetwaarden. Zo kan men de Rayleigh- en Mie-verstrooiing in de atmosfeer van de aarde simuleren. Die is sterk golflengte-afhankelijk en zorgt bijvoorbeeld voor de blauwe kleur van de hemel en de rode kleur van een zonsondergang.

De formules zien er heel complex uit, maar om participerende media te begrijpen, hoeft men eigenlijk alleen figuur 5.15 voor ogen te houden. De uitgaande radiantie uit punt x in richting Θ is afhankelijk van alle stralen die ergens het medium binnendringen in y . Onderweg kunnen er 4 dingen gebeuren met het licht dat van y naar x reist:

emissie: er komt energie bij doordat het medium licht uitstraalt (b.v. een vlam).

absorptie: er verdwijnt energie door interactie met stofdeeltjes die licht omzetten in warmte of andere straling.

inkomende verstrooiing: er komt licht van elders in het medium en door een botsing met een stofdeeltje reist het mee naar x .



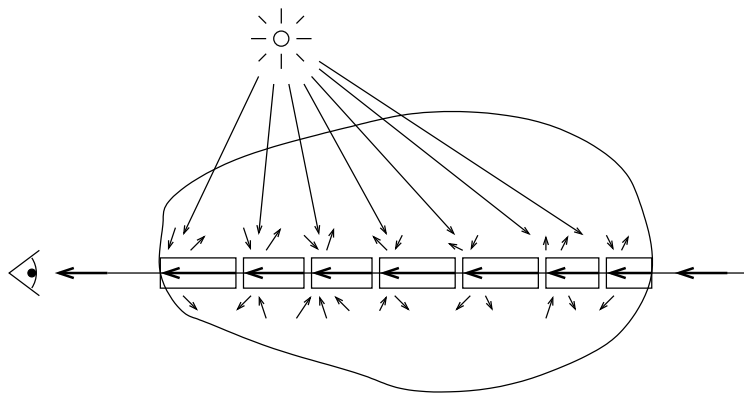
Figuur 5.15: Schematische voorstelling van de interactie met een participerend medium op verschillende plaatsen z tussen y en x .

uitgaande verstrooiing: licht botst en reist in een andere richting verder.

Als het medium homogeen is, d.w.z. de 4 interacties treden op met een kans die onafhankelijk is van de plaats in het medium, dan kunnen er een aantal integralen vereenvoudigd worden. De wet van Beer-Lambert-Bouguer zegt bijvoorbeeld dat de uitdoving τ in een homogeen medium gegeven wordt door:

$$\tau(z, x, \lambda) = \exp(-r_{zx}\sigma_t(\lambda)). \quad (5.19)$$

In niet-homogene media kan men die vereenvoudiging niet maken. De integralen moeten op een numerieke manier berekend worden, namelijk door een techniek die men *ray marching* noemt (Jensen en Christensen 1998). Daarbij wordt de straal in het medium verdeeld in kleine stapjes, en op ieder stapje kan er wel een lokaal homogeen medium verondersteld worden (figuur 5.16).



Figuur 5.16: Schematische voorstelling van *ray marching*.

Interferentie in een dunne film

De plaats waar je in het dagelijkse leven meestal interferentie ziet, is in een dunne film. Dat is een dunne laag van een doorschijnende stof, zoals bijvoorbeeld een olielaagje, een zeepbel, bepaalde coatings en verfsoorten, . . .

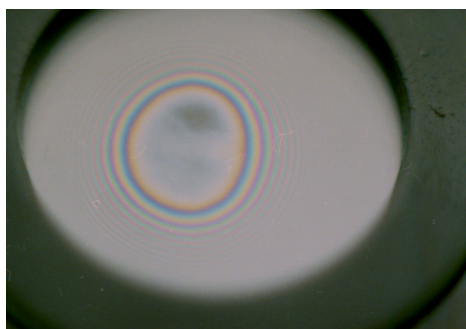
Er is enorm veel gepubliceerd over het renderen van interferentie in een dunne film en over concrete toepassingen. We geven een kort overzicht van het belangrijkste gerelateerde werk en daarna wordt een concreet model toegelicht.

6.1 Voorgaand werk

Het fysische model voor interferentie in een dunne film is relatief eenvoudig, maar een correcte wiskundige beschrijving blijkt niet zo vanzelfsprekend. Smits en Meyer stelden als eersten een manier voor om interferentie te visualiseren (Smits en Meyer 1992). Ze behandelden concreet olielaagjes, zeepbellen en ook de ringen van Newton. Dat zijn meetkundige plaatsen waar de dunne laag dezelfde dikte heeft en dus dezelfde kleur veroorzaakt. Meestal zijn die zichtbaar als regenboogkleurige ringen. Hetzelfde effect is ook te zien op de plaats waar een gebogen glazen voorwerp een ander oppervlak raakt (figuur 6.1).

Ook Dias beschrijft het algemene effect van interferentie (Dias 1991) en de ringen van Newton (Dias 1994). De belangrijkste beperking is dat enkel de interferentie in het weerkaatste licht wordt berekend.

Sun *et al.* hebben een veralgemeende methode bedacht om ook het doorgaande licht correct te laten interfereren (Sun *et al.* 1999, Sun 2000). Deze methode hadden we eerst geïmplementeerd, maar ze bleek nog niet realistisch genoeg. Als het licht of de zichtstraal heel vlak invalt op de zeepbel, dus aan



Figuur 6.1: Ringen van Newton rond het raakpunt van een bolle lens aan een ander oppervlak.

de zijkanten, dan wordt de formule van Fresnel niet correct toegepast. Het gevolg was een zeepbel met een erg donkere rand.

Om dit probleem op te lossen, stellen Jazkowski en Rzeszut een volledig model op voor alle Fresnel-coëfficiënten die in de zeepbel voorkomen (Jazkowski en Rzeszut 2003). Hun aanpak laat toe om de rekentijd te beperken door controle over het aantal interne weerkaatsingen in de zeepfilm. Hun uitgangspunt was dat de bijdrage van meerdere interne reflecties meestal toch verwaarloosbaar is voor een realistische weergave. Ze beschrijven bovendien een eenvoudig model voor de dikte van een zeepbel, waardoor er ook ringen van Newton in de kleur van de zeepbel komen.

Glassner beschrijft de scheikundige samenstelling van het zeepmengsel en de geometrie van rakende zeepbellen (Glassner 2000a, Glassner 2000b). Dit is een interessant onderwerp voor realistisch gemodelleerde scènes, maar het valt buiten het bestek van deze thesis.

Gondek *et al.* bespreken vooral verflagen, coatings en een gelijkaardig effect in parels (Gondek *et al.* 1994). Dit wordt door Schramm *et al.* uitgebreid naar meerdere lagen van coatings (Schramm *et al.* 1997). Ze renderen ook de interferentie in de veren van een vogel. Hirayama *et al.* demonstreren dat de kleuren in gelaagde coatings duidelijker worden als het oppervlak ruw is (Hirayama *et al.* 2001a, Hirayama *et al.* 2001b).

6.2 Theoretisch model

Algemeen

Het fysische principe achter interferentie in een dunne film is schematisch weergegeven in figuur 6.2. De film heeft twee evenwijdige oppervlakken, en heeft een andere brekingsindex dan de omgeving. Een lichtstraal valt op het bovenste oppervlak van de film. Daar bepalen de Fresnel-coëfficiënten (voor optisch isotrope diëlectrica) in welke verhouding het vermogen weerkaatst (R) en doorgelaten wordt (T):

$$R = \frac{\sin^2(\theta_i - \theta_r)}{\sin^2(\theta_i + \theta_r)} \cdot \left(1 + \frac{\cos^2(\theta_i + \theta_r)}{\cos^2(\theta_i - \theta_r)} \right), \quad (6.1)$$

$$T = \frac{n_2}{n_1} \cdot \frac{4 \cos^2 \theta_i \sin^2 \theta_r}{\sin^2(\theta_i + \theta_r) \cos^2(\theta_i - \theta_r)} \cdot (1 + \cos^2(\theta_i - \theta_r)),$$

waarbij de wet van Snellius de hoek θ_r bepaalt, de richting waarin de indringende straal loopt:

$$\frac{\sin \theta_i}{\sin \theta_r} = \frac{n_2}{n_1}. \quad (6.2)$$

De weerkaatste straal wordt natuurlijk perfect gespiegeld ten opzichte van de normaal. Het weerkaatste licht maakt een fasesprong van een halve golflengte, omdat het weerkaatst aan een dichter medium dan de omringende lucht:

$$n_2 \text{ (zeep)} > n_1 \text{ (lucht)}. \quad (6.3)$$

Met het indringende licht gebeurt aan het onderste oppervlak weer een splitsing door de formules van Fresnel, alleen is er hier nooit een fasesprong. Zo kan een straal in principe oneindig intern blijven reflecteren.

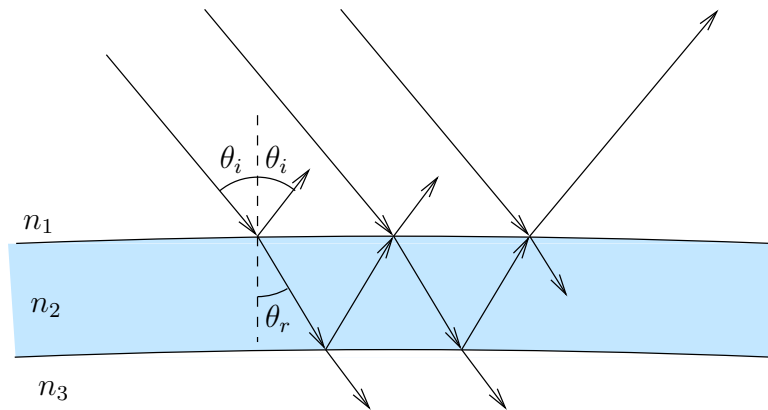
Interferentie vereist een verschil in afgelegde weg. Dat verschil wordt hier veroorzaakt door de interne reflecties en ook de fasesprong kan beschouwd worden als een verschil in afgelegde weg. Bovendien moeten we de *optische* weglengte berekenen. Die verschilt van de *geometrische* weglengte als volgt:

$$l_{\text{optisch}} = n \cdot l_{\text{geometrisch}}. \quad (6.4)$$

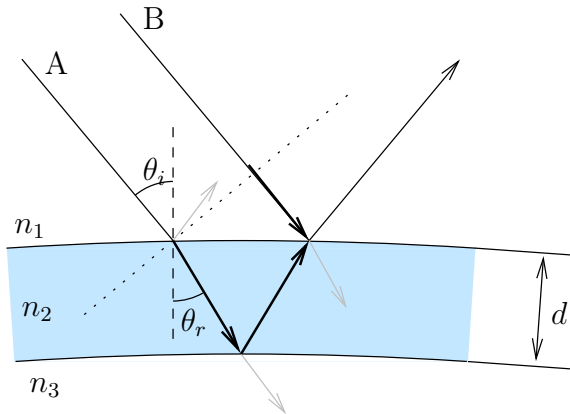
In lucht zijn die weglengtes bijna hetzelfde ($n_{\text{lucht}} = 1,00029$), maar in de zeepoplossing wordt het verschil belangrijk ($n_{\text{zeep}} = 1,5$).

Interne reflecties

We beschouwen een straal die in de film indringt, één keer intern weerkaatst, en er bovenaan weer uitkomt. Die straal zal bijvoorbeeld interfereren met



Figuur 6.2: Fysisch model voor interferentie in een dunne film.

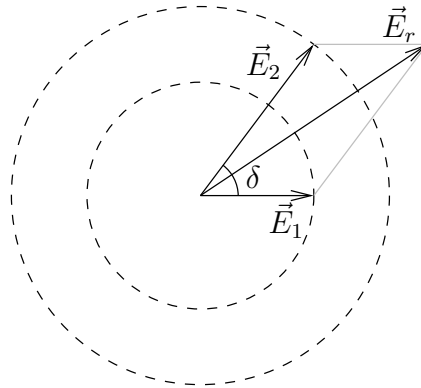


Figuur 6.3: Verschil in afgelegde weg.

een straal die aan het bovenste oppervlak direct weerkaatst. Dit is meteen de belangrijkste bijdrage aan de visuele indruk. Meer interne reflecties of complexere situaties kunnen vaak verwaarloosd worden. Het verschil in afgelegde weg voor deze bijdrage is getekend in figuur 6.3.

De invallende stralen leggen dezelfde weg af tot aan de stippellijn. Door de kleine afstanden die hier meespelen, mogen we veronderstellen dat de stralen aan de stippellijn coherent zijn. Let op, dit geldt enkel voor een *dunne* film, met afmetingen in de grootte-orde van de golflengte. Straal B moet vanaf de stippellijn nog een eindje verder lopen tot aan het oppervlak, en maakt daar een fasesprong van een halve golflengte:

$$l_1 = 2d \tan \theta_r \sin \theta_i + \lambda/2. \quad (6.5)$$



Figuur 6.4: Fasorvoorstelling van de superpositie van twee golven.

Straal A legt in de zeeplaag een *optische* weglengte af van:

$$l_2 = 2dn_2 / \cos \theta_r. \quad (6.6)$$

Het verschil tussen deze afstanden l_1 en l_2 zorgt voor de interferentie:

$$\Delta l = 2dn_2 \cos \theta_r + \lambda/2, \quad (6.7)$$

of als we vermenigvuldigen met het golfgetal k :

$$\delta = \frac{4\pi}{\lambda} dn_2 \cos \theta_r + \pi. \quad (6.8)$$

Als we de twee elektrische golven met amplitudes E_1 en E_2 superponeren, wordt de resulterende amplitude E_r gegeven door:

$$E_r^2 = E_1^2 + E_2^2 + \underbrace{2E_1E_2 \cos \delta}_{\text{interferentie}}. \quad (6.9)$$

Deze formule kan eenvoudig aangetoond worden via de cosinusregel in de fasorvoorstelling van de golven (figuur 6.4).

Het relatieve vermogen van de straal aan het einde van zo'n complex pad wordt gegeven door een vermenigvuldiging van de Fresnel-coëfficiënten, bijvoorbeeld voor één interne reflectie:

$$T_{12} \cdot R_{23} \cdot T_{21}, \quad (6.10)$$

waarbij R_{ij} en T_{ij} de reflectie- resp. transmissiecoëfficiënt zijn, komende van medium i aan het oppervlak met medium j . Het vermogen van de resulterende straal vermindert, m.a.w. het kwadraat van de amplitude verkleint met die factor.

Bij k interne weerkaatsingen heen en weer wordt de formule:

$$\begin{aligned} T_{12} \cdot R_{23} \cdot R_{21} \cdot R_{23} \cdots T_{21} &= T_{12} \cdot R_{23}^k \cdot R_{21}^{k-1} \cdot T_{21} \\ &= T_{12} \cdot R_{23} \cdot T_{21} \cdot X_k. \end{aligned} \quad (6.11)$$

De functie X_k wordt recursief gedefinieerd als:

$$\begin{aligned} X_k &= 1 + R_{21}R_{23}X_{k-1} + 2\sqrt{R_{21}R_{23}X_{k-1}} \cos \delta, \\ X_1 &= 1. \end{aligned} \quad (6.12)$$

Een gelijkaardige berekening geldt voor de straal die door de film dringt.

Dit fysisch model werkt enkel bij een *dunne* film, omdat we veronderstellen dat het licht coherent blijft over de afstanden die we beschouwen. De optica zegt dat deze veronderstelling geldig is als de afstanden in de grootteorde van de golflengte liggen. Bij een *dunne* film hebben we inderdaad afmetingen rond $1 \mu\text{m}$. Merk ook op dat de brekingsindices n als constant mogen beschouwd worden. Dispersie, veroorzaakt door de golflengte-afhankelijkheid van de brekingsindex, is over deze afstanden verwaarloosbaar.

Dikte van de film

De dikte van de film ligt zoals gezegd in de buurt van $1 \mu\text{m}$, maar voor de ringen van Newton is vooral het verloop van de dikte belangrijk. Door de zwaartekracht zal de zeep naar beneden zakken, zodat de film bovenaan dunner wordt dan onderaan. Een fysisch correcte berekening voor de dikte zou nodeloos zwaar worden, maar een eenvoudige lineaire benadering voldoet niet. Ook de benadering van twee perfect bolvormige oppervlakken geeft niet het gewenste resultaat. Jazzkowski en Rzeszut stellen een handig empirisch model voor, namelijk een exponentieel verloop (Jazzkowski en Rzeszut 2003):

$$d(a) = d_{\min} \exp\left(\frac{a}{h} \ln\left(\frac{d_{\max}}{d_{\min}}\right)\right). \quad (6.13)$$

De dikte d op hoogte a varieert van d_{\min} op hoogte 0 (bovenaan) tot d_{\max} op hoogte h (onderaan).

6.3 Implementatie

Het fysische model past niet zonder meer in de architectuur van een standaard ray tracer. We volgen traditioneel maar één inkomende straal, terwijl

de interferentie veroorzaakt wordt door het samenspel tussen twee of meer stralen. De optica biedt gelukkig redding: bij de beschouwde afstanden mogen we veronderstellen dat er een coherente, evenwijdige lichtbundel invalt op het zeepoppervlak. We zullen dus maar één inkomende straal expliciet volgen, en we veronderstellen dat de andere stralen er ook zijn.

Het model van Jaszkowski en Rzeszut biedt een volledige implementatie van het fysische model. Dat betekent dat we zowel een BRDF (listing 6.1) als een BTDF (listing 6.2) implementeren. Daarbij maken we gebruik van de functie X_k (listing 6.3).

We moeten ook aannemen dat de uitgaande straal vertrekt op hetzelfde punt waar de inkomende straal invalt. De film heeft dus eigenlijk geen dikte voor de berekening van de stralen, alleen voor de reflectantie en transmittantie. Deze veronderstelling moeten we maken om aan de architectuur van de ray tracer te voldoen.

Listing 6.1: ThinFilm BRDF.

```

Vector& wo;           // uitgaande richting
Vector* wi;          // inkomende richting
float index1, index2; // brekingsindices buiten en binnen
float indexFilm;     // brekingsindex film
float thickness;     // dikte van de film in micrometer
FresnelDielectric fresnel12; // Fresnel-formule buitenste oppervlak
FresnelDielectric fresnel23; // Fresnel-formule binnenste oppervlak
Spectrum Kr;        // kleur van de speculaire reflectie

// Inkomende richting
*wi = Vector(-wo.x, -wo.y, wo.z);

// Komt de straal van buiten de zeepbel?
bool entering = CosTheta(wo) > 0.;

// Richting van de straal in de film (wet van Snellius)
float sintr2 = SinTheta2(wo) * index1 * index2 / indexFilm / indexFilm;

// Totale interne reflectie, geen indringende straal, geen interferentie
if (sintr2 > 1.)
    return Kr/fabsf(CosTheta(*wi));

float costr = sqrtf(max(0.f, 1.f - sintr2));
if (entering) costr = -costr;

// Verschil in weglengte, in radialen
float dtmp = 4000. * M_PI * indexFilm * thickness * fabsf(costr);
float samples[100];
for (int i = 0; i < 100; i++) {
    float wvl = 360. + i*(830.-360.)/99.;
    samples[i] = cos(dtmp/wvl);
}
Spectrum cosd(samples, 100, 360., 830.);

// Fresnel-coëfficiënten voor reflectie en transmissie aan de twee oppervlakken
Spectrum R12 = (entering ? fresnel12.Evaluate(CosTheta(*wi))
                    : fresnel23.Evaluate(CosTheta(*wi)));
Spectrum R21 = (entering ? fresnel12.Evaluate(costr)
                    : fresnel23.Evaluate(costr));
Spectrum R23 = (entering ? fresnel23.Evaluate(-costr)
                    : fresnel12.Evaluate(-costr));
Spectrum T12 = (Spectrum(1.) - R12)*indexFilm/index1;
Spectrum T21 = (Spectrum(1.) - R21)*index1/indexFilm;

// Aantal interne reflecties
Spectrum X = Xk(1, R21, R23, cosd);
Spectrum tmp = T12*R23*T21*X;
return Kr * (R12 + tmp + 2*((R12*tmp).Sqrt()*cosd)) / fabsf(CosTheta(*wi));

```

Listing 6.2: ThinFilm BTDF.

```

Vector& wo;           // uitgaande richting
Vector* wi;          // inkomende richting
float index1, index2; // brekingsindices buiten en binnen
float indexFilm;     // brekingsindex film
float thickness;     // dikte van de film in micrometer
FresnelDielectric fresnel12; // Fresnel-formule buitenste oppervlak
FresnelDielectric fresnel23; // Fresnel-formule binnenste oppervlak
Spectrum Kt;        // kleur van de speculaire transmissie

// Komt de straal van buiten de zeepbel?
bool entering = CosTheta(wo) > 0.;

// Richting van de straal aan de andere kant van de film (wet van Snellius)
float sintt2 = SinTheta2(wo) * index1 * index1 / index2 / index2;

// Totale interne reflectie
if (sintt2 > 1.)
    return 0.;

float costt = sqrtf(max(0.f, 1.f - sintt2));
if (entering) costt = -costt;

// Inkomende richting
*wi = Vector(-wo.x * index1 / index2, -wo.y * index1 / index2, costt);

// Richting van de straal in de film (wet van Snellius)
float sintr2 = SinTheta2(wo) * index1 * index1 / indexFilm / indexFilm;

// Totale reflectie
if (sintr2 > 1.)
    return 0.;

float costr = sqrtf(max(0.f, 1.f - sintr2));
if (entering) costr = -costr;

// Verschil in weglengte, in radialen
float dtmp = 4000.*M_PI*indexFilm*thickness*fabsf(cosr);
float samples[100];
for (int i = 0; i < 100; i++) {
    float wvl = 360. + i*(830.-360.)/99.;
    samples[i] = cos(dtmp/wvl);
}
Spectrum cosd(samples, 100, 360., 830.);

// Fresnel-coëfficiënten voor reflectie en transmissie aan de twee oppervlakken
Spectrum R12 = (entering ? fresnel12.Evaluate(CosTheta(wo))
    : fresnel23.Evaluate(CosTheta(wo)));
Spectrum R21 = (entering ? fresnel12.Evaluate(costr)
    : fresnel23.Evaluate(costr));
Spectrum R23 = (entering ? fresnel23.Evaluate(-costr)
    : fresnel12.Evaluate(-costr));
Spectrum T12 = (Spectrum(1.) - R12)*indexFilm/index1;
Spectrum T21 = (Spectrum(1.) - R21)*index1/indexFilm;
Spectrum T23 = (Spectrum(1.) - R23)*index2/indexFilm;

// Aantal interne reflecties
Spectrum X = Xk(1, R21, R23, cosd);
Spectrum tmp = (index1/index2)*T12*T23;
return Kt * (tmp + tmp*R23*R21*X + 2*tmp*((R23*R21*X).Sqrt()*cosd))
    * index2*index2/(index1*index1*fabsf(CosTheta(*wi)));

```

Listing 6.3: ThinFilm functie X_k (formule 6.12).

```
int k; // aantal interne reflecties
Spectrum& R21, R23; // Fresnel-coefficienten
Spectrum& cosd; // verschil in weglengte

if (k==1) return Spectrum(1.);
Spectrum Xk1 = Xk(k-1, R21, R23, cosd);
return Spectrum(1.) + R21*R23*Xk1 + 2*(R21*R23*Xk1).Sqrt()*cosd;
```



(a) © Jan-Willem Bijl



(b)

Figuur 6.5: (a) Een foto van een zeepbel en (b) een gerenderde gelijkaardige zeepbel.

6.4 Resultaten

Figuur 6.5(a) toont een foto van een zeepbel die we zo goed mogelijk hebben nageemaakt in figuur 6.5(b). We gebruiken omgevingslicht van een *environment map*. Door de visuele indruk te vergelijken, hebben we de parameters van de zeepbel vrij nauwkeurig kunnen schatten.

De dikte van de film varieert van 530 nm bovenaan tot 730 nm onderaan. Dat is een te klein verloop om duidelijk horizontale ringen van Newton te zien. We zien wel concentrische gekleurde ringen, omdat de afstand δ (vergelijking 6.8) groter wordt naarmate de zichtstraal vlakker invalt.

In figuur 6.6 zijn de horizontale ringen beter zichtbaar. De dikte verloopt van 200 nm tot 1500 nm.



Figuur 6.6: Ringen van Newton in een zeepbel met een groter dikte-verloop (van 200 nm tot 1500 nm).

DiffRACTIE aan gegroefde materialen

De opvallende kleuren van een compact disc (cd) zijn te verklaren aan de hand van de microstructuur van het oppervlak. De informatie op een cd wordt opgeslagen in een spiraalvormig spoor van putjes. De informatie wordt uitgelezen door de reflectie van laserlicht op het oppervlak te meten. In een putje zorgt het verschil in weglengte voor destructieve interferentie en is de gemeten reflectie kleiner.

Het is niet verwonderlijk dat het oppervlak van een cd ook interageert met andere lichtbronnen. Het periodieke rooster van putjes gedraagt zich als een buigingsrooster of *tralie*. Stralen interfereren met elkaar als ze in nabije putjes weerkaatsen.

7.1 Voorgaand werk

Er zijn verschillende manieren voorgesteld om een cd te visualiseren. Moravec beschrijft een volledige renderer op basis van golfoptica i.p.v. de geometrische optica die in ray tracing wordt gebruikt (Moravec 1981). Door golfoppervlakken te propageren door de ruimte zou men in principe een cd kunnen visualiseren, maar de methode heeft te kampen met een gebrek aan geheugen en processorkracht. De microstructuur van het oppervlak moet expliciet gemodelleerd worden, waardoor de methode beperkt bleef tot enkele eenvoudige toepassingen zoals de interferentieproef van Young (coherent licht door twee spleten). De miljarden putjes van een cd zouden zelfs voor hedendaagse machines een probleem vormen. Kajiya gebruikt een vereenvoudigde vorm van golfoptica in ray tracing, met dezelfde beperkingen (Kajiya 1985).

Een betere manier om dit probleem te benaderen, is door een analytisch reflectiemodel af te leiden. Het belangrijke verschil met gewone microfacet-

modellen zoals Blinn of Cook-Torrance, is dat het met golfoptica wordt afgeleid. Het resultaat is een eenvoudige BRDF voor een gewone ray tracer. Bahar en Chakrabarti legden de basis voor deze methode (Bahar en Chakrabarti 1987). Stam past de methode toe op anisotrope materialen, o.a. de cd (Stam 1999). Hij converteert het spectrale resultaat naar RGB voor gebruik in gewone ray tracers. Dit is het eerste succesvolle model voor de visualisatie van een cd.

Sun *et al.* maken de terechte opmerking dat het model van Stam niet helemaal overeenstemt met de werkelijkheid (Sun *et al.* 1999b, Sun *et al.* 2000, Sun 2000). Ze stellen een nieuw model voor, met een nauwkeurigere microstructuur en complexere golfoptica. Ze converteren hun resultaat ook niet naar RGB omdat hun ray tracer al volledig spectraal werkt. Dit model hebben we uitgebreid bestudeerd en geïmplementeerd.

7.2 Beschrijving van de microstructuur

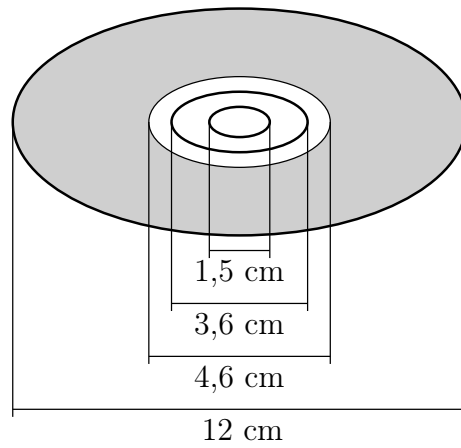
Het spiraalvormige spoor op het oppervlak van een cd bestaat uit miljarden putjes. De putjes zijn 125 nm diep en 600 nm breed. De lengte van de putjes en de afstand tussen de putjes varieert van 900 nm tot 3300 nm in stappen van 300 nm, afhankelijk van de gegevens op de cd. Iedere stap van 300 nm betekent een volgende 1-bit in een putje, of een volgende 0-bit op het *land*.¹ Er wordt een codering toegepast om de synchronisatie niet te verliezen bij lange rijen identieke bits, vergelijkbaar met *bit stuffing* in datatransmissie. Hierdoor is de grootst mogelijke lengte 3300 nm. Als de gegevens voldoende willekeurig zijn, is iedere volgende bit met gelijke kans een 0 of een 1. De gemiddelde lengte van een putje of een tussenafstand is dan:

$$900 \text{ nm} + 300 \text{ nm} \cdot \sum_{i=1}^8 \frac{1}{2^i} = 1198,8 \text{ nm} \approx 1200 \text{ nm}. \quad (7.1)$$

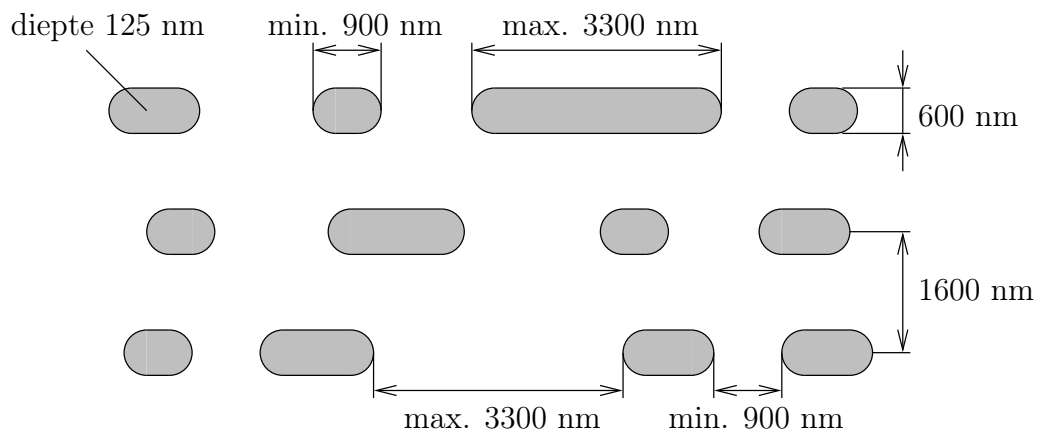
De gemiddelde afstand van het midden van een putje tot het midden van het volgende putje is dus 2400 nm. De afstand tussen de sporen is 1600 nm. Alleen deze laatste twee waarden worden gebruikt in het geïmplementeerde model. Figuren 7.1 en 7.2 geven een overzicht van de macroscopische en microscopische dimensies van een compact disc.

Figuur 7.3 toont een compact disc onder een puntlichtbron. Er zijn duidelijk verschillende gekleurde gebieden zichtbaar.

¹Het oppervlak tussen de putjes wordt in de literatuur *land* genoemd.



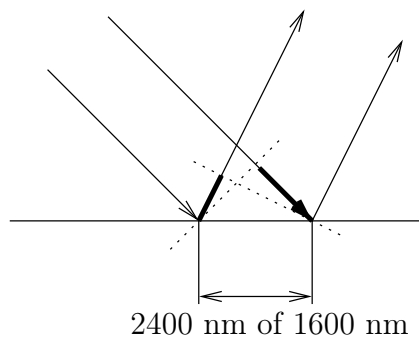
Figuur 7.1: Macroscopische dimensies van een compact disc.



Figuur 7.2: Microscopische dimensies van een compact disc.



Figuur 7.3: Foto van een compact disc onder een puntlichtbron.



Figuur 7.4: Model voor diffractie aan een compact disc. De dikkere lijn geeft het verschil in afgelegde weg.

- De *hoofdstrip* heeft een radiaal kleurverloop dat ontstaat door diffractie tussen sporen op een afstand van 1600 nm.
- De twee *nevenstrips* hebben een transversaal kleurverloop dat veroorzaakt wordt door putjes op een afstand van 2400 nm.

7.3 Wiskundig model

Zoals vermeld houden we enkel rekening met de afstand van 1600 nm voor de hoofdstrip en 2400 nm voor de nevenstrips. We houden zelfs geen rekening met de diepte van de putjes. We beschouwen het oppervlak van de cd als een effen vlak. Lichtstralen die invallen op een afstand van 1600 nm in radiale richting, of 2400 nm in tangentiële richting, zullen interfereren.

De twee beschouwde lichtstralen leggen een verschillende weg af. De redenering is schematisch weergegeven in figuur 7.4. De twee stralen lopen gelijk tot aan de eerste stippellijn, waar de linkse straal invalt. De rechtse straal heeft dan nog een stukje af te leggen tot aan het oppervlak. Bij de

tweede stippellijn is de rechtse straal ook weerkaatst, maar dan moet de linkse straal nog een eindje bijbenen. Voorbij de tweede stippellijn lopen ze weer samen.

Het verschil in afgelegde weg is te berekenen uit de inkomende en uitgaande hoeken en de afstand tussen de punten op het oppervlak. Zoals bij gewone interferentie moet dat verschil een geheel aantal golflengtes bedragen:

$$d(\hat{\omega}_i + \hat{\omega}_o) \cdot \hat{v} = n\lambda, \quad (7.2)$$

waarbij $\hat{\omega}_i$ en $\hat{\omega}_o$ de inkomende resp. uitgaande richtingen zijn. De richting \hat{v} is radiaal of tangentieel aan de sporen en de afstand d is de relevante afmeting in die richting.

Nevenstrips

Voor de nevenstrips wordt dit concreet:

$$2400 \text{ nm} (\hat{\omega}_i + \hat{\omega}_o) \cdot \hat{x} = n\lambda, \quad (7.3)$$

met \hat{x} de richting rakend aan de sporen. Het gehele getal n geeft aan of het gaat om primaire nevenstrips ($n = \pm 1$) of secundaire nevenstrips ($n = \pm 2, \pm 3, \dots$). Secundaire nevenstrips hebben identiek hetzelfde kleurverloop, maar ze liggen verder uit elkaar. Ze zijn veel zwakker dan de primaire nevenstrips, zodat ze zelden zichtbaar zijn. Het is dus voldoende om voor enkele waarden n een bijdrage te berekenen.

In principe zouden die bijdragen discrete pieken zijn: in de bovenstaande vergelijking met discrete waarden n bestaat de oplossing uit discrete waarden λ . Maar herinner dat de afstand 2400 nm geen exacte afstand was, maar een gemiddelde. Om dit weer te geven in het reflectantiespectrum gebruiken we Gauss-curves met een zekere breedte:

$$f^2 = \frac{c_1 \sqrt{\pi}}{2\sigma_1} \exp(-4(dk(\hat{\omega}_i + \hat{\omega}_o) \cdot \hat{x} - 2\pi n)^2 / \sigma_1^2) + c_2, \quad (7.4)$$

voor $n = \pm 1, \pm 2, \pm 3, \dots$. De getallen c_1 , c_2 en σ_1 zijn waarden die in theorie kunnen bepaald worden door een grondige studie van de microstructuur. Wij hebben ze zo goed mogelijk benaderd door de visuele indruk van gerenderde afbeeldingen te vergelijken met foto's.

Hoofdstrip

De hoofdstrip ziet er als volgt uit:

$$1600 \text{ nm} (\hat{\omega}_i + \hat{\omega}_o) \cdot \hat{y} = n\lambda, \quad (7.5)$$



Figuur 7.5: Een grammofoonplaat met in het midden een compact disc.

met \hat{y} de radiale richting. De factor $(\hat{\omega}_i + \hat{\omega}_o) \cdot \hat{y}$ is begrensd tussen 0 en ± 2 . Aangezien de golflengtes van het zichtbare licht tussen 360 en 830 nm liggen (ruim geschat), kunnen we de factor n beperken tussen:

$$\frac{1600 \text{ nm}}{830 \text{ nm}} (\hat{\omega}_i + \hat{\omega}_o) \cdot \hat{y} \leq n \leq \frac{1600 \text{ nm}}{360 \text{ nm}} (\hat{\omega}_i + \hat{\omega}_o) \cdot \hat{y}. \quad (7.6)$$

Het volstaat dus om interferentie voor enkele waarden van n in rekening te brengen. Voor die waarden bekomen we wel een echte piek, want de waarde 1600 nm is een vaste afstand. Voor de grootte van de piek baseren we ons op de nevenstrip (7.4) met $n = 0$:

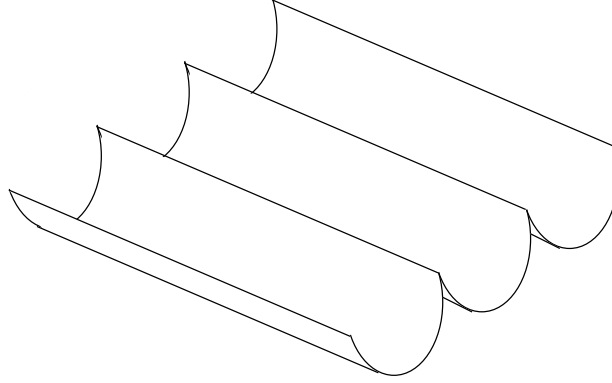
$$g^2 = c_3 \cdot \left(\frac{c_1 \sqrt{\pi}}{2\sigma_2} \exp(-4(dk(\hat{\omega}_i + \hat{\omega}_o) \cdot \hat{x})^2 / \sigma_2^2) + c_2 \right) + c_4. \quad (7.7)$$

De getallen c_3 , c_4 en σ_2 zijn opnieuw bepaald door de visuele indruk.

We hebben geen afdoende verklaring voor het feit dat we de formule van de nevenstrips hergebruiken voor de hoofdstrip. De fysisch verantwoorde implementatie beschreven in (Sun 2000) gaf geen realistische resultaten. Aangezien in het model de fysische basis vaak sterk vereenvoudigd wordt, leek het ons aanvaardbaar om ook in dit geval de visuele indruk te laten primeren op de fysische correctheid.

Anisotrope component

De reflectie van een cd bevat ook een anisotrope speculaire component. Die wordt veroorzaakt door de cirkelvormige sporen, net zoals bij de groeven van een grammofoonplaat. Op figuur 7.5 is deze component zichtbaar als een radiale *highlight* die samenvalt met de hoofdstrip. Merk op dat deze component geen rekening houdt met de golflengte, dus dit is geen diffractieve component.



Figuur 7.6: Oppervlak met cilindervormige groeven.

Voor onze toepassing volstaat een eenvoudig model van cilindrische groeven (figuur 7.6). Voor de afleiding verwijzen we naar (Sun *et al.* 2000, Sun 2000). Het resultaat is een eenvoudige formule die wat weg heeft van een Phong-lob (Phong 1975), maar dan anisotroop en met een bijkomend exponentieel verval:

$$R_{\text{aniso}} = \exp\left(-c_5 |(\hat{\omega}_i + \hat{\omega}_o) \cdot \hat{x}|^\beta\right), \quad (7.8)$$

met c_5 en β positieve experimentele waarden.

De diffractieve componenten (de hoofd- en nevenstrips) worden vermenigvuldigd met deze anisotrope component. Dit is de reden waarom de secundaire nevenstrips zelden zichtbaar zijn, en we de berekening van de nevenstrips kunnen beperken tot $n = \pm 3$.

Poulin en Fournier geven een uitgebreidere afleiding van reflectie aan cilindrische groeven (Poulin en Fournier 1990). Dit leidt tot een moeilijkere formule, maar de theorie kan ook gebruikt worden om c_5 en β te schatten.

Isotrope speculaire en diffuse component

Om het af te maken, voegen we nog een gewone speculaire en diffuse component toe. Een compact disc is in de eerste plaats een spiegelend oppervlak. De volledige reflectantie wordt dan:

$$f_r = c_0 \cdot R_{\text{aniso}} \cdot (f^2 + g^2) + c_{\text{spec}} |\hat{\omega}_h \cdot \hat{n}|^\alpha + c_{\text{diff}} (\hat{\omega}_o \cdot \hat{n}). \quad (7.9)$$

De eenheidsvector $\hat{\omega}_h$ is de *halve-hoek-vector*, de bissectrice van de inkomende en uitgaande richting:

$$\hat{\omega}_h = \frac{\hat{\omega}_i + \hat{\omega}_o}{\|\hat{\omega}_i + \hat{\omega}_o\|}. \quad (7.10)$$

7.4 Implementatie

Net zoals bij interferentie moeten we nog een bijkomende vereenvoudiging maken om te voldoen aan de *interface* van de ray tracer. We volgen maar één inkomende en uitgaande straal, en we nemen aan dat de andere nodige stralen er ook zijn. De wetten van de optica laten deze veronderstelling toe. De implementatie bestaat enkel uit een BRDF (listing 7.1).

De kleuring (shading) gebeurt in PBRT in een lokaal coördinatenstelsel met als oorsprong het punt waar de stralen invallen. De richtingen \hat{x} , \hat{y} en \hat{n} zijn dankzij de parametrisatie van de schijf-primitief gelijk aan $(1, 0, 0)$, $(0, 1, 0)$ en $(0, 0, 1)$ respectievelijk.

7.5 Resultaten

Het model geeft maar goede resultaten als alle constanten goed gekozen zijn. Tabel 7.1 bevat de waarden die wij gebruiken.

Ter vergelijking hebben we een cd gefotografeerd en gerenderd vanuit verschillende camerastandpunten (figuur 7.7). Zelfs vanuit uitzonderlijke posities ziet het model er realistisch uit. Als voorbeeld geven we een cd die bijna vanuit de perfect gereflecteerde richting gefotografeerd is (figuur 7.8).

Om ons model te verifiëren hebben we dit ook gedaan met een *digital versatile disc* (dvd) (figuur 7.9). Een dvd heeft een fijnere microstructuur dan een cd: 740 nm tussen de sporen en 1400 nm tussen de putjes.

Listing 7.1: Diffractive BRDF.

```

Vector& wo; // uitgaande richting
Vector* wi; // inkomende richting
Vector x = Vector(1,0,0); // richting x
Vector y = Vector(0,1,0); // richting y
Vector n = Vector(0,0,1); // richting n (normaal)
float dtrack = 1600.f; // afstand tussen de sporen
float dpit = 2400.f; // afstand tussen de putjes
float cdiff, cspec, alpha; // constanten voor de gewone componenten
float c0, c5, beta; // constanten voor de anisotrope component
float c1, c2, c3, c4; // constanten voor de diffractieve componenten
float signal, sigma2; // breedte van de Gauss-curven

// Gewone speculaire en diffuse component
float Rnondiff = cdiff*(Dot(wo, n))
    + cspec*pow(fabsf(Dot(Normalize(wi+wo), n)), alpha);

// Anisotrope speculaire component
float Raniso = c0*exp(c5*pow(fabsf(Dot(wi+wo, x)), beta));

// Hoofdstrip
float temp = dtrack * fabsf(Dot(wi+wo, y));

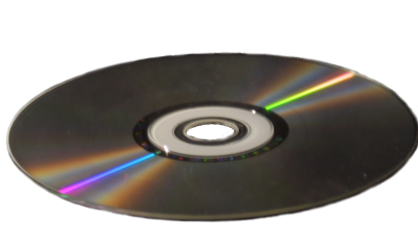
// aantal pieken
int nspikes = (int)(temp/360.f) - (int)(temp/830.f);
if (nspikes < 0) nspikes = 0;
Spike spikes[nspikes];

// berekening van de pieken
int n = 1;
int nb = 0;
float wvl = temp/n;
while (wvl >= 360.f) {
    if (wvl <= 830.f) {
        float tf = 2.f*M_PI/wvl * dpit * Dot(wi+wo, x);
        float f = c1*sqrt(M_PI)/(2.f*sigma2)
            *exp(-4.f*tf*tf/sigma2/sigma2) + c2;
        spikes[nb++] = Spike(wvl, Rnondiff + Raniso*f*c3 + c4);
    }
    wvl = temp/(++n);
}

// Nevenstrips
int nsamples = 40;
float samples[nsamples];
for (int i = 0; i < nsamples; i++) {
    float wvl = 360.f + i*(830.f-360.f)/(nsamples-1.f);
    float tf = 2.f*M_PI/wvl * dpit * Dot(wi+wo, x);
    float f = 0.f;
    for (int j = -3; j <= 3; j++) {
        if (j != 0) {
            float ttf = tf - 2.f*M_PI*j;
            f += exp(-4.f*ttf*ttf/signal/signal);
        }
    }
    f = c1*sqrt(M_PI)*f/(2.f*sigma2) + c2;
    samples[i] = Rnondiff + f*Raniso;
}

SAMPLEDSPECTRUM_PTR sampled =
    SAMPLEDSPECTRUM_PTR(new SampledSpectrum(nsamples, 360.f, 830.f, samples));
Spectrum res;
res.spec = COMPOSITESPECTRUM_PTR(new CompositeSpectrum(sampled, nb, spikes));
return res;

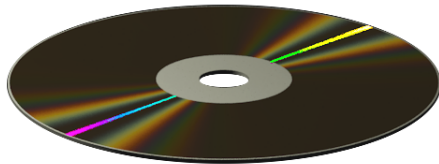
```

(a) Foto 1



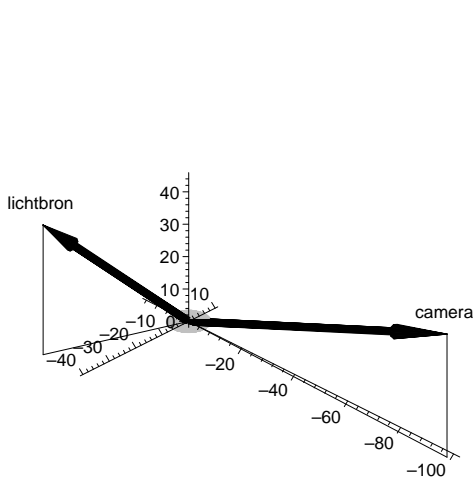
(b) Foto 2



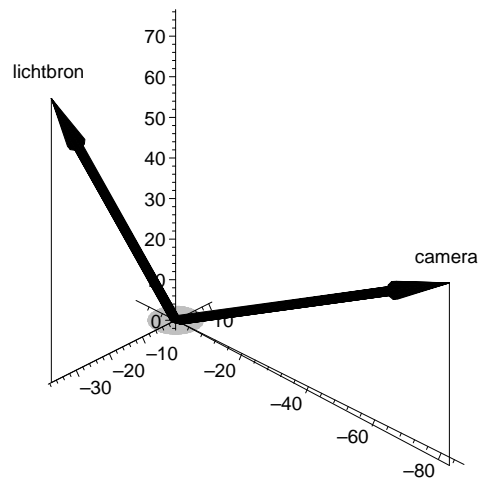
(c) Gerenderd 1



(d) Gerenderd 2



(e) Opstelling 1

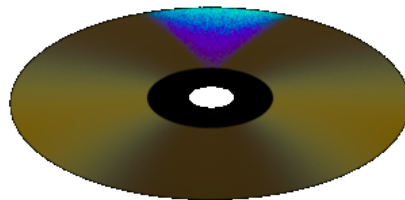


(f) Opstelling 2

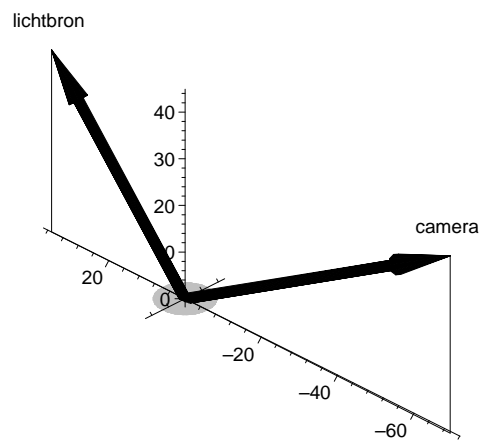
Figuur 7.7: Foto's van een cd en gerenderde cd's met de opgemeten posities van camera en lichtbron.



(a) Foto



(b) Gerenderd



(c) Opstelling

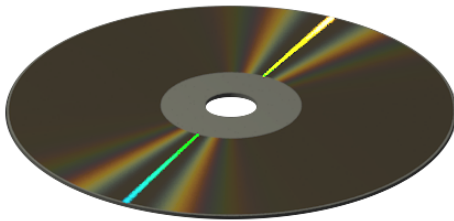
Figuur 7.8: Foto van een cd en gerenderde cd met de opgemeten positie van camera en lichtbron, bijna onder de perfect gereflecteerde richting.



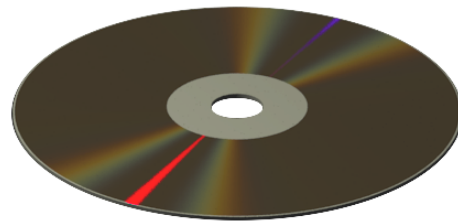
(a) Foto van een cd



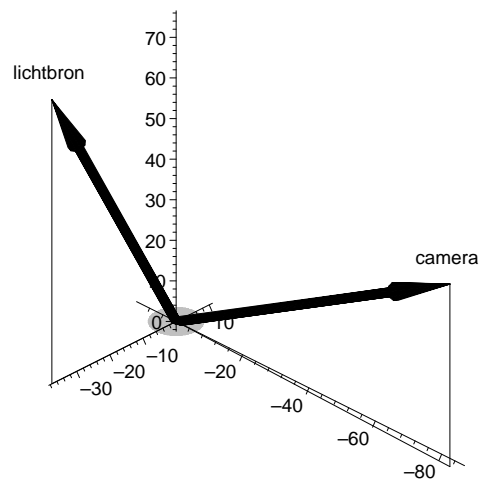
(b) Foto van een dvd



(c) Gerenderde cd



(d) Gerenderde dvd



(e) Opstelling

Figuur 7.9: Foto's van een cd en een dvd, en de gerenderde cd en dvd met de opgemeten posities van camera en lichtbron.

Tabel 7.1: Richtwaarden voor de gebruikte constanten.

constante	waarde	constante	waarde
c_0	0,005	c_{spec}	50,0
c_1	1,0	c_{diff}	0,001
c_2	0,01	α	10000,0
c_3	1000,0	β	2,0
c_4	0,0	σ_1	3,14
c_5	10,0	σ_2	0,4

Dispersie

Eén van de subtielere spectrale effecten die men in het dagelijkse leven tegenkomt, is dispersie. Iedereen herinnert zich ongetwijfeld de proef uit de lessen fysica waar wit licht door een prisma gesplitst wordt in de kleuren van de regenboog. Dispersie is echter meer dan dat. Wanneer men door een glas water kijkt, ziet men soms regenboogkleuren aan de rand van objecten. Bij een projectie met een overheadprojector zie je vaak een rode en blauwe rand (chromatische aberraties van de lens). Ook de kleuren in een diamant zijn dispersieve verschijnselen.

Na het overzicht van het voorgaand werk volgt een beschrijving van de technieken die gebruikt kunnen worden om dispersieve effecten correct te berekenen. Tenslotte worden onze implementatie en de resultaten hiervan weergegeven.

8.1 Voorgaand werk

Om dispersie te simuleren zal men een gevolgde straal aan een oppervlak in verschillende richtingen moeten splitsen. Thomas noemt deze techniek *spread rays* (Thomas 1986).

Dispersie is vaak zichtbaar in brandvlekken. Arvo beschrijft een bijkomende stap in het radiositeitsalgoritme om brandvlekken in de *mesh* op te slaan (Arvo 1986). Hij ontwerpt de techniek van achterwaartse ray tracing, waarbij het licht vanuit de lichtbron naar de scène gevolgd wordt. Hij noemt zijn datastructuur een *illumination map*, onderverdeeld in *patches* zoals bij radiositeit. Uitbreidingen die vooral de dispersie verbeteren, zijn te vinden in (Gartaganis 1992) en (Collins 1994, Collins 1997).

In hoeveel stralen men moet splitsen is een punt van discussie. Meyer argumenteert dat in een aangepaste kleurenruimte 4 stralen voldoen (Meyer 1988), Musgrave gebruikt voor hoge precisie 13 stralen (Musgrave 1989).

Evans en McCool beschrijven *stratified wavelength clusters*, een manier om deze technieken in moderne globale belichtingsalgoritmes zoals bidirectionele path tracing te verwerken (Evans en MCCOOL 1999). Ze geven ook drie manieren om stralen te splitsen aan een dispersief oppervlak:

degradatie: Begin met een volledig spectrum, en beschouw vanaf het oppervlak nog maar één (willekeurige) golflengte.

splitsing: Begin met een volledig spectrum, en beschouw vanaf het oppervlak een aantal aparte golflengtes.

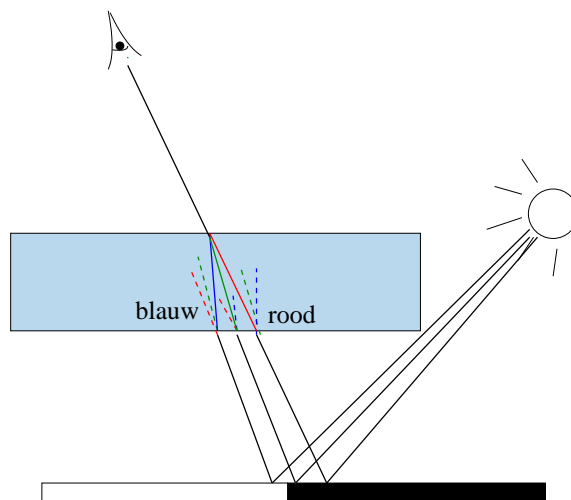
uitstel: Begin met één golflengte, en pas als blijkt dat er op het pad nergens dispersie optrad, berekent men de rest van het spectrum.

Iedere manier heeft voor- en nadelen, maar alledrie versnellen ze de rendering door enkel voor dispersieve paden de golflengte te beschouwen. Ze stoten echter op de beperking van bidirectionele stochastische ray tracing: lichtpaden en zichtpaden moeten op een diffuus oppervlak eindigen om ze te kunnen verbinden (L(S|D)*DD(S|D)*E).

De aanpak van Arvo doet waarschijnlijk wel een belletje rinkelen. Het is een verre voorloper van de eerste stap in foton mapping. De techniek kan eenvoudig aangepast worden voor foton mapping. Er zijn echter nog geen belangrijke vorderingen gepubliceerd specifiek over dispersie in foton mapping. Dezelfde beperking als bij bidirectionele path tracing geldt.

8.2 Directe en indirecte dispersie

Dispersie doet zich voor bij de golflengte-afhankelijke breking van licht (zie sectie 4.1 op pagina 38). Het kan zich in twee situaties voordoen, die elk een licht andere aanpak zullen vereisen bij het renderen van dispersie. Ten eerste kan het zijn dat men rechtstreeks door een dispersief object kijkt (zie ook figuur 8.1). Dit wordt verder directe dispersie genoemd. De lichtstralen komen hierbij eventueel eerst terecht op een oppervlak, waarna ze ons oog bereiken doorheen het dispersieve object. Het dispersieve effect zal het grootst zijn wanneer er een groot verschil is tussen de reflectantie op het pad van bijvoorbeeld de rode en de blauwe straal. In figuur 8.1 gaan de rode en groene straal via een donker oppervlak en de blauwe straal via een lichter oppervlak. Door in de aangegeven richting te kijken, zal dus voornamelijk blauw waargenomen worden. Dit is bijvoorbeeld wat je kan zien wanneer je door een glas naar de letters van een krant kijkt. Als het glas voldoende



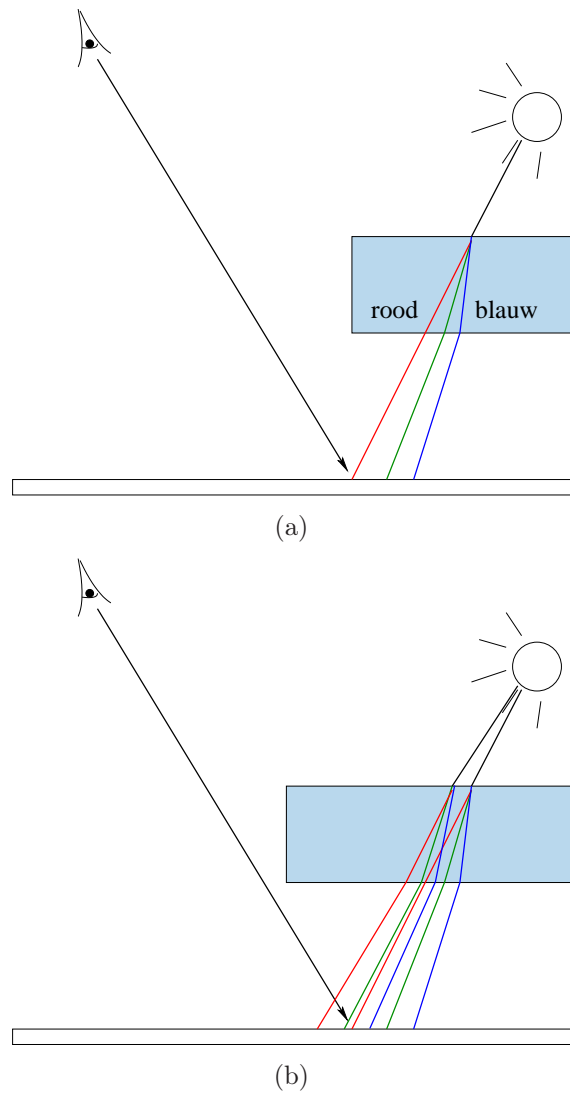
Figuur 8.1: Dispersie bij het kijken doorheen een object (directe dispersie).

dispersief is, zullen er door het grote contrastverschil tussen de letters en het papier aan de rand van de letters kleurovergangen verschijnen.

De tweede manier waarop dispersie voorkomt, is gelijkaardig aan caustics of brandvlekken (zie figuur 8.2). De lichtstralen gaan doorheen een dispersief object en komen op een niet-speculaire ondergrond terecht voor ze ons oog bereiken. Daarom wordt dit indirecte dispersie genoemd. Vermits op de plaats waarnaar men kijkt mogelijk stralen met verschillende golflengtes vanuit verschillende richtingen toekomen, krijgt men een dispersieve brandvlek. Deze zal veelal gekleurde randen hebben (rood en blauw). In het midden komen verschillende golflengtes samen, waardoor het geheel wit wordt. Dit is wat er gebeurt bij een overheadprojector. Wanneer de lichtbundel zeer smal is, kan het zijn dat er geen wit gebied is maar een volledig kleurverloop van rood naar blauw (voor een witte lichtbron).

8.3 Technieken

Zoals reeds vermeld, kan dispersie zich zowel op een directe als indirecte manier voordoen. Voor elk van deze twee manieren zullen twee licht verschillende technieken gebruikt moeten worden, die hieronder behandeld worden. Tenslotte kunnen de technieken ook gecombineerd worden.



Figuur 8.2: Dispersie en weerkaatsing op een oppervlak (indirecte dispersie); (a) met één lichtstraal; (b) met twee lichtstralen.

8.3.1 Directe dispersie

Voor directe dispersie kan men de stralen vanuit het oog door het dispersief object laten breken, zoals dat gebeurt wanneer er geen dispersie is. Dat blijkt uit figuur 8.1: van de drie oorspronkelijke stralen vanuit de lichtbron komt van de eerste straal enkel de blauwe component toe in het oog volgens de richting waarin men kijkt. De rode en groene component van die straal volgen een andere weg. Van de tweede straal komt enkel de groene component toe in ons oog en van de derde straal enkel de rode. Dit kan ook beschouwd worden als onze zichtstraal, die dispersief gebroken wordt door het object. Eén van de mogelijke aanpakken om dispersie te renderen is dan ook om na de breking meerdere monochromatische stralen doorheen de scene te volgen. Deze manier van werken kan meestal eenvoudig toegevoegd worden aan een al dan niet stochastische ray tracer.

Er zijn nog wel enkele vrijheden. Zo moet beslist worden hoeveel gebroken stralen er gebruikt zullen worden. Te weinig stralen leveren fouten op: er moeten voldoende monochromatische stralen gevolgd worden om uiteindelijk accurate spectra te bekomen. Te veel stralen leiden echter tot lange rekestijden. Ook het kiezen van de golflengtes van de stralen ligt niet vast. Een mogelijkheid is een equidistante verdeling te kiezen over het volledige zichtbare gebied. Beter is een stochastische gestratificeerde bemonstering van het spectrum. Wanneer op het moment van bemonsteren het spectrum tot dan toe al gekend is, is het aangewezen importance sampling te gebruiken op dat spectrum. Stel bijvoorbeeld dat er een rode filter tussen het oog en het dispersief object staat. Wanneer de zichtstraal bij het object komt, weet men op dat moment reeds dat enkel het rode gebied een bijdrage zal leveren. Het is dan zinloos om het blauwe gebied nauwkeurig te bemonsteren.

8.3.2 Indirecte dispersie

De aanpak van directe dispersie kan niet als zodanig gebruikt worden voor indirecte dispersie. Het probleem is analoog aan het probleem met brandvlekken: de richtingen via dewelke het licht van de lichtbron doorheen het dispersief object op een oppervlak terecht komt, zijn niet gekend (figuur 8.2(b)). De efficiëntste manier om dit probleem op te lossen, is te vertrekken vanaf de lichtbron, bijvoorbeeld met foton mapping of bidirectionele ray tracing. De stralen vanuit de lichtbron kunnen dan dispersief gebroken worden. Zoals bij de directe dispersie moet ook hier een keuze gemaakt worden voor het aantal stralen en de golflengtes. Importance sampling volgens het spectrum van de lichtbron kan ook hier nuttig zijn.

8.3.3 Combinatie

Een algoritme zoals bidirectionele ray tracing of foton mapping waarbij zowel vanuit het oog als vanuit de lichtbron(nen) vertrokken wordt met het volgen van lichtstralen, kan aangepast worden zodat het beide vormen van dispersie aankan. De techniek voor directe dispersie wordt toegevoegd aan het deel dat stralen schiet vanuit het oog. De indirecte dispersie wordt dan ingewerkt in het deel dat stralen vanuit een lichtbron schiet.

8.4 Implementatie

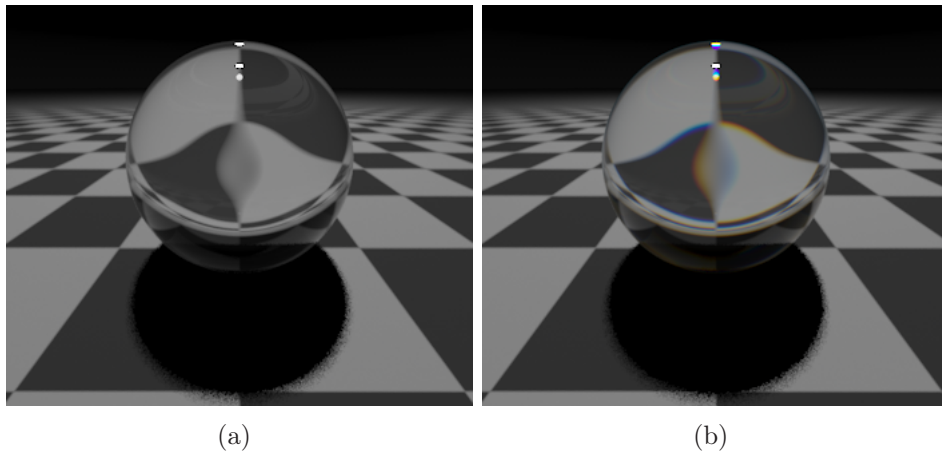
De implementatie van dispersie in PBRT vereist een aanpassing aan de interface van de BRDF. De methode voor het bemonsteren van een BRDF heeft bij dispersie eventueel extra informatie nodig. Wanneer men na een eerste dispersieve breking een monochromatische straal volgt, is het essentieel dat vanaf dan enkel de bijdrage voor de golflengte van de straal in rekening gebracht wordt. Meer nog, als bij een tweede dispersieve breking van de straal opnieuw een golflengte gekozen wordt, zal de totale bijdrage van de straal (het product van de bijdragen van elk deel van het pad) gelijk zijn aan 0, omdat de golflengtes van de verschillende delen van het pad dan niet noodzakelijk hetzelfde zijn.

Omdat ons spectraal raamwerk niet de mogelijkheid biedt om importance sampling uit te voeren op een spectrum, wordt steeds het volledige zichtbare gebied bemonsterd. De monsters over het spectrum zijn ook equidistant verdeeld. Een aanpassing naar stochastisch bemonsterde golflengtes is niet moeilijk, maar kon wegens tijdsbrek niet meer uitgevoerd worden.

8.5 Resultaten

In figuur 8.3 wordt een bol op een dambordpatroon afgebeeld zonder en met dispersie. In het midden van de bol is het duidelijkst een kleurverloop te zien, maar ook aan de randen van de tegels die zichtbaar zijn in de bol en aan de rand van de reflecties van de lichtbron bevindt er zich een kleurverloop. Dit is een voorbeeld van directe dispersie.

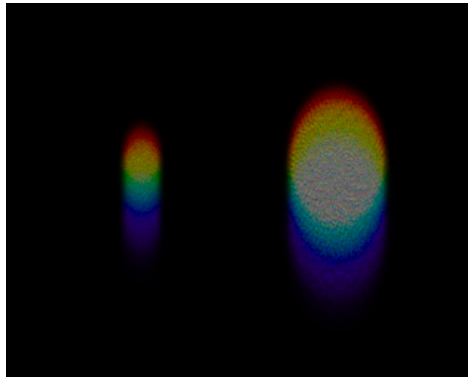
In figuur 8.4 staat een voorbeeld van indirecte dispersie. Dit zijn de caustics die door een prisma en twee parallelle lichtbundels (een smalle links en een brede rechts) op een oppervlak gegenereerd werden. In de linker brandvlek verlopen de kleuren van rood via groen naar blauw. Bij de bredere



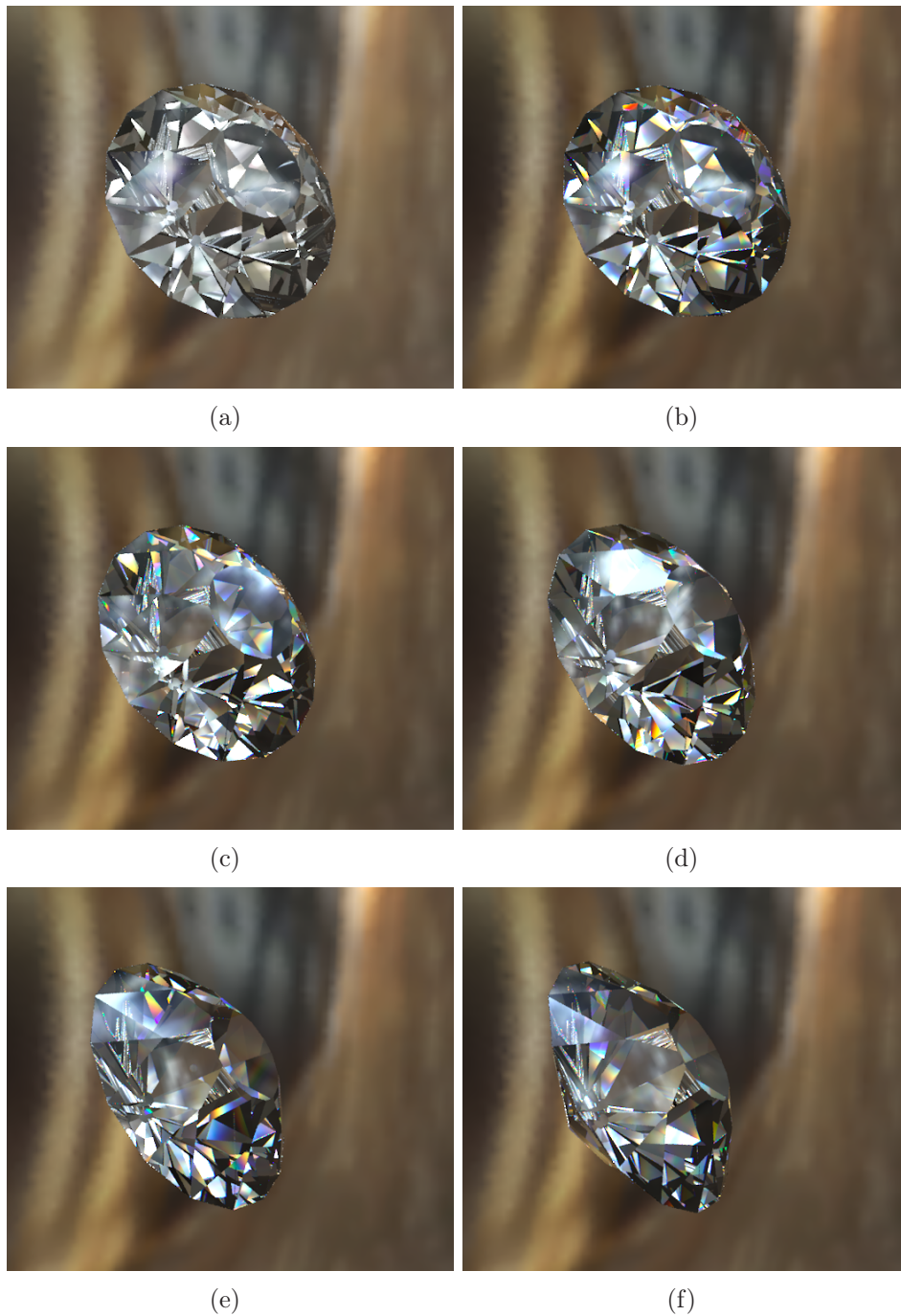
Figuur 8.3: Glazen bol (a) zonder dispersie, (b) met dispersie.

bundel overlappen er meerdere kleuren in het midden waardoor dit wit wordt.

Figuur reffig:diamond bevat enkele beelden uit een animatie van een roterende diamant. In 8.5(a) wordt de diamant in de beginpositie weergegeven zonder dispersie. Figuur 8.5(b) bevat de diamant in dezelfde positie, maar met dispersie. De volgende figuren tonen de diamant met dispersie geroteerd over verschillende hoeken.



Figuur 8.4: Dispersieve caustics. Links: smalle lichtbundel. Rechts: bredere lichtbundel.



Figuur 8.5: (a) Geen dispersie. (b-f) Dispersie in een roterende diamant.

Absorptie in volumes

Licht dat doorheen een transparant medium reist, wordt meestal gedeeltelijk geabsorbeerd. Hoe groter de afstand is die het licht doorheen het medium aflegt, hoe sterker de absorptie. Zo is de vloeistof in de hals van een fles meestal lichter van kleur dan in de buik van de fles.

In dit hoofdstuk wordt, na het voorgaand werk, de theorie behandeld die absorptie van licht beschrijft. Daarbij wordt even stilgestaan bij het gebruik van RGB bij het renderen. Vervolgens wordt onze implementatie toegelicht, gevolgd door enkele resultaten.

9.1 Voorgaand werk

Er zijn vele publicaties over het renderen van volumes algemeen. Ze beschrijven vooral het fysisch model van scattering aan stofdeeltjes in wolken (Rushmeier en Torrance 1987), en hoe het efficiënt kan gerenderd worden (Jensen en Christensen 1998).

Er zijn slechts enkele gerelateerde werken specifiek over realistische absorptie. Dat komt omdat een absorptiecoëfficiënt in het RGB-model geen fysische betekenis heeft (Hall 1989). De absorptiecoëfficiënt wordt in RGB meestal proefondervindelijk geschat door een visuele vergelijking.

Sun *et al.* bespreken een spectrale aanpak van absorptie met opgemeten absorptiecoëfficiënten (Sun *et al.* 1999a, Sun 2000). Ze maken de vergelijking door de opgemeten coëfficiënt naar RGB te converteren. Uit hun resultaten blijkt overduidelijk dat de berekeningen met RGB in dit geval helemaal niet dezelfde betekenis hebben als de fysisch gebaseerde spectrale methode.

9.2 Theorie

De absorptie van een medium wordt beschreven door de absorptiecoëfficiënt $\sigma_a(x \rightarrow \hat{\omega}, \lambda)$. Deze geeft de kans aan waarmee licht van een bepaalde golflengte λ dat in punt x in een richting $\hat{\omega}$ reist over een eenheidsafstand, geabsorbeerd wordt. Wanneer het medium isotroop is voor absorptie, is σ_a onafhankelijk van de richting $\hat{\omega}$. Dat is meestal het geval. In een homogeen medium is σ_a onafhankelijk van de positie x .

Beschouw een lichtstraal met radiantie $L(x \rightarrow \hat{\omega}, \lambda)$ die in een absorberend medium een elementaire afstand ds aflegt vanaf het punt x in een richting $\hat{\omega}$. De radiantie na de absorptie is dan $L(x + ds \hat{\omega} \rightarrow \hat{\omega}, \lambda)$ en wordt gegeven door

$$L(x + ds \hat{\omega} \rightarrow \hat{\omega}, \lambda) = L(x \rightarrow \hat{\omega}, \lambda) - \sigma_a(x \rightarrow \hat{\omega}, \lambda)L(x \rightarrow \hat{\omega}, \lambda)ds. \quad (9.1)$$

Het verschil in radiantie $dL(x \rightarrow \hat{\omega}, \lambda)$ is bijgevolg

$$\begin{aligned} dL(x \rightarrow \hat{\omega}, \lambda) &= L(x + ds \hat{\omega} \rightarrow \hat{\omega}, \lambda) - L(x \rightarrow \hat{\omega}, \lambda) \\ &= -\sigma_a(x \rightarrow \hat{\omega}, \lambda)L(x \rightarrow \hat{\omega}, \lambda)ds \end{aligned} \quad (9.2)$$

Voor de absorptie tussen twee punten x en $y = x + d\hat{\omega}$ (op een afstand d van elkaar volgens de richting $\hat{\omega}$) levert dit:

$$L(y \rightarrow \hat{\omega}, \lambda) = L(x \rightarrow \hat{\omega}, \lambda) \exp\left(-\int_0^d \sigma_a(x + s\hat{\omega} \rightarrow \hat{\omega}, \lambda)ds\right) \quad (9.3)$$

Voor een isotroop en homogeen medium herleidt dit zich tot

$$L(y \rightarrow \hat{\omega}, \lambda) = L(x \rightarrow \hat{\omega}, \lambda)e^{-\sigma_a(\lambda)d}. \quad (9.4)$$

Vergelijking (9.4) is algemeen bekend als de wet van Beer.¹

Absorptie en RGB

Bij het implementeren van absorptie in een RGB-model wordt de absorptiecoëfficiënt $\sigma_a(\lambda)$ vaak benaderd door drie coëfficiënten, σ_r , σ_g en σ_b voor

¹Deze wet werd onafhankelijk afgeleid door Bouguer, Lambert en Beer. De wetten van Lambert en Beer waren oorspronkelijk gedefinieerd voor homogeen transparante oplossingen met een lage concentratie, waar de wet van Bouguer onder alle omstandigheden geldt. Het verschil tussen beide is echter niet relevant voor de verdere discussie.

respectievelijk de rode, groene en blauwe component. Het lijkt logisch om voor de drie coëfficiënten de RGB-waarden te nemen die bij $\sigma_a(\lambda)$ horen. Dus

$$\begin{aligned}\sigma_r &= \int_{-\infty}^{\infty} \sigma_a(\lambda) \bar{r}(\lambda) d\lambda \\ \sigma_g &= \int_{-\infty}^{\infty} \sigma_a(\lambda) \bar{g}(\lambda) d\lambda \\ \sigma_b &= \int_{-\infty}^{\infty} \sigma_a(\lambda) \bar{b}(\lambda) d\lambda,\end{aligned}\tag{9.5}$$

met $\bar{r}(\lambda)$, $\bar{g}(\lambda)$ en $\bar{b}(\lambda)$ de color matching functies (horende bij testspectra $R(\lambda)$, $G(\lambda)$ en $B(\lambda)$) voor het RGB-model.

Men berekent dan (voor een homogeen medium) de uiteindelijke kleur (S_r, S_g, S_b) na absorptie over een afstand d als

$$(S_r, S_g, S_b) = (I_r e^{-\sigma_r d}, I_g e^{-\sigma_g d}, I_b e^{-\sigma_b d})\tag{9.6}$$

met I_r , I_g en I_b de rode, groene en blauwe component van de kleur vóór de absorptie (analoog aan 9.5). De drie coëfficiënten S_r , S_g en S_b beschrijven alle spectra $S(\lambda)$ die metameer zijn met $S_r R(\lambda) + S_g G(\lambda) + S_b B(\lambda)$. Of dus,

$$\begin{aligned}S(\lambda) &\cong S_r R(\lambda) + S_g G(\lambda) + S_b B(\lambda) \\ &= I_r e^{-\sigma_r d} R(\lambda) + I_g e^{-\sigma_g d} G(\lambda) + I_b e^{-\sigma_b d} B(\lambda)\end{aligned}$$

Gebruik makende van (9.5) en de definitie van I_r , I_g en I_b vinden wordt dit voor de RGB-benadering dus dat

$$\begin{aligned}S(\lambda) &\cong R(\lambda) \exp\left(-\int_{-\infty}^{\infty} \sigma_a(\lambda) \bar{r}(\lambda) d\lambda\right) \int_{-\infty}^{\infty} I(\lambda) \bar{r}(\lambda) d\lambda \\ &+ G(\lambda) \exp\left(-\int_{-\infty}^{\infty} \sigma_a(\lambda) \bar{g}(\lambda) d\lambda\right) \int_{-\infty}^{\infty} I(\lambda) \bar{g}(\lambda) d\lambda \\ &+ B(\lambda) \exp\left(-\int_{-\infty}^{\infty} \sigma_a(\lambda) \bar{b}(\lambda) d\lambda\right) \int_{-\infty}^{\infty} I(\lambda) \bar{b}(\lambda) d\lambda.\end{aligned}\tag{9.7}$$

De correcte spectrale benadering levert echter dat

$$\begin{aligned}S(\lambda) &= I(\lambda) e^{-\sigma_a(\lambda) d} \\ &\cong R(\lambda) \int_{-\infty}^{\infty} I(\lambda) e^{-\sigma_a(\lambda) d} \bar{r}(\lambda) d\lambda \\ &+ G(\lambda) \int_{-\infty}^{\infty} I(\lambda) e^{-\sigma_a(\lambda) d} \bar{g}(\lambda) d\lambda \\ &+ B(\lambda) \int_{-\infty}^{\infty} I(\lambda) e^{-\sigma_a(\lambda) d} \bar{b}(\lambda) d\lambda.\end{aligned}\tag{9.8}$$

Als het RGB-model correct wil zijn, zal het spectrum $S(\lambda)$ uit (9.7) hetzelfde moeten zijn als dat in (9.8). Dit is immers het correcte fysische spectrum waarmee het RGB-resultaat metameer moet zijn om ook correct te zijn. Bij een nadere inspectie van de twee vergelijkingen, blijkt dat beide vergelijkingen een metameer beschrijven voor het spectrum $S(\lambda)$, opgebouwd uit een lineaire combinatie van drie identieke basisfuncties ($R(\lambda)$, $G(\lambda)$ en $B(\lambda)$). Het is dan noodzakelijk dat de coëfficiënten van de drie basisfuncties ook gelijk zijn. Uit de vergelijkingen blijkt echter dat dit algemeen niet het geval is. Ook voor andere manieren om σ_r , σ_g en σ_b te bepalen (bijvoorbeeld als de waarden van $\sigma_a(\lambda)$ voor drie discrete golflengtes), zullen de coëfficiënten niet altijd hetzelfde zijn. Hieruit kan besloten worden dat met het RGB-model geen correcte absorptie berekend kan worden.

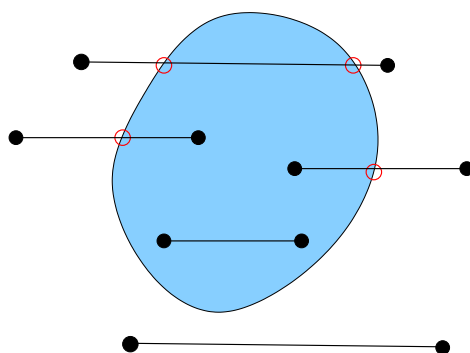
Het voorgaande wil niet zeggen dat men met RGB geen afbeelding kan maken die er hetzelfde uitziet als de spectrale. Men kan, voor een bepaalde $\sigma_a(\lambda)$ en een bepaalde afstand d waarden vinden voor σ_r , σ_g en σ_b zodanig dat het uiteindelijke RGB-spectrum wel metameer is met het correcte spectrum. Deze waarden zullen dan echter niet gelden voor alle afstanden d .

9.3 Implementatie

In PBRT is absorptie geïmplementeerd voor participerende media (Volume Regions). Dergelijke volumes waren oorspronkelijk gelimiteerd tot een (vershoven, geroteerde en/of gescaleerde) op de hoofdassen uitgelijnde balk. Andere vormen kon men eventueel enkel bekomen door de parameters op de punten van een driedimensionaal rooster op te geven.

Om op een praktische manier een willekeurige vorm te gebruiken als grens voor een absorberend medium, werd de oorspronkelijke implementatie aangepast. Dit is enkel geïmplementeerd voor een homogeen volume, maar zou zonder veel aanpassing ook kunnen gebruikt worden bij de andere volumes.

Om over te gaan naar een willekeurige vorm (bijvoorbeeld een bol, kegel of mesh), moesten hoofdzakelijk twee dingen aangepast worden. Ten eerste het invoerformaat en de interface voor het aanmaken van een volumegebied, omdat dit enkel voorzien was voor een balk. Belangrijker echter is de manier waarop de afstand doorheen het medium bepaald wordt van een straal, of meer specifiek het deel van de straal met $t \in [t_{\min}, t_{\max}]$. Vroeger werd dit bepaald door de doorsnede van het beschouwde deel van de straal met het balkvormige bounding volume te zoeken. Dat kon zeer efficiënt gebeuren. Nu moeten echter de punten gezocht worden waar de straal in en uit het willekeurige object gaat. Er moet, net zoals bij de balkvormige volumes, ook



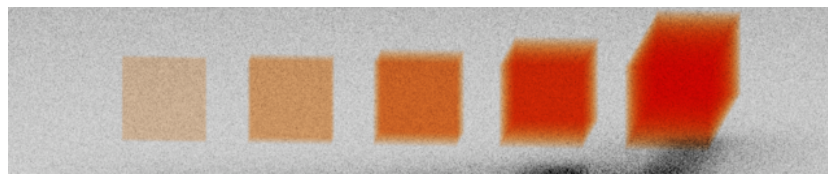
Figuur 9.1: Verschillende situaties bij het bepalen van de lengte van de overlapping van een deel van een straal met een volume.

rekening gehouden worden met het deel van de straal dat we beschouwen. Begin- en eindpunt kunnen beiden buiten het volume liggen, beiden erbinnen of één erbuiten en één erbinnen (figuur 9.1). De interface van PBRT laat echter niet toe meerdere intersecties van een straal en een willekeurig object in één keer te bepalen. Onze nieuwe implementatie doet dit daarom zelf.

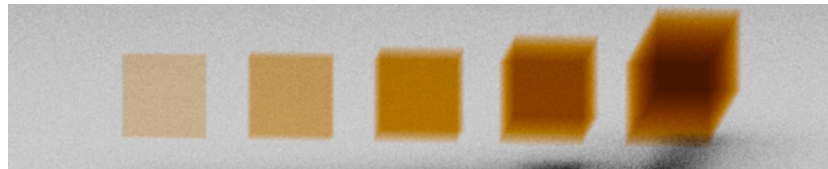
Men zou kunnen aanbrengen dat een dergelijke aanpak teveel werk is voor het zuiver implementeren van absorptie in homogene volumes. Het lijkt eenvoudiger en zinvoller om een nieuw materiaal te definiëren dat deze absorptie in rekening brengt bij de shading. Inderdaad, men zou bij het verlaten van een absorberend object de afstand tot de oorsprong van de straal kunnen nemen en het spectrum vermenigvuldigen met de exponentiële factor uit (9.4). Er hoeft dan geen participierend medium of volume integrator gebruikt te worden. We zijn echter van mening dat onze aanpak meer voordelen biedt: niet-homogene media kunnen op dezelfde manier aangepakt worden, waar dat met de eenvoudige implementatie niet meer kan. Ook is het in de huidige implementatie van PBRT niet mogelijk om bij de shading de oorsprong van de straal of de afstand tot de oorsprong op te vragen. Tenslotte heeft onze aanpak het mogelijk gemaakt om participerende media te definiëren aan de hand van een willekeurige vorm.

9.4 Resultaten

In figuur 9.2 wordt een vergelijking gegeven tussen absorptie via RGB en via een spectraal model. Het is duidelijk te zien dat het RGB-resultaat sterk afwijkt van het spectrale resultaat. De reden daarvoor werd in sectie 9.2 uitvoerig besproken. In de spectrale figuur is ook duidelijk te zien hoe de kleur



(a)

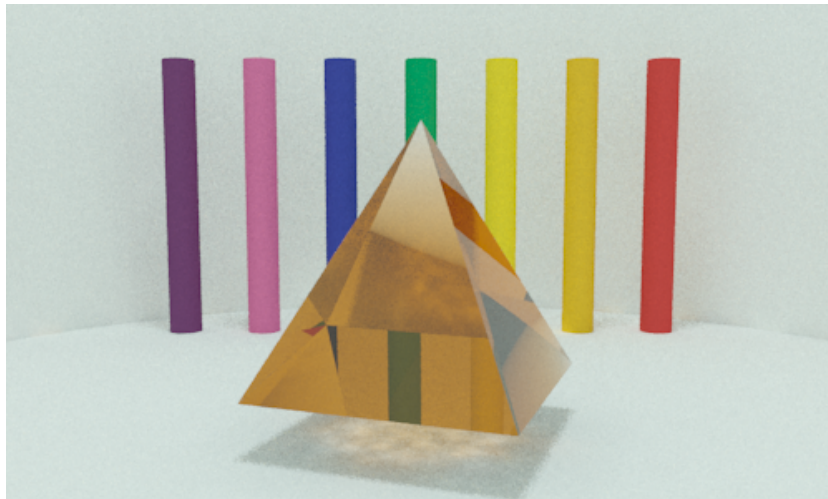


(b)

Figuur 9.2: Absorptie in steeds dikker wordende balken (a) met het RGB-kleurmodel; (b) met een spectraal model (uniform bemonsterd, 31 monsters tussen 400 en 700 nm).

donkerder wordt met toenemende dikte. Het midden van de uiterst rechtse balk is veel donkerder dan de zijkanten en dan de andere (dunnere) balken. (Deze figuur werd met een orthografische projectie gemaakt in plaats van een perspectiefprojectie, om de verschillende balken onvervormd te kunnen bekijken.)

Figuur 9.3 bevat een piramide met een amberkleurige homogene absorptie. De top van de piramide is veel lichter dan de basis. Deze figuur toont ook het gebruik van willekeurige vormen als volume.



Figuur 9.3: Absorptie in een glazen piramide. Merk op dat door de absorptie de top veel lichter gekleurd is dan de basis.

Opgemeten BRDF's

Vele complexe materialen kunnen niet realistisch weergegeven worden met een eenvoudige analytische BRDF. Als alternatief kan men een materiaalmodel opmeten met een gonioreflectometer, een toestel dat op verschillende inkomende en uitgaande richtingen de reflectantie meet.

De data kunnen dan gebruikt worden om een analytisch model zoals de Lafortune-BRDF te fitten. Zo kan men de hoeveelheid gegevens beperken voor matig complexe materialen zoals matte verf, baksteen, huid, ...

Voor de meest complexe materialen, zoals glanzende verf en autolak, moet men de ruwe data bijhouden en bij iedere evaluatie van de BRDF interpoleren. Als men bovendien beschikt over spectrale meetwaarden, kan men in principe alle mogelijke materialen onder iedere belichting correct weergeven.

10.1 Voorgaand werk

De meeste traditionele BRDF's zijn bij het ontwerp of achteraf gestaafd met meetwaarden van echte materialen. De voorgestelde modellen zijn echter zelden geschikt voor de complexe materialen die verder behandeld worden. Meestal gebeuren de metingen ook met RGB-kleuren en een beperkt aantal inkomende en uitgaande richtingen, om geheugenruimte en werk te besparen.

Men kan voor de meeste materialen wel een microstructuur modelleren en er een reflectantiefunctie uit afleiden. Dit is echter veel werk voor één specifiek materiaal. Cabral *et al.* leiden als eersten een BRDF af van een willekeurig concreet microfacetmodel, met een techniek gerelateerd aan bump mapping (Cabral *et al.* 1987). Men modelleert de microfacetten, en men berekent de gemiddelde reflectantie van een oppervlak door Monte Carlo integratie. De verkregen data kunnen in ruwe vorm bijgehouden worden, maar in die tijd was het zeker nog aangeraden om er een eenvoudiger model op te fitten.

Westin *et al.* breiden die techniek uit voor fysisch correcte modellering, ook voor complexere anisotrope materialen (Westin *et al.* 1992). Hanrahan en Krueger passen deze methode toe op subsurface scattering in een ge-laagd materiaal (Hanrahan en Krueger 1993). Gondek *et al.* modelleren o.a. glanzende verfsoorten aan de hand van een complexe microstructuur (Gondek *et al.* 1994). Ze beschouwen een laag van bindmiddel met daarin een verdeling van pigmentpartikels. Hun datastructuur is een adaptief verfijnde geodetische hemisfeer, om belangrijke smalle lobben toch met voldoende resolutie te bemonsteren.

Lafortune *et al.* beschrijven een analytische BRDF die voldoet aan de fysische vereisten (wederkerigheid, behoud van energie) en tegelijk gemakkelijk te fitten is op meetwaarden (Lafortune *et al.* 1997). Ze maken een fysisch correcte veralgemening van de Phong-lob en ze nemen de som van een drietal lobben om de meeste materialen te visualiseren. Merk op dat die lobben geen lineaire basisfuncties zijn. In voorgaande modellen werden telkens lineaire combinaties van allerlei basisfuncties gebruikt (vaak sferische harmonischen en wavelets).

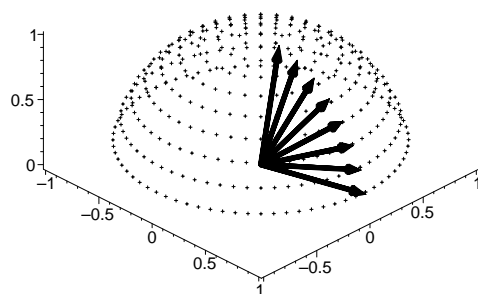
Marschner *et al.* gebruiken een beeldgebaseerde techniek om een materiaal op te meten (Marschner *et al.* 1999). Met een foto van een gekend voorwerp kan men vele waarden tegelijk opmeten. Een lichtbron beschijnt verschillende plaatsen op het voorwerp onder een andere hoek en een foto vangt de reflectie op ieder van die plaatsen op. Na enige verwerking beschikt men over een grote hoeveelheid data. Een beeldgebaseerde techniek houdt wel in dat de data als RGB-kleuren worden opgemeten.

10.2 Meetgegevens

De bibliotheek van meetgegevens van Lafortune, Marschner en hun collega's is beschikbaar op de website van het Cornell University Light Measurement Laboratory.¹ De metingen gebeurden met een geautomatiseerde gonioreflectometer om binnen redelijke tijd voldoende metingen te kunnen doen. Hoewel het model van Lafortune enkel uitgewerkt is voor RGB, werden een aantal metingen toch spectraal uitgevoerd.

De spectrale metingen zijn genomen op 8 inkomende hoeken (10° tot 80°), 180 uitgaande richtingen uniform verdeeld over een halve hemisfeer en telkens op 31 golflengtes (400 tot 700 nm). Door de wederkerigheid van de BRDF en de symmetrie van isotrope materialen kan het aantal metingen enigszins

¹<http://www.graphics.cornell.edu/research/measure/>



Figuur 10.1: Posities van de inkomende richtingen (pijlen) en uitgaande richtingen (puntjes) op de hemisfeer.

beperkt worden, maar op figuur 10.1 blijkt toch duidelijk dat metingen van die omvang enkel met een geautomatiseerd proces kunnen gedaan worden.

Voor de evaluatie van een BRDF wordt er geïnterpoleerd tussen de k dichtste burens van de gezochte richtingen. Een meerdimensionale lineaire interpolatie blijkt goed te werken.

De gebruikte materialen hebben een smalle piek in de perfect gereflecteerde richting. Het is met dit soort meettechnieken te moeilijk om het gebied van die piek met voldoende resolutie te bemonsteren. Daarom is de piek expliciet uit de metingen weggelaten. Bij het renderen moet men nog een aparte Fresnel-reflectie berekenen. Voor de brekingsindex n in de formule van Fresnel wordt bijvoorbeeld de brekingsindex van het bindmiddel van de verf gebruikt.

10.3 Implementatie

De implementatie bestaat enkel uit een BRDF (listing 10.1). De meetpunten zijn beschikbaar in een niet-gesorteerde rij. De dataset is meestal beperkt tot ongeveer 1500 punten. Voor grotere datasets kan het zoeken naar de k dichtste burens versneld worden met een boomstructuur.

Er is een optie toegevoegd om eenvoudig resultaten te kunnen vergelijken tussen RGB en spectra. Dezelfde spectrale dataset wordt gebruikt, maar het resultaat wordt omgezet naar RGB. De omzetting mag voor of na de interpolatie gebeuren, want dat is een lineaire bewerking die het metamere karakter behoudt.

Listing 10.1: SampledBDF.

```

Vector& wo;                // uitgaande richting
Vector* wi;                // inkomende richting
SampledInfo* data;        // meetpunten (positie en waarde)
int datalen;              // aantal meetpunten
int nb = min(8, datalen);  // gewenst aantal dichtste buren
NeighborSampledInfo neighbors[nb]; // voorlopig geselecteerde meetpunten met afstand
float d;                  // afstand tot het dichtste geselecteerde meetpunt
double factors[nb];       // interpolatiegewichten van de meetpunten
double sum;               // som van de interpolatiegewichten
boolean usergb;          // resultaat in RGB gewenst

// Selecteer de eerste nb meetpunten
for(unsigned int i = 0; i < nb; i++) {
    d = min(data[i].dist(wo, wi), data[i].dist(wi, wo));
    neighbors[i] = NeighborSampledInfo(&(data[i]), d);
}

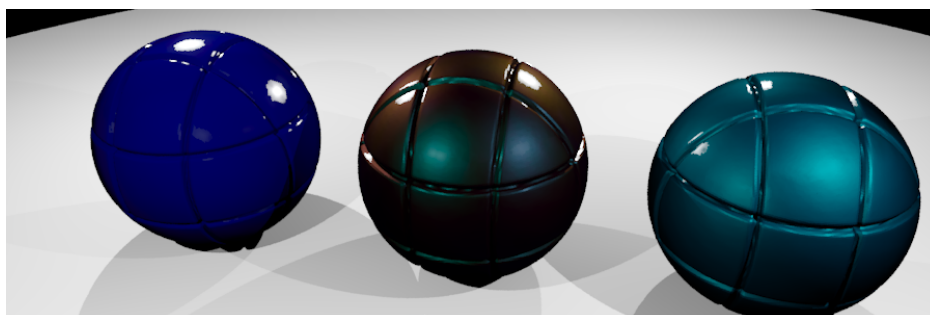
// Sorteert de geselecteerde meetpunten volgens afstand
sort(neighbors, neighbors+nb);

// Vervang de verste geselecteerde buur door een dichtere meetpunt
// en houd de rij gesorteerd
for (int i = nb; i < datalen; i++) {
    d = min(data[i].dist(wo, wi), data[i].dist(wi, wo));
    if(d < neighbors[nb-1].distance) {
        neighbors[nb-1] = NeighborSampledInfo(&(data[i]), d);
        sort(neighbors, neighbors+nb);
    }
}

// Interpoleer tussen de definitief geselecteerde buren
for(unsigned int i = 0; i < nb; i++) {
    factors[i] = 1.;
    for(unsigned int j = 0; j < nb; j++) {
        if(i!=j) {
            factors[i] *= neighbors[j].distance;
        }
    }
    sum += factors[i];
}
Spectrum result = Spectrum(0.);
for(unsigned int i = 0; i < nb; i++) {
    result += Spectrum(neighbors[i].info->samples, 31, 400.f, 700.f)*(factors[i]/sum);
}

// Converteer het resultaat naar RGB indien gewenst
if (usergb) {
    float rgb[3];
    result.RGB(rgb);
    return Spectrum(rgb, true);
} else {
    return result;
}

```

Figuur 10.2: Drie gegroefde bollen, v.l.n.r. blauwe acryl (plastic), Mystic autolak, Cayman autolak.

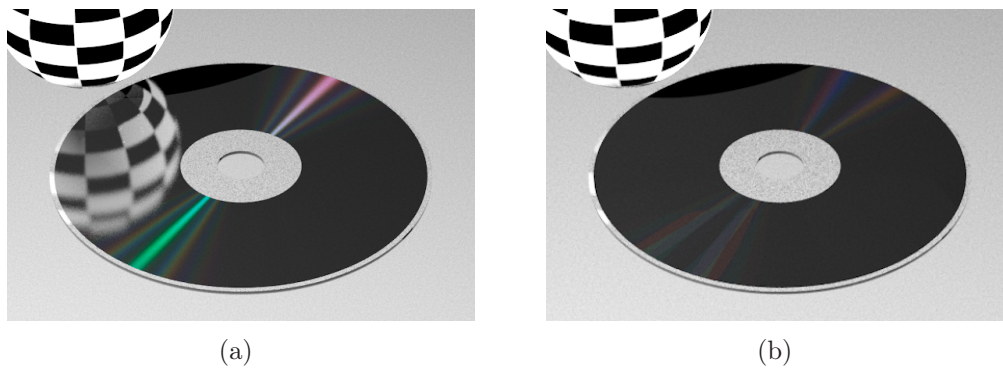
10.4 Resultaten

Figuur 10.2 toont drie gegroefde bollen met materialen waar duidelijk het nut van opgemeten BRDF's blijkt. Vooral de middelste bol heeft een opmerkelijke autolak, die van kleur verandert afhankelijk van de hoek waaronder men ernaar kijkt.

De afbeelding met RGB ziet er echter bijna exact hetzelfde uit. RGB werkt voor deze complexe materialen niet slechter dan voor eenvoudige materialen. Men komt in het dagelijkse leven weinig materialen tegen die heel duidelijk voordeel hebben bij een spectrale aanpak, behalve dan materialen met een golflengte-afhankelijk effect.

Zoals vermeld is de techniek niet geschikt voor de scherpe piek van de perfecte reflectie. Ook bij andere pieken, zoals pieken in de golflengte, zal deze methode falen. Als voorbeeld werd de compact disc uit hoofdstuk 7 gerenderd met een opgemeten BRDF. De meting gebeurde virtueel door de analytische BRDF van de cd te bemonsteren. Op deze manier hoeven we geen gonioreflectometer te gebruiken en kunnen we zelfs veel meer meetpunten nemen. Dat is ook nodig, omdat de kijkrichting belangrijk wordt bij een anisotroop materiaal.

Ondanks de meer dan 200 000 meetpunten die berekend werden, toont figuur 10.3 maar een pover resultaat. De reflectie van de bol verdwijnt, zoals voorspeld, maar die zou eenvoudig terug kunnen toegevoegd worden. De hoofdstrip, opgebouwd uit enkel pieken van een composietspectrum, verdwijnt ook volledig. De nevenstrips, een som van redelijk smalle Gauss-curves, ontwaarden in een aantal gekleurde vegen. Dit zou men met een reële meting gemakkelijk kunnen oplossen, aangezien de meeste spectroradiometers een volledig spectrum tegelijk kunnen opmeten, en niet aparte golfleng-



Figuur 10.3: De compact disc gerenderd (a) met de analytische BRDF uit hoofdstuk 7, en (b) met de virtueel opgemeten BRDF.

tes. Men moet dan de metingen opslaan in een composietmodel en ook correct interpoleren tussen de spectrale pieken.

Implementatie

In dit hoofdstuk wordt een overzicht gegeven van de geïmplementeerde elementen. Ook worden er enkele opmerkingen en problemen vermeld die we bij de implementatie zijn tegengekomen. Er wordt ook even stilgestaan bij de performantie van de spectrale aanpak. Tenslotte wordt een overzicht gegeven van welke beslissingen bij het compileren genomen worden, en hoe deze aangepast kunnen worden.

11.1 Overzicht

De implementatie bestaat uit twee grote delen: het spectraal raamwerk enerzijds en de integratie en uitbreiding van de bestaande raytracer anderzijds.

Het spectraal raamwerk bevat twee belangrijke subsystemen: de verschillende voorstellingswijzen voor spectra en de conversie daartussen. Deze onderdelen werden reeds besproken in sectie 3.2 op pagina 33.

Ons oorspronkelijk idee was om het raamwerk te integreren met RenderPark/XRML¹. Omdat dit echter een enorm groot systeem was waarin we ons moesten inwerken, onze implementatie inpassen en het bovendien bijna onmogelijk bleek eenvoudig te debuggen, zijn we overgestapt naar een kleinere maar uiterst zorgvuldig gedocumenteerde fysisch gebaseerde raytracer, namelijk PBRT. Deze raytracer hoort bij het boek *Physically Based Rendering* (Pharr en Humphreys 2003). Het raamwerk integreren met RenderPark/XRML lijkt ons echter nog steeds mogelijk.

De integratie met PBRT bestaat erin om de originele implementatie van een spectrum van PBRT te vervangen door een verwijzing naar een spectrum uit het raamwerk. Hierbij is het belangrijk om de oorspronkelijke interface van PBRT te behouden, zodat de andere bestaande code niet aangepast dient te worden. Bovendien dient hier ook de conversie naar het juiste resultaattype

¹<http://www.renderpark.be>

uitgevoerd te worden, zoals beschreven in sectie 3.2.2.

Verder hebben we enkele nieuwe materialen geïmplementeerd, zoals een diffractief materiaal (de compact disc), een dunne film en dispersief glas. Het homogeen volume dat reeds bestond, werd gevoelig aangepast en uitgebreid voor het berekenen van absorptie in willekeurige volumes. Meer details over de implementatie van deze materialen staat in de desbetreffende hoofdstukken.

11.2 Opmerkingen

C++ en geheugenbeheer

Voor beide auteurs was dit is het eerste C++-project van enige omvang. Onze ervaring met C++ was dan ook beperkt tot enkele kleine, eenvoudige programmeeroefeningen. Dit heeft in het begin geleid tot frustraties om het spectraal raamwerk werkend te krijgen en te integreren in PBRT. Hoe langer we echter bezig waren met de implementatie, hoe vlotter het ging.

Eén van de elementen van C++ waarmee echter problemen bleven opduiken, was het geheugenbeheer. Het renderen van een eenvoudige spectrale scène ging al snel tot 1 Gb geheugen gebruiken, onaanvaardbaar veel dus. De oorzaak hiervan was natuurlijk een grote hoeveelheid niet-vrijgegeven objecten. We hebben toen beslist om te werken met zogenaamde *smart pointers*.² Op enkele problemen met het interne geheugenbeheer van PBRT na, loste dit ons probleem met geheugenlekken grotendeels op.

Interpolatie

Omdat interpolatie van de waarden van een spectrum redelijk veel voorkomt in het spectraal raamwerk, hebben we beslist om een Interpolator-interface te voorzien. Er zijn op dit moment drie manieren voorzien om te interpoleren, maar dit zou eenvoudig uitgebreid kunnen worden. De geïmplementeerde manieren zijn lineair (**LinearInterpolator**), de dichtste meetwaarde (**NearestInterpolator**) en de meetwaarde die op de dichtstbijzijnde kleinere golflengte ligt (**LeftInterpolator**).

²Meer bepaald de Boost library, <http://www.boost.org>

11.3 Performantie

Eén van de grote bedenkingen die steeds gemaakt worden bij volledig spectrale rendering, is dat dit niet efficiënt zou zijn. Onze huidige implementatie is inderdaad heel wat trager dan de oorspronkelijke implementatie van PBRT. De grootste reden hiervoor is volgens ons echter niet het feit dat er spectraal gewerkt wordt. Zo vereist de keuzevrijheid van de voorstellingswijze voor een spectrum het gebruik van virtuele methodes en indirecties. Het gebruik van de ‘smart pointers’ levert ook wat vertraging, die echter het gebruiksgemak ten goede komt. Deze methodes zijn inherent trager dan de geoptimaliseerde originele implementatie. Wanneer men de code zou aanpassen zodat er slechts één voorstellingswijze (bijvoorbeeld een composietspectrum met een bemonsterd zacht verlopend gedeelte) mogelijk is, kan er zeer sterk geoptimaliseerd worden. Ons gebrek aan ervaring in C++ is een tweede reden waarom we vermoeden dat onze huidige implementatie niet zo efficiënt is als ze zou kunnen zijn. Dit wordt geïllustreerd door enkele cijfers:

	Origineel	Spectraal	Eigen RGB
scène 1	4.0 s	29.5 s	16.2 s
scène 2	2.6 s	18.5 s	9.3 s

De eerste kolom geeft de tijd die de originele implementatie van PBRT nodig had om een scène te renderen (in RGB). De middelste kolom bevat de tijden voor dezelfde scène met onze implementatie en met de spectra afgeleid uit de RGB-waarden. De gebruikte spectra zijn uniform bemonsterde spectra, met 20 monsters van 360 nm tot 830 nm. In de laatste kolom worden de tijden weergegeven die nodig waren om dezelfde scène te renderen gebruik makend van onze eigen RGB-implementatie.

Het valt onmiddellijk op dat onze implementatie heel wat trager is, om de redenen die hierboven aangehaald zijn. Dat dit niet hoofdzakelijk veroorzaakt wordt door het feit dat er spectraal gewerkt wordt, maar wel door de inefficiënties in de implementatie, blijkt uit de vergelijking met de tijden voor onze RGB-implementatie. Door van de originele implementatie over te schakelen op onze implementatie, wordt het hele proces ongeveer vier keer trager. De overschakeling van onze RGB-implementatie naar een volledig spectrale aanpak zorgt voor een verdubbeling van de rekestijd. Een verdubbeling is niet veel, als men weet dat men van een voorstelling met 3 getallen overgaat naar één met 20 getallen (of dus meer dan 6 keer zoveel). Zoals gezegd zijn we ervan overtuigd dat deze rekestijden nog fel kunnen dalen door een optimalisatie van de code voor één bepaald type spectrum.

11.4 Beslissingen

Conversiemethode

Er zijn twee verschillende methodes geïmplementeerd om een RGB-waarde om te zetten naar een spectrum, namelijk de methode van Smits en Sun. Nu wordt steeds de methode van Smits gekozen. Om de methode van Sun te gebruiken, wijzigt men in `RGBConverter.h` uit het raamwerk het standaard-type bij constructie van een convertor van `SMITS` naar `SUN`.

Spectrumimplementatie

De standaardimplementatie voor een spectrum die gebruikt wordt, is op dit moment een composietspectrum. Dit kan gewijzigd worden in `color.h` door de typedefinitie van `SpectrumImplementation` te wijzigen. Deze wijziging heeft effect op alle spectra die geen expliciet type gekregen hebben in het invoerbestand.

Resultaattype

Het type van het resultaat van een bewerking tussen twee spectra wordt vastgelegd in `color.h`. Voor een bepaald type linker- en rechteroperand kan daar bepaald worden wat het resultaattype van de bewerking is.

12.1 Overzicht

In dit eindwerk werd het gebruik van kleur in computer graphics van naderbij bekeken. Gebruik makende van de eigenschappen van licht, optica en de aspecten van menselijke waarneming, werd de fysisch gebaseerde benadering van kleur voorgesteld. Hiertoe werd gebruik gemaakt van het spectrum van het licht.

Om spectra voor te stellen en ermee te rekenen, zijn er verschillende mogelijkheden. We hebben een spectraal raamwerk ontworpen en de belangrijkste manieren om spectra voor te stellen hierin geïmplementeerd. Ook hebben we een bestaande fysisch gebaseerde ray tracer (PBRT) aangepast zodat gebruik gemaakt wordt van het raamwerk.

12.2 Spectraal raamwerk

De relatie met de traditioneel gebruikte kleurmodellen in computer graphics, namelijk RGB en XYZ, werd aangegeven. Het blijkt dat een kleur volledig gekarakteriseerd kan worden door drie waarden, maar dat deze methodes niet volstaan om correcte berekeningen met kleur uit te voeren. Hiervoor is een spectrale aanpak vereist.

Er werd een raamwerk van verschillende spectrale representaties voorgesteld en geïmplementeerd. De vier belangrijkste representaties uit de literatuur werden bestudeerd en er werd aangetoond dat er voor ieder mogelijk spectrum een passende representatie is. De belangrijkste voor- en nadelen van de voorstellingswijzen worden kort overlopen:

- Het bemonsterde spectrum werkt met meetwaarden op een aantal equidistante golf lengtes en wordt geïnterpoleerd met een methode naar keuze. Dit biedt eenvoudige controle over de omvang en nauwkeurigheid

van de gegevens. Het is algemeen een goede keuze voor spectra zonder pieken.

- Het getabuleerde spectrum heeft meetwaarden op willekeurige golflengtes. Dit heeft als voordeel dat men adaptief zou kunnen verfijnen in belangrijke gebieden. Op die manier kunnen eventueel ook spectra met pieken voorgesteld worden. De hoeveelheid gegevens is een klein nadeel.
- Het Fourierspectrum biedt een oplossing voor zacht verlopende spectra. Het beperkte aantal coëfficiënten kan een voordeel zijn. De bewerkingen zijn veel trager dan bijvoorbeeld het bemonsterde spectrum, maar ze zouden geoptimaliseerd kunnen worden.
- Het composietspectrum is een goede oplossing om spectra met pieken voor te stellen. Het zacht verlopende deel kan met een willekeurig ander spectrum worden uitgedrukt en de pieken worden met een golflengte en een gewicht uitgedrukt. De bewerkingen zijn ondanks de tweeledige opbouw nog vrij snel.

Bewerkingen tussen spectra van verschillende klassen zijn mogelijk dankzij automatische conversies. De architectuur werd ontworpen om uitbreidbaar te zijn met extra representaties. Er zijn er heel wat te vinden in de literatuur die zeker de moeite waard zijn om eens te proberen.

Nog een mogelijke richting voor toekomstig werk is een studie van de nodige hoeveelheid gegevens om nauwkeurig en efficiënt te werken. De spectrale eigenschappen van het menselijk oog kunnen gebruikt worden om de rendering te versnellen of om een foutmetriek voor afbeeldingen te ontwerpen.

12.3 Spectrale effecten

Naast correcte berekeningen zijn er ook enkele spectrale effecten die niet via RGB kunnen gesimuleerd worden. Deze effecten zijn afhankelijk van de golflengte van het invallend licht. We hebben deze effecten, gebruik makende van het spectraal raamwerk, geïmplementeerd in de bestaande ray tracer.

Er werden vier effecten behandeld, die samen ongeveer het volledige bereik van de golfoptica omvatten:

- Interferentie in zeepbellen werd op een fysisch gebaseerde manier afgeleid. De voorgestelde oplossing kan efficiënt in de meeste renderers toegevoegd worden. De resulterende afbeeldingen zijn een overtuigende weergave van de realiteit. Interferentie in meerdere lagen is triviaal af te leiden.

- Diffractie van een compact disc werd behandeld als een gedeeltelijk fysisch gebaseerd, gedeeltelijk fenomenologisch model. De resultaten zijn al mooi, maar er is nog ruimte voor verbetering. Een volledig fysisch gebaseerd model ligt binnen de mogelijkheden.
- Dispersie werd uitgewerkt op een manier die nog veel ruimte laat voor verbetering. Het effect kan in een ray tracer niet op een gecentraliseerde plaats toegevoegd worden. Er werden spectrale parameters toegevoegd aan de interface van de kernalgoritmes en de materialen. Daardoor wordt de toevoeging van andere effecten uit de golfoptica veel eenvoudiger. De afbeeldingen zijn veelbelovend, maar het algoritme moet nog veel efficiënter worden. Er is een aanzet gegeven voor het adaptief splitsen in golflengtes afhankelijk van de dispersiviteit van het materiaal. *Importance sampling* op de spectra zal ongetwijfeld voor een belangrijke versnelling zorgen.
- Van absorptie in volumes werd bewezen dat een spectrale aanpak de enige fysisch correcte is. De besproken methode maakt het mogelijk om opgemeten media accuraat te visualiseren.

De opgemeten BRDF's hadden niet het verhoopte resultaat. Met reflectanties van materialen moet voor de correctheid wel spectraal gerekend worden, maar het verschil met RGB is bij reële materialen zelden zichtbaar. Grotere verschillen zouden optreden bij reflectantiespectra met pieken, zoals bij fluorescentie.

We hebben kort vermeld hoe men fluorescentie zou kunnen toevoegen. Dit zou een grote aanpassing in de interface van de materialen vergen. De aanpassing zou toch de moeite waard zijn, omdat dit effect vrij veel voorkomt in de natuur.

Polarisatie is meestal niet zo opvallend en het is ook geen spectraal effect. Het is wel een effect dat door de golfoptica kan verklaard worden. Dankzij onze aanpassingen in het kader van dispersie wordt het toevoegen van polarisatie een stuk eenvoudiger.

12.4 Slot

We hebben aangetoond dat het gebruik van spectra bij het genereren van fotorealistische afbeeldingen noodzakelijk is om correcte resultaten te bekomen. Ook de behandeling van enkele spectrale effecten die in het dagelijks leven frequent voorkomen, vereist deze aanpak. We menen dan ook dat er meer aandacht besteed mag worden aan het gebruik van spectra en dat de traditionele aanpak via RGB niet meer volstaat.



Bibliografie

- James R. Arvo, augustus 1986. Backward Ray Tracing. In *ACM SIGGRAPH '86 Course Notes - Developments in Ray Tracing*, deel 12.
- Ezekiel Bahar en Swapan Chakrabarti, juli 1987. Full-Wave Theory Applied to Computer-Aided Graphics for 3D Objects. *IEEE Computer Graphics and Applications*, 7-7 (1987), 46–60.
- James F. Blinn, juli 1977. Models of Light Reflection For Computer Synthesized Pictures. In *Computer Graphics (SIGGRAPH '77 Proceedings)*, deel 11, p. 192–198.
- James F. Blinn, augustus 1978. Simulation of Wrinkled Surfaces. In *Computer Graphics (SIGGRAPH '78 Proceedings)*, deel 12, p. 286–292.
- Max Born en Emil Wolf, 1999. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. Cambridge University Press, 7^e druk.
- Brian Cabral, Nelson Max, en Rebecca Springmeyer, juli 1987. Bidirectional Reflection Functions from Surface Bump Maps. In *Computer Graphics (SIGGRAPH '87 Proceedings)*, deel 21, p. 273–281.
- Edwin E. Catmull, december 1974. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. Dissertatie, Dept. of CS, University of Utah.
- Steven Collins, november 1994. Rendering Crystal Glass. In *Proceedings of the Second Irish Computer Graphics Workshop*.
- Steven Collins, 1997. *Wavefront Tracking for Global Illumination Solution*. Dissertatie, Trinity College Dublin.
- Rob L. Cook en Kenneth E. Torrance, augustus 1981. A reflectance model for computer graphics. *Computer Graphics (SIGGRAPH '81 Proceedings)*, 15-3 (1981), 307–316.

- Maria Lurdes Dias, maart 1991. Ray Tracing Interference Color. *IEEE Computer Graphics and Applications*, 11-2 (1991), 54–60.
- Maria Lurdes Dias, mei 1994. Ray tracing interference color: visualizing Newton's rings. *IEEE Computer Graphics and Applications*, 14-3 (1994), 17–20.
- Philip Dutré, Philippe Bekaert, en Kavita Bala, 2003. *Advanced Global Illumination*. A K Peters.
- Glenn F. Evans en Michael D. McCool, juni 2–4 1999. Stratified Wavelength Clusters for Efficient Spectral Monte Carlo Rendering. In *Proceedings of the Conference on Graphics Interface (GI-99)*, p. 42–49.
- Judith Gartaganis, 1992. *A Wave-Based Illumination Model for Graphics*. Dissertatie, University of Alberta.
- Andrew Glassner, september/oktober 2000a. Andrew Glassner's Notebook: Soap Bubbles: Part 1. *IEEE Computer Graphics and Applications*, 20-5 (2000), 76–84.
- Andrew Glassner, november/december 2000b. Andrew Glassner's Notebook: Soap Bubbles: Part 2. *IEEE Computer Graphics and Applications*, 20-6 (2000), 99–109.
- Jürgen Golz en Donald I. A. MacLeod, mei 2003. Colorimetry for CRT displays. *Journal of the Optical Society of America A*, 20-5 (2003), 769–781.
- Jay S. Gondek, Gary W. Meyer, en Jonathan G. Newman, juli 1994. Wavelength Dependent Reflectance Functions. In *Proceedings of SIGGRAPH '94*, Andrew Glassner, redactie, deel 28, p. 213–220.
- J. Guild, 1931. The colorimetric properties of the spectrum. *Philosophical Transactions of the Royal Society of London*, A230 (1931), 149–187.
- Roy A. Hall, 1989. *Illumination and Color in Computer Generated Imagery*. New York, Springer-Verlag.
- Pat Hanrahan en Wolfgang Krueger, augustus 1993. Reflection from Layered Surfaces Due to Subsurface Scattering. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, deel 27, p. 165–174.

- H. Hirayama, K. Kaneda, H. Yamashita, en Y. Monden, juni 2001a. An accurate illumination model for objects coated with multilayer films. *Computers and Graphics*, 25-3 (2001), 391–400.
- H. Hirayama, K. Kaneda, H. Yamashita, Y. Yamaji, en Y. Monden, 2001b. Visualization of optical phenomena caused by multilayer films based on wave optics. *The Visual Computer*, 17(2) (2001), 106–120.
- Mark Huyse, 2001. *Natuurkunde (volume C)*. Cursusdienst VTK vzw.
- Dariusz Jazzkowski en Janusz Rzeszut, juli 2003. Interference colours of soap bubbles. *The Visual Computer*, 19(4) (2003), 252–270.
- Henrik Wann Jensen, 2001. *Realistic Image Synthesis Using Photon Mapping*. A K Peters.
- Henrik Wann Jensen en Per H. Christensen, juli 1998. Efficient Simulation of Light Transport in Scenes With Participating Media Using Photon Maps. In *SIGGRAPH 98 Conference Proceedings*, p. 311–320.
- Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, en Pat Hanrahan, augustus 2001. A Practical Model for Subsurface Light Transport. In *SIGGRAPH 2001 Conference Proceedings*, Annual Conference Series, p. 511–518.
- James T. Kajiya, juli 1985. Anisotropic Reflection Models. In *Computer Graphics (ACM SIGGRAPH '85 Proceedings)*, deel 19(3), p. 15–21.
- James T. Kajiya, augustus 1986. The Rendering Equation. In *Computer Graphics (ACM SIGGRAPH '86 Proceedings)*, deel 20, p. 143–150.
- Eric P. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, en Donald P. Greenberg, augustus 1997. Non-Linear Approximation of Reflectance Functions. In *Computer Graphics (ACM SIGGRAPH '97 Proceedings)*, deel 31, p. 117–126.
- Stephen R. Marschner, Stephen H. Westin, Eric P. F. Lafortune, Kenneth E. Torrance, en Donald P. Greenberg, 1999. Image-Based BRDF Measurement Including Human Skin. In *Rendering Techniques '99*, p. 131–144.
- Gary W. Meyer, januari 1988. Wavelength Selection for Synthetic Image Generation. *Computer Vision, Graphics, and Image Processing*, 41-1 (1988), 57–79.

- Hans P. Moravec, augustus 1981. 3D Graphics and the Wave Theory. In *Computer Graphics (SIGGRAPH '81 Proceedings)*, deel 15(3), p. 289–296.
- F. Kenton Musgrave, juni 1989. Prisms and Rainbows: a Dispersion Model for Computer Graphics. In *Proceedings of Graphics Interface '89*, p. 227–234.
- Duc Quang Nguyen, Ronald P. Fedkiw, en Henrik Wann Jensen, juli 2002. Physically Based Modeling and Animation of Fire. In *Proceedings of the 29th Conference on Computer Graphics and Interactive Techniques (SIGGRAPH-02)*, *ACM Transactions on Graphics*, deel 21, p. 721–728.
- Matt Pharr en Greg Humphreys, 2003. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, Elsevier.
- Bui-Tuong Phong, juni 1975. Illumination for Computer Generated Pictures. *Communications of the ACM*, 18-6 (1975), 311–317.
- Pierre Poulin en Alain Fournier, augustus 1990. A Model for Anisotropic Reflection. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, Forest Baskett, redactie, deel 24, p. 273–282.
- Holly E. Rushmeier en Kenneth E. Torrance, juli 1987. The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium. In *Computer Graphics (ACM SIGGRAPH '87 Proceedings)*, deel 21, p. 293–302.
- Morgan Schramm, Jay Gondek, en Gary Meyer, mei 1997. Light Scattering Simulations using Complex Subsurface Models. In *Proceedings of the Graphics Interface '97*, Wayne A. Davis, Marilyn Mantei, en R. Victor Klassen, redactie, p. 56–67.
- Peter Shirley en R. Keith Morley, 2003. *Realistic Ray Tracing*. A K Peters, tweede druk.
- Brian Smits, 1999. An RGB to Spectrum Conversion for Reflectances. *Journal of Graphics Tools*, 4-4 (1999), 11–22.
- Brian E. Smits en Gary Meyer, 1992. Newton's Colors: Simulating Interference Phenomena in Realistic Image Synthesis. In *Proceedings Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics, 1990*, K. Bouatouch en C. Bouville, redactie, p. 185–194.

- Jos Stam, 1999. Diffraction Shaders. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, p. 101–110. ACM Press/Addison-Wesley Publishing Co.
- Yinlong Sun, juli 2000. *A Spectrum-Based Framework for Realistic Image Synthesis*. Dissertatie, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada.
- Yinlong Sun, F. David Fracchia, Thomas W. Calvert, en Mark S. Drew, juli/augustus 1999. Deriving Spectra from Colors and Rendering Light Interference. *IEEE Computer Graphics and Applications*, 19-4 (1999), 61–67.
- Yinlong Sun, F. David Fracchia, en Mark S. Drew, 1998. A Composite Model for Representing Spectral Functions. Technical Report TR 1998-18, Simon Fraser University.
- Yinlong Sun, F. David Fracchia, en Mark S. Drew, oktober 1999a. Rendering The Phenomena of Volume Absorption in Homogeneous Transparent Materials. In *2nd Annual IASTED International Conference on Computer Graphics and Imaging (CGIM'99)*, p. 283–288.
- Yinlong Sun, F. David Fracchia, Mark S. Drew, en Thomas W. Calvert, september 1999b. Rendering Iridescent Colors of Optical Disks. Technical Report TR 1999-08, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada.
- Yinlong Sun, F. David Fracchia, Mark S. Drew, en Thomas W. Calvert, juni 2000. Rendering Iridescent Colors of Optical Disks. In *Rendering Techniques 2000, Proceedings of the Eurographics Workshop*, p. 341–352.
- Spencer W. Thomas, januari 1986. Dispersive refraction in ray tracing. *The Visual Computer*, 2-1 (1986), 3–8.
- Gregory J. Ward, juli 1992. Measuring and modeling anisotropic reflection. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, deel 26, p. 265–272.
- Stephen H. Westin, James R. Arvo, en Kenneth E. Torrance, juli 1992. Predicting Reflectance Functions From Complex Surfaces. In *Computer Graphics (ACM SIGGRAPH '92 Proceedings)*, deel 26, p. 255–264.
- Turner Whitted, juni 1980. An Improved Illumination Model for Shaded Display. *Communications of the ACM*, 23-6 (1980), 343–349.

W.D. Wright, 1928. A re-determination of the trichromatic coefficients of the spectral colours. *Transactions of the Optical Society*, 30 (1928), 141–164.

Aanpassingen in het .pbrt invoerformaat

Onze implementatie is toegevoegd aan het invoerformaat van PBRT. De verschillen tussen de officiële PBRT versie 1.02 en onze aangepaste versie worden beschreven.

A.1 Lijsten van parameters

PBRT ondersteunt standaard 7 datatypes voor parameters: **integer**, **float**, **point**, **vector**, **normal**, **color** en **string**.

Het type **color** nam oorspronkelijk drie decimale waarden voor een RGB-kleur. We hebben dit type aangepast zodat de RGB-kleur dadelijk wordt omgezet naar een bemonsterd spectrum, met de omzettingmethode van Smits (sectie 2.6.2, p.24). Er zijn bijkomende datatypes voorzien voor de verschillende voorstellingswijzen van spectra. Al deze types kunnen gebruikt worden waar er normaal een **color** verwacht wordt.

- **rgbspectrum** neemt drie decimale waarden en houdt die bij als RGB spectrum. Er gebeurt geen omzetting naar een bemonsterd spectrum zoals bij **color**.
- **samplespectrum** neemt een willekeurig aantal decimale getallen en maakt een bemonsterd spectrum. De eerste twee getallen worden geïnterpreteerd als een onder- en bovengrens voor de golflengte. De rest van de getallen zijn bemonsterde waarden op gelijke afstanden tussen de gegeven grenzen.
- **fourierspectrum** neemt eveneens een willekeurig aantal decimale getallen en maakt een Fourierspectrum. De eerste twee zijn opnieuw de

onder- en bovengrens van de golflengte. De rest zijn de Fouriercoëfficiënten, afwisselend van de cosinus- en sinustermen.

- **tabulatedspectrum** neemt een even aantal decimale getallen, afwisselend een golflengte en een bemonsterde waarde en maakt een getabuleerd spectrum.
- **compositespectrumsampled** maakt een composietspectrum met als zachtverlopend deel een bemonsterd spectrum. De eerste waarde is een positief geheel getal n dat het aantal monsters voor het bemonsterde spectrum aangeeft. De volgende twee waarden zijn decimale getallen voor de onder- en bovengrens van de golflengte voor het bemonsterde spectrum. Daarna worden er n decimale waarden als monsters geïnterpreteerd. Alle decimale waarden die daar nog op volgen, worden per twee als een golflengte en gewicht van een piek beschouwd.
- **compositespectrumfourier** maakt een composietspectrum met als zachtverlopend deel een Fourierspectrum. De eerste waarde is een positief geheel getal n dat het aantal Fouriercoëfficiënten aangeeft. De volgende twee waarden zijn decimale getallen voor de onder- en bovengrens van de golflengte voor het Fourierspectrum. Daarna worden er n decimale waarden geïnterpreteerd als Fouriercoëfficiënten, afwisselend van de cosinus- en sinustermen. Alle decimale waarden die daar nog op volgen, worden per twee als een golflengte en gewicht van een piek beschouwd.
- **rgbfromsampled** neemt dezelfde waarden als **sampledspectrum**: een onder- en bovengrens voor de golflengte en een aantal monsters. Het bemonsterde spectrum wordt onmiddellijk omgezet naar een RGB-spectrum door toepassing van de color matching functies.
- **rgbfromtabulated** neemt dezelfde waarden als **tabulatedspectrum**: afwisselend een golflengte en monster. Het getabuleerde spectrum wordt onmiddellijk omgezet naar een RGB-spectrum door toepassing van de color matching functies.

A.2 Uitdrukkingen

PBRT is gebaseerd op een linkshandig assenstelsel. Dit is een arbitraire keuze, maar ze is van belang voor geometrische transformaties. De **LookAt**-transformatie bevatte een fout waardoor er omgeschakeld werd naar een

rechtshandig assenstelsel. Om compatibel te blijven met de meegeleverde voorbeeldscènes hebben de auteurs besloten om dit niet te verbeteren. Wij hebben de verbetering wel aangebracht om consistent te kunnen modelleren.

A.3 Plug-Ins

De architectuur van PBRT is ontworpen om volledig modulair en “pluggable” te zijn. Bijna alles kan als plug-in toegevoegd worden zonder aanpassingen aan de rest van het systeem.

Materialen

Naam	Klasse
"diffractive"	Diffractive
"dispersiveglass"	DispersiveGlass
"filter"	Filter
"sampledbrdf"	SampledMaterial
"sampleddiffractive"	SampledDiffractiveMaterial
"thinfilm"	ThinFilm

Specifieke parameters

diffractive::c0,c1,c2,c3,c4

Type: float texture

Standaardwaarde: 0.005, 1, 0.01, 1000, 0

Omschrijving: Experimentele coëfficiënten van de diffractieve component.

diffractive::alfa

Type: float texture

Standaardwaarde: 10000

Omschrijving: De exponent van de isotrope speculaire reflectie, die de glans van het materiaal bepaalt.

diffractive::beta

Type: float texture

Standaardwaarde: 2

Omschrijving: De exponent van de anisotrope speculaire reflectie, die het aantal zichtbare nevenstrips bepaalt.

diffractive::track_a

Type: float texture

Standaardwaarde: 300

Omschrijving: De staplengte van de putjes, in nm.

diffraction::track_b

Type: float texture

Standaardwaarde: 1600

Omschrijving: De afstand tussen de sporen, in nm.

diffraction::cdiff

Type: float texture

Standaardwaarde: 0.001

Omschrijving: De coëfficiënt van diffuse reflectie.

diffraction::cspec

Type: float texture

Standaardwaarde: 50

Omschrijving: De coëfficiënt van isotrope speculaire reflectie.

dispersiveglass::Kr

Type: color texture

Standaardwaarde: 1

Omschrijving: De reflectiviteit van het oppervlak.

dispersiveglass::Kt

Type: color texture

Standaardwaarde: 1

Omschrijving: De transmissiviteit van het oppervlak.

dispersiveglass::index

Type: float texture

Standaardwaarde: 1.60342

Omschrijving: De brekingsindex.

dispersiveglass::B1,B2,B3,C1,C2,C3

Type: float texture

Standaardwaarde: 1.3104463, 0.19603426, 0.96612977, 0.00958633048, 0.0457627627, 115.011883

Omschrijving: De coëfficiënten van de Sellmeier-vergelijking.

filter::Kt

Type: color texture

Standaardwaarde: 1

Omschrijving: De transmissiviteit van het oppervlak.

sampledbrdf::index

Type: float texture

Standaardwaarde: 1.5

Omschrijving: De brekingsindex.

sampledbrdf::Kr

Type: color texture

Standaardwaarde: 0

Omschrijving: De coëfficiënt van speculaire reflectie.

sampledbrdf::filename

Type: **string**

Standaardwaarde: Geen

Omschrijving: De bestandsnaam van de meetgegevens. Het bestand moet in ASTM Standard E1392-96 formaat opgemaakt zijn, maar de waarden moeten gescheiden worden door spaties i.p.v. komma's.

sampledbrdf::usergb

Type: **bool**

Standaardwaarde: false

Omschrijving: Geeft aan of de meetgegevens naar RGB geconverteerd moeten worden na de interpolatie.

thinfilm::index1

Type: **float texture**

Standaardwaarde: 1

Omschrijving: De brekingsindex aan de buitenkant van de film.

thinfilm::index2

Type: **float texture**

Standaardwaarde: **index1**

Omschrijving: De brekingsindex aan de binnenkant van de film.

thinfilm::indexfilm

Type: **float texture**

Standaardwaarde: 1.5

Omschrijving: De brekingsindex van van de film.

thinfilm::thickness

Type: **float texture**

Standaardwaarde: 0.5

Omschrijving: De dikte van de film, in μm .

thinfilm::Kr

Type: **color texture**

Standaardwaarde: 1

Omschrijving: De reflectiviteit van het oppervlak.

thinfilm::Kt

Type: **color texture**

Standaardwaarde: 0

Omschrijving: De transmissiviteit van het oppervlak.

Texturen

Naam	Klasse
"gravitational"	Gravitational

Gravitational is een 3D textuur. De evaluatie gebruikt een positie in de textuuruimte. De transformatiematrix op het ogenblik van de creatie van de textuur bepaalt de transformatie van de objectruimte naar de textuuruimte.

Specifieke parameters

gravitational::dmin

Type: float

Standaardwaarde: 0.2

Omschrijving: De minimale dikte of dichtheid.

gravitational::dmax

Type: float

Standaardwaarde: 1.5

Omschrijving: De maximale dikte of dichtheid.

Volumes

Naam	Klasse
"homogeneous"	HomogeneousVolume

De oorspronkelijke implementatie van **HomogeneousVolume** was beperkt tot balkvormige volumes uitgelijnd op de assen. We hebben de werking veralgemeend tot willekeurige volumes. De gebruikte vorm is het **Shape**-object dat direct na het **Volume**-commando volgt.

Lichtbronnen

Naam	Klasse
"laser"	Laser

Specifieke parameters

laser::I

Type: color

Standaardwaarde: 1

Omschrijving: De kleur van het licht.

laser::radius

Type: float

Standaardwaarde: 0.001

Omschrijving: De straal van de lichtbundel.

`laser::from`

`laser::to`

Type: `point`

Standaardwaarde: zie omschrijving

Omschrijving: Twee punten die de richting van de lichtbundel bepalen. De standaardwaarden zijn resp. $(0, 0, 0)$ en $(0, 0, 1)$. Dit is een lichtbron die langs de z -as naar beneden schijnt.

Integrators

De oppervlak-integrators zijn de algoritmes voor ray tracing en globale belichting die geen rekening houden met participerende media. We hebben al deze algoritmes¹ aangepast om dispersie te renderen. Aan het invoerformaat is niets veranderd.

De volume-integrators, die wel rekening houden met participerende media, zijn niet aangepast.

¹We hebben de verbeterde foton map `exphotonmap` niet aangepast omdat die pas vanaf 25 april beschikbaar was.