

**Henrik Juhana Paananen**

# **Fysikaalinen valaistus ja reaaliaikaiset heijastukset**

Tietotekniikan pro gradu -tutkielma

4. toukokuuta 2016

Jyväskylän yliopisto

Tietotekniikan laitos

**Tekijä:** Henrik Juhana Paananen

**Yhteystiedot:** henrik.j.paananen@student.jyu.fi

**Ohjaajat:** Sanna Mönkölä ja Tuomo Rossi

**Työn nimi:** Fysikaalinen valaistus ja reaaliaikaiset heijastukset

**Title in English:** Physically-based shading and real-time reflections

**Työ:** Pro gradu -tutkielma

**Suuntautumisvaihtoehto:** Ohjelmistotekniikka

**Sivumäärä:** 81+0

**Tiivistelmä:** Epäsuora valaistus, kuten ympäristön heijastukset, ovat tärkeä osa uskottavaa tietokonegrafiikkaa. Tarvitaan kuitenkin fysikaalinen valaistusmalli, jotta epäsuoran valon vaikutus olisi realistinen. Vasta viime aikojen näytönohjainten tehokkuuden kasvu on mahdollistanut fysikaalisen epäsuoran valaistuksen mallintamisen reaaliajassa. Koska epäsuoran valaistuksen laskeminen on erittäin raskasta, joudutaan kuitenkin tekemään arviointia, jotta reaaliaikainen toteutus olisi mahdollista. Tässä tutkielmassa esitellään fysikaalisen valaistuksen perustaa ja annetaan esimerkkejä reaaliaikaisista heijastusmenetelmistä. Lopuksi evaluoidaan kahta heijastustekniikkaa, ympäristötutkaimia ja kuvapuskuriheijastuksia, fysikaalisen valaistuksen näkökulmasta. Evaluointi tapahtuu toteuttamalla tekniikat ja arvioimalla saatua lopputulosta uskottavuuden, toimivuuden ja suorituskyvyn kannalta. Huomattiin, että usean tekniikan yhdisteleminen on tarpeen, sillä eri tekniikat soveltuvat vain tietynlaisiin heijastuksiin. Lisäksi tekniikoissa on useita ongelmakohtia, jotka rajoittavat niiden käytettävyyttä.

**Avainsanat:** reaaliaikainen grafiikka, fysikaalinen valaistus, heijastukset, kaksisuuntainen heijastusjakaumafunktio, kuvapuskuriheijastukset, ympäristötutkaimet

**Abstract:** Indirect lighting, like reflections of the environment, have an important role in believable computer graphics. To make the indirect lighting realistic, a physically based lighting model is needed. Only as of lately, the modeling of physically based indirect lighting

in real-time has been possible, due to the recent increase of GPU-power. As calculating the indirect lighting is expensive, approximation of the phenomenon is still needed to make real-time implementation possible. In this thesis we introduce the basis of physically based lighting and give examples of real-time reflection methods. Finally two reflection techniques, environment probes and screen space reflections, are evaluated from the physically based lighting point of view. The evaluation is carried out by implementing these techniques and assessing their plausibility and performance. It was discovered that combining several techniques is necessary, as each technique can only produce some particular kind of reflections. In addition it was found that the techniques have several problematic situations, which limit their usability.

**Keywords:** real-time graphics, physically-based shading, reflections, bidirectional reflectance distribution function, screen space reflections, environment probes

## Termiluettelo

Fysikaalinen renderöinti	(engl. <i>physically based rendering, PBR</i> ) on yleinen termi, jolla tarkoitetaan fysiikkaan perustuvaa renderöintiä. Renderöinnissä käytetään fysikaalisiin suureisiin ja mitattuihin arvoihin pohjautuvia materiaalikuvauksia ja valoja pintojen valaisussa. Tämän lisäksi virtuaalinen kamera on mallinnettu vastaamaan reaalimaailman kamerassa tai silmässä tapahtuvia ilmiöitä, joista aiheutuvat esimerkiksi liike- ja syvyysepäterävyys (engl. <i>motion blur, depth of field</i> ).
Fysikaalinen valaistus	(engl. <i>physically based shading, PBS</i> ) on termi, jota käytetään, kun puhutaan fysiikkaan perustuvaan valon vuorovaikutuksesta pinnan kanssa valaistaessa kappaleen pintaa.
Fysikaalinen varjostus	katso <i>fysikaalinen valaistus</i> .
Kaksisuuntainen heijastusjakaumafunktio	(engl. <i>bidirectional reflectance distribution function, BRDF</i> ) on funktio, joka kuvaa valon heijastumista pinnasta valon tulokulman ja katselusuunnan mukaan.
Primitiivi	on yksinkertaisin geometrinen muoto, josta monimuotoisemmat kappaleet koostuvat. Näitä ovat esimerkiksi piste, jana ja kolmio. Yleensä kolmiulotteiset kappaleet pitää jakaa kolmioprimitiiveiksi ennen kuin ne voidaan piirtää.
Suora varjostus	(engl. <i>forward shading</i> ) on perinteinen tapa piirtää grafiikkaa näytönohjaimella. Jokaista primitiiviä piirrettäessä käydään läpi primitiiviin vaikuttavat valot ja suoritetaan valaistus.
Valaistusmalli	(engl. <i>illumination model, shading model</i> ) kuvaa valon käyttäytymistä erilaisilla pinnoilla.
Verteksi	on monikulmion kärkipiste.
Viivästetty varjostus	(engl. <i>deferred shading</i> ) on tekniikka, jossa valaistus ja geometrian piirto on erotettu toisistaan. Ensin piirretään geometria niin kutsuttuun g-puskuriin, jonka jälkeen käydään kaikki g-puskurin kuvaelementit valaistaan.



## Matemaattiset merkinnät

$\mathbb{R}$	on reaalilukujen joukko.
$\mathbb{R}^3$	on kolmiulotteisten reaalivektorien joukko.
$x$	on reaaliarvoinen muuttuja tai funktio.
$[a, b]$	on suljettu väli, jonka alaraja on $a$ ja yläraja $b$ .
$\mathbf{n}$	on vektoriarvoinen muuttuja tai funktio.
$\ \mathbf{n}\ $	on vektorin $\mathbf{n}$ pituus.
$\hat{\mathbf{n}}$	on yksikkövektori, jonka pituus on 1 ja suunta $\mathbf{n}$ , toisin sanoen $\hat{\mathbf{n}} = \frac{\mathbf{n}}{\ \mathbf{n}\ }$ .
$\vec{\omega}$	on avaruuskulma, jonka yksikkö steradiaani [sr].
$\frac{\partial f}{\partial x}$	on funktion $f$ osittaisderivaatta muuttujan $x$ suhteen.
$\nabla \cdot \mathbf{F}$	on vektorifunktion $\mathbf{F}$ divergenssi eli läheisyys. Kolmiulotteiselle vektorifunktiolle $\mathbf{F} = (F_x, F_y, F_z)$ divergenssi on $\nabla \cdot \mathbf{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z}$
$\nabla \times \mathbf{F}$	on vektorifunktion $\mathbf{F}$ roottori eli pyörre. Kolmiulotteiselle vektorifunktiolle $\mathbf{F} = (F_x, F_y, F_z)$ roottori on $\nabla \times \mathbf{F} = \left( \frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z}, \frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x}, \frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \right)$

## Kuviot

Kuvio 1. (a) Valon säteen heijastuminen. (b) Valon säteen taittuminen. ....	10
Kuvio 2. Valon heijastuminen, taittuminen ja absorboituminen optisesti epätasaisesta pinnasta. (a) Pintaan saapuu valoa. (b) Valoa taittuu pinnan alle ja osa tästä valosta taittuu takaisin pinnan yläpuolelle kimpoiltuaan ensin väliaineen hiukkasista. (c) Osa pinnan alle taittuneesta valosta absorboituu. (d) Valoa heijastuu peilimäisesti pinnasta. Koska pinta ei ole täysin tasainen, ei kaikki valo heijastu täysin samaan suuntaan. ....	11
Kuvio 3. Kuviossa on esitetty polarisaation geometria valon heijastuessa ja taittuessa. Vektorit $\mathbf{i}_{\parallel}$ ja $\mathbf{i}_{\perp}$ ovat rajapintaan tulevan valon yhdensuuntainen ja kohtisuora polarisaatioakseli. Heijastuneen valon yhdensuuntainen polarisaatioakseli on $\mathbf{r}_{\parallel}$ ja kohtisuora polarisaatioakseli $\mathbf{r}_{\perp}$ . Vastaavasti $\mathbf{t}_{\parallel}$ ja $\mathbf{t}_{\perp}$ ovat taittuneen valon yhdensuuntainen ja kohtisuora polarisaationakseli. ....	12
Kuvio 4. Profiilikuva mikrotahkoteorian mukaisesta pinnasta. Kuviossa on esitetty muutamien mikrotahkojen normaaleja $\mathbf{m}$ ja makropinnan normaali $\mathbf{n}$ . ....	19
Kuvio 5. Kuviossa on esitetty puolivektorin $\mathbf{h}$ geometria. Vektorien $\boldsymbol{\omega}_i$ ja $\mathbf{h}$ välinen kulma $\theta$ on sama kuin vektorien $\boldsymbol{\omega}_o$ ja $\mathbf{h}$ . ....	20
Kuvio 6. (a) Mikrogeometria varjostaa pinnalle saapuvaa valoa. (b) Mikrogeometria peittää pinnalta lähtevää valoa. ....	21
Kuvio 7. Kuvassa on esitetty tasoheijastusten toimintaperiaate. Kohdassa (1) näkyy virtuaalikameran näkymä ja kohdassa (2) näkymä on projisoitu heijastustavan geometrian pinnalle. ....	24
Kuvio 8. Kuvassa (a) jokaisella kappaleella on oma ympäristökartta. Kuvassa (b) ympäristökartat on kaapattu niin kutsutuista ympäristötutkaimista. Kuviossa liukuväritetyt ympyrät kuvaavat ympäristökarttojen sijaintia. ....	36
Kuvio 9. Kuvio esittää heijastusvektorin korjauksen geometrian. Korjattu heijastusvektori $\boldsymbol{\omega}'_r$ saadaan heijastuspisteen $\mathbf{p}'$ ja ympäristökartan projektion keskipisteen $\mathbf{p}_{env}$ erotuksena. ....	37
Kuvio 10. Kuviossa on näkymä, jossa lähes täysin peiliheijastavassa tasossa näkyy heijastus kahdesta seinästä. Kuvassa (a) heijastukset on toteutettu perinteisellä ympäristökartoituksella. Kuvassa (b) heijastukset on toteutettu suorakulmaisella särmiöllä lokalisoitulla ympäristökartalla. Kuva (c) esittää näkymän geometrian. Musta kartio osoittaa virtuaalikameran sijainnin, vihreä kuutio havainnollistaa ympäristökartan projektion keskipisteen ja oranssi rautalankasärmiö kuvaa ympäristökartan lokalisointiin käytetyn geometrian. ....	38
Kuvio 11. Kuviossa on havaittavissa lokalisoinnin epätarkkuus, kun lokalisointi perustuu laatikkosijaisgeometriaan. Kuvassa (a) tuolin heijastus näyttää litistyneen seinää vasten. Kuvassa (b) tarkempi heijastus on toteutettu kuvapuskuriheijastuksena. ....	38
Kuvio 12. Kuvio havainnollistaa etäisyyteen perustuvaa karheuden korjausta. ....	40

Kuvio 13. Kuviossa ympäristötutkaimen lokalisoinnin avulla saatu arvio etäisyyteen perustuvasta karheuden aiheuttamasta sumennuksesta. Kuvassa (a) karheuden korjaus ei ole käytössä ja kuvassa (b) etäisyyteen perustuva karheuden korjaus on käytössä. ....	41
Kuvio 14. Funktioiden $\hat{A}$ interpolointi funktioista $\hat{A}_\sigma$ , kun $\sigma = 0, 0.5, 1$ . ....	44
Kuvio 15. Kuvio esittää erotuskuvan arvion ja referenssin välillä. Erotuskuvien väriarvot on kaksinkertaistettu, jotta ero olisi paremmin havaittavissa. Kuvan (a) arviossa on kolme interpoloitavaa funktiota, joiden aste on neljä ( $k = 3, m = n = 4$ ). Kuvan (b) arviossa on neljä interpoloitavaa funktiota, joiden aste on neljä ( $k = 4, m = n = 4$ ). Kuva (c) on suodatettua painotettua otantaa käyttäen renderöity referenssi. Kuvien ylärivissä materiaali on metallinen ja alarivissä dielektrinen. Molemmilla diffuusi albedo on musta. ....	47
Kuvio 16. Funktioiden $\hat{A}$ ja $\hat{B}$ kuvaajat ja absoluuttiset virheet. Etenkin funktion $\hat{A}$ kuvaajasta on nähtävissä lineaarisen interpoloinnin vaikutus. ....	49
Kuvio 17. Kuvissa (a) ja (b) on käytetty ympäristön heijastusjakaumafunktion arviota. Kuvassa (a) ympäristöstä tulevan valon näyte otetaan peiliheijastussuunnasta kun taas kuvassa (b) näytteistetään normaalijakauman dominoivasta suunnasta. Kuva (c) on referenssikuva, joka on renderöity Monte Carlo -menetelmällä käyttäen suodatettua painotettua otantaa. Kuva (d) on kuvien (b) ja (c) välinen erotuskuva, jonka väriarvot on kaksinkertaistettu, jotta ero olisi paremmin havaittavissa. ....	50
Kuvio 18. Kuvassa (a) on pelkän suoran valaistuksen vaikutus. Kuvassa (b) näkyy heijastuspuskuri, joka on saatu kuvapuskuriheijastusten säteenseurantaa käyttäen. Kuvassa (c) suora ja epäsuora valaistus on yhdistetty naiivisti ottamatta huomioon heijastavan pinnan fysikaalisia ominaisuuksia. ....	52
Kuvio 19. Kuvio esittää kuvapuskuriheijastusten säteenseurannassa vastaan tulevat eri tilanteet. Kuvassa (a) heijastussäde törmää kuvapuskuriin. Kuvassa (b) säde kulkee kappaleen takana, mutta liian kaukana, että törmäys tapahtuisi edessä olevaan kappaleeseen. Kuvassa (c) säde törmää kappaleen etuseinään, vaikka todellinen törmäys tapahtuisi kappaleen takaseinään. Kuvassa (d) säde ei päädy kuvapuskurin alueelle. ....	53
Kuvio 20. Kuvassa on kuvapuskuriheijastusten tuottama heijastuspuskuri, jossa näkyy säteenseurannan harppauksesta johtuvia virheitä. Lattiassa heijastuvan pöydän välitaso ei heijastu kokonaan. ....	54
Kuvio 21. Kuvio esittää heijastusgeometrian, jossa piste $\mathbf{p}$ heijastuu pisteessä $\mathbf{p}'$ . Kuvapuskuriheijastuksissa pisteeseen $\mathbf{p}'$ pisteestä $\mathbf{p}$ saapuvaa radianssia arvioidaan valaistuspuskurissa olevalla pisteestä $\mathbf{p}$ kameraan saapuvalla radianssilla. ....	55
Kuvio 22. Kuvio esittää karheasta pinnasta katselusuuntaa heijastuvaa valoa kartiona, joka vastaa heijastusjakauman merkittävimpiä suuntia. ....	57
Kuvio 23. Kuviossa näkyy lähellä olevan puutason heijastuksen vuotaminen kauempana olevalle taustalle. Kuvan kontrastia on lisätty, jotta virhe tulisi selvemmin esille. ....	59

Kuvio 24. Kuviossa fysikaalisten kuvapuskuriheijastusten tulos. Lattian heijastuksissa on havaittavissa, kuinka materiaalin karheus vaikuttaa heijastusten sumeuteen. Lattian karheus arvot ovat $\sigma = 0.08, 0.24, 0.4$ . . . . .	60
Kuvio 25. Kuviossa lattian karheusarvo ei ole vakio, mistä johtuen heijastuksen sumeus vaihtelee. . . . .	61
Kuvio 26. Kuviossa näkyy kuvapuskuriheijastusten ja lokalisoitujen ympäristötutkaimien yhdistämisen tulos. . . . .	61
Kuvio 27. Kuviossa analyyttisen valon ja ympäristötutkaimen aiheuttamat heijastukset eivät täsmää, joten heijastus näkyy kahtena. . . . .	61

## Taulukot

Taulukko 1. Listaus luvussa käsitellyistä heijastusmenetelmistä. Tekniikkasarakkeessa on annettu tekniikka, johon menetelmä perustuu. EM tarkoittaa ympäristökartoitusta, GEO virtuaaligeometriaa ja RT säteenseurantaa. . . . .	24
Taulukko 2. Näytönohjaimella suoritettujen operaatioiden aikavaativuuden arvioita normalisoituna yhteenlaskuoperaation $A$ mukaan. Arviot on saatu suorittamalla eri operaatioita useita kertoja Nvidia Geforce GTX 960 -näytönohjaimella. . . . .	45
Taulukko 3. Funktioiden $\hat{A}$ ja $\hat{B}$ arvioitu aikavaativuus, maksimivirhe ja virheiden neliöllinen keskiarvo eri määrillä interpoloitavia funktioita ja funktioiden asteilla. Hintasarake esittää karkean arvion kokonaisvaativuudesta, jossa operaatioiden vaativuuksina on käytetty taulukossa 2 esitettyjä yhteenlaskun mukaan normalisoituja aikoja. . . . .	46
Taulukko 4. Funktioiden $A(\beta, \sigma)$ ja $B(\beta, \sigma)$ painotettua otantaa käyttäen laskettuihin datapisteisiin sovitettujen funktioiden, kolmen $\hat{A}_\sigma$ ja kolmen $\hat{B}_\sigma$ ( $\sigma = 0, 0.5, 1$ ) funktion, maksimivirhe ja virheiden neliöllinen keskiarvo, kun $\beta$ saa arvot väliltä $[0, 1]$ . . . . .	48
Taulukko 5. Funktioiden $\hat{A}(\beta, \sigma)$ , $\hat{B}(\beta, \sigma)$ ja $\hat{\rho}_{env}(\beta, \sigma, F_0 = 1)$ maksimivirhe ja virheiden neliöllinen keskiarvo, kun $\sigma$ saa arvot väliltä $[0, 1]$ ja $\beta$ saa arvot väliltä $[0, 1]$ . . . . .	48
Taulukko 6. Taulukko esittää renderöinnin eri vaiheisiin käytetyn ajan kuvaruutua kohden. . . . .	63

# Sisältö

1	JOHDANTO .....	1
2	VALON FYSIIKKAA .....	4
2.1	Maxwellin yhtälöt .....	4
2.2	Sähkömagneettinen aaltoliike.....	6
2.3	Polarisaatio .....	8
2.4	Valon taittuminen ja heijastuminen.....	9
2.5	Fresnel-yhtälöt.....	10
2.6	Interferenssi ja diffraktio.....	14
3	FYSIKAALINEN VALAISTUS .....	15
3.1	Radianssi ja irradianssi.....	15
3.2	Renderöintiyhtälö .....	16
3.3	Kaksisuuntainen heijastusjakaumafunktio .....	17
3.4	Mikrotahkomallit .....	19
4	HEIJASTUSMENETELMÄT .....	23
4.1	Tasoheijastukset .....	23
4.2	Ympäristökarttoihin perustuvat menetelmät .....	25
4.3	Muita kuvapohjaisia menetelmiä .....	28
4.4	Virtuaaligeometriaan perustuvat menetelmät .....	29
4.5	Säteenseuranta syvyyspuskuriin .....	30
5	METODI .....	33
5.1	Tekniikoiden valinta .....	33
5.2	Heijastusjakaumafunktion valinta .....	33
6	REAALIAIKAISET HEIJASTUKSET.....	35
6.1	Ympäristötutkaimet .....	35
6.1.1	Ympäristötutkaimien vaikutus ja asettelu .....	36
6.1.2	Ympäristökartan lokalisointi .....	37
6.1.3	Ympäristökartan käyttäminen fysikaalisessa valaistuksessa.....	40
6.2	Kuvapuskuriheijastukset .....	51
6.2.1	Kuvapuskuriheijastusten rajoitteita .....	51
6.2.2	Kuvapuskuriheijastukset fysikaalisessa valaistuksessa .....	55
6.3	Menetelmien yhdistäminen ja suorituskyky.....	59
7	JOHTOPÄÄTÖKSET.....	65
	LÄHTEET .....	67

# 1 Johdanto

Viime vuosina reaaliaikaisissa sovelluksissa, kuten peleissä, on nähty siirtymistä fysikaalisiin valaistusmalleihin. Tämä on huomattavissa esimerkiksi SIGGRAPH-konferenssissa järjestetystä kurssista, jossa useat pelialan ammattilaiset esittävät omia ratkaisujaan ja löytöjään fysikaaliseen valaistukseen siirtymisessä (McAuley ym. 2012, 2013; Hill ym. 2014, 2015). Syy fysikaalisten valaistusmallien käyttöönottoon on se, että fysikaalisesti oikeammat mallit tuottavat realistisempia kuvia kuin esimerkiksi tähän mennessä yleisesti käytetyt Phong- ja Blinn–Phong-mallit. Lisäksi materiaalit toistuvat eri valaistuksessa yhdenmukaisemmin, mikä helpottaa esimerkiksi tekstuuri- ja valaisusuunnittelijoiden työtä.

Tärkeä osa realistista valaistusta ovat ympäristön heijastukset (Kozlowski ja Kautz 2007). Heijastukset välittävät tietoa materiaalista, josta kappaleet koostuvat sekä kappaleiden sijainnista suhteessa toisiin kappaleisiin (Popescu ym. 2006; Roger ja Holzschuch 2006). Kaikki luonnolliset materiaalit peiliheijastavat valoa ainakin vähän ja osa materiaaleista, kuten tunnetusti metallit, heijastavat lähes kaiken niitä valaisevasta valosta. Osa tästä heijastuvasta valosta tulee suoraan valonlähteestä; osa taas on toisista kappaleista heijastunutta valoa. Yhden valonsäteen näkökulmasta valon heijastuminen on suhteellisen yksinkertainen ongelma, joka on ratkaistavissa säteenseuranta-algoritmeilla. Kokonaisuutena ilmiö on kuitenkin hyvin monimutkainen ja mahdoton ratkaista tarkasti reaaliajassa nykyisellä grafiikkalaitteistolla (Szirmay-Kalos ym. 2009).

On kuitenkin kehitetty monia approksimoivia tekniikoita, joilla heijastukset pystytään saavuttamaan uskottavasti myös reaaliajassa. Näistä tekniikoista ei ole paljon kokoavia tutkimustuloksia, jotka olisi julkaistu viime aikoina tai jotka olisivat erityisen kattavia. Erityisesti reaaliaikaisen ja fysikaalisen valaistuksen kontekstissa aihetta on käsitelty vähän, johtuen vasta viime vuosina tapahtuneesta fysikaalisiin valaistusmalleihin siirtymisestä reaaliaikaisessa grafiikassa.

Fysikaalinen varjostus (engl. *physically-based shading*), fysikaalinen valaistus (engl. *physically-based lighting*) ja fysikaalinen renderöinti (engl. *physically-based rendering*) ovat termejä, joilla tarkoitetaan hieman eri asioita, mutta joskus termejä käytetään myös

ristiin. Yleisesti tietokonegraafikassa varjostuksella (engl. *shading*) viitataan laskentaan, jolla selvitetään pinnan pisteestä lähtevä valo, joka on joko saapunut sille valonlähteestä tai on pinnan emittoima (Glassner 1995, s. 721). Valaistus taas viittaa valaistusmalliin, joka määrittää valonlähteiden luonteen. Renderöinti kattaa varjostuksen ja valaistuksen ohella kuvan muodostuksen. Termillä ”fysikaalinen” tarkoitetaan, että renderöinnin eri vaiheet suoritetaan joillain fysikaalisiin malleihin perustuvilla menetelmillä (Glassner 1995, s. 885).

Fysikaalisella renderöinnillä voi olla useita päämääriä, joista Shirley ym. (2012) mainitsevat kolme: ennustava renderöinti, uskottava renderöinti ja visuaalisesti rikas renderöinti. Näistä ennustavan renderöinnin tulisi tuottaa vastaava tulos renderöitävästä tilasta tai ympäristöstä, kuin jos se rakennettaisiin oikeasti. Tällaista renderöintiä käytetään yleensä suunnittelu- ja simulaatio-ohjelmissa. Uskottava renderöinti tuottaa realistisia kuvia, jotka kuitenkin voivat olla pielessä. Tätä käytetään esimerkiksi peleissä ja visualisaatio-ohjelmissa. Visuaalisesti rikkaasta renderöintiä käytetään, kun halutaan saavuttaa sama visuaalinen rikkaus kuin oikeassa maailmassa, mutta lopputulos saa kuitenkin olla tyylitelty. Tällaista käytetään esimerkiksi piirrosanimaatioissa.

Kappaleisiin vaikuttava valaistus voidaan jakaa kahteen osaan sen mukaan, kuinka se päätyy katselijan verkkokalvolle tai kameran kuvakennolle. Suora valaistus tulee suoraan valonlähteestä ja epäsuora valaistus on taas ympäristöstä heijastunutta valoa. Suora valaistus voidaan yleensä laskea analyttisesti, kun tiedetään, millainen valonlähde on kyseessä. Epäsuoralle valaistukselle ei kuitenkaan ole helppoa ratkaisua, sillä valo voi kimpoilla kappaleista tai taittua useasti, ennen kuin se päätyy kameraan. Lokaalit valaistusmallit (engl. *local illumination model*) ottavat huomioon vain suoran valaistuksen, mutta globaalit valaistusmallit (engl. *global illumination model*) mallintavat myös epäsuoraa valaistusta (esimerkiksi Greger ym. 1998; Glassner 1995, s. 725).

Tämän tutkielman tarkoituksena on esittää sopivia ratkaisuja uskottavaan renderöintiin reaaliaikaisissa sovelluksissa. Tällöin ollaan valmiita tinkimään laadusta reaaliaikaisen nopeuden saavuttamiseksi. Tutkielmassa käsitellään erityisesti epäsuoran heijastuneen valon mallintamista, eikä siis perehdytä kappaleen sisään taittuneeseen valoon. Teoriaosiossa esitetyt konseptit voidaan kuitenkin yleistää myös kappaleiden läpi kulkevan valon mallintamiseen. Tutkielmassa esiteltävät päätelmät ja tekniikoiden arviointi auttavat reaaliaikaisen grafiikka-

moottorin heijastustekniikkojen valitsemisessa ja toteuttamisessa.

Loppuosa tutkielmasta jakautuu seitsemään lukuun. Luvussa 2 käsitellään valon fysikaalisia ominaisuuksia ja kootaan ymmärrys valon ja materiaalin vuorovaikutuksesta. Luvussa 3 käsitellään fysikaalisen valaistuksen teoriaa, joka pohjautuu fysikaaliseen optikkaan. Seuraavaksi siirrytään reaaliaikaisten heijastusmenetelmien pariin. Luvussa 4 esitetään aiempaa tutkimusta reaaliaikaisista heijastusmenetelmistä. Luku 5 esittelee valittujen heijastustekniikoiden arvioinnissa käytettävät kriteerit. Luvussa 6 arvioidaan valittuja heijastustekniikoita ja esitetään yksityiskohtia niiden toteutuksesta fysikaalisen valaistuksen näkökulmasta ja lopuksi luvussa 7 esitetään arvioinnista saadut johtopäätökset.



## 2 Valon fysiikkaa

Valo koostuu eri aallonpituuksilla siirtyvästä energiasta. Tästä syystä valon sanotaan olevan spektraalista. Ihmisen silmän kolmenlaiset tappisolut ovat herkistyneet havaitsemaan valoa vain tiettyjen aallonpituuksien tienoilla. Nämä aallonpituudet vastaavat kutakuinkin punaista, vihreätä ja sinistä väriä, joiden aallonpituudet ovat noin 650nm, 530nm ja 450nm. Näiden kolmella aallonpituudella aistitun energian suhteellinen määrä saa aivot tulkitsemaan valon eri värisenä. Todellisuudessa valon spektrijakauma on jatkuva, mutta johtuen ihmisen rajallisesta kyvystä erotella värejä, riittää yleensä esittää valo käyttäen vain kolmea aallonpituutta vastaavaa komponenttia. Kun jatkossa käsitellään valoa, ellei erikseen ole toisin mainittu, oletetaan, että kyseessä on vain yhtä aallonpituutta  $\lambda$  sisältävää valoa. Kuva, joka koostuu useasta väristä, saadaan muodostettua, kun käytetään aallonpituuksia  $\lambda_R, \lambda_G, \lambda_B$ , jotka vastaavat punaista, vihreää ja sinistä väriä.

### 2.1 Maxwellin yhtälöt

Luvun esitys perustuu pääosin Knightin oppikirjaan (Knight 2008, s. 818–1084). Jotta voisimme ymmärtää fysikaalisen valaistuksen teoriaa, on hyvä aloittaa valon fysikaalisesta luonteesta. Valo koostuu sekä hiukkasista että sähkömagneettisesta säteilystä. Valon sähkömagneettista luonnetta kuvaavat Maxwellin yhtälöt. Näistä yhtälöistä on sekä integraalittain differentiaaliesitykset. Yhtälöissä esiintyvät vektorikentät sähkövuon tiheys  $\mathbf{D}$ , magneettivuon tiheys  $\mathbf{B}$ , sähkökenttä  $\mathbf{E}$ , magneettikenttä  $\mathbf{H}$  ja sähkövirran tiheys  $\mathbf{J}$ .

*Gaussin laki sähkökentille* integraaliesityksenä on

$$\oint_S \mathbf{D} \cdot \mathbf{n} \, da = \frac{Q_{in}}{\epsilon_0} = \oint_V \rho \, dv. \quad (2.1)$$

Integraalissa esiintyvä  $S$  on mielivaltainen pinta ja  $V$  on mielivaltainen tilavuus. Vektori  $\mathbf{n}$  on integroimispinnan ulkonormaali. Muuttuja  $Q_{in}$  on pinnan sisältämä kokonaisvaraus, vakio  $\epsilon_0$  on luonnonvakio tyhjiön permittiivisyys ja  $\rho$  on varaustiheys.

Gaussin laki sähkökentille (tai pelkästään Gaussin laki) kertoo, että suljetun pinnan yli in-

tegroimalla sähkökentän tiheyttä saadaan pinnan sisään jäämä kokonaisvaraus. Toisaalta integroimalla varaustiheyttä yli mielivaltaisen tilavuuden, saadaan tilavuuden sisältämä kokonaisvaraus. Yhtälön differentiaaliesitys puolestaan on

$$\nabla \cdot \mathbf{D} = \rho , \quad (2.2)$$

jonka tulkinta on yksinkertaisesti se, että varaustiheys riippuu sähkövuon läheisyydestä.

*Gaussin laki magneettikentille* on analoginen sähkökentille esitetyn lain kanssa. Sen integraaliesitys on

$$\oint_S \mathbf{B} \cdot \mathbf{n} \, da = 0 . \quad (2.3)$$

Gaussin laki magneettikentille esittää, että magneettivuon yli suljetun pinnan on aina nolla. Differentiaaliesitys yhtälöstä on

$$\nabla \cdot \mathbf{B} = 0 . \quad (2.4)$$

*Maxwell–Faradayn induktiolain* integraaliesitys on

$$\oint_C \mathbf{E} \cdot ds = - \frac{\partial}{\partial t} \oint_S \mathbf{B} \cdot \mathbf{n} \, da . \quad (2.5)$$

Yhtälössä  $C$  on mielivaltainen suljettu silmukka, joka rajaa pinnan  $S$ . Yhtälö voidaan tulkita siten, että suljettuun silmukkaan indusoituu virta, jos sen rajaaman pinnan läpi kulkeva magneettikenttä muuttuu ajan mukaan. Indusoitunut sähkökenttä on magneettikentän muutosta vastustava. Differentiaaliesitys yhtälöstä on

$$\nabla \times \mathbf{E} = - \frac{\partial \mathbf{B}}{\partial t} . \quad (2.6)$$

*Maxwell–Ampèren laki* integraaliyhtälönä on

$$\oint_C \mathbf{H} \cdot ds = \oint_S \mathbf{J} \cdot \mathbf{n} \, da + \oint_S \mathbf{D} \cdot \mathbf{n} \, da . \quad (2.7)$$

Yhtälö kertoo, että suljetun silmukan läpi kulkeva sähkövirta on magneettikentän viivaintegraali silmukan yli. Maxwell–Ampèren lain differentiaaliesitys on

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}, \quad (2.8)$$

joka osoittaa, että sähkövuon muutos saa aikaan magneettikentän.

Maxwellin yhtälöiden lisäksi sähkövuon tiheys  $\mathbf{D}$  ja sähkövirran tiheys  $\mathbf{J}$  voidaan esittää sähkökentän  $\mathbf{E}$  funktiona, ja magneettikentän tiheys  $\mathbf{B}$  sähkökentän  $\mathbf{H}$  funktiona (Räbinä 2006, s. 11)

$$\mathbf{D} = \mathbf{D}(\mathbf{E}), \quad \mathbf{B} = \mathbf{B}(\mathbf{H}), \quad \mathbf{J} = \mathbf{J}(\mathbf{E}). \quad (2.9)$$

Jos väliaine on isotrooppista, eli suunnasta riippumatonta, voidaan nämä kenttien riippuvuudet esittää väliaineen ominaisuuksilla muodossa

$$\mathbf{D} = \varepsilon \mathbf{E}, \quad \mathbf{B} = \mu \mathbf{H}, \quad \mathbf{J} = \sigma \mathbf{E}, \quad (2.10)$$

missä  $\varepsilon$  on permittiivisyys,  $\mu$  on permeabiliteetti ja  $\sigma$  on sähkönjohtavuus. Nämä muuttujat ovat väliaineen sähkömagneettisia ominaisuuksia ja voivat olla paikasta riippuvia. Jos väliaineen ominaisuudet ovat riippumattomia paikasta, sanotaan väliainetta homogeeniseksi.

Tyhjiö on homogeeninen ja sen sähkönjohtavuus ja varaustiheys ovat luonnollisesti nolla. Erikoistapaus Maxwellin yhtälöistä tyhjiössä voidaan esittää differentiaaliyhtälöinä

$$\nabla \cdot \mathbf{E} = 0 \quad (2.11)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (2.12)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (2.13)$$

$$\nabla \times \mathbf{B} = \varepsilon_0 \mu_0 \frac{\partial \mathbf{E}}{\partial t}. \quad (2.14)$$

## 2.2 Sähkömagneettinen aaltoliike

Seuraavaksi johdetaan Maxwellin yhtälöistä aaltoyhtälö, joka osoittaa että sähkömagneettinen säteily tosiaan on aaltoliikettä. Yleinen aaltoyhtälö on muotoa

$$\frac{1}{v^2} \frac{\partial^2 \phi}{\partial t^2} = \nabla^2 \phi, \quad (2.15)$$

missä  $v$  on aaltoliikkeen nopeus ja funktio  $\phi = \phi(\mathbf{x}, t)$  kuvaa aaltoliikkeen amplitudia pisteessä  $\mathbf{x}$  ajan hetkellä  $t$ . Yhtälössä  $\nabla^2$  on Laplacen operaattori

$$\nabla^2 \phi = \left( \frac{\partial^2 \phi}{\partial x_1^2} + \frac{\partial^2 \phi}{\partial x_2^2} + \dots + \frac{\partial^2 \phi}{\partial x_n^2} \right). \quad (2.16)$$

Oletetaan tyhjiö, jolloin voimme käyttää Maxwellin yhtälöistä tyhjiölle erikoistettuja muotoja (2.11–2.14).

Kun tiedetään, että vektorikentille pätee yhtälö

$$\nabla \times (\nabla \times \mathbf{A}) = \nabla(\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A}, \quad (2.17)$$

voidaan sitä soveltaa Maxwellin yhtälöihin. Otetaan ensin yhtälön (2.13) molemmista puolia roottori, jolloin saadaan

$$\nabla \times (\nabla \times \mathbf{E}) = \nabla \times \left( -\frac{\partial \mathbf{B}}{\partial t} \right). \quad (2.18)$$

Sovelletaan tähän yhtälöä (2.17), jolloin saadaan

$$\nabla(\nabla \cdot \mathbf{E}) - \nabla^2 \mathbf{E} = \nabla \times \left( -\frac{\partial \mathbf{B}}{\partial t} \right). \quad (2.19)$$

Soveltamalla yhtälöä (2.11),  $\nabla \cdot \mathbf{E} = 0$ , saadaan

$$-\nabla^2 \mathbf{E} = -\frac{\partial}{\partial t} (\nabla \times \mathbf{B}). \quad (2.20)$$

Lopulta soveltamalla termiin  $\nabla \times \mathbf{B}$  Maxwellin yhtälöä (2.14) saadaan sähkökentälle yhtälö

$$\varepsilon_0 \mu_0 \frac{\partial \mathbf{E}}{\partial t} = \nabla^2 \mathbf{E}, \quad (2.21)$$

mikä vastaa yleistä aaltoyhtälöä. Vastaavasti yhtälöstä (2.14) saadaan magneettivuon tiheydelle aaltoyhtälö

$$\varepsilon_0 \mu_0 \frac{\partial \mathbf{B}}{\partial t} = \nabla^2 \mathbf{B}. \quad (2.22)$$

Näistä aaltoyhtälöistä nähdään, että sähkömagneettisen säteilyn nopeus tyhjiössä on

$$c_0 = 1/\sqrt{\epsilon_0\mu_0} . \quad (2.23)$$

Vastaavasti pystytään osoittamaan, että valon nopeus väliaineessa on

$$c = 1/\sqrt{\epsilon\mu} , \quad (2.24)$$

kun väliaineen permittiivisyys on  $\epsilon$  ja permeabiliteetti  $\mu$ .

### 2.3 Polarisaatio

Sähkö- ja magneettikenttä ovat kohtisuorassa aallon etenemissuunnan kanssa. Lisäksi kentät ovat kohtisuorassa toisiaan vasten. Mikään ei kuitenkaan määrää, että sähkö- ja magneettikentän suuntien olisi oltava vakio tasolla, jonka normaali on aaltoliikkeen etenemissuunta. Perinteisesti polarisaatio määritellään sähkökentän mukaan. Sähkökenttä voidaan jakaa kahdeksi sähkökentän kanssa samalla tasolla olevaksi kohtisuoraksi komponentiksi. Sähkömagneettisen säteilyn polarisaation määrää vaihe-ero, joka näiden kahden komponentin suuruksien välillä on. Näitä kahta vektoria kutsutaan polarisaatioakseleiksi ja sattumanvaraisesti toista nimitetään yhdensuuntaiseksi ja toista kohtisuoraksi akseliksi.

Näille akseleille projisoidut sähkökentän komponentit ovat  $\mathbf{E}_{\parallel}$  yhdensuuntaiselle akselille ja  $\mathbf{E}_{\perp}$  kohtisuoralle akselille. Yksittäinen aalto on aina polarisoitunutta, mutta useasta aallosta koostuva säteily on aaltojen polarisaation summa. Tämä johtuu siitä, että sähkömagneettinen säteily noudattaa superpositioperiaatetta. Kun säteilyn sähkökentän komponentit  $\mathbf{E}_{\parallel}$  ja  $\mathbf{E}_{\perp}$  korreloivat toisiaan ajassa, on säteily polarisoitunutta. Jos näillä komponenteilla ei ole korrelaatiota, on säteily polarisoimatonta. Toisaalta säteily voi olla osittain polarisoitunutta, jolloin se sisältää polarisaatioltaan vaihtelevaa aaltoliikettä, josta riittävä osa on kuitenkin samankaltaisesti polarisoitunutta. Säteilyn  $\mathbf{E}_{\parallel}$  komponentin suuntaisesti polarisoitunutta osaa kutsutaan usein p-polarisoituneeksi (enlg. *parallel*) ja  $\mathbf{E}_{\perp}$  komponentin suuntaisesti polarisoitunutta s-polarisoituneeksi (saks. *senkrecht*, kohtisuora).

Tavalliset valonlähteet tuottavat polarisoimatonta valoa, mutta esimerkiksi laserin tuottama valo sisältää hyvin tarkasti polarisoitunutta aaltoliikettä. Lisäksi esimerkiksi auringon valo, joka kulkee ilmakehän läpi, polarisoituu osittain.

## 2.4 Valon taittuminen ja heijastuminen

Kun valo kulkee homogeenisessa väliaineessa, kulkee se suoraviivaisesti. Jos valo kohtaa materiaalin rajapinnan, se joko taittuu tai heijastuu. Taitekerroin, materiaalin sähkömagneettinen ominaisuus, on vastuussa materiaalin vaikutuksesta valon kulkuun. Se on valon aallonpituudesta riippuva kompleksiluku

$$N(\lambda) = \eta(\lambda) + j\kappa(\lambda) , \quad (2.25)$$

missä reaaliosa  $\eta$  on yksinkertainen tai reaalinen taitekerroin ja reaaliluku  $\kappa$  on niin kutsuttu vaimennuskerroin. Reaaliarvoinen taitekerroin,

$$\eta = c_0/c , \quad (2.26)$$

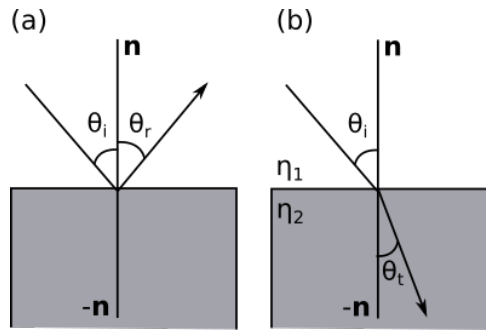
kuvaava valon nopeutta tyhjiössä,  $c_0$ , suhteessa valon nopeuteen väliaineessa,  $c$ . Koska valon nopeus tyhjiössä on suurin mahdollinen nopeus, täytyy olla  $\eta \geq 1$ . Kun kahden väliaineen, 1 ja 2, reaaliarvoiset taitekertoimet ovat  $\eta_1$  ja  $\eta_2$ , sanotaan, että väliaine 1 on optisesti harvempi kuin väliaine 2, jos  $\eta_1 < \eta_2$ . Vastaavasti sanotaan, että väliaine 2 on optisesti tiheämpää kuin väliaine 1.

Valon taittuminen rajapinnassa selittyy valon nopeuden muutoksella. Kun valon etenemisnopeus pienenee, säilyy kuitenkin aaltoliikkeen jakson aika samana, joten aallonpituuden täytyy lyhentyä. Kun valo tulee rajapintaan nolasta poikkeavassa kulmassa, samaan aaltorintamaan kuuluvat säteet kohtaavat rajapinnan eri aikaan. Kun kahta eri aikaan rajapinnan kohdannutta sädettä tarkastellaan, näistä toinen on kulkenut lyhemmän matkan uudessa väliaineessa. Tällöin aaltorintama on vaihtanut suuntaansa, toisin sanoen valo on taittunut.

Kuviossa 1 on esitetty valon heijastuminen ja taittuminen. Kuviossa  $\mathbf{n}$  on pinnan normaali,  $\theta_i$  on valon tulokulma,  $\theta_r$  on heijastuskulma,  $\theta_t$  on taitekulma. Lisäksi  $\eta_1$  on pinnan yläpuolisen ja  $\eta_2$  on pinnan alapuolisen materiaalin taitekerroin.

$$\theta_i = \theta_r \quad (2.27)$$

$$\eta_1 \sin \theta_i = \eta_2 \sin \theta_t \quad (2.28)$$



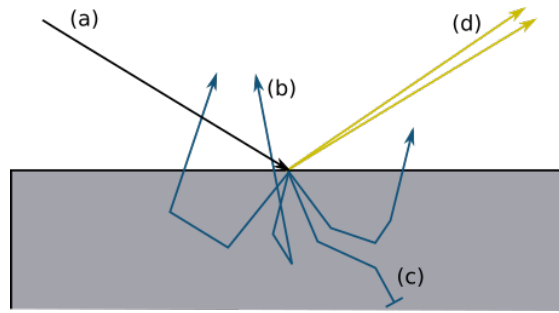
Kuvio 1. (a) Valon säteen heijastuminen. (b) Valon säteen taittuminen.

Heijastuslaki (2.27) sanoo, että valon säteen tulokulma on yhtä suuri kuin heijastuskulma. Taittumislaki, joka tunnetaan myös Snellin lakina (2.28), taas sitoo tulokulman ja taitekulman toisiinsa materiaalien taitekertoimiin perustuvalla suhteella.

Kun pinnan mikroskooppiset muodot ovat pienempiä kuin näkyvän valon aallonpituus ( $< 300\text{nm}$ ), sanotaan että pinta on optisesti tasainen. Tällöin pintaa voidaan geometrisen optiikan tasolla käsitellä kuin se olisi täysin tasainen. Koska mikään luonnollinen pinta ei ole optisesti tasainen, vaikka geometrisen optiikan näkökulmasta näin olisikin. Kuviossa 2 on esitetty valon heijastumista ja taittumista optisesti epätasaiselta pinnalta. Kuviossa näkyy, kuinka osa valosta taittuu pinnan alle, kimpoilee väliaineen hiukkasista ja päätyy taittumaan takaisin pinnan ylle. Osa pinnan alle taittuneesta valosta absorboituu. Osa valosta taas heijastuu peilimäisesti pinnasta. Koska pinnan pisteestä heijastuva valo voi olla sekä peiliheijastunutta että pinnan alta taittumisen myötä takaisin heijastuvaa valoa, jaotellaan nämä heijastuneen valon osat yleensä omiksi termeikseen. Näitä termejä kutsutaan nimillä peiliheijastunut eli spekulaarinen termi ja hajaheijastunut eli diffuusi termi.

## 2.5 Fresnel-yhtälöt

Tämän luvun sisältö perustuu lähinnä Lvovskyn artikkeliin (Lvovsky 2013). Niin kutsuttu Fresnel-termi kertoo, kuinka suuri osa valosta heijastuu tai taittuu, kun valo kohtaa väliaineiden rajapinnan. Tämä termi riippuu valon tulokulmasta ja väliaineiden taitekertoimista. Lisäksi Fresnel-heijastus riippuu valon polarisaatiosta. Kuvio 3 esittää polarisoituneen valon heijastumisen geometrian. Kuviossa pinnalle tulevan valon polarisaatioakselit ovat  $\hat{i}_{\parallel}$  p-



Kuvio 2. Valon heijastuminen, taittuminen ja absorboituminen optisesti epätasaisesta pinnasta. (a) Pintaan saapuu valoa. (b) Valoa taittuu pinnan alle ja osa tästä valosta taittuu takaisin pinnan yläpuolelle kimpoiltuaan ensin väliaineen hiukkasista. (c) Osa pinnan alle taittuneesta valosta absorboituu. (d) Valoa heijastuu peilimäisesti pinnasta. Koska pinta ei ole täysin tasainen, ei kaikki valo heijastu täysin samaan suuntaan.

polarisoituneelle ja  $\mathbf{i}_\perp$  s-polarisoituneelle sähkökentälle. Vastaavasti heijastuneen valon polarisaatioakselit ovat  $\mathbf{r}_\parallel$  ja  $\mathbf{r}_\perp$ .

Fresnel-termi on määritelty eri tavoin p- ja s-polarisoituneelle valolle. Lisäksi heijastuneelle valolle termi on eri kuin taittuneelle valolle. Fresnel-yhtälöitä on siis yhteensä neljä. Jos  $E_i$  on saapuvan,  $E_r$  on heijastuneen ja  $E_t$  on taittuneen sähkökentän amplitudi, niin heijastavuuskerroin sähkökentän amplitudille on tällöin

$$F_r = E_r/E_i, \quad (2.29)$$

ja läpäisyvyyskerroin sähkökentän amplitudille on

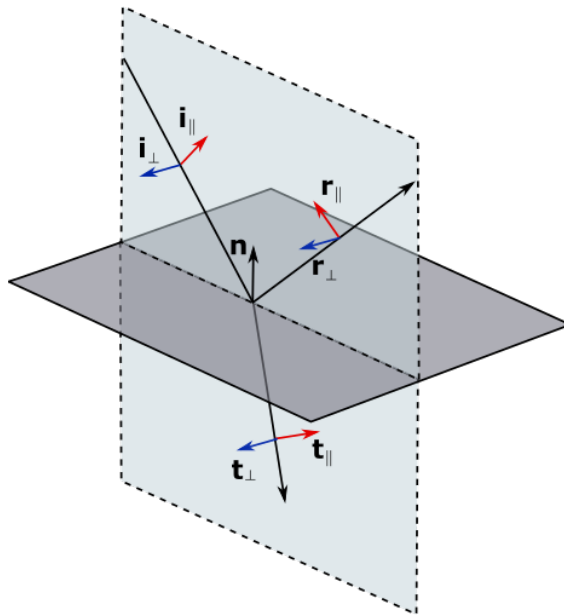
$$F_t = E_t/E_i. \quad (2.30)$$

Fresnel-termi p-polarisoituneelle heijastuneelle valolle on

$$F_{r\parallel} = \frac{(\eta_1/\mu_1) \cos \theta_t - (\eta_2/\mu_2) \cos \theta_i}{(\eta_1/\mu_1) \cos \theta_t + (\eta_2/\mu_2) \cos \theta_i}, \quad (2.31)$$

missä  $\eta_1$  ja  $\eta_2$  ovat väliaineiden reaaliset taitekertoimet ja  $\mu_1$  ja  $\mu_2$  ovat väliaineiden permeabiliteetit. Kulmat  $\theta_i$  ja  $\theta_t$  ovat valon tulokulma ja taittumiskulma. Fresnel-termi s-polarisoituneelle heijastuneelle valolle on





Kuvio 3. Kuviossa on esitetty polarisaation geometria valon heijastuessa ja taittuessa. Vektorit  $\mathbf{i}_{\parallel}$  ja  $\mathbf{i}_{\perp}$  ovat rajapintaan tulevan valon yhdensuuntainen ja kohtisuora polarisaatioakseli. Heijastuneen valon yhdensuuntainen polarisaatioakseli on  $\mathbf{r}_{\parallel}$  ja kohtisuora polarisaatioakseli  $\mathbf{r}_{\perp}$ . Vastaavasti  $\mathbf{t}_{\parallel}$  ja  $\mathbf{t}_{\perp}$  ovat taittuneen valon yhdensuuntainen ja kohtisuora polarisaatioakseli.

$$F_{r\perp} = \frac{(\eta_1/\mu_1) \cos \theta_i - (\eta_2/\mu_2) \cos \theta_t}{(\eta_1/\mu_1) \cos \theta_i + (\eta_2/\mu_2) \cos \theta_t} . \quad (2.32)$$

Taittuneelle p-polarisoituneelle valolle Fresnel-termi on

$$F_{t\parallel} = \frac{2(\eta_1/\mu_1) \cos \theta_i}{(\eta_1/\mu_1) \cos \theta_i + (\eta_2/\mu_2) \cos \theta_t} . \quad (2.33)$$

Vastaavasti s-polarisoituneelle taittuneelle valolle Fresnel-termi on

$$F_{t\perp} = \frac{2(\eta_1/\mu_1) \cos \theta_i}{(\eta_1/\mu_1) \cos \theta_i + (\eta_2/\mu_2) \cos \theta_t} . \quad (2.34)$$

Koska suurin osa arkipäiväisistä materiaaleista ei ole magnetisoituneita, voidaan Fresnel-yhtälöitä yksinkertaistaa olettamalla, että  $\mu_1 = \mu_2 = \mu_0$ , missä  $\mu_0$  on tyhjiön permeabiliteetti.

Nyt Fresnel-termi heijastuneelle p-polarisoituneelle valolle on

$$F_{r\parallel} = -\frac{\tan(\theta_i - \theta_t)}{\tan(\theta_i + \theta_t)} , \quad (2.35)$$

ja Fresnel-termi heijastuneelle s-polarisoituneelle valolle on

$$F_{r\perp} = -\frac{\sin(\theta_i - \theta_t)}{\sin(\theta_i + \theta_t)} . \quad (2.36)$$

Vastaavasti Fresnel-termi taittuneelle p-polarisoituneelle valolle yksinkertaistuu muotoon

$$F_{t\parallel} = \frac{2 \sin \theta_t \cos \theta_i}{\sin(\theta_i + \theta_t) \cos(\theta_i - \theta_t)} , \quad (2.37)$$

ja Fresnel-termi taittuneelle s-polarisoituneelle valolle yksinkertaistuu muotoon

$$F_{t\perp} = \frac{2 \sin \theta_t \cos \theta_i}{\sin(\theta_i + \theta_t)} . \quad (2.38)$$

Esitetyt Fresnel-yhtälöt (2.31–2.34) ovat heijastavuus- ja läpäisykertoimia sähkömagneettisen säteilyn amplitudille. Kuitenkin käytännössä tarvitsemme yleensä kertoimet intensiteetille. Nämä kertoimet saadaan, kun tiedetään, että epämagneettisessa väliaineessa, jonka reaalinen taitekerroin on  $\eta$ , amplitudin  $E$  ja intensiteetin  $I$  välillä pätee yhtäsuuruus

$$I = 2\eta c_0 \varepsilon_0 |E|^2 , \quad (2.39)$$

missä  $c_0$  on valon nopeus tyhjiössä ja  $\epsilon_0$  on tyhjiön permittiivisyys. Nyt Fresnel-heijastavuuskertoimille pätee

$$F_R = I_r/I_i = \frac{2\eta_1 c_0 \epsilon |E_r|^2}{2\eta_1 c_0 \epsilon |E_i|^2} = \left(\frac{E_r}{E_i}\right)^2 = F_r^2. \quad (2.40)$$

Tästä yhtäsuuruudesta saadaan intensiteetin heijastavuuskertoimille yhtälöt

$$F_{R\parallel} = F_{r\parallel}^2 = \frac{\tan^2(\theta_i - \theta_t)}{\tan^2(\theta_i + \theta_t)}, \quad (2.41)$$

$$F_{R\perp} = F_{r\perp}^2 = \frac{\sin^2(\theta_i - \theta_t)}{\sin^2(\theta_i + \theta_t)}. \quad (2.42)$$

Taittuneelle valolle intensiteetin läpäisyvyyskertoimien yhtälöt puolestaan ovat

$$F_{T\parallel} = 1 - F_{R\parallel} = \frac{4 \sin \theta_i \sin \theta_t \cos \theta_i \cos \theta_t}{\sin^2(\theta_i + \theta_t) \cos^2(\theta_i - \theta_t)}, \quad (2.43)$$

$$F_{T\perp} = 1 - F_{R\perp} = \frac{4 \sin \theta_i \sin \theta_t \cos \theta_i \cos \theta_t}{\sin^2(\theta_i + \theta_t)}. \quad (2.44)$$

## 2.6 Interferenssi ja diffraktio

Yksi aaltoliikkeen fundamentaali ominaisuus on usean aallon interferenssi. Kun kahden aaltoliikkeen poikkeamat ovat pisteessä saman suuntaiset, vahvistavat ne toisiaan. Jos aaltoliikkeiden poikkeamat ovat erisuuntaiset, heikentävät ne toisiaan. Tämä aaltojen superpositio-periaate pätee myös valolle.

Kun valo heijastuu kahdesta eri kerroksesta, esimerkiksi lätäkön pinnalla olevasta ohuesta öljykerroksesta, aiheuttaa interferenssi heijastuneelle valolle öljyn pinnalle ominaista värjäytymistä. Tämä johtuu siitä, että osa valosta heijastuu eri kerroksesta kuin toinen osa, jolloin näiden heijastuvien valojen interferenssi joko vahvistaa tai heikentää tietyllä aallonpituudella kulkevaa valoa.

Diffraktio on toinen ilmiö, joka on aaltoliikkeelle ominainen. Kun aaltorintama kulkee esteen ohitse, ei se kulje suoraviivaisesti, vaan taittuu esteen taakse. Esimerkiksi varjojen pehmeys johtuu osittain diffraktiosta. Interferenssistä ja diffraktiosta saa lisätietoa Knightin oppikirjasta (Knight 2008, s. 634–691).

### 3 Fysikaalinen valaistus

Kun valoa mallinnetaan käyttäen sen sähkömagneettista aaltoluonnetta ja Maxwellin yhtälöitä, puhutaan aalto-optiikasta. Näin valon käyttäytymistä voidaan mallintaa tarkemmin ottaen huomioon diffraktio, interferenssi ja polarisaatio. Tämä on kuitenkin hyvin raskasta, joten tietokonegrafiikassa tyydytään yleensä käyttämään yksinkertaisempaa mallia, niin kutsuttua *sädeoptiikkaa* (tai *geometrinen optiikka*). (Glassner 1995, s. 563)

Jatkossa keskitymme vain sädeoptiikkaan ja sivuutamme interferenssin, diffraktion ja polarisaation. Perusteena on yksinkertaisesti se, että nämä ilmiöt ovat liian harvinaisia ja vaikuttavat vain vähän lopulliseen kuvaan. Tällä päätöksellä menetämme tarkkuutta, mutta lopullinen valon mallinnus tulee olemaan nopeampaa ja soveltuu siis paremmin toteutettavaksi reaaliajassa.

#### 3.1 Radianssi ja irradianssi

Radiometria antaa peruskäsitteet ja suuret mitatulle valoenergialle. Kuten Glassner (1995, s. 648) mainitsee, radiometrian historiassa erilaiset merkintätavat ovat vaikeuttaneet eri lähteistä tulevien artikkelien ymmärtämistä. Lisäksi radiometrian perussuureet ovat suunnallisia, mutta niitä käsitellään silti lähinnä skalaariarvoisina, mikä vaikeuttaa niiden ymmärtämistä ja niille tehtävien operaatioiden esittämistä. Mukailen seuraavissa merkinnöissä Glassnerin käyttämiä.

Radiometrian suureista tärkeimpiä ovat radianssi, irradianssi, intensiteetti ja säteilyvuo (engl. *radiant flux*, *radiant power*). Säteilyvuo  $\Phi$  on tietyn alan läpivirtaava säteily energia  $Q$  aikayksikköä kohden,

$$\Phi = dQ / dt \quad [W = J/s]. \quad (3.1)$$

Intensiteetti  $I$  on säteilyvuo avaruuskulmayksikköä kohden,

$$I = d\Phi / d\vec{\omega} \quad [W/sr]. \quad (3.2)$$

Irradianssi  $E$  on säteilyvuo pinta-alayksikköä kohden,

$$E = d\Phi / dA \quad [W/m^2]. \quad (3.3)$$

Radianssi  $L$  kertoo pinnalle saapuvan tai siltä poistuvan säteilytehon määrän avaruuskulmayksikköä ja projisoitua pinta-alayksikköä kohden,

$$L = \frac{d^2\Phi}{d\vec{\omega} dA \cos \theta} \quad [W/(sr \cdot m^2)]. \quad (3.4)$$

Radianssin määritelmässä  $dA \cos \theta$  on projisoitu pinta-ala, joka viittaa joko valonlähteen pinta-alaan, kun kyseessä on lähteen emittoima radianssi, tai vastaanottavan pinnan pinta-alaan, kun kyseessä on pinnalle saapuva radianssi.

## 3.2 Renderöintiyhtälö

Renderöintiyhtälö (Kajiya 1986; Immel, Cohen ja Greenberg 1986) on keskeisessä roolissa tietokonegraafikassa. Se kertoo pinnan pisteestä  $\mathbf{x}$  tiettyyn suuntaan lähtevän radianssin. Renderöintiyhtälö esitetään yleisessä muodossa

$$L_o(\mathbf{x}, \boldsymbol{\omega}_o, \lambda, t) = L_e(\mathbf{x}, \boldsymbol{\omega}_o, \lambda, t) + \int_{\Omega} \rho(\mathbf{x}, \boldsymbol{\omega}_o, \boldsymbol{\omega}_i, \lambda, t) L_i(\mathbf{x}, \boldsymbol{\omega}_i, \lambda, t) \cos \theta d\boldsymbol{\omega}_i, \quad (3.5)$$

missä  $\boldsymbol{\omega}_o$  on pinnan pisteestä pois päin lähtevän radianssin suunta, yleensä katselusuunta, ja  $\boldsymbol{\omega}_i$  on tulevan valon suunta. Muuttuja  $\lambda$  on valon aallonpituus ja  $t$  on ajan hetki.  $L_o$  on pisteestä pois lähtevä radianssi ja  $L_i$  on pisteeseen  $\mathbf{x}$  saapuva radianssi. Joukko  $\Omega$  kattaa kaikki pisteessä  $\mathbf{x}$  nähtävissä olevat suunnat<sup>1</sup>. Funktio  $\rho$  on niin kutsuttu kaksisuuntainen heijastusjakaumafunktio, eli BRDF (engl. *bidirectional reflectance distribution function*), joka kertoo miten valo vuorovaikuttaa pinnan kanssa, ja  $L_e$  on pinnan pisteen itsensä emittoima valo. Kulma  $\theta$  on pinnan normaalin ja tulevan valon suunnan välinen. Kosinitermi  $\cos \theta$  painottaa tulevan valon määrää riippuen siitä, kuinka paljon pintaa on nähtävissä valon suunnasta. Termin alla olevalla viivalla tarkoitetaan, että tulevan valon suunnat voivat olla vain pinnan yläpuolella, eli  $\cos \theta \geq 0$ . Toisin sanoen kaikki pinnan pisteestä nähtävät valon tulosuunnat,

1. Näin esiteltyinä renderöintiyhtälö sisältää implisiittisesti näkyvyystermin.

$\Omega$ , sisältää vain pinnan yläpuoliset suunnat<sup>2</sup>. On hyvä huomata, että usein renderöintiyhtälössä ei eksplisiittisesti esitetä pistettä  $\mathbf{x}$ , aallonpituutta  $\lambda$  ja ajanhetkeä  $t$ , vaikka nämä yhtälöön kuuluvatkin.

Reflektanssiyhtälö (3.6) on renderöintiyhtälön erikoistapaus, jossa pinnan emittoima valo jätetään huomioimatta.

$$L_o(\boldsymbol{\omega}_o) = \int_{\Omega} \rho(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) L_i(\boldsymbol{\omega}_i) \cos \theta \, d\boldsymbol{\omega}_i \quad (3.6)$$

Tästä lähin käsittelemme vain heijastunutta valoa, joten unohtamme emittoidun valon ja keskitymme pelkkään reflektanssiyhtälöön.

### 3.3 Kaksisuuntainen heijastusjakaumafunktio

Kaksisuuntainen heijastusjakaumafunktio, BRDF (engl. *bidirectional reflectance distribution function*), on reflektanssiyhtälössä (3.6) esiintynyt funktio  $\rho(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i)$ . Se kuvaa pinnan ja valon välistä suhdetta. Kaksisuuntaisen heijastusjakaumafunktion toiminta voidaan ajatella intuitiivisesti kahdella tapaa. Ensimmäinen tulkinta on, että funktio kertoo, kuinka paljon pinnan pisteeseen saapuvista suunnista  $\boldsymbol{\omega}_i$  heijastuu valoa suuntaan  $\boldsymbol{\omega}_o$ . Toinen, yhtä pätevä tulkinta on, että funktio esittää pinnasta suuntiin  $\boldsymbol{\omega}_i$  heijastuneen ja sironneen valon suhteellisen määrän yhdelle suunnasta  $\boldsymbol{\omega}_o$  tulleelle valon säteelle.

Jos valon määrän jakauma ei riipu valon suunnasta tai katselusuunnasta, on heijastusjakauma *isotrooppinen*. Useimmat materiaalit voidaan mallintaa isotrooppisina, mutta toiset materiaalit vaativat *anisotrooppisia* malleja, jotta niille ominaiset piirteet voidaan esittää luotettavasti. Anisotrooppista mallia tarvitaan esimerkiksi hiusten tai harjatun metallin renderöintiin.

Realistisen fysikaalisen heijastusjakauman ominaisuuksia ovat positiivisuus, energian säilyminen ja vastavuoroisuus (engl. *reciprocity*). Positiivisuus tarkoittaa, että heijastusjakaumafunktion arvo on aina positiivinen, mikä tulee siitä, että valo ei voilla negatiivista. Energian säilyminen on intuitiivisesti selvää; pisteestä lähtevän valon määrä ei voi olla suurempi kuin

---

2. Tämä on yleensä selvää, mutta joskus yhtälöä arvioitaessa pinnan pisteen lokaalia normaalia on muutettu, esimerkiksi normaalikartoituksella (engl. *normal mapping*) siten, että se osoittaaakin pinnan alle. Tällöin kosinitermi olisi negatiivinen, joka ei ole fysikaalisesti mielekäästä, joten arvo rajoitetaan välille  $[0, 1]$ .

siihen saapuneen valon. Matemaattisesti tämä esitetään epäyhtälöllä

$$\int_{\Omega} \rho(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) \cos \theta \, d\boldsymbol{\omega}_i \leq 1, \quad \forall \boldsymbol{\omega}_o \in \Omega. \quad (3.7)$$

Vastavuoroisuus, tarkemmin Helmholtzin vastavuoroisuusperiaate, johtuu valon fysikaalisista ominaisuuksista. Sen mukaan heijastusjakauman tulee olla symmetrinen suuntien  $\boldsymbol{\omega}_1$ :n ja  $\boldsymbol{\omega}_2$ :n suhteen,

$$\rho(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2) = \rho(\boldsymbol{\omega}_2, \boldsymbol{\omega}_1), \quad \forall \boldsymbol{\omega}_1, \boldsymbol{\omega}_2 \in \Omega. \quad (3.8)$$

Tätä periaatetta hyödynnetään esimerkiksi kaksisuuntaisessa säteenseurannassa. Periaate toimii, jos materiaali ainoastaan heijastaa valoa. Muussa tapauksessa, kun valo taittuu materiaalin rajapinnassa, tulee käyttää yleisempää vastavuoroisuusperiaatetta, joka ottaa huomioon materiaalien taitekertoimet (Veach 1997).

Yksinkertaisin heijastusjakaumafunktio on vakiofunktio,

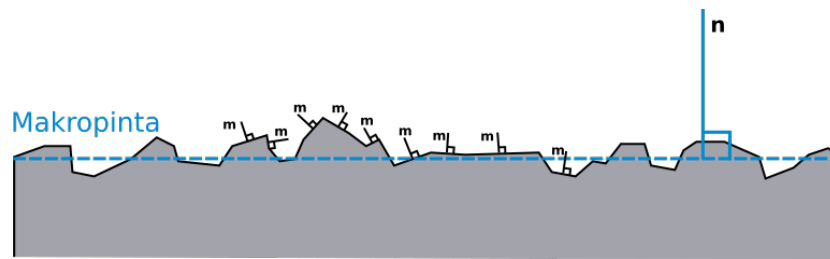
$$\rho_{\text{Lambert}}(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) = \frac{\alpha}{\pi}, \quad (3.9)$$

joka tunnetaan Lambertin diffuusina heijastusmallina. Yhtälön (3.9) vakio  $\alpha = \alpha_\lambda$  on niin kutsuttu albedo, jonka arvo on väliltä  $[0, 1]$ . Se kuvaa paljonko materiaali kyseisessä pisteessä heijastaa valoa (jonka aallonpituus on  $\lambda$ ). Muun osan valosta materiaali absorboi. Korjaustermi  $\frac{1}{\pi}$  tulee energian säilymisestä. Olkoon pinnan albedo  $\alpha = 1$  ja  $\boldsymbol{\omega}_o$  on pinnan normaalin suunta, jolloin energian säilymis-integraali saa suurimman arvonsa. Tällöin epäyhtälöstä (3.7) saadaan Lambertin vakio heijastusjakaumafunktiolle

$$\int_{\Omega} \frac{\alpha}{\pi} \cos \theta \, d\boldsymbol{\omega}_i = \frac{1}{\pi} \cdot \pi = 1 \leq 1, \quad (3.10)$$

joten energia säilyy.

Heijastusjakaumafunktion rajoite on se, ettei se ota huomioon valon taittumista ja valon siirtymistä väliaineessa. Jos näitä ilmiöitä halutaan mallintaa, käytetään tähän kaksisuuntaista transmianssin jakaumafunktiota, BTDF (engl. *bidirectional transmittance distribution function*). Yhdessä heijastusjakauma- ja transmianssin jakaumafunktio muodostavat kaksi-



Kuvio 4. Profilikuva mikrotahkoteorian mukaisesta pinnasta. Kuviossa on esitetty muutamien mikrotahkojen normaaleja  $m$  ja makropinnan normaali  $n$ .

suuntaisen sirontajakaumafunktion, BSDF (engl. *bidirectional scattering distribution function*).

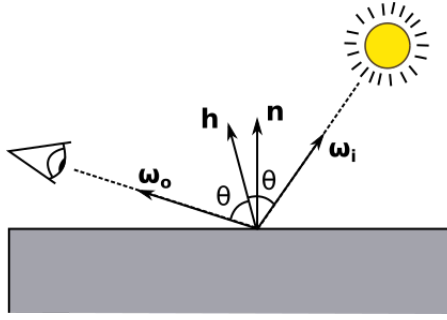
Valon mallintaminen käyttäen sirontajakaumafunktiota toimii samoin kuin valon mallintaminen heijastusjakaumafunktiolla. Erona on se, että mallinnetaan eri osaa valosta. Se osa valosta, joka ei heijastu suoraan pinnasta, taittuu pinnan alle. Valo matkaa pinnan alla kunnes joko absorboituu tai taittuu takaisin ulos kappaleesta — potentiaalisesti eri pisteestä, josta se taittui kappaleen sisään. Lopulta kokonaisen sirontajakauman laskeakseen tarvitsee vain heijastusjakauma- ja transmianssin jakaumafunktioiden arvot summata. Tutkielmassa keskitytään vain valon heijastumiseen, joten jatkossa käsitellään vain heijastusjakaumafunktioita.

### 3.4 Mikrotahkomallit

Yleensä pelkkä vakio-BRDF ei riitä esittämään materiaalin ominaisuuksia uskottavasti. Varsinkin peiliheijastava materiaali vaatii tarkemman mallin, jotta sen katselusuunnasta ja valon tulosuunnasta riippuvat ominaisuudet saadaan mallinnettua. Mikrotahkoteoriaan (engl. *microfacet theory*) (Torrance ja Sparrow 1967) perustuvat heijastusjakaumafunktiot ovat yleisiä peiliheijastavien materiaalien esittämisessä. Teorian mukaan kappaleen pinta koostuu pienistä peilimäisistä heijastustasoista, mikrotahkoista. Tämä on esitetty kuviossa 4. Tämän lisäksi kuviossa näkyy makropinta: makroskooppisen mittakaavan pinta, joka approksimoi oikeaa mikropintaa.

Sen sijaan, että kappaleen pinnan mikrogeometriaa käsiteltäisiin omana geometrianaan, mik-





Kuvio 5. Kuviossa on esitetty puolivektorin  $\mathbf{h}$  geometria. Vektorien  $\boldsymbol{\omega}_i$  ja  $\mathbf{h}$  välinen kulma  $\theta$  on sama kuin vektorien  $\boldsymbol{\omega}_o$  ja  $\mathbf{h}$ .

rotahkoteoria käsittelee mikropinnan muotoja makroskooppisella tasolla tilastollisesti. Tällöin oletetaan, että pintaa tarkastellaan riittävän kaukaa, niin että mikrogeometria ei ole nähtävissä. Tämän mikroskooppisen geometrian koon tulee olla suurempi kuin valon aallonpituus, mutta pienempi kuin makrogeometria.

Mikrotahkoteorian mukaan vain ne mikrotahkot, joiden normaalin suunta on täsmälleen niin kutsutun puolivektorin (engl. *half-vector*) suunta, vaikuttavat pinnan valaisuun. Puolivektori,

$$\mathbf{h} = \hat{\mathbf{h}} = \frac{\boldsymbol{\omega}_i + \boldsymbol{\omega}_o}{\|\boldsymbol{\omega}_i + \boldsymbol{\omega}_o\|}, \quad (3.11)$$

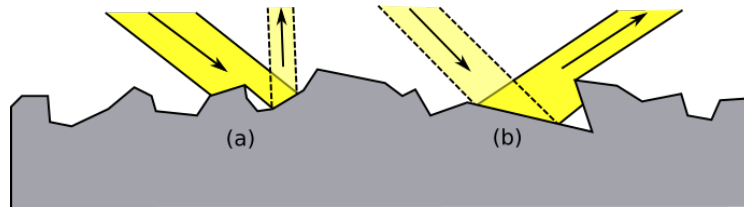
on yksikkövektori, joka on valon tulosuunnan ja katselusuunnan puolivälissä. Puolivektorin geometria on esitetty kuviossa 5.

Mikrogeometria voi vähentää heijastuvan valon määrää kahdella tavalla. Se voi joko varjostaa pinnalle tulevaa valoa tai se voi peittää pinnalta lähtevää valoa. Tämä on havainnollistettu kuviossa 6. Lisäksi osa valosta saattaa heijastua mikrogeometriasta useita kertoja. Mikrotahkoteoriassa vain valon varjostuminen ja peittyminen otetaan huomioon.

Mikrotahkoteorian pohjalta on kehitetty useita mikrotahkomalleja, mutta yleisimmin käytetty lienee yhtälössä (3.12) esitetty Cook–Torrance-malli (Cook ja Torrance 1981).

$$\rho(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) = \frac{F(\boldsymbol{\omega}_i, \mathbf{h})G(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \mathbf{h})D(\mathbf{h})}{\pi \cos(\theta_o) \cos(\theta_i)} \quad (3.12)$$

Yhtälössä funktio  $F = F_\lambda$  on Fresnel-termi, joka kuvaa heijastuvan valon suhteellista määrää



Kuvio 6. (a) Mikrogeometria varjostaa pinnalle saapuvaa valoa. (b) Mikrogeometria peittää pinnalta lähtevää valoa.

tulokulman mukaan. Funktio  $G$  on geometrinen vaimennuskerroin, joka riippuu valon tulokulmasta ja katselukulmasta. Se kertoo kuinka paljon kyseisessäkin tilanteessa pinnan mikrogeometria varjostaa pinnalle saapuvaa valoa tai peittää pinnalta lähtevää valoa. Funktio  $D$  on normaalijakaumafunktio (engl. *normal distribution function, NDF*), joka kertoo suunnan  $\mathbf{h}$  mukaisten mikrotahkojen normaalien suhteellisen määrän. Normaalijakauma kuvaa siis pinnan karheutta, jota voidaan parametrisoida yhdellä karheusparametrilla  $\sigma$ , jos pinta on isotrooppinen, tai kahdella parametrilla  $\sigma_t$  ja  $\sigma_b$  (tangentin ja bitangentin suuntainen karheus), jos kyseessä on anisotrooppinen pinta. Karheusparametrit on esitetty yleensä mikrotahkojen kaltevuuden neliöllisenä keskiarvona (Cook ja Torrance 1981). Näin ollen niiden arvot ovat väliltä  $[0, 1]$ .

Cook–Torrance-mikrotahkomalli ei määrää tiettyjä funktioita  $G$  ja  $D$  käytettäväksi, vaikka Cook ja Torrance antoivatkin esimerkkifunktiot alkuperäisessä artikkelissaan. Uusia mikrotahko-heijastusjakaumafunktioita on siis mahdollista kehittää vaihtamalla näitä funktioita. Walter ym. (2007) antavat esimerkkejä eri normaalijakaumafunktioista ja geometrisistä vaimennuskertoimista. He listaavat Phongin empiirisen normaalijakaumafunktion, Beckmannin gaussilaisen normaalijakaumafunktion ja esittävät oman GGX-funktion. Walter ym. esittävät normaalijakaumafunktion uutena, mutta vastaavan funktion löysivät aiemmin Trowbridge ja Reitz (1975). Geometrisistä vaimennuskertoimista he mainitsevat Smithin geometrisen vaimennuskertoimen, joka ottaa huomioon pinnan karheuden, toisin kuin Cookin ja Torrancen esittämä vaimennuskerroin.

Lisäksi Schlick (1994) esitti useita edullisia approksimaatioita Cook–Torrance-mallille. Hän antoi myös Fresnel-termille yksinkertaisemman ja intuitiivisemman kaavan, joka onkin otettu käyttöön laajasti tietokonegraafiikassa (Walter ym. 2007). Schlickin Fresnel-termissä para-

metrina ei ole materiaalien taitekerroin, vaan Fresnel-reflektanssi valon tulokulman ollessa nolla. Tämä parametrisointi on intuitiivisempi ja voidaan käsittää materiaalin valon aallonpituudesta riippuvana peiliheijastus-albedona, jota yleisesti kutsutaan myös peiliheijastusväriksi (engl. *specular color*).

Cook–Torrance-mallin lisäksi on kehitetty muitakin mikrotahkomalleja. Esimerkiksi Oren ja Nayar (1994) erikoistivat mikrotahkoteorian diffuuseille pinnoille. Heidän mallissaan mikrotahkot eivät ole peiliheijastavia, vaan diffuuseja heijastuspintoja. Näin saatu malli huomio pinnan karheuden ja pystyy siten esittämään oikean maailman diffuuseja materiaaleja huomattavasti paremmin kuin pelkkä Lambertin malli. Smith ja Hancock (2009) esittävät mikrotahkomallin, jossa peiliheijastavaa ja diffuusia termiä ei ole erotettu toisistaan. Näin saatu malli on heidän mukaansa fysikaalisesti oikeampi. Tämä on totta, jos diffuusin ja peiliheijastavan termin erottaa toisistaan ja summaa yhteen ottamatta huomioon energian säilymistä. Energian säilyminen diffusoituneen ja peiliheijastuneen valon välillä seuraa Fresnel-termistä, joka kertoo, mikä osa valosta heijastuu ja mikä osa taittuu, kun se kohtaa materiaalin rajapinnan.

## 4 Heijastusmenetelmät

Heijastusmenetelmiä on kehitetty useita, joista osa soveltuu paremmin toteutettavaksi näytönohjaimella kuin toiset. Tekniikat voidaan kategorisoida usealla tavalla, mutta yksi tapa on ajatella niitä suoralla, jonka toisessa päässä on ympäristökartoitus ja toisessa päässä säteen seuranta (Popescu ym. 2006). Ympäristökartoituksessa käytetään rasterointia ja ympäristön oletetaan olevan äärettömän kaukana ilman geometriaesitystä. Säteen seurannassa taas ympäristöllä on tarkka geometria. Osa tekniikoista löytää näiden kahden välimaastosta kompromissin yksinkertaistamalla geometriaesitystä esimerkiksi yksinkertaisemman sijaisgeometrian tai syvyyksesityksen avulla.

Jotkin heijastustekniikat kykenevät renderöimään rekursiivisesti useita peräkkäisiä heijastuksia, heijastuksia, jotka syntyvät heijastavan kappaleen heijastaessa toisia heijastavia kappaleita. Ensimmäistä heijastusta kutsutaan ensimmäisen asteen heijastukseksi, toista heijastusta toisen asteen heijastukseksi ja niin edelleen. Toiset heijastusmenetelmät eivät ota huomioon kuin yhden heijastuksen. Tällöin myöskään heijastava kappale ei voi heijastaa itse itseään.

Taulukossa 1 on listattu menetelmät, joita luvussa käsitellään. Kyseiset menetelmät on valittu tarkastelun kohteeksi johtuen siitä, että ne on mahdollista toteuttaa näytönohjaimella.

### 4.1 Tasoheijastukset

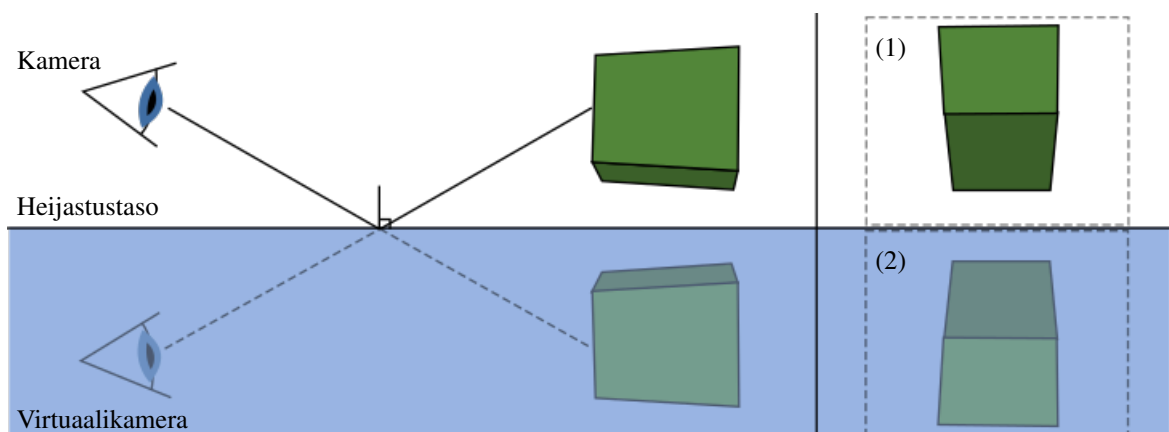
Tasoheijastukset on yksi yksinkertaisimmista heijastustekniikoista, jonka rajoitteena on se, että heijastavan geometrian tulee olla tasomainen. Tekniikassa virtuaalinen kamera peilataan heijastustason mukaan ja ympäristö piirretään peilatus kameran näkökulmasta. Lopullista kuvaa renderöitäessä heijastetun kameran heijastuskuvaa käytetään tasoheijastavalle geometrialle. Tätä on havainnollistettu kuvassa 7.

Tekniikalla pystytään saamaan tarkkoja peiliheijastuksia, vaikkakin vain tasomaiselle geometrialle. Tämä on kuitenkin erityisen oleellista, sillä tasomaisilla pinnoilla heijastusten virheet olisi helpompi havaita kuin monimuotoisemmilla pinnoilla.

Fuhrmann, Tobler ja Maierhofer (2004) ehdottavat perinteisten tarkkojen tasoheijastusten

Taulukko 1. Listaus luvussa käsitellyistä heijastusmenetelmistä. Tekniikkasarakkeessa on annettu tekniikka, johon menetelmä perustuu. EM tarkoittaa ympäristökartoitusta, GEO virtuaaligeometriaa ja RT säteenseurantaa.

Menetelmä	Lähde	Tekniikka
Tasoheijastukset	Fuhrmann, Tobler ja Maierhofer 2004	-
Parametrisoidut ympäristökartat	Hakura, Snyder ja Lengyel 2001	EM
Referenssiympäristökartat	Meyer ja Loscos 2003	EM
Syvyysympäristökartta	Patow 1995	EM
Palloharmoninen heijastuskartta	Ramamoorthi ja Hanrahan 2001, 2002	EM
Objektin syvyyskartta	Zhang, Hsieh ja Shen 2008	-
Valokenttä	Yu, Yang ja McMillan 2005	-
Suorakulmainen impostori	Popescu ym. 2006	-
Virtuaalinen geometria	Estalella ym. 2006; Roger ja Holzschuch 2006	GEO
Kuvapuskuriheijastukset	Tiago Sousa ja Schulz 2012; McGuire ja Mara 2014	RT
Sumeat kuvapuskuriheijastukset	Valient 2014; Hermanns 2015	RT



Kuvio 7. Kuvassa on esitetty tasoheijastusten toimintaperiaate. Kohdassa (1) näkyy virtuaalikameran näkymä ja kohdassa (2) näkymä on projisoitu heijastustavan geometrian pinnalle.

parannusta, jossa pinnan karheus ja Fresnel-reflektanssi otetaan huomioon. Tekniikassa heijastusta sumennetaan suotimella, jonka ala riippuu pinnan karheudesta ja katseluetäisyydestä.

## 4.2 Ympäristökarttoihin perustuvat menetelmät

Ympäristökartoitus (engl. *environment mapping*) on yksi yleisimmin käytetyistä heijastustekniikoista. Tekniikan esittelivät Blinn ja Newell vuonna 1976 nimellä heijastuskartoitus, jossa ympäristöstä tuleva valo saatiin ympäristöä esittävistä kuvasta napakoordinaatistoa käyttäen. Tekniikassa kappaleen ympäristö renderöidään tekstuuriin niin kutsutuksi ympäristökartaksi käyttäen sopivaa projektiota. Myöhemmin kappaletta renderöitäessä lasketaan valaistavan pisteen normaalin ja katselusuunnan mukainen heijastussuunta, jonka mukaan aiemmin renderöidystä ympäristökartasta otetaan näyte, jota käytetään kyseiseen pisteeseen heijastuneena valona. Menetelmässä oletetaan, että ympäristökartta, joka yleensä renderöidään kappaleen keskipisteestä katsottuna, on hyvä arvio heijastuksesta kappaleen pinnan kaikissa pisteissä. Tämä pätee, jos heijastuva ympäristö on äärettömän kaukana kappaleesta.

Erilaisten parametrisointien ansiosta ympäristökartta voi olla esitetty usealla eri tavalla. Koska ympäristön eri suunnat muodostavat yksikköpallon, täytyy hyvän parametrisoinnin pystyä kuvaamaan yksikköpallon kaikki suunnat kaksiulotteisena tekstuurina. Yksi esimerkki parametrisoinnista on pallomainen ympäristökartta, joka vaatii vain yhden tekstuurin, mutta johon projektion tuomasta vääristyksestä sen laatu on heikko. Kaksoisparaboloidi-projektiota käyttäen saadaan ympäristönäkymä jaettua kahteen puolipalloon, ja näin laatu on huomattavasti parempi kuin pallomaisen ympäristökartan projektion (Heidrich ja Seidel 1998). Pallokoordinaatistossa esitetty panoraamamainen ympäristökartta on myös hyvin yleinen ja kykenee esittämään koko ympäristön hyvällä tarkkuudella. Eniten käytetty parametrisointi on kuitenkin kuutiokartoitus (Greene 1986). Ympäristöstä renderöidään kuusi tekstuuria 90:n asteen perspektiivisellä projektiolla. Jokainen tekstuuri kuvaa ympäristökarttana käytettävän kuutiotekstuurin tahkoa. Kuutiokartoitus sopii erityisen hyvin ympäristökartaksi, sillä kykenee esittämään ympäristön hyvällä tarkkuudella ilman suurta vääristystä.

Ympäristökartoitus soveltuu hyvin reaaliaikaiseen grafiikkaan ja sitä onkin käytetty laajasti,

varsinkin, kun näytönohjaintuki kuutiotekstuureille yleistyi. Perinteisesti ympäristökartoitusta on käytetty kappalekohtaisesti, jolloin jokaiselle kappaleelle täytyy generoida oma ympäristökartta. Tämä voi olla kuitenkin raskasta, jos kappaleita on paljon. Jos kappaleet liikkuvat täytyy ympäristökartta generoida uudelleen jokaisen kappaleen liikkeessa. Jos kappaleet ovat kuitenkin pieniä, eikä tarvita täysin tarkkoja heijastuksia, voidaan ympäristökarttoja päivittää harvemmin. Toisaalta, jos jokainen kappale säilyttää omaa ympäristökarttaansa, on tällöin muistintarve lineaarinen kappaleiden määrään nähden.

Tekniikkaan on tehty monia parannuksia ja muutoksia. Sen sijaan, että jokaisella kappaleella olisi oma ympäristökarttansa, voidaan muistintarvetta vähentää käyttämällä staattisesti ympäristöön sijoiteltuja ”ympäristötutkaimia” (engl. *environment probe*). Tällöin, jos ympäristön koko ja tutkaimien tiheys suhteessa kappaleiden määrään on pienempi, säästyy muistia. Ympäristön heijastukset saadaan laskettua kappaleelle esimerkiksi interpoloimalla muutama lähimmän tutkaimen ympäristökarttaa. Näin saatu heijastus ei kuitenkaan ole erityisen tarkka, jos heijastuva ympäristö on oikeasti kappaleen läheisyydessä eikä äärettömän kaukana. Tämä johtuu siitä, että ympäristökartan esittämä heijastus on tarkka ainoastaan pisteessä, josta se on piirretty, siis ympäristöprojektion keskipisteessä. Heijastusten tarkentamiseksi on kehitetty tekniikoita, joilla heijastusta voidaan korjata.

Patow (1995) käyttää ympäristökartoille syvyyspuskuria, jonka avulla löydetään heijastussäteen ja ympäristögeometrian leikkauspiste. Tätä leikkauspistettä käyttäen voidaan heijastussuuntaa korjata, jotta myös jonkin matkan päässä, kuitenkin riittävän lähellä, alkuperäisestä projektion keskipisteestä olevalle pisteelle saadaan tarkka<sup>1</sup> heijastus. Szirmay-Kalos ym. (2005) kuvailevat vastaavaan tekniikan, joka tukee useita rekursiivisia heijastuksia ja refraktiivisia materiaaleja. Tekniikka käyttää useita kuutiotekstuureita kuvaamaan ympäristön materiaaliominaisuuksia ja etäisyyttä projektion keskipisteestä. Tekniikka mahdollistaa usean syvyyskerroksen käyttämisen syvyyskuorintatekniikan (Nagy ja Klein 2003; Liu ym. 2009) tavoin, jolloin heijastettava ympäristö voidaan esittää tarkemmin. Näin kovera heijastava kappale kykenee heijastamaan osia itsestään.

Hakura, Snyder ja Lengyel (2001) käyttävät heijastusten lokalisoimiseen useita ympäristökarttoja, joita he kutsuvat parametrisoiduiksi ympäristökartoiksi, ja saavuttavat lähes sä-

---

1. Tarkkuus riippuu ympäristökartan resoluutiosta ja syvyysarvon tallennustarkkuudesta.

teenseurantaa vastaavia tuloksia. He jakavat heijastuvan geometrian kaukaiseen ja läheiseen, jolloin eri ympäristökarttoja voidaan käyttää eri etäisyyden ympäristöä varten. Läheistä ympäristöä varten käytetään yksinkertaista sijaisgeometriaa (laatikko, pallo tai ellipsoidi), jonka avulla heijastussuunta voidaan korjata samaan tapaan kuin syvyyspuskuria käytettäessä. Björke (2004) esittää myös heijastusten lokalisointia sijaisgeometrian avulla. Saman idean esittelevät Sébastien ja Zanuttini (2012), mutta lisäävät sijaisgeometriaksi myös kuperan monitahokkaan, jolla ympäröivää geometriaa voidaan mallintaa tarkemmin.

Meyer ja Loscos (2003) käyttävät useita referenssiympäristökarttoja, jotka on sijoitettu ympäristöön harvasti. Nämä ympäristökartat sisältävät myös syvyyspuskurin. Kappaleen liikkuessa tälle luodaan oma ympäristökartta yhdistämällä referenssiympäristökarttoja. Sen sijaan, että lähimmät ympäristökartat yhdistettäisiin yksinkertaisella lineaarisella interpolatiolla, lasketaan heijastussuunnat jokaiselle referenssiympäristökartalle erikseen käyttäen syvyystietoa hyväksi, jolloin ympäristökarttojen yhdistäminen tuottaa tarkemman tuloksen. Näin liikkuvan kappaleen ympäristökarttaa päivitettäessä ei tarvitse piirtää koko ympäristöä uudelleen, vaan riittää että heijastusympäristö renderöidään yhdistäen lähistön esirenderöidyt referenssiympäristökartat. Tässä ongelmana on se, että jokaista kappaletta varten tarvitaan oma ympäristökartta. Nykyisellä grafiikkalaitteistolla on kuitenkin mahdollista ohittaa kappalekohtaisen ympäristökartan eksplisiittinen tallentaminen ja käyttää referenssiympäristökarttoja sellaisenaan valaisuun. Tällöin on hyvä asettaa yläraja yhtä kappaletta varten käytettävien ympäristökarttojen määrälle.

Ramamoorthi ja Hanrahan (2001; 2002) käyttävät ympäristökarttoja BRDF-funktion reaaliaikaiseksi laskemiseksi. He muuntavat ympäristökartan spatiaalisen esityksen palloharmoiniseksi (engl. *spherical harmonic*) taajuusesitykseksi. Heidän tekniikkansa on hyvä esittämään BRDF-funktion vaihtelua, mutta vaatii paljon tekstuureita. Esimerkiksi tekstuurien määrä on 18 yhtä ympäristökarttaa kohden, kun palloharmoinisen sarjan termeistä käytetään vain kahta. Mitä enemmän termejä käytetään, sitä tarkempi kuvaus BRDF-funktiosta saadaan, mutta tekstuurien tarve kasvaa nopeasti<sup>2</sup>. Kuitenkin, Ramamoorthi ja Hanrahan (2002) esittävät, että taajuusesitys vaatii keskimäärin noin kymmenesosan tekstuuriresoluution spatiaaliseen esitykseen verrattuna saavuttaen vastaavan laadun. Jotta tarkkuus olisi tarpeeksi

---

2. Muistintarve on  $2 \cdot (P_B + 1)^2$  tekstuuria, kun  $P_B$  on palloharmoinisen sarjan termien määrä.



hyvä esittämään tarkkoja peiliheijastuksia, tarvittaisiin useita tekstuureja, jolloin tekniikan tehokkuus heikkenee. Tämä johtuu siitä, että mitä enemmän termejä on, sitä vaativampi valaistus on suorittaa laskennallisesti. Tekniikka ei siis sovi erityisen peiliheijastusten esittämiseen.

### 4.3 Muita kuvapohjaisia menetelmiä

Yu, Yang ja McMillan (2005) käyttävät 4-ulotteista valokenttää (engl. *light field*) tavallisen ympäristökartoituksen sijaan. Valokenttä koostuu kuudesta lohkoista, jotka sisältävät ympäristön renderöitynä eri pisteissä kuudesta eri suunnasta. Tällä ympäristön esitysmuodolla saadaan heijastuksiin liikeparallaksia, joka puuttuu yksinkertaisesta ympäristökartoituksesta. Tekniikan muistintarve on huomattavasti suurempi kuin tavallisen ympäristökartoituksen, mutta se kykenee esittämään ympäristön tarkemmin. Jos renderöitävä ympäristö on riittävän pieni ja halutaan tarkemmat heijastukset kuin ympäristökartoitus mahdollistaa, voi 4-ulotteinen valokenttä olla hyvä ratkaisu. Nykylaitteistolla sen suorituskyky ei ole ongelma, mutta muistivaativuus ei skaalaudu erityisen hyvin suurempiin ympäristöihin.

Zhang, Hsieh ja Shen (2008) ehdottavat heijastustekniikkaa, jossa heijastuvan ympäristön kappaleita varten renderöidään syvyyskuutiotekstuuri, joka esittää kappaleen muotoa. Tätä kappaleen esitysmuotoa hyväksikäyttäen saadaan heijastussäteen tarkka leikkauspiste löydettyä. Tällöin algoritmin vaativuus ei ole riippuvainen heijastettavien kappaleiden geometriasta. Tosin, jos kappale on kovera, tarvitaan useista kerroksista koostuva syvyystekstuuri, ja algoritmi joutuu etsimään näistä kerroksista oikean leikkauspisteen. Tekniikan ongelmana on se, että sen muistinvaativuus on lineaarinen heijastettavien kappaleiden, eli ympäristössä olevien kappaleiden määrään nähden. Näin ollen tekniikka ei skaalaudu hyvin kappaleiden määrään mukaan. Lisäksi, jos kappale on animoitu ja muuttaa muotoaan, joudutaan kappaleen syvyyskuutiotekstuuri renderöimään uudelleen.

Popescu ym. (2006) käyttävät oikean heijastavan geometrian sijaan yksinkertaisia teksturoituja suorakulmioita, impostoreita (engl. *billboard impostor*) sijaisgeometriana. Lisäksi he käyttävät vaihtoehtoisesti syvyyspuskuriesitystä saavuttaakseen tarkemmat heijastukset. Koska kaikkea geometriaa ei ole järkevä esittää suorakulmaisella sijaisgeometrialla, käyt-

tää tekniikka tavallista ympäristökarttaa, jos säde ei törmää sijaisgeometriaan. Tekniikka kykenee toisen tai useamman asteen heijastuksiin, luonnollisesti vähentäen suorituskykyä ensimmäisen asteen heijastuksiin verrattuna. Tekniikan ongelmana on, että sen suorituskyky riippuu sijaisgeometriaan käytettävien suorakulmioiden määrästä. Lisäksi sijaisgeometriaa käyttävien kappaleiden valitseminen jää joko artistin harteille, tai sitä varten täytyy kehittää automaattinen algoritmi.

#### **4.4 Virtuaaligeometriaan perustuvat menetelmät**

Estalella ym. (2006) esittävät virtuaaligeometriaan perustuvan heijastusmenetelmän, jossa jokaista heijastavaa kappaletta kohden muodostetaan virtuaalinen ympäristögeometria heijastuvista kappaleista. Tekniikassa renderöidään ensin heijastavien kappaleiden sijainti ja normaali kahteen tekstuuriin. Tämän jälkeen heijastuva geometrian verteksit renderöidään pisteprimitiiveinä käyttäen frakmenttivarjostinohjelmaa, joka laskee heijastuneen geometrian verteksit käyttäen sijainti- ja normaalitekstuureja. Varjostinohjelma tulostaa heijastuneet verteksit kuvapuskuriin, joka kopioidaan verteksipuskuriin, joka on nyt koko heijastunut ympäristö. Tätä verteksipuskuria käyttäen piirretään heijastunut ympäristö heijastavalle kappaleelle. Nykyaikainen toteutus voisi laskea heijastuneen geometriantehokkaammin käyttäen verteksivarjostinta ja muunnospalaute-ominaisuutta (engl. *transform feedback*).

Roger ja Holzschuch (2006) ehdottavat samankaltaista geometriaan perustuvaa menetelmää kuin Estalella ym. (2006). Tekniikat eroavat heijastuneen geometrian verteksien sijaintien etsintään käytettävän algoritmin osalta. Tulokset molemmilla tekniikoilla ovat samankaltaisia ja kärsivät samoista ongelmista. Tekniikat pystyvät renderöimään tarkkoja peiliheijastuksia, mutta eivät tue heijastavan pinnan karheutta ja siitä johtuvaa heijastuksen sumentumista. Ympäristön geometrian tulee olla tarpeeksi hienojakoista, jotta heijastuvaan geometriaan ei tule epäjatkuvuutta tai suoria, vaikka heijastuksen tulisi kaareutua. Lisäksi toisen tai korkeamman asteen heijastusten renderöinti ei ole suoraan mahdollista ja aiheuttaisi suuria skaalautuvuus ongelmia. Jo yhden heijastavan kappaleen tapauksessa koko ympäristö tulee potentiaalisesti renderöidä kahteen kertaan.

## 4.5 Säteenseuranta syvyyspuskuriin

Säteenseuranta on luonnollisin tapa tarkkojen heijastusten toteuttamiseen, mutta ei tehokkuussyistä ole sopiva reaaliaikaiseen renderöintiin. Säteenseurannan tehokkuus on hyvin riippuvainen ympäristön geometrisesta kompleksisuudesta, joten se ei skaalaudu suuriin ympäristöihin. On kuitenkin mahdollista yksinkertaistaa ympäristön esitysmuoto rasteroinnin avulla pelkäksi korkeuskartaksi, johon säteenseuranta, tai tarkemmin säteen marssinta (engl. *ray-marching*), on paljon tehokkaampaa. Säteenseuranta tarvitsee suorittaa vain kerran jokaista kuvaelementtiä kohden, joten se ei riipu enää ympäristön geometrisesta kompleksisuudesta. Tämä ajatus on vastaava kuin se, jonka esimerkiksi Patow (1995) esitti ympäristökartoituksen lokalisointiin. Koska syvyyssesityksen käyttäminen ympäristökarttojen kanssa vaatii enemmän muistia ja suorituskykyä jokaista ympäristökarttaa kohden, ei tekniikka skaalaudu erityisen hyvin suureen määrään ympäristökarttoja. Sen sijaan, jos samaa tekniikkaa käytetään vain renderöitävää päänäkymää varten, saadaan tekniikka toimimaan paremmin reaaliajassa.

Tekniikan esitteli ensin Tiago Sousa ja Schulz (2012) nimellä reaaliaikaiset lokaalit heijastukset (engl. *real-time local reflections*), mutta tämän jälkeen tekniikan nimeksi on vakiintunut vapaasti käännettynä kuvapuskuriheijastukset (engl. *screen-space reflections*). Tekniikka toimii siten, että jokaista kuvaelementtiä kohden lasketaan heijastussäteen alkupiste ja suunta käyttäen esirenderöityä syvyys- ja normaalipuskuria. Seuraavaksi otetaan askelia sädettä pitkin ja verrataan säteen syvyyttä syvyyspuskurin syvyysarvoon. Jos syvyysero on tarpeeksi pieni, säde törmää. Tämä törmäyspiste voidaan projisoida edellisen kuvaruudun näkymään ja ottamalla näyte edellisen kuvaruudun väripuskurista saadaan heijastunut väri. Tekniikan suurimpana rajoitteena on kuvapuskurissa olevan tiedon määrä; syvyyspuskuri ei ole täydellinen esitys ympäristöstä, eivätkä kaikki heijastussäteet päädy kuvapuskurissa näkyvälle alueelle.

Viivästetty varjostus tai valaistus (engl. *deferred shading*) on renderöintitekniikka, jossa geometrian piirto ja valaistus on erotettu toisistaan. Geometriavaiheessa, niin kutsutussa g-puskurivaiheessa, geometrian kuvaavat attribuutit tallennetaan tekstuuriin, jotka myöhemmin valaistusvaiheessa luetaan valaistuksen laskemista varten. Kuvapuskuriheijastusten yksinkertaisen toteutuksen vaiheet viivästettyä varjostusta käytettäessä ovat

1. **G-puskurivaihe.** Piirretään g-puskuri, joka koostuu valaisuun tarvittavista attribuuteista, joita ovat diffuusikerroin, peiliheijastuskerroin, karheus, pinnan normaali ja syvyys.
2. **Suora valaistus.** Käydään läpi kaikki analyttiset valonlähteet ja piirretään additiivisesti valoja vastaava geometria. Varjostin laskee jokaiselle kuvaelementille kyseisen valon aiheuttaman suoran valaistuksen. Lopputuloksena saadaan *valaistuspuskuri*.
3. **Säteenseuranta.** Lasketaan jokaiselle kuvaelementille heijastusvektori, jonka suuntaan askeltamalla etsitään törmäyspiste syvyyspuskurin kanssa. Otetaan heijastusnäyte valaistuspuskurista törmäyspistettä vastaavasta kohdasta. Lopputuloksena saadaan *heijastuspuskuri*.
4. **Heijastusten yhdistäminen.** Piirretään additiivisella sekoituksella heijastuspuskuri valaistuspuskuriin.

Kuvapuskuriheijastusten naiivi toteutus marssii sädettä epäoptimaalisesti. Jos sädettä pitkin edetään lineaarisesti kolmiulotteisessa avaruudessa, päädytään ottamaan ylimääräisiä näytteitä syvyyspuskurista lähempänä säteen alkupistettä ja liian vähän näytteitä kauempana. McGuire ja Mara (2014) esittävät parannuksen tekniikkaan, jossa näytteitä otetaan juuri oikea määrä. He kulkevat sädettä pitkin lineaarisesti kuvapuskurin koordinaatistossa, jolloin jokainen kuvaelementti käsitellään enintään kerran.

Yksinkertaisella kuvapuskuriheijastustekniikalla saadaan hyvä arvio heijastuvasta radianssista, mutta se ei vielä ota huomioon heijastavan pinnan heijastusjakaumafunktiota. Fysikaalisesti tarkka tapa tähän olisi ottaa useita heijastusnäytteitä heijastusjakaumafunktion normaalijakaumafunktion mukaisesti. Tällöin siis tarvitsee seurata useampia säteitä kuvaelementtiä kohden. Luonnollisesti siis Monte Carlo -näytteistys käyttäen suodatettua painotettua otantaa kuulostaa järkevältä. Tehokkuuden kannalta useampien näytteiden ottaminen voi kuitenkin olla liikaa reaaliaikaisen suorituksen saavuttamiseksi.

Hermanns (2015) kuvaa tekniikan, joka seuraa säteiden sijaan kartioita. Tekniikan nimi on kuvapuskurikartionseuranta (engl. *screen space cone tracing*). Tekniikassa kartio approksimoi useita heijastussäteitä. Kun kartion törmäyspistettä etsitään, käytetään syvyyspuskurista mipmap-ketjuun esisuodatettua versiota, joka vastaa kartion alaa. Mitä kauempana kartion alkupisteestä ollaan, sitä suuremman alan kartio peittää.

Valient (2014) (pelissä *Killzone: Shadow Fall*) käyttää yksinkertaisempaa lähestymistapaa. Heijastuspuskurista esisuodatetaan mipmap-ketjuun sumennetut versiot, jotka vastaavat eri karheutta. Kun heijastuspuskuri yhdistetään lopulliseen kuvaan, käytetään kyseisen pisteen karheutta vastaavaa sumennettua heijastuspuskuria lähteenä.

Säteenseurantaa voidaan nopeuttaa ottamalla pidempiä harppauksia etsittäessä törmäyspistettä. Luonnollisesti tällöin voidaan myös vahingossa ohittaa syvyyspuskurissa olevia pieniä yksityiskohtia, jolloin heijastuksista puuttuu osia. Toisaalta on mahdollista, että törmäys havaitaan vasta kun ollaan jo kappaleen sisällä, jolloin törmäyspisteen sijainti ei ole tarkka. Tilannetta voidaan parantaa tarkennusvaiheella, joka toteutetaan esimerkiksi binäärihakuna (McGuire ja Mara 2014).

Widmer ym. (2015) esittelevät mukautuvan kiihdytysrakenteen, jolla kuvapuskurisäteenseurantaa voidaan nopeuttaa. Kiihdytysrakenne mahdollistaa tyhjän tilan ohittamisen säteenseurannan aikana, jolloin törmäyspiste löydetään nopeammin. Kiihdytysrakenne perustuu neli-puuhun ja muodostetaan syvyyspuskurista.

## 5 Metodi

Seuraavassa luvussa keskitytään tarkemmin kahteen heijastustekniikkaan: ympäristötutkaimiin ja kuvapuskuriheijastuksiin. Luvussa arvioidaan näiden tekniikoiden yleisiä piirteitä, ongelmakohtia ja toteutusta fysikaalisen valaistuksen näkökulmasta. Arviointia varten molemmista tekniikoista on toteutettu fysikaalista valaistusta tukevat versiot.

### 5.1 Tekniikoiden valinta

Syy näiden tekniikoiden valintaan on se, että niiden toteutus ei vaadi erityisiä ominaisuuksia grafiikkalaitteistolla. Ympäristökartoitukseen perustuvat tekniikat skaalautuvat hyvin ja niille on luonnollinen toteutus rasterointiin pohjautuvalla grafiikkalaitteistolla. Myös kuvapuskuriheijastusten tehokas toteutus on mahdollista ja tekniikan suorituskykyä saadaan skaalattua esimerkiksi muuttamalla heijastuspuskurin kokoa. Molemmat tekniikat voidaan toteuttaa siten, että uskottava ja analyttisen valaistuksen kanssa yhdenmukainen fysikaalinen valaistus on mahdollista. Tekniikat eivät myöskään vaadi esilaskentaa, joten ne sopivat dynaamisiin ympäristöihin.

Fysikaalisesti uskottava heijastustekniikka mukailee fysikaalisesti todenmukaista mallia, siten että heijastuksissa ilmenevät tietyt ominaisuudet. Näitä ovat karheuteen ja etäisyyteen perustuva heijastusten sumentuminen sekä heijastusten intensiteetin riippuvuus heijastusjakaumafunktiosta. Luonnollisesti uskottavuuteen vaikuttaa myös heijastusten ja heijastuvan geometrian vastaavuus. Seuraavassa luvussa arvioidaan valittujen tekniikoiden laatua näistä näkökulmista. Lisäksi arvioidaan tekniikoiden toteutuksen suorituskykyä.

### 5.2 Heijastusjakaumafunktion valinta

Seuraavassa luvussa esitellyt tekniikat on toteutettu käyttäen Cook–Torrance-mallin mukaisista heijastusjakaumafunktiota peiliheijastavan osuuden laskemiseen ja Lambertin diffuusia mallia diffuusisti heijastavan osuuden laskemiseen. Energian säilyminen otetaan huomioon tekemällä diffuusi albedo ja peiliheijastus-albedo toisistaan riippuvaiseksi. Jos materiaali

on metallia, diffuusi albedo on nolla ja peiliheijastus-albedo on materiaalin väritekstuurin mukainen. Jos materiaali on taas dielektrinen, diffuusi albedo saadaan väritekstuurista ja peiliheijastus-albedo on vakio 0.04.

Mallissa Fresnel-terminä on käytetty Schlickin arviota, geometrisenä vaimennuskertoimena käytetään Schlickin arviota Smithin antamasta vaimennuskertoimesta (Schlick 1994) ja normaalijakaumafunktiona GGX-funktiota (Walter ym. 2007). Schlickin Fresnel-termin muotoa on hyödynnetty ympäristön heijastusjakaumafunktion yksinkertaistamisessa. Lisäksi heijastusjakaumafunktion näytteistys Monte Carlo -menetelmiä varten on oltava mahdollista. Näitä rajoitteita lukuun ottamatta heijastusjakaumafunktion termien valinta on vapaata.

## 6 Reaaliaikaiset heijastukset

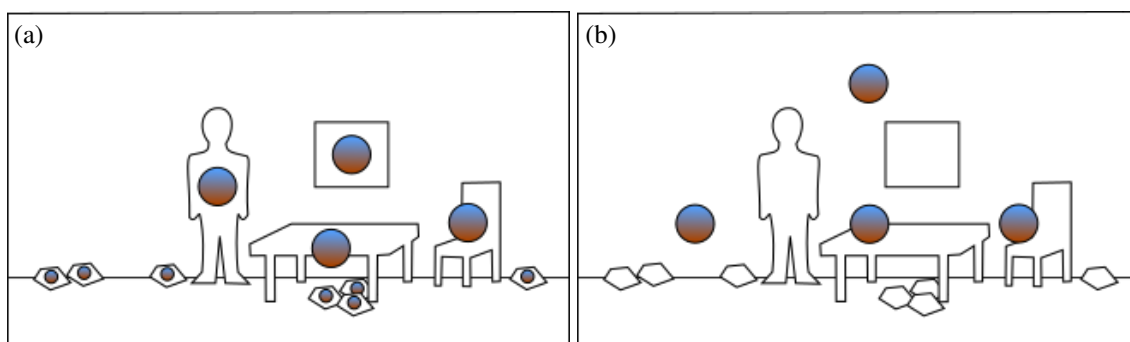
Reaaliaikaisilla heijastustekniikoilla on vaatimuksena suorituskyky. Ongelma, jota tekniikat pyrkivät ratkaisemaan, on hyvin haastava nykyaikaisella rasterointiin perustuvalla grafiikkalaitteistolla. Heijastusongelma, kuten mikä tahansa epäsuoran valon renderöintiongelma, on luonnostaan rekursiivinen. Jotta tiedetään pisteestä lähtevä radianssi, joudutaan integroimaan radianssi kaikista pisteeseen saapuvista suunnista. Jotta näistä suunnista tuleva radianssi saadaan laskettua, täytyy jokaisessa suunnassa olevalle heijastavalle tai emittoivalle pisteelle ensin suorittaa sama integrointi. Tällaisen rekursion esittäminen grafiikkalaitteistolla ei ole luonnollista ja tehokasta. Jos ongelma pystytään muuttamaan rasterointiongelmaksi, voidaan kuitenkin saada hyötyä grafiikkalaitteistosta. Tämä onkin reaaliaikaisten heijastustekniikoiden yleisin lähestymistapa.

### 6.1 Ympäristötutkaimet

Ympäristökartoituksella on mahdollista approksimoida pisteeseen saapuvaa epäsuoraa valaistusta hyvinkin tarkasti, mutta tämä pätee vain projektion keskipisteessä. Perinteisesti jokaista kappaletta varten on piirretty oma ympäristökartta, jonka oletetaan olevan tarpeeksi hyvä arvio myös kappaleen pinnan eri pisteissä. Jos kuitenkin esitettävä ympäristö sisältää niin suuren määrän kappaleita, että jokaiselle kappaleelle ei pystytä säilömään muistiin omaa ympäristökarttaa, voidaan ympäristökartat erottaa kappaleista. Tällöin ympäristökartat on järkevämpi kaapata ennalta määräytyistä ympäristön pisteistä ja käyttää näitä ympäristötutkaimia laskettaessa kappaleeseen saapuva epäsuora valaistus. Kuvio 8 havainnollistaa tätä. Kuvioista huomataan, että kun ympäristön sisältäessä paljon pieniä kappaleita, käyttämällä ympäristötutkaimia selvittää paljon vähemmällä määrällä tallennettuja ympäristökarttoja. Lisäksi, jos ympäristö muuttuu, voidaan ympäristökartat haluta piirtää uudelleen, jolloin pienempi määrä ympäristökarttoja on tehokkaampi päivittä.

Ongelmaksi jää ympäristötutkaimien asettelu ja niiden vaikutuksen laskeminen valaistavalle pisteelle. Lisäksi, koska kaapattujen ympäristökarttojen projektion keskipiste ei välttämättä ole yhdenkään kappaleen lähetyvillä, ei sen kaappaama radianssi ole sellaisenaan hyvä





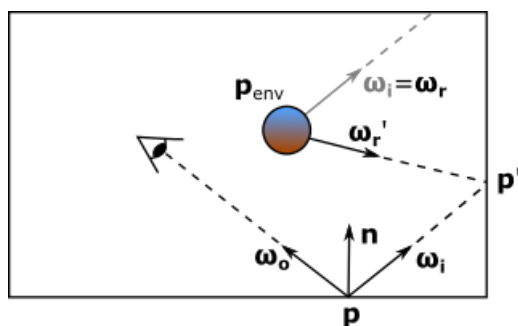
Kuvio 8. Kuvassa (a) jokaisella kappaleella on oma ympäristökartta. Kuvassa (b) ympäristökartat on kaapattu niin kutsutuista ympäristötutkaimista. Kuviossa liukuväritetyt ympyrät kuvaavat ympäristökarttojen sijaintia.

arvio kappaleiden valaisuun. Ympäristökartan vaikutus valaistavassa pisteessä saadaan tarkemmaksi korjaamalla heijastusvektoria suuntaan, josta saapuva radianssi tulisi näytteistää kyseisestä ympäristökartasta. Tätä kutsutaan ympäristökartan lokalisoinniksi.

### 6.1.1 Ympäristötutkaimien vaikutus ja asettelu

Tutkielmassa ei ole erityisesti keskitytty ympäristötutkaimien vaikutuksen laskemiseen tai niiden asetteluun. Kuitenkin, jotta tulokset olisivat hyviä, tulisi näihinkin kiinnittää huomiota. Tutkielmassa esitetyissä kuvissa on käytetty yksinkertaisia ympäristöjä, jolloin tutkaimien asettelu käsin ei ole liian työlästä. Jos ympäristö on kuitenkin laaja, on parempi toteuttaa tutkaimien asettelu algoritmisesti. Tutkaimet on järkevintä sijoittaa paikkoihin, joissa niillä saadaan paras vaikutus. Näitä ovat tyhjät tilat, jotka ovat sopivalla etäisyydellä ympäristön geometriasta.

Tutkaimien vaikutuksen interpolointi kuvaelementtiin heijastuvaa valoa laskettaessa on myös hankalaa. On tärkeää, että kahden tutkaimen valaistusta painotetaan siten, ettei niiden päällekkäinen vaikutus kaksinkertaista heijastunutta valaistusta. Kun tutkaimet on aseteltu tasaisin välein, voidaan niistä muodostaa tetrahedraalisia ryhmiä. Jokainen tetrahedronin sisällä oleva kuvapiste saa valaisunsa interpoloituna ympäristötutkaimista, jotka ovat tetrahedronin kärkipisteissä. Lisätietoa tutkaimien asettelusta ja niiden vaikutuksen laskemisesta saa esimerkiksi Valientin diaesityksestä (Valient 2014).



Kuvio 9. Kuvio esittää heijastusvektorin korjauksen geometrian. Korjattu heijastusvektori  $\omega_r'$  saadaan heijastuspisteen  $p'$  ja ympäristökartan projektion keskipisteen  $p_{env}$  erotuksena.

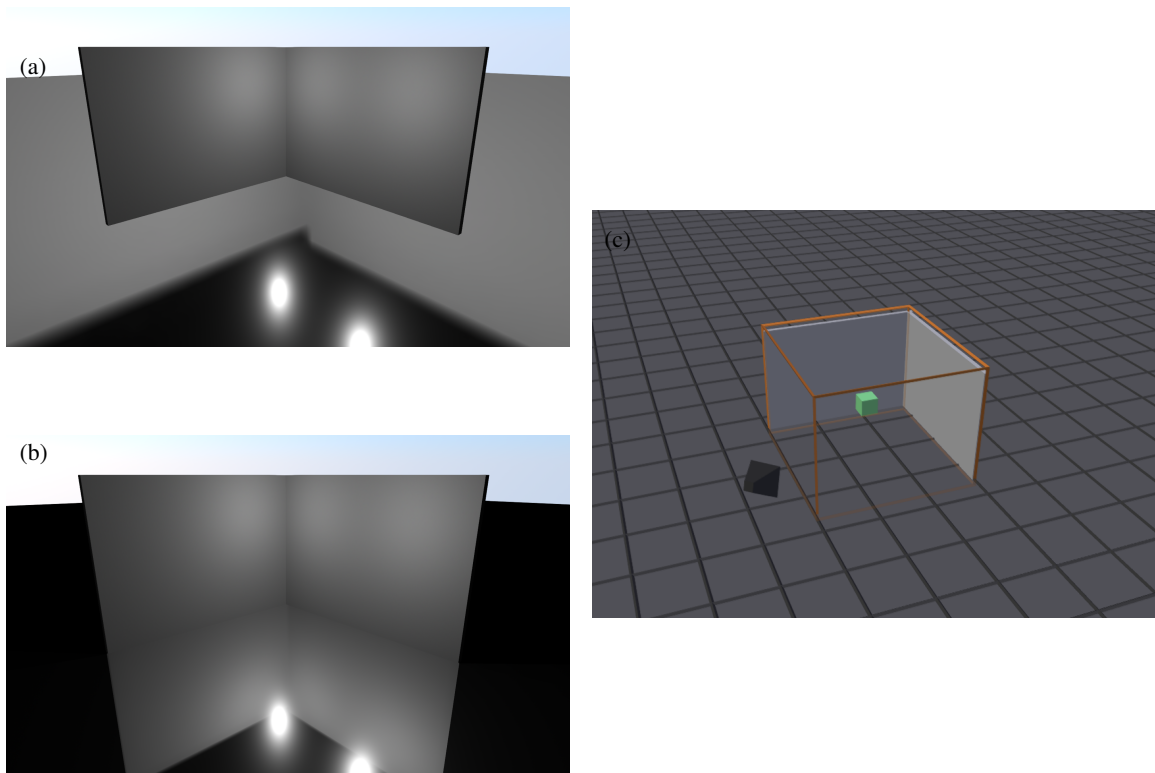
Tutkimuksessa esitetyissä ympäristötutkaimia käyttävissä kuvissa ei ole toteutettu tetrahedraalista interpolointia. Lisäksi ympäristötutkaimet on sijoitettu käsin siten, että ongelmakohdat on minimoitu.

### 6.1.2 Ympäristökartan lokalisointi

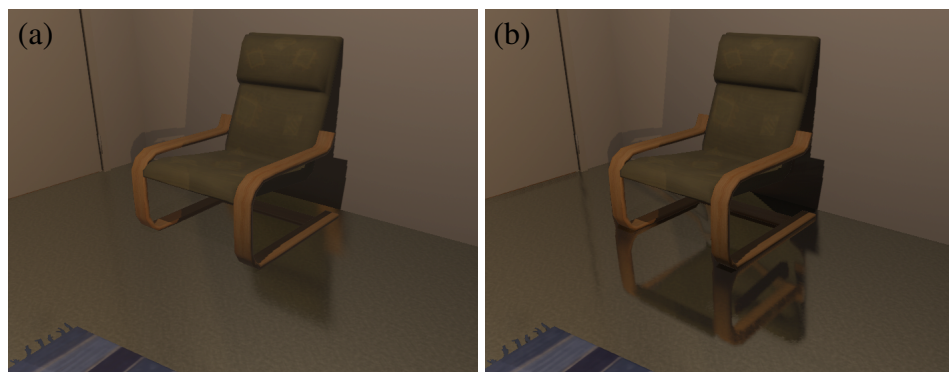
Koska ympäristötutkaimen ympäristökarttaan tallennettu radianssi on tarkka vain projektion keskipisteessä, muissa pisteissä arviota voidaan tarkentaa muuttamalla suuntaa, josta näyte otetaan. Jotta ympäristöstä saapuva epäsuorasti heijastuvan valon heijastussuunta saadaan korjattua, tarvitaan arvio ympäristökarttaan piirretystä geometriasta. Tämä voi olla joko syvyyskartta, jolla saadaan tarkempia tuloksia, tai ympäristöä mukaileva sijaisgeometria, kuten laatikko tai pallo. Kuviossa 9 on esitetty heijastusvektorin korjaus tapauksessa, jossa sijaisgeometria on laatikko. Kuten kuviossa näkyy, projektion keskipisteen ei tarvitse sijaita sijaisgeometrian keskipisteessä.

Kuviossa 10 on esitetty lokalisoinnin vaikutus heijastuksiin. Lokalisoitu heijastus vastaa selvästi tarkemmin heijastuvaa geometriaa. Kuvan lokalisointi on tehty laatikkosijaisgeometrialla, jolloin kaikki pisteet arvioidaan heijastuksen leikkauspistettä laskiessa olevan laatikon reunalla. Tästä johtuva ongelma heijastusgeometrian litistymisen voidaan nähdä kuviossa 11. Koska huoneen seiniä mukaileva laatikko ei selvästikään ole tarkka esitys kuvassa näkyvästä geometriasta, eivät todellinen geometria ja heijastus myöskään täsmää.

Tätä ongelmaa saadaan pienennettyä, kun heijastavat pinnat ovat tarpeeksi karheita, jolloin



Kuvio 10. Kuviossa on näkymä, jossa lähes täysin peiliheijastavassa tasossa näkyy heijastus kahdesta seinästä. Kuvassa (a) heijastukset on toteutettu perinteisellä ympäristökartoituksella. Kuvassa (b) heijastukset on toteutettu suorakulmaisella särmiöllä lokalisoidulla ympäristökartalla. Kuva (c) esittää näkymän geometrian. Musta kartio osoittaa virtuaalikameran sijainnin, vihreä kuutio havainnollistaa ympäristökartan projektion keskipisteen ja oranssi rautalankasärmiö kuvaa ympäristökartan lokalisointiin käytetyn geometrian.



Kuvio 11. Kuviossa on havaittavissa lokalisoinnin epätarkkuus, kun lokalisointi perustuu laatikkosijaisgeometriaan. Kuvassa (a) tuolin heijastus näyttää litistyneen seinää vasten. Kuvassa (b) tarkempi heijastus on toteutettu kuvapuskuriheijastuksena.

karheudesta aiheutuva heijastuksen sumeus piilottaa suurimman osan häiritsevistä todellisen geometrian ja heijastuksissa havaittavan geometrian eroista. Jos tarvitsee heijastaa tarkkoja peiliheijastuksia, tällöin voi olla järkevämpää käyttää syvyyspuskuriin perustuvaa lokalisoimista. Tällöin tekniikka on hyvin lähellä kuvapuskuriheijastustekniikkaa. Tämä tuo lisää tarkkuutta heijastuksiin, mutta myös vie huomattavasti kauemmin laskea. Usean syvyyspuskurilokalisoidun tutkaimen käyttäminen voi olla liian hidasta, joten niitä kannattaa käyttää vain, kun laadulla on väliä.

Etäisyyteen perustuva heijastusten sumeus on mahdollista mallintaa lokalisoimisen ansiosta. Lokalisointi antaa arvion heijastuvan geometrian etäisyydestä valaistavaan pisteeseen, jonka lisäksi tiedämme ympäristötutkaimen projektion keskipisteen sijainnin. Voimme arvioida karheutta  $\sigma$  vastaavan kartion, jonka alueelta suurin osa heijastuvasta valosta saapuu pisteeseen. Tämän lisäksi arvioimme kartion leikkauksen heijastuvan geometrian kanssa ympyräksi, jonka säde on  $r$ . Kartion avauskulma  $\alpha$  suhteutuu ympyrän säteeseen  $r$  trigonometrian mukaan

$$\tan \frac{\alpha}{2} = \frac{r}{d}, \quad (6.1)$$

missä  $d$  on heijastussäteen kulkema matka. Samoin, kun tarkastellaan kyseistä ympyrää tutkaimen projektion keskipisteestä, saadaan

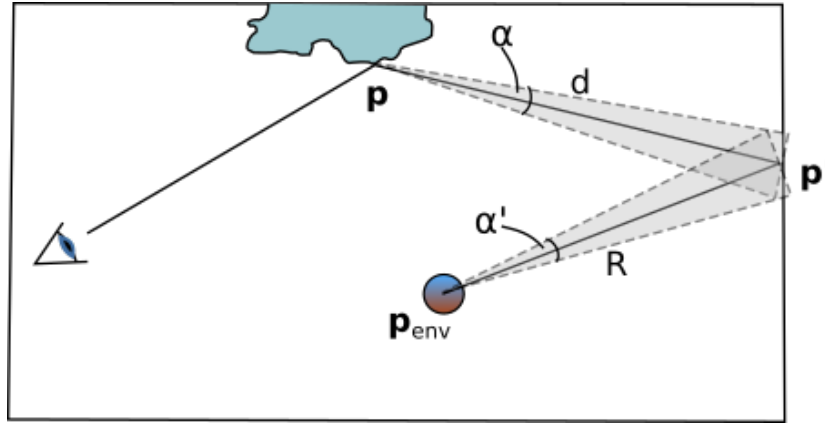
$$\tan \frac{\alpha'}{2} = \frac{r}{R}, \quad (6.2)$$

missä  $\alpha'$  on tutkaimen suunnasta avautuvan kartion avauskulma ja  $R$  on etäisyys projektion keskipisteestä  $\mathbf{p}_{env}$  heijastuvan geometrian törmäyspisteeseen  $\mathbf{p}'$ . Tilannetta havainnollistaa kuvio 12.

Koska karheus on suoraan verrannollinen kartion avauskulman tangentiin, voimme selvittää alkuperäisestä karheudesta  $\sigma$  etäisyyskorjatun karheuden

$$\sigma' = \frac{d}{R} \cdot \sigma. \quad (6.3)$$

Tulos on intuitiivinen, sillä säteen kulkeman matkan  $d$  ollessa suurempi kuin matka projektion keskipisteestä törmäyspisteeseen, on korjattu karheusarvo suurempi kuin alkuperäinen.



Kuvio 12. Kuvio havainnollistaa etäisyyteen perustuvaa karheuden korjausta.

Toisaalta, kun etäisyys  $d$  lähestyy nollaa, lähestyy korjattu karheuskin nollaa aiheuttaen heijastuksen tarkentumisen. Etäisyyteen perustuva karheuden korjaus on havaittavissa kuviossa 13.

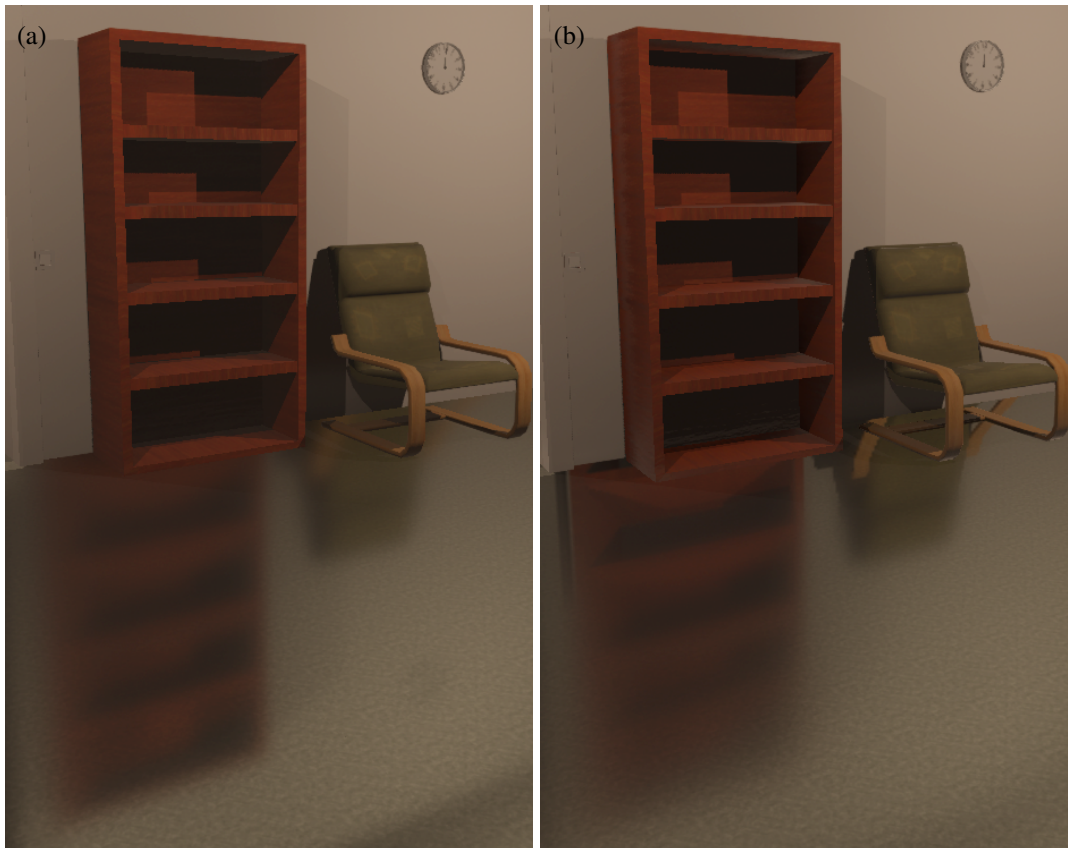
### 6.1.3 Ympäristökartan käyttäminen fysikaalisessa valaistuksessa

Reflektanssiyhtälö (3.6) on 4-ulotteinen isotrooppisen materiaalin tapauksessa ja 5-ulotteinen anisotrooppisen materiaalin tapauksessa. Lähdetään reflektanssiyhtälöstä

$$L(\mathbf{n}, \boldsymbol{\omega}_o) = \int_{\Omega} \rho(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) L(\boldsymbol{\omega}_i) (\mathbf{n} \cdot \boldsymbol{\omega}_i) d\boldsymbol{\omega}_i, \quad (6.4)$$

missä  $\mathbf{n}$  on pinnan normaali. Vektorit  $\mathbf{n}$  ja  $\boldsymbol{\omega}_o$  voidaan esittää pallokoordinaatistossa  $\mathbf{n} = (\alpha, \beta)$  ja  $\boldsymbol{\omega}_o = (\theta, \phi)$ . Näin nähdään, että isotrooppisen materiaalin reflektanssi on 4-ulotteinen. Anisotrooppiselle materiaalille tarvitaan vielä tangentin suunta, joka on kohtisuorassa pinnan normaaliin nähden. Näin anisotrooppinen reflektanssi on 5-ulotteinen. Reflektanssin taulukoiminen ei näin ollen ole erityisen tehokas ratkaisu. Vaikka taulukointi tarvitsee tehdä vain kerran referenssipistettä (ympäristökartan projektion keskipiste) kohden, on tallennettavaa dataa liian paljon, jotta sitä olisi helppo käsitellä nykyisellä grafiikkalaitteistolla.

Onkin siis joustavampaa laskea reflektanssi ajon aikana. Jotta tämä on reaaliajassa mahdollista, on käytettävä Monte Carlo -menetelmiä ja esimerkiksi suodatettua painotettua



Kuvio 13. Kuviossa ympäristötutkaimen lokalisoinnin avulla saatu arvio etäisyyteen perustuvasta karheuden aiheuttamasta sumennuksesta. Kuvassa (a) karheuden korjaus ei ole käytössä ja kuvassa (b) etäisyyteen perustuva karheuden korjaus on käytössä.

otantaa (engl. *filtered importance sampling, FIS*) (Colbert ja Křivánek 2007). Monte Carlo -menetelmät perustuvat tilastolliseen satunnaisotantaan. Monte Carlo -integroinnissa integraali diskretisoidaan ja otetaan  $N$  satunnaista näytettä, jolloin saadaan estimaatti

$$\int_{\Omega} \rho(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) L_i(\boldsymbol{\omega}_i) \underline{(\boldsymbol{n} \cdot \boldsymbol{\omega}_i)} \, d\boldsymbol{\omega}_i \approx \frac{1}{N} \sum_{k=1}^N \rho(\boldsymbol{\omega}_o, \boldsymbol{\omega}_k) L_i(\boldsymbol{\omega}_k) \underline{(\boldsymbol{n} \cdot \boldsymbol{\omega}_k)} . \quad (6.5)$$

Mitä useampi näyte integraalista otetaan, sitä lähempänä estimaatin arvo on integraalin oikeaa arvoa. Painotettu otanta ohjaa ottamaan näytteitä niiden tärkeyden perusteella. Kun otetaan enemmän näytteitä integraalin kohdista, jotka vaikuttavat enemmän lopputulokseen, vähenee integraalin varianssi ja estimaatti konvergoi nopeammin. Otannan painottaminen on mahdollista, kun integraalin luonteesta tiedetään jotain, esimerkiksi tässä tapauksessa kaksisuuntaisen heijastusjakaumafunktion muoto. Painotus tapahtuu todennäköisyys tiheysfunktiota (engl. *propability density function, PDF*)  $p$  käyttäen, jolloin estimaatiksi saadaan

$$\int_{\Omega} \rho(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) L_i(\boldsymbol{\omega}_i) \underline{(\boldsymbol{n} \cdot \boldsymbol{\omega}_i)} \, d\boldsymbol{\omega}_i \approx \frac{1}{N} \sum_{k=1}^N \frac{\rho(\boldsymbol{\omega}_o, \boldsymbol{\omega}_k) L_i(\boldsymbol{\omega}_k) \underline{(\boldsymbol{n} \cdot \boldsymbol{\omega}_k)}}{p(\boldsymbol{\omega}_o, \boldsymbol{\omega}_k)} . \quad (6.6)$$

Satunnaisotanta aiheuttaa lopputulokseen kohinaa, joka johtuu liian vähäisestä määrästä näytteitä. Kohinan poistamiseksi yleensä käytetään kvasisatunnaisia vähäisen poikkeaman lukujonoja (engl. *low discrepancy sequence*). Näin kohina muuttuu säännölliseksi, joka ei myöskään ole mielekäästä lopputuloksen kannalta. Kuvan laadun parantamiseksi alhaisilla näytemäärillä, voidaan käyttää suodatettua painotettua otantaa. Tällöin esisuodatettua eli sumennettua signaalia näytteistetään, kun näytteitä tarvittaisiin useita. Toisin sanoen esisuodatettu signaali suotimen alalta antaa arvion integraalin arvosta kyseiseltä alalta.

Koska edelleen integraalin laskeminen Monte Carlo -menetelmällä, vaikkakin vain vähällä näytemäärällä, on edelleen raskasta, voidaan kyseistä integraalia arvioida lisää. Oletetaan aluksi heijastusjakauman isotrooppisuus. Tekemällä oletus, että termit  $L_i(\boldsymbol{\omega}_i)$  ja  $\rho(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i)$  eivät riipu toisistaan, voidaan estimaatin summalauseke jakaa kahdeksi

$$\frac{1}{N} \sum_{k=1}^N \frac{\rho(\boldsymbol{\omega}_o, \boldsymbol{\omega}_k) L_i(\boldsymbol{\omega}_k) \underline{(\boldsymbol{n} \cdot \boldsymbol{\omega}_k)}}{p(\boldsymbol{\omega}_o, \boldsymbol{\omega}_k)} \approx \left( \frac{1}{N} \sum_{k=1}^N \frac{\rho(\boldsymbol{\omega}_o, \boldsymbol{\omega}_k) \underline{(\boldsymbol{n} \cdot \boldsymbol{\omega}_k)}}{p(\boldsymbol{\omega}_o, \boldsymbol{\omega}_k)} \right) \left( \frac{1}{N} \sum_{k=1}^N L_i(\boldsymbol{\omega}_k) \right) . \quad (6.7)$$

Tämä arvio on tarkka, jos pisteeseen saapuva radianssi on yksiväristä, ja aiheuttaa yleensä huomaamattoman vähän virhettä, jos ympäristöstä tuleva radianssi on monimuotoisem-

paa. Kun summa on jaettu kahdeksi summaksi, voidaan ne laskea erikseen ja kertoa sitten keskenään lopputuloksen saamiseksi. Radianssin summa voidaan laskea etukäteen ympäristökarttaan konvolvoimalla se heijastusjakaumafunktion  $\rho$  ja karheuden  $\sigma$  mukaan. Ympäristökarttana käytettävän kuutiotekstuurin mipmap-ketjun tasot sopivat hyvin säilyttämään eri karheutta vastaavaa esisuodatettua radianssin summaa. Karheudelle  $\sigma = 0$  käytetään suurinta tasoa ja karheudelle  $\sigma = 1$  pienintä. Konvoluution suorittamiseen voidaan käyttää suodatettua painotettua otantaa.

Kun tiedetään, että heijastusjakaumafunktiossa  $\rho$  käytetään Schlickin arviota Fresnel-heijastuskertoimesta, saadaan funktiota yksinkertaistettua. Schlickin Fresnel-heijastuskerroin voidaan muuntaa muotoon

$$F = F_{Schlick}(\mathbf{\omega}_i, \mathbf{h}) = F_0 + (1 - F_0)(1 - (\mathbf{\omega}_i \cdot \mathbf{h})^5) \quad (6.8)$$

$$= F_0(1 - (1 - (\mathbf{\omega}_i \cdot \mathbf{h})^5)) + (1 - (\mathbf{\omega}_i \cdot \mathbf{h})^5) \quad (6.9)$$

$$= F_0(1 - F_p) + F_p. \quad (6.10)$$

Näin heijastusjakaumafunktion summasta saadaan

$$\frac{1}{N} \sum_{k=1}^N \frac{\rho(\mathbf{\omega}_o, \mathbf{\omega}_k)}{p(\mathbf{\omega}_o, \mathbf{\omega}_k)} (\mathbf{n} \cdot \mathbf{\omega}_k) \quad (6.11)$$

$$= \frac{1}{N} \sum_{k=1}^N \frac{\rho(\mathbf{\omega}_o, \mathbf{\omega}_k)}{p(\mathbf{\omega}_o, \mathbf{\omega}_k) F} (F_0(1 - F_p) + F_p) (\mathbf{n} \cdot \mathbf{\omega}_k) \quad (6.12)$$

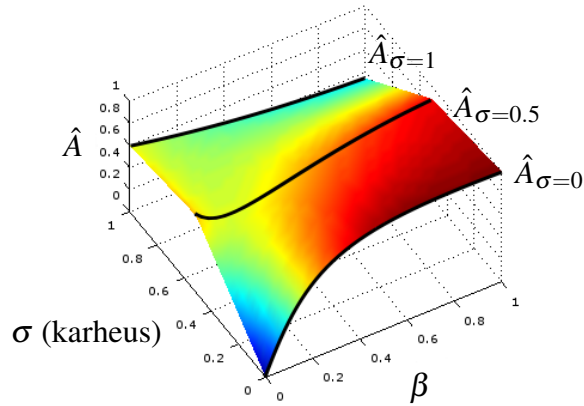
$$= F_0 \frac{1}{N} \sum_{k=1}^N \frac{\rho(\mathbf{\omega}_o, \mathbf{\omega}_k)}{p(\mathbf{\omega}_o, \mathbf{\omega}_k) F} (1 - F_p) (\mathbf{n} \cdot \mathbf{\omega}_k) + \frac{1}{N} \sum_{k=1}^N \frac{\rho(\mathbf{\omega}_o, \mathbf{\omega}_k)}{p(\mathbf{\omega}_o, \mathbf{\omega}_k) F} F_p (\mathbf{n} \cdot \mathbf{\omega}_k). \quad (6.13)$$

Näin muotoiltuna saadaan parametri  $F_0$  ulos heijastusjakaumafunktion summasta. Nimitetään tämä heijastusjakauman summafunktio ympäristön heijastusjakaumaksi ja merkitään sitä  $\rho_{env}$ . Funktiosta huomataan, että se on muotoa

$$\rho_{env}(\mathbf{n}, \mathbf{\omega}_o, \sigma, F_0) = F_0 \cdot A(\mathbf{n}, \mathbf{\omega}_o, \sigma) + B(\mathbf{n}, \mathbf{\omega}_o, \sigma), \quad (6.14)$$

missä  $\sigma \in [0, 1]$  on valaistavan pisteen karheus. Ympäristön heijastusjakauma riippuu nyt enää siis kahdesta vektoriarvosta ja pinnankarheudesta. Itse asiassa funktiot  $A$  ja  $B$  riippuvat vain vektorien  $\mathbf{n}$  ja  $\mathbf{\omega}_o$  välisestä kulmasta, tai sen kosinista  $(\mathbf{n} \cdot \mathbf{\omega}_o)$ . Nimitetään tämä kosini-





Kuvio 14. Funktion  $\hat{A}$  interpolointi funktioista  $\hat{A}_\sigma$ , kun  $\sigma = 0, 0.5, 1$ .

termi  $\beta$ :ksi ( $\beta \in [0, 1]$ ) merkintöjen helpottamiseksi.

Heijastusjakaumafunktion parametrifunktiot  $A$  ja  $B$  voidaan taulukoida parametrien  $\sigma$  ja  $\beta$  mukaan ja tallentaa kaksiulotteiseen kaksikanavaiseen tekstuuriin (Karis 2013). Tämä kuitenkin vaatii yhden ylimääräisen tekstuurinäytteen jokaista valaistua pikseliä kohden ja varaa yhden ylimääräisen teksturointiyksikön. Jos ylimääräinen tekstuurinäyte on liikaa, vaihtoehtona on sovittaa ympäristön heijastusjakaumafunktioon analyttinen funktio  $\hat{\rho}_{env}$ , joka on muotoa

$$\hat{\rho}_{env}(\beta, \sigma, F_0) = F_0 \cdot \hat{A}(\beta, \sigma) + \hat{B}(\beta, \sigma). \quad (6.15)$$

Funktioille  $A$  ja  $B$  voidaan löytää hyvä approksimaatio käyttäen niin kutsuttuja Padé-funktioita. Funktiota  $f$  arvioiva Padé-funktio on muotoa  $[n, m]_f(x) = P_n(x)/Q_m(x)$ , missä  $P_n$  on  $n$ -asteinen ja  $Q_m$   $m$ -asteinen polynomi. Kyseinen Padé-approksimaatio alkuperäisestä funktiosta saadaan käyttämällä funktion Taylorin sarjakehitelmää apuna. Tässä kuitenkin teemme poikkeuksen ja sovitamme Padé-funktion kertoimet numeerisesti, sillä funktioiden  $A$  ja  $B$  analyttisten arvioiden muodostaminen ei ole tarpeen.

Sovitetaan ensin funktiot  $\hat{A}_\sigma(\beta) = [n, m]_A(\beta, \sigma)$  ja  $\hat{B}_\sigma(\beta) = [n, m]_B(\beta, \sigma)$  karheusarvoille  $\sigma_i, i = 1, 2, \dots, k$ . Lopullinen arvio  $\hat{A}$  saadaan interpoloimalla lineaarisesti karheusarvon mukaan funktioita  $\hat{A}_\sigma$  ja tehdään samoin arvion  $\hat{B}$  saamiseksi. Kuvio 14 havainnollistaa tilannetta funktion  $\hat{A}$  tapauksessa.

Taulukko 2. Näytönohjaimella suoritettujen operaatioiden aikavaativuuden arvioita normalisoituna yhteenlaskuoperaation  $A$  mukaan. Arviot on saatu suorittamalla eri operaatioita useita kertoja Nvidia Geforce GTX 960 -näytönohjaimella.

$A$	$M$	$D$	$C$
1.00	1.00	1.97	7.80

$A$  ja  $B$  funktioiden arvioitu laskennallinen vaativuus, kun polynomien arvoa laskettaessa käytetään Hornerin muotoa, on

$$cost(k, n, m) = (k - 1)(2D + (n + m)M + (n + m + 4)A + 2C), \quad (6.16)$$

missä  $m$  ja  $n$  ovat Padé-funktion polynomien asteet ja  $k$  on interpoloitavien eri karheusarvoa vastaavien Padé-funktioiden määrä. Muuttuja  $D$  vastaa jakolaskun vaativuutta,  $M$  kertolaskun vaativuutta,  $A$  yhteen- tai vähennyslaskun vaativuutta ja  $C$  vertailuoperaation vaativuutta. Taulukossa 3 on listattu muutamia aikavaativuuksia eri muuttujien  $k$ ,  $m$  ja  $n$  arvoilla. Eri operaatioiden aikavaativuuksien arviot on listattu taulukossa 2. Operaatioiden suorittamiseen kuluva aika riippuu hyvin paljon näytönohjaimesta ja peräkkäisten operaatioiden keskinäisistä riippuvuuksista, joten nämä arviot ovat vain suuntaa-antavia. Maksimivirhe (engl. *maximum absolute error*)  $ME$  on laskettu koko ympäristön heijastusjakaumafunktion arviolle  $\hat{\rho}_{env}$  käyttäen kaavaa

$$ME(\hat{\rho}_{env}) = \max\{|\hat{\rho}_{env,i} - \rho_{env,i}| : i = 1, \dots, N\}, \quad (6.17)$$

ja virheiden neliöllinen keskiarvo (engl. *root mean square error*,  $RMSE$ ) on laskettu yhtälöllä

$$RMSE(\hat{\rho}_{env}) = \sqrt{\frac{\sum_{i=1}^N (\hat{\rho}_{env,i} - \rho_{env,i})^2}{N}}, \quad (6.18)$$

missä  $N$  on painotettua otantaa käyttäen laskettujen referenssidatapisteiden määrä.

Taulukoitujen virheiden perusteella näyttää siltä, että virhe on pienempi kun interpoloitavien funktioiden määrä on neljä ( $k = 4$ ). Kuitenkin eri lopputuloksia vertailemalla nähdään, että visuaalista virhettä on havaittavissa vähiten tilanteessa, jossa interpoloitavia funktioita on

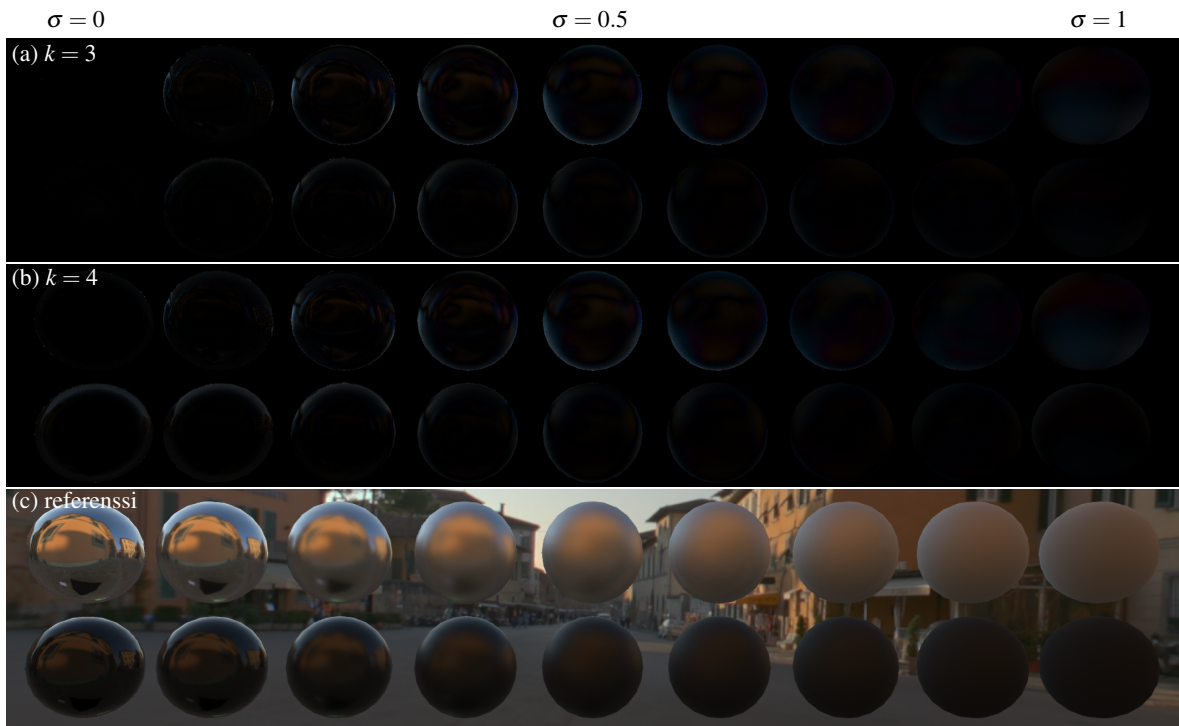
Taulukko 3. Funktion  $\hat{A}$  ja  $\hat{B}$  arvioitu aikavaativuus, maksimivirhe ja virheiden neliöllinen keskiarvo eri määrillä interpoloitavia funktioita ja funktioiden asteilla. Hintasarake esittää karkean arvion kokonaisvaativuudesta, jossa operaatioiden vaativuuksina on käytetty taulukossa 2 esitettyjä yhteenlaskun mukaan normalisoituja aikoja.

$k$	$m$	$n$	Operaatiot	Hinta	$ME(\hat{\rho}_{env})$	$RMSE(\hat{\rho}_{env})$
3	2	2	$4D + 8M + 16A + 4C$	63.08	0.381575	0.045139
3	3	3	$4D + 12M + 20A + 4C$	71.08	0.359193	0.045223
3	4	4	$4D + 16M + 24A + 4C$	79.08	0.359774	0.045215
4	2	2	$6D + 12M + 24A + 6C$	94.64	0.370717	0.032562
4	3	3	$6D + 18M + 30A + 6C$	106.62	0.350944	0.028058
4	4	4	$6D + 24M + 36A + 6C$	118.62	0.358444	0.027879

kolme ja funktioiden aste on neljä ( $k = 3, m = n = 4$ ). Ero referenssikuvaan tilanteissa, joissa  $k = 3$  ja  $k = 4$  on esitetty kuviossa 15. Kuvioista havaitaan, että pienellä karheusarvolla neljän interpoloitavan funktion tapauksessa virhe on suurempi kuin kolmen funktion tapauksessa pallojen reunalla, jossa katselusuunnan ja normaalin välinen kulma on suuri. Muualla virheen määrä on molemmille arvioille samaa luokkaa. Virheen käyttäytyminen selittyy sillä, että kolmen interpoloitavan funktion tilanteessa virhe on jakautunut tasaisemmin paikoissa, joissa tarkkuudella on visuaalisesti enemmän merkitystä.

Oletettavasti virhe pienenesi, jos interpoloitavia funktioita olisi viisi tai enemmän, mutta kasvattamalla funktioiden astetta ei virhettä saada pienennettyä juurikaan. Useamman kuin neljän interpoloitavan funktion tilanteessa arvion vaativuus kasvaa jo epäkäytännölliseksi. Lisäksi jo kolmen interpoloitavan funktion visuaalinen virhe on niin pientä, ettei useamman funktion interpoloinnille ole mitään syytä. Jatkossa onkin käsitelty vain tilannetta, jossa interpoloitavia funktioita on kolme ja niiden aste on neljä.

Taulukossa 4 on listattu sovitettujen  $\hat{A}_\sigma$  ja  $\hat{B}_\sigma$  funktioiden maksimivirhe ja virheiden neliöllinen keskiarvo. Molemmat funktiot saavat arvoja väliltä nolasta yhteen, joten virheen suuruus on keskimäärin hyvin pientä. Taulukossa 5 on listattu maksimivirhe ja virheiden neliöllinen keskiarvo funktioille  $\hat{A}$ ,  $\hat{B}$  ja  $\hat{\rho}_{env}$ . Huomataan, että suurempi virhe aiheutuu, kun



Kuvio 15. Kuvio esittää erotuskuvan arvion ja referenssin välillä. Erotuskuvien väriarvot on kaksinkertaistettu, jotta ero olisi paremmin havaittavissa. Kuvan (a) arviossa on kolme interpoloitavaa funktiota, joiden aste on neljä ( $k = 3$ ,  $m = n = 4$ ). Kuvan (b) arviossa on neljä interpoloitavaa funktiota, joiden aste on neljä ( $k = 4$ ,  $m = n = 4$ ). Kuva (c) on suodatettua painotettua otantaa käyttäen renderöity referenssi. Kuvien ylärivissä materiaali on metallinen ja alarivissä dielektrinen. Molemmilla diffuusi albedo on musta.

Taulukko 4. Funktioiden  $A(\beta, \sigma)$  ja  $B(\beta, \sigma)$  painotettua otantaa käyttäen laskettuihin datapisteisiin sovitettujen funktioiden, kolmen  $\hat{A}_\sigma$  ja kolmen  $\hat{B}_\sigma$  ( $\sigma = 0, 0.5, 1$ ) funktion, maksimivirhe ja virheiden neliöllinen keskiarvo, kun  $\beta$  saa arvot väliltä  $[0, 1]$ .

	$ME(\hat{A}_\sigma)$	$ME(\hat{B}_\sigma)$	$RMSE(\hat{A}_\sigma)$	$RMSE(\hat{B}_\sigma)$
$\sigma = 0$	0.007497	0.008585	0.000557	0.000629
$\sigma = 0.5$	0.002784	0.000672	0.000283	0.000071
$\sigma = 1$	0.000988	0.000114	0.000067	0.000008

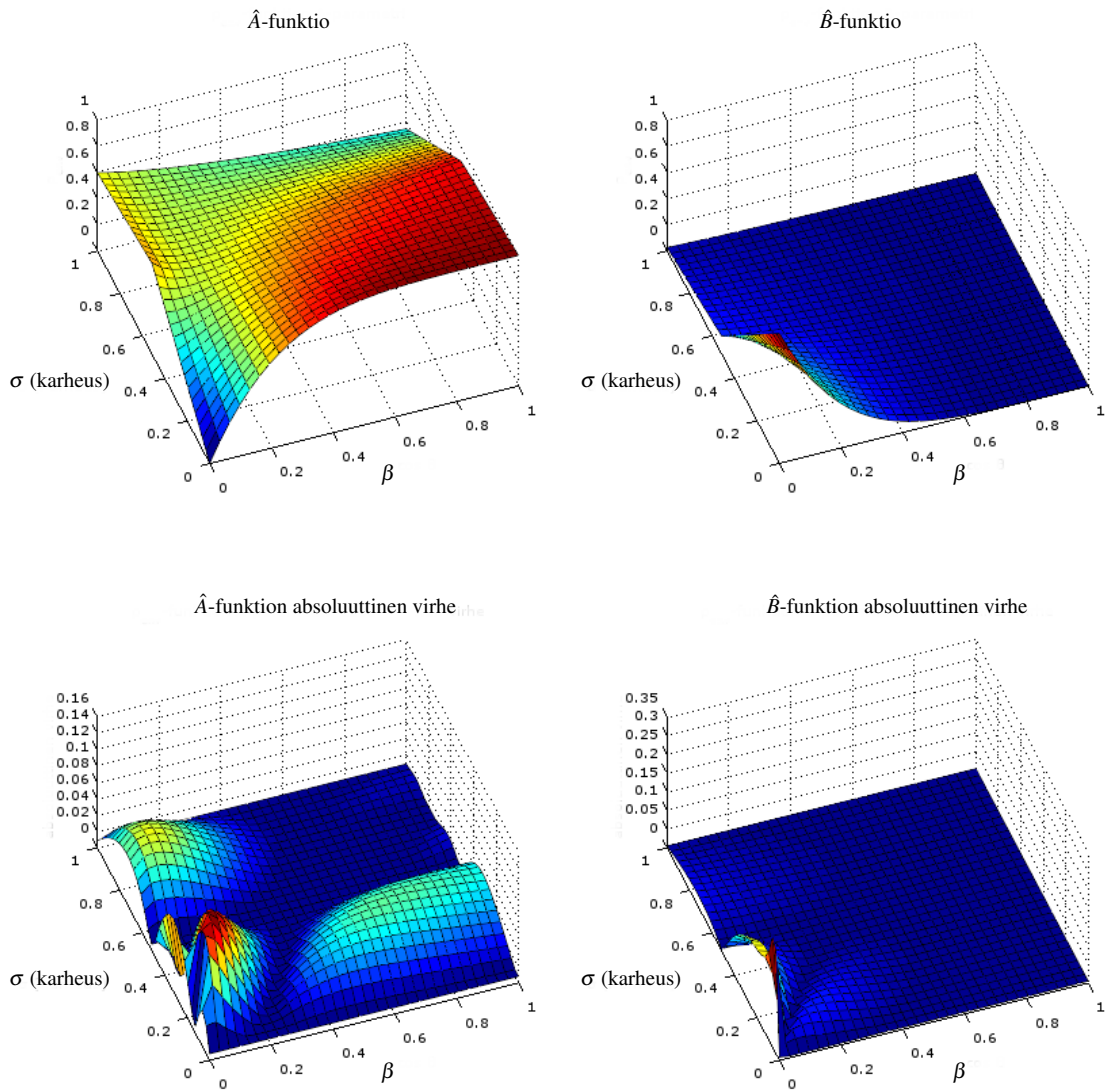
Taulukko 5. Funktioiden  $\hat{A}(\beta, \sigma)$ ,  $\hat{B}(\beta, \sigma)$  ja  $\hat{\rho}_{env}(\beta, \sigma, F_0 = 1)$  maksimivirhe ja virheiden neliöllinen keskiarvo, kun  $\sigma$  saa arvot väliltä  $[0, 1]$  ja  $\beta$  saa arvot väliltä  $[0, 1]$ .

$ME(\hat{A})$	$ME(\hat{B})$	$ME(\hat{\rho}_{env})$	$RMSE(\hat{A})$	$RMSE(\hat{B})$	$RMSE(\hat{\rho}_{env})$
0.123433	0.320045	0.359774	0.035204	0.025565	0.045215

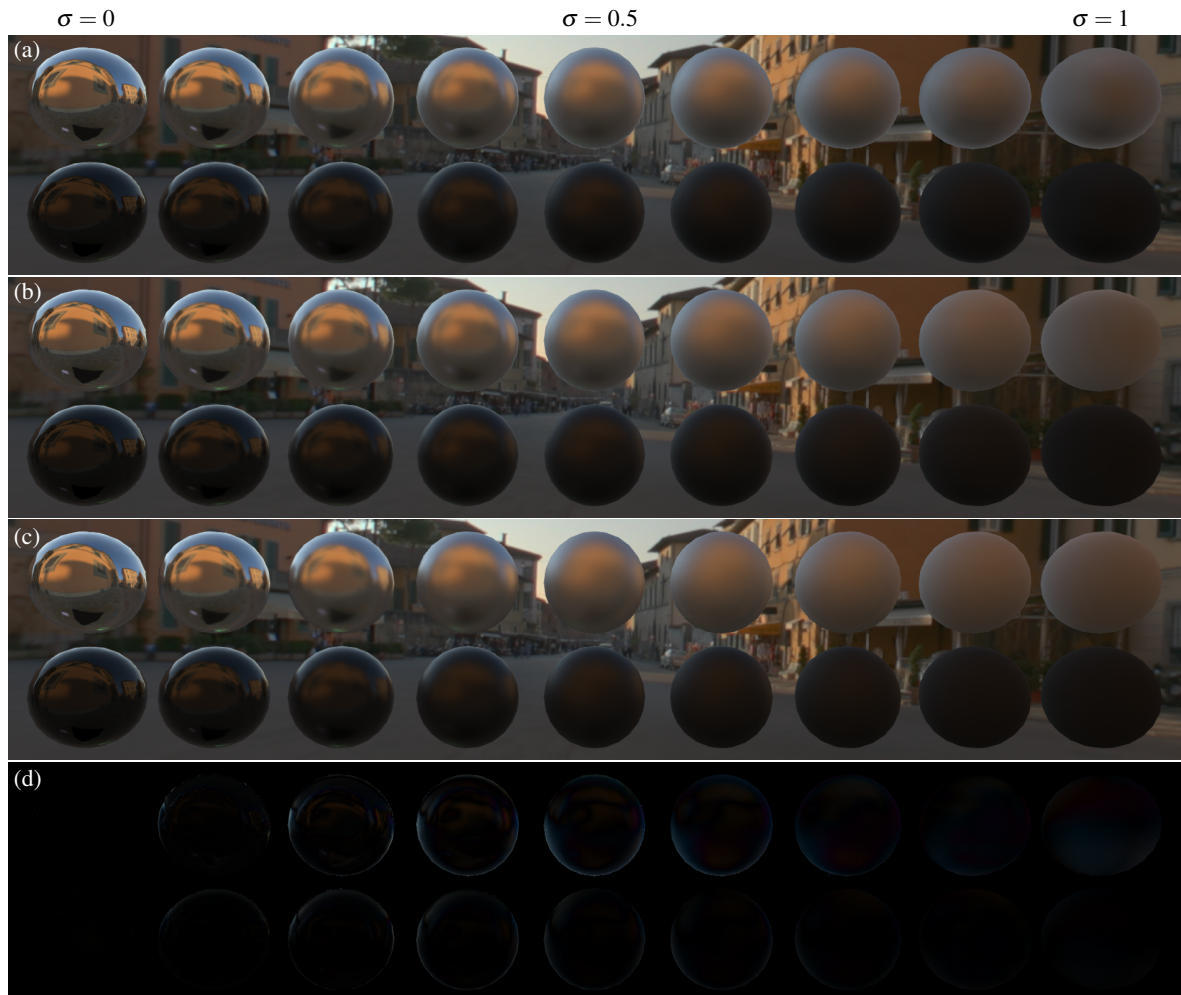
funktioita interpoloidaan lineaarisesti lopullisen  $\hat{A}$  ja  $\hat{B}$  funktion arvon saamiseksi. Näin saadut funktiot  $\hat{A}$  ja  $\hat{B}$  ja niiden absoluuttiset virheet on esitetty kuviossa 16. Etenkin funktion  $\hat{A}$  kuvaajasta on huomattavissa lineaarisen interpoloinnin vaikutus. Jokaiselle parametrin  $\beta$  arvolle funktio on suora välillä  $\sigma \in [0, 0.5]$  ja välillä  $\sigma \in [0.5, 1]$ .

Kuviossa 17 on esitetty vertailu arvioivan toteutuksen ja referenssinä Monte Carlo -menetelmällä painotettua otantaa käyttävän toteutuksen välillä. Kuvissa jokaisen pallon karheus kasvaa lineaarisesti siirryttäessä vasemmalta oikealle. Lisäksi jokaisen kuvan ylärivissä materiaali on metallinen ja alarivissä dielektrinen eli sähköneriste. Vertailusta nähdään, että lineaarinen interpolointi ei aiheuta suurta visuaalista virhettä lopputulokseen, vaan suurempi virhe tulee ympäristökartan näytteistysuunnasta. Kun suuntana käytetään pelkkää peiliheijastussuuntaa (kuvio 17a), virhe kasvaa suuremmaksi karheuden kasvaessa. Jos sen sijaan näytteistetään normaalijakauman mukaisesta dominoivasta suunnasta (kuvio 17b), saadaan virhe pidettyä huomattavasti pienempänä suurille karheusarvoille.

Dominoiva suunta  $\mathbf{D}$  on saatu kokeellisesti interpoloimalla peiliheijastussuunnasta  $\boldsymbol{\omega}_r$  normaalin  $\mathbf{n}$  suuntaan karheuden  $\sigma$  mukaan



Kuvio 16. Funktioiden  $\hat{A}$  ja  $\hat{B}$  kuvaajat ja absoluuttiset virheet. Etenkin funktion  $\hat{A}$  kuvaajasta on nähtävissä lineaarisen interpoloinnin vaikutus.



Kuvio 17. Kuvissa (a) ja (b) on käytetty ympäristön heijastusjakaumafunktion arviota. Kuvassa (a) ympäristöstä tulevan valon näyte otetaan peiliheijastussuunnasta kun taas kuvassa (b) näytteistetään normaalijakauman dominoivasta suunnasta. Kuva (c) on referenssikuva, joka on renderöity Monte Carlo -menetelmällä käyttäen suodatettua painotettua otantaa. Kuva (d) on kuvien (b) ja (c) välinen erotuskuva, jonka väriarvot on kaksinkertaistettu, jotta ero olisi paremmin havaittavissa.

$$D(\omega_r, n, \sigma) = (1 - \sigma^3)\omega_r + \sigma^3 n . \quad (6.19)$$

Näin saadun dominoivan suunnan taustalla on ajatus karheuden vaikutuksesta heijastukseen. Mitä suurempi karheus, sitä diffuusimpi heijastus on. Interpoloimalla peiliheijastusmallin ja diffuusin mallin välillä, voidaan saada kokeellinen arvio dominoivasta suunnasta.

## 6.2 Kuvapuskuriheijastukset

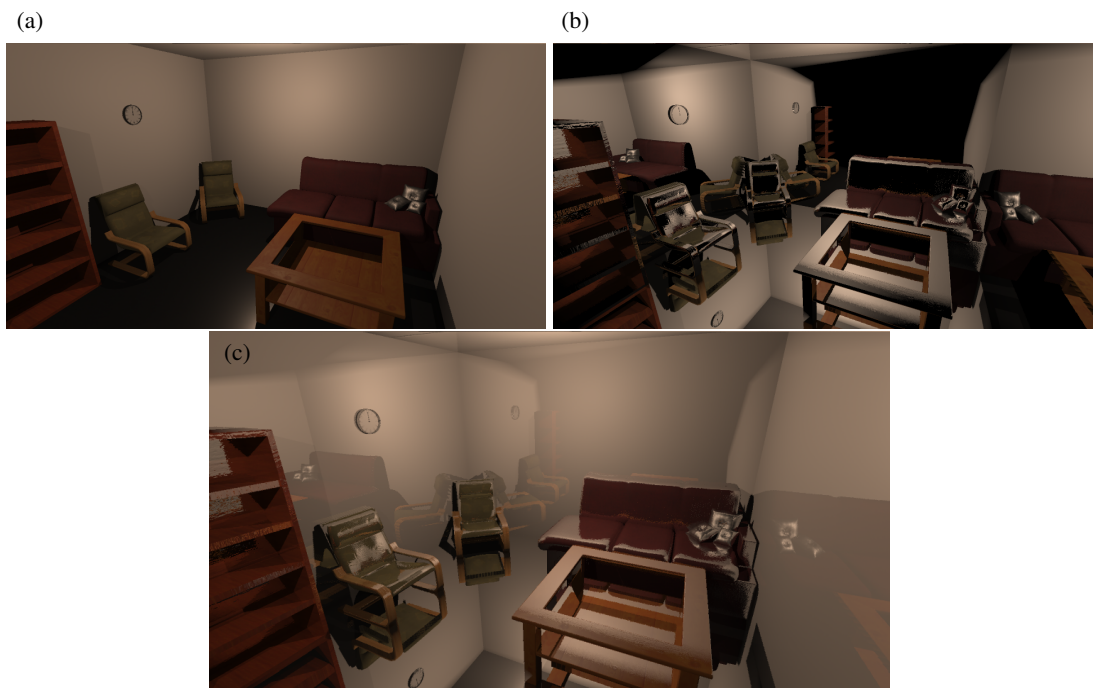
Kuvapuskuriheijastustekniikka soveltuu tuottamaan lokaaleja heijastuksia ympäristön kappaleista, sekä karheista että peilimäisistä pinnoista. Koska tekniikka tukeutuu vain kuvapuskurissa olevaan informaatioon, ei sen suorituskyky ole riippuvainen ympäristön geometriasta. Toisaalta tämä on myös tekniikan suurin ongelma: vain kuvapuskurissa näkyvä geometria voi heijastua. Tämä näkyy kuvion 18 esittämästä heijastuspuskurista, jossa huoneen takaseinästä lähteneet heijastussäteet eivät päädy kuvapuskurin alueelle, vaan kameran taakse. Huoneen takaseinä jää siis heijastuspuskurissa mustaksi ilman ympäristöstä heijastuvaa valoa. Toisaalta paikoissa, joissa heijastussäteet osuvat kuvapuskuriin, ovat heijastukset erittäinkin tarkkoja ja uskottavia.

### 6.2.1 Kuvapuskuriheijastusten rajoitteita

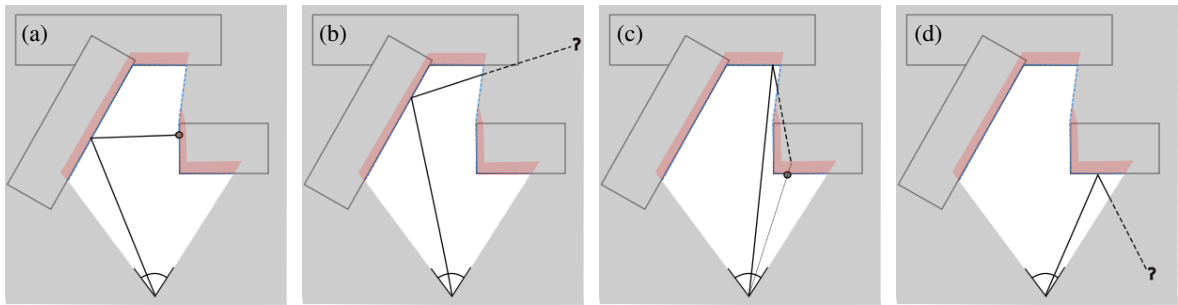
Kuvapuskuriin perustuvat tekniikat kärsivät kaikki samasta ongelmasta: kuvapuskurin antamasta rajallisesta informaatiosta. Sama pätee kuvapuskuriheijastuksiin. Heijastuksen säteen seuranta voi epäonnistua, kun heijastussäde kulkee kuvapuskurin alueen ulkopuolelle, tai hypätä geometrian ohitse, jos askelluksen välistystä on kasvatettu tehokkuuden lisäämiseksi. Koska syvyyspuskuri esittää vain yhden syvyysarvon, täytyy törmäyspisteen etsintää varten määritellä kuvaelementille paksuus. Koska kuvaelementin paksuus on vakio, se ei vastaa oikean geometrian paksuutta.

Kuviossa 19 on esitetty eri tilanteita, joita säteenseurannassa tulee vastaan. Kuvassa punertava alue katselupisteestä nähtävän geometrian etuosassa on arvio kuvaelementtien paksuudesta. Osa säteistä törmää kuvapuskurissa olevaan geometriaan. Usein säde kuitenkin kulkee todellisen geometrian lävitse, koska kuvaelementin paksuus oli liian pieni, tai törmää ohu-





Kuvio 18. Kuvassa (a) on pelkän suoran valaistuksen vaikutus. Kuvassa (b) näkyy heijastuspuskuri, joka on saatu kuvapuskuriheijastusten säteenseurantaa käyttäen. Kuvassa (c) suora ja epäsuora valaistus on yhdistetty naiivisti ottamatta huomioon heijastavan pinnan fysikaalisia ominaisuuksia.

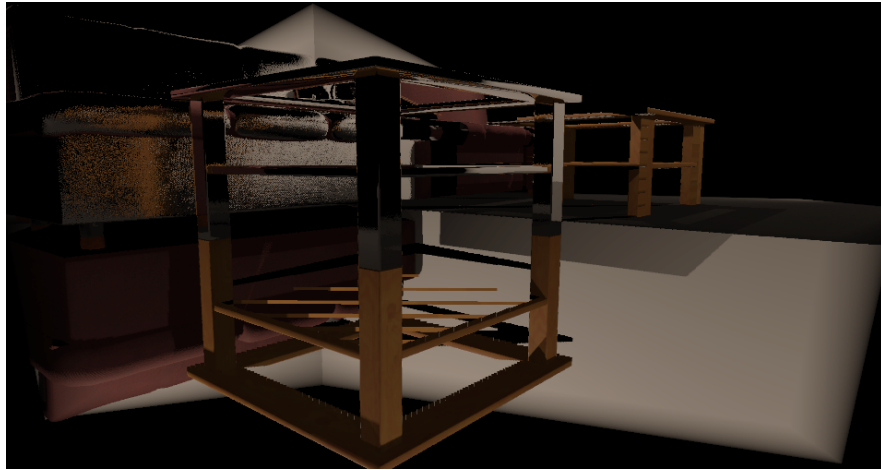


Kuvio 19. Kuvio esittää kuvapuskuriheijastusten säteenseurannassa vastaan tulevat eri tilanteet. Kuvassa (a) heijastussäde törmää kuvapuskuriin. Kuvassa (b) säde kulkee kappaleen takana, mutta liian kaukana, että törmäys tapahtuisi edessä olevaan kappaleeseen. Kuvassa (c) säde törmää kappaleen etuseinään, vaikka todellinen törmäys tapahtuisi kappaleen taka-seinään. Kuvassa (d) säde ei päädy kuvapuskurin alueelle.

een geometriaan, koska paksuus oli liian suuri. Paremman arvion geometrian paksuudelle saa, jos etukäteen piirretään syvyyspuskuri, johon on piirretty pelkästään geometrian taustatahkot. Nyt kuvaelementin paksuuden saa normaalin syvyyspuskurin ja taustatahkosyvyyspuskurin arvojen erotuksena. Ongelmana tässä ratkaisussa on se, että koko näkymä tulee renderöidä kahdesti.

Heijastussäteen päätyessä kuvapuskurin ulkopuolelle, näkyy heijastuksissa selkeä reuna, jossa heijastusinformaatio päättyy. Usein halutaan, että tämä muutos olisi hienovaraisempi. Tämä voidaan toteuttaa himmentämällä heijastuksen voimakkuutta, kun heijastussäteen törmäyspiste lähenee kuvapuskurin reunaa. Samoin, jos säteenseurannalle on tehokkuussyistä määritelty maksimietäisyys, voidaan heijastuksen voimakkuutta vähentää, kun säteen kulke-matka lähestyy maksimietäisyyttä. Näillä toimenpiteillä terävät muutokset heijastuksissa saadaan piilotettua.

Vaikka kuvapuskuriheijastuksilla voidaan saada hyvinkin tarkkoja peiliheijastuksia, on yleensä heijastuksissa nähtävissä erilaisia virheitä. Osa virheistä aiheutuu säteenseurannan tarkkuudesta. Säteenseurannan nopeuttamiseksi jokaista kuvaelementtiä ei välttämättä tarkasteta, vaan edetään useampia kuvaelementtejä kerrallaan. Näin harppaamalla voidaan vahingossa ohittaa geometriaa, johon säteen olisi pitänyt törmätä. Toisaalta liian paksuksi oletettu kuvaelementti aiheuttaa vääriä heijastuksia. Näitä virheitä havainnollistaa kuvio 20,

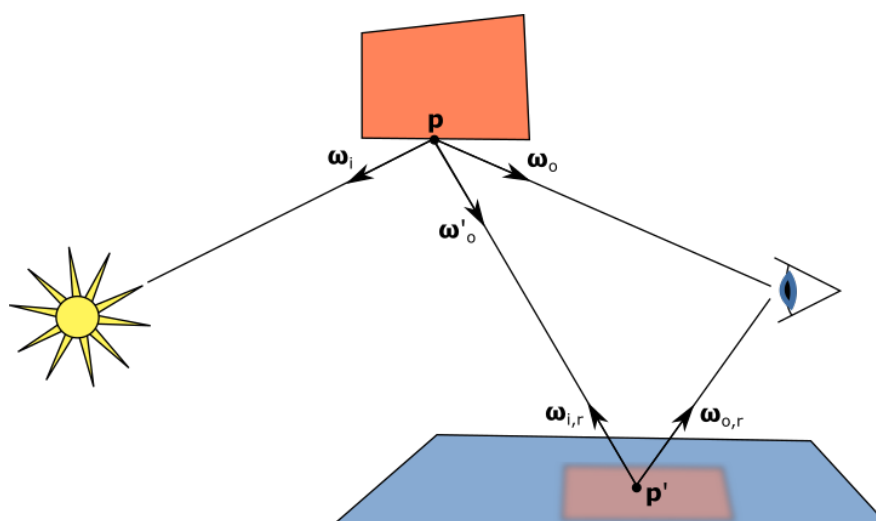


Kuvio 20. Kuvassa on kuvapuskuriheijastusten tuottama heijastuspuskuri, jossa näkyy säteenseurannan harppauksesta johtuvia virheitä. Lattiassa heijastuvan pöydän välitaso ei heijastu kokonaan.

jossa lattiassa heijastuvan pöydän välitaso on selvästi osin jäänyt heijastumatta. Kuviossa havaittavia virheitä saadaan piilotettua, kun käytetään tarpeeksi suurta karheusarvoa. Näin heijastukset ja niissä näkyvät virheet sumentuvat, ja virheitä on vaikeampi erottaa. Harppaus aiheuttaa heijastuksissa myös porrasmaisia virheitä. Näitä voidaan vähentää tai piilottaa aiheuttamalla säteen alkupisteeseen pientä vaihtelua. Näin porrasmaiset virheet muuttuvat rakeisiksi, jotka lisätyn karheuden aiheuttaman sumeuden mukana piiloutuvat ihmissilmältä paremmin, kuin säännöllisemmät ja terävämmät porrasmaiset virheet.

Kuvapuskuriheijastustekniikka ei sovellu esimerkiksi peilien toteuttamiseen. Jos esimerkiksi ensimmäisen persoonan näkökulmasta käyttäjä haluaa nähdä avatarinsa heijastuksen peilistä, ei tätä voida kuvapuskuriheijastuksilla toteuttaa. Tasoheijastukset soveltuvat tähän paremmin.

Lisäksi kuvapuskuriheijastukset eivät tue osittain läpinäkyvää geometriaa, kuten lasia tai läpinäkyvyyttä hyödyntäviä efektejä kuten partikkeliefektejä. Näiden efektien piirto tapahtuu yleensä myöhemmin, kun kuvapuskuriheijastusvaihe on jo suoritettu. Näin läpinäkyvyyttä hyödyntävät kappaleet eivät pysty hyötymään kuvapuskuriheijastuksista, eivätkä ne myöskään heijastu muissa kappaleissa.



Kuvio 21. Kuvio esittää heijastusgeometrian, jossa piste  $p$  heijastuu pisteessä  $p'$ . Kuvapuskuriheijastuksissa pisteeseen  $p'$  pisteestä  $p$  saapuvaa radianssia arvioidaan valaistuspuskurissa olevalla pisteestä  $p$  kameraan saapuvalla radianssilla.

### 6.2.2 Kuvapuskuriheijastukset fyysisessä valaistuksessa

Kuvapuskuriheijastuksissa valaistuspuskurista otettu heijastusnäyte ei vastaa fyysisesti oikeaa radianssia. Valaistuspuskuri kuvaa näkymän suoraa valaistusta, joka on laskettu analyttisten valojen tulosuunnasta ja katselusuunnasta. Se ei tällöin vastaa kuvaruudun pisteeseen  $p'$  heijastuvaa radianssia, joka laskettaisiin pisteeseen heijastuva valon tulosuunnasta ja heijastuvaan pisteeseen  $p$  saapuvan valon tulosuunnasta. Kuvio 21 havainnollistaa tätä ongelmaa. Pisteeseen  $p$  analyttisestä valonlähteestä saapuva radianssi on  $L(\omega_o, p)$ , kun taas pisteeseen  $p'$  pisteestä  $p$  heijastuva radianssi on  $L(\omega'_o, p)$ .

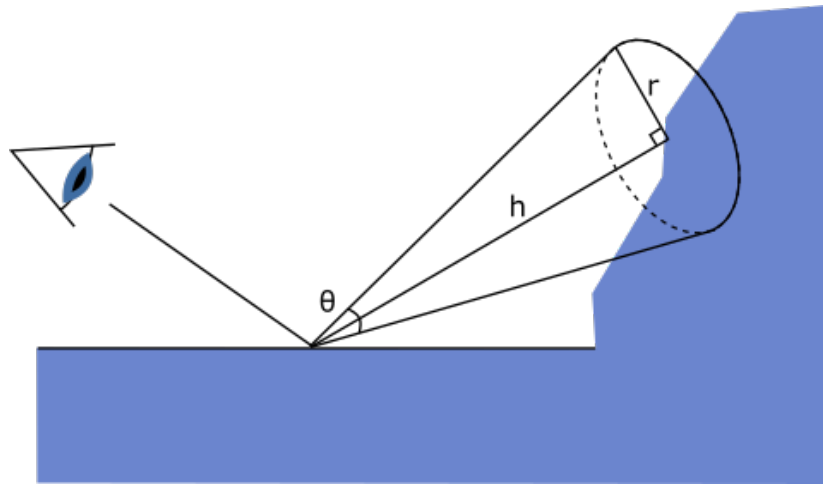
Jos haluttaisiin fyysisesti tarkempi tulos, tulisi suora valaistus laskea jokaiselle heijastuvalle pisteelle uudelleen, kun kuvapuskuriheijastusta lasketaan. Tämä tarkoittaisi, että jouduttaisiin lukemaan heijastuvan pisteen g-puskuriin tallennetut materiaali-parametrit ja laskemaan kaikkien pisteeseen vaikuttavien valojen aiheuttama suora valaistus kyseisessä pisteessä. Tämä ei kuitenkaan sovellu tehokkaasti viivästettyyn varjostukseen. Sen sijaan tyydytään pitämään kuvapuskurissa olevaa radianssia tarpeeksi hyvänä arviona kuvapuskurin muille pisteille.

Koska säteenseurantaa suoritettaessa tarvitaan valaistuspuskuri, josta heijastuva valaistus voidaan näytteistää, täytyy valaistuspuskurin piirron olla valmis ennen säteenseurannan aloittamista. Jotta tämä viive saadaan poistettua, voidaan käyttää edellisen piirretyn kuvaruudun valaistuspuskuria, jonka näkymään löydetty törmäyspiste voidaan projisoida näytteistystä varten. Tästä saadaan myös yllättävä toinen hyöty: arvio toisen asteen heijastuksille. Jos edellisen kuvaruudun valaistuspuskuri sisältää ensimmäisen asteen heijastusinformaatiota, niin sitä näytteistäessä uusi heijastuspuskuri sisältää toisen asteen heijastusinformaatiota. Tämä logiikka pätee rekursiivisesti niin kauan, kunnes heijastusketju katkeaa, kun heijastussäde ei päädy kuvapuskurin alueelle. Tarkemmin sanottuna, kuvaelementin heijastusaste kasvaa jokaisella uudella heijastuspuskurin piirrolla, jos edellinen kuvapuskuri sisältää edellisen heijastusasteen informaatiota kyseiselle kuvaelementille.

Jotta kuvapuskuriheijastukset toimisivat fysikaalisesti uskottavalla tavalla, tulee heijastusten sumentua, kun pinnan karheus kasvaa. Tämä aiheutuu siitä, että mitä karheampi pinta on, sitä laajemmalla alalta tulevaa valoa pinta heijastaa katselusuuntaan. Jotta sumennuksen määrä vastaa samaa kuin analyyttisten valojen aiheuttama suoralle valolle, täytyy sen toimia saman heijastusjakaumafunktion mukaisesti. Suoraviivaisin ja fysikaalisesti oikein tapa olisi käyttää Monte Carlo -näytteistystä ja suorittaa säteenseuranta usealle eri suuntiin lähteville säteelle ja yhdistää lopputulos. Tämä vaatii kuitenkin todella paljon säteenseurantaa, joka on kuvapuskuriheijastuksissa ajallisesti vaativin vaihe. Parempi olisi siis käyttää suodatettua painotettua otantaa, kuten ympäristökartoituksessa. Näin säteenseurannan määrää saataisiin vähennettyä, mutta edelleen tekniikka olisi erittäin raskas.

Tekniikasta saadaan tehokkaampi, jos oletetaan, että tarkan säteenseurannan tuloksena saatavan heijastuspuskurin sumentaminen vastaa tarpeeksi tarkasti sumentumista karheudesta johtuen. Siis sen sijaan, että suoritettaisiin säteen seurantaa useaan kertaan jokaiselle kuvaelementille, arvioidaan että tarkassa heijastuspuskurissa kuvaelementin naapurusto vastaa niiden säteiden tuloksia, jotka kuvaelementille saataisiin, jos seurattaisiin useampia säteitä. Tämä toteutetaan esisumentamalla heijastuspuskuri heijastusjakaumaa vastaavaksi mipmapketjuksi.

Tiettyä karheutta vastaava sumennuksen määrä riippuu siitä, miltä alalta merkittävä osa valoa saapuu pisteeseen. Tämä ala taas riippuu heijastusjakauman muodosta ja etäisyydestä.



Kuvio 22. Kuvio esittää karheasta pinnasta katselusuuntaa heijastuvaa valoa kartiona, joka vastaa heijastusjakauman merkittävimpiä suuntia.

Heijastusjakauman muotoa voidaan arvioida kartiolla, jonka kärki on tarkasteltavassa pisteessä ja suuntautuu heijastussuuntaan kuvion 22 osoittamalla tavalla. Kartion kärjen avautumiskulma  $\theta$  saadaan laskettua käyttämällä heijastusjakauman painotetussa otannassa käytettävää näytteistys suunnan antavaa kaavaa. Näytteistys suunta pallokoordinaatistosta karteesiseen koordinaatistoon GGX-normaalijakaumalle saadaan kaavalla

$$\cos \theta = \sqrt{(1 - \xi) / (1 + (\sigma^2 - 1) \cdot \xi)} \quad (6.20)$$

$$\sin \theta = \sqrt{1 - \cos^2 \theta}, \quad (6.21)$$

missä  $\xi \in [0, 1]$  on satunnaismuuttuja. Tästä saamme ratkaistua arvon kulmalle  $\theta$ . Painotetussa otannassa satunnaismuuttuja  $\xi$  määrää painotetun suunnan koko puolipallon sisältä. Koska emme halua kartion olevan puolipallo, kiinnitämme muuttujan  $\xi$  arvon. Tämä arvo voidaan löytää kokeellisesti piirtämällä kartio ja normaalijakauma napakoordinaatistossa ja etsimällä arvo  $\xi$ , joka kattaa tärkeimmän osan normaalijakauman vaikutuksesta.

Tiedämme säteen kulkeman matkan ja haluamme saada selville kartion pohjan pinta-alan. Saamme tämän käyttämällä trigonometriaa ja trigonometrasta yhtäsuuruutta

$$\tan \frac{\theta}{2} = \frac{1 - \cos \theta}{\sin \theta}. \quad (6.22)$$

Tätä hyödyntämällä saamme

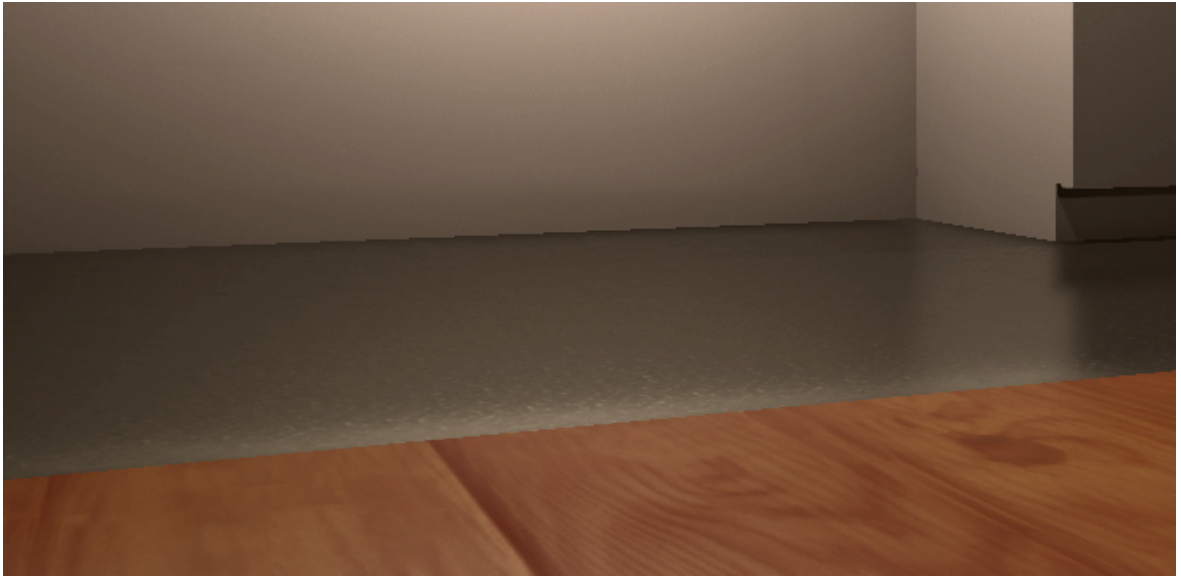
$$\tan \frac{\theta}{2} = \frac{r}{h} \Rightarrow r = \frac{1 - \cos \theta}{\sin \theta} \cdot h, \quad (6.23)$$

missä  $r$  on kartion pohjan säde ja  $h$  on säteen kulkema matka. Tämä kartion koko voidaan muuttaa kuvapuskurin mipmap-ketjun tasoksi näytteistämistä varten. Arvo voidaan kirjoittaa esimerkiksi heijastuspuskurin neljänteen värikanavaan tai omaan puskuriin myöhempää käyttöä varten.

Mipmap-tasojen luominen ja sumentaminen on hyvä tehdä gaussilaisella suotimella, sillä se on mahdollista toteuttaa tehokkaasti. Tavallisesti suodin, jonka leveys on  $W$  ja korkeus  $H$  kuvaelementtiä, vaatii  $W \cdot H$  operaatiota kuvaelementtiä kohden. Gaussilainen suodin on kuitenkin jaettavissa pysty- ja vaakasuuntaiseen vaiheeseen ja on siten toteutettavissa  $W + H$  operaatiolla kuvaelementtiä kohden. Koska gaussilainen suodin ei kuitenkaan ota huomioon syvyysepäjätkuvuutta sumennuksessa, voi se aiheuttaa heijastusten ”vuotamista”. Tästä aiheutuvat virheet ilmenevät lähellä olevien kappaleiden reunan kirkkauden ja värin muutoksena. Virhe ei ole ilmeinen kameran pisyessä paikoillaan, mutta on liikkeessä helpommin havaittavissa. Kuvio 23 havainnollistaa kyseistä virhettä. Paremmen tuloksen saisi käyttämällä bilateraalista sumennusta, joka ottaa huomioon syvyysepäjätkuvuuden painottamalla otettuja näytteitä syvyyden samankaltaisuudella. Se on kuitenkin raskaampi toteuttaa, sillä sitä ei voida jakaa kahteen vaiheeseen.

Lopullinen kuva saadaan, kun yhdistetään alkuperäinen valaistuspuskuri ja heijastuspuskuri. Heijastuspuskuria näytteistetään sumennusta vastaavalta mipmap-tasolta, josta saadaan kuvaelementille saapuva epäsuoran heijastuneen valon määrä. Tätä radianssia pitää painottaa heijastusjakaumafunktiolla, johon voidaan käyttää samaa ympäristön heijastusjakaumafunktion arviota, jota käytettiin ympäristötutkaimille.

Kuviossa 24 nähdään kuvapuskuriheijastusten karheudesta johtuva heijastuksen sumennus eri karheusarvoilla. Lattian materiaali on asetettu metalliseksi, jotta heijastukset tulevat selkeämmin esille. Kuvasta voidaan havaita, kuinka heijastuksen sumennus on sitä suurempi mitä pidemmän matkan heijastussäde on kulkenut. Kuviossa 25 lattian karheus ei ole vakio, vaan tulee karheustekstuurista, jossa arvo vaihtelee huomattavasti.



Kuvio 23. Kuviossa näkyy lähellä olevan puutason heijastuksen vuotaminen kauempana olevalle taustalle. Kuvan kontrastia on lisätty, jotta virhe tulisi selvemmin esille.

### 6.3 Menetelmien yhdistäminen ja suorituskyky

Koska kuvapuskuriheijastustekniikka ei kykene tuottamaan heijastuksia kaikelle geometrialle, on tärkeää, että käytössä on toissijainen tekniikka, joka paikkaa tilanteet, joissa kuvapuskuriheijastukset eivät toimi. Ympäristökartoitus, jossa heijastukset on lokalisoitu käyttäen sijaisgeometriaa, toimii hyvin, jos sijaisgeometria mukailee ympäristöä tarpeeksi tarkasti. Muussa tapauksessa voi olla parempi käyttää syvyyspuskurilla toteutettua lokalisointia.

Ongelmana tekniikoiden käyttämisessä yhdessä on esimerkiksi geometria, joka kuvaa valonlähdettä, kuten lamppua tai kynttilän liekkiä. Jos valonlähde tuottaa analyyttisen peiliheijastuksen, on tässä tilanteessa valosta peiliheijastuva valo mahdollisesti otettu huomioon kahteen kertaan. Ensin analyyttisen valon lähteen muodossa ja sen jälkeen emissiivisen geometrian muodossa kuvapuskuriheijastuksessa tai lokalisoitussa ympäristökartoituksessa. Virhe on selkeämmin havaittava, jos nämä heijastukset eivät osu kohdikkain, esimerkiksi ympäristökartan lokalisoinnin epätarkkuudesta johtuen. Kuviossa 26 analyyttisen valonlähteen ja emissiivisen geometrian peiliheijastus kohdistuvat päällekkäin, jolloin virhe ei ole häiritsevää. Kuviossa 27 sama valon lähde näkyy heijastuneena eri kuvakulmasta. Nyt selvästikään heijastukset eivät kohtaa, jolloin heijastus näkyy kahtena.

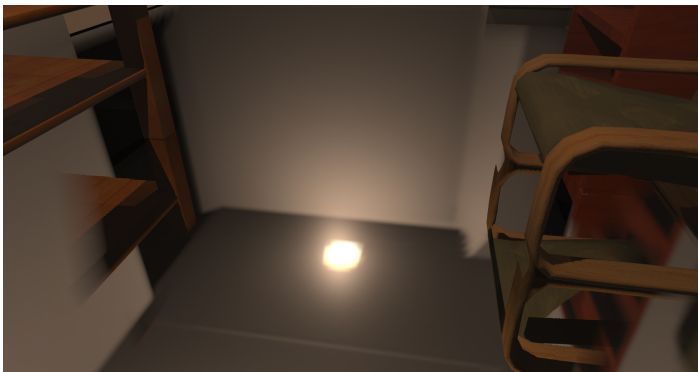




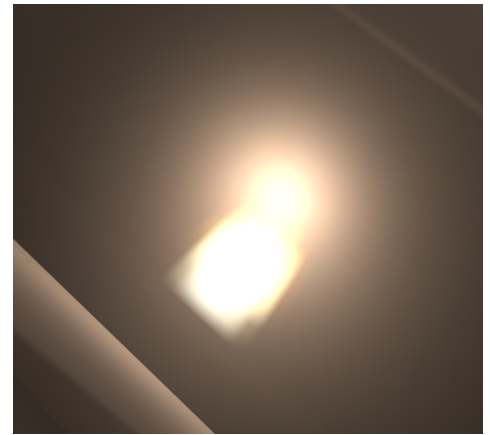
Kuvio 24. Kuviossa fysikaalisten kuvapuskuriheijastusten tulos. Lattian heijastuksissa on havaittavissa, kuinka materiaalin karheus vaikuttaa heijastusten sumeuteen. Lattian karheus arvot ovat  $\sigma = 0.08, 0.24, 0.4$ .



Kuvio 25. Kuviossa lattian karheusarvo ei ole vakio, mistä johtuen heijastuksen sumeus vaihtelee.



Kuvio 26. Kuviossa näkyy kuvapuskuriheijastusten ja lokalisoitujen ympäristötutkaimien yhdistämisen tulos.



Kuvio 27. Kuviossa analyttisen valon ja ympäristötutkaimen aiheuttamat heijastukset eivät täsmää, joten heijastus näkyy kahtena.

Yksi ratkaisu tähän ongelmaan olisi käyttää pelkkää analyttistä peiliheijastusta valonlähteen geometrian mallintamiseen. Tämä on mahdollista, jos emissiivistä geometriaa ei renderöitäisi ympäristökarttoihin ja valaistuspuskuriin ennen kuvapuskuriheijastusten suorittamista. Koska analyttinen valaistus ei yleensä vastaa valongeometriaa tarkasti, ei valon geometria heijastuksissa täsmää niiden haluttua muotoa. Toisaalta, jos valongeometriasta tuleva valaistus huomioidaan eri tekniikoilla useamman kerran, voi valonlähteestä olla havaittavissa useampia peiliheijastuksia, koska eri tekniikat eivät osu kohdikkain.

Ongelma, johon kuvapuskuriheijastukset ja ympäristökartoitus eivät ole sopiva ratkaisu, on tarkat peiliheijastukset. Tarkkoja peiliheijastuksia varten on parempi käyttää tasoheijastuksia. Tasoheijastustekniikan toimiminen vain tasomaisille pinnoille ei ole ongelma, sillä yleensä pinnat, joiden pitäisi heijastaa tarkkoja peiliheijastuksia ovat juuri tasomaisia. Jos pinta ei ole tasomainen, epätarkemmat tekniikat toimivat hyvin, sillä ihminen ei yleensä huomaa heijastusten tarkkuudessa olevia puutteita, koska heijastus on muutenkin pinnan epätasaisuudesta johtuen epätarkka.

Yksi mahdollinen lähestymistapa tekniikoiden yhdistämiseen on käyttää kameraa seuraavaa dynaamista ympäristötutkainta. Tällainen tutkain voitaisiin toteuttaa käyttäen syvyyspuskurilla toteutettua lokalisointia, jolloin heijastukset ovat tarkemmat. Kauempana kamerasta voidaan käyttää sopivasti sijoiteltuja yksinkertaisemmalla sijaisgeometrialla lokalisoituja ympäristötutkaimia. Jos dynaamisen tutkaimen resoluutio on tarpeeksi suuri, voi olla ettei erillistä kuvapuskuriheijastusta tarvita. Koska ympäristötutkain ja kuvapuskuriheijastukset toimivat samoin, kun tutkain on lokalisoitu syvyysesitystä käyttäen, ovat tekniikat identtiset, jos tutkain seuraa kameraa. Näin tekniikalla olisi mahdollista toteuttaa paremmin myös peilipintoja, koska tutkain kaappaa informaatiota joka puolelta kameraa. Ongelmana on, että kuutiotekstuurilla toteutettuna ympäristö on renderöitävä kuusi kertaa. Jos kuutiotekstuurin resoluutio on suuri, tämä on erittäin hidasta.

Taulukossa 6 on listattu tekniikoiden eri vaiheiden vaatima aika kuvaruutua kohden<sup>1</sup>. Tekniikoiden ajankäyttö mitattiin piirtämällä kuvion 25 näkymä useita kertoja resoluutiolla  $1920 \times 1080$ . Taulukon ympäristötutkaimen piirto tarkoittaa tutkaimen ympäristökartan

---

1. Testilaitteisto: Intel Core-i5 6500, neljä ydintä, 8Gt keskusmuistia, Nvidia Geforce GTX 960 -näytönohjain, 4Gt näytönohjainmuistia

Taulukko 6. Taulukko esittää renderöinnin eri vaiheisiin käytetyn ajan kuvaruutua kohden.

Vaihe	Aika/kuvaruutu
G-puskuri	0.710ms
Suora valaistus	2.485ms
Ympäristötutkaimen piirto	2.101ms
Tutkainvalaistus	0.773ms
Säteenseuranta	11.000ms
Mipmap-ketju	2.211ms
Yhdistäminen	0.416ms
Yhteensä	19.696ms

päivittämistä ja mipmap-ketjun luontia käyttäen suodatettua painotettua otantaa. On huomattava, että ensimmäiselle mipmap-tasolle ei suoriteta suodatusta, koska näin säästymme turhalta työltä. Korkeimman mipmap-ketjun tason halutaan juuri oleva tarkka eikä sumennettu. Tutkainvalaistus vastaa näkymän valaisua ympäristötutkaimia käyttäen. Säteenseuranta on kuvapuskuriheijastusten ensimmäinen vaihe, jonka jälkeen säteenseurannan tuloksesta luodaan mipmap-ketju. Yhdistämisvaiheessa suora ja epäsuora valaistus yhdistetään lopulliseksi valaistuspuskuriksi.

Koska säteenseurantavaihe on kuvapuskuriheijastusten hitain vaihe, sen optimointi tai muuten tehostaminen vaikuttaa koko tekniikan nopeuteen huomattavasti. Yksi suoraviivainen tapa on vähentää otettujen askelten määrää. Kun karheus on korkea, piilottaa sen aiheuttama heijastuksen sumennus suurimmat virheet. Tästä syystä askelten määrää voidaan vähentää karheuden kasvaessa laadun juurikaan kärsimättä. Voidaan siis valita pienin ja suurin askelharppaus, joiden väliltä kuvaelementin karheuden mukaan interpoloidaan käytetty harppaus. On myös mahdollista jättää säteenseuranta kokonaan suorittamatta, jos karheus on suurempi kuin jokin valittu yläraja, ja jättää näiden kuvaelementtien heijastukset toissijaisen tekniikan varaan. Tätä on hyödynnetty tutkielman kuvissa, jotka esittävät sumennettuja kuvapuskuriheijastuksia.

Toinen yleisesti käytetty nopeutuskeino on vähentää säteenseurannan määrää suorittamalla

säteenseuranta kuvapuskurille, joka on skaalattu puolikkaan kokoiseksi leveys- ja korkeussuunnassa. Näin tarvittu säteenseuranta kutistuu neljäsosaksi täysikokoiseen puskuriin verrattuna. Lisäksi pienemmästä kuvapuskurista näytteistäminen on tehokkaampaa välimuistin kannalta, koska peräkkäiset näytteet ovat todennäköisesti lähempänä toisiaan muistissa.

Koska näin saatu heijastuspuskuri on pienempi kuin alkuperäinen puskuri, joudutaan se skaalamaan takaisin alkuperäisen puskurin kokoiseksi valaistuspuskuriin yhdistämistä varten. Tämä on järkevintä tehdä bilateriaalisella skaalauksella, joka näytteiden painoarvoa laskiessa ottaa huomioon naapurikuvaelementtien samankaltaisuuden. Bilateraalin skaalaus verrattuna naiiviin bilineaariseen skaalaukseen vähentää virheitä, jotka johtuvat pienemmästä heijastuspuskurista. Silti se ei kuitenkaan pysty luomaan puuttuvaa informaatiota, joten tulos on hieman heikompi, kuin täysikokoisen kuvapuskurin säteenseuranta. Kun säteenseurannan ongelmien piilottamiseksi heijastavien pintojen karheus on tarpeeksi suuri, ei heijastuspuskurin resoluution alentamisesta kuitenkaan tule näkyviä ongelmia. Tutkielman kuvissa ei ole käytetty matalampi resoluutioista heijastuspuskuria.

## 7 Johtopäätökset

Tutkielmassa havaittiin, että reaaliaikaisten fysikaalista valaistusta tukevien heijastusten toteuttaminen vaatii paljon arviointia. Lisäksi luvussa 6 käsitellyissä tekniikoissa huomattiin olevan paljon ongelmakohtia.

Ympäristötutkaimille ongelman tuottaa lokalisoinnissa sijaisgeometrian tarkkuus esittää ympäristön geometriaa. Lisäksi tutkaimien asettelu ympäristöön käsin on työlästä ja algoritmisen asettelun toteuttaminen voi myös tuottaa ongelmia. Tutkielmassa ei erityisesti käsitelty tutkainten interpolointiin liittyviä ongelmia tai niiden ratkaisuja, mutta joustavan ja yleisesti toimivan toteutuksen laatiminen vaatii näidenkin huomioon ottamista. On siis selvää, että tekniikka vaatii paljon työtä, jotta tulokset olisivat uskottavia riippumatta ympäristöstä.

Kuvapuskuriheijastusten suurin ongelma on kuvapuskurin epätäydellinen esitys ympäristöstä. Tästä syystä tekniikka ei kykene esittämään heijastuvaa geometriaa, joka ei ole kameran nähtävissä. Tämä on myös syy, josta johtuen tekniikan käyttäminen ainoana heijastustekniikkana ei toimi uskottavasti. Tarvitaan toissijainen tekniikka, joka paikkaa heijastukset niissä tilanteissa, joissa kuvapuskuritekniikka epäonnistuu. Nämä tilanteet tulee ottaa huomioon säteenseurantavaiheessa, jossa heijastuksissa ilmeneviä virheitä tapahtuu esimerkiksi säteen harppauksen ollessa liian suuri.

Kuvapuskuriheijastuksilla on kuitenkin mahdollista saada erittäinkin uskottavia heijastuksia. Fysikaalinen valaistus on mahdollista toteuttaa arvioimalla heijastuksen karheudesta johtuva sumentuminen esisumentamalla heijastuspuskurin sisältö. Tämä arviointi ei tuota niin suurta virhettä, jotta lopputuloksen uskottavuus kärsisi liikaa. Kun toissijaisena tekniikkana käyttää ympäristötutkaimia, saadaan tekniikat sopimaan yhteen hyvinkin saumattomasti. Tämä tosin vaatii, että tutkaimen lokalisoinnissa käytetty arvio ympäristöstä täsmää todelliseen geometriaan tarpeeksi hyvin. Muuten on käytettävä suurta karheusarvoa, ettei yhteensopivuus näy liian häiritsevästi.

Tarkoilla peilipinnoilla eivät kumpikaan, ympäristötutkaimet tai kuvapuskuriheijastukset, ole parhaimmillaan. Ensinnäkin kuvapuskuritekniikka pettää, jos kameran suuntaa heijastavaa tasoa kohti, koska heijastuksessa tulisi näkyä kameran takana oleva ympäristö. Ympäristö-

tutkaimelle tilanne aiheuttaa ongelmia, johtuen lokalisoinnin tarkkuuden puutteesta. Onkin siis parempi käyttää esimerkiksi tasoheijastuksia, joilla on mahdollista esittää tarkkoja peiliheijastuksia. Jotta kaikki yleisimmät tilanteet saadaan katettua, joudutaankin toteuttamaan useita eri heijastustekniikoita, joista jokainen tekniikka pystyy suoriutumaan vain tietynlaisesta tilanteesta. Näin ollen niiden käyttäminen yhdessä vaatii ylimääräistä työtä ja suunnittelua, että lopputulos olisi kuitenkin pätevä erilaisissa ympäristöissä lisäämättä ympäristön suunnittelijan työtä liikaa.

Reaaliaikaisessa grafiikassa fysikaalisen valaistuksen yleistettyä uskottavuus on kasvanut huomattavasti. Osaltaan reaaliaikaisen epäsuoran valaistuksen toteutukset, kuten tutkielmas-  
sa käsitellyt heijastustekniikat, ovat tuoneet reaaliaikaista tietokonegrafiikkaa lähemmäs elokuvatuotannon fotorealistista laatua. Kuitenkin voidaan todeta, että reaaliaikaisten heijastusten toteutus on vielä ratkaisematon ongelma. Tällä hetkellä käytössä olevat menetelmät vaativat paljon työtä, jotta häiritsevimmät ongelmat saadaan poistettua tai piilotettua. Ennen kuin perinteistä säteenseurantaa pystytään suorittamaan näytönohjaimilla tarpeeksi nopeasti, joudutaan kuitenkin tyytymään erilaisiin säteenseurannan ja rasteroinnin välimuotoihin ja tekniikoiden sekoitukseen.

Tutkielmassa on muutamia rajoitteita, joihin ei ole kiinnitetty huomiota. Tutkielmassa ei esimerkiksi ole otettu huomioon näkyvyysfunktioita, jota tarvitaan uskottavien heijastusten saavuttamiseksi (Kozlowski ja Kautz 2007). Näkyvyyttä pystyttäisiin arvioimaan kuvapuskuritekniikalla, jossa kuvaelementin lähistön syvyysarvoja näytteistämällä voidaan saada arvio sekä diffuusien että peiliheijastuneen epäsuoran valon näkyvyydelle käsiteltävässä pisteessä.

Tutkielmassa ei myöskään käsitelty anisotrooppisia valaistusmalleja, joita tietynlaiset materiaalit tarvitsevat, jotta ne voitaisiin kuvata uskottavasti. Kuvapuskuriheijastuksia ei myöskään ole vertailtu referenssitoteutukseen.

Käsiteltyjen tekniikoiden toteutusta ei ole optimoitu, eikä niitä ole testattu erityisen laajasti erilaisilla ympäristöillä. Näin ollen tarkkaa analyysiä tekniikoiden suorituskyvystä ei voitu suorittaa.

## Lähteet

Bjorke, Kevin. 2004. “Image-Based Lighting”. Luku 19 teoksessa *GPU Gems*. Addison-Wesley Professional. Viitattu 4. helmikuuta 2016. [https://developer.nvidia.com/gpugems/GPUGems/gpugems\\_ch19.html](https://developer.nvidia.com/gpugems/GPUGems/gpugems_ch19.html).

Blinn, James F., ja Martin E. Newell. 1976. “Texture and Reflection in Computer Generated Images”. *Commun. ACM* (New York, NY, USA) 19, numero 10 (lokakuu): 542–547. ISSN: 0001-0782. doi:10.1145/360349.360353.

Colbert, Mark, ja Jaroslav Krivánek. 2007. “GPU-Based Importance Sampling”. Luku 20 teoksessa *GPU Gems 3*. Addison-Wesley Professional. Viitattu 5. helmikuuta 2016. [https://developer.nvidia.com/gpugems/GPUGems3/gpugems3\\_ch20.html](https://developer.nvidia.com/gpugems/GPUGems3/gpugems3_ch20.html).

Cook, Robert L., ja Kenneth E. Torrance. 1981. “A Reflectance Model for Computer Graphics”. *SIGGRAPH Comput. Graph.* (New York, NY, USA) 15, numero 3 (elokuu): 307–316. ISSN: 0097-8930. doi:10.1145/965161.806819.

Estalella, Pau, Ignacio Martin, George Drettakis ja Dani Tost. 2006. “A GPU-driven Algorithm for Accurate Interactive Reflections on Curved Objects”. Teoksessa *Proceedings of the 17th Eurographics Conference on Rendering Techniques*, 313–318. EGSR '06. Nicosia, Cyprus: Eurographics Association. ISBN: 3-905673-35-5. doi:10.2312/EGWR/EGSR06/313-318.

Fuhrmann, Anton L., Robert F. Tobler ja Stefan Maierhofer. 2004. “Real-time Glossy Reflections on Planar Surfaces”. Teoksessa *Proceedings of the 3rd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, 87–91. AFRI-GGRAPH '04. Stellenbosch, South Africa: ACM. ISBN: 1-58113-863-6. doi:10.1145/1029949.1029966.

Glassner, Andrew S. 1995. *Principles of digital image synthesis: Vol 1 & 2*. Toimittanut Brian A. Barsky. Principles of digital image synthesis. Morgan Kaufmann Publishers. ISBN: 1-55860-276-3.



- Greene, N. 1986. “Environment Mapping and Other Applications of World Projections”. *Computer Graphics and Applications, IEEE* 6, numero 11 (marraskuu): 21–29. ISSN: 0272-1716. doi:10.1109/MCG.1986.276658.
- Greger, Gene, Peter Shirley, Philip M. Hubbard ja Donald P. Greenberg. 1998. “The Irradiance Volume”. *IEEE Comput. Graph. Appl.* (Los Alamitos, CA, USA) 18, numero 2 (maaliskuuta): 32–43. ISSN: 0272-1716. doi:10.1109/38.656788.
- Hakura, Ziyad S., John M. Snyder ja Jerome E. Lengyel. 2001. “Parameterized Environment Maps”. Teoksessa *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, 203–208. I3D '01. New York, NY, USA: ACM. ISBN: 1-58113-292-1. doi:10.1145/364338.364402.
- Heidrich, Wolfgang, ja Hans-Peter Seidel. 1998. “View-independent Environment Maps”. Teoksessa *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware*, 39–45. HWWS '98. Lisbon, Portugal: ACM. ISBN: 1-58113-097-X. doi:10.1145/285305.285310.
- Hermanns, Lukas. 2015. “Screen space cone tracing for glossy reflections”. Viitattu 2. maaliskuuta 2016. <http://publica.fraunhofer.de/documents/N-336466.html>.
- Hill, Stephen, Stephen McAuley, Brent Burley, Danny Chan, Luca Fascione, Naty Iwanicki Michałand Hoffman, Wenzel Jakob, David Neubelt, Angelo Pesce ja Matt Pettineo. 2015. “Physically Based Shading in Theory and Practice”. Teoksessa *ACM SIGGRAPH 2015 Courses*, 22:1–22:8. SIGGRAPH '15. Los Angeles, California: ACM. ISBN: 978-1-4503-3634-5. doi:10.1145/2776880.2787670.
- Hill, Stephen, Stephen McAuley, Jonathan Dupuy, Yoshiharu Gotanda, Eric Heitz, Naty Hoffman, Sébastien Lagarde ym. 2014. “Physically Based Shading in Theory and Practice”. Teoksessa *ACM SIGGRAPH 2014 Courses*, 23:1–23:8. SIGGRAPH '14. Vancouver, Canada: ACM. ISBN: 978-1-4503-2962-0. doi:10.1145/2614028.2615431.
- Immel, David S., Michael F. Cohen ja Donald P. Greenberg. 1986. “A Radiosity Method for Non-diffuse Environments”. *SIGGRAPH Comput. Graph.* (New York, NY, USA) 20, numero 4 (elokuu): 133–142. ISSN: 0097-8930. doi:10.1145/15886.15901.

- Kajiya, James T. 1986. "The Rendering Equation". Teoksessa *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, 143–150. SIGGRAPH '86. New York, NY, USA: ACM. ISBN: 0-89791-196-2. doi:10.1145/15922.15902.
- Karis, Brian. 2013. *Real Shading in Unreal Engine 4*. Tekninen raportti. Epic Games. Viitattu 5. helmikuuta 2016. [http://blog.selfshadow.com/publications/s2013-shading-course/karis/s2013\\_pbs\\_epic\\_notes\\_v2.pdf](http://blog.selfshadow.com/publications/s2013-shading-course/karis/s2013_pbs_epic_notes_v2.pdf).
- Knight, Randall Dewey. 2008. *Physics For Scientists And Engineers: A Strategic Approach*. San Francisco: Pearson/Addison Wesley. ISBN: 978 0 321 50780 8.
- Kozlowski, Oscar, ja Jan Kautz. 2007. "Is Accurate Occlusion of Glossy Reflections Necessary?" Teoksessa *Proceedings of the 4th Symposium on Applied Perception in Graphics and Visualization*, 91–98. APGV '07. Tübingen, Germany: ACM. ISBN: 978-1-59593-670-7. doi:10.1145/1272582.1272601.
- Liu, Fang, Meng-Cheng Huang, Xue-Hui Liu ja En-Hua Wu. 2009. "Bucket Depth Peeling". Teoksessa *SIGGRAPH 2009: Talks*, 50:1–50:1. SIGGRAPH '09. New Orleans, Louisiana: ACM. ISBN: 978-1-60558-834-6. doi:10.1145/1597990.1598040.
- Lvovsky, Alexander I. 2013. "Fresnel Equations". Luku 63 teoksessa *Encyclopedia of Optical Engineering*, 1–6. Helmikuu. ISBN: 0-8247-0940-3, viitattu 12. joulukuuta 2015. <http://iqst.ca/quantech/pubs/2013/fresnel-ee.pdf>.
- McAuley, Stephen, Stephen Hill, Naty Hoffman, Yoshiharu Gotanda, Brian Smits, Brent Burley ja Adam Martinez. 2012. "Practical Physically-based Shading in Film and Game Production". Teoksessa *ACM SIGGRAPH 2012 Courses*, 10:1–10:7. SIGGRAPH '12. Los Angeles, California: ACM. ISBN: 978-1-4503-1678-1. doi:10.1145/2343483.2343493.
- McAuley, Stephen, Stephen Hill, Adam Martinez, Ryusuke Villemin, Matt Pettineo, Dimitar Lazarov, David Neubelt ym. 2013. "Physically Based Shading in Theory and Practice". Teoksessa *ACM SIGGRAPH 2013 Courses*, 22:1–22:8. SIGGRAPH '13. Anaheim, California: ACM. ISBN: 978-1-4503-2339-0. doi:10.1145/2504435.2504457.

- McGuire, Morgan, ja Michael Mara. 2014. “Efficient GPU Screen-Space Ray Tracing”. *Journal of Computer Graphics Techniques (JCGT)* 3, numero 4 (joulukuu): 73–85. ISSN: 2331-7418. <http://jcgt.org/published/0003/04/04/>.
- Meyer, Alexandre, ja Céline Loscos. 2003. “Real-time Reflection on Moving Vehicles in Urban Environments”. Teoksessa *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, 32–40. VRST '03. Osaka, Japan: ACM. ISBN: 1-58113-569-6. doi:10.1145/1008653.1008662.
- Nagy, Zoltan, ja Reinhard Klein. 2003. “Depth-peeling for texture-based volume rendering”. Teoksessa *Computer Graphics and Applications, 2003. Proceedings. 11th Pacific Conference on*, 429–433. Lokakuu. doi:10.1109/PCCGA.2003.1238289.
- Oren, Michael, ja Shree K. Nayar. 1994. “Generalization of Lambert’s Reflectance Model”. Teoksessa *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, 239–246. SIGGRAPH '94. New York, NY, USA: ACM. ISBN: 0-89791-667-0. doi:10.1145/192161.192213.
- Patow, Gustavo A. 1995. *Accurate Reflections Through a Z-Buffered Environment Map*. Viitattu 26. helmikuuta 2016. <http://ima.udg.edu/~dagush/papers/sccc95.pdf>.
- Popescu, Voicu, Chunhui Mei, Jordan Dauble ja Elisha Sacks. 2006. “Reflected-Scene Impostors for Realistic Reflections at Interactive Rates”. *Computer Graphics Forum* 25 (3): 313–322. ISSN: 1467-8659. doi:10.1111/j.1467-8659.2006.00950.x.
- Ramamoorthi, Ravi, ja Pat Hanrahan. 2001. “An Efficient Representation for Irradiance Environment Maps”. Teoksessa *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, 497–500. SIGGRAPH '01. New York, NY, USA: ACM. ISBN: 1-58113-374-X. doi:10.1145/383259.383317.
- . 2002. “Frequency Space Environment Map Rendering”. *ACM Trans. Graph.* (New York, NY, USA) 21, numero 3 (heinäkuu): 517–526. ISSN: 0730-0301. doi:10.1145/566654.566611.

- Roger, David, ja Nicolas Holzschuch. 2006. “Accurate Specular Reflections in Real-Time”. *Computer Graphics Forum* 25 (3): 293–302. ISSN: 1467-8659. doi:10.1111/j.1467-8659.2006.00948.x.
- Räbinä, Jukka. 2006. “Valon sironnan simulointi Monte Carlo -menetelmällä”. Tutkielma, Jyväskylän yliopisto.
- Schlick, Christophe. 1994. “An Inexpensive BRDF Model for Physically-based Rendering”. *Computer Graphics Forum* 13 (3): 233–246. ISSN: 1467-8659. doi:10.1111/1467-8659.1330233.
- Sébastien, Lagarde, ja Antoine Zanuttini. 2012. “Local Image-based Lighting with Parallax-corrected Cubemaps”. Teoksessa *ACM SIGGRAPH 2012 Talks*, 36:1–36:1. SIGGRAPH ’12. Los Angeles, California: ACM. ISBN: 978-1-4503-1683-5. doi:10.1145/2343045.2343094.
- Shirley, Peter, R. Keith Morley, Peter-Pike Sloan ja Chris Wyman. 2012. “Basics of Physically-based Rendering”. Teoksessa *SIGGRAPH Asia 2012 Courses*, 2:1–2:11. SA ’12. Singapore, Singapore: ACM. ISBN: 978-1-4503-1913-3. doi:10.1145/2407783.2407785.
- Smith, W.A.P., ja E.R. Hancock. 2009. “Specular and diffuse reflectance in microfacet models”. Teoksessa *Image Processing (ICIP), 2009 16th IEEE International Conference on*, 3781–3784. Marraskuu. doi:10.1109/ICIP.2009.5414345.
- Szirmay-Kalos, László, Barnabás Aszódi, István Lazányi ja Mátyás Premecz. 2005. *Approximate ray-tracing on the GPU with distance impostors*. doi:10.1111/j.1467-8659.2005.0m894.x.
- Szirmay-Kalos, László, Tamás Umenhoffer, Gustavo Patow, László Szécsi ja Mateu Sbert. 2009. “Specular Effects on the GPU: State of the Art”. *Computer Graphics Forum* 28 (6): 1586–1617. ISSN: 1467-8659. doi:10.1111/j.1467-8659.2009.01350.x.
- Tiago Sousa, Nickolay Kasyan, ja Nicolas Schulz. 2012. “CryENGINE 3: Three Years of Work in Review”. Luku 4 teoksessa *GPU PRO 3: Advanced Rendering Techniques*, 147–148. CRC Press. ISBN: 9781439887943.

- Torrance, K. E., ja E. M. Sparrow. 1967. "Theory for off-specular reflection from roughened surfaces". *Journal of the Optical Society of America* 57 (syyskuu): 1105–1114. Viitattu 14. marraskuuta 2015. <http://www.graphics.cornell.edu/~westin/pubs/TorranceSparrowJOSA1967.pdf>.
- Trowbridge, T. S., ja K. P. Reitz. 1975. "Average irregularity representation of a rough surface for ray reflection". *Journal of the Optical Society of America* 65, numero 5 (toukokuu): 531–536. doi:10.1364/JOSA.65.000531.
- Valient, Michal. 2014. *Reflections and volumetrics of Killzone Shadow Fall*. Viitattu 28. maaliskuuta 2016. <http://advances.realtimerendering.com/s2014/index.html>.
- Walter, Bruce, Stephen R. Marschner, Hongsong Li ja Kenneth E. Torrance. 2007. "Microfacet Models for Refraction Through Rough Surfaces". Teoksessa *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, 195–206. EGSR'07. Grenoble, France: Eurographics Association. ISBN: 978-3-905673-52-4. doi:10.2312/EGWR/EGSR07/195-206.
- Veach, Eric. 1997. "Robust monte carlo methods for light transport simulation". Tohtorinväitöskirja, Stanford University. Viitattu 4. joulukuuta 2015. [http://graphics.stanford.edu/papers/veach\\_thesis/thesis.pdf](http://graphics.stanford.edu/papers/veach_thesis/thesis.pdf).
- Widmer, S., D. Paják, A. Schulz, K. Pulli, J. Kautz, M. Goesele ja D. Luebke. 2015. "An Adaptive Acceleration Structure for Screen-space Ray Tracing". Teoksessa *Proceedings of the 7th Conference on High-Performance Graphics*, 67–76. HPG '15. Los Angeles, California: ACM. ISBN: 978-1-4503-3707-6. doi:10.1145/2790060.2790069.
- Yu, Jingyi, Jason Yang ja Leonard McMillan. 2005. "Real-time reflection mapping with parallax." Teoksessa *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games*, 133–138. I3D '05. Washington, District of Columbia: ACM. ISBN: 1-59593-013-2. doi:10.1145/1053427.1053449.
- Zhang, Cheng, Hsien hsi Hsieh ja Han wei Shen. 2008. *Real-time Reflections on Curved Objects Using Layered Depth Textures*. Tekninen raportti. Viitattu 28. tammikuuta 2016. <ftp://ftp.cse.ohio-state.edu/pub/tech-report/2007/TR55.pdf>.