

**Valtteri Koskivuori**

**Johdatus säteenseurannan soveltamiseen fotorealististen  
kuvien tuottamiseksi**

Tietotekniikan kandidaatintutkielma

31. joulukuuta 2020

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

**Tekijä:** Valtteri Koskivuori

**Yhteystiedot:** vakoskiv@student.jyu.fi

**Ohjaajat:** Sanna Juutinen, Tytti Saksa ja Timo Tiihonen

**Työn nimi:** Johdatus säteenseurannan soveltamiseen fotorealististen kuvien tuottamiseksi

**Title in English:** Introduction to ray tracing methods for photorealistic rendering

**Työ:** Kandidaatintutkielma

**Opintosuunta:** Tietotekniikka

**Sivumäärä:** 23+0

**Tiivistelmä:** Tutkielmassa käsitellään miten tietokonegrafiikka ja säteenseuranta toimii, sekä miten todellisuuden valoilmiöitä kyetään mallintamaan säteenseurannalla. Tutkielmassa keskitytään nimenomaan ei-reaaliaikaisiin menetelmiin, jotka mallintavat todellisuuden ilmiöitä mahdollisimman tarkasti. Tutkielma pyrkii toimimaan johdantona perusteisiin, joten siinä käsitellään ainoastaan perusteet sekä muutama menetelmä, joilla näiden perusmenetelmien toimivuutta voidaan parantaa.

**Avainsanat:** L<sup>A</sup>T<sub>E</sub>X, gradu3, pro gradu -tutkielmat, kandidaatintutkielmat, käyttöohje

**Abstract:** This thesis examines how computer graphics and raytracing works and how different principles of the real world can be adapted to be simulated using it. We will concentrate specifically in offline rendering methods that model the real world as accurately as possible. The thesis aims to serve as a starting point for further research in computer graphics. Therefore, we will only cover the basics and a few methods for improving upon performance and accuracy.

**Keywords:** L<sup>A</sup>T<sub>E</sub>X, gradu3, Master's Theses, Bachelor's Theses, user's guide

## Termiluettelo

Fotorealismi	Keinotekoisesti luotu kuva, joka näyttää aivan kameralla otetulta valokuvalta.
Pikseli	Digitaalisen valokuvan elementti, joka näyttää värejä.
Polku	Useammasta valonsäteestä muodostettu polku, joka kimpoaa satunnaisesti ympäristössä, keräten valoinformaatiota.
Renderöinti	Kuvan luominen tietokoneella annetun mallin perusteella.
Säde	Simuloitu valonsäde, jolla on lähtöpiste ja suunta.
Säteenseuranta	Menetelmä, joka tuottaa valokuvan seuraamalla valonsäteitä ympäristössä (Whitted 2005).
Tietokonegrafiikka	Tietotekniikan ala, joka tutkii visuaalisen sisällön luomista ja esittämistä tietokoneella.
Kiihdytysrakenne	Tietorakenne, joka nopeuttaa säteiden ja kappaleiden välisten leikkauspisteiden löytämistä.
Kolmioverkko	Yleinen tapa mallintaa mitä tahansa kolmiulotteista muotoa.
Polygoni	Kolmiulotteinen kolmio, joka koostuu kolmesta kärkipisteestä.
Transformaatio	Matemaattinen operaatio, joka toteuttaa halutun muutoksen kappaleelle.
Yksikkövektori	Vektori, jonka pituus on 1.

## **Kuviot**

Kuvio 1. Esimerkkikuva renderöity C-ray -renderöntiohjelmalla. Kuvan 3D-mallit on tehnyt Sean Graham (CC-BY) .....	1
Kuvio 2. Lasimateriaali testiympäristössä. ....	12
Kuvio 3. Mattamateriaali testiympäristössä. ....	13
Kuvio 4. Muovimateriaali testiympäristössä. ....	13
Kuvio 5. Metallimateriaali testiympäristössä.....	14

# Sisältö

1	JOHDANTO .....	1
2	TIETOKONEGRAFIIKAN PERUSTEET .....	3
2.1	Digitaalisten valokuvien anatomia .....	3
2.2	Vektoriavaruudet .....	3
2.2.1	Euklidinen avaruus .....	4
2.2.2	Vektorit .....	4
2.2.3	Matriisit .....	4
2.2.4	Lokaalit ja globaalit koordinaatistot .....	5
2.3	Kolmiulotteisen maailman muotojen mallintaminen .....	5
3	SÄTEENSEURANNAN PERUSTEET .....	6
3.1	Kamera ja säteet .....	6
3.2	Kiihdytsrakenteet .....	7
4	MONTE CARLO JA PATH TRACING .....	9
4.1	Epäsuora valo .....	9
5	MATERIAALIEN MALLINTAMINEN .....	11
5.1	Lasi .....	12
5.2	Mattapinta .....	13
5.3	Muovi .....	13
5.4	Metalli .....	14
5.5	Muut pinnoille yhteiset ominaisuudet .....	14
6	LISÄKIRJALLISUUTTA .....	16
7	YHTEENVETO .....	17
	LÄHTEET .....	18

# 1 Johdanto



Kuvio 1. Esimerkkikuva renderöity C-ray -renderöintiohjelmalla. Kuvan 3D-mallit on tehnyt Sean Graham (CC-BY)

Miten tietokoneella saadaan aikaan valokuvia, jotka näyttävät aivan kameralla otetuilta? Miten valon ominaisuuksia kyetään mallintamaan tietokoneella, ja miten näitä menetelmiä sovelletaan renderöintiohjelmistoissa? Tässä tutkielmassa pyritään vastaamaan näihin kysymyksiin, ja rakentamaan lukijalle hyvä lähtökohta alkaa tutkimaan käsiteltyjä aiheita syvällisemmin.

Tietokonegrafiikka on tietotekniikan tutkimuksen alana valtava, ja jopa tutkielmassa käsiteltävä osajoukko, eli säteenseuranta, kattaa vuosikymmenien tutkimuksen. Tässä tutkielmassa käsitellään muutamia suhteellisen yksinkertaisia menetelmiä, joilla ratkaistaan monia säteenseurannan pohjimmaisia ongelmia ja päästään suhteellisen lähelle fotorealismia. Kuten monissa muissa elämän osa-alueissa, tässäkin pätee vähenevien tuottojen laki. Valtaosa fotorealismiin vaadittavista menetelmistä voidaan saavuttaa yllättävän yksinkertaisilla menetelmillä, mutta viimeiset yksityiskohdat vaativat vuosien opiskelun ja tutkimustyön.

Tutkielmassa käytettävät havainnollistavat valokuvat on luotu kirjoittajan luomalla C-ray-renderöntiiohjelmalla, jota tämän tutkielman kirjoittaja on hiljalleen parannellut vuodesta 2015 alkaen. Ohjelman avoin lähdekoodi on saatavissa MIT-lisenssillä osoitteesta <https://github.com/vkoskiv/c-ray>.

## 2 Tietokonegrafiikan perusteet

Tietokonegrafiikan käsittelyyn tarvitaan aluksi pohjatietoa visuaalisen sisällön luomisesta ja esittämisestä, sekä aiheelle olennaisesta matematiikasta. Käsittelemme väriteoriaa sekä lineaarialgebran perusoperaatioita, joita hyödynnetään säteenseurannassa. Käsiteltäväksi on nimettiin valittu etenkin matemaattisella puolella vain tarvittavat operaatiot, ja todistukset on jätetty kokonaan pois. Lukija voi perehtyä lisää tarvittaviin lineaarialgebran perusteisiin ja tietokonegrafiikan perusteisiin esim. kirjan (McGuire 2018) pohjalta.

### 2.1 Digitaalisten valokuvien anatomia

Kaikki tietokoneen ruudulla näkyvä sisältö koostuu kuvaelementeistä eli pikseleistä. Tietokoneen muistissa näitä väriarvoja voidaan esittää lukuisilla eri tavoilla, mutta yleisimmin käytetään *32-bittistä* väriavaruutta. Tässä väriavaruudessa on 8 bittiä jokaiselle kolmesta pääväristä, sekä lisäksi 8 bittiä kuvaamaan läpinäkyvyyttä. Jokainen näistä värikanavista pystyy siis näyttämään  $2^8$  eli 256 eri kirkkauden tasoa (0 – 255). Fyysisesti tietokoneen ruudun kuvaelementti koostuu yleisimmin taas kolmesta pääväristä eli punaisesta, vihreästä sekä sinisestä. Näiden kolmen päävärin valoisuutta säätämällä tämä pikseli kykenee siis näyttämään  $256^3$  eri värejä. Kuvan tarkkuus esitetään yleensä pikseleissä muotoa *leveys* × *korkeus*, esim.  $1280 \times 800$  eli 1280 pikseliä leveä, ja 800 pikseliä korkea.

### 2.2 VektoriavaruuDET

Kaikki moniulotteista avaruutta matemaattisesti käsittelevä tutkimus käyttää lineaarista algebraa pohjanaan. Tarvitaan siis jonkinlainen korkean tason käsitys tällaisista perusoperaatioista. Tämän tutkielman ymmärtämiseen riittää yksinkertaiset vektori- ja matriisioperaatiot, joita käsitellään tässä luvussa.



### 2.2.1 Euklidinen avaruus

Euklidisessa avaruudessa ortonormaalit kantavektorit  $i, j$  ja  $k$  virittävät yleisimmin käytössä olevan kolmiulotteisen avaruuden koordinaatiston. Perinteisemmin voidaan näiden kantavektorien suuntaisia akseleita kutsua nimillä  $x, y$  ja  $z$ . Tällainen koordinaatisto sopii parhaiten todellisuuden muotojen mallintamiseen. Todellisuuden mittayksiköt voidaan sovittaa tähän järjestelmään helpottamaan ympäristöjen mallintamista todenmukaisessa skaalassa. Voidaan sopia, että yksi abstrakti mittayksikkö tässä koordinaatistossa vastaa yhtä metriä mallinnettavassa ympäristössä.

### 2.2.2 Vektorit

Matemaattisesti vektorille löytyy eri tieteenaloilta monia määritelmiä, mutta tietokonegraafikassa käsitellään vektoreita yksinkertaisesti pisteinä kolmessa ulottuvuudessa, eli matemaattisemmin  $(x, y, z)$ . Jaetaan nämä vektorit vielä kahteen eri kategoriaan: Suuntavektorit ja paikkavektorit. Käytännössä ne ovat identtisiä rakenteeltaan, mutta ne tulkitaan eri tavalla. Paikkavektori tulkitaan absoluuttisena sijaintina avaruudessa, kun taas suuntavektori tulkitaan deltana eli muutoksena jostakin muusta, määrittelemättömästä sijainnista. Suuntavektorit ovat yleensä yksikkövektoreita (engl. unit vector), eli suunnasta huolimatta niiden pituus on aina 1.0.

### 2.2.3 Matriisit

Matriisit ovat olennaisia tietokonegraafikassa. Tässä tutkielmassa ei käsitellä todistuksia eri matriisioperaatioille, vaan keskitytään hyödyntämään niitä. Matriisioperaatioita käytetään kappaleiden käsittelyyn avaruudessa. Tällaista käsittelyä kutsutaan *transformaatioksi* (Pharr, Jakob ja Humphreys 2016). Erilaisia transformaatioita on valtavasti, mutta tässä tutkielmassa käsitellään kolmea yleisintä tyyppiä; *Skaalaus* eli kappaleen koon muuttaminen, *translaatio* eli kappaleen liikuttelu avaruudessa, sekä *rotaatio* eli kappaleen asennon pyörittely sen keskipisteen ympäri. Halutut transformaatiomatriisit kerrotaan yhteen yhdeksi komposiittimatriisiksi, jolla voidaan vaihtaa globaalista koordinaatistosta kappaleen paikalliseen koordinaatistoon. Käydään tarkemmin käytännön toteutusta seuraavassa kappaleessa. Kaik-

ki virtuaalisten kappaleiden liikuttelu, pyörittely ja venytys tapahtuu näitä transformaatioita käyttäen. On olennaista, että renderöintimoottori tukee näitä operaatioita, sillä ne mahdollistavat kappaleiden sijoittelun lopullista kuvaa varten (Pharr, Jakob ja Humphreys 2016).

#### **2.2.4 Lokaalit ja globaalit koordinaatistot**

Monesti tietokonegrafiikassa halutaan käsitellä useita koordinaatistoja samaan aikaan, ja siirtyä vaivattomasti niiden välillä. Esimerkiksi maailman koordinaatiston lisäksi jokaisella näkymässä esiintyvällä kappaleella voi olla oma, paikallinen koordinaatisto. Kappaleella on siis oma origo, jonka suhteessa pinnan muodot kuvaillaan.

### **2.3 Kolmiulotteisen maailman muotojen mallintaminen**

Todellisuudessa esiintyvien kappaleiden muotojen mallintaminen tietokoneella voidaan toteuttaa karkeasti jaoteltuna kahdella eri tavalla: suoraan pinnan muotoa kuvailevalla matemaattisella kaavalla, tai hyödyntämällä primitiivejä, esimerkiksi kolmioita (McGuire 2018). Todentuntuisiin valokuviiin tarvitaan yleisluontoinen keino mallintaa kaikenlaisia maailmassa esiintyviä muotoja. Yleisimmin tämä toteutetaan juurikin kolmiulotteisilla kolmioilla, eli polygoneilla (McGuire 2018). Tällainen kolmio koostuu kolmesta kärkipisteestä, normaali-vektoreista sekä mahdollisesti muista pinnan ominaisuuksia kuvailevista lisätiedoista. Kolmen kärkipisteen väliin jäävä alue on kolmion pinta. Näitä kolmioita yhteen sitomalla kyettään mallintamaan mikä tahansa euklidisessa avaruudessa oleva muoto. Tällainen approksimaatio lähestyy todellista muotoa ja siten virhe pienenee, kun kolmioiden määrää lisätään (McGuire 2018). Käytännössä tällaiset 3D-mallit koostuu listasta pisteitä (engl. vertices), listasta pinnan normaaleita (engl. surface normals) sekä listasta pintoja eli kolmioita. Renderöintiohjelmiston tulkki lukee nämä listat tiedostosta ja prosessoi niistä sopivan kuvauksen välimuistiin nopeaa lukemista varten.

### 3 Säteenseurannan perusteet

Kaikki säteenseurannan periaatteet ja menetelmät pohjautuvat suoraan todellisiin valon ominaisuuksiin. Tässä tutkielmassa jätetään fysiikan lait valon aaltoilmiöistä ja muista muuttujista käsittelemättä, joten valonsäteitä voidaan approksimoida suoraan geometrisesti vektoreilla. Todellisuudessa valoa vapautuu ympäristöön valonlähteistä, se kimpoaa eri tavoilla kaikilta näkyvissä olevilta pinnoilta. Pieni osa näistä 'valonsäteistä' kimpoaa lopulta tarkkailijan silmään tai kameran linssiin. Näitä säteiden kimpoamisia, heijastumisia ja muita ilmiöitä tapahtuu niin valtavasti, että niiden mallintaminen tietokoneella olisi käytännössä mahdotonta. Säteenseurannassa tämä paradigma käännetään päinvastoin. Säteitä seurataan kameran ulospäin (Pharr, Jakob ja Humphreys 2016). Nämä säteet osuvat pintoihin, tarkkailevat pinnan ominaisuuksia ja sitten tarvittaessa kimpoavat pinnasta muualle ympäristöön tarkastelemaan valaistusta ja muita ominaisuuksia. Tällä tavoin mallinnetaan ainoastaan valonsäteet, joilla on mahdollisuus lisätä jotakin informaatiota lopulliseen kuvaan. On muistettava, että kameran linssin pinta-ala tai ihmisen silmän linssin pinta-ala on todella pieni - eli se todennäköisyys, että jokin ympäristössä kimpoileva valonsäde osuu siihen, on myös äärimmäisen pieni. Siksi tehdään suunnan kääntäminen, vaikka se saattaakin aluksi vaikuttaa epäintuitiiviselta.

#### 3.1 Kamera ja säteet

Säteenseurannassa käytettävät virtuaaliset kamerat kykenevät mallintamaan hyvinkin tarkasti oikeissa kameroissa tapahtuvaa optiikkaa (Pharr, Jakob ja Humphreys 2016). Voidaan siis mallintaa suoraan oikeista fyysisistä kameroista löytyvää optiikkaa ja ohjata säteitä tällaisten virtuaalisten linssien läpi. Lähtökohtana voidaan kuitenkin pitää yksinkertaisempaa linssitöntä neulanreikä (engl. pinhole) -kameraa, jossa kameran aukko on äärettömän pieni. Kaikkien valonsäteiden lähtöpiste on tämä aukon piste, ja säteet ohjataan kohti tarkasteltavaa ympäristöä. Tällainen kamera mallintaa siis antiikinaikaista camera obscuraa. Tällainen äärettömän pieni aukko on luonnollisesti mahdotonta saavuttaa fyysisessä kamerassa, sillä se päästäisi äärettömän vähän valoa filmille tai sensorille. Visuaalisesti tällaisen 'mahdotto-man' kameran vaikutus lopulliseen kuvaan on äärettömän syvä tarkkuuskenttä (engl. depth-

of-field). Kaikki kuvassa näkyvä on siis täysin tarkka syvyydestä huolimatta (Pharr, Jakob ja Humphreys 2016).

Valonsädettä mallinnetaan suunta- ja paikkavektorin yhdistelmällä. Valonsäteiden ja erilaisien pintojen välisten leikkauspisteiden löytämiseen on monia kaavoja, joiden käsittely on esitetty erinomaisesti esimerkiksi teoksessa McGuire 2018.

### 3.2 Kiihdytysrakenteet

Todellisuuden muotojen kuvailemiseen tarkasti tarvitaan malleja, joiden kolmioiden määrä nousee satoihin tuhansiin tai jopa miljooniin (Pharr, Jakob ja Humphreys 2016). Naiivisti toteutettuna säteenseuranta-algoritmit eivät kykene käsittelemään näin suurta määrää kolmioita (Pharr, Jakob ja Humphreys 2016). Säteen ja 3D-mallin leikkauspisteitä hakeva algoritmi joutuu käymään läpi jokaisen näistä tuhansista kolmioista löytääkseen oikean kolmion, ja sen jälkeen oikean leikkauspisteen sen kolmion sisältä. Tämä joudutaan tekemään jokaiselle säteelle, joita jopa suhteellisen pienessä kuvassa, esimerkiksi  $1280 \times 800$  pikseliä, on yli miljoona. Tässä tapahtuu turhaa laskemista, sillä valtaosa kolmioista on todennäköisesti kaukana käsiteltävän säteen vaikutusalueesta, eikä niillä ole mitään mahdollisuutta tulla säteen eteen (Pharr, Jakob ja Humphreys 2016). Tarvitaan jokin menetelmä näiden turhien laskujen minimoimiseksi.

Kiihdytysrakenteet (engl. acceleration structure) on säteenseurannan kontekstissa binääripuu, joka rakenteellaan nopeuttaa valonsäteiden ja kolmioiden välisien leikkauksien löytämistä. Erilaisia kiihdytysrakenteita on monia, mutta otetaan tarkasteltavaksi yksinkertainen esimerkki: k-d-puut (engl. k-d tree) (Bentley 1975). Bentleyyn esittämän binääripuurakenteen nimi tulee sanoista k-dimensional tree, eli suomeksi k-ulotteinen puu. Tällaisen tietorakenteen rakentaminen koostuu kolmioverkon jakamisesta pienempiin osiin ja näiden pienempien osien sijoittamisesta hierarkkisesti binääripuun eri tasoille. Yleisissä toteutuksissa algoritmi tarkastelee kolmioverkon eri akselien pituuksia ja valitsee niistä ensin pisimmän. Sitten verkko halkaistaan keskeltä tällä pisimmällä akselilla, ja eri puolet tallennetaan tämänhetkisen solmun vasemmalle ja oikealle puolelle. Tätä jatketaan rekursiivisesti, kunnes jokin heuristiikka päättää jakamisen. Yleinen heuristiikka tällaisen jakamisen päättämiseksi on MacDonald ja

Booth 1989 esittämä kolmiojoukon ympäröivän laatikon (engl. bounding box) pinta-alaa tarkasteleva menetelmä, (engl. surface area heuristic), joka laskee jakamisen kannattavuuden. Tämän menetelmän esittämänä tätä jakamista jatketaan, kunnes jakamisen 'hinta' on liian suuri. Tämänhetkisen solmun hinta lasketaan kertomalla sen sisältämien alkioiden määrä sen laatikon pinta-alalla. Tätä hintaa verrataan lapsisolmujen hintojen summaan, ja jos tämä summa ylittää tämänhetkisen solmun summan, jakaminen lopetetaan.

Kiihdytysrakenteita voidaan hyödyntää myös korkeamman tason leikkausten löytämiseen (Pharr, Jakob ja Humphreys 2016). Jos näkymässä on esimerkiksi tuhansia malleja, joista jokaisella on tuhansia kolmioita, voidaan valmistella jokaisen mallin oman k-d-puun lisäksi korkeamman tason hierarkia joka käsittää jokaisen mallin omana yksikkönään. Osumaa haakeva algoritmi siis käy ensin läpi korkeamman tason rakenteen löytääkseen oikean mallin, ja vasta sitten alkaa etsimään oikeaa primitiiviä tämän mallin sisältä. Yleinen korkeamman tason hierarkian kuvaava rakenne on Wald 2007 esittelemä rajaava volyyminhierarkia eli BVH (engl. bounding volume hierarchy). Tällaisessa hierarkiassa jokaisen mallin tilavuus sijoitetaan hierarkkisesti jonkun suuremman kokonaisuuden tilavuuteen. Tällä tavalla saadaan nopeasti selvitettyä karkeasti tieto, mistä oikea primitiivi voisi käsiteltävälle säteelle löytyä.

## 4 Monte Carlo ja Path tracing

Monia todellisuuden valoilmioita on mahdotonta simuloida analyyttisillä keinoilla. Tarkastellaan kuuluisaa renderointiyhtälöä (Kajiya 1986):

$$L_o(\mathbf{x}, \omega_o, \lambda, t) = L_e(\mathbf{x}, \omega_o, \lambda, t) + \int_{\Omega} f_x(\mathbf{x}, \omega_i, \omega_o, \lambda, t) L_i(\mathbf{x}, \omega_i, \lambda, t) (\omega_i \cdot \mathbf{n}) d\omega_i$$

Jossa  $L_o(x, \omega_o, \lambda, t)$  on pinnasta heijastuvan valon säteily aallonpituudella  $\lambda$  suuntaan  $\omega_o$ , ajassa  $t$ , sijainnista  $x$ .  $L_e(x, \omega_o, \lambda, t)$  on valonlähteen säteily,  $\int_{\omega} \dots \omega_i$  on integraali  $\Omega$ :n yli,  $\Omega$  on pisteen  $n$  ympärille keskitetty yksikköpuolipallo (engl. unit hemisphere).  $f_r(x, \omega_i, \omega_o, \lambda, t)$  on kaksisuuntainen heijastusjakaumafunktio (engl. bidirectional reflectance distribution function), joka kertoo suunnasta  $\omega_i$  suuntaan  $\omega_o$  heijastuneen valon suhteen, sijainnissa  $x$ , ajassa  $t$ .  $\omega_i$  on saapuvan valon suunnan negaatio,  $L_i()$  on aallonpituuden  $\lambda$  spektrisäteily joka tulee kohti pinnan pistettä  $x$ , suunnasta  $\omega_i$ , ajassa  $t$ .  $n$  on pinnan normalivektori pisteessä  $x$ , ja  $\omega_i \cdot n$  on pinnasta heijastuvaa valoa heikentävä termi, yleensä approksimoitu muotoon  $\cos \theta_i$ , jossa  $\theta_i$  on saapuvan valonsäteen ja pinnan normaalivektorin välinen kulma (Pharr, Jakob ja Humphreys 2016). Yhtälössä olevan integraalitermin ratkaisu valovoimaisuuden mittaamiseen äärettömän monesta suunnasta on mahdotonta, joten on keksittävä vaihtoehtoinen ratkaisu. Tähän voi soveltaa numeerista Monte Carlo -menetelmää. Tämä menetelmä evaluoi integraalia ottamalla näytteitä satunnaisesti eri pisteissä, ja ottamalla sitten näytteistä keskiarvon päästäkseen lähelle ratkaisua. Tässä ratkaisun karkeassa approksimaatiossa on luonnollisesti paljon heittoa, joten näitä satunnaisnäytteitä tarvitaan monta. Näytteitä lisäämällä päästään asteittain lähemmäksi todellista tulosta.

### 4.1 Epäsuora valo

Todellisuudessa valo kimpoaa ympäristön pinnoista monta kertaa ennen valotehon ehtymistä. Säteenseurantaa voidaan siis jalostaa ns. polunseurannaksi (engl. path tracing), jossa säteen annetaan kimmota useita kertoja, ja se 'kerää' valoa ympäristöstä. Tällainen lähestymistapa mahdollistaa epäsuoran valaistuksen mallintamisen. Huomataan, että monesti polunseurantaa kuvaillaan myös säteenseuranta -termillä, vaikka niissä onkin käytännössä eroavaisuuksia (Pharr, Jakob ja Humphreys 2016).

Kuvitellaan huone, jossa on pöytä. Pöydän yläpuolella on lamppu, ja huone on muuten täysin pimeä. Suoran näköyhteyden puutteesta huolimatta pöydän alla on hämärää, mutta ei täysin pimeää. Voidaan siis päätellä, että jopa näennäisesti valoa heijastamattomat pinnat heijastaa valoa jonkin verran. Eihän mitään pintaa voitaisikaan tarkkailla, ellei siitä heijastu valoa tarkastelijan silmään. Kimmonneiden säteiden suunnan valitseminen on mielenkiintoinen haaste säteenseurannassa (Pharr, Jakob ja Humphreys 2016). Oikea lähestymistapa suunnan määrittämiseen riippuu täysin pinnan ominaisuuksista, joita käsitellään tarkemmin seuraavassa kappaleessa.

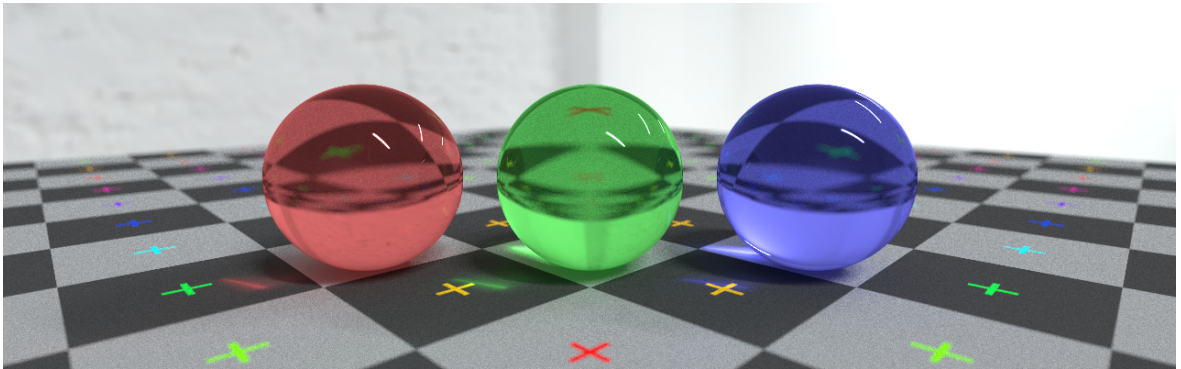
## 5 Materiaalien mallintaminen

Realistisen valokuvan luomiseksi on äärimmäisen tärkeää mallintaa monia maailmassa esiintyvien pintojen ominaisuuksia mahdollisimman tarkasti. Tätä tarkoitusta varten on kehitetty lukuisia fysiikkaan pohjautuvia kaavoja, jotka mallintavat erilaisien materiaalien ominaisuuksia erittäin todenmukaisesti. Esimerkkikuvat tuottanut C-ray-renderöintiohjelma ei ota huomioon kappaleen 4 yhtälössä esiteltyjä aallonpituuden fyysisiä ominaisuuksia. Kirjoitushetkellä C-ray approksimoi valon ominaisuuksia lineaarisia RGB -arvoja käyttäen (Koskivuori 2015-2020). Tässä kappaleessa kuvaillaan muutamia yksinkertaisia malleja erilaisten pintatyyppien mallintamiseen. Käytännössä tämä pinnan ominaisuuksien tarkastelu eli evaluointi tapahtuu kun säde on osunut pintaan. Pintamateriaalin ominaisuuksia tarkastellaan ja heijastuneen tai mahdollisesti taittuneen säteen ominaisuudet määritellään niiden perusteella. Näitä ominaisuuksia määrittävää BSDF-funktio (Bidirectional Scattering Distribution Function), joka on yksinkertaisempi muunnos kappaleen 4 yleistetystä BRDF-funktiosta (Pharr, Jakob ja Humphreys 2016).

C-ray-renderöintiohjelma kykenee yhdistelemään eri materiaalien ominaisuuksia lukuisilla tavoilla. Tämä on saavutettu syntaksipuun (engl. abstract syntax tree) tapaisella solmukaaviolla, jossa jokainen solmu suorittaa tietyn operaation materiaalia tarkastellessa. Solmu voi esimerkiksi sekoittaa kahta eri materiaalia sekoitusparametrin perusteella, Monte Carlo -menetelmää hyödyntäen. Sekoitusparametri voidaan taas johtaa toista solmua käyttäen esimerkiksi shakkikuviota tuottavasta solmusta, jonka lopputuloksena voidaan saavuttaa esimerkiksi kuvissa nähtävän 'lattian' pinnan ulkonäkö. Tällaisen solmukaavion toiminnan yksityiskohdat menevät jo reilusti aihealueen ulkopuolelle, joten lukija voi halutessaan tarkastella suoraan toteutuksen lähdekoodia lähdeluettelosta löytyvästä osoitteesta (Koskivuori 2015-2020).



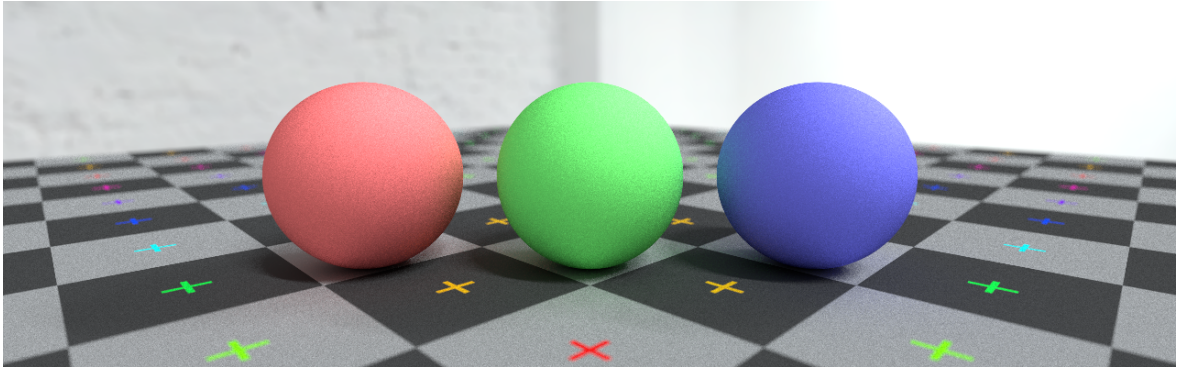
## 5.1 Lasi



Kuvio 2. Lasimateriaali testiympäristössä.

Lasimateriaalia evaluoidessa tarkastellaan pintaan osuvan säteen kulmaa suhteessa pinnan normaaliin osumakohdassa. Tästä kulmasta voidaan laskea todennäköisyysjakauma säteen heijastumiselle tai taittumiselle. Tätä jakaumaa seuraavaksi tarkastellaan monte-carlo menetelmällä valitsemalla satunnaisesti polun seuraavaksi säteeksi joko taittunut tai heijastunut säde jakauman perusteella (Pharr, Jakob ja Humphreys 2016). Lasimateriaalille on aina määritelty taittumiskerroin (engl. IOR, index of refraction). Tämän taittumiskertoimen suhde ympäröivän materiaalin vastaavaan kertoimeen määrittelee, kuinka voimakkaasti taittunut säde taittuu. Yleensä tämä ympäröivä materiaali on ilma, jonka taittumiskerroin on 1.0. Lasille yleiset taittumiskertoimen arvot ovat väliltä 1.44 – 1.90 (McGuire 2018). Kun valonsäde osuu lasikappaleen sisäpintaan, tämä prosessi toistuu käänteisesti.

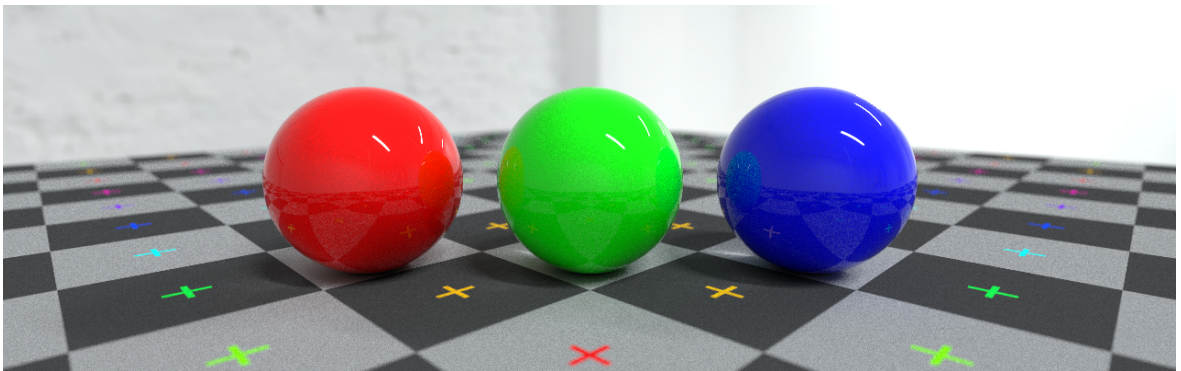
## 5.2 Mattapinta



Kuvio 3. Mattamateriaali testiympäristössä.

Diffuusipinta kuvailee yleisesti karkeita pintoja. Voidaan kuvitella tasaiselta tuntuvan mutta mikroskooppisella tasolla karkean pinnan muoto, jossa on paljon eri asennoissa olevia kohtia joista säteet heijastuvat satunnaiseen suuntaan. Tällaisen pinnan mallintaminen tehdään valitsemalla kimmonneen säteen suunnaksi satunnainen suuntavektori, jonka suunnan todennäköisyys jakautuu tasaisesti kaikkiin suuntiin (McGuire 2018).

## 5.3 Muovi

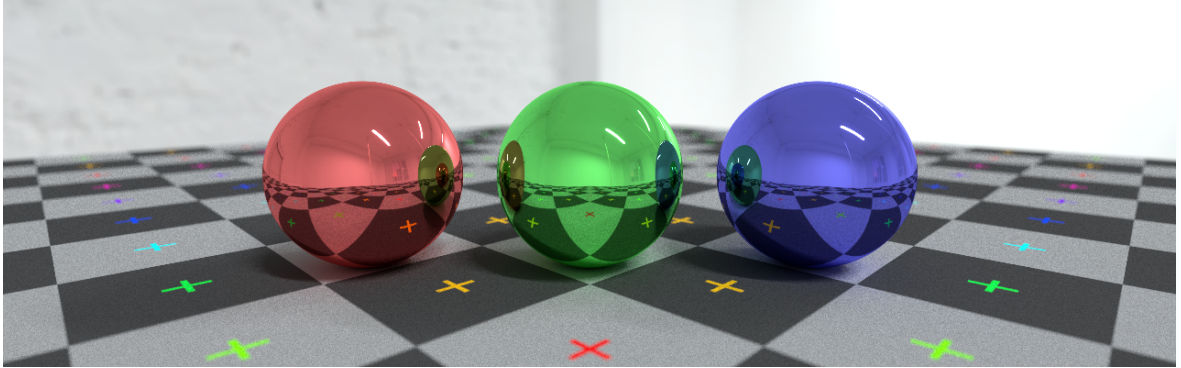


Kuvio 4. Muovimateriaali testiympäristössä.

Kuvassa 4 nähdään yksinkertainen malli, jossa sekoitetaan diffuusipinnan ja kiiltävän pinnan ominaisuuksia. Voidaan kuvitella, että karkean värillisen pinnan päällä on ohut lakkakerros, jota voidaan mallintaa lasimateriaalilla. Tämän lakkakerroksen väri on valkoinen, joten heijastuksetkin ilmenevät alkuperäisessä värissään. Huomaa, että kattovalojen heijastukset ovat

valkoisia, ei pinnan värin värjäämiä.

## 5.4 Metall



Kuvio 5. Metallimateriaali testiympäristössä.

Metalli on yksinkertaisesti mallinnettavissa oleva materiaali. Pintaan osuva säde heijastetaan samassa kulmassa suhteessa pinnan normaaliin, polun keräämää valoa attenuoidaan pinnan värin mukaan ja säde jatkaa kimpoamisia muualla. Tällä saadaan aikaan peilimäinen pinta, joka heijastaa lähes kaiken valon (Shirley 2016).

## 5.5 Muut pinnoille yhteiset ominaisuudet

Muita pinnan ominaisuuksia, kuten läpinäkyvyyttä, heijastavuutta ja muotoa voidaan muuttaa materiaaliparametreilla. Esimerkiksi pinnan normaalivektorin suuntaa muuttamalla voidaan mallintaa pieniä muutoksia pinnan muodossa, muuttamatta varsinaisesti pinnan geometriaa (Pharr, Jakob ja Humphreys 2016). Karkeus metallille ja muille materiaaleille saadaan aikaan diffuusimateriaalin tavoin biasoimalla kimmonneen säteen suuntaa satunnaisella vektorikomponentilla, jonka pituus määrittää karkeuden tason (Pharr, Jakob ja Humphreys 2016).

Fyysisen maailman valoilmioita mallintaessa on syytä ottaa huomioon tilanteet, jotka saattavat johtaa fysiikan lakien rikkomiseen. Virtuaalisessa ympäristössä tämä on hyvin helppo saavuttaa, sillä fysiikan lait eivät päde, ellei niiden asettamia rajoituksia oteta huomioon renderöintialgoritmeissa. Esimerkiksi energiaperiaate (engl. conservation of energy) tarkoittaa

valon kontekstissa sitä, että pinnalta heijastunut valoteho  $L_o$  ei voi ylittää pinnalle heijastetun valon  $L_i$  tehoa. Valotehoa ei siis saa 'ilmestyä' tyhjästä. Todellisuuden rajoitukset myös määrittää, että mikään pinta ei voi olla niin täydellisen heijastava, että 100% siihen osuneesta valosta heijastuu takaisin ympäristöön. Sama on totta myös toisessa ääripäässä: Mikään pinta ei voi olla niin tumma, että se kaappaisi kaiken siihen kohdistuneen valotehon (Pharr, Jakob ja Humphreys 2016).

## 6 Lisäkirjallisuutta

Tämän tutkielman esittelemät menetelmät ovat vain pintaraapaisu yhteen keinoon saavuttaa lähes fotorealistisia valokuvia. Tutkielmassa laajalti hyödynnetty, Pharr, Jakob ja Humphreys 2016, on noin 1200 sivun pituinen kirja, joka käsittelee erittäin yksityiskohtaisesti menetelmiä fotorealistiseen renderöintiin. Teos on niin kattava, että se on yleisesti hyväksytty referenssiteos tietokonegrafiikan alalla. Aloittelijalle huomattavasti paremmin sopiva kokonaisuus on (Shirley 2016), joka käsittelee vaiheittain etenevän toteutuksen askel kerrallaan. Toinen erinomainen referenssi on (McGuire 2018), joka sisältää helposti haettavan katalogin eri menetelmiä. Kiinnostuneen lukijan ohjaan myös tarkastelemaan (Koskivuori 2015-2020) lähdekoodia, joka on pyritty pitämään helposti seurattavana kokonaisuutena.

## 7 Yhteenveto

Tutkielmassa esitetyissä menetelmissä varsinaiseen fysiikkaan pohjautuvat kaavat valon mallinnukseen puuttuvat, ja niiden paikalla on kuvaukset yksinkertaisista approksimaatioista, joilla päästään lähelle fotorealistisia tuloksia. Tällaiset approksimaatiot ovat arvokkaita oppimisen näkökulmasta, sillä ne vapauttavat lukijan tutkimaan säteenseurannan kokonaisuutta monimutkaisten yksityiskohtien sijaan. Nämä yksinkertaiset mallit on myös helppo korvata myöhemmin todenmukaisemmilla fysiikkaan pohjautuvilla malleilla, kun pohjatieto säteenseurannan menetelmistä on vahvempi.

## Lähteet

Jon Louis Bentley. 1975. “Multidimensional Binary Search Trees Used for Associative Searching”. Stanford University.

James T. Kajiya. 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.

Valtteri Koskivuori. 2015-2020. *C-ray renderöintimoottorin lähdekoodi*. Saatavilla WWW-muodossa, <https://www.github.com/vkoskiv/c-ray>, viitattu 20.12.2020.

J. David MacDonald ja Kellogg S. Booth. 1989. “Heuristics for Ray Tracing Using Space Subdivision”. Teoksessa *Graphics Interface '89*. University of Waterloo.

Morgan McGuire. 2018. *The Graphics Codex*. 2.14. Casual Effects. <http://graphicscodex.com>.

Matt Pharr, Wenzel Jakob ja Greg Humphreys. 2016. *Physically Based Rendering: From Theory To Implementation, 3rd Edition*. Morgan Kaufmann.

Peter Shirley. 2016. *Ray tracing in one weekend*.

Ingo Wald. 2007. “On fast construction of SAH-based bounding volume hierarchies”. Teoksessa *2007 IEEE Symposium on Interactive Ray Tracing*, 33–40. IEEE.

Turner Whitted. 2005. “An improved illumination model for shaded display”. Teoksessa *ACM Siggraph 2005 Courses*, 4. ACM.