

# **Valaistusmallinnus reaaliaikaisessa 3D-grafiikassa**

Miko Mynttinen

Kandidaatintutkielma

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Helsinki, 8. toukokuuta 2015

Tiedekunta — Fakultet — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author			
Miko Mynttinen			
Työn nimi — Arbetets titel — Title			
Valaistusmallinnus reaaliaikaisessa 3D-grafiikassa			
Oppiaine — Läroämne — Subject			
Tietojenkäsittelytiede			
Työn laji — Arbetets art — Level		Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Kandidaatintutkielma		8. toukokuuta 2015	24
Tiivistelmä — Referat — Abstract			
<p>Tässä tutkielmassa tarkastellaan reaaliaikaisen kolmiulotteisen tietokonegrafiikan mallinnusta, renderöinnin toteutusta piirtoliukuhihnamallilla, ja valaistusmallinnusta. Valaistumalleista tarkastellaan erityisesti Phongin valaistusmallia, jonka fysikaalinen perusta käydään läpi ja siitä johdetaan valaistusyhtälö. Valaistusyhtälö evaluoidaan tietokoneohjelmalla samalle monikulmiomallille kolmella eri varjostustekniikalla: tasaisella, Gouraudin ja Phongin varjostuksella. Tuloksena saatuja kuvia vertaillaan ja huomataan, että tasaisella varjostuksella kappaletta mallintava monikulmioverkko jää näkyviin. Gouraudin varjostus ratkaisee tämän ongelman mattapinnoilla, mutta sen heikkous spekulaaarisesti heijastavilla pinnoilla motivoi Phongin varjostuksen käyttöön. Phongin varjostuksella valaistusmallin spekulaaarisen heijastuman vaimeneminen näyttää tasaiselta ja lopputulos on todenmukaisuudeltaan tyydyttävä.</p>			
<p>ACM Computing Classification System (CCS):</p> <p><b>Computing methodologies → Rendering</b></p> <p><b>Computing methodologies → Reflectance modeling</b></p>			
Avainsanat — Nyckelord — Keywords			
valaistus, varjostus, tietokonegrafiikka			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — Övriga uppgifter — Additional information			

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Kolmiulotteinen tietokonegrafiikka</b>	<b>2</b>
2.1	Kolmiulotteisen tietokonegrafiikan mallinnus . . . . .	2
2.2	Valaistusmallinnus . . . . .	4
2.3	Piirtoliukuhihna . . . . .	6
2.4	Näytönohjaimet . . . . .	7
<b>3</b>	<b>Phongin valaistusmalli</b>	<b>9</b>
3.1	Fysikaalinen perusta . . . . .	9
3.2	Valaistussyhtälö . . . . .	12
3.3	Blinnin-Phongin valaistusmalli . . . . .	13
<b>4</b>	<b>Varjostus</b>	<b>15</b>
4.1	Tasainen varjostus . . . . .	15
4.2	Gouraudin varjostus . . . . .	15
4.3	Phongin varjostus . . . . .	19
<b>5</b>	<b>Yhteenveto</b>	<b>22</b>
	<b>Lähteet</b>	<b>23</b>

# 1 Johdanto

Kolmiulotteisella tietokonegrafikalla tarkoitetaan kuvien tuottamista tietokoneella käyttäen lähteenä kolmiulotteisia malleja. Tässä tutkielmassa käsitellään etenkin reaaliaikaisessa kolmiulotteisessa tietokonegrafiikassa käytettäviä tekniikoita. Reaaliaikaisuus tarkoittaa, että kuvia tuotetaan yli 30 kuvaa sekunnissa. Tämä asettaa rajoituksia käytettävien tekniikoiden aikavaatimuksiin.

Tietokonegrafikalla pyritään usein tuottamaan todenmukaisia kuvia, joten käytettävät tekniikat perustuvat yleensä luonnon toiminnan mallintamiseen. Näistä tekniikoista etenkin valaistusmallinnuksella on suuri vaikutus tuotettujen kuvien todenmukaisuuden vaikutelmaan, koska kaikki näkeminen perustuu valon heijastumiseen [HvDM<sup>+</sup>13, s. 101-113]. Tässä tutkielmassa tarkastellaan yleisesti käytettyä Phongin valaistusmallia ja vertaillaan kuinka erilaiset tavat käyttää tätä valaistusmallia vaikuttavat lopputuloksena saatavan kuvan laatuun.

Tutkielman rakenne etenee seuraavasti. Luvussa 2 tutustutaan kolmiulotteisen tietokonegrafiikan ja valaistusmallinnuksen perusteisiin ja toteutukseen nykyaikaisilla näytönohjaimilla. Luvussa 3 tarkastellaan Phongin valaistusmallia. Valaistusmallin fysikaalinen perusta käydään läpi ja siitä johdetaan valaistuksessa käytettävä matemaattinen malli eli valaistusyhtälö. Luvussa 4 tutustutaan tapoihin evaluoida valaistusyhtälö kuvan pikseleille ja vertaillaan näiden vaikutusta tuotettujen kuvien todenmukaisuuteen.

## 2 Kolmiulotteinen tietokonegrafiikka

Kuvan tuottamista kolmiulotteisesta mallista kutsutaan renderöinniksi. Näkymän renderöinti tapahtuu projisoimalla mallinnettu kolmiulotteinen maailma kaksiulotteiselle tasolle. Tätä voidaan ajatella siten, että mallinnetusta maailmasta otetaan virtuaalisella kameralla valokuva.

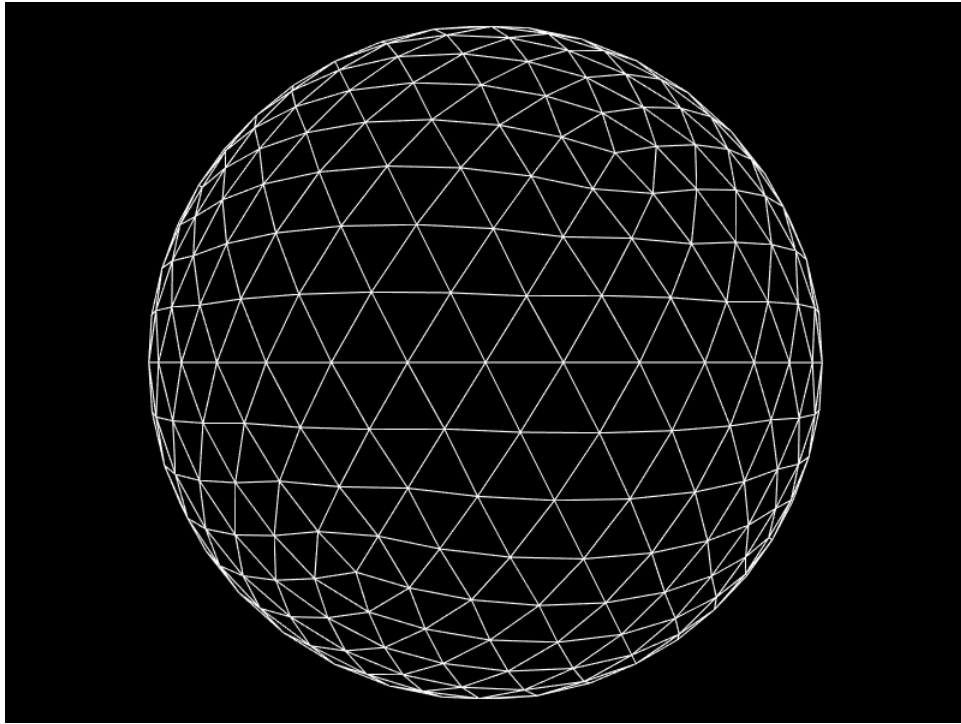
Tässä luvussa käsitellään ensin tapoja mallintaa kolmiulotteista maailmaa ja sen valaistusta, jonka jälkeen käydään läpi renderöinnin eri vaiheet ja niiden toteutus näytönohjaimella.

### 2.1 Kolmiulotteisen tietokonegrafiikan mallinnus

Kolmiulotteisen tietokonegrafiikan maailma on kokoelma malleja, jotka kuvaavat siinä olevia kappaleita. Eräs yksinkertainen malli on lista kolmiulotteisen avaruuden pisteitä, joista jokainen kolmen peräkkäisen pisteen joukko tulkitaan kolmion määrittävien kulmapisteiden joukkona. Tämä onkin nykyään yleisin tapa mallintaa kappaleita, sillä useimmat reaaliaikaista tietokonegrafiikkaa tuottavat järjestelmät odottavat syötteenä annettujen mallien koostuvan kolmioista [AMHH08, s. 8]. Pisteitä, janoja ja kolmioita kutsutaan usein piirtoprimitiiveiksi, koska monimutkaisemmat kappaleet mallinnetaan niiden yhdistelmillä. Esimerkiksi suorakulmio saadaan rakennettua neljästä kulmapisteestä, joista muodostetaan kaksi kolmiota yhdellä jaetulla sivulla. Kuvassa 1<sup>1</sup> havainnollistetaan kolmioverkolla mallinnettua palloa. Myös muunlaiset monikulmioverkot ovat yleisesti käytössä ja esimerkiksi nelikulmioita käytetään usein kolmiulotteisten mallinnusohjelmien kanssa [HvDM<sup>+</sup>13, s. 636]. Nämä monikulmioverkot voidaan kuitenkin edelleen jakaa kolmioverkoiksi, joten tässä tutkielmassa käytetään kolmion ja kolmioverkon sijasta usein myös termejä monikulmio ja monikulmioverkko.

---

<sup>1</sup>Kuvat on tuotettu C++-kielellä ohjelmoidulla tietokoneohjelmalla.



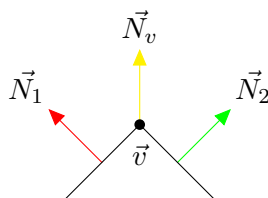
Kuva 1: Kolmioista muodostettu pallo.

Yhtälöihin perustuvat mallit voivat kuvata esimerkiksi käyriä tai pintoja. Kun kappaleen määrittämä yhtälö evaluoidaan halutuilla arvoilla kuten skaalauskerroimella ja sijainnilla, saadaan tuloksena kappaleen määrittävät pisteet. Kun pisteet tulkitaan janojen päätepisteinä, saadaan renderöityä käyrä. Saadut pisteet voidaan myös tulkita monikulmioina, jotka kolmioidaan eli jaetaan kolmioiksi, ja renderöidään pintoina.

Näkymän kolmiulotteisiin malleihin liitetään usein myös muuta tietoa kuten tekstuureja, joita voidaan ajatella kappaleen päälle liimattavina kuvina. Tekstuureja käytetään esimerkiksi kappaleen pinnan materiaalin määrittämiseen ja kuvien esittämiseen kappaleen pinnalla. Tekstuureja käytettäessä liitetään malliin yleensä myös tekstuurikoordinaatit, joilla määritetään tekstuurin sijainti kappaleen pinnalla.

Muuta malleihin usein sisällytettävää tietoa ovat kulmapisteisiin liitetyt

kulmapistenormaalit. Kulmapistenormaali on kulmapistettä ympäröivien pintojen normaalivektorien normalisoitu keskiarvo. Kulmapistenormaalia havainnollistetaan kuvassa 2.

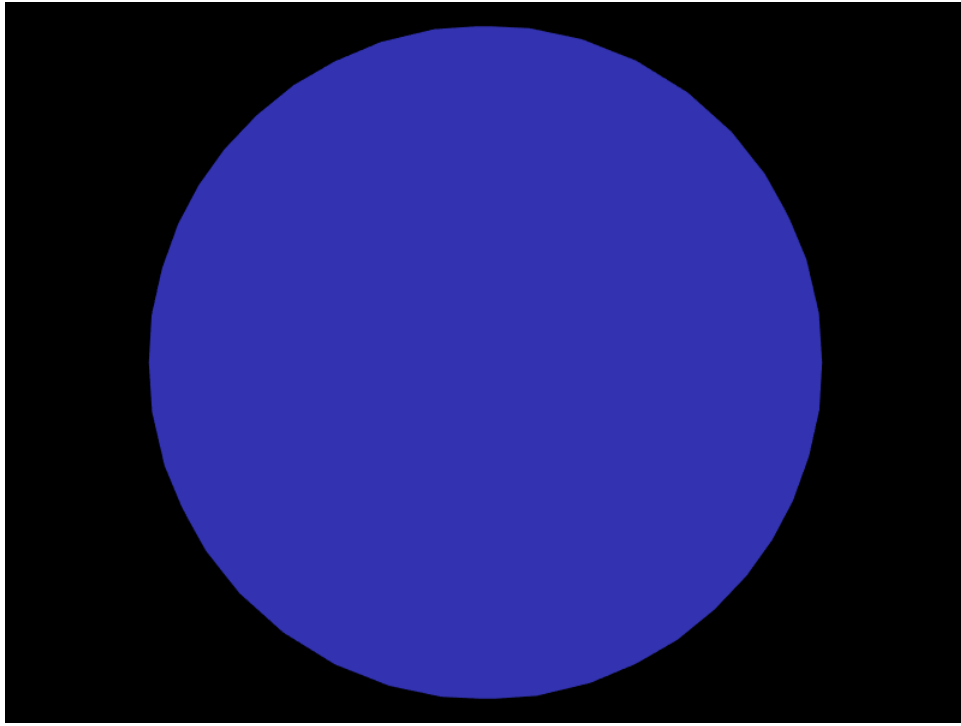


Kuva 2: Kahden pinnan normaalivektorit  $\vec{N}_1$  ja  $\vec{N}_2$ , pintojen jakama kulmapiste  $\vec{v}$ , ja kulmapistenormaali  $\vec{N}_v$ .

## 2.2 Valaistusmallinnus

Valaistusmallinnus tarkoittaa valojen ja varjojen simulointia empiirisillä tai fysiikkaan perustuvilla matemaattisilla malleilla. Näillä malleilla kolmiulotteiseen maailmaan voidaan sijoittaa valonlähteitä, joista lähtevän valon vaikutukset näkyvät kappaleiden pintojen kirkkauden ja värin muutoksina. Valaistusmallit pyrkivät mallintamaan valon käyttäytymistä luonnossa: kappaleet, joihin ei osu valoa eivät näy, ja kappaleet joihin osuu valoa, näkyvät vaihtelevalla kirkkaudella riippuen valonlähteiden määrästä, sijainnista ja kirkkaudesta. Kappaleen kirkkauden ja värin lisäksi valaistusmallit mallintavat kappaleiden materiaaleja. Materiaalit määrittävät esimerkiksi kappaleen heijastavuuden.

Mikäli valaistusmallinnusta ei olisi, ei kuvissa näkyisi varjoja ja kappaleet näyttäisivät tasaisen mattapintaisilta ja litteiltä. Kuvassa 3 havainnollistetaan kolmiulotteista mallia, joka on renderöity tasaisella värillä ilman valaistusmallinnusta. Kuvasta nähdään kuinka pallon syvyysvaikutelma katoaa täysin. Valaistusmallinnus on siis tärkeä osa realismiin pyrkivää tietokonegrafiikkaa.



Kuva 3: Tasaisella värillä renderöity pallo.

Valaistusmallit voidaan jakaa karkeasti kahteen luokkaan: paikallisiin ja globaaleihin valaistusmalleihin. Paikallisilla valaistusmalleilla simuloidaan vain valonlähteestä suoraan kappaleen pinnan pisteisiin osuvaa valoa, eli valon heijastumista pintojen välillä ei oteta huomioon. Globaalit valaistusmallit mallintavat näiden lisäksi myös valon heijastumista pintojen välillä ja ovat täten laskennallisesti paljon vaativampia. Täysin globaaleja valaistusmalleja käytetään harvemmin reaaliaikaisessa tietokonegrafikassa, jonka vuoksi tässä tutkielmassa keskitytään lokaaleihin valaistusmalleihin.

Valaistusmallilla tarkoitetaan valon käyttäytymisen mallia ja siitä johdettua valaistusyhtälöä. Valaistusyhtälö määrittää kappaleen pinnan yksittäisen pisteen heijastaman valon määrän eli pisteen kirkkauden.

Pisteen kirkkautta kutsutaan etenkin vanhemmissa tietokonegrafikan ar-



tikkeleissa intensiteetiksi, mutta nykyään tarkempien fysiikkaan perustuvien valaistusmallien yhteydessä radianssi on täsmällisempi termi [HvDM<sup>+</sup>13, s. 700]. Radianssi on radiometrian suure, joka kuvaa pinnalta heijastuvan säteilyn määrää. Koska tässä tutkielmassa käsitellään klassisia ja osittain empiirisiä valaistusmalleja, käytetään termiä intensiteetti. Harmaasävykuvia käsitellessä intensiteetin voidaan ajatella olevan liukuluku välillä  $[0, 1]$ , missä arvo 0.0 vastaa mustaa ja 1.0 valkoista. Värien esittämiseen tietokonegraafikassa käytetään yleensä punaisesta, vihreästä ja sinisestä väristä koostuvia RGB-kolmikoita, jotka 24-bittisillä väreillä rajoitetaan välille  $[0, 255]$ . Eräs keino mallintaa värikuvien valaistusta on skalaarikertolasku. Olkoon pikselin  $P$  alkuperäinen väriarvo  $c = (r, g, b)$ . Pikselin lopullinen väriarvo saadaan kertomalla kolmikko valaistusyhtälöstä saadulla intensiteetillä  $I$ , eli  $I_P = I \cdot c = (Ir, Ig, Ib)$ .

Pinnan pisteen tai pikselin värin määrittystä valaistusyhtälön avulla kutsutaan varjostamiseksi, jota käsitellään luvussa 4.

## 2.3 Piirtoliukuhihna

Kolmiulotteisen kuvan tuottamisen voidaan ajatella tapahtuvan yhteen suuntaan kulkevalla liukuhihnalla, jonka jokainen vaihe tuottaa ulostulona seuraavan vaiheen syötteen. Tässä käsiteltävä yksinkertaistettu piirtoliukuhihna määritellään Akenine-Möllerin ja kumppanien mallin [AMHH08, s. 11-27] mukaisesti. Tämä malli jakaa liukuhihnan kolmeen vaiheeseen: sovellus-, geometria- ja rasterointivaiheisiin. On tärkeää huomata että tämä on vain korkean tason käsitteellinen jako ja varsinaiset ohjelmisto- ja laitteistototeutukset ovat monimutkaisempia. Seuraavassa luvussa käsitellään miten tämä piirtoliukuhihnan malli on toteutettu nykyaikaisilla näytönohjaimilla.

Nykyaikaisissa tietokonejärjestelmissä renderöinti aloitetaan yleensä kes-

kussuorittimella ajettavalla sovellusvaiheella, jonka jälkeen geometria- ja rasterointivaiheet suoritetaan näytönohjaimella. Sovellusvaiheessa grafiikkaohjelmisto valmistelee tietokoneen näytönohjaimen renderöintiä varten. Ohjelmisto myös siirtää näytönohjaimen muistiin renderöinnissä tarvittavat tiedot kuten kappaleiden kolmioiden kulmapistetiedot ja tekstuurit.

Geometriavaiheessa suoritetaan kulmapiste- ja kolmiokohtaiset operaatiot kuten näkymän kolmioiden kulmapisteiden kääntö, siirto ja skaalaus. Muita tässä vaiheessa suoritettavia tehtäviä ovat projektio- ja perspektiivimuunnokset, joilla kolmiulotteinen näkymä rajataan ja kuvataan kaksiulotteiselle tasolle.

Rasterointivaiheessa kuvan jokaisen pikselin väriarvo tallennetaan näytönohjaimen väripuskuriin. Väripuskuri on näytönohjaimen muistissa sijaitseva taulukko, joka sisältää pikseleiden värin punaisen, vihreän ja sinisen komponentin kolmikkona. Pikseleiden värit määräytyvät sen kolmion mukaan, minkä alueelle pikselit jäävät. Mikäli pikseli jää usean kolmion alueelle, määritetään väri käyttämällä kolmioiden syvyystietoja, joiden avulla pikselille valitaan päällimmäisenä olevan kolmion väri. Mikäli kolmiolle on määritetty tekstuuri, haetaan pikselin väriarvo tekstuurista tekstuurikoordinaattien mukaisesti. Rasterointivaiheen jälkeen näytönohjaimen väripuskuri voidaan siirtää kuvaruudulle tai esimerkiksi tallentaa tiedostoksi.

## 2.4 Näytönohjaimet

Nykyään yleisin tapa tuottaa reaaliaikaista kolmiulotteista tietokonegrafiikkaa on käyttää grafiikkaan ja rinnakkaislaskentaan suunniteltua näytönohjainta, joka sisältää yhden tai useamman grafiikkasuorittimen [HvDM<sup>+</sup>13, s. 18]. Tärkeimmät syyt grafiikkasuorittinten käyttöön tietokonegrafiikassa ovat renderöinnin tehokas rinnakaistuvuus pikseli- ja monikulmiotasolla [Cro97],

ja grafiikkasuorittimen optimointi etenkin grafiikkalaskennassa käytettyihin laskutoimituksiin, kuten skalaarien ja vektorien kerto- ja yhteenlaskuun ja näiden yhdistelmiin [AMHH08, s. 32].

Nykyaikaiset grafiikkasuorittimet koostuvat jopa tuhansista monikäyttöisistä suoritinytimistä, joita kutsutaan ohjelmoitaviksi varjostinyksiköiksi. Varjostinyksiköillä voidaan suorittaa varjostinohjelmointiin tarkoitetuilla ohjelmointikielillä kirjoitettuja varjostinohjelmia eli varjostimia. Varjostimet on kehitetty piirtoliukuhinnan eri vaiheiden toteuttamiseen laitteistotasolla siten, että käyttäjä voi ohjelmoimalla muuttaa näytönohjaimen piirtoliukuhinnan toimintaa [HvDM<sup>+</sup>13, s. 927-931].

Piirtoliukuhinnan kannalta kaksi tärkeintä varjostinta ovat kulmapistevarjostin, joka suorittaa piirtoliukuhinnan kulmapistekohtaiset operaatiot, ja pikselivarjostin, joka suorittaa pikselikohtaiset operaatiot. Nämä varjostimet siis suorittavat piirtoliukuhinnan geometria- ja rasterointivaiheiden toiminnot.

Grafiikkaohjelmisto syöttää kulmapistetiedot ensimmäisenä kulmapistevarjostimelle, joka suoritetaan kerran jokaiselle näkymän kulmapisteelle. Kulmapisteet muunnetaan ohjelmoijan määrittämällä tavalla, jonka jälkeen niitä käytetään mallien piirtoprimitiivien piirtämiseen. Kulmapistevarjostimen ulostulo syötetään pikselivarjostimelle pikselikohtaisesti interpoloituina [HvDM<sup>+</sup>13, s. 935-939], joka on hyödyllistä esimerkiksi luvussa 4 käsiteltävän Gouraudin ja Phongin varjostuksen toteuttamisessa.

Pikselivarjostin ajetaan kulmapistevarjostuksen jälkeen näkymän piirtoprimitiivien alueille jääville pikseleille. Pikselivarjostimen ulostulona saadaan pikselin väriarvo ja se kirjoitetaan usein suoraan näytönohjaimen väripuskuriin.

### 3 Phongin valaistusmalli

Phongin valaistusmalli on Bui-Tuong Phongin [Pho75] vuonna 1975 kehittämä paikallinen valaistusmalli. Phongin kehittämällä valaistusmallilla voidaan laskea valon intensiteetti pinnan pisteessä, kunhan tiedetään katsojan, pinnan pisteen ja valonlähteen suhteelliset sijainnit. Pinnan materiaalin heijastavuutta ja valon ominaisuuksia voidaan muuttaa mallin vakiokertoimien avulla.

Tässä luvussa käsitellään valaistusmallin fysikaalinen perusta, josta johdetaan valaistusyhtälö. Lisäksi esitellään yleisesti käytetty valaistusmallin laskennallisesti helpompi muunnos.

#### 3.1 Fysikaalinen perusta

Tämä käsittely pohjautuu Antti Puhakan kirjan *3D-grafikka* [Puh08] lukuun 11.

Phongin empiirisessä mutta osittain fysiikkaan perustuvassa mallissa pinnan pisteeseen osuvan valon intensiteetin voidaan ajatella koostuvan kolmesta eri komponentista: taustavalosta, diffuusisti heijastuvasta valosta ja spekulaaarisesti heijastuvasta valosta.

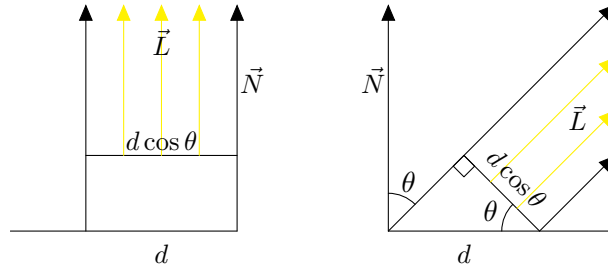
Diffuusi heijastuminen tarkoittaa valonlähteestä saapuvan valonsäteen heijastumista pinnan pisteestä kaikkiin suuntiin tasaisesti. Diffuusia heijastumista voidaan mallintaa termillä

$$I = k_d I_p,$$

missä  $I$  on pinnasta heijastuvan valon intensiteetti,  $k_d$  on pinnan diffuusin heijastumisen kerroin ja  $I_p$  on pintaan osuvan valon intensiteetti.

Pistemäistä ja äärettömän kaukana sijaitsevaa valonlähdettä voidaan mallintaa siten, että kaikilla valonlähteestä lähtevillä valonsäteillä ajatel-

laan olevan sama suuntavektori. Jos tiedetään pinnan normaalivektori  $\vec{N}$ , valonlähteeseen osoittava suuntavektori  $\vec{L}$  ja näiden välinen kulma  $\theta$ , voidaan pinnalta heijastuvan valon intensiteetti laskea Lambertin kosinilailla. Lambertin kosinilain mukaan pinnalta heijastuvan valon intensiteetti riippuu valonlähteen suuntavektorin ja pinnan normaalivektorin kulman kosinista. Tämä tarkoittaa, että valon intensiteetti on pienimmillään kun kulma on 90 astetta ja suurimmillaan kun kulma on 0 astetta. Lambertin lakia havainnollistetaan kuvassa 4, missä pinnalla alueelle  $d$  saapuu valonsäteitä tapauksissa  $\theta = 0^\circ$  (vasemmalla) ja  $\theta = 45^\circ$  (oikealla).



Kuva 4: Lambertin kosinilaki. Kuvassa  $d$  on alueen pinta-ala,  $\vec{L}$  valonlähteen suuntavektori,  $\vec{N}$  pinnan normaalivektori ja  $\theta$  näiden vektoreiden välinen kulma.

Kuvasta nähdään, että vasemmalla valonsäteiden poikkipinta-ala on pienempi joten pinnalle saapuu enemmän valonsäteitä ja täten pintaan osuvan valon intensiteetti on korkeampi. Pinnalle saapuvan valon intensiteetti riippuu siis kertoimesta  $\cos \theta$  ja pinnalta heijastuvan valon intensiteetti voidaan laskea kaavalla

$$I = k_d I_p \cos \theta . \quad (1)$$

Kun pinnan normaalivektori  $\vec{N}$  ja valonlähteeseen osoittava suuntavektori  $\vec{L}$  oletetaan normalisoiduiksi, voidaan kaava 1 kirjoittaa muodossa

$$I = k_d I_p \vec{N} \cdot \vec{L} .$$

Taustavalo tarkoittaa valoa joka on peräisin pintojen välisistä heijastumista. Tällaisen valon tarkka mallinnus on sekä fysikaalisesti että laskennallisesti vaikeaa, joten taustavaloa mallinnetaan usein vakiotermillä

$$\text{taustavalon termi} = k_d I_a ,$$

missä  $k_d$  on pinnan diffuusin heijastumisen kerroin ja  $I_a$  taustavalon intensiteetti. Vaikka tämän mallin mukaisesti myös taustavalo on pinnasta diffuusisti heijastuvaa valoa, voidaan taustavalon käyttäytymistä tarvittaessa muuttaa korvaamalla kerroin  $k_d$  taustavalon diffuusin heijastumisen kertoimella  $k_a$ .

Spekulaaria heijastumista tapahtuu kiiltävillä pinnoilla, joilla osa valonlähteestä tulevasta valosta heijastuu pinnalta katsojaa kohti voimakkaammin kuin pelkästään diffuusista heijastumisesta. Tällaista heijastumista esiintyy esimerkiksi muovi-, lasi- ja metallipinnoilla.

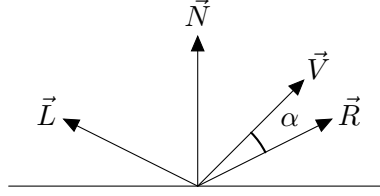
Spekulaarin heijastumisen mallissa pinnan oletetaan olevan epätäydellinen peilipinta, eli että täydellisestä peilipinnasta poiketen valo heijastuu myös muihin suuntiin kuin täydelliseen heijastussuuntaan  $\vec{R}$ . Muihin suuntiin heijastuvan valon oletetaan vaimenevan.

Täydellinen heijastussuunta  $\vec{R}$  voidaan laskea peilaamalla normalisoitu saapuvan valon valonlähteen suuntavektori  $\vec{L}$  normalisoidun pinnan normaalivektorin  $\vec{N}$  suhteen kaavalla

$$\vec{R} = 2(\vec{L} \cdot \vec{N})\vec{N} - \vec{L} .$$

Phongin valaistusmallissa spekulaaristi heijastuva valo vaimenee täydellisen heijastussuunnan vektorin  $\vec{R}$  ja katsojaan päin osoittavan suuntavektorin  $\vec{V}$  välisen kulman  $\alpha$  mukaan kertoimella  $(\cos \alpha)^n$ . Eksponentti  $n$  määritetään empiirisesti ja se kuvaa pinnan heijastuksen vaimenemisen voimakkuutta. Suurempi eksponentti mallintaa nopeampaa vaimenemista. Näitä vektoreita

ja kulmaa  $\alpha$  havainnollistetaan kuvassa 5.



Kuva 5: Spekulaarisen heijastumisen suuntavektorit. Kuvassa  $\vec{L}$  on valonlähteen suuntavektori,  $\vec{N}$  pinnan normaalivektori,  $\vec{V}$  katsojan suuntavektori ja  $\vec{R}$  täydellisen heijastussuunnan vektori.

Spekulaarin heijastumisen termi, eli pinnalta spekulaarisesti heijastuvan valon intensiteetti voidaan laskea kaavalla

$$I = k_s I_p (\cos \alpha)^n,$$

josta saadaan vektoreiden  $\vec{V}$  ja  $\vec{R}$  pistetulolla

$$I = k_s I_p (\vec{V} \cdot \vec{R})^n,$$

missä  $k_s$  on pinnan spekulaarin heijastumisen kerroin ja  $I_p$  pintaan osuvan valon intensiteetti.

### 3.2 Valaistusyhtälö

Aiemman mallin perusteella voidaan pinnan pisteiden lopullinen valaistus laskea taustavalon, diffuusin ja spekulaarisen termin summana

$$I = k_d I_a + k_d I_p \vec{N} \cdot \vec{L} + k_s I_p (\vec{V} \cdot \vec{R})^n,$$

kunhan  $\vec{N} \cdot \vec{L} \geq 0$ . Mikäli tämä pistetulo on negatiivinen, on pinta suunnattu valonlähteestä poispäin.

Pinnalle osuvan valon voimakkuus on summautuva. Mikäli näkymä sisältää useita valonlähteitä, saadaan valaistus summana kaikkien valonlähteiden

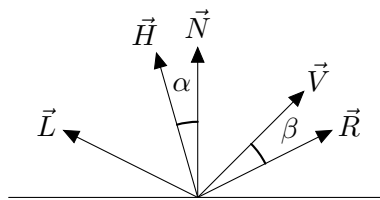
yli laskemalla

$$I = k_d I_a + \sum_{v \in V_{\text{valonlähteet}}} I_v (k_d (\vec{N} \cdot \vec{L}_v) + k_s (\vec{V} \cdot \vec{R}_v)^n),$$

missä  $I_v$  on summattavasta valonlähteestä pintaan saapuvan valon intensiteetti,  $L_v$  valonlähteeseen osoittava suuntavektori ja  $R_v$  valonlähteen täydelliseen heijastussuuntaan osoittava vektori. Aiempi yksittäisen valonlähteen suuntavektorin ja pinnan normaalivektorin tarkastelu pitää suorittaa jokaiselle valonlähteelle erikseen.

### 3.3 Blinnin-Phongin valaistusmalli

James F. Blinn julkaisi vuonna 1977 artikkelin [Bli77] joka käsitteli speulaarisen heijastumisen tarkempaa mallintamista. Artikkelissaan Blinn esittelee muunnelman Phongin valaistusmallista, jossa Phongin valaistusyhtälön täydellisen heijastussuunnan vektorin  $\vec{R}$  sijasta käytetään niin sanottua puolivälivektoria  $\vec{H}$ . Puolivälivektori voidaan ajatella pinnanormaalina, jolla katsojan suuntavektori  $\vec{V}$  olisi valonlähteen suuntavektorin  $\vec{L}$  täydellinen heijastussuunta. Tätä havainnollistetaan kuvassa 6.



Kuva 6: Blinnin puolivektori. Kuvassa  $\vec{L}$  on valonlähteen suuntavektori,  $\vec{H}$  puolivälivektori,  $\vec{N}$  pinnan normaalivektori,  $\vec{R}$  täydellisen heijastussuunnan vektori ja  $\vec{V}$  katsojan suuntavektori.

Blinnin puolivälivektorin käyttämiseen motivoi täydellisen heijastussuunnan vektorin laskemisen vaativuus. Täydellisen heijastussuunnan vektorin laskemiseen tarvitaan kaksi kertolaskua, pistetulo ja vähennyslasku. Puoli-



välivektorin laskemiseen tarvitaan kaksi yhteenlaskua ja jakolasku. Tämä pieneltä optimoinnilta kuulostava muutos on kuitenkin hyödyllinen etenkin Phongin varjostuksen yhteydessä, missä valaistusyhtälöä voidaan evaluoida miljoonia kertoja jokaisen kuvan renderöinnin aikana. Myöhempi valaistummalleja vertaillut tutkimus [NDM04] on myös todennut, että puolivälivektorin käytöllä saavutetaan realistisempi valaistus.

Puolivälivektori  $\vec{H}$  saadaan kaavalla

$$\vec{H} = \frac{\vec{L} + \vec{V}}{|\vec{L} + \vec{V}|},$$

missä  $\vec{L}$  on valonlähteen suuntavektori ja  $\vec{V}$  katsojan suuntavektori.

Blinnin-Phongin valaistusyhtälö saadaan Phongin valaistusyhtälöstä korvaamalla vektorit  $\vec{R}$  ja  $\vec{V}$  vektoreilla  $\vec{H}$  ja  $\vec{N}$ , jolloin saadaan kaava

$$I = k_d I_a + \sum_{v \in \text{Valonlähteet}} I_v (k_d (\vec{N} \cdot \vec{L}_v) + k_s (\vec{H}_v \cdot \vec{N})^n).$$

Kuvasta 6 nähdään, että vektoreiden  $\vec{H}$  ja  $\vec{N}$  kulma  $\alpha$  eroaa vektoreiden  $\vec{R}$  ja  $\vec{V}$  kulmasta  $\beta$ . Siksi Phongin ja Blinnin-Phongin valaistusyhtälöissä joudutaan käyttämään eri arvoja heijastumisen vaimenemista mallintavalle eksponentille  $n$ .

Iästään huolimatta Blinnin-Phongin valaistusmallia käytetään yhä laajasti grafiikkalaitteistoissa ja grafiikkarajapinnoissa [HvDM<sup>+</sup>13]. Se on esimerkiksi OpenGL-grafiikkarajapinnan vanhempien versioiden käyttämä valaistusmalli [SA08].

## 4 Varjostus

Yleisesti varjostus tarkoittaa kuvan tummennusta tai väritystä valaistuksen ja kolmiulotteisen vaikutelman tuottamiseksi. Tässä luvussa varjostuksella tarkoitetaan monikulmion pinnan pisteiden väritystä valaistusyhtälön avulla.

Käytännössä varjostus määrittää miten usein valaistusyhtälö evaluoidaan. Vaikka tämä määritelmä on nykyaikaisten näytönohjainten varjostinyksiköiden yleistymisen jälkeen hieman vanhentunut, sopii se tässä käsiteltävien klassisten varjostustekniikoiden yhteydessä.

Tässä luvussa käsitellään kolme yleisintä varjostusmenetelmää: monikulmiokohtainen eli tasainen varjostus, interpoloitu kulmapistekohtainen varjostus eli Gouraudin varjostus, ja interpoloitu pikselikohtainen varjostus eli Phongin varjostus.

### 4.1 Tasainen varjostus

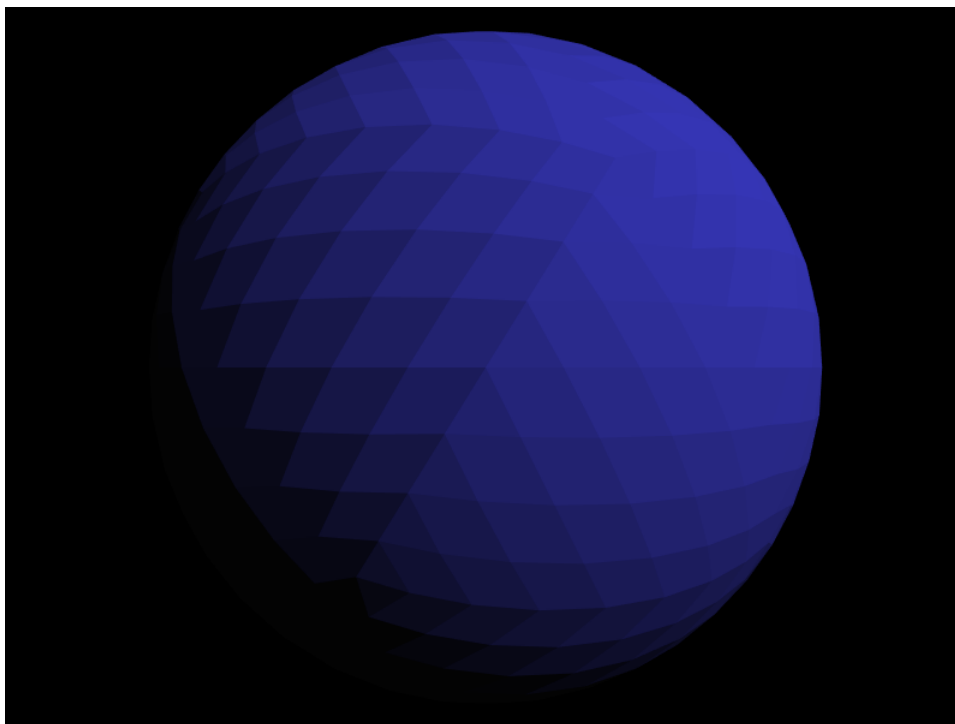
Tasaiseksi varjostukseksi kutsutaan varjostusta jossa monikulmion pinnan kaikki pikselit väritetään samalla värillä, eli valaistusyhtälö evaluoidaan vain kerran jokaiselle kappaleen monikulmiolle [AMHH08, s. 113, 155].

Valaistusyhtälössä pinnan normaalivektorina voidaan käyttää joko monikulmion pinnan normaalivektoria [War69] tai monikulmion yhden kulmapisteen kulmapistenormaalaa [HvDM<sup>+</sup>13, s.127].

Tasaista varjostusta havainnollistetaan kuvassa 7. Kuvasta nähdään tasaisen varjostuksen ongelma: kappaletta mallintavan monikulmioverkon monikulmioiden väliset rajat erottuvat selvästi.

### 4.2 Gouraudin varjostus

Gouraudin varjostus [Gou71] on Henri Gouraudin vuonna 1971 kehittämä varjostusmenetelmä joka perustuu lineaariseen interpolointiin. Motivaatio



Kuva 7: Phongin valaistusmalli ilman spekulaaia termiä tasaisella varjostuksella.

Gouraudin varjostuksen kehitykseen syntyi tasaisen varjostuksen ongelmista kaarevien pintojen yhteydessä. Ensimmäiset ratkaisut kaarevien pintojen mallinnukseen perustuivat monikulmioverkkoihin, jotka koostuivat suuresta määrästä pieniä monikulmioita. Tällaisilla malleilla ja tasaisella varjostuksella luoduissa kuvissa nähdään selkeästi monikulmioiden välisten rajojen korostuminen.

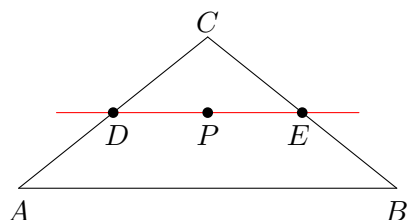
Gouraud toteaa monikulmiorajojen korostumisen kappaleen pinnalla johtuvan ihmissilmän verkkokalvon neuronien vaikutuksista viereisiin neurooneihin. Tätä vaikutusta kutsutaan lateraaliseksi inhibitioksi, ja se aiheuttaa pintojen välisten intensiteettierojen korostumista [Gla95, s. 29]. Ilmiön voi helposti huomata esimerkiksi Machin raidoiksi (Mach band effect) kutsutussa optisessa harhassa, missä tasaväriset palkit näyttävät tummemmilta kirkkaan

palkin vieressä ja vastaavasti kirkkaammilta tumman palkin vieressä.

Gouraudin mukaan eräs ratkaisu monikulmiorajojen häivytykseen on monikulmioiden välisten värierojen pienentäminen monikulmiomäärän lisäyksellä. Hän kuitenkin totetaa tämän epäkäytännölliseksi tallennustilan ja aikavaatimusten nousun takia.

Tehokkaampi ratkaisu on seuraavaksi esiteltävä Gouraudin varjostus, joka säilyttää alkuperäisen kappaletta mallintavan monikulmioverkon. Siinä valaistusyhtälö evaluoidaan käyttäen monikulmion kulmapistenormaaleja ja interpoloidaan lineaarisesti monikulmion sisälle jääville pisteille. Gouraudin alkuperäisessä artikkelissa varjostus on esitelty käyttäen nelikulmiota, mutta tässä menetelmää sovelletaan kolmiolle.

Olkoot  $A$ ,  $B$  ja  $C$  kolmion kulmapisteitä kaksiulotteisella tasolla ja  $P$  piste tämän kolmion sisällä. Olkoon lisäksi  $D$  pisteen  $P$  y-koordinaatin määrittämä jakopiste sivulla  $AC$  ja  $E$  vastaava jakopiste sivulla  $BC$ . Tämä esitetään kuvassa 8.



Kuva 8: Gouraudin varjostuksen interpolointi.

Merkitään  $\alpha = \frac{|AD|}{|AC|}$  ja  $\beta = \frac{|BE|}{|BC|}$ , missä merkintä  $|XY|$  tarkoittaa sivun  $XY$  pituutta. Olkoot  $S_A$ ,  $S_B$ ,  $S_C$  kulmapisteiden  $A$ ,  $B$  ja  $C$  valaistusarvot. Pisteiden  $D$  ja  $E$  valaistusarvot  $S_D$  ja  $S_E$  voidaan nyt laskea interpoloimalla

kaavoilla

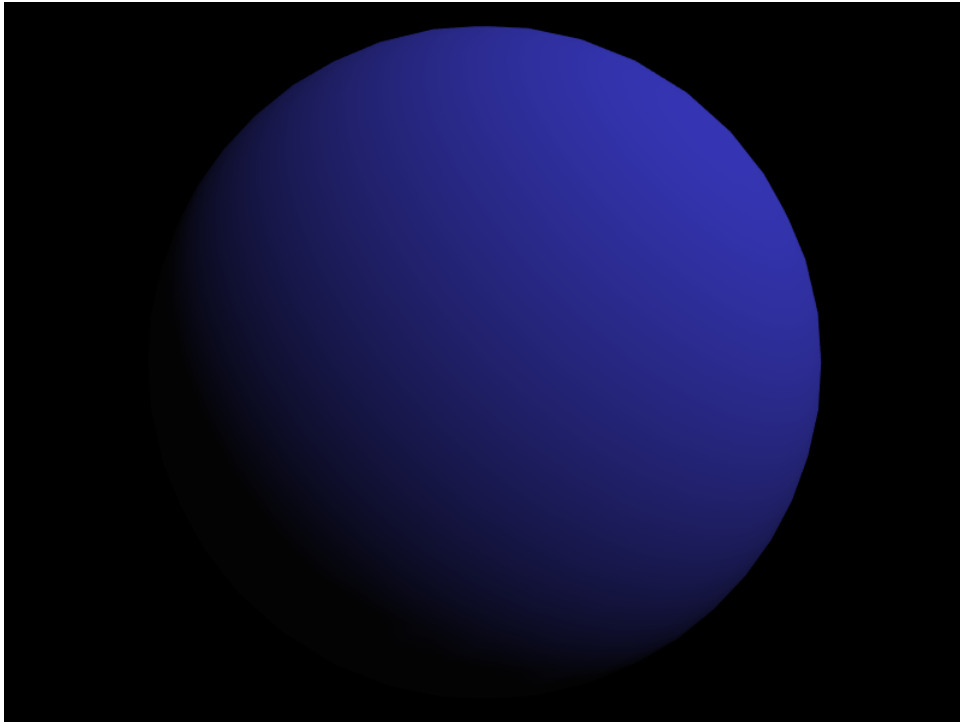
$$S_D = (1 - \alpha)S_A + \alpha S_C$$

$$S_E = (1 - \beta)S_B + \beta S_C.$$

Kun asetetaan  $\gamma = \frac{|DP|}{|DE|}$ , voidaan pisteen P interpoloitu valaistusarvo  $S_P$  laskea kaavalla

$$S_P = (1 - \gamma)S_D + \gamma S_E.$$

Kuvasta 9 nähdään, että Gouraudin varjostus soveltuu hyvin mattapinnoille. Se tuottaa tasaista varjostusta sulavamman väriliukuman pinnan monikulmioiden välille ja häivyttää kappaletta mallintavan monikulmioverkon.



Kuva 9: Phongin valaistusmalli ilman spekolaaria termiä Gouraudin varjostuksella.

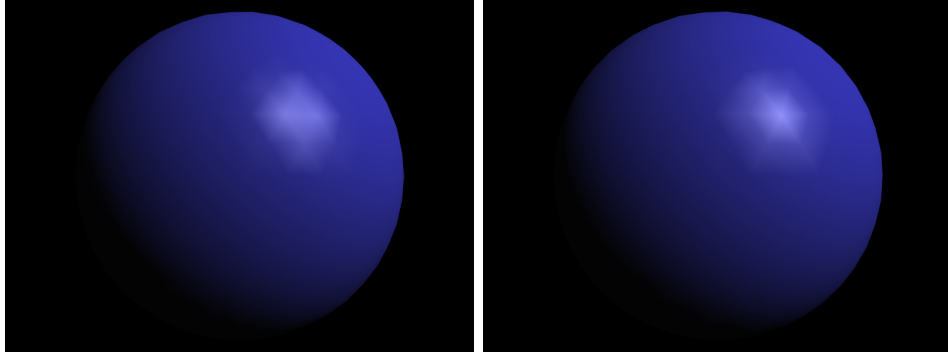
Nykyaikaisilla näytönohjaimilla Gouraudin varjostus toteutetaan kulmapistevarjostimilla liittämällä jokaiseen kulmapisteeseen kulmapistenormaali, jolla valaistussyhtälö evaluoidaan. Kulmapistevarjostimen ohjelmakoodissa väriarvon sisältävä muuttuja määritellään interpoloitavaksi. Kun nyt pikseli-varjostin suoritetaan näiden kulmapisteiden mallintamien monikulmioiden sisälle jääville pikseleille, välitetään interpoloitu väriarvo väripuskuriin, jolloin ollaan valmiita.

### 4.3 Phongin varjostus

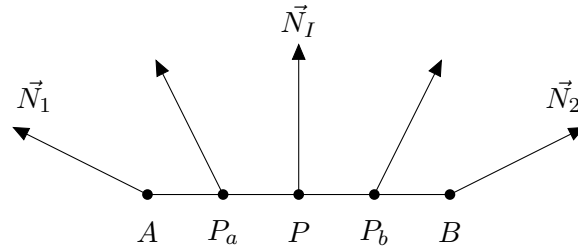
Bui-Tuong Phong esitteli Phongin valaistusmallista kertovan artikkelin [Pho75] yhteydessä myös uuden varjostustekniikan, jota nykyään kutsutaan Phongin varjostukseksi tai pikselikohtaiseksi varjostukseksi. Phongin varjostus häivyttää Gouraudin varjostuksessa havaittavan kappaleen monikulmioverkon korostumisen valaistussyhtälön vaihtuessa pinnan monikulmioiden välillä. Tämä Gouraudin varjostuksen heikkous nähdään etenkin spekulaaresesti heijastavilla pinnoilla, missä spekulaaresen heijastumisen huippukohdan muoto riippuu kappaletta mallintavista monikulmioista. Syy tähän on korkeimpien valaistusarvojen sijainti kappaleen kulmapistenormaalien kohdalla.

Gouraudin varjostuksen ongelmia havainnollistetaan kuvassa 10. Kuvassa nähdään kuinka kappaleen kääntyessä spekulaarinen heijastus siirtyy kappaletta mallintavien monikulmioiden normaalivektoreiden ja valonlähteen välisen kulman muuttuessa.

Phongin varjostus perustuu kulmapistenormaalien lineaariseen interpolointiin pikselikohtaisesti. Tässä esitellään kahden kulmapistenormaalin interpolointi monikulmion yhdellä sivulla. Olkoon  $\vec{N}_1$  ja  $\vec{N}_2$  monikulmion kulmapistenormaaleja sivulla  $AB$ , ja olkoon piste  $P$  näiden välissä. Tätä havainnollistetaan kuvassa 11. Merkitään  $\alpha = \frac{|AP|}{|AB|}$ , jolloin interpoloitu nor-



Kuva 10: Phongin valaistusmalli Gouraudin varjostuksella. Palloja on käännetty eri tavalla, jolloin spekulaarinen heijastuma näyttää eri muotoiselta. Vastaavaa ilmiötä ei tapahdu Phongin varjostuksella.



Kuva 11: Kulmapistenormaalien interpolointi. Kuvassa  $\vec{N}_1$  ja  $\vec{N}_2$  ovat kulmapistenormaaleja ja  $\vec{N}_I$  on näistä interpoloitu pisteen  $P$  normaalivektori. Merkitsemättömät vektorit ovat normaalivektoreita pisteissä  $P_a$  ja  $P_b$ .

maalivektori  $\vec{N}_I$  voidaan laskea pisteelle  $P$  sivulla AB kaavalla

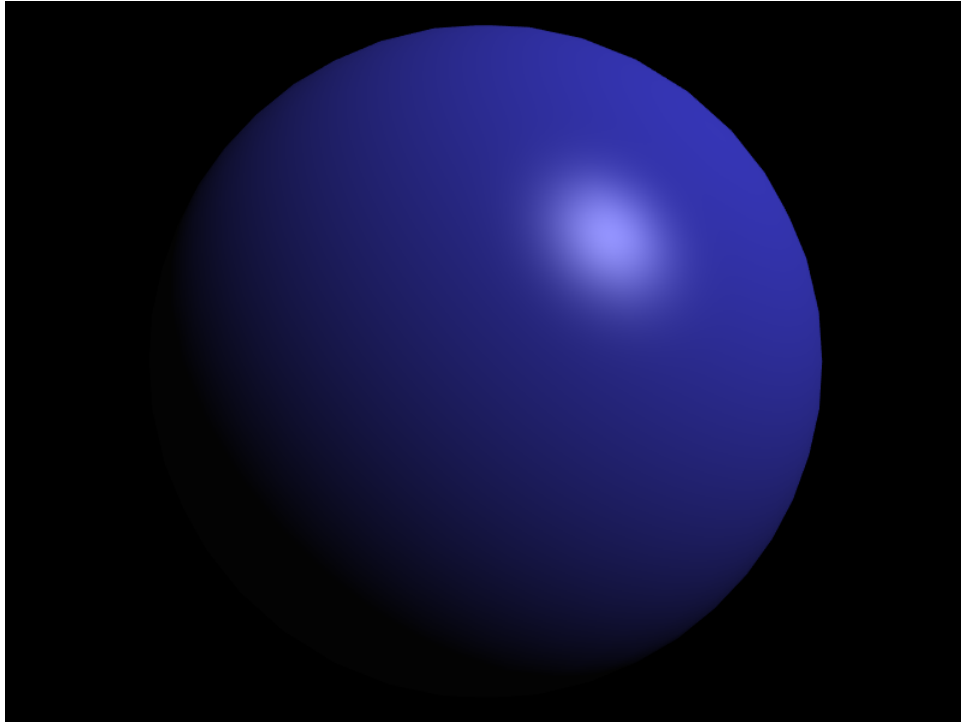
$$\vec{N}_I = (1 - \alpha)\vec{N}_1 + \alpha\vec{N}_2.$$

Tätä menetelmää soveltamalla valaistusyhtälö voidaan evaluoida monikulmion sisälle jääville pikseleille käyttäen pikselikohtaisia normaalivektoreita, jotka on laskettu monikulmion kaikista kulmapistenormaaleista.

Nykyaikaisilla näytönohjaimilla Phongin varjostus toteutetaan merkitsemällä kulmapistevarjostimen ohjelmakoodissa kulmapistenormaalit interpoloitaviksi, jonka jälkeen valaistusyhtälö voidaan evaluoida pikselivarjostimella valmiiksi interpoloiduilla pikselikohtaisilla normaalivektoreilla.

Kuvasta 12 nähdään, että Phongin varjostus Gouraudin varjostusta

todenmukaisemman lopputuloksen. Spekulaarisen heijastuman väriliukuma on pehmeämpi ja palloa mallintavaa monikulmioverkkoa ei voida enää havaita.



Kuva 12: Phongin valaistusmalli Phongin varjostuksella.



## 5 Yhteenveto

Tässä tutkielmassa käsiteltiin reaaliaikaisen kolmiulotteisen tietokonegrafiikan ja valaistusmallinnuksen perusteet. Tutkielmassa huomattiin, kuinka tasaista varjostusta käytettäessä kuvat näyttävät auttamattoman keinotekoisilta selvästi näkyvän monikulmioverkon takia. Gouraudin varjostus parantaa kuvanlaatua huomattavasti ja soveltuu mattapinnoille. Gouraudin varjostuksen heikkous näkyy spekulaaarisesti heijastavilla pinnoilla, joilla pintaa mallintava monikulmioverkko korostuu spekulaaarisen heijastuman alueella. Lisäksi Gouraudin varjostus ei ole invariantti kun kappale liikkuu.

Phongin valaistusmalli yhdistettynä Phongin varjostukseen tarjoaa tehokkaan tavan mattapintojen ja spekulaaaristi heijastavien pintojen likimääräiseen mallintaan. Mikäli tehokkuutta tarvitaan lisää, voidaan käyttää hybridiratkaisua, missä mattapinnoille käytetään Gouraudin varjostusta.

Phongin valaistusmallia, Gouraudin varjostusta ja Phongin varjostusta käytetään edelleen yleisesti 3D-mallinnusohjelmissa, pelimoottoreissa ja piirtorajapinnoissa. Tasaisen varjostuksen käyttö nykyään rajoittuu lähinnä retrotyyliin peleihin.

Tässä esitellyn valaistusmallin ja varjostustekniikoiden jälkeen on kehitetty tarkemmin fysiikkaan pohjautuvia malleja, jotka perustuvat esimerkiksi säteenjäljitykseen, mallinnettavan pinnan rakennetta kuvaaviin todennäköisyysjakaumiin, tai BRD-funktioihin (bidirectional reflectance distribution). Nämä laskennallisesti vaativammat valaistusmallit tuottavat todenmukaisempia kuvia laskentatehon kustannuksella. Vaikka osa näistä tekniikoista on jo käytössä, on esimerkiksi täysin säteenjäljitykseen perustuva renderöinti reaaliajassa toistaiseksi mahdotonta tämän hetken näytönohjaimilla.

## Lähteet

- [AMHH08] Akenine-Moller, Thomas, Haines, Eric ja Hoffman, Naty: *Real-Time Rendering*. A. K. Peters, Ltd., Natick, MA, USA, 3. painos, 2008.
- [Bli77] Blinn, James F.: *Models of Light Reflection for Computer Synthesized Pictures*. SIGGRAPH Comput. Graph., 11(2):192–198, heinäkuu 1977.
- [Cro97] Crockett, Thomas W.: *An introduction to parallel rendering*. Parallel Computing, 23(7):819–843, heinäkuu 1997.
- [Gla95] Glassner, Andrew S.: *Principles of digital image synthesis: Vol. 1*, nide 1. Elsevier, 1995.
- [Gou71] Gouraud, Henri: *Continuous Shading of Curved Surfaces*. Computers, IEEE Transactions on, C-20(6):623–629, kesäkuu 1971.
- [HvDM<sup>+</sup>13] Hughes, John F., Dam, Andries van, McGuire, Morgan, Sklar, David F., Foley, James D., Feiner, Steven K. ja Akeley, Kurt: *Computer graphics: principles and practice*. Addison-Wesley Professional, Boston, MA, USA, 3. painos, 2013.
- [NDM04] Ngan, Addy, Durand, Frédo ja Matusik, Wojciech: *Experimental Validation of Analytical BRDF Models*. Teoksessa *ACM SIGGRAPH 2004 Sketches*, SIGGRAPH '04, sivu 90, New York, NY, USA, 2004. ACM.
- [Pho75] Phong, Bui Tuong: *Illumination for Computer Generated Pictures*. Commun. ACM, 18(6):311–317, kesäkuu 1975.

- [Puh08] Puhakka, Antti: *3D-grafiikka*. Talentum, Helsinki, Finland, 1. painos, 2008.
- [SA08] Segal, Mark ja Akeley, Kurt: *The OpenGL Graphics System: A Specification (Version 3.0 - September 23, 2008)*, syyskuu 2008.
- [War69] Warnock, John E.: *A hidden surface algorithm for computer generated halftone pictures*. tekninen raportti, DTIC Document, kesäkuu 1969.