

Teknillinen korkeakoulu
Tietotekniikan osasto
Automaatiotekniikan laitos

Oskar Lönnberg

Painojäljen mallipohjainen visualisointi näyttöjärjestelmässä

Diplomityö, joka on jätetty tarkastettavaksi diplomi-insinöörin
tutkintoa varten Espoossa 8.3.1994

Työn valvoja: professori Pirkko Oittinen

Tekijä ja työn nimi: Oskar Lönnberg Painojäljen mallipohjainen visualisointi näyttöjärjestelmässä	
Päivämäärä: 8.3.1994	Sivumäärä: 85
Osasto: Tietotekniikan osasto	Professuuri: Aut-75 Graafinen tekniikka
Työn Valvoja: Professori Pirkko Oittinen	
<p>Tämän diplomityön tavoitteena on visualisoida tietokoneen näytöllä perinteisillä painomenetelmillä tuotettua painojälkeä, joka on luotu synteettisesti paperi- ja painovärimallien perusteella. Tavoitteeseen liittyy sopivien paperi- ja painovärimallien muodostaminen. Visualisoinnin tarkoituksena on pystyä tarkastelemaan painotulosta ennen varsinaista painatusta ja tällä tavalla korvata erilaiset koepainatukset ja vedostukset.</p> <p>Visualisoinnissa generoidaan kuva painojäljestä, jota tarkastellaan näyttöjärjestelmässä. Kuvan generointi tapahtuu tietokonegrafiikan menetelmiä hyväksi käyttäen. Kuvan generointi toteutetaan rakentamalla tietokoneohjelma, johon on implementoitu paperi- ja painovärimallit.</p> <p>Tässä työssä käytettävä paperimalli perustuu paperin karheusrakenteeseen ja valon heijastusominaisuuksiin. Karheusrakenne on toteutettu matriisina, joka sisältää paperin pinnan korkeusvaihtelun. Lisäksi paperimalli sisältää paperin valon heijastus ominaisuudet, jotka määrittellään koko näkyvän valon aallonpituusalueelle.</p> <p>Painovärimalli perustuu painovärin leviämiseen ja valon heijastusominaisuuksiin. Painovärin leviämisominaisuudet määritetään painovärin kontaktina paperin kanssa sekä leviämismatriisina. Tämä matriisi sisältää painovärin leviämisfunktion vasteen. Heijastusominaisuudet on toteutettu kuten paperille.</p> <p>Visualisoitava pinta generoidaan hyväksi käyttäen Phongin ja Blinnin esittämiä heijastusmalleja. Phongin heijastusmalli on valittu, koska se on yleisesti käytössä oleva heijastusmalli kaupallisissa tuotteissa. Blinnin heijastusmalli valittiin toiseksi heijastusmalliksi, koska se pyrkii mallittamaan todellista heijastusilmiötä.</p> <p>Työssä generoidaan kuvia kehitettyjen paperi- ja painovärimallien sekä valittujen heijastusmallien avulla. Työn tuloksissa tarkastellaan visualisoinnin heikkouksia sekä visualisoinnin vastaavuutta todellisuuden kanssa. Käytettyjen mallien avulla pystytään generoimaan kuvia painojäljestä.</p>	

Author and name of the thesis: Oskar Lönnberg Modelbased Visualizing of Printed Matter on Computer Screen	
Date: 8.3.1994	Number of pages: 85
Faculty: Faculty of Computer Science	Professorship: Aut-75 Graphic Arts Technology
Supervisor: Professor Pirkko Oittinen	
<p>The aim of this thesis is to visualize printed matter on computer screen which is produced by conventional printing methods. The printed matter is created synthetically using paper and printing ink models which are evaluated in this thesis. The purpose of visualization is to provide possibility of observing printed matter before the actual printing. With visualization printed matter test printing can be replaced.</p> <p>In visualization a picture of printed matter is generated which is to be observed on computer screen. The picture is generated with methods used in computer graphics. The generation of the picture is realized with a computer program where paper and printing ink models are implemented.</p> <p>The paper model used in this thesis is based on the roughness structure and light reflection characteristics of paper. The roughness is realized in a matrix which contains the differences of the paper surface depth. The light reflection characteristics of the paper are specified in the wave area of the visible electromagnetic waves.</p> <p>The printing ink model is based on the spreading and the light reflection characteristics of printing ink. The spreading characteristics of printing ink is defined with the contact of paper and printing ink and in a spreading matrix. The spreading matrix contains values of desirable spreading function. The light reflection characteristics are implemented in the same manner as for the paper.</p> <p>The surface to be visualized is generated with illumination models introduced by Phong and Blinn. The illumination model introduced by Phong was chosen because it is widely used in commercial products. The illumination model introduced by Blinn was chosen because it tends to model the illumination effect of real surfaces.</p> <p>Pictures were generated in according to illumination models and paper and printing ink models introduced in this thesis. The reality of visualized printed matter and the weaknesses of the visualization are evaluated in the results of this thesis. The results imply that pictures of printed matter can be generated with models used in this thesis.</p>	

Alkusanat

Tämä työ on tehty Teknillisen korkeakoulun graafisen tekniikan laboratoriossa.

Esitän kiitokseni työ valvojalle professori Pirkko Oittiselle mielenkiintoisesta aiheesta sekä Tarja Peromaalle ja koko muulle graafisen tekniikan laboratorion henkilökunnalle saamastani avusta ja hyvistä neuvoista. Lisäksi haluaisin kiittää Marjo Taalikkaa hänen osuvista kommenteistaan sekä rajattomasta kärsivällisyydestään.

Espoossa 4.5.1995

Oskar Lönnberg

SISÄLLYSLUETTELO

MATEMAATTISET MERKINNÄT

1 JOHDANTO	1
2 VALON HEIJASTUSMALLIT TIETOKONEGRAFIKASSA	2
2.1 Valon käyttäytyminen optisella rajapinnalla	2
2.2 Diffuusi heijastuminen	5
2.3 Phongin heijastusmalli	7
2.4 Blinnin heijastusmalli	10
2.4.1 Geometriatermi	11
2.4.2 Fresnelin heijastuksen määrittäminen	14
2.4.3 Distribuutiofunktio	16
2.5 Pintastruktuurin kuvaaminen tekstuurin avulla	18
2.6 Valaistuksen mallittaminen	21
2.7 Värin mallittaminen	22
3 PAINOJÄLJEN VISUALISOINNIN TARKASTELTAVAT MATERIAALIT	23
3.1 Paperi	23
3.1.1 Paperin karheusmalli	23
3.1.2 Paperin heijastusominaisuuksien määrittäminen	27
3.2 Painovärin leviäminen painettavalle pinnalle	28
3.3 Optisen kuvan muodostuminen painojäljestä	30
4 PAINOJÄLJEN VISUALISOINNIN ASETTAMAT VAATIMUKSET NÄYTTÖJÄRJESTELMÄLLE	33
4.1 Kuvan muodostus CRT-näyttöjärjestelmässä	33
4.2 Näyttöjärjestelmän kalibrointi	35
5 VISUALISOINNIN IMPLEMENTOINTI	38
5.1 Käytetty ohjelmointiympäristö ja valmiit ohjelmat	38

5.2 Ohjelmiston rakenne	38
5.2.1 Pääohjelma	40
5.2.2 Input/Output/Tiff	41
5.2.3 Heijastusmalli	41
5.2.4 Materiaalien mallit	41
5.2.5 Väri- ja suuntavektorit	44
5.3 Heijastusmallin ratkaisu	45
5.4 Värin implementointi	49
5.5 Paperin karheusmalli	52
5.5.1 Karheusmatriisi	52
5.5.2 Pinnan normaalin määrittäminen	54
5.5.3 Itsensä varjostus ja maskaus	57
5.6 Painovärin leviämismallin implementointi	58
5.7 Valonlähteen mallittaminen	61
6 OHJELMALLA TEHTY KOESARJA	63
6.1 Valonmallin parametrit	65
6.2 Paperimallin parametrit	67
6.3 Painovärimallin parametrit	71
7 TUKIMUSTULOSTEN TARKASTELU	74
7.1 Tutkimuksessa käytettyjen mallien todenmukaisuus ja käyttökelpoisuus	74
7.2 Mallien tehokkuus ja ohjelman konekapasiteetin tarve	77
8 YHTEENVETO	80
LÄHTEET	82
LIITTEET	

MATEMAATTISET MERKINNÄT

Vektorimerkinnät

N	Pinnan normaalin yksikkövektori, jonka oletetaan olevan samalla puolella pintaa kuin mistä valo tulee
L	Valon tulosuunnan ilmaiseva yksikkövektori
R	Yksikkövektori, joka ilmaisee peiliheijastussuunnan säteelle L
E	Tarkastelusuunnan osoittava yksikkövektori
H	Mikropinnan normaalin suunta, joka aiheuttaa peiliheijastumisen valon tulosuunnasta L katselusuuntaan E .
T	Taittuneen valonsäteen suuntainen yksikkövektori
Q_x, Q_y	Pinnan gradienttivektorit
P	Häiriövektori
A • B	Pistetulo, joka lasketaan vektorien <i>x</i> , <i>y</i> ja <i>z</i> komponenttien avulla seuraavasti
$\mathbf{A} \cdot \mathbf{B} = \mathbf{A}_x \mathbf{B}_x + \mathbf{A}_y \mathbf{B}_y + \mathbf{A}_z \mathbf{B}_z$	
A ⊗ B	Ristitulo eli vektoritulo, jolla saadaan vektorien määrämien tason normaali selville laskemalla seuraavasti
$\mathbf{A} \otimes \mathbf{B} = \begin{vmatrix} \mathbf{x} & \mathbf{y} & \mathbf{z} \\ \mathbf{A}_x & \mathbf{A}_y & \mathbf{A}_z \\ \mathbf{B}_x & \mathbf{B}_y & \mathbf{B}_z \end{vmatrix}$	

Kulmamerkinnät

<i>i</i>	Valon tulosuunnan L ja pinnan normaalin N välinen kulma
<i>e</i>	Pinnan normaalin N ja tarkastelusuunnan E välinen kulma
<i>r</i>	Pinnan normaalin N ja heijastuneen säteen R välinen kulma
<i>t</i>	Taittuneen säteen ja pinnan normaalia vastakkaiseen suuntaan olevan vektorin välinen kulma eli T :n ja -N :n välinen kulma
<i>v</i>	Heijastuneen säteen R ja tarkastelusuunnan E välinen kulma
<i>h</i>	Pinnan normaalin N ja mikropinnan normaalin H välinen kulma
<i>l</i>	Valonsäteen -L ja valolähteen todellisen suunnan välinen kulma

Kuvankäsittely

(x,y)	Piste xy -koordinaatistossa
X,Y	Kuvan dimensiot pikseleinä
d	Pikselin koko
$f(x,y)$	Paperin karheusfunktio
z_{max}	Maksimikarheus funktiolle $f(x,y)$
z_c	Kontaktipinta karheusfunktiolle $f(x,y)$
K	Karheusmatriisi
a,b	Matriisien dimensiot

Käytetyt yksiköt

m	metri; 1 m = 1000 mm
mm	millimetri; 1 mm = 1000 μ m
μ m	mikrometri, 1 μ m = 1000 nm
nm	nanometri
bit	bitti; saa arvon 0 tai 1
B	tavu; 1 B = 8 bittiä
kB	kilotavu; 1 kB = 1024 B
MB	megatavu; 1 MB = 1024 kB
pix	pikseli; kuva-alkio

1 JOHDANTO

Paperille tuotetun painojäljen visualisoinnin merkitys on kasvanut tietokoneiden yleistymisen myötä. Yhä useammalla on mahdollisuus käyttää ammattitason taitto-työkaluja, mutta harvalla on mahdollisuus koevedosten tekemiseen. Kun on kyse esitteestä, mainoksesta tai muusta julkaisusta, tekijä haluaa varmistaa lopputuloksen ulkonäön ennen varsinaista paperille painamista. Jos tulos ei ole tyydyttävä ensimmäisen koepainatuksen jälkeen, joudutaan tekemään useampia koepainatuksia. Tämä tulee kalliiksi. Turhien painatusten välttämiseksi julkaisun tekijän pitäisi pystyä varmistamaan painotuotteensa ulkonäkö riittävän tarkasti ennen varsinaista painatusta. Tähän tarvitaan vedostustyöasema, joka visualisoi painojälkeä.

Tämän diplomityön tavoitteena on visualisoida tietokoneen näytöllä painojälkeä, joka on luotu synteettisesti paperi- ja painovärimallien perusteella. Tavoitteena on tehdä riittävän hyviä kuvia painojäljen visualisoimiseksi. Työssä syvennytään lisäksi paperin karheuden esittämiseen, koska painoväriin leviämisen ja paperin visualisoinnin kannalta on tärkeää saada paperin karheus esille. Paperin ja painoväriin malleissa keskitytään niiden visuaaliseen ulkonäköön vaikuttaviin ominaisuuksiin. Painojäljen visualisoinnissa pyritään simuloimaan yksiväristä kompaktia painettua pintaa. Kompaktiin yksiväriseen pintaan on keskitytty sen yksinkertaisuuden vuoksi. Kun pystytään mallittamaan ja visualisoimaan kompakti yksivärinen pinta, voidaan tämä helposti laajentaa moniväriseen painatukseen.

Paperin ja painoväriin malleissa sekä visualisoinnissa keskitytään mikrometrien kokoiseen vaihteluun. Tähän tasoon on keskitytty, koska taso on hieman ihmisen näkökyvyn alapuolella ja koska visualisoinnissa voidaan käyttää geometristä optiikkaa. Näkyvän valon aallonpituus on pienempää kuin yksi mikrometri. Paperin karheuden visuaaliset erot näkyvät parhaiten mikrometritasolla. Tätä suuremmat painojäljen ominaisuudet liittyvät painatuksen mekaanisiin ominaisuuksiin, joita ei tässä työssä ole tarkoitettu käsitellä.

Teoriaosuudessa käsitellään tutkimusongelman kannalta olennaisia algoritmeja sekä tietokonegrafiikassa käytettyjä heijastusmalleja. Näistä malleista haetaan ongelman ratkaisun kannalta tehokkaita ja todenmukaisia heijastusmalleja. Malleja on tarkoitettu soveltaa paperin ja painoväriin mallien visualisointiin. Visualisointi suoritetaan näyttöjärjestelmän asettamissa rajoissa. Paperi- ja painovärimallien generoiminen kuvaksi suoritetaan tietokonegrafiikan työkalulla ja menetelmillä. Tutkimuksessa suunnitellaan tietokoneohjelma, johon on implementoitu paperin ja painoväriin mallit ja algoritmit painojäljen kuvien simuloimiseen. Tietokoneohjelma tehdään C-ohjelmointikielellä PC-laitteistoympäristössä.

Tämä diplomityö liittyy Teknillisen korkeakoulun graafisen tekniikan laboratorion tutkimuksiin. Laboratorion tutkimushankkeista työ sivuaa elektronisen painamisen projektia. Työ liittyy läheisesti Jukka-Matti Löppösen diplomityöhön ”CRT-pohjainen visualisointi”^[25]. Löppönen visualisoi työssään CCD-kameralla kuvattuja todellisia paperipintoja. Hän tarkastelee lähinnä CRT-näytön ominaisuuksia visualisoinnissa. Oma diplomityöni käsittelee visualisointia painojäljen synteettisen generoinnin kannalta.

2 VALON HEIJASTUSMALLIT TIETOKONEGRAFIIKASSA

Heijastusmalli on funktio, joka määrää kappaleen jokaisen pisteen intensiteetin riippuen valonlähteestä, kappaleesta ja havaitsijan sijainnista. [8] [11] [12] [14] [39] Heijastusmallien kehitys on seurannut tietokoneiden laskentakapasiteetin kehitystä. Mitä tehokkaampia koneita on tullut markkinoille sitä monimutkaisempia heijastusmalleja on käytetty esineiden visualisointiin.

Visualisoinnilla tarkoitetaan tiedon muuttamista kaksi- tai kolmedimensionaaliseksi kuvaksi, joka auttaa ymmärtämään tiedon luonnetta. Tässä työssä visualisoinnin tavoitteena on tuottaa ihmisen näköjärjestelmän kannalta todellisuutta vastaava kuva eli tavoitteena on, että painojäljen mallien avulla pystytään simuloimaan todellista painojälkeä.

2.1 Valon käyttäytyminen optisella rajapinnalla

Valo on sähkömagneettista säteilyä, joka on muodostunut kohtisuoraan toisiaan vastaan olevista magneettivuosta ja sähkökentästä, joiden voimakkuus vaihtelee jaksollisesti samassa tahdissa. Valon aallonpituus vaihtelee 380 nanometrin (nm) - 770 nanometrin välillä. Valon spektri, valon eri aallonpituudet, vaikuttaa havaittuun värihavaintoon. Valon heijastumiset ja taittumiset johtuvat valon sähkömagneettisesta luonteesta. Valon käyttäytymistä voidaan kuitenkin mallittaa geometrisillä malleilla eli geometrisellä optiikalla, jos tarkasteltavien pintojen vaihtelu on suurempaa kuin tarkasteltava valon aallonpituus. Jos pinnan vaihtelu on pienempää kuin valon aallonpituus, tapahtuu rajapinnassa difraktio, jonka mallittaminen ja käsittely tapahtuu aalto-optiikalla, joka on tietojenkäsittelykapasiteettia kuluttavaa. [12]

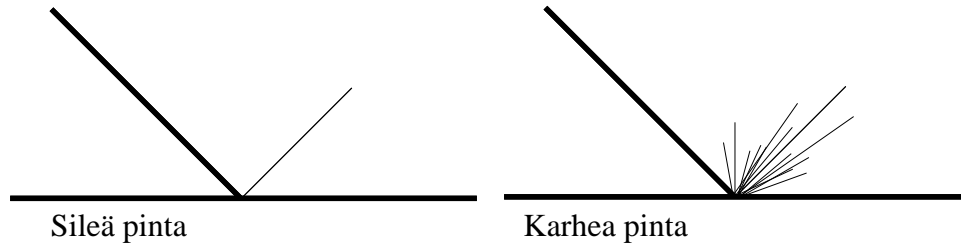
Tietokonegrafiikassa tarkastellaan erilaisten aineiden välisiä rajapintoja ja niiden aiheuttamia valon heijastumisia ja taittumisia. Sähköä johtavista aineista (esim. metallit) valo /*Kuva 1*/ heijastuu lähes kokonaan materiaalien rajapinnasta. Tämä johtuu sähköä johtavissa aineissa olevista vapaista elektroneista sekä valon sähkömagneettisesta luonteesta. Sähköä johtavan aineen pinnat ovat opaakkeja eli materiaalin läpi ei pääse valoa. [12]

Dielektrisissä aineissa eli aineissa, joissa ei ole vapaita elektroneja, tapahtuu valon sekä heijastumista että taittumista /*Kuva 2*/. Puhdas dielektrinen materiaali on läpinäkyvä (esim. lasi). Osa pintaan tulevasta valosta taittuu rajapinnan läpi riippuen valon tulokulmasta ja pinnan materiaalien ominaisuuksista. Jos materiaali on rakentunut monista pienistä osasista kuten hiekka, materiaali vaikuttaa opaakilta. Tämä johtuu lukuisista taittumisista ja heijastumisista materiaalin sisällä. [12]

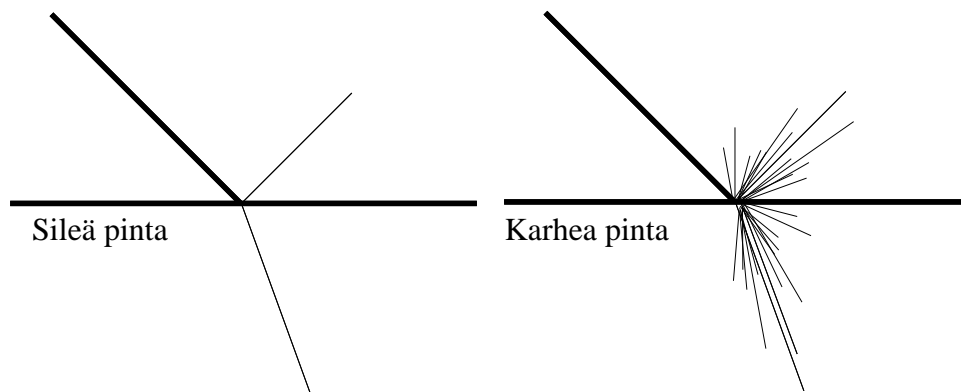
Pinnan sileydestä riippuu, mikä on heijastumisen ja taittumisen jakauma avaruuden yli kullakin materiaalilla. Jos pinta on karhea, valo siroaa eli hajaantuu laajalle alueelle. Kun valo tulee samansuuntaisesti pinnan tason kanssa, tapahtuu täydellinen heijastuminen kaikilla aineilla. [12]

Komposiittimateriaalit ovat aineita, jotka ovat muodostuneet sähköä johtavista ja

dielektristä aineista (esim. maalit). Komposiittimateriaaleissa tapahtuu valon heijastumista ja taittumista materiaalin rajapinnalta sekä hajaheijastumista materiaalin sisällä. Valon kulkiessa materiaalin läpi se siis siroaa ja absorboituu eli valo imeytyy materiaaliin.^[12]

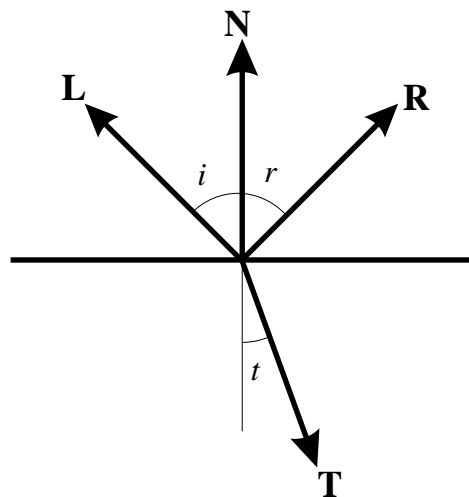


Kuva 1 Valon säteen heijastuminen sähköä johtavasta aineesta ^[12]



Kuva 2 Valon säteen heijastuminen dielektristä aineesta ^[12]

Kun tarkasteltavat dimensiot ovat suurempia kuin valon aallonpituus, valoa voidaan mallittaa geometrisilla malleilla. Geometrisissä valomalleissa valo oletetaan säteeksi. Tällöin on helppo käsitellä valon heijastumisia ja taittumisia vektorigeometrian avulla /Kuva 3/.^[12]



Kuva 3 Valon heijastuminen ja taittuminen optiselta rajapinnalta

Kun valo heijastuu tasaiselta pinnalta, valon heijastuskulma on yhtä suuri kuin valon tulokulma eli

$$i = r \quad (1)$$

jossa

i on valon tulokulma ja

r on valon heijastuskulma /Kuva 3/.

Taittuva säde noudattaa Snellin lakia ^{[12] [39]}

$$n_i \sin(i) = n_t \sin(t) \quad (2)$$

jossa

t on taittuneen säteen taittumiskulma,

n_i on optinen taitekerroin materiaalille, josta tuleva säde tulee ja

n_t on optinen taitekerroin materiaalille, johon säde taittuu.

Kuten kaavoista (1) ja (2) huomataan; heijastuneen valonsäteen tai taittuneen valonsäteen avulla voidaan määrittää tulevan valonsäteen suunta. Kaavojen (1) ja (2) avulla voidaan valonsädettä seurata takaperin valonlähteeseen. Tätä ominaisuutta käytetään hyväksi tietokonegrafiikassa Ray Tracing algoritmissä. ^{[8] [12] [14] [39]}

Tulevan valon intensiteetin jakautuminen taittuneen ja heijastuneen säteen kesken

riippuu Fresnelin yhtälöistä. Nämä yhtälöt ovat johdetut Maxwell'n yhtälöistä elektromagneettisille kentille sileällä pinnalla. Fresnelin yhtälöt riippuvat materiaalien taitekertoimista n_i ja n_t , valon polarisaatiosta sekä valon tulokulmasta. Yhtälöissä (3) ja (4) lasketaan valon heijastuneen osan amplitudin suhde tulevan valon amplitudiin. Amplitudit määritetään valonsäteen polarisaatioiden suhteen, pinnan suuntaan sekä kohtisuoraan pintaa vasten:

$$r_{II} = \frac{n_t (\mathbf{N} \cdot \mathbf{L}) + n_i (\mathbf{N} \cdot \mathbf{T})}{n_t (\mathbf{N} \cdot \mathbf{L}) - n_i (\mathbf{N} \cdot \mathbf{T})} \quad (3)$$

$$r_{\perp} = \frac{n_i (\mathbf{N} \cdot \mathbf{L}) + n_t (\mathbf{N} \cdot \mathbf{T})}{n_i (\mathbf{N} \cdot \mathbf{L}) - n_t (\mathbf{N} \cdot \mathbf{T})} \quad (4)$$

jossa

\mathbf{N} on pinnan normaalin yksikkövektori,

\mathbf{L} on valon tulosunnan yksikkövektori ja

\mathbf{T} on taittuneen säteen yksikkövektori /Kuva 3/.

Valon energia on suhteessa valon amplitudin neliöön. Tietokonegrafiikassa oletetaan valon olevan ympyräpolarisoitunutta, jolloin saadaan heijastuneen valon amplitudin suhde tulevan valon amplitudiin kaavoista (3) ja (4) muotoon

$$F_r = \frac{1}{2} (r_{II}^2 + r_{\perp}^2) \quad (5)$$

Tästä Fresnelin heijastuksesta F_r saadaan taittuneen säteen energian suhde tulevan säteen energiaan energian säilymisperiaatteen mukaan $F_t = 1 - F_r$.^[12] Kun taittumisilmiötä tarkastellaan sähköä johtavissa aineissa, Fresnelin yhtälöt saavat toisenlaisen muodon. Fresnelin yhtälöitä sähköä johtaville aineille ei esitetä tässä, koska työssä tarkasteltavat materiaalit ovat dielektrisiä aineita. Sähköä johtavissa aineissa Fresnelin yhtälöt saavat arvoja lähellä yksi, kun taas dielektreillä aineilla Fresnelin yhtälöt ovat eksponentiaalisia valonsäteiden tulokulman suhteen.^[12]

2.2 Diffuusi heijastuminen

Esineiden visualisointiin käytetään tietokonegrafiikassa useita erilaisia heijastusmalleja. Kaikki mallit eivät kuitenkaan vastaa todellisia valon heijastusominaisuuksia. Yksinkertaisin heijastusmalli, jota käytetään tietokonegrafiikassa, on diffuusin heijastuksen malli [2] [3] [11] [12] [14] [36] [39]. Vieläkin yksinkertaisempia

heijastusmalleja kuin diffuusi heijastusmalli on olemassa. Tällaiset heijastusmallit eivät vastaa todellista valon käyttäytymistä ja tämän vuoksi ne jätetään tässä huomiotta.

Diffuusissa heijastusmallissa tarkasteltavan pinnan oletetaan heijastavan valoa tasaisesti joka suuntaan eli täysin diffuusisti. Tässä heijastusmallissa oletetaan tarkasteltavalta pinnalta puuttuvan valon heijastuksen suuntavalikoivat säteet. Diffuusista heijastusmallia käytetään hyväksi kehittyneemmissä heijastusmalleissa diffuusin termin esittämiseen.

Pinnan diffuusia heijastusominaisuutta kuvataan Lambertin cosini-lailla, joka olettaa pinnan heijastavan pinnalle tulleen valon intensiteetin tasaisesti joka suuntaan^{[8] [11] [52]}:

$$I = p_d I_L \cos(i) \quad (6)$$

jossa

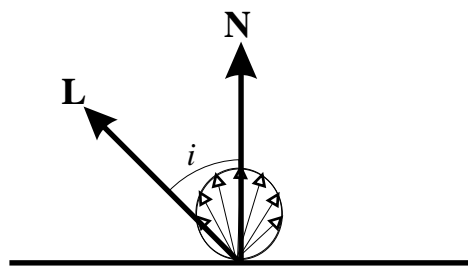
I pinnan intensiteetti,

I_L on tulevan valonsäteen intensiteetti,

p_d on diffuusin heijastuksen kerroin, joka riippu tarkasteltavasta pinnasta ja

i on pinnan normaalin ja valon tulosuunnan välinen kulma. /Kuva 4/.

Kaavassa (6) valonlähteen oletetaan olevan pistemäinen. Termi $I_L \cos(i)$ kaavassa (6) kuvaa pinnan illuminanssia.



Kuva 4 Karhean pinnan diffuusi heijastus cosini-lain mukaan

Kun valon tulokulma on yli 90° , eli valo tulee pinnan takaa, ei yhtälö (6) enää päde. Koska valo läpäisee kappaleen, mallia joudutaan tarkastelemaan taittumisilmiönä. Jos valonlähteet on sijoitettu niin, että kaikkiin tarkasteltaviin pintoihin ei osu suora valaistus, kappaleen pintaan syntyy varjo. Tällaisesta varjokohdasta yksityiskohdat häviävät näkyvistä ja kohta tulee mustaksi. Jotta vältettäisiin mustat kohdat, malliin lisätään vakiollinen diffuusi eli ambientti valaistus^{[3] [11] [12] [14] [39]}. Tämän valaistuksen oletetaan tulevan kaikilta suunnilta. Kun lisätään ambientti valaistus ja kirjoitetaan

kulman i cosini vektorien avulla, saadaan kaava (6) seuraavaan muotoon:

$$I = p_a I_A + p_d I_L \max(0, \mathbf{N} \cdot \mathbf{L}) \quad (7)$$

jossa

I_A on diffuusin valaistuksen intensiteetti pinnalla,

p_a on diffuusin valaistuksen kerroin,

\mathbf{N} on pinnan normaalin yksikkövektori ja

\mathbf{L} on valon tulosuunnan yksikkövektori. /Kuva 4/.

$\max()$ operaatiolla vältetään yli 90 asteen kulmat eli valo ei tule kappaleen takaa. On huomattava, että tässä käsiteltävät mallit ovat pinnan heijastusmalleja eivätkä kappaleen geometriamalleja. Erilaiset algoritmit kuten Ray tracing -algoritmit käsittelevät kappaleiden muodosta aiheutuvia ongelmia; esimerkiksi toisen kappaleen aiheuttama varjo tarkasteltavalle pinnalle. Ray tracing -algoritmeista on varsin paljon kirjallisuutta [8] [12] [14] [36] [39].

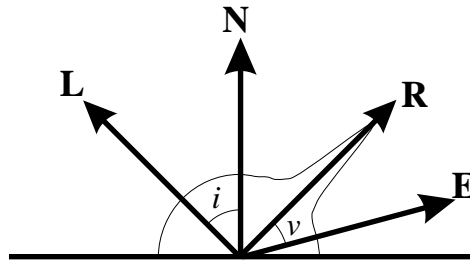
Diffuusit heijastusmallit eivät ota huomioon kappaleen pinnan karheudesta aiheutuvaa pinnan varjostusta itselleen. Ambientin valaistuksen aiheuttama termi yleensä ilmaistaan yhdellä vakiotermillä, koska diffuusi valaistus ei muutu kuvassa.

Heijastuskertoimet p_a ja p_d kaavoissa (6) ja (7) ilmaisevat absorboituneen valon määrän eli aiheuttavat pinnan tummuus/vaaleus vaikutelman. Näiden kertoimien arvot liikkuvat välillä [0,1].

Diffuusissa heijastusmallissa pinnan oletetaan olevan täysin diffuusisti valoa heijastava. Koska todelliset pinnat ovat harvoin täysin diffuusisti valoa heijastavia, tämä rajoittaa mallin käyttöä pyrittäessä realistisiin kuvan generointeihin. Reaaliaikaisissa sovellutuksissa mallia käytetään sen vähäisen laskentakapasiteetin tarpeen takia. Laskentakapasiteetin tarvetta pystytään vielä vähentämään olettamalla valonlähteen olevan äärettömän kaukana. Tällöin pystytään käyttämään vakiollista vektoria \mathbf{L} valonlähteelle kuvan generoinnin aikana.

2.3 Phongin heijastusmalli

Diffuusi heijastus on käyttökelpoinen useissa tapauksissa. Se ei kuitenkaan anna todellista kuvaa materiaalien pinnoista, koska heijastuksesta puuttuu kokonaan peiliheijastuksen aiheuttamat kiiltohuiput materiaalin pinnalla. Cosini -laissa otetaan huomioon ainoastaan valon tulosuunta ja pinnan normaali. Pinnan tarkastelusuuntaa ei oteta, koska sillä ei ole merkitystä Lambertin lain mukaan. Peiliheijastuksen eli spekuläärin heijastuksen vaikutusta määrittäessä tarvitaan katselusuunnan paikkatieto. Phong on esittänyt yksinkertaisen mallin kiillon esittämiseksi materiaalin pinnassa [36].



Kuva 5 Peiliheijastuksen suunta ja jakauma karheilla pinnoilla

Täysin sileissä pinnoissa heijastus tapahtuu ainoastaan kaavan (1) osoittamaan peiliheijastussuuntaan. Karheissa pinnoissa kiiltohuippu leviää täydellisen heijastuskulman ympärille. Phong olettaa peiliheijastuksen jakautuvan peiliheijastuskulman ympärille tasaisesti kosinin potenssin mukaan /Kuva 5/. Phong kirjoittaa heijastusfunktionsa ^[36] seuraavaan muotoon:

$$I = C_p [\cos(i) (1 - d) + d] + W(i) \cos^n(v) \quad (8)$$

jossa

C_p on heijastuskerroin pinnan tietylle pisteelle tietyllä valon aallonpituudella,

i on valon tulokulman ja pinnan normaalin välinen kulma,

v on peiliheijastuskulman ja katselukulman välinen kulma,

$W()$ on funktio, joka antaa spekulaarin heijastuksen kertoimen välillä $[0,1]$,

d on ympäristön diffuusin heijastuksen kerroin ja

n on potenssi, joka kuvaa pinnan kiiltoa eli spekulaarin komponentin leveyttä.

Termi d kaavassa (8) kuvaa ympäristön diffuusia heijastusta tarkasteltavalle pinnalle. Tällä termillä Phong korvaa diffuusissa heijastuksessa (7) käytetyn ambientin, joka estää pinnan mustat kohdat.

Phong olettaa mallissaan valonlähteen olevan äärettömän kaukana, jolloin valon tulokulman voidaan olettaa olevan vakio koko tarkasteltavalle pinnalle. Näin yksinkertaistetaan mallia. $W()$ ja potenssi n kuvaavat pinnan kiilto-ominaisuuksia. Mitä suurempi n :n tai funktion $W()$:n arvo on sitä kiiltävämpi pinta. Täydellisellä peilipinnalla n lähestyy äärettömyyttä ja $W()$ on yksi. Kun tarkastellaan täydellistä peiliheijastuspintaa, spekulaarin heijastuksen termi on nolla muualla kuin peiliheijastussuunnassa, jossa se on yksi.

Phong antaa tyypillisesiksi arvoiksi n :lle 1 ja 10 välillä ja $W()$:lle 0.10 ja 0.80 välillä ^[36]. Nämä arvot ovat empiirisesti haettuja eikä arvoille ole määritelty fysikaalisia perusteita. Dielektrit materiaalit esim. lasi heijastavat vähän valoa takaisin pienillä valon tulokulmilla, kun taas suuret valon tulokulmat aiheuttavat lähes täydellisen heijastumisen. Tästä ilmiöstä johtuu funktion $W()$ riippuvuus valon tulokulmasta.

Yhtälö (8) voidaan kirjoittaa uudestaan pinnan normaalia, valon tulosuuntaa ja katselusuuntaa kuvaavien yksikkövektoreiden avulla /*Kuva 5*/. Järjestämällä kaavan (8) tekijöitä saadaan kaava muotoon, jossa ambientti, diffuusi ja spekulariheijastus on esitetty summan muodossa:

$$I = C_p d + C_p (1 - d) (\mathbf{L} \cdot \mathbf{N}) + W(i) (\mathbf{R} \cdot \mathbf{E})^n \quad (9)$$

jossa

\mathbf{N} on pinnan normaalin yksikkövektori,

\mathbf{L} on valon tulosunnan yksikkövektori,

\mathbf{R} on peiliheijastussuunnan osoittava yksikkövektori ja

\mathbf{E} katselusuunnan osoittava yksikkövektori.

Peiliheijastussuunnan osoittava vektori \mathbf{R} saadaan lasketuksi pinnan normaalin \mathbf{N} ja valon tulosuunnan \mathbf{L} avulla seuraavasta kaavasta

$$\mathbf{R} = \mathbf{N} (2\mathbf{N} \cdot \mathbf{L}) - \mathbf{L} \quad (10)$$

Phongin mallissa oletetaan valonlähteestä tulevan valon olevan valkoista ja sen oletetaan tulevan vakiollisella intensiteetillä yli koko tarkasteltavan pinnan. Kuten monessa muussakin heijastusmallissa tehdään tämä oletus, jotta säästettäisiin laskukapasiteetin tarvetta. Lisäämällä tulevan valon intensiteetin vaikutus ja kirjoittamalla kertoimet toiseen muotoon saadaan kaava (9) seuraavaan muotoon ^[14]:

$$I = p_a I_A + I_L [p_d (\mathbf{L} \cdot \mathbf{N}) + p_s (\mathbf{R} \cdot \mathbf{E})^n] \quad (11)$$

jossa

p_a, p_d ovat valon aallonpituuksista riippuvia kertoimia ja

p_s on valon tulokulmasta riippuvainen kerroin.

Phongin heijastusmallin oletus valonlähteen sijaitsemisesta äärettömän kaukana

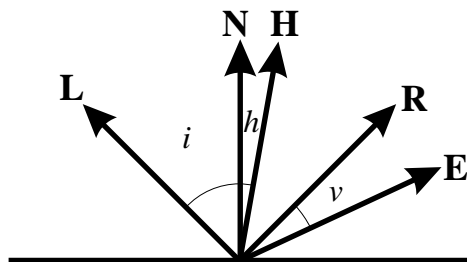
vähentää laskukapasiteetin tarvetta, mutta saatavat kuvat eivät ole realistisia kaikille pinnoille. Oletamus, että valonlähde sijaitsee äärettömän kaukana, ei ole rajoitus diffuusien pintojen visualisoinnissa. Kiiltoa mallitettaessa on valonlähteen sijainnilla suuri vaikutus. Phongin heijastusmallissa kiiltohuiput ovat valkoisia, koska malli olettaa kiiltohuipun heijastavan valon sellaisenaan. Tämän ansiosta Phongin algoritmilla tuotetuilla kuvilla on muovimainen vaikutelma. ^[12]

Phongin mallia käytetään runsaasti kaupallisissa sovellutuksissa sen suhteellisen yksinkertaisuuden johdosta. Reaaliaikaisuutta vaativiin sovellutuksiin on Phongin mallille kehitetty tehokkaita laskenta-algoritmeja ^[2]. Tässä työssä tarkastellaan Phongin mallia yhtenä heijastusmallivaihtoehtona visualisoinnin toteutuksessa.

2.4 Blinnin heijastusmalli

Blinn ^[3] on esittänyt heijastusmallin, joka perustuu Torrance & Sparrow ^[47] ^[48] ja Trowbridge & Reitzin ^[49] töille. Tämä malli vastaa todellisuutta paremmin kuin Phongin heijastusmalli. Blinn mallittaa materiaalin pintaa pienillä mikropinnoilla, jotka oletetaan täydellisesti peiliheijastaviksi pinnoiksi. Näiden mikropintojen kulmien vaihtelun jakauma pinnan normaalin suhteen kuvaa pinnan karheutta. Mitä karheampi pinta sitä leveämpi on mikropintojen kulmien jakauma. Peiliheijastuksen oletetaan muodostuvan näiden mikropintojen ansiosta. Kiinnostava mikropinnan suunta, joka voidaan ilmaista mikropinnan normaalin yksikkövektorilla \mathbf{H} / Kuva 6/, on täydellisen peiliheijastuksen kulma valon tulosuunnan ja tarkastelusuunnan välillä. Jos valo tulee suunnasta \mathbf{L} ja \mathbf{E} on tarkastelusuunta, niin tarkastelusuuntaan nähden kiinnostava mikropinnan normaali \mathbf{H} saadaan yhtälöstä

$$\mathbf{H} = \frac{\mathbf{L} + \mathbf{E}}{|\mathbf{L} + \mathbf{E}|} \quad (12)$$



Kuva 6 Heijastusmallin vektorit

Blinnin muotoilema malli eroaa Phongin mallista peiliheijastustermien osalta. Blinn ^[3] muotoili mallin seuraavaan muotoon:

$$I = p_a I_A + I_L \left[p_d \max(0, \mathbf{N} \cdot \mathbf{L}) + p_s \frac{DGF}{\mathbf{N} \cdot \mathbf{E}} \right] \quad (13)$$

jossa

D on mikropintojen jakaumafunktio,

G on geometriatermi ja

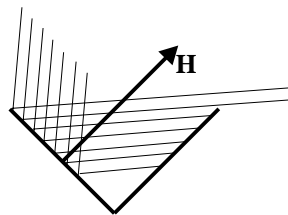
F on Fresnelin heijastuksen kerroin.

Termit D , G ja F vähentävät heijastumisen vaikutusta ja saavat arvoja välillä $[0,1]$. Täydellisellä peiliheijastavalla pinnalla arvot ovat ykkösiä. Mallissa oletetaan valon tulevan pinnan etupuolelta, josta johtuu $\max()$ diffuusissa termissä kaavassa (13). Blinnin heijastusmallissa ei oteta huomioon taittumista eli valon transitiota materiaalin pinnan läpi. [3]

2.4.1 Geometriatermi

Kaavassa (13) pistetulo $\mathbf{N} \cdot \mathbf{E}$ peiliheijastustermissä kertoo pinnan kallistuman katsojaan nähden. Pistetulo on pinnan normaalin yksikkövektorin ja katselusuunnan yksikkövektorin välisen kulman kosini. Mitä enemmän pinta on kallistunut sitä suurempi osa pintaa nähdään tietyllä avaruuskulmalla. Peiliheijastuma on kääntäen verrannollinen termiin $\mathbf{N} \cdot \mathbf{E}$.

Edellistä efektiä heikentää geometriatermi G . Geometriatermillä G pyritään kuvaamaan pinnan luomia varjoja itselleen. Kun geometriatermiä lasketaan, mallissa oletetaan mikropintojen muodostuvan V :n muotoisista vaoista. Näissä vaoissa mikropinnat ovat toisiaan vastaan ja molemmat vaon reunat ovat yhtä pitkiä ja kulmat pinnan normaaliin \mathbf{N} nähden ovat vastakkaissuuntaiset /Kuva 7/. Vaot, joiden toisen sivun normaali on \mathbf{H} :n suuntainen, ovat kiinnostavia, koska näistä pinnoista tapahtuu peiliheijastus katselusuuntaan. [3] [8] [12]



Kuva 7 V :n muotoinen mikropinta ja siitä heijastuva valo^[3]

Kun lasketaan geometriatermiä G , oletetaan vaosta tulevan valon heijastuvan vain kerran mikropinnan seinämästä. \mathbf{H} :n suuntainen vaon seinä joko 1) heijastaa kaiken siihen tulevan valon tai 2) estää osan heijastuneen valon poistumisen vaosta varjostamalla tai 3) estää osan tulevan valon saapumisen vakoon maskaamalla. Näistä kolmesta termistä voidaan laskea geometriatermi G vaikuttava ominaisuus minimoimalla. Kun kaikki valo heijastuu vaosta, G :n arvo on yksi. Maskaaminen ja

varjostaminen pystytään laskemaan, kun projisoidaan tarkastelusuunnan osoittava vektori \mathbf{E} \mathbf{N} :n ja \mathbf{H} :n muodostamalle tasolle. Ongelmaa voidaan tällöin tarkastella kahdessa dimensiossa /Kuva 8/. Tässä voidaan todeta, että jos vaihdetaan tarkastelusuunnan ja tulevaa valoa osoittavien vektoreiden paikkaa, voidaan ongelma ratkaista analogisesti sekä valon varjostukselle että pinnan maskaukselle. ^{[3] [12]}

Valon määrä, joka heijastuu projisoituun tarkastelusuuntaan \mathbf{E}_p , joka on yksikkövektori, voidaan laskea m :n ja l :n suhteiden avulla /Kuva 8/ seuraavasti:

$$G = 1 - \frac{m}{l} \quad (14)$$

Kaavan (14) suhde pätee myös tarkastelusuuntaan \mathbf{E} . Kuvasta 8 saadaan kulmien avulla lausuttua m :n ja l :n suhde

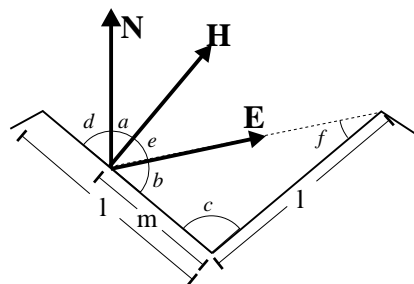
$$\frac{m}{l} = \frac{\sin(f)}{\sin(b)} \quad (15)$$

Huomattakoon, että trigonometriaa hyväksi käyttäen pystymme lausumaan kulmat b ja f muiden kulmien avulla

$$\sin(b) = \cos(e) \quad (16)$$

$$\cos(b) = \sin(e) \quad (17)$$

Yhtälöt (16) ja (17) saadaan suoraan johdettua, kun tiedetään kulmien e :n ja b :n summan olevan suorakulma /Kuva 8/.



Kuva 8 Geometriatermin laskeminen ^{[3] [12]}

Kulma f voidaan lausua kulmien b ja c avulla

$$\sin(f) = \sin(b+c) = \sin(b)\cos(c) + \cos(b)\sin(c) \quad (18)$$

Vaon symmetrisyyden perusteella voidaan kulma c lausua kulman d avulla, koska $c = 2d$. Vastaavasti kulma d voidaan lausua kulman a avulla, koska $d = \pi/2 - a$.

$$\cos(c) = 1 - 2\sin^2(d) = 1 - 2\cos^2(a) \quad (19)$$

$$\sin(c) = 2\cos(d)\sin(d) = 2\sin(a)\cos(a) \quad (20)$$

Kun sijoitetaan yhtälöt (16), (17), (19) ja (20) yhtälöön (18), saadaan $\sin(f)$ lausuttua seuraavassa muodossa:

$$\begin{aligned} \sin(f) &= \cos(e)(1 - \cos^2(a)) \\ &+ 2\sin(e)\cos(a)\sin(a)\sin(b)\cos(c) \\ &= \cos(e) - 2\cos(a)(\cos(e)\cos(a) - \sin(e)\cos(a)) \\ &= \cos(e) - 2\cos(a+e) \end{aligned} \quad (21)$$

Todetaan kahden yksikkövektorin välisen kulman cosinin olevan yhtäsuuri kuin yksikkövektoreiden pistetulo, jolloin yhtälöissä (21) johdetut tulokset saadaan seuraavaan muotoon, kun kulmia vastaavat vektorit ovat kuvassa 8 olevat vektorit ^[3]:

$$\sin(f) = \mathbf{H} \cdot \mathbf{E}_p - 2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{E}_p) \quad (22)$$

Koska \mathbf{E}_p on \mathbf{E} projektio \mathbf{N} :n ja \mathbf{H} :n määrittämälle tasolle, on $\mathbf{N} \cdot \mathbf{E}_p = \mathbf{N} \cdot \mathbf{E}$ ja $\mathbf{H} \cdot \mathbf{E}_p = \mathbf{H} \cdot \mathbf{E}$. Geometria termi G voidaan lausua edellä johdettujen tulosten perusteella, jotka sijoitetaan yhtälöön (14):

$$G = \frac{2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{E})}{(\mathbf{E} \cdot \mathbf{H})} \quad (23)$$

Yhtälö (23) ottaa huomioon ainoastaan vaon varjostuksen. Maskaus saadaan laskettua analogisesti varjostukseen nähden, koska tällöin tarkastellaan vektorin \mathbf{E} sijasta vektoria \mathbf{L} ja saadaan kuvaan 8 nähden analoginen kuvio ja analoginen yhtälö maskauksen laskemiseksi. Kun vielä otetaan huomioon mahdollisuus, että kaikki tuleva valo

voi heijastua mikropinnasta, saadaan yhtälö geometriatermin laskemiseksi^[3]

$$G = \min\left(1, \frac{2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{E})}{(\mathbf{E} \cdot \mathbf{H})}, \frac{2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{L})}{(\mathbf{E} \cdot \mathbf{H})}\right) \quad (24)$$

On huomattava, että kaavassa (24) jakaja $\mathbf{E} \cdot \mathbf{H} = \mathbf{L} \cdot \mathbf{H}$, koska \mathbf{H} on määritetty puolittamaan \mathbf{E} :n ja \mathbf{L} :n välinen kulma kaavassa (12). Tästä johtuu että voidaan kirjoittaa jakajat kaavassa (24) samaan muotoon. ^{[3] [12]}

2.4.2 Fresnelin heijastuksen määrittäminen

Termi F kaavasssa (13) tarkoittaa Fresnel-heijastumista materiaalista. Fresnel termi ennustaa, kuinka paljon tulevasta valon intensiteetistä heijastuu pinnasta ja kuinka paljon taittuu. Kaavojen (3), (4) ja (5) avulla pystytään johtamaan Blinnin ^[3] käyttämä Fresnel yhtälö, jossa lasketaan heijastuminen ympyräpolarisoituneelle valolle seuraavasti:

$$F = \frac{1}{2} \left(\frac{\sin^2(i - x)}{\sin^2(i + x)} + \frac{\tan^2(i - x)}{\tan^2(i + x)} \right) \quad (25)$$

jossa

$$\sin(x) = \sin(i)/n,$$

i on tulevan valonsäteen ja mikropinnan normaalin välinen kulma ja

n on pinnan materiaalin taitekerroin.

Kaava (25) voidaan johtaa edelleen laskennallisesti tehokkaampaan muotoon:

$$F = \frac{1}{2} \frac{(g - c)^2}{(g + c)^2} \left\{ 1 + \frac{[c(g + c) - 1]^2}{[c(g - c) + 1]^2} \right\} \quad (26)$$

jossa

$$c = \cos(i) = \mathbf{L} \cdot \mathbf{H} \text{ ja}$$

$$g^2 = n^2 + c^2 - 1.$$

Kaavoissa (25) ja (26) on kuitenkin puutteena materiaalikerroin. Tietokonegrafiikassa kuvataan usein materiaaleja, jotka eivät ole todellisia tai joiden ominaisuuksia ei

tunneta tarkoin. Tämän vuoksi tarvitaan tapa jolla voidaan approksimoida materiaalin taitekerrointa.

Cook ^[7] on esittänyt tavan aproksimoida materiaalin taitekerrointa. Kun oletetaan, että tarkasteltavan valon tulosuunta on sama kuin tarkastelusuunta eli $\mathbf{L}=\mathbf{E}=\mathbf{H}$ ja väliaine jonka läpi valo kulkee on ilma, jolloin taitekerroin n_i kaavoissa (3) ja (4) voidaan pyöristää yhdeksi. Kaavat (3), (4) ja (5) supistuvat tällöin muotoon

$$F = \frac{(n-1)^2}{(n+1)^2} \quad (27)$$

Tämä kaava pätee dielektrisille materiaaleille. Kaava (27) voidaan ratkaista materiaalin taitekertoimen n suhteen:

$$n = \frac{1 + \sqrt{F}}{1 - \sqrt{F}} \quad (28)$$

Kaavan (28) puutteena on termi F , joka pitää määrittää esimerkiksi mittaamalla.

Kaavalla (28) saadaan taitekerroin yhdelle spektrin aallonpituudelle kerrallaan. Loput aallonpituudet saadaan laskettua vastaavasti, jos on käytettävissä valmis spektriaineisto termille F . Tietokonegrafiikassa yksinkertaistetaan usein materiaalin taitekerroin laskemalla kaikkien aallonpituuksien taitekertoimien keskiarvo. Tämä saadaan lasketuksi helposti kaavasta (28) sijoittamalla spektritermin F paikalle keskiarvoheijastus F_{ave} :

$$F_{ave} = \frac{1}{b-a} \int_a^b F(x) dx \quad (29)$$

jossa

a, b ovat tarkasteltavan spektrin ääriarvojen aallonpituudet.

Toisaalta materiaalin taitekertoimen lisäksi tarvitaan heijastusspektri, jonka approksimointiin ja laskemiseen Cook ^[7] esitti tavan:

$$F_{\lambda}(i) = F_{\lambda}(0) - \left[F_{\lambda}\left(\frac{\pi}{2}\right) - F_{\lambda}(0) \right] \left[\frac{F_{ave}(i) - F_{ave}(0)}{F_{ave}\left(\frac{\pi}{2}\right) - F_{ave}(0)} \right] \quad (30)$$

jossa

λ on tarkasteltava aallonpituus ja

i on tarkastelukulma

Kaava (30) supistuu, koska kulman ollessa $\Pi/2$ eli tarkastellaan kulmaa, jossa tapahtuu grazing ilmiö, heijastusermit F_{λ} ja F_{ave} saavat arvokseen yksi. Termi $F_{ave}(0)$ on määritetty suoraan kaavassa (29) ja eri aallonpituuksille annettu nollakulma-arvo $F_{\lambda}(0)$ on käyttäjän määritettävä. Tarkasteltavalle kulmalle määritetty keskiarvoheijastus $F_{ave}(i)$ saadaan laskettua kaavoista (3), (4) ja (5) tai (25) tai (26), joihin materiaalin taitekertoimeksi on sijoitettu kaavojen (28) ja (29) avulla laskettua taitekertoimen keskiarvo.

2.4.3 Distribuutiofunktio

Distribuutiofunktio kuvaa mikropintojen jakaumaa. Phongin heijastusmallissa ^[36] distribuutio on periaatteessa cosinin potenssi kuten kaavassa (8):

$$D(h) = \cos^{c_1}(h) \quad (31)$$

jossa

c_1 on potenssi, joka kuvaa pinnan kiiltoa ja

h on pinnan normaalin ja mikropinnan normaalin välinen kulma.

Phong ^[36] esittää omassa mallissaan peiliheijastumisen kulman v (8) avulla, joka on katselusuunnan ja peiliheijastussuunnan välinen kulma, kun taas Blinn ^[3] esittää pinnan normaalin ja mikropinnan normaalin välisenä kulmana h , joka on puolet kulmasta v . Tämä ei kuitenkaan muuta jakauman muotoa, vaan lähinnä nopeuttaa laskutoimituksia, koska ei tarvitse laskea peiliheijastussuunnan yksikkövektoria \mathbf{R} . Torrance & Sparrow ^[48] käyttävät gaussista jakaumaa distribuutiofunktiona:

$$D(h) = e^{-c_2^2 h^2} \quad (32)$$

jossa

c_2 on mikropintojen vaihtelun varianssi.

Suuret c :n arvot vastaavat karheita mattapintoja ja pienet arvot kiiltäviä pintoja. Trowbridge & Reitz ^[49] osoittivat, että yleinen joukko pintoja voidaan mallittaa pyörähdysellipsoideilla

$$D(h) = \left(\frac{c_3^2}{\cos^2(h(c_3^2 - 1)) + 1} \right)^2 \quad (33)$$

jossa

c_3 on ellipsoidin epäkeskisyyys.

Täysin ideaalisille peiliheijastus pinnoille c_3 on nolla ja diffuuseille pinnoille c_3 on yksi. Jokaisella näistä distribuutiofunktioista on maksimi yksi, kun h on nolla. Kun \mathbf{H} on pinnannormaalien \mathbf{N} suuntainen ja tarkastelusuunta \mathbf{E} on peiliheijastussuunnan \mathbf{R} kanssa yhdensuuntainen.

Edellä esitetyt kaavat (31), (32) ja (33) kuvaavat mikropintojen jakaumaa. Mikro-pinnoilla pyritään mallittamaan speulaaria heijastusta. Distribuutiofunktio voi saada jakauman, jonka arvo on yksi, kun tarkastellaan peiliheijastussuuntaa ja muualla jakauman arvo on nolla. Tällaiset jakaumat kuvaavat täydellistä peiliheijastuspintaa. Kun mikropintojen jakauma saa arvon nolla kaikilla tarkastelukulmilla, tarkasteltava pinta heijastaa valon täysin diffuusisti.

Hyvin pinnan kiiltoa kuvaa betakulma h' , jossa peiliheijastuksen intensiteetti putoaa puoleen. Kertoimien c_1 , c_2 ja c_3 puolen intensiteetin arvot, kun h' on vektorin \mathbf{H} kulma pinnan normaaliin \mathbf{N} nähden puolen intensiteetin kohdalla, saadaan seuraavista kaavoista:

$$c_1 = -\frac{\ln 2}{\ln(\cos(h'))} \quad (34)$$

$$c_2 = \frac{\sqrt{\ln 2}}{h'} \quad (35)$$

$$c_3 = \sqrt{\frac{\cos^2(h') - 1}{\cos^2(h') - \sqrt{2}}} \quad (36)$$

Suurin ero Blinnin ^[3] mallissa ja Phongin ^[36] mallissa syntyy, kun kappaletta valaistaan sivulta. Tällöin Blinnin heijastusmallia käytettäessä syntyy voimakas spekulari-heijastus, joka johtuu grazing ilmiöstä.

Blinn ^[3] olettaa mallissaan peiliheijastuksen olevan valon lähteen värinen kuten Phongkin. Tämä saa aikaan generoituun pintaan muovimaisen vaikutelman. Cook ja Torrance ^[7] paransivat heijastusmallia lisäämällä pinnan värin vaikutuksen peili-heijastukseen. Näin saadaan mallitettua metallisia pintoja.

Muita, jotka ovat esittäneet parannuksia heijastusmalleihin Torrance & Sparrow:n ^[47] ^[48] ja Blinnin ^[3] työn pohjalta, ovat Greenberg ^[11] ja Rushmeier ja Torrance ^[40], jotka ovat laajentaneet heijastusmallin käsittelyä radiocity¹ algoritmeihin. Blinn ^[5] on kehittänyt heijastusmallin käsittelyä myös pilvimäisille esineille sekä pölyisille pinnoille. Tässä heijastusmallissa oletetaan pinnan muodostuneen pienistä pallon-muotoisista partikkeleista, jotka ovat jakautuneet tasaisesti tiettyyn tilavuuteen.

Blinnin ^[3] käsittelemät distribuutiofunktiot ovat symmetrisiä. Heijastusmallia voi laajentaa mallittamaan anisotrooppisia pintoja, joissa näkyy esimerkiksi työstö-suunnan jättämät vaikutukset pinnan heijastusominaisuuksiin. Esimerkkejä anisotrooppisen heijastuksen käsittelystä ovat esittäneet esimerkiksi Puolin & Four-nier ^[37], jotka olettavat anisotrooppisen pinnan muodostuvan pienistä sylintereistä, jotka kontrolloivat mikropintojen distribuutiota. Yasuda, Yokoi, Toriwaki & Inagaki ^[56] ovat mallittaneet kankaan pintaa muodostamalla kankaan struktuurista mallin ja toisen mallin mikropinnoille eli distribuutiofunktion. Ward ^[50] pyrkii mittaamaan eri pintojen heijastuksia.

Realistisen visualisoinnin ongelmana on määrittää kertoimia ja potensseja eri mallien parametreille, jotta ne vastaisivat haluttua lopputulosta. Tämä on yleensä suoritettu kokeellisesti iteroimalla tulos halutun näköiseksi. Valmiita kirjastoja eri materiaaleille ei ole kehitetty.

Tässä työssä tarkastellaan Phongin heijastusmallin lisäksi Blinnin heijastusmallia. Blinnin heijastusmalli on valittu, koska se pyrkii mallittamaan todellista valon heijastumista.

2.5 Pintastruktuurin kuvaaminen tekstuurin avulla

Tekstuuri on värin ja/tai karheuden säännönmukaista vaihtelua pinnalla. Kappaleen pintaan voidaan liittää haluttu pintakuvionti tekstuurin avulla ^[6] ^[39]. Näin vältetään kappaleen liian tarkka mallittaminen. Tekstuurin liittäminen on erikseen määritellyn

1. Radiocity algoritmit laskevat pinnan energiamäärää, jonka perusteella määritetään valo-energian säteily. ^[12]

kuvion maalaamista kappaleen pintaan. Käytännössä tämä voidaan suorittaa koordinaatistomuunnoksella tekstuurin koordinaatistosta kappaleen koordinaatistoon. Jos \mathbf{x} on kappaleen avaruutta osoittava koordinaatistovektori ja \mathbf{x}' on tekstuurin avaruutta osoittava koordinaatistovektori, voidaan kappaleen pinnalle kohtaan \mathbf{x} hakea oikea tekstuurielementti kohdasta \mathbf{x}' seuraavalla koordinaatistomuunnoksella:

$$\mathbf{x}' = f(\mathbf{x}) \quad (37)$$

jossa

$f()$ on koordinaatiston muunnosfunktio, jolla voidaan muuttaa koordinaatisto kolmesta dimensiosta kahteen tai kolmeen dimensioon.

Kun tekstuuri lisätään tasaiseen pintaan, se ei muuta pinnan muotoa vaan lisää pintakuviointin kappaleen pintaan. Tekstuuri voidaan myös määrittää kolmiulotteiseksi ^[35], jolloin tekstuuripisteen hakeminen tapahtuu teksturiavaruuden tietystä pisteestä.

Karhean pinnan tekstuuria voidaan mallittaa valokuvaamalla karhea pinta ja digitoimalla se tekstuuriksi. Kun karheutta kuvaava tekstuuri liitetään kappaleeseen, siihen saadaan karheutta simuloiva pintakuvio. Tällainen karheus näyttää kuitenkin samalta vaikka tarkastelusuuntaa tai valonlähteen suuntaa muutettaisiin. Tämä johtuu siitä, että karhea kuviointi ei muuta pinnan normaalin suuntaa, jolloin saataisiin valoilmiöt näkymään erillisinä eri suunnista tarkasteltuna.

Blinn ^[4] esitti metodin, jolla voidaan muodostaa karhea pintakuviointi kappaleen pinnalle. Kuviointi luodaan muuttamalla pinnan normaalin suuntaa tekstuurin aiheuttaman häiriön mukaisesti, jolloin saadaan valon vaikutukset näkyviin.

Minkä tahansa pinnan pisteen $Q(u, w)$ osittaisderivaatat Q_u ja Q_w pinnan parametrien u :n ja w :n suhteen määräävät pinnan $Q(u, w)$ tangentiaalitason, jonka normaali saadaan osittaisderivaattojen ristitulona

$$\mathbf{N} = \mathbf{Q}_u \otimes \mathbf{Q}_w \quad (38)$$

jossa

$\mathbf{Q}_u, \mathbf{Q}_w$ ovat pinnan $Q(u, w)$ osittaisderivaattojen suuntaa osoittavia vektoreita u :n ja w :n suhteen. Huomattava on, että \mathbf{N} ei ole välttämättä yksikkövektori.

Blinn määritteli häiriöfunktion $P(u, w)$, jonka vaikutus pintaan $Q(u, w)$ saadaan

$$Q'(u, w) = Q(u, w) + P(u, w) \mathbf{N} \quad (39)$$

jossa

\mathbf{N} on pinnan $Q(u, w)$ yksikkönormaalivektori.

Tällöin tekstuuri-informaation omaavan pinnan normaali saadaan laskettua $Q'(u, w)$ osittaisderivaattojen avulla:

$$\mathbf{N}' = \mathbf{Q}_u' \otimes \mathbf{Q}_w' \quad (40)$$

jossa osittaisderivaatat \mathbf{Q}_u' ja \mathbf{Q}_w' ovat tekstuuriittomien osittaisderivaattojen ja pinnan normaalin avulla lausuttuna seuraavia:

$$\begin{aligned} \mathbf{Q}_u' &= \mathbf{Q}_u + P_u \mathbf{N} + P \mathbf{N}_u \\ \mathbf{Q}_w' &= \mathbf{Q}_w + P_w \mathbf{N} + P \mathbf{N}_w \end{aligned} \quad (41)$$

jossa

P_u, P_w ovat häiriöfunktio P :n osittaisderivaattoja u :n ja w :n suhteen ja

\mathbf{N} on pinnannormaalin yksikkövektori.

Koska häiriöfunktio P on pieni, voidaan yhtälöissä (41) jättää viimeiset termit huomiotta, jolloin saadaan (41) muotoon

$$\begin{aligned} \mathbf{Q}_u' &= \mathbf{Q}_u + P_u \mathbf{N} \\ \mathbf{Q}_w' &= \mathbf{Q}_w + P_w \mathbf{N} \end{aligned} \quad (42)$$

Tällöin tekstuuri-informaation omaava pinnan normaalivektori saadaan yhtälöiden (40) ja (42) avulla:

$$\mathbf{N}' = \mathbf{Q}_u \otimes \mathbf{Q}_w + P_u (\mathbf{N} \otimes \mathbf{Q}_w) + P_w (\mathbf{Q}_u \otimes \mathbf{N}) + P_u P_w (\mathbf{N} \otimes \mathbf{N}) \quad (43)$$

jossa

\mathbf{N} on pinnannormaalin yksikkövektori,

$P_u P_w$ ovat häiriöfunktio P :n osittaisderivaattoja u :n ja w :n suhteen ja

$\mathbf{Q}_u, \mathbf{Q}_w$ ovat pinnan $Q(u, w)$ osittaisderivaattojen suuntaa osoittavia vektoreita u :n ja w :n suhteen.

Viimeinen termi kaavassa (43) voidaan supistaa pois, koska se on identtisesti nolla. Ensimmäinen termi kaavassa (43) on häiriötön pinnan normaalivektori. Tällöin voidaan kaava (43) kirjoittaa muotoon

$$\mathbf{N}' = \mathbf{N} + \mathbf{D} \quad (44)$$

jossa \mathbf{D} saadaan seuraavasta kaavasta:

$$\mathbf{D} = P_u (\mathbf{N} \otimes \mathbf{Q}_w) - P_w (\mathbf{N} \otimes \mathbf{Q}_u) \quad (45)$$

Häiriöfunktio $P(u, w)$ on helppo määrittää esimerkiksi matriisina, jolloin saadaan tietty toistuva pintakuvio kappaleen pintaan. Tällä tavalla määritetty häiriöfunktio ei kuitenkaan muuta kappaleen tilavuus- ja reunainformaatiota. Se näkyy esimerkiksi, kun pyöreään kappaleeseen liitetään karheustekstuuri, reuna pysyy pyöreänä eikä siihen tule pintakuvioinnin mukaista epätasaisuutta.

2.6 Valaistuksen mallittaminen

Edellä esitetyissä heijastusmalleissa ei ole otettu huomioon valonlähdettä sen todellisiin ominaisuuksiin perustuen, vaan valonlähdettä on käytetty yksinkertaistettuna mallina. Heijastusmalleissa on oletettu valonlähteen olevan pistemäinen. Valonlähde sijoitetaan yleensä äärettömän kauaksi laskujen helpottamiseksi. Tällä saavutetaan se etu, että valonlähteen suuntaa kuvaava vektori voidaan olettaa vakiolliseksi.

Todellisilla valonlähteillä on jokin koko ja muoto tarkasteltavassa avaruudessa. Mahdollisimman tarkka valonlähteen mallittaminen edellyttäisi integrointia jonkin avaruuskulman ylitse, kun määritetään yhden pisteen saamaa valoenergiaa. Integroinnin raskaudesta johtuen mallinnetaan todellisia valonlähteitä usealla pistemäisellä valonlähteellä kuten Whitted^[53] laskemalla usean pistemäisen valonlähteen vaikutus yhteen. Tämä malli voidaan yksinkertaistaa yhteen pistemäiseen valonlähteeseen, jonka intensiteetti ja sijainti on laskettu todellisesta valonlähteestä. Etäisyyden

huomioiminen voidaan hoitaa lisäämällä etäisyyden käänteislukukerroin valaistukseen.

Warn ^[51] suuntasi valonlähteen vaikutusta kertomalla cosini-termillä valon vaikutuksen. Cosini-termi muodostui tietyn valaistuksen suunnan ja lasketun suunnan välisen kulman cosinista. Warn määrittä valonlähteen suuntaavuuden vastaavanlaisella termillä kuin Phong (8) määritti spekulaaarin heijastuksen:

$$I_L = I \cos^n(l) \quad (46)$$

jossa

I on valolähteen intensiteetti valolähteen todelliseen suuntaan,

l on valonsäteen -L ja valolähteen todellisen suunnan välinen kulma ja

n on potenssi, joka määrää valolähteen suuntaavuuden.

Monimutkaisempia valonlähteiden mallituksia käsittelee esimerkiksi Houle & Fiume^[18]. Tässä työssä käytetään kuitenkin pistemäistä valonlähdettä sen yksinkertaisuuden ja tehokkuuden vuoksi.

2.7 Värin mallittaminen

Valon ja kappaleen värillisyyttä ei ole varsinaisesti huomioitu edellä esitetyissä malleissa (6), (7), (11) ja (13). Näiden mallien avulla lasketaan tarkasteltavan pisteen intensiteetti. Kappaleen värillisyydestä voidaan tallentaa kertoimiin p_a , p_d ja p_s muuttamalla ne vektoreiksi, joiden alkioiden arvot kuvaavat RGB arvoja tai vastaavasti jonkun muun esitystavan arvoja. Jos käytetään RGB arvoja Phongin ja Blinnin mallien kertoimina, saadaan tulokseksi (3x1) RGB-vektori. Näillä kerroinvektoreilla kuvataan pinnan värillisyyden muodostavat ominaisuudet. Vastaavasti voidaan tulevan valon intensiteetin värillisyyden tallentaa vektoriin, jossa on valon värillisyyden arvot.

Tässä työssä värillisyyden mallinnetaan spektrivektorien avulla. Spektrivektorit sijoitetaan heijastusmallien kertoimien p_a , p_d ja p_s sekä valonlähteen intensiteettivektoreiksi I_A ja I_L . Tarkasteltava spektri on näkyvän valon aallonpituusalueella 380 nm - 780 nm. Spektrivektori on muodostettu spektristä nanometrin askeleen välein.

3 PAINOJÄLJEN VISUALISOINNISSA TARKASTELTAVAT MATERIAALIT

Perinteisesti painojälki käsitetään mekaanisesti paperille painettuna kuvana ja/tai tekstinä, esimerkiksi sanomalehdet ja viikkolehdet. Tällaiset tuotteet tehdään offset-, flekso- tai syväpainomenetelmillä. Näissä painojälki muodostetaan painavalla pinnalla, joka sisältää kuvien ja tekstin painamista varten tarvittavan informaation. Näillä painavilla pinnoilla kuljetetaan painoväri paperille. Ominaista perinteisissä menetelmissä on painopinnan kertakäyttöisyys. Tämä tarkoittaa sitä, että painavalla pinnalla pystytään painamaan yhdenlaista tuotetta useita kappaleita eli kun vaihdetaan painettava tuote joudutaan vaihtamaan myös uudet painopinnat.

Uusina menetelminä ovat tulleet niin sanotut “Non Impact Printing” -menetelmät (NIP), joissa kuvat ja teksti muodostetaan ilman painavia pintoja. Painoväri kuljetetaan esimerkiksi sähköisten varausten avulla oikeisiin kohtiin. Ominaista näille menetelmille on tuotteiden pienet painosmäärät, painamisen joustavuus ja nopea siirtyminen uuden tuotteen painamiseen. Tällaisia painomenetelmiä ovat esimerkiksi lämpö-, mustesuihku-, matriisi- ja elektrofotograafiset tulostimet. Osa toimii ainoastaan erikoisvalmisteisilla painoalustoilla, osa toimii “tavallisella” paperilla.

Painoalustojen suhteen löytyy muitakin vaihtoehtoja kuin paperi. Muita mahdollisia painoalustoja ovat esimerkiksi tekstiilit, joista löytyy jokapäiväisestä elämästä runsaasti esimerkkejä. Oma kategoriansa on erilaisten pakkausmateriaalien painaminen.

Tässä työssä keskitytään painojälkeen, joka on tuotettu paperille perinteisillä menetelmillä. Tarkastelu voidaan kuitenkin helposti laajentaa NIP-menetelmien puolelle valitsemalla sopivat painojäljen muodostuksen malli ja sen parametrit. Painojäljen visualisoinnin tarkasteltavat materiaalit ovat paperi ja painoväri, erityisesti näiden heijastusominaisuuksiin vaikuttavat tekijät. Painojäljen suhteen rajoitutaan yksivärisiin painoväripintoihin ja lähemmin vielä kompakteihin painoväripintoihin. Näin ollen tarkastelusta voidaan jättää pois valon heijastumiset eri painovärikerrosten välillä ja keskittytään valon heijastumisiin ja taittumisiin ilman ja painovärin tai paperipinnan välillä.

3.1 Paperi

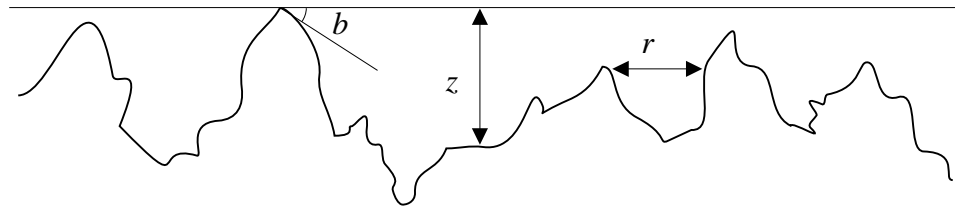
Paperin tarkastelussa keskitytään paperin karheuteen ja heijastusominaisuuksiin. Paperin karheus sijoittuu osaksi alueelle, jossa tapahtuu valon diffraktioilmiöitä, mutta nämä ominaisuudet jätetään tarkastelusta pois. Pienimmät tässä yhteydessä tarkasteltavat paperin karheusominaisuudet ovat mikrometrien luokkaa, koska näkyvän valon aallonpituus on 380 nm:n ja 780 nm:n välillä.

3.1.1 Paperin karheusmalli

Paperi muodostuu kemiallisesti tai mekaanisesti valmistetuista puukuiduista, joiden pituus on noin 3 mm ja leveys noin 30 μm suuruusluokkaa ^[41]. Paperissa olevien kuitujen koko riippuu suuresti paperimassan tekotavasta. Lopullisen karheuden määrävät täyteaine- ja päälystysainepigmentit sekä paperin jälkikäsittely. Paperin tasaisuuteen ja kiiltoon vaikuttaa kalanterointi ja sen kertojen lukumäärä sekä erilaiset

päällystykset. Paperin karheuden ominaisuudet määräytyvät paperin tekotavan mukaan. Tässä työssä tarkasteltava karheus muodostuu lähinnä partikkeleiden pakkautumisesta, kuitukarheudesta ja kuitujen koosta. Karheudella on tietty orientaatio, jos kuitujen on annettu järjestyä valmistuksen aikana.

Painojäljen visualisoinnin kannalta keskeisin ominaisuus on paperin karheus. Paperin pinnat ovat luonteeltaan stokastisia. Jos tarkastellaan kahta paperin pintaa, pinnat eivät ole täysin samanlaisia, vaikka valmistusprosessi on sama molemmille pinnoille. Jotta voitaisiin kuvata tällaisia karheita pintoja, käytetään hyväksi tilastollisia menetelmiä. Seuraavassa tarkastellaan karheuden eri ominaisuuksia, joita voidaan paikallisesti määrittää karheuden syvyys z , epätasasuuksien leveys r ja kaltevuus b tasaiseen referenssipintaan nähden /Kuva 9/.^[30]



Kuva 9 Karheusprofiilin ominaisuuksia^[30]

Tilastollinen pinnankarheus voidaan määrittää karheuden syvyyden z ja epätasaisuuksien leveyden avulla. Tällöin voidaan luonnehtia pinnalle yhteisjakauma $p(r, z_0, z)$, joka ilmoittaa todennäköisyyden sille, että kaksi pinnan pistettä, joiden etäisyys toisistaan on r ovat syvyyksien z_0 ja z välillä. Yhteisjakauman $p(r, z_0, z)$ avulla voidaan määrittää korkeuden todennäköisyysjakauma $p(z)$ ^[30]

$$p(z) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(r, z_0, z) dr dz_0 \quad (47)$$

Korkeusjakaumalle $p(z)$ voidaan määrittää erilaisia tunnuslukuja, joita löytyy lähteistä^[29] ja^[30]. Lauseke $p(z)dz$ ilmaisee todennäköisyyden sille, että korkeus sijaitsee z ja $z+dz$ korkeuksien välissä, jolloin saadaan keskisyvyyden odotusarvo ilmaistua syvyyden z ja todennäköisyyden $p(z)$ avulla^[30]

$$\bar{z} = \int_{-\infty}^{\infty} zp(z) dz = 0 \quad (48)$$

Tarkasteltava pinta voidaan kuvata valitsemalla referenssitaso aina niin että edellinen

yhtälö (48) käy toteen. Kun tarkastellaan karheaa pintaa, ei niinkään tarkastella pinnan keskisyvyyden arvoa vaan muita parametrejä, jotka kuvaavat paremmin karheutta. Varianssi eli *rms* -arvo¹ jakaumalle $p(z)$ saadaan määritettyä seuraavasti [30]

$$rms^2 = \int_{-\infty}^{\infty} (z - \bar{z})^2 p(z) dz \quad (49)$$

Pinnan karheutta kuvaava parametri saadaan esille aritmeettisen keskiarvon avulla, jos todennäköisyyden $p(z)$ odotusarvo on nolla.

$$z_a = \int_{-\infty}^{\infty} |z| p(z) dz \quad (50)$$

Parametri z_a pystytään ilmaisemaan varianssin avulla, jos tunnetaan korkeuden z todennäköisyys funktio $p(z)$. Tarkasteltavan pinnan maksimisyvyys z_{max} saadaan määritettyä maksimin ja minimin avulla seuraavasti:

$$z_{max} = \max(z) - \min(z) \quad (51)$$

Lisäksi voidaan määrittää maksimi- ja minipisteen välinen xy-tason suuntainen etäisyys. Kun tarkasteltavan paperipinnan päälle asetetaan referenssipinta, jollaiseksi voidaan ajatella painamistilannetta, halutaan tietää paperipinnan karheuden ja referenssipinnan väliin jäävä tilavuus. Tämä voidaan laskea seuraavasti yli tarkasteltavan pinnan: [30]

$$V = \int z^3 p(z) dz \quad (52)$$

Tässä on tarkasteltu pinnan korkeuden suhteen karheuden ominaisuuksia. Tarkastelu voidaan suorittaa myös pinnan etäisyyksien perusteella. Pinnan etäisyyksille voidaan myös määrittää todennäköisyysfunktio

¹.rms = “Root Mean Square”

$$p(r) = \int_{-\infty}^{\infty} p(r, z_0 = z, z) dz \quad (53)$$

Kaltevuusjakauma saadaan laskettua seuraavasti:

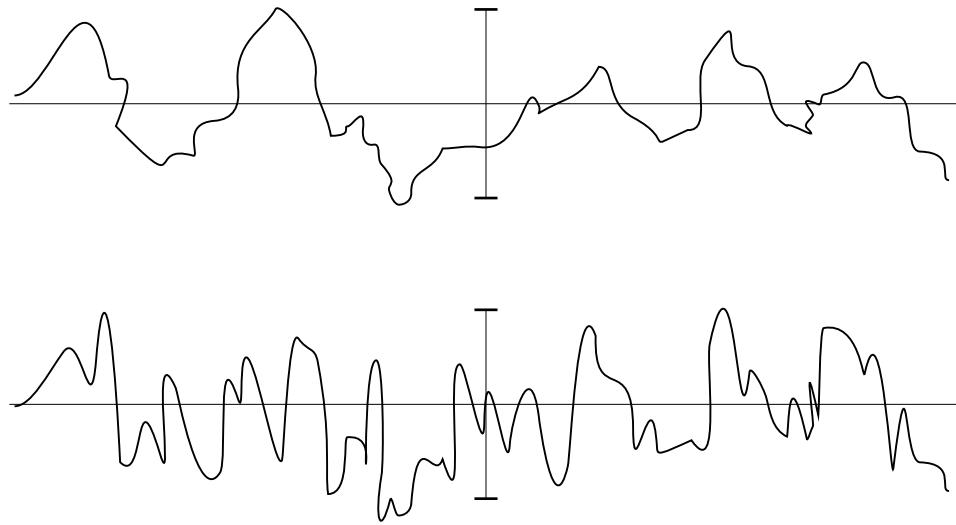
$$p(\tan\beta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(r, z_0 = (z - r \tan\beta), z) dr dz \quad (54)$$

Korkeusjakauma $p(z)$ ja kaltevuusjakauma $p(\tan\beta)$ ovat luonteeltaan gaussisia tai vastaavan tyyppisiä jakaumia. Tämä tarkoittaa, että on jokin odotusarvo, jonka ympärillä tapahtuu vaihtelua. Leveysjakauma $p(r)$ on luonteeltaan monotonisesti vähenevä funktio eli mitä suurempi etäisyys, sitä pienempi todennäköisyys.

Kun tarkastellaan syvyyden *rms*-arvoa, saadaan erilaisille pinnoille sama *rms*-arvo, vaikka intuitiivinen profiilin tarkastelu osoittaa, että pinnat ovat erilaisia /*Kuva 10*/. Autokorrelaatiolla saadaan esiin pintojen rakenne. Autokorrelaatiofunktio määritetään seuraavasti:

$$A(r) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} z_0 z p(r, z_0, z) dz_0 dz - \bar{z}^2 \quad (55)$$

Autokorrelaatio kertoo kahden etäisyydellä r olevan pisteen korrelaation eli samankaltaisuuden. Autokorrelaatiofunktion $A(r)$ fourier muunnos on tarkasteltavan pinnan tehospektri [19] [30] [34]. Tehospektristä pystytään määrittämään tarkasteltavan pinnan vallitsevat taajuudet.



Kuva 10 Karheat pinnat, joilla on sama *rms*-arvo, mutta pinnat ovat silti erilaisia ^[29]

Edellä käsiteltyjen ominaisuuksien perusteella pystytään pinta määrittämään tilastollisesti. Tällainen satunnainen karhea pinta voidaan määrittää esimerkiksi todennäköisyssyvyysjakaumalle $p(z)$ gaussisena jakaumana. Tarkasteltaessa heijastusmalleja tarvitaan pinnan kaltevuus, jonka perusteella voidaan määrittää tarkasteltavan kohdan pinnan normaali. Tämä pystytään määrittämään kahdesta vierekkäisestä syvyysjakauman pisteestä. Vastaavasti pinnan korkeuden perusteella määritetään pinnan itsensävarjostus.

Ongelmana on määrittää jakauman parametrit niin, että ne vastaavat todellista paperin pintaa. Paperin karheutta mitattaessa mitataan yleensä yksiulotteisia karheusprofii-leja. Näiden yksidimensionaalisten profiilien avulla ei pystytä määrittämään karheuden jakaumaa kaksidimensionaaliselle tasolle.

3.1.2 Paperin heijastusominaisuuksien määrittäminen

Paperin karheus voidaan jakaa optiseen karheuteen sekä mikro- ja makrokarheuteen ^{[30] [31]}. Optinen karheus on $0.1 \mu m$ - $1 \mu m$ kokoluokkaa. Mikrokarheus on $1 \mu m$ - $100 \mu m$ kokoluokkaa ja makroskooppinen karheus on $100 \mu m$ - $1000 \mu m$ kokoluokkaa. Karheus voidaan ilmaista optisen karheuden sekä mikro- ja makrokarheuden summana:

$$rms^2 = rms_{opt}^2 + rms_{mikro}^2 + rms_{makro}^2 \quad (56)$$

Edellisessä kappaleessa 3.1.1 käsiteltiin lähinnä mikro- ja makrokarheutta. Optinen karheus voidaan määrittää samoin kuin mikro- ja makrokarheus. Tässä työssä ei

käsitellä optista karheutta, koska se aiheuttaa diffraktioilmiöitä valolle ^[29]. Työn kannalta olennaisia pinnan heijastusominaisuuksia tarkastellaan geometrisen optiikan avulla.

Kun määritetään paperin ominaisuuksia, puhutaan kiillosta ja heijastusominaisuudesta sekä näiden avulla määritetyistä muista ominaisuuksista. Tässä keskitytään erityisesti mikrokiiltoon ja mikroheijastukseen, koska tutkimusongelman kannalta kiinnostavia ovat ominaisuuksien vaihtelut mikrotasolla. Suhteellinen heijastus määritetään seuraavasti ^[31]

$$R = \frac{I_{mitattu}}{I_{referenssi}} \quad (57)$$

Kaavassa (57) verrataan tarkasteltavan pinnan heijastusta referenssi-intensiteettiin, joka voi olla joko mittausvalon intensiteetti tai tilanteeseen sopivan referenssipinnan intensiteetti. Jos ilmoitetaan pinnan heijastusominaisuus, tulee ilmoittaa myös mittausgeometria. Suhteellisen heijastuksen mittausulos vaihtelee mittausgeometrian mukaan. Yleinen mittausgeometriaa on $45^\circ/0^\circ$ astetta. Tämä tarkoittaa, että valon tulokulma on 45° astetta ja mittauskulma on nolla astetta mitattavaan pintaan nähden. Koska heijastusominaisuus on suhteellinen mittayksikkö, mittaukset eivät ole toistettavissa. Heijastusominaisuus riittää kuitenkin diffuusin heijastuksen määrittämiseen, koska paperipinnasta tarvitaan diffuusin heijastuksen vaihtelu tietyn pinnan yli eikä absoluuttista arvoa. ^[31]

Kiillolla tarkoitetaan pinnan suuntavalikoivia heijastusominaisuuksia. Kiiltoa määritettäessä on tarkoitus saada spekuloin heijastuksen piikki esille. Kiilto määritetään samanlaisilla laitteistoilla kuin densiteettikin, mutta mittausgeometria on peiliheijastuskulman mukainen. TAPPI T480 standardin mukainen kiillon mittausgeometria on $75^\circ/75^\circ$ astetta. Toinen yleisesti käytetty mittausgeometria on $45^\circ/45^\circ$ astetta. Kiillon mittauslaitteissa on yleensä pienempi kulma-aperatuuri kuin diffuusin heijastusominaisuuden mittauslaitteissa.

Paperipinnan spektraalisten ominaisuuksien määrittämiseen tarvitaan spektrofotometriä. Spektraaliset ominaisuudet joudutaan määrittämään sekä diffuusille että spekuloin heijastukselle. Mitattu paperin heijastusspektri löytyy liitteestä 1.

Puukuidun taitekerroin on 1.55 ^[41]. Lopullinen paperin taitekerroin muodostuu paperin rakenneosien perusteella. Taitekerrointa tarvitaan Fresnel-termin määrittämiseen. Fresnel-termiä voidaan kuitenkin approksimoida kappaleessa 2.4.2 osoitetulla tavalla heijastusspektrin avulla.

3.2 Painoväriin leviäminen painettavalle pinnalle

Tässä työssä tarkastellaan painoväriin leviämismallia, jonka ovat esittäneet Oittinen ja Saarelma ^[32]. Painoväriin siirtämisproblematiikkaa ei käsitellä. Tässä keskitytään pai-

noväriin kontaktiin ja leviämiseen, kun painoväri on siirretty painoalustalle.

Painoväriin leviämismalli, jonka Oittinen ja Saarelma ovat esittäneet, perustuu painettavan pinnan topologiaan eli pinnan karheuteen. Painoväriin leviämismalli on spatiaaliavaruudessa formuloitu seuraavanlaiseksi:

$$g(x) = A(x)f(x) \otimes h(x) [s + dz(x)] \quad (58)$$

jossa

$A(x)$ on kontaktifunktio painoväriin ja paperin välillä,

$f(x)$ on kuvasignaali,

$h(x)$ on painoväriin leviämiskerto, jolla suoritetaan kovoluutio kuvasignaalfunktiolle,

s on halkeamiskerroin,

d on depositiokerroin ja

$z(x)$ on paperin karheussyvyyden ilmaiseva funktio.

Malli perustuu neljään eri tyyppiseen ilmiöön, jotka kontrolloivat painoväriin asettumista painettavalle pinnalle. Nämä neljä ilmiötä ovat painoväriin ja paperin kontakti, painoväriin leviäminen, painoväriin asettuminen karheuden mukaan sekä painoväriin halkeaminen. Äärimmäisissä tapauksissa mallia kontrolloi yksi näistä ilmiöistä.

Kun malli (58) supistetaan nolllaamalla muut, kuin kontaktiin vaikuttavat termit, saadaan kontaktikontrolloitu termi esiin

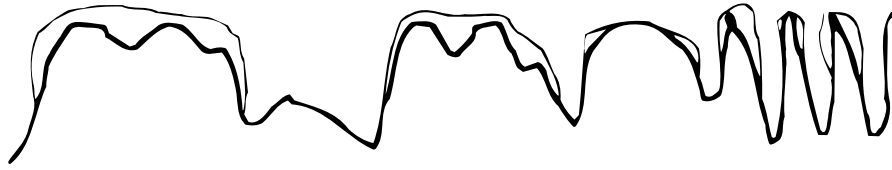
$$g(x) = sA(x)f(x) \quad (59)$$



Kuva 11 Kontaktikontrolloitu painoväriin leviäminen ^[32]

Muut ilmiöt saadaan esille vastaavasti. Painoväriin leviämisen termi on muotoa

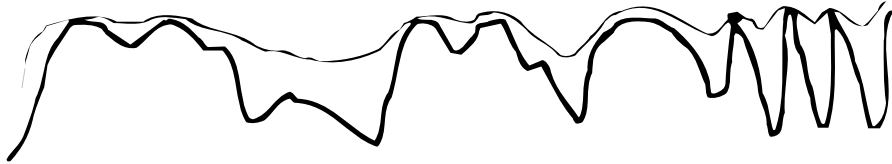
$$g(x) = s[A(x)f(x)] \otimes h(x) \quad (60)$$



Kuva 12 Leviäminen kontrolloi painovärin asettumista paperipinnalle^[32]

Kun tarkastellaan painovärin asettumista pinnan karheuden suhteen, malli (58) supistuu seuraavaaksi

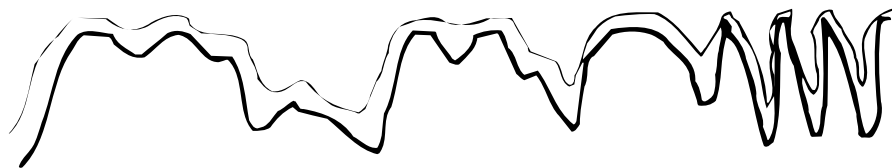
$$g(x) = f(x) [s + dz(x)] \quad (61)$$



Kuva 13 Karheussyvyys kontrolloi painovärin asettumista^[32]

Halkeamisominaisuus saadaan esiin vastaavasti

$$g(x) = sf(x) \quad (62)$$

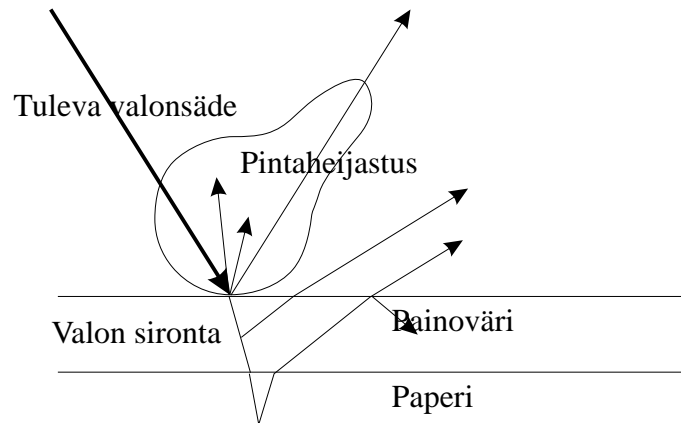


Kuva 14 Halkeaminen kontrolloi painovärin asettumista^[32]

3.3 Optisen kuvan muodostuminen painojäljestä

Painojäljen optisen kuvan muodostuminen tapahtuu, kun valo vuorovaikuttaa painetun pinnan kanssa. Kun valo osuu painojälkeen, osa valosta heijastuu painojäljen pinnasta ja osa taittuu painojäljen sisään. /Kuva 15/. Painojäljen pinnasta heijastuneen valon jakauma sisältää diffuusia ja spekulaaia heijastusta. Painovärin sisään taittuneesta valosta osa absorboituu. Valo absorboituu jonkin spektraalisen jakauman

mukaan riippuen painoväripigmentistä. ^[42]



Kuva 15 Valon heijastuminen painojäljestä

Se osa valosta, joka ei ole absorboitunut, jatkaa matkaansa muihin painovärikerroksiin ja paperiin. Sironnan ja kerrosten rajapintojen heijastumisen ansiosta valo palaa ulos painovärikerroksesta. Paperipohjaan edennyt valo vastaavasti absorboituu ja siroaa. Kun valo etenee eri kerrosten välillä, saattaa tapahtua heijastuksia kerrosten rajapinnasta. Näiden ilmiöiden summana syntyy optinen kuva painojäljestä. ^[42]

Painoväriin valon absorptiokykyä kuvaa absorptiokerroin. Lambert Beerin lain mukaan valon intensiteetti, joka läpäisee äärettömän ohuen kerroksen dx , absorboituu valon absorptiokertoimen ja intensiteetin tulon mukaisesti ^[42]

$$\frac{dI}{dx} = -kI \quad (63)$$

jossa

I on intensiteetti ja

k on absorptiokerroin.

Kun edellinen kaava integroidaan todellisen äärellisen kerroksen x yli, saadaan kerroksen läpi tuleva intensiteetti laskettua seuraavasti ^[42]

$$I = I_0 e^{-kx} \quad (64)$$

jossa

I_0 on pintaan tuleva intensiteetti.

Suhde I/I_0 ilmaisee tarkasteltavan kerroksen läpäisevyyden. Painovärikerroksen läpi menevää valon intensiteettiä, joka heijastuu paperista takaisin ja palaa saman painovärikerroksen läpi takaisin, voidaan johtaa kaavasta (64). Paperista takaisinheijastuneen säteen intensiteetti on

$$I = I_0 e^{-kx} R_{paper} e^{-kx} \quad (65)$$

jossa

R_{paper} on paperin heijastuvuuskerroin.

Kaavaa (65) voidaan edelleen supistaa

$$\frac{I}{I_0} = R_{paper} e^{-2kx} \quad (66)$$

Kaavoissa (65) ja (66) oletetaan, että valon säde palaa paperista kuljettuaan painovärikerroksen läpi. Tilannetta, jossa valonsäde palaisi rasteripisteen ulkopuolelta tarkastelijan silmään, ei oteta huomioon. Läpäisevän ja heijastuvan valon intensiteetti useammalle painovärikerrokselle voidaan johtaa vastaavasti kuin edelliset kaavat .

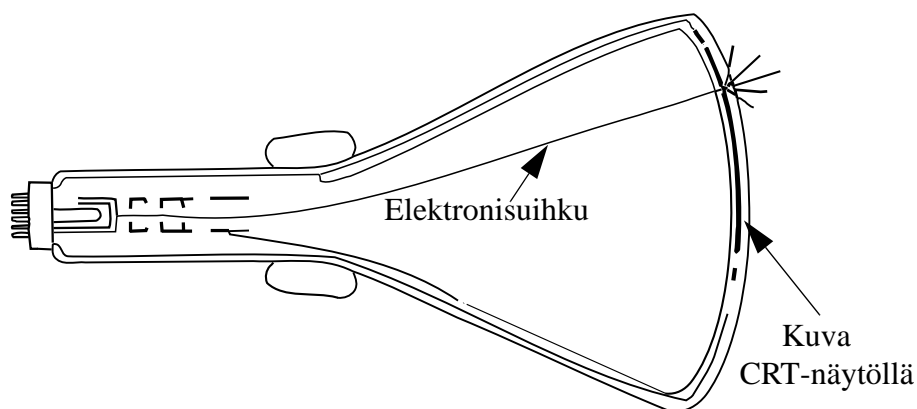
4 PAINOJÄLJEN VISUALISOINNIN ASETTAMAT VAATIMUKSET NÄYTTÖJÄRJESTELMÄLLE

Näyttöjärjestelmällä tarkoitetaan tässä tietokoneen monitoria ja monitorin ohjainta. Tietokoneella voi olla monitorina useita erilaisia, eri periaatteella toimivia laitteita. Tässä keskitytään kuitenkin CRT¹-monitoriin sen yleisyyden ja hyvän suorituskyvyn ansiosta. Ohjainlaitteelta ja monitorilta vaaditaan kykyä näyttää täysvärικuvia. Täysvärικuvalla tarkoitetaan tässä kuvaa, joka on koodattu kolmella tavulla eli 24 bitillä. Täysvärικuvassa jokaiselle RGB-osaväriille on varattu kahdeksan bittiä.

Kun vertaillaan kuvaa CRT-näyttöjärjestelmässä ja painettua kuvaa paperilla, on näillä periaatteellisilla eroja kuvan muodostuksessa. CRT-näyttöjärjestelmä muodostaa optisen kuvasignaalin aktiivisesti, kun taas painojäljen muodostama optinen kuva riippuu ympäröivistä olosuhteista. Vastaavasti CRT-näyttöjärjestelmässä väri muodostuu additiivisesti ja painojäljessä kuva muodostuu subtraktiivisesti.

4.1 Kuvan muodostus CRT-näyttöjärjestelmässä

CRT-monitorin katseltava pinta on päällystetty loisteaineella, joka emittoi valoa, kun sitä pommitetaan suurienergisellä elektronisuihkulla /*Kuva 16*/. Loisteaine koostuu fosforyhdisteistä ja se on katodisädeputken sisäpinnalla. Tavallisesti fosforyhdisteet lähettävät valoa vain lyhyen aikaa. Jälkiloisto aika riippuu fosforyhdisteestä. Jälkiloisto ajasta riippuen loisteainepintaa pitää virkistää sopivalla taajuudella, jotta kuva pysyisi tasaisen kirkkaana. Loisteaineen pinnalle elektronit kohdistetaan sähkömagneettisesti. Elektronisuihkun voimakkuutta säätämällä aikaansaadaan eri kirkkaustasoja näytöllä. [46]

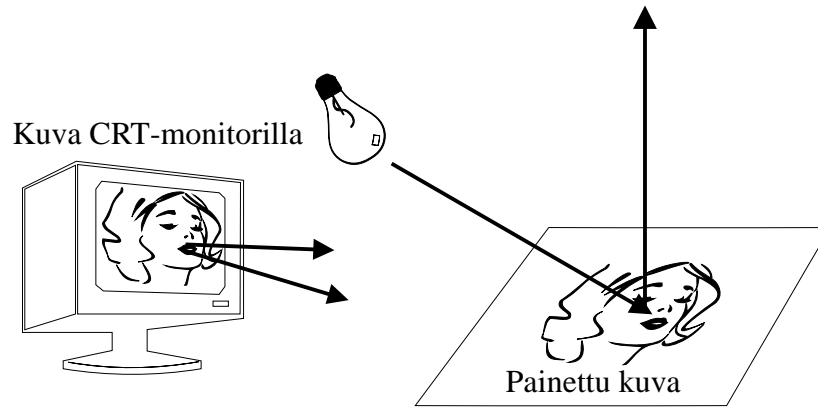


Kuva 16 Poikkileikkaus CRT-monitorista

Väri generoidaan CRT-monitorilla useammalla eri elektronisuihkulla, jotka kohdistetaan eri spektraalisen vasteen omaaville fosforyhdisteille. Näin voidaan tuottaa esimerkiksi CRT-monitorissa RGB värikuva.

1. CRT='Cathode Ray Tube' eli katodisädeputki

Optinen kuva muodostuu CRT-monitorilla aktiivisesti eli CRT-monitori tuottaa valon. Paperilla kuva muodostuu ympäröivän valon vaikutuksesta /*Kuva 17*/. CRT-monitorilla on aktiivisuutensa ansiosta ominaisuuksia, joita ei esiinny painojäljessä. CRT-monitorin painojäljestä poikkeavat ominaisuudet ovat ajallisesti vaihtelevat optiset ominaisuudet. Tällaisia ominaisuuksia ovat välkkyminen, ryömintä ja värinä ^[1].



Kuva 17 Optisen kuvan muodostuminen CRT-näyttöjärjestelmässä sekä painojäljessä

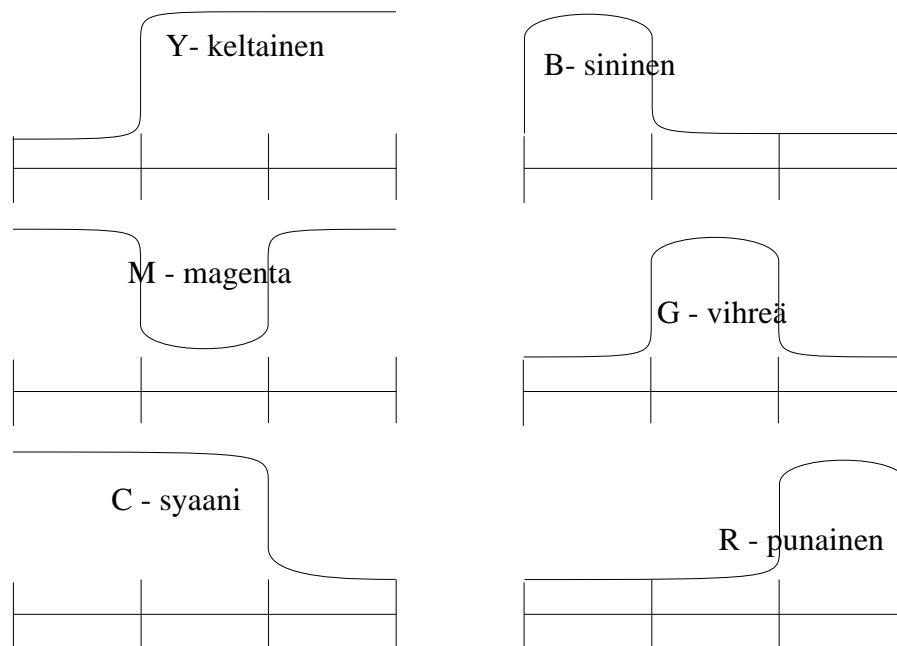
Välke (flicker) on monitorin luminanssin ajallista vaihtelua. Välkkynä ei suoranaisesti vaikuta yksityiskohtien toistokykyyn, mutta se rasittaa silmiä, kun joudutaan työskentelemään näyttöjen kanssa. Välkkynään vaikuttaa kohteen kirkkaus ja näytön koko. Mitä suurempi näyttö ja mitä kirkkaampi kuva, sitä suurempi virkistystaajuus ja pidempi loisteaineen jälkiloisto tarvitaan näytölle.

Ajelehtiminen ja värinä (jitter) ovat monitorilla olevan merkin paikallista vaihtelua ajan suhteen. Ajelehtiminen on merkkien hidasta siirtymistä paikasta toiseen. Tämä aiheuttaa merkkien sekoittumista ja epäselvyyttä. Värinä on suuritaajuisia merkin aseman vaihtelua ja tämä aiheuttaa epäterävyyttä merkeissä.

CRT-näyttöjärjestelmässä on kiinteä monitorista johtuva resoluutio. Kun visualisoidaan paperipintaa yhtäsuurella resoluutiolla kuin painojäljessä on, voidaan CRT-monitorin kiinteä resoluutio kiertää säätämällä katseluetäisyyttä. Tässä ongelmana on CRT-monitorin koko. Mitä kauempaa monitoria katsotaan, sitä pienemmän avaruuskulman monitorin tarkastelualue peittää.

Kun tarkastellaan kuvaa näyttöjärjestelmässä ja halutaan tulostaa sama kuva painetuna kuvana, joudutaan tekemään konversio RGB signaalista CMY signaaliksi. Ideaalisten RGB- ja CMY-värisignaalien spektraaliset vasteet ovat toistensa vastakohtia /*Kuva 18*/. Tämä konversio on periaatteessa yksinkertainen.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (67)$$



Kuva 18 Ideaalisten RGB ja CMY signaalien spektraaliset vasteet

Kun halutaan visuaalisesti samankaltainen tulos näyttöjärjestelmän kuvalle ja painetulle kuvalle, joudutaan kalibroimaan systeemi ottaen huomioon käytetty näyttöjärjestelmä, käytettävät painomenetelmät ja -värit .

4.2 Näyttöjärjestelmän kalibrointi

Kun tarkastellaan näyttöjärjestelmässä olevaa kuvaa ja tulostetaan tämä paperille, halutaan molempien kuvien olevan samanlaisia sekä geometrialtaan että väriominaisuuksiltaan. Tavallisesti tämä WYSIWYG-käsite (What You See Is What You Get) ymmärretään ainoastaan geometriseltä kannalta eli ympyrä on ympyrä sekä monitorilla että paperilla. Yhä useammin nämä vaatimukset asetetaan myös väriominaisuuksille, jotka ovat tämän työn kannalta tärkeitä. Täydelliseen WYSIWYG:iin pyrittäessä tarvitaan koko systeemin kattavaa kalibrointia.

Koko kuvantuotantosysteemiä, kuvattavasta asiasta, kuvaan näyttöjärjestelmässä ja paperitulosteessa, joudutaan kalibroimaan. Yleensä tuotantoympäristössä keskitytään kalibroimaan vain näyttö- ja tulostusjärjestelmä. Kalibrointi voidaan suorittaa tilastollisesti tai malleihin ja teoriaan pohjautuen. Tilastollisesti kalibrointi suoritetaan mittaamalla riittävästi pisteitä, joiden perusteella generoidaan taulukko ns. LUT (Look Up Table). Taulukon perusteella suoritetaan värin korjailut. Tämän työn kannalta on kuitenkin tärkeämpää kalibroida näyttöjärjestelmä itsenäisesti, koska tuloskuva generoidaan synteettisesti ja sitä tarkastellaan monitorilla.

CRT-näytön input-output suhde on epälineaarinen ja RGB-primäärien kromaattisuusarvot riippuvat toteutustekniikasta eli ovat hyvin laitekohtaisia. Kun näyttöjärjestelmää kalibroidaan, pyritään linearisoimaan input-output suhde ja värit esitetään laitteistoriippumattoman koordinaatiston avulla. Nämä koordinaatit muutetaan laitteistoriippuvaiseksi sopivien laitekohtaisten muunnosparametrien avulla. Geometrisiä

vääristymiä pystytään korjaamaan erilaisten koordinaatistomuunnosten avulla. Koska CRT-näyttöjärjestelmä on tarpeeksi lineaarinen geometrisessä mielessä ja koska geometriset transformaatiot ovat laskennallisesti raskaita operaatioita, geometristen vääristymien korjaukseen ryhdytään harvoin.

Laitteistoriippumattomia värikoordinaatistoja ovat esimerkiksi CIE-standardin mukaiset tristimuluskoordinaatit tai HVS-koordinaatisto (Hue, Value, Saturation) ^[42]. HVS-koordinaatteja käsitellessä muutetaan värisävyä, harmautta ja kirkkautta. Tällaisten termien avulla värin käsittely on helpompaa kuin suoraan RGB-arvojen käsittely. Haluttaessa kirkkaampi värisävy jostain RGB-värisävystä joudutaan muuttamaan kaikkia kolmea RGB-vektorin arvoa. Muunnos HVS-koordinaateista näyttöjärjestelmäriippuvaisiksi RGB-koordinaateiksi on yksinkertainen matriisioperaatio

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} H \\ V \\ S \end{bmatrix} \quad (68)$$

jossa

A on laiteriippuvainen muunnosmatriisi.

Vastaavasti voidaan määrittää tulostimesta riippuvainen muunnosmatriisi, CMY-koordinaattimuunnokselle. Jos käsiteltävät koordinaatit ovat laiteriippuvaisia RGB-koordinaatteja, voidaan määrittää käänteismatriisi muunnosmatriisille **A** ja määrittää RGB-arvoja vastaavat HVS-arvot, jonka jälkeen voidaan määrittää vastaavat CMY-arvot.

XYZ-tristimuluskoordinaatit lasketaan seuraavasti spektristä:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{tr}^T \\ \mathbf{Y}_{tr}^T \\ \mathbf{Z}_{tr}^T \end{bmatrix} \mathbf{I}_{spectrum} \quad (69)$$

jossa

$\mathbf{X}_{tr}, \mathbf{Y}_{tr}, \mathbf{Z}_{tr}$ vektorit CIE standardin mukaisia spektraalisia herkkyyksiä näkyvän valon alueella.

Tristimuluskoordinaattien muuttaminen laitteistosta riippuviksi RGB-arvoiksi tapahtuu samalla tavalla kuin kaavassa (68).

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (70)$$

jossa

B on muunnosmatriisi XYZ-koordinaateista laiteriippuviksi RGB-koordinaateiksi.

XYZ-tristimuluskoordinaatteja voidaan käsitellä samalla tapaa kuin HVS-koordinaatteja, mutta tristimuluskoordinaattien käsittely ei ole yhtä helppoa kuin HVS-koordinaattien. Toisaalta näkyvän valon spektrit ovat absoluuttisia ja voidaan mitata.

Laitteistoriippumattomien koordinaattien käyttö ei riitä, jos ei tiedetä muunnosmatriisia koordinaatistomuunnoksille. Muunnosmatriisi voidaan määrittää mittamalla, mutta nämä mittaukset pätevät ainoastaan tietyille mittauksen aikana olleille säädöille. Jos kontrastia tai kirkkautta muutetaan, muuttuu muunnosmatriisi. ^[42]

Monitorin kalibrointi tehdään säätämällä valkotasapaino tietyn värilämpötilan mukaiseksi, koska CRT-näyttöjärjestelmä voidaan olettaa valonlähteeksi. Tämä säätö tehdään kolmen fosforin luminanssin ollessa maksimissa. Yleisimpiä värilämpötiloja, joihin monitorit säädetään, ovat 9300 K ja 6500 K eli D65 valonlähde. D65 valonlähde vastaa päivänvaloa. Eri standardit määrittelevät erilaisia värilämpötiloja, joiden mukaan monitori kalibroidaan.

Näytön linearisointi suoritetaan gamma-korjauksella. CRT-näytön elektronisäteen energian kasvattaminen lineaarisesti ei kasvata luminanssia lineaarisesti. Näiden kahden välinen suhde on eksponentiaalinen. Tästä johtuen voidaan gamma-korjaus formuloida seuraavasti:

$$L(v) = (kv)^\gamma \quad (71)$$

jossa

k on näytöstä riippuvainen kerroin ja

γ on näytöstä riippuvainen korjaustermi.

Gamma-korjauksen potenssi vaihtelee useimmilla näytöillä 2.2 ... 2.5 välillä. Markkinoilta löytyy useita valmiita ohjelmisto- ja laitteistopaketteja näyttöjärjestelmien kalibrointiin ^[54].

5 VISUALISOINNIN IMPLEMENTOINTI

Työn toitteena on visualisoida painojälkeä tietokoneen näyttöjärjestelmässä. Visualisointia varten rakennetaan ohjelmisto, joka ottaa huomioon heijastus-, painoväri- sekä paperimallit. Ohjelmistolla generoidaan Tiff kuvia, joita tarkastellaan tietokoneen monitorilla.

5.1 Käytetty ohjelmointiympäristö ja valmiit ohjelmat

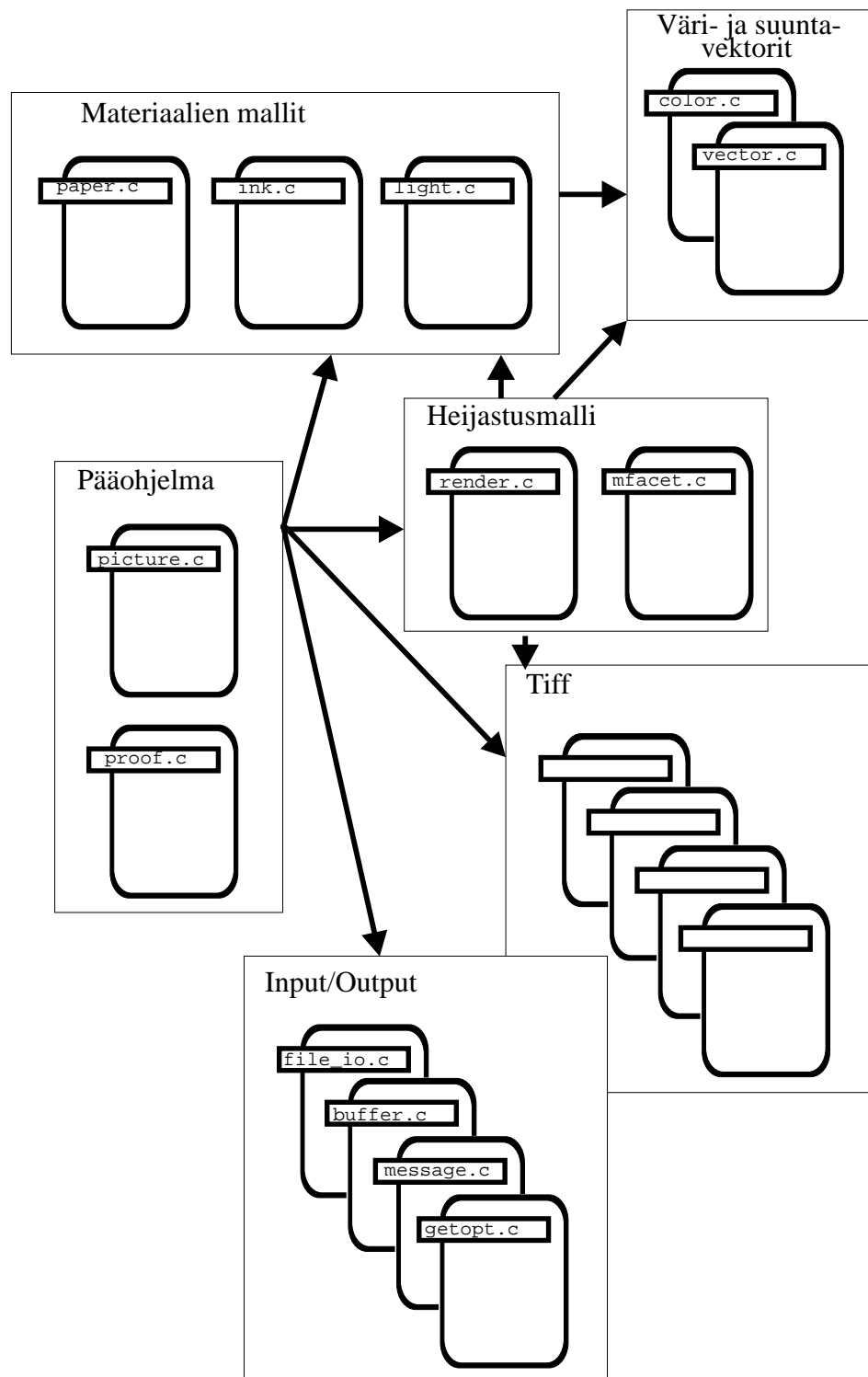
Ohjelmisto on tehty Ansi-C ^[21] standardin mukaisella C-kielen syntaksilla. Ohjelmiston kehitysympäristönä on käytetty Borlandin C++ 3.1 kääntäjää ja Application Frameworks ympäristöä. Borlandin kehitysympäristöä käytettiin DOS ohjelman luomiseen. Tähän ratkaisuun päädyttiin, koska tällöin ei tarvinnut välittää käyttöliittymästä, joka tämän ongelman kannalta oli merkityksetön. Fyysisenä laiteympäristönä on käytetty PC-laitteistoa. Laskennan kannalta tärkeitä ominaisuuksia koneessa ovat prosessorityyppi ja kellotaajuus sekä keskusmuistin koko. Prosessorina tietokoneessa oli Intelin 486 sarjan prosessori, jonka kellotaajuus oli 50 MHz. Keskusmuistia tietokoneessa oli 8 megatavua.

C-kielen yleisyyden ansiosta ohjelmistossa on ollut mahdollisuus käyttää valmista ohjelmakoodia hyväksi. Phongin ja Blinnin heijastusmallien implementoinnin kannalta tärkeä valmis ohjelmapaketti on Hall:n jakama ohjelmakoodi, joka perustuu hänen kirjaansa ^[12]. Tätä ohjelmapakettia on käytetty soveltuvin osin värin, mikropintojen ja vektorien implementointiin. Kuvien käsittelyyn ja tallentamiseen on käytetty tiff-kirjastoa, joka on haettavissa ftp-protokollan avulla internet tiedostopalvelimesta 'sgi.com'. Tiff-kirjastosta on käytetty mahdollisimman yksinkertaista versiota, josta on karsittu kuvan pakkaamiset ja muut ylimääräiset ominaisuudet pois. Lisäksi on käytetty optioiden käsittelyyn tarkoitettua ohjelmatiedostoa 'getopt.c'. Käytettyjen valmiiden ohjelmakoodien tekijänoikeusilmoitukset ovat liitteessä 2.

Valmiiden kuvien tarkastelussa pyritään hakemaan visuaalisia eroja generoiduille kuville. Tarkasteluun on käytetty Microsoftin Windows ympäristössä toimivia ohjelmia, Alduksen PhotoStyler versio 1.1 ja Adoben PhotoShop versio 2.5. Ohjelmat ovat pikselikarttakuviin perustuvia kuvan muokkausohjelmia. Molemmat ohjelmat ovat ominaisuuksiltaan samankaltaisia ja soveltuvat hyvin RGB Tiff-kuvien tarkasteluun.

5.2 Ohjelmiston rakenne

Ohjelmisto on rakennettu moduleista, joilla kullakin on oma tehtävänsä. Ohjelmiston rakennetta suunniteltaessa on otettu huomioon toiminnan selkeys ja havainnollisuus. Tehokkuustekijöiden huomiointi on jätetty vähälle, koska ohjelman tavoitteena ei ollut tehokas ja nopea toiminta, vaan visuaalisten lopputulosten tarkastelu. Ohjelman jakamisessa osiin on ollut periaatteena ohjelman helppo siirrettävyys muihin käyttöympäristöihin.



Kuva 19 Painojäljen visualisointiohjelmiston rakenne

Ohjelma on jaettu kuuteen päälohkoon: pääohjelma, input/output, tiff, heijastusmalli, väri- ja vektoriavaruudet sekä materiaalien mallit /Kuva 19/. Pääohjelma sisältää ylimmän tason ohjaukset ohjelmistolle. Muut lohkot sisältävät omat tarkat tehtävänsä. Tarkemmat kuvaukset lohkojen sisällöstä on esitelty jäljempänä. Kuvassa 19 näkyy lohkojen väliset kommunikaatiolinkit. Vähemmän tärkeät kommunikaatiolinkit loh-

kojen välillä on jätetty pois selkeyden takia.

5.2.1 Pääohjelma

Pääohjelma lohko sisältää kuvan generoinnin kannalta ylimmän tason ohjaukset, kuten kuvan koon määrittelyt. Pääohjelmalohko sisältää kaksi C-kielistä tiedostoa ‘proof.c’ ja ‘picture.c’. Näiden kahden tiedoston välillä ‘picture.c’ käsittelee kaikki kuvan generoinnin ohjaukseen liittyvät tiedot ja ‘proof.c’ käsittelee komentorivin optioiden tulkitsemiset ja ohjelman käynnistys- ja lopetusrutiinit. Tarkemmat ohjeet ohjelman käytöstä ja optioista löytyvät liitteestä 3.

Tiedosto ‘picture.c’ sisältää funktioita, jotka toimivat käyttöliittymänä kuvatietorakenteeseen. Kuvatietorakenne sisältää kuvan generoinnin kannalta olennaiset tiedot /*Esimerkki 1*/. Kuvan koko tietorakenteessa ilmaistaan X ja Y akselin suuntaan millimetreinä ja pikselikokona. Kuvan koon ja pikselikoon antaa ohjelman käyttäjä. Näiden tietojen avulla lasketaan kuvalle pikselikoko, joka tallennetaan myös tietorakenteeseen. Pikselikoko saadaan laskettua seuraavasti:

$$\begin{aligned} X &= \frac{1000x}{d} \\ Y &= \frac{1000y}{d} \end{aligned} \tag{72}$$

Jossa

x, y ovat kuvan fyysiset koot millimetreinä

d on pikselin koko mikrometreinä

```
typedef struct {
    double sizeX, sizeY; /*Size in millimeters */
    double DotSize;      /*Resolution, size of pixel */
    double ViewDirection; /*The view angle */
    int    PixX, PixY;    /*Size in pixels*/
    int    IllumModel;    /*Illumination model*/
    String name;          /*Name of the picture file */
    String light;         /*Name of the light file*/
    String paper;         /*Name of the paper file*/
    String ink;           /*Name of the ink file*/
    LogicalUseInk;        /*TRUE if ink is used*/
    TIFF *tif;            /*Pointer to TIFF structure*/
} PictureStruct;
```

Esimerkki 1 Kuvatietorakenne

Viittaukset materiaalien malleihin on tallennettu tiedostojen niminä tietorakenteeseen. Näiden nimien avulla avataan materiaalien parametritiedostot ja alustetaan materiaalien tietorakenteet. Generoitavan kuvan nimi ja osoitin kuvatietorakenteeseen on tallennettu tietorakenteeseen. Lisäksi tietorakenteessa on tiedot käytettävästä

heijastusmallista ja tarkastelusuunnasta.

5.2.2 Input/Output/Tiff

Input/Output moduli on liitäntäpinta käyttäjän ja ohjelman välillä sekä käyttöjärjestelmän ja ohjelman välillä. Tähän lohkoon on eroteltu tiedoston hallinta, ruudulle tulostus, muistin hallinta ja optioiden käsittely. Näihin tiedostoihin ei tosin ole sijoitettu valmiiden ohjelmatiedostojen muistin hallintaa ja tiedostojen käsittelyä. Valmiit ohjelmamodulit kuten Tiff-kirjasto käyttää omia rutiinejaan tiedoston hallintaan sekä virheilmoitusten tulostukseen. Input/Output lohko sisältää neljä tiedostoa: 'Message.c', 'Buffer.c', 'File_io.c' ja 'Getopt.c'.

Tiedosto 'Message.c' käsittelee ruudulle tulostuksen. Modulissa on valmiit rutiinit virheilmoitusten käsittelyyn ja viestien tulostukseen. Kun 'Message.c' antaa virheilmoituksen, moduli lopettaa koko ohjelman toiminnan. Lisäksi tiedostossa on erityinen funktio lyhyen käyttöohjeen tulostamiseen.

Tiedosto 'Buffer.c' sisältää muistin hallintafunktioita. Muistin varaaminen ja vapauttaminen on toteutettu makrojen avulla. Muuten tiedosto sisältää merkkijonopuskureiden käsittelyyn tarvittavia funktioita. Näillä funktioilla pystytään hakemaan puskurista kokonaislukuja, desimaaleja sekä tavallisia sanoja.

Tiedosto 'File_io.c' sisältää tiedosto-operaatioita. Tämä moduli rakentaa linkitetyn listan avatuista tiedostoista. Linkitetty lista sisältää tiedoston nimen ja osoittimen avattuun tiedostoon. Listan avulla moduli etsii oikean tiedoston ja suorittaa sille oikean operaation. Tämä moduli on tarkoitettu lähinnä ASCII-muotoisten tiedostojen käsittelyyn. Moduli lukee ja kirjoittaa rivin kerrallaan avattuihin tiedostoihin.

Tiedosto 'Getopt.c' sisältää funktion optioiden tulkitsemiseen. Tämä funktio lukee komentorivillä annetut argumentit ja palauttaa löydetyn option arvon kutsuneelle ohjelmalle. Jos optio vaatii argumentin, getopt funktio antaa osoittimen löydettyyn option argumenttiin.

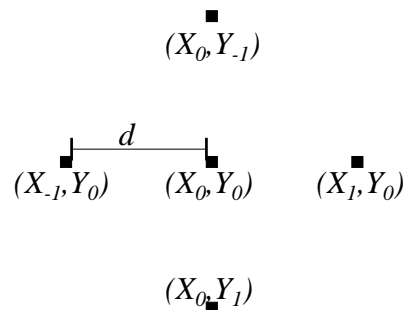
5.2.3 Heijastusmalli

Valon heijastusmalli on implementoitu 'render.c' ja 'mfacet.c' tiedostoissa. Tiedosto 'mfacet.c' sisältää useita funktioita mikropintojen distribuution laskeamiseen. Tämä tiedosto on luotu Hallin kirjassa^[12] olevien esimerkkien mukaan. Tiedostoon 'render.c' on implementoitu Phongin ja Blinnin heijastusmallit. Tiedosto 'render.c' käyttää hyväkseen materiaalien implementointeja tiedoissa 'paper.c', 'ink.c' ja 'light.c'.

5.2.4 Materiaalien mallit

Materiaalien mallit on jaettu kolmeen eri tiedostoon, 'paper.c', 'ink.c' ja 'light.c'. Kussakin tiedostossa on tietorakenne eri materiaalien malleille, johon käyttöliittymän muodostaa tiedoston sisältämät funktiot. Näiden tiedostojen tehtävänä on alustaa mallien tietorakenteet, lukea mallien parametrit tiedostoista ja konvertoida parametrit ohjelmalle sopivaan muotoon.

Tiedosto ‘paper.c’ sisältää paperimallin tietorakenteen /*Esimerkki 2*/. Paperimalli perustuu karheusmatriisiin, joka on tallennettu tietorakenteessa Rough matriisiin. Karheusmatriisiin on tallennettu paperipinnan korkeusvaihtelu. Korkeusvaihtelu on koodattu matriisiin kahdeksalla bitillä eli korkeusvaihtelu voi saada kokonaislukuarvoja 0 ja 255 välillä. Todellisen etäisyyden 0 ja 255 välillä kertoo maksimikarheus Range, johon on tallennettu maksimietäisyys mikrometreinä. Karheusmatriisin peittoalueen käsittelyyn on kaksi kokomuuttujaa tietorakenteessa. Toinen kokomuuttuja kertoo rivien ja sarakkeiden lukumäärän matriisissa ja toinen kertoo matriisin todellisen koon mikrometreissä. Todellinen koko lasketaan pikselikoon avulla, joka on tallennettu tietorakenteeseen. Pikselikoko kertoo karheusmatriisin kahden neljäkonnektiivisen pisteen etäisyyden toisistaan /*Kuva 20*/. Painaessa muodostuvan kontaktipinnan etäisyyden maksimiarvosta 255 kertoo Contact muuttuja, johon on tallennettu kontaktipinnan etäisyys mikrometreinä.



Kuva 20 Neljäkonnektiivisten pisteiden etäisyys karheusmatriisissa. Etäisyys vastaa käsitellyn pikselin kokoa

```
typedef struct {
    double PixelSize; /* Size in um */
    ColorType Diffuse; /* Diffuse coefficient and scale */
    double DiffuseScale;
    ColorType Specular; /* Specular coefficient and scale */
    double SpecularScale;
    double SpecularBeta;
    ColorType Ambient; /* Ambient Coefficient and scale */
    double AmbientScale;
    double Contact; /* Contact level for ink transfer */
    double Range; /* Max roughness tells the maximum possible
                   roughness between 0 and 255 in roughness
                   matrix the value is in micro meters */
    IPoint ISize; /* Number of rows and columns in matrix */
    DPoint DSize; /* Actual size of matrix */
    IPoint IBeta; /* Size of Beta matrix (rows/columns)*/
    DPoint DBeta; /* Actual size of Beta matrix */
    BetaType **Beta;
    RoughType **Rough;
} PaperStruct ;
```

Esimerkki 2 Paperimallin tietorakenne

Paperipinnan heijastus- ja väriominaisuudet on tallennettu värivektoreihin ja näiden kertoimiin. Ambientin heijastuksen värin kertoo Ambient värivektori ja sen kertoimen kertoo AmbientScale. Diffuusilla ja spekulaarilla heijastuksella on vastaavat värivektorit ja kertoimet mallin tietorakenteessa. Spekulaarilla heijastuksella on betakulma, josta tarkemmin kappaleessa 2.4.3. Betakulmalle on tallennettu mallin tietorakenteeseen peruskulma radiaaneina, sekä betakulman vaihtelu matriisiin Beta. Betamatriisilla on vastaavanlaiset kokomuuttujat kuin karheusmatriisillakin eli matriisin kattaman alueen koko ja matriisin rivien ja sarakkeiden lukumäärä.

Tiedosto 'light.c' sisältää valonlähteen mallin tietorakenteen /*Esimerkki 3*/. Valonlähteen mallin tietorakenne sisältää pistemäisen valonlähteen ominaisuudet ja ambientin valaistuksen ominaisuudet. Muuttujat X, Y ja Z ilmaisevat pistemäisen valonlähteen sijainnin xyz-avaruudessa. Lisäksi valonlähteellä on intensiteetti sekä spektraalinen jakauma, joka on tallennettu Color -vektoriin. Ambientillä valaistuksella on vastaavat ominaisuudet kuin valonlähteelläkin. Ambientiltä valaistuksella ei ole sijaintia kuten valonlähteellä.

```
typedef struct {
    double      X,Y,Z;
    double      Intensity;
    double      AmbientIntensity;
    ColorType    Color;
    ColorType    AmbientColor;
} LightStruct ;
```

Esimerkki 3 Valonlähteen mallin tietorakenne

Tiedosto 'ink.c' sisältää painovärimallin tietorakenteen /Esimerkki 4/. Painovärimallissa ei ole karheusominaisuuksia kuten paperin mallissa, koska painoväri levittäytyy paperin pinnalle ja imeytyy sen sisään. Painovärimallin tietorakenne perustuu rasterimatriisiin, joka sisältää painokuvion rasterirakenteen. Tämä matriisi on tallennettu tietorakenteen Image muuttujaan. Painovärirasteri sisältää arvot nolla tai yksi, jotka ilmaisevat painovärimallin rasterirakenteen kyseisessä kohdassa. Kyseiseen kohtaan joko välittyy painoväriä tai ei välity. Image muuttujan käsittelyssä käytetään kahta kokomuuttujaa ISize ja DSize kuten paperimallinkin tietorakenteessa. Kokomuuttujat kuvaavat painovärimatriisin peittämää aluetta sekä painovärimatriisin rivien ja sarakkeiden lukumäärää. Painovärimatriisin todellinen koko saadaan pikselikoon avulla. Painovärimallin paksumuuttuja ImageScale, johon on tallennettu painovärimallin paksumuuttuja mikrometreinä. Painovärirasterin sijainnin ilmaisee Location muuttuja, joka ilmaisee rasterimatriisin vasemman yläkulman sijainnin kuvan origoon nähden.

Painoväriin siirtymistä ja leviämistä paperille kuvaa joukko muuttujia tietorakenteessa. Leviämistä kontrolloidaan konvoluutio matriisilla *Convolution*, joka sisältää painoväriin leviämiskertoimien arvoja. Konvoluutiomatriisiin käsittelyssä käytetään konvoluutiomatriisin koko muuttujia sekä konvoluutiomatriisin keskuspuheen osoittavaa pistettä. Lisäksi tietorakenteessa on halkeamiskerroin *Splitting*, joka kuvaa painoväriin halkeamista painettavalle pinnalle, depositio kerroin *Deposition* sekä siirtotermi *Transfer*, johon tallennetaan todellisuudessa siirtyvän painoväriin

määrä kullekin pisteelle.

Painetun pinnan spektrin ominaisuudet sekä heijastusominaisuudet on tallennettu värivektoreihin Ambient, Diffuse ja Specular sekä näiden kertoimiin AmbientScale, DiffuseScale ja SpecularScale. Spekulaarilla heijastuksella on betakulma, joka on tallennettu omaan muuttujaansa radiaaneina. Betakulmalla ei ole vaihtelua kuten paperimallissa on. Painovärimalli olettaa painovärin säilyttävän saman betakulman jokaiselle pisteelle, kun painoväri siirtyy paperille. Lisäksi on absorptio termi Absorption, joka kuvaa valon imeytymistä painoväriin.

```
typedef struct {
    double      PixelSize;
    DPoint      Location;
    IPoint      ISize;
    DPoint      DSize;
    ImageType   **Image;
    int         PicType;
    double      ImageScale;
    IPoint      IConv;
    DPoint      DConv;
    ConvType    **Convolution;
    IPoint      IConvCenter;
    double      Splitting;
    double      Deposition;
    double      Absorption;
    double      Transfer;
    ColorType   Ambient;
    double      AmbientScale;
    ColorType   Diffuse;
    double      DiffuseScale;
    ColorType   Specular;
    double      SpecularScale;
    double      SpecularBeta;
} InkStruct ;
```

Esimerkki 4 Painovärimallin tietorakenne

5.2.5 Väri- ja suuntavektorit

Heijastusmallin implementoinnissa käytetään runsaasti vektorialgebraa. Vektorien laskutoimitukset on toteutettu omassa tiedostossaan 'vector.c', josta löytyy peruslaskutoimitukset sekä vektorien alustamiset. Spektrien käsittelemiseen on oma tiedostonsa 'color.c', jossa on spektrivektorien alustusoperaatiot, väriavaruuksien muunnokset sekä väriavaruuden sovittamisoperaatioita. Väriavaruuden sovittamisella tarkoitetaan tässä pisteen väriarvon siirtämistä monitorin värialueelle, jos pisteen väriarvo ylittää monitorin näyttökyvyn. Tiedostot 'vector.c' ja 'color.c' on rakennettu Hallin jakamien tiedostojen ja kirjan^[12] pohjalta.

5.3 Heijastusmallin ratkaisu

Ohjelmistossa on käytetty Blinnin ja Phongin heijastusmalleja. Phongin mallia käytetään, koska se on yleisin heijastusmalli kaupallisissa sovellutuksissa. Blinnin mallia käytetään, koska malli on pyrkii mallittamaan todellista heijastusilmiötä paremmin kuin Phongin malli. Molempia malleja on modifioitu jonkin verran luvussa 2 esitetyistä muodoista. Myöhemmät esimerkiksi Greenbergin^[11] ja Rushmeier & Torrance^[40] mallit lisäävät heijastusilmiön tarkasteluun materiaalin läpäisevät ja muista kappaleista heijastuvat valonsäteet. Nämä ovat kuitenkin epäolennaisia tämän työn kannalta.

Kuvan laskeminen on toteuttu sisäkkäisillä silmukoilla /*Esimerkki 5*/. Silmukat laskevat valon heijastuksen pikseli kerrallaan koko kuvan alueelta. Silmukoiden käsittelemien pikselien resoluutio on sama kuin kuvan pikseliresoluutio. Kuten esimerkin 5 silmukoista näkee, valonlähteen suuntavektoria ei määritetä silmukoiden sisällä. Varsinaisen heijastusmallin RGB arvon laskee esimerkissä 5 funktio `Blinn()`, joka käyttää hyväkseen pinnan normaalia, valon tulosuuntaa, tarkastelusuuntaa sekä tarkasteltavaa pistettä. Todellinen implementointi kuvan generointisilmukoille ottaa huomioon painoväriin ja heijastusmallin, jotka on jätetty esimerkistä 5 pois.

```
for(row=0;row<picture.Y;row++)
{
    px.x=0.0;
    for(i=0;i<picture.X;i++)
    {
        PaperGetNormalVector(&paper,&px);
        mtl.SpecularPower=MFacetBlinnInit(
            PaperGetSpecularBeta(&px));
        rgb=Blinn(&paper,&light,&view,&px);
        PutPixel(buf,RGB,RED,i,(value_t)rgb->r);
        PutPixel(buf,RGB,GREEN,i,(value_t)rgb->g);
        PutPixel(buf,RGB,BLUE,i,(value_t)rgb->b);
        px.x+=picture.PixelSize;
    }
    if(WriteRowBuffer(picture.Tif,buf,row)==FALSE)
        MessageError("Error in writing to TIFF file");
    px.y+=picture.PixelSize;
}
```

Esimerkki 5 Yksinkertaistetut sisäkkäiset silmukat, joka toteuttavat kuvan laskemisen Blinnin heijastusmallin avulla

Heijastusmallifunktiot `Blinn()` ja `Phong()` toteuttavat heijastusmallin implementoinnin. Implementoinnissa on otettu spektrien avulla materiaalien väriominaisuudet huomioon. Phongin malli on toteutettu kaavalla, joka vastaa kappaleessa 2.3 esitettyä kaavaa (11):

$$\mathbf{I} = \mathbf{P}_a^T \mathbf{I}_A + \mathbf{I}_L^T [\mathbf{P}_d (\mathbf{L} \cdot \mathbf{N}) + \mathbf{P}_s (\mathbf{R} \cdot \mathbf{E})^n] \quad (73)$$

jossa

\mathbf{I}_A on ambientin valaistuksen spektri,

\mathbf{I}_L on valonlähteen spektri,

$\mathbf{P}_a, \mathbf{P}_d, \mathbf{P}_s$ ovat materiaalin ambientin, diffuusin ja spekuläärin heijastuksen kertoimien spektrivektorit,

$\mathbf{L}, \mathbf{N}, \mathbf{E}, \mathbf{R}$ ovat valonlähteen, pinnan normaalin, katselusuunnan ja peili-heijastus suunnan suuntavektorit,

n on spekuläärin heijastuksen potenssi, joka määritetään betakulman avulla ja

\mathbf{I} on tulokseksi saatava spektrivektori.

Kaikki spektrikertoimet on toteutettu vektoreille, jotka sisältävät spektrin arvot nanometrin välein välillä 380 nm ja 780 nm. Jos valonlähteestä tuleva valo ei saavuta tarkasteltavaa pistettä, jätetään kaavasta (73) jälkimmäinen termi kokonaan huomiotta. Tällöin kaavasta (73) tarkastellaan ainoastaan termiä $\mathbf{P}_a^T \mathbf{I}_A$. Tämän aiheuttaa valonlähteen spektrin termi \mathbf{I}_L , joka menee nolnaan, kun valonlähteestä ei osu suoraan valo.

Phongin malli olettaa, että spekuläärin heijastuksen piikki on valonlähteen valon värinen. Kaavasta (73) voidaan todeta että, jos kerroin \mathbf{P}_s on tasainen spektrivektori, täyttää kaava (73) Phongin oletuksen. Muussa tapauksessa materiaalin pinta antaa oman värivaikutelmansa spekuläärille heijastukselle.

Blinnin malli on toteutettu kappaleessa 2.4 esitetyn kaavan (13) mukaan:

$$\mathbf{I} = \mathbf{P}_a^T \mathbf{I}_A + \mathbf{I}_L^T \left(\mathbf{P}_d (\mathbf{N} \cdot \mathbf{L}) + \mathbf{F} \frac{DG}{(\mathbf{N} \cdot \mathbf{E})} \right) \quad (74)$$

jossa

$\mathbf{I}_A, \mathbf{I}_L$ ovat ambientin valon ja valonlähteen spektrit,

$\mathbf{P}_a, \mathbf{P}_d$ ovat materiaalin ambientin ja diffuusin heijastuksen spektrivektorit,

$\mathbf{N}, \mathbf{L}, \mathbf{E}$ ovat pinnan normaalin, valon tulosuunnan sekä tarkastelusuunnan suuntavektorit,

D on mikropintojen distributio,

G on geometriatermi, joka on laskettu kuten kappaleessa 2.4.1

johdettiin ja

\mathbf{F} on Fresnell heijastusvektori jokaiselle spektrin arvolle.

Distribuutiotermi on laskettu pinnan normaalin \mathbf{N} ja tarkasteltavien mikropintojen suuntavektorin \mathbf{H} avulla:

$$D = (\mathbf{N} \bullet \mathbf{H})^n = \left(\mathbf{N} \bullet \frac{\mathbf{L} + \mathbf{E}}{|\mathbf{L} + \mathbf{E}|} \right)^n \quad (75)$$

Kaavasta (75) distribuutiotermi on samaa muotoa kuin Phongin mallin (73) spekulaa-ritermi. Kaavassa (74) Fresnell termi on määritetty kuten kappaleessa 2.4.2 määrittämällä ensin materiaalin taitekerroin ja keskiarvoheijastus:

$$F_{ave} = \frac{1}{n} \sum_{i=1}^n F_i \quad (76)$$

jossa

F_i on käyttäjän antaman spekulaa-ritermi heijastusvektorin i :s termi ja

n on spektrivektorin dimensio.

Keskiarvotaitekerroin saadaan keskiarvoheijastuskertoimen avulla suoraan kaavasta (28). Näiden jälkeen voidaan määrittää Fresnel-heijastuksen \mathbf{F} spektrivektori:

$$\mathbf{F} = \mathbf{P}_s + (\mathbf{1} - \mathbf{P}_s) \left[\frac{F - F_{ave}}{1 - F_{ave}} \right] \quad (77)$$

jossa

\mathbf{P}_s on käyttäjän antama spekulaa-ritermi heijastuksen kerroin eli materiaalin heijastusspektri, kun tarkastelukulma on nolla,

F_{ave} saadaan kaavasta (76) ja

F saadaan laskettua kappaleen 2.1 kaavoista (3), (4) ja (5) seuraavasti

$$F = \frac{1}{2} \left\{ \left[\frac{n_{ave} (\mathbf{N} \cdot \mathbf{L}) + (\mathbf{N} \cdot \mathbf{T})}{n_{ave} (\mathbf{N} \cdot \mathbf{L}) - (\mathbf{N} \cdot \mathbf{T})} \right]^2 + \left[\frac{(\mathbf{N} \cdot \mathbf{L}) + n_{ave} (\mathbf{N} \cdot \mathbf{T})}{(\mathbf{N} \cdot \mathbf{L}) - n_{ave} (\mathbf{N} \cdot \mathbf{T})} \right]^2 \right\} \quad (78)$$

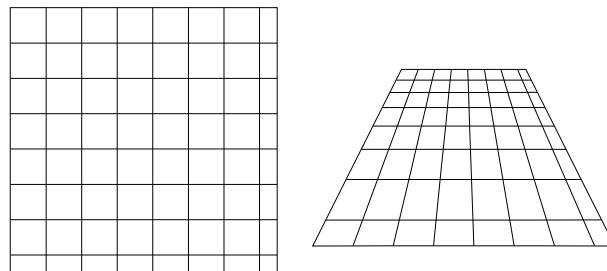
Kappaleessa 2.4 on kaavassa (13) $\max()$ termi, joka on jätetty pois kaavasta (74) tarpeettomana. Valonlähteen oletetaan aina olevan tarkasteltavan pinnan yläpuolella. Jos pinnalle ei osu valonlähteen suora valo, otetaan huomioon ainoastaan ambientti termi $\mathbf{P}_a^T \mathbf{I}_A$.

Kun tarkastelukulma pinnan normaaliin nähden lähenee 90 astetta, termi $\mathbf{N} \cdot \mathbf{E}$ lähestyy nollaa. Tästä johtuen spekulaarinen heijastuksen termi lähestyy ääretöntä, jos tarkastelukulma lähestyy suoraa kulmaa. Tämä tarkoittaisi sitä, että pinnasta heijastuu enemmän valoa kuin siihen tulee. Tämä ei pidä paikkaansa, koska tarkasteltavat pinnat eivät ole valonlähteitä. Kaava (74) supistetaan muotoon

$$\mathbf{I} = \mathbf{P}_a \mathbf{I}_A + \mathbf{I}_L \quad (79)$$

kun jakaja $\mathbf{N} \cdot \mathbf{E}$ saa pienempiä arvoja kuin 0.0001. Tällöin heijastuu kaikki valonlähteestä tuleva valo tarkastelusuuntaan. Tämä ei kuitenkaan pidä täysin paikkansa, muuten kuin valonlähteen tulokulmalla, joka on 90 astetta pinnan normaaliin nähden ja vastakkaiseen suuntaan tarkastelukulmaan nähden. Jos valonlähteen tulokulma on eri, syntyy generoitavaan kuvaan virheellisiä vaaleita kohtia, jotka heijastavat kaiken valon valonlähteestä tarkastelusuuntaan.

Suuria tarkastelukulmia pitää välttää, koska paperipintaa generoitaessa ei ole otettu huomioon pinnan kiertoa tasojen suuntaisten akselien suuhteen tai perspektiivimuunnosta. /Kuva 21/ Tästä löytyy tarkemmin alan kirjallisuudesta [8] [14]. Koska minikäänlaista kallistusta tai kiertoa tarkasteltavalle pinnalle ei tapahdu, kohde ei näytä aidolta tarkastelukulmilla, jotka ovat nolasta poikkeavia. Pienillä tarkastelukulmilla ei näy eroa todellisuudessakaan, mutta kun tarkastelukulma lähenee 90 astetta, erot kasvavat suuriksi todellisuuden ja generoidun kuvan välillä.



Kuva 21 Tarkastelusuunnan muutoksessa aiheutuvat muutokset pinnalle

5.4 Värin implementointi

Heijastusmallissa väri lasketaan spektrivektoreista. Kun heijastusmallin avulla on saatu laskettua heijastusspektri, muutetaan tulokseksi saatu spektri RGB-vektoriksi ja tallennetaan kuvatiedostoon. Spektrivektori sisältää aallonpituuksien arvot välillä 380 nm ja 780 nm nanometrin välein.

Spektrin muuntaminen RGB-arvoiksi tapahtuu ensin muuttamalla spektri tristimulus-arvoiksi, jotka vuorostaan muutetaan RGB arvoiksi. Spektrin muuttaminen tristimulusarvoiksi tapahtuu sovittamalla spektri tristimuluskäyrälle. Tristimuluskäyrinä on käytetty CIE standardin mukaisia käyriä, jotka löytyvät Juddin ja Wyszeckin kirjasta ^[20]. Tristimulusarvot on laskettu seuraavanlaisella yhtälöllä

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{XYZ} = s \mathbf{I}^T \begin{bmatrix} \mathbf{X}_{tr} & \mathbf{Y}_{tr} & \mathbf{Z}_{tr} \end{bmatrix} \quad (80)$$

jossa

\mathbf{I} on spektri, jonka tristimulusarvoa lasketaan,

s on skaalaustekijä ja

$\mathbf{X}_{tr}, \mathbf{Y}_{tr}, \mathbf{Z}_{tr}$ ovat ihmisen näkökykyä vastaavat tristimulusspektrit.

Skaalaustekijä s on laskettu \mathbf{Y}_{tr} spektrin avulla:

$$s = \frac{1}{\sum_i Y_i} \quad (81)$$

jossa

Y_i on tristimulusvektorin \mathbf{Y}_{tr} paikalla i oleva arvo.

Kun tristimulusarvo on laskettu värispektrille, muutetaan tristimulusarvo xyz monitorin käsittelemään RGB muotoon. Tämä tapahtuu yksikertaisen muunnosmatriisin avulla:

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \Lambda^{-1} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (82)$$

jossa

Λ on konversio matriisi RGB arvojen muuttamiseksi tristimulus-arvoiksi.

Ohjelman implementoinnissa on käytetty NTSC standardin mukaista konversiomatriisia /Liite 1/. Konversiomatriisi on riippuvainen käytettävästä monitorista ja sen kyvystä toistaa väriavaruutta. Ennen kun *rgb*-arvo voidaan tulostaa, täytyy se muuttaa väriavaruuden alueelle, jonka monitori pystyy näyttämään. Yksinkertainen väriavaruuden kutistaminen voitaisiin tehdä poistamalla ylisuuret punaisen, vihreän tai sinisen arvot. Tämä kuitenkin aiheuttaisi värivirheitä. Tässä sovellutuksessa on pyritty säilyttämään värisävyn arvo ja pienennetty intensiteettiä. Tämä tapahtuu pienentämällä kaikkia *rgb*-värivektorin arvoja samassa suhteessa:

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix}_{out} = \mathbf{D}_{axis} + d \begin{bmatrix} \Delta r \\ \Delta g \\ \Delta b \end{bmatrix} \quad (83)$$

jossa

d on kerroin, jonka perusteella määrätään intensiteetin pieneneminen tai suureneminen ja

\mathbf{D}_{axis} on *rgb* värivektorin suunta, joka saadaan seuraavasta kaavasta

$$\mathbf{D}_{axis} = \begin{bmatrix} (r + g + b) / 3 \\ (r + g + b) / 3 \\ (r + g + b) / 3 \end{bmatrix} \quad (84)$$

Kaavassa (83) oleva differenssi *rgb*-vektori saadaan laskettua seuraavasta kaavasta

$$\begin{bmatrix} \Delta r \\ \Delta g \\ \Delta b \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix} - \mathbf{D}_{axis} \quad (85)$$

Kaavassa (83) oleva d kerroin on yksi, jos kaikki rgb -vektorin arvot ovat välillä $[0,1]$. Muussa tapauksessa kerroin d sovitetaan niin, että rgb -vektorin arvot sijoittuvat monitorin toiminta-alueelle välille $[0,1]$ /Esimerkki 6/:

$$d_0 = 1 \tag{86}$$

$$d_n = \begin{cases} \frac{-D_i}{\Delta x} d_{n-1}, x < 0 \\ d_{n-1}, 0 < x < 1 \\ \frac{1 - D_i}{\Delta x} d_{n-1}, x > 1 \end{cases}$$

jossa

x on jokin rgb -vektorin väriarvoista,

Δx on vastaava differenssivektorin arvo ja

D_i on kaavan (84) \mathbf{D}_{axis} vektorin alkion termi.

Kaava (86) iteroidaan läpi kaikkien rgb -vektorin arvojen.

```
diff_mult = 1.0;

if (in_rgb.r > 1.0)
    {if((tmp_mult=(MAX_CLIP-axis_rgb)/diff.r)<diff_mult)
        diff_mult = tmp_mult;}
else if (in_rgb.r < 0.0)
    {if((tmp_mult=(MIN_CLIP-axis_rgb)/diff.r)<diff_mult)
        diff_mult = tmp_mult;}

if (in_rgb.g > 1.0)
    {if((tmp_mult=(MAX_CLIP-axis_rgb)/diff.g)<diff_mult)
        diff_mult = tmp_mult;}
else if (in_rgb.g < 0.0)
    {if((tmp_mult=(MIN_CLIP-axis_rgb)/diff.g)<diff_mult)
        diff_mult = tmp_mult;}

if (in_rgb.b > 1.0)
    {if((tmp_mult=(MAX_CLIP-axis_rgb)/diff.b)<diff_mult)
        diff_mult = tmp_mult;}
else if (in_rgb.b < 0.0)
    {if((tmp_mult=(MIN_CLIP-axis_rgb)/diff.b)<diff_mult)
        diff_mult = tmp_mult;}
```

Esimerkki 6 Kertoimen d määrittäminen. MAX_CLIP ja MIN_CLIP ovat suurin arvo ja pienin arvo resoluution rajoissa

5.5 Paperin karheusmalli

Paperin karheus on mallinnettu karheustekstuurina, joka generoidaan karheusmatriisin avulla. Karheustekstuurin avulla saadaan paperin mikropintojen normaalit lasketua.

Koordinaatisto, jossa tarkastellaan paperin pintaa, on xyz-koordinaatisto. Xy-taso on asetettu tarkasteltavan paperipintan tasoon niin, että paperin pinnan normaalivektori on z-akselin suuntaan. Tarkasteltava paperipinta kulkee origon kautta. Nämä oletukset on asetettu helpottamaan vektorilaskentaa.

5.5.1 Karheusmatriisi

Pinnan karheusvaihtelut on mallinnettu matriisilla, joka sisältää pisteiden korkeusvaihtelun /Kuva 22/. Karheusmatriisi on kaksidimensioinen matriisi, jonka koon ja resoluution käyttäjä voi määrätä.

Karheusmatriisiin on tallennettu tietyn kokoinen alue, jota käytetään tekstuurina paperin pintaan generoitaessa. Pisteiden korkeudet on digitoitu kahdeksalla bitillä eli pisteiden korkeudet matriisissa ovat lukuja välillä [0,255]. Mitä suurempi on matriisin pisteen arvo, sitä korkeammalla piste on. Todellisen korkeuden ilmoittaa maksimi korkeus, joka kertoo todellisen etäisyyden välillä [0,255]. Todellinen korkeus z saadaan seuraavasti

$$z = f(x, y) = z_{max}g(x, y) \quad (87)$$

jossa

$f(x, y)$ on funktio joka antaa todellisen korkeuden,

z_{max} on fyysinen maksimietäisyys minimin ja maksimin välillä ja

$g(x, y)$ antaa (x, y) pisteen arvon karheusmatriisissa

Funktio $g(x, y)$ antaa (x, y) pistettä lähimpänä olevan karheusmatriisin pisteen (x_k, y_k) arvon. Funktion toteutuksessa ei olla interpoloitu arvoja matriisipisteiden välillä, joten korkeusarvoissa z esiintyy virheitä.

$$g(x, y) = \mathbf{K}\left(\text{round}\left(\frac{x \bmod a}{\Delta x}\right), \text{round}\left(\frac{y \bmod b}{\Delta y}\right)\right) \quad (88)$$

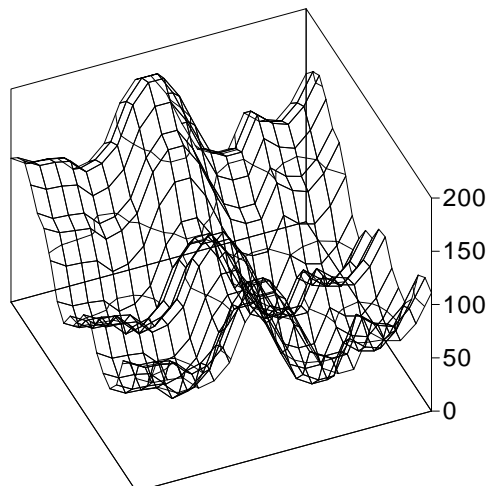
jossa

\mathbf{K} on karheusmatriisi, jonka elementtiä haetaan,

a, b ovat matriisin \mathbf{K} peittämän alueen fyysiset koot ja

$\Delta x, \Delta y$ ovat elementtien x ja y suuntaisten etäisyyksien fyysiset välit matriisissa \mathbf{K} .

Tässä toteutuksessa Δx ja Δy ovat yhtäsuuria.



Kuva 22 Profiilikuva 20 kertaa 20 pistettä kokoisesta karheusmatriisista. Karheusmatriisi on tehty todellisen paperin karheusprofiilista.

Karheusmatriisin alkiot on tallennettu kahdeksalla bittillä. Tällä ratkaisulla pyritään

vähentämään tietokoneen muistikapasiteetin tarvetta. Jos karheusmatriisi esitetään sata kertaa sata kokoisella pikseli matriisilla, tämä tarvitsee 10000 tavua tilaa työmuistista. Jos sama matriisi esitetään `double` tyyppin liukulukuina, se veisi kahdeksan kertaa suuremman tilan muistista eli noin 80 kilotavua. `Double` tyyppi varaa muistia 64 bittiä.

Karheusmatriisiin liittyy läheisesti mikropintojen distribuutio. Karheusmatriisista saadaan mikropintojen normaalit, kun taas mikropintojen distribuutio kuvaa tarkasteltavan kohdan heijastusominaisuuksia. Mikropintojen distribuutio on toteutettu matriisilla **B**, joka sisältää peiliheijastuksen intensiteetin puoliintumiskulmien arvoja (kappale 2.4.3). Tämän puoliintumiskulmamatriisin avulla voidaan liittää paikallisia kiilto-ominaisuuksia paperin pintaan. Matriisia **B** käsitellään samaan tapaan kuin karheusmatriisia **K**.

Paperipinnan mallille voidaan määrittää kontaktipinnan taso. Kontaktipinnan taso määritetään mikrometreinä kuten maksimikarheuskin. Kontaktipinta ilmoittaa etäisyyden maksimipisteestä, jonka arvo on 255 karheusmatriisissa. Kontakti tapahtuu kun

$$f(x, y) \geq z_{max} - z_c \quad (89)$$

jossa

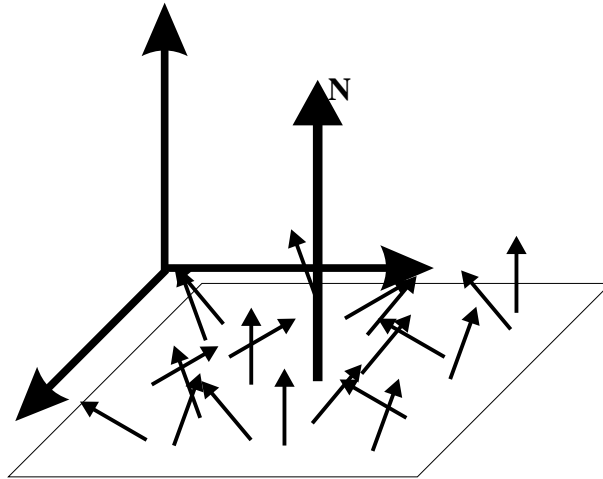
$f(x,y)$ on tarkasteltavan pisteen (x,y) karheus,

z_{max} on fyysinen maksimikarheus karheusmatriisissa ja

z_c on kontaktipinnan etäisyys maksimikarheudesta.

5.5.2 Pinnan normaalin määrittäminen

Koska tarkasteltava paperin pinta on makromittakaavassa taso, on myös pinnan normaali koko pinnan yli vakiollinen. Kun tarkastellaan yhtä pientä aluetta paperilla, on pinnan normaalilla vaihtelua paperin karheusrakenteen aiheuttaman struktuurin mukaisesti. Paperin karheusrakenteen voidaan olettaa olevan häiriö, joka aiheuttaa pinnan normaalissa vaihtelua. /Kuva 23/ Tämä häiriö voidaan generoida karheusmatriisista, josta tarkemmin kappaleessa 5.5.1 .



Kuva 23 Paperipinnan normaali ja karheustekstuurin aiheuttamat häiriövektorit

Häiriösuunta on tangentiaalitasen normaali tarkasteltavassa pisteessä. Tangentiaalitasen normaali voidaan laskea, kuten kappaleessa 2.5 käsiteltiin, pinnan osittaisderivaattojen avulla. Tarkasteltavan pinnan häiriön osittaisderivaatat x :n ja y :n suhteen on aproksimoitu seuraavasti differenssien avulla

$$P_x[f(x, y)] = \frac{1}{2} (f(x - \Delta x, y) - f(x + \Delta x, y)) \quad (90)$$

$$P_y[f(x, y)] = \frac{1}{2} (f(x, y - \Delta y) - f(x, y + \Delta y)) \quad (91)$$

jossa

(x, y) on tarkasteltava piste,

$f(x, y)$ on häiriöfunktio, joka saadaan karheusmatriisista ja

Δx on differenssin askel, joka tässä tapauksessa on karheusmatriisin pikselikoko.

P_x :n ja P_y :n avulla voidaan laskea vektorit

$$\mathbf{P}_x = \begin{bmatrix} \Delta x \\ 0 \\ P_x \end{bmatrix}, \mathbf{P}_y = \begin{bmatrix} 0 \\ \Delta y \\ P_y \end{bmatrix} \quad (92)$$

joiden avulla voidaan määrittää häiriövektori \mathbf{P} ristitulona kuten kaavassa (38) on tehty. Kaavassa (92) kerroin P_x :n ja P_y :n edessä

$$\mathbf{P} = \mathbf{P}_x \otimes \mathbf{P}_y = \begin{bmatrix} -\Delta y P_x \\ -\Delta x P_y \\ \Delta x \Delta y \end{bmatrix} \quad (93)$$

Pinnan normaali \mathbf{N}' , johon on vaikuttanut häiriöfunktio voidaan laskea P_x :n ja P_y :n avulla kaavoista (44) ja (45):

$$\mathbf{N}' = \mathbf{N} + P_x (\mathbf{N} \otimes \mathbf{Q}_y) - P_y (\mathbf{N} \otimes \mathbf{Q}_x) \quad (94)$$

Kun kaavassa (94) \mathbf{N} on z -akselin suuntainen yksikkövektori, niin \mathbf{Q}_x ja \mathbf{Q}_y ovat x - ja y -akselin suuntaisia yksikkövektoreita. Tällöin saadaan uudeksi pinnan normaaliksi

$$\mathbf{N}' = \begin{bmatrix} -P_x \\ -P_y \\ 1 \end{bmatrix} \quad (95)$$

Kaavojen (93) ja (95) yhdenmuotoisuuden perusteella voidaan käyttää suoraan häiriövektoria \mathbf{P} pinnan normaalina, koska alkuperäinen pinnan normaalivektori \mathbf{N} on koko tarkasteltavan pinnan yli z -akselin suuntainen yksikkövektori.

Koska karheusmatriisi on kooltaan rajallinen, syntyy epäjatkuvuuskohtia paikkoihin, jossa siirrytään uuteen karheusmatriisin monikertaan. Epäjatkuvuuskohtien yli kaavoilla (90) ja (91) laskettu differenssi antaa virheellisiä häiriövektoreita. Epäjatkuvuuskohdat on otettu huomioon reunapisteissä. Jos piste (x_0, y_0) on reunapiste, siirretään piste (x_0, y_0) matriisin reunalta pois:

$$x = \begin{cases} x = 1; (x + 1) \\ 0 < x < a; x \\ x = a; (x - 1) \end{cases} \quad (96)$$

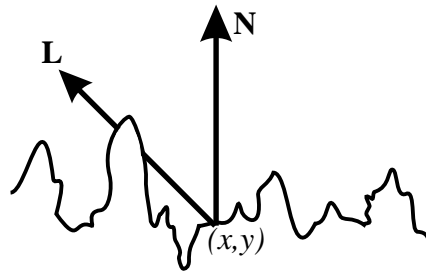
$$y = \begin{cases} y = 1; (y + 1) \\ 0 < y < b; y \\ y = b; (y - 1) \end{cases} \quad (97)$$

jossa

a, b ovat karheusmatriisin koot.

5.5.3 Itsensä varjostus ja maskaus

Kun valo tulee vinosti pinnan normaaliin nähden, syntyy itsensä varjostumisilmiö (selfshadow). /Kuva 24/ Tämä aiheuttaa varjoja tarkasteltavalle pinnalle. Samanlainen ilmiö syntyy, kun tarkastelusuunta on vino pinnan normaaliin nähden. Tällöin ei nähdä tarkasteltavaa pistettä vaan pistettä varjostava kohta.



Kuva 24 Itsensä varjostusilmiö, pinta estää valon tulon pisteeseen (x, y) , kun valon tulosuunta on L

Itsensä varjostumisilmiössä haetaan suoran $z=L(x, y)$ ja pinnan $z=f(x, y)$ leikkauspisteitä (x, y, z) :

$$(L(x, y) = f(x, y)) > z_0 \quad (98)$$

jossa

z_0 on tarkasteltavan pisteen (x_0, y_0) korkeus.

Kun yhtälölle (98) löytyy ratkaisu, varjostaa pinta itseään. Vastaava yhtälö voidaan muodostaa maskauksen tarkasteluun. Maskauksen ja itsensävarjostuksen implementointi eroaa siinä, että maskauksessa haetaan näkyvää pistettä ja itsensävarjostuksessa haetaan varjostusta tarkasteltavalle pisteelle. Yhtälölle (98) voidaan hakea ratkaisu iteratiivisesti:

$$x_n = x_{n-1} + \Delta x_L \quad (99)$$

$$y_n = y_{n-1} + \Delta y_L \quad (100)$$

$$f(x_n, y_n) \geq L(x_n, y_n) \quad (101)$$

Kaavoissa (99) ja (100) on askeleet Δx_L ja Δy_L , joiden suuruudesta riippuu kuinka suuri virhe syntyy iteroitaessa. Askeleiden Δx_L ja Δy_L suhde määräytyy valontulosuunnan osoittavasta vektorista \mathbf{L} :

$$\Delta y_L = \frac{L_j}{|L_i|} \Delta x \quad (102)$$

Kun ratkaistaan kaavaa (98), käytetään kaavassa (101) suurempi tai yhtäsuuri kuin merkkiä, koska ratkaisu algoritmi on iteratiivinen. Tällöin löytyy kohta, jossa valo osuu tarkasteltavaan pintaan. Piste (x_0, y_0) on tarkasteltavan pinnan kohta, josta lähdetään liikkeelle valonlähde kohden, jonka suunnan määrää valon tulosuunnan osoittava vektori \mathbf{L} . Kaavan (101) pinnan korkeus $f(x_n, y_n)$ saadaan yhtälöistä (87) ja (88) ja valonsäteen korkeus kohdassa (x_n, y_n) saadaan seuraavasti

$$L(x_n, y_n) = L_k \sqrt{(x_n - x_0)^2 + (y_n - y_0)^2} + z_0 \quad (103)$$

jossa

L_k on vektorin \mathbf{L} z -akselin suuntainen komponentti.

Maskauksessa jatketaan iterointia kunnes karheusmatriisin maksimikorkeus on ylitetty. Iteroinnin kuluessa pistetään muistiin viimeinen suoran $z=L(x,y)$ ja pinnan $z=f(x,y)$ ratkaisu, jolloin saadaan selville varsinaisesti nähtävä piste.

5.6 Painoväriin leviämismallin implementointi

Painoväriin liittäminen generoituun paperimalliin tapahtuu tarkastelemalla painoväriin leviämistä paperipinnalle sekä painoväriin valon absorptiota ja heijastumista. Painoväri on toteutettu matriisina \mathbf{I} , joka kuvaa generoitavan kuvan rasterirakennetta. Tämän matriisin käsittely tapahtuu samalla tavalla kuin karheusmatriisin käsittely.

Painoväriin leviäminen lasketaan kaavasta, jolla saadaan selville painoväriin paksuus kohdassa (x,y) :

$$I(x, y) = A(x, y) i(x, y) \otimes h(x, y) [s + dz(x, y)] \quad (104)$$

jossa

$i(x, y)$ on painoväriin rasteristikuva-funktio, joka lasketaan matriisista **I**,

$A(x, y)$ on painoväriin kontaktifunktio,

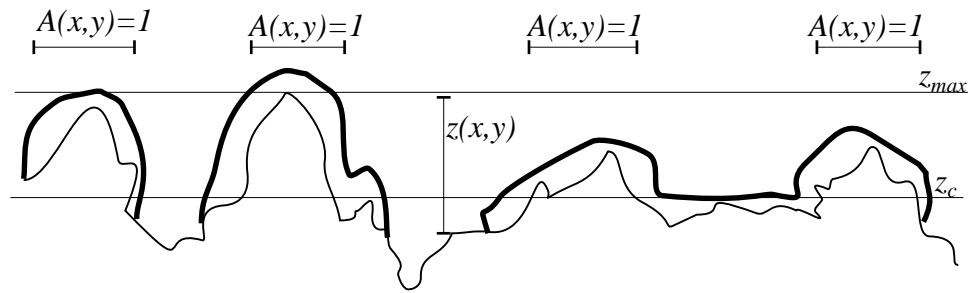
$h(x, y)$ on konvoluutiofunktio,

s on splitting-kerroin,

d on depositiokerroin ja

$z(x, y)$ on paperin karheussyvyys maksimikorkeudesta.

Kontaktifunktio $A(x, y)$ saa joko arvon nolla tai yksi riippuen painoväriin ja paperin kontaktista /*Kuva 25*/.



Kuva 25 Painoväriin leviäminen paperipinnalle, jonka kontaktitaso on z_c

$$A(x, y) = \begin{cases} 0, & z_{max} - f(x, y) > z_c \\ 1, & z_{max} - f(x, y) \leq z_c \end{cases} \quad (105)$$

jossa

$f(x, y)$ on paperin karheusfunktio,

z_{max} on maksimikarheus.

z_c on kontakti etäisyys maksimikorkeuteen nähden.

Kaavassa (104) funktio $z(x, y)$ lasketaan seuraavasti paperin karheudesta.

$$z(x, y) = z_{max} - f(x, y) \quad (106)$$

Konvoluutiofunktio $h(x,y)$ kaavassa (104) on toteuttu matriisina. Konvoluutio lasketaan pisteelle (x,y) konvoluutiomatriisin keskimmäisen pisteen suhteen, joka asetetaan pisteeseen (x,y) . Konvoluutiomatriisin alkioden summan tulee olla yksi, koska leviämässä ei synny eikä vähene painoväriä. Konvoluutiolauseke konvoluutiomatriisin avulla laskettuna voidaan kirjoittaa seuraavaan muotoon

$$i(x,y) \otimes h(x,y) = \sum_{k=-a}^a \left(\sum_{l=-b}^b i(x-k, y-l) H(a-k, b-l) \right) \quad (107)$$

jossa

$H(x,y)$ on konvoluutiomatriisin arvo pisteessä (x,y) ja konvoluutiomatriisin koko on $(2a+1) \times (2b+1)$.

Painovärimallin leviäminen on toteuttu niin, että mallin leviäminen lasketaan ainoastaan painojäljen sisällä. Tämä tarkoittaa sitä, että painoväriä ei varsinaisesti leviä painettavan alueen ulkopuolelle.

Painojäljestä heijastunut pinta lasketaan paperin ja painoväriin heijastusominaisuuksien perusteella. Painojäljen heijastusominaisuudet riippuvat painoväriin kerroksen paksuudesta paperin pinnalla sekä absorbtio-ominaisuuksista. Painovärikerroksen valon transmissio saadaan laskettua kaavasta

$$T(x,y) = e^{-2kI(x,y)} \quad (108)$$

jossa

k on painoväriin valon absorptiokerroin ja

$I(x,y)$ on painovärikerroksen paksuus tarkasteltavassa pisteessä.

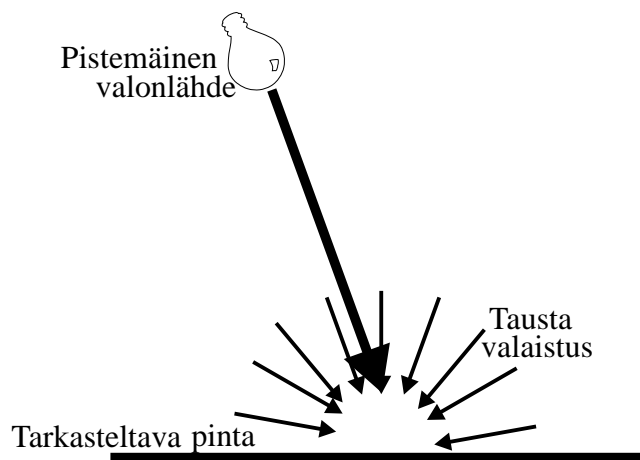
Kaavassa (108) kerroin kaksi johtuu siitä, että valo läpäisee painovärikerroksen kaksi kertaa. Ensin valo kulkee painovärikerroksen läpi paperin pintaan, josta tapahtuu heijastus ja valo kulkee uudestaan painovärikerroksen. Diffuusin heijastuksen spektrikerroin \mathbf{P}_s saadaan laskettua transmission avulla:

$$\mathbf{P}_d(x,y) = T(x,y) \mathbf{P}_{d,paper} + (1 - T(x,y)) \mathbf{P}_{d,ink} \quad (109)$$

Kaavalla (109) saadaan paikasta (x,y) riippuvainen diffuusin heijastuksen kerroin. Vastaavasti kuin diffuusin heijastuksen kertoimen ohjelma laskee ambientin heijastuksen kertoimen. Spekulaarin heijastuksen kerroin on suoraan painoväriin spekulaarin heijastuksen kerroin, koska spekulaariheijastus tapahtuu materiaalin pinnasta.

5.7 Valonlähteen mallittaminen

Tarkasteltavalle pinnalle tuleva valo on mallitettu pistemäisenä valonlähteenä sekä taustavalaistuksena /*Kuva 26*/. Pistemäisen valonlähteen etäisyys on äärettömän kaukana tarkasteltavasta pinnasta ja valonlähde säteilee tasaisesti joka puolelle ympärilleen. Taustavalaistuksen oletetaan tulevan tasaisesti joka suunnasta tarkasteltavaan pintaan nähden. Valon väriominaisuudet on implementoitu spektrivektorina, josta tarkemmin kappaleessa 5.4. Pistemäisellä valonlähteellä ja taustavalaistuksella on molemmilla oma spektrivektori. Pistemäisellä valonlähteellä generoidaan spekulaarit intensiteettiipiikit ja diffuusit ominaisuudet tarkasteltavan pinnan heijastusominaisuuksiin. Taustavalaistuksella määrätään tarkasteltavan pinnan heijastuksen perustaso.



Kuva 26 Valonlähteen ja taustavalaistuksen suunta tarkasteltavaan pintaan nähden

Valon mallituksessa on suoritettu joukko yksinkertaistuksia todellisiin valonlähteisiin nähden, jotta pystyttäisiin vähentämään valon mallin laskennallista raskautta. Koska ongelmana oli tarkastella paperipinnan heijastusominaisuuksia eikä niinkään valon ominaisuuksia, voidaan näitä yksinkertaistuksia tehdä. Vallitsevan valon ominaisuudet liittyvät tosin läheisesti pinnan antamaan heijastusefektiin, jolloin kompromissina tehty valomalli ei anna välttämättä tarpeeksi todellista vaikutelmaa tarkasteltavan pinnan heijastusominaisuuksista.

Etäisyysoletuksen ansiosta voidaan käyttää vakiollista suuntavektoria valonlähteelle koko tarkasteltavan pinnan yli. Normalisoitu suuntavektori valonlähteelle voidaan laskea valonlähteen xyz -koordinaatiston sijainnin ja tarkasteltavan pisteen perusteella:

$$\mathbf{L}_{(a,b)} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_L = \sqrt{(X-a)^2 + (Y-b)^2 + Z^2} \left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - \begin{bmatrix} a \\ b \\ 0 \end{bmatrix} \right) \quad (110)$$

jossa

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \text{ on valonlähteen sijainti vektori origon suhteen ja}$$

(a,b) on pisteen sijainti tarkasteltavalla tasolla.

Kun valonlähde on äärettömän kaukana, voidaan valonlähde olettaa pistemäiseksi. Valonlähteen pistemäisyydestä johtuen voidaan käyttää yhtä vakiollista suuntavektoria osoittamaan valonlähteen sijainti. Kaavaa (110) voidaan yksinkertaistaa, kun tarkasteltava piste voidaan siirtää origoon:

$$\mathbf{L} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_L = \sqrt{X^2 + Y^2 + Z^2} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (111)$$

Todellisilla valonlähteillä on jokin suunta, johon ne heijastavat suurimman osan valoenergiastaan. Tämä johtuu todellisten valonlähteiden fysikaalisista ominaisuuksista kuten hehkulangan ominaisuuksista sekä valonlähteitä ympäröivistä esineistä kuten lampunvarjostimista. Käytetyssä valomallissa oletetaan valonlähteen säteilevän energiaansa tasaisesti joka suuntaan. Tämä ei vaikuta visuaaliseen lopputulokseen, koska valomallissa on oletettu valonlähteen sijaitsevan äärettömän kaukana. Tällöin voidaan käyttää vakiollista suuntavektoria valonlähteelle, josta seuraa että pinnalle saapuva valo on tasaista. Tästä johtuen suunnatun valon aiheuttama intensiteetin vaihtelu tarkasteltavalla pinnalla ei toteudu.

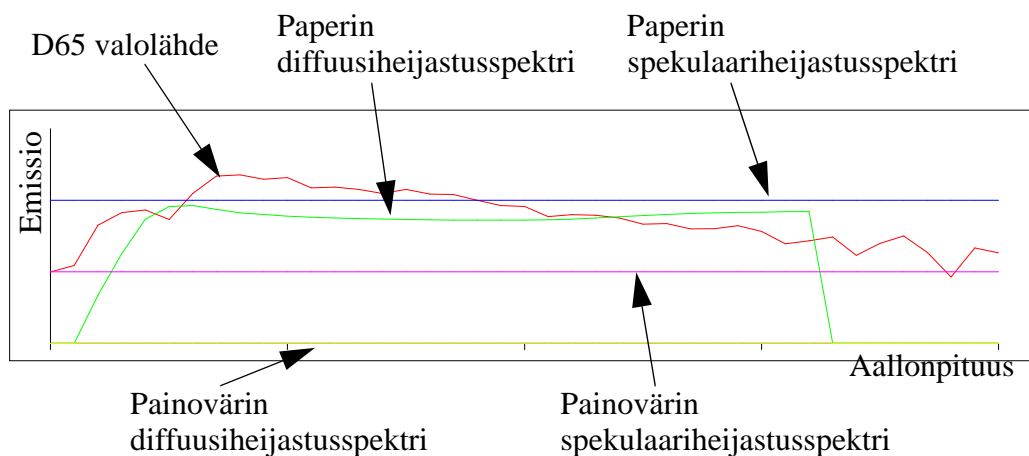
Pistemäinen valonlähde aiheuttaa mustia varjoja pinnalle. Todellisuudessa valo heijastuu pintaa ympäröivistä kappaleista, joka estää mustien kohtien syntymisen. Taustavalon on lisätty malliin poistamaan mustia kohtia, koska tarkasteltavan pinnan ympärille ei ole mallinnettu muita pintoja, jotka heijastaisivat valonlähteestä tulevaa valoa tarkasteltavalle pinnalle.

6 OHJELMALLA TEHTY KOESARJA

Tähän työhön tehdyllä ohjelmalla tehtiin koesarja, jolla tutkittiin mallien eri parametrien vaikutusta generoitavaan kuvaan. Täydellinen koesarja löytyy liitteestä 4. Generoitavat kuvat olivat 5 mm x 5 mm kokoisia, joiden resoluutio oli 20 μm . Paperipinnalle generoitiin 2 mm x 2 mm kompakti yksivärinen painoväripinta.¹

Koesarjassa generoiduille kuville ei suoritettu visuaalista testausta standardisoiduissa katseluolosuhteissa. Koesarjassa tutkittiin eri parametrien vaikutusta visuaaliseen ulkonäköön. Tulokset ovat subjektiivisia, joten ne vaativat lisäselvitystä. Pyrkimyksenä oli saada parametrien muuntelun avulla mahdollisimman aidon näköinen paperipinta, jolle oli generoitu mahdollisimman aidon näköinen painojälki. Kutakin mallin parametriä tarkasteltiin itsenäisesti oletusarvojen puitteissa. Ainoa poikkeus oli valon tulokulma, jota tutkittiin tarkastelukulman suhteen.

Valonlähteen spektrinä on käytetty kaikissa kuvissa samaa standardin mukaista D65 valonlähteen spektriä /Liite 1/. Paperin diffuusin ja ambientin valaistuksen heijastusspektrinä on käytetty normaalista kopiopaperista mitattua heijastusspektriä /Liite 1/. Paperin spekulaaarin heijastuksen spektrinä on käytetty tasaista spektriä, joka vastaa valkoista pintaa. Painoväriin diffuusin ja ambientin heijastuksen spekteinä on käytetty tasaista spektriä, joka vastaa mustaa. Painoväriin spekulaaarina heijastusspektrinä on käytetty tasaista valkoisen spektriä, joka heijastaa suurimman osan valoa valonlähteen värisenä. /Kuva 27/



Kuva 27 Painojäljen generoinnissa käytettyjen materiaalien spektrien vastet

Painojäljen genroinnissa käytettiin oletusarvoisesti tiettyjä mallien parametrejä, jotka on listattu taulukossa 1. Nämä oletusarvot on valittu siten, että jonkin parametrin muutos näkyisi helposti, kun verrataan oletusarvoilla tehtyä kuvaa ja uutta kuvaa.

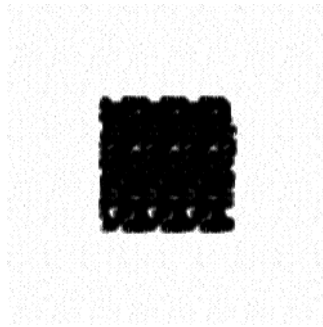
1. Kaikki tässä kappaleessa esitetyt testikuvat on suurennettu ja alunperin ne on tarkoitettu CRT-näyttöjärjestelmässä tarkasteltaviksi. Kuvat on esitetty tässä kappaleessa eri parametrien vaikutuksen havainnollistamiseksi.

Maksimi karheus ja valonlähteen intensiteetti ja kulma on valittu siten, että karheus näkyisi generoidusta kuvasta. Heijastusmallina on käytetty Blinnin heijastusmallia, jos heijastusmallilla ei ole ollut merkitystä tutkittavan ominaisuuden näkymiseen kuvassa. Tällaisia ominaisuuksia ovat esimerkiksi painoväriin kontaktikorkeus. Muussa tapauksessa kuvat on generoitu sekä Blinnin että Phongin heijastusmalleilla. Tarkastelugeometria on 45° valon tulosuunta ja 0° tarkastelusuunta, jos toisin ei ole ilmoitettu. Tarkastelugeometria on ilmoitettu tarkasteltavan pinnan normaalin suhteen.

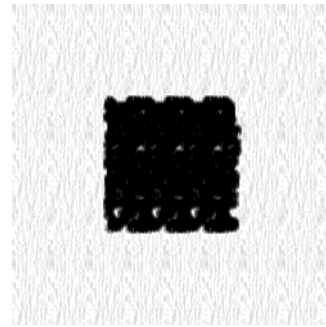
Taulukko 1 Käytetyt oletusarvot eri mallien parametreille

	Paperi	Painoväri	Valo
Resoluutio (Tuloskuva $20\ \mu\text{m}$)	$20\ \mu\text{m}$	$20\ \mu\text{m}$	---
Spekulaarin heijastuksen kerroin	0,60	0,80	---
Diffuusin heijastuksen kerroin	0,40	0,20	---
Ambientin heijastuksen kerroin	0,40	0,20	---
Beta kulma	0,30 rad	0,05 rad	---
Kontaktikorkeus	$4,0\ \mu\text{m}$	---	---
Maksimi karheus	$10,0\ \mu\text{m}$	---	---
Halkeamiskerroin	---	0,50	---
Depositiokerroin	---	0,50	---
Absorptiokerroin	---	0,90	---
Painoväriin paksuus	---	$1,0\ \mu\text{m}$	---
valonlähteen kulma tarkasteltavan pinnan normaaliin nähden	---	---	45°
valonlähteen intensiteettikerroin	---	---	3,00
Ambientin valaistuksen kerroin	---	---	0,60

Oletusarvoilla generoidut tuloskuvat näkyvät kuvassa 28. Blinnin malli luo sävykkäämmän kuvan kuin Phongin malli. Tämän vuoksi Blinnin mallilla saadaan karheusvaihtelu paremmin esille kuin Phongin mallilla.



Phongin heijastusmalli



Blinnin heijastusmalli

Kuva 28 Karheusinformaation näkyminen eri heijastusmalleilla

6.1 Valonmallin parametrit

Valonmallin parametrit ovat valon tulokulma ja pistemäisen valonlähteen ambientin valaistuksen kertoimet. Ensimmäisenä selvitettiin valon tulokulman vaikutus kuvan visuaaliseen ulkonäköön. Valon tulokulma muutettiin avaruuskoordinaateiksi niin, että valon tulokulma oli päinvastainen tarkastelukulmaan nähden. Avaruuskoordinaateiksi muuttaminen tehtiin seuraavan kaavan mukaan:

$$\begin{aligned} x &= -\cos(i) \\ y &= 0 \\ z &= \sin(i) \end{aligned} \tag{112}$$

jossa

i on valon tulokulma.

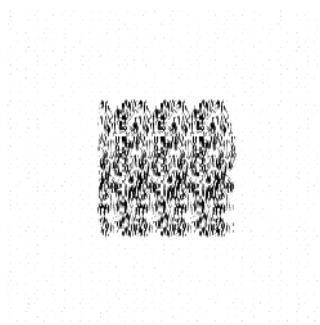
Taulukko 2 Käytetyt valonmallin parametrien arvot, kun tutkittiin valon tulokulman ja tarkastelukulman vaikutusta

Valon tulokulma	Diffuusi tarkastelukulma	Spekulaari tarkastelukulma
0°	0°	0°
25°	0°	25°
35°	0°	35°
45°	0°	45°
75°	0°	75°
85°	0°	85°

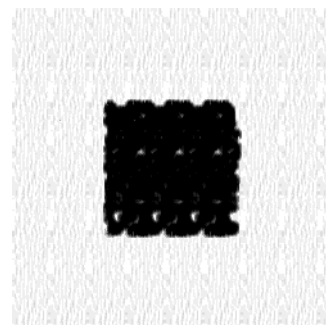
Kaavassa (112) ei ole otettu valonlähteen etäisyyttä huomioon, koska ohjelma olettaa valonlähteen olevan äärettömän kaukana. Tällöin voidaan käyttää vakiollista valovektoria, jonka kulma pinnan normaaliin nähden on haluttu i . Käytetyt kulmat valon tulosuunnalle i ja tarkastelukulmat on luetteloitu taulukossa 2. Kuvat generoitiin sekä Phongin että Blinnin heijastusmalleilla.

Pinnan itsensävarjostumisilmiö tuli selvästi esille suurilla valon tulokulmilla. Jo 75° asteen tulokulmalla itsensävarjostumisilmiö oli kuvassa huomattava. Blinnin heijastusmalli teki kuvasta huomattavasti tummemman kuin Phongin malli.

Kun tutkittiin spekulaaareilla heijastusgeometrialla pintaa, huomattavaa oli Phongin mallin hajottava vaikutus kuvaan. Phongin malli pirstoi kompaktin painojäljen /Kuva 29/. Blinnin malli toi esille valkoisia pisteitä, kun kulma oli 85°.



Phongin heijastusmalli



Blinnin heijastusmalli

Kuva 29 Peiliheijastussuunnan 45°/45° vaikutus Phongin ja Blinnin heijastusmalleilla

Valonlähteen mallin intensiteetin vaikutusta tutkittiin varioimalla pistemäisen valonlähteen kerrointa ja ambientin valaistuksen kerrointa. Kertoimet on lueteltu taulukossa 3. Jos kertoimet olivat liian suuria, kuvan paperipinnasta tuli helposti täy-

sin valkoinen ilman karheusstruktuuria. Tämä tapahtuu, kun ambienttikerroin saa arvon, joka ylittää rajan 1,0. Muuten kertoimia varioitiin niin, että päästäisiin mahdollisimman lähelle valkoista paperipintaa, jolla on karheusstruktuuri vielä näkyvissä.

Jos käytetään pelkkää ambienttia valaistusta eli pistemäisen valonlähteen kerroin on nolla, paperipinnan karheusstruktuuria ei saada esille. Pistemäinen valonlähde aiheuttaa varjot tarkasteltavalle pinnalle. Pistemäisen valonlähteen kerrointa voidaan vaihdella paljon vapaammin ilman että tarkasteltavasta paperipinnasta tulee valkoinen. Painovärin näkyvyyteen ei kertoimien vaihtelulla ollut kovin suurta vaikutusta, koska painovärin heijastusspektrien vasteet ovat niin pieniä verrattuna paperin heijastusspektrien vasteisiin.

Taulukko 3 Valonlähteen mallin kertoimet

Valonlähteen kerroin	Ambientti kerroin
0,0	1,0
0,0	4,0
2,0	2,0
2,5	0,5
3,0	0,5
3,0	0,6
3,0	0,7
3,0	1,0
3,0	3,0
4,0	0,0
4,0	1,0
4,0	2,0
4,0	3,0
4,0	4,0

6.2 Paperimallin parametrit

Paperimallin yhteydessä tutkitut parametrit ovat betakulma, kontaktikorkeus, maksimikarheus, spekkulaarin ja diffuusin heijastuksen kertoimet sekä karheusmatriisi. Näitä parametreja tarkasteltiin oletusarvoihin suhteen. Vain spekkulaarin heijastuksen kerrointa tutkittiin yhdessä diffuusin heijastuksen kertoimen kanssa, koska näiden suuruudet ovat riippuvaisia toisistaan.

Betakulman vaikutusta tutkittiin eri parametrien avulla, jotka on lueteltu taulukossa 4. Betakulma ilmaisee sen mikropintojen jakauman kulman, jossa peiliheijastuksen spekkulaari intensiteetti putoaa puoleen. Betakulma ei kuitenkaan kerro koko totuutta spe-

kulaarin intensiteetin suuruudesta, koska betakulma vaihtelee riippuen betakulmamatriisita. Tämä matriisi on käyttäjän määriteltävissä. Betakulmamatriisi määrää betakulman muutoksen tarkasteltavan pinnan yli. Tämä matriisi pidettiin vakiona jokaiselle betakulmalle.

Taulukko 4 Betakulman parametrit. Maksimi beta-arvo on $\pi/2$ rad

0,1 rad
0,3 rad
0,5 rad
0,8 rad
1,2 rad

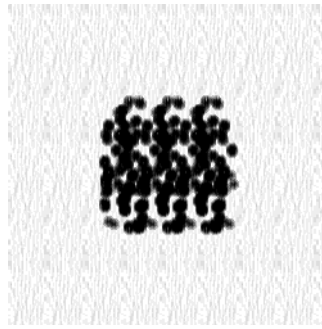
Betakulmien vaihtelulla ei saatu kuvissa merkittäviä muutoksia aikaan. Osaksi tämä johtuu tarkastelugeometriasta. Tarkastelugeometria oli diffuusi eli valontulokulma 45° ja tarkastelukulma 0° pinnan normaaliin nähden. Betakulma ilmaisee spekulaa- rin heijastuksen ominaisuuksia. Osaksi betakulmien vaihtelun huomaamattomuus joh- tui paperin karheusstruktuurista, joka peitti betakulman aiheuttamat muutokset omaan karheusinformaatioonsa.

Paperin kontaktikorkeus vaikuttaa painoväriin kontaktipinta-alaan paperin kanssa. Mitä suurempi kontaktikorkeus sitä suurempi pinta-ala. Referenssinä kontakti- korkeuden määrittämiseen on käytetty maksimikarheutta eli karheusstruktuurin kor- keinta kohtaa. Kontaktikorkeuden parametrin arvot ja suhteet maksimikarheuteen on lueteltu taulukossa 5.

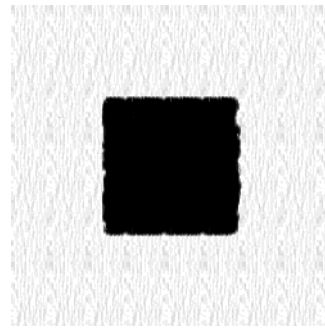
Taulukko 5 Kontaktikorkeuksien suhteet maksimikarheuteen ja arvot, kun maksimikarheus on $10,0 \mu\text{m}$

10/100	$1,0 \mu\text{m}$
30/100	$3,0 \mu\text{m}$
50/100	$5,0 \mu\text{m}$
80/100	$8,0 \mu\text{m}$
100/100	$10,0 \mu\text{m}$

Kontaktikorkeuden muutokset vaikuttavat selvästi kompaktin painojäljen yhtenäisyy- teen /Kuva 30/. Suurilla yli 50 prosentin kontaktikorkeuksilla painojälki on yhtenäi- nen. Painojäljen reunat suoristuvat ja lähestyvät ideaalista painojälkipintaa kontaktikorkeuden kasvaessa. Toisaalta tämä kompaktin painojäljen reunojen suoris- tuminen johtuu osittain algoritmien toteutuksesta. Pienillä kontaktikorkeuksilla saa- daan karheusmatriisin säännönmukaisuudet ja korkeat huippukohdat esille kuten kuvassa 30.



Kontaktikorkeus 3 μm



Kontaktikorkeus 5 μm

Kuva 30 Kontaktikorkeuden vaikutus painoväripinnan yhtenäisyyteen

Maksimikarheuden merkitys on selvä. Mitä karheampi pinta sitä suurempi on itsensä-varjostuksen merkitys. Tämä myös näkyi generoiduissa kuvissa. Karheammat kuvat olivat tummempia. Toisaalta paperin itsensävarjostukseen vaikuttaa myös karheusmatriisi. Maksimikarheuden arvot, joita tarkasteltiin, näkyvät taulukossa 6.

Taulukko 6 Tutkitut maksimikarheuksien arvot

5,0 μm
10,0 μm
16,0 μm
22,0 μm
40,0 μm

Paperin spekulaaaria ja diffusia parametriä tarkasteltiin yhdessä, koska molempien parametrien summan tulisi olla yksi eli

$$p_s = 1 - p_d \quad (113)$$

jossa

p_s on spekulaaarin heijastuksen kerroin ja

p_d on diffuusin heijastuksen kerroin.

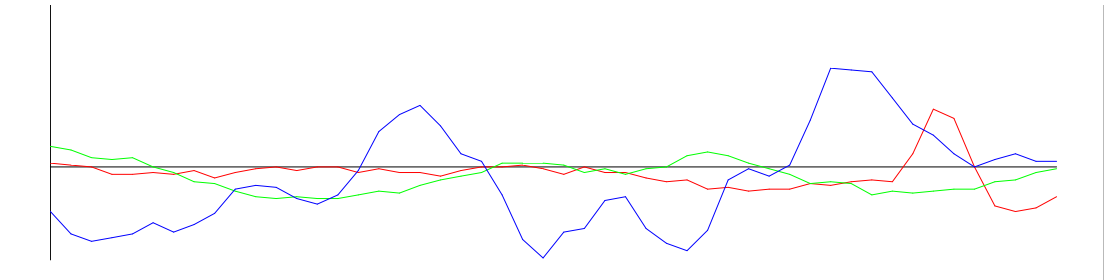
Spekulaaarin ja diffuusin parametrien arvot vaihtelivat välillä [0,1]. Ambientin heijastuksen kerroin sai saman arvon kuin diffuusin heijastuksen kerroin. Spekulaari parametri ei kuitenkaan voinut saada arvoa yksi toteutusteknisistä syistä /Taulukko 7/. Paperin vaaleus määräytyy spektristä, jota on käytetty. Mitä suurempi oli spekulaaarin heijastuksen kerroin sitä tummemmaksi kuva muodostui. Tämä johtuu diffuusin valon vähentymisestä ja varjojen jyrkkenemisestä. Toisaalta tarkastelugeometrian ollessa diffuusi eli valon tulokulma 45° ja tarkastelukulma 0° eivät spekulaaarit huippu-

valokohdat näkyneet kuvassa. Diffuusin kertoimen ollessa taas suuri tarkasteltava pinta näytti tasaisemmalta ja karheusstrukturi hävisi näkyvistä.

Taulukko 7 Spekulaarin ja diffuusin heijastuksen kertoimet

Spekulaari kerroin	Diffuusi kerroin
0,00	1,00
0,10	0,90
0,30	0,70
0,50	0,50
0,60	0,40
0,80	0,20

Karheusmatriisin vaikutusta tutkittiin käyttämällä erilaisista karheussignaaleista muodostettuja matriiseja. Lisäksi tutkittiin eri tapoja generoida karheusmatriisi. Karheussignaalien arvot venytettiin ensin välille $[0,255]$, jolloin voitiin käyttää kaikkia kahdeksan bitin sallimia signaalin tasoja hyväksi. Karheusmatriiseissa käytettiin kaikissa samaa $10,0 \mu\text{m}$ maksimikarheutta, jolloin signaalit olivat toisiinsa verrattavissa. Jos olisi haluttu todenmukaisia karheusmatriiseja, olisi maksimikarheus pitänyt määrätä karheussignaalin mukaan /Kuva 31/. Kuvassa 31 karheussignaaleista on kuvattu ainoastaan osa, mutta jo näistä kuvista saa käsityksen signaalien erilaisuudesta.



Kuva 31 Kuva karheussignaaleista

Karheusmatriisin generointi tapahtui paloittelemalla osa yksidimensionaalisesta karheussignaalista pienemmiksi yhtäpitkiksi vektoreiksi, jotka liitettiin karheusmatriisin sarakkeisiin vierekkäin. Toinen tapa millä karheusmatriisi tehtiin, oli kahden vektorin erottaminen karheussignaalista. Erotetut vektorit kerrottiin keskenään kuten kaavassa:

$$\mathbf{K} = \frac{1}{2} \mathbf{X}_1 \mathbf{X}_2^T \quad (114)$$

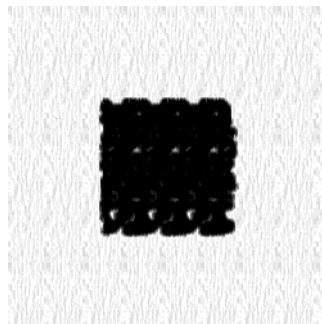
jossa

\mathbf{X}_1 on $a \times 1$ kokoinen vektori karheussignaalista,

\mathbf{X}_2 on $b \times 1$ kokoinen vektori karheussignaalista ja

\mathbf{K} on $a \times b$ kokoinen karheusmatriisi.

Karheusmatriiseista generoitiin kuvia sekä Phongin että Blinnin heijastusmallilla. Kertomalla generoiduissa karheusmatriiseissa oli selvä säännöllisyys, joka näkyy myös kuvassa 32. Säännöllisyys näkyy sekä tarkasteltavassa paperipohjassa että generoitavassa painojäljessä. Säännöllisyys kerrotussa karheusmatriisissa on ymmärrettävää, jos tarkastellaan kaavaa (114). Vektorin \mathbf{X}_1 alkio n vaikuttaa koko karheusmatriisiin \mathbf{K} n :nteen riviin ja vastaavasti vektorin \mathbf{X}_2 alkio m vaikuttaa koko karheusmatriisiin \mathbf{K} m :nteen sarakkeeseen. Vierekkäin asetetut karheussignaalin vektorit muodostavat varsin epäsäännöllisen matriisin, joka tähän tarkoitukseen on sopivampi.



Vektorit vierekkäin



Kerrottu karheusmatriisi

Kuva 32 Karheusmatriisin vaikutus generoitavaan kuvaan

Kuvassa 32 vasemmanpuoleisessa paperipinnassa on käytetty paperin karheussignaalia, joka on pilkottu pienemmiksi osiksi ja ladottu vierekkäin matriisiin. Oikeanpuoleinen paperipinta on muodostettu samasta karheus signaalista kuin edellinen mutta karheusmatriisi on tehty kaavan (114) mukaisesti. Tässä kuvassa näkyy selvä säännöllisyys. Ideaalinen tapa muodostaa karheusmatriisi olisi matriisin muodostaminen kaksidimensionaalisesta karheusdatasta.

6.3 Painovärimallin parametrit

Painovärimallia tutkittaessa on otettu huomioon absorptio-, depositio- ja halkeamisparametrit sekä speulaarin ja diffuusin heijastuksen kertoimet. Näiden kertoimien vaikutuksen näkyvyys riippuu tarkasteltavan paperin karheudesta. Jos tarkasteltava paperi on sileä, niin esimerkiksi depositiokertoimella ei ole suurtakaan merkitystä.

Absorptiokertoimella määrätään painovärin valon absorptiokyky. Tämän parametrin vaikutus näkyy, kun tutkitaan ohuita painovärikerroksia. Tällöin paperin heijastusominaisuudet näkyvät painovärikerroksen läpi. Tarkastellut absorptiokertoimet on taulukossa 8. Absorptiokertoimen vaihtelulla ei saatu huomattavia muutoksia aikaan generoitavissa kuvissa. Pienet muutokset generoiduissa kuvissa näkyivät lähinnä painovärikerroksen reuna-alueilla.

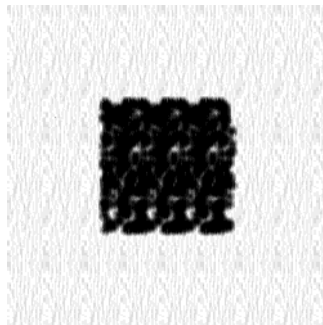
Taulukko 8 Valon absorptiokertoimet

0,20
0,90
1,16

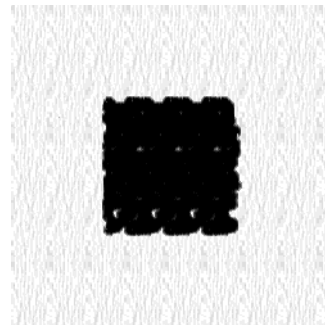
Depositiokerroin määrää painovärin asettumista karheuskorkeuden perusteella. Mitä pienempi depositiokerroin on, sitä huonommin painoväri asettuu syviin karheuskuoppiin. Tämä näkyy esimerkiksi kuvassa 33. Taulukossa 9 on tarkastellut depositiokertoimet.

Taulukko 9 Depositiokertoimet

0,20
0,50
0,80



Depositiokerroin 0,2



Depositiokerroin 0,8

Kuva 33 Depositiokerroinien vaikutus kompaktin painoväripinnan yhtenäisyyteen

Painovärin halkeamiskerointa vaihtelemalla ei saatu aikaan havaittavaa muutosta tuloskuvissa. Tarkastellut kertoimet on taulukossa 10.

Taulukko 10 Halkeamiskertoimen arvot

0,20
0,40
0,50
0,60
0,80

Painovärimallin spekulairin ja diffuusin heijastuksen kertoimia tutkittiin samalla tapaa kuin paperimallinkin kertoimia. Painoväriin heijastuskertoimien vaihtelulla ei saatu havaittavia muutoksia aikaan generoitavissa kuvissa. Tämä johtui diffuusista tarkastelugeometriasta ja painoväriin heijastusspektrien pienestä vasteesta. Kuten heijastusparametrien vaikutuksen esille saaminen niin myös muidenkin painovärimallin parametrin vaikutuksen esille saaminen on hankalaa. Tämä saattaa johtua oletusarvoista. Heijastusparametreilla on kuitenkin selvä merkitys painoväripinnan visualisoituun ulkonäköön.

Tutkittavien materiaalien spektrit pysyivät koko ajan vakiona. Spektrien vaikutus painojäljen väriin on jätetty pois, koska spektrien kokonaisvaltainen tarkastelu vaatisi runsaasti kuvan generointeja. On tärkeämpää ensin pystyä generoimaan mahdollisimman todenmukainen kuva paperista, jonka jälkeen voidaan tutkia spektrien ominisuuksien vaikutusta painojälkeen. Toisaalta on turha generoida spektrejä, koska niitä on runsaasti esillä kirjallisuudessa ja koska paperin tai painoväriin spektri pystytään mittaamaan suhteellisen helposti.

Parametrien vaikutusta visuaaliseen ulkonäköön on tutkittu näytöllä, jota ei ole kalibroitu. Kalibroinnilla olisi voinut saada näkymään enemmän painojäljen visuaalisia ominaisuuksia etenkin värialueiden tummassa ja vaaleassa päässä. Toisaalta on haettu selviä subjektiivisia visuaalisia eroja parametrien vaikutuksille, jolloin kalibroinnin merkitys ei ole suuri. Tämän vuoksi tietokoneen näyttöä ei kalibroitu.

7 TUKIMUSTULOSTEN TARKASTELU

Kappaleessa 6 käytiin läpi visualisoinnin lopputulokseen vaikuttavia visualisointimallien parametreja. Haluttu paperipohja ja yksivärinen painojälki pystytään tämän työn visualisointimalleilla generoimaan, kun valitaan sopivat parametrit malleille. Mallien parametreilla pystytään kontrolloimaan paperin karheutta, painoväriin leviämistä sekä paperin ja painoväriin väriin ja valon heijastukseen vaikuttavia ominaisuuksia.

7.1 Tutkimuksessa käytettyjen mallien todennukaisuus ja käytökelpoisuus

Tutkimuksessa käytetty paperimalli perustuu paperin karheusrakenteeseen. Koska karheus on toteutettu matriisina, joudutaan kuvaa generoitaessa matriisia käyttämään tekstuurina, joka maalataan useampaan kertaan tarkasteltavalle pinnalle. Tästä johtuen syntyy matriisin vastakkaisten reunojen välille rajapintoja, jotka näkyvät kuvassa intensiteettihiypyksinä. Toisaalta, jos karheusmatriisi on pieni tai muuten säännöllinen, syntyy kuvaan jaksollisuutta, jonka huomaa helposti. Todellisuudessa paperissa ei esiinny jaksollista säännöllisyyttä. Karheusmatriisia suurentamalla voidaan aina poistaa säännöllisyydet, jotka johtuvat karheusmatriisin koosta. Tässä tarvitaan kuitenkin enemmän muistikapasiteettia. Toisaalta haitallista jaksollisuutta saattaa syntyä vaikka matriisia kasvatetaan.

Paperipinnan karheuden generointiin on muitakin tapoja kuin karheusmatriisi. Karheus voidaan generoida esimerkiksi fraktaalialgebran avulla. Fraktaalialgebran avulla pinnat muodostetaan tietyn kaavan mukaan, joka sisältää pinnan struktuuriinformaation. Fraktaaligeometrian kaavat ovat yleensä seuraavan muotoisia:

$$\begin{aligned} f_0(\mathbf{x}) &= a(\mathbf{x}) \\ f_n(\mathbf{x}) &= g(f_{n-1}(\mathbf{x})) \end{aligned} \tag{115}$$

jossa

$a(\mathbf{x})$ on alustusfunktio ja

$g(\)$ on iteroinnin kaava, joka sisältää pinnan karheusstruktuuriinformaation.

Tällainen iterointikaava saa tuloksensa, kun $f_n(\mathbf{x})$ on ylittänyt tietyn rajan tai kun tietty määrä iterointeja on suoritettu eli $n > n_z$. Fraktaalialgebraa hyväksi käyttäen pystytään karheusinformaatio tallentamaan pieneen tilaan. Karheusinformaatio on pakattu lyhyeen kaavaan. Haittana fraktaaligeometrian käytössä on kuitenkin sopivan kaavan, joka simuloi hyvin paperipintaa, löytäminen. Toisaalta pintaa generoitaessa tarvitaan runsaasti iterointeja. Jos esimerkiksi iteroinnin raja on sadassa iteroinnissa, tarvitaan karkeasti sata kertaa enemmän laskutoimituksia pinnan korkeuden määrittämiseen kuin karheusmatriisin avulla tarvitaan.

Paperin pinta on satunnainen karhea pinta. Tällainen pinta voidaan generoida

stokastisesti. Paperipinnan stokastisessa generoinnissa tarvitaan parametrejä luonnehtimaan stokastista prosessia. Näiden parametrien kohdalleen asettelu vaatii tarkkaa paperin karheusrakenteen tuntemusta. Eräs rajoittava tekijä paperin karheuden muodostamisessa stokastisesti on satunnaislukugeneraattori. Jotta saadaan aikaiseksi satunnainen pinta, tarvitaan tehokas ja hyvä satunnaislukugeneraattori. Jos esimerkiksi käytetään kellon tuhannesosia satunnaislukuina, syntyy tiettyjä jaksoja generoituun painojälkeä esittävään pintaan, koska pinnan jokaiselle pikselille suoritetaan samat operaatiot.

Valon heijastumiseen genroitavasta pinnasta vaikuttaa mikropintojen jakauma. Tässä työssä on käytetty mikropintojen jakaumina yksinkertaisia funktioita. Mikropintojen jakaumina on käytetty Blinnin mallin distribuutiofunktiota kuten kaavassa (75). Phongin mallissa käytetään vastaavaa kahden vektorin välisen kulman kosinin potenssia ^[36]. Jotta saataisiin lähempänä todellisuutta oleva mikropintojen jakauma, voidaan käyttää monimutkaisempaa jakaumaa tai käyttää useamman jakauman lineaarista kombinaatiota /*Kuva 34*:/

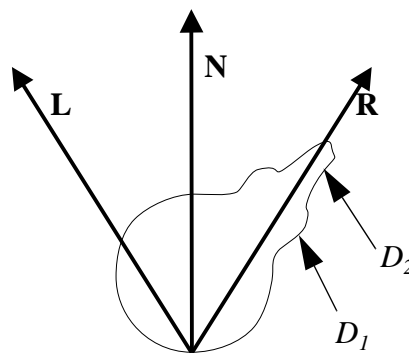
$$D(h) = aD_1(h) + (1-a)D_2(h) \quad (116)$$

jossa

h on mikropinnan normaalin \mathbf{H} ja pinnan normaalin \mathbf{N} välinen kulma
ja

a on kerroin, joka on välillä $[0,1]$.

Toisaalta käytetyt mikropintojen jakaumat ovat symmetrisiä. Paperilla on kuitenkin tietty orientaatio, joka aiheuttaa erilaisia ominaisuuksia pituus- ja leveyssuuntaan. Tämän orientaation liittämällä asymmetrisinä mikropintojen jakaumina saataisiin paperin orientaatiosta aiheutuvat heijastusominaisuudet esiin.



Kuva 34 Kahden erilaisen mikropintajakauman D_1 ja D_2 kombinaatio

Tämän tutkimuksen paperimallissa (kappale 5.4) on paperille liitetty tietty värisävy spektraalisten heijastusominaisuuksien muodossa. Tämä värisävy on sama koko tarkasteltavan pinnan suhteen. Värisävyn vaihtelu syntyy spekulaarin, diffuusin ja

ambientin heijastuksen summana. Toisaalta, kun esimerkiksi tarkastellaan kopiopaperia, joka on valmistettu kierrätetyistä materiaaleista, paperista löytyy erilaisia kuituja, joiden väri vaihtelee esimerkiksi kuidussa olevan painovärin takia. Tämän työn paperimallissa ei ole otettu huomioon tällaista kuitujen spatiaalista väri vaihtelua.

Tämän työn painovärin (kappale 5.6) leviämismallin puutteena on leviämisen rajoittuminen painettavan alan sisälle ja painamisen kohinan puute. Painovärin leviämisessä ei olla käsitelty leviämistä, jolla olisi jokin tietty suunta. Tätä voidaan kuitenkin tarkastella konvoluutiomatriisilla, joka levittää painoväriä joihinkin suuntiin enemmän kuin toisiin. Kaikki painamisen mekaniikasta johtuvat virheet kuten vierintä puuttuvat tämän työn painovärin mallista. Toisaalta voidaan olettaa, että painamisen mekaniikasta ei aiheudu virheitä. Oletukseen liittyy kysymys siitä, missä määrin kannattaisi mallittaa ja simuloida tiettyä painamisprosessia, vai keskittyäkö pelkästään siihen mitä tapahtuu kun painoväri on jo paperilla. Vastaavasti, miten lasertulostimien painovärit asettuvat paperille. Tässä työssä käytetty painovärin leviämismalli koskee ainoastaan perinteistä painamista.

Painojäljen reunojen epätasaisuus ei yksin johdu painovärin leviämisestä vaan myös painamisessa syntyvästä kohinasta. Kohina aiheuttaa ideaalisen reunan lähelle pisteitä, jotka eivät kuulu reunan rajoittamaan elementtiin. Tämä aiheuttaa myös tyhjiä kohtia, joilla pitäisi olla painoväriä. Kohina ei rajoitu ainoastaan reunojen lähelle, mutta se on reunan lähellä merkityksellisintä. Yksittäiset pienet pisteet eivät välttämättä erotu, koska silmä alipäästösuodattaa nämä pois.

Tässä työssä valonlähde (kappale 5.7) on mallitettu pistemäisenä valonlähteenä, joka sijaitsee äärettömän kaukana. Tällaista valonlähdetä on fysikaalisessa mielessä mahdollon toteuttaa. Todellisella valonlähteellä on jokin dimensio ja rajallinen etäisyys, jotka karakteroivat valonlähdetä. Valonlähteen mallissa on tehty runsaasti yksinkertaistuksia, koska on oletettu, että generoitavaa pintaa tarkastellaan laboratorio-olosuhteissa.

Valonlähteen etäisyys vaikuttaa kuitenkin tutkitulle pinnalle saapuneeseen energiaan. Saapunut energia vähenee etäisyyden neliön suhteen

$$E = \frac{E_0}{r^2} \quad (117)$$

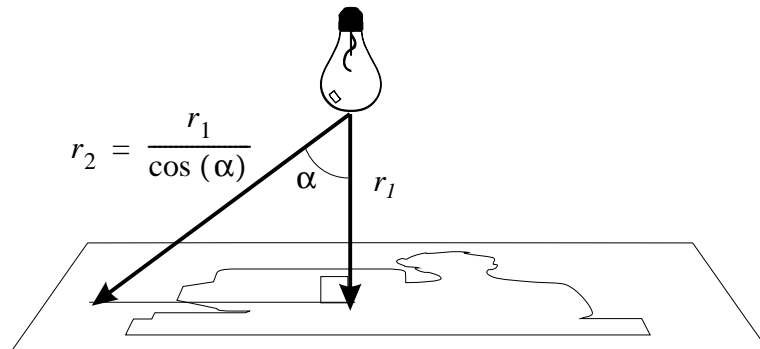
jossa

r on valolähteen etäisyys tarkasteltavasta pisteestä ja

E_0 on valolähteen emittoima energia.

Kun tarkastellaan suurta pinta-alaa valonlähteen etäisyyteen nähden, on pinnalle saapuneella energialla huomattavaa vaihtelua pinnan yli /Kuva 35/. Kun kuvassa 35 kulma α lähestyy 90 astetta, lähestyy $\cos(\alpha)$ nollaa. Etäisyyksien r_2 ja r_1 suhde kasvaa, mitä suurempi kulma niiden välissä on. Jos tutkittavat intensiteetit ovat vielä suhteellisen matalia, näkyy tämä ero tummempina reunoina. Etäisyys voidaan kuitenkin

huoletta jättää huomiotta, koska tarkasteltavat pinnat ovat pinta-alaltaan pieniä ja valonlähteet suhteellisen kaukana.



Kuva 35 Valonlähteen etäisyyden merkitys valon tulokulmaan

Jos lähdetään simuloimaan todellisia katseluolosuhteita, tarvitaan monimutkaisempi valaistusmalli. Valaistusmallin tulee sisältää useampia valonlähteitä ja todellisia etäisyyksiä. Lisäksi tulisi huomioida erilaisista pinnoista heijastuneet valosäteet tarkasteltavalle pinnalle. Tällaiset mallit ovat kuitenkin epäoleellisia tämän työn kannalta.

7.2 Mallien tehokkuus ja ohjelman konekapasiteetin tarve

PC tason tietokone, jonka keskusprosessori oli Intelin 50 MHz 486DX prosessori ja 8 MB keskusmuistia, kulutti 40 min 33 s aikaa 500 x 500 pikselin kuvan generointiin. Generoitu kuva kuvasi 10 mm x 10 mm paperin pintaa, jolle oli generoitu 2 mm x 2 mm kompakti painoväripinta. Heijastusmallina käytettiin Phongin heijastusmallia. Vastaavan kuvan generointiin Blinnin mallilla kului 52 min 18 s.

Edellä olevat pitkät laskenta-ajat eivät tue interaktiivisuutta. Ajatellen mahdollista sovellusta, vedostustyöasemaa, jossa käyttäjä simuloi jotain painopintaa, suurin osa ajasta kuluisi odotteluun. Tällaisessa sovelluksessa ensimmäiset generoinnit harvoin tuottavat haluttua tulosta. Jotta päästäisiin todelliseen interaktiivisuuteen, täytyy kuvan genrointi aikojen olla alle minuutin pituisia.

Taulukko 11 Pikselin ja koko kuvan generointiin kuluva aika

ms/pix	500x500 pix	800x600 pix
0.1	6 h 56 min 40 s	13 h 20 min 0 s
0.01	41 min 40 s	1 h 20 min 0 s
0.001	4 min 10 s	8 min 0 s
0.0001	25 s	48 s

Edellisessä kuvassa, joka oli genroitu Phongin mallilla, kului aikaa yhden pikselin laskuun noin 10 ms. Tämä pikselin laskenta-aika pitäisi pystyä pudottamaan sadasosaan 0,1 ms, jolloin normaali 800 x 600 pikselin monitorin kokoisen kuvan gene-

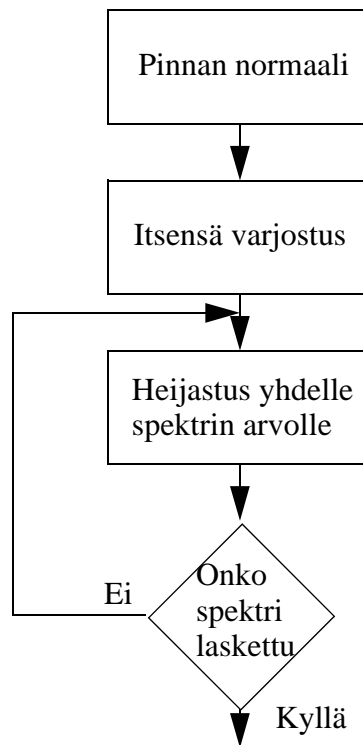
rontiin kuluu noin 48 s /*Taulukko 11*/. Tietokoneessa, jonka prosessorin kellotaajuus on 50 MHz, yhden prosessorikäskyn suorittamiseen kuluu minimissä aikaa 0,02 ms. Koska Intelin 486DX prosessori on CISC¹ prosessori, todelliseen prosessorikäskyyn kuluu enemmän aikaa kuin 0,02 ms. Jos pikselin laskemiseen annetaan 0,1 ms, voidaan tänä aikana 50 MHz prosessorilla suorittaa 5 käskyä maksimissaan. Tällaiseen interaktiivisuuteen pääseminen nykyisillä PC-tason tietokoneilla on mahdottomuus. Jo muistiin kirjoitus- ja muistista lukuoperaatiot kuluttavat aikaa yli yhden kellojakson.

Vedostustyöaseman toteutusta tarkastellessa yksi mahdollisuus on generoida valmiiksi paperipohjien kuvat. 800x600 pikselin kokoinen pakkaamaton RGB kuva, jossa kukin osaväri on tallennettu kahdeksalla bitillä, käyttää 1,4 MB muistikapasiteettia. Jos ajatellaan erilaisten valmiiden paperilaatujen ja väri vaihtoehtojen tallentamista kuvakirjastoksi vedostustyöasemalle, kahdensadan kuvan tallentaminen vie 280 MB muistikapasiteettia. Tällainen määrä kuvia vaatii normaalia suuremman massamuistin. Tosin generoidut paperipohjien kuvat ovat hyvin saman sävyisiä, jolloin ne pakatessa tiivistyvät helposti kolmanteen osaansa. Tällöin massamuistikapasiteetin tarve ei ole niin suuri. Tällaisissa tapauksissa yhtenä rajoittavana tekijänä pakkaus- ja purkuohjelmien tehokkuus.

Viime kädessä kuvakirjaston tehokkuus riippuu painojäljen generoinnin tehokkuudesta. Kun painojälkeä generoidaan, joudutaan käyttämään samantapaista algoritmia heijastusmallille kuin paperin generointiin. Jos painojälki peittää koko tarkasteltavan alueen, tehokkuus ei nouse valmiin paperipohjan myötä. Tehokkuus päin vastoin laskee, koska algoritmi joutuu hakemaan paperipohjan pikselin väriarvon muistista ja konvertoimaan sen näkyvän valon spektriä.

Mahdollisen sovelluksen tehokkuuden lisäämiseksi yksinkertaisin keino on algoritmin tehostaminen. Suurin osa ohjelman ajasta kuluu sisimmän silmukan suorittamisessa eli tässä tapauksessa yhden pikselin laskemisessa /*Kuva 36*/. Erityisesti, kun tarkastellaan pikselin arvon määrittämistä, väriarvo lasketaan silmukalla, joka käy koko näkyvän spektrin läpi yhden nanometrin välein. Tämän silmukan laskutoimitus suoritettiin tässä työssä käytetyssä ohjelmassa 401 kertaa. Kun näiden kertojen lukumäärää pystytään vähentämään, tehostuu ohjelma huomattavasti.

-
1. CISC (Complex Instruction Set Circuit) prosessoriarkkitehtuuri käyttää monimutkaista ja suurta käskykantaan verrattuna RISC (Reduced Instruction Set Circuit) arkkitehtuuriin, jossa suoritetaan yhden kellojakson aikana yksi käsky. CISC-prosessori käyttää vaihtelevan määrän kellojaksoja käskyn suorittamiseen



Kuva 36 Yksinkertaistettu algoritmi pikselin heijastusarvon määrittämiseksi

Spektri voidaan esimerkiksi muuttaa RGB-silmukaksi, jolloin silmukoiden laskukerrat supistuvat kolmeksi. Tällöin laskenta-aika supistuu alle prosenttiin alkuperäisestä. Kun käytetään laitteistoriippuvia koordinaatteja, vältetään spektrin muuttamiselta ensin tristimuluskoordinaateiksi, jotka muutetaan RGB-koordinaateiksi. Toisaalta, kun lasketaan laitteistoriippuvilla koordinaateilla, laskenta on koneriippuvaista. Kun siirrytään laitteistoriippumattomiin koordinaatteihin esimerkiksi XYZ-tristimuluskoordinaatteihin, RGB-koordinaattien saamiseksi joudutaan laskemaan matriisioperaatio XYZ-tristimuluskoordinaateille. Kun lasketaan RGB-koordinaateilla, värivirheiden mahdollisuus on suuri.

Visualisoinnissa käytetyt algoritmit asettavat omat vaatimuksensa tietokonearkkitehtuurille. Kuvaa generoitaessa lasketaan reaalityyppisillä. Jos tietokoneessa on matemaattikkaprosessori, niin algoritmin suoritus nopeutuu huomattavasti. Esimerkiksi Intelin 486DX prosessori sisältää reaalityyppiprosessorin. Keskusmuistin tarve riippuu lähinnä siitä, missä ympäristössä ja minkälaisia ohjelmia on auki, kun kuvaa generoidaan. Jos kuvaa generoitaessa on muistia syöviä ohjelmia käynnissä, joudutaan suorittamaan runsaasti luku- ja kirjoitusoperaatioita massamuistille.

Vedostustyöasemaa suunniteltaessa tulee ottaa huomioon kuvan generoinnin erityispiirteet. Pikselien heijastusarvot voidaan laskea toisistaan riippumatta. Tämän takia voidaan kuvan generointiin soveltaa rinnakkaista arkkitehtuuria, jolloin lasketaan enemmän kuin yhden pikselin arvo kerrallaan. Tällöin vältetään hitaiden kellotaajuuksien aiheuttamalta rajoitukselta. Toisaalta erikoispiirit ovat aina kalliita ja hankalasti ylläpidettäviä. Kun käytetään yleistä avointa arkkitehtuuria ohjelman suorittamiseen, säilytetään helppo siirrettävyys ja ylläpidettävyys.

8 YHTEENVETO

Tämän diplomityön tavoitteena oli visualisoida painojälkeä tietokoneen näytöllä. Tavoitteeseen liittyy pyrkimys visualisoida painojälkeä ennen varsinaista painamisprosessia. Tavoite jakautui paperin ja painoväriin visualisointimallien luomiseen sekä mallien avulla painojäljen kuvan generointiin.

Pyrkimyksenä oli generoida riittävän hyvä ja todellisuutta vastaava kuva perinteisillä menetelmillä tuotetusta painojäljestä. Tutkimusongelmat jakautuivat sopivan valon heijastusmallin löytämiseen sekä paperin karheuden esittämiseen. Kirjallisuudessa esitetyt heijastusmallit on tarkoitettu kappaleiden geometrian eikä niinkään karheiden pintojen esittämiseen. Tästä johtuen jouduttiin käyttämään lähdekirjallisuutta soveltuvin osin. Paperin karheuden mallittamisessa nousi ongelmaksi karheuden esittäminen riittävän satunnaisesti, jotta saavutettasiin paperille ominainen visuaalinen rakenne ja karheusvaikutelma.

Tarkasteltaviksi heijastusmalleiksi valittiin Phongin ja Blinnin esittämät heijastusmallit. Phongin heijastusmalli valittiin, koska kaupalliset sovellutukset lähes yksinomaan soveltavat sitä sen yksinkertaisuuden vuoksi. Blinnin esittämä heijastusmalli valittiin, koska se pyrkii mallittamaan todellista heijastusilmiötä.

Painojälkeä tarkasteltiin mikrometritasolla. Paperi- painovärimallit luotiin tarkastelemalla materiaalien ominaisuuksia, jotka vaikuttavat visuaalisen ulkonäköön.

Paperissa keskityttiin karheuden esittämisen lisäksi heijastusominaisuuksiin. Paperin karheutta mallinnettiin matriisin avulla, joka sisälsi paperin karheusvaihteluinformaation kaksidimensionaalisenä profiilikarttana. Heijastusominaisuudet mallinnettiin spekulaarin ja diffuusin heijastuksen avulla. Spekulaarille ja diffuusille heijastuksille määritettiin emissiospektri, jonka perusteella määräytyi paperin väri ja heijastuksen intensiteetti.

Painoväriä mallinnettiin painoväriin leviämisoiminaisuuksilla ja heijastusominaisuuksilla. Painoväriä määritettiin vastaanvanlaiset heijastusominaisuudet kun paperillekin. Painoväriin leviämismalli kuvasi mekaanisen painatuksen aiheuttamaa painoväriin siirtymistä ja leviämistä paperille. Kun määritetään painoväriin määrää jossain tarkasteltavassa kohdassa, painoväriin leviämismalli käyttää hyväkseen paperin karheutta ja kontaktikorkeutta. Varsinainen painoväriin leviämisoiminaisuus riippuu konvoluutiomatriisista, joka kuvaa painoväriin leviävyyttä.

Mallien perusteella rakennettiin tietokoneohjelma, joka generoi Tiff-kuvan paperin pinnasta ja/tai painojäljestä haluttujen parametrien suhteen. Ohjelma generoi kuvat kuvapikseli kerrallaan yhdistäen paperin sekä painoväriin mallien parametrit haluttuun heijastusmalliin.

Mallien parametrien vaihtelulla saatiin aikaan kuvia, jotka vastasivat painatuksessa syntynyttä painojälkeä. Parametrejä pyrittiin tarkastelemaan alueilla, joissa selvästi näkyi parametrin vaikutus. Kuville ei tehty mitään visuaalisia testauksia vastaavien painojälkikuvien kanssa. Merkittävimpänä ominaisuutena kuvan todenmukaisuuteen vaikutti paperin karheusmatriisi. Jos karheusmatriisi oli ”hyvä”, saatiin todenmukaisia paperipintoja ja niille generoituja painojälkiä.

Tutkimuksen tavoitteet saavutettiin ja tutkimuksen ongelmat pystyttiin ratkaisemaan varsin hyvin. Uutena ongelmana syntyi generoinnin vaatima laskukapasiteetti, joka vaatii tarkempaa ja spesifioidumpaa toteutusta, jotta pystyttäisiin minimoimaan generointiin kuluva aika.

Tämän tutkimuksen perusteella on mahdollista tarkastella eri paperipintojen vaikutusta painojälkeen. Visualisoinnin hyöty tulee erityisesti esille, kun valitaan paperipohjaa tai paperipohjan ja painomenetelmän yhdistelmää tietylle painotyölle. Tällainen tarkastelu vaatii kuitenkin ensin tarkkaa paperin painoväriin parametrien määrittämistä. Tarkkaa syventymistä vaativat erityisesti paperin karheuden määrittäminen ja sen esittäminen, koska visualisoinnin lopputulos riippuu olennaisesti paperin karheudesta.

Mahdollisen tuotteen kehitykseen vaaditaan paperin karheuden määrittämisen lisäksi tarkkaa algoritmien analysointia ja optimointia. Muita mahdollisia jatkotutkimuskohteita ovat moniväristen rasteripintojen visualisointi sekä erilaisten painomenetelmien tehokas mallittaminen.

LÄHTEET

- [1] Barten, P. G. J., *Image quality of CRT Displays*. Seminar M-2, Society for Information Displays, SID 1990
- [2] Bishop, G., Weimer, D. M., *Fast Phong Shading*. SIGGRAPH 1986 Proceedings, Computer Graphics vol. 20, no. 4 (1986), 192-198
- [3] Blinn, J. F. *Models of Light Reflection for Computer Synthesized Pictures*. SIGGRAPH 1977 Proceedings, Computer Graphics vol. 11, no. 2 (1977), 192-198
- [4] Blinn, J. F., *Simulation of Wrinkled Surfaces*. Proc. SIGGRAPH 1978, Atlanta, Ga., pp. 286-292
- [5] Blinn, J. F., *Light Reflection Functions for Simulation of Clouds and Dusty Surfaces*. SIGGRAPH 1982 Proceedings, Computer Graphics, vol 16, no 3, pp. 21-29
- [6] Blinn, J.F., Newell, M.E. *Texture and Reflection in Computer Generated Images*. Communications of the ACM vol. 19, no. 10 (Oct. 1976), 542-547
- [7] Cook, R. L., Torrance, K. E., *A Reflectance Model for Computer Graphics*. SIGGRAPH 1981 Proceedings, Computer Graphics, vol 15, no 3, pp. 307-316
- [8] Foley, J. D., van Dam, A., Feiner, S. K., Hughes, J. F., *Computer Graphics Principles and Practice*. Second Edition, Addison-Wesley, 1990
- [9] Forsyth, D., Zisserman, A. *Reflections on Shading*. IEEE Transactions on pattern analysis and machine intelligence vol. 13, no. 7 (July 1991), 671-679
- [10] Gonzales, R. C., Wintz, P., *Digital Image Processing*. Second Edition. Addison-Wesley Publishing Company, 1987.
- [11] Greenberg, D. B. *Light Reflection Models for Computer Graphics*. Science vol. 244, no. 14 (April 1989), 166-173
- [12] Hall, R., *Illumination and Color in Computer Generated Imagery*. Springer-Verlag, New York, 1989
- [13] Healey, G., Binford, T. O. *Local Shape from Specularity*. Computer Vision, Graphics and Processing no. 42 (1988), 62-86
- [14] Hearn, D., Baker, M.P. *Computer Graphics*. Prentice-Hall, USA, 1986, pp. 276-282
- [15] Hopgood, F. R. A., Hubbold, R. J., Duce, D. A., *Advances in Computer Graphics II*. Springer-Verlag, 1986
- [16] Horn, B. K. P. *Determining Lightness from an Image*. Computer Graphics and

Image Processing no. 3 (1974) , 277-299

- [17] Horn, B. K. P. *Understanding Image Intensities*. Artificial Intelligence no. 8 (1977), 201-231
- [18] Houle, C., Fiume, E., *Light-Source Modeling Using Pyramidal Light Maps*. CVGIP: Graphical Models and Image Processing, vol 55, no. 5, September 1993, pp. 346-358
- [19] Jackson, L. B., *Digital Filters and Signal Processing*. Second Edition. Kluwer Academic Publishers, 1989
- [20] Judd, D. B., Wyszecki, G., *Color in Business, Science and Industry*. John Wiley & Sons, Inc. 1975
- [21] Kernighan, B. W., Ritchie, D. M., *The C Programming Language*. 2. Edition, Prentice Hall, 1988
- [22] Kivelä, S. K., *Matriisilasku ja lineaarialgebra*. Otakustantamo 1980.
- [23] Korpela, J., Larmela, T., *C-ohjelmointikieli*. OtaData, 1988
- [24] Lee, C-H., Rosenfeld, A. *Improved Methods of Estimating Shape from Shading Using the Light Source Coordinate System*. Artificial Intelligence no. 26 (1985), 125-143
- [25] Löppönen, J-M., *CRT-pohjainen visualisointi*. Teknillinen korkeakoulu, Graafisen tekniikan laboratorio. Diplomityö 1993.
- [26] Marr, D. *Vision*. Freeman, New York, 1982, 239-250
- [27] Meyer-Ardent, J. R., *Introduction to Classical and Modern Optics*. Prentice-Hall, 1972
- [28] Nayar, S. K., *Surface Reflection: Physical and Geometrical Perspectives*. IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 13, no. 7 (July 1991), 611-634
- [29] Ogilvy, J. A., *Theory of Wave Scattering from Random Rough Surfaces*. Adam Hilger, 1991
- [30] Oittinen, P., Saarelma, H. *Graafinen materiaalteknikka*. Otakustantamo, 1987.
- [31] Oittinen, P., Saarelma, H. *Kuvatekninenlaatu*. Otatieto, Espoo, 1992
- [32] Oittinen, P., Saarelma, H. *Basic Principles Related to the Influence of the Paper Surface in Printing*. IS&T's 46th Annual Conference (1993), 378-381
- [33] Oittinen, P., Saarelma, H. *Influence of Optical Surface Properties of Paper on*

- Information Capacity*. Paper and Timber vol 75, no. 1-2 (1993), 66-71
- [34] Oppenheim, A. V., Schafer, R. W., *Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, N.J., 1975, pp. 26-34
 - [35] Peachey, D. R., *Solid Texturing of Complex Surfaces*. SIGGRAPH 1985 Proceedings, Computer Graphics, vol 19, no 3, pp. 279-286
 - [36] Phong B. T. *Illumination for Computer Generated Pictures*. Communications of the ACM vol. 18, no. 6 (June 1975), 311-317
 - [37] Poulin, P., Fournier, A., *A Model for Anisotropic Reflection*. SIGGRAPH 1990 Proceedings, Computer Graphics, vol 24, no 4 (August 1990), pp. 273-282
 - [38] Press, W. H., Flannery, B. P., Teukolsky, S. A., Vetterling, W. T., *Numerical Recipes in C, The Art of Scientific Computing*. Cambridge University Press, 1988
 - [39] Rogers, D. F., *Procedural Elements for Computer Graphics*. McGraw-Hill, New York, 1985
 - [40] Rushmeier, H. E., Torrance, K. E., *Extending the Radiosity Method to Include Specularly Reflecting and Translucent Materials*. ACM Transaction on graphics, vol. 9, no. 1 (January 1990), pp. 1-27
 - [41] Ryti, N., *Paperitekniikan perusteet*. Otakustantamo, 1988.
 - [42] Saarelma H., Oittinen P., *Fundamentals of Printing Technology*. Helsinki University of Technology, Laboratory of Graphics Arts Technology, 1993
 - [43] Schalkoff, R. J., *Digital Image Processing and Computer Vision*. John Wiley & Sons inc., 1989
 - [44] Smith, G.B. *The Relationship Between Image Irradiance and Surface Orientation*. IEEE Computer Society Conference on computer Vision and Pattern Recognition. Proceeding 1983, IEEE Computer Society Press: Silver Spring, 14-19
 - [45] Szeliski, R. *Fast Shape from Shading*. Image Understanding vol. 53, no. 2 (March 1991), 129-153
 - [46] Tannas, Jr., Lawrence, E., *Flat-panel displays and CRT's*. Van Nostrand Reinhold Company, New York 1985
 - [47] Torrance, K. E., Sparrow, E. M. *Polarization, Directional Distribution, and Off-Specular Peak Phenomena in Light Reflected from Roughened Surfaces*. Journal of the Optical Society of America vol 56, no. 7 (July 1967), 916-925
 - [48] Torrance, K. E., Sparrow, E. M. *Theory for Off-specular Reflection From Roughened Surfaces*. Journal of the Optical Society of America vol 57, no. 9

(September 1967), 1105-1114

- [49] Trowbridge, T. S., Reitz, K. P., *Average Irregularity Representation of a Roughened Surface for Ray Reflection*. Journal of the Optical Society of America vol 65, no. 5 (May 1975), 531-536
- [50] Ward, G., *Measuring and Modeling Anisotropic Reflection*. SIGGRAPH 1992 Proceedings, Computer Graphics, vol 26, no 2 (July 1992), pp.265-272
- [51] Warn, D. R. *Lighting Controls for Synthetic Images*. Computer Graphics vol. 17, no. 3 (July 1983), 13-21
- [52] Watt, A., *3D Computer Graphics*. Addison-Wesley, 1989
- [53] Whitted, T. *An Improved Illumination Model for Shaded Display*. Communications of the ACM vol. 23, no. 6 (June 1980), 343-349
- [54] Wolcott, J., *SuperMatch Display Calibrator User's Manual*. SuperMac Technology, 1990
- [55] Wyszecki, G., Stiles, W. S., *Color Science, Concepts and Methods, Quantitative Data and Formulae*. 2nd Edition, John Wiley & Sons, 1982
- [56] Yasuda, T., Yokoi, S., Toriwaki, J-I., Inagaki, K. *A Shading Model for Cloth Objects*. IEEE Computer Graphics & Applications (Nov. 1992), 15-24
- [57] Zheng, Q., Chellappa, R. *Estimation of Illuminant Direction, Albedo, and Shape from Shading*. IEEE Transactions on pattern analysis and machine intelligence vol. 13, no. 7 (July 1991), 680-70

Liite 1 Painojäljen generoinnissa käytettyjä, spektrin vasteita ja karheussignaaleja

Paperin diffuusi spektrin on mitattu tavallisesta kopiopaerista Minolta CM-1000 spektrofotometrillä, joka käyttää mittausvalolähteenään D65 standardin mukaista valolähdettä. D65 standardivalolähteen spektri on otettu Wyszecki & Stilesin kirjasta *“Color Science, Concepts and Methods, Quatitative Data and Formulae”*, 1982. Valolähteen spektri on normitettu niin, että aallonpituus 560 nanometriä saa arvon yksi. Muut spektrit ovat tasasaisia.

Aallon pituus	D65 valolähde	Paperin diffuusi spektri	Paperin spekulaari spektri	Painoväriin diffuusi spektri	Painoväriin spekulaari spektri
380	0,4993	0	1	0,0001	0,5
390	0,546	0	1	0,0001	0,5
400	0,827	0,3371	1	0,0001	0,5
410	0,9142	0,6264	1	0,0001	0,5
420	0,9337	0,8685	1	0,0001	0,5
430	0,8663	0,9563	1	0,0001	0,5
440	1,0481	0,9639	1	0,0001	0,5
450	1,1696	0,9374	1	0,0001	0,5
460	1,1777	0,9132	1	0,0001	0,5
470	1,1482	0,9011	1	0,0001	0,5
480	1,1589	0,8903	1	0,0001	0,5
490	1,0878	0,882	1	0,0001	0,5
500	1,0933	0,8773	1	0,0001	0,5
510	1,0778	0,8731	1	0,0001	0,5
520	1,0478	0,8693	1	0,0001	0,5
530	1,0768	0,8664	1	0,0001	0,5
540	1,044	0,8641	1	0,0001	0,5
550	1,0404	0,8626	1	0,0001	0,5
560	1	0,8622	1	0,0001	0,5
570	0,9634	0,8625	1	0,0001	0,5
580	0,9579	0,8631	1	0,0001	0,5
590	0,8869	0,8642	1	0,0001	0,5

Aallon pituus	D65 valolähde	Paperin diffuusi spektri	Paperin spekulaari spektri	Painoväriin diffuusi spektri	Painoväriin spekulaari spektri
600	0,9002	0,8682	1	0,0001	0,5
610	0,8961	0,8762	1	0,0001	0,5
620	0,8771	0,8852	1	0,0001	0,5
630	0,833	0,8942	1	0,0001	0,5
640	0,8372	0,9018	1	0,0001	0,5
650	0,8005	0,9082	1	0,0001	0,5
660	0,8024	0,9126	1	0,0001	0,5
670	0,823	0,9158	1	0,0001	0,5
680	0,7831	0,9168	1	0,0001	0,5
690	0,6974	0,9208	1	0,0001	0,5
700	0,7163	0,9215	1	0,0001	0,5
710	0,7437	0	1	0,0001	0,5
720	0,6162	0	1	0,0001	0,5
730	0,6991	0	1	0,0001	0,5
740	0,7511	0	1	0,0001	0,5
750	0,6361	0	1	0,0001	0,5
760	0,4643	0	1	0,0001	0,5
770	0,6683	0	1	0,0001	0,5
780	0,634	0	1	0,0001	0,5

Karheusmatriisin luomiseen käytettyjä karheussignaaleja. Oletus arvoisesti on käytetty karheusmatriisia, joka on tehty signaalista 3. Signaaleista on ainoastaan osa tässä esimerkinvuoksi, koska todelliset karheussignaalit, joita on käytetty, ovat 1024 x 1 suuruisia vektoreita.

Paperi 1	Paperi 2	Paperi 3
2	11	-24
1	9	-36
0	5	-40
-4	4	-38

Paperi 1	Paperi 2	Paperi 3
-4	5	-36
-3	0	-30
-4	-3	-35
-2	-8	-31
-6	-9	-25
-3	-13	-12
-1	-16	-10
0	-17	-11
-2	-16	-17
0	-17	-20
0	-17	-15
-3	-15	-2
-1	-13	19
-3	-14	28
-3	-10	33
-5	-7	22
-2	-5	7
0	-3	3
0	2	-15
1	2	-39
-1	2	-49
-4	1	-35
0	-3	-33
-3	-1	-18
-3	-4	-16
-6	-1	-33
-8	0	-41
-7	6	-45
-12	8	-34
-11	6	-7

Paperi 1	Paperi 2	Paperi 3
-13	2	-1
-12	-1	-5
-12	-4	1
-9	-9	25
-10	-8	53
-8	-9	52
-7	-15	51
-8	-13	37
7	-14	23
31	-13	17
26	-12	7
0	-12	0
-21	-8	4
-24	-7	7
-22	-3	3
-16	-1	3

Liite 2 Käytettyjen valmiiden ohjelmapakettien tekijänoikeudet

Liite 2.1 “Illumination and Color in Computer Generated Imagery”-kirjan tekijän Roy Hallin jakama ohjelmakoodi

“Illumination and Color in Computer Generated Imagery”

Source for code given in the Appendices

(c)Roy Hall - 1988,1989

This code was developed to test and demonstrate concepts discussed in “Illumination and Color for Computer Generated Imagery” (1989 from Springer-Verlag). While the code was used in the generation of the diagrams and images presented in the text, it may still contain bugs. The user is cautioned to independently verify that the code is suitable to his/her needs. This code is distributed on an “as is” basis with all faults and without any implied or expressed warranties or support from either the author, Roy Hall, or from Springer-Verlag Publishers.

I am making this code available to the user community. You are free to use it as you wish. In return I request two things. First, that this README file always stays with any copy of the source, more specifically, that proper credit is given to the original author. Second, that any corrections, improvements, etc. are passed back to the author for inclusion in subsequent distribution and editions of the book.

Please note that the code was written for clarity of presentation, not for performance. You will find that optimizations can be easily made. Please do not report these as I will not incorporate them into future releases. Note also that this code reflects current work I am involved in and may differ slightly from that presented in the text.

There have been some additions to the distribution (16 July 90):

Corrections: list of corrections for the book and the code

Slectral/: a directory containing a few spectral curves to help get you started

Roy Hall
Program of Computer Graphics
120 Rand Hall
Cornell University
Ithaca, NY 14853
roy@wisdom.graphics.cornell.edu

Liite 2.2 Tiff-library

/*

* Copyright (c) 1991, 1992 Sam Leffler

* Copyright (c) 1991, 1992 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

*/

Liite 2.3 Getopt.c

/*

* Copyright (c) 1987 Regents of the University of California.

* All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

This product includes software developed by the University of California, Berkeley

and its contributors.

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Liite 3 Proof -ohjelman käyttöohje

Nimi

proof - Paperin pinnan ja painojäljen visualisointi ohjelma

Lyhyt käyttöohje

```
proof [-fHIPB] [-t kuva] [-ipl tiedosto]
      [-xyd koko] [-V suunta]
```

Kuvaus

Ohjelma on tarkoitettu paperipohjan ja painojäljen generointiin. Ohjelma generoi paperin, painovärin ja valolähteen mallien perusteella kuvia paperista ja painojäljestä. Ohjelmalla pystytään visualisoimaan paperin karheusrakenteen aiheuttamaa pinnan heijastusominaisuuksien vaihtelua ja painojäljen asettumista painettavalle paperille. Ohjelma pystyy simuloimaan ainoastaan yksiväristä painojälkeä.

Ohjelma laskee pikseli kerrallaan heijastusominaisuudet tarkasteltaville pinnoille. Heijastus malleina käytetään, joko Phongin tai Blinnin heijastusmallia. Phongin malli on yksikertaisempi ja karkeampi heijastusmalli. Phongin malli on myös nopeampi laskea kuin Blinnin malli, mutta Blinnin malli vastaa paremmin todellista heijastusta tarkasteltavasta todellisesta pinnasta.

Ohjelma käyttää tiff-formaattia kuvien tallentamiseen. Ohjelma tuottaa täysväri- eli 24 bitin värikuvia.

Optiot

Optioiden avulla määrätään ohjelman toimintatapa. Optiot kirjoitetaan argumentteineen komentoriville ohjelman nimen jälkeen. Optioiden syötöjärjestyksellä ei ole mitään väliä, koska ohjelma aloittaa kuvan generoinnin optioiden tulkinnan jälkeen. ilman optioita käynnistys. Osa optioista vaatii argumentin, jolloin argumentti kirjoitetaan sellaisenaan option jälkeen. Jos ohjelma käynnistetään ilman argumentteja, ohjelma oletusarvoja optioille ja niiden argumenteille. Oletusarvoinen kuva on 500 kertaa 500 pikselin kokoinen kuva, jonka generointiin kuluu runsaasti aikaa.

Ohjelman käyttöohjeen tulostus.

- H Tulostaa kuvaruudulle lyhyen käyttöohjeen ja optioiden kuvaukset. Ohjelma lopettaa toimintansa tämän option jälkeen.

Seuraavilla optioilla määrätään kuvan dimensiot. Optiot vaativat lukuarvon argumenttikseen.

- x kuvan horisontaalinen eli x-akselin suuntainen koko. Koko ilmoitetaan millimetreinä. Oletusarvona kuvan x koolle on 10 mm.

- y kuvan vertikaalinen eli y-akselin suuntainen koko. Koko ilmoitetaan millimetreinä. Oletusarvona kuvan y-koolle on 10 mm.
- d pikselikoko mikrometreinä. Pikseli oletetaan neliön muotoiseksi generoitaessa kuvia. Pikselikoon ja kuvan x- ja y-dimensioiden perusteella voidaan määrittää kuvan sisältämät pikselimäärät. Oletusarvoisesti pikselikoko on 20 mikrometriä.

Seuraavilla optioilla määrätään tiedostot, joita ohjelma käyttää. Nämä optiot vaativat aina argumenttina tiedoston nimen.

- t tiff-kuvatiedoston nimi. Oletusarvoisesti, jos tätä optiota ei anneta kuva tallennetaan 'picture.tif' nimiseen tiedostoon. Ohjelma automaattisesti liittää päätteeseen '.tif' tiedostonimeen, jos sitä ei olla erikseen määrätty.
- p paperitiedoston nimi. Jos tätä optiota ei käytetä, ohjelma etsii oletusarvoisesti 'paper.p' nimistä tiedostoa. Ohjelma automaattisesti liittää päätteeseen '.p' tiedostonimeen, jos sitä ei olla erikseen määrätty.
- i painoväritiedoston nimi. Jos tätä tai '-I' optiota ei käytetä, ohjelma ei generoi painojälkeä kuvaan. Ohjelma automaattisesti liittää päätteeseen '.i' tiedostonimeen, jos sitä ei olla erikseen määrätty.
- l valolähdetiedoston nimi. Jos tätä optiota ei käytetä, ohjelma etsii oletusarvoisesti 'light.l' nimistä tiedostoa. Ohjelma automaattisesti liittää päätteeseen '.l' tiedostonimeen, jos sitä ei olla erikseen määrätty.

Jos halutaan generoida painojälki kuvaan täytyy määrittää, joko '-i' tai '-I' optio.

- I Ohjelma generoi painojäljen mukaan kuvaan. Ohjelma käyttää 'ink.i' nimistä tiedostoa generoitaessa painojälkeä. Tällä optiolla ei ole argumenttia.

Seuraavilla optioilla määrätään käytettävä heijastusmalli. Optiot '-P' ja '-B' ovat toisensa poissulkevia. Jos määrätään molemmat optiot, käytetään heijastusmallia, joka on määrätty jälkimmäisenä komentorivillä.

- P Ohjelma käyttää Phongin valaistusmallia. Tällä optiolla ei ole argumenttia. Jos ei ole erikseen määrätty muuta, käytetään Phongin heijastusmallia.
- B Ohjelma käyttää Blinnin valaistusmallia. Tällä optiolla ei ole argumenttia.

Ohjelmalle voidaan antaa tarkastelukulma. Tämä kulma ei kuitenkaan vastaa täysin todellisuutta, koska kuvalle ei suoriteta perspektiivi muunnosta.

- V Tarkastelukulma pinnan normalin suhteen. Kulma kertoo poikkeaman pinnan normaalista x-akselin suuntaan. Optio vaatii argumenttikseen kulman asteina. Oletusarvoisesti kulma on nolla astetta ja

maksimi kulma on 90 astetta.

Ohjelmassa on moodi, jolloin ohjelma voi generoida useita kuvia kerrallaan erilaisilla optioilla.

- f Ohjelma lukee optiot `'proof.scr'` nimisestä tiedostosta. Optiot luetaan tiedostosta rivikerrallaan kuin ne olisivat kommentorivillä annettuja optioita. Tällä optiolla ei ole argumenttia.

Scriptiajon toiminta

Useampia kuvia voidaan generoida kerralla, kun käytetään `proof`-ohjelman scripti-moodia. Scripti-moodi käyttää samoja optioita kuin kommentorivilläkin käytetään. Tiedosto, jota scripti-moodi käyttää on nimeltään `'proof.scr'`. Tiedosto sisältää optiot, joita halutaan syöttää ohjelmalle.

Scripti-moodi helpottaa useiden samankaltaisten kuvien genrointia, koska vanhat määrittymiset säilyvät, jos niitä ei kumota uusilla optioilla. Näin voidaan helposti generoida useita kuvia, joilla on esimerkiksi sama koko ja sama painovärimallin tiedosto, mutta paperimallit vaihtelevat. Myös kommentorivillä annetut optiot vaikuttavat, jos niitä ei kumota niillä optioilla, jotka on annettu scripti tiedostossa.

```
#
# This is an example of script file
#
-x 3 -y 3 -P -p pap11 -t all_p
-B -p pap11 -t all_b
-P -p pap12 -t a12_p
-B -p pap12 -t a12_b
```

Esimerkki 1. `'proof.scr'` tiedoston rakenne

Scripti-moodi generoi kuvat samassa järjestyksessä kuin ne on määritelty scripti-tiedostossa. Kommentteja voidaan lisätä tiedostoon, kirjoittamalla rivin alkuun `'#'` merkki, jolloin rivi jätetään huomiotta.

Tiedeostojen formaatit ja materiaalmallien sisältö

`Proof`-ohjelma lukee materiaalien mallien parametrit tiedostosta. Ohjelma tarvitsee aina paperin parametrit ja valolähteen parametrit sisältävät tiedostot. Jos ohjelma generoi painojäljen kuvaan, ohjelma tarvitsee painoväriin parametrit sisältävän tiedoston. Tiedostojen nimet ovat oletusarvoisesti `'paper.p'`, `'light.l'` ja `'ink.i'` ja ohjelma hakee näitä tiedostoja samasta hakemistosta missä ohjelma on käynnistetty ellei polkua anneta tiedoston nimen mukana.

Kaikissa kolmessa tiedostossa on samantapainen rakenne. Kommenttirivit alkavat `'#'` merkillä ja parametrit merkitään sanoilla, jotka on erikseen esitetty jäljempänä. Ohjelma tulkitsee parametrien haussa isot ja pienet kirjaimet, joilla on merkitys.

Ohjelma lukee ainoastaa parametrinimen ensimmäisen kirjaimen, jonka perusteella tulkitaan parametririvi. Parametrien järjestyksellä ei ole mitään väliä ainoastaan, jos käytetään pikselin kokoa sen tulisi olla määritettynä ensimmäiseksi.

Tiedostoissa on kolmea eri parametri tyyppiä. Tavalliset parametrit kirjoitetaan yhdelle riville, jotka alkavat parametrin määrittävällä sanalla ja parametrin arvolla tai arvoilla. Spektri tyyppisissä parametreissa on ensimmäisellä rivillä spektrin nimi sekä kertoimia ja seuraavat rivit sisältävät spektrin allonpituudet pienimmästä suurimpaan ja niiden vastaavat arvot. Alloonpituuksia ei tarvitse ilmoittaa tasaisin välimatkoin, koska ohjelma suorittaa lineaarisen interpolaation arvojen välillä. Spektri tyyppisten parametrien vektorit päättyvät aina kommenttiriviin. Lisäksi on matriisiparametrejä, joiden ensimmäisellä rivillä on matriisin määre, koko ja mahdollisesti kertoimia. Seuraavat rivit sisältävät matriisin arvot. Matriisi päättyy aina kommenttiriviin.

Paperin parametrit

Paperin mallin tärkein osa on karheusmatriisi, joka antaa karheusvaikutelman generoitavalle pinnalle. Värien ominaisuudet saadaan heijastusspektreistä. Kaikilla muilla paitsi karheusmatriisilla on oletusarvo, joten ne voidaan jättää määrittämättä.

`pixel_size` Tällä parametrilla määrätään pikselikoko karheusmatriisille ja betakulmamatriisille. Pikselikoko ilmaistaan mikrometreinä. Tämä parametri tulisi määrittää ensimmäisenä. Oletusarvona tällä parametrilla on $20\ \mu m$.

`Ambient` Tällä parametrilla määrätään ambientin valaistuksen aiheuttaman heijastuksen spektri ja kerroin. Ambientti termi on lineaarikombinaatio diffuusista ja spekulaarista heijastuksesta. Oletusarvona on tasainen spektri ja kerroin 0.20.

`Diffuse` Tällä parametrilla määrätään diffuusin heijastuksen spektri ja kerroin. Oletusarvona on tasainen spektri ja kerroin 0.20.

`Specular` Tällä parametrilla määrätään spekulaarin heijastuksen spektri, kerroin ja betakulman arvo. Betakulma kertoo kulman peiliheijastuskulmaan nähden, joka puolittaa heijastuneen intensiteetin. Diffuusin ja spekulaarin heijastusten kertoimien summa tulisi olla yksi. Beta kulma ilmaistaan radiaaneina ja betakulman vaihtelu määrätään erillisellä matriisilla. Oletusarvona on tasainen spektri, kerroin 0.80 ja betakulma 0.50.

`angle_of_beta` Tällä parametrilla määrätään betakulman vaihtelu. Tämän matriisin arvo lisätään edellä esitettyyn betakulmaan. Jos tätä matriisia ei määritetä, betakulmalla ei ole vaihtelua.

`contact` Tämä parametri kertoo kontaktietäisyyden pinnan maksimi korkeudesta. Oletusarvoisesti tapahtuu kontakti koko pinna korkeudelta.

`max_roughness` Tämä parametri kertoo maksimi karheuden karheusmatriisissa. Oletusarvona on $10.0\ \mu m$

roughness Tällä parametrilla annetaan karheus struktuuri generoitavalle paperin pinnalle. Tämä parametri pitää aina määrittää. Matriisin arvot ovat kokonaislukuja 0 ja 255 välillä. Mitä suurempi karheusmatriisi luodaan sitä monimutkaisempia pintoja pystytään genroimaan. Pienillä karheusmatriiseilla tulee pinnan jaksollisuus helposti esiin.

```
#This is an example file of paper structure
pixel_size      20.0
#
Ambient         0.4
380 0.3371
780 0.9215
#
Specular        0.60    0.5
380 1.0
780 1.0
#
Diffuse         0.40
380 0.0
430 0.3371
650 0.9215
780 0.0
#
angle_of_beta 2 2
0.0 0.5
0.5 0.0
#
contact 4.0
#
max_roughness 10.0
#
roughness 3 3
21 0 125
0 21 0
255 255 255
#
```

Esimerkki 2. Paperimallin tiedoston rakenne

Valolähteen parametrit

Valolähde koostuu yhdestä pistemäisestä valolähteestä ja ympäristön hajavalaistuksesta eli ambientista termistä. Diffuusit ja spekulaarit heijastukset ovat riippuvaisia pistemäisestä valolähteestä ja ambientti heijastus on riippuvainen ambientista valaistuksesta.

X, Y, Z Näillä parametreilla ilmaistaa pistemäisen valolähteen sijainti koordinaatistossa. Jotta valolähde valaisisi paperin pinnan tarkasteltavaa pintaa, tulisi Z termin olla positiivinen. Sijaintikoordinaatit ilmaistaan millimetreinä. Oletusarvona on sijainti (1000,1000,1000).

color Tämä termi kertoo pistemäisen valolähteen spektrin ja kerroin

kuvaa valolähteen intensiteettiä. Oletusarvona on tasainen spektri ja kerroin 255.

ambient_light Tämä termi kuvaa ambientin valaistuksen spektriä ja kerrointa. Oletusarvona on tasainen spektri ja kerroin nolla, joka tarkoittaa että ambienttia valaistusta ei oteta huomioon.

```
#
# This is an example file of Ligth structure
#
X      -0.7071
Y      -0.7071
Z       0.7071
#
color      3
380 0.4993
510 1.0778
630 0.8330
780 0.6340
#
ambient_light 0.6
380 0.4993
510 1.0778
630 0.8330
780 0.6340
#
```

Esimerkki 3. Valolähdemallin tiedoston rakenne

Painoväriin parametrit

Jos painojälki lisätään kuvaan käytetään painoväritiedostoa, joka sisältää painoväriin parametrit. Painovärimalli vaatii, että määrätään convolutiomatriisi ja kuvan tyyppi. Muilla parametreilla on oletusarvot, joita ohjelma käyttää, jos ei niitä ole määritetty.

pixel_size Tällä parametrilla määrätään pikselikoko kuvalle ja konvoluutiomatriisille. Pikselikoko ilmaistaan mikrometreinä. Tämä parametri tulisi määrittää ensimmäisenä. Oletusarvona tällä parametrilla on 20 μm .

splitting Tämä parametri kertoo painoväriin halkeamis kertoimen, joka on luku nollan ja yhden välillä. Oletusarvo on 0.5.

deposition Tämä parametri kertoo depositiokertoimen painoväriin. Oletusarvo on 0.5.

Absorption Tämä parametri kertoo painoväriin absorbtio kertoimen. Oletusarvo on 2.

Ambient Tällä parametrilla määrätään ambientin valaistuksen aiheuttaman heijastuksen spektri ja kerroin. Ambientti termi on lineaarikombinaatio diffuusista ja spekulaarista heijastuksesta. Oletusarvona on tasainen spektri ja kerroin 0.10.

Diffuse Tällä parametrilla määrätään diffuusin heijastuksen spektri ja kerroin. Oletusarvona on tasainen spektri ja kerroin 0.10 .

Specular Tällä parametrilla määrätään spekulaarin heijastuksen spektri, kerroin ja betakulman arvo. Betakulma kertoo kulman peiliheijastuskulmaan nähden, joka puolittaa heijastuneen intensiteetin. Diffuusin ja spekulaarin heijastusten kertoimien summa tulisi olla yksi. Beta kulma ilmaistaan radiaaneina. Oletusarvona on tasainen spektri, kerroin 0.90 ja betakulma 0.20 .

convolution Tällä parametrilla määrätään konvoluutiomatriisi, joka kuvaa painoväriin leviämistä painettavalle pinnalle. Matriisin kertoimien summan tulisi olla yksi. Matriisilla ei ole oletusarvoa.

```
# This is an example file of ink structure
pixel_size    20.0
#
splitting 0.5
#
deposition 0.5
#
Absorption 0.9
#
Ambient      0.2
380 0.0001
780 0.0001
#
Specular      0.8    0.05
380 0.5
780 0.5
#
Diffuse      0.2
380 0.0001
780 0.0001
#
convolution 2 2
0.0 0.25
0.25 0.50
#
location 1.5 1.5
#
#image DOT 2.0
image BOX 2 2 1.0
#image MATRIX 3 3 1.0
#000
#011
#100
#
```

Esimerkki 4. Painovärimallin tiedoston rakenne. Tässä painovärimallissa käytetään laatikkotyypistä kuviota. Muut kuviot on kommentoitu, jotta ne eivät häiritsisi painoväritiedoston tulkintaa.

location Tällä parametrilla määrätään kuvan sijainti

generoitavalla pinnalla. Kuvan sijaintikoordinaatit ilmaistaan millimetreinä. Kuvan parametrien tuli molempien olla positiivisia ja kuvan tarpeeksi suuri, jotta painojälki saadan kokonaan näkyviin. Oletusarvona on sijainti (0.0 , 0.0) kovalle.

`image` Tällä parametrillä määrätään kuvan tyyppi ja tarkasteltavalle pinnalle siirtyvän painoväriin paksuus. Painoväriin paksuus ilmaistaan mikrometreinä. Riipuen kuvan tyypistä on myös muita argumentteja parametrillä. Kuvan tyypit ovat piste 'DOT', laatikko 'BOX' ja kuvio 'MATRIX'. Kun generoidaan piste kuvaan, argumentiksi tarvitaan ainoastaan painoväriin paksuus esimerkiksi 'image DOT 1.5'. Kun painojäljeksi määritetään laatikko tarvitaan argumentteina myös laatikon koko. Laatikon koko ilmoitetaan millimetreinä. Esimerkiksi 'image BOX 1.5 3.0 1.0', jossa viimeinen argumentti on painoväriin paksuus. Lisäksi voidaan määrittää kuvio matriisiin muodossa, jossa matriisiin alkio saa arvon nolla tai yksi riippuen siitä sisältääkö kyseinen kohta painoväriä vai ei. Esimerkiksi 'image MATRIX 10 10 1.0', jossa kaksi ensimmäistä numeroargumenttia ilmaisevat matriisiin koon ja viimeinen argumentti ilmaisee painoväriin paksuuden. Matriisikuvio kirjoitetaan parametrin perään ilman tyhjiä merkkejä matriisialkioiden välissä.

Esimerkkejä

Paperin pinnasta kuvan tekeminen Blinnin mallia hyväksi käyttäen, jonka koko on 3 mm x 3 mm. Tulostetun kuvan nimi on 'picture.tif'

```
proof -x 3 -y 3 -B
```

Kuvan tekeminen painoväriä hyväksi käyttäen. Haettava painoväritiedosto on nimeltään 'ink.i'. Tulostettavan tiedoston koko on 5 mm x 5 mm, pikselikoko on 10 μm ja nimi on 'test.tif'

```
proof -x 5 -y 5 -d 10 -I -t test
```

Tulostetaan useita kuvatiedostoja joiden koko on 10 mm x 10 mm. Tässä täytyy kuitenkin ottaa huomioon se, että kuvan koko saatetaan muuttaa scriptitiedostossa 'proof.scr'

```
proof -x 10 -y 10 -f
```

Liite 4 Koesarja

Tässä on koesarjaan tekemiseen käytetty tiedosto `proof.scr`, joka sisältää argumentit `proof` ohjelmalle. Argumentit kertovat missä tiedostoissa sijaitsevat tarkasteltavien parametrien arvot. Ajokelpoinen tietokoneneohjelma ja koesarjan tiedostot sijaitsevat mukana tulevalla tietokonenelevykkeellä. Levyke on PC-formaattia oleva 3,5 “ 1,44 MB levyke.

Koesarjan voi suorittaa komennolla `'proof -f'`, jolloin ajetaan tässä olevan koesarjan mukainen testi. Ennen kuin ohjelma ajetaan, kannattaa tarkistaa, että kova-levyllä, johon tiedostot on kopioitu, on tarpeeksi tilaa. Koesarja tuottaa 88 tiff kuvatiedostoa, jotka tarvitsevat tilaa 16 MB. Koesarjan suorittamiseen menee 486DX 50 MHz tietokoneella noin 15 tuntia.

```
# Testi sarja valon suunnan muutoksien vaikutuksista
#   Kuvien koko 5 x 5 mm
#   Painoväri pinnan koko 2 x 2 mm
#   Paperi lajeja on kolme, jotka vaihtelevat karheutensa
#       suhteen, heijastus spektri on vakio kaikilla
#   Valon tulosuunta vaihtelee 0 ja 75 asteen välillä
#   Tarkastelusuuntana on joko 0 astetta tai valon
#       tulosuuntaan nähden vastakkainen suunta
#   Kuvat generoidaan sekä Blinnin että Phongin heijastus
#       mallia käyttäen
#
# Normaalit testi pinnat
#
-x5 -y5 -B -V0 -ppaper -llight -iink -tnormal_b
-x5 -y5 -P -V0 -ppaper -llight -iink -tnormal_p
#
#
#   Valon tulosuunnan vaikutus
#
-x5 -y5 -iink -ppaper -llight_0 -B -V0 -tld00_00b
-x5 -y5 -iink -ppaper -llight_0 -P -V0 -tld00_00p
#
-x5 -y5 -iink -ppaper -llight_25 -B -V0 -tld25_00b
-x5 -y5 -iink -ppaper -llight_25 -P -V0 -tld25_00p
-x5 -y5 -iink -ppaper -llight_25 -B -V25 -tld25_25b
-x5 -y5 -iink -ppaper -llight_25 -P -V25 -tld25_25p
#
-x5 -y5 -iink -ppaper -llight_35 -B -V0 -tld35_00b
-x5 -y5 -iink -ppaper -llight_35 -P -V0 -tld35_00p
-x5 -y5 -iink -ppaper -llight_35 -B -V35 -tld35_35b
-x5 -y5 -iink -ppaper -llight_35 -P -V35 -tld35_35p
#
-x5 -y5 -iink -ppaper -llight_45 -B -V0 -tld45_00b
-x5 -y5 -iink -ppaper -llight_45 -P -V0 -tld45_00p
-x5 -y5 -iink -ppaper -llight_45 -B -V45 -tld45_45b
-x5 -y5 -iink -ppaper -llight_45 -P -V45 -tld45_45p
#
-x5 -y5 -iink -ppaper -llight_75 -B -V0 -tld75_00b
-x5 -y5 -iink -ppaper -llight_75 -P -V0 -tld75_00p
-x5 -y5 -iink -ppaper -llight_75 -B -V75 -tld75_75b
-x5 -y5 -iink -ppaper -llight_75 -P -V75 -tld75_75p
#
```

```

-x5 -y5 -iink -ppaper -llight_85 -B -V0 -tld85_00b
-x5 -y5 -iink -ppaper -llight_85 -P -V0 -tld85_00p
-x5 -y5 -iink -ppaper -llight_85 -B -V85 -tld85_85b
-x5 -y5 -iink -ppaper -llight_85 -P -V85 -tld85_85p
#
#
#   Valon intensiteetin vaikutus
#
-x5 -y5 -iink -ppaper -llgt40_40 -B -V0 -tli40_40b
-x5 -y5 -iink -ppaper -llgt40_30 -B -V0 -tli40_30b
-x5 -y5 -iink -ppaper -llgt40_20 -B -V0 -tli40_20b
-x5 -y5 -iink -ppaper -llgt40_10 -B -V0 -tli40_10b
-x5 -y5 -iink -ppaper -llgt40_00 -B -V0 -tli40_00b
#
-x5 -y5 -iink -ppaper -llgt30_30 -B -V0 -tli30_30b
-x5 -y5 -iink -ppaper -llgt20_20 -B -V0 -tli20_20b
-x5 -y5 -iink -ppaper -llgt00_40 -B -V0 -tli00_40b
-x5 -y5 -iink -ppaper -llgt00_10 -B -V0 -tli00_10b
#
-x5 -y5 -iink -ppaper -llgt30_10 -B -V0 -tli30_10b
-x5 -y5 -iink -ppaper -llgt30_07 -B -V0 -tli30_07b
-x5 -y5 -iink -ppaper -llgt30_06 -B -V0 -tli30_06b
-x5 -y5 -iink -ppaper -llgt30_05 -B -V0 -tli30_05b
-x5 -y5 -iink -ppaper -llgt25_05 -B -V0 -tli25_05b
#
#
#   Paperin vaikutus
#
#   Beta
#
-x5 -y5 -iink -ppap_b01 -llight -B -V0 -tpb01b
-x5 -y5 -iink -ppap_b03 -llight -B -V0 -tpb03b
-x5 -y5 -iink -ppap_b05 -llight -B -V0 -tpb05b
-x5 -y5 -iink -ppap_b08 -llight -B -V0 -tpb08b
-x5 -y5 -iink -ppap_b12 -llight -B -V0 -tpb12b
#
#
#   Contact
#
-x5 -y5 -iink -ppap_c01 -llight -B -V0 -tpc01b
-x5 -y5 -iink -ppap_c03 -llight -B -V0 -tpc03b
-x5 -y5 -iink -ppap_c05 -llight -B -V0 -tpc05b
-x5 -y5 -iink -ppap_c08 -llight -B -V0 -tpc08b
-x5 -y5 -iink -ppap_c10 -llight -B -V0 -tpc10b
#
#
#   Max roughness
#
-x5 -y5 -iink -ppap_m05 -llight -B -V0 -tpm05b
-x5 -y5 -iink -ppap_m10 -llight -B -V0 -tpm10b
-x5 -y5 -iink -ppap_m16 -llight -B -V0 -tpm16b
-x5 -y5 -iink -ppap_m22 -llight -B -V0 -tpm22b
-x5 -y5 -iink -ppap_m40 -llight -B -V0 -tpm40b
#
#
#   Specular & diffuse
#
-x5 -y5 -iink -ppap_s00 -llight -B -V0 -tps00b

```

```
-x5 -y5 -iink -ppap_s10 -llight -B -V0 -tps10b
-x5 -y5 -iink -ppap_s30 -llight -B -V0 -tps30b
-x5 -y5 -iink -ppap_s50 -llight -B -V0 -tps50b
-x5 -y5 -iink -ppap_s60 -llight -B -V0 -tps60b
-x5 -y5 -iink -ppap_s80 -llight -B -V0 -tps80b
#
#
# Roughness Matriisin vaikutus
#
-x5 -y5 -iink -ppaper_07 -llight -B -V0 -tpr07b
-x5 -y5 -iink -ppaper_08 -llight -B -V0 -tpr08b
-x5 -y5 -iink -ppaper_11 -llight -B -V0 -tpr11b
#
-x5 -y5 -iink -ppaper_07 -llight -P -V0 -tpr07p
-x5 -y5 -iink -ppaper_08 -llight -P -V0 -tpr08p
-x5 -y5 -iink -ppaper_11 -llight -P -V0 -tpr11p
#
-x5 -y5 -iink -ppap_07_c -llight -B -V0 -tpr07_c_b
-x5 -y5 -iink -ppap_08_c -llight -B -V0 -tpr08_c_b
-x5 -y5 -iink -ppap_11_c -llight -B -V0 -tpr11_c_b
#
# Painoväarin vaikutus
#
# Absorptio value
#
-x5 -y5 -iink_a02 -ppaper -llight -B -V0 -tia02b
-x5 -y5 -iink_a09 -ppaper -llight -B -V0 -tia09b
-x5 -y5 -iink_a16 -ppaper -llight -B -V0 -tia16b
#
#
# Dekomposition level
#
-x5 -y5 -iink_d02 -ppaper -llight -B -V0 -tid02b
-x5 -y5 -iink_d05 -ppaper -llight -B -V0 -tid05b
-x5 -y5 -iink_d08 -ppaper -llight -B -V0 -tid08b
#
#
# Splitting
#
-x5 -y5 -iink_s02 -ppaper -llight -B -V0 -tis02b
-x5 -y5 -iink_s04 -ppaper -llight -B -V0 -tis04b
-x5 -y5 -iink_s05 -ppaper -llight -B -V0 -tis05b
-x5 -y5 -iink_s06 -ppaper -llight -B -V0 -tis06b
-x5 -y5 -iink_s08 -ppaper -llight -B -V0 -tis08b
#
#
# Specular & Diffuse
#
-x5 -y5 -iink_l00 -ppaper -llight -B -V0 -til00b
-x5 -y5 -iink_l03 -ppaper -llight -B -V0 -til03b
-x5 -y5 -iink_l05 -ppaper -llight -B -V0 -til05b
-x5 -y5 -iink_l08 -ppaper -llight -B -V0 -til08b
-x5 -y5 -iink_l99 -ppaper -llight -B -V0 -til99b
#
# End of test
#
```