

POSITRONIEMISSIOTOMOGRAFIAN
REKONSTRUKTIOALGORITMEISTA JA NIIDEN
SIMULAATIOTUTKIMUKSISTA

Jani Lindén

Pro gradu -tutkielma
Marraskuu 2013

MATEMATIIKAN JA TILASTOTIETEEN LAITOS
TURUN YLIOPISTO

Sisältö

1	Johdanto	1
1.1	PET-kuva ja rekonstruktio	1
1.2	Tutkielman tavoite ja materiaalit	4
2	Rekonstruktio menetelmien perusteita	6
2.1	Kuva-avaruus, kantafunktiot, projisiosuorat ja -matriisi	6
2.2	Analyttiset rekonstruktio menetelmät	13
2.3	Iteratiiviset algoritmit	17
2.3.1	Järjestettyjen osajoukkojen menetelmä	21
2.3.2	Projisiosuorittain laskettava algoritmi	23
3	Kuulista	27
3.1	Projektion laskennasta kuulahilassa	27
3.2	Kuulan määritelmä ja parametrit	31
3.3	Fourier-ehto	35
4	Rekonstruktion simulointi	40
4.1	Johdanto	40
4.2	Materiaali ja menetelmät	40
4.2.1	Ellipsoidin viivaintegraali	41
4.2.2	Projisiosuoran ja kuulahilan leikkauspinta	44
4.2.3	Viivaintegraalien testaus	46
4.2.4	Kuvan geometria ja rekonstruktion yksityiskohtia	48
4.2.5	Arviointimenetelmät	52
4.3	Rekonstruktioalgoritmin tulokset	54
4.3.1	Visuaalinen tarkastelu	55
4.3.2	Mielenkiintoalueiden analyysi ja koko kuvan virheet	56
4.3.3	Profilikäyrät	61
4.3.4	Muut RAMLA-parametrin λ arvot	66
5	Tulosten soveltaminen lääketieteellisessä kuvantamisessa	69
5.1	Johdanto	69
5.2	Materiaali ja menetelmät	71
5.3	Tulokset	71
6	Yhteenveto	76
	Kirjallisuutta	79
7	Liite I, rekonstruktioparametrien valintaan liittyvä koodi.	82

1 Johdanto

Tutkielman tavoite on johtaa ja ohjelmoida erilaisia kuvan rekonstruktio-menetelmiä positroniemissiotomografiassa (PET). Mallinnuksen pohjalla on kirjallisuudesta koottu teoria, jota on tutkielmassa kehitetty mallinnukseen vaadittavien parametrien valinnan helpottamiseksi. Teoriaosuuden jälkeen esitellään MATLAB-ohjelmointikielellä toteutetut simulaatiot niin analyytiseen, kuin iteratiiviseenkin rekonstruktioon. Tutkielma on tehty Turun yliopistollisen keskussairaalan PET-keskuksessa, yhteistyössä keskuksen tutkijoiden kanssa.

1.1 PET-kuva ja rekonstruktio

Positroniemissiotomografiassa (PET) käytetään sylinterin muotoista kuvantamislaitetta, jonka kehällä olevat detektorit havaitsevat kuvattavasta kohteesta tulevia fotoneja. Säteilylähde on kehon normaaleja molekyyleja jäljittelevä molekyyli, johon on lisätty radioaktiivinen isotooppi. Ennen radioaktiivista hajoamistaan merkkiaine noudattaa kehon omien molekyylien jakaumaa, jota on muuten mahdoton havainnoida. Positroniemissiotomografiassa käytetään positronisäteileviä isotooppeja, joista irtoavat positronit annihiloituvat väliaineen elektronien kanssa synnyttäen γ -säteilyä. Annihilatiopisteestä γ -hiukkaset lähtevät vastakkaisiin suuntiin, jolloin kahden havainnon sattuessa yhtä aikaa kahdessa eri detektorissa tiedetään, että annihilaatio on tapahtunut detektorien välisellä janalla, projisiosuoralla.

Kuvauksen aikana kerätään suuri määrä havaintoja tuhansille eri projisiosuorille. Kuvauksen jälkeen näistä havainnoista pitää matemaattisesti rekonstruoida kuva. Rekonstruktio menetelmät perustuvat viivaintegraaliin. Jokainen projisiosuora vastaa tiettyä viivaintegraalia yli kuva-avaruuden, ja rekonstruktio menetelmien tavoitteena on laskea mitatuista projisiosuorien arvoista hajoamisten todellinen jakauma avaruudessa.

Kuvan rekonstruktio menetelmät voidaan jakaa kahteen luokkaan: analyttisiin ja iteratiivisiin. Analyttiset menetelmät perustuvat olettamukseen, että kuva-avaruus ja mittaavaruus ovat molemmat jatkuvia, eli kamerassa pitäisi olla äärettömän monta detektoria. Jos tämä oletus olisi voimassa, voi-

taisiin kuva rekonstruoida täydellisesti havaituista projisiosuorien arvoista. Kameran detektorien ollessa kuitenkin diskreetisti aseteltuja, oletus rikkoutuu, ja kuvaan tulee virhettä. Analyttisistä menetelmistä esitellään tässä tutkielmassa suodatetun takaisinprojektion algoritmi (Filtered Back Projection, FBP [13]). Matemaattisessa mielessä FBP on algoritmina yksinkertainen. Rekonstruoitavan kuvan projisioista, eli Radon-muunnoksista, lasketaan Fourier-muunnokset. Nämä muunnokset suodatetaan Fourier-avaruudessa, minkä jälkeen kuva rekonstruoidaan ensin käänteisellä Fourier-muunnoksella, ja tämän jälkeen Radon-muunnoksen liittomuunnoksella, takaisinprojektiolla. FBP onkin nopea laskea, mutta avaruuden diskretisoinnista ja mittaukseen liittyvästä kohinasta johtuen se ei ole optimaalinen.

Iteratiivisissa menetelmissä lähdetään liikkeelle jostain kuvan alkuarvauksesta, minkä jälkeen havaittujen projisiosuorien arvojen avulla näitä alkuarvauksia muutetaan iteraatiokierros kerrallaan. Tässä tutkielmassa esitellään suurimman uskottavuuden estimointiin perustuvat odotusarvon maksimointi-algoritmi (Expectation Maximisation, EM [21]) sekä sen variantti, järjestettyjen osajoukkojen odotusarvon maksimointi-algoritmi (Ordered Subsets Expectation Maximisation, OSEM, [8]). Laskennallisesti iteratiiviset menetelmät ovat kuvan kantafunktioiden viivaintegraalien arvoja sisältävän projisioatriisin alustamisen jälkeen lineaarialgebraa sisältäen lähinnä vektorien sisätuloja sekä skalaarilla kertomista.

Riveittäin laskettava suurimman uskottavuuden algoritmi (Row Action Maximum Likelihood Algorithm, RAMLA [5]) on muokattu versio edellä mainituista iteratiivisista rekonstruktio menetelmistä. Se poikkeaa hieman EM- ja OSEM -algoritmeista, mutta sen on todistettu suppenevan samaan suurimman uskottavuuden ratkaisuun kuin EM ja OSEM. EM- ja OSEM- algoritmit päivittävät kuvaa joko kaikkien projisiosuorien arvojen laskemisen jälkeen tai OSEM-tapauksessa aina tietyn osajoukon läpikäymisen jälkeen. RAMLA taas päivittää kuvaa jokaisen projisiosuoran laskemisen jälkeen. Näillä muutoksilla RAMLA on saatu suppenemaan huomattavasti EM- ja OSEM-menetelmiä nopeammin.

Alkujaan kuvattavaa aluetta PET-kameran sisällä, kuva-avaruutta, on mallinnettu kiinni toisissaan olevilla kuutioilla, vokseleilla. Vokselit eivät me-

ne lainkaan päällekkäin mahdollistaen nopean laskennan. Rekonstruoitaessa kuvaa iteratiivisesti tulee jokaisen projisiosuoraa leikkaavan vokselin vaikutus laskea. Vokselin vaikutus projisiosuoraan saadaan laskemalla summa kunkin projisiosuoralle osuvan vokselin pisteen arvo, eli laskemalla viivaintegraali vokselin läpi. Vokselin yli integroiminen on yksinkertaista, koska vokselit saavat tasaisesti arvon 1 koko kuution sisällä. Rekonstruktion perustana, kantafunktioina, vokselit olivat ensimmäinen valinta juuri laskennan helppouden takia.

Vokselit ovat kuvan kantafunktioina kuitenkin karkeita, sillä niiden arvo muuttuu rajalla suoraan ykkösestä nolnaan. Samoin ne mallintavat mittausprosessia huonosti. Tätä korjaamaan on ehdotettu tilavuuselementtejä, joiden arvo lähestyy nolaa etäisyyden kasvaessa elementin keskuksesta. Kuvan arvon approksimaatio jokaisessa kuva-avaruuden pisteessä on ainakin kahden eri elementin arvojen yhdistelmä. Elementit asetetaan tasaisin välein kuva-avaruuteen, jossa ne muodostavat atomihilaa muistuttavan rakenteen.

Gaussiset kantafunktiot eivät sovellu tarkoitukseen hyvin, sillä ne pitää jossain vaiheessa katkaista, jolloin funktio muuttuu epäjatkuvuuspisteessä nolaksi, tai raja tasoitetaan jonkin toisen kaavan mukaisesti. Sen sijaan tutkielmassa esitellään perhe kolmiparametrisiä pallomaisesti symmetrisiä tilavuuselementtejä (kuulia, engl. blob), jotka ovat kellokäyrämäisen muotoisia kuten Gaussiset kantafunktiot, mutta jotka samalla vähenevät tasaisesti saavuttaen arvon nolla pallon reunalla.

Kuulilla on useita etuja vokseleihin nähden. Ne tuottavat sileämpiä kuvia, mutta parametreja sopivasti säätämällä voidaan tuottaa myös kontrastiltaan hyviä rekonstruktioita. Lisäksi kuulat lisäävät kuvaan vokseleita vähemmän kohinaa. Kuulat ovat kuitenkin laskennallisesti huomattavasti vaativampia kuin vokselit, sillä niiden saamat arvot kuva-avaruudessa muuttuvat jatkuvasti. Lisäksi kuulien päällekkäisyys kuva-avaruudessa lisää luonnollisesti niihin liittyvää laskentaa.

1.2 Tutkielman tavoite ja materiaalit

Tämän tutkielman tavoitteena on mallintaa ja simuloida kuulia sekä RAMLA-menetelmää tietokoneella. Ensin esitellään näihin liittyvä teoria, jonka jälkeen teoriaa sovelletaan simulaatioon. Simulaatiolla pyritään luomaan järjestelmä, johon voidaan sijoittaa minkä tahansa skannerin geometrian parametrit testausta varten. Simulaatioiden perusteella etsitään optimaaliset parametrien arvot niin kuvan kontrastin, virheen kuin kohinankin suhteen. Lopuksi optimaalisia parametreja testataan oikeilla mittauksilla ja vertaillaan PET-keskuksen Philips Gemini TF PET-MRI kameran oletusparametrien tuottamia kuvia simulaatiolla optimoituihin. Tavoitteena on kehittää parametrijoukkoja, jotka tuottavat haluttujen kelpoisuusvaatimusten mukaisia rekonstruktioita.

Luvussa 2 esitellään positroniemissiotomografiassa käytettävien rekonstruktio menetelmien perusteita. Ensiksi tarkastellaan yleisesti rekonstruktioon liittyviä ongelmia, ja luvun lopussa esitellään algoritmit analyttiseen ja iteratiiviseen ratkaisuun. Erityisesti tarkastellaan RAMLA-menetelmää, esitellään sen oletukset ja kaavat, sekä yhdistetään RAMLA-algoritmi OSEM-algoritmiin. Luku perustuu lähteisiin [1], [2], [3], [5], [6], [8], [11], [12], [13], [14], [19], [21] ja [24].

Luvussa 3 syvennyttään tarkemmin kuuliin. Luvussa tarkastellaan syitä kuulien käyttöön, erinäisiä laskennallisia etuja ja ongelmia, sekä menetelmiä joilla kuulien pallomainen luonne voidaan parhaiten hyödyntää laskennassa. Lisäksi tarkastellaan ehtoja parametrien valinnalle, sekä esitellään kuuliin liittyvät integraali- ja Fourier-muunnoskaavat. Luku perustuu lähteisiin [4], [7], [10], [15], [16], [17] ja [18].

Luvussa 4 johdetaan ja toteutetaan simulaatio mielivaltaisen PET-kameran geometriaan perustuen, sekä esitellään tulokset eri parametreille. Simulaatio toteutetaan MATLAB-ohjelmointikielellä. Simulaatio rakennetaan aikaisemmin käsitellyn teorian päälle, tietyin laskennallisiin tarkoituksiin tehdyin helpotuksin. MATLAB-ympäristön ja analyttisen aivofantomien [20] lisäksi simulaatiossa ei käytetä muita valmiita sovelluksia tai työkaluja, vaan simulaatio luodaan tyhjästä tätä tutkielmaa varten. Tulosten analyysissä apu-

na käytetään lähteitä [?] ja [23].

Luvussa 5 sovelletaan luvun 4 tuloksia oikeaan mittausdataan, sekä esitelään kuvauksiin liittyvät tulokset. Tuloksista käydään läpi eri iteraatiomääriin ja mielenkiintoalueille (Region of Interest, ROI) laskettavia kelpoisuuslukuja, ja vertaillaan luvussa 4 optimoituja parametreja skannerin oletusparametreihin. Apuna käytetään anatomista aivofantomia [9], josta rekonstruoi-tava data on saatu.

Tutkielmassa esiintyvät kuvat on piirretty Mathematica- ja MATLAB-ohjelmistoilla.

2 Rekonstruktio menetelmien perusteita

Seuraavassa luvussa tutustutaan PET-skannerin geometriaan ja siihen liittyviin käsitteisiin. Kuvattava alue, kuva-avaruus, täytetään tilavuuselementeillä, kantafunktioilla, joissa tapahtuvia hajoamisia lasketaan kerätyistä havainnoista. Havaintojen ja kantafunktioiden yhdistämiseksi määritellään ja johdetaan integraalit kantafunktioiden yli avaruudessa \mathbb{R}^3 .

Integraalien esittämisen jälkeen johdetaan analyttisiä ja iteratiivisia algoritmeja kuvan rekonstruoimiseen. Analyttisten menetelmien pohjalla oleva Fourier-leikelause todistetaan ja sen pohjalta esitetään kolme algoritmia. Iteratiivisten algoritmien käsittely aloitetaan johtamalla suurimman uskottavuuden estimaattiin pohjautuva algoritmi, jota muokkaamalla esitetään vielä kaksi muuta algoritmia, jotka suppevat suoraa suurimman uskottavuuden algoritmia nopeammin.

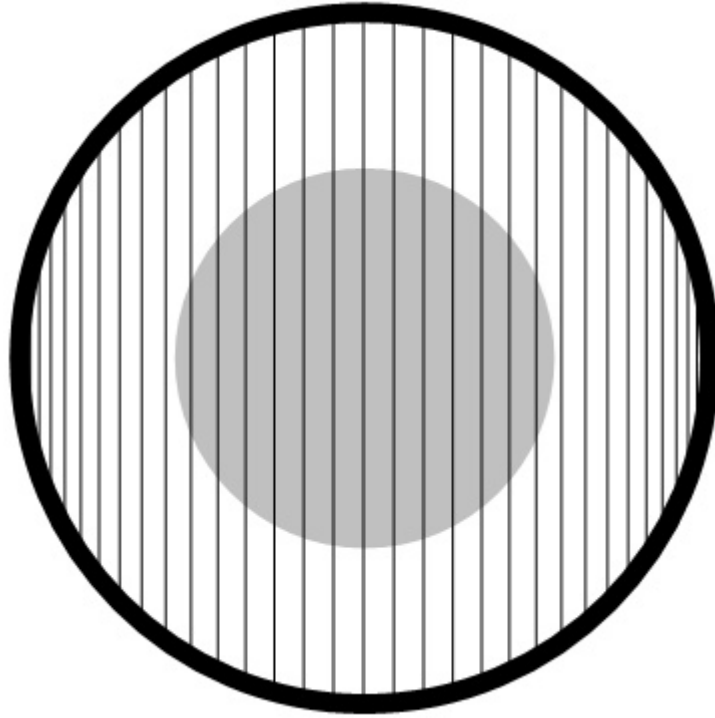
2.1 Kuva-avaruus, kantafunktiot, projisiosuorat ja -matriisi

PET-skannerissa on useita kehiä, joilla on jokaisessa sama määrä detektoreita. Detektorit yhdistäviä suoria kutsutaan projisiosuoriksi (LOR, Line of response). Jos jokaisella kehällä on D detektoria, niin on mahdollista määrittää

$$\binom{D}{2} = \frac{D!}{(D-2)!2!} = \frac{D(D-1)}{2}$$

projisiosuoraa jokaiselle kehälle. Tuloksena on kaksiulotteinen taso, leike, jokaiselle kameran kehälle. Lisäksi voidaan määrittää eri kehille päättyviä projisiosuoria, jolloin saadaan kameran akseliin nähden vinoja leikkeitä. Leikkeet ovat kaksiulotteisia tasoja, mutta tässä tapauksessa puhutaan kolmiulotteisesta (3D) keräyksestä.

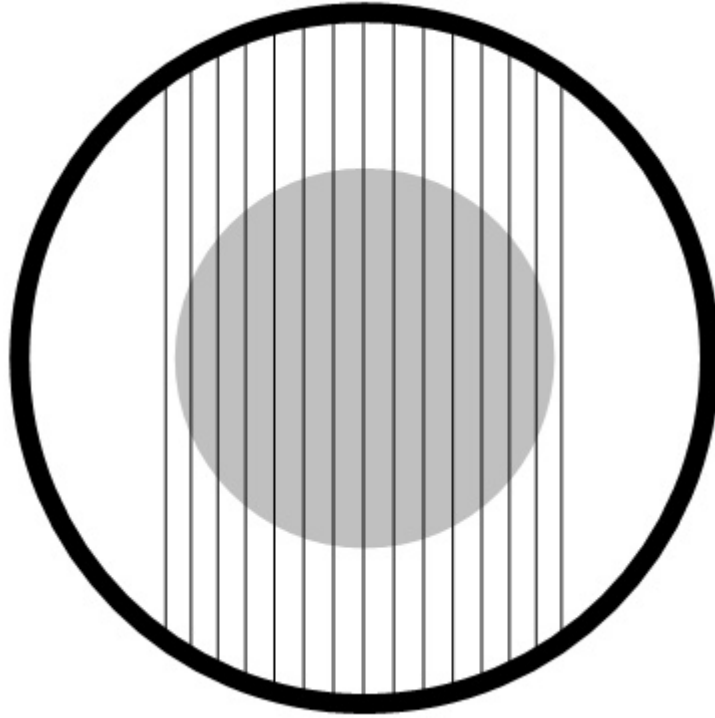
Kaikkien mahdollisten projisiosuorien käyttäminen ei kuitenkaan ole mielekäästä. Jokaisesta kulmasta katsottuna voidaan piirtää $D/2$ vierekkäistä projisiosuoraa. Kuvassa 1 on piirretty kuva tilanteesta, jossa $D = 74$ ja yhdensuuntaisia projisiosuoria on 37. Kuvan keskellä oleva harmaa ympyrä on kuvattava kohde. Kaikkein lähimmät projisiosuorat ovat skannerin geomet-



Kuva 1: Yhdestä kulmasta lähtevät 37 projisiosuoraa.

riasta johtuen hyvin lähellä toisiaan, eivätkä anna juuri mitään informaatiota kuvattavasta kohteesta, esimerkiksi vierekkäisten detektorien välillä tapahtunut hajoaminen ei voi olla lähtöisin kehän keskellä olevasta kuvattavasta kohteesta. Tällöin on perusteltua käsitellä ainoastaan esimerkiksi keskimäinen puolikas projisiosuorista, jolloin jokaista kulmaa kohti jää enää $D/4$ projisiosuoraa. Näin ollaan puolitettu tarvittava laskenta kuvanlaadun paljonkaan heikkenemättä. Kuvassa 2 on jätetty pois 22 laitimmaista projisiosuoraa. Kuvassa 3 on piirretty keskimäiset 15 projisiosuoraa kuudella eri kulmalla θ , jotka ovat kaikki $\pi/6$ välein toisistaan. Koska projisiosuorat kulkevat ringin reunasta reunaan, täyttävät välillä $[0, 5\pi/6]$ olevat projisiosuorien kulmat koko ringin tasaisesti.

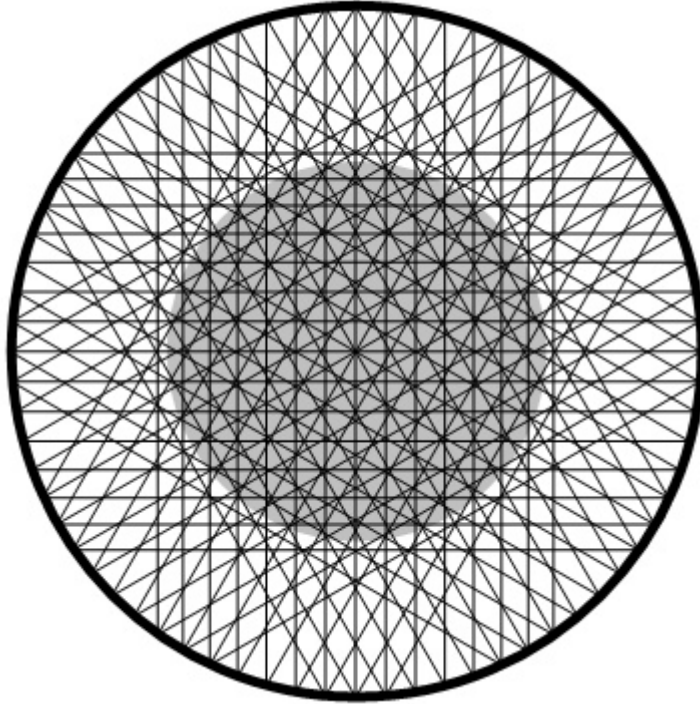
Jos kuvan jakauma tunnettaisiin, niin saataisiin integroimalla projisiosuorien yli laskettua arvot projisiosuorille. Nyt kuvan ollessa tuntematon, viivaintegraalien kaavojen avulla voidaan tunnetuista projisiosuorien arvoista laskea kuvan jakauma. Jotta voidaan integroida yli projisiosuorien, on mää-



Kuva 2: Yhdestä kulmasta lähtevät keskimmäiset 15 projisiosuoraa.

riteltävä ensin seuraavana esiteltävä kantafunktion käsite.

PET-skannerin kuvattava alue, kuva-avaruus T täytetään pisteillä (x, y, z) , jotka määritellään koordinaattiakselien suhteen. Olkoon nyt kuvaavaruudessa T yhteensä m kappaletta pistettä $\{(x_j, y_j, z_j)\}_{j=1}^m$, jotka indeksoidaan välille $j \in [1, \dots, m]$. Oletetaan, että pisteet (x_j, y_j, z_j) muodostavat kuvaavaruuteen tasaisen hilan solmut, joiden väli koordinaattisuunnissa on hilavakio Δ . Nyt pisteet (x_j, y_j, z_j) diskretisoivat kuvaavaruuden T . Viivaintegroinnin mahdollistamiseksi jokaiseen pisteeseen (x, y, z) liitetään kantafunktio $\Phi(x, y, z)$, joka antaa jokaiselle kuvaavaruuden T pisteelle jonkin arvon väliltä $[0, 1]$. Yleisesti $\Phi(x, y, z)$ on origokeskeinen, mutta se voidaan keskittää hilan pisteeseen (x_j, y_j, z_j) muuttamalla kantafunktion argumentit muotoon $\Phi(x - x_j, y - y_j, z - z_j) = \Phi_j(x, y, z)$. Merkitään kuvan arvoa pisteessä (x, y, z) merkinnällä $\bar{f}(x, y, z)$ ja vektorilla X kuvan arvoa hilapisteissä. Nyt approksimoidaan kuvaa siirrettyjen pisteiden ja kantafunktioiden



Kuva 3: Projisiosuorat kuudella eri kulmalla.

skaalattuna superpositiona seuraavasti:

$$\bar{f}(x, y, z) \approx \sum_{j=1}^m X_j \Phi_j(x, y, z). \quad (2.1)$$

Vokselitapauksessa, kantafunktio Φ valitaan siten, että se saa arvon 1, kun $|x|, |y|, |z| \leq \Delta/2$, ja arvon 0 muulloin. Tällöin yhtälössä (2.1) kantafunktio muuttuu muotoon

$$\Phi_j(x, y, z) = \begin{cases} 1, & \text{jos } \max(|x - x_j|, |y - y_j|, |z - z_j|) \leq \Delta/2, \\ 0, & \text{muulloin.} \end{cases}$$

Koska vokselit eivät mene päällekkäin, saa yhtälön (2.1) summa kussakin pisteessä (x, y, z) vain yhden nollasta eroavan termin, jolloin kuvan arvo on pisteessä (x, y, z) vaikuttavan vokselin arvo.

Tässä tutkielmassa käsiteltävän kuulan tapauksessa kantafunktio on pallomainen

$$\Phi(x, y, z) = b\left(\sqrt{x^2 + y^2 + z^2}\right) = b(r), \quad (2.2)$$

ja se saa nolasta poikkeavia arvoja, kun r on pienempi kuin kuulalle määriteltä säde a . Pallomaisuuden takia kuulien on mentävä osittain päällekkäin, jotta kuva-avaruuden T jokaiseen pisteeseen (x, y, z) osuu ainakin yksi kuula. kuulien tapauksessa määritellään, että jokaiseen kuva-avaruuden T pisteeseen (x, y, z) pitää ylettyä ainakin kaksi kuulaa, eli kuulun säde a on vähintään yhtä suuri kuin hilavakio Δ .

Määritetään projisiosuora $L_{s_i, \theta}$ avaruudessa \mathbb{R}^3 kaavalla

$$L_{(s_i, \theta)} = \{(x, y, z) \in \mathbb{R}^3 \mid s_i = x \cos \theta + y \sin \theta, z = z_0\}, \quad (2.3)$$

missä θ on kulma y -akselin suhteen ja s suoran kohtisuora poikkeama origosta. Kaava (2.3) määrittelee vain detektoritason sisäiset projisiosuorat. Myöhemmin koordinaatiston kierron yhteydessä projisiosuora laajennetaan myös detektoritasojen välisiin projisiosuoriin. Seuraavassa tarkastelussa oletetaan, että yksi ulottuvuuksista on vakio, eli $z = z_0$.

Olkoon $\bar{p}(s, \theta)$ kuvan \bar{f} projektio viivaa $L_{(s, \theta)}$ pitkin, eli Radon-muunnos $\bar{p}(s, \theta) = \mathcal{R}\{f(x, y)\}$. Viivaintegraali $\bar{p}(s, \theta)$ voidaan määritellä integraalina yli kaikkien suoran $L_{(s, \theta)}$ pisteiden. Käyttämällä apuna yhtälöä (2.1) saadaan

$$\begin{aligned} \bar{p}(s, \theta) &= \int_{L_{(s_i, \theta)}} \bar{f}(x, y, z) dl \\ &\approx \int_{L_{(s_i, \theta)}} \sum_{j=1}^m X_j \Phi_j(z, y, z) dl \\ &= \sum_{j=1}^m X_j \int_{L_{(s_i, \theta)}} \Phi_j(z, y, z) dl. \end{aligned} \quad (2.4)$$

Integraalirajojen saattamiseksi helpompaan muotoon otetaan käyttöön Diracin delta-funktio [6]

$$\begin{cases} \delta(x) = 0, & \text{kun } x \neq 0, \\ \int_{-\infty}^{\infty} \delta(x) = 1, & \text{kun } x = 0 \end{cases} \quad (2.5)$$

jonka avulla kaava (2.4) saadaan muotoon

$$\bar{p}(s_i, \theta) \approx \sum_{j=1}^m X_j \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(x \cos \theta + y \sin \theta - s_i) \Phi_j(x, y, z) dx dy. \quad (2.6)$$

Samoin yhtälön (2.4) ensimmäinen yhtäsuuruus voidaan kirjoittaa vastaavassa muodossa

$$\begin{aligned}
& \bar{p}(s_i, \theta) \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y, z) \delta(x \cos \theta + y \sin \theta - s_i) dx dy \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(s \cos \theta - u \sin \theta, s \sin \theta + u \cos \theta, z) \delta(0) \begin{vmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{vmatrix} ds du \\
&= \int_{-\infty}^{\infty} f(s_i \cos \theta - u \sin \theta, s_i \sin \theta + u \cos \theta, z) du,
\end{aligned} \tag{2.7}$$

missä suoritetaan normaali kaksiulotteinen koordinaatiston kierto

$$\begin{aligned}
\begin{pmatrix} s \\ u \end{pmatrix} &= \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \\
\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} s \\ u \end{pmatrix}
\end{aligned} \tag{2.8}$$

vastaamaan projisiosuoraa, sekä hyödynnetään kaavan (2.5) jälkimmäistä ehtoa. Kierto muuttaa integraalit yhtälössä (2.7) yksiulotteiseksi, sillä kierretyssä koordinaatistossa ei ole viivaintegraalin pisteitä u -akselin ulkopuolella.

Tarkastellaan yhtä projisiosuorien muodostamaa mielivaltaista leikettä, joka voi olla myös vino. Kierretään koordinaatistoa leikkeen suuntaiseksi siten, että päästään edellä olleeseen kaksiulotteiseen tilanteeseen eli $(x, y, z) \mapsto (s, u, t)$, jossa $t = t_0$ on vakio. Kierto suoritetaan matriisilla

$$\begin{aligned}
\Upsilon(\theta, \phi) &= \Upsilon_z(\theta) \Upsilon_x(\phi) = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix} \\
&= \begin{pmatrix} \cos \theta & \sin \theta \cos \phi & \sin \theta \sin \phi \\ -\sin \theta & \cos \theta \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix},
\end{aligned}$$

missä ϕ on projisiosuoran kulma suhteessa kameran akseliin ja θ detektoritason sisäinen projisiosuoran ja y -akselin välinen kulma. Koordinaattimuun-

nokset saadaan kaavoista

$$(s, u, t) = \Upsilon^{-1}(\theta, \phi)(x, y, z) = \begin{pmatrix} x \cos \theta - y \sin \theta \\ x \cos \phi \sin \theta + y \cos \phi \cos \theta - z \sin \phi \\ x \sin \phi \sin \theta + y \cos \theta \sin \phi + z \cos \phi \end{pmatrix}$$

ja

$$\begin{aligned} (x, y, z) = \Upsilon(\theta, \phi)(s, u, t) &= \begin{pmatrix} s \cos \theta + u \cos \phi \sin \theta + t \sin \phi \sin \theta \\ -s \sin \theta + u \cos \phi \cos \theta + t \cos \theta \sin \phi \\ -u \sin \phi + t \cos \phi \end{pmatrix} \\ &= \begin{pmatrix} \Xi \\ \Gamma \\ \Lambda \end{pmatrix}. \end{aligned}$$

Kierretyn koordinaatiston määrittämisen tason viimeinen koordinaatti t on projisiosuoria laskettaessa aina vakio. Projisiosuora $L_{s_i, \theta}$ saadaan nyt kaavasta

$$L_{(s, \theta)} = \{(s, u, t) \in \mathbb{R}^3 | s = s_i, t = t_0\},$$

missä verrattuna kaavaan (2.3) taso on kierretty projisiosuoran suuntaiseksi eli projisiosuoran kulma γ -akselin suhteen on 0. Siis $\cos(\theta - \theta) = \cos 0 = 1$, $\sin(\theta - \theta) = 0$ ja muuttuja t on vakio. Parametri s_i on edelleen projisiosuoran kohtisuora etäisyys origosta. Käyttäen hyväksi kiertoa saadaan yhtälö (2.6) muotoon

$$\begin{aligned} \bar{p}(s_i, \theta) &\approx \sum_{j=1}^m X_j \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(0) |\Upsilon| \Phi_j(\Xi, \Gamma, \Lambda) ds du \\ &= \sum_{j=1}^m X_j \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Phi_j(\Xi, \Gamma, \Lambda) ds du. \end{aligned} \tag{2.9}$$

Muuttujan t yli ei tarvitse erikseen integroida, sillä se on vakio. Lisäksi $|\Upsilon| = 1$.

Koska yhtälöiden (2.4) ja (2.9) integraalien arvoja tarvitaan jokaisella iteraatiokierroksella on ne järkevä laskea etukäteen ja tallentaa matriisimuodossa myöhempää käyttöä varten. Merkitäänkin reaaliarvoista projisiomatriisia

$A = \{a_{ij}\}_{i,j=1}^{n,m}$ ja

$$a_{ij} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Phi_j(\Xi, \Gamma, \Lambda) ds du = \int_{L(s_i, \theta)} \Phi_j(\Xi, \Gamma, \Lambda) dl.$$

Projisiomatriisissa on nyt n riviä, eli yhtä monta riviä kuin mallissa on projisiosuoria, sekä m saraketta, eli yhtä monta saraketta kuin kuva-avaruudessa T on alkioita (x_j, y_j, z_j) . Nyt voidaan merkitä

$$\bar{p}(s_i, \theta) = \bar{p}_i \approx \sum_{j=1}^m a_{ij} X_j.$$

Yhtälöstä (2.6) saadaan seuraavat jaksollisuus- ja symmetriaominaisuudet

$$\begin{cases} \bar{p}(s_i, \theta) = \bar{p}(s_i, \theta + 2k\pi), & \forall k \in \mathbb{N} \\ \bar{p}(s_i, \theta) = \bar{p}(-s_i, \theta \pm \pi), \end{cases}$$

joiden avulla projisiomatriisin alkioiden laskentaa voidaan nopeuttaa.

Projisiomatriisin A ja kuvavektorin X avulla mittaus voidaan esittää muodossa

$$b \approx AX, \quad (2.10)$$

jossa b on mitattujen havaintojen vektori kaikille projisiosuorille i . Iteratiivisten rekonstruktioalgoritmien tavoitteena on löytää kuvavektori X siten, että yhtälö (2.10) toteutuu. Tämä tapahtuu lähtemällä jostain alkuarvauksessa X^0 ja muuttamalla tuota alkuarvausta jonkin säännön mukaan, kunnes iteraatiokierroksella k yhtälö $b \approx AX^k$ on tarpeeksi lähellä tarkkaa ratkaisua. Myöhemmin johdettava suurimman uskottavuuden menetelmään perustuvissa iteratiivissa rekonstruktioalgoritmeissa tuloksena saatava kuvavektori X on jakauma, joka on suurimman uskottavuuden estimaatti havaituille arvoille b . Ennen iteratiivisten menetelmien johtamista on kuitenkin hyödyllistä tarkastella analyttisiä menetelmiä kuvavektorin X laskemiseksi.

2.2 Analyttiset rekonstruktio menetelmät

Analyttisissä rekonstruktio menetelmissä mitatut projisiosuorien arvot b_i käsitetään yhtälön (2.4) viivaintegraalien \bar{p}_i arvoiksi ja kuvan $\bar{f}(x, y, z)$ arvot

lasketaan Fourier-muunnoksia apuna käyttäen. Tässä tutkielmassa käsitellään nyt selkeyden vuoksi vain kaksiulotteista tapausta, jossa projisiosuorat kulkevat xy -tasolla. Kolmiulotteisen tapauksen todistus on huomattavasti pidempi, ja esitetty artikkelissa [2]. Todistetaan nyt kaksiulotteinen Fourier-leikelause (engl. Fourier Slice Theorem), jonka seurauksena saadaan analyyttisten rekonstruktio menetelmien peruslähtökohta.

Lause 1. *Fourier-leikelause*

Objektin kaksiulotteinen Fourier-muunnos keskuskulmalla θ on yhtäsuuri kuin objektin kulmassa θ lasketun projektion yksiulotteinen Fourier-muunnos.

Todistus. Käytetään kierrettyä koordinaatistoa kuten yhtälössä (2.7) ja merkitään lisäksi

$$Q = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix}.$$

Käytetään nyt apuna yhtälöä (2.7), koordinaatistomuutoksia (2.8) ja n -ulotteisen avaruuden Fourier-muunnoksen kaavaa

$$\hat{f}_n(\boldsymbol{\xi}) = \mathcal{F}_n(f(\mathbf{x})) = \int_{\mathbb{R}^n} f(\mathbf{x}) e^{-2i\pi \mathbf{x} \cdot \boldsymbol{\xi}} d\mathbf{x},$$

missä \mathbf{x} ja $\boldsymbol{\omega}$ ovat avaruuden \mathbb{R}^n vektoreita ja $\mathbf{x} \cdot \boldsymbol{\xi}$ on näiden sisätulo. Nyt polaarikoordinaatistossa määritetyn kuvan $f_p(\omega, \theta)$ kaksiulotteinen Fourier-muunnos voidaan kirjoittaa muodossa

$$\begin{aligned} \mathcal{F}_2(f_p(\omega, \theta)) &= \mathcal{F}_2(f(\omega \cos \theta, \omega \sin \theta)) \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-2i\pi(x\omega \cos \theta + y\omega \sin \theta)} dx dy \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-2i\pi\omega(x \cos \theta + y \sin \theta)} dy dx \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(s \cos \theta - u \sin \theta, s \sin \theta + u \cos \theta) e^{-2\pi s \omega} |Q| du ds \\ &= \int_{-\infty}^{\infty} \bar{p}(s, \theta) e^{-2\pi s \omega} ds = \mathcal{F}_1(\bar{p}(s, \theta)) = \hat{\bar{p}}(s, \theta), \end{aligned}$$

missä

$$|Q| = \begin{vmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{vmatrix} = 1.$$

□

Jos yhtälön viivaintegraalien $\bar{p}(s, \theta)$ yksiulotteiset Fourier-muunnokset täyttävät koko kaksiulotteisen Fourier-avaruuden, niin voidaan Fourier-leikelauseen mukaan kuva $f(x, y)$ rekonstruoida suoraan käänteisellä kaksiulotteisella Fourier-muunnoksella. Kuitenkin oikeissa sovellutuksissa joudutaan tinkimään täydestä kaksiulotteisesta Fourier-avaruudesta ja turvautumaan diskretisointiin, sillä projisiosuoria on vain äärellinen määrä. Tämä johtaa virheeseen rekonstruoidussa kuvassa. Lisäksi käänteinen kaksiulotteinen Fourier-muunnos on työläs laskea. Fourier-leikelauseen perusteella voidaan kuitenkin määrittää rekonstruktioalgoritmi 1, suora Fourier-rekonstruktio (Direct Fourier Reconstruction, DIFT) jossa $F_p(r, \phi)$ on napakoordinaattie-

Algoritmi 1 Suora Fourier-rekonstruktio

Syöte Projektiot $\bar{p}(s, \theta)$, kun $\theta \in [0, \pi)$

Tuloste Kuva $f(x, y)$

Suoritusvaiheet

- | | |
|--|--|
| 1. Laske muunnos | $\hat{p}(s, \theta) = \mathcal{F}_1(\bar{p}(s, \theta))$ |
| 2. Määritä napakoordinaattiesitys | $F_p(r, \phi) = \hat{p}(s, \theta)$ |
| 3. Muunna koordinaatisto karteesiseksi | $F_p(r, \phi) \mapsto F(\alpha, \beta)$ |
| 4. Laske käänteinen Fourier-muunnos | $f(x, y) = \mathcal{F}_2^{-1}(F(\alpha, \beta))$ |
-
-

sitys Fourier-avaruudessa ja $F(\alpha, \beta)$ karteesinen koordinattiesitys Fourier-avaruudessa.

Algoritmin 1 vaiheessa 3 tarvittavan koordinaattimuunnoksen laskemisessa esiintyy huomattavaa interpolaatiovirhettä, jos avaruus on kovin harvasti täytetty. Esitelläänkin takaisinprojektion käsite, jota hyödyntämällä saadaan ideaalitapauksessa algoritmin 1 kanssa saman vastauksen antavia algoritmeja, jotka kuitenkin harvasti täytetyn avaruuden tapauksessa tuottavat vähemmän virhettä.

Takaisinprojektio $b(x, y)$ määritellään integraaliin perustuvan takaisinprojektio-operaattorin \mathcal{B} avulla. Apuna käytetään yhtälön (2.8) koordinaattimuunnoksia

$$b(x, y) = \mathcal{B}(\bar{p}(s, \theta)) = \int_0^\pi g(s, \theta) d\theta = \int_0^\pi g(x \cos \theta + y \sin \theta, \theta) d\theta.$$

Funktiota $b(x, y)$ kutsutaan myös laminogrammiksi. Takaisinprojektiolla on seuraavat ominaisuudet:

1. Lineaarisuus:

$$\mathcal{B}(a_1 \bar{p}_1(s, \theta) + a_2 \bar{p}_2(r, \phi)) = a_1 \mathcal{B}(\bar{p}_1(s, \theta)) + a_2 \mathcal{B}(\bar{p}_2(r, \phi))$$

2. Siirtoinvarianttius:

$$b(x - a_1, y - a_2) = (\mathcal{B} \circ \mathcal{R})(f(x - a_1, y - a_2)) \quad (2.11)$$

3. Konvoluutio:

$$b(x, y) = f(x, y) * \frac{1}{\sqrt{x^2 + y^2}} \iff f(x, y) = \sqrt{x^2 + y^2} * b(x, y),$$

missä operaattori $*$ merkitsee konvoluutiota. Ominaisuuksien (2.11) avulla voidaan määrittää kaksi uutta rekonstruktioalgoritmia, joissa ei tarvitse laskea interpolaatiovirhettä tuottavaa koordinaattimuunnosta. Ensimmäisenä takaisinprojektio-suodatus-rekonstruktio (Backprojection Filter Reconstruction, BPF), jossa nimensä mukaan lasketaan ensin takaisinprojektio, minkä jälkeen lasketaan Fourier-muunnos ja dekonvoloidaan muunnettu takaisinprojektio. Lopuksi lasketaan käänteinen Fourier-muunnos lopullisen kuvan saamiseksi. Algoritmissa 2 joudutaan yhä laskemaan kaksiulotteinen Fourier-muunnos, sekä sen käänteismuunnos, mihin kuluu paljon laskenta-aikaa. Algoritmissa esiintyvää konvoluutiofunktiota $\sqrt{\alpha^2 + \beta^2}$ kutsutaan muotonsa mukaan kartiosuodatukseksi. Dekonvoluution voi tehdä myös yksiulotteisessa Fourier-avaruudessa projektion Fourier-muunnokseen. Tämän jälkeen suodatettu projektio saadaan käänteisellä Fourier-muunnoksella, minkä jälkeen varsinainen kuva saadaan takaisinprojektiolla. Tätä menetelmää kutsutaan suodatus-takaisinmuunnos-rekonstruktioksi (Filter Backprojektion, FBP), ja se on kuvattu Algoritmissa 3.

Algoritmissa 3 ei tarvitse enää laskea kaksiulotteista Fourier-muunnosta, jolloin suoritus aika paranee merkittävästi. Mikäli oltaisiin jatkuvassa ja ide-

Algoritmi 2 Takaisinprojektio-suodatus-rekonstruktio	
Syöte Projektiot $\bar{p}(s, \theta)$, kun $\theta \in [0, \pi)$	
Tuloste Kuva $f(x, y)$	
Suoritusvaiheet	
1. Laske takaisinprojektio	$b(x, y) = \mathcal{B}(\bar{p}(s, \theta))$
2. Laske Fourier-muunnos	$B(\alpha, \beta) = \mathcal{F}_2(b(x, y))$
3. Laske dekonvoluutio	$F(\alpha, \beta) = \sqrt{\alpha^2 + \beta^2} B(\alpha, \beta)$
4. Laske käänteinen Fourier-muunnos	$f(x, y) = \mathcal{F}_2^{-1}(F(\alpha, \beta))$

aalissa tilanteessa, kaikki kolme algoritmia tuottaisivat saman tuloksen, mutta oikeassa mittauksessa esiintyvä kohina, Algoritmin 1 interpolaatiovirheet harvassa kulma-avaruudessa ja Algoritmin 2 kaksiulotteinen Fourier-muunnos ovat johtaneet Algoritmin 3 muodostumiseen analyttisten rekonstruktio menetelmien standardiksi.

2.3 Iteratiiviset algoritmit

Analyttiset rekonstruktio menetelmät perustuvat sekä mittaus-, että kuva-avaruuksien jatkuvuusoletuksiin, joten ne ovat lähtökohtaisesti virheelliset, sillä oikeat mittaus- ja kuva-avaruudet ovat diskreettejä. Lisäksi mittaus on luonteeltaan tilastollista, sillä havaittu aineisto ei ole yksikäsitteisesti oikea, vaan on eräs satunnaisotos varsinaisesta jakaumasta. Tässä tutkielmassa käsiteltävät iteratiiviset rekonstruktio menetelmät perustuvat yhtälön (2.10) suurimman uskottavuuden ratkaisun iteroitintiin diskreetissä avaruudessa.

Olkoon A projisio matriisi ja X kuvavektori. Projisio matriisin alkio a_{ij} voidaan tulkita todennäköisyydeksi, että alkiossa j tapahtunut hajoaminen on havaittu projisiosuoralla i . Todennäköisyydet ovat toisistaan riippumattomia ja oletetaan, että ne noudattavat Poisson-jakaumaa. Olkoon $H(j)$ Poisson-jakautunut satunnaismuuttuja, sille montako fonia alkioista j lähtee kuvauksen aikana ja $P(i)$ Poisson-jakautunut satunnaismuuttuja havaituille foneille projisiosuoralla i . Nyt $H(j) \sim Po(X_j)$ ja $P(i) \sim Po(\lambda_i)$, joissa λ_i ja X_j ovat tuntemattomia keskiarvoja.

Todennäköisyys, että alkiossa j tapahtuu k hajoamista, on Poisson-

Algoritmi 3 Suodatus-takaisinprojektio-rekonstruktio

Syöte Projektiot $\bar{p}(s, \theta)$, kun $\theta \in [0, \pi)$ **Tuloste** Kuva $f(x, y)$

Suoritusvaiheet

- | | |
|-------------------------------------|--|
| 1. Laske Fourier-muunnos | $\hat{p}(\omega, \theta) = \mathcal{F}_1(\bar{p}(s, \theta))$ |
| 2. Laske dekonvoluutio | $\hat{P}(\omega, \theta) = \omega \hat{p}(\omega, \theta)$ |
| 3. Laske käänteinen Fourier-muunnos | $\bar{P}(s, \theta) = \mathcal{F}_1^{-1}(\hat{P}(\omega, \theta))$ |
| 4. Laske takaisinprojektio | $f(x, y) = \mathcal{B}(\bar{P}(s, \theta))$ |
-
-

jakauman todennäköisyysfunktion mukaan

$$P(H(j) = k) = e^{-X_j} \frac{X_j^k}{k!}.$$

Koska projisioatriisin A alkio a_{ij} tulkittiin toisistaan riippumattomiksi todennäköisyyksiksi, voidaan satunnaismuuttujalle $P(i)$ määrittää dekompositio, jossa

$$\lambda_i = \sum_{j=1}^m a_{ij} X_j.$$

Nyt todennäköisyys sille, että projisiosuoralla i tapahtuu b_i hajoamista, on

$$P(P(i) = b_i) = e^{-\lambda_i} \frac{\lambda_i^{b_i}}{b_i!} = \exp\left(-\sum_{j=1}^m a_{ij} X_j\right) \frac{\left(\sum_{j=1}^m a_{ij} X_j\right)^{b_i}}{b_i!}.$$

Kaikki projisiosuorien arvot b_i tunnetaan, mutta keskiarvoja λ_i ei, joten käytetään suurimman uskottavuuden menetelmää keskiarvojen estimointiin. Suurimman uskottavuuden funktio mitatuille havainnoille on

$$L(\boldsymbol{\lambda}) = L(\lambda_1, \lambda_2, \dots, \lambda_n | b_i) = \prod_{i=1}^n e^{-\lambda_i} \frac{\lambda_i^{b_i}}{b_i!}.$$

Suurimman uskottavuuden funktio kannattaa kuitenkin linearisoida ottamalla siitä logaritmi. Molemmilla funktioilla on sama maksimikohta. Logaritmi-

nen suurimman uskottavuuden funktio on muotoa

$$\begin{aligned}
l(\boldsymbol{\lambda}) &= \ln L(\lambda_1, \lambda_2, \dots, \lambda_n | b_i) = \ln \prod_{i=1}^n e^{-\lambda_i} \frac{\lambda_i^{b_i}}{b_i!} \\
&= \sum_{i=1}^n \left(\ln e^{-\lambda_i} \frac{\lambda_i^{b_i}}{b_i!} \right) \\
&= \sum_{i=1}^n (-\lambda_i + b_i \ln \lambda_i - \ln b_i!) \\
&= \sum_{i=1}^n \left(-\sum_{j=1}^m a_{ij} X_j + b_i \ln \left(\sum_{j=1}^m a_{ij} X_j \right) - \ln b_i! \right).
\end{aligned}$$

Viimeistä termiä ei tarvitse ottaa huomioon, sillä se on parametrien λ suhteen vakio, eli maksimoitavaksi jää lauseke

$$\begin{aligned}
l(\boldsymbol{\lambda}) &= \sum_{i=1}^n (-\lambda_i + b_i \ln \lambda_i) \\
&= \sum_{i=1}^n \left(-\sum_{j=1}^m a_{ij} X_j + b_i \ln \left(\sum_{j=1}^m a_{ij} X_j \right) \right) \quad (2.12) \\
&= l(\mathbf{X}).
\end{aligned}$$

Otetaan lausekkeen (2.12) osittaisderivaatta minkä tahansa parametrin X_j suhteen. Merkitään mielivaltaista parametria X_v , $v \in \{1, \dots, m\}$. Saadaan

$$\frac{\partial l(\mathbf{X})}{\partial X_v} = -\sum_{i=1}^n a_{iv} + \sum_{i=1}^n \frac{b_i a_{iv}}{\sum_{j=1}^m X_j a_{ij}}. \quad (2.13)$$

Derivoidaan yhtälö (2.13) uudelleen, tällä kertaa mielivaltaisen parametrin X_u suhteen, missä $u \in \{1, \dots, m\}$. Saadaan

$$\frac{\partial^2 l(\mathbf{X})}{\partial X_v \partial X_u} = -\sum_{i=1}^n \frac{b_i a_{iv} a_{iu}}{\left(\sum_{j=1}^m X_j a_{ij} \right)^2} \leq 0,$$

eli suurimman uskottavuuden funktio (2.12) on konkaavi. Nyt Karush-Kuhn-Tucker ehdoista [19] saadaan yhtälö mielivaltaisen parametrin X_v optimipis-

teelle

$$\begin{aligned}
0 &= X_v \frac{\partial l(\mathbf{X})}{\partial X_v} \\
&= -X_v \sum_{i=1}^n a_{iv} + \sum_{i=1}^n \frac{b_i X_v a_{iv}}{\sum_{j=1}^m X_v a_{ij}}
\end{aligned} \tag{2.14}$$

ja

$$\frac{\partial l(\mathbf{X})}{\partial X_v} \leq 0 \quad \text{jos } X_v = 0. \tag{2.15}$$

Yhtälössä (2.14) on kerrottu yhtälö

$$0 = \frac{\partial l(\mathbf{X})}{\partial X_v}$$

termillä X_v . Nyt oletetaan, että jokaisen projisiomatriisin sarakkeen summa on yksi, eli

$$\sum_{i=1}^n a_{ij} = 1 \quad \forall j \in \{1, \dots, m\}. \tag{2.16}$$

Oletus ei rajoita ratkaisun yleisyyttä, mutta yksinkertaistaa sitä. Käytännössä tämä ehto ei kuitenkaan toteudu, sillä kaikkia alkiosta j lähteviä fotoneja ei havaita äärellisellä määrällä detektoreja. Ehto (2.16) saadaan kuitenkin voimaan tekemällä muutos

$$\begin{cases} \epsilon_v = X_v \sum_{i=1}^n a_{iv} \\ \alpha_{ij} = a_{ij} / \sum_{i=1}^n a_{ij}, \end{cases} \tag{2.17}$$

jolloin yhtälöt (2.14) ja (2.15) saadaan muotoon

$$0 = -\epsilon_v + \sum_{i=1}^n \frac{b_i \epsilon_v \alpha_{iv}}{\sum_{j=1}^m \epsilon_j \alpha_{ij}} \tag{2.18}$$

ja

$$0 = -1 + \sum_{i=1}^n \frac{b_i \alpha_{iv}}{\sum_{j=1}^m \epsilon_j \alpha_{ij}} \leq 0 \quad \text{jos } \epsilon_v = 0. \tag{2.19}$$

Yhtälöt (2.18) ja (2.19) toteuttavat nyt ehdon (2.16) ja yhtälöt (2.14) ja (2.15) ratkaiseva kuvavektori \mathbf{x} saadaan muunnoksen (2.17) avulla yhtälöistä (2.18) ja (2.19), joten voidaan yleisyyttä rajoittamatta olettaa, että ehto (2.16) on voimassa.

Merkitsemällä yhtälön (2.18) ensimmäistä muuttujaa merkinnällä $\epsilon_v^{(k)}$, jossa k on iteraatiokierroksen indeksi ja jälkimmäisen termin muuttujaa merkinnällä $\epsilon_v^{(k-1)}$ saadaan odotusarvon maksimointialgoritmin (Expected Maximisation, EM) kaava

$$\epsilon_v^{(k)} = \epsilon_v^{(k-1)} \sum_{i=1}^n \frac{b_i \alpha_{iv}}{\sum_{j=1}^m \epsilon_j^{(k-1)} \alpha_{ij}}. \quad (2.20)$$

Jokaisella iteraatiokierroksella käydään läpi kaikki indeksit $v \in \{1, \dots, m\}$. Summan sisällä oleva lauseke on kaikille alkioille v sama, joten se tarvitsee laskea vain kerran iteraatiokierrosta kohti. Ottamalla muunnokset (2.16) toiseen suuntaan, saadaan kaava (2.20) kirjoitettua muotoon

$$X_v^{(k)} = \frac{X_v^{(k-1)}}{\sum_{i=1}^n a_{iv}} \sum_{i=1}^n a_{iv} \frac{b_i}{\sum_{j=1}^m x_j^{(k-1)} a_{ij}} = \frac{X_v^{(k-1)}}{\sum_{i=1}^n a_{iv}} \sum_{i=1}^n a_{iv} \langle A_i, X^{(k-1)} \rangle. \quad (2.21)$$

Varsinainen EM-algoritmi lähtee liikkeelle alkuarvauksesta $\mathbf{X}^0 > 0$, joka voi olla mielivaltainen, tai suppenemisen nopeuttamiseksi FBP-algoritmin tuottama kuva. Tämän jälkeen toistetaan kaavaa (2.21) kunnes algoritmi suppenee. Mikäli ratkaisu on olemassa algoritmi konvergoi siihen, sillä $l(\mathbf{x}^{k+1}) > l(\mathbf{x}^k)$ [24]. Algoritmi pysäytetään kun joko $\mathbf{X}^{k+1} = \mathbf{X}^k$ tai kun $l(\mathbf{x}^{k+1}) - l(\mathbf{x}^k) < \delta$, jossa δ on etukäteen määrätty vakio. Vaihtoehtoisesti algoritmi voidaan pysäyttää ennen suppenemista, kun indeksi $k = K$, jossa K on algoritmia alustettaessa päätetty iteraatiomäärä. EM-algoritmi on tiivistetty Algoritmissa 4.

2.3.1 Järjestettyjen osajoukkojen menetelmä

EM-algoritmissa kuva päivitettiin ainoastaan kerran kaikkien projisiosuorien vaikutuksen huomioimisen jälkeen. Tällöin algoritmin suppeneminen on hi-

Algoritmi 4 Iteratiivinen EM-rekonstruktio

Syöte Projektiot $\bar{p}(s, \theta)$, kun $\theta \in [0, \pi)$. Mahdollinen alkuarvaus \mathbf{X}^0

Tuloste Kuva $f(x, y)$

Suoritusvaiheet

1. Määritä lopetusehto:

- riittävän pieni ero iteraatioiden välillä tai
- tietty iteraatiomäärä.

2. Alusta projisiomatriisi A ja tarvittaessa alkuarvaus \mathbf{X}^0 .

3. Toista kaavaa (2.21), kunnes lopetusehto toteutuu.

dasta ja voidaan tarvita kymmeniä iteraatioita hyvälaatuisen kuvan tuottamiseen. Laskentaa saadaan nopeutettua päivittämällä kuvaa useammin jokaista iteraatiokierrosta kohti. Järjestettyjen osajoukkojen menetelmä (Ordered Subsets Expected Maximisation, OSEM) jakaa kuvan useaan osajoukkoon, joita jokaista käsitellään EM-menetelmän tavoin. Jokaisen osajoukon käsittelemisen jälkeen kuvaa päivitetään, joten kuvaa päivitetään iteraatiokierroksen aikana yhtä monta kertaa kuin rekonstruktiossa käytetään osajoukkoja.

Olkoon nyt k täyden iteraatiokierroksen indeksi ja $q \in \{1, \dots, p\}$ alisyklin indeksi ja S_q se projisiosuorien osajoukko, jota kulloinkin käsitellään. Määritellään alisyklit ja iteraatiot yhdistävät merkinnät $X^{(k,0)} = X^{k-1}$ ja $X^{(k,p)} = X^k$. Algoritmi suorittaa siis täyden iteraation, kun on käyty läpi kaikki alisyklit. Määritellään vielä summatermi

$$c_{qj} = \sum_{l \in S_q} a_{lj},$$

joka vastaa EM-algoritmissa käytettyä projisiomatriisin sarakesummaa

$$\sum_{i=1}^n a_{ij}.$$

Kirjoitetaan nyt EM-algoritmin kaava muodossa

$$X_j^{(k,q)} = \frac{X^{(k,q-1)} d_{qj}}{c_{qj}}, \quad \text{missä} \quad d_{qj} = \sum_{l \in S_q} \frac{a_{lj} b_l}{\langle A_l, X_j^{(k,q-1)} \rangle}, \quad (2.22)$$

indekseille $j = 1, \dots, m$, ja $q = 1, \dots, p$. Vektori A_l on projisiomatriisin rivi l ja osoittajan arvo b_l on mitattu projisiosuoran l arvo. Kaikki kuvan alkioiden j arvot päivitetään jokaisella alisyklillä i . Lauseke d_{qj} pysyy samoina kaikille alkiuille alisyksin sisällä. OSEM-algoritmi on esitetty Algoritmissa 5.

Algoritmi 5 Iteratiivinen OSEM-rekonstruktio

Syöte Projektiot $\bar{p}(s, \theta)$, kun $\theta \in [0, \pi)$. Mahdollinen alkuarvaus \mathbf{X}^0

Tuloste Kuva $f(x, y)$

Suoritusvaiheet

1. Määritä lopetusehto:
 - riittävän pieni ero iteraatioiden välillä tai
 - tietty iteraatiomäärä.
 2. Alusta projisiomatriisi A ja tarvittaessa alkuarvaus \mathbf{X}^0 .
 3. Määritä alisyklit S_q .
 4. Toista kaavaa (2.22), kunnes lopetusehto toteutuu.
-

2.3.2 Projisiosuorittain laskettava algoritmi

Riveittäin laskettava suurimman uskottavuuden algoritmi (Row Action Maximum Likelihood Algorithm, RAMLA) on muokattu versio edellisessä kappaleessa esitetystä OSEM-algoritmista. Siinä on muutamia eroavaisuuksia, joita käsitellään seuraavaksi.

Olkoon taas A aiemmin määritelty projisiomatriisi ja merkitään sen alkiota a_{ij} . Olkoon $X^0 > 0$ jokin alkuarvaus kuvan alkiuille sekä relaksaatioparametri $\lambda_k > 0$ siten, että $\lambda_k a_{ij} \leq 1$ kaikille $i \in \{1, \dots, n\}$ ja $j \in \{1, \dots, m\}$. Relaksaatioparametrin λ_k tulee täyttää ehdot

$$\begin{cases} i) & \lim_{k \rightarrow \infty} \lambda_k = 0, \\ ii) & \sum_{k=1}^{\infty} \lambda_k = \infty. \end{cases} \quad (2.23)$$

Käytännössä ei lasketa äärettömyyteen asti, joten selkeä väheneminen iteraatiokierrokselta toiselle riittää hyvälaatuisen kuvan tuottamiseen.

Analogisesti edellisen kappaleen OSEM-tapauksen kanssa; olkoon nyt k täyden iteraatiokierroksen indeksi ja $i \in \{1, \dots, n\}$ alisyklin indeksi eli se pro-

jisiosuora, jota kulloinkin käsitellään. Määritellään samoin myös alisyklit ja iteraatiot yhdistävät merkinnät $X^{(k,0)} = X^{k-1}$ ja $X^{(k,n)} = X^k$. Algoritmi suorittaa täyden iteraation, kun on käyty läpi kaikki projisiosuorat. Johdetaan RAMLA-kaava OSEM-algoritmista asettamalla ensin relaksaatioparametri $\lambda_k^i = 1/c_{ij}$ ja suurentamalla OSEM-tapauksen osajoukkojen määrä vastaamaan projisiosuorien määrää. Saadaan kaava

$$X_j^{(k,i)} = X_j^{(k,i-1)} \lambda_k^i d_i, \quad \text{missä } d_{ij} = \frac{a_{ij} b_i}{\langle A_i, x^{(k,i-1)} \rangle}. \quad (2.24)$$

Nyt $1/c_{ij} = 1/a_{ij} = \lambda_k^i$ eli $(1 - a_{ij} \lambda_k^i) = 0$, joten voidaan kirjoittaa kaava (2.24) muotoon

$$X_j^{(k,i)} = X_j^{(k,i-1)} (1 - a_{ij} \lambda_k^i) + X_j^{(k,i-1)} \lambda_k^i d_i, \quad \text{missä } d_{ij} = \frac{a_{ij} b_i}{\langle A_i, x^{(k,i-1)} \rangle}.$$

Vielä tässä vaiheessa arvon 0 saavaa ensimmäistä termiä tarvitaan jatkossa, jotteivat kaikki kuvavektorin alkiot suppene kohti nollaa parametrin λ lähestyessä nollaa iteraatiomäärän kasvaessa. Kiinnitetään nyt relaksaatioparametri λ vakioksi kaikille alisykleille $i \in \{1, \dots, n\}$, sekä vastaamaan ehtoja (2.23), jolloin saadaan varsinainen RAMLA-kaava

$$X_j^{(k,i)} = X_j^{(k,i-1)} (1 - \lambda_k a_{ij}) + X_j^{(k,i-1)} \lambda_k a_{ij} d_i, \quad (2.25)$$

missä

$$d_i = \frac{b_i}{\langle A_i, X^{(k,i-1)} \rangle}$$

ja λ_k pysyy vakiona koko iteraatiokierroksen ajan eli laskettaessa yli kaikkien alisykliä $i = 1, \dots, n$. Projisiosuorien läpikäymisjärjestyksellä on väliä lopputuloksena saatavan kuvan kannalta, mutta eri tapoja järjestää projisiosuorat ei tarkastella tässä tutkielmassa. Termi d_i saatiin myös riippuvaiseksi ainoastaan kulloinkin käsiteltävästä alisyklistä, joten sen arvoa voidaan käyttää jokaiselle kuvavektorin alkiolle. Tämä antaa pienen laskenta-ajan nopeutuksen OSEM-algoritmiin verrattuna.

Artikkelissa [5] kirjoitettiin RAMLA-kaava yhtäpitävään muotoon

$$X_j^{(k,i)} = X_j^{(k,i-1)} + \lambda_k X_j^{(k,i-1)} a_{ij} \left(\frac{b_i}{\langle A_i, X^{(k,i-1)} \rangle} - 1 \right). \quad (2.26)$$

Sulkujen sisällä oleva lauseke pysyy samoina kaikille kuvan alkiuille. Kaavasta (2.26) nähdään koko RAMLA-algoritmin idea. Kaavassa ensimmäisenä terminä olevaan vanhaan kuvavektorin arvoon $X_j^{(k,i-1)}$ lisätään tai siitä vähennetään sulkujen sisällä laskettu projisiosuorakohtainen suhteellinen ero, joka skaalataan kuhunkin kuvavektorin alkioon X_j liittyvällä projisiomatriisin alkiolla a_{ij} . Relaksaatioparametri λ rajoittaa jälkimmäisen termin tuottamaa muutosta ja iteraatioiden edetessä nopeuttaa kuvan konvergenssia pakottamalla jälkimmäisen termin lähestymään nollaa. RAMLA-algoritmin vaiheet on kuvattu Algoritmissa 6.

Algoritmi 6 Iteratiivinen RAMLA-rekonstruktio

Syöte Projektiot $\bar{p}(s, \theta)$, kun $\theta \in [0, \pi)$. Mahdollinen alkuarvaus \mathbf{X}^0

Tuloste Kuva $f(x, y)$

Suoritusvaiheet

1. Määritä lopetusehto:

- riittävän pieni ero iteraatioiden välillä tai
- tietty iteraatiomäärä.

2. Alusta projisiomatriisi A ja tarvittaessa alkuarvaus \mathbf{X}^0 .

3. Määritä järjestys projisiosuorien läpikäymiseen.

4. Toista kaavaa (2.27), kunnes lopetusehto toteutuu.

Kaavoista (2.21), (2.22) ja (2.25) nähdään, että ensimmäinen termi on RAMLA-menetelmässä positiivinen ja toinen termi on selvästi ei-negatiivinen, kunhan termille b_i määrätään ei-negatiivisuusrajoite. EM ja OSEM-algoritmeissa ei ole RAMLA-kaavan ensimmäistä termiä, mutta nekin ovat ei-negatiivisia kunhan havainnoille b_i on määrätty ei-negatiivisuusrajoite. Tällöin positiivisesta alkuarvauksesta lähtien pysytään aina positiivisella lukualueella kaikille alkiuille j . Ei-negatiivisuusrajoite saadaan voimaan joko muuttamalla kaikki sirona- ja satunnaishavaintokorjauksen jälkeen negatiiviset termit b_i positiivisiksi, tai ottamalla termeiksi b_i kaikki havainnot, ja lisäämällä sironnan sekä satunnaisten havaintojen vaikutus nimittäjään. Tällöin RAMLA-kaava (2.26) muuttuu muotoon

$$X_j^{(k,i)} = X_j^{(k,i-1)} + \lambda_k X_j^{(k,i-1)} a_{ij} \left(\frac{b_i}{\langle A_i, X^{(k,i-1)} \rangle + s_i + r_i} - 1 \right), \quad (2.27)$$

jossa s_i kuvastaa arvioituja sironneita havaintoja ja r_i arvioituja satunnaisia havaintoja. Lisäksi kaikkiin termeihin a_{ij} on laskettu attenuaatiokorjaus, sekä termeissä b_i on otettu huomioon normalisaatio-, puoliintumis- ja kiteiden palautumisaikakerroin.

EM-kaava vaatii laskennallisesti asympotoottisesti yhtä paljon työtä iteraatiota kohden kuin RAMLA (jokainen alisykli vastaa EM-summan yhtä termiä). Samoin laskentatyö OSEM-menetelmään nähden on asympotoottisesti yhtä suuri täyttä iteraatiota kohden, sillä RAMLA on vain edellä kuvatulla tavalla muokatun OSEM-algoritmin erityistapaus.

Artikkelissa [5] saatiin seuraavanlainen vastaavuus EM ja RAMLA iteraatioiden kontrastin välille. 45 iteraatiota EM:llä vastasi yhtä RAMLA iteraatiota, sekä 80 EM iteraatiota vastasi 4 RAMLA iteraatiota.

Relaksaatioparametriin λ_k on kiinnitettävä huomiota, sillä konvergenssin nopeus ja kuvan laatuominaisuudet iteraatiota kohden muuttuvat, mikäli relaksaatioparametria ja sen kehittymistä muutetaan.

3 Kuulista

Tutkielmassa tarkasteltavat pallomaiset tilavuuselementit, kuulat, on järjestetty samalla tavalla karteesiseen hilaan kuin vokselitkin, keskipisteiden välin ollessa Δ . Vokseleista poiketen kuulat menevät kuitenkin myös päällekkäin. Minimivaatimuksena on, että jokaisessa avaruuden pisteessä ollaan vähintään kahden kuulan sisä- tai reunapisteessä. Määritellään kuulan säde a hilavakion Δ avulla $a = k\Delta$, $k \in \mathbb{R}_+$. Laskennan keskittyessä kuula-avaruuteen voidaan joko olettaa, että hilavakio $\Delta = 1$, tai sitten Δ yksikköineen kumoutuu pois laskennan aikana. Näin säteellä a ei ole laskuissa mittayksikköä lainkaan. Pie-nin mahdollinen säde on 1, jolloin kuulan reunat hipaisevat viereisen kuulan keskipistettä. Tätä tilannetta on kuvattu kuvassa 4 (vasen).

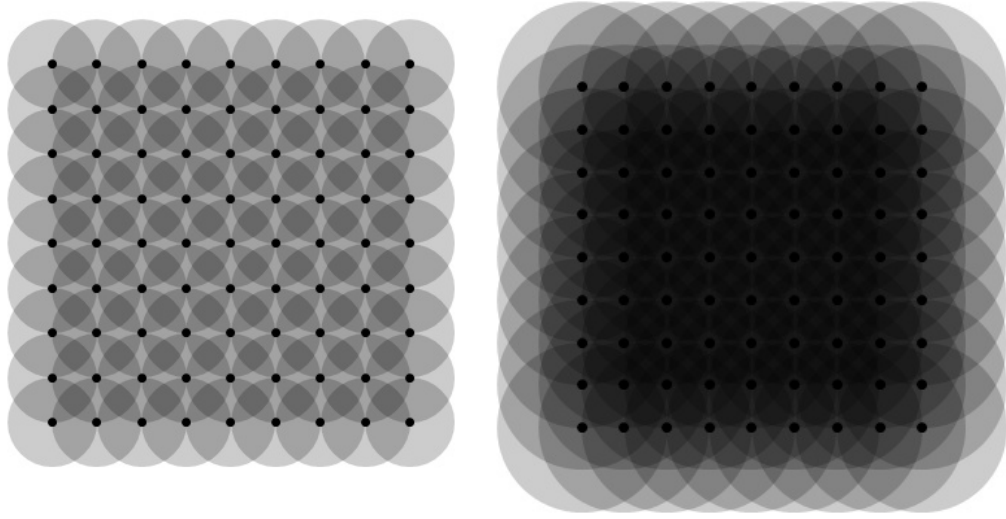
Näin harva hila ei kuitenkaan ole optimaalinen, sillä avaruus täyttyy epätasaisesti kuulien vaikutuksesta. Usein onkin kuvan kannalta parempi määrittää kuulien säde $a > 1$. Kuvassa 4 (oikea) on piirretty tilanne jossa $a = 2$.

Toisin kuin vokselit kuulat eivät saa vakioarvoa 1 sisäpisteissä, vaan kuulan arvo putoaa keskipisteen arvosta 1 rajapisteen arvoon 0. Seuraavassa kappaleessa tarkastellaan parametreja, jotka vaikuttavat kuulan muotoon, sekä esitetään kuvaa 4 vastaava tilanne muotoparametrin vaikutuksen huomioon ottamisen jälkeen.

Vokselihila on rajoittunut karteesiseen rakenteeseen, sillä vokselit eivät mene päällekkäin. Kuulat taas voidaan pakata myös vapaammin, sillä kuulien tapauksessa päällekkäisyyttä jo määritelmällisesti vaaditaan. Pakkausteorias-ta saadaan kaksi vaihtoehtoa karteesiseen hilaan, kappalekeskeinen kuutio (Body-Centered Cube, BCC) sekä tahkokeskeinen kuutio (Face-Centered Cu-be), kuva 5. Lisäksi kuulia on mahdollista pakata esimerkiksi kuusikulmai-sesti, mutta tällaista tapausta ei tarkastella tässä tutkielmassa.

3.1 Projektion laskennasta kuulahilassa

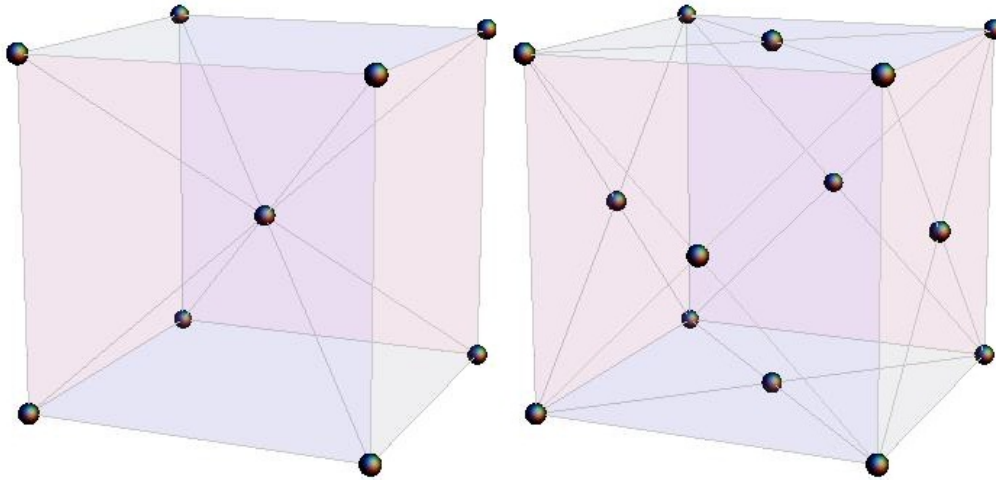
Projisiomatriisin suuren koon vuoksi (projisiomatriisi voi sisältää jopa 10^{13} alkiota) ei ole järkevää pitää sitä kokonaan muistissa, vaan on tarkoituksen-mukaisempaa hyödyntää kuulien ominaisuuksia laskennan vähentämiseksi. Jokainen kuula on pallomaisesti symmetrinen, joten riittää taulukoida vii-



Kuva 4: Kuulahilat säteen arvoilla $a = 1$ (vasen) ja $a = 2$ (oikea). Vasemmasta kuvasta nähdään kuulien välissä olevat liian pienestä säteestä johtuvat vaaleammat alueet.

vaintegraalin arvoja yhdeltä tasolta ja sitten tarpeen mukaan siirtää projisiota kaikkien mahdollisten projisiosuorien laskemiseen. Tuloksena on hakutaulu. Hakutaulussa on taulukoituna viivaintegraalien arvot projisiosuorille, jotka kulkevat eri etäisyydellä kuulan keskipisteestä. Nyt voidaan kuulien ja projisiosuorien läpikäynti suorittaa kahdella tapaa. Kuulalähtöisessä algoritmossa käydään läpi yksi kerrallaan kaikki kuula-avaruuden alkiot, ja tarkastetaan, mitkä projisiosuorat osuvat kullekin kuulalle. Kun löydetään projisiosuora, joka kulkee kuulan läpi, katsotaan, mikä hakutaulun alkioista vastaa parhaiten kyseessä olevaa tilannetta, ja valitaan tämän alkion arvo viivaintegraalin arvoksi.

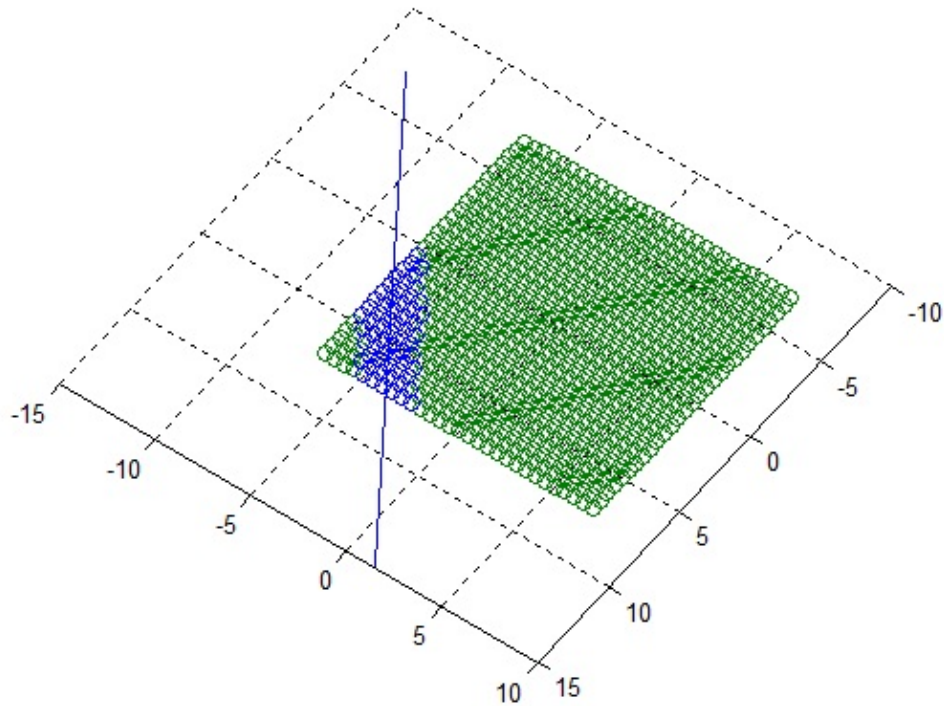
Projisiosuoralähtöisessä lähestymistavassa lasketaan projisiosuora kerrallaan kaikki kyseiseen projisiosuoraan vaikuttavat kuulat ja niiden viivaintegraalit kyseiselle projisiosuoralle. Tämä tapahtuu ensiksi laskemalla projisiosuoralle suuntavektori (θ, ϕ) siten, että θ on detektoritason sisäinen kulma ja ϕ on projisiotason ja kameran akselin välinen kulma ("tilt"). Seuraavaksi määritetään se kuulahilan karteesinen koordinaattiakseli, jota suuntavektori on lähinnä. Olkoon tämän koordinaattiakselin suuntavektori η . Tämän jäl-



Kuva 5: BCC-hila (vasen) ja FCC-hila (oikea).

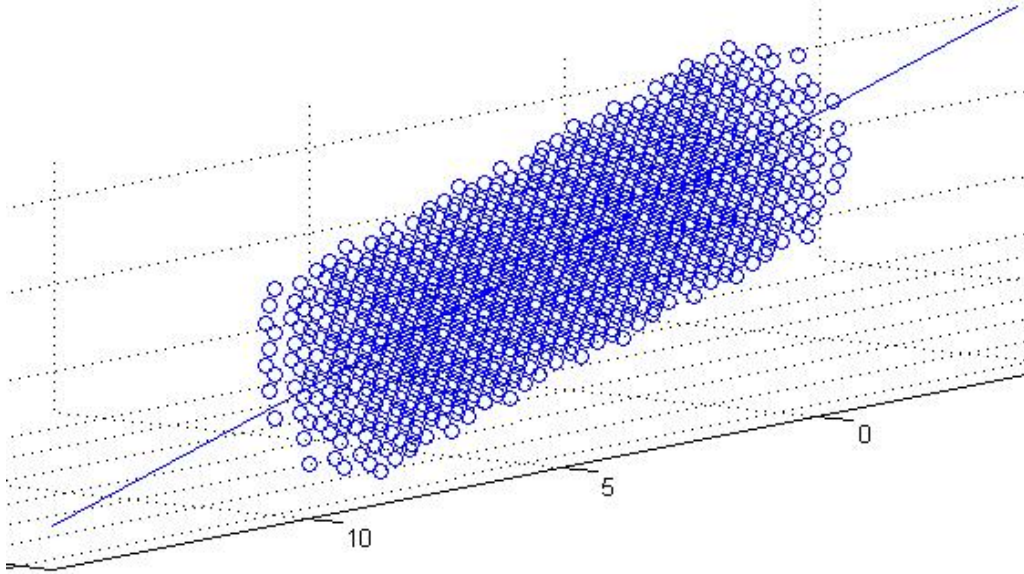
keen määritetään halkaisijaltaan kuulan halkaisijan suuruinen sylinteri, jonka akselina projisiosuora kulkee. Sylinterin alkupää on projisiosuoran lähtödetektorissa ja toinen pää päätedetektorissa. Kulkiessaan kuva-avaruuden T läpi sylinteri leikkaa koordinaattisuuntien virittämiä tasoja. Sylinterin kohdatessa kuulahilan tason, jonka normaalivektori on η , muodostuu ellipsin muotoinen leikkauspinta. Kaikki kuulat, joiden keskipisteet sijaitsevat tällä ellipsillä, kontribuovat kyseiseen projisiosuoraan, sillä projisiosuora kulkee kuva-avaruudessa T niiden ulkopinnan sisäpuolella. Tämän jälkeen jokaisen kontribuoivan kuulan arvo kyseiselle projisiosuoralla lasketaan viivaintegraalilla, sillä alueella, jolla projisiosuoran ja kuulan keskipisteen välinen etäisyys on pienempi kuin kuulan säde. Tilannetta on havainnollistettu kuvissa 6 ja 7.

Jos kuulahila on asetettu symmetrisesti skannerin geometriaan nähden samaa elliptistä leikkauspintaa ja sen kuulien kontribuutioita voidaan käyttää triviaalein muutoksin usealla projisiosuoralle. Viivaintegraaleja tietyille kulmille ja projisiosuorien etäisyyksille kuulien keskipisteistä voidaan laskea etukäteen ja muodostaa systeemimatriisista tiivistetty jalanjälkitaulukko (foot print table). Jalanjälkitaulukossa on indeksoitu ne kuulat, jotka osuvat kullekin projisiosuoralle. Näin ollen ne pitää luonnollisesti identifioida pelkän painon antamisen lisäksi. Lisäksi projisiosuorista tallennetaan kulmiin ja de-



Kuva 6: Yhdellä kuulatasolla projisiosuoralle osuvat kuulat.

tektoreihin liittyviä tietoja. Jokaista projisiosuoran kulmaa ja etäisyyttä origosta varten tallennetaan suoraan vain yksi jalanjälkitaulukon rivi. Lopuille saman suuntaisille projisiosuorille voidaan laskea niille osuvat kuulat vain indeksejä muuttamalla. Lisäksi kuulahilan ollessa pääakselien suhteen symmetrinen samalla projisiotasolla kulmat ovat symmetrisia siten, että neljää kulmaa vastaavat projisiosuorat voidaan yhdistää yhdeksi jalanjälkitaulukon riviksi $(\theta, \pi/2 - \theta, \pi/2 + \theta$ ja $\pi - \theta)$. Vinojen projisiotasojen kulmat ϕ ja $-\phi$ vastaavat toisiaan, jolloin nekin voidaan yhdistää. Näin jalanjälkitaulukko voidaan tiivistää vielä kahdeksasosaan pelkkään indeksivaihtoon perustuvasta tiivistyksestä. Jalanjälkitaulukon luomisen jälkeen tietylle projisiosuoralle osuvien kuulien vaikutuksen laskeminen ja iteratiivisten algoritmien käyttäminen on varsin suoraviivaista.



Kuva 7: Kaikki projisiosuoralle osuvat kuulat.

3.2 Kuulan määritelmä ja parametrit

Määritellään kuulan arvo etäisyydellä r keskipisteestään seuraavasti

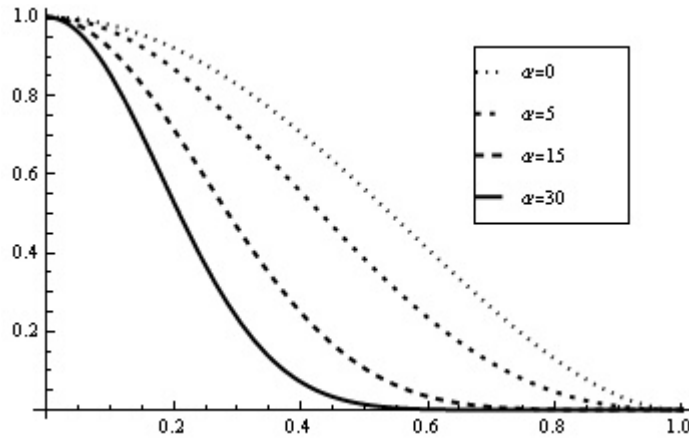
$$b_{m,a,\alpha}(r) = \begin{cases} \frac{1}{I_m(\alpha)} \left[\sqrt{1 - (r/a)^2} \right]^m I_m \left[\alpha \sqrt{1 - (r/a)^2} \right], & 0 \leq r \leq a \\ 0, & r > a, \end{cases} \quad (3.1)$$

jossa I_m on ensimmäisen tyyppin modifioitu Besselin funktio astetta m , $a > \Delta$ on kuulan säde ja α on kuulan muotoa kontrolloiva parametri. Aste m määrittää, montako kertaa derivoituva kuula on rajallaan. Jos $m = 0$, niin kuula ei ole jatkuva kun $r = a$. Jos $m > 0$, niin kuulalla on $m - 1$ jatkuvaa derivaattaa kun $r = a$. Tässä tutkielmassa oletetaan että $m = 2$, jollei muuta mainita. Nyt kuulaa merkitään $b_{m,a,\alpha}(r) = b_{a,\alpha}$. Pienet parametrin α arvot tuottavat hitaasti väheneviä kuulia ja suuret parametrin α arvot tuottavat kapeahuipuisia ja pitkähäntäisiä kuulia. Yleisesti parametrien a ja α arvot riippuvat kuvan ja datan piirteistä, mutta voidaan sanoa, että säde tulee valita siten että kuulan puoliarvoveys (Full Width at Half Maximum, FWHM) ei ylitä mittauksen erottelukyvyn puoliarvoveyttä. Jos kuulan FWHM on tätä suu-

rempi, niin kuvamallilla on pienempi näytteenottotaajuus kuin mittauksella, josta laskostumisen takia seuraa yliestimointia kuvan korkean kontrastin alueilla. Yliestimointi vielä korostuu iteraatiomääriä lisättäessä.

Esimerkiksi jos $FWHM_{PET} = u$, niin kuula $b_{a,\alpha}$ valitaan siten, että $FWHM_{blob} < 2a \cdot 0,6 < u$ (kuva 8), missä $a = k\Delta$. Jos parametri α on valittu käytännön rekonstruktion kannalta järkevästi, on hilavakio valittava siten, että $\Delta < u/(1,2k)$. Koska vakio k on kiinnitetty kuulaparametrin a arvon kautta, niin hilavakion Δ ja skannerin geometriasta tulevan vakion u välillä on lineaarinen riippuvuus. Kyseessä oli nyt varsin suoraviivainen tarkastelu, mutta se osoittaa, että loppujen lopuksi kuulien ja kuulahilan parametrit on suhteutettava skannerin geometriaan.

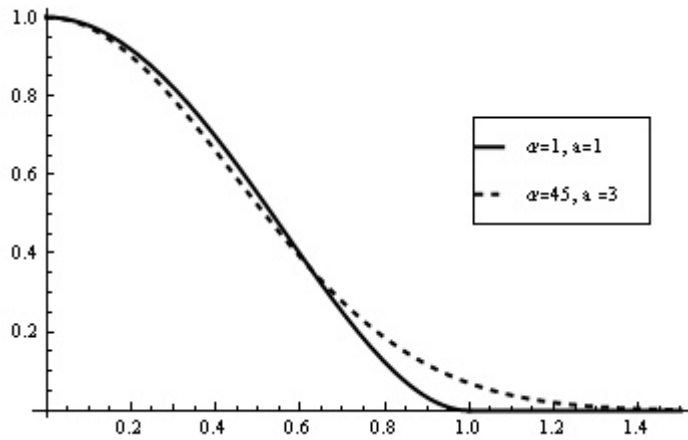
Kuulien profiilikäyriä on esitetty kuvassa 8. Jokaisessa käyrässä on säde $a = 1$, ja parametrin α arvot ovat yhtenäisessä viivassa 30 ja seuraavissa viivoissa järjestyksessä 15, 5 ja 0. Kuulien profiilikäyrät on mahdollista so-



Kuva 8: Kuulien arvoja eri parametrin α arvoilla.

vittaa saman kaltaisiksi useilla eri parametrien a ja α arvoilla. Kuvassa 9 on esitetty kaksi erilaista profiilikäyrää. Yhtenäisellä viivalla on piirretty $b_{1,1}(r)$ ja katkoviivalla $b_{3,45}(r)$. Parametrin $\alpha = 45$ ei ole rekonstruktion kannalta kovinkaan realistinen, mutta se antaa esimerkin eri parametristen kuulien profiilien samankaltaisuudesta.

Piirrettäessä kuulan arvo kolmiulotteisena saadaan pallo, jonka läpinäky-



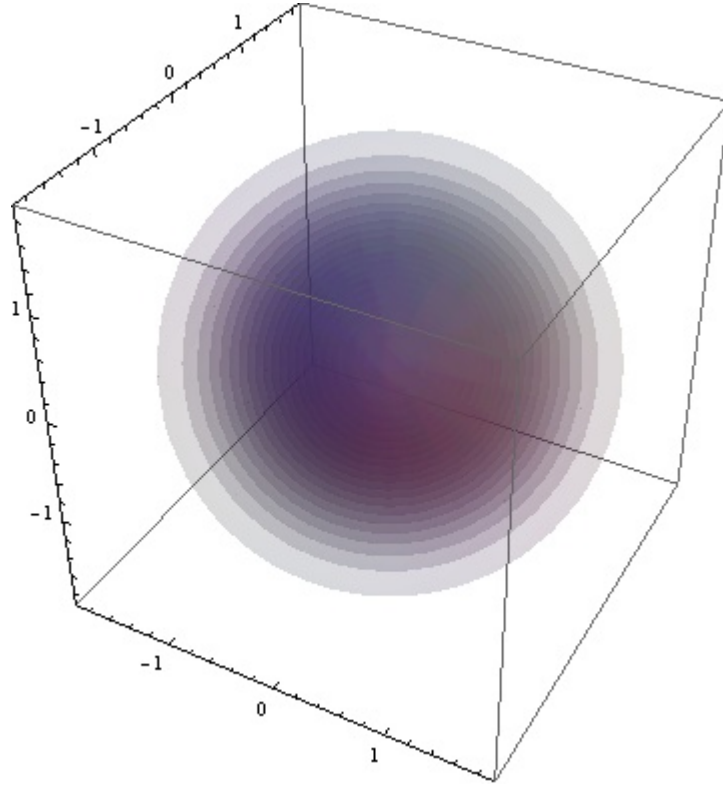
Kuva 9: Kuulien profiilikäyrät eri parametrien a ja α arvoilla.

vyys kasvaa lähestyttäessä reunoja. Tämä on kuvattu kuvassa 10. Kuvassa 11 on havainnollistettu kolmiulotteisen kuulan poikkileikkaus muotoparametrin arvoilla $\alpha = 0, 3, 9$ ja säteen arvolla $a = 1$. Valkoinen ympyrä kuvaa kuulan reunaa, jossa $r = a = 1$.

Rekonstruoitaessa kuvaa saadaan kuulan säteelle a loogisesti sekä ala-, että yläraja. Yläraja saatiin edellä puoliarvoveyden kautta, mutta myös tätä pienemmät säteen arvot voivat olla käytännöllisesti liian suuria tarvittavan laskentatehon kasvaessa. Mitä suurempi säde a on, sitä enemmän kuulat menevät päällekkäin ja sitä enemmän laskentaa jokaisen projisiosuoran päivittämiseen tarvitaan.

Alaraja saadaan aiemmassa kappaleessa kuvatuista projisiosuoria ympäröivistä sylintereistä. Mikäli säde a on liian pieni, eivät projisiosuoria ympäröivät sylinterit äärellisen määränsä takia riitä kattamaan koko kuva-avaruutta. Tässä tapauksessa osa kuulista ei profiloitu yhteenkään projisiosuoraan ja kuva rakennetaan liian tiheään kuulahilaan. Tällöin kuvan käyttäytyminen voi olla hyvinkin epäsäännöllistä ja vaikeasti ennustettavaa.

Edellisessä kappaleessa esitellyn jalanjälkitaulun viivaintegraalit yli viivan, jonka kohtisuora etäisyys kuulan keskipisteestä on s , saadaan kaavasta (3.1)



Kuva 10: Kuula avaruudessa \mathbb{R}^3 .

seuraavasti [10]:

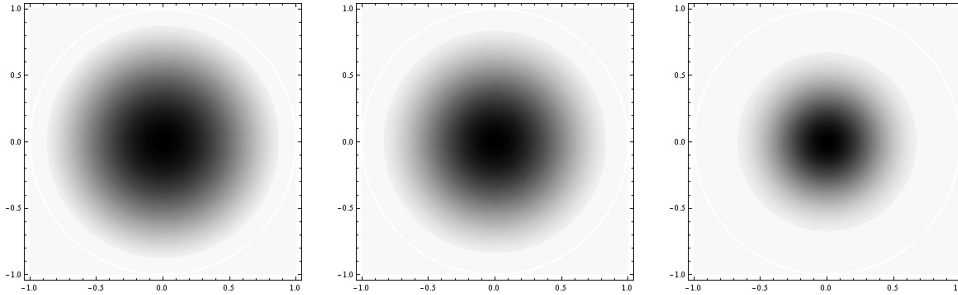
$$\begin{aligned}
 & p_{m,a,\alpha}(s) \\
 &= \frac{a}{I_m[\alpha]} \left(\frac{2\pi}{\alpha} \right)^{1/2} \left[\sqrt{1 - (s/a)^2} \right]^{m+1/2} I_{m+1/2} \left[\alpha \sqrt{1 - (s/a)^2} \right]. \quad (3.2)
 \end{aligned}$$

Kaavassa esiintyy nyt kaksi eriasteista Besselin funktiota. Käyttämällä modifioitua Besselin pallofunktiota $i_n(z)$ voidaan kaavan (3.2) jälkimmäinen Besselin funktio kirjoittaa muodossa:

$$I_{m+1/2} \left[\alpha \sqrt{1 - (s/a)^2} \right] = I_{m+1/2} [h] = \sqrt{\frac{2h}{\pi}} i_m(h), \quad (3.3)$$

missä $h = \alpha \sqrt{1 - (s/a)^2}$. Useimmiten käytetty asteen m arvo on 2, jolloin kaava (3.3) voidaan kirjoittaa hyperbelifunktioiden avulla

$$\sqrt{\frac{2h}{\pi}} i_2(h) = \sqrt{\frac{2x}{\pi}} \frac{(h^2 + 3) \sinh h - 3h \cosh h}{h^3}. \quad (3.4)$$



Kuva 11: Kuulan poikkileikkaus parametrien arvoilla $a = 1$ ja $\alpha = 0$ (vasen), $\alpha = 3$ (keski), sekä $\alpha = 9$ (oikea). Parametrin α kasvaessa kuulasta tulee terävähuippuisempi ja kuulan arvo putoaa nopeammin kohti nollaa.

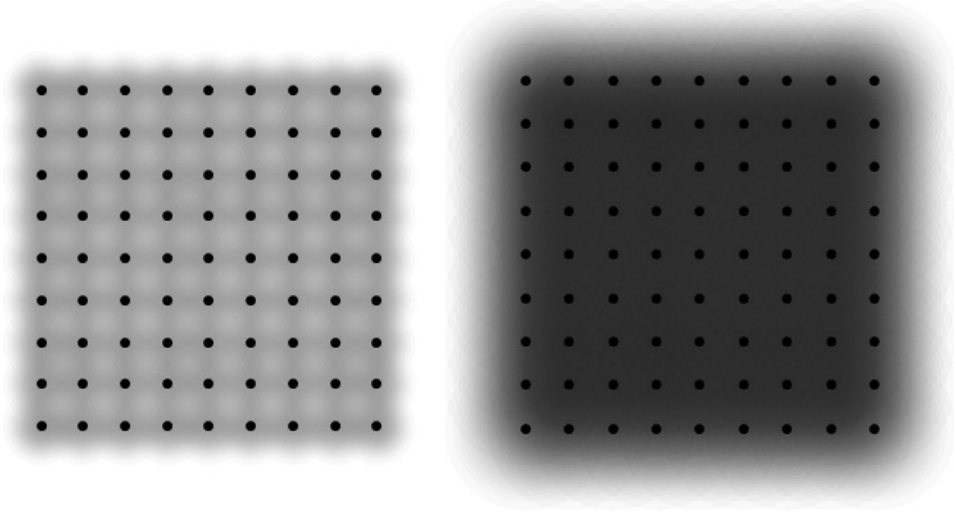
Tämä muoto vähentää Besselin funktiossa esiintyvien äärettömän pituisten summien laskennan tarvetta, jolloin laskennan katkaisusta johtuva virhe pienenee. Kuitenkin arvon h lähestyessä nollaa kaavan (3.4) käyttäytyminen muuttuu hyvinkin herkäksi nimittäjässä olevan muuttujan h takia. Esimerkiksi MATLAB laskee kuitenkin Besselin funktioita hyvinkin nopeasti, tuhansia sekunnissa, joten tällaiselle muunnokselle ei usein pitäisi olla tarvetta.

Nyt kun on määritelty kuulan parametrit esitetään kuva 12, jota vastaa kuvan 4 tilannetta, kun $\alpha = 3$. Kuvista nähdään, että kun kuvassa 12 (vasen) säde $a = 1$, niin kuva-avaruus ei täyty tasaisesti, vaan kuulien keskipisteiden väliin jää vaaleampia alueita. Kuvassa 12 (oikea) säde $a = 2$ ja kuva-avaruus täyttyy tasaisemmin kuulista.

3.3 Fourier-ehto

Parametrit a ja α voi valita myös tarkastelemalla kuulan Fourier-muunnosta. Yleisesti funktion $g(x)$ Fourier-muunnos saadaan kaavasta

$$\hat{g}(Y) = \int_{\mathbb{R}^n} g(x) e^{-i2\pi Y x} d\mathbf{x}, \quad (3.5)$$



Kuva 12: Osa kuulahlilasta säteillä $a = 1$ (vasen) ja $a = 2$ (oikea) ja paramet-
rin arvolla $\alpha = 3$. Vasemmasta kuvasta nähdään kuulien välissä olevat liian
pienestä säteestä johtuvat vaaleammat alueet.

missä Y on taajuusmuuttuja. Kuulan Fourier-muunnokselle saadaan seuraava
kaava [10]:

$$\hat{b}_{n,m,a,\alpha}(R) = \begin{cases} \frac{(2\pi a^2)^{n/2} \alpha^m}{I_m[\alpha]} \frac{I_{n/2+m} \left[\sqrt{\alpha^2 - (2\pi a R)^2} \right]}{\left[\sqrt{\alpha^2 - (2\pi a R)^2} \right]^{n/2+m}}, & \text{kun } 2\pi a R < \alpha, \\ \frac{(2\pi a^2)^{n/2} \alpha^m}{I_m[\alpha]} \frac{J_{n/2+m} \left[\sqrt{(2\pi a R)^2 - \alpha^2} \right]}{\left[\sqrt{(2\pi a R)^2 - \alpha^2} \right]^{n/2+m}}, & \text{kun } 2\pi a R > \alpha, \\ \infty, & \text{kun } 2\pi a R = \alpha, \end{cases} \quad (3.6)$$

missä $J_p(x)$ on ensimmäisen tyyppin Besselin funktio astetta p . Yleensä on
kiinnitetty $n = 3$ ja $m = 2$, jolloin kaava (3.6) muuttuu muotoon

$$\hat{b}_{a,\alpha}(R) = \begin{cases} \frac{(\sqrt{2\pi a^2})^3 \alpha^2}{I_2[\alpha]} \frac{I_{7/2} \left[\sqrt{\alpha^2 - (2\pi a R)^2} \right]}{\left[\sqrt{\alpha^2 - (2\pi a R)^2} \right]^{7/2}}, & \text{kun } 2\pi a R < \alpha, \\ \frac{(\sqrt{2\pi a^2})^3 \alpha^2}{I_2[\alpha]} \frac{J_{7/2} \left[\sqrt{(2\pi a R)^2 - \alpha^2} \right]}{\left[\sqrt{(2\pi a R)^2 - \alpha^2} \right]^{7/2}}, & \text{kun } 2\pi a R > \alpha, \\ \infty, & \text{kun } 2\pi a R = \alpha. \end{cases} \quad (3.7)$$

Parametrien a ja α valintaa ohjaava ehto saadaan tarkastelemalla tark-
kuutta, jolla vakiofunktio $f(x, y, z) = 1$ voidaan kuvata kuulien avulla. Eli

yhtälössä (2.1) sijoitetaan $X_j = 1$ kaikilla $j = 1, \dots, m$, sekä sijoitetaan siihen yhtälö (2.2), jolloin saadaan kuvalle yhtälö

$$\bar{f}(x, y, z) = \sum_{j=1}^m b \left(\sqrt{(x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2} \right). \quad (3.8)$$

Tarkoitus olisi nyt saada $\bar{f}(x, y, z) \approx 1$ kaikissa kuulahilan sisäpisteissä. Johdatellaan tätä yksiulotteisessa ja rajoittamattomassa tapauksessa, jolloin yhtälö (3.8) tiivistyy muotoon

$$\bar{f}(x) = \Delta \sum_{j=-\infty}^{\infty} b(x - \Delta j). \quad (3.9)$$

Kaavasta (3.5) saadaan nyt yksiulotteinen yksinkertaistus kuulun Fourier-muunnoksen yleiselle muodolle.

$$\hat{b}(Y) = \int_{-\infty}^{\infty} b(x) e^{-i2\pi Y x} dx. \quad (3.10)$$

Käyttämällä yhtälöön (3.9) Poissonin summauskaavaa [4], saadaan yhtälö

$$\Delta \sum_{j=-\infty}^{\infty} b(x - \Delta j) = \sum_{k=-\infty}^{\infty} \hat{b}(k/\Delta) e^{-i2\pi(k/\Delta)x}.$$

Erotetaan nyt tilanne $k = 0$ oikean puolen summasta ja sovelletaan siihen yhtälöä (3.10), mistä saadaan

$$\begin{aligned} \Delta \sum_{j=-\infty}^{\infty} b(x - \Delta j) &= \sum_{k \neq 0} \hat{b}(k/\Delta) e^{-i2\pi(k/\Delta)x} + \hat{b}(0) e^0 \\ &= \sum_{k \neq 0} \hat{b}(k/\Delta) e^{-i2\pi(k/\Delta)x} + \int_{-\infty}^{\infty} b(x) dx. \end{aligned}$$

Jos nyt on

$$\begin{cases} \hat{b}(0) = 1, \\ \hat{b}(k/\Delta) = 0, \quad \text{kaikille } k \neq 0, \end{cases} \quad (3.11)$$

niin kaavan (3.9) arvo on 1 kaikilla $x \in \mathbb{R}$. Koska Δ on tasavälisen hilan hilavakio, niin $1/\Delta$ on hilan taajuus, eli resoluutio jolla kuva voidaan esittää. Nyt ehto (3.11) vaatii, että kantafunktion Fourier-muunnos menee nolnaan kaikilla hilan taajuuden monikerroilla. Signaalinkäsittelyn näkökulmasta tämä tarkoittaa ettei Fourier-muunnoksen nollataajuustermiin tapahdu

laskostumista. Kuulat eivät täysin täytä ehtoa, mutta sopivalla parametrien valinnalla päästään halutun lähelle indeksin k kasvaessa.

Edellisestä esimerkistä voidaan nyt palata kolmiulotteiseen tilanteeseen. Nyt yksiulotteisen tilanteen indeksin k sijasta tulee käyttää kolmeen kokonaislukuun perustuvaa etäisyysmittaa, $\sqrt{i^2 + j^2 + k^2}$, jossa indeksit $i, j, k \in \mathbb{N}$ ja ne vastaavat karteesisen koordinaatiston pääakselien suuntia. Nyt ehdon (3.11) sijasta vaaditaan hieman löysempiä ehtoja

$$\begin{cases} \hat{b}(0) = 1, \\ \hat{b}(1/\Delta) = 0, \\ \hat{b}(\sqrt{i^2 + j^2 + k^2}/\Delta) \approx 0 \quad \text{kaikille } i + j + k > 1. \end{cases} \quad (3.12)$$

Yhtälön (3.7) alemmasta muodosta nähdään, että Fourier-muunnos menee nollassa, kun Besselin funktio $J_{7/2} \left[\sqrt{(2\pi a R)^2 - \alpha^2} \right] = 0$. Valitaan

$$R_0(l) = \frac{1}{2\pi a} \sqrt{\alpha^2 + u(l)^2}, \quad l \in \mathbb{Z}_+,$$

missä $u(l)$ on Besselin funktion $J_{7/2}(x)$ nollakohtien funktio. Funktion $u(l)$ arvot saadaan laskettua yhtälöitä (3.3) ja (3.4) muistuttavan Besselin pallofunktioita käyttävän identiteetin

$$J_{7/2}(x) = \sqrt{\frac{2x}{\pi}} j_3 = \sqrt{\frac{2x}{\pi}} \left(\left(\frac{15}{x^3} - \frac{6}{x} \right) \frac{\sin(x)}{x} - \left(\frac{15}{x^2} - 1 \right) \frac{\cos(x)}{x} \right) = 0$$

avulla. Tällä valinnalla saadaan $J_{7/2} \left[\sqrt{(2\pi a R_0)^2 - \alpha^2} \right] = J_{7/2}[u] = 0$. Taulukossa 1 on taulukoitu muutamia ensimmäisiä funktion $u(l)$ arvoja.

Ehtojen (3.12) toinen ehto saadaan voimaan valitsemalla parametrit a ja α siten, että $R_0 = 1/\Delta$. Valitsemalla parametrin α arvo tarpeeksi suureksi kuula heikkenee nopeasti puoliarvoveyden jälkeen, jolloin hilan jaksoista johtuvat säännölliset toistot Fourier-muunnoksen eri taajuuskomponentteihin minimoituvat. Sopiville parametrien arvoille voidaan johtaa parametrie-

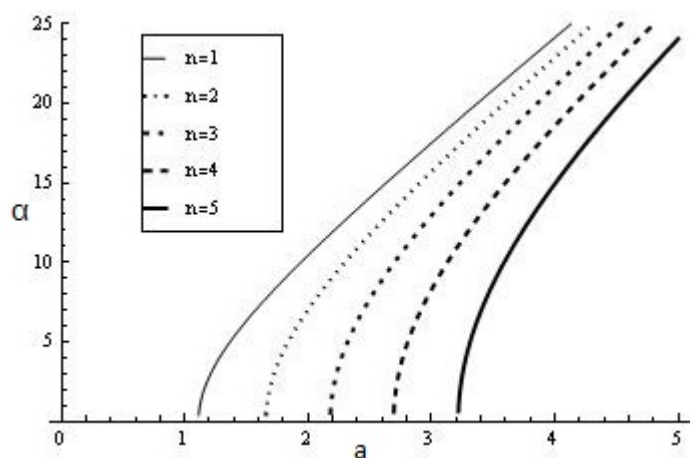
Taulukko 1: Fourier-muunnoksen nollakohtia

l	$u(l)$	R_0
1	6,988	$\frac{1}{2\pi a} \sqrt{\alpha^2 + 48,8311}$
2	10,417	$\frac{1}{2\pi a} \sqrt{\alpha^2 + 108,516}$
3	13,698	$\frac{1}{2\pi a} \sqrt{\alpha^2 + 187,635}$
4	16,923	$\frac{1}{2\pi a} \sqrt{\alpha^2 + 286,408}$
5	20,121	$\frac{1}{2\pi a} \sqrt{\alpha^2 + 404,887}$
6	23,304	$\frac{1}{2\pi a} \sqrt{\alpha^2 + 543,087}$
7	26,476	$\frac{1}{2\pi a} \sqrt{\alpha^2 + 701,019}$
8	29,642	$\frac{1}{2\pi a} \sqrt{\alpha^2 + 878,684}$
9	32,803	$\frac{1}{2\pi a} \sqrt{\alpha^2 + 1076,08}$
10	35,961	$\frac{1}{2\pi a} \sqrt{\alpha^2 + 1293,22}$
n	$u(n)$	$\frac{1}{2\pi a} \sqrt{\alpha^2 + u(n)^2}$

sitys. Käytetään taulukon 1 viimeistä riviä, jolloin saadaan

$$\begin{aligned}
 R_0 &= \frac{1}{2\pi a} \sqrt{\alpha^2 + u(n)^2} = 1/\Delta \\
 &\Leftrightarrow \Delta \sqrt{\alpha^2 + u(n)^2} = 2\pi a \\
 &\Leftrightarrow \Delta^2 (\alpha^2 + u(n)^2) = 4\pi^2 a^2 \\
 &\Leftrightarrow \Delta^2 \alpha^2 - 4\pi^2 a^2 = -\Delta^2 u(n)^2 \\
 &\Leftrightarrow \frac{\alpha^2}{-u(n)^2} + \frac{4\pi^2 a^2}{\Delta^2 u(n)^2} = \frac{(a/\Delta)^2}{(u(n)/2\pi)^2} - \frac{\alpha^2}{u(n)^2} = 1,
 \end{aligned} \tag{3.13}$$

joka on hyperbelin yhtälö. Yhtälöstä hyödynnetään luonnollisesti vain sitä haaraa, jossa a , sekä α ovat positiivisia. Kuvassa 13 on piirretty hyperbelin positiivisia haaroja eri parametrin $u(n)$ arvoille. Kuvan käyrät vastaavat kekeellisesti artikkelissa [17] saatuja hyviä "parametrilaaksoja". Artikkelissa [17] parametrilaaksot saatiin empiirisellä testauksella, mutta nyt johdettiin analyttiset käyrät, joista saadaan parametreja halutulla tarkkuudella. Edellä saadut hyperbelit vastaavat tutkielman pohjalla olevaan kysymykseen siitä, millaisia arvoja kuulien parametreiksi pitää laittaa, jotta kuva rekonstruoi-tuu kunnolla. Nyt on mahdollista suorittaa simulaatiot, koska järkeviä parametrijohdistelmia voidaan generoida tarvittava määrä.



Kuva 13: Parametrien a ja α yhteyttä kuvaavia hyperbelejä, kun $n \in \{1, \dots, 5\}$.

4 Rekonstruktion simulointi

4.1 Johdanto

Kuulien ja RAMLA-algoritmin käytännön toteutus ohjelmoitiin MATLAB-ohjelmointikielellä. Testattavana kohteena käytettiin kaksiulotteisen välille $[0, 5]$ skaalattua Shepp-Loganin testikappaleen muunnosta, jossa ellipsit määriteltiin kolmiulotteisina ellipsoideina (kuva 14). Ellipsoideille laskettiin analyyttisesti viivaintegraalin arvot mielivaltaisille avaruuden \mathbb{R}^3 suorille. Analyyttisestä kaavasta pystyttiin laskemaan kaikille projisiosuorille mitatut arvot RAMLA-algoritmin suorittamiseksi.

RAMLA-algoritmia laskettaessa käytetään kaavaa (2.26), kuitenkin niin, että jokaisen projisiosuoran kohdalla päivitetään vain ne kuulat, jotka saadaan jalanjälkitaulukosta tarvittavien indeksisiirtojen jälkeen. Muiden kuulien arvo pysyisi samana, sillä projisiosuora ei niitä leikkaa.

4.2 Materiaali ja menetelmät

Simulaatioiden pohjalla käytetään aikaisemmissa luvuissa johdettua teoriaa. RAMLA-algoritmia käytetään lähes sellaisenaan ja projisiosuorista jätetään



Kuva 14: Shepp-Loganin testikappale. Kuvakoko rekonstruktion aikana 176x176. Reunan arvo on 5, kallon sisustan 1, kallon sisällä olevien kirkkaiden kohtien 1.5 ja mustien alueiden 0.

laitimmainen neljännes detektorin kummaltakin puolelta huomiotta, kuvan 2 tapaan. Yhtälöstä (3.13) saadaan yhteys kuulan säteen ja parametrin α välille ja simuloinnissa käytetyt parametrit saadaan kuvan 13 eri käyriltä. RAMLA-algoritmin relaksaatioparametri λ eri iteraatioille lasketaan kaavasta

$$\lambda_k = \lambda_0/k, \quad (4.1)$$

missä λ_0 on muuteltava parametri. Yhtälön (4.1) määrittelemä parametri λ toteuttaa yhtälön (2.23), sillä parametrin λ_k pohjana oleva harmoninen sarja hajaantuu. Käytännössä iteraatioiden määrä on pieni, joten parametrin λ muutosnopeudella ei ole suurta laskennallista merkitystä.

4.2.1 Ellipsoidin viivaintegraali

Ellipsoidin viivaintegraali projisiosuoraa pitkin on ellipsoidin ulkopinnan ja projisiosuoran leikkauspisteiden välimatka avaruudessa \mathbb{R}^3 . Määritellään pro-

jisiosuora lähtödetektorin paikkavektorin ja suoran suuntavektorin avulla

$$\begin{cases} x = x_0 + r\alpha_x \\ y = y_0 + r\alpha_y \\ z = z_0 + r\alpha_z, \end{cases} \quad (4.2)$$

missä $r \in \mathbb{R}$, vektori $\{\alpha_x, \alpha_y, \alpha_z\}$ on suoran yksikkösuuntavektori ja vektori $\{x_0, y_0, z_0\}$ on lähtödetektorin paikkavektori. Määritellään yleinen koordinaattiakselien suuntainen ellipsoidi keskipisteensä ja puoliakseliensä pituu- den avulla

$$\frac{(x - d_x)^2}{a^2} + \frac{(y - d_y)^2}{b^2} + \frac{(z - d_z)^2}{c^2} = 1, \quad (4.3)$$

missä $\{a, b, c\}$ ovat puoliakselien pituudet ja $\{d_x, d_y, d_z\}$ ellipsoidin keskipiste. Kirjoittamalla yhtälö (4.3) matriisien ja vektorien avulla saadaan

$$(x - d_x, y - d_y, z - d_z) \begin{pmatrix} \frac{1}{a^2} & 0 & 0 \\ 0 & \frac{1}{b^2} & 0 \\ 0 & 0 & \frac{1}{c^2} \end{pmatrix} (x - d_x, y - d_y, z - d_z) = 1,$$

josta voidaan tiivistää muoto

$$(\mathbf{X} - \mathbf{d})M(\mathbf{X} - \mathbf{d}) = 1. \quad (4.4)$$

Tämän tutkielman tapauksessa tarvitaan kuitenkin ellipsoideja, joita on kierretty z -akselin ympäri. Tätä varten kerrotaan kaavan (4.4) matriisi M vasemmalta rotaatiomatriisilla

$$R = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

ja oikealta rotaatiomatriisin transpoosilla R^T . Nyt kaava (4.4) muuttuu muotoon

$$(\mathbf{X} - \mathbf{d}) \begin{pmatrix} \frac{\cos^2 \theta}{a^2} + \frac{\sin^2 \theta}{b^2} & \frac{\cos \theta \sin \theta}{b} - \frac{\cos \theta \sin \theta}{a} & 0 \\ \frac{\cos \theta \sin \theta}{b} - \frac{\cos \theta \sin \theta}{a} & \frac{\cos^2 \theta}{b^2} + \frac{\sin^2 \theta}{a^2} & 0 \\ 0 & 0 & \frac{1}{c^2} \end{pmatrix} (\mathbf{X} - \mathbf{d}) = 1,$$

mistä saadaan matriisit auki kertomalla

$$\begin{aligned} & \frac{(x - d_x)^2 \cos^2 \theta}{a^2} - \frac{2(x - d_x)(y - d_y) \sin \theta \cos \theta}{a^2} + \frac{(y - d_y)^2 \sin^2 \theta}{a^2} \\ & + \frac{(x - d_x)^2 \sin^2 \theta}{b^2} + \frac{2(x - d_x)(y - d_y) \sin \theta \cos \theta}{b^2} + \frac{(y - d_y)^2 \cos^2 \theta}{b^2} \\ & + \frac{(z - d_z)^2}{c^2} = 1. \end{aligned} \quad (4.5)$$

Huomataan yhtälön (4.5) neliömuoto ja sijoitetaan siihen vielä yhtälöt (4.2), jolloin saadaan yhtälö

$$\begin{aligned} & \frac{1}{a^2} ((x_0 + r\alpha_x - d_x) \cos \theta - (y_0 + r\alpha_y - d_y) \sin \theta)^2 + \frac{(z_0 + r\alpha_z - d_z)^2}{c^2} \\ & + \frac{1}{b^2} ((x_0 + r\alpha_x - d_x) \sin \theta + (y_0 + r\alpha_y - d_y) \cos \theta)^2 = 1. \end{aligned}$$

Kerrotaan termit auki ja järjestellään yhtälö muuttujan r suhteen toisen asteen yhtälöksi

$$Ar^2 + Br + C = 0, \quad (4.6)$$

missä

$$\begin{aligned} A &= \frac{1}{a^2} (\alpha_x \cos \theta - \alpha_y \sin \theta)^2 + \frac{1}{b^2} (\alpha_x \sin \theta + \alpha_y \cos \theta)^2 + \frac{\alpha_z^2}{c^2}, \\ B &= \frac{2}{a^2} (\alpha_x \cos \theta - \alpha_y \sin \theta) ((x_0 - d_x) \cos \theta - (y_0 - d_y) \sin \theta) \\ & \quad + \frac{2}{b} (\alpha_x \sin \theta + \alpha_y \cos \theta) ((x_0 - d_x) \sin \theta + (y_0 - d_y) \cos \theta) + \frac{2\alpha_z(z_0 - d_z)}{c^2}, \\ C &= \frac{1}{a^2} ((x_0 - d_x) \cos \theta - (y_0 - d_y) \sin \theta)^2 \\ & \quad + \frac{1}{b^2} ((x_0 - d_x) \sin \theta + (y_0 - d_y) \cos \theta)^2 + \frac{(z_0 - d_z)^2}{c^2} - 1. \end{aligned}$$

Mikäli toisen asteen yhtälöllä on kaksi reaalilukuratkaisua, eli diskriminantti $D = B^2 - 4AC > 0$, niin projisiosuora leikkaa ellipsoidin kahdessa eri pisteessä kulkien siis näiden pisteiden välisen janan J ellipsoidin sisällä. Koska projisiosuoran suuntavektori $\{\alpha_x, \alpha_y, \alpha_z\}$ on samalla yksikkövektori, niin janan J pituus on sama kuin yhtälön (4.6) juurien erotus, eli

$$\|J\| = |r_1 - r_2| = \left| \frac{-B + \sqrt{D}}{2A} - \frac{-B - \sqrt{D}}{2A} \right| = \frac{2\sqrt{D}}{|2A|} = \frac{\sqrt{D}}{A}.$$

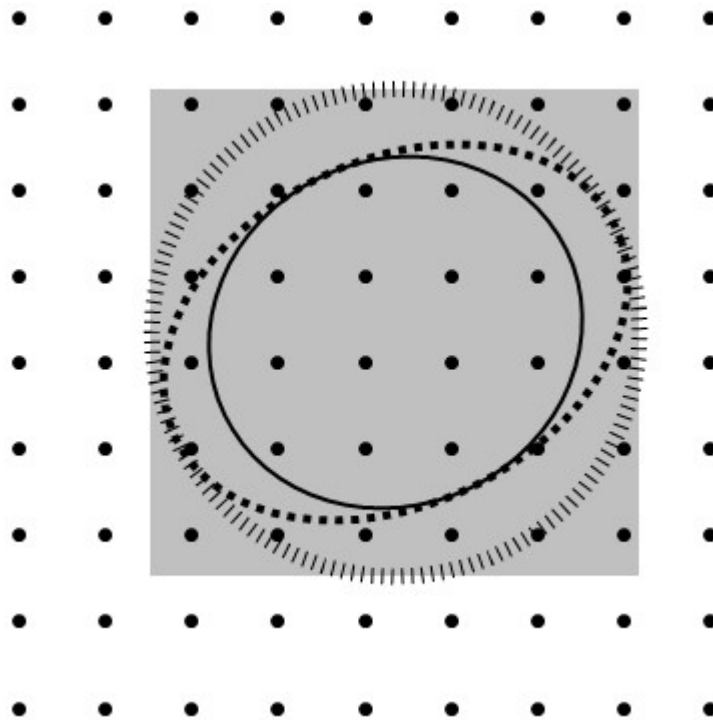
Viimeinen yhtäsuuruus seuraa ominaisuudesta $A > 0$. Laskemalla jokaisen projisiosuoran ja ellipsoidin leikkauspisteet saadaan kaikki projisiomatriisin alkiot simulaatiota varten.

4.2.2 Projisiosuoran ja kuulahilan leikkauspinta

Luvussa 3.1 saatua ellipsiehtoa voidaan nyt muokata laskennan helpottamiseksi, mistä on piirretty esimerkki kuvassa 15. Kuvassa on piirrettynä kuulahilan taso, jossa on yhtenäisellä viivalla piirretty ellipsi A . Ellipsi A on nyt yksi mahdollinen projisiosuoraa ympäröivän sylinterin ja tason leikkauspinta. Ellipsin A laskemiseen tarvitaan paljon suoritinaikaa, minkä lisäksi myös sen tarkistamiseen, onko kuula ellipsin sisällä vai ei kuulu huomattavasti laskentakapasiteettia. Onkin laskennallisesti tarkoituksenmukaista laajentaa analyttistä ellipsiä isommaksi, vaikkakin tällöin joudutaan tarkastamaan erikseen jokainen laajennettuun alueeseen kuuluva kuula. Tällöin vaadittu pisteen etäisyys suorasta laskutoimitus on kuitenkin suhteellisen nopea vektorioperaatio.

Relaksoidaan ellipsiä A ensin tarkastelemalla sylinterin ja tason normaalivektorin η välistä kulmaa τ . Mikäli kulma on 0, niin ellipsi A on ympyrä, jonka säde on kuulan säde a . Jos kulmaa kasvatetaan, ellipsin pidempi pääakseli alkaa pidentyä, pienemmän pysyessä säteen a pituisena. Jos kulma τ kasvaa suuremmaksi kuin $\pi/4$, niin projisiosuoran suuntavektori (θ, ϕ) olisi lähempänä jonkin toisen koordinaattisuunnan määrittämää tasoa. Näin saadaan kulman τ ylärajaksi $\pi/4$. Tätä vastaava ellipsi B on piirretty kuvaan 15 katkoviivalla. Ellipsin B pääakselin pituus on $2a/\sin(\pi/4)$. Nyt ei tarvitse enää ottaa huomioon kulman τ vaikutusta.

Toinen relaksoitava ominaisuus on ellipsin B rotaatio kuulahilan tason suhteen. Kulma ω on 0, kun ellipsin B pääakseli on tason x -akselin suuntainen ja $\omega = \pi/2$ kun pääakseli on y -akselin suuntainen. Koska ellipsi on symmetrinen, niin $\omega \in [0, \pi)$. Relaksoidaan ellipsiä B piirtämällä sen ympärille ympyrä C , jonka halkaisija on ellipsin B halkaisija, eli $2a/\sin(\pi/4)$. Kulman ω laskeminen on työläs vektori- ja matriisiongelmaksi, joten siirryttäessä ympyrään C säästetään huomattavasti laskenta-aikaa.



Kuva 15: Ellipsin laajentaminen neliöksi. Keskimmäinen ellipsi on varsinainen leikkausellipsi. Tätä ympäröi suurimman mahdollisen kohtaamiskulman mukaan laskettu ellipsi. Relaksoimalla saadaan ympyrä, joka on vielä lopuksi neliöity.

Ympyrään C kuulumisen on välttämätön ehto sille, että kuulan keskipiste kuuluu myös ellipsiin A . Riittää siis laskea jokaisen tason kuulan etäisyys projisiosuoran ja tason leikkauspisteestä P , ja mikäli kuulan keskipiste kuuluu ympyrään C laskea tämän jälkeen kuulan keskipisteen etäisyys suorasta. Etäisyyden leikkauspisteestä laskeminen vaatii neliöönkorotus- ja neliöjuurioperaatioita, joten on kyseenalaista kannattaako siihen uhrata aikaa, vai pitäisikö laskea suoraan pisteen etäisyys suorasta. Jos nyt relaksoidaan ympyrä C neliöksi D , joka on varjostettu kuvaan 15 harmaalla, niin saadaan välttämättömän ehdon laskennan vaatimaa laskenta-aikaa vähennettyä huomattavasti. Tarkasteltaessa kuuluuko kuulan keskipiste ympyrään tulee laskea useita kertolaskuja, jotta saadaan etäisyyksien toisia potensseja. Tarkasteltaessa kuuluuko kuulan keskipiste neliöön tarvitsee vain laskea kuulan

koordinaattien ja pisteen P välisten koordinaattien erotusten itseisarvoja, mikä säästää laskenta-aikaa. Mikäli kaikki erotusten itseisarvot ovat pienempiä, kuin neliön D sivun puolikas, eli ympyrän C säde $a/\sin(\pi/4)$, voidaan siirtyä laskemaan varsinainen pisteen etäisyys suorasta.

Käytännön testaus relaksoinnin vaikutuksesta laskentanopeuteen suoritettiin MATLAB-ohjelmistolla. Laskettaessa joka projisiosuoralle ja kuulahilan tasolle ellipsi A , vertaamalla kuulia siihen laskennan nopeus oli noin 1,5 projisiosuoraa sekunnissa. Laskettaessa ainoastaan leikkauspiste P sekä sitä ympäröivä neliö D ja laskemalla neliöön kuuluvien kuulien etäisyys projisiosuorasta oli laskennan nopeus noin 25 projisiosuoraa sekunnissa.

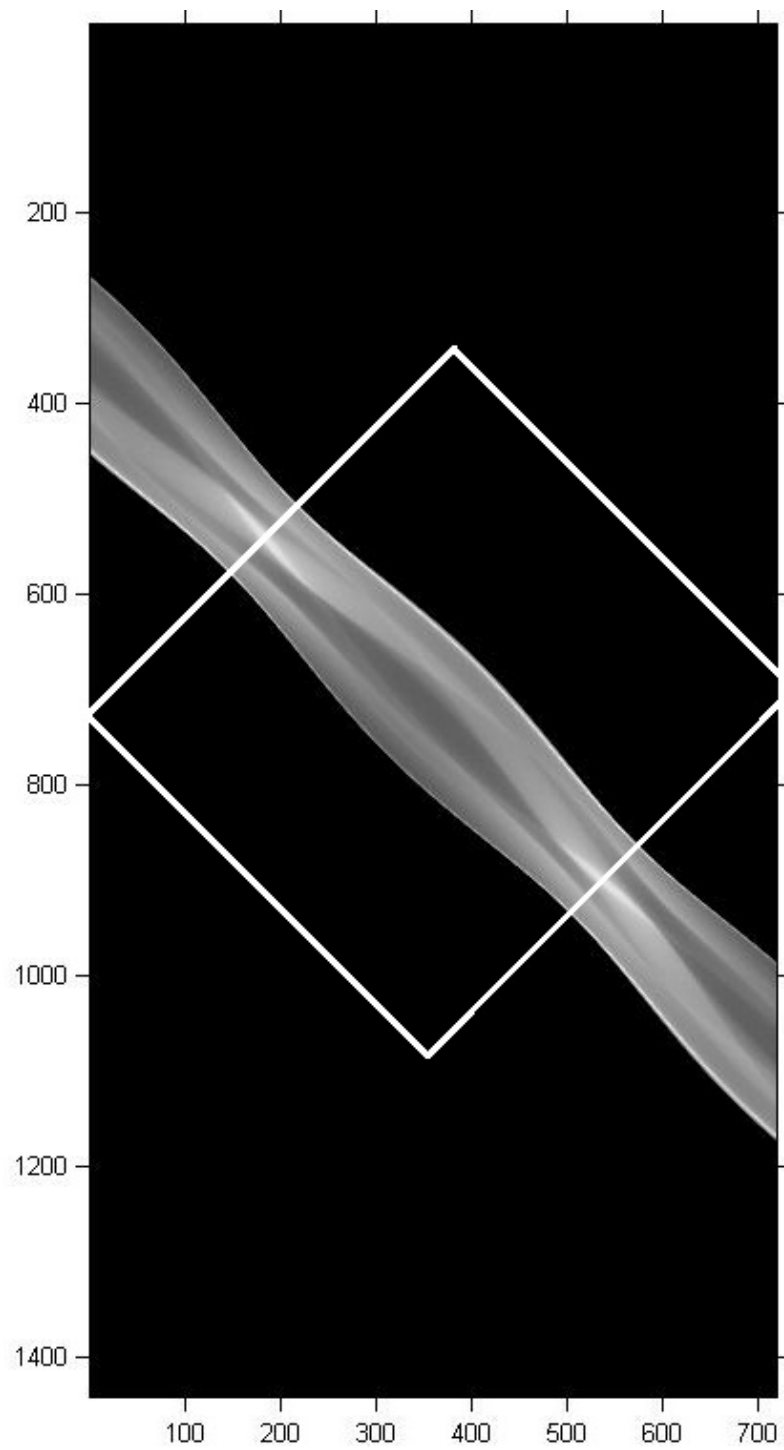
4.2.3 Viivaintegraalien testaus

Ennen varsinaisen RAMLA-algoritmin testausta tarkasteltiin ellipsoidien analyttisen viivaintegraalien laskennan tarkkuutta. Testaus suoritettiin 720 projisiokulman ja 21 projisiotason systeemissä, ja projisiosuorista järjestettiin matriisi siten, että x -akselilla on projisiosuoran lähtödetektorin indeksi $i \in \{1, \dots, 720\}$ ja y -akselilla projisiosuoran loppudetektorin indeksi $j \in \{1, \dots, 1440\}$. Kuvassa 16 on piirretty edellä kuvailtu matriisi kuvan keskimmaiselta leikkeeltä, jossa testikappale on lähinnä kuvan 14 kaksiulotteista tapausta.

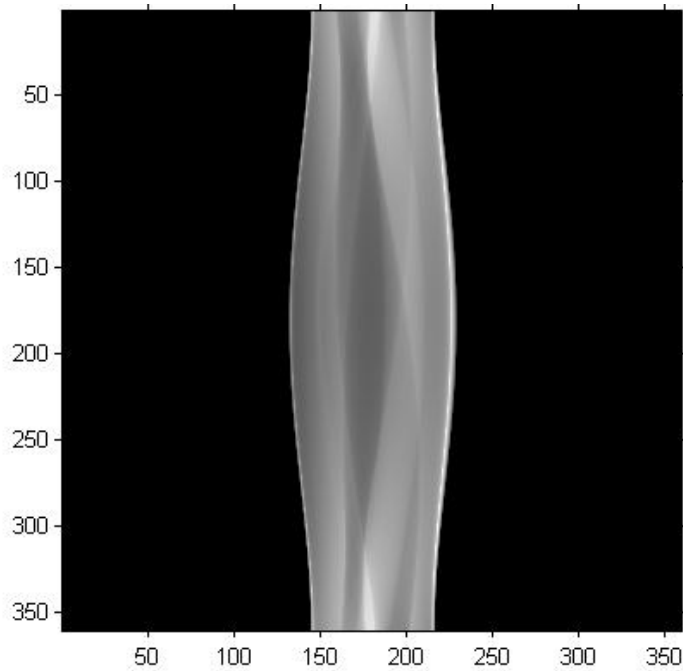
Indeksi j kulkee kaksinkertaisen matkan, koska määritellään $j > i$, jolloin esimerkiksi detektoriympyrän neljänneestä neljännekseen toiseen neljännekseen kulkevan projisiosuoran loppudetektorin indeksiin lisätään yksi täysi kierros, eli 720.

Projisiosuorien matriisista otettiin kuvan 16 valkealla vinoneliöllä reunustettu alue, jota kääntämällä saatiin kaikki saman suuntaiset projisiosuorat samalle riville. Kyseistä esitystä kutsutaan sinogrammiksi ja se on piirretty kuvassa 17.

Sinogrammissa on nyt vain puolet alkuperäisen matriisin kulmista, sillä loput kulmista tarkoittaisivat vain samojen projisiosuorien kulkemista toiseen suuntaan. Kuvan rekonstruointiin käytettiin MATLAB-ohjelmiston valmista funktiota, joka laskee annetusta sinogrammista käänteisen radon-



Kuva 16: Järjestetyt projisiosuorat ja sinogrammiahiota ympäröivä laatikko.



Kuva 17: Sinogrammileike.

muunnoksen. Nyt kuvan 17 projisiosuorat eivät ole kuva-avaruudessa tasavälein, mitä MATLAB pitää oletuksena, vaan projisiosuorat lähenevät toisiaan lähestyttäessä reunoja (kuva 1). Tasavälisyysoletuksesta luopumisella ei kuitenkaan ole kuvan keskiosissa, missä kuvattava kohde sijaitsee, suurtakaan vaikutusta. Kuvassa 18 on esitetty analyttisesti rekonstruoitu kolmiulotteisen Shepp-Loganin keskimäinen leike.

4.2.4 Kuvan geometria ja rekonstruktion yksityiskohtia

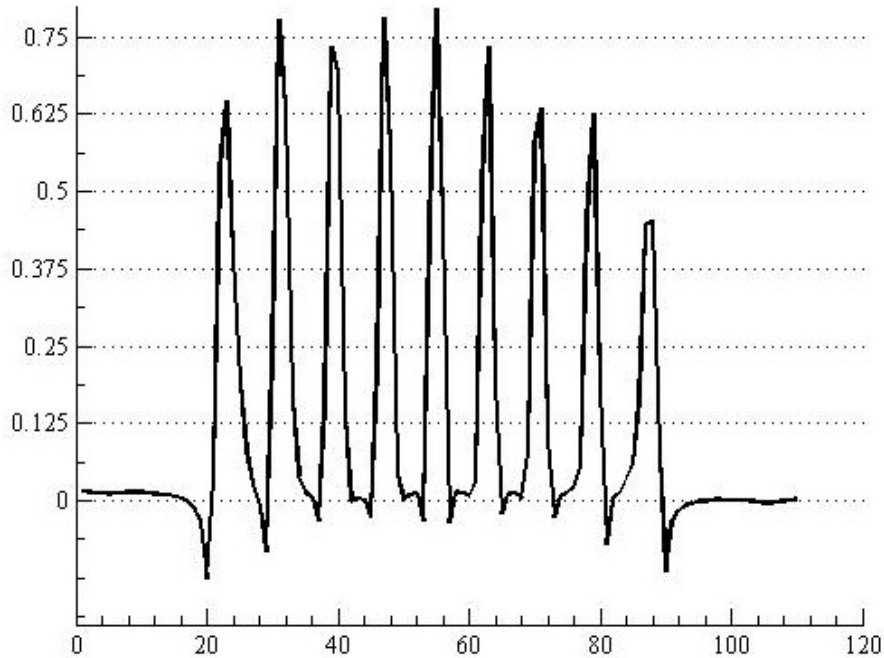
Alustavien kokeilujen jälkeen päädyttiin käyttämään geometriaa, jossa on 316 kulmaa, 49 rinkiä, ja hilavakio $\Delta = 1$ mm. Simulaatiossa ringin säde on 80 mm, ja kuulahilaksi valittiin 97^3 kokoinen BCC-tyyppinen hila (kuva 5) sisältäen yhteensä 1 769 472 kuulaa. Kuvan 2 tapaan laittimaiset 148 projisiosuoraa on poistettu, jolloin kuvattavan alueen säde on maksimissaan 59 mm. Osa tästä alueesta jätetään kuitenkin tyhjäksi, ja tutkimuksen kohteena oleva Shepp-Loganin testikappale mahtuu 44 mm säteisen ympyrän sisään. Kyseisen geometrian puoliarvoväyettä ar-



Kuva 18: Analyytisesti rekonstruoitu Shepp-Loganin testikappale. Projisiokulmien lukumäärä 720.

vioitiin simuloimalla pistemäisten kappaleiden rekonstruktioitumista, minkä tuloksen profiilikäyrästä etsittiin kohdat, joissa profiilikäyrän arvo putoaa puoleen alkuperäisten kappaleiden arvosta. Profiilikäyrä on esitetty kuvassa 19. Profiilikäyrän perusteella saadut puoliarvovälit ovat vasemmalta oikealle $\{1,7690; 1,8960; 1,8840; 1,7050; 1,7480; 1,5290; 1,5830; 1,2000; -\}$. Viimeisen piikin puoliarvoväliä ei voitu määrittää, sillä se aliestimoiuu voimakkaasti, eikä nouse yli vaadittavan arvon 0,5. Puoliarvoveyksiens keskiarvo on 1,6643 mm, ja mediaani 1,7265 mm. Myöhemmissä tarkasteluissa simulaation puoliarvoveyden voidaan katsoa olevan noin 1,7 mm. Puoliarvoveyden laskemiseen käytettiin nyt odotettua puoliarvoa mitatun sijaan, sillä käytetty kuvamatriisi on puoliarvoveyden testaamiseen nähden pieni ja samalla päästään korjaamaan pisteiden ei-infinitesimalisuutta.

Simulaatiossa käytetty rekonstruktioalgoritmi kykenee laskemaan myös renkien välisiä, vinoja projisiosuoria, mutta ne jätettiin tässä tutkielmassa pois ja tyydyttiin kaksikulotteisiin kuviin laskenta-ajan rajoittamiseksi. Lopullisella geometrialla iteraatioon kului pienimmällä säteen arvolla $a = 1,8$ noin neljä tuntia kasvaen tästä aina 30 tuntiin säteen ollessa suurimmillaan



Kuva 19: Rekonstruoitujen pistekappaleiden profilikäyrä. Keskimmäinen huippu sijaitsee kuva-avaruuden keskipisteessä, ja muut tasaisesti sen ympärillä. Alkuperäistä arvoa 1 ei saavuteta yhdessäkään profilikäyrän piikissä, vaan aktiivisuus leviää ympäristöön. Puoliarvovälit on laskettu etsimällä x -akselin arvot profilikäyrän ja suoran $y = 0,5$ leikkauspisteistä.

$a = 4$. Iteraatiomäärän ollessa 5 jokaista taulukon 2 parametrijhdistelmää kohti laskenta-aika oli jo kaksiulotteisessa tapauksessa tuntuva, ja vinojen projisiosuorien lisääminen olisi välittömästi moninkertaistanut tarvittavan ajan. Iteraatiomääräksi valittiin 5, sillä artikkelissa [22] päädyttiin vastaavaan tulokseen iteraatiomäärien arvoja tarkasteltaessa. Kamerassa oletusarvoiset kaksi iteraatiota eivät riitä ja viiden jälkeen muutokset alkavat olla merkityksettömiä. Laskenta suoritettiin pääasiassa PET-keskuksen tietokoneilla ja servereillä.

Projisiosuorien läpikäyntiin RAMLA-algoritmin aikana sovelletaan ennen laskentaa luotuja satunnaisia hajautustauluja, jotka täyttävät täydellisen ja minimaalisen hajautuksen ehdot kuvaten kokonaisluvun välillä $\{1, \dots, q\}$ toi-

seksi tai mahdollisesti samaksi kokonaisluvuksi välille $\{1, \dots, q\}$. Vaihtoehtona testattu projisiosuorien läpikäynti peräkkäisessä järjestyksessä on yksinkertaisempi ja laskennallisesti hieman nopeampi, mutta tuottaa kuviin selkeästi havaittavan puolieron, mikä katsottiin hyvin epäsuotavaksi ominaisuudeksi.

Varsinainen rekonstruktion muodostama kuva on esitettävä kuulien sijaan vokseleina, jolloin yksittäisessä leikkeessä on pikseleitä, mikä mahdollistaa kuvan järkevän tallentamisen ja tarkastelun. Kuulia ei kuitenkaan voi konvertoida suoraan vokseliarvoiksi keskittämällä esitettävän kuvan vokselit kuulien keskipisteisiin ja siirtämällä kuulien rekonstruktiossa saamat arvot suoraan vokseliarvoiksi. Kuulat menevät kuva-avaruudessa päällekkäin, jolloin yhteen kuva-avaruuden pisteeseen vaikuttaa aina useampi kuin yksi kuula. Tämä on otettava huomioon konversiossa kuula-avaruudesta vokseliavaruuteen.

Käytännössä konversio tapahtuu käyttämällä hyväksi kaavaa (2.1), johon on nyt sijoitettu kaava (2.2). Eli yksittäisen pisteen arvo kuva-avaruudessa saadaan kaavalla

$$\begin{aligned} \bar{f}(x, y, z) &= \sum_{j=1}^m X_j b \left(\sqrt{(x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2} \right) \\ &= \sum_{j=1}^m X_j b(r_j). \end{aligned} \tag{4.7}$$

Sijoitetaan vielä määritelmä (3.1), jolloin saadaan

$$b(r_j) = \begin{cases} \frac{1}{I_2(\alpha)} (1 - (r_j/a)^2) I_2 \left[\alpha \sqrt{1 - (r_j/a)^2} \right], & 0 \leq r_j \leq a \\ 0, & r_j > a. \end{cases}$$

Nyt kuva-avaruus täytetään vokseleilla ja jokaisen vokselin arvoksi valitaan kaavan (4.7) arvo siten, että $\{x, y, z\}$ on kyseisen vokselin keskipiste. Menetelmä tuottaa pientä virhettä, sillä vokselin reuna-alueilla siihen saattaa vaikuttaa kuula, joka ei kuitenkaan ylety vokselin keskipisteeseen. Kuitenkin, jos vokselit valitaan riittävän pieneksi ja kuulien parametri α riittävän isoksi, on virhe lähes merkityksetön. Lisäksi numeerissa laskennassa ei ole tarvetta käydä läpi kaikkia eri kuulia, sillä suurin osa niistä ei vaikuta yksittäiseen vokseliin. Tässä tutkielmassa vokselin keskipisteeseen sijoitettiin kuulan sädetä hieman suurempi kuutio ja ainoastaan ne kuulat, joiden keskipiste oli

tämän kuution sisällä, laskettiin mukaan kaavaan (4.7).

4.2.5 Arviointimenetelmät

Rekonstruktioparametreiksi valittiin 30 yhdistelmää (taulukko 2), joiden tarkoituksena on simuloida käytännön kannalta oleellisia tilanteita mahdollisimman hyvin ollen kuitenkin vielä järjellisessä ajassa laskettavia. Liitteessä I on esitetty koodit, joilla parametreja voidaan valita. Pienimmillä säteillä jou-

Taulukko 2: Parametriyhdistelmät. Jos parametriyhdistelmä $\{a, \alpha\}$ on mukana simulaatiossa, niin käytetyt relaksaatioparametrin λ_1 arvot on ilmoitettu taulukossa. Lisäksi jokaiselle yhdistelmälle on ilmoitettu puoliarvoveveys (FWHM).

Säde a	Käytettävä hyperbeli	Parametrin λ arvot	FWHM
$a = 1,8$	1 ($\alpha = 8,8926$)	{1; 0,6; 0,35}	1,2751
	2 ($\alpha = 4,4038$)	{1; 0,6; 0,35}	1,6092
	3 (α ei määr.)	-	-
	4 (α ei määr.)	-	-
$a = 2,5$	1 ($\alpha = 14,0680$)	{1; 0,6; 0,35}	1,4670
	2 ($\alpha = 11,7569$)	{1; 0,6; 0,35}	1,5830
	3 ($\alpha = 7,6879$)	{1; 0,6; 0,35}	1,8710
	4 (α ei määr.)	-	-
$a = 3,2$	1 ($\alpha = 18,8528$)	{1; 0,6; 0,35}	1,6512
	2 ($\alpha = 17,1972$)	-	1,7203
	3 ($\alpha = 14,7181$)	{1; 0,6; 0,35}	1,8419
	4 ($\alpha = 10,8559$)	{1; 0,6; 0,35}	2,0941
$a = 4,0$	1 ($\alpha = 24,1417$)	{1; 0,6; 0,35}	1,8440
	2 ($\alpha = 22,8722$)	-	1,8904
	3 ($\alpha = 21,0718$)	-	1,9624
	4 ($\alpha = 18,5808$)	{1; 0,6; 0,35}	2,0776

duttiin jättämään joitain parametreja pois, sillä Fourier-ehtoja toteuttavia, kuvan 13 hyperbelejä vastaavia parametrin α arvoja ei ole tarpeeksi. Suurimmalla säteellä $a = 4$ valittiin vain kaksi neljästä parametrin α arvosta

laskenta-ajan säästämiseksi. Esimerkiksi jalanjälkitaulun laskeminen on noin viisi kertaa työläämpää, kun säteen arvo $a = 4$, verrattuna tilanteeseen, jossa $a = 1,8$. Taulukosta 2 nähdään kuitenkin välistä jätettyjen parametrien α arvojen olevan suhteellisesti lähellä mukaan otettua arvoa $\alpha = 24,141$, jolloin on oletettavissa, että kuvatkin ovat samankaltaiset. Samoin säteellä $a = 3,2$ valittiin vain kolme parametriyhdistelmää neljästä laskenta-ajan säästämiseksi. Suurimmilla säteen a arvoilla rikotaan aiemmin esitelty puoliarvoveysehdo, eli kuulan puoliarvoveveys on skannerin geometrian puoliarvovevyyttä 1,7 suurempi. Kyseessä on kuitenkin simuloitu geometria, joten sille laskettu puoliarvoveveys on suuntaa-antava. Suurimmilla säteen arvoilla voi kuitenkin esiintyä puoliarvovehdon rikkoutumisesta johtuvaa virhettä.

Tuloksia tarkasteltiin vertailemalla referenssikuvana olevaa Shepp-Loganin testikuvaa rekonstruktioalgoritmin tuottamiin kuviin. Vertailussa käytettiin mielenkiintoalueiden (Region of Interest, ROI) keskiarvojen ja hajonnan vertailua (kuva 20), sekä koko kuvan alueelta laskettuja keskimääräisen virheen (Mean Absolute Error, MAE), sekä keskimääräisen neliövirheen (Mean Square Error, MSE) tunnuslukuja. Virheet skaalattiin vielä referenssikuvasta laskettuihin summiin ja neliösummiin, jolloin saatiin normalisoidut tunnusluvut; normalisoitu keskimääräinen virhe (Normalised Mean Absolute Error, NMAE):

$$NMAE(\mathbf{X} - \hat{\mathbf{X}}) = \frac{\sum_{j=1}^m |X_j - \hat{X}_j|}{\sum_{j=1}^m X_j}$$

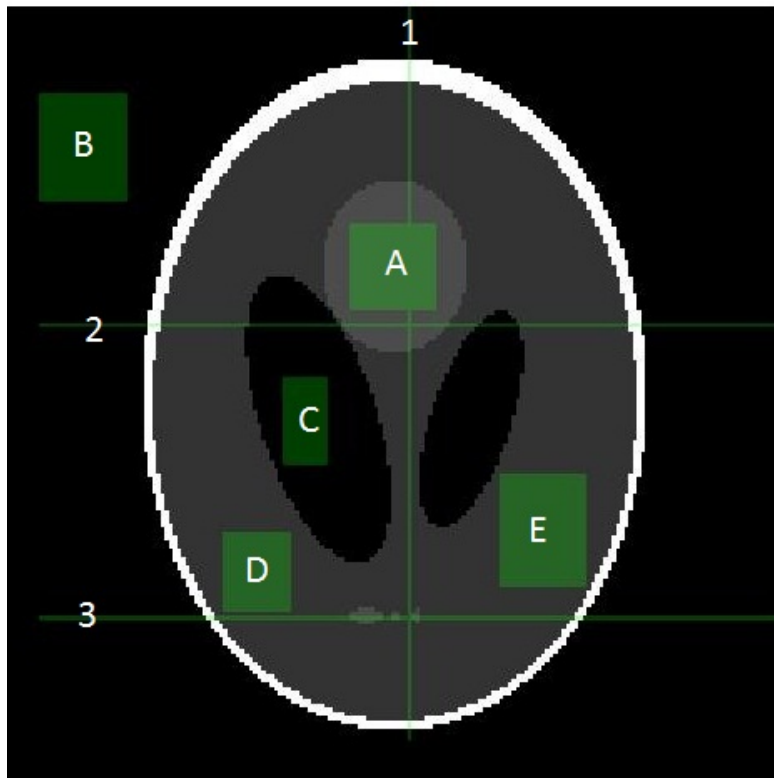
ja normalisoitu keskimääräinen neliövirhe (Normalised Mean Square Error, NMSE):

$$NMSE(\mathbf{X} - \hat{\mathbf{X}}) = \frac{\sum_{j=1}^m (X_j - \hat{X}_j)^2}{\sum_{j=1}^m X_j^2},$$

joissa \mathbf{X} on referenssikuvan kuvavektori ja $\hat{\mathbf{X}}$ rekonstruoidun kuvan kuvavektori. Virheluvut NMAE ja NMSE voidaan esittää myös prosentuaalisesti,

joilloin esimerkiksi $NMSE(\mathbf{X} - \hat{\mathbf{X}}) = 0,2$ tarkoittaa, että kuvien välillä on neliövirhemiesä 20 % ero.

Kuvista laskettiin myös profiilikäyrät, joita vertailtiin referenssin ja rekonstruoitujen kuvien välillä. Profiilikäyrät laskettiin kolmesta kohtaa testikuvaa, mikä on esitetty kuvassa 20. Keskiarvon ja hajonnan tarkastelulla py-



Kuva 20: Referenssikuva, johon on piirretty mielenkiintoalueet A-E, sekä profiilikäyrien laskemisviivat 1-3.

ritään tarkastelemaan eri alueiden rekonstruktion tarkkuutta, profiilikäyrillä reunojen rajausta ja normalisoiduilla virheluvuilla koko kuvan rekonstruktion tarkkuutta.

4.3 Rekonstruktioalgoritmin tulokset

Tulokset on jaettu kolmeen osaan. Ensimmäisessä osassa tarkastellaan saatuja kuvia visuaalisesti, sekä tehdään huomioita kuvien hajonnasta ja sil-

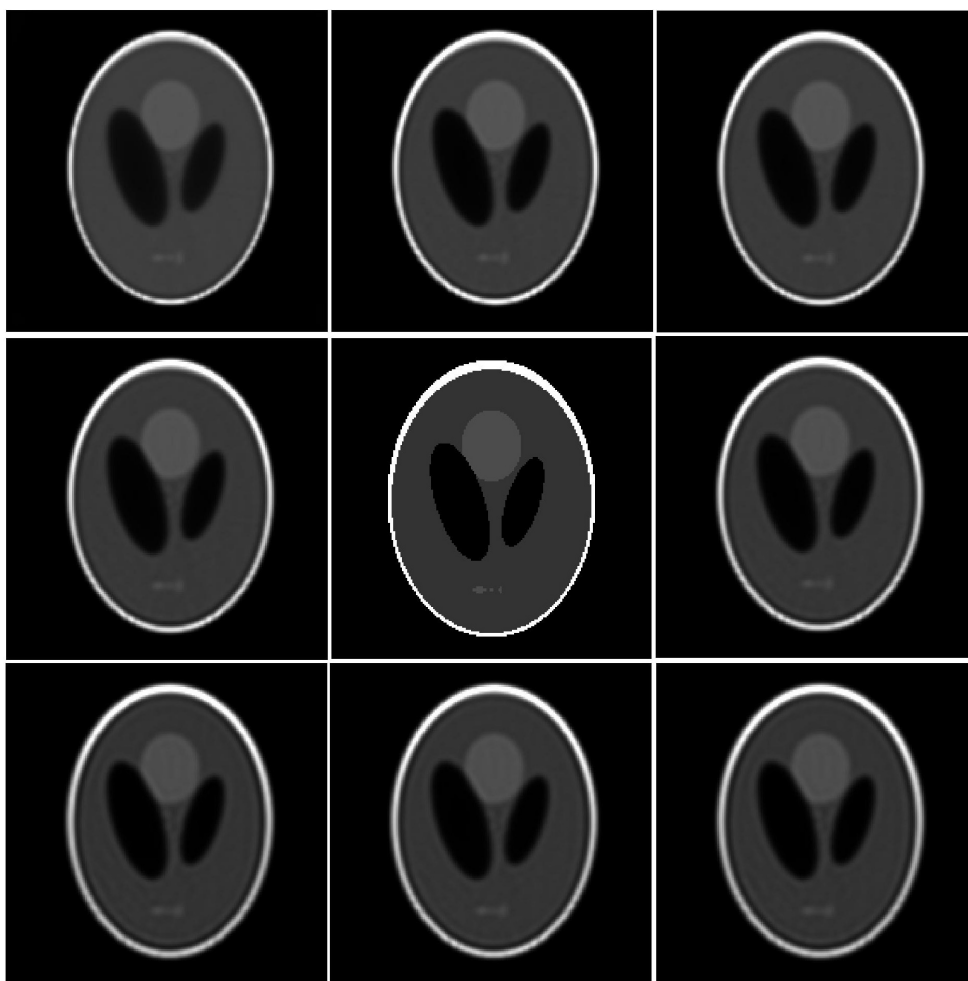
minnähtävistä virheistä. Toisessa osiossa vertaillaan eri mielenkiintoalueiden keskiarvoja sekä hajontaa ja tarkastellaan koko kuvasta laskettuja virhelukuja. Viimeisessä osassa piirretään kuvista profiilikäyriä ja vertaillaan näin eri parametrien vaikutusta rekonstruktioon.

Kaikki analyysit on suoritettu RAMLA-algoritmin relaksaatioparametrin arvolla $\lambda_1 = 1$. Relaksaatioparametrin arvoihin $\lambda_1 = 0,6$ ja $\lambda_1 = 0,35$ liittyviä tuloksia vertaillaan relaksaatioparametrin arvolla $\lambda_1 = 1$ saatuihin tuloksiin omassa kappaleessaan. Artikkelissa [5] saatu optimaalinen arvo $\lambda_1 \approx 0,12$ on tämän tutkielman kannalta liian pieni, sillä suhteellisen pienestä projisio-kulmien määrästä johtuen kuulien arvot eivät ehdi muuttua tarpeeksi, jos relaksaatioparametri λ on pieni.

4.3.1 Visuaalinen tarkastelu

Aluksi eri parametrijhdistelmien tuottamia kuvia tarkasteltiin visuaalisesti. Kuvassa 21 on esitetty referenssikuva, sekä relaksaatioparametrilla $\lambda_1 = 1$ saatuja kuvia iteraatiomäärän ollessa 5. Huomataan, että suuremmilla säteen arvoilla kuvissa esiintyy reunojen suuntaisia virheitä harmaalla alueella. Nämä virheet korostuvat kun parametria α pienennetään. Lisäksi kallon ja harmaan alueen väliin muodostuu musta rinki, jota ei ole referenssikuvassa. Virhe on suurempi säteen a arvon kasvaessa ja parametrin α arvon vähenessä. Paras kuvanlaatu on säteiden a arvoilla 1,8 ja 2,5. Paras kompromissi kuvan tarkkuuden ja virheiden välillä näyttäisi olevan parametrijhdistelmillä $\{a = 1,8; \alpha = 4,4038\}$ ja $\{a = 2,5; \alpha = 14,0680\}$.

Kuvassa 22 on taas kuvattu kolmelle eri parametrijhdistelmälle useampi iteraatio, jolloin virheen väheneminen ja rekonstruktion konvergoituminen on helppo huomata. Huomataan, että kuulien säteen kasvaessa kuvan tasaisten alueiden pikseliarvojen hajonta kasvaa, mutta samalla reunan ylikorostuminen vähenee. Iteraatiomäärien kasvaessa kuva selvästi tasoittuu ja virheen määrä vähenee. Suuremmilla säteillä nähdään yhä reunojen suuntaisia virheitä, kun taas pienemmillä säteillä kuva on viidellä iteraatiolla tasaisempi.



Kuva 21: Rekonstruktioita viidellä iteraatiolla. Parametriyhdistelmät ylävasemmalta lukien: $\{a = 1,8; \alpha = 8,8926\}$, $\{a = 1,8; \alpha = 4,4038\}$, $\{a = 2,5; \alpha = 14,0680\}$, $\{a = 2,5; \alpha = 11,7569\}$, referenssikuva, $\{a = 3,2; \alpha = 18,8528\}$, $\{a = 3,2; \alpha = 10,8559\}$, $\{a = 4,0; \alpha = 24,1417\}$ ja $\{a = 4,0; \alpha = 18,5808\}$.

4.3.2 Mielenkiintoalueiden analyysi ja koko kuvan virheet

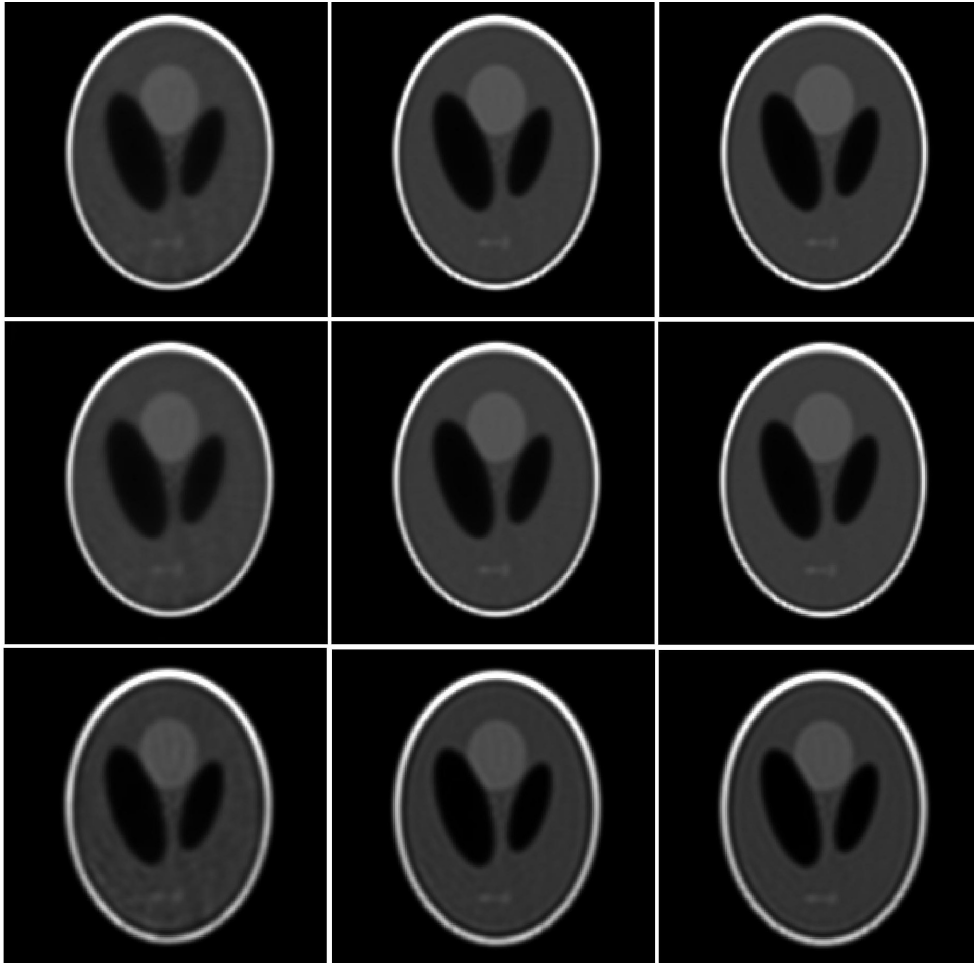
Kuvien rekonstruktion mielenkiintoalueiden keskiarvot on esitetty taulukossa 3, hajonnat taulukossa 4 ja virheluvut taulukossa 5. Referenssikuvan keskiarvot mielenkiintoalueittain ovat $\{1,5; 0; 0; 1; 1\}$. Taulukosta 3 huomataan, että pienillä säteillä rekonstruktiossa esiintyy ylikorostusta, joka vähenee säteen kasvaessa. Lisäksi parametrin α pienentäminen vähentää ylikorostusta. Sä-

Taulukko 3: Mielenkiintoalueiden keskiarvot eri parametriyhdistelmillä. Mielenkiintoalueiden nimet samoin kuin kuvassa 20. Mukana iteraatiot $\{1, 3, 5\}$.

Säde a	Käytettävä hyperbeli	Mielenkiintoalue				
		A	B	C	D	E
1,8	1 ($\alpha = 8,8926$)	1,655	0,042	0,317	1,217	1,219
		1,647	0,006	0,229	1,229	1,226
		1,648	0,002	0,201	1,231	1,228
	2 ($\alpha = 4,4038$)	1,660	$6,67 \times 10^{-4}$	0,140	1,147	1,139
		1,656	$4,49 \times 10^{-6}$	0,083	1,154	1,148
		1,656	$4,01 \times 10^{-7}$	0,068	1,154	1,149
2,5	1 ($\alpha = 14,0680$)	1,642	$1,01 \times 10^{-3}$	0,145	1,131	1,126
		1,636	$1,17 \times 10^{-5}$	0,088	1,139	1,133
		1,636	$1,15 \times 10^{-6}$	0,072	1,139	1,135
	2 ($\alpha = 11,7569$)	1,608	$1,29 \times 10^{-5}$	0,108	1,091	1,080
		1,602	$3,07 \times 10^{-7}$	0,061	1,097	1,090
		1,602	$1,39 \times 10^{-8}$	0,050	1,096	1,091
	3 ($\alpha = 7,6879$)	1,538	$2,74 \times 10^{-7}$	0,066	1,026	1,006
		1,534	$7,19 \times 10^{-12}$	0,034	1,027	1,021
		1,532	$3,10 \times 10^{-14}$	0,026	1,025	1,022
3,2	1 ($\alpha = 18,8528$)	1,574	$1,54 \times 10^{-5}$	0,086	1,059	1,044
		1,570	$7,95 \times 10^{-9}$	0,047	1,062	1,056
		1,569	$1,62 \times 10^{-10}$	0,038	1,061	1,057
	3 ($\alpha = 14,7181$)	1,535	$1,58 \times 10^{-7}$	0,064	1,024	1,003
		1,531	$2,75 \times 10^{-12}$	0,033	1,024	1,019
		1,530	$1,00 \times 10^{-14}$	0,025	1,023	1,020
	4 ($\alpha = 10,8559$)	1,513	$1,52 \times 10^{-10}$	0,050	0,999	0,979
		1,509	0	0,024	1,000	0,998
		1,507	0	0,018	0,999	1,000
4,0	1 ($\alpha = 24,1417$)	1,533	$8,16 \times 10^{-8}$	0,062	1,021	1,000
		1,528	$8,75 \times 10^{-13}$	0,031	1,021	1,016
		1,527	$2,00 \times 10^{-15}$	0,024	1,020	1,017
	4 ($\alpha = 18,5808$)	1,513	$9,02 \times 10^{-11}$	0,050	0,999	0,979
		1,510	0	0,024	1,001	0,999
		1,507	0	0,018	1,000	1,000
Referenssikuva		1,5	0	0	1	1

Taulukko 4: Mielenkiintoalueiden hajonnat eri parametrijhdistelmillä. Mielenkiintoalueiden nimet samoin kuin kuvassa 20. Mukana iteraatiot $\{1, 3, 5\}$.

Säde a	Käytettävä hyperbeli	Mielenkiintoalue				
		A	B	C	D	E
1,8	1 ($\alpha = 8,8926$)	0,019	0,025	0,039	0,022	0,020
		0,008	0,005	0,036	0,018	0,020
		0,007	0,002	0,035	0,016	0,022
	2 ($\alpha = 4,4038$)	0,021	$8,41 \times 10^{-4}$	0,035	0,038	0,034
		0,013	$9,98 \times 10^{-6}$	0,028	0,031	0,035
		0,012	$9,64 \times 10^{-7}$	0,025	0,029	0,037
2,5	1 ($\alpha = 14,0680$)	0,020	$1,28 \times 10^{-3}$	0,035	0,035	0,032
		0,012	$2,19 \times 10^{-5}$	0,029	0,031	0,034
		0,011	$1,15 \times 10^{-6}$	0,026	0,030	0,036
	2 ($\alpha = 11,7569$)	0,035	$6,97 \times 10^{-7}$	0,030	0,066	0,056
		0,020	$2,92 \times 10^{-11}$	0,022	0,052	0,051
		0,013	$4,07 \times 10^{-12}$	0,023	0,036	0,041
	3 ($\alpha = 7,6879$)	0,035	$6,97 \times 10^{-7}$	0,030	0,066	0,056
		0,020	$2,92 \times 10^{-11}$	0,022	0,052	0,051
		0,017	$1,49 \times 10^{-13}$	0,020	0,050	0,052
3,2	1 ($\alpha = 18,8528$)	0,028	$2,90 \times 10^{-5}$	0,032	0,050	0,046
		0,016	$2,40 \times 10^{-8}$	0,024	0,044	0,044
		0,015	$2,40 \times 10^{-10}$	0,022	0,042	0,045
	3 ($\alpha = 14,7181$)	0,036	$4,14 \times 10^{-7}$	0,031	0,068	0,058
		0,021	$1,15 \times 10^{-11}$	0,023	0,056	0,053
		0,018	$4,80 \times 10^{-14}$	0,021	0,053	0,053
	4 ($\alpha = 10,8559$)	0,046	$5,60 \times 10^{-10}$	0,032	0,092	0,074
		0,026	0	0,023	0,070	0,066
		0,022	0	0,021	0,067	0,066
4,0	1 ($\alpha = 24,1417$)	0,037	$2,22 \times 10^{-7}$	0,031	0,071	0,061
		0,021	$3,81 \times 10^{-12}$	0,023	0,058	0,054
		0,019	$1,20 \times 10^{-14}$	0,021	0,055	0,055
	4 ($\alpha = 18,5808$)	0,046	$3,37 \times 10^{-10}$	0,032	0,092	0,076
		0,027	0	0,023	0,072	0,067
		0,023	0	0,021	0,068	0,068
Referenssikuva		0	0	0	0	0



Kuva 22: Rekonstruktioita eri iteraatioilla. Iteraatiomäärät vasemmalta oikealle: 1,3,5. Parametriyhdistelmät ylhäältä alaspäin: $\{a = 1,8; \alpha = 4,4038\}$, $\{a = 2,5; \alpha = 14,0680\}$ ja $\{a = 4,0; \alpha = 24,1417\}$.

teen a arvoilla 3,2 ja 4,0 ja käytettäessä hyperbelejä 3 ja 4 ovat mielenkiintoalueiden keskiarvot lähes oikeat kaikissa alueissa, aluetta C lukuunottamatta. Alue C on musta alue kuvan keskellä, jolloin sinne tulee kontribuutiota myös niiltä kuulilta, jotka sijaitsevat korkean aktiivisuuden alueilla. Tämä vaikutus kuitenkin pienenee säteen kasvaessa. Iteraatiomäärän kasvaessa yhdestä kolmeen keskiarvot hakeutuvat pääsääntöisesti lähemmäksi oikeaa arvoa. Poikkeuksena alue E, joka sijaitsee lähellä sekä mustaa aluetta, että kirkasta reunaa. Reunalla tapahtuvat virheet voivat hieman vaikuttaa keskiarvoihin,

Taulukko 5: Virhelukujen arvot eri parametrijhdistelmillä. Luvut laskettu koko leikkeestä. Mukana iteraatiot $\{1, 3, 5\}$.

Säde a	Käytettävä hyperbeli	Virheluku	
		NMAE	NMSE
$a = 1,8$	1 ($\alpha = 8,8926$)	0,405; 0,340; 0,327	0,145; 0,125; 0,121
	2 ($\alpha = 4,4038$)	0,293; 0,275; 0,271	0,118; 0,110; 0,101
$a = 2,5$	1 ($\alpha = 14,0680$)	0,293; 0,272; 0,268	0,120; 0,110; 0,107
	2 ($\alpha = 11,7569$)	0,274; 0,258; 0,254	0,121; 0,112; 0,110
	3 ($\alpha = 7,6879$)	0,265; 0,246; 0,242	0,131; 0,128; 0,123
$a = 3,2$	1 ($\alpha = 18,8528$)	0,269; 0,252; 0,248	0,125; 0,117; 0,115
	3 ($\alpha = 14,7181$)	0,271; 0,251; 0,247	0,134; 0,127; 0,125
	4 ($\alpha = 10,8559$)	0,286; 0,267; 0,263	0,148; 0,141; 0,139
$a = 4,0$	1 ($\alpha = 24,1417$)	0,274; 0,255; 0,251	0,137; 0,129; 0,127
	4 ($\alpha = 18,5808$)	0,291; 0,272; 0,268	0,151; 0,143; 0,142

mikä näkyy taulukossa. Iteraatioiden kasvattaminen kolmesta viiteen ei tuota juurikaan muutoksia keskiarvoihin, paitsi tummilla alueilla, jotka pienenevät kohti nollaa. Iteraatioiden jatkaminen yli viiden ei tunnu mielekkäältä, sillä keskiarvot eivät enää muutu juurikaan iteraatioita lisäämällä.

Referenssikuvissa mielenkiintoalueiden hajonnat ovat nolliä, joten mitä pienempi hajonta rekonstruktiossa on, sitä paremmin rekonstruktio kuvaa tasaista mielenkiintoaluetta. Taulukosta 4 nähdään, että taustaa lukuunottamatta hajonnat kasvavat säteen kasvaessa. Osaltaan hajontaa selittää puoliarvoleveysehdon rikkoituminen suuremmilla säteen a arvoilla. Parametrin α pienentäminen kasvattaa hajontaa. Tämä on vastakkainen ilmiö taulukon 3 keskiarvoja vertailtaessa havaittuun nähden. Hajonta myös pienenee iteraatiomääriä lisättäessä, muutamia alueen E poikkeuksia lukuunottamatta. Samoin kuin keskiarvojen kohdalla iteraatiomäärän kasvattaminen kolmesta viiteen ei juurikaan vaikuta hajontaan. Parametrien erisuuntaisesta vaikutuksesta johtuen hajonnan ja keskiarvon välillä tulee valita kompromissi, missä kumpikaan ominaisuus ei kärsi liikaa.

Virhetaulukkoa 5 tarkasteltaessa huomataan, että iteraatiomäärän kasvat-

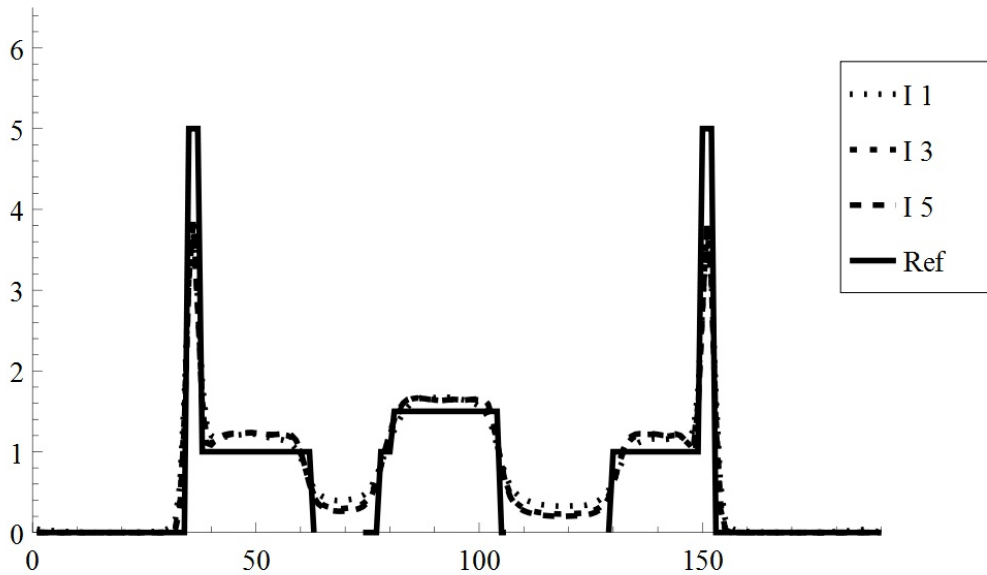
taminen poikkeuksetta vähentää virhettä. Mielenkiintoinen ominaisuus kuitenkin on, että virhe ei reagoi lineaarisesti säteeseen a ja parametriin α , vaan taulukosta on nähtävissä käännekohta, jonka jälkeen säteen kasvattaminen lisää virhettä. Absoluuttisissa virheissä (NMAE) käännekohta tapahtuu säteiden $a = 3,2$ ja $a = 4,0$ välissä. Tätä ennen virhe pieneni kuulan sädettä kasvatettaessa, mutta säteellä 4.0 virhe kääntyi taas kasvuun. Neliövirheissä (NMSE) käännekohta tapahtuu jo säteiden $a = 2,5$ ja $a = 3,2$ välissä. Parametrin α vaikutus oli samankaltainen. Pienillä kuulan säteen a arvoilla parametrin α kasvattaminen vaikuttaa virhettä vähentävästi, mutta säteen kasvaessa, parametrin α kasvattaminen alkaakin kasvattaa kuvassa esiintyvää virhettä. Molemmilla virhetyypeillä (NMAE ja NMSE) muutos tapahtui säteen $a = 1,8$ jälkeen. Puoliarvoveysehdon rikkoutuminen osaltaan vaikuttaa virheen lisääntymiseen suurimmilla säteen arvoilla.

4.3.3 Profilikäyrät

Profilikäyrät on esitetty edellisissä tarkasteluissa mielenkiintoisiksi havaituille parametrijhdistelmille. Jokaisessa kaaviossa on referenssikuva, sekä eri iteraatioita omilla käyrillään. Profilikäyriä on kolme (kuva 20). Ensimmäisen tarkoitus on mitata paksun reunan rekonstruktiota, rekonstruktion tarkkuutta mustien alueiden välillä, sekä profilikäyrän tasaisuutta kuljettaessa kuvan alalaidan pienten kirkkaiden kappaleiden yli. Referenssikäyrä on mielenkiintoaluetta A ympäröivän kirkkaan alueen ja kuvan alalaidan välillä tasainen.

Toinen profilikäyrä mittaa mustien alueiden rekonstruktion tarkkuutta kuvan keskiosissa, sekä reunojen käyttäytymistä. Referenssikäyrässä muutokset ovat jyrkkiä kuvapisteen arvon muuttuessa suoraan alueen arvosta toiseen. Kolmas profilikäyrä tarkastelee kuvan alalaidan pienten kirkkaiden alueiden rekonstruktion tarkkuutta. Referenssikuvassa on kolme pientä piikkiä muuten tasaisen alueen keskellä. Kaikki profilikäyrät alkavat kuvan 20 oikeasta tai yläreunasta ja kulkevat koko kuvan yli.

Pienillä säteen arvoilla on havaittavissa jo mielenkiintoalueiden tarkastelussa esiin tullut tasoero, kuva 23. Eri iteraatioiden käyrät ovat hyvin samalla tasolla kaikkialla paitsi rekonstruoitavan kuvan mustilla alueilla, joissa

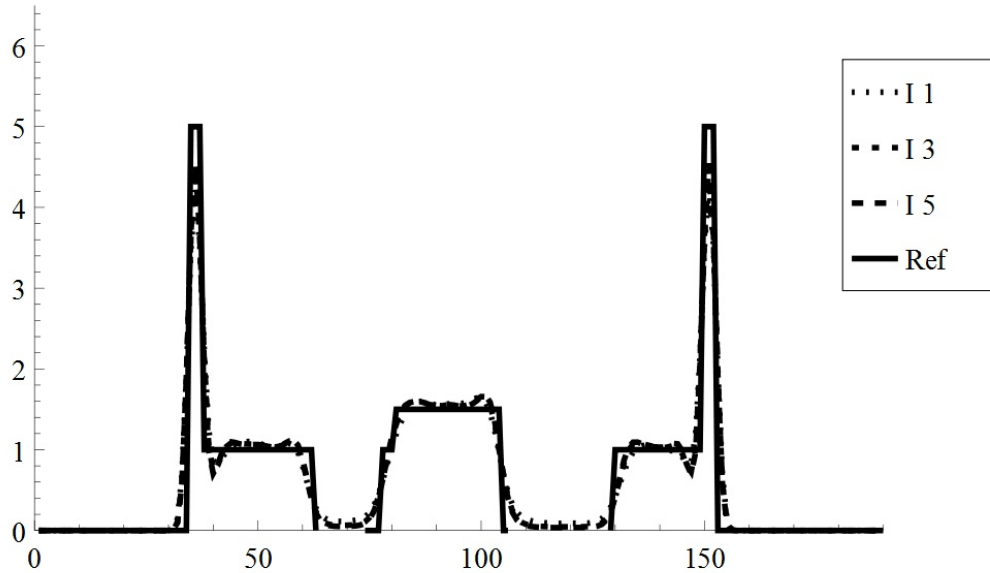


Kuva 23: Profilikäyrä 2 parametreilla $a = 1,8$ ja $\alpha = 8,8926$. Kuva yliestimoituu selkeästi.

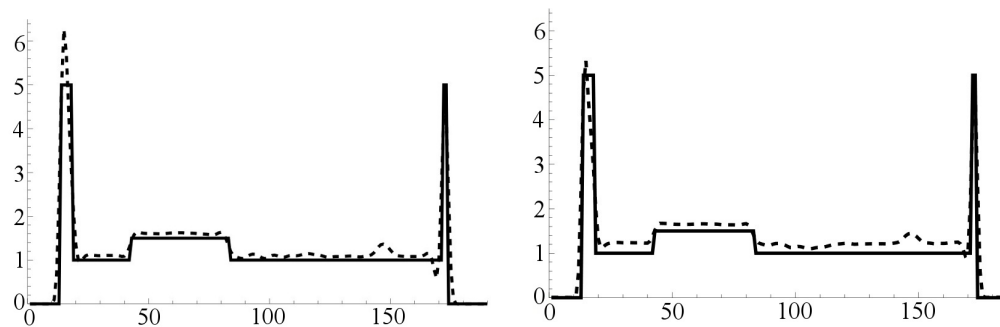
iteraatioiden lisääminen parantaa rekonstruktion tarkkuutta. Kuulan säteen kasvattaminen vähentää yliestimointia, mutta samalla ohuiden piikkien, eli nyt kuvan reunan kallon, lähelle tulee selkeä aliestimoituminen, kuva 24. Koska profilikäyrät eivät juurikaan eroa eri iteraatiomäärillä, esitetään tästä eteenpäin kuvia, joissa on ainoastaan iteraatio 5 sekä referenssikäyrä.

Profilikäyrissä 1 huomataan käyrän pysyvän tasaisena suurimman osan aikaa muutamaa poikkeusta lukuunottamatta. Profilikäyrän keskellä, alueella, jossa käyrä kulkee lähellä molempia mustia alueita, nähdään profilikäyrässä epätasaisuutta, minkä lisäksi vähäisellä iteraatiomäärällä käyrä tippuu ympäristöönsä alemmaksi, kuva 25a. Käyrän oikeassa laidassa on nähtävissä jokaisella parametrijohdistelmällä kumpu, joka seuraa käyrän sijoittumisesta aivan pienten kirkkaiden kohteiden läheisyyteen. Osa aktiivisuudesta vuotaa harmaalle alueelle, mikä näkyy profilikäyrän kumpuna. Kuulien päällekkäisyydestä johtuen ilmiöstä ei voi päästä kokonaan eroon, mutta pienemmällä säteellä ja suuremmalla parametrin α arvolla sitä voidaan rajoittaa, kuva 25b.

Reunoilla huomataan kaksi erilaista ilmiötä. Profilikäyrän oikeassa reu-



Kuva 24: Profiilikäyrä 2 parametreilla $a = 3,2$ ja $\alpha = 18,8528$. Kuva estimoituu hyvin, mutta kallon rajoilla näkyy reuna-artefakta.

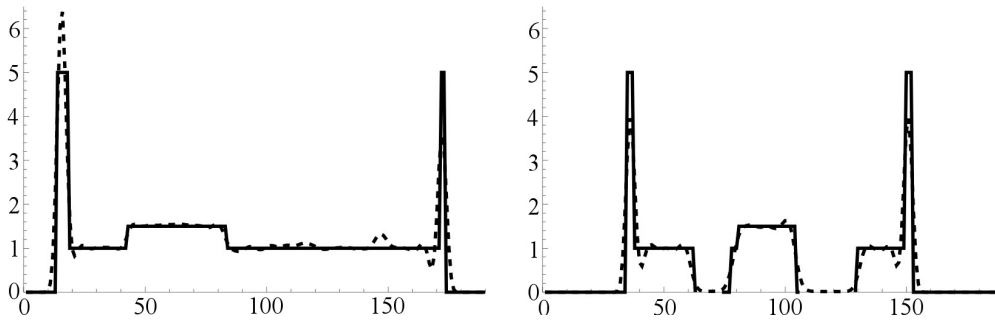


(a) $a = 2,5$ ja $\alpha = 11,7569$. Vasen reuna yliestimoiduu ja oikealla aktiivisuuden vuotoa harmaalle alueelle.

(b) $a = 1,8$ ja $\alpha = 8,8926$. Käyrä pääosin yliestimoidunut, mutta oikealla oleva kumpu on pienempi kuin kuvassa 25a. Vasen reuna ei yliestimoidu.

Kuva 25: Profiilikäyriä 1 eri parametrien a ja α arvoilla. Referenssikäyrä yhtenäisellä, ja 5. iteraatio katkoviivalla.

nassa tapahtuu jo edellä mainittu kapeaan reunaan liittyvä reunan viereinen aliestimoituminen, minkä lisäksi varsinainen reuna on aliestimoitunut. Reunan aliestimoituminen johtunee mustalla taustalla olevien kuulien vaikutuksesta ohueen reunaan, sekä suurimmilla säteen a arvoilla tapahtuvaan puoliarvovevyyshdon rikkoutumisesta. Käyrän vasemmassa reunassa oleva paksumpi reuna taas yliestimoituu rajusti, mutta sen vieressä ei ole havaittavissa juurikaan aliestimoitumista ennen kuin suurimmalla testatulla säteellä, kuva 26a. Kuvan 25b tilanne on testatuista parametrijhdistelmistä ainoa, jolla profiilikäyrän 2 vasen reuna ei selkeästi yliestimoidu.



(a) $a = 4,0$ ja $\alpha = 18,5808$. Kuva on estimoitunut oikein, kumpua ja reuna- virheitä lukuunottamatta. Vasemmas- sa reunassa on pieni kuoppa.

(b) $a = 3,2$ ja $\alpha = 10,8559$. Verrattaes- sa kuvaan 24 reunoilla esiintyvät vir- heet korostuvat. Reunan kuoppa joh- tuu liian suuresta puoliarvovevyydestä.

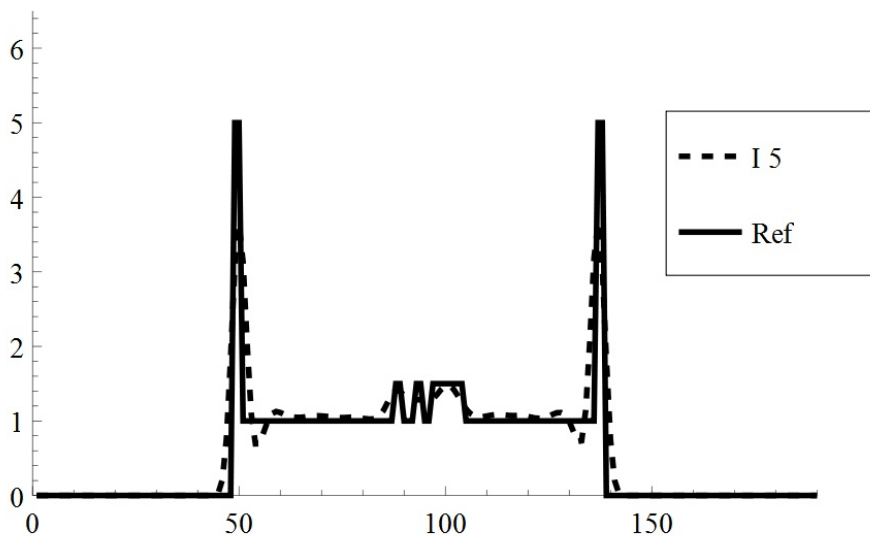
Kuva 26: Vasemmalla profiilikäyrä 1, ja oikealla profiilikäyrä 2. Referenssi- käyrä yhtenäisellä, ja 5. iteraatio katkoviivalla.

Profiilikäyrässä 2 huomataan samanlainen ohueen reunaan liittyvä ilmiö kuin edellä, kuva 24. Lisäksi eri alueiden välillä tapahtuvat siirtymät ovat loivia, mistä seuraa yliestimoituminen mustien alueiden reunalla, ja vastaa- vasti aliestimoituminen kirkkaiden alueiden reunoilla. Lisäksi parametrin α pienentäminen kasvattaa kirkkaiden alueiden reunalla havaittavaa aliestimoi- tumista, kuva 26b.

Pienemmän parametrin α arvon takia kuulan profiili on leveämpi, mistä seuraa reunan arvon aliestimoituminen suurta parametrin α arvoa vastaa- van rekonstruktion reunaa voimakkaammin. Tämä johtuu siitä, että reunan

päällä sijaitseva ison arvon saava leveä kuula levittää vaikutustaan myös ympäristöön kapeaa kuulaa enemmän, jolloin leveän kuulan tulee olla arvoltaan pienempi, jottei piikin ympärillä oleva tasainen alue yliestimoidu.

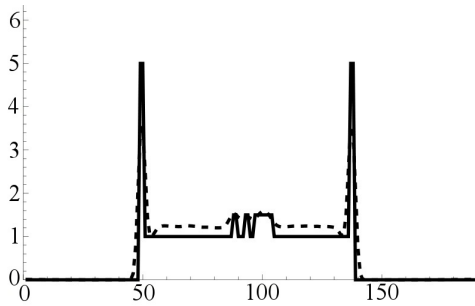
Kolmas profiilikäyrä mittaa pienten yksityiskohtien estimoitumisen tarkkuutta rekonstruktiossa. Profiilikäyrä kulkee kaikkien kolmen pienen ja kirkkaan alueen yli kuvan alaosassa. Referenssikuvassa profiilikäyrä on luonnollisesti tarkkarajainen, mutta rekonstruktioissa käyrät ovat loivia, eikä eri alueiden välillä ole selkeää eroa, kuva 27. Etenkin kaksi pienintä aluetta sulautuvat



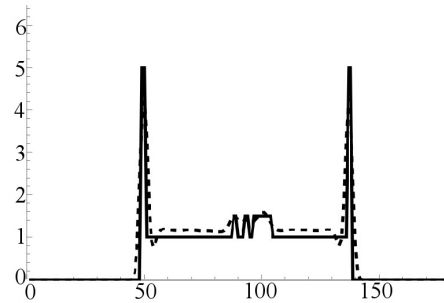
Kuva 27: Profiilikäyrä 3 parametreilla $a = 3,2$ ja $\alpha = 18,5808$. Kirkkaat kohteet erottuvat ympäristöstään, mutteivat juurikaan toisistaan.

yhteen. Huonoimmilla parametrijhdistelmillä kirkkaat alueet eivät juurikaan edes erotu ympäristöstään, kuva 28a, mutta parametrin α pienentäminen parantaa välittömästi kirkkaiden alueiden erottumista, kuva 28b

Profiilikäyrien perusteella suuret kuulan säteen a arvot tuottavat laajoilla alueilla tarkimman rekonstruktion, kun taas pienet säteen a arvot yliestimoiduvat. Kahden toisistaan huomattavasti poikkeavan alueen rajalla suuri säteen arvo kuitenkin tuottaa profiilikäyriin selkeästi havaittavan reunavirheen, joka osaltaan johtune puoliarvoleveysehdon rikkoutumisesta. Sädettä ei kannata kasvattaa tarpeettomasti; jo sädettä $a = 3,2$ vastaavat profiilikäy-



(a) $a = 1,8$ ja $\alpha = 8,8926$. Kirkkaat kohteet vain hieman ympäristöään voimakkaampia. Selkeä yliestimointuminen harmaalla alueella.



(b) $a = 1,8$ ja $\alpha = 4,4038$. Reunavirhe voimistuu, mutta käyrä estimoituu paremmin. Kirkkaat kohdat erottuvat selkeämmin kuin kuvassa 28a.

Kuva 28: Profilikäyriä 3 eri parametrien a ja α arvoilla. Referenssikäyrä yhtenäisellä, ja 5. iteraatio katkoviivalla.

rät antavat referenssikuvan kanssa yhteneviä laajojen alueiden arvoja, mutta pienemmällä reuna-artefaktoilla kuin sädetä $a = 4,0$ vastaavat profilikäyrät.

Parametrin α muutokset vaikuttavat lähinnä kuvan reunoilla. Pienet parametrin α arvot kasvattavat reuna-artefaktoja ja samalla aliestimoivat ohuita reunoja suurilla parametrin α arvoilla enemmän. Profilikäyrien tarkastelun valossa kannattaa valita säde tarpeeksi suureksi, jotta keskiarvot estimoituvat oikein, ja tämän jälkeen maksimoida parametri α käyttämällä mahdollisimman pienen järjestysluvun hyperbeliä kaavasta (3.13). Parametrin α kasvattaminen pienentää kuulun puoliarvovaleveyttä, jolloin puoliarvovaleveysehtokin toteutuu helpommin.

4.3.4 Muut RAMLA-parametrin λ arvot

Tässä tutkielmassa päätettiin suorittaa laajamittainen tarkastelu vain yhdelle kolmesta testatusta parametrin λ arvoista. Pienemmillä parametrin λ arvoilla kuvan konvergoituminen tarvitsee useamman iteraation, mutta kuvanlaadussa tai mielenkiintoalueiden arvoissa ei ole viidellä iteraatiolla merkittäviä eroja. Taulukossa 6 on vertailtu mielenkiintoalueiden keskiarvoja, ja taulukossa 7 mielenkiintoalueiden hajontoja eri parametrijhdistelmillä.

Taulukko 6: Mielenkiintoalueiden keskiarvoja eri parametriyhdistelmillä. Mielenkiintoalueiden nimet samoin kuin kuvassa 20. Kaikki yhdistelmät iteraatiomäärällä 5. Jokaisessa solussa parametrin λ arvot järjestyksessä $\{0,35; 0,6; 1\}$.

Säde a	Käytettävä hyperbeli	Mielenkiintoalue				
		A	B	C	D	E
1,8	1 ($\alpha = 8,8926$)	1,638	0,070	0,344	1,206	1,211
		1,643	0,017	0,265	1,220	1,221
		1,647	0,002	0,201	1,231	1,229
3,2	1 ($\alpha = 18,8528$)	1,562	$1,09 \times 10^{-4}$	0,092	1,041	1,046
		1,564	$4,90 \times 10^{-7}$	0,058	1,052	1,051
		1,569	$1,62 \times 10^{-8}$	0,038	1,061	1,057
4,0	1 ($\alpha = 24,1417$)	1,529	$1,79 \times 10^{-3}$	0,127	1,000	1,004
		1,522	$5,28 \times 10^{-10}$	0,037	1,013	1,012
		1,527	$2,00 \times 10^{-15}$	0,024	1,020	1,017

Taulukosta 6 nähdään, että parametrin λ arvo 1 suorituu paremmin kylmillä alueilla ja pienemmät arvot paremmin kuumilla alueilla. Kuumilla alueilla arvo $\lambda = 1$ yleistimoi kuvaa pienempiä parametrin λ arvoja enemmän. Ero on kuitenkin suurimmillaan 2 % suuruusluokkaa, mikä jää täysin säteen a ja muotoparametrin α vaihteluilla saatujen erojen varjoon.

Hajontataulukosta nähdään eri parametrin λ arvojen välillä eroja, mutta ne ovat sadas- tai tuhannesosien suuruusluokkaa. Kylmillä alueilla $\lambda_1 = 1$ tuottaa taas parempia arvoja, mutta häviää hieman useassa kuumassa mielenkiintoalueessa. Suurimmalla osalla kuumista alueista parametrien arvot $\lambda_1 = 1$ ja $\lambda_1 = 0,6$ ovat hyvin lähellä toisiaan. Parametrin λ arvo 1 tuottaa kylmillä alueilla pienemmän hajonnan, muttei häviä juurikaan kuumilla alueilla, joten sen käyttö on suositeltavaa.

Myös profiilikäyrät pienemmillä parametrin λ arvoilla ovat hyvin samankaltaisia verrattuna parametrin λ arvolla 1 saatuihin profiilikäyriin. Pienemmillä parametrin λ arvoilla tosin jotkin yksityiskohdat, kuten profiilikäyrän 1 keskiosa rekonstruoituvat huonommin. Kuvassa 29 on havainnollistettu tätä ilmiötä. Pienellä parametrin λ arvolla profiilikäyrään jää selkeä kuoppa, joka

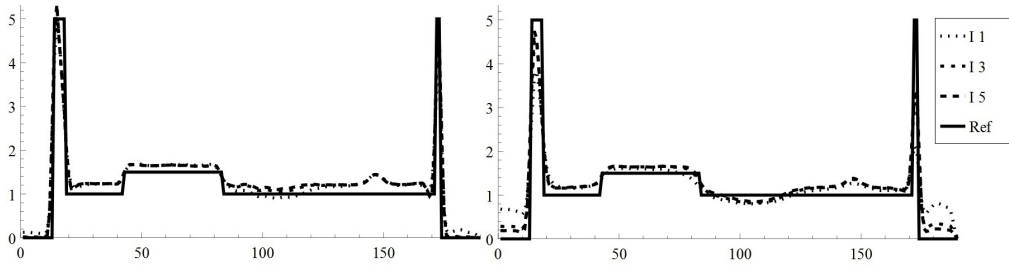
Taulukko 7: Mielenkiintoalueiden hajontoja eri parametriyhdistelmillä. Mielenkiintoalueiden nimet samoin kuin kuvassa 20. Kaikki yhdistelmät iteraatiomäärällä 5. Jokaisessa solussa parametrin λ arvot järjestyksessä $\{0,35; 0,6; 1\}$.

Säde a	Käytettävä hyperbeli	Mielenkiintoalue				
		A	B	C	D	E
1,8	1 ($\alpha = 8,8926$)	0,026	0,037	0,037	0,012	0,017
		0,010	0,012	0,037	0,017	0,017
		0,007	0,002	0,035	0,016	0,022
3,2	1 ($\alpha = 18,8528$)	0,012	$1,77 \times 10^{-4}$	0,030	0,038	0,038
		0,012	$1,20 \times 10^{-6}$	0,025	0,042	0,043
		0,015	$5,82 \times 10^{-10}$	0,022	0,042	0,045
4,0	1 ($\alpha = 24,1417$)	0,014	$4,00 \times 10^{-6}$	0,027	0,051	0,052
		0,016	$1,87 \times 10^{-9}$	0,024	0,055	0,055
		0,019	$1,20 \times 10^{-14}$	0,021	0,055	0,055

taas oletusarvolla $\lambda = 1$ tasoittuu iteraatioita lisättäessä.

Erikseen testattu parametrin λ arvo 0,1 on liian pieni, sillä tällöin kuva ei lähde konvergoitumaan oikeaan skaalaan, kuva 30. Voidaankin sanoa, että riittävän suuren parametrin λ arvon löytämisen jälkeen ei ole rekonstruktion kannalta suurtakaan väliä mikä arvo lopulta valitaan. Suurempi arvo parametrille λ parantaa kuitenkin hieman konvergenssin tarkkuutta, joten todetaan olevan järkevää pyrkiä mahdollisimman suuren parametrin λ arvon käyttämiseen.

Ylärajaa parametrille λ ei ole eksplisiittisesti määritelty, mutta sen tulee täyttää RAMLA-algoritmia määriteltäessä esitetty ehto $\lambda_k a_{ij} \leq 1$, kun a_{ij} on projisiomatriisin alkio. Näin ollen parametrin λ arvo 1 on turvallinen, sillä määritelmällisesti $a_{ij} \leq 1$. PET-kameran projisiokulmien määrää lisätessä voidaan pyrkiä pienentämään valmiin kuvan pikselien keskihajontaa pienentämällä parametria λ , mutta tämä tarkastelu tulee suorittaa erikseen jokaiselle projisiokulmien asetelulle.



Kuva 29: Profiilikäyrät 1 eri parametriyhdistelmällä ja iteraatioilla. Parametreina $a = 1,8$ ja $\alpha = 8,8926$. Vasemmalla RAMLA-parametri $\lambda = 1$, ja oikealla $\lambda = 0,35$. Oikeanpuoleinen kuva konvergoi liian nopeasti.

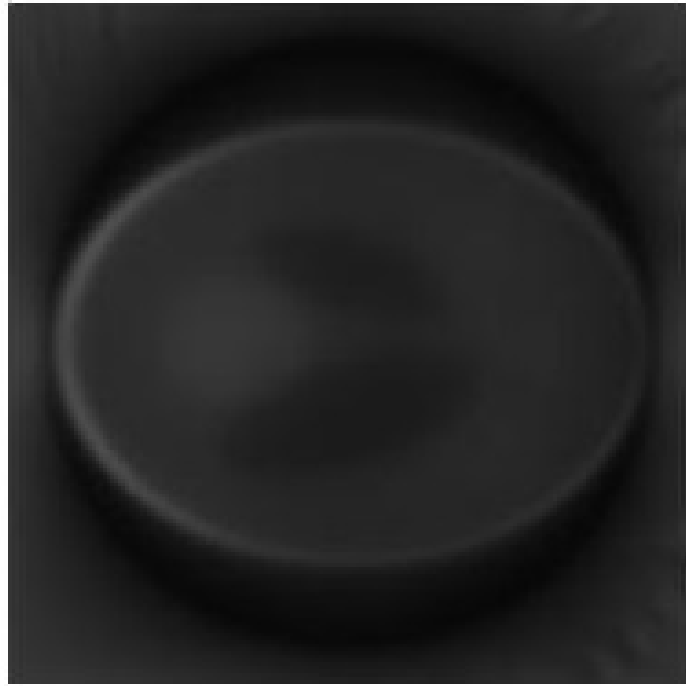
5 Tulosten soveltaminen lääketieteellisessä kuvantamisessa

Luvussa tarkastellaan muutamia erilaisia rekonstruktioparametrien yhdistelmiä anatomisen aivofantomien kuvantamisessa. Kamerana käytettiin Philips Gemini TF PET-MRI -kameraan liitettyä offline-rekonstruktio-yksikköä. Rekonstruktion pohjana käytettiin taulukon 1 ja kaavan (3.13) tuottamia hyperbelejä. Parametreista kiinnitettiin kuulan säde a , jonka jälkeen kokeiltiin eri muotoparametrin α arvoja. Kuvien tulokinnan lopputulemana saatiin kliiniseen kuvantamiseen soveltuvia parametrien valintaan liittyviä ohjeistuksia.

5.1 Johdanto

Edellisessä luvussa saadut simuloituvat ratkaisut ovat kliinisen tutkimuksen kannalta osittain suuntaa-antavia, sillä kliinisissä PET-kameroissa ja kuvauksissa on useita ominaisuuksia, joita ei simulaatiomallissa otettu huomioon. Tehtäessä useita lyhyitä kuvauksia PET-kameralla jonkin molekyylin jakautumisen ajallisen muutoksen selvittämiseksi jää väistämättä yksittäisestä kuvauksesta saatavaan dataan vähemmän havaintoja kuin yhteen pitkään kuvaukseen. Tämä statistiikan lasku asettaa haasteen rekonstruktioalgoritmile, joka ei saa hajota kuvan kohinasuhteen huonontuessa.

Samoin oikea kamera on herkempi kuvassa 12 havainnollistetulle alitäytölle, jolloin liian pienellä kuulan säteen a arvolla rekonstruoituun kuvaan tulee



Kuva 30: Kuva parametreilla $a = 1,8$, $\alpha = 8,8926$, $\lambda_1 = 0,1$. Kuva skaalattu välille $[0, 2]$ normaalin $[0, 5]$ sijasta.

selkeä jälki käytetystä kuulahilasta. Suurien ja profiililtaan leveiden kuulien tuottama tasainen pienihajontainen kuva ei myöskään ole hyvä ominaisuus, sillä kliinisessä kuvantamisessa on usein tärkeää erottaa pieniä yksityiskohtia, jotka helposti jäävät leveän kuulan tuottaman tasoitusvaikutuksen alle.

Onkin tärkeää tarkastella vielä erikseen oikeasta kamerasta saatavaa dataa, ja suorittaa tälle tarpeeksi laaja testaus, jotta eri tilanteiden vaatimuksiin löydetään oikeat parametrit. Tätä tutkielmaa kirjoitettaessa optimoinnin kohteena oleva Philipsin Gemini TF PET-MRI -kamera tuotti oletusasetuksillaan hyvin siloitellun näköisiä kuvia, joista on vaikea tai jopa mahdotonta erottaa pieniä kohteita. Tämän luvun tavoitteena on tuottaa muutamia nyrkkisääntöjä, tai parametriyhdistelmiä, joilla rekonstruktioiden tarkkuutta voidaan tarpeen vaatiessa parantaa.

5.2 Materiaali ja menetelmät

Pääosa rekonstruktioista suoritettiin kolmiulotteisesta anatomisesta testikapaleesta [9] saatuun dataan. Pääkallon mallinen, aivojen anatomisen rakenteen sisältävä kappale oli täytetty radioaktiivista fluorin isotooppia F-18 sisältävää muokattua glukoosia, joka on yleisesti käytetty merkkiaine PET-tutkimuksissa.

Anatomiselle aivofantomille käytetyt parametriyhdistelmät löytyvät taulukosta 8. Parametrit pyrittiin pitämään lähellä oletusparametreja $a = 2,8$ ja

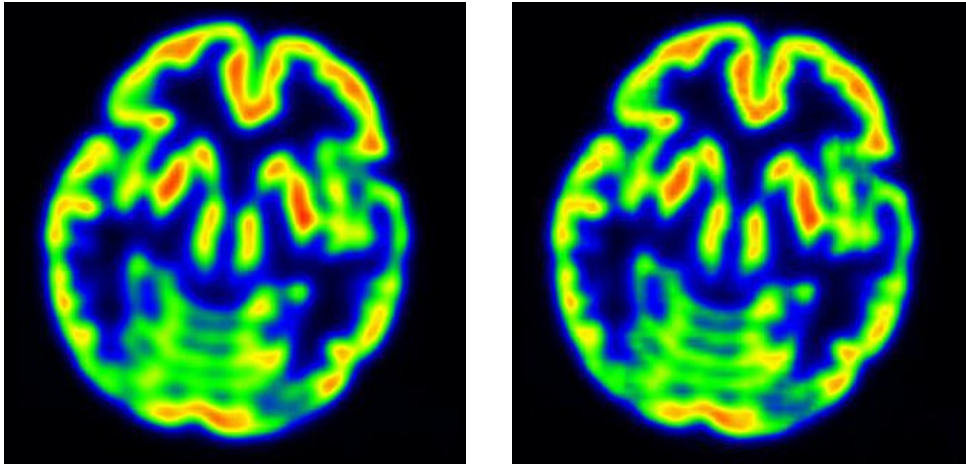
Taulukko 8: Aivofantomimallilla suoritettujen kuvausten parametriyhdistelmät.

Yhdistelmän nimi	Säde a	Parametri α
Oletusparametrit	2,8	6,3716
Kuula 1	2,8	16,1456
Kuula 2	2,5	7,6879
Kuula 3	2,3	4,6049
Dynaaminen kuula	2,3	4,1338

$\alpha = 6,3716$, joten simulaatiossa käytettyjä säteitä $a = 3,2$ ja $a = 4,0$ ei testattu. Myös dynaamisen kuulan parametrit ovat valmistajan oletusparametreja. Relaksaatioparametria λ ei päästy testaamaan, sillä Philipsin tapa käsitellä RAMLA-algoritmin relaksaatioparametria ei ollut eksplisiittisesti ilmoitettu.

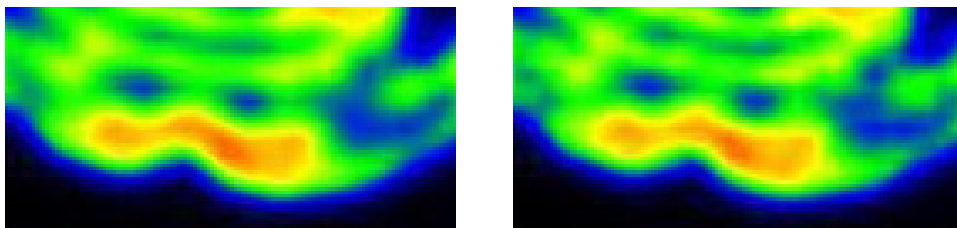
5.3 Tulokset

Kuvadatasta kävi ilmi, että pysyteltäessä lähellä Philipsin oletusasetuksia kuulien parametrien muutoksilla ei ollut juurikaan vaikutuksia kuvanlaatuun (kuva 31). RAMLA-algoritmin suppenemisen tarkastelemiseksi tehtiin vielä iteraatiot 10 ja 30 eri parametriyhdistelmillä ja huomattiin, että iteraatioiden 10 ja 30 välillä ei ole enää eroa. Kiinnostavimmat mahdolliset erot eri parametriyhdistelmien välillä löydetään siis jo kymmenellä iteraatiolla. Kuvassa 32 on havainnollistettu suuremman parametrin α vaikutusta. Kuva tulee tarkemmaksi; matalan aktiivisuuden siniset alueet ovat selvärajaisempia ja

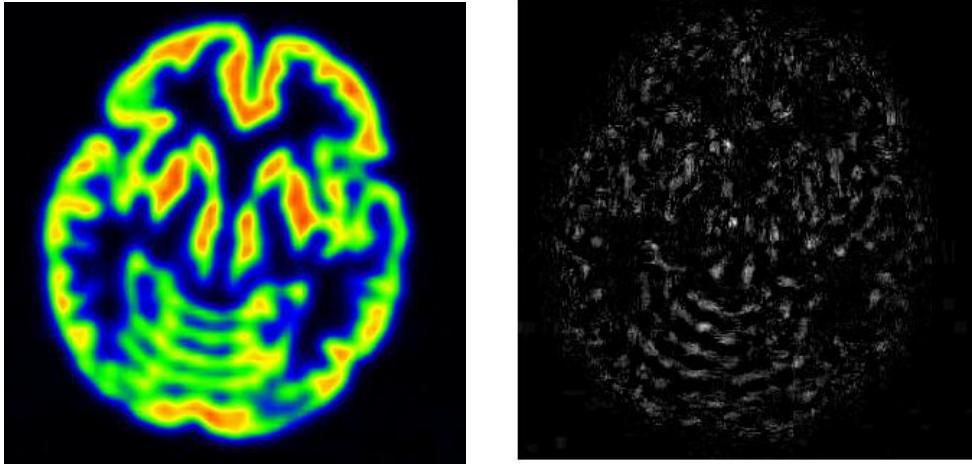


Kuva 31: Vasemmalla oletusarvoinen kuva ($a = 2,8; \alpha = 6,317$), ja oikealla kuula 1 ($a = 2,8; \alpha = 16,1456$). Iteraatiomäärät molemmissa 2. Selkeää eroa ei ole havaittavissa.

tummempia. Erot eivät ole suuria korkean aktiivisuuden kuvissa, sillä rekonstruktioparametrien erojen tuottama vaihtelu jäänee skannerin mittausprosessissa tuottaman virheen alle. Parametreja olisi tullut tutkia laajemmalla skaalalla suurten erojen löytämiseksi. Kuvassa 33 on vielä esitetty kuula 3, sekä erotuskuva oletusparametrein rekonstruoituun kuvaan. Kuvan valkeilla alueilla kuulalla 3 rekonstruoitu kuva on suurempi kuin oletusparametrein rekonstruoitu kuva, mikä vastaa edellisessä luvussa havaittua pienillä säteillä esiintyvää ylikorostusta.



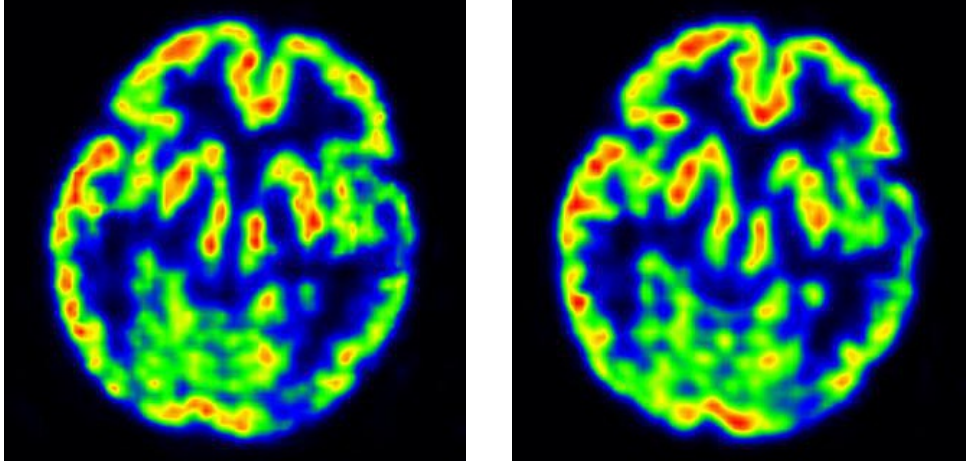
Kuva 32: Vasemmalla oletusarvoinen kuva ($a = 2,8; \alpha = 6,317$), ja oikealla kuula 1 ($a = 2,8; \alpha = 16,1456$). Kuvat tarkennettu leikkeen alareunaan. Iteraatiomäärät molemmissa 10. Oikean puolen kuvassa on hieman tarkempi kontrasti.



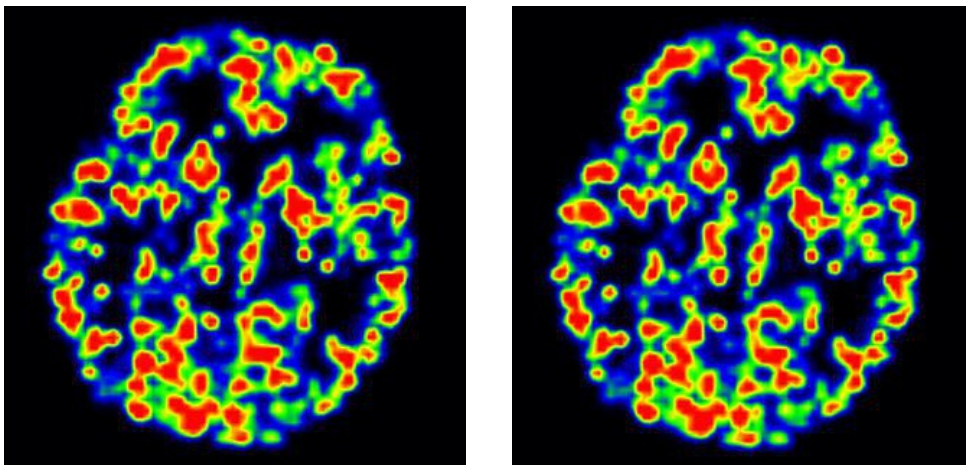
Kuva 33: Vasemmalla kuulalla 3 ($a = 2,3; \alpha = 4,6049$) rekonstruoitu kuva, ja oikealla kuulalla 3 ja oletusparametrein rekonstruoidun kuvan erotus. Erotuskuvassa on havaittavissa pientä yliestimointia kuulalla 3.

Dynaamisissa kuvissa lisämuuttujana on yksittäisen kuvausjakson pituus. Kuvausjakson lyhetessä yksittäisten havaintojen merkitys kasvaa, mistä johtuen satunnaisuudella on isompi vaikutus kerättyyn dataan. On oletettavaa, että kuulun säteen pientyessä ja parametrin α arvon kasvaessa rekonstruktion tarkkuus heikkenee, sillä kuulat eivät mene tarpeeksi päällekkäin yksittäisten poikkeavien havaintojen vaikutuksen vähentämiseksi. Kuvassa 34 on esitetty kaksi rekonstruktiota, jotka tuottavat ymmärrettävän kuvan. Iteraatiomääränä on nyt vain 2, sillä dynaamisen kuvasarjan rekonstruoiminen on paljon työläämpää kuin yksittäisen staattisen kuvan; jokainen dynaamisen kuvasarjan kuvausjakso rekonstruoidaan kuten yksittäinen staattinen kuva. Kuula 1 on säteeltään suurempi, mutta vastaavasti myös sen parametrin α arvo on suurempi. Parametrin α muutoksen vaikutus näyttää olevan voimakkaampi kuin kuulien säteissä oleva ero. Molemmat kuvat häviävät kuvanlaadussa selkeästi korkean aktiivisuuden kuville.

Skannerin dynaamisten kuvausten oletusparametreilla rekonstruoitu kuva ei rekonstruoinut kunnolla testatuilla kuvausjaksojen pituuksilla (kuva 35). Käytetyn kuulun säde on edellä kuvattuja onnistuneita rekonstruktioita pienempi, vain 2,3. Lisäksi dynaaminen kuula ei sijaitse aivan kuvan 13 käyrillä,



Kuva 34: Vasemmalla kuulaa 1 ($a = 2,8; \alpha = 16,1456$) tuottama kuva, ja oikealla kuulalla 2 ($a = 2,5; \alpha = 7,6879$) rekonstruoitu kuva. Kuula 2 tuottaa pienemmästä säteestään huolimatta tasaisempaa kuvanlaatua. Pieni parametrin α arvo tuottaa laajoja kuulia, jotka ovat robusteja matalan aktiivisuuden kuvissa. Iteraatiomäärä on 2.



Kuva 35: Vasemmalla dynaamisen kuulaa ($a = 2,3; \alpha = 4,1338$) tuottama kuva kahdella iteraatiolla, ja oikealla iteraatiomäärä on 20. Kuva on suurilta osin konvergoitunut jo kahden iteraation jälkeen. Väärän parametrivalinnan seurauksena rekonstruktion voidaan sanoa hajonneen, ja kuva on konvergoitunut väärään lopputulokseen.

vaan on hieman niiden alapuolella. Tämä tuo lisävirhettä kuvaan. Kuvassa kirkkaat alueet vuorottelevat lähes olemattoman aktiivisuuden kanssa, mikä ei ole anatomisen esityksen kannalta tyydyttävää. Dynaamisessa rekonstruktiossa parametrien valinta on selkeästi tärkeämpää kuin korkean aktiivisuuden staattisissa kuvissa.

6 Yhteenveto

Kuulien käyttö rekonstruktiossa antaa käyttäjälle valinnanvaraa eri ominaisuuksien korostamisen suhteen. Tässä tutkielmassa ei erikseen tarkasteltu vokselirekonstruktioita, mutta ne ovat luonteeltaan jäykempiä, sillä ainoa säädeltävä vokseliominaisuus on hilan rakenne. Kuulien päällekkäisyyttä ja muotoa muuttamalla saadaan käytännössä loputtomasti erilaisia yhdistelmiä testattavaksi.

Kaikki kuulien parametrien yhdistelmät eivät kuitenkaan ole järkeviä. Kuulan säteen tulee olla tarpeeksi iso, jotta päällekkäisyyttä esiintyy. Lisäksi kuulien parametrit tulee valita sopivasti, jotteivat kuulien arvot Fourier-avaruudessa vuoda viereisiin kuuliin. Tätä ehtoa varten tässä tutkielmassa jatkettiin artikkelissa [18] johdateltua teoriaa. Artikkelissa johdetun empiirisen lähestymistavan sijasta laskettiin analyyttiset käyrät (kuva 13), joilta parametrien valinta tulee Fourier-ehdon mukaan suorittaa. Nämä hyperbelit vastaavat kysymykseen: "mitä arvoja parametreille voidaan antaa?". Hyperbelien johtaminen täyttää siten tutkielman alussa esitetyn tavoitteen sopivien parametrijoukkojen löytämisestä.

Kuulien parametrien simulaatiotestaus paljasti kompromissien teon välttämättömyyden parametrien valinnassa. Yhtä oikeaa yhdistelmää ei ole löydettävissä, vaan parametrit tulee valita siten, että kulloisenkin kuvauksen kannalta tärkeimmät ominaisuudet saadaan riittävän hyväksi ilman muiden ominaisuuksien liiallista heikkenemistä. Mielenkiintoalueiden keskiarvoja tarkastellessa mahdollisimman suuri säde a ja mahdollisimman pieni muotoparametrin α arvo tuottavat parhaan tuloksen. Hajontamielessä taas kuulan pieni säde ja suuri parametrin α arvo tuottavat parhaat tulokset kuumilla alueilla, kun taas tummilla alueilla suuri säde ja pieni α tuottaa vähäisen hajonnan.

Visuaalisessa tarkastelussa suuret säteen a arvot tuottavat reunavirhettä kuvaan, kun taas pienemmällä säteellä suoritettut rekonstruktiot ovat tasaisempia referenssikuvan tasaisilla alueilla. Lisäksi profiilikäyrien tarkastelussa suuret säteen arvot tuottavat selkeän reunavirheen, kun taas pienet säteen arvot yliestimoiivat kuvaa sen koko laajuudella. Suurella säteen a arvoilla

havaittavat ongelmat johtuvat osaltaan myös puoliarvovevyyshdon rikkoutumisesta. Koko kuvan laajuisia virheitä tarkasteltaessa huomattiin säteen kasvun yleisesti vähentävän virhettä, mutta parametrin α arvon vaikutus oli erilainen eri säteen arvoilla. Kompromissina kaikkien eri ominaisuuksien suhteen voidaan pitää sädettä 3,2 valiten parametrin α hyperbeliltä 3 ja mahdollisesti myös sädettä 2,5 valiten parametrin α hyperbeliltä 2 tai 3. Näin valiten kuvissa esiintyy vain lievää yliestimointia, hajonta ei kasva liian suureksi, kuvan visuaaliset sekä profiilikäyrien ominaisuudet ovat tyydyttävät, eikä puoliarvovevyyshähtoa juurikaan rikota.

Anatomisella aivomallilla suoritetuissa rekonstruktioissa ei ollut juurikaan havaittavissa eroja korkean aktiivisuuden staattisilla kuvilla. Yhtenä selityksenä voinee olla oikeassa kuvauksessa esiintyvä kohina, joka saattaa vähentää parametrijhdistelmien välisiä eroja simulaatioissa käytettyjä täysin tasaisia alueita paremmin. Simulaatioiden suorittaminen kohinaisella datalla on mielenkiintoinen jatkotutkimuksen kohde.

Päästäessä dynaamiseen, lyhyempien kuvausjaksojen tilanteeseen parametrijhdistelmien väliset erot alkoivat näkyä. Mittauksessa saatu vähäinen aineisto suosii mahdollisimman robusteja kuulia, jolloin suuri säde ja pieni parametrin α arvo tuottavat parhaat rekonstruktiot. Liian pienen säteen vaikutus kuvanlaatuun on dramaattinen, eikä kuvaa pahimmassa tapauksessa voida käyttää potilastutkimuksiin lainkaan. Robustisuuden vaatimus dynaamisissa tutkimuksissa on niin tärkeä, että staattisiin ja dynaamisiin tilanteisiin tulisikin käyttää omia, kyseiseen tarkoitukseen optimoituja, parametrijhdistelmiä.

Simulaatiossa myös tarkasteltu RAMLA-algoritmin relaksaatioparametrin λ vaikutus jäi pieneksi. Pienet parametrin λ arvot aikaansaavat joissain tilanteissa kuvan liian aikaisen konvergoitumisen, mistä johtuen kuvaan voi jäädä virhettä. Liian matalan parametrin λ arvon valinta johtaa epäonnistuneeseen kuvaan, jonka ominaisuudet riippuvat paljon kuulien lähtöarvosta. Pienemmät parametrin λ arvot antoivat suurimmillaan 2 % pienemmän yliestimointimisen kuvan kirkkailla alueilla, mutta pärjäisivät huomattavasti huonommin tummilla alueilla. Parametrin λ arvo on myös sidoksissa siihen, montako projisiosuoraa keskimäärin kulkee yksittäisen kuulan läpi. Mitä suurempi tämä

luku on, sitä pienemmäksi parametrin λ voi valita kuvanlaadun kärsimättä. Simulaatioissa käytetty geometria oli vain yksittäistapaus, joten johtopäätöksiä oikeassa skannerissa käytettävään parametrin λ arvoon ei voida antaa, vaan optimointi ja valinta tulee tehdä skannerikohtaisesti.

Luvussa 3 johdetun teorian pohjalta saatiin valittua parametrijhdistelmiä, jotka täyttävät Fourier-ehdon. Parametrijhdistelmillä saatiin toisistaan eroavia rekonstruktioita, joista voitiin päätellä eri parametrien vaikutusta kuvanlaatuun. Oikean kohinaisen datan tarkastelulla suuria eroja kuulien välillä ei löydetty korkean aktiivisuuden kuvissa. Simulaatiomallin jatkotutkimuksilla voidaan selvittää, johtuuko ilmiö oikeassa mittauksessa olevasta kohinasta. Dynaamisen kuvadatan rekonstruktiossa havaittiin rekonstruktion hajoaminen liian pienen säteen vaikutuksesta. Tutkielman tavoitteet saavutettiin ja kliiniseen tutkimustyöhön kyettiin johtamaan käyttökelpoisia parametrijhdistelmiä.

Kirjallisuutta

- [1] S. Alenius, *On Noise Reduction in Iterative Image Reconstruction Algorithms for Emission Tomography: Median Root Prior* Väitöskirja, Tampereen teknillinen korkeakoulu, 1999.
- [2] A. Averbuch Y. Shkolnisky, *3D Fourier based discrete Radon transform* School of Computer Science, Tel Aviv University, Tel Aviv, Israel, 2002.
- [3] B. Bendriem, D. W. Townsend, *The Theory and Practice of 3D PET*, Kluwer Academic Publishers, Dordrecht, Alankomaat, 1998.
- [4] J. J. Benedetto, G. Zimmermann, *Sampling multipliers and the Poisson summation formula*, The Journal of Fourier Analysis and Applications, Volume 3, Number 5, 505-523, 1997.
- [5] J. Browne, A. R. De Pierro, *A Row-Action Alternative to the EM Algorithm for Maximizing Likelihoods in Emission Tomography* IEEE Transactions on Medical Imaging, VOL. 15, NO. 5, 1996.
- [6] P. Dirac, *The principles of quantum mechanics*, Oxford University Press, Oxford, 1930.
- [7] Z. Hu, W. Wang, E. E. Gualtieri, Y. I. Hsieh, J. S. Karp, S. Matej, M. J. Parma, C. H. Tung, E. S. Walsh, M. Werner, D. Gagnon, *An LOR-based Fully-3D PET Image Reconstruction using a Blob-basis Function* IEEE Nuclear Science Symposium Conference Record, 2007.
- [8] H. M. Hudson, R. S. Larkin, *Accelerated image reconstruction using ordered subsets of projection data*, IEEE Trans. Medical Imaging, 13 (4), 601609, 1994.
- [9] H. Iida, et al, *Three-dimensional brain phantom containing bone and grey matter structures with a realistic head contour*, Ann Nucl Med 27:25-36, 2013.

- [10] F. Jacobst, S. Matej, R. M. Lewitt, *Image Reconstruction Techniques for PET*, ELIS Technical Report R9810, MIPG Technical Report MIPG24, 1998.
- [11] J. Johansson, *Radon muunnoksen diskretoinnista 3. aine*, Turun yliopisto, 2006.
- [12] J. Johansson, *Tomografiamenetelmän käänteisongelman ratkaisusta* Pro gradu -tutkielma, Matematiikan laitos, Turun yliopisto, 2007.
- [13] A. C. Kak, M. Slaney, *Principles of Computerized Tomographic Imaging*, New York, NY, IEEE Press, 1988.
- [14] R. Klén, *PET Basics*, Turun Yliopisto, 2005.
- [15] R. M. Lewitt, *Multidimensional digital image representations using generalized Kaiser-Bessel window functions*, J. Opt. soc. Am. Vol 7, No.10, 1990.
- [16] R. M. Lewitt, *Alternatives to voxels for image representation in iterative reconstruction algorithms*, Phys. Med. Biol., Vol 37, No 3, 1992.
- [17] S. Matej, R. M. Lewitt, *Efficient 3D Grid for Image Reconstruction Using Spherically-Symmetric Volume Elements* Nuclear Science Symposium and Medical Imaging Conference, IEEE Conference Record, 1994.
- [18] S. Matej, R. M. Lewitt, *Practical Considerations for 3-D Image Reconstruction Using Spherically Symmetric Volume Elements*, IEEE Transactions on Medical Imaging Vol 15, No 1, 1996.
- [19] M. Mäkelä, *Konvekssi analyysi ja optimointi*, luentomoniste, Turun Yliopisto, 2012.
- [20] M. Schabel, *3-D Shepp-Logan Phantom*, MathWorks File Exchange, File ID 9416, 2006.
- [21] L. A. Shepp, Y. Vardi *Maximum Likelihood Reconstruction for Emission Tomography*, IEEE Transactions on Medical Imaging, Vol. MI-1, No. 2, 1982.

- [22] J. Teuvo, J. Johansson, J. Linden, V. Saunavaara, T. Tolvanen, M. Teräs, *Specification and estimation of sources of bias affecting neurological studies in PET/MR with an anatomical brain phantom*, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Available online 10 September 2013, ISSN 0168-9002, <http://dx.doi.org/10.1016/j.nima.2013.09.002>.
- [23] T. Uygur, *Cosine Domain Gap-Filling for the PET Sinograms*, Progradu tutkielma. Tampereen Yliopisto, 2008.
- [24] Y. Vardi, L. A. Shepp, L. Kaufman, *A Statistical Model for Positron Emission Tomography* *Journal of the American Statistical Association*, Vol 80, No. 389, 1985.

7 Liite I, rekonstruktioparametrien valintaan liittyvä koodi.

Alla esiteltävien funktioiden avulla voidaan valita rekonstruktiossa käytettävät Fourier-ehdon täyttävät parametrit. Kuvan 13 käyrät voidaan piirtää laskemalla funktioiden `giveRadius` ja `giveAlpha` avulla riittävästi pistepareja, jotka sitten yhdistetään toisiinsa.

```

Funktio giveRadius
function f=giveRadius(alpha,valley)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funktio ottaa syötteenä parametrin alfa arvon, sekä
hyperbelin
% järjestysnumeron
% Funktio palauttaa säteen a arvon, joka vastaa annettua
% parametrin alfa arvoa parametrin valley määrittämällä
% hyperbelillä.
% Syötteitä koskevat rajoitteet
% alpha>0
% valley = 1,2,3,...,10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%alustetaan nollakohdat Taulukosta 1 saaduilla arvoilla
nollat=[6.987932000500520;
10.41711854737936;
13.69802315324925;
16.92362128521384;
20.12180617445382;
23.30424698893965;
26.47676366453913;
29.64260454031581;
32.80373238519611;
35.96140580470903];

%ratkaistaan säde yhtälön (3.13) viimeisestä muodosta,
% ja sijoitetaan parametrit kaavaan.
f=sqrt(alpha^2/(2*pi)^2+(nollat(valley)/(2*pi))^2);

```

```

Funktio giveAlpha
function f=giveAlpha(a,valley)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funktio ottaa syötteenä parametrin säde a arvon, sekä
% hyperbelin järjestysnumeron
% Funktio palauttaa muotoparametrin alfa arvon, joka vastaa
% annettua säteen a arvoa parametrin valley määrittämällä
% hyperbelillä
% Syötteitä koskevat rajoitteet
% a>0
% valley = 1,2,3,...,10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%alustetaan nollakohdat Taulukosta 1 saaduilla arvoilla
nollat=[6.987932000500520;
10.41711854737936;
13.69802315324925;
16.92362128521384;
20.12180617445382;
23.30424698893965;
26.47676366453913;
29.64260454031581;
32.80373238519611;
35.96140580470903];

%ratkaistaan alfa yhtälön (3.13) viimeisestä muodosta,
% ja sijoitetaan parametrit kaavaan. Mikäli säde on liian
% pieni, jotta vastaavaa parametrin
%alfa arvoa voidaan määrittää, palautetaan 0
f=sqrt(max(0,(a*2*pi)^2-nollat(valley)^2));

```