Del II

Iterative metoder til rekonstruktion af PET billeder

Indholdsfortegnelse

Kapitel 1

Indledning

Tre-dimensional opsamling af PET data, og efterfølgende rekonstruktion med tre-dimensionale rekonstruktions algoritmer, vil give en langt højere følsomhed med hensyn til detekterede henfald. Dette kan i PET udnyttes på flere måder — der kan rekonstrueres billeder med bedre tælletalsstatistik i sinogrammet, strålingsdosen til patienten kan nedsættes, eller skanningstiden forkortes.

Denne del af projektet vil omhandle metoder til rekonstruktion af PET data opsamlet i tre dimensioner. Der tages udgangspunkt i de to-dimensionale iterative rekonstruktionsteknikker, da disse er anvendelige på vilkårlige data opsamlings geometrier. Diskussion og implementering af disse, vil af praktiske årsager blive holdt i to dimensioner, parallelt med de i den første del af rapporten omtalte algoritmer.

Der er blevet implementeret tre forskellige iterative rekonstruktions algoritmer, og implementeringen af disse vil blive diskuteret. Algoritmerne vil blive testet med kunstigt generede data. En sammenligning af de implementerede rekonstruktions algoritmers rekonstruktionskvalitet og rekonstruktionshastighed vil blive foretaget. En sammenligning med de i den første del af projektet udviklede direkte metoder, vil også blive foretaget. Algoritmerne er ikke blevet testet med rigtige data, da det ikke var praktisk muligt arbejde med den størrelse sinogrammer der optages med Rigshospitalets PET skanner.

1.1 Skanner geometrier

Der er igennem tiden konstrueret mange forskellige typer skannere, med mange forskellige detektorgeometrier og opsamlingsmetoder. Hver geometri vil have specielle fysiske karakteristika, og problemer med hensyn til billedrekonstruktion. Der henvises til Townsend [Tow94], Budinger [Bud79] og Brooks [Bro81] for en gennemgang af forskellige typer karakteristiske skanner geometrier og data opsamlingsmetoder. General Electric *Advance* PET skanneren fra på Rigshospitalet (se figur 1.1) er en såkadt *multi-slice* skanner, hvor detektorerne er ordnet i flere ringe. Diskussionen af tre-dimensionale skanner geometrier, vil derfor blive begrænset til denne type multi-slice skannere.

Ved brug af standard to-dimensionale rekonstruktionsmetoder, vil PET skannere sjældent nå deres fulde opløsningspotentiale, grundet statistiske begrænsninger af de opsamlede emissions data (få tælletal i sinogrammet). Fremskridt i PET teknologien har ført til udviklingen af systemer med mindre detektorer, der giver større spatial opløsning. For at udnytte denne opløsningsforbedring fuldt ud skal antallet af detekterede sammenfald forøges med tre gange den radiale opløsningsforbedring. Ud over dette, er følsomheden i akse-retningen proportional med kvadratet på skivetykkelsen (se bla. Cherry [Che91]). Hvis en opløsningsforbedring skal

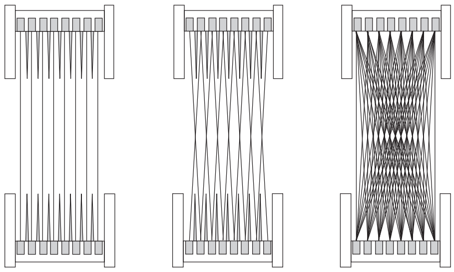
Figur 1.1: General Electric *Advance* PET skanner.

udnyttes fuldt ud, er det derfor nødvendigt at finde metoder, hvormed der kan kompenseres for effektivitetstabet forbundet med denne. For studier, hvor statistikken af emissions-skannet ikke er et problem, kan den øgede følsomhed bruges til at mindske strålingsdosen givet til patienten eller til at nedsætte skanningstiden.

De fleste kommercielle multi-slice PET skannere er begrænsede til at optage sammenfald mellem detektorer der ligger i samme plan (*direkte planer*) som på figur 1.2a, og detektorer der ligger i tilstødende ringe (*kryds-planer*) som på figur 1.2b, når der opereres i to-dimensional mode. Data fra et kryds-plan bliver normalt brugt til at rekonstruere en skive, der ligger i området mellem de to tilstødende detektorringe, den er optaget med. Disse projektioner, fra de direkte planer og kryds planerne, bliver derefter rekonstrueret med en rutine baseret på to-dimensional filtreret tilbageprojektion (se rapportens del I). Multi-ring skannere er derfor i stand til at producere et tre-dimensionalt volumen billede som en kombination af en række uafhængige to-dimensionale skiver. Hver af disse skiver er fysisk afgrænset af de imellem detektorringene placerede skillevægge, *septa*[1]. Disse er designet således, at de dæmper maksimalt for scatter og henfald, der ikke sker i planerne, samtidig med at de tillader sammenfaldslinier imellem kryds-planerne. Ydermere afskærmer de detektorerne for aktivitet uden for det aktuelle interesseområde. Denne opbygning af en PET scanner vil, ifølge Cherry [Che91], begrænse den procentdel af henfald der detekteres, til under 0,5% i de direkte planer, og under 0,75% i kryds planerne. Når skillevæggene fjernes fra en almindelig multi-ring PET scanner som på figur 1.2c, vil skanneren fremstå geometrisk som et cylindrisk array af diskrete detektorer, der i princippet vil være i stand til at fremstille tre-dimensionale billeder, da alle sammenfaldslinier mellem detektorerne i de forskellige planer kan udnyttes. Ud over at forøge følsomheden vil fjernelsen af skillevæggene også have andre konsekvenser. Disse inkluderer:

- Den forøgede andel af enkelt-detektioner[2], stammende fra henfald både indenfor og udenfor interesse området, vil føre til et øget antal forkerte sammenfalds detektioner og potentielle dødtidsproblemer,
- Den grove sampling i akse-retningen, kan medføre artifakter i det rekonstruerede billede,
- Da skillevæggene er fjernet, vil andelen af scatter øges, da der også registreres scatter uden for de normale planer,

[1]septum — skillevægge mellem detektorringene, kan være konstruerede som tilspidsede blyskiver.
[2]Kun én detektor rammes dvs. ingen sammenfald registreres.



Figur 1.2: Forskellige konfigurationer af en otte rings multi-ring skanner. a) Direkte planer med 8 sammenfalds planer. b) Kryds-planer med 7 sammenfalds planer. c) Fuld tre-dimensional samplings konfiguration, med 64 mellem-planer for et 8 rings system.

- Inkluderingen af andre sammenfalds linier, med en ikke uniform densitet, vil føre til billeder med ikke uniforme støj egenskaber.

For at få fuld udnyttelse af den øgede effektivitet, skal billed-rekonstruktionen foretages tre-dimensionalt. Eksperimentielle studier med ægte tre-dimensional rekonstruktion har vist, at effektiviteten som forventet vil stige mange gange, dog vil antallet af enkelt-detektioner og andelen af scatter ligeledes stige (se bla. Colsher [Col80] og Dahlbom [Dah89]).

Nyere skannere er i stand til automatisk at fjerne skillevæggene afhængigt af, om der skal opsamles data i to- eller tre-dimensional konfiguration.

1.2 Tre-dimensionale rekonstruktionsmetoder

Den mest anvendte ægte tre-dimensionale rekonstruktions algoritme i kommercielle skannere idag, er en tre-dimensional filtreret tilbageprojektions algoritme, der oprindeligt blev udviklet af Colsher [Col80]. Den metode der bruges i praksis i dag, er udviklet af Kinahan og Rogers [Kin89], og er en videreudvikling af Colshers algoritme, baseret på re-projektion af de projektionsvinkler der ikke kan måles. Metoden bliver gerne kaldt for *Kinahan & Rogers* rekonstruktion. Det er denne metode Rigshospitalets skanner bruger til tre dimensional rekonstruktion.

De iterative metoder til tre-dimensional rekonstruktion er ikke forskellige fra de to-dimensionale, hvor der gerne bruges ART (Gordon, [Gor70]) eller expectation maximization metoder (Shepp og Vardi, [She82]).

1.3 Iterative metoder

Inden for projektets tidsmæssige rammer har det ikke været muligt at nå til egentlig tre-dimensional rekonstruktion, men som det vil blive vist, er den teori der skal anvendes ved to- og tre-dimensional iterativ rekonstruktion den samme, blot er kompleksiteten større ved tre-dimensional rekonstruktion, med hensyn til tidsforbrug og i høj grad hukommelse. Derfor vil det følgende kun omhandle to-dimensional iterativ rekonstruktion, men vil uden problemer kunne anvendes til tre dimensioner.

Den første CT skanner opfundet af Hunsfield, brugte en series expansion metode, praktisk taget lig ART[3] til at rekonstruere billeder med. I dag bruger alle kommercielt tilgængelige

[3] Se kapitel 3.

skannere filtrerede tilbageprojektions metoder til rekonstruktionen. Spørgsmålet er, hvorfor de iterative metoder overhovedet skal studeres?

De iterative metoder er generelt langsommere end transformationsmetoderne til at rekonstruere billeder, da de er kendt for at konvergere langsomt. Det kan siges, at tiden arbejder for de iterative rekonstruktionsmetoder, da fremtidig computer teknologi vil gøre det muligt at forøge beregningshastigheden så meget, at transformationsmetodernes fordele frem for de iterative vil mindskes. De iterative metoder har på flere punkter bedre egenskaber end transformationsmetoderne, disse inkluderer

- Fleksibilitet med hensyn til opsamlingsgeometri,
- Evne til at rekonstruere objekter med høj kontrast,
- Rekonstruktion fra få eller ukomplette projektioner.

Den vigtigste grund er de iterative metoders fleksibilitet. De er nemme at implementere til en vilkårlig detektorgeometri, da det er den samme løsningsmetode der bruges uanset opsamlingsgeometrien for skanneren. Den eneste forskel på praktisk implementering af de iterative metoder ved forskellige data opsamlings geometrier er udregningen af en *transformations matrix*[4] der svarer til den aktuelle detektorgeometri, enten to- eller tre-dimensionalt. Direkte metoder til rekonstruktion baserer sig på digitale filtre der gør, at punktkilder i rummet vil få en vis rumlig udstrækning. Det gør de iterative metoder ikke automatisk, og vil derfor være i stand til at bedre at rekonstruere billeder med høj kontrast. Dette er f.eks. nyttigt i forbindelse med industrielle anvendelser af CT på f.eks. maskindele med meget veldefinerede kanter.

1.4 Resumé

I dette kapitel er tre-dimensional rekonstruktion med multi-slice PET skannere blevet præsenteret. De iterative rekonstruktionsmetoders egenskaber er blevet beskrevet, og det er blevet valgt at undersøge disse i denne del af rapporten.

[4] Se kapitel 2.

Kapitel 2

Matrix beskrivelse af rekonstruktion

I dette kapitel, vil billedrekonstruktionen modelleret som et lineært ligningssystem blive introduceret. Problemer i forbindelse med denne modellering vil blive beskrevet. Forskellige implementeringsmæssige problemstillinger vil blive diskuteret.

2.1 Det lineære ligningssystem

Et generelt lineært ligningssystem på formen

$$\begin{array}{ccccccccc} a_{0,0}x_0 & + & a_{0,1}x_1 & + & \dots & + & a_{0,(N-1)}x_{N-1} & = & b_0 \\ a_{1,0}x_0 & + & a_{1,1}x_1 & + & \dots & + & a_{1,(N-1)}x_{N-1} & = & b_1 \\ a_{2,0}x_0 & + & a_{2,1}x_1 & + & \dots & + & a_{2,(N-1)}x_{N-1} & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{(M-1),1}x_0 & + & a_{(M-1),2}x_1 & + & \dots & + & a_{(M-1),(N-1)}x_{N-1} & = & b_{M-1} \end{array} , \qquad (2.1)$$

beskriver hvorledes N ubekendte x_n, $n = 0, 1, \dots, N-1$ indbyrdes relaterer med M ligninger. Koefficienterne $a_{m,n}$ med $m = 0, 1, \dots, M-1$ og $n = 0, 1, \dots, N-1$ er kendte størrelser, ligesom værdierne b_m på den højre side.

Hvis $M = N$, vil der være lige så mange ligninger som ubekendte, og der er en chance for, at ligningssystemet kan løses for et unikt sæt af x_n'er. Der er dog flere omstændigheder der kan gøre, at der analytisk ikke kan findes en unik løsning. Hvis en eller flere af de M ligninger er en lineær kombination af andre, vil der optræde en tilstand kaldet *række-degeneration*. Hvis alle ligningerne indeholder variable i den eksakt samme lineære kombination kaldes dette *søjle-degeneration*. Et ligningssystem der er degenereret kaldes singulært. Er $M < N$, er systemet *underbestemt*, med færre ligninger end ubekendte, og der vil enten være ingen, eller mere end én løsning til systemet. Er $M > N$, er systemet *overbestemt* med flere ligninger end ubekendte, og der er normalt ingen eksakt løsning til systemet. Som kompromisløsning bruges ofte den løsning, der ligger tættest på at opfylde alle ligningerne samtidig.

Løsningen af et ligningssystem som i formel (2.1) kan, så længe M og N er i størrelsesorden op til nogle hundrede, foretages med de standard metoder der eksisterer til løsning af almindelige lineære ligningssystemer, såsom Gauss elimination eller singulær værdi dekomposition. De problemer, der forekommer ved løsning af sådanne, vil gerne være af numerisk natur, da afrundingsfejl kan degenerere ligningssystemet undervejs.

Løsningen af meget store problemer, hvor M og N er nogle tusinde eller flere, kan i praksis ikke foretages med standard metoderne. Hvis koefficienterne er sparse[1], kan der dog bruges

[1]sparse — sparsom, kun få koefficienter med en værdi forskellig fra nul. Se afsnit 2.5.

iterative metoder udviklet til at tage hensyn til dette.

2.1.1 Matricer

Ved hjælp af matricer, kan formel (2.1) skrives på matrix-vektor form som

$$\mathbf{Ax} = \mathbf{b}. \tag{2.2}$$

Koefficienterne i ligningssystemet, kan skrives som indholdet af koefficient matricen $\mathbf{A}$, som

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,(N-1)} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,(N-1)} \\ a_{2,0} & a_{2,1} & \cdots & a_{2,(N-1)} \\ \vdots & \vdots & & \vdots \\ a_{(M-1),0} & a_{(M-1),1} & \cdots & a_{(M-1),(N-1)} \end{bmatrix}, \tag{2.3}$$

og de to vektorer $\mathbf{x}$ og $\mathbf{b}$ som

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{(N-1)} \end{bmatrix} \quad \text{og} \quad \mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{(M-1)} \end{bmatrix}. \tag{2.4}$$

For matricen $\mathbf{A}$, vil $\mathbf{a}_m$ betegne den m'te række som en vektor. En matrix-vektor multiplikation som i (2.2), vil for $\mathbf{b} \in \mathbf{R}^M$, $\mathbf{x} \in \mathbf{R}^N$ og $\mathbf{A} \in \mathbf{R}^{M \times N}$ være defineret som

$$\mathbf{b} = \mathbf{Ax} \quad \Rightarrow \quad b_m = \sum_{n=0}^{N-1} a_{m,n} x_n, \qquad m = 0, 1, \ldots, M-1. \tag{2.5}$$

Det indre produkt mellem to vektorer betegnes $[\cdot]$, og vil for hver $\mathbf{a}, \mathbf{b} \in \mathbf{R}^N$ være

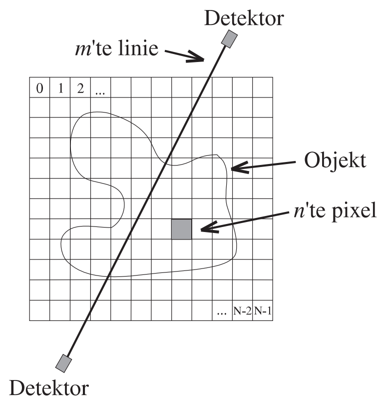
$$\mathbf{a} \cdot \mathbf{b} = \sum_{n=0}^{N-1} a_n b_n. \tag{2.6}$$

Disse definitioner og terminologier vil blive brugt i de følgende afsnit af rapporten.

2.2 Modellering af billedrekonstruktion

Det problem der skal løses, er det generelle rekonstruktionsproblem (se del I af rapporten). Det objekt der skal måles, er et tre-dimensionalt objekt hvis indre struktur er ukendt. I PET forsøges denne estimeret ved at injicere en radioaktiv emitter ind i blodet. Den emitterede stråling måles langs en række diskrete sammenfaldslinier mellem skannerens detektorer. Fra disse målinger skal distributionen af den fysiske parameter f.eks. hjerneaktivitet der influerer på målingerne estimeres.

De iterative metoder tager udgangspunkt i en diskret model af det objekt der skal rekonstrueres. Et net af kvadratiske billedelementer indføres, således at de dækker hele objektet. Disse billedelementer eller pixels nummereres f.eks. fra toppen til bunden som på figur 2.1. Den søgte størrelse, dæmpningskoefficienten for CT, eller emissionsintensiteten for PET, antages at have en konstant værdi x_n over hele den n'te pixel[2] for $n = 0, 1, \ldots, N-1$. Detektorerne antages at være punkter, og de diskrete sammenfaldslinier modelleres som rette linier. Den m'te sammenfaldslinies længde gennem den n'te pixel betegnes $a_{m,n}$ for alle linier $m = 0, 1, \ldots, M-1$ og alle pixels $n = 0, 1, \ldots, N-1$. Det antages at denne vægt repræsenterer den n'te pixels andel i emissionen langs den m'te linie.



Figur 2.1: Diskretisering af det rekonstruerede objekt.

Den fysiske måling af emissionen langs den m'te sammenfaldslinie, modelleres som linieintegralet af den ukendte emissions funktion mellem de to detektorer der definerer den m'te linie. I den diskrete model vil dette linieintegrale blive en diskret sum, og hele modellen kan beskrives med ligningssystemet

$$b_m \approx \sum_{n=0}^{N-1} a_{m,n} x_n, \qquad m = 0, 1, \ldots, M-1. \tag{2.7}$$

Denne måde at beskrive rekonstruktionsproblemet på, er baseret på *series expansion* metoden (se f.eks. [Her80]), hvor den funktion f der skal rekonstrueres, approksimeres med en lineær kombination af et endeligt antal kendte basisfunktioner f_n i polære koordinater[3] (r, ϕ) som

$$f(r, \phi) \approx \sum_{n=0}^{N-1} x_n \, f_n(r, \phi). \tag{2.8}$$

Disse basisfunktioner kan f.eks. modelleres med nærmeste nabo approksimation som

$$f_n(r, \phi) = \begin{cases} 1 & \text{hvis } (r, \phi) \text{ tilhører den } n'\text{te pixel} \\ 0 & \text{ellers} \end{cases}, \tag{2.9}$$

hvor rekonstruktions algoritmen skal estimere de ukendte værdier for x_n. Hvis det antages at målingerne afhænger lineært af det objekt der skal rekonstrueres, så kan rekonstruktionen skrives som i formel (2.7), hvor $a_{m,n}$ er værdien af den m'te måling af den n'te basisfunktion.

På matrix form kan formel (2.7) skrives som

$$\mathbf{b} = \mathbf{A}\mathbf{x}, \tag{2.10}$$

hvor $\mathbf{b} \in \mathbf{R}^M$ er en vektor repræsentation af de aktuelle projektioner, $\mathbf{x} \in \mathbf{R}^N$ er billed-vektoren, og $M \times N$ matricen $\mathbf{A} \in \mathbf{R}^{M \times N}$ er *transformations* matricen. Dette er en matrix beskrivelse af

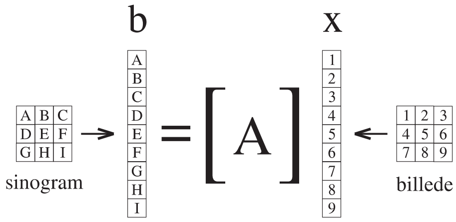
[2] Indekserne m og n, bruges i denne del af rapporten i en ny betydning.

[3] For ikke at sammenblande $f(x, y)$ og x_m.

den originale lineære Radon transformation som defineret i del I af rapporten. Sammenhængen mellem denne og formel (2.10), kan illustreres som

$$\begin{array}{ccccc} \check{f}(\rho,\theta) & = & \mathcal{R} & f(x,y) \\ \downarrow & & \downarrow & \downarrow \\ \mathbf{b} & = & \mathbf{A} & \mathbf{x} \end{array} \tag{2.11}$$

Den praktiske konvertering af de diskrete værdier for billedet f, og sinogrammet $\check{f}$ til vektorerne $\mathbf{x}$ og $\mathbf{b}$, kan i princippet foretages i henhold til en vilkårlig definition, blot transformations matricen $\mathbf{A}$ bliver udregnet i overensstemmelse med denne, da måden billederne omsættes til vektorer på ikke vil ændre på egenværdierne for $\mathbf{A}$. I denne rapport vil der blive brugt den i litteraturen mest gængse metode, hvor billederne nummereres startende fra øverste venstre hjørne, og gemmes linievis, som vist på figur 2.2.



Figur 2.2: Konvertering fra billeder til vektorer.

2.3 Valg af transformations matrix

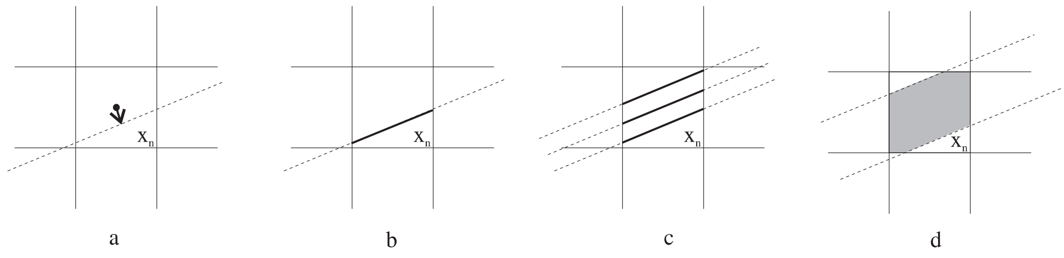
Hvorledes skal $a_{m,n}$ modelleres, dvs. sandsynligheden for at en emission i billedelementet n detekteres af detektor-par m? Sammenfaldslinien mellem de to detektorer kan som i forrige afsnit modelleres som en ideel linie mellem centrene af de to detektorer. Den ønskede sandsynlighed, kan derefter f.eks. findes som længden af denne linie igennem n.

Metoden hvormed transformations matricen genereres, vil have betydning for hvilke sæt af løsninger de iterative metoder vil konvergere imod. Der eksisterer mange metoder hvormed denne matrix kan dannes, der alle tager udgangspunkt i en mere eller mindre idealiseret model af de fysiske størrelser forbundet med detekteringen af positron-par.

I denne rapport vil der kun blive set på situationen, hvor billedelementerne i det rekonstruerede billede modelleres som kvadratiske pixels. I litteraturen er dog blevet studeret forskellige alternativer til denne approksimation med gode resultater (se f.eks. Lewitt [Lew92]).

Den simpleste metode at modellere transformationen på er, ved at undersøge om linien m krydser pixelen n, for alle m og n. Hvis linien krydser som på figur 2.3a, sættes n's andel i emissionen til Δx. Dette svarer til en simpel nærmeste nabo approksimation. Hvis sammenfaldslinierne antages at være ideelle linier som på figur 2.3b, kan der bruges *ray-tracing* eller *stråle-drevet* evaluering af transformations matricen hvor $a_{m,n}$ bliver længden af m igennem n for alle m og n. Denne model kan forbedres som vist på figur 2.3c og 2.3d, hvor der bruges oversampling hhv. integration over hele arealet mellem de to detektorer, til at estimere emissionsandelen for den n'te pixel.

Der er blevet implementeret en rutine til udregning af $\mathbf{A}$ matricen, der kan bruge nærmeste nabo, lineær interpolation eller stråle-drevet integration.



Figur 2.3: Forskellige måder at modellere pixelen x_n's værdi i rekonstruktionen med. a) Nærmeste nabo. b) Stråle drevet. c) Stråle-drevet med oversampling. d) Areal afhængig.

2.3.1 Tre-dimensional rekonstruktion med iterative metoder

Hvis iterative metoder skal bruges til at rekonstruere tre-dimensionale volumen estimater for emission fra et objekt, er forskellen fra de viste to-dimensionale metoder at det rekonstruerede objekt bliver diskretiseret i tre dimensioner f.eks. med tre-dimensionale kubiske volumenelementer, *voxels*. De forskellige tre-dimensionale sammenfaldslinier mellem alle brugte kombinationer af detektorer bruges derefter til at estimere det n'te rum-elements andel i emissionen for den m'te sammenfaldslinie som i det to-dimensionale tilfælde. Flere har vist, at denne modellering af transformations matricen i tre dimensioner er mere kritisk end i det to-dimensionale tilfælde (se bla. Lewitt [Lew83]), hvorfor flere alternative samplingsstrategier er blevet foreslået.

Den eneste forskel der reelt er på selve rekonstruktionen i tre dimensioner med de iterative metoder er derfor kompleksiteten, da ligningssystemet bliver mange gange større, løsningsmetoderne er de samme som i to dimensioner. I praksis lader ligningssystemet sig løse, og der kan med de iterative metoder rekonstrueres bedre billeder end med de traditionelle direkte tre-dimensionale metoder (se f.eks. Kinahan [Kin94]).

2.4 Løsning af lineære ligningssystemer

Løsning af lineære ligningssystemer, er en veletableret gren af matematikken. I forbindelse med matrix formuleringen af billedrekonstruktionen optræder der dog nogle specielle problemer, der skal tages hensyn til. Systemet i formel (2.10) er ekstremt stort, da både $\mathbf{b}$ og $\mathbf{x}$ begge kan have over 10^4 elementer for at rekonstruere billeder med realistisk opløsning. Matricen $\mathbf{A}$ kan derfor i princippet nemt have over 10^8 elementer, men vil være meget sparse, da det gerne er under 1% af elementerne der har en værdi, der reelt er forskellig fra nul.

Systemet er som regel overbestemt (iflg. Censor [Cen83]), hvilket betyder at der ofte ikke vil eksistere en entydig løsning $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, hvor $\mathbf{A}^{-1}$ er den inverse af $\mathbf{A}$. Ydermere er det ikke sikkert, at en eksakt algebraisk løsning vil være mere ønskelig i rekonstruktionsøjemed, end en anderledes defineret løsning, grundet måleusikkerhed, støjfyldte data, og diskretiseringen af det originale rekonstruktionsproblem.

Brugen af series expansion metoder til løsning af billedrekonstruktions problemet, kan beskrives som følgende sekvens

1) Det originale rekonstruktionsproblem formuleres.

2) Diskretisering af rekonstruktionsområdet.

3) Opstilling af et algebraisk ligningssystem.

4) Valg af passende løsningsmetode.

5) Implementering.

I trin 1) vil den fysiske geometri for selve skanneren blive brugt, og der tages højde for alle de fysiske karakteristika for systemet (Emission, dæmpning ol.). Som regel vil mere eller mindre idealiserede antagelser for disse blive brugt i selve modellen. I det næste trin bliver rekonstruktionsområdet diskretiseret. Det er i dette punkt at series expansion metoderne skiller sig ud fra de almindelige transformationsmetoder. En transformationsmetode behandler problemet i en kontinuert udformning lige indtil implementeringsfasen, hvor de nødvendige antagelser om inversionsformlen formuleres således, at en numerisk implementering bliver mulig. I trin 3) fører diskretiseringen til et system af algebraiske ligninger som ovenfor beskrevet, for hvilke der i trin 4) vælges en passende løsningsmetode til, i lyset af de specielle karakteristika det opstillede ligningssystem har. Som nævnt vil systemet være stort, meget sparse og normalt dårligt konditioneret. Denne løsningsmodel vil ofte blive kombineret med et optimerings kriterium, defineret ud fra ligningssystemet, eller med et system af uligheder udledt fra det, som en række grænser indenfor hvilke optimeringen skal foretages. Denne fase er meget vigtig, da beslutninger taget her, vil influere på hvilke sæt af 'løsninger' der vil komme i betragtning. Den grundlæggende idé er, at bruge informationen i systemet i formel (2.10), på en måde der afspejler den begrænsede tro på ligningssystemets løsninger, ved f.eks. at medtage *a priori* information om det rekonstruerede objekt. Der er således flere måder at vælge en løsningsmodel på, og her bør det igen være den aktuelle applikation der bestemmer valget. Det sidste trin er udviklingen af en rekonstruktions algoritme, der skal være i stand til at håndtere de problemer, der er i rekonstruktionen, og skal kunne implementeres på en computer. Den implementerede rekonstruktions algoritme, kan derefter evalueres i forhold til det originale rekonstruktionsproblem.

Der er dermed ingen direkte forbindelse mellem de direkte og de iterative rekonstruktionsmetoder. Dog har Older [Old93] ved hjælp af matrix formuleringer vist, at de filtrerede tilbageprojektionsmetoder finder en approksimativ løsning til rekonstruktionen formuleret som et mindste kvadraters problem.

2.5 Implementering af sparse strukturer

En matrix eller vektor struktur kaldes for *sparse*, hvis hovedparten af elementerne har værdien nul. Hvis denne viden udnyttes, kan programmer ved passende lagring og udnyttelse af sparsiteten reducere lager- og beregningskravene væsentligt.

En matrix vil i en computer kræve lagerplads svarende til en floating point lagerplads pr. element samt en smule administration. Implementeret med enkelt præcisions tal (32 bit), kræves derfor mindst $4MN$ bytes lagerplads for en $M \times N$ matrix. Ved at udnytte forhåndsviden om sparsitet kan dette lagerkrav reduceres til at være proportionalt med antallet af elementer der er forskellige fra nul.

Den metode der bliver benyttet i det implementerede software, kan kaldes for *søjle indekseret lagring*, hvor kun søjlenumrene og værdien på de elementer der er forskellige fra nul i hver række bliver gemt. For at gemme den $M \times N$ store matrix $\mathbf{A}$, skal der bruges to vektorer `value` og `index` for hver række. Den første vektor gemmer de flydende talværdier, den anden heltals søjle indekser. Ydermere skal anvendes en vektor `Nm`, der gemmer hvor mange elementer der er i hver række. To variable `M` og `N` repræsenterende dimensionerne på matricen, gemmes også. Som eksempel, vil matricen

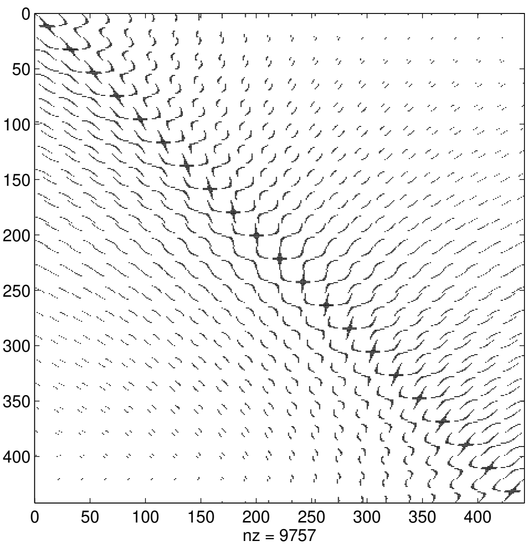
$$\begin{bmatrix} 0 & 0 & 17 & 30 \\ 47 & 13 & 0 & 0 \\ 0 & 7 & 0 & 0 \end{bmatrix} \tag{2.12}$$

kunne repræsenteres på sparse form. For første række haves værdierne 17 og 30 på anden hhv. tredie plads som eneste elementer forskellige fra nul. Således bliver `Nm[0]`=2, `index[0]`=[2,3] og `value[0]`=[17,30]. Dimensionerne er `M`=3 og `N`=4. For denne matrix er der ikke nogen reel pladsbesparelse, men for meget store matricer, hvor relativt få af elementerne er forskellige fra nul, vil der med fordel kunne bruges sparse matrix strukturer.

2.6 Løsning af sparse lineære ligningssystemer

Den store størrelse af ligningssystemet i formel (2.10) gør, at det er nødvendigt at løse ligningssystemet ved hjælp af iterative metoder. Dog er flere alternative metoder blevet afprøvet i praksis, Xu et al. [Xu,93a], har bla. vist en ikke iterativ algebraisk metode, der løser ligningssystemet ved hjælp af en polynomisk approksimation af transformations matricen.

Direkte og algebrariske metoder til løsning af sparse matrix systemer vil være afhængige af strukturen af sparsiteten. Sparse matricer optræder i mange former for problemer, og i de tilfælde hvor sparsiteten har en regulær struktur, har dette ofte ført til udviklingen af specielle løsningsmetoder til lige netop den type sparsitet. I tilfældet med Radon rekonstruktion, vil sparsiteten i transformations matricen dog ikke have en struktur, der kan bruges til at simplificere problemet med. På figur 2.4 er vist en 441×441 transformations matrix, svarende til et sinogram på 21×21 værdier, og et output billede med 21×21 pixels. Denne transformations matrix er genereret med en nærmeste nabo approksimation til sammenfaldslinierne således, at der kun er værdier på nul hhv. Δx i matricen. Elementer med værdien Δx er vist som sorte prikker. Matricen har 194.481 elementer, hvoraf de 9.757 (5,02%) er forskellige fra nul.



Figur 2.4: Transformations matrix på 441×441 elementer. Elementer med værdi forskellig fra nul er vist sorte.

Den valgte implementering af sparse matrix strukturer som beskrevet i afsnit 2.5 gør, at operationer på en sparse matrix, såsom indsættelse af et element eller ændring af en værdi, bliver kompliceret en del i forhold til operationer på matricer gemt med konventionelle metoder. At denne implementering er anvendelig i forbindelse med iterative metoder skyldes, at det er tilstrækkeligt at kende transformations matricen $\mathbf{A}$, eller dens transponerede $\mathbf{A}^T$'s operation på en vektor. Transformations matricen skal således ikke manipuleres, efter den er blevet dannet.

Den måde de implementerede iterative rekonstruktions algoritmer arbejder på, er ved først at danne hele transformations matricen, og derefter udføre iterationerne. Matricen genereres række for række og hver række lagres i en sparse matrix struktur. Dette kræver, at den computer rutinerne skal køre på, har kapacitet nok til at have hele transformations matricen i hukommelsen på én gang, hvilket selv sagt kan være problematisk ved rekonstruktion af større billeder. En anden fremgangsmåde er, at danne rækkerne og søjlerne i transformations matricen når de skal bruges til en operation, uden at gemme dem bagefter. Dette reducerer lagerkravene betydeligt på bekostning af forøget beregningstid. Da størstedelen af beregningstiden for en iterativ algoritme går med at udregne transformations matricen, ses at i f.eks. ART opdateringen (formel 3.5) skal der udregnes én række i transformations matricen for hver iteration. Hvis antallet af iterationer ender med at blive ca. ti gange antallet af rækker i transformations matricen, vil denne implementering være ca. en faktor ti gange langsommere end hvis hele matricen udregnes én gang for alle. således er der to måder at vælge at implementere en iterativ metoder på:

- Langsom med lille lagerforbrug,
- Hurtig med stort lagerforbrug, og dermed begrænsninger på størrelsen af de rekonstruerede billeder.

Ved at udregne og bruge hele transformations matricen på en gang, evt. ved at hente den fra disk, fås et mere realistisk billede af de iterative algoritmers praktiske beregningshastighed i forhold til hinanden og i forhold til transformationsmetoderne. Det er valgt at lade de implementerede algoritmer arbejde med hele transformations matricen på én gang, dvs. den hurtige måde.

2.7 Program til iterativ rekonstruktion

I denne del af projektet er der udviklet et program til rekonstruktion med iterative metoder, der indeholder implementeringer af de i de næste kapitler beskrevne rekonstruktions algoritmer. Dette programs opbygning vil kort blive beskrevet her.

2.7.1 Matrix-vektor bibliotek

Under udarbejdelsen af denne del af projektet, er der i et samarbejde mellem Jesper James Jensen og Peter Toft, blevet udviklet et generelt sæt af rutiner til manipulering af sparse matrix- og vektor strukturer (se appendix C). Disse er implementeret med en lagring af elementerne i sparse strukturer som beskrevet i afsnit 2.5.

Dette bibliotek, `sparse.c` er udviklet således, at det kan udføre hovedparten af de generelle matrix- og vektor udregninger og opdateringer der foretages i forskellige iterative algoritmer til PET rekonstruktion. Sparse biblioteket indeholder bla.

- Definition af `SparseMatrix`, `SparseVektor` og `Vektor` strukturerne, der indeholder sparse matricer, sparse vektorer hhv. almindelige vektorer,
- Rutiner til at allokere og frigøre disse strukturer,
- Konvertering fra almindelige vektorer til sparse vektorer,
- Rutiner til at danne sparse matricer ved hjælp af sparse vektorer,
- Udregning af sparse matrix-vektor produkter og sparse vektor-vektor indre produkter,

- Beregning af forskellige normer for matricer og vektorer,
- Transponering af sparse matricer,
- Multiplikation med den transponerede til en sparse matrix uden at denne dannes eksplicit.
- Rutiner til at lagre og hente sparse matricer i eget fil-format, samt til at gemme i MatLab format.

At bruge et sådan standard bibliotek gør, at forskellige iterative algoritmer meget nemt og hurtigt kan implementeres og afprøves.

2.7.2 Valgte rekonstruktions algoritmer

Der er til denne del af projektet valgt at implementere et repræsentativt udsnit af de eksisterende iterative rekonstruktions algoritmer. Der er blevet implementeret følgende algoritmer:

- ART (Algebraric Reconstruction Technique).
- ML-EM (Maximum Likelihood Expectation Maximization).
- LS-CG (Least Squares Conjugate Gradients).

EM algoritmen er den for tiden mest beskrevne iterative rekonstruktions algoritme i litteraturen, og lover meget god rekonstruktions kvalitet. ART algoritmen er en variant af en af de første iterative rekonstruktions algoritmer til CT og PET, og en god indgangsvinkel til rekonstruktion fra projektioner, da den er simpel at forstå og let at implementere. CG algoritmen er en velbeskrevet og anerkendt metode til at løse ligningssystemer med, og er også set brugt i PET sammenhæng.

2.7.3 'it' programmet

Ved hjælp af sparse matrix biblioteket, er de tre valgte iterative rekonstruktions algoritmer blevet implementeret, og samlet til et program, `it`. Der gøres også brug af de under den første del af projektet udviklede biblioteker til bla. billedbehandling. Programmet er opbygget efter samme model som `iradon` programmet fra projektets del I. Dette betyder, at alle billedformater og samplingsparametre håndteres på samme måde, og programmerne er i stand til at rekonstruere de samme billeder ud fra de samme sinogrammer, hvis rekonstruktionsparametrene opgives ens. Dette muliggør en sammenligning mellem de direkte og de iterative rekonstruktionsmetoder. Brugen af `it` programmet er beskrevet i appendix A, en udskrift af kildekoden findes i appendix B og en LaTeX-formateret udskrift af funktionshovederne findes i appendix C.

2.8 Resumé

I dette kapitel er det blevet vist hvorledes PET rekonstruktion i to og tre dimensioner kan modelleres som et lineært ligningssystem, der kan løses med iterative metoder. Den brugte implementering af sparse matrix strukturer er blevet vist, og det udviklede program, `it` er blevet præsenteret. I de følgende kapitler vil de tre implementerede rekonstruktions algoritmer blive gennemgået.

Kapitel 3

Algebraisk rekonstruktionsteknik

I dette kapitel vil den almindelige ART (Algebraric Reconstruction Technique) algoritme til iterativ rekonstruktion blive gennemgået, og implementeringen af denne vil blive beskrevet. Flere aspekter omkring generelle problemer ved iterative metoder vil blive diskuteret.

3.1 ART algoritmen

Den første algebraiske rekonstruktionsmetode til CT rekonstruktion, ART algoritmen blev publiceret i 1970 af Gordon et al. [Gor70]. Grundet rekonstruktionsproblemets store størrelse er denne konstrueret som en iterativ proces, startende med en approksimation af billede vektoren $\mathbf{x}^0 \in \mathbf{R}^N$. I hvert iterativt trin bliver det nuværende estimat for billedet $\mathbf{x}^k$, opdateret til et nyt estimat $\mathbf{x}^{k+1}$, ved for en sammenfaldslinie m, kun at opdatere de pixels denne linie krydser. Uoverensstemmelsen mellem den målte projektion b_m og projektionen med det nuværende gæt, $\sum_{n=o}^{N-1} a_{m,n}x_n^k$ udregnet fra det nuværende billede $\mathbf{x}^k$, fordeles mellem elementerne langs den m'te linie proportionalt med deres vægt $a_{m,n}$ i transformationen. Således bliver elementer langs den m'te linie opdateret til den m'te måling uden at ændre på resten af billedet. Hvis den m'te række i matricen $\mathbf{A}$, skrives som vektoren $\mathbf{a}_m$, kan ART beskrives med følgende skridt: Først initialiseres $\mathbf{x}^0$ til en given størrelse, dernæst udføres følgende opdatering for hver iteration k:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \frac{b_m - \mathbf{a}_m \cdot \mathbf{x}^k}{\mathbf{a}_m \cdot \mathbf{a}_m}\mathbf{a}_m. \tag{3.1}$$

Udvælgelsen af de linier m, der opdateres med som funktion af iterationsnummeret k, kan foretages på forskellige måder. De to mest simple, og mest anvendte metoder baserer sig på udvælgelse af linierne i rækkefølge som

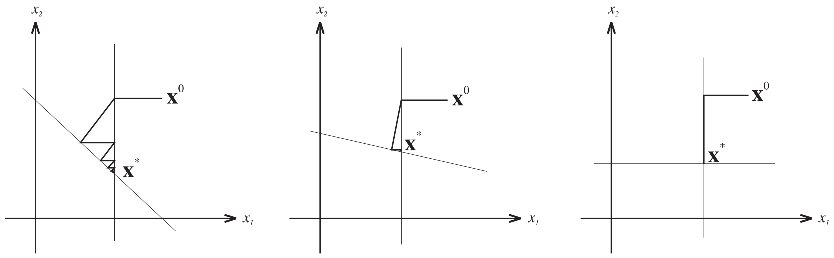
$$m = m_k = k \bmod M, \tag{3.2}$$

eller tilfældig udvælgelse af m for hver iteration som

$$m = m_k = \mathrm{random}(M), \tag{3.3}$$

hvor $a \bmod b$ er modolus operatoren, og $a = \mathrm{random}(b)$ returnerer et tilfældigt helt tal, $a \in \mathbf{Z}$ hvor $0 \leq a < b$. Det viser sig at konvergensen af ART algoritmen, er meget afhængig af den metode hvormed m_k vælges. Eksperimentielle studier har vist, at fremgangsmåden med tilfældige tal, som i formel (3.3), er den bedste generelle metode. Herman [Her93b] har udarbejdet en metode baseret på udvælgelse afhængigt af opsamlingsgeometrien, som giver bedre resultater end tilfældig udvælgelse. Guan og Gordon [Gua94], har udviklet en udvælgelsesmetode, der svarer til bit-reverseringen i en én–dimensional FFT. Denne har vist at give hurtigere konvergens.

ART har udover den intuitivt meningsfyldte fortolkning, også en geometrisk fornuftig fortolkning. Sættet $\mathbf{H}_m = \{\mathbf{x} \in \mathbf{R}^N | \mathbf{a}_m \cdot \mathbf{x} = b_m\}$, der består af alle løsninger til den m'te ligning fra formel (3.1) er et hyperplan i $\mathbf{R}^N$, hvis normal vektor er $\mathbf{a}_m$. Punktet $\mathbf{x}^{k+1}$ er således den ortogonale projektion af $\mathbf{x}^k$ på $\mathbf{H}_m$. ART vil derfor successivt projicere løsningerne ind på hyperplanet $\mathbf{H}_m$ repræsenteret af ligningerne i systemet i formel (3.1) hvor rækken m er udvalgt efter (3.2) eller (3.3). Hvis der eksisterer en unik løsning til systemet i formel (2.10), vil alle hyperplaner for $m = 0, 1, \ldots, M-1$ krydse i et unikt punkt, løsningen til ligningssystemet. På figur 3.1 er vist proceduren, der finder løsningen ($\mathbf{x}^*$) for formel (2.10) med to variable, bemærk forskellen i antallet af iterationer der skal foretages, når vinklen mellem linierne ændres.



Figur 3.1: Iterativ løsning af formel (2.10) med to ukendte vha. ART.

For at bedømme en sådan iterativ proces, må nogle matematiske spørgsmål stilles omkring sekvensen af iterationer, $\mathbf{x}^k$, $k = 0, 1, \ldots, \infty$. Vil denne serie konvergere? Hvad vil den konvergere mod — hvad er naturen af $\mathbf{x}^\infty$? Hvad er konvergenshastigheden? Disse spørgsmål omhandler alle den asymptotiske natur af sekvensen af iterationer. I praksis vil en sådan iterativ proces blive stoppet efter et endeligt antal iterationer, og en bestemt iteration vil blive brugt som approksimation til den ønskede og uopnåelige grænse-vektor.

Dermed bliver den vigtigste faktor i evalueringen af denne iterative proces, hvorledes $\mathbf{x}_k$ i de første iterationer opfører sig, og hvor god $\mathbf{x}_k$ er i forhold grænsen $\mathbf{x}^\infty$ som approksimation til den ønskede billede vektor. Det generelle problem med iterative rekonstruktions algoritmer er, at de genererer en sekvens af billed-vektorer, der konvergerer mod en grænse vektor der antages at have forskellige ønskelige egenskaber. Denne grænse vektor kan af gode grunde aldrig nås i praksis, og det er spørgsmålet hvor mange af disse egenskaber der så kan genfindes i den tilnærmede løsning.

3.2 Dæmpningsparameter

I en implementering af ART, vil der blive brugt en dæmpningsparameter til at styre opdateringen med. Denne dæmpningsparameter er en sekvens af tal λ_k, normalt defineret som

$$\epsilon_1 \leq \lambda_k \leq 2 - \epsilon_2, \qquad \epsilon_1, \epsilon_2 > 0, \tag{3.4}$$

for $k = 0, 1, \ldots, \infty$. Disse vil optræde i opdaterings algoritmen som

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda_k \frac{b_m - \mathbf{a}_m \cdot \mathbf{x}^k}{\mathbf{a}_m \cdot \mathbf{a}_m} \mathbf{a}_m, \tag{3.5}$$

og vil tillade at over- eller under opdatere den ortogonale projektion. Vigtigheden af at bruge denne dæmpningsparameter er i litteraturen eksperimentielt blevet vist af mange (se f.eks. Herman [Her80]). Der eksisterer ingen generel optimal dæmpningsparameter, da valget af denne vil afhænge af selve den medicinske applikation, det geometriske arrangement af detektorer, støjens indflydelse, og det antal iterationer der vil blive brugt.

Dermed bliver den eneste pålidelige måde hvormed man kan bestemme de optimale dæmpningsparametre for en aktuel rekonstruktions applikation, ved hjælp af eksperimenter. Hvis det antages, at der eksisterer et udvalg af repræsentative datasæt for den aktuelle opgave, og der bliver brugt et kvalitetsmål, der reflekterer rekonstruktionenskvalitet i forhold til den aktuelle applikation, til at vurdere de rekonstruerede data med, kan der for hver iteration k, findes en λ_k der er optimal. Denne procedure involverer en søgning efter den λ_0 der minimerer det opstillede kvalitetsmål efter første iteration. Dernæst findes, ved brug af λ_0, den λ_1 der minimerer kvalitetsmålet efter anden iteration og så fremdeles. Ved brug af denne metode fås ifølge Herman [Her93b] overlegen rekonstruktionskvalitet i forhold til brugen af en konstant dæmpningsfaktor.

3.3 Startgæt

Før den iterative proces startes, skal startvektoren $\mathbf{x}^0$, der repræsenterer et startbillede, initialiseres. Dette gælder for alle de iterative rekonstruktions algoritmer. Startgættet $\mathbf{x}^0$ kan sættes til en bestemt konstant gråtoneværdi for alle pixels, eller et rigtigt startgæt f.eks. udregnet ved hjælp af en transformations algoritme kan bruges. Kaufmann [Kau87] argumenterer for brugen af et gråtone billede, medmindre et andet billede af den samme patient er tilgængeligt, da omkostningerne ved at bruge en anden algoritme ikke står mål med den hurtige konvergens for f.eks. den accelererede EM algoritme. Da formålet i denne rapport er at teste rekonstruktions algoritmerne mod hinanden, vil resultaterne ikke afspejle algoritmernes rigtige opførsel, hvis et startgæt fra en fremmed rekonstruktions algoritme bliver indført.

Som første gæt til algoritmerne vil derfor her blive brugt et jævnt gråtone billede baseret på en antagelse om konstant værdi i alle pixels. Et fornuftigt valg af denne værdi er et estimat af middelværdien i det rekonstruerede billede ud fra de opsamlede data som

$$\bar{x} = \frac{\sum_{n=0}^{N-1} b_n}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} a_{m,n}}, \tag{3.6}$$

hvor $\mathbf{x}^0$ initialiseres med $x_n^0 = \bar{x},\ n = 0, 1, \ldots, N-1$. Dette startgæt vil blive brugt for alle de iterative rekonstruktions algoritmer.

3.4 Implementering af ART

Der er implementeret en ART algoritme med dæmpning som i formel (3.5). Denne er udelukkende blevet implementeret ved hjælp at sparse toolboksen (se afsnit 2.7.1), bortset fra en enkelt funktion, lavet specielt for at opnå højere hastighed ved udregning af opdateringen $\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha \mathbf{a}_m$, som kun bruges af ART algoritmen (se funktionen `ARTUpdateAddVector` i appendix C). Der er i alle de implementerede iterative algoritmer mulighed for enten at bruge et på forhånd udregnet billede som startgæt, eller et estimat for middelværdien udregnet som i formel (3.6).

I modsætning til de andre algoritmer, opdaterer ART kun billedet ud fra én sammenfaldslinie pr. iteration. CG og EM algoritmerne opdaterer hele billedet ved hjælp af alle sammenfaldslinier i hver iteration. ART vil derfor skulle køre M iterationer, før al informationen i sinogrammet er blevet brugt til at opdatere billedet med. Derfor vil en *iteration* af ART fremover svare til M *opdateringer* af formel (3.5).

I den implementerede ART rutine, er dæmpningsfaktoren λ_k implementeret ved hjælp af to konstanter α og β, hvor dæmpningsfaktoren udregnes før hver iteration som

$$\lambda_k = \alpha\beta^k, \tag{3.7}$$

hvilket gør det muligt at bruge en konstant dæmpningsfaktor, eller at lade den stige eller aftage efter hver iteration (M opdateringer).

Kapitel 4

Konjugerede gradienters metode

I dette kapitel, vil de konjugerede gradienters metode til løsning af et lineært ligningssystem blive præsenteret, og den benyttede fremgangsmåde diskuteret.

4.1 Den kvadratiske form

Konjugerede gradienters metoder er en generel måde at løse et $N \times N$ lineært system som i formel (2.2). Løsningen er baseret på en minimering af den kvadratiske form

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x} \cdot \mathbf{A}\mathbf{x} - \mathbf{b} \cdot \mathbf{x} + c, \tag{4.1}$$

hvor c er en konstant. Den kvadratiske form er symmetrisk og positiv definit afhængigt af $\mathbf{A}$. Den kvadratiske forms gradient er

$$f'(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}. \tag{4.2}$$

Den kvadratiske form f, vil være minimeret når gradienten er nul. Dermed relaterer minimeringen af kvadratiske former tæt til løsningen af lineære systemer.

4.2 CG algoritmen

Her vil en generel iterativ konjugerede gradienters løsnings algoritme blive præsenteret. Metoden er en velbeskrevet metode der løser sparse systemer og ligninger. For en nærmere beskrivelse se bla. Scales [Sca93] og Hansen [Han92].

Konjugerede gradienter fortager minimeringen af den kvadratiske form i formel (4.1) ved at generere en række søgeretninger $\mathbf{p}^k$ og forbedrede minimeringer $\mathbf{x}^{k+1}$. For hver iteration findes en værdi α, der minimerer $f(\mathbf{x}^k + \alpha\mathbf{p}^k)$, og $\mathbf{x}^{k+1}$ sættes lig det nye punkt $\mathbf{x}^k + \alpha\mathbf{p}^k$. Søgeretningen $\mathbf{p}^k$ og $\mathbf{x}^{k+1}$ genereres således at $\mathbf{x}^{k+1}$ også minimerer f over hele det vektor–rum af retninger allerede afsøgt, $[\mathbf{p}^0, \mathbf{p}^1, \ldots, \mathbf{p}^k]$. Efter K iterationer haves en minimering for hele vektorrummet.

Først initialises $\mathbf{x}^0$, sæt $\mathbf{p}^0 = \mathbf{r}^0 = \mathbf{b} + \mathbf{A}\mathbf{x}^\mathbf{0}$. Derefter udregnes følgende opdateringer for $k = 0, 1, 2, \ldots$

$$\begin{array}{lcl} \alpha & = & (\mathbf{r}^k \cdot \mathbf{r}^k)/(\mathbf{p}^k \cdot \mathbf{A} \cdot \mathbf{p}^k) \\ \mathbf{x}^{k+1} & = & \mathbf{x}^k + \alpha\mathbf{p}^k \\ \mathbf{r}^{k+1} & = & \mathbf{r}^k - \alpha\mathbf{A}\mathbf{p}^k \\ \beta & = & (\mathbf{r}^{k+1} \cdot \mathbf{r}^{k+1})/(\mathbf{r}^k \cdot \mathbf{r}^k) \\ \mathbf{p}^{k+1} & = & \mathbf{r}^{k+1} + \beta\mathbf{p}^k \end{array} \tag{4.3}$$

Den $\mathbf{A}$ matrix der bliver genereret til PET rekonstruktion, vil generelt ikke være symmetrisk, og vil ikke kunne udnytte denne generelle CG metode. For en ikke singulær matrix $\mathbf{A}$, vil $\mathbf{A}^T\mathbf{A}$ dog være symmetrisk og positiv definit. Derfor kan en løsning være at bruge konjugerede gradienters metode på ligningssystemet

$$\mathbf{A}^T \cdot \mathbf{b} = (\mathbf{A}^T\mathbf{A}) \cdot \mathbf{x}. \tag{4.4}$$

Det skal undgås at danne produktet $\mathbf{A}^T\mathbf{A}$, da dette normalt ikke vil være sparse selv om $\mathbf{A}$ er det. Da konjugerede gradienters metode ikke kræver kendskab til selve $\mathbf{A}$ matricen, kan matrix–vektor multiplikationen tænkes foretaget ved først at danne produktet $\mathbf{A} \cdot \mathbf{x}$, og derefter at gange $\mathbf{A}^T$ på. Denne fremgangsmåde bør dog *ikke* bruges i forbindelse med konjugerede gradienters metode, da konditionstallet for $\mathbf{A}^T\mathbf{A}$ vil være kvadratet af $\mathbf{A}$'s konditionstal. Et stort konditionstal, vil både øge antallet af krævede iterationer, og begrænse den nøjagtighed hvormed en løsning kan findes. Løsningen ifølge Scales [Sca93], er at sætte ligningen $\mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{A}\mathbf{x}$ på formen $\mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x})$, hvor subtraktionen blive udført før multiplikationen. Resultatet er følgende fremgangsmåde: Vælg en $\mathbf{x}_0$, sæt $\mathbf{s}^0 = \mathbf{b} - \mathbf{A}\mathbf{x}^\mathbf{0}$, $\mathbf{r}^0 = \mathbf{p}^0 = \mathbf{A}^T\mathbf{s}^0$, $\mathbf{q}^0 = \mathbf{A}\mathbf{p}^\mathbf{0}$, og udfør følgende opdateringer for $k = 0, 1, 2, \ldots$

$$\begin{array}{lcl}
\alpha & = & (\mathbf{r}^k \cdot \mathbf{r}^k)/(\mathbf{q}^k \cdot \mathbf{q}^k) \\
\mathbf{x}^{k+1} & = & \mathbf{x}^k + \alpha \mathbf{p}^k \\
\mathbf{r}^{k+1} & = & \mathbf{A}^T\mathbf{s}^k \\
\mathbf{s}^{k+1} & = & \mathbf{s}^k - \alpha \mathbf{q}^k \\
\beta & = & (\mathbf{r}^{k+1} \cdot \mathbf{r}^{k+1})/(\mathbf{r}^k \cdot \mathbf{r}^k) \\
\mathbf{p}^{k+1} & = & \mathbf{r}^{k+1} + \beta \mathbf{p}^k \\
\mathbf{q}^{k+1} & = & \mathbf{A}\mathbf{p}^{k+1}
\end{array} \tag{4.5}$$

4.3 Implementering af CG

Der er ved hjælp af sparse biblioteket blevet implementeret en CG-LS algoritme som den vist i formel (4.5). En speciel funktion der kan udregne vektor-opdateringen $\mathbf{a} = \mathbf{b} + \alpha\mathbf{c}$ er lavet for at forøge beregningshastigheden (se funktionen `CGUpdateAddVector` i appendix C).

Kapitel 5

Expectation Maximization

I dette kapitel, vil der blive givet en kort gennemgang af en *Expectation Maximization* (EM) metode til PET rekonstruktion. Den valgte metode er *Maximum Likelihood* baseret (ML-EM). Baggrunden for algoritmen vil kort blive vist, og nogle aspekter af den praktiske implementering vil blive diskuteret.

5.1 EM algoritmen

Statistiske metoder til rekonstruktion i PET, baserer sig på optimering af en målfunktion i forhold til det rekonstruerede billede. Ofte brugte målfunktioner er maximum likelihood (ML), mindste kvadrater (LS) eller maximum *a posteriori* (MAP). Den valgte målfunktion, maximum likelihood, er repræsentativ for de statistiske metoder, og en af de mest anvendte.

Grundlaget for ML-EM metoden, er parallel med ART og CG metoderne. Der haves en vektor $\mathbf{b} \in \mathbf{R}^M$ repræsenterende de målte projektioner, hvor b_m er antallet af foton par målt af detektorparret m. Den størrelse der skal estimeres, er antallet af emissioner inden for et område af det diskretiserede objekt. For billedelementet n, er den ukendte emission $\mathbf{x}_n$. Det er fra fysikken krævet, at alle $x_n \geq 0$, og det ideelle ville være, hvis antallet af emitterede foton-par svarer til antallet af detekterede par.

Maximum likelihood metoden, foreslået af Shepp og Vardi i 1982 [She82], er baseret på den antagelse, at emissionernes antal står i forhold til en rumlig Poisson-proces i et bestemt område. Hvis b_m'erne antages at være uafhængige Poisson variable med middelværdi λ_m^*, har Shepp og Vardi vist, at $\mathbf{x}$ kan findes ved at maksimere likelihood funktionen

$$L(\mathbf{x}) = \prod_{m=0}^{M-1} e^{-\lambda_m^*} \frac{\lambda_m^{*b_m}}{b_m!} \ , \tag{5.1}$$

hvor M er antallet af diskrete sammenfaldslinier mellem detektorerne. Den vektor $\mathbf{x}$ der maksimerer $L(\mathbf{x})$, vil også maksimere $l(\mathbf{x}) = \log(L(\mathbf{x}))$, hvis gradient er nemmere at udregne end gradienten for $L(\mathbf{x})$. Gradienten for $l(\mathbf{x})$ kan vises at være

$$\frac{\partial l(\mathbf{x})}{\partial x_n} = -1 + \sum_{m=0}^{M-1} \frac{a_{m,n} b_m}{\sum_{n'=0}^{N-1} a_{m,n'} x_{n'}}. \tag{5.2}$$

Vardi [Var85], viser at $l(\mathbf{x})$ er konkav, og at Kuhn-Tucker betingelserne

$$0 = x_n \frac{\partial l(\mathbf{x})}{\partial x_n} \qquad \text{for } n = 0, 1, \ldots, N-1 \text{ og } x_n > 0, \tag{5.3}$$

og

$$\frac{\partial l(\mathbf{x})}{\partial x_n} \geq 0 \qquad \text{for } n = 0, 1, \ldots, N-1 \text{ og } x_n = 0, \tag{5.4}$$

vil kunne finde et maksimum for $l(\mathbf{x})$, således at $\mathbf{x} \geq 0$. Dette fører til den originale EM opdaterings algoritme, som foreslået af Shepp og Vardi [She82] som

$$x_n^{k+1} = x_n^k \sum_{m=0}^{M-1} \frac{a_{m,n} b_m}{\sum_{n'=0}^{N-1} a_{m,n'} x_{n'}^k}. \tag{5.5}$$

Dette afsnit er ment som en kort introduktion til EM algoritmen, for yderligere information kan anbefales Shepp og Vardi [She82], Vardi [Var85], Kaufmann [Kau93] og Ollinger [Oll94].

5.2 Implementering af EM

Den specifikke version af EM der er blevet implementeret i dette projekt, bruger følgende opdateringsskridt

$$x_n^{k+1} = \frac{x_n^k}{\sum_{m=0}^{M-1} a_{m,n}} \sum_{m=0}^{M-1} \frac{a_{m,n} b_m}{\sum_{n'=0}^{N-1} a_{m,n'} x_{n'}^k}. \tag{5.6}$$

Denne algoritme er oprindeligt foreslået af Lange og Carson [Car85], og har eksperimentielt vist sig at give en rekonstruktionskvalitet, der er andre EM algoritmer overlegen (se Herman [Her93b]). Umiddelbart ser opdateringsformlen avanceret ud, men kan opskrives på en simplere måde. Først initialiseres $\mathbf{x}^0$, sæt $s_n = \sum_{m=0}^{M-1} a_{m,n}$ for $n = 0, 1, \ldots, N-1$. Derefter udføres følgende opdateringer for $k = 0, 1, 2, \ldots$

$$\begin{array}{lcll} \mathbf{r} & = & \mathbf{A}\mathbf{x}^k & \\ d_m & = & b_m / r_m & \text{for } m = 0, 1, \ldots, M-1 \\ \mathbf{y} & = & \mathbf{A}^T \mathbf{d} & \\ x_n^{k+1} & = & x_n^k y_n / s_n & \text{for } n = 0, 1, \ldots, N-1 \end{array} \tag{5.7}$$

I anden og fjerde linie ses, at der bliver foretaget en division. For at undgå problemer med division med nul for rækker og søjler i $\mathbf{A}$ der ingen værdier har, skal implementeringen af EM tage højde for dette, og resultatet af en division med nul skal sættes til nul svarende til, at den givne værdi i sinogrammet ikke får bidrag fra billedet. EM algoritmen er blevet implementeret med sparse biblioteket, pånær opdateringen $x_n^{k+1} = x_n^k y_n / s_n$ for $n = 0, 1, \ldots N-1$, der er lavet specielt til EM (se funktionen `EMUpdateMultVector` i appendix C).

Kapitel 6

Forbedringer af de iterative algoritmer

I dette kapitel, vil to simple metoder til forbedring af rekonstruktions algoritmerne blive præsenteret. Begge er meget simple versioner af metoder, der i praksis har vist sig at have gavnlig effekt på rekonstruktionskvaliteten.

6.1 Regularisering

Der er foretaget indledende studier af regularisering af ligningssystemer, med udgangspunkt i billed rekonstruktionsproblemet i denne rapport.

Løsningen af rekonstruktionsproblemet med iterative metoder, foretages ved at løse ligningssystemet i formel (2.10). Regularisering af dette ligningssystem kan illustreres ved definition af to funktioner $\mathcal{A}$ og $\mathcal{B}$. Den første $\mathcal{A}$ måler hvorledes den fundne løsning er i overensstemmelse med modellen og sinogrammet $\mathbf{b}$. Når $\mathcal{A}$ er minimeret, er overensstemmelsen mellem den fundne løsning og modellen god, ofte alt for god da løsningen kan blive ustabil, oscillerende eller på andre måder urealistisk, da $\mathcal{A}$ alene måler på et ofte degenereret minimeringsproblem som før beskrevet.

Det er her $\mathcal{B}$ kommer ind, den kan afspejle en størrelse der beskriver hvorledes løsningen stemmer overens med en på forhånd bestemt egenskab ved de ønskede løsninger, f.eks. variationer i værdierne defineret ud fra *a priori* viden om løsningens struktur. $\mathcal{B}$ kaldes for en *regulariserings operator*. At minimere $\mathcal{B}$ alene vil ikke give en løsning, der har noget at gøre med de målte data.

Ideen med at regularisere er at minimere $\mathcal{A} + \lambda\mathcal{B}$ istedet for blot at minimere $\mathcal{A}$. På figur 6.1, er vist en skematisk fremstilling af afvejningen mellem forskellige værdier af $0 < \lambda < \infty$.

I praksis bliver regulariseringen medtaget i løsningen af ligningssystemet ved at erstatte minimeringen af målfunktionen

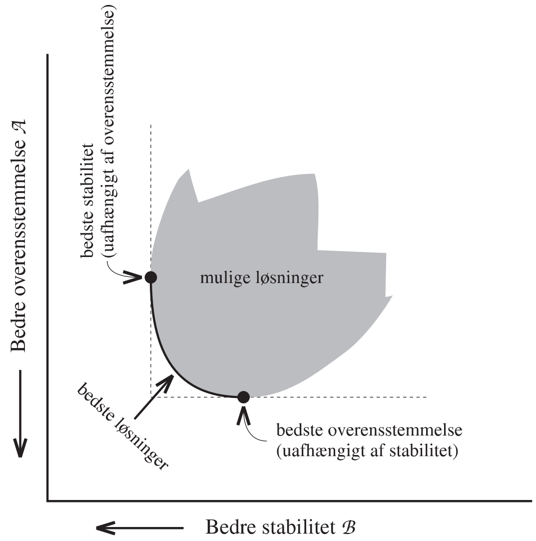
$$||\mathbf{Ax} - \mathbf{b}||^2 \tag{6.1}$$

med minimering af

$$||\mathbf{Ax} - \mathbf{b}||^2 + \lambda||\mathbf{Rx}||^2, \tag{6.2}$$

hvor der f.eks. kan vælges $\mathbf{R} = \partial^n$, $n = 0, 1, 2, \ldots$ hvor ∂^n er den n'te ordens differential operator. *Normalligningerne* forbundet med at sætte gradienten af målfunktionen i formel (6.2) til nul er

$$(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{R}^T\mathbf{R})\mathbf{x} = \mathbf{A}^T\mathbf{b}. \tag{6.3}$$



Figur 6.1: Løsningen af et inverst problem kan stilles op som en afvejning mellem to optimeringer: overensstemmelse mellem løsningen og modellen, her betegnet $\mathcal{A}$, eller en blødhed eller stabilitet af løsningen, her betegnet $\mathcal{B}$. De mulige løsninger er vist skematisk som det skraverede område.

Denne type regularisering er nem at implementere i de udviklede rutiner. Transformationsmatricen $\mathbf{A}$ forlænges med regulariseringsleddet som

$$\tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{R} \end{pmatrix}, \tag{6.4}$$

og vektoren $\mathbf{x}$ med nuller som

$$\tilde{\mathbf{b}} = \begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix}. \tag{6.5}$$

Da $\tilde{\mathbf{A}}^T\tilde{\mathbf{b}} = \mathbf{A}^T\mathbf{b}$ og $\tilde{\mathbf{A}}^T\tilde{\mathbf{A}} = \mathbf{A}^T\mathbf{A} + \lambda\mathbf{R}^T\mathbf{R}$, vil en mindste kvadraters løsning af $\tilde{\mathbf{A}}\mathbf{x} = \tilde{\mathbf{b}}$ opfylde formel (6.3). For at indkorporere regularisering som i formel (6.2) i de implementerede iterative algoritmer, skal transformations matricen blot have adderet regulariseringsleddet på og $\mathbf{b}$-vektoren udvides med samme antal nuller.

Da rekonstruktionsproblemet reelt er to-dimensionalt, er et populært valg af regulariseringsoperator en fem-punkts approksimation af Laplace-operatoren, forskellen mellem værdien i et punkt og gennemsnittet af dets fire naboer i det rektangulære raster grid. Denne svarer til at filtrere med den to-dimensionale filterkerne:

0	-1	0
-1	4	-1
0	-1	0

for hver punkt i $\mathbf{x}$. Der er implementeret en funktion der kan tilføje omtalte regulariseringsled til $\mathbf{A}$-matricen og udvide $\mathbf{b}$-vektoren med nuller, således at ligningssystemet kan løses med de almindelige iterative algoritmer.

Det ses på opdateringerne i EM algoritmen i formel (5.7), at regulariseringsleddet vil gå ud under udregningerne idet højresiden er nul. Regularisering vil derfor ikke ændre på resultatet. I følge Hansen [Han92] er CG algoritmen selvregulariserende. Det er således forventet at det er ART algoritmen der vil få mest gavn af denne type regularisering.

6.2 Constrains

En måde hvormed rekonstruktionskvaliteten og konvergensen af en iterativ rekonstruktionsmetode kan forbedres er, ved at bruge forhåndsviden om det objekt der skal rekonstrueres, i forbindelse med opstilling af modeller til rekonstruktionen. Dette er især anvendeligt i forbindelse med statistiske rekonstruktionsmetoder.

Denne forhåndsviden gør at man kan lægge forskellige bånd, *constrains* på rekonstruktions algoritmerne. En constrain operator kan udgøre en beregningsmæssig effektiv måde, hvormed der kan inkluderes ikke parametrisk deterministisk *a priori* viden om det rekonstruerede objekt i rekonstruktionen. En constrain operator kan implementeres således, at en person der har forhåndsviden om det objekt der skal rekonstrueres, kan bruge denne viden til at ændre det rekonstruerede billede med.

Der er i de implementerede algoritmer indbygget en simpel form for constrain. Denne gør det muligt at definere et interval, inden for hvilket værdierne for estimatet x_n er gyldige. Værdier for x_n, der falder udenfor dette område, vil blive trunkeret til laveste hhv. højeste tilladelige værdi, x_{min} og x_{max}. Hvis der tages udgangspunkt i ikke negativitet af den emitterede stråling i PET, er et fornuftigt valg at sætte $x_{min} = 0$. Værdien for x_{max}, kan sættes til en værdi svarende til den forventede maksimale intensitet, eller ikke bruges.

For alle kørsler med de iterative metoder, er værdien af x_{min} sat til 0. Der har ikke været brugt x_{max} constrains.

6.3 Resumé

I dette kapitel er to simple metoder til at forbedre rekonstruktionskvaliteten af de iterative rekonstruktions metoder blevet gennemgået, regularisering og constrains. Der vil senere blive vist hvorledes brugen af regularisering på ART algoritmen ændrer på det rekonstruerede billede.

Kapitel 7

Resultater

I dette kapitel, vil der blive præsenteret resultaterne af en række kørsler der er blevet foretaget med de implementerede iterative rekonstruktions algoritmer, for at vurdere deres rekonstruktionskvalitet. Der er ydermere foretaget kørsler, der skal illustrere de implementerede algoritmers muligheder for at lave zoom, deres rekonstruktionskvalitet i forbindelse med få projektioner og projektioner med støj. Det vil blive beskrevet under hvilke forhold kørslerne er blevet foretaget, og hvilke resultater der er opnået.

7.1 Vurdering af resultater

Som beskrevet i første del af rapporten, bør vurderingen af rekonstruktionskvaliteten for en rekonstruktions algoritme ses i lyset af den aktuelle applikation. I dette tilfælde forefindes kun et reference billede af det fantom, der skal rekonstrueres. Derfor vil vurderingen blive foretaget som en subjektiv sammenligning af de to billeder og en objektiv måling af en fejl-norm mellem referencebilledet og det rekonstruerede billede. Der vil blive brugt det samme L_2-mål som beskrevet i del I af rapporten, her vil dette dog blive brugt til at sammenligne vektorer med, og er for to vektorer, $\mathbf{a}, \mathbf{x} \in \mathbf{R}^n$, defineret som

$$L_2 = \sqrt{\frac{1}{\sigma_a^2 N} \sum_{n=0}^{N-1} (x_n - a_n)^2} \;, \tag{7.1}$$

hvor $\mathbf{a}$ er reference-vektoren, og $\mathbf{x}$ er det rekonstruerede gæt, og

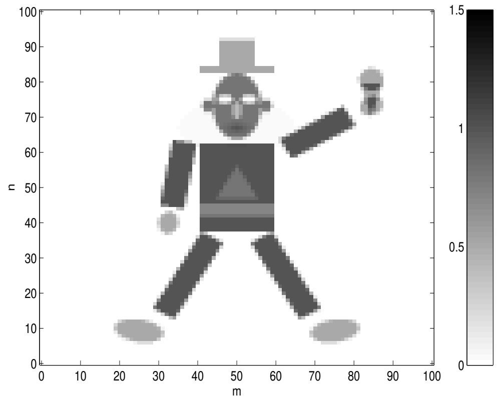
$$\sigma_a = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} (a_n - \bar{a})^2}, \qquad \bar{a} = \frac{1}{N} \sum_{n=0}^{N-1} a_n. \tag{7.2}$$

L_2-målet beskriver den normaliserede standardafvigelse af forskellen mellem det rekonstruerede billede og det brugte referencebillede af fantomet.

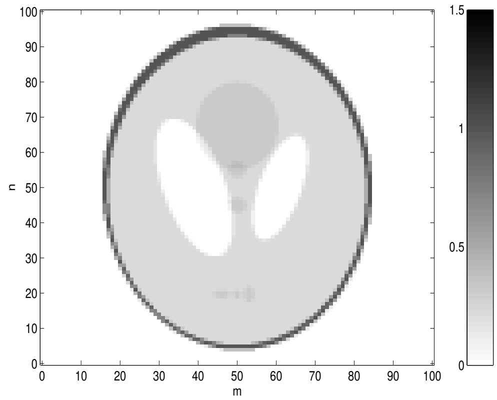
7.2 Valg af testbilleder

De testbilleder der vil blive brugt er *man*, *phantom* og *radial* fantomerne beskrevet i del I af rapporten. Af praktiske grunde er det med de iterative metoder ikke muligt at rekonstruere billeder af en realistisk størrelse på det anvendte computer udstyr. Hvis der skal bruges samme størrelse sinogrammer og billeder som i del I af rapporten, vil transformations matricen kræve over 80 MB hukommelse. Der er valgt den største sinogram- og billedstørrelse det var praktisk muligt at arbejde med.

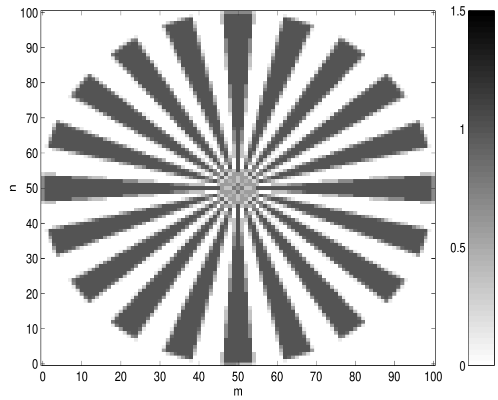
Sinogrammerne der rekonstrueres, er lavet med en størrelse på (125×100) samples i ρ- hhv. θ-retningen. Disse rekonstrueres til billeder med en dimension på (101×101) punkter. På figurene 7.1 til 7.3, er vist referencebillederne for de tre brugte fantomer. Disse er alle vist med gråtoner, repræsenterende værdier i intervallet $[0; 1.5]$, således at overshoot fejl i rekonstruktionen kan visualiseres. Ved generering af alle fantomer og sinogrammer er der blevet anvendt oversampling. Billeder i dette kapitel er alle præsenteret med deres numeriske værdier, dvs. negative værdier vises som positive for at lette detekteringen af baggrundsstøj med den brugte gråtone-mapning af billedværdierne.



Figur 7.1: *man* fantom.



Figur 7.2: *phantom* fantom.



Figur 7.3: *radial* fantom.

7.3 Valg af transformations matrix

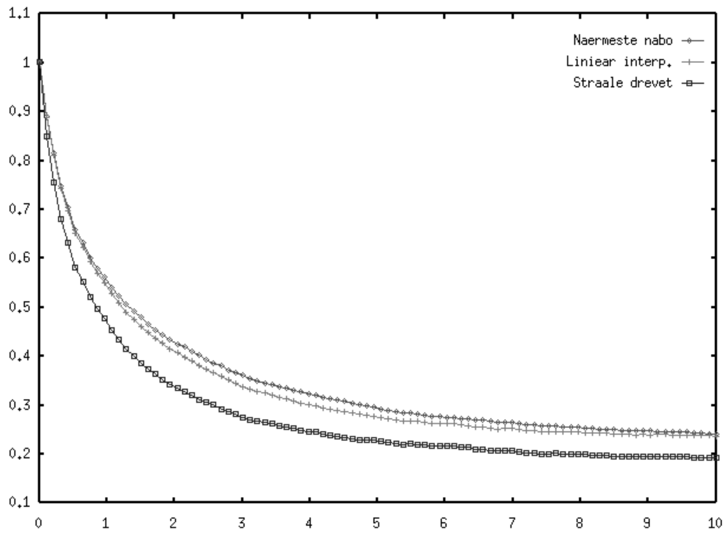
Som omtalt i afsnit 2.3, vil valget af transformations matricen være med til at bestemme hvilken nøjagtighed en given løsning til rekonstruktionsproblemet kan findes med. Der er blevet implementeret tre måder hvormed denne matrix kan udregnes. Disse er baseret på

- Nærmeste nabo approksimation,

- Lineær interpolation,
- Stråle-drevet approksimation.

Da ART algoritmen er den mest følsomme algoritme over for opbygningen af transformationsmatricen (se Shepp og Vardi [She82]), vil denne blive brugt til at vurdere metoderne med.

Der er blevet lavet et forsøg med ART, hvor hver af de tre metoder til generering af transformationsmatricen er blevet afprøvet. Alle elementer med en værdi over 0.05 er medtaget i matricen. Dæmpningsparameteren var i alle tre forsøg konstant, $\lambda_k = 0.1$. Det udregnede L_2-mål er vist på figur 7.4, som funktion af antallet af iterationer. Som det ses, er stråledrevet generering af matricen den metode der giver det mindste fejlmål. En visuel inspektion af de rekonstruerede billeder viser, at ved lineær interpolation vil der forekomme en anelse flere højfrekvente rekonstruktionsfejl i billedet, end ved stråledrevet approksimation. Nærmeste nabo approksimation har en normeringsfejl, der skyldes den grove tilnærmelse der foretages ved at sætte alle billedelementernes andel til samme konstant, bare de krydser linierne.

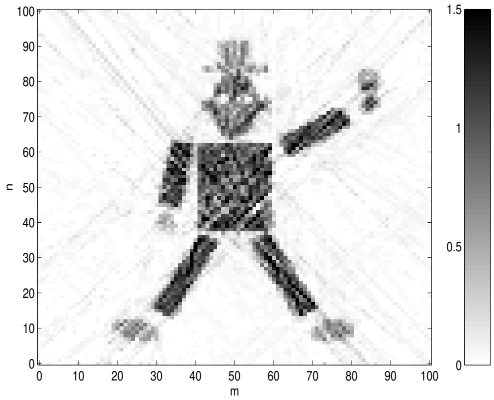


Figur 7.4: L_2-mål for ART ($\lambda_k = 0.1$), som funktion af antallet af iterationer k, med forskellige transformations matricer.

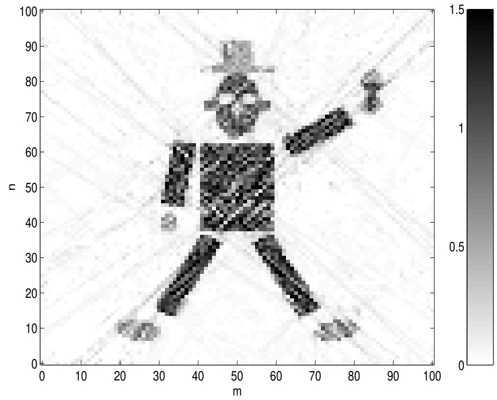
Fremover vil alle forsøg blive kørt med en transformations matrix genereret med stråle drevet interpolation, hvor alle værdier over 0.05 er medtaget. Med de valgte billeddimensioner, får transformations matricen dimensionerne (12.500 × 10201), hvor 1.684.882 (1.32%) af de 127.512.500 elementer har en værdi over 0.05 ved rekonstruktion af *man* fantomet. Matricen fylder omkring 14 MB, lagret med sparse strukturer.

7.4 Valg af dæmpningsparameter for ART

Som omtalt i kapitel 3, er dæmpningsparameteren λ_k i ART opdaterings algoritmen meget vigtig for konvergensen af denne. På figur 7.5 og 7.6 er vist rekonstruktionen af man fantomet med ART algoritmen med en dæmpningsparameter $\lambda_k = 1.0$. Billederne er gemt efter første hhv. tiende iteration. Som det ses, findes der allerede efter første iteration et godt gæt på den korrekte løsning. Dette gæt bliver dog ikke meget bedre derefter, da de opdateringer der foretages er så store, at løsningen ikke kan indkredses ordentligt. Som det ses, er begge billeder præget af højfrekvent støj, og kanterne i fantomet er aftegnet meget skarpt. Med en høj dæmpningsparameter kan de høje spatiale frekvenser derfor estimeres hurtigt, mens de lave frekvenser ikke kan bestemmes.

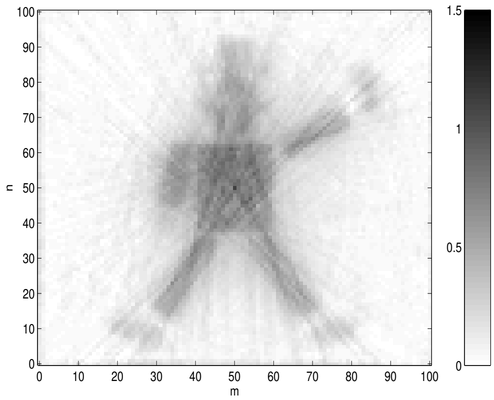


Figur 7.5: ART for $k = 1$, $\lambda_k = 1.0$.

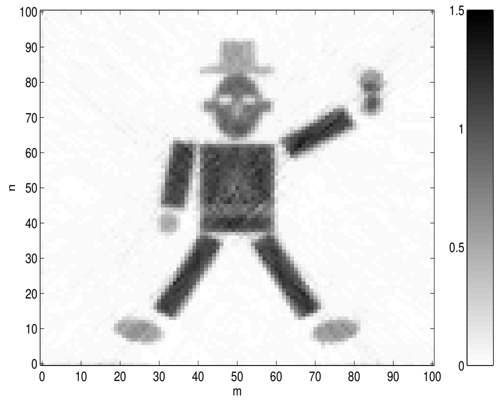


Figur 7.6: ART for $k = 10$, $\lambda_k = 1.0$.

På figur 7.7 og 7.8 er vist billeder for første hhv. tiende iteration for ART med konstant lav dæmpningparameter $\lambda_k = 0.05$. Med denne dæmpning ses, at billedet konvergerer langsommere end for $\lambda_k = 1.0$. Til gengæld vil den løsning der konvergeres mod blive bestemt med større nøjagtighed, da de foretagne opdateringer er relativt små. Efter tiende iteration, kan trekanten på maven ses, men kanterne er mere uskarpe end efter én iteration med $\lambda_k = 1.0$.

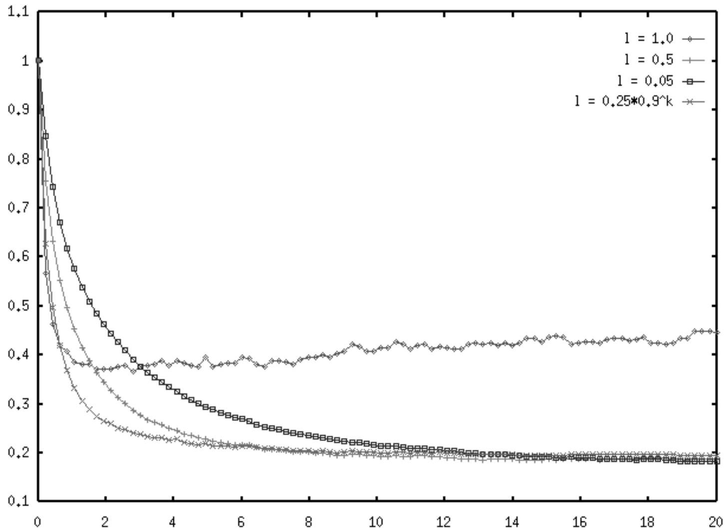


Figur 7.7: ART for $k = 1$, $\lambda_k = 0.05$.

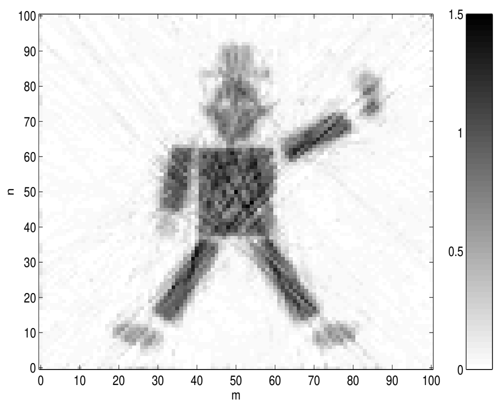


Figur 7.8: ART for $k = 10$, $\lambda_k = 0.05$.

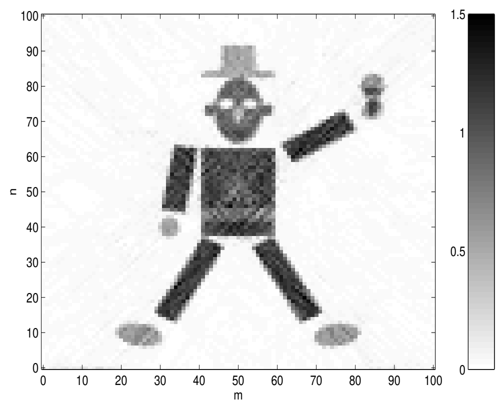
Ingen af de to valgte dæmpningsparametre er specielt optimale, og et kompromis mellem rekonstruktionskvalitet og konvergenshastighed bør findes. En mulighed er at bruge en høj dæmpning i de første iterationer for at få et rimeligt estimat for kanterne, og derefter lavere dæmpning, således at de små strukturer kan blive estimeret korrekt. På figur 7.9 er afbildet L_2-målet for rekonstruktionen af *man* fantomet som funktion af antallet af iterationer med forskellige λ_k. En eksperimentielt valgt dæmpningsfaktor $\lambda_k = 0.25 \cdot 0.9^k$, har vist sig at være et godt kompromis mellem god rekonstruktionskvalitet og konvergenshastighed. Efter fire iterationer har rekonstruktion med denne dæmpningsfaktor et mindre L_2-mål end efter 12 iterationer med dæmpningsfaktoren $\lambda_k = 0.05$. På figur 7.7 og 7.8 er vist de rekonstruerede billeder for $\lambda_k = 0.25 \cdot 0.9^k$ for første hhv. tiende iteration. Det ses, at det resulterende billede både har skarpt aftegnede kanter, og at de små strukturer findes pænt. Denne dæmpningsfaktor er brugt for alle efterfølgende tests med ART.



Figur 7.9: L_2-mål for ART ($\lambda_k = 0.1$), som funktion af antallet af iterationer k, med forskellige transformations matricer.



Figur 7.10: ART for $k = 1$, $\lambda_k = 0.25 \cdot 0.9^k$.



Figur 7.11: ART for $k = 10$, $\lambda_k = 0.25 \cdot 0.9^k$.

7.5 Stopkrav

Alle de iterative algoritmer forventes at ville konvergere mod en eller andet ønsket løsning. Hvorledes bestemmes hvilken løsning der skal vælges som den endelige? Ved at opstille et mål for rekonstruktionernes godhed udregnet fra et referencebillede, kan der testes på ændringen af dette mål fra en iteration til den næste. Når denne ændring kommer under en minimumsværdi, eller hvis den vender, stoppes der. Dette er selvsagt svært at automatisere, specielt ved rigtige studier, hvor der ikke er referencebilleder at sammenligne med.

Der er ikke indbygget nogen form for stopkrav i de implementerede rutiner. Ved alle kørsler med de iterative rutiner, er der blevet anvendt både fejlmål og objektiv vurdering af billederne efter hver iteration, til at bestemme den iteration der repræsenterer den bedste afvejning mellem rekonstruktionskvalitet (billedets udseende og L_2-mål) og hastighed.

7.6 Sammenligning

I de efterfølgende afsnit, vil de tre rekonstruktions algoritmers rekonstruktionskvalitet blive sammenlignet, dels med hinanden, dels med rekonstruerede billeder fra en Filtered Backprojection rutine. Sammenligningen vil blive foretaget ved hjælp af L_2-målet, og ved visuel inspektion.

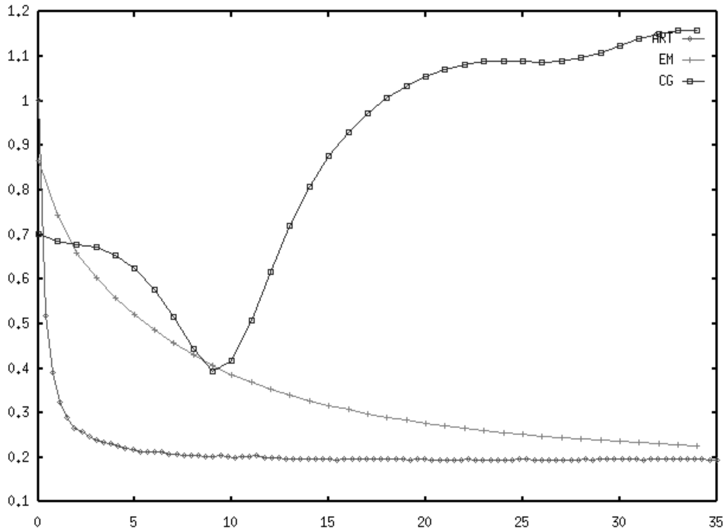
7.6.1 Rekonstruktion af man fantomet

Det analytisk genererede billede af *man* fantomet ser ud som på figur 7.1. Dette er blevet rekonstrueret med de tre iterative rekonstruktions algoritmer. L_2-målene for de 35 første iterationer er vist på figur 7.12. De rekonstruerede billeder repræsenterende den 35'ende iteration hhv. den iteration der er valgt som bedste kompromis mellem rekonstruktionskvalitet/hastighed for de forskellige rutiner, er vist fra figur 7.13 til 7.18. På figur 7.19 er vist et billede rekonstrueret med en Filtered Backprojection rutine.

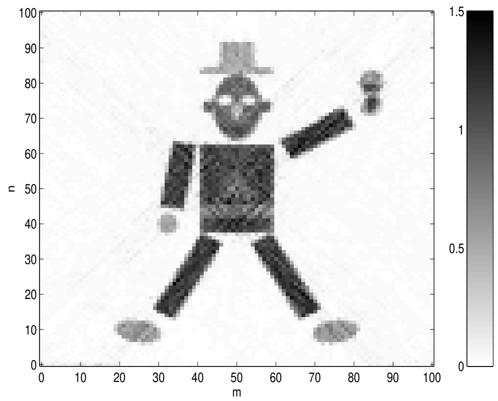
Det ses på figur 7.12 at ART algoritmen konvergerer meget hurtigere end EM algoritmen, efter 7 iterationer er nået et L_2-mål der ikke bliver forbedret meget ved at fortsætte iterationerne. EM algoritmen derimod konvergerer langsomt, og vil først efter 100 iterationer have nået samme L_2-mål som ART efter 7 iterationer. CG algoritmen, konvergerer mod en løsning som på figur 7.17, hvorefter den divergerer og begynder at introducere artifakter i rekonstruktionen.

Som endeligt estimat for de tre algoritmer er derfor valgt iterationerne 7, 35 og 9 for ART, EM hhv. CG, som repræsentation for den bedste rekonstruktion i forhold til antallet af iterationer for hver algoritme. De opnåede L_2-mål for de iterative rekonstruktions algoritmer, og for rekonstruktion med en filtered backprojection rutine, er:

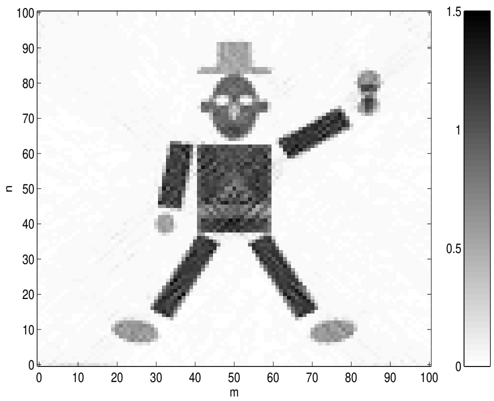
	k	L_2
ART	7	0.207
ART	35	0.195
EM	35	0.227
EM	100	0.195
CG	9	0.395
FB		0.149



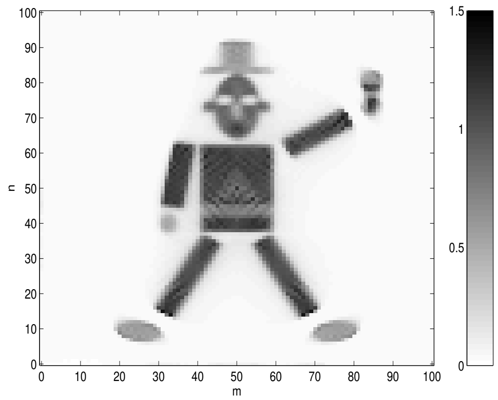
Figur 7.12: L_2-mål som funktion af antallet af iterationer k, for de tre iterative rekonstruktions algoritmer, ved rekonstruktion af *man* fantomet.



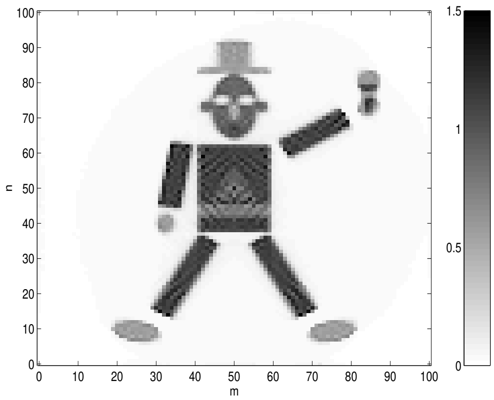
Figur 7.13: ART for $k = 7$.



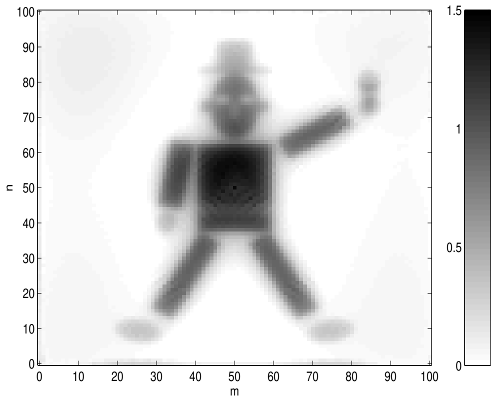
Figur 7.14: ART for $k = 35$.



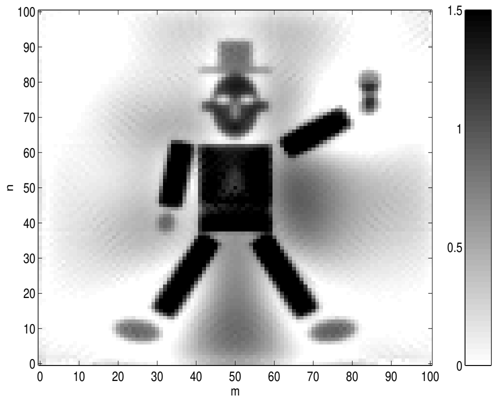
Figur 7.15: EM for $k = 35$.



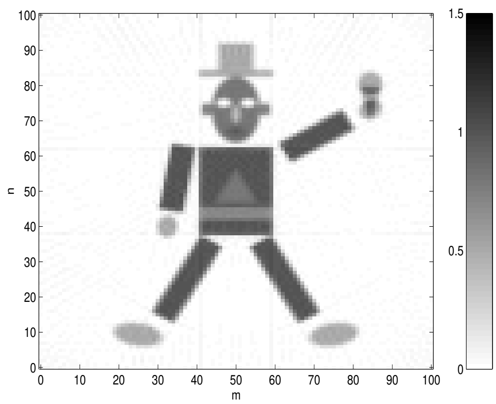
Figur 7.16: EM for $k = 100$.



Figur 7.17: CG for $k = 9$.



Figur 7.18: CG for $k = 35$.



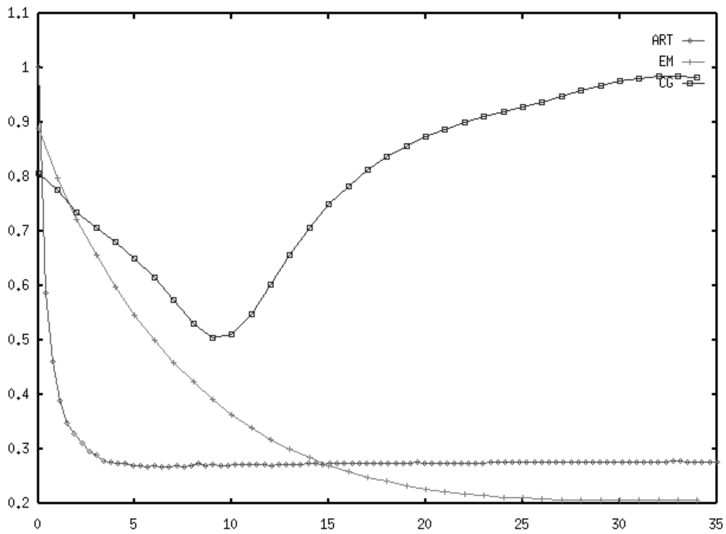
Figur 7.19: Filtered Backprojection.

7.6.2 Rekonstruktion af phantom fantomet

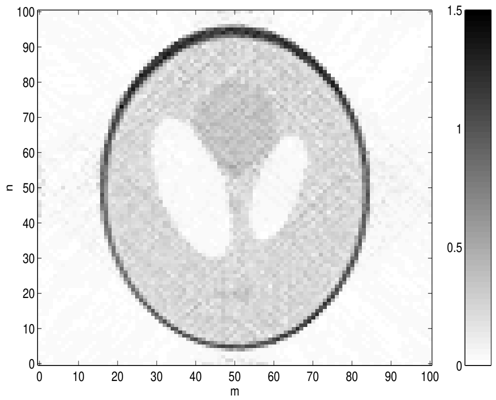
Ved rekonstruktion af *phantom* fantomet, opnår de iterative algoritmer en visuel kvalitet der ligger tæt op af Filtered Backprojection rutinens. De rekonstruerede billeder er vist fra figur 7.21 til 7.24, og på figur 7.20 er vist L_2-målet som funktion af antallet af iterationer for de første 35 iterationer.

Fejlmålet for EM algoritmen er mindst efter 35 iterationer, og efter 7 iterationer fås ikke bedre fejlmål med ART algoritmen. Fejlmålet for CG har efter 11 iterationer et minimum. De iterative algoritmer er i stand til at rekonstruere dette fantom med en kvalitet der ligger meget tæt op ad Filtered Backprojection rekonstruktionen. Fejl målene for de rekonstruktioner der er valgt som endelige er:

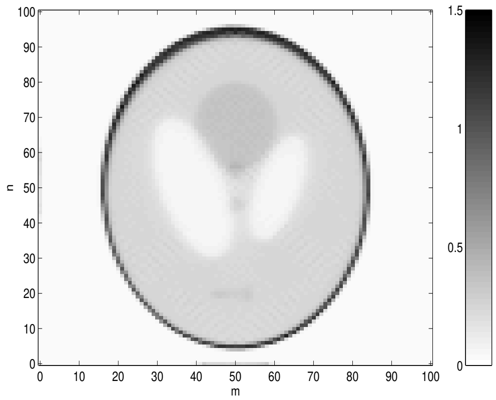
	k	L_2
ART	7	0.272
EM	35	0.208
CG	11	0.510
FB		0.149



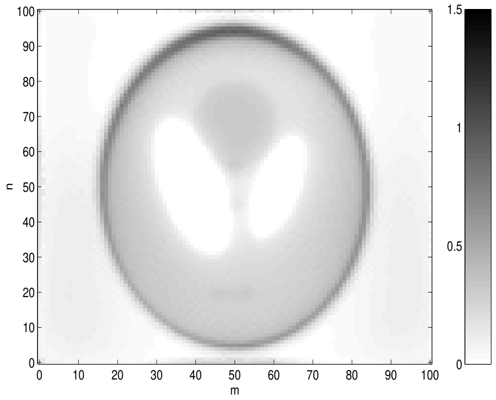
Figur 7.20: L_2-mål som funktion af antallet af iterationer k, for de tre iterative rekonstruktions algoritmer, ved rekonstruktion af *phantom* fantomet.



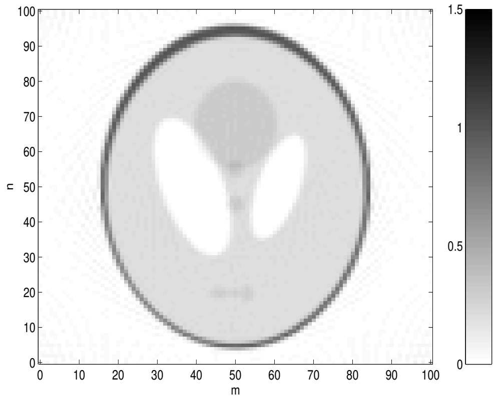
Figur 7.21: ART for $k = 7$.



Figur 7.22: EM for $k = 35$.



Figur 7.23: CG for $k = 11$.



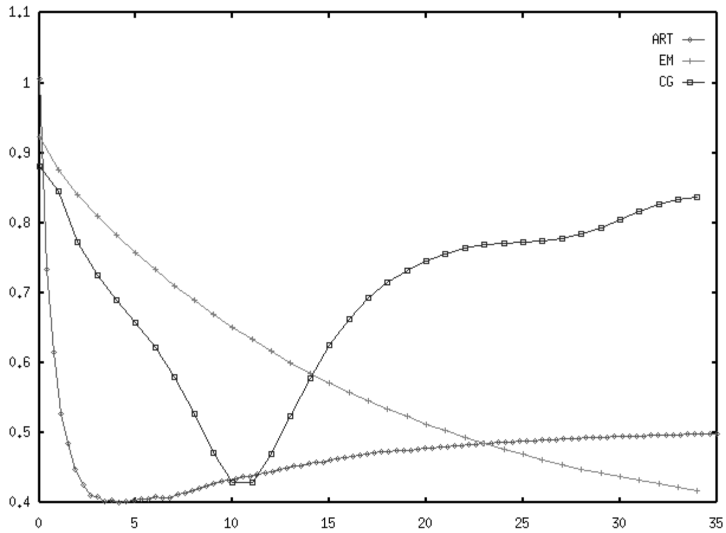
Figur 7.24: Filtered Backprojection.

7.6.3 Rekonstruktion af radial fantomet

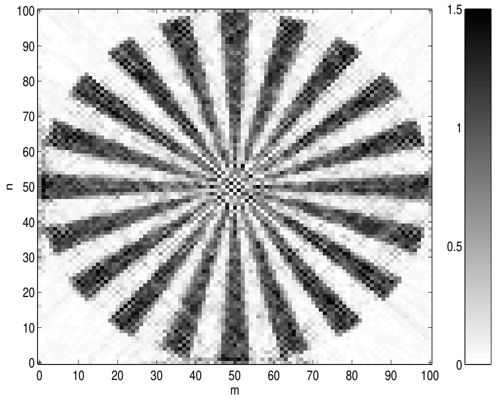
De iterative rekonstruktions algoritmer, har forholdsvis sværere ved at rekonstruere dette fantom, i forhold til Filtered Backprojection rutinen. Dette ses både på den visuelle kvalitet af billederne og på L_2-målene. De rekonstruerede billeder er vist fra figur 7.26 til 7.29. På figur 7.25 er vist L_2-målet som funktion af antallet af iterationer for de første 35 iterationer.

EM algoritmen når først et minimum for L_2-målet efter 55 iterationer, hvorefter dette begynder at stige. ART algoritmen når et minimum efter fire iterationer, og divergerer derefter væk fra denne løsning. Disse effekter, der ikke er set ved rekonstruktion af de andre fantomer, kan skyldes *radial* fantomets specielle udseende med høje spatiale frekvenser. EM algoritmen rekonstruerer det billede der visuelt er det bedste bud på en løsning, og opnår det bedste L_2-mål af de iterative rutiner. De opnåede L_2-mål for rekonstruktion af dette fantom er:

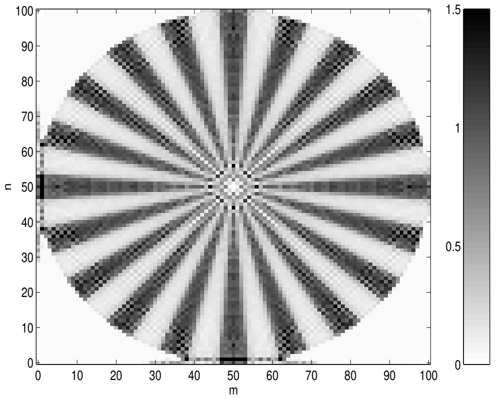
	k	L_2
ART	4	0.405
EM	55	0.380
CG	11	0.430
FB		0.208



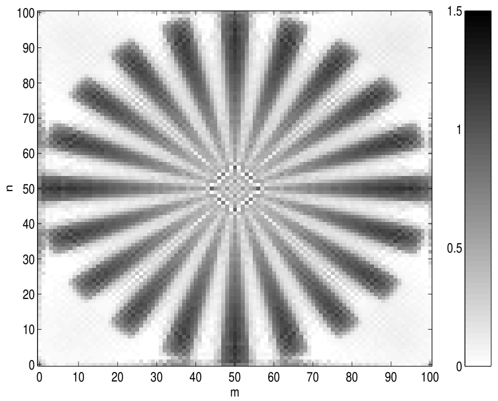
Figur 7.25: L_2-mål som funktion af antallet af iterationer k, for de tre iterative rekonstruktions algoritmer, ved rekonstruktion af *radial* fantomet.



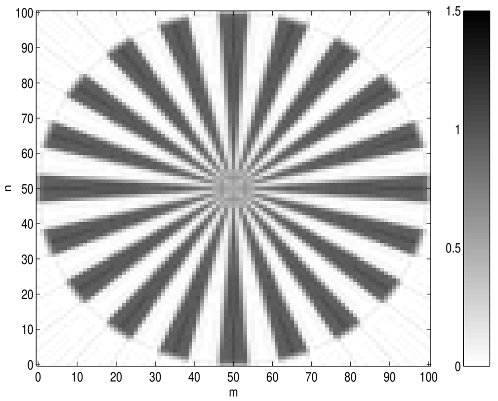
Figur 7.26: ART for $k = 4$.



Figur 7.27: EM for $k = 55$.



Figur 7.28: CG for $k = 11$.

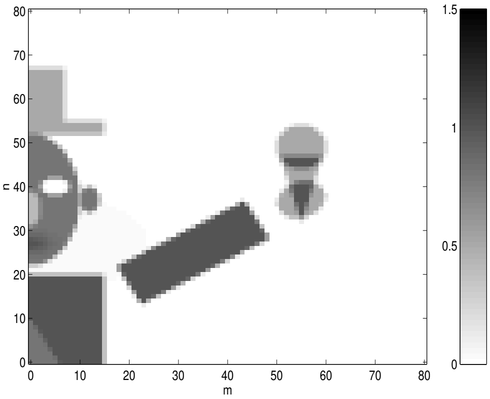


Figur 7.29: Filtered Backprojection.

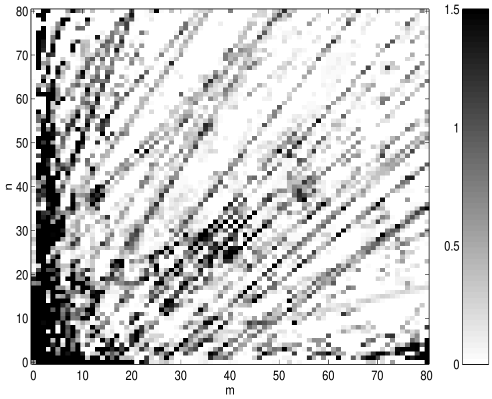
7.7 Zoom i outputbilledet

Der er lavet et forsøg på at opnå zoom i outputbilledet med de iterative metoder. På figur 7.30, er vist det interesseområde af *man* fantomet der ønskes rekonstrueret. På figur 7.31 til 7.33 er vist de bedste estimater de iterative algoritmer fandt frem til.

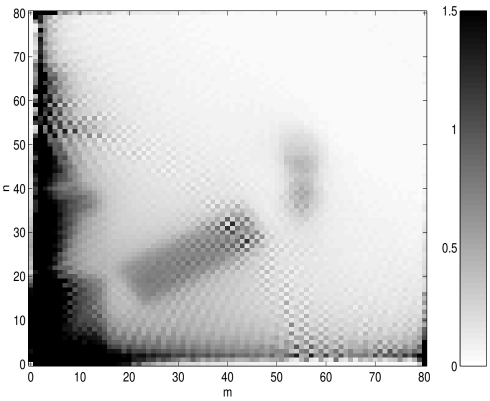
Ligningssystemet vil, hvis interesseområdet ikke dækker hele det objekt projektionerne er udregnet fra, blive meget overbestemt. Det ses på figur 7.31 at ART opdateringerne fejler fra starten af, hvilket skyldes den måde algoritmen arbejder på. Det er ikke muligt at fremadprojektere de fundne estimater således at al informationen i sinogrammet udnyttes. På billederne fra EM og CG algoritmen genfindes strukturen af fantomet, men denne er overlejret af megen støj. Dette er tildels forventet, da ligningssystemet bliver meget singulært ved at fokusere på et mindre område, end det de fysiske projektioner er optaget fra.



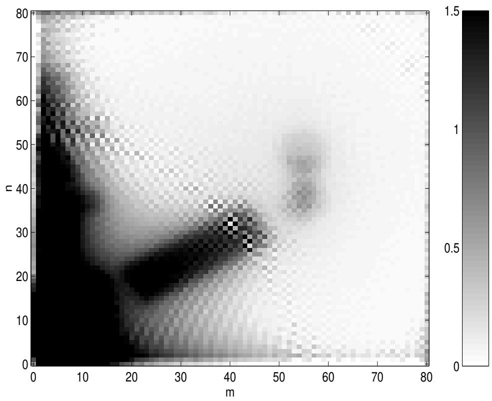
Figur 7.30: Interesseområde.



Figur 7.31: ART for $k = 2$.



Figur 7.32: EM for $k = 6$.



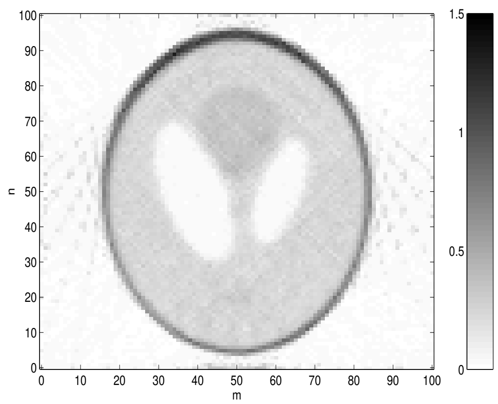
Figur 7.33: CG for $k = 6$

7.8 Rekonstruktion ud fra få projektioner

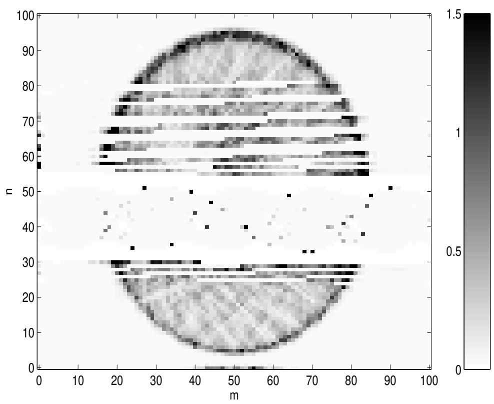
I følge litteraturen er en af de iterative rekonstruktionsmetoders fortrin fremfor de direkte metoder, at de er bedre til at rekonstruere billeder, der er optaget med få projektioner. For at undersøge dette er der kørt to forsøg med *phantom* fantomet. Dette er blevet rekonstrueret fra sinogrammer med $T = 20$ hhv. $T = 10$ samplinger af θ. Samplingen i ρ-retningen er uforandret $R = 125$. På figur 7.34 til 7.37 er vist rekonstruktionerne for $T = 20$, for de tre iterative rekonstruktions algoritmer, og for Filtered Backprojection rutinen (FB). På figur 7.38 til 7.41 er vist rekonstruktionerne for $T = 10$. For alle rekonstruktionerne gælder, at der er valgt den iteration der gav det bedste L_2-mål i forhold til antallet af iterationer. Fejlmålene er for rekonstruktionerne udregnet til:

	$T = 10$		$T = 20$	
	k	L_2	k	L_2
ART	40	0.556	40	0.335
EM	1	0.991	20	1.087
CG	10	0.626	10	0.563
FB		0.866		0.613

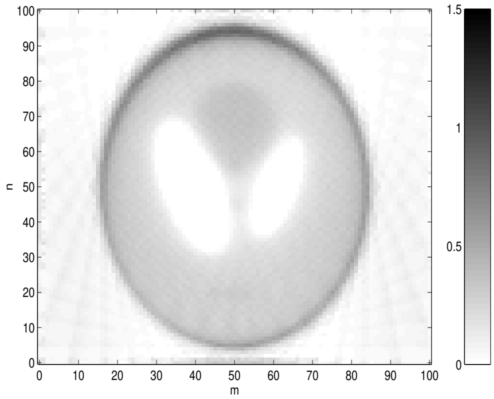
EM algoritmen er ikke i stand til at rekonstruere disse sinogrammer tilfredsstillende. For $T = 20$ har L_2-målet et minimum ved 20'ende iteration hvorefter der konvergeres mod en helt anden løsning. På billederne for EM ses, at nogle af værdierne i det rekonstruerede billede er nulstillet. Dette kan tyde på, at der forekommer problemer med division med nul i EM opdateringen, ved rekonstruktion af få projektioner. EM algoritmen bør derfor ændres til ikke at sætte værdier i outputbilledet til nul ved division med nul, men i stedet initialisere dem til det udregnede estimat på middelværdien $\bar{x}$. Dette er ikke blevet afprøvet. For $T = 10$ er EM algoritmen ikke i stand til at rekonstruere fantomet. ART og CG algoritmerne konvergerer som før, og som det ses, vil disse for både $T = 20$ og $T = 10$ lave billeder der har mindre visuelle artifakter og mindre L_2-mål end Filtered Backprojection rutinens rekonstruktioner.



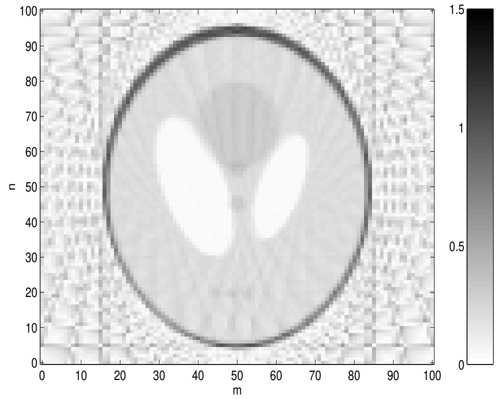
Figur 7.34: $T = 20$, ART for $k = 40$.



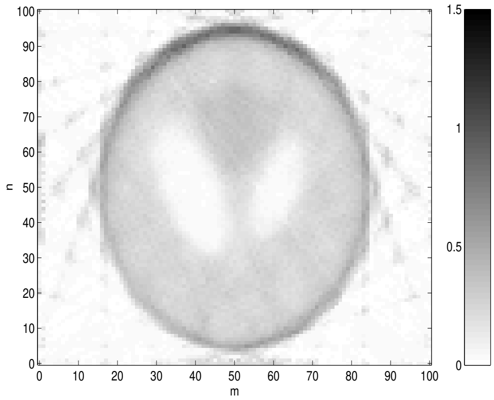
Figur 7.35: $T = 20$, EM for $k = 20$.



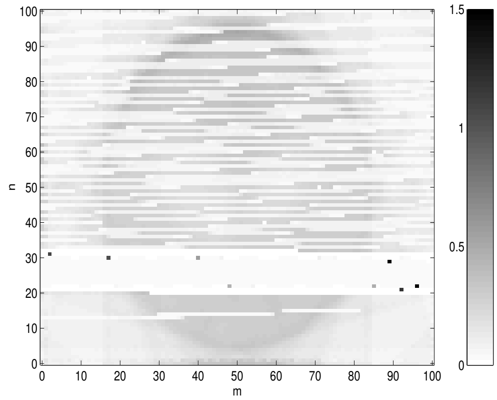
Figur 7.36: $T = 20$, CG for $k = 10$.



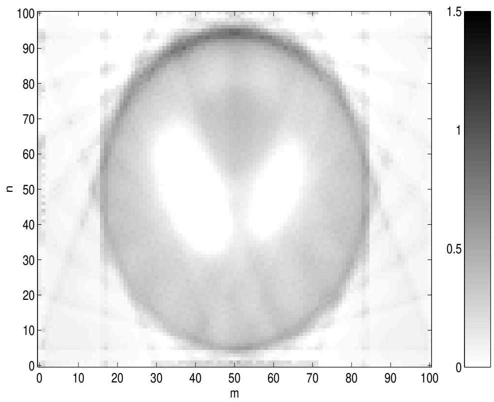
Figur 7.37: $T = 20$, FB.



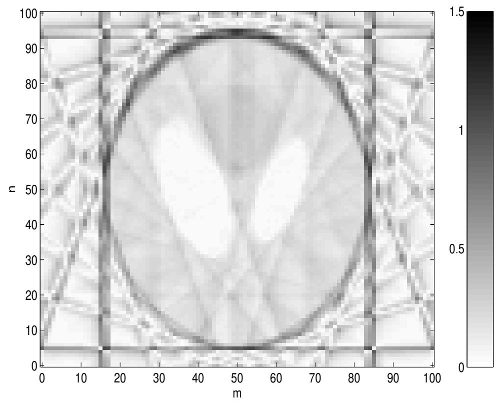
Figur 7.38: $T = 10$, ART for $k = 40$.



Figur 7.39: $T = 10$, EM for $k = 1$.



Figur 7.40: $T = 10$, CG for $k = 10$.



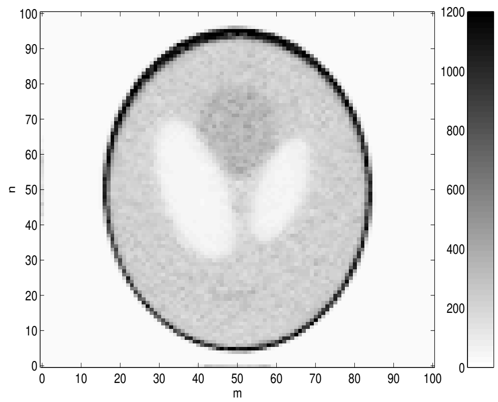
Figur 7.41: $T = 10$, FB.

7.9 EM rekonstruktion af projektioner med støj

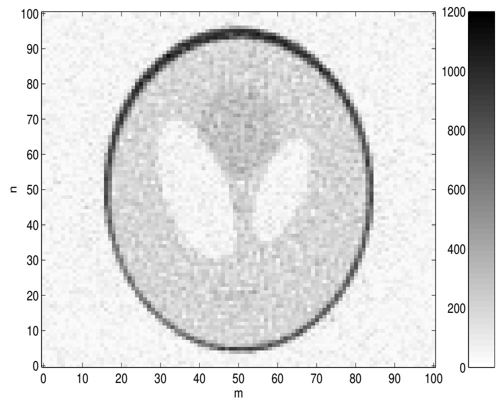
Da EM algoritmen tager udgangspunkt i en Poisson model for den emitterede stråling (se bla. Shepp og Vardi [She82]), må det forventes, at denne vil være i stand til at rekonstruere bedre billeder end f.eks. de direkte metoder, hvis der rekonstrueres virkelige data, der kan modelleres som resultatet af en Poisson-proces i hver pixel. Det er ikke muligt at rekonstruere sinogrammer fra Rigshospitalets skanner med de iterative metoder på det benyttede computerudstyr, grundet størrelsen af disse. Derfor er der blevet produceret tre sinogrammer der simulerer rigtigt opsamlede data, som en individuel Poisson-proces for hver pixel. Disse er genereret ud fra sinogrammet af *phantom* fantomet. De tre sinogrammer har størrelsen (125×100) punkter, og er simuleringer af forskellige tælletal c i sinogrammet: $2.5 \cdot 10^6$, $1.25 \cdot 10^6$ og $0.25 \cdot 10^6$. På billederne fra figur 7.42 til 7.47 ses de af EM hhv. Filtered Backprojection (FB) rutinen rekonstruerede billeder. Der er for EM valgt den iteration der gav det mindste L_2-mål. L_2-målene er for de viste billeder:

	$c = 2.5 \cdot 10^6$		$c = 1.25 \cdot 10^6$		$c = 0.25 \cdot 10^6$	
	k	L_2	k	L_2	k	L_2
EM	35	0.205	29	0.232	18	0.350
FB		0.309		0.395		0.708

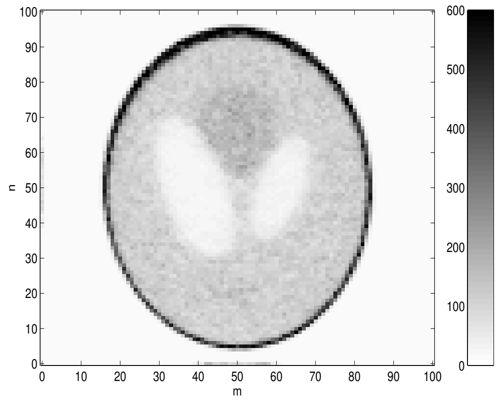
Det ses på billederne, at rekonstruktionerne med EM algoritmen generelt er de mindst støjfyldte, og at det er nemmere på disse at skelne de små strukturer i fantomet. I tabellen ovenfor ses, at EM algoritmen producerer billeder med et mindre L_2-mål end Filtered Backprojection algoritmen i alle tre tilfælde. Det kan konkluderes, at EM algoritmen er i stand til at reducere den statistiske støj i rekonstruktionerne i forhold til de direkte metoder uden at bruge ekstra filtrering.



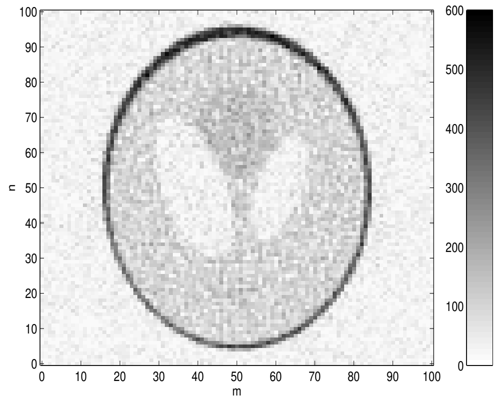
Figur 7.42: $c = 2.5 \cdot 10^6$, EM for $k = 35$.



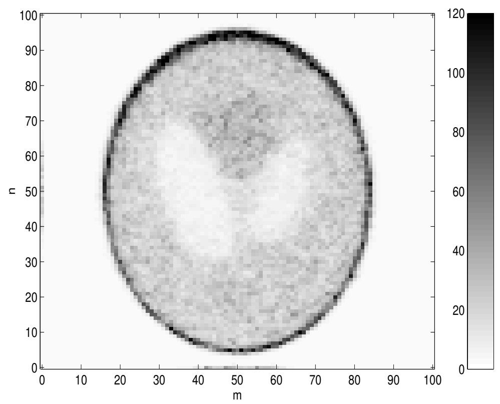
Figur 7.43: $c = 2.5 \cdot 10^6$, FB.



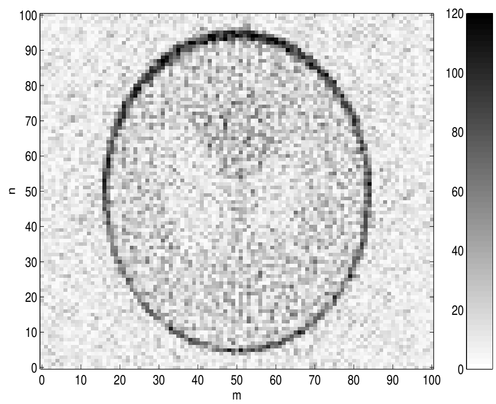
Figur 7.44: $c = 0.125 \cdot 10^6$, EM for $k = 29$.



Figur 7.45: $c = 1.25 \cdot 10^6$, FB.



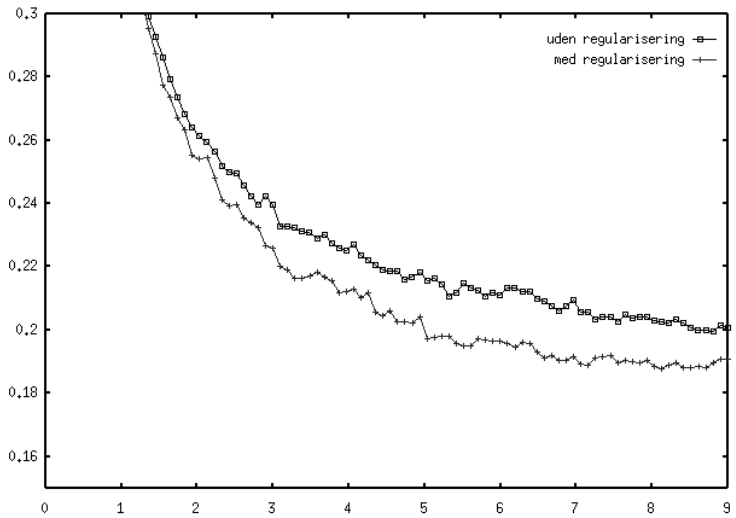
Figur 7.46: $c = 0.25 \cdot 10^6$, EM for $k = 18$.



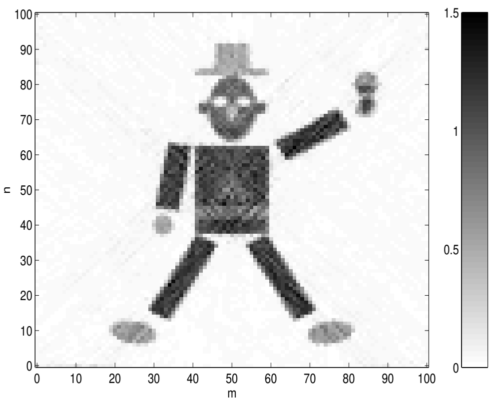
Figur 7.47: $c = 0.25 \cdot 10^6$, FB.

7.10 ART rekonstruktion med regularisering

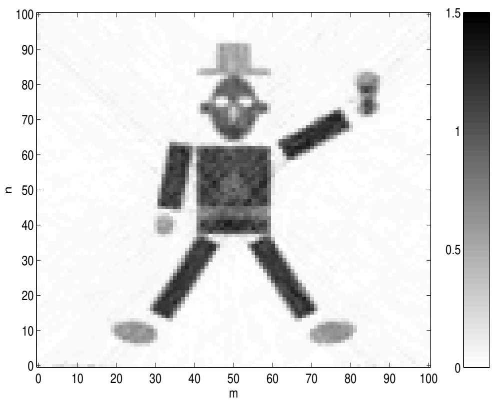
Der er foretaget en kørsel med ART algoritmen med og uden regularisering, for at se om dette vil hjælpe på konvergensen af denne. På figur 7.48 er vist L_2-målet for ART rekonstruktion af *man* fantomet med og uden regularisering. For begge kørsler var regulariseringsparameteren λ sat til 1. På figur 7.49 og 7.50 er vist de respektive billeder. Det ses, at indføring af regulariseringsleddet, som ventet, har en dæmpende effekt på de høje frekvenser i billedet. L_2-målet er for rekonstruktionen med regularisering en anelse mindre end uden regularisering. Fejlmålene efter ni iterationer blev for ART rekonstruktion uden regularisering 0.201, og med regularisering 0.191. Regularisering har dermed en dæmpende effekt på de høje spatiale frekvenser i det rekonstruerede billede, og vil have en gavnlig effekt på kvaliteten af de rekonstruerede billeder fra ART algoritmen. Den brugte regularisering var den i afsnit 6.1 omtalte Laplace operator.



Figur 7.48: L_2-mål som funktion af antallet af iterationer k, ved rekonstruktion af *man* fantomet med ART, med og uden regularisering. Bemærk den ændrede inddeling af akserne.



Figur 7.49: ART for $k = 9$ uden regularisering.



Figur 7.50: ART for $k = 9$ med regularisering

7.11 Rekonstruktionshastighed

De tre iterative rekonstruktion algoritmers rekonstruktionshastighed er afhængig af flere faktorer, ikke mindst den tid det tager at udregne én iteration, og hvor mange iterationer der skal udføres, før et fornuftigt estimat til løsningen er nået. Alle nedenstående tal er målt for rekonstruktion fra det anvendte (125×100) punkters sinogram til et billede på (101×101) punkter. De tre algoritmers tidsforbrug t, i sekunder pr. iteration er målt til:

	t/s
ART	2.4
EM	2.0
CG	2.1

Udover selve iterationerne, skal den tid der bruges på at udregne transformations matricen medregnes. For de brugte sinogram- og billedstørrelser, tager udregningen af denne ca. 90 sekunder. Denne matrix udregnes dog kun én gang, og lagres for en given transformationsgeometri. Indlæsningen af matricen fra disk (14 Mb), kan dog tage op til 10 sekunder. Det forventede tidsforbrug ved implementering med den 'langsomme' metode som beskrevet i afsnit 2.6, kan estimeres ved at se på opdateringsformlerne. EM og CG behøver to hele udregninger af $\mathbf{A}$ for hver iteration, da de både opererer på $\mathbf{A}$ og $\mathbf{A}^T$. ART behøver kun én udregning pr. iteration. Hvis det antal iterationer der skal udføres for at få en god rekonstruktion bestemmes ud fra kørslerne på de tre fantomer, som 7, 35 og 9 iterationer for ART, EM hhv. CG, vil den samlede rekonstruktionstid, T_1 og T_2 for de implementerede algoritmer (målt) hhv. for den alternative implementering (estimeret) blive:

	T_1/s	T_2/s
ART	16.6	646
EM	70.0	6.370
CG	19.9	1.640

Disse tider kan sammenlignes med tiderne for rekonstruktion med de direkte metoder fra rapportens del I, hvor de målte rekonstruktionstider t er:

	t/s
FB	2.6
FAB	4.3
CSBL	3.1
CSCH	10.2

hvor CSBL og CSCH er central slice rekonstruktion med bi-lineær hhv. chirp-z interpolation. Det ses, at filtered back-projection rutinen er i stand til at rekonstruere det samme billede over 25 gange hurtigere end EM algoritmen. Rekonstruktion med ART algoritmen er hurtigere men opnår ikke helt så gode resultater som EM algoritmen. Alle kørsler er foretaget på en 486DX4-100MHz baseret computer, med LINUX som operativ system. Programmerne er alle kompileret med GNU-C version 2.5.8, med optimeringsparameteren `-O2`.

7.12 Vurdering af metoderne

Ud fra de udførte rekonstruktioner af fantom billeder, må det konkluderes, at den implementerede EM algoritme er den af de tre afprøvede iterative rekonstruktions algoritmer der producerer de bedste rekonstruktioner. ART algoritmen konvergerer mange gange hurtigere end EM, men kunne ikke rekonstruere billeder med en helt så høj visuel kvalitet. CG algoritmen var ikke i stand til at lave billeder af samme kvalitet som de to andre algoritmer. Ingen af de tre iterative rutiner er praktisk anvendelige på ordinært computerudstyr grundet deres store hukommelseskrav. Dette betyder, at kun begrænsede billede størrelser er blevet rekonstrueret i forhold til testene foretaget i del I af rapporten. En implementering af den omtalte 'langsomme' metode til iterativ rekonstruktion, vil dog være i stand til at rekonstruere vilkårlige billedstørrelser på bekostning af beregningstiden. En ændring af de implemplementerede rutiner således at de arbejder efter den 'langsomme' model vil ikke kræve meget. Beregningshastigheden for de iterative metoder har vist sig ikke at være så stort et problem som antaget ved den valgte implementering.

Der er i litteraturen blevet præsenteret flere måder at opnå hurtigere konvergens med EM algoritmen på og således gøre den mere anvendelig. Kaufmann [Kau87] viser en accelereret EM algoritme, der ydermere ved brug af en alternativ sampling af det rekonstruerede billede, kan opnå en hukommelsesbesparelse på en faktor otte over almindelig EM.

Ingen af de tre algoritmer er i stand til at rekonstruere de valgte fantomer med et L_2-mål bedre end en Filtered Backprojection rutine under normale omstændigheder. Dog var den visuelle kvalitet af især de af EM rutinen rekonstruerede billeder helt på højde med de rekonstruerede billeder fra Central Slice- og Filtered Backprojection baserede rutiner, grundet fraværet af de systematiske artifakter der mange gange forekommer i billeder rekonstrueret med direkte metoder. Ved rekonstruktion af sinogrammer, genereret så de simulerer rigtige sinogrammer som en individuel Poisson-proces i hvert punkt, har EM algoritmen genereret billeder med en meget bedre visuel kvalitet og mindre L_2-mål end rekonstruktion med Filtered Backprojection. Dette taler for EM algoritmens anvendelse på rigtige PET data, da disse formodes at have denne karakter.

Ved rekonstruktion af sinogrammer med få projektioner, er især ART algoritmen i stand til at lave billeder, der er de direkte metoder overlegne, både med hensyn til fejlmål og visuel kvalitet.

CG algoritmen har vist samme opførsel igennem alle kørsler. Til at starte med er den selv-regulariserende effekt god, hvorefter fejl og artifakter/svingninger tager over og ødelægger løsningen. Dette er forventeligt da $\mathbf{A}^T\mathbf{A}$ er meget singulær. CG algoritmen i den implementerede version må derfor antages at have begrænset anvendelighed inden for PET rekonstruktion.

7.13 Resumé

I dette kapitel er de tests, der er blevet kørt med de implementerede iterative rekonstruktions algoritmer blevet præsenteret. Resultaterne er blevet kommenteret undervejs, og de iterative rekonstruktions algoritmers anvendelighed til rekonstruktion af PET billeder er blevet diskuteret.

Kapitel 8

Konklusion

Der er i denne del af projektet, blevet implementeret og undersøgt tre forskellige iterative algoritmer til rekonstruktion af PET billeder. Metoderne stiller rekonstruktionsproblemet op som et lineært ligningssystem, og itererer sig frem til en løsning af dette. Da der kun er den fysiske model til forskel mellem to- og tre dimensional rekonstruktion ved brug af iterative metoder, er alle de omtalte løsningsmetoder brugbare til tre-dimensional rekonstruktion.

Det vurderes, at især EM og ART algoritmerne vil være i stand til at opnå rekonstruktionskvalitet på højde med de direkte metoder, da der er mulighed for yderligere forbedringer af de i denne rapport brugte modeller og antagelser. Forbedringer vil bla. kunne opnås ved at bruge mere virkelighedstro modeller for den forventede emission i det rekonstruerede objekt, f.eks. baseret på en anden sampling af det rekonstruerede objekt end med kvadratiske billedelementer.

Ved rekonstruktion af virklighedstro sinogrammer konstrueret som en individuel Poissonproces i hver pixel, har EM algoritmen vist resultater der er væsentligt bedre end Filtered Backprojection algoritmens. Dette taler for anvendelsen af EM algoritmen på virkelige data.

I forbindelse med rekonstruktion ud fra få projektioner, er der med især ART algoritmen opnået resultater, der er overlegne de af Filtered Backprojection rekonstruerede billeder.

Det lykkedes ikke at få de forventede resultater ud af den implementerede CG algoritme, da denne i alle forsøgene konvergerede mod en mere eller mindre god løsning, hvorefter den divergerede.

Bibliografi

[Axe94] Caroline Axelsson. *Direct Fourier Methods in 3D-Reconstruction from Cone-Beam Data.* PhD thesis, Linköbing University, Dep. of Electrical Engineering, S-581 83 Linköbing, Sweden, Jan. 1994. E-mail caroline@isy.liu.se.

[Bro81] R. A. Brooks, V. J. Sanks et al. Design considerations for positron emission tomography. *IEEE Transactions on Biomedical Engineering*, BME–28(2):158–76, February 1981.

[Bud79] T. F. Budinger et al. Trends and prospects for circular ring positron cameras. *IEEE Transactions on Nuclear Science*, NS–26(2):2742–45, April 1979.

[Car85] R. E. Carson and K. Lange. The em parametric image reconstruction algorithm. *Journal of the American Statistical Association*, 80(389):20–5, March 1985.

[Cen83] Yair Censor. Finite series-expansion reconstruction methods. In *Proceedings of the IEEE*, volume 71, pages 409–19, March 1983.

[Che88] C-T. Chen et al. Maximum likelihood reconstruction in pet and tofpet. *Mathematics and Computer Science in Medical Imaging*, F–39:319–29, 1988.

[Che91] Simon R. Cherry, Magnus Dahlbom and Edward J. Hoffman. 3d pet using a conventional multislice tomograph without septa. *Journal of Computer Assisted Tomography*, 15(4):655–68, July 1991.

[Che92] Simon R. Cherry, Magnus Dahlbom amd Edward J. Hoffman. Evaluation of a 3d reconstruction algorithm for multi–slice pet scanners. *Physics in Medicine & Biology*, 37(3):799–90, March 1992.

[Cla92] R. Clack. Towards a complete description of three-dimensional filtered backprojection. *Physics in Medicine & Biology*, 37(3):645–60, March 1992.

[Col80] James G. Colsher. Fully three–dimensional positron emission tomography. *Physics in Medicine & Biology*, 25(1):103–15, January 1980.

[Dah89] Magnus Dahlbom et al. A study of the possibility of using a mulit–slice pet system for 3d imaging. *IEEE Transactions on Nuclear Science*, 36(1):1066–71, February 1989.

[Dea93] S. R. Deans. *The Radon Transform and Some of Its Applications.* Krieger Publishing Company, Malabar, Florida, 1993. Previously John Wiley & Sons Inc., 1983.

[Def89] M. Defrise, D. W. Townsend and R. Clack. Three-dimensional image reconstruction from complete projections. *Physics in Medicine & Biology*, 34(5):573–87, 1989.

[DeG94] Timothy R. DeGrado et. al. Performance characteristics of a whole-body pet scanner. *The Journal of Nuclear Medicine*, 35:1398–406, August 1994.

[Fre94] R. Freifelder, J. S. Karp et al. Design and preformance of the head penn-pet scanner. *IEEE conference records, Nuclear Science & Medical Imaging*, 1994.

[Gor70] R. Gordon, R. Bender and G. T. Herman. Algebraric reconstruction techniques (art) for the three–dimensional reconstruction af an object from projections. *Journal of Theoretical Biology*, 29:471–81, 1970.

[Gro88] S. I. Grossman and W. R. Derrick. *Advanced engineering mathematics.* Harper & Row, New York, 1988.

[Gua94] Huaiqun Guan and Richard Gordon. A projection access order for speedy convergenve of art (algebraic reconstruction technique): a multilevel scheme for computed tomography. *Physics in Medicine & Biology*, 39:2005–22, 1994.

[Gue93] T. M. Guerreroe et al. Evaluation and application of reprojection methods for 3d pet. *IEEE Conference records on Neuclear Science & Medical Imaging*, pages 1101–5, 1993.

[Han92] Per Christian Hansen. Regularization tools; a matlab package for analysis and solution of discrete ill-posed problems. Report UNIC-92-03, UNIC, UNIC, Technical University of Denmark, DK-2800 Lyngby, 1992.

[Her80] G. T. Herman. *Image Reconstruction from Projections: The fundamentals of Computerized Tomography.* Academic Press, 1980.

[Her93a] T. J. Herbert. A union of deterministic and stochastic methods for image reconstruction. *IEEE conference records, Nuclear Science & Medical Imaging*, pages 1317–21, 1993.

[Her93b] Gabor T. Herman and Lorraine B. Meyer. Algebraric reconstruction techniques can be made computationally efficient. *IEEE transactions on Medical Imaging*, 12(3):600–9, September 1993.

[Kau87] Linda Kaufmann. Implementing and accelerating the em algorithm for positron emission tomograhy. *IEEE Transactions on Medical Imaging*, MI-6(1):37–51, March 1987.

[Kau93] Linda Kaufmann. Maximum likelihood, least squares, and penalized least squares for pet. *IEEE Transsctions on Medical Imaging*, 12(2), June 1993.

[Ker91] Brian W. Kernighan and Dennis M. Ritchie. *The C programming language.* Prentice Hall, 2 edition, 1991.

[Kin89] P. E. Kinahan and J. G. Rogers. Analytic 3d image reconstruction using all detected events. *IEEE Transactions on Nuclear Science*, 36(1):964–8, February 1989.

[Kin93] Paul Kinahan and Joel Karp. Determining when 3d reconstruction methods are required in volume pet imaging. *IEEE Conference records, Nuclear Science and Medical Imaging*, 2:904–6, 1993.

[Kin94] Paul E. Kinahan, S. Matej, J. S. Karp, G. T. Herman and R. M. Lewitt. A comparison of transform and iterative reconstruction techniques for a volume imaging pet scanner with a large axial acceptance angle. *Physics in Medicine & Biology*, 39, 1994.

[Lew83] Robert M. Lewitt. Reconstruction algorithms: Transform methods. *Proceedings of the IEEE*, 71(3):390–408, March 1983.

[Lew92] R. M. Lewitt. Alternatives to voxels for image representation in iterative reconstruction algorithms. *Physich in Medicine & Biologi*, 37(3):705–16, March 1992.

[Lou83] Alfred K. Louis and Frank Natterer. Mathematical problems of computerized tomography. *Proceedings of the IEEE*, 71(3):379–89, March 1983.

[Mac83] Albert Macovski. Physical problems of computerized tomography. *Proceedings of the IEEE*, 71(3):373–8, March 1983.

[Mat90] Samuel Matej and Ivan Balja. A high-speed reconstruction from projections using direct fourier method with optimized parameters — an experimental analysis. *IEEE Transactions on Medical Imaging*, 9(4):421–29, December 1990.

[Med83] B. P. Medoff, W. R. Brody and A. Macovski. The use of a priori information in image reconstruction from limited data. *IEEE Conference paper, Acoustics and Speech Processing*, 1983.

[Mei94] Steven R. Meikle, Brian F. Hutton et al. Accelerated em reconstruction in total body pet: potential for improving tumour detectability. *Physics in Medicine & Biology*, 39:1698–1704, 1994.

[Nuy93] Johan Nuyts, Paul Suetens and Luc Mortelmans. Acceleration of maximum likelihood reconstruction, using frequency amplification and attenuation compensation. *IEEE Transactions on Medical Imaging*, 12(4), December 1993.

[Old93] Julia K. Older and Paul C. Johns. Matrix formulation of computerized tomograph reconstruction. *Physics in Medicine & Biology*, 38:1051–65, March 1993.

[Oll93] J. M. Ollinger. Reconstruction-reprojection of transmission scans and the variance of pet images. *IEEE conference records, Nuclear Science & Medical Imaging*, pages 1696–9, 1993.

[Oll94] John M. Ollinger. Maximum–likelihood reconstruction of transmission images in emission computed tomography via the em algorithm. *IEEE Transaction on Medical Imaging*, 13(1):89–101, March 1994.

[Pet81] T. M. Peters. Alogrithms for fast back– and re–projection in computed tomography. *IEEE Transactions on Neuclear Science*, NS–28(4):3641–3647, 1981.

[Pre92] William H. Press et al. *Numerical Recipes in C.* Cambridge University Press, 2. edition, 1992.

[Rad17] Johann Radon. Über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten. *Ber. Ver. Sächs. Akad. Wiss. Leipzig, Math-Phys. Kl.*, 69:262–77, April 1917. In german. English translation in S. R. Deans: The Radon Transform and Some of Its Applications.

[Rog87] J. G. Rogers, R. Harrop and P. E. Kinahan. The theory of three-dimensional image reconstruction for pet. *IEEE Transactions on Medical Imaging*, MI-6(3):239–43, September 1987.

[Ros82] A. Rosenfeld and A. C. Kak. *Digital Picture Processing*, volume 1. Academic Press, 2. edition, 1982.

[Sca93] John A. Scales. Iterative methods for large, sparse, inverse calculations. Lecture notes, Colorado School of Mines, 76 Olcott Drive, White River Junction, Vermont 05001, April 1993.

[Sch89] Robert J. Schalkoff. *Digital Image Processing and Computer Vision.* John Wiley & Sons, 1989.

[She78] L. A. Shepp and J. B. Krustal. Computerized tomography: The new medical x-ray technology. *American Mathematics Monthly*, 85:420–39, 1978.

[She82] L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Transactions on Medical Imaging*, MI-1(2):113–22, October 1982.

[Sun93] Winston Y. Sun, Stephen E. Derenzo and Thomas F. Budinger. An adaptive wiener filter for pet based on a novel technique for spectral estimation. *IEEE conference records, Nuclear Science & Medical Imaging*, 1993.

[Tow89] D. W. Townsend, T. Spinks, T. Jones et al. Three dimensional reconstruction of pet data from a multi–ring camera. *IEEE Transactions on Nuclear Science*, 36(1):1056–65, February 1989.

[Tow91] D. W. Townsend, T. Spinks, T. Jones et al. Fully three-dimensional reconstruction for a pet camera with retractable septa. *IEEE Transactions on Medical Imaging*, 10(4):505–13, December 1991.

[Tow94] D. Townsend, L. Byars, M. Defrise et al. Rotating positron tomographs revisited. *Physics in Medicine & Biology*, 39:401–10, 1994.

[Tru87] H. J. Trussell, H. Orun–Ozturk and M.R. Civanlar. Errors in reprojection methods in computerized tomography. *IEEE Transactions on Medical Imaging*, MI-6(3):220–7, September 1987.

[Var85] Y. Vardi, A. Shepp and L. Kaufmann. A statistical model for positron emission tomography. *Journal of the American Statistical Association*, 80(389):8–37, March 1985.

[Xu,93a] Xiao-Liang Xu, Jeih-San Liow and Stephen C. Strother. A non-iterative linear algebraic algorithm for image reconstruction. *IEEE Transactions on Medical Imaging*, pages 1198–1200, 1993.

[Xu,93b] Xiao-Liang Xu, Jeih-San Liow and Stephen C. Strother. A novel iterative reconstruction algorithm for emission computed tomography. *IEEE Transactions on Medical Imaging*, pages 503–6, 1993.

Appendix A

Program til iterativ rekonstruktion

I dette appendix, vil der blive givet en kort beskrivelse af brugen af `it` programmet, udviklet til iterativ rekonstruktion af PET billeder i forbindelse med anden del af dette projektet.

A.1 Ini filen

Programmet til iterativ rekonstruktion, `it` er lige som programmet `iradon` udviklet i del I, baseret på indlæsning af transformations-parametrene fra en initialiseringsfil. For at lette genereringen af denne, og dermed brugen af programmet, er der også blevet lavet en X-Windows 'skal', `It_Shell` til dette program. Programmet køres af TCL/TK kommandofortolkeren *wish* under Linux. På figur A.1 er vist skærmbilledet for dette program.

A.2 Parametre

Herunder gives en kort beskrivelse af de parametre der kan bruges i programmet. Disse indtastes i `It_Shell` som vist på figur A.1

`InFileName` Input fil, sinogram for det billede der skal rekonstrueres. Billedet kan vises med 'show' knappen.

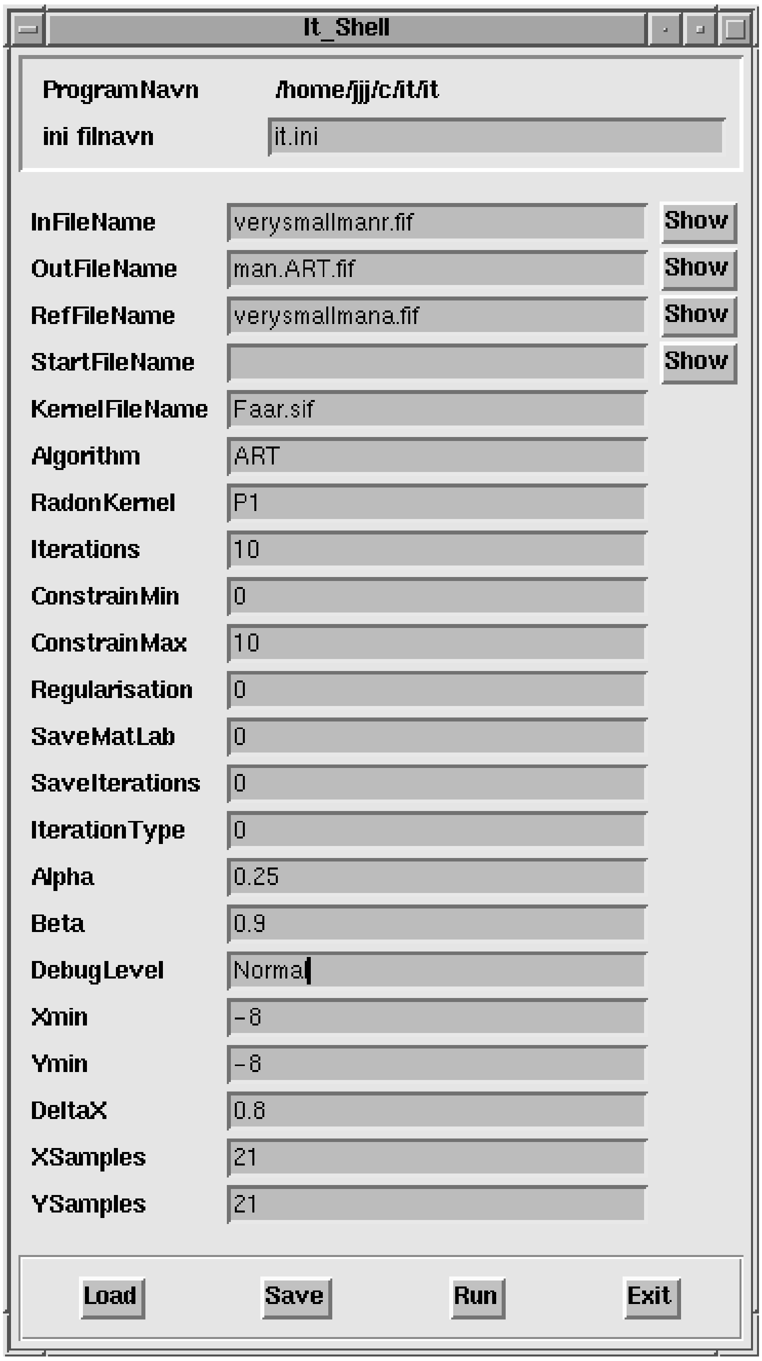
`OutFileName` Navn det rekonstruerede billede skal gemmes under. Efternavnet bestemmer fil-formatet. Billedet kan efter rekonstruktionen vises med 'show' knappen.

`RefFileName` Navn på referencebillede. Hvis dette billede er oplyst, bliver samplings-parametrene for outputbilledet taget fra dette billede. L_2 mål bliver udregnet med dette billede som referencebillede. Billedet kan vises med 'show' knappen.

`StartFileName` Billede der skal bruge som startgæt for iterationerne. Skal have samme samplingsparametre som referencebilledet. Hvis intet referencebillede er oplyst, tages samplingsparametrene til outputbilledet fra dette billede.

`KernelFileName` Navn transformations matricen skal gemmes med/hentes fra. Hvis en fil med det opgivne navn eksisterer, undersøges det om den indeholder en matrix der kan bruges til den aktuelle geometri, hvis ikke genereres en ny matrix, og den gemmes under dette navn.

`Algorithm` Metode hvormed rekonstruktionen skal foretages. Der kan vælges `ART`, `EM` eller `CG`.



Figur A.1: Skærmbillede for `It_Shell`.

`RadonKernel` Den metode der skal bruges til at generere transformations matrix med. Der kan vælges mellem `RNN`, `RL` eller `P1`, for nærmeste nabo approksimation, lineær interpolation eller stråle drevet approksimation.

`Iterations` Antallet af iterationer der skal foretages.

`ConstrainMin` Mindste tilladte værdi i de udregnede gæt på løsningen. Efter hver iteration bliver alle værdier i $\mathbf{x}$-vektoren der ligger under denne trunkeret.

`ConstrainMax` Højeste tilladte værdi i de udregnede gæt på løsningen. Efter hver iteration bliver alle værdier i $\mathbf{x}$-vektoren der ligger over denne trunkeret.

`Regularisation` Vælger om der skal bruges regularisering, sættes til 0 for ingen regularisering, 1 for regularisering med den implementerede Laplace kerne.

`SaveMatLab` Sættes til 1, hvis transformations matricen også skal gemmes i et MatLab læsbart format efter genereringen. 0 hvis der ikke skal gemmes.

`SaveIterations` Sættes forskellig fra 0, hvis der skal gemmes et gæt på løsningen efter hver iteration som et billede.

`IterationType` ART-specifik parameter. Bestemmer hvorledes linierne udvælges efter hver iteration. 0 for tilfældig udvælgelse, 1 for linievis udvælgelse.

`Alpha` ART-specifik parameter. Bestemmer hvorledes dæmpningsparameteren udregnes.

`Beta` ART-specifik parameter. Bestemmer hvorledes dæmpningsparameteren udregnes.

`DebugLevel` Bestemmer hvor meget information om kørslen der ønskes. Kan sættes til `Normal`, `Silent`, `Hardcore` eller `Debug`

`Xmin` Hvis intet reference- eller startbillede er oplyst, bliver samplings-parametrene for output billedet taget fra denne og de følgende parametre. Bestemmer mindste x-koordinat.

`Ymin` Bestemmer mindste y-koordinat.

`DeltaX` Samplingsafstanden i output billedet. Både Δx og Δy sættes lig `Delta x`.

`XSamples` Antallet af samples i x-retningen.

`YSamples` Antallet af samples i y-retningen.

Appendix B

'it' program filer

Herunder er en kort beskrivelse af de programfiler der er udviklet til del II af projektet, deres placering på den medfølgende diskette og udskriftens placering i dette appendix. Programudskrifterne er globalt nummereret i nederste højre hjørne, og nummereret i øverste højre hjørne inden for hver enkelt programfil. En LaTeX-formatteret udskrift af funktionsbeskrivelserne for de enkelte funktioner findes i appendix C.

Program	**Placering**	**Beskrivelse**	**Udskrift**
`Makefile`	`/it`	Makefil til `it` programmet	side 1
`it.h`	`/tools`	Funktions prototyper til `itmain.c`	side 2
`itini.h`	`/tools`	Generelle type definitioner	side 3
`itmain.c`	`/it`	Hovedprogram	side 4
`amatrix.c`	`/it`	Udregning af **A** matricen	side 6
`art.c`	`/it`	ART opdateringsfunktion	side 11
`em.c`	`/it`	EM opdateringsfunktion	side 13
`cg.c`	`/it`	CG opdateringsfunktion	side 15
`itreadargs.c`	`/it`	Initialiseringsrutiner	side 17
`sparse.h`	`/tools`	Header til `sparse.c`	side 20
`sparse.c`	`/tools`	Sparse matrix bibliotek	side 21
`it.ini`	`/it`	Eksempel på ini-fil til `iradon`	side 36
`itshell`	`/ishell`	Shell program til `it`	side 37

Ud over disse filer, bruges også de under del I af projektet udviklede rutiner til bla. billedbehandling. Disse er beskrevet i første del af rapporten.

Appendix C

Funktionsbeskrivelser for ‘it’

På de følgende sider er en LaTeX-formateret udskrift af de funktionsbeskrivelser der findes i programmerne til iterativ rekonstruktion.