Extension of DIRA (Dual-Energy Iterative Algorithm) to 3D Helical CT

Magnus Björnfot
Master of Science Thesis in Electrical Engineering

Extension of DIRA (Dual-Energy Iterative Algorithm) to 3D Helical CT

Magnus Björnfon

LiTH-ISY-EX--17/5057--SE

Supervisor: Alexandr Malusek
IMH, Linköpings universitet

 Examiner: Maria Magnusson
ISY, Linköpings universitet

Division of Computer Vision
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Copyright © 2017 Magnus Björnford
Abstract

There is a need for quantitative CT data in radiation therapy. Currently there are only few algorithms that address this issue, for instance the commercial Direct-Density™ algorithm. In scientific literature, an example of such an algorithm is DIRA. DIRA is an iterative model-based reconstruction method for dual-energy CT whose goal is to determine the material composition of the patient from accurate linear attenuation coefficients. It has been implemented in a two dimensional geometry, i.e., it could process axial scans only. There was a need to extend DIRA so that it could process projection data generated in helical scanning geometries. The newly developed algorithm (DIRA-3D) implemented (i) polyenergetic semi-parallel projection generation, (ii) mono-energetic parallel projection generation and (iii) the PI-method for image reconstruction. The computation experiments showed that the accuracies of the resulting LAC and mass fractions were comparable to the ones of the original DIRA. The results converged after 10 iterations.
## Contents

1 Introduction ......................................................... 1
   1.1 Motivation ..................................................... 1
   1.2 Aim ............................................................. 2
   1.3 Problem Formulation .......................................... 2
   1.4 Limitations .................................................... 3

2 Background .......................................................... 5
   2.1 Photon attenuation ............................................. 5
   2.2 Material Decomposition ....................................... 6
     2.2.1 Two-Material Decomposition ............................. 7
     2.2.2 Three-Material Decomposition ........................... 7
   2.3 CT-scanner ..................................................... 8
   2.4 Reconstruction algorithms ................................... 8
     2.4.1 Filtered backprojection ................................ 8
     2.4.2 Iterative reconstruction ................................. 9
     2.4.3 Available reconstruction algorithms ................... 10
   2.5 DIRA-2D ........................................................ 10
   2.6 Forward projection generation ............................... 12
   2.7 PI-method ..................................................... 13
   2.8 DIRA-3D ........................................................ 17

3 Methods ............................................................. 19
   3.1 Regular DIRA-3D reconstruction ............................. 19
     3.1.1 Phantom definition ................................... 19
     3.1.2 Projection generation ................................ 20
     3.1.3 Measured projections .................................. 22
     3.1.4 User-specific settings in DIRA-3D ....................... 22
     3.1.5 Reconstruction ......................................... 23
   3.2 Long object DIRA-3D reconstruction .......................... 23
   3.3 Error calculation ............................................. 24

4 Results .............................................................. 25
   4.1 Regular DIRA-3D reconstruction ............................. 25
1

Introduction

1.1 Motivation

Computed tomography (CT) uses the transmission of x-rays through the body in order to create 2-D or 3-D images. CT is a fast and flexible imaging method with many different applications. The CT-scanner consists of a detector and an x-ray source which rotate around the patient. The detector measures projections for a set of projection angles and a reconstruction algorithm reconstructs these data to images. See Ref. [3] for more information.

A single energy CT-scanner obtains the projections using a single photon spectrum. A DECT scanner uses two photon spectra obtained for two different x-ray tube voltages. This can be achieved in multiple ways. Two common methods are rapid voltage switching and dual-source CT. There is also a possibility to do two separate scans at two different tube voltages but this has the disadvantage of introducing temporal artefacts. DECT has the benefit over single energy CT by being able to obtain more information about the scanned object. See Ref. [5] for more information.

Radiation therapy uses ionizing radiation to kill malignant (cancer) cells. In external beam therapy the patient is irradiated by high energy particles produced by an accelerator or radioactive source. In brachytherapy a sealed radiation source is placed inside or next to the area requiring treatment. In this case low energy photons are typically used. The transport of high energy particles is less sensitive to material composition of the patient body than the transport of low energy photons used in brachytherapy where the photo-electric effect, which strongly depends on the atomic number, plays an important role. For this reason knowledge about the material composition of patient tissues is important for accurate
radiation treatment planning especially in brachytherapy.

DIRA [12] is an iterative model-based reconstruction method for DECT whose goal is to determine the material composition of the patient from accurate linear attenuation coefficients. DIRA uses filtered backprojection for reconstruction of images. The iterative loop calculates the mono-energetic and polyenergetic projections in a parallel beam geometry. It has been implemented in a two-dimensional geometry, i.e., it could process axial scans only. The results were promising and so there was a desire to extend DIRA so that it could reconstruct data obtained from helical scans.

As mentioned before, accurate estimates about the material composition of the patient are especially important for radiation treatment plans in cancer treatment where they can improve the accuracy of spatial dose distribution. Computed tomography (CT) is used for these purposes but it does not use its full potential. Especially when the new modalities like the dual-energy computed tomography (DECT) or spectral computed tomography are considered. Creating an algorithm which is able to achieve that goal for DECT in a helical scanning geometry would certainly benefit the field of oncology.

1.2 Aim

The aim of this thesis is to expand the DIRA algorithm to 3D. This means that the algorithm should be able to reconstruct the linear attenuation coefficients (LAC) from helical DECT data and perform material decomposition using these LAC while removing the beam hardening and cone-beam artefacts.

1.3 Problem Formulation

The main goal is to incorporate 3D reconstruction into the DIRA algorithm. This will require many changes to the existing DIRA algorithm, for instance:

• The two-dimensional filtered backprojection should be changed to a three-dimensional reconstruction algorithm. Examples of 3D reconstructed algorithms are the PI-method [23], weighted filtered backprojection [19] (WFBP) and the Katsevich algorithm [8]. The source code of the PI-method was available to the author.

• The generation of parallel beam forward projections should be complemented with a generation of cone-beam projections.

• The iterative loop should be modified so that the two aspects mentioned above work together.

This has to be implemented in an algorithm and the algorithm has to be tested. It raises the following questions:
• Can the new algorithm be implemented in a way similar to the one used in DIRA?
• Can the new algorithm using the PI-method achieve similar accuracy as DIRA?
• Is the new algorithm able to reduce cone-beam and beam hardening artefacts?
• How well does the new algorithm perform with truncated data (long object problem)?

The text of this report should follow the recommendations in ‘anvisningar för exjobbsrapport på ISY document’ which describes the structure and recommended content of individual sections.

1.4 Limitations

• Direct semi-parallel projections were used instead of cone-beam projections rebinned to semi-parallel projections due to time limitations.
• The semi-parallel projections used as input to the reconstruction algorithm were calculated for a voxelized phantom with a relatively coarse resolution to shorten the computational time.
• The source code of WFBP [19] and Katsevich method [8] were not available to the author. This lead to a strong preference for the PI-method [23] whose code was readily available.
2.1 Photon attenuation

Photons used in diagnostic CT (energy 5-190 keV), interact with matter by photoelectric absorption, coherent scattering and incoherent scattering. Photoelectric absorption is a process in which a photon transfers its entire energy to an electron inside an electron shell of an atom. This causes the electron to leave the atom with an energy equal to

\[ T = h \nu - U, \]  

where \( U \) is the binding energy of the electron and \( h \nu \) is the photon energy, \( \nu \) denotes the frequency and \( h \) denotes Planck’s constant. Photoelectric absorption can only occur if the photon energy is greater than \( U \). The photon vanishes, always delivering its full energy. The atomic cross-section, \( \sigma \), for photoelectric absorption strongly depends on the atomic number, \( Z \), of the material [1, p. 140]

\[ \sigma \propto \frac{Z^4}{(h\nu)^3}. \]  

Incoherent scattering is a process in which a photon collides with a bound electron and transmits a part of its energy to the electron which is then emitted from the atom. This interaction changes the photon direction of flight. The atomic cross-section for incoherent scattering is linearly proportional to the atomic number \( Z \) (Attix, 1986, p.132). For water the incoherent scattering is the dominant interaction for energies from 20 keV to 30 MeV [1, p. 125]. Incoherent scattering has a higher probability to occur on looser bound electrons.

Coherent scattering is a process in which photons are scattered by bound atomic
Background

Electrons. The photon direction of flight is changed and the change in photon energy is negligible. [16, p. 214]

The overall effect of the above mentioned processes can be described by the macroscopic cross-section, $\mu$, which is also called the linear attenuation coefficient (LAC). The number of primary photons, $N$, in a mono-energetic beam can be described as

$$N = N_0 e^{-\int L \mu dl},$$

(2.3)

where $N_0$ is the amount of photons emitted from the source, $\mu$ is the LAC and $\int L \mu dl$ is the line integral through the patient's body. The measured signal in current CT-detectors is typically proportional to the energy imparted to the detector, i.e., the intensity of the beam is $I = NE$ where $N$ is the number of photons in the beam and $E$ is the energy of the photons. Equation (2.3) can be written as

$$I = I_0 e^{-\int L \mu dl}.$$  

(2.4)

For a mono-energetic beam the line integral can be determined from the measured intensities as

$$\int L \mu dl = -\ln \frac{I}{I_0}.$$  

(2.5)

2.2 Material Decomposition

One of the major benefits of using a DECT and a model-based reconstruction is that the material composition and density of the patient can be estimated. There are however approximations that have to be made and the material decomposition is not guaranteed to work for all materials. Ref. [12] suggests two different decompositions, one assumes a mixture consists of two materials plus density and the other a mixture of three materials. The first assumption is that the sum of the cross sections of the individual atoms is equal to the cross section of the molecule. The mass attenuation coefficient (MAC) relates to the linear attenuation coefficient as

$$\mu_m = \frac{\mu}{\rho},$$

(2.6)

where $\mu_m$ is the mass attenuation coefficient, $\mu$ is the linear attenuation coefficient and $\rho$ is the density. The first assumption leads to

$$\mu_m(E_i) = \sum_k w_k \mu_{m,k}(E_i),$$

(2.7)

where $\mu_m(E_i)$ is the mass attenuation coefficient at energy $E_i$ for the mixture, $w_k$ is the mass fraction for material $k$ and $\mu_{m,k}(E_i)$ is the mass attenuation coefficient for material $k$. The second assumption is based on the conservation of mass,

$$\sum_k w_k = 1.$$  

(2.8)
2.2 Material Decomposition

It is important to note that $w_k$ ideally is the mass-fraction which fulfills $0 \leq w_k \leq 1$ and (2.8). However in practical implementations this is not always true which means that $w_k$ may be negative. The first step of any material decomposition is to decide which areas should be decomposed to what. This can be done by simple thresholding on the reconstructed volumes representing LACs. [11]

### 2.2.1 Two-Material Decomposition

Two-material decomposition mathematically decomposes a material into a doublet of two materials. It also determines the density of the mixture. A common example is a decomposition of bone into bone marrow and compact bone. The mathematical solution to the two-material decomposition is as follows. A linear equation system can be formed using (2.8) and (2.7):

$$Ax = b,$$

where $A$ is a $3 \times 3$ matrix, $x$ is a column vector containing $\{w_1, w_2, \frac{1}{\rho}\}$ and $b$ is a column vector containing $\{0, 0, 1\}$. The content of $A$ is

$$A = \begin{pmatrix} \mu_{m,1}(E_1) & \mu_{m,2}(E_1) & -\mu(E_1) \\ \mu_{m,1}(E_2) & \mu_{m,2}(E_2) & -\mu(E_2) \\ 1 & 1 & 0 \end{pmatrix}.$$  

This can be solved as $x = A^{-1}b$ since $\mu(E_i)$ is known (from a measurement) and $\mu_{m,k}(E_i)$ is a tabulated value for the specified doublet. [11]

### 2.2.2 Three-Material Decomposition

In three-material decomposition the goal is to decompose a material into a mixture of three materials (a triplet). The density is not calculated directly in the equation system. Therefore a new assumption has to be made that the partial molar volumes of the triplet components have to be the same in the mixture and the standalone components. This assumption provides the density of the mixture as

$$\frac{1}{\rho} = \sum_{k=1}^{3} \frac{w_k}{\rho_k},$$  

(2.10)

where $k$ denotes the materials in the mixture. A linear equation system can be formed just like (2.9) where $b$ is the exact same vector, $x = \{w_1, w_2, w_3\}$ and $A$ is

$$A = \begin{pmatrix} \frac{\mu(E_1)}{\rho_1} - \mu_{m,1}(E_1) & \frac{\mu(E_1)}{\rho_2} - \mu_{m,2}(E_1) & \frac{\mu(E_1)}{\rho_3} - \mu_{3,m}(E_1) \\ \frac{\mu(E_2)}{\rho_1} - \mu_{m,1}(E_2) & \frac{\mu(E_2)}{\rho_2} - \mu_{m,2}(E_2) & \frac{\mu(E_2)}{\rho_3} - \mu_{3,m}(E_2) \\ 1 & 1 & 1 \end{pmatrix}.$$  

This system can also be solved as $x = A^{-1}b$. The density can be calculated using (2.10). Equation (2.10) is not valid for densities close to 0 when the densities of the base materials are similar to densities of soft tissues (1 g/cm$^3$) or larger. This is why low density materials such as air should be modeled using two-material decomposition. [11]
2.3 CT-scanner

A CT-scanner consists of an x-ray tube and a detector array. Both the x-ray tube and the detector array rotate around the patient inside a gantry. CT-scanners can operate in two different modes. In an axial scan the table is stationary during the rotation of the gantry. In a helical scan the table moves continuously through the rotating gantry. The detector array consists of a number of detector elements. The signal measured by the detector array at one projection angle is typically called a projection. A set of projections from all projection angles $\theta$ is called a sinogram.

2.4 Reconstruction algorithms

2.4.1 Filtered backprojection

Filtered backprojection (FBP) mathematically reconstructs the image $f(x, y)$ from the projection data $p_\theta(r)$, see for instance Ref. [7]. This is done by backprojecting every projection-ray along the ray it was measured. There is however a slight correction that has to be made in order to get good results. If a simple backprojection algorithm is used, the image will become blurred by the function $\frac{1}{|r|}$. The backprojection is described mathematically for a 2D image by

$$ f_b(x, y) = \pi \int_0^\infty \int_{-\infty}^{\infty} p_\theta(r) \delta(x \cos \theta + y \sin \theta - r) dr d\theta, \quad (2.11) $$

where $\delta$ is the dirac function and $f_b(x, y)$ is the blurred image. The relation between the blurred image $f_b(x, y)$ and the correct image $f(x, y)$ is

$$ f_b(x, y) = f(x, y) * \frac{1}{|r|}, \quad (2.12) $$

where $*$ denotes convolution. The blurring can be compensated for by filtering the projections according to

$$ \begin{cases} \mathcal{F}\{p_\theta(r)\} = P_\theta(k) \\ q_\theta(k) = \mathcal{F}^{-1} [P_\theta(k) |k|] \end{cases}, \quad (2.13) $$

where $\mathcal{F}$ denotes the fourier transform and $|k|$ is a ramp-filter. For the full mathematical derivation see Ref. [7]. Using the filtered projections the backprojection gives

$$ f(x, y) = \pi \int_0^\infty \int_{-\infty}^{\infty} q_\theta(r) \delta(x \cos \theta + y \sin \theta - r) dr d\theta. \quad (2.14) $$

Summarizing (2.13) and (2.14) into a step by step algorithm gives:

- Obtain projections $p_\theta(r)$ from the CT-scanner.
2.4 Reconstruction algorithms

- Ramp-filter the projections with \(|k|\) which gives \(q_\theta(r)\).
- Backproject the ramp-filtered projections.

Filtered backprojection is shown in figure 2.1.

![Filtered backprojection](image)

**Figure 2.1:** A schematic drawing showing how filtered backprojection works. Source: Maria Magnusson

2.4.2 Iterative reconstruction

Iterative reconstruction is a reconstruction using an iterative algorithm. An initial guess of the reconstructed image is performed. Then this guess is refined based on the supplied raw data. Iterative reconstruction algorithms require more computational power than non-iterative reconstruction algorithms and it is necessary to find a suitable stopping criteria. This is to ensure that the computational cost is reasonable. There exists different variants, such as algebraic, statistical and model-based methods. All iterative reconstruction methods, however, follow this basic algorithm assuming the input is the measured projections:

1. Make an initial guess of the object function.
2. Compute forward projections of this guess.
3. Apply a correction term which can be calculated as e.g. the difference between the measured projections and the computed forward projections.
4. If the correction term is small enough, stop.
5. Reconstrcut the corrected data.
6. Repeat from step 2.

As previously mentioned there are different variants of iterative reconstruction. Algebraic methods just try to model the geometry of the projections and image. Statistical models the statistics of the photons. The aim is to supress noise in the
reconstructed image. Statistical methods have been utilized in PET and SPECT for a long time.

The goal of model-based methods is to completely model the physics of the CT-scanner such as the polyenergetic energy spectra of the photons, scattering of photons in the object etc. There is a need to find a balance between how many properties that are modeled and the computational complexity. Therefore only a few properties are modeled. There are multiple subcategories for model-based methods such as geometric modeling, physical modeling and prior object information modeling. Physical modeling includes the modeling of the polyenergetic energy spectra of the photons, scattering and artefacts that occur due to the polyenergetic energy spectra such as beam hardening. This can be used to reduce beam hardening effects giving a better image quality. [2]

### 2.4.3 Available reconstruction algorithms

Model-based iterative reconstruction methods (MBIR) are available in commercial CT-scanners, for example ADMIRE [17] (Siemens healthcare), VEO [4] (GE healthcare), FIRST [22] (Toshiba medical systems) and IMR [15] (Philips healthcare). Unfortunately there is not much information freely available about these implementations. All the manufacturers claim that their algorithms improve image quality while reducing patient dose compared to the older non-iterative methods. None of the manufacturers, however, claims an improvement of the accuracy in reconstructed CT numbers which is needed in radiation therapy. This issue has been recently addressed by the DirectDensity [18] algorithm from Siemens that provides electron density images independent of used x-ray tube voltage setting. This algorithm still uses the single energy CT approach.

Examples of DECT algorithms published in scientific literature are ‘Exact dual energy material decomposition from inconsistent rays’ (MDIR) [9] and ‘Reconstruction algorithm for polyenergetic CT imaging’ (RAPTCTI) [24]. MDIR used the Feldkamp-type filtered backprojection with a very small cone-beam angle reducing the cone-beam artefacts in an iterative loop performing two-material decomposition (e.g., (photoelectric effect, Compton scattering) or (water, bone) bases) but other reconstruction algorithms could be used as well. RAPTCTI is capable of eliminating beam hardening artefacts by incorporating the polyenergetic characteristics of the x-ray beam into the reconstruction process [24].

### 2.5 DIRA-2D

DIRA is a model-based iterative reconstruction algorithm that uses filtered back-projection, automatic segmentation and multimaterial tissue decomposition [12]. The implementation described in Ref. [12] reconstructs 2D axial scans. It uses Joseph’s method [6] for calculations of forward projections, see section 2.6. The algorithm performs the following steps (see also figure 2.2):

1. Measured polyenergetic projections for two tube voltages are obtained.
2. At iteration 0, water beam-hardening correction (WBHC) is performed, otherwise not. (Not indicated in the figure.)

3. Reconstruction with filtered backprojection (FBP) is performed and reconstructed images $\mu_1$ and $\mu_2$ are obtained.

4. The images $\mu_1$ and $\mu_2$ are automatically segmented into pre-defined tissues and organs. In the presented example, a threshold segmentation to soft and bone tissue combined with filling of bone cavities is used.

5. Classification into mass fractions of base materials is performed. For soft tissue, three-material decomposition is used and for bone tissue two-material decomposition is applied.

6. Parallel forward polyenergetic projections are calculated.

7. Parallel monoenergetic projections are calculated.

8. The measured projections are corrected using the correction term: $P_{M,U_i} + (P_{E_i} - P_{U_i})$, where $P_{M,U_i}$ are the measured polyenergetic projections, $P_{E_i}$ are the calculated mono-energetic projections and $P_{U_i}$ are the calculated polyenergetic projections.

9. Continue from step 3 unless the required number of iterations have been reached.

Figure 2.2: A flowchart of the DIRA-2D algorithm where the water beam hardening correction is omitted. Source: Maria Magnusson

The main idea of the algorithm is to convert polyenergetic projections into monoenergetic ones. The monoenergetic projections can be accurately reconstructed using the filtered backprojection. To avoid misinterpretation this algorithm is referred to as DIRA-2D in this report.
2.6 Forward projection generation

The projections are calculated using

$$P = \ln \frac{I_0}{I}. \quad (2.15)$$

In (2.15) $I_0$ is calculated for an ideal energy integrating detector as

$$I_0 = \int_0^{E_{\text{max}}} EN(E)dE, \quad (2.16)$$

where $E$ is the photon energy and $N(E)$ is the energy spectrum of photons emitted from the x-ray tube. The quantity $I$ is calculated as

$$I = \int_0^{E_{\text{max}}} EN(E) \exp \left[ - \int_L \mu(x, y, z, E) dl \right] dE, \quad (2.17)$$

where $\mu(x, y, z, E)$ is the LAC of pixel $(x, y, z)$ at energy $E$, $\int_L dl$ is a line integral through the object. This calculation is time consuming, since the line integrals must be calculated for all energies in the energy spectrum.

The calculation of projections change slightly when material decomposition is introduced. The line integrals are calculated through volume fractions of the different base materials. Equation (2.15) and (2.16) are used with

$$I = \int_0^{E_{\text{max}}} EN(E) \exp \left[ - \sum_k \mu_k(E) l_k \right] dE, \quad (2.18)$$

where $\mu_k$ is the LAC of the $k$th base material and $l_k$ is computed as

$$l_k = \rho_k^{-1} \int_L \rho(x, y, z) w_k(x, y, z) dl, \quad (2.19)$$

where $\rho_k$ is the tabulated density of the $k$th base material, $\rho(x, y, z)$ is the calculated density in voxel $(x, y, z)$ and $w_k(x, y, z)$ is the mass fraction of the $k$th material in voxel $(x, y, z)$. For two-material decomposition $\rho(x, y, z)$ is received directly by solving (2.9). In case of three-material decomposition (2.10) must also be used. [12]

A monoenergetic projection $P_{E_i}$, where $E_i$ is a specific energy, is calculated as

$$P_{E_i} = \sum_k \mu_k(E_i) l_k, \quad (2.20)$$

where $i = 1, 2$ in this case. The line integral $\int_L dl$ can be calculated using for instance Josephs method [6] described below for a 2D geometry. The line integral is approximated by a sum.
Let $f$ be a discrete 2D function. The data points are illustrated with dots in figure 2.3. The task is to calculate the line integrals along the two lines. For angles $-45^\circ \leq \phi \leq 45^\circ$ the sum is calculated as $\Delta x \sum_{n=1}^{N} f(n)$ where $\Delta x$ is the distance between two sample points in an equidistant grid alongside the ray and $f(n)$ are interpolated values taken in this grid, see left image in figure 2.3. For angles $45^\circ \leq \phi \leq 135^\circ$ a similar approach is used with the difference that the $n$ coordinate is changed to the $m$ coordinate. For the geometry in figure 2.3 the formulas are:

$$
\begin{align*}
\text{l} &= \begin{cases} \\
\Delta x \sum_{n=1}^{N} f(n)/|\cos\phi|, & -45^\circ \leq \phi \leq 45^\circ, \\
\Delta x \sum_{m=1}^{M} f(m)/|\sin\phi|, & 45^\circ \leq \phi \leq 135^\circ,
\end{cases}
\end{align*}
$$

(2.21)

where $l$ is the resulting line integral and $f(n)$ or $f(m)$ are the interpolated values in the $n$ or $m$ direction. This method can be extended in three dimensions. Equation (2.21) is then extended to three cases instead of two and bilinear interpolation is used instead of linear interpolation. [23]

### 2.7 PI-method

A third generation CT-scanner uses cone-beam projections, see figure 2.4. In this figure, $\kappa$ is the cone-angle, $\gamma$ is the fan-angle and $P$ represents the pitch of the helix (the height of one complete helix turn, measured parallel to the axis of the helix). The maximum value of the cone-angle is called the cone-beam angle and is denoted $\kappa_{\text{max}}$ and the maximum value of the fan-angle is called the fan-beam angle and is denoted $\gamma_{\text{max}}$. The $s$-axis starts at the point $C$, the cross-section between the central ray and the $z$-axis, and is aligned with the $z$-axis. The x-ray source is visualized as a dot which rotates around the $z$-axis in a helical trajectory. The detector is cylindrical and has its centre in the x-ray source.
In the PI-method cone-beam projections are converted to semi-parallel projections, see figure 2.5. This procedure involves interpolation and is called rebinning. The mathematical relation between a cone-beam projection, \( p(\beta, \gamma, s) \), and the corresponding semi-parallel projection, \( p^P(\theta, t, s) \), is

\[
p^P(\theta, t, s) = p(\beta, \gamma, s),
\]

where the relations between the parameters \((\theta, t, s)\) and \((\beta, \gamma, s)\) are

\[
\begin{align*}
\theta &= \beta + \gamma \\
t &= R \sin \gamma
\end{align*}
\]

In (2.23) \( R \) is the radius of the helix. Projections to the \((x, y)\) plane from the rays generating the semi-parallel projections are parallel line segments, see figure 2.5. In the \((v, z)\) plane the rays generating the semi-parallel projections are fan-beam shaped.
The PI-method assumes that the detector is positioned between the two turns of the helix (Tam window), see figure 2.5. Then in the \((t, s)\) plane the rays define a rectangular area called the PI-detector. The \(s\)-axis is now shown in the figure. It has been shown that the PI-detector gives complete and non-redundant projection data, i.e., no redundant data are present during reconstruction [23]. The PI-detector restricts the illumination interval of a voxel \((x, y, z)\) to exactly 180° [23]. A basic outline for the simplest version of the PI-method is:

1. Obtain cone-beam projections from the CT-scanner.
2. Form semi-parallel projections from the cone-beam projections using (2.22) and (2.23).
3. Throw away all projections outside the Tam window.
4. Pre-weight with \(\cos \kappa\).
5. Perform rampfiltering along the rows of the virtual planar detector.
6. Perform three-dimensional backprojection along the semi-parallel rays.

The PI-method is not an exact method and small artefacts can be seen in the reconstructed images. In general, the larger the cone-beam angle is the larger artefacts are observed. [23]

Ref. [10] presents an iterative algorithm using the PI-method that suppresses cone-beam artefacts and errors arising from the long object problem. There is also a version of the iterative algorithm (see figure 2.6) that does not handle the long object problem; this algorithm consists of the following steps:
1. Initialize the reconstructed volume $V_0$ containing the LAC to zero.

2. Obtain semi-parallel projections.

3. Set $i = 1$.

4. Pre-weight, ramp-filter and backproject the projections using the PI-method. Receive $R_i$ where $R_1$ is the full reconstruction volume and $R_i$, $i > 1$ is the reconstruction from the difference of the projections.

5. Compute $V_i = V_{i-1} + \alpha_i R_i$ where $\alpha_i$ is a scale factor.

6. Produce new semi-parallel projections through $V_i$.

7. Form the correction term $P_0 - P_i$, where $P_0$ are the original semi-parallel projections and $P_i$ are the generated projections for iteration $i$.

8. Increment $i$.

Steps 4-8 are repeated a predefined number of times. The parameter $\alpha_1$ should be set to 1 and $\alpha_i$, $i > 1$, can be set to a suitable value $\leq 1$.

![Figure 2.6: A flowchart of an iterative PI-method implementation without handling the long object problem. Source: Maria Magnusson](image)

A problem known as the long object problem appears when an object scanned in a CT-scanner is longer than the helical scanning path. For instance when only the region of the human body covering the liver is scanned. In this case some voxels at the ends of the long object are not illuminated from a sufficient number of projection angles, i.e., there are missing projection data. As a result some of the reconstructed LAC will be affected by strong artefacts. [10]

The version that handles the long object problem only introduces two new steps: extrapolation of the initial projections and masking of the correction term. The extrapolation is made because the rampfilter is very sensitive to sharp edges in the projection data. The mask then removes the extrapolated data.
2.8 DIRA-3D

DIRA-3D is an extension of DIRA-2D that includes the reconstruction of helical CT projections. The main differences between these two algorithms are:

- DIRA-3D uses the PI-method for reconstruction while DIRA-2D uses FBP.
- DIRA-3D uses semi-parallel projections and parallel projections while DIRA-2D only uses parallel projections.

The algorithm performs the following steps (see also figure 2.8):

1. Obtain measured semi-parallel polyenergetic projections, $P_{M,U_i}$, for two different tube voltages, $U_i$, $i = 1, 2$, giving $P_{M,U_1}$ and $P_{M,U_2}$.

2. Reconstruct these projections so that the reconstructed LAC at energy $E_i$ is $\mu_i = \Delta \mu_i + \mu_{P_i}$, where $\mu_{P_i}$ is the reconstructed LAC from mono-energetic parallel projections at energy $E_i$ and $\Delta \mu_i$ is the result of reconstructing $P_{M,U_i} - P_{U_i}$, where $P_{U_i}$ are the computed semi-parallel projections. For the first iteration, $P_{U_i} = P_{E_i} = 0$ and thus $\mu_i$ is the reconstruction of the $P_{M,U_i}$ only.

3. Perform automatic threshold segmentation on $\mu_1$ and $\mu_2$.

4. Classify tissues using the material decomposition methods (section 2.2).

5. Calculate polyenergetic projections $P_{U_i}$ for semi-parallel geometry, see (2.15), (2.16) and (2.18) in section 2.6.

6. Calculate monoenergetic projections $P_{E_i}$ for parallel geometry, see (2.20) in section 2.6.

Points 2-6 are repeated a predefined number of times.

![Figure 2.7: A flowchart showing a implementation of the PI-method used to handle the long object problem. Source: Maria Magnusson](image)
Figure 2.8: A flowchart of the DIRA-3D algorithm. Source: Maria Magnusson
Two CT experiments were performed. The first one used regular DIRA-3D reconstruction and is described in section 3.1. The second one used missing projection data due to the long object problem and the version that could handle this problem; it is described in section 3.2.

### 3.1 Regular DIRA-3D reconstruction

#### 3.1.1 Phantom definition

The phantom consisted of six ellipsoids. The ellipsoids were located in two layers where one layer consisted of four ellipsoids and the other one consisted of two ellipsoids, see figure 3.1. The resolution of this phantom was $128 \times 128 \times 48$ voxels.
Figure 3.1: A colormap of a section of the original phantom, where the values of one define voxels belonging to the phantom. Two slices are shown: $z = 10$ (a) and $z = 24$ (b), where $z$ is the voxel index in the voxel array.

The material of each ellipsoid is defined in table 3.1.

Table 3.1: Elemental composition in atomic fractions and corresponding density for materials in the six ellipsoids forming the phantom.

<table>
<thead>
<tr>
<th>ellipsoid</th>
<th>Material</th>
<th>Atomic composition</th>
<th>Density (g/cm$^3$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>Adipose t.</td>
<td>H 0.626017, C 0.270965, N 0.003161, O 0.099287, Na 0.000241, S 0.000173, Cl 0.000156</td>
<td>0.95</td>
</tr>
<tr>
<td>$S_2$</td>
<td>Muscle</td>
<td>H 0.631619, C 0.073790, N 0.015151, O 0.277367, Na 0.000271, P 0.000403, S 0.000584, Cl 0.000176, K 0.000639</td>
<td>1.05</td>
</tr>
<tr>
<td>$S_3$</td>
<td>Lipid</td>
<td>H 0.621918, C 0.341890, O 0.036192</td>
<td>0.92</td>
</tr>
<tr>
<td>$S_4$</td>
<td>Protein</td>
<td>H 0.480981, C 0.326573, N 0.089152, O 0.101004, S 0.002291</td>
<td>1.35</td>
</tr>
<tr>
<td>$S_5$</td>
<td>Water</td>
<td>H 0.666667, O 0.333333</td>
<td>1.00</td>
</tr>
<tr>
<td>$S_6$</td>
<td>Compact bone</td>
<td>H 0.403076, C 0.149395, N 0.033840, O 0.316004, Na 0.001473, Mg 0.000929, P 0.034249, S 0.001056, Ca 0.059978</td>
<td>1.92</td>
</tr>
</tbody>
</table>

3.1.2 Projection generation

The scanning geometry for the semi-parallel projection generation is shown in figure 3.2. The parallel-projection generation was done slice by slice.
Figure 3.2: The scanning geometry of the semi-parallel projection generation through the phantom showing projection number 0, 200, 400, 600 and 800 and highlighting $z = 14$. Source: Ref. [10].

Semi-parallel scanning parameters

The size of the PI-detector was extended by several pixels in the margins to allow interpolation of pixel values. A subset of parameters for semi-parallel projection generation used in the CT experiments (figure 2.5):

- Number of projection angles (regular) = 800 (0-799).
- Number of helix turns = 2.
- Pitch of helix = 32 voxels.
- Cone beam angle = $\pm 3.138074^\circ$.
- Voxel size: $\Delta x = 2.76$ mm.
- Detector size: $N_t \times N_s = 192 \times 32$ pixels.
- Extended detector size: $N_{t2} \times N_{s2} = 194 \times 36$ pixels.
- Detector pixel size: $\Delta t \times \Delta s = \Delta x/1.5 \times \Delta x/2$.
- Forward projection interpolation filter: linear interpolation in the $t$-direction, sincot interpolation (see below) in the $s$-direction.

The sincot filter is more sinc-like than the linear interpolation filter.

Parallel scanning parameters

A subset of parameters for parallel projection generation used in the CT experiments:

- Number of detector elements = 128.
- Number of projection angles = 400.
- Angular interval = $[0^\circ, 180^\circ]$.
- Forward projection interpolation filter: linear interpolation.

See Appendix A for all parameters used by the system.
3.1.3 Measured projections

The measured projections were simulated using line integrals in (2.18) with \( l_k = \int m_k(x, y, z) dl \) where \( m_k \) are the masks for the different ellipsoids in the phantom described in section 3.1.1. The line integrals were produced in a semi-parallel geometry and using the energy spectra in figure 3.3. The algorithm for the calculation of measured projections was implemented in MATLAB and executed as a part of the main DIRA-3D algorithm.

![Energy spectra](image.png)

*Figure 3.3: The photon energy spectra used in the DIRA-3D algorithm for the x-ray tube voltages of 80 and 140 kV.*

3.1.4 User-specific settings in DIRA-3D

**Tissue Classification**

The phantom was segmented into air, soft tissue and bone regions. The corresponding LAC thresholds were 12 m\(^{-1}\) and 33 m\(^{-1}\) for the low energy image. The value of 12 m\(^{-1}\) was chosen since it is slightly below the LAC of lipid (20 m\(^{-1}\)) at \( E = 50 \) keV. The value of 33 m\(^{-1}\) was chosen since it is slightly above the LAC for protein (28 m\(^{-1}\)) at \( E = 50 \) keV. Air was decomposed into a doublet of (lipid, water). Soft tissue was decomposed into a triplet of (lipid, protein, water). Bone was decomposed into a doublet of (compact bone, bone marrow). As a consequence the ellipsoid \( S_1 \) containing bone was decomposed into the (compact bone, bone marrow) doublet and ellipsoids \( S_2 \ldots S_6 \) were decomposed into the (lipid, protein, water) triplet.
3.1.5 Reconstruction

The reconstruction via DIRA-3D used the geometry and parameters described in section 3.1.2. The interpolation filters used for backprojection are linear interpolation in all directions. A Hanning filter is used together with the rampfilter in the parallel geometry.

3.2 Long object DIRA-3D reconstruction

This CT experiment used the same phantom, scanner geometry and scanning parameters as the experiment described in the previous section. The only difference was that the projection data were truncated as in the long object problem described in section 2.7. The truncation of the projection data was simulated by setting certain regions in the projection data to zero, see figure 3.4.

The regions with the missing data are shown in figure 3.5. Note that the size of the missing data region is larger close to the end of the object.

Figure 3.4: The mask defining the missing data as seen from the (rectangular) PI-detector. Left image shows projection angle 375, center image shows projection angle 400 and right image shows projection angle 425. White color indicates missing data.

Figure 3.5: The regions with missing data at $z = 12$ (a), $z = 14$ (b) and $z = 16$ (c).

Ref. [10] proposed to treat the long object problem by generating projections even for regions with missing data. This approach was implemented in DIRA-3D in steps 1 and 5 (section 2.8); $P_{MU_i}$ and $P_{U_i}$ were truncated using the mask described in figure 3.4.
3.3 Error calculation

The relative error of reconstructed LAC was estimated as

$$\delta(\bar{\mu}) = (\bar{\mu} - \mu_t)/\mu_t,$$

(3.1)

where $\mu_t$ is the tabulated value and $\bar{\mu}$ is the average of the calculated LAC in a spherical region of interest (ROI). The ROI was defined as a sphere with a radius of one third of the evaluated ellipsoid radius (in the $x, y$ plane) and positioned in the center of the evaluated ellipsoid. The relative errors of the mass density were defined similarly.
4.1 Regular DIRA-3D reconstruction

4.1.1 Reconstruction for different iterations

Figure 4.1 shows the disappearance of the cone-beam artefact with the increasing number of iterations. A visual comparison of images (d) and (e) in figure 4.1 shows that there is not much difference between results obtained at iterations 4 and 25. Figure 4.5 indicates that the algorithm converges already after 10 iterations. LACs reconstructed at both energies are shown in figure 4.2 for the 25th iteration; such a high iteration number is used to make sure that the algorithm has converged. Figure 4.3 shows the reconstruction for slice $z = 24$ and iterations 1, 5 and 25. The streaks in reconstructed images are most likely caused by partial volume artefacts.
Figure 4.1: Colormaps of LAC (in m$^{-1}$) for iterations 1 (a), 2 (b), 3 (c), 4 (d) and 25 (e) of regular DIRA-3D reconstruction for the slice $z = 14$ and photon energy $E_1 = 50$ keV. The axis labels show pixel indexes.
Figure 4.2: Colormaps of LAC (in m$^{-1}$) for energies $E_1 = 50$ keV (a) and $E_2 = 88.5$ keV (b) of regular DIRA-3D reconstruction for the slice $z = 10$ and iteration number 25. The axis labels show pixel indexes.

Figure 4.3: The same as in figure 4.1 for slice $z = 24$ and iteration 1 (a), 5 (b) and 25 (c).
Figure 4.4 shows how the reconstructed LAC varied with position inside and at the borders of the bone ellipsoid. This dependence on position was different in the $x$- and $y$-directions compared to the $z$-direction most likely because the parallel projection generation and reconstruction was done slice by slice. Also the Hanning filter used together with the rampfilter performed smoothing in the $x$- and $y$-directions. There is a small overshoot in the $z$-direction which is not visible in the $x$- and $y$-directions where the LAC profile appears smooth. It is important to note that changes to the algorithm may affect the severity of these effects.

Figure 4.4: Reconstructed LAC as a function of the distance, $d$ (in voxels), from the estimated center of the bone ellipsoid for low energy, $E_1 = 50$ keV. The profiles are taken in the $(x, y)$ plane along the $x$ axis (a), $(x, y)$ plane along the $y$ axis (b) and in the $(x, z)$ plane along the $z$ axis (c).

Figure 4.5 shows that the relative error converged to a value close to zero already after 10 iterations. The functions are plotted using a solid line for visual guidance only. Table 4.1 shows the values for the 25th iteration.

Figure 4.5: The relative errors of the LAC as a function of the number of iterations for $E_1 = 50$ keV (a) and $E_2 = 88.5$ keV (b).
Table 4.1: The relative errors $\delta(\bar{\mu}_1)$ and $\delta(\bar{\mu}_2)$ of the reconstructed LAC for energies $E_1 = 50$ keV and $E_2 = 88.5$ keV, respectively.

<table>
<thead>
<tr>
<th>Material</th>
<th>$\delta(\bar{\mu}_1)$ (%o)</th>
<th>$\delta(\bar{\mu}_2)$ (%o)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adipose T.</td>
<td>0.256</td>
<td>0.224</td>
</tr>
<tr>
<td>Compact Bone</td>
<td>1.240</td>
<td>0.280</td>
</tr>
<tr>
<td>Muscle</td>
<td>-0.433</td>
<td>0.097</td>
</tr>
<tr>
<td>Lipid</td>
<td>0.530</td>
<td>0.312</td>
</tr>
<tr>
<td>Protein</td>
<td>-0.112</td>
<td>0.042</td>
</tr>
<tr>
<td>Water</td>
<td>-0.409</td>
<td>0.018</td>
</tr>
</tbody>
</table>

4.1.2 Material decomposition results

Two-material decomposition

The two-material decomposition of the mineral bone to the (compact bone, bone marrow) doublet resulted in mass fractions of $(1.003, -0.00280)$ and the mass density of $1.92 \times 10^3$ kg/m$^3$ for the 25th iteration (see figure 4.6); the true values were $(1, 0)$ and $1.92 \times 10^3$ kg/m$^3$. Figures 4.8 and 4.9 show that these mass fractions and mass densities, respectively, converged already after 10 iterations. Note that the mass-fractions are related via (2.8), i.e., $w_2 = 1 - w_1$.

The two-material decomposition of air to the (lipid, water) doublet resulted in the mass density of approximately zero, see figure 4.7. The mass fractions of lipid and water were of little interest in this case.

![Color maps showing the mass density (right) and the mass fractions of compact bone (left) and bone marrow (middle) for the bone ellipsoid, 25 iterations, $z = 10$ and the two-material decomposition.](image-url)
Figure 4.7: Color maps showing the mass density (right) and the mass fractions of lipid (left) and water (middle) for air, 25 iterations, \( z = 10 \) and the two-material decomposition.

Figure 4.8: The mass-fractions of compact bone (blue) and bone marrow (orange) for the two-material decomposition of the ellipsoid made of bone as a function of the number of iterations.
4.1 Regular DIRA-3D reconstruction

![Relative error of the density for the MD2 compactbone](image)

**Figure 4.9:** The relative error of the density calculated from two material decomposition for the bone ellipsoid as a function of the number of iterations.

**Three-material decomposition**

All materials in the ellipsoids except the compact bone were decomposed into a triplet of (lipid, protein, water). Figures 4.10 and 4.11 show the mass-fractions of triplet components after 25 iterations for slices $z = 10$ and $z = 24$, respectively. Figure 4.12 shows how the mass fraction of the lipid varied with the number of iterations; the calculated mass fraction with its true value is presented in table 4.2. Similar data are shown in figure 4.13 and table 4.3 for the protein, and figure 4.14 and table 4.4 for water.
Figure 4.10: The color maps showing the mass fractions of lipid (top left), protein (top right), water (lower left) and the density (lower right) for the slice $z = 10$ and 25 iterations. The ellipsoids contain adipose tissue, muscle and lipid, see figure 4.2.

Figure 4.11: The color maps showing the mass fractions of lipid (top left), protein (top right), water (lower left) and the density (lower right) for the slice $z = 24$ and 25 iterations. The ellipsoids contain water and protein, see figure 4.3.
4.1 Regular DIRA-3D reconstruction

**Figure 4.12:** The mass fractions of lipid as functions of the iteration number for all ellipsoids except the one made of bone obtained using three-material decomposition.

**Table 4.2:** The calculated mass fraction, $w$, and the true mass fraction, $w_t$, of lipid for the materials in the ellipsoids.

<table>
<thead>
<tr>
<th>Material</th>
<th>$w$ (%)</th>
<th>$w_t$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adipose T.</td>
<td>70.02</td>
<td>70.1</td>
</tr>
<tr>
<td>Muscle</td>
<td>-12.29</td>
<td>-12.8</td>
</tr>
<tr>
<td>Lipid</td>
<td>99.54</td>
<td>100</td>
</tr>
<tr>
<td>Protein</td>
<td>0.164</td>
<td>0.00</td>
</tr>
<tr>
<td>Water</td>
<td>0.507</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Figure 4.13: The mass fractions of protein as functions of the iteration number for all ellipsoids except the one made of bone obtained using three-material decomposition.

Table 4.3: The calculated mass fraction, $w$, and the true mass fraction, $w_t$, of protein for the materials in the ellipsoids.

<table>
<thead>
<tr>
<th>Material</th>
<th>$w$ (%)</th>
<th>$w_t$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adipose T.</td>
<td>2.903</td>
<td>2.90</td>
</tr>
<tr>
<td>Muscle</td>
<td>13.04</td>
<td>12.8</td>
</tr>
<tr>
<td>Lipid</td>
<td>0.231</td>
<td>0.00</td>
</tr>
<tr>
<td>Protein</td>
<td>100.1</td>
<td>100</td>
</tr>
<tr>
<td>Water</td>
<td>-0.263</td>
<td>0.00</td>
</tr>
</tbody>
</table>
4.2 Long object DIRA-3D reconstruction

Figures presented in this section were obtained using the same method as figures in section 4.1. This approach allows a direct comparison between them.

### 4.2.1 Reconstruction for different iterations

Figure 4.15 shows that the reconstructed LAC converged similarly to the LAC in figure 4.1 in the areas without missing data. In the regions with the missing data the LAC can be highly inaccurate [10]. In our case, regions with missing data are shown in figure 3.5 and the artefacts can be seen in the circle segment behind the mineral bone and adipose tissue ellipsoids in figure 4.15. For the first iteration the data were highly inaccurate and the overall accuracy improved for the 25th iteration. For slice \( z = 10 \), with no missing data, a visual comparison shows no difference, see figures 4.2 and 4.16.

**Figure 4.14:** The mass fractions of water as functions of the iteration number for all ellipsoids except the one made of bone obtained using three-material decomposition.

**Table 4.4:** The calculated mass fraction, \( w \), and the true mass fraction, \( w_t \), of water for the materials in the ellipsoids.

<table>
<thead>
<tr>
<th>Material</th>
<th>( w ) (%)</th>
<th>( w_t ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adipose T.</td>
<td>27.07</td>
<td>27.0</td>
</tr>
<tr>
<td>Muscle</td>
<td>99.25</td>
<td>100</td>
</tr>
<tr>
<td>Lipid</td>
<td>0.1799</td>
<td>0.00</td>
</tr>
<tr>
<td>Protein</td>
<td>-0.2542</td>
<td>0.00</td>
</tr>
<tr>
<td>Water</td>
<td>99.23</td>
<td>100</td>
</tr>
</tbody>
</table>
The relative errors (figure 4.18) converged to almost 0 as in table 4.1 except the values for the protein and water ellipsoids, see table 4.5, where the relative errors were 37% and 17%, respectively, for $E_1 = 50$ keV. The reason for this was the amount of missing data in slice $z = 24$, see figure 4.17.

**Figure 4.15:** Color maps of LAC (in $m^{-1}$) for iterations 1 (a), 2 (b), 3 (c), 4 (d) and 25 (e) for the slice $z = 14$ and photon energy $E_1 = 50$ keV obtained using long object DIRA-3D reconstruction. The axis labels show pixel indexes.
4.2 Long object DIRA-3D reconstruction

Figure 4.16: Color maps of LAC (in m$^{-1}$) for energies $E_1 = 50$ keV (a) and $E_2 = 88.5$ keV (b) for the slice $z = 10$ and iteration number 25 obtained using the long object DIRA-3D reconstruction. The axis labels show pixel indexes.

Figure 4.17: The same as in figure 4.15 for slice $z = 24$ and iteration 1 (a), 5 (b) and 25 (c).

Figure 4.18: The relative errors of the LAC as a function of the number of iterations for $E_1 = 50$ keV (a) and $E_2 = 88.5$ keV (b).
Table 4.5: The relative errors $\delta(\bar{\mu}_1)$ and $\delta(\bar{\mu}_2)$ of the reconstructed LAC for energies $E_1 = 50$ keV and $E_2 = 88.5$ keV, respectively.

<table>
<thead>
<tr>
<th>Material</th>
<th>$\delta(\bar{\mu}_1)$ (‰)</th>
<th>$\delta(\bar{\mu}_2)$ (‰)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adipose T.</td>
<td>0.250</td>
<td>0.250</td>
</tr>
<tr>
<td>Compact Bone</td>
<td>1.240</td>
<td>0.270</td>
</tr>
<tr>
<td>Muscle</td>
<td>-0.438</td>
<td>0.091</td>
</tr>
<tr>
<td>Lipid</td>
<td>0.523</td>
<td>0.306</td>
</tr>
<tr>
<td>Protein</td>
<td>-370</td>
<td>-360</td>
</tr>
<tr>
<td>Water</td>
<td>-165</td>
<td>-109</td>
</tr>
</tbody>
</table>

4.2.2 Material decomposition results

Two-material decomposition

The missing data in evaluated slices had little impact on the two-material decomposition for the bone ellipsoid mostly because the ROI was outside of the region with the missing data. The decomposition of the mineral bone ellipsoid resulted in mass fractions of $(1.003, -0.002832)$ and the mass density of $1.9179 \times 10^3$ kg/m$^3$ for the 25th iteration (see figure 4.19). A comparison of figures 4.8 and 4.21 shows that the difference is negligible. The density (see figure 4.22) and the decomposition of air (see figure 4.20) show no real difference from the results in section 4.1.

Figure 4.19: Color maps showing the mass density (right) and the mass fractions of compact bone (left) and bone marrow (middle) for the bone ellipsoid obtained using 25 iterations, $z = 10$ and the two-material decomposition.
Figure 4.20: Color maps showing the mass density (right) and the mass fractions of lipid (left) and water (middle) for air obtained using 25 iterations, $z = 10$ and the two-material decomposition.

Figure 4.21: The mass-fractions of compact bone (blue) and bone marrow (orange) for the two-material decomposition of the ellipsoid made of bone as a function of the number of iterations.
Figure 4.22: The relative error of the density calculated from two material decomposition for the bone ellipsoid as a function of the number of iterations.

Three-material decomposition

The three-material decomposition applied on ellipsoids in the second layer provided highly inaccurate results. A visual comparison of figures 4.23 and 4.10 shows no differences in the mass fractions. The figure 4.24 shows that missing data can lead to wrong predictions of mass fractions (cf. figure 4.11 which is not based on missing data). The ellipsoids located in the first layer had mass fractions close to those presented in section 4.1 while the ellipsoids located in the second layer were affected by the missing data. This is represented in figures 4.25, 4.26, 4.27 and tables 4.6, 4.7, 4.8.
4.2 Long object DIRA-3D reconstruction

Figure 4.23: The color maps showing the mass fractions of lipid (top left), protein (top right), water (lower left) and the density (lower right) for the slice $z = 10$ and after 25 iterations. The ellipsoids contain adipose tissue, muscle and lipid, see figure 4.2.

Figure 4.24: The color maps showing the mass fractions of lipid (top left), protein (top right), water (lower left) and the density (lower right) for the slice $z = 24$ and after 25 iterations. The ellipsoids contain water and protein, see figure 4.3.
Figure 4.25: The mass fractions of lipid as functions of the iteration number for all ellipsoids except the one made of bone obtained using three-material decomposition. The water and protein ellipsoids were affected by missing data.

Table 4.6: The calculated mass fraction, $w$, and the true mass fraction, $w_t$, of lipid for the materials in the ellipsoids.

<table>
<thead>
<tr>
<th>Material</th>
<th>$w$ (%)</th>
<th>$w_t$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adipose T.</td>
<td>70.03</td>
<td>70.10</td>
</tr>
<tr>
<td>Muscle</td>
<td>-12.29</td>
<td>-12.80</td>
</tr>
<tr>
<td>Lipid</td>
<td>99.54</td>
<td>100</td>
</tr>
<tr>
<td>Protein</td>
<td>59.42</td>
<td>0.00</td>
</tr>
<tr>
<td>Water</td>
<td>98.79</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Material Decomposition for Protein

Figure 4.26: The mass fractions of protein as functions of the iteration number for all ellipsoids except the one made of bone obtained using three-material decomposition. The water and protein ellipsoids were affected by missing data.

Table 4.7: The calculated mass fraction, $w$, and the true mass fraction, $w_t$, of protein for the materials in the ellipsoids.

<table>
<thead>
<tr>
<th>Material</th>
<th>$w$ (%)</th>
<th>$w_t$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adipose T.</td>
<td>2.918</td>
<td>2.900</td>
</tr>
<tr>
<td>Muscle</td>
<td>13.04</td>
<td>12.80</td>
</tr>
<tr>
<td>Lipid</td>
<td>0.231</td>
<td>0.00</td>
</tr>
<tr>
<td>Protein</td>
<td>2.305</td>
<td>100</td>
</tr>
<tr>
<td>Water</td>
<td>-0.059</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Figure 4.27: The mass fractions of water as functions of the iteration number for all ellipsoids except the one made of bone obtained using three-material decomposition. The water and protein ellipsoids were affected by missing data.

Table 4.8: The calculated mass fraction, $w$, and the true mass fraction, $w_t$, of water for the materials in the ellipsoids.

<table>
<thead>
<tr>
<th>Material</th>
<th>$w$ (%)</th>
<th>$w_t$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adipose T.</td>
<td>27.05</td>
<td>27.00</td>
</tr>
<tr>
<td>Muscle</td>
<td>99.26</td>
<td>100</td>
</tr>
<tr>
<td>Lipid</td>
<td>0.177</td>
<td>0.00</td>
</tr>
<tr>
<td>Protein</td>
<td>13.32</td>
<td>0.00</td>
</tr>
<tr>
<td>Water</td>
<td>0.209</td>
<td>100</td>
</tr>
</tbody>
</table>
5.1 Result

Partial volume effect appears for instance when bone fills a part of a voxel and air fills the other part. The reconstruction algorithm then averages the LAC of these two materials. The weighted average is affected by the amount of bone and air in the voxel. The partial volume effect leads to partial volume artefacts in the reconstructed image. The artefacts demonstrate themselves as borders with different intensity than the voxels in the central part of the region or as streaks emanating from the border. The partial volume artefacts are more apparent in low resolution images. In our case the maximum diameter of the ellipsoid was only 26 pixels with the lowest value being 13 pixels. This resulted in circular regions around the bone ellipsoid (figure 4.6) and in streaks between for instance the muscle and bone ellipsoids (figure 4.1).

Partial volume effect combined with edge effects generated by the image reconstruction algorithm results in profiles that are not constant across the ellipsoid, see figure 4.4. To eliminate the artefacts close to the ellipsoid edges the spherical ROI used for analysis used the diameter of 8 voxels. This is the maximum diameter that can be used without the result being affected by overshoot in the z direction.

5.1.1 Regular DIRA-3D

The first couple of iterations were strongly affected by cone-beam and beam-hardening artefacts, see figure 4.1. Similar observations were described in the literature. In case of cone-beam artefacts see for instance the performance of the PI-method for mono-energetic photons in Ref. [10]. For the beam hardening
artefact see for instance Ref. [12].

For the 25th iteration almost no cone-beam or beam hardening artefacts were present in the reconstructed LAC colormaps, see figure 4.2. Moreover the measured values were close to true values, see table 4.1. These two facts indicate that the novel DIRA-3D algorithm works properly. It is important to note that the convergence of the algorithm has not been theoretically proven yet. A further analysis of the convergence rate could show that a lower number of iterations than 25 is needed for convergence, see section 5.3.

There is a linear relation between the reconstructed LAC and the mass fractions of the doublet and triplet components, and thus the accuracy of the mass fraction reflects the accuracy of the LAC. In some cases the system of linear equations may be ill conditioned causing amplification of the errors.

### 5.1.2 Long object reconstruction

Iterative algorithms not designed to handle the long object problem suffer from the fact that the region with useful data diminishes with each iteration step [10]. In DIRA-3D this would lead to the removal of any slice with missing data, for instance any slice above $z = 10$ in figure 4.16, would be discarded. The handling of the long object problem as described in section 3.2 eliminated the problem with the diminishing ROI. Both the convergence rate and the accuracy of the results were similar to the case without truncated data in the regions with no missing data, see figure 4.15. As for slices with more missing data (see figure 4.17) the algorithm never converged because it did not have sufficient data for reconstruction. This is expected from theory [10].

### 5.2 Method

As discussed in section 1.3, alternatives to the PI-method are WFBP [19] and Katsevich algorithm [8].

- WFBP has the advantage of being able to reconstruct data outside the Tam window. It reduces the patient dose since the algorithm uses all the projection data. It is used in SIEMENS CT-scanners.

- The Katsevich algorithm is accurate, it gives no cone-beam artefacts. The disadvantage of the algorithm is that it discards data outside the Tam window.

In both cases, the source code was not available to the author. The PI-method has the advantage of being a simple algorithm whose source code was readily available to the author (the code used in Ref. [10]). The disadvantages of the PI-method are that (i) it increases patient dose because it discards projection data and (ii) the method is non-exact.

There are not many algorithms comparable to DIRA-3D. One algorithm that can be compared is MDIR [9]. Ref. [9] presents two experiments. The first one fea-
tures two orthogonal scans of a phantom with a scanning radius of 100 meters. This results in almost parallel incoming rays producing almost no cone-beam artefacts. The second one used a real CT-scanner operating in spiral mode. However the scanned object consisted only of cylinders which produced no cone-beam artefacts. For this reason the ability of MDIR to suppress cone-beam artefacts is questionable.

The computational load of DIRA-3D was an issue during the algorithm development. To reduce the CPU time three phantoms with similar structure (see figure 3.1) but with different resolutions were used:

- $32 \times 32 \times 12$, CPU time of approximately 1 second per iteration.
- $64 \times 64 \times 24$, CPU time of approximately 10 seconds per iteration.
- $128 \times 128 \times 48$, CPU time of approximately 100 seconds per iteration.

The first two phantoms were used for code development only. It is expected that a phantom with the resolution of $256 \times 256 \times 96$ would need approximately 1000 seconds per iteration.

The MATLAB/C environment was chosen since MATLAB provides a debugger and several helpful GUI functions (such as visualization) while still being able to execute native C code [13]. This also means that there should be no benefit in speed from implementing the algorithm in the C language since all computational heavy steps were executed as C code.

### 5.2.1 Phantom

There is several aspects of the phantom that could be improved. The phantom was simpler than the phantom used in DIRA-2D [12] and had a lower resolution. The lower resolution affected the presence of artefacts in the image as discussed in section 5.1. The phantom was loaded as a voxel array. In a realistic case the algorithm would load cone-beam projections and rebinn them into the input semi-parallel projections.

### 5.2.2 Source criticism

This work is based on Refs. [10, 12, 23]. Ref. [10] was published by IEEE as a peer reviewed scientific article. Ref. [23] was published by the Linköping University as doctoral thesis. Both of these sources discuss different approaches to solve the presented problem. The approach providing the best result was taken from Ref. [10]. The original PI method was taken from Ref. [23]. Ref. [12] was published in the Medical Physics journal as a peer reviewed scientific article.
5.3 Possible extensions

5.3.1 WFBP instead of PI-method

The PI-method is unfavorable to WFBP since it removes usable data (section 2.7). There are more articles describing the application of WFBP (see Ref. [21] for more information) and the method is implemented in commercial CT-scanners [19]. For DIRA-3D it would allow to reduce the patient dose by using all the projection data and the results would be more directly comparable to results published in scientific literature. The method could be implemented as described in Ref. [20].

5.3.2 Water-beam hardening correction

Water-beam hardening correction is used in DIRA-2D to speed up the convergence of the algorithm [12]. Water-beam hardening correction is performed on the measured polyenergetic projections before the initial reconstruction. In case of a water phantom it fully suppresses the beam hardening artefact. For other phantoms consisting of for instance soft tissues and bone it typically improves the result. For phantoms containing a lot of bone, however, a special treatment may be needed. It is expected that the water beam hardening correction in DIRA-3D would increase the convergence speed. The implementation could be done as in DIRA-2D i.e., in the 0th iteration, see Ref. [12].

5.3.3 Speed the algorithm up using parallel computing

Parallel computing has the potential to speed up the algorithm. For instance MATLAB parfor-loops shortened the computation times to approximately 1/3: from about 300 s per iteration to 100 s on a server with 12 core Intel Xeon X5650 at 2.67 GHz and 82 GB of RAM. Further increase in speed could be achieved by using specialized hardware, for instance the graphics processing units (GPUs). However, previous work showed that the performance of DIRA-2D does not scale linearly on GPUs [14] since there is a need to transfer large amount of data between the GPU and the RAM.

5.3.4 Cone-beam projection data

Real CT-scanners produce cone-beam data. To validate DIRA-3D against real CT data the ability to work with cone-beam projections has to be implemented. It can be done by cone-beam projection to semi-parallel projection rebinning. This was not done due to time restrictions (see section 1.4).
The aim was to expand the DIRA algorithm so that it would be able to reconstruct the LAC from helical DECT data and perform subsequent material decomposition. This was achieved by implementing (i) polyenergetic semi-parallel projection generation, (ii) mono-energetic parallel projection generation and (iii) the PI-method for image reconstruction in the newly designed DIRA-3D algorithm. The original idea of the DIRA, i.e., the conversion of non-ideal projections to ideal projections in the iterative loop was preserved in DIRA-3D. In this algorithm the ideal projections were parallel mono-energetic projections which the FBP can reconstruct exactly. This choice suppressed beam-hardening and cone-beam artefacts.

The computation experiments showed that the accuracies of the resulting LAC and mass fractions were comparable to the ones of the original DIRA [12]. For instance the relative errors of LACs determined in the spherical ROIs were less than 1.24 ‰.

The introduction of truncated data in the computation experiment showed that the algorithm converged in the regions not affected by the missing data at the same rate as in the case with no truncated data. In the regions affected by the missing data the LAC can not be trusted in general. In our case some regions in the affected slices were strongly distorted but still provided useful visual information.

The results converged after 10 iterations for the low resolution phantom; one iteration took about 100 seconds. It is expected that (i) the same number of iterations would be needed for high resolution phantoms, and (ii) hardware and software optimization would help to achieve reconstruction times usable in clinical practice.
Appendix
A.1 Parameter file

This section describes the input file of the LoopNormal.c code (provided by Maria Magnusson). The LoopNormal.c code implements the PI-method as described by Ref. [10], see also figure 2.6. This input file is also read by the DIRA-3D code. Nevertheless some parameters are not used there.

```
# Parameter file
lNamebase = kulor128

# Directories and phantom filenames
phantomDir = ../voxelize
resultDir = tests
phantomFilename = -
voxPhantomFilename = kulor_phm_vox128Smooth

# Geometrical properties
gammaMax = 26.0
coneBeamAngle = low
R = 2.28
Nlambda = 2
resFactS = 2
resFactT = 1.5

# Iteration coefficients (not used in DIRA-3D)
alpha1 = 1.0
alpha2 = 1.0
```
alpha3 = 1.0
alpha4 = 1.0
alpha5 = 1.0
alpha6 = 1.0

# Interpolation
fInterpolationT = sincot
fFiltLenT = 6
a1T = 0.75
a2T = 0.52
fInterpolationS = linear
fFiltLenS = 2
a1S = 1
a2S = 1
bInterpolationT = linear
bFiltLenT = 2
bInterpolationS = linear
bFiltLenS = 2
pInterp = 3
pInterpBack = 1

# Other test data
lowPassT = 0
longTest = 0
recSource = digitalPhantom
projCmpTest = 0
complementaryRays = 0
lpFiltT = 0
lpFiltS = 0
printD0 = 1

## A.2 Routine calculating the detector parameters

This section describes the calculation of detector parameters for the semi-parallel geometry. The MATLAB routine is based on the calculations of detector parameters in the LoopNormal.c code. The parameter s is a structure containing the parameters of the input file.

```matlab
function [ Nt, Ns, Ntheta, NlambdaTheta, Rx, P, deltaT, ...
, deltaS, tMax, voxelWidth, offsT, offsS, Nt2, Ns2, s ] = 
calculateDetectorParam(s, Nx, ~, ~)
%Function for calculating all the needed parameters for the
detector (semi-parallel geometry)

SYMMETRIC_MODELS = 0;
```
%CALCULATE DETECTOR SIZE
Nt = ceil(Nx*s.resFactT/2)*2;
switch (s.coneBeamAngle)
    case 'high'
        Ns = Nx /4*s.resFactS;
    case 'low'
        Ns = Nx /8 *s.resFactS;
    otherwise
        Ns = Nx /16 *s.resFactS;
end

switch (Nx)
    case 32
        Ntheta = 100;
    case 64
        Ntheta = 200;
    case 128
        Ntheta = 400;
    case 256
        Ntheta = 800;
    otherwise display ('Error, Nx does not equal 32 , 64 ,
                       128 or 256');
end

NlambdaTheta = s.Nlambda * Ntheta;
Rx = s.R * Nx/2;
P = Ns/s.resFactS*2;
fprintf ('Pitch = %f deltaX = %f length units\n\n', P, P/(Nx/2));
fprintf ('Cone beam angle = +− %f\n', atan(P/4/Rx)*180/pi);

deltaT = Nx/Nt;
deltaS = 1/s.resFactS;
s.gammaMax = s.gammaMax * pi/180;
fprintf ('gammaMax = %f\n', s.gammaMax);
tMax = sin(s.gammaMax)*Rx;
voxelWidth = 6.9000e−04 * 4; %pixel−size used in DIRA−2D * 4
offsT = floor (s.bFiltLenT/deltaT/2);
if SYMMETRIC_MODELS
    offsS = 0;
else
    if (s.pInterpBack == 1)
        offsS = floor (s.bFiltLenS/2/deltaS);
    else
o f f s S = floor( Rx / (Rx-Nx/2) * s . b F i l t L e n S / 2 / d e l t a S );

end

N t 2 = N t + 2 * o f f s T ;
N s 2 = N s + 2 * o f f s S ;
end


