Improving Robustness of Tomographic Reconstruction Methods

Folkert Bleichrodt

# Improving Robustness of Tomographic Reconstruction Methods

ISBN: 978-94-6259-869-0

Folkert Bleichrodt

## Uitnodiging

U bent van harte uitgenodigd voor het bijwonen van de openbare verdediging van mijn proefschrift

***Improving Robustness of Tomographic Reconstruction Methods***

op dinsdag 10 november 2015 om 15:00 in de Senaatskamer van het Academiegebouw van de Universiteit Leiden, Rapenburg 73 te Leiden. Aansluitend is een receptie.

## Invitation

You are cordially invited to attend the public defence of my PhD thesis

***Improving Robustness of Tomographic Reconstruction Methods***

on Tuesday 10th November 2015 at 15:00 in the Senate Room of the Academy Building of the Leiden University, Rapenburg 73 in Leiden. Followed by a reception.

Folkert Bleichrodt
fbleichrodt@gmail.com
06-17278348

# Improving Robustness of Tomographic Reconstruction Methods

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof. mr. C.J.J.M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op dinsdag 10 november 2015
klokke 15:00 uur

door

Folkert Bleichrodt

geboren te Gouda
in 1987

Promotor: Prof. dr. K. Joost Batenburg

Samenstelling van de promotiecommissie

| | | |
|---|---|---|
| Voorzitter: | Prof. dr. Peter Stevenhagen | |
| Secretaris: | Prof. dr. S.J. Edixhoven | |
| Overige leden: | Prof. dr. Sara Bals | (Universiteit Antwerpen) |
| | Prof. dr. Rob H. Bisseling | (Universiteit Utrecht) |
| | Dr. Tristan van Leeuwen | (Universiteit Utrecht) |

# Improving Robustness of Tomographic Reconstruction Methods

Folkert Bleichrodt

ISBN: 978-94-6259-869-0

The abstract pattern on the cover was generated by a Matlab script. A projection matrix was generated for the 2D parallel beam geometry, 12 projection angles, 1448 detector elements and a reconstruction size of $1024 \times 1024$. Using the singular value decomposition the first 21 singular values and corresponding singular vectors were computed. The pattern on the cover is based on the 21st right-singular vector.



The research in this thesis has been financially supported by the Netherlands Organisation for Scientific Research (NWO), programme 639.072.005. It was carried out at Centrum Wiskunde & Informatica (CWI), Amsterdam.

# Contents


**1 Introduction and outline of this thesis** **1**
1.1 Tomography . . . 1
1.1.1 Application areas . . . 2
1.1.2 Challenges in tomographic reconstruction . . . 4
1.2 Mathematics of tomography . . . 4
1.2.1 The Radon transform . . . 4
1.2.2 Algebraic reconstruction methods . . . 5
1.3 Artifacts . . . 6
1.3.1 Mechanical instabilities . . . 6
1.3.2 Low-dose and limited data . . . 8
1.3.3 Other unmodeled errors . . . 8
1.4 Overview . . . 9

**2 Automatic optimization of alignment parameters for tomography datasets** **11**
2.1 Introduction . . . 11
2.2 Methods and implementation . . . 13
2.2.1 Model and notation . . . 13
2.2.2 Levenberg–Marquardt . . . 17
2.2.3 Computing the Jacobian . . . 17
2.2.4 Multi-resolution . . . 18
2.3 Experiments . . . 19
2.4 Results . . . 20
2.5 Discussion . . . 23
2.6 Conclusions . . . 24

**3 Aligning projection images from binary volumes** **25**
3.1 Introduction . . . 25
3.2 Method and implementation . . . 27
3.2.1 Geometrical parameters . . . 27
3.2.2 Mathematical formulation . . . 30
3.2.3 Projection matching . . . 31
3.2.4 Discrete tomography . . . 33
3.3 Projection matching with discrete tomography . . . 33
3.4 Experiments and results . . . 36
3.4.1 Experiment I – the effect of discrete tomography . . . 38

3.4.2 Experiment II – aligning projection data . . . 38
3.4.3 Experiment III – aligning noisy projection data . . . 40
3.4.4 Performance considerations . . . 42
3.5 Discussion . . . 44
3.6 Conclusions . . . 45

**4 SDART: an algorithm for discrete tomography from noisy projections 47**
4.1 Introduction . . . 47
4.2 The original DART algorithm . . . 48
4.2.1 Notation and concepts . . . 49
4.2.2 Algorithm details . . . 50
4.3 Soft DART . . . 52
4.3.1 Selecting a penalty matrix . . . 53
4.3.2 Solving the soft constrained system . . . 54
4.4 A numerical study . . . 55
4.4.1 Behavior of DART compared to SDART . . . 55
4.4.2 Selecting the regularization parameter . . . 58
4.5 Experiments and results . . . 61
4.5.1 Experiment I – basic validation . . . 62
4.5.2 Experiment II – the effect of noise . . . 63
4.5.3 Experiment III – adding more projection angles . . . 65
4.5.4 Experiment IV – experimental data . . . 66
4.6 Conclusions . . . 69

**5 Analysis and removal of offset and scaling artifacts in tomography 71**
5.1 Introduction . . . 71
5.2 Origin of offsets and scalings in projection data . . . 73
5.2.1 Tomography . . . 73
5.2.2 Beer–Lambert's law . . . 74
5.3 Analysis of global offset artifacts . . . 75
5.4 Offset estimation algorithm . . . 79
5.4.1 Global offset estimation . . . 80
5.4.2 Local offset estimation . . . 82
5.5 Scale estimation algorithm . . . 83
5.6 Experiments and results . . . 84
5.6.1 Slice-based simulation experiments . . . 85
5.6.2 3D simulation experiments . . . 90
5.6.3 Experimental electron tomography dataset . . . 92
5.7 Discussion and conclusions . . . 94

**6 Robust artifact reduction in tomography using Student's t data fitting 97**
6.1 Introduction . . . 97
6.2 Methods . . . 98
6.3 Experiments and Results . . . 100
6.3.1 Metal artifact reduction . . . 100
6.3.2 Defective camera pixels . . . 102

6.3.3 Randomized projection images . . . 102
6.4 Discussion and conclusions . . . 103

**7 Easy implementation of advanced tomography algorithms using the ASTRA toolbox with Spot operators 105**
7.1 Introduction . . . 105
7.2 Tomography . . . 106
7.3 Software implementation . . . 108
7.3.1 The ASTRA toolbox . . . 108
7.3.2 The Spot toolbox . . . 109
7.3.3 The ASTRA Spot operator . . . 110
7.4 Case studies . . . 112
7.4.1 Custom SIRT implementation . . . 112
7.4.2 Cone beam reconstruction . . . 113
7.4.3 Sparse image reconstruction . . . 115
7.4.4 Sparse wavelet reconstruction . . . 115
7.4.5 TV-minimization using the Chambolle–Pock algorithm . . . 117
7.5 Performance benchmarks . . . 119
7.5.1 The forward and backprojection operations . . . 120
7.5.2 Overhead of the Spot toolbox . . . 121
7.6 Discussion and conclusions . . . 124
7.A Geometries in the ASTRA toolbox . . . 126
7.A.1 Volume geometry . . . 127
7.A.2 3D parallel beam . . . 127
7.A.3 Cone beam . . . 128

**Bibliography 129**

**Samenvatting 139**

**Curriculum Vitae 143**

**Acknowledgement 145**

# Chapter 1

# Introduction and outline of this thesis

In this chapter we introduce tomography and discuss the principal concepts of this thesis. We discuss several challenges in tomographic reconstruction and we provide a motivation for the development of methods for dealing with these challenges. Finally, we provide an overview of the main results of this thesis.

## 1.1 Tomography

Tomography is a technique to reconstruct the three dimensional structure of an object from projection images of that object. A well known application is X-ray imaging used for medical diagnosis. From a single X-ray image a lot of information can be obtained: for example, it is possible to detect fractures in bones (see Fig. 1.1), or to locate contrast agents injected in blood vessels. Because the morphology of the human body is well-known we can often derive this 3D structural information from a 2D projection image. However, if no general shape information of the object is known beforehand, we cannot retrieve this information. Therefore, tomographic reconstruction is an essential step to retrieve 3D structural information.

Figure 1.1: X-ray image of a fractured collarbone. Photo: J. Bizzie.

In tomographic imaging, a set of projection images is acquired from different view angles by directing a beam of radiation (*e.g.*, X-rays) to the object and recording the intensity profile of the beam after it has passed through the object on a detector. Since the geometry of this projection acquisition is known with great precision, the view directions and positions of the detector create a correspondence between the measured 2D projections and the 3D density distribution of the object. The intensity recorded by a single point on the detector is directly related to the amount of material and its density on the line between the detector and the radiation source. Mathematically, the measurements are approximated by line integrals. The problem of reconstruction is to find a (usually discretized) representation of the unknown object that matches these line integrals as close as possible. Since the number of projections is finite and measurements are distorted by noise, the reconstruction problem in tomography is an ill-posed inverse problem.

After the projections have been acquired, a *reconstruction algorithm* is applied to compute an approximate solution to the inverse problem, which is called a *reconstruction*. A reconstruction algorithm typically consists of a sequence of *forward* and *backprojection* operations. The *filtered backprojection* (FBP) algorithm consists of a filtering step and only one backprojection [KS01]. There are other methods that iteratively refine the reconstruction and are based on a linear equation system. These methods are known as *algebraic reconstruction methods*. A few methods that are commonly used are ART, SART, SIRT and LSQR [GBH70; AK84; Gil72; PS82].

### 1.1.1 Application areas

Computerized tomography is a well established technique in the medical imaging community. The medical scanners that are used today have a very high accuracy and stability with respect to the imaging hardware. Most challenges that can be encountered in the reconstruction problem are now well understood.

This situation is quite different for a broad variety of tomography applications in the experimental scientific community. Due to the development of image acquisition techniques that encompass the nanoscale (electron microscopy [Sco+12]) up to astronomic scales (astro tomography [ABS12]), there are still major challenges that need to be addressed. Each of these applications demands sophisticated and specialized imaging equipment. In these experiments there are currently major challenges that have not been fully addressed with respect to stability of the hardware, mathematical modeling and distortions in the optics system and stability of the source, among many others. We now discuss some of these application domains.

#### Electron tomography

Electron microscopy is an imaging technique that uses an electron beam that interacts with a sample and is often used in biology and materials science. Due to the small de Broglie wavelength sub-ångström resolution can be achieved [Aer+11;

Sco+12]. Initially only 2D projection images were obtained and studied, where either the unscattered beam is recorded or (part of) the scattered electron beam. However, these techniques only provide limited analysis of 3D structures. Tomography became possible when tilt sample holders were introduced. Tomography at these small scales demands extremely stable imaging hardware to prevent movement of the sample or tilt axis that is greater than the pixel size of the detector.

A major challenge in electron tomography is to estimate and correct for this sample drift and instabilities of the microscope after the data has been recorded (referred to as *alignment*). Moreover, most reconstruction algorithms are based on the assumption that the measured intensity depends linearly on the thickness of the sample, but in reality the image formation is partly nonlinear, *e.g.*, due to multiple scattering of electrons within the sample [Bro+12]. Therefore, the reconstruction can potentially be improved by incorporating such nonlinear effects.

#### Desktop micro-CT

For the observation of small samples at the micron scale, desktop X-ray micro-CT scanners provide a reliable way to routinely scan objects of a few centimeters in size. These scanners are often made by commercial parties and are typically employed for scanning of biological, geological and industrial samples [Bri+10; GSW00; Car+12]. For certain applications, the degrees of freedom in the geometry of desktop scanners are not sufficient and therefore custom scanners have been developed for use in laboratories [Mas+13].

Due to imperfections or miscalibration of the camera, circular image distortions, known as *ring artifacts*, are often encountered in micro-CT. Since the X-ray beam is not purely monochromatic (*i.e.*, has multiple energy levels), each part of the spectrum of the X-rays is attenuated differently. This leads to *beam hardening* artifacts in the reconstruction if this effect is not correctly modeled in the reconstruction algorithm [Cas+02].

#### Synchrotron tomography

Synchrotron facilities provide a high intensity monochromatic X-ray source for high quality projection acquisition, achieving sub-micron resolution [Wil+11]. Synchrotron X-ray imaging has many advantages compared to scanning with laboratory setups. Due to the high intensity of the beam, projection images can be obtained very rapidly and with high signal-to-noise ratio. Moreover, the generated X-ray beam is almost parallel, such that projections of slices of the object are independent, which allows slice based reconstruction. By making the beam monochromatic, beam hardening effects can be fully eliminated. This makes synchrotron tomography ideally suited for in-situ experiments and 4D tomography, where a time series of reconstructions of the sample is obtained [Mar+11; Mom+11].

In synchrotron tomography there are still challenges with respect to alignment, ring artifacts and phase-contrast effects that cause nonlinearity. Phase-contrast can be exploited to further improve the contrast of the reconstructions [Pag+02].

### 1.1.2 Challenges in tomographic reconstruction

These developments in scanning techniques and experiments have introduced new challenges for the reconstruction problem:

- **Stability:** As the scale of the imaged sample decreases, the stability of the scanner hardware becomes increasingly important;
- **Low-dose:** The radiation dose is often limited due to destructive nature of the radiation, or to reduce acquisition time;
- **Limited data:** To limit radiation dose the number of projections can be small;
- **Unpredictable data perturbations:** The projection data can be perturbed in ways that can not be modeled effectively.

Reconstruction methods, such as analytical reconstruction using filtered backprojection or algebraic reconstruction methods, are all affected by these problems. If these methods are applied without pre- or post-processing the data, errors are introduced in the reconstruction. Each algorithm has its own strong and weak points with respect to these problems. For example, the filtered backprojection is rather robust with respect to noise, while algebraic reconstruction methods in general perform better if a small number of projections are available.

After an introduction to the mathematics we will discuss the challenges we mentioned and their effect on the reconstruction in more detail.

## 1.2 Mathematics of tomography

In this section we will introduce the mathematics involved in tomographic reconstruction.

### 1.2.1 The Radon transform

The mathematics behind the projection acquisition can be described by the *Radon transform* [NW01; Hel99]. In this section we focus on the Radon transform in two dimensions, where we consider a single slice of the object.

In tomographic reconstruction we aim to recover the density or attenuation function

$$f : \mathbb{R}^2 \mapsto \mathbb{R}$$

of the object we consider. The function $f$ assigns to each point of the object $(x, y) \in \mathbb{R}^2$ an attenuation value, which depends on the material at that position. Note that the domain of $f$ is two-dimensional, because we focus here on a single

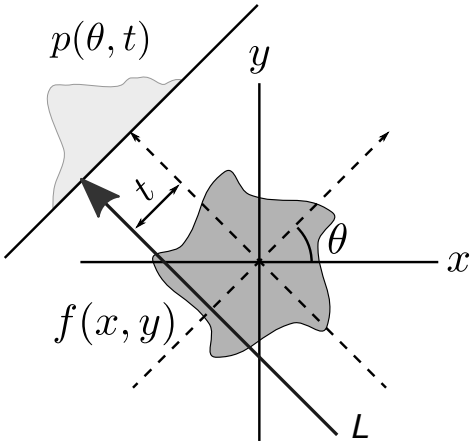



Figure 1.2: Schematic of the Radon transform of the attenuation function $f$ in two dimensions.

slice of the object. A projection is modeled as a set of line integrals of the function $f$, along straight lines. An example for the parallel beam geometry is shown in Fig. 1.2. The line $L$ can be parametrized with respect to the arc length $\ell$:

$$(x(\ell), y(\ell)) = (\ell \sin\theta + t\cos\theta, t\sin\theta - \ell\cos\theta).$$

which leads to

$$\mathscr{R}[f](\theta, t) := p(\theta, t) = \int_{-\infty}^{\infty} f(\ell\sin\theta + t\cos\theta, t\sin\theta - \ell\cos\theta)\,\mathrm{d}\ell,$$

which is the Radon transform of $f$. Exact reconstruction of $f$ is possible by inverting the Radon transform if we know $p(\theta, t)$ for $\theta \in [0, \pi)$ and $t \in \mathscr{T}$, where $\mathscr{T} \subset \mathbb{R}$ is the support of $p(\theta, t)$ with respect to $t$. However, in a practical CT scanner, we only have a sampling of $p(\theta, t)$, since the number of projections is finite and the set $\mathscr{T}$ is discrete due to the discrete nature of the detector. Therefore, a discrete model of the projection acquisition is needed.

### 1.2.2 Algebraic reconstruction methods

Throughout this thesis we focus on the class of algebraic reconstruction methods. These methods are based on a discretization of the unknown attenuation function $f$ as an image, *i.e.*, we represent the function on a grid of pixels (or voxels, in 3D), where the pixel value (referred to as *gray value*) represents the attenuation coefficient of the material within the pixel. The projections are also discretized on a pixel grid (which is an image in the three dimensional case).

Using this discrete representation we can model not only a parallel beam geometry, but we can also represent fan beam and cone beam geometries. An example of such a discrete model is illustrated in Fig. 1.3. The gray values $x_j$

along the depicted line form a piecewise constant function. Therefore, the line integral along this line is just a linear combination of the gray values, where the weights $w_{ij}$ are determined by the length of the line intersecting that pixel. For each pixel on the detector we can write:

$$p_i := \sum_j w_{ij} x_j,$$

where $w_{ij}$ is the length of the line segment in pixel $j$. This discretization is known as the line model [Sid85]. The full set of equations leads to the linear equation system:

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p}, \tag{1.1}$$

where the (vectorized) image $\boldsymbol{x}$ is related to the modeled projections $\boldsymbol{p}$ by the *projection matrix* $\boldsymbol{W}$.

Note that the gray values are unknown and we need to obtain them from the projection data $\boldsymbol{p}$ by solving the linear system (1.1). In a practical experiment the projections are perturbed by a noise term $\boldsymbol{\epsilon}$

$$\tilde{\boldsymbol{p}} := \boldsymbol{p} + \boldsymbol{\epsilon}. \tag{1.2}$$

Algebraic reconstruction methods compute an approximate solution of Eq. (1.1) by minimizing a cost function $\rho(\boldsymbol{W}\boldsymbol{x} - \tilde{\boldsymbol{p}})$, which penalizes the difference between the computed projections $\boldsymbol{W}\boldsymbol{x}$ of the image $\boldsymbol{x}$ and the vector $\tilde{\boldsymbol{p}}$ of measured projections:

$$\underset{\boldsymbol{x}}{\text{minimize}}\ \rho(\boldsymbol{W}\boldsymbol{x} - \tilde{\boldsymbol{p}}). \tag{1.3}$$

The common choice for $\rho(\cdot)$ is the $\ell_2$-norm in which case Eq. (1.3) is reduced to a least squares problem.

## 1.3 Artifacts

In this section we will discuss the effects of the problems described in Section 1.1.2 on the discretized model of Eq. (1.1). These problems lead to errors in the governing equations Eq. (1.1) which subsequently lead to errors in the reconstruction. If these errors are not corrected by preprocessing or post-processing the data, or by adapting the equation system to match the experimental setting, image distortions, known as *artifacts*, are produced in the reconstruction. Common image distortions are blurring, streaks, noise or smearing of image details.

### 1.3.1 Mechanical instabilities

At small scales it is very challenging to keep the scanner setup in perfect *alignment*. Slight vibrations in the detector or source can cause shifts and rotations in the projection images. These rigid motions have to be corrected for to obtain accurate reconstructions.

In Fig. 1.4 a series of projection images is shown recorded in an electron microscope. A clear shift can be observed in both horizontal and vertical directions.

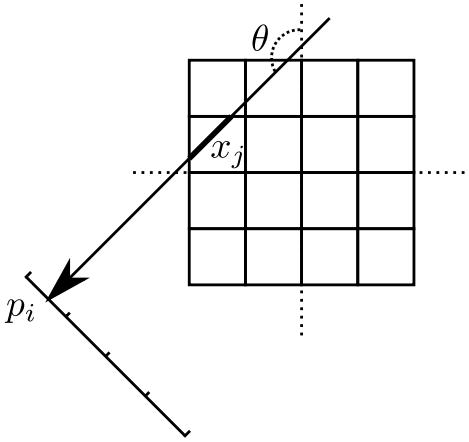



Figure 1.3: Discrete representation of the projection acquisition using a line length model. The unknown gray value corresponding to pixel $j$ is denoted $x_j$ and its contribution to detector measurement $p_i$ is $w_{ij}x_j$, where $w_{ij}$ is the length of the line segment indicated in bold. The projection angle is indicated by $\theta$.

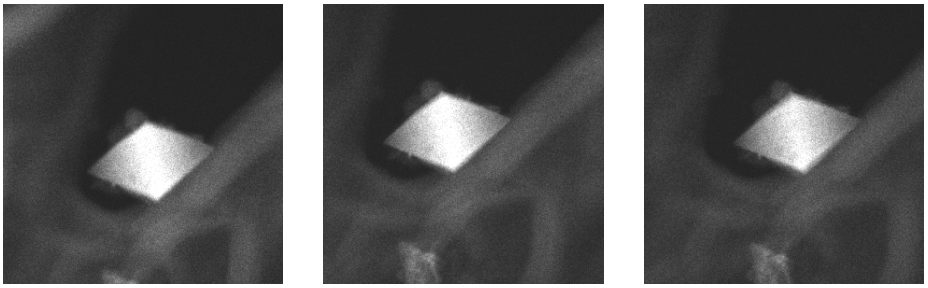


Figure 1.4: Three consecutive projection images recorded by an electron microscope. Considerable shifts in the projections can be observed, due to mechanical instability. Source: Electron Microscopy for Materials Science (EMAT), University of Antwerp.

Ideally the rotation axis of the object is centered with respect to the center of the detector for every projection image.

Essentially, the geometry of the projection acquisition is distorted such that the projection matrix $\boldsymbol{W}$ does not capture the actual geometry of the scanner. The projection matrix of Eq. (1.1) has to be corrected for these distortions. In Chapters 2 and 3 we discuss the problem of alignment and present two methods that are based on correction of the projection matrix by estimation of alignment parameters.

### 1.3.2 Low-dose and limited data

In many tomography applications it is desirable to reduce the scanning time as much as possible, to limit radiation dose or to increase throughput. A reduction of the scanning time will result in a reduction of the number of projections, the signal-to-noise ratio of the projections, or both. Reducing the number of projections reduces the number of equations in Eq. (1.1), which results in a larger nullspace of the projection matrix $\boldsymbol{W}$. Therefore, obtaining an accurate solution (which is close to the ground truth image) becomes more difficult. If the projection images are noisy, Eq. (1.1) is typically not consistent, due to the noise term $\boldsymbol{\epsilon}$ in Eq. (1.2). The effect of noise and a low number of projections on the reconstruction is illustrated in Fig. 1.5.

The linear system (1.1) is often underdetermined. Therefore, the least squares solution is not uniquely determined. Most algebraic reconstruction methods converge to the least squares solution that has smallest $\ell_2$-norm (an example is shown in Fig. 1.5b). However, there is no guarantee that such a reconstruction will be similar to the scanned object, due to the inability to recover nullspace components of the matrix $\boldsymbol{W}$. If the number of projections is particularly small, the nullspace has a large dimension, which further exacerbates this problem. A powerful technique to improve the accuracy of a reconstruction from few projections is by incorporating prior knowledge about the scanned sample. Imposing prior knowledge about certain properties of the sample on the reconstruction reduces the solution space drastically.

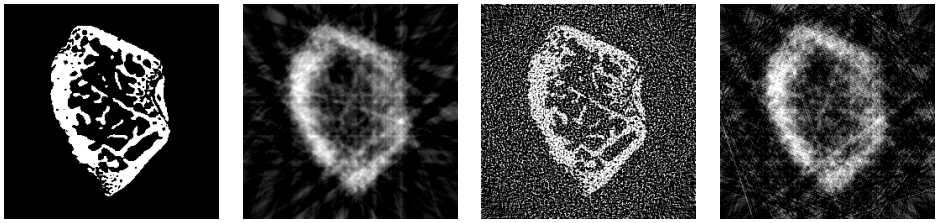
A particular type of such prior knowledge is exploited in *discrete tomography*, where the sample is assumed to consist of only a few different materials, each corresponding to a constant gray value in the reconstruction. Usually this prior is combined with a smoothness prior, where we assume that neighboring pixels in the reconstruction have similar gray values.

The linear system (1.1) is typically inconsistent (has no exact solution), for example due to noise on the projection data, $\boldsymbol{\epsilon} \neq \mathbf{0}$. Therefore, we can obtain an approximate solution of Eq. (1.1) by solving Eq. (1.3), for example by using a least squares method. Not all reconstruction algorithms are equally suitable for dealing with noisy data. The SIRT algorithm is well-known for generating smooth reconstructions from noisy data, whereas the standard Ram–Lak filter in FBP amplifies high frequency noise.

In case we deal with both noise and a small number of projections the reconstruction problem is especially challenging. In Chapter 4 we discuss a reconstruction method that aims to improve the results for discrete tomography from noisy data.

### 1.3.3 Other unmodeled errors

Besides the errors we discussed previously, there are other errors that cause artifacts in the reconstruction. Mainly, these errors are caused by physical effects that were not captured in the linear model of Eq. (1.1). For example, there can be nonlinear effects in the camera optics that record projection images. Or the direction of the rays might not be along straight lines, for example due



(a) original (b) few projections (c) noise (d) few noisy projections

Figure 1.5: The effect of a low number of projections and noise on the reconstruction. (a) is the ground truth image, a rat femur bone; (b) – (d) are reconstructions.

to scattering. An example of structured scattering can be observed if there is crystalline structure in the sample, which results in diffraction patterns on the detector. Often it is very difficult to capture these phenomena in a system of equations that can still be solved effectively.

If an algebraic reconstruction method is applied to these kinds of datasets, the nonlinearities are not captured by Eq. (1.1) and artifacts are produced. In Chapter 6 we describe a method that can be used if the nonlinear effects are present in only parts of the projection data. In that case we can still use algebraic reconstruction techniques and ignore the nonlinear effects by using a suitable penalty function $\rho(\cdot)$ in Eq. (1.3).

## 1.4 Overview

In this section we will give a brief overview of the main results of this thesis.

In Chapter 2 we introduce a method for automatic optimization of alignment parameters. We pose the problem of estimating geometrical perturbations as a nonlinear least squares problem, with the objective to optimize consistency of Eq. (1.1). We employ the Levenberg–Marquardt optimization method to compute a 2D reconstruction and simultaneously find these alignment parameters.

In discrete tomography an image is reconstructed that consists of a few gray values corresponding to the materials in the observed object. Constraining the feasible set of gray values in the reconstruction is a very powerful regularization that is effective even if the number of projections is small. In Chapter 3 we present an alignment method which can be used for datasets with few projections of 3D binary volumes (*two gray level* images). The alignment method is a variant of the method proposed in Chapter 2. In contrast, the reconstruction and the alignment problem are not solved simultaneously. In an iterative process, a reconstruction step using a discrete (binary) tomography algorithm is followed by an alignment phase using Levenberg–Marquardt. This process iteratively refines both the reconstruction and a subset of the alignment parameters in 3D (working in the plane of the detector).

One example of a discrete tomography algorithm is DART (Discrete Algebraic Reconstruction Technique). This algorithm includes a segmentation step to force pixels to the set of a priori known gray values and uses a boundary update step to improve the reconstruction [BS11]. However, DART is not as accurate if the signal-to-noise ratio of the projection data is very low, because the boundary update step is hampered by the noise. In Chapter 4 we demonstrate a variant of DART, called Soft DART which uses a set of soft constraints on the reconstruction which improves the reconstruction quality if the signal-to-noise ratio of the projections is very low.

In Chapter 5 we present an algorithm for removal of a particular class of artifacts, caused by offsets or scaling of the projection data. We assume that for each projection an offset is added to each pixel in that projection image, or that the gray values are scaled. This can be caused by variations of the source intensity, or by image post-processing after the projections have been acquired. We present three different algorithms: one to remove a *global offset*, where the offset is the same for each projection image, removal of a *local offset* where the offset is different for each projection image and a scale retrieval algorithm, to estimate relative scale factors between projection images.

Algebraic reconstruction methods optimize consistency of the reconstruction with the observed projections in the $\ell_2$-norm. The underlying assumption of this approach is that the noise in the projection data has a Gaussian distribution. However, in cases where large outliers are present in the projection data, the equation system (1.1) is not consistent and the reconstruction will be fitted to these outliers, resulting in artifacts in the reconstruction. In Chapter 6 we use a penalty function for the residual that is based on the maximum likelihood estimate from the Student's t distribution, which assigns a smaller penalty to outliers. As a result the effect of outliers is strongly reduced which improves the reconstruction quality. No preprocessing is required to locate the outliers. We demonstrate the effectiveness of this approach on a 3D cone-beam simulated dataset for a series of perturbations in the projection data.

In Chapter 7 we introduce a Matlab interface for the ASTRA toolbox based on the Spot toolbox. The ASTRA toolbox is a software toolbox for tomographic reconstruction that provides reconstruction algorithms, as well as building blocks for creating new algorithms. These building blocks consist mainly of the GPU (and CPU) accelerated *forward* and *backprojection* operations. These operations correspond to matrix products of $\boldsymbol{W}$ and $\boldsymbol{W}^\intercal$ respectively, as used in Eq. (1.1). The proposed interface exposes these operations to Matlab as matrix-like operators, which enables the use of standard Matlab code to employ the GPU back end of the ASTRA toolbox. We have used this interface extensively to rapidly implement and develop the algorithms presented in this thesis.

# Chapter 2

# Automatic optimization of alignment parameters for tomography datasets

## 2.1 Introduction

Tomography deals with the problem of reconstructing an object from projections [KS01]. Projections are measured by a scanning device at varying orientations with respect to the object. Each projection consists of a series of intensity measurements (*e.g.*, from X-rays) along straight lines, which approximate line integrals of the object density. In the reconstruction problem, an object density function is computed that matches the set of projections as close as possible. The reconstruction problem is an ill-conditioned inverse problem that can be solved using numerical methods.

As the resolution of tomography scanners has increased substantially in recent years, it has become more and more difficult to achieve sufficient mechanical stability, which is needed to keep all projections in perfect *alignment* during the scan. Ideally, all geometrical parameters of the scanning geometry (*i.e.*, source positions, detector positions, beam angles) are known with high accuracy for each scan. In practice, however, various types of distortions can occur (*e.g.*, due to instabilities), causing deviations between the assumed geometrical parameters and the actual geometry.

Tomography has a wide range of applications, ranging from industrial quality control of large objects using X-rays down to imaging of nanomaterials by electron microscopy. In particular at the smallest scales, problems with the alignment of the projection data form a key bottleneck for the quality of the reconstructed

This chapter is based on the publications:
F. Bleichrodt and K. Batenburg. "Automatic optimization of alignment parameters for tomography datasets". In: *Image Analysis*. Vol. 7944. LNCS. Springer, 2013, pp. 489–500
F. Bleichrodt, J. Sijbers, J. de Beenhouwer, and K. J. Batenburg. "An alignment method for fan beam tomography". In: *Tomography of Materials and Structures*. Ghent University press, 2013, pp. 103–106

image. For example, in electron tomography the specimen has to be recentered for each recorded image as the sample stage is not eucentric, causing lateral shifts in the projection images [JS91]. In high-resolution X-ray tomography, the rotation axis may not be perfectly centered at the detector, leading to structured shifts in the projections. In addition, limited accuracy of the rotation stage leads to uncertainties about the exact projection angles. As a result, inconsistencies are present in the system of equations governing the reconstruction problem. These inconsistencies must be resolved to obtain accurate reconstructions.

We remark that the alignment problem for tomography is fundamentally different from some other problems also named "alignment" in the image processing literature [Sze06; VW97; ZF03]. Compared to, for example, the alignment of photographs in a stitching problem [Sze06], the key difference is that for tomographic alignment, the 3D object itself is related to the (unaligned) projections by a complex inverse problem. Therefore, projections from different angles can often not be directly compared and can only be related to each other by solving this inverse problem. This also makes it impossible to use image registration methods [ZF03] for the type of alignment we consider.

A range of tomographic alignment algorithms have been proposed, which can generally be divided in two classes: methods using fiducial markers and methods based on automatic, markerless alignment. Marker based alignment is often applied in electron tomography for biological samples [Fra92]. Small, dense particles are distributed among the sample, which can be tracked accurately in consecutive projections. A system of equations, relating the marker positions in the projection domain and their position in the sample, can be solved to compute the alignment parameters with a high degree of accuracy. The method requires a long preparation time and the use of markers can result in artifacts in the reconstructed image. Instead of fiducial markers, features in the projection data can also act as markers, [BHE01].

For algorithms that do not use markers, a well known approach is cross-correlation [Die+92; Fit+99]. Here it is assumed that consecutive projections are similar and differ in a smooth way, thereby making strong assumptions about the unknown object. By finding the maximum cross-correlation between successive projections, it is possible to make a rough estimation of the alignment parameters that can be described as an affine transformation of the projections. The main problem of this method is its low accuracy.

Other markerless methods are based on minimizing the inconsistencies between the forward projections of the reconstructed image and the measured projections. These methods, called *projection distance minimization* methods henceforth, are a more general approach to the alignment problem. See for example [HB11; Kym+03; Par+12; YNP05]. Other methods focus on an error measure based on the reconstruction [KSZ11], or use passive auto-focus [Kin+11].

In this chapter, a new markerless alignment method based on projection distance minimization is presented. We propose the Levenberg–Marquardt Projection Distance Minimization algorithm (LMPDM). Similar to the algorithm proposed in [YNP05], the alignment and reconstruction problem are solved simultaneously. The objective of combined alignment and reconstruction is posed as a nonlinear

least squares optimization problem and a numerical method is employed for solving it.

Instead of the Quasi-Newton BFGS method used in [YNP05], we choose Levenberg–Marquardt (LM), which has been shown to yield better convergence for certain least squares problems, as discussed in chapter 10 of [NW06]. When implementing a numerical scheme for this nonlinear least squares problem, several design choices must be made, with respect to computation of numerical derivatives and image resolution. We demonstrate that these design choices are crucial to the success of the LM algorithm in recovering the alignment parameters.

Our experimental results, based on simulated projection data, show that if a multi-resolution scheme is combined with local smoothing of the Jacobian, our LMPDM algorithm is capable of recovering the alignment parameters with high accuracy. Also, the underlying tomography software library is implemented on the GPU, which makes the algorithm scalable.

This chapter is structured as follows. Mathematical background and implementation details are discussed in Section 2.2. In Section 2.3 and Section 2.4, a series of experiments is described and the results are presented. Section 2.5 contains a discussion of the results. Section 2.6 concludes this chapter.

## 2.2 Methods and implementation

This section will formulate the alignment problem in a mathematical context and introduce the notation. Subsequently, the LMPDM method and its implementation details will be discussed. Furthermore, design choices are explained that improve the accuracy of the alignment algorithm.

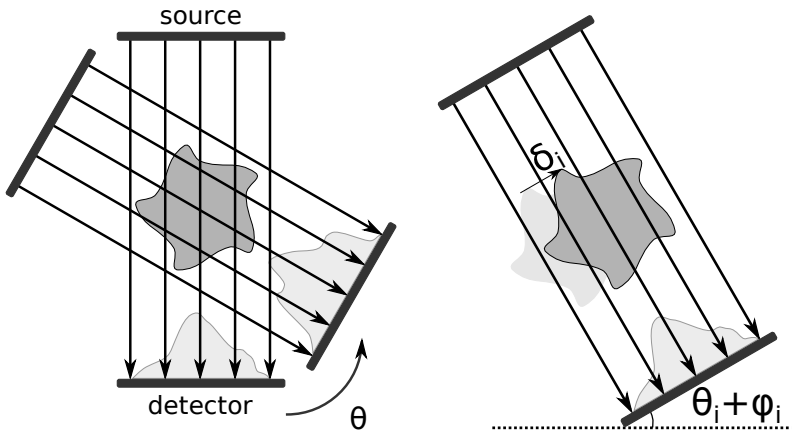
### 2.2.1 Model and notation

The object from which the projections are acquired can be modeled by a gray value image $f : \mathbb{R}^2 \to \mathbb{R}$. A projection at angle $\theta$ is the collection of line integrals over the lines $l_{\theta,t} = \{(x, y) : x \cos\theta + y \sin\theta = t\}$ for detector positions $t \in \mathscr{T} \subset \mathbb{R}$, where $\mathscr{T}$ denotes the discrete set of detector positions, see Fig. 2.1. The geometry we consider here is called the *parallel beam* geometry (because the beams are parallel). Later on we also consider the *fan beam* geometry, where the beams originate from a single point source, see Fig. 2.2.

The relation between the object and its projections $P(\theta, t)$ is given by the Radon transform

$$P(\theta, t) = \mathscr{R}[f](\theta, t) \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y)\bar{\delta}(x \cos\theta + y \sin\theta - t)\,\mathrm{d}x\,\mathrm{d}y, \tag{2.1}$$

with $\bar{\delta}$ the Dirac delta function. By discretizing the image $f$, the set of angles, and the set of detector positions, and numerically approximating the Radon transform we arrive at the algebraic representation of the tomography problem. In this form




(a) (b)

Figure 2.1: Parallel beam geometry for the two dimensional case: (a) A tomographic scan: the dark gray region represents the object along with its projection below. The detector-source pair rotates around the object; (b) Projection acquisition at angle $\theta_i$ with angular offset $\phi_i$. The object has a shift of $\delta_i$ in the detector plane with respect to its assumed position.

the object and its projections are related by a linear operator

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p}, \tag{2.2}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ represents the unknown object, $\boldsymbol{W} \in \mathbb{R}^{M\times N}$ is the projection operator and $\boldsymbol{p} \in \mathbb{R}^M$ is the measured set of projections [KS01]. From this point on, we focus on the reconstruction of a single slice of the object, *i.e.*, a 2D image from a set of 1D projections. The object is represented as a two dimensional pixel grid with $N$ pixels. Let $K$ be the number of projections of the object that have been acquired by a detector having $D$ discrete elements. The total number of line projections is then given by $M \equiv KD$. The projection operator is a sparse matrix with $w_{ij}$ modeling the contribution of pixel $j$ to the projection value measured by detector $i$. So the inner product of row $i$ of $\boldsymbol{W}$ and the object $\boldsymbol{x}$ gives a discrete approximation of the line integral over a line perpendicular to detector $i$.

Projections of the object are recorded at a discrete set of angles

$$\begin{aligned}\boldsymbol{\theta} &= \theta_1, \ldots, \theta_K,\\ 0 \le \theta_1 &< \cdots < \theta_K \le \pi.\end{aligned}$$

Up until now, we have assumed that the measurements correspond perfectly with the Radon transform. In practice, each of the projections have a perturbation in the angles as well as the object position. These are represented in the alignment parameters

$$\begin{aligned}\boldsymbol{\delta} &= \delta_1, \ldots, \delta_K,\\ \boldsymbol{\phi} &= \phi_1, \ldots, \phi_K.\end{aligned} \tag{2.3}$$

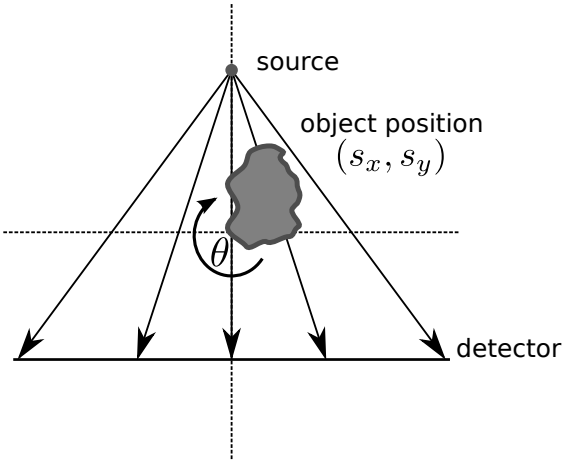



Figure 2.2: Geometrical parameters for the fan beam geometry for a flat detecor. The position of the object is denoted by $(s_x, s_y)$.

as illustrated in Fig. 2.1b. Accordingly, in the continuous case, a single line projection at angle $\theta_i$ and detector offset $t$ is represented by the Radon transform including the alignment parameters:

$$\mathscr{R}[f](\theta_i + \phi_i, t + \delta_i) = \int_{-\infty}^{+\infty} \int_{-\infty}^{\infty} f(x,y)\bar{\delta}(x\cos(\theta_i + \phi_i) + y\sin(\theta_i + \phi_i) - (t + \delta_i))\,\mathrm{d}x\,\mathrm{d}y.$$

In the discrete model, the coefficients in our projection operator depend on the geometry,

$$\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\delta})\boldsymbol{x} = \boldsymbol{p}. \tag{2.4}$$

This expression for $\boldsymbol{W}$ is not easily available in closed form, but it can be evaluated numerically. Note that the projection angles $\boldsymbol{\theta}$ are known, while the perturbations in the projection angles, denoted by $\boldsymbol{\phi}$, and the detector shifts $\boldsymbol{\delta}$ are unknown.

In the experiments of Section 2.3 we also consider the *fan beam* geometry, which is illustrated in Fig. 2.2. For this geometry a shift of the object in the direction of the source causes a magnification of the projections, which is not the case for the parallel beam geometry. Therefore, in this geometry we have introduced the parameters $s_x$ and $s_y$ which indicate the position of the object with respect to the origin. We assume that the distance between the source and the detector is fixed. Note that the object position is different for every projection angle, leading to the parameter vectors:

$$\boldsymbol{s}_x = s_x^{(1)}, \ldots, s_x^{(K)},$$
$$\boldsymbol{s}_y = s_y^{(1)}, \ldots, s_y^{(K)},$$

where $K$ is the number of projection angles. This leads to the equation:

$$\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{s}_x, \boldsymbol{s}_y)\boldsymbol{x} = \boldsymbol{p}, \tag{2.5}$$

where $\boldsymbol{\theta}$ is known and the perturbations in the projection angles $\boldsymbol{\phi}$ and object positions $\boldsymbol{s}_x$ and $\boldsymbol{s}_y$ are unknown. In the rest of this section we focus on the parallel beam geometry, but the same derivation and alignment method are used for the fan beam geometry.

In an experimental setup, the projections contain noise and the perturbations of the geometrical parameters are not known. Therefore, the system in Eq. (2.4) is inconsistent. Alignment involves estimating the unknown alignment parameters in Eq. (2.3). Minimizing the least squares residual of Eq. (2.4) seems to be a good approach, because in the absence of noise and when the alignment parameters are known exactly, then Eq. (2.4) is consistent.

Now we can define the objective of combined alignment and reconstruction as a minimization problem of the *projection distance*, defined by the following cost function

$$\min_{\boldsymbol{x},\boldsymbol{\phi},\boldsymbol{\delta}} \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\delta}, \boldsymbol{x})\|_2^2 := \min_{\boldsymbol{x},\boldsymbol{\phi},\boldsymbol{\delta}} \frac{1}{2}\|\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\delta})\boldsymbol{x} - \boldsymbol{p}\|_2^2, \tag{2.6}$$

with $\boldsymbol{r}$ the residual, and similarly for the fan beam geometry:

$$\min_{\boldsymbol{x},\boldsymbol{\phi},\boldsymbol{s}_x,\boldsymbol{s}_y} \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{s}_x, \boldsymbol{s}_y)\|_2^2. \tag{2.7}$$

The $\ell_2$-norm is chosen because it allows us to use least squares solvers and it has some nice properties, due to its simplicity. Alternative distance or similarity measures such as mutual information can be employed here and might give satisfying results as well.

In Eq. (2.6), the minimization with respect to $\boldsymbol{x}$ is a linear inverse problem that yields a reconstructed image. The minimization with respect to $\boldsymbol{\delta}$ and $\boldsymbol{\phi}$ can be seen as a nonlinear model fitting problem. The combination in the full cost function is, hence, a nonlinear least squares problem.

Projection matching algorithms such as [Par+12], consider the same cost function as in Eq. (2.6), however, an alternating approach is employed. Those methods repeatedly alternate between minimizing Eq. (2.6) with respect to the gray values $\boldsymbol{x}$ (and keeping the alignment parameters fixed) and minimization of Eq. (2.6) with respect to the alignment parameters $\boldsymbol{\phi}$ and $\boldsymbol{\delta}$ (keeping $\boldsymbol{x}$ fixed). Such methods are heuristic in nature and it is not guaranteed that this approach converges to a local minimum. This is why we chose to minimize over the full set of variables at the same time.

The cost function seems suitable to solve by using one of the standard algorithms from numerical optimization. A method specifically aimed at these kinds of problems is the Newton-type algorithm Levenberg–Marquardt. However, due to numerical problems, a straightforward implementation often does not yield an accurate alignment. In the following sections, we will demonstrate that problem-specific design choices in the implementation are essential for accurate parameter estimation.

### 2.2.2 Levenberg–Marquardt

Levenberg–Marquardt, see chapter 10 of [NW06], is an iterative method that generates a sequence of input vectors $\{\boldsymbol{y}_k\} = \{(\boldsymbol{x}^{(k)}, \boldsymbol{\phi}^{(k)}, \boldsymbol{\delta}^{(k)})\}$ that have monotonically decreasing cost function values. Each iteration has the basic form

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \boldsymbol{\eta}_k, \tag{2.8}$$

where the descent direction $\boldsymbol{\eta}_k$ is found by minimizing a quadratic model of the objective function using gradient information:

$$\min_{\boldsymbol{\eta}_k} \left\| \begin{bmatrix} \boldsymbol{J}_k \\ \sqrt{\lambda_k}\boldsymbol{I} \end{bmatrix} \boldsymbol{\eta}_k + \begin{bmatrix} \boldsymbol{r}_k \\ \boldsymbol{0} \end{bmatrix} \right\|_2^2 \tag{2.9}$$

with $\boldsymbol{J}_k$ the Jacobian of the residual $\boldsymbol{r}_k$ and $\lambda_k$ a regularization parameter. This parameter limits the norm of the search direction and acts as a trust-region. It is adjusted based on the accuracy of the quadratic model.

The linear least squares problem in Eq. (2.9) can be solved using one of the many available least squares solvers.

### 2.2.3 Computing the Jacobian

For computing the Jacobian of the residual we use a combination of an analytical expression and a numerical approximation. With respect to the image $\boldsymbol{x}$ the Jacobian is given by $\boldsymbol{J}_{\boldsymbol{x}} = \boldsymbol{W}$, but for the derivative with respect to the parameters $\boldsymbol{\delta}$ and $\boldsymbol{\phi}$ we do not have such an expression. Therefore we approximate the gradients in the Jacobian by a central finite differences scheme:

$$\nabla_{\delta_i} \boldsymbol{r}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\delta}, \boldsymbol{x}) = \frac{\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\delta} + h\hat{\boldsymbol{e}}_i)\boldsymbol{x} - \boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\delta} - h\hat{\boldsymbol{e}}_i)\boldsymbol{x}}{2h}, \tag{2.10}$$

where $\hat{\boldsymbol{e}}_i$ is the $i$th basis vector. A similar expression is used for $\boldsymbol{\phi}$.

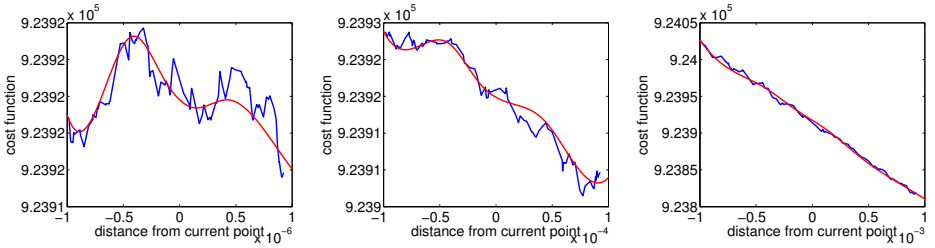
As illustrated in Fig. 2.3, our GPU-implementation of the cost function in Eq. (2.6) shows irregularities at small scales. These are introduced by the discretization of the problem domain, by floating-point errors involved in computing the cost function, and by noise in the projection data. This behavior makes the accuracy of the numerical Jacobian in Eq. (2.10) highly dependent on the step size $h$. Therefore, a robust method for choosing a good step size $h$ is needed.

Methods proposed in literature for computing numerical derivatives on discrete, noisy data are not feasible in our implementation, due to their computational intensity [Cha11; HL82]. As an alternative, we propose the following method. We sample the cost function in the direction of $\boldsymbol{\phi}$ (and similarly for $\boldsymbol{\delta}$):

$$s_\phi(\alpha) = \frac{1}{2}\|\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\phi} + \alpha\boldsymbol{1}, \boldsymbol{\delta})\boldsymbol{x} - \boldsymbol{p}\|_2^2, \tag{2.11}$$

at the equidistant points

$$\alpha = -8h, -7h, \ldots, 7h, 8h.$$




(a) $h = 10^{-6}$ (b) $h = 10^{-4}$ (c) $h = 10^{-3}$

Figure 2.3: Comparison of the cost function at different scales. The plots show the cost function in the range $\frac{1}{2}\|\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\delta} \pm h\mathbf{1})\boldsymbol{x} - \boldsymbol{p}\|_2^2$, together with its approximating spline. The irregular behavior starts to disappear for $h = 10^{-3}$.

Here $\mathbf{1}$ is a vector of which each element is 1. The sample points with odd indices are used to generate a spline. If the cost function is smooth at the current scale $h$, we can assume that the spline is a close approximation to the cost function. As an error measure for this we compute the difference between the sample points with even indices and the generated spline and normalize to yield a relative error. By computing this error for several scales $h$, we can select the scale for which the error is minimal. The cost function at this scale does not show irregularities due to the discretization. This $h$ is then used in Eq. (2.10) as step size. Fig. 2.3 illustrates this method.

Sampling these values to compute a step size $h$ is costly, hence the step sizes are computed once at the beginning of the algorithm. It is recomputed only after a transition between resolutions, because our algorithms employs a multi-resolution technique as discussed in the next section.

### 2.2.4 Multi-resolution

One of the main difficulties in applying the alignment algorithm in practice is the computational scale. It is not uncommon to have datasets containing billions of detector values. A conventional approach to reduce the computation time is to apply multi-resolution techniques. We utilize this technique by running the algorithm repeatedly, going from a coarse to a fine representation of the data. The output of one run serves as the input of the next. Low frequency components of the error are removed first at coarse grids. This approach refines the solution by gradually removing higher frequency components of the alignment error.

In our case the domain of the multi-resolution technique is the reconstructed image and a sinogram (set of projections). We have chosen to match the pixel size of the image with the size of a detector element. This makes the implementation easier, since the sinogram and reconstructed image can simply be resized when going from coarse to fine representation.

Lowering the resolution makes the images smoother, hence multi-resolution acts as a regularization of the optimization problem in Eq. (2.6). For example, the detector shift is measured in the number of detector elements. So on a coarse

grid, the detector shift is reduced by the same factor by which the grid has been resized. Essentially, the initial values become closer to the optimal values. This makes it more likely to find the global minimum and possibly skip local minima. The effect of applying multi-resolution is shown in Section 2.4.

## 2.3 Experiments

A series of simulation experiments was carried out to evaluate the capabilities of the LMPDM algorithm. In the simulations we used the following hardware: a workstation with an Intel Core i7-2600K CPU@3.40 GHz combined with a Geforce GTX 570 GPU. For the forward and backprojection operations, a GPU implementation was used.

First, we have applied LMPDM to three parallel beam simulated datasets based on phantom images shown in the left column of Fig. 2.4. The datasets consist of projections at 100 angles, which were generated from the phantom images. The equidistant angles are in the range $[0, \pi)$ and random, uniformly distributed offsets $\phi_i \in [-0.9°, 0.9°]$ were added. The error in the angles is at most $\pm 0.9°$, such that the ordering of the angles is preserved. Also, for each angle a uniform random shift $\delta_i \in [-10, 10]$ was applied. The maximum shift of 10 detector pixels is approximately 5 percent of the image size, which is $256 \times 256$. The detector has 256 detector elements per projection. Poisson noise was applied to the projections using a photon count of $10^5$, to simulate moderate experimental noise. The projection matrix $\boldsymbol{W}$ is computed by the method of Joseph [Jos82], using a GPU implementation.

The final dataset is for a fan beam geometry, from the phantom image shown in Fig. 2.7a. The phantom image is based on a reconstructed slice of size $512 \times 512$ from a 3D experimental dataset of a metal foam. A total of 120 equiangular projections were simulated using fan beam geometry where the distance between the detector and source was eight times the image width (which corresponds to a total of 4096 pixels). This results in a fan angle of 14.25°. The detector width was 1024 pixels, which is wider than the ground truth image to deal with the magnification from the fan beam. For this experiment we randomly sample object displacements $(\boldsymbol{s}_x, \boldsymbol{s}_y)$ uniformly from the interval $[-10, 10] \times [-10, 10]$ (in units of detector pixels) and the angular offsets are randomly, uniformly sampled from $[-0.1°, 0.1°]$. This corresponds to object motion of approximately 2% of the object size. We applied Poisson noise to the projection data. The noise level is based on the simulated photon count (in this case $10^5$) used for acquiring the projections. After the alignment a SIRT reconstruction with nonnegativity constraints was applied using the geometrical parameters obtained by the optimization routine.

The method we employ for solving the quadratic model in Eq. (2.9) is LSMR [FS11]. As a stop criterion for LMPDM, the change in parameters relative to their norm is monitored. If this falls below a certain threshold, the algorithm stops. The same holds for the norm of the gradient $\|\boldsymbol{J}^\intercal \boldsymbol{r}\|$ of the cost function in Eq. (2.6). In any case the algorithm transitions to a higher resolution, or is terminated, when a total of 100 LM-iterations is reached.

For comparison, we have also employed a cross-correlation algorithm for the parallel beam datasets. This method estimates object shifts by correlating consecutive projections. Cross-correlation on two discrete real signals $f$ and $g$ is defined as:

$$(f \star g)(i) := \sum_{j=1}^{L} f(j)g(i+j) \tag{2.12}$$

where $L$ is the length of the reference signal $f$. Usually zero-padding of $g$ is needed. The cross-correlation attains its maximum value when the two signals align, or match as closely as possible. The corresponding $i$ gives us the shift between the signals. To allow sub-pixel precision in the alignment, prior to the cross-correlation, the projections were upsampled by a factor of ten.

A region of the first projection, that is in view for all projections, acts as reference. To this, the second projection is correlated, estimating the relative shift. Then the second projection acts as reference to which the third projection is aligned and so on. Note that we assume here, that the first projection is perfectly aligned. If this was not the case, the projections are shifted away from the center of rotation, which still produces alignment artifacts.

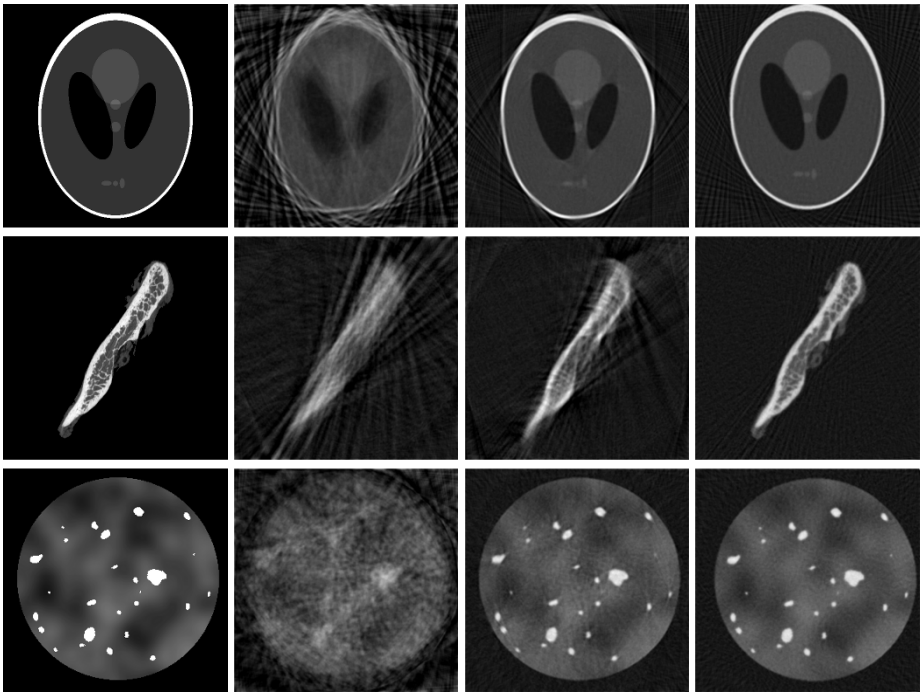
## 2.4 Results

The qualitative results are given in Fig. 2.4. Column 2 shows the unaligned reconstructions, where 300 iterations of the algebraic reconstruction method SIRT were performed [KS01]. These show the impact of small perturbations in the geometry. Details are blurred and the background is filled with stripes.

The third column shows SIRT reconstructions using the alignment parameters found by the cross-correlation method. Since this method cannot retrieve angular offsets, the resulting artifacts are still visible. For the mandible bone dataset, cross-correlation clearly fails. Due to the fact that the sample is flat, projections from different angles have very different width. Therefore, without stretching of the projections, their correlation is rather limited. Many streaks inside the objects remain. This is because the shift parameters are not found accurately. If we look at the difference between the found alignment parameters and their true values for the Shepp-Logan phantom in Fig. 2.6, it is clear that cross-correlation does not yield sub-pixel accuracy. The LMPDM method, however, achieves an accuracy of approximately one tenth of the pixel size.

The alignment results of our method LMPDM are given in the last column of Fig. 2.4. Here, the details are much clearer and the streaks are almost gone. Overall, the reconstructions are lacking some sharpness. Note that in the LMPDM aligned Shepp-Logan image, a shift has occurred with respect to the phantom image. This is because the alignment parameters are invariant to a global shift or rotation of the object. In our error measurements, this global shift and rotation have been removed first.

In Fig. 2.5, the convergence history is shown. The curves show step-wise convergence behavior. This is the result of the multi-resolution approach. At some

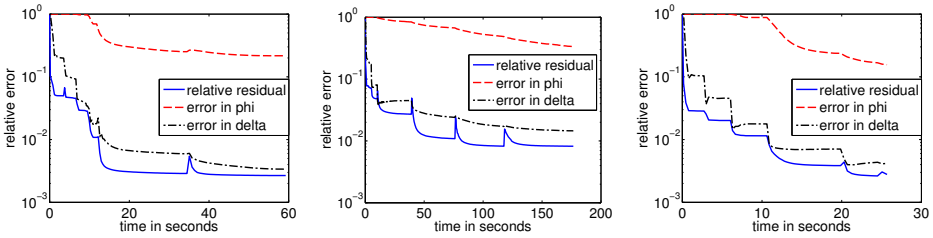


(a) phantom images (b) unaligned (c) cross-correlation (d) LMPDM

Figure 2.4: Overview of the results: (a) phantom images of size $256 \times 256$, the Shepp-Logan head phantom, a mandible bone and particles phantom respectively; (b) the unaligned SIRT reconstructions; (c) SIRT reconstructions aligned by cross-correlation; (d) SIRT reconstructions using alignment parameters found by LMPDM.

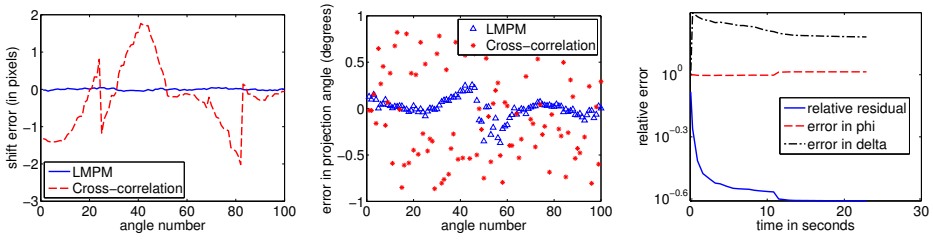
point, the algorithm cannot improve the parameters at the current resolution. Therefore, a transition to a higher resolution occurs. At the higher resolution, finer details can be resolved and the errors can be reduced further. Note that jumps occur in the residual at these resolution transitions. The residual is not invariant with respect to the image size. Therefore, this behavior is expected and does not indicate a convergence problem. For the Shepp-Logan and particle dataset, we see that the error in $\boldsymbol{\phi}$ starts to drop at higher resolutions ($64 \times 64$), while the shifts are refined at all resolution. The reason for this is that the alignment of the projection angles requires details to be present in the reconstruction. The shifts however can align quite well to a low quality image.

The importance of the multi-resolution approach combined with an automatically selected step size in Eq. (2.10) becomes apparent when the Levenberg–Marquardt routine is used on a single resolution, with fixed step sizes of $h = 10^{-6}$ in Eq. (2.10). These step sizes have an order of magnitude that is generally considered to give accurate finite differences. The step sizes produced by our




(a) Shepp-Logan (b) mandible (c) particles

Figure 2.5: Convergence history of the simulations using multi-resolution and automatically selected step size. The horizontal axis shows the wall clock time. On the vertical axis, the relative residuals are shown $\|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{p}\|_2/\|\boldsymbol{p}\|_2$, $\|\boldsymbol{\phi} - \boldsymbol{\phi}_{\text{true}}\|_2/\|\boldsymbol{\phi}_{\text{true}}\|_2$ and $\|\boldsymbol{\delta} - \boldsymbol{\delta}_{\text{true}}\|_2/\|\boldsymbol{\delta}_{\text{true}}\|_2$.




(a) Error in shift (b) Error in angles (c) Single resolution run

Figure 2.6: These plots show the difference between the found alignment parameters and their true values for the Shepp-Logan dataset. Since cross-correlation methods cannot find perturbations in angles, the stars in the scatter plot in (b) are the initial values of $\boldsymbol{\phi}$. In (c), convergence is shown for LMPDM without using multi-resolution and with step sizes of $10^{-6}$ for computig the Jacobian with respect to both $\boldsymbol{\delta}$ and $\boldsymbol{\phi}$.

spline method are in the order of $h_\delta = 1$ and $h_\phi = 0.1$. The results in Fig. 2.6c point out that the alignment parameters are not found and that the error even increases. This shows that the proposed methods for multi-resolution and local smoothing of the Jacobian are essential to achieve high accuracy.

Finally we look at the results of the fan beam geometry dataset, which is shown in Fig. 2.7. The ground truth is given in Fig. 2.7a. In the initial unaligned SIRT reconstruction Fig. 2.7b, all details are missing and the positions of the cavities in the metal foam cannot be accurately determined. However, in the aligned reconstruction Fig. 2.7c, all large cavities are visible, albeit that some smaller ones are still missing. Overall the sharpness of the reconstruction is reduced.

The results demonstrate that without alignment, qualitative or quantitative analysis of reconstructions can be very limited. A good alignment routine can improve the quality substantially. In Fig. 2.8 the convergence is shown for the alignment algorithm (squares) as well as LM applied to the reduced problem of

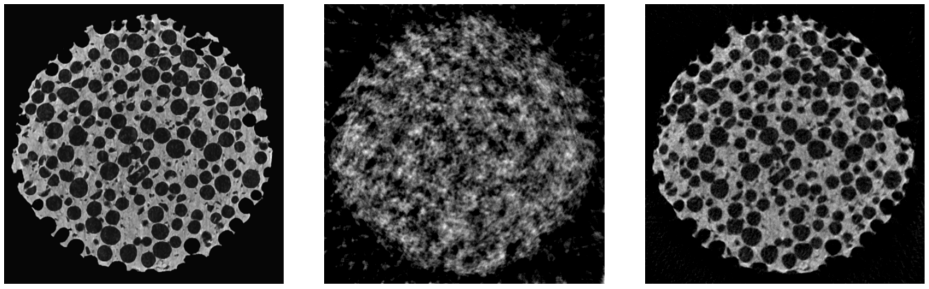


Figure 2.7: Simulation results of the fan beam alignment algorithm, (a) The metal foam phantom; (b) An unaligned SIRT reconstruction; (c) A SIRT reconstruction after alignment.

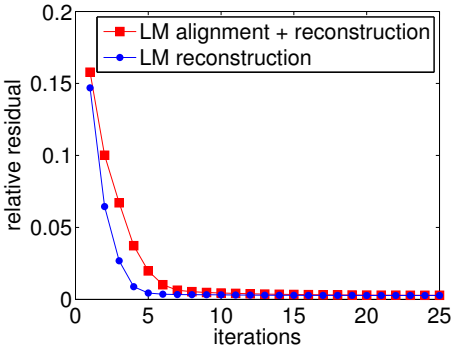



Figure 2.8: Convergence of the fan beam alignment compared to a reconstruction with LM applied on an aligned dataset.

minimizing Eq. (2.7) over $\boldsymbol{x}$ alone with the alignment parameters fixed to their true values. Both simulations converge to the same residual, suggesting that the alignment parameters cannot be improved further in terms of the residual error. This shows that, even with small amounts of noise in the projection data, a good alignment is possible.

## 2.5 Discussion

From the results we can see that our proposed method, LMPDM, performs well on the selected phantom data. However, a straightforward, naive implementation of Levenberg–Marquardt is bound to fail. The reason for this is the irregular behavior of the objective function Eq. (2.6) due to the single precision code. The methods we have introduced, an automatically selected step size combined with a multi-resolution technique, are sufficient to solve this problem. On the one hand, the improved accuracy of the Jacobian yields more accurate descent directions

for LM, which improves convergence. Moreover, multi-resolution helps to find a minimum of Eq. (2.6) even if the perturbations in the parameters are large. Most projection matching alignment algorithms like [Par+12] require an initial coarse alignment if this is the case.

The results from the multi-resolution LMPDM in Fig. 2.5 show, that at low resolutions ($16 \times 16$, $32 \times 32$), the errors in the shifts decrease rapidly and only a few iterations are needed for convergence. This suggests that there might exist a more optimal selection of the resolutions and their order. Perhaps a multigrid method with an efficient intergrid transfer operator could improve performance.

The numerical results from the fan beam dataset in Fig. 2.7 suggest that alignment of fan beam projection data can be performed accurately by the proposed method, even in presence of noise.

The run times we have measured, in the order of a minute, show that LMPDM is an efficient method, suitable for experimental datasets for the reconstruction of 2D slices.

## 2.6 Conclusions

A new markerless alignment algorithm based on projection matching has been proposed. Using a robust technique to compute the Jacobian combined with a multi-resolution scheme, the accuracy of the LM optimization algorithm can be improved substantially. The resulting LMPDM algorithm performs well even if the perturbations in the alignment parameters are large. The timing results show that the method is efficient enough to be used on 2D experimental datasets. For future research, it is interesting to generalize the algorithm to 3D, which adds a challenge in computational scale, as well as the added complexity of the geometrical parameters. Also, one can experiment with extra terms in the cost function in Eq. (2.6), such as prior knowledge, or use other distance measures.

# Chapter 3

# Aligning projection images from binary volumes

## 3.1 Introduction

Tomography deals with the reconstruction of an object based on projections; see Fig. 3.1. Projection images are acquired by scanning devices, such as X-ray based medical scanners or transmission electron microscopes [BF11; MD09]. For high resolution microtomography or nanoscale imaging, the stability of the scanner hardware is a limiting factor in the reconstruction quality [MW03]. Motion of the object or limited accuracy of the mechanics leads to unaligned projection images that produce *alignment* artifacts in the reconstruction. Algorithms for aligning the projection images are essential to fully exploit modern detectors with high pixel density.

In electron tomography, the projection images are created using a beam of electrons. The instability of the sample holder in the electron microscope and technical limitations can lead to severe distortions in the geometry. Especially in the position of the object [HB11].

Another area in which alignment is important is in a synchrotron setup. A synchrotron produces monochromatic X-rays using a particle accelerator. This results in high resolution projection images that do not suffer from beam hardening. However, the instruments used in the projection acquisition are often not completely aligned before starting the experiment, resulting in the need for post-acquisition alignment. Also, during in-situ experiments it is challenging to keep the scanner setup in perfect alignment [Wil+11].

Current alignment methods are based on tracking of fixated markers, or are purely data-driven using a (markerless) projection dataset of the object [BHE01;

This chapter has been published with minor modifications as:
F. Bleichrodt, J. De Beenhouwer, J. Sijbers, and K. J. Batenburg. “Aligning projection images from binary volumes”. In: *Fundamenta Informaticae* 135(1) (2014), pp. 21–42

Fit+99; Fra92]. The latter *markerless* methods are often used in applications at nano scales, or other domains where using markers is not possible or not feasible. In many cases, a variant of cross-correlation techniques is employed. The cross-correlation between consecutive images with low angular separation can be exploited to estimate in-plane transformations of the projection images. In *projection matching* methods, an intermediate reconstruction is formed. By generating forward projections of the reconstructions and comparing them with the observed projections, it is possible to refine the alignment [Par+12]. However, to obtain a reasonably accurate reconstruction, a relatively large number of projection images is required.

In recent years, a substantial number of publications have appeared about image reconstruction from highly limited data, in the fields of discrete tomography (DT) and compressive sensing (CS) [Bat05; BS11; CRT06; Sch+05]. In discrete tomography, the unknown object is known to consist of just a few materials. Therefore, the number of gray values corresponding to these materials is small and possibly known in advance. This information can be used as prior knowledge in the reconstruction algorithm, to limit the solution space and yield more accurate reconstructions. In compressive sensing, sparseness of the total variation of the object (*i.e.*, the $\ell_1$-norm of the gradient image) can be exploited to reconstruct the object from a few projection images.

In this chapter, we focus on DT, but the same concepts can be applied to CS techniques as well. To apply discrete tomography effectively, an aligned dataset is needed, as well as a good estimation of the gray levels. Methods have been proposed in the literature for gray level estimation [BAS11], but for alignment, using markers is the only option. All of the markerless methods require a substantial number of projection images to perform well:

- for methods that exploit similarity between projections, the angular gap cannot be too large;
- for projection matching, to generate a decent reconstruction, enough projections should be available.

In this chapter, we propose the Discrete Tomography Projection Matching (DTPM) method. DTPM incorporates a discrete tomography reconstruction algorithm in the projection matching alignment. We demonstrate with simulation experiments, that using this prior knowledge in the alignment phase, a substantially better alignment can be obtained. In comparison to standard projection matching methods, the DTPM alignment results in more accurate reconstructions.

The structure of this chapter is as follows: in Section 3.2 the acquisition geometry for parallel beam tomography is described in detail. Here we introduce the geometrical parameters that are estimated by our alignment algorithm. We also give a formal, mathematical introduction to tomographic reconstruction. We show how the geometric parameters are included in the reconstruction and we define alignment by projection matching. A short description of discrete tomography is presented as well. In Section 3.3 we introduce our proposed DTPM alignment method. We show how discrete tomography is incorporated in the

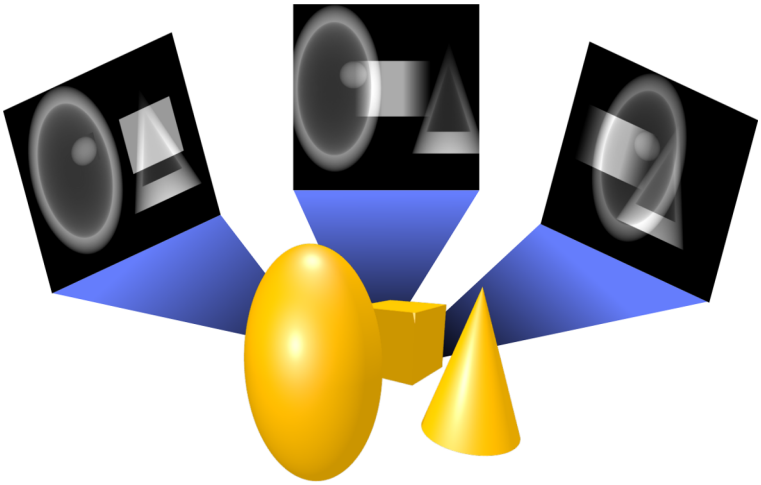


Figure 3.1: Acquisition of projection images from several angles. This figure shows the object with unknown interior and three of its projection images.

projection matching method. In Section 3.4 we describe a series of simulation experiments that were performed, also on noisy data, and the results are discussed. In Section 3.5 we discuss considerations that should be taken into account when applying our algorithm on data from real experiments. Finally, we conclude this chapter in Section 3.6.

## 3.2 Method and implementation

In this section we will explain the geometrical parameters that determine the projection geometry. We focus on the parallel beam geometry, illustrated in Fig. 3.2, but the method can be extended to other geometries as well. Furthermore, the mathematical background and algorithm details are given for the alignment method that estimates these parameters.

### 3.2.1 Geometrical parameters

A typical setup for projection acquisition in tomography is illustrated in Fig. 3.2. We assume that the radiation source emits parallel beams that are perpendicular to the detector plane. The object is positioned at the origin of the $x$-$y$-$z$ coordinate system and the $z$-axis indicates the rotation axis. The projection angle $\theta$ gives the rotation of the object around the $z$-axis. Note that rotation of the source and detector around the object is equivalent to rotation of the object itself (except for the sign). In our alignment algorithm, we do not align the projection angle $\theta$, because we want to consider the same parameters that are typically used in image registration techniques.

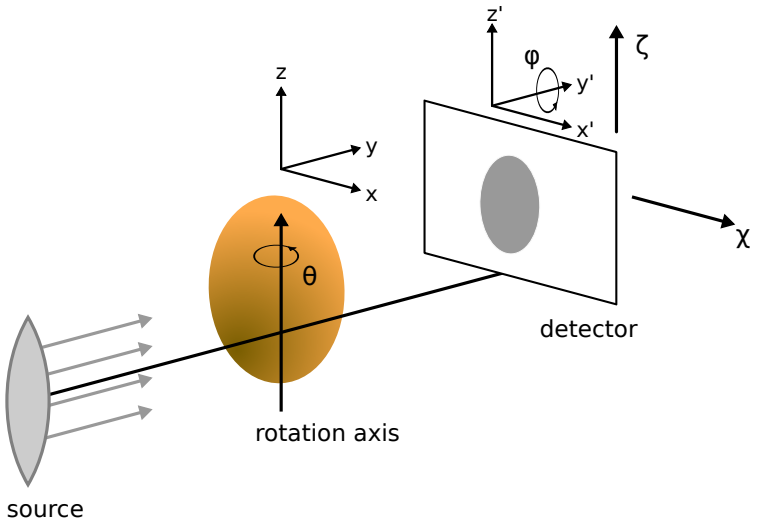



Figure 3.2: Overview of the geometrical parameters describing the geometry of the projection acquisition. The object in the middle is projected onto the detector on the right. The parallel rays are perpendicular to the detector.

To model shift and rotation in the projection domain (*i.e.*, in the plane of the detector, both of the object as well as the detector itself) we define a second coordinate system $x', y', z'$ that initially is aligned with the $x, y, z$ coordinate system. The origin of the $x', y', z'$ coordinates indicates the center of the detector. For the projection angle $\theta = 0$, the horizontal axis of the detector is aligned with the $x'$-axis and its vertical axis with the $z'$-axis. An in-plane shift of the detector (or of the object) can then be modeled by a detector shift of $(\chi, 0, \zeta)$ in the $x', y', z'$ system. Note that the projection images are spanned in the $x'$-$z'$-plane. Therefore, the shift can be considered as a 2D vector in the $x'$-$z'$-plane. Since we consider a parallel beam geometry, object motion in the $y'$-direction does not change projections.

To model in-plane rotation of the detector or object, we introduce the parameter $\phi$. The detector is rotated around the $y'$-axis over an angle $\phi$. The coordinate system of the detector and that of the object are then related by the following coordinate transformations:

$$\begin{aligned} x' &= (x + \chi)\cos\phi + (z + \zeta)\sin\phi, \\ y' &= y, \\ z' &= -(x + \chi)\sin\phi + (z + \zeta)\cos\phi. \end{aligned} \tag{3.1}$$

So the coordinates are coupled by a translation and rotation operator.

Four parameters are sufficient to define the geometry of the acquisition of one projection image. In a tomography scanner, projection images are recorded at multiple angles,

$$\theta_1, \theta_2, \ldots, \theta_K \in V \subset [0, 2\pi). \tag{3.2}$$

In general, most datasets contain projection images with corresponding projection angles in the range $V = [0, \pi)$. For each projection image we have an in-plane shift,

$$(\chi_1, \zeta_1), (\chi_2, \zeta_2), \ldots, (\chi_K, \zeta_K), \tag{3.3}$$

and an in-plane rotation,

$$\phi_1, \phi_2, \ldots, \phi_K. \tag{3.4}$$

In markerless alignment methods based on image registration techniques, it is very common to estimate the parameters $\boldsymbol{\chi}$, $\boldsymbol{\zeta}$ and $\boldsymbol{\phi}$, since these parameters act in the plane of the detector. For this reason, we also consider the parameters $\boldsymbol{\chi}$, $\boldsymbol{\zeta}$ and $\boldsymbol{\phi}$ in our alignment method and do not estimate the other parameters. We note that the approach presented here can be extended to include other alignment parameters as well.

In the case of a perfectly aligned scanner setup that matches our theoretic model of the geometry, we would have $\chi_k = \zeta_k = \phi_k \equiv 0$ and the projection angles are known exactly. In reality, there is a shift in the order of 1 to 10 detector pixels and an in-plane rotation of several degrees. To some extent, the projection angles are known with limited accuracy.

These perturbations in the geometry are mainly caused by

- **Calibration errors** – If the rotation axis and the center of the detector are not aligned, a structural shift or in-plane rotation is introduced. This can be prevented by calibrating the hardware precisely. Calibrating, however, is a time consuming task and difficult if the pixel size of the detector is very small.
- **Mechanical inaccuracies** – The limited accuracy of the mechanics, such as the goniometer (the motor that selects the projection angle), is another source of misalignment.
- **Random motion** – Since the object is not completely fixed, some motion may occur when the goniometer rotates the object to the next projection angle. Also, the object might be moving while scanning, for example if a microscopy sample drifts within the sample holder.

For common reconstruction algorithms, it is assumed that no perturbations in the geometry are present and the projection angles $\tilde{\theta}_i$ are known exactly. The geometry that they impose is given by the set of parameters:

$$\left.\begin{aligned} \chi_i &= 0 \\ \zeta_i &= 0 \\ \phi_i &= 0 \\ \theta_i &= \tilde{\theta}_i \end{aligned}\right\} \quad i = 1, \ldots, K. \tag{3.5}$$

In reality, small perturbations are common even in the projection angles. Therefore, the geometry of the reconstruction method needs to be adjusted, to match the true geometry of the projection acquisition. For this purpose we need to introduce a mathematical framework that allows us to incorporate the geometry in the

reconstruction. In the next section, we introduce this mathematical framework and discuss how we can incorporate the alignment in the reconstruction method.

### 3.2.2 Mathematical formulation

The object can be related to its projections in a continuous manner by means of the Radon transform [NW01]. To solve the tomographic reconstruction problem, we define a discrete model of the Radon transform. The object is defined on a grid of unit cubes called *voxels*. Each voxel is assumed to have a constant gray value $x_j$. These are stored as a vector $\boldsymbol{x} \in \mathbb{R}^N$. The gray value is proportional to the attenuation coefficient of the corresponding material. The projection domain, consisting of a series of projection images, is discretized into a series of pixel grids. A detector value $p_i$ is modeled as a weighted sum (a single line projection) of the object gray values

$$p_i := \sum_{j=1}^{N} w_{ij} x_j. \tag{3.6}$$

The weight $w_{ij}$ models the attenuation of the source ray $i$ caused by the material within voxel $j$. Different weight models exist that approximate the physical interaction of the radiation source with the object. In Fig. 3.3, as an example the strip model is illustrated.

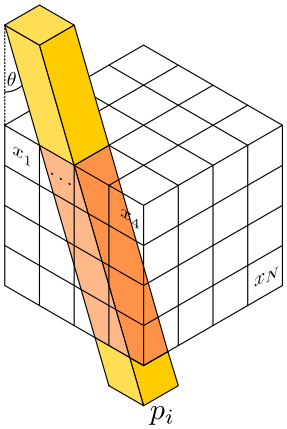



Figure 3.3: Discretization of the forward projection in three dimensions. The projection $p_i$ is a weighted sum of the gray values. The weight $w_{ij}$ of voxel $j$ is determined by the intersection volume of ray $i$ and the voxel. This is the so-called strip model. The ray is indicated in yellow.

We can now describe the relation between the measurements $\boldsymbol{p}$ and the unknown object $\boldsymbol{x}$ by a system of linear equations

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p}. \tag{3.7}$$

The projection matrix $\boldsymbol{W}$ has dimensions $M \times N$, the line projections $\boldsymbol{p} \in \mathbb{R}^M$ and the object $\boldsymbol{x} \in \mathbb{R}^N$ are stored as column vectors. The number of projection images

is $K$ and the number of detector pixels is $D$, so the number of rows in $\boldsymbol{W}$ is given by $M = KD$.

The weights $w_{ij}$ are not only determined by the ray width or voxel size, but they also depend on the geometry. This can be seen in Fig. 3.3. Hence, by writing the *geometrical parameters* as vectors,

$$\begin{aligned}\boldsymbol{\theta} &= (\theta_1, \ldots, \theta_K)^\intercal,\\ \boldsymbol{\chi} &= (\chi_1, \ldots, \chi_K)^\intercal,\\ \boldsymbol{\zeta} &= (\zeta_1, \ldots, \zeta_K)^\intercal,\\ \boldsymbol{\phi} &= (\phi_1, \ldots, \phi_K)^\intercal,\end{aligned}$$

the dependency of the projection operator on the geometry is posed as a nonlinear equation system

$$\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x} = \boldsymbol{p}. \tag{3.8}$$

In practice, it is not straightforward to use a closed-form expression for the weights $w_{ij}$ as function of the geometrical parameters. They are generated on-the-fly based on the geometrical parameters by means of a ray tracing type algorithm. More details of this operation are discussed in Section 3.3.

If the unknown object can be represented exactly on a voxel grid, the system is consistent if the geometrical parameters are perfectly aligned and when the projection images do not contain noise. This enables us to formulate an alignment method as an algorithm for minimizing the inconsistency of Eq. (3.8):

$$\underset{\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\chi},\boldsymbol{\zeta},\boldsymbol{\phi}}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x} - \boldsymbol{p}\|^2. \tag{3.9}$$

The factor $\frac{1}{2}$ is introduced to avoid a factor two in the gradient of this cost function and simplify the notation. There are many approaches to solve this optimization problem. Minimization with respect to $\boldsymbol{x}$ is a linear optimization problem, which is highly underdetermined. The dependency on the projection angles $\boldsymbol{\theta}$ makes Eq. (3.9) also a nonlinear problem. In addition to the large scale of the data, it is a very difficult optimization problem. However, by using the right approach, it is typically possible to solve this problem accurately. In the next section we will discuss in detail the approach of projection matching.

### 3.2.3 Projection matching

In this section we will explain the projection matching method used for the alignment.

Minimizing the projection distance defined in Eq. (3.9) can be done in several ways. The most important goal is to find an accurate reconstruction $\boldsymbol{x}$. Without an accurate geometrical parameter set, this is not possible. Moreover, to estimate the geometrical parameters, a reasonable reconstruction should be available. It seems that this problem can only be solved effectively by considering the full optimization problem Eq. (3.9) and estimate $\boldsymbol{x}$ and the geometrical parameters simultaneously. This approach has been proposed in [BB13; YNP05]. Since the

inverse problem in Eq. (3.8) is heavily underdetermined (*e.g.*, $N = 512 \times 512 \times 512$ and $M = 10 \times 512 \times 512$), ill-posed and nonlinear, this is a difficult task.

As an alternative approach, we can alternate between reconstruction and alignment. In this case, we treat both problems individually. During the reconstruction, we assume that the geometry is known, and during alignment, the reconstruction is fixed,

$$\text{Reconstruction}: \quad \underset{\boldsymbol{x}}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x} - \boldsymbol{p}\|^2, \tag{3.10a}$$

$$\text{Alignment}: \quad \underset{\boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\zeta}, \boldsymbol{\phi}}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x} - \boldsymbol{p}\|^2. \tag{3.10b}$$

Some algebraic reconstruction methods specifically solve Eq. (3.10a) in a certain norm. An example is the Simultaneous Iterative Reconstruction Technique (SIRT) [KS01]. Therefore, Eq. (3.10a) can be solved by employing a suitable reconstruction algorithm. In the second step, generated projections from the reconstruction using the forward model in Eq. (3.8), are matched to the observed projections. This sub-problem only has a solution when the difference between the ground truth and reconstruction is in the null space of the projection operator $\boldsymbol{W}$, which is normally not the case. Nevertheless, given a reconstruction containing some details or crude outlines of the object, it may be possible to improve the alignment parameters. The resulting family of alignment methods is termed *projection matching*. Examples of these methods are discussed in [HB11; Par+12]. All of these algorithms use an intermediate reconstruction step and afterwards apply a form of alignment that is often based on image registration of projection images and forward projections of the reconstruction. The basic structure of these methods is illustrated in a flowchart in Fig. 3.4. In the experiment and results section, we compare our method with a standard projection matching algorithm that incorporates SIRT as the reconstruction method. We will refer to this method as PM-SIRT.

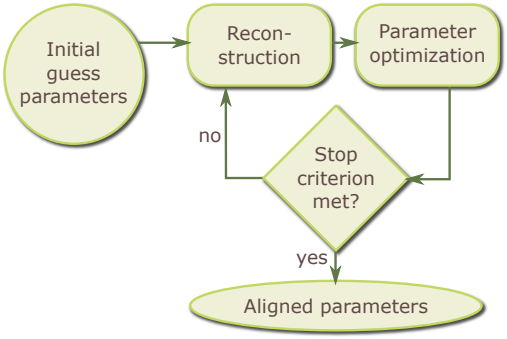



Figure 3.4: A flowchart of the projection matching algorithm that estimates the geometrical parameters.

### 3.2.4 Discrete tomography

Many objects that are scanned in tomography consist of just a few materials. Each material has a corresponding, uniform gray value in the reconstruction. For example, when scanning bones (ex vivo), the reconstruction has two gray values, one for the bone and one for the background (assuming that the bone is approximately homogeneous). In this case, we would like to reconstruct a binary volume. The discrete nature of the gray values can be exploited in the reconstruction. Instead of solving Eq. (3.9) over a continuous domain, we can limit the domain of $\boldsymbol{x}$ to the set of gray values that are known in advance. Let $R = \{\rho_1, \dots, \rho_l\}$ be the set of gray values in the ground truth image. The reconstruction problem that is considered in *discrete tomography* is posed as

$$\underset{\boldsymbol{x} \in \{\rho_1, \dots, \rho_l\}^N}{\text{minimize}} \; \|\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x} - \boldsymbol{p}\|^2 \tag{3.11}$$

In the experiment section, we consider projection datasets from an object with few gray values in the reconstruction (in this case two, these objects are called binary volumes). This allows us to use reconstruction algorithms for discrete tomography to approximately solve Eq. (3.11). Possible candidates for the reconstruction algorithm are described in [BS11; BT09; Sch+05].

## 3.3 Projection matching with discrete tomography

To our knowledge, employing a discrete tomography prior in projection matching has never been proposed before. As discussed previously, all common markerless alignment methods are not as accurate when the number of projection images is small. By applying a discrete tomography reconstruction algorithm we can introduce prior knowledge in a projection matching method, to alleviate this limitation.

Essentially, the reconstruction phase Eq. (3.10a) in the projection matching algorithm is performed by a discrete tomography reconstruction algorithm that solves Eq. (3.11). We employ the binary reconstruction algorithm proposed by Schüle *et al.*, from now on referred to as the DC-algorithm [Sch+05]. Here DC stands for Difference of Convex functions, which is a superclass of convex functions. This algorithm solves Eq. (3.10a) augmented with two priors using D.C. programming:

$$\underset{\boldsymbol{x} \in [0,1]^N}{\text{minimize}} \; \frac{1}{2}\|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{p}\|^2 + \alpha \boldsymbol{x}^\intercal \boldsymbol{L}\boldsymbol{x} + \frac{1}{2}\mu\langle \boldsymbol{x}, \boldsymbol{e} - \boldsymbol{x}\rangle, \quad 0 < \mu \in \mathbb{R}. \tag{3.12}$$

Here $\boldsymbol{e} := (1, \dots, 1)$. The middle term of Eq. (3.12) is a smoothness prior where the matrix $\boldsymbol{L}$ represents a difference operator between pixels and their neighbor pixels [Sch+05]. The final term steers the solution to a binary volume. The parameters $\alpha$ and $\mu$ express weights to these terms.

From just a few projection images it is possible to make a perfect reconstruction, under some smoothness conditions on the ground truth [Sch+05]. It has not

been investigated how well this algorithm performs in case of misaligned data. However, our numerical results suggest that the reconstructions in the early stages of the alignment algorithm are good enough for achieving convergence in the parameter estimation Eq. (3.10b).

The DT prior that is controlled by the parameter $\mu$ gradually forces the reconstruction towards a binary solution. That is, in the implementation $\mu$ is gradually increased while the change in the data fidelity term becomes smaller. The final DT reconstruction is approximately binary, but simply rounding by thresholding yields a true binary solution. This allows also the use of other DT reconstruction methods, such as the DART algorithm which also produces a discrete solution [BS11]. The reason we choose for the DC-algorithm by Schüle *et al.* is that DART is not as robust in case the projections are very noisy.

The alignment of the projection images in the second sub-problem Eq. (3.10b) is solved by employing Levenberg–Marquardt. This method is a gradient-based trust region method [NW06]. The algorithm computes a sequence $\{\boldsymbol{y}_k\}$ of alignment parameters,

$$\boldsymbol{y}_k = (\boldsymbol{\chi}_k, \boldsymbol{\zeta}_k, \boldsymbol{\phi}_k)^\intercal, \quad k = 1, 2, \ldots$$

that yield a decreasing cost function value Eq. (3.10b). The alignment parameters are updated by a descent direction $\boldsymbol{\eta}_k$,

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \boldsymbol{\eta}_k \tag{3.13}$$

where the descent direction is found by solving the following equation:

$$\left(\boldsymbol{J}_k^\intercal \boldsymbol{J}_k + \lambda_k \boldsymbol{D}^2\right) \boldsymbol{\eta}_k = -\boldsymbol{J}_k^\intercal \boldsymbol{r}_k. \tag{3.14}$$

The residual $\boldsymbol{r}_k := \boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi}_k, \boldsymbol{\zeta}_k, \boldsymbol{\phi}_k)\boldsymbol{x}_k - \boldsymbol{p}$ has Jacobian $\boldsymbol{J}_k := \nabla_{\boldsymbol{y}_k} \boldsymbol{r}_k$. The descent direction is found by minimizing a second order Taylor approximation of the cost function represented by Eq. (3.14). The parameter $\lambda_k$ controls the step size of the weighted descent direction, *i.e.*, it limits $\|\boldsymbol{D}\boldsymbol{\eta}_k\|$. The scaling (diagonal) matrix $\boldsymbol{D}$ is necessary to incorporate the different scales of the geometrical parameters. For example, shifts are in the order of voxels, while the in-plane rotation is in radians and is therefore about two orders of magnitude smaller. In our approach we selected the scalings manually. The scales of shifts $(\boldsymbol{\chi}, \boldsymbol{\zeta})$ are unchanged, *i.e.* $d_{ii} = 1$ for corresponding scales. The rotations $(\boldsymbol{\phi})$ are scaled in units of radians, *i.e.* $d_{ii} \approx \pi/180$. For further details of the Levenberg–Marquardt method we refer the reader to [MNT04; Mor78].

As noted previously, the projection matrix $\boldsymbol{W}$ is generated on-the-fly. Therefore, the matrix is never fully formed in memory, which enables reconstruction of large datasets. Storing a full matrix is not feasible for many practical applications. The details of the ray tracing method are as follows: for each projection angle, the ray paths incident to each detector pixel are traced through the volume and the weights are measured based on an interpolation method by Joseph [Jos82]. The direction of the rays, and therefore the voxels that are intersected are determined by the alignment parameters in Eq. (3.8). Since we take this approach for computing a projection, and because representing this matrix in closed form is not

practical, we do not use an analytic expression for the Jacobian matrix. Instead, we employ a central finite difference approximation for gradients:

$$\nabla_{\chi_i} \boldsymbol{r} \approx \frac{\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi} + h\hat{\boldsymbol{e}}_i, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x} - \boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi} - h\hat{\boldsymbol{e}}_i, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x}}{2h}, \quad i = 1, \ldots, K \tag{3.15}$$

where $\hat{\boldsymbol{e}}_i$ is the $i$th basis vector. The gradients of the other parameters are computed likewise. The coefficients of $\boldsymbol{W}$ are only piecewise continuously differentiable with respect to the alignment parameters. This is due to discretization of the reconstruction volume. However, the experimental results in Section 3.4 suggest that the finite difference approach is effective and is not hampered by the fact that $\boldsymbol{W}$ is not differentiable in every point.

The step size of the finite difference method $h$ is selected automatically, following the same procedure as proposed in [BB13]. Here we found that the step size is important for the accuracy of the gradient. If it is too large, the approximation Eq. (3.15) is very poor. If the step size is too small, discretization effects and limited precision distort the measurement of the gradient. These effects are visible at fine scales in the cost function as erratic oscillations. The method we employed in [BB13] compares the cost function in Eq. (3.10b) to a spline fitted to sample points of the cost function. More sample points are compared to the spline and the relative fit yields a measure for the smoothness of the cost function. A step size $h$ determines the distance between sample points. The smallest step size is selected for which the cost function behaves relatively smooth, to make the gradient more robust against these discretization effects. The selected step size behaves like a constant and does not change much in our pilot experiments. Therefore, it is computed once after the initial reconstruction and is fixed during the rest of the computations. This improves the performance of the method, since computing the step size is not cheap.

Since each parameter affects only detector values at a single projection angle, the Jacobian has the following sparse structure

$$\boldsymbol{J} := \begin{pmatrix} \boldsymbol{g}_1^{\chi} & & & \boldsymbol{g}_1^{\zeta} & & & \boldsymbol{g}_1^{\phi} & & \\ & \boldsymbol{g}_2^{\chi} & & & \boldsymbol{g}_2^{\zeta} & & & \boldsymbol{g}_2^{\phi} & \\ & & \ddots & & & \ddots & & & \ddots \\ & & \boldsymbol{g}_K^{\chi} & & & \boldsymbol{g}_K^{\zeta} & & & \boldsymbol{g}_K^{\phi} \end{pmatrix} \tag{3.16}$$

with $\boldsymbol{g}_1^{\chi}$ the numerical gradient of $\boldsymbol{r}$ with respect to $\chi_1$. Due to the independence of the parameters between different projection images, these finite differences can be computed very efficiently. It costs only two forward projections for each parameter:

$$\begin{pmatrix} \boldsymbol{g}_1^{\chi} \\ \vdots \\ \boldsymbol{g}_K^{\chi} \end{pmatrix} := (\boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi} + h\boldsymbol{e}, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x} - \boldsymbol{W}(\boldsymbol{\theta}, \boldsymbol{\chi} - h\boldsymbol{e}, \boldsymbol{\zeta}, \boldsymbol{\phi})\boldsymbol{x})/(2h)$$

with $\boldsymbol{e} := (1, \ldots, 1)^{\mathsf{T}}$.

The full algorithm is listed in pseudo code in Algorithm 1. The Jacobian is needed to solve Eq. (3.14). We employ the linear least squares solver LSMR to solve this equation [FS11]. Note that the Jacobian is generated by computing Eq. (3.16). In the case we need to multiply the Jacobian with a vector, such as in LSMR, it can be efficiently computed by inner products. Since the Jacobian is constant in LSMR, the descent direction is found efficiently, even though we need to solve a system of equation. When the descent direction $\boldsymbol{\eta}_k$ is found, the value $\rho$ is computed. The numerator represents the decrease in the cost function Eq. (3.10b). The denominator is the decrease in the quadratic model of the cost function based on a second order Taylor approximation:

$$\frac{1}{2}\|\boldsymbol{r}(\boldsymbol{y}+\boldsymbol{\eta})\|^2 \simeq L(\boldsymbol{\eta}) := \frac{1}{2}\boldsymbol{r}(\boldsymbol{y})^\intercal \boldsymbol{r}(\boldsymbol{y}) + \boldsymbol{\eta}^\intercal \boldsymbol{J}^\intercal \boldsymbol{r} + \frac{1}{2}\boldsymbol{\eta}^\intercal \boldsymbol{J}^\intercal \boldsymbol{J}\boldsymbol{\eta}. \tag{3.17}$$

So, if $\rho$ is strictly positive, we have found a descent direction. If, in addition, the fraction $\rho$ is close to 1 the quadratic model is in good correspondence with the cost function. In that case, the damping parameter $\lambda$ is decreased, such that a larger step is taken in the next iteration. Note that this parameter limits the $\ell^2$ norm of the descent direction. Therefore, the step size is increased only if the quadratic model is accurate. Otherwise the step size is decreased.

## 3.4 Experiments and results

In this section we discuss the results of a series of simulation experiments that have been performed to evaluate the capabilities of the DTPM algorithm, and compare it to projection matching without the use of prior knowledge. Based on a binary volume, we generated projection data and introduced misalignment in the geometry. From this data we can compare the accuracy of the standard projection matching using SIRT (PM-SIRT) as a reconstruction algorithm, with DTPM. Both algorithms are also tested for robustness against noise in the projection data.

The forward model in Eq. (3.7) can be used to generate projection data from a given volume $\boldsymbol{x} \in \mathbb{R}^N$. For creating forward and back projections as well as the SIRT and DC reconstruction algorithms the ASTRA toolbox was used [PBS11; PBS13]. Matlab was used as a scripting language that accesses the underlying GPU code through the C++ mex interface. The hardware used for the simulations was a workstation with Intel Core i7-2600K CPU@3.40 GHz and a Geforce GTX 570 GPU.

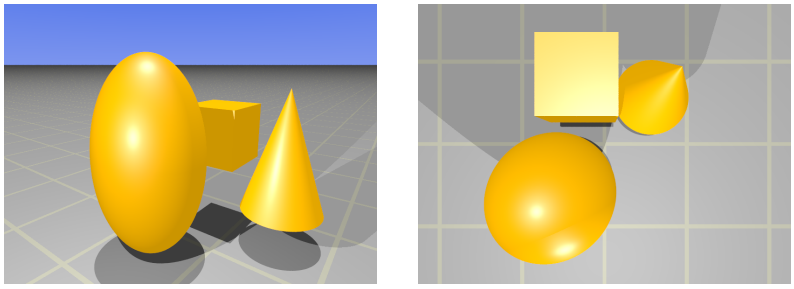
The phantom we consider is depicted in Fig. 3.5. It consists of the union and differences of several convex shapes. All objects are hollow, except for the cube, and the ellipsoid encloses a solid sphere. These simple shapes allow reconstructions from just a few projection images. In Matlab, we generated the phantom on a $64 \times 64 \times 64$ voxel grid. The middle slice of the object is shown in Fig. 3.8a.

**Algorithm 1** DTPM

**Input:** Projection data $\boldsymbol{p}$, initial geometry $\bar{\boldsymbol{y}}_0 = (\boldsymbol{\chi}_0, \boldsymbol{\zeta}_0, \boldsymbol{\phi}_0)^\intercal$
**Output:** A binary reconstruction $\boldsymbol{x}_{i_{\max}}$ and aligned geometry $\bar{\boldsymbol{y}}_{i_{\max}}$

Let $\mathrm{DC}[\boldsymbol{y}](\boldsymbol{p})$ denote the operator that produces a binary DC reconstruction operating on $\boldsymbol{p}$, for a given geometry $\boldsymbol{y}$.
The parameter $\epsilon$ controls the convergence criterion, that monitors the relative change in the parameters.

**for** $i = 0, \ldots, i_{\max}$ **do**
  *reconstruction phase*
  $\boldsymbol{x}_i = \mathrm{DC}[\bar{\boldsymbol{y}}_i](\boldsymbol{p})$
  *alignment phase, using Levenberg–Marquardt*
  $\lambda = 5$
  $\nu = 2$
  $\boldsymbol{y}_0 = \bar{\boldsymbol{y}}_i$
  **for** $k = 0, \ldots, k_{\max}$ **do**
    $\boldsymbol{r}_k := \boldsymbol{W}(\boldsymbol{y}_k)\boldsymbol{x}_i - \boldsymbol{p}$
    compute $\boldsymbol{J}_k(\boldsymbol{x}_i)$
    Solve $\left(\boldsymbol{J}_k^\intercal \boldsymbol{J}_k + \lambda \boldsymbol{D}^2\right)\boldsymbol{\eta}_k = -\boldsymbol{J}_k^\intercal \boldsymbol{r}_k$
    **if** $\|\boldsymbol{\eta}_k\| \leq \epsilon(\|\boldsymbol{y}_k\| + \epsilon)$ **then**
      *convergence*
      $k = k_{\max}$
    **else**
      $\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \boldsymbol{\eta}_k$
      $\rho = (\|\boldsymbol{r}_k\|^2 - \|\boldsymbol{r}_{k+1}\|^2)/(L(\boldsymbol{0}) - L(\boldsymbol{\eta}_k))$
      **if** $\rho > 0$ **then**
        $\lambda \leftarrow \lambda \max\{\frac{1}{3}, 1 - (2\rho - 1)^3\}$
        $\nu = 2$
      **else**
        $\lambda \leftarrow \lambda \nu$
        $\nu \leftarrow 2\nu$
        $\boldsymbol{y}_{k+1} = \boldsymbol{y}_k$
      **end if**
    **end if**
  **end for**
  $\bar{\boldsymbol{y}}_{i+1} = \boldsymbol{y}_{k_{\max}}$
**end for**



(a) Front view (b) Top view

Figure 3.5: Binary phantom consisting of an ellipsoid, a cone and a cube. A plane is shown for more perspective and improved visibility. The phantom can be enclosed in a cube consisting of $2 \times 2 \times 2$ grid cells (as seen from the top view).

### 3.4.1 Experiment I – the effect of discrete tomography

In the first experiment we focus on the effects of DT reconstruction. Therefore we look at a perfectly aligned dataset and do not consider alignment.

The strength of the DC-algorithm is in its ability to accurately reconstruct a dataset from just a few projections. It is not clear for this phantom, how many projection angles are necessary to create an accurate reconstruction. Therefore, we compare SIRT and DC reconstructions for a varying number of projection angles. We generated projection data for 50 angles. A subset of these projection images were used to create a reconstruction.

Fig. 3.6 shows a plot of the number of incorrect voxels, called the *voxel error*, as a function of the number of projection angles. The number of projection angles that were considered are: 3 to 10, 20, 25, 30, 40 and 50. The smoothness parameter $\alpha = 0.08$ for DC was chosen empirically.

As expected, DC outperforms SIRT substantially in the limited angle case. However, this difference disappears if many projection images are used. Therefore we focus our simulations on the limited angle case (less than 10 projection angles).

### 3.4.2 Experiment II – aligning projection data

In the second experiment, perturbations were applied to the projection geometry used to simulate the projection images. From the parameters in Section 3.2.1, we have included the in-plane shift $(\chi, \zeta)$ and in-plane rotation $\phi$. Because these parameters operate in the plane of the detector, the alignment phase Eq. (3.10b) can be seen as an image registration problem, in which the observed projection image acts as the reference image.

An in-plane detector shift of maximum ±10 detector pixels, both in $y$- and $z$-directions was simulated. The in-plane rotation is at most ±15°. These perturbations were generated randomly with uniform distribution. The misaligned projection data was used as input for the alignment software.

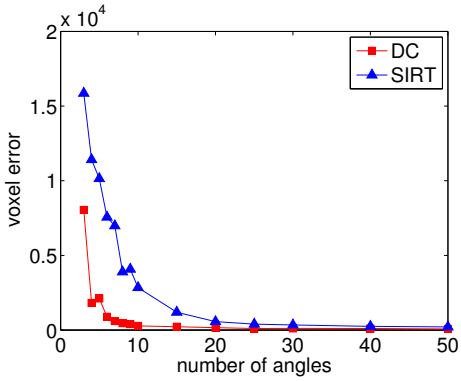
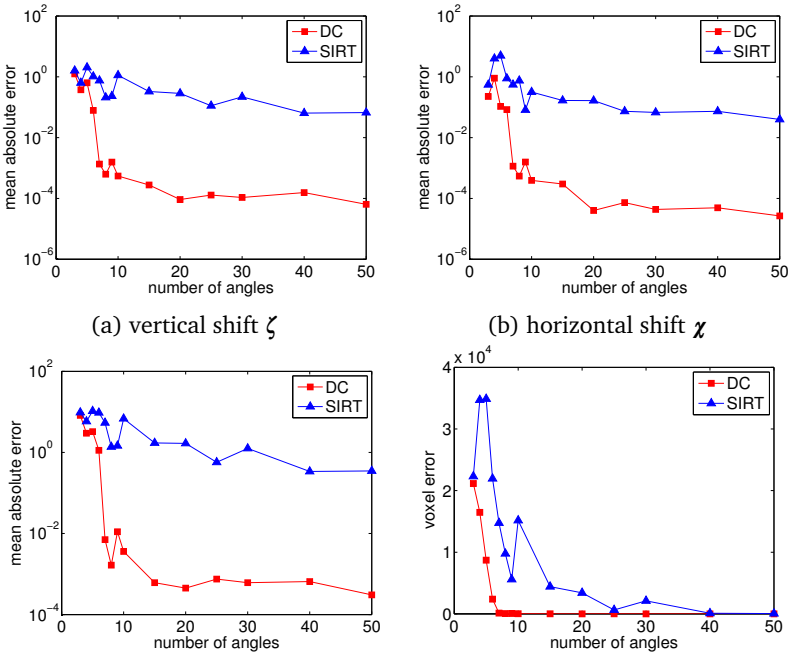



Figure 3.6: Number of incorrect voxels, in comparison with the ground truth, for a varying number of projection angles.

In Figs. 3.7a to 3.7c, the final alignment error for the three parameters is shown for a varying number of projection angles. The errors are averaged over all projection angles, so they represent the mean error per projection image. With a maximum misalignment of 10 voxels (in the object position), the initial average error should be close to 5, while the initial average error for the in-plane rotation will be around 7.5°. The performance of PM-SIRT is unsatisfying and improves only moderately when the number of angles is increased. In case of shifts, the parameters are found with an accuracy of 1 to 2 detector pixels. Such errors still produce smearing in the reconstruction. The DTPM algorithm using DC also fails with 3 to 5 projection angles. However, for 8 projection images, the average error in the shift is in the order of $10^{-4}$, so sub-pixel accuracy is achieved. The voxel error in Fig. 3.7d indicates that this kind of accuracy in the alignment is sufficient, since the corresponding voxel error is close to zero. Hence, the DC approach seems suitable in the limited angle case and achieves a higher accuracy overall.

For a qualitative comparison, the middle slices of the phantom and reconstructions are shown in Fig. 3.8. The reconstructions are computed from eight projection images. From these results it is clear that the alignment error for PM-SIRT is not small enough to eliminate all alignment artifacts. In contrast, the DTPM alignment produces an almost perfect reconstruction. It is not clear, however, if the difference in reconstruction quality is due to the alignment. It might be the case that DC creates a more accurate final reconstruction, while the alignment parameters are comparable. To exclude this possibility we have also created a DC reconstruction using the alignment parameters found by PM-SIRT, as shown in Fig. 3.8d. This DC reconstruction still contains artifacts in the surface of the objects. This indicates that the intermediate DC reconstructions yield a better alignment, and it is not just the final discrete tomography reconstruction that accounts for the differences in quality.




(a) vertical shift $\zeta$ (b) horizontal shift $\chi$

(c) in-plane rotation $\phi$ (d) pixel error

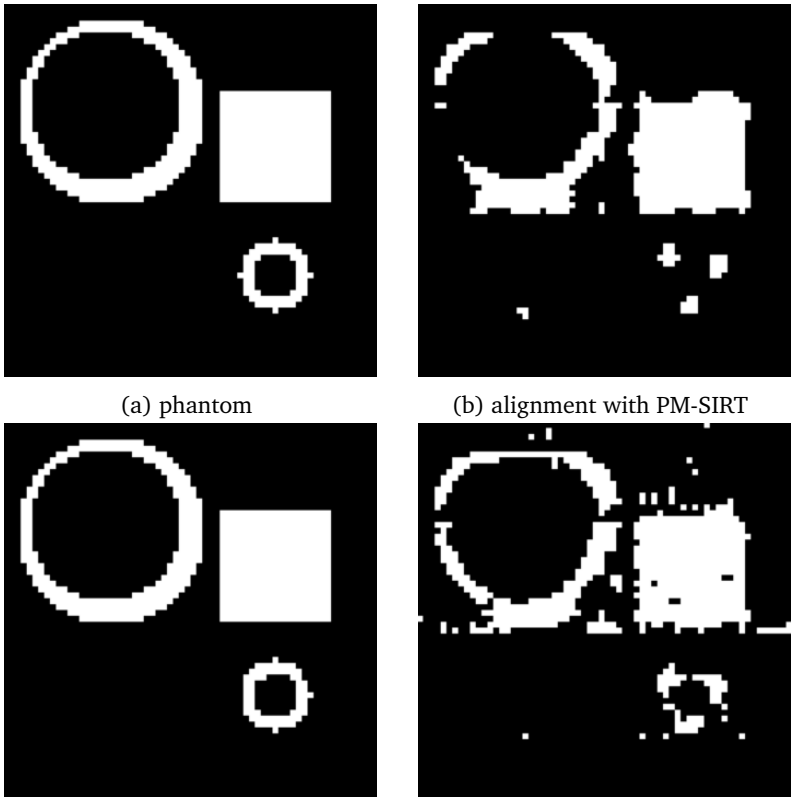
Figure 3.7: Alignment results for varying numbers of projection angles. The error measure is the absolute error averaged over the number of projection angles: $\|\chi - \chi^*\|/K$ where $\chi^*$ is the true shift. Similar measures are used for the other parameters. (d) A plot of the voxel error reveals that the DTPM alignment produces perfect reconstructions, if the number of angles is large enough.

### 3.4.3 Experiment III – aligning noisy projection data

In this experiment we test the robustness against noise of the DTPM algorithm, to determine if the discrete tomography prior still works well if the projection data contains noise.

We focus on reconstructing from eight projection images. From the previous experiment we found that this number of angles should be enough for accurate alignment. We applied Poisson noise to the projection data. The amount of noise is indicated by the simulated photon count for the projection data. A lower photon count corresponds to a higher noise level. In this particular case, we vary the noise level to match a simulated photon count from $10^3$ to $5 \times 10^6$. These can be considered moderate to limited noise levels.

In Fig. 3.9, the absolute, averaged alignment errors are shown for the full range of noise levels. These plots show that alignment is very difficult when the noise levels are high. Clearly, it is not possible to accurately align a dataset if the noise level crosses a certain threshold. The DTPM alignment seems decent if the photon count is at least $10^6$ (small amount of noise). In this case, sub-pixel



(a) phantom (b) alignment with PM-SIRT

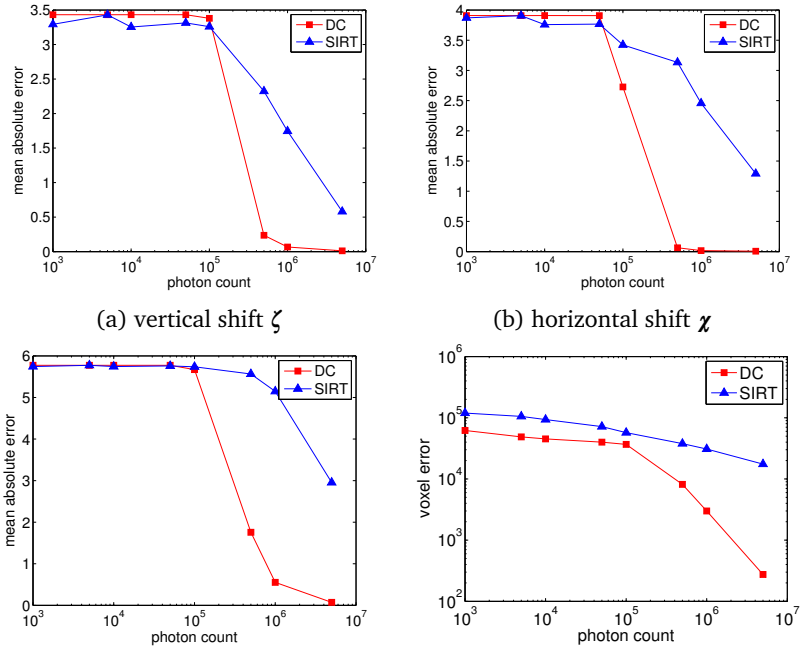
(c) alignment with DTPM (d) DC with PM-SIRT parameters

Figure 3.8: Comparison of a reconstructed slice (the middle one) resulting from PM-SIRT and DTPM. The reconstruction is based on 8 projection images. (d) Shows a DC reconstruction using the geometrical parameters found by PM-SIRT. This shows that the quality is not due to the final reconstruction using DC, but due to the alignment.

accuracy is achieved, while in-plane rotation is found with an accuracy of a few degrees. The voxel error in Fig. 3.9d reveals that the DTPM reconstructions are reasonable for photon counts above $10^5$. A voxel error of $10^4$ corresponds to approximately 4% incorrect voxels. The PM-SIRT alignment results are much worse. While the vertical alignment improves with increased photon counts, the horizontal shift and in-plane rotation do not improve much. Presumably, due to the noise, details are missing in the SIRT reconstruction, such that the alignment step does not improve the parameters in a direction that will create a better reconstruction in the next iteration.

To visually assess the quality of the reconstructions, we show the middle slices in Fig. 3.10a and Fig. 3.10c for PM-SIRT and DTPM respectively, corresponding to a dataset with simulated photon count of $10^6$. Although the DTPM reconstruction

is not as good as in the noiseless case, the contours of the objects are clearly visible. Moreover, the surfaces of the ellipsoid and cone in the DTPM reconstruction are fully closed. The cube is showing very minor salt and pepper noise, but its cross-section is clearly a square. The PM-SIRT reconstruction, in contrast, fails to produce the cone and the surface of the ellipsoid is far from closed. Again we reconstructed using DC with the alignment parameters from PM-SIRT. This reconstruction, shown in Fig. 3.10b, is slightly better, because the cross-section of the cone is visible. The result again indicates that the alignment parameters found by DTPM are more accurate and that not the final DC reconstruction causes the qualitative differences. This is confirmed by the final errors in the alignment parameters shown in Fig. 3.11. The horizontal and vertical shifts are found with sub-pixel accuracy by DTPM. The in-plane rotation is accurate up to 1 or 2 degrees. In comparison, the alignment by PM-SIRT is very poor.




(a) vertical shift $\zeta$ (b) horizontal shift $\chi$

(c) in-plane rotation $\phi$ (d) voxel error

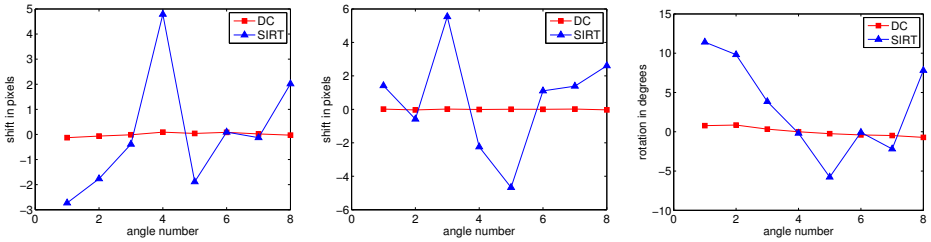
Figure 3.9: Alignment results for eight projection images perturbed by Poisson noise.

### 3.4.4 Performance considerations

In this subsection we compare and discuss the computation times of PM-SIRT and DTPM. As computation times depend on the implementation and the particular dataset that is considered, we provide a rather general discussion of the performance. The alignment phase is the same for both methods and therefore has the

(a) PM-SIRT (b) DC with PM-SIRT parameters (c) DTPM

Figure 3.10: Alignment result for eight projection images. The Poisson noise has an intensity corresponding to a photon count of $10^6$. The figures show the middle slice of the reconstruction resulting from the alignment algorithm. In (b) a DC reconstruction is shown, using the aligned parameters found by PM-SIRT.




(a) residual vertical shift (b) residual horizontal shift (c) residual in-plane rotation

Figure 3.11: The errors in the alignment parameters after alignment. The dataset used 8 projection images perturbed by noise, corresponding to a photon count of $10^6$.

same complexity. Differences in computation times for the alignment are caused by differences in the number of iterations that are required for convergence. However, the largest difference is due to the reconstruction algorithm that is employed. The method SIRT requires a forward projection and a back projection in each iteration. This is computed as two matrix vector products, using Eq. (3.7) (one by $\boldsymbol{W}$ and one by $\boldsymbol{W}^\intercal$). The method DC has an inner and outer loop. The outer loop controls the parameter $\mu$, but does not perform actual computations. The inner loop consists of a forward and back projection and is therefore comparable to a SIRT iteration. In the experiments, the number of SIRT iterations was kept constant at 300 iterations. Typically, the number of outer loop iterations of DC is half that number. Each iteration of the outer loop, the inner loop is run on average 10 times, so in total the DC method performs about 5 times as many matrix vector products, compared to SIRT.

For any given dataset it will be difficult to estimate the performance between

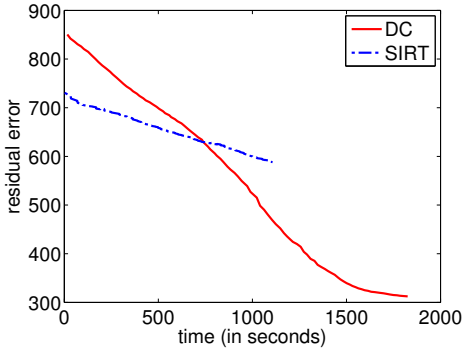



Figure 3.12: Computation times.

DC and SIRT, because we do not know the number of inner or outer loop iterations that will be performed by DC. The number of iterations of both loops is determined by their convergence criteria. Therefore, we compare performance results of a single run of the algorithm based on our phantom dataset.

In order to compare computation times, the residual error of both methods are plotted against the computation time, shown in Fig. 3.12. The same noisy dataset was used as in experiment III. This result shows that DTPM is about two times slower compared to PM-SIRT. However, the final residual error of PM-SIRT is achieved much faster using DTPM. In that sense, DTPM converges faster and is a more efficient method. Notice that the initial residual error of DTPM is higher than that of PM-SIRT. This shows that the DC reconstruction method is not as accurate compared to SIRT when the alignment error is large. Most likely, the discrete tomography prior does not improve the reconstruction if the data consistency is small, due to misalignment.

## 3.5 Discussion

The results from the experiments on simulated projection data show that the DTPM method achieves a lower error in the reconstruction as well as the alignment parameters, when compared to PM-SIRT. In the experiments we applied both methods to a dataset using simulated projection data from a phantom object. In this section we will discuss our expectations and considerations for applying our method to data from real experiments.

First of all, the results of experiment I show that 8 projections are enough to accurately reconstruct the binary phantom object. If more than 20 projection are available, the difference between SIRT and DC reconstructions is only minor. So in order to benefit from using the DTPM alignment method on real data, the number of projections should be within a certain interval.

When reconstructing from experimental data we cannot determine the voxel error of a reconstruction, since we do not have a ground truth. Neither can we determine the optimal number of projection angles using the voxel error.

However, it is still possible to estimate a reasonable number of projections. If a manually aligned dataset could be obtained, we can compare residual errors between SIRT and DC. The residual error compares simulated projections from the final reconstruction with the measured projection data. Using this norm, we can create a similar plot as shown in Fig. 3.6. Although the residual error has no one-to-one correspondence with the voxel error, it still can give an approximate insight to the real error.

Secondly, the amount of noise should be limited in order to benefit from using the DTPM alignment method. From Fig. 3.9 we see that the alignment fails if the signal-to-noise ratio is too low. Acceptable noise levels can be determined by qualitatively observing the aligned reconstruction. As an alternative, the total variance of the projection images could be considered as a measure for the amount of noise.

Moreover, it is important to note that alignment errors can be present in the projection data that cannot be modeled by our set of alignment parameters. Although the alignment parameters we consider have typically the largest perturbations in an experimental dataset, other errors can be present as well. For example, we do not consider rotation of the object around the $x$-axis. A small rotation would result in vertical "shrinking" of the projection image and certain features would overlap. Such an alignment error can only be corrected for if the rotation around the $x$-axis is included in the parameters of DTPM. However, this parameter is highly correlated with the shift parameters used in DTPM. Likewise, errors in the rotation angle are difficult to find, due to the high nonlinear nature of the parameter. Nevertheless, we think that estimating in-plane shifts and rotations will improve the alignment of an experimental dataset even if other alignment errors exist.

Finally, the choice of the smoothness prior parameter $\alpha$ is important in the DC reconstruction. This parameter reflects the spatial coherency of solutions. The phantom in Fig. 3.5 is very smooth indeed. In reality, this prior might not be as accurate, depending on the nature of the object. Selecting the parameter $\alpha$ can be achieved in a similar fashion as choosing the number of projections. By minimizing a measurable error norm, such as the residual error, we can select the value for $\alpha$ that results in the lowest error.

## 3.6 Conclusions

A new method was proposed for alignment of binary tomography datasets from limited data. Prior knowledge of the binary gray values was included as a regularization method. It was found that the use of discrete tomography in a projection matching method, results in more accurate intermediate reconstructions. As a result, the subsequent alignment step by matching projection images is better defined.

For aligning the projection images, a variant of the Levenberg–Marquardt algorithm was used. By using finite differences for computing derivatives, the method does not require analytic gradients.

From our numerical results for simulated data, we see that the binary tomography algorithm DC yields much more accurate reconstructions in the case of few projection images (less than 15) [Sch+05]. The projection matching method DTPM is able to effectively align datasets from 6 projection images or more. The alignment combined with SIRT fails to find accurate alignment parameters with few projection angles. Only for a large number of angles the results are improving.

Qualitatively, the difference between PM-SIRT and DTPM was clear. While DTPM results in almost perfect reconstructions, for datasets with limited noise, the PM-SIRT reconstructions have many artifacts. By creating a DC reconstruction combined with geometrical parameters found by PM-SIRT, we showed that differences in quality in the reconstructions are mainly due to the alignment accuracy.

The results show that employing binary tomography as regularization is an essential step in projection matching alignment, when only a few projection images are available.

# Chapter 4

# SDART: an algorithm for discrete tomography from noisy projections

## 4.1 Introduction

In tomographic imaging, a three dimensional object is reconstructed from a series of projection images that have been acquired over a range of angles. Tomography has a wide variety of applications, ranging from medical imaging to materials science [Grü+03; MD09; Neu97; Rop+03; Zen10]. In many of these applications, the object under investigation consists of only a few different materials, each corresponding to a particular gray level in the reconstructed image. Therefore, the set of *gray values* in a reconstruction should be small and discrete. Most common reconstruction algorithms, such as the Filtered Back Projection or SART, produce a continuous range of gray values [KS01]. However, it has been shown that incorporating this set of admissible gray values as prior knowledge can lead to superior image quality in the reconstruction, especially if the set of projection images is small [Bat05; BS11; CRT06; Sch+05]. This type of tomography is known as *discrete tomography*.

The Discrete Algebraic Reconstruction Technique (DART) is one such algorithm that exploits the discrete nature of the object. It assumes that the gray values corresponding to the different compositions of the object are known a priori [BS11]. If only the number of different gray values is known, and not their actual values, these gray values can be adaptively estimated during reconstruction by using PDM-DART [ABS12]. DART is an iterative method, which aims to solve a system of linear equations that models the tomographic projection process. In each iteration, a reconstructed image is segmented, *i.e.*, the gray values are thresholded to the nearest a priori known gray value. It is assumed that the interior regions

This chapter has been published with minor modifications as:
F. Bleichrodt, F. Tabak, and K. J. Batenburg. "SDART: An algorithm for discrete tomography from noisy projections". In: *Computer Vision and Image Understanding* 129 (2014), pp. 63–74

of this segmentation are segmented with high accuracy and that most errors are located on the boundaries. The key idea behind DART is to reduce the system of equations in each iteration by fixing or removing these interior pixels/voxels (*i.e.*, unknowns) from the equations. The governing equations in tomography are ill-conditioned and rank deficient. Due to this dimension reduction, the equation system becomes increasingly better determined. Nevertheless, removing unknowns assumes that the gray values of the corresponding pixels are correct. Only the remaining *free* pixels are iteratively refined. Therefore, the operation of fixing a pixel imposes a hard constraint on the solution of the equation system and can only be effective if the selection criterion for a pixel being fixed or free is sufficiently accurate.

In practice, we see that the interior regions of the segmentation initially contain many errors. However, since the boundaries evolve due to the update steps, these pixels will be corrected eventually and the algorithm can converge to the correct solution. A problem occurs when the projection data contain noise. Imposing hard constraints on non-boundary pixels leads to noise being spread mainly over boundary pixels. This leads to major errors in the update steps applied to the boundary pixels. As a result, edges will be less resolved in the reconstruction and convergence problems can arise.

In this chapter, we propose an alternative to the hard constraints imposed in DART. We introduce a set of relaxation parameters that imposes soft constraints on the pixel values. The parameters penalize deviation from the current segmented value of a pixel. Subsequently, the relaxed system with soft constraints is solved and the parameters are updated based on the intermediate reconstruction and segmentation. The proposed method is called Soft DART, to indicate the use of soft constraints. By using a *penalty matrix*, flexibility is increased in comparison to DART. It enables us to impose confidence levels on individual pixels instead of indicating if a pixel is correct or not. The results of our simulation experiments suggest that for a suitably chosen set of relaxation parameters and for datasets with low signal-to-noise ratios (SNR), SDART produces a reconstruction closer to the ground truth when compared to DART.

The outline of this chapter is as follows: first we will briefly discuss the DART algorithm in Section 4.2 and show some of its limitations. In Section 4.3 we will introduce SDART: a new variant of DART that includes a soft constraint. Possible choices for selecting the soft constraints are discussed. In Section 4.4 we compare the behavior of DART and SDART to see the effect of the soft constraints. We also discuss how to select an important regularization parameter that is used in SDART. In Section 4.5 an overview is given of the experiments and the results are discussed. We conclude the chapter in Section 4.6.

## 4.2 The original DART algorithm

In this section we will briefly introduce the notation and concepts of DART and summarize the main details of the algorithm.

### 4.2.1 Notation and concepts

The governing equations in tomography can be posed as a linear system, which models the tomographic projection process. The linear model is generic, but for simplicity we will focus on the reconstruction of a 2D slice of the object from 1D detector measurements. The generalization to three dimensions is straightforward. Throughout this chapter we consider a parallel beam geometry, which is illustrated in Fig. 4.1. In our implementation the geometry can be easily changed to fan or cone beam geometry. In fact, one of the experiments from Section 4.5 is based on a cone beam dataset.

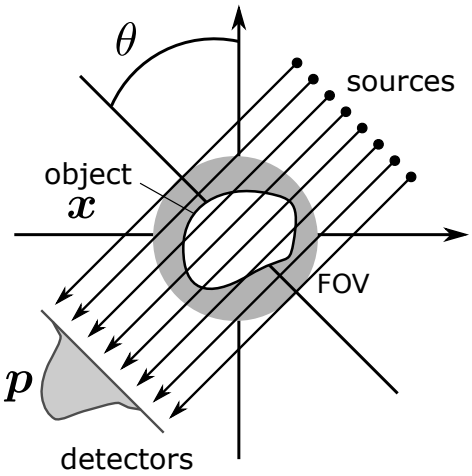



Figure 4.1: Schematic of the parallel beam geometry. The incident rays are parallel. The sources and detectors rotate around the object, indicated by the projection angle $\theta$, such that a circular field of view (FOV) is formed.

Let $\boldsymbol{x} \in \mathbb{R}^N$ denote a vector containing the gray value of each pixel in the unknown object. The vector $\boldsymbol{p} \in \mathbb{R}^M$ contains the detector measurements. The 1D detector has $D$ elements and $K$ projections are available. The total number of line projections is therefore $M = KD$. We now introduce the projection operator $\boldsymbol{W} \in \mathbb{R}^{M \times N}$, which relates the object to its projections:

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p}. \tag{4.1}$$

Reconstruction methods aimed at solving Eq. (4.1) are referred to as *algebraic reconstruction methods*. Examples of such methods based on Kaczmarz' method are ART, SIRT or SART [KS01]. Since the system of equations is usually underdetermined, and in practice no solution exists due to noise, it is typically solved in a least squares sense:

$$\underset{\boldsymbol{x} \in \mathbb{R}^N}{\text{minimize}} \; \|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{p}\|_2^2, \tag{4.2}$$

such that an object is found that matches with the observed data optimally. In discrete tomography, the small, discrete set of admissible gray values

$$R = \{\rho_1, \ldots, \rho_l\}$$

is known a priori. We can include this prior knowledge as constraints in the optimization problem:

$$\underset{\boldsymbol{x} \in \{\rho_1, \ldots, \rho_l\}^N}{\text{minimize}} \; \|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{p}\|_2^2. \tag{4.3}$$

### 4.2.2 Algorithm details

DART combines a continuous algebraic reconstruction method (ARM) with a segmentation step and uses heuristics to improve on this segmentation. The ARM that is typically used is SIRT or SART [KS01]. In principle, any linear least squares solver is suitable. Also the Krylov subspace methods such as CGLS or LSQR [PS82].

The flowchart in Fig. 4.2a illustrates the algorithm and its computational steps. We will briefly summarize these:

1. An initial continuous reconstruction $\boldsymbol{x}_c$ is computed using an algebraic reconstruction method.

2. The reconstruction is segmented by applying thresholding. All pixel values are rounded to the nearest gray value in the set $R$.

3. Those pixels that have at least one (out of eight) neighbor with a different gray value (called *boundary pixels*) are free. In addition, a random subset of image pixels is also selected to be free. The columns of $\boldsymbol{W}$ corresponding to the pixels that are not free (*fixed pixels*) are removed from Eq. (4.1) and their projections are subtracted from the right-hand side.

4. The solution of the reduced system is refined by applying an ARM to the free pixels.

5. If a stop criterion is not met, the free (boundary) pixels are smoothed. The smoothing step is performed by means of a discrete convolution of a $3 \times 3$ kernel with the image. The middle pixel of the kernel is weighted by a smoothing factor $b$, the other pixels in the kernel are weighted by $(1-b)/8$. Although the smoothing is not used in every implementation of DART [Alp+13], it is used in the original paper [BS11]. The process repeats from step 2.

The thresholding step rounds the pixel values of the reconstruction to the nearest a priori known gray value. We use the same notation for this operation as presented in [BS11]:

$$T(\boldsymbol{x}) : \mathbb{R}^N \mapsto R. \tag{4.4}$$

Note that the discrete nature of the gray values is not exploited in the ARM iterations. Instead, DART relies on the segmentation to produce solutions with discrete

gray values. Nevertheless, the strength of DART is based on the observation that pixels in the interior of a homogeneous region are likely to be thresholded correctly [BS11]. This result can be found empirically. A possible explanation is that least squares methods in general reconstruct low-frequent components of the solution prior to the high frequencies. Since large homogeneous regions (including the background) are part of the low-frequent components of the image, the non-boundary pixels are better resolved compared to the boundary pixels. This idea is used to classify the segmented image $\boldsymbol{x}_s$ into the sets of fixed pixels $F$ and free pixels $U$.

Formally, the sets $F$ and $U$ are then defined as index sets:

$$F = \left\{ i \mid x_i = x_{i+rn+q}, \quad \text{for all } q, r \in \{-1, 0, 1\} \right\}, \tag{4.5}$$

$$U = F^c, \tag{4.6}$$

where $F^c$ denotes the mathematical complement of $F$. In addition, the set of free pixels $U$ is combined with a random subset of pixels. Each pixel has a probability $(1-p)$ to be included in the set of free pixels. The probability $0 < p \leq 1$ is referred to as *fix probability*. This random set of free pixels improves the reconstruction of "holes" in the object, which are typically not found easily.

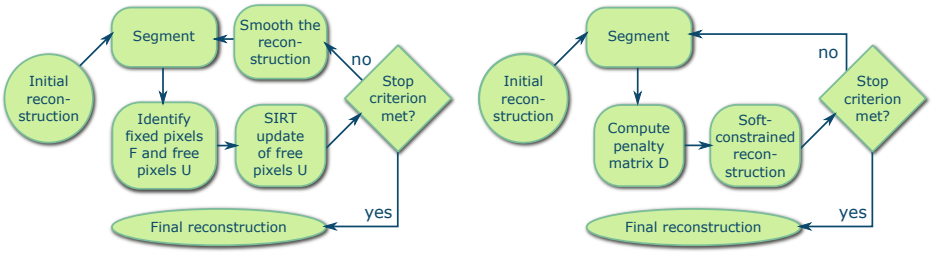
Since pixels in the interior regions are likely to be correct, they are removed from the equation system Eq. (4.1) and subtracted from the right-hand side. To fix a pixel $i \in F$, we apply the following operation on the linear system:

$$\begin{pmatrix} | & & | & | & & | \\ \boldsymbol{w}_1 & \cdots & \boldsymbol{w}_{i-1} & \boldsymbol{w}_{i+1} & \cdots & \boldsymbol{w}_N \\ | & & | & | & & | \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_{i-1} \\ x_{i+1} \\ \vdots \\ x_N \end{pmatrix} = \boldsymbol{p} - v_i \boldsymbol{w}_i, \tag{4.7}$$

where $v_i$ is the segmented gray value of pixel $i$. Subsequently, other pixels in $F$ are treated in an analogous way. This leads to a *reduced system* that has fewer unknowns. The pixels in the free set $U$ are refined by iterating an ARM on the reduced system.

This process is repeated in each DART iteration. The boundary pixels are determined from the complete image, not only from the free pixels corresponding to the reduced system. Therefore, the elimination of pixels in the fixed set always starts from the full system in Eq. (4.1). As a consequence, pixels that were previously fixed can be free in a consecutive DART iteration. Therefore, errors in the interior regions can be corrected in a later stage due to evolution of the boundaries.

Batenburg and Sijbers show that with increasing noise levels, the DART reconstructions have a large pixel error (*i.e.* the number of pixels that have a wrong gray value in the reconstruction compared to the ground truth) [BS11]. Since only boundary pixels are free, the noise has a large effect on the boundary update. To remedy this problem, the fix probability can be decreased such that noise is




(a) DART (b) SDART

Figure 4.2: The flowcharts illustrate the DART and SDART algorithms. The SIRT update can be replaced by another ARM. The smoothing step in DART is used to suppress the effect of noise on the boundary update. Note that SDART includes a penalty matrix $\boldsymbol{D}$, that represents the soft constraints used in the subsequent reconstruction step. SDART does not include a smoothing step.

also spread over a large random subset of pixels. While this improves the accuracy of DART with noisy projection data to some extent, it does introduce heavy salt and pepper noise, as was also observed in [Alp+13].

## 4.3 Soft DART

The main contribution of this chapter is to introduce a different approach to classify and improve incorrectly segmented pixels. Instead of fixing pixels and updating free pixels, we propose to solve a relaxed system (*i.e.*, under soft constraints) as an alternative to the ARM update step. In this section we will discuss the main details of the new approach.

A flowchart of the new method is shown in Fig. 4.2b. SDART follows the same steps as DART, but it does not eliminate unknowns (pixel values) from Eq. (4.1).

We propose to introduce a soft constrained optimization problem. Let $\boldsymbol{v}$ be the segmentation of the intermediate reconstruction and let $\boldsymbol{D} \in \mathbb{R}_+^{N\times N}$ be a diagonal matrix with nonnegative real entries ($d_{ii} \geq 0$, it is referred to as *penalty matrix*). We then introduce the relaxed reconstruction problem:

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \left\| \begin{pmatrix} \boldsymbol{W} \\ \lambda \boldsymbol{D} \end{pmatrix} \boldsymbol{x} - \begin{pmatrix} \boldsymbol{p} \\ \lambda \boldsymbol{D}\boldsymbol{v} \end{pmatrix} \right\|_2^2 \equiv \underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{p}\|_2^2 + \lambda^2 \|\boldsymbol{D}(\boldsymbol{x} - \boldsymbol{v})\|_2^2. \quad (4.8)$$

In this setting, the diagonal matrix element $d_{ii}$ gives a penalty to pixel $i$ for deviating from its segmented value $v_i$. If $d_{ii}$ is large, only small deviations are allowed, while the pixel can be considered "free" if $d_{ii} = 0$.

The system in Eq. (4.8) is solved by a linear least squares solver, *e.g.*, a reconstruction method such as SIRT or a Krylov subspace method such as CGLS. The algorithm is started with initial guess $\boldsymbol{x}_0 = \boldsymbol{x}_c$, the reconstructed image (with

continuous gray levels) that resulted from the previous SDART iteration. The parameter $\lambda$ is introduced both for regularization, as well as to compensate the difference in scale of the two terms in the cost function in Eq. (4.8). Note that the term $\|\boldsymbol{W}\boldsymbol{x}-\boldsymbol{p}\|_2^2$ depends on the number of angles, whereas $\|\boldsymbol{D}(\boldsymbol{x}-\boldsymbol{v})\|_2^2$ does not. As a result, the scaling between the two terms is important and the value of $\lambda$ needs to be adjusted accordingly.

The entries of $\boldsymbol{D}$ depend on the current reconstruction, so the matrix $\boldsymbol{D}$ needs to be updated at each SDART iteration. The main advantage of this approach, compared to using hard constraints, is that no pixel will be truly fixed. Therefore, noise in the projection data will be distributed over the entire image (proportional to $\boldsymbol{D}$). Moreover, the relaxation parameters $d_{ii}$ can express a confidence level for the accuracy of pixel's $i$ gray value. The confidence level can be based on any error measure for the reconstruction we have. Due to the generality of Eq. (4.8) we can even choose a different reference image $\boldsymbol{v}$ instead of the segmented reconstruction.

Naturally, the increased flexibility comes with a price. The system has gained $N$ unknowns as well as $N$ equations, making it more costly to solve Eq. (4.8). In addition, we lose the efficiency resulting from the removal of columns from Eq. (4.1). Instead, each SDART iteration will be as costly as the first. As will be explained in Section 4.3.2, an efficient implementation can still lead to satisfactory performance. The full algorithm is presented in pseudo code in Algorithm 2. Note that a stopping criterion is not included in the algorithm description. In general, the question when to stop an algorithm to obtain the best solution is a very difficult one. Therefore, a reasonable choice is to terminate the algorithm when the relative change in the solution is small. This can be achieved by using a fixed number of iterations. In the experiments section we use 30 to 50 iterations, which is enough for all the datasets we considered.

### 4.3.1 Selecting a penalty matrix

In our simulation experiments in Section 4.5, we consider two different penalty matrices, defined as follows:

#### DART criterion

For validation purposes, we introduce a penalty matrix that should result in SDART mimicking the original DART algorithm. SDART does not allow us to fix pixels, but instead we can give non-boundary pixels a very large weight. By giving a weight of zero to boundary pixels, we do not put any restrictions on those pixels. The resulting penalty matrix is given by

$$d_{ii} := \begin{cases} 10^6, & i \in F \\ 0, & i \in U. \end{cases} \tag{4.9}$$

These weights are found to be effective from preliminary simulation experiments. We refer to SDART using this penalty matrix as SDART-ORIG, the first variant of SDART.

**Neighbor criterion**

In DART, a pixel is fixed when all 8 neighbors have the same gray value. If at least one neighbor has a different gray value, the pixel is considered free. This leads to a relatively fat boundary. Therefore, a logical choice for $\boldsymbol{D}$ would be to give a penalty that is proportional to the number of neighbors $b_i$ that have a different gray value, *i.e.*,

$$b_i := \sum_{r=-1}^{1} \sum_{q=-1}^{1} \mathbf{1}_{\{x_i \neq x_{i+rn+q}\}}, \tag{4.10}$$

where $\mathbf{1}_{\{\}}$ is an indicator function that is 1 if the condition is true and 0 otherwise. In this way, boundary pixels have different weights that reflect their position in the boundary. As a result, the boundary can be considered to be narrower. The penalty matrix is then defined as

$$d_{ii} := \frac{100}{3^{b_i}}. \tag{4.11}$$

Note that $d_{ii}$ is an exponential, monotonically decreasing function in $b_i$. The factor 3 in the denominator was chosen based on early simulation experiments. If the factor is too small or too large, the reconstruction will either not change much in each iteration or noise is distributed more on the boundary, respectively. This version of SDART is referred to as SDART-NB.

### 4.3.2 Solving the soft constrained system

In this section we will go into more detail how the soft constrained optimization problem in Eq. (4.8) is solved.

This optimization problem involves solving the system

$$\begin{pmatrix} \boldsymbol{W} \\ \lambda \boldsymbol{D} \end{pmatrix} \boldsymbol{x} = \begin{pmatrix} \boldsymbol{p} \\ \lambda \boldsymbol{D}\boldsymbol{v} \end{pmatrix} \tag{4.12}$$

in a least squares sense.

The matrix $\boldsymbol{W}$ has $M$ rows and $N$ columns and is typically very large, especially in the three dimensional case, where the number of voxels/pixels in the reconstruction grid and in the projection data, is large. In Eq. (4.12) we added another $N$ rows to the system matrix and right-hand side. However, in our implementation using the ASTRA toolbox [PBS13; PBS11], the full matrix is never formed explicitly. If a matrix–vector product is computed, we split this operation in two parts: $\boldsymbol{W}\boldsymbol{x}$ and $\lambda \boldsymbol{D}\boldsymbol{x}$. The first matrix–vector product is computed by generating $\boldsymbol{W}$ on the fly to avoid high memory usage. This can be done efficiently due to our GPU implementation. The second part $\lambda \boldsymbol{D}\boldsymbol{x}$ is simply an inner product, because $\boldsymbol{D}$ is diagonal and is stored as a vector. Therefore, the computational overhead compared to solving Eq. (4.2) is small.

For solving Eq. (4.12) we can apply any linear least squares solver. However, we noticed during preliminary experiments that methods based on Kaczmarz' method such as SIRT [KS01], have slow convergence and do not yield very

**Algorithm 2** SDART

**Input:** Projection data $\boldsymbol{p}$
**Output:** Segmented reconstruction $\boldsymbol{x}_s$

1. Let $\boldsymbol{x}_c$ be the initial CGLS reconstruction from projection data $\boldsymbol{p}$.
Compute the initial segmentation $\boldsymbol{x}_s = T(\boldsymbol{x}_c)$.

**repeat**

*/* Setting up the soft constraints */*
2. Compute the matrix $\boldsymbol{D}$ based on $\boldsymbol{x}_s$.
3. Set $\boldsymbol{v} = \boldsymbol{x}_s$.
4. Set $\boldsymbol{x}_0 = \boldsymbol{x}_c$.

*/* Soft constrained reconstruction */*
5. Apply CGLS with initial solution $\boldsymbol{x}_0$ to solve:

$$\underset{\boldsymbol{x}_c}{\text{minimize}} \; \|\boldsymbol{W}\boldsymbol{x}_c - \boldsymbol{p}\|_2^2 + \lambda^2 \|\boldsymbol{D}(\boldsymbol{x}_c - \boldsymbol{v})\|_2^2$$

*/* segmentation */*
6. Compute $\boldsymbol{x}_s = T(\boldsymbol{x}_c)$.

**until** convergence

accurate results. Krylov subspace methods perform better in this case. We found that the method CGLS performs very well and methods such as LSQR and LSMR are suitable too [FS11; PS82], but they all have slightly different results. This is why we have chosen to combine CGLS with SDART in our numerical experiments in Section 4.5.

## 4.4 A numerical study

In this section we highlight differences in behavior of DART and SDART using numerical experiments. We also introduce an experimental way to compute the regularization parameter $\lambda$ that has an important role in the convergence of SDART. We want to point out that this section serves the reader to illustrate the different behavior between DART and SDART. Therefore, the choice of our phantom shown in Fig. 4.3 is somewhat arbitrary.

### 4.4.1 Behavior of DART compared to SDART

The effect of the hard versus soft constraints of DART and SDART can be visualized by looking at the evolution of boundary pixels (*i.e.* free pixels in case of DART).

Since SDART has no concept of free pixels, we show the penalty matrix $\boldsymbol{D}$ after rescaling its values such that the elements are in the range $[0,1]$, *i.e.*, we show the image $\{x_i\}$ such that

$$x_i = 1 - \frac{d_{ii}}{\max_j(d_{jj})}. \tag{4.13}$$

In this example we consider the cylinder block phantom, shown in Fig. 4.3. The projection data were computed by forward projecting the image, using a parallel beam geometry. In total 25 projections were computed at equidistant angles in the domain $[0,\pi)$. The projection data were perturbed by Poisson noise. Noise due to a limited photon count, which is encountered in many types of tomography, follows a Poisson distribution. The intensity of the noise is quantified by the photon count of the incident X-ray beam, when no object is between the source and detector. In other words, this represents the total dose that is emitted during the full scan of the object. In this case, noise was simulated corresponding to a photon count of $10^3$. In DART, the fix probability was set to 0.99. With very low signal-to-noise ratios, a lower fix probability is preferred, but this would make it difficult to show the boundary evolution.

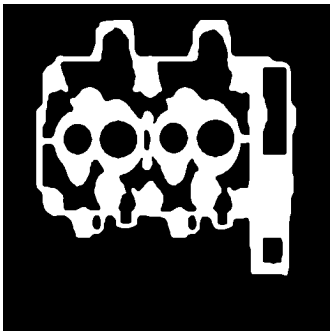


Figure 4.3: The cylinder phantom of size $512 \times 512$.

In Fig. 4.4 the boundary evolution of both DART and SDART-NB (using the neighbor criterion) for the first three full iterations is shown. Note that for SDART-NB, we show the weights represented in Eq. (4.13). A pixel is black if the corresponding penalty of the pixel is maximum, $\max_i d_{ii}$. This is comparable to a fixed pixel in DART. A white pixel corresponds to a minimum penalty. The pixel attains any other gray value if it is in between these extrema. This representation of the "amount of fixedness" of pixels is not directly comparable to DART's free and fixed pixels. However, we think that these images give insight in the different ways that DART and SDART update the reconstruction.

The initial boundary of DART in Fig. 4.4a is only slightly refined in the next iterations. Although the boundary becomes thinner, many of the background pixels are indicated as free pixels. In SDART-NB, we see a similar thinning of the boundaries. Moreover, the contours of the ground truth image are approximated more accurately. Background pixels have a large weight (indicated by black

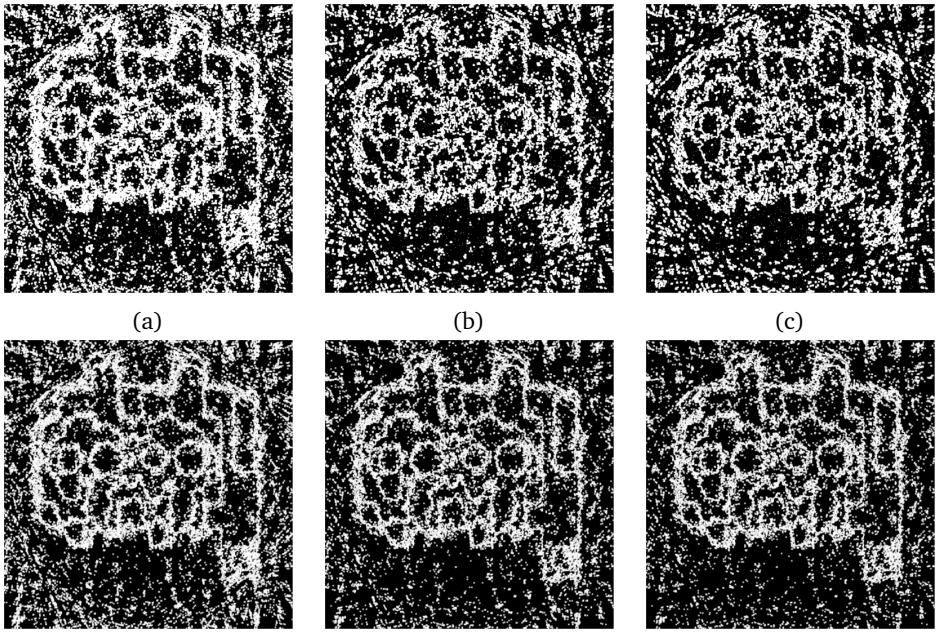



Figure 4.4: Comparison of the boundary evolution of "free" pixels in DART, (a) – (c) and SDART-NB (d) – (f), for the first three iterations.

pixels). This shows that the background is more homogeneous rather than that noise is producing clusters, as is the case in DART.

Images of the boundary evolution, do not give a clear insight in the quality of the reconstructions. Therefore, we also show the segmented intermediate reconstructions in Fig. 4.5. DART is distributing a significant part of the noise throughout the background, where many pixels are free. Another consequence of noise is visible in the jagged boundaries of the cylinder block. The reconstructions of SDART-NB have finer and more distinct boundaries. In addition, background noise is reduced within consecutive iterations.

The behavior shown in this example depends largely on the set of soft constraints imposed by the matrix $\boldsymbol{D}$. For example, a strategy of fixing pixels similar to DART (*e.g.* SDART-ORIG) is very effective for projection data without noise [BS11], while it fails in cases with heavy noise. Therefore, finding a single penalty matrix $\boldsymbol{D}$ that is accurate in all possible datasets is unlikely. An adaptive approach might be more successful. For example, the order of magnitude of the weights can be changed according to noise levels. In case of low noise levels, high weights steer the solution more to the segmentation, while smaller weights prevent overfitting to noise. We see an important role here for the parameter $\lambda$ in Eq. (4.8). It can be used to assign a larger weight to the data fidelity term or to correspondence to the segmented solution.

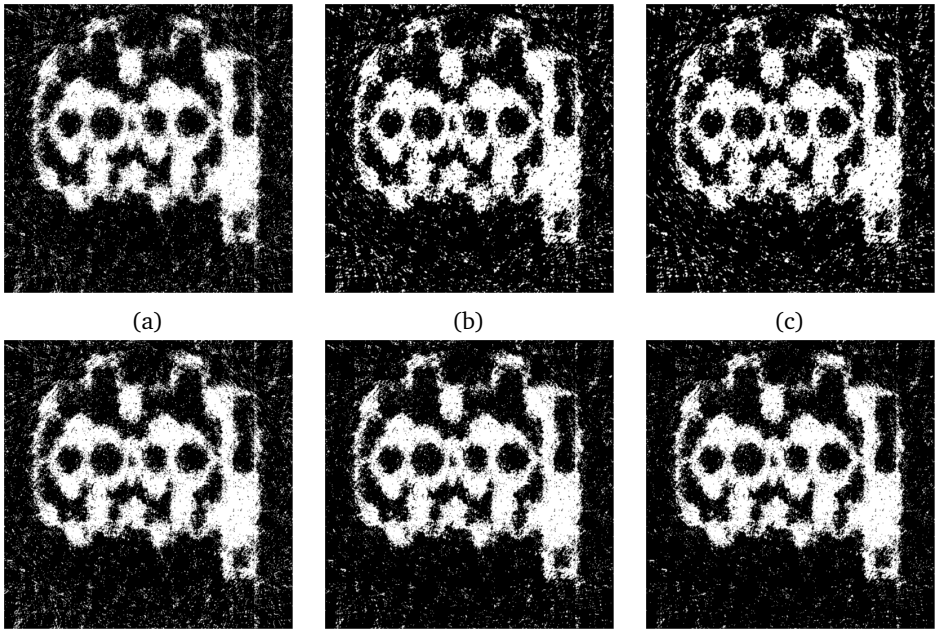


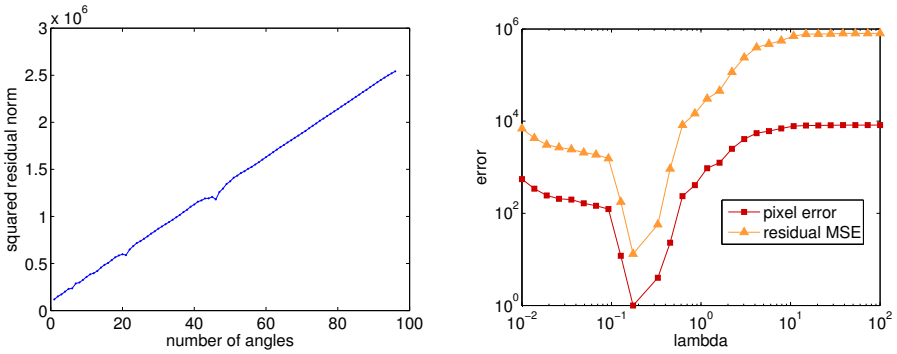
Figure 4.5: Comparison of the intermediate segmentations of the first three iterations of DART, (a) – (c) and SDART-NB, (d) – (f). These correspond to the boundaries in Fig. 4.4.

### 4.4.2 Selecting the regularization parameter

In this section we will discuss how to select a value for the regularization parameter $\lambda$ that is close to optimal, where we use the term *optimal reconstruction* to refer to the reconstruction with smallest pixel error over all possible choices for $\lambda$. Recall that the pixel error indicates the number of pixels that do not have the right gray value compared to the ground truth. We have chosen this error norm over, *e.g.*, a chi-squared or Jaccard distance, since our images are inherently discrete. A chi-squared measure is more suitable when comparing two images that are continuous with respect to their pixel values. Moreover, we expect that our findings will not be changed significantly when another error norm is used.

Consider the second formulation of the cost function in Eq. (4.8). It consists of two terms: a data fidelity term $\|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{p}\|_2^2$ and a discrete tomography prior $\|\boldsymbol{D}(\boldsymbol{x} - \boldsymbol{v})\|_2^2$. The order of magnitude of these terms is in general not directly comparable. Therefore, a regularization parameter $\lambda$ is added to properly adjust the bias to the discrete tomography prior.

Note that the magnitude of the data fidelity term depends strongly on the current solution $\boldsymbol{x}$ (and thus on the ground truth) as well as the number of projection angles. Adding more projection angles results in more rows in $\boldsymbol{W}$ as well as more elements in $\boldsymbol{p}$. By making the assumption that a projection image




(a) (b)

Figure 4.6: (a) Linear behavior of the data fidelity term $\|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{p}\|_2^2$ with respect to the number of projection angles; (b) For simulated projection data without noise, the residual mean squared error of an SDART reconstruction follows the same curve as the true pixel error, if the regularization parameter is varied.

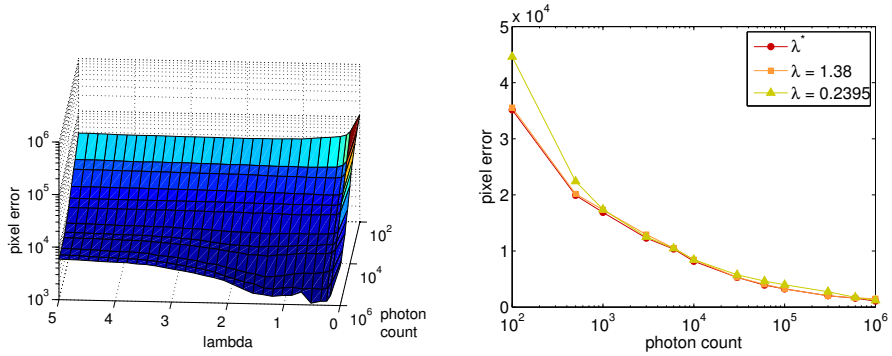
does not change in a small angular range, we can assume that the sum of squared residuals $\|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{p}\|_2^2$ behaves linearly in the number of angles. (Provided that the additional angles are close to the original angles.)

To verify this assumption, we perform a small numerical experiment. Considering the Shepp–Logan phantom, the bottom image in Fig. 4.10a, we simulate projection data for a fixed number of angles (*e.g.* 50). For these projection data we reconstruct the image by using 4 iterations of LSQR (as explained in the previous section). This yields an approximate reconstruction. Consequently we construct the projection operator $\boldsymbol{W}$ and corresponding right-hand side $\boldsymbol{p}$ (by forward projecting the ground truth image) for a varying number of equidistant projection angles. In Fig. 4.6a the squared residual norm is plotted for a varying number of angles, indeed showing a linear curve.

The last term in Eq. (4.8) does not depend on the number of projection angles. Instead it depends linearly on the number of pixels in the reconstruction. Since this is also true, to some extent, for the data fidelity term, no adjustments should be necessary if the number of reconstruction pixels is changed (*e.g.* a value $\lambda$ at low resolution can be found that is also suitable for high resolution reconstructions).

Due to the linearity of the data fidelity term, we can extrapolate $\lambda$ if a value is known for a dataset with few projection angles.

We still lack a way of determining a good value for $\lambda$ for a given dataset. For this goal we can use the residual $\ell_2$-norm. If the projection data are consistent with the ground truth (no noise), there is usually a good correspondence between the true (pixel) error and the residual of Eq. (4.1). We can exploit this to find a value for $\lambda$ for which the pixel error of the SDART reconstruction is minimal. In Fig. 4.6b we have plotted the pixel errors of the SDART-NB reconstructions and the value for $\lambda$ that was used. The result suggests that there exists an optimal $\lambda$




(a) (b)

Figure 4.7: Choosing the optimal value for lambda. (a) The mesh plot of the pixel error; (b) The pixel errors for $\lambda = 0.24$ are very close to the pixel errors for the optimal $\lambda^*$.

at which the pixel error takes its minimum value. Moreover, the residual $\ell_2$-norm agrees with this minimum. Therefore, if an initial guess for $\lambda$ is known, we can compute several SDART reconstructions for different $\lambda$ in a small range. If a minimum of the residual is found, we also found the optimal $\lambda$.

Of course this method will fail if noise is present in the projection data. Then the correspondence between the residual $\ell_2$-norm and the true error is in general very poor. Nevertheless, we will show that a value found for $\lambda$ for a phantom dataset (or a dataset with very low noise levels) may also give near-optimal results when high noise levels are considered.

In Fig. 4.7a, a mesh plot is shown of the pixel error for a range of $\lambda$ as well as several photon counts. For these computations the cylinder phantom in Fig. 4.3 was used and projections at 20 angles were computed. We see that the minimum pixel error is attained at $\lambda \approx 0.5$ for data with high signal-to-noise ratio (SNR). If the SNR is decreased, the optimal value for $\lambda$ does not seem to change much.

The optimal $\lambda$ for each SNR still varies slightly. The corresponding pixel errors are minimum at that specific SNR. We also selected a constant $\lambda^* \approx 1$ that attains pixel errors closest to these minimum pixel errors in a least squares sense. This is the optimal choice if we fix $\lambda$. The pixel errors are shown in Fig. 4.7b. We also plotted these curves for $\lambda = 1.38$ and $\lambda = 0.24$ (which was optimal for the noiseless case as shown in Fig. 4.6b). From this result we can see that while $\lambda^*$ is the better choice overall, choosing $\lambda = 0.24$, the same as in the noiseless case, produces near-optimal results. The key conclusion is that the value for $\lambda$ as in the noiseless case also works well in low-dose datasets.

In Fig. 4.8, the pixel error and residual are plotted for a dataset with limited noise (a photon count of $10^6$), for varying $\lambda$. From these data we see that there still is a good correspondence between the minimum pixel error and the minimum of the residual MSE. This implies that we can effectively estimate $\lambda$ for a dataset with high SNR. Consequently, datasets collected from the same object with low

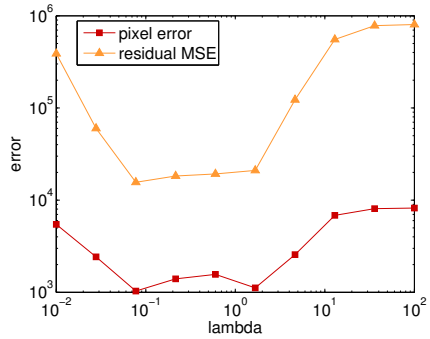



Figure 4.8: Correspondence between the residual mean squared error (MSE) and the true pixel error for relatively high photon count ($= 10^6$).

SNR can use the same $\lambda$.

## 4.5 Experiments and results

In this section we describe the experiments that were performed and their results. The main focus is to compare DART with SDART and a variety of well established reconstruction methods. In the experiments we focus on different conditions for which SDART is more accurate in comparison with DART. In addition, we explore the two sets of soft constraints $\boldsymbol{D}$ used in SDART-ORIG and SDART-NB. To validate our findings in the simulation experiments, we apply (S)DART to an experimental dataset with a low signal-to-noise ratio.

The results are compared to other well-known reconstruction methods. We include as a reference the filtered back projection (FBP) method, which is widely used for its speed and robustness[KS01]. However, a requirement for FBP is a large amount of angles, to avoid under-sampling, a requisite that is not met in the limited angle case. The algebraic reconstruction techniques SIRT and SART are included as well. Both methods are derived from ART, but their update steps are different. SART will update the reconstruction using one projection image at a time, while SIRT updates the reconstruction using all angles. Finally, we include the binary algebraic reconstruction technique BART, which is a method for reconstructing binary images [Her73]. This algorithm is implemented in the tomography package SNARK09 [KDH13], which we use in combination with the script as presented in [Alp+13]. This code follows the same computational steps as presented in the flowchart of the original paper in [Her73].

In the simulation experiments we consider three phantom datasets, shown in Fig. 4.10a, referred to as: *blob*, *cylinders* and *Shepp–Logan* (from top to bottom). The phantom images are of size $512 \times 512$. Projections were computed over an equidistant set of angles in the range of $[0, \pi)$. A parallel beam geometry is simulated with a 1D detector of 512 pixels, the same width as the phantom images. When noise is applied to projection data, it is sampled from a Poisson

distribution. Intensity of the noise is expressed in simulated photon counts, a lower photon count indicating more noise.

Each DART run is initialized by 40 iterations of the SIRT reconstruction algorithm [KS01]. DART uses 40 intermediate SIRT iterations to refine the boundary pixels. SDART is initialized by 40 CGLS iterations and uses 70 iterations to solve the soft constrained system in Eq. (4.8). SDART uses more intermediate iterations, since SDART solves the full equation system in Eq. (4.8), while DART solves a system of equations that has significantly fewer unknowns. Therefore, increasing the intermediate ARM iterations in DART will not significantly decrease the pixel error. In fact, for noisy projection data it is preferred to decrease the number of ARM iterations, to prevent overfitting to noise.

### 4.5.1 Experiment I – basic validation

In the first experiment we compare simulations of DART, the two variants of SDART, BART, SART, FBP and SIRT on noiseless data. The projection data are simulated using the forward model. This allows us to compare the reconstructions with the ground truth.

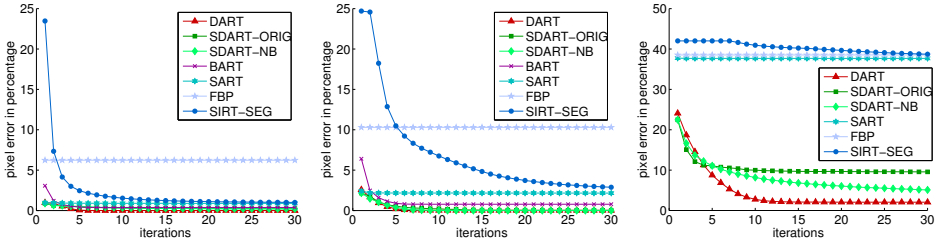
Note that the BART algorithm can only be applied on datasets from binary images, which rules out the Shepp–Logan phantom. The FBP, SART and SIRT methods do not provide a segmented image, so we cannot directly measure the pixel error. Therefore, we include a final segmentation step after the reconstruction. Similarly to the segmentation step in DART and SDART, the images are thresholded.

The number of projection angles was chosen as follows: 10 for the blob phantom, 25 for the cylinder phantom and 30 for the Shepp–Logan phantom. Using these number of projection angles, accurate reconstructions can be obtained in all three cases.

From the results of these experiments, shown in Fig. 4.9, we can conclude the following. In all cases, the filtered back projection is not accurate in terms of the pixel error. Clearly, the method is not suitable for the limited angle case. The methods SIRT and SART are comparable in accuracy, although SIRT converges slower to the minimum pixel error. For the Shepp–Logan phantom, the number of angles is clearly not enough to properly reconstruct, without using prior information. The BART algorithm achieves almost similar pixel errors when compared to DART and SDART. As we have discussed previously, BART could not be applied to the Shepp–Logan phantom.

Next, we focus on the differences between DART and SDART. SDART-NB with the neighbor constraints is in each case more accurate than SDART-ORIG. The accuracy of SDART-NB is comparable to DART in all datasets except for the blob image with hole. Since DART uses a fix probability of 0.99, it is able to detect the hole in the phantom. SDART has no random subset of free pixels and therefore is unable to detect the hole. This random subset could, however, easily be added to SDART to find holes such as in this case.

Based on the results in this experiment, we have decided not to include FBP and SART in the remaining experiments. For FBP, the number of angles is not




(a) Blob with hole (b) Cylinders (c) Shepp–Logan

Figure 4.9: The graphs show the convergence of DART and SDART compared to other common reconstruction algorithms.

enough to achieve a small pixel error. The SART algorithm achieves a similar pixel error compared to SIRT, but converges faster. However, we prefer SIRT over SART, because in each iteration, a back projection from all angles is performed. This way, noise in the data is averaged over all projection angles, instead of only one. Therefore, SIRT is expected to perform better in the case of high noise levels. Moreover, SIRT and SART are from the same family of algorithms and are expected to have similar results.

### 4.5.2 Experiment II – the effect of noise

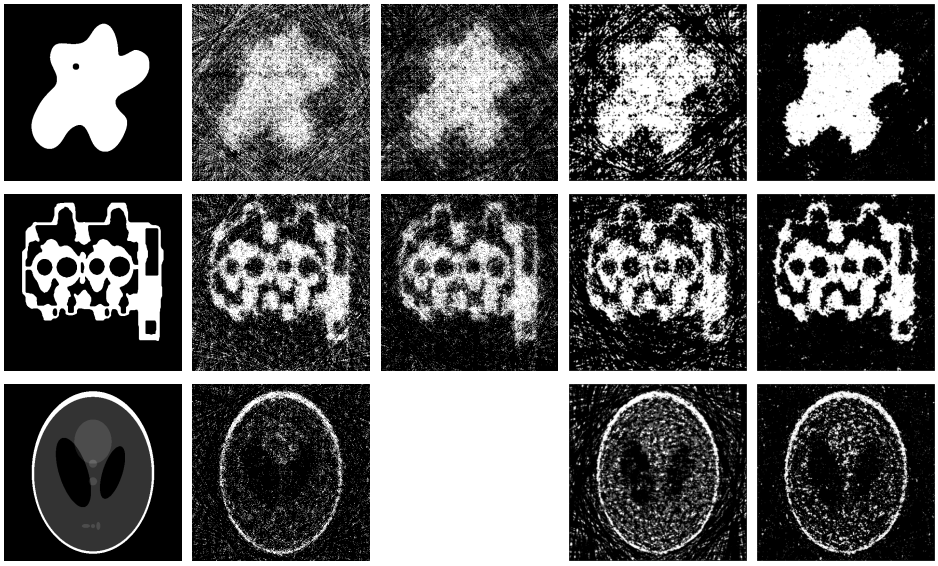
In the second experiment, we compare the performance of DART, SDART and BART over a large range of noise levels. As has already been indicated, DART results in poor reconstruction accuracy if the signal-to-noise ratio is low [BS11]. We expect that SDART will be more robust in this case.

We varied the noise levels from a photon count of $10^2$ (very high noise level) to $10^6$ (very low amount of noise). For each noise level a new sinogram was generated and Poisson noise was applied. The number of projection angles, 10, 25 and 30 for the blob, cylinder and Shepp–Logan phantom images respectively, was chosen such that a good reconstruction quality is possible when no noise is present in the projection data.

As the start solution for DART, a segmented SIRT reconstruction is used based on 40 iterations. The final (S)DART reconstruction should be an improvement of this initial reconstruction. Therefore, we also compare the results with this initial segmentation.

The results from experiment II are listed in Fig. 4.11. The errors, in percentages, indicate the percentage of pixels that have been segmented to the wrong gray value compared to the phantom image.

The pixel error of the (S)DART reconstructions are in general smaller than those of the initial segmentation. However, in a small interval of noise levels, DART shows a small regression in comparison with the initial segmentation. Apparently DART has problems in convergence for this particular interval of noise. This issue is especially visible in the case for the blob phantom for photon counts in between $10^3$ to $10^4$.



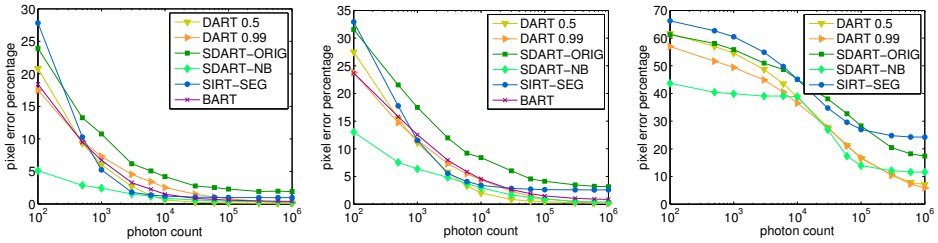
(a) (b) (c) (d) (e)

Figure 4.10: (a) Phantom images of dimension $512 \times 512$ pixels used in simulation experiments, referred to as: blob, cylinders and Shepp–Logan (from top to bottom); (b) The segmented SIRT reconstructions. The pixel errors corresponding from top to bottom reconstructions are: 27.6%, 18.2% and 41.9%; (c) The final BART reconstruction for blob and cylinders phantoms. The corresponding pixel errors, from top to bottom: 18.4% and 15.9%; (d) The final DART reconstructions. The pixel errors are: 17.3%, 13.7% and 48.1%; (e) The final SDART reconstructions. The pixel errors are: 3.9%, 7.7% and 39.9%.

The BART algorithm was also applied to the blob phantom and the cylinder phantom. It is performing very similar to the DART algorithm with a fix probability of 0.99. This leads us to the conclusion that BART is suitable for reconstructing binary objects, if the noise level is not too high.

For the blob phantom at high SNRs (photon counts in the range $10^4$-$10^6$), it seems that 10 projection angles is enough to have very accurate reconstructions, just by thresholding of the initial SIRT reconstruction. This is different, however, for high noise levels. At a photon count of $10^2$ we see that SDART produces an accurate reconstruction. However, the pixel error of DART and BART is increasing very rapidly if the photon count becomes smaller than $10^4$. The same trend, to some degree, can be observed in the other two phantom images. At high SNR, DART and SDART-NB perform equally well. SDART-ORIG, using the DART criterion, is performing badly. Instead of improving the initial segmented reconstruction, the pixel error is increased. We can conclude that SDART-ORIG is not suitable for noisy datasets.

The pixel errors do not give clear insight of the actual quality of the reconstructions. A pixel error of, say, 10% does not indicate where the incorrect pixels are




(a) Blob with hole (b) Cylinders (c) Shepp–Logan

Figure 4.11: The pixel error is expressed in terms of the noise level (in photon count).

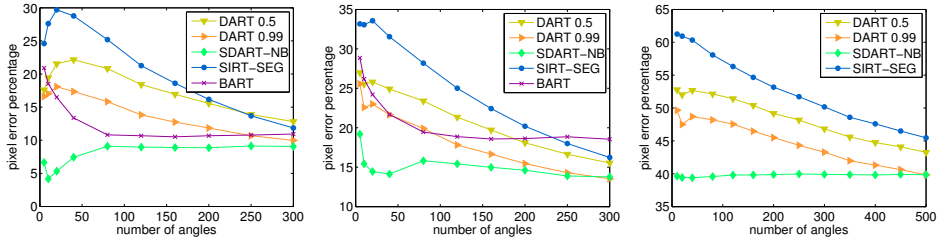
located and how the corresponding reconstruction looks. Therefore, we also show the reconstructed images in Figs. 4.10b to 4.10e. The blob phantom projection data has corresponding photon count of 100, the cylinder data 500 and for the Shepp–Logan data the photon count was 1000. In the fourth column of Fig. 4.10d we see DART reconstructions with fix probability 0.99. The SDART-NB reconstructions are shown in the last column in Fig. 4.10e. The difference between the reconstructions is clear, especially in the background. Moreover, the SDART reconstructions suffer far less from salt and pepper noise that is clearly visible in the DART reconstructions. The qualitative differences are supported by the corresponding pixel errors as listed in the caption of Fig. 4.10. The quality of the Shepp–Logan reconstruction is less impressive. Presumably, the number of gray values (6 in total) in the reconstruction is too much. The strength of the prior is reduced if the total number of distinct gray values is large.

From these data it becomes clear that employing SDART has an advantage over DART and BART if the signal-to-noise ratio is very low. In some cases SDART can generate good to reasonable reconstructions, while DART and BART perform badly.

### 4.5.3 Experiment III – adding more projection angles

Datasets with very low signal-to-noise ratios and few projection angles, in general, result in reconstructions with poor quality. Applying discrete tomography algorithms such as DART might show slight improvements in accuracy, but it is known that DART has problems with noisy datasets, which was also observed by Alpers *et al.* [Alp+13]. We have seen from the previous experiment that SDART-NB is favorable over DART in this case (from now on we do not include SDART-ORIG in the experiments). It is not clear, however, if this benefit is maintained when the number of projection angles is increased.

In the third experiment we compare the accuracy of DART, SDART and BART on the phantom datasets for a fixed, low signal-to-noise ratio, but the number of projection angles is varied. For the blob and cylinder phantom we chose a photon count of $10^2$. For the Shepp–Logan phantom a photon count of $10^3$ was used. Other details are the same as in experiment I. Since the performance of DART




(a) Blob with hole, photon count of $10^2$

(b) Cylinders, photon count of $10^2$

(c) Shepp–Logan, photon count of $10^3$

Figure 4.12: Simulation experiment at high noise levels and an increasing number of projection angles.

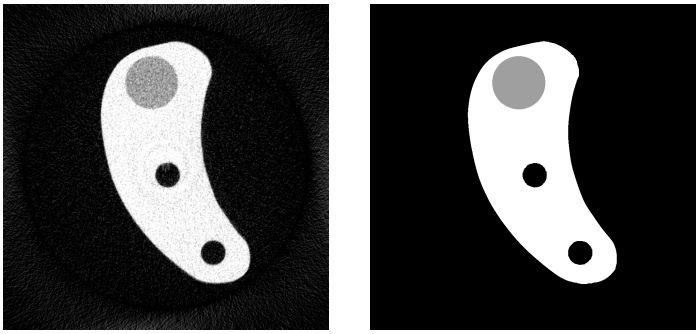
on noisy datasets depends strongly on the fix probability, we run DART for two values of the fix probability.

From the results in Fig. 4.12 we see that increasing the number of projection angles decreases the pixel error for DART, BART and the segmented SIRT reconstruction. Especially for the cylinder phantom, a large decrease in the pixel error can be achieved. In each case, SDART is still more accurate in comparison to DART and BART. However, there are two surprising results shown in this plot. First of all, DART with a fix probability of 0.5 is less accurate than DART with a fix probability of 0.99. From Fig. 4.11, we see that this is indeed the case for most noise levels. Apparently this behavior changes if the noise level is very high. Secondly, the pixel errors of SDART are not decreasing monotonically with an increasing number of projection angles. In fact, the pixel error is increasing for the blob phantom. The weights of SDART-NB were determined for a dataset at a fixed number of projections. Therefore, the number of projections is not incorporated in the weights. These results suggest that such an approach should be taken in order to avoid this behavior. Since the focus of this chapter is on discrete tomography for the limited angle case, we leave this open for further research. We want to emphasize that the weights defined in Eq. (4.10) do work well in the limited angle case, which is the domain of discrete tomography.

The BART algorithm performs more consistently as the pixel error drops when the number of projection angles is increased. For the Shepp–Logan phantom, the pixel error of SDART-NB is more or less constant. We expect that there is still significant room for improvement of the performance of SDART for cases where a relatively large number of projection angles are available, by further exploring the possible choices for the matrix $\boldsymbol{D}$.

### 4.5.4 Experiment IV – experimental data

To validate the simulation experiments, we applied SDART to an experimental dataset. For this experiment we used a hardware phantom. The hardware phantom consists of plexiglass and has a nearly convex shape with three holes drilled through it in the vertical direction. A SIRT reconstruction of the central



(a) SIRT Reconstruction (b) segmentation

Figure 4.13: (a) A SIRT reconstruction from all 600 projection images; (b) Segmentation of the SIRT reconstruction. The segmentation was manually improved on the edges.

slice is shown in Fig. 4.13a. The large hole was filled with white spirit, the middle hole with water and the last hole contains air. Note that air and water have very similar attenuation coefficients, resulting in almost identical gray values in the reconstruction.

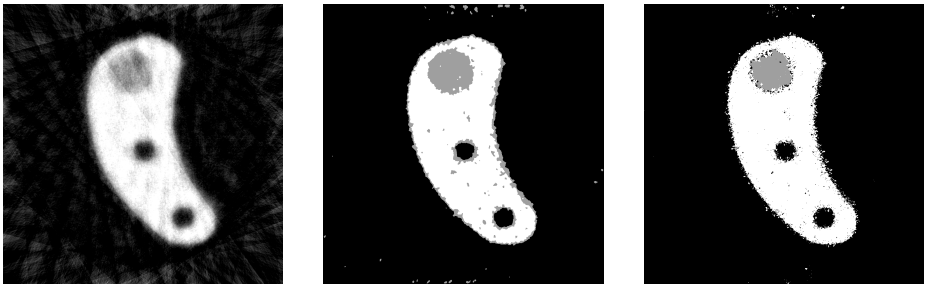
The projection data were acquired by a Skyscan 1172 microtomography X-ray scanner. A total of 600 projection images of $1000 \times 600$ pixels were acquired over a full tilt-range of 360°. The tilt increment between images was 0.6°. With a pixel size of $25\,\mu$m, a slice ($1000 \times 1000$ pixels) has physical dimensions of $25 \times 25$ mm.

The scanner has a cone beam geometry, however, since the object is uniform in the vertical direction, we focus on reconstructing the central slice using fan beam geometry. The sinogram was extracted from the projection images.

The SIRT reconstruction from all 600 projections leads to an accurate segmentation of the object (by thresholding). In Fig. 4.13b this segmentation is shown. Manual adjustments were applied to the segmentation at the boundaries, where the segmentation was distorted by noise. The gray values for this dataset were estimated using the algorithm proposed by Batenburg *et al.* [BAS11]. The segmentation can be used as a ground truth. Although it will differ from the actual ground truth, we can assume that it is reasonably accurate to allow also quantitative analysis of the accuracy of SDART.

For the (S)DART reconstructions we use a subset of 20 projections, with equiangularly distributed projection angles in the range $[0, \pi)$. The value of $\lambda = 2$ in SDART is computed based on the residual norm using the full dataset, as described in Section 4.4.2. The number of intermediate ARM iterations in DART is 20, and the number of DART iterations is 300. For SDART-NB, we use 70 intermediate iterations and a total of 30 SDART iterations. The difference in iterations between SDART and DART is large, because DART converges slowly in terms of iterations, but iterations are fast. SDART converges quickly, but iterations are far more costly.

In Fig. 4.14 we see the final SIRT, DART and SDART reconstructions. The



(a) SIRT (b) DART (c) SDART

Figure 4.14: The dataset considered here consists of a subset of 20 projections (from 600 total); (a) The final SIRT reconstruction; (b) The final DART reconstruction; (c) The final SDART reconstruction.

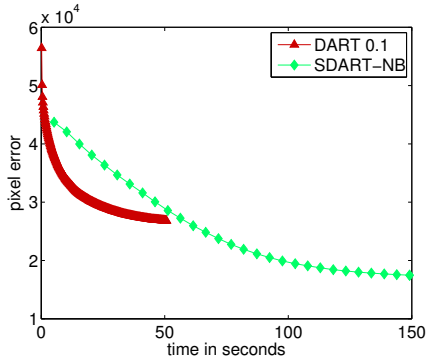



Figure 4.15: The pixel error of DART and SDART are plotted against the computation time. Note that DART requires far more iterations, but SDART takes three times as long to converge.

SIRT reconstruction is very noisy. When compared to the DART and SDART reconstructions the advantage of using prior knowledge is obvious. Although the DART and SDART reconstructions look similar, we can also point out clear differences. In the DART reconstruction, many gray areas are visible (white spirit) in the whiter area (the plexiglass). These form large clusters of pixels. SDART shows these clusters as well, but they are very small in comparison, just a few pixels. The result is that SDART's boundaries are sharper, but they look distorted by salt and pepper noise. DART's boundaries are smoother, but the presence of gray areas might lead to the erroneous conclusion that there is actually liquid there, instead of plexiglass.

In Fig. 4.15 we plotted the pixel error versus the computation time. Note that we do not have the actual pixel error, but we compare the reconstructions with the segmentation from Fig. 4.13b. The DART reconstruction has a pixel error of 2.68% and the pixel error of SDART is 1.74%. This is a significant decrease

in the pixel error of 35%. In terms of computation time we see that DART finishes in one third of the time it takes SDART to converge although it requires substantially more iterations. We should note that the DART implementation used here was optimized. It should be possible to speed up SDART by moving more computations, such as the detection of the boundaries, to the GPU. We want to emphasize that we introduce SDART here as a method that is more robust than DART on noisy data. Therefore we did not focus on computational efficiency, but rather on accuracy. From the results in Fig. 4.14 we see that the SDART reconstruction provides significantly different reconstructions. These may help in better understanding the true morphological nature of the object.

## 4.6 Conclusions

We proposed a new variant of DART that introduces a set of soft constraints to replace the hard constraints. We have seen that the hard constraints in DART lead to problems if the projection data contain a high level of noise. Our new method, named SDART, was introduced to enhance the robustness of DART for noisy projection data. The soft constraints allow noise to be spread across the whole image domain. As a result, boundaries of the object are less influenced by the noise, leading to sharper edges.

Two sets of soft constraints, or penalty matrices, were introduced. The first variant, SDART-ORIG, mimics the original DART algorithm. The other variant, called SDART-NB, discriminates boundary pixels by the number of surrounding pixels with a different gray value.

We performed several simulation experiments that compare the accuracy of SDART with DART, BART and SIRT. The results from noiseless data show that SDART has similar, but slightly less accuracy compared to DART. The accuracy of SDART-NB compared to BART is slightly improved. On datasets with very low signal-to-noise ratios, SDART-NB outperforms DART and BART by large. The results of SDART-ORIG were not accurate in this case and SDART-NB is the preferred method for noisy projection data. The qualitative results show that SDART-NB is less prone to salt and pepper noise. Results from experimental data, containing a large amount of noise, further support that SDART-NB is more accurate compared to DART. For this particular dataset, the pixel error of the SDART-NB reconstruction was approximately 35% smaller than that of DART. In this case, the difference in quality between the DART and SDART-NB reconstructions was less obvious visually. The SDART-NB reconstruction has sharper edges, distorted by some salt and pepper noise. DART produces clusters of a gray value on the edges that is different from the plexiglass interior. This might lead to the false conclusion that liquid (white spirit) adhered to these edges. From the SDART reconstruction it is clear that this is not the case. This shows that the SDART reconstruction can give additional insight, when conclusions about the DART reconstruction are not decisive.

So far, we have investigated only two possible choices for the penalty matrix $\boldsymbol{D}$. Compared to DART, SDART can encode a more specific representation of the prior

by using continuous weights. We expect that SDART can be further improved by using more sophisticated choices for this matrix, which will be investigated in future research.

# Chapter 5

# Analysis and removal of offset and scaling artifacts in tomography

## 5.1 Introduction

Tomography is an imaging technique for determining three dimensional structures from two dimensional projection images. An object is illuminated, from various angles, by an X-ray or electron source and the unabsorbed intensity is recorded by a detector. After the projection acquisition, a *reconstruction algorithm* is applied to generate a 3D volume from the projection data. This volume can be interpreted as a stack of grayscale images, where the *gray values* are proportional to the attenuation of the corresponding materials in the physical object.

In X-ray tomography contrast is obtained through absorption of X-ray photons. Each material in the sample attenuates the X-ray beam, quantified by its thickness and attenuation coefficient. The path traveled through the sample and the materials on this path determine the photon count observed on the detector.

In electron tomography, a beam of electrons in vacuum is used to generate contrast (in an electron microscope) instead of X-rays. Electrons have a different interaction with matter. Part of the signal is scattered, either elastically or inelastically. Another part of the signal, referred to as direct beam, has no interaction with the material and passes freely through the (often very thin) sample. In *Transmission Electron Microscopy* (TEM), part of the transmitted electron signal is recorded. The most common imaging technique records the *bright field* image, which is the direct beam that is focussed on an imaging plane. Another approach is *Scanning* TEM (STEM), where the incoming electron wave is focussed on a spot. The sample is then *scanned* by moving this spot across the sample.

One of the major challenges within the field of tomography is the ability of obtaining *quantitative* information from the reconstructed projection images, concerning the size, shape and density of the 3D structures in the object. In order to do so, an additional, and currently subjective, segmentation step is required after the reconstruction to determine the correspondence between gray values in the reconstruction and different compositions in the original structure.

Quantitative interpretation of tomographic reconstructions is often hampered by the presence of *reconstruction artifacts*: structured distortions of the reconstructed volume. Most of these artifacts belong to one of the following categories:

- **Artifacts caused by structural data errors introduced during acquisition.** This category includes artifacts caused by nonlinear effects of the image formation process, such as diffraction contrast, as well as misalignment between the images in the projection data that cannot be fully corrected. Another common source is from detector inefficiencies, which cause ring artifacts.
- **Artifacts caused by the limited amount of measured data.** This category includes *truncation artifacts* that are introduced when the sample extends beyond the field-of-view of the detector, as well as *missing wedge artifacts* common in electron tomography, caused by the limited angular range of the microscope.

These artifacts cause subsequent segmentation problems.

This chapter deals with three types of similar reconstruction artifacts which mainly fall in the first category and are caused by:

- a *global offset* present in the projection data: a constant that has been added to each pixel in every projection image,
- a *local offset* on the projection data: a different constant added to each projection image,
- a *scaling* of the projection data: the intensity scaling of the projection images changes with each projection angle.

Note that we assume the projection images are linearized, meaning that a value of 0 for a pixel in the projection data should correspond to a line that only passes through free space and does not intersect with the object. Higher values correspond to lines that do intersect with the object. We will go into more detail about this in Section 5.2. In practice, the zero level of the projection data is affected by various acquisition parameters of the scanner or fluctuations in the radiation source intensity, which may result in an offset on the data. Manipulating the dataset using various image processing tools, *e.g.*, to align the projection images before reconstruction, may also result in offsets on the projection data.

A common approach to deal with a global offset in electron tomography is to subtract the minimum value of the projection data, which corresponds to the background intensity, if the background is visible in any of the projection images [Gon15]. However, this approach is not feasible the object is larger than the field of view of the detector (no background visible), or in case of a local offset. Therefore, in this chapter we introduce a method that can also be used in the latter, more challenging situation.

In this chapter we analyze the effect of a global offset on the reconstruction obtained by filtered backprojection and derive the reconstruction artifact analytically. Then we introduce an iterative algorithm for estimation of a global offset,

based on algebraic reconstruction methods. Similar algorithms are derived for estimation of a local offset and scaling factors. In a series of simulation experiments we study the effects of certain reconstruction parameters on the estimation algorithms, such as the shape of the reconstruction area. Finally we apply the local and global offset estimation algorithms to an experimental dataset.

This chapter is structured as follows: in Section 5.2, offset and scaling artifacts are introduced and their causes are explained. In Section 5.3 we determine analytically the effect of a global offset on the reconstruction. In Section 5.4, we present an algorithm for estimating offsets based on the available projection data. The estimated offset can then be subtracted from the data before reconstruction, resulting in the reduction, or even removal, of offset artifacts. In Section 5.5 we present an algorithm for estimation of scale factors of the projection data. Section 5.6 presents a series of simulation experiments, aimed at validating the proposed estimation algorithms. The simulation experiments are followed by one example where the proposed local offset estimation algorithm has been applied to experimental datasets. Section 5.7 provides a discussion and concludes this chapter.

## 5.2 Origin of offsets and scalings in projection data

In this section we briefly introduce tomography, followed by a description of phenomena that introduce offsets or scaling in the acquired projection images. These offsets or scale factors can be introduced by physical instrument effects and by image manipulation after data acquisition.

### 5.2.1 Tomography

First we describe the mathematical model used for the reconstruction problem. Fig. 4.1 shows a schematic view of the parallel beam acquisition geometry for tomography. A detector measures the intensity of the radiation (*e.g.* X-ray) emitted from the source along a straight line

$$\ell_{\theta,t} = \{(x,y) : x\cos\theta + y\sin\theta = t\}$$

where $\theta$ indicates the rotation angle (with respect to the object) and $t$ denotes the position of the detector pixel. Essentially each detector measurement approximates a line integral along the line $\ell_{\theta,t}$

$$p_\theta(t) := \int_\ell f(x,y)\,\mathrm{d}s.$$

where the object is presented by an image function $f(x,y)$. This line integral can be rewritten by parameterization of the line $\ell$ using the Dirac delta $\delta(\cdot)$ function, which leads to the *Radon transform* [NW01],

$$p_\theta(t) := \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - t)\,\mathrm{d}x\,\mathrm{d}y.$$

To reconstruct the image function $f(x,y)$ from the projections $p_\theta(t)$ the Radon transform should be inverted, which can be achieved by the *filtered backprojection* method (FBP) [KS01], provided that infinitely many projections are acquired for the entire interval $\theta \in [0,\pi)$. In practice, due to a finite number of projections and discretization of the data, an inexact inversion is obtained using FBP.

In this chapter we focus on *algebraic* or *iterative reconstruction methods* which are based on a discretization of the image function $f(x,y)$ in pixels and of the projection images $p_\theta(t)$. The image is represented as vector $\boldsymbol{x} \in \mathbb{R}^N$ and so is the projection data $\boldsymbol{p} \in \mathbb{R}^M$. A single detector pixel measurement can be seen as a combination of the pixel values and the contribution of pixel $x_j$ to detector pixel $p_i$ is determined by the weight $w_{ij}$:

$$p_i = \sum_j w_{ij} x_j,$$

which we call the *ray sum*. The full set of equations leads to the following linear system:

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p} \tag{5.1}$$

where $\boldsymbol{W} \in \mathbb{R}^{M\times N}$ is called the *projection matrix*. Algebraic reconstruction methods such as SIRT, ART and LSQR typically compute a (weighted) least squares solution of Eq. (5.1) [KS01; GBH70; PS82].

### 5.2.2 Beer–Lambert's law

In absorption contrast tomography, the intensity of the radiation $I$ that is measured on the detector is related to the intensity of the source $I_0$ by the Beer–Lambert law [IC88]:

$$I = I_0 \mathrm{e}^{-\int_L \mu(\ell)\,\mathrm{d}\ell} \tag{5.2}$$

assuming a monochromatic beam, where $\int_L \mu\,\mathrm{d}l$ is the line integral of the attenuation coefficient in the direction of the ray. Therefore, the measured projection data are normalized prior to the reconstruction:

$$\int_L \mu(\ell)\,\mathrm{d}\ell = -\log\left(\frac{I}{I_0}\right). \tag{5.3}$$

The source intensity is measured before scanning by taking an image without the object inside the beam. This so called *flat field* should be uniform, but often imperfections can be observed. Such imperfections can be caused by dust particles or nonlinearities of the detector pixels. By normalization (or *flat field correction*) such imperfections are partly removed.

In case the source intensity is not stable, the intensity of the flat field can vary with consecutive projections. However, the flat field correction will be applied using the initial source strength, which introduces an error. The recorded projections (in terms of photon counts) is

$$I = I_0 \lambda_k \mathrm{e}^{-\int_L \mu(\ell)\,\mathrm{d}\ell}, \quad k = 1,\ldots,K \tag{5.4}$$

where $\lambda_k$ describes the variations in the source intensity. After normalization, we obtain:

$$-\log\left(\frac{I}{I_0}\right) = \int_L \mu(\ell)\,\mathrm{d}\ell - \log(\lambda_k). \tag{5.5}$$

So the measured projections after normalization (line integrals) now include an offset that depends on the $k$-th projection angle.

In modalities such as HAADF-STEM (High Angular Annular Dark Field Scanning Transmission Electron Microscopy) approximations of the line integrals are measured directly:

$$I = I_0 \int_L \mu(\ell)\,\mathrm{d}\ell. \tag{5.6}$$

In this case, any variations in the flat field lead to multiplicative errors in the measurements:

$$\frac{I}{I_0} = \lambda_k \int_L \mu(\ell)\,\mathrm{d}\ell, \quad k = 1,\ldots,K.$$

Another common source for offsets is in the post-processing of the projection images, where several image manipulations may lead to a global scaling and offset of the projection data.

## 5.3 Analysis of global offset artifacts

To analyze the effect of a global offset on the reconstruction, we follow the analytical analysis that forms the basis of the filtered backprojection algorithm. An exact inversion formula for the Radon transform can be obtained as a composition of the following steps:

- Fourier transform of the projections:

$$P_\theta(u) := \mathscr{F}\{p_\theta\}(u) = \int_{-\infty}^{\infty} p_\theta(t) e^{-2\pi i t u}\,\mathrm{d}t$$

- Application of a ramp filter in the Fourier domain:

$$Q_\theta(u) := P_\theta(u)|u|$$

- Inverse Fourier transform of the filtered Fourier domain data:

$$q_\theta(t) := \mathscr{F}^{-1}\{Q_\theta\}(t) = \int_{-\infty}^{\infty} Q_\theta(u) e^{2\pi i t u}\,\mathrm{d}u$$

- Backprojection of the filtered projections:

$$f(x,y) = \int_0^\pi q_\theta(x\cos\theta + y\sin\theta)\,\mathrm{d}\theta$$

We consider a finite detector of length 1, therefore, the projection data are measured in the interval $t \in [-\frac{1}{2}, \frac{1}{2}]$. The region outside this interval is set to zero. We now investigate the result of an offset of 1 present in all projections. Ignoring the projections from the actual object (we focus on the artifact), we have, for any projection angle $\theta$,

$$p_\theta(t) = \operatorname{rect}(t) = \begin{cases} 1 & |t| \leq \frac{1}{2} \\ 0 & |t| > \frac{1}{2} \end{cases},$$

along with its Fourier transform

$$P_\theta(u) = \mathscr{F}\{p_\theta\}(u) = \operatorname{sinc}(u) = \frac{\sin(\pi u)}{\pi u}.$$

Applying a ramp filter to the projection data in the Fourier domain results in

$$Q_\theta(u) = P_\theta(u)|u| = \begin{cases} -\frac{\sin(\pi u)}{\pi} & u < 0 \\ \frac{\sin(\pi u)}{\pi} & u \geq 0 \end{cases},$$

which yields, after applying the inverse Fourier transform:

$$q_\theta(t) := \mathscr{F}^{-1}\{Q_\theta\}(t) = \frac{1}{(\frac{1}{2}-t)(\frac{1}{2}+t)2\pi^2}.$$

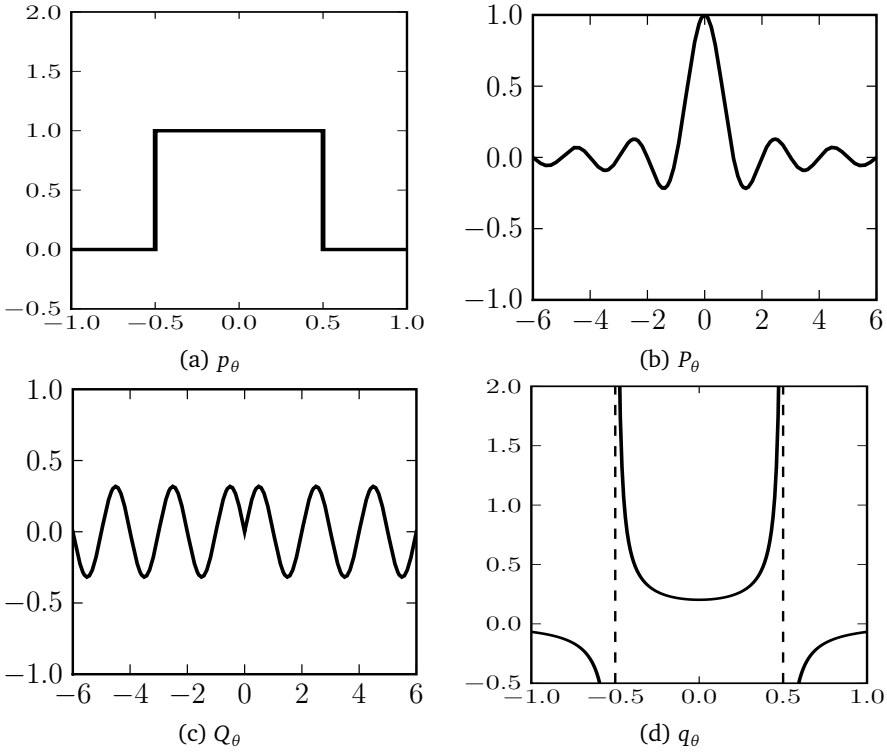
Fig. 5.1 shows graphs of the functions $p_\theta$, $P_\theta$, $Q_\theta$ and $q_\theta$.
The backprojection of $q_\theta(t)$ leads to

$$f(x,y) = \int_0^\pi \frac{1}{(\frac{1}{2}-t)(\frac{1}{2}+t)2\pi^2}\,\mathrm{d}\theta = \frac{4\pi}{2\pi^2\sqrt{1-4(x^2+y^2)}}$$

with $t = x\cos\theta + y\sin\theta$.

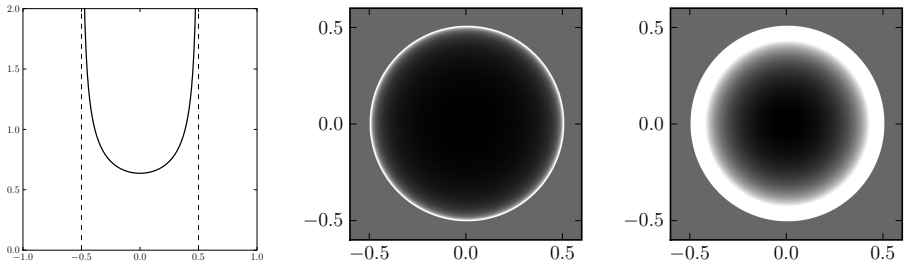
Two images of the function $f$ are shown in Figs. 5.2b and 5.2c. They are the same, except that they show a different interval of intensities. The bright ring around the reconstruction corresponds to the poles of $q_\theta$, at the edges of the detector. Fig. 5.2a illustrates that although the intensity variations within this circle are much smaller than at the boundary, the interior is not constant as well. Therefore, an offset on the projection data will result in a continuously varying, radially symmetric artifact in the reconstruction. Segmentation operations such as thresholding, that rely on the fact that similar structures have a gray level that is independent of their position in the sample, may therefore lead to erroneous results. The intensities of the offset artifact in the reconstruction are directly proportional to the value of the offset. For small offsets, its contribution is negligible, whereas major artifacts can be observed for large offsets with respect to the values of the actual projection data.

When using backprojection algorithms, such as filtered backprojection, the impact of offset artifacts is typically limited, as most of the structure of interest is contained in the slowly varying part of the offset intensity field. The effect of an offset on the projection data is more complicated for iterative reconstruction methods, such as ART or SIRT. Such algorithms reconstruct a certain *area*, where



(a) $p_\theta$ (b) $P_\theta$ (c) $Q_\theta$ (d) $q_\theta$

Figure 5.1: (a) Offset on the projection data; (b) Fourier transform of the offset; (c) Result of applying a ramp-filter in the Fourier domain (d) Offset on the projection data after filtering.



(a) $f(x,0)$ (b) $f(x,y)$, wide intensity scaling window (c) $f(x,y)$, narrow intensity scaling window

Figure 5.2: Filtered backprojection of an offset; (a) along the horizontal axis (b) full image: wide intensity window, which illustrates the bright circle formed at the edges of the detector (c) full image: narrow intensity window, which illustrates the gradual change in intensity in the interior of the circle.

the object function $f$ is assumed to be zero outside this area. Generally, it is advantageous to restrict this area as much as possible such that it still entirely contains the object. For flat objects, such as many electron microscopy samples, it is advantageous to reconstruct a flat rectangular area, instead of a full square. By using a rectangle, the entire area outside this rectangle is effectively constrained to be zero, which drastically reduces the number of unknowns in the reconstruction problem, thereby leading to a more accurate reconstruction. Fig. 5.3b shows a reconstruction computed by SIRT on a square which has been cropped, whereas Fig. 5.3c shows the same reconstruction computed by SIRT on a rectangle. The difference in quality can be clearly observed. Note that in this case no offset or scaling was applied to the projection data.

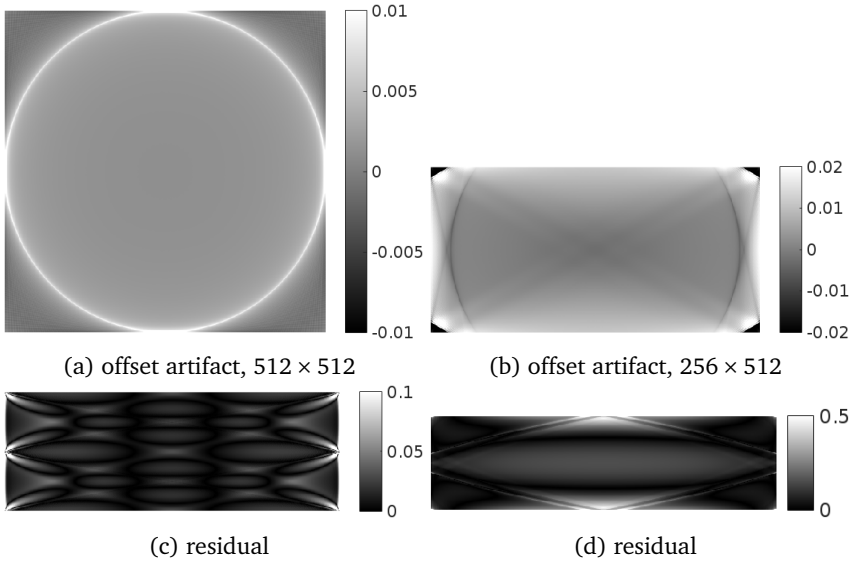
(a) flat phantom

(b) reconstruction on a square, cropped

(c) reconstruction on a rectangle

Figure 5.3: Comparison of two different reconstruction volumes for projection data without offset or scaling applied; (a) the flat phantom image of size $64 \times 512$; (b) a SIRT reconstruction on a square volume, $512 \times 512$, which has been cropped to the size of the phantom; (c) a SIRT reconstruction on a flat volume, $64 \times 512$.

The degrading effect of a projection data offset on the reconstruction becomes much stronger if the reconstruction area is made smaller than the outer circle of the basic offset artifact. This effect is demonstrated in Fig. 5.4, which shows two SIRT reconstructions of the offset on different reconstruction volumes. The $512 \times 512$ reconstruction corresponds to an offset of 1 on a detector 512 pixels wide for projection angles $\pm 90°$, with $1°$ increments. The $256 \times 512$ reconstruction corresponds to an offset of 1 on a detector 450 pixels wide for projection angles $\pm 60°$, also with $1°$ increments. SIRT was run for 100 iterations. The absolute values of the corresponding residual projections are shown in Figs. 5.4c and 5.4d, *i.e.* a forward projection of the reconstruction minus the original projections. Note that in Fig. 5.4a the size of the detector was 512 pixels, which means that the reconstruction size is just large enough to contain the full circular offset artifact. The corresponding residual is relatively small, see Fig. 5.4c, but some inconsistencies remain especially on the left and right sides. The circular artifact cannot be reproduced on a smaller reconstruction area of $256 \times 512$, as shown in Fig. 5.4b. Therefore, the residuals are larger in this case. The residuals in top and bottom of Fig. 5.4d are most prominent. So a numerical reconstruction of the offset artifact is not consistent (has nonzero residual), which is even more




(a) offset artifact, 512 × 512 (b) offset artifact, 256 × 512

(c) residual (d) residual

Figure 5.4: SIRT reconstructions of the offset artifact on two reconstruction volume sizes. The absolute value of the residual projections corresponding to the reconstructions of 512 × 512 and 256 × 512 are shown in (c) and (d) respectively. The residuals in (d) were truncated at 0.02.

pronounced if the reconstruction area is rectangular, where one of the dimensions of the reconstruction grid is smaller than the detector.

## 5.4 Offset estimation algorithm

To reduce, or even remove offset artifacts when using iterative reconstruction methods such as SIRT, the unknown offset must be estimated from the available projection data. It can then be subtracted from the data before applying the reconstruction algorithm. In some cases, one or more projection images contain a region that is not occupied by the sample, where the beam (collection of rays) only intersects with vacuum. An example of such a region is shown in Fig. 5.5. In Section 5.6.3 we describe this dataset in detail. In such cases, the offset, which we denote as $\lambda \in \mathbb{R}$, can be determined directly from the projection image, *e.g.*, by averaging the pixel values inside one or more of such regions or by simply determining the minimum value of the projections, as is done in [Gon15]. Here, we consider a more general case, where the offset cannot be estimated directly from the set of projection data.

The offset estimation problem becomes much more complicated when the entire field-of-view in all projection images is covered by the sample. If the structure of interest is contained within a supporting material of constant thickness,

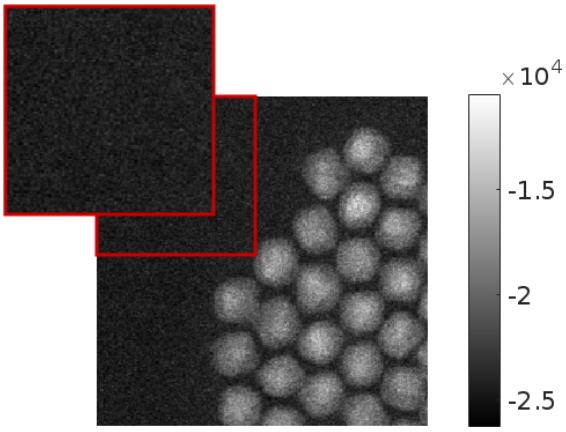



Figure 5.5: Background intensity of an experimental HAADF-STEM dataset, from EMAT (Electron Microscopy for Materials Science, Antwerp).

the thickness of the support in the direction of the beam is proportional to $\frac{1}{\cos\theta}$, where $\theta$ is the incidence angle of the beam. If regions can be identified in each projection image that contain only the supporting material, the offset can be estimated by fitting the function $\lambda + \frac{k}{\cos\theta}$ to the average projection values in these regions, where both the offset parameter $\lambda$ and the proportionality constant $k$ are estimated simultaneously.

### 5.4.1 Global offset estimation

It was shown analytically in Section 5.3 that a global offset on the projection data leads, after filtering with the ramp filter, to a filtered projection that tends to infinity at the boundaries of the detector. After backprojection, this results in a reconstruction that cannot be represented as a grid of finite pixel values. This is not exactly the case for the SIRT reconstructions of the global offset as shown in Fig. 5.4a, but still parts of the residual do not converge to zero and inconsistencies remain. If one dimension of the reconstruction grid is smaller than the detector width, these inconsistencies are more pronounced. Therefore, the presence of an offset in the projection data can lead to an *inconsistent* reconstruction problem: there exists no reconstruction that matches the data.

The result of applying a reconstruction algorithm to inconsistent projection data depends quite heavily on the particular reconstruction algorithm. Here, we restrict the discussion to the case of SIRT as it allows for a clear mathematical analysis.

We denote the SIRT reconstruction of a vector $\boldsymbol{p}$ of projection data by $\mathrm{S}(\boldsymbol{p})$. SIRT converges to a reconstruction $\bar{\boldsymbol{x}} = \mathrm{S}(\boldsymbol{p})$ for which

$$\|\boldsymbol{W}\bar{\boldsymbol{x}} - \boldsymbol{p}\|_{\boldsymbol{R}}$$

is minimal, where $\|\boldsymbol{x}\|_{\boldsymbol{R}} = \sqrt{\boldsymbol{x}^{\mathsf{T}}\boldsymbol{R}\boldsymbol{x}}$ denotes a norm based on a weighted sum of squares [GB08].

So, SIRT converges to a reconstruction that corresponds as closely as possible with the given projection data. If the data is inconsistent, due to noise or an offset on the data, this property will still hold, but the *projection distance*,

$$d(\boldsymbol{x}, \boldsymbol{p}) = \|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{p}\|_2,$$

will become larger. This property can be used to estimate the offset, by computing the offset for which $d(\mathrm{S}(\boldsymbol{p}_\tau), \boldsymbol{p}_\tau)$ is minimal, where $\boldsymbol{p}_\tau = \boldsymbol{p} - \tau\boldsymbol{e}$. Each element of the vector $\boldsymbol{e}$ is 1 (where $\boldsymbol{e}$ has the same length as $\boldsymbol{p}$). Note that $\boldsymbol{p}$ is the set of recorded projection data including an unknown offset and $\boldsymbol{p}_\tau$ is the projection data with a correction term for the global offset. This leads to the following formal model of the offset estimation problem:

$$\underset{\tau}{\text{minimize}}\ \|\boldsymbol{W}(S(\boldsymbol{p}_\tau)) - \boldsymbol{p}_\tau\|_2 \tag{5.7}$$

Computing $d(S(\boldsymbol{p}_\tau), \boldsymbol{p}_\tau)$ for a single offset requires the computation of a SIRT reconstruction, which may take considerable time. Searching an entire interval of potential offsets $\tau$ is computationally unfeasible. Fortunately, Eq. (5.7) can be solved by computing only *two* SIRT reconstructions, by exploiting the linearity of the SIRT algorithm.

As demonstrated in [KS01], every iteration of the SIRT algorithm performs a linear transformation on the output of the previous iteration. Therefore, the result after a finite number of SIRT iterations is the *composition* of a finite number of linear transformations, which is again a linear transformation:

$$\mathrm{S}(\lambda\boldsymbol{p} + \gamma\boldsymbol{b}) = \lambda\mathrm{S}(\boldsymbol{p}) + \gamma\mathrm{S}(\boldsymbol{b}).$$

The forward projection operation $\boldsymbol{W}$ is also a linear transformation. Therefore, we can write the objective of the minimization problem as:

$$\begin{aligned} d(S(\boldsymbol{p}_\tau), \boldsymbol{p}_\tau) &= \|\boldsymbol{W}(\mathrm{S}(\boldsymbol{p} - \tau\boldsymbol{e})) - (\boldsymbol{p} - \tau\boldsymbol{e})\|_2 \\ &= \|\boldsymbol{W}\mathrm{S}(\boldsymbol{p}) - \tau\boldsymbol{W}\mathrm{S}(\boldsymbol{e}) - \boldsymbol{p} + \tau\boldsymbol{e}\|_2 \\ &= \|\tilde{\boldsymbol{p}} - \tau\tilde{\boldsymbol{e}}\|_2, \end{aligned} \tag{5.8}$$

where $\tilde{\boldsymbol{p}} = \boldsymbol{W}\mathrm{S}(\boldsymbol{p}) - \boldsymbol{p}$ and $\tilde{\boldsymbol{e}} = \boldsymbol{W}\mathrm{S}(\boldsymbol{e}) - \boldsymbol{e}$. The vector $\tilde{\boldsymbol{p}}$ corresponds to the difference between the measured data (including the offset) and the computed projections based on its SIRT reconstruction. The vector $\tilde{\boldsymbol{e}}$ corresponds to the difference between an offset of 1 and the computed projections based on its SIRT reconstruction. Note that the expression in Eq. (5.8) is minimal if $\tilde{\boldsymbol{e}}$ and $\tilde{\boldsymbol{p}} - \tau\tilde{\boldsymbol{e}}$ are perpendicular. That is, we can compute $\tau$ by a vector projection of $\tilde{\boldsymbol{p}} - \tilde{\boldsymbol{e}}$ onto $\tilde{\boldsymbol{e}}$:

$$\tau = \frac{\tilde{\boldsymbol{p}} \cdot \tilde{\boldsymbol{e}}}{\tilde{\boldsymbol{e}} \cdot \tilde{\boldsymbol{e}}}, \tag{5.9}$$

where we assume that $\tilde{\boldsymbol{e}} \neq \boldsymbol{0}$, meaning that the SIRT reconstruction of the offset is inconsistent (has nonzero residual). If $\tilde{\boldsymbol{e}} = \boldsymbol{0}$, then Eq. (5.8) is independent of $\tau$ and we cannot retrieve it in this manner.

To compute $\tilde{\boldsymbol{e}}$ and $\tilde{\boldsymbol{p}}$ we indeed need only two SIRT reconstructions. However, there is an alternative approach for solving Eq. (5.7) that is more efficient. First note that the following equation

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p} - \tau\boldsymbol{e},$$

is consistent if $\tau$ equals the true offset on the projections $\tau^*$, where

$$\boldsymbol{p} = \boldsymbol{p}^* + \tau^*\boldsymbol{e},$$

and $\boldsymbol{p}^* = \boldsymbol{W}\boldsymbol{x}^*$ are the projections of the ground truth image $\boldsymbol{x}^*$. If we move the offset correction term to the left-hand side, we obtain the following linear system:

$$\begin{aligned} \boldsymbol{W}\boldsymbol{x} + \tau\boldsymbol{e} &= \boldsymbol{p}, \\ [\boldsymbol{W}, \boldsymbol{e}]\begin{bmatrix}\boldsymbol{x}\\ \tau\end{bmatrix} &= \boldsymbol{p}, \end{aligned} \tag{5.10}$$

which can be solved by a least squares solver. In this way, the reconstruction and the offset parameter can be estimated simultaneously, which reduces the amount of computations substantially when compared to Eq. (5.9). In our experiments of section Section 5.6 we use the least squares method LSQR [PS82] to solve Eq. (5.10).

Note that the solutions found by Eq. (5.9) and Eq. (5.10) are inherently different. The vector projection method using SIRT solves the problem in two optimization steps:

$$\bar{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{p}_\tau\|_{\boldsymbol{R}},$$

and subsequently

$$\underset{\tau}{\text{minimize}}\ \|\boldsymbol{W}\bar{\boldsymbol{x}} - \boldsymbol{p}_\tau\|_2.$$

Note that the SIRT reconstruction $\bar{\boldsymbol{x}}$ does not necessarily have a minimum residual in the $\ell_2$-norm, due to the weighted norm $\|\cdot\|_{\boldsymbol{R}}$. Also, SIRT is known to converge slowly and might be terminated early in practice. The method using LSQR applied to the system in Eq. (5.10) solves the following optimization problem,

$$\underset{\boldsymbol{x},\tau}{\text{minimize}}\ \|\boldsymbol{W}\boldsymbol{x} + \boldsymbol{p}_\tau\|_2.$$

Furthermore, LSQR has the property that it computes the smallest norm solution, *i.e.*, it finds a solution for which $\|(\boldsymbol{x}^\intercal, \tau)\|_2$ is smallest. In Section 5.6.1 we will explore the difference between these two approaches.

### 5.4.2 Local offset estimation

The same idea can be applied if the offset is different for every projection. First we order all equations corresponding to a each projection angle:

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p} - \begin{pmatrix} \lambda_1\boldsymbol{e} \\ \lambda_2\boldsymbol{e} \\ \vdots \\ \lambda_K\boldsymbol{e} \end{pmatrix}, \tag{5.11}$$

where $\boldsymbol{e}$ is a column vector of $D$ ones, where $D$ is the number of pixels per projection image. If we move the offset correction to the left-hand side, the linear system is written as:

$$\begin{pmatrix} \boldsymbol{W}_1 \\ \boldsymbol{W}_2 \\ \vdots \\ \boldsymbol{W}_K \end{pmatrix} \boldsymbol{x} + \begin{pmatrix} \lambda_1 \boldsymbol{e} \\ \lambda_2 \boldsymbol{e} \\ \vdots \\ \lambda_K \boldsymbol{e} \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{p}_K \end{pmatrix} = \boldsymbol{p},$$

which leads to:

$$\begin{pmatrix} \boldsymbol{W}_1 & \boldsymbol{e} & & \\ \boldsymbol{W}_2 & & \boldsymbol{e} & \\ \vdots & & & \ddots \\ \boldsymbol{W}_K & & & \boldsymbol{e} \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_K \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{p}_K \end{pmatrix}. \tag{5.12}$$

The least squares solution will yield both a vector of offsets and the corresponding reconstructed image.

## 5.5 Scale estimation algorithm

As discussed in Section 5.2, for certain projection acquisition methods, source intensity fluctuations can lead to a scaling of projection data. An approach similar to the local offset estimation can be applied to the estimation of scalings. However, some modifications are necessary to ensure the estimation algorithm works reliably in this case. In this section we discuss these modifications.

Note that a scaling $\alpha$ applied to all projection images results in a scaling of the gray values in the SIRT reconstruction:

$$\mathrm{S}(\alpha \boldsymbol{p}) = \alpha \mathrm{S}(\boldsymbol{p}).$$

due to the linearity of SIRT. The same holds for the solution of least squares methods:

$$\underset{\boldsymbol{x}}{\text{minimize}} \, \|\boldsymbol{W}\boldsymbol{x} - \alpha \boldsymbol{p}\|_2^2 \equiv \underset{\boldsymbol{y}}{\text{minimize}} \, \alpha^2 \|\boldsymbol{W}\boldsymbol{y} - \boldsymbol{p}\|_2^2 \tag{5.13}$$

for $\boldsymbol{x} = \alpha \boldsymbol{y}$. Therefore, if the projection vector $\boldsymbol{p}$ is scaled by a single factor, *i.e.*, a *global scaling*, we cannot determine this scaling $\alpha$ based on the residuals of a reconstruction (obtained by SIRT, LSQR, or another method). Both the original and scaled variants of the projections have the same residual after reconstruction, up to a scaling. Note that if each projection image is scaled by a different factor, there is no reconstruction that matches this projection data, because a reconstruction pixel would have different projections (in intensity) depending on the projection angle. Therefore, we can fix this *local scaling* and change it to a global scaling where artifacts in the reconstruction are removed.

Let $\kappa_1^*, \ldots, \kappa_K^*$ be a set of scale factors applied to the original projections $\boldsymbol{p}^*$ (which is in the column space of $\boldsymbol{W}$ in Eq. (5.1)):

$$\begin{pmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_K \end{pmatrix} = \begin{pmatrix} \kappa_1^* \boldsymbol{p}_1^* \\ \vdots \\ \kappa_K^* \boldsymbol{p}_K^* \end{pmatrix},$$

The goal is to retrieve these scale factors from the scaled projection data $\boldsymbol{p}$.

Let $\sigma_1, \sigma_2, \ldots, \sigma_K$ be a set of scale factors which we use to correct the scaled projections,

$$\boldsymbol{W}\boldsymbol{x} = \begin{pmatrix} \sigma_1 \boldsymbol{p}_1 \\ \sigma_2 \boldsymbol{p}_2 \\ \vdots \\ \sigma_K \boldsymbol{p}_K \end{pmatrix}, \tag{5.14}$$

*i.e.*, $\sigma_i = 1/\kappa_i^*$ is optimal, where we assume that $\kappa_i \neq 0$ for any $i = 1, \ldots, K$. Note that the estimation problem in Eq. (5.14) has a trivial solution, $\boldsymbol{x} = \boldsymbol{0}$ and $\sigma_i = 0$, for $i = 1, \ldots, K$. If each projection is scaled by zero, the projection data is consistent to a reconstruction that is zero everywhere. The least squares solver LSQR applied to Eq. (5.14) finds the smallest norm solution $\|(\boldsymbol{x}^\intercal, \sigma_1, \ldots, \sigma_K)\|_2$. Therefore, it will converge to this trivial solution. To avoid the trivial solution we introduce the parameter:

$$\tau_i = 1 - \sigma_i.$$

The introduction of this parameter leads to the following linear system, after we move the unknowns to the left-hand side:

$$\begin{pmatrix} \boldsymbol{W}_1 & \boldsymbol{p}_1 & & & \\ \boldsymbol{W}_2 & & \boldsymbol{p}_2 & & \\ \vdots & & & \ddots & \\ \boldsymbol{W}_K & & & & \boldsymbol{p}_K \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \tau_1 \\ \tau_2 \\ \vdots \\ \tau_K \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{p}_K \end{pmatrix}. \tag{5.15}$$

We use LSQR to solve Eq. (5.15). If the parameter $\tau_i$ tends to zero, due to LSQR finding a smallest norm solution, the corresponding scaling $\sigma_i$ tends to 1, which corresponds to projections that are not scaled. In this way, the trivial solution of zeros is avoided and the least squares solution is close to unity, which is reasonable if we assume that the scale factors are close to 1.

## 5.6 Experiments and results

In this section we perform a series of simulation experiments on 2D and 3D simulation datasets. We quantify the accuracy of our method for retrieving global and local offsets and scaling factors of projections. Finally we apply our proposed method on an experimental dataset obtained by HAADF STEM microscopy which exhibits offset and/or scaling artifacts.

### 5.6.1 Slice-based simulation experiments

In this section we compare results of the global offset, local offset and scale estimation algorithms. If we refer to a global offset, local offset or scale problem then we have applied one of the following to the projection data (not combined and unless specified otherwise):

- a global offset of 100,
- a random local offset sampled uniformly from $(5, 55)$,
- a linear scaling uniformly distributed from 0.1 to 2 (not random).

Note that for the phantom that we consider, shown in Fig. 5.7, the average intensity of the projection data without offsets is approximately 60. The gray values in the ground truth are 0.5 (background) and 1 for the object. Projection data is generated for the angles $\pm 60°$, with $1°$ increments, by using the projection matrix from Eq. (5.1). We use the ASTRA toolbox to generate the projection matrix on-the-fly using the CPU [PBS13], without storing the matrix elements for memory considerations. The strip model is used to generate the projection matrix, the matrix elements are based on the area of intersection between one ray (part of the beam corresponding to a detector pixel) and a pixel [Zhu+08].

Note that we have two different ways to estimate a global offset: one involves computing a vector projection from Eq. (5.9) using two SIRT reconstructions, the other involves solving Eq. (5.10) using LSQR. In Fig. 5.4, we investigated the effect of the height of the reconstructed volume on the SIRT reconstruction of a global offset. The height is defined as the $y$-component of the reconstruction volume, as shown in the schematic of the geometry in Fig. 5.6. It seems that on a square reconstruction area, with sides as large as the detector, the residual of the offset artifact is close to zero. Therefore, we expect that it is difficult to recover offsets in this case by minimizing the residual compared to a case where the reconstruction domain is rectangular.

Note that in practice a reconstruction volume might not be particularly flat. First of all, the sample might not be very thin (size in the $y$-direction of Fig. 5.6). Secondly, if the sample thickness is only known approximately, it is safer to have a reconstruction height that is slightly larger than the thickness of the sample. If the height of the reconstruction area is smaller than the actual thickness of the sample, severe artifacts will be generated on the upper and lower boundaries of the reconstruction. Therefore, in the first experiment we investigate the effect of the height of the reconstruction area on the accuracy of the offset and scale estimation algorithms.

We apply the proposed estimation algorithms to the corresponding datasets described previously. We used a detector size of 310 pixels to generate the projections and a reconstruction size of $64 \times 512$. LQSR applied to global offset estimation (Eq. (5.10)), is iterated 250 times. SIRT for computing the vector projection for determining a global offset (Eq. (5.9)), is iterated 250 times. LSQR applied to local offset estimation (Eq. (5.12)), is run for 300 iterations. LSQR applied to the scale estimation algorithm (Eq. (5.15)), is iterated 400 times.

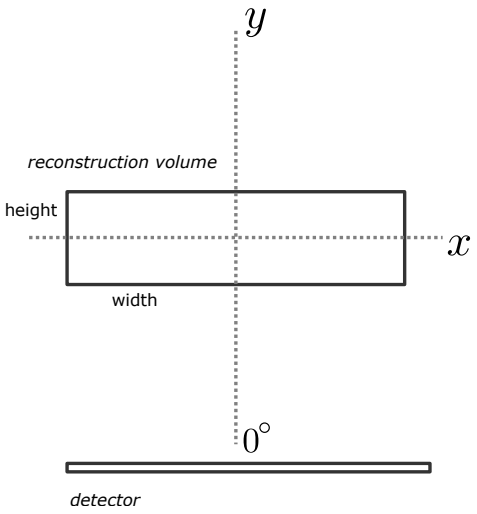



Figure 5.6: Representation of the geometry for flat reconstruction areas. The detector is placed at $\theta = 0°$, and the rotation axis is the $z$-axis (perpendicular to $x$- and $y$-axes). Note that in 3D the reconstruction volume and detector also have a component in the $z$-direction. Typical rotation angles are $\theta \in [-s_\theta, +s_\theta]$ and $s_\theta < 90°$.

Figure 5.7: Ground truth image of size $64 \times 512$, representing a part of a slice of a cylinder block.

The relative errors are shown in Fig. 5.8, meaning relative to the true offset/scaling:

$$|\boldsymbol{\lambda} - \boldsymbol{\lambda}_{\text{true}}|/|\boldsymbol{\lambda}_{\text{true}}|.$$

As can be seen from Fig. 5.8 the height of the reconstruction has a significant impact on the accuracy of these methods. Up to 50% of the image width, the accuracy is still acceptable, but for larger heights some accuracy is lost. This is probably due to the effect we saw in Fig. 5.4, where the smaller reconstruction area has larger residuals from the global offset artifact. Therefore, on a smaller reconstruction area, an improvement in the global offset estimation has a larger reduction of the residual. This might explain why the global offset estimation is most accurate if the reconstruction domain matches the phantom size. A similar reasoning can be applied to the accuracy of the local offset estimation. These results suggest that the reconstruction height is not very crucial for the offset retrieval, provided that the reconstruction hight is smaller than 60% of the reconstruction width.

We should note that we cannot directly compare the recovered scaling and the true scaling of the projection data, because we can only find it up to a global

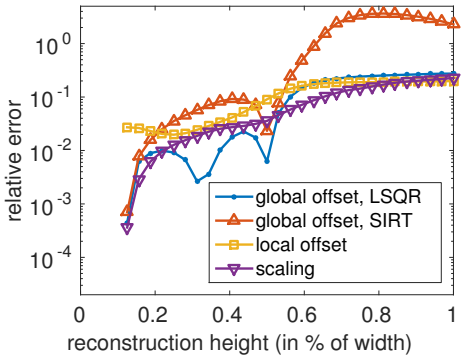


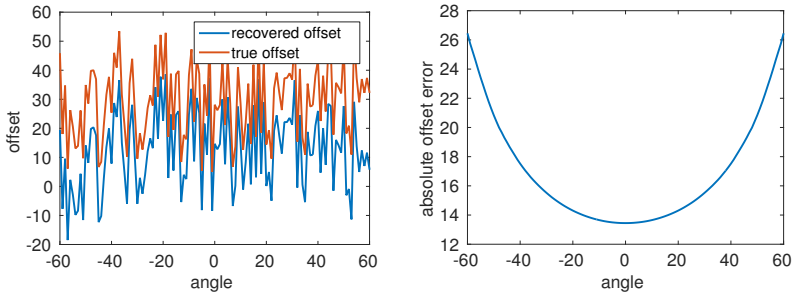

Figure 5.8: Influence of the height of the reconstruction volume on the offset and scale estimation algorithms. The relative error is shown of the recovered offset and scale factors with respect to the true offset and scale factors.

scaling. Therefore, we remove this common global scaling between the recovered scaling and true scaling for computing the relative error. A correction should also be applied to the error of the local offset. In Fig. 5.9a we plotted the recovered local offset and the true local offset. Note that it seems the difference between the two is a global offset. However, in Fig. 5.9b we see that this is not the case, a very smooth curve remains. This curve corresponds approximately to projections of a constant volume. Note that for a volume of height $\delta_x$, the length $L$ of a ray through the center of the volume is

$$L = \delta_x / \cos\theta,$$

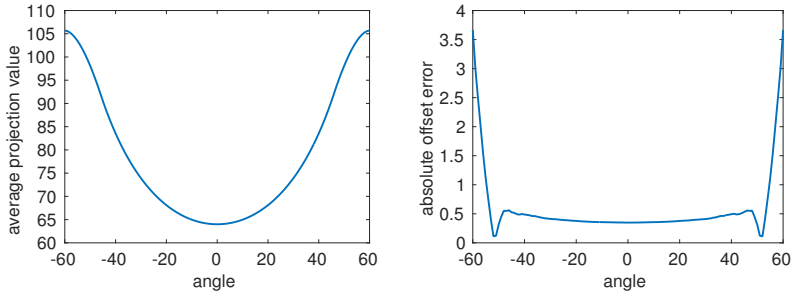
which holds in a certain range of angles (at least $\theta < 90°$, depending on the size of the volume). The same holds approximately for rays that do not go through the center of the volume, except for rays that intersect with the left or right edges of the volume. Therefore, it is likely that the local estimation algorithm using LSQR applied to Eq. (5.12) finds the reconstruction up to a constant. This constant does not increase the residual of Eq. (5.12) since its projections are then subtracted by the local offset estimate. This explains why the local offset estimation is smaller than the true local offset. Fortunately, a constant added to the reconstruction does not change the structure of the reconstruction. In the computation of the local offset estimation error we therefore correct for this offset on the reconstruction (which we already applied in Fig. 5.8).

Most datasets obtained from flat samples in electron microscopy have a gap in the angular range (leading to *missing wedge* artifacts) and the projection images are truncated, meaning that only a part of the sample is visible on the detector [ATM06]. Therefore, we performed experiments to see the effect of these *limited data* problems on the offset and scale retrieval algorithm. From this point forward, we do not include the vector projection method using SIRT for retrieving a global offset, since the method using LSQR is more accurate in the results in Fig. 5.8. In Fig. 5.10a the effect of a missing wedge is shown for the same dataset we used in




(a) local offset estimate

(b) absolute error




(c) average value of projections of a constant volume

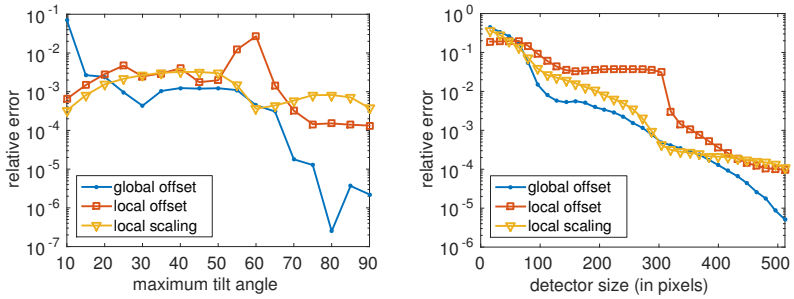
(d) final absolute error

Figure 5.9: (a) Comparison of the recovered local offset and true offset; (b) The absolute difference; (c) Average projection values of a constant volume (where each image pixel is 1), not the similarity to the absolute error of the recovered local offset; (d) Final absolute error after subtracting a multiple of the curve in (c).

the previous experiment, except for a different angular range. Note that the angle on the horizontal axis indicates the maximum rotation angle, *e.g.* 90° indicates an angular range of $[-90°, 90°]$ with 1° increments. The missing wedge does not seem to be influencing the results for realistic rotation angles.

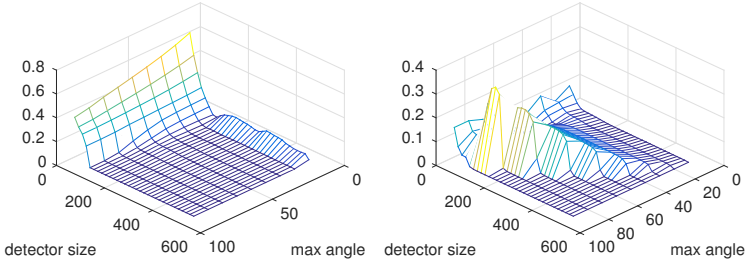
In case of truncation, the local and global offset estimation fails if the detector is smaller than the height of the reconstruction volume, see Fig. 5.10b. Note that the size of the detector determines the size of the circular offset artifact and in this case the offset artifact fully fits inside the reconstruction area. The local offset and scale retrieval are slightly more susceptible to the amount of truncation, but a detector size of 100 pixels seems sufficient.

In experimental data from electron microscopy a missing wedge and truncation are both present. In Fig. 5.11 we show the effect of a combination of these. In this case we see that the effect is more severe. The global offset and scale estimation do not seem to be influenced much. For certain combinations of detector size and angular range the error of the local offset estimation increases considerably, but if the truncation is not severe for a relatively large angular range, this is not a

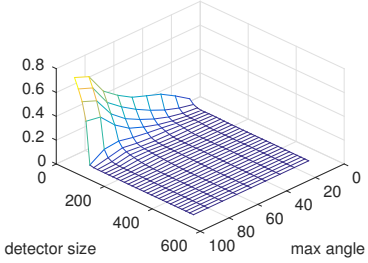



(a) missing wedge (b) truncation

Figure 5.10: Influence of limited data on the offset and scale estimation algorithm.




(a) global offset (b) local offset




(c) scaling

Figure 5.11: Relative error of the retrieved offset/scalings in case a limited angular range is combined with truncation. Truncation is indicated by detector size and missing wedge by the maximum tilt angle.

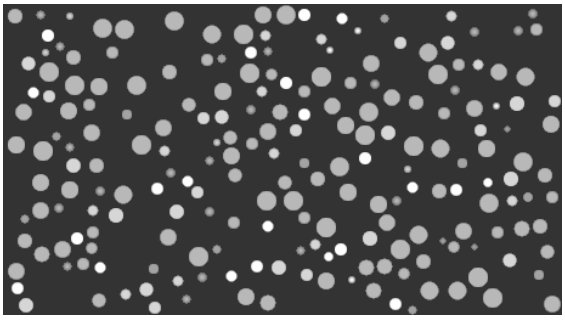


Figure 5.12: Slice in the $z$-direction of the particle phantom of size $460 \times 256 \times 64$.

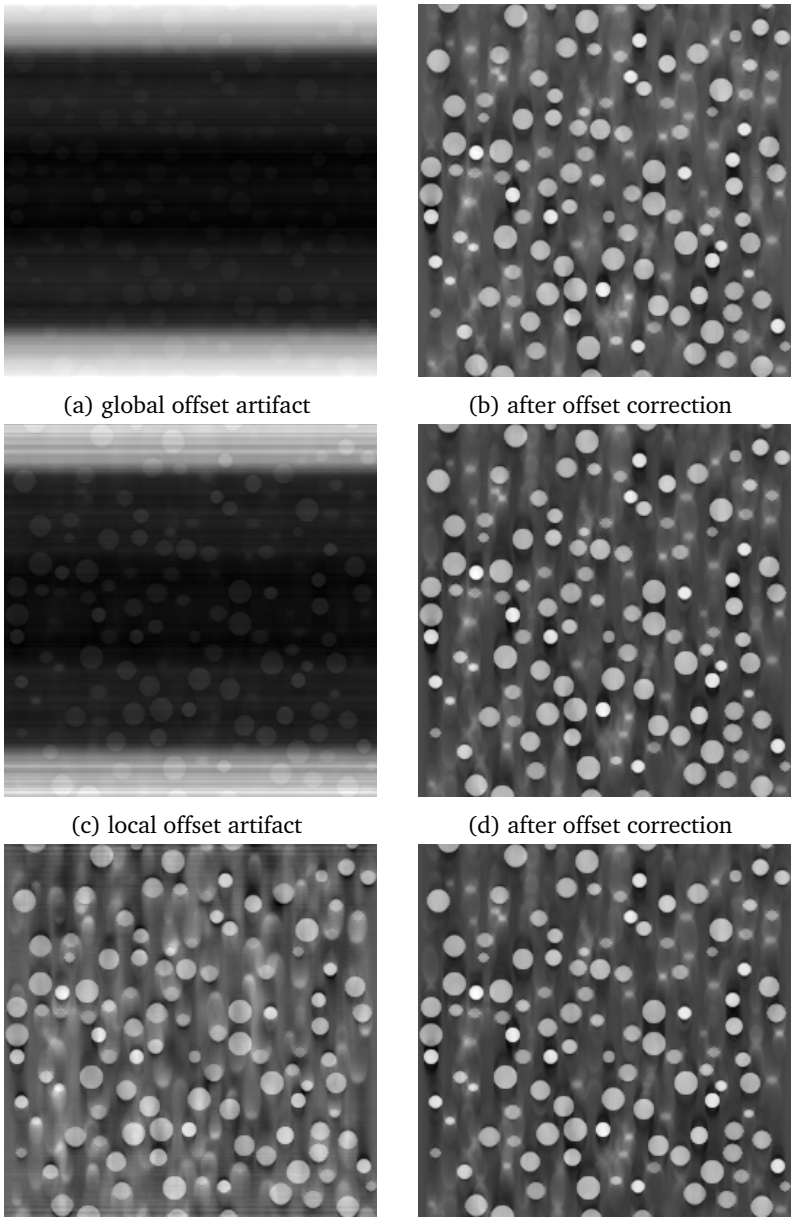
problem. The scale retrieval algorithm is not very sensitive to the missing wedge, but the truncation can be an issue in extreme cases.

### 5.6.2 3D simulation experiments

We consider a particles in substrate phantom of size $460 \times 256 \times 32$ of which a central slice is shown in Fig. 5.12. A total of 121 projections were simulated for a detector of $256 \times 256$ pixels using parallel beam geometry and an angular range of $\pm 60°$. The same offsets and scalings were applied as described in the beginning of Section 5.6.1. The datasets are reconstructed on a volume of size $460 \times 256 \times 32$ pixels. For the implementation of the algorithms we use the ASTRA toolbox for the GPU accelerated forward and backprojection operations [PBS13]. The hardware we used consists of a workstation with an Intel Core i7-2600K@3.4 GHz CPU, 16 GB of system RAM and an NVIDIA GTX 570 GPU.

First we compare the results of the global and local offset and scale estimation on the reconstructions qualitatively using LSQR with 250, 400 and 600 iterations respectively. After obtaining the offsets or scale factors we reconstruct the corrected projection data using 100 iterations of LSQR. In Fig. 5.13 central slices of the reconstructions before and after correction are shown. We show the part of the reconstruction that is in the field of view of the detector for every projection image. The proposed methods are able to significantly reduce the artifacts due to offsets on, and scaling of, the projection images. Some vertical smearing effects can be observed, but this is expected due to the limited angular range of $\pm 60°$ (missing wedge artifacts).

In the next experiment we test the effect of the number of projection angles on the offset or scale retrieval. It is not directly clear if increasing the number of projection images results in a better estimation of the offset and scales, because we add equations to the systems in Eq. (5.12) and Eq. (5.15) and at the same time introduce another unknown (local offset or scale factor). For this experiment we use a subset of the projection images used in the previous experiment, such that the projection images are approximately equiangular distributed in the interval $\pm 60°$. The results shown in Fig. 5.14a indicate that the offsets and scale factors can be found more accurately if the number of projections is increased, except for



(a) global offset artifact (b) after offset correction

(c) local offset artifact (d) after offset correction

(e) local scaling artifact (f) after scaling correction

Figure 5.13: Qualitative comparison of corrected and uncorrected reconstructions. Central slices in the $z$-direction are shown of size $256 \times 256$ (the part that is always in the field of view of the detector).

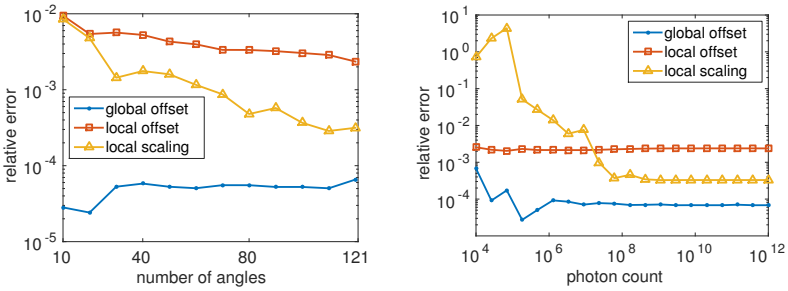



Figure 5.14: (a) Error with respect to the number of projection angles; (b) Error with respect to noise level (expressed in simulated photon counts).

the global offset. However, the accuracy is not dependent to a great extent on the number of projection angles, even for a limited number of projection images the result of the offset and scale estimation is accurate up to 2 significant digits or more.

We also considered the effect of Poisson noise. We simulated the noise and varied the intensity of the noise, which is indicated by the simulated photon counts (lower photon counts means lower signal-to-noise ratio). The results are shown in Fig. 5.14b. The noise does have an effect on the global offset and scale retrieval for datasets with low signal-to-noise ratio. For the global offset estimation, the results are accurate for all noise levels. For the scale factors the noise level has a larger influence. The local estimation algorithm is far less influenced by the noise level and achieves a good accuracy overall.

### 5.6.3 Experimental electron tomography dataset

In the final experiment we test the offset estimation algorithm on an experimental dataset obtained with an electron microscope, using the HAADF-STEM technique. In Fig. 5.15 a projection image of size $512 \times 512$ is shown. The object consists of PbSe/CdSe core/shell nanocrystal particles that are studied in materials science [Bal+11; Cas+12]. The dataset consists of 151 projections from tilt angles between $\pm 75°$ (1° tilt increments) and was obtained by a FEI TITAN$^3$ 50–80 electron microscope using a parallel beam geometry. Because we assume that the offsets are constant for a single projection image, we can simply restrict the reconstruction to a few slices (in the $x$-direction, see Fig. 5.6). This saves a lot of memory and computation time. In this experiment we reconstruct 50 slices in the $x$-direction. We choose a total of 150 slices in the $z$-direction, which results in a reconstruction volume of $50 \times 512 \times 150$.

The background intensity of the dataset is negative, which suggests that the projection data is not scaled, but has a negative offset. Because we do not know if the projections contain a global or local offset, we apply the local offset estimation algorithm as described in Section 5.4.2. The result of a local offset estimation is shown in Fig. 5.16. The retrieved offset indeed indicates a negative offset. The

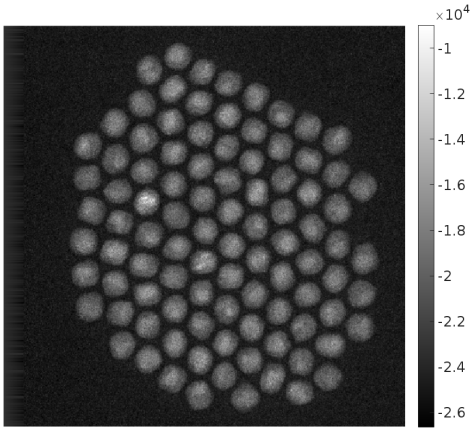



Figure 5.15: Projection image of size $512 \times 512$.

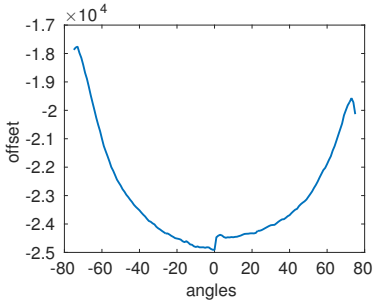
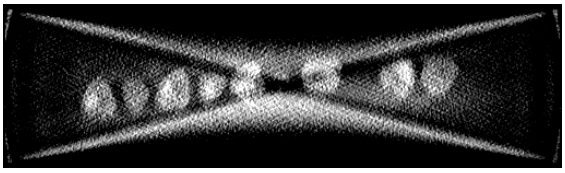


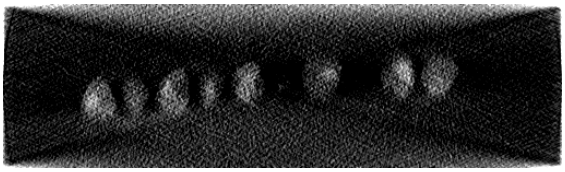

Figure 5.16: The retrieved offset.

core-shell particles are supported by a carbon grid, which has very low contrast in the projection images. Note that the thickness of this support (*i.e.* the path length of the electrons through the support) is inversely proportional to the cosine of the projection angle. This might explain why we the offset behaves like $1/\cos\theta$, because it is a superposition of the offset caused by the support material and a negative global offset. As a result, simply subtracting (the mean of) the background intensity would not be sufficient. Note that the local offset estimation can therefore also be used to reduce the effect of the support material on the reconstruction.

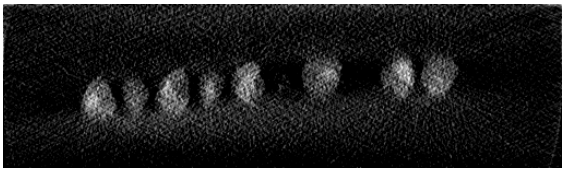
We subtract the retrieved offset from the projection data and compute an LSQR reconstruction, see Fig. 5.17c. We compare this with a reconstruction where the minimum value is subtracted from the projections (an estimate of the background intensity), shown in Fig. 5.17b. The difference in quality can be seen especially on the top and bottom part of the reconstruction, where artifacts are still visible in Fig. 5.17b. The reconstruction after local offset removal is much improved. Compared to an LSQR reconstruction that is not corrected for offset, see Fig. 5.17a, the reconstruction has improved significantly. These results show



(a) Original LSQR reconstruction



(b) LSQR reconstruction with background value subtracted



(c) LSQR reconstruction after offset correction

Figure 5.17: Comparison of the slices in the $x$-direction of the original reconstruction and corrected reconstructions, by removing the minimum value from the projection data (background), and after local offset estimation.

that the proposed offset estimation is an effective technique for removing offset artifacts, without the need for manual estimation of the background intensity.

## 5.7 Discussion and conclusions

During the acquisition of the projection images for tomography, an offset on or scaling of the gray values of the projection images can be introduced by fluctuations in the radiation source's intensity. The offset can be a constant added to each gray value of each projection image, which is a *global offset*, or the offset can be constant only for the pixels in a single projection image, which is referred to as a *local offset*. In our analysis of the filtered backprojection reconstruction of a global offset, we found that the offset causes an additive artifact in the reconstruction that has the shape of a disk. By enforcing a rectangular reconstruction domain, the global offset causes an inconsistency in the reconstruction and by minimizing inconsistency of the reconstruction with respect to a negative correction term, the offset can be found accurately.

We extended the algorithm such that it can be applied to retrieve a local offset and scaling of projection data. We assume that each scale factor scales the gray values of one projection image. These algorithms work in a similar way

as the global offset estimation: a least squares solver (LSQR) is employed to simultaneously compute a reconstruction and find the unknown offsets or scale factors.

In a series of simulation experiments we investigated the effect of *limited data* problems that are typically encountered in electron tomography, such as a limited angular range or truncation of projections (if the sample is not contained in the field of view of the detector). Moreover, we determined how strongly the results of the offset and scale estimation algorithms depend on the shape of the reconstruction, in particular the height. Our conclusions are that a missing wedge or truncation should not pose a problem, if the severity of these effects is not too large. The effect of a missing wedge was less strong, even for a very small tilt range of $\pm 20^\circ$ accurate results could be obtained. The effect of truncation was much larger: for the test image that was 512 pixels wide, results were not so accurate if the detector was smaller than 100, 200 and 300 pixels for the global offset, scaling and local offset estimation respectively. We observed that the offset and scale estimation algorithms yield more accurate estimations if the height of the reconstruction domain is small. On this smaller reconstruction domain the influence of the offset or scaling of the projections is much more pronounced in the residuals of the corresponding reconstructions. Therefore, the offset and scale estimation algorithms are able to retrieve the offsets or scalings more accurately in this case, compared to the case where the reconstruction area is square.

The result of the experimental electron tomography dataset shows that offsets can be found which significantly improve the reconstruction even in cases where no background is visible. The effect of offset artifacts can be substantially reduced.

# Chapter 6

# Robust artifact reduction in tomography using Student's t data fitting

## 6.1 Introduction

Tomography is a technique for reconstructing a 3D volume from 2D projection images, such as X-rays obtained in CT scanners. A 3D reconstruction can be obtained from the projection images by solving an inverse problem. In algebraic reconstruction methods a linear system of equations is solved that represents a discretization of the Radon transform [NW01; KS01]:

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p}. \tag{6.1}$$

The *projection matrix* $\boldsymbol{W} \in \mathbb{R}^{M\times N}$ relates pixel values in the tomographic reconstruction $\boldsymbol{x} \in \mathbb{R}^N$ (*gray values*) to discrete detector measurements $\boldsymbol{p} \in \mathbb{R}^M$. In experiments the projections are perturbed by an unknown noise vector $\boldsymbol{\epsilon}$,

$$\tilde{\boldsymbol{p}} = \boldsymbol{p} + \boldsymbol{\epsilon}.$$

Most algebraic methods such as SIRT, CGLS or LSQR [Bjö96; GB08; PS82] optimize the consistency of the reconstruction in the Euclidean norm, which leads to a least squares solution:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{W}\boldsymbol{x} - \tilde{\boldsymbol{p}}\|_2^2. \tag{6.2}$$

It is well known that this approach is equivalent to finding the maximum likelihood estimate (MLE) of $\boldsymbol{x}$ under the assumption that the error term or noise $\boldsymbol{\epsilon}$ is Gaussian distributed [Pre+07]. However, the $\ell_2$-norm assigns a heavy penalty to outliers in the projection data. Outliers may arise due to acquisition problems

This chapter has been accepted for publication in: *Proceedings of Fully3D*, 2015.

ranging from hardware problems to physical effects such as scattering or photon starvation due to high density particles [BF12]. Because these errors are heavily penalized by the $\ell_2$-norm, the solution of Eq. (6.2) will be fitted to these outliers, producing artifacts in the reconstruction.

In this chapter we propose the use of algebraic methods combined with the Student's t penalty function to solve the reconstruction problem in Eq. (6.1). The Student's t distribution has heavy tails meaning that outliers in the noise are penalized less compared to the $\ell_2$-norm. Therefore the Student's t MLE of the reconstruction should be influenced less by such outliers.

Many methods for artifact reduction are aimed to remove or suppress outliers from the projection data [Gu+06; PDX12; Vel+10; Wan+96], which rely heavily on the accuracy of segmentation techniques to locate outliers. By minimizing the Student's t penalty of the data-fit there is no need for segmentation and therefore the method is not biased by the result of a segmentation step.

We explain the method for finding the Student's t MLE of the reconstruction in Section 6.2. Subsequently, results are presented for a series of 3D cone-beam simulation experiments for reduction of several kinds of artifacts in Section 6.3. Finally, we discuss the results and conclude the chapter in Section 6.4.

## 6.2 Methods

In general, maximum likelihood estimation of $\boldsymbol{x}$ in Eq. (6.1) gives rise to a maximization problem

$$\max_{\boldsymbol{x}} \rho(\boldsymbol{W}\boldsymbol{x} - \tilde{\boldsymbol{p}}),$$

where $\rho(\cdot)$ is the probability density function (PDF) of the probability distribution of the noise $\boldsymbol{\epsilon}$. In practice, the problem is posed as a minimization problem by taking the $-\log$:

$$\min_{\boldsymbol{x}} -\log \rho(\boldsymbol{W}\boldsymbol{x} - \tilde{\boldsymbol{p}}).$$

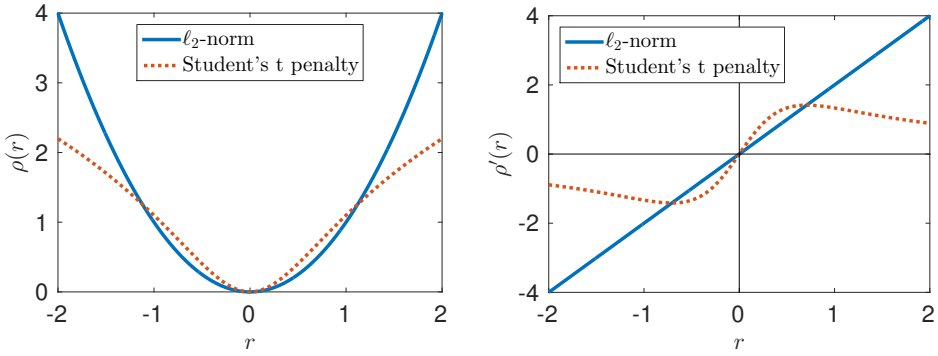
The resulting estimate $\widehat{\boldsymbol{x}}$ can be interpreted as the most likely solution of Eq. (6.1) under the assumption that the noise is indeed distributed according to $\rho$. When $\rho$ represents the Gaussian PDF, this leads to the conventional least squares formulation, Eq. (6.2). When the data contain large outliers, the Gaussian assumption is violated and a different PDF has to be employed. A possible choice is the multivariate Student's t distribution

$$\rho(\boldsymbol{r}) \propto \prod_i (1 + r_i^2/\nu)^{-(\nu+1)/2},$$

where $\nu$ is the variance. Such an assumption on the noise allows for large outliers to be present in the residual, whereas under a Gaussian assumption large outliers are extremely unlikely and thus the reconstruction will aim to fit them.

The *penalty* derived from the Student's t distribution is

$$p(\boldsymbol{r}) = \sum_i \log(1 + r_i^2/\nu), \tag{6.3}$$




(a) Penalty functions (b) Influence-functions

Figure 6.1: Least squares and Student's t penalty functions and corresponding influence-functions with $\nu = 0.5$.

and a graph is shown in Fig. 6.1a. The maximum likelihood estimate is now obtained by solving

$$\min_{\boldsymbol{x}} p(\boldsymbol{W}\boldsymbol{x} - \tilde{\boldsymbol{p}})$$

using Newton's method [NW06]. This leads to an iterative method of the form

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{s}^{(k)},$$

where $\alpha_k$ is the stepsize, determined by a backtracking linesearch and $\boldsymbol{s}^{(k)}$ is obtained by solving

$$\boldsymbol{W}^\intercal \boldsymbol{H}^{(k)} \boldsymbol{W} \boldsymbol{s}^{(k)} = -\boldsymbol{W}^\intercal \boldsymbol{g}^{(k)}. \tag{6.4}$$

Here, the gradient $\boldsymbol{g}^{(k)}$ and diagonal matrix $\boldsymbol{H}^k$ are given in terms of the residual $\boldsymbol{r}^{(k)} = \boldsymbol{W}\boldsymbol{x}^{(k)} - \tilde{\boldsymbol{p}}$ as

$$g_i^{(k)} = \frac{2r_i}{\nu + r_i^2},$$

and

$$h_{ii}^{(k)} = \frac{2}{\nu + r_i^2}.$$

Note that we can use any algebraic method to solve Eq. (6.4), but in our case we chose the CG method. In effect, the algorithm repeatedly performs a reconstruction with a weighted residual, where the weight $(\nu + r_i^2)^{-1}$ down-weights large residuals.

If we look at the so-called *influence-function* [Ham+05] of Eq. (6.3) in Fig. 6.1b which is defined by the gradient, it is clear that the influence of large residuals $r^2 \gg \nu$ is small. However, for $r^2 < \nu$ the influence behaves similar to a least squares penalty. The role of $\nu$ can be seen as tuning parameter to indicate the magnitude of outliers. This parameter can be adjusted automatically [AL12], however, in our experiments we estimate the parameter empirically.

(a) 3D rendering (b) slice

Figure 6.2: (a) 3D rendering of the Shepp–Logan head phantom with a wedge cut out of the sample; (b) central slice of size $256 \times 256$.

From this point forward we will refer to the methods for MLE estimation using least squares and Student's t penalties as LSQR-MLE and ST-MLE respectively, where we use the method LSQR for minimizing the $\ell_2$-norm.

## 6.3 Experiments and Results

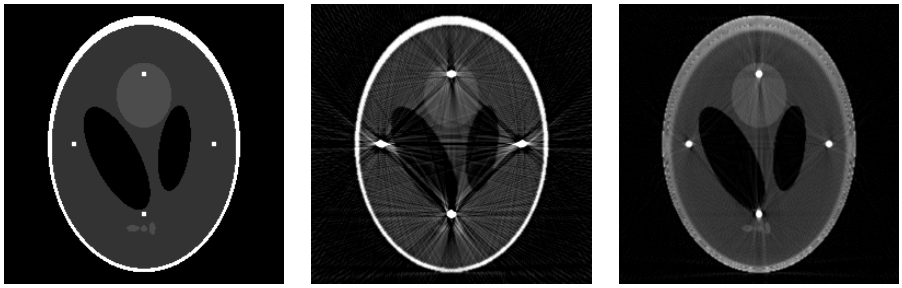
In these simulation experiments we consider a 3D Shepp–Logan head phantom of size $256 \times 256 \times 256$ of which a central slice is shown in Fig. 6.2b. We used the ASTRA tomography toolbox [PBS13] to generate 180 projection images with 1° angular separation using the cone-beam geometry. The detector has a size of $284 \times 284$ pixels and was positioned in the origin. The projection matrix is generated on-the-fly by the GPU back end of the toolbox using a slice interpolation kernel [Jos82].

In the following sections we will discuss several distortions or perturbations in the projection images that cause severe artifacts in the reconstruction and we compare a least squares approach to data fitting using the Student's t penalty function.

### 6.3.1 Metal artifact reduction

In this experiment we consider the 3D Shepp–Logan head phantom with six small dense particles that represent metal implants (density is 10 times that of the outer "skull" region). A single slice is shown in Fig. 6.3a, the six particles form the vertices of an octahedron.

In the area of the detector where the metal implants are projected the data becomes corrupted due to beam hardening, scatter and photon starvation. For this experiment we focus on the effects of photon starvation. In the projection data we simulated a saturation due to photon starvation by setting the region



(a) slice of phantom (b) LSQR-MLE (c) ST-MLE

Figure 6.3: Metal particles Shepp–Logan head phantom and corresponding least squares fit and Student's t fit.

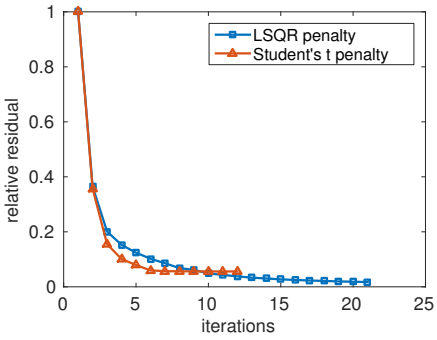
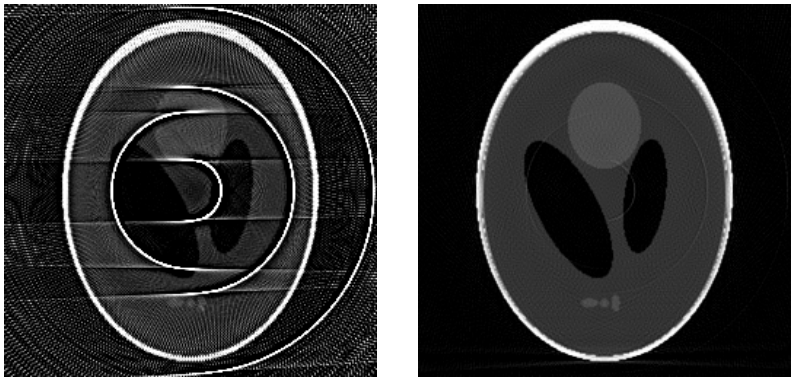



Figure 6.4: Convergence of the $\ell_2$-norm compared to Student's t penalty. These are relative residuals.

corresponding to the metal objects to a constant, large value. The effect of this missing or corrupted data if we apply LSQR-MLE is shown in Fig. 6.3b. Usually, these regions in the projection data are either ignored or filled in by interpolation or inpainting techniques [Gu+06; Vel+10; Wan+96]. These methods rely on sophisticated segmentation techniques in order to locate the metal implants.

We show a convergence plot in Fig. 6.4 of both penalty functions. This figure shows that the ST-MLE method converges rapidly compared to LSQR-MLE. Note, however, that the ST-MLE method requires solving of Eq. (6.4) in each iteration and is therefore significantly more costly. In all of the following experiments, the ST-MLE method converges in approximately 10 iterations.

Our proposed method ST-MLE is able to suppress most of the artifacts, as shown in Fig. 6.3c, while still reconstructing the metal implants without needing to locate the outliers in the projection images. There is an underestimation of the gray value of the skull area, however, visually the reconstruction is very useful for detecting also smaller details, such as the three ellipses below the bottom metal particle. Moreover, the ST-MLE solution can be used initially to obtain a better



(a) LSQR-MLE (b) ST-MLE

Figure 6.5: Defective camera pixels lead to semicircular reconstruction artifacts. The Student's t solution is much less affected by these artifacts.

segmentation of the metal particles.

### 6.3.2 Defective camera pixels

In the second experiment we simulate the effect of defective camera pixels. We assume that the detector has several "dead" detector pixels which measure no photons at all. This produces bright pixels in the projection images that are constant between projections. The uncorrected projection data will produce ring artifacts which are typically removed by inpainting of dead pixels [PDX12].

We simulated a dataset with 100 randomly selected dead pixels which we set to a constant value of two times the maximum value of the projection data. The least squares solution is shown in Fig. 6.5a. The artifacts are severe, but the Student's t approach in Fig. 6.5b is able to remove the artifacts almost completely.

In Fig. 6.6 we show the effect of increasingly many dead pixels on the mean squared error of the reconstruction compared to the ground truth. Surprisingly, even if the number of dead pixels is close to 50% of the total number of detector pixels the ST-MLE solution does not seem to be influenced by this missing data.

### 6.3.3 Randomized projection images

In the final experiment we created a dataset of which we replaced 50 from the 180 projections by completely random images (white noise) with average intensity similar to the other projection images. Although this is not a very realistic dataset, we want to see how far we can stress our ST-MLE method and see if it can ignore such inconsistent data.

The LSQR-MLE solution is shown in Fig. 6.7a, which is very noisy due to the randomized projections. The ST-MLE solution (Fig. 6.7b) suffers far less from the random projections and only shows mild noise. There are some streak artifacts because the projection images in these directions are missing, but this is expected.

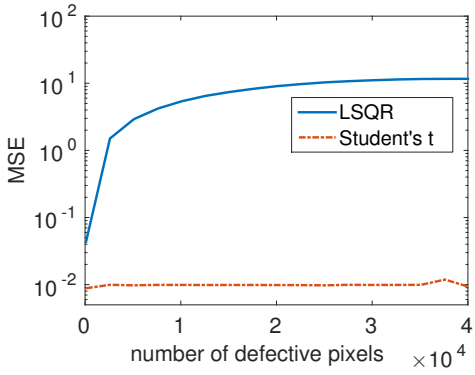
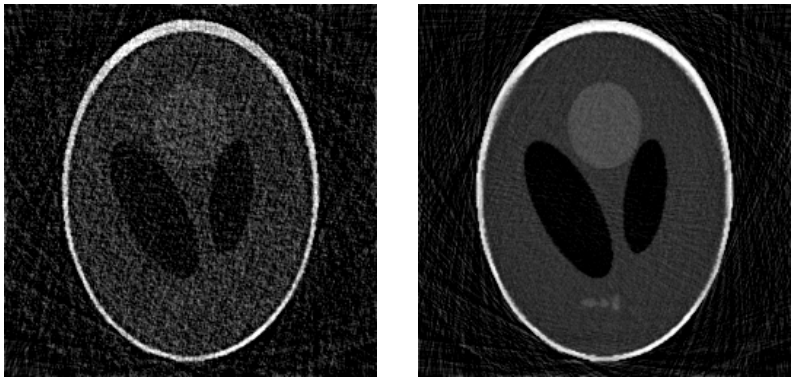



Figure 6.6: Mean squared error of the reconstruction compared to the ground truth for an increasing number of dead detector pixels.



(a) LSQR-MLE (b) ST-MLE

Figure 6.7: LSQR and Student's t fit for dataset with 50 randomized projection images out of 180 total projection images.

We also compared LSQR-MLE and ST-MLE on datasets with an increasing number of random projections. Of course we cannot expect that the ST-MLE solution will be unaffected by this as was the case in the previous experiment, because we are essentially removing projections. However, the result shown in Fig. 6.8 indicates that the ST-MLE method is beneficial for each of these dataset and is a large improvement over the least squares solution.

## 6.4 Discussion and conclusions

In this chapter we have discussed the Student's t penalty function that can be used in combination with Newton's optimization approach to produce the maximum likelihood estimate of the tomographic reconstruction problem Eq. (6.1)

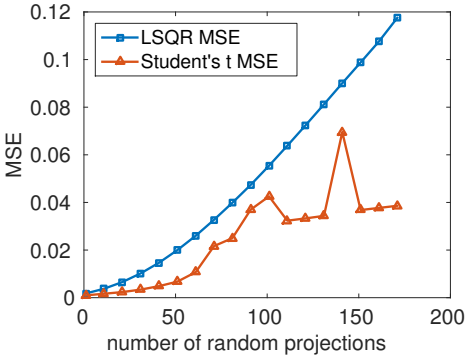



Figure 6.8: Mean squared error of the reconstructions for an increasing number of random projections replacing the original projections.

corresponding to the Student's t distribution. In our experiments we have seen that perturbations introduced in the projection data due to hardware problems or photon starvation from metal implements are significantly reduced using our proposed method ST-MLE when compared to algebraic reconstruction methods that minimize the Euclidean norm of the residual (LSQR-MLE). In contrast to other methods for artifact reduction, there is no need to locate outliers in the projection data by segmentation methods. Therefore, the ST-MLE method can be applied effectively without any preprocessing steps. Moreover, the Student's t penalty can be used in combination with other reconstruction algorithms and image priors and has other potential use cases such as artifact reduction from diffraction effects.

# Chapter 7

# Easy implementation of advanced tomography algorithms using the ASTRA toolbox with Spot operators

## 7.1 Introduction

Tomography is an imaging technique for reconstructing an object from projections. In medical imaging, projections can be obtained as X-ray images by CT scanners. In the scientific community, many devices and setups are used for tomographic data acquisition, from electron microscopes to large synchrotron facilities [Don+06; Küb+05]. Hardware advances have pushed the ability to image on smaller scales and at large pixel densities. Projection images in the order of 4000 by 4000 pixels can now be obtained routinely [Hu+14]. At the same time, many innovative tomography applications are inherently limited in the number of projections that can be acquired, and their associated noise level.

In recent years, we have seen many advances in reconstruction algorithms for tomography that incorporate prior knowledge about the scanned object. Examples can be found in sparse reconstruction techniques and discrete tomography [BS11; FNW07; GO09; Sch+05; SJP12; SP08]. The benefit of these methods is their ability to produce accurate reconstructions from limited projection data. To develop such algorithms, high-level mathematical scripting languages such as Matlab are commonly used due to the complex mathematics involved. As a result, the initial implementations may not be suitable for dealing with large experimental datasets due to the inherent performance and memory limits of the scripting platform.

This chapter has been published with minor modifications as:
F. Bleichrodt, T. van Leeuwen, W. J. Palenstijn, W. van Aarle, J. Sijbers, and K. J. Batenburg. "Easy implementation of advanced tomography algorithms using the ASTRA toolbox with Spot operators". In: *Numerical Algorithms* (2015), pp. 1–25

Tomography algorithms are usually constructed by combining two linear operators – *forward projection* and *backprojection* – with additional algorithmic steps. For large-scale datasets, the corresponding matrices are too large to store explicitly. For this reason, Matlab implementations based on explicit matrix computations cannot be used in a straightforward manner. Efficient, parallel (GPU) implementations are used instead [AF11; Jan+09]. Although software packages that exploit parallelism are widespread [Chi+11; KMM96; Ped+10; Riv12; Thi+12], it is not trivial to use these software implementations in combination with algorithms written in scripting languages.

The work presented in this chapter is based on two popular toolboxes for Matlab: the ASTRA toolbox [15; PBS11; PBS13] and the Spot toolbox [BF14]. The ASTRA toolbox is a Matlab toolbox for tomographic reconstruction, based on high-performance GPU primitives. It supports multiple geometries (parallel beam, fan beam, cone beam) with highly flexible source/detector positioning. The Spot toolbox exposes external implementations of linear operations through a standard Matlab matrix interface.

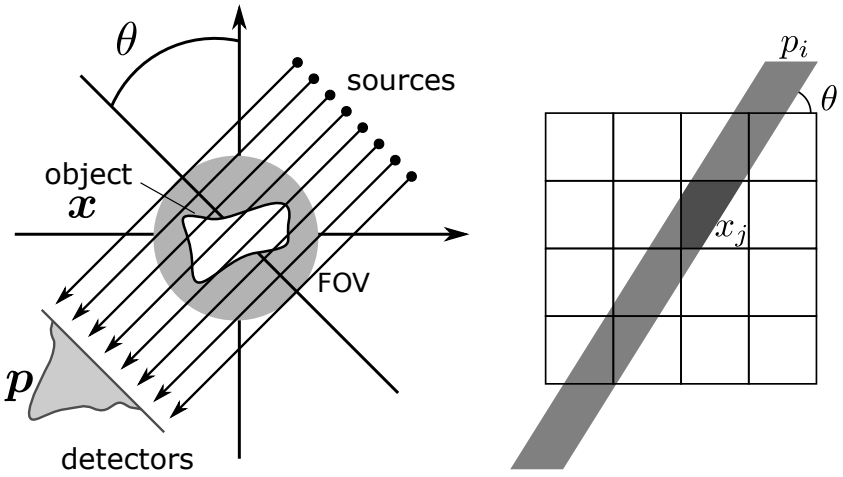
Our key contribution is the introduction of a Spot operator for ASTRA, which we have named `opTomo`. We will show how it can be used to easily develop complex tomography algorithms that are directly applicable to large datasets. Our examples show that the `opTomo` operator enables the use of a range of built-in and external Matlab packages for tomography. Additionally, the Spot operator can be used to develop new algorithms without having to deal with complex implementation details. The code resembles pseudocode and is therefore easy to understand and maintain. Moreover, the code is highly generic, since it can still be used with explicit matrices.

We focus on the Matlab interface of the ASTRA toolbox rather than the optional *Python* interface. Matlab is commonly used by applied mathematicians who work on new reconstruction methods, linear solvers, sparse reconstruction etc. Therefore, many Matlab templates are available of such algorithms which can benefit from using our `opTomo` Spot operator.

The chapter is structured in six sections. First we give a short introduction to tomography in Section 7.2. In Section 7.3 we describe the software elements that are used for implementing the `opTomo` operator. Several use cases of the `opTomo` operator are described in Section 7.4. To give an idea of the efficiency that is achieved by using the `opTomo` operator, performance benchmarks are discussed in Section 7.5. Finally, conclusions are drawn in Section 7.6.

## 7.2 Tomography

As an example of the scanning geometry we first introduce the common parallel beam geometry, illustrated in Fig. 7.1a. In this setup, the detector consists of an array of pixels that measure the radiation intensity along parallel lines. Projections are measured along a range of angles, rotating around the object. The object is subsequently reconstructed from these projections by a tomographic reconstruction algorithm.




(a) geometry (b) model

Figure 7.1: Tomography with parallel beam geometry. The left image shows the geometry of a typical parallel beam scanner. The image on the right shows the corresponding discretization. The object is represented by an image and a projection is modeled as a linear combination of the pixel values.

Until recently, analytical reconstruction methods such as filtered backprojection (FBP) [KS01] were used almost exclusively due to their computational efficiency and accuracy, provided that sufficient data is available. Throughout this chapter we will focus on *algebraic reconstruction methods* (see, *e.g.*, Chapter 7 of [KS01] and [AK84; DLR77; Gil72; GBH70]). These methods are based on a particular discretization of the data in *pixels* and involve algebraic equations for the values of the pixels. This approach offers more flexibility for enforcing constraints on the reconstructed image, in contrast to analytical methods. In algebraic reconstruction methods, the object is assumed to have a constant density in each pixel, which is represented in the image by the *gray value*. The contribution of an object pixel to a detector pixel measurement is proportional to its gray value. In many cases, a weight for the pixel is determined from the length (line model) or area of intersection of the beam and the pixel (strip model), but there are many other options [DB04; Jos82; Lew92; Sid85]. The strip model is illustrated in Fig. 7.1b.

This linear relation between object pixels and a detector pixel measurement is expressed by the *ray sum* or *line projection*

$$p_i = \sum_{j=1}^{N} w_{ij} x_j, \tag{7.1}$$

where $w_{ij}$ is the weight assigned to image pixel $j$ and detector pixel $i$. The full set of equations leads to the following linear system:

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p}. \tag{7.2}$$

The object $\boldsymbol{x} \in \mathbb{R}^N$ and the projection data $\boldsymbol{p} \in \mathbb{R}^M$ are represented by vectors. The sparse matrix $\boldsymbol{W} \in \mathbb{R}^{M \times N}$, referred to as *projection matrix* or *system matrix*, holds the weights of each pixel.

In many applications, the system in Eq. (7.2) is underdetermined and the projection matrix does not have full row rank. This results in a challenging ill-posed reconstruction problem. A basic approach for solving it is by minimizing the residual norm, which is referred to as *projection distance*:

$$\underset{\boldsymbol{x}}{\text{minimize}} \; \|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{p}\|_2^2. \tag{7.3}$$

If the system is underdetermined, there is no unique solution. Moreover, if noise is present in the projection data, the system of equations can be inconsistent. Regularization techniques should be used to alleviate both of these problems.

## 7.3 Software implementation

In this section we discuss the implementation of the `opTomo` operator and introduce the software tools that are used.

### 7.3.1 The ASTRA toolbox

The ASTRA toolbox is an open source software package for tomographic reconstruction and algorithm design [PBS13]. The toolbox provides tools and building blocks for the development and implementation of tomographic reconstruction methods. Moreover, it provides many popular reconstruction algorithms, such as filtered backprojection (FBP) and several iterative reconstruction methods such as SIRT and CGLS [BE79; Gil72]. The toolbox has a Matlab and Python interface, which give access to the *forward* and *backprojection* operations. These operations are based on the model in Eq. (7.2). The forward projection generates projections from an image vector, *i.e.*, this corresponds to multiplying an image vector by the projection matrix $\boldsymbol{W}$. A backprojection corresponds to multiplication by $\boldsymbol{W}^\intercal$, the transpose of the projection operator. The ASTRA toolbox uses ray-tracing techniques to compute these matrix-vector products, such that matrices are not stored, but their elements are generated when needed. High memory usage is avoided in this way; only the reconstruction and the projections should fit in memory (and possibly a few copies, depending on the reconstruction algorithm). The GPU can be used for fast computation of the forward and backprojection steps, allowing large datasets to be processed in reasonable time.

We will not go into detail about the inner workings of ASTRA's Matlab interface, but rather briefly introduce the main ideas to call the GPU backend. In Listing 7.1, a utility function is shown for the forward projection algorithm on the GPU through Matlab's *mex* interface. A few details need to be clarified here. The `data` Matlab array represents (a slice of) the object. It can be a phantom image or a (partial) reconstruction, from which we want to compute projections. The computation of the projections depends on the particular scanner geometry being used. Therefore, the `proj_geom` structure contains details about the beam type

(parallel, fan, cone), the angles at which projections need to be generated and the detector dimensions. The output consists of a data identifier (used internally in ASTRA) and a Matlab array containing the projection data. Similarly, the volume geometry or `vol_geom` structure contains details about the dimension of the object, from which projections are computed. Because the ASTRA toolbox is very flexible in setting up geometries, many acquisition schemes can be modeled, including circular cone-beam, helical cone-beam, and laminography setups. More details about the geometries are given in Section 7.A. For further details about the use of and possibilities of the ASTRA toolbox, we refer the reader to [PBS13].

Listing 7.1: Utility function for forward projection

```
% forward projection
[id, sinogram] = astra_create_sino_cuda(data, proj_geom, ...
    vol_geom);
```

### 7.3.2 The Spot toolbox

For implementing algorithms based on a linear operation, it is very convenient to use matrix-vector notation, as used in Matlab, because it is similar to the mathematics and results in clean and concise code. However, in many use cases it is not practical to form the matrix corresponding to the linear operation explicitly. As a solution, the Spot toolbox provides a Matlab framework that wraps linear operations into Matlab objects that act like matrices [BF14]. The toolbox introduces a new kind of data type (by using *classes*), called *Spot operators*.

A Spot operator for a linear operation $\boldsymbol{A}$ relies on (external) software implementations of the following operations:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{7.4}$$

$$\boldsymbol{y} = \boldsymbol{A}^{\mathsf{T}}\boldsymbol{x}. \tag{7.5}$$

The matrix operations listed in Table 7.1 are overloaded for Spot operators and are based on these basic operations of Eq. (7.4) and Eq. (7.5). Most Matlab *functions* which would not directly support a Spot operator are overloaded as well. One example is the `sum` function. Applying any operation listed in Table 7.1 to a Spot operator (except for division), does not produce a matrix, but generates another Spot operator of a different type. If the Spot operator is applied to a vector, the result will be computed based on the implementation of Eq. (7.4) and Eq. (7.5). All other operations can be derived from these. For example, if we want to extract rows or columns from a matrix, we use the parentheses syntax and subscript sets $s_1$ and $s_2$ to indicate which rows and columns we want to extract.

Since the matrix elements of a Spot operator $\boldsymbol{A}$ are never stored explicitly, some operations can be slower than expected. For example, if a vector is multiplied by the first row of $\boldsymbol{A}$ (using Matlab notation), `y = A(1,:)*x`, Spot uses the implementation of Eq. (7.4) and effectively computes:

Table 7.1: Operations overloaded by Spot.

| **matrix operations** | | | |
|---|---|---|---|
| ctranspose | A' | plus | A+B |
| divide | A\B | subsasgn | A(s1,s2,..,sn)= B |
| horzcat | [A B] | subsref | A(s1,s2,..,sn) |
| minus | A-B | transpose | A.' |
| mldivide | A\B | uminus | -A |
| mpower | A^i | uplus | +A |
| mrdivide | A/B | vertcat | [A; B] |
| mtimes | A*B | | |

```
1    y = A*x;
2    y = y(1);
```

which takes more time if the first row of $\boldsymbol{A}$ could have been formed explicitly. For the same reasons, operations that work element-wise are not implemented for Spot operators, such as element-wise multiplication: `B = A.*A`, or using functions such as `norm(A)`. However, in many typical use cases, such operations can be avoided or if possible can be implemented at a lower level (*e.g.*, in the ASTRA toolbox).

By using the Spot toolbox, Matlab code that uses matrices can now also be used with Spot operators that are linked to fast, external implementations of operations Eq. (7.4) and Eq. (7.5). So, without modification, the same Matlab code can be used with different implementations of the linear operations.

### 7.3.3 The ASTRA Spot operator

For the tomography Spot operator, which we refer to as `opTomo`, we use the forward and backprojection operations from the ASTRA toolbox as implementations of Eq. (7.4) and Eq. (7.5), based on the model in Eq. (7.2).

In code Listing 7.2 the construction of an `opTomo` object is shown, requiring three arguments:

1. The linear model used to generate $\boldsymbol{W}$
2. The projection geometry
3. The volume geometry

These arguments are used to set up the forward and backprojection algorithms of ASTRA and to allocate data structures. With the creation of this new Spot operator, we can directly use the matrix operations listed in Table 7.1. In line 5 of Listing 7.2 we compute a forward projection of a sample image. We will now explain the internals of these operations in greater detail. The matrix multiplication operation `mtimes` is overloaded for `opSpot` (the superclass from which all Spot operators

Matlab
```
p = W*x;
```
↓
Spot
```
% overloaded in superclass
opSpot.mtimes(W,x)
```
↓
opTomo operator
```
% multiply function contains ASTRA code
% input argument two: 1 - no transpose, 2 - transpose
opTomo.multiply(x,1)
```
↓
ASTRA
```
% ASTRA code for forward projection
x = reshape(x, vsize);
% store data in C++
astra_mex_data3d('store', vol_id, x)
% run forward projection
astra_mex_algorithm('iterate', cfg_fp);
% obtain Matlab array
p = astra_mex_data3d('get', sino_id);
```

Figure 7.2: Typical code flow of the opTomo operator.

are derived). Therefore, the image is passed through the `mtimes` function to the `multiply` function of opTomo. We implemented the `multiply` function which: reshapes the vector to an image, passes the data to the ASTRA toolbox and calls the forward projection algorithm. Similarly, the backprojection is called if we use the transpose of the opTomo operator (also through the `multiply` function). This typical code flow is illustrated in Fig. 7.2 and shows how the components (Matlab, Spot, opTomo, ASTRA) are connected.

Through the opTomo operator, we expose the forward and backprojection operations of ASTRA to Matlab. By choosing the first argument in line 1 of Listing 7.2 we can choose the model used for the forward and backprojection (to generate $\boldsymbol{W}$ in Eq. (7.2)). For the CPU projectors, the models `'linear'` [Jos82], `'line'` [Sid85] and `'strip'` [Zhu+08] are available. If we pass the option `'cuda'`, the fast GPU projector is used, which is based on the Joseph interpolation kernel [Jos82] for the forward projection and uses a pixel-driven method with linear interpolation for the backprojection [GZ10b]. Note that the backprojector in ASTRA is not exactly equivalent to the transpose of the matrix corresponding to the forward projector. This design was chosen to greatly improve performance of the backprojector [PBS11]. As a result, the corresponding matrices of these operators are not fully consistent with Eq. (7.4) and Eq. (7.5). However,

Listing 7.2: opTomo operator

```
1  % Create a tomography Spot operator 'opTomo'
2  W = opTomo('cuda', proj_geom, vol_geom);
3
4  % can be used to create projection data as a vector
5  p = W*im(:);
6
7  % reconstruction using a Krylov subspace method
8  x = lsqr(W,p);
```

it was shown that an unmatched forward and backprojector can even improve convergence rates of reconstruction algorithms [GZ10a; ZG00].

Listing 7.2 illustrates that using the opTomo operator we can compute projection data by using intuitive syntax similar to Eq. (7.2). The code in line 8 reconstructs the ground truth image from its projections. By using opTomo, the code in Listing 7.2 is short and stays close to the mathematics, is generic and easy to follow for someone not familiar with the toolbox.

## 7.4 Case studies

In the previous sections we have discussed the motivation and implementation details of the opTomo Spot operator. In this section we will demonstrate that the opTomo operator in combination with ASTRA enables the application of Matlab scripts, ranging from simple scripts to large external packages, to large experimental datasets.

### 7.4.1 Custom SIRT implementation

Our first use case is an implementation of SIRT [Gil72] using opTomo. This example demonstrates the simplicity of implementing existing or new algorithms based on their pseudocode. Although SIRT is already implemented in ASTRA, using the code of the current example can have benefits. For example, if we want to use Tikhonov regularization with SIRT we can simply do this by concatenating Spot operators. To see this, note that Tikhonov regularization is based on Eq. (7.3) with an additional penalty term $\lambda\|\boldsymbol{x}\|_2^2$ on the Euclidean norm of the solution. We can rewrite this problem as:

$$\underset{\boldsymbol{x}}{\text{minimize}} \left\| \begin{pmatrix} \boldsymbol{W} \\ \sqrt{\lambda}\boldsymbol{I} \end{pmatrix} \boldsymbol{x} - \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{0} \end{pmatrix} \right\|_2^2. \tag{7.6}$$

The corresponding concatenated matrices and right-hand side can now be used as input for SIRT to enable Tikhonov regularization.

The pseudocode of SIRT is listed in Algorithm 3. Note that SIRT converges to a weighted least squares solution [GB08],

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{p}\|_R^2 \tag{7.7}$$

where the norm $\|\boldsymbol{u}\|_R^2 = \boldsymbol{u}^\intercal \boldsymbol{R}\boldsymbol{u}$, is scaled by inverse row sums.

**Algorithm 3** SIRT

**Input:** Projection data $\boldsymbol{p}$, projection operator $\boldsymbol{W}$ and initial guess $\boldsymbol{x}^0$.
**Output:** Reconstruction $\boldsymbol{x}$.

*Compute inverse column sums:*
$c_j = 1/\sum_{i=1}^{M} w_{ij}$ for $j = 1, \ldots, N$

*Compute inverse row sums:*
$r_i = 1/\sum_{j=1}^{N} w_{ij}$ for $i = 1, \ldots, M$

Let $\boldsymbol{C} = \mathrm{diag}(\boldsymbol{c})$ and $\boldsymbol{R} = \mathrm{diag}(\boldsymbol{r})$

**for** $k = 0, 1, \ldots$ **do**
  $\boldsymbol{u}^k = \boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}^k$
  $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \boldsymbol{C}\boldsymbol{W}^\intercal \boldsymbol{R}\boldsymbol{u}^k$
**end for**

We use the notation $\mathrm{diag}(\boldsymbol{x})$ to represent a diagonal matrix with $\boldsymbol{x}$ on its diagonal. SIRT iteratively refines an image vector $\boldsymbol{x}$ by adding a weighted back-projection of the residual projection data. Such an algorithm is presented very compactly in matrix-vector products. As a result, the Matlab code for SIRT, shown in Listing 7.3, is almost identical to the pseudocode.

### 7.4.2 Cone beam reconstruction

In practice, most tomographic scanners use a point X-ray source that emits a cone-shaped beam, in contrast to a parallel beam. This *cone beam* geometry is also supported by the ASTRA toolbox. A detailed description of the geometric parameters is given in Section 7.A.

Listing 7.2 can directly be applied if the `proj_geom` structure has been set up for a cone beam geometry, as the geometry is not hard-coded in the algorithm. To demonstrate that the code is not restricted to small test problems, we have applied it to a large dataset. The dataset consists of projections from a metal foam and are of size $1000 \times 524$ taken at 511 angles. The reconstruction grid was $1000 \times 1000 \times 524$. We used LSQR for the reconstruction and stopped the computation after 100 iterations. The central slice and isosurface of the reconstruction are shown in Fig. 7.3.

For the computation we used a workstation with an NVIDIA Tesla C2070 GPU. The details of the hardware are given in Section 7.5.1. The computation took

Listing 7.3: The SIRT algorithm using opTomo

```
% To use Tikhonov regularization:
% V = [W;lambda * opEye(size(W,2))];
% p = [p; zeros(size(W,2),1)];
% and use V below instead of W

% determine scaling matrices
r = 1./sum(W,2);
c = 1./sum(W,1);

c(c==Inf) = 0;
r(r==Inf) = 0;

% set up diagonal Spot 'matrices'
C = opDiag(c);
R = opDiag(r);

for i = 1:maxit
    % compute residual
    u = p - W*x;

    % update current solution
    x = x + C*W'*R*u;
end
```

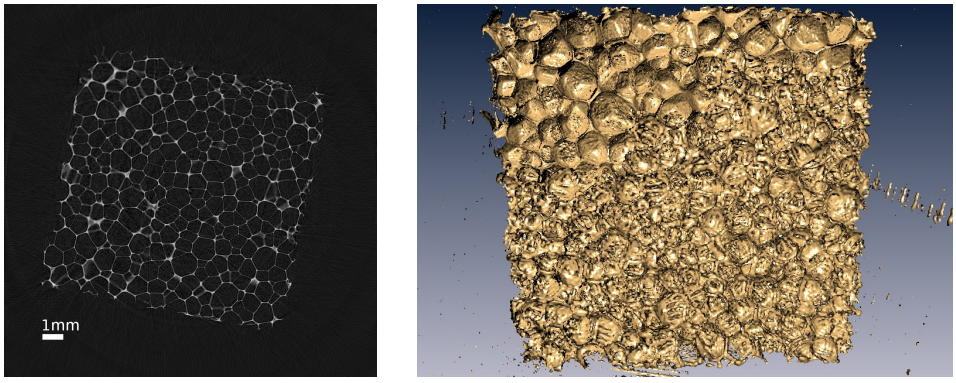



Figure 7.3: Cone beam dataset acquired using a Skyscan 1172. The left image shows one slice of the reconstruction of size $1000 \times 1000 \times 524$. On the right an isosurface is rendered in 3D, showing the structure of the pores.

1 hour and 51 minutes, which is about 67 seconds per LSQR iteration. For this dataset, a forward projection takes about 44 seconds and the backprojection takes about 21 seconds.

### 7.4.3 Sparse image reconstruction

Ideas and methods used in compressive sensing are now commonly applied in tomography. If the object is sparse in a suitable basis, it can often be reconstructed accurately by using $\ell_1$-regularization. Algorithms using $\ell_1$-regularization are more elaborate to implement compared to linear least squares solvers.

One approach to compute a sparse solution of the algebraic reconstruction problem in Eq. (7.3) is basis pursuit denoising, formulating the sparse reconstruction problem as minimizing the $\ell_1$-norm of the image under consistency conditions:

$$\underset{\boldsymbol{x}}{\text{minimize}} \; \|\boldsymbol{x}\|_1 \text{ subject to } \|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{p}\|_2 \leq \sigma. \tag{7.8}$$

The $\ell_1$-norm promotes solutions with few nonzeros. The parameter $\sigma$ is an estimate of the noise level.

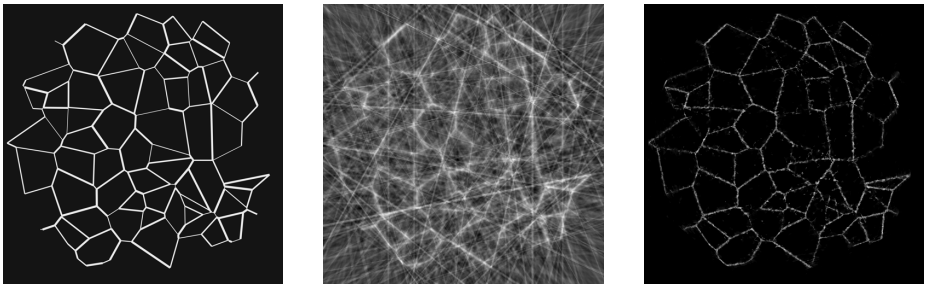
The basis pursuit denoising approach is implemented in the Matlab package SPGL1, written by Friedlander and van den Berg [BF08]. This solver is based on matrix-vector products and is therefore suitable for Spot operators without any modifications to the code. In the following experiments we will use SPGL1 for sparse image reconstruction.

In Fig. 7.4a a foam phantom is shown that has around 7% of nonzero pixels. This dataset is therefore very suitable for sparse image reconstruction. The ground truth image has dimensions $8192 \times 8192$. In total 25 projections were generated using Eq. (7.2) with a total of 512 detector elements per projection angle. The projection data was reconstructed at an image size of $512 \times 512$. In Fig. 7.4b a least squares solution computed with LSQR is shown. The high angular separation of the projections and the sparse character of the ground truth results in many streak artifacts. Resolving the edges of this reconstruction, *e.g.*, by segmentation, will be difficult. In Fig. 7.4c a reconstruction using SPGL1 and the `opTomo` operator is shown. This result shows that including sparsity priors during the reconstruction drastically improves the quality of the reconstruction. Moreover, it is easier to resolve the edges of the foam, by segmentation.

For computations we used a workstation with an NVIDIA GTX 570 GPU, the details of the hardware are given in Section 7.5.1. LSQR was set to stop after 100 iterations or if a relative residual of 0.01 was achieved. In total 11 iterations were needed and the total runtime was 162 ms. The SPGL1 routine was set to a fixed number of 100 iterations, which took 3.5 s. Both the forward- and backprojection took about 4 ms.

### 7.4.4 Sparse wavelet reconstruction

The approach of the previous section is not limited to objects that are sparse in a pixel basis, but can also be used with other sparsity priors. For example, images that have few edges and large homogeneous regions with a constant gray value (such as the Shepp–Logan phantom in Fig. 7.5a), have a sparse representation in a Haar wavelet basis. In this case, we need to minimize the $\ell_1$-norm of the wavelet coefficients. Note that an image $\boldsymbol{x}$ can be decomposed in its wavelet coefficients



(a) phantom (b) LSQR solution (c) SPGL1 solution

Figure 7.4: A foam phantom presented as a sparse image. An algebraic reconstruction method LSQR is compared to the method SPGL1. The SPGL1 method exploits the sparsity prior of the solution.

$\boldsymbol{y}$, by using a linear transformation:

$$\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x}. \tag{7.9}$$

The matrix $\boldsymbol{B}$ is formed from the basis vectors corresponding to the discrete wavelet decomposition. Because $\boldsymbol{B}$ is unitary for the Haar wavelet, the image can be formed from its wavelet decomposition by multiplying the coefficient vector $\boldsymbol{y}$ with its transpose $\boldsymbol{B}^\intercal$ from the left. The system matrix is adjusted to incorporate the wavelet coefficients:

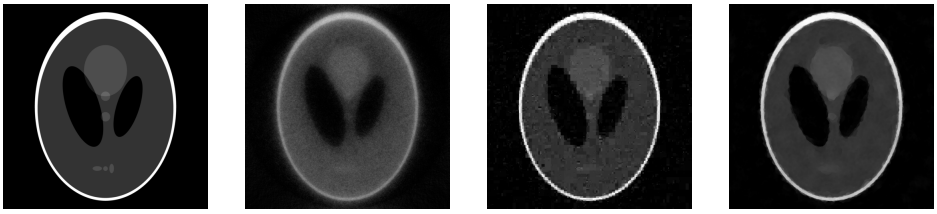
$$\underset{\boldsymbol{y}}{\text{minimize}} \; \|\boldsymbol{y}\|_1 \text{ subject to } \|\boldsymbol{W}\boldsymbol{B}^\intercal\boldsymbol{y} - \boldsymbol{p}\|_2 \leq \sigma. \tag{7.10}$$

Note that this approach is the same as the basis pursuit denoising problem in Eq. (7.8), which is solved with SPGL1. The linear operator involved is now a combination of the wavelet operator and tomography operator. The corresponding Spot operators can be combined in a straightforward manner, as shown in Listing 7.4. This results in very compact code that is easy to understand from a mathematical perspective.

Listing 7.4: Combined Spot operator

```
% Projection operator
W = opTomo('cuda', proj_geom, vol_geom);
% 2D wavelet operator
B = opWavelet2(n, n, 'Haar', [], levels);
sigma = 200;
y_spgl1 = spgl1(W*B', sinogram(:), [], sigma);
```

We applied the algorithm on a dataset based on the Shepp–Logan phantom of size $4096 \times 4096$. A total of 100 projections were generated with 4096 detector



(a) phantom (b) LSQR solution (c) wavelet based SPGL1 solution (d) TV-min solution

Figure 7.5: Comparison of several reconstruction algorithms. LSQR is an algebraic reconstruction method. The SPGL1 solution exploits sparsity of the ground truth in a Haar wavelet basis. The TV-min solution exploits sparsity of the gradient image.

elements per angle. The data was perturbed by applying a moderate amount of Poisson noise to the projection data. The reconstruction size was also $4096 \times 4096$.

The results are shown in Fig. 7.5. Note that we abort LSQR (Fig. 7.5b) after three iterations, to prevent overfitting (such that it has a similar $\ell_2$-norm as the SPGL1 solution). For brevity, we do not provide quantitative details on the noise generation, but the noise level can be observed quite clearly in the LSQR reconstruction. The resulting reconstruction contains substantial noise and details have been blurred. The wavelet based SPGL1 solution in Fig. 7.5c is an improvement. Due to the shape of the Haar wavelet, most of the noise is in the detail coefficients that are likely to be suppressed. Although the Haar wavelet is causing block-like artifacts, the edges are better pronounced compared to the least squares solution. For comparison, we also show the results of applying the Chambolle–Pock algorithm for Total Variation (TV) minimization, which will be introduced in the next section. For the Shepp-Logan phantom, TV-minimization appears to be a more suitable prior as shown in Fig. 7.5d. Both the wavelet and total variation minimization method suppress high gradients and therefore produce reconstructions with less noise, compared to LSQR.

For these computations we used a workstation with a GTX 570 GPU. LSQR took 3.8 s, SPGL1 138 s, Chambolle–Pock 679 s, with a total number of iterations of 3, 30 and 350 respectively. For this dataset the forward- and backprojections took 338 ms and 278 ms.

### 7.4.5 TV-minimization using the Chambolle–Pock algorithm

For images consisting of large homogeneous regions, Total Variation based priors are commonly used. Formally, the (anisotropic) total variation of an image is defined as:

$$\mathrm{TV}_{\ell_1}(\boldsymbol{x}) = \sum_{i=1}^{m}\sum_{j=2}^{n} |x_{(i-1)n+j} - x_{(i-1)n+j-1}| + \sum_{i=2}^{m}\sum_{j=1}^{n} |x_{(i-1)n+j} - x_{(i-2)n+j}| \qquad (7.11)$$

We assume that the image vector corresponds to an $m \times n$ image, stored row-wise, where $N = mn$.

In practice, TV-minimization is often applied in a generic minimization approach where the data consistency term is mixed with the TV-norm:

$$\underset{\boldsymbol{x} \in \mathbb{R}_+^N}{\text{minimize}} \; \frac{1}{2} \|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{p}\|_2^2 + \lambda \mathrm{TV}_{\ell_1}(\boldsymbol{x}). \tag{7.12}$$

Note that $\mathbb{R}_+^N$ denotes the set of nonnegative real numbers.

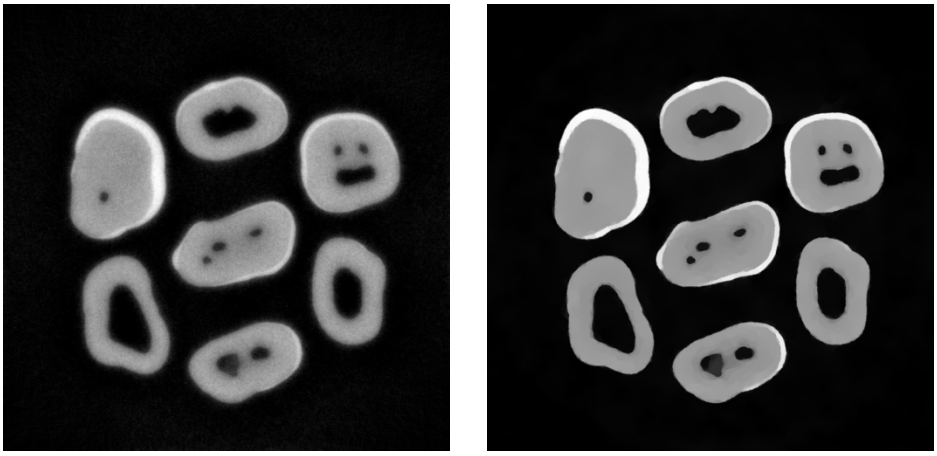
One method for solving this convex minimization problem was presented by Chambolle and Pock [CP11]. Their approach is to use a primal-dual formulation of the problem. The algorithm they propose is short and has few parameters.

Sidky *et al.* discussed and elaborated on the Chambolle–Pock TV-minimization method applied to the tomography problem [SJP12]. They present the essential part of the algorithm in four lines of pseudocode. Compared to other TV-minimization algorithms, for example based on FISTA [BT09], the implementation in a Matlab environment is straightforward. Because it only uses matrix-vector operations, Spot operators can be used.

In addition to the projection operator, a discrete TV operator based on Eq. (7.11) is needed, which computes horizontal and vertical differences of an image. The TV operator can either be constructed from a diagonal band matrix, with 1 and $-1$ on the (sub)diagonals, or it can be formulated as an image processing step using a convolution. In this case, the vertical differences are computed using a convolution of the filter $[-1, 1]$ and the image. In the horizontal difference operator, this filter is simply transposed. To implement the Chambolle–Pock TV-minimization algorithm, we chose to construct a TV Spot operator `opTV` based on the image convolution, because it is fast.

We applied the algorithm to a dataset from the ESRF synchrotron facility. This dataset was recorded at beamline ID19, which is dedicated for high-resolution diffraction topography. The monochromatic source's energy level was 60 keV. In total 1500 projection images were measured from seven teeth, each image having a resolution of $2048 \times 290$. For this dataset, which has been obtained using a high beam intensity, a standard FBP reconstruction can provide very accurate results. To make the reconstruction problem more challenging, Poisson noise was applied to the experimental projection data to simulate a dataset with a low signal-to-noise ratio. TV-minimization is not only an effective technique to apply on limited angle datasets, but should enhance the quality of reconstructions for noisy datasets as well.

For this example, we focus on the reconstruction of a single slice and run the Chambolle–Pock algorithm for 200 iterations. Even for a single slice, the Chambolle–Pock algorithm would require at least 23 GB memory if matrices were formed explicitly (in single precision). Therefore, on most workstations this Matlab code would not have been applicable to this dataset without using the ASTRA toolbox. And without using the `opTomo` and `opTV` operators, the implementation would have required substantially more effort. The reconstruction was run on a workstation with GTX 570 GPU (see Section 7.5.1 for details of the hardware) and took 248 seconds. The forward and backprojections took 213 ms



(a) LSQR solution (b) TV-min solution

Figure 7.6: Reconstructed slice for the ESRF teeth dataset.

and 245 ms respectively. The TV Spot operator opTV does not employ the GPU, but is performed in Matlab. Multiplication by this operator takes approximately 80 ms.

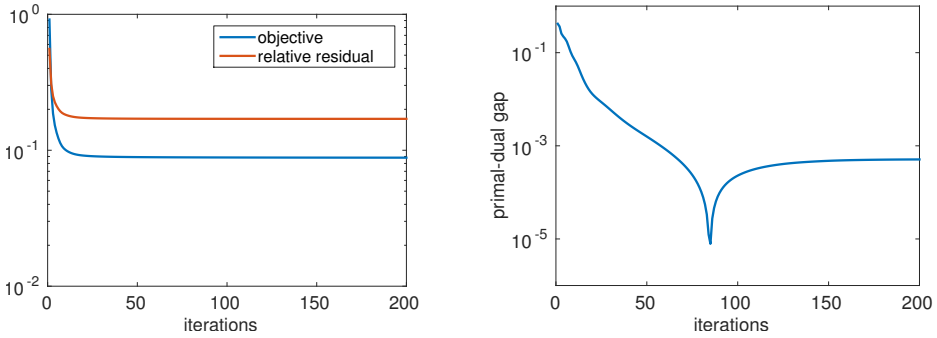
The reconstructions are shown in Fig. 7.6. The sparse gradient prior results in a much sharper reconstruction that has more homogeneous areas of constant gray values. Moreover, the background is completely black, due to the nonnegativity constraint specified in Eq. (7.12) and supported by the Chambolle–Pock algorithm. Also the noise does not affect the reconstruction as much as it does in a least squares solution.

To assess the convergence of the Chambolle–Pock algorithm we have plotted the relative residual and the objective function from Eq. (7.12) in Fig. 7.7a. In Fig. 7.7b, the (conditional) primal-dual gap is shown [SJP12]. Note that the relative residual does not converge to zero, due to the noise in the projection data. The primal-dual gap is initially negative, but becomes positive after about 80 iterations and converges to $5 \times 10^{-4}$.

This example illustrates that using the Spot operator, the pseudocode given in the paper from Sidky *et al.* [SJP12], can directly benefit from the fast GPU ASTRA back end, which enables application of Chambolle–Pock to a real experimental dataset.

## 7.5 Performance benchmarks

In this section we show benchmarks of the GPU and CPU code of the ASTRA toolbox in combination with the Spot operator. Since the memory use of explicit matrices is the key limitation in standard Matlab code, we compare memory usage of the major components of the forward projection operator, such as copies of




(a) convergence of relative residual and objective

(b) absolute value of the primal-dual gap

Figure 7.7: Convergence results of the Chambolle–Pock algorithm applied to the experimental teeth dataset from ESRF. The primal-dual gap starts out negative and around 80 iterations becomes positive which explains the dip in the absolute value of the primal-dual gap (on a logarithmic scale).

volume data and projection data, with that of explicit Matlab matrices. Finally, we measure the computational overhead that is introduced by the Spot toolbox, when using the `opTomo` operator

### 7.5.1 The forward and backprojection operations

The code that we run as a benchmark is line 5 of Listing 7.2. All the benchmarks were timed using the `timeit` function of Matlab, which takes care of "warming up" the CPU and GPU. The wall clock time is averaged over the total number of runs, which is chosen automatically by `timeit`. The machine we used for benchmarking was a Linux workstation with an Intel Core i7-2600K@3.4 GHz CPU with 16 GB of system RAM and a NVIDIA GTX 570 GPU. We compared the results with a TESLA C2070 GPU in a server machine with Intel Core i7-3930K@3.2 GHz and 64 GB of system RAM. We report results only for the Linux version of the software. For a similar workstation running the Windows operating system (also supported by the ASTRA toolbox), similar results were observed. A pre-release of the ASTRA toolbox version 1.6 was used.

First we explicitly form the system matrix as a sparse Matlab matrix. This can be done using a utility function in ASTRA. Then we also time the code for two `opTomo` operators: one of type `'cuda'`, and one of type `'linear'`. Both of these generate the matrix elements on the fly based on the slice-interpolation kernel [Jos82]. The CUDA version uses the GPU, while the other uses the CPU code of the ASTRA toolbox. The Shepp–Logan phantom was used in both 2D and 3D cases, with sizes $n \times n$ and $n \times n \times n$ respectively. A slice of the phantom is shown in Fig. 7.5a. For each experiment, a square (or line in 2D) detector was used that matches the width and height of the phantom in combination with a parallel beam geometry. The number of angles was fixed at 100.

In Fig. 7.8, the timings of the 2D forward projection are shown for different sizes of the phantom. We were able to use explicit matrices in Matlab up to phantom sizes of $2048 \times 2048$, above which the explicit system matrix no longer fits in memory. For these data, 9 GB of memory was required. The ASTRA CPU code is somewhat slower than the use of explicit matrices, which is expected due to the need to generate matrix elements and overhead from the Spot operator. Also note that the GPU code outperforms the CPU code for image sizes larger than $32 \times 32$.

In the 3D case it is not practical to form matrices and we omit the use of explicit Matlab matrices. Instead we compare two different GPUs, the GTX 570 and TESLA C2070. The results show that both cards are performing similarly, but the TESLA card can reconstruct a larger volume of $512 \times 512 \times 512$, since it has more memory (5.4 GB compared to 1.3 GB).

The estimated memory use of the forward projection is listed in Table 7.2 (in terms of data elements). It has been determined as follows: for the forward projection, the volume and its projections need to be stored. For the 2D case, this is $n^2$ for the volume and $kn$ for the projections, where $k$ is the number of angles. Similarly, the 3D volume consists of $n^3$ voxels and projections are $kn^2$. For the Matlab code, additionally the matrix should be stored, which has the same number of rows as detector measurements and the same number of columns as volume pixels/voxels. If we assume that at most $3n$ voxels have a nonzero contribution to a detector measurement, this adds $3n$ times the number of detector elements to the storage requirements. In Fig. 7.9 we have plotted the memory use corresponding to Table 7.2 for single precision floats. Note that the GPU code uses twice the storage for the input of the forward projection algorithm (which is the volume itself). This is because GPU textures are used to speed up data access.

We also benchmarked the backprojection operation of the ASTRA toolbox using the Spot operator. The results are shown in Fig. 7.10 and they are comparable to the timings of the forward projection. The memory use of the backprojection operation is the same as the forward projection, except that the GPU code needs to store the input twice and output once. For the backprojection, the input corresponds to the projection data, which is usually somewhat smaller than the volume.

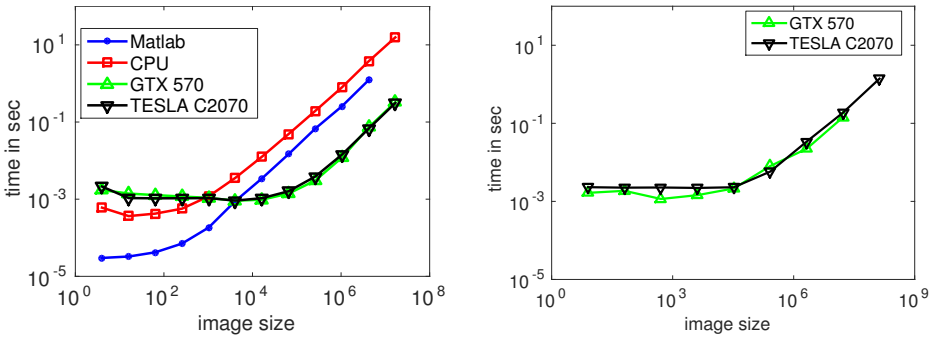
Nowadays, CPU memory is limited to several 10s of gigabytes in a regular workstation, or up to hundreds of gigabytes in a high-end server. For example, 32 GB RAM limits the image size to approximately $300 \times 300 \times 300$ for explicit matrices. Whereas GPU RAM is limited to 12 GB of RAM. However, since the memory requirements of the GPU code are lower, the maximum image size is now $900 \times 900 \times 900$, for this amount of RAM. We see a clear improvement in the maximum size of datasets that can be handled by a single GPU.

### 7.5.2 Overhead of the Spot toolbox

The use of the Spot operator `opTomo` results in computational overhead due to the Spot toolbox, compared to using the ASTRA toolbox directly. In order to quantify

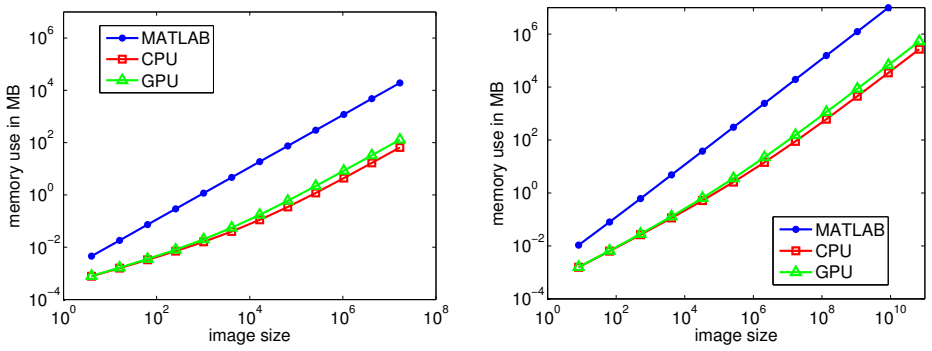
Table 7.2: Memory use of the forward projection operation

| dimensions | Matlab | ASTRA CPU | ASTRA GPU |
|---|---|---|---|
| 2D | $(3k+1)n^2+kn$ | $n^2+kn$ | $2n^2+kn$ |
| 3D | $(3k+1)n^3+kn^2$ | $n^3+kn^2$ | $2n^3+kn^2$ |



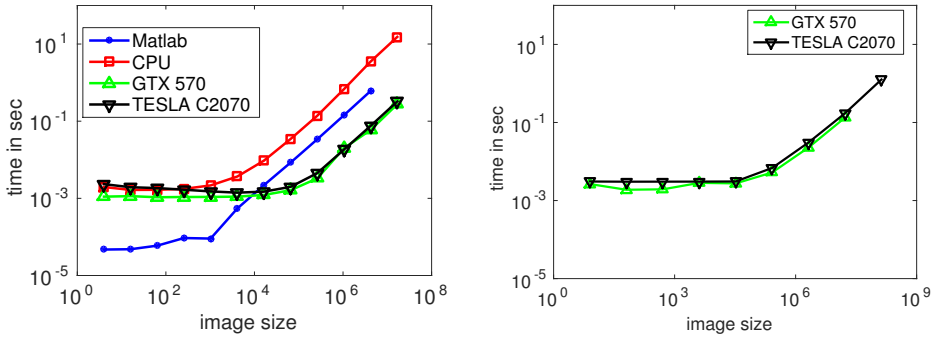

(a) 2D (b) 3D

Figure 7.8: Performance of the forward projection operator. The image size represents the total number of pixels/voxels in the volume.




(a) 2D (b) 3D

Figure 7.9: Memory use of the forward projector. The image size represents the total number of pixels/voxels in the volume.




(a) 2D (b) 3D

Figure 7.10: Performance of the backprojection operator. The image size represents the total number of pixels/voxels in the volume.

this overhead, we benchmark the LSQR algorithm as provided by Matlab using three different ways to call it:

1. using `opTomo`
2. a version using ASTRA's Matlab interface optimized for code length, referred to as *ASTRA util*,
3. a version using ASTRA's Matlab interface optimized for speed, referred to as *ASTRA optim*.

LSQR allows the use of a *function handle* instead of a matrix. The function handle, `afun`, is specified, such that `afun(x,'notransp')` returns `A*x` and `afun(x,'transp')` returns `A'*x`. The second variant of LSQR (ASTRA util) uses a function handle of the ASTRA utility functions for the forward and backprojection, as show in Listing 7.5. This can be done with a few lines of code and is a straightforward implementation which is most likely used in practice. However, these functions allocate memory and set up the algorithm whenever they are called, which is every LSQR iteration.

In the third variant (ASTRA optim) we use an implementation at a lower level that preallocates memory and sets up the algorithm, the same as is done in the operator `opTomo` internally. However, this optimized code requires substantially more lines of code. Because the ASTRA code in this version is the same as used in `opTomo`, it allows us to see the exact overhead caused by the Spot toolbox.

In Table 7.3 and Table 7.4 the computation times of the three variants of LSQR using 20 iterations are shown for various sizes of the volume/detector. The same settings are used as in the previous paragraph. The code was run on the workstation with the GTX 570 GPU.

In the 2D case, shown in Table 7.3, ASTRA util and `opTomo` have comparable run times. If we look at the overhead caused by Spot by comparing ASTRA optim and `opTomo` we see that for small $n$ the overhead is large. For $n = 32$, LSQR using

Table 7.3: 2D: LSQR runtimes in seconds

| n | ASTRA optim | ASTRA util | opTomo |
|---|---|---|---|
| 32 | 0.020 | 0.039 | 0.040 |
| 64 | 0.022 | 0.042 | 0.044 |
| 128 | 0.036 | 0.057 | 0.060 |
| 256 | 0.085 | 0.112 | 0.111 |
| 512 | 0.278 | 0.308 | 0.306 |
| 1024 | 1.137 | 1.140 | 1.178 |
| 2048 | 4.885 | 5.030 | 5.040 |

Table 7.4: 3D: LSQR runtimes in seconds

| n | ASTRA optim | ASTRA util | opTomo |
|---|---|---|---|
| 32 | 0.138 | 0.176 | 0.158 |
| 64 | 0.408 | 0.444 | 0.437 |
| 128 | 2.379 | 2.519 | 2.434 |
| 256 | 15.615 | 16.506 | 15.643 |

`opTomo` is twice as slow compared to LSQR using function handle ASTRA optim. However, the total overhead is only 20 ms. If the amount of work is increased the relative overhead becomes drastically smaller. For a data size of $n = 2048$, the overhead of the `opTomo` operator compared to ASTRA optim is 155 ms. Relative to the total runtime, this overhead is 3%.

In 3D, see Table 7.4, the amount of work compared to the overhead becomes large. In this case, the ASTRA utility functions are slower than `opTomo` due to overhead from memory allocation and initializing the forward and backprojection algorithms. The overhead of `opTomo` compared to ASTRA optim is still in the order of 20 ms. Therefore, the relative overhead with respect to total runtime is almost negligible in 3D. For $n = 256$, the relative overhead is 0.2%.

In Listing 7.5 and Listing 7.6 we compare the amount of code that is needed for ASTRA util and `opTomo`. We omit the code used for ASTRA optim, since it requires about 50 lines of code. From these code snippets it is clear that the `opTomo` operator can hide a lot of code and interface details that were necessary for implementing ASTRA util. Moreover, the optimizations used in ASTRA optim are also part of the `opTomo` Spot operator.

## 7.6 Discussion and conclusions

Advances in hardware for tomographic projection acquisition have led to an increase in data sizes. At the same time, advances in computational methods for limited data reconstruction have resulted in a broad range of algorithms that are powerful, yet highly computationally demanding. As a result, reconstruction

Listing 7.5: LSQR ASTRA util

```
% set up data size (100 projection angles)
vsize = [n,n,n];
psize = [n,100,n];
% set up function handle
f = @(x, type) Afun2(x, type, vsize, psize, proj_geom, ...
    vol_geom);
% create forward projections
p = f(im, 'notransp');
% solve with lsqr
x = lsqr(f, p);

..

% used for function handle f
function y = Afun2(x, type, vsize, psize, proj_geom, ...
    vol_geom)
  if strcmp(type, 'notransp')
    % vector to volume
    x = reshape(x, vsize);
    % fp
    [y_id,y] = astra_create_sino3d_cuda(x, proj_geom, ...
        vol_geom);
    astra_mex_data3d('delete', y_id);
  else
    % vector to projections
    x = reshape(x, psize);
    % bp
    [y_id, y] = astra_create_backprojection3d_cuda(x, ...
        proj_geom, vol_geom);
    astra_mex_data3d('delete', y_id);
  end
  y = y(:);
end
```

Listing 7.6: LSQR opTomo

```
% create Spot operator
W = opTomo('cuda', proj_geom, vol_geom);
% create forward projections
p = W*im(:);
% solve with lsqr
x = lsqr(W,p);
```

software has to be implemented for parallel computation architectures such as computer clusters and graphics processing units (GPUs) to handle large-scale datasets efficiently.

Many novel reconstruction algorithms are prototyped in a high-level scripting language such as Matlab. The syntax for these languages can be similar to mathematical notation, which makes prototyping easier. Due to the nature of high-level languages, these implementations are often not suitable to apply on large-scale datasets. For tomographic datasets, the system matrix corresponding to the linear model of the forward projection $\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p}$, is too large to store explicitly for even moderately sized datasets. The `opTomo` operator is able to bridge the gap between the flexibility of Matlab scripts on one hand, and the fast and scalable GPU back end of the ASTRA toolbox. The `opTomo` operator allows using matrix syntax to call the fast GPU projection and backprojection implementations of the ASTRA toolbox. By overloading many common matrix operations, the Spot toolbox delivers an effective framework for linear operators.

We remark that the `opTomo` operator exposes ASTRA's highly efficient implementations of the forward and backprojection operations, but when more detailed access to the matrix is required, the user may find that certain operations are either computationally inefficient or simply not implemented. This includes operations that work element-wise on the matrix, such as element-wise multiplication. Therefore, Matlab scripts that rely on such operations, for example to compute column or row norms, are currently not supported. Usually, it is possible to work around these limitations and as the ASTRA toolbox is continuously evolving, efforts are currently ongoing to extend the range of operations for which the `opTomo` operator provides a high level of efficiency.

In our benchmarks we have seen that a very small overhead is paid by using the Spot operator for ASTRA. As the data sizes increase the relative overhead becomes negligible and should not pose any problems.

Our software is freely available under an open source license (GPL), enabling easy implementation of novel advanced reconstruction algorithms in materials science, biomedical imaging, and other fields.

## 7.6 Availability of source code and data

The ASTRA toolbox and `opTomo` operator can be downloaded as open source software [15]. The Spot toolbox is available separately [BF14].

## 7.A Geometries in the ASTRA toolbox

In this section we describe the use of volume and projection geometries for the ASTRA toolbox. Since the 2D geometries can be embedded into a 3D geometry, we will not discuss these separately.

### 7.A.1 Volume geometry

The volume geometry describes the dimensions of the reconstruction area in terms of voxels. The reconstruction volume is always centered around the origin, and its voxels are cubes of unit size. This defines the Cartesian coordinate system in which the rest of the geometry is specified. To set up a reconstruction volume we can use the following Matlab code:

```
vol_geom = astra_create_vol_geom(vx, vy, vz);
```

which results in a structure that contains the size of the reconstruction volume.

### 7.A.2 3D parallel beam

The parallel beam geometry for the 3D case is similar to the 2D case, except that the detector is two dimensional.

In the ASTRA toolbox this geometry can be specified in one of two ways:

```
proj_geom = astra_create_proj_geom('parallel3d', ...
    det_spacing_x, det_spacing_y, det_row_count, ...
    det_col_count, angles);
proj_geom_vec = astra_create_proj_geom('parallel3d_vec', ...
    det_row_count, det_col_count, vectors);
```

In the first case, a circular path (rotating around the $z$-axis) of the source and detector is used. The parameters `det_spacing_x` and `det_spacing_y` specify the distance between two adjacent detector pixels. The `det_row_count` and `det_col_count` determine the number of rows and columns of the detector. The `angle` array contains all angles in radians at which projections are measured.

In the second case, besides the number of rows and columns of the detector, an array `vectors` is passed. This array has $K$ rows, one for each angle. A row contains the following parameters in order:

- `rayX`, `rayY`, `rayZ`. This vector gives the direction of the rays.
- `dx`, `dy`, `dz`. These are the $x$, $y$ and $z$ coordinates of the center of the detector.
- `ux`, `uy`, `uz`. The vector from detector pixel $(0,0)$ to $(0,1)$.
- `vx`, `vy`, `vz`. The vector from detector pixel $(0,0)$ to $(1,0)$.

In Fig. 7.11, we show an example of the geometric parameters. In this example, a projection is taken at an angle of 0°. Note that this specifies the acquisition of one projection image and therefore, the parameters for all other angles should be passed as well. Using the vectors $\boldsymbol{d}$ and $\boldsymbol{ray}$, many projection acquisition schemes can be parametrized.

Note that it is not necessary for the detector and volume pixel to have the same size, although this setting is commonly used for reconstructions.

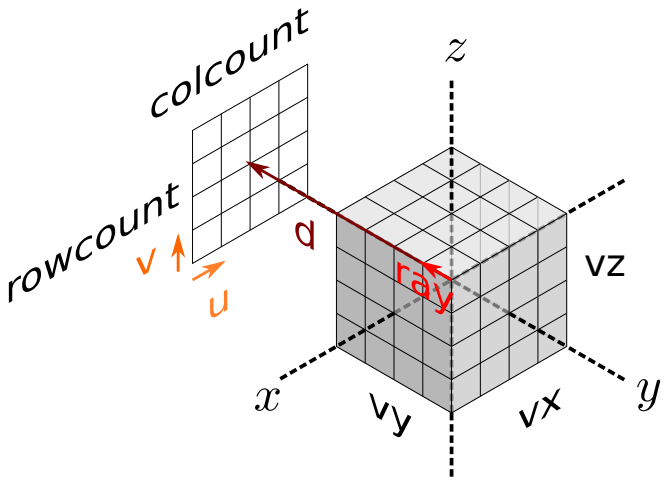



Figure 7.11: Illustration of the parameters for the volume and 3D parallel beam projection geometries.

### 7.A.3 Cone beam

Setting up a cone beam geometry is very similar to setting up a 3D parallel beam geometry. In this case, the rays are not parallel, but they originate from a single point: the ray source. Therefore, instead of indicating the ray direction, the center of the ray source is passed.

```
1    proj_geom = astra_create_proj_geom('cone', det_spacing_x, ...
         det_spacing_y, det_row_count, det_col_count, angles, ...
         source_origin, origin_det);
2    proj_geom = astra_create_proj_geom('cone_vec', ...
         det_row_count, det_col_count, vectors);
```

# Bibliography


[15] *ASTRA - Tomographic Reconstruction toolbox*. https://github.com/astra-toolbox/astra-toolbox. [Online; accessed 1-March-2015]. 2015.

[ABS12] W. van Aarle, K. J. Batenburg, and J. Sijbers. "Automatic parameter estimation for the discrete algebraic reconstruction technique (DART)". In: *IEEE Transactions on Image Processing* 21(11) (2012), pp. 4608–4621.

[Aer+11] S. van Aert, K. J. Batenburg, M. D. Rossell, R. Erni, and G. van Tendeloo. "Three-dimensional atomic imaging of crystalline nanoparticles". In: *Nature* 470(7334) (2011), pp. 374–377.

[AF11] J. Agulleiro and J.-J. Fernandez. "Fast tomographic reconstruction on multicore computers". In: *Bioinformatics* 27(4) (2011), pp. 582–583.

[AK84] A. H. Andersen and A. C. Kak. "Simultaneous algebraic reconstruction technique (SART): A superior implementation of the ART algorithm". In: *Ultrasonic Imaging* 6(1) (1984), pp. 81–94.

[AL12] A. Y. Aravkin and T. van Leeuwen. "Estimating nuisance parameters in inverse problems". In: *Inverse Problems* 28(11) (2012), pp. 115016–115028.

[Alp+13] A. Alpers, R. J. Gardner, S. König, R. S. Pennington, C. B. Boothroyd, L. Houben, R. E. Dunin-Borkowski, and K. J. Batenburg. "Geometric reconstruction methods for electron tomography". In: *Ultramicroscopy* 128 (2013), pp. 42–54.

[ATM06] I. Arslan, J. R. Tong, and P. A. Midgley. "Reducing the missing wedge: High-resolution dual axis tomography of inorganic materials". In: *Ultramicroscopy* 106(11) (2006), pp. 994–1000.

[Bal+11] S. Bals, M. Casavola, M. A. van Huis, S. van Aert, K. J. Batenburg, G. van Tendeloo, and D. Vanmaekelbergh. "Three-dimensional atomic imaging of colloidal core–shell nanocrystals". In: *Nano Letters* 11(8) (2011), pp. 3420–3424.

[BAS11] K. J. Batenburg, W. van Aarle, and J. Sijbers. "A semi-automatic algorithm for grey level estimation in tomography". In: *Pattern Recognition Letters* 32(9) (2011), pp. 1395–1405.

[Bat05] K. J. Batenburg. "An evolutionary algorithm for discrete tomography". In: *Discrete Applied Mathematics* 151(1) (2005), pp. 36–54.

[BB13] F. Bleichrodt and K. Batenburg. "Automatic optimization of alignment parameters for tomography datasets". In: *Image Analysis*. Vol. 7944. LNCS. Springer, 2013, pp. 489–500.

[BE79] Å. Björck and T. Elfving. "Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations". In: *BIT Numerical Mathematics* 19(2) (1979), pp. 145–163.

[BF08] E. van den Berg and M. P. Friedlander. "Probing the Pareto frontier for basis pursuit solutions". In: *SIAM Journal on Scientific Computing* 31(2) (2008), pp. 890–912.

[BF11] F. E. Boas and D. Fleischmann. "Evaluation of two iterative techniques for reducing metal artifacts in computed tomography". In: *Radiology* 259(3) (2011), pp. 894–902.

[BF12] F. E. Boas and D. Fleischmann. "CT artifacts: Causes and reduction techniques". In: *Imaging in Medicine* 4(2) (2012), pp. 229–240.

[BF14] E. van den Berg and M. P. Friedlander. *Spot - A Linear-Operator Toolbox*. July 2014. url: http://www.cs.ubc.ca/labs/scl/spot/.

[BHE01] S. Brandt, J. Heikkonen, and P. Engelhardt. "Automatic alignment of transmission electron microscope tilt series without fiducial markers". In: *Journal of Structural Biology* 136(3) (2001), pp. 201–213.

[Bjö96] Å. Björck. *Numerical methods for least squares problems*. SIAM, 1996.

[Ble+13] F. Bleichrodt, J. Sijbers, J. de Beenhouwer, and K. J. Batenburg. "An alignment method for fan beam tomography". In: *Tomography of Materials and Structures*. Ghent University press, 2013, pp. 103–106.

[Ble+14] F. Bleichrodt, J. De Beenhouwer, J. Sijbers, and K. J. Batenburg. "Aligning projection images from binary volumes". In: *Fundamenta Informaticae* 135(1) (2014), pp. 21–42.

[Ble+15] F. Bleichrodt, T. van Leeuwen, W. J. Palenstijn, W. van Aarle, J. Sijbers, and K. J. Batenburg. "Easy implementation of advanced tomography algorithms using the ASTRA toolbox with Spot operators". In: *Numerical Algorithms* (2015), pp. 1–25.

[Bri+10] H. M. Britz, J. Jokihaara, O. V. Leppänen, T. Järvinen, and D. M. L. Cooper. "3D visualization and quantification of rat cortical bone porosity using a desktop micro-CT system: A case study in the tibia". In: *Journal of Microscopy* 240(1) (2010), pp. 32–37.

[Bro+12] W. van den Broek, A. Rosenauer, B. Goris, G. T. Martinez, S. Bals, S. van Aert, and D. van Dyck. "Correction of non-linear thickness effects in HAADF STEM electron tomography". In: *Ultramicroscopy* 116 (2012), pp. 8–12.

[BS11] K. J. Batenburg and J. Sijbers. “DART: A practical reconstruction algorithm for discrete tomography”. In: *IEEE Transactions on Image Processing* 20(9) (2011), pp. 2542–2553.

[BT09] A. Beck and M. Teboulle. “A fast iterative shrinkage-thresholding algorithm for linear inverse problems”. In: *SIAM Journal on Imaging Sciences* 2(1) (2009), pp. 183–202.

[BTB14] F. Bleichrodt, F. Tabak, and K. J. Batenburg. “SDART: An algorithm for discrete tomography from noisy projections”. In: *Computer Vision and Image Understanding* 129 (2014), pp. 63–74.

[Car+12] S. Carmignato, A. Pierobon, P. Rampazzo, M. Parisatto, and E. Savio. “CT for industrial metrology-accuracy and structural resolution of CT dimensional measurements”. In: *Conference on Industrial Computed Tomography (ICT), Wels*. 2012, pp. 19–21.

[Cas+02] E. van de Casteele, D. van Dyck, J. Sijbers, and E. Raman. “An energy-based beam hardening model in tomography”. In: *Physics in Medicine and Biology* 47(23) (2002), p. 4181.

[Cas+12] M. Casavola, M. A. van Huis, S. Bals, K. Lambert, Z. Hens, and D. Vanmaekelbergh. “Anisotropic cation exchange in PbSe/CdSe core/shell nanocrystals of different geometry”. In: *Chemistry of Materials* 24(2) (2012), pp. 294–302.

[Cha11] R. Chartrand. “Numerical differentiation of noisy, nonsmooth data”. In: *ISRN Applied Mathematics* 2011 (2011).

[Chi+11] S. Chilingaryan, A. Mirone, A. Hammersley, C. Ferrero, L. Helfen, A. Kopmann, T. dos Santos Rolo, and P. Vagovic. “A GPU-based architecture for real-time data assessment at synchrotron experiments”. In: *IEEE Transactions on Nuclear Science* 58(4) (2011), pp. 1447–1455.

[CP11] A. Chambolle and T. Pock. “A first-order primal-dual algorithm for convex problems with applications to imaging”. In: *Journal of Mathematical Imaging and Vision* 40(1) (2011), pp. 120–145.

[CRT06] E. J. Candès, J. Romberg, and T. Tao. “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information”. In: *IEEE Transactions on Information Theory* 52(2) (2006), pp. 489–509.

[DB04] B. De Man and S. Basu. “Distance-driven projection and backprojection in three dimensions”. In: *Physics in Medicine and Biology* 49(11) (2004), pp. 2463–2475.

[Die+92] K. Dierksen, D. Typke, R. Hegerl, A. J. Koster, and W. Baumeister. “Towards automatic electron tomography”. In: *Ultramicroscopy* 40(1) (1992), pp. 71–87.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum likelihood from incomplete data via the EM algorithm”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39(1) (1977), pp. 1–38.

[Don+06] P. C. J. Donoghue, S. Bengtson, X.-p. Dong, N. J. Gostling, T. Huldtgren, J. A. Cunningham, C. Yin, Z. Yue, F. Peng, and M. Stampanoni. "Synchrotron X-ray tomographic microscopy of fossil embryos". In: *Nature* 442(7103) (2006), pp. 680–683.

[Fit+99] E. E. Fitchard, J. S. Aldridge, P. J. Reckwerdt, and T. R. Mackie. "Registration of synthetic tomographic projection data sets using cross-correlation". In: *Physics in Medicine and Biology* 43(6) (1999), p. 1645.

[FNW07] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems". In: *IEEE Journal of Selected Topics in Signal Processing* 1(4) (2007), pp. 586–597.

[Fra92] J. Frank. *Electron tomography: Three-dimensional imaging with the transmission electron microscope*. Plenum Press, 1992.

[FS11] D. C.-L. Fong and M. Saunders. "LSMR: An iterative algorithm for sparse least-squares problems". In: *SIAM Journal on Scientific Computing* 33(5) (2011), pp. 2950–2971.

[GB08] J. Gregor and T. Benson. "Computational analysis and improvement of SIRT". In: *IEEE Transactions on Medical Imaging* 27(7) (2008), pp. 918–924.

[GBH70] R. Gordon, R. Bender, and G. T. Herman. "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography". In: *Journal of Theoretical Biology* 29(3) (1970), pp. 471–481.

[Gil72] P. Gilbert. "Iterative methods for the three-dimensional reconstruction of an object from projections". In: *Journal of Theoretical Biology* 36(1) (1972), pp. 105–117.

[GO09] T. Goldstein and S. Osher. "The split Bregman method for L1-regularized problems". In: *SIAM Journal on Imaging Sciences* 2(2) (2009), pp. 323–343.

[Gon15] L. C. Gontard. "Removing the effects of the "dark matter" in tomography". In: *Ultramicroscopy* 154 (2015), pp. 64–72.

[Grü+03] K. Grünewald, P. Desai, D. C. Winkler, J. B. Heymann, D. M. Belnap, W. Baumeister, and A. C. Steven. "Three-dimensional structure of herpes simplex virus from cryo-electron tomography". In: *Science* 302(5649) (2003), pp. 1396–1398.

[GSW00] M. van Geet, R. Swennen, and M. Wevers. "Quantitative analysis of reservoir rocks by microfocus X-ray computerised tomography". In: *Sedimentary Geology* 132(1) (2000), pp. 25–36.

[Gu+06] J. Gu, L. Zhang, G. Yu, Y. Xing, and Z. Chen. "X-ray CT metal artifacts reduction through curvature based sinogram inpainting". In: *Journal of X-ray Science and Technology* 14(2) (2006), pp. 73–82.

[GZ10a] R. Guedouar and B. Zarrad. “A comparative study between matched and mis-matched projection/back projection pairs used with ASIRT reconstruction method”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 619(1) (2010), pp. 225–229.

[GZ10b] R. Guedouar and B. Zarrad. “A new reprojection method based on a comparison of popular reprojection models”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 619(1) (2010), pp. 270–275.

[Ham+05] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust statistics: The approach based on influence functions*. Wiley, 2005.

[HB11] L. Houben and M. Bar Sadan. “Refinement procedure for the image alignment in high-resolution electron tomography”. In: *Ultramicroscopy* 111(9) (2011), pp. 1512–1520.

[Hel99] S. Helgason. *The Radon transform*. 2nd edition. Vol. 5. Progress in Mathematics. Birkhäuser, 1999.

[Her73] G. T. Herman. “Reconstruction of binary patterns from a few projections”. In: *International Computing Symposium*. Vol. 1974. North-Holland Publishing Co., Netherlands. 1973, pp. 371–378.

[HL82] J. M. Hyman and B. Larrouturou. “The numerical differentiation of discrete functions using polynomial interpolation methods”. In: *Applied Mathematics and Computation* 10 (1982), pp. 487–506.

[Hu+14] Q. Hu, M. T. Ley, J. Davis, J. C. Hanan, R. Frazier, and Y. Zhang. “3D chemical segmentation of fly ash particles with X-ray computed tomography and electron probe microanalysis”. In: *Fuel* 116 (2014), pp. 229–236.

[IC88] J. D. Ingle Jr. and S. R. Crouch. *Spectrochemical analysis*. Prentice Hall, 1988.

[Jan+09] B. Jang, D. Kaeli, S. Do, and H. Pien. “Multi GPU implementation of iterative tomographic reconstruction algorithms”. In: *IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2009. ISBI’09.* IEEE. 2009, pp. 185–188.

[Jos82] P. M. Joseph. “An improved algorithm for reprojecting rays through pixel images”. In: *IEEE Transactions on Medical Imaging* 1(3) (1982), pp. 192–196.

[JS91] Z. Jing and F. Sachs. “Alignment of tomographic projections using an incomplete set of fiducial markers”. In: *Ultramicroscopy* 35(1) (1991), pp. 37–43.

[KDH13] J. Klukowska, R. Davidi, and G. T. Herman. “SNARK09–A software package for reconstruction of 2D images from 1D projections”. In: *Computer Methods and Programs in Biomedicine* 110(3) (2013), pp. 424–440.

[Kin+11] A. Kingston, A. Sakellariou, T. Varslot, G. Myers, and A. Sheppard. “Reliable automatic alignment of tomographic projection data by passive auto-focus”. In: *Medical Physics* 38(9) (2011), pp. 4934–4945.

[KMM96] J. R. Kremer, D. N. Mastronarde, and J. R. McIntosh. “Computer visualization of three-dimensional image data using IMOD”. In: *Journal of Structural Biology* 116(1) (1996), pp. 71–76.

[KS01] A. C. Kak and M. Slaney. *Principles of computerized tomographic imaging*. Vol. 33. Classics in Applied Mathematics. SIAM, 2001.

[KSZ11] A. Katsevich, M. Silver, and A. Zamyatin. “Local tomography and the motion estimation problem”. In: *SIAM Journal on Imaging Sciences* 4(1) (2011), pp. 200–219.

[Küb+05] C. Kübel, A. Voigt, R. Schoenmakers, M. Otten, D. Su, T.-C. Lee, A. Carlsson, and J. Bradley. “Recent advances in electron tomography: TEM and HAADF-STEM tomography for materials science and semiconductor applications”. In: *Microscopy and Microanalysis* 11(5) (2005), pp. 378–400.

[Kym+03] A. Z. Kyme, B. F. Hutton, R. L. Hatton, D. W. Skerrett, and L. R. Barnden. “Practical aspects of a data-driven motion correction approach for brain SPECT”. In: *IEEE Transactions on Medical Imaging* 22(6) (2003), pp. 722–729.

[Lew92] R. M. Lewitt. “Alternatives to voxels for image representation in iterative reconstruction algorithms”. In: *Physics in Medicine and Biology* 37(3) (1992), pp. 705–716.

[Mar+11] H. Markötter, I. Manke, P. Krüger, T. Arlt, J. Haussmann, M. Klages, H. Riesemeier, C. Hartnig, J. Scholta, and J. Banhart. “Investigation of 3D water transport paths in gas diffusion layers by combined in-situ synchrotron X-ray radiography and tomography”. In: *Electrochemistry Communications* 13(9) (2011), pp. 1001–1004.

[Mas+13] B. Masschaele, M. Dierick, D. van Loo, M. N. Boone, L. Brabant, E. Pauwels, V. Cnudde, and L. van Hoorebeke. “HECTOR: A 240kV micro-CT setup optimized for research”. In: *Journal of Physics: Conference Series*. Vol. 463. 1. IOP Publishing. 2013, p. 012012.

[MD09] P. A. Midgley and R. E. Dunin-Borkowski. “Electron tomography and holography in materials science”. In: *Nature Materials* 8(4) (2009), pp. 271–280.

[MNT04] K. Madsen, H. B. Nielsen, and O. Tingleff. *Methods for non-linear least squares problems (2nd ed.)* 2004. url: http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3215/pdf/imm3215.pdf.

[Mom+11] A. Momose, W. Yashiro, S. Harasse, and H. Kuwabara. "Four-dimensional X-ray phase tomography with Talbot interferometry and white synchrotron radiation: dynamic observation of a living worm". In: *Optics Express* 19(9) (2011), pp. 8423–8432.

[Mor78] J. J. Moré. "The Levenberg–Marquardt algorithm: Implementation and theory". In: *Numerical analysis*. Vol. 630. Lecture Notes in Mathematics. Springer, 1978, pp. 105–116.

[MW03] P. A. Midgley and M. Weyland. "3D electron microscopy in the physical sciences: the development of Z-contrast and EFTEM tomography". In: *Ultramicroscopy* 96(3) (2003), pp. 413–431.

[Neu97] C. Neubauer. "Intelligent X-ray inspection for quality control of solder joints". In: *IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part C* 20(2) (1997), pp. 111–120.

[NW01] F. Natterer and F. Wübbeling. *Mathematical methods in image reconstruction*. Vol. 5. Monographs on Mathematical Modeling and Computation. SIAM, 2001.

[NW06] J. Nocedal and S. J. Wright. *Numerical optimization*. 2nd edition. Springer Series in Operations Research and Financial Engineering. Springer, 2006, pp. 245–262.

[Pag+02] D. Paganin, S. C. Mayo, T. E. Gureyev, P. R. Miller, and S. W. Wilkins. "Simultaneous phase and amplitude extraction from a single defocused image of a homogeneous object". In: *Journal of microscopy* 206(1) (2002), pp. 33–40.

[Par+12] D. Y. Parkinson, C. Knoechel, C. Yang, C. A. Larabell, and M. A. Le Gros. "Automatic alignment and reconstruction of images for soft X-ray tomography". In: *Journal of Structural Biology* 177(2) (2012), pp. 259–266.

[PBS11] W. J. Palenstijn, K. J. Batenburg, and J. Sijbers. "Performance improvements for iterative electron tomography reconstruction using graphics processing units (GPUs)". In: *Journal of Structural Biology* 176(2) (2011), pp. 250–253.

[PBS13] W. J. Palenstijn, K. J. Batenburg, and J. Sijbers. "The ASTRA Tomography Toolbox". In: *13th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE*. 2013.

[PDX12] Y. Pan, F. De Carlo, and X. Xiao. "Ring artifact removal for microtomography in synchrotron radiation". In: *SPIE proceedings*. Vol. 8313. International Society for Optics and Photonics. 2012, pp. 29–36.

[Ped+10] S. Pedemonte, A. Bousse, K. Erlandsson, M. Modat, S. Arridge, B. F. Hutton, and S. Ourselin. "GPU accelerated rotation-based emission tomography reconstruction". In: *Nuclear Science Symposium Conference Record (NSS/MIC)*. IEEE. 2010, pp. 2657–2661.

[Pre+07] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge Univ Press, 2007.

[PS82] C. C. Paige and M. A. Saunders. "LSQR: An algorithm for sparse linear equations and sparse least squares". In: *ACM Transactions on Mathematical Software (TOMS)* 8(1) (1982), pp. 43–71.

[Riv12] M. L. Rivers. "tomoRecon: High-speed tomography reconstruction on workstations using multi-threading". In: *SPIE Proceedings*. Vol. 8506. International Society for Optics and Photonics. 2012.

[Rop+03] D. Ropers, U. Baum, K. Pohle, K. Anders, S. Ulzheimer, B. Ohnesorge, C. Schlundt, W. Bautz, W. G. Daniel, and S. Achenbach. "Detection of coronary artery stenoses with thin-slice multi-detector row spiral computed tomography and multiplanar reconstruction". In: *Circulation* 107(5) (2003), pp. 664–666.

[Sch+05] T. Schüle, C. Schnörr, S. Weber, and J. Hornegger. "Discrete tomography by convex–concave regularization and D.C. programming". In: *Discrete Applied Mathematics* 151(1) (2005), pp. 229–243.

[Sco+12] M. C. Scott, C.-C. Chen, M. Mecklenburg, C. Zhu, R. Xu, P. Ercius, U. Dahmen, B. C. Regan, and J. Miao. "Electron tomography at 2.4-ångström resolution". In: *Nature* 483(7390) (2012), pp. 444–447.

[Sid85] R. L. Siddon. "Fast calculation of the exact radiological path for a three-dimensional CT array". In: *Medical Physics* 12(2) (1985), pp. 252–255.

[SJP12] E. Y. Sidky, J. H. Jørgensen, and X. Pan. "Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle–Pock algorithm". In: *Physics in Medicine and Biology* 57(10) (2012), pp. 3065–3091.

[SP08] E. Y. Sidky and X. Pan. "Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization". In: *Physics in Medicine and Biology* 53(17) (2008), pp. 4777–4807.

[Sze06] R. Szeliski. "Image alignment and stitching: A tutorial". In: *Foundations and Trends® in Computer Graphics and Vision* 2(1) (2006), pp. 1–104.

[Thi+12] K. Thielemans, C. Tsoumpas, S. Mustafovic, T. Beisel, P. Aguiar, N. Dikaios, and M. W. Jacobson. "STIR: Software for tomographic image reconstruction release 2". In: *Physics in Medicine and Biology* 57(4) (2012), pp. 867–883.

[Vel+10] W. J. Veldkamp, R. M. Joemai, A. J. van der Molen, and J. Geleijns. "Development and validation of segmentation and interpolation techniques in sinograms for metal artifact suppression in CT". In: *Medical Physics* 37(2) (2010), pp. 620–628.

[VW97] P. Viola and W. M. Wells III. "Alignment by maximization of mutual information". In: *International Journal of Computer Vision* 24(2) (1997), pp. 137–154.

[Wan+96] G. Wang, D. L. Snyder, J. A. O'Sullivan, and M. W. Vannier. "Iterative deblurring for CT metal artifact reduction". In: *IEEE Transactions on Medical Imaging* 15(5) (1996), pp. 657–664.

[Wil+11] J. J. Williams, K. E. Yazzie, N. C. Phillips, N. Chawla, X. Xiao, F. De Carlo, N. Iyyer, and M. Kittur. "On the correlation between fatigue striation spacing and crack growth rate: A three-dimensional (3-D) X-ray synchrotron tomography study". In: *Metallurgical and Materials Transactions A* 42(13) (2011), pp. 3845–3848.

[YNP05] C. Yang, E. G. Ng, and P. A. Penczek. "Unified 3-D structure and projection orientation refinement using quasi-Newton algorithm". In: *Journal of Structural Biology* 149(1) (2005), pp. 53–64.

[Zen10] G. Zentai. "X-ray imaging for homeland security". In: *International Journal of Signal and Imaging Systems Engineering* 3(1) (2010), pp. 13–20.

[ZF03] B. Zitová and J. Flusser. "Image registration methods: a survey". In: *Image and Vision Computing* 21(11) (2003), pp. 977–1000.

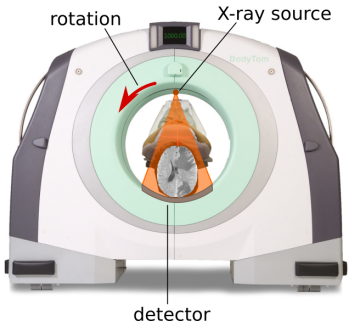
[ZG00] G. L. Zeng and G. T. Gullberg. "Unmatched projector/backprojector pairs in an iterative reconstruction algorithm". In: *IEEE Transactions on Medical Imaging* 19(5) (2000), pp. 548–555.

[Zhu+08] J. Zhu, X. Li, Y. Ye, and G. Wang. "Analysis on the strip-based projection model for discrete tomography". In: *Discrete Applied Mathematics* 156(12) (2008), pp. 2359–2367.

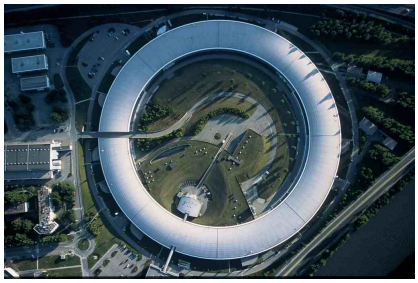
# Samenvatting

Tomografie is een techniek voor het reconstrueren van doorsnedes van objecten zonder deze fysiek open te snijden. Hiervoor wordt gebruik gemaakt van een scanner die bestaat uit een stralingsbron en een detector. Het object bevindt zich tussen de stralingsbron en detector. Op de detector wordt een intensiteitsprofiel gemeten van de straling nadat deze het object heeft gepasseerd. Na een normalisatiestap verkrijgen we een *projectiebeeld* dat de cumulatieve dichtheid weergeeft van het object in de richting van de straling. Door projectiebeelden vanuit verschillende hoeken vast te leggen kan een 3D beeld gevormd worden van de inwendige structuren. Hiervoor is een wiskundige berekening nodig die wordt uitgevoerd volgens een reconstructie algoritme.

Computer tomografie (CT) voor medische diagnostiek is een van de meest bekende toepassingen van tomografie. Hiervoor wordt een CT scanner gebruikt die werkt met Röntgen straling, zoals geïllustreerd in Fig. 7.12. Met Röntgen straling kunnen structuren van kleiner dan een micrometer worden gereconstrueerd, maar voor medische toepassingen wordt meestal gewerkt met een nauwkeurigheid in de orde van een millimeter. Er bestaan ook meer geavanceerde toepassingen van tomografie waarbij het afbeelden van structuren op micrometer of nanometer schaal wel noodzakelijk is, zoals bijvoorbeeld in microbiologie en materiaalkunde (Fig. 7.13). Op zulke kleine schalen is het zeer uitdagend om een reconstructie te verkrijgen van hoge kwaliteit. Dit komt omdat instabiliteiten en verstoringen tijdens het scannen leiden tot onnauwkeurigheden in de berekeningen die nodig




Figuur 7.12: Medische CT-scanner.



(a) ESRF synchrotron

(b) FEI elektronenmicroscoop, Berkeley lab

Figuur 7.13: Geavanceerde toepassingen van tomografie.

zijn voor een reconstructie.

Het onderzoek in dit proefschrift is erop gericht om reconstructie algoritmes robuuster te maken ten opzichte van deze instabiliteiten en verstoringen.

Eén van de uitdagingen van tomografie op kleine schaal is het voorkomen van ongewenste bewegingen en rotaties van onderdelen van de scanner of van het gescande voorwerp. Deze kunnen leiden tot onzekerheden in de geometrie van de opname van de projectiebeelden, wat leidt tot *artefacten* in de reconstructie. Dit zijn beeldelementen die niet overeenkomen met fysieke kenmerken van het gescande object. De geometrie die gebruikt wordt in het reconstructie algoritme komt in dit geval niet overeen met de werkelijke geometrie. In hoofdstuk 2 en 3 beschrijven we twee verschillende methodes om de werkelijke geometrie te benaderen met behulp van de projectiebeelden.

Een ander probleem is ruis. Ruis treedt onder andere op wanneer fotonen, die je ook kunt beschouwen als “stralingsdeeltjes”, niet langs rechte lijnen door het gescande voorwerp bewegen, maar worden verstrooid. Vooral bij lage intensiteit van de stralingsbron of bij korte belichtingstijd bij de opname van de projectiebeelden is het effect van ruis significant. In hoofdstuk 4 introduceren we een reconstructie algoritme dat minder last heeft van ruis en ook gebruikt kan worden in het uitdagende geval wanneer er slechts weinig projectiebeelden beschikbaar zijn.

De intensiteit van de stralingsbron is in het algemeen niet constant, of kan niet nauwkeurig worden gemeten. Het gevolg kan zijn dat het nulniveau van de projectiebeelden verloren gaat. Dit leidt eveneens tot artefacten in de reconstructie. In hoofdstuk 5 bespreken we een algoritme dat het nulniveau kan benaderen zodat artefacten kunnen worden gecorrigeerd.

Al deze methodes richten zich op zeer specifieke verstoringen. Soms zijn er verstoringen die minder specifiek zijn of het wiskundige model in het reconstructie algoritme is niet volledig. Dit soort fouten zijn vaak moeilijk te lokaliseren en het is niet direct duidelijk hoe je hiervoor moet corrigeren. In hoofdstuk 6 introduceren

we een reconstructie methode die in het algemeen iets langzamer is, maar veel robuuster in het geval van dit soort niet-specifieke verstoringen.

Met de bevindingen van dit proefschrift is er een stap gezet om reconstructie algoritmes voor tomografie breder toepasbaar te maken, ook voor experimentele datasets waarbij de projectiebeelden onnauwkeurigheden bevatten.

# Curriculum Vitae

Folkert Bleichrodt was born in Gouda in 1987. He attended the Christelijke Scholengemeenschap De Goudse Waarden (Gouda) from 1999, where he completed the Atheneum program. From 2005 he studied Applied Mathematics with a Master in Scientific Computing at the mathematics department of the University of Utrecht, the Netherlands. He obtained his bachelor's degree in 2008 and his master's degree in 2011 (with honors). For the master's degree he wrote a thesis entitled "Accelerating finite differences for solving a barotropic ocean model on the GPU" under the supervision of Prof. dr. R.H. Bisseling (Utrecht University) and Prof. dr. ir. H.A. Dijkstra (IMAU - Institute for Marine and Atmospheric Research, Utrecht). From 2011 he worked as a PhD student at Centrum Wiskunde & Informatica (CWI) in Amsterdam, under supervision of Prof. dr. K.J. Batenburg. He presented his work on international conferences in Helsinki, Ghent, Hong Kong and Newport (USA). From 2015 he is employed at the CWI as a postdoctoral researcher where he works on a joint project with ExxonMobil.

# Acknowledgement

I would like to express my gratitude foremost to my supervisor and promotor Joost Batenburg. I have learned a lot from you in my four years pursuing my PhD degree. Also thank you for allowing me to work on these very fun side projects such as the radio tomography project.

I would like to thank my colleagues for the great athmosphere at the CWI: Jeroen Bédorf, Debarati Bhaumik, Rob Bisseling, Laurent van den Bos, Daan Crommelin, Jesse Dorrestijn, Svetlana Dubinkina, Anne Eggels, Bram van Es, Qian Feng, Wagner Fortes, Jason Frank, Lech Grzelak, Zaza van der Have, Willem Haverkort, Shashi Jain, Barry Koren, Prashant Kumar, Bart de Leeuw, Tristan van Leeuwen, Alvaro Leitao, Keith Myerscough, Zsolt Nika, Margreet Nool, Kees Oosterlee, Willem Jan Palenstijn, Daan Pelt, Linda Plantagie, Benjamin Sanderse, Sangeetika Ruchi, Marjon de Ruijter, Anton van der Stoep, Frank Tabak, Nick Verheul, Wander Wadman, Jeroen Witteveen, Paul de Zeeuw, Zhichao Zhong, Xiaodong Zhuge. Especially I would like to thank my roommates Daan Pelt and Jeroen Bédorf, we had a great time at the conferences and the trips to Hong Kong and New York were great. Shashi Jain and Wagner Fortes, thanks for the enjoyable and memorable ping pong matches we had. I also would like to thank the PhD activity commity for arranging the many fun weekend trips and other activities.

Also special thanks goes to Willem Jan Palenstijn. Thanks for allowing me to pick your brain on many occasions and proofreading many of my papers.

I owe a lot to the supporting staff, especially Nada Mitrovic and Duda Tepsic. Duda, thank you for helping me with my many computer problems.

Tim van der Meij and Alyssa Milburn, I really enjoyed working with you on the radio tomography project. It was a challenging and fun project which even took us to the DIY-store where we had to improvise a stand from PVC pipes and a parasol base. The project was succesful as we presented it several times at the open day at CWI and it even got some air time on Belgium national television.

Finally, I am very grateful for my loving parents. Without their understanding and support this project would not be as successful. And of course my dear brother, Robert-Jan, thank you for your support and advise you gave me, also as a fellow scientist. Els, thanks for your generous hospitality. Last but not least, I want to thank my dear grandmother who is always supportive.

*Folkert, September 2015*

Propositions accompanying the thesis

# "Improving Robustness of Tomographic Reconstruction Methods"

by Folkert Bleichrodt

1. The limited precision of the floating-point representation rarely leads to visible artifacts in the reconstructed tomography image. However, this limited precision can impose a major obstacle for consistency optimization of alignment parameters.
(*Chapters 2 and 3*)

2. Consider the update step of DART, where the following problem is solved:

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}}\ \|\boldsymbol{W}\boldsymbol{x}-\boldsymbol{p}\|_2^2 \quad \text{subject to}\ \ x_i = v_i \ \text{ for } i \in F.$$

   See Chapter 4 for the notation.
   Let $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ be a diagonal matrix with nonnegative real entries

$$D_{ii} := \begin{cases} 0 & \text{if } i \in U \\ C & \text{if } i \in F, \end{cases}$$

   where $C$ is a constant and $U = \{1,\dots,N\} \setminus F$. For sufficiently large $C$, replacing the update step by

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}}\ \|\boldsymbol{W}\boldsymbol{x}-\boldsymbol{p}\|_2^2 + \|\boldsymbol{D}(\boldsymbol{x}-\boldsymbol{\nu})\|_2^2,$$

   leads to an SDART algorithm that yields reconstructions similar to DART.
   (*Chapter 4*)

3. Let $\boldsymbol{W} \in \mathbb{R}^{M\times N}$ be a discretized Radon transform operator for the parallel beam geometry (see Eq. (5.1) of this thesis). Let $\boldsymbol{e} = (1,\dots,1) \in \mathbb{R}^M$ and let S be a linear reconstruction algorithm. The problem

$$\underset{\tau}{\text{minimize}}\ \|\boldsymbol{W}(\mathrm{S}(\boldsymbol{p}-\tau\boldsymbol{e}))-(\boldsymbol{p}-\tau\boldsymbol{e})\|_2$$

   can be solved using two evaluations of the algorithm S.
   (*Chapter 5*)

4. Even if 50% of the pixels in each projection image is replaced by a large value, and the location of these "corrupted" data are unknown, meaningful reconstruction results with limited artifacts can still be obtained.
   (*Chapter 6*)

5. Radio tomographic imaging can potentially be used for anonymous tracking of customers, even those without a smartphone.

6. The tomographic reconstruction community would benefit greatly from a centralized, unified database of tomographic datasets and corresponding high quality reconstructions. For experimentalists this is a tool to validate precision and accuracy of new tomographic scanner setups. Algorithm developers can test and compare the accuracy of their methods.

7. In a PhD project that is aimed to develop and implement numerical algorithms, the development time is equally important as the final computation time of the algorithm. Having a working algorithm is the first milestone, while efficiency can be considered as a next step.

8. Reconstruction is only part of an entire pipeline consisting of: acquisition, preprocessing, reconstruction, post-processing and quantitative analysis. Therefore, when optimizing this pipeline, each step should not optimize its output, but should optimize the input of the next step in the pipeline.

9. When interfacing between Matlab and C-type languages, the mixing of column-major order and row-major order of arrays forms the recipe for a major headache.