

***SOURCE FIRING PATTERNS AND  
RECONSTRUCTION ALGORITHMS FOR A  
SWITCHED SOURCE,  
OFFSET DETECTOR CT MACHINE***

Thompson, William Michael

2010

MIMS EPrint: **2016.49**

Manchester Institute for Mathematical Sciences  
School of Mathematics

The University of Manchester

Reports available from: <http://eprints.maths.manchester.ac.uk/>

And by contacting: The MIMS Secretary  
School of Mathematics  
The University of Manchester  
Manchester, M13 9PL, UK

ISSN 1749-9097

SOURCE FIRING PATTERNS AND  
RECONSTRUCTION ALGORITHMS  
FOR A SWITCHED SOURCE, OFFSET  
DETECTOR CT MACHINE

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER  
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY  
IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

2010

**William Michael Thompson**  
School of Mathematics

# Contents

<b>Abstract</b>	<b>10</b>
<b>Declaration</b>	<b>11</b>
<b>Copyright Statement</b>	<b>12</b>
<b>Acknowledgements</b>	<b>13</b>
<b>1 Introduction</b>	<b>14</b>
1.1 A Brief History of CT . . . . .	14
1.2 The RTT System . . . . .	15
1.2.1 The Basic Design Concept . . . . .	15
1.2.2 Drawbacks Of The RTT System . . . . .	17
1.2.3 The RTT Geometry . . . . .	18
1.2.4 Two Modes of Operation . . . . .	18
1.3 Organisation of the Thesis . . . . .	19
<b>2 The Continuum Model</b>	<b>21</b>
2.1 The Mathematical Model . . . . .	21
2.2 The Fundamental Transforms . . . . .	22
2.3 Properties of the Transforms . . . . .	24
2.3.1 The Fourier Slice Theorem . . . . .	24
2.3.2 The Formal Adjoint Operator $R^*$ . . . . .	25
2.3.3 Inversion Formulae . . . . .	26
2.3.4 Stability Estimates . . . . .	27

2.4	Limited Angle Tomography . . . . .	27
2.5	The RTT Problem . . . . .	27
2.5.1	Sources on a Curve in $\mathbb{R}^3$ . . . . .	28
2.5.2	Sources on a Cylinder . . . . .	29
2.5.3	Relations Between the 3D X-Ray and Radon Transforms . . . . .	30
2.5.4	The RTT Geometry – an Idealised Model . . . . .	31
2.5.5	Uniqueness of Solution . . . . .	36
2.5.6	Inversion and Stability Estimates . . . . .	38
<b>3</b>	<b>2D RTT Reconstruction</b>	<b>39</b>
3.1	The RTT20 Machine . . . . .	39
3.2	Analytical Reconstruction Methods . . . . .	41
3.2.1	The Initial Approach . . . . .	42
3.2.2	Interpolation to Parallel Beams . . . . .	43
3.2.3	Triangulation Based Interpolation . . . . .	43
3.2.4	Results with Phantom Data . . . . .	45
<b>4</b>	<b>Algebraic Reconstruction Methods</b>	<b>49</b>
4.1	Modelling the Projection Process . . . . .	50
4.2	Iterative Solution Methods . . . . .	51
4.2.1	Iterative Algorithms Conventionally Used in CT . . . . .	51
4.2.2	Conjugate Gradient Least Squares . . . . .	54
4.3	Applying Algebraic Reconstruction to RTT20 . . . . .	56
4.3.1	Calculating $A$ . . . . .	56
4.3.2	Solving Using CGLS . . . . .	56
4.3.3	Regularisation . . . . .	57
4.4	Results With Phantom Data . . . . .	57
<b>5</b>	<b>3D RTT Reconstruction</b>	<b>62</b>
5.1	The RTT80 System . . . . .	62
5.2	The Firing Order . . . . .	63

5.2.1	A Formal Definition . . . . .	63
5.2.2	Rotational Invariance . . . . .	64
5.2.3	The Inverse Firing Order . . . . .	65
5.3	Reconstruction Algorithms . . . . .	65
5.3.1	The RayConStruct Algorithm . . . . .	65
5.3.2	Rebinning Based Methods . . . . .	67
5.3.3	Algebraic Reconstruction . . . . .	68
5.4	Applying Algebraic Reconstruction to RTT80 . . . . .	69
5.4.1	The Structure of $A$ . . . . .	70
5.4.2	Reducing the Block Size . . . . .	71
5.4.3	Choosing an Appropriate Resolution . . . . .	74
5.4.4	Calculating the Coefficients . . . . .	75
5.4.5	Implementing the Reconstruction . . . . .	75
5.4.6	Regularisation . . . . .	77
<b>6</b>	<b>Optimising the Firing Order</b>	<b>78</b>
6.1	Types of Firing Order . . . . .	79
6.1.1	Rotationally Invariant Firing Orders . . . . .	79
6.1.2	Non-Rotationally Invariant Firing Orders . . . . .	81
6.2	Heuristic Arguments for the Choice of Firing Order . . . . .	82
6.2.1	An Even Sampling Lattice . . . . .	82
6.2.2	Even Angular Distribution . . . . .	83
6.2.3	Firing Orders to Compare . . . . .	84
6.3	Visualising the Firing Order . . . . .	85
6.3.1	The Sampling Pattern . . . . .	85
6.3.2	The Ray Density . . . . .	85
6.3.3	The Angular Distribution . . . . .	88
6.3.4	Expected Results . . . . .	91
6.4	Numerical Comparisons . . . . .	91
6.4.1	Condition Number Estimation . . . . .	91

6.4.2	Condition Number Results . . . . .	95
6.4.3	Calculating the SVD . . . . .	98
6.4.4	SVD Results . . . . .	99
6.4.5	Projections Onto The Singular Vectors . . . . .	106
<b>7</b>	<b>Reconstruction Results</b>	<b>110</b>
7.1	Results With Phantom data . . . . .	110
7.1.1	Calculating Phantom Data . . . . .	110
7.1.2	Results . . . . .	111
7.2	Working With Real Data . . . . .	112
7.2.1	The RTT Data Format . . . . .	112
7.2.2	Data Preprocessing . . . . .	118
7.2.3	Correcting for Defective Sources and Detectors . . . . .	122
7.2.4	Approximate Reconstruction with the RTT Firing Order . . . . .	122
7.2.5	Restrictions on the Firing Order . . . . .	123
7.3	Results with Real Data . . . . .	124
<b>8</b>	<b>Conclusions and Future Work</b>	<b>126</b>
8.1	Conclusions . . . . .	126
8.2	Future Work . . . . .	127
8.2.1	Improved Modelling of the Projection Process . . . . .	127
8.2.2	Reconstruction from Real Data . . . . .	128
8.2.3	Analytical Inversion of the New Model . . . . .	128
8.2.4	Two-Dimensional Imaging of Dynamic Processes . . . . .	128
	<b>Bibliography</b>	<b>130</b>
<b>A</b>	<b>Elementary Distribution Theory</b>	<b>138</b>

# List of Tables

2.1	Values of the constants for RTT80 . . . . .	36
5.1	Values of $a$ , $b$ and $V_x V_y$ for different resolutions . . . . .	75
6.1	Mean and standard deviation of ray density for the different firing orders	88
6.2	Values of constants in equation (6.10) for different firing orders – 1 slice	95
6.3	Values of constants in equation (6.10) for different firing orders – 12 slices . . . . .	98

# List of Figures

2.1	Example of a sinogram . . . . .	23
2.2	Example of limited angle reconstruction . . . . .	28
2.3	Source positions on a regular lattice . . . . .	29
2.4	The idealised RTT geometry . . . . .	32
2.5	The plane $\Pi_{\alpha,x}$ and the measured region . . . . .	34
2.6	Cross-section showing upper limit for $\alpha_1$ . . . . .	34
2.7	Transaxial cross-section showing positions of $x'$ and $w'$ . . . . .	35
2.8	The open set $\Omega$ of points on $S^2$ for which the Radon transform is known	36
3.1	The RTT20 geometry . . . . .	40
3.2	Maximum $z$ -axis deviation per pixel . . . . .	41
3.3	Angles in degrees between adjacent detectors in the active region for source 1 . . . . .	42
3.4	Sampling points in parallel-beam sinogram space of the RTT20 data .	44
3.5	A small section of the Delaunay triangulation of the RTT20 data . .	45
3.6	RTT sinogram of the RTT20 test phantom . . . . .	46
3.7	Parallel-beam sinograms obtained using Delaunay triangulation based interpolation . . . . .	47
3.8	Reconstructions of the parallel-beam sinograms using <code>iradon</code> . . . . .	48
4.1	Discretisation of the projection process . . . . .	51
4.2	Reconstruction of resolution phantom – no regularisation . . . . .	58
4.3	Reconstruction of resolution phantom – regularised with Laplacian, $\alpha = 1$ . . . . .	59

4.4	Reconstruction of resolution phantom – regularised with Laplacian, $\alpha = 5$ . . . . .	60
4.5	2-norm of the residuals for the reconstructions in figures 4.2–4.4 . . .	61
4.6	2-norm of the solutions for the reconstructions in figures 4.2–4.4 . . .	61
5.1	The RTT80 geometry . . . . .	63
5.2	‘Windmill’ artefact shown over 8 consecutive slices . . . . .	66
5.3	‘Zigzag’ artefact . . . . .	67
5.4	Enlarged cross-section of RTT80 geometry showing lines of rotational symmetry . . . . .	72
6.1	Constructing an approximately equilateral triangle lattice . . . . .	83
6.2	Plots of the sampling pattern for the different firing orders . . . . .	86
6.3	Ray density images for the different firing orders . . . . .	87
6.4	Images of standard deviation of the angular distribution for the differ- ent firing orders . . . . .	89
6.5	Distributions of angles for individual voxels for the different firing orders	90
6.6	Visualisation of the angular distribution for a small region of the re- construction slice using radial histograms for each voxel . . . . .	92
6.7	Condition number by discretisation size – 1 slice . . . . .	96
6.8	Condition number by discretisation size – 12 slices . . . . .	97
6.9	Condition number by volume slices – $20 \times 20 \times 1.39\text{mm}$ voxels . . . . .	98
6.10	Singular values of $A$ representing 1 slice . . . . .	100
6.11	First 6 right singular vectors of $A$ for sequential firing order . . . . .	101
6.12	First 6 right singular vectors of $A$ for RTT firing order . . . . .	102
6.13	First 6 right singular vectors of $A$ for random firing order . . . . .	103
6.14	First 6 right singular vectors of $A$ for $k = 35$ firing order . . . . .	104
6.15	Singular values of $A$ representing 12 slices . . . . .	105
6.16	‘Phantom’ for singular vector projections . . . . .	106
6.17	Projections onto singular vectors for the different firing orders . . . . .	107
6.18	Approximation by first $p$ singular vectors – sequential . . . . .	108

6.19	Approximation by first $p$ singular vectors – RTT . . . . .	108
6.20	Approximation by first $p$ singular vectors – random . . . . .	109
6.21	Approximation by first $p$ singular vectors – $k = 35$ . . . . .	109
7.1	Reconstruction of the multi-cuboid phantom for the different firing orders – 20 iterations CGLS with no additional regularisation . . . . .	113
7.2	Reconstruction of the multi-cuboid phantom for the different firing orders – 50 iterations CGLS with no additional regularisation . . . . .	114
7.3	Reconstruction of the $z$ resolution phantom for the different firing orders – 20 iterations CGLS with no additional regularisation . . . . .	115
7.4	Reconstruction of the $z$ resolution phantom for the different firing orders – 50 iterations CGLS with no additional regularisation . . . . .	116
7.5	2-norm of the residual at each iteration of the CGLS algorithm for the multi-cuboid phantom . . . . .	117
7.6	2-norm of the residual at each iteration of the CGLS algorithm for the resolution test phantom . . . . .	117
7.7	Example RTT80 calibration light image . . . . .	119
7.8	Example RTT80 calibration dark image . . . . .	120
7.9	Example uncalibrated and calibrated RTT80 sinograms . . . . .	121
7.10	Reconstruction of a test bag from real data scanned using the RTT firing order and reconstructed using the approximation of section 7.2.4	125

# The University of Manchester

William Michael Thompson

Doctor of Philosophy

Source Firing Patterns and Reconstruction Algorithms for a Switched Source, Offset Detector CT Machine

December 31, 2010

## Abstract

We present a new theoretical model and reconstruction results for a new class of fast x-ray CT machine, the Real Time Tomography (RTT) system, which uses switched sources and an offset detector array. We begin by reviewing elementary properties of the Radon and x-ray transforms, and limited angle tomography. Through the introduction of a new continuum model, that of sources covering the surface of a cylinder in  $\mathbb{R}^3$ , we show that the problem of three-dimensional reconstruction from RTT data reduces to inversion of the three-dimensional Radon transform with limited angle data. Using the Paley-Wiener theorem, we then prove the existence of a unique solution and give comments on stability and singularity detection.

We show, first in the two-dimensional case, that the conjugate gradient least squares algorithm is suitable for CT reconstruction. By exploiting symmetries in the system, we then derive a method of applying CGLS to the three-dimensional inversion problem using stored matrix coefficients.

The new concept of source firing order is introduced and formalised, and some novel visualisations are used to show how this affects aspects of the geometry of the system. We then perform a detailed numerical analysis using the condition number and SVD of the reconstruction matrix  $A$ , to show that the choice of firing order affects the conditioning of the problem. Finally, we give reconstruction results using phantom data that support the numerical analysis.

# Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

# Copyright Statement

- i.** The author of this thesis (including any appendices and/or schedules to this thesis) owns any copyright in it (the “Copyright”) and s/he has given The University of Manchester the right to use such Copyright for any administrative, promotional, educational and/or teaching purposes.
- ii.** Copies of this thesis, either in full or in extracts, may be made **only** in accordance with the regulations of the John Rylands University Library of Manchester. Details of these regulations may be obtained from the Librarian. This page must form part of any such copies made.
- iii.** The ownership of any patents, designs, trade marks and any and all other intellectual property rights except for the Copyright (the “Intellectual Property Rights”) and any reproductions of copyright works, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property Rights and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.
- iv.** Further information on the conditions under which disclosure, publication and exploitation of this thesis, the Copyright and any Intellectual Property Rights and/or Reproductions described in it may take place is available from the Head of the School of Mathematics.

# Acknowledgements

Firstly I would like to thank Bill, for the inspiration and for believing in me.

Secondly, my thanks go to my mother Pauline for the financial support over the past year, and to the rest of my family for being there.

Finally, my thanks go to Lennie, for her constant devotion and support, and for putting up with me over the last few months, I couldn't have done this without you!

# Chapter 1

## Introduction

X-ray computed tomography, or CT, is a technique for imaging slices through a body or object by measuring x-ray projections through the object from a range of angles; the word tomography comes from the Greek *τομος*, meaning slice. We begin with a short history of the evolution of CT.

### 1.1 A Brief History of CT

The origins of CT can be traced back to 1917, with the publication of the Swiss mathematician Johann Radon’s paper “On the Determination of Functions From Their Line Integrals Along Certain Manifolds”<sup>1</sup> [55]. The next important development in CT came from Cormack in 1963 [8], who developed important theory for CT from a medical perspective, completely unaware of the work of Radon. The first practical implementation of a CT system was made by Hounsfield in 1972 [24], who later went on to share the Nobel prize for Medicine with Cormack in 1979.

The evolution of CT through the 1970’s is described by Kalender as the “4 generations of CT”, for example in [30] or [31]. This can be summarised as follows:

1<sup>st</sup> generation: Single source, single detector, parallel beam geometry; translate and rotate between measurements

---

<sup>1</sup>English translation from original German title “ber die Bestimmung von Funktionen durch ihre Integralwerte lngs gewisser Mannigfaltigkeiten”

2<sup>nd</sup> generation: Single source, multiple detectors, partial fan-beam geometry;  
translate and rotate between measurements

3<sup>rd</sup> generation: Single source, multiple detectors, fan-beam geometry; rotating  
source and detectors

4<sup>th</sup> generation: Single source, full ring of multiple detectors, fan-beam geometry;  
only source rotates

These four generations applied to two-dimensional scanners, imaging a single slice through the object. Evolution through the 1980's and 1990's saw the development of three-dimensional scanning, first in a slice-by-slice manner, where the object or patient being scanned would be physically moved between slices.

Developments since then have essentially concentrated solely on the 3<sup>rd</sup> generation of CT scanner. In the late 1980's to early 1990's, with the development of multi-row, area detectors, fast three-dimensional scanning became a reality, with the source describing a helical path around the object being scanned. Through the 1990's and into the 21<sup>st</sup> century, detector sizes have become larger and larger, enabling faster and faster scanning times and increased spatial resolution.

Scanning times with these 3<sup>rd</sup> generation conventional scanners are limited, however, by the physical constraints of rotating a source and detectors around the object. The very fastest conventional scanners perform around 3 full source revolutions per second; for example, modern dual-source scanners described in [31]. Clearly, if we are to improve the scan time, a radical redesign of the scanner is needed; this is where RTT comes in.

## 1.2 The RTT System

### 1.2.1 The Basic Design Concept

The RTT system is a new type of x-ray CT machine developed by the security company Rapiscan systems. The motivation behind the design of the RTT system

comes from baggage scanning for airport security. The aim of the system is to produce full 3D scans of bags as they pass through on a conveyor belt at  $0.5\text{ms}^{-1}$ ; hence Real Time Tomography or RTT. As mentioned in section 1.1, the barrier to achieving this using conventional helical cone-beam CT scanners lies in the laws of physics and the limited speed with which one can physically rotate an x-ray source around on a gantry.

Rather than physically rotating a single x-ray source about the scanner, the RTT system instead uses a fixed ring of multiple sources and several fixed rings of detectors, positioned around the region of interest. The sources are switched on and off in sequence, creating the same effect as source movement. In this way, we can think of the RTT system in terms of Kalender's CT generations as a '5th generation' of CT scanner.

The principal advantage of this system over traditional CT scanners is that with no heavy machinery to move around, we are only limited by the speed with which the sources can be switched on and off. Currently this results in a maximum of 30 virtual source revolutions per second, which should be enough to produce 3D reconstructions at the required belt speed.

Of potential secondary benefit, and certainly of interest, is that with a switched source system we have the ability to switch the sources in almost any order we choose. In a conventional CT scanner the physical rotation of the source means that we are forced to move it in a circular path. This therefore traces out a helical source trajectory as the object or body moves through the scanner in the  $z$  direction. It would seem like the natural thing to do in the RTT system would be to switch the sources on and off consecutively in order to achieve the same effect; however, there is no particular reason why we should do this (in fact it is not actually possible in the real RTT system for practical reasons).

We refer to the source switching pattern as a *firing order*. It seems logical to expect that this will have an effect on the outcome of the reconstruction; this will be explored in detail in later chapters.

## 1.2.2 Drawbacks Of The RTT System

Of course, this system is not without its disadvantages; these are summarised in this section.

### Offset Detectors

Due to the construction of the RTT system using complete rings of sources and detectors, in order for rays from a particular source to not pass through the back of the detectors nearest that source, we must offset the detector region by a small amount. This actually leads to a totally different scanning geometry from conventional CT machines, where we always have rays in, and on both sides of, a ‘transaxial’ plane perpendicular to the axis of motion through the scanner. This leads to major challenges in developing a suitable reconstruction algorithm for the RTT system, which will be explored in this thesis.

### Lack of Collimation

Since the detectors must be capable of detecting rays from several sources, they must be capable of detecting rays from a wide range of angles; therefore there is very little in the way of detector collimation. The implication of this is higher levels of scatter being detected which, depending on the material being scanned, can lead to a high level of noise in the measurements. It is possible that the increased scatter could be exploited to our advantage; this is a whole research topic in itself and will not be covered here.

### Polygonal Rings

Of slightly less significance, but still a problem with smaller size RTT scanners, is the fact that, due to the physical constraints of manufacturing the scanner, the source and detector rings are not actually rings, but polygons. This creates an uneven angular sampling of the source and detector positions, which can be a problem for conventional analytical reconstruction algorithms.

### 1.2.3 The RTT Geometry

A general RTT system consists of a fixed ring of sources and several rings of detectors arranged on parallel planes. We define a Cartesian coordinate system such that each ring of sources and detectors is located in a plane parallel to the  $xy$ -plane and is centred on the  $z$ -axis. We think of the object being scanned as stationary while the scanner moves; the direction of motion being along the positive  $z$ -axis. In reality, due to the way the machine is constructed, the sources and detectors are actually arranged in straight blocks of 32 and 16 respectively, resulting in source and detector polygons. However, for simplicity's sake these will still be referred to as rings.

We define the number of sources as  $N_S$ , the number of detectors per ring as  $N_D$  and the number of rings as  $N_R$ . The set of all sources will be of particular importance and is denoted by

$$S = \{s_1, \dots, s_{N_S}\}. \quad (1.1)$$

Similarly, we denote the set of all detectors as

$$D = \{d_1, \dots, d_{N_D N_R}\}. \quad (1.2)$$

Detectors are numbered first by position on the detector ring, then by ring number. For a given source  $s_i$ , the active detector region  $D_i$  is the subset of  $D$  defined by the limits  $[\alpha, \alpha + \delta]$  as

$$D_i = \{d_j \in D : j = (m-1)N_R + n, m = 1, \dots, N_R, n = ((\alpha, \dots, \alpha + \delta) \bmod N_D) + 1\}, \quad (1.3)$$

where  $1 \leq \alpha \leq N_D$  and  $1 \leq \delta \leq N_D$ .

### 1.2.4 Two Modes of Operation

In addition to using the RTT system to produce 3D scans of entire objects, by keeping the object stationary within the scanner we can also use it to produce images of a two-dimensional cross-section of part of an object. Of course, with static objects this has been possible since the very early days of CT but the difference with RTT is that,

given the speed of the system, we now have the potential to image time-dependent processes with a high degree of temporal resolution.

This opens up some very exciting possibilities; for example, we could use the RTT system to produce images of the inside of mixing fluids or granular flows. In particular, it is anticipated that we will be able to use RTT to validate theoretical results in granular flow. This type of imaging also has the potential to be used industrially, for example in imaging flow through oil pipes or mixing of liquids in chemical plants.

This gives us two problems to investigate; the main problem of 3D reconstruction from the RTT system, and the considerably less complicated sub-problem of two-dimensional time dependent reconstruction.

### 1.3 Organisation of the Thesis

We give here a brief outline of the organisation of the thesis.

In chapter 2 we introduce the fundamental mathematical concepts of CT reconstruction, and develop theory for the RTT system in the continuum case.

Chapter 3 looks at two-dimensional reconstruction from the RTT20 system by analytical methods, while chapter 4 introduces algebraic reconstruction in the two-dimensional case and applies this to RTT20 reconstruction.

In chapter 5 we consider the practical problem of three-dimensional reconstruction from the RTT80 system. We introduce the important concept of the firing order and develop a method of applying algebraic reconstruction to RTT80 data using stored matrix coefficients.

In chapter 6 we further develop the concept of the firing order and consider the problem of its optimisation. We introduce ways of visualising its effect and perform numerical comparisons between some different firing orders, with the conclusion that the standard helical trajectory actually represents the worst case for the RTT system.

In chapter 7 we give results of applying the reconstruction algorithm of chapter 5 to phantom RTT80 data, for three different firing orders. The results support the

numerical analysis of chapter 6.

Finally, chapter 8 summarises our conclusions and offers some suggestions for further work.

# Chapter 2

## The Continuum Model

We begin our investigation by reviewing the mathematical model of computed tomography and the elementary properties of the fundamental transforms; much of the introductory work in this chapter is based on the notation of Frank Natterer's books [47] and [48]. We then derive a new mathematical model for the RTT system and go on to prove the existence of a unique solution.

### 2.1 The Mathematical Model

The fundamental problem of x-ray computed tomography is to reconstruct the unknown density of an object, given only knowledge of the intensity lost by a set of x-ray beams passed through it. We let  $f(x)$  represent the density of the object at the point  $x \in \mathbb{R}^3$  and represent the x-ray beams as straight lines. Since the function  $f$  represents density in some real physical object, we generally assume that  $f$  has compact support. We assume that the intensity lost by an x-ray beam passing through a small distance  $\delta x$  in the neighbourhood of  $x$  depends linearly on the density at that point, and neglect all other physical effects such as scattering and beam hardening.

Consider an x-ray beam as the line  $L$  given by  $x = a + s\theta$ , where  $a \in \mathbb{R}^3$  and  $\theta \in S^2$ , the unit sphere, and let  $I(x)$  represent the intensity of the beam at the point  $x \in L$ . Then we have

$$\frac{dI(x)}{ds} = -f(x)I(x). \quad (2.1)$$

Let the measured intensity of  $L$  be represented by  $I$ , and let the initial intensity be  $I_0$ . Then integrating (2.1) along the line  $L$  gives

$$\frac{I}{I_0} = \exp\left(-\int_L f(x)dx\right), \quad (2.2)$$

which on rearranging becomes

$$\int_L f(x)dx = -\log\left(\frac{I}{I_0}\right). \quad (2.3)$$

Therefore, the intensity data give us the line integrals of  $f$ , and the reconstruction problem becomes that of recovering a function  $f$  from its line integrals. This was first solved in an abstract setting by Radon in 1917 in his paper [55]; the integral transform introduced in the following section is named in his honour.

## 2.2 The Fundamental Transforms

We now introduce the integral transforms which form the mathematical foundations of computed tomography. In the following section, we assume that the function  $f$  belongs to the Schwartz space  $\mathcal{S}(\mathbb{R}^n)$ , defined in appendix A.

For  $f \in \mathcal{S}(\mathbb{R}^n)$ , we denote by  $R$  the  $n$ -dimensional Radon transform. For  $\theta \in S^{n-1}$  and  $s \in \mathbb{R}$ , where  $S^n = \{x \in \mathbb{R}^n : |x| = 1\}$  is the unit  $n$ -sphere, we have

$$Rf(\theta, s) = \int_{x \cdot \theta = s} f(x)dx = \int_{\theta^\perp} f(s\theta + y)dy, \quad (2.4)$$

where  $\theta^\perp$  denotes the orthogonal complement of  $\theta$ , giving the integral of  $f$  over the hyperplane perpendicular to  $\theta$  and distance  $s$  from the origin. Therefore  $R$  maps a function  $f$  to the set of its integrals over hyperplanes in  $\mathbb{R}^n$ . Note that  $Rf(\theta, s) = Rf(-\theta, -s)$ , so  $Rf$  defines an even function on the unit cylinder  $Z = S^{n-1} \times \mathbb{R}$ . It is clear from the definition that  $R$  defines a linear operator on  $\mathcal{S}(\mathbb{R}^n)$ .

In two dimensions, we can view the Radon transform of a compactly supported function  $f$  as an image, since in this case  $S^1$  is simply the one-dimensional unit circle. By the evenness property of  $Rf$ , we need only consider  $\theta$  in the range  $[0, \pi)$ . Such an image is known as a *sinogram*; since the Radon transform of a delta distribution is

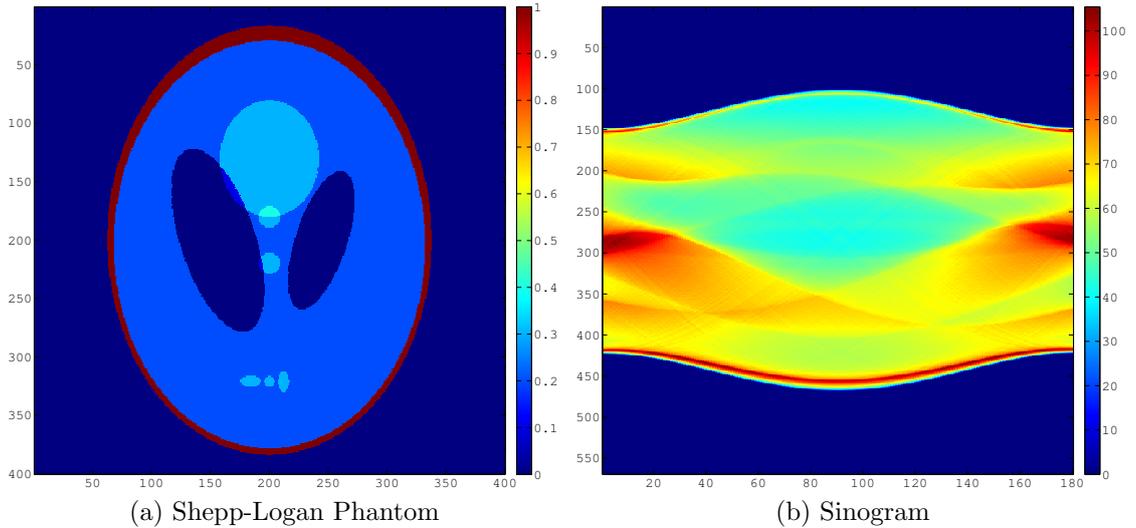


Figure 2.1: Example of a sinogram

a sine curve, if we think of an image  $f$  as a collection of delta distributions, then by the linearity of  $R$ , the resulting  $Rf$  appears as a set of superimposed sine curves. An example of this is shown in figure 2.1 for a two-dimensional Shepp-Logan phantom, designed to represent a slice through the human head.

We will sometimes use the alternative notation

$$R_{\theta}f(s) = Rf(\theta, s). \quad (2.5)$$

In the case of x-ray CT, we have integrals over *lines* in  $\mathbb{R}^n$ . We therefore introduce the  $n$ -dimensional x-ray transform, denoted by  $X$ , mapping the function  $f \in \mathcal{S}(\mathbb{R}^n)$  to the set of its line integrals in  $\mathbb{R}^n$ . For  $\theta \in S^{n-1}$  and  $x \in \mathbb{R}^n$ , we have

$$Xf(\theta, x) = \int_{-\infty}^{\infty} f(x + s\theta) ds, \quad (2.6)$$

giving the integral of  $f$  along the straight line through  $x$  in direction  $\theta$ . If  $x$  is moved in the direction  $\theta$ ,  $Xf(\theta, x)$  does not change. Hence,  $Xf$  defines a function on the tangent bundle  $T = S^{n-1} \times \theta^{\perp}$ .

Again, we sometimes use the alternative notation

$$X_{\theta}f(x) = Xf(\theta, x). \quad (2.7)$$

For some  $\theta \in \mathbb{R}^n$ ,  $X_{\theta}$  is referred to as the *projection* of  $f$  onto  $\theta^{\perp}$ . In the case  $n = 2$  the x-ray transform can be considered equivalent to the Radon transform.

We can think of the x-ray transform as taking projections through the support of  $f$  along sets of parallel rays. However, in most practical CT problems, barring synchrotron CT, we do not measure such sets of rays. We therefore introduce the  $n$ -dimensional divergent beam transform, denoted by  $D$ . For  $a \in \mathbb{R}$  and  $\theta \in S^{n-1}$ , we have

$$Df(a, \theta) = \int_0^\infty f(a + s\theta) ds, \quad (2.8)$$

giving the integral along the half-line with end point  $a$  in direction  $\theta$ . We will sometimes use the alternative notation

$$D_a f(\theta) = Df(a, \theta). \quad (2.9)$$

We can identify the divergent beam transform with the x-ray transform by

$$Xf(\theta, x) = Df(x, \theta) + Df(-x, \theta). \quad (2.10)$$

## 2.3 Properties of the Transforms

We now describe some properties of the fundamental transforms, and introduce some important theorems.

### 2.3.1 The Fourier Slice Theorem

For  $f \in \mathcal{S}(\mathbb{R}^n)$ , we denote by  $\hat{f}$  and  $\check{f}$  respectively the Fourier and inverse Fourier transforms of  $f$ , defined as follows

$$\hat{f}(\omega) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} e^{-ix \cdot \omega} f(x) dx, \quad (2.11)$$

$$\check{f}(x) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} e^{ix \cdot \omega} f(\omega) d\omega. \quad (2.12)$$

We note that for  $f \in \mathcal{S}(\mathbb{R}^n)$ , the Fourier and inverse Fourier transforms  $\hat{f}$  and  $\check{f}$  are also in  $\mathcal{S}(\mathbb{R}^n)$ .

We have the following theorem relating the Radon and x-ray transforms with the Fourier transform, sometimes referred to as the projection-slice theorem.

**Theorem 2.1** (Fourier Slice Theorem). *Let  $f \in \mathcal{S}(\mathbb{R}^n)$ ; then for  $\theta \in S^{n-1}$  we have*

$$\widehat{R_\theta f}(s) = (2\pi)^{(n-1)/2} \hat{f}(s\theta), \quad \text{where } s \in \mathbb{R}, \quad (2.13)$$

$$\widehat{X_\theta f}(x) = (2\pi)^{1/2} \hat{f}(x), \quad \text{where } x \in \theta^\perp. \quad (2.14)$$

*Proof.* See, for example, [47], p11. □

Since the Fourier transform has a well-defined inverse, this proves that the Radon and x-ray transforms are also invertible, and provides us with inversion formulae. However, although in principle we could use these formulae for direct inversion of the transforms, this is rarely done in practice since it involves calculating the inverse Fourier transform in 2 or 3 dimensions.

### 2.3.2 The Formal Adjoint Operator $R^*$

Using the notation of [58], the Schwartz spaces  $\mathcal{S}(\mathbb{R}^n)$  and  $\mathcal{S}(Z)$  are equipped with inner products

$$\langle f_1, f_2 \rangle = \langle f_1, f_2 \rangle_{\mathcal{S}(\mathbb{R}^n)} = \int_{\mathbb{R}^n} f_1(x) \overline{f_2(x)} dx, \quad (2.15)$$

$$(g_1, g_2) = (g_1, g_2)_{\mathcal{S}(Z)} = \int_{S^{n-1}} \int_{-\infty}^{\infty} g_1(\theta, s) \overline{g_2(\theta, s)} ds d\theta, \quad (2.16)$$

where the bar denotes the complex conjugate. Then the *formal adjoint operator*  $R^*$  is defined by

$$(Rf, g) = \langle f, R^*g \rangle. \quad (2.17)$$

Since we are only considering real valued functions, we may drop the complex conjugation and after some manipulation obtain

$$R^*g(x) = \int_{S^{n-1}} g(\theta, \theta \cdot x) d\theta. \quad (2.18)$$

Geometrically, the formal adjoint operator  $R^*$  is the dual of  $R$  in the sense that while  $R$  integrates over all points on a hyperplane,  $R^*$  integrates over all hyperplanes through a point. Its action can be viewed as projecting back the data  $Rf$  onto its image in  $\mathbb{R}^n$ ; for this reason  $R^*$  is often referred to as the *backprojection* operator.

### 2.3.3 Inversion Formulae

We are now in a position to introduce a practical method of inversion for the Radon transform. We first introduce the Riesz potential, defined as follows:

**Definition 2.1.** For a function  $f$  defined on  $\mathbb{R}^n$ , and  $\alpha < n$ , the *Riesz potential* is the linear operator  $I^\alpha$  given by

$$\widehat{I^\alpha f}(\xi) = |\xi|^{-\alpha} \hat{f}(\xi). \quad (2.19)$$

For  $f$  defined on  $Z$  or  $T$ , the Riesz potential is defined as acting on the second variable. The Radon inversion formula is given by the following theorem:

**Theorem 2.2.** *Let  $f \in \mathcal{S}(\mathbb{R}^n)$ . Then for any  $\alpha < n$ , we have*

$$f = \frac{1}{2}(2\pi)^{1-n} I^{-\alpha} R^* I^{\alpha-n+1} g, \quad \text{where } g = Rf. \quad (2.20)$$

*Proof.* For a proof of this see, for example, [47], p19. □

Putting  $\alpha = 0$  in (2.20) gives

$$f = \frac{1}{2}(2\pi)^{1-n} R^* I^{1-n} g, \quad (2.21)$$

which is commonly known as *filtered backprojection*, since the data  $g$  are first filtered by the one-dimensional filter  $I^{1-n}$ , and then backprojected by  $R^*$ . Since the filter involved is only one-dimensional, this inversion formula can be implemented numerically very efficiently.

Similarly to the Radon transform inversion, we can derive explicit inversion formulae for the X-ray transform by

$$f = \frac{1}{|S^{n-2}|} (2\pi)^{-1} I^{-\alpha} X^* I^{\alpha-1} g, \quad \text{where } g = Xf, \quad (2.22)$$

where  $X^*$  is the formal adjoint of the X-ray transform. Again, for the derivation and proof of this, see [47], p19.

### 2.3.4 Stability Estimates

The Schwartz space  $\mathcal{S}(\mathbb{R}^n)$  is the natural space to start any investigation of the Radon and x-ray transforms, since it is closed under the Fourier transform. However, the Schwartz space is not the natural space in which to consider the practical inversion problem, since in general we cannot assume the function  $f$  is smooth. Natterer gives more general results in [47] by first proving continuity in  $L^2$  spaces (p17), and then stability estimates in Sobolev spaces in section II.5.

## 2.4 Limited Angle Tomography

For  $f \in \mathcal{S}(\mathbb{R}^n)$ , the inversion formula of (2.21) requires knowledge of the Radon transform for *all* values of  $\theta \in S^{n-1}$ . Limited angle tomography refers to the case where we only have Radon transform data for  $\theta$  in some subset  $\Omega \subset S^{n-1}$ . Figure 2.2 shows the effect in two dimensions of reconstructing a limited angle version of the Radon transform data for the Shepp-Logan phantom of figure 2.1, setting the unknown data to zero.

The artefacts present in the image are typical for reconstruction of limited angle data. We see that singularities (i.e. edges) in directions tangent to angles whose rays are not measured are no longer visible. We also get strong streaking artefacts in the direction of the missing angles. Intuitively, this is what we should expect, since the only way of detecting such a singularity is to detect the difference in tangent ray integrals in some neighbourhood of the singularity. This is treated in the framework of microlocal analysis by Quinto for the x-ray transform in [53], and for the Radon transform in [54].

## 2.5 The RTT Problem

We now consider the problem of reconstruction in 3D with the RTT geometry. The basic problem of 3D reconstruction is essentially the inversion of the 3D x-ray transform. However, inversion using explicit formulae of the type 2.22 requires knowledge

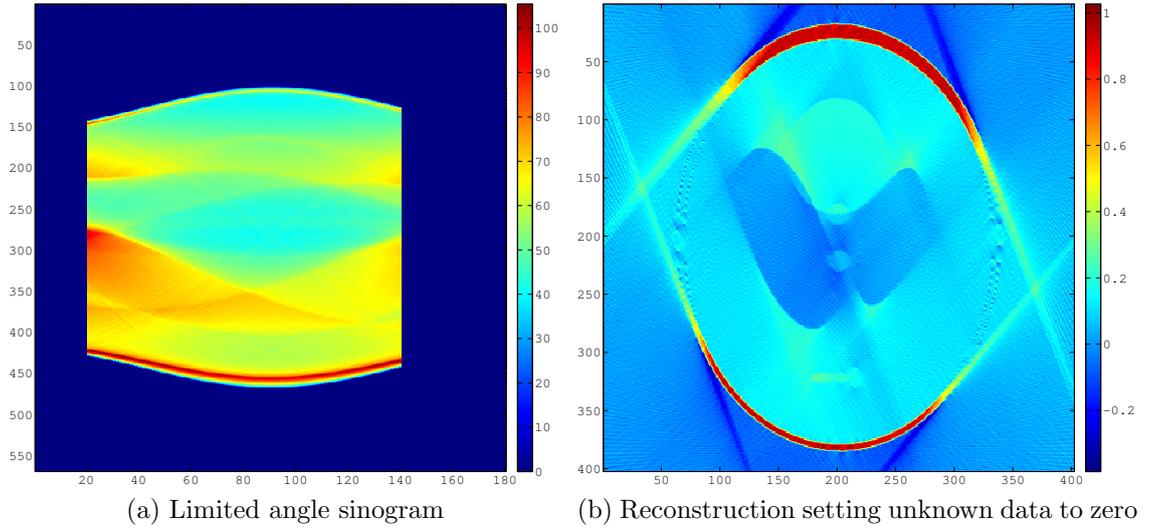


Figure 2.2: Example of limited angle reconstruction

of the integrals of  $f$  over *all* lines in  $\mathbb{R}^3$ . In most practical situations, and in particular the RTT geometry, we clearly cannot measure all the line integrals and so in this sense the problem is one of limited data.

Considering all lines in  $\mathbb{R}^3$ , this forms a four-dimensional set, since almost all lines in  $\mathbb{R}^3$  may be parameterised by their points of intersection with two parallel planes. For the problem of reconstructing the unknown three-dimensional density function, it seems intuitive that certain three-dimensional subsets of the set of all lines in  $\mathbb{R}^3$  should be sufficient. Indeed, the three-dimensional set formed by all lines in the set of all planes parallel to some plane  $\Pi$  is certainly sufficient, since we have complete two-dimensional data in every such plane. We have to show that the subset of line integrals that we can measure in the RTT case is sufficient to determine the function  $f$  uniquely.

### 2.5.1 Sources on a Curve in $\mathbb{R}^3$

The standard continuum model for conventional 3D CT reconstruction is to consider divergent beam sources located at all points on some smooth curve  $L \subset \mathbb{R}^3$ . This gives us a three-dimensional set of lines, since for each point on the one-dimensional curve  $L$ , almost all lines through that point may be parameterised by their intersection with a plane not containing that point. We assume that  $f$  is of compact support

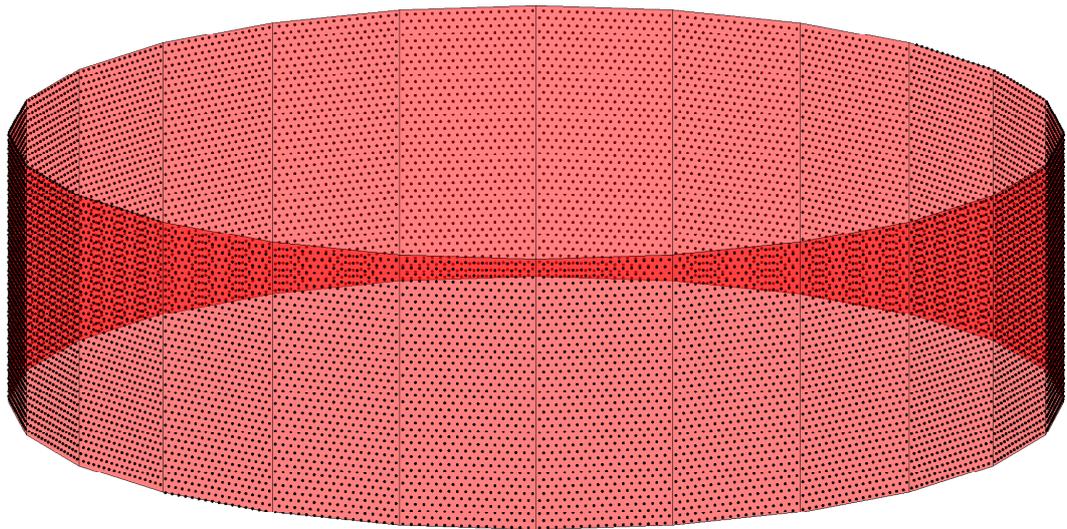


Figure 2.3: Source positions on a regular lattice

$\Omega \subset \mathbb{R}^3$ . An inversion formula was proved for this problem by Tuy in 1983 [66], where the curve  $L$  is assumed to be bounded and satisfies the condition that every plane intersecting  $\Omega$  also intersects the curve  $L$  in at least one point.

This is known as Tuy's condition and is clearly applicable to the problem of standard helical cone-beam CT reconstruction. However, for any  $a \in L$ , Tuy's inversion method assumes knowledge of  $D_a f(\theta)$  for *all*  $\theta \in S^2$ . Clearly this can never happen in practical situations, and in particular the RTT geometry with its restricted size, offset detectors. We therefore seek an alternative model.

### 2.5.2 Sources on a Cylinder

With the RTT system the sources may be fired in any order we choose. Motivated by our knowledge of helical scan cone-beam CT, it seems natural for us to think of the firing order as defining the trajectory  $L \subset \mathbb{R}^3$  that a virtual source or sources will follow. However, consider the firing order whose source positions are shown in figure 2.3. Here it is not clear whether there should be a single helical trajectory at a narrow pitch, or several helical trajectories at a much wider pitch.

We therefore propose the following alternative model; instead of regarding the firing order as defining one or more one-dimensional trajectories, we simply view it

as defining a sampling pattern on the surface of a cylinder.<sup>1</sup> For an arbitrary number of sources, the firing order may always be chosen so as to arrange the sources on a regular lattice as in the example above. Therefore, in the limit as the number of sources tends to infinity, the sources cover the whole surface of the cylinder, so in the continuous case we may assume that we have source positions at all points on the boundary of some cylinder. Hence the RTT problem becomes that of inversion of the divergent beam transform for sources on the surface of a cylinder.

It should be noted that the set of lines for this problem is actually a four-dimensional set, since, for example, a point on a vertical cylinder can be parameterised by its positions relative to its angular and  $z$ -axis coordinates. Then for any such point, almost all lines can be parameterised as in the case for sources on a curve by their points of intersection with a plane not containing that point. Therefore, it seems intuitive that line integrals given on this four-dimensional set should be enough to determine the function  $f$ .

### 2.5.3 Relations Between the 3D X-Ray and Radon Transforms

We wish to prove uniqueness of solution for the RTT system by relating the three-dimensional divergent-beam transform with sources on a cylinder with the three-dimensional Radon transform. Relations between the three-dimensional parallel-beam are simple to derive, since we just integrate the parallel rays across planes to get the corresponding plane integrals. The case of the divergent beam transform is not so easy, as due to the divergence, we cannot simply integrate over planes.

Grangeat proved the following result, relating the divergent-beam transform with the first derivative of the three-dimensional Radon transform:

**Theorem 2.3.** *Let  $f \in \mathcal{S}(\mathbb{R}^3)$ . Then for  $\theta \in S^2$ ,  $x \in \mathbb{R}^3$  and  $x \cdot \theta = s$ , we have*

$$\frac{\partial}{\partial s}(Rf)(\theta, s) = \int_{S^2 \cap \alpha^\perp} \frac{\partial}{\partial \theta}(Df)(x, \beta) d\beta, \quad (2.23)$$

---

<sup>1</sup>In the real RTT80 system, the sources will actually be on the surface of a 24 sided prism, as in figure 2.3. However, in our idealised model we assume that this is a cylinder.

where  $\frac{\partial}{\partial \theta}$  denotes the partial derivative in the direction  $\theta$ , acting on the second argument of  $Df$ .

The proof of this is given in [14] and [15]; however, this is really a corollary of a more general result given by Hamaker et al. in [17], which also yields other formulae such as those of Smith [63], and Gel'fand and Goncharov [11].

We will take a more direct approach to the relation between the divergent-beam and Radon transforms; by making careful assumptions about the geometry of the system, the geometry of sources at all points on the surface of a cylinder allows us to construct sets of parallel rays. We may then simply integrate over these to get the Radon transform; this is dealt with in the next section.

#### 2.5.4 The RTT Geometry – an Idealised Model

Let  $x \in \mathbb{R}^3$  be represented in cylindrical polar coordinates  $(r_x, \phi_x, z_x)$ , and let  $Z_a = \{x \in \mathbb{R}^3 : r_x \leq a\}$  be the cylinder of radius  $a$  centred on the  $z$ -axis. The boundary of such a cylinder is denoted by  $C_a = \{x \in \mathbb{R}^3 : r_x = a\}$ . We then define the geometry of the idealised RTT system as follows.

Let  $Z_{a,l}$  be some subset of the cylinder  $Z_a$ , of finite length  $l$  in the  $z$  direction and centred at the origin, and let the function  $f \in L^2(Z_{a,l})$  be supported on this. Let the sets of source positions and possible detector positions be defined as  $C_b$  and  $C_d$  respectively, where  $a < d < b$ . Let  $\epsilon_2 > \epsilon_1 > 0$  represent the source-detector offsets in the  $z$  direction, and let the angular extents of the detectors be  $-\gamma, \gamma$ . Then, relative to some source position  $x \in C_b$ , the active detector region is defined as the subset of  $C_d$  given by

$$D_x = \{y \in C_d : (\pi + \phi_x - \gamma \leq \phi_y \leq \pi + \phi_x + \gamma), (z_x + \epsilon_1 \leq z_y \leq z_x + \epsilon_2)\}, \quad (2.24)$$

as shown in figure 2.4.

The geometry is assumed to be shift-invariant, so that the active detector region is the same viewed from the perspective of any source position  $x \in C_b$ . Now, considering an arbitrary  $x \in C_b$ , let  $\alpha \in S^2$  such that the plane  $\Pi_{\alpha,x}$ , with equation  $y \cdot \alpha = s$  and

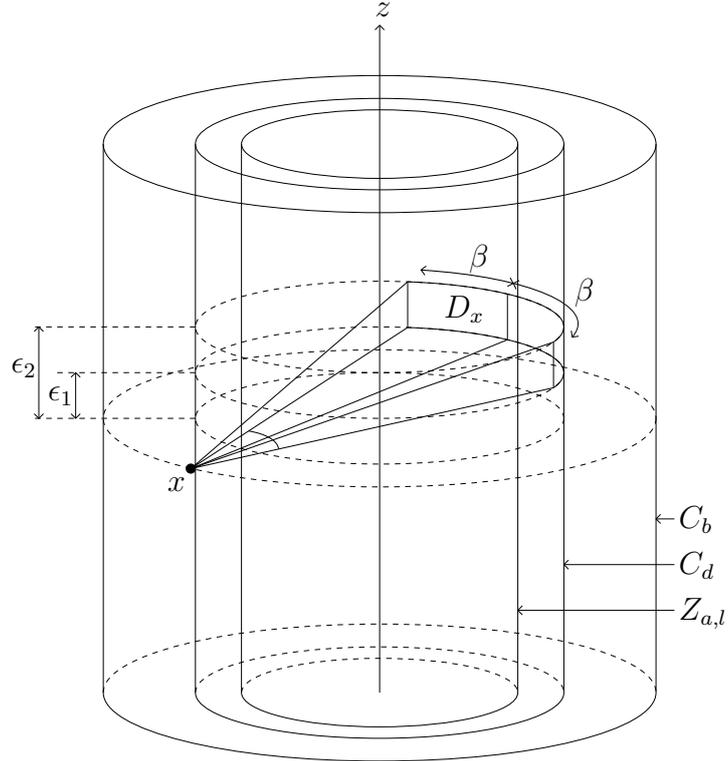


Figure 2.4: The idealised RTT geometry

containing  $x$ , for some  $s \in \mathbb{R}$ , intersects the plane  $z = z_x$  in a line tangent to  $C_b$  at the point  $x$ . Then any  $\alpha \in S^2, s \in \mathbb{R}$  defines such an  $x$  uniquely and we may write

$$x = x(\alpha, s), \quad \text{for some } \alpha \in S^2, s \in \mathbb{R}. \quad (2.25)$$

Let  $L_x$  be the line in  $\Pi_{\alpha,x}$  intersecting  $x$  and the  $z$ -axis, and let  $w$  be the point where  $L_x$  meets the active detector region  $D_x$ . We then make the following assumption:

**Assumption 2.1.** The detector region extents are defined such that for any  $x = x(\alpha, s) \in C_b$ , there exists some  $\alpha \in S^2$ , defined as above, such that all rays in the plane  $\Pi_{\alpha',x}$ , parallel to  $L_x$  and intersecting the support cylinder  $Z_{a,l}$ , for  $\alpha'$  in an arbitrary small neighbourhood of  $\alpha$ , are measured.

The consequence of assumption 2.1, and the fact that  $f$  is assumed to be of compact support, is that there exists an open set  $\Omega \subset S^2$ , defined by the detector extents  $\epsilon_1, \epsilon_2$  and radii  $a, b, d$  of the support, source and detector cylinders, with the

following property:

For any  $x \in C_b$ , there exists  $\alpha \in \Omega$ ,  $s \in \mathbb{R}$  such that all line integrals parallel to the line  $L_x$  in the plane  $\Pi_{\alpha,x}$  are measured. (2.26)

Clearly, by (2.25), the converse of this is also true, so for any  $\alpha \in \Omega$ ,  $s \in \mathbb{R}$ , there exists  $x \in C_b$  such that all line integrals parallel to the line  $L_x$  in the plane  $\Pi_{\alpha,x}$  are measured. This gives us the following theorem.

**Theorem 2.4.** *For  $f \in L^2(Z_{a,l})$ , the data  $Df(x, \theta)$  for  $x \in C_b$  and  $\theta$  as defined by the active detector region for each  $x$ , determine uniquely the three-dimensional Radon transform for all  $\alpha \in \Omega$  and  $s \in \mathbb{R}$ .*

*Proof.* Let  $\alpha \in \Omega$ ,  $s \in \mathbb{R}$ . Then by (2.25) and assumption 2.1, we can find a unique  $x = x(\alpha, s) \in C_b$  such that all rays parallel to  $L_x$  in the plane  $\Pi_{\alpha,x}$  given by  $y \cdot \alpha = s$  and intersecting the support of  $f$  are measured. Since  $f$  is of compact support, all rays not measured are zero and we may integrate over the plane  $\Pi_{\alpha,x}$  in the direction perpendicular to  $L_x$ , giving the Radon transform  $Rf(\alpha, s)$ . Therefore the result is proved.  $\square$

We can define  $\Omega$  explicitly in terms of  $\epsilon_1, \epsilon_2, a, b$  and  $d$  as follows. First, we represent  $\alpha$  in spherical polar coordinates by  $\alpha = (1, \alpha_1, \alpha_2)$ . Therefore, for particular  $x \in C_b$ ,  $\alpha$  is defined solely by the  $\alpha_1$  coordinate.  $\alpha_1$  is the angle between the line  $L$  and the  $z$ -axis, as shown in figure 2.5.

The upper limit for  $\alpha_1$  is given by the angle of the line  $L_x$  passing through  $x$  and  $w$ , when the point  $w$ , shown in figure 2.5, lies on the circle  $z = z_x + \epsilon_2, r = d$ . A cross-section of this is shown in two dimensions in figure 2.6 and gives the following condition:

$$\frac{b+d}{\tan \alpha_1} < \epsilon_2. \quad (2.27)$$

Let  $L_{x'}$  be the line in  $\Pi_{\alpha,x}$ , parallel to  $L_x$  and tangent to the support cylinder  $Z_a$ , and let the points of intersection of  $L_{x'}$  with the source and detector cylinders  $C_b$  and  $C_d$  respectively be  $x'$  and  $w'$ . Then the lower limit of  $\alpha_1$  is given by the angle of the

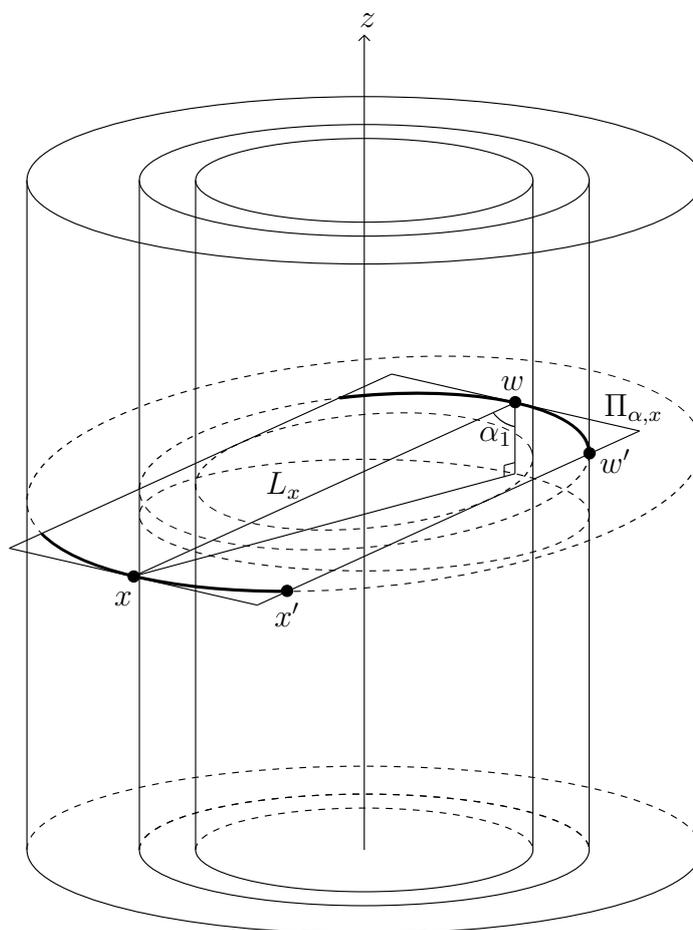


Figure 2.5: The plane  $\Pi_{\alpha, x}$  and the measured region

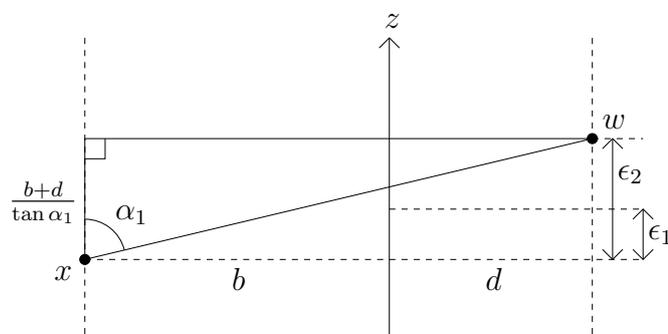
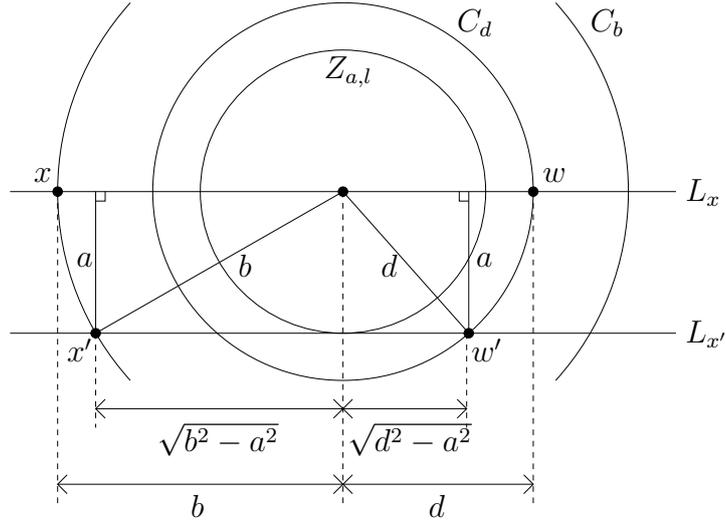


Figure 2.6: Cross-section showing upper limit for  $\alpha_1$


 Figure 2.7: Transaxial cross-section showing positions of  $x'$  and  $w'$ 

line  $L_{x'}$  when the point  $w'$  lies on the circle  $z = z_{x'} + \epsilon_1, r = d$ . The projection of this line onto a transaxial plane is shown in figure 2.7, giving a horizontal length of  $(\sqrt{d^2 - a^2} + \sqrt{b^2 - a^2})$ . Therefore, similar to the derivation of (2.27), this gives the condition:

$$\frac{\sqrt{d^2 - a^2} + \sqrt{b^2 - a^2}}{\tan \alpha_1} > \epsilon_1. \quad (2.28)$$

Combining (2.27) and (2.28) and re-arranging gives

$$\frac{b + d}{\epsilon_2} < \tan \alpha_1 < \frac{\sqrt{d^2 - a^2} + \sqrt{b^2 - a^2}}{\epsilon_1}, \quad (2.29)$$

and since  $\tan$  is increasing on  $(-\pi, \pi)$  we get

$$\arctan\left(\frac{b + d}{\epsilon_2}\right) < \alpha_1 < \arctan\left(\frac{\sqrt{d^2 - a^2} + \sqrt{b^2 - a^2}}{\epsilon_1}\right), \quad (2.30)$$

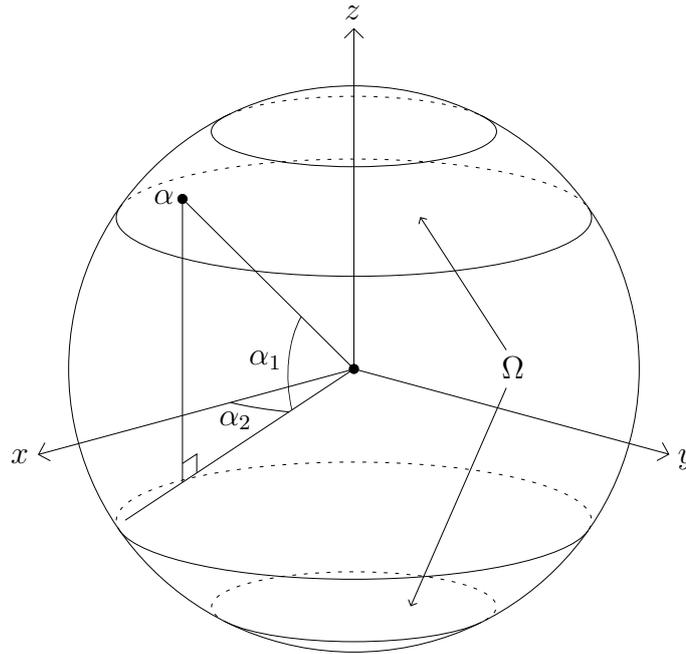
as shown in figure 2.8.

Approximate values of the constants for the RTT80 system are given in table 2.1.

Using (2.26) therefore gives, for the RTT80 system,

$$88.01^\circ < \alpha_1 < 88.37^\circ. \quad (2.31)$$

Although this is a very narrow range of angles, this shows that the theory presented here is applicable to the practical problem of reconstruction from the RTT80 data. The fact that the measured set of angles,  $\Omega$ , does not contain angles in a neighbourhood of  $\alpha_1 = 90^\circ$  has an important consequence. Given that no rays are

Figure 2.8: The open set  $\Omega$  of points on  $S^2$  for which the Radon transform is known

Constant	Value (mm)
$a$	400
$b$	620
$d$	460.7
$\epsilon_1$	19.998
$\epsilon_2$	37.498

Table 2.1: Values of the constants for RTT80

measured at all in planes perpendicular to angles in this neighbourhood, by the discussion in section 2.4, we cannot expect to be able to recover singularities tangent to these planes. For the RTT80 system, this essentially means we cannot reconstruct edges perpendicular to the direction of motion through the scanner. We also expect to see the limited angle type artefacts shown in figure 2.2b in reconstructions made from the RTT80 data.

### 2.5.5 Uniqueness of Solution

By theorem 2.4, the problem of RTT reconstruction reduces to that of the inversion of the three-dimensional Radon transform  $Rf(\alpha, s)$  for the limited angle data given by  $\alpha \in \Omega$ . Therefore, in order to prove uniqueness, it is enough to prove that the function  $f$  is uniquely determined by its Radon transform on this limited angle data. To do

this, we make use of a theorem commonly known as the Paley-Wiener theorem. The original version of this applied to functions, the version given here is due to Schwartz and applies to distributions; we state the theorem as in [58], p399. For definitions of the spaces of test functions and distributions, see appendix A.

**Theorem 2.5** (Paley-Wiener-Schwartz). *Let  $f \in \mathcal{E}'(\mathbb{R}^n)$ , the set of distributions with compact support on  $\mathbb{R}^n$ . Let  $U = \text{supp } f$  and let  $H_U$  be the support function of  $U$ , defined as*

$$H_U(\xi) = \sup_{x \in U} (x \cdot \xi). \quad (2.32)$$

*Then the Fourier transform  $\hat{f}(\xi), \xi \in \mathbb{C}^n$  is an entire analytic function. Additionally, there exists a constant  $c > 0$  and integer  $N$  such that*

$$|\hat{f}(\xi)| \leq c(1 + |\xi|)^N e^{H(\text{Im } \xi)}, \quad \xi \in \mathbb{C}^n. \quad (2.33)$$

*Conversely, let  $F(\xi), \xi \in \mathbb{C}^n$  be an entire analytic function, let  $H$  be the support function of a compact convex set  $K \subset \mathbb{R}^n$  and suppose that*

$$|F(\xi)| \leq c(1 + |\xi|)^N e^{H(\text{Im } \xi)}, \quad \xi \in \mathbb{C}^n, \quad (2.34)$$

*for some constants  $c$  and  $N$ . Then there exists a unique  $f \in \mathcal{E}'(\mathbb{R}^n)$  such that  $\hat{f} = F$  and  $\text{supp } f \subset K$ .*

*Proof.* For a proof of this in the case when  $U$  is a ball see [22], p22 or [72], p162.  $\square$

We are now in a position to prove uniqueness in the RTT case.

**Theorem 2.6.** *Let  $f \in L^2(Z_{a,l})$ . Then the the data  $Df(x, \theta)$  for  $x \in C_b$  and  $\theta$  as defined by the active detector region for each  $x$  determine  $f$  uniquely.*

*Proof.* By theorem 2.4, the three-dimensional Radon transform is known uniquely for all  $\alpha \in \Omega$  and  $s \in \mathbb{R}$ . Therefore, let  $\alpha \in \Omega$ , then by the Fourier Slice Theorem, the Fourier transform  $\hat{f}(\xi)$  is known uniquely for all  $\xi$  on the line  $\xi = \lambda\alpha, \lambda \in \mathbb{R}$ .

Let  $K$  be the open conical set in  $\mathbb{R}^3$  defined by  $K = \{\xi \in \mathbb{R}^3 : \xi = \lambda\alpha, \lambda \in \mathbb{R}, \alpha \in \Omega\}$ . Then by the above,  $\hat{f}(\xi)$  is known uniquely for all  $\xi \in K$ . For  $f \in L^2(Z_{a,l})$ , we may represent  $f$  as a distribution of compact support, therefore  $f \in \mathcal{E}'(\mathbb{R}^3)$ . So by the

Paley-Wiener theorem  $\hat{f}(\xi)$  is an entire analytic function, and therefore by analytic continuation,  $\hat{f}(\xi)$  is determined uniquely for all  $\xi \in \mathbb{C}^n$ . Hence  $f(x)$  is determined uniquely for all  $x \in \mathbb{R}^3$  and the result is proved.  $\square$

### 2.5.6 Inversion and Stability Estimates

An explicit inversion method for Radon transform data known only for some subset of  $S^{n-1}$  is given by Ramm in [57], based on reducing the problem to that of the inversion of the Fourier transform known in some arbitrarily small ball, and then extending this using analytic continuation. Numerical aspects of the inversion procedure are discussed in [56]. This is extremely ill-posed and therefore highly unstable; it is unknown whether this could be applied to practical problems of reconstructing real data.

Theoretically, we should be able to apply a similar reconstruction method to the RTT system. Given that we have the Fourier transform data for some conical set  $K$ , defined as in the proof of theorem 2.6, the conditioning of the problem should be better than for the case of a ball, since we have more data to work with. However, it is still unknown whether such a reconstruction algorithm would be of any practical use.

# Chapter 3

## 2D RTT Reconstruction

We begin our study of the practical problem of RTT reconstruction by looking at the reconstruction a two-dimensional slice through an object. Our ultimate aim in this respect is to achieve real-time imaging of moving objects; however, in this chapter we will only be considering the static case.

The problem is essentially just the inversion of the two-dimensional divergent beam transform, which has been well understood for quite some time. However, the RTT geometry leads to a highly irregular sampling of the projections which presents some challenges.

### 3.1 The RTT20 Machine

RTT20 is a small-scale prototype RTT machine built by Rapiscan to demonstrate the new technology and attract investment for further development of commercial RTT scanners. The machine has now been acquired by the university, where it is expected to be used for research into granular flow and fluid dynamics, specifically to image slices through fluids in motion. It is also possible that the machine could be useful in industrial applications, for imaging flow through oil pipes for example.

A two-dimensional cross-section of the geometry of RTT20 is shown in figure 3.1. The detectors are actually offset from the sources in the  $z$  direction by 5.48mm, giving a maximum offset of approximately 2.5mm within the region of interest (ROI), which

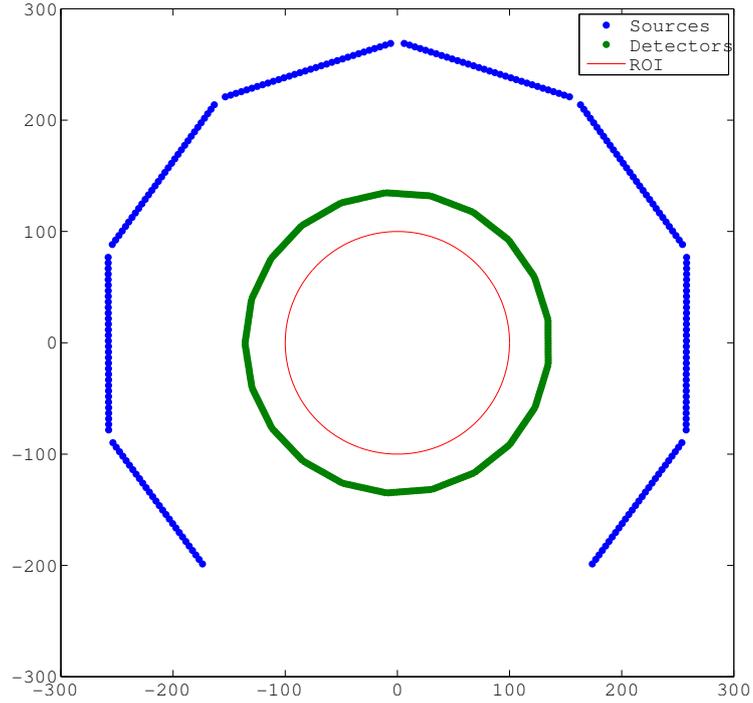
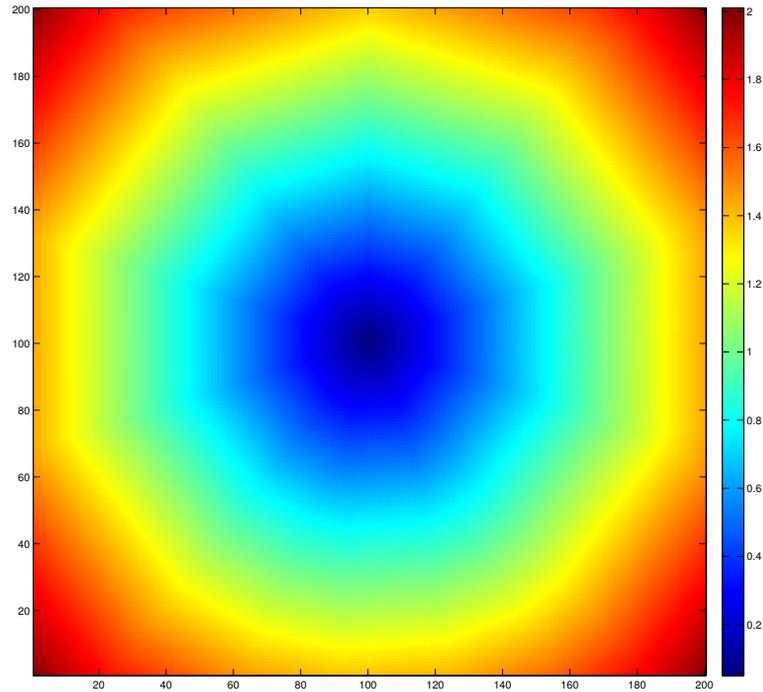


Figure 3.1: The RTT20 geometry

technically means that the geometry of RTT20 is not actually two-dimensional, but rather a kind of ‘tilted fan beam’ arrangement. In practice, since the offset within the ROI is relatively small, in this chapter we will ignore this offset and perform approximate two-dimensional reconstruction on a flat slice. To see the extent of the approximation, figure 3.2 shows the maximum  $z$ -axis deviation of the rays intersecting each pixel, relative to the plane of average  $z$  values, in a  $200 \times 200$  grid covering the entire ROI. This should give a measure of how much  $z$ -axis resolution we can expect in the reconstruction; we therefore expect higher resolution in the centre of the image.

The sources are arranged in 8 blocks of 32; however, we note that there are two ‘missing’ blocks of sources at the bottom, creating a large gap in the ring. The two blocks adjacent to the gap also do not use their outermost 4 sources giving a total of 248 sources. There is one full ring of detectors arranged in 21 blocks of 16 giving 336 detectors in total. The tunnel diameter is 20cm (hence RTT20) giving the ROI as a circle of diameter 200mm.

Figure 3.2: Maximum  $z$ -axis deviation per pixel

## 3.2 Analytical Reconstruction Methods

As introduced in section 2.3.3, filtered backprojection is a highly efficient and easily implemented method for reconstruction from two-dimensional parallel beam Radon transform data. The numerical implementation of this generally calculates the one-dimensional filter by means of discrete convolution. Implementations also exist for the direct inversion of two-dimensional divergent beam data, assuming an equiangular sampling of the projections with source positions on some circle and either equally spaced collinear or equiangular sampling of the rays within each projection. These formulae define *weighted* filtered backprojections, which incur considerably higher computational overheads than the standard parallel beam filtered backprojection. A good introduction to these implementations is given in [29], chapter 3.

In the case of RTT20, looking back at figure 3.1 we see that due to the relatively small size of the system, the polygonal shape of the source ring is exaggerated, making the sampling of the source position far from equiangular. The gaps between neighbouring source blocks are also significant, creating what equates to a missing source between them.

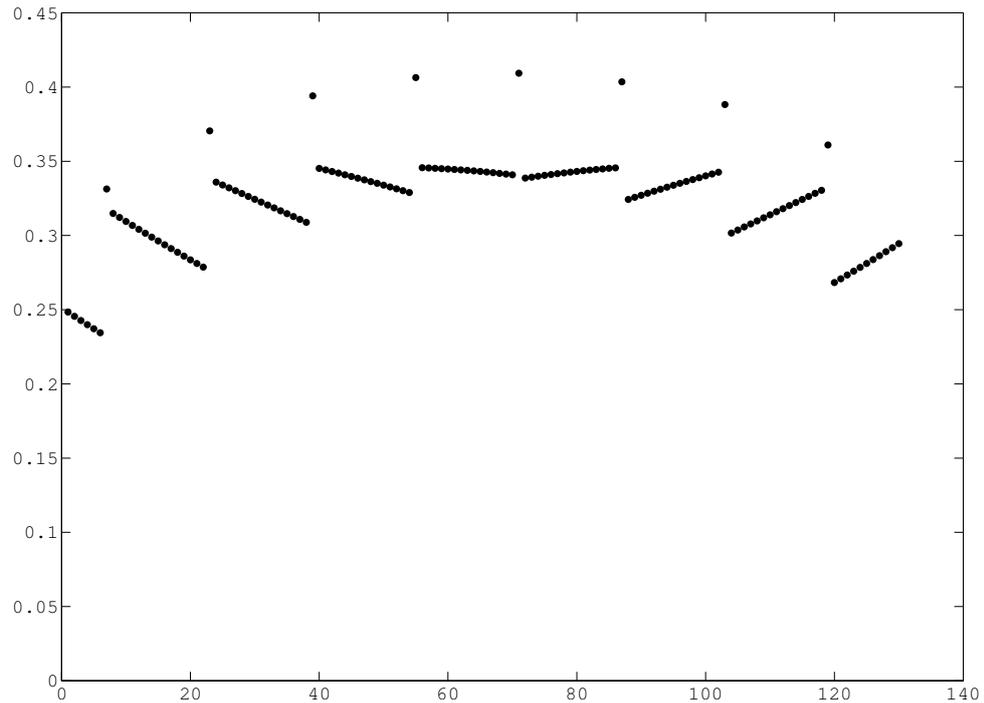


Figure 3.3: Angles in degrees between adjacent detectors in the active region for source 1

Looking at the detectors, we obviously do not have an equal linear spacing. Also, a similar phenomenon as with the sources occurs with the detectors and the gaps between the detector banks, making the sampling of rays within each projection far from equiangular. This is shown in figure 3.3, which shows the angles between adjacent rays for source number 1 (at the bottom left).

### 3.2.1 The Initial Approach

Looking at the RTT20 geometry and figure 3.3, it would seem that neither of the direct fan-beam reconstruction algorithms would be suitable for RTT20 reconstruction. However, the initial approach taken by CXR was to apply the equiangular fan beam formula, as described in [29], chapter 3. Results of this initial approach were somewhat less than satisfactory, with significant streak artefacts present in the reconstruction. Clearly if we are to apply one of these methods, some kind of interpolation will have to be used.

### 3.2.2 Interpolation to Parallel Beams

The preliminary approach to the interpolation problem was to focus on the gaps between the source blocks and attempt to interpolate the ‘missing’ sources. This could be achieved, for example, by the methods discussed in [34], using the concept of optical flow (see [23]) across the gaps to interpolate the missing data. However, this would do nothing to take account of the uneven sampling created by using linear blocks of sources, nor the uneven angular distribution of the rays within each projection.

A so-called *rebinning* of fan beam projection data to parallel beams is a standard technique in fan beam reconstruction; see for example [29], chapter 3. Given the additional advantage that standard filtered backprojection is computationally more efficient, it therefore seems much more sensible to interpolate the RTT projection data to parallel beams.

In order to do this, for every ray in the RTT data, we calculate the  $\theta$  and  $s$  values for that ray as if it were from parallel beam Radon transform data. Thus every sampling point in the RTT sinogram is associated with a point in parallel beam sinogram space and the problem becomes one of two-dimensional scattered data interpolation. Figure 3.4 shows the sampling of parallel beam sinogram space created by the RTT system. This shows the data plotted over the full range  $[-\pi, \pi)$ ; when performing the interpolation, we can either discard the data outside the range  $[-\pi/2, \pi/2)$ , or reverse the roles of source and detector outside of this range in order to keep angles within  $[-\pi/2, \pi/2)$ .

### 3.2.3 Triangulation Based Interpolation

One way to approach the interpolation of the scattered RTT data is using a triangulation of the data points, as described in [1]. The method uses two steps; we first compute a triangulation, then interpolate within the triangles. Obviously, for a given set of data points, there are many possible triangulations; it seems desirable that we should use one which avoids as much as possible creating long, thin triangles. Such

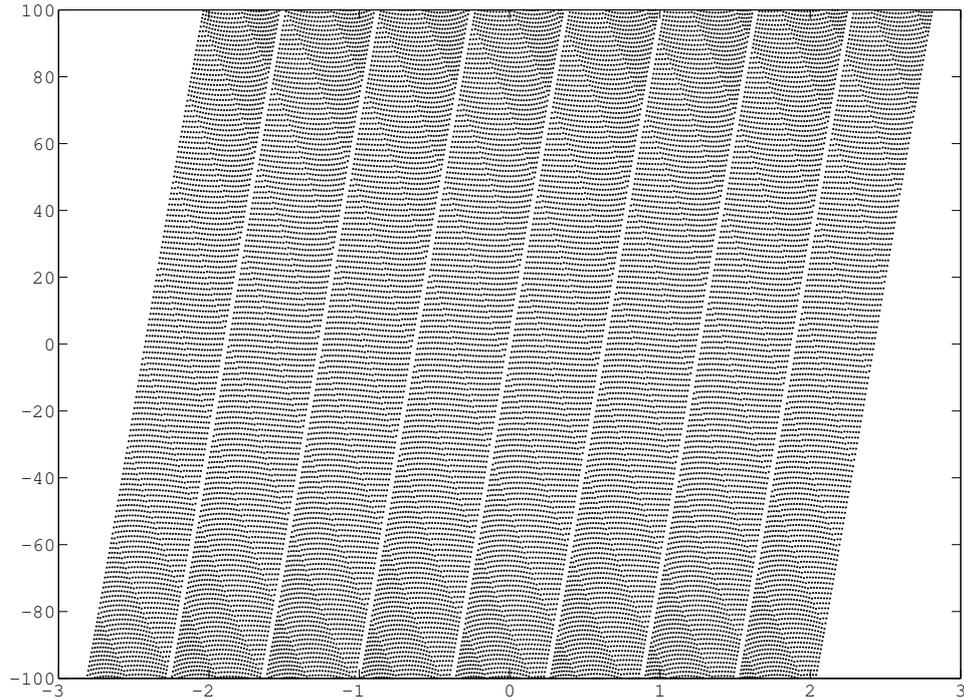


Figure 3.4: Sampling points in parallel-beam sinogram space of the RTT20 data

a triangulation is given by the *Delaunay triangulation*.

For a given set of points  $P$  in  $\mathbb{R}^2$ , the Delaunay triangulation is a triangulation  $T(P)$  with the property that the circumcircle of any triangle in  $T(P)$  contains no points of  $P$ . It should be noted that this triangulation is not unique; considering for example four points arranged in a square, the two possible triangulations are both valid Delaunay triangulations. To calculate the triangulation, we use the MATLAB function `griddata`, which also performs the interpolation step. Interpolation is performed within the triangles by using linear or cubic interpolation.

In order to find such a triangulation for the RTT data, we need to scale the axes appropriately; we have  $\theta$  data in the range  $[-\pi, \pi)$  and  $s$  data in the range  $[-100, 100]$ . Applying the Delaunay triangulation procedure to this data will result in rather long and thin triangles. Somewhat arbitrarily, we scale the data so that the axes cover an equal range; the resulting triangulation is shown in figure 3.5.

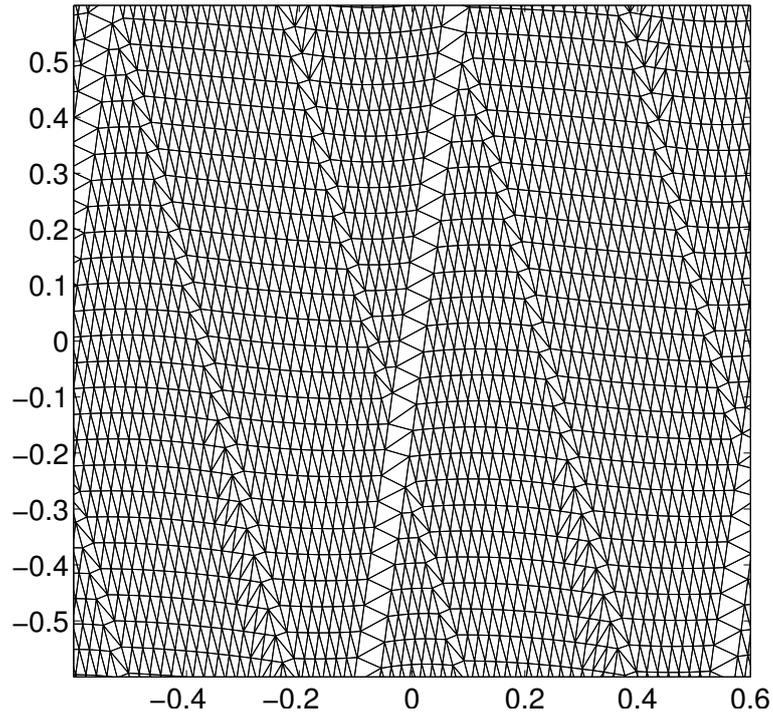


Figure 3.5: A small section of the Delaunay triangulation of the RTT20 data

### 3.2.4 Results with Phantom Data

Phantoms were calculated for simple rectangular and circular objects by assuming an infinite extent in the  $z$  direction and analytically calculating the exact path length of each measured ray through the object, from the centre of the source to the centre of the detector. By the linearity of the ray transform, more complicated phantoms can be created by summing the data for each simple object. We then add 5% Gaussian noise to the resulting data.

The phantom we use is a simple resolution test type phantom, consisting of 16 rectangles of height 80mm in the  $y$  direction and of width and spacing 8, 7.5,  $\dots$ , 0.5mm in the  $x$  direction. The RTT sinogram for this phantom is shown in figure 3.6. Linear and cubically interpolated parallel-beam sinograms obtained using the Delaunay triangulation with `griddata` are shown in figure 3.7.

Reconstruction of the interpolated parallel-beam sinograms was done using the MATLAB function `iradon`, using cubic interpolation and Ram-Lak filter, with the reconstruction region covering the whole circular ROI. Results of this are shown in figure 3.8. In both cases the rectangles are clearly discernible down to 2mm; the

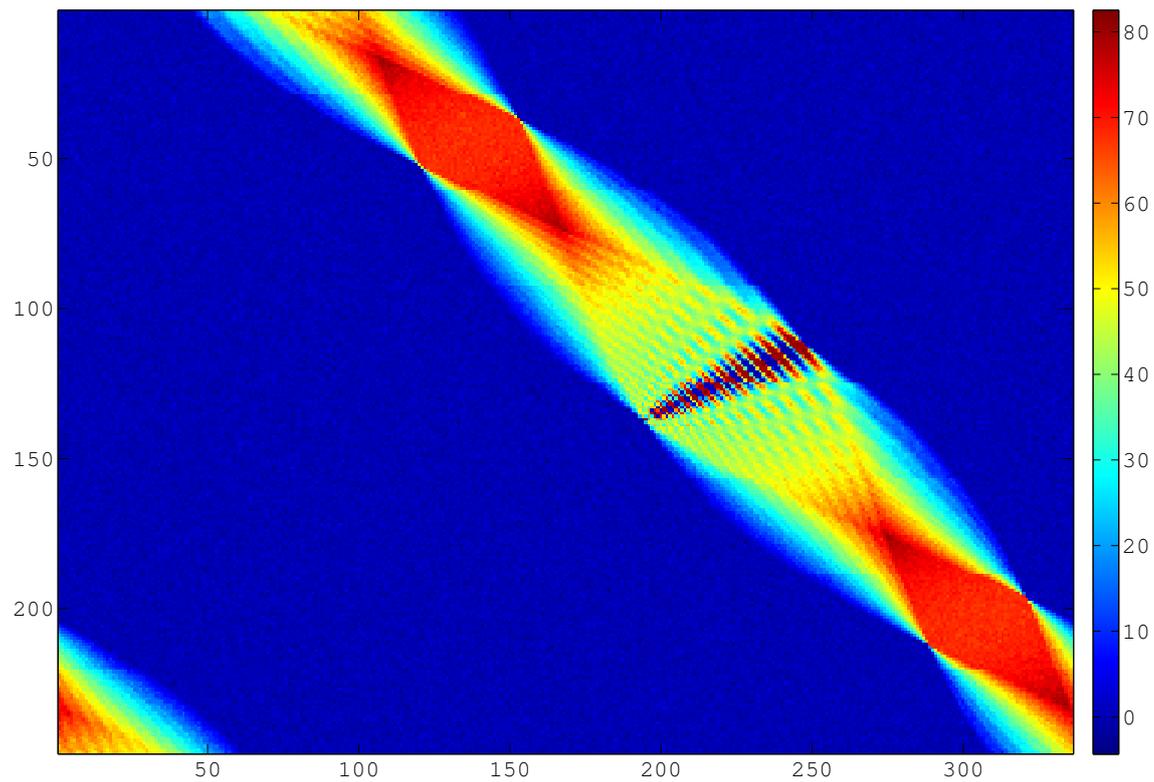


Figure 3.6: RTT sinogram of the RTT20 test phantom

cubically interpolated case is slightly smoother, which is to be expected.

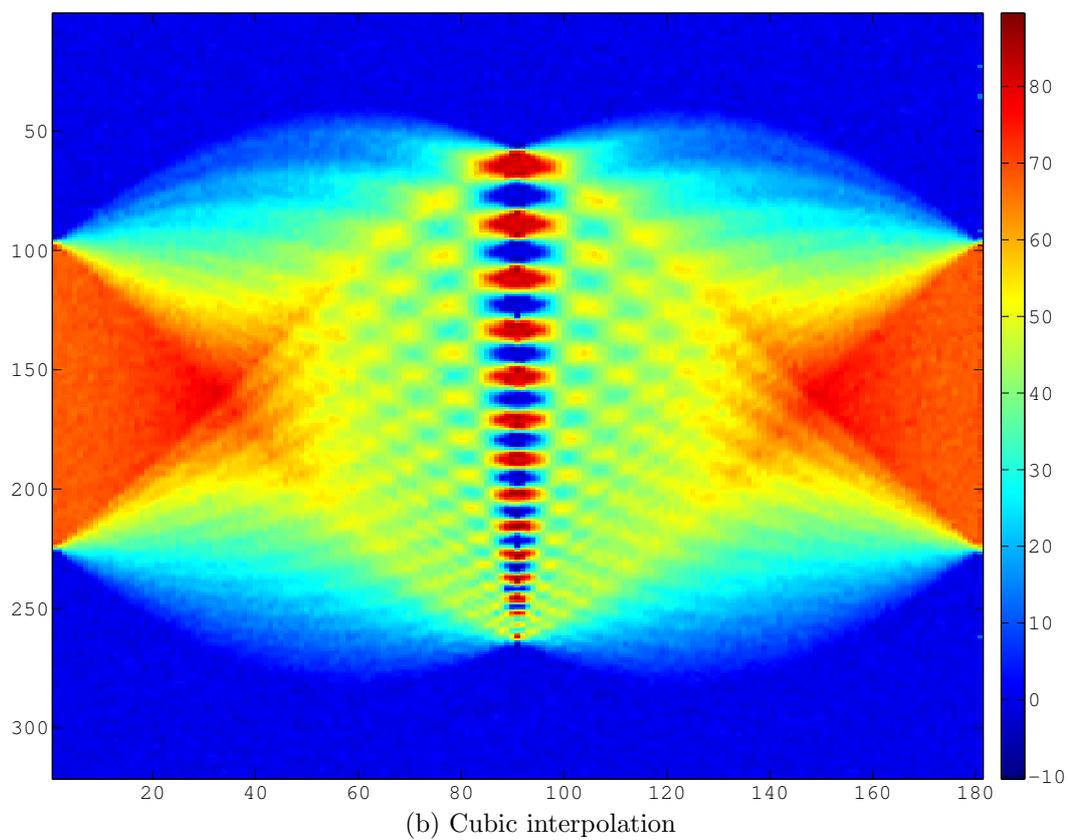
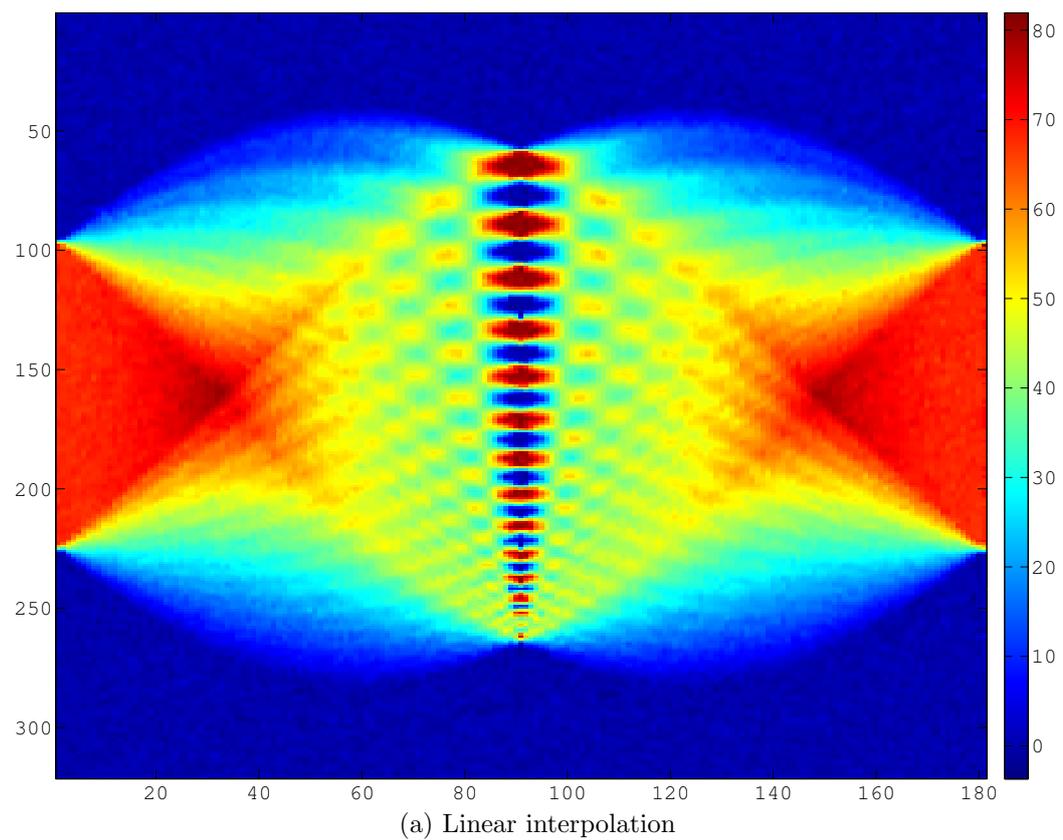


Figure 3.7: Parallel-beam sinograms obtained using Delaunay triangulation based interpolation

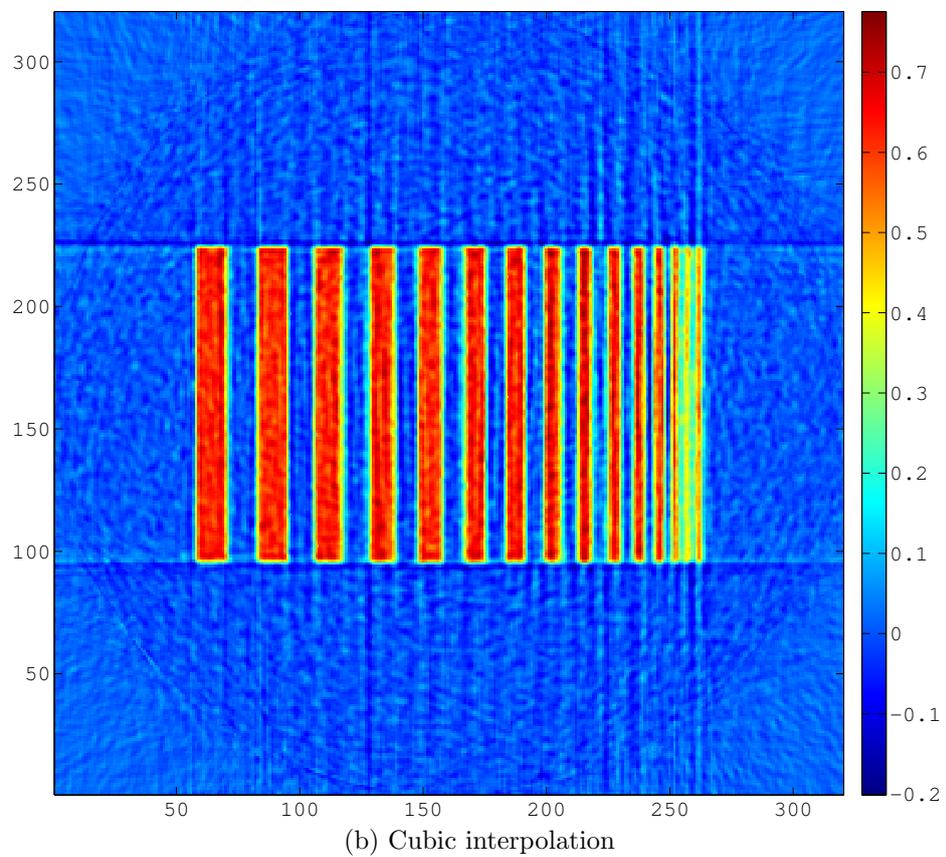
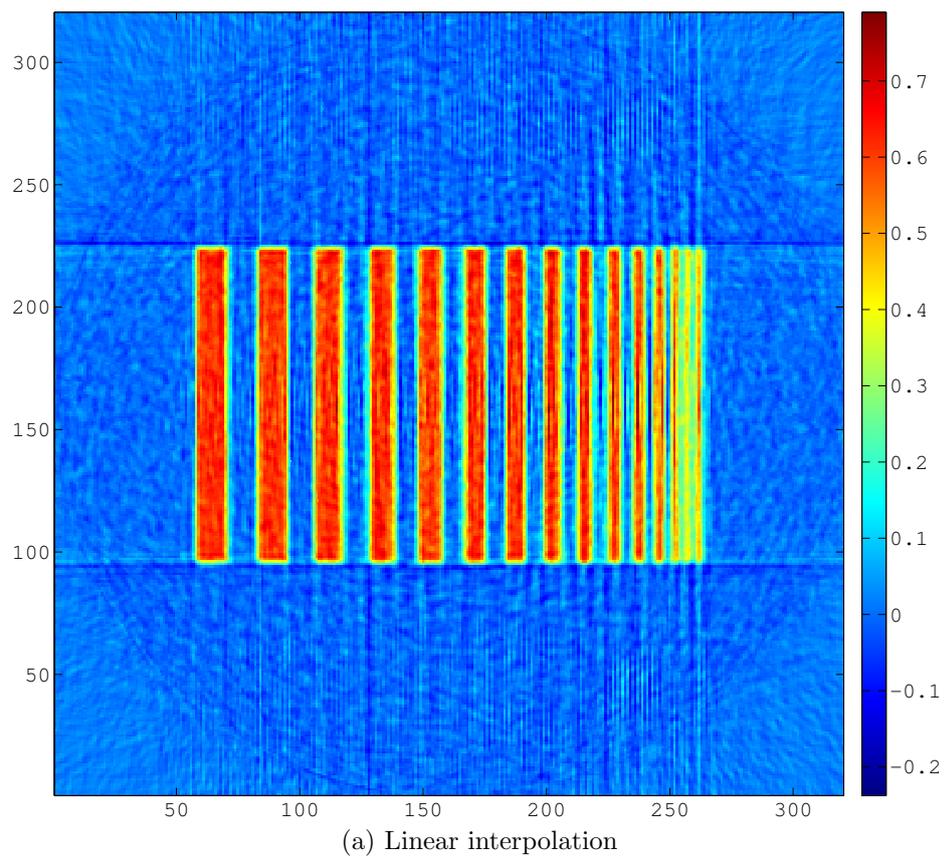


Figure 3.8: Reconstructions of the parallel-beam sinograms using `iradon`

## Chapter 4

# Algebraic Reconstruction Methods

The reconstruction methods described so far have all relied on numerical implementation of an analytic inversion formula, which necessarily has to make certain assumptions about the geometry of the system and sampling of the rays. We now turn our attention to an entirely different way of looking at the reconstruction problem; algebraic reconstruction. This is essentially a direct discretisation of the divergent beam equation, and effectively breaks the forward projection process down into a system of linear equations which can then be solved to reconstruct the density function  $f$ .

This has both advantages and disadvantages over the analytical approach. The main disadvantage is speed of reconstruction; algebraic methods can take considerably longer than the highly efficient analytical algorithms. This is a big disadvantage for time-critical applications such as the airport security applications that the RTT system was designed for; however, for the scientific applications for which we envisage the RTT20 machine at the university being used, all reconstruction will be taking place offline so this will not be a significant problem.

The main advantage of algebraic reconstruction is that the reconstruction process no longer makes any assumptions about the scanner geometry or ray sampling. Hence the gaps between sources and non-equiangular spacing of the rays should no longer matter, making this an attractive choice for reconstruction from the RTT system if reconstruction time is not important.

## 4.1 Modelling the Projection Process

In order to discretise the divergent beam equation and form a system of linear equations, we must first have a mathematical model of the projection process. We discretise the function  $f(x)$  into a regular grid of cubic voxels, referenced by a single index  $i = 1, \dots, N$ . The value of each  $f_i$  represents the average value of the underlying continuous function  $f$  over the volume of the cube. Considering all rays in all projections, we number the rays  $1, \dots, M$  and denote the measured value of the  $j^{\text{th}}$  ray as  $b_j$ . We model rays by lines of zero thickness, with end points at the centre of the relevant source and detector surfaces.

We now consider the intersection of the  $j^{\text{th}}$  ray with the  $i^{\text{th}}$  voxel, and denote the length of this intersection by  $a_{ij}$ . Each  $a_{ij}$  represents the contribution that the value of  $f$  in the  $i^{\text{th}}$  voxel makes to the integral along the  $j^{\text{th}}$  ray; in the two-dimensional case, this is shown in figure 4.1. We note that for any given ray, most of the  $a_{ij}$  will be zero since a ray only intersects a small number of the voxels in the grid. Considering the  $j^{\text{th}}$  ray, the total contribution to that ray is given by summing over all  $a_{ij}$  for that ray; hence we write

$$\sum_{i=1}^N a_{ij} f_i = b_j, \quad \text{for all } 1 \leq j \leq M. \quad (4.1)$$

This gives us the desired system of linear equations, which can be written in matrix form as

$$A\mathbf{x} = \mathbf{b}, \quad (4.2)$$

where  $A$  is the  $M \times N$  matrix of coefficients  $a_{ij}$ ,  $\mathbf{x} = (f_1, \dots, f_N)^T$  is an  $N$ -dimensional vector representing the discretised density function  $f$ , and  $\mathbf{b} = (b_1, \dots, b_M)^T$  is the  $M$ -dimensional vector of projection data.

We note that this is by no means the only way of modelling the projection process; we should in theory be able to take into account the thickness of the rays and the physical properties of the sources and detectors. It is also possible to use alternative basis functions for the voxels, providing a better sampling.

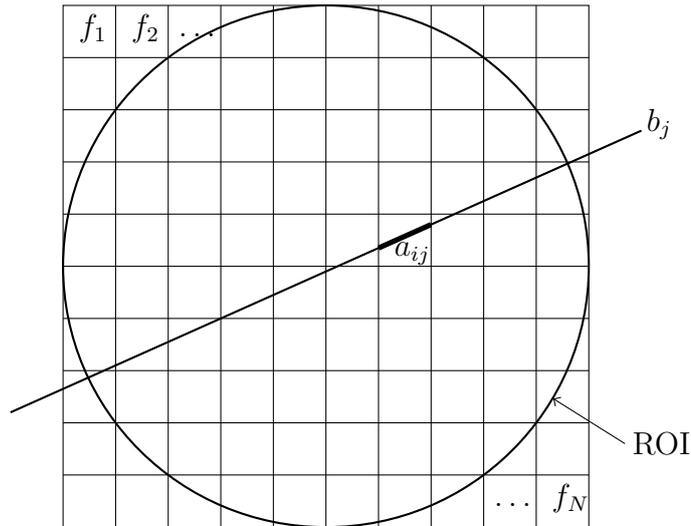


Figure 4.1: Discretisation of the projection process

## 4.2 Iterative Solution Methods

Given measured data  $\mathbf{b}$ , we wish to solve the system in (4.2) for  $\mathbf{x}$ . The number of rows and columns of  $A$  will in general not be equal and the measurements will usually be noisy, so some form of regularised least squares solution will usually be sought. The main problem we have is that the matrix  $A$  will in general be far too big to find an explicit generalised inverse and solve using this. However, given the very sparse nature of  $A$ , we can apply one of many iterative methods to find a solution.

A general iterative method starts with some initial guess of the solution,  $\mathbf{x}^{(0)}$ , and then refines this by a sequence of approximations  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}$  for some integer number of iterations  $n$ .

### 4.2.1 Iterative Algorithms Conventionally Used in CT

We begin our discussion of iterative methods by looking at those methods whose use for CT reconstruction is well understood. A good introduction to these methods from the perspective of CT reconstruction may be found in [68] or [29], chapter 7. A more thorough treatment from the perspective of general inverse problems may be found in [19], chapter 6.

### The Kaczmarz Method

Let the system of equations  $\mathbf{Ax} = \mathbf{b}$  be defined as in (4.2), and let the  $i^{\text{th}}$  row of the matrix  $A$  be represented by  $\mathbf{a}_i^T$ , for  $1 \leq i \leq M$ . Starting with some arbitrary initial guess  $\mathbf{x}^{(0)}$ , the Kaczmarz method is the iteration procedure given by

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \omega^{(n)} \left( \frac{b_i - \mathbf{a}_i^T \mathbf{x}^{(n)}}{\mathbf{a}_i^T \mathbf{a}_i} \right) \mathbf{a}_i, \quad n = 0, 1, 2, \dots, \quad i = (n \bmod M) + 1, \quad (4.3)$$

where  $\omega^{(n)}$  is a real *relaxation parameter*, which may or may not depend on the iteration number, and  $b_i$  is the  $i^{\text{th}}$  element of  $\mathbf{b}$ . One iteration of the algorithm is represented by one complete sweep through all equations; we generally start with initial guess  $\mathbf{x}^{(0)} = \mathbf{0}$ . The effect of the relaxation parameter is to control the rate of convergence, as explained in [68], which, for  $\omega^{(i)} < 1$  also provides a regularising effect, at the expense of slower convergence (see, for example, [29], chapter 7).

Conceptually, the effect of the iteration is as follows; consider the approximated solution at the  $i^{\text{th}}$  step,  $\mathbf{x}^{(i)}$ . We calculate the projection of this along the  $i^{\text{th}}$  ray and subtract this from the  $i^{\text{th}}$  data value  $b_i$ . The result of this can be thought of as a correction factor, which is normalised and then backprojected onto the reconstruction volume and added to the approximate solution.

The Kaczmarz method operates on the matrix  $A$  one row at a time and is hence known as a *row action* method. As such, the result for each row of  $A$  is dependent on the results from previous rows, which makes the method less suitable for parallel implementation.

In the CT community, the Kaczmarz method is commonly known as Algebraic Reconstruction Technique, or ART, and was first applied to image reconstruction problems by Gordon, Bender and Herman in [13]. The method is a generalisation of Kaczmarz's original method given in [28] which applied only to square, invertible matrices.

### A Generalised Kaczmarz Method

The Kaczmarz method can be generalised into the block iteration scheme

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + A_i^T \Omega^{(n)} (\mathbf{b}_i - A_i \mathbf{x}^{(n)}), \quad n = 0, 1, 2, \dots, \quad i = (n \bmod M) + 1, \quad (4.4)$$

where  $A_i$  and  $\mathbf{b}_i$  represent a block of  $L$  equations of the system 4.2, and  $\Omega^{(n)}$  is an  $L \times L$  relaxation matrix. Clearly in the case  $L = 1$ , for an appropriate choice of  $\Omega$  this is equivalent to the Kaczmarz method, and in the case  $L = M$ , is equivalent to the Landweber method described in the next section. Convergence results for this method, and therefore the Kaczmarz and Landweber methods also, are proved by Eggermont et al. in [9].

### The Landweber Method

For the system of equations  $A\mathbf{x} = \mathbf{b}$ , defined again as in (4.2), then starting with some arbitrary initial guess  $\mathbf{x}^{(0)}$ , the Landweber method is the iteration procedure given by

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \tau A^T (\mathbf{b} - A\mathbf{x}^{(n)}), \quad n = 0, 1, 2, \dots, \quad (4.5)$$

where  $\tau$  is a real relaxation parameter.

Conceptually, this is equivalent to calculating the correction factors of the ART algorithm for all rays and then applying them simultaneously. For this reason, the method is often known as Simultaneous Iterative Reconstruction Technique, or SIRT, and was introduced as such for applications in image reconstruction by Gilbert in [12]. The method was introduced by Landweber in the context of solving general Fredholm integral equations of the first kind in [35]; however, it is originally attributed to Cimmino [7], in the context of solving systems of linear equations. Therefore, the method also goes by the name of the Cimmino method, amongst others, but is referred to here as Landweber.

It can be shown (for example in [2], p260) that convergence of the method is guaranteed for  $\tau$  satisfying

$$0 < \tau < \frac{2}{\sigma_{\max}^2}, \quad (4.6)$$

where  $\sigma_{\max}$  is the largest singular value of  $A$ . It is also shown that Landweber iteration acts as a filter on the SVD of the linear operator represented by  $A$ ; therefore, the number of iterations  $n$  acts as a regularisation parameter.

Since the Landweber method acts on all projections at the same time, this makes it much more suitable for parallel implementation than the row action methods. However, convergence for Landweber can be very slow; this can be explained by the following. The Landweber method is simply the Richardson iteration

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \omega \mathbf{r}^{(n)}, \quad n = 0, 1, 2, \dots, \quad (4.7)$$

where  $\mathbf{r}^{(n)} = \mathbf{b} - A\mathbf{x}^{(n)}$  is the *residual* at the  $n^{\text{th}}$  step, applied to the normal equations  $A^T A \mathbf{x} = A^T \mathbf{b}$ . Considering the quadratic form associated with (4.7), the residual always points in the direction of maximum negative gradient. Therefore, at each step of the algorithm, we are moving in the ‘right’ direction, but by a constant amount each time; this is clearly far from optimal. One way to optimise convergence is to consider minimising the quadratic form associated with (4.7) along the line in the direction of the residual; this gives us the so-called method of steepest descent.

## 4.2.2 Conjugate Gradient Least Squares

The conjugate gradient method is a well-known iterative method for solving sparse systems of equations  $A\mathbf{x} = \mathbf{b}$  where the matrix  $A$  is positive-definite. For a simple, well-written derivation and explanation of this, see [61]; for a more rigorous treatment see, for example [59], section 6.7. An analysis of the convergence rate of the method is given in [67].

Since the matrix  $A^T A$  is positive-definite for any matrix  $A$ , we may apply the Conjugate Gradient algorithm to the normal equations  $A^T A \mathbf{x} = A^T \mathbf{b}$ ; the resulting algorithm is known as Conjugate Gradient Least Squares, or CGLS. The algorithm was originally proposed by Hestenes and Stiefel [21], and is stated here as it appears

in [19], p143. The  $n^{\text{th}}$  step of the algorithm is described, for  $n = 1, 2, \dots$ , as follows

$$\begin{aligned}\alpha_n &= \frac{\|A^T \mathbf{r}^{(n-1)}\|_2^2}{\|\mathbf{A} \mathbf{d}^{(n-1)}\|_2^2}, \\ \mathbf{x}^{(n)} &= \mathbf{x}^{(n-1)} + \alpha_n \mathbf{d}^{(n-1)}, \\ \mathbf{r}^{(n)} &= \mathbf{r}^{(n-1)} - \alpha_n \mathbf{A} \mathbf{d}^{(n-1)}, \\ \beta_n &= \frac{\|A^T \mathbf{r}^{(n)}\|_2^2}{\|A^T \mathbf{r}^{(n-1)}\|_2^2}, \\ \mathbf{d}^{(n)} &= A^T \mathbf{r}^{(n)} + \beta_n \mathbf{d}^{(n-1)},\end{aligned}\tag{4.8}$$

where  $\mathbf{r}^{(n)}$  is the residual at the  $n^{\text{th}}$  step and  $\mathbf{d}^{(n)}$  is an auxiliary vector of length  $N$ . We start with some arbitrary  $\mathbf{x}^{(0)}$  and set  $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A} \mathbf{x}^{(0)}$ ,  $\mathbf{d}^{(0)} = A^T \mathbf{r}^{(0)}$ . Usually, we take  $\mathbf{x}^{(0)} = 0$ , giving  $\mathbf{r}^{(0)} = \mathbf{b}$  and  $\mathbf{d}^{(0)} = A^T \mathbf{b}$ . It should be noted that the above algorithm avoids explicitly calculating the matrix product  $A^T A$ .

The approximated solution at the  $n^{\text{th}}$  step,  $\mathbf{x}^{(n)}$ , lies in the *Krylov subspace*

$$\mathcal{K}_n(A^T A, A^T \mathbf{b}) = \text{span} \{A^T \mathbf{b}, A^T A A^T \mathbf{b}, \dots, (A^T A)^{n-1} A^T \mathbf{b}\}.\tag{4.9}$$

The CGLS algorithm also has a regularising effect, similar to the Landweber method, with the number of iterations  $n$  again playing the role of regularisation parameter. However, unlike the Landweber method, the regularising effect is not well-defined; a rigorous analysis of this is given in [19], sections 6.3 and 6.4.

Although the use of CGLS for tomography problems is not widespread, we find examples of its use in limited data problems in [32], in seismic tomography in, for example [60] and [36], and in a more general CT context in [52].

The main problem with methods like CGLS which involve the normal equations is that the condition number of the matrix  $A^T A$  is the square of the condition number of  $A$ ; therefore, for ill-conditioned problems convergence can be slow. However, if the condition number of  $A$  is not particularly high, we can reasonably expect CGLS to converge more quickly than Landweber; we therefore choose to apply CGLS to the RTT20 problem.

## 4.3 Applying Algebraic Reconstruction to RTT20

We now consider the practicalities of applying algebraic reconstruction to RTT20 data. The RTT20 system has 248 sources; each of these has an active detector region consisting of 130 detectors, so a complete set of data consists of 32240 measurements. We cover the entire 20cm reconstruction circle with a  $200 \times 200$  grid of  $1\text{mm}^2$  pixels, giving 40,000 in total. Since the reconstruction region only covers the circle inscribed within this square, this actually gives only 31,428 pixels within the ROI, making the system effectively slightly overdetermined. With modern computer hardware, the system matrix  $A$  is easily stored in memory, and so can be calculated in advance.

### 4.3.1 Calculating $A$

To create a 2D reconstruction grid, but also take into account the  $z$ -axis offset of the detectors, we use a  $200 \times 200$  voxel square grid, covering the whole 200mm diameter ROI, and extend the voxels in the  $z$  direction so that each voxel is  $1 \times 1 \times 10\text{mm}$  in size. The coefficients of  $A$  are then calculated using the raytracing algorithm of Jacobs et al. described in [25], which is itself a more efficient version of Siddon's algorithm of [62]. This has been implemented as a C `.mex` routine by David Szotten; for an explanation of this, see [64], section 5.1.4. The results are stored as a sparse matrix in MATLAB.

### 4.3.2 Solving Using CGLS

To solve the system for  $\mathbf{x}$ , we use the MATLAB function `cglsl.m` provided in Hansen's Regularisation Tools package [20]. This applies CGLS to the normal equations implicitly, without forming the considerably denser  $A^T A$ . The function returns  $\mathbf{x}$  after each iteration and also outputs the 2-norm of the residual at each step to track convergence.

### 4.3.3 Regularisation

Although with CGLS, the number of iterations plays the role of a regularisation parameter, it is unclear how many iterations should be performed in order to provide the correct degree of regularisation. We may therefore apply additional Tikhonov regularisation by solving the augmented system

$$\begin{bmatrix} A \\ \alpha L \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix}, \quad (4.10)$$

where  $\alpha$  is the regularisation parameter and  $L$  is either the  $N \times N$  identity matrix or finite difference approximation to a differential operator, for example the Laplacian.

## 4.4 Results With Phantom Data

Results are given for phantom data, using the same phantom as in section 3.2.4. Figures 4.2–4.4 show the results of reconstruction after 10, 20, 50 and 100 iterations with respectively no regularisation, and regularisation with  $L$  as the finite-difference Laplacian for  $\alpha = 1$  and  $\alpha = 5$ . Figure 4.5 shows plots of the 2-norm of the residual in each case, as a measure of convergence. We also plot the 2-norm of the solution in each case; this is shown in figure 4.6.

We see that in the unregularised case, the solution norm appears to be growing without bound. If we were to continue iterating then we would be adding components to the image corresponding to the small singular values of the matrix  $A$ . With the addition of Tikhonov regularisation, we see that the solution norm seems to be tending to some clearly defined bound. This well-defined convergence behaviour allows us to avoid having to choose some arbitrary number of iterations to stop at.

We see that in the regularised case with  $\alpha = 5$ , results are comparable to those obtained by interpolation and filtered backprojection in section 3.2.4.

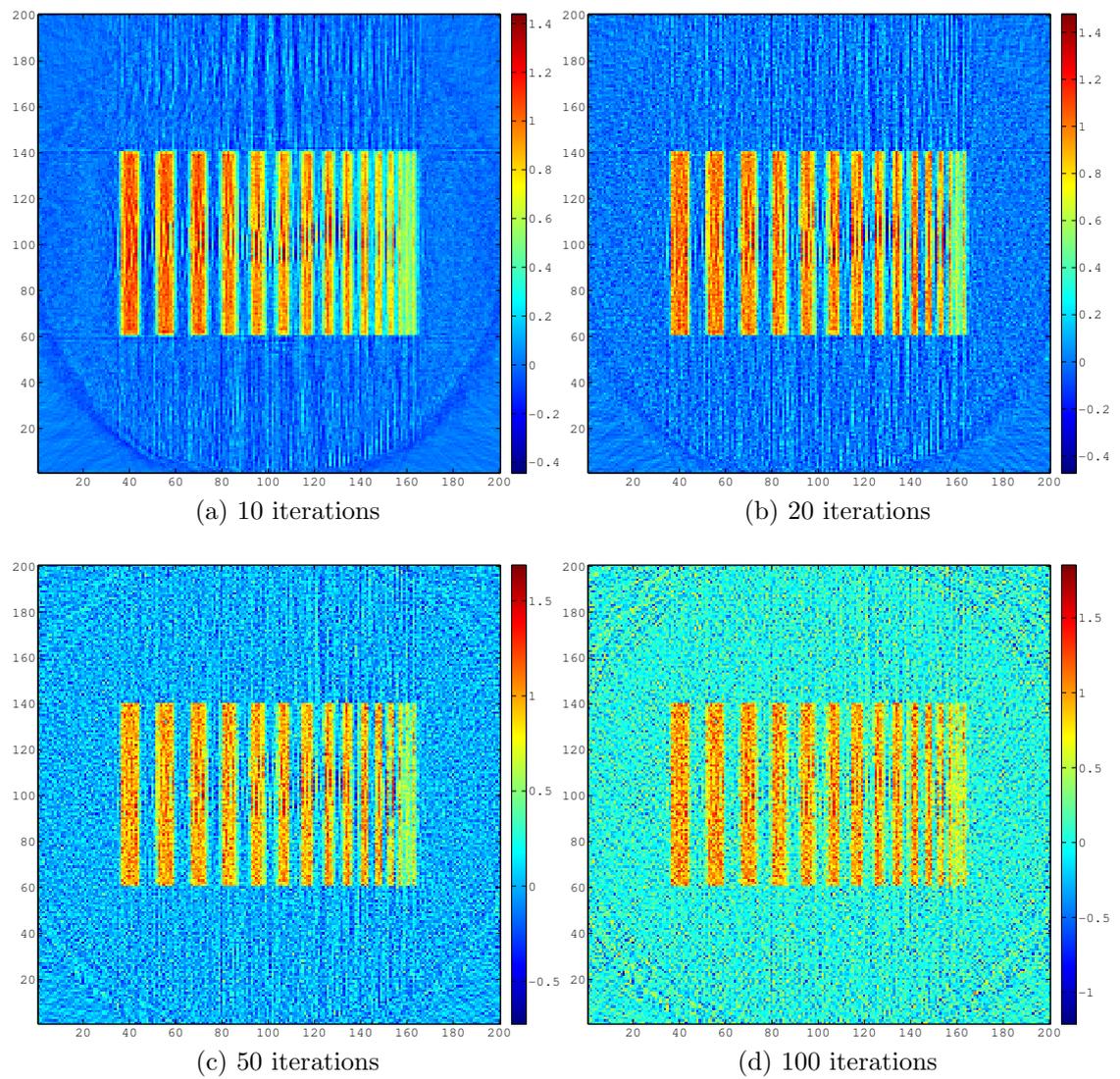


Figure 4.2: Reconstruction of resolution phantom – no regularisation

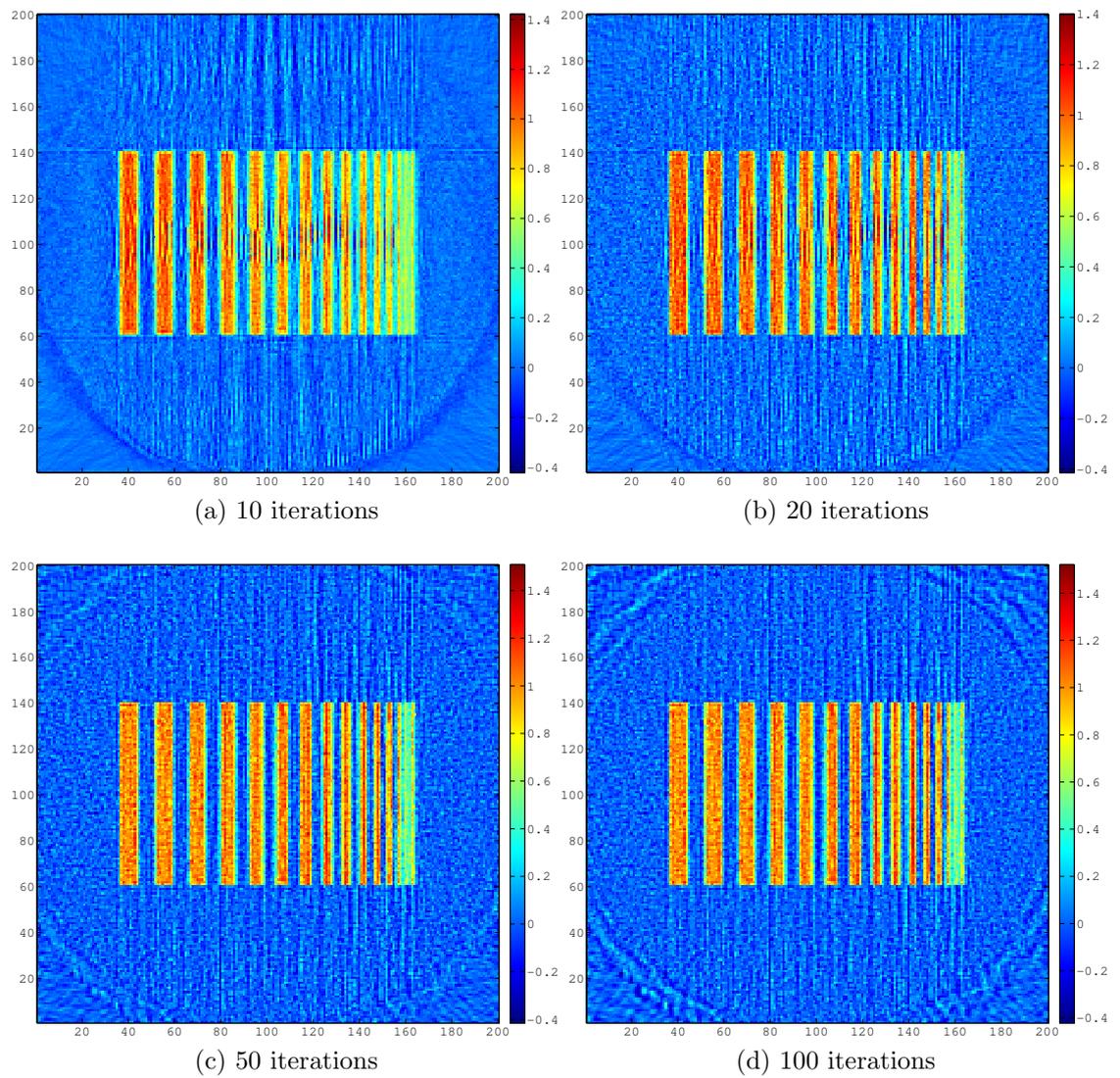


Figure 4.3: Reconstruction of resolution phantom – regularised with Laplacian,  $\alpha = 1$

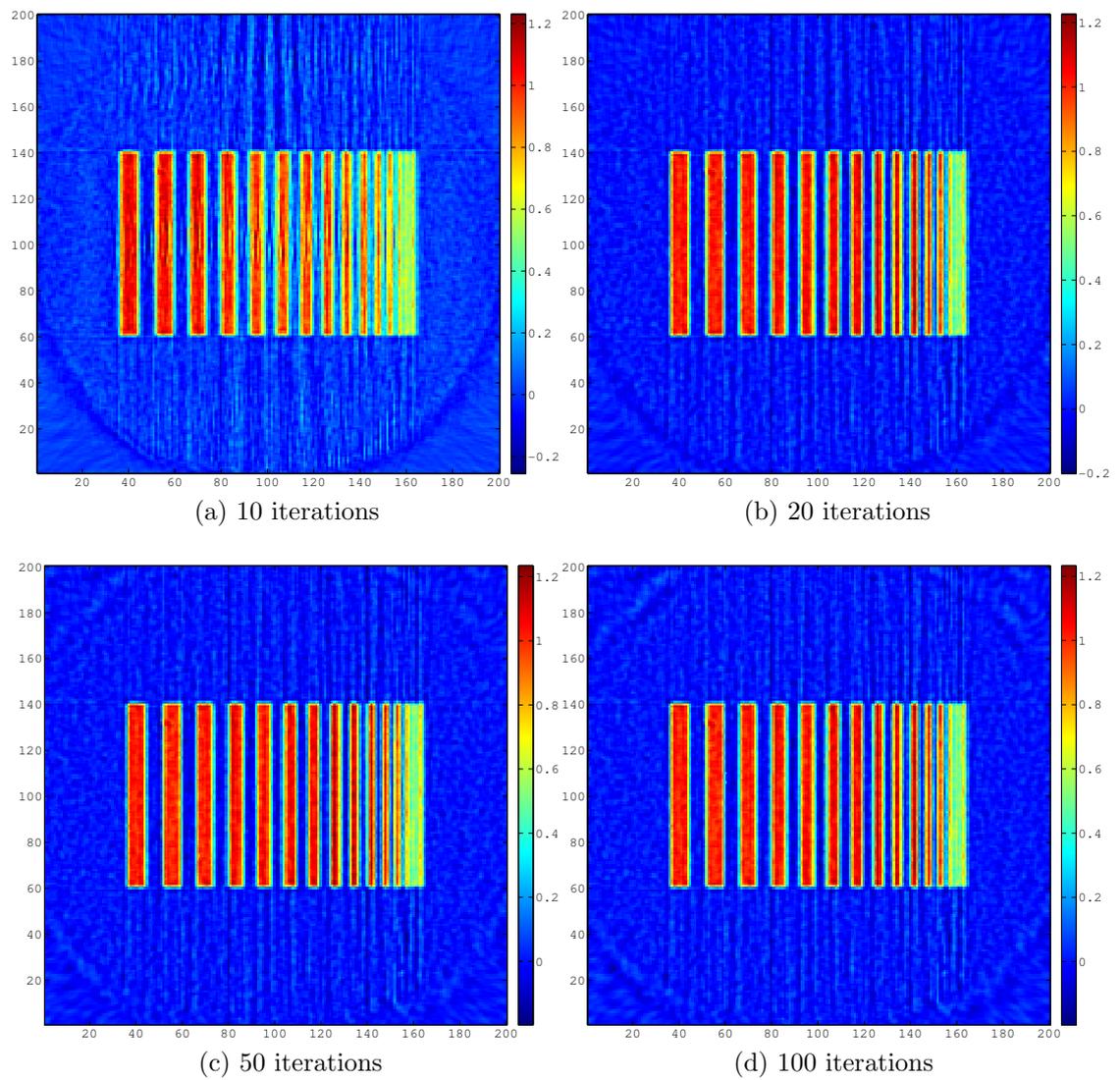


Figure 4.4: Reconstruction of resolution phantom – regularised with Laplacian,  $\alpha = 5$

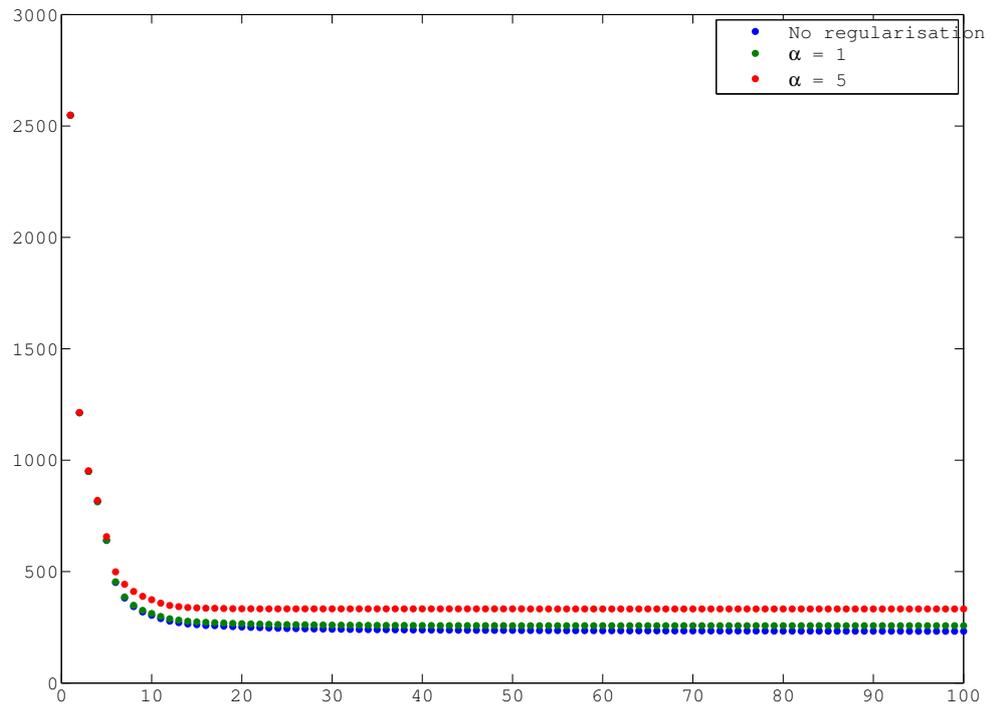


Figure 4.5: 2-norm of the residuals for the reconstructions in figures 4.2–4.4

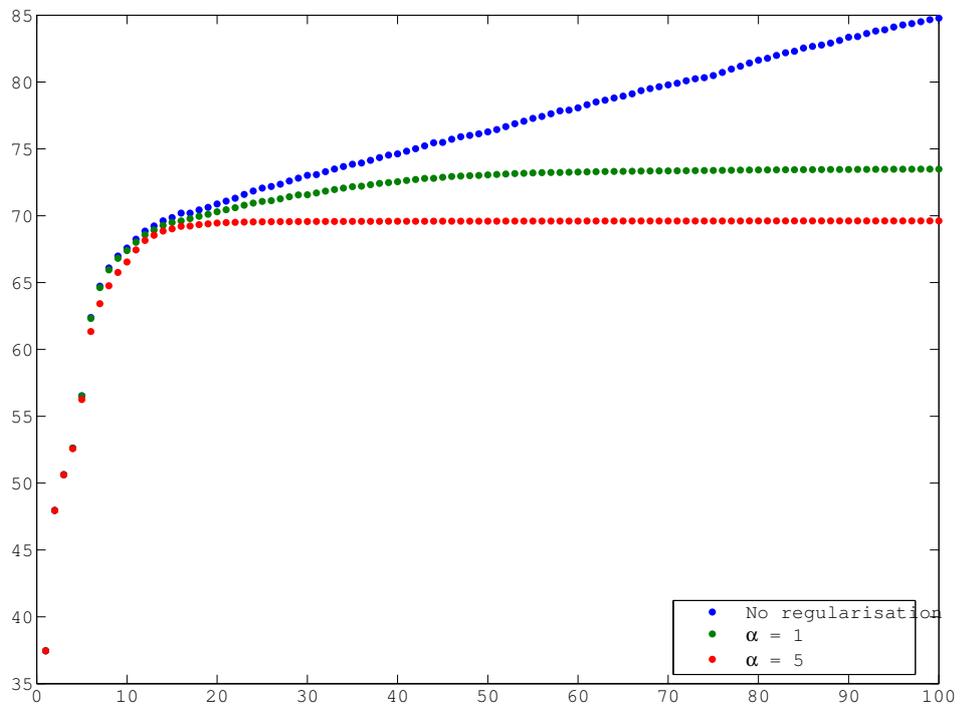


Figure 4.6: 2-norm of the solutions for the reconstructions in figures 4.2–4.4

# Chapter 5

## 3D RTT Reconstruction

We now consider the problem of 3D reconstruction from the RTT80 system.

### 5.1 The RTT80 System

RTT80 is the final production model of the machine that is expected to be used in airport baggage screening. The main purpose of the machine is to detect direct threats to aircraft, which essentially means improvised explosive devices. This would generally mean a lump, or sheet, of solid explosive material or a liquid explosive with a means of detonation connected by wires to some kind of electronic controller. Thus it is important to be able to accurately measure density, and also to resolve detailed structures which could potentially be wiring.

The geometry of the RTT80 machine is shown in figure 5.1. It consists of 768 sources arranged in 24 blocks of 32, and 8 rings of 1152 detectors arranged in 72 blocks of 16. A precisely controlled belt can be set to either a slow speed of  $250\text{mms}^{-1}$  or a fast speed of  $500\text{mms}^{-1}$ . At the slow speed, all sources are used at 15 revolutions per second, whilst at the fast speed only alternate sources are used at 30 revolutions per second. In both cases the length in the  $z$ -direction covered during one revolution is  $16.67\text{mm}$ ; this is known as the  $z$ -pitch, which will be denoted by  $\pi_z$ .

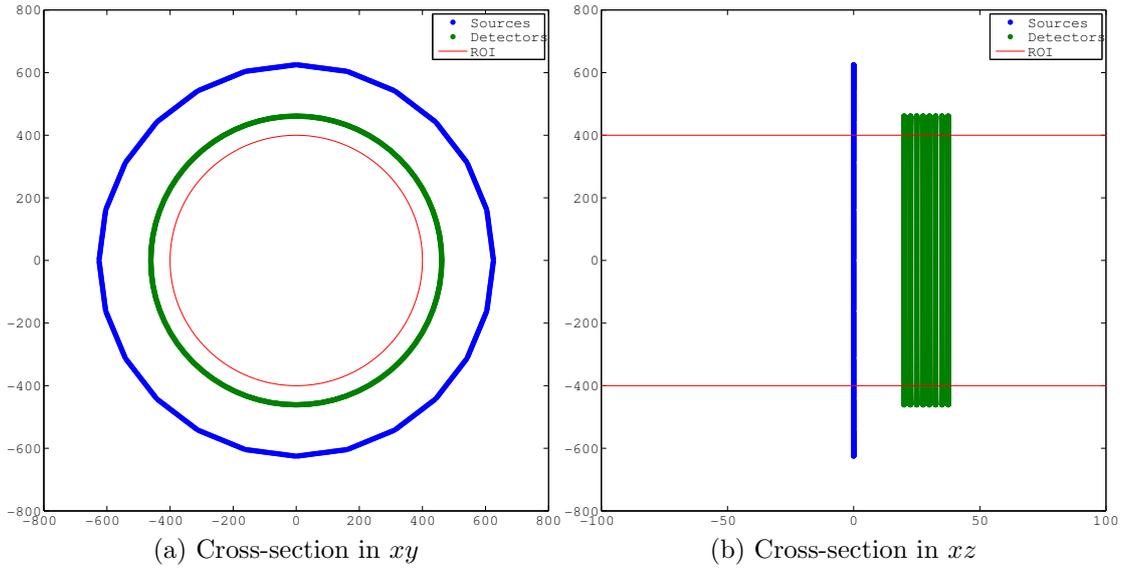


Figure 5.1: The RTT80 geometry

## 5.2 The Firing Order

As introduced in section 1.2, one of the major differences between the RTT system and conventional CT scanners is the ability to switch the sources in almost any order we desire. This presents us with the challenge of investigating the effect the firing order has on reconstruction and in some way optimising this. The problem of optimising the firing order will be treated in chapter 6; we just introduce some basic definitions here.

### 5.2.1 A Formal Definition

The firing order is defined as follows:

**Definition 5.1.** For an RTT system with  $N_S$  sources, a **firing order** of **period 1** revolution is defined as the periodic extension of the sequence

$$\dots, \phi(1), \dots, \phi(N_S), \dots, \quad (5.1)$$

determined by some function

$$\phi : \{1, \dots, N_S\} \rightarrow \{1, \dots, N_S\}. \quad (5.2)$$

This definition may be extended to cover firing orders of an arbitrary period  $R$  revolutions by associating with each revolution a revolution number  $r \pmod{R}$  and defining multiple functions  $\phi_r$  for each  $1 \leq r \leq R$ . The firing order of period  $R$  revolutions is then defined as the periodic extension of the sequence

$$\dots, \phi(1), \dots, \phi(RN_S), \dots, \quad (5.3)$$

defined by the function

$$\phi(i + rN_S) = \phi_r(i). \quad (5.4)$$

Although theoretically it is possible to have a completely aperiodic firing order, this would not be particularly useful in practice and so will not be considered here.

The firing order determines the sequence in which the physical sources in the RTT system are switched on and off; for a particular firing order  $\phi$ , as we let  $i$  run from 1 to  $RN_S$ , the sources  $s_{\phi(1)}, \dots, s_{\phi(RN_S)}$  are switched in sequence.

Although it is not strictly necessary to do so, it makes sense to consider only firing orders where the functions  $\phi_r$  are bijective (i.e. permutations of  $\{1, \dots, N_S\}$ ). This ensures all sources are used and justifies the use of the term revolution, since a full set of projections from the physical sources  $s_1, \dots, s_{N_S}$  is analogous to a complete revolution of the gantry of a conventional CT scanner. Given this restriction, and the assumption that the firing order has finite period, without loss of generality we can adopt the convention that for any firing order,  $\phi(1) = 1$ .

## 5.2.2 Rotational Invariance

An important property of the firing order is the idea of rotational invariance, defined as follows:

**Definition 5.2.** For an RTT system with  $N_S$  sources, the firing order of period  $R$  revolutions defined by the function  $\phi$  is said to be **rotationally invariant** order- $n$  if for  $1 \leq n < N_S$  and some fixed integer  $k$  we have:

$$\phi(i) \equiv (\phi(i + n) - k) \pmod{RN_S} \quad \text{for all } i = 1, \dots, RN_S. \quad (5.5)$$

Interpreting this geometrically, what this means is that from the perspective of some source  $s_i$ , if we then move to source  $s_{i+n}$ , the positions of all other sources in three-dimensional space *relative to the source we are at* do not change.

### 5.2.3 The Inverse Firing Order

A useful concept is that of the inverse firing order; for a firing order  $\phi(i)$  of period 1 revolution, this is simply defined by the inverse function  $\phi^{-1}(j)$ , if this exists, where  $j$  represents a source number. We can extend this definition to firing orders with higher period by associating a revolution number with each source as fired by the order. Then the inverse firing order for revolution  $r$  is defined by the inverse function  $\phi_r^{-1}(j_r)$ , where  $j_r$  is a source number for revolution  $r$ .

For a given source  $s_j$ , the inverse firing order gives us the position of that source within the firing order, and therefore the  $z$ -coordinate of that source relative to the  $z$ -coordinate of the first source in the order.

## 5.3 Reconstruction Algorithms

We give here an overview of reconstruction algorithms applicable to RTT80. Using the model of the divergent beam transform with sources on a cylinder, the RTT80 problem has been shown to be one of limited angle reconstruction; hence any reconstruction algorithm will be approximate.

### 5.3.1 The RayConStruct Algorithm

The algorithm currently in use on the prototype and final production versions of RTT80 is referred to in the Rapiscan literature as ‘Advanced Cone-Beam Back Projection’ or ACBP, and was developed by RayConStruct. Due to licensing issues, full details of this algorithm are unavailable; however, from the limited information we do have, we are told that it is an FDK based approximate algorithm, presumably based on some modified version of the cone-beam backprojection algorithm of Feldkamp,

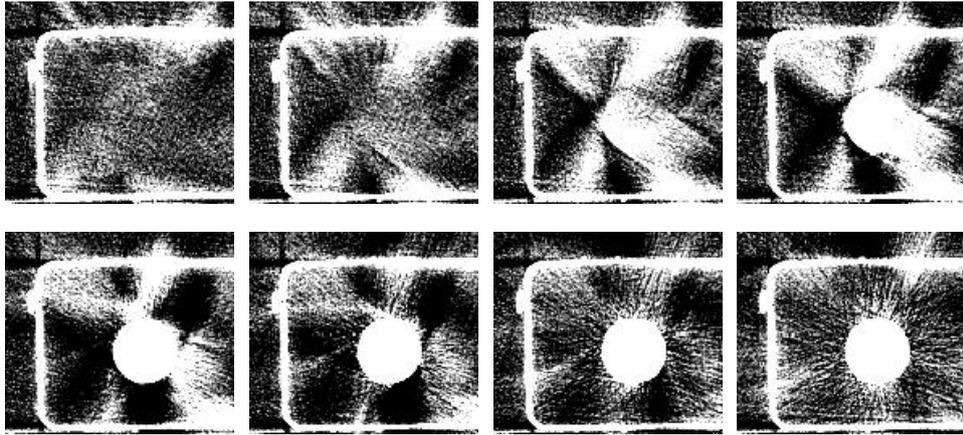


Figure 5.2: ‘Windmill’ artefact shown over 8 consecutive slices

Davis and Kress given in [10]; a good survey of such algorithms is given in [65]. The method reconstructs onto a  $1.04 \times 1.04 \times 1.04$ mm voxel grid.

This method has already been proven to be efficient enough for online reconstruction in the required time and gives acceptable image quality. However, there are some undesirable artefacts present in the reconstructed volumes; it is unknown whether these are attributable to the algorithm or firing order used, or the design of the machine itself.

The first of these is known as the ‘windmill’ artefact, and is characterised by streaks appearing in a distinctive 4-spoked pattern at high contrast interfaces in the  $z$  direction, shown in figure 5.2.<sup>1</sup> This can affect the accuracy of automatic threat detection segmentation algorithms, since streaks created by a high density object can obscure low density objects in the reconstruction. It is likely that this is a consequence of the limited angle nature of the RTT data, since interfaces in the  $z$  direction represent singularities tangent to planes in which no rays are measured. However, it is possible that the effects of this could be mitigated to an extent by the choice of firing order; the 4-spoked patterns being a consequence of the approximately 4-helical firing order.

Secondly, we have the ‘zigzag’ artefact, characterised by gaps appearing in thin sheets every unit of  $z$ -pitch, as shown in figure 5.3.<sup>2</sup> Again, this can affect the accuracy

---

<sup>1</sup>Figure courtesy of Rapiscan

<sup>2</sup>Figure courtesy of Rapiscan

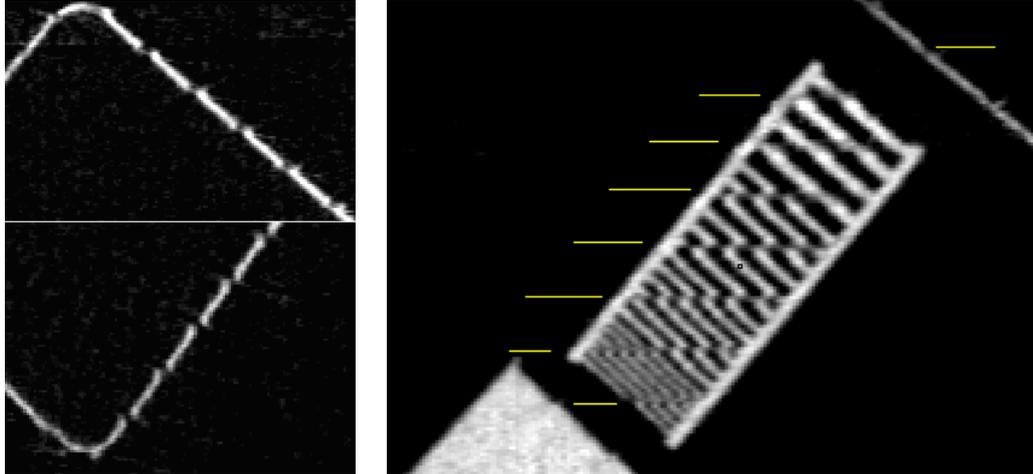


Figure 5.3: ‘Zigzag’ artefact

of automatic threat detection algorithms, and has been attributed to the fact that in the radial centre of the reconstruction volume, in a neighbourhood of the  $z$ -axis, the  $z$ -pitch is greater than the relative extent of the detectors in the  $z$  direction. It is possible though, that due to the angular distribution created by the firing order (see section 6.2.2), this could be an artefact of the specific firing order used.

### 5.3.2 Rebinning Based Methods

Rebinning based methods seek to approximate the complicated 3D reconstruction problem by a set of simple two-dimensional reconstruction problems which can be solved efficiently by inversion of the two-dimensional Radon transform. The simplest way to do this is to approximate the 3D cone-beam data by fan or parallel-beam data in a series of transaxial slices; see, for example [4] or [49]. However, this is far from optimal, especially for the RTT system with its offset detectors.

Various more advanced rebinning methods have been proposed; for conventional helical scan CT machines, rebinning to tilted planes was proposed in [26] and [27], and to general non-planar surfaces in [43].

For the RTT system, a more advanced rebinning algorithm, multi-sheet surface rebinning, has recently been proposed by Betcke in [3]. The essence of this is that for the RTT system, the optimal rebinning surface is that of a double cone. Reconstructing in two dimensions onto the superimposed double surfaces presents us with

an additional deconvolution step in the reconstruction process, increasing the computational overhead but results are promising. The method also gives us flexibility in the choice of firing order used.

### 5.3.3 Algebraic Reconstruction

Algebraic reconstruction for RTT80 can be viewed as a direct discretisation of the divergent beam transform equation with sources on a cylinder. Algebraic reconstruction has the considerable disadvantage of being much slower than the analytical methods, especially for 3D reconstruction of moderately large data sets, as we have with RTT80. It is therefore extremely unlikely that algebraic methods could be used for online RTT80 reconstruction at the current time. However, we justify its consideration for three reasons; firstly, if we can show that algebraic reconstruction can produce better reconstruction than the analytical methods but in a longer time, we may still be able to use it in airport security as an additional level of offline inspection. If a bag has been identified as a potentially serious threat and cannot be cleared without further inspection then it could be placed in a holding system while the algebraic reconstruction process is run on it, potentially identifying objects which could not be identified using the analytical methods.

Secondly, as affordable computer power increases over the next few years, we may get to a position where we are able to implement an algebraic algorithm for the online reconstruction; hence it makes sense to look into it now with the future in mind. Finally, from a purely theoretical point of view, algebraic reconstruction gives us a way to directly apply reconstruction using the new model and hopefully provide some validation of it; it also gives us no restriction on the firing order we can use.

Perhaps due to the success of the analytical methods, and their ease of efficient implementation, algebraic methods have not seen much application to three-dimensional cone-beam CT reconstruction.<sup>3</sup> Studies we have found concentrate mainly on implementations of Kaczmarz and Landweber type algorithms and the choice of relaxation

---

<sup>3</sup>See [51] for a discussion on why the FDK based analytical methods are still in widespread use in commercial CT scanners.

matrix or voxel basis functions; see, for example [5], [6] and [45]. Much work has been done on improving the efficiency of implementations of these methods; see for example [46] and [44]; and more recently in [70] and [40], through the use of commodity graphics hardware.

We propose a different approach – to use the CGLS algorithm for reconstruction and assess its effectiveness for the RTT case. The CGLS algorithm should in theory converge more quickly than the traditional Landweber based algorithms; studies such as [52] indicate that the CGLS algorithm may indeed be a better choice for CT problems. If we can show that the method is effective then it is possible that more efficient implementations could also be developed for CGLS.

## 5.4 Applying Algebraic Reconstruction to RTT80

Given the potential advantages of algebraic reconstruction over the known analytical methods, it seems natural to expect that we should be able to produce high quality images from RTT80 data using algebraic reconstruction. We now consider the practical implications of doing this.

The main problem we face is that for a typical object scanned in RTT80, the matrix  $A$  will have many millions of rows and many millions of columns. Even using a sparse format, this is far too big to be stored in memory. For this reason it is usually the case that the matrix coefficients are worked out ‘on the fly’ and not stored in memory at all. Therefore, in order to perform reconstruction at an acceptable speed, a simplified model of the projection process such as that introduced in section 4.1 is used.

By exploiting structure in the matrix  $A$ , we now present an efficient way of storing the coefficients, which could in the future enable us to use a more accurate, computationally expensive model of the projection process. It is also possible that this structure may allow us to improve the efficiency of the calculations if we were to compute the coefficients on the fly.

### 5.4.1 The Structure of $A$

Let  $V \subset \mathbb{R}^3$  represent the reconstruction volume, discretised into voxels of size  $v_x$ ,  $v_y$  and  $v_z$  in the  $x$ ,  $y$  and  $z$  directions respectively, and let  $V_x$ ,  $V_y$  and  $V_z$  be the total number of voxels in each direction. We think of the volume as being sliced in the  $z$  direction into slices of thickness 1 voxel. As with the existing RTT algorithm, we would like to set up the reconstruction grid so that there are an integer number of slices in the  $z$  direction for every unit of  $z$ -pitch. Letting  $N_{SR}$  represent the number of slices per revolution this gives:

$$v_z = \frac{\pi_z}{N_{SR}}. \quad (5.6)$$

The rows of  $A$  represent rays and are ordered first by revolution, then by source, then by detector ring and finally by detector. The columns of  $A$  represent voxels, ordered by  $z$ , then  $y$ , then  $x$ . Since we chose an integer number of slices per revolution, it should be clear that for the section of reconstruction volume representing one revolution, the ray paths through this will be the same for each revolution. Therefore  $A$  has the following block Toeplitz structure

$$A = \begin{bmatrix} A_1 & 0 & 0 & \dots & 0 & 0 \\ A_2 & A_1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ A_m & A_{m-1} & A_{m-2} & \dots & 0 & 0 \\ 0 & A_m & A_{m-1} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & A_m & A_{m-1} \\ 0 & 0 & 0 & \dots & 0 & A_m \end{bmatrix}, \quad (5.7)$$

where each of the blocks  $A_i$  is of size  $N_S N_R N_D$  rows by  $V_x V_y N_{SR}$  columns, representing the coefficients for all rays from one complete revolution intersecting a block of  $N_{SR}$  slices. One block-column of  $A$  therefore represents all rays intersecting  $N_{SR}$  slices of the reconstruction volume; we let with  $M$  equal to the number of blocks, and hence revolutions, needed to completely specify all such rays.

This immediately gives us a more economical way of storing  $A$ ; we just store each of the blocks  $A_1, \dots, A_M$  and apply multiplication by  $A$  and its transpose using the stored blocks. However, to reconstruct with  $N_{SR} = 24$  and  $v_x = v_y = 1\text{mm}$ , even storing  $A$  like this will take approximately 40GB of memory. At the current time of writing, given additional memory overheads in storing the reconstruction volume and applying the algorithm, this would require the use of very expensive computer hardware. We will now show that through a careful choice of the firing order, the memory overhead can be cut dramatically and a full resolution reconstruction can be achieved using fairly modest consumer hardware.

### 5.4.2 Reducing the Block Size

Looking at figure 5.4, we see that the geometry of the RTT80 machine is actually rotationally symmetrical. Each block of 32 sources is lined up with 3 blocks of 16 detectors, so from the point of view of a source, if we start at some source  $s_i$  and then move to another in a different block, but in the same position within the block, the geometry of the machine will look the same, but rotated through some angle  $\theta$ . If we denote by  $b(s_i)$  the position of a source within the block then we have  $b(s_i) = i \bmod 32$ , since there are 32 sources in each block.

Recall that an order-1 rotationally invariant firing order is one of the form  $\phi(i) = ki \bmod N_S + 1$  for some integer  $k$  coprime to  $N_S$ . So for RTT80 this means orders of the form

$$\phi(n) = (ki \bmod 768) + 1, \quad (5.8)$$

where  $k$  is coprime to 768. Note that  $k$  being coprime to 768 also implies that  $k$  is coprime to 32; hence the sequence  $(b(s_i))$  repeats itself every 32 sources. Taking the symmetry of the system into account, this means that if we divide the firing order up into subsequences of length 32, then from the point of view of the first source in each subsequence, the geometry of the system looks the same but rotated by some angle  $\theta$ .

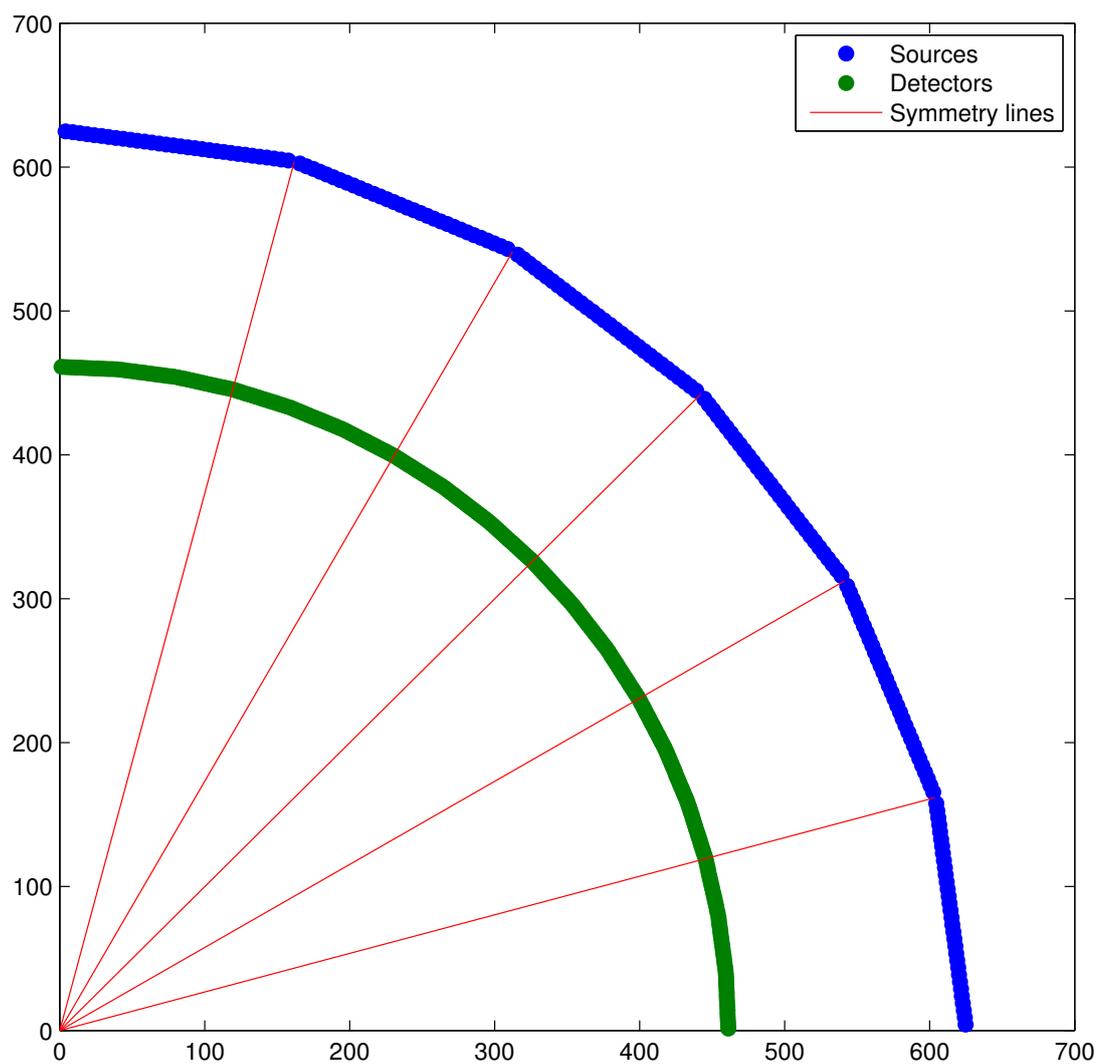


Figure 5.4: Enlarged cross-section of RTT80 geometry showing lines of rotational symmetry

Now if we slice the reconstruction volume up in the  $z$  direction so that the thickness of each slice contains the  $z$  values of 32 sources then each slice contains exactly the same ray paths, but rotated through  $\theta$ . This gives a maximum of 24 slices per revolution with a slice thickness  $v_z$  of 0.69mm. We can also set  $v_z$  to include an integer multiple of 32 sources in the slice thickness, as long as this multiple divides 24, giving 12, 8, 6, 4, 3, 2 or 1 slices per revolution. Letting  $m_{\text{block}}$  and  $n_{\text{block}}$  represent the number of rows and columns respectively in each block, this gives us:

$$m_{\text{block}} = \frac{N_S N_R N_D}{N_{SR}}; \quad n_{\text{block}} = V_x V_y N_{SR}. \quad (5.9)$$

The angle subtended by each block of sources is  $15^\circ$ , therefore for a firing order of the form (5.8) with  $k$  coprime to 768, the angle  $\theta$ , in degrees, is given by

$$\theta(k) = 15 \cdot \frac{24}{N_{SR}} (k \bmod N_{SR}). \quad (5.10)$$

Now, let  $S_z \subseteq V$  represent a slice, where  $z = 1, \dots, V_z$  is the slice number. Then we can define the transformation  $\Theta(S_z)$  representing the rotation of each slice as

$$\Theta(S_z) = \begin{bmatrix} \cos(\theta(k)(z \bmod 24)) & \sin(\theta(k)(z \bmod 24)) & 0 \\ -\sin(\theta(k)(z \bmod 24)) & \cos(\theta(k)(z \bmod 24)) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (5.11)$$

Now we construct the blocks of the matrix  $A$  using only the coefficients for the slice  $S_1$ . Taking  $N_{SR} = 24$  and  $v_x = v_y = 1\text{mm}$ , the storage requirements for storing this much simplified version of  $A$  are roughly 2GB; obviously much less than storing coefficients for every slice. Solving the system of equations represented by this matrix will reconstruct the slices of the volume transformed by  $\Theta(S_z)$ . It is a simple process to apply the inverse transformation and recover the original volume  $V$ .

We can generalise the block storage of  $A$  to firing orders with higher order rotational invariance. However, the block size may need to be larger; if we use the maximum number of slices per revolution, we will need to store coefficients for a number of slices equal to the order of the rotational invariance. For example, the rotationally invariant extension of the RTT firing order is the 4-helix, defined with a

period of 4 revolutions. Taking  $N_{SR} = 24$ , we can represent  $A$  by a block of coefficients for 4 slices; each block of 4 slices in the reconstructed volume will be rotated by 15 degrees. Letting  $O(\phi)$  be the order of the rotational invariance, in general we have

$$m_{\text{block}} = \frac{N_S N_R N_D O(\phi)}{N_{SR}}; \quad n_{\text{block}} = V_x V_y N_{SR} O(\phi). \quad (5.12)$$

### 5.4.3 Choosing an Appropriate Resolution

Clearly, for a given reconstruction volume  $V$ , the number of voxels in the discretisation of  $V$  gives us the number of columns in the matrix  $A$ , and hence the number of unknowns in the system of equations represented by  $A$ . For a given set of data, the number of rows of  $A$  cannot change, so changing the resolution of the discretisation will affect whether the system is over or under-determined. We therefore wish to choose values for  $v_x$ ,  $v_y$  and  $v_z$  to give as fine a discretisation as possible without making the system of equations represented by  $A$  underdetermined.

The number of rays intersecting the whole reconstruction volume can be expressed as a simple linear equation as follows. Let the total number of rays be  $r$ ; then thinking of the volume as a stack of slices, the smallest possible volume is just a single slice; we let the number of rays intersecting this be  $b$ . Then each slice we add subsequently adds a constant number of rays to this; let this constant number be represented by  $a$ . Then the total number of rays is given by

$$r = aV_z + b. \quad (5.13)$$

The size of the reconstruction volume is set to  $800 \times 800 \times v_z V_z$  mm. We let  $v_{xy} = v_x = v_y$  and, as noted in section 5.4.2, we choose  $v_z$  according to (5.6) by setting  $N_{SR}$  equal to some factor of 24. Some values of  $a$ ,  $b$  and  $V_x V_y$ , the number of voxels per slice are summarised in table 5.1. The value of  $b$  changes slightly depending on the firing order, therefore the values given are approximate.

Considering the limit as the number of slices tends to infinity, in order to ensure the system will not be underdetermined for any value of  $V_z$  we just consider the

$N_{SR}$	$v_z$	$v_{xy}$	$a$	$b$	$V_x V_y$
24	0.69	1.0	167,936	$\sim 5,300,000$	640,000
12	1.39	1.0	335,872	$\sim 5,300,000$	640,000
12	1.39	1.39	335,872	$\sim 5,300,000$	331,776
12	1.39	2.0	335,872	$\sim 5,300,000$	160,000

Table 5.1: Values of  $a$ ,  $b$  and  $V_x V_y$  for different resolutions

relationship between  $b$  and  $V_x V_y$ . We therefore choose a resolution of  $1.39 \times 1.39 \times 1.39$ mm cubic voxels.

#### 5.4.4 Calculating the Coefficients

In order to calculate the coefficients for the required blocks of  $A$ , we must make sure that we use enough complete revolutions to cover all the rays passing through the slice or slices covered by the blocks. With  $N_{SR} = 12$  and up to 2 slices per block it is sufficient to calculate  $A$  over 3 full revolutions, giving source  $z$  coordinates in the range 0–50mm. We position the reconstruction slice at  $z = 50$ mm and set the  $x$  and  $y$  ranges to  $-400 - 400$ mm to completely cover the 80cm tunnel. If more slices are needed then we simply increase the number of revolutions we calculate over.

The coefficients  $a_{ij}$  are then found as in section 4.3.1. The resulting sparse matrix contains all blocks  $A_i$  needed to completely recover  $A$  in the reconstruction process. It will also contain some zero blocks at the top which can be deleted. The number of non-zero blocks needed to represent  $A$  is denoted as above by  $M$ .

An important point to note is that in order for the block process to work at different rotation angles, the ordering of the rows (i.e. rays) of the computed matrix, and the projection data itself, by detector number must be *relative to the number of the first active detector for a particular source*.

#### 5.4.5 Implementing the Reconstruction

The discretised reconstruction volume is represented by the vector  $\mathbf{x}$  and the projection data by the vector  $\mathbf{b}$ . Elements of  $\mathbf{x}$  are ordered first by growing  $z$ -coordinate (or equivalently slice number), then by growing  $y$ -coordinate and finally  $x$ -coordinate, so

$\mathbf{x}$  can be divided into  $V_z$  sub-vectors of length  $V_x V_y$ , each representing a slice. The elements of  $\mathbf{b}$  are ordered in the same order as the rows of  $A$  as described in section 5.4.4.

Applying the CGLS algorithm (or indeed many of the other iterative methods) involves calculating matrix-vector products of  $A$  and  $A^T$  with vectors the same size as  $\mathbf{x}$  and  $\mathbf{b}$  respectively; storing  $A$  in block form we obviously cannot evaluate these directly. The matrix  $A$  has the following Kronecker product decomposition

$$A = J_1 \otimes A_1 + J_2 \otimes A_2 + \dots + J_M \otimes A_M, \quad (5.14)$$

where  $M$  is as defined above and the  $J_i$  are defined as follows

$$J_i = \begin{bmatrix} 0_{(i-1) \times N} \\ I_N \\ 0_{(M-i) \times N} \end{bmatrix}, \quad i = 1, \dots, M, \quad (5.15)$$

where  $N$  is the total number of blocks needed to represent the whole reconstruction volume  $V$ ,  $I_n$  is the  $n \times n$  identity matrix and  $0_{m \times n}$  is the  $m \times n$  all zero matrix. This suggests the following process.

We reshape the vector  $\mathbf{x}$ , representing the reconstruction volume, into the matrix  $X$ , whose columns are the blocks of  $\mathbf{x}$  containing the number of slices represented by the blocks of  $A$ . The number of columns of  $X$  is equal to  $N$ ; for the order-1 rotationally invariant firing orders, this is equal to the number of slices in the reconstruction volume. The product  $A\mathbf{x}$  can then be expressed in terms of the sum of products of the blocks  $A_i$  with  $X$  as follows:

$$A\mathbf{x} = \text{vec} \left\{ \sum_{i=1}^M \begin{bmatrix} 0_{m_{\text{block}} \times (i-1)} & A_i X_{[:,(M-i+1):(N-i+1)]} & 0_{m_{\text{block}} \times (M-i)} \end{bmatrix} \right\}, \quad (5.16)$$

where  $\text{vec}$  denotes the *vectorisation* of the matrix on the right hand side, and the subscript on  $X$  defines sub-matrices of  $X$  using MATLAB colon notation. A similar decomposition exists for calculating  $A^T \mathbf{b}$ .

These processes have been implemented as openmp parallelised C `.mex` routines which are called from a modified version of `cgl_s.m` from Hansen's Regularisation Tools; for information about openmp see [50]. Due to the way MATLAB handles

sparse arrays it is quicker to store both the original blocks of  $A$  and transposed versions for use in these routines, since indexing down columns is much quicker than indexing across rows.

### 5.4.6 Regularisation

Explicit Tikhonov regularisation can be applied in the same way as in section 4.3.3. However, in order to apply this, the transformation of (5.11) and its inverse would have to be applied at every iteration. For this reason, further regularisation is not implemented in the current method; it is hoped that in future this will be able to be applied.

# Chapter 6

## Optimising the Firing Order

We now consider the problem of finding a firing order that is in some sense optimal. Our interpretation and ultimate choice of the firing order is determined to a large extent by the mathematical model we use for the system and the resulting reconstruction algorithm. Therefore the optimisation process must take this into account. Clearly if we are modelling the system in the conventional CT sense where it is assumed that a source follows a given trajectory in  $\mathbb{R}^3$  then it makes sense to think of the firing order as defining such a trajectory or trajectories. Thus in this case we would be inclined to look for firing orders where the structure of such trajectories is clear.

However, using the model introduced in section 2.5 it no longer makes sense to do this. Whether we are reconstructing analytically or algebraically using this model, we now view the firing order as defining a sampling pattern on the surface of a cylinder. Therefore we may now consider firing orders which do not define a clear trajectory, but instead define a sampling pattern on the surface of a cylinder.

Regardless of the model used, it is also interesting to see how the firing order affects the geometry of the system. The firing order has a significant effect on the distribution of x-ray energy within the reconstruction volume and also on the distribution of the illumination angles of the rays intersecting particular regions of the volume.

As remarked in section 5.2, in all of what follows we regard a firing order as being defined over  $R$  revolutions by some bijective functions  $\phi_r$ , for  $1 \leq r \leq R$ . In order

to simplify the problem, we also only consider firing orders of period 1, or structured such that the functions  $\phi_r$  for  $r \geq 2$  are completely defined by  $\phi_1$ ; thus we have a finite number of firing orders and the problem is essentially combinatorial in nature.

## 6.1 Types of Firing Order

Considering all the possible period 1 firing orders would give us  $768!$  possible functions  $\phi$ , which is clearly too many to consider individually. In any case, it seems sensible to consider mainly firing orders with some kind of structure which fits in with the mathematical model we are using. We therefore classify firing orders into types with certain properties; the key property we use is that of rotational invariance.

This work was started before the new theoretical model of sources on a cylinder was discovered; hence certain firing orders are sometimes still referred to by the type of trajectory they can be viewed as approximating. However, it is interesting to compare properties of firing orders optimised for both models in order to provide validation of the theory.

### 6.1.1 Rotationally Invariant Firing Orders

The rotationally invariant firing orders can be split into two sub-types as follows:

#### Order-1 Rotational Invariance

Of particular importance will be firing orders with order-1 rotational invariance; these are defined over one revolution by functions  $\phi$  of the form:

$$\phi(i) = (k(i - 1) \bmod N_S) + 1, \quad (6.1)$$

where  $k$  is some integer coprime to  $N_S$ . Because  $k$  is coprime to  $N_S$ , these firing orders are always of period 1 revolution and have some useful properties which will be explained in future sections. As we change the value of  $k$  we find that we get cases where the firing order can be viewed as defining a clear trajectory and others where the sampling pattern is much more even.

A special case of the order-1 rotationally invariant firing order is the **sequential** firing order. This is the ‘obvious choice’ of firing order in the sense of conventional CT; it is the period 1 firing order defined simply by the identity mapping  $\phi(i) = i$ , giving a classical single helical source trajectory.

For an order-1 rotationally invariant firing order as defined in (6.1) we note that the inverse firing order is defined in the same way as follows:

$$\phi^{-1}(j) = (k^{-1}(j - 1) \bmod N_S) + 1, \quad (6.2)$$

where  $k^{-1}$  is the inverse of  $k$  modulo  $N_S$ .

For RTT80 with 768 sources, we find a total of 256 order-1 rotationally invariant firing orders. However,  $k = 1, \dots, 383$  can be viewed as equivalent to  $k = 385, \dots, 767$ ; hence there are 128 possibilities for firing orders with order-1 rotational invariance.

### Higher Order Rotational Invariance

Higher order rotationally invariant firing orders are necessarily defined over periods of greater than one revolution, and can be viewed as a generalisation of the order-1 case above, where we choose the integer  $k$  such that  $\gcd(k, N_S) > 1$ . If we let  $m = \gcd(k, N_S)$  then the sequence created by (6.1) repeats every  $N_S/m$  sources. In order to avoid this we therefore need to introduce a second increment  $q$ , coprime to  $k$ , that gets added every  $N_S/m$  sources. If we let  $q = 1$  this can be expressed by functions  $\phi_r$ , for  $1 \leq r \leq N_S/m$ , defined as follows

$$\phi_r(i) = \left[ (k(i - 1) + m(r - 1) + \lfloor ((i - 1)m/N_S) \rfloor) \bmod N_S \right] + 1, \quad (6.3)$$

where  $\lfloor \cdot \rfloor$  is the round towards negative infinity operator. Such a firing order is often referred to as a multi-helix, since it can be viewed as defining helical trajectories for multiple sources.

## 6.1.2 Non-Rotationally Invariant Firing Orders

### The RTT80 Firing Order

The firing order defined in the RTT80 specification is referred to in the Rapiscan literature as 4-helical but is in fact only the first revolution of the 4-helix firing order on 768 sources. As such it lacks rotational invariance, but can be thought of as being approximately order-4 rotationally invariant, since at the belt speed of the RTT machine, the maximum  $z$ -axis error relative to the 4-helix order is approximately 0.065mm (see section 7.2.4).

### Random Firing Orders

Although theoretically we could define a totally random firing order, this would not be of much practical use. Therefore, for RTT80, we think of a random firing order as a period 1 revolution firing order defined by a random permutation of the integers 1–768. This can be thought of as a random sampling of points on a cylinder and gives a good point of reference for the new theory.

### Some Other Firing Orders

Several other structured firing order schemes were devised during the initial investigations; these included discrete approximations to saddle trajectories, counter-rotating helices and the so-called ‘inverse binary’ firing order where, for a theoretical RTT system with  $N_S = 2^n$  for some  $n \in \mathbb{Z}$ , sources enumerated in binary are switched according to the sequence created by reversing the order of the bits and counting in sequence.

## 6.2 Heuristic Arguments for the Choice of Firing Order

### 6.2.1 An Even Sampling Lattice

Using the new continuum model for the RTT system introduced in section 2.5, the firing order is viewed as creating a pattern of sampling points on the surface of a cylinder. In reality, due to the construction of the RTT80 machine this is actually a 24 sided prism, which is viewed as approximating a cylinder.

For a given belt speed and source firing rate of the RTT80 machine, for some fixed length in the  $z$  direction we will have the same number of sources regardless of our choice of firing order. Therefore, the number of sampling points over this length cannot change, only the distribution of them on the surface. It therefore seems logical to expect that the distribution of the source points on the surface of the prism should be as even as possible.

The most even distribution of discrete points on a surface is given by a regular equilateral triangle lattice. Ignoring the effect of the gaps created between neighbouring source segments, by considering order-1 rotationally invariant firing orders we can achieve a distribution very close to this ideal over the flat prism surfaces.

Consider 3 neighbouring sources  $s_i, s_{i+1}, s_{i+2}$  in the same source block; the distance between these within the block is 5mm. Now consider the triangle formed by the  $z$  positions  $z_i, z_{i+1}, z_{i+2}$  of these sources. For an order-1 rotationally invariant firing order defined as in (6.1), the distance in the  $z$  direction between neighbouring sources is given by the inverse firing order and so taking  $z_i = 0$  we get  $z_{i+1} = (k^{-1}\pi_z)/768$ , where  $k^{-1}$  is the inverse of  $k$  modulo 768. To construct the triangle, for  $k^{-1} > 384$ , we take  $z_{i+2}$  in the same revolution, and for  $k^{-1} < 384$ , we take  $z_{i+2}$  in the previous revolution, giving  $z_{i+2} = \pi_z(k/384 - 1)$ , as shown in figure 6.1.

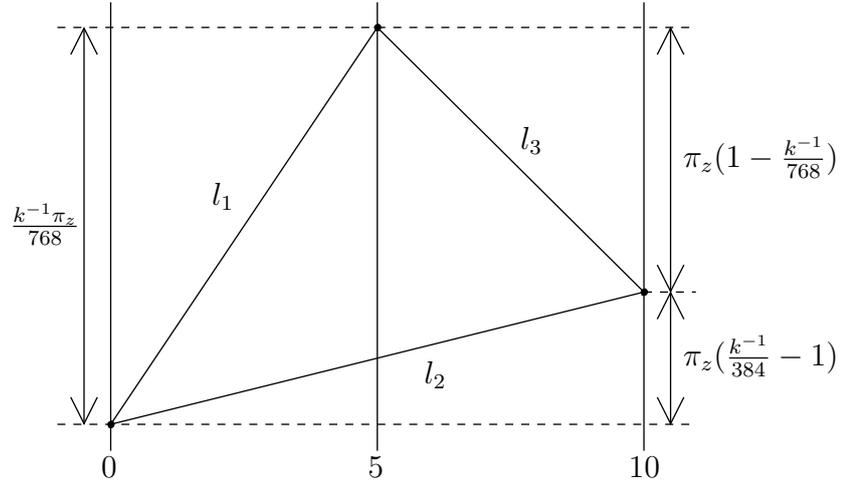


Figure 6.1: Constructing an approximately equilateral triangle lattice

We get the following expressions for the lengths  $l_1, l_2, l_3$  of the sides of the triangle:

$$l_1^2 = 25 + \left(\frac{k\pi_z}{768}\right)^2, \quad (6.4)$$

$$l_2^2 = 100 + \pi_z^2 \left(\frac{k}{384} - 1\right)^2, \quad (6.5)$$

$$l_3^2 = 25 + \pi_z^2 \left(1 - \frac{k}{768}\right)^2. \quad (6.6)$$

Plotting the standard deviation of  $l_1, l_2, l_3$  for each value of  $k$  shows a minimum at  $k = 384$ , which would give side lengths  $l_1 = 9.72, l_2 = 10, l_3 = 9.72$ , so almost equilateral. Therefore, we wish to choose  $k$  such that its inverse modulo 768 is close to 384.

### 6.2.2 Even Angular Distribution

Since with the RTT80 machine the angular extent in the  $z$  direction is limited, the angles we refer to here are the angles of projections of the rays onto transaxial planes. The initial approach to the RTT80 reconstruction problem was to look at rebinning type reconstruction algorithms, which perform approximate 3D reconstruction by breaking the problem down into a series of 2D reconstruction problems. In this case, in order to get an even sampling of rays on the 2D surfaces, we should attempt to keep the distribution of the angles of rays intersecting all regions of the volume as even as possible. Indeed, in Betcke's work on multi-sheet surface rebinning, a regular

sampling of the volume is important for stability of the deconvolution process (see [3]).

It also makes sense to think of the distribution of angles in the context of direct inversion and algebraic reconstruction. Although the RTT problem is already one of limited angles, if the set of possible known angles is  $\Omega$ , as in section 2.5.4, then the distribution of ray angles in certain regions of the reconstruction volume affects the sampling of  $\Omega$  in those regions. If there are regions in the volume where the distribution of illumination angle is uneven, resulting in few rays intersecting the region from some range of angles, then we effectively only have Radon data for some subset of  $\Omega$  within that region. This would lead to limited angle type artefacts within the region; hence, again, we would like to keep the distribution of angles as even as possible.

Considering a rotationally invariant firing order with increment  $k$ , in order to achieve an even distribution of angles, it seems intuitive that we should choose  $k$  so that, over the distance in the  $z$  direction of the size of one voxel, there is as wide a distribution of angles as possible. We actually find that for most values of  $k$  coprime to 768 the distribution of angles is good in this sense; it is easier to consider those values of  $k$  where the distribution is narrower. This essentially happens for values of  $k$  near to the more extreme factors of 768, such as  $k = 1, 5, 193, 257$  or  $383$ .

### 6.2.3 Firing Orders to Compare

Our reference firing orders will be the sequential and RTT80 orders, plus a random permutation over 1 revolution. Based on the arguments of the previous two sections, we compare these against the order-1 rotationally invariant firing order for  $k = 35$ . This value of  $k$  is close to optimal in the sense of providing an even sampling of the surface of the cylinder, and is also good in the sense of providing an even distribution of illumination angles. For convenience, we abbreviate these 4 firing orders by sequential, RTT, random and  $k = 35$ .

## 6.3 Visualising the Firing Order

Before making numerical comparisons it is enlightening to visualise certain aspects of the firing order.

### 6.3.1 The Sampling Pattern

We can view the distribution of the sampling points by simply plotting them in three dimensions, as seen in figure 2.3. However, we can actually view the sampling pattern more clearly by plotting just a small section of the surface in two dimensions; in effect, what we are doing is cutting the prism open down the side and flattening it out. We examine a section of the surface 1 source block wide and over 10 revolutions in the  $z$  direction. The results of this are shown in figure 6.2.

We see that the sequential and RTT firing orders both sample along definite lines, whereas the  $k = 35$  firing order gives a more even, regular sampling pattern.

### 6.3.2 The Ray Density

It is useful to look at the distribution of the rays within the reconstruction volume, since this will show if there are any ‘holes’ in the illumination pattern which could potentially cause a nullspace, for a high enough belt speed. We can achieve this by summing the columns of the reconstruction matrix  $A$ , giving the total length of intersection of the idealised rays through each voxel. This quantity will be referred to as the ray density, and can be viewed as an image. Results for a single slice through the whole  $800 \times 800\text{mm}$  reconstruction volume at  $1.4 \times 1.4 \times 1.4\text{mm}$  resolution are shown in figure 6.3. For the order-1 rotationally invariant firing orders all other slices can be obtained by simple rotations. For the other orders, the single slice is representative of the volume as a whole.

We see that for the sequential firing order, there are regions of the slice with very high and very low ray density; it is reasonable to expect that reconstruction within these regions might not be as good quality as reconstruction in the higher density regions. Indeed, if the belt speed were increased, these regions could actually become

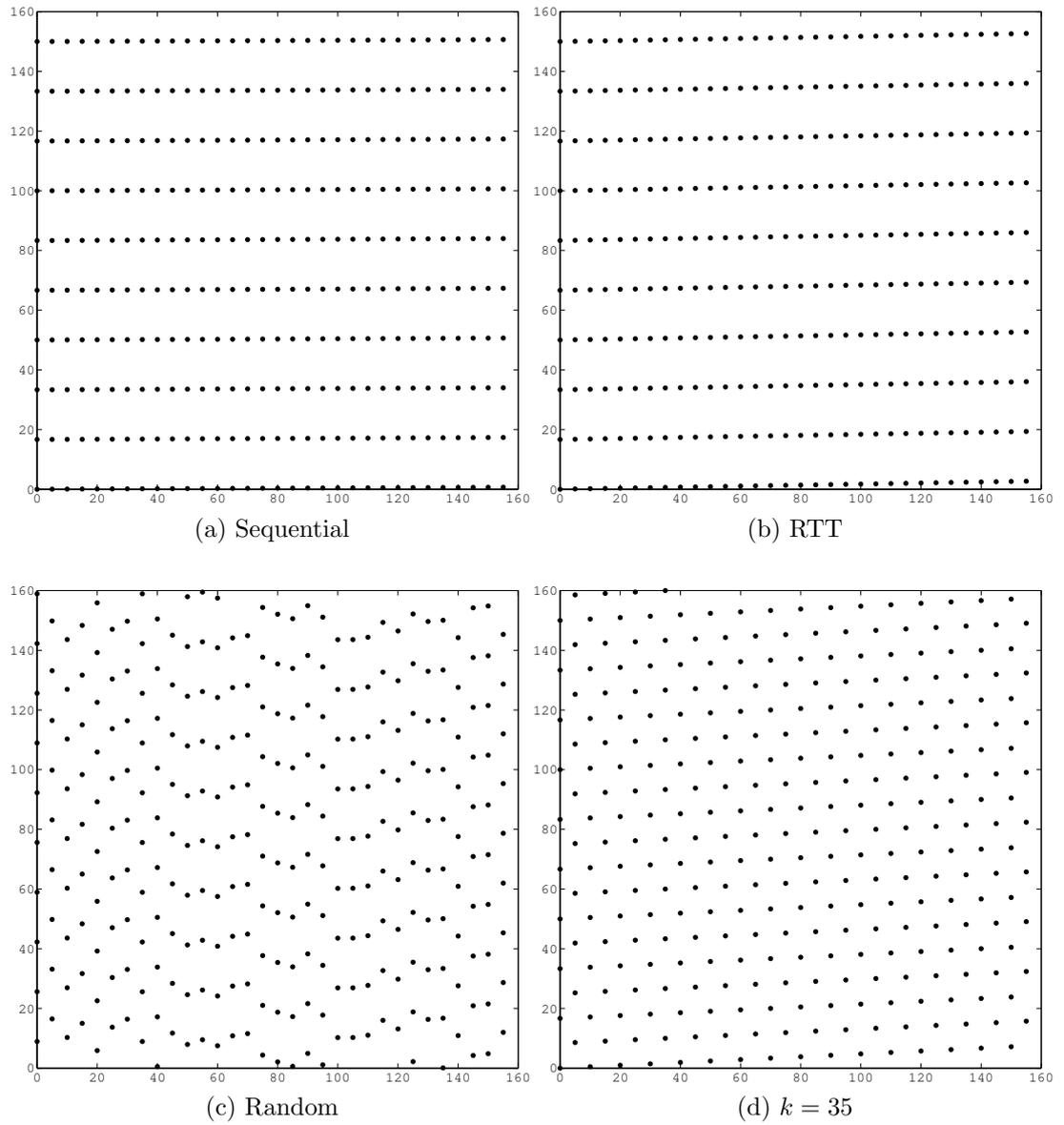


Figure 6.2: Plots of the sampling pattern for the different firing orders

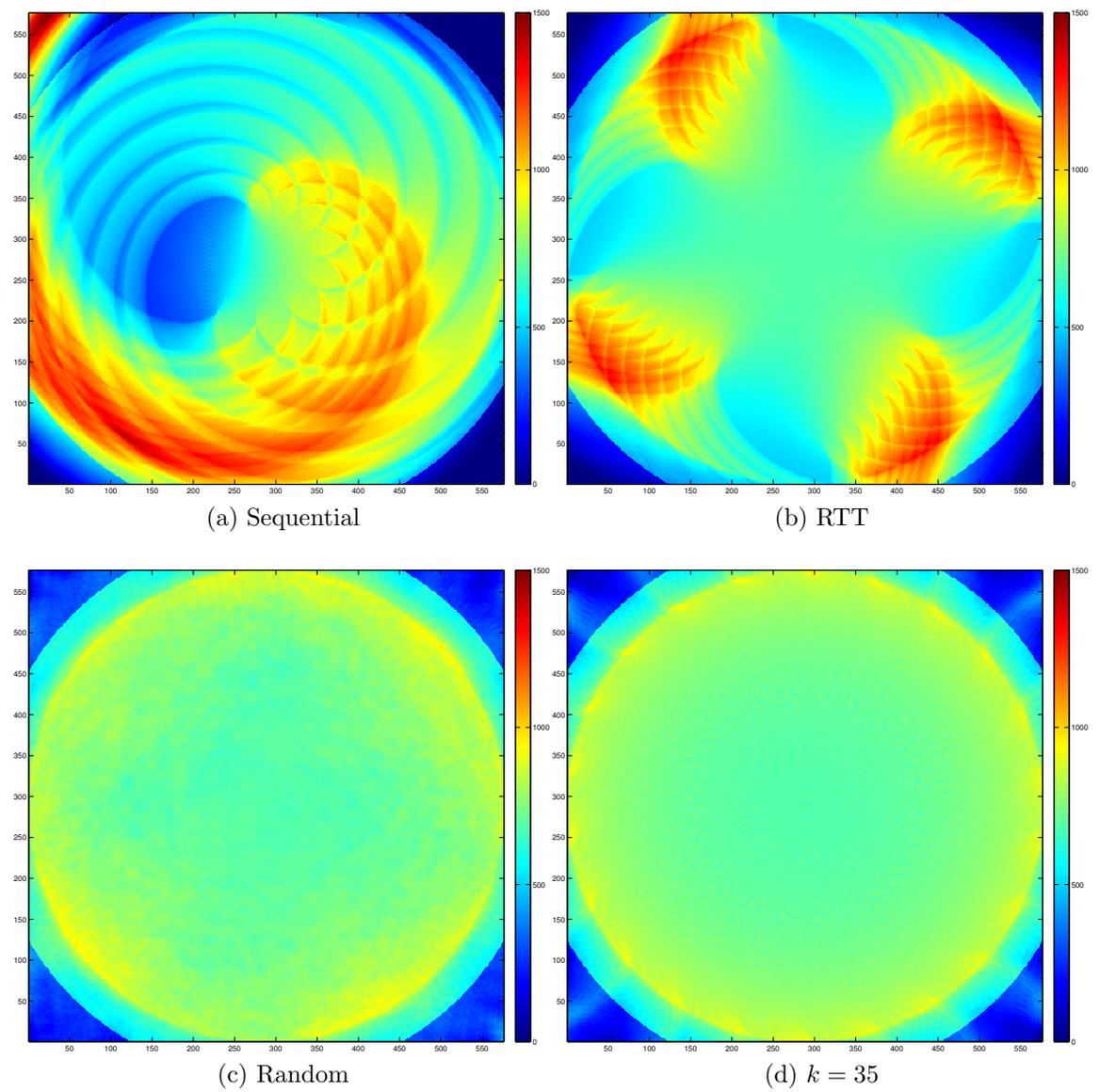


Figure 6.3: Ray density images for the different firing orders

Firing Order	$\mu$	$\sigma$
Sequential	758.3425	230.6479
RTT	758.4147	169.8920
Random	758.3571	58.0326
$k = 35$	758.3417	53.5454

Table 6.1: Mean and standard deviation of ray density for the different firing orders

a nullspace. We see that some unevenness remains with the RTT firing order, but for the random and  $k = 35$  orders, the distribution of ray density is much more even. To quantify this, mean and standard deviation for the ray density within the circular ROI are given in table 6.1.

### 6.3.3 The Angular Distribution

It is also useful to look at the distribution of the angles from which the rays originated, as this will reveal regions of the reconstruction volume which are unevenly illuminated. As in section 6.2.2, the angle is taken to mean the angle within the plane of the projection of a ray onto a transaxial plane. Since the rays are not actually two-dimensional, in order to consider the distribution of sampled angles within the set of possible angles  $\Omega$ , we take the angle of the detector relative to the source, giving the full range of angles  $[-\pi, \pi)$ .

Each row of the reconstruction matrix  $A$  represents a ray, so for each row of  $A$  we calculate the angle of the projection of this ray in the  $xy$ -plane. We then sum down the columns of  $A$  as in section 6.3.2, but this time bin the results into angular bins of width  $10^\circ$ . In order to normalise the distributions, for each voxel, we divide each bin by the total ray density for that voxel. This gives us the relative distribution of angles of illumination for each voxel represented by  $A$ ; the problem is how do we visualise this data.

Given that we are mainly interested in seeing evenness of illumination angles, we can simply look at an image of the standard deviation of the angular data for each voxel. If there are regions in the image showing higher standard deviation then these voxels are likely to be illuminated from an uneven distribution of angles. Results for a

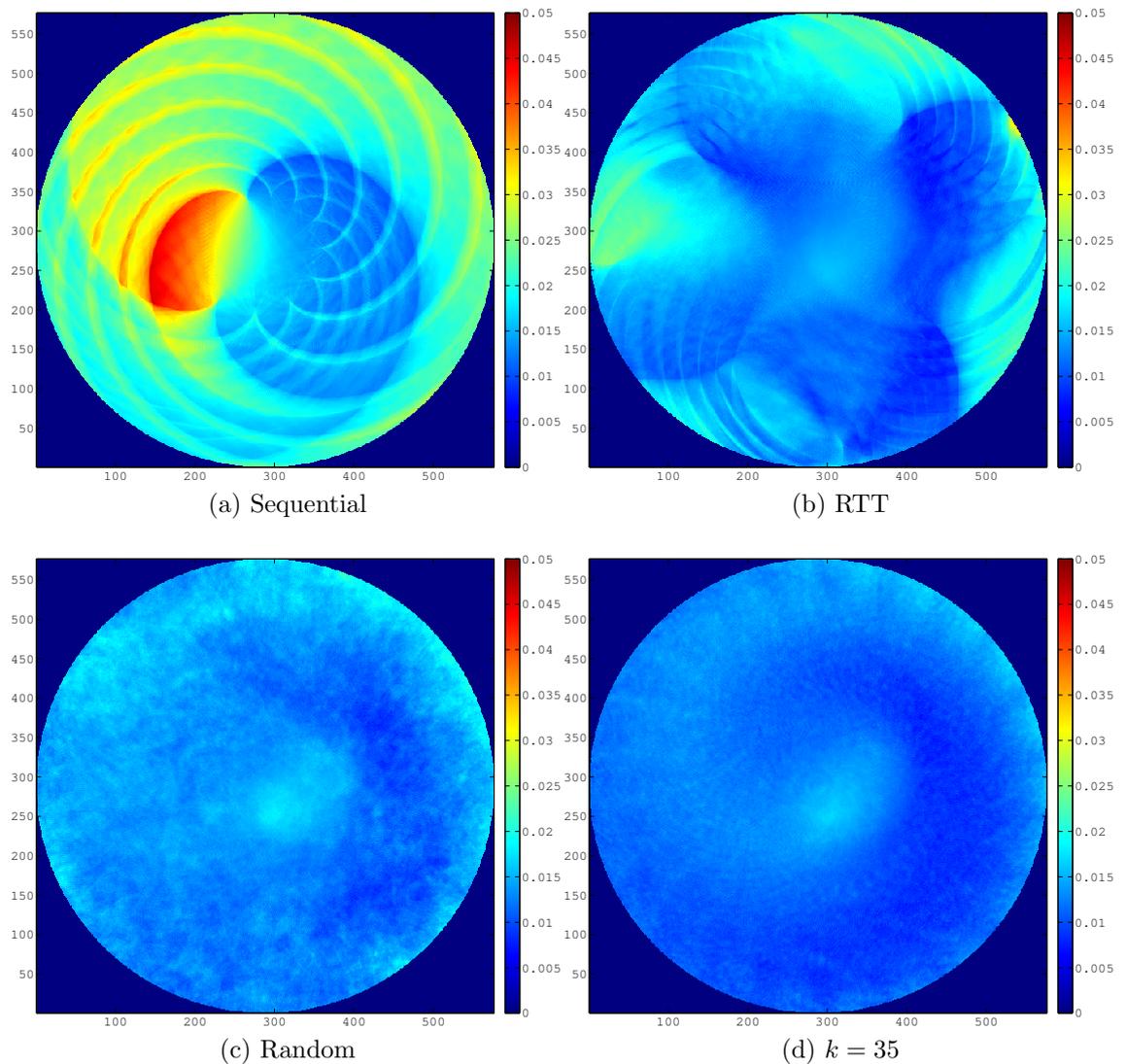


Figure 6.4: Images of standard deviation of the angular distribution for the different firing orders

single slice through the whole  $800 \times 800$ mm reconstruction volume at  $1.4 \times 1.4 \times 1.4$ mm resolution are shown in figure 6.4. We ignore voxels outside the ROI as these tend to be illuminated from a much narrower range of angles, causing poor scaling of the colour map within the ROI.

We see that the RTT, random and  $k = 35$  firing orders all have relatively low standard deviation over the entire ROI, the only exception to this being some regions near the edge for the RTT firing order. However, for the sequential firing order, we have regions where the standard deviation is much higher. Figure 6.5 shows bar graphs of the distribution of angles for individual ‘worst case’ voxels for each firing

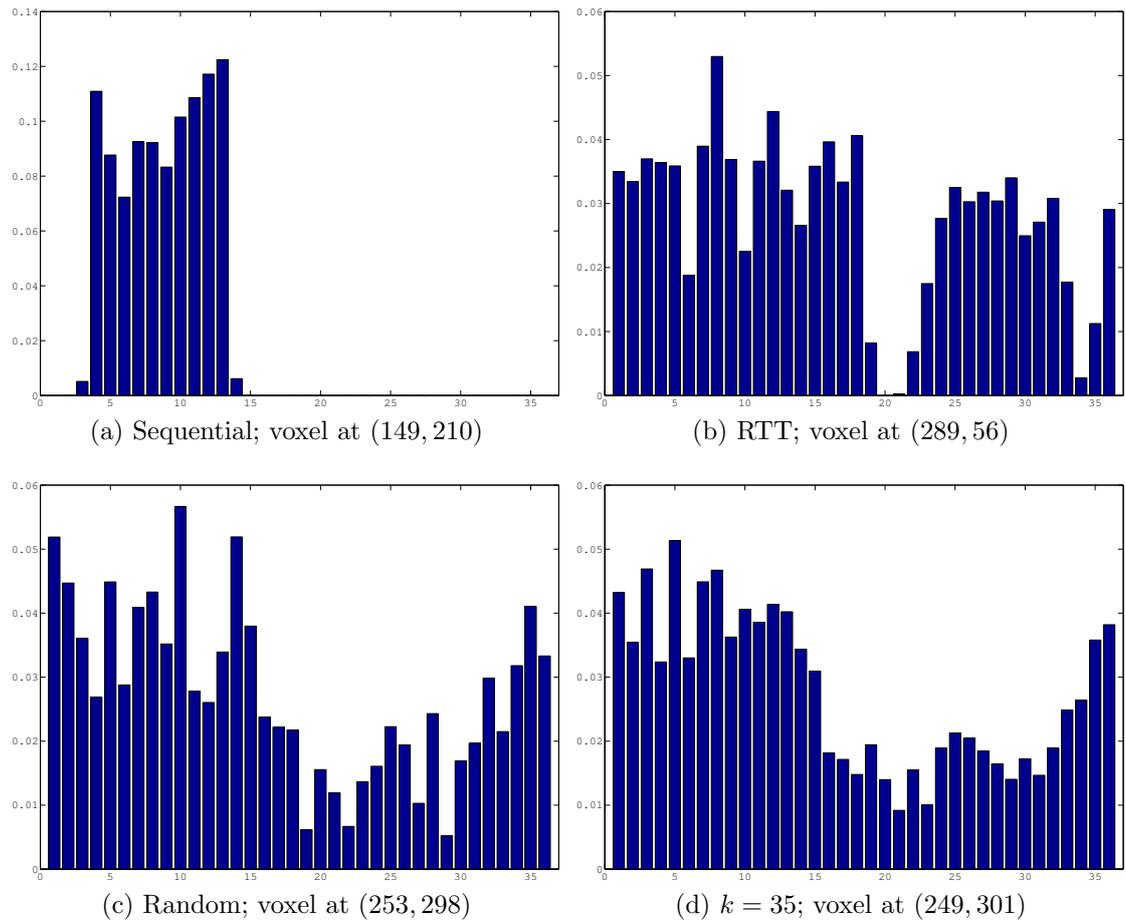


Figure 6.5: Distributions of angles for individual voxels for the different firing orders order.<sup>1</sup>

Figure 6.5a shows us that with the sequential firing order, there are voxels that are not illuminated at all from a wide range of angles. It should be noted that the distribution of angles for this particular voxel is quite typical for the voxels within the low ray density region of figure 6.3a; this can be seen using visualisations of the type shown in figure 6.6, which plots an angular bar graph showing the distribution at each voxel location, the colour being set by the ray density of the voxel, according to the same colour mapping as figure 6.3. For clarity, the angular bins have been increased in size to  $20^\circ$ .

Figures 6.5b, 6.5c and 6.5d show a much more even distribution of angles for individual high standard deviation voxels with the other firing orders. Although

<sup>1</sup>Actually, for the RTT order, there are worse cases than the distribution for the voxel shown. However, these are very close to the edge, and therefore of little practical consequence.

there is still some unevenness, apart from with the RTT firing order where a small gap remains, there are no gaps in the distribution. A visualisation of a small region at the centre of the reconstruction slice is shown in figure 6.6b, showing much more even distribution across this ‘worst case’ region than the sequential firing order.

### 6.3.4 Expected Results

From observations made about the visualisations in this section, we can reasonably expect the sequential firing order to perform significantly worse than the other firing orders. In the regions within the reconstruction volume where the distribution of illumination angles is particularly uneven we may expect to see limited angle type artefacts. We can also reasonably expect the  $k = 35$  firing order to perform better than the RTT order, although perhaps only slightly.

## 6.4 Numerical Comparisons

Our main numerical tools for analysing differences between firing orders are the SVD and condition number of the reconstruction matrix  $A$ , representing a discretisation of the divergent beam equation describing the system. We assume in the following that the matrix  $A$  is large, sparse, real-valued and of size  $m \times n$ .

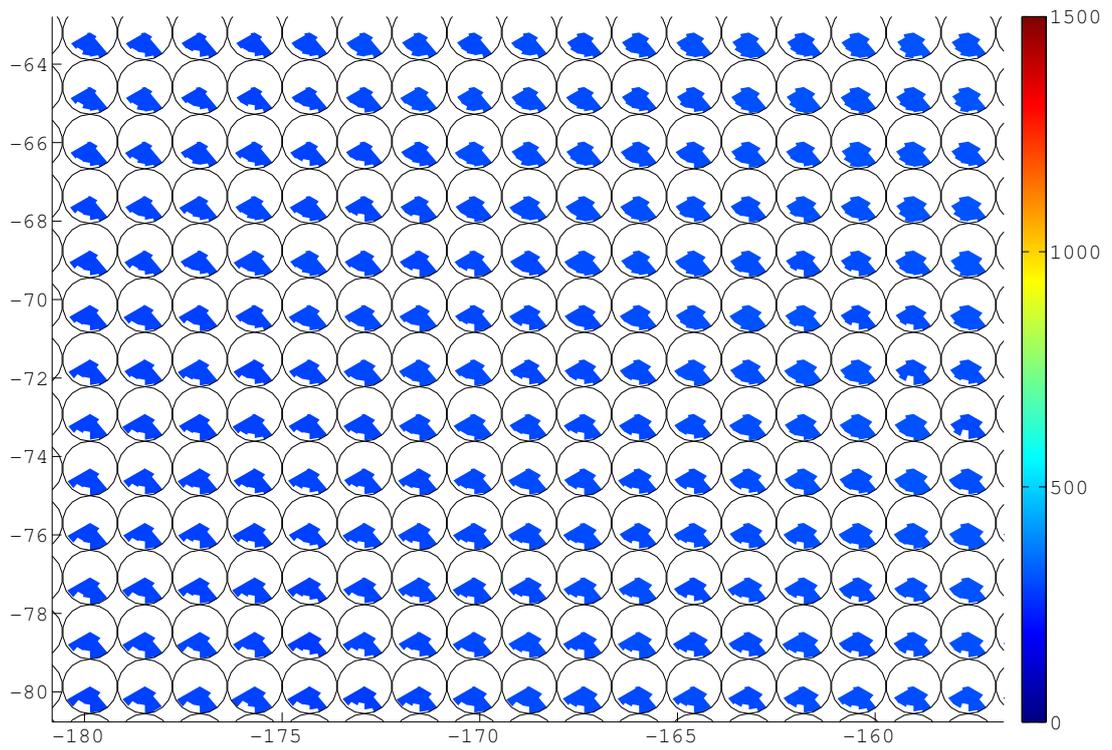
### 6.4.1 Condition Number Estimation

We denote by  $\kappa(A)$  the standard 2-norm condition number, defined as

$$\kappa(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = \frac{\sigma_{max}(A)}{\sigma_{min}(A)}, \quad (6.7)$$

where  $\sigma_{max}(A)$  and  $\sigma_{min}(A)$  are the largest and smallest singular values respectively of  $A$ . Hence, we may calculate  $\kappa(A)$  by first calculating the SVD of  $A$ . For an arbitrary real-valued  $m \times n$  matrix  $A$ , we denote the SVD by the respective  $m \times m$ ,  $m \times n$  and  $n \times n$  matrices  $U$ ,  $\Sigma$  and  $V$  such that

$$A = U\Sigma V^T = \sum_{i=1}^n \mathbf{u}_i \sigma_i \mathbf{v}_i^T. \quad (6.8)$$



(a) Sequential

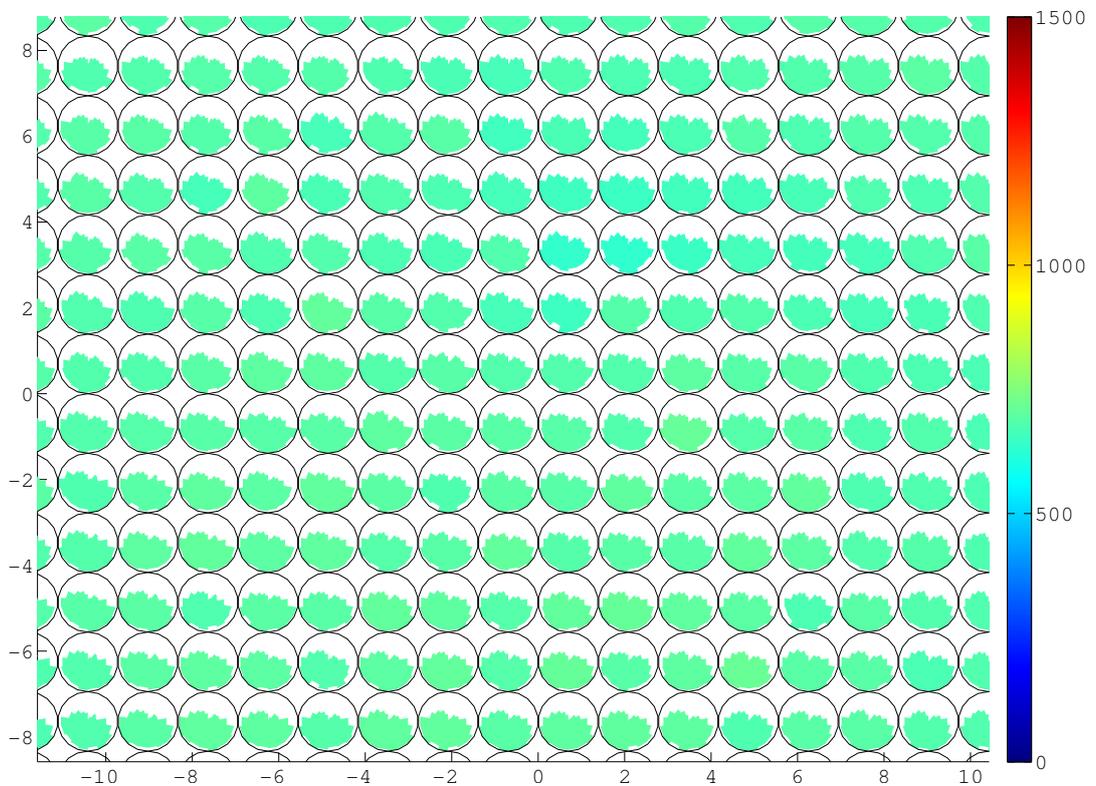
(b)  $k = 35$ 

Figure 6.6: Visualisation of the angular distribution for a small region of the reconstruction slice using radial histograms for each voxel

The matrix  $\Sigma$  is the diagonal matrix whose diagonal entries  $\sigma_1, \dots, \sigma_n$  are the singular values of  $A$ . The columns of the matrices  $U = (\mathbf{u}_1, \dots, \mathbf{u}_m)$  and  $V = (\mathbf{v}_1, \dots, \mathbf{v}_n)$  are the left and respectively right singular vectors.

The size of  $A$  is essentially determined by the size of the reconstruction volume and the coarseness of the discretisation of this volume; the problem we have is that the calculation of the full SVD is a computationally expensive and memory intensive process, and at the size of a typical practical reconstruction problem, the size of  $A$  makes this impossible to achieve with current computer hardware.

Therefore, our initial thoughts were to consider estimation algorithms such as that of Hager [16], which estimates the 1-norm condition number and is implemented in the MATLAB function `condst`. Also considered were statistical condition estimation techniques such as [33], and approaches based on data mining techniques such as given in [71] and [18]. However, since we are interested in *comparing* the condition numbers of the matrices for different firing orders, it was decided that none of these methods would be suitable; a direct calculation approach was therefore sought.

In order to calculate  $\kappa(A)$  we do not need the full SVD; only the first and last singular values. We have the following well known relationship

$$A^T A = V \Sigma^T U^T U \Sigma V^T = V \Sigma^T \Sigma V^T, \quad (6.9)$$

which shows that the eigenvalues of the square matrix  $A^T A$  are the squares of the singular values of  $A$ .

For a large, sparse, square matrix  $B$ , the Implicitly Restarted Arnoldi Method is an efficient method of calculating small sets of the eigenvalues of  $B$ ; this is implemented in ARPACK library routines which are called from the MATLAB function `eigs` (see [37]). Hence we can use `eigs` to calculate the largest and smallest eigenvalues of  $A^T A$  and then take square roots to calculate  $\kappa(A)$ . This has important implications for numerical precision; the tolerance to rounding error when calculating the eigenvalues of  $A^T A$  with double precision arithmetic is approximately  $10^{-16}$ . Since we then have to take the square root to get the corresponding singular values, this gives us a tolerance of only approximately  $10^{-8}$ . Therefore any singular values

calculated as less than this must be assumed to be zero.

For a typical matrix  $A$  representing the RTT80 projection process, using a computer with 48GB memory, this technique allows us to calculate the condition number via matrices  $A^T A$  up to size approximately  $100,000 \times 100,000$ . However, the size of the matrix  $A^T A$  for a typical reconstruction volume is much larger than this; we therefore work with smaller volumes and at coarser discretisations. The coarseness of the discretisation clearly has an effect on the condition number of  $A$ , hence we can calculate condition numbers for  $A$  at varying discretisation sizes and examine this relationship.

From looking at the ray density images in section 6.3.2, we observe that with certain firing orders, particularly the sequential order, we get a ‘hole’ of low ray density that rotates from slice to slice. Such a ‘hole’ would be invisible at lower  $z$ -axis resolution; therefore, our approach is to keep the  $z$  resolution high at 12 slices per unit  $z$ -pitch but vary the resolution in the  $x$  and  $y$  directions. Clearly, our maximum resolution in the  $x$  and  $y$  directions is going to be affected by the number of slices we use in the  $z$  direction. Therefore, we do this for a single slice, giving an upper limit on the  $x, y$  resolution of 2mm, and also for a whole unit of  $z$ -pitch (i.e. 12 slices), giving an upper limit of 11mm. In both cases, we use a fixed volume and vary the size of the voxels to fit the volume. In order to avoid creating a nullspace by considering voxels outside of the reconstruction area that are not intersected by any rays, we restrict the  $x$  and  $y$  extents of the volume so as to cover a square inscribed within the 80cm circle.

The condition number will also be affected by the size of the reconstruction volume. In order to examine this relationship, we calculate the condition number for different volume sizes by keeping the  $z$  resolution high at 12 slices per unit  $z$ -pitch, and for the same  $x, y$  resolution, varying the number of slices in the volume. Again, we use the same  $x, y$  extents as before. Using approximately  $20 \times 20 \times 1.4$ mm voxels allows us to vary the size of the volume from 1–60 slices.

Firing order	a	b
Sequential	-1.1031	214.14
RTT	-0.8507	50.22
Random	-0.5646	14.86
$k = 35$	-0.5755	15.65

Table 6.2: Values of constants in equation (6.10) for different firing orders – 1 slice

## 6.4.2 Condition Number Results

### Dependence on Discretisation Size in $x, y - 1$ Slice

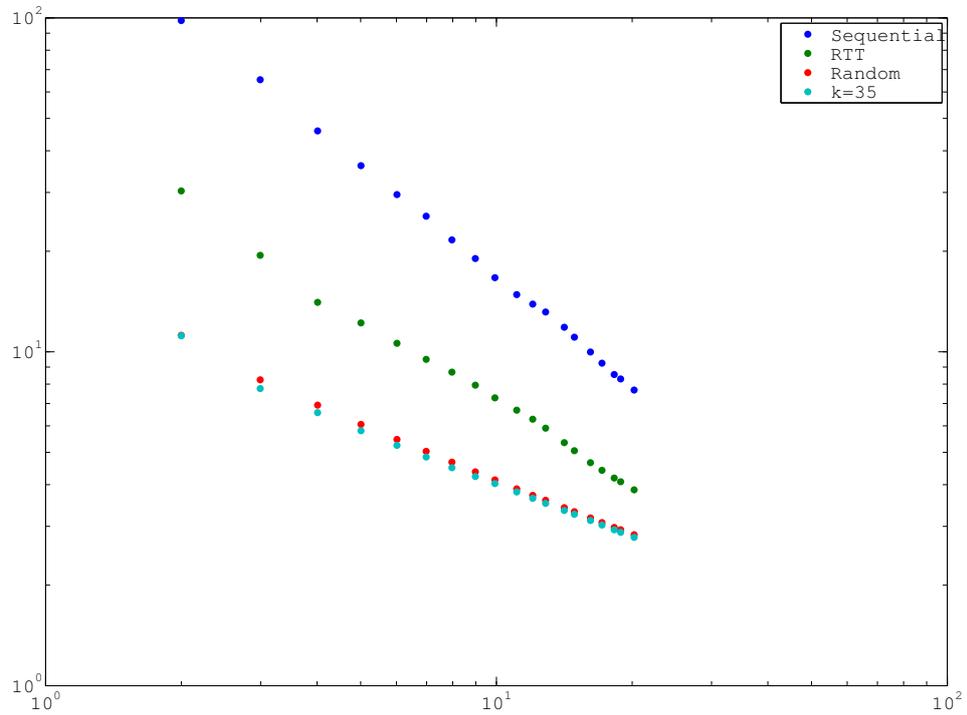
The condition number was calculated at  $x, y$  discretisation sizes from approximately 2–20mm. Results of this for the different firing orders are shown plotted on a log-log scale in figure 6.7a. As we see, the points lie approximately on straight lines, suggesting a relationship of the form

$$\kappa(A) = b.v_{xy}^a, \quad (6.10)$$

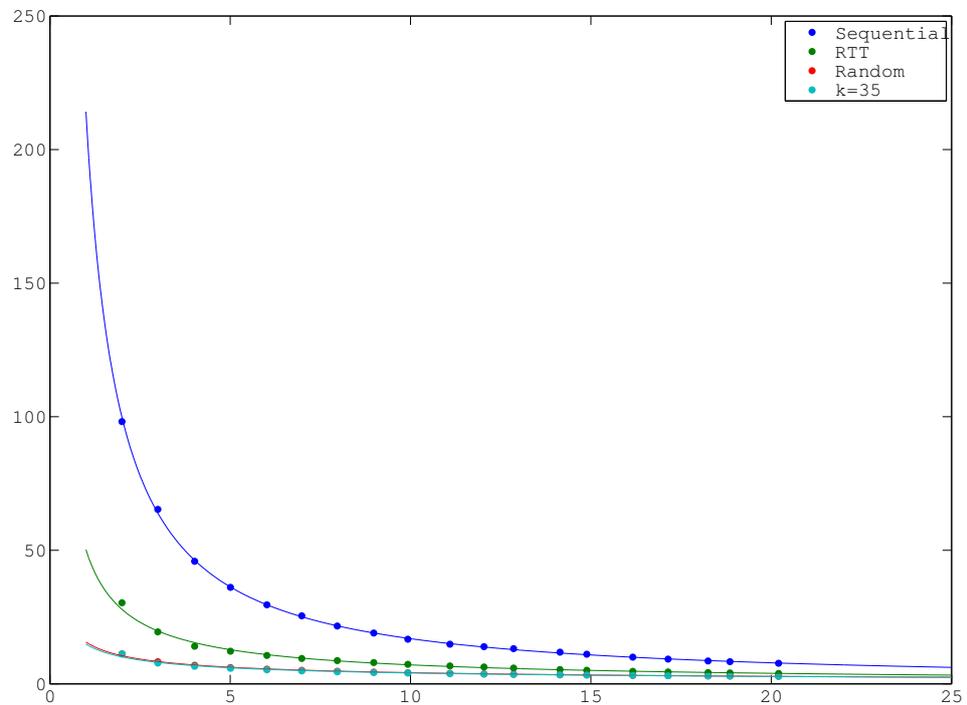
where  $v_{xy}$  is the discretisation size in the  $x$  and  $y$  directions. The values of the constants  $a$  and  $b$  are summarised for the different firing orders in table 6.2. The results and least-squares fit curves of these relationships are shown plotted on a normal scale in figure 6.7b. We see that the condition number for the sequential firing order is significantly higher than that for the other firing orders, which is as we expect.

### Dependence on Discretisation Size in $x, y - 12$ Slices

The condition number was calculated at  $x, y$  discretisation sizes from approximately 11–50mm. As for the single slice case, results for the different firing orders are plotted on a log-log scale in figure 6.8a, and on a normal scale with least-squares fit curves of equations of type (6.10) in figure 6.8b. Values of the constants in these equations are given in table 6.3. We see less difference between the firing orders this time; however, the sequential firing order still has higher condition number. A possible reason for the smaller differences could be that the minimum  $xy$  voxel size of 11mm simply gives too coarse a discretisation for the differences to be noticeable.

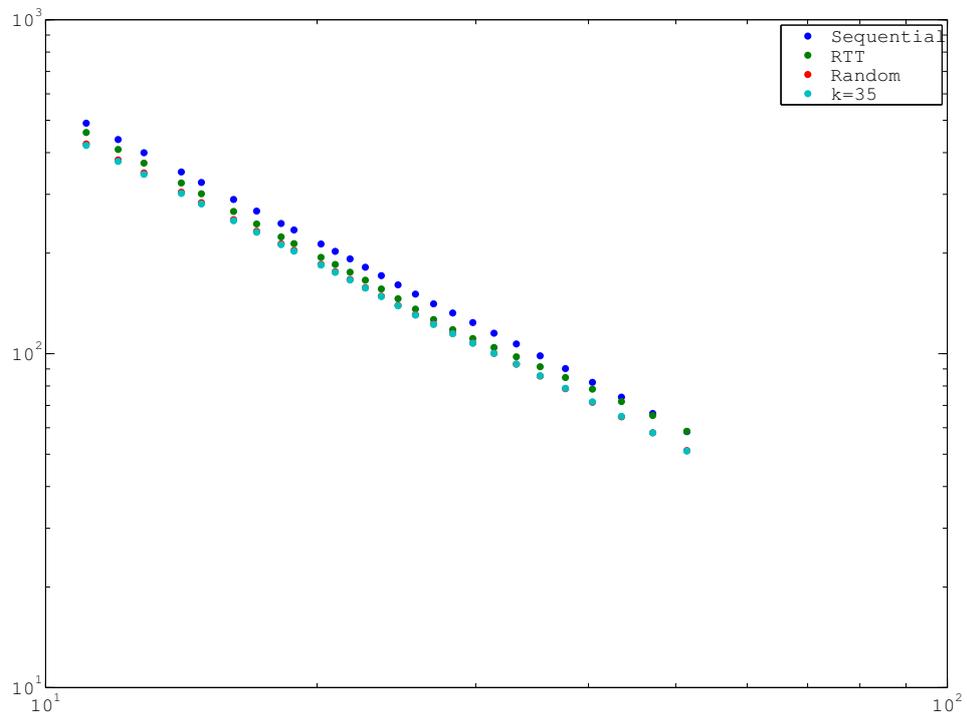


(a) Log-log scale

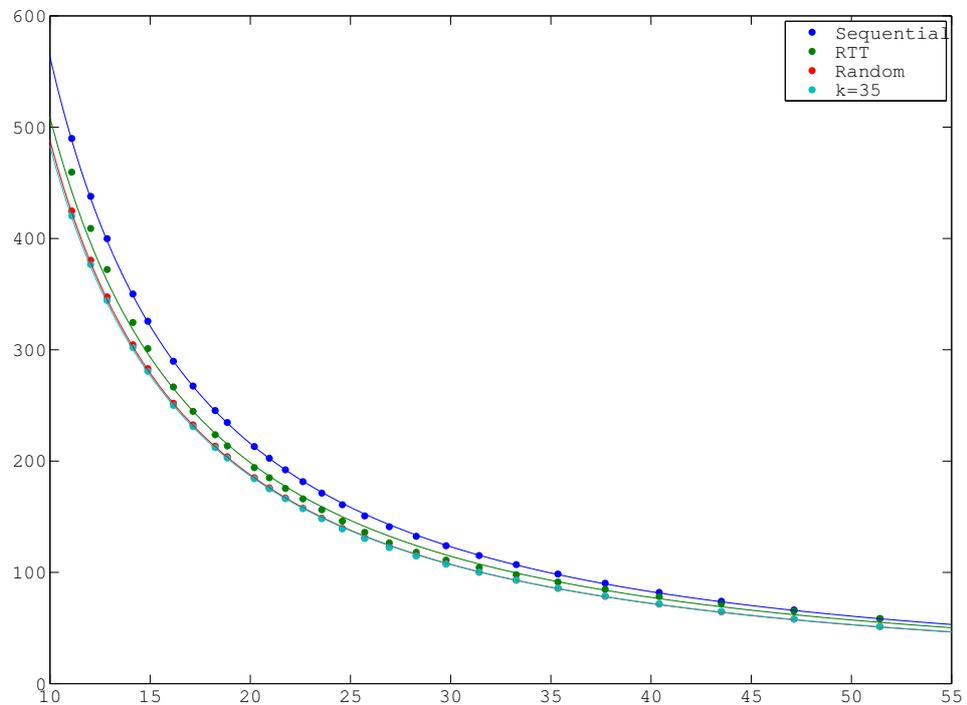


(b) Normal scale, with least-squares fit curves

Figure 6.7: Condition number by discretisation size – 1 slice



(a) Log-log scale

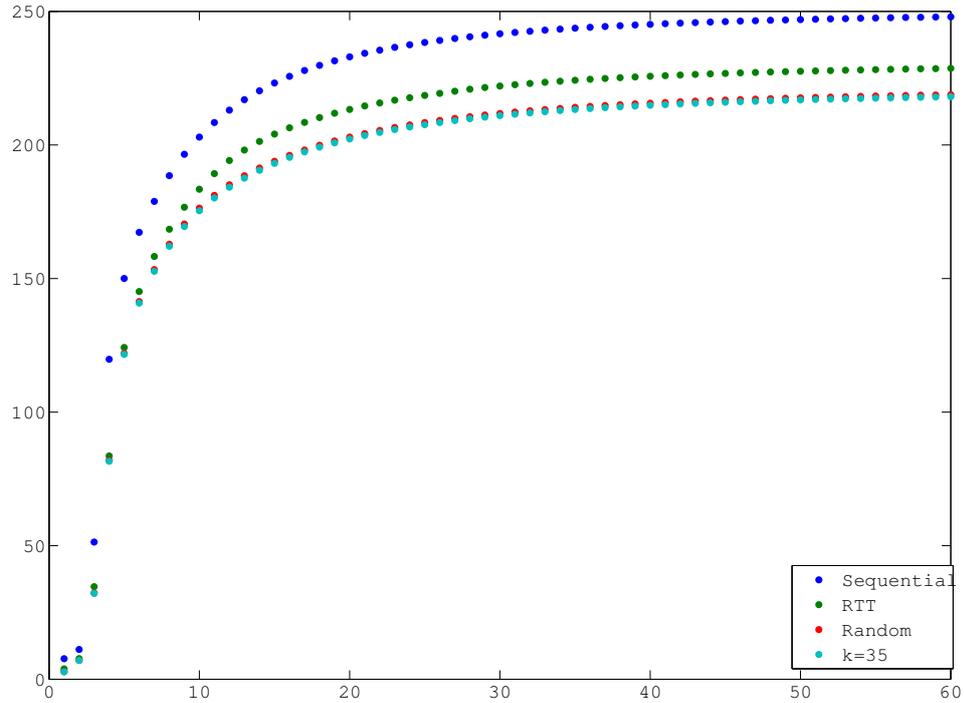


(b) Normal scale, with least-squares fit curves

Figure 6.8: Condition number by discretisation size – 12 slices

Firing order	a	b
Sequential	-1.3848	13,670
RTT	-1.3580	11,610
Random	-1.3807	11,740
$k = 35$	-1.3721	11,380

Table 6.3: Values of constants in equation (6.10) for different firing orders – 12 slices

Figure 6.9: Condition number by volume slices –  $20 \times 20 \times 1.39\text{mm}$  voxels

### Dependence on Volume Size

The condition number was calculated at a discretisation size of approximately  $20 \times 20 \times 1.39\text{mm}$  voxels, for reconstruction volumes with between 1–60 slices in the  $z$  direction. Results are plotted on a normal scale in figure 6.9. We see that for all firing orders, the condition number increases with the size of the reconstruction volume; again, the condition number is higher for the sequential firing order.

### 6.4.3 Calculating the SVD

In order to calculate the full SVD we need to work at a much reduced resolution. Given that we are only interested in the singular values and the right singular vectors, we compute the SVD using the normal matrix  $A^T A$ , as we did for the condition

number calculations. For the condition number calculations, the MATLAB function `eigs` was used since we only needed the smallest and largest singular values. For the full SVD, it is actually more efficient to convert the matrix  $A^T A$  to full format and use the MATLAB function `eig`. Using an 8 core Intel Xeon machine with 48GB RAM this restricts the size of the matrix we can work with to approximately  $30000 \times 30000$ . Again, in order to avoid creating a nullspace by considering voxels outside the ROI, we restrict ourselves to the central square region inscribed within the 80cm tunnel.

We calculate the SVD for volumes of a single slice and 12 slices, keeping the  $z$  resolution at 12 slices per unit  $z$ -pitch. For the single slice case, we use an  $x, y$  resolution of  $\sim 4\text{mm}$ ; for the 12 slice case we use  $\sim 12\text{mm}$ .

#### 6.4.4 SVD Results

##### 1 Slice

Normal and log scale plots of the singular values are shown in figures 6.10a and 6.10b respectively. We see that for the sequential and RTT firing orders, the small singular values decay more rapidly, which is to be expected based on the condition number results.

Images of the first 6 right singular vectors are shown for each firing order in figures 6.11–6.14. Some interesting patterns can be seen; in particular, we note that the the singular vectors for the  $k = 35$  firing order are smooth and symmetric, while those for the sequential and RTT orders are highly asymmetric.

##### 12 Slices

Normal and log scale plots of the singular values are shown in figures 6.15a and 6.15b respectively. Although the differences are smaller, we still see a more rapid decay of the small singular values with the sequential and RTT firing orders.

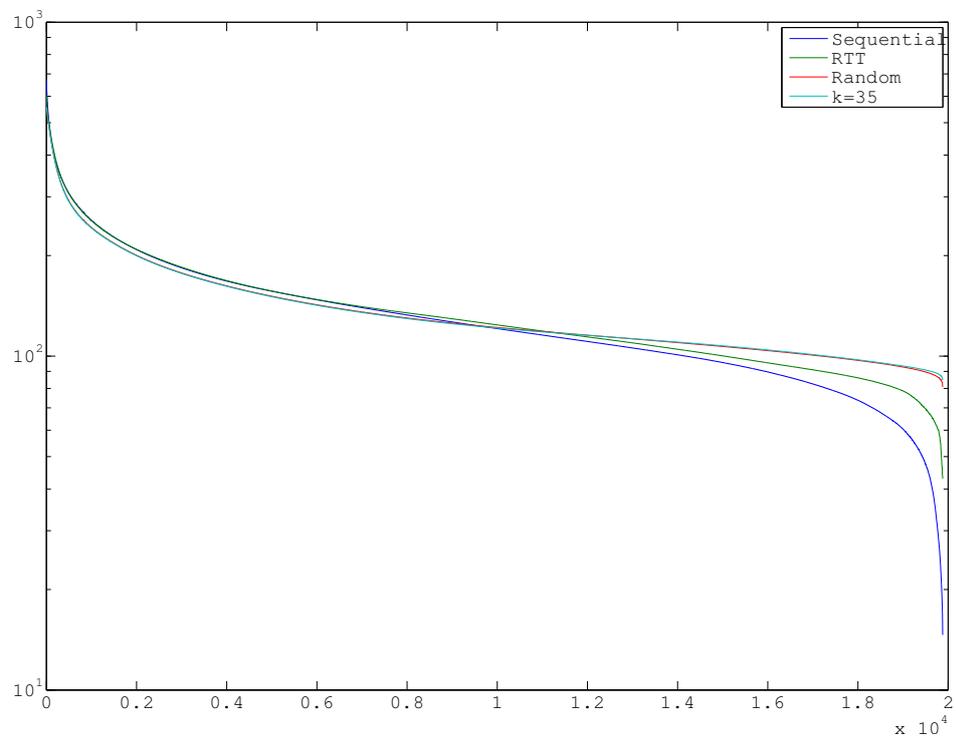
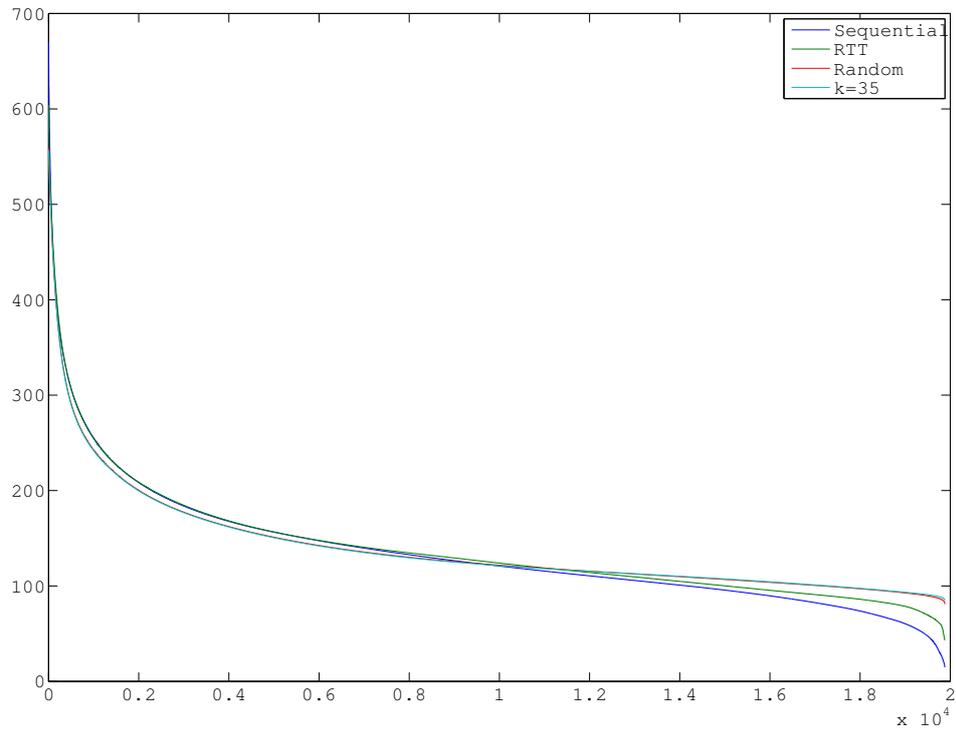
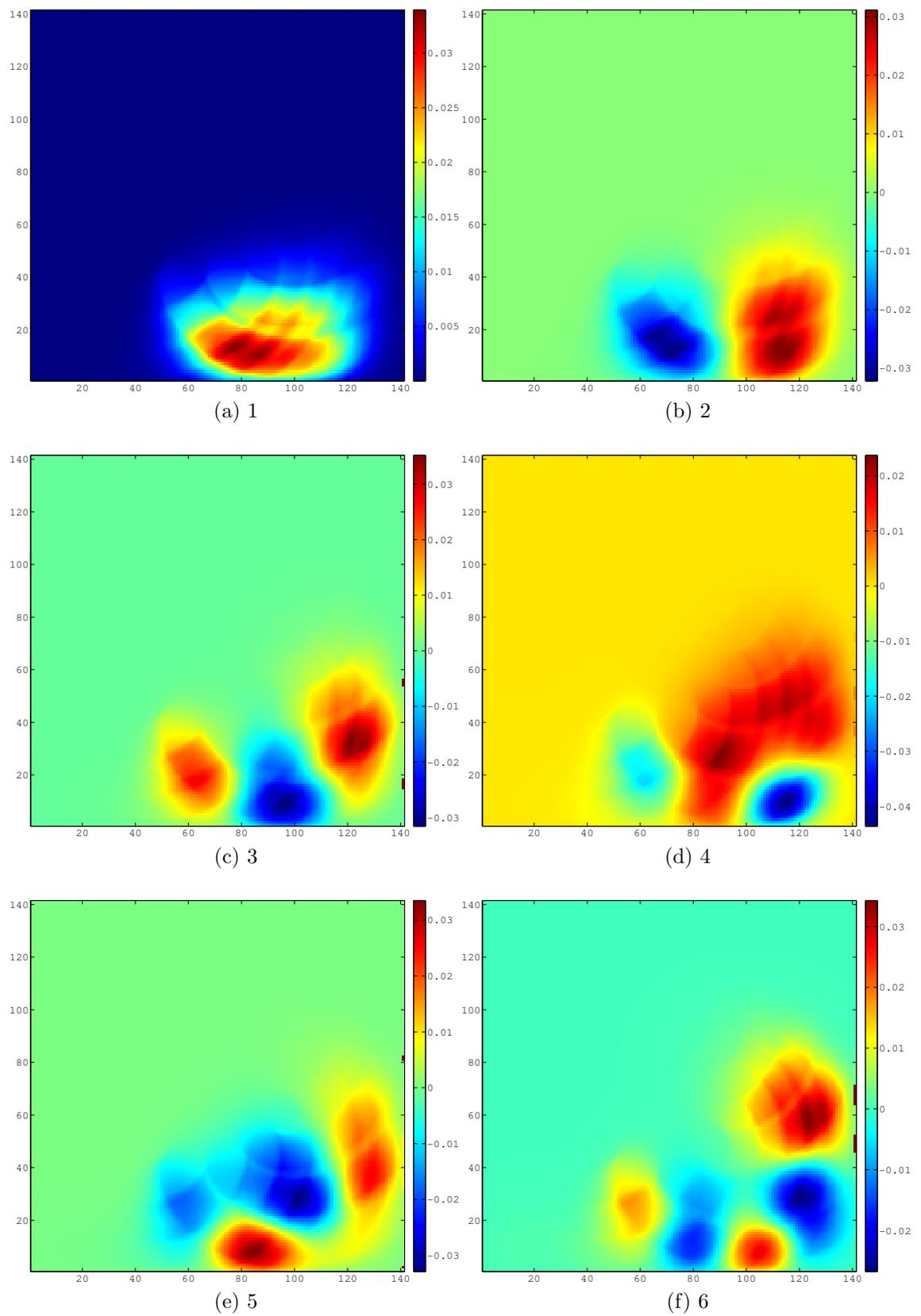
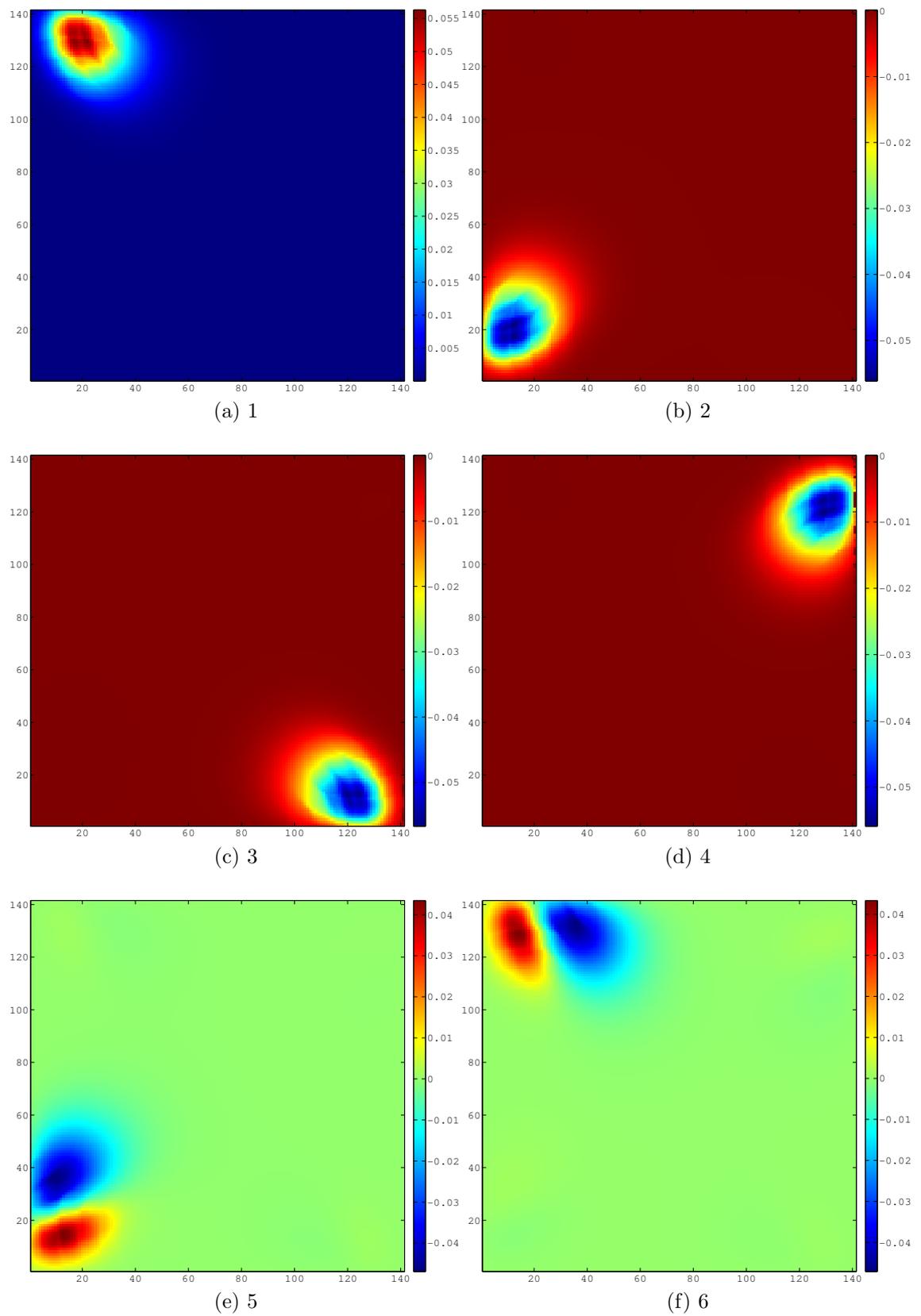
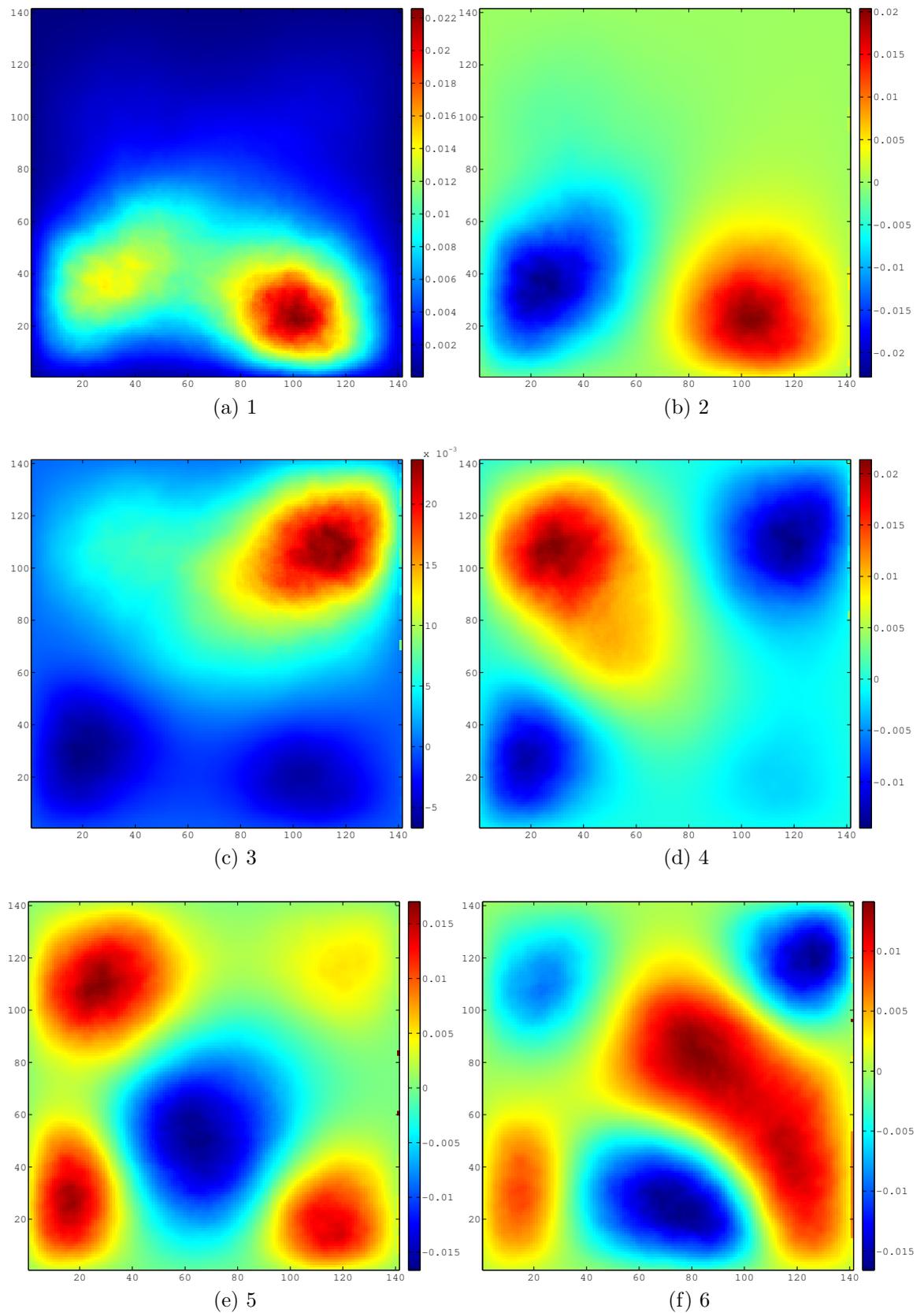
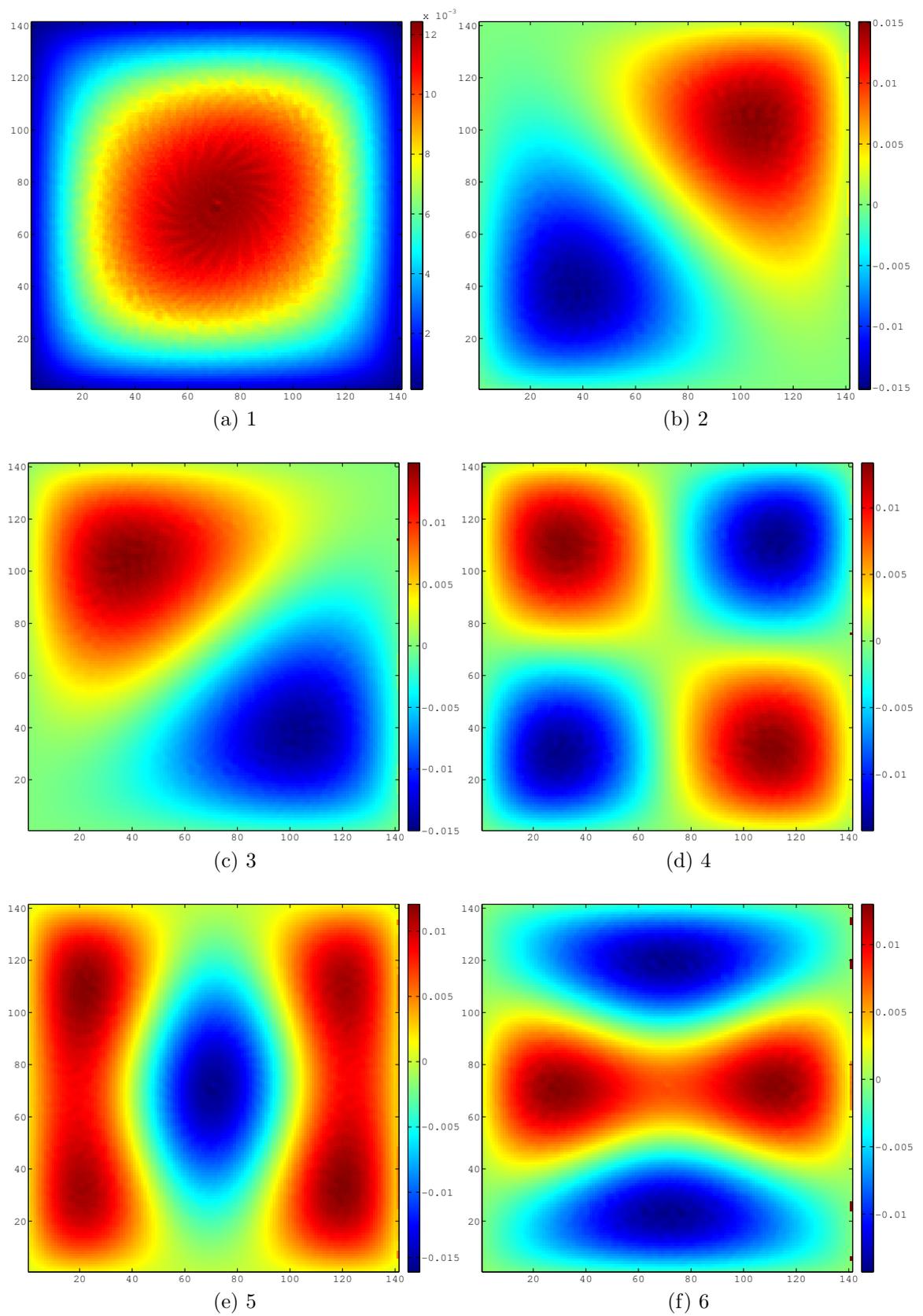


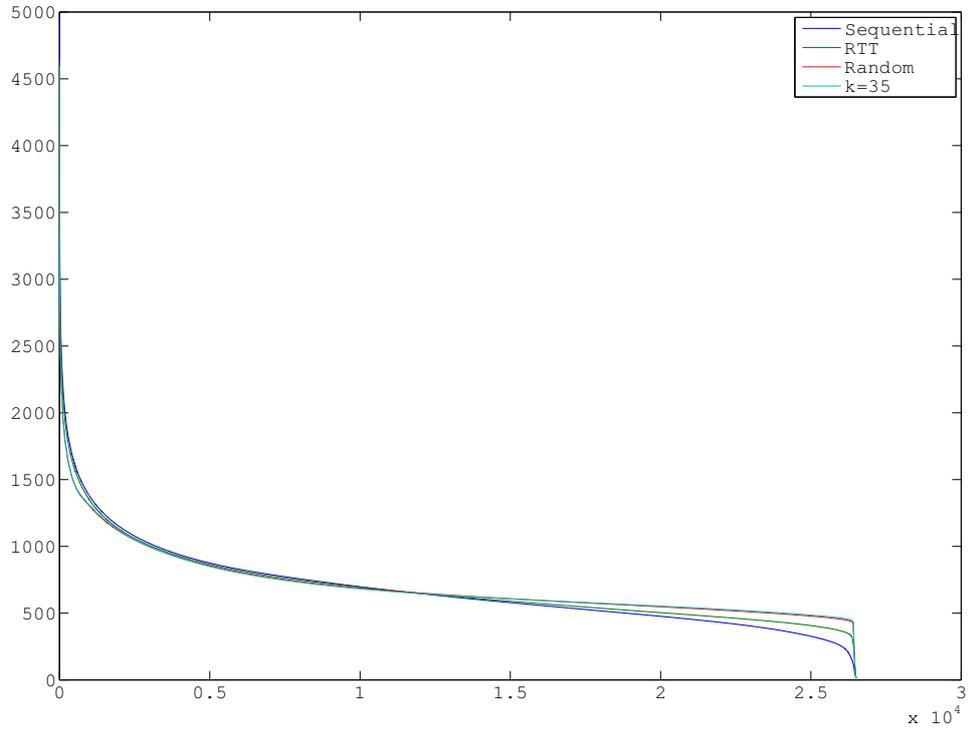
Figure 6.10: Singular values of  $A$  representing 1 slice

Figure 6.11: First 6 right singular vectors of  $A$  for sequential firing order

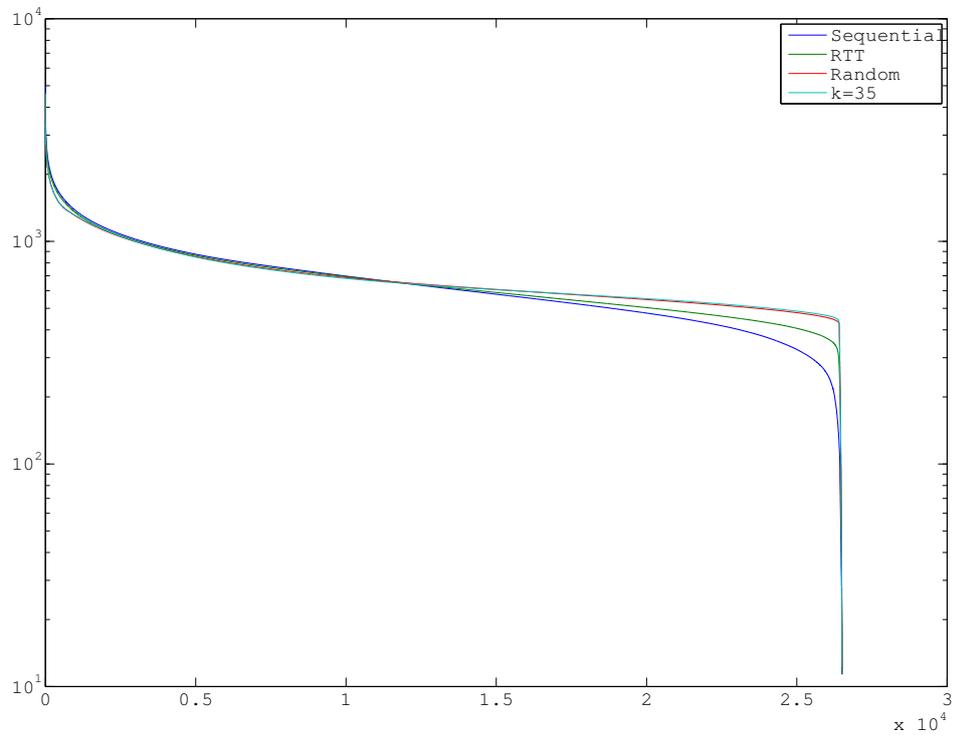
Figure 6.12: First 6 right singular vectors of  $A$  for RTT firing order

Figure 6.13: First 6 right singular vectors of  $A$  for random firing order

Figure 6.14: First 6 right singular vectors of  $A$  for  $k = 35$  firing order



(a) Normal scale



(b) Log scale

Figure 6.15: Singular values of  $A$  representing 12 slices

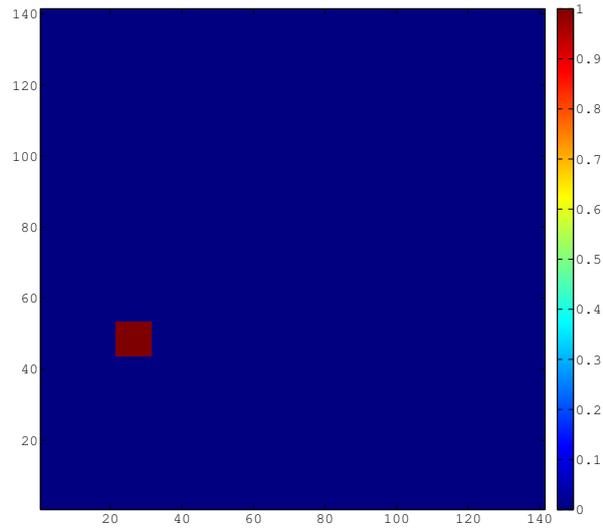


Figure 6.16: ‘Phantom’ for singular vector projections

### 6.4.5 Projections Onto The Singular Vectors

Any vector  $\mathbf{x}$ , of length  $n$ , representing the discretisation of an object density function, may be written in terms of the right singular vectors of  $A$  as follows

$$\mathbf{x} = \sum_{i=1}^n (\mathbf{v}_i^T \mathbf{x}) \mathbf{v}_i. \quad (6.11)$$

Here, each  $\mathbf{v}_i^T \mathbf{x}$  is the projection of  $\mathbf{x}$  onto the  $i^{\text{th}}$  singular vector. Each such projection represents the relative contribution of that singular vector to the image  $\mathbf{x}$ . For a particular image  $\mathbf{x}$ , it is interesting to look at the distribution of the projections onto the singular vectors, and also to examine approximations of  $\mathbf{x}$  by sums of the form (6.11) for only the first  $p$  singular vectors, for some  $p < n$ .

In order to do this, for simplicity and ease of visualisation, we consider only a single slice reconstruction volume, and use the ‘phantom’ image in figure 6.16. This consists of a square of 1’s against a background of 0’s, positioned so that it corresponds to the low ray density ‘hole’ of the sequential firing order.

Graphs of the absolute value of the projections, averaged over every 47 consecutive projections for ease of visualisation, are shown for each firing order in figure 6.17. We see that the plot for the sequential firing order shows a much greater dependence on the singular vectors corresponding to the small singular values.

Images of approximations to the phantom by sums of the form (6.11) for the first  $p$

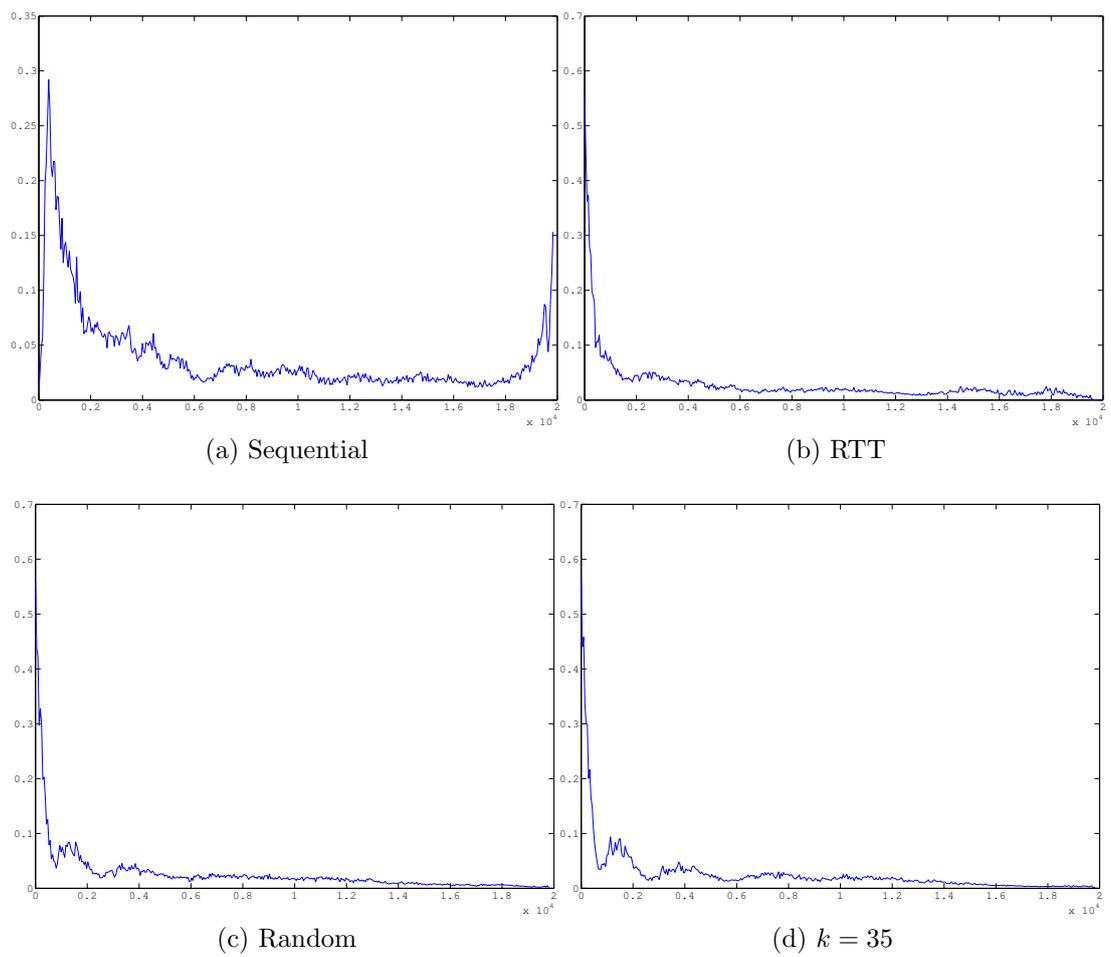


Figure 6.17: Projections onto singular vectors for the different firing orders

singular vectors are shown in figures 6.18–6.21. As expected, results for the sequential firing order are significantly worse than for the others, with even the  $p = 10000$  approximation showing significant artefacts. Results for the random and  $k = 35$  firing orders are similar and noticeably better than the other two.

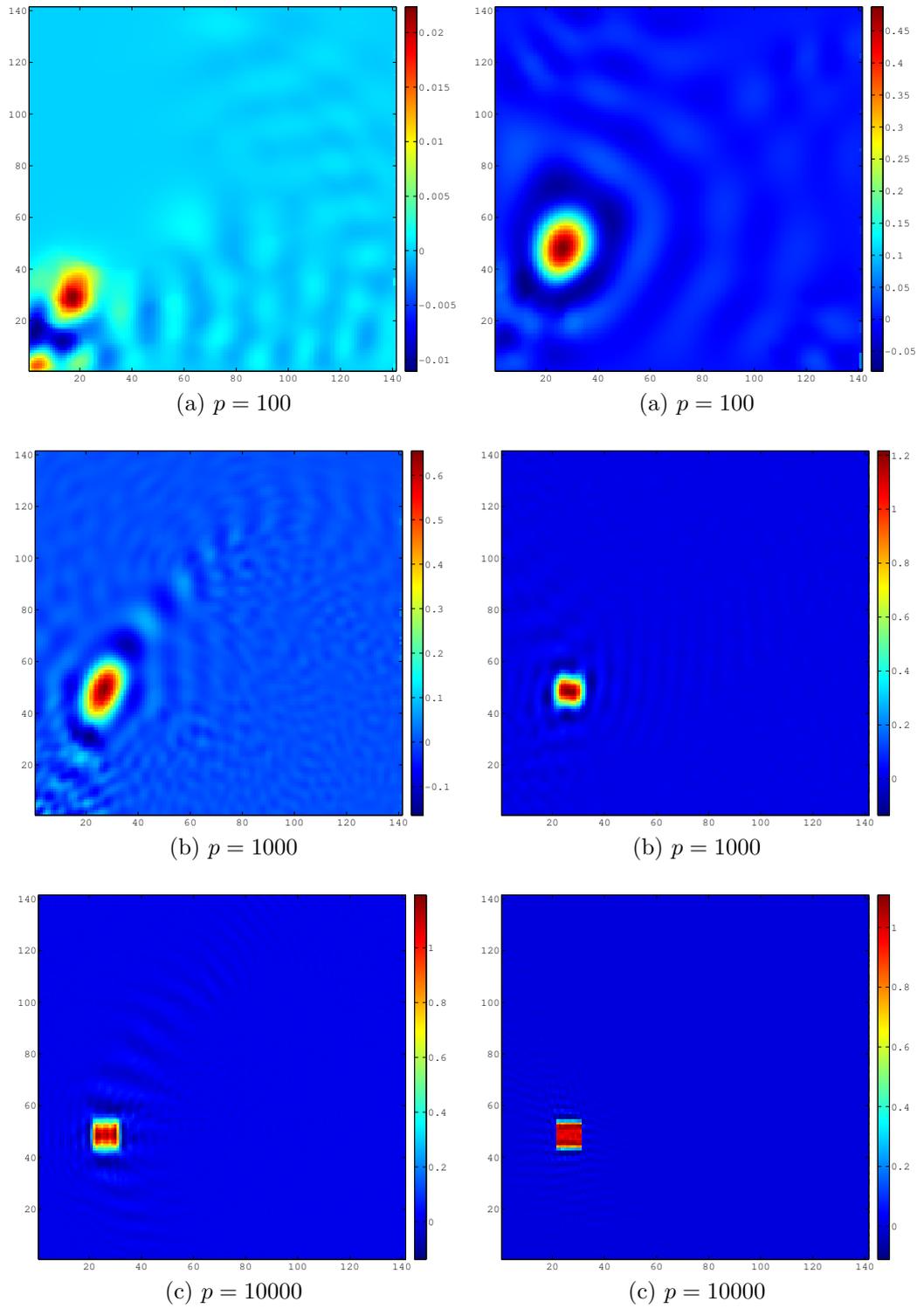


Figure 6.18: Approximation by first  $p$  singular vectors – sequential

Figure 6.19: Approximation by first  $p$  singular vectors – RTT

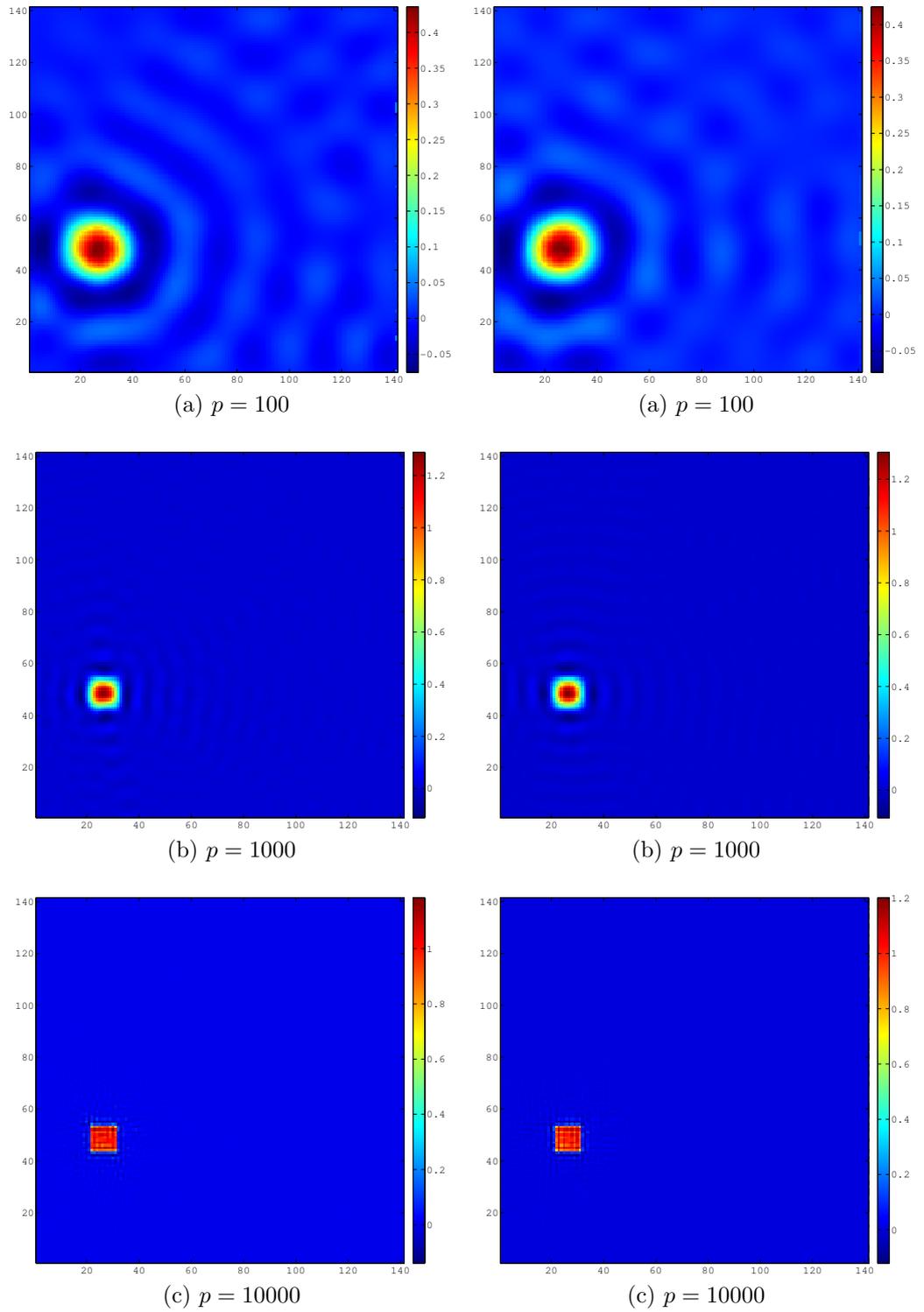


Figure 6.20: Approximation by first  $p$  singular vectors – random

Figure 6.21: Approximation by first  $p$  singular vectors –  $k = 35$

# Chapter 7

## Reconstruction Results

We give reconstruction results using the method of section 5.4, with the aim of examining the effect of the firing order. Reconstruction for both phantom and real data is performed at the resolution of  $1.39 \times 1.39 \times 1.39$ mm voxels, as discussed in section 5.4.3.

### 7.1 Results With Phantom data

We give results of reconstruction from simple phantoms for the sequential, RTT and  $k = 35$  firing orders. Since the random firing order lacks rotational invariance, the matrix  $A$  does not have the structure exploited in the process of section 5.4.2 for reducing the memory requirement, so reconstruction by this method at the required resolution is not possible due to memory constraints.

#### 7.1.1 Calculating Phantom Data

Phantom data for simple rectangular objects were calculated analytically by essentially the same method as for the two-dimensional case in section 3.2.4, with the added complication that we now have to consider edges in the  $z$  direction. Before calculating the ray paths, we set a number of full revolutions to calculate over based on the size of the object, and then calculate  $z$ -coordinates for all source positions based on the firing order. The data are stored in a four-dimensional MATLAB array,

the dimensions representing source number, detector number, detector ring number and revolution number. This is then reshaped into a vector which is sorted so that the projections for each source start at the first active detector, as stated in section 5.4.4. By the linearity of the ray transform, more complicated phantoms are created by simply summing the data for the single objects.

Two simple phantoms were created; the first consists of a  $14 \times 14$  array of  $20 \times 20 \times 30$ mm cuboids, positioned so that the long edges are parallel to the  $z$ -axis and spaced 20mm apart in the  $x$  and  $y$  directions. This will be known as the *multi-cuboid* phantom.

The second phantom is a  $z$  resolution test and consists of a stack of cuboids, edge length 250mm in the  $x$  and  $y$  directions, and of thickness 5, 4.5,  $\dots$ , 0.5mm in the  $z$  direction. This will be known as the *resolution test* phantom. In both cases and for all firing orders, 5% Gaussian noise was added.

## 7.1.2 Results

### Multi-cuboid Phantom

Reconstruction results for the multi-cuboid phantom after respectively 20 and 50 iterations are shown in figures 7.1 and 7.2. For each firing order we show a transaxial slice through the centre of the cuboids and a sagittal slice through the centre of the 8<sup>th</sup> row of cuboids. The images for the sagittal slices are stretched in the  $z$  direction to make them clearer. A plot of the 2-norms of the residuals for each firing order, showing the convergence rate, is shown in figure 7.5.

As we see, the reconstructions for the  $k = 35$  firing order show far fewer artefacts than those for the sequential and RTT firing orders, which is as we expected. The transaxial slice for the sequential firing order shows typical limited angle type artefacts for some of the cuboids, presumably located in the low ray density ‘hole’ of figure 6.3a. If we look at other slices in the volume, we see these same type of artefacts, but at rotated positions in the slice, backing up this theory. We see these regions as the off-vertical streaks in the sagittal slice.

Although the artefacts in the sequential and RTT firing order images have become much reduced after 50 iterations, the  $k = 35$  firing order still gives better looking results after just 20 iterations. All reconstructions show an increased level of noise after 50 iterations, which is as expected, since we are not using any additional Tikhonov regularisation.

### Resolution Test Phantom

Reconstruction results for the resolution test phantom after respectively 20 and 50 iterations are shown in figures 7.3 and 7.4. For each firing order we show a transaxial slice through the centre of the thickest cuboid and a sagittal slice through the centre of the cuboids. Again, the images for the sagittal slices are stretched in the  $z$  direction to make them clearer. A plot of the 2-norms of the residuals for each firing order, showing the convergence rate, is shown in figure 7.6.

Since the cuboids in the resolution test phantom have edges that are parallel to planes in which no rays are measured (the planes parallel to the  $xy$ -plane), we expect this to be a difficult phantom to reconstruct using any of the firing orders under test. Looking at the reconstructions, this is indeed the case, with all reconstructions suffering extensively from the typical limited angle streak type artefacts. However, the sequential firing order still gives significantly worse results, with significant artefacts even in the transaxial slice. The RTT and  $k = 35$  firing orders are comparable, with the  $k = 35$  firing order showing the fewest artefacts. Again, this is as we expect.

## 7.2 Working With Real Data

There are additional considerations to be taken into account when working with real data; these are described in this section.

### 7.2.1 The RTT Data Format

The output from the RTT80 detectors is processed by two separate integrator circuits, to allow for data from one integrator to be processed while the other collects

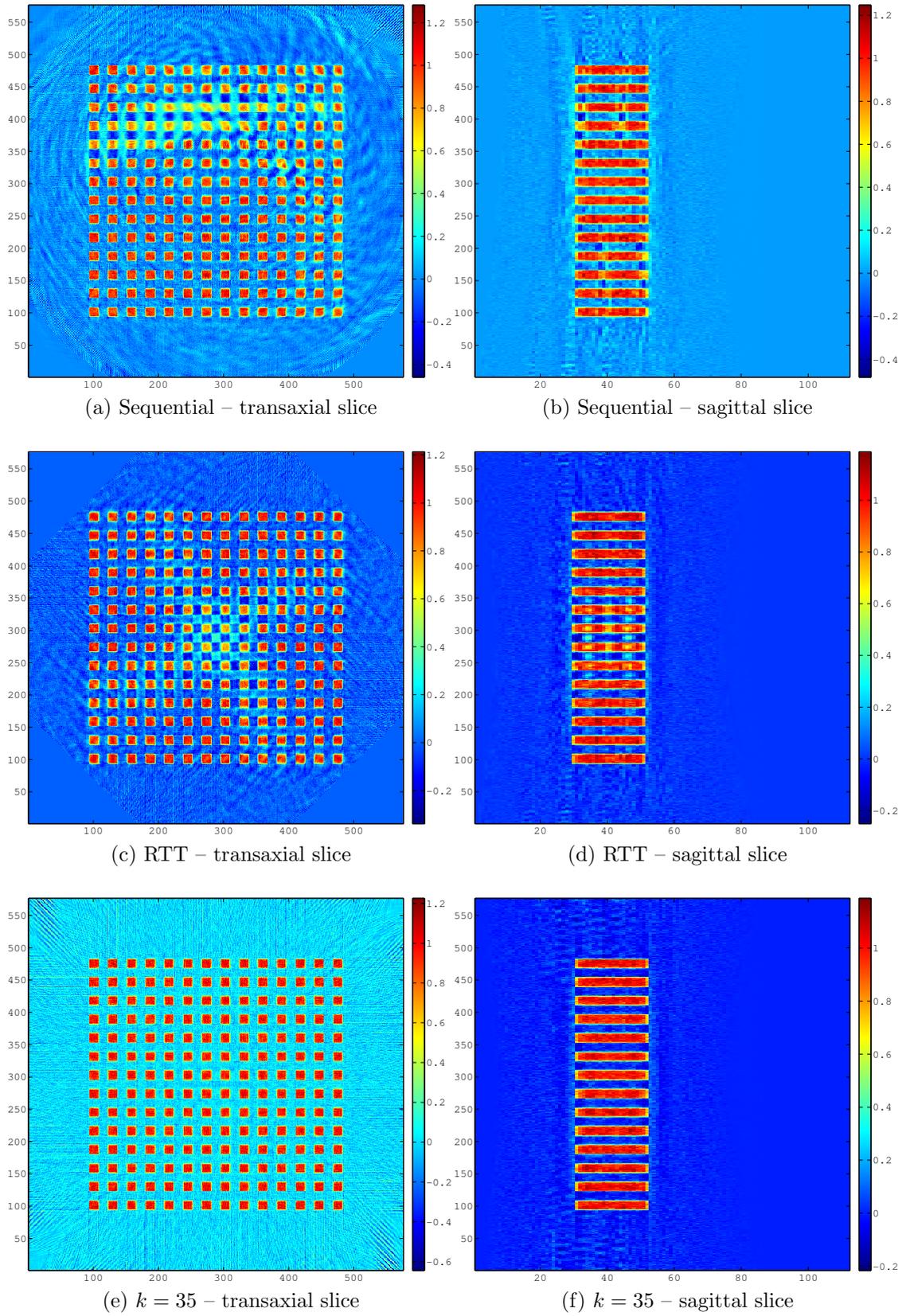


Figure 7.1: Reconstruction of the multi-cuboid phantom for the different firing orders – 20 iterations CGLS with no additional regularisation

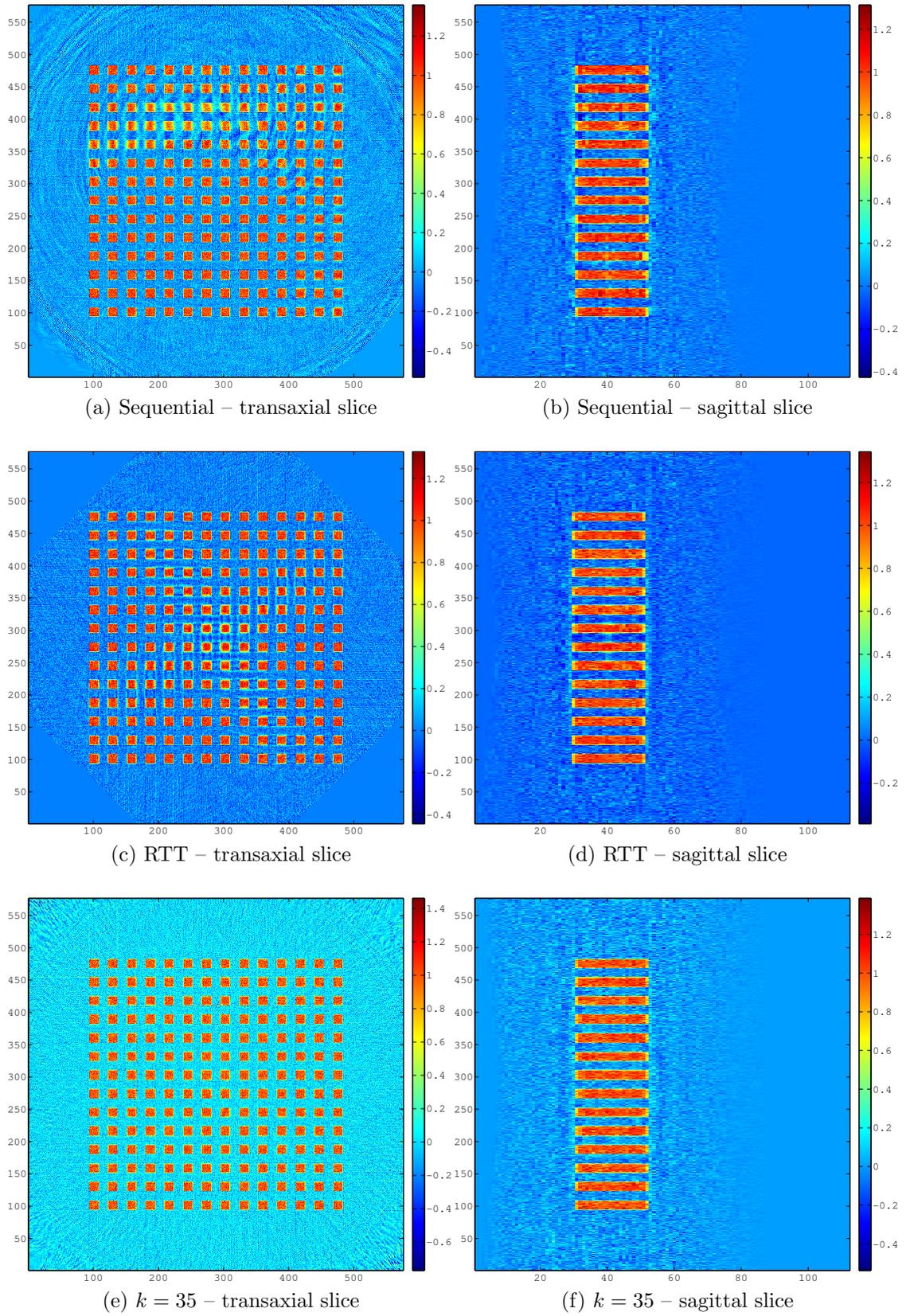


Figure 7.2: Reconstruction of the multi-cuboid phantom for the different firing orders – 50 iterations CGLS with no additional regularisation

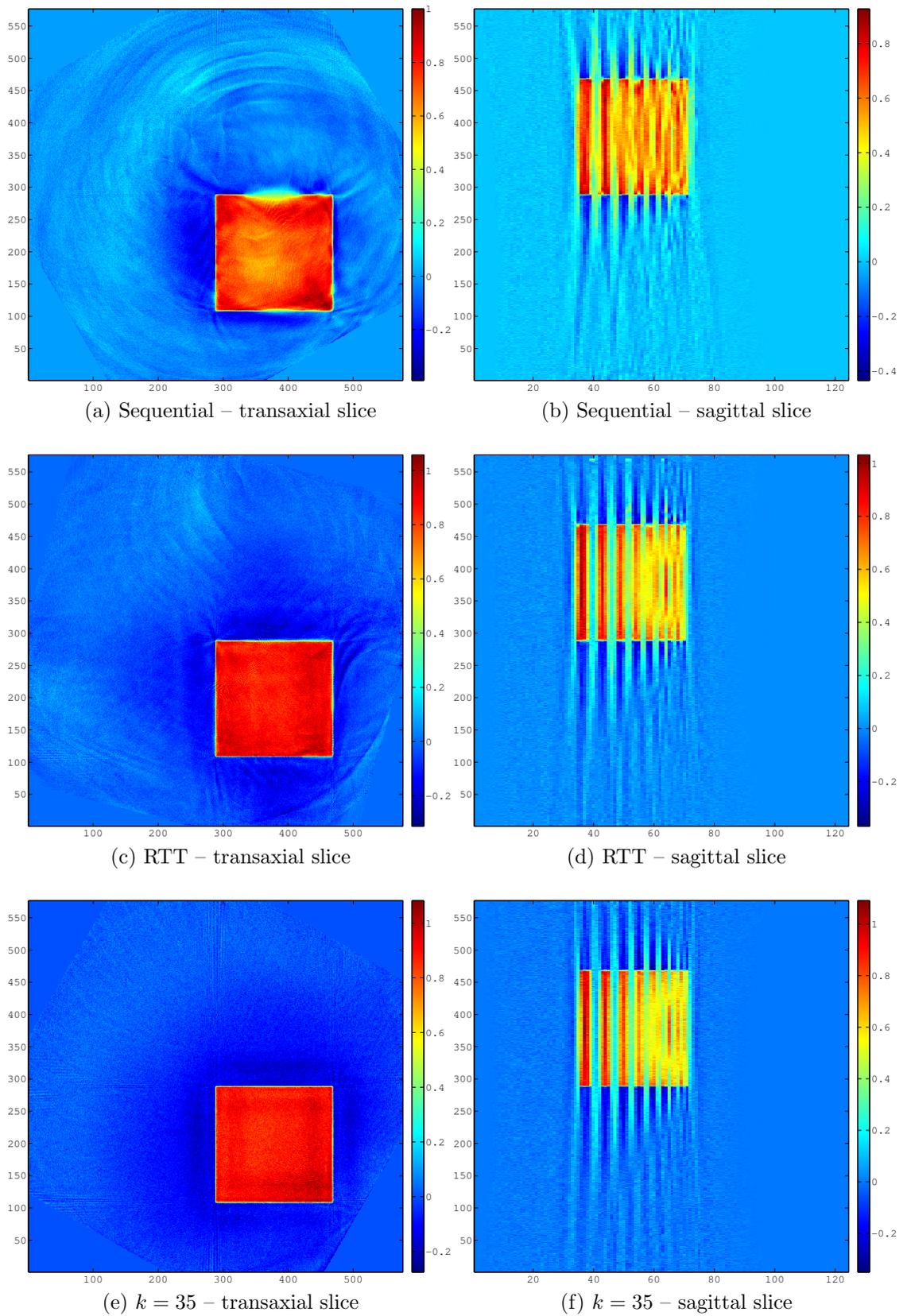


Figure 7.3: Reconstruction of the  $z$  resolution phantom for the different firing orders – 20 iterations CGLS with no additional regularisation

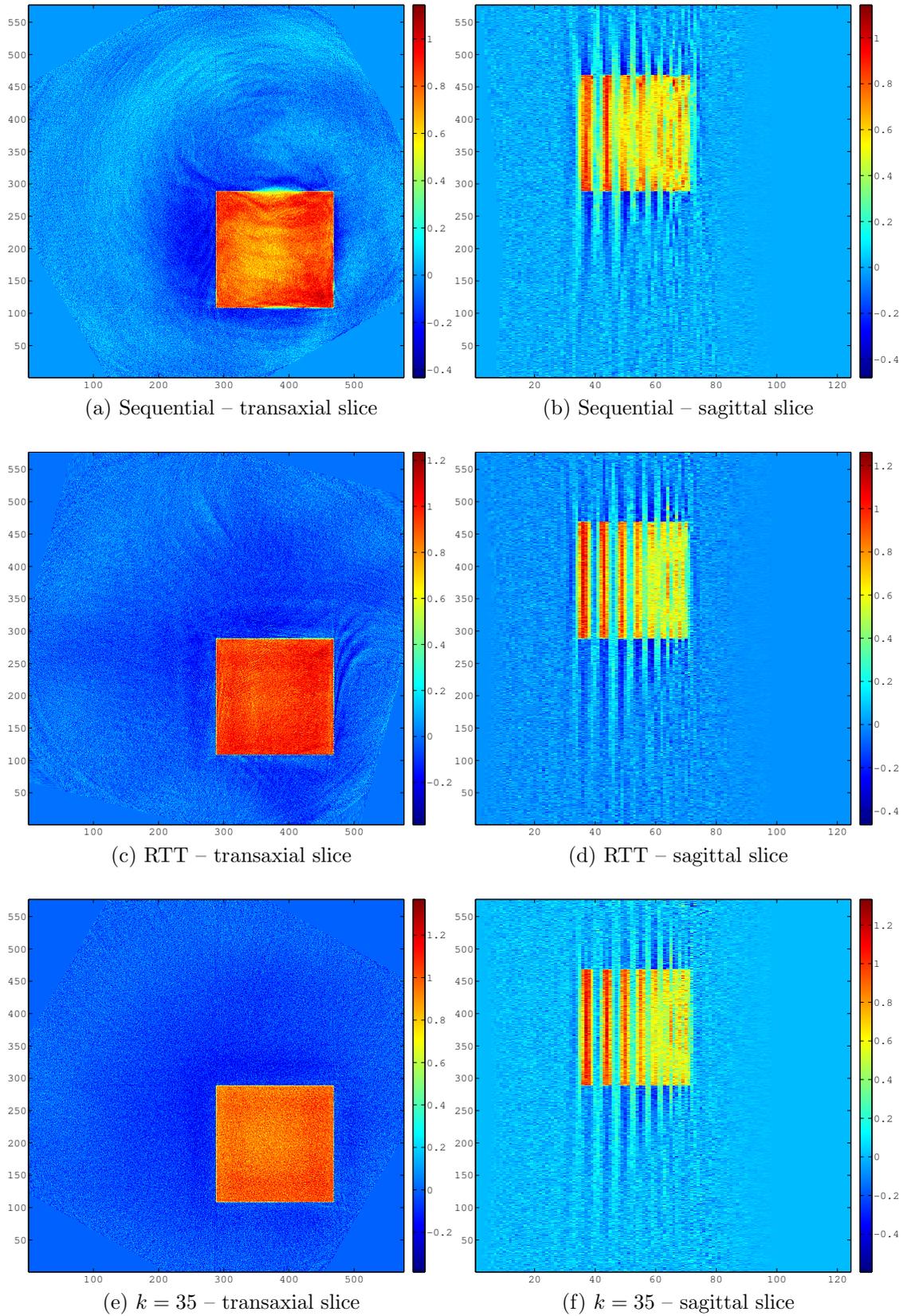


Figure 7.4: Reconstruction of the  $z$  resolution phantom for the different firing orders – 50 iterations CGLS with no additional regularisation

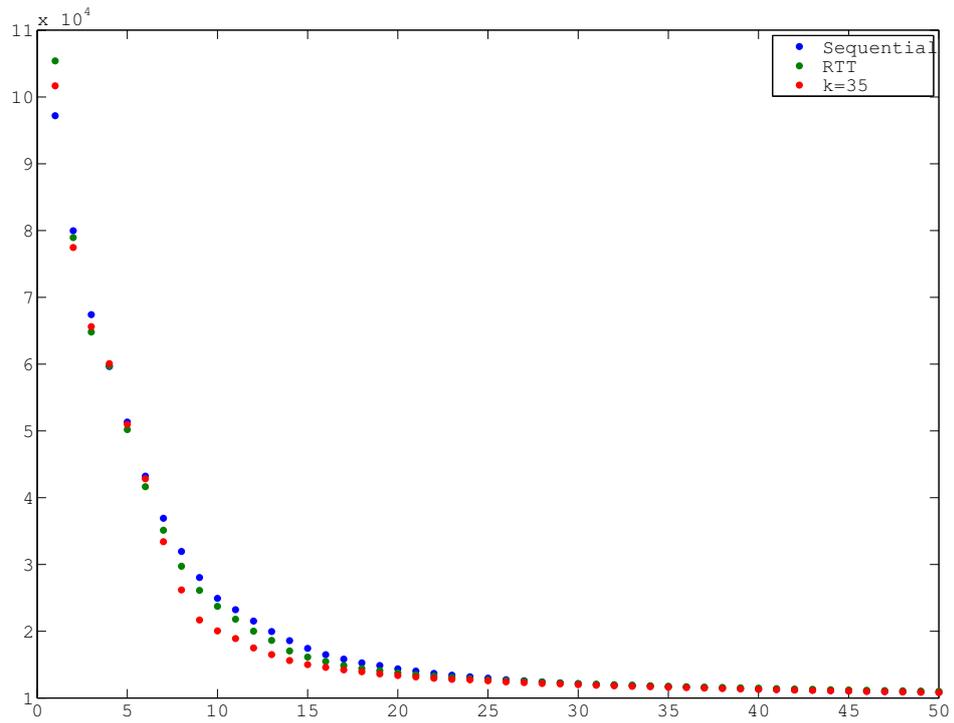


Figure 7.5: 2-norm of the residual at each iteration of the CGLS algorithm for the multi-cuboid phantom

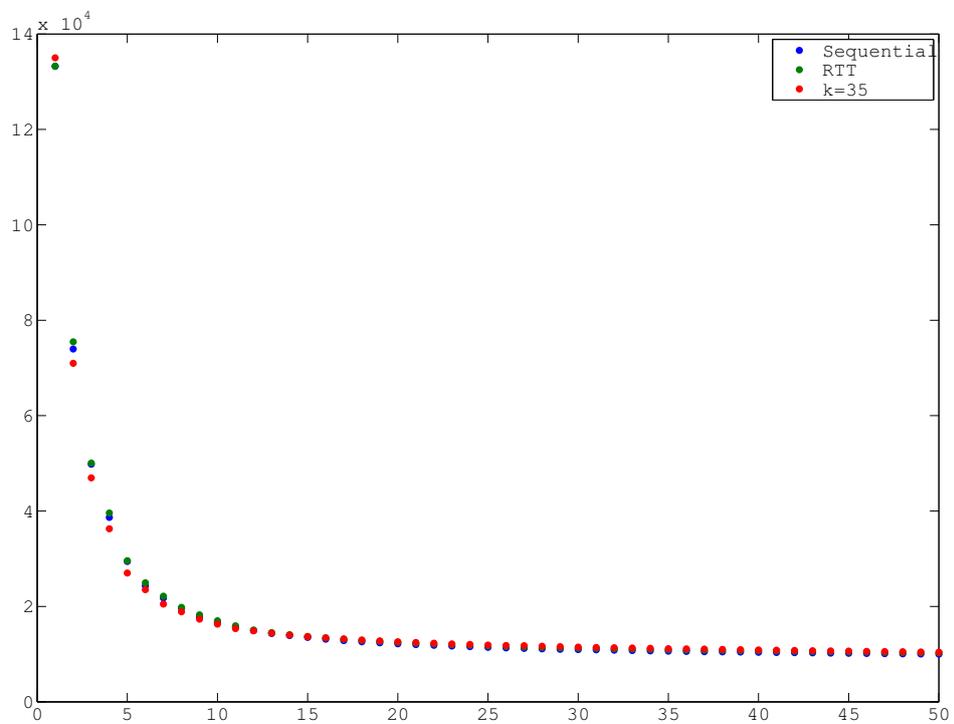


Figure 7.6: 2-norm of the residual at each iteration of the CGLS algorithm for the resolution test phantom

more data. Data from the detector integrators is passed to the online reconstruction hardware in an HD video format. The data provided by Rapiscan for our analysis has been converted to the `.pfm` format, which is easily read by MATLAB. We load the RTT data into a four-dimensional MATLAB array, with the order of the dimensions being source number, detector number, detector ring number and then revolution number, and apply calibration before performing reconstruction. For a given set of RTT data, we let the total number of revolutions be  $N_{rev}$ .

Our model of the RTT80 machine for the phantom reconstructions and matrix calculation adopted the convention that the source  $z$ -coordinates increased with the belt travel, so that the belt effectively moves in the direction of the negative  $z$ -axis. The Rapiscan convention is the other way around; therefore, to get the correct reconstruction, we reverse the order of the sources and revolutions in the RTT data and use the reverse firing order for the reconstruction matrix.

## 7.2.2 Data Preprocessing

There are several procedures that can be applied to the raw RTT data as a pre-processing step; these include corrections for physical effects such as scatter, beam hardening and afterglow, caused by the sources' tendency to carry on emitting x-rays for a short time after switching off. However, these corrections will not be discussed here.

The main calibration procedure is known as normalisation; the purpose of this is to correct for output differences between individual sources and differences in the responses of the detectors and associated electronics. The basic procedure for achieving this uses so-called *light* and *dark* images. The light image, denoted by  $\mathbf{l}$ , is what we get by firing the sources in order with the RTT80 machine empty and averaging over all revolutions measured; an example of this is shown in figure 7.7. The sinusoidal patterns in this image are caused by the tunnel and conveyor belt of the RTT80 machine.

The dark image, denoted by  $\mathbf{d}$ , as its name implies, is the result of measuring all

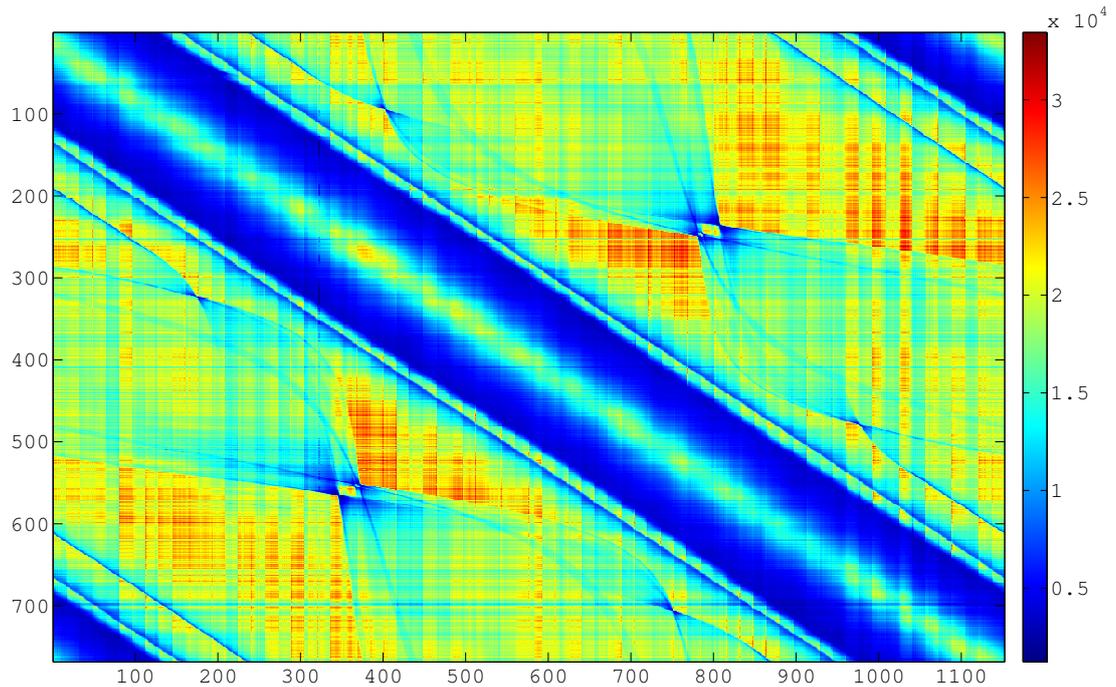


Figure 7.7: Example RTT80 calibration light image

detectors for the sources in sequence, but without actually firing the sources. In this case, in addition to averaging over the revolutions measured, we also calculate the average for each of the two integrators; an example of this is shown in figure 7.8. It should be noted that in figures 7.7 and 7.8 the rows (representing each source) have been permuted according to the inverse firing order, in order to make the images clearer; the firing order used was  $k = 35$ .

The light and dark images give calibration coefficient for all source-detector pairs over one revolution. If we let  $\mathbf{x}$  be the raw, uncalibrated data for one revolution, then the calibrated data  $\mathbf{x}_{\text{cal}}$  is obtained from by

$$\mathbf{x}_{\text{cal}} = \frac{\mathbf{x} - \mathbf{d}}{\mathbf{1} - \mathbf{d}}. \quad (7.1)$$

Figure 7.9 shows uncalibrated and calibrated sinograms for detector ring 1 of a revolution in the middle of the scan of a test bag, scanned with the  $k = 35$  firing order. Again, the order of the projections has been permuted by the inverse firing order to make things easier to see. The colour map for the calibrated data has been windowed to the range  $[0, 1]$ .

The raw uncalibrated data are given as 16-bit integers, directly from the detector

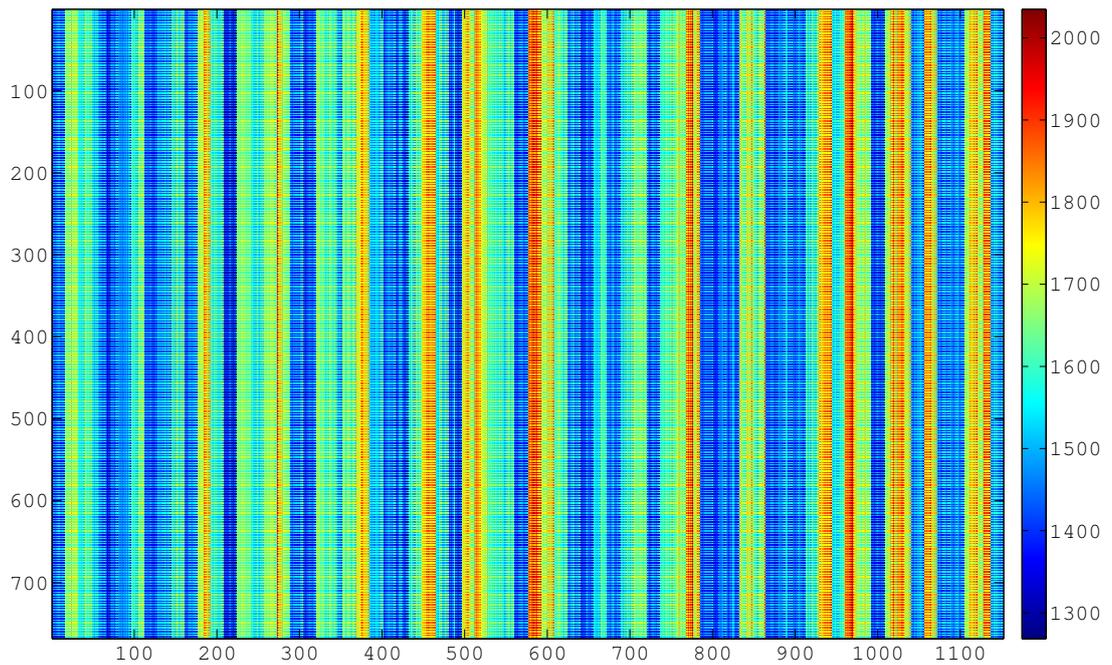
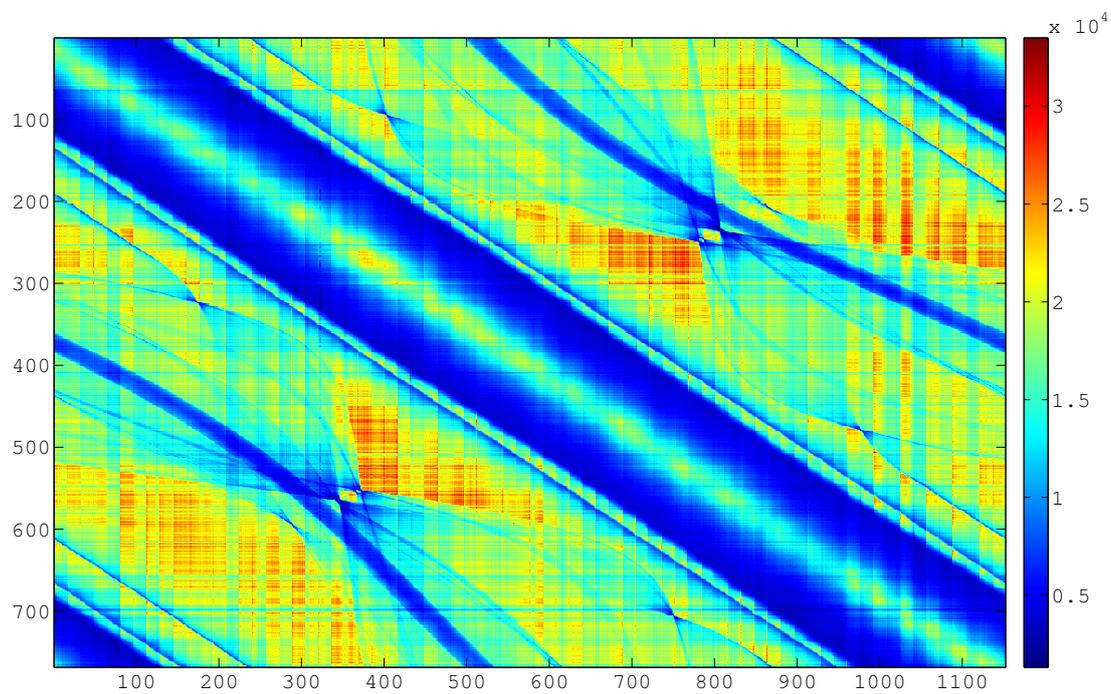


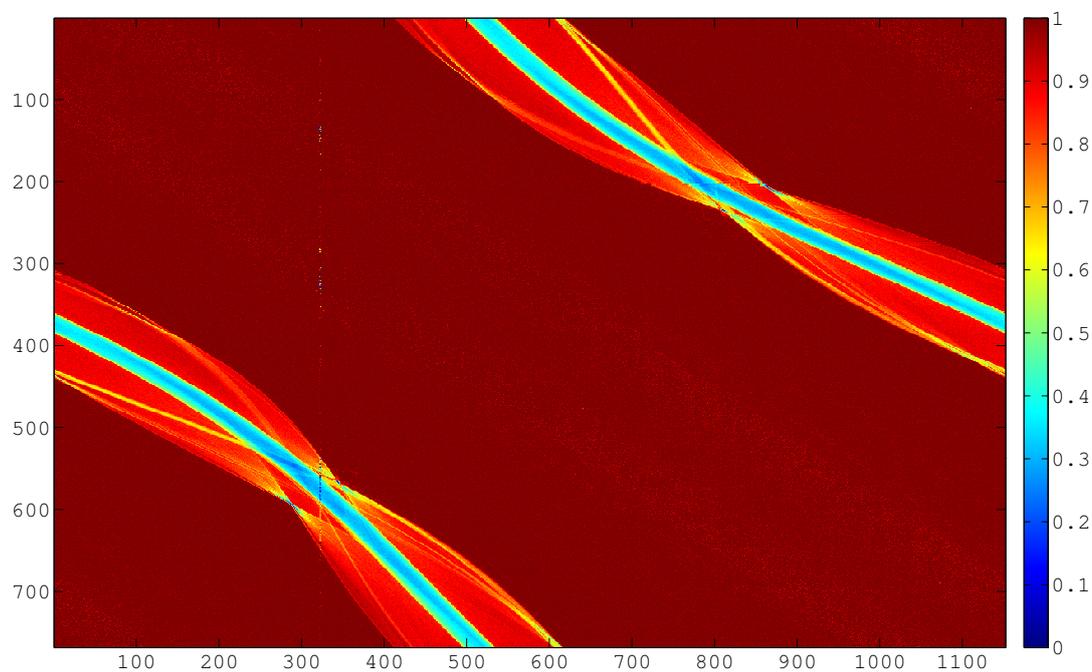
Figure 7.8: Example RTT80 calibration dark image

integrators' analogue-to-digital conversion. Theoretically, after the calibration procedure, all values in the calibrated data, representing relative intensity, should be between 0 and 1. However, in practice this is rarely the case; although *most* of the calibrated data do indeed lie between 0 and 1, we get some negative and also some extreme positive values. This is mainly due to the presence of faulty detectors; an example of this is shown by the vertical streak in figure 7.9b.

Since the calibrated data represent relative intensity, we must take the logarithm before performing reconstruction. The choice of what to do with the extreme valued data before taking the logarithm, in order to avoid creating negative attenuation values, is fairly arbitrary. It should be noted that most of the extreme values are for source-detector pairs outside the active detector regions, so these can be ignored. We usually still find that the majority of the remaining data lie in some range  $[0-\epsilon_1, 1+\epsilon_2]$ , for some small values  $\epsilon_1, \epsilon_2$ . Any remaining extreme values are then clipped to lie in this range, which is then scaled to  $[0, 1]$ .



(a) Raw uncalibrated data



(b) Calibrated data

Figure 7.9: Example uncalibrated and calibrated RTT80 sinograms

### 7.2.3 Correcting for Defective Sources and Detectors

The most severe form of source and detector faults are respectively non-emitting sources and detectors producing saturated output. The presence of the effects of both of these in the data causes severe streak artefacts in the reconstruction. If a non-emitting source is identified during the machine setup process, then that source is disabled and a neighbouring one used instead. This will obviously cause inconsistency in the data but is better than having no data at all for that source. In theory, we could simply update the firing order and corresponding change in the reconstruction matrix to take account of the change in sources, but this would destroy the rotational invariance property of the firing order and is therefore not done in our method.

In order to correct for faulty detectors, the procedure currently employed for the analytical algorithm is to interpolate the data from neighbouring detectors. However, with algebraic reconstruction, there is no need to do this; we can simply set the data from such sources and the corresponding rows of the matrix  $A$  to zero. This can be done without affecting rotational invariance by zeroing out the relevant elements of the matrix-vector product  $A\mathbf{x}$ , within the CGLS algorithm.

### 7.2.4 Approximate Reconstruction with the RTT Firing Order

Although the RTT firing order lacks strict rotational invariance, we can actually approximate the order-4 rotationally invariant 4-helix firing order by some simple permutations of the data. The period of the 4-helix is 4 revolutions, so letting  $r$  represent the revolution number, for each source  $s_i$ , we wish to permute the source numbers as follows:

$$\begin{aligned}
 r - 1 \equiv 0 \pmod{4} & \quad i \mapsto i, \\
 r - 1 \equiv 1 \pmod{4} & \quad i \mapsto (i + 192) \pmod{768}, \\
 r - 1 \equiv 2 \pmod{4} & \quad i \mapsto (i + 384) \pmod{768}, \\
 r - 1 \equiv 3 \pmod{4} & \quad i \mapsto (i + 576) \pmod{768}.
 \end{aligned}$$

This has the effect of lining up the data source positions with those of the 4-helix firing order. Of course, the  $z$ -coordinates of these sources will not be correct, since we cannot change the order in which they were actually fired. However, the  $z$ -axis errors are actually very small; the greatest difference between the actual and assumed firing orders is 3 successive projections, giving a maximum  $z$ -axis difference of  $3 \times (\pi_z/768) \approx 0.065\text{mm}$ .

### 7.2.5 Restrictions on the Firing Order

We hope to be able to compare between firing orders using reconstructions from real data. Currently, there are some limitations imposed on the firing order by the design of the RTT80 machine; the first of these is due to the controller electronics.

**Limitation 1.** The period of the firing order must be one revolution, equal to 768 projections.

The second limitation is due to the type of x-ray source used and exists to give the sources sufficient time to cool between operations.

**Limitation 2.** For every set of three consecutive projections (in time), sources from three separate blocks must be used.

Note that this not only includes sources within the period of the firing order, but also when moving from the end of one period to the beginning of the next. As a consequence of this, for an order-1 rotationally invariant firing order of the type given in (6.1), we get the following restrictions on the value of  $k$ :

$$32 \leq k \leq 368 \quad \text{and} \quad 400 \leq k \leq 736, \quad (7.2)$$

since values outside these ranges would lead to sources in the same block being fired after only 1 or 2 time steps. Clearly, this limitation disallows the use of the sequential firing order, for example.

### 7.3 Results with Real Data

Figure 7.10 shows the results of reconstruction of a test bag using the RTT firing order; the scan for this was made during machine testing at Manchester Airport. The reconstruction shows numerous artefacts; these could be for a variety of reasons. Firstly, the machine used had several faulty sources which have been substituted by other sources; a similar procedure had been used for the faulty detectors, but using interpolation. If the raw data had been given and the faulty sources and detectors identified, we would expect algebraic reconstruction to give better results with the faulty sources and detectors omitted.

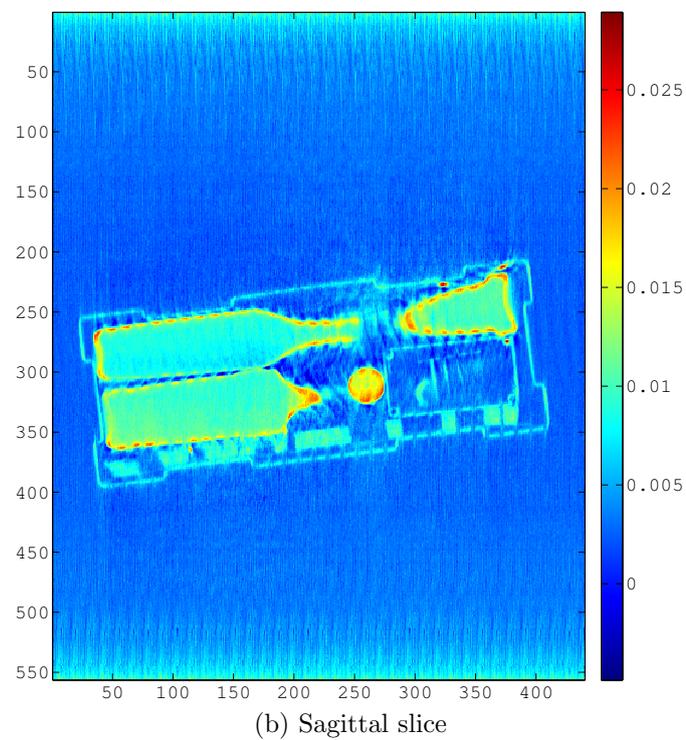
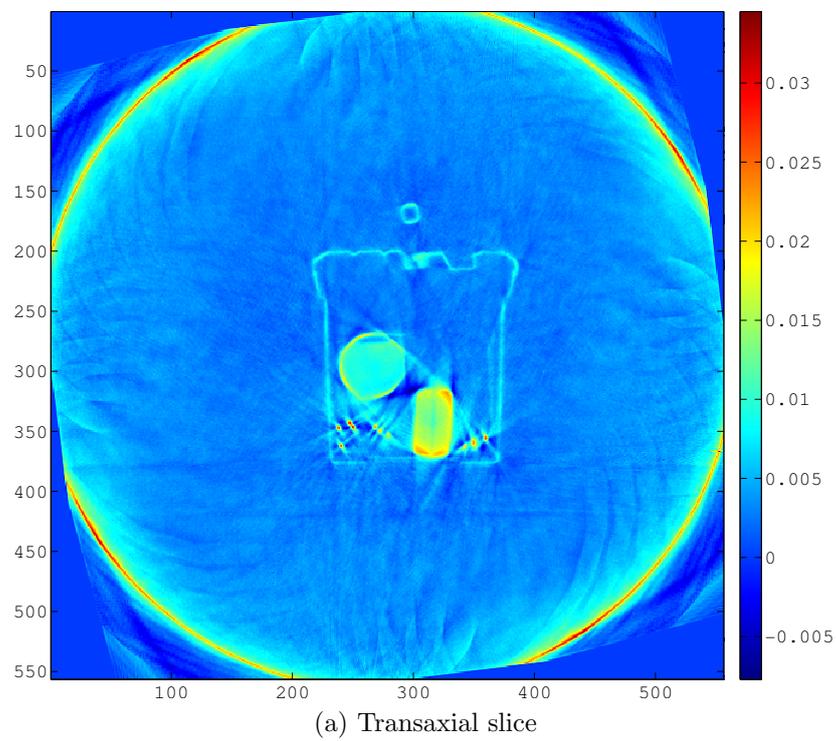


Figure 7.10: Reconstruction of a test bag from real data scanned using the RTT firing order and reconstructed using the approximation of section 7.2.4

# Chapter 8

## Conclusions and Future Work

We give in this chapter the conclusions of this investigation into reconstruction from switched source, offset detector x-ray CT machines. However, some aspects of the work present new and open problems; we therefore give some ideas on where these investigations should go next.

### 8.1 Conclusions

In chapter 2 we introduced a new mathematical continuum model of the geometry of a CT machine, applicable to the case of the RTT system; that of sources covering the surface of a cylinder in  $\mathbb{R}^3$ . This allowed us to prove the existence of a unique solution to the problem of three-dimensional reconstruction from the RTT system, and to show that the problem can be reduced to that of the inversion of the three-dimensional Radon transform with limited angle data. This also allowed us to comment on the detection of singularities of the function  $f$ , and to predict that reconstructions will typically suffer from limited angle type artefacts around edges parallel to transaxial planes.

In chapter 4 we showed that the Conjugate Gradient Least Squares algorithm can be applied to the problem of two-dimensional fan-beam CT reconstruction, and gives good results even with poorly distributed angular sampling, as with the RTT20 system. We also showed that an additional Tikhonov regularisation term gives the

method well-defined convergence behaviour, as opposed to just letting the number of iterations perform the role of regularisation parameter.

In chapter 5 we introduced and formalised the important new concept of the firing order for switched source CT machines. Properties of the firing order and their effect on the conditioning of the problem were then investigated further in chapter 6. The effect of the firing order on the sampling of source points on the surface of the cylinder, and the distribution of ray angles within the reconstruction volume allowed us to heuristically predict which firing orders might be in some sense optimal. Through some novel visualisations of these effects we were then able to perform a comparison between four different firing orders, and predict which would give the best and worst results. These predictions were then confirmed with a thorough numerical comparison between the four different firing orders, analysing the condition number and SVD of the reconstruction matrix  $A$ . The main results of this were that, for this type of CT machine, the sequential firing order, giving a standard helical source trajectory, is the worst choice of firing order, and that a firing order giving an even sampling lattice on the surface of a cylinder gives much better results.

By exploiting symmetries in the system and the rotational invariance property of certain firing orders, we developed a way of applying the CGLS algorithm to the RTT80 system using stored matrix coefficients, allowing us to quickly and easily perform three-dimensional reconstruction for a wide variety of firing orders. In chapter 7, this was used to compare reconstructions of two simple phantoms, confirming the results of the numerical comparisons of chapter 6, and also providing some validation of the model of chapter 2.

## 8.2 Future Work

### 8.2.1 Improved Modelling of the Projection Process

It was noted in chapter 4 that the model of the projection process used here is by no means the only one; since the reconstruction method of chapter 5 uses stored

matrix coefficients, we should be able to introduce more complicated modelling of the projection process, as long as this does not reduce the sparsity of the matrix  $A$  by too much. Rather than modelling the sources and detectors simply as points, and the rays as lines, it would be interesting to see the effect on real data reconstruction of modelling the physical size and properties of the sources and detectors.

It would also be interesting to change the basis functions used to discretise the continuous function  $f$  that we are imaging. In terms of sampling theory, the simple voxel grid is less than ideal; the use of spherically symmetric Kaiser-Bessel window functions as alternatives to voxels was presented by Lewitt in [38] and [39]. Practical aspects of applying this to image reconstruction problems are dealt with in [41] and [42]. These basis functions have been used by Carvalho, Herman and Matej in their work on algebraic reconstruction from helical cone-beam CT in a medical context, for instance in [5] and [6].

## 8.2.2 Reconstruction from Real Data

Ideally, to further validate our results on the optimisation of the firing order, we would like to perform further reconstructions from real RTT80 data. Although, due to constraints imposed by the design of the RTT80 machine, it is impossible to test the sequential firing order with real data, it would still be enlightening to perform a comparison between the RTT and  $k = 35$  firing orders.

## 8.2.3 Analytical Inversion of the New Model

It would be interesting to see if it is possible to apply the analytical inversion method of section 2.5.6 to the practical problem of RTT80 reconstruction, and to assess the practicality of applying this to reconstruction from real data.

## 8.2.4 Two-Dimensional Imaging of Dynamic Processes

As introduced in section 1.2.4, a potentially very interesting possibility with the RTT system is the ability to image a two-dimensional slice through time-dependent,

dynamic processes. It has been proposed to apply this to the experimental validation of some theoretical results in granular flow, using small glass beads as our granular material. Although the RTT20 machine has now been acquired by the University, it is not yet clear whether the experiments will be performed in this, or an RTT80 machine owned by Rapiscan; the theory in both cases is the same.

This presents us with the problem of improving the temporal resolution of the system. An initial thought is to choose a firing order such that every full revolution can be divided into an integer number of more sparsely sampled sub-revolutions, allowing us to trade temporal against spatial resolution in the reconstruction process. For instance, for RTT80, we could choose the firing order

$$1, 9, \dots, 761, 2, 10, \dots, 762, \dots, 8, 16, \dots, 768, \quad (8.1)$$

giving 8 sub-revolutions sampled with an angular frequency one eighth that of the maximum.

Another thought is to apply Kalman filtering to the projections in order to get better estimates of the state of the system at particular points in time. If we think of each reconstructed slice as a frame of a film, then ideally, all projections for a particular frame would be taken at the same time. Of course, this is not the case in reality, as each frame will be reconstructed from projections taken at a range of time values. For fast moving objects this will introduce motion artefacts into the reconstructions; it is hoped that the Kalman filter can be used to reduce such artefacts. For an introduction to the Kalman filter, see, for example [69].

# Bibliography

- [1] Isaac Amidror. Scattered data interpolation methods for electronic imaging systems: a survey. *Journal of Electronic Imaging*, 11(2):157–176, April 2002.
- [2] Mario Bertero and Patrizia Boccacci. *Introduction to Inverse Problems in Imaging*. Institute of Physics Publishing, 1998.
- [3] M. M. Betcke and W. R. B. Lionheart. Multisheet surface rebinning methods for cone beam ct. To appear, 2010.
- [4] H. Bruder, M. Kachelriess, S. Schaller, K. Stierstorfer, and T. Flohr. Single slice rebinning reconstruction in spiral cone-beam computed tomography. *IEEE Transactions on Medical Imaging*, 19(9), September 2000.
- [5] B. M. Carvalho, G. T. Herman, and S. Matej. Art for helical cone-beam ct reconstruction. In *3D-2001 – The Sixth International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, pages 249–252, 2001.
- [6] Bruno M. Carvalho and Gabor T. Herman. Helical ct reconstruction from wide cone-beam angle data using art. *Computer Graphics and Image Processing, Brazilian Symposium on*, 0:363, 2003.
- [7] G. Cimmino. Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari. *La Ricerca Scientifica Roma XVI (Series II)*, Anno IX 1:326–333, 1938.
- [8] A. M. Cormack. Representation of a function by its line integrals, with some radiological applications. *Journal of Applied Physics*, 34(9):2722–2727, 1963.

- [9] P.P.B Eggermont, G.T. Herman, and A. Lent. Iterative algorithms for large partitioned linear systems, with applications to image reconstruction. *Linear Algebra and its Applications*, 40:37–67, October 1981.
- [10] L. A. Feldkamp, L. C. Davis, and J. W. Kress. Practical cone-beam algorithm. *J. Opt. Soc. Am. A*, 1(6):612–619, Jun 1984.
- [11] I. M. Gel'fand and A. B. Goncharov. Recovery of a compactly supported function starting from its integrals over lines intersecting a given set of points in space – english translation. *Soviet Math Dokl.*, 34:373–376, 1987.
- [12] Peter Gilbert. Iterative methods for the three-dimensional reconstruction of an object from projections. *Journal of Theoretical Biology*, 36(1):105 – 117, 1972.
- [13] Richard Gordon, Robert Bender, and Gabor T. Herman. Algebraic reconstruction techniques (art) for three-dimensional electron microscopy and x-ray photography. *Journal of Theoretical Biology*, 29(3):471 – 481, 1970.
- [14] Pierre Grangeat. *Analyse d'un système d'imagerie 3D par reconstruction à partir de radiographies X en géométrie conique*. PhD thesis, Ecole Nationale Supérieure des Telecommunications Paris, 1987.
- [15] Pierre Grangeat. Mathematical framework of cone beam 3d reconstruction via the first derivative of the radon transform. In Gabor Herman, Alfred Louis, and Frank Natterer, editors, *Mathematical Methods in Tomography*, volume 1497 of *Lecture Notes in Mathematics*, pages 66–97. Springer Berlin / Heidelberg, 1991.
- [16] W. Hager. Condition estimates. *SIAM Journal on Scientific and Statistical Computing*, 5(2):311–316, 1984.
- [17] C. Hamaker, K. T. Smith, D. C. Solmon, and S. L. Wagner. The divergent beam x-ray transform. *Rocky Mountain Journal of Mathematics*, 10:253–283, 1980.
- [18] Dianwei Han and Jun Zhang. A comparison of two algorithms for predicting the condition number. In *Proceedings of the Sixth International Conference on*

- Machine Learning and Applications*, ICMLA '07, pages 223–228, Washington, DC, USA, 2007. IEEE Computer Society.
- [19] Per Christian Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*. Society for Industrial and Applied Mathematics, 1998.
- [20] Per Christian Hansen. Regularization tools - a matlab package for analysis and solution of discrete ill-posed problems - version 3.0 for matlab 5.2. *Numer. Algorithms*, 6:1–35, 1998.
- [21] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, December 1952.
- [22] L. Hormander. *Linear Partial Differential Operators*. Springer, 1964.
- [23] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1–3):185 – 203, 1981.
- [24] G. N. Hounsfield. Computerized transverse axial scanning (tomography): Part 1. Description of system. *Br J Radiol*, 46(552):1016–1022, 1973.
- [25] F. Jacobs, E. Sundermann, B. De Sutter, and I. Lemahieu. A fast algorithm to calculate the exact radiological path through a pixel or voxel space. *J. Comput. Inform. Technol.*, 6:89–94, 1998.
- [26] M. Kachelriess, T. Fuchs, S. Schaller, and W.A. Kalender. Advanced single-slice rebinning for tilted spiral cone-beam ct. *Medical Physics*, 28(6):1033–1041, June 2001.
- [27] M. Kachelriess, S. Schaller, and W.A. Kalender. Advanced single-slice rebinning in cone-beam spiral ct. *Medical Physics*, 27(4):754–772, April 2000.

- [28] S Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bulletin International de l'Académie Polonaise des Sciences et des Lettres*, A35:355–357, 1937.
- [29] A.C. Kak and M. Slaney. *Principles of computerized tomographic imaging*. IEEE Service Center, Piscataway, NJ, 1988.
- [30] W.A. Kalender. *Computed Tomography: Fundamentals, System Technology, Image Quality, Applications, 2nd Edition*. Publicis MCD Verlag, 2005.
- [31] Willi A Kalender. X-ray computed tomography. *Physics in Medicine and Biology*, 51(13):R29, 2006.
- [32] S. Kawata and O. Nalcioglu. Constrained iterative reconstruction by the conjugate gradient method. *Medical Imaging, IEEE Transactions on*, 4(2):65–71, June 1985.
- [33] C. S. Kenney, A. J. Laub, and M. S. Reese. Statistical condition estimation for linear systems. *SIAM J. Sci. Comput.*, 19:566–583, March 1998.
- [34] H. Kostler, M. Prummer, U. Rude, and J. Hornegger. Adaptive variational sinogram interpolation of sparsely sampled ct data. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 778–781, 2006.
- [35] L. Landweber. An iteration formula for fredholm integral equations of the first kind. *American Journal of Mathematics*, 73(3):pp. 615–624, 1951.
- [36] J. M. Lees and R. S. Crosson. Bayesian art versus conjugate gradientf methods in tomographic seismic imaging: An application at mount st. helens, washington. *Lecture Notes-Monograph Series*, 20:pp. 186–208, 1991.
- [37] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods*. Society for Industrial and Applied Mathematics, 1997.

- [38] R M Lewitt. Alternatives to voxels for image representation in iterative reconstruction algorithms. *Physics in Medicine and Biology*, 37(3):705, 1992.
- [39] Robert M. Lewitt. Multidimensional digital image representations using generalized kaiser-bessel window functions. *J. Opt. Soc. Am. A*, 7(10):1834–1846, Oct 1990.
- [40] Y. Lu, W. Wang, S. Chen, Y. Xie, J. Qin, Wai-Man Pang, and Pheng-Ann Heng. Accelerating algebraic reconstruction using cuda-enabled gpu. In *Computer Graphics, Imaging and Visualization, 2009. CGIV '09. Sixth International Conference on*, pages 480–485, Aug 2009.
- [41] S. Matej and R.M. Lewitt. Efficient 3d grids for image reconstruction using spherically-symmetric volume elements. *Nuclear Science, IEEE Transactions on*, 42(4):1361–1370, August 1995.
- [42] S. Matej and R.M. Lewitt. Practical considerations for 3-d image reconstruction using spherically symmetric volume elements. *Medical Imaging, IEEE Transactions on*, 15(1):68–78, February 1996.
- [43] Frederic Noo Michel Defrise and Hiroyuki Kudo. Rebinning-based algorithms for helical cone-beam ct. *Physics in Medicine and Biology*, 46(11), 2001.
- [44] K. Mueller and R. Yagel. Rapid 3-d cone-beam reconstruction with the simultaneous algebraic reconstruction technique (sart) using 2-d texture mapping hardware. *Medical Imaging, IEEE Transactions on*, 19(12):1227–1237, Dec 2000.
- [45] K. Mueller, R. Yagel, and J.J. Wheller. Anti-aliased three-dimensional cone-beam reconstruction of low-contrast objects with algebraic methods. *Medical Imaging, IEEE Transactions on*, 18(6):519–537, June 1999.
- [46] Klaus Mueller, Roni Yagel, and John J. Wheller. Fast implementations of algebraic methods for three-dimensional reconstruction from cone-beam data. *IEEE Trans. Med. Imag., vol*, 18:538–548, 1999.

- [47] Frank Natterer. *The Mathematics of Computerized Tomography*. Society for Industrial and Applied Mathematics, 2001.
- [48] Frank Natterer and Frank Wubbeling. *Mathematical Methods in Image Reconstruction*. Society for Industrial and Applied Mathematics, 2001.
- [49] Frederic Noo, Michel Defrise, and Rolf Clackdoyle. Single-slice rebinning method for helical cone-beam ct. *Physics in Medicine and Biology*, 44(2), 1999.
- [50] Openmp Architecture Review Board, <http://openmp.org/wp/openmp-specifications/>. *Openmp Application Program Interface*.
- [51] Xiaochuan Pan, Emil Y Sidky, and Michael Vannier. Why do commercial ct scanners still employ traditional, filtered back-projection for image reconstruction? *Inverse Problems*, 25(12):123009, 2009.
- [52] Elena Loli Piccolomini and Fabiana Zama. The conjugate gradient regularization method in computed tomography problems. *Applied Mathematics and Computation*, 102(1):87 – 99, 1999.
- [53] E. T. Quinto. Singularities of the x-ray transform and limited data tomography in  $\mathbb{R}^2$  and  $\mathbb{R}^3$ . *SIAM Journal on Mathematical Analysis*, 24(5):1215–1225, 1993.
- [54] E. T. Quinto. An introduction to x-ray tomography and radon transforms. In G. Olafsson and E. T. Quinto, editors, *The Radon Transform, Inverse Problems, and Tomography*, 2005.
- [55] J Radon. Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten. *Akad. Wiss.*, 69:262–277, 1917.
- [56] A. G. Ramm. Inversion of limited-angle tomographic data. *Computers and Mathematics with Applications*, 22(4/5):101–111, 1991.
- [57] A.G. Ramm. Inversion of the radon transform with incomplete data. *Mathematical Methods in the Applied Sciences*, 15:159–166, 1992.

- [58] A.G Ramm and A.I Katsevich. *The Radon Transform and Local Tomography*. CRC Press, 1996.
- [59] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2003.
- [60] J. A. Scales. Tomographic inversion via the conjugate gradient method. *Geophysics*, 52(2):179–185, 1987.
- [61] Jonathan R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.
- [62] R. L. Siddon. Fast calculation of the exact radiological path for a 3-dimensional ct array. *Medical Physics*, 12(2):252–255, 1985.
- [63] B. D. Smith. Image reconstruction from cone-beam projections: Necessary and sufficient conditions and reconstruction methods. *Medical Imaging, IEEE Transactions on*, 4(1):14–25, march 1985.
- [64] D. Szotten. *Limited Data Problems in X-Ray and Polarized Light Tomography*. PhD thesis, University of Manchester, 2010.
- [65] H. Turbell. *Cone-Beam Reconstruction Using Filtered Backprojection*. PhD thesis, Linköping University, 2001.
- [66] Heang K. Tuy. An inversion formula for cone-beam reconstruction. *SIAM Journal on Applied Mathematics*, 43(3):pp. 546–552, 1983.
- [67] A. van der Sluis and H. A. van der Vorst. The rate of convergence of conjugate gradients. *Numerische Mathematik*, 48:543–560, 1986. 10.1007/BF01389450.
- [68] Max A. Viergever. Introduction to discrete reconstruction methods in medical imaging. In M. A. Viergever and A. Todd-Pokropek, editors, *Mathematics and Computer Science in Medical Imaging*, 1988.

- [69] Greg Welch and Gary Bishop. An introduction to the kalman filter. *University of North Carolina at Chapel Hill*, 1995.
- [70] Fang Xu and K. Mueller. Accelerating popular tomographic reconstruction algorithms on commodity pc graphics hardware. *Nuclear Science, IEEE Transactions on*, 52(3):654 – 663, June 2005.
- [71] Shuting Xu and Jun Zhang. A new data mining approach to predicting matrix condition numbers. *Communications in Information and Systems*, 4(4):325–340, 2004.
- [72] K. Yosida. *Functional Analysis*. Springer, 1980.

# Appendix A

## Elementary Distribution Theory

We give here a summary of the elementary definitions of the spaces of test functions and distributions, based mainly on the notation of [58], section 14.2. Before introducing the various spaces, we first introduce the concept of a *multiindex*, defined as follows:

**Definition A.1.** An  $n$ -dimensional *multiindex*  $\alpha$  is an  $n$ -tuple of non-negative integers  $\alpha_i$ , where  $1 \leq i \leq n$ ; so

$$\alpha = (\alpha_1, \dots, \alpha_n). \quad (\text{A.1})$$

For multiindex  $\alpha$ , we define the absolute value  $|\alpha|$  and partial derivative  $\partial^\alpha$  by

$$|\alpha| = \alpha_1 + \dots + \alpha_n, \quad (\text{A.2})$$

$$\partial^\alpha = \frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \cdots \frac{\partial^{\alpha_n}}{\partial x_n^{\alpha_n}}. \quad (\text{A.3})$$

We now introduce the spaces of *test functions*,  $\mathcal{D}$ ,  $\mathcal{S}$  and  $\mathcal{E}$ . In the following, we take the word ‘smooth’ to mean infinitely differentiable.

**Definition A.2.** The space  $\mathcal{D} = C_0^\infty(\mathbb{R}^n)$  is the space of smooth functions with compact support. We define a topology on  $\mathcal{D}$  by defining the limit of a sequence of functions in  $\mathcal{D}$  as follows; a sequence of functions  $\phi_n \in \mathcal{D}$  converges in  $\mathcal{D}$  to zero if there exists a compact set  $K$  such that for all  $n \geq 0$ ,  $\text{supp}(\phi_n) \subset K$  and, for any integer  $m$ ,

$$\sup_{x \in K, |\alpha| \leq m} |\partial^\alpha \phi_n(x)| \rightarrow 0 \quad \text{as } n \rightarrow \infty, \quad (\text{A.4})$$

where  $\alpha$  is a multiindex.

**Definition A.3.** The space  $\mathcal{S} = \mathcal{S}(\mathbb{R}^n)$  is the space of smooth functions which decay rapidly at infinity with all their derivatives; this is known as the Schwartz space, after Laurent Schwartz. Formally, we have  $\phi \in \mathcal{S}$  if and only if the norms

$$\|\phi\|_{m,k} = \sup_{x \in \mathbb{R}^n, |\alpha| \leq k} |(1 + |x|)^m \partial^\alpha \phi(x)|, \quad (\text{A.5})$$

are finite for all  $m, k \in \mathbb{N}$ . A sequence of functions  $\phi_n \in \mathcal{S}$  converges in  $\mathcal{S}$  to zero if, for any integers  $m$  and  $k$ ,

$$\|\phi_n\|_{m,k} \rightarrow 0 \quad \text{as } n \rightarrow \infty. \quad (\text{A.6})$$

**Definition A.4.** The space  $\mathcal{E} = C^\infty(\mathbb{R}^n)$  is the space of smooth functions with with no restriction on their growth at infinity. A sequence of functions  $\phi_n \in \mathcal{E}$  converges in  $\mathcal{E}$  to zero if, for any integer  $k$  and compact set  $K \subset \mathbb{R}^n$ ,

$$\sup_{x \in K, |\alpha| \leq k} |\partial^\alpha \phi_n(x)| \rightarrow 0 \quad \text{as } n \rightarrow \infty. \quad (\text{A.7})$$

Corresponding to the spaces  $\mathcal{D}$ ,  $\mathcal{S}$  and  $\mathcal{E}$ , we have the dual spaces  $\mathcal{D}'$ ,  $\mathcal{S}'$  and  $\mathcal{E}'$ . Elements of  $\mathcal{D}'$  are known as *distributions*, elements of  $\mathcal{S}'$  are known as *tempered distributions* and elements of  $\mathcal{E}'$  are known as *distributions with compact support*. The embeddings  $\mathcal{D} \rightarrow \mathcal{S}$  and  $\mathcal{S} \rightarrow \mathcal{E}$  are continuous with dense images, so therefore, the dual mappings  $\mathcal{E}' \rightarrow \mathcal{S}'$  and  $\mathcal{S}' \rightarrow \mathcal{D}'$  are also continuous with dense images.