Problems in Parametric Motion Correction for Limited-Angle Flat-Panel Imaging



Joseph D. Field

Mansfield College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity Term 2026

At last

Statement of Originality

The author declares that this thesis is a presentation of their original research work. Except where otherwise explicitly stated, this thesis contains no material previously published or written by another person, nor material which has been accepted for the award of any other degree or diploma at any university or institution.

Abstract

Tomographic imaging methods are fundamentally constrained by the geometry and physics of data acquisition. In modern settings—particularly in limited-angle imaging, and multi-emitter devices—these constraints interact in ways that may not be fully addressed by reconstruction algorithms alone. Additionally, when motion occurs during acquisition, even simple rigid-body effects can invalidate modelling assumptions that underpin standard reconstruction approaches, leading to loss of image clarity or the appearance of systematic artefacts. This thesis develops a modelling-centred framework for algebraic reconstruction, in which the forward operator itself is treated as the primary object of analysis. With this perspective, we view reconstruction quality, numerical stability, and computational cost as consequences of particular modelling and optimisation choices for dealing with motion correction.

The first part of this thesis establishes the mathematical foundations required to support this approach. We discuss classical results from tomography, including the Radon transform and the Fourier slice theorem, to better understand the effects that limited-angle acquisition may have on both the visibility and stable reconstruction of image singularities. A flexible forward model is introduced, incorporating an algebraic reconstruction formulation for a multi-emitter device. Modelling choices and algorithmic decisions are made explicit, to ensure that later extensions—particularly full rigid-body motion and non-Cartesian discretisation—may be introduced without altering the underlying framework.

The second part of this thesis investigates the effect of object motion through the embedding of motion models directly into the forward operator. Rather than treating motion as an ad-hoc perturbation, rigid-body models are systematically analysed, generating operators with explicit continuity properties. The exploration of this forward-model viewpoint demonstrates stability with respect to small perturbations, revealing how instability enters the reconstruction problem. Numerical experiments identify objective functions which appear to be globally well-behaved, but locally nonconvex, highlighting the difficulty in determining motion parameters for even simple examples. Experimental observations lead to a diagnostic discussion on multilevel optimisation for nonsmooth functions, presented as a tool for understanding resolution-dependent error, and the interaction between model discretisation and optimisation

in nonsmooth settings. Additional discussion reframes the problem under the lens of alternative discretisations, with a focus on non-overlapping polyhedral basis elements.

The third part of this thesis introduces new algorithmic contributions, motivated by this forward operator perspective. Classical ray traversal methods for squares and cubes are discussed before being extended to non-Cartesian grids, enabling exact computation of radiological paths without interpolation. Two algorithms are developed for two-dimensional hexagonal geometries: a lightweight method exploiting the underlying geometry with low per-step computational cost, and a per-cell parallel algorithm with predictable computational complexity. These ideas are then extended to three dimensions, via the development of a new traversal algorithm through rhombic dodecahedral voxels, allowing forward operators to be efficiently constructed directly on non-cubic grids. Numerical results for all algorithms show favourable computational performance, while reconstructions show that new discretisations can be easily integrated into existing algebraic reconstruction pipelines.

Collectively, this thesis argues that many practical challenges in tomographic reconstruction are best understood through the careful modelling of the forward operator. By showing that discretisation choice is flexible, this work provides a geometry-agnostic framework for extending existing tomographic methods beyond traditional Cartesian assumptions. The results included offer both conceptual clarity and practical tools for the design of future imaging systems, and for reconstruction methodologies under non-ideal conditions.

Acknowledgements

There are many people whom I would like to thank for their assistance and support throughout this work. Those not named have certainly not been forgotten.

My supervisor, Raphael Hauser, deserves more thanks than I can properly express. His continued support, guidance, and understanding made it possible for this thesis to come to fruition. Additionally, I thank Andrew Thompson for his supervision in the early stages of this work. I am also grateful to Jared Tanner and Ander Biguri for their careful reading of this thesis and for a stimulating viva examination.

I would like to acknowledge those who financially supported this research, including the Engineering and Physical Sciences Research Council (EPSRC) and the University of Oxford Centre for Doctoral Training in Industrially Focused Mathematical Modelling (InFoMM), through grant EP/L015803/1. I also thank Adaptix for their sponsorship of this work, and my industrial supervisors, Paul Betteridge and Gil Travish, whose expertise was invaluable in bridging the gap between my mathematical goals and real-world expectations.

To Colin Please and Chris Breward, I offer my sincerest gratitude for inviting me to be a part of InFoMM. Without them, I would never have met the wonderful, kind, and interesting people across all cohorts of this CDT. This was an experience that I will never forget. I also thank Jonathan Mason, Laura O'Mahony, Amanda Guthland Elmore, Sarah Howle, Sophie Hurden, and the wider InFoMM family.

I thank the members of Cohort 3 in particular for being such an inspiring group of people to work with—I am thrilled to finally join you in the rarefied air.

I give my deepest thanks to Gabriella Papp for her encouragement, insights, and most of all her compassion. I also thank Emma Winston for helping me to understand myself, and for guiding me onto the path that led to the end of this thesis.

My final thanks are to Samantha Stephenson for pointing me in the direction of InFoMM all those years ago, and for convincing me that I was good enough to do this; to Melanie Beckerleg, one of the warmest people I've ever known, for encouraging me to continue when I didn't see how; to Katrina Moisley for her unending efforts to support me, and for helping me get over the finish line. Without you all, this thesis would not exist.

Contents

I Industrial Problem and Mathematical Model **1**

1 Introduction **3**

1.1 Historical overview of X-ray imaging 3

1.2 Acknowledgement of industrial partnership 6

1.3 Contribution of work 7

1.4 Outline of thesis 7

2 Background, forward model, and reconstruction **10**

2.1 Foundations of tomography and tomosynthesis 10

2.2 Model and geometry assumptions 27

2.3 The tomographic reconstruction problem 37

2.4 Model limitations 44

3 Modelling approaches for motion correction **49**

3.1 Overview of existing methods 49

3.2 A general parametric motion model 54

3.3 A nested optimisation formulation 60

4 Experimental and modelling considerations **62**

4.1 Simulated phantoms 63

4.2 Choice of solver 65

4.3 Sub-domain discretisation 72

4.4 Device and geometry parameters 74

II Experimental Results **77**

5 Bodies undergoing translational motion **80**

5.1 Forward model of translational motion . . . 80
5.2 Experimental results: fixed translation in 2D . . . 85
5.3 Experimental results: fixed translation in 3D . . . 92
5.4 Regularity of the inner problem . . . 95

6 Further rigid-body motion **97**
6.1 Forward model of rotational motion . . . 97
6.2 Experimental results: fixed rotation in 2D . . . 100
6.3 Experimental results: fixed rotation in 3D . . . 102
6.4 Forward model of rigid-body motion . . . 105
6.5 Experimental results: rigid-body motion . . . 107

7 Potential methodology improvements **111**
7.1 Multilevel optimisation . . . 111
7.2 Alternative reconstruction bases . . . 128

III Alternative Discretisations for Tomography **136**

8 Hexagonal grid representation **139**
8.1 Hexagonal grid model . . . 140
8.2 Ray traversal through Cartesian grids . . . 142
8.3 Ray traversal through a hexagonal grid . . . 146
8.4 Extending Siddon's algorithm to hexagonal pixels . . . 158
8.5 Numerical results . . . 173
8.6 Hexagonal image reconstruction . . . 176

9 Dodecahedral grid representation **181**
9.1 Extending Siddon's algorithm to rhombic dodecahedral voxels . . . 183
9.2 Numerical results . . . 197

10 Conclusions and extensions **202**

References **205**

List of Figures

1.1 Simple diagram of X-ray acquisition 4
1.2 Simple diagram of focal plane tomography 5

2.1 Radon transform in $\mathbb{R}^2$ 14
2.2 Fixed-angle transform in $\mathbb{R}^2$ 15
2.3 Singularities of a two-density object 19
2.4 Fourier slice theorem 21
2.5 Undersampling in Fourier domain 21
2.6 Simple diagram of fan-beam geometry 24
2.7 3D acquisition with partial coverage 26
2.8 2D acquisition with partial coverage 28
2.9 Discretisation of fan-beam emissions 29
2.10 2D pixel-ray intersection 34
2.11 Discretisation of cone-beam emissions 36

4.1 2D numerical phantoms 63
4.2 3D numerical phantom 64
4.3 Median slices of the Shepp–Logan phantom 65
4.4 2D phantoms: discrete gradient support 66
4.5 2D reconstruction: stationary, $(m/n) \approx 0.3$ 68
4.6 2D λ parameter sweep, $(m/n) \approx 0.3$ 69
4.7 2D λ parameter sweep, $(m/n) \approx 0.1$ 70
4.8 2D reconstruction: stationary, $(m/n) \approx 0.1$ 70
4.9 3D λ parameter sweep, $(m/n) \approx 0.3$ 71
4.10 3D reconstruction: stationary, $(m/n) \approx 0.3$ 72
4.11 Sub-domain restriction for moving objects 73

5.1 Continuity of the translation map 82

5.2 2D reconstruction: x-translation, $(m/n) \approx 0.3$ 86
5.3 2D reconstruction: z-translation, $(m/n) \approx 0.3$ 87
5.4 2D reconstruction: large x-translation, $(m/n) \approx 0.3$ 89
5.5 2D x-translation parameter sweep, $(m/n) \approx 0.3$ 89
5.6 2D reconstruction: 'optimal' x-translation, $(m/n) \approx 0.3$ 90
5.7 2D reconstruction: general translation, $(m/n) \approx 0.3$ 91
5.8 3D reconstruction: in-plane translation, $(m/n) \approx 0.3$ 92
5.9 3D reconstruction: out-of-plane translation, $(m/n) \approx 0.3$ 93
5.10 3D reconstruction: general translation, $(m/n) \approx 0.3$ 94
5.11 Surface and contour plot of translation objective function 95

6.1 2D reconstruction: bounded rotation, $(m/n) \approx 0.3$ 100
6.2 2D restricted rotation parameter sweep, $(m/n) \approx 0.3$ 101
6.3 2D full rotation parameter sweep, $(m/n) \approx 0.3$ 101
6.4 2D reconstruction: unbounded rotation, $(m/n) \approx 0.2$ 102
6.5 3D reconstruction: axial/sagittal/coronal rotation, $(m/n) \approx 0.2$ . . . 103
6.6 3D reconstruction: unbounded rotation in all axes, $(m/n) \approx 0.2$ . . . 104
6.7 Surfaces of rotation objective function 105
6.8 2D reconstruction: rigid-body motion, $(m/n) \approx 0.2$ 108
6.9 3D reconstruction: rigid-body motion, $(m/n) \approx 0.2$ 108
6.10 Surface and contour plot of rigid-body objective function 109

7.1 Objective function surfaces for different spatial resolutions 112
7.2 Objective function surfaces for different spatial resolutions 114
7.3 Approximate descent directions for different spatial resolutions 123
7.4 Noise basins around global minima 127
7.5 Inexact reconstruction from inexact parameters 128

8.1 Simple diagram of Siddon's algorithm 145
8.2 Effect of offset ray origins for hexagonal grid traversal 147
8.3 Construction of a hexagonal pixel (hexel) 148
8.4 Hex-FVT, Example 1: $(\varrho_1 > \varrho_2)$ 151
8.5 Hex-FVT, Example 2: $(\varrho_1 < \varrho_2)$ 152
8.6 Average computational cost of Hex-FVT 154
8.7 Per-intersection cost of FVT vs. Hex-FVT 156
8.8 Average error growth for Hex-FVT 157

8.9 Radiological path through a hexel grid 159
8.10 Ray intersections through a family of parallel lines 161
8.11 Projected intersections from a family of parallel lines 162
8.12 Classification of ‘odd’ and ‘even’ intersections 163
8.13 Bounding lines for ray origin 167
8.14 Computation of a hexel index from Cartesian coordinates 171
8.15 Average computational cost of Hex-Siddon 173
8.16 Average error growth for Hex-Siddon and Hex-FVT 174
8.17 Average computational cost of Hex-Siddon (GPU implementation) . . 175
8.18 2D hexagonal reconstruction: ‘optimal’ x-translation, $(m/n) \approx 0.3$ . . 177
8.19 2D reconstruction: comparison of pixels and hexels 178
8.20 Objective function similarity for pixel and hexel forward operators . . 179

9.1 Construction of a rhombic dodecahedral voxel (roxel) 182
9.2 Radiological path through a roxel grid 185
9.3 Genuine intersections in a planar set 186
9.4 Projection of ‘odd’ and ‘even’ intersections 187
9.5 Projections under a modulo map 188
9.6 Intersection checks under a modulo map 189
9.7 Average computational cost of Rox-Siddon 198
9.8 Average computational cost of Rox-Siddon (GPU implementation) . . 199

Part I

Industrial Problem and Mathematical Model

Overview of Part I

Part I of this thesis establishes the mathematical modelling framework that underpins the reconstruction and motion correction analysis implemented in later parts. The aim is not to provide an exhaustive treatment of tomographic theory, but to fix notation, geometry, and assumptions in a manner that is appropriate for the acquisition and discretisation strategies considered in this work.

Chapter 1 provides historical and conceptual context for X-ray imaging and tomography, situating this thesis within the development of tomosynthesis and limited-angle imaging. The contributions of this work are then presented, along with an overview of the entire thesis structure.

Chapter 2 introduces the mathematical objects required to formalise the reconstruction problem, including the Radon and X-ray transforms in two and three dimensions, together with a discussion of limited-angle effects and the visibility of singularities. These tools are used to motivate the ill-posedness inherent in the imaging geometries considered in this work, and to distinguish between detectability and stable localisation of features. Models for image acquisition are presented, with a focus on the construction of a discrete forward operator, followed by discussion of methods for solving the resulting image reconstruction problem. This chapter ends with further discussion of model choices and their limitations, relating to work undertaken in Part III.

Chapter 3 extends the discrete forward model and reconstruction framework used throughout the remainder of the thesis, enabling the effect of motion to be directly captured in the linear operator. This supports the choice to present motion correction and image reconstruction as distinct problems, to be interleaved as part of a nested optimisation framework.

Part I is then closed by Chapter 4 with a discussion on considerations for computational implementation of the problem, regarding both the numerical algorithm and numerical phantoms.

Chapter 1

Introduction

Contents

1.1 Historical overview of X-ray imaging 3
1.2 Acknowledgement of industrial partnership 6
1.3 Contribution of work 7
1.4 Outline of thesis 7

1.1 Historical overview of X-ray imaging

X-ray imaging has long been investigated under the framework of medical utility, beginning soon after the discovery of X-rays by W. C. Röntgen in the late 19$^{\text{th}}$ century [124] and continuing to span into numerous threads of modern research and development. While X-ray technology has numerous non-clinical uses—non-destructive materials testing and maintenance [44, 67], irradiation of food and water supplies [108, 156], inspection of art and historical artefacts [31, 127]—its utility in non-invasive internal imaging has been a mainstay for medical diagnosis and evaluation for over a century.

An inherent performance flaw of conventional X-ray imaging is that during acquisition only a single X-ray image is obtained, collapsing information of a measured three-dimensional object onto a two-dimensional plane—see Figure 1.1. There is, therefore, an unavoidable overlap of information in the direction of the projection, and so medical issues may be possible to identify but exact internal locations may be uncertain.

This issue was quickly recognised in the development of X-ray imaging, and in the early 20$^{\text{th}}$ century there were investigations into focal plane tomography techniques such as *planigraphy* [13, 160] and *laminography* [84]—precursors to future tomosynthetic

and tomographic methods. In effect, these processes worked by allowing both the X-ray emitter and detector to move along parallel planes, so that it was possible to infer information of the object within a particular focal plane. Unfortunately, this information came at the expense of creating additional out-of-plane artefacts which could still occlude objects in the focal plane—see Figure 1.2.

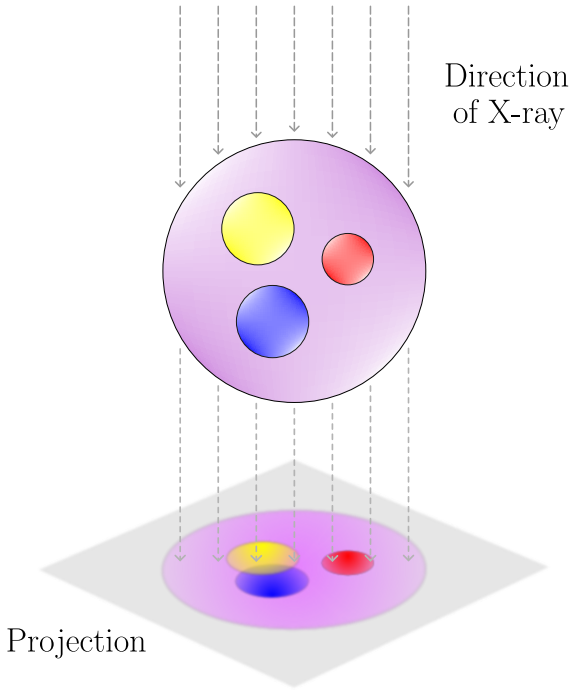


Figure 1.1: X-ray acquisition causes projected overlap between the yellow and blue internal structures, as well as a loss of depth information for all three internal structures. Without utilising other information, accurately determining the internal three-dimensional arrangement is essentially impossible.

While this was an improvement under certain conditions, it was not a viable way of reconstructing three-dimensional images due to the integrated exposures causing blurred artefacts within the final image. As such, this method of reconstruction lay dormant until a brief revival in tomosynthesis occurred in the 1970s, through investigations into constructing three-dimensional images from a 'stack' of lamino-graphs [57, 80, 106]. However, there were still certain limitations due to the available technology which meant that the realisation of this methodology was slow.

While the utility of these techniques was being rediscovered, the invention of computed tomography (CT) [4, 75] made it feasible to obtain three-dimensional images with much sharper diagnostic accuracy, essentially putting the development of tomosynthesis at a standstill. While CT was similar in style to the development

of previous methods, the primary improvement was that images could be acquired from any angle around a patient, meaning that the obstruction of focal planes was greatly reduced. In particular, out-of-plane artefacts that were often found via earlier tomographic methods were generally not seen in CT images, and so CT development became a primary focus of medical tomographic research.

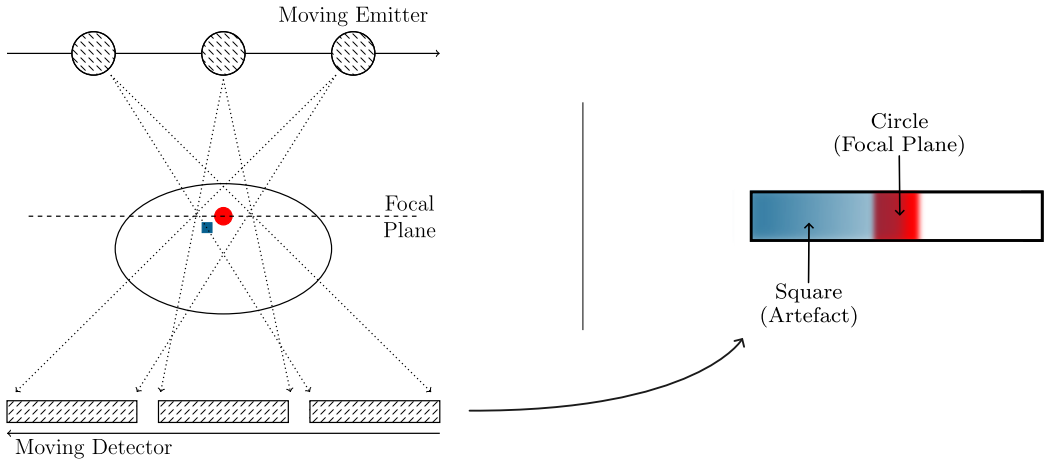


Figure 1.2: In focal plane tomography, the emitter and detector move relative to one another around the centre of the red circle, with the position of this circle defining the focal plane to be reconstructed. Objects not in this plane, such as the blue square, appear as smeared artefacts in the resulting image with the smearing being proportional to the distance from the focal plane.

The development of certain tomographic principles meant that it was possible, in theory, to also guarantee that an object could be reconstructed to arbitrary[1] accuracy from measurements sampled from viewing angles across a full circular range. The same principles also made it clear that purely tomosynthetic methods would forever fundamentally lack guarantees for arbitrary reconstruction accuracy—this concern will be briefly discussed in Section 2.1.3.

Even without theoretical reconstruction guarantees, there has been a return to tomosynthetic imaging due, in part, to advances in detector panel technology and improved computational processes. With these new improvements it has become possible to utilise tomosynthesis as a lower-cost alternative to CT, with the inevitability of imperfect image reconstruction being an acceptable trade-off for a reduction in both the acquisition time and total radiation cost. Within the past two decades there has

[1]Accuracy is inherently related to the number of X-ray measurements taken, as well as the uniformity of these measurements. As such, a high-accuracy image will generally require more measurements than a low-accuracy image.

been a growing use of digital tomosynthesis (DT) in mammography [39, 58, 128, 129] and, more recently, in COVID-19 diagnosis and assessment [7, 20, 118].

Throughout the continued development of tomosynthetic imaging methods, many improvements have focused either on the detector panel technology or the computational approaches used to reconstruct images from digitised data. In contrast, there has been little development in X-ray emitter technology, with most tomosynthetic and tomographic setups working from the principle of using a single high-power X-ray emitter to obtain measurement data. The work in this thesis is related, in part, to the development of a low-power, multi-emitter X-ray array. In particular, this work has been undertaken in partnership with Adaptix Ltd., who make such devices for both clinical and non-clinical use.

1.2 Acknowledgement of industrial partnership

The work included in this thesis has been jointly funded by the Engineering and Physical Sciences Research Council (EPSRC) and Adaptix Ltd., through the Industrially Focused Mathematical Modelling (InFoMM) Centre for Doctoral Training at the University of Oxford.

This work has been undertaken with a focus on a medical imaging device developed by Adaptix which will be discussed in more detail in Chapter 2. The development of a new type of X-ray emitter device has meant that much of the existing theory for tomosynthetic and tomographic reconstruction is not directly applicable without augmentation. Adaptix have previously shared research partnerships with K. Patel [116] and M. Klodt [86, 87], both under the supervision of Prof. R. Hauser. These investigations were undertaken with respect to device proof-of-concept, device geometry, and reconstruction methodologies. Additional work by M. Klodt was focused on improving existing reconstruction approaches for bodies that were completely static.

The work in this thesis focuses on the use case where there may be some underlying motion present during the imaging process, e.g. a patient may have difficulty remaining still for a full image acquisition, or there may be some unavoidable motion which must be accounted for. With this in mind, the work presented in this thesis can be viewed as a feasibility study for the ability to correct for simplified motions, and does not directly build upon work previously undertaken in partnership with Adaptix.

1.3 Contribution of work

The contribution of this work is primarily focused on device utility and reconstruction accuracy for the anticipated use cases of the Adaptix device. From an application-focused perspective, the main contributions of this thesis are:

- A proof-of-concept model for simple parametrised motion correction, through the inclusion of motion effects within the algebraic forward operator;
- Results showing that motion-corrected reconstructions can often be clearer than stationary reconstructions of the same object;
- An imaging and reconstruction framework built directly on hexagonal pixel discretisation, with no need for interpolation from square pixels;
- An imaging and reconstruction framework built directly on rhombic dodecahedral voxel discretisation, with no need for interpolation from cubic voxels.

This work also contains contributions along more theoretical lines, including:

- A new algorithm for exact ray traversal through hexagonal grids, with a computational cost of 4–5 floating point additions and 4–5 floating point comparisons per step;
- A new, parallel-per-cell algorithm for exact ray traversal through hexagonal grids which has both *a priori* computational cost and data storage upper bounds;
- An extended parallel-per-cell algorithm for exact ray traversal through rhombic dodecahedral grids which also has both *a priori* computational cost and data storage upper bounds.

1.4 Outline of thesis

Part I of this thesis contains four chapters, with this introduction being the first. Chapter 2 begins with an overview of tomographic and tomosynthetic imaging, including a discussion on reconstruction issues that can occur in digital tomosynthesis that are not as prevalent in other methodologies. From this we explore the theoretical limitations of DT methods to show why reconstruction guarantees cannot be achieved from certain sets of limited data. We then present a discrete forward

operator for image acquisition, and discuss possible methods for solving the general image reconstruction problem—it is here that we choose a purely algebraic approach. This chapter then closes with an overview of the assumptions and limitations present in this work. In Chapter 3 we first discuss existing methods for motion correction, leading to the choice of parametrised motion models as the focus of investigation. An extension to the algebraic reconstruction problem is presented, in which motion effects are directly modelled within the forward operator. This extension requires the development of a nested optimisation framework, in which motion correction and image reconstruction are solved in an interleaved manner. In Chapter 4 we discuss additional considerations related to the computational implementation of our forward operator and imaging framework, as well as discussing all numerical phantoms and algorithms used throughout.

Part II consists of three chapters, with the first two containing modelling and experimental results for a progression of parametrised rigid-body motion models. Chapter 5 focuses on the problem of reconstructing images that have been corrupted by translational motion, in which we explore both two- and three-dimensional geometries. Results are presented which show that reconstructions can be obtained that are *accurate to within diagnostic utility*, i.e. are accurate enough for human interpretation, and which can have better clarity than reconstructions of stationary objects. Chapter 6 opens with a discussion of the modelling and reconstruction of objects undergoing rotational motion, followed by investigations into correcting complete rigid-body motion. In Chapter 7 we discuss possible approaches for improving the motion correction methodology, focusing on the feasibility of multilevel optimisation and the idea that detectability of rigid-body motion is not tied to a fixed image resolution. This view of motion correction is discussed under the framework of recursive, multilevel optimisation, whereby we explore the benefits and complications that may come with this new formulation, and the empirical evidence relating such ideas to our work. Under this new methodology, we discuss the computational cost of finding approximate function minima, and the inherent instability of ill-posed inverse problems. We end Chapter 7 with a discussion of suitable bases for image representation, positing that alternative discretisations may be more suitable for solving the motion correction problem.

Part III focuses on the re-discretisation of the forward model using non-Cartesian cells, with Chapter 8 focusing on the utility of hexagonal pixels for 2D image re-

construction. In particular, we include two methods for exact ray traversal through hexagonal pixel grids, the first of which benefits from exploiting hexagonal geometries to require only 4–5 floating point additions and 4–5 floating point comparisons per step. The second algorithm, although computationally heavier, is *embarrassingly parallel* while still computing exact ray traversal distances through hexagonal grids. Numerical results show that both parallel and serial implementations of this new algorithm can achieve considerable speed-up compared to a classical ray-marching algorithm. Chapter 8 closes with a discussion of reconstruction clarity using hexagonal pixels, presenting a comparison against square pixel reconstructions of the same phantom. Chapter 9 follows a similar structure, but with a focus on extending our new algorithm to a three-dimensional voxel grid. We present an extended algorithm for ray traversal through rhombic dodecahedral voxels which enables forward operators to be computed directly on non-cubic grids without the need for interpolation. This algorithm is also embarrassingly parallel, and we present numerical results for both parallel and serial implementations. Finally, in Chapter 10 we conclude this work and discuss possible directions for future research.

Chapter 2

Background, forward model, and reconstruction

Contents

2.1 Foundations of tomography and tomosynthesis 10
2.2 Model and geometry assumptions 27
2.3 The tomographic reconstruction problem 37
2.4 Model limitations 44

2.1 Foundations of tomography and tomosynthesis

This chapter introduces the mathematical and geometric background required to formulate the tomographic reconstruction problem studied in this thesis. We begin by reviewing the Radon and X-ray transforms as idealised models of projection data, emphasising their geometric interpretation and the distinction between data sufficiency, uniqueness, and stable recoverability. The consequences of limited-angle acquisition are then discussed, including the visibility of singularities and the role of angular sampling, with complementary insight provided by the Fourier slice theorem. Classical analytic reconstruction methods are briefly outlined, before attention is restricted to fan-beam and cone-beam geometries and the associated completeness conditions governing three-dimensional imaging. The chapter concludes by motivating the specific acquisition geometry and practical constraints of the device considered in this work, establishing the modelling assumptions and inverse problem framework further developed in Chapter 3.

2.1.1 Introduction

The foundations of tomosynthetic imaging methods have existed for over a century, but for much of this time it was not a technology that found much utility. Focal plane tomographic methods were able to isolate a chosen plane-of-focus, but they would often be corrupted by artefacts from surrounding planes—see Figure 1.2. With these approaches there was no possibility to reliably obtain clear three-dimensional reconstructions of any particular object. With the development of computed tomography (CT) in the second half of the 20^{th} century [4, 75], it became possible to form a stack of tomographic planes which could then be viewed as an ensemble to represent a single three-dimensional object. Following the breakthrough of practical three-dimensional imaging, research activity concentrated heavily on CT reconstruction methods, with comparatively less attention given to other tomographic technologies.

Another important reason for CT becoming such a ubiquitous methodology is that it meets specific tomographic principles which help to guarantee that accurate images can be obtained, under certain conditions. In effect, it is generally true that if more measurements are acquired then the accuracy of the final reconstruction will improve. Importantly, this statement rests on the fact that measurements for CT reconstruction are obtained from a set of viewing angles that 'see' enough of the object being measured. By this we mean that for a set of viewing angles

$$\theta_\kappa \in [0, 2\pi) \qquad \text{for } \kappa = 1, \ldots, K, \tag{2.1}$$

with each determining the angle from which measurements are taken, the interval $[0, 2\pi)$ is sampled with sufficient angular density.

Viewing angles are often modelled to be uniformly spaced, i.e.

$$\theta_{\kappa+1} - \theta_\kappa = \delta\theta \qquad \text{for } \kappa = 1, \ldots, (K-1) \;\text{ and }\; \delta\theta \in \mathbb{R}, \tag{2.2}$$

which helps to ensure that the interval $[0, 2\pi)$ is adequately sampled. As multiple revolutions are made by the CT gantry, the viewing angles may instead be parametrised by fixed time intervals such that $\theta_\kappa = \theta(t_\kappa)$ for uniformly spaced times

$$t_\kappa \in [0, T] \qquad \text{for } \kappa = 1, \ldots, K. \tag{2.3}$$

Here, $\theta(t) = \omega t$ is a linear function with respect to a frequency ω, taken modulo 2π, and T is the total acquisition time[1].

[1] In general, the acquisition rate of a CT scanner is fast enough, relative to the revolution speed of the gantry, that (2.2) is well-approximated.

Although CT research has been a major focus of the medical imaging community for many decades, improvements in associated technologies have enabled other methodologies, such as digital tomosynthesis[2] (DT) to be re-evaluated as a feasible imaging approach. DT is an attractive alternative to CT as it generally requires a much smaller radiation dose[3] than CT, but this comes with the trade-off of not acquiring images over a large enough angular range, i.e. the viewing angles $\{\,\theta_\kappa\,\}_{\kappa=1}^{K}$ do not sample the interval $[0, 2\pi)$ sufficiently. As we will discuss, this ultimately makes the goal of high-accuracy reconstruction very difficult to achieve due to the ill-posedness of the resulting reconstruction problem. Even with this constraint, DT has still had a wide range of implementations, from dental imaging [78], chest and hand imaging [7, 43], as well as non-clinical uses such as non-destructive testing for building materials [93].

It may also not necessarily be the case that a limited angular range is a concession only made when using DT, as there are many use cases when it is just not feasible to image an object over a full angular range, e.g. when the object is too large to be imaged completely, or when only a certain *region-of-interest* is to be examined. In these cases, the full angular range cannot be sampled by *any* imaging method, and so DT can be an attractive modality given its inherently reduced radiation cost. However, for the purposes of this work we assume that there is no localised region-of-interest, and the object must be reconstructed in its entirety, such that limited-angle difficulties will affect the final reconstruction.

Before presenting the imaging model used throughout this thesis, we will first discuss the more general concepts of Radon transforms and X-ray transforms, and their link to the tomographic principles that underlie the inherent difficulties in DT imaging.

2.1.2 Important transforms for medical imaging

We present the construction of the Radon transform of a continuous two-dimensional function, following the presentation in English translations of Radon's original paper [37, 123]. This then naturally leads to the definition of the X-ray transform, which is more naturally suited to medical imaging in three dimensions.

[2]Digital tomosynthesis is sometimes abbreviated as DTS, or may be augmented depending on its utility; for example, DBT often refers to *digital breast tomosynthesis*. Throughout this work we will simply use DT.

[3]For example, typical radiation doses for a chest CT scan are between 4–8 mSv (millisieverts; 1 Sv = 1 J kg^{-1}), and around 1.5 mSv for *low-dose* CT, while DT doses are often one or two orders of magnitude smaller [52].

2.1.2.1 The Radon transform in two dimensions

We identify points $\mathbf{r} \in \mathbb{R}^2$ with Cartesian coordinates (x, y) and let $g(\mathbf{r}) = g(x, y)$ be a real-valued function which has the following properties:

1. $g(x, y)$ is continuous;
2. the integral $\iint_{\mathbb{R}^2} \frac{|g(x,y)|}{\sqrt{x^2+y^2}} \,\mathrm{d}x\,\mathrm{d}y$ converges;
3. for all $\mathbf{r} \in \mathbb{R}^2$ it holds that $\lim_{R\to\infty} \int_0^{2\pi} g(x + R\cos\phi, y + R\sin\phi)\,\mathrm{d}\phi = 0$.

Under these three regularity conditions, the *Radon transform* is defined as follows.

Definition 2.1.1 (Radon transform)**.** Let $\mathbf{p}_0 \in \mathbb{R}^2$, $\mathbf{v} \in \mathbb{R}^2 \setminus \{\mathbf{0}\}$, and let $g(\mathbf{r})$ satisfy the properties above. Then for any straight line

$$\ell := \{\, \mathbf{p}_0 + s\mathbf{v} \mid s \in \mathbb{R} \,\} \subset \mathbb{R}^2, \tag{2.4}$$

the Radon transform, $\mathcal{R}g$, along ℓ is defined as

$$(\mathcal{R}g)(\ell) := \int_\ell g(x, y)\,\mathrm{d}\ell. \tag{2.5}$$

Here $\mathrm{d}\ell$ denotes the arc length measure along ℓ, so that the integral is independent of both the parametrisation of ℓ and the direction in which it is traversed.

By choosing $\mathbf{p}_0 = (t\cos\theta, t\sin\theta)^\top$ to be the orthogonal projection $\mathbf{p}_0 = \mathrm{proj}_\ell(\mathbf{0})$, and setting the pair[4] $\{\mathbf{w}, \mathbf{w}^\perp\}$ to be defined by $\mathbf{w} = \mathbf{p}_0/\|\mathbf{p}_0\|_2$ and $\langle \mathbf{w}, \mathbf{w}^\perp \rangle = 0$, we may instead parametrise ℓ as

$$\begin{aligned} \ell(s) &= (t\cos\theta - s\sin\theta,\ t\sin\theta + s\cos\theta)^\top, && (2.6)\\ &= t\mathbf{w} + s\mathbf{w}^\perp, && (2.7) \end{aligned}$$

for all $s \in \mathbb{R}$. Alternatively, ℓ is the set[5]

$$\ell(t, \theta) = \{\, \mathbf{r} \in \mathbb{R}^2 \mid \langle \mathbf{r}, \mathbf{w} \rangle = t \,\}. \tag{2.8}$$

[4] As the Radon transform is independent of the direction of traversal, both $\mathbf{w}^\perp = (\sin\theta, -\cos\theta)^\top$ and $\mathbf{w}^\perp = (-\sin\theta, \cos\theta)^\top$ are valid choices—we choose the latter.

[5] Should it ever occur that $\mathbf{p}_0 = \mathbf{0}$, i.e. ℓ passes through the origin, we set $\mathbf{w} = (\cos\theta, \sin\theta)^\top$ as a unit vector orthogonal to ℓ and set $t = 0$, so that the representation (2.8) remains correct.

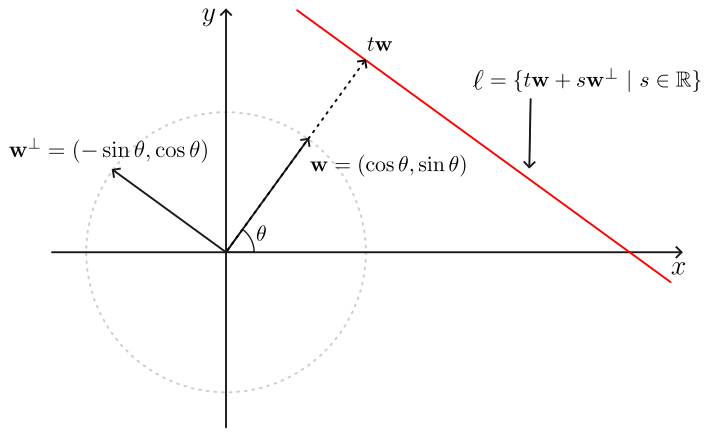


Figure 2.1: The red line shows the set ℓ defined by the pair (t, θ), over which the two-dimensional Radon transform may then be defined.

Associating all straight lines $\ell \subset \mathbb{R}^2$ with their arguments $(t, \theta) \in \mathbb{R} \times [0, 2\pi)$, the Radon transform is equivalently defined by

$$(\mathcal{R}g)(t, \theta) = \int_{\ell(t,\theta)} g(\mathbf{r}) \, \mathrm{d}\ell, \tag{2.9}$$

$$= \int_{-\infty}^{\infty} g(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta) \, \mathrm{d}s. \tag{2.10}$$

An alternative formulation is the distributional representation

$$(\mathcal{R}g)(t, \theta) = \int_{\mathbb{R}^2} g(\mathbf{r}) \, \delta\left(t - \langle \mathbf{r}, \mathbf{w}(\theta) \rangle\right) \, \mathrm{d}\mathbf{r}, \tag{2.11}$$

where $\mathbf{w}(\theta) = (\cos\theta, \sin\theta)^\top$ and $\delta(\cdot)$ denotes the one-dimensional Dirac delta distribution. In either form, it holds that $(\mathcal{R}g)(t, \theta) = (\mathcal{R}g)(-t, \theta + \pi)$, reflecting the fact that the Radon transform is defined over non-oriented lines.

It is helpful to understand how the Radon transform acts under a change in t for a fixed angle θ^*, before discussing the tomographic reconstruction problem more broadly. We define the *fixed-angle Radon projection*[6] of g by the function

$$p_{\theta^*}(t) := (\mathcal{R}g)(t, \theta^*), \tag{2.12}$$

$$= \int_{\mathbb{R}} g(t\cos\theta^* - s\sin\theta^*, t\sin\theta^* + s\cos\theta^*) \, \mathrm{d}s. \tag{2.13}$$

For a chosen angle θ^* the projection function $p_{\theta^*}(t)$ is defined for all $t \in \mathbb{R}$, and is obtained by integrating over all lines $\{ \ell(t, \theta^*) \mid t \in \mathbb{R} \}$, which we can interpret as the complete one-dimensional projection of g in the direction of $\mathbf{w}^\perp$—see Figure 2.2.

[6]This is often referred to as a *single Radon projection* or a *Radon slice*, although these are normally given in terms of the projection direction, which would be $\theta^* \pm \pi/2$ in this context.

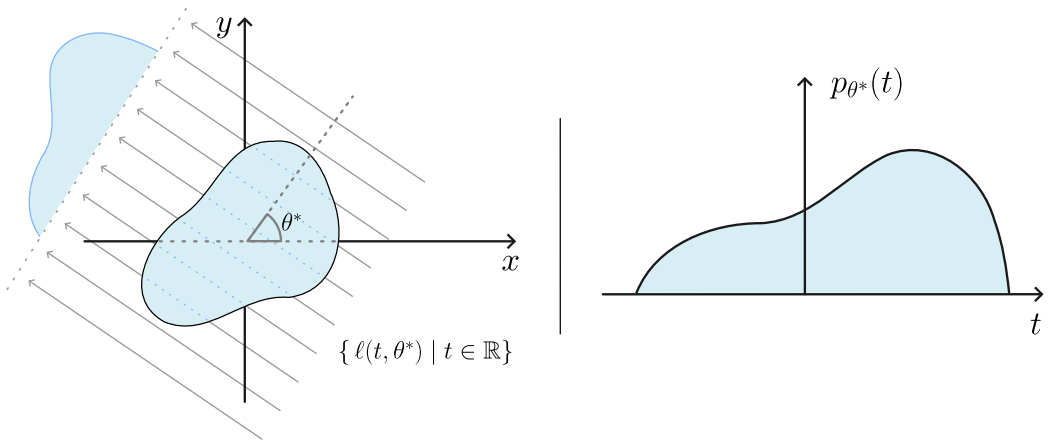


Figure 2.2: The fixed-angle Radon transform produces a one-dimensional function $p_{\theta^*}(t)$, associated with the set of integral values along the lines $\{\, \ell(t, \theta^*) \mid t \in \mathbb{R} \,\}$.

We may therefore define a distinct projection function $p_\theta(t)$ for all angles $\theta \in [0, 2\pi)$, with the property that $p_{\theta+\pi}(t) = p_\theta(-t)$. The complete set of projections,

$$\{\, p_\theta(t) \mid \theta \in [0, \pi) \,\}, \tag{2.14}$$

is the maximal non-redundant information obtainable about $g(\mathbf{r})$ through line integrals along straight lines. It is with respect to this set that the tomographic reconstruction problem is defined, which we will present in Section 2.1.2.4.

2.1.2.2 The Radon transform in higher dimensions

The Radon transform can be generalised to $\mathbb{R}^n$ for functions $g(\mathbf{r}) = g(x_1, \ldots, x_n)$, with the sufficient conditions now being defined in terms of the larger space:

1. $g(\mathbf{r})$ is continuous;
2. the integral $\displaystyle\int_{\mathbb{R}^n} \frac{|g(\mathbf{r})|}{\|\mathbf{r}\|_2^{n-1}} \, \mathrm{d}\mathbf{r}$ converges;
3. for all $\mathbf{r} \in \mathbb{R}^n$ it holds[7] that $\displaystyle\lim_{R \to \infty} \int_{\mathbb{S}^{n-1}} g(\mathbf{r} + R\boldsymbol{\omega}) \, \mathrm{d}\boldsymbol{\omega} = 0$.

For a unit vector $\boldsymbol{\theta}^* \in \mathbb{S}^{n-1}$ the Radon transform is defined with respect to integrals over the hyperplane family $\{\, \mathbf{H}(t, \boldsymbol{\theta}^*) \mid t \in \mathbb{R} \,\}$, where each hyperplane is defined by

$$\mathbf{H}(t, \boldsymbol{\theta}^*) := \{\, \mathbf{r} \in \mathbb{R}^n \mid \langle \mathbf{r}, \boldsymbol{\theta}^* \rangle = t \,\}. \tag{2.15}$$

[7] Here, we use the notation $\mathbb{S}^{n-1} := \{\, \boldsymbol{\omega} \in \mathbb{R}^n \mid \|\boldsymbol{\omega}\|_2 = 1 \,\}$ to define the unit sphere in $\mathbb{R}^n$.

As such, the n-dimensional Radon transform $\mathcal{R}g$ is

$$(\mathcal{R}g)(t, \boldsymbol{\theta}^*) := \int_{\mathbf{H}(t,\boldsymbol{\theta}^*)} g(\mathbf{r}) \, \mathrm{d}\sigma(\mathbf{r}), \tag{2.16}$$

where $\mathrm{d}\sigma(\mathbf{r})$ is the $(n-1)$-dimensional Hausdorff measure, generalising the arc length integral in (2.5) to higher dimensions.

As this definition is in terms of $(n-1)$-dimensional hyperplanes, the Radon transform does not directly model medical X-ray imaging in three dimensions, where measured data correspond to integrals along one-dimensional affine subspaces of $\mathbb{R}^3$. This motivates the introduction of the X-ray transform, which more accurately models the physical acquisition process.

2.1.2.3 The X-ray transform

Let $g(\mathbf{r})$ satisfy the regularity conditions stated in Section 2.1.2.2 for the n-dimensional Radon transform[8]. The *X-ray transform*, denoted by $\mathcal{X}g$, is defined as the integral along a straight line $\ell \subset \mathbb{R}^n$:

$$(\mathcal{X}g)(\ell) := \int_{\ell} g(\mathbf{r}) \, \mathrm{d}\ell, \tag{2.17}$$

where $\mathrm{d}\ell$ denotes the arc length measure along ℓ.

Parametrising ℓ by an initial point $\mathbf{p}_0 \in \ell$ and a direction vector $\mathbf{v} \in \mathbb{S}^{n-1}$, we may instead write $\mathcal{X}g$ as

$$(\mathcal{X}g)(\mathbf{p}_0, \mathbf{v}) = \int_{\mathbb{R}} g(\mathbf{p}_0 + s\mathbf{v}) \, \mathrm{d}s. \tag{2.18}$$

The particular choice of $\mathbf{p}_0 \in \ell$ is not important, as a different $\widetilde{\mathbf{p}}_0 \in \ell$ corresponds to a variable change $s \mapsto s + c$ for some constant $c \in \mathbb{R}$, and so we take $\mathbf{p}_0 = \mathrm{proj}_{\ell}(\mathbf{0})$ to be the closest point of ℓ to the origin, yielding a parametrisation analogous to the offset used in the definition of the Radon transform.

In two dimensions, the Radon transform and X-ray transform coincide and certain results from Radon transform theory are useful for understanding limitations of image reconstruction in higher dimensions. For a chosen $\mathbf{v}$, the fixed-direction transform is defined as the complete set of integrals along all lines parallel to $\mathbf{v}$—in Section 2.1.2.1 this set was parametrised by a single offset $t\mathbf{w}(\theta)$ for $t \in \mathbb{R}$, which yielded the set $\{\, p_\theta(t) \,\}_{\theta \in [0,\pi)}$.

[8] While the natural condition for the X-ray transform is $g \in L^1(\mathbb{R}^n)$, due to the construction of line integrals, we will keep these sufficient (and overly-strong) conditions.

In n dimensions, fixing a direction $\mathbf{v} \in \mathbb{S}^{n-1}$ leaves $(n-1)$ transverse degrees of freedom, so the fixed-direction projection is naturally a function of $(n-1)$ variables, e.g. $p(t_1, \ldots, t_{n-1}; \mathbf{v})$ whose arguments parametrise the orthogonal complement of $\mathrm{span}(\mathbf{v})$. Further details of the X-ray transform can be found in the paper of Faridani [50], with more detailed theory found in the thesis of Solmon [136].

We note here that the above descriptions of the Radon and X-ray transforms are specifically for *parallel-beam* models, i.e. ones in which the X-ray source is modelled as a point at infinity, and all straight lines ℓ intersect only at the point of infinity. However, there exist similar parametrisations for different imaging models that instead use either *fan-beam* or *cone-beam* geometries—these models will be discussed in Section 2.1.4, as they are more directly related to our work.

2.1.2.4 The tomographic reconstruction problem

While the X-ray transform admits a natural formulation in $\mathbb{R}^n$, the remainder of this work focuses exclusively on two- and three-dimensional reconstruction problems, which are of primary relevance in imaging applications. The fundamental objective of both tomographic and tomosynthetic reconstruction is to recover the function $g(\mathbf{r})$ from a set of X-ray projections, defined by

$$\mathcal{X}_{\mathcal{A}} := \{\, (\mathcal{X}g)(\mathbf{p}_0^{(\alpha)}, \mathbf{v}^{(\alpha)}) \,\}_{\alpha \in \mathcal{A}}, \tag{2.19}$$

for pairs $(\mathbf{p}_0^{(\alpha)}, \mathbf{v}^{(\alpha)})$ enumerated by an indexing set $\mathcal{A}$ [29, 30, 123].

In general, a unique solution requires a continuum of projections to be used in the reconstruction method, provided a stable inversion process exists [121]. In this idealised setting, the stated regularity assumptions ensure pointwise uniqueness of the reconstruction on compact domains; in weaker L^p-based frameworks uniqueness is understood only up to equality almost everywhere, i.e. two solutions may differ only on a set of measure zero.

2.1.3 Limited-angle difficulties

For practical purposes in which only a finite set of images is acquired, theoretical uniqueness in the continuous inverse problem is no longer guaranteed, although this is rarely critical for medical diagnoses. A more significant constraint arises in limited-angle tomosynthesis, as even if an inversion process did exist for an infinite number of measurements it would be highly ill-conditioned and unsuitable for computational

implementation. Results from Davison [32] show that when projection data are restricted to an angular interval of length strictly less than π, e.g. $[\varepsilon - \pi/2, \pi/2 - \varepsilon]$ for $\varepsilon > 0$, then any inversion of the restricted Radon operator—and hence of the corresponding X-ray operator—is severely ill-conditioned, with the ill-conditioning growing exponentially as ε increases. Therefore, while the issue of non-uniqueness affects all discrete reconstruction processes, it is the undersampling of angular information that constitutes the primary obstacle to obtaining valid reconstructions.

The problem tackled in this thesis suffers from having limited angular information, and so we present a basic illustration of the difficulties that may arise, following ideas by Quinto [120, 121]. In the framework of imaging inverse problems, some internal features cannot be located under certain constraints due to singularities in the object; in this context, a singularity refers to a jump discontinuity in the object, such as a sharp boundary between regions of different density. Quinto's notion of visibility concerns whether such singularities induce corresponding non-smooth behaviour in the projection data and are therefore recoverable. In particular, if a dataset does not contain a family of rays whose directions are aligned so as to be parallel to a jump discontinuity, then the associated boundary does not induce a singularity in the projection data and cannot be stably reconstructed from limited-angle measurements alone.

As a very simple example—presented in Figure 2.3—we imagine that the object to reconstruct consists of a unit cube with an interior density $\rho_1 = 1$ with a second unit cube directly on top of it with an interior density $\rho_2 = 2$. We have not yet discussed how X-ray attenuation is modelled, but let us imagine that the only data set that we have is a set of integral projections in the direction $\mathbf{v} = (0, 0, -1)^\top$.

Owing to the homogeneity of the cubes and the simple geometry, all line integrals passing through the object in this direction have the same value. Consequently, the resulting projection is

$$p(x, y) = \begin{cases} c & \text{if } \ 0 < x < 1\,,\, 0 < y < 1, \\ 0 & \text{otherwise,} \end{cases} \tag{2.20}$$

for some $c \in \mathbb{R}$. From these data alone, it is impossible to distinguish between the two cubes, as the measurements collapse to a single constant value.

If instead we take integral measurements in the direction $\mathbf{v} = (-1, 0, 0)^\top$, the

resulting projection is defined on the (y, z)-plane and exhibits a clear separation:

$$p(y,z) = \begin{cases} c_1 & \text{if } \ 0 < y < 1\,,\ 0 < z < 1, \\ c_2 & \text{if } \ 0 < y < 1\,,\ 1 < z < 2, \\ 0 & \text{otherwise,} \end{cases} \tag{2.21}$$

for some $c_1, c_2 \in \mathbb{R}$, with $c_1 \neq c_2$. Under the assumption that the object contains a single planar interface, the location of the corresponding boundary at $z = 1$ is therefore recoverable from these data.

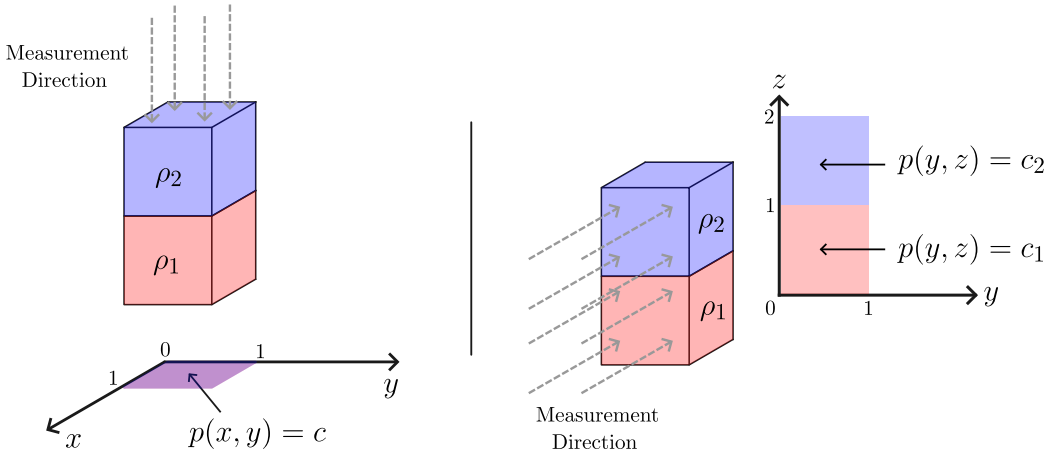


Figure 2.3: Measurements taken along the direction of the z-axis cannot detect the singularity between the two objects. However, measurements taken along the direction of the x-axis not only detect it, but can also identify its location.

Between these two simple cases, for a general viewing direction $\mathbf{v}$, which is neither parallel nor orthogonal to the z-axis, one may expect the resulting projection $p(t_1, t_2; \mathbf{v})$ to be a piecewise smooth function of the transverse variables. The presence of the interface at $z = 1$ will influence the measured data; however, in this simple example, only when measurements are taken along rays sufficiently aligned so as to be parallel to the boundary does the associated jump discontinuity manifest as a singularity in the data. Therefore, for this example the singularity at $z = 1$ is stably localisable only from families of rays with directions $\mathbf{v}$ parallel to the interface, while for other directions the boundary cannot be recovered in a stable manner [120].

With this in mind, it is reasonable to expect that reconstruction processes which lack measurements over a sufficiently wide angular range will have directions for which boundaries are invisible in the sense described above. Many of the difficulties arising in limited-angle tomography may be understood by considering the corresponding loss

of information in the Fourier domain. To illustrate this connection, we present the *Fourier slice theorem* and relate it to the Radon/X-ray transform in two dimensions.

Theorem 2.1.1 (Fourier slice theorem)**.** *Let $f(u, v)$ be the two-dimensional Fourier transform of the function $g(x, y)$, and let $p_{\theta^*}(t)$ be the projection of $g(x, y)$ with respect to a fixed angle θ^*. Then the one-dimensional Fourier transform of $p_{\theta^*}(t)$ is equal to the restriction of the two-dimensional Fourier transform of $g(x, y)$ along the line through the origin at angle θ^*.*

Proof. Without loss of generality, we may assume that the projection angle is $\theta^* = 0$, as the result for general projection angles follows immediately by rotational invariance of both the Radon and Fourier transforms. The projection along $\theta^* = 0$ is therefore

$$p_{\theta^*}(x) = \int_{-\infty}^{\infty} g(x, y)\, \mathrm{d}y, \tag{2.22}$$

while the two-dimensional Fourier transform is given by

$$f(u, v) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y)\, e^{-2\pi i(xu+yv)}\, \mathrm{d}x\, \mathrm{d}y. \tag{2.23}$$

The straight line through the Fourier domain at angle $\theta^* = 0$ is simply

$$f(u, 0) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y)\, e^{-2\pi ixu}\, \mathrm{d}x\mathrm{d}y, \tag{2.24}$$

$$= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} g(x, y)\, \mathrm{d}y \right) e^{-2\pi ixu}\, \mathrm{d}x, \tag{2.25}$$

where the interchange of integrals is justified by the regularity assumptions on g. Therefore,

$$f(u, 0) = \int_{-\infty}^{\infty} p_{\theta^*}(x) e^{-2\pi ixu}\, \mathrm{d}x, \tag{2.26}$$

which is the one-dimensional Fourier transform of $p_{\theta^*}(x)$. □

The primary use of Theorem 2.1.1 is to understand that while an object may be measured in the sense of having an X-ray projection pass through it, there will be inherent undersampling in the Fourier domain, meaning that there are directions in the Cartesian frame along which frequencies cannot be determined.

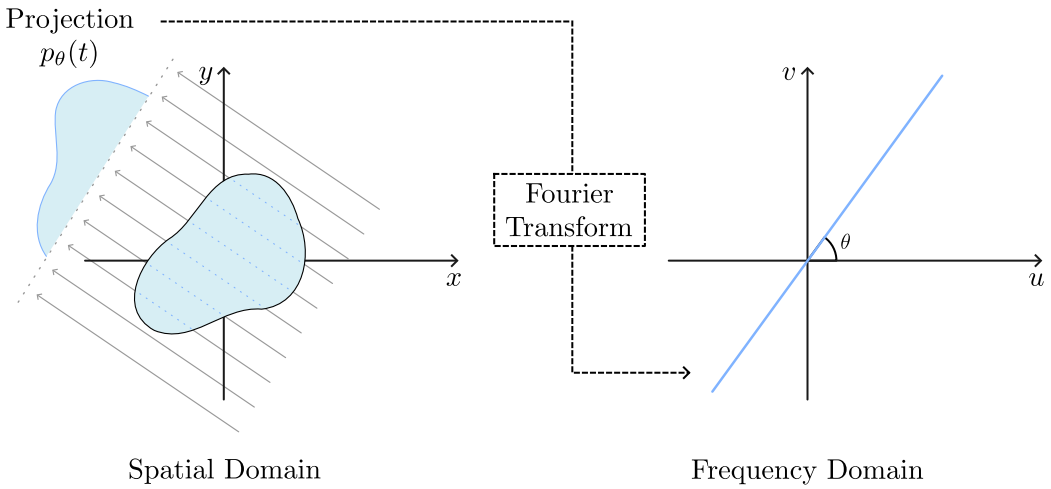


Figure 2.4: The Fourier slice theorem gives a relation between the measurement space and the frequency space, suggesting an equivalence of invertibility depending on the sampling in the Fourier domain.

As shown by the example in Figure 2.3, if the available viewing directions do not sample Fourier space in directions normal to a given boundary, then that boundary cannot be stably recovered. From the perspective of the Fourier domain, for a general reconstruction problem, even with a continuum of measurements over angles $\theta \in [-\eta, \eta]$ for some $\eta < \pi/2$ there exists frequency information oriented in a direction $\theta^* \in [-\pi/2, -\eta] \cup [\eta, \pi/2]$, without which the object cannot be reconstructed entirely.

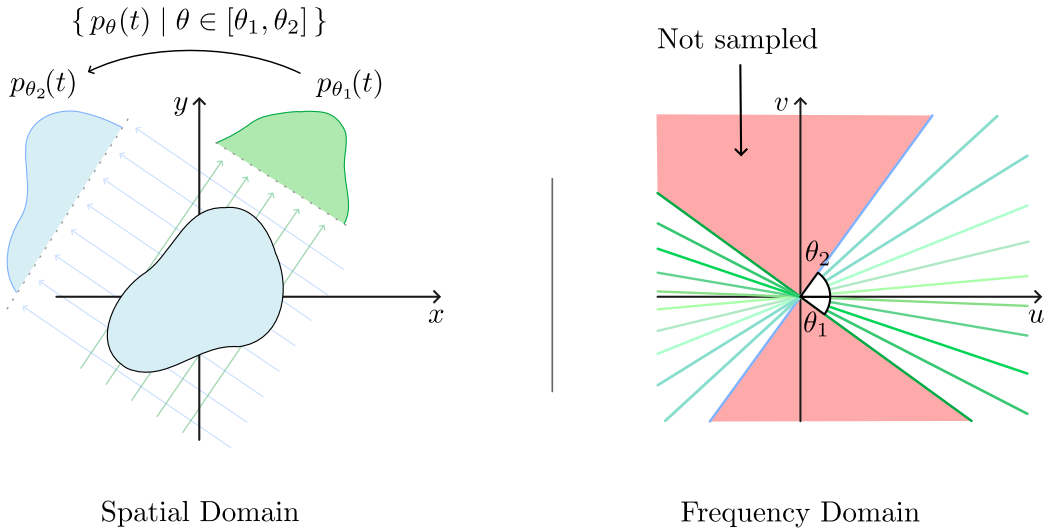


Figure 2.5: Undersampling in the Fourier domain is directly related to limited angular measurements in the spatial domain. Therefore, for limited-angle problems there will, in general, be frequency information along certain directions that cannot be recovered in the reconstruction process.

2.1.3.1 Analytic reconstruction

Let us imagine a two-dimensional function $g(x, y)$ and assume the existence of a complete set of X-ray transforms, i.e. the set $\mathcal{X}_{\mathcal{A}}$ is indexed by all pairs[9] $(t, \theta) \in \mathbb{R} \times [0, \pi)$ such that it contains all possible information about $g(x, y)$ that can be obtained via lower-dimensional projections.

To solve the tomographic reconstruction problem, one may imagine it is possible to redistribute the measured projection data back into the image domain in an attempt to accumulate contributions from all projections passing through a given point (x, y). With such a thought, one would rediscover the idea of "backprojection" whereby for any point (x, y) we sum the contributions of all projections that pass through it. For parallel-beam models, this results in an integral of the form

$$b(x, y) = \int_0^{\pi} (\mathcal{R}g)(x \cos\theta + y \sin\theta, \theta) \, \mathrm{d}\theta. \tag{2.27}$$

It is important to understand that the backprojection operator is not an inversion of the projection operator—rather it acts as an adjoint operator, and so an analytic backprojection rarely results in the original image. There is a large body of research into *filtered backprojection* schemes, many of which rely on applying a linear frequency-domain filter to $\mathcal{R}g$ before performing the adjoint integral. This area is not a focus of this thesis, but we refer the reader to [49, 83] for further details.

2.1.4 Alternative beam geometries

The theory discussed so far is in terms of parallel rays[10], as this is the foundation of the papers by Radon [123] and Cormack [29, 30], and has a simple formulation in terms of geometric rays. However, further ray geometries have been explored, leading to results in fan-beam reconstruction [74, 158] as well as cone-beam reconstruction [51], which are more directly relevant to the acquisition geometry considered in this thesis.

[9] Recall that we do not need repeated information from the interval $[\pi, 2\pi)$, due to the symmetry of the X-ray transform.

[10] Historically, X-ray models refer to both *rays* and *beams*, so these terms will be used interchangeably from now on, unless the precise geometric definition of a ray is needed.

2.1.4.1 Fan-beam model

A natural extension to the two-dimensional parallel-beam X-ray model is that of a fan-beam X-ray—see Figure 2.6—which accounts for the spatial position of an emitter, rather than assuming all rays are generated from a far-field point source.

We assume that there exists a fixed focal point $\mathbf{c} = (x_0, y_0)^\top$ which acts as the source of the fan-beam, to which we can associate all lines in the plane that intersect $\mathbf{c}$. If we have a natural origin for the plane, then lines intersecting $\mathbf{c}$ may be parametrised by two angles: α, which gives the polar angle of $\mathbf{c}$ with respect to the origin, and β, which determines the relative direction of a particular line intersecting $\mathbf{c}$. We choose $\beta = 0$ to correspond to the line through $\mathbf{c}$ that intersects the origin, and we take the fan to be symmetric about this line, so that all lines through $\mathbf{c}$ may be associated with a pair[11] $(\alpha, \beta) \in [0, 2\pi) \times [-\beta_{\max}, \beta_{\max}]$, where $\beta_{\max}$ makes the fan symmetrically bounded.

For a chosen (α, β) pair, we denote the associated line through $\mathbf{c}$ by $\eta(\alpha, \beta)$, and define $\varphi(\alpha, \beta) \in [0, 2\pi)$ to be the angle of its direction vector, measured anti-clockwise from the x-axis. Using the same notation as in Section 2.1.2.1, we associate $\eta(\alpha, \beta)$ with the line $\ell = \{\, \mathbf{c} + s\mathbf{v} \mid s \in \mathbb{R} \,\}$, where

$$\mathbf{v}(\alpha, \beta) = \big(\cos(\varphi(\alpha, \beta)), \sin(\varphi(\alpha, \beta))\big)^\top. \tag{2.28}$$

Choosing $\mathbf{v}^\perp(\alpha, \beta) = \big(-\sin(\varphi(\alpha, \beta)), \cos(\varphi(\alpha, \beta))\big)^\top$, each line is then identified with parallel-beam coordinates (t, θ) via

$$t(\alpha, \beta) = \langle \mathbf{v}^\perp(\alpha, \beta), \mathbf{c} \rangle, \tag{2.29}$$

$$\theta(\alpha, \beta) = \left(\varphi(\alpha, \beta) + \frac{\pi}{2}\right) \pmod{2\pi}, \tag{2.30}$$

so that, for a fixed focal point $\mathbf{c}$, the mapping $(\alpha, \beta) \mapsto (t, \theta)$ is smooth away from the branch cut at $\theta = 0$, introduced by the modulo 2π reduction. Since a line and its reversal correspond to (t, θ) and $(-t, \theta + \pi)$, we restrict θ to $[0, \pi)$ by identifying antipodal pairs, recovering the Radon parametrisation of Section 2.1.2.1.

Given the existence of such a map, and allowing the position of $\mathbf{c}$ to vary, rays between the fan-beam and the parallel-beam models may be identified, such that the total set of lines

$$\{\, \ell(\mathbf{c}(r, \alpha, \beta), \mathbf{v}(\alpha, \beta)) \mid (r, \alpha, \beta) \in \mathbb{R} \times [0, 2\pi) \times [0, 2\pi) \,\} \tag{2.31}$$

[11] As $\mathbf{c}$ may be associated with the polar coordinate pair (r, α), this means that all lines may be (non-uniquely) associated with triples $(r, \alpha, \beta) \in [0, \infty) \times [0, 2\pi) \times [0, 2\pi)$.

is equal[12] to

$$\{\, \ell(t,\theta) \mid (t,\theta) \in \mathbb{R} \times [0,\pi) \,\}. \tag{2.32}$$

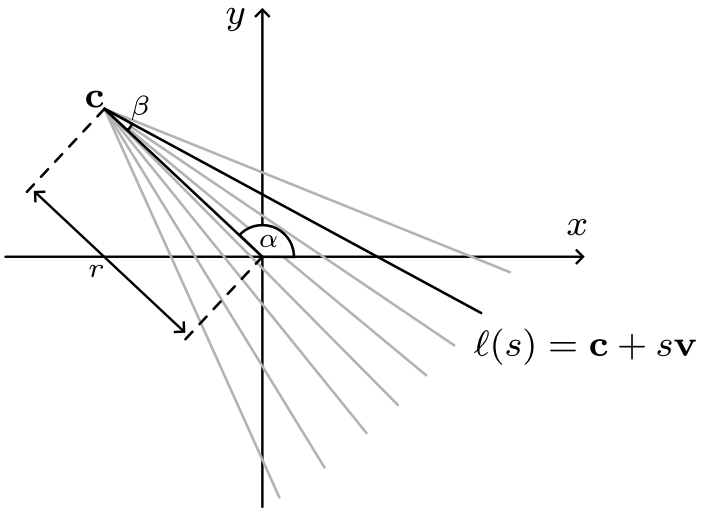


Figure 2.6: The position of the source can be described by polar coordinates (r, α), and the relevant beam can then be identified by the triple (r, α, β).

Consequently, fan-beam geometries do not introduce new ray information, but rather reorganise the same family of line integrals into emitter-centred coordinates. As a result, issues such as limited-angle acquisition and incomplete angular coverage remain present in the fan-beam setting.

2.1.4.2 Cone-beam model

A natural extension of the fan-beam model is that of a cone-beam for three-dimensional imaging, which can be interpreted as the rotation of a fan-beam model about the central line defined by $\beta = 0$. As in the two-dimensional case, we describe each line geometrically in terms of a source location and a direction. Let $\mathbf{c} \in \mathbb{R}^3$ denote a fixed cone-beam source point. Then any ray emitted from $\mathbf{c}$ may be written in the form $\ell = \{\, \mathbf{c} + s\mathbf{v} \mid s \in \mathbb{R} \,\}$, where $\mathbf{v} \in \mathbb{S}^2$ is a unit direction vector.

To parametrise the family of directions $\mathbf{v}$ associated with a cone-beam source, we proceed by extending the fan-beam parametrisation. We introduce three angular parameters (γ, α, β) with the following roles:

- γ specifies a rotation of the fan-beam plane about the central ray passing through the origin, determining the orientation of the fan within the cone;

[12] Here, equality is understood in terms of the sets rather than the parametrisations, as the map $(r, \alpha, \beta) \mapsto (t, \theta)$ is not injective.

- α determines the rotation, within the fan-beam plane, required to align the central ray with the source location $\mathbf{c}$;
- β specifies the relative angular deviation of a ray from the central ray within the fan.

For a fixed value of γ, the parameters (α, β) therefore describe a fan-beam geometry identical to the two-dimensional case, embedded in a plane whose orientation is controlled by γ. Each cone-beam ray is naturally indexed by a source point[13] $\mathbf{c}$ and a direction $\mathbf{v}(\gamma, \alpha, \beta) \in \mathbb{S}^2$. Thus, each triple (γ, α, β) determines a unique cone-beam ray $\eta(\gamma, \alpha, \beta)$, emanating from $\mathbf{c}$, which we identify with the parallel-beam X-ray line

$$\ell(\gamma, \alpha, \beta) = \{\, \mathbf{c} + s\mathbf{v}(\gamma, \alpha, \beta) \mid s \in \mathbb{R} \,\}. \tag{2.33}$$

In full generality, the location of each source $\mathbf{c}$ can also be described by the four-tuple[14] $(r, \gamma, \alpha, \beta)$ such that the complete set of cone-beam rays can be mapped (non-uniquely) to the complete set of parallel-beam rays.

Unlike the fan-beam case, cone-beam data do not admit a direct reparametrisation in terms of plane integrals, and no analogue of the Fourier slice theorem can be applied. Instead, the question of whether a cone-beam acquisition contains sufficient information for reconstruction is understood through global completeness conditions on the source trajectory. A fundamental set of results due to Tuy [141] and Smith [135] states that a cone-beam dataset is complete provided that every plane intersecting the object also intersects the source trajectory. This requirement, commonly referred to as the *Tuy–Smith sufficiency condition*, is purely geometric in nature; it characterises when the acquisition geometry is theoretically sufficient for exact reconstruction, but does not prescribe a reconstruction algorithm.

It is important to note that this condition addresses the global recoverability of the object, rather than the local stability of reconstruction. In contrast, the local analysis developed by Quinto [120, 121] characterises which singularities of an object are visible from a given acquisition geometry, depending on whether rays exist whose directions are compatible with the orientation of those singularities. In this sense, the Tuy–Smith condition guarantees global data sufficiency and tells us whether reconstruction is at all possible, while local visibility results describe what structural information can be stably recovered in practice.

[13] As in the fan-beam model, this point is not unique.

[14] The relationship is not as clear as for the fan-beam case, so we do not derive it here.

2.1.5 Modelled device utility

In most conventional DT systems, the X-ray emitter must be positioned at a substantial distance from the patient—this is required for both geometric coverage and to mitigate localised radiation concentration at the skin of a patient. Adaptix have developed a flat-panel X-ray source comprising an array of low-power emitters which is compact enough for clinical bedside use.

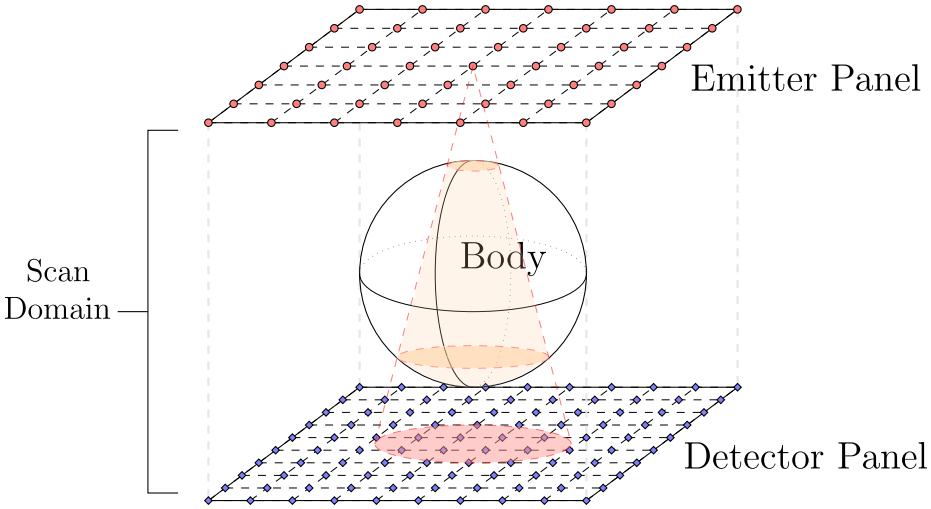


Figure 2.7: During image acquisition, the region of the body being measured will be contained within the domain bounded by the two panels. When a single emitter is fired it may measure only a section of the body, projecting onto a subset of the total set of detectors.

At the level of an individual emitter, data acquisition is equivalent to the cone-beam model described above, but at a substantially reduced radiation dose. By sequencing emitters in turn, the device can therefore acquire multiple two-dimensional projections whose combined radiation exposure is comparable to that of a single conventional radiograph. From this collection of measurements, a three-dimensional reconstruction may be obtained with significantly lower radiation dose than a full CT scan.

Unlike conventional DT systems, in which data acquisition is generated through controlled mechanical motion of the emitter–detector pair, this device remains fixed in space. Projection diversity is therefore achieved by electronically sequencing the set of spatially separated emitters. While this eliminates the need for precise mechanical motion, it also means that each emitter illuminates the patient from a distinct centre of projection. In contrast to traditional CT and DT approaches—where the emitter–detector pair moves around a chosen point in space—the emitters in this device do

not share a common fulcrum. As a result, the acquired data are mutually offset, with a deterministic parallax characterised by the emitter spacing and panel geometry.

Due to the use of a multi-emitter array, individual emitters may capture only partial views of the object, as illustrated in Figure 2.7. It will, however, always be assumed that the entire object is contained within the union of all emitter cone-beams.

2.1.6 Industrial problem

A particular focus of this thesis is to investigate the effects of motion during the measurement acquisition process, and to correct DT reconstructions with respect to motions that may occur in practical use cases. During a complete firing sequence of the emitters there may be small shifts in an object's position, as well as internal motion, or there may be some small motions from the device during the acquisition phase. Since the emitters are fired sequentially rather than simultaneously, the acquired dataset will then not correspond to a single static configuration of the body, but instead will represent a temporally interleaved collection of projections from multiple states. As a result, consecutive measurements are related by unknown transformations, and the body may undergo changes in position, orientation, or even internal deformation between emitter firings.

Motion will therefore inevitably alter the measurements obtained, relative to those that would be acquired if the patient remained stationary, leading to a degradation in reconstruction accuracy if such effects are not accounted for. In more extreme cases, motion may be sufficiently drastic that conventional reconstruction methods fail altogether, due to the fundamental disagreement between projections regarding the location and structure of the object being imaged. The work included in this thesis is intended to be an initial study of simple motions—namely, rigid-body motions—and their effects on both feasibility of correction, and clarity of reconstruction. From certain perspectives, while the device being modelled is designed to *not* need mechanical motion, many of the results in Part II may be interpreted as motion being induced by the device setup, rather than the object being imaged.

2.2 Model and geometry assumptions

In this thesis we discuss results for both two- and three-dimensional reconstructions, and so we present the formulation of both geometries and models here. Although the

two share a common structure, we present both explicitly to avoid ambiguity, referring to them as DG-2 for the two-dimensional setup and DG-3 for the three-dimensional setup.

2.2.1 DG-2: Two-dimensional device geometry

To simplify the acquisition geometry while retaining its essential features, we restrict our attention to a two-dimensional setting, shown schematically in Figure 2.8.

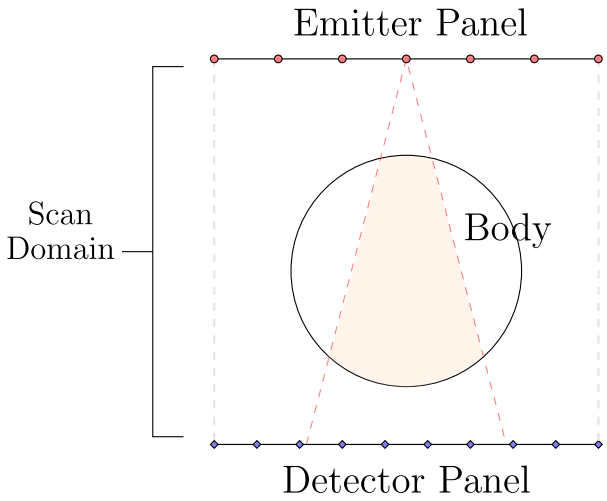


Figure 2.8: Two-dimensional schematic with fan-beam geometry. As in the three-dimensional model, any single emitter will generally not measure the entire body.

The emitter device is used in conjunction with a high-resolution detector panel, with the device and panel positioned opposite one another and having parallel facing normals[15]. In general, the detector panel need not be the same size as the emitter device, in which case the region bounded by the two panels would be trapezoidal. In this work, however, we assume that the device and panel have identical dimensions, and model the bounded region as a rectangle, which we refer to as the *scan domain*[16]. Throughout the modelling that follows, we work with a normalised, dimensionless formulation of the acquisition process. Spatial coordinates are measured relative to the scan domain geometry, and X-ray intensity values are treated as normalised quantities rather than absolute physical units. All physical constants and unit conversions may be absorbed into the attenuation coefficients without loss of generality, as only relative attenuation is required for reconstruction.

[15]This alignment is assumed throughout this work and can be controlled in practical deployments.

[16]Some terminology, such as 'scan domain', is used interchangeably in the two- and three-dimensional settings, with the intended model clear from context.

We model the scan domain as having height h and width w, and use the coordinate pair $(x, z) \in [0, w] \times [0, h]$ to denote locations inside the scan domain. Within the scan domain, there exists an object-of-interest to be reconstructed, which we refer to simply as *the object* or *the body*. The object can be represented as a density field $\rho(x, z)$, which may either be continuous or piecewise-continuous. We denote by $\mathcal{B} \subset [0, w] \times [0, h]$ the support of the object, that is, the subset of the scan domain on which $\rho(x, z)$ is non-zero. The density field over the entire scan domain may then be written as

$$\widetilde{\rho}(x, z) = \begin{cases} \rho(x, z) & \text{for } (x, z) \in \mathcal{B}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.34}$$

The object may have the topological structure of a simply connected subset of the scan domain, or it may be a union of multiple disjoint components; no restrictive assumptions are imposed on its topology. However, in line with standard practice in medical imaging, we assume that the object consists of finitely many regions of approximately homogeneous density, excluding pathologically complex configurations.

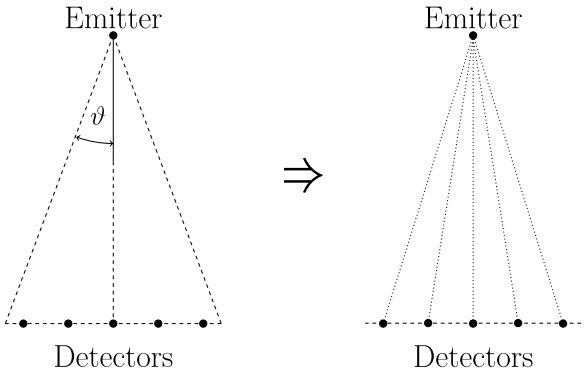


Figure 2.9: Each radiation fan is discretised by half-lines connected to the detectors that are contained within its projection onto the detector panel.

Each emitter projects a fan-beam of radiation onto the detector panel, where intensity measurements are made by all detectors that fall within this projection. For modelling purposes, a fan-beam is defined by its collimation half-angle ϑ and the location of its point source, which together determine the illuminated region of the detector panel. Emitters and detectors are modelled as idealised point sources and point sinks, and the continuum of X-rays is modelled as a discrete collection of geometric rays connecting each emitter–detector pair. A fan-beam may therefore be represented by a finite set of half-lines originating at the emitter and passing through selected detector points at the base of the fan, as illustrated in Figure 2.9.

This modelling approach does not explicitly account for the effects of ray scattering; such phenomena may instead be incorporated as noise terms within the later algebraic formulation of the reconstruction problem, if required.

During measurement acquisition, we assume that no overlap occurs between fan-beams from different emitters. This is akin to constraining all emitters to fire at distinct times, or stating that emitters which fire at the same time are positioned such that their fan-beams do not overlap; in this work we focus only on the first interpretation. Reconstruction methods based on overlapping measurements have been studied previously by Klodt and Hauser [86]; however, all results presented in this thesis are obtained under the simplifying assumption of non-overlapping beams.

As illustrated by Figure 2.8, no single fan-beam needs to completely cover the body, and a sequence of emitters is therefore required to measure the body in its entirety. Since accurate reconstruction requires each part of the body to be measured multiple times from different directions, the firing of additional emitters typically increases the total information content of the acquired data, except in cases where a fan-beam does not intersect the body at all. While it may be possible to identify a subset of emitters sufficient for coverage, we do not impose such a restriction, and instead assume that all emitters are sequenced during image acquisition.

For a given configuration of emitters and detectors, there will, in general, exist regions of the scan domain that are not intersected by any fan-beam, and hence are not measured[17]. We therefore assume that no relevant object structure lies within such regions, i.e. the body is positioned so as to be fully contained within the region covered by the union of the fan-beams.

Each measurement acquired by the device corresponds to a collection of rays emanating from a single emitter and terminating on the detector panel, and may therefore be described using the fan-beam parametrisation introduced in Section 2.1.4.1. In particular, each ray can be indexed either by an emitter–detector pair or, equivalently, by a suitable pair $(\mathbf{c}, \mathbf{v})$ and mapped to standard fan-beam parameters (r, α, β) via a geometry-specific reparametrisation. Consequently, the geometric considerations discussed previously—such as incomplete angular coverage and limited visibility—remain formally applicable in the present setting, with the additional complication that different emitters correspond to distinct centres of projection.

[17] Complete coverage of the scan domain occurs only in the limiting case $\vartheta = \pi/2$.

2.2.2 DG-2: Two-dimensional domain geometry

While the object is assumed to be described by a density field $\rho(x, z)$, numerical reconstruction requires discretisation of the scan domain. For many DT systems the domain discretisation is chosen to be anisotropic due to the limited angular range, as this causes information in the z-direction to be measured less effectively than in the x-direction; however, in this work we employ isotropic pixels. As will be demonstrated in Part II, when accounting for certain motion effects an isotropic discretisation can be adopted with little loss in either computational performance or reconstruction accuracy. However, we note that this will generally not be the case for all motions, due to the geometric physical constraints of the device itself.

We will model the scan domain with isotropic square pixels of side length Δ, with N_x and N_z denoting the number of pixels in the horizontal and vertical directions respectively, i.e. $\Delta = w/N_x = h/N_z$. We define the pixel p_{n_x,n_z}, for $n_x = 1, \dots, N_x$ and $n_z = 1, \dots, N_z$, as the set

$$p_{n_x,n_z} = \{\, \mathbf{r} \in \mathbb{R}^2 \mid \|\mathbf{r} - \mathbf{p}_{n_x,n_z}\|_\infty < \Delta/2 \,\}, \tag{2.35}$$

where $\mathbf{p}_{n_x,n_z} := \frac{\Delta}{2}(2n_x - 1, 2n_z - 1)^\top$ represents the centre of the pixel. The strict inequality in (2.35) means that all pixels are open subsets of the scan domain, and are chosen such that

$$\bigcup_{\substack{1 \le n_x \le N_x \\ 1 \le n_z \le N_z}} \overline{p}_{n_x,n_z} = [0, w] \times [0, h]\,, \tag{2.36}$$

where $\overline{p}_{n_x,n_z}$ denotes the closure of pixel p_{n_x,n_z}.

For reasons that will become clear, we may also enumerate pixels over an index $k = 1, \dots, n = N_x N_z$, such that the map $k(n_x, n_z) = n_x + (n_z - 1)N_x$ forms a correspondence between the two-dimensional pixel orientation and a vectorisation of the pixel indices—this correspondence then allows us to refer to specific pixels

$$p_k = p_{k(n_x,n_z)} := p_{n_x,n_z}. \tag{2.37}$$

We discretise the scan domain using the two-dimensional pixelisation defined in (2.35), and seek a discrete approximation to the density field $\rho(x, z)$ in which each pixel p_{n_x,n_z} is assumed to be homogeneous. Under this assumption, the density field may be represented as a finite collection of constant-valued pixels, allowing X-ray attenuation to be modelled as a finite-dimensional linear operator; this will be discussed further in Section 2.2.3.

By convention, the density field $\rho(x, z)$ is taken to be strictly positive on the support of the object—in the discrete form, this corresponds to allowing pixel densities to take only non-negative values. The spatial support of the reconstructed object is not prescribed in advance, but is determined implicitly as part of the reconstruction process. Pixels assigned zero density in the reconstruction correspond to regions outside the support of the object[18], consistent with the continuum model described in (2.34).

We denote the total number of emitters and detectors to be n_E and n_D respectively, and for each index-pair $(i, j) \in \{1, \ldots, n_E\} \times \{1, \ldots, n_D\}$ define the emitter–detector pair (e_i, d_j) which is connected by a unique ray, denoted by r_{ij}.

The initial X-ray intensity at emitter e_i is denoted by $\mathcal{I}_i$, and the corresponding measurement recorded by detector d_j is denoted by $\mathcal{I}_{ij}$. The detected intensity $\mathcal{I}_{ij}$ is indexed by both emitter and detector, while the emitted intensity depends only on the emitter index. Assuming all emitters have equal output power[19], the global emitter intensity is defined as $\mathcal{I}_0$, i.e. $\mathcal{I}_i = \mathcal{I}_0$ for all $i \in \{1, \ldots, n_E\}$.

Intensity measured by a detector will be less than or equal to the emitted intensity, due to photon–matter interactions, meaning $\mathcal{I}_{ij} \leq \mathcal{I}_0$ for all index-pairs (i, j). As scattering effects are not modelled explicitly, any reduction in intensity is attributed solely to absorption. In particular, if the ray r_{ij} does not intersect any region of non-zero density, we assume that $\mathcal{I}_{ij} = \mathcal{I}_0$.

2.2.3 A model for X-ray attenuation

While a variety of models exist, the dominant contribution to X-ray attenuation comes from the material-dependent attenuation coefficient, integrated along the ray path. In this work, energy-dependent scattering and interaction effects are neglected, as they are not required for the reconstruction framework considered here.

[18] This choice may also be interpreted as setting the background attenuation to zero, effectively normalising measurements relative to the surrounding medium. This convention does not preclude the possibility of reconstructing multiply-connected objects; rather, any holes or cavities are regarded as lying outside the object itself.

[19] This assumption is not vital for the modelling process, nor indeed the reconstruction process in general. If instead it were the case that each emitter had a particular strength, perhaps within some tolerance band, then this would merely cause a multiplicative effect in the attenuation process which could be normalised if the individual strengths were either known, or well-approximated. It should be noted, however, that non-uniform emitter strengths will affect the noise characteristics of the measurements; a flat-field correction can account for the intensity variation, but will also alter the effective noise level in the corrected data.

We will model X-ray attenuation under the following assumptions:

A1. For a fixed X-ray energy, the linear attenuation coefficient, $\mu(\mathbf{r})$, is spatially dependent and proportional to the material density, $\rho(\mathbf{r})$;

A2. Photons propagate along straight-line trajectories which can be modelled as geometric rays;

A3. X-ray intensity may be treated as a continuous, deterministic quantity along each ray.

Under these assumptions, the attenuation process is modelled using the *Beer–Lambert law* for X-ray absorption [49]. We fix a beam ℓ and parametrise it by arc length s such that any point $\mathbf{r} \in \ell$ is of the form

$$\mathbf{r}(s) = \mathbf{p}_0 + s\mathbf{v}, \tag{2.38}$$

for the unit direction vector $\mathbf{v}$, and for some fixed point $\mathbf{p}_0 \in \ell$. The Beer–Lambert law states that an infinitesimal *relative* decrease in the beam intensity $\mathcal{I}$ at a point $\mathbf{r} \in \ell$ is directly proportional to the distance travelled along the beam. Therefore, $\mathcal{I}$ may be regarded as a function of the arc length s, satisfying

$$\frac{\Delta\mathcal{I}}{\mathcal{I}} = -\mu(\mathbf{r}(s))\Delta s. \tag{2.39}$$

Taking the limit $\Delta s \to 0$, this leads to the linear differential equation

$$\frac{\mathrm{d}\mathcal{I}}{\mathrm{d}s} = -\mu(\mathbf{r}(s))\mathcal{I}. \tag{2.40}$$

If we assume an initial intensity $\mathcal{I}_0$, then for an X-ray intersecting a body $\mathcal{B}$ which has a spatially dependent attenuation coefficient $\mu(\mathbf{r})$, the attenuated intensity is described by

$$\mathcal{I} = \mathcal{I}_0 \exp\left(-\int_\ell \mu(\mathbf{r})\, \mathrm{d}s\right), \tag{2.41}$$

where $\mathcal{I}$ is the measured intensity after attenuation, and ℓ is the path of the X-ray beam. In the event that the measured body is entirely homogeneous, the linear attenuation coefficient is constant within the body, and so $\mu(\mathbf{r}) = \mu\mathbb{1}_{\mathcal{B}}(\mathbf{r})$, where $\mathbb{1}_{\mathcal{B}}$ denotes the indicator function for $\mathcal{B}$, and $\mu \in \mathbb{R}$. From this, the integral in (2.41) simplifies to

$$\mathcal{I} = \mathcal{I}_0 \exp\left(-\int_\ell \mu\mathbb{1}_{\mathcal{B}}(\mathbf{r})\, \mathrm{d}s\right), \tag{2.42}$$

$$= \mathcal{I}_0 \exp\left(-L\mu\right), \tag{2.43}$$

where L denotes the intersection length of the ray through the body.

For a single homogeneous pixel p_k, let a_{ij}^k denote the length of intersection between the ray r_{ij} and the pixel, and let x_k denote[20] the constant attenuation coefficient associated with p_k—see Figure 2.10. Under this assumption, the Beer–Lambert law reduces to

$$\mathcal{I}_{ij} = \mathcal{I}_0 \exp\left(-a_{ij}^k x_k\right). \tag{2.44}$$

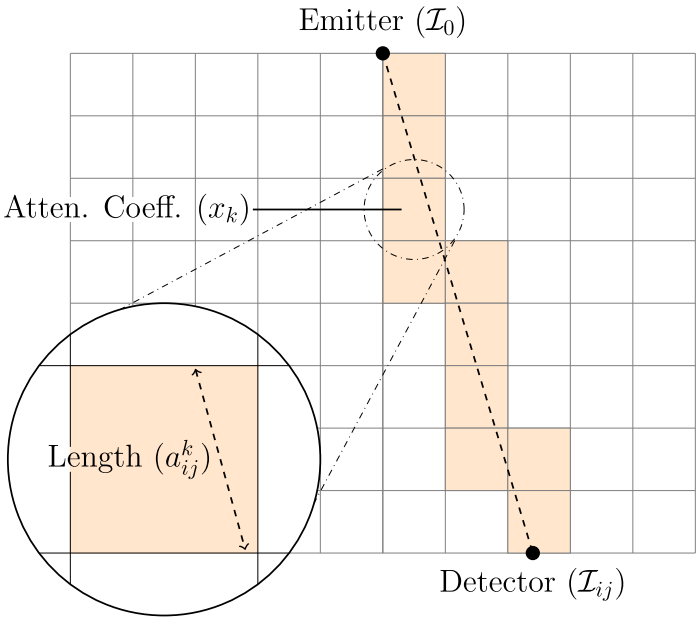


Figure 2.10: The Beer–Lambert law states that the measured intensity $\mathcal{I}_{ij}$ is proportional to $\mathcal{I}_0$, with a multiplicative factor depending on the intersection lengths, a_{ij}^k, and attenuation coefficients, x_k, of the intersected cells.

In making the problem tractable, i.e. by attempting to reconstruct a discrete attenuation field[21], the integral term in (2.41) is instead approximated by a finite sum of homogeneous contributions. In a typical imaging acquisition, each ray will travel through multiple pixels before reaching the detector panel, with each pixel contributing a distinct exponential term for its particular attenuation contribution. Therefore, the combined effect results in the equation

$$\mathcal{I}_{ij} = \mathcal{I}_0 \prod_{k=1}^{n} \exp\left(-a_{ij}^k x_k\right) = \mathcal{I}_0 \exp\left(-\sum_{k=1}^{n} a_{ij}^k x_k\right), \tag{2.45}$$

[20]The change is for notational convenience, in anticipation of the linear system introduced later.

[21]While we have previously discussed the density field $\rho(\mathbf{r})$, the use of the Beer–Lambert law means that we will be attempting to find the underlying *attenuation* field, $\mu(\mathbf{r})$, which is proportional to $\rho(\mathbf{r})$, under Assumption A1. We will refer to the attenuation field as the object of interest from here onwards.

where n is the total number of pixels in the scan domain, i.e. $n = N_x N_z$. Note that if a particular pixel p_k is not intersected by ray r_{ij} then $a_{ij}^k = 0$ and it contributes nothing to the sum in (2.45).

2.2.4 DG-3: Three-dimensional device geometry

Here we extend the two-dimensional model described in Sections 2.2.1 to 2.2.3 into three dimensions. Unless otherwise stated, all modelling assumptions and notation are inherited directly.

The device is used in conjunction with a high-resolution detector panel, with the device and panel positioned opposite one another on distinct parallel planes. For the purposes of this work, we assume that the device and detector panel have identical dimensions, so that the region bounded between them forms a cuboidal volume, which we will also refer to as the scan domain. We model the scan domain as a cuboid of height h and rectangular cross-sectional area $w_1 \times w_2$, and discretise it using isotropic cubic voxels.

As in the two-dimensional case, no *a priori* assumptions are imposed on the body beyond it being bounded and topologically reasonable. The body can be represented as a density field $\rho(x, y, z)$, which may either be continuous or piecewise-continuous, and we overload the notation $\mathcal{B}$ to also denote the support of the body, such that $\mathcal{B} \subset [0, w_1] \times [0, w_2] \times [0, h]$. The density field over the entire scan domain may then be written as

$$\widetilde{\rho}(x, y, z) = \begin{cases} \rho(x, y, z) & \text{for } (x, y, z) \in \mathcal{B}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.46}$$

Each emitter projects a cone of radiation onto the detector panel, with intensity measurements recorded by all detectors that lie within the conic projection. The radiation cone is characterised by its collimation half-angle ϑ, which determines the extent of the projection onto the detector panel. Emitters and detectors are modelled in the same way as DG-2. Under this idealisation, a radiation cone can be represented as a finite collection of half-lines, each originating at the emitter and passing through a detector element contained within the conic projection—see Figure 2.11.

During measurement acquisition, we assume that radiation cones from different emitters do not overlap. As shown in Figure 2.7, a single radiation cone is not guaranteed to fully cover the body, and so multiple emitter positions are required to achieve volumetric coverage. Accordingly, we assume that all emitters in the device

are sequenced during image acquisition. We also assume that the body is positioned such that it is fully contained within the union of all radiation cones.

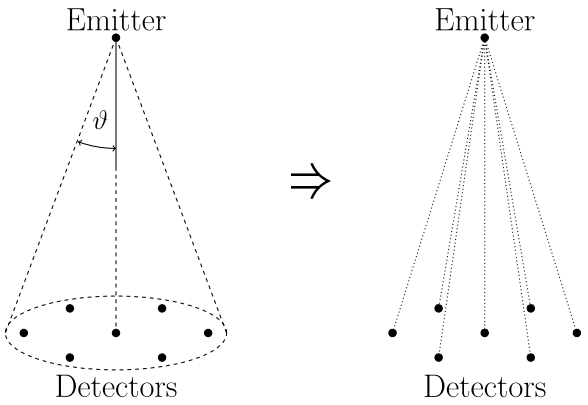


Figure 2.11: Each radiation cone is discretised by half-lines connected to the detectors that are contained within its projection onto the detector panel.

2.2.5 DG-3: Three-dimensional domain geometry

We model the scan domain with isotropic voxels of side length Δ, with N_x, N_y, and N_z denoting the number of voxels in the axial directions respectively, i.e. $\Delta = w_1/N_x = w_2/N_y = h/N_z$. We define the voxel v_{n_x,n_y,n_z}, for $n_x = 1,\dots,N_x$, $n_y = 1,\dots,N_y$, and $n_z = 1,\dots,N_z$, as the set

$$v_{n_x,n_y,n_z} = \{\, \mathbf{r} \in \mathbb{R}^3 \mid \|\mathbf{r} - \mathbf{q}_{n_x,n_y,n_z}\|_\infty < \Delta/2 \,\}, \tag{2.47}$$

where $\mathbf{q}_{n_x,n_y,n_z} := \frac{\Delta}{2}(2n_x - 1, 2n_y - 1, 2n_z - 1)^\top$ represents the centre of the voxel. The strict inequality in (2.47) means that all voxels are open subsets of the scan domain, and so

$$\bigcup_{\substack{1\le n_x\le N_x\\ 1\le n_y\le N_y\\ 1\le n_z\le N_z}} \overline{v}_{n_x,n_y,n_z} = [0,w_1]\times[0,w_2]\times[0,h]\,, \tag{2.48}$$

where $\overline{v}_{n_x,n_y,n_z}$ denotes the closure of voxel v_{n_x,n_y,n_z}.

For notational convenience, voxels may also be identified by a single index $k = 1,\dots,n$, where $n = N_xN_yN_z$, and we write v_k when the specific axial indices are not required.

The sole distinction from the two-dimensional case is that rays now intersect three-dimensional voxels rather than planar pixels. The attenuation model in three dimensions is identical to that of the two-dimensional case, and we therefore adopt the

Beer–Lambert law without modification. For a discretised voxel model, the measured intensity corresponding to the ray r_{ij} is given by

$$\mathcal{I}_{ij} = \mathcal{I}_0 \prod_{k=1}^{n} \exp\left(-a_{ij}^k x_k\right) = \mathcal{I}_0 \exp\left(-\sum_{k=1}^{n} a_{ij}^k x_k\right), \tag{2.49}$$

where x_k denotes the attenuation coefficient of voxel v_k. Here, a_{ij}^k represents the length of intersection between the ray r_{ij} and the three-dimensional voxel v_k; if the ray does not intersect the voxel then $a_{ij}^k = 0$.

2.3 The tomographic reconstruction problem

For both DG-2 and DG-3, by dividing both sides of (2.49) by $\mathcal{I}_0$ and taking the natural logarithm we obtain

$$\ln\left(\frac{\mathcal{I}_{ij}}{\mathcal{I}_0}\right) = -\sum_{k=1}^{n} a_{ij}^k x_k, \tag{2.50}$$

for each ray r_{ij}. The complete set of measurements can therefore be represented by a system of linear equations, $\mathbf{Ax} = \mathbf{b}$, for which

$$\mathbf{A} = \left(a_{ij}^k\right), \quad \mathbf{x} = \left(x_k\right), \quad \text{and} \quad \mathbf{b} = \left(\ln\left(\mathcal{I}_0/\mathcal{I}_{ij}\right)\right). \tag{2.51}$$

In this context, rows of $\mathbf{A}$ correspond to rays r_{ij}, while columns correspond to voxels[22] in the discretised scan domain, and, under the absorption-only model, each component of $\mathbf{b}$ is non-negative.

Since each ray can only intersect voxels along its path—that is, only a very small proportion of the total set of voxels—each row of the matrix $\mathbf{A}$ is sparse[23]. In a similar sense, each voxel will likely be intersected by only a small proportion of the total set of rays, and so each column of $\mathbf{A}$ will also be sparse. In this case, however, the sparsity pattern is spatially non-uniform, with voxels near the centre of the domain being intersected more frequently than those towards the boundary. The resulting sparsity structure enables the system to remain computationally tractable, even for large-scale problems.

[22] As both DG-2 and DG-3 have similar setups and shared notation, we will invariably use the term voxels to refer to both voxels *and* pixels, with the choice being clear from context. Should we need to stress particular information about either voxels *or* pixels separately, it will be made obvious.

[23] The number of non-zero entries in any row is bounded above by $(N_x + N_z - 1)$ for DG-2, and bounded above by $(N_x + N_y + N_z - 2)$ for DG-3.

From a clinical perspective, any image acquisition process is subject to a fixed maximum radiation dose. This naturally imposes a constraint on the total number of measurements, corresponding to a fixed number of X-rays and hence a fixed number of rows in the system matrix $\mathbf{A}$. However, in medical imaging it is often desirable to obtain a reconstruction at the highest possible spatial resolution, motivating an increase in the number of voxels that can be detected and distinguished. This corresponds to increasing the number of columns of $\mathbf{A}$, and to the consideration of increasingly underdetermined linear systems in order to resolve finer structural detail. We therefore primarily consider system matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ where $m < n$, while typically keeping m and n of comparable order[24].

2.3.1 Approaches to tomographic reconstruction

With the linear system $\mathbf{Ax} = \mathbf{b}$ established, we now survey the principal classes of algorithms that have been proposed for its solution. These differ both in their theoretical foundations and in the assumptions they make about the structure of the problem. We begin with analytical methods, which operate directly on the continuous projection data, before discussing algebraic iterative methods, which treat image reconstruction as a numerical linear algebra problem. We will then discuss the ideas behind ℓ_1-constrained approaches, which introduce explicit sparsity regularisation and form the basis of the reconstruction framework adopted in this work.

2.3.1.1 Analytical backprojection and the FDK algorithm

Historically, the dominant class of reconstruction algorithms has been analytical methods, which operate directly on the continuous projection data and invert the forward model via an explicit formula. The canonical example in two-dimensional parallel-beam tomography is filtered backprojection (FBP) [83]—briefly mentioned in Section 2.1.3.1—which recovers a signal $g(x, y)$ from its Radon transform by applying a ramp filter in the frequency domain, followed by a backprojection step that smears filtered projections back across the image domain. The ramp filter compensates for the over-representation of low-frequency components that occurs under simple backprojection, and the resulting algorithm is exact in the limit of continuous, noiseless data with full angular coverage.

[24] In numerical experiments—discussed in Part II—the system is typically undersampled with the ratio m/n ranging between approximately 10% and 30%, depending on the acquisition geometry.

For cone-beam CT (CBCT)—the acquisition geometry most relevant to our work—the equivalent analytical reconstruction is the Feldkamp–Davis–Kress (FDK) algorithm [51]. FDK is an approximate extension of FBP to three dimensions under a circular source trajectory, and consists of three steps: weighting each projection to account for cone-beam divergence; convolving each projection with a ramp filter; and back-projecting the filtered projections into the reconstruction volume, with a further distance-dependent weighting. FDK is exact within the central plane of the circular trajectory, and provides a good approximation for objects with limited extent in the axial direction. However, it introduces cone-beam artefacts for objects with significant out-of-plane variation and is, in general, only approximate for arbitrary three-dimensional objects.

Neither FBP nor FDK is suited to the acquisition geometry considered here, as both require a full, regular angular sampling of projection data, and artefacts arising from angular undersampling cannot be corrected analytically. In the discrete setting, the backprojection operation corresponds to multiplication by $\mathbf{A}^\top$—it is an adjoint operation, not an inversion, and so $\mathbf{A}^\top\mathbf{b}$ is often blurry and low-contrast. This motivates instead solving the linear system $\mathbf{Ax} = \mathbf{b}$ directly, which we now discuss.

2.3.1.2 Algebraic iterative reconstruction methods

A natural alternative to approximating analytical inversion is to treat the reconstruction problem directly as a linear system $\mathbf{Ax} = \mathbf{b}$ and apply iterative methods to determine a solution. This class of methods operate by repeatedly refining an estimate $\mathbf{x}^{(k)}$ using the residual $\mathbf{b} - \mathbf{Ax}^{(k)}$, and are well-suited to the sparse, large-scale systems that arise in tomographic reconstruction. The foundational method in this class is the Algebraic Reconstruction Technique (ART) [60], equivalent to the Kaczmarz method [82] for which a thorough treatment can be found in [69]. ART processes rows of $\mathbf{A}$ sequentially, projecting the current estimate onto the hyperplane defined by each equation in turn:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda \frac{b_i - \langle \mathbf{a}_i, \mathbf{x}^{(k)} \rangle}{\|\mathbf{a}_i\|_2^2} \mathbf{a}_i, \tag{2.52}$$

where $\mathbf{a}_i$ denotes the ith row of $\mathbf{A}$, b_i the corresponding measurement, $\lambda > 0$ is a relaxation parameter, and $i = k \pmod m + 1$ for vectors $\mathbf{x}^{(k)} \in \mathbb{R}^n$. While ART can converge rapidly, its sequential row-by-row structure means the image must be updated once per row per iteration, which is computationally expensive for systems with large numbers of measurements.

The Simultaneous Algebraic Reconstruction Technique (SART) [5], along with its extensions[25], improves upon the classical algorithm by processing groups of rays collectively. The standard SART update is then

$$\mathbf{x}_j^{(k+1)} = \mathbf{x}_j^{(k)} + \lambda \frac{1}{\sum\limits_{i \in \Omega} \mathbf{a}_{ij}} \sum_{i \in \Omega} \frac{\mathbf{a}_{ij}}{\|\mathbf{a}_i\|_1} \left(b_i - \langle \mathbf{a}_i, \mathbf{x}^{(k)} \rangle \right), \tag{2.53}$$

where Ω is a subset of projections that interact with the reconstruction element $\mathbf{x}_j^{(k)}$. Here, we note that the term $\|\mathbf{a}_i\|_1$ describes the total intersection length of the ray associated with $\mathbf{a}_i$, while $\sum_{i \in \Omega} \mathbf{a}_{ij}$ describes the partial column sum of rays which interact with element $\mathbf{x}_j$. Therefore, one may update *all* terms at once via

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda \, \mathbf{D}_{\mathrm{C}}^{-1} \mathbf{A}^\top \mathbf{D}_{\mathrm{R}}^{-1} \left(\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)} \right), \tag{2.54}$$

where $(\mathbf{D}_{\mathrm{C}})_{jj} = \sum\limits_{i=1}^{m} \mathbf{a}_{ij}$ and $(\mathbf{D}_{\mathrm{R}})_{ii} = \sum\limits_{j=1}^{n} \mathbf{a}_{ij}$ are diagonal matrices which normalise against column norms and row norms, respectively. By enabling large-scale updates of $\mathbf{x}^{(k)}$, SART retains the convergence guarantees of ART [79] while greatly reducing the amount of noise in the final reconstruction, resulting in much better image quality in practical reconstruction.

A computationally attractive alternative is provided by Krylov subspace methods, in particular the Conjugate Gradient for Least Squares (CGLS) algorithm which minimises the ℓ_2 residual $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ over a sequence of Krylov subspaces $\mathcal{K}_p(\mathbf{A}^\top \mathbf{A}, \mathbf{A}^\top \mathbf{b})$. This method is equivalent to applying the conjugate gradient method to the normal equations $\mathbf{A}^\top \mathbf{A}\mathbf{x} = \mathbf{A}^\top \mathbf{b}$ without forming them explicitly, and the LSQR algorithm [114] is the standard numerically stable implementation, being equivalent to CGLS in exact arithmetic. Both methods are well-suited to large sparse systems where only matrix-vector products with $\mathbf{A}$ and $\mathbf{A}^\top$ are available.

All of the methods described above seek to minimise a data fidelity term alone, without any mechanism for incorporating prior information about the structure of $\mathbf{x}$. In the underdetermined setting—where $m < n$ and the system admits infinitely many solutions consistent with the data—this is insufficient. The solution selected will depend on the initialisation and the implicit geometry of the iterative scheme or, in the case of Krylov methods, on the choice of stopping criterion, with no guarantee of physical interpretability. This motivates the introduction of explicit regularisation, which we discuss in the following subsection.

[25] Further variants of SART include the *Ordered-Subset* method (OS-SART) [148], the *Frequency-Adaptive* method (FA-SART) [76], and the *Variable-Weighted* method (VW-OS-SART) [115], among others.

2.3.1.3 Compressed sensing and ℓ_1-regularised approaches

A principled approach to regularising the underdetermined system $\mathbf{Ax} = \mathbf{b}$ is provided by compressed sensing theory [23, 40], which establishes conditions under which sparse signals can be exactly or stably recovered from a small number of linear measurements. A key requirement is that the measurement operator $\mathbf{A}$ satisfies certain incoherence or restricted isometry conditions, which are typically guaranteed when $\mathbf{A}$ is constructed from random elements which are drawn from a suitable distribution or ensemble.

Previous work undertaken by Patel [116], in partnership with Adaptix, exploits this by allowing emitters and detectors to take uniformly distributed random positions within unit square panels, with collimation half-angles drawn uniformly from the interval $[0, \pi/2]$. This randomness permits the use of compressed sensing theory in parts of the reconstruction analysis, for example through properties such as the restricted isometry property of the system matrix $\mathbf{A}$. In the present work, however, the acquisition geometry is fixed and deterministic, reflecting the physical structure of the Adaptix device—we therefore cannot appeal directly to compressed sensing recovery theory. Nevertheless, sparsity remains a well-motivated and practically effective regularising assumption for medical images, provided that the discrete attenuation distribution admits a sparse representation under an appropriate transform. That is, while the attenuation field itself need not be sparse in the voxel basis, its representation in a suitable transformed domain—such as gradients or wavelet bases—may be. We make no claim that the resulting system satisfies the conditions under which compressed sensing recovery guarantees are typically derived. Rather, sparsity is used here as a regularising assumption reflecting the underlying structure of medical images, which is well-reported in the medical imaging literature.

A direct enforcement of sparsity would require minimisation of the ℓ_0-seminorm[26], which counts the number of non-zero components of a discrete signal. Such formulations lead to combinatorial optimisation problems that are computationally intractable in general. A well-established result in convex optimisation [25, 26, 41, 42] is that the ℓ_1-norm provides the tightest convex relaxation of the ℓ_0-seminorm, and so minimising the ℓ_1-norm promotes sparse solutions while remaining tractable for efficient optimisation. This insight forms the basis of a wide class of sparsity-regularised inverse problems and motivates the use of ℓ_1-based regularisation in the present setting.

[26] Strictly speaking, this is not a true seminorm as it fails to satisfy positive homogeneity. However, the term *seminorm* is nevertheless standard in the optimisation literature and we adopt it here.

With this in mind, our task is to obtain $\mathbf{x}^*$ as a solution to the following problem:

$$\mathbf{x}^* \in \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^n} \left\{ \lambda \|\mathbf{\Phi x}\|_1 + \frac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|_2^2 \right\} \quad \text{subject to } \mathbf{x} \geq \mathbf{0}, \tag{2.55}$$

where $\mathbf{\Phi}$ is a sparsifying transform to be specified later, and $\lambda > 0$ is a regularisation parameter. The non-negativity constraint $\mathbf{x} \geq \mathbf{0}$ reflects the physical assumption that attenuation coefficients are non-negative in our model. The quadratic data fidelity term in (2.55) arises naturally from a statistical interpretation of the data acquisition process. If the observed data are modelled as the true line integrals corrupted by additive Gaussian noise, then minimising the ℓ_2-norm of the residual corresponds to a maximum-likelihood estimate of the underlying attenuation field. Although X-ray photon counts are fundamentally Poisson distributed, the combination of logarithmic preprocessing and photon aggregation leads to an effective Gaussian noise model at the reconstruction stage. The ℓ_2 term therefore provides a principled and widely adopted measure of data consistency in algebraic reconstruction methods [49].

In the absence of the non-negativity constraint, the formulation in (2.55) is commonly referred to as *Basis Pursuit Denoising* (BPDN) [25, 26]. The value $\lambda > 0$ acts as a relaxation parameter, allowing solutions to deviate from satisfying the linear system in order to promote sparsity in the transformed coefficients $\mathbf{\Phi x}$. We emphasise the use of arg min to denote the set of global minimisers, as no assumptions are imposed on the problem structure that would guarantee uniqueness of the solution; in contrast, BPDN is often presented using min notation when uniqueness is implicitly assumed. BPDN may also be formulated as the related problem

$$\mathbf{x}^* \in \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{\Phi x}\|_1 \quad \text{subject to } \|\mathbf{Ax} - \mathbf{b}\|_2^2 \leq \varepsilon, \tag{2.56}$$

for a prescribed $\varepsilon \geq 0$.

The formulations in (2.55)–(2.56) are equivalent in the sense that, for any chosen value of $\lambda > 0$, the penalised problem in (2.55) corresponds to the constrained formulation in (2.56) for some (generally unknown) value of ε, and vice versa. When $\varepsilon = 0$, the constrained formulation in (2.56) reduces to the standard *Basis Pursuit* problem

$$\mathbf{x}^* \in \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{\Phi x}\|_1 \quad \text{subject to } \mathbf{Ax} = \mathbf{b}. \tag{2.57}$$

In the statistical literature, a related formulation is given by the non-negative LASSO problem [139], in which sparsity is enforced through an explicit constraint on the

ℓ_1-norm of the transformed coefficients. In this setting, one seeks a solution

$$\mathbf{x}^* \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 \quad \text{subject to } \|\mathbf{\Phi}\mathbf{x}\|_1 \leq \xi \text{ and } \mathbf{x} \geq \mathbf{0}, \tag{2.58}$$

for some $\xi > 0$. This formulation generally requires prior knowledge about the sparsity level of the signal, encoded through the choice of ξ, whereas the penalised formulation in (2.55) offers greater flexibility by allowing sparsity to be controlled indirectly via the parameter λ.

Throughout this work we adopt the penalised formulation (2.55), as the regularisation parameter λ provides a practically convenient and interpretable mechanism for balancing data fidelity against sparsity, without requiring prior knowledge of the solution's sparsity level.

2.3.2 Regularisation and sparsity transforms

The formulation in (2.55) leaves the sparsifying transform $\mathbf{\Phi}$ unspecified, and the choice of transform is central to determining which class of images can be stably recovered. Several transforms have been proposed for this purpose in the tomographic reconstruction literature. Wavelet bases [102] provide a multiscale decomposition in which natural images are approximately sparse, and have been applied to image reconstruction with some success [24, 113]. More directionally sensitive representations, such as *curvelets* [21, 22] and *shearlets* [65], offer improved sparsity for images containing edges or elongated features at multiple orientations, and have been explored in limited-angle tomography settings [17, 153].

In this work, we restrict attention to a well-established choice in medical image reconstruction—the discrete spatial derivative operator, $\mathbf{\Phi} = \mathbf{D}$—as objects in medical imaging typically exhibit sparsity in their spatial gradients rather than in the image domain itself. This motivates the use of the operator $\mathbf{D}$, in which sparsity is imposed on local differences between neighbouring pixels or voxels. Penalisation through the term $\|\mathbf{D}\mathbf{x}\|_1$ corresponds to total variation (TV) regularisation, promoting reconstructions that are piecewise constant, with sharp interfaces between regions of differing attenuation. Such behaviour aligns naturally with the structure of many anatomical objects, which consist of approximately homogeneous regions separated by well-defined boundaries.

More expressive sparsity transforms, such as those mentioned above, can provide strong sparsity for certain image classes although they introduce additional modelling

choices, including basis selection and scale truncation, and their interaction with object motion is non-trivial. Since the focus of this work lies in geometric modelling and reconstruction under motion, we adopt the discrete derivative as a minimal and model-compatible sparsifying transform. The suitability of this choice for medical imaging is illustrated explicitly in Chapter 4, where representative phantoms are examined both in the image domain and under discrete differentiation.

2.4 Model limitations

The models described in this chapter have many assumptions and simplifications which may not accurately represent the underlying physical processes. We assume that the following considerations have negligible effect, apart from Section 2.4.4 which is discussed further in Part III of this thesis.

2.4.1 Monochromatic spectrum

In the forward model described in this chapter, we implicitly assume that the emitted X-ray spectrum is monochromatic, meaning all photons are treated as having the same energy. Under this assumption each photon interacts equivalently with matter, allowing the attenuation process to be described using a single, spatially dependent coefficient, $\mu(\mathbf{r})$.

A more general description of X-ray attenuation accounts for the full polychromatic energy spectrum of the X-ray source. In this case, the measured intensity $\mathcal{I}$ is given by the double integral form of the Beer–Lambert law

$$\mathcal{I} = \mathcal{I}_0 \int_E f(\epsilon) \exp\left(- \int_\ell \mu(\mathbf{r}, \epsilon)\, \mathrm{d}s \right) \mathrm{d}\epsilon, \tag{2.59}$$

where E represents the full energy spectrum of the source, $\mu(\mathbf{r}, \epsilon)$ is the attenuation coefficient depending on both spatial position and photon energy, and $f(\epsilon)$ represents the energy fluence distribution, related to both the X-ray energy and the number of photons occurring at that energy level.

After discretisation, (2.59) then leads to

$$\mathcal{I}_{ij} = \mathcal{I}_0 \sum_{l=1}^{N_\epsilon} f_l \exp\left(- \sum_{k=1}^{n} a_{ij}^k x_k(\epsilon_l) \right), \tag{2.60}$$

resulting in a nonlinear inverse problem that is significantly more challenging to solve. For this reason, most algebraic reconstruction methods adopt a linearised formulation

by assuming an effectively monochromatic spectrum—the resulting model is then what we present in (2.49).

While real X-ray sources are inherently polychromatic, the monochromatic approximation is widely used and provides a tractable zeroth-order model of the attenuation process. Polychromatic effects give rise to phenomena such as *beam hardening*, in which lower-energy photons are preferentially absorbed as the beam traverses matter, producing a spatially varying effective energy distribution [16]. In certain imaging contexts, beam hardening can introduce reconstruction artefacts that require explicit correction [72], but such effects are not modelled in this work.

2.4.2 Non-divergent beams

In modelling X-ray propagation, we represent each measurement as arising from an idealised geometric ray travelling along a straight line between an emitter and a detector. This abstraction is adopted in both fan-beam and cone-beam configurations, where the source emits photons over a finite collimation half-angle ϑ.

Physically, X-ray beams exhibit angular spread, and photon arrival at the detector is governed by stochastic processes that can be described using probabilistic models. In the present work, however, the beam geometry is interpreted as a continuum of infinitesimally narrow rays parametrised by their emission directions, rather than as a probabilistic distribution of photon trajectories. Under this interpretation, each ray follows a straight-line path, and divergence is captured geometrically through the family of ray directions rather than through beam broadening.

This modelling choice preserves the large-scale geometric structure of the acquisition process while avoiding the substantial complexity associated with particle transport or detector noise models. As a result, our forward model remains comparable to standard ray-based projection operators used in algebraic reconstruction methods.

2.4.3 No scattering or refraction

In this work, X-ray attenuation is modelled solely through absorption and penetration, while scattering and refraction effects are neglected. This assumption allows photon paths to be represented as idealised geometric rays and is consistent with the Beer–Lambert framework used in our model.

In practical imaging systems, scattering processes such as Compton and Rayleigh interactions introduce additional noise and can lead to artefacts in reconstructed

images [83]. Accurately accounting for these effects requires substantially more complex forward models, often involving stochastic particle transport or scatter correction schemes.

For the acquisition regimes considered in this thesis, empirical testing on stationary objects within the Adaptix system indicates that incorporating explicit scattering models increases computational cost without yielding a commensurate improvement in reconstruction quality.

2.4.4 Cubic voxels as basis elements

As described in Section 2.2, the discretised reconstruction problem assumes that the underlying attenuation field $\mu(\mathbf{r})$ can be approximated by a finite expansion over a collection of spatial elements. This may be written generally as an expansion in indicator basis functions $\{ b_\gamma \}_{\gamma=1}^{M}$ such that for any $\mathbf{r} \in \mathbb{R}^3$,

$$\mu(\mathbf{r}) \approx \sum_{\gamma=1}^{M} c_\gamma b_\gamma(\mathbf{r}), \tag{2.61}$$

where each coefficient c_γ represents the attenuation associated with a single voxel.

The cubic voxel basis is the near-universal choice in three-dimensional tomography owing to historical convention, as well as mathematical and computational convenience. Each voxel has compact support of equal size, with interiors satisfying

$$\operatorname{int}\left(\operatorname{supp}(b_{\gamma_i})\right) \cap \operatorname{int}\left(\operatorname{supp}\left(b_{\gamma_j}\right)\right) = \emptyset \qquad \text{for } \gamma_i \neq \gamma_j. \tag{2.62}$$

This structural simplicity allows reconstruction algorithms to perform local, piecewise updates affecting individual voxels, and leads to sparse system matrices with a transparent geometric interpretation. However, this simplicity comes at the cost of limited smoothness in the reconstructed field. Since neighbouring voxels do not overlap, there is no inherent mechanism for approximating continuity across voxel boundaries, and regularisation techniques are therefore required to mitigate block artefacts and noise amplification.

Beyond cubic voxels, a variety of alternative basis representations have been explored in the tomographic reconstruction literature. Overlapping radially symmetric bases, such as Kaiser–Bessel window functions [94, 95], offer improved smoothness and isotropy at the cost of strong coupling between neighbouring elements. B-spline bases provide a further class of smooth, compactly supported functions with controllable

regularity, and have been applied in both two- and three-dimensional reconstruction settings [15, 107].

Non-Cartesian tessellations can offer denser neighbourhood stencils and improved rotational symmetry, compared to their Cartesian counterparts, and have been studied in the context of improving isotropy and regularisation behaviour. In two dimensions, hexagonal pixel grids have received minor attention as an alternative to square pixels, with applications in both CT and PET reconstruction methodologies [1, 88]. Extending these developments to three dimensions is considerably less developed in the tomographic reconstruction literature, although the geometry of rhombic dodecahedral tessellation has been investigated in the context of digital image representation [11, 12]. A detailed treatment of both hexagonal and rhombic dodecahedral discretisations, and their integration into the algebraic reconstruction framework developed here, is presented in Part III.

2.4.5 Point-sink detector model

Throughout this work, each detector element is modelled as an idealised point sink, located at a lattice position described by the resolution of the detector grid being modelled. Under this idealisation, the intensity measurement associated with a given emitter–detector pair is taken to be the value of the integrated attenuation field at a single point, and each ray r_{ij} terminates precisely at the detector location d_j.

An alternative modelling approach is to include finite-area spatial averaging, such that each detector has a physical active area. Under this framework, each measurement is modelled as the integral of arriving rays which are projected onto this area, often convolved with a particular kernel, such as multidimensional $\mathrm{sinc}(\cdot)$ or $\mathrm{rect}(\cdot)$ functions. This then produces a detector blur operator that accounts for the finite aperture of a physical detector element and can, in principle, improve model accuracy when the detector size is non-negligible relative to the object discretisation.

In the present work, the focus is on modelling image acquisition through the construction of linear operator $\mathbf{A}$, where matrix elements represent exact geometric intersection lengths between rays and voxels. The inclusion of a spatial averaging kernel would act directly on the measurement vector $\mathbf{b}$, with a non-trivial relationship between the operator and the affected measurements. To maintain system consistency, the kernel could, in principle, be modelled as a convolution operator $\mathfrak{C}$, yielding the

modified system

$$\mathfrak{C}\mathbf{A}\mathbf{x} = \mathfrak{C}\mathbf{b}. \tag{2.63}$$

However, this convolution operator would naturally mix rows of $\mathbf{A}$, and it is not clear that $\mathfrak{C}\mathbf{A}$ would maintain the sparsity needed to be computationally tractable. Therefore, the point-sink model is retained, with any discrepancy from the physical detector area expected to contribute to the residual within the existing regularisation framework.

Collectively, the modelling choices made in this work prioritise geometric clarity and computational tractability, providing a controlled setting in which the effects of acquisition geometry and motion can be isolated and studied.

Chapter 3

Modelling approaches for motion correction

Contents

3.1 Overview of existing methods 49
3.2 A general parametric motion model 54
3.3 A nested optimisation formulation 60

This chapter introduces the modelling framework used to incorporate motion into the tomographic reconstruction problem. We begin by briefly situating our approach within the broader motion correction literature, to clarify the modelling choices adopted in this work. We then introduce a general, parametric description of object motion during data acquisition, together with a set of assumptions on the acquisition timescale that render the resulting inverse problem tractable. Motion is incorporated directly into the forward operator, leading to a family of motion-dependent measurement models indexed by admissible motion parameters. Finally, we reformulate the tomographic reconstruction problem into the structure of a nested optimisation problem, in which both the reference image and the motion parameters are unknown. This formulation provides a general framework that will be specialised in subsequent chapters to translational, rotational, and rigid-body motion models in both two and three dimensions.

3.1 Overview of existing methods

The correction of motion artefacts is a pervasive problem in medical imaging, arising across modalities whenever acquisition times are sufficiently long for patient or device motion to corrupt measurements. A wide range of correction strategies have therefore

been developed, differing both in the stage at which motion is addressed and in the assumptions made about its structure. Many approaches require additional investment in the imaging procedure, perhaps through specialised hardware or modified acquisition protocols, in order to obtain auxiliary information about the underlying motion.

Before surveying specific approaches, it is useful to distinguish between two conceptually distinct sub-problems that are often conflated in the literature. *Motion estimation* refers to the problem of determining what motion occurred during acquisition, whether from external hardware, additional measurements, or the acquired data itself. *Motion correction* refers to the problem of accounting for that estimated motion within the reconstruction. In practice, many methods address both problems simultaneously; however, maintaining this distinction is useful when assessing what information each approach requires and at what stage it becomes available. In this section, we briefly survey several commonly used classes of motion correction techniques, with the aim of situating the modelling choices adopted in this work within the broader literature.

3.1.1 Prospective motion correction

Prospective motion correction (PMC) methods aim to mitigate motion artefacts by incorporating motion information directly during data acquisition. In this setting, motion is measured, or inferred, in real time and used to adapt the acquisition process so that the resulting data are consistent with a fixed reference frame. Approaches range from explicit motion tracking of the patient [155], to in-situ compensation strategies that dynamically modify the acquisition geometry or sampling trajectory [151]. Motion tracking is most commonly achieved using optical systems that require direct coupling to the patient, typically in conjunction with fiducial markers, though in certain modalities electromagnetic or signal-based measurements may also be exploited to provide real-time feedback. While PMC techniques can be highly effective, their applicability is often constrained by additional hardware requirements and modality-specific acquisition constraints. PMC methods are well-suited to magnetic resonance imaging (MRI) [100, 154], as data acquisition is often slower than other modalities and trajectories can be modified dynamically with greater ease.

PMC typically requires dedicated tracking systems that must be tightly integrated with the acquisition hardware, and relies on the assumption that motion can be measured accurately at acquisition time. In contrast, in this work we model a fixed

acquisition geometry that does not rely on access to specialised tracking hardware, nor real-time control of the imaging device or the object being imaged.

3.1.1.1 Specific motion compensation

Motion compensation methods aim to mitigate artefacts by explicitly accounting for motion within the acquisition strategy, the reconstruction process, or both. Rather than attempting to remove motion entirely, these approaches seek to reduce its adverse effects by exploiting redundancy in the data or modelling pipeline.

In some modalities, compensation is achieved during data acquisition. In particular, compensation approaches may include gating or triggering strategies, in which data are acquired only during selected phases of a physiological cycle [63, 64]. In accounting for motion directly in the acquisition process, for example by acquiring data at specific points in the cardiac cycle [18], it is possible to discretise motion into a small number of quasi-static states.

Other motion compensation techniques make use of additional measurements to estimate motion during acquisition. A prominent example in MRI is the use of navigator echoes [101], in which supplementary sequences are interleaved with the primary acquisition to probe motion-induced phase variations. These measurements can then be used to estimate motion parameters, or to weight and reorder the acquired data prior to reconstruction.

More generally, motion compensation naturally extends to reconstruction methods in which motion is incorporated directly into the forward model [133, 150]. In this setting, motion is treated as an explicit component of the data generation process, allowing image reconstruction and motion estimation to be coupled within a single modelling framework. This perspective provides a direct link to model-based iterative reconstruction methods and motivates the formulations considered later in this chapter.

3.1.2 Retrospective motion correction

Retrospective motion correction (RMC) refers to post-processing approaches in which motion is addressed after data acquisition has been completed. These approaches are often based on image registration methods, whereby transformations are estimated in order to align images that have been affected by motion [68, 112]. Abstractly, given two images, ι_1 and ι_2, registration methods seek a transformation Π such that $\iota_2(\mathbf{r}) \approx \iota_1(\Pi(\mathbf{r}))$ for all $\mathbf{r}$ in the relevant domain. Π is typically restricted to a rigid, or

a smoothly-parametrised non-rigid family. The success of such approaches generally relies on the existence of sufficient overlap between images, as well as the visibility of corresponding anatomical or structural features.

In the work presented here, these assumptions may be violated by the acquisition geometry. In particular, neighbouring emitter views may not measure the same subset of the object, leading to view-dependent visibility and partial support in the reconstructed images. As a result, the mapping from a spatial transformation of the domain to its induced effect on the measurement data is generally not unique, and registration-based motion estimates will be poorly constrained or unstable. These limitations motivate approaches in which motion is incorporated directly into the forward model, allowing consistency to be enforced at the level of the measurement set as a whole, rather than treating individual measurements in isolation.

A distinct class of retrospective approaches addresses the motion estimation problem without relying on image registration or external hardware, by extracting motion information directly from the acquired measurement data. In clinical *positron emission tomography* (PET), principal component analysis (PCA) applied to sinogram data can identify the dominant modes of temporal variation across the acquisition, which typically correspond to physiological motion signals such as respiration or cardiac rhythm [145, 146, 147]. The extracted components serve as surrogate signals for retrospective gating or phase-binning of the data, enabling motion-resolved reconstruction without any additional hardware investment. This data-driven approach is the method of clinical preference in certain commercial PET systems, and illustrates the estimation–correction distinction noted above—PCA provides a motion surrogate signal, but a separate correction step is still required to act on that estimate. Such methods rely on sufficient temporal data redundancy and periodic motion structure, and are not directly applicable to the constrained, limited-angle, non-periodic acquisition geometry considered in this work.

3.1.3 Computer vision and machine learning methods

An alternative class of approaches draws on techniques from computer vision to estimate motion directly from image data, without relying on explicit modelling of the acquisition process or on additional motion measurements. In medical imaging, such methods are commonly applied to image sequences affected by physiological motion,

including respiratory and cardiac dynamics [55], but often require large amounts of training or reference data.

Classical computer vision approaches, such as optical flow, typically estimate a dense displacement field describing the apparent motion between successive images. While effective in many settings, these techniques require the estimation of a full vector field over the image domain, leading to increased computational cost and heightened sensitivity to noise in undersampled settings.

More recently, machine learning (ML) methods have been proposed for motion correction in a range of imaging modalities [27, 91, 99, 137]. In these approaches, neural networks are trained either to estimate entire motion fields or to directly map motion-corrupted data to motion-corrected images, often bypassing the need to explicitly incorporate a motion model. Although such approaches can achieve impressive results, they typically rely on large amounts of representative training data; this is an issue for our particular problem, as the relevant datasets do not readily exist. Additionally, these methods generally require large amounts of computational resources, which is not well-suited to the computational constraints of the device considered in this work.

For these reasons, while computer vision and ML-based methods form an active area of research, they are not the focus of the present work.

3.1.4 Parametrised motion correction

In this work, motion correction is approached through low-dimensional, parametrised motion models. This choice is motivated both by the intended application and by the structure of the underlying reconstruction problem. The work presented here is an initial study into motion correction for a portable imaging device, for which acquisition geometry and computational resources are inherently constrained. In particular, the system is not assumed to have access to specialised tracking hardware, external markers, or high-performance computing resources.

We therefore treat motion correction from a modelling perspective. By restricting attention to parametrised motion models, we assume that the dominant motion effects may be described by only a small number of variables, thereby maintaining computational feasibility without substantially increasing the ill-posedness of the inverse problem. Moreover, the primary sources of motion in our work are not assumed to be completely arbitrary; in addition to possible body motion, device motion and

mechanically-induced vibrations may play a significant role. We assume that these may be adequately described by rigid-body transformations, and consequently restrict our attention to rigid-body motion throughout this work.

Within this framework, rigid models are introduced and analysed in a controlled manner, progressing from isolated translations and rotations, to full rigid-body motion in both two and three dimensions. More general motion models, including oscillatory or non-rigid motion, fall outside the scope of the present work.

3.2 A general parametric motion model

In Chapter 2 we introduced a model for X-ray attenuation and image reconstruction, such that the reconstruction problem can be represented by solving a (typically underdetermined) linear system. We now need to discuss the addition of motion into our model, to determine whether it is at all possible to correct for motion effects in a system which is already information-limited, both in the total number of measurements and in angular coverage.

We assume that during acquisition the object undergoes a continuous time-dependent motion. We model this using a Lagrangian framework, in which the motion may be described by a continuous family of maps $\{ \Psi_t \}$ such that

$$\Psi_t : \mathbb{R}^d \to \mathbb{R}^d \quad \text{for } d \in \{2, 3\}, \tag{3.1}$$

where $\Psi_t(\mathbf{r})$ denotes the location at time t of the material point that was at $\mathbf{r}$ at a chosen reference time[1] $t = 0$. For each fixed time t, the map Ψ_t is assumed to be continuous in space, and the family $\{ \Psi_t \}$ is assumed to depend continuously on time. Under these assumptions, the joint map

$$(\mathbf{r}, t) \mapsto \Psi_t(\mathbf{r}) \tag{3.2}$$

is continuous on $\mathbb{R}^d \times \mathbb{R}_{\geq 0}$. No linearity or differentiability assumptions are imposed, and additional structure is introduced only when specialising to particular motion models. Rather than attempting to reconstruct a separate image at each exposure time, we view the attenuation coefficient as a time-dependent field $\mu(\mathbf{r}, t)$ generated by transporting a single reference field $\mu_0(\mathbf{r}) := \mu(\mathbf{r}, 0)$ under the map Ψ_t. In particular, we write

$$\mu(\mathbf{r}, t) = \mu_0(\Psi_t^{-1}(\mathbf{r})), \tag{3.3}$$

[1] Generally, the time $t = 0$ will be taken to be the start of the acquisition process.

so that the observed time dependence arises purely from motion of the object, rather than from changes in its intrinsic properties.

In later sections we will restrict Ψ_t to represent only rigid-body motions, allowing the motion to be parametrised by a low-dimensional vector u lying in an admissible motion space $\mathcal{U}$.

3.2.1 Timescale assumptions

In order to model motion during the acquisition process without introducing additional ill-posedness, we impose two assumptions relating the characteristic timescales of motion and measurement.

B1. *Motion is negligible during a single exposure.*
Let Δt_E denote the time taken for an emitter to fire. We assume that the displacement of the object during this interval is small compared to the spatial resolution of the reconstruction grid. Consequently, each measurement can be associated with a single acquisition time t_κ, and no motion-induced blurring occurs within an individual acquisition; that is, from the perspective of the firing emitter, the body appears momentarily stationary.

B2. *Motion is not faster than emitter sequencing.*
Let Δt_S denote the time between successive emitter firings. We assume that the motion evolves on a timescale comparable to, or slower than Δt_S, so that the sequence of measurements samples the motion trajectory with sufficient temporal density. In particular, the motion is not permitted to undergo rapid changes between firings that would be undetectable during the acquisition process. This may be interpreted as assuming that the temporal motion signal is effectively band-limited relative to the sampling rate[2].

Under these assumptions, the acquisition process produces a sequence of projection measurements indexed by times[3] $\{ t_\kappa \}_{\kappa=1}^{K}$, each corresponding to a snapshot of the object under the identified motion map Ψ_{t_κ}.

[2]This interpretation is only meaningful for motion models with a frequency component, i.e. rotation and full rigid-body motion.

[3]As each emitter fires exactly once during acquisition, the number of acquisition times equals the number of emitters, i.e. $K = n_E$.

To clarify the role of these assumptions, we note that if motion were significant during a single exposure, the resulting data would exhibit irreversible motion blur, rendering accurate reconstruction impossible regardless of post-processing. Conversely, if the motion evolved on a timescale much faster than the emitter sequencing, distinct motion states would not be adequately sampled, again preventing reliable correction. The assumptions above isolate an intermediate regime in which motion is present in the data, but remains accessible to modelling and correction.

3.2.2 A reference for motion

When an object moves during acquisition, the detectors collectively obtain the total measurement vector $\widehat{\mathbf{b}}$. This differs from the vector $\mathbf{b}$ that would be obtained from a stationary object, since each measurement corresponds to a different state of the object during the acquisition process.

A naïve approach to reconstruction would be to ignore this distinction and attempt to solve a stationary inverse problem of the form

$$\mathbf{x}^* \in \underset{\mathbf{x}\in\mathbb{R}^n}{\arg\min} \left\{ \lambda\|\mathbf{D}\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{A}\mathbf{x} - \widehat{\mathbf{b}}\|_2^2 \right\} \quad \text{subject to} \quad \mathbf{x} \geq \mathbf{0}, \tag{3.4}$$

where $\mathbf{A}$ is the system matrix corresponding to a stationary object.

This formulation is incorrect, however, as the assumption of a single forward operator $\mathbf{A}$ acting on a fixed object is violated when motion is present. Under Assumption B1, all rays associated with a given emitter firing may be treated as occurring at a single time, but measurements from different emitters correspond to distinct motion states of the object. Therefore, the measured data $\widehat{\mathbf{b}}$ cannot, in general, be explained by a single stationary system matrix acting on a single image.

If the object moves between successive emitter flashes, then each emitter samples the object at a distinct state of motion. If discretised on the reconstruction grid, this corresponds to a family of objects $\{\,\widehat{\mathbf{x}}_\kappa\,\}_{\kappa=1}^K$, where each $\widehat{\mathbf{x}}_\kappa$ represents a discretisation of the attenuation field $\mu(\mathbf{r}, t_\kappa)$. These images are not treated as independent unknowns; rather, they are related through the underlying motion map and arise from a single reference object.

In particular, if $\widehat{\mathbf{x}}_0$ denotes the discretisation of a reference density $\mu_0(\mathbf{r})$, then each $\widehat{\mathbf{x}}_\kappa$ may be approximated as the result of applying a motion-induced transformation to $\widehat{\mathbf{x}}_0$. The relationship between the reference image $\widehat{\mathbf{x}}_0$ and the motion-dependent image $\widehat{\mathbf{x}}_\kappa$ is determined by the underlying motion map at time t_κ. At the continuous level,

this corresponds to transporting the reference density $\mu_0(\mathbf{r})$ under the motion map Ψ_{t_κ} to obtain $\mu(\mathbf{r}, t_\kappa)$. Upon discretisation, this induces a transformation mapping between $\widehat{\mathbf{x}}_0$ and $\widehat{\mathbf{x}}_\kappa$.

We do not assume that this transformation is linear at the discrete level. In general, the discretised images $\widehat{\mathbf{x}}_\kappa$ depend on the motion map, the choice of discretisation, and any interpolation or resampling required to represent the transported density on a fixed grid. While the motion acts continuously on attenuation fields, its discretised counterpart is required only to be a well-defined mapping from the reference image $\widehat{\mathbf{x}}_0$ to the image $\widehat{\mathbf{x}}_\kappa$ which is observed at time t_κ.

We associate a forward operator $\mathbf{A}_\kappa$ to each time t_κ, mapping the reference image $\widehat{\mathbf{x}}_0$ to the corresponding block of measurements $\widehat{\mathbf{b}}_\kappa$. The dependence of $\mathbf{A}_\kappa$ on κ therefore arises from the motion-induced transformation relating $\widehat{\mathbf{x}}_0$ to $\widehat{\mathbf{x}}_\kappa$.

With this notation, the full measurement vector $\widehat{\mathbf{b}}$ may be written as the concatenation of the individual measurement blocks,

$$\widehat{\mathbf{b}} = \begin{bmatrix} \widehat{\mathbf{b}}_1 \\ \widehat{\mathbf{b}}_2 \\ \vdots \\ \widehat{\mathbf{b}}_K \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_K \end{bmatrix} \widehat{\mathbf{x}}_0 \; + \; \boldsymbol{\eta}. \tag{3.5}$$

Here $\boldsymbol{\eta}$ describes not only measurement noise but also modelling and discretisation error arising from approximating the continuous, time-dependent acquisition process by a finite-dimensional, motion-dependent forward operator.

In the formulation given by (3.5), motion is incorporated entirely through the family of forward operators $\{\, \mathbf{A}_\kappa \,\}_{\kappa=1}^{K}$, while the unknown object is represented by a single reference image $\widehat{\mathbf{x}}_0$.

3.2.3 A motion-parametrised forward operator

Rather than treating the stacked system matrix in (3.5) as an arbitrary element of $\mathbb{R}^{m\times n}$, we regard it as being generated by a low-dimensional motion parameter. Let $\mathcal{U}$ denote an admissible motion space, whose elements parametrise the family of motion maps[4] $\{\, \Psi_t \,\}_{t\,\in\,[0,T]}$. For each $u \in \mathcal{U}$, the associated motion induces a family of forward operators $\{\, \mathbf{A}_\kappa(u) \,\}_{\kappa=1}^{K}$, one for each acquisition time t_κ.

[4]The maps $\{\, \Psi_t \,\}$ are now defined for all $t \in [0, T]$, such that $T \geq n_E \Delta t_E + (n_E - 1)\Delta t_S$; that is, we assume the existence of such maps for *at least* the entire acquisition process.

We therefore define a mapping

$$\Gamma : \mathcal{U} \to \mathbb{R}^{m \times n}, \qquad u \mapsto \mathbf{A}(u) = \begin{bmatrix} \mathbf{A}_1(u) \\ \mathbf{A}_2(u) \\ \vdots \\ \mathbf{A}_K(u) \end{bmatrix}, \tag{3.6}$$

where $\mathbf{A}(u)$ denotes the stacked system matrix induced by the motion parameter u.

No general regularity properties of the map Γ are assumed at this level of abstraction. In particular, Γ need not be linear, differentiable, or globally continuous, as its behaviour depends on the specific motion model, acquisition geometry, and discretisation of the reconstruction domain. Rather than attempting to characterise Γ in full generality, we instead examine its structure in specific motion models of interest.

We note that the image $\Gamma(\mathcal{U})$ is a proper, highly constrained subset of $\mathbb{R}^{m \times n}$, determined by the acquisition geometry, the timings of emitter firings, and the discretisation choice. In particular, not every matrix in $\mathbb{R}^{m \times n}$ is physically admissible[5]; only those arising from an admissible motion $u \in \mathcal{U}$ are consistent with the measurement process.

With this notation, the forward model may be written compactly as

$$\widehat{\mathbf{b}} = \mathbf{A}(\widehat{u})\, \widehat{\mathbf{x}}_0 + \boldsymbol{\eta}, \tag{3.7}$$

where $\mathbf{A}(\widehat{u})$ denotes the system induced by the true motion $\widehat{u}$, $\widehat{\mathbf{x}}_0$ denotes the reference object, and $\boldsymbol{\eta}$ accounts for measurement noise and modelling error. It is important to note that, in practice, the measurement vector $\widehat{\mathbf{b}}$ is not obtained by applying the discrete forward operator $\mathbf{A}(\widehat{u})$ to $\widehat{\mathbf{x}}_0$ exactly. The true attenuation field is a piecewise-continuous function, and any discrete representation $\widehat{\mathbf{x}}_0$ is necessarily an approximation to it at the chosen reconstruction resolution.

3.2.4 A general algebraic framework for motion correction

We briefly discuss a more abstract view of the mapping from a motion space to the family of forward operators that can arise in our model, and hence to the measurements that those operators can obtain. The purpose of this discussion is not to introduce new assumptions, but rather to clarify choices made in our work, and to discuss where non-uniqueness is unavoidable.

[5]For example, any matrix which has full support—that is, $\mathbf{A}_{ij} \neq 0$ for all i, j—is not feasible, as this corresponds to *all* rays intersecting *all* voxels.

3.2.4.1 Mapping motion to forward operators

Let $\mathcal{U}$ denote a finite-dimensional space of admissible motions, and let $u \in \mathcal{U}$ represent an admissible motion. For fixed acquisition times $\{ t_\kappa \}$ we define the set of maps $\{ \Psi_{t_\kappa} \}$ such that

$$\Gamma : \mathcal{U} \to \Gamma(\mathcal{U}) \subseteq \mathbb{R}^{m \times n}, \tag{3.8}$$

where Γ is the map which concatenates the effect of each Ψ_{t_κ} into a stacked operator presented in (3.6). Therefore, for any $u \in \mathcal{U}$ and times $\{ t_\kappa \}_{\kappa=1}^K$ we associate the set of forward operators[6] $\{ \mathbf{A}(u; t_\kappa) \}_{\kappa=1}^K$ whose entries encode voxel-ray intersection lengths for each time t_κ.

We assume throughout that the acquisition times $\{t_1, \dots, t_K\}$ are fixed and known. This assumption is essential, as if the firing times are allowed to vary then the map from motion to measurements becomes strictly weaker. For example, in a simple one-dimensional velocity model, one can trade velocity against time—a doubled velocity together with halved time intervals may produce identical measurements between the two acquisition models.

More generally, we could instead define $\mathcal{T}$ to be the set of all sequences of admissible firing times $\{ t_\kappa \}_{\kappa=1}^K$—this may be identified with the set of non-negative ordered lists $\{ t_1, \dots, t_K \}$ for which $t_1 = 0$. Therefore, defining the map[7]

$$\Upsilon : \mathcal{U} \times \mathcal{T} \to \mathbb{R}^{m \times n}, \tag{3.9}$$

$$(u, \{ t_\kappa \}) \mapsto \mathbf{A}(u; \{ t_\kappa \}), \tag{3.10}$$

destroys injectivity for even the simplest of parametrisable motions. It is therefore unnatural to treat the firing timing sequence $\{t_1, \dots, t_K\}$ as an unknown. Moreover, even treating it inexactly leads to the same intractability, and so we will always model the device as having fixed, instantaneous firing.

3.2.4.2 Mapping forward operators to measurements

Given a discrete object $\mathbf{x} \in \mathbb{R}^n$—representing an approximation to an attenuation field μ—each matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ induces a data vector $\mathbf{b} \in \mathbb{R}^m$ by applying the operator to

[6] In Section 3.2.3, the dependence on $\{ t_\kappa \}$ is suppressed in the notation—more precisely we should write $\mathbf{A}_\kappa(u; t_\kappa)$, but we implicitly treat $\{ t_\kappa \}$ as fixed data throughout.

[7] $\mathbf{A}(u; \{ t_\kappa \})$ is now the operator defined in (3.6), dependent on the particular firing times $\{ t_\kappa \}$.

the object. We therefore introduce the map

$$\Lambda : \mathbb{R}^{m\times n} \to \mathbb{R}^m, \tag{3.11}$$

$$\mathbf{A} \mapsto \mathbf{A}\mathbf{x}, \tag{3.12}$$

which is clearly not injective for any fixed $\mathbf{x}$. A point to observe is that the restriction map $\Omega = \Lambda \restriction_{\Gamma(\mathcal{U})}$ is also not necessarily injective for a general parametrised motion.

Although $\Gamma(\mathcal{U})$ is a low-dimensional, parametrised subset of $\mathbb{R}^{m\times n}$—whose dimension will typically be equal to $\dim(\mathcal{U}) \ll m$—it is, generally, not a linear subspace of $\mathbb{R}^{m\times n}$. If $\Gamma(\mathcal{U})$ were a linear subspace of $\mathbb{R}^{m\times n}$ of dimension $\dim(\mathcal{U})$, one would expect the restriction map Ω to be injective on $\Gamma(\mathcal{U})$ for generic objects $\mathbf{x}$, by a simple dimension-counting argument, since $\dim(\mathcal{U}) \ll m$.

Even if Γ were injective, the lack of linearity of the space $\Gamma(\mathcal{U})$ means that Ω need not be injective. In particular, distinct motions may produce identical measurements for an unknown object $\mathbf{x}$, perhaps due to symmetries of $\mathbf{x}$ or due to limited acquisition geometry. This means that, even with known timing sequence, it is generally not possible to deduce u directly from the data vector alone, and it is not possible to infer $\mathbf{x}$ without additional modelling assumptions. This is why we adopt a motion-operator viewpoint, searching over admissible motions $u \in \mathcal{U}$—and, hence, over admissible forward matrices $\mathbf{A}(u)$—rather than attempting to infer operator information directly from the data, which would be ill-posed even before discretisation or regularisation is considered.

3.3 A nested optimisation formulation

The forward model in (3.7) depends on both the unknown reference image $\widehat{\mathbf{x}}_0$ and the unknown motion parameter $\widehat{u} \in \mathcal{U}$. If the motion were known—that is, we knew the value $\widehat{u}$—then reconstruction would reduce to a standard inverse problem with a fixed forward operator, $\mathbf{A}(\widehat{u})$. Conversely, if the image were known, the data could perhaps be used to infer the motion, although this would be a highly ill-posed problem. In practice, neither $\widehat{u}$ nor $\widehat{\mathbf{x}}_0$ is known *a priori*.

The coupling of these two problems necessitates a joint estimation of the image and the motion together, albeit not with equal priority. We formulate this as a nested optimisation problem, in which the image is reconstructed for a given motion parameter, and the motion parameter is then chosen to best explain the observed

data, to the extent permitted by the available data. Specifically, for each $u \in \mathcal{U}$ we consider the reconstruction problem

$$\mathbf{x}^*(u) \in \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^n} \left\{ \lambda\|\mathbf{D}\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{A}(u)\mathbf{x} - \widehat{\mathbf{b}}\|_2^2 \right\} \quad \text{subject to} \quad \mathbf{x} \geq \mathbf{0}, \tag{3.13}$$

where $\mathbf{D}$ is our chosen sparsifying transform and $\lambda > 0$ is a regularisation parameter—we will refer to this as the *inner* problem. Throughout this work, we denote the optimal value of (3.13) by $\mathcal{F}(u)$, where

$$\mathcal{F}(u) := \min_{\mathbf{x}\in\mathbb{R}^n} \left\{ \lambda\|\mathbf{D}\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{A}(u)\mathbf{x} - \widehat{\mathbf{b}}\|_2^2 \right\} \quad \text{subject to} \quad \mathbf{x} \geq \mathbf{0}. \tag{3.14}$$

Using this form, the motion parameter may be estimated by minimising the following problem,

$$u^* \in \operatorname*{arg\,min}_{u\in\mathcal{U}} \mathcal{F}(u). \tag{3.15}$$

This will be referred to as the *outer* problem, which is, in general, non-convex and may admit multiple minimisers. The notion of using a reconstruction $\mathbf{x}(u)$ to update a candidate choice for u is theoretically appealing but practically intractable. The inversion of the argmin in (3.13) is ill-posed, as it requires the identification of mn variables—the entries of $\mathbf{A}(u)$ for an updated parameter u—from only the n entries of $\mathbf{x}$. Nevertheless, under mild assumptions the outer problem (3.15) is well-posed in the sense that a minimiser u^* exists. In particular, if $\mathcal{U}$ is compact and $\mathcal{F}(u)$ is lower-semicontinuous on $\mathcal{U}$, then (3.15) admits at least one solution u^*, and a corresponding optimal reconstruction $\mathbf{x}^*$ is then obtained by solving the inner problem at u^*.

We do not, in general, expect uniqueness of either the motion parameter or the reconstructed image. Rather than attempting to analyse existence and uniqueness in full generality, we focus in subsequent sections on specific motion models, where additional geometric structure permits explicit analysis and computation.

Chapter 4

Experimental and modelling considerations

Contents

4.1 Simulated phantoms 63
4.2 Choice of solver 65
4.3 Sub-domain discretisation 72
4.4 Device and geometry parameters 74

Before presenting experimental results for our motion-correction approach, it is necessary to address some practical modelling considerations that underpin the experiments in subsequent chapters. These ideas do not form part of the theoretical development of the forward model or the optimisation framework, but they play a central role in ensuring that the numerical experiments are well-defined and easily reproducible.

Firstly, we introduce the synthetic test objects—from here onwards, referred to as *phantoms*—used for our reconstructions to provide ground-truth objects for evaluating reconstruction quality under varying acquisition and motion scenarios. Secondly, we discuss the choice of numerical solver used to address the inner reconstruction problem, with an emphasis on suitability and computational stability rather than algorithmic detail. The focus of this thesis is not on *how* to solve the problem presented in (3.13)–(3.15), but rather on determining whether it can be approached with existing optimisation algorithms. Finally, we describe an additional constraint to our presented model, in the form of a reconstruction sub-domain. By focusing on a sub-domain which lies entirely within the scan domain, object motion may be modelled consistently without introducing non-physical boundary effects.

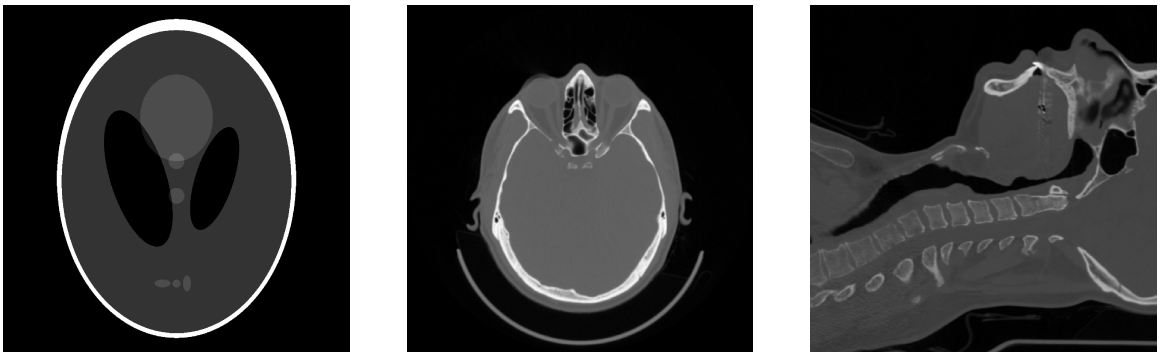
These elements lie at the intersection of modelling and experimentation. They are not specific to any single motion model, nor are they intrinsic to the underlying physics of data acquisition, and for this reason they are presented separately from the core modelling chapters. Together, they establish a common experimental framework that will be used consistently throughout the numerical experiments that follow.

4.1 Simulated phantoms

Throughout this thesis, we make use of the Shepp–Logan phantom [130, 131] in both two and three dimensions. This phantom consists of a superposition of ellipsoidal regions with piecewise-constant attenuation values and has become a standard benchmark in tomographic imaging. We also use more realistic phantoms, in the form of sampled CT datasets, so that the applicability of our approach can be tested on more realistic data. Visualisations of the phantoms used in the numerical experiments are provided alongside the corresponding results—see Figures 4.1 and 4.3—and will be referenced where relevant.

4.1.1 Two-dimensional phantoms

The two-dimensional Shepp–Logan phantom is defined by 10 ellipses of varying sizes, orientations, and grey levels (representing different attenuation coefficients), and is often used as the standard model of a human head for the purposes of testing image reconstruction algorithms—see Figure 4.1a.



(a) ‘SLP-2D’ (b) ‘HEAD-2D’ (c) ‘NECK-2D’

Figure 4.1: (‘SLP-2D’) The Shepp–Logan phantom, as built by the `phantom()` function in MATLAB. (‘HEAD-2D’) The median axial slice of a human CT dataset of the head and neck; Dataset tag: `CQ500CT172`. (‘NECK-2D’) The median coronal slice of a human CT dataset of the head and neck; Dataset tag: `CQ500CT172`.

As a comparison, we also use two data-derived digital phantoms obtained from the CT CQ500 Database [28, 122], a publicly available collection of non-contrast head CT scans acquired under routine clinical protocols. The CQ500 Database contains a wide range of three-dimensional CT reconstructions, and the two-dimensional images used as phantoms—see Figures 4.1b and 4.1c—are taken as slices from the `CQ500CT172` dataset[1]. Although not designed as a controlled physical phantom, the CQ500 dataset provides anatomically faithful attenuation structure, including high-contrast cranial bone, and is well-suited for evaluating reconstruction behaviour. Therefore, the use of a phantom with large areas of homogeneity alongside more realistic phantoms with finer structure will hopefully give indications of when our motion correction approaches work well.

4.1.2 Three-dimensional phantom

For the three-dimensional reconstruction problem we will use only one phantom; the three-dimensional Shepp–Logan phantom [54, 89]—see Figure 4.2—implemented via `phantom3()`, available on The MathWorks Community File Exchange [126].

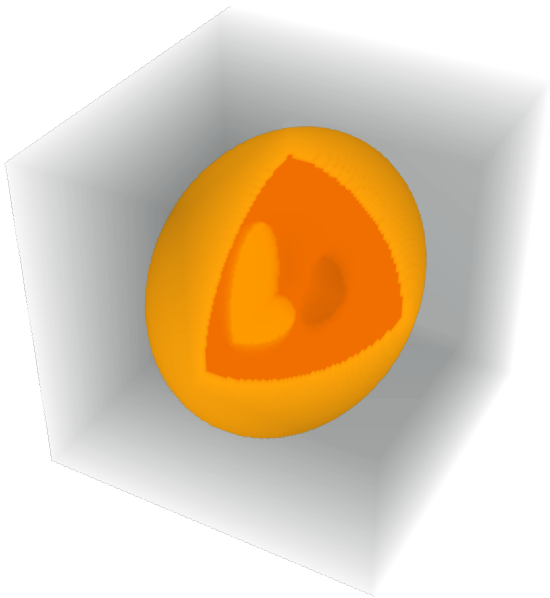


Figure 4.2: ('SLP-3D') The Shepp–Logan phantom, as built by the `phantom3()` function in MATLAB, with one octant removed to (partially) show internal structure.

[1]Specifically, the two phantoms are taken from the 'CT Thin Details' sub-dataset, with Figure 4.1b being the median axial slice, and Figure 4.1c being the median coronal slice.

The structure of this phantom is very similar to the two-dimensional case, in that it is composed of homogeneous ellipsoids of different sizes, orientations, and grey levels.

As this work is primarily focused on validating our motion-induced forward model, through the construction of $\mathbf{A}(u)$, our purpose is not to benchmark anatomical realism across multiple bodies. We therefore prioritise a phantom that is stable under resampling, can be constructed at arbitrary resolutions, and remains representative of the attenuation contrast structures typical of the imaging scenarios considered.



(a) Median axial slice. 'AX-3D'

(b) Median sagittal slice. 'SAG-3D'

(c) Median coronal slice. 'COR-3D'

Figure 4.3: Median slices of SLP-3D, through all three axis-aligned planes. These planes are specifically chosen to best represent the internal structure along all three planar directions.

4.2 Choice of solver

The reconstruction problems considered throughout this thesis give rise to large-scale, convex optimisation problems of the form introduced in (3.13). Problems of this type are well studied in imaging literature, and a range of algorithms already exist for solving them. The focus of this work lies in modelling and motion correction, rather than in the development of new optimisation algorithms, and so we restrict attention to existing solver frameworks and treat them as black-box inner routines.

In all numerical experiments presented in this thesis, the reconstruction problem is solved using the TVAL3 algorithm [96, 97, 98]; an algorithm which uses operator splitting to efficiently handle the non-smooth TV term in (3.13). Specifically, TVAL3 implements Alternating Direction Method of Multipliers (ADMM)-type updates, with alternating steps for enforcing data fidelity and promoting piecewise-constant structure via the TV penalty. TVAL3 is widely used in tomographic imaging applications, and

both its scalability and its compatibility with non-negativity constraints make it well suited to our problem. Recall that total variation regularisation enforces sparsity on the discrete gradient of the reconstructed image rather than on the image values themselves. For a discrete signal $\mathbf{x}$, the total variation is denoted by

$$\|\mathbf{x}\|_{\mathrm{TV}} := \|\mathbf{D}\mathbf{x}\|_1, \tag{4.1}$$

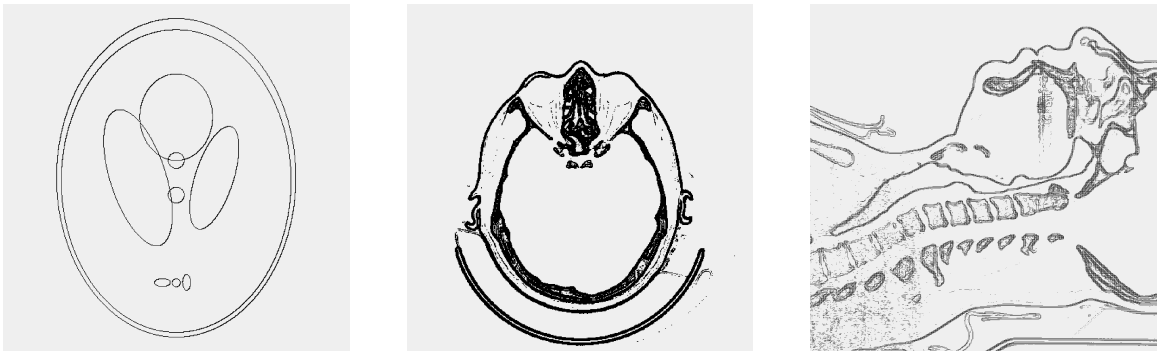
where $\mathbf{D}$ denotes the discrete derivative operator. In two-dimensional problems[2], this is

$$\|\mathbf{x}\|_{\mathrm{TV}} = \sum_{n_x,n_z} |\mathbf{x}_{n_x+1,n_z} - \mathbf{x}_{n_x,n_z}| + |\mathbf{x}_{n_x,n_z+1} - \mathbf{x}_{n_x,n_z}|, \tag{4.2}$$

and for three-dimensional problems[3] it is

$$\|\mathbf{x}\|_{\mathrm{TV}} = \sum_{n_x,n_y,n_z} |\mathbf{x}_{n_x+1,n_y,n_z} - \mathbf{x}_{n_x,n_y,n_z}| + |\mathbf{x}_{n_x,n_y+1,n_z} - \mathbf{x}_{n_x,n_y,n_z}| + |\mathbf{x}_{n_x,n_y,n_z+1} - \mathbf{x}_{n_x,n_y,n_z}|. \tag{4.3}$$

In Figure 4.4 we include representative images of the two-dimensional phantoms under the action of the discrete derivative operator.



(a) (b) (c)

Figure 4.4: The supports of all three phantoms are greatly reduced under the TV-norm: (SLP-2D) $\approx 3\%$; (HEAD-2D) $\approx 10\%$; (NECK-2D) $\approx 7\%$. All images are computed at a resolution of 1024×1024, to better sample the TV-norm of the body to be imaged, rather than approximate the expected TV-norm of a lower-resolution reconstruction. Empirically, the support ratios do not change much under resolution change, due to the nature of the discrete derivative.

In all cases, the proportion of nonzero entries in the discrete gradient $\mathbf{D}\mathbf{x}$ is less than 11% of the total number of pixels, indicating that the choice of TV regularisation

[2]Strictly speaking, this is the *anisotropic* implementation of the TV-norm—the isotropic form is $\|\mathbf{x}\|_{\mathrm{TV_{iso}}} = \sum_{n_x,n_z} \sqrt{(\mathbf{x}_{n_x+1,n_z} - \mathbf{x}_{n_x,n_z})^2 + (\mathbf{x}_{n_x,n_z+1} - \mathbf{x}_{n_x,n_z})^2}$.

[3]An analogous isotropic form exists for three-dimensional problems.

is appropriate for our particular phantoms. This choice promotes piecewise-constant reconstructions and is particularly effective for objects with sharp interfaces, as is the case for the phantoms considered in this work. In this respect, from now on we will assume that the inner problem is specifically written in the form

$$\mathbf{x}^*(u) \in \underset{\mathbf{x}\in\mathbb{R}^n}{\arg\min} \left\{ \lambda\|\mathbf{x}\|_{\mathrm{TV}} + \frac{1}{2}\|\mathbf{A}(u)\mathbf{x} - \widehat{\mathbf{b}}\|_2^2 \right\} \quad \text{subject to} \quad \mathbf{x} \geq \mathbf{0}, \tag{4.4}$$

and the outer problem is equivalently

$$u^* \in \underset{u\in\mathcal{U}}{\arg\min} \left\{ \min_{\mathbf{x}\in\mathbb{R}^n} \left\{ \lambda\|\mathbf{x}\|_{\mathrm{TV}} + \frac{1}{2}\|\mathbf{A}(u)\mathbf{x} - \widehat{\mathbf{b}}\|_2^2 \right\} \quad \text{subject to} \quad \mathbf{x} \geq \mathbf{0} \right\}. \tag{4.5}$$

Although TVAL3 is originally designed for two-dimensional reconstruction problems, its underlying formulation extends naturally to higher dimensions. For the purposes of this thesis, the solver was augmented to operate on three-dimensional domains by extending the discrete derivative operators and associated proximal steps to volumetric data. No changes were made to the optimisation strategy itself, and the solver is otherwise used in its standard form.

No claim is made that TVAL3 is optimal for all problem instances considered in this work. Rather, it is employed as a representative and reliable solver that allows the effects of our motion model to be examined without conflating these factors with solver-specific behaviour.

4.2.1 Utility of TVAL3

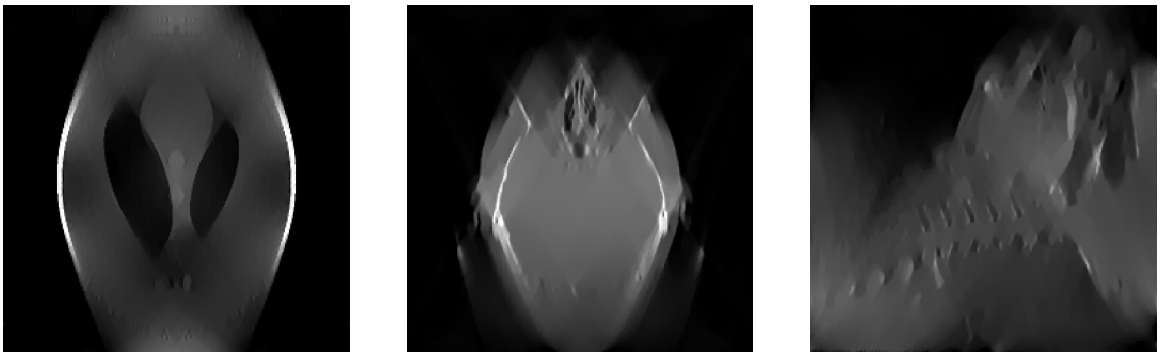
The effectiveness of TV regularisation depends critically on the choice of the regularisation parameter λ in (4.4), which governs the trade-off between data fidelity and sparsity of the reconstructed gradient. While the optimal choice of λ may depend on factors such as noise level, sampling geometry, and particular object motion, it is desirable in practice to work within a regime where reconstruction quality is stable and predictable. To assess the suitability of TVAL3 for the reconstruction problems considered here, we perform a representative parameter sweep over λ for a stationary-body reconstruction problem.

4.2.1.1 Stationary-body reconstructions

For comparison against results discussed in Part II, we include reconstructions obtained by keeping the phantom stationary[4], for which we solve (4.4) with $\widehat{u} = 0$ being known—

[4]As $u \in \mathcal{U}$ represents a sample from a low-dimensional parameter space, we may identify $\mathcal{U}$ with a subset of $\mathbb{R}^p$, such that $p = \dim(\mathcal{U})$, and associate the stationary model with the parameter $\mathbf{0} \in \mathbb{R}^p$.

see Figure 4.5. In some sense, these reconstructions would traditionally be regarded as the idealised case for image acquisition, as motion is often seen to have negative effects on reconstruction. As we will soon show, the inclusion of motion, as long as it can be both measured and corrected for, is not always detrimental to image reconstruction.



(a) (b) (c)

Figure 4.5: $[(m/n) \approx 0.3, \lambda = 10^{-3}]$[5] Reconstructions recover some small internal structures but exhibit substantial blurring artefacts. For SLP-2D this largely manifests through out-of-support artefacts (high and low z-positions), as well as smooth gradients throughout the body, rather than disjoint homogeneity. For HEAD-2D and NECK-2D, the reconstruction error appears to be loss of high-frequency information rather than a dominance of out-of-support artefacts. However, they too clearly suffer from blurring artefacts in the z-direction.

To determine the best choice for λ—or, rather, a suitable interval $[\lambda_{\min}, \lambda_{\max}]$—we compute three error metrics against the stationary reconstructions. We denote the ground truth image to be $\mathbf{x}^{(g)}$ and the reconstruction to be $\mathbf{x}^{(r)}$, to then define the Root Mean Square Error (RMSE)

$$\mathrm{RMSE}(\mathbf{x}^{(g)}, \mathbf{x}^{(r)}) = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_k^{(g)} - \mathbf{x}_k^{(r)})^2}, \tag{4.6}$$

and Signal-to-Noise Ratio (SNR)

$$\mathrm{SNR}(\mathbf{x}^{(g)}, \mathbf{x}^{(r)}) = 20\log_{10}\left(\frac{\|\mathbf{x}^{(g)}\|_2}{\|\mathbf{x}^{(g)} - \mathbf{x}^{(r)}\|_2}\right), \tag{4.7}$$

which are classical error metrics in signal processing[6].

Given that our focus is to recover images for which we expect local areas of strong similarity—particularly for SLP-2D and SLP-3D—the Structural Similarity Index

[5] Reconstructed image captions will start with the brackets $[(m/n) \approx —, \lambda = —]$, where (m/n) denotes the level of undersampling of the image, and λ denotes the regularisation parameter.

[6] Both `rmse()` and `snr()` can be called directly within MATLAB.

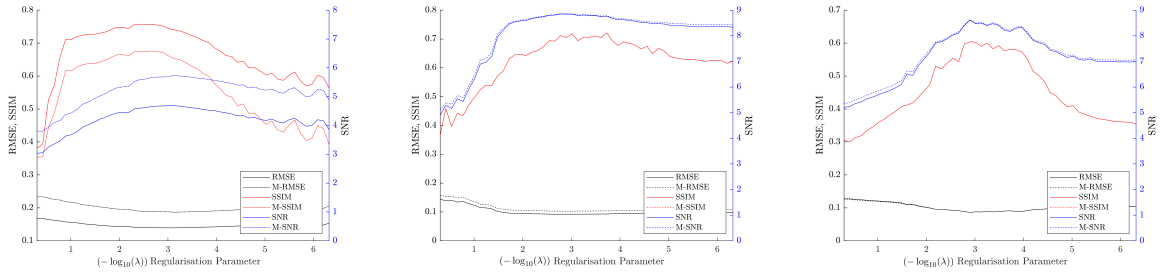
(SSIM) [149] is also an appropriate metric for reconstructions in our particular problem. By taking the image means $\{\bar{\mathbf{x}}^{(g)}, \bar{\mathbf{x}}^{(r)}\}$, variances $\{\sigma^2_{\mathbf{x}^{(g)}}, \sigma^2_{\mathbf{x}^{(r)}}\}$, and covariance $\sigma_{\mathbf{x}^{(g)}\mathbf{x}^{(r)}}$, the SSIM is defined by

$$\mathrm{SSIM}(\mathbf{x}^{(g)}, \mathbf{x}^{(r)}) = \frac{(2\bar{\mathbf{x}}^{(g)}\bar{\mathbf{x}}^{(r)} + C_1)(2\sigma_{\mathbf{x}^{(g)}\mathbf{x}^{(r)}} + C_2)}{((\bar{\mathbf{x}}^{(g)})^2 + (\bar{\mathbf{x}}^{(r)})^2 + C_1)(\sigma^2_{\mathbf{x}^{(g)}} + \sigma^2_{\mathbf{x}^{(r)}} + C_2)}, \tag{4.8}$$

for $C_1 = (K_1 L)^2$, $C_2 = (K_2 L)^2$, where $K_1 = 10^{-2}$, $K_2 = 3K_1$, and L is the dynamic range of the image[7]. We will also report masked versions, denoted by a prefix 'M-,' which will be defined by restricting each metric to the known support of the ground truth image[8].

4.2.1.2 Two-dimensional stationary phantoms

We perform a parameter sweep over λ for each of the three reference phantoms at a sampling rate of $(m/n) \approx 0.3$, finding that values around $\lambda = 10^{-3}$ give the best reconstruction quality across all three.



(a) SLP-2D (b) HEAD-2D (c) NECK-2D

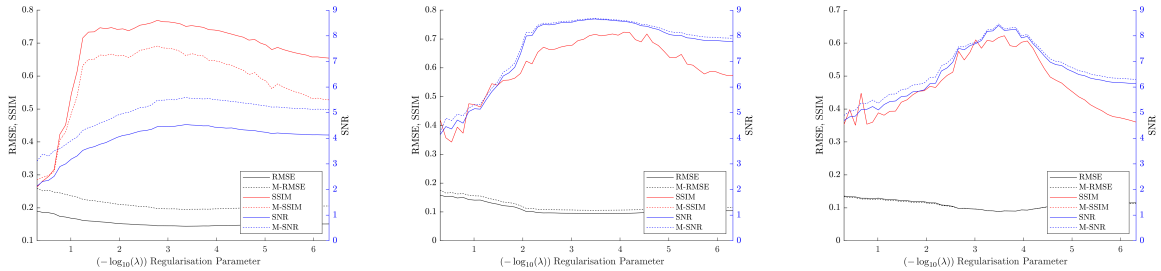
Figure 4.6: At a sampling rate of approximately 30%, all three phantoms are most accurately reconstructed for values of $\lambda \approx 10^{-3}$, as this is where the SNR and SSIM peaks are approximately located. We note that for decreasing values of λ, SLP-2D and NECK-2D have much sharper drops than HEAD-2D, possibly due to HEAD-2D having smaller support within the total image.

The results of the parameter sweep for reconstructions with approximately 30% sampling are shown in Figure 4.6. While the plots suggest that reconstruction accuracy can vary greatly for different values of λ—assuming that the error metrics are a reasonable proxy for visual clarity—we find that values of $\lambda \in [10^{-4}, 10^{-3}]$ generally

[7] `ssim()` may also be called directly in MATLAB, where L is taken to be the dynamic range of the input data type.

[8] As SSIM is an averaging measure over a fixed window, it cannot be easily redefined over a non-square support. Therefore, 'M-SSIM' is instead the region-of-interest version, ROI-SSIM, for which the SSIM is calculated over the smallest bounding box that the phantom lies within.

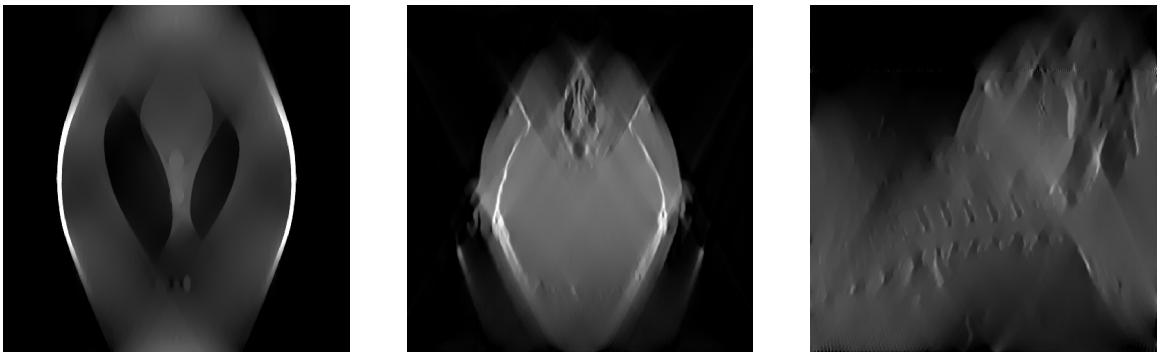
provide the best reconstructions, yet comparing them visually we find they are very similar. To avoid fine-tuning the regularisation parameter for each individual solve of (4.4), we fix the value of $\lambda_{2D} = 10^{-3}$ for all reconstructions.



(a) SLP-2D (b) HEAD-2D (c) NECK-2D

Figure 4.7: At 10% sampling, the relevant λ value for SLP-2D does not change much. For HEAD-2D and NECK-2D there is some drift into the interval $\lambda \in [10^{-4}, 10^{-3}]$, but empirical data suggests that the visual difference within this interval is almost indistinguishable.

To test the relationship between undersampling rates and regularisation, we also perform the parameter sweep for λ using a ratio of $m/n \approx 0.1$. Plots in Figure 4.7 suggest that the optimal value of λ shifts slightly, which is expected for balancing out the increased magnitude of $\|\mathbf{x}\|_{\text{TV}}$ in (4.4). However, a value of $\lambda = 10^{-3}$ provides very similar reconstructions to those obtained with a ratio of $(m/n) \approx 0.3$—see Figure 4.8.



(a) (b) (c)

Figure 4.8: $[(m/n) \approx 0.1, \lambda = \lambda_{2D}]$ Reconstructions for all three phantoms are visually similar to those with 30% sampling. Therefore, while reconstructing images at higher resolutions is still able to obtain (somewhat) recognisable reconstructions, there seems to be no benefit to inflating the total computational cost.

The authors of the TVAL3 algorithm advise that users choose regularisation pa-

rameters in the range

$$\lambda \in [1.2 \times 10^{-4}, 6.3 \times 10^{-2}]. \tag{4.9}$$

Our choice of λ falls clearly within this interval[9], indicating that the reconstruction problems arising in this work fall within the regime for which TVAL3 is known to perform reliably.

4.2.1.3 Three-dimensional stationary phantoms

We perform a similar parameter sweep of λ for stationary reconstructions of SLP-3D, to ensure that we do not mistakenly use a regularisation parameter that is suitable for two-dimensional problems but not for three-dimensional ones.

As can be seen from the results in Figure 4.9, the optimal value of λ is slightly larger than in the two-dimensional case, and so we choose the value $\lambda_{3D} = 5 \times 10^{-3}$.

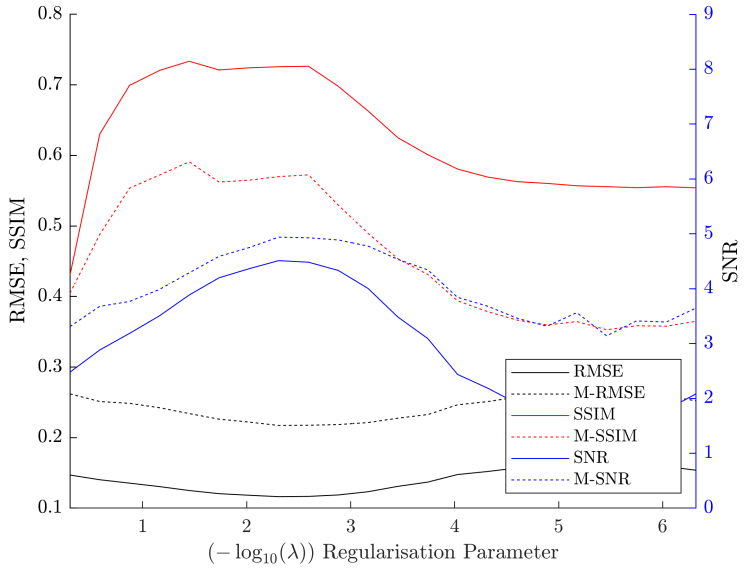
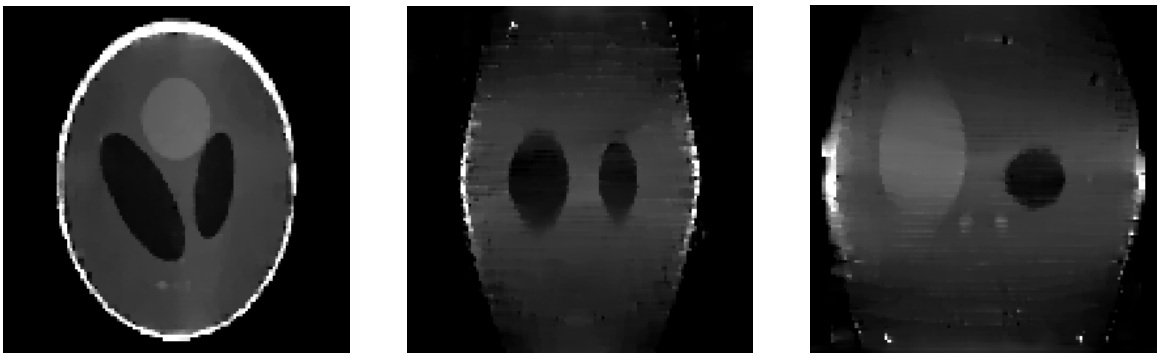


Figure 4.9: At approximately 30% sampling, the suitable values for λ appear to increase slightly for three-dimensional reconstructions of SLP-3D.

In Figure 4.10 we include reconstructions of the stationary SLP-3D phantom. As expected from the two-dimensional reconstructions, SAG-3D and COR-3D have the least clear reconstructions, due to the anisotropic effect of measurements being greatly aligned with the z-axis.

[9]TVAL3 implements the regularisation term as a multiplier μ that multiplies the ℓ_2 term in (4.4). Therefore, the advised bounds of $\mu \in [2^4, 2^{13}]$ roughly correspond to (4.9).



(a) (b) (c)

Figure 4.10: $[(m/n) \approx 0.3, \lambda = \lambda_{3D}]$ Reconstructions show a clear fidelity difference between AX-3D, versus both SAG-3D and COR-3D. However, even with AX-3D being aligned with the (x, y) plane, there are still artefacts at the boundary of the phantom. For three-dimensional reconstruction with no additional motion, we generally expect axial slices to exhibit better clarity and edge reconstruction than sagittal or coronal slices.

4.3 Sub-domain discretisation

In the forward model introduced earlier, the system matrix $\mathbf{A}$ acts on a reconstruction domain corresponding to the region enclosed by the acquisition geometry. In the absence of motion, this presents no particular difficulty. However, when object motion is introduced—even rigid-body motion—it becomes undesirable to allow the reconstruction domain itself to intersect the acquisition apparatus. Doing so will lead to non-physical configurations in which the reconstruction grid intersects the emitter or detector panels, or extends beyond the region for which measurements are defined. While this may be alleviated with hard-coded logical checks for all voxels in each time step, we expect this to inflate computational cost.

To address this, rather than reconstructing the entire scan domain, we restrict attention to a compact sub-domain contained strictly within it—see Figure 4.11. Computationally, the sub-domain is implemented in such a way that for all admissible motion parameters, the transformed sub-domain remains contained within the scan domain at all times, and therefore cannot intersect the scan domain boundary or the acquisition geometry. In reality, the cause-and-effect of this design choice is reversed, with the acquisition process governing the size of the sub-domain. If motion is detected during the acquisition process—whether by the device operator or some other means—and the object is assumed not to have moved beyond the measurement region, then

this effectively constrains the admissible motion parameter space[10]. So, while this restriction is added to guarantee that the numerical implementation is always stable, this reflects a physically reasonable assumption about practical acquisition scenarios.

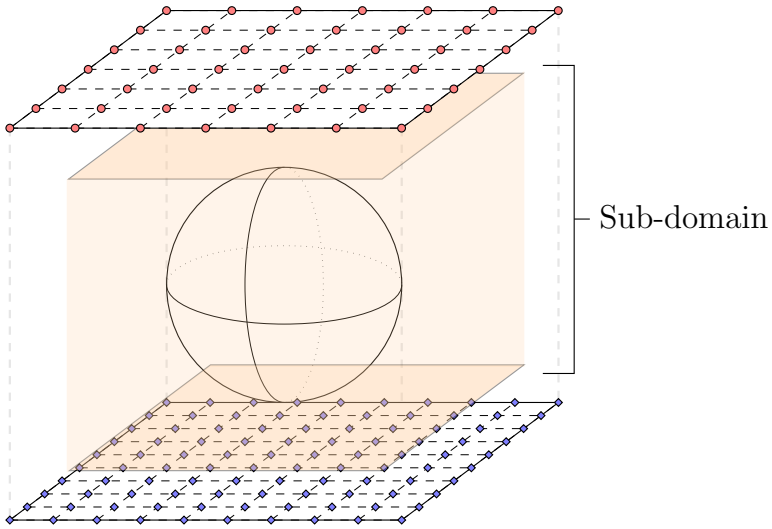


Figure 4.11: By reducing the reconstruction domain to a strict subset of the scan domain, we effectively separate the effects of motion from the device geometry itself. Therefore, the flexibility of the sub-domain allows for the modelling of moving voxels, rather than a changing attenuation field moving through a fixed grid, better serving the algebraic construction in Section 3.2.2.

Motion is then modelled as a transformation acting on the sub-domain, rather than the total scan domain, and so the model described in Section 3.2 has exactly the correct structure. In essence, the problem that we wish to solve is still (4.5), but the parametrised operators $\mathbf{A}(u)$ now act only on the sub-domain.

From a modelling perspective, the sub-domain restriction allows motion to be represented without altering the definition of the forward operator, or introducing boundary artefacts associated with truncated or invalid ray interactions. From a computational perspective, it simplifies the handling of moving objects by building constraints within the geometry itself, as only the contents of the sub-domain are

[10]For example, if the object somehow moved in-plane by 5% of the device width, but was assumed to have not ruined the acquired images, this would constrain the sub-domain to be at least 5% smaller. This, in a sense, is a modelling assumption of the following form: any spatial position $\mathbf{r}$, for which there exists a time $t \in [0, T]$ such that the map $\Psi_t(\mathbf{r})$ does not lie within the scan domain, is inherently unimportant. This does not mean that the position, nor its discrete representation, is not ever measured; rather that if this position leaves the scan domain at any point then it was not important enough to control within the acquisition setup.

discretised and reconstructed[11].

In an ideal setting, the reconstruction sub-domain would coincide with the true support of the object. In practice, this support is not always accurately known. We therefore select a sub-domain that is simple to describe and manipulate geometrically, and which is sufficiently large to contain the object throughout its motion.

In our numerical implementations, all geometries are normalised to a unit square (or cube in three dimensions), with the sub-domain taken to be a centrally aligned region scaled by a fixed factor relative to the enclosing scan domain—this scale is fixed, for most models, at a factor of 0.8 along all three axes. We take 0.8 to provide a geometric buffer against boundary artefacts without shrinking the sub-domain so much as to not be effectively measured. Experimental results suggest that reconstructions are robust to modest changes in scale—for example, a scale of 0.7 does not lead to a significant decrease in reconstruction quality.

4.4 Device and geometry parameters

Given the large number of parameters in our model, we outline those used for all two- and three-dimensional reconstructions in the following tables:

Parameter	Notation	Value
Number of emitters	n_E	100
Number of detectors	n_D	301
Collimation half-angle (°)	ϑ	30
Domain width	w	1.0
Domain height	h	1.0
Sub-domain scale	—	0.8
Sub-domain centre $(\cdot, \cdot)$	—	(0.5, 0.5)
Resolution in x-direction	N_x	256
Resolution in z-direction	N_z	256

Table 4.1: Device and geometry parameters for two-dimensional reconstruction.

[11]From a modelling perspective, we now assert that the region outside of the sub-domain has an attenuation coefficient of $\mu = 0$.

Parameter	Notation	Value
Total number of emitters	n_E	$10 \times 10 = 100$
Total number of detectors	n_D	$101 \times 101 = 10201$
Collimation half-angle (°)	ϑ	30
Domain width	w_1	1.0
Domain depth	w_2	1.0
Domain height	h	1.0
Sub-domain scale	—	0.8
Sub-domain centre $(\cdot, \cdot, \cdot)$	—	(0.5, 0.5, 0.5)
Resolution in x-axis	N_x	128
Resolution in y-axis	N_y	128
Resolution in z-axis	N_z	128

Table 4.2: Device and geometry parameters for three-dimensional reconstruction.

Option	Value	Description
`opts.mu`	$1/\lambda_{2D}$ or $1/\lambda_{3D}$	$\mu = 1/\lambda$ in TVAL3
`opts.tol`	10^{-8}	Stopping tolerance
`opts.TVL2`	`true`	TV-L2 optimisation
`opts.TVnorm`	1	Anisotropic TV-norm
`opts.isreal`	`true`	Real reconstruction
`opts.nonneg`	`true`	Non-negative reconstruction

Table 4.3: Parameters for TVAL3.

Concluding remarks for Part I

Part I has established the modelling framework used throughout the thesis, discussing the mathematical structure of tomographic operators, as well as the fundamental limitations imposed by incomplete data. Our emphasis has been on clarity of representation rather than analytic theory of reconstruction, with a focus on identifying which features are recoverable under restricted acquisition geometries.

Chapter 2 placed our image acquisition model within the broader theory of Radon and X-ray transforms, highlighting the ill-posedness inherent in limited-angle tomosynthesis. We then developed the discrete, algebraic perspective of our forward operator, and clarified why neither uniqueness nor stability can be guaranteed under realistic use case constraints.

Chapter 3 then introduced the modelling assumptions required to extend the stationary-body model, such that motion can be directly incorporated into the forward operator. Alongside this development, we broadened the classical algebraic framework into one for which a nested optimisation scheme describes the interleaved solving of motion correction and image reconstruction.

Chapter 4 discussed additional considerations which allow this problem to be tractable, and presented the numerical phantoms to be used in later experiments. We also discussed the chosen numerical solver to be used throughout this thesis, presenting reconstructions of stationary bodies under a realistic acquisition geometry.

Part II will now explore diagnostic implications of rigid-body motion within our established framework, in which motion is treated as an intrinsic component of our algebraic forward operator.

Part II

Experimental Results

Overview of Part II

The work presented in Part II is concerned with the modelling and analysis of rigid-body motion within algebraic tomographic reconstruction. Rather than developing new motion-correction algorithms for our limited datasets, our focus is instead on understanding how motion enters the forward operator itself, and how this dependence may alter the behaviour and stability of the nested optimisation problem. As such, motion will be largely treated as a design component of the forward model, rather than as an external perturbation to reconstructed images.

In Chapter 5 we restrict our modelling framework to the case of constant translation. To establish our model as a feasible proof-of-concept, we clearly outline the underlying effects of motion on the reconstruction domain, providing continuity properties for the parametric forward operator. We then include a series of reconstructions for two- and three-dimensional phantoms affected by *known* motion parameters, to determine what levels of motion, when corrected for, may actually improve image clarity. Numerical experiments suggest that the resulting reconstructions have greater clarity than those obtained from stationary bodies, although the determination of correct motion parameters is not necessarily trivial. However, we include empirical data to propose that solving the nested optimisation problem may be possible using standard optimisation techniques.

In Chapter 6 rotational motion is considered, highlighting the potential benefits of including further motion into the acquisition process. Here we also examine experimental data resulting from controlled motion, comparing reconstructions to those obtained from the stationary problem discussed in Part I. Again, we present experimental results which suggest that existing optimisation methods may be helpful in solving problems involving rotational motion. This chapter concludes with a discussion on full rigid-body motion, finding that obtained reconstructions are much clearer if the correct motion parameters are known.

A present theme throughout Chapters 5 and 6 is that even when forward operators are proven to act stably under perturbations, the induced optimisation landscapes may be highly non-smooth and poorly conditioned. Numerical experiments are therefore used diagnostically, to visualise cost surfaces and identify feasibility of correction under idealised circumstances. In Chapter 7 we present a broader methodological discussion of optimisation under model inaccuracy and resolution dependence. Here, we situate the preceding results in the context of multilevel optimisation, emphasising the limitations of gradient-based methods when objective functions are nonsmooth. Rather than proposing a specific optimisation algorithm, this discussion provides a conceptual framework for interpreting the behaviours observed in Chapters 5 and 6. We conclude Chapter 7 with a discussion on alternative bases for image representation, suggesting that there may be bases which are more suitable to solving the nested optimisation problem. This discussion establishes the conceptual framework for Part III, where the question of basis choice is addressed constructively through the development of new ray traversal algorithms for non-Cartesian discretisations.

Chapter 5

Bodies undergoing translational motion

Contents

5.1 Forward model of translational motion 80
5.2 Experimental results: fixed translation in 2D 85
5.3 Experimental results: fixed translation in 3D 92
5.4 Regularity of the inner problem 95

To introduce the problem of motion effects on image acquisition we begin by reducing our model to only two spatial dimensions, to gain insights into issues that may occur. Difficulties that arise in three-dimensional models can often be better understood through two-dimensional analysis; results are therefore presented in two dimensions, with extensions to three dimensions following analogously.

5.1 Forward model of translational motion

For a sequence of firing times $t_1, \ldots, t_K \in [0, T]$, we consider the scenario of an object moving due to a (normalised) velocity parameter[1] $u \in \mathcal{U} \subset \mathbb{R}^2$. We define u to be normalised to the firing interval length T, which allows us to equivalently interpret it as the *total translational motion* that occurs over the time interval $[0, T]$. In this way, the initial attenuation field, $\mu_0(\mathbf{r})$, moves smoothly in the direction of u such that

$$\mu(\mathbf{r}, t) = \mu_0(\mathbf{r} - ut/T) \qquad \text{for } t \in [0, T]. \tag{5.1}$$

[1]We assume that each admissible motion is uniquely identified by a parameter vector, so that the admissible motion set $\mathcal{U}$ may be identified with a subset of $\mathbb{R}^p$, where p is the total number of motion parameters.

As we expect the translation to be smooth, at each time[2] t_κ the field shifts to

$$\mu(\mathbf{r}, t_\kappa) = \mu_0(\mathbf{r} - u t_\kappa / T). \tag{5.2}$$

By virtue of the sub-domain restriction, we assume that the magnitude of u is small enough to ensure that the object does not leave the domain at any point during the image acquisition process; we therefore take $\mathcal{U}$ to be a bounded subset of $\mathbb{R}^2$. As stated in Section 4.3, we impose that the sub-domain is scaled by a factor 0.8 along all axes, such that the total translation between μ_0 and $\mu(\mathbf{r}, T)$ can be up to 10% of the total scan domain in each axial direction[3].

5.1.1 Further consideration of the motion operator

As the firing times $\{ t_\kappa \}$ are fixed, the translational motion model is simple enough that the map described in Section 3.2.3,

$$\Gamma : \mathcal{U} \to \Gamma(\mathcal{U}) \subsetneq \mathbb{R}^{m \times n}, \tag{5.3}$$

is, in general, injective[4]. This can be justified as follows.

For a fixed set of firing times $\{ t_\kappa \}_{\kappa=1}^K$, let $\boldsymbol{\pi} : \{1, \dots, K\} \to \{1, \dots, K\}$ denote the firing sequence, where emitter $e_{\boldsymbol{\pi}(\kappa)}$ fires at time t_κ. The block $\mathbf{A}_\kappa(u)$—the submatrix corresponding to measurements acquired at time t_κ—represents the forward operator obtained by instantiating the motion model Ψ_{t_κ} at that time, together with the measurement geometry associated with emitter $e_{\boldsymbol{\pi}(\kappa)}$. In general, it will not be true[5] that two distinct motions may be measured by the same sequence of emitters $\{ e_{\boldsymbol{\pi}(\kappa)} \}_{\kappa=1}^K$ to construct identical forward operators $\{ \mathbf{A}_\kappa(u) \}_{\kappa=1}^K$.

[2] Recall that, under idealised emission and sequencing, each firing time occurs at $t_\kappa = (\kappa - 1)(\Delta t_E + \Delta t_S)$, with $\Delta t_E \ll \Delta t_S$. Therefore, to good approximation, the firing times are $t_\kappa = (\kappa - 1)\Delta t_S$.

[3] This comes both from assuming that the sub-domain is centred within the scan domain, and that the sub-domain is axially aligned with the scan domain; we are free to impose both conditions. The sub-domain restriction could instead be imposed in a way whereby knowledge of an observed motion guided an alternate choice of the centre of the sub-domain, to maximise the total set of admissible parameters.

[4] We are unable to easily quantify the probability that Γ is injective, as it is tightly related to the emitter geometry as well as the particular firing sequence $\{ t_\kappa \}$. However, in realistic imaging situations, with non-negligible motion parameters and non-negligible firing times, injectivity is likely.

[5] The modelling of rigid-body motion is providing a foundation here, as the measurements of a subset of pixels enables us to know the location of all other pixels in the grid. If we instead allowed non-rigid behaviour then two distinct motions may differ in regions of the sub-domain which are not 'seen' by $e_{\boldsymbol{\pi}(\kappa)}$ in that instant, thereby constructing equivalent forward operators as the difference was not detected.

Additionally, as the acquisition of $\mu(\mathbf{r}, t_\kappa)$ reflects a measurement of the translated sub-domain, under certain conditions we can claim that the map Γ is continuous.

To be more specific in our construction, for any chosen $u \in \mathcal{U}$ the action of Γ is the assignment of a matrix of time-dependent entries, each of which represents an intersection between a ray r_{ij} and a pixel p_k, as a function of time. In particular, we extend the notation a_{ij}^k from Chapter 2 to define a *pixel-ray intersection function* $a_{ij}^k(t; u)$, for $t \in [0, T]$, which is dependent on the choice of u.

Without loss of generality, we assume that the sub-domain is aligned with the scan domain, such that all pixel edges are aligned with the x and z coordinate axes. We take a pixel p_k and ray r_{ij}, whereby the ray has an orientation with respect to the pixel. To be precise, r_{ij} has a natural angle ϕ associated with its direction, taken anticlockwise from the positive x-axis.

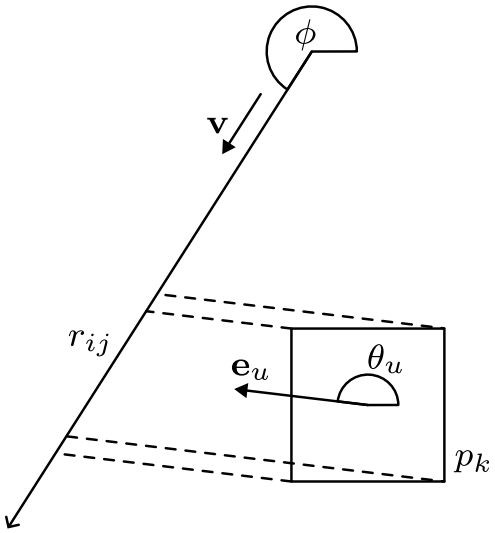


Figure 5.1: A pixel p_k moving in a direction $\mathbf{e}_u$ will intersect the ray r_{ij} if $\theta_u \in (\phi - \pi, \phi)$.

We represent the ray r_{ij} through the parametrisation

$$r_{ij} = \{\, \mathbf{p}_0 + s\mathbf{v} \mid s \in \mathbb{R} \,\}, \tag{5.4}$$

where $\mathbf{p}_0$ is a point on the ray—we may choose this to be the location of the associated emitter—and $\mathbf{v} = (\cos\phi, \sin\phi)^\top$ is the unit direction of travel. We take any admissible parameter u and define the unit vector of the same direction[6], $\mathbf{e}_u = (\cos\theta_u, \sin\theta_u)^\top$ for the appropriate angle θ_u.

[6]While we allow $u \in \mathcal{U}$ to be described generally, here we encourage the interpretation that u is a vector $\mathbf{u} \in \mathbb{R}^2$.

Under the map $k = k(n_x, n_z)$, pixel k has a known centre $\mathbf{p}_{n_x,n_z}$. We wish to analyse the behaviour of the pixel-ray intersection length as a function of time as the pixel travels in the direction $\mathbf{e}_u$, that is, as the pixel centre moves along the path $\mathbf{p}_{n_x,n_z} + t\mathbf{e}_u$ for $t \geq 0$.

We note that if $\theta_u \notin (\phi - \pi, \phi)$ and $a_{ij}^k(0; u) = 0$, then $a_{ij}^k(t; u) = 0$ for all $t \geq 0$; if the pixel and ray do not intersect and the pixel does not move towards the ray, then they will never intersect. Therefore, we need only look at motions u for which $\theta_u \in (\phi - \pi, \phi)$.

Label the pixel vertices[7] by $\{\, \nu_l \,\}_{l=1}^4$ and measure the orthogonal distances of each to r_{ij}, denoting them by $\{\, d_l \,\}_{l=1}^4$. Assuming none of the pixel faces are parallel to r_{ij}, this guarantees the existence of unique extremal distances $d_{\min}$, $d_{\max}$—the minimum and maximum values of $\{\, d_l \,\}$—such that the distances are ordered by $\{d_{\min}, \delta_1, \delta_2, d_{\max}\}$. Moving the pixel p_k along the path $t\mathbf{e}_u$ finds that the first intersection occurs when $t = d_{\min}/\cos(\theta_u - \phi + \pi/2)$. Therefore, by rescaling t to $\tilde{t} = t(\cos(\theta_u - \phi + \pi/2))$, the pixel vertices will intersect r_{ij} at times $\{d_{\min}, \delta_1, \delta_2, d_{\max}\}$, for which the intersection length is

$$a_{ij}^k(\tilde{t}; u) = \begin{cases} 0, & \text{if } \tilde{t} < d_{\min}, \\ \xi_1(\tilde{t} - d_{\min}), & \text{if } d_{\min} \leq \tilde{t} \leq \delta_1, \\ \xi_2, & \text{if } \delta_1 \leq \tilde{t} \leq \delta_2, \\ \xi_3(d_{\max} - \tilde{t}), & \text{if } \delta_2 \leq \tilde{t} \leq d_{\max}, \\ 0, & \text{if } \tilde{t} > d_{\max}, \end{cases} \tag{5.5}$$

for constants $\xi_1, \xi_2, \xi_3 \in \mathbb{R}$.

If r_{ij} is instead parallel to an edge of p_k then the function $a_{ij}^k(\tilde{t}; u)$ will feature jump discontinuities at the point of pixel-ray intersection. Therefore, if and only if r_{ij} is not parallel to a pixel edge, the behaviour of $a_{ij}^k(t)$ is piecewise linear[8] in t, regardless of the choice of u. If this condition holds for all rays r_{ij} and all pixels p_k then all entries of $\mathbf{A}(t; u)$ are piecewise-linear. Moreover, a perturbation in u will only shift the locations of the piecewise joins, and so all entries $a_{ij}^k(t; u)$ are stable.

We may therefore claim the following proposition about the map Γ.

Proposition 5.1.1 (Continuity of the translation operator map)**.** For the determination of translational motion forward operators, the map $\Gamma : \mathcal{U} \to \Gamma(\mathcal{U})$ is continuous if for all $(i, j) \in \{1, \ldots, n_E\} \times \{1, \ldots, n_D\}$ and $k \in \{1, \ldots, n\}$ ray r_{ij} is not parallel to any edge of pixel p_k.

[7] The vertices will have the form $\nu_l = \mathbf{p}_{n_x,n_z} + \frac{\Delta}{2}(\pm 1, \pm 1)$, for all four combinations of $(\pm 1, \pm 1)$.
[8] As $\tilde{t}$ is just a rescaling of t, it is clearly piecewise-linear in both.

Proof. From the discussion above, if no pixel edge is parallel to r_{ij} then all associated $a_{ij}^k(t;u)$ are piecewise-linear functions of t, for any $u \in \mathcal{U}$. If this holds for all rays r_{ij} then this equivalently means that all entries of $\mathbf{A}(u)$ are piecewise-linear functions. Therefore, for a fixed set of times $\{\, t_\kappa \,\}$, the values of $\mathbf{A}(u;\{\, t_\kappa \,\})$ are continuous with respect to u. Since $\mathbf{A}(u;\{\, t_\kappa \,\}) \in \mathbb{R}^{m\times n}$ has finitely many entries, each of which is a continuous function of u, the induced map $\Gamma : \mathcal{U} \to \Gamma(\mathcal{U})$ is continuous with respect to any matrix norm. □

In general, if the condition[9] in Proposition 5.1.1 is not met, then this does not mean that Γ is pathologically ill-behaved. Rather, if there exists a ray r_{ij} parallel to a pixel edge—and therefore parallel to *all* pixels at the same edge—then, depending on the magnitude of u, there may be rows of $\mathbf{A}(u)$ whose entries are no longer continuous as they exhibit jump discontinuities. This is a direct realisation of the notion of local 'singularities' [121] that was discussed in Section 2.1.3.

We note that the feasibility of this condition is entirely dependent on the chosen acquisition geometry. For example, if the discretised sub-domain is aligned with the emitter and detector panel, then the continuity condition is met if we ensure that no detector is collinear with any emitter in the direction of the z-axis. Within our numerical implementation, this is a very simple condition to meet[10], and so we can ensure that the induced forward operators are stable.

5.1.2 Justification for motion-operator approach

It is worth emphasising that although the underlying components of the forward model are linear when considered in isolation—the attenuation model, the motion model, and the algebraic reconstruction framework—this does not imply that the induced motion-operator map Γ is itself linear in the motion parameters. The nonlinearity arises from the geometric manner in which motion acts on the discretised domain: changes in u modify the spatial configuration of pixel-ray intersections, and hence the structure of the forward operator, in a way that is not preserved under superposition. As a result, parametrised motion models will, in general, lead to a nonlinear dependence of the operator $\mathbf{A}(u)$ on u. This distinction is essential when interpreting any regularity

[9] Under realistic motion correction scenarios—in particular, when $\mathcal{U}$ contains motions whose ray-perpendicular component exceeds a pixel width—the converse of Proposition 5.1.1 also holds, so that this condition is necessary as well as sufficient.

[10] This explains our parameter choices in Tables 4.1 and 4.2.

properties of Γ and motivates treating the motion dependence of the forward operator as an inherently nonlinear effect.

The discussion above gives further justification for approaching the problem from a forward-operator perspective, versus finding element-wise perturbations for candidate choices of a matrix $\mathbf{A}(u)$. Individual matrix entries $\mathbf{A}_{ij}(u)$ cannot be varied independently, as they are jointly constrained by the underlying geometry and motion model. This makes any attempt to directly optimise over the motion-valued map Γ—for example, by treating the $\mathbf{A}_{ij}(u)$ as decoupled variables, or by differentiating through Γ—both computationally impractical and mathematically ill-posed. Even when each entry is piecewise smooth in the motion parameters, the positions of local non-differentiability in the entries $\mathbf{A}_{ij}(u)$ depend strongly on u, leading to an objective function $\mathcal{F}(u)$ whose regularity cannot be controlled. These considerations motivate treating the function $\mathcal{F}(u)$ as a black-box within a hierarchical optimisation framework, rather than attempting to manipulate the forward operator directly through (4.5).

5.2 Experimental results: fixed translation in 2D

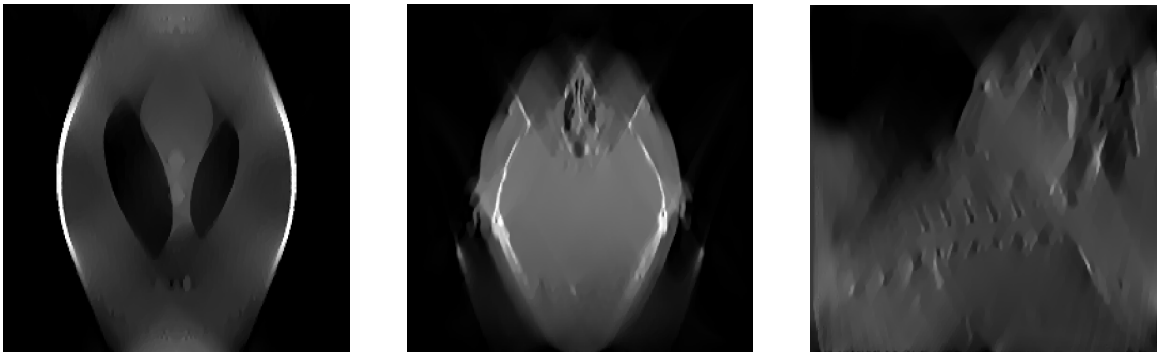
Before discussing approaches for analysing (4.5) over motions $u \in \mathcal{U}$, we first verify that the current framework is feasible, in the sense that it is possible to reconstruct a *diagnostically useful* image when the motion parameter $\widehat{u}$ is known.

As discussed in Section 3.2.4, while the map Γ is injective for this simple problem, the resulting forward operators are not injective. However, we hope to find that when the forward operator $\mathbf{A}(\widehat{u})$ is known then the solution to (4.4) is an acceptable representation of the underlying object—that is, we hope that $\mathbf{x}^*(\widehat{u})$ is a good approximation to $\widehat{\mathbf{x}}_0$. In a clinical setting, knowledge of $\widehat{u}$ is not expected without the use of additional external measurements, and knowledge of $\widehat{\mathbf{x}}_0$ is either very limited or non-existent.

The reconstructions presented here are not intended to demonstrate optimal image quality. As the acquisition geometries are severely undersampled, and the regularisation parameters are not extensively tuned for this discretisation, reconstructed images exhibit relatively low SNR and SSIM. This is expected due to the ill-posedness of the problem, and the lack of sufficiently diverse and well-sampled data measurements. Therefore, the purpose of our results is instead to validate the proposed forward model and demonstrate end-to-end feasibility.

5.2.1 Axis-aligned motion

We expect that there will be a difference in reconstruction accuracy when motion directions are aligned strongly with the z-direction—this expectation comes from the assumed difficulties in resolving pixel data in the direction most aligned with X-ray beam direction. Due to device limitations on angular range, results included here suggest that it may be easier to correct translational motion that occurs parallel to the emitter plane—this will be further validated in Section 5.3.



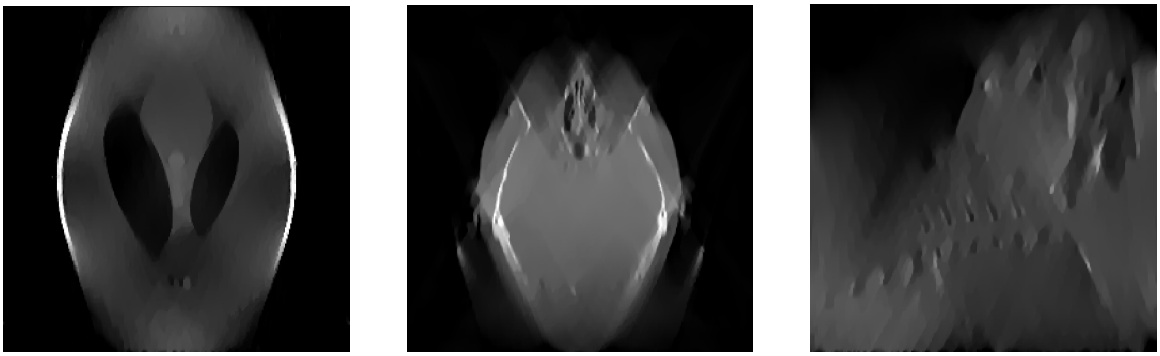
(a) (b) (c)

Figure 5.2: $[(m/n) \approx 0.3, \lambda = \lambda_{2D}]$ All phantoms are translated in the x-direction by a total distance of $x = 0.08$, such that they shift appreciably within the scan domain but they do not move outside of it. Reconstruction fidelity is almost identical to those shown in Figure 4.5, with only SLP-2D gaining a marginal improvement in visibility of the lower section of its support. HEAD-2D and NECK-2D continue to show high levels of blur artefacts.

In Figure 5.2 we show that there is little visible difference in the reconstruction accuracies of the three phantoms under small motions in the x-direction, both in terms of support reconstruction and internal details. However, empirical results for SLP-2D show slight improvements across all three metrics while the error results for both HEAD-2D and NECK-2D get slightly worse. This suggests that while the motion is exactly compensated in this model, the overall acquisition has either not improved the geometric coverage of the sub-domain, or has not led to a meaningful improvement in the conditioning of the restricted system matrix—for example via reduced column coherence—and hence provides little additional information for (4.4) to exploit.

In Figure 5.3 we present reconstructions for phantoms undergoing motion in the z-direction, moving a total distance of 0.08. As expected, the reconstructions are little improved under motion in this direction, due to the relative alignment of the measurements and the motion. We have observed that the error metrics for the

SLP-2D are only marginally better, while for HEAD-2D and NECK-2D they are again worse. Visually, the reconstructions are similar, so the error results do not directly correspond to being structurally different; in particular, as the SNR and SSIM values of stationary images are so low, the visual test is that of determining whether the reconstruction looks more like the phantom to a human observer.



(a) (b) (c)

Figure 5.3: $[(m/n) \approx 0.3, \lambda = \lambda_{2D}]$ All phantoms are translated in the z-direction by a total distance of $z = -0.08$. Similar to the results shown in Figure 5.2, clear visual improvement within the reconstructions is difficult to determine.

The minor gains made in the reconstruction of SLP-2D are likely due to the phantom consisting of a small number of homogeneous ellipses. Therefore, a small horizontal or vertical shift in a set of rays will still measure a relatively similar part of the phantom, and so we can expect the system matrix to be marginally more effective, for what remains a very similar measurement vector $\widehat{\mathbf{b}}$. However, for the realistic data of HEAD-2D and NECK-2D, any relative shift in measurement rays may easily miss smaller details—of which both phantoms have many—without making the system matrix any better-behaved. As such, the marginal gains that are made for SLP-2D are not necessarily realised on more realistic data.

For clarity, we remind the reader that these reconstructions have been obtained by selecting the correct motion parameter $\widehat{u}$ and then solving (4.4). In that sense, we expect them to somewhat represent the 'best possible' reconstruction available, compared to if we were solving the problem without knowing the motion first. An additional factor which has not been mentioned yet is the particular sequencing of the emitters. As all emitters in the device are unable to be fired at once, there is a need to sequence them individually. However, there are particular motions for which a pathologically related imaging sequence could either be helpful, or detrimental.

As a toy example, we imagine that the discrete translation $u_d = u/(K-1)$ is identical to the spacing between two neighbouring emitters; in our normalised model, this would be[11] $w/(n_E - 1)$. If the emitters were sequenced in order, from left-to-right, this would mean that the measurements from two neighbouring emitters would be identical—the body would shift between measurements by the exact amount for which the system matrix samples no new data.

While the likelihood of circumstances such as this is very slim, a way to alleviate the chance is to have the emitters fire in a random sequence. For all results obtained throughout this work, the emitters are sequenced in a randomly-sampled ordering which is then *fixed* for all experiments, i.e. no experiments are compared for which the firing sequences are different. For the actual device, there may be a particular sequencing—spiralled, diagonal passes, knight's tour—that would be most effective at covering the scan domain approximately uniformly over the time interval $[0, T]$. The optimisation of spatio-temporal emitter coverage falls outside the scope of this work.

5.2.1.1 Sub-domain relaxation

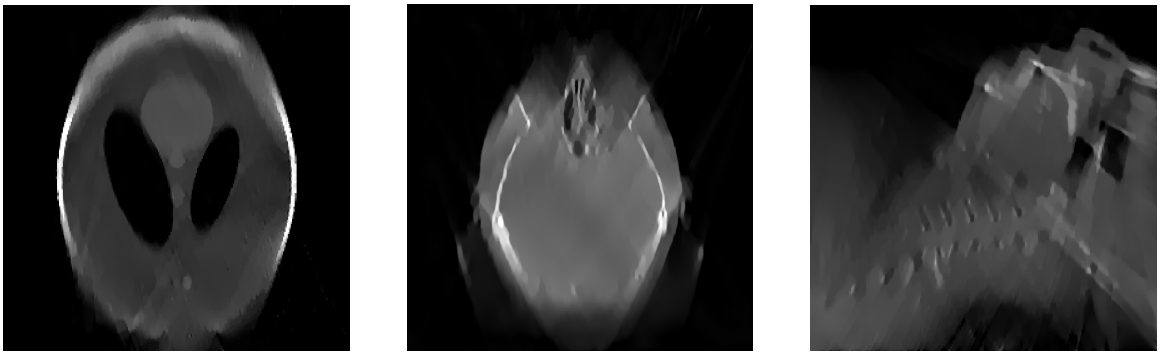
The discussion in Section 4.3 presented the restriction of a sub-domain for motion modelling, so that the phantom could be moved through the scan domain without concerns about the non-physical effects of boundary intersections. This was enforced as a modelling constraint, but the results above show that while translational motion may marginally improve the error results, the direct effect is difficult to determine from the reconstructions alone.

The sub-domain constraint is enforced primarily for motion in the z-direction, and our computational model does allow for the sub-domain to translate freely in the x-direction. We include here results from allowing the sub-domain to partially leave the scan domain—which was previously assumed to have negative effects—showing that the obtained reconstructions are visibly clearer than those discussed above.

As we present in Figure 5.4, reconstructions from models which allow the sub-domain to partially leave the scan-domain are much clearer than those obtained when the sub-domain is fully restricted. The support of SLP-2D is much more visible, and the internal curvature of the darker ellipsoids is far better represented. While the smaller

[11] In our numerical implementation, to ensure that no emitter is on the very edge of the boundary we actually pad out by two 'null' emitters that do not fire. This has little impact on the model, nor on this toy example.

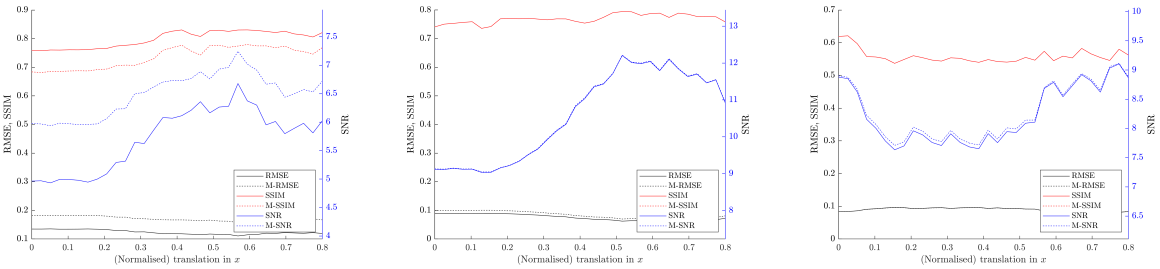
details of the phantom are not perfectly reconstructed, the overall clarity of the image is much better than originally expected.



(a) (b) (c)

Figure 5.4: $[(m/n) \approx 0.3, \lambda = \lambda_{2D}]$ All phantoms are translated in the x-direction by a (normalised) total distance of $x = 0.4$, such that by the end of acquisition approximately 30% of the sub-domain lies outside of the scan domain. While the supports of both SLP-2D and HEAD-2D are more forgiving, all three phantoms end the acquisition with part of their support outside of the scan domain.

Similarly, the reconstruction of HEAD-2D is much clearer, with both the support and internal details being more distinguishable than in Figures 5.2 and 5.3. For both CT-data phantoms, the identification of the support is greatly improved, especially for NECK-2D which previously had no sharp outlines in its reconstruction.

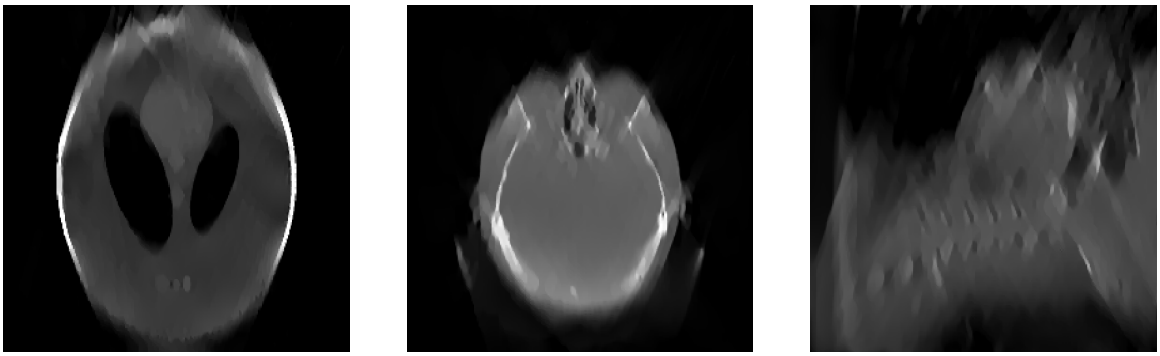


(a) SLP-2D (b) HEAD-2D (c) NECK-2D

Figure 5.5: Results for all three phantoms suggest that moving the sub-domain an appreciable distance outside of the scan domain results in more representative images, if the SNR is used as the proxy for reconstruction accuracy. However, for NECK-2D the SSIM drops by $\sim 10\%$ for $x > 0.1$, suggesting that bodies with high levels of internal detail are not improved by such motions—see Figure 5.6c. This highlights the fact that neither SNR nor SSIM are necessarily the optimal metrics for determining error of an unknown body in a limited-angle measurement setup.

The results here indicate an important point: our original restriction on the sub-domain was too tight. This will be relaxed for all further results, for which the

sub-domain is only constrained to not intersect the emitter or detector panels. Beyond this restriction the sub-domain will be free to leave the scan domain. In Figure 5.5 we include plots which identify a tolerance for exiting the scan domain for each phantom, such that image reconstruction metrics can improve for considerable motions beyond the domain boundary.



(a) (b) (c)

Figure 5.6: $[(m/n) \approx 0.3, \lambda = \lambda_{2D}]$ Reconstructions under the following normalised x-translation: SLP-2D (0.567); HEAD-2D (0.521); NECK-2D (0.781). For all three phantoms the sub-domain (partially) leaves the scan domain, suggesting that motion beyond the scan domain may improve reconstruction fidelity. Results for NECK-2D are perhaps spurious, as the 'optimal' reconstruction is (visually) less clear than those obtained in Figure 5.4.

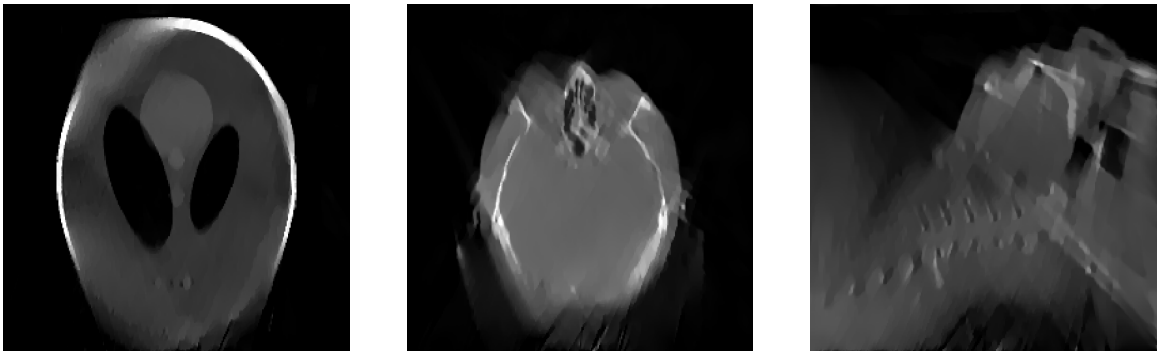
By allowing the phantom to move beyond the boundary of the scan domain, this naturally enables pixels to be measured from angles that they otherwise would not have been. Additionally, pixels will be measured in far more distinct *groupings*—by this we mean the sets of associated pixels that are all intersected by the same ray. For stationary bodies, many groupings are very similar, due to the close proximity of the neighbouring pixels, and the specific discretisation of the fan-beam rays. In particular, pixels toward the top of the domain—and therefore close to the emitters—are in groupings that will contain many of their neighbours, not merely one or two. For bodies undergoing motion, there is a much wider range of groupings, and solvers like TVAL3 are better able to identify regions where pixel similarity is locally discontinuous, e.g. at edges between sharply different attenuation values.

Allowing partial exit of the object from the reconstruction domain introduces greater variability in pixel-ray intersection patterns, reducing systematic correlations between columns of the forward operator. This increased geometric diversity improves the condi-

tioning of the inverse problem as a whole[12] and leads to more stable reconstructions—in Figure 5.6 we include reconstructions obtained for the 'optimal' allowed motion, as inferred from the data in Figure 5.5.

5.2.2 Mixed direction motion

In addition to results shown above, we find that if a phantom undergoes mixed-direction translation—with components in both x and z, while meeting the new sub-domain restriction—then reconstructions are still more accurate than those obtained from stationary experiments.



(a) (b) (c)

Figure 5.7: $[(m/n) \approx 0.3, \lambda = \lambda_{2D}]$ All phantoms are translated along the vector $(x, z) = 0.4(\cos(\pi/80), \sin(\pi/80))^\top$. All three reconstructions have clearer representations of their support, and the internal details for both SLP-2D and NECK-2D are sharper.

Looking at the results presented in Figure 5.7, it is particularly reassuring that although the z-direction is the least effective at gaining additional measurement information and negatively affected both HEAD-2D and NECK-2D, it does not seem to cause dominant effects in the case of a mixed direction model.

The results here appear to show that translational motion, when large enough and when corrected for adequately, can aid in the reconstruction process to obtain clearer reconstructions than if the body were stationary. However, for a three-dimensional model we do not necessarily expect that allowing the body to leave the scan domain is guaranteed to improve images, as it will likely result in measurements being missed from a larger number of emitters than in our two-dimensional model.

[12] In experiments, we observe that allowing for translations between $x = 0$ and $x = 0.4$ there is a reduction in the total number of valid rays—rays that intersect the sub-domain—by about 15%. However, while the amount of total information measured is reduced, in these examples the total *effective* information clearly increases.

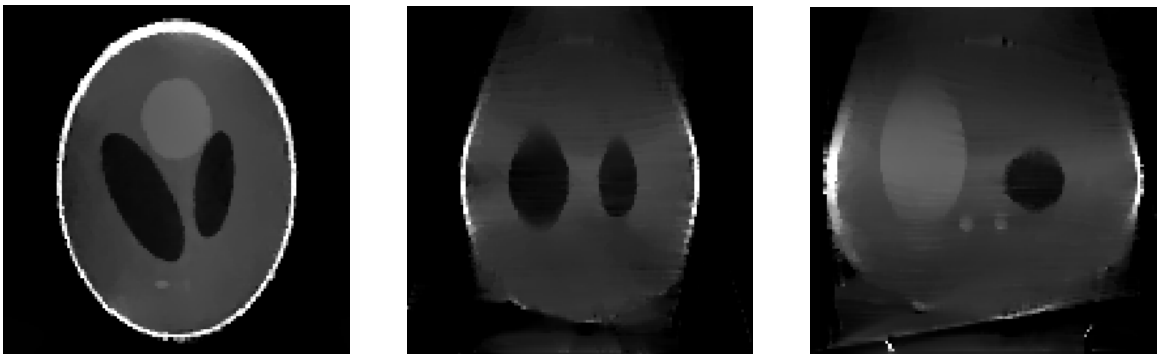
5.3 Experimental results: fixed translation in 3D

Many of the details presented in Section 5.1.1 extend to the three-dimensional model with little difficulty, so we will not discuss them in much detail here. In particular, the existence of a continuous map $\Gamma : \mathcal{U} \to \Gamma(\mathcal{U})$ still holds, only now the continuity argument rests on the requirement that the directions of rays r_{ij} are not orthogonal to any voxel face normals.

For reconstructions of SLP-3D, we restrict attention to the median slices shown in Figure 4.3, rather than the entire three-dimensional image shown in Figure 4.2. While we now allow ourselves to look at motions which may cause the sub-domain to leave the scan domain, we still restrict u to lie in a compact subset $\mathcal{U} \subset \mathbb{R}^3$.

5.3.1 Cardinal direction velocities

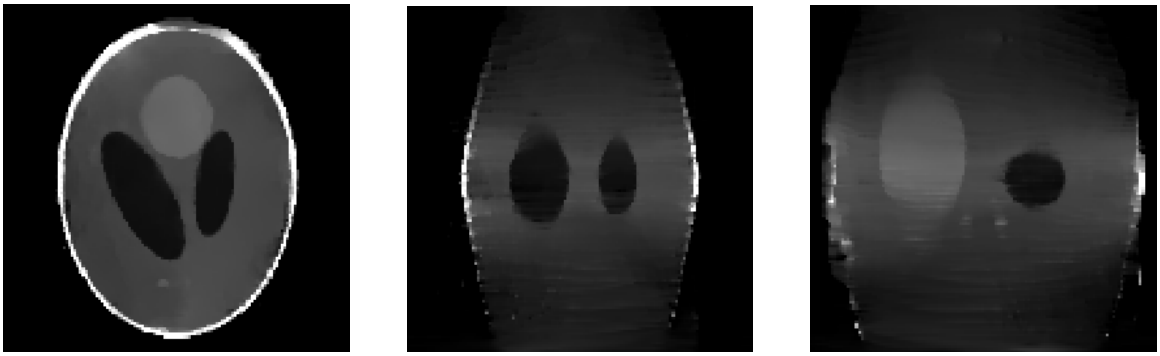
As with our two-dimensional model, we expect motion in the direction most perpendicular to the acquisition direction to be the most beneficial to improving images. Therefore, we expect reconstructions of phantoms moving parallel to the (x, y)-plane to be clearer than phantoms moving in the z-direction. Exploiting the symmetry of the geometry, we will not look at x- and y-directions separately, but will instead look at *in-plane* and *out-of-plane* motion, where the plane of reference is parallel to the emitter plane.



(a) (b) (c)

Figure 5.8: $[(m/n) \approx 0.3, \lambda = \lambda_{3D}]$ Reconstructed median slices of SLP-3D under the in-plane motion $u = 0.08(\cos(\pi/3), \sin(\pi/3), 0)^\top$. Visibility of the AX-3D axial slice is still clear, although smaller details have been lost from the stationary phantom in Figure 4.3. For both SAG-3D and COR-3D, there is better clarity at the bottom of the image, determining the edge of the support with more definition, while the top of the domain is still obscured.

In Figure 5.8 we include results of the body being translated by in-plane motion, for which the sub-domain does not leave the scan domain. Similarly, in Figure 5.9 we present reconstructions for the body being shifted solely in the z-direction by a normalised distance of 0.08.



(a) (b) (c)

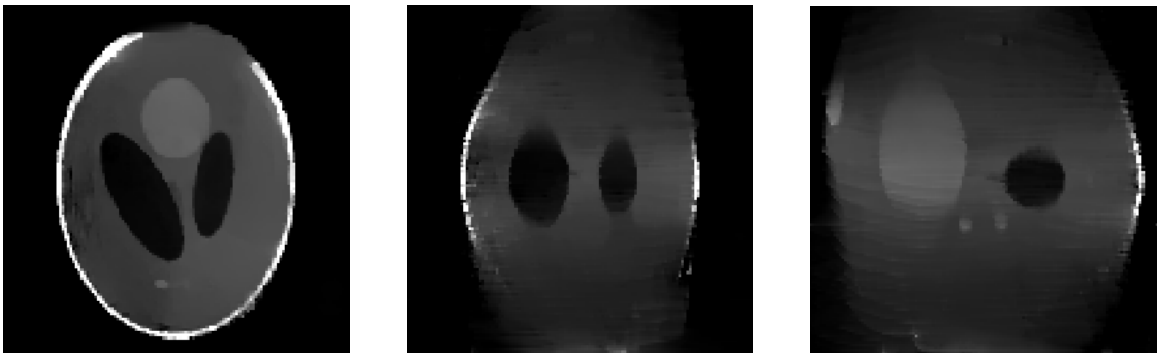
Figure 5.9: $[(m/n) \approx 0.3, \lambda = \lambda_{3D}]$ Reconstructed median slices of SLP-3D under the motion $u = (0, 0, 0.08)^\top$. Reconstruction clarity in the top and bottom of the domain is poor, which is expected from a motion model aligned with the direction of measurement acquisition.

The results show that in-plane motion can improve reconstruction, albeit anisotropically, as the benefits are primarily seen at the bottom of the reconstructed sub-domain; for out-of-plane motion the improvements are negligible, as expected.

5.3.2 Mixed direction velocities

Similar to the two-dimensional case, we find that if we choose a translation with both in-plane and out-of-plane components then the reconstructions are still reasonably clear. This suggests that any destructive effect that may arise from out-of-plane motion is weaker than the beneficial effect of the in-plane motion—see Figure 5.10.

In additional experiments we have found that when the out-of-plane motion is much larger than the in-plane motion then results degrade, as the body approaches the purely out-of-plane regime illustrated in Figure 5.9. Again, we recall that the reconstructions shown here have been obtained by solving (4.4) with a known $\widehat{u}$ to build the correct forward operator $\mathbf{A}(\widehat{u})$.



(a) (b) (c)

Figure 5.10: $[(m/n) \approx 0.3, \lambda = \lambda_{3D}]$ Reconstructed median slices of SLP-3D under the motion $u = 0.08(\cos(\pi/3)\sin(\pi/6), \sin(\pi/3)\sin(\pi/6), \cos(\pi/6))^\top$. There is marginal benefit to allowing mixed direction translations to affect the three-dimensional model. As expected, COR-3D shows that measurements may be missed entirely, which can be seen by the sharper diagonal cut-off on the left-side of the reconstruction.

5.3.3 Introduction of motion into stationary problems

While the problem presented in Chapter 3 is one concerned with *object* motion during acquisition, these preliminary results suggest a different analysis may aid reconstruction clarity. If we instead consider the reversed interpretation of the model—the body is stationary but the device itself moving—the results shown here suggest that for the device being modelled it is beneficial to include motion in the acquisition phase, as long as the motion can be accurately parametrised and measured. In effect, using a two-dimensional translation—particularly an in-plane translation—is similar to the ideas of focal plane tomography described in Chapter 2. Indeed, we can instead reframe the acquisition process as being affected by a fixed motion *from the perspective of the device* while aiming to keep the body within the total measurement domain. If the motion is too great, then the device moves away from the body—this is akin to the phantoms above moving outside the scan domain. If the motion of the device slowly zooms into the body, this is reflective of the results obtained for the out-of-plane velocities.

Therefore, an immediate takeaway of the results above is that the measurement device should, where possible, be allowed to move, with the condition that the motion itself must be well-controlled and well-measured. We note that while the device is designed to operate without mechanical motion, with emitters fired electronically in sequence, the inclusion of translational motion can clearly improve reconstruction

accuracy. Before discussing experimental results for rotational effects in Chapter 6, we briefly examine the regularity of $\mathcal{F}(u)$ through empirical evidence.

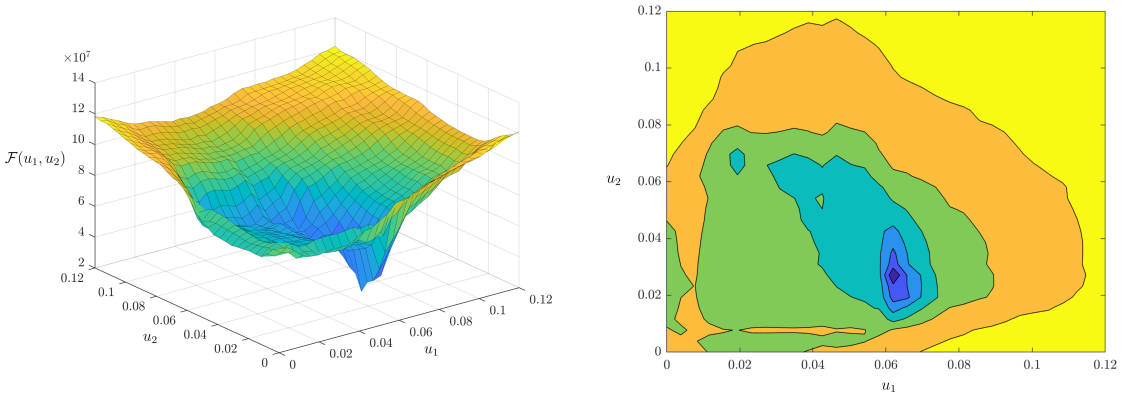
5.4 Regularity of the inner problem

In Figure 5.11a we show a representative objective function surface for the type of problem discussed above. In particular, we model the SLP-3D phantom shifted by a total translation[13] $\widehat{u} = (0.0625, 0.0325, 0)^\top$, and plot the surface of the function

$$\mathcal{F}(u; \widehat{u}) = \min_{\mathbf{x} \in \mathbb{R}^n} \left\{ \lambda \|\mathbf{x}\|_{\mathrm{TV}} + \frac{1}{2} \|\mathbf{A}(u)\mathbf{x} - \widehat{\mathbf{b}}(\widehat{u})\|_2^2 \right\} \quad \text{subject to} \quad \mathbf{x} \geq \mathbf{0}, \tag{5.6}$$

where the notation $\mathcal{F}(u; \widehat{u})$ is used here to highlight that Figure 5.11 has been made with knowledge of the true pair $(\widehat{u}, \widehat{\mathbf{x}}_0)$ such that

$$\widehat{\mathbf{b}}(\widehat{u}) = \mathbf{A}(\widehat{u})\, \widehat{\mathbf{x}}_0 + \boldsymbol{\eta}. \tag{5.7}$$



(a) Surface of $\mathcal{F}(u; \widehat{u})$, for $u = (u_1, u_2)$, and $\mathcal{U} = [0, 0.12] \times [0, 0.12]$.

(b) Contour plot of $\mathcal{F}(u; \widehat{u})$.

Figure 5.11: This surface appears sufficiently well-behaved that standard optimisation methods may be expected to identify regions of interest. However, the focus here is on understanding the behaviour of the objective function $\mathcal{F}(u; \widehat{u})$ rather than on algorithmic navigation. For clarity, (u_1, u_2) are the first two terms of $u \in \mathbb{R}^3$, as $\widehat{u}_3$ is known to be zero.

However, we do not expect that $\widehat{\mathbf{b}}$ is a result of the product $\mathbf{A}(u^*)\mathbf{x}^*$ for any minimising pair $(u^*, \mathbf{x}^*)$—instead it is the reference data around which a pair is obtained, such that (5.6) is minimal.

[13] We choose to look at in-plane motion, as we have seen that recovers better information.

The surface in Figure 5.11a is approximated by values of $u \in [0, 0.12] \times [0, 0.12]$ for which (5.6) outputs a data value. Until this point, we have merely assumed that $\mathcal{F}(u)$ will increase with worse approximations to the true parameter $\widehat{u}$, and so these plots now show this assumption to hold, at least in the cases examined here. As we show, although we are unable to make guarantees about the regularity of $\mathcal{F}(u)$, it does not appear to be pathologically irregular. However, as the output itself is obtained from solving (4.4), we cannot expect this surface to be differentiable. Similarly, as we investigate discrete representations, we can also not assume any kind of continuity.

Although we lack any regularity guarantees about $\mathcal{F}(u)$, from the behaviour of the plots in Figure 5.11 we may expect standard and simple optimisation approaches to be able to search the parameter space $\mathcal{U}$ effectively. The theoretical grounding for this expectation lies in the continuity of the map $\Gamma : u \mapsto \mathbf{A}(u)$, established in Proposition 5.1.1—this will be expanded further by Propositions 6.1.1 and 6.4.1 in Chapter 6. The practical implication of this—that $\mathcal{F}(u; \widehat{u})$ is not pathologically irregular—will then be further supported by representative surfaces for rotational and rigid-body motion. The relationship between these observations and a practical optimisation strategy is then discussed in Chapter 7.

Chapter 6

Further rigid-body motion

Contents

6.1 Forward model of rotational motion 97
6.2 Experimental results: fixed rotation in 2D 100
6.3 Experimental results: fixed rotation in 3D 102
6.4 Forward model of rigid-body motion 105
6.5 Experimental results: rigid-body motion 107

In this chapter we will look at the effects of rotation on both two- and three-dimensional bodies, before analysing reconstructions of phantoms undergoing full rigid-body motion.

6.1 Forward model of rotational motion

Similar to the approach in Chapter 5, for a sequence of firing times $\{ t_\kappa \} \subset [0, T]$, we consider the scenario of measuring an object undergoing uniform rotational motion, $\theta \in \mathcal{U} \subset \mathbb{R}$, around a centre $\mathbf{c} \in \mathbb{R}^2$. We define θ to be normalised to the firing interval length T, which allows us to interpret it as the *total angular rotation* that occurs over the interval $[0, T]$. In our discrete view of motion, we view the object as rotating by $\theta_d = \theta/(K-1)$ between successive firing times. We choose the centre of rotation to be the centre of the sub-domain, though this choice is made for experimental convenience rather than being a modelling requirement. By fixing the centre of rotation we reduce the problem to a one-dimensional model, to extract insights which carry over to the three-dimensional model.

Rotation about the centre of the scan domain is modelled by pulling back the attenuation field under a rigid rotation about $\mathbf{c}$, with the time dependence entering

only through the rotation angle. For a set of discrete rotations θ_d between emitters firing, and an initial attenuation density field $\mu_0(\mathbf{r})$, the transformed field at time t_κ will take the form

$$\mu(\mathbf{r}, t_\kappa) = \mu_0(\mathbf{c} + R(-(\kappa - 1)\theta_d)(\mathbf{r} - \mathbf{c})), \tag{6.1}$$

where $R(\zeta)$ denotes the rotation matrix for angle ζ, such that $R^{-1}(\zeta) = R(-\zeta)$.

6.1.1 Consideration of the rotation operator

In keeping with the basic analysis in Section 5.1.1, we are able to show that the motion map $\Gamma : \mathcal{U} \to \Gamma(\mathcal{U})$ inducing rotation has similar behaviour to the map inducing translation. This result is not used directly in subsequent analysis, but serves to further justify that the motion operator $\mathbf{A}(u)$ is stably defined in a connected subset of $\mathbb{R}^{m\times n}$, rather than ranging over arbitrarily discontinuous sets of matrices.

Proposition 6.1.1 (Continuity of the rotation operator map)**.** Assume that the geometry parameters in Table 4.1 are such that for any orientation of the pixel grid around $\mathbf{c}$, no ray r_{ij} is tangential[1] to the edge of any pixel p_k. Then for each (i, j, k) the map $\theta \mapsto a_{ij}^k(t; \theta)$ is continuous on $\mathcal{U}$, and hence $\Gamma : \mathcal{U} \to \Gamma(\mathcal{U})$ is continuous.

Proof. Let p_k be any pixel in the discretisation, and let θ be any total angular rotation, for which we may define[2] $\theta_t = t\theta$ for $t \in [0, 1]$. Under a rotation θ_t about the centre $\mathbf{c}$ we will denote the pixel by $p_k(\theta_t)$, and define it by the intersection of open half-spaces

$$p_k(\theta_t) = \bigcap_{\tau=1}^{4} \{\, \mathbf{r} \in \mathbb{R}^2 \mid \langle \mathbf{r}, \mathbf{n}_\tau(\theta_t)\rangle < b_\tau(\theta_t) \,\}, \tag{6.2}$$

where $\mathbf{n}_\tau(\theta_t)$ denotes the rotated pixel normals, and $b_\tau(\theta_t)$ denotes the relevant rotated half-space offsets. Both $\mathbf{n}_\tau(\theta_t)$ and $b_\tau(\theta_t)$ depend smoothly on θ_t, and hence they depend smoothly on (t, θ).

Parametrising a ray r_{ij} by $\mathbf{r}(s) = \mathbf{p}_0 + s\mathbf{v}$, the intersection length between r_{ij} and $p_k(\theta_t)$ is described by the inequalities $\langle \mathbf{p}_0 + s\mathbf{v}, \mathbf{n}_\tau(\theta_t)\rangle < b_\tau(\theta_t)$, for $\tau = 1, \ldots, 4$. Rearranging, we find

$$s\langle \mathbf{v}, \mathbf{n}_\tau(\theta_t)\rangle < b_\tau(\theta_t) - \langle \mathbf{p}_0, \mathbf{n}_\tau(\theta_t)\rangle. \tag{6.3}$$

[1] Here 'tangential' means that r_{ij} overlaps a pixel edge over a non-trivial line segment.
[2] In this context, $t \in [0, 1]$ is the rescaled acquisition time, obtained by normalising $[0, T]$ to $[0, 1]$.

Defining $\gamma_\tau(\theta_t) := \langle \mathbf{v}, \mathbf{n}_\tau(\theta_t) \rangle$ and $\delta_\tau(\theta_t) := b_\tau(\theta_t) - \langle \mathbf{p}_0, \mathbf{n}_\tau(\theta_t) \rangle$, it is clear that (6.3) has smooth functions of θ_t on either side, which are also smooth in (t, θ).

If $\gamma_\tau(\theta_t) > 0$ then we find that $s < \alpha_\tau^+(\theta_t) := \delta_\tau(\theta_t)/\gamma_\tau(\theta_t)$, else if $\gamma_\tau(\theta_t) < 0$ then instead $s > \alpha_\tau^-(\theta_t) := \delta_\tau(\theta_t)/\gamma_\tau(\theta_t)$. The entry and exit parameters of the ray are always determined by these constraints[3], and so a point $\mathbf{r}(s)$ on the ray r_{ij} lies inside the pixel if and only if

$$s > \alpha_\tau^-(\theta_t) \quad \text{for all } \tau \text{ such that } \gamma_\tau(\theta_t) < 0, \tag{6.4}$$

$$\text{and} \quad s < \alpha_\tau^+(\theta_t) \quad \text{for all } \tau \text{ such that } \gamma_\tau(\theta_t) > 0. \tag{6.5}$$

Therefore, s must lie in the interval $(s^-(\theta_t), s^+(\theta_t))$ for

$$s^-(\theta_t) = \max_{\tau\,:\,\gamma_\tau(\theta_t) < 0} \alpha_\tau^-(\theta_t), \qquad s^+(\theta_t) = \min_{\tau\,:\,\gamma_\tau(\theta_t) > 0} \alpha_\tau^+(\theta_t). \tag{6.6}$$

For all τ satisfying $\gamma_\tau(\theta_t) \neq 0$, $s^-(\theta_t)$ and $s^+(\theta_t)$ are continuous—they are the extrema of finitely many continuous functions—and so the pixel-ray intersection is simply

$$a_{ij}^k(\theta_t) = \max\{0, (s^+(\theta_t) - s^-(\theta_t))\}. \tag{6.7}$$

Then $a_{ij}^k(t;\theta) := a_{ij}^k(\theta_t)$ is continuous in θ_t, and hence in both θ and t. As such, all entries of $\mathbf{A}(t;\theta)$ are continuous in θ, and so Γ maps $\mathcal{U}$ continuously into $\Gamma(\mathcal{U})$. □

The restriction on geometric parameters is not something that we expect to control, as it depends on a combinatorial relationship between the particular rays and pixels. However, for a fixed emitter geometry and grid resolution—i.e. a finite number of rays and pixels—each intersection length $a_{ij}^k(t;\theta)$ is, at most, discontinuous in t at finitely many points, for any particular $\theta \in \mathcal{U}$. Therefore, as with the translation map, $\Gamma : \mathcal{U} \to \Gamma(\mathcal{U})$ is at least piecewise continuous. The three-dimensional analogue follows the same argument, now requiring smoothness with respect to three rotation angles $(\theta_1, \theta_2, \theta_3)$—we will not discuss this here.

By virtue of the sub-domain restriction, we assume that the total rotation, $\theta = (K-1)\theta_d$, is small enough to ensure that the sub-domain does not intersect the emitter or detector panel. For a square sub-domain of size $w_s \times w_s$, the maximal angle[4]

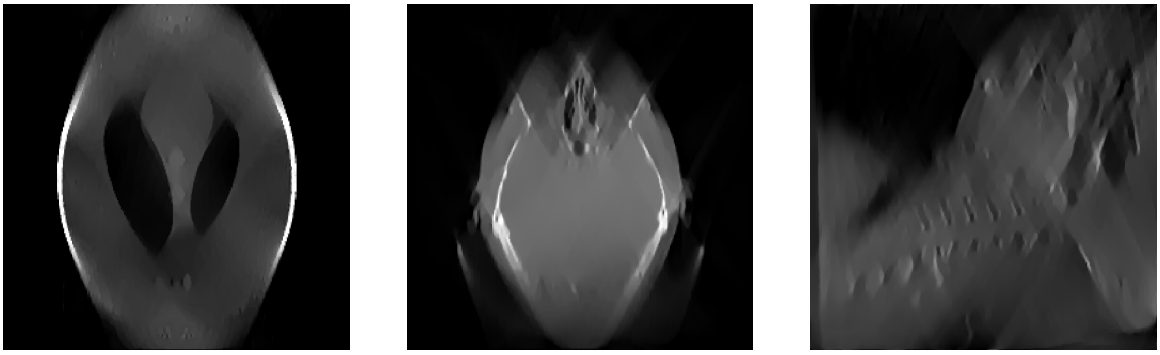
[3]Some care is required when $\gamma_\tau(\theta_t) = 0$, as (6.3) reads $0 < \delta_\tau(\theta_t)$ while the non-tangency assumption only ensures that $\delta_\tau(\theta_t) \neq 0$. The constraint no longer involves s at all, and so it is either satisfied or violated according to the sign of $\delta_\tau(\theta_t)$. In the former case, the constraint may be discarded; in the latter, the ray misses the pixel and $a_{ij}^k(t;\theta) = 0$.

[4]Note that if $w_s < 1/\sqrt{2}$, no finite maximal angle exists.

that the body can be rotated is $\theta_{\max}(w_s) = \frac{\pi}{4} - \arccos\left(\frac{1}{w_s\sqrt{2}}\right)$. For our particular model, with $w_s = 0.8$, this means that the maximal rotation is $\theta_{\max} = 0.2987$ radians, or around 17°. In light of the results in Chapter 5, we expect that even $u = \theta_{\max}$ may not produce reconstructions that are clearly distinguishable from those in Figure 4.5, and so we will include additional results for experiments with smaller sub-domains, for which rotation is unrestricted.

6.2 Experimental results: fixed rotation in 2D

In Figure 6.1 we present results for all phantoms under a rotation $u = 0.29$ radians, which is chosen to ensure that no part of the sub-domain intersects the emitter or device panel. As expected, the added angular information is not impactful enough to make the reconstructions markedly different from the stationary body case. SLP-2D has a slight reduction in out-of-support errors at the bottom of the image, but both HEAD-2D and NECK-2D have very little change.



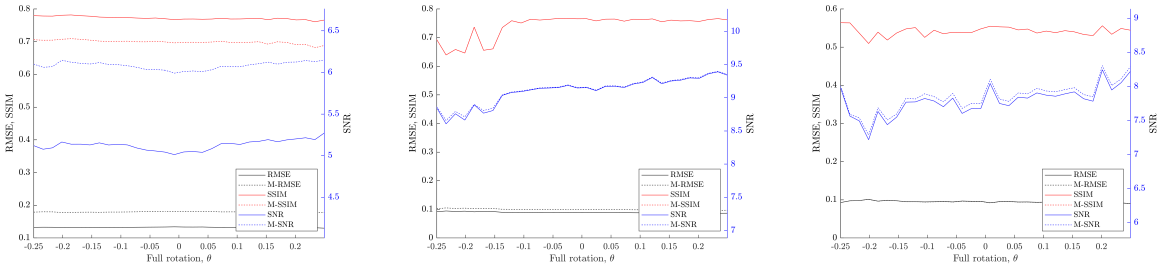
(a) (b) (c)

Figure 6.1: $[(m/n) \approx 0.3, \lambda = \lambda_{2D}]$ All phantoms are rotated around their centre by $u \approx \theta_{\max}$. In all three cases, the difference from the stationary model is difficult to discern.

In Figure 6.2 we include results that show that small rotations improve reconstruction accuracy slightly, provided the rotation is corrected for. The slight improvements in SNR and SSIM do not necessarily reflect a visibly accurate improvement, which is likely due to the low starting values of these metrics.

In Figure 6.3 we present results suggesting that we should expect improved reconstructions from allowing the sub-domain to rotate by a larger angle $u \in [-\pi, \pi]$. By similar arguments to those in Chapter 5, this idea makes sense in principle but the

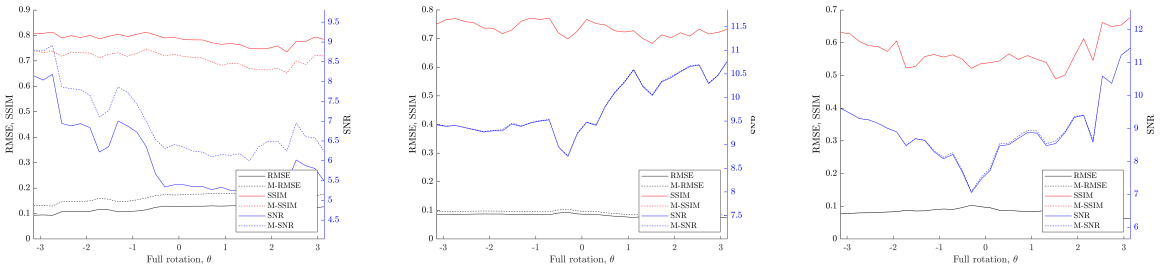
nature of a randomised emitter sequence means that there is no guarantee of more geometric coverage over all parts of the sub-domain.



(a) SLP-2D (b) HEAD-2D (c) NECK-2D

Figure 6.2: These plots suggest that for all three phantoms, the inclusion of rotation will barely improve the overall clarity of reconstructions—experimentally we observe that the effects are minor, as shown in Figure 6.1.

This can be seen in Figure 6.4, where the full rotation of the sub-domain allows smaller details to be identified more clearly—particularly for SLP-2D—yet there are clear streak artefacts from directional information not being uniformly sampled. In these experiments the sub-domain is scaled by a factor of 0.7, rather than 0.8, which removes the constraint of the maximal rotation angle.

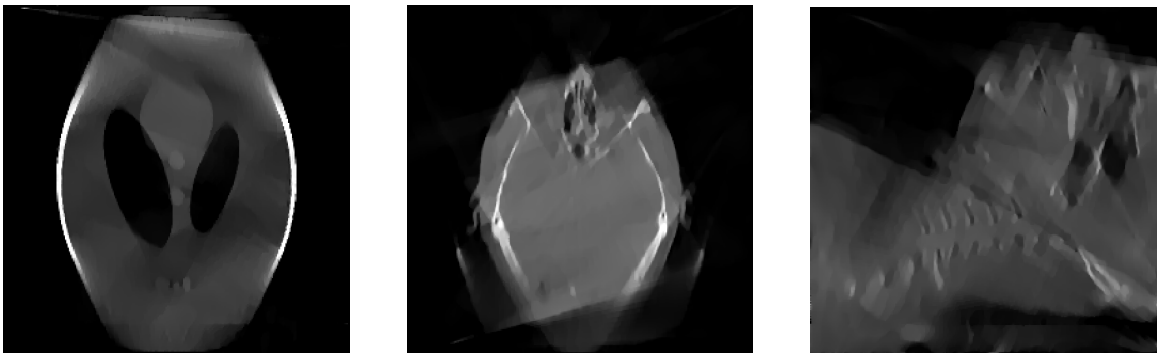


(a) SLP-2D (b) HEAD-2D (c) NECK-2D

Figure 6.3: By relaxing the rotation restriction, we expect that reconstructions should improve when allowing the sub-domain to rotate by at least a half rotation. The expected reconstruction accuracy is dependent on the direction of rotation—this is likely related to the amount of 'effective' measurements made during acquisition.

In this case, phantoms are measured from a greater set of viewing angles, although the domain is slightly smaller and so there are fewer measurements overall. As can be inferred from the three reconstructions, the particular firing sequence that the emitter panel undergoes likely has a great impact on the eventual reconstruction, as for each phantom there appear to have been viewing angles—and therefore particular times, t_κ—for which measured data density was lower than for other angles. This can be

seen especially in SLP-2D and HEAD-2D, which have sharper lines at the edge of the support, indicating a lack of information along those particular directions.



(a) (b) (c)

Figure 6.4: $[(m/n) \approx 0.2, \lambda = \lambda_{2D}]$ All phantoms are rotated around their centre by a half-rotation, $u = \pi$. While the increased angular range for these experiments improves the locality of the body support (particularly for HEAD-2D) the internal details are still not well identified. However, compared to the results of stationary bodies in Figure 4.5, these are clear improvements.

The full rotation of NECK-2D is physically unrealistic, since in practice the emitter panel would intersect the phantom's chest region. Therefore, this dataset should instead be viewed as one in which there is a high spread of information throughout the sub-domain, rather than restricted to a localised support. The results for NECK-2D do not suggest that this helped reconstruction, as the finer internal details are still not distinguishable.

6.3 Experimental results: fixed rotation in 3D

The reconstructions for the two-dimensional phantoms suggest that while rotation can improve the fidelity of the image, the resulting acquisition sequence is not an approximation to a fully-controlled rotating gantry[5]. A possible hindrance to the two-dimensional problem is that the fired emitters are not able to be directed, as with fan-beam CT, and so the total overlap of measured information is gathered in a less structured way, with a strong dependence on the particular emitter sequence.

[5]Indeed, if the motion was modelled through the device, rather than the body, this would correspond to the centre of the emitter smoothly travelling on an arc around the sub-domain, with a normal direction always facing to the centre of the body. However, due to the spatial offsets of the emitters, the fan beams of each are not guaranteed to sample the same sections of the sub-domain proportionately.

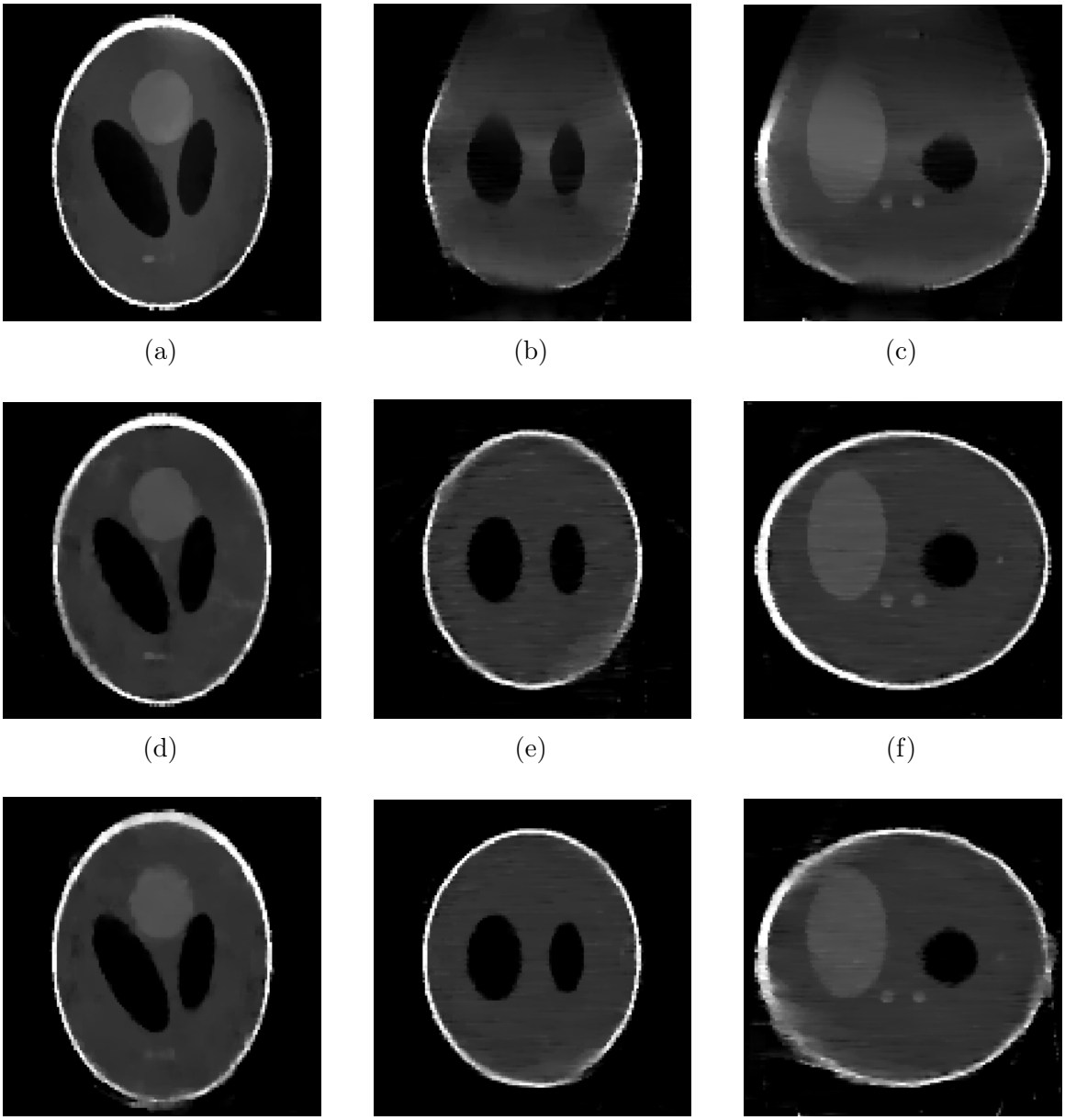
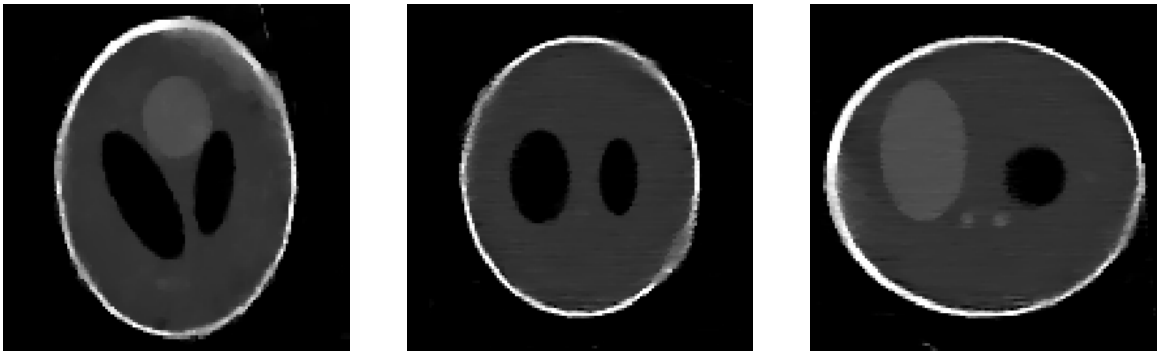


Figure 6.5: $[(m/n) \approx 0.2, \lambda = \lambda_{3D}]$ (a,b,c) Rotation by π in the axial plane. (d,e,f) Rotation by π in the sagittal plane. (g,h,i) Rotation by π in the coronal plane. These reconstructions show that for three-dimensional models the addition of rotation improves the fidelity of the images by a much greater amount than in two-dimensional models, with the clearest benefits being observed by the inclusion of sagittal and coronal planar rotation.

In comparison, results for SLP-3D show that the inclusion of rotation in either the coronal or sagittal plane can improve reconstructions drastically—see Figure 6.5—while we find that the reconstructions obtained by correcting for axial rotation are comparatively poorer. The reason for this is likely because rotation in the axial plane can be equivalently viewed as rotating the emitter panel around its own centre. In comparison, rotation in either the sagittal or coronal plane can be instead viewed as the rotation of the device *around* the body. The surprising result is that although the emitters are not fired in a specifically-chosen sequence, and therefore there is no direct control over which parts of the body are measured, the reconstructions show very few signs of the undersampled directional information that was seen in the two-dimensional reconstructions.

Extrapolating from these results, in Figure 6.6 we present reconstructions obtained by correcting for the parameter $\widehat{u} = (\pi, \pi, \pi)^\top$. Under this parametrisation the reconstruction fidelity suggests that viewing angles are sampled more uniformly as the object rotates through all three axes at once.



(a) (b) (c)

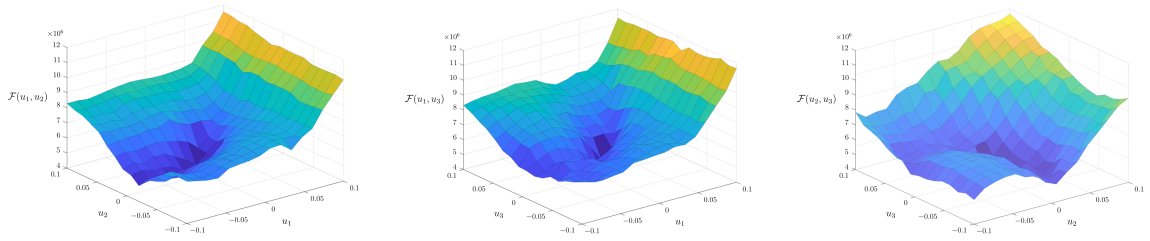
Figure 6.6: $[(m/n) \approx 0.2, \lambda = \lambda_{3D}]$ Mixed rotation model, with $\widehat{u} = (\pi, \pi, \pi)^\top$. Allowing for all three planes to have rotation, with no dominance along either axis, the results show a much clearer support, and smaller details in the coronal plane are more apparent. However, some small details in the axial plane have been lost.

6.3.1 Cost function surfaces for rotation models

These results confirm that rotation can be correctly modelled through the forward operator, $\mathbf{A}(u)$, and that solving (4.4) with the correctly sampled $\widehat{u}$ does return an image that is diagnostically helpful.

In Figure 6.7 we include representatives of the cost function $\mathcal{F}(u; \widehat{u})$ with underlying parameters $\widehat{u} = (\pi, \pi, \pi)^\top$. The three images are obtained by slicing through the

three-dimensional parameter space, centred on $\widehat{u}$, with each plane looking at only two variables: Figure 6.7a holds $\widehat{u}_3 = \pi$ fixed; Figure 6.7b holds $\widehat{u}_2 = \pi$ fixed; Figure 6.7c holds $\widehat{u}_1 = \pi$ fixed.



(a) Axial-Sagittal. (b) Axial-Coronal. (c) Sagittal-Coronal.

Figure 6.7: The three surfaces above suggest that correcting for rotational motion should be no more difficult than correcting for translational motion. In particular, these surfaces represent three subspace slices through a three-dimensional optimisation problem, i.e. solving for a completely mixed rotation model. The model-minimum is normalised here to be $(0, 0, 0)^\top$, such that the axes represent the *difference* in parameter estimates from $\widehat{u}$.

Similar to the functional behaviour shown in Figure 5.11, the surfaces in Figure 6.7 show no pathological difficulties, and give no indication that solving (4.4) is globally unstable. Therefore, it is reasonable to expect that conventional optimisation methods for nonsmooth functions are applicable to problems of this type.

6.4 Forward model of rigid-body motion

With translation and rotation both presenting results that suggest they can be accounted for—particularly if the parameters are either known or controllable—this does not preclude difficulties arising when the two effects are combined. In this section we confirm that the forward-operator associated with full rigid-body motion is still stable under perturbations—thereby establishing that our approach remains well-controlled from a modelling perspective—and we present a representative objective function slice from a six-dimensional problem.

6.4.1 Modelling rigid-body motion (2D)

We consider the problem of modelling an object undergoing rigid motion, such that the parameter space has three degrees of freedom; in particular, we shall say that $u = (v, \theta) \in \mathbb{R}^2 \times \mathbb{R}$, for which v is the parameter describing total translation and θ is

the parameter for total rotation[6]; in this sense, v acts on the *centre* of the sub-domain, and θ rotates around it. The choice of centre of rotation can be absorbed into the translation parameter, and so we may generically think of the centre of rotation $\mathbf{c}$ as being the centre of the sub-domain.

For the set of firing times $\{ t_\kappa \}$ and initial attenuation field, $\mu_0(\mathbf{r})$, the transformed field at each time will be

$$\mu(\mathbf{r}, t_\kappa) = \mu_0(\mathbf{c} + R(-(\kappa - 1)\theta_d)(\mathbf{r} - \mathbf{c} - (\kappa - 1)v_d)), \tag{6.8}$$

where R is as defined in Section 6.1.

6.4.1.1 Consideration of the rigid-body motion operator

Although rigid-body motion is composed of translational and rotational components, the map from motion parameters to the discrete forward operator is nonlinear and cannot be obtained by a superposition of the corresponding maps. However, the relationship between the parameters and the discretisation is regular enough that the result follows by extending the arguments of Propositions 5.1.1 and 6.1.1.

Proposition 6.4.1 (Continuity of the rigid-body operator map)**.** For the determination of rigid-body motion forward operators, the map $\Gamma : \mathcal{U} \to \Gamma(\mathcal{U})$ describing rigid-body motion is continuous on $\mathcal{U} \setminus \mathcal{S}$, where $\mathcal{S} \subset \mathcal{U}$ is a finite union of lower-dimensional sets corresponding to tangency events between rays r_{ij} and pixels p_k. On the entire space $\mathcal{U}$, it holds that Γ is piecewise continuous.

Proof. This result follows by adapting the geometric arguments used in the translational and rotational motion models. Let $(v, \theta) \in \mathbb{R}^2 \times \mathbb{R}$ define the total translation, and total rotational motion, such that $v_t = tv$ and $\theta_t = t\theta$ are both well-defined for all $t \in [0, 1]$. We also define the rigid transformation

$$\mathcal{T}_{v_t,\theta_t}(\mathbf{r}) := \mathbf{c} + R(\theta_t)(\mathbf{r} - \mathbf{c}) + v_t, \tag{6.9}$$

where $R(\theta_t)$ is the standard 2D rotation matrix. For a pixel p_k, define the moved pixel $p_k(v_t, \theta_t) = \mathcal{T}_{v_t,\theta_t}(p_k)$.

As in the rotational model, $p_k(v_t, \theta_t)$ is the intersection of four half-spaces

$$p_k(v_t, \theta_t) = \bigcap_{\tau=1}^{4} \{ \mathbf{r} \in \mathbb{R}^2 \mid \langle \mathbf{r}, \mathbf{n}_\tau(\theta_t) \rangle < b_\tau(\theta_t) + \langle v_t, \mathbf{n}_\tau(\theta_t) \rangle \}, \tag{6.10}$$

[6]Therefore, between two emitters firing the local shifts are $v_d = v/(K - 1)$ and $\theta_d = \theta/(K - 1)$.

where all terms depend smoothly on (v_t, θ_t), and therefore on (t, v, θ).

If we parametrise a ray r_{ij} by $\mathbf{r}(s) = \mathbf{p}_0 + s\mathbf{v}$, then substituting this into the half-space inequalities and solving for s finds that intersections occur at

$$s_\tau(v_t, \theta_t) = \frac{b_\tau(\theta_t) + \langle v_t, \mathbf{n}_\tau(\theta_t)\rangle - \langle \mathbf{p}_0, \mathbf{n}_\tau(\theta_t)\rangle}{\langle \mathbf{v}, \mathbf{n}_\tau(\theta_t)\rangle}, \tag{6.11}$$

whenever[7] $\langle \mathbf{v}, \mathbf{n}_\tau(\theta_t)\rangle \neq 0$. Away from $\mathcal{S}$, the intersection length $a_{ij}^k(t; v, \theta) := a_{ij}^k(v_t, \theta_t)$ is therefore given by the difference of the appropriate extrema of $\{ s_\tau(v_t, \theta_t) \}_{\tau=1}^4$, following the convention of Proposition 6.1.1. Hence, away from $\mathcal{S}$, $a_{ij}^k(t; v, \theta)$ depends continuously on (t, v, θ) and, since $\mathbf{A}(t; v, \theta)$ has finitely many entries, for a fixed set of times $\{ t_\kappa \}$ the map $\Gamma : (v, \theta) \mapsto \mathbf{A}(v, \theta)$ is continuous over $\mathcal{U} \setminus \mathcal{S}$, and piecewise continuous over $\mathcal{U}$. □

The analysis above[8] confirms that the construction of the forward operator is stable under perturbations of the motion parameters for all motion models that we have looked at. However, as stated previously, small perturbations in u may induce large changes in the minimiser $\mathbf{x}^*(u)$ of (4.4), and consequently in the value of $\mathcal{F}(u)$ itself. It is therefore the intrinsic ill-posedness of the inner inverse problem, rather than any instability in the forward model, that gives rise to the irregular behaviour observed in the objective function surfaces.

6.5 Experimental results: rigid-body motion

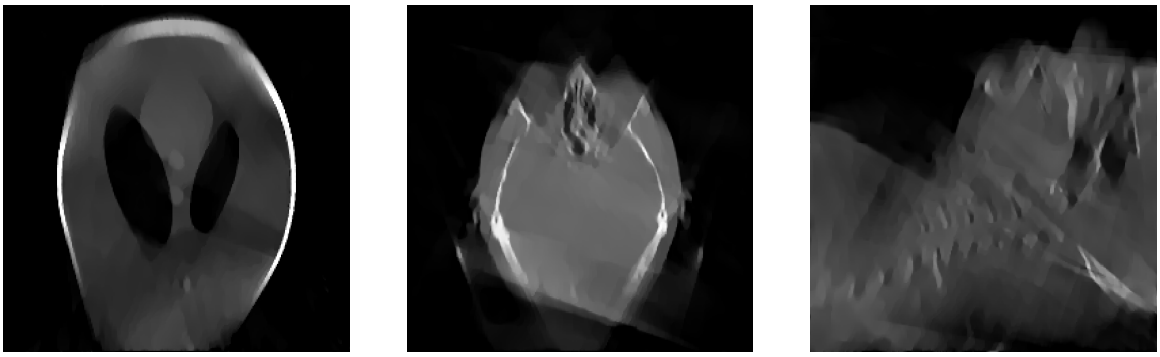
Rather than presenting a sequence of reconstructions for different rigid-body motions, we will instead look at two- and three-dimensional reconstructions together. Each model has a single representative example to showcase possible strengths and limitations to allowing full rigid-body motion to corrupt measurements.

While results in Figure 5.4 show that allowing the sub-domain to slowly leave the scan domain can improve image clarity, the results above in Figure 6.8 suggest that adding in additional rotation during translation does not necessarily improve reconstruction fidelity. As was the case with results in Figure 6.4, the inclusion of a full

[7] The denominator vanishes when r_{ij} is parallel to the edge with normal $\mathbf{n}_\tau(\theta_t)$. If the numerator also vanishes then the ray overlaps the edge, and it is exactly these configurations that constitute the set $\mathcal{S}$. If the numerator is non-zero, the ray is parallel to the edge at a positive distance, and the corresponding constraint is then either satisfied or violated independently of s—the analysis in the proof of Proposition 6.1.1 extends to rigid-body motions, so that $a_{ij}^k(t; v, \theta)$ is continuous.

[8] The analysis of the three-dimensional case is similar but requires a lot more construction, so we do not discuss it here.

revolution—which might have been expected to be beneficial—can instead highlight particular directions along which measurements are undersampled.



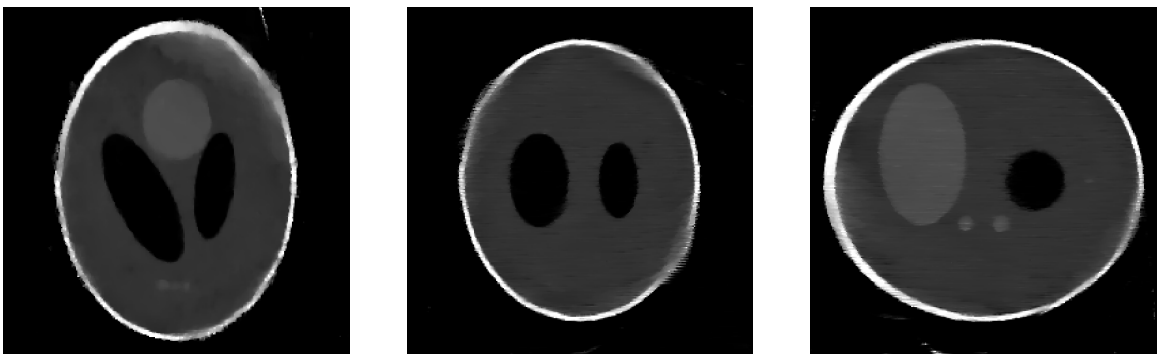
(a) (b) (c)

Figure 6.8: $[(m/n) \approx 0.2, \lambda = \lambda_{2D}]$ All phantoms are rotated by a full-revolution, while slowly exiting the scan domain: $\widehat{u} = (0.4, 0, 2\pi)^\top$. The combined effects of translation and rotation do not produce a corresponding improvement in reconstruction quality. SLP-2D has a slightly sharper support edge, but this is only fortunately related to the order in which emitters fire as the body moves.

Representative reconstructions in Figure 6.9 show that the benefits of incorporating rotation are preserved even in the presence of (possibly minor) translational motions. The model parameter used for these images is

$$\widehat{u} = (0.08\cos(\pi/3)\sin(\pi/6), 0.08\sin(\pi/3)\sin(\pi/6), 0.08\cos(\pi/6), \pi, \pi, \pi)^\top, \quad (6.12)$$

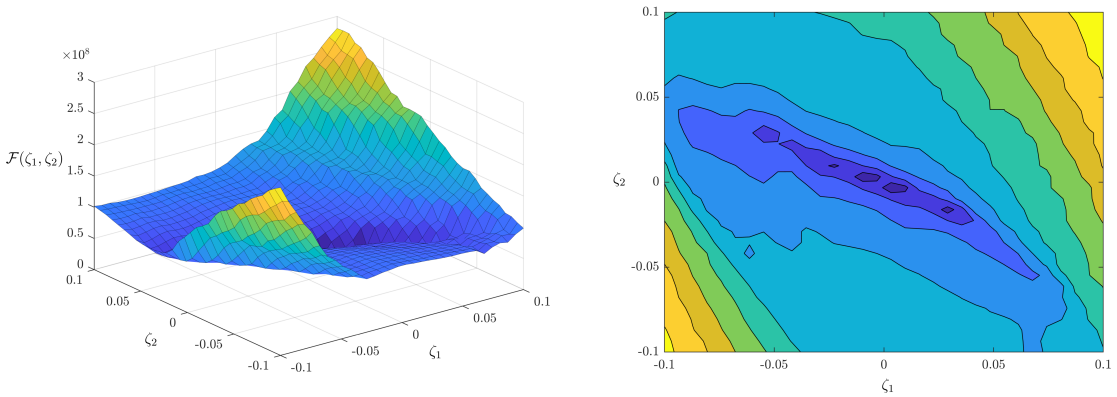
that is, the combined motion parameters from Figure 5.10 and Figure 6.6.



(a) (b) (c)

Figure 6.9: $[(m/n) \approx 0.2, \lambda = \lambda_{3D}]$ The combination of both rotational and translational motion in the three-dimensional model enables the reconstruction of median slices with clear support *and* internal fidelity. However, in comparison to Figure 6.6, the additional translation appears to have added marginal improvements.

In Figure 6.10 we include a representative example of a two-dimensional slice through the search space $\mathcal{U} \subset \mathbb{R}^6$; further experimentation has found that randomly sampled hyperplanes through $\mathcal{U}$ often have this type of behaviour. As can be seen in the contour plot, the location of the minimum becomes more obscured in this higher-dimensional setting, although it may be possible to navigate $\mathcal{U}$ effectively using subspace restrictions, if this behaviour is representative of the general case.



(a) Surface of the cost function $\mathcal{F}(u; \widehat{u})$, for $u \in \mathcal{U} \subset \mathbb{R}^6$.

(b) Contour plot of $\mathcal{F}(u; \widehat{u})$.

Figure 6.10: Two-dimensional slice through a randomly sampled affine subspace of $\mathcal{U}$, centred on the true parameter $\widehat{u}$. Axes are labelled with respect to the difference from $\widehat{u}$, so that the minimum occurs at (or near) zero, where ζ_1 and ζ_2 are linear combinations of the six motion parameters. Empirical results for rigid-body motion suggest that cost functions generally have similar behaviour to what is shown here, with directional alignment of the surface perhaps being caused by the difference in effective parameter bounds for translation versus rotation.

There is a clear difference between the surface shown in Figure 6.10 and those previously shown, as the gradient behaviour is strongly directional for the full rigid-body motion model. While this surface is only a two-dimensional sampling of a six-dimensional parameter space, the alignment of the surface behaviour suggests that the problem may be poorly conditioned along certain parameter directions.

This directional sensitivity can be understood through the differing effects of translation and rotation on the system matrix $\mathbf{A}(u)$. A translational perturbation δv induces a roughly uniform displacement of all voxels, with the change in each ray intersection length scaling proportionally to $\|\delta v\|_2$ relative to the voxel size. In contrast, a rotational perturbation $\delta\theta$ acts with a lever-arm effect—voxels at distance r from the centre of rotation are displaced by $r\delta\theta$, so that peripheral voxels experience

substantially larger displacements than central voxels, for the same perturbation. This effect is present in both DG-2 and DG-3, and becomes more pronounced as the spatial support of the phantom increases.

In three dimensions, the anisotropy is further compounded by the nontrivial structure of the rotation group SO(3). Unlike translations, three-dimensional rotations do not commute and so the rotation matrix depends nonlinearly on the rotational parameters describing the yaw, pitch, and roll. This suggests that the rotational subspace of $\mathcal{U}$ has an inherently more complex effect on $\mathbf{A}(u)$ than the translational subspace. As a result, $\mathcal{F}(u)$ is considerably more sensitive to perturbations in the rotational parameters than in the translational parameters. This imbalance in parameter sensitivity is an intrinsic feature of the rigid-body parametrisation rather than an artefact of the particular phantom used in the modelling, or the particular isolated hyperplane chosen shown above. This presents a clear challenge in the optimisation of rigid-body problems, and is identified here as a direction for future work. The optimisation strategy discussed in Chapter 7 treats $\mathcal{U}$ without explicit parameter scaling, which may present an avenue for future improvement.

The discussion in this chapter, and earlier work in Chapter 5, has led us to understand that the forward operator proposed in this work is stably defined and therefore provides a viable approach for motion correction. Diagnostic experiments have further shown that for an operator induced by any kind of rigid-body motion, the resulting reconstruction is generally diagnostically useful, in that internal details of the phantom remain identifiable. However, due to the inherent issue of limited-angle data, when motion parameters are small this may mean that the reconstruction difference is negligible, or worse, while solving (4.5) is still very difficult. As such, it may be that if only slight motions occur during image acquisition then the resulting data may not warrant the additional computational cost of motion correction.

As discussion here has moved from our original intent of analysing body motion to the inverted representation of device motion, the results obtained give a clear indication that certain motions, particularly three-dimensional rotations, can greatly improve reconstruction quality if they are adequately corrected for. In Chapter 7 we will focus discussion more on the optimisation strategies that may be employed for solving the original problem; that is, how to efficiently navigate the six-dimensional parameter space of a function $\mathcal{F}(u)$ which will be treated as a non-smooth, black-box function.

Chapter 7

Potential methodology improvements

Contents

7.1 Multilevel optimisation 111
7.2 Alternative reconstruction bases 128

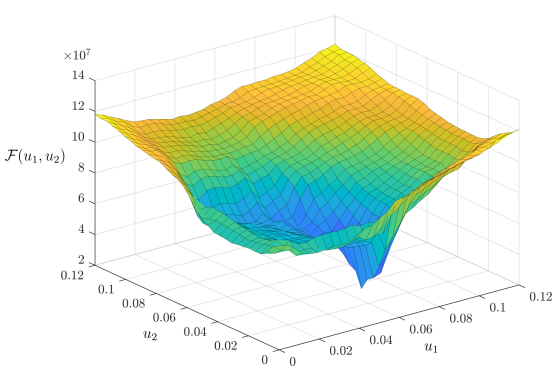
In Chapters 5 and 6 we presented a series of representative objective function surfaces for particular models of rigid-body motion. Each surface is impacted by both the particular geometry and acquisition setup of the chosen model, as well as the particular motion map $\Gamma : \mathcal{U} \to \Gamma(\mathcal{U})$ which induces the function $\mathcal{F}(u)$.

The lack of available information about $\mathcal{F}(u)$ makes it difficult to formally address with optimisation theory, especially with any kind of convergence guarantees. Therefore, in this chapter we instead present an approach that works empirically, but for which formal constructive theory is less understood.

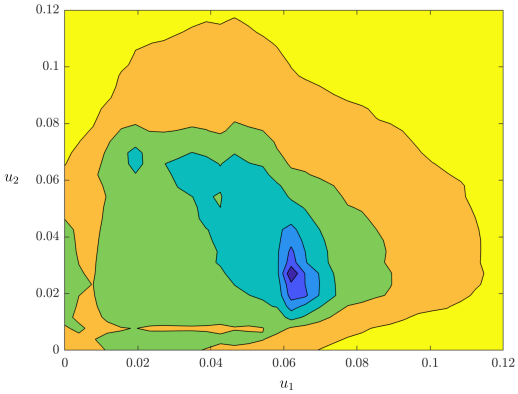
7.1 Multilevel optimisation

Throughout the problems presented, the particular parameters chosen—e.g. number of emitters, number of detectors, collimation angle—have been chosen to best represent the device being modelled in this work. In particular, the number of emitters in the device is a hard constraint, and so the only way to increase the number of measurements is to use a detector panel with a higher detector density. As such, the resolution of the discretised body, and therefore the size of the forward operator, has been chosen to represent an undersampling rate of between 10–30%.

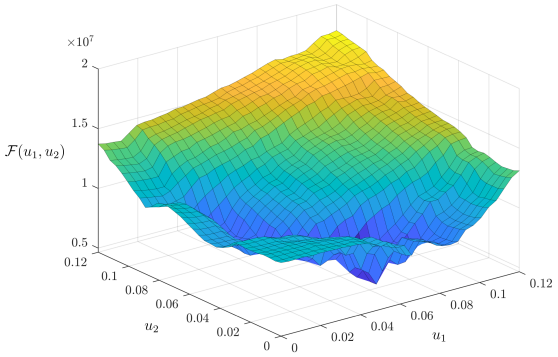
For high-resolution models[1], the construction of the forward operator can be very costly, and much effort has been spent on making sure that it can be constructed quickly. However, it is a fundamental issue in the solving of (4.5) that navigating the search space $\mathcal{U}$ requires forward operators to be constructed which, depending on the candidate u, may bear little resemblance to the true operator $\mathbf{A}(\widehat{u})$, and therefore may be a large computational investment for little payoff.



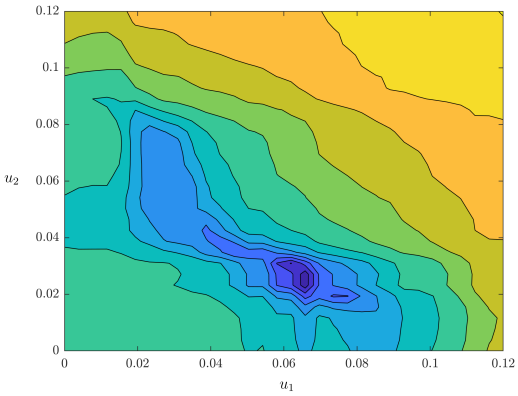
(a) $\mathcal{F}(u_1, u_2)$ at 128^3 voxel resolution.



(b) Contour plot of $\mathcal{F}(u_1, u_2)$ at 128^3 voxel resolution.



(c) $\mathcal{F}(u_1, u_2)$ at 64^3 voxel resolution.



(d) Contour plot of $\mathcal{F}(u_1, u_2)$ at 64^3 voxel resolution.

Figure 7.1: (a,b) These surfaces are already shown in Figure 5.11, for which a 128^3 spatial resolution model is solved in (4.5). (c,d) These plots are obtained from solving the same problem but with a spatial resolution of only 64^3 voxels. If the behaviour of the lower-resolution model is similar enough to the high-resolution model, we may expect optimisation algorithms to work well enough to find a good approximation to $\widehat{u}$, to then be verified in the higher-resolution model.

[1]While the term 'high-resolution' depends entirely on the context of the required image, we will refer to models for which the number of parameters is 10^6 or greater as 'high-resolution'. In particular, our 3D reconstructions involving SLP-3D are already in this regime, with $128^3 \approx 2 \times 10^6$ voxels each.

We now instead take the opposing viewpoint, such that we may exploit lower-resolution models to determine a good approximation to $\widehat{u}$ more quickly. Empirically, we have observed that the computational cost of solving (4.4) does not scale linearly with the number of unknowns—iterative solvers require repeated matrix-vector products whose cost grows with both the problem dimension and the number of iterations required for convergence. To reduce the total computational cost of solving (4.5), it may instead be preferable to use cheaper linear systems to navigate the space $\mathcal{U}$ more effectively.

Therefore, while using low-resolution models may increase the number of functional evaluations required to find a satisfactory approximation of $\widehat{u}$, the total computational cost may be significantly reduced as the evaluations will be with respect to much smaller systems. We observe that for the models discussed in Chapters 5 and 6 there is a visible similarity between the objective functions $\mathcal{F}(u)$ at different resolutions—we present a representative example in Figure 7.1.

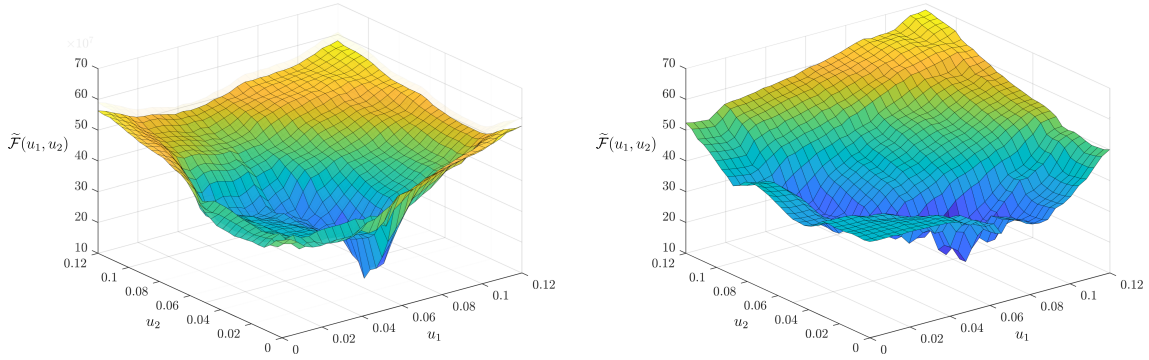
While the behaviour of the two surfaces is clearly not identical, they share two important properties: first, $\mathcal{F}(u)$ is minimised in a region close to the true parameter $\widehat{u}$; second, although the function $\mathcal{F}(u)$ is nonconvex, the general surface behaviour suggests that $\mathcal{F}(u)$ increases as approximations to $\widehat{u}$ become worse. We note that the objective function values are not directly comparable across resolutions, as the absolute scale of $\mathcal{F}(u)$ depends on the number of voxels in the discretisation. We may therefore rescale by n to give a per-voxel objective $\widetilde{\mathcal{F}}(u) = \mathcal{F}(u)/n$—see Figure 7.2 for a more direct comparison.

Both plots are presented on the same z-axis scale, to better compare the relative steepness of the two objective functions. Under this normalisation, the surfaces at different resolutions are directly comparable, and the empirical similarity between the 64^3 voxel model and 128^3 voxel model is retained.

7.1.1 Multilevel recursion for nonlinear optimisation

A standard way to exploit a hierarchy of discretisations in nonlinear optimisation is to treat the problem as inherently multilevel, in which one has a family of objective functions $\{ f_l \}_{l=0}^{L}$, such that increasing l corresponds to a reduction in spatial resolution, and lower evaluation cost. In this setting, *multilevel optimisation* borrows the philosophy of multigrid solvers in numerical linear algebra: coarse levels are used to obtain information cheaply, e.g. directions or approximate corrections, and this

information is transferred to finer levels via restriction and prolongation operators. Work by Nash [109] formalises this by coupling an optimisation iteration to the discretisation hierarchy, so that solutions to coarse-level problems accelerate progress on the finest-level problem.



(a) Normalised objective $\widetilde{\mathcal{F}}$ at resolution $n = 128^3$.

(b) Normalised objective $\widetilde{\mathcal{F}}$ at resolution $n = 64^3$.

Figure 7.2: The per-voxel normalisation $\widetilde{\mathcal{F}}(u) = \mathcal{F}(u)/n$ suggests that lower-resolution models are suitable substitutes to better navigate the feasible parameter space $\mathcal{U}$.

A particularly well-known implementation of this idea is *recursive multilevel trust-region* by Gratton et al. [61, 62]. In this framework, each level l has its own trust-region model—typically built from first- and second-order information of a smooth objective function—and coarse-level sub-problems are used recursively to compute candidates for steps at finer levels. Crucially, the theory relies on *model consistency*, such that the coarse model approximates the fine model in a controlled way, while also requiring smoothness, so that truncated Taylor expansion models behave well.

In our approach, we are only inheriting the motivating idea to use cheaper models, rather than the availability of the classical framework. Our cost function $\mathcal{F}(u)$ is effectively a black-box, and so it may easily break the usual assumptions needed to justify quadratic trust-region models across different discretisation levels. As a result, we discuss our multilevel models under the view of empirical similarity, rather than guaranteed model consistency between levels.

7.1.2 First-order approximation of inexact surfaces

While there is no guarantee that any version of $\mathcal{F}(u)$ is regular, the general behaviour of the surfaces seems well-controlled from the perspective of where the minimum

is located, i.e. the surfaces do not appear to present pathological difficulties for optimisation, although it is certainly likely that local minima exist. With this in mind, we may discuss the possibility of approximating first-order information to obtain search directions along which to perform line-search minimisation.

Finite-difference schemes can be fragile even for smooth functions, as balancing numerical error against truncation error can cause rapid degradation of the difference quotient [73, 111]. For noisy functions, or functions for which global behaviour is known but local behaviour is not, it is often better to use sample-based linear models to estimate the best fit over a local stencil. In essence, we ignore the lack of differentiability of $\mathcal{F}(u)$ and instead estimate a direction that reflects the resolvable, large-scale trend.

7.1.2.1 Least-squares gradient estimation

We present here work by Belward et al. [8] and Turner et al. [140] on estimation of gradients using a least-squares approach on scattered data. The particular utility presented in these papers is that of interpolation for finite-volume discretisation, and for surface reconstruction; however, the methods are general and can be used for gradient approximation. The result we discuss here is constructed for surfaces represented by functions of two variables, i.e. $f : D \subset \mathbb{R}^2 \to \mathbb{R}$, although it can be extended for higher-dimensional forms.

We first assume that the function representing the surface $z = f(x, y)$ is differentiable up to order n, and that there exists a set of m points $\{(x_i, y_i)\}_{i=1}^m$ for which a set of data $\{f(x_i, y_i)\}_{i=1}^m$ is associated. Additionally, we choose a point $\mathbf{a} = (a_x, a_y) \in D$ at which we want to approximate the gradient vector $\nabla f(\mathbf{a})$. Typically, we assume that $\mathbf{a}$ lies within the convex hull of the data points $\{(x_i, y_i)\}_{i=1}^m$. However, no assumption is made about the relationship between $f(\mathbf{a})$ and the evaluation of the data, $\{f(x_i, y_i)\}_{i=1}^m$.

We redefine the points $\boldsymbol{\nu}_i := (x_i, y_i)$ via the relationship[2] $\boldsymbol{\nu}_i = \mathbf{a} + h_i\boldsymbol{\eta}_i$, such that h_i is the Euclidean distance $\|\boldsymbol{\nu}_i - \mathbf{a}\|_2$ and $\boldsymbol{\eta}_i = (\eta_{x_i}, \eta_{y_i})$ is the unit vector $(\boldsymbol{\nu}_i - \mathbf{a})/h_i$.

We then take an nth order Taylor expansion for each data point $\boldsymbol{\nu}_i$, such that

$$f(\mathbf{a} + h_i\boldsymbol{\eta}_i) = f(\mathbf{a}) + h_i\frac{(\boldsymbol{\eta}_i \cdot \nabla)f(\mathbf{a})}{1!} + \cdots + h_i^n\frac{(\boldsymbol{\eta}_i \cdot \nabla)^n f(\mathbf{a})}{n!} + \mathcal{O}(h^{n+1}). \tag{7.1}$$

Grouping zeroth-order terms gives approximate directional derivatives

$$q_i = \frac{f(\mathbf{a} + h_i\boldsymbol{\eta}_i) - f(\mathbf{a})}{h_i}, \tag{7.2}$$

[2]The notation $\boldsymbol{\eta}_i$ here has no connection to the modelling error $\boldsymbol{\eta}$ defined in (3.7).

from which we may form a linear system $\mathbf{B}\boldsymbol{\gamma} = \mathbf{q}$, with the unknown vector being

$$\boldsymbol{\gamma} := \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial x \partial y}, \frac{\partial^2 f}{\partial y^2}, \ldots, \frac{\partial^n f}{\partial x \partial y^{n-1}}, \frac{\partial^n f}{\partial y^n} \right)^\top, \tag{7.3}$$

and the system matrix having the form

$$\mathbf{B} = \begin{bmatrix} \eta_{x_1} & \eta_{y_1} & \frac{h_1}{2}\eta_{x_1}^2 & h_1\eta_{x_1}\eta_{y_1} & \frac{h_1}{2}\eta_{y_1}^2 & \cdots & \frac{h_1^{n-1}}{(n-1)!}\eta_{x_1}\eta_{y_1}^{n-1} & \frac{h_1^{n-1}}{n!}\eta_{y_1}^n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \eta_{x_m} & \eta_{y_m} & \frac{h_m}{2}\eta_{x_m}^2 & h_m\eta_{x_m}\eta_{y_m} & \frac{h_m}{2}\eta_{y_m}^2 & \cdots & \frac{h_m^{n-1}}{(n-1)!}\eta_{x_m}\eta_{y_m}^{n-1} & \frac{h_m^{n-1}}{n!}\eta_{y_m}^n \end{bmatrix}. \tag{7.4}$$

In the minimal case, if only gradient approximations are needed, we may restrict to $\boldsymbol{\gamma} \in \mathbb{R}^2$ and form an $m \times 2$ overdetermined system, solved via the normal equations[3]

$$\widetilde{\boldsymbol{\gamma}} = (\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{q}, \tag{7.5}$$

yielding a least-squares approximation to the true gradient $\nabla f(\mathbf{a})$. By retaining higher-order information in $\mathbf{B}$—that is, assuming greater regularity of f and expanding to order n—gradient estimates may be obtained which benefit from improved convergence properties. The following proposition, taken from [140] without proof, makes this precise.

Proposition 7.1.1. Let $f : D \subset \mathbb{R}^2 \to \mathbb{R}$ be in the class C^n, let $\mathbf{a} \in D$, and let $\mathcal{B}(\mathbf{a}) \subset D$ be an open neighbourhood of $\mathbf{a}$. Suppose around the point $\mathbf{a}$ we have m neighbouring points $\boldsymbol{\nu}_i$ for $i = 1, \ldots, m$, with $\{\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_m\} \subset \mathcal{B}(\mathbf{a})$.

Let us further suppose that the mixed derivatives $\frac{\partial^n f}{\partial x^{n-j} \partial y^j}$ are Lipschitz continuous within $\mathcal{B}(\mathbf{a})$ for $j = 0, \ldots, n$, such that each has an associated Lipschitz constant $\mathcal{L}_j$.

Defining the matrix

$$\mathbf{E}_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{2 \times p}, \tag{7.6}$$

for $p = \frac{(n+1)(n+2)}{2} - 1$, and taking

$$\widetilde{\boldsymbol{\gamma}} := \underset{\boldsymbol{\gamma} \in \mathbb{R}^p}{\arg\min} \, \|\mathbf{B}\boldsymbol{\gamma} - \mathbf{q}\|_2 \tag{7.7}$$

to be the least-squares solution, where $\mathbf{B}$ and $\boldsymbol{\gamma}$ are defined as above, we approximate the local gradient at $\mathbf{a}$ by $\mathbf{E}_1 \widetilde{\boldsymbol{\gamma}}$.

[3] As is well-known in numerical linear algebra, the normal equations are often ill-conditioned and so it may be useful to precondition the system with respect to inverse-distance (squared, or otherwise) to give more weight to data points for which h_i is small.

A bound on the error in the least-squares gradient approximation is then given by

$$\|\nabla f(\mathbf{a}) - \mathbf{E}_1\widetilde{\boldsymbol{\gamma}}\|_2 \leq \frac{\mathcal{L}_{\max} h_{\max}^n}{(n+1)!\sigma_{\min}} \sqrt{\sum_{i=1}^{m} \|\boldsymbol{\eta}_i\|_1^{2n}}, \tag{7.8}$$

where $\sigma_{\min}$ is the smallest singular value of $\mathbf{B}$ (assuming $\mathbf{B}$ has full rank), $\mathcal{L}_{\max} = \max\limits_{0 \leq j \leq n} \{ \mathcal{L}_j \}$, $h_{\max} = \max\limits_{1 \leq i \leq m} \{ h_i \}$, and n denotes the smoothness of f.

The authors of [140] note that while the bound suggests $\mathcal{O}(h_{\max}^n)$ accuracy, the singular values of the matrix $\mathbf{B}$ depend on the values $\{ h_i \}_{i=1}^m$ and so this may be misrepresentative. However, the effects of $\sigma_{\min}$ can be alleviated via orthogonal projections—specifically, projecting onto the subspace orthogonal to higher-order columns of $\mathbf{B}$—such that the associated constant in (7.8) is independent of $\{ h_i \}$, and the convergence is truly $\mathcal{O}(h_{\max}^n)$. This agrees with previous numerical results in [9] which achieve quadratic convergence rates for least-squares matrices $\mathbf{B} \in \mathbb{R}^{m \times 5}$.

7.1.3 Nondifferentiable multilevel recursion

Given the computational cost faced in solving (4.5) at a high spatial resolution, it would be beneficial to draw on multilevel theory, using cheaper function evaluations to obtain useful directional information, to then navigate the parameter space $\mathcal{U}$ more easily. As experiments suggest, and the nondifferentiability of the one-norm attests, we cannot assume that the functions to be optimised fall into the classes described above. However, given certain modifications, it may be possible to borrow ideas from both to come to a middle ground of utility.

We now bring these ideas together within a framework for approximating gradients, with the goal of using cheaper approximations as proxies for more expensive models.

7.1.3.1 Assumptions on evaluated cost functions

With the perspective of our empirical results, let us assume the existence of a family of functions $\{ \mathcal{F}_l \}_{l=0}^L$ of the form

$$\mathcal{F}_l(u) = p_l(u) + \varepsilon_l \psi_l(u), \tag{7.9}$$

for which, over a domain $D \subset \mathbb{R}^d$ appropriate to the problem, we assume

- $p_l(u)$ to be smooth;

- ε_l to be a scalar governing error magnitude;
- $\psi_l(u)$ to be Lipschitz continuous with constant $\mathcal{L}_{\psi_l}$, but not differentiable.

Regarding our particular cost functions at different resolutions, we may interpret these terms as follows: $p_l(u)$ is a function that governs the general behaviour of $\mathcal{F}_l(u)$, perhaps just in a local area, but is not directly accessible; ε_l is a scaling term describing the observation that, while $\mathcal{F}_l(u)$ may be locally nonsmooth, it has large-scale behaviour that seems well-controlled, for which standard optimisation methods may make progress; $\psi_l(u)$ describes the effects of solver tolerance, discretisation artefacts, and nonsmoothness induced by our forward operator, for which we lose any expectation of differentiability.

Similar to the work by Gratton et al. [61, 62], we are merely assuming the existence of such functions for modelling purposes, although there is no means by which we can verify this assumption holds in practice.

7.1.3.2 Inexact least-squares gradient approximation

In the absence of noise and modelling error, we would be able to use the multilevel theory of [61, 109] to optimise a family of functions $\{\mathcal{F}_l\}_{l=0}^{L}$, with each being computationally cheaper than the previous. In particular, we would be able to obtain values for $\nabla\mathcal{F}_l$ at successively lower levels, and use gradient information as the reference between each level. Without the luxury of differentiability, we instead rely on first-order approximations using values of $\mathcal{F}_l$, with the goal of approximating ∇p_l in a region of $\mathcal{D}$ where the surface behaviour is well-behaved.

For the following, we set aside the multilevel framework momentarily and focus on a single function

$$\mathcal{F}(u) = p(u) + \varepsilon\psi(u), \tag{7.10}$$

which shares the properties given in Section 7.1.3.1. We will call $\mathcal{F}(u)$ the *model* function, and $p(u)$ the *underlying* function, to then ask the question

How well can a linear approximation of $\mathcal{F}$ approximate ∇p?

Following [8, 140], we assume that there exists a set of m points $\{\, u_i \,\}_{i=1}^{m}$ to which a set of data $\{\, \mathcal{F}(u_i) \,\}_{i=1}^{m}$ is associated, and a point $\mathbf{a}$ at which we wish to approximate the gradient. In this new framework, we can attribute each datum

$$\mathcal{F}(u_i) = p(u_i) + \varepsilon\psi(u_i) \tag{7.11}$$

as a point-wise combination of the underlying function plus a (scaled) model error term. Repeating a similar construction of the least-squares problem in Section 7.1.2.1 we have a linear system $\mathbf{B}\boldsymbol{\gamma} = \mathbf{q}^{\varepsilon}$, for which (7.2) now becomes

$$q_i^{\varepsilon} = \frac{\mathcal{F}(\mathbf{a} + h_i\boldsymbol{\eta}_i) - \mathcal{F}(\mathbf{a})}{h_i} \tag{7.12}$$

$$= \frac{p(\mathbf{a} + h_i\boldsymbol{\eta}_i) + \varepsilon\psi(\mathbf{a} + h_i\boldsymbol{\eta}_i) - p(\mathbf{a}) - \varepsilon\psi(\mathbf{a})}{h_i}, \tag{7.13}$$

$$= \frac{p(\mathbf{a} + h_i\boldsymbol{\eta}_i) - p(\mathbf{a})}{h_i} + \frac{\varepsilon(\psi(\mathbf{a} + h_i\boldsymbol{\eta}_i) - \psi(\mathbf{a}))}{h_i}, \tag{7.14}$$

$$= q_i + \epsilon_i. \tag{7.15}$$

By this, we show that the pointwise first-order information at $\mathbf{a}$ towards each data point $f(u_i)$ is equal to the directional derivative information of p with a describable error term, ϵ_i.

In this way, we can write the linear system as $\mathbf{B}\boldsymbol{\gamma} = \mathbf{q} + \boldsymbol{\epsilon}$, where $\mathbf{q}$ is the vector that we would obtain from solving the least-squares problem with p alone[4] and $\boldsymbol{\epsilon}$ is the vector of describable errors, formed by concatenating ϵ_i terms.

As we assume that $\mathcal{F}$ is not differentiable, we do not expect to obtain the same representative gradient vector $\mathbf{E}_1\boldsymbol{\gamma}$ presented in Section 7.1.2.1. However, we may still solve the least-squares problem on the perturbed linear system to obtain first-order information—we will denote this by $\nabla\!\!\!\!\!\!\diagup\,\mathcal{F}(\mathbf{a})$. We use this notation to describe a form which we want to approximate a gradient ∇p without having direct access to p—it is clear that $\nabla\!\!\!\!\!\!\diagup\,\mathcal{F}$ exists by virtue of the least-squares problem.

With this in mind, we state the relationship between $\nabla\!\!\!\!\!\!\diagup\,\mathcal{F}$ and ∇p through the following proposition.

Proposition 7.1.2. Assume the same local least-squares construction as in Proposition 7.1.1, with $\mathbf{a} \in D$, $\mathcal{B}(\mathbf{a}) \subset D$, and a set of neighbouring points $\{\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_m\} \subset \mathcal{B}(\mathbf{a})$. Let[5] $\mathcal{F} : D \subset \mathbb{R}^2 \to \mathbb{R}$ be defined by $\mathcal{F} = p + \varepsilon\psi$ having the properties given in Section 7.1.3.1: p is smooth and so $p \in C^n(\mathcal{B}(\mathbf{a}))$; ε controls the error magnitude within $\mathcal{B}(\mathbf{a})$; ψ is Lipschitz continuous over $\mathcal{B}(\mathbf{a})$ with constant $\mathcal{L}_\psi$. Additionally assume that each mixed derivative $\frac{\partial^n p}{\partial x^{n-j}\partial y^j}$ is Lipschitz continuous on $\mathcal{B}(\mathbf{a})$ with constant $\mathcal{L}_j$, for $j = 0, \ldots, n$, and define $\mathcal{L}_{\max} = \max_{0 \le j \le n} \{\, \mathcal{L}_j \,\}$.

[4] Again, we stress that this is *not* representative of us being able to control model noise when solving (4.5); p is merely a smooth proxy for the general, possibly local, behaviour of the function.

[5] As stated before, this argument can be extended to higher dimensions using identical arguments.

Assuming that $\mathbf{B}$ (defined by (7.4)) is full-rank, a bound on the error in the least-squares gradient approximation is then given by

$$\|\nabla\mathcal{F}(\mathbf{a}) - \nabla p(\mathbf{a})\|_2 \leq \frac{\varepsilon\mathcal{L}_\psi\sqrt{m}}{\sigma_{\min}} + \frac{\mathcal{L}_{\max}h_{\max}^n}{(n+1)!\sigma_{\min}}\sqrt{\sum_{i=1}^m \|\boldsymbol{\eta}_i\|_1^{2n}}. \tag{7.16}$$

Proof. Assuming that $\mathbf{B}$ has full column rank, the least-squares operator may be written as $\mathbf{B}^\dagger = (\mathbf{B}^\top\mathbf{B})^{-1}\mathbf{B}^\top$, such that $\|\mathbf{B}^\dagger\|_2 = 1/\sigma_{\min}$ where $\sigma_{\min}$ is the smallest singular value of $\mathbf{B}$. Using similar notation to Proposition 7.1.1, it follows that

$$\begin{aligned}
\widetilde{\boldsymbol{\gamma}}^\varepsilon &:= \mathbf{B}^\dagger\mathbf{q}^\varepsilon, && (7.17)\\
&= \mathbf{B}^\dagger(\mathbf{q}+\boldsymbol{\epsilon}), && (7.18)\\
&= \mathbf{B}^\dagger\mathbf{q} + \mathbf{B}^\dagger\boldsymbol{\epsilon}. && (7.19)
\end{aligned}$$

Isolating the first-order terms by $\nabla\mathcal{F} = \mathbf{E}_1\widetilde{\boldsymbol{\gamma}}^\varepsilon$, we then find

$$\begin{aligned}
\nabla\mathcal{F} &= \mathbf{E}_1\mathbf{B}^\dagger\mathbf{q} + \mathbf{E}_1\mathbf{B}^\dagger\boldsymbol{\epsilon}, && (7.20)\\
&= \nabla p + \mathbf{E}_1\mathbf{B}^\dagger\boldsymbol{\epsilon}. && (7.21)
\end{aligned}$$

Therefore, the obtained linear approximation $\nabla\mathcal{F}$ is a perturbation of the underlying approximation ∇p by a value which we will define as $\nabla\psi := \mathbf{E}_1\mathbf{B}^\dagger\boldsymbol{\epsilon}$. We use a simple splitting argument to determine an upper bound for $\|\nabla\mathcal{F} - \nabla p\|_2$, such that

$$\begin{aligned}
\|\nabla\mathcal{F} - \nabla p\|_2 &= \|\nabla\mathcal{F} - \nabla p + \nabla p - \nabla p\|_2, && (7.22)\\
&\leq \|\nabla\mathcal{F} - \nabla p\|_2 + \|\nabla p - \nabla p\|_2, && (7.23)\\
&= \|\nabla\psi\|_2 + \|\nabla p - \nabla p\|_2. && (7.24)
\end{aligned}$$

We note that the second term on the right-hand side is precisely that which is obtained by the unperturbed least-squares approximation, and so can be bound by the result in Proposition 7.1.1. Therefore, we obtain a bound for $\|\nabla\psi\|_2 = \|\mathbf{E}_1\mathbf{B}^\dagger\boldsymbol{\epsilon}\|_2$ as follows:

$$\begin{aligned}
\|\mathbf{E}_1\mathbf{B}^\dagger\boldsymbol{\epsilon}\|_2 &\leq \|\mathbf{E}_1\|_2\,\|\mathbf{B}^\dagger\|_2\,\|\boldsymbol{\epsilon}\|_2, && (7.25)\\
&= \|\mathbf{B}^\dagger\|_2\,\|\boldsymbol{\epsilon}\|_2, && (7.26)\\
&= \frac{1}{\sigma_{\min}}\|\boldsymbol{\epsilon}\|_2, && (7.27)\\
&\leq \frac{1}{\sigma_{\min}}\varepsilon\mathcal{L}_\psi\sqrt{m}, && (7.28)
\end{aligned}$$

where the final inequality follows from Lipschitz continuity of ψ over the sampling neighbourhood, whereby

$$|\epsilon_i| = \frac{|\varepsilon(\psi(\mathbf{a} + h_i\boldsymbol{\eta}_i) - \psi(\mathbf{a}))|}{h_i} \leq \frac{\varepsilon\mathcal{L}_\psi h_i}{h_i} = \varepsilon\mathcal{L}_\psi. \tag{7.29}$$

It therefore follows that

$$\|\tilde{\nabla}\mathcal{F} - \nabla p\|_2 \leq \|\tilde{\nabla}\psi\|_2 + \|\nabla p - \tilde{\nabla} p\|_2, \tag{7.30}$$

$$\leq \frac{\varepsilon\mathcal{L}_\psi\sqrt{m}}{\sigma_{\min}} + \frac{\mathcal{L}_{\max} h_{\max}^n}{(n+1)!\sigma_{\min}} \sqrt{\sum_{i=1}^{m} \|\boldsymbol{\eta}_i\|_1^{2n}}. \tag{7.31}$$

□

This result is a standard least-squares perturbation estimate for data errors, and may be viewed heuristically as the propagation $\Delta\boldsymbol{\gamma} = \mathbf{B}^\dagger\Delta\mathbf{q}$. While the appearance of the factor $\sqrt{m}$ may seem problematic, in most implementations m is chosen *a priori* by defining a fixed local stencil so this term does not grow with resolution. In fixed-dimension problems, such as those considered here, Euclidean unit vectors satisfy a uniform bound on their ℓ_1-norm (e.g. $\|\boldsymbol{\eta}_i\|_1 \leq \sqrt{2}$ in $\mathbb{R}^2$), and so the summation term in the bound scales like $\mathcal{O}(\sqrt{m})$. Finally, the use of a global Lipschitz constant is often too conservative, and may be replaced by a smaller value within a given sampling neighbourhood.

7.1.3.3 Recursive nonlinear optimisation

The purpose of this basic analysis is to determine whether it is plausible to adopt a multilevel theory for our problem. We have found empirically that searching the motion space $\mathcal{U}$ can be sped up by using lower resolution models to more quickly identify regions that the correct parameter $\widehat{u}$ may lie in. Essentially, our concern is whether we can give a concrete understanding of when using lower-resolution models will present useful information that aids in solving the higher-resolution problem. Our problem appears amenable to a multilevel approach but, as the standard assumptions of model consistency and smoothness do not fit our objectives, we give a small, noise-aware extension that yields practical diagnostics, rather than convergence guarantees.

As in Section 7.1.3.1, we assume the existence of a family of functions $\{\,\mathcal{F}_l\,\}_{l=0}^{L}$ such that $\mathcal{F}_l = p_l + \varepsilon_l\psi_l$, where p_l, ε_l, and ψ_l have the properties previously stated. The function $\mathcal{F}_0$ is the most expensive function that we wish to deal with, and the

number of levels L is not fixed in advance. All that is required is that the cost of a single evaluation of $\mathcal{F}_l$ is strictly more expensive than evaluating $\mathcal{F}_{l+q}$ for all $q \in \{1, \ldots, L-l\}$, and all $l \in \{0, \ldots, L-1\}$.

For simplicity, we will discuss a bi-level scenario: a problem in which the functions $\mathcal{F}_l$ and $\mathcal{F}_{l+1}$ are two versions of a similar model problem, with $\mathcal{F}_{l+1}$ using a *cheap* model while $\mathcal{F}_l$ uses a *costly* model. A simple method would be to use the cheap model to approximate first-order information, using the least-squares approximations above, evaluated around a current iterate $u^{(k)}$. This search direction can then be used to find a new candidate $u^{(k+1)}$ for which

$$\mathcal{F}_{l+1}(u^{(k+1)}) - \mathcal{F}_{l+1}(u^{(k)}) < c_{l+1,k+1}, \tag{7.32}$$

where $c_{l+1,k+1} < 0$ is a tolerance which, generally, will depend on both the model level and the iteration number.

Upon finding a new candidate $u^{(k+1)}$, the costly model then determines whether the condition

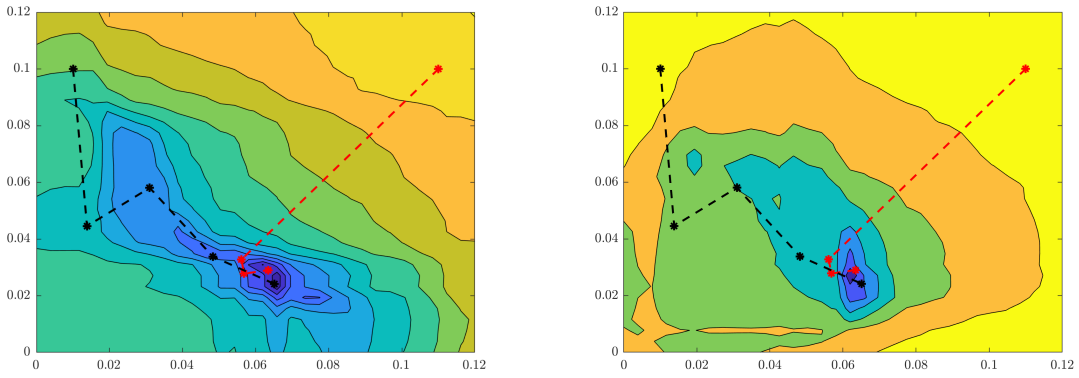
$$\mathcal{F}_l(u^{(k+1)}) - \mathcal{F}_l(u^{(k)}) < c_{l,k+1} \tag{7.33}$$

is met for a difference tolerance $c_{l,k+1} < 0$. If so, the newly sampled iterate is a genuine decrease of the costly function, and has likely been found with far fewer computational resources.

However, we should not expect that a decrease at one iterate implies the cheap model will continue to provide useful directions at subsequent iterates. Unfortunately, as we have described throughout this work, there is no guarantee that $\mathcal{F}(u)$ is well-behaved throughout $\mathcal{U}$, and so consistent agreement between models at all iterates cannot be assumed. A more suitable approach would be to use the cheaper model when the local behaviour between the two models is similar—that is, the local first-order behaviour is similar enough that a search direction employed on both models from the same point produces decreasing behaviour. In simple experiments, we have found this to be generally true; in Figure 7.3 we present results from optimising $\mathcal{F}(u)$ using a forward operator constructed over a model with a resolution of 64^3 voxels. The plot in Figure 7.3a shows the directional line searches of a pseudo-gradient descent algorithm, approximating first-order behaviour using a stencil of 10 data points with $h_i = h_{\max} = 0.01$. The first-order search direction is obtained by solving the inexact least-squares problem, and so we do not expect it to be the locally optimal direction.

However, for a large-scale line-search through the search space $\mathcal{U}$, we find that the approximations are still helpful.

The image in Figure 7.3b shows the contour of Figure 7.1b overlaid with the search directions obtained from the 64^3 resolution model in Figure 7.3a. As can be seen—also referring to the surface plot in Figure 7.1a—while the search directions from the cheaper model are not necessarily locally optimal for the costly model, the resulting behaviour is still a sequential decrease in the cost function. Moreover, the total computational cost of sampling 10 stencil points on the cheap model is, empirically, roughly twice the cost of a single function evaluation of the costly model.



(a) (b)

Figure 7.3: While the application of cheaper-model directional information has no guarantee of being useful for costly models, in experiments we see that it can greatly reduce computation time. In (a) we show the search trajectories from two different start points, with each trajectory computing approximate gradients and then a line-search along the direction. In (b) we show the same paths overlaid on the more expensive cost function, showing that the resulting steps adequately decrease the cost function for both models.

Ultimately, we want to ask the question

$$\textit{When can we expect } \tilde{\nabla}\mathcal{F}_{l+1}(u) \textit{ to be a good proxy for } \tilde{\nabla}\mathcal{F}_l(u)\textit{?}$$

The only condition that we will assume is that the underlying models $p_l(u)$ and $p_{l+1}(u)$ have similar behaviour to one another; we will quantify this in the analysis. If this were not true, then there would be no basis for assuming that the evaluation functions $\mathcal{F}_l(u)$ and $\mathcal{F}_{l+1}(u)$ would be related at all.

Using the analysis in Section 7.1.3.2, we know that $\tilde{\nabla}\mathcal{F}_l(u) = \nabla p_l(u) + \zeta_l(u)$, where $\zeta_l = \tilde{\nabla}\psi_l + (\tilde{\nabla}p_l - \nabla p_l)$ is the total approximation error presented in Proposition 7.1.2, with a bound $\|\zeta_l(u)\|_2 \leq \delta_l$. Similarly, we may describe the cheaper

function $\overset{\blacktriangledown}{\nabla}\mathcal{F}_{l+1}(u) = \nabla p_{l+1}(u) + \zeta_{l+1}(u)$, with $\|\zeta_{l+1}(u)\|_2 \leq \delta_{l+1}$. If we assume that the gradients $\nabla p_l(u)$ and $\nabla p_{l+1}(u)$ have some measure of consistency,

$$\|\nabla p_l(u) - \nabla p_{l+1}(u)\|_2 \leq \omega_l, \tag{7.34}$$

then it follows, by the triangle inequality, that

$$\|\overset{\blacktriangledown}{\nabla}\mathcal{F}_l(u) - \overset{\blacktriangledown}{\nabla}\mathcal{F}_{l+1}(u)\|_2 \leq \omega_l + \delta_l + \delta_{l+1}. \tag{7.35}$$

This analysis shows that the error between the two first-order approximations is bounded by the consistency between the underlying function gradients, as well as the combined effect of the approximation at each level. A more important metric to focus on is when the two approximations point in the same 'downhill' half-space[6], for which a sufficient condition is $\langle \overset{\blacktriangledown}{\nabla}\mathcal{F}_l(u), \overset{\blacktriangledown}{\nabla}\mathcal{F}_{l+1}(u)\rangle > 0$.

We therefore find that

$$\begin{aligned}\langle \overset{\blacktriangledown}{\nabla}\mathcal{F}_l(u), \overset{\blacktriangledown}{\nabla}\mathcal{F}_{l+1}(u)\rangle = \langle \nabla p_l(u), \nabla p_{l+1}(u)\rangle &+ \langle \nabla p_l(u), \zeta_{l+1}(u)\rangle \\ &+ \langle \zeta_l(u), \nabla p_{l+1}(u)\rangle \\ &+ \langle \zeta_l(u), \zeta_{l+1}(u)\rangle,\end{aligned} \tag{7.36}$$

for which the first term has the lower bound

$$\begin{aligned}\langle \nabla p_l(u), \nabla p_{l+1}(u)\rangle &= \langle \nabla p_l(u), \nabla p_l(u)\rangle + \langle \nabla p_l(u), \nabla p_{l+1}(u) - \nabla p_l(u)\rangle && (7.37)\\ &= \|\nabla p_l(u)\|_2^2 + \langle \nabla p_l(u), \nabla p_{l+1}(u) - \nabla p_l(u)\rangle && (7.38)\\ &\geq \|\nabla p_l(u)\|_2^2 - \|\nabla p_l(u)\|_2\, \omega_l && (7.39)\\ &= \|\nabla p_l(u)\|_2\, (\|\nabla p_l(u)\|_2 - \omega_l). && (7.40)\end{aligned}$$

Therefore, if the gradient of the underlying costly model is sufficiently large, such that $\|\nabla p_l(u)\|_2 > \omega_l$, this guarantees that the two underlying model gradients point in the same half-space.

The remaining terms in (7.36) can be bounded conservatively, obtaining

$$\begin{aligned}|\langle \nabla p_l(u), \zeta_{l+1}(u)\rangle| &\leq \|\nabla p_l(u)\|_2\, \delta_{l+1}, && (7.41)\\ |\langle \zeta_l(u), \nabla p_{l+1}(u)\rangle| &\leq \|\nabla p_{l+1}(u)\|_2\, \delta_l, && (7.42)\\ |\langle \zeta_l(u), \zeta_{l+1}(u)\rangle| &\leq \delta_l \delta_{l+1}. && (7.43)\end{aligned}$$

[6]For any set of points $\{ u_i \}_{i=1}^m$ there is no guarantee that the least-squares solution will give a decrease direction. Therefore, we are only concerned with the idea that if $\overset{\blacktriangledown}{\nabla}\mathcal{F}_{l+1}(u)$ gives a particular direction, then $\overset{\blacktriangledown}{\nabla}\mathcal{F}_l(u)$ would have given us a similar enough direction had we spent the computational effort there.

Since $\|\nabla p_{l+1}(u)\|_2 \leq \|\nabla p_l(u)\|_2 + \omega_l$, we get the lower bound

$$\langle \tilde{\nabla}\mathcal{F}_l(u), \tilde{\nabla}\mathcal{F}_{l+1}(u)\rangle \geq \|\nabla p_l(u)\|_2^2 - \|\nabla p_l(u)\|_2 (\omega_l + \delta_l + \delta_{l+1}) - \omega_l\delta_l - \delta_l\delta_{l+1}. \quad (7.44)$$

This can be interpreted as follows: as long as the *underlying* descent direction dominates the combined effects of both the model mismatch (ω_l) and the inexact gradient approximations (δ_l and δ_{l+1}) then we will find directional agreement. While this is a conservative result, the failure modes agree with what we observe; if $\|\nabla p_l(u)\|_2$ is too small then model error can dominate and the surface becomes too noisy to optimise reliably. Additionally, if ω_l is large then the models are too distinct, and there is no reason for the gradient approximations to be similar.

A further restriction is not just to require that $\langle \tilde{\nabla}\mathcal{F}_l(u), \tilde{\nabla}\mathcal{F}_{l+1}(u)\rangle$ be positive, but to state that there must be a minimum angle between the two. This, understandably, puts more restrictions on the underlying functions and the gradients that they produce, leading to the following proposition.

Proposition 7.1.3. Let δ_l, δ_{l+1}, and ω_l be the bounds given above, and assume $\delta_l \leq \|\nabla p_l(u)\|_2$, $\delta_{l+1} \leq \|\nabla p_{l+1}(u)\|_2$, and $\omega_l \leq \min(\|\nabla p_l(u)\|_2, \|\nabla p_{l+1}(u)\|_2)$. For a chosen angle $\varphi \in [0, \pi]$, a sufficient condition for $\angle(\tilde{\nabla}\mathcal{F}_l(u), \tilde{\nabla}\mathcal{F}_{l+1}(u)) \leq \varphi$ is

$$\begin{aligned}\arcsin\left(\frac{\delta_l}{\|\nabla p_l(u)\|_2}\right) + \arcsin\left(\frac{\delta_{l+1}}{\|\nabla p_{l+1}(u)\|_2}\right) \\ + \arcsin\left(\frac{\omega_l}{\min(\|\nabla p_l(u)\|_2, \|\nabla p_{l+1}(u)\|_2)}\right) \leq \varphi. \quad (7.45)\end{aligned}$$

Proof. If $\delta_l \leq \|\nabla p_l(u)\|_2$ then the angle between $\nabla p_l(u)$ and $\tilde{\nabla}\mathcal{F}_l(u)$ satisfies

$$\angle(\nabla p_l(u), \tilde{\nabla}\mathcal{F}_l(u)) \leq \arcsin\left(\frac{\delta_l}{\|\nabla p_l(u)\|_2}\right). \quad (7.46)$$

Similarly, if $\delta_{l+1} \leq \|\nabla p_{l+1}(u)\|_2$ then

$$\angle(\nabla p_{l+1}(u), \tilde{\nabla}\mathcal{F}_{l+1}(u)) \leq \arcsin\left(\frac{\delta_{l+1}}{\|\nabla p_{l+1}(u)\|_2}\right). \quad (7.47)$$

Defining $\varpi := \min(\|\nabla p_l(u)\|_2, \|\nabla p_{l+1}(u)\|_2)$, the assumption $\omega_l \leq \varpi$ then finds

$$\angle(\nabla p_l(u), \nabla p_{l+1}(u)) \leq \arcsin\left(\frac{\omega_l}{\varpi}\right). \quad (7.48)$$

Using the triangle inequality for angles on the unit sphere,

$$\angle(\tilde{\nabla}\mathcal{F}_l, \tilde{\nabla}\mathcal{F}_{l+1}) \leq \angle(\tilde{\nabla}\mathcal{F}_l, \nabla p_l) + \angle(\nabla p_l, \nabla p_{l+1}) + \angle(\nabla p_{l+1}, \tilde{\nabla}\mathcal{F}_{l+1}). \quad (7.49)$$

Substituting the bounds above yields the required inequality[7]. □

An important step in this work is to understand that although there is an expected relationship between the measurement data $\widehat{\mathbf{b}}$ and the minimising pair $(u^*, \mathbf{x}^*)$, we are free to alter the underlying model and still try to solve the same problem. To be clear, outside of the physically constrained parameters describing the device, the measurement vector $\widehat{\mathbf{b}}$ is the only algebraic construction which is specified by the particular problem that we want to solve. As such, the resolution of the reconstruction is a variable that we are free to tune, which until now we have only done through the lens of testing undersampling ratios for reconstructions under known motions.

As the plots in Figure 7.3 suggest, navigation can be accelerated by using lower-resolution spatial models to more quickly identify tighter regions of $\mathcal{U}$ more cheaply. However, due to the existence of noisier areas around possible minimisers—see Figure 7.4—improving search space navigation does not entirely remove the need to evaluate the costly objective eventually. In fact, it may be that the noise levels for the higher-resolution model may be large enough that an entirely new framework for motion correction must be applied in this region. Experiments suggest that high-resolution evaluations of $\mathcal{F}(u)$ dominate the total runtime; the dominant cost is not how $\mathcal{U}$ is navigated overall, but is instead the cost of repeatedly constructing and applying high-resolution forward operators within each objective evaluation.

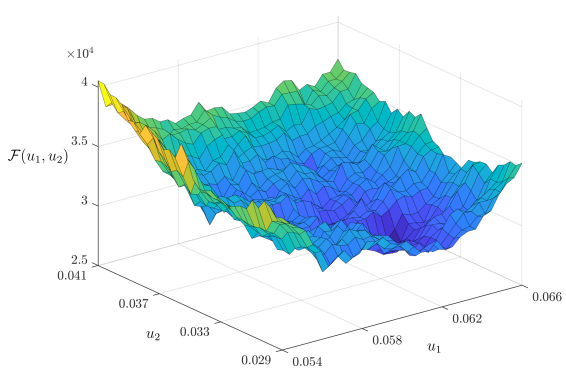
Additionally, while the images in Figure 7.4 suggest the need for costly high-resolution models, it is often the case that the high-resolution model itself is not adequate for accurately determining the true value of $\widehat{u}$. Worse still, depending on the resolution of the problem[8] we find that approximations to $\widehat{u}$ which are close enough to exist within a noisy region do not necessarily reconstruct anything diagnostically useful—see Figure 7.5.

This behaviour indicates that while the forward operator $\mathbf{A}(u)$ varies continuously with respect to the motion parameters $u \in \mathcal{U}$, the inner problem induced by $\mathbf{A}(u)$ is significantly more sensitive. Small perturbations in u, particularly in regions where the data fidelity term is dominated by noise or undersampling artefacts, can lead to

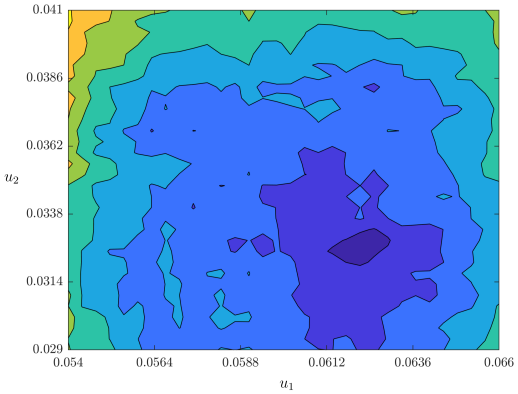
[7]The three $\arcsin(\cdot)$ terms have clear interpretations: the first bounds the angular deviation of $\nabla\!\!\!/\mathcal{F}_l$ from ∇p_l; the second bounds the angular deviation of $\nabla\!\!\!/\mathcal{F}_{l+1}$ from ∇p_{l+1}; the third bounds the angular deviation between ∇p_l and ∇p_{l+1} due to model mismatch. The condition then states that the total accumulated angular error from all three sources must not exceed φ.

[8]We note that it is not the sole effect of the model resolution, as the number of measurements affects this greatly, which, in turn, is related to the collimation angle, the number of emitters, and the number of detectors, and other geometric parameters of the acquisition setup.

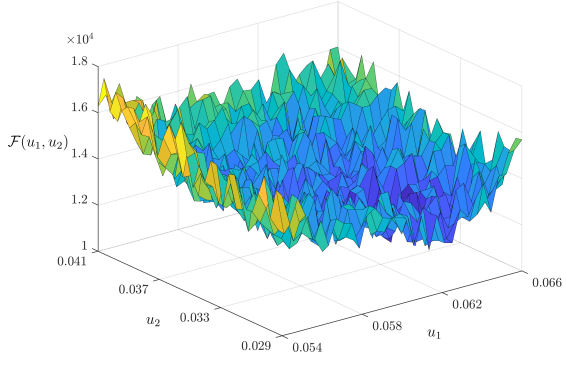
disproportionately large changes in the reconstructed image. From this perspective, accurate estimation of motion parameters does not necessarily lead to a high-quality reconstruction in our framework. The instability observed in the reconstructed images reflects the noise in the data, as well as the interaction between the discretised forward operator, the chosen representation, and the inherent ill-posedness of the inverse problem.



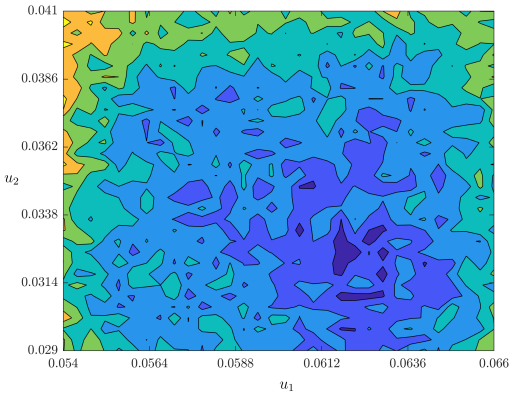
(a) Basin of Figure 7.1a around $\widehat{u}$.



(b) Contours of Figure 7.1b around $\widehat{u}$.



(c) Basin of Figure 7.1c around $\widehat{u}$.

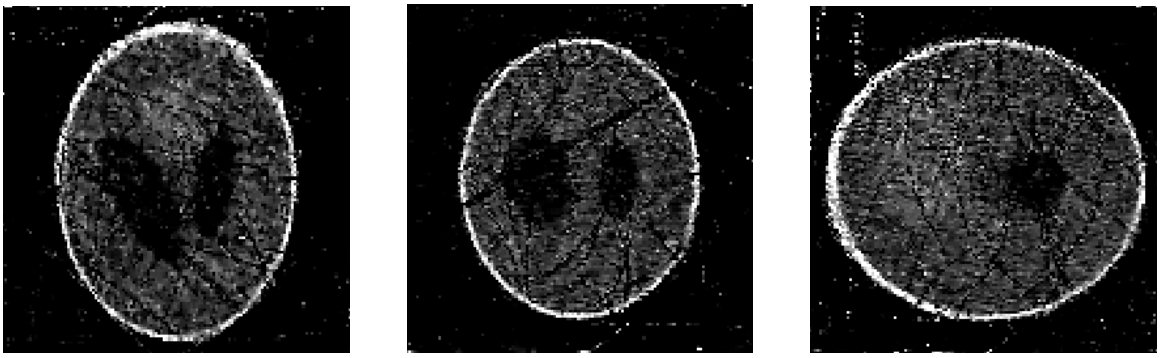


(d) Contours of Figure 7.1d around $\widehat{u}$.

Figure 7.4: Restricting attention to a region around the expected minimum, the behaviour of the cheaper model is almost entirely lost in noise. We note that the true motion parameter, $\widehat{u} = (0.0625, 0.0325)^\top$, is bounded within the lower levels of the contour map, but will almost certainly be impossible to determine.

This view motivates a shift in focus from optimisation strategy to image representation—if high-resolution evaluations of $\mathcal{F}(u)$ are unavoidable, and may not actually provide the level of reconstruction accuracy that we require, then improvements may instead come from reducing the sensitivity of the reconstruction process itself. In Part III, we investigate whether alternative spatial discretisations—

specifically periodic non-Cartesian geometries in two and three dimensions—can yield forward operators with improved numerical behaviour, leading to reconstructions that are more robust to small perturbations, without dramatically increasing computational cost.



(a) (b) (c)

Figure 7.5: $[(m/n) \approx 0.2, \lambda = \lambda_{3D}]$ Reconstructions of SLP-3D under rotation with parameter $\widehat{u} = (\pi, \pi, \pi)^\top$, from an incorrect parameter $u^\times$ for which $\|\widehat{u} - u^\times\|_2 < 10^{-3}$. Reconstructions under the forward operator $\mathbf{A}(u^\times)$ do not recover SLP-3D well, which is obvious when compared to images in Figure 6.6 for which $u^\times = \widehat{u}$. The approximate parameter $u^\times$ falls well within the noisy region surrounding $\widehat{u}$, from which it is very difficult to make more direct progress, and so alternative correction methods are needed to improve reconstruction beyond this.

7.2 Alternative reconstruction bases

The algebraic reconstruction framework developed in this thesis has, until this point, discretised the attenuation field $\mu(\mathbf{r})$ exclusively over a cubic voxel grid. While computationally convenient, this choice imposes an inherent anisotropy on both the forward model and the TV regularisation. Since TV regularisation penalises differences between face-connected neighbours, and these connections are aligned with the Cartesian axes, the regularisation is inherently anisotropic with respect to other directions. The choice of spatial basis therefore not only affects computational cost, but also determines the smoothness of the reconstructed field, the isotropy of the neighbourhood structure, and the class of objects that can be stably recovered from limited projection data. This section surveys the principal classes of basis representation that have been proposed in the tomographic reconstruction literature, before motivating the specific non-Cartesian discretisations developed in Part III.

Basis representations for tomographic reconstruction can be broadly organised into three categories: *non-overlapping tessellation bases*, in which the support of each basis element is disjoint from its neighbours; *smooth overlapping bases*, in which neighbouring elements share support and are coupled through the system matrix; and *transform-domain representations*, in which sparsity is imposed not on the spatial basis directly but on the coefficients of a secondary transform applied to the image. Each category offers distinct tradeoffs between smoothness, isotropy, computational tractability, and system matrix structure.

7.2.1 Non-overlapping tessellation bases

The simplest and most widely used class of basis functions are indicator functions supported on the cells of a spatial tessellation. Each basis element from the set $\{b_\gamma\}_{\gamma=1}^{M}$ is the characteristic function of a single cell v_γ, so that

$$b_\gamma(\mathbf{r}) = \mathbb{1}_{v_\gamma}(\mathbf{r}) = \begin{cases} 1 & \text{if } \mathbf{r} \in v_\gamma, \\ 0 & \text{otherwise,} \end{cases} \tag{7.50}$$

and the attenuation field is then approximated as

$$\mu(\mathbf{r}) \approx \sum_{\gamma=1}^{M} c_\gamma b_\gamma(\mathbf{r}), \tag{7.51}$$

where c_γ is constant over each cell. The interiors are disjoint by construction, so the system matrix $\mathbf{A}$ inherits a sparse structure in which each entry $\mathbf{A}_{ij}$ records the intersection length of a particular ray r_i with cell v_j. The near-universal choice in three-dimensional tomography is cubic voxels, in which the tessellation is induced by the lattice $\mathbb{Z}^3$, scaled by a voxel side length Δ. The dominance of cubic voxels reflects historical convention and computational convenience. The regular grid admits simple memory addressing, efficient traversal algorithms, and straightforward implementation of finite-difference operators for TV regularisation. Its principal limitation is anisotropy—the six face-connected neighbours of a cubic voxel are aligned with the Cartesian axes, potentially introducing axis-aligned artefacts in undersampled reconstructions.

Alternative non-overlapping tessellations have been studied to mitigate this anisotropy. In two dimensions, hexagonal grids provide six equidistant face-connected neighbours compared to the four axis-aligned neighbours of a square pixel, yielding

a more isotropic neighbourhood stencil that is particularly advantageous for TV regularisation [85]. In three dimensions, the cells induced by the *body-centred cubic* (BCC) and *face-centred cubic* (FCC) lattices offer improved rotational symmetry, and have been studied in the context of volume rendering and signal representation [11, 12, 46]. However, neither lattice has seen widespread adoption in tomographic reconstruction, in part due to the additional complexity of implementing efficient ray-traversal algorithms on non-Cartesian grids. A further non-overlapping alternative is provided by tetrahedral mesh bases [10], which offer flexible, non-uniform partitions that can conform to object boundaries. However, the variable element geometry complicates the construction of efficient ray-traversal algorithms and may not yield the regular sparse structure that makes the algebraic framework computationally feasible at the scales considered here.

7.2.2 Smooth overlapping bases

A fundamentally different approach replaces indicator functions with smooth, overlapping basis elements. In this setting, neighbouring basis functions share support, introducing coupling between elements in the system matrix. The benefit is a smoother approximation of the continuous attenuation field, at the cost of a denser system matrix and more complex local update rules.

7.2.2.1 Kaiser–Bessel window functions

A well-studied class of smooth overlapping bases in tomographic reconstruction are the generalised Kaiser–Bessel window functions, introduced by Lewitt [94, 95] and sometimes referred to as *blob* basis functions. These are radially symmetric functions parametrised by a shape order $m \in \mathbb{N}$, a taper parameter $\alpha > 0$, and a support radius $a > 0$, defined by

$$b^{(m,\alpha,a)}(\mathbf{r}) = \begin{cases} \dfrac{(1-(r/a)^2)^{m/2}\, I_m\!\left(\alpha\sqrt{1-(r/a)^2}\right)}{I_m(\alpha)} & \text{if } r \le a, \\ 0 & \text{otherwise,} \end{cases} \tag{7.52}$$

where $r = \|\mathbf{r}\|_2$ and $I_m(\cdot)$ is the modified Bessel function of the first kind of order m.

The parameters (m, α, a) allow the smoothness, concentration, and spatial extent of the basis function to be controlled independently—larger m values produce a smoother function with more rapid decay towards the boundary of the support, larger α values

concentrate the function more strongly near the centre, and larger a values increase the support radius. The spherical symmetry of (7.52) simplifies the computation of projection integrals considerably, as the line integral of $b^{(m,\alpha,a)}$ along any ray at distance s from the centre of the basis element depends only on s. This makes Kaiser–Bessel bases computationally attractive for standard tomographic geometries, particularly in fan-beam and cone-beam CT. However, the overlapping support means that the matrix entries are non-zero for all elements $b^{(m,\alpha,a)}$ whose supports intersect the ray ℓ. This is necessarily a larger set than for indicator functions whenever the support radius a exceeds the lattice spacing, decreasing the sparsity of $\mathbf{A}$ and increasing the cost of both matrix construction and iterative updates. Furthermore, any local update to a coefficient c_γ affects (7.51) throughout the support of b_γ, complicating the local update rules that iterative solvers rely on in the indicator basis setting.

7.2.2.2 B-splines and Box splines

B-splines [142, 143, 144] provide an additional class of compactly supported basis functions with controllable regularity, and have found use in both two- and three-dimensional image reconstruction [15, 107]. The univariate B-spline of order n is defined recursively by the convolution

$$\beta^n(x) = \begin{cases} \mathbb{1}_{[-1/2,\,1/2)}(x) & \text{if } \; n = 0, \\ \beta^{n-1}(x) * \beta^0(x) & \text{if } \; n \geq 1, \end{cases} \tag{7.53}$$

so that β^n is the $(n+1)$-fold self-convolution of the box function β^0. The resulting function is a piecewise polynomial of degree n, with support $[-(n+1)/2, (n+1)/2]$ and regularity $\mathcal{C}^{n-1}(\mathbb{R})$. The compact support bounds the number of non-zero entries in $\mathbf{A}$ per ray, making B-splines computationally manageable [33]. However, for $n \geq 1$ the overlapping support introduces coupling between neighbouring coefficients, and system matrix entries require integration of a piecewise polynomial along each ray, which is more involved than the intersection-length computation used for indicator bases.

Box splines [34] generalise B-splines to multivariate settings over non-Cartesian lattices by extending convolutions along a set of prescribed directions encoded in a direction matrix $\mathbf{W} \in \mathbb{R}^{d \times s}$. This provides a natural framework for constructing smooth bases on BCC lattices [47, 48], and box splines on these lattices have been studied for tomographic reconstruction and volume rendering [53], where they offer

improved isotropy over Cartesian tensor-product B-splines. However, their practical adoption in tomographic reconstruction remains limited, in part due to the additional complexity of implementing efficient projection operators on non-Cartesian grids.

7.2.3 Transform-domain representations

A third class of approaches does not modify the spatial basis directly, but instead imposes sparsity on the coefficients of a secondary transform $\mathbf{\Phi}$ applied to the image, as in the formulation of (2.55). The spatial basis remains a standard voxel grid, but the regularisation is designed to exploit structure in a transformed domain.

Wavelet bases provide a multiscale decomposition of the image in which piecewise smooth signals are approximately sparse, and have been applied to CT reconstruction in several settings [24, 113], particularly for denoising and compressed sensing recovery. More directionally sensitive representations include curvelets, which provide near-optimal sparsity for images dominated by smooth curves and edges at multiple scales and orientations, and shearlets, which form a mathematically complete framework for multidirectional, multiresolution analysis and have been applied to limited-angle tomography [17, 153]. Contourlets [38] provide a discrete counterpart to curvelets, while framelet and tight-frame approaches [19] offer overcomplete dictionaries that can be adapted to specific image classes. More recently, data-driven approaches have replaced hand-crafted transforms with learned representations, including dictionary learning [2] and deep learning priors [6, 134]. While these approaches can provide strong sparsity for specific image classes, they introduce substantial additional modelling choices and their interaction with object motion is non-trivial.

7.2.4 Motivation for present work

The preceding survey identifies a fundamental tension between smoothness and computational tractability. Overlapping bases—Kaiser–Bessel, B-splines, box splines—offer improved approximation of continuous fields but at the cost of a denser system matrix, more complex projection operators, and loss of the sparse, geometrically transparent structure that supports computational tractability.

Non-overlapping tessellation bases retain this structure but can introduce anisotropy through the neighbourhood geometry of the underlying lattice. This motivates the investigation of optimal non-overlapping, non-Cartesian tessellations as a middle ground—retaining the computational advantages of indicator bases while improving

the isotropy of both the forward operator and the TV regularisation. In two dimensions, the hexagonal tessellation is the natural candidate, offering a six-connected isotropic neighbourhood and favourable properties for TV-based reconstruction [85]. In three dimensions, the rhombic dodecahedral tessellation provides an analogous improvement, with a twelve-connected face-neighbourhood that more uniformly samples the sphere of directions than the six-connected cubic neighbourhood.

Exact construction of the system matrix $\mathbf{A}$ via ray-element intersection lengths has not previously been demonstrated for non-Cartesian tessellation bases in an algebraic reconstruction framework, and the development of such algorithms for hexagonal and rhombic dodecahedral grids is the primary contribution of Part III.

Concluding remarks for Part II

Part II establishes a consistent framework for incorporating rigid-body motion directly into algebraic tomographic forward models. By treating motion as part of the operator rather than as an image-space correction, the analysis clarifies how rigid-body transformations interact with the domain discretisation, to understand the effects on the resulting system matrix. The continuity results presented provide a minimal theoretical foundation for this modelling choice, while avoiding assumptions strong enough to obscure the intrinsic difficulties of the problem.

The results included in Chapters 5 and 6 also make clear that continuity alone is insufficient to guarantee favourable optimisation behaviour. Even modest motion can incur unfavourable features in the associated objective functions, particularly under limited-angle acquisition. These effects are not necessarily artefacts of the specific solver used here, but arise from the interaction between motion, discretisation, and the ill-posedness of the objective itself. As such, these chapters should be read as a diagnostic analysis of problem structure, rather than as an attempt to engineer a complete motion-correction pipeline.

In Chapter 7 we outline possible routes for improvement. The first focuses on the utility of low-resolution reconstructions, which act as a cheap alternative for finding parameter approximations quickly. Empirical results suggest that the nested optimisation problem can be solved using multilevel approaches, yet there is no guarantee that lower-resolution information will always be useful. The second possible improvement comes from choosing a different domain discretisation, with the expectation that it may be more suitable for determining underlying motion, or is a better representation of the discrete object. The conclusions of Chapter 7 motivate a shift in emphasis for the remainder of the thesis.

If motion-induced instability is fundamentally linked to the discrete representation of the forward operator, then improvements may be sought not only through optimisation strategies, but through redesign of the underlying spatial discretisation and

ray-element interaction models themselves. Part III therefore turns to alternative discretisations and traversal algorithms, investigating whether changes at the level of representation can alter the numerical behaviour of the chosen solver, and ultimately the reconstruction quality of images undergoing rigid-body motion.

Part III

Alternative Discretisations for Tomography

Overview of Part III

As discussed in Part II, experimental results show that it is possible to obtain recognisable reconstructions for bodies undergoing motion, particularly when motion parameters are known. In addition to this, the cost function surfaces that result from these types of parametrised models may lend themselves to standard optimisation approaches to find neighbourhoods of the underlying parameters, perhaps exploiting cheaper functions to obtain pertinent information. However, it is unavoidable that once in a close neighbourhood of the true parameters it is necessary to utilise high-resolution models that have much greater computational cost.

The work presented in Part III reflects an attempt to alleviate this problem; rather than working under a new optimisation framework, we instead approach the problem from the perspective of model representation. The problem discussed throughout this thesis is fundamentally tied to the difficulties that result from limited-angle acquisition. Therefore, in our algebraic framework it may be preferable to use a discretisation of cells which have denser neighbourhood stencils, such that each voxel has more neighbours with which information can be shared, improving the isotropy of the TV regularisation. With this in mind, we investigate whether using a voxel basis more suited to exploiting a connected geometry will make the inner problem easier to solve, or may yield more stable objective function behaviour under perturbations.

Investigating this idea in two dimensions, the natural extension from square pixels—which have four edge-sharing neighbours—is hexagonal pixels. The primary drawback with using hexagonal pixels, at least compared to square pixels, is that it is generally much slower to compute the system matrix, due to weaker algorithms supporting the computation of pixel-ray intersections. To solve this problem, we develop two new algorithms for computing pixel-ray intersections through a hexagonal grid, presenting both in Chapter 8. The first algorithm is an extension of the *Fast Voxel Traversal* algorithm [3], developed for square and cubic grids. The second algorithm is an

extension of *Siddon's* algorithm [132] which, in this new hexagonal framework, allows for intersection lengths to be computed in an embarrassingly parallel fashion.

The three-dimensional analogue of these extensions is to transition from cubic voxels to rhombic dodecahedral voxels—tessellating cells which fill three-dimensional space, each of which has a neighbourhood stencil of 12 identical cells. For this choice of voxels, exact voxel-ray intersection lengths do not appear to have been previously studied in the tomographic reconstruction literature, and so we present a new algorithm for determining intersections which is also is fully parallelisable. This work comprises the remainder of Part III, and is presented in Chapter 9.

Chapter 8

Hexagonal grid representation

Contents

8.1 Hexagonal grid model 140
8.2 Ray traversal through Cartesian grids 142
8.3 Ray traversal through a hexagonal grid 146
8.4 Extending Siddon's algorithm to hexagonal pixels 158
8.5 Numerical results 173
8.6 Hexagonal image reconstruction 176

The representation of images using hexagonal pixels has been an open, albeit small, area of research since the early 1960s, emerging alongside the advent of digital image processing, with early work by Petersen [117] determining that hexagonal pixels form the best arrangement for sampling band-limited signals in two dimensions. For many practices in *hexagonal image processing* (HIP), hexagonal pixels are constructed via a sampling over a square pixel grid, with weighting suited to the particular purpose [71, 92]. Work by Mersereau [103] focused on the development of the hexagonal discrete Fourier transform, and different problems in HIP and computer vision have been tackled by a number of authors [45, 59, 110, 125]—a broader overview of HIP can be found in [105], which presents a general framework for the field itself. Still, the developments of HIP have not usurped the long-standing tradition of square-pixel discretisation, with its simplicity of both understanding and implementation.

Of particular relation to our work is the notion of total variation over hexagonal grids, for which a good overview can be found in [85]. The nature of hexagonal pixels—we will refer to these throughout as *hexels*—means that each has six isotropically connected neighbours, as opposed to the classical 4-cell or anisotropic 8-cell neighbourhoods of a

square pixel. Therefore, in a sense, more neighbouring information can be directly accessed by a single hexel than can be accessed by a square pixel. The analytical theory behind Total Variation methods on hexagonal grids is quite technical, but from the view of numerical implementation it is no more complex than for square pixels, simply applied to a different stencil. Results in [85] show that TV denoising over a hexagonal 6-cell stencil can achieve better results than an 8-cell stencil in a square pixel framework. While denoising is a different problem to what we have considered so far in this thesis, the utility of TV regularisation over hexagonal grids has clear overlap. It is with this result that we approach the forward operators in this work, using hexagonal grid discretisations to then reconstruct images with TV-norm regularisation over a 6-cell neighbourhood.

From the perspective of our work so far, the choice of discretisation interacts directly with the behaviour of the regularisation, i.e. the choices of both $\mathbf{D}$ and λ in (3.13). Since TV regularisation promotes piecewise-constant structure through local neighbour differences, the underlying pixel neighbourhood implicitly defines the directions along which edges are preferentially preserved or penalised. The increased angular uniformity of the hexagonal grid offers a more natural discretisation for TV-based reconstruction, particularly in undersampled regimes where anisotropy in the measurement direction can introduce dominant axis-aligned artefacts in the reconstructions. It is known that using hexagonal basis elements for image sampling enables better representation of curved surfaces [104, 105], and so we look to determine whether phantoms, particularly SLP-2D, can be more accurately reconstructed using a hexagonal grid.

For medical imaging specifically, hexagonal grids have been investigated for a number of modalities, including CT [138], PET [1], and *single photon emission computed tomography* (SPECT) [152]. Within these studies, almost all approaches obtain projection data without the computation of exact intersection lengths, using alternative resampling methods [88, 159]. As such, there is no current framework in which exact measurements are made naturally over a hexagonal grid, so the redesign of our forward operator $\mathbf{A}(u)$ will be built on these new developments.

8.1 Hexagonal grid model

Rebuilding the forward operator $\mathbf{A}(u)$ using hexel elements is naturally expressed through the notion of a *planar lattice*. A planar lattice is a discrete subset of $\mathbb{R}^2$,

induced by a basis $\{\mathbf{m}_1, \mathbf{m}_2\} \subset \mathbb{R}^2$. Taking $\mathbf{M}$ to be the matrix with columns $\mathbf{m}_1, \mathbf{m}_2$, the lattice is defined[1] by

$$\Lambda(\mathbf{M}) = \{\, \mathbf{M}\mathbf{z} \mid \mathbf{z} \in \mathbb{Z}^2 \,\}. \tag{8.1}$$

For hexels, the vectors are[2] chosen to be $\mathbf{m}_1 = (0, 1)^\top$, $\mathbf{m}_2 = (\cos(\pi/6), \sin(\pi/6))^\top$. For any member of the lattice, $\mathbf{c} \in \Lambda(\mathbf{M})$, we denote the associated Voronoi cell by

$$V(\mathbf{c}) := \{\, \mathbf{r} \in \mathbb{R}^2 \mid \|\mathbf{r} - \mathbf{c}\|_2 < \|\mathbf{r} - \mathbf{p}\|_2 \quad \forall\, \mathbf{p} \in \Lambda(\mathbf{M}) \setminus \{\mathbf{c}\} \,\}, \tag{8.2}$$

i.e. the set of all points $\mathbf{r} \in \mathbb{R}^2$ which are closer to $\mathbf{c}$ than any other member of $\Lambda(\mathbf{M})$.

The operator entries $\mathbf{A}_{ij}(u)$ will have similar behaviour to when they are computed via square pixels, in that all elements endow Γ with the same type of continuity guarantees[3], dependent on the underlying motion. We may therefore expect that there should be no inherent issue from hexagonal grids that specifically makes the forward operator less stable.

To understand the algorithm needed to construct our forward operator, we first discuss how pixel-ray intersections have been computed for the previous models, as until now this has been assumed without explicit discussion. In practice—especially as our problem involves recomputing operators for each solve of (4.5)—the computation of pixel-ray intersections can dominate the total runtime if not performed efficiently. In the following section we will discuss two efficient algorithms for computing intersection lengths through square grids—the Fast Voxel Traversal (FVT) algorithm, and Siddon's algorithm—which both have $\mathcal{O}(N)$ computational cost for grids of size $\mathcal{O}(N^d)$, for $d \in \{2, 3\}$. We will then present adaptations to our new hexel grid.

As a reminder of notation, for a ray r_{ij} our goal is to compute the *radiological path* given by

$$\mathfrak{L} = \sum_{k=1}^{n} a_{ij}^k x_k. \tag{8.3}$$

For an arbitrary ray r, we will instead denote the intersection lengths by ℓ_k, such that we index strictly over the pixels which intersect the ray, i.e. those for which $\ell_k \neq 0$. As we are only interested in computing the intersection lengths ℓ_k, we will use the term *radiological path* to refer more loosely to the set of lengths $\{\, \ell_k \,\}$ that constitute the distances travelled through a sequence of successive pixels.

[1] This implicitly defines the lattice as containing the point $\mathbf{0} \in \mathbb{R}^2$.

[2] Any rotation and/or translation of this vector pair is also valid—the pair chosen here creates the 'flat-top' oriented hexagonal grid.

[3] This holds from the fact that hexagons are convex polygons, so the analysis discussed in Propositions 5.1.1, 6.1.1 and 6.4.1 is also true in this context.

8.2 Ray traversal through Cartesian grids

Before presenting the new ray-traversal algorithm for hexagonal grids, we first review the two standard algorithms for Cartesian grids on which our extensions are based.

8.2.1 The Fast Voxel Traversal algorithm

The Fast Voxel Traversal algorithm, introduced by Amanatides and Woo [3], is a highly efficient method for traversing a uniform grid of axis-aligned square pixels along a ray. Unlike general ray-geometry intersection methods, which may involve costly geometry-specific computations, FVT exploits the regularity of the pixel grid to perform traversal using only simple arithmetic operations, producing an ordered list of pixel-ray intersection lengths and their corresponding pixel indices.

For a ray with origin $\mathbf{q} = (q_x, q_y)^\top \in \mathbb{R}^2$ and normalised direction $\mathbf{v} = (v_x, v_y)^\top \in \mathbb{R}^2$ the FVT algorithm determines the sequence of pixels that the ray intersects as it progresses through the grid. The key idea is to advance from one pixel to the next along the ray, determining at each step which pixel boundary is crossed first. The algorithm is presented in two stages: *Initialisation* and *Iteration*.

8.2.1.1 Initialisation

Let the pixel grid be indexed by integer coordinates in the x and y directions[4], and let each pixel be a square with side length $\Delta > 0$. The distance travelled along the ray between successive x coordinates is $\Delta_x = \Delta/|v_x|$, and for y coordinates it is $\Delta_y = \Delta/|v_y|$.

The algorithm begins by identifying the initial pixel index[5]

$$[\![i_x, i_y]\!] = [\![\lfloor q_x/\Delta \rfloor, \lfloor q_y/\Delta \rfloor]\!], \tag{8.4}$$

and the axis-aligned index shifts, $m_x = \operatorname{sgn}(v_x)$ and $m_y = \operatorname{sgn}(v_y)$.

The index shifts update the running pixel index as the ray passes through the grid, updating either i_x or i_y depending on the direction of the crossing. In the direction of $\mathbf{v}$ we compute the distances to the next pixel boundary for both x- and y-coordinates,

[4]Without loss of generality, we may assume that the grid *defines* the axial directions.

[5]To differentiate index pairs from points in $\mathbb{R}^2$ we shall denote the *'index one-two'* by $[\![1, 2]\!]$, while the *position one-two* by $(1, 2)$. Rather than writing '*the pixel with index* $[\![i, j]\!]$', we will simply write '*pixel* $[\![i, j]\!]$'.

denoting them t_x and t_y, respectively. These values can be explicitly computed, first by computing the relative position of the first pixel boundary to the origin

$$b_x = \begin{cases} \Delta\,(i_x + 1) & \text{if } v_x > 0, \\ \Delta\, i_x & \text{if } v_x < 0, \\ \text{undefined} & \text{if } v_x = 0, \end{cases} \qquad b_y = \begin{cases} \Delta\,(i_y + 1) & \text{if } v_y > 0, \\ \Delta\, i_y & \text{if } v_y < 0, \\ \text{undefined} & \text{if } v_y = 0, \end{cases} \tag{8.5}$$

and then computing

$$t_x = \begin{cases} (b_x - q_x)/v_x & \text{if } v_x \neq 0, \\ \infty & \text{if } v_x = 0, \end{cases} \qquad t_y = \begin{cases} (b_y - q_y)/v_y & \text{if } v_y \neq 0, \\ \infty & \text{if } v_y = 0. \end{cases} \tag{8.6}$$

This is the setup for initialisation, requiring up to 40 floating point operations [3].

8.2.1.2 Iteration

The iteration loop is very simple, and is presented explicitly in Algorithm 1.

Algorithm 1: Fast Voxel Traversal—Iteration Loop

Data: $\{t_x, t_y, \Delta_x, \Delta_y, i_x, i_y, m_x, m_y\}$

```
while BreakFlag = 0
    if t_x < t_y                          /* x boundary is crossed first */
        t_x ← t_x + Δ_x ;                          // update x distance
        i_x ← i_x + m_x ;                          // update x index
    else                                  /* y boundary is crossed first */
        t_y ← t_y + Δ_y ;                          // update y distance
        i_y ← i_y + m_y ;                          // update y index
    end
    if BreakCondition                    /* termination subroutine check */
        BreakFlag = 1 ;
    end
end
```

In this implementation, `BreakCondition` acts as a subroutine[6] which determines whether the ray has exited the grid, or the maximum allowable ray distance has been reached. The efficiency of the FVT algorithm stems from the simplicity of this loop, which requires only one floating point comparison, one floating point addition, and an integer addition per step.

[6] `BreakCondition` is equivalent to the subroutine `NextVoxel` in the original paper [3].

8.2.2 Siddon's algorithm

An alternative algorithm for computing the exact path through a square-pixel grid was proposed earlier by Siddon[7] [132], and there have been several refinements which reduce the total computation time [66, 77, 157].

The framework of Siddon's algorithm is built on the concept that square pixels are not independent elements but are instead the volumes resulting from the intersection of sets of parallel lines that are perpendicular to one another[8]. This allows the radiological path to be computed via the total intersection of a ray with separate sets of parallel lines, rather than via a dynamic method which determines intersected pixels in serial.

8.2.2.1 Algorithm details

We assume that a two-dimensional rectangular volume is represented by a basis of $N_x \times N_y$ rectangular pixels, each with an area $dA = dxdy$, and that the radiological path starts and ends outside the volume, at points $P_1 = (x_1, y_1)^\top$ and $P_2 = (x_2, y_2)^\top$, respectively. The path has a total length $L = \|P_2 - P_1\|_2$ and is defined parametrically by

$$P(t) = P_1 + t(P_2 - P_1) \quad \text{for } t \in [0, 1]\,. \tag{8.7}$$

The path first intersects the volume at a distance $t_{\min}$ from P_1 and last intersects the volume at a distance $t_{\max}$ from P_1. Therefore, the radiological path only concerns the contributions of intersected pixels from the sub-path defined by

$$\{P(t) \mid t \in [t_{\min}, t_{\max}]\}. \tag{8.8}$$

The initial index $[\![i_{\min}, j_{\min}]\!]$ and the final index $[\![i_{\max}, j_{\max}]\!]$ are calculated with respect to an origin pixel, indexed by $[\![0, 0]\!]$, with centre $(b_x + dx/2, b_y + dy/2)^\top \in \mathbb{R}^2$.

Pixels are defined as volumes created from two families of equally spaced parallel lines: a set of vertical lines with regular spacing dx and a set of horizontal lines with regular spacing dy. Siddon's algorithm determines the distance to successive parallel lines in the x- and y-directions, with respect to the direction of the path, exploiting the fact that the lines are spaced uniformly. In this way, the algorithm computes two sets

[7] Siddon's original paper contains computations for radiological paths in 3D, which can easily be reduced to only two dimensions.

[8] We are using loose language here—by saying that the two sets $\mathcal{S}_1, \mathcal{S}_2$ are perpendicular, we mean that if $l_1 \in \mathcal{S}_1$ and $l_2 \in \mathcal{S}_2$, then $l_1 \perp l_2$.

of distances: $\{t_i^x\}_{i=1}^{M_x}$ containing the t-parametrised distances in the x-direction, and $\{t_j^y\}_{j=1}^{M_y}$ containing the t-parametrised distances in the y-direction. These distances can be computed directly via

$$t_i^x = \frac{(b_x + (i-1)dx) - x_1}{x_2 - x_1} \quad \text{for} \quad i = 1, 2, \ldots, M_x, \tag{8.9}$$

$$t_j^y = \frac{(b_y + (j-1)dy) - y_1}{y_2 - y_1} \quad \text{for} \quad j = 1, 2, \ldots, M_y, \tag{8.10}$$

and only those in the interval $(0, 1)$ are kept, i.e. only those corresponding to crossings along the sub-path.

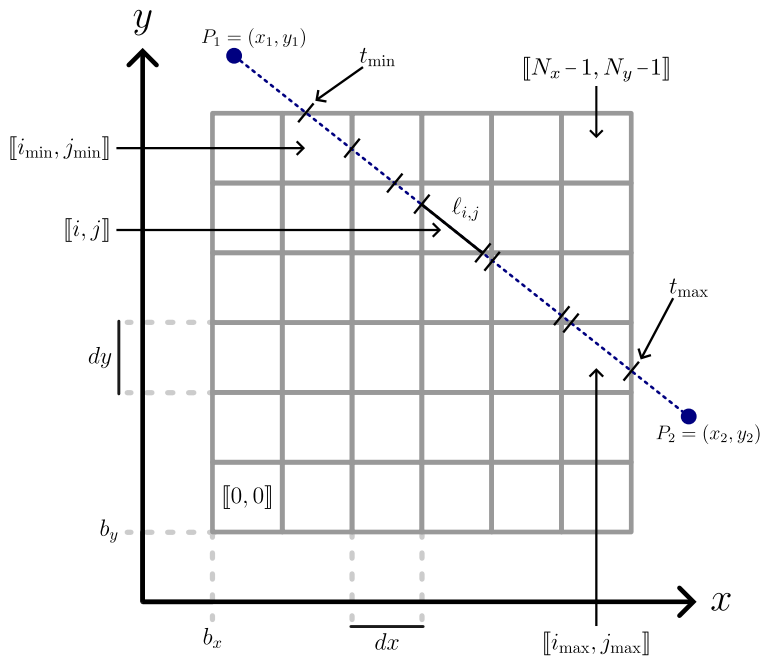


Figure 8.1: The radiological path through a rectangular domain of $N_x \times N_y$ pixels. The path enters pixel $[\![i_{\min}, j_{\min}]\!]$ at a distance $t_{\min}$ from P_1 and exits pixel $[\![i_{\max}, j_{\max}]\!]$ at a distance $t_{\max}$ from P_1. The intersection length of the path with pixel $[\![i, j]\!]$ is denoted by $\ell_{i,j}$.

It is worth distinguishing between two related terms: intersection *distance* refers to the location at which a pixel boundary is crossed by the path, while intersection *length* refers to the measure of overlap between the path and a particular pixel support. The set $\mathcal{T} = \{t_i^x\} \cup \{t_j^y\}$ is then sorted into an ascending ordered[9] set $\{t_1, t_2, \ldots, t_{n-1}\}$, with $t_m < t_{m+1}$, which gives the ordered sequence of distances from P_1 at which

[9]It is technically possible for there to exist an index m for which $t_m = t_{m+1}$. However, this would correspond to a ray passing through a pixel vertex rather than an edge, which, for an arbitrary ray, happens with probability zero.

the path crosses a pixel boundary. Denoting $t_0 := t_{\min}$ and $t_n := t_{\max}$, the total set $\{t_0, t_1, \ldots, t_{n-1}, t_n\}$ specifies all intersection distances that occur, including entry and exit points, so that the intersection lengths are simply

$$\ell_m = (t_m - t_{m-1})L \quad \text{for } m = 1, \ldots, n. \tag{8.11}$$

Additionally, as intersection distances mark the transition from one pixel to the next, the particular ordering of the set $\mathcal{T}$ also returns the ordered indices

$$\{\, [\![i_{\min}, j_{\min}]\!], \ldots, [\![i_{\max}, j_{\max}]\!] \,\}, \tag{8.12}$$

of the intersected voxels, via index shifts analogous to the m_x, m_y increments of the FVT algorithm. Siddon's algorithm therefore returns all indices $[\![i_k, j_k]\!]$ and associated intersection lengths $\ell_{i_k,j_k} := \ell_k$ that constitute the radiological path. In Figure 8.1 we include a diagram showing the location of the intersections, as well as the start and end points of the path.

8.3 Ray traversal through a hexagonal grid

Although there is some literature on hexagonal image sampling for medical imaging, no efficient ray-traversal algorithms appear to have been developed for hexagonal grids in this context, which is what we require to construct our forward operators $\mathbf{A}(u)$. Most work relies on resampling methods on square pixels to approximate line integrals, such as pixel-driven methods [119], bilinear and trilinear interpolation methods [81, 90], and distance-driven methods [35, 36]. Other works instead use re-implementations of existing transforms to define the forward operator [159].

A natural approach to ray marching through a hexagonal grid would be to sequentially step through intersected hexels, keeping track of both the pixel index as well as the computed intersection length. While conceptually simple, this sequential approach is considerably slower than efficient implementations of Siddon's algorithm or FVT for square pixel grids. At each step in the ray-marching algorithm the ray enters a particular edge on a pixel. There are three[10] possible edges that it can exit, and so it is required to compute three inner products and take the minimum value to determine through which edge the ray exits. While inner products are not necessarily

[10]Note that once the direction of the ray is chosen, each pixel can only be entered from three possible edges, and exited from the other three (ignoring pathological vertex-intersection cases).

a computational bottleneck, when compared to the simplicity of marching in the x- or y-direction this approach is far more costly. Therefore, if motion operators are to be reconstructed under a different discretisation, the repeated construction of the system matrix should not become the dominant computational cost in solving (4.5).

The alternative we present is to augment existing traversal algorithms to handle pixel grids for which they were not originally designed. As FVT is the standard algorithm for traversing square pixel grids, we will augment it first. However, as we will later show in this chapter, and further in Chapter 9, Siddon's algorithm holds an underlying flexibility which makes it far better suited for adaptation to non-square, and non-cubic grids.

8.3.1 Hex-FVT: An extension to hexagonal grids

While the original FVT algorithm exploits the regularity of the x- and y-boundaries, it also subtly exploits the linear independence of the two families of grid lines. This then guarantees that each time the ray intersects one of the horizontal or vertical lines, the intersection corresponds to a genuine pixel-edge.

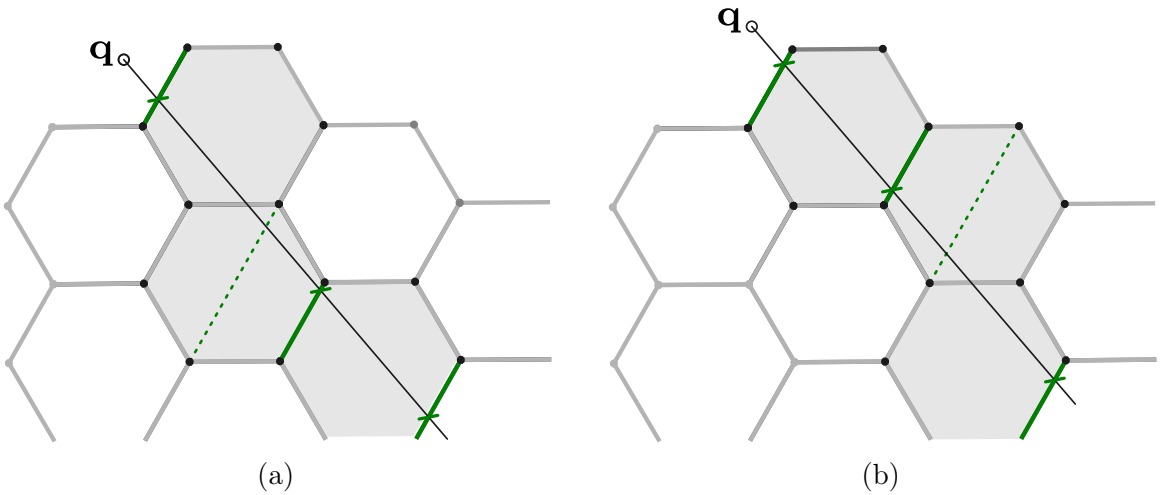


Figure 8.2: The intersection of a ray with a hexagonal edge 'support-line' does not guarantee that it occurs at a genuine hexel edge. Two rays travelling in the same direction may hit distinct sequences of edges, shown here in green—we have included the dashed line to show the extended 'support line', which would be found by naïve implementations of the FVT.

As such, and as is done in Siddon's algorithm, intersections are accepted without logical checks. For hexels this is no longer true—see Figure 8.2—as there may be a

discrepancy between a hexel edge and the line[11] l from which the edge is a subset. To extend the FVT algorithm to hexagonal grids, we therefore require a way to account for this difference, such that the algorithm can distinguish genuine hexel-edge intersections from spurious support-line crossings.

8.3.1.1 Initialisation

Much like the family of axis-aligned lines in the FVT algorithm, we will utilise three sets of lines with angles α, $\beta := \alpha + \pi/3$, and $\gamma := \alpha + 2\pi/3$, denoted by $\mathcal{L}_\alpha, \mathcal{L}_\beta$, and $\mathcal{L}_\gamma$, respectively—see Figure 8.3. The edges of our hexagonal grid will be formed from subsets of these families of lines, such that for flat-top oriented hexagons[12] we have $\alpha = -\pi/3$. For all three families, the spacing between two neighbouring parallel lines is a fixed distance R, equal to the *inradius* of the hexagonal cells.

We assume the existence of a hexagonal grid that can be enumerated by a three-dimensional indexing scheme $[\![i_\alpha, i_\beta, i_\gamma]\!]$, such that crossing a member of $\mathcal{L}_\alpha, \mathcal{L}_\beta$, or $\mathcal{L}_\gamma$ corresponds to the respective index shifts[13]

$$m_\alpha = \pm[\![0, 1, -1]\!], \qquad m_\beta = \pm[\![1, 0, -1]\!], \qquad m_\gamma = \pm[\![1, -1, 0]\!]. \tag{8.13}$$

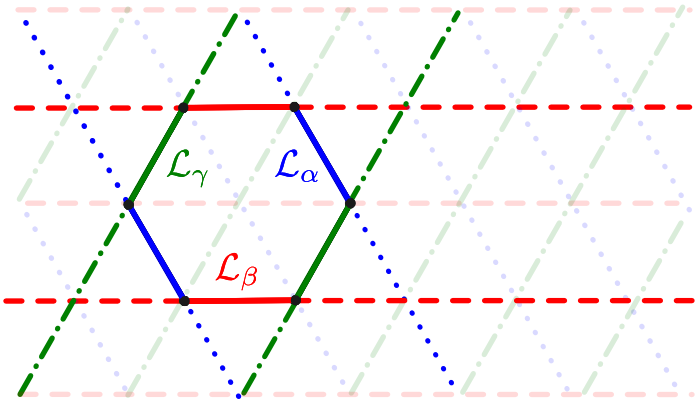


Figure 8.3: Each hexel is bounded by three pairs of parallel lines spaced by a distance $2R$: two from the set $\mathcal{L}_\alpha$ (blue, dotted); two from the set $\mathcal{L}_\beta$ (red, dashed); and two from the set $\mathcal{L}_\gamma$ (green, dot-dashed). In this example $\alpha = -\pi/3$, $\beta = 0$, and $\gamma = \pi/3$.

We also assume that the start of our ray $\mathbf{q} = (q_x, q_y)^\top$, lies within a distance $R/2$ of

[11] We will refer to the particular line as the *support line* of the edge.

[12] The orientation of the hexagonal grid does not affect the algorithm; this is for graphical and explanatory convenience.

[13] This is referred to as a *cubic* indexing scheme, such that for all hexels $i_\alpha + i_\beta + i_\gamma = 0$.

the grid[14], otherwise we may simply project $\mathbf{q}$ along the ray until this condition is satisfied. In effect, $\mathbf{q}$ lies in a hexel that is at most one index shift away from a hexel within the grid.

8.3.1.2 Iteration

In the direction of $\mathbf{v}$ we compute the distance to the nearest line in each family, i.e. the closest members of $\mathcal{L}_\alpha$, $\mathcal{L}_\beta$, and $\mathcal{L}_\gamma$, noting that this does not guarantee that the closest point corresponds to a genuine edge, but merely a point along the associated support line.

We denote the distances from $\mathbf{q}$ to the next line in each family by d_α, d_β and d_γ. By assuming that $\mathbf{q}$ lies close enough to the grid, the smallest of the three distances corresponds to a genuine edge intersection. We create the ordered[15] list

$$(s_1, s_2, s_3) = \texttt{sort}\left(d_\alpha, d_\beta, d_\gamma\right), \tag{8.14}$$

such that the sorted distances are $s_1 \leq s_2 \leq s_3$, where $s_1 = \min\left(d_\alpha, d_\beta, d_\gamma\right)$. The ray forms angles with each family of parallel lines, denoted by ϕ_α, ϕ_β, and ϕ_γ; for example, $\phi_\alpha = \arccos\left(\langle \mathbf{v},\ (\cos(\alpha), \sin(\alpha))^\top \rangle\right)$. We then create the permuted lists

$$(a_1, a_2, a_3) = \texttt{order}\left(\phi_\alpha, \phi_\beta, \phi_\gamma\right), \tag{8.15}$$

$$(w_1, w_2, w_3) = \texttt{order}\left(m_\alpha, m_\beta, m_\gamma\right), \tag{8.16}$$

where the `order` operator arranges elements into the associated ordering that maps $(d_\alpha, d_\beta, d_\gamma)$ to (s_1, s_2, s_3) in (8.14). In this way, the determination of s_1 allows us to also determine the relevant index w_1 at each step, allowing both the intersection length and the updated pixel index to be determined at each genuine edge crossing.

The iteration loop of the original FVT algorithm is very simple, as it requires increasing the shortest available step by a fixed distance—either Δ_x or Δ_y—and leaving the other alone. A key difference when extending FVT to hexagonal grids is that at least two of the three distances must be updated at each step, and possibly all three.

[14] For a square grid the convex hull of the pixels is equivalent to the support of the pixels, and so projecting $\mathbf{q}$ along the ray onto the convex hull guarantees that the projection lies on a pixel edge. For a hexel grid this is no longer the case, but the projection onto the convex hull will be no more than one inradius away from the nearest hexel edge.

[15] As with the square pixel grid, it is possible that this list will only be partially ordered. However, this corresponds to the ray intersecting a hexel vertex, which happens with probability zero.

8.3.1.3 Hex-FVT: Update step

First we update the shortest distance, corresponding to a genuine edge intersection, to the value

$$\varrho_1 = s_1 + 2R\csc(a_1). \tag{8.17}$$

This step is required, as if the ray passes through successive edges from the same family then the closest it can possibly be is the distance between two hexagonal edges along the direction of $\mathbf{v}$—see Figures 8.4 and 8.5.

The second-shortest distance does not correspond to a genuine edge intersection, and so it is updated to

$$\varrho_2 = s_2 + R\csc(a_2), \tag{8.18}$$

as the next possible edge in that family is *at least* a distance $R\csc(a_2)$ away.

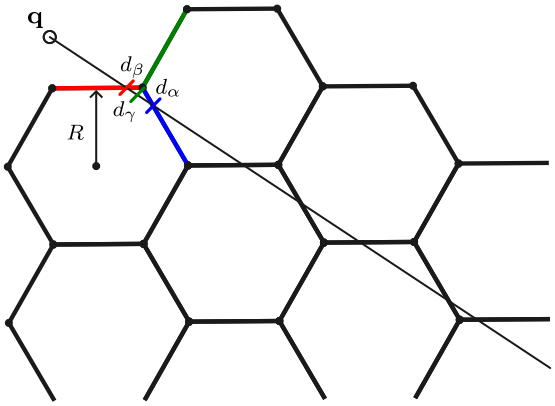
The farthest distance is only updated if either one of two conditions is met. The first condition is to check whether the current farthest distance is already too far from the most recent edge intersection:

$$\varrho_3 = \begin{cases} s_3 - R\csc(a_3) & \text{if } |s_1 - s_3| > R\csc(a_3), \\ s_3 & \text{otherwise.} \end{cases} \tag{8.19}$$

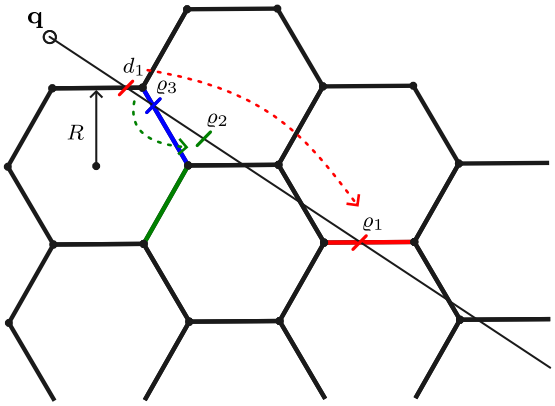
This is needed to determine whether the farthest distance is reachable in the next step of the algorithm. The second condition is to check the relative order of $\{\varrho_1, \varrho_2\}$ compared to $\{s_1, s_2\}$. If (8.19) results in $\varrho_3 = s_3$, then we must also compute

$$\varrho_3 = \begin{cases} s_3 + R\csc(a_3) & \text{if } \varrho_1 < \varrho_2 \\ s_3 & \text{otherwise.} \end{cases} \tag{8.20}$$

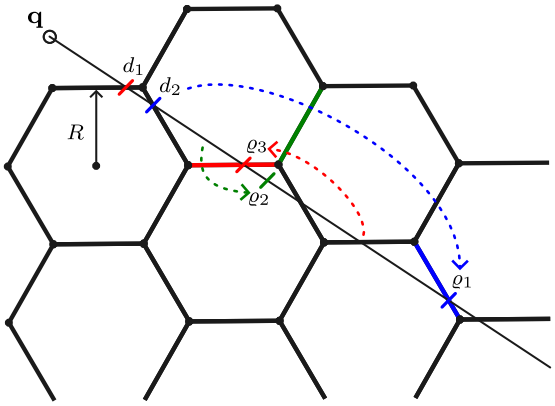
The updated list $(\varrho_1, \varrho_2, \varrho_3)$ then holds the unordered distances to the next-intersected lines for the three families—that is, it is analogous to the original distances $(d_\alpha, d_\beta, d_\gamma)$—and the algorithm proceeds to the next step. To create a looping iteration in the algorithm, the only difference is that the ordering in (8.15, 8.16) will continue over the lists (a_1, a_2, a_3) and (w_1, w_2, w_3), so that the continual updating is tracked at each step.



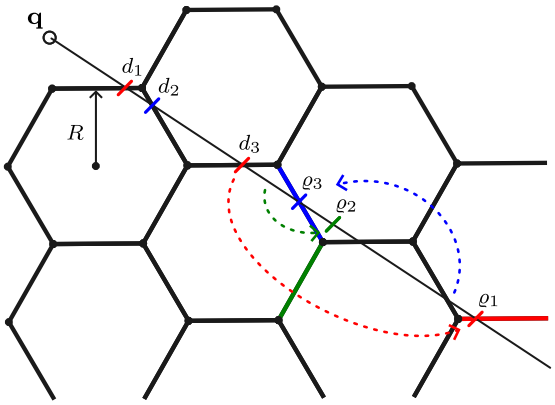
(a) **Step 1.** Distances from $\mathbf{q}$ to the nearest lines in $\mathcal{L}_\alpha, \mathcal{L}_\beta$, and $\mathcal{L}_\gamma$ are denoted by d_α, d_β, and d_γ. The ordered distances are $(s_1, s_2, s_3) = (d_\beta, d_\gamma, d_\alpha)$, and so the shortest distance is d_β, which will be saved as d_1.



(b) **Step 2.** $s_1 \mapsto \varrho_1$ by $2R\csc(\phi_\beta)$, $s_2 \mapsto \varrho_2$ by $R\csc(\phi_\gamma)$, and $s_3 \mapsto \varrho_3$ with no change. The ordered distances are $(s_1, s_2, s_3) = (\varrho_3, \varrho_2, \varrho_1)$, and so the shortest distance is ϱ_3, which will be saved as d_2.

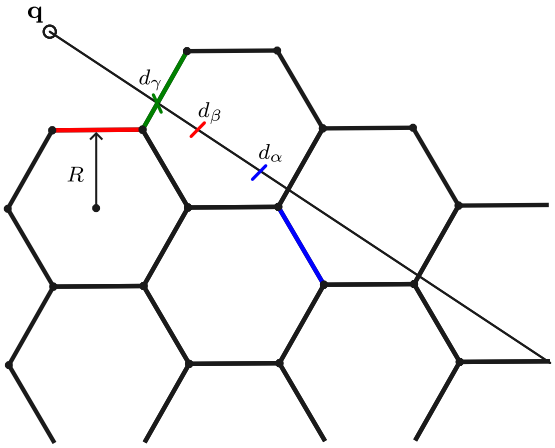


(c) **Step 3.** $s_1 \mapsto \varrho_1$ by $2R\csc(\phi_\alpha)$, $s_2 \mapsto \varrho_2$ by $R\csc(\phi_\gamma)$, and $s_3 \mapsto \varrho_3$ by $-R\csc(\phi_\beta)$. The ordered distances are $(s_1, s_2, s_3) = (\varrho_3, \varrho_2, \varrho_1)$, and so the shortest distance is ϱ_3, which will be saved as d_3.

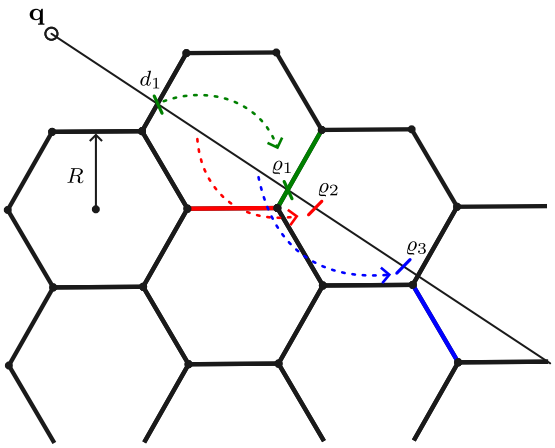


(d) **Step 4.** $s_1 \mapsto \varrho_1$ by $2R\csc(\phi_\beta)$, $s_2 \mapsto \varrho_2$ by $R\csc(\phi_\gamma)$, and $s_3 \mapsto \varrho_3$ by $-R\csc(\phi_\alpha)$. The ordered distances are $(s_1, s_2, s_3) = (\varrho_3, \varrho_2, \varrho_1)$, and so the shortest distance is ϱ_3, which will be saved as d_4.

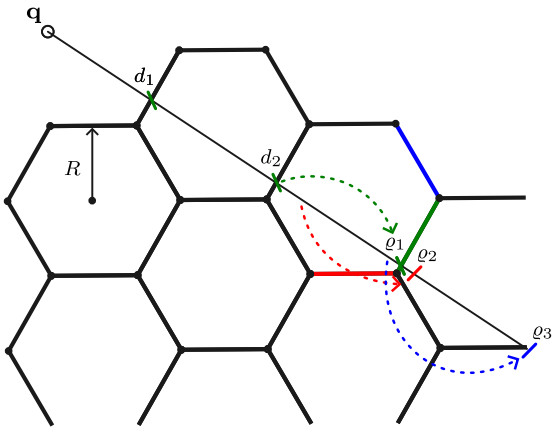
Figure 8.4: **Example 1: ($\varrho_1 > \varrho_2$), repeated steps.** In this example, the lines from $\mathcal{L}_\gamma$ are always the second-shortest distance away from the current iterate. As such, they are repeatedly pushed back, and never intersected. We denote the sequence of genuine intersection distances by $d_1, d_2, d_3, d_4, \ldots$, distinct from the family distances $(d_\alpha, d_\beta, d_\gamma)$, with which we start the traversal.



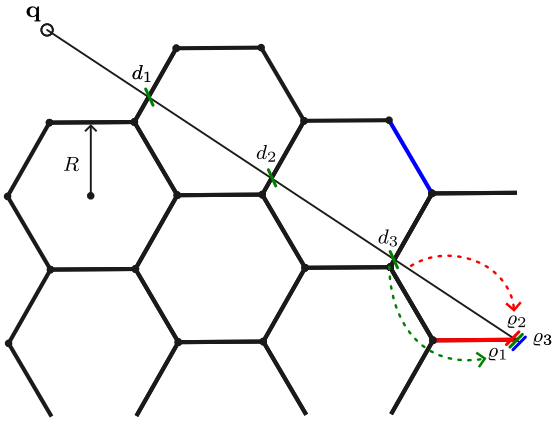
(a) **Step 1.** Distances from $\mathbf{q}$ to the nearest lines in $\mathcal{L}_\alpha, \mathcal{L}_\beta$, and $\mathcal{L}_\gamma$ are denoted by d_α, d_β, and d_γ. The ordered distances are $(s_1, s_2, s_3) = (d_\gamma, d_\beta, d_\alpha)$, and so the shortest distance is d_γ, which will be saved as d_1.



(b) **Step 2.** $s_1 \mapsto \varrho_1$ by $2R\csc(\phi_\gamma)$ and $s_2 \mapsto \varrho_2$ by $R\csc(\phi_\beta)$. As $\varrho_1 < \varrho_2$, $s_3 \mapsto \varrho_3$ by $R\csc(\phi_\alpha)$. The ordered distances are $(s_1, s_2, s_3) = (\varrho_1, \varrho_2, \varrho_3)$, and so the shortest distance is ϱ_1, which will be saved as d_2.



(c) **Step 3.** $s_1 \mapsto \varrho_1$ by $2R\csc(\phi_\gamma)$ and $s_2 \mapsto \varrho_2$ by $R\csc(\phi_\beta)$. As $\varrho_1 < \varrho_2$, $s_3 \mapsto \varrho_3$ by $R\csc(\phi_\alpha)$. The ordered distances are $(s_1, s_2, s_3) = (\varrho_1, \varrho_2, \varrho_3)$, and so the shortest distance is ϱ_1, which will be saved as d_3.



(d) **Step 4.** $s_1 \mapsto \varrho_1$ by $2R\csc(\phi_\gamma)$ and $s_2 \mapsto \varrho_2$ by $R\csc(\phi_\beta)$. As $\varrho_1 > \varrho_2$, $s_3 \mapsto \varrho_3$ with no change. The ordered distances are $(s_1, s_2, s_3) = (\varrho_2, \varrho_1, \varrho_3)$, and so the shortest distance is ϱ_2, which will be saved as d_4.

Figure 8.5: **Example 2: ($\varrho_1 < \varrho_2$), repeated steps.** In this example, the ordering of the distances to lines in $\mathcal{L}_\alpha$ and $\mathcal{L}_\beta$ switches, meaning that the lines from $\mathcal{L}_\gamma$ are repeatedly intersected. Again, we denote the sequence of genuine intersection distances by $d_1, d_2, d_3, d_4, \ldots$, distinct from the original distances $(d_\alpha, d_\beta, d_\gamma)$. We note that the completion of Step 4 results in Step 1 of Example 1, in Figure 8.4.

Similar to the Fast Voxel Traversal algorithm, this approach will then also have a break condition for reaching the end of the ray, or leaving the hexagonal grid, which can be implemented by simply checking whether the remaining distance is less than a certain tolerance.

8.3.2 Pseudocode and numerical results

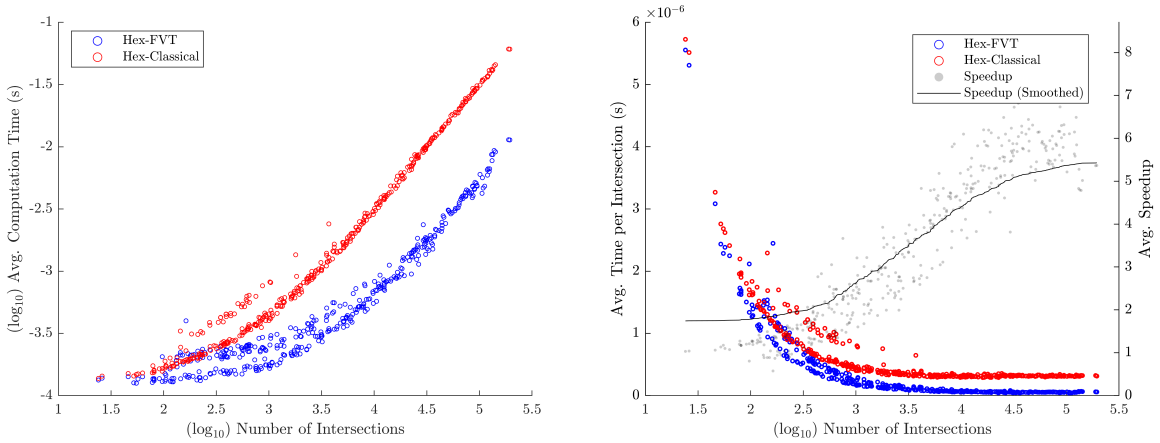
The inner loop of the Hex-FVT algorithm is presented in Algorithm 2, where we use the convention of overwriting variables, and for compactness we use (h_1, h_2, h_3) to initially denote $(R\csc(\phi_\alpha), R\csc(\phi_\beta), R\csc(\phi_\gamma))$. One cycle of this loop costs 4–5 floating point comparisons and 4–5 floating point additions, which is less than that of standard ray-marching algorithms for traversing hexagonal grids.

Algorithm 2: Hex-FVT—Iteration Loop

```
   INITIALISATION:
 1 (ϱ1, ϱ2, ϱ3) ← (dα, dβ, dγ) ;                    // intersection distances
 2 (h1, h2, h3) ;                                    // intra-family steps
 3 (w1, w2, w3) ← (mα, mβ, mγ) ;                     // index shifts
 4 ι ← ⟦iα, iβ, iγ⟧ ;                                // index tracker
   ITERATION:
 5 while BreakFlag = 0
 6 |   (s1, s2, s3) ← sort (ϱ1, ϱ2, ϱ3) ;                  // sort distances
 7 |   (h1, h2, h3) ← order (h1, h2, h3) ;                 // reorder steps
 8 |   (w1, w2, w3) ← order (w1, w2, w3) ;                 // reorder shifts
 9 |   ι ← ι + w1 ;                                        // update index
10 |   append(s1) ;                                  // save distance to list
11 |   append(ι) ;                                   // save index to list
12 |   if |s1 − s3| > h3                        /* feasible distance check */
13 |   |   (ϱ1, ϱ2, ϱ3) ← (s1 + 2h1, s2 + h2, s3 − h3);
14 |   else
15 |   |   if ϱ1 < ϱ2                              /* altered order check */
16 |   |   |   (ϱ1, ϱ2, ϱ3) ← (s1 + 2h1, s2 + h2, s3 + h3);
17 |   |   else
18 |   |   |   (ϱ1, ϱ2, ϱ3) ← (s1 + 2h1, s2 + h2, s3);
19 |   |   end
20 |   end
21 |   if BreakCondition          /* termination subroutine check */
22 |   |   BreakFlag = 1 ;
23 |   end
24 end
```

The original Fast Voxel Traversal algorithm is able to compute both pixel-ray intersection lengths as well as pixel indices, although there are cases where only one of these is required. For example, if one wants to implement a version of Bresenham's algorithm [14] for drawing line segments on hexagonal domains, then only the intersected pixel indices are needed. Conversely, it may be important to accurately calculate a particular distance through a domain, without requiring information about the specific pixels that were intersected. This optionality is captured in our algorithm, as the `append`$(\cdot)$ steps in Algorithm 2 can be omitted if only pixel indices or only intersection lengths are required.

In Figure 8.6 we show the computational speed-up compared to an implementation of a standard ray-marching algorithm which requires edge-ray projections to be computed at each step of the algorithm. For clarity, we use the suffix 'Classical' to refer to ray-marching algorithms that perform sequential edge-projections and minimiser computations—this suffix will be used throughout this chapter and in Chapter 9.



(a) Avg. computation time, per ray.

(b) Avg. time, per intersection.

Figure 8.6: (a) The computational cost of Hex-FVT is linear in the number of intersections, which itself is ultimately linear in the dimension of the hexagonal grid. (b) Average intersection costs of Hex-FVT compared to classical marching methods can attain a computational speed-up of up to 7×, depending on the hexel grid resolution.

Both 'Hex-FVT' and 'Hex-Classical' have been written in MATLAB 2023A and run on a commercially available laptop (Surface Book 3, Intel i7-1065G7 CPU, 32GB RAM, Windows 11 Pro). Neither implementation has been fully optimised, although simplifications were made in the classical ray-marching algorithm to ensure that there was no unreasonable inflation of computational cost. For example, edge-ray intersection locations are obtained with direct computation rather than repeatedly

building a 2×2 system to then invert, and repeated computations are calculated outside of loops where possible. As our results show, our implementation of Hex-FVT is faster than the classical approach when rays intersect more than 10^2 pixels, and the possible speed-up is $\sim$5–7$\times$ faster when rays intersect more than 10^4 pixels.

In the context of constructing forward operators, we have previously shown no examples where a single ray intersects $\mathcal{O}(10^4)$ pixels. However, these results suggest that if the ray setup costs could be grouped—as is generally done when building the matrix $\mathbf{A}(u)$—then the speed benefit of using Hex-FVT to build the operator would be clearly apparent.

8.3.2.1 Additional numerical remarks

In Hex-FVT, distance updates require only integer multiples of elements of (h_1, h_2, h_3). As such, the inner loop may be implemented 'symbolically' in the sense that explicit floating-point distances are not required until termination: we can instead propagate integer coefficients and evaluate distances numerically only when comparisons are needed. This would further reduce the number of floating point operations and may reduce rounding error accumulation, since final distances are computed by scalar multiplication rather than accumulated through long chains of additions. The main practical caveat is that frequent recasting to numeric values can introduce nontrivial overhead if performed inside the inner loop[16].

For practical purposes, our implementation is purely numeric and the observed accumulation error grows in a controlled manner with the number of intersections. Empirically, the average per-intersection error increases slowly with n, the total number of intersections, consistent with floating-point error accumulation for an algorithm which sums a sequence of lengths [73].

8.3.2.2 Numerical robustness

If we compare Hex-FVT to the classical FVT algorithm through a square grid, we find that the overall computational expense is not that much more—see Figure 8.7.

[16]Rather than storing a symbolic copy of both (s_1, s_2, s_3) and (w_1, w_2, w_3), a lightweight alternative is to store integer coefficients alongside a numeric running copy used for comparisons, to then numerically compute the algebraic representation only once at the end. This could also be implemented as a ternary code, indicating which component of (s_1, s_2, s_3) was selected at each step. An additional consideration is that the `order`$(\cdot)$ operations are not inherently vital—it is necessary to determine the smallest element of (s_1, s_2, s_3) but the reordering in Lines 7–8 of Algorithm 2 is primarily for bookkeeping. Therefore, any reimplementation may dispense with reordering and instead keep track of the indices which represent 'smallest' and 'largest' distances at the current iteration.

Results show that for large intersection numbers ($n \geq 10^5$), performing ray traversal over hexels is approximately 1.2× more expensive than traversal over square pixels; that is, the cost ratio between the two methods is roughly constant. In practice this is what we observe, as the (wall-clock) time needed to construct the forward operator for hexels is roughly the same as when using square pixels. This is likely due to the fact that constructing the matrix requires multiple overhead costs that can dominate, and so the per-intersection cost is amortised more effectively across all rays.

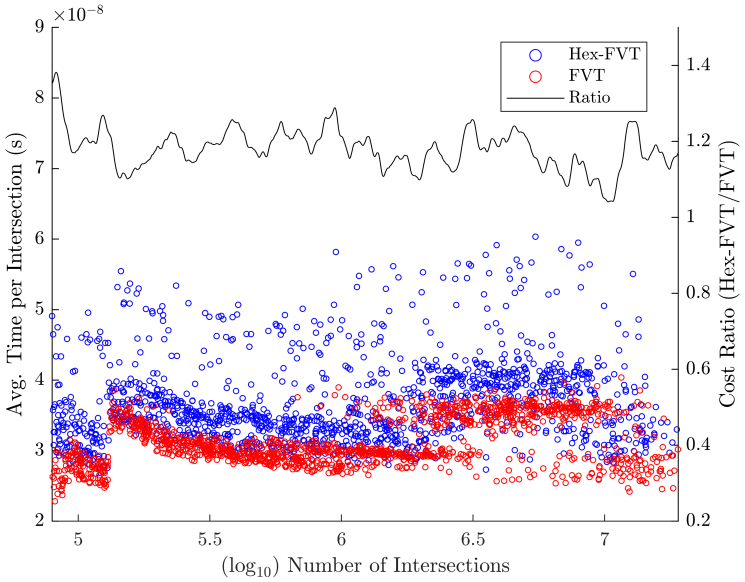
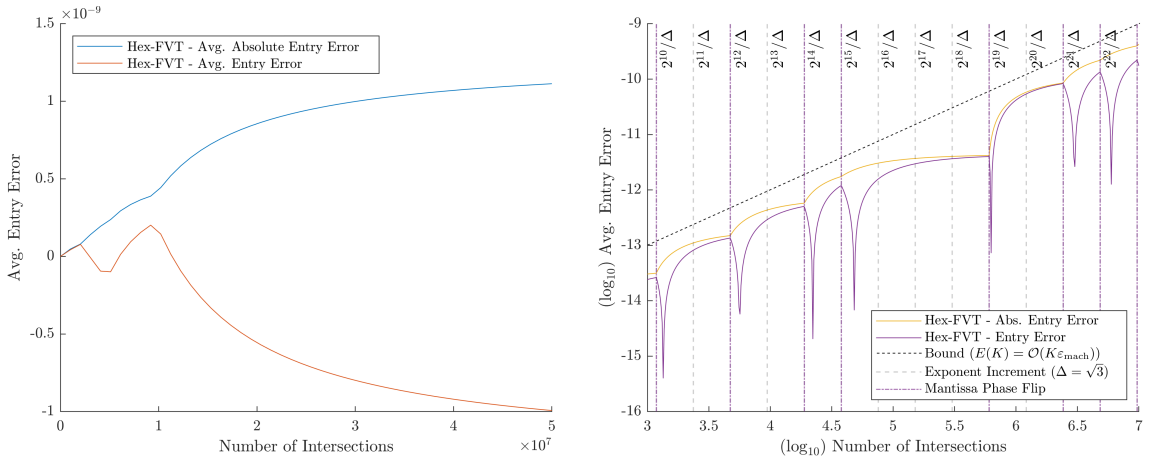


Figure 8.7: Average costs per intersection between hexagonal grids and square grids are comparable using Hex-FVT as the hexagonal intersection algorithm. The tighter variance of the original FVT algorithm is likely due to the behaviour of rays through square grids, which have a computable *a priori* number of intersections.

As an alternative test of robustness we analyse the errors incurred when computing the intersections through the centres of a sequence of vertically-aligned hexels, such that the intersection length through each pixel is $\Delta = \sqrt{3}$, i.e. the inradius is $R = \sqrt{3}/2$. As we show in Figure 8.8b, the average error of any particular intersection is small enough to maintain expected first-order accumulation errors; that is, if n is the total number of intersections computed then the absolute error grows proportionally to $\mathcal{O}(n\varepsilon_{\text{mach}})$, where $\varepsilon_{\text{mach}}$ denotes machine precision.

Interestingly, an additional feature of Hex-FVT is that the signed error has a parity bias depending on the length of the ray such that it is possible to identify parity flips in the mantissa of the floating point representation of the total intersection length. As we show in Figure 8.8a, the signed error oscillates while the absolute error grows

monotonically, indicating that error rounding can cause partial cancellation at specific scales.



(a) (b)

Figure 8.8: (a) The average error for a calculated intersection length grows in absolute value for longer rays. However, over time the signed error exhibits clear oscillations aligned with powers-of-two in the accumulated intersection count. This behaviour is consistent with mantissa parity changes induced by exponent increments in floating-point arithmetic. (b) The sign switches occur when the total length of the ray is of the form $n = 2^{\omega}/\Delta$.

The purple line in Figure 8.8b shows the existence of 'fangs', with each indicating a sign change in the average error of Hex-FVT intersection computations. As we show, the locations at which these switches occur is at path lengths n for which $n\Delta = 2^{\omega}$ for some $\omega \in \mathbb{N}$. The non-uniformity of the sign switches shows that the condition $n\Delta = 2^{\omega}$ is necessary, but not sufficient.

8.3.2.3 Applicability of extending Hex-FVT to 3D

The FVT algorithm can be immediately extended to three dimensions—this is shown in the original paper [3]—but this is not the case for Hex-FVT.

The three-dimensional analogue of hexels—rhombic dodecahedral voxels—are formed from six pairs of parallel faces and so the logic needed to determine the order of intersected faces is more involved than for hexels. For hexagonal traversal, when constructing the set of distances (s_1, s_2, s_3), we know that the line associated with s_2 cannot be intersected next. As such, at each step there are only two possible hexagonal edges which can be intersected.

In the case of rhombic dodecahedra, the voxel elements result from the intersection of six pairs of parallel planes, and the logic holds that if we order the intersection distances, $(s_1, s_2, s_3, s_4, s_5, s_6)$, then the second distance cannot correspond to a genuine intersection. However, depending on the relative orientation of the ray direction $\mathbf{v}$ to the face normals, the remaining distances, (s_3, s_4, s_5, s_6), may all be valid, and can occur in any order. Therefore, the inner loop of a three-dimensional extension of Hex-FVT would require a substantially more complex branching structure that would likely dominate the per-step cost. However, as mentioned previously, Siddon's algorithm offers considerably more flexibility for non-standard extensions and so we will discuss this next.

8.4 Extending Siddon's algorithm to hexagonal pixels

As in Section 8.3.1, we first assume that there exists a two-dimensional domain which we can discretise using non-overlapping hexels. We wish to compute the radiological path from a point $P_1 = (x_1, y_1)^\top$ to a point $P_2 = (x_2, y_2)^\top$, such that the path has a total length $L = \|P_2 - P_1\|_2$ and is defined parametrically by

$$P(t) = P_1 + \frac{t}{L}(P_2 - P_1) \quad \text{for} \quad t \in [0, L]\,. \tag{8.21}$$

Throughout this extension we will use t to denote physical distance along the ray rather than the proportional parameter $t \in [0, 1]$ used in the original Siddon formulation [132]. This is for clarity, rather than for any computational efficiency, as the algorithm can be scaled trivially in any implementation.

Adopting the approach of the original paper, we do not treat each hexel as an indivisible unit, but instead take them as volumes resulting from the intersection of three sets of parallel lines. To describe any particular pixel in the domain, we sample pairs of lines from three distinct sets with the following properties:

1. Each set contains (countably) infinitely many uniformly spaced parallel lines;
2. Each set has the same spacing between neighbouring lines;
3. Each pair of sets is separated by a rotation of $2\pi/3$ radians.

In this way, we form hexels from the same sets of lines described in Section 8.3.1, and we will continue to denote these sets by $\mathcal{L}_\alpha$, $\mathcal{L}_\beta$, and $\mathcal{L}_\gamma$, respectively, for $\alpha = -\pi/3$, $\beta = 0$, and $\gamma = \pi/3$.

For simplicity, we will assume that the hexagonal grid has a similar structure[17] to that shown in Figure 8.9, and we will refer to it as having a size of $N_x \times N_y$ hexels. Under this description, N_y is the number of vertically aligned pixels in the left-most column and N_x is the number of distinct columns; for example, the grid shown in Figure 8.9 has a size of 5×4. This convention is merely used to define the total number of hexels in the grid and is not directly related to the indexing $[\![i, j, k]\!]$.

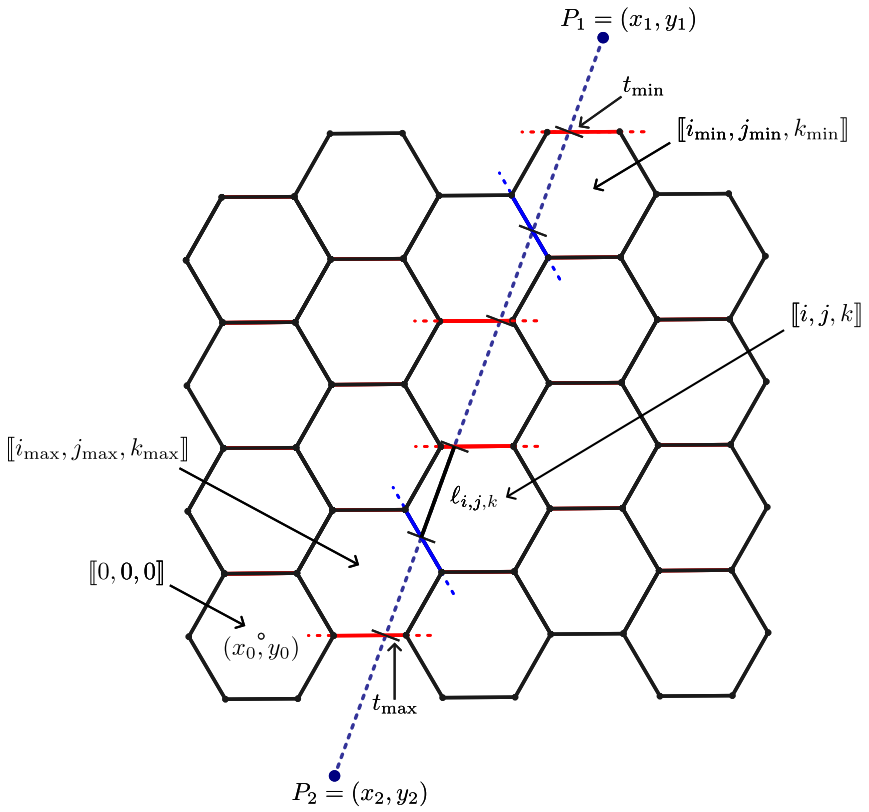


Figure 8.9: The path will intersect a sequence of hexel edges that are subsets of members in the sets $\mathcal{L}_\alpha$, $\mathcal{L}_\beta$, and $\mathcal{L}_\gamma$. The example shown here only intersects members of $\mathcal{L}_\alpha$ (blue) and $\mathcal{L}_\beta$ (red).

The lower left pixel has index $[\![0, 0, 0]\!]$ and centre $\mathbf{r}_0 = (x_0, y_0)^\top \in \mathbb{R}^2$, and the path first intersects pixel $[\![i_{\min}, j_{\min}, k_{\min}]\!]$ and last intersects pixel $[\![i_{\max}, j_{\max}, k_{\max}]\!]$. In Figure 8.9 we show how a path will intersect a set of hexel edges that are subsets of lines from the sets $\mathcal{L}_\alpha$, $\mathcal{L}_\beta$, and $\mathcal{L}_\gamma$.

As mentioned previously, a standard approach to computing successive intersection lengths is to use a ray-marching algorithm that sequentially determines which hexel edges are intersected in order, terminating once the path end point P_2 is reached. The

[17] If the grid did not have this particular shape then such a grid can always be extended to satisfy this property.

result is then a sequence of intersection distances that can be used to give a sequence of non-uniform intersection lengths, $\ell_{i,j,k}$. Although this is a very simple method to implement, there is redundant computation at each step, since determining which edge is intersected requires comparisons with all neighbouring edges.

Rather than computing a sequence of non-uniform distances—as both classical ray-marching methods and Hex-FVT do—we instead choose to compute a sequence of uniform distances for the sets $\mathcal{L}_\alpha$, $\mathcal{L}_\beta$, and $\mathcal{L}_\gamma$, much like the original Siddon's algorithm computes uniform distances in the x- and y-directions. We denote the length[18] of the hexel edge by w, so that the lines in each set are equally separated by the inradius distance $R = \sqrt{3}w/2$. Without loss of generality, we may assume that both P_1 and P_2 are each within a distance $R/2$ of the hexel grid.

As an example[19], in Figure 8.10 we show how the path $P(t)$ will intersect an ordered sequence of members of the set $\mathcal{L}_\beta$. Here, all intersections are equally spaced by a distance $R\csc\phi$, where ϕ is the angle the ray makes with all lines in $\mathcal{L}_\beta$. With this approach, the problem of computing which edges are successively intersected transforms into computing which intersections with lines from $\mathcal{L}_\beta$ correspond to a true hexel boundary in Figure 8.9. Similar to the discussion of Hex-FVT, intersections that correspond to a true hexel edge will be called *genuine*, whereas intersections that do not will be called *spurious*. Similar computations can be performed independently for the sets $\mathcal{L}_\alpha$ and $\mathcal{L}_\gamma$, resulting in three ordered sets of intersection distances similar to the t-parametrised sets of Siddon's algorithm.

8.4.1 Determining genuine intersections

For the set $\mathcal{L}_\beta$, we denote the first intersection distance by t_0^β, so that the ray intersects the line Π_M (for some M to be determined) at the location $P(t_0^\beta)$. The next distance is t_1^β, associated with the location[20] $P(t_1^\beta) \in \Pi_{M\pm1}$, and all subsequent distances are simply

$$t_k^\beta = t_0^\beta + kR\csc(\phi), \tag{8.22}$$

[18] It will become clear that this length is a more natural variable with which to associate our new approach.

[19] The example presented through Figures 8.10 to 8.12 looks at only 10 members of $\mathcal{L}_\beta$—this is for illustrative purposes, as the number of members depends on the size of the hexel grid and the positions of P_1 and P_2.

[20] While the example shows $P(t_1^\beta) \in \Pi_{M-1}$, for $M = 8$, this simple relationship is not always true. The relationship between the subscript of t_k^β and the intersecting plane Π_M depends on the relative alignment of the vectors $(P_1 - P_2)$ and $(P_1 - \mathbf{r}_0)$. This will be discussed in detail in Section 8.4.2.

for intersection locations $P(t_k^\beta) \in \Pi_{M\pm k}$. We project each $P(t_k^\beta)$ onto the line Π_0, to a distance s_k from $\mathbf{r}_0$, such that

$$s_k = s_0 - ks, \tag{8.23}$$

where $s_0 = \langle (1,0)^\top, (P(t_0^\beta) - \mathbf{r}_0) \rangle$ is the projected distance of the first intersection $P(t_0^\beta)$ and $s = R\cot(\phi)$ is as shown in Figure 8.10.

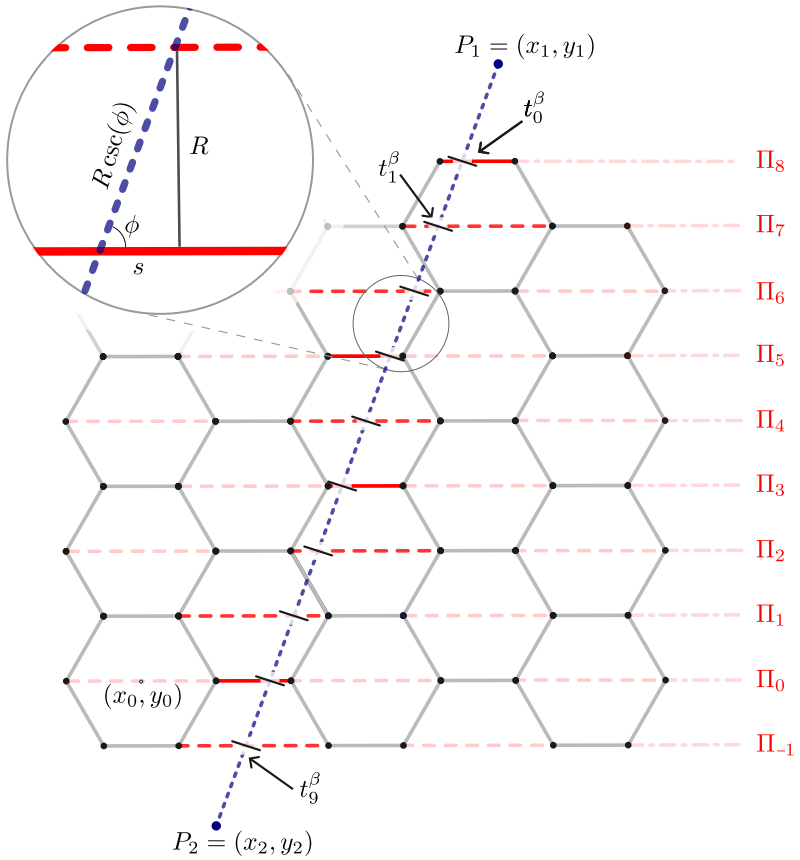


Figure 8.10: The path intersects a sequence of lines from the set $\mathcal{L}_\beta$, at distances $t_k^\beta = t_0^\beta + kR\csc(\phi)$ from the point P_1. In this example, the genuine intersections occur with the lines Π_8, Π_5, Π_3, and Π_0 (in order) which we denote by the ray intersecting a full line, not a dashed line. Comparing distances to those in Figure 8.9, we see that $t_0^\beta = t_{\min}$, but $t_9^\beta > t_{\max}$, meaning that it cannot be genuine.

The pattern of genuine versus spurious intersections in any line Π_K is completely determined by the parity of K—for example, Π_4 and Π_6 in Figure 8.11 have the same horizontal periodicity of dashed lines, while Π_5 does not. As such, we may treat the projection of the set $\{P(t_k^\beta) \in \Pi_K \mid K \pmod 2 = 0\}$ as being represented by an 'even line' Π_{even}, and the remaining $\{P(t_k^\beta) \in \Pi_K \mid K \pmod 2 = 1\}$ being represented by an 'odd line' Π_{odd}.

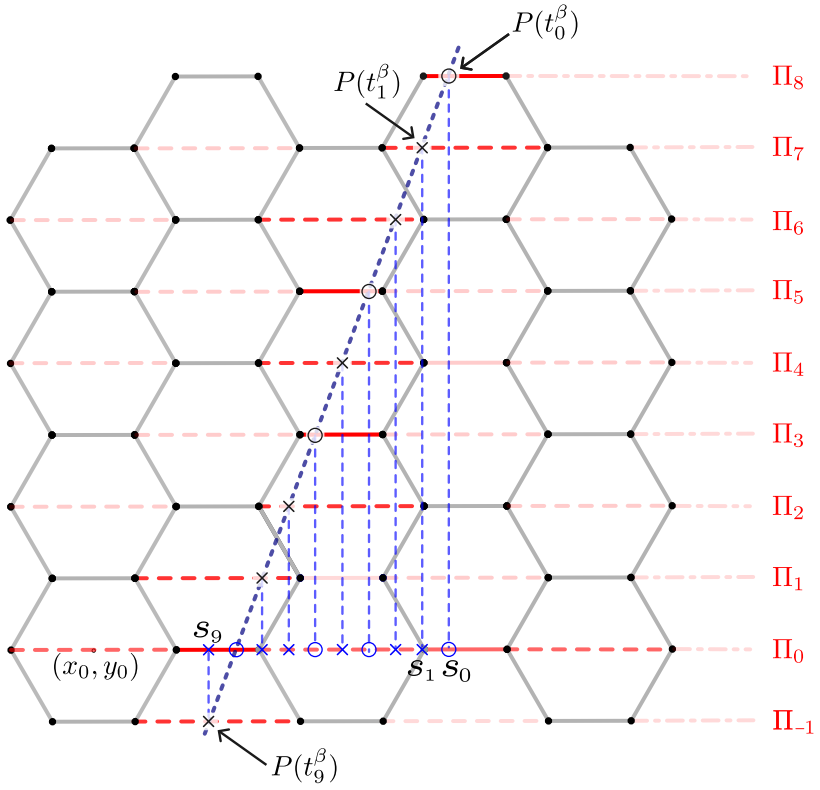


Figure 8.11: The intersection locations $P(t_k^\beta)$ are projected onto Π_0, where we have used circles to denote a genuine intersection and crosses to denote a spurious intersection.

As we show in Figure 8.12, projecting the locations based on the parity of the line subscript makes it clear that an intersection is genuine if its projection falls in a specific interval with respect to $\mathbf{r}_0$. Determining whether an intersection $P(t_k^\beta) \in \Pi_M$ is genuine is equivalent to computing

$$\sigma_k = \begin{cases} (s_k + w/2) \pmod{3w} & \text{if} \quad M \pmod 2 = 1, \\ (s_k - w) \pmod{3w} & \text{if} \quad M \pmod 2 = 0, \end{cases} \tag{8.24}$$

and then checking if $\sigma_k \in [0, w]$.

To be more concise, we combine the cases in (8.24) into

$$\sigma_k^\star = \left(s_k - w + \frac{3w}{2}\Big(M \pmod 2\Big)\right) \pmod{3w}. \tag{8.25}$$

The logical check is then

$$\sigma_k = \begin{cases} 1 & \text{if } \sigma_k^\star \in [0, w]\,, \\ 0 & \text{otherwise,} \end{cases} \tag{8.26}$$

where a 1 corresponds to a genuine intersection, and a 0 corresponds to a spurious intersection. In practice, the value of σ_k is later implemented as a multiplier for t_k^β so

that all spurious distances are zeroed out, leaving only the set of genuine distances, which we denote by $\{\, t_k^\beta \,\}_{\text{gen}}$.

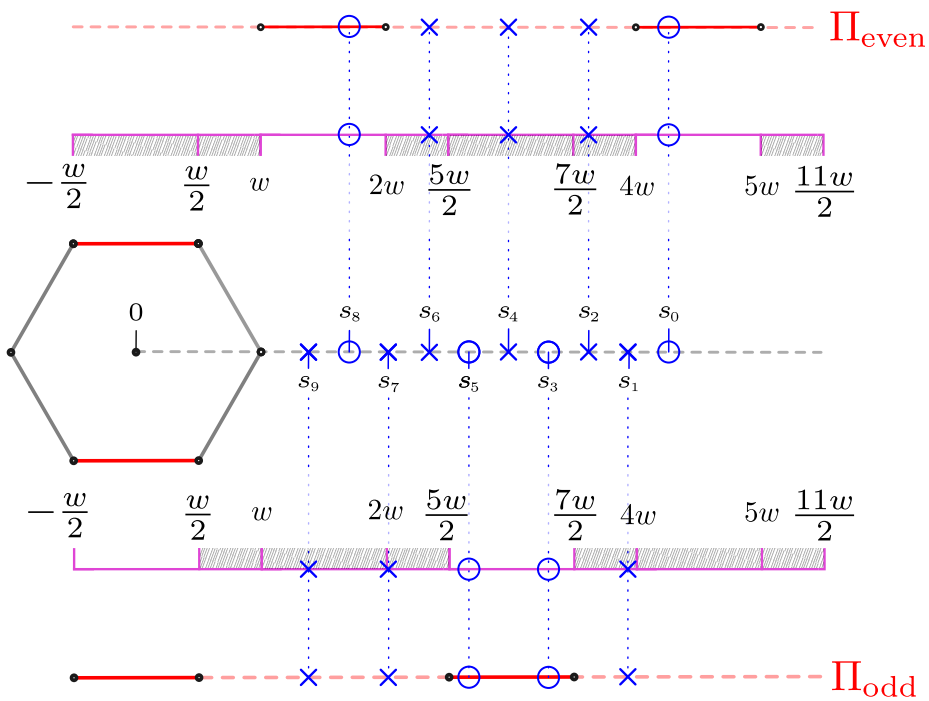


Figure 8.12: Whether $P(t_k^\beta)$ is genuine or spurious can be determined by the location of its projection onto either Π_{odd}, or Π_{even}, where the choice of 'odd' or 'even' depends on the parity of M for which $P(t_k^\beta) \in \Pi_M$. Genuine intersections that are 'even' fall in the interval $[w, 2w]$ (modulo $3w$), while genuine intersections that are 'odd' fall in the interval $[-w/2, w/2]$ (modulo $3w$). In this figure, any projections that pass through the grey boxes are spurious and all others are genuine.

In this way, our algorithm is an extension of the original algorithm of Siddon. In the original, the two sets of lines are orthogonal—any line from one set is perpendicular to all lines from the other set. Because of this, all intersections that occur are genuine[21], and there is no possibility for a spurious intersection to arise. In the case of hexels, the alignment of the sets cannot guarantee that all intersections are genuine and so the logical check in (8.26) *must* be carried out.

The process described above can also be carried out independently for the sets $\mathcal{L}_\alpha$ and $\mathcal{L}_\gamma$, forming the total set of real intersection distances

$$\mathcal{R} = \{\, t_i^\alpha \,\}_{\text{gen}} \cup \{\, t_j^\beta \,\}_{\text{gen}} \cup \{\, t_k^\gamma \,\}_{\text{gen}}. \tag{8.27}$$

[21]More precisely, it is the linear independence of the two families of lines, rather than their orthogonality, that guarantees all intersections are genuine. Orthogonality happens to hold for square grids but is not the essential property—any two linearly independent families of parallel lines would have the same property.

This set can then be sorted into an ordered t-parametrised set

$$\mathcal{R}_{\text{sort}} = \{t_0, t_1, \ldots, t_k, t_{k+1}\}, \tag{8.28}$$

containing the ordered sequence of distances at which genuine hexel boundaries are crossed, so that the intersection lengths are simply

$$\ell_m = (t_m - t_{m-1}) \qquad \text{for} \quad m = 1, \ldots, k+1. \tag{8.29}$$

Naïvely taking the union of the sets in (8.27) and then ordering the elements has a time complexity of $\mathcal{O}(k \log(k))$. However, the sets $\{ t_i^\alpha \}_{\text{gen}}$, $\{ t_j^\beta \}_{\text{gen}}$ and $\{ t_k^\gamma \}_{\text{gen}}$ are already sorted, which can be exploited. A collection of K sorted lists with a combined total of n entries can be merged into a single sorted list using *K-way merging*, which has a time complexity of $\mathcal{O}(n \log(K))$. Therefore, forming the set $\mathcal{R}_{\text{sort}}$ has a time complexity of $\mathcal{O}(\log(3)k) = \mathcal{O}(k)$, and is linear in the number of intersections that a ray makes with the hexel grid.

An additional component of Siddon's algorithm is the ability to compute the ordered indexing list of pixels that are intersected by the path. This is also possible for hexels and is described at the end of Section 8.4.2, where we implement the same indexing system as for the Hex-FVT algorithm.

8.4.2 Hex-Siddon: Algorithmic details

Here we present the primary computations of the algorithm, grouping them into natural 'stages'. Without loss of generality, we will set $\beta = 0$, $\alpha = -\pi/3$, and $\gamma = \pi/3$.

Stage 0: Initialisation

For a grid of $N_x \times N_y$ hexels, first calculate the smallest axis-aligned bounding box $[X_{\min}, X_{\max}] \times [Y_{\min}, Y_{\max}]$ that bounds the grid. The lower left hexel has the index $[\![0, 0, 0]\!]$ and its centre will be at the point $\mathbf{r}_0 = (x_0, y_0)^\top \in \mathbb{R}^2$. The points P_1 and P_2 are connected by a path with gradient $G = (y_2 - y_1)/(x_2 - x_1)$ and direction $\mathbf{v} = (x_2 - x_1, y_2 - y_1)^\top$. Without loss of generality, we will assume that $y_1 > Y_{\max} > Y_{\min} > y_2$, to ensure that the start and end points are above and below the hexel grid.

Stage I: Compute range of parametric distances

Project the points $P_1 = (x_1, y_1)^\top$ and $P_2 = (x_2, y_2)^\top$ along the path onto the rectangular hull to new points $\widetilde{P}_1$ and $\widetilde{P}_2$, such that

$$\widetilde{P}_1 = \begin{cases} (X_{\min}, y_1 + G(X_{\min} - x_1))^\top & \text{if } x_1 < X_{\min} < x_2 \\ & \text{and } G < (Y_{\max} - y_1)/(X_{\min} - x_1), \\ (X_{\max}, y_1 + G(X_{\max} - x_1))^\top & \text{if } x_2 < X_{\max} < x_1 \\ & \text{and } G > (Y_{\max} - y_1)/(X_{\max} - x_1), \\ (x_1 + (Y_{\max} - y_1)/G, Y_{\max})^\top & \text{otherwise,} \end{cases} \tag{8.30}$$

and

$$\widetilde{P}_2 = \begin{cases} (X_{\min}, y_2 + G(X_{\min} - x_2))^\top & \text{if } x_2 < X_{\min} < x_1 \\ & \text{and } G > (Y_{\min} - y_2)/(X_{\min} - x_2), \\ (X_{\max}, y_2 + G(X_{\max} - x_2))^\top & \text{if } x_1 < X_{\max} < x_2 \\ & \text{and } G < (Y_{\min} - y_2)/(X_{\max} - x_2), \\ (x_2 + (Y_{\min} - y_2)/G, Y_{\min})^\top & \text{otherwise.} \end{cases} \tag{8.31}$$

It is possible that either projection may fall on a hexel edge at height $Y_{\min}$ or $Y_{\max}$, but with probability zero will either projection land on a hexel vertex on the vertical line $x = X_{\min}$ or $x = X_{\max}$. Therefore, the projections $\widetilde{P}_1$ and $\widetilde{P}_2$ do not automatically define the distances $t_{\min}$ and $t_{\max}$ as shown in Figure 8.9. To treat the sets $\{\mathcal{L}_\alpha, \mathcal{L}_\beta, \mathcal{L}_\gamma\}$ equally, it is useful to instead use a point $Q_1 = (1-\tau)P_1 + \tau\widetilde{P}_1$, for some τ that ensures that Q_1 is no more than a single inradius away from the rectangular hull. This forces Q_1 to exist within a 'boundary' hexel that is adjacent to the grid, so that the first intersected hexel will be part of the original grid. Defining Q_2 in a similar way, we take these as the points from which we will compute all distances, with $L = \|Q_1 - Q_2\|_2$ being the total distance of the path.

Stage II: Compute distances

We denote the parallel and orthogonal directions of the sets $\mathcal{L}_\alpha$, $\mathcal{L}_\beta$, and $\mathcal{L}_\gamma$ by

$$\mathbf{v}_\alpha = (\cos(\alpha), \sin(\alpha))^\top, \qquad \mathbf{v}_\alpha^\perp = (-\sin(\alpha), \cos(\alpha))^\top, \tag{8.32a}$$

$$\mathbf{v}_\beta = (\cos(\beta), \sin(\beta))^\top, \qquad \mathbf{v}_\beta^\perp = (-\sin(\beta), \cos(\beta))^\top, \tag{8.32b}$$

$$\mathbf{v}_\gamma = (\cos(\gamma), \sin(\gamma))^\top, \qquad \mathbf{v}_\gamma^\perp = (-\sin(\gamma), \cos(\gamma))^\top. \tag{8.32c}$$

We then compute the angles made between the path and sets $\mathcal{L}_\alpha$, $\mathcal{L}_\beta$, and $\mathcal{L}_\gamma$:

$$\phi_\alpha = \arccos\left(\langle \widehat{\mathbf{v}},\ \mathbf{v}_\alpha \rangle\right), \tag{8.33a}$$

$$\phi_\beta = \arccos\left(\langle \widehat{\mathbf{v}},\ \mathbf{v}_\beta \rangle\right), \tag{8.33b}$$

$$\phi_\gamma = \arccos\left(\langle \widehat{\mathbf{v}},\ \mathbf{v}_\gamma \rangle\right), \tag{8.33c}$$

where $\widehat{\mathbf{v}} := \mathbf{v}/\|\mathbf{v}\|_2$ is the unit direction of the path and the principal angles are taken to be in the interval $[0, \pi)$.

To determine the number of lines passed through each set from the centre of hexel $[\![0, 0, 0]\!]$ to Q_1 we first compute the orthogonal distances

$$p_\alpha = \left\langle (Q_1 - \mathbf{r}_0),\ \mathbf{v}_\alpha^\perp \right\rangle, \tag{8.34a}$$

$$p_\beta = \left\langle (Q_1 - \mathbf{r}_0),\ \mathbf{v}_\beta^\perp \right\rangle, \tag{8.34b}$$

$$p_\gamma = \left\langle (Q_1 - \mathbf{r}_0),\ \mathbf{v}_\gamma^\perp \right\rangle, \tag{8.34c}$$

and take the signed integer proportion of each, with respect to the hexel inradius,

$$N_\alpha = \operatorname{sgn}(p_\alpha)\, \lfloor (|p_\alpha|/R) \rfloor, \tag{8.35a}$$

$$N_\beta = \operatorname{sgn}(p_\beta)\, \lfloor (|p_\beta|/R) \rfloor, \tag{8.35b}$$

$$N_\gamma = \operatorname{sgn}(p_\gamma)\, \lfloor (|p_\gamma|/R) \rfloor. \tag{8.35c}$$

The sets $\mathcal{L}_\alpha$, $\mathcal{L}_\beta$, and $\mathcal{L}_\gamma$ have the following forms:

$$\mathcal{L}_\alpha = \{(x, y)^\top \in \mathbb{R}^2 \mid (y - y_0) = (x - x_0)\tan(-\pi/3) + 2KR\}, \tag{8.36a}$$

$$\mathcal{L}_\beta = \{(x, y)^\top \in \mathbb{R}^2 \mid (y - y_0) = KR\}, \tag{8.36b}$$

$$\mathcal{L}_\gamma = \{(x, y)^\top \in \mathbb{R}^2 \mid (y - y_0) = (x - x_0)\tan(\pi/3) + 2KR\}, \tag{8.36c}$$

for all $K \in \mathbb{Z}$. For each set, we denote the particular line associated with $K \in \mathbb{Z}$ by Π_K^α, Π_K^β, and Π_K^γ, respectively. The point Q_1 is then bounded by the six lines

$$\{\Pi_{N_\alpha}^\alpha, \Pi_{N_\alpha + \operatorname{sgn}(p_\alpha)}^\alpha, \Pi_{N_\beta}^\beta, \Pi_{N_\beta + \operatorname{sgn}\left(p_\beta\right)}^\beta, \Pi_{N_\gamma}^\gamma, \Pi_{N_\gamma + \operatorname{sgn}(p_\gamma)}^\gamma\}, \tag{8.37}$$

as we show in Figure 8.13.

The relative orientation of the vectors $(Q_1 - \mathbf{r}_0)$ and $(Q_1 - Q_2)$ determines which line will be crossed first by the path, so we identify the first-intersected lines by the

subscripts

$$N_\alpha^* = \begin{cases} \min\{N_\alpha,\ N_\alpha + \operatorname{sgn}(p_\alpha)\} & \text{if } \left\langle (Q_2 - Q_1),\ \mathbf{v}_\alpha^\perp \right\rangle < 0, \\ \max\{N_\alpha,\ N_\alpha + \operatorname{sgn}(p_\alpha)\} & \text{otherwise,} \end{cases} \tag{8.38a}$$

$$N_\beta^* = \begin{cases} \min\{N_\beta,\ N_\beta + \operatorname{sgn}(p_\beta)\} & \text{if } \left\langle (Q_2 - Q_1),\ \mathbf{v}_\beta^\perp \right\rangle < 0, \\ \max\{N_\beta,\ N_\beta + \operatorname{sgn}(p_\beta)\} & \text{otherwise,} \end{cases} \tag{8.38b}$$

$$N_\gamma^* = \begin{cases} \min\{N_\gamma,\ N_\gamma + \operatorname{sgn}(p_\gamma)\} & \text{if } \left\langle (Q_2 - Q_1),\ \mathbf{v}_\gamma^\perp \right\rangle < 0, \\ \max\{N_\gamma,\ N_\gamma + \operatorname{sgn}(p_\gamma)\} & \text{otherwise.} \end{cases} \tag{8.38c}$$

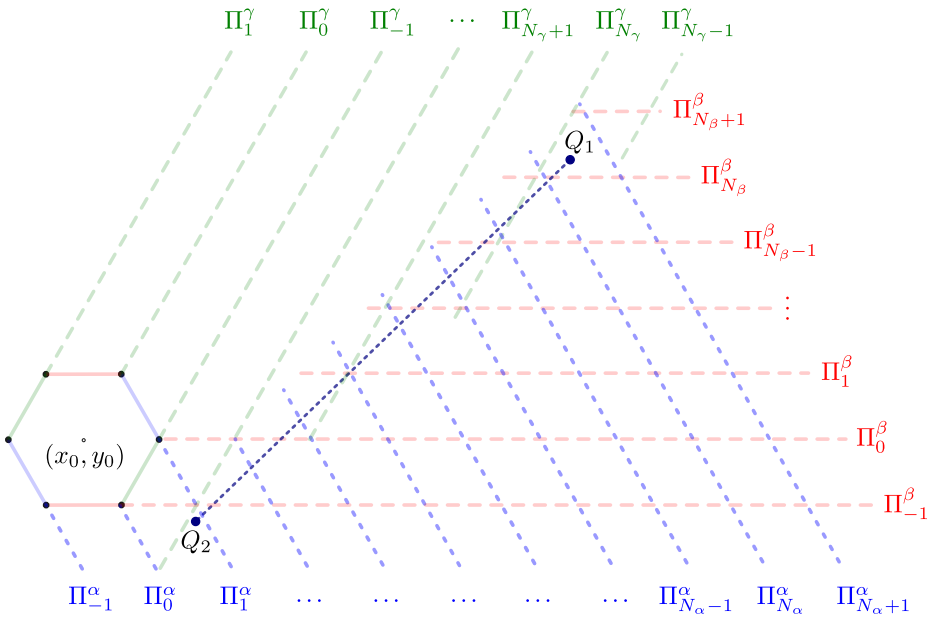


Figure 8.13: In this example, N_α and N_β are positive, while N_γ is negative. Due to the relative alignment of $(Q_1 - \mathbf{r}_0)$ and $(Q_1 - Q_2)$, the first path-intersected lines in each set are $\Pi^\alpha_{N_\alpha}$, $\Pi^\beta_{N_\beta}$ and $\Pi^\gamma_{N_\gamma}$, respectively.

As such, the first-intersected lines are $\Pi^\alpha_{N_\alpha^*}$, $\Pi^\beta_{N_\beta^*}$ and $\Pi^\gamma_{N_\gamma^*}$, and their respective intersection locations are

$$\mathbf{d}^\alpha_{N_\alpha^*} = \left(\frac{2RN_\alpha^* - \xi}{G - \tan(-\pi/3)} + x_0,\ \frac{2GRN_\alpha^* - \tan(-\pi/3)\xi}{G - \tan(-\pi/3)} + y_0 \right)^\top, \tag{8.39a}$$

$$\mathbf{d}^\beta_{N_\beta^*} = \left(\frac{RN_\beta^* - \xi}{G} + x_0,\ RN_\beta^* + y_0 \right)^\top, \tag{8.39b}$$

$$\mathbf{d}^\gamma_{N_\gamma^*} = \left(\frac{2RN_\gamma^* - \xi}{G - \tan(\pi/3)} + x_0,\ \frac{2GRN_\gamma^* - \tan(\pi/3)\xi}{G - \tan(\pi/3)} + y_0 \right)^\top, \tag{8.39c}$$

where $\xi = (y_1 - y_0) - G(x_1 - x_0)$.

Using a similar notation[22] to Siddon's original algorithm, the distances from Q_1 to each first intersection are

$$t_0^\alpha = \|Q_1 - \mathbf{d}_{N_\alpha^*}^\alpha\|_2, \qquad t_0^\beta = \|Q_1 - \mathbf{d}_{N_\beta^*}^\beta\|_2, \qquad t_0^\gamma = \|Q_1 - \mathbf{d}_{N_\gamma^*}^\gamma\|_2. \tag{8.40}$$

We then compute

$$T_\alpha = \left\lfloor \frac{L - t_0^\alpha}{R\csc(\phi_\alpha)} \right\rfloor, \qquad T_\beta = \left\lfloor \frac{L - t_0^\beta}{R\csc(\phi_\beta)} \right\rfloor, \qquad T_\gamma = \left\lfloor \frac{L - t_0^\gamma}{R\csc(\phi_\gamma)} \right\rfloor, \tag{8.41}$$

which give the total number of lines from each set crossed by the path. It is then trivial to compute successive intersection distances for each set

$$t_k^\alpha = t_0^\alpha + kR\csc(\phi_\alpha) \qquad \text{for } k = 1, \ldots, T_\alpha, \tag{8.42a}$$
$$t_k^\beta = t_0^\beta + kR\csc(\phi_\beta) \qquad \text{for } k = 1, \ldots, T_\beta, \tag{8.42b}$$
$$t_k^\gamma = t_0^\gamma + kR\csc(\phi_\gamma) \qquad \text{for } k = 1, \ldots, T_\gamma. \tag{8.42c}$$

The set $\{t_0^\alpha, \ldots, t_{T_\alpha}^\alpha, t_0^\beta, \ldots, t_{T_\beta}^\beta, t_0^\gamma, \ldots, t_{T_\gamma}^\gamma\}$ then contains the full set of intersection distances for all lines, genuine and spurious.

Stage III: Determine genuine distances

To compute whether the intersections are genuine or spurious, we first compute the set-aligned projection lengths

$$s_\alpha = R\cot(\phi_\alpha), \qquad s_\beta = R\cot(\phi_\beta), \qquad s_\gamma = R\cot(\phi_\gamma), \tag{8.43}$$

and the relative set-aligned distances of the first intersections for each line

$$s_0^\alpha = \left\langle (\mathbf{d}_{N_\alpha^*}^\alpha - \mathbf{r}_0),\, \mathbf{v}_\alpha \right\rangle, \tag{8.44a}$$
$$s_0^\beta = \left\langle (\mathbf{d}_{N_\beta^*}^\beta - \mathbf{r}_0),\, \mathbf{v}_\beta \right\rangle, \tag{8.44b}$$
$$s_0^\gamma = \left\langle (\mathbf{d}_{N_\gamma^*}^\gamma - \mathbf{r}_0),\, \mathbf{v}_\gamma \right\rangle. \tag{8.44c}$$

By measuring distances relative to the centre $\mathbf{r}_0 = (x_0, y_0)^\top$, we can perform the simple calculations shown in Figure 8.11. Due to the equal spacing between all parallel lines,

[22] In this form we use the distances directly, such that each is in the interval $(0, L)$, while Siddon's algorithm scales down to the interval $(0, 1)$ to then rescale at the end. Since intersection distances are computed multiplicatively, rather than through sequential additions, floating-point accumulation errors are not a concern.

the intersection locations are then

$$s_k^\alpha = s_0^\alpha - k s_\alpha \qquad \text{for } k = 1, \ldots, T_\alpha, \tag{8.45a}$$

$$s_k^\beta = s_0^\beta - k s_\beta \qquad \text{for } k = 1, \ldots, T_\beta, \tag{8.45b}$$

$$s_k^\gamma = s_0^\gamma - k s_\gamma \qquad \text{for } k = 1, \ldots, T_\gamma. \tag{8.45c}$$

In (8.26), the logical checks were written in terms of the subscript parity of the line that is intersected, but this is equally determined by the parity of the first-intersected[23] line:

$$\sigma_k^\alpha = \begin{cases} 1 & \text{if } \left(s_k^\alpha - w + \frac{3w}{2}\Big((k + N_\alpha^*) \pmod 2 \Big) \right) \pmod{3w} \in [0, w], \\ 0 & \text{otherwise,} \end{cases} \tag{8.46a}$$

$$\sigma_k^\beta = \begin{cases} 1 & \text{if } \left(s_k^\beta - w + \frac{3w}{2}\Big((k + N_\beta^*) \pmod 2 \Big) \right) \pmod{3w} \in [0, w], \\ 0 & \text{otherwise,} \end{cases} \tag{8.46b}$$

$$\sigma_k^\gamma = \begin{cases} 1 & \text{if } \left(s_k^\gamma - w + \frac{3w}{2}\Big((k + N_\gamma^*) \pmod 2 \Big) \right) \pmod{3w} \in [0, w], \\ 0 & \text{otherwise.} \end{cases} \tag{8.46c}$$

Multiplying each intersection distance by its associated logical value, the set

$$\mathcal{R} = \{ t_0^\alpha \sigma_0^\alpha, \ldots, t_{T_\alpha}^\alpha \sigma_{T_\alpha}^\alpha, t_0^\beta \sigma_0^\beta, \ldots, t_{T_\beta}^\beta \sigma_{T_\beta}^\beta, t_0^\gamma \sigma_0^\gamma, \ldots, t_{T_\gamma}^\gamma \sigma_{T_\gamma}^\gamma \} \tag{8.47}$$

contains only genuine intersection distances and zeroed-out spurious intersections. Removing the zero entries, which correspond to spurious intersections, we are then left with the set

$$\mathcal{R}_{\text{gen}} = \{\, t_i^\alpha \,\}_{\text{gen}} \cup \{\, t_j^\beta \,\}_{\text{gen}} \cup \{\, t_k^\gamma \,\}_{\text{gen}}. \tag{8.48}$$

This set contains information only about the distances to genuine hexel edges, from which the intersection lengths can be computed directly.

Stage IV: Create index shift lists

The set $\mathcal{R}_{\text{gen}}$ contains only genuine intersection distances, so these can be used to calculate the list of pixel index shifts that occur while traversing the path. We use a cubic coordinate system, $[\![i, j, k]\!]$ for which $i + j + k = 0$, such that crossing a

[23] Note that in (8.25), the line index M is a function of both k and the first-intersected line index N^*. Importantly, we need not concern ourselves with the specific form of M—which depends on the direction in which the ray travels—but instead exploit the fact that the parity of M is identical to the parity of $(k + N^*)$.

member of $\mathcal{L}_\alpha$ is associated with the index shift $[\![0, 1, -1]\!]$, crossing a member of $\mathcal{L}_\beta$ is associated with $[\![1, 0, -1]\!]$, and crossing a member of $\mathcal{L}_\gamma$ is associated with $[\![1, -1, 0]\!]$.

We then compute the three index shifts

$$\chi_\alpha = \operatorname{sgn}\left(\left\langle (Q_2 - Q_1),\, \mathbf{v}_\alpha^\perp \right\rangle\right) [\![0, 1, -1]\!], \tag{8.49a}$$

$$\chi_\beta = \operatorname{sgn}\left(\left\langle (Q_2 - Q_1),\, \mathbf{v}_\beta^\perp \right\rangle\right) [\![1, 0, -1]\!], \tag{8.49b}$$

$$\chi_\gamma = \operatorname{sgn}\left(\left\langle (Q_2 - Q_1),\, \mathbf{v}_\gamma^\perp \right\rangle\right) [\![1, -1, 0]\!], \tag{8.49c}$$

and create the following sets

$$\mathcal{A} = \{\chi_\alpha,\ \chi_\alpha,\ \ldots,\ \chi_\alpha\}, \tag{8.50a}$$

$$\mathcal{B} = \{\chi_\beta,\ \chi_\beta,\ \ldots,\ \chi_\beta\}, \tag{8.50b}$$

$$\mathcal{C} = \{\chi_\gamma,\ \chi_\gamma,\ \ldots,\ \chi_\gamma\}, \tag{8.50c}$$

such that $|\mathcal{A}| = |\{\, t_i^\alpha \,\}_{\text{gen}}|$, $|\mathcal{B}| = |\{\, t_j^\beta \,\}_{\text{gen}}|$, and $|\mathcal{C}| = |\{\, t_k^\gamma \,\}_{\text{gen}}|$. The total set of index shifts, denoted $\mathcal{H} = \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$, has the same number of elements as $\mathcal{R}_{\text{gen}}$, and will be sorted according to the same permutation as $\mathcal{R}_{\text{gen}}$.

Stage V: Sort distances and compute intersection lengths

The set $\mathcal{R}_{\text{gen}}$ has elements in the range $(0, L)$, with the probability of repeated values being essentially zero, as this would correspond to the path intersecting a hexel vertex rather than an edge. We can therefore sort the elements using K-way merging, for $K = 3$, to create an ordered t-parametrised set

$$\mathcal{R}_{\text{sort}} = \{t_0, t_1, \ldots, t_k, t_{k+1}\}, \tag{8.51}$$

and then compute the intersection distances throughout the pixel grid to be

$$\ell_m = (t_m - t_{m-1}) \qquad \text{for} \quad m = 1, \ldots, k+1. \tag{8.52}$$

Stage VI: Sort indices and compute ordered index list

Let $\boldsymbol{\pi}$ be the permutation that maps $\mathcal{R}$ to $\mathcal{R}_{\text{sort}}$, i.e. $(\mathcal{R}_{\text{sort}})_m = \mathcal{R}_{\boldsymbol{\pi}(m)}$ for $m = 1, \ldots, k+2$. The same permutation can be applied to $\mathcal{H}$ without reference to the values of $\mathcal{H}$, such that the result is the list $\mathcal{H}_{\text{sort}}$ for which $(\mathcal{H}_{\text{sort}})_m = \mathcal{H}_{\boldsymbol{\pi}(m)}$ for $m = 1, \ldots, k+2$. The list $\mathcal{H}_{\text{sort}}$ then contains a sequence of index shifts associated with the ordered distances in $\mathcal{R}_{\text{sort}}$.

We constrained Q_1 to exist in a boundary hexel, such that if we extend the grid by one hexel in each direction, to a size of $(N_x + 2) \times (N_y + 2)$ then Q_1 exists in a pixel with some index $[\![q_1, q_2, q_3]\!]$. For the purpose of computing the indices of real pixels, this means that the sequence can be computed with respect to the index of the first intersected pixel. The determination of the pixel index from a point $Q_1 \in \mathbb{R}^2$ is not as straightforward for hexels as it is for square pixels, but it is efficiently computable.

We assume that our hexel grid has a flat-top orientation[24], that there is an identifiable $[\![0, 0, 0]\!]$ hexel, and that the index steps $[\![0, 1, -1]\!]$ and $[\![1, 0, -1]\!]$ are associated with the vectors $\mathbf{v}_1 = (\sqrt{3}R, R)^\top$ and $\mathbf{v}_2 = (0, 2R)^\top$, respectively. These vectors define the distances between centres of neighbouring hexels along the chosen basis directions.

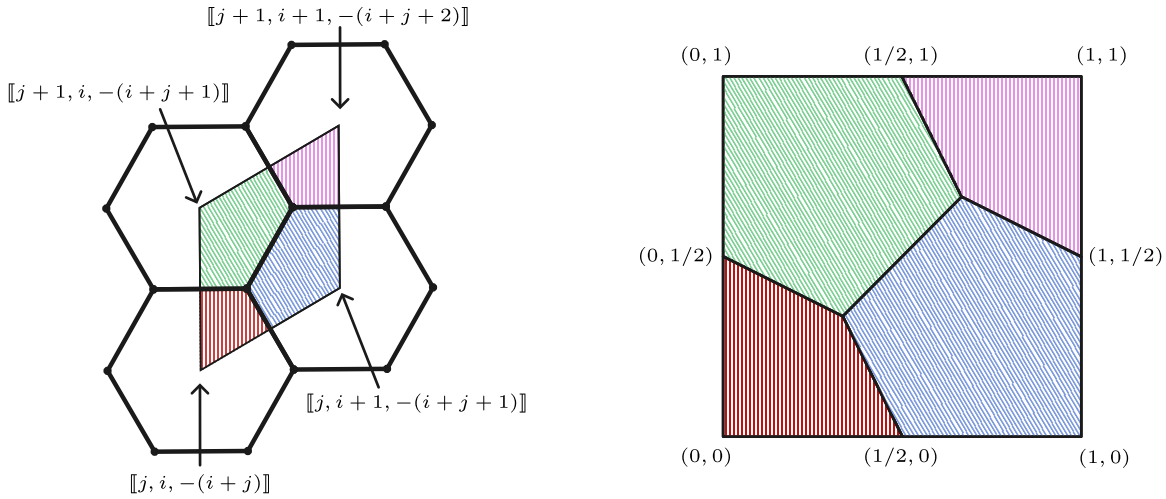


Figure 8.14: Determining the correct hexel index can be performed by computing which of the four regions of the unit square the fractional coordinates $(a_1, a_2)^\top$ are in.

We compute the decomposition $Q_1 = X\mathbf{v}_1 + Y\mathbf{v}_2$ in the $\{\mathbf{v}_1, \mathbf{v}_2\}$ basis, with respect to the centre of hexel $[\![0, 0, 0]\!]$, take $i = \lfloor X \rfloor$ and $j = \lfloor Y \rfloor$ and compute the fractional distance from the centre of hexel $[\![j, i, -(i + j)]\!]$ with respect to $\mathbf{v}_1$ and $\mathbf{v}_2$ by

$$a_1 = X - i, \qquad a_2 = Y - j. \tag{8.53}$$

From here, it is clear that the point Q_1 is located in one of four hexels—see Figure 8.14—and determining which hexel is correct requires at most three inequality checks[25], as we show in Algorithm 3.

[24] For a 'pointed-top' indexing system, we merely rotate the vectors by $\theta = \pi/3$; details can also be found in [70].

[25] The comments in Algorithm 3 indicate the candidate vertices from Figure 8.14. Alternatively, it is possible to instead find the four closest hexel centres and then compare the Euclidean distance to each—this works since each hexel is equivalently the Voronoi cell of its centre point.

Algorithm 3: Hexel Index from Cartesian Coordinates

Data: $\{a_1, a_2, i, j\}$

if $a_2 > a_1$ // {(0,0),(0,1),(1,1)}
  if $a_2 + a_1/2 < 1/2$ // {(0,0)}
    $[\![q_1, q_2, q_3]\!] \leftarrow [\![j, i, -(i+j)]\!]$;
  else if $a_2 + 2a_1 > 2$ // {(1,1)}
    $[\![q_1, q_2, q_3]\!] \leftarrow [\![j+1, i+1, -(i+j+2)]\!]$;
  else // {(0,1)}
    $[\![q_1, q_2, q_3]\!] \leftarrow [\![j+1, i, -(i+j+1)]\!]$;
  end
else // {(0,0),(1,0),(1,1)}
  if $a_2 + 2a_1 < 1$ // {(0,0)}
    $[\![q_1, q_2, q_3]\!] \leftarrow [\![j, i, -(i+j)]\!]$;
  else if $a_2 + a_1/2 > 1$ // {(1,1)}
    $[\![q_1, q_2, q_3]\!] \leftarrow [\![j+1, i+1, -(i+j+2)]\!]$;
  else // {(1,0)}
    $[\![q_1, q_2, q_3]\!] \leftarrow [\![j, i+1, -(i+j+1)]\!]$;
  end
end

The sequence of elements $\mathcal{H}_{\text{sort}} = \{\chi_0, \chi_1, \ldots, \chi_k, \chi_{k+1}\}$ is then used to find the index of the mth intersected hexel:

$$[\![i_m, j_m, k_m]\!] = [\![i_{m-1}, j_{m-1}, k_{m-1}]\!] + \chi_m, \tag{8.54}$$

$$= [\![q_1, q_2, q_3]\!] + \sum_{n=1}^{m} \chi_n. \tag{8.55}$$

If the index of a specific intersected hexel is required without computing all preceding indices, we can take the distance $\widetilde{t} = (t_{m-1} + t_m)/2$ and then find the associated pixel index of the point $P(\widetilde{t})$ using Algorithm 3. Therefore, as with Siddon's algorithm, our approach allows the intersection length and index of any particular hexel to be determined independently of the preceding traversal.

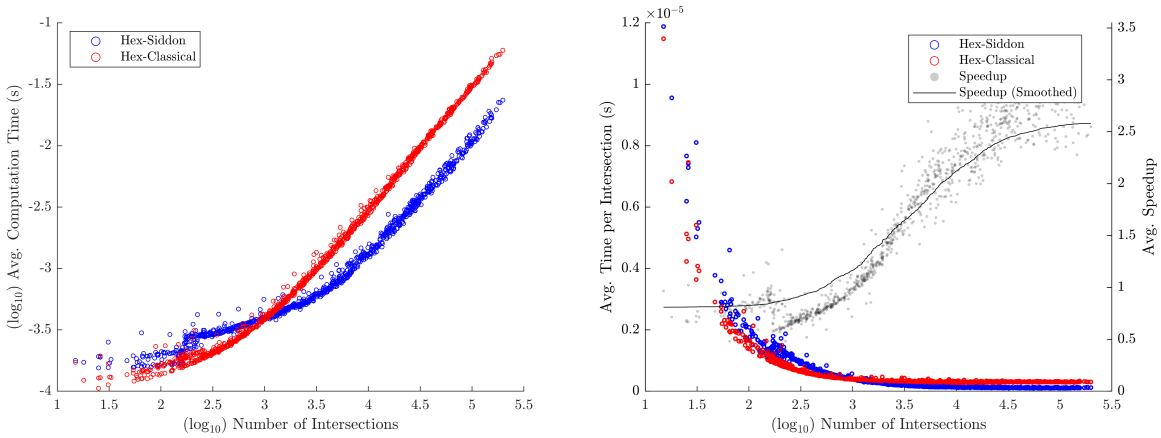
We now have all the elements necessary to compute the radiological path

$$\mathfrak{L} = \sum_{(i,j,k)\,\in\,\Omega} \ell_{i,j,k} = \sum_{m=1}^{k+1} \ell_{i_m, j_m, k_m}, \tag{8.56}$$

for $\Omega := \{(i, j, k) \mid \ell_{i,j,k} \neq 0\}$.

8.5 Numerical results

We compare the implementation of our algorithm against the classical ray-marching algorithm tested against Hex-FVT. We compute successive intersections through hexels, for grids of size $N \times N$ where N ranges between $\mathcal{O}(10^2)$ and $\mathcal{O}(10^5)$, presenting results in Figure 8.15. Each data point represents a randomly sampled ray with origin $P_1 \in \mathbb{S}^1$ and direction $\mathbf{v} \in \mathbb{S}^1$; the total dataset is 1000 rays of increasing length, with the number of intersections being proportional to the total length of the ray[26]. In keeping with Siddon's algorithm, the computational cost of our hexagonal extension is linear in the number of intersections, i.e. it is $\mathcal{O}(N)$ for a grid of size $N \times N$.



(a) (b)

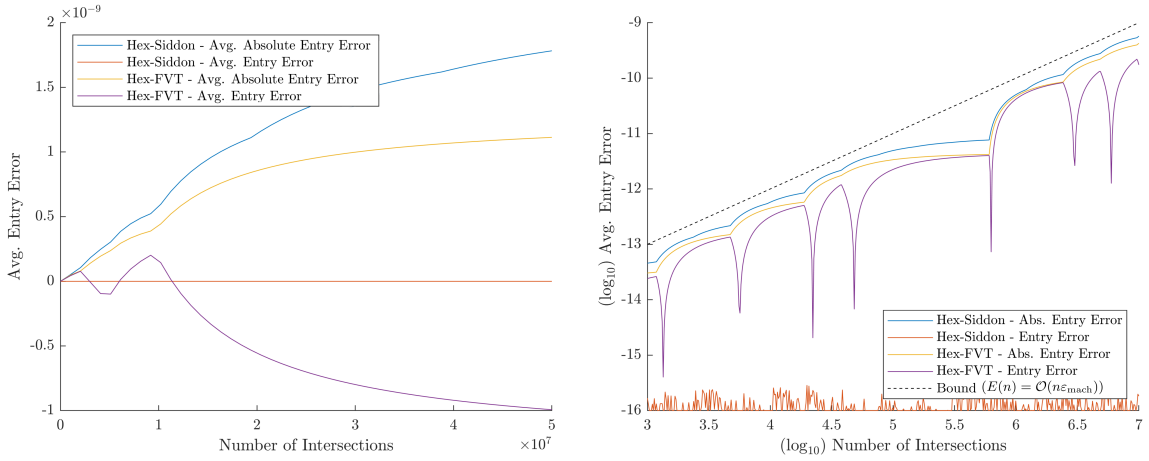
Figure 8.15: (a) The computational cost of our new method (Hex-Siddon) is linear in the number of intersections, which is also linear in the dimension of the hexagonal grid, on average. (b) Amortised costs of Hex-Siddon compared to classical marching methods can attain a computational speed-up of up to $\sim 2.5\times$, depending on the grid resolution.

Implementations for 'Hex-Classical' (the standard ray-marching algorithm) and 'Hex-Siddon' (our new algorithm) are written in MATLAB 2023A and run on a commercially available laptop (Surface Book 3, Intel i7-1065G7 CPU, 32GB RAM, Windows 11 Pro). Even after applying minor optimisations to the classical ray-marching implementation, Hex-Siddon achieves significant computational reductions, with a speed-up factor of $\sim 2.5\times$ for hexel grids with up to 5×10^5 intersections. It is likely that much of the speed-up is achieved due to the elimination of sequential information, allowing many

[26] As the rays are randomly sampled, it is not possible to guarantee that the number of intersections is a fixed multiple of the total length of the ray. This may explain some of the scattering for the shorter rays, although for longer rays it appears that the large number of intersections becomes the dominant cost, and the effect becomes less significant.

steps of the Hex-Siddon implementation to be vectorised directly in MATLAB. In addition, the calculations for our new algorithm are now embarrassingly parallel—as we will show in Section 8.5.1—as Hex-Siddon does not rely on looping information, and so the main calculations can be performed in any order prior to the sorting step. While alternatives to Siddon's algorithm on square pixel grids have argued that the sorting step can be a computational bottleneck, for the hexagonal extension the simplicity and efficiency of computing intersection lengths more than justifies the additional sorting step.

In addition to offering large computational speed-up, our method is more numerically robust than both Hex-Classical and Hex-FVT, as we show in Figure 8.16b. The blue line in Figure 8.16a shows that the average error magnitude of computed intersection lengths increases monotonically, and is consistently larger than the error of Hex-FVT. However, the red line shows that there is no biased error, as the average error over all intersections is on the order $\mathcal{O}(\varepsilon_{\text{mach}})$.



(a) (b)

Figure 8.16: (a) The entry-wise errors of Hex-Siddon are slightly larger than for Hex-FVT. However, when averaged over the entire path, the average entry error is $\mathcal{O}(\varepsilon_{\text{mach}})$. (b) The average absolute error of entries using Hex-Siddon is still small enough to be bounded by $\mathcal{O}(n\varepsilon_{\text{mach}})$, over paths with n total intersections.

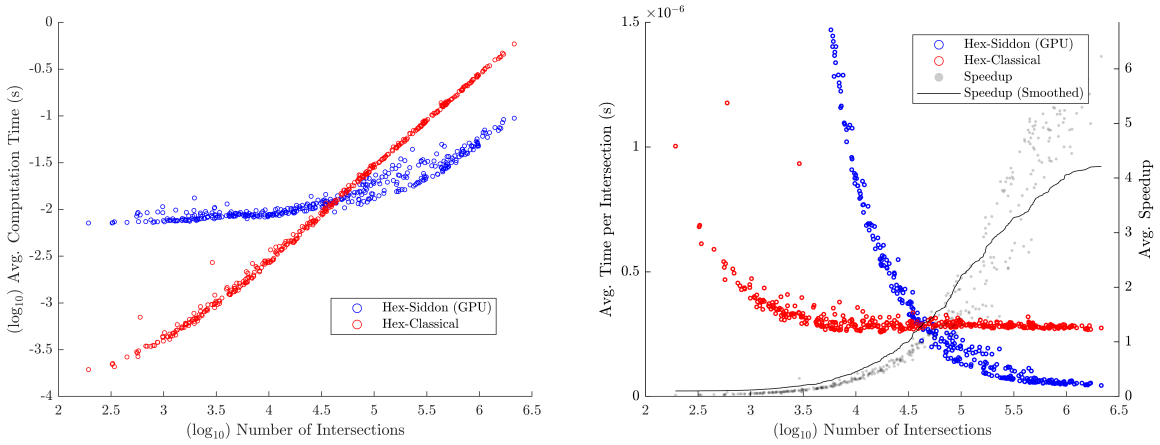
This is more clearly seen in Figure 8.16b where the average errors of Hex-FVT have the overall behaviour of $\mathcal{O}(n\varepsilon_{\text{mach}})$, while the average error for Hex-Siddon does not grow with the scale of the total path. It follows that for a path whose pixels have inradius $\mathcal{O}(1)$, the relative error of the sum of intersection lengths will be $\mathcal{O}(\varepsilon_{\text{mach}})$.

8.5.1 Parallelisability

Our proposed algorithm already demonstrates a performance improvement over classical cell-stepping approaches. In its serial MATLAB implementation, the method achieves an approximate $\sim 2.5\times$ speed-up, primarily due to the elimination of per-cell traversal logic and a reduced number of conditional operations along each ray. This confirms that the algorithmic restructuring alone—independent of hardware acceleration—yields a significant efficiency gain.

To further evaluate performance, a parallelised version of the algorithm was implemented using MATLAB's Parallel Computing Toolbox, leveraging GPU acceleration via `gpuArray()` and associated functions. Importantly, we have not used any tailored parallel programming platform, and have not focused on writing architecture-specific kernels, so all acceleration is obtained directly through MATLAB's existing GPU abstractions. As such, our implementation is relatively portable and straightforward to reproduce.

As can be seen in Figure 8.17a, for a sufficiently large number of intersections (roughly $N > 10^{9/2}$) the total computation time is, on average, substantially less than for the serial algorithm. In fact, it appears that until $N \sim 10^4$ the dominant cost is the data transfer and setup costs for using `gpuArray()` and similar MATLAB-specific functions.



(a) (b)

Figure 8.17: (a) The computational cost is still linear, although the initial setup costs are more demanding than for the non-parallelised implementation. (b) Average costs of Hex-Siddon using parallel computing can attain a computational speed-up of up to $\sim 4-6\times$, depending on the grid resolution. As expected, the benefit of using parallel programming is best seen for larger numbers of intersections.

The asymptotic behaviour is better seen in Figure 8.17b, where the parallelised version achieves an approximate 4–6× speed-up compared to the classical cell-stepping implementation. This behaviour is consistent with the overhead amortisation expected in GPU computing: once data transfer and kernel launch costs become negligible relative to total computation, the high arithmetic throughput of the GPU can be fully utilised.

All experiments were conducted on a consumer-grade GPU, specifically an NVIDIA Quadro RTX 3000 (Max-Q) integrated into a Microsoft Surface Book 3. As such, the reported gains should be interpreted as conservative: no high-end compute GPU was used, and no effort was made to hand-optimise memory access patterns or instruction-level parallelism via custom kernels. The observed performance improvements therefore reflect a lower bound on what could be achieved with a dedicated implementation on a specific hardware architecture.

Taken together, these results highlight two key points. First, the majority of the performance improvement arises from algorithmic design, rather than hardware specialisation. Second, even with only a simplistic GPU implementation, substantial acceleration can be obtained for large-scale problems using high-level parallel tools. This suggests clear potential for further speed-ups in future work through custom kernel development and more aggressive hardware-specific optimisation.

The decision to not implement a custom GPU kernel is intentional: the primary goal was to optimise and validate the algorithmic structure itself within a high-level programming environment, rather than to pursue maximum hardware-specific performance. Using MATLAB's Parallel Computing Toolbox enabled quicker code development and iteration, and direct comparison with the existing serial implementations.

8.6 Hexagonal image reconstruction

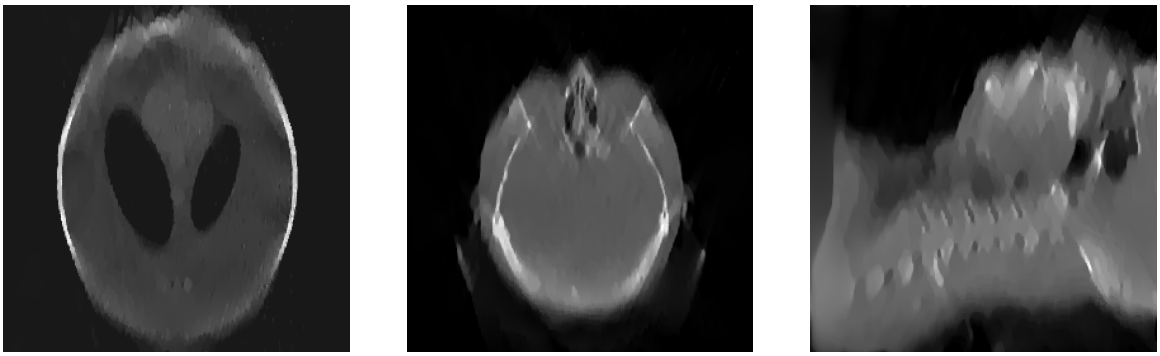
As this chapter was introduced with the idea of producing hexel reconstructions of the phantoms discussed in Chapters 5 and 6, we now present reconstructions obtained using our implementation of Hex-Siddon, along with an augmented version of TVAL3.

In particular, through the indexing scheme $[\![i, j, k]\!]$ introduced for our hexagonal grid, the anisotropic TV-norm for hexagonal images becomes

$$\|\mathbf{x}\|_{\mathrm{TV}}^{\mathrm{Hex}} = \sum_{i,\,j,\,k} |\mathbf{x}_{i,j+1,k-1} - \mathbf{x}_{i,j,k}| + |\mathbf{x}_{i+1,j,k-1} - \mathbf{x}_{i,j,k}| + |\mathbf{x}_{i+1,j-1,k} - \mathbf{x}_{i,j,k}|. \tag{8.57}$$

In effect, the structure of the hexagonal TV-norm has the same three-term forward-difference structure as the cubic TV-norm in three dimensions[27], with each hexel's six edge-neighbours playing the role of a cubic voxel's six face-neighbours.

Hexagonal reconstructions are defined over the set of full hexels whose centres lie within the reconstruction domain, leading to a non-rectangular boundary, which has been cropped in the images presented in Figure 8.18. Partial cells are deliberately avoided to maintain uniform cell construction, so that image comparisons are restricted to cells within the sub-domain. For comparison against a 256×256 resolution square pixel image, the associated hexagonal image consists of 238×275 hexels—a reduction of 0.13%. While this inherently creates a forward operator which is smaller, this is expected to have negligible effect on reconstruction quality.



(a) (b) (c)

Figure 8.18: $[(m/n) \approx 0.3, \lambda = \lambda_{\text{2D}}]$ Reconstructions obtained by allowing each phantom to be translated in the positive x direction by the amount which maximises their SNR—reconstructions can be compared directly to those in Figure 5.6.

In Figure 8.19 we present a comparison between reconstructions of SLP-2D at a pixel resolution of 500×500 and a hexel resolution of 537×465, with a total difference of $\sim 0.11\%$ in the number of cells. These reconstructions demonstrate that hexagonally discretised forward operators are a viable and valuable tool for algebraic image reconstruction; not only is the computational cost similar to that of square pixels, but the image fidelity is comparable or even better in certain circumstances.

[27]Passing hexel indices $[\![i, j, k]\!]$ directly into TVAL3 is not straightforward, as TVAL3 has a hard-coded check that a two-dimensional image is of size $N_x \times N_y$ to ensure that all internal operators are built correctly. We therefore use a two-variable index $[\![I, J]\!]$ over a non-orthogonal lattice basis $\{\mathbf{m}_1, \mathbf{m}_2\} \subset \mathbb{R}^2$, to which any hexel index $[\![i, j, k]\!]$ uniquely maps, and compute the TV-norm as

$$\|\mathbf{x}\|_{\text{TV}}^{\text{Hex}} = \sum_{I,J} |\mathbf{x}_{I+1,J} - \mathbf{x}_{I,J}| + |\mathbf{x}_{I,J+1} - \mathbf{x}_{I,J}| + |\mathbf{x}_{I\pm1,J\pm1} - \mathbf{x}_{I,J}|. \tag{8.58}$$

The subscript $(I \pm 1, J \pm 1)$ in the third term will depend on the particular choice of $\{\mathbf{m}_1, \mathbf{m}_2\}$.

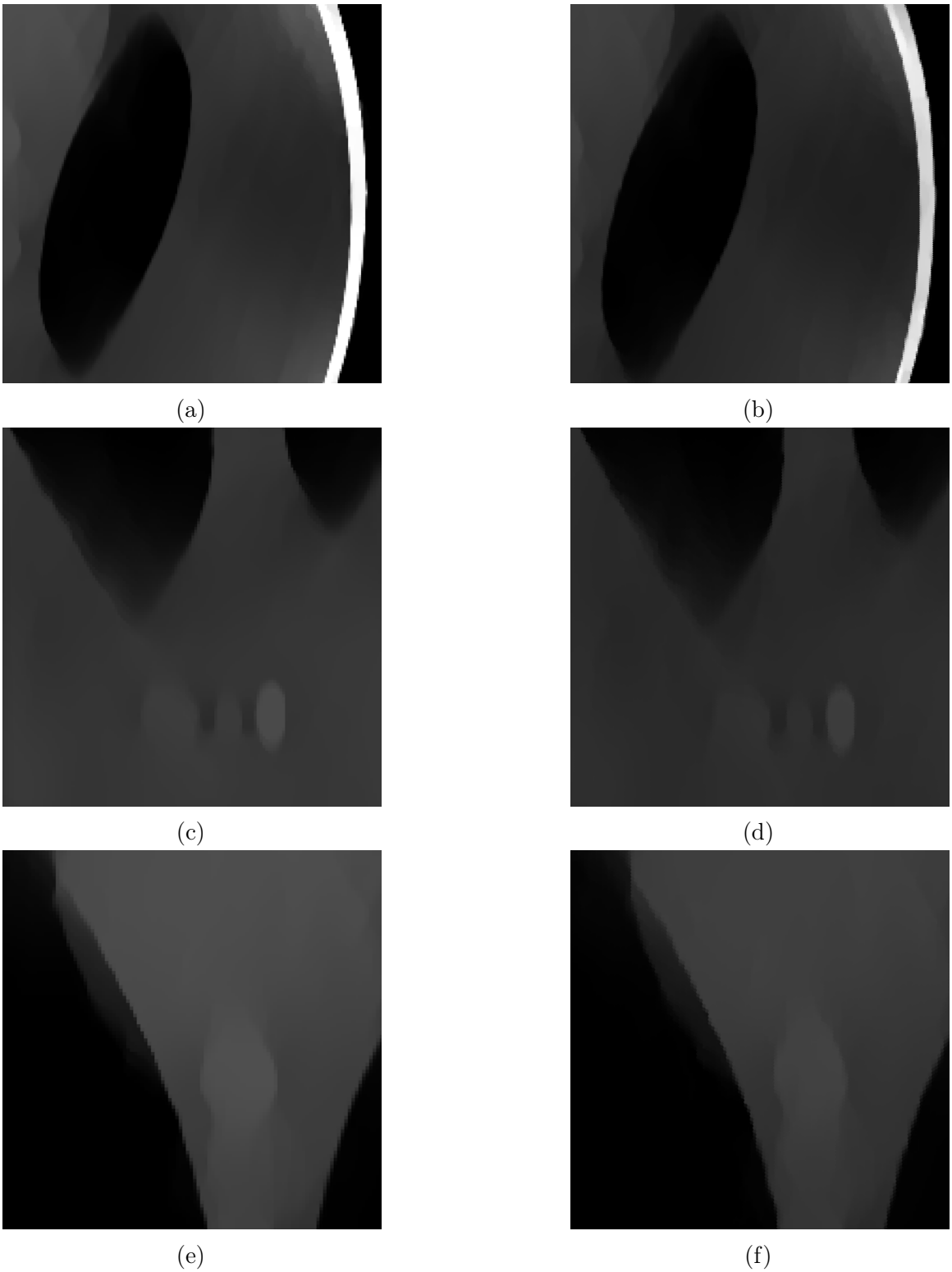
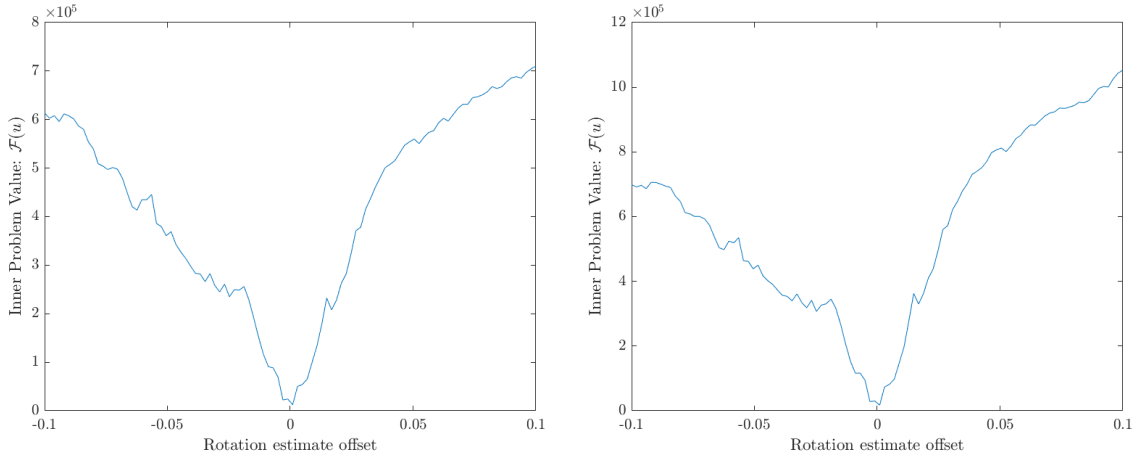


Figure 8.19: (a,c,e) Pixel reconstruction. (b,d,f) Hexel reconstruction. These images show that hexel reconstruction can distinguish the same internal details as pixels can, with almost no difference in image clarity. While internal curvature is similar for both, images (a) and (b) show that hexel representation of the body edge is slightly smoother, with fewer 'staircasing' effects.

While the images in Figure 8.19 validate our hexagonal reconstruction framework, demonstrating a clear benefit of hexagonal discretisation for optimisation is more challenging. In Figure 8.20 we show two parameter sweeps for a one-dimensional rotation model, where SLP-2D is transformed under the parameter $\widehat{u} = 0.15$. Both plots are normalised to have the minimum occur around zero.



(a) (b)

Figure 8.20: (a) Cost function for a 1D rotation model, using square pixels. (b) Cost function for a 1D rotation model, using hexels. For both plots the data has been normalised around $\widehat{u} = 0.15$ so that the plotted minimum occurs at, or rather *near*, zero. The plots show very similar behaviour, up to a scaling, with neither discretisation necessarily being preferable for correcting rotational motion.

The plots in Figure 8.20 are encouraging, as they validate that a hexagonal forward operator invokes similar behaviour of $\mathcal{F}(u)$ via the solution of (4.4). For further implementations it may be worthwhile to fine-tune the chosen λ for the hexagonal implementation; all results in this chapter were obtained with $\lambda = 10^{-3}$ as tests similar to those shown in Figure 4.6 identified this choice as the best for stationary hexagonal problems. It is plausible that to solve (4.5) reliably, the value of λ must also be taken as a parameter that is augmented throughout a particular optimisation scheme; this investigation is beyond the scope of our work.

The results included in this chapter show that hexagonal discretisations are a viable option for undersampled image reconstruction—image clarity and fidelity are on par with results obtained using square pixel grids, and the total computation time involved is comparable. The algorithmic improvements of both Hex-FVT and Hex-Siddon allow forward operators $\mathbf{A}(u)$ to be built without additional difficulties, finding that the reconstruction of edges using hexagonal grids is marginally more effective.

These results naturally lead to deeper investigations into parameter-tuning for alternative discretisation choices, but this is beyond the scope of the current work. To close Part III, in Chapter 9 we will extend the algorithmic framework developed in this chapter to show that parallel-per-cell algorithms are amenable to a wider range of non-standard discretisations, without introducing additional computational overhead.

Chapter 9

Dodecahedral grid representation

Contents

9.1 Extending Siddon's algorithm to rhombic dodecahedral voxels 183

9.2 Numerical results 197

The results in Chapter 8 show that reconstructions obtained using a hexagonally discretised forward operator can be as accurate as those obtained under traditional discretisations. Additionally, the wall-clock time taken to construct the forward operator is measured to be similar to the time needed for square pixel models.

The direct extension of this idea to three dimensions requires the use of rhombic dodecahedral voxels—12-faced polyhedra that tessellate three-dimensional space. This direct association follows from the following fact: the hexagonal lattice $\Lambda(\mathbf{M})$ is, up to isometry, the lattice of centres of disks in the densest packing of congruent disks in the plane. Therefore, the regular hexagonal tiling coincides with the Voronoi tessellation of this particular lattice. In three dimensions, the densest lattice packing of congruent spheres is given by the face-centred cubic (FCC) lattice, of which the Voronoi cells are rhombic dodecahedra, yielding a space-filling tiling of $\mathbb{R}^3$.

Beyond this geometric correspondence, rhombic dodecahedral voxels—for brevity, we will refer to these elements as *roxels* (for **r**hombic-v**oxels**)—possess properties that make them a natural, if largely unexplored, candidate for three-dimensional reconstruction. Roxels exhibit a higher degree of angular isotropy than Cartesian cubes, with each cell sharing identical face geometry and a uniform set of twelve nearest neighbours. This symmetry leads to more uniform sampling of local interactions, which can influence both the discretisation error and regularisation of iterative reconstruction.

Despite these potential advantages, roxel-based discretisations have received little attention in the imaging literature, with only a small number of papers considering them for the representation of volumetric data [11, 12, 56]. The work in this chapter therefore explores roxels not as a geometric novelty, but as a principled alternative discretisation whose properties may better align with the rotational invariance of tomographic acquisition.

The primary focus of this chapter is to extend the development of the Hex-Siddon algorithm into three dimensions, so that the construction of forward operators using this new discretisation is not a computational hurdle. The inclusion of this extended algorithm into the complete reconstruction pipeline, using roxel-based forward operators, will be undertaken in future work. As we will briefly discuss towards the end of this chapter, the generalisation of Siddon's algorithm to roxels opens up a general class of periodic tilings for which cell-parallelisable ray traversal is feasible.

Similar to the construction of a hexel, the structure of a roxel is naturally defined through a lattice $\Lambda(\mathbf{L})$, now in three dimensions, over the matrix whose column vectors[1] are $\mathbf{l}_1 = (1,0,1)^\top$, $\mathbf{l}_2 = (0,1,1)^\top$, and $\mathbf{l}_3 = (1,1,0)^\top$. The associated set of Voronoi cells,

$$V(\mathbf{c}) := \{\, \mathbf{r} \in \mathbb{R}^3 \mid \|\mathbf{r}-\mathbf{c}\|_2 < \|\mathbf{r}-\mathbf{p}\|_2 \quad \forall\, \mathbf{p} \in \Lambda(\mathbf{L}) \setminus \{\mathbf{c}\} \,\}, \tag{9.1}$$

then defines the total set of roxels—see Figure 9.1 for a representative image.

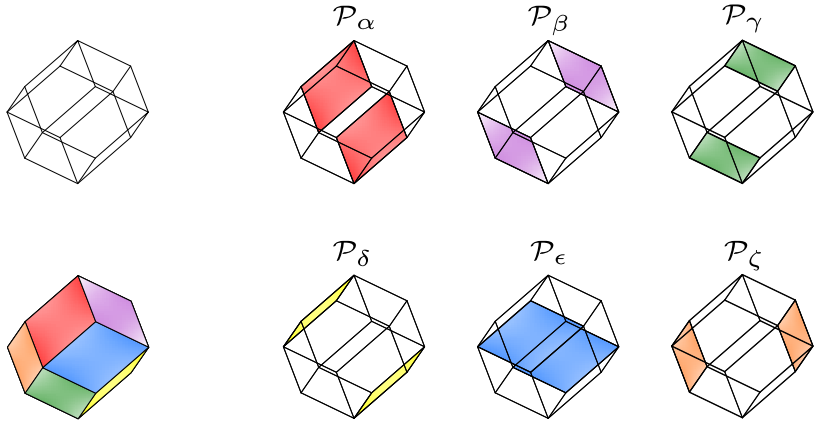


Figure 9.1: A roxel is defined as the volume bounded by six pairs of parallel planes, which we denote by $\mathcal{P}_\alpha, \mathcal{P}_\beta, \ldots, \mathcal{P}_\zeta$. The distance between each pair is the same for all pairs, guaranteeing the isotropy of the roxel element.

[1] Much like the basis vectors for the hexagonal lattice, this choice is non-unique.

9.1 Extending Siddon's algorithm to rhombic dodecahedral voxels

The computational speed-up of the hexagonal extension of Siddon's algorithm extends naturally to three-dimensional voxelisation. The overview of the method in this chapter is similar to the hexagonal pixel approach, and so we will adopt language and notation from Chapter 8. In short, the development for Hex-Siddon can be understood within the following modelling framework:

1. Hexagonal pixels are constructed via the intersection of three pairs of equally-spaced parallel lines;

2. Rays which intersect hexagonal pixels can be interpreted as intersecting distinct sequences of equispaced parallel lines;

3. Pixel-ray intersection lengths can be computed by determining whether a ray intersects a parametrised subset of any given line.

It is with these three notions that we may extend Siddon's algorithm to our chosen roxel discretisation. Where relevant, we will borrow language from Chapter 8 when it carries the same meaning and the context is unambiguous.

9.1.1 Method for rhombic dodecahedral voxel grids

We begin by assuming that there exists a compact three-dimensional volume which can be discretised using non-overlapping roxels. With respect to this volume, our aim is to compute the radiological path given by (8.3) from a point $P_1 = (x_1, y_1, z_1)^\top$ to a point $P_2 = (x_2, y_2, z_2)^\top$, such that the path has length $L = \|P_2 - P_1\|_2$ and is defined parametrically by

$$P(t) = P_1 + \frac{t}{L}(P_2 - P_1) \quad \text{for} \quad t \in [0, L]\,. \tag{9.2}$$

The path first intersects the volume at a distance $t_{\min}$ from P_1 and last intersects the volume at a distance $t_{\max}$ from P_1. As such, the determination of the radiological path concerns only the contributions of intersected roxels along the sub-path defined by $\{P(t) \mid t \in [t_{\min}, t_{\max}]\}$. To describe any particular roxel, we sample pairs of planes from six distinct sets with the following properties:

1. Each set contains (countably) infinitely many uniformly spaced parallel planes;

2. All sets have the same spacing between neighbouring planes;

3. The orientation of each set is specified by an associated normal vector.

As a reference example, we can identify the six sets—and therefore the faces shown in Figure 9.1—with the normal vectors presented in Table A. Similar to the approach in Chapter 8, this approach is invariant to isometries. We denote the six sets by $\mathcal{P}_\alpha, \mathcal{P}_\beta, \ldots, \mathcal{P}_\zeta$, with each associated to the respective vector $\mathbf{n}_\alpha, \mathbf{n}_\beta, \ldots, \mathbf{n}_\zeta$. For example, $\mathcal{P}_\alpha$ will only contain planes of the form

$$\{\mathbf{r} \in \mathbb{R}^3 \mid \langle \mathbf{r}, \mathbf{n}_\alpha \rangle = \rho\}, \tag{9.3}$$

for certain values of $\rho \in \mathbb{R}$.

Set of Planes	Normal Vector	Index Shift
$\mathcal{P}_\alpha$	$\mathbf{n}_\alpha = (1,0,1)^\top$	$\pm[\![1,0,-1,0]\!]$
$\mathcal{P}_\beta$	$\mathbf{n}_\beta = (0,1,1)^\top$	$\pm[\![1,-1,0,0]\!]$
$\mathcal{P}_\gamma$	$\mathbf{n}_\gamma = (-1,0,1)^\top$	$\pm[\![0,1,0,-1]\!]$
$\mathcal{P}_\delta$	$\mathbf{n}_\delta = (0,-1,1)^\top$	$\pm[\![0,0,1,-1]\!]$
$\mathcal{P}_\epsilon$	$\mathbf{n}_\epsilon = (1,1,0)^\top$	$\pm[\![1,0,0,-1]\!]$
$\mathcal{P}_\zeta$	$\mathbf{n}_\zeta = (-1,1,0)^\top$	$\pm[\![0,1,-1,0]\!]$

Table A: Normal vectors and index shifts for each roxel face normal.

We say that the roxel grid has a size[2] of $N_x \times N_y \times N_z$, where N_z is the number of vertically aligned roxels in a column, while N_x and N_y define the number of distinct columns along either direction of the discretised volume—for example, the grid shown in Figure 9.2 has a size of $5 \times 5 \times 3$. This convention is not related to the indexing system used for the roxel grid, which uses a quartic coordinate system, extending the cubic system for a hexagonal grid.

A quartic coordinate system $[\![i,j,k,l]\!]$ is constrained such that $i+j+k+l=0$, and is defined with respect to an 'origin' roxel which has index $[\![0,0,0,0]\!]$ and centre $\mathbf{r}_0 = (x_0, y_0, z_0)^\top \in \mathbb{R}^3$. When computing the radiological path, we state that the path first intersects the voxel $[\![i_{\min}, j_{\min}, k_{\min}, l_{\min}]\!]$ and last intersects the voxel $[\![i_{\max}, j_{\max}, k_{\max}, l_{\max}]\!]$.

To compute the radiological path, we therefore need to determine the intersected roxel indices $[\![i,j,k,l]\!]$, and the associated intersection lengths $\ell_{i,j,k,l}$. In this approach, we will compute distinct sequences of uniform distances for the sets $\mathcal{P}_\alpha, \mathcal{P}_\beta, \ldots, \mathcal{P}_\zeta$

[2]As with the hexagonal case, we are free to inflate the size of a grid so that it has this property.

separately, before merging them afterward. All roxels have twelve rhombic faces, each of which is defined by a short face-diagonal r_1 and a long face-diagonal $r_2 = \sqrt{2}r_1$. This means that for any chosen set, the distance between two neighbouring parallel planes is $R := r_1/\sqrt{2}$. Without loss of generality, we may assume that P_1 and P_2 are each within a distance $R/2$ of the roxel grid—this can be ensured by projecting both points along the path onto the smallest cuboidal hull which bounds the grid itself.

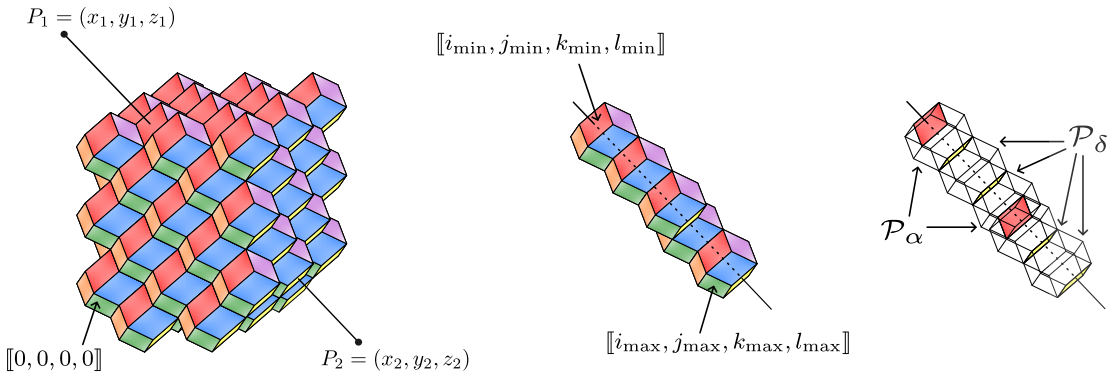


Figure 9.2: The path intersects a sequence of roxel faces that lie on planes from the sets $\mathcal{P}_\alpha, \mathcal{P}_\beta, \ldots, \mathcal{P}_\zeta$. In this particular example, the path only intersects members of $\mathcal{P}_\alpha$ (red) and $\mathcal{P}_\delta$ (yellow).

To have a clear example[3] of our setup, in Figure 9.3 we show how the path intersects an ordered sequence of planes from the set $\mathcal{P}_\alpha$. Here, all intersections are equally spaced by a distance $R|\sec(\phi)|$, where ϕ is the vector angle[4] that the path makes with the normal vector of all planes in $\mathcal{P}_\alpha$. It is clear that the path intersects only planes that separate the points P_1 and P_2, i.e. planes for which there exist $t \in [0, L]$ such that $P(t)$ lies on them. Similarly, it is also clear that while these intersections exist, they do not necessarily correspond to a face intersection that occurs in the roxel discretisation.

Our approach is therefore analogous to the hexagonal case—the problem of computing which faces are successively intersected is reframed as a problem of computing which intersections with planes from $\mathcal{P}_\alpha$ correspond to a true roxel face, i.e. those faces highlighted in the example in Figure 9.2. Intersections that correspond to a true

[3]The example shown in Figures 9.2 and 9.3 looks at only eight members of $\mathcal{P}_\alpha$—this is for illustrative purposes, as the number of members depends on the size of the roxel grid and the relative positions of P_1 and P_2.

[4]In Section 8.4.1 the angle was taken to be between the ray and the direction of the support line. In three dimensions it is more useful to take the angle made with the normal instead, resulting in the use of a $\sec(\cdot)$ term. Additionally, we use the absolute value of the $\sec(\cdot)$ term, to ensure that the spacing is positive regardless of the orientation of the path.

roxel boundary will be called genuine, whereas intersections that do not will be called spurious. All planes in $\mathcal{P}_\alpha$ have the same unit normal vector $\widehat{\mathbf{n}} = \frac{1}{\sqrt{2}}(1,0,1)^\top$, and can therefore be uniquely parametrised by

$$\Pi_m := \left\{ \mathbf{r} = (x,y,z)^\top \in \mathbb{R}^3 \mid (\mathbf{r} - \mathbf{r}_0) \cdot \widehat{\mathbf{n}} = mR \right\}, \tag{9.4}$$

for all $m \in \mathbb{Z}$. In this way, the plane which passes through the centre of the $[\![0,0,0,0]\!]$ roxel is given the label Π_0, and all other planes may be described relative to it.

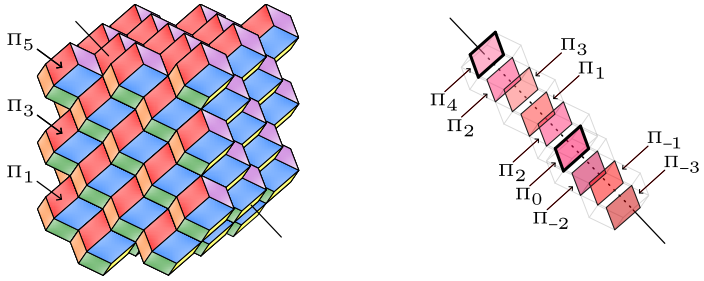


Figure 9.3: In this example, the genuine intersections occur with the planes Π_4 and Π_0 (in order), which we denote with a bolder outline around the intersected face. While all faces are from the same set $\mathcal{P}_\alpha$, we have used two hues of red to differentiate the staggering of planes by their alternating subscript parity, with pinker shades representing even-numbered planes[5]. Note that the plane Π_2 is represented twice, due to the staggering of neighbouring roxels—it acts as the 'top' of one roxel and the 'base' of another. The extension of each plane Π_M means that the ray in this example will intersect the set of planes $\{\Pi_M \mid M = -2,-1,\ldots,3,4\}$, while inside the grid.

Starting from the point P_1, we denote the distance to the first intersected plane by t_0, so that the path intersects some plane $\Pi_M \in \mathcal{P}_\alpha$ (for some M not yet determined) at the location $P(t_0)$. The uniform spacing of members of $\mathcal{P}_\alpha$ guarantees that distances between successive intersections are identical, and all subsequent distances are then

$$t_k = t_0 + kR|\sec(\phi)|, \tag{9.5}$$

for intersection locations[6] $P(t_k) \in \Pi_{M\pm k}$.

Looking at members of $\mathcal{P}_\alpha$, we find that the locations of genuine roxel faces follow a regular pattern determined by the parity of the plane subscript K. Projecting along the direction $-\widehat{\mathbf{n}}$, genuine face locations are seen to occupy a fixed region of an

[5]The colouring is intended to make clear that roxels from neighbouring columns are staggered, while the opacity of the faces increases for lower-indexed planes.

[6]As with the Hex-Siddon algorithm, the relationship between the subscript of t_k and the intersected plane $\Pi_{M\pm k}$ depends on the relative alignment of the vectors $(P_2 - P_1)$ and $(\mathbf{r}_0 - P_1)$, which is the cause for the necessary '$\pm$'.

associated rectangle—see Figure 9.4. Knowing that the path travels a fixed distance between planes, the projected distance between successive intersections along the plane is also constant. Therefore, we can find the discrete points on each plane to determine whether a face was intersected.

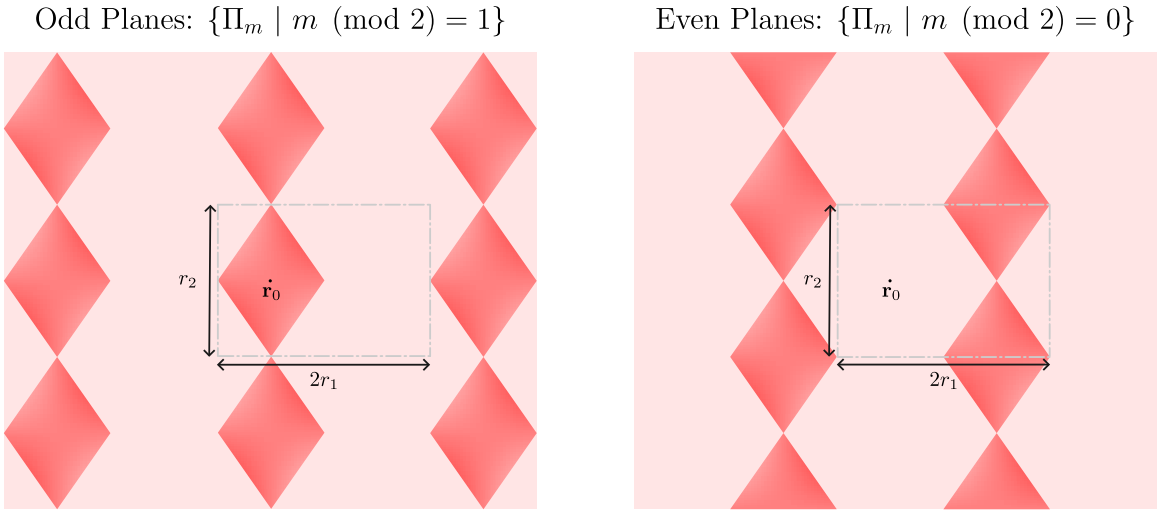


Figure 9.4: A projection of 'odd' and 'even' planes, viewed along the direction $\mathbf{n} = (-1, 0, -1)^\top$. For both cases, the locations of rhombic faces exist in a regular pattern, described by a tessellating rectangle of size $2r_1 \times r_2$. The projected location of $\mathbf{r}_0$ is shown in each figure, from which the location of all planar intersections can then be computed.

The planes in $\mathcal{P}_\alpha$ are parallel to the plane spanned by $\widehat{\mathbf{v}}_1 = (0, 1, 0)^\top$ and $\widehat{\mathbf{v}}_2 = \frac{1}{\sqrt{2}}(-1, 0, 1)^\top$, and so the path direction unit vector $\widehat{\mathbf{p}} = (P_2 - P_1)/L$ may be decomposed by the right-handed orthogonal basis $\{\widehat{\mathbf{v}}_1, \widehat{\mathbf{v}}_2, \widehat{\mathbf{n}}\}$, such that

$$\widehat{\mathbf{p}} = a\widehat{\mathbf{v}}_1 + b\widehat{\mathbf{v}}_2 + c\widehat{\mathbf{n}}. \tag{9.6}$$

Since successive planes in $\mathcal{P}_\alpha$ are separated by a distance R along the normal direction, between intersections the path travels

$$d_1 = \frac{aR}{c}, \qquad d_2 = \frac{bR}{c}, \tag{9.7}$$

along directions $\widehat{\mathbf{v}}_1$ and $\widehat{\mathbf{v}}_2$, respectively.

The location of the first intersection, relative to $\mathbf{r}_0$, can be decomposed via the same basis to find the initial relative coordinates $(P(t_0) - \mathbf{r}_0) = q_1\widehat{\mathbf{v}}_1 + q_2\widehat{\mathbf{v}}_2$. Therefore,

each planar intersection occurs[7] at the point

$$\mathbf{s}_k = (x_k,\ y_k)^\top, \tag{9.8}$$

$$= (q_1, q_2)^\top \pm k(d_1, d_2)^\top, \tag{9.9}$$

for $k \in \mathbb{N}$, with respect to the basis $\{\widehat{\mathbf{v}}_1, \widehat{\mathbf{v}}_2\}$.

In Figure 9.5 we show the projection of the example path in Figure 9.2 onto odd and even planes, with crosses denoting the location of a spurious intersection and circles denoting genuine intersections. As expected, the intersections between planes of the same parity occur at regular intervals and, in this example, the spacing is large enough to skip a genuine face on an odd-numbered plane. This is why in Figure 9.2 not all roxel faces are intersected.

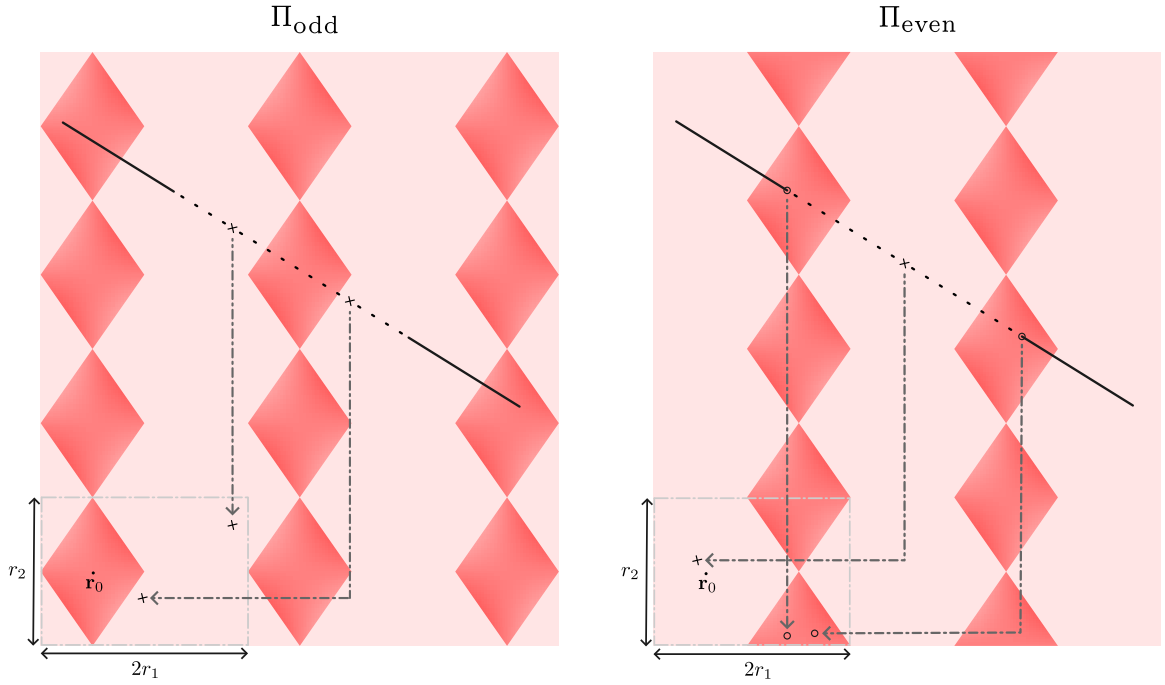


Figure 9.5: Intersection locations can be remapped under modulo operations to a canonical rectangle, defined in relation to the origin centre $\mathbf{r}_0$, from which it is then simple to compute whether an intersection is genuine or spurious. To match the diagram in Figure 9.3, the dashed diagonal line represents the internal length travelled between the intersected planes from $\mathcal{P}_\alpha$. The intersected planes are Π_4 and Π_0, and so both genuine intersections exist on the 'even' plane, as expected—genuine intersections are depicted by circles, while spurious intersections are depicted by crosses. These figures should be viewed as the roxel analogue to Figure 8.12.

[7]Again, the use of $\pm$ in the definition of $\mathbf{s}_k$ is related to the relative alignment of the vectors $(P_2 - P_1)$ and $(\mathbf{r}_0 - P_1)$.

Due to the regular tessellation of the planes, determining whether an intersection is genuine can be performed by looking at the relative location within a rectangle of size $2r_1 \times r_2$. Therefore, to determine whether an intersection $\mathbf{s}_k \in \Pi_K$ is genuine we first compute the remapped coordinates $\tilde{\mathbf{s}}_k = (\tilde{x}_k,\ \tilde{y}_k)^\top$ defined by

$$\tilde{\mathbf{s}}_k = \left(\left(x_k + \frac{r_1}{2} \right) \pmod{2r_1} - \frac{r_1}{2},\ \left(y_k + \frac{r_2}{2} \right) \pmod{r_2} - \frac{r_2}{2} \right)^\top \tag{9.10}$$

if K is odd, and

$$\tilde{\mathbf{s}}_k = \left(\left(x_k - \frac{r_1}{2} \right) \pmod{2r_1} - \frac{r_1}{2},\ y_k \pmod{r_2} - \frac{r_2}{2} \right)^\top \tag{9.11}$$

if K is even, from which we compute whether $(\tilde{x}_k,\ \tilde{y}_k)^\top$ lies within the rhombic subset—see Figure 9.6.

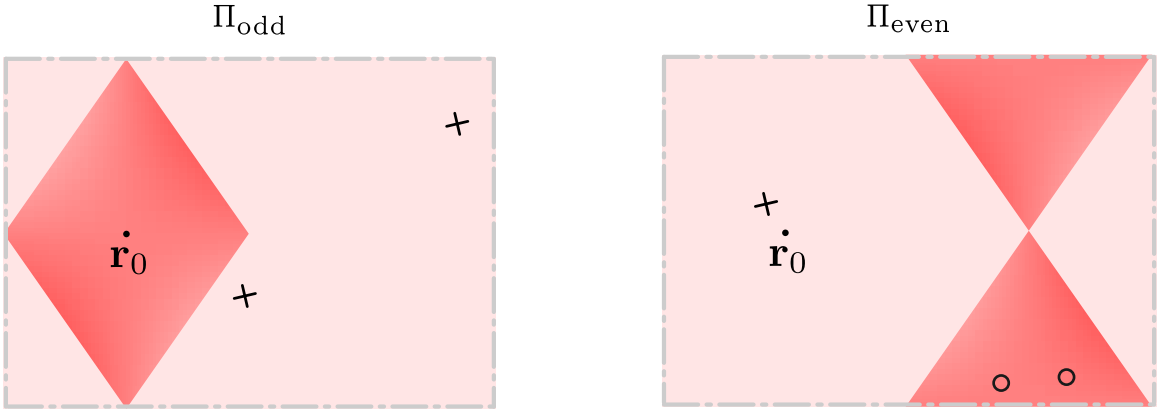


Figure 9.6: Whether an intersection is genuine or spurious can be determined by the location of its projection onto either Π_{odd}, or Π_{even}, where the choice of 'odd' or 'even' depends on the parity of K for which $s_k \in \Pi_K$. Genuine intersections that are 'odd' fall in the rhombus in the left figure, while genuine intersections that are 'even' fall in the 'split' rhombus in the right figure. Any projections that fall outside of these regions are spurious.

Exploiting the regularity of the tessellation, to determine whether an intersection lies within the rhombic area we simply need to compute the 1-norm of the vector $\Theta_k := (2\tilde{x}_k/r_1,\ 2\tilde{y}_k/r_2)^\top$, denoting

$$\sigma_k = \begin{cases} 1 & \text{if } \|\Theta_k\|_1 \leq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{9.12}$$

Here, a value of 1 corresponds to a genuine intersection, and a value of 0 corresponds to a spurious intersection. As with the implementation of Hex-Siddon, the value of σ_k is used as a direct multiplier for the intersection t_k, so that spurious intersections are zeroed out in the computations, leaving only the set of genuine distances $\{ t_k \}_{\text{gen}}$.

Denoting the genuine distances obtained with respect to the set $\mathcal{P}_\alpha$ by $\{\, t_k^\alpha \,\}_{\text{gen}}$, an identical process can be carried out independently for each of the sets $\mathcal{P}_\beta, \ldots, \mathcal{P}_\zeta$, forming the total set of genuine intersection distances

$$\mathcal{R} = \{\, t_i^\alpha \,\}_{\text{gen}} \cup \{\, t_j^\beta \,\}_{\text{gen}} \cup \{\, t_k^\gamma \,\}_{\text{gen}} \cup \{\, t_l^\delta \,\}_{\text{gen}} \cup \{\, t_m^\epsilon \,\}_{\text{gen}} \cup \{\, t_n^\zeta \,\}_{\text{gen}}. \tag{9.13}$$

This set can then be sorted into a t-parametrised set

$$\mathcal{R}_{\text{sort}} = \{t_0, t_1, \ldots, t_k, t_{k+1}\} \tag{9.14}$$

that contains the successive distances that contribute to computing the radiological path. To sort this set we, again, use K-way merging, meaning that the sort algorithm has a time complexity of $\mathcal{O}(\log(6)k) = \mathcal{O}(k)$ and is therefore linear in the number of intersections that a ray makes with the roxel grid. The intersection lengths that make up the total path are then simply

$$\ell_m = (t_m - t_{m-1}) \quad \text{for } m = 1, \ldots, k+1. \tag{9.15}$$

Finally, for the complete determination of the radiological path it is also necessary to compute the indices of each intersected roxel—this follows an identical approach to the hexagonal case, and is discussed at the end of the next section.

9.1.2 Rox-Siddon: Algorithmic details

In this section we outline the necessary computations of this algorithm, organised into the same stage structure as the hexagonal algorithm. For simplicity, we take the planes to be oriented as given by Table A, although all computations hold for arbitrary grid orientations.

Stages 0 and I: Initialisation

These stages are identical to those described in Chapter 8, in that we project P_1 and P_2 onto bounding boxes for the roxel grid, finding new points Q_1 and Q_2. This is achieved by computing the intersection of the ray with each face of the bounding box. With probability zero, Q_1 or Q_2 will coincide with a roxel vertex, and so the projected points will be within a single roxel inradius of the grid.

Stage II: Compute distances

We denote the unit normal vectors for the sets $\mathcal{P}_\alpha, \mathcal{P}_\beta, \ldots, \mathcal{P}_\zeta$ by

$$\widehat{\mathbf{n}}_\alpha = (1,0,1)^\top/\sqrt{2}, \qquad \widehat{\mathbf{n}}_\beta = (0,1,1)^\top/\sqrt{2}, \tag{9.16a}$$

$$\widehat{\mathbf{n}}_\gamma = (-1,0,1)^\top/\sqrt{2}, \qquad \widehat{\mathbf{n}}_\delta = (0,-1,1)^\top/\sqrt{2}, \tag{9.16b}$$

$$\widehat{\mathbf{n}}_\epsilon = (1,1,0)^\top/\sqrt{2}, \qquad \widehat{\mathbf{n}}_\zeta = (-1,1,0)^\top/\sqrt{2}. \tag{9.16c}$$

We then compute the angles made between the path and all normal vectors,

$$\phi_\alpha = \arccos\left(\langle \widehat{\mathbf{p}},\ \widehat{\mathbf{n}}_\alpha\rangle\right), \qquad \phi_\beta = \arccos\left(\langle \widehat{\mathbf{p}},\ \widehat{\mathbf{n}}_\beta\rangle\right), \tag{9.17a}$$

$$\phi_\gamma = \arccos\left(\langle \widehat{\mathbf{p}},\ \widehat{\mathbf{n}}_\gamma\rangle\right), \qquad \phi_\delta = \arccos\left(\langle \widehat{\mathbf{p}},\ \widehat{\mathbf{n}}_\delta\rangle\right), \tag{9.17b}$$

$$\phi_\epsilon = \arccos\left(\langle \widehat{\mathbf{p}},\ \widehat{\mathbf{n}}_\epsilon\rangle\right), \qquad \phi_\zeta = \arccos\left(\langle \widehat{\mathbf{p}},\ \widehat{\mathbf{n}}_\zeta\rangle\right), \tag{9.17c}$$

where $\widehat{\mathbf{p}} = (Q_2 - Q_1)/L$ is the unit direction of the path and the principal angles are taken to be in the interval $[0, \pi)$.

To determine the number of planes passed through each set from the centre of roxel $[\![0,0,0,0]\!]$ to Q_1 we first compute the orthogonal distances

$$q_\alpha = \langle (Q_1 - \mathbf{r}_0),\, \widehat{\mathbf{n}}_\alpha\rangle\,, \qquad q_\beta = \langle (Q_1 - \mathbf{r}_0),\, \widehat{\mathbf{n}}_\beta\rangle\,, \tag{9.18a}$$

$$q_\gamma = \langle (Q_1 - \mathbf{r}_0),\, \widehat{\mathbf{n}}_\gamma\rangle\,, \qquad q_\delta = \langle (Q_1 - \mathbf{r}_0),\, \widehat{\mathbf{n}}_\delta\rangle\,, \tag{9.18b}$$

$$q_\epsilon = \langle (Q_1 - \mathbf{r}_0),\, \widehat{\mathbf{n}}_\epsilon\rangle\,, \qquad q_\zeta = \langle (Q_1 - \mathbf{r}_0),\, \widehat{\mathbf{n}}_\zeta\rangle\,, \tag{9.18c}$$

and take the signed integer proportion of each, with respect to the roxel inradius,

$$N_\alpha = \operatorname{sgn}(q_\alpha)\,\lfloor |q_\alpha|/R \rfloor\,, \qquad N_\beta = \operatorname{sgn}(q_\beta)\,\lfloor |q_\beta|/R \rfloor\,, \tag{9.19a}$$

$$N_\gamma = \operatorname{sgn}(q_\gamma)\,\lfloor |q_\gamma|/R \rfloor\,, \qquad N_\delta = \operatorname{sgn}(q_\delta)\,\lfloor |q_\delta|/R \rfloor\,, \tag{9.19b}$$

$$N_\epsilon = \operatorname{sgn}(q_\epsilon)\,\lfloor |q_\epsilon|/R \rfloor\,, \qquad N_\zeta = \operatorname{sgn}(q_\zeta)\,\lfloor |q_\zeta|/R \rfloor\,. \tag{9.19c}$$

The sets $\mathcal{P}_\alpha, \ldots, \mathcal{P}_\zeta$ have the following forms:

$$\mathcal{P}_\alpha = \left\{\mathbf{r} \in \mathbb{R}^3 \mid \langle (\mathbf{r} - \mathbf{r}_0),\, \widehat{\mathbf{n}}_\alpha\rangle = mR\right\}, \tag{9.20a}$$

$$\mathcal{P}_\beta = \left\{\mathbf{r} \in \mathbb{R}^3 \mid \langle (\mathbf{r} - \mathbf{r}_0),\, \widehat{\mathbf{n}}_\beta\rangle = mR\right\}, \tag{9.20b}$$

$$\mathcal{P}_\gamma = \left\{\mathbf{r} \in \mathbb{R}^3 \mid \langle (\mathbf{r} - \mathbf{r}_0),\, \widehat{\mathbf{n}}_\gamma\rangle = mR\right\}, \tag{9.20c}$$

$$\mathcal{P}_\delta = \left\{\mathbf{r} \in \mathbb{R}^3 \mid \langle (\mathbf{r} - \mathbf{r}_0),\, \widehat{\mathbf{n}}_\delta\rangle = mR\right\}, \tag{9.20d}$$

$$\mathcal{P}_\epsilon = \left\{\mathbf{r} \in \mathbb{R}^3 \mid \langle (\mathbf{r} - \mathbf{r}_0),\, \widehat{\mathbf{n}}_\epsilon\rangle = mR\right\}, \tag{9.20e}$$

$$\mathcal{P}_\zeta = \left\{\mathbf{r} \in \mathbb{R}^3 \mid \langle (\mathbf{r} - \mathbf{r}_0),\, \widehat{\mathbf{n}}_\zeta\rangle = mR\right\}, \tag{9.20f}$$

for all $m \in \mathbb{Z}$. For each set, we denote the planes associated with $K \in \mathbb{Z}$ by $\Pi^{\alpha}_{K}, \ldots, \Pi^{\zeta}_{K}$, respectively. The point Q_1 is then bounded by the twelve planes

$$\{\Pi^{\alpha}_{N_\alpha}, \Pi^{\alpha}_{N_\alpha+\mathrm{sgn}(q_\alpha)}, \Pi^{\beta}_{N_\beta}, \Pi^{\beta}_{N_\beta+\mathrm{sgn}\left(q_\beta\right)}, \Pi^{\gamma}_{N_\gamma}, \Pi^{\gamma}_{N_\gamma+\mathrm{sgn}(q_\gamma)}, \ldots \\ \Pi^{\delta}_{N_\delta}, \Pi^{\delta}_{N_\delta+\mathrm{sgn}(q_\delta)}, \Pi^{\epsilon}_{N_\epsilon}, \Pi^{\epsilon}_{N_\epsilon+\mathrm{sgn}(q_\epsilon)}, \Pi^{\zeta}_{N_\zeta}, \Pi^{\zeta}_{N_\zeta+\mathrm{sgn}\left(q_\zeta\right)}\}. \tag{9.21}$$

The relative orientation of the vectors $(Q_1 - \mathbf{r}_0)$ and $(Q_2 - Q_1)$ determines which plane (from each family) will be crossed first by the path, so we identify the first-intersected planes by the subscripts

$$N^*_\alpha = \begin{cases} \min\{N_\alpha,\ N_\alpha + \mathrm{sgn}\,(q_\alpha)\} & \text{if } \langle (Q_2 - Q_1),\, \widehat{\mathbf{n}}_\alpha \rangle < 0, \\ \max\{N_\alpha,\ N_\alpha + \mathrm{sgn}\,(q_\alpha)\} & \text{otherwise,} \end{cases} \tag{9.22a}$$

$$N^*_\beta = \begin{cases} \min\{N_\beta,\ N_\beta + \mathrm{sgn}\,(q_\beta)\} & \text{if } \langle (Q_2 - Q_1),\, \widehat{\mathbf{n}}_\beta \rangle < 0, \\ \max\{N_\beta,\ N_\beta + \mathrm{sgn}\,(q_\beta)\} & \text{otherwise,} \end{cases} \tag{9.22b}$$

$$N^*_\gamma = \begin{cases} \min\{N_\gamma,\ N_\gamma + \mathrm{sgn}\,(q_\gamma)\} & \text{if } \langle (Q_2 - Q_1),\, \widehat{\mathbf{n}}_\gamma \rangle < 0, \\ \max\{N_\gamma,\ N_\gamma + \mathrm{sgn}\,(q_\gamma)\} & \text{otherwise,} \end{cases} \tag{9.22c}$$

$$N^*_\delta = \begin{cases} \min\{N_\delta,\ N_\delta + \mathrm{sgn}\,(q_\delta)\} & \text{if } \langle (Q_2 - Q_1),\, \widehat{\mathbf{n}}_\delta \rangle < 0, \\ \max\{N_\delta,\ N_\delta + \mathrm{sgn}\,(q_\delta)\} & \text{otherwise,} \end{cases} \tag{9.22d}$$

$$N^*_\epsilon = \begin{cases} \min\{N_\epsilon,\ N_\epsilon + \mathrm{sgn}\,(q_\epsilon)\} & \text{if } \langle (Q_2 - Q_1),\, \widehat{\mathbf{n}}_\epsilon \rangle < 0, \\ \max\{N_\epsilon,\ N_\epsilon + \mathrm{sgn}\,(q_\epsilon)\} & \text{otherwise,} \end{cases} \tag{9.22e}$$

$$N^*_\zeta = \begin{cases} \min\{N_\zeta,\ N_\zeta + \mathrm{sgn}\,(q_\zeta)\} & \text{if } \langle (Q_2 - Q_1),\, \widehat{\mathbf{n}}_\zeta \rangle < 0, \\ \max\{N_\zeta,\ N_\zeta + \mathrm{sgn}\,(q_\zeta)\} & \text{otherwise.} \end{cases} \tag{9.22f}$$

As such, the first-intersected planes are $\Pi^{\alpha}_{N^*_\alpha}, \Pi^{\beta}_{N^*_\beta}, \ldots, \Pi^{\zeta}_{N^*_\zeta}$, and their respective intersection locations[8] are

$$\mathbf{d}^{\alpha}_{N^*_\alpha} = Q_1 + \left(\frac{N^*_\alpha R - \langle (Q_1 - \mathbf{r}_0), \widehat{\mathbf{n}}_\alpha \rangle}{\langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_\alpha \rangle} \right) \widehat{\mathbf{p}}, \tag{9.23a}$$

$$\mathbf{d}^{\beta}_{N^*_\beta} = Q_1 + \left(\frac{N^*_\beta R - \langle (Q_1 - \mathbf{r}_0), \widehat{\mathbf{n}}_\beta \rangle}{\langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_\beta \rangle} \right) \widehat{\mathbf{p}}, \tag{9.23b}$$

$$\mathbf{d}^{\gamma}_{N^*_\gamma} = Q_1 + \left(\frac{N^*_\gamma R - \langle (Q_1 - \mathbf{r}_0), \widehat{\mathbf{n}}_\gamma \rangle}{\langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_\gamma \rangle} \right) \widehat{\mathbf{p}}, \tag{9.23c}$$

$$\mathbf{d}^{\delta}_{N^*_\delta} = Q_1 + \left(\frac{N^*_\delta R - \langle (Q_1 - \mathbf{r}_0), \widehat{\mathbf{n}}_\delta \rangle}{\langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_\delta \rangle} \right) \widehat{\mathbf{p}}, \tag{9.23d}$$

$$\mathbf{d}^{\epsilon}_{N^*_\epsilon} = Q_1 + \left(\frac{N^*_\epsilon R - \langle (Q_1 - \mathbf{r}_0), \widehat{\mathbf{n}}_\epsilon \rangle}{\langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_\epsilon \rangle} \right) \widehat{\mathbf{p}}, \tag{9.23e}$$

$$\mathbf{d}^{\zeta}_{N^*_\zeta} = Q_1 + \left(\frac{N^*_\zeta R - \langle (Q_1 - \mathbf{r}_0), \widehat{\mathbf{n}}_\zeta \rangle}{\langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_\zeta \rangle} \right) \widehat{\mathbf{p}}. \tag{9.23f}$$

[8] Assuming that the path is not parallel to the plane.

The distances from Q_1 to each first intersection are

$$t_0^\alpha = \|Q_1 - \mathbf{d}_{N_\alpha^*}^\alpha\|_2, \qquad t_0^\beta = \|Q_1 - \mathbf{d}_{N_\beta^*}^\beta\|_2, \qquad t_0^\gamma = \|Q_1 - \mathbf{d}_{N_\gamma^*}^\gamma\|_2, \tag{9.24a}$$

$$t_0^\delta = \|Q_1 - \mathbf{d}_{N_\delta^*}^\delta\|_2, \qquad t_0^\epsilon = \|Q_1 - \mathbf{d}_{N_\epsilon^*}^\epsilon\|_2, \qquad t_0^\zeta = \|Q_1 - \mathbf{d}_{N_\zeta^*}^\zeta\|_2. \tag{9.24b}$$

We then compute

$$T_\alpha = \left\lfloor \frac{L - t_0^\alpha}{R|\sec(\phi_\alpha)|} \right\rfloor, \qquad T_\beta = \left\lfloor \frac{L - t_0^\beta}{R|\sec(\phi_\beta)|} \right\rfloor, \qquad T_\gamma = \left\lfloor \frac{L - t_0^\gamma}{R|\sec(\phi_\gamma)|} \right\rfloor, \tag{9.25a}$$

$$T_\delta = \left\lfloor \frac{L - t_0^\delta}{R|\sec(\phi_\delta)|} \right\rfloor, \qquad T_\epsilon = \left\lfloor \frac{L - t_0^\epsilon}{R|\sec(\phi_\epsilon)|} \right\rfloor, \qquad T_\zeta = \left\lfloor \frac{L - t_0^\zeta}{R|\sec(\phi_\zeta)|} \right\rfloor, \tag{9.25b}$$

which give the total number of planes crossed in each set before reaching the end of the path. It is then trivial to compute successive intersection distances for each set

$$t_k^\alpha = t_0^\alpha + kR|\sec(\phi_\alpha)| \qquad \text{for } k = 1, \ldots, T_\alpha, \tag{9.26a}$$

$$t_k^\beta = t_0^\beta + kR|\sec(\phi_\beta)| \qquad \text{for } k = 1, \ldots, T_\beta, \tag{9.26b}$$

$$t_k^\gamma = t_0^\gamma + kR|\sec(\phi_\gamma)| \qquad \text{for } k = 1, \ldots, T_\gamma, \tag{9.26c}$$

$$t_k^\delta = t_0^\delta + kR|\sec(\phi_\delta)| \qquad \text{for } k = 1, \ldots, T_\delta, \tag{9.26d}$$

$$t_k^\epsilon = t_0^\epsilon + kR|\sec(\phi_\epsilon)| \qquad \text{for } k = 1, \ldots, T_\epsilon, \tag{9.26e}$$

$$t_k^\zeta = t_0^\zeta + kR|\sec(\phi_\zeta)| \qquad \text{for } k = 1, \ldots, T_\zeta. \tag{9.26f}$$

The total set

$$\{t_0^\alpha, \ldots, t_{T_\alpha}^\alpha, t_0^\beta, \ldots, t_{T_\beta}^\beta, t_0^\gamma, \ldots, t_{T_\gamma}^\gamma, t_0^\delta, \ldots, t_{T_\delta}^\delta, t_0^\epsilon, \ldots, t_{T_\epsilon}^\epsilon, t_0^\zeta, \ldots, t_{T_\zeta}^\zeta\} \tag{9.27}$$

contains the total set of intersection distances for all lines, both genuine and spurious.

Stage III: Determine genuine distances

For each set, we define the 'short-diagonal' and 'long-diagonal' vectors which, alongside their respective associated normal vectors, form right-handed orthogonal bases for each set:

$$\widehat{\mathbf{v}}_1^\alpha = (0,1,0)^\top, \qquad \widehat{\mathbf{v}}_2^\alpha = (-1,0,1)^\top/\sqrt{2}, \tag{9.28a}$$

$$\widehat{\mathbf{v}}_1^\beta = (-1,0,0)^\top, \qquad \widehat{\mathbf{v}}_2^\beta = (0,-1,1)^\top/\sqrt{2}, \tag{9.28b}$$

$$\widehat{\mathbf{v}}_1^\gamma = (0,-1,0)^\top, \qquad \widehat{\mathbf{v}}_2^\gamma = (1,0,1)^\top/\sqrt{2}, \tag{9.28c}$$

$$\widehat{\mathbf{v}}_1^\delta = (1,0,0)^\top, \qquad \widehat{\mathbf{v}}_2^\delta = (0,1,1)^\top/\sqrt{2}, \tag{9.28d}$$

$$\widehat{\mathbf{v}}_1^\epsilon = (0,0,-1)^\top, \qquad \widehat{\mathbf{v}}_2^\epsilon = (-1,1,0)^\top/\sqrt{2}, \tag{9.28e}$$

$$\widehat{\mathbf{v}}_1^\zeta = (0,0,-1)^\top, \qquad \widehat{\mathbf{v}}_2^\zeta = (-1,-1,0)^\top/\sqrt{2}. \tag{9.28f}$$

To compute whether the intersections are genuine, we first compute the set-aligned projection coordinates of the first planar intersections for each set:

$$\mathbf{s}_0^{\alpha} = \big(\langle (P(t_0) - \mathbf{r}_0), \widehat{\mathbf{v}}_1^{\alpha}\rangle,\ \langle (P(t_0) - \mathbf{r}_0), \widehat{\mathbf{v}}_2^{\alpha}\rangle\big)^{\top} \qquad \text{w.r.t } \{\widehat{\mathbf{v}}_1^{\alpha}, \widehat{\mathbf{v}}_2^{\alpha}\}, \tag{9.29a}$$

$$\mathbf{s}_0^{\beta} = \big(\langle (P(t_0) - \mathbf{r}_0), \widehat{\mathbf{v}}_1^{\beta}\rangle,\ \langle (P(t_0) - \mathbf{r}_0), \widehat{\mathbf{v}}_2^{\beta}\rangle\big)^{\top} \qquad \text{w.r.t } \{\widehat{\mathbf{v}}_1^{\beta}, \widehat{\mathbf{v}}_2^{\beta}\}, \tag{9.29b}$$

$$\mathbf{s}_0^{\gamma} = \big(\langle (P(t_0) - \mathbf{r}_0), \widehat{\mathbf{v}}_1^{\gamma}\rangle,\ \langle (P(t_0) - \mathbf{r}_0), \widehat{\mathbf{v}}_2^{\gamma}\rangle\big)^{\top} \qquad \text{w.r.t } \{\widehat{\mathbf{v}}_1^{\gamma}, \widehat{\mathbf{v}}_2^{\gamma}\}, \tag{9.29c}$$

$$\mathbf{s}_0^{\delta} = \big(\langle (P(t_0) - \mathbf{r}_0), \widehat{\mathbf{v}}_1^{\delta}\rangle,\ \langle (P(t_0) - \mathbf{r}_0), \widehat{\mathbf{v}}_2^{\delta}\rangle\big)^{\top} \qquad \text{w.r.t } \{\widehat{\mathbf{v}}_1^{\delta}, \widehat{\mathbf{v}}_2^{\delta}\}, \tag{9.29d}$$

$$\mathbf{s}_0^{\epsilon} = \big(\langle (P(t_0) - \mathbf{r}_0), \widehat{\mathbf{v}}_1^{\epsilon}\rangle,\ \langle (P(t_0) - \mathbf{r}_0), \widehat{\mathbf{v}}_2^{\epsilon}\rangle\big)^{\top} \qquad \text{w.r.t } \{\widehat{\mathbf{v}}_1^{\epsilon}, \widehat{\mathbf{v}}_2^{\epsilon}\}, \tag{9.29e}$$

$$\mathbf{s}_0^{\zeta} = \big(\langle (P(t_0) - \mathbf{r}_0), \widehat{\mathbf{v}}_1^{\zeta}\rangle,\ \langle (P(t_0) - \mathbf{r}_0), \widehat{\mathbf{v}}_2^{\zeta}\rangle\big)^{\top} \qquad \text{w.r.t } \{\widehat{\mathbf{v}}_1^{\zeta}, \widehat{\mathbf{v}}_2^{\zeta}\}. \tag{9.29f}$$

Between successively intersected planes within each set the planar projection of the path travels distances

$$(d_1^{\alpha}, d_2^{\alpha})^{\top} = (\langle \widehat{\mathbf{p}}, \widehat{\mathbf{v}}_1^{\alpha}\rangle R / \langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_{\alpha}\rangle,\ \langle \widehat{\mathbf{p}}, \widehat{\mathbf{v}}_2^{\alpha}\rangle R / \langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_{\alpha}\rangle)^{\top} \qquad \text{w.r.t } \{\widehat{\mathbf{v}}_1^{\alpha}, \widehat{\mathbf{v}}_2^{\alpha}\}, \tag{9.30a}$$

$$(d_1^{\beta}, d_2^{\beta})^{\top} = (\langle \widehat{\mathbf{p}}, \widehat{\mathbf{v}}_1^{\beta}\rangle R / \langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_{\beta}\rangle,\ \langle \widehat{\mathbf{p}}, \widehat{\mathbf{v}}_2^{\beta}\rangle R / \langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_{\beta}\rangle)^{\top} \qquad \text{w.r.t } \{\widehat{\mathbf{v}}_1^{\beta}, \widehat{\mathbf{v}}_2^{\beta}\}, \tag{9.30b}$$

$$(d_1^{\gamma}, d_2^{\gamma})^{\top} = (\langle \widehat{\mathbf{p}}, \widehat{\mathbf{v}}_1^{\gamma}\rangle R / \langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_{\gamma}\rangle,\ \langle \widehat{\mathbf{p}}, \widehat{\mathbf{v}}_2^{\gamma}\rangle R / \langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_{\gamma}\rangle)^{\top} \qquad \text{w.r.t } \{\widehat{\mathbf{v}}_1^{\gamma}, \widehat{\mathbf{v}}_2^{\gamma}\}, \tag{9.30c}$$

$$(d_1^{\delta}, d_2^{\delta})^{\top} = (\langle \widehat{\mathbf{p}}, \widehat{\mathbf{v}}_1^{\delta}\rangle R / \langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_{\delta}\rangle,\ \langle \widehat{\mathbf{p}}, \widehat{\mathbf{v}}_2^{\delta}\rangle R / \langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_{\delta}\rangle)^{\top} \qquad \text{w.r.t } \{\widehat{\mathbf{v}}_1^{\delta}, \widehat{\mathbf{v}}_2^{\delta}\}, \tag{9.30d}$$

$$(d_1^{\epsilon}, d_2^{\epsilon})^{\top} = (\langle \widehat{\mathbf{p}}, \widehat{\mathbf{v}}_1^{\epsilon}\rangle R / \langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_{\epsilon}\rangle,\ \langle \widehat{\mathbf{p}}, \widehat{\mathbf{v}}_2^{\epsilon}\rangle R / \langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_{\epsilon}\rangle)^{\top} \qquad \text{w.r.t } \{\widehat{\mathbf{v}}_1^{\epsilon}, \widehat{\mathbf{v}}_2^{\epsilon}\}, \tag{9.30e}$$

$$(d_1^{\zeta}, d_2^{\zeta})^{\top} = (\langle \widehat{\mathbf{p}}, \widehat{\mathbf{v}}_1^{\zeta}\rangle R / \langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_{\zeta}\rangle,\ \langle \widehat{\mathbf{p}}, \widehat{\mathbf{v}}_2^{\zeta}\rangle R / \langle \widehat{\mathbf{p}}, \widehat{\mathbf{n}}_{\zeta}\rangle)^{\top} \qquad \text{w.r.t } \{\widehat{\mathbf{v}}_1^{\zeta}, \widehat{\mathbf{v}}_2^{\zeta}\}. \tag{9.30f}$$

All projected planar intersections then occur at the points

$$\mathbf{s}_k^{\alpha} := (x_k^{\alpha},\ y_k^{\alpha})^{\top} = \mathbf{s}_0^{\alpha} + k(d_1^{\alpha}, d_2^{\alpha})^{\top} \qquad \text{for } k = 1, \ldots, T_{\alpha}, \tag{9.31a}$$

$$\mathbf{s}_k^{\beta} := (x_k^{\beta},\ y_k^{\beta})^{\top} = \mathbf{s}_0^{\beta} + k(d_1^{\beta}, d_2^{\beta})^{\top} \qquad \text{for } k = 1, \ldots, T_{\beta}, \tag{9.31b}$$

$$\mathbf{s}_k^{\gamma} := (x_k^{\gamma},\ y_k^{\gamma})^{\top} = \mathbf{s}_0^{\gamma} + k(d_1^{\gamma}, d_2^{\gamma})^{\top} \qquad \text{for } k = 1, \ldots, T_{\gamma}, \tag{9.31c}$$

$$\mathbf{s}_k^{\delta} := (x_k^{\delta},\ y_k^{\delta})^{\top} = \mathbf{s}_0^{\delta} + k(d_1^{\delta}, d_2^{\delta})^{\top} \qquad \text{for } k = 1, \ldots, T_{\delta}, \tag{9.31d}$$

$$\mathbf{s}_k^{\epsilon} := (x_k^{\epsilon},\ y_k^{\epsilon})^{\top} = \mathbf{s}_0^{\epsilon} + k(d_1^{\epsilon}, d_2^{\epsilon})^{\top} \qquad \text{for } k = 1, \ldots, T_{\epsilon}, \tag{9.31e}$$

$$\mathbf{s}_k^{\zeta} := (x_k^{\zeta},\ y_k^{\zeta})^{\top} = \mathbf{s}_0^{\zeta} + k(d_1^{\zeta}, d_2^{\zeta})^{\top} \qquad \text{for } k = 1, \ldots, T_{\zeta}, \tag{9.31f}$$

with respect to the planar basis of each set.

In (9.10) and (9.11), the remapped coordinates were written in terms of the subscript parity of the plane being intersected, but they can instead be rewritten with respect

to the parity of the first-intersected plane:

$$\begin{aligned}\tilde{x}_k^\alpha &= \Big(x_k^\alpha + r_1\big((k+N_\alpha^*) \pmod 2\big) - \frac{r_1}{2}\Big) \pmod{2r_1} - \frac{r_1}{2},\\ \tilde{y}_k^\alpha &= \Big(y_k^\alpha + \frac{r_2}{2}\big((k+N_\alpha^*) \pmod 2\big)\Big) \pmod{r_2} - \frac{r_2}{2},\end{aligned} \tag{9.32a}$$

$$\begin{aligned}\tilde{x}_k^\beta &= \Big(x_k^\beta + r_1\big((k+N_\beta^*) \pmod 2\big) - \frac{r_1}{2}\Big) \pmod{2r_1} - \frac{r_1}{2},\\ \tilde{y}_k^\beta &= \Big(y_k^\beta + \frac{r_2}{2}\big((k+N_\beta^*) \pmod 2\big)\Big) \pmod{r_2} - \frac{r_2}{2},\end{aligned} \tag{9.32b}$$

$$\begin{aligned}\tilde{x}_k^\gamma &= \Big(x_k^\gamma + r_1\big((k+N_\gamma^*) \pmod 2\big) - \frac{r_1}{2}\Big) \pmod{2r_1} - \frac{r_1}{2},\\ \tilde{y}_k^\gamma &= \Big(y_k^\gamma + \frac{r_2}{2}\big((k+N_\gamma^*) \pmod 2\big)\Big) \pmod{r_2} - \frac{r_2}{2},\end{aligned} \tag{9.32c}$$

$$\begin{aligned}\tilde{x}_k^\delta &= \Big(x_k^\delta + r_1\big((k+N_\delta^*) \pmod 2\big) - \frac{r_1}{2}\Big) \pmod{2r_1} - \frac{r_1}{2},\\ \tilde{y}_k^\delta &= \Big(y_k^\delta + \frac{r_2}{2}\big((k+N_\delta^*) \pmod 2\big)\Big) \pmod{r_2} - \frac{r_2}{2},\end{aligned} \tag{9.32d}$$

$$\begin{aligned}\tilde{x}_k^\epsilon &= \Big(x_k^\epsilon + r_1\big((k+N_\epsilon^*) \pmod 2\big) - \frac{r_1}{2}\Big) \pmod{2r_1} - \frac{r_1}{2},\\ \tilde{y}_k^\epsilon &= \Big(y_k^\epsilon + \frac{r_2}{2}\big((k+N_\epsilon^*) \pmod 2\big)\Big) \pmod{r_2} - \frac{r_2}{2},\end{aligned} \tag{9.32e}$$

$$\begin{aligned}\tilde{x}_k^\zeta &= \Big(x_k^\zeta + r_1\big((k+N_\zeta^*) \pmod 2\big) - \frac{r_1}{2}\Big) \pmod{2r_1} - \frac{r_1}{2},\\ \tilde{y}_k^\zeta &= \Big(y_k^\zeta + \frac{r_2}{2}\big((k+N_\zeta^*) \pmod 2\big)\Big) \pmod{r_2} - \frac{r_2}{2}.\end{aligned} \tag{9.32f}$$

The scaled position vectors are then

$$\Theta_k^\alpha = (2\tilde{x}_k^\alpha/r_1,\ 2\tilde{y}_k^\alpha/r_2)^\top, \qquad \Theta_k^\beta = (2\tilde{x}_k^\beta/r_1,\ 2\tilde{y}_k^\beta/r_2)^\top, \tag{9.33a}$$

$$\Theta_k^\gamma = (2\tilde{x}_k^\gamma/r_1,\ 2\tilde{y}_k^\gamma/r_2)^\top, \qquad \Theta_k^\delta = (2\tilde{x}_k^\delta/r_1,\ 2\tilde{y}_k^\delta/r_2)^\top, \tag{9.33b}$$

$$\Theta_k^\epsilon = (2\tilde{x}_k^\epsilon/r_1,\ 2\tilde{y}_k^\epsilon/r_2)^\top, \qquad \Theta_k^\zeta = (2\tilde{x}_k^\zeta/r_1,\ 2\tilde{y}_k^\zeta/r_2)^\top. \tag{9.33c}$$

The logical multipliers for identifying genuine intersections are then computed by

$$\sigma_k^\alpha = \begin{cases}1 & \text{if } \|\Theta_k^\alpha\|_1 \le 1,\\ 0 & \text{otherwise,}\end{cases} \qquad \sigma_k^\beta = \begin{cases}1 & \text{if } \|\Theta_k^\beta\|_1 \le 1,\\ 0 & \text{otherwise,}\end{cases} \tag{9.34a}$$

$$\sigma_k^\gamma = \begin{cases}1 & \text{if } \|\Theta_k^\gamma\|_1 \le 1,\\ 0 & \text{otherwise,}\end{cases} \qquad \sigma_k^\delta = \begin{cases}1 & \text{if } \|\Theta_k^\delta\|_1 \le 1,\\ 0 & \text{otherwise,}\end{cases} \tag{9.34b}$$

$$\sigma_k^\epsilon = \begin{cases}1 & \text{if } \|\Theta_k^\epsilon\|_1 \le 1,\\ 0 & \text{otherwise,}\end{cases} \qquad \sigma_k^\zeta = \begin{cases}1 & \text{if } \|\Theta_k^\zeta\|_1 \le 1,\\ 0 & \text{otherwise.}\end{cases} \tag{9.34c}$$

We multiply the intersection distances by their corresponding logical values to form

the set

$$\mathcal{R} = \{t_0^\alpha\sigma_0^\alpha, \ldots, t_{T_\alpha}^\alpha\sigma_{T_\alpha}^\alpha, t_0^\beta\sigma_0^\beta, \ldots, t_{T_\beta}^\beta\sigma_{T_\beta}^\beta, t_0^\gamma\sigma_0^\gamma, \ldots, t_{T_\gamma}^\gamma\sigma_{T_\gamma}^\gamma, \ldots \\ t_0^\delta\sigma_0^\delta, \ldots, t_{T_\delta}^\delta\sigma_{T_\delta}^\delta, t_0^\epsilon\sigma_0^\epsilon, \ldots, t_{T_\epsilon}^\epsilon\sigma_{T_\epsilon}^\epsilon, t_0^\zeta\sigma_0^\zeta, \ldots, t_{T_\zeta}^\zeta\sigma_{T_\zeta}^\zeta\}, \tag{9.35}$$

which contains only genuine intersections and spurious zeros. Removing the zeros, we are then left with the set

$$\mathcal{R}_{\text{gen}} = \{\, t_i^\alpha \,\}_{\text{gen}} \cup \{\, t_j^\beta \,\}_{\text{gen}} \cup \{\, t_k^\gamma \,\}_{\text{gen}} \cup \{\, t_l^\delta \,\}_{\text{gen}} \cup \{\, t_m^\epsilon \,\}_{\text{gen}} \cup \{\, t_n^\zeta \,\}_{\text{gen}}, \tag{9.36}$$

containing only the distances corresponding to genuine face intersections throughout the roxel grid.

Stage IV: Create index shift lists

The set $\mathcal{R}_{\text{gen}}$ contains only genuine intersection distances, so these can be used to calculate the list of roxel index shifts that occur while traversing the path—we refer to the indexing convention presented in Table A.

To account for the directional dependence of the indexing with respect to the particular ray, the signed index shifts are denoted

$$\chi_\alpha = \text{sgn}\left(\langle (Q_2 - Q_1),\, \widehat{\mathbf{n}}_\alpha \rangle\right) [\![1, 0, -1, 0]\!], \tag{9.37a}$$
$$\chi_\beta = \text{sgn}\left(\langle (Q_2 - Q_1),\, \widehat{\mathbf{n}}_\beta \rangle\right) [\![1, -1, 0, 0]\!], \tag{9.37b}$$
$$\chi_\gamma = \text{sgn}\left(\langle (Q_2 - Q_1),\, \widehat{\mathbf{n}}_\gamma \rangle\right) [\![0, 1, 0, -1]\!], \tag{9.37c}$$
$$\chi_\delta = \text{sgn}\left(\langle (Q_2 - Q_1),\, \widehat{\mathbf{n}}_\delta \rangle\right) [\![0, 0, 1, -1]\!], \tag{9.37d}$$
$$\chi_\epsilon = \text{sgn}\left(\langle (Q_2 - Q_1),\, \widehat{\mathbf{n}}_\epsilon \rangle\right) [\![1, 0, 0, -1]\!], \tag{9.37e}$$
$$\chi_\zeta = \text{sgn}\left(\langle (Q_2 - Q_1),\, \widehat{\mathbf{n}}_\zeta \rangle\right) [\![0, 1, -1, 0]\!]. \tag{9.37f}$$

We then create the following sets

$$\mathcal{A} = \{\chi_\alpha,\ \chi_\alpha,\ \ldots,\ \chi_\alpha\}, \qquad \mathcal{B} = \{\chi_\beta,\ \chi_\beta,\ \ldots,\ \chi_\beta\}, \tag{9.38a}$$
$$\mathcal{C} = \{\chi_\gamma,\ \chi_\gamma,\ \ldots,\ \chi_\gamma\}, \qquad \mathcal{D} = \{\chi_\delta,\ \chi_\delta,\ \ldots,\ \chi_\delta\}, \tag{9.38b}$$
$$\mathcal{E} = \{\chi_\epsilon,\ \chi_\epsilon,\ \ldots,\ \chi_\epsilon\}, \qquad \mathcal{F} = \{\chi_\zeta,\ \chi_\zeta,\ \ldots,\ \chi_\zeta\}, \tag{9.38c}$$

such that $|\mathcal{A}| = |\{\, t_i^\alpha \,\}_{\text{gen}}|$, $|\mathcal{B}| = |\{\, t_j^\beta \,\}_{\text{gen}}|$, $|\mathcal{C}| = |\{\, t_k^\gamma \,\}_{\text{gen}}|$, $|\mathcal{D}| = |\{\, t_l^\delta \,\}_{\text{gen}}|$, $|\mathcal{E}| = |\{\, t_m^\epsilon \,\}_{\text{gen}}|$, and $|\mathcal{F}| = |\{\, t_n^\zeta \,\}_{\text{gen}}|$. The set

$$\mathcal{H} = \mathcal{A} \cup \mathcal{B} \cup \mathcal{C} \cup \mathcal{D} \cup \mathcal{E} \cup \mathcal{F} \tag{9.39}$$

has the same number of elements as $\mathcal{R}_{\text{gen}}$, and will be sorted into the same order.

Stage V: Sort distances and compute intersection lengths

The set $\mathcal{R}_{\text{gen}}$ has elements in the range $(0, L)$, with the probability of repeated values being zero, as this would correspond to the path intersecting a voxel vertex or edge, rather than a face. We can therefore sort the elements to create an ordered t-parametrised set

$$\mathcal{R}_{\text{sort}} = \{t_0, t_1, \ldots, t_k, t_{k+1}\}, \tag{9.40}$$

and compute the intersection distances throughout the voxel grid to be

$$\ell_m = (t_m - t_{m-1}) \qquad \text{for} \quad m = 1, \ldots, k+1. \tag{9.41}$$

Stage VI: Sort indices and compute ordered index list

Let $\boldsymbol{\pi}$ be the permutation that maps $\mathcal{R}$ to $\mathcal{R}_{\text{sort}}$, then $\boldsymbol{\pi}$ can be applied to $\mathcal{H}$, without reference to the specific elements of $\mathcal{H}$. The result is a set $\mathcal{H}_{\text{sort}}$ that contains a sequence of index shifts associated with the ordered distances in $\mathcal{R}_{\text{sort}}$.

We constrained the position Q_1 to exist in a boundary roxel, that is, if we extend the grid by one roxel in each direction then Q_1 exists in a roxel with some index[9] $[\![q_1, q_2, q_3, q_4]\!]$. The sequence of elements $\mathcal{H}_{\text{sort}} = \{\chi_0, \chi_1, \ldots, \chi_k, \chi_{k+1}\}$ is then used to find the index of the mth intersected voxel,

$$\begin{aligned}[\![i_m, j_m, k_m, l_m]\!] &= [\![i_{m-1}, j_{m-1}, k_{m-1}, l_{m-1}]\!] + \chi_m, \\ &= [\![q_1, q_2, q_3, q_4]\!] + \sum_{n=1}^{m} \chi_n. \end{aligned} \tag{9.42}$$

Using the notation from (8.3), we now have all the elements necessary to compute the radiological path

$$\mathfrak{L} = \sum_{(i,j,k,l)\,\in\,\Omega} \ell_{i,j,k,l} = \sum_{m=1}^{k+1} \ell_{i_m,j_m,k_m,l_m}, \tag{9.43}$$

for $\Omega := \{(i, j, k, l) \mid \ell_{i,j,k,l} \neq 0\}$.

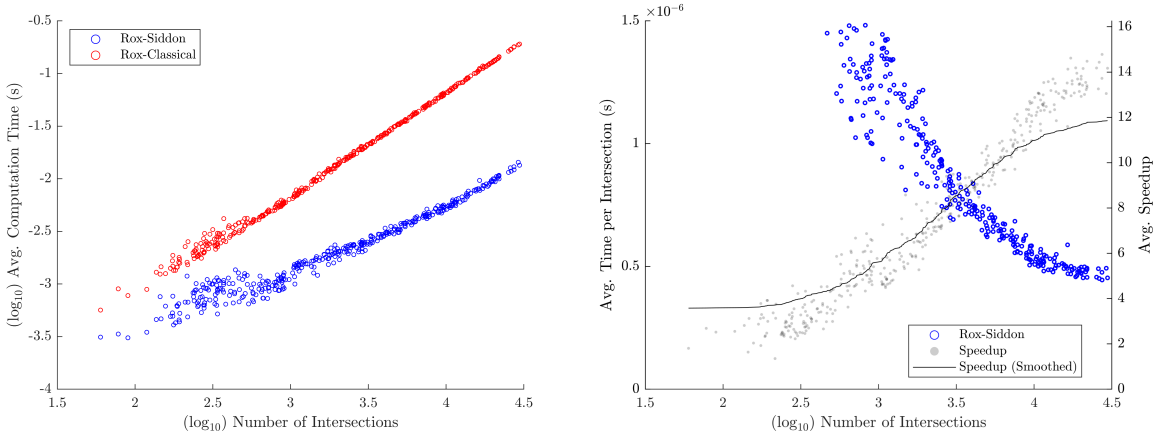
9.2 Numerical results

The numerical results included here are presented in the same order as in Chapter 8; we first compare computational speed-up for Rox-Siddon against classical ray-marching through a roxel grid, and then discuss the potential benefits of parallelisation.

[9] Determining a roxel index can be done by decomposing Q_1 into the chosen voxel basis, rounding values to the nearest integers, and then checking the 2-norm against all twelve neighbours. Similar to the hexagonal grid, this works because roxels are the Voronoi cells induced by the FCC lattice.

9.2.1 Computational cost

The total computational cost of the algorithm is linear in the number of intersections, as expected, and is considerably cheaper than conventional ray-marching. For sufficiently long paths, any computational overheads are dominated by the traversal cost, so that the average cost per intersection stabilises; in practice[10] this regime is reached for paths comprising approximately $10^{9/5}$ intersections. We present results in Figure 9.7, showing this clearly.



(a) (b)

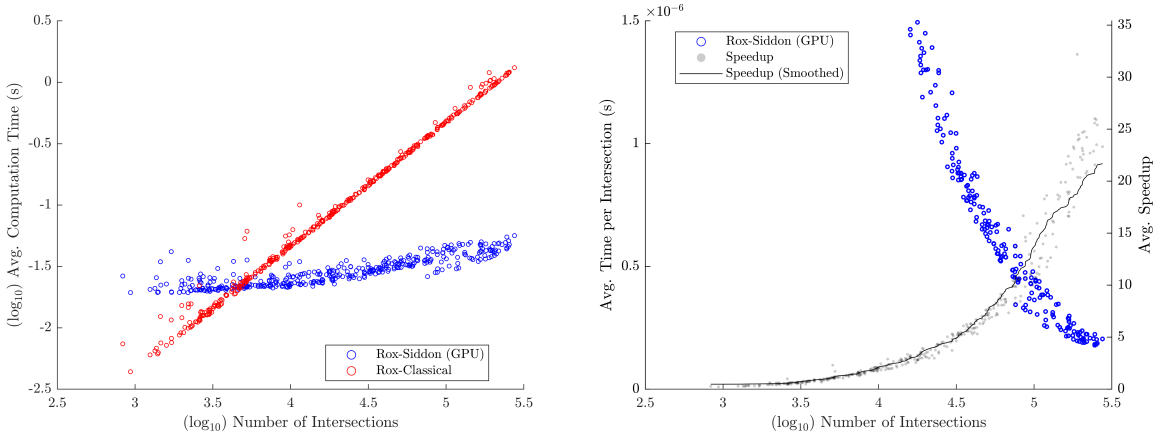
Figure 9.7: (a) The computational cost of Rox-Siddon is linear in the number of intersections, which itself is ultimately linear in the dimension of the roxel grid. (b) Per-intersection costs of the Rox-Siddon algorithm can attain a computational speed-up of up to $\sim$ 12–15$\times$ compared to classical ray-marching, depending on the total number of intersections along a ray.

Both implementations are written in MATLAB 2023A and run on a commercially available laptop (Surface Book 3, Intel i7-1065G7 CPU, 32GB RAM, Windows 11 Pro). Again, neither implementation has been fully optimised, although simplifications were made in the classical ray-marching algorithm to ensure that there was no unreasonable inflation of computational costs. As the algorithmic architecture of our implementation is essentially identical to the implementation of Hex-Siddon, the numerical robustness is comparable to that of Hex-Siddon. As such, the relative error between the length of the true radiological path, and the total sum of computed intersection lengths will be $\mathcal{O}(\varepsilon_{\text{mach}})$.

[10] Specifically, in practice on a single local machine.

9.2.2 Parallelisability

As the structure of Rox-Siddon builds directly on the framework of Hex-Siddon, this algorithm is also fully parallelisable in the sense that per-cell computations are decoupled. In Figure 9.8 we present results showing additional computational gains resulting from a parallel implementation through the MATLAB Parallel Computing Toolbox, with all parallel processes being run on an NVIDIA Quadro RTX 3000 (Max-Q).



(a) (b)

Figure 9.8: (a) As with Hex-Siddon, the asymptotic computational cost is still linear, although the initial setup costs are more demanding than for the non-parallelised implementation. As such, the linear behaviour is observed for much larger numbers of intersections. (b) Average speed-up using a naïve parallel implementation approaches ~ 25–$30\times$ for rays intersecting more than 10^5 roxels.

These additional results confirm that Rox-Siddon is not only parallelisable, but the parallel implementation yields substantially greater speed-ups for rays intersecting large numbers of roxels. In our implementation, no attempt has been made to exploit GPU-specific features, such as custom kernels, memory coalescing, or architecture-dependent optimisations. We therefore expect that performance is likely limited by data movement rather than arithmetic throughput—that is, the implementation is memory-bound rather than compute-bound—and so our timings should be interpreted as a conservative baseline, indicating the potential for further gains under a more specialised parallel implementation.

9.2.3 Extension to general tilings

We close this chapter with a brief comment on the generality exposed by extending Siddon's algorithm to non-standard voxel tilings. The method presented in this chapter, and in Chapter 8, is not specifically built around the geometric behaviour of Voronoi cells on a lattice, although the constructions presented here naturally form such structures.

Let us imagine a general periodic tiling in $\mathbb{R}^2$—the extension to $\mathbb{R}^3$ follows analogously—and for simplicity let us imagine that it is constructed from a single monotile. This covers a wide variety of cases, from classical square tilings, to general Voronoi tilings, to more exotic structures including non-convex tilings. Suppose we require the computation of an exact line integral through this tiling for which distinct intersections must be known exactly. Rather than relying on costly ray-marching schemes that step sequentially through the tiling, one may instead exploit the geometric structure directly, so that traversal can be formulated within this new 'Extended Siddon' framework. While the full development of this approach is beyond the scope of this thesis, the key requirement for extending this framework is captured by the steps illustrated in Figures 8.12 and 9.5. That is, we ask the question

Does the tiling admit a discrete support of edges over the ambient space?

While the present work does not attempt to formalise or classify tilings under this criterion, the constructions explored here suggest that such a property plays a central role in enabling efficient traversal. In particular, tilings which admit a discrete, globally indexed edge structure appear to support ray traversal schemes in which computational parallelism extends not only across rays but also across individual tiles.

Concluding remarks for Part III

In Part III we have investigated the impact of spatial discretisation as a modelling choice in algebraic tomographic reconstruction. Specifically, we have focused on algorithm design to ensure that the construction of forward operators does not become a computational bottleneck when solving the outer problem presented in Part II.

In two dimensions, hexagonal discretisations were examined as a concrete departure from Cartesian grids. By extending both the Fast Voxel Traversal algorithm and Siddon's algorithm to hexagonal lattices, we showed that forward operators can be constructed efficiently while exhibiting different neighbourhood structures and angular symmetries. Numerical experiments demonstrated that, although hexagonal grids incur slightly higher per-intersection costs, the resulting reconstructions are comparable to those built on square grids, with the most visible benefits appearing at high-contrast boundaries. While our results support the extended framework of using hexagonal discretisation for image reconstruction, the tuning of the regularisation parameter λ may reveal greater differences between the two discretisations.

This work was further extended to three dimensions through the use of rhombic dodecahedral voxels, suggesting that the traversal algorithms developed here reflect a geometry-agnostic philosophy, embedding intersection logic in a form compatible with massively parallel execution. The collective results of Part III show that redesigning the forward operator offers a complementary and largely unexplored route to improving algebraic reconstruction pipelines. Rather than proposing a universally optimal discretisation, we establish a framework for understanding how spatial representation and computational cost interact under a generalised traversal algorithm. This provides a foundation for future work on adaptive, non-Cartesian, and application-specific reconstruction models.

Chapter 10

Conclusions and extensions

This thesis has examined the problem of tomographic reconstruction under strict geometric and physical constraints, with a particular focus on limited-angle acquisition, object motion, and non-standard discretisation. Our work has centred on the construction of forward models, where the reconstruction problem is treated through a sequence of design choices for modelling image acquisition and domain discretisation, each of which imposes structural constraints on what may be stably recovered. This viewpoint has guided both the theoretical framing of our problem, as well as the algorithmic developments presented.

Part I of this thesis established the modelling foundations required for our perspective. Classical results from tomography—Radon and X-ray transforms, Fourier slice theorem, singularity detection—were reviewed to give a thorough understanding of the problems expected to arise in our work. For the modelling of a multi-emitter device, we also presented parametrisations of both fan-beam and cone-beam X-ray models, emphasising that both share the same underlying mathematical framework for characterising reconstructible objects. In this part of the thesis, we fixed a normalised geometry and attenuation model, along with a discretisation framework that was flexible enough to incorporate the action of motion effects. By incorporating motion into the construction of the forward operator itself, all explored models were then presented under the same formalism.

Part II of this thesis explored parametric motion as an explicit component of the forward operator, focusing on the progression of translation, rotation, and full rigid-body motion. For each model, we explicitly defined the effect of parametrised motion at the level of voxel-ray intersections, leading to operators that depend nonlinearly on the chosen parameters. Continuity results for the maps from motion parameters

to forward operators were established, clarifying that our operator-focused approach is stable to perturbations. This led to the understanding that sources of instability enter the motion-correction problem through modelling choices, such as geometric constraints and discretisation resolution, as well as through the ill-posedness of the inverse problem. Our objective was not to develop a new optimisation algorithm, but rather to diagnose the effects of motion from simple models. Numerical experiments exposed the structure of the resulting objective functions, illustrating the existence of nonsmooth, nonconvex landscapes which have identifiable regions containing function minima. An accompanying discussion on multilevel optimisation was presented not as a framework for a convergent solver, but as a tool for understanding resolution-dependent modelling error in nonsmooth problems. Further discussion focused on the possible choices of reconstruction basis, which guided the work in the third part of this thesis.

In Part III we introduced the most novel contributions: new ray traversal algorithms for non-Cartesian discretisations, with a focus on hexagonal pixels in two dimensions, and rhombic dodecahedral voxels in three dimensions. Results demonstrated that existing algorithms can be generalised beyond axis-aligned grids, provided that the underlying geometric structure can be exploited appropriately. The resulting algorithms achieve exact intersection lengths, predictable numerical behaviour, and favourable computational complexity compared to traditional algorithms. These developments were motivated by the desire to improve reconstruction quality through forward model design, and have been shown to integrate well into existing algebraic reconstruction frameworks. Importantly, the discretisations chosen in this work may be treated as interchangeable modelling choices within a geometry-agnostic methodology, enabling their advantages to be extended to a much richer family of geometric structures.

In total, this thesis argues that many of the practical difficulties encountered in modern tomographic reconstruction—particularly in limited-angle and motion-corrupted settings—are best understood as consequences of modelling inverse problems, rather than deficiencies of a particular reconstruction algorithm. By developing tools that operate directly at the level of the forward operator, our work provides a clearer framework for comparing discretisation choices, and designing acquisition geometries that may better exploit physical and computational constraints.

Several directions for future work follow from this perspective. The ray traversal framework developed here extends immediately to other periodic tilings and higher-

order polyhedral discretisations, suggesting a broad class of forward models that remain unexplored in tomographic imaging. This could be further developed by modelling the forward operator via neural networks, as the freedom from Cartesian geometric constraints may make non-standard discretisations particularly suited to learned forward models. That is, a learned approach may treat the discretisation geometry itself as a trainable parameter, to find the best representation for a given task.

With respect to the discussion in Chapter 7, the extended ideas of inexact, multilevel recursion point toward a theory of resolution-dependent optimisation that does not rely on classical smoothness or consistency assumptions. The explicit distinction between modelling, discretisation, and optimisation adopted throughout the thesis provides a natural interface for incorporating further acquisition technologies, such as multi-device systems, without reworking the underlying mathematical framework.

This thesis has sought to contribute to a broader modelling philosophy for tomographic imaging under non-ideal conditions. By focusing our approach on modelling clarity and forward operator structure, we hope to complement existing analytic and algorithmic approaches, and provide a foundation on which more flexible reconstruction methods may be built.

References

[1] S. T. Abbas and J. Sivaswamy, *PET image reconstruction and denoising on hexagonal lattices*, in Proceedings of the 2015 IEEE International Conference on Image Processing, IEEE, 2015, pp. 3481–3485.

[2] M. Aharon, M. Elad, and A. Bruckstein, *K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation*, IEEE Transactions on Signal Processing, 54 (2006), pp. 4311–4322.

[3] J. Amanatides and A. Woo, *A fast voxel traversal algorithm for ray tracing*, in Eurographics, vol. 87, 1987, pp. 3–10.

[4] J. Ambrose, *Computerized transverse axial scanning (tomography): Part 2. Clinical application*, The British Journal of Radiology, 46 (1973), pp. 1023–1047.

[5] A. H. Andersen and A. C. Kak, *Simultaneous algebraic reconstruction technique (SART): A superior implementation of the ART algorithm*, Ultrasonic Imaging, 6 (1984), pp. 81–94.

[6] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb, *Solving inverse problems using data-driven models*, Acta Numerica, 28 (2019), pp. 1–174.

[7] E. Baratella, B. Ruaro, C. Marrocchio, G. Poillucci, C. Pigato, A. M. Bozzato, F. Salton, P. Confalonieri, F. Crimi, B. Wade, E. Quaia, and M. A. Cova, *Diagnostic accuracy of chest digital tomosynthesis in patients recovering after COVID-19 pneumonia*, Tomography, 8 (2022), pp. 1221–1227.

[8] J. A. Belward, I. W. Turner, and M. Ilić, *On derivative estimation and the solution of least squares problems*, Journal of Computational and Applied Mathematics, 222 (2008), pp. 511–523.

[9] J. A. Belward, I. W. Turner, and M. N. Oqielat, *Numerical investigations of linear least squares methods for derivative estimation*, The Proceedings of ANZIAM, 50 (2008), pp. C844–C857.

[10] A. Biguri, H. Towsyfyan, R. Boardman, and T. Blumensath, *Numerically robust tetrahedron-based tomographic forward and backward projectors on parallel architectures*, Ultramicroscopy, 214 (2020). Article ID: 113016.

[11] R. Biswas, G. Largeteau-Skapin, R. Zrour, and E. Andres, *Rhombic dodecahedron grid—coordinate system and 3D digital object definitions*, in Proceedings of the International Conference on Discrete Geometry for Computer Imagery, Springer, 2019, pp. 27–37.

[12] R. Biswas, G. Largeteau-Skapin, R. Zrour, and E. Andres, *Digital objects in rhombic dodecahedron grid*, Mathematical Morphology - Theory and Applications, 4 (2020), pp. 143–158.

[13] A. E. M. Bocage, *Procédé et dispositifs de radiographie sur plaque en mouvement.* French patent No. 536,464, 1921. Brevet d'invention, République Française.

[14] J. E. Bresenham, *Algorithm for computer control of a digital plotter*, IBM Systems Journal, 4 (1965), pp. 25–30. Reprinted in *Seminal Graphics*, ACM Press, 1998.

[15] T. Briand and P. Monasse, *Theory and practice of image B-spline interpolation*, Image Processing On Line, 8 (2018), pp. 99–141.

[16] R. A. Brooks and G. Di Chiro, *Beam hardening in X-ray reconstructive tomography*, Physics in Medicine & Biology, 21 (1976), pp. 390–398.

[17] T. A. Bubba, G. Kutyniok, M. Lassas, M. März, W. Samek, S. Siltanen, and V. Srinivasan, *Learning the invisible: A hybrid deep learning-shearlet framework for limited angle computed tomography*, Inverse Problems, 35 (2019). Article ID 064002.

[18] M. Buehrer, J. Curcic, P. Boesiger, and S. Kozerke, *Prospective self-gating for simultaneous compensation of cardiac and respiratory motion*, Magnetic Resonance in Medicine, 60 (2008), pp. 683–690.

[19] J.-F. Cai, H. Ji, Z. Shen, and G.-B. Ye, *Data-driven tight frame construction and image denoising*, Applied and Computational Harmonic Analysis, 37 (2014), pp. 89–105.

[20] I. Calvo, S. SantaCruz-Calvo, M. G. Aranzana, P. Mármol, J. Á. Luque, I. Peral, E. M. Quijada, C. Gómez, C. Borrego, and J. Marín, *Digital tomosynthesis and COVID-19: An improvement in the assessment of pulmonary opacities*, Archivos de Bronconeumología (English Edition), 56 (2020), pp. 761–763.

[21] E. J. Candès and D. L. Donoho, *Curvelets: A surprisingly effective nonadaptive representation for objects with edges*, in Curve and Surface Fitting: Saint-Malo 1999, A. Cohen, C. Rabut, and L. L. Schumaker, eds., Vanderbilt University Press, Nashville, TN, 2000, pp. 105–120.

[22] E. J. Candès and D. L. Donoho, *Curvelets and reconstruction of images from noisy radon data*, in Proceedings of Wavelet Applications in Signal and Image Processing VIII, vol. 4119, SPIE, 2000, pp. 108–117.

[23] E. J. Candès, J. K. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences, 59 (2006), pp. 1207–1223.

[24] R. H. Chan, T. F. Chan, L. Shen, and Z. Shen, *Wavelet algorithms for high-resolution image reconstruction*, SIAM Journal on Scientific Computing, 24 (2003), pp. 1408–1432.

[25] S. S. Chen and D. L. Donoho, *Basis pursuit*, in Proceedings of the 1994 28th Asilomar Conference on Signals, Systems and Computers, vol. 1, IEEE, 1994, pp. 41–44.

[26] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM Review, 43 (2001), pp. 129–159.

[27] Z. Chen, K. Pawar, M. Ekanayake, C. Pain, S. Zhong, and G. F. Egan, *Deep learning for image enhancement and correction in magnetic resonance imaging—state-of-the-art and challenges*, Journal of Digital Imaging, 36 (2023), pp. 204–230.

[28] S. Chilamkurthy, R. Ghosh, S. Tanamala, M. Biviji, N. G. Campeau, V. K. Venugopal, V. Mahajan, P. Rao, and P. Warier, *Deep learning algorithms for detection of critical findings in head CT scans: A retrospective study*, The Lancet, 392 (2018), pp. 2388–2396.

[29] A. M. Cormack, *Representation of a function by its line integrals, with some radiological applications*, Journal of Applied Physics, 34 (1963), pp. 2722–2727.

[30] A. M. Cormack, *Representation of a function by its line integrals, with some radiological applications. II*, Journal of Applied Physics, 35 (1964), pp. 2908–2913.

[31] M. Cotte, J. Susini, J. Dik, and K. Janssens, *Synchrotron-based X-ray absorption spectroscopy for art conservation: Looking back and looking forward*, Accounts of Chemical Research, 43 (2010), pp. 705–714.

[32] M. E. Davison, *The ill-conditioned nature of the limited angle tomography problem*, SIAM Journal on Applied Mathematics, 43 (1983), pp. 428–448.

[33] C. de Boor, *A practical guide to splines*, vol. 27 of Applied Mathematical Sciences, Springer New York, 1978.

[34] C. de Boor, K. Höllig, and S. Riemenschneider, *Box splines*, vol. 98 of Applied Mathematical Sciences, Springer-Verlag, 1993.

[35] B. De Man and S. Basu, *Distance-driven projection and backprojection*, in Proceedings of the 2002 IEEE Nuclear Science Symposium, vol. 3, IEEE, 2002, pp. 1477–1480.

[36] B. De Man and S. Basu, *Distance-driven projection and backprojection in three dimensions*, Physics in Medicine & Biology, 49 (2004), pp. 2463–2475.

[37] S. R. Deans, *The Radon transform and some of its applications*, John Wiley & Sons, 1983.

[38] M. N. Do and M. Vetterli, *The contourlet transform: An efficient directional multiresolution image representation*, IEEE Transactions on Image Processing, 14 (2005), pp. 2091–2106.

[39] J. T. Dobbins III, *Tomosynthesis imaging: At a translational crossroads*, Medical physics, 36 (2009), pp. 1956–1967.

[40] D. L. Donoho, *Compressed sensing*, IEEE Transactions on Information Theory, 52 (2006), pp. 1289–1306.

[41] D. L. Donoho, *For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution*, Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences, 59 (2006), pp. 797–829.

[42] D. L. Donoho and M. Elad, *Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization*, Proceedings of the National Academy of Sciences, 100 (2003), pp. 2197–2202.

[43] J. Duryea, J. Dobbins III, and J. Lynch, *Digital tomosynthesis of hand joints for arthritis assessment*, Medical Physics, 30 (2003), pp. 325–333.

[44] S. K. Dwivedi, M. Vishwakarma, and A. Soni, *Advances and researches on non destructive testing: A review*, Materials Today: Proceedings, 5 (2018), pp. 3690–3698.

[45] J. C. Ehrhardt, *Hexagonal fast Fourier transform with rectangular output*, IEEE Transactions on Signal Processing, 41 (1993), pp. 1469–1472.

[46] A. Entezari, R. Dyer, and T. Möller, *From sphere packing to the theory of optimal lattice sampling*, in Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration, Springer, 2009, pp. 227–255.

[47] A. Entezari, M. Nilchian, and M. Unser, *A box spline calculus for the discretization of computed tomography reconstruction problems*, IEEE Transactions on Medical Imaging, 31 (2012), pp. 1532–1541.

[48] A. Entezari, D. Van De Ville, and T. Möller, *Practical box splines for reconstruction on the body centered cubic lattice*, IEEE Transactions on Visualization and Computer Graphics, 14 (2008), pp. 313–328.

[49] C. L. Epstein, *Introduction to the mathematics of medical imaging*, SIAM, 2007.

[50] A. Faridani, *Introduction to the mathematics of computed tomography*, Inside Out: Inverse Problems and Applications, 47 (2003), pp. 1–46.

[51] L. A. Feldkamp, L. C. Davis, and J. W. Kress, *Practical cone-beam algorithm*, Journal of the Optical Society of America A, 1 (1984), pp. 612–619.

[52] A. Ferrari, L. Bertolaccini, P. Solli, P. O. Di Salvia, and D. Scaradozzi, *Digital chest tomosynthesis: The 2017 updated review of an emerging application*, Annals of Translational Medicine, 6 (2018). Article ID: 91.

[53] B. Finkbeiner, U. R. Alim, D. Van De Ville, and T. Möller, *High-quality volumetric reconstruction on optimal lattices for computed tomography*, in Proceedings of Eurographics / IEEE-VGTC Symposium on Visualization 2009, vol. 28, Wiley Online Library, 2009, pp. 1023–1030.

[54] H. M. Gach, C. Tanase, and F. Boada, *2D & 3D Shepp-Logan phantom standards for MRI*, in Proceedings of the 2008 19th International Conference on Systems Engineering, IEEE, 2008, pp. 521–526.

[55] J. Gao, Y. Yang, P. Lin, and D. S. Park, *Computer vision in healthcare applications*, Journal of Healthcare Engineering, 2018 (2018). Article ID: 5157020.

[56] E. Garduño, G. T. Herman, and H. Katz, *Boundary tracking in 3D binary images to produce rhombic faces for a dodecahedral model*, in Proceedings of the 9th European Signal Processing Conference, IEEE, 1998, pp. 1–4.

[57] J. B. Garrison, D. G. Grant, W. H. Guier, and R. J. Johns, *Three dimensional roentgenography*, American Journal of Roentgenology, 105 (1969), pp. 903–908.

[58] G. Gennaro, A. Toledano, C. Di Maggio, E. Baldan, E. Bezzon, M. La Grassa, L. Pescarini, I. Polico, A. Proietti, A. Toffoli, and P. C. Muzzio, *Digital breast tomosynthesis versus digital mammography: A clinical performance study*, European Radiology, 20 (2010), pp. 1545–1553.

[59] M. J. E. Golay, *Hexagonal parallel pattern transformations*, IEEE Transactions on Computers, C-18 (1969), pp. 733–740.

[60] R. Gordon, R. Bender, and G. T. Herman, *Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography*, Journal of Theoretical Biology, 29 (1970), pp. 471–481.

[61] S. Gratton, A. Sartenaer, and P. L. Toint, *Recursive trust-region methods for multiscale nonlinear optimization*, SIAM Journal on Optimization, 19 (2008), pp. 414–444.

[62] S. Gratton and P. L. Toint, *Multi-secant equations, approximate invariant subspaces and multigrid optimization*, tech. report, Technical Report, Department of Mathematics, FUNDP, Namur (B), 2007.

[63] R. Grimm, S. Fürst, M. Souvatzoglou, C. Forman, J. Hutter, I. Dregely, S. I. Ziegler, B. Kiefer, J. Hornegger, K. T. Block, and S. G. Nekolla, *Self-gated MRI motion modeling for respiratory motion compensation in integrated PET/MRI*, Medical Image Analysis, 19 (2015), pp. 110–120.

[64] M. Guckenberger, A. Richter, J. Boda-Heggemann, and F. Lohr, *Motion compensation in radiotherapy*, Critical Reviews in Biomedical Engineering, 40 (2012), pp. 187–197.

[65] K. Guo, G. Kutyniok, and D. Labate, *Sparse multidimensional representations using anisotropic dilation and shear operators*, Wavelets and Splines, 14 (2006), pp. 189–201.

[66] G. Han, Z. Liang, and J. You, *A fast ray-tracing technique for TCT and ECT studies*, in Proceedings of the 1999 Nuclear Science Symposium and Medical Imaging Conference, vol. 3, IEEE, 1999, pp. 1515–1518.

[67] R. Hanke, T. Fuchs, and N. Uhlmann, *X-ray based methods for non-destructive testing and material characterization*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 591 (2008), pp. 14–18.

[68] M. S. Hansen, T. S. Sørensen, A. E. Arai, and P. Kellman, *Retrospective reconstruction of high temporal resolution cine images from real-time MRI using iterative motion correction*, Magnetic Resonance in Medicine, 68 (2012), pp. 741–750.

[69] P. C. Hansen, *Discrete inverse problems: Insight and algorithms*, SIAM, 2010.

[70] I. Her, *Geometric transformations on the hexagonal grid*, IEEE Transactions on Image Processing, 4 (1995), pp. 1213–1222.

[71] I. Her and C.-T. Yuan, *Resampling on a pseudohexagonal grid*, CVGIP: Graphical Models and Image Processing, 56 (1994), pp. 336–347.

[72] G. T. Herman, *Correction for beam hardening in computed tomography*, Physics in Medicine & Biology, 24 (1979), pp. 81–106.

[73] N. J. Higham, *Accuracy and stability of numerical algorithms*, SIAM, 2002.

[74] B. K. P. Horn, *Fan-beam reconstruction methods*, Proceedings of the IEEE, 67 (1979), pp. 1616–1623.

[75] G. N. Hounsfield, *Computerized transverse axial scanning (tomography): Part 1. Description of system*, The British Journal of Radiology, 46 (1973), pp. 1016–1022.

[76] V. Israel-Jost, *FA-SART: A frequency-adaptive algorithm in pinhole SPECT tomography*, SIAM Journal on Scientific Computing, 30 (2008), pp. 819–836.

[77] F. Jacobs, E. Sundermann, B. De Sutter, M. Christiaens, and I. Lemahieu, *A fast algorithm to calculate the exact radiological path through a pixel or voxel space*, Journal of Computing and Information Technology, 6 (1998), pp. 89–94.

[78] U. Je, H. Cho, M. Lee, J. Oh, Y. Park, D. Hong, C. Park, H. Cho, S. Choi, and Y. Koo, *Dental cone-beam CT reconstruction from limited-angle view data based on compressed-sensing (CS) theory for fast, low-dose X-ray imaging*, Journal of the Korean Physical Society, 64 (2014), pp. 1907–1911.

[79] M. Jiang and G. Wang, *Convergence of the simultaneous algebraic reconstruction technique (SART)*, IEEE Transactions on Image Processing, 12 (2003), pp. 957–961.

[80] R. J. Johns, D. G. Grant, and J. B. Garrison, *Three dimensional radiography*, Transactions of the American Clinical and Climatological Association, 82 (1971), pp. 136–140.

[81] P. M. Joseph, *An improved algorithm for reprojecting rays through pixel images*, IEEE Transactions on Medical Imaging, 1 (1982), pp. 192–196.

[82] S. Kaczmarz, *Angenäherte Auflösung von Systemen linearer Gleichungen*, Bulletin International de l'Académie Polonaise des Sciences et des Lettres, Classe des Sciences Mathématiques et Naturelles, Série A, 35 (1937), pp. 355–357. English translation: S. Kaczmarz, *Approximate solution of systems of linear equations*, International Journal of Control, 57 (1993), pp. 1269–1271.

[83] A. C. Kak and M. Slaney, *Principles of computerized tomographic imaging*, SIAM, 2001.

[84] J. Kieffer, *X-ray device and method of technique*. U.S. Patent No. 1,954,321, 1934.

[85] C. Kirisits, *Total variation denoising on hexagonal grids*. Preprint, arXiv:1204.3855 [math.OC], 2012.

[86] M. Klodt and R. Hauser, *3D image reconstruction from X-ray measurements with overlap*, in Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 19–33.

[87] M. Klodt and R. Hauser, *Nonlinear compressed sensing for multi-emitter X-ray imaging*, in Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, Springer, 2017, pp. 189–204.

[88] M. Knaup, S. Steckmann, O. Bockenbach, and M. Kachelriess, *CT image reconstruction using hexagonal grids*, in Proceedings of the 2007 IEEE Nuclear Science Symposium, vol. 4, IEEE, 2007, pp. 3074–3076.

[89] C. G. Koay, J. E. Sarlls, and E. Özarslan, *Three-dimensional analytical magnetic resonance imaging phantom in the Fourier domain*, Magnetic Resonance in Medicine, 58 (2007), pp. 430–436.

[90] T. Köhler, H. Turbell, and M. Grass, *Efficient forward projection through discrete data sets using tri-linear interpolation*, in Proceedings of the 2000 IEEE Nuclear Science Symposium, vol. 2, IEEE, 2000, pp. 113–115.

[91] T. Küstner, K. Armanious, J. Yang, B. Yang, F. Schick, and S. Gatidis, *Retrospective correction of motion-affected MR images using deep learning frameworks*, Magnetic Resonance in Medicine, 82 (2019), pp. 1527–1540.

[92] A. F. Laine, S. Schuler, W. Huda, J. C. Honeyman-Buck, and B. G. Steinbach, *Hexagonal wavelet processing of digital mammography*, in Proceedings of Medical Imaging 1993: Image Processing, vol. 1898, SPIE, 1993, pp. 559–573.

[93] J. J. Lee and C.-K. Kim, *Tomosynthesis feasibility study for visualization of interiors of wood columns surrounded with walls*, Journal of the Korean Wood Science and Technology, 50 (2022), pp. 246–255.

[94] R. M. Lewitt, *Multidimensional digital image representations using generalized Kaiser–Bessel window functions*, Journal of the Optical Society of America A, 7 (1990), pp. 1834–1846.

[95] R. M. Lewitt, *Alternatives to voxels for image representation in iterative reconstruction algorithms*, Physics in Medicine & Biology, 37 (1992), pp. 705–716.

[96] C. Li, *An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing*, Master's thesis, Rice University, 2010.

[97] C. Li, H. Jiang, P. Wilford, Y. Zhang, and M. Scheutzow, *A new compressive video sensing framework for mobile broadcast*, IEEE Transactions on Broadcasting, 59 (2013), pp. 197–205.

[98] C. Li, W. Yin, H. Jiang, and Y. Zhang, *An efficient augmented Lagrangian method with applications to total variation minimization*, Computational Optimization and Applications, 56 (2013), pp. 507–530.

[99] T. Li, M. Zhang, W. Qi, E. Asma, and J. Qi, *Motion correction of respiratory-gated PET images using deep learning based image registration framework*, Physics in Medicine & Biology, 65 (2020). Article ID: 155003.

[100] J. Maclaren, M. Herbst, O. Speck, and M. Zaitsev, *Prospective motion correction in brain imaging: A review*, Magnetic Resonance in Medicine, 69 (2013), pp. 621–636.

[101] C. Malamateniou, S. J. Malik, S. J. Counsell, J. M. Allsop, A. K. McGuinness, T. Hayat, K. Broadhouse, R. G. Nunes, A. M. Ederies, J. V. Hajnal, and M. A. Rutherford, *Motion-compensation techniques in neonatal and fetal MR imaging*, American Journal of Neuroradiology, 34 (2013), pp. 1124–1136.

[102] S. Mallat, *A wavelet tour of signal processing*, Elsevier, 1999.

[103] R. M. Mersereau, *The processing of hexagonally sampled two-dimensional signals*, Proceedings of the IEEE, 67 (1979), pp. 930–949.

[104] L. Middleton and J. Sivaswamy, *Framework for practical hexagonal-image processing*, Journal of Electronic Imaging, 11 (2002), pp. 104–114.

[105] L. Middleton and J. Sivaswamy, *Hexagonal image processing: A practical approach*, Springer, 2005.

[106] E. R. Miller, E. M. McCurry, and B. Hruska, *An infinite number of laminagrams from a finite number of radiographs*, Radiology, 98 (1971), pp. 249–255.

[107] F. Momey, L. Denis, C. Burnier, E. Thiébaut, J.-M. Becker, and L. Desbat, *Spline driven: high accuracy projectors for tomographic reconstruction from few projections*, IEEE Transactions on Image Processing, 24 (2015), pp. 4715–4725.

[108] S. R. Moosekian, S. Jeong, B. P. Marks, and E. T. Ryser, *X-ray irradiation as a microbial intervention strategy for food*, Annual Review of Food Science and Technology, 3 (2012), pp. 493–510.

[109] S. G. Nash, *A multigrid approach to discretized optimization problems*, Optimization Methods and Software, 14 (2000), pp. 99–116.

[110] A. L. Nel, *Hexagonal image processing*, in COMSIG 1989 Proceedings: Southern African Conference on Communications and Signal Processing, IEEE, 1989, pp. 109–113.

[111] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer, 2006.

[112] S. Ouadah, M. Jacobson, J. W. Stayman, T. Ehtiati, C. Weiss, and J. H. Siewerdsen, *Correction of patient motion in cone-beam CT using 3D–2D registration*, Physics in Medicine & Biology, 62 (2017), pp. 8813–8831.

[113] A. Ouahabi, *A review of wavelet denoising in medical imaging*, in Proceedings of the 8th International Workshop on Systems, Signal Processing and their Applications, IEEE, 2013, pp. 19–26.

[114] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Transactions on Mathematical Software (TOMS), 8 (1982), pp. 43–71.

[115] J. Pan, T. Zhou, Y. Han, and M. Jiang, *Variable weighted ordered subset image reconstruction algorithm*, International Journal of Biomedical Imaging, 2006 (2006). Article ID: 010398.

[116] K. Patel, *Imaging with X-ray emitter arrays*, Master's thesis, University of Oxford, 2012.

[117] D. P. Petersen and D. Middleton, *Sampling and reconstruction of wave-number-limited functions in N-dimensional Euclidean spaces*, Information and Control, 5 (1962), pp. 279–323.

[118] J. Plasencia-Martínez, A. Moreno-Pastor, M. Lozano-Ros, C. Jiménez-Pulido, I. Herves-Escobedo, G. Pérez-Hernández, and J. M. García-Santos, *Digital tomosynthesis improves chest radiograph accuracy and reduces microbiological false negatives in COVID-19 diagnosis*, Emergency Radiology, 30 (2023), pp. 465–474.

[119] Z. Qiao, G. Redler, Z. Gui, Y. Qian, B. Epel, and H. Halpern, *Three novel accurate pixel-driven projection methods for 2D CT and 3D EPR imaging*, Journal of X-ray Science and Technology, 26 (2018), pp. 83–102.

[120] 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 (1993), pp. 1215–1225.

[121] E. T. Quinto, *An introduction to X-ray tomography and Radon transforms*, in Proceedings of Symposia in Applied Mathematics, vol. 63, 2006, pp. 1–23.

[122] Qure.ai, *Non-contrast head/brain CT CQ500 dataset*, 2018. Archived URL: `https://web.archive.org/web/20220816011051/http://headctstudy.qure.ai/`. Academic Torrents Mirror: `https://academictorrents.com/details/47e9d8aab761e75fd0a81982fa62bddf3a173831`. Accessed 13 July 2026.

[123] J. Radon, *Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten*, Berichte über die Verhandlungen der Königlich Sächsischen Gesellschaft der Wissenschaften zu Leipzig, Mathematisch-Physische Klasse, 69 (1917), pp. 262–277. English translation: J. Radon (trans. P. C. Parks), *On the determination of functions from their integral values along certain manifolds*, IEEE Transactions on Medical Imaging, 5 (1986), pp. 170–176.

[124] W. C. Röntgen, *On a new kind of rays*, Science, 3 (1896), pp. 227–231.

[125] A. Rosenfeld and J. L. Pfaltz, *Distance functions on digital pictures*, Pattern Recognition, 1 (1968), pp. 33–61.

[126] M. Schabel, *3D Shepp–Logan phantom*. MATLAB Central File Exchange, 2006. `https://www.mathworks.com/matlabcentral/fileexchange/9416-3d-shepp-logan-phantom`. Accessed 13 July 2026.

[127] M. Schreiner, B. Frühmann, D. Jembrih-Simbürger, and R. Linke, *X-rays in art and archaeology: An overview*, Powder Diffraction, 19 (2004), pp. 3–11.

[128] I. Sechopoulos, *A review of breast tomosynthesis. Part I. The image acquisition process*, Medical Physics, 40 (2013). Article ID: 014301.

[129] I. Sechopoulos, *A review of breast tomosynthesis. Part II. Image reconstruction, processing and analysis, and advanced applications*, Medical Physics, 40 (2013). Article ID: 014302.

[130] L. A. Shepp and B. F. Logan, *The Fourier reconstruction of a head section*, IEEE Transactions on Nuclear Science, 21 (1974), pp. 21–43.

[131] L. A. Shepp and B. F. Logan, *Reconstructing interior head tissue from X-ray transmissions*, IEEE Transactions on Nuclear Science, 21 (1974), pp. 228–236.

[132] R. L. Siddon, *Fast calculation of the exact radiological path for a three-dimensional CT array*, Medical Physics, 12 (1985), pp. 252–255.

[133] A. Sisniega, J. W. Stayman, J. Yorkston, J. H. Siewerdsen, and W. Zbijewski, *Motion compensation in extremity cone-beam CT using a penalized image sharpness criterion*, Physics in Medicine & Biology, 62 (2017), pp. 3712–3734.

[134] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, *Implicit neural representations with periodic activation functions*, Advances in Neural Information Processing Systems, 33 (2020), pp. 7462–7473.

[135] B. D. Smith, *Image reconstruction from cone-beam projections: Necessary and sufficient conditions and reconstruction methods*, IEEE Transactions on Medical Imaging, 4 (1985), pp. 14–25.

[136] D. C. Solmon, *The X-ray transform*, PhD thesis, Oregon State University, 1974.

[137] V. Spieker, H. Eichhorn, K. Hammernik, D. Rueckert, C. Preibisch, D. C. Karampinos, and J. A. Schnabel, *Deep learning for retrospective motion correction in MRI: A comprehensive review*, IEEE Transactions on Medical Imaging, 43 (2023), pp. 846–859.

[138] R. Strand, *Using the hexagonal grid for three-dimensional images: Direct Fourier method reconstruction and weighted distance transform*, in Proceedings of the 18th International Conference on Pattern Recognition, vol. 2, IEEE, 2006, pp. 1169–1172.

[139] R. Tibshirani, *Regression shrinkage and selection via the Lasso*, Journal of the Royal Statistical Society Series B: Statistical Methodology, 58 (1996), pp. 267–288.

[140] I. W. Turner, J. A. Belward, and M. N. Oqielat, *Error bounds for least squares gradient estimates*, SIAM Journal on Scientific Computing, 32 (2010), pp. 2146–2166.

[141] H. Tuy, *An inversion formula for cone-beam reconstruction*, SIAM Journal on Applied Mathematics, 43 (1983), pp. 546–552.

[142] M. Unser, A. Aldroubi, and M. Eden, *B-spline signal processing. I. Theory*, IEEE Transactions on Signal Processing, 41 (1993), pp. 821–833.

[143] M. Unser, A. Aldroubi, and M. Eden, *B-spline signal processing. II. Efficiency design and applications*, IEEE Transactions on Signal Processing, 41 (1993), pp. 834–848.

[144] M. Unser, A. Aldroubi, and M. Eden, *A family of polynomial spline wavelet transforms*, Signal Processing, 30 (1993), pp. 141–162.

[145] M. D. Walker, K. M. Bradley, and D. R. McGowan, *Evaluation of principal component analysis-based data-driven respiratory gating for positron emission tomography*, The British Journal of Radiology, 91 (2018). Article ID: 20170793.

[146] M. D. Walker, A. J. Morgan, K. M. Bradley, and D. R. McGowan, *Evaluation of data-driven respiratory gating waveforms for clinical PET imaging*, EJNMMI Research, 9 (2019). Article ID: 1.

[147] M. D. Walker, A. J. Morgan, K. M. Bradley, and D. R. McGowan, *Data-driven respiratory gating outperforms device-based gating for clinical ^{18}F-FDG PET/CT*, Journal of Nuclear Medicine, 61 (2020), pp. 1678–1683.

[148] G. Wang and M. Jiang, *Ordered-subset simultaneous algebraic reconstruction techniques (OS-SART)*, Journal of X-ray Science and Technology, 12 (2004), pp. 169–177.

[149] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image quality assessment: From error visibility to structural similarity*, IEEE Transactions on Image Processing, 13 (2004), pp. 600–612.

[150] E. B. Welch, A. Manduca, R. C. Grimm, H. A. Ward, and C. R. Jack Jr, *Spherical navigator echoes for full 3D rigid body motion measurement in MRI*, Magnetic Resonance in Medicine, 47 (2002), pp. 32–41.

[151] N. White, C. Roddey, A. Shankaranarayanan, E. Han, D. Rettmann, J. Santos, J. Kuperman, and A. Dale, *PROMO: Real-time prospective motion correction in MRI using image-based tracking*, Magnetic Resonance in Medicine, 63 (2010), pp. 91–105.

[152] A. Yabushita and K. Ogawa, *Image reconstruction with a hexagonal grid*, in Proceedings of the 2002 IEEE Nuclear Science Symposium, vol. 3, IEEE, 2002, pp. 1500–1503.

[153] S. Yi, D. Labate, G. R. Easley, and H. Krim, *A shearlet approach to edge analysis and detection*, IEEE Transactions on Image Processing, 18 (2009), pp. 929–941.

[154] M. Zaitsev, B. Akin, P. LeVan, and B. R. Knowles, *Prospective motion correction in functional MRI*, NeuroImage, 154 (2017), pp. 33–42.

[155] M. Zaitsev, C. Dold, G. Sakas, J. Hennig, and O. Speck, *Magnetic resonance imaging of freely moving objects: Prospective real-time motion correction using an external optical motion tracking system*, NeuroImage, 31 (2006), pp. 1038–1050.

[156] Z. B. Zehi, A. Afshari, S. Noori, B. Jannat, and M. Hashemi, *The effects of X-ray irradiation on safety and nutritional value of food: A systematic review article*, Current Pharmaceutical Biotechnology, 21 (2020), pp. 919–926.

[157] H. Zhao and A. J. Reader, *Fast ray-tracing technique to calculate line integral paths in voxel arrays*, in Proceedings of the 2003 IEEE Nuclear Science Symposium, vol. 4, IEEE, 2003, pp. 2808–2812.

[158] S.-R. Zhao and H. Halling, *A new Fourier method for fan beam reconstruction*, in Proceedings of the 1995 IEEE Nuclear Science Symposium and Medical Imaging Conference, vol. 2, IEEE, 1995, pp. 1287–1291.

[159] X. Zheng, *Computer simulations for the effectiveness of CT image reconstruction using hexagonal grids versus the usual square grids with the same sampling rate*, in Proceedings of the 9th International Conference on Biomedical Imaging, Signal Processing, 2024, pp. 98–103.

[160] B. G. Ziedses des Plantes, *Eine neue Methode zur Differenzierung in der Röntgenographie (planigraphie) [(English trans.) A new method of differentiation in radiography (planigraphy)]*, Acta Radiologica, 13 (1932), pp. 182–192.