

Durham E-Theses

Regularized Multigrid Optimization for Material Reconstruction from Single Medical X-ray Images

WESTMACOTT, HENRY,JAMES

How to cite:

WESTMACOTT, HENRY,JAMES (2023) *Regularized Multigrid Optimization for Material Reconstruction from Single Medical X-ray Images*, Durham theses, Durham University. Available at Durham E-Theses Online: <http://etheses.dur.ac.uk/15450/>

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a [link](#) is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

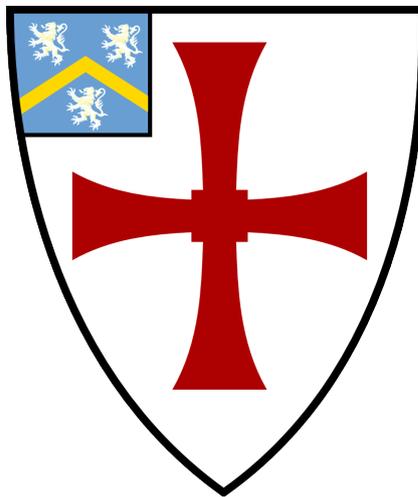
The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full Durham E-Theses policy](#) for further details.

Regularized Multigrid Optimization for Material Reconstruction from Single Medical X-ray Images

Henry Westmacott

A thesis presented for the degree of
Doctor of Philosophy



Department of Computer Science
The University of Durham
United Kingdom
January 2023

Regularized Multigrid Optimization for Material Reconstruction from Single Medical X-ray Images

Henry Westmacott

Abstract

This thesis presents a novel technique for the estimation of 3D structural and material composition of anatomies imaged with X-rays. These estimates are produced from a single image with associated X-ray detector data. This method is made possible with access to software for X-ray simulation and segmentation, both developed by and provided to us by IBEX Innovations.

This work combines existing concepts from optimization and multi-grid methods to present a novel concept for using domain knowledge to sufficiently constrain an otherwise unsolvable problem to produce valuable output. Specifically, it is shown that by transforming knowledge about the shape and composition of anatomies into regularizing functions, we can produce models of their internal structure that are accurate enough to simulate X-ray scatter, and thereby remove noise from the final images in a physics-guided way.

By introducing weighted penalties for results that do not conform to expectations from domain knowledge, which are informed by IBEX's neural network for X-ray segmentation, we can estimate the shape and material composition of a 3D object from a single image which - in theory - does not contain enough information to produce such a model.

This work makes use of an X-ray simulation tool and associated data created by IBEX innovations and provided to us. We have created an optimization algorithm that iteratively processes this data with the IBEX simulation tool, then updates the estimated material composition of the imaged anatomy by imposing regularizing functions that penalize models that do not conform to our expectations about real anatomies. This is implemented on multi-grid,

showing improved reconstruction quality and speed by producing coarse models first, followed by a custom algorithm for optimally selecting coarsening and refining of the model to produce the most accurate model.

By using IBEX’s simulation algorithm, we show that we can constrain an otherwise ill-posed problem. These novel tools allow us to solve the problem of estimating 3D material composition from a single image, by considering simple features of organic shapes such as continuity and smoothness. We demonstrate that with access to sufficiently powerful simulation tools, even simple assumptions about our target facilitate intuitive material estimations.

The algorithm presented in this thesis has certain limitations. We are only able to produce models of anatomies at low resolutions, constructed of just two distinct materials, without fully capturing the 3D structure of the anatomy. Nonetheless, we demonstrate that it is possible to capture enough structural information to produce an accurate scatter estimate, which would not be possible without the research we present here. These limitations are imposed to simplify our solutions such that they can be found using conventional hardware, and to constrain our problem into the scope of feasibility. Furthermore, the choice to limit our models to 2.5D and just two materials reflects the models used by IBEX Innovations and their X-ray simulation method, which we require for our optimization.

To our understanding, no other published work in this field has applied an approach like ours to X-ray image reconstruction. Inferring from a single image not just depth information, but also an abstraction of information about the internal structure, in a way that is physically motivated. We hope that this concept could be applied to other problems in future, where systems are well understood but hindered by limited data availability or high capture costs.

Acknowledgements

This work was made possible thanks to funding from the European Regional Development Fund (ERDF), through the Intensive Industrial Innovation Programme (IIIP).

I'd like to thank both of my supervisors, Ioannis Ivrisimtzis and Tobias Weinzierl, for all the help they have offered me during the course of performing this research. I have on numerous occasions been struck by how lucky I have been to work with such thought-provoking, supportive and friendly people.

Thanks of course must also go to the staff at IBEX Innovations, especially Paul Scott and Richard McWilliam, for providing access to the software and data that have made this work possible, and for the time and energy invested in me helping find my feet in this work.

Thank you to Julie McLoughlin for all your help ironing out details of this collaboration, and for setting my mind at ease during some of the more stressful points.

A huge thank you all of the friends who have offered encouragement, discussion and distraction during this long journey. Thank you for the holidays, pub trips, common-room drinks and zoom calls. You can all now stop asking when it will finally be finished. Thank you to all of my family for all of your love and support, and for inspiring me with science and fun from the beginning.

Finally, thank you to my long-suffering partner, Fran, for ensuring I had food to eat and a roof to sleep under during this endeavour. Without your support I suspect this wouldn't have been possible.

Contents

Declaration	ix
List of Figures	x
Nomenclature	xxi
1 Introduction	1
1.1 Context	1
1.2 Problem	3
1.3 Objectives	4
1.3.1 Challenges	5
1.4 Contributions	6
1.4.1 Novelty	7
1.5 Limitations	8
1.6 The Structure of the Thesis to Follow	9
1.6.1 Chapter 2 - Problem outline	9
1.6.2 Chapter 3 - Background	9
1.6.3 Chapter 4 - Exploratory Research	10
1.6.4 Chapter 5 - Regularized Descent	10
1.6.5 Chapter 6 - Multi-scale Methods	11
1.6.6 Chapter 7 - Results	11

1.6.7	Chapter 8 - Conclusion	12
2	Problem Outline	13
2.1	Scatter as an Entryway	15
2.2	The Structure of TrueView	16
2.2.1	Simulated Databases	21
2.3	Segmentation	22
2.4	Data Description	23
2.4.1	Device Measurements	23
2.4.2	Datasets	25
3	Background	32
3.1	Simulators	32
3.1.1	Monte Carlo Simulation	32
3.1.2	Geant4	33
3.2	Optimisation	34
3.2.1	Ill-posed Problems	34
3.2.2	Regularization	35
3.3	X-ray Imaging	38
3.3.1	Scatter	38
3.3.2	Scatter Removal	38
3.4	Segmentation	40
3.5	Speed and Efficiency	41
3.6	Explainable Artificial Intelligence	42
3.7	Depth Estimation and 3D Reconstruction	43
3.8	Conclusion	44
4	Problem Formulation and Exploratory Studies	46
4.1	Formulation as an Optimisation Problem	46
4.2	Exploratory Research	47
4.2.1	Simplified Scatter Model	47

4.2.2	Gradient Descent	50
4.2.3	Stochastic Gradient Descent	50
4.2.4	Mini-batch Gradient Descent	51
4.2.5	Simulated Annealing	52
4.2.6	Test Cases	53
4.3	Regularization in the Exploratory Setting	56
4.3.1	Simple Segmentation Regularizer	57
4.3.2	Simple Smoothness Regularizer	62
4.3.3	Simple Regularizers Combined	64
5	Regularized Descent	72
5.1	Simulation via TrueView	72
5.1.1	Comparison against exploratory research	77
5.2	Regularization	79
5.2.1	Smoothness	79
5.2.2	Segmentation	84
5.2.3	Physicality	85
5.2.4	Continuity	89
5.3	Descent methods	92
5.3.1	Bi-directional Greedy Descent	92
5.3.2	Stochastic Gradient Descent	95
5.3.3	Newton's Method	96
5.4	Conclusion	99
6	Multi-scale	100
6.1	Introduction	101
6.2	Resolution-Speed Relationship	102
6.3	Limitations on Resolutions Available	103
6.3.1	Simulated and Pseudo-pixel Resolutions	104
6.3.2	Algebraic Re-sampling	106

6.4	Translating Spatial Models Between Resolutions	108
6.4.1	Coarsening Resolution	109
6.4.2	Refining Resolution	110
6.4.3	Interpolation/Rescaling Methods	112
6.4.3.1	Bilinear Interpolation	114
6.4.3.2	Nearest Neighbour / Injection	115
6.4.3.3	Nearest Neighbour with Local Averaging	116
6.4.3.4	Comparison of Results	117
6.5	Paths Through Resolutions	118
6.5.1	V-Cycles	118
6.5.2	Full Multi-grid	120
6.5.3	Results	122
6.5.3.1	Single Grid	123
6.5.3.2	Refining Grids	124
6.5.3.3	Sequential Coarsening and Refinement	126
6.6	Resolution-dependencies in the Regularizers	127
6.7	Conclusion	130
7	Validation on Natural Anatomies	131
7.1	Experimental Setup	131
7.1.1	Hardware Details and Execution Time	132
7.2	Quantitative Analysis - Mark II arm phantom (M2AP)	132
7.3	Full Body Phantom	135
7.3.1	Upper Arm	136
7.3.2	Lower Arm	138
7.3.3	Upper Leg	140
7.3.4	Lower Leg	141
7.4	Cadaver	143
7.4.1	Skull	143
7.4.2	Shoulder	145

7.4.3	Wrist	146
7.5	Scatter Correction	148
7.5.1	Scatter Correction of Phantom Limb Scans	149
7.5.2	Scatter Correction of Cadaver Scans	156
8	Conclusion	161
8.1	Future Work	162
8.1.1	Better ML Informed Prior	163
8.1.2	Predicting Bone Overlap	163
8.1.3	Additional Material Databases	164
8.2	Final Remarks	164
	Bibliography	166
	Appendix A Pseudocode	179
A.1	Bi-directional Greedy Descent	180
A.2	Continuity regularizer	181
A.3	Cost function which the gradient descent aims to minimise	182
A.4	Perturbation to material estimate	184
A.5	Selecting the next cycle case	186
A.6	Finding the gradient of J numerically for the current material estimation	188
A.7	Re-scale a thickness alloy model	189
A.8	Segmentation regularizer	190
A.9	Cycle of optimization at a single resolution	191
A.10	Smoothness regularizer	192
A.11	Cycle of optimization across many resolutions (V-cycle)	193

Declaration

The work in this thesis is based on research carried out within the Department of Computer Science, University of Durham, England. No part of this thesis has been submitted elsewhere for any other degree or qualification, and it is the sole work of the author unless referenced to the contrary in the text.

Research presented in this thesis has been accepted for publication at The 7th International Conference on Image and Graphics Processing.

Copyright © 2023 by Henry Westmacott.

“The copyright of this thesis rests with the author. No quotation from it should be published without the author’s prior written consent and information derived from it should be acknowledged”.

List of Figures

1.1	The first X-ray published by Wilhelm Röntgen (Röntgen, 1895), of his wife Anna Bertha Ludwig's hand, with a ring on her ring finger.	1
1.2	X-ray scan of pelvis phantom, with scatter (left) and with scatter removed via an Anti-scatter grid (ASG). Images taken from Huda et al. (2008)	3
1.3	X-ray scans of two objects with different material distributions in the plane orthogonal to the detector will not necessarily produce noticeably different images. Shading in this image is used to represent optical density/X-ray absorption. The vertical arrows in this figure represent direct (unscattered) photon paths, and are aligned orthogonally to the imaging plane.	6
2.3	Example image from the calibration set for the Full body Phantom dataset.	24
2.4	X-ray image of Mark II Arm Phantom, designed to approximate the anatomy of the human upper-arm. No contrast windowing has been applied.	26

2.5	X-ray images of regions of samples (clockwise from top left: upper-arm, lower-arm, upper-leg, lower-leg), taken from a phantom designed to replicate human anatomy. Each of these images has been contrast windowed to maximise readability.	27
2.6	X-ray images of regions of anatomy (clockwise from top left: shoulder, skull, wrist), taken from a human cadaver. Each of these images has been contrast windowed to maximise readability.	28
2.1	Diagram showing the broadest structure of the problem we seek to solve. The goal is to produce an accurate model that represents the real structure and material makeup of the imaged object, by iteratively comparing the simulated X-ray images of varying models to the true X-ray image. In (A), we see the traditional imaging process, where X-rays are projected through an object to produce a image corresponding approximately to the absorption of the model. (B) shows how we might represent such an object in our model, where only thickness and alloy can be encoded. In (C), an example model is used to produce a simulated X-ray image. Finally, in (D), that simulated image is compared against the real image, and this difference is used to inform that changes that should be applied to the model. In the real system, we will iterate through steps (C) and (D) until the simulated image is a good match to the real. . . .	30
2.2	Flowchart showing the structure of the TrueView algorithm	31
4.1	Highly simplified model of scatter dynamics, representing the ill-posed quality of the problem, where the model parameters relating to one pixel influence the intensities of the other pixels nearby.	48
4.2	Densities of voxels in a $3 \times 3 \times 2$	49
4.3	Results of stochastic gradient descent on test case 2.	54
4.4	Results of stochastic gradient descent on test case 3.	55
4.5	Results of stochastic gradient descent on test case 4.	56

4.6	Results of stochastic gradient descent on test case 2, with an added regularization term relating to the segmentation.	59
4.7	Results of stochastic gradient descent on test case 3, with an added regularization term relating to the segmentation.	60
4.8	Results of stochastic gradient descent on test case 4, with an added regularization term relating to the segmentation	61
4.9	Results of stochastic gradient descent on test case 2, with an added regularization term relating to the smoothness.	64
4.10	Results of stochastic gradient descent on test case 3, with an added regularization term relating to the smoothness.	65
4.11	Results of stochastic gradient descent on test case 4, with an added regularization term relating to the smoothness.	66
4.12	Results of stochastic gradient descent on test case 4, with an added regularization term relating to the smoothness. This figure shows the effect for a central-band, bone pixel instead of an outer-band, soft-tissue pixel. The visible difference is due to the restriction of smoothness along edges.	67
4.13	Smoothness regularizer applied to test-case 4, on a 9x9x2 voxel grid. The graphs on the right show the segmentation for this grid setup (above), and how the edge-detection will respond along one of the three axes (below).	68
4.14	Smoothness and segmentation regularizers applied to test case 2.	68
4.15	Smoothness and segmentation regularizers applied to test case 3.	69
4.16	Smoothness and segmentation regularizers applied to test case 4.	69
4.17	Smoothness and segmentation regularizers applied to a new test case, where all voxels have been alter by a constant value.	70
4.18	Smoothness and segmentation regularizers applied to a new test case, where randomly generated, 1% noise has been added to each voxel.	71

5.1	Down-sampled X-ray image for the Mark II Arm Phantom (left), and the simulation produced by Trueview for that anatomy (right).	74
5.2	Responses in h and max norms to perturbations in thickness and alloy for single voxels, and all voxels simultaneously.	74
5.3	ground truth thickness and alloy information for the M2AP	77
5.4	Result of five cycles (6500 iterations) of unregularized optimization, starting from ground truth information.	77
5.5	h -norm of image residuals and Frobenious norms over \mathbf{t} and $\boldsymbol{\mu}$ for five cycles (6500 iterations) of unregularized optimization, starting from ground-truth information.	78
5.6	Thickness and alloy models estimated from unregularized optimization .	79
5.7	Example segmentation of the mark II arm phantom (left), and the edges along which thickness (middle) and alloy (right) smoothness would not be active.	80
5.8	Result of a failed convergence, hampered by a overly strong regularization over the smoothness.	83
5.9	Appropriately tuned implementation of the smoothness regularizer, leading to a more accurate model than the unregularized case	83
5.10	Five cycles of unregularized optimization from highly inaccurate initial condition. Without the segmentation regularizer, the Frobenius norm over the alloy residuals is incapable of convergence.	86
5.11	Five cycles of optimization with only the segmentation regularizer active, from highly inaccurate initial condition. The presence of segmentation-based regularization has allowed the alloy residuals to converge alongside the thickness residuals.	86
5.12	Example of how the cost function for a given $(\mathbf{t}, \boldsymbol{\mu})$ parameter changes when this regularizer is introduced. The plateau for values outside the physical range is removed, creating a slope that can be descended. . . .	88

5.13	Depiction of invalid alloy values after five cycles of unregularized optimisation.	89
5.14	Five cycles of optimization applied to the M2AP, with continuity regularization over the ℓ_1 applied. This reduces sharp changes in either parameter space, rendering a more organic estimate.	91
5.15	A single iteration of bi-directional greedy descent, where b , is the initial value of $\mathbf{t}, \boldsymbol{\mu}_{x,y}$, and a and c are the modified values in either direction. In this case, the current best estimate will be updated to a , as it has a lower cost than c , despite being a local minimum.	93
5.16	A single iteration of bi-directional greedy descent (subsequent to Figure 5.15), where b , is the initial value of $\mathbf{t}, \boldsymbol{\mu}_{x,y}$, and a and c are the modified values in either direction. In this case, the current best estimate will not be updated, as both a and c have higher associated costs.	93
5.17	A single iteration of bi-directional greedy descent, where b , is the initial value of $\mathbf{t}, \boldsymbol{\mu}_{x,y}$, and a and c are the modified values in either direction. In this case, the current best estimate be updated to c , as c has the lowest associated cost.	94
5.18	Example of a finite-difference gradient estimation for Stochastic gradient descent (SGD). a represents the initial state, and b some small modification made to that state. Both of these states are evaluated, and their difference is used to make a first-order approximation of the gradient at a , denoted here by the red dashed line, g	96
5.19	Normalised Frobenius norms over thickness and alloy residuals (left) and normalised cost function (right) over five cycles of optimization using stochastic gradient descent, without regularization.	97

5.20	Example of the gradient estimation in Newton’s method for optimization. b represents the initial state, and a and c represent small alterations to the positive and negative of b . These three states are evaluated, and used to estimate the second-order gradient at b , such as indicated by the red, dashed line.	98
5.21	Normalised Frobenius norms over thickness and alloy residuals (left) and normalised cost function (right) over five cycles of optimization using Newton’s method, with all regularizers active. From the divergence over the cost function, we can infer that gradient-based techniques are unlikely to be successful.	98
6.1	Run-time comparison of a set of down-sample factors for a femur image, using non-optimised research code. The values on the y axis are not representative of the values for optimised code, but the relationship between the down-sample factor and run-time is expected to be representative. Note that this figure includes additional resolutions (DS67,DS29) to the ones used when processing the images in our algorithms, as they fall between the factor-two intervals we opted for.	102
6.2	An example of a single scatter kernel of a simulated image. The x and y dimensions of this kernel are the same as the image from which it was produced. To produce a simulated X-ray, the element-wise sum is taken of a stack of these kernels, with one kernel produced for each scatter centre within the image. The value on the z axis is a representation of photon-count.	104
6.3	M2AP X-ray image, down-sampled to (a) DS89, (b) DS45, (c) DS23, and (d) DS11.	105
6.4	Graphical depiction of the states involved in down-sampling (coarsening) a spatial model for the first time. Solid arrows represent resolution transitions, and dotted lines represent optimization iterations.	109

6.5	Graphical depiction of the states involved in up-sampling a spatial model for the first time. Solid arrows represent resolution transitions, and dotted lines represent optimization iterations.	112
6.6	Graphical description of a coarsening step followed by a refining one (a V cycle). a , b and c represent the re-scaled states, before any iterations have been carried out at that resolution. a' and b' represent the states at those resolutions after iterations have been performed. Solid arrows represent resolution transitions, and dotted lines represent optimization iterations.	112
6.7	Graphic showing the artifact-preserving process for identifying misalignment between the segmentation at source and target resolutions. In this figure, the red, crosshatched squares represent the segmentation at a coarse, source resolution, and the blue, solid squares represent the segmentation at the fine, target resolution. Part (d) shows areas where the up-sampled source segmentation does not align with the target segmentation. As such, pixels like the one expanded upon in (f) will be assigned the value of the mean of adjacent, similarly-classified pixels. In the case of (f), it's final value will be $\frac{(3)+(6)+(7)+(8)+(9)}{5}$	113
6.8	Example of an update to thickness and alloy, calculated at a coarse resolution. For this to be applied to the high resolution model, it needs to be accurately up-sampled.	114
6.9	Thickness and alloy update from Figure 6.8, re-sampled using a injection followed by segmentation-guided bilinear interpolation.	114
6.10	Thickness and alloy update from Figure 6.8, re-sampled using injection.	115
6.11	Thickness and alloy update from Figure 6.8, re-sampled using a injection followed by segmentation-guided local averaging.	116

6.12 Frobenius norm over thickness residuals (left) and alloy residuals (middle), alongside cost function for each interpolation method (right), over the course of many iterations at different resolutions. Dotted vertical lines signify changes in resolutions. 117

6.13 Repeated blacklisting and V-cycles, starting from a fine grid. We can see that after each V-cycle, a further coarse grid is blacklisted, reducing the number of grids available in future, and reducing the length of each subsequent V-cycle. 120

6.14 Convergence in the thickness and alloy residual norms (left) and cost function J (right), reconstructing the M2AP at just the fine-grid level without grid changes. Only minimal convergence can be seen over the residual norms due to the small impact of fine-grid changes. 123

6.15 Convergence in the thickness and alloy residual norms (left) and cost function J (right), reconstructing the M2AP at multiple grid levels, sequentially refining from coarse to fine. The first few iterations make a sharp reduction in both thickness residuals and cost function. The sharp increase in cost function when refining onto the finest grid (at approximately iteration 3100) comes from an issue normalizing the cost function at different grid levels, and does *not* represent deterioration of the model. 125

6.16 Convergence in the thickness and alloy residual norms (left) and cost function J (right), reconstructing the M2AP at multiple grid levels, where refinement and v-cycles of coarsening followed by refinement are implemented in response to relative changes in the cost function. The initial reduction in the thickness and alloy norms is slower than in Figure 6.15, but the convergence over the whole reconstruction run is smoother, and reaches a lower residual norm in both thickness and alloy. 126

7.1	Frobenius norm over thickness any alloy residuals against ground-truth data (upper-left), cost value (upper-right), final estimated thickness (lower-left) and bone thickness $((1 - \mu) * \mathbf{t})$ (lower-right) for the M2AP, initialised at 15cm thickness and alloy guided by the segmentation.	134
7.2	Frobenius norms over residuals between reconstructed thickness and alloy values and the reconstructions obtained by IBEX Innovations (left), and the cost function per iteration during the iterative reconstruction process. (right) Vertical dashed lines represent points where grid resolution was able to change.	136
7.3	Final estimated thickness (left) and bone thickness $((1 - \mu) * \mathbf{t})$ (right) for the upper arm phantom, initialised at 15cm thickness and alloy guided by the segmentation.	137
7.4	Frobenius norm over thickness any alloy residuals against gold-standard data (upper-left), cost value (upper-right), final estimated thickness (lower-left) and bone thickness $((1 - \mu) * \mathbf{t})$ (lower-right) for the lower arm phantom, initialised at 15cm thickness and alloy guided by the segmentation.	139
7.5	Frobenius norm over thickness any alloy residuals against gold-standard data (upper-left), cost value (upper-right), final estimated thickness (lower-left) and bone thickness $((1 - \mu) * \mathbf{t})$ (lower-right) for the upper leg phantom, initialised at 15cm thickness and alloy guided by the segmentation.	141
7.6	Frobenius norm over thickness any alloy residuals against gold-standard data (upper-left), cost value (upper-right), final estimated thickness (lower-left) and bone thickness $((1 - \mu) * \mathbf{t})$ (lower-right) for the lower leg phantom, initialised at 15cm thickness and alloy guided by the segmentation.	142
7.7	Reconstructed surfaces of overall object thickness (left) and bone thickness (right) of the cadaver skull.	144

7.8	Reconstructed surfaces of overall object thickness (left) and bone thickness (right) of the cadaver shoulder.	146
7.9	Reconstructed surfaces of overall object thickness (left) and bone thickness (right) of the cadaver wrist.	148
7.10	Uncorrected (left) and scatter-corrected (right) X-ray scans of the phantom upper arm and corresponding histograms, <i>without</i> contrast windowing to improve readability. It is expected that the reader will struggle to discern visible differences between the images, particularly if the thesis is viewed in print. This figure serves to demonstrate the importance of contrast windowing.	149
7.11	Uncorrected (left) and scatter-corrected (right) X-ray scans of the phantom upper arm and corresponding histograms, with contrast windowing to improve readability.	151
7.12	Uncorrected (left) and scatter-corrected (right) X-ray scans of the phantom lower arm and corresponding histograms, with contrast windowing to improve readability.	152
7.13	Uncorrected (left) and scatter-corrected (right) X-ray scans of the phantom upper leg and corresponding histograms, with contrast windowing to improve readability.	153
7.14	Histograms of the uncorrected (left) and scatter-corrected (right) upper leg scan shown in Figure 7.13, separated according the segmentation of the original image into bone, soft-tissue and open-beam regions.	153
7.15	Highlighted region of interest on the uncorrected (left) and corrected (right) upper leg scan. In this figure, contrast windows for both uncorrected and corrected scans have been set to have equal range, but centred at different intensities.	155
7.16	Uncorrected (left) and scatter-corrected (right) X-ray scans of the phantom lower leg and corresponding histograms, with contrast windowing to improve readability.	155

7.17	Uncorrected (left) and scatter-corrected (right) X-ray scans of the cadaver skull and corresponding histograms, with contrast windowing to improve readability. Note that the histogram binning in this figure and the equivalent histograms in Figures 7.18 and 7.20 is chosen arbitrarily. These images are arrays of float values, and as such there is not a binning that corresponds to the precision of the images. Instead, we opt to use the same histogram binning across all histograms in this chapter.	157
7.18	Uncorrected (left) and scatter-corrected (right) X-ray scans of the cadaver shoulder and corresponding histograms, with contrast windowing to improve readability. To reiterate from the caption of Figure 7.17, the low precision in the histograms does not indicate low precision in the images.	158
7.19	Highlighted region of interest on the uncorrected (left) and corrected (right) shoulder scan. In this figure, contrast windows for both uncorrected and corrected scans have been set to have equal range, but centred at different intensities.	159
7.20	Uncorrected (left) and scatter-corrected (right) X-ray scans of the cadaver wrist and corresponding histograms, with contrast windowing to improve readability.	160

Nomenclature

SGD Stochastic gradient descent

BGD Batched gradient descent

M-BGD Mini-batched gradient descent

M2AP Mark II arm phantom

CT Computed tomography

PMMA Polymethyl methacrylate

LAC Linear attenuation coefficient

SWF Scatter window fraction

SA Simulated annealing

CNN Convolutional neural network

GAN Generative adversarial network

ASG Anti-scatter grid

FMG full multi-grid

MC Monte Carlo

CBCT Cone-beam computed tomography

Introduction

1.1 Context

Since their discovery by Wilhelm Röntgen in 1895, X-rays have been a valuable tool in medicine, thanks to their ability to reveal internal details of the body.



Figure 1.1: The first X-ray published by Wilhelm Röntgen (Röntgen, 1895), of his wife Anna Bertha Ludwig's hand, with a ring on her ring finger.

Because they can pass through soft organic tissues relatively unobstructed, but are reflected and absorbed by bone, X-ray imaging has become the first point-of-call in identifying issues relating to bone health, from fractures to arthritis.

While other, more complex imaging techniques have developed since, the low-cost and versatility of projectional radiography has kept them relevant in the 100+ years since their discovery.

In the last 50 years, one more elegant imaging technique that has developed out of X-ray imaging is Computed tomography (CT), where a series of X-ray images are taken from a range of angles, then processed computationally to generate a three-dimensional model of the anatomy or object in question.

While highly valuable, CT scanning comes with associated downsides as well as benefits. First, with any X-ray imaging process, there is an associated radiation cost to the patient. An occasional X-ray image is a minor radiation dose. Even on a highly-absorbing part of the anatomy such as the chest, a planar X-ray will generally give a patient a radiation dose of approximately 0.02mSv, comparable to the background radiation one would absorb over the course of 20 days. Comparatively, a CT scan of the chest will typically expose the patient to around 8mSv, which is on-the-order of background radiation dosage a typical person will experience over the course of three years (Wall and Hart (1997)).

Furthermore, CT scans are more time consuming for radiographers, require more expensive and bulky equipment, and are more stressful for patients (Heyer et al. (2015)).

There is, evidently, value in the high-quality, three-dimensional imaging that is possible with CT. But the associated costs and barriers mean that for most patients, most of the time, planar X-ray projection will be the imaging technique with which their problems are evaluated.

Thus, it is important to consider the disadvantages of planar X-ray imaging, and to research the ways in which these downsides can be mitigated, if at all possible.

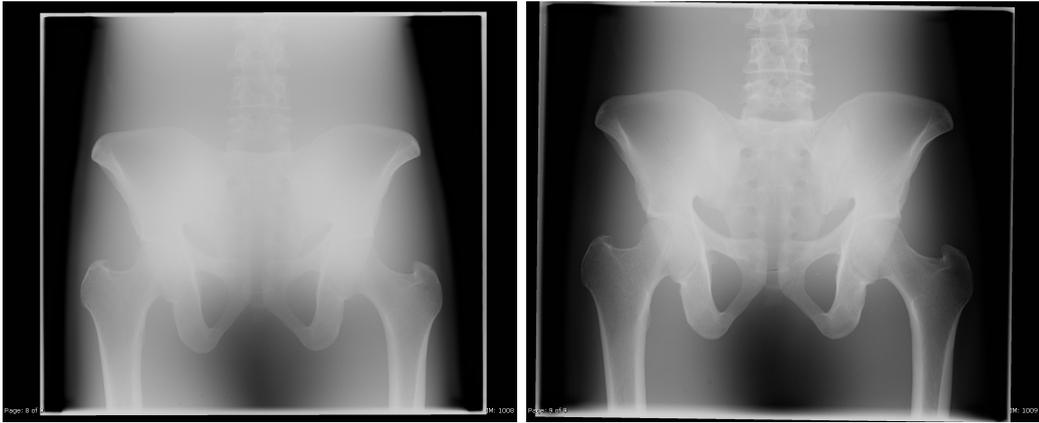


Figure 1.2: X-ray scan of pelvis phantom, with scatter (left) and with scatter removed via an ASG. Images taken from Huda et al. (2008)

1.2 Problem

One substantial downside to X-ray imaging is that of scatter. In any kind of optics, when photons are passing through a non-vacuum medium, there is a chance for those photons to interact with the substance they are passing through, and divert course. For photons at X-ray energies, Compton scattering is the dominant cause of this effect (Kelsey, 1984; Yao and Leszczynski, 2009).

In an imaging setting, these diverted photons represent noise. They have followed a path that does not directly correspond to their position in the image, so the data they carry is not valuable. In a bulk effect, this leads to a clouding of the image, and reduced contrast. This contrast-reducing noise is referred to as scatter. Scatter is an inherently low-frequency, highly blurred effect. It is the impact of a many errant photons travelling along unpredictable paths, thus well-defined edges in the scatter are not only rare, but unrealistic in the majority of cases.

This effect becomes more substantial in an image such as a chest X-ray, where a medical professional might be trying to discern a fracture in a rib, despite varying densities in tissues either side of it, and a second layer of ribs above/below. In this situation, even minor reductions in contrast can make a complex image very time consuming to evaluate. To further complicate things X-rays passing through a

larger part of the body (such as the chest, compared to, e.g. the wrist) have more matter to interact with and scatter off, leading to increased levels of contrast-reducing scatter.

From this context, we can see that the value that an accurate scatter correction - one that replicates the effect of an anti-scatter grid, improving readability - can add to an image is significant. Because of this, techniques have developed over time to attempt to remove or correct-for scatter in X-ray imaging. These techniques will be discussed in detail in Section 3.3.2, but we can consider an overview here for the sake of contextualisation.

1.3 Objectives

Our work seeks to improve the state-of-the-art in X-ray scatter estimation, by producing a three dimensional model of the anatomy being imaged, from which Monte Carlo (MC) simulations of X-ray scatter can be produced. Specifically, we want to include information about the internal structure of the anatomy in these models, and use knowledge about the approximate structure of typical anatomies to attempt to resolve this 3D model in an iterative way. This is achieved by constructing an optimisation problem that converges towards an accurate model that closely aligns with the ground truth data, and produces a beneficial and realistic X-ray image.

In doing this, we more broadly seek to show that an ill-posed problem such as reconstruction of a 3D body from a 2D measurement is possible through regularized optimization, and can be used to produce valuable outputs.

Scatter correction as an optimization problem allows us to use the expectations we have about the model — based on simple anatomical features — to constrain the problem enough that a solution can be found despite uninformative initial conditions.

Crucially, our underlying task is to produce models of imaged bodies that are appropriate for use with the scatter simulation methods provided to us by IBEX Innovations (as will be further discussed in Chapter 2). In this work we do not attempt to improve the quality of scatter simulation, instead trying to resolve models of appropriate type for use with IBEX software in a robust, explainable way, using inferred knowledge about anatomies.

Finally, we seek to produce these models in as short of a time frame as possible, so that their outputs (as scatter estimates and, by extension, scatter-corrected images) can be valuable in a clinical setting. We achieve this through the use of multi-scale analysis, which makes rapid progress on coarse models before resolving detail more slowly on finer models.

1.3.1 Challenges

Resolving even coarse 3D models of bodies from single, noisy X-ray images is not a trivial task. As will be reinforced in Chapter 2, the information contained within an X-ray image tells us nothing about the spatial distribution of material orthogonal to the imaging plane.

Part of the way that this issue is combated has been provided to us by IBEX, in their choice of model. The models that IBEX use to represent an imaged body and produce simulations from contain no information about the distribution of materials in the plane orthogonal to the image. These models do parameterise the depth of the whole object along that imaging vector, but do not make any estimation of the varying distribution of materials along that dimension. Instead, varying materials along that vector are homogenised into a single alloy parameter, which describes the proportional combination of tissue types that occur in that region. This leads us to the important limitation that though our technique is seeking to produce a 3D model, this model contains very limited information and is *not* comparable to the type of reconstruction produced in CT imaging.

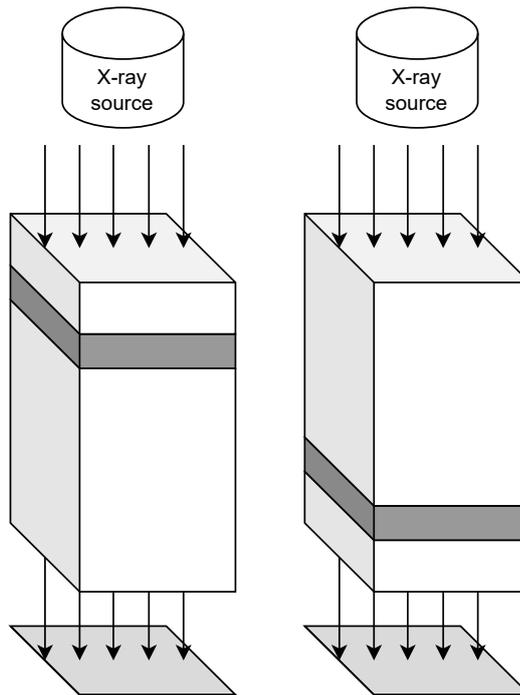


Figure 1.3: X-ray scans of two objects with different material distributions in the plane orthogonal to the detector will not necessarily produce noticeably different images. Shading in this image is used to represent optical density/X-ray absorption. The vertical arrows in this figure represent direct (unscattered) photon paths, and are aligned orthogonally to the imaging plane.

1.4 Contributions

The product of our research is an algorithm that produces thickness-alloy models of imaged anatomies, suitable for X-ray simulation and by extension scatter correction. The models are produced in an iterative method, where *inaccurate* models are gradually refined over many iterations, to produce increasingly accurate estimates of thickness and alloy values. Structuring this problem as an optimization problem — and allowing it to be solved iteratively — is achieved through the development of functional regularizers, custom to the problem of resolving structure of 3D anatomies. Thus, we not only present a technique for resolving models of X-ray imaged anatomies, but more generally for solving ill-posed problems via injection of functional domain knowledge. Further, we implement this solution in a multi-scale fashion, and show that further regularization and efficiency can be

gained through the use of a range of coarse and fine resolutions.

1.4.1 Novelty

To our understanding, no other work has attempted to apply domain knowledge as regularization to constrain the problem of 3D reconstruction from X-ray images. Other work has used regularization for improved reconstruction in CT imaging (Niu et al., 2014), but this is typically based around the expected smoothness of the output images, as opposed to the reconstructed model itself.

Reconstruction of the surface of an object from a single image is an active area of research, and has applications in a wide range of areas, such as autonomous vehicles and robotic sensing. However, such methods typically have fewer limitations than we see in our problem, making use of photon time-of-flight, multiple viewpoints, or many sequential images, such as taken from a recorded video. (Niu et al., 2018), for example, presents a method using Convolutional neural network (CNN)s to reconstruct a 3D model from an RGB image, with topological understanding as part of that reconstructed model. (Nguyen et al., 2021) demonstrates accurate 3D reconstruction from a single image by making use of speckle images. We note here that part of the novelty of our work is that we seek to reconstruct multiple parameters about the object (this can be thought of as a body surface and bone surface being reconstructed simultaneously), with access to data that contains information about internal structure.

In the field of scatter correction, recent work has attempted to estimate and remove scatter from medical X-rays via deep neural networks, and has demonstrated success (Lee and Lee, 2019). This work simulates artificial X-ray images (with known scatter and direct-beam components), and trains CNNs to predict the high- and low-frequency components of this scatter. This however provides no understanding of the internal structure of the imaged anatomy, and would need to be retrained for different detectors and set-ups.

1.5 Limitations

There are a number of limitations imposed on us by the decision to interface directly with pre-existing code from an industrial sponsor, alongside self-imposed limitations to prevent the scope of the project expanding beyond the point of achievability.

First, our 3D models of the imaged bodies will not attempt to capture true information about the distribution of materials in three dimensions. This comes down to the challenge described in 1.3.1, that a two dimensional image does not contain the required information for full 3D reconstruction in a readily-accessible way. We do, however, seek to extract *as much* information about 3D structure as possible from the individual X-ray images.

Second, we impose a limitation on ourselves that our reconstructed models, and by extension scatter estimates, will be produced at significantly lower resolutions than the provided X-ray image. In Chapter 5, we will demonstrate that the computational cost - defined simply as the amount of CPU time required - of producing a single simulated image increases polynomially with the number of parameters in our model. Further, the total number of iterations required to produce an accurate model also increases with the number of parameters. As our models are inherently tied to images and their resolutions, the only way to decrease the number of parameters is to decrease the resolution of the model and the images it produces. Thus, we typically try to find models that produce simulated X-ray images at a down-sample factor of between 100 and 20, compared to the input X-ray. The modelled anatomy will therefore also be sampled at a lower rate than in the X-ray image.

Third, we seek to produce these down-sampled models in the shortest practical time. This algorithm ultimately seeks to improve the quality of X-ray images used in a clinical setting (as well as providing additional information). Thus, the results of the processing should be available to the clinician as rapidly as possible. This

limitation encouraged the work detailed in Chapter 6, where we seek to use the down-sampled nature of the model to our advantage and reduce run-time.

Fourth, we restrict ourselves to easily available computational hardware for running the algorithm. This limitation ties into the second, as producing simulated X-ray images at higher resolutions requires rapidly increasing quantities of RAM on the device producing the simulated images. While a product consisting of a cloud-based version of this algorithm is conceivable, what we seek to produce should be suitable for implementation on devices within clinical settings.

1.6 The Structure of the Thesis to Follow

Here we provide an overview of the chapters that comprise this thesis.

1.6.1 Chapter 2 - Problem outline

In Chapter 2 we describe the specific context for the current problem, in terms of the work carried out by IBEX Innovations. Our work is designed in conjunction with industrial research, specifically the IBEX TrueView algorithm. As such, it is important to make the reader aware of the capabilities and limitations of this system.

In this chapter we also describe the segmentation method that we make use of, which was produced for IBEX Innovations specifically, and informs a significant proportion of our solution to the physics-based material estimation problem.

1.6.2 Chapter 3 - Background

With the industrial context of our work better defined, in Chapter 3 we describe the state of related work in various areas surrounding our problem space. This includes the state-of-the-art, as well as the related work surrounding the ideas

we make use of. Novel work and seminal textbooks from research on simulators, optimization, X-ray imaging and scatter dynamics, segmentation, and multi-grid are all discussed and presented, along with work that fills the niches in their various crossovers. Finally, we will conclude by re-stating the relevance of these research areas to our problem, and carving out the outline of the research we are exploring in this thesis.

1.6.3 Chapter 4 - Exploratory Research

In Chapter 4 we begin to outline the bulk of the experimental work carried out while researching for this thesis. We begin by describing a toy problem, which was constructed to allow for rapid testing of several of the key ideas from which our research began. This toy problem is designed to mimic the relevant and interesting properties of the problem we seek to solve, without the necessarily high computational cost of true X-ray simulation.

We then explore the concept of regularization as it applies to our problem, and define a set of functional regularizers based on domain knowledge for implementation in the exploratory toy problem.

1.6.4 Chapter 5 - Regularized Descent

In of Chapter 5 we translate the exploratory work of Chapter 4 into the TrueView modality, where the same principles can be applied to the real X-ray imaging problem, with real images and accurate X-ray simulation. Most importantly, this is where we detail the functional regularizers we have constructed, as they are applied to X-ray images of cadavers and anatomical phantoms. Each regularizer is presented independently, and justified mathematically as well as empirically according to the results of ablation studies.

We then present a range of iterative descent method algorithms, explaining why we opted for the non-conventional one implemented in the experiments presented in

this chapter. We compare our technique against two standard methods: Stochastic Gradient Descent and Newton's method.

1.6.5 Chapter 6 - Multi-scale Methods

In Chapter 6, we discuss how we use ideas from multi-grid analysis to provide additional regularization and efficiency to our algorithm. We detail the way that our algorithm is set up to use different resolutions, and the limitations imposed on us. Crucially, we explain how changing resolution is beneficial to the processing pipeline we have developed, and how we translate the model developed in chapter 5 between resolution grids that do not necessarily align. We then consider how these resolutions can be sequentially ordered for most effective use, and how our regularizers need to be re-considered to be effective across all resolutions.

1.6.6 Chapter 7 - Results

In Chapter 7 we present results to validate our algorithm, by reconstruction the spatial models of artificial objects designed to resemble human limbs (known as phantoms), as well as X-ray scans of complex regions of a human cadaver. These reconstructions allow us to evaluate the successes and limitations of our algorithm's results.

We follow these reconstructions with the results of scatter correction, based on the scatter estimations produced from the models. These corrected X-ray images show that, in cases where scattered radiation is substantial, there is a noticeable impact on the final images to reduce haze and bring the whole object into a tighter intensity range.

1.6.7 Chapter 8 - Conclusion

Finally, in Chapter 8 we conclude the thesis, with a summary of the work presented and our proposed steps for future work that would build on this research.

Problem Outline

Fundamentally, we seek to resolve the internal structure of an anatomy based on a single X-ray image. At its core, this problem is ill-posed, as the data contained in an X-ray image tell us only how many photons were absorbed along their paths, and not where along this path the absorption occurred, or what rate this absorption occurred at. The fundamental idea to this, that we will refer back to is this:

In a single X-ray image, it is impossible to distinguish between a narrow, highly absorbent material, and a deep, transmissive one.

We are provided with a grey-scale image, with pixel values representing the number of photons detected at that corresponding point on the detector – typically in the range 0 to 10000. This tells us only which regions of the object are more opaque in the X-ray spectrum relative to the plane of the detector, and which regions are not.

Despite this fact, medical professionals are able to infer complex three-dimensional information from X-ray images, and do so frequently. This is because they have access to a significant body of domain knowledge regarding the anatomies being imaged. A radiologist would likely struggle to accurately predict the shape and internal structure of a machine part being imaged with X-rays — they are not able to miraculously solve an ill-posed problem — but they can make accurate inferences based on the prior understanding of anatomy that they have. Indeed, most human

observers who have encountered X-rays in their lifetime would be able to make some basic inferences about the 3D structure of the imaged anatomy, based only on their experience with human proportions.

In this work, we seek to represent the most simple aspects of this domain knowledge algorithmically, so that an image processing pipeline could also make reasonable assessments about the 3D structure of an imaged anatomy, and return them as a quantified output to the user.

There are two techniques for encoding this domain knowledge, either through a comparative, learned approach based on many examples, or through a functional approach, in which the domain knowledge is implemented in the form of an optimiser. The first requires substantial training data, consisting of pairs of images and their 3D model counterparts. A prediction system (such as a convolutional, deep neural network) could then be trained on these input-output pairs to predict 3D structure for any X-ray image. The downside to this technique is the lack of explain-ability. The prediction system is treated like a black box, and any end-user of the output is given no indication of how the final prediction was reached. For many applications this is a minor issue, but in medical contexts this justification is crucial. This is an idea that we will return to in Chapter 3.

The second, optimisation-based method requires no training data beyond the image that is being resolved (i.e it is a single-shot solution), but it does require accurate models of the physical processes involved in imaging. If the physics is modelled incorrectly, then the result is not reflective of the real anatomy. We are limited in availability of X-ray images, but do have access to an accurate X-ray image simulator, therefore we opted for this second method.

This idea is shown in Figure 2.1. In part A, we have an anatomy being imaged, producing 2D data from an unknown 3D composition. In part B, we take a simple, anatomy-like structure, and represent it's structure with matrices for thickness (the depth of the object relative to the imaging plane) and alloy (a representation of the

materials found along a straight line between source and that point on the detector.) In part C, we feed those matrices, along with simulated X-ray scatter kernels, into an emulator, to produce a realistic X-ray image of the simple distribution made in B. Finally, we compare the simulated image against the real, and attempt to find the simulated distribution which most closely matches the real image. A close match implies that our simulated distribution is an accurate model of the unknown, real anatomy that was imaged.

2.1 Scatter as an Entryway

The scatter issue that is intrinsic to X-ray imaging described in the introduction is where our research begins. X-ray scatter occurs when photons are deflected off their incident path during the X-ray imaging. This leads to them striking the detector at a point that does not correspond to a normally-incident, straight line path, and creating a clouding noise component in the image. This reduces contrast and readability in the final image, but it also serves to encode a 3D representation of the object in the final image, albeit a non-intuitive one. Accurately predicting and removing this scatter component can lead to a clearer image, and can serve as proof of an accurate model of the anatomy.

This is the problem that IBEX Innovations' TrueView technology sets out to solve. By carrying out simulations of the paths of photons through a model of the anatomy being X-rayed, this technology creates a sufficiently accurate replica of the original X-ray image. This simulation can then be broken-down to separate the scatter component from the direct beam component of the image. The scatter estimation alone can then be subtracted from the original, input X-ray image, creating an accurate scatter-correction, without loss of fine-grain detail.

Initially, research was focused on profiling and investigating the current algorithm to identify weaknesses. This was initially done with thought given to reducing processing time, in an effort to minimise the time required to produce such a scatter

correction from a given image. The existing TrueView algorithm was broken down into its core functions, and each of these was profiled to gather an understanding of where bottlenecks occurred. The top-level function that connects these general steps, referred to as the "Main Loop", consisted of a for loop, proceduarally adjusting and evaluating given simulated models, and iteratively making improvements based on this feedback. It quickly became apparent from the profiling that in a high proportion of the iterations, the adjustments made were not valuable, and ended up being dismissed. Furthermore, it was identified that, because each iteration of this loop is wholly dependant on the one before, there was no potential at that time for parallelization.

This immediately led to research into ways that the algorithm could be made more efficient, in terms of making the most use of each step in the iterative process, and to do so in a parallelizable way.

2.2 The Structure of TrueView

At the beginning of the research period, the existing TrueView algorithm was analysed and profiled. This algorithm has been developed over the course of several years by a changing team of developers. As such, understanding the algorithm comprised of discussions with current members of the development team, as well as time spent stepping through it individually, notating the processes going on at any given point. A significant portion of the work underpinning the existing IBEX software is discussed in detail by Lopez et al. (2018).

First, it is important to consider how the TrueView algorithm achieves its goal of removing scatter from X-ray images. The algorithm seeks to create a model of the object being imaged, and simulate the interactions of X-ray photons with this model. This simulation can be separated into direct and scattered radiation, and the scattered component can be subtracted from the input image. More information about the existing TrueView algorithm is published by Targett et al.

(2022), where the software developed by IBEX is applied to mobile chest X-ray imaging. Simulating photon interactions with complex anatomies is not a trivial challenge, and so to achieve this within time-frames that are clinically valuable (ideally sub-one minute), the complexity of the model must be reduced.

Much of the complexity reduction of the process comes from the choice of model used to represent the imaged anatomy. Instead of trying to represent the anatomy as a full three-dimensional, highly detailed model (with many degrees of freedom for minor, result-altering variation) the model is abstracted in ways that reduce detail, but preserve the scatter behaviour. The model is voxelised, and is sampled at a relatively low spatial sampling rate. In other words, while our input X-ray image maybe hundreds or thousands of pixels in each dimension, the model used to represent the image's subject could only be tens of voxels in each dimension. Furthermore, the model does not truly represent depth, but instead stores only the thickness of the object (measured at a normal to the imaging plane), and the alloy of materials along that normal vector between X-ray source and imaging plane. This alloy is a scalar value, representing the ratio of just two materials: Bone and soft tissue. If the straight line path between X-ray source and detector passes through 5cm of bone and 5cm of soft tissue before reaching a given detector pixel, $I_{x,y}$, the model (L)'s voxel would have a thickness of $t_{x,y} = 10$ (5 + 5), and an alloy of $\mu_{x,y} = 0.5$. If it were 10cm of soft tissue and no bone, then the thickness would be unchanged, but the alloy would become $\mu_{x,y} = 1$, and if it were 10cm of bone, the alloy would become $\mu_{x,y} = 0$. Remember that, because the input image has been down-sampled considerably, the pixels referred to here (x, y) are not true pixels on the X-ray detector, but local groups of pixels. These two matrices of values give us our full model, $L(\mathbf{t}, \boldsymbol{\mu})$.

It is important to consider here the limitations of this model. Firstly, it contains no information about the distribution of materials within the third dimension, orthogonal to the imaging plane. It cannot indicate whether a bone is above a region of soft tissue or vice versa, only that that cross section contains both of those materi-

als in a given ratio. This is why we use the term alloy, as it implies a homogeneous mixing, which is the assumed state of the two materials. This has impacts on the accuracy of the simulated image, as hard boundaries introduce scattering effects, and no such hard boundaries can be considered along this dimension. Secondly, it represents only two types of biological tissue, which is clearly unrepresentative of real anatomies. If we consider a chest X-ray, for example, the incident photons will interact with bone, muscle, dense muscle in the heart, air gaps within the lungs and more. For the sake of reducing the degrees of freedom of the model, any material other than bone is abstracted as soft-tissue.

Further reduction of the model's complexity comes during the simulation, and will be discussed later in this chapter.

The TrueView algorithm iteratively makes alterations to this down-sampled, abstracted model of the imaged anatomy, trying to resolve a model that is an accurate representation of the real anatomy. The structure of that iteration is as follows:

- All pre-computed data is loaded. This includes pre-simulated databases of radial scatter kernels, reification data, and the X-ray image(s) in question.
- The segmentation neural network (X-net, Bullock et al. (2019)) is used to predict bone, soft-tissue, open beam and collimator segments of the input image(s).
- An initial estimate for thickness and alloy is produced, based on the result of the segmentation. This assumes a constant thickness across the object, and a constant alloy in each of the soft tissue and bone regions. The Main Loop begins.
- The first scatter estimate is produced. To do this:
 - For each pixel in the down-sampled X-ray image, I , a ray-tracing algorithm determines the distance to all other pixels, and the combination

of materials each of these straight-line paths pass through, based on the model of alloys, $\boldsymbol{\mu}_{x,y}$.

- This combination of distance and alloy (i.e. ratio of bone to soft tissue on the straight line path) is looked up in the database of scatter kernels, attempting to find the nearest match to those two parameters. This gives an estimate of the scatter kernel for pixels that have scattered along that particular path.
 - These simulated kernels are combined to form an estimate of scatter for the entire image.
- A Direct-beam estimate can then also be produced, using a similar method but without the need for the ray-tracing. Here, for each pixel in the down-sampled X-ray image $I_{x,y}$, the simulated direct-beam intensity is found according to the scatter kernel that corresponds to that pixel's pair of associated thickness and alloy parameters, $(\mathbf{t}, \boldsymbol{\mu})_{x,y}$.
 - These new estimates of scatter and direct-beam are reified, by comparing against real images taken on the detector. This ensures that the simulated images more accurately match the output of the particular device and settings being used.
 - A convergence test is carried out, to determine if the current model of the object is sufficiently accurate to produce reliable scatter and direct beam estimates. On the first iteration, this is virtually impossible, as no changes have yet been made to the model from the naive initial estimate.
 - If convergence has not yet occurred, and the defined maximum number of iterations to run through has not been reached, then the algorithm enters into the process of making modifications to the spatial model.
 - A history matching process is carried out, using information from the segmentation of the raw image and a Gaussian process emulator to produce an

intermediate spatial model, which is the current spatial model with alterations applied.

- A second scatter and direct beam estimate is carried out, using the same simulation process as in the previous steps, using the new intermediate spatial model.
- A new convergence test now determines if the intermediate spatial model has improved upon the one prior.
 - If it has, then the spatial model is updated to the values of the intermediate, and a Boolean flag is set such that, on the next iteration, the first scatter and direct beam estimate is not calculated, since it would be equivalent to the one that has been used to evaluate the intermediate spatial model.
 - If the convergence test statistic has not been improved by the intermediate spatial model, then the outgoing spatial model is updated to the average of the new intermediate spatial model, and the previous best estimate (according to the test statistic), with the opposite sign to the current. This is to say, that if the test statistic determines that the new intermediate spatial model is an *over*-estimate of thickness or alloy, then it will be averaged with the best previous *under*-estimate.
- The current iteration number increments by one, and the loop continues.

This structure is shown in flowchart form in Figure 2.2.

In this existing algorithm, profiling work quickly identified the scatter estimation based on the spatial model as the most computationally expensive step. This was an unsurprising discovery, but informative when combined with the knowledge that on a significant number of iterations, the spatial model has not improved, and thus extra scatter computations have to be performed.

Furthermore, because of the nature of the algorithm's dependence on the previous iteration at all times, there was not potential for parallelization.

Further profiling work was carried out to investigate the relationships between processing-time and the resolution it was carried-out on. Because the scatter estimation technique is based on a simple ray-trace, it scales very poorly with resolution. This is mitigated by the fact that the scatter is fundamentally low-frequency, and thus can be accurately derived from a down-sampled version of the input image. As such, native-resolution X-ray images are never modelled in this work. The scatter estimates are derived from down-sampled copies of the X-ray images, and thus the model themselves also represent down-sampled models of the real anatomies. For context, the standard down-sampling factor in use at IBEX when the research began was a factor of 23. It is understood that this was chosen empirically. In some cases a factor of 11 would be used instead, for situations where the value of the spatial model's thickness and alloy data were being evaluated.

2.2.1 Simulated Databases

For the TrueView algorithm to function accurately, it relies on pre-computed databases of scatter kernels, created by Geant4 simulations. The databases contain the 1D radial scatter kernels corresponding to various combinations of materials, through which simulated X-ray photons pass. The simulated objects through which simulated photons travel and scatter consist of a core cylinder, with a given thickness and alloy, and a cylindrical shell surrounding the first cylinder, also with its own thickness and alloy. This allows the simulation process to capture the physics of scatter in material transitions, where photons are diverted from their path when travelling from one distinct material to another, after being initially scattered by the core material.

For estimating scatter, these kernels are down-sampled according to the resolution desired for the scatter estimate. These kernels are simulated over a range of thick-

nesses and alloys, but for computational efficiency the database is further expanded by emulated kernels, interpolated in the ranges between simulated results. While this produces a less accurate library of kernels than is possible (because not all kernels are strictly the result of simulated physics), it is necessary to reduce the time and memory costs of loading a large database, storing it in memory, and searching it for every kernel lookup.

2.3 Segmentation

The ability to accurately segment an X-ray image into regions of bone and soft tissue is a crucial, underpinning technology of the scatter correction methods in use at IBEX. It is understood that determining the composition of a material made of various densities from a single intensity image is ill-posed, as it is impossible to distinguish a thick, low-absorption material from a thin, high-absorption material.

This ill-posedness is mitigated where we attempt to resolve material information for an object of a single, known density. The segmentation allows us to make such an estimate for regions that have been classified as only containing soft tissue. This confirmed density data can be fed into a history matching algorithm, and compared against ground truth measurement images taken on the given X-ray imaging device, to hone in on the most accurate material estimates for the object in question.

For the purposes of my research, this segmentation has been achieved by a CNN, named XNet, produced by Bullock et al. (2019). This is a network architecture designed in a collaboration between research students and the business, and as such is highly optimised to accurately segment images the images IBEX Innovations typically deal with into the regions of interest to the TrueView algorithm.

Images fed into the segmenting network are monochrome, planar X-ray images, which are classified into four distinct regions:

- Bone, indicating pixels where some amount of bone is present.

- Soft tissue, where no bone has been detected, and the organic material is thus entirely muscle, skin, organs etc.
- Open beam, where no absorbing material is detected at all, and the X-ray has reached the detector unaffected.
- Collimator, where sheets of metal have been intentionally placed in the path of the incident X-rays, to narrow the imaged region and minimise dose to sensitive areas of the body.

This network was trained on hand-labelled images, produced by staff at IBEX Innovations.

2.4 Data Description

In this section, we clarify details about the data required for this method. The IBEX TrueView scatter estimation method that our work is built around requires detector measurements that are taken prior to creating a scatter estimation for an X-ray image. These measurements only need to be taken once, and can then be stored and loaded in at run-time whenever a scatter estimation is made for images captured on that device.

2.4.1 Device Measurements

For an accurate X-ray simulation to be produced, the computational simulation must be calibrated to the X-ray detection machine it is trying to replicate. Failure to properly calibrate in this way would lead to scatter simulations that were consistently over- or under-bright for a given detector, and the simulated scatter would thus degrade image quality when subtracted from the raw image.

The solution that IBEX have developed for this calibration step is to image a series of plastic and aluminium plates of known thickness. Simulated scatter kernels

corresponding to X-rays passing through equivalent blocks of Polymethyl methacrylate (PMMA) and aluminium can then be compared against the true imaged results, and the simulated images can be re-calibrated. By capturing and storing ground truth detector data, and accessing it during the simulation algorithm, the simulated X-ray images can be calibrated in a non-uniform way, as opposed to a calibration applied only to the simulated kernels. By selecting this method, IBEX have allowed their X-ray simulation to calibrate for bright and dark regions of the detector, and well as inter-detector variability.

For each of the datasets presented in this thesis, there is an accompanying set of calibration images, comprising of ten images of aluminium plates, varying in thickness from 0.5 to 5cm, ten images of PMMA plastic, varying in thickness from 1.5 to 20cm, and one image of the unobstructed detector, with nothing placed between source and detector. an example of such a calibration image is included in Figure 2.3.

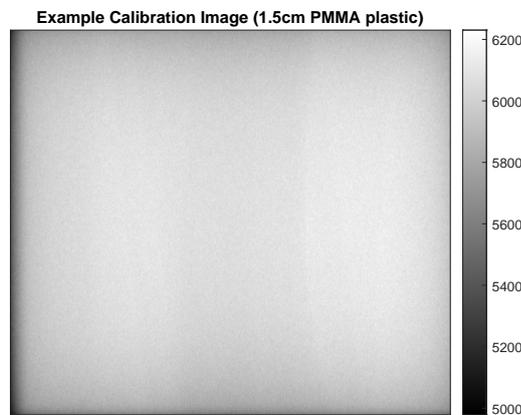


Figure 2.3: Example image from the calibration set for the Full body Phantom dataset.

Additionally for each dataset we have a required materials database, containing the simulated scatter kernels through different material combinations at a given down-sample factor, and a reification database containing statistics about the detector and imaging setup, such as the distance between the X-ray source and detector, which is also resolution-specific. Both of these required databases have been pro-

duced and supplied by IBEX and are essential to their X-ray simulation tool.

Because of the requirement on this device-based data for calibration, we are limited with the datasets we have access to. The calibration plates, though relatively simple, have been designed and manufactured by IBEX. This means that, at time of writing, the only way for this calibration data to be generated is for IBEX staff to image these calibration plates on a detector of interest.

For my research project, the outcome of this is that I have limited access to datasets that I can process with my algorithm. Publicly-available datasets published online will not be suitable, because the IBEX X-ray simulation that is essential to the algorithm I have produced will not correctly calibrate to images without the ground truth data. Any material estimations created for such publicly-available datasets would therefore have biases that cannot be measured, and the results would be unreliable.

2.4.2 Datasets

Because of these restrictions on the images that can be processed with our algorithm, we discuss here the existing datasets that we have access to for development and testing.

Mark II Arm Phantom (M2AP)

The M2AP is a simple construction, designed by IBEX, to mimic the scatter dynamics we might expect to observe in a human arm. Made of aluminium and PMMA, reasonable analogues for the X-ray scatter dynamics of bone and soft-tissue respectively, the M2AP is expected to display similar amounts of X-ray scattering that we would expect to observe when imaging the human upper-arm.

The benefit to using this device, as opposed to the phantoms that are discussed below, is that the M2AP is machined to known dimensions, and is simple to model. Its uniform thickness along its length make it easy to observe differences in scatter and detected X-ray intensity at different regions of the image.

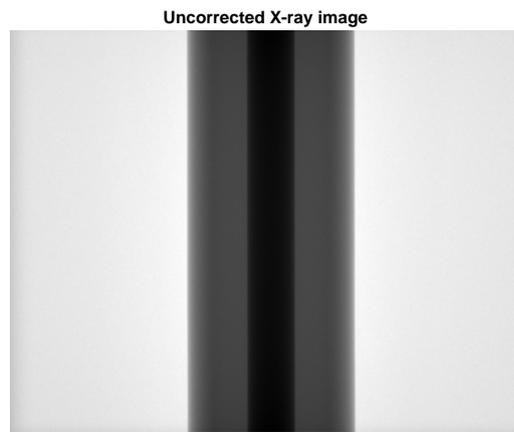


Figure 2.4: X-ray image of Mark II Arm Phantom, designed to approximate the anatomy of the human upper-arm. No contrast windowing has been applied.

Full body phantom

Unlike the M2AP, phantoms are deigned to more accurately represent the variation in shape that are observed when imaging the human body. The dataset referred to as the full body phantom consists of X-ray scans of the upper and lower arms and legs of this phantom, providing data that should closely mimic the X-ray interactions with real human limbs. the X-ray scans of the full body phantom are included below, with some contrast windowing to enhance readability.

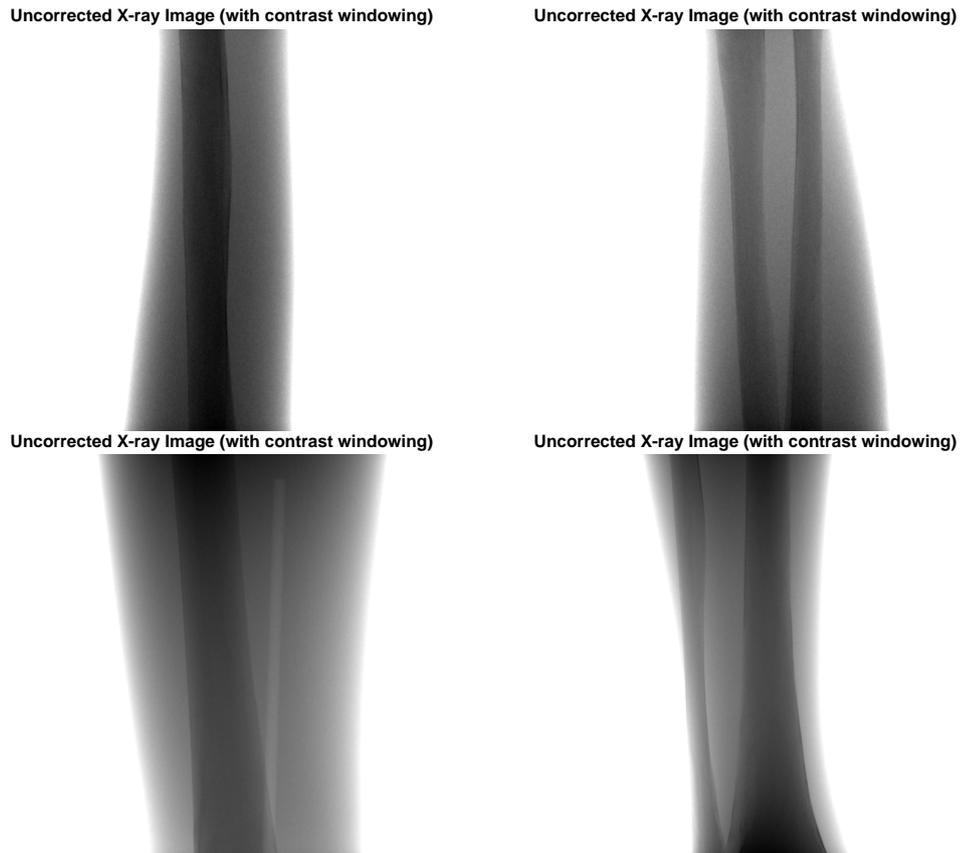


Figure 2.5: X-ray images of regions of samples (clockwise from top left: upper-arm, lower-arm, upper-leg, lower-leg), taken from a phantom designed to replicate human anatomy. Each of these images has been contrast windowed to maximise readability.

Cadaver scans

The final dataset we have access to is a set of scans of a human cadaver, acquired by IBEX for the purpose of assessing scatter correction algorithms. This dataset contains images of regions of anatomy that are harder to model in a phantom, and where scatter effects are more significant.

This dataset consists of images of a wrist, shoulder and skull. Images from this dataset, contrast windowed but without any scatter correction applied, are included below. Compared to images from the full body phantom dataset, these images, and the shoulder and skull in particular, are harder interpret. This is primarily the significant impact of scattering radiation, which makes it difficult to find a satisfactory

contrast window that allows us to examine the whole image simultaneously.

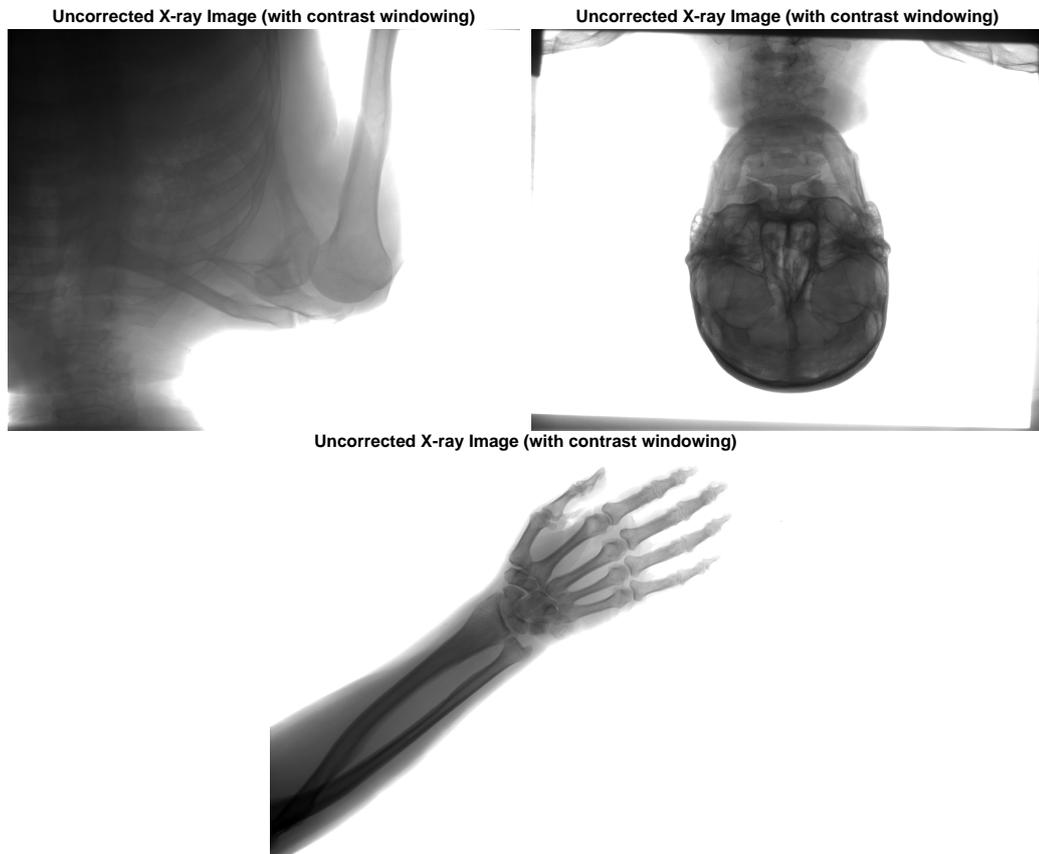


Figure 2.6: X-ray images of regions of anatomy (clockwise from top left: shoulder, skull, wrist), taken from a human cadaver. Each of these images has been contrast windowed to maximise readability.

Because of the complex anatomy of these samples, and the associated challenge of scatter correction, these samples will be used to ultimately validate the success of our material estimation algorithm. This follows the logic that an accurate material estimation facilitates accurate simulation of the X-rays (via IBEX's X-ray simulation algorithm described previously), and accurate X-ray simulation facilitates meaningful scatter correction of the X-ray images.

In complex images such as these, the effects of successful scatter correction should be visually clear to the reader, thus in the absence of ground truth material data for the samples it is this visual improvement from scatter correction that will validate our success.

The other alternative for validation of our method's success would be CT data collected for the samples in this dataset at the same time as the X-ray images we will be analysing. Unfortunately, no such CT (or similar) data was made available for any of the datasets we will be using for our experiments.

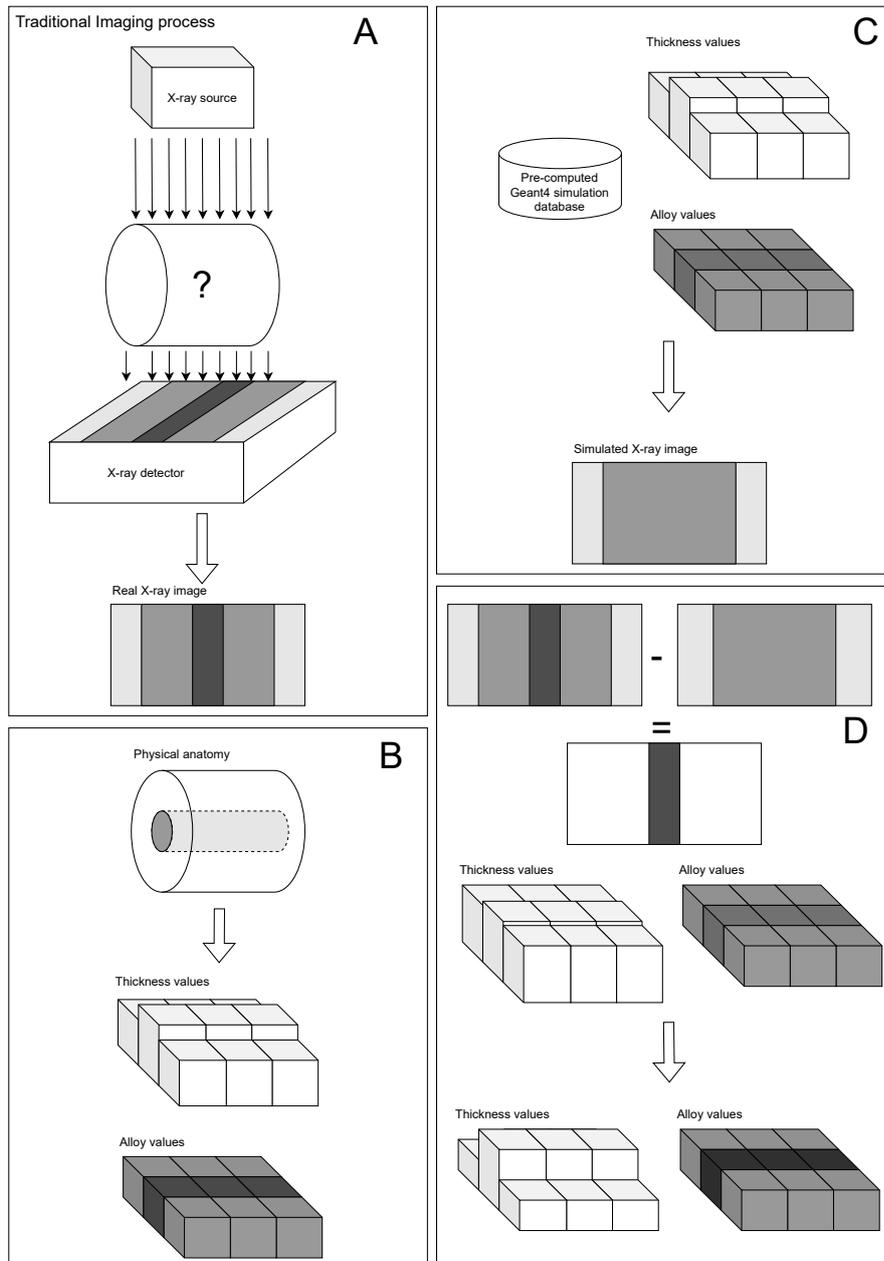


Figure 2.1: Diagram showing the broadest structure of the problem we seek to solve. The goal is to produce an accurate model that represents the real structure and material makeup of the imaged object, by iteratively comparing the simulated X-ray images of varying models to the true X-ray image. In (A), we see the traditional imaging process, where X-rays are projected through an object to produce an image corresponding approximately to the absorption of the model. (B) shows how we might represent such an object in our model, where only thickness and alloy can be encoded. In (C), an example model is used to produce a simulated X-ray image. Finally, in (D), that simulated image is compared against the real image, and this difference is used to inform that changes that should be applied to the model. In the real system, we will iterate through steps (C) and (D) until the simulated image is a good match to the real.

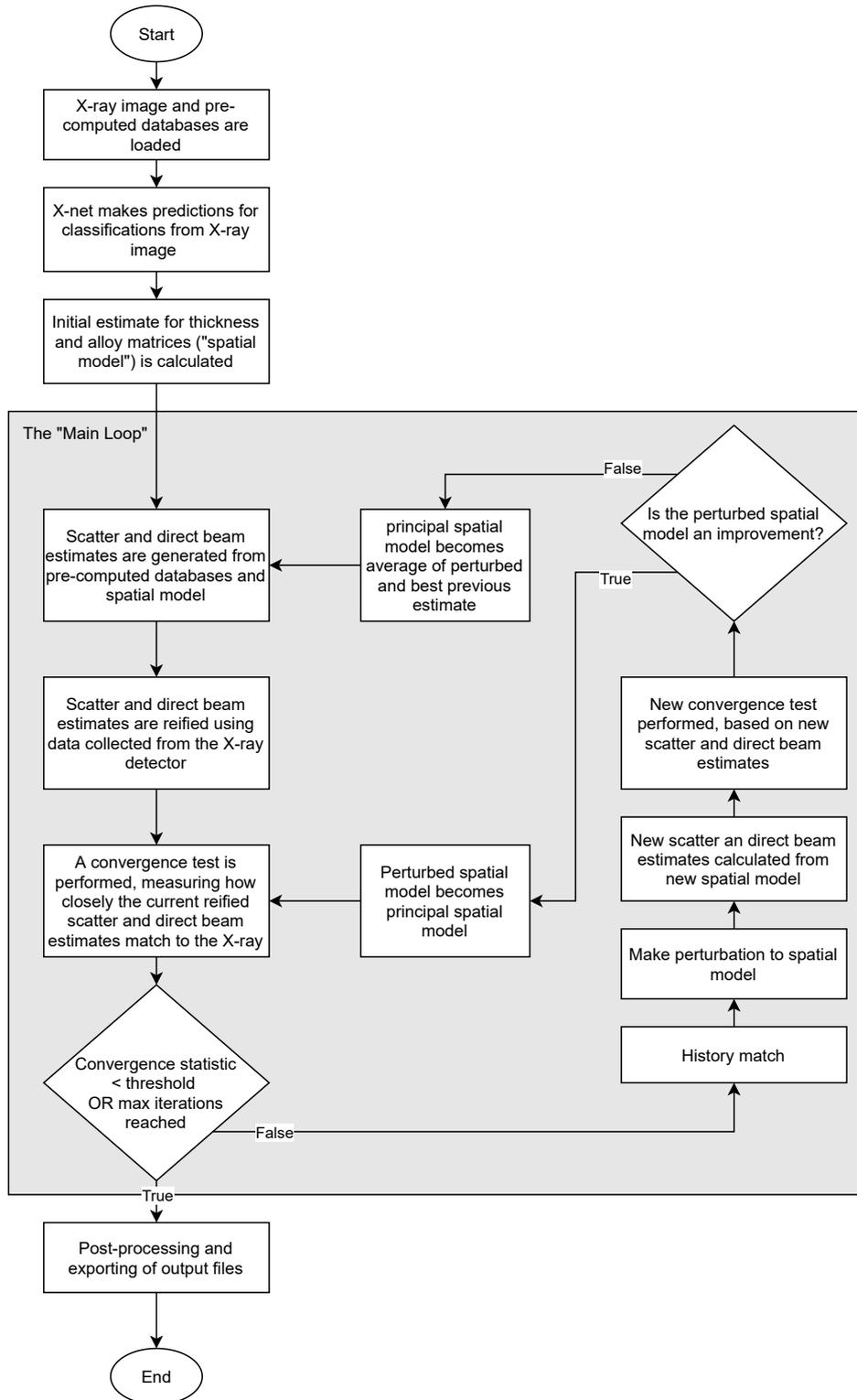


Figure 2.2: Flowchart showing the structure of the TrueView algorithm

Background

In this chapter we present an overview of the work that has informed our research direction, and illustrate the niche that we identified where valuable, novel research could be performed. This chapter includes sections for all the relevant research areas that our work encompasses, and explains the state-of-the-art that we seek to build upon with this thesis.

3.1 Simulators

In this section, we investigate the state-of the art around X-ray simulation techniques. The ability to produce a reasonably accurate simulated X-ray image is essential to our work.

3.1.1 Monte Carlo Simulation

MC methods - named for the gambling culture of the area in Monaco - are based on the principle of repeated, probabilistic sampling. (Prügel-Bennett, 2020) It describes how a probabilistic process may be simulated many times in order to learn more about the properties of that process. This is a technique that is suitable for the simulation of imaging processes, as the photon interactions with matter can be described as probabilistic events. In the field of X-ray imaging, this technique

was employed as early as the late 1990s (Verhaegen et al., 1999), in order to provide dosimetry data that is otherwise difficult to accurately measure.

Later studies showed how this could be used to produce realistic chest X-ray images at many tube voltages, at high resolution with realistic noise (Moore et al., 2012). This research was used to evaluate the quality of images for different body types and different imaging setups by experienced professionals, and gave indication of optimal imaging setups and the patient dose associated with them.

3.1.2 Geant4

Geant4 is a simulation toolkit, designed to model the interactions of particles in many applications (Agostinelli et al., 2003). Originally released in 1998, the Geant4 software was developed in collaboration with CERN, with a goal to produce a robust physics-based particle simulator in the object-oriented programming language C++. Development is ongoing, with the most-recent stable release (at time of writing) on December 10 2021. Geant4 is open-source, and available for free to the public.

One area that Geant4 is well suited to is the simulation of medical imaging modalities, such as X-ray photon interactions. This toolkit allows highly accurate MC simulations to be produced from voxel models, such as in di Franco et al. (2020), where Geant4 is used to produce simulated "clinical-like" mammography images. One limitation on this is the compute time. With the hardware available, di Franco et al. were able to simulate 1.4×10^5 photon interactions per second, but acknowledge that around 10^{10} simulated interactions are needed to simulated realistic breast CT projections.

In comparison, Alnewaini et al. (2017) uses the Geant4 toolkit to produce real-time MC estimations of dose to radiology staff during common imaging procedures. This impressive compute-time for an intensive algorithm such as MC is achieved by

pre-computing components of the interaction which will not change with imaging set-up, such as scattered radiation from the floor and ceiling.

3.2 Optimisation

Based on the dynamics of the problem that we seek to solve, which are set out in Chapters 4 and 5, we understand that we are attempting to optimise an ill-posed, non-convex, discontinuous function of many dimensions. Specifically, we seek to find the optimal set of thickness and alloy values in a spatial model that minimises our cost function, which first and foremost is the difference between a simulated X-ray image and a real image of the object we are reconstructing. This can be thought of as a constrained optimization problem, as we are limited in the selection of values by our simulator, but modifications to our cost function mean that unconstrained optimization is a better definition (Nemhauser et al., 1989).

In this subsection we will set out what is meant by these terms, and how similar problems are being solved in recent research. Non-convex optimisation in particular has taken on renewed research interest in recent years due to its applicability to common machine learning problems (Jain and Kar, 2017).

3.2.1 Ill-posed Problems

The problem that we seek to solve with the work set out in this thesis is ill-posed, as it does not have one stable solution according to the (limited) data available (Kabanikhin, 2011). The ill-posed nature of tomography — that is, the separation of a 2D image space into 3D components — has been well discussed and researched at this point (Lavrent'ev et al., 2014), and iterative reconstruction algorithms for X-ray tomography imaging such as in CT are thought by some to be reaching the limits of their clinical value (Mileto et al., 2019).

Using domain knowledge as part of a system for inverting ill-posed problems has seen recent research, with Maier et al. (2019a). This work reformulates the matrix-operation inherent in CT imaging and factors them into a neural network, thereby injecting prior knowledge about the physics of the problem into the solution. This performs well on CT reconstruction from imperfect data, outperforming other state-of-the-art algorithms. Similar ideas are applied to Cone-beam computed tomography (CBCT) imaging from incomplete data in Würfl et al. (2018).

The novelty that we introduce is the application of iterative, optimisation-based ill-posed problem-solvers to single-shot, 2D X-ray images. This seeks to produce a coarse, low-accuracy—but nonetheless valuable—3D reconstruction.

Bakushinsky et al. (2010) outlines a iterative regularization gradient technique for ill-posed problems.

3.2.2 Regularization

The problem that we seek to solve in this thesis relates to finding the optimal solution in a high-dimensional space. To attempt to make this problem solve-able, we seek to reduce the complexity of the cost function we minimise by employing regularization (Bühlmann and van de Geer, 2011; Engl et al., 1996). This idea is influenced by work by Carola-Bibiane Schönlieb and Ozan Öktem, where knowledge about a forward operator or system is encoded into a cost function as a regularizer (Lunz et al., 2019).

Arridge et al.’s Solving Inverse Problems Using Data-Driven Models (Arridge et al., 2019) describes a variational approach to solving inverse problems of the form $f = T(u) + n$, where we wish to determine u from f , with n denoting noise. This is done by minimising a convex function that is regularised using knowledge about the forward-problem and the type of solutions we might expect.

For image reconstruction, these assumptions could consider the sparsity of the gradients of the image u , such that it is recognised to be generally smooth, with

harsh gradients only occurring at edges in the image. Generative adversarial network (GAN)s and adversarial networks are considered, and their use is proposed as a replacement for the regularisation term. This takes the form of a network trained to recognise poor solutions to the inverse problem and provide a high penalty to the minimisation function in these cases, allowing a deep neural network to be used to determine the optimal parameters for the forward function T .

They describe how the use of deep neural networks can be beneficial in cases of inverse, ill-posed problems where the measurement space and solution space are different. Though this is not the case in our problem, the concepts of regularised gradient descent show promise, with the potential for more complex regularisers to be applied as the problem is scaled up.

Section 6.2 of Arridge et al.'s review has particular relevance to the problem of scatter correction and materials determination, as it describes the approaches that can be applied in the cases on non-perfect forward operators. The current physics engine approach IBEX apply to remove scatter from images is effective, but too computationally expensive to be viable. To mitigate this, a potential avenue of research could be to use expensive but accurate scatter corrections to learn a knowledge-driven, simple analytical model of the forward operator.

Comparable methods have been researched for resolving images based on ill-posed data, through formulation as an optimization problem regularized via the l_1 term (Birdi et al., 2017). Though this work is focused on astronomical interferometry images, the underlying principles are applicable to the work presented in this thesis.

Latha and Sahay (2020) seeks to resolve the 3D structure of objects through regularization from prior knowledge. However, this is achieved through a multi-shot approach, using many images at different focus points to infer depth information.

Physics-informed techniques also fall under the principle of regularization we intend to harness, where knowledge about the physics of the underlying system is used to constrain an optimization (Yang and Perdikaris, 2018; Karniadakis et al., 2021).

Adler and Öktem (2017) consider a technique for functional regularized reconstruction of CT data via learned gradients. Not unlike the work presented in this thesis, Adler and Öktem seek to develop new techniques which use both classical regularization and deep learning. In this work, novel deep-learning techniques are used to partially-learn the gradient mapping of CT reconstruction from simulated CT training data. As the work in this article is concerned with imaging data, it particularly notes the relevance of variations regularization in the ℓ_1 norm (Scherzer et al., 2009), a principle that we consider in Chapters 4 and 5.

Bilal and Arif (2019) present work which seeks to replace traditional functional regularization in image reconstruction with a CNN based method. Similarly, Lunz et al. (2019) investigate how traditional functional regularization encoding knowledge about the model and forward operator can be learned by a neural network via an adversarial approach, as is typically seen in GANs. They then successfully apply this technique to image de-noising in CT.

Some work, such as Gilton et al. (2021), questions the robustness of techniques that seek to learn the forward operator in inverse problem solving. They propose a technique for model adaptation, so make the learned models more accurate in the face of so-called *domain drift*, where the model the work is validated with differs slightly from the one it was trained to invert. We too acknowledge this issue, but instead seek to resolve it by utilising prior knowledge about the expected distribution of estimated values (via a CNN), as opposed to knowledge about the forward operator.

Li et al. (2020) proposes a neural network alternative to classical Tikhonov regularization, and applies it to the problem of low-dose, incomplete CT imaging. This is shown to provide well-posedness to the problem, and is shown to be applicable to related problems thanks to the data consistency term in the proposed solution.

Baguer et al. (2020) considers a combination of deep, learned regularization techniques and more classical regularization for ill-posed problem solving in medical

imaging. This seeks to resolve issues with end-to-end learned methods, such as over-fitting to the training data provided leading to reduced generalizability, while also providing a stronger theoretical backbone to the solution. This makes use of the idea of the Deep Image Prior (Ulyanov et al., 2018), a recent, piece of image restoration research that uses deep neural networks without any training for ill-posed restoration problems.

3.3 X-ray Imaging

3.3.1 Scatter

In X-ray imaging, scattering is a phenomenon that can occur, where photons interact with the matter they are travelling through and change course. This scattered radiation generally leads to a clouded effect on the final image, reducing contrast and making the images less readable by radiology professionals (Dhawan, 2011; Hsieh, 2003). This scatter is, however, a function of the 3D composition of the object that is being imaged, and as such can contain formation that is otherwise inaccessible in a 2D imaging modality (Speller, 1999).

3.3.2 Scatter Removal

Scatter correction in CT imaging, which involves the acquisition of a series of X-ray images taken from different angles around the patient, is increasingly explored in scientific literature. This is likely due to the increased effect scatter can have on CT images, resulting from the different imaging geometry CT requires when compared with standard X-ray imaging. This generally appears to be done via MC simulations (Meyer et al., 2009; Zhao et al., 2016; Lee and Chen, 2015), although neural network based approaches appear to be on the rise (Hansen et al., 2018; Maier et al., 2018), with Maier et al. combining the computational efficiency of deep neural networks with the output of a MC Simulation, achieving good scat-

ter estimates in around 10ms per projection at 384×256 pixels (Maier et al., 2019b).

Non-software based approaches to scatter correction are also important to consider. Any software-based system has to be able to compete with physical scatter rejection techniques to have value. Physical techniques have limitations, but can be a convenient solution for existing systems. ASGs, which have already been mentioned, are the most common of these methods, but an increase in the distance between the patient and the X-ray detector can also lead to a significant reduction in scatter in the cases where it is feasible (Sisniega et al., 2013; Persliden and Carlsson, 1997), as well as the implementation of a bowtie filter (Mail et al., 2009). Combinations of both software-based methods and physical methods have also shown promising results (Stankovic et al., 2017), though could incur the increased dosage requirements of ASGs, which would have a negative impact of their feasibility for medical use when compared to software-only approaches.

Mentrup et al. (2016) compares the scatter-correction capability of an iterative method based on MC simulations of X-ray beams against an ASG. This work specifically discusses bedside chest X-ray imaging, which is routine in hospitals for patients with reduced mobility. They simulate this imaging via a chest phantom, and evaluate the contrast enhancement by placing discs of aluminium inside the phantom, which the imaging system will struggle to resolve under noisy conditions. Their scatter model is based on work by Bertram et al. (2006), where combinations of simulated X-ray pencil beams are passed through water in spheres of different thickness. They iteratively adjust this thickness and minimise the error between the simulated image and the true, then separate the scattered portion of the simulated radiation to remove it from the true image. They conclude that their (relatively crude) software-based scatter correction produces grid-like de-scattering, in a way that is invariant from one patient to the next. Further, it can reduce the patient radiation dosage when compared to ASG scatter removal.

Related to scatter removal techniques, it is also worth briefly discussing the multi

absorption plate (MAP) technology developed by IBEX and Nordson Dage for material estimation in X-ray inspection of manufactured products such as printed circuit boards (Scott and Krastev, 2017; Cowling et al., 2016). The MAP is a hardware device that sits between the X-ray source and object in a conventional inspection set-up. The MAP modulates the incident X-rays in a known spatial pattern, "act[ing] like a complex colour filter for the X-rays". This modulation of the rays changes the energy distribution the object experiences, and allows material information to be extracted in post-processing.

3.4 Segmentation

The term segmentation refers to the process of identifying regions within an image that belong to particular classes, for the purpose of better image understanding. Areas of the image can be separated and classified as distinct from one another, to allow for the identification of objects in a photograph, or valuable data in a cluttered image (Zhang, 2017; Snyder and Qi, 2004).

Recently machine learning techniques have become the most commonly employed method for medical image segmentation. This trend appears to have begun through the rise in classification-based segmentation methods — which segment the image according to a categorical assignment made to each pixel, based on some training data (Norouzi et al., 2014) — has now led to a convention of deep learning techniques, such as CNNs and U-nets and their variations (Hesamian et al., 2019; Jha et al., 2020; Wang et al., 2018).

The most significant caveat to these more modern segmentation techniques is the aforementioned training data. To be able to train a robust CNN, substantial volumes of training data are required, requiring many labelled images available to the segmentation researchers. Sourcing these labelled images can be a time-consuming, impractical task.

As such, techniques to reduce the negative impact of small data sets, such as data augmentation (Zhao et al., 2019) and unsupervised segmentation (Zhang et al., 2020) have begun to emerge.

Yan et al. (2019) for example uses a GAN to produce realistic but synthetic lung X-ray images and their segmented counterparts, for the use as training data in supervised medical image analysis.

3.5 Speed and Efficiency

As will be covered in detail in Chapter 6, we borrow ideas from multi-scale analysis in an effort to regularize the solution and increase efficiency. Acknowledging that we attempt to find a reasonable solution for the physical dimensions and internal structure of part of an anatomy on a discretised grid, we can consider exploring how changing the frequency of that discretization can affect our solution. In doing so we can make use of both the error-smoothing and coarse-grid correction that multi-grid can offer us, as are outlined in Chapter 2 of Trottenberg (2001).

The concepts of algebraic multi-grid described in Stüben (2001) are of interest to us as well, as they allow us to make better use of the extra information provided by a segmentation of our image, which will be discussed in more detail in Section 6.3.2.

Comparable techniques are applied to grey scale images in order to better analyse texture information for classification (Silva and Florindo, 2019). In this work, grey scale native-resolution images are analysed alongside their down-sampled counterparts, as it is recognised that different resolutions will better record data about different frequency features. This is a technique that we try to harness the value of as well.

Recent work has been carried out to apply comparable multi-grid techniques for efficiency in medical image processing, specifically patch-based scatter correction

in X-ray imaging (Gu et al., 2018). In this work Gu et al. use multi-grid ideas to process medical images at coarse resolutions before processing at fine ones. This means that their computationally expensive technique of correcting X-ray scatter based on relatively large patch sizes and search areas can be carried out in reasonable time. This is one of the same principles we make use of in our multi-scale analysis.

Hauptmann et al. (2020) uses multi-scale methods in inverse problem solving to make an unfeasibly slow and memory-intensive problem into a feasible one. Specifically, this work talks about CT reconstruction and the use of neural networks in *learned iterative reconstruction*, where networks incorporate handcrafted features of domain knowledge, as was in the case in Maier et al. (2019a)

3.6 Explainable Artificial Intelligence

Though Artificial Intelligence has seen wide uptake in recent years, it has historically suffered from criticisms around explainability. The application of AI algorithms in safety critical contexts has led to a developing field of explainable AI research, which broadly aims to create AI models that can be reviewed and understood in fields where risk must be minimised. Minh et al. (2022) provides a comprehensive review of the state of explainable AI circa 2021/2022, and identifies three key benefits to additional explainability in black-box models: *"guaranteeing fairness during the learning process, such as identifying and removing the bias in a dataset, improving the system's robustness by indicating the possible noise that could affect [sic] the performance, and ensuring that the model uses only the essential features to infer the output."*

Loh et al. (2022) also reviews the state of explainable AI, but focusing specifically on applications in the medical domain between 2011 and 2021. This review identifies SHapley Additive exPlanations (SHAP) as the most prevalent form of explainable AI for identifying relevant features in healthcare data at time of writing, with 45 of

the 99 studies they consider using SHAP. This review makes clear their view that explainable AI in healthcare is already being rapidly developed and deployed, even concluding that *"many AI models have achieved operator-like performance but are still not used because the lack of operator confidence limits them"*.

Ghassemi et al. (2021) argues against this viewpoint in their paper *The false hope of current approaches to explainable artificial intelligence in health care*. In this work the authors argue that explainable AI methods currently being researched for healthcare applications do not provide the benefits they claim to hold. More specifically, the authors present research to demonstrate that popular methods to make AI explainable, such as saliency maps in imaging applications, are often vague, difficult to interpret, and victim to confirmation bias.

The authors of this review do not set out a viewpoint that is fundamentally opposed to explainable AI. Indeed, they cite the value of explainable AI tools for better understanding bias in datasets and models. They do however make clear their view that in healthcare contexts, explain-ability is an important goal because it generates trust in individual patients that their problems have been approached with sufficient oversight. The author's problem with this is that current explainable AI methods do not help patients understand the process behind decision making on an individual level.

3.7 Depth Estimation and 3D Reconstruction

Also of interest but not related to the sections described above, Sayed et al. (2022) describes a novel technique for depth estimation - a comparable problem to the one we seek to solve. This work uses multiple high-quality views of an indoor scene to estimate depth in the images.

Attempts at 3D reconstruction of anatomy from 2D images have been made in recent years, often attempting to produce CT-like models. Kasten et al. (2020) uses deep learning and bilateral views of knee X-rays to reconstruct 3D models of

the imaged anatomy. This deep neural network is trained using simulated bilateral X-ray images produced from CT scans, but is intentionally restricted to just images of the human knee, to allow the network to learn the typical structure, making 3D reconstruction easier.

Shen et al. (2019) and more recently Tan et al. (2022) each present deep neural networks for producing CT-like outputs from a single X-ray image. Both of these works produce good quality CT-like 3D representations of the imaged anatomy, but require considerable quantities of training data to develop models capable of such reconstruction.

One of the earliest works to reconstruct 3D volumes from single X-ray images, Henzler et al. (2018) presents a deep learning model which reconstructs bone surfaces from mammalian X-ray images. This method utilises a multi-scale-like approach, in which the coarse output of the deep neural network is fused with the high-resolution images to produce a fine, 3D reconstruction.

3.8 Conclusion

Based on the work that has been described in this chapter, we begin to identify an area of research with the potential for significant improvements in the field of algorithmically enhanced X-ray imaging. We have identified value in physically motivated medical image enhancement, as this provides a degree of trust to medical professionals who have to make decisions regarding people's well-being based on the data we provide.

We have identified regularized optimisation methods as an avenue for solving ill-posed, inverse problems such as ours, and seen an emergence of machine learning implemented as part of this regularization pipeline.

The value of MC simulations in X-ray image enhancement has been shown, and we have seen how they can be used to produce realistic X-ray simulations suitable

for the construction of an optimiser that iteratively produces more accurate X-ray images.

Finally, we have seen how techniques from multi-grid analysis have demonstrated value in image analysis and ill-posed problem solving, where coarse resolutions allow for faster computation, with the knock-on effect of smoothness regularization implicit in their application.

Together, this leads us towards a research focus that combines these ideas to produce a physically motivated, iterative and multi-scale algorithm for efficiently resolving the material distribution within an imaged anatomy from a single scan. This optimiser is predicated on an accurate X-ray scatter simulation, but it also facilitates more accurate scatter estimates, which can be used to reduce the level of noise in diagnostic images in a clinical setting.

This is achieved through regularization via domain knowledge and CNN prediction, taking the anatomical understanding that radiologists use every day and reformulating it such that it can be applied mathematically to make an ill-posed problem solvable.

Problem Formulation and Exploratory Studies

4.1 Formulation as an Optimisation Problem

The principle of transforming the material estimation and scatter correction problem that IBEX’s software tries to solve into a more classical optimization problem was rapidly identified as a valuable area of research. The problem is ill-posed (Engl and Groetsch, 2014), but the human observer can often derive far more information about an imaged anatomy than this ill-posedness would suggest. By making use of learned, human experience, it is trivial for even someone with no medical training to recognise that an X-ray of a human anatomy usually depicts bones surrounded on all sides by a layer of soft tissue. Just as the history matching — where the current estimation is compared against previously collected, known data (Astfalk, 1996) — in the TrueView algorithm uses this knowledge to infer thickness, the human eye does too. A trained eye can go several steps further, using their understanding of human anatomy to recognise different organs from soft tissue density (Darby et al., 2011), tumors from unexpected regions of varying density, or osteoarthritis from the separation within joints (Piperno et al., 1998).

We determined that valuable research could be performed in this area, as the

concept of applying domain knowledge to constrain a formally ill-posed problem has applications across many problem areas (Xian et al., 2012; Crockett et al., 2019; Liao and Ji, 2009), provided it can be demonstrated to work on this well-defined problem.

4.2 Exploratory Research

Some initial experiments were proposed to attempt to explore the problem space and assess the ill-posed quality of the problem. These experiments were supposed to allow us to design prototypes for our final algorithm, without the more intense development overhead involved in creating code to interface directly with the existing scatter correction model. In this section, we present the ideas that we knew would become part of our final algorithm, so that we can develop and test rapid prototypes for them.

4.2.1 Simplified Scatter Model

Initially, a highly simplified model of scatter dynamics was developed, allowing the broad concept of the problem to be explored without the need for any specialised databases or detector information. It was understood at this time that the scatter model being tested - and by extension problem dynamics - were incorrect, but were intended to be a suitable enough abstraction of the real problem to test the capabilities of optimization techniques.

Based on a system originally developed for experimentation with CT by IBEX Innovations, the model that was constructed consisted of two layers of voxels, composed of aluminium and PMMA. These materials were chosen as reasonable analogues for bone and soft-tissue, respectively, (Watanabe and Constantinou, 2006), and combined at different ratios into so-called alloys. This leads to each voxel having an associated alloy value, μ , between 0 and 100 representing the percentage

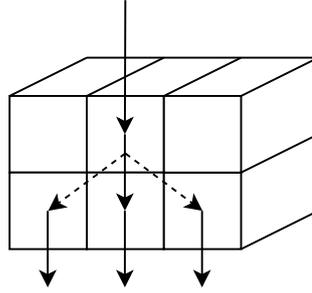


Figure 4.1: Highly simplified model of scatter dynamics, representing the ill-posed quality of the problem, where the model parameters relating to one pixel influence the intensities of the other pixels nearby.

of PMMA in a PMMA-aluminium alloy. The X-ray attenuation details of these materials were sourced from Hubbell and Seltzer (1995). This structure of voxels with associated alloy values is denoted A .

The two separate layers were designed to replicate the thickness aspect of the object, as in principle the number of voxels stacked upon each other could be increased or decreased in different regions to represent non-homogeneous thickness.

In this idealised system, simulated photons of various energies are projected vertically down through each pixel, scattering into neighbouring pixels in the layer below (shown in Figure 4.1). The scatter and absorption dynamics were determined by two values, the Linear attenuation coefficient (LAC) and Scatter window fraction (SWF) of each voxel. This attenuation coefficient and scatter fraction was found for the nearest integer alloy percentage in a precomputed lookup table.

The result of this simulation process was a matrix image, available as either the counts in each of the energy bins, or the integral across them, as an intensity value for each pixel. As our research was focused on non-CT imaging, only the intensity integral was considered, as this is the output most commonly associated with the imaging modality.

An example of this simulated imaging, and the input from which it would be derived, are shown in Figure 4.2.

This gave us a simulated image $L(A)$, as a function of the 3D array of inputs. We

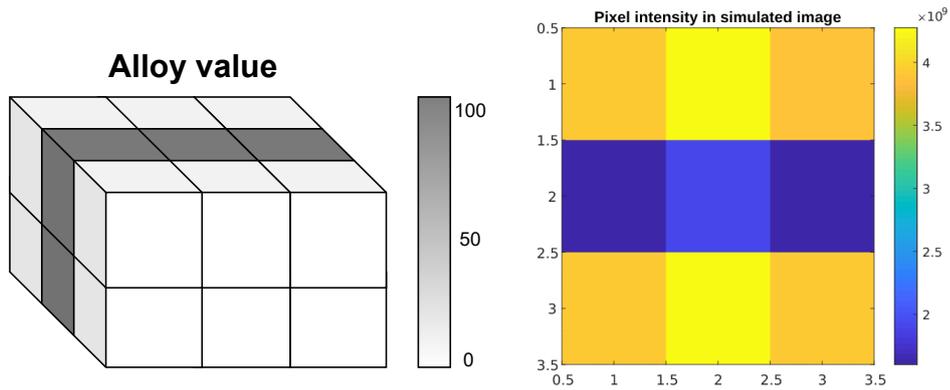


Figure 4.2: Densities of voxels in a $3 \times 3 \times 2$ setup, and the simulated image generated by that setup.

can then formulate a minimization problem, seeking to find the 3D voxel structure A corresponding to the point

$$\min |I - L(A)| \quad (4.1)$$

where I is our input image. In these experiments, where the scatter simulation process is simplified beyond the point of realism, it is unreasonable to expect the system to converge to any sort of real X-ray image. Thus, we simplify the models further to reduce the time required to produce results, and make the relationships between model and output more clear. The experiments here concern 2D, typically either 3×3 or 9×9 pixels, depending on the scale of model required to demonstrate the effect in question. These small images were chosen to minimise the processing time, while still getting meaningful results. Thus the input images were also the results of the simulation process, for a pre-set distribution of alloys A_{True} .

This reduced the problem's complexity somewhat, as it was known that a solution of Eq.4.1 was achievable by the simulator, and no bias would be involved. This, in effect, represents the problem with access to a perfect X-ray simulator.

With the problem reformulated as an optimization, a selection of algorithms could be trialled to attempt to recover the known ground truth input data from an ill-posed output.

4.2.2 Gradient Descent

A finite-difference gradient descent algorithm which takes some 1D function which it seeks to minimise the error, $y = f(x)$.

This algorithm can be applied to multivariate functions, by taking derivatives in each of the possible dimensions, and combining as vectors. This can be done by estimating the gradient of each dimension of the multivariate function separately, following the same process as described above, but estimating a partial derivative with respect to one of our variables:

$$\frac{\partial}{\partial x}y(x, z) = \frac{y_0^* - y_0}{x_0^* - x_0}, \quad \frac{\partial}{\partial z}y(x, z) = \frac{y_0^* - y_0}{z_0^* - z_0}$$

Updates to descend the multivariate function can then be applied either sequentially (estimating a different partial derivative after each) or simultaneously (by combining updates to each variable into a single change).

Choice of a reasonable damping factor C is essential to efficient convergence. Too small, and each step will only traverse a small part of the distance to the nearest minimum, and many steps will be required for any sort of convergence. Too large, and the step could go past the nearest minimum, leading to a case where the minimum cannot be reached. Furthermore, it worth emphasising that this method is only suitable for finding *local* minima, and its search for a *global* minimum will be easily hampered by a local minimum close to the start point.

4.2.3 Stochastic Gradient Descent

Because of the need to calculate many gradients for each iterative step (one for each dimension of the function being minimised), gradient descent is an inherently costly algorithm, and becomes prohibitively expensive in high-dimensional problems, such as the one we are attempting to solve. The solution to this is SGD (Shalev-Shwartz and Ben-David, 2014).

SGD follows the same basic structure as gradient descent, but where gradient descent take the derivatives across all dimensions (item 4), SGD instead only takes the derivative with respect to a single dimension, and thus only moves in that direction. This leads to a higher number of total iterations to converge in multivariate functions compared to gradient descent, but the reduced cost per-step means that it can be a more efficient algorithm overall.

The first experiments carried out on the simplified scatter model were done using this SGD technique, as it was understood from the beginning that even if batched gradient descent could be cost-efficient for the simplified case, there was no feasible, foreseeable outcome where batched gradient descent could be appropriate for the real problem we ultimately intended to solve.

4.2.4 Mini-batch Gradient Descent

Considering the two techniques described above in 4.2.2 and 4.2.3, a natural hybrid of the two can be imagined, where gradients are taken for some number of the possible parameters B , such that $N > B > 1$, where N is the total number of variable parameters.

This naturally provides a tune-able trade-off between the accuracy of a batched gradient descent, and the efficiency of a stochastic gradient descent. This is achieved by defining some number of parameters which are to be investigated in a given mini-batch, then defining some method the algorithm should use to select this subset of parameters. In the simplest case, this mini-batch size would be constant, and the parameters to vary would be selected at random each iteration, though naturally more elegant choices can be made for both of these if deemed necessary (Khairat et al., 2017; Dekel et al., 2012).

4.2.5 Simulated Annealing

Simulated annealing (SA) is an optimization algorithm inspired by thermodynamics of cooling metals and crystal structures, from condensed matter physics. In this algorithm, parameters of a function to be optimised are iteratively altered by random perturbations, which can lead to a result closer or further from the desired global minimum. (Kirkpatrick et al., 1983)

In the case of deterioration, the updates are accepted with some probability, proportional to a decreasing, iteration/time-dependent temperature function. As more iterations pass, and the temperature gradually cools, fewer and fewer detrimental updates are accepted, until a point where only improvements can be made.

One key benefit to this algorithm for us is that, because all updates are random, small perturbations, there is no need to calculate the derivative on the function being optimised. For our experiments with gradient descent, the lack of a derivative requires a numerical gradient be calculated on each iteration and is assumed to be an accurate estimate of the derivative. SA requires no such assumptions.

The ability of the algorithm to occasionally jump upwards, away from a more optimal solution, reduces the likelihood of the optimization becoming trapped in a local minimum. This allows the algorithm to find the global minimum, even in a problem-space with many local minima that would prevent convergence to the global minimum of a gradient descent algorithm. The trade-off to this, is that SA is fundamentally probabilistic, and while it is able to leave local minima and find a global minimum, it is not guaranteed to do so. As such, even with a reasonably accurate initial guess, SA could take longer to converge on an accurate solution than other algorithms, as it will attempt to traverse a wider area of the problem space, to determine whether the current minimum is the global one.

This is a very valuable property for a optimization algorithm, but given that our problem case allows us to make a reasonable guess for the final state before any iterations have been carried out, SA may not be the best-suited method. The

decision was nonetheless made to evaluate it on this simple problem, for the sake of comparison to other techniques.

4.2.6 Test Cases

A series of test-cases were proposed for this simplified model, to assess the response of each algorithm to different restrictions and initial inputs. These test cases are as follows:

1. Begin with the ground truth as input, to verify that the algorithm as written will halt and not diverge when the correct result has been reached.
2. Constrained alterations, where a single parameter is altered by the smallest possible amount, and changes over the course of the algorithm are restricted to just this parameter. This aims to verify that the algorithm descends as expected.
3. Incorrectly constrained alterations. This takes a comparable form to the previous case, except the one unrestricted parameter is *not* the same one that has been altered initially. This verifies that in a 1D case where the correct solution cannot be reached, other parameters do not diverge.
4. Unconstrained alterations, where a single pixel is altered by the smallest possible amount as prior, and the algorithm is free to make updates to any parameter.
5. Added noise. Any pixel can be altered by the algorithm, and the initial state is the ground truth plus Gaussian noise. A series of tests were performed with this case, with different levels of noise.

These test cases were applied to the $9 \times 9 \times 2$ model with up to 1000 iterations of batched gradient descent, as shown in Figures 4.3, 4.4 and 4.5. No figure is shown for the first test case, as the behaviour was exactly as anticipated, in that the

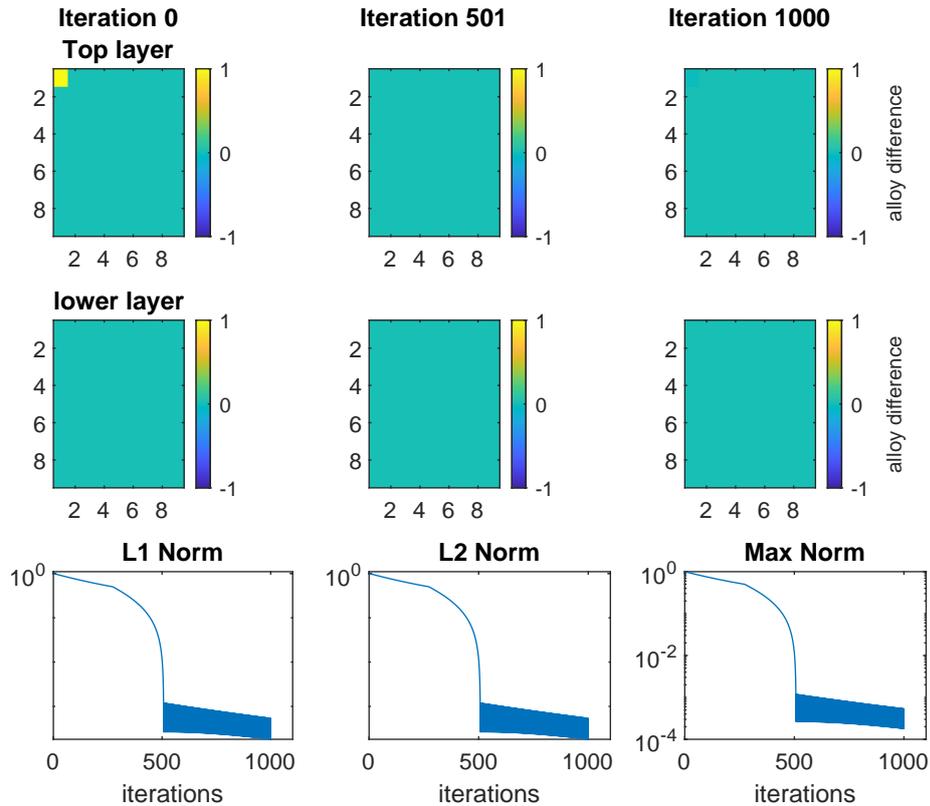


Figure 4.3: Results of stochastic gradient descent on test case 2.

algorithm immediately halted with no iterations of descent, as it was recognised that a minimum had already been reached, and the objective function $|I - L(A)| = 0$ had already been solved.

The other test cases however yielded results different from the expected performance. For the second test case (Figure 4.3), the one altered pixel does converge as expected, but because of the step size, it is unable to reach the exact solution. This leads to continuous oscillation around the true solution, approaching but not reaching it. Visual inspection of the values of the voxels shown in the images of differences at iteration 1000 shows us that the convergence has led in the correct direction when heavily constrained as it is in this test case.

Test case two, however, showed less desirable behaviour. Because the updates the algorithm is able to make are constrained to a single pixel, and this pixel is *not* the one that is initially altered, the algorithm was unable to correct the alteration.

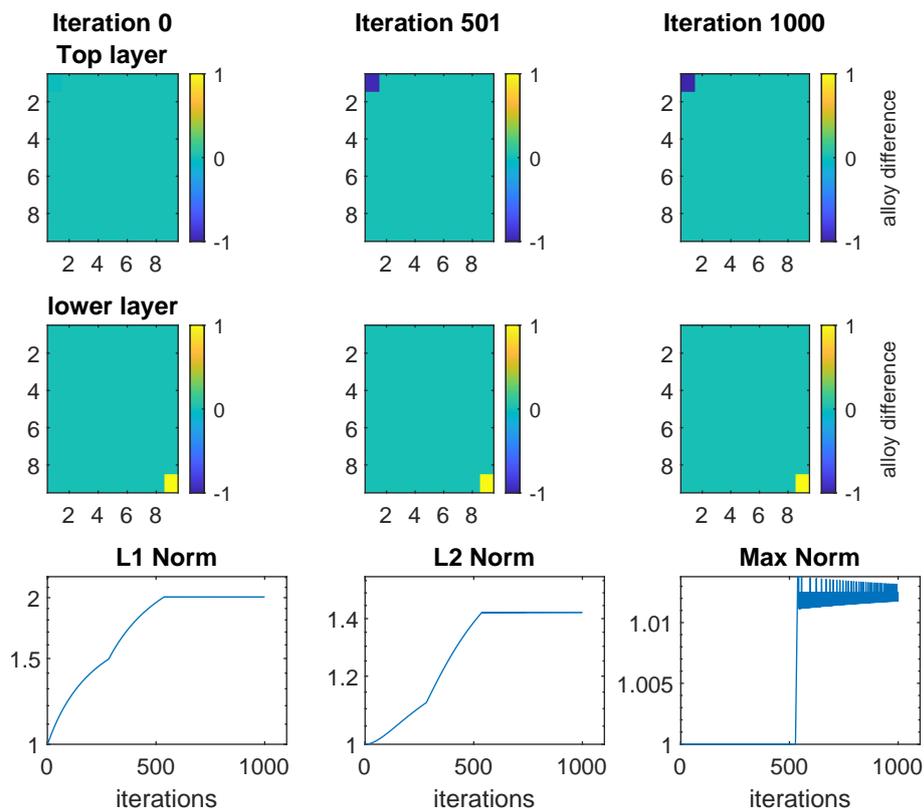


Figure 4.4: Results of stochastic gradient descent on test case 3.

The test aimed to investigate how the gradient descent would respond to this, given the ill-posedness of the problem. As Figure 4.4 shows, the algorithm attempted to correct the initial difference in the only way it could, but adjusting the available pixel to compensate. This led to a situation where two pixels were different from the ground truth by a magnitude of one, at which point the solution began to oscillate, unable to make any further improvements.

Test case 4 showed a small amount of progress, but was generally limited by the number of iterations and the relaxation coefficient applied. Here we see the L1 norm increase, but the L2 norm varies around the 1 value, and the max makes some progress towards zero. Clearly, conclusive results about this test case cannot be made from so few iterations.

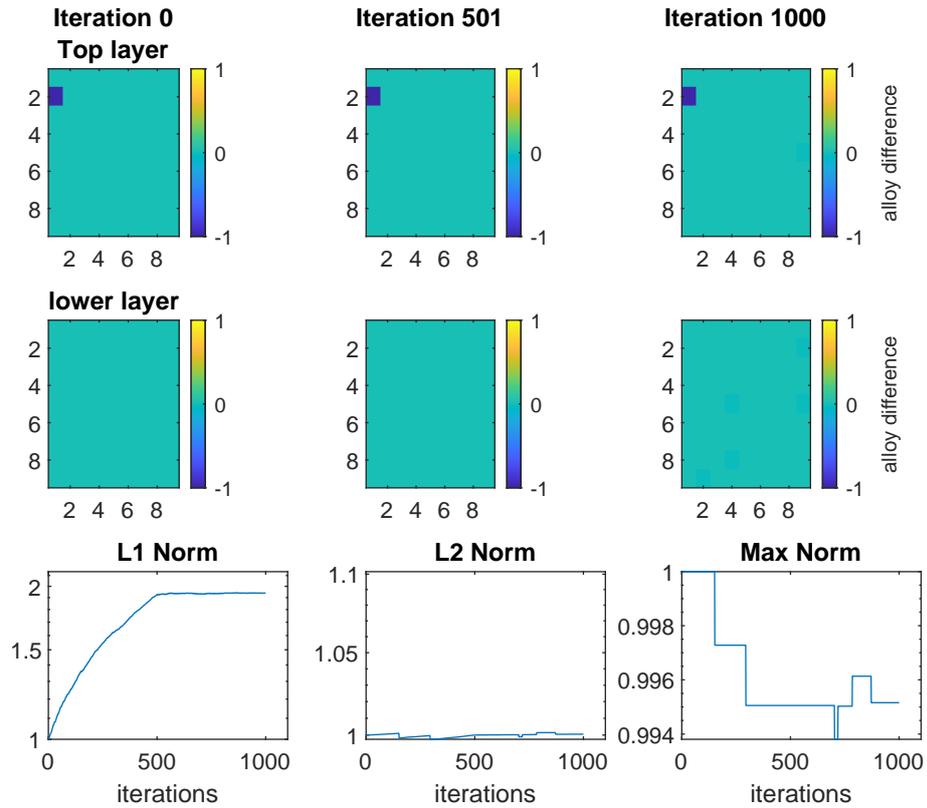


Figure 4.5: Results of stochastic gradient descent on test case 4.

4.3 Regularization in the Exploratory Setting

From these results, it is clear that the gradient descent, and other optimization algorithms are functioning as expected, but are insufficient to solve the problem at hand. This is not an unexpected result, as it makes clear that the problem is as ill-posed as previously expected, and further work is required to reformulate it into something that is classically solvable.

Given the research conducted into optimization, and the possibilities it could unlock with regard to parallelization, the decision was taken to investigate how regularization could be used to modify the cost-function of this simple test case, to make the problem solvable .

This means that a technique had to be investigated, to appropriately inject domain-

knowledge into this system, such that *the same* optimization algorithms could be applied to a new function, which would lead to the optimal parameters for the original problem. A function had to be constructed which would accurately represent the behaviour of the original function, but distorted in such a way that it would be sufficiently convex to be reliably solvable .

4.3.1 Simple Segmentation Regularizer

The most immediately available source of this domain knowledge that is available for the non-simplified problem of correcting an X-ray image is the segmented image, as discussed in 2.3. Segmentation provides reliable information about the values of the alloy of voxels for specific regions of the image, which begins to constrain the problem considerably, by locking-in one of the two parameters for large regions of the image. This makes the search-space for those areas 1D instead of 2D, and therefore much easier to optimise.

However, it is crucial to note that this segmentation is imperfect, and while it is generally accurate, it is probable that in a real-world, clinical setting, it will mis-estimate the occasional pixel.

As such, accepting the output of the segmentation as truth and permanently fixing the alloy parameters according to its estimation is likely to lead to sub-optimal models. We therefore want to construct a function that respects the output of the segmentation as advice, rather than strict orders.

It is trivial to add a segmentation to this simple problem, as we are only interested in its output, and not the means by which it is reached. While the segmentation would be provided by a CNN in the real problem, for now it can be manually constructed to accurately reflect the alloy distribution. We simply produce a binary image, with the same x and y dimensions as the model we are attempting to resolve. We arbitrarily pick a value for "bone is present" (selected as 1), and let the other value take the label for "no bone is present" (0 in this case).

Considering the potential knowledge that can be leveraged from the segmentation, we recognised that inferences can be made about the alloy of any voxel. First, for pixels classified as "no bone is present", we expect that the alloy of corresponding voxels should be exactly equal to 100, representing 100% soft-tissue. Second, for pixels classified as "bone is present", we have no estimate for the alloy value, *except* that it is not equal to 100, as that would imply there is not a bone. We cannot however make any estimate for what value the voxels *should* take, as the segmentation cannot give us an estimate for *how much* bone is present.

Quantifying this, we introduce an additional term to the function we are minimising, equal to the current alloy of that voxel, for all voxels that have been segmented as soft tissue (and would therefore be expected to have an alloy of 0). This contribution is multiplied by a set weighing factor, and added to the current cost function. This means that the regularizer introduces a penalty for all voxels that have been classified by the segmentation as soft-tissue, which is proportional to the difference between the current alloy and expected alloy for that voxel.

First, we re-run the second test case, but with a segmentation regularizer term. This has a known, predictable outcome and functioned as expected for the unregularized case, and serves as the benchmark for the regularizers. The result of this can be seen in Figure 4.6. We see that, as expected, the BGD algorithm converges towards the know solution, and then stops when the exact solution is reached. This actually outperforms the unregularized case somewhat, as the regularizer helps the algorithm settle in the exact global minimum, instead of oscillating around it as seen in Figure 4.3.

Figure 4.7 shows test case three, as described previously. Here, a crucial difference can be seen in contrast to the unregularized case shown in Figure 4.4. Crucially, the parameter that is permitted to vary now does not alter to its most extreme allowed value in an effort to compensate for a distant error. This is because the regularizer encourages behaviour that aligns with our expectations. Instead, the alterable parameter varies until it is incurring a cost that balances the difference

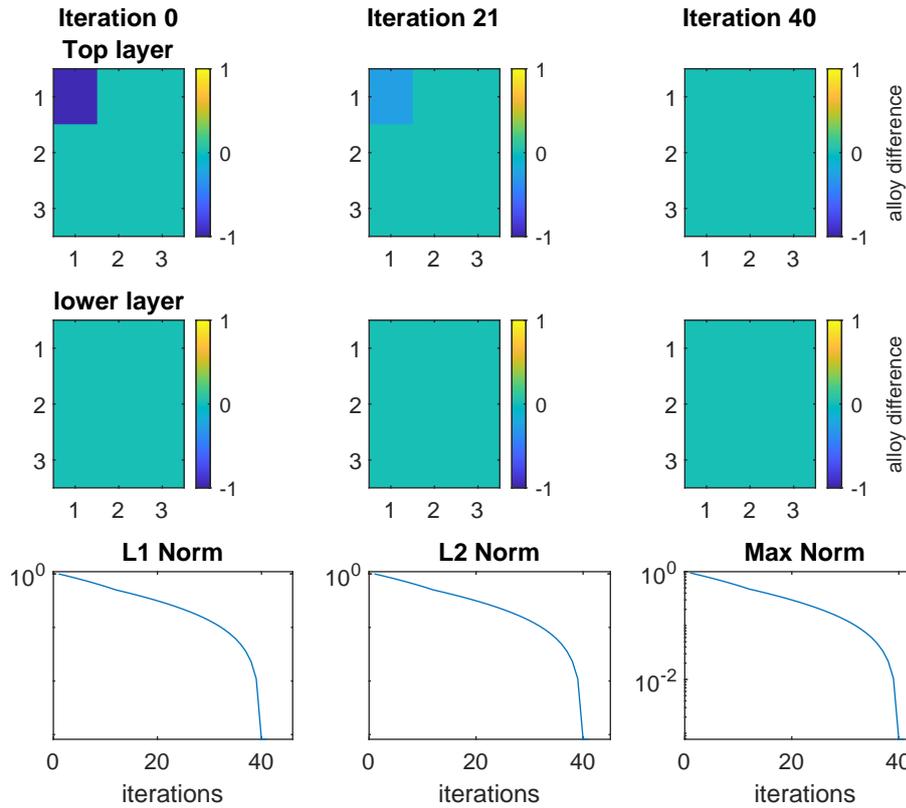


Figure 4.6: Results of stochastic gradient descent on test case 2, with an added regularization term relating to the segmentation.

between the simulated and true images, at which point it stops changing and settles. The value at which it settles is determined by the weighting of the regularizer, and can thus be increased/decreased accordingly.

For the setup shown in Figure 4.7, this change provides no additional value. We have, in effect, introduced an additional parameter set before-hand which determines the final value of a pixel. Considering this more generally though, a factor has been added that can and does compensate for erroneous behaviour, leading to a more accurate solution overall.

Figure 4.8 shows test case 4 with the new regularizer. This too shows significant improvement on the unregularized case, as the erroneous value converges towards the true. Unfortunately this is counteracted by the slight deterioration of adjacent values, for the same compensation reason as discussed for test case 3. The

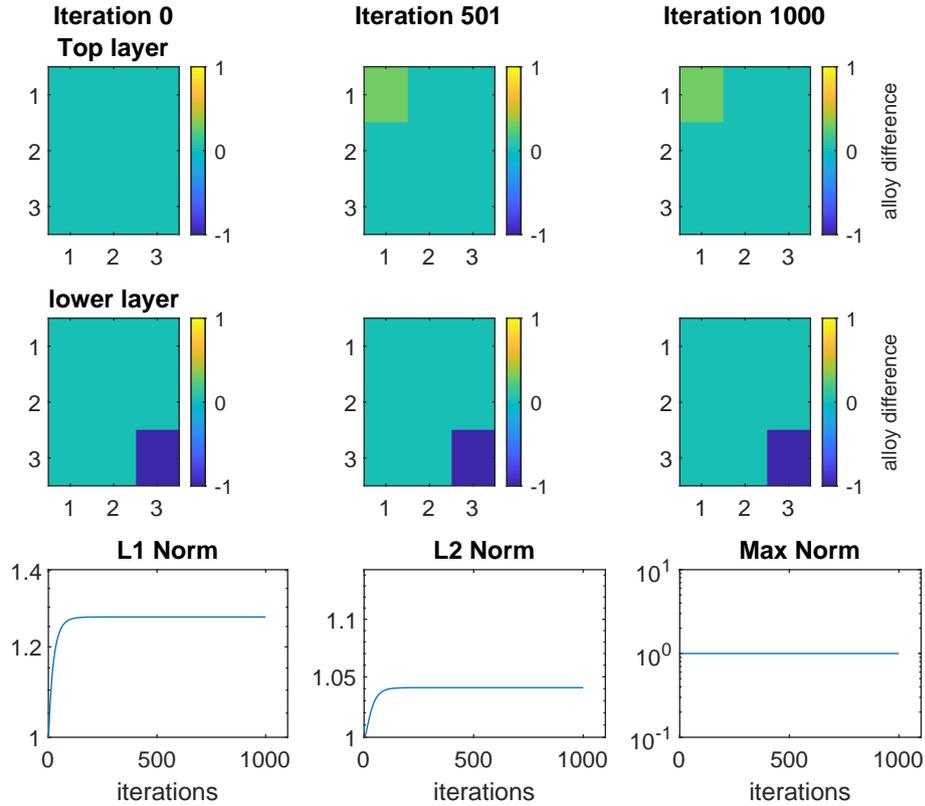


Figure 4.7: Results of stochastic gradient descent on test case 3, with an added regularization term relating to the segmentation.

important factor to note here is that, because of the values selected for the ground truth, a small offset of ~ 0.35 like the one seen in various voxels is functionally equivalent to the ground truth value. This is because the values selected as ground truth in this test case are the limits of the acceptable range (1 and 100), and values reached by the estimator that are outside of this acceptable range are truncated to be within it for the purposes of the simulation. As such the voxels with slight differences from zero in the final iterations are incorrect because they are slightly outside of the valid range, and are treated as the nearest valid value for simulation purposes. This allows the algorithm to converge on iteration 997, at the point where all voxels either are, or are treated as, their ground-truth value.

In mathematical terms, we now construct a cost function $J(A)$ of which the image residuals in 4.1 is just a part. We introduce a term based on segmentation, giving

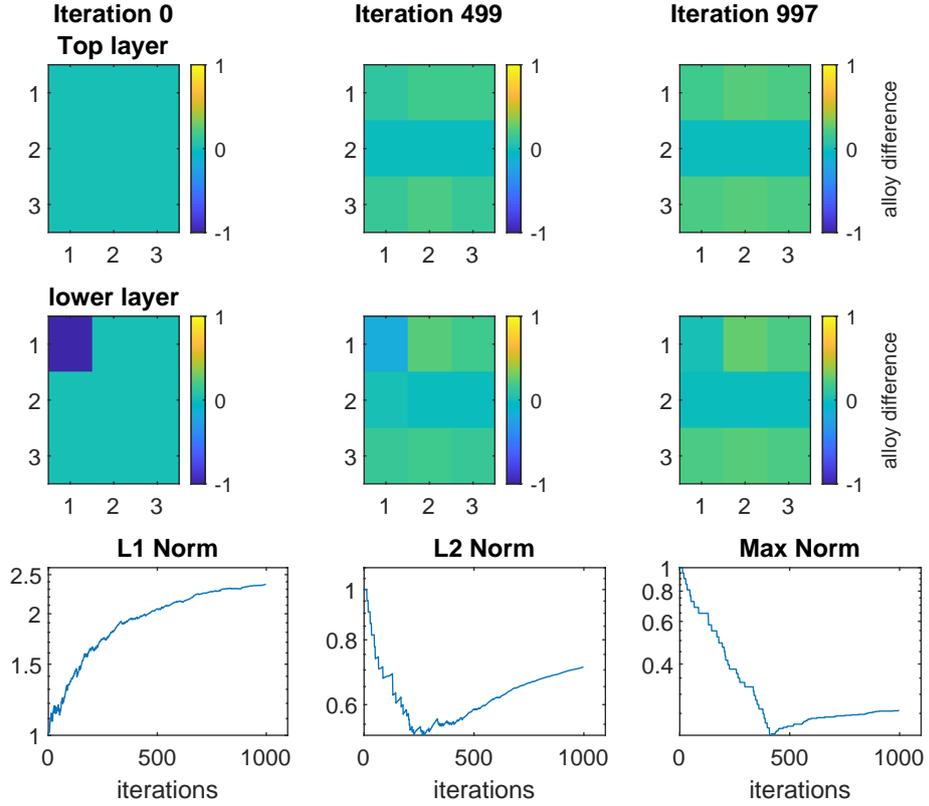


Figure 4.8: Results of stochastic gradient descent on test case 4, with an added regularization term relating to the segmentation

us a new function to minimise:

$$J(A) = \alpha_a |I - L(A)| + \alpha_b ((1 - Seg) \odot A) - (Seg \odot [A < 100]) \quad (4.2)$$

where \odot denotes the Hadamard product (i.e. an element-wise multiplication), α_a, α_b represent relative weighting factors for each component that can be manually tuned, Seg represents the previously described 2D binary segmentation classifier, repeated through the z dimension, and $[A < 100]$ represents a binary mask matrix with the same dimensions as A , with values set such that:

$$[A < 100]_{x,y,z} = \begin{cases} 1 & \text{where } A_{x,y,z} < 100 \\ 0 & \text{else} \end{cases}$$

4.3.2 Simple Smoothness Regularizer

The second regularization term that was devised was linked to the smoothness of the final model. It had been recognised that one highly spurious estimation could lead to a cascade effect, where other parameters would be highly under/over-estimated to compensate for one early mistake. It was identified that constraints on the smoothness of the solution would limit how far an individual change could modify a parameter, and reduce the likelihood of this negative feedback loop occurring. Further, it was recognised that enforcing some expectation of the smoothness was a reasonable concept, as even in the final algorithm, where we attempt to solve over real anatomies instead of manually constructed blocks, the organic nature of the imaged objects implies some degree of smoothness.

To determine a quantity for the current smoothness of the array of alloy values A , a partial second-derivative, three-tap filter was convolved with the array in x and y . This takes the form:

$$\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

This filter can be applied at any point of interest in the array along each of the three dimensions (with replication padding at the edges). This yields three variables dx, dy, dz which describe the smoothness at the point of interest along each dimension.

With a bit of thought, we can recognise that this is not a fully accurate representation of the assumptions we can make from our domain knowledge. Specifically, we expect sharp changes in the value of A along the boundaries of bones, where the alloy of the organic material changed sharply from a high percentage of soft-tissue to a low-percentage of soft-tissue. Further, we know exactly where we should expect to observe these boundaries within the array, thanks to the segmentation described in 4.3.1. An edge-detection filter is applied to the segmentation, which

serves as a mask, indicating where this smoothness regularizer should and should not be applied. The edge-detection is a binary array with the same dimensions as Seg , where:

$$Edge_{Seg} = \begin{cases} 1 & \text{where an edge is present} \\ 0 & \text{else} \end{cases} \quad (4.3)$$

Then we simply introduce $(\mathbf{1} - Edge_{Seg})$ as a multiplicative term with the smoothness calculation, such that the result is always zero along edges in the segmentation.

Finally, we introduce weighting factors for each of the three dimensions, so that in future the smoothness along each of the three dimensions can be varied relative to each other if desired. This is then added to equation 4.2, to give:

$$J(A) = \dots + \alpha_{cx}dx(\mathbf{1} - Edge_{Seg}) + \alpha_{cy}dy(\mathbf{1} - Edge_{Seg}) + \alpha_{cz}dz \quad (4.4)$$

Note that the edge detection term is not present in the term related to the z dimension, as the edge detection is derived from the segmentation, which only carries information about the x and y dimensions.

Finally, some results for test cases two, three and four are included to compare this regularizer against the segmentation-based one described in the previous section.

Figure 4.9 reconfirms our assumption that the regularizer does not actively inhibit the convergence of a parameter when the error and permissible updates are confined to a single dimension of the problem space, with the final result settling in the optimal value.

In much the same way as with the segmentation regularization shown in Figure 4.7, Figure 4.10 gives us a tuneable hyper-parameter that can control the divergence that occurs in test case three, limiting how much non-smoothness is acceptable, relative to a given residual between real and simulated image.

Figure 4.11 shows how the regularizer can smooth-out a single-voxel error to a more homogeneous final estimate. Figure 4.12 shows this same setup, but where the randomly-selected pixel is from the central horizontal band, which has been defined

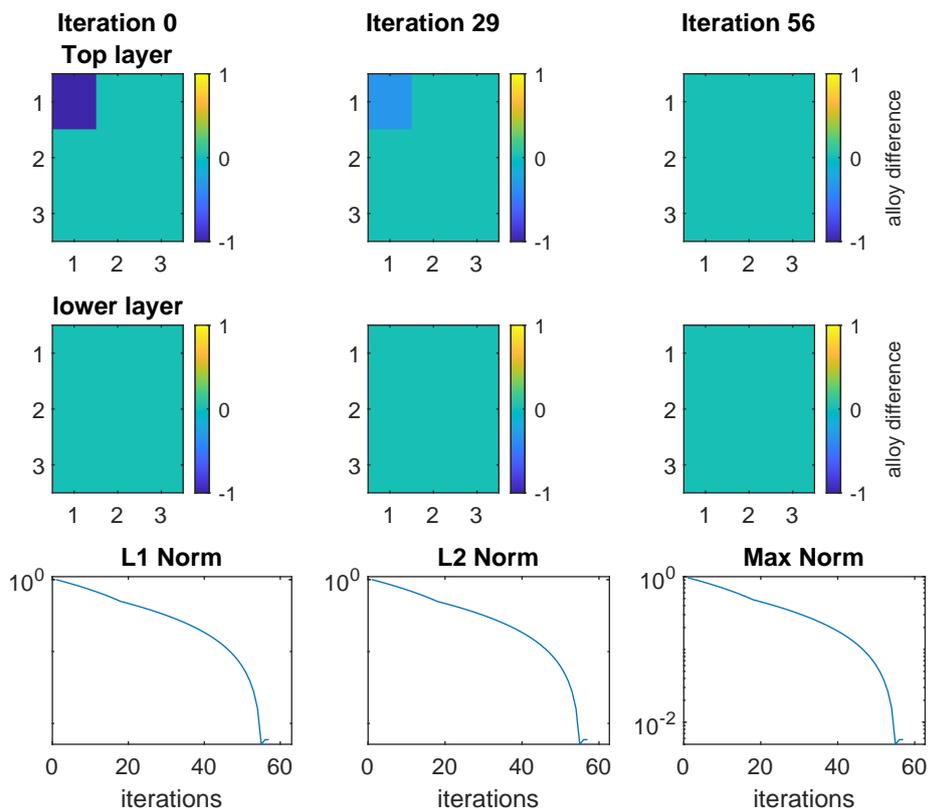


Figure 4.9: Results of stochastic gradient descent on test case 2, with an added regularization term relating to the smoothness.

as the higher-alloy, bone voxels in the segmentation. Part of the reason these two cases show a difference is because of the restrictions on the smoothness regularizer in a 3×3 voxel case. In a 3×3 setup when a central band is segmented as different to the rest, all voxels will be classified as edge along the perpendicular dimension, meaning the smoothness can only apply along the remaining two dimensions. This is illustrated further in Figure 4.13, where we present the simulated segmentation for a 9×9 model, as well as the voxels within that model where smoothness along the vertical axis will be ignored, due to the proximity of a known edge.

4.3.3 Simple Regularizers Combined

This section concludes with some demonstrations of the effectiveness of the two regularizers combined, on the test cases laid out previously, as well as some others

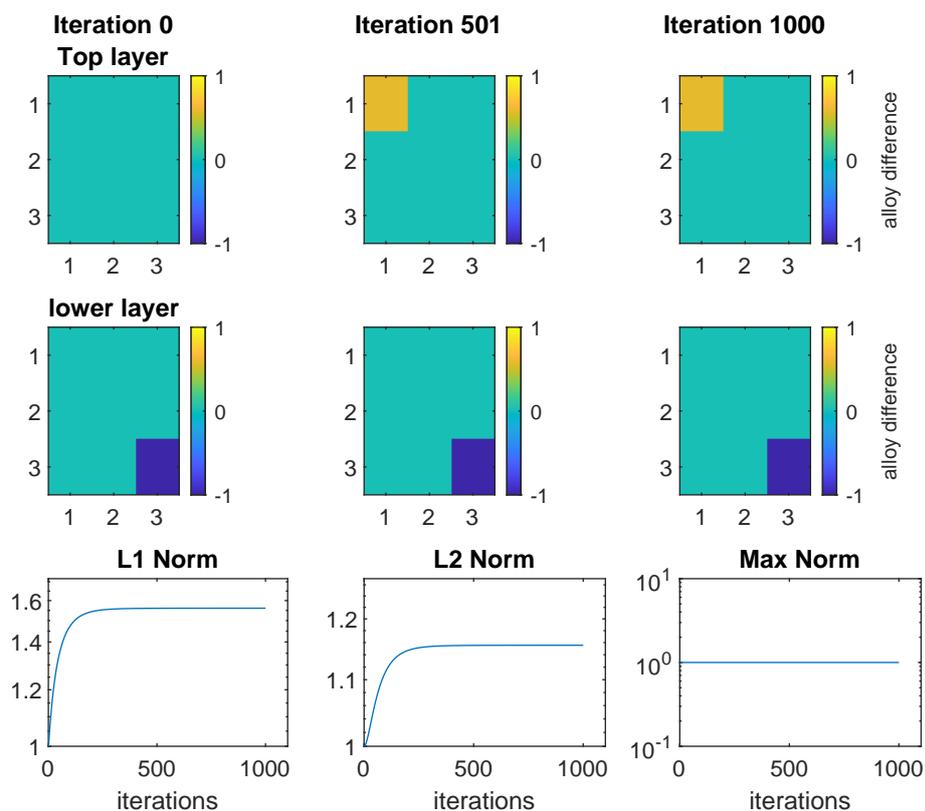


Figure 4.10: Results of stochastic gradient descent on test case 3, with an added regularization term relating to the smoothness.

of interest.

Figures 4.14, 4.15 and 4.16 lay out the results of test cases 2, 3 and 4. As expected based on the results of subsections 4.3.1 and 4.3.2, these converge without major issue to an estimate that is closer than the initial state.

Figure 4.17 shows the response of the algorithm to a previously un-investigated test case, where all voxels have been altered by the same constant value. This is of interest as it represents a more accurate representation of the initial condition that the true algorithm will encounter. It is highly unlikely that the initial state provided to the true gradient descent pipeline will be equal to the ground truth for all pixels except one. More commonly, the initial state will be set to some constant, which could be near to but not equal to the true value. This is not to de-value the other test cases, as these accurately represent the state the true pipeline will

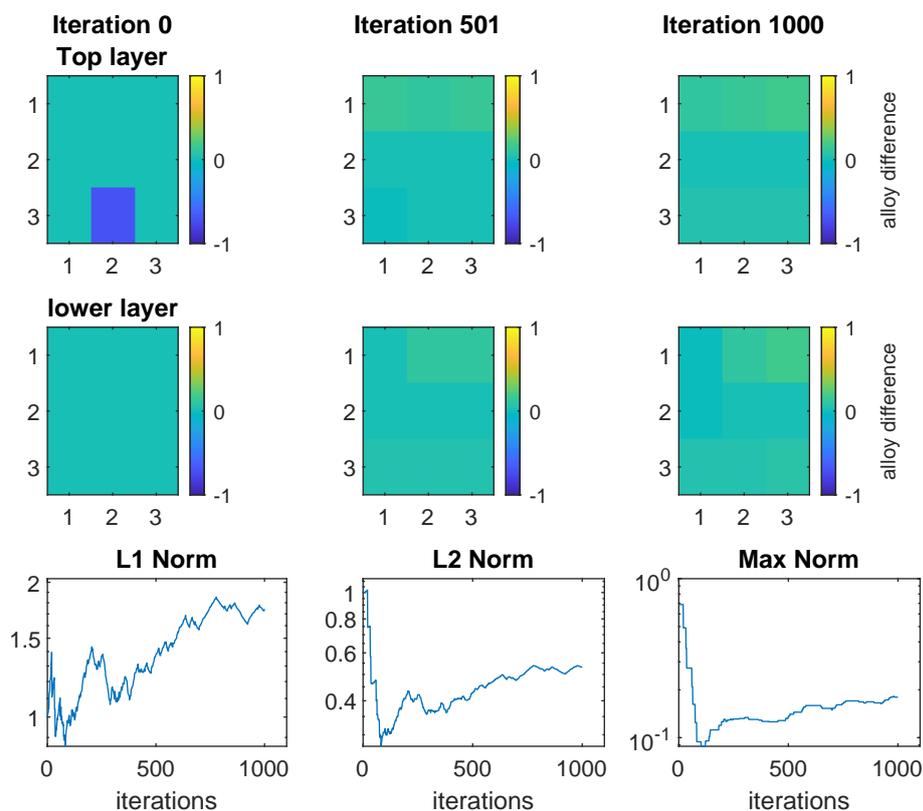


Figure 4.11: Results of stochastic gradient descent on test case 4, with an added regularization term relating to the smoothness.

encounter after some number of iterations, where individual voxels differ from their surroundings. Nevertheless, the algorithm shows capability in converging towards a more accurate estimation in this setup.

Figure 4.18 shows a case where all voxels have been altered by different values, representing 1% Gaussian noise. This again represents a scenario that the true algorithm is likely to encounter, where successive iterations have led to a situation where none of the voxels are exactly correct, but differ from the ground truth by varying degrees. Though the max norm increases in this case, the L_2 norm shows that the estimation generally converges towards something more accurate, even if the values of individual voxels have travelled in the wrong direction to compensate for each other.

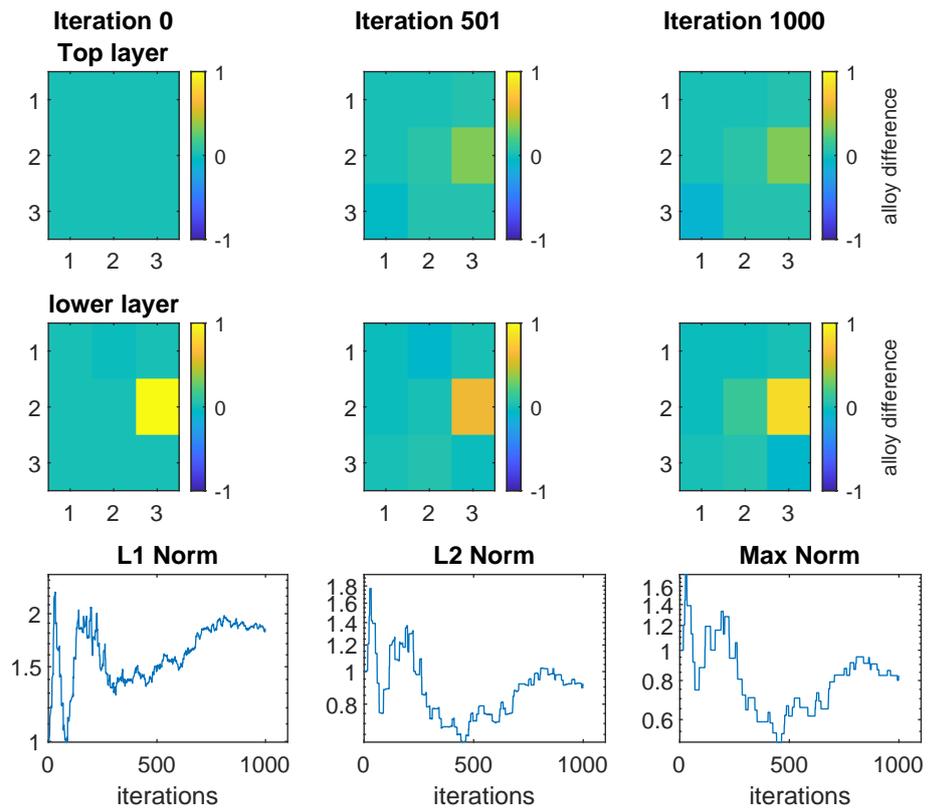


Figure 4.12: Results of stochastic gradient descent on test case 4, with an added regularization term relating to the smoothness. This figure shows the effect for a central-band, bone pixel instead of an outer-band, soft-tissue pixel. The visible difference is due to the restriction of smoothness along edges.

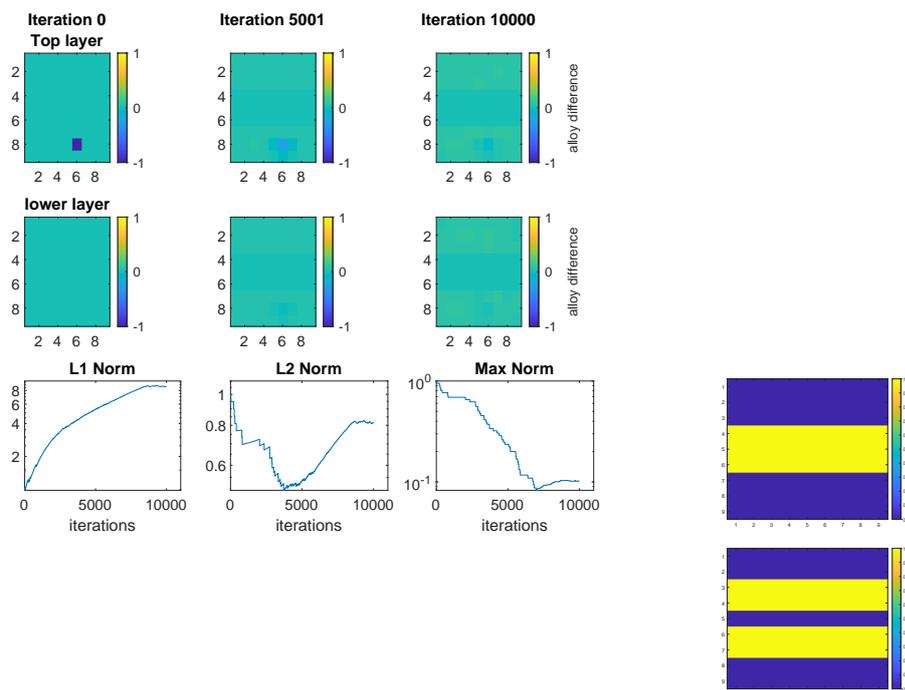


Figure 4.13: Smoothness regularizer applied to test-case 4, on a 9x9x2 voxel grid. The graphs on the right show the segmentation for this grid setup (above), and how the edge-detection will respond along one of the three axes (below).

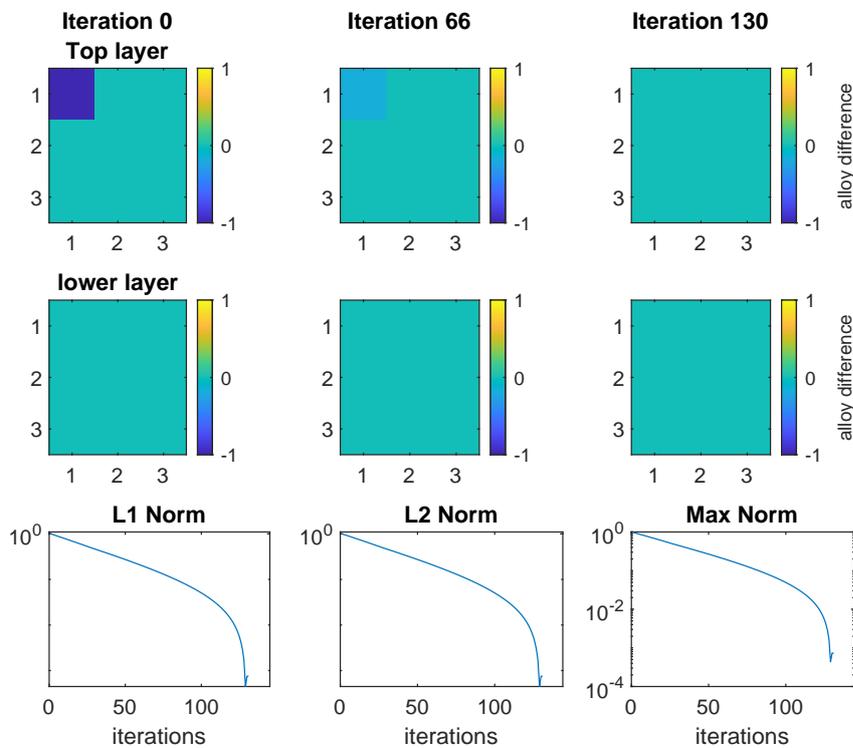


Figure 4.14: Smoothness and segmentation regularizers applied to test case 2.

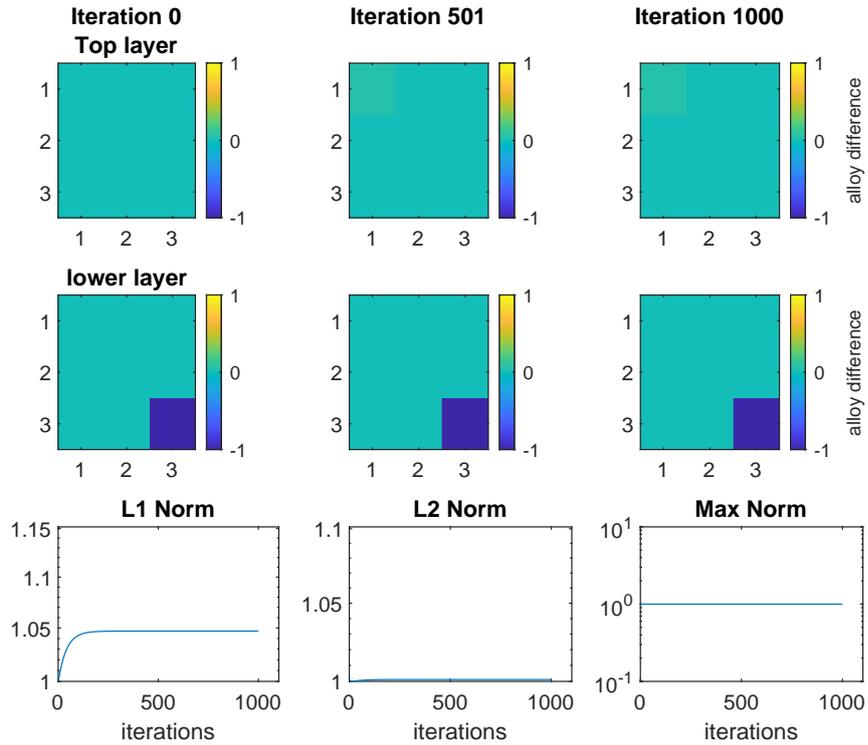


Figure 4.15: Smoothness and segmentation regularizers applied to test case 3.

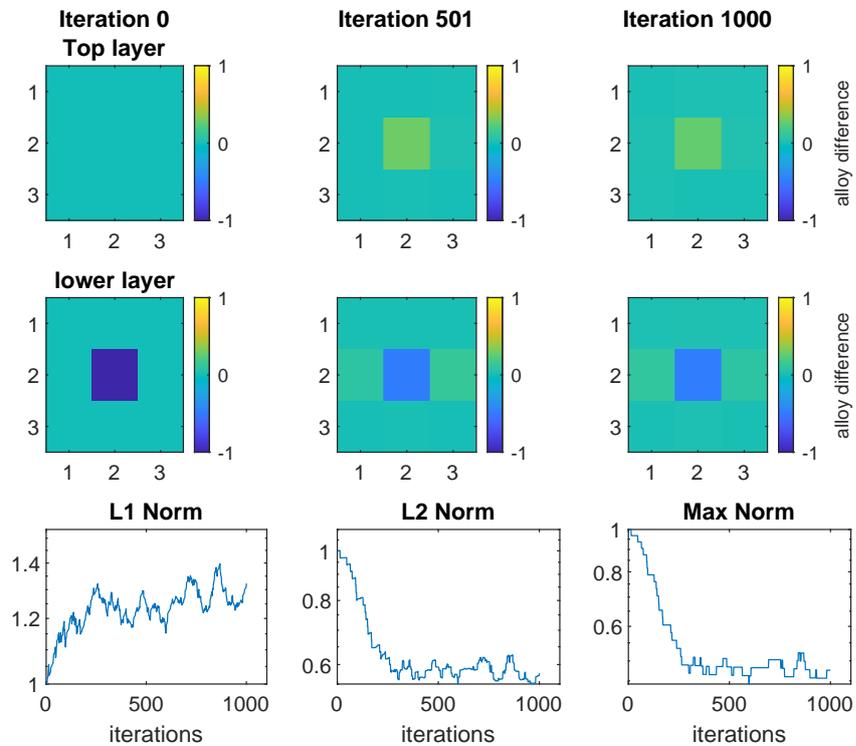


Figure 4.16: Smoothness and segmentation regularizers applied to test case 4.

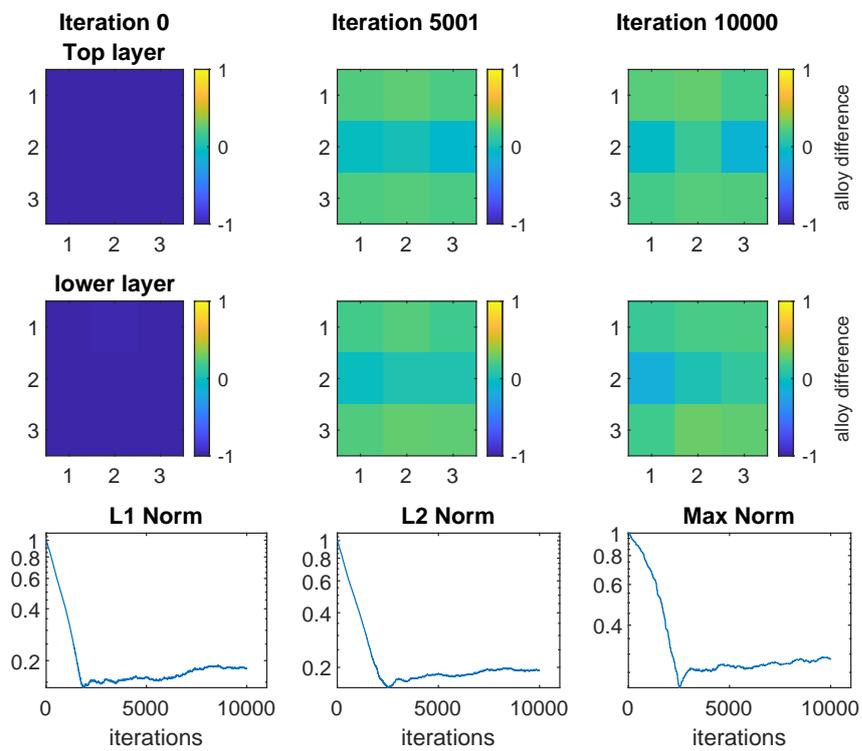


Figure 4.17: Smoothness and segmentation regularizers applied to a new test case, where all voxels have been alter by a constant value.

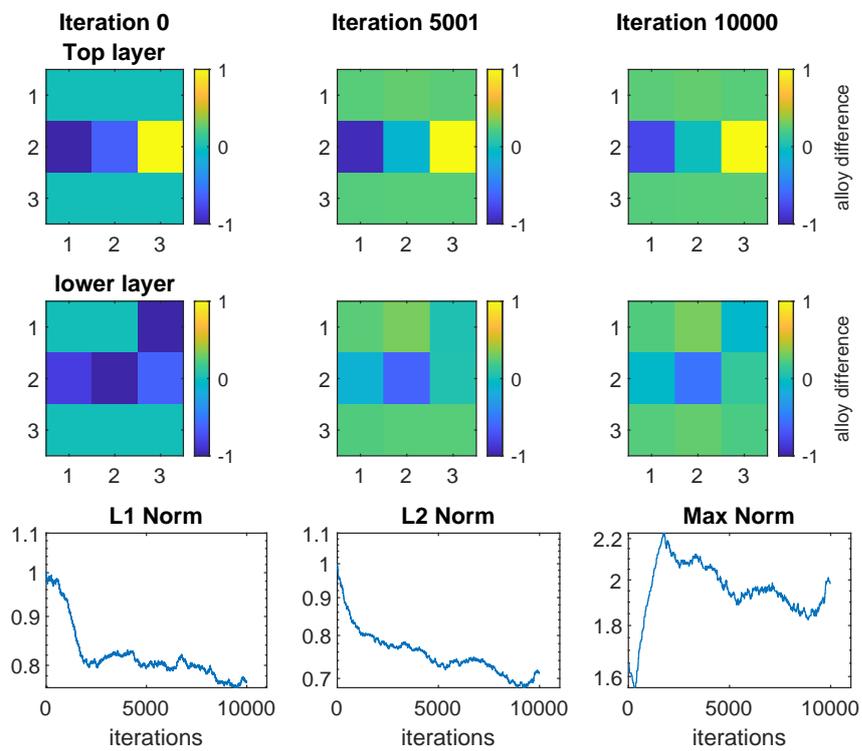


Figure 4.18: Smoothness and segmentation regularizers applied to a new test case, where randomly generated, 1% noise has been added to each voxel.

Regularized Descent

Following the development of the prototype algorithm described in Chapter 4, in this chapter we present the work involved in adapting these ideas into a functioning material estimation algorithm. This encompasses the physics-based scatter correction algorithm that we make use of, the form of the regularizers (both existing and new), and the descent algorithm we choose to find the optimal state.

5.1 Simulation via TrueView

The research carried out up to this point has all helped to show that the principles of regularized gradient descent could be the key to unlocking the problem at hand, allowing us to make accurate estimations of the composition of anatomies from a single, mono-energetic X-ray image. However, without access to an accurate simulation of the physics involved, we are confined to studying the theory of the problem, as opposed to attempting to solve it. The Trueview software, provided by IBEX Innovations, is the more advanced simulator that can allow us to progress into more accurate simulation.

Based on results of the particle simulator Geant4 (Agostinelli et al., 2003), Trueview can create an approximation of the X-ray image that would be produced on a given detector for an anatomy provided to it. Similarly to the model described in 4.2.1,

Trueview uses voxels, each with a given alloy value representing the percentage of bone in a bone–soft-tissue mixture to describe the density of a voxel. Unlike the previous model though, it does not use stratified layers of voxels stacked upon each other, but rather considers the object as being made of a single layer of voxels, which are in-homogeneous in thickness. This creates a setup where each pixel in the down-sampled image is assigned a single voxel in the model, and each voxel is defined by just two parameters: thickness and alloy, defined as t and μ respectively. Knowledge of this system was the guiding force behind the choice to begin experimentation with a two-voxel layer system, as it represents the problem in a similar matter, i.e. two parameters per pixel.

The Trueview algorithm was provided in the form of development source-code used by the research team at IBEX Innovations to develop and enhance the capabilities of the tool. This algorithm can produce a simulated X-ray image, divided into scatter and direct-beam components, from a voxel model (referred to from now on as the *spatial model*) and databases of Geant4 simulations, to accurately describe the spread of X-ray photons through materials, and what response that would correspond to on the X-ray detector in question. As with the simplified problem, Geant4 produces simulated scatter kernels for X-ray beams passing through a range of materials. Instead of PMMA and aluminium, the physics of X-ray attenuation in human anatomy is more accurately represented here, by simulating the attenuation of alloys of water and HA400. These materials are reasonably accurate analogues for general soft-tissue and bone, respectively (Maaß et al., 2011).

The research described from this point on therefore consists of the development of various functions designed to wrap-around and run alongside the Trueview algorithm. First, providing structure to call the simulation functions in an iterative manner, and carry out mathematical operations between these function calls to gradually refine the model into a more accurate estimation.

An important factor to note here is that the advantage of the Trueview simulation is not its accuracy, but its speed. Figure 5.1 shows the real, down-sampled X-

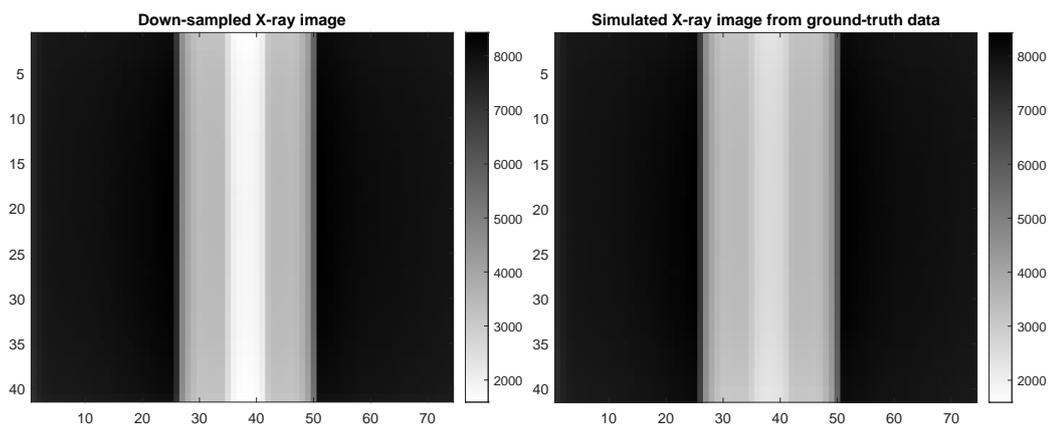


Figure 5.1: Down-sampled X-ray image for the Mark II Arm Phantom (left), and the simulation produced by Trueview for that anatomy (right).

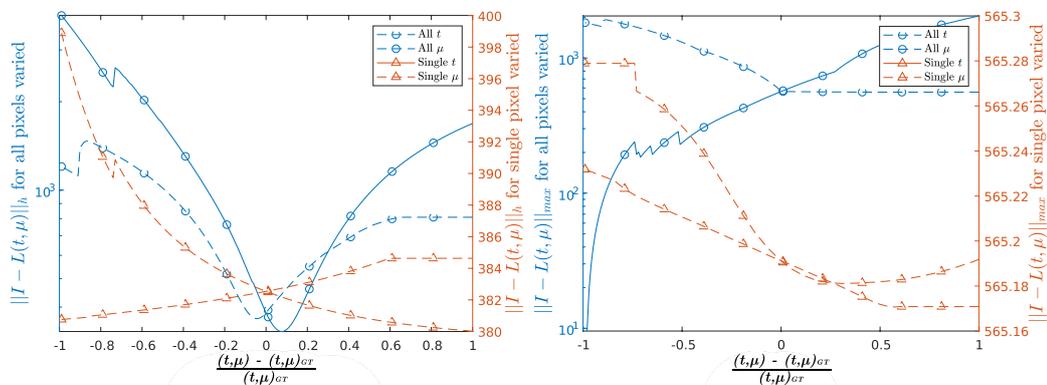


Figure 5.2: Responses in h and max norms to perturbations in thickness and alloy for single voxels, and all voxels simultaneously.

ray image provided to the algorithm, alongside the simulation that the Trueview algorithm produces for that anatomy. This is an image of the M2AP, a rounded block of PMMA with an aluminium rod core designed to approximate the layout of a human upper-arm, but with a simple, known thickness and density.

It is essential to recognise here that, despite access to ground-truth data, the approximations that the simulator makes mean that the simulated X-ray image has distinct differences from the real image. This is explored further in Figure 5.2. In Figure 5.2 the responses of two different norms over image residuals ($I - L(t, \mu)$) are plotted against the perturbation applied to the model. Each graph shows four different modes of perturbation, altering the thickness or alloy of a single voxel in the centre of the model, or altering the thickness or alloy of all voxels at once.

As is immediately apparent from the h -norm (discussed below) graph the minimum of the norm when all voxels are altered lies not at zero (i.e where the model is exactly equal to the ground truth), but slightly above or below zero for thickness and alloy, respectively. This shows that the difference between the real and simulated images from the M2AP is minimised when either the thickness is slightly overestimated, or the alloy is slightly underestimated.

This confirms that the simulator runs over-bright. Either increasing thickness or increasing density (reducing μ) would cause more photons to be absorbed or scattered, and fewer would reach the detector, leading to a darker image. As both of these lead to a more accurate simulation, the simulator must be brighter than it should be by a small amount.

Because alloy value corresponds to the proportion of soft-tissue in an idealised material composed of just soft-tissue and bone, and bone absorbs more X-rays than soft tissue, a smaller alloy implies a denser, more X-ray absorbing material.

It is also important to address here the down-sampling factor that has been applied to both images in Figure 5.1, and to the images used in the calculations for Figure 5.2. The X-ray simulation algorithm developed by IBEX Innovations is based on a ray-tracing algorithm, where photons are expected to scatter along a range of possible paths through the model, and the model is used to estimate the density of the material each of these photon beams will experience along its given path. For each pixel in the image, a path must be calculated to each other pixel. Thus for an image with N pixels, $N(N - 1) = O(N^2)$ paths must be calculated every time a simulation is produced.

The native resolution of the X-ray image used in Figure 5.1 is 1719×963 pixels, making N prohibitively large to perform even a single X-ray simulation on conventional hardware. As such, the X-ray images and the databases used to produce the simulation must be down-sampled. A standard within the company that supplied the code is a down-sampling factor of 23, chosen by IBEX to accurately represent

scatter physics at high speed, and used in this work to more closely align with their work. This means that both the x and y dimensions of the $(\mathbf{t}, \boldsymbol{\mu})$ model and simulated image will be approximately equal to 1719×963 each divided by 23, leading to an image of 75×42 . This still requires nearly ten million paths calculated per simulated image, but thanks to modern graphics-processing hardware optimised for ray-tracing operations, such simulations are achievable in reasonable time, i.e seconds, not hours.

This has a knock-on benefit for our algorithm, as stochastic gradient descent requires multiple simulations for each parameter update, and the number of parameter updates scales with the number of pixels. If each parameter is incorrect by some small amount, then the minimum number of iterations required to create a correct estimate is $4N$, leading to $O(N^3)$ passes to resolve a single image.

The basic structure of the function we seek to optimise in order to reconstruct the anatomy being imaged comes from this simulation.

$$J(\mathbf{t}, \boldsymbol{\mu})_{\text{Unregularized}} = \|I - L(\mathbf{t}, \boldsymbol{\mu})\|_h \quad (5.1)$$

Where $L(\mathbf{t}, \boldsymbol{\mu})$ represents the output of the TrueView simulator, derived from input matrices $(\mathbf{t}, \boldsymbol{\mu})$, and I represents the down-sampled input X-ray image. The h -norm shown in equation 5.1 is a construction of our own, and is defined as follows:

$$\|x\|_h = \theta \|x\|_{\text{fro}}^2 + (1 - \theta) \|x\|_{\text{Max}}^2, \quad 0 \leq \theta \leq 1$$

This provides a tunable norm (in terms of θ) which represents a combination of the Frobenius norm, and the max norm over the image residuals. This is used because it is recognised that there is value in each of these norms for minimising the difference, as they represent large-area and small-area errors, respectively.

Equation 5.1 represents the unregularized case for our problem. This will be the jumping-off point for defining the functional regularizers in the sections that follow.

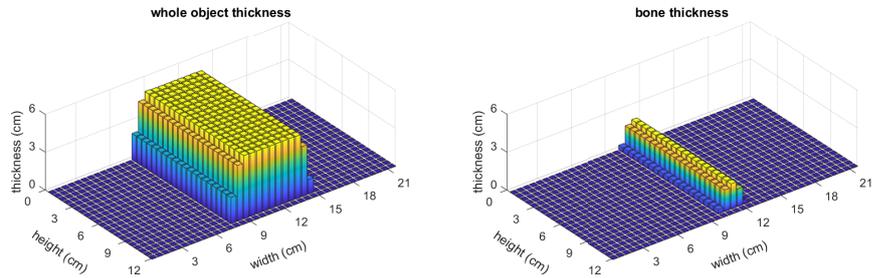


Figure 5.3: ground truth thickness and alloy information for the M2AP

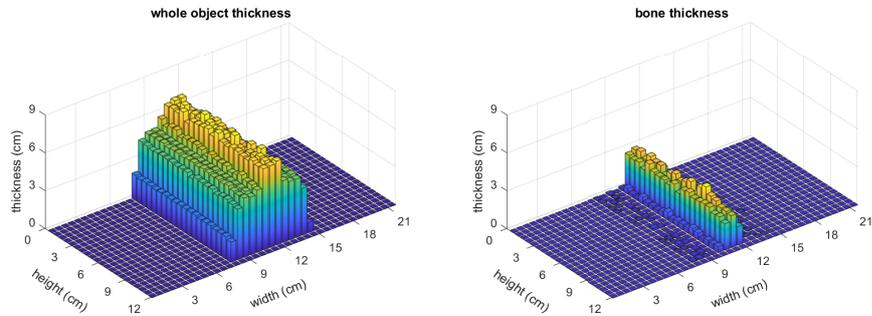


Figure 5.4: Result of five cycles (6500 iterations) of unregularized optimization, starting from ground truth information.

5.1.1 Comparison against exploratory research

Figure 5.4 shows the equivalent of test case 1 from Chapter 4, where we set off the optimiser with ground truth information as the initial state (shown for comparison in Figure 5.3). The early-stopping criteria used in the experiments discussed in 4.2.6 is unavailable to us here, as we understand that the X-ray simulator is good, but imperfect, and as such we cannot expect an exact match between the simulated image and the real image, even for the ground truth data.

The result of this absence of early-stopping, combined with the ill-posed nature of the problem, resulting in the estimate seen in Figure 5.4, where many iterations

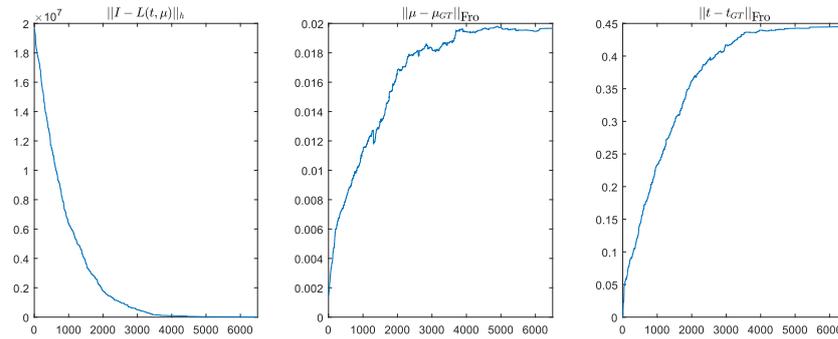


Figure 5.5: h -norm of image residuals and Frobenius norms over \mathbf{t} and $\boldsymbol{\mu}$ for five cycles (6500 iterations) of unregularized optimization, starting from ground-truth information.

have been carried out, creating an estimate which is less accurate than the initial state.

Figure 5.5 illustrates this issue further. In this figure, the norms over the thickness and alloy residuals (difference between final predicted and ground truth) are shown alongside the norm over which the unregularized optimiser has minimised. The problem we face is transparently clear when we recognise that as the image residuals decrease, the thickness and alloy residuals increase — the exact opposite behaviour to what we require to solve the problem. We reflect here on the statement of the problem from the start of Chapter 2:

In a single X-ray image, it is impossible to distinguish between a narrow, highly absorptive material, and a deep, transmissive one.

Precisely that problem is being observed here. An accurate simulated image is being produced from an entirely inaccurate model, where thickness is increased where alloy should be decreased, and vice-versa.

Thus, it is made all the more clear that functional regularization is required. The form of the function of the problem we are solving needs to be modified and warped such that the ground truth information represents a minimum in our cost function, so that further iterations do not lead to divergence.

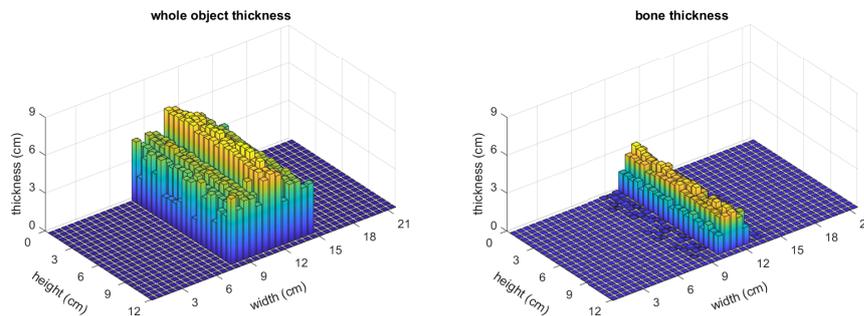


Figure 5.6: Thickness and alloy models estimated from unregularized optimization

5.2 Regularization

Because the system that was constructed to explore the problem area should be representative of the real problem, the regularizers described in 4.3 should be suitable application for a problem based on real physics. The segmentation and smoothness regularizers described in 4.3.1 and 4.3.2 respectively were designed to be relevant to the real problem, so they are implemented here in more-or-less the same form as they were set out previously.

Naturally, the implementation has to be written from scratch, as the regularizers are being applied to a different model representing a comparable physical system. Those differences are set out in the following sections.

Figure 5.6 shows the unregularized result we will compare against in the following sections. This is the result of five cycles — or 6500 iterations — of our iterative optimiser.

5.2.1 Smoothness

The smoothness regularizer takes a comparable form as previously designed, by convolving a Gaussian filter with a set of parameters, weighting this in accordance

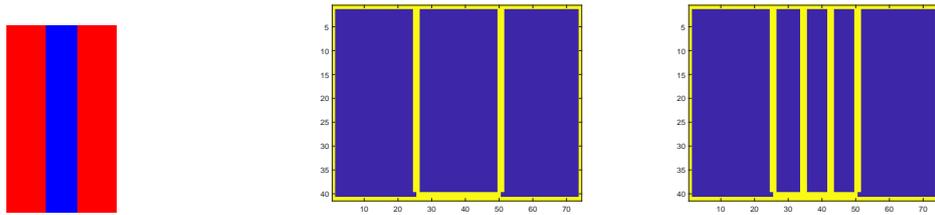


Figure 5.7: Example segmentation of the mark II arm phantom (left), and the edges along which thickness (middle) and alloy (right) smoothness would not be active.

with an edge detection, and letting the magnitude of this weighted convolution determine the appropriate smoothness penalty.

Instead of applying this process to all three dimensions of a 3D model of voxels, this must be applied along the x and y dimensions of the thickness and alloy parameters. We had previously reasoned why it would be appropriate to apply this penalty to alloys (providing the expected edges are preserved), and similar reasoning can be used to justify its application to thicknesses. Similarly to alloy, thickness is a representation of a spatial property of an organic object, sampled at a rate which is generally small compared to the object's size. This means we can expect that there will be no sharp changes in the thickness of the object, except for along the objects boundaries. This limitation on the assumption is the same as one of the limitations placed on the assumption of alloy smoothness, and applies to the boundary between open-beam and not-open-beam classifications in the segmentation of the X-ray image.

In an effort to combat an error that could occur where non-smooth artefacts could propagate along diagonals, the 1D partial second derivative filter was replaced with two 2D partial second derivatives of the form

$$F = \begin{bmatrix} -1 & 2 & -1 \\ -4 & 8 & -4 \\ -1 & 2 & -1 \end{bmatrix}$$

and its transpose, F^T , for smoothness along the x and y axes respectively.

The edge detection algorithm was replaced by a similar 1D Gaussian convolution

to that described in 4.3.2, but applied to the segmentation, and combined with a second step that takes the ceiling of the normalised result, such that it returns a Boolean output of 0 where a pixel is surrounded by pixels that have been given the same classification, and 1 otherwise.

Two different edge detection binary masks were produced here, for the boundaries of the whole anatomical object, and the boundaries of just the bone region.

Furthermore, in the case of the smoothness of the thickness parameters, it was reasoned that a sharp on-off binary mask applied as the weighting of the regularizer for different pixels was not an optimal implementation. This was based on two principal factors:

1. At high resolutions (or low down-sample factors, if preferred) and for certain anatomies, the region over which the thickness transitions from near-constant to zero is likely to be spread over more than one pseudo-pixel. In this case, limiting the region over which smoothness is de-prioritised to a single (pseudo-)pixel could stand to reduce the accuracy of the solution near to these edges.
2. If the segmentation (or human labeler, for that matter) has misjudged the edge of the object, then the identified edge will be meaningless. Nonetheless, it seems sensible to put faith in the segmentor, and not disregard its output entirely. Thus, pixels close-to the identified edge should be treated as edge-like.

This leads us to apply a Gaussian filter to the detected edges, with the intention of blurring them out. This blurring is then normalised, so that the penalty is deactivated along the detected edges, and gradually reduced for pixels near to these edges. A sharp change in thickness will be more heavily penalised the further it is from a detected edge.

This effect is only applied to the thickness, as the sampling rates and anatomy scales used make it unlikely that sharp changes in bone-thickness/alloy will take

place over more than one pseudo-pixel.

The sum of the x and y Gaussian convolutions with the alloys and thicknesses give us alloy- and thickness-smoothness images respectively, where each pixel represent an approximate measure for how non-smooth the estimate is at that point. These smoothness images are then multiplied with the two edge detection masks in the case of alloy, and with the blurred object edges in the case of thickness, to give us a measure of smoothness at each pixel, weighted by how relevant that smoothness is to the problem at hand.

The Frobenius norm is then taken for each of these images, and is squared and halved in an effort to accurately functionally represent that this is a second-order derivative. The values are then multiplied by pre-set weighting hyper-parameters, α_{D1} and α_{D2} , for thickness and alloy respectively. We use two different weighing parameters here because we expect the acceptable rate of change of thickness and alloy to be different. The values for both alloy and thickness smoothness are returned by the function containing the above, and both values are added to the general cost function, $J(\mathbf{t}, \boldsymbol{\mu})$

Formulating this in relation to equation 5.1 gives us:

$$\begin{aligned} J(\mathbf{t}, \boldsymbol{\mu})_{\text{Smoothed}} &= \alpha_{D1} \frac{1 - \text{edge}(I_{\mathbf{t}})}{2} \|\Delta \mathbf{t}\|_{fro}^2 \\ &+ \alpha_{D2} \frac{1 - \text{edge}(I_{\boldsymbol{\mu}})}{2} \|\Delta \boldsymbol{\mu}\|_{fro}^2 \end{aligned} \quad (5.2)$$

Adding this to $J(\mathbf{t}, \boldsymbol{\mu})_{\text{Unregularized}}$ gives us a compound cost-function, containing domain knowledge as well as an estimate of accuracy from the simulator.

It is important to note that the tuning factor, α_D , on this regularizer (and all other regularizers discussed in this chapter) has to be carefully selected. Naturally, a value too low will tend towards the unregularized, and therefore unsolveable, problem. However, a weighting factor too high will prohibit valuable updates to the model, tending towards a smooth but vastly incorrect estimation.

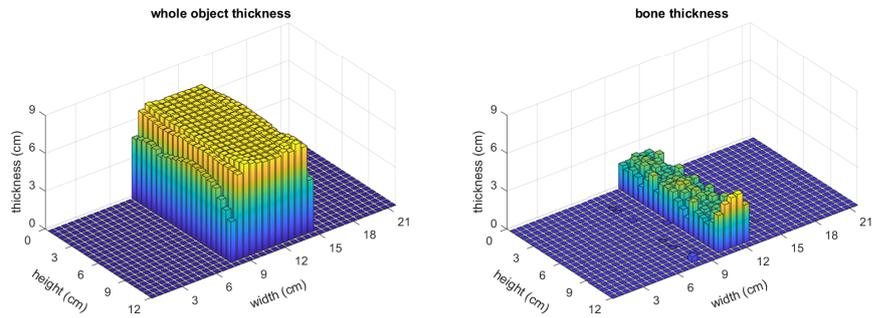


Figure 5.8: Result of a failed convergence, hampered by an overly strong regularization over the smoothness.

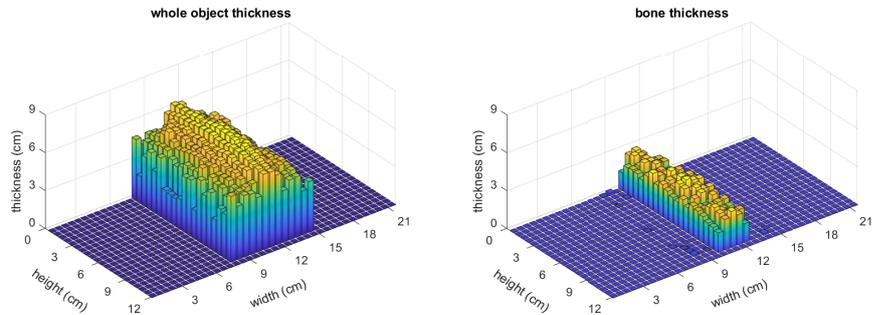


Figure 5.9: Appropriately tuned implementation of the smoothness regularizer, leading to a more accurate model than the unregularized case

This is shown in Figure 5.8, where the majority of thickness values have been prevented from diverging from their initial value of 7cm, because any such change would introduce a penalty in the smoothness which is then magnified by an overly-large weighting factor (10^5 in this case). The edges, where the regularizer is gradually re-activated, do however show the appropriate alterations.

By comparison, Figure 5.9 shows a successful implementation of the smoothness regularizer, with a final model that displays more overall smoothness in thickness, at a reasonably accurate value. The value of the thickness is not perfectly correct, as it differs from the true by between 20% and 40%, but it has allowed a more accurate alloy estimation to be made, by constraining one of the two parameters

within a reasonable bound, and thereby significantly reducing the ill-posed-ness of the problem.

5.2.2 Segmentation

The segmentation regularizer follows the structure of the one proposed for the example problem closely. In principle, the example problem’s mechanics should match exactly with the real physics for the purpose of this regularizer.

For regions of the image that have been segmented as soft-tissue-only, the regularizer will add a penalty proportional to the difference between the model’s alloy in that region, and the predicted alloy (which will be equal to 1 at these points).

In regions where the segmentation has identified a combination of soft-tissue and bone, the regularizer will penalise any alloy parameter that has been estimated outside the valid range $0 \leq \mu < 1$. As explained in 4.3.1, the domain knowledge provided by the segmentation does not provide any estimate for what the alloy in these regions *should* be, only what it *should not* be. Therefore the magnitude of the penalty term here is constant, not proportional to some difference from an expected value.

This is achieved functionally as a multiplication of operators applied to the segmentation matrix and the alloy matrix of the model. For pixels classified as soft-tissue by the segmentation, an operator is applied to the segmentation to yield a binary mask, valued at 1 for pixels given that classification, and zero otherwise. This is multiplied element-wise by $(1 - \mu)$, and multiplied again by an operator applied to μ , which gives a binary mask valued at 1 where the alloy satisfies $(\mu < 1)$.

For pixels classified as a soft-tissue–bone alloy by the segmentation, a matrix is produced from an element-wise multiplication of two binary masks, one indicating the pixels classified as soft-tissue–bone alloy, and a second indicating alloys in the alloy matrix with a value $(\mu \geq 1)$. This second binary mask is denoted by $[\mu \geq 1]$, and is equal to 1 where the condition in the brackets is satisfied..

These steps produce two matrices, describing the penalty parameter for a given pixel in terms of mis-classified soft-tissue and bone, respectively. The Frobenius norm of each of these matrices is calculated, and each scalar output is multiplied by a pre-determined weighting factor, α_C . The sum of these two scalars is added to the general cost function, $J(\mathbf{t}, \boldsymbol{\mu})$.

$$J(\mathbf{t}, \boldsymbol{\mu})_{\text{Segmentation}} = \alpha_C \left(\|seg_1 \odot (1 - \boldsymbol{\mu})\|_{fro} + \|seg_2 \odot [\boldsymbol{\mu} \geq 1]\|_{fro} \right) \quad (5.3)$$

Where, as defined in Subsection 4.3.1 \odot denotes the Hadamard product, an element-wise multiplication of two matrices.

When we consider the impact this regularizer has when acting independently of the others, it's effect can be visually subtle. Particularly on the example model we demonstrated the smoothness regularizer on (i.e. Figures 5.6, 5.9), the alloys are initialised as generally correct values, and there is no initial issue for the regularizer to correct. If we instead initialise the model with significantly erroneous alloy values, then the value the regularizer provides is immediately clear in the convergence of the Frobenius norm over alloy residuals. Exactly these data are presented in Figures 5.10 and 5.11. It is clear to see in these figures that the accuracy of the alloy estimation is significantly better thanks to the segmentation regularizer. Despite the fact that the cost function actually converges less in the regularized case, the alterations made to that function mean that the overall accuracy of the result is better.

5.2.3 Physicality

As with the simulator designed for the simple problem-case in 4.2.1, the output of this, more complex simulator is not defined for model parameter values outside of an accepted range, and will cause the simulator to fail. Up to now, that issue has been resolved by just constraining the values to within the valid range. When the

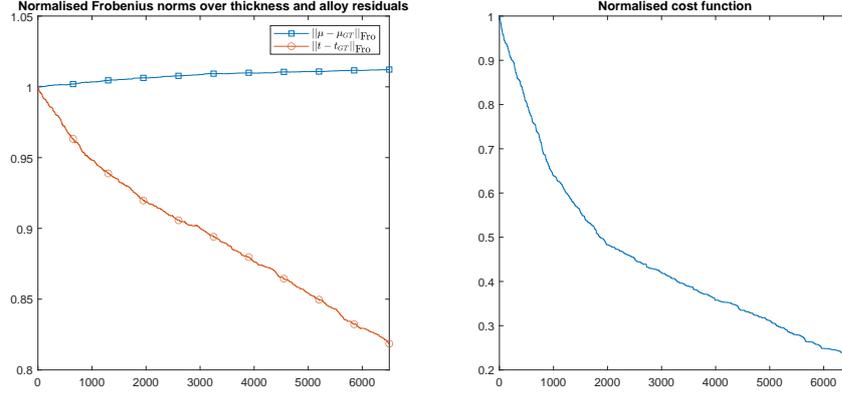


Figure 5.10: Five cycles of unregularized optimization from highly inaccurate initial condition. Without the segmentation regularizer, the Frobenius norm over the alloy residuals is incapable of convergence.

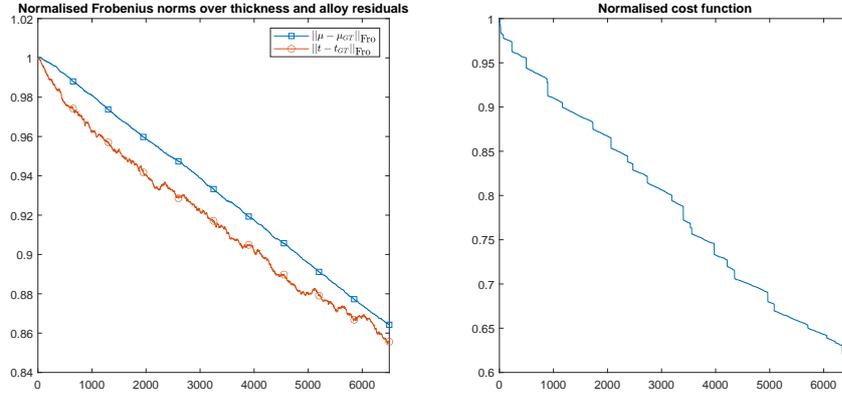


Figure 5.11: Five cycles of optimization with only the segmentation regularizer active, from highly inaccurate initial condition. The presence of segmentation-based regularization has allowed the alloy residuals to converge alongside the thickness residuals.

alloy was previously restricted to $0 \leq \mu \leq 100$, values of $\mu < 0$ were treated as equivalent in accuracy as $\mu = 0$, with the same being true for $\mu > 100$ being treated as $\mu = 100$. The model was constrained when it was provided to the simulator, but a value outside the valid range was considered acceptable otherwise.

It seems reasonable to recognise, however, that this is a sub-optimal implementation. This is especially true when the smoothness regularization is brought into consideration. These limits on the range of acceptable values represent further domain knowledge about the system which can be encoded, and thus should be.

The alloy in the more accurate model is ill-defined in much the same way as in the exploratory problem, in that only values in the range $0 \leq \mu \leq 1$ will produce a well-defined output. This makes intuitive sense, as an alloy of two materials cannot be made up of more than 100% of a given material, nor less than 0%.

The limits on the thickness parameters is less intuitive, but limits are helpfully provided by the construction of the simulator. Naturally, any thickness value $t < 0$ represents a physically impossible input, and is thus ill-defined. For the upper-limit on thickness, one can reason that in practice X-ray detectors have a practical limit to the size of an object they can image. This upper limit is however very dependent on the settings and set-up of the detector. The upper limit imposed on the problem for us is that the database of simulations used to produce a simulated image only includes entries up to a given value, which becomes the greatest possible thickness for it to estimate. If a problem-case frequently required estimation of thicknesses significantly greater than this maximum (in a veterinary application, for example), increasing the upper-limit is simply a case of pre-computing a wider range of simulated scatter kernels which can be searched when producing a simulated image. This will have an associated effect on run-time, as searching a database with a greater number of entries will take longer.

To balance these impacts, the upper-limit to thickness selected for this implementation is 30 centimeters. As with previous regularizers, this domain knowledge is encoded through two matrices, containing the relative penalty of a given pixel for thickness and alloy respectively.

For the alloy penalty, this penalty matrix is the element-wise sum of two operators applied to the alloy matrix. First, an element-wise minimum of zero and the current element $\min\{0, \mu\}$, thereby returning zero for elements that are equal-to or greater-than zero, and the value of the alloy at that element otherwise. Second, an element-wise maximum of zero and the alloy at that element minus 1 $\max\{0, \mu - 1\}$, returning zero for elements below the allowed maximum, and the magnitude by which they exceed the maximum otherwise.

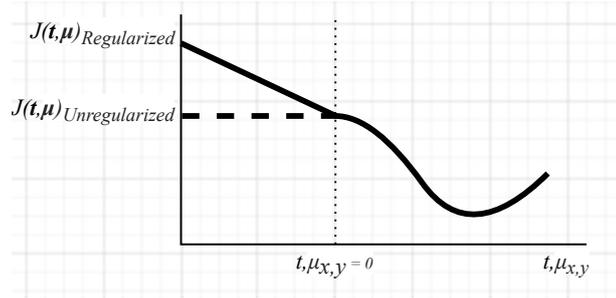


Figure 5.12: Example of how the cost function for a given $(\mathbf{t}, \boldsymbol{\mu})$ parameter changes when this regularizer is introduced. The plateau for values outside the physical range is removed, creating a slope that can be descended.

The result of this element-wise sum is a matrix where the magnitude of each value represents how far corresponding element in the alloy matrix is from the valid range. The sign of these values corresponds to whether they are above or below the valid range, but this is meaningless since only the Frobenius norm of the matrix will be considered, which involves taking the square of each element.

The same process is applied to the matrix of thickness values, but with the second operator taking the maximum of the elements of the thickness matrix and 30, $\max\{0, \mathbf{t} - 30\}$, to give a comparable output.

The Frobenius norm of each of these operator sums is taken, and summed together. This scalar is multiplied by a weighting factor α_B , and added to the general cost function.

$$\begin{aligned}
 J(\mathbf{t}, \boldsymbol{\mu})_{\text{Physicality}} = & \alpha_B \left(\|\min\{\mathbf{t}, 0\} + \max\{0, \mathbf{t} - \mathbf{t}_{\max}\}\|_{fro} \right. \\
 & \left. + \|\min\{\boldsymbol{\mu}, 0\} + \max\{0, \boldsymbol{\mu} - 1\}\|_{fro} \right) \quad (5.4)
 \end{aligned}$$

This alteration is shown graphically for a single parameter in Figure 5.12.

To demonstrate this added value, it is worth presenting the unregularized case in a different context. Figure 5.13 shows the magnitude of the difference between the alloy values in the final, unregularized estimation and their nearest physically valid value. While most of the estimated values are zero here (i.e. within the

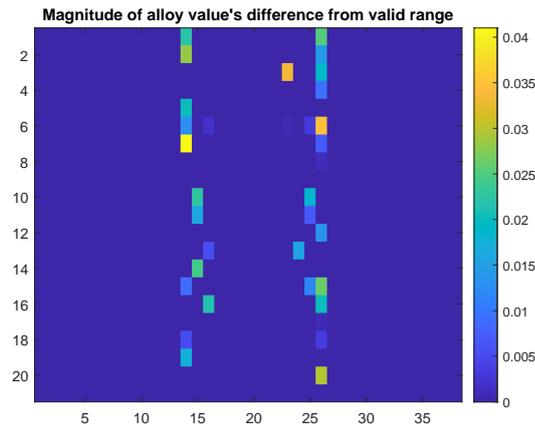


Figure 5.13: Depiction of invalid alloy values after five cycles of unregularized optimisation.

valid range), there is a significant number that lie outside of this range. With the regularizer active, there are no non-zero values, for either thickness or alloy.

As a physically invalid value cannot improve the simulated image (due to constraints on the simulator), removal of any invalid values can only have a positive impact on the material estimation.

Pseudocode for this regularizer is included in Appendix A.3.

5.2.4 Continuity

With the combined impact of the regularizers defined in 5.2.1, 5.2.2 and 5.2.3, it was recognised from experimental work that these three regularizers were insufficient to accurately estimate realistic anatomies.

Because of the absence of smoothness constraints along edges, these regions were developing sharp spikes, which then had the potential to propagate away from those edges.

Given that the smoothness regularized defined in 5.2.1 is analogous to an l_2 regularizer frequently seen in neural networks, it seemed logical to continue with the analogy, and introduce a regularizer linked to the l_1 norm.

In related applications, l_1 regularization seeks to minimise the absolute value of coefficients of the function being estimated. In our case, the function being estimated can be replaced with a function describing how thickness or alloy changes with respect to the x and y axes. We do not have access to the coefficients describing this function, but we can approximate the effect by trying to minimise the derivative of the function.

To achieve this, we introduce a similar convolution to the one described in 5.2.1. Whereas that regularizer convolved the spatial parameters with a Gaussian function, this regularizer will convolve with filters to represent the first derivative.

The filters used in this convolution are

$$\begin{bmatrix} -1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & -1 & 1 \end{bmatrix}$$

to determine continuity along the x dimension, and the transpose of these matrices,

$$\begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$$

to determine continuity along the y dimension.

The two convolution windows are used to look left and right (or up and down, in the transpose case) of a given parameter, producing two different matrices for each dimension of each parameter. An element-wise maximum is taken of these two is taken, such that the most significant contribution from the two is considered.

The x and y components for each parameter are then summed element-wise, producing a continuity measurement matrix with the same dimensions as the parameter matrices. Finally, the Forbenius norm of each matrix is multiplied by a weighting variable, α_E , and the two weighted scalars are added to the cost function.

$$J(\mathbf{t}, \boldsymbol{\mu})_{\text{Continuity}} = \alpha_E \left(\|\Delta_h \mathbf{t}\|_{fro} + \|\Delta_h \boldsymbol{\mu}\|_{fro} \right) \quad (5.5)$$

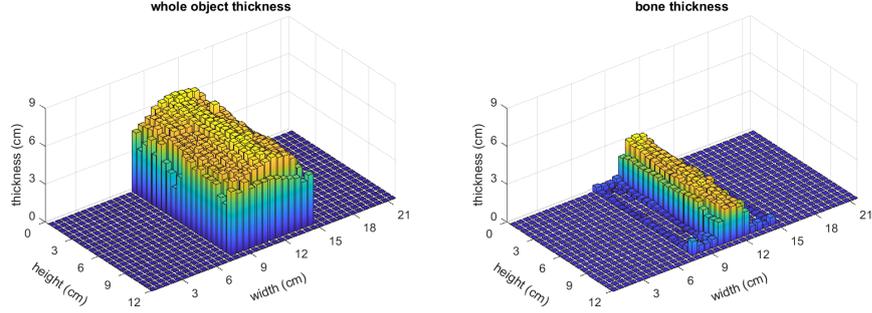


Figure 5.14: Five cycles of optimization applied to the M2AP, with continuity regularization over the ℓ_1 applied. This reduces sharp changes in either parameter space, rendering a more organic estimate.

As with the smoothness regularizer defined in 5.2.1, the benefit in the continuity regularizer can be recognised from visual inspection. Figure 5.14 presents the regularized complement to Figure 5.6, with only the continuity regularizer active. While the final result is inaccurate, it does nonetheless preserve the appropriate approximate structure. This is the impact we are hoping for, and when combined with other regularizers, we can expect an overall more accurate result.

This and the previous subsections have provided us with four regularizers, presented in Equations 5.2, 5.3, 5.4 and 5.5. These serve to guide -but not constrain - our optimization algorithm, such that minimizing our cost function J also minimises the difference between real and estimated thickness t and alloy μ . Thus, the form of the cost function J , first defined in Eq. 5.1 is modified to become:

$$\begin{aligned}
 J(\mathbf{t}, \boldsymbol{\mu})_{\text{Regularized}} = & \|I - L(\mathbf{t}, \boldsymbol{\mu})\|_h & (5.6) \\
 & + J(\mathbf{t}, \boldsymbol{\mu})_{\text{Smoothed}} \\
 & + J(\mathbf{t}, \boldsymbol{\mu})_{\text{Segmentation}} \\
 & + J(\mathbf{t}, \boldsymbol{\mu})_{\text{Physicality}} \\
 & + J(\mathbf{t}, \boldsymbol{\mu})_{\text{Continuity}},
 \end{aligned}$$

or

$$\begin{aligned}
 J(\mathbf{t}, \boldsymbol{\mu})_{\text{Regularized}} = & \quad \|I - L(\mathbf{t}, \boldsymbol{\mu})\|_h & (5.7) \\
 + & \quad \alpha_{D1} \frac{1 - \text{edge}(I_{\mathbf{t}})}{2} \|\Delta \mathbf{t}\|_{fro}^2 \\
 + & \quad \alpha_{D2} \frac{1 - \text{edge}(I_{\boldsymbol{\mu}})}{2} \|\Delta \boldsymbol{\mu}\|_{fro}^2 \\
 + & \quad \alpha_C \left(\| \text{seg}_1 \odot (1 - \boldsymbol{\mu}) \|_{fro} \| \text{seg}_2 \odot [\boldsymbol{\mu} \geq 1] \|_{fro} \right) \\
 + & \quad \alpha_B \left(\|\min\{\mathbf{t}, 0\} + \max\{0, \mathbf{t} - \mathbf{t}_{\max}\}\|_{fro} \|\min\{\boldsymbol{\mu}, 0\} + \max\{0, \boldsymbol{\mu} - 1\}\|_{fro} \right) \\
 + & \quad \alpha_E \left(\|\Delta_h \mathbf{t}\|_{fro} + \|\Delta_h \boldsymbol{\mu}\|_{fro} \right).
 \end{aligned}$$

5.3 Descent methods

The results presented up to this point have been achieved using a custom greedy algorithm for optimization, as opposed to the SGD algorithm described in Section 4.2.3. In this section, we seek to justify that decision, and explain the algorithm that was chosen for producing results.

5.3.1 Bi-directional Greedy Descent

First, it is important to re-state that the nature of the problem space in which we are trying to descend is high-dimensional, non-smooth, and non-convex (Rockafellar, 2015). As such, the assumptions that are made in finite-difference gradient-estimation and SGD can not necessarily be expected to hold. This resulted in issues (explored in the subsequent subsections) where SGD and similar methods would lead to over-stepping of minima, resulting in computational work being wasted, moving towards less-accurate estimates of the spatial model. The solution to this problem was to replace the gradient-estimating algorithm with a greedy algorithm (Edmonds, 2008).

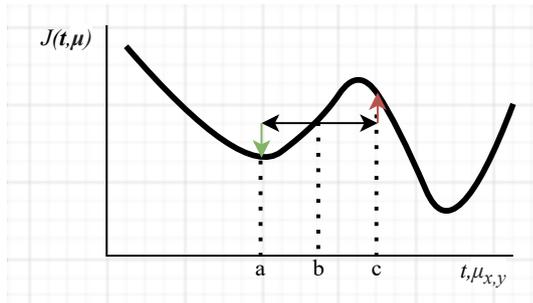


Figure 5.15: A single iteration of bi-directional greedy descent, where b , is the initial value of $\mathbf{t}, \mu_{x,y}$, and a and c are the modified values in either direction. In this case, the current best estimate will be updated to a , as it has a lower cost than c , despite being a local minimum.

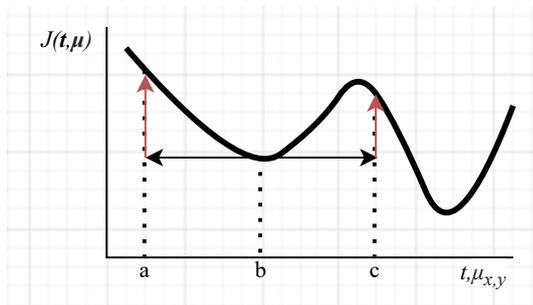


Figure 5.16: A single iteration of bi-directional greedy descent (subsequent to Figure 5.15), where b , is the initial value of $\mathbf{t}, \mu_{x,y}$, and a and c are the modified values in either direction. In this case, the current best estimate will not be updated, as both a and c have higher associated costs.

For each iteration of this optimization algorithm, a single thickness or alloy parameter is randomly selected, and modified by a small random value in both the positive and negative directions. Each of these modified states is evaluated via the cost function, and compared against the cost value of the original, un-modified state, $J(\mathbf{t}, \boldsymbol{\mu})$. Whichever of these three states has the lowest cost J is accepted, and the modification becomes part of the current best-estimate. If neither modification has a lower cost than the un-modified, then the spatial model remains unchanged. Finally, the value of the cost function for the accepted state is stored for the subsequent iteration, so that the cost of the un-modified case need-not be calculated at the next iteration. This process is shown visually in Figure 5.15.

In this way, no alteration can be made to the spatial model which leads to a worse estimate than on the iteration before, *i.e.* the cost function can only remain

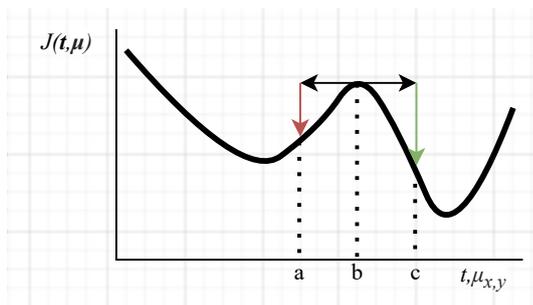


Figure 5.17: A single iteration of bi-directional greedy descent, where b , is the initial value of $\mathbf{t}, \mu_{x,y}$, and a and c are the modified values in either direction. In this case, the current best estimate be updated to c , as c has the lowest associated cost.

constant or decrease.

It is clear that this algorithm is not robust to local minima (as is the case in Figure 5.15). Once stuck within a suitable large local well, small changes in either direction for that parameter will result in higher cost values, and the best-estimate will remain exactly where it is. Given that we have just clarified that the problem is non-convex, it would thus be reasonable to question this choice of descent method. The solution comes from the fact that the problem is high-dimensional, and that while local minima do exist within that space, we expect that the nature of the problem and the cost function we have defined mean that a suitable wide local minima in all dimensions is unlikely to occur with correctly tuned parameters.

Furthermore, we can define the way that the small random modification values are generated, such that while they are generally small, larger values are rarely selected, allowing the algorithm to jump out of local minima, with enough iterations. One could go further here, making the probability distribution of update values a function of iterations, such that over time fewer and fewer large jumps occur, leading to an algorithm more akin to simulated annealing, as in 4.2.5.

Pseudocode for this function is included in Appendix A.1.

5.3.2 Stochastic Gradient Descent

Before the descent algorithm described in 5.3.1 was developed, more traditional optimization algorithms were implemented and tested. The first is SGD, implemented in the same way as described in 4.2.3. In this algorithm, a single thickness or alloy parameter is randomly selected, and altered by a small value, δ . This modified spatial model is then fed into the simulator, and the simulated image is used to calculate the modified cost function, $J(\mathbf{t} + \delta_{x,y}^t, \boldsymbol{\mu})$ or, $J(\mathbf{t}, \boldsymbol{\mu} + \delta_{x,y}^\mu)$ depending on which parameter was modified. This modified cost function, along with the cost function of the unmodified case allows us to approximate the gradient of the cost function with respect to the parameter that was modified, via

$$\frac{\partial J}{\partial \mathbf{t}_{x,y}} = \frac{J(\mathbf{t}, \boldsymbol{\mu}) - J(\mathbf{t} + \delta_{x,y}^t, \boldsymbol{\mu})}{\delta}. \quad (5.8)$$

$$\frac{\partial J}{\partial \boldsymbol{\mu}_{x,y}} = \frac{J(\mathbf{t}, \boldsymbol{\mu}) - J(\mathbf{t}, \boldsymbol{\mu} + \delta_{x,y}^\mu)}{\delta}.$$

This finite-difference gradient estimation is shown in Figure 5.18.

The value of $\mathbf{t}, \boldsymbol{\mu}$ is then updated by a quantity proportional to

$$\frac{\partial J}{\partial \mathbf{t}_{x,y}} \quad \text{or} \quad \frac{\partial J}{\partial \boldsymbol{\mu}_{x,y}},$$

weighted by a pre-defined weighting/relaxation coefficient, ω , usually $\omega < 1$. This leads to steps along the parameter space in the direction that reduced the value of the cost at that point, with the steps being longer where the gradient is reducing rapidly, and shorter where the change is more shallow. The weighting seeks to prevent over-stepping minima, which can lead to oscillations around a minimum, instead of convergence.

One substantial failing of the algorithm is that it is not robust to plateaus in the parameter space, where the value of $J(\mathbf{t}, \boldsymbol{\mu}_{x,y})$ does not vary with $\mathbf{t}, \boldsymbol{\mu}_{x,y}$. This is the case, for example, in the unregularized case for values outside the allowed

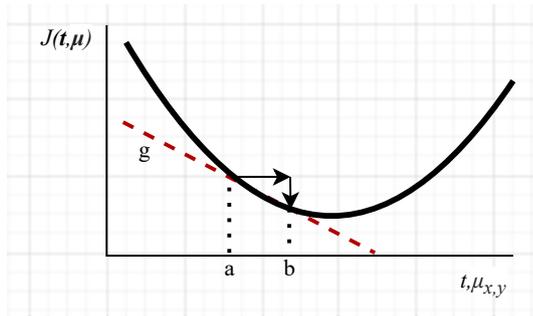


Figure 5.18: Example of a finite-difference gradient estimation for SGD. a represents the initial state, and b some small modification made to that state. Both of these states are evaluated, and their difference is used to make a first-order approximation of the gradient at a , denoted here by the red dashed line, g .

limits of the simulator. In these cases, the gradient is zero, and no updates can be made. Furthermore, in a local or global minimum, the direction opposite to the one tested by δ will be perceived by this algorithm as down-hill, causing the descent algorithm to erroneously step in a direction that leads to a worse estimate of $\mathbf{t}, \boldsymbol{\mu}_{x,y}$.

Figure 5.19 shows five cycles (approximately 6500 iterations) of SGD, applied to the reconstruction of the spatial model for the M2AP. In this instance, regularization has not been applied. From the graph of the cost function during the execution of the algorithm, it can clearly be seen that the shape of $J(\mathbf{t}, \boldsymbol{\mu}_{x,y})$ cannot be accurately estimated using a finite-difference gradient estimation method, as the cost function consistently *increases* instead of decreasing. Since no convergence is observed, it is reasonable to assume that the application of regularizers is going to have no significant, beneficial impact on the quality of the results produced by SGD.

5.3.3 Newton's Method

Newton's method is a technique for finding solutions to equations of the form $f(x) = 0$ by following their derivative towards the axis-intersection (Teodorescu et al., 2013; Levy, 2009). This can be utilised to instead find the minimum of a function, by trying to find solutions to the function's first derivative. At these

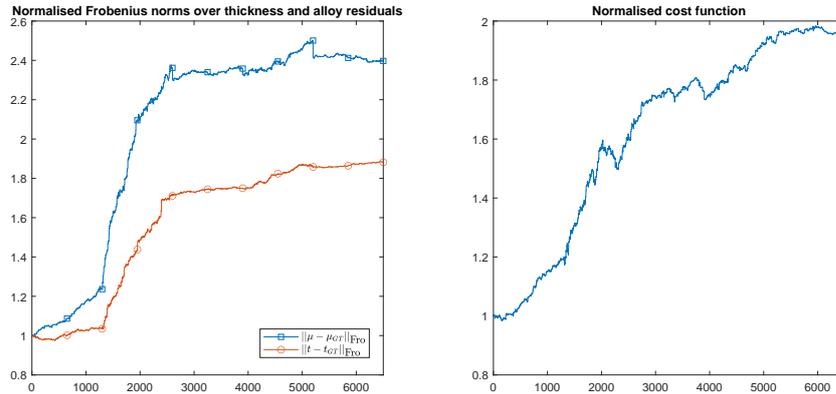


Figure 5.19: Normalised Frobenius norms over thickness and alloy residuals (left) and normalised cost function (right) over five cycles of optimization using stochastic gradient descent, without regularization.

minima (as well as maxima and saddle-points), the derivative is equal to zero, $f'(x) = 0$.

To find the solution to a function Newton's method iteratively calculates the tangent line to a function at a given point, then steps to the solution of that tangent line function. This takes the mathematical form:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Therefore, to instead find the point at which $f'(x) = 0$, we iterate over:

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)} \quad (5.9)$$

In a finite-difference scheme, this is more computationally expensive than SGD, because it requires an estimation of the second derivative. To approximate the second derivative an additional point must be evaluated, close to the current value. In our implementation, we choose $x \pm \delta$. As Newton's method can be thought of calculating the tangent to a function at a point and attempting to solve it, this method can be thought of as calculating a parabolic function through the sampled points, and trying to find the minimum of that parabola.

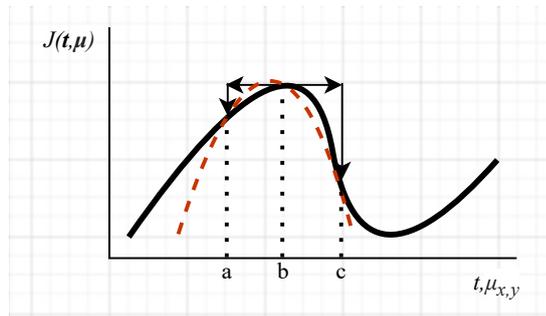


Figure 5.20: Example of the gradient estimation in Newton's method for optimization. b represents the initial state, and a and c represent small alterations to the positive and negative of b . These three states are evaluated, and used to estimate the second-order gradient at b , such as indicated by the red, dashed line.

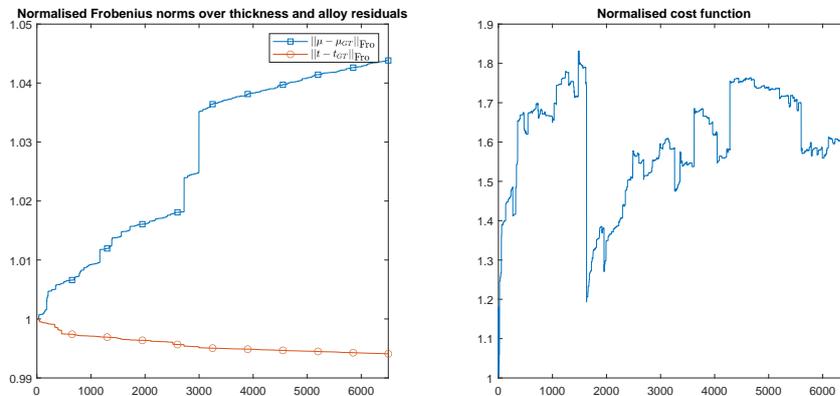


Figure 5.21: Normalised Frobenius norms over thickness and alloy residuals (left) and normalised cost function (right) over five cycles of optimization using Newton's method, with all regularizers active. From the divergence over the cost function, we can infer that gradient-based techniques are unlikely to be successful.

Figure 5.21 shows an attempted convergence over the M2AP, using Newton's method and all regularizers active. While the thickness has been able to converge a small amount, the alloy has deteriorated to a far more significant degree. In some ways even worse to observe is that the cost function has simply not shows convergence from the initial state. Because of the shape and structure of the function we are minimising over, using the gradient of that cost function appears to be non-informative about the location of the global minimum.

5.4 Conclusion

This chapter has covered the most substantial part of the research presented in this thesis. We have adapted the ideas introduced in Chapter 4 to realistic X-ray simulations, then honed our regularizers such that they constrain the real problem enough to become solveable. Four regularizers were presented in the end, each covering different aspects of the model's relationship with real anatomies.

We then explored the different algorithms for optimizing a parametric model according to some non-differentiable cost function. Two classical algorithms were considered and evaluated, to justify our decision to opt for a more hand-crafted algorithm for optimization.

This leaves us with an algorithm that is capable of achieving our stated goal: we are now able to produce a model of an imaged anatomy, from which simulations can be produced to remove scatter. However, we are not able to produce these models efficiently. In the following chapter, we will investigate what techniques can be implemented to increase the time-efficiency of the algorithm we have described here.

Multi-scale

Here we outline the time problem, and the solution we propose using many resolutions. We thus need to explain how we move through resolutions. We established in the previous chapters the need for speed. The algorithm works reasonably well, but it is very slow to achieve results. Since moving to real anatomies and a true simulation method, the algorithm has slowed down, as expected.

At the target resolution, creating a simulated X-ray image using IBEX's X-ray simulation tool takes on the order of 4 seconds. If our spatial model consists of 2000 variable parameters (as it does for the M2AP), applying a single modification to each parameter and evaluating the effect on the cost function takes more than 2 hours. Realistically, each parameter will require more than one update, as the regularizers will reduce the scale of any acceptable modification. Indeed, when demonstrating the regularizers in the previous chapter, the model has been optimizing the spatial model for 6000+ iterations.

For our algorithm to have value in a clinical setting without code optimization, we need to be minimizing the execution time, targeting a value less than a typical work day (8 hours), at least. Multi-scale is the natural solution to this - processing a lower resolution image is faster, and requires fewer steps to make the same change.

6.1 Introduction

When we introduced the physics-based simulation to our algorithm in Section 5.1 of Chapter 5, we made cursory mention to the down-sample factor at which the images were being processed. We described how, because of the inherent high-resolution of the X-ray images, and the ray-tracing nature of the processing algorithm, the images had to be down-sampled to be processed in reasonable time on consumer hardware. We further explained that a down-sampling factor of 23 was selected based on the current best-practise guidelines of the sponsor and provider of the TrueView software, IBEX Innovations.

In practice, 23 is not the only down-sampling factor used by the company, and that the optimal value changes depending on the goal of the work it is being used for. In commercial research and product development, there are times when additional compute-time can be dedicated to high-quality material estimations, as well as times when an even-coarser estimate will more-than suffice.

In this chapter, we will be discussing why this value was selected, comparing the trade-offs involved in varying it, and then exploring how a range of values instead of a single constant can provide added value to the algorithm in several areas. This is not a new concept. multi-grid methods have been used for increased efficiency in many problems before (?). In this chapter, we want to highlight the research work carried out to apply this concept to our problem of interest.

Before continuing, we introduce a short-hand notation for describing these down-sampling factors. For a down-sampling factor of 23, for example, we mean that the x and y pixel dimensions of the input image have each been divided by 23, then rounded to the nearest whole integer. This algorithmically-reduced image will then be referred to as a DS23 image, and we will use this shorthand for the rest of the chapter, with DSN for images that have been reduced by a factor N .

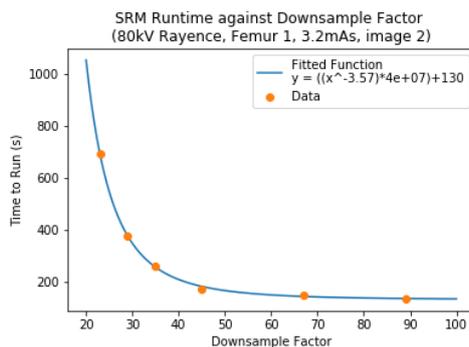


Figure 6.1: Run-time comparison of a set of down-sample factors for a femur image, using non-optimised research code. The values on the y axis are not representative of the values for optimised code, but the relationship between the down-sample factor and run-time is expected to be representative. Note that this figure includes additional resolutions (DS67, DS29) to the ones used when processing the images in our algorithms, as they fall between the factor-two intervals we opted for.

6.2 Resolution-Speed Relationship

Because a set of down-sampling factors have been in use with the software in question for some time, the concept of exploring how these could be used to our advantage was one of the first elements considered in our investigation. It is true that DS23 is used as the default value in most work carried out by the research and development team at IBEX Innovations, as it produces an accurate approximation of scatter dynamics in complex images in short time (< 1 minute for optimised non-research code, at last measurement).

This led us to investigating how the compute time for processing a given image using standard settings varied with the resolution at which it was processed.

Figure 6.1 shows the relationship between down-sample factor and run-time. As can be seen, the compute-time impact is relatively imperceptible for coarse resolutions down to DS45, after which the time to process a given image for the same number of iterations rapidly increases, approaching $4\times$ the DS45 speed for the standard of DS23.

The benefits of processing at very coarse resolutions increase as with the number of evaluations required, i.e as the number of iterations increases, or from descent meth-

ods that require more evaluations of the cost function, such as Newton’s method. Nonetheless, it seems apparent that significant time reductions could be made by only small reductions to the total number of pixels in the image, from a start-point of DS23.

6.3 Limitations on Resolutions Available

In Figure 6.1, it is clear to see that only a relatively small set of values for the down-sample factor have been selected, which makes it is important to explain the choices of DS values. We have stated previously that DS23 is used as an in-house standard, along with a select set of other down-sample factors. A valid question at this point is, why? Why not simulate over a more frequently-sampled range, and why DS23 and not, say, DS20?

This limit on the resolutions that can be processed comes from the simulated Geant4 databases, upon which the X-ray simulator is constructed. Geant4 produces scatter kernels for the spread of X-ray beams through a range of materials, sampled at discrete, regular intervals of 1% (0% bone, 1% bone, 2% bone...). These scatter kernels must then be adapted into a form accepted by the TrueView algorithm with the expectation that those scatter kernels will be applied to a down-sampled image. As such, this scatter data is itself down-sampled, so that it is accurate when applied to an image (or spatial model) where a given physical dimension is represented by a different given number of pixels (or voxels). These down-sampled scatter kernel databases are distinct from one another, and not simply re-scaled versions of each other, as they are required to accurately capture physics at each scale. As such, we were limited to the databases provided by IBEX Innovations.

As such, in order to avoid massive restructuring of the underlying TrueView code and creation of a new data format for containing many resolutions, both of which would be outside the scope of the project, we select a set of resolutions we seek to

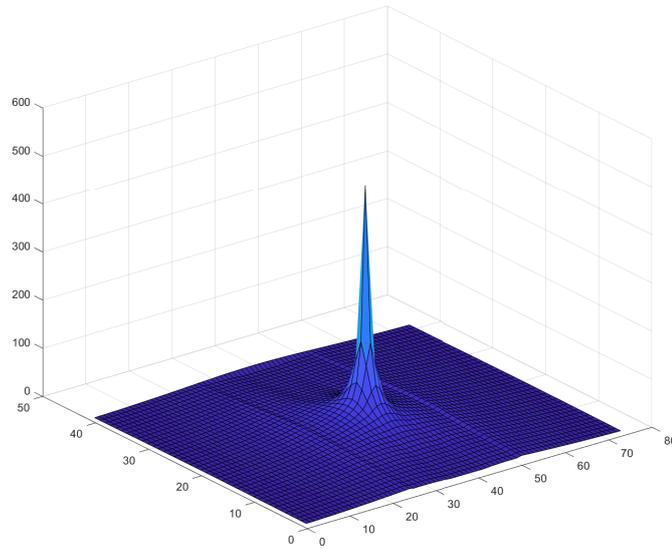


Figure 6.2: An example of a single scatter kernel of a simulated image. The x and y dimensions of this kernel are the same as the image from which it was produced. To produce a simulated X-ray, the element-wise sum is taken of a stack of these kernels, with one kernel produced for each scatter centre within the image. The value on the z axis is a representation of photon-count.

investigate, down-sample the Geant4 data to those various resolutions, and store them for later use.

One characteristic of this simulation process developed by IBEX, is that the scatter kernels are developed as a radially symmetric, one-dimensional function, with the first value representing the centre of the incident photon beam. This means that the kernel itself will have odd-numbered pixel dimensions in both x and y , to ensure there is a single, definitive centre. An example of such a scatter kernel, which would be one term in a sum of kernels to produce a simulated image, is shown in Figure 6.2.

6.3.1 Simulated and Pseudo-pixel Resolutions

For most of the experiments in this thesis, the pre-simulated databases of scatter information selected are DS11, DS23, DS45 and DS89. Figure 6.3 shows the M2AP

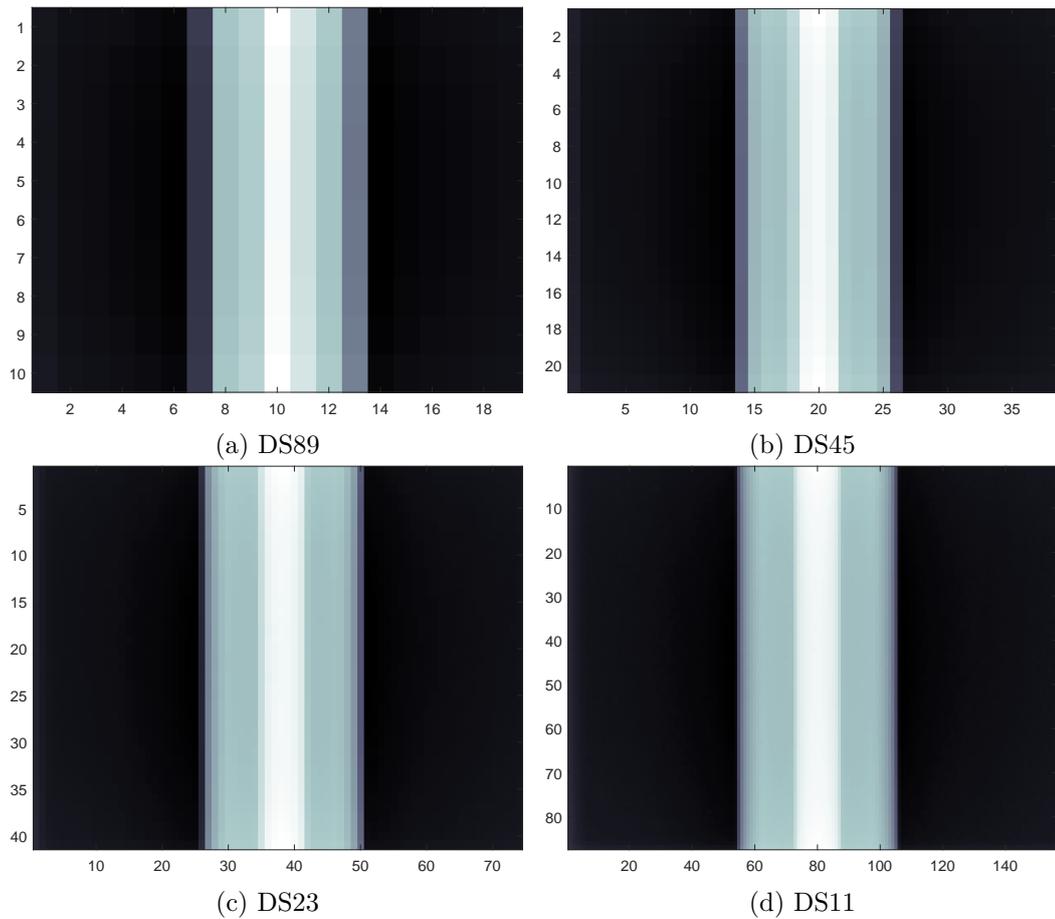


Figure 6.3: M2AP X-ray image, down-sampled to (a) DS89, (b) DS45, (c) DS23, and (d) DS11.

down-sampled to these four resolutions, to give an idea of how much information is stored in each.

For each of these resolutions, the speed-up from using coarse estimations comes from two sources:

1. Each single iteration becomes faster. As the coarser spatial model has fewer elements, the ray-tracing process of making an X-ray simulation, which is the bottleneck of our computations, becomes faster.
2. Fewer iterations are required. As there are fewer parameters defining the output of the simulator, it takes fewer iterations of a stochastic process such as SGD to converge on a more accurate model.

There is a trade-off to these benefits, as the coarser resolutions are less capable of making a sufficiently accurate simulation. Beyond DS89, the ray-tracing process is not anymore the speed increase of a single iteration is negligible (as seen in Figure 6.1), and the inaccuracy of the simulation is significant. We can recognise, however, that this trade-off is inherently tied to the first of these benefits, and not the second. Put differently, if we reduce the number of alterable parameters in the model without changing its resolution, the time cost per simulation will not be changed, but the number of iterations required to optimise that particular model will decrease, leading to faster computation. This speed increase comes at the cost of decreased model accuracy, but these coarse parameterizations can be followed by a small number of iterations at the full parameterization, to recover that lost accuracy in an overall reduced time-frame.

To implement this idea, we introduce pseudo-pixel resolutions, where the simulation is carried out at the lowest available data-based resolution (DS89 in this case), but where many parameters are linked in such a way that they are algorithmically treated as a single parameter. i.e. they are initialised at the same value, and any change applied to one is applied to all the linked parameters. For example, we might construct a pseudo-pixel resolution DS178, which will simulate an image from a spatial model with the same x and y dimensions as DS89. However, groups of 4 adjacent parameters are linked into a single parameter, such that the total number of alterable parameters is $4\times$ smaller than at DS89.

6.3.2 Algebraic Re-sampling

With the idea of pseudo-pixel resolutions laid out in the previous subsection, we consider how this idea could be taken further, to reduce the number of alterable parameters in ways besides grouping them into larger and larger patches. This can make use of information about the structure of the object, obtained from the segmentation, so that the parameter groupings are more conducive to accurate

estimates. This borrows from the concept of algebraic multi-grid, as described in Stüben (2001).

For example, we define an algebraically re-sampled *DSInf*, in which the entire spatial model is reduced down to the fewest parameters possible, which in this case is the following three:

1. A single thickness value assigned to all bone and soft-tissue (as defined by the segmentation) parameters of the model.
2. An alloy value for all parameters classified as soft-tissue by the segmentation.
3. An alloy value for all parameters of the model classified as bone by the segmentation.

At this *DSInf* pseudo-pixel resolution, a single stochastic iteration, updating the single thickness value, applies now to the whole model, and allows us to rapidly check whether that change results in a more accurate model. Making this kind of change could be incredibly time consuming in coarse schemes where large blocks of parameters are grouped together, as the magnitude of the change made to each one is randomly selected at each iteration. If, as an example, the initial estimate of thickness is based off the median thickness value in our simulated database — around 15cm — but the image being resolved is of the fingers, then it will take many successive iterations to gradually reduce the thickness of the object to a reasonable point, and the whole while the algorithm will be fighting against the smoothness regularizer, further reducing the magnitude of the steps that get accepted. With *DSInf*, the same change can be made in relatively few iterations.

Further, in an effort to prevent the possibility of artifacts along the transitions between bone and soft-tissue, we define our other pseudo-pixel resolutions (such as DS178) in a similarly algebraic way. When we first mentioned DS178 at the end of Subsection 6.3.1, we described how it could link square patches of four parameters from the DS89 model. However, this could create issues where those

2×2 patches span one of the aforementioned transitions, and applying the same alloy adjustment to all four parameters would be inappropriate. Thus, we instead define the pseudo-pixel resolutions as follows:

First, we divide the image into n horizontal bands, where $n = y_I(\frac{h}{h^*})$, where h is the down-sample factor at which the model will be simulated, and h^* is the pseudo-pixel down-sample factor we are creating. For example, if an model has x, y dimensions of $[8, 8]$ at DS89, the pseudo-pixel resolution DS178 will first divide that 8×8 model into 4 horizontal bands.

Second, we divide each of those bands according to the segmentation. In the case of the M2AP, where we observe only one, vertically aligned bone, each of the horizontal bands would be divided into exactly three vertical components (left soft-tissue, bone, right soft-tissue). At coarser pseudo-pixel resolutions, the height of these linked patches will decrease, but the width will not.

This assumes that the bones in the anatomy being imaged are vertically aligned. In a case such as where the M2AP is perfectly *horizontally* aligned, this method will suffer, breaking the model down into n bands with width x_I . This is something we choose not to address, the the majority of images in our test set are vertically aligned.

6.4 Translating Spatial Models Between Resolutions

Of course, very coarse resolutions such as DSInf described in 6.3.2 and DS178 described in 6.3.1 will not be capable of accurately resolving a spatial model on their own. While we cannot exclude the possibility that a coarse model will suffice for accurate simulation of the X-ray, we can assume that in most cases a finer resolution better, and indeed this is the target for the final output.

In such a multi-resolution scheme, we need to define techniques for converting the information in the spatial model between these resolutions. The main technical

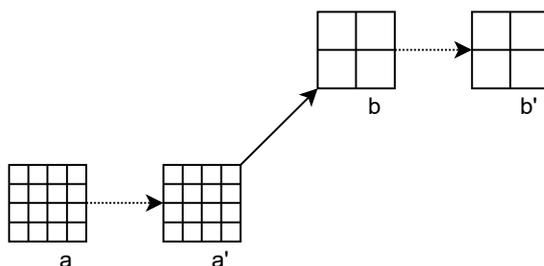


Figure 6.4: Graphical depiction of the states involved in down-sampling (coarsening) a spatial model for the first time. Solid arrows represent resolution transitions, and dotted lines represent optimization iterations.

challenge is that these resolutions do not necessarily align well. As can be seen with a keen eye in Figure 6.3 the down-sample factors selected do not perfectly align with each other, nor do they align well with edges of the object being imaged.

To address this problem, the boundaries of the object and the materials within derived from the segmentation CNN are computed independently at each resolution, by providing the raw X-ray image at that resolution to the CNN to label. The complication with this is that if standard image re-scaling algorithms are used, it is possible that when moving from one resolution to the next, a small area along a boundary that was previously classified as bone could now be classified as soft-tissue, for example.

6.4.1 Coarsening Resolution

A high-resolution estimate of the spatial model contains more information than a coarse one. We nonetheless want to use coarse resolutions for their efficiency, thus the fine-to-coarse transition algorithm is designed with the understanding that when we coarsen we will have to go back to the finer resolution later, to recover lost information. We want to ensure that the information at the fine resolution is not lost when we move into a coarse resolution, so before the coarsening process begins, the fine-resolution state (such as a' in Figure 6.6) is recorded. How this recorded state will be used is explained in Subsection 6.4.2.

During the description of this algorithm, we will refer to the initial, fine resolution as the source resolution, and the output, coarser resolution as the target resolution.

In this coarsening process, we begin by determining the work done by the optimiser so far. The initialization algorithm that produces a raw initial state from the segmentation is re-run for the current resolution, and is subtracted from the current estimate for the spatial model. We then initialise the spatial model again, producing an initial state (as it was prior to any iterations of optimization) based on the segmentation of the image at the new, target resolution.

The source resolution's segmentation is then down-sampled to the target resolution dimensions using nearest-neighbour interpolation. For that we MATLAB's built-in image re-scaler (`imresize`). The down-sampled source segmentation, and the target segmentation are compared, to identify points where the image re-scaling algorithm does not align with the native segmentation. This informs us of the points where re-scaled optimization data will also not align, and have to be re-scaled separately.

Put differently, this means that where a pixel has been classified as soft-tissue at the new resolution, but was classified as bone in the previous resolution, it's value will be the mean of the values of neighbourhood pixels that are classified as soft-tissue. This process is applied to thickness values, and soft-tissue and bone alloy values separately, to reduce artifacts in the re-sampling process.

6.4.2 Refining Resolution

In this subsection, the same nomenclature as in Subsection 6.4.1 will be used, but reversed, so that the coarser resolution is referred to as the source, and the finer resolution as the target.

When refining resolution, there are two different schemes that we consider. First is the case where a new resolution is reached for the first time, in a gradually increasing scheme, such as the one shown in Figure 6.5. In this case, the thickness

and alloy components of the model are up-sampled as plain, grey-scale images using MATLAB's bilinear image resizing algorithm (`imresize`). This will lead to some bleeding around known edges, but in a direction that is physically motivated. In other words, the thickness will reduce at edges, and the proportion of bone will gradually decrease to zero at the edges of bones. Both of these are approximate representations of the organic nature of anatomies that we would expect in most cases.

Second is the case that is shown between b' and c in Figure 6.6, where we refine the resolution after temporary processing at a coarser resolution. Here, we can implement the complimentary half to the algorithm described in Subsection 6.4.1.

In this second case, the recorded state described previously in subsection 6.4.1 comes into consideration. In this situation, instead of calculating a work-done image based on the differences from the initial state, we can calculate a work done based on the state the spatial model was in prior to down-sampling. Referring to Figure 6.6 for labels for the different states, the following steps are taken:

1. b' is subtracted from b to find an array representing work done at this resolution, $\Delta b = b - b'$
2. Δb is separated into thickness, bone and soft-tissue components according the the segmentation, and up-sampled via MATLAB's built-in image re-scaler, with nearest-neighbour interpolation.
3. The up-sampled components are smoothed with a segmentation-guided local average. This process is explained in more detail in 6.4.3.3. In this, each pixel is given the value of the mean of the similarly-classified values in the pixel's 3x3 neighbourhood.
4. The up-sampled components are masked, using the target resolution segmentation, to prevent alterations intended for soft-tissue bleeding into bone regions and vice-versa.

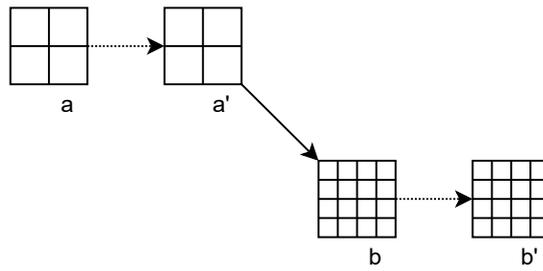


Figure 6.5: Graphical depiction of the states involved in up-sampling a spatial model for the first time. Solid arrows represent resolution transitions, and dotted lines represent optimization iterations.

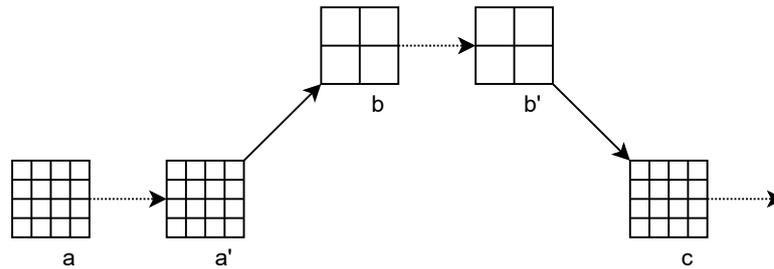


Figure 6.6: Graphical description of a coarsening step followed by a refining one (a V cycle). a , b and c represent the re-scaled states, before any iterations have been carried out at that resolution. a' and b' represent the states at those resolutions after iterations have been performed. Solid arrows represent resolution transitions, and dotted lines represent optimization iterations.

5. The masked, target-resolution alloy and thickness work-done images are applied to the recorded state, a' , to produce c . Iterations at the target resolution can now occur.

6.4.3 Interpolation/Rescaling Methods

When we described the technique for up-sampling a coarse thickness-alloy model, we explained the method that we have opted for after much experimentation. In this section we evaluate various techniques for rescaling a model, assuming we have recorded before and after data, allowing us to infer the work done on a coarse resolution and up-sample it. In this section, we will be comparing the techniques ability to resample the coarse work done images shown below.

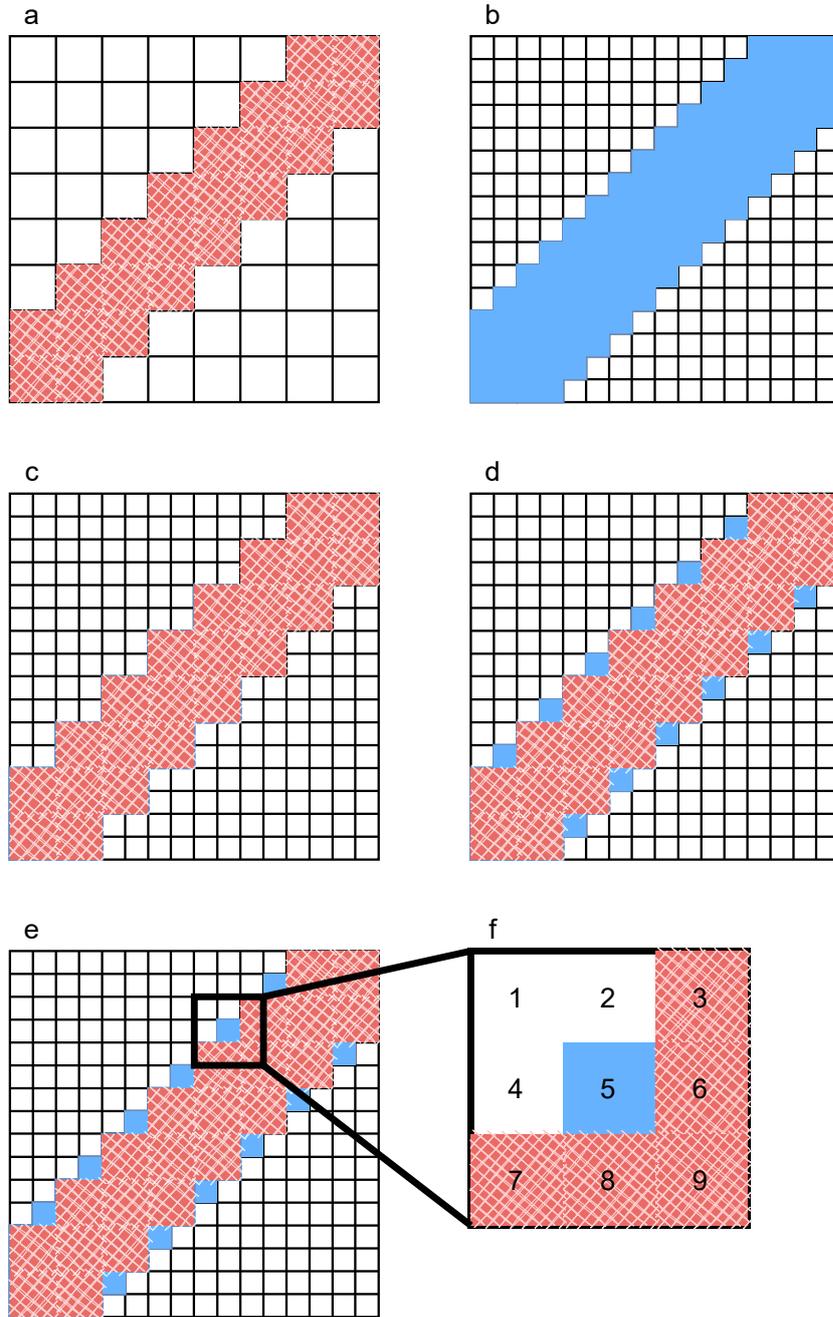


Figure 6.7: Graphic showing the artifact-preserving process for identifying misalignment between the segmentation at source and target resolutions. In this figure, the red, crosshatched squares represent the segmentation at a coarse, source resolution, and the blue, solid squares represent the segmentation at the fine, target resolution. Part (d) shows areas where the up-sampled source segmentation does not align with the target segmentation. As such, pixels like the one expanded upon in (f) will be assigned the value of the mean of adjacent, similarly-classified pixels. In the case of (f), it's final value will be $\frac{(3)+(6)+(7)+(8)+(9)}{5}$

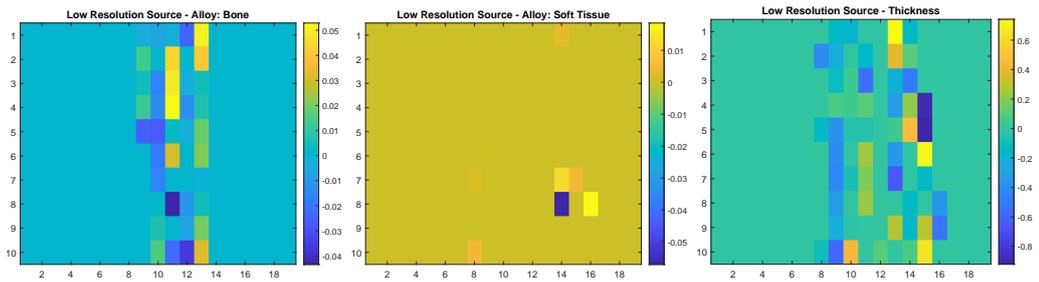


Figure 6.8: Example of an update to thickness and alloy, calculated at a coarse resolution. For this to be applied to the high resolution model, it needs to be accurately up-sampled.

6.4.3.1 Bilinear Interpolation

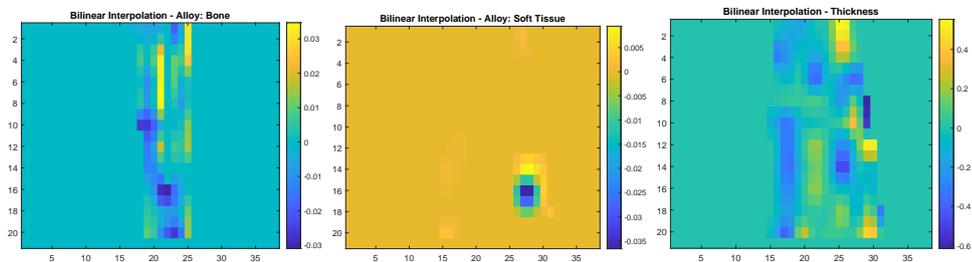


Figure 6.9: Thickness and alloy update from Figure 6.8, re-sampled using a injection followed by segmentation-guided bilinear interpolation.

Bilinear interpolation is carried out by MATLAB’s built-in `imresize` function (MathWorks). In the example here, the coarse model’s alloy data is separated into *bone* and *soft-tissue* components, according to the coarse segmentation. Each of these components, along with the thickness model data, is up-sampled via the `imresize` function. These up-sampled data are then masked with the fine resolution segmentation, to prevent bleeding, and re-combined. In this implementation, bleeding of the surrounding values (zeros, in this case) *into* the meaningful data will occur, but this impact should be minimal.

Bilinear interpolation is also used when increasing the resolution to a given resolution for the first time, when the model is permanently refined. In this case, the model is *not* separated into components, and the thickness and alloy components are simply treated as grey-scale images.

6.4.3.2 Nearest Neighbour / Injection

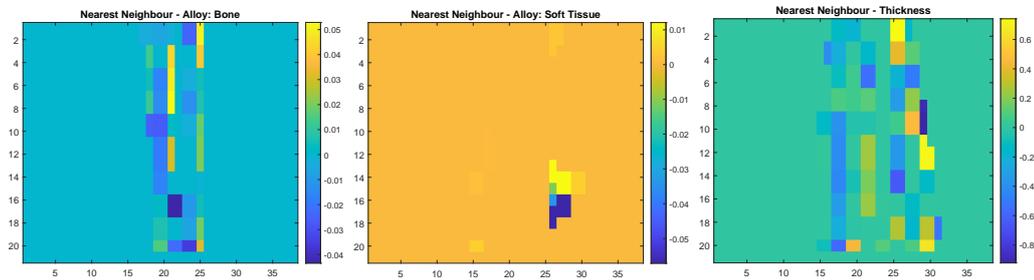


Figure 6.10: Thickness and alloy update from Figure 6.8, re-sampled using injection.

Interpolation by injection is — similarly to the bilinear method in 6.4.3.1 — is principally achieved via MATLAB’s `imresize` function (MathWorks), plus some additional computation. In this case, the `imresize` is applied with the nearest neighbour interpolation option (`nearest`) active. In a case where the coarse and fine grids perfectly aligned, such as one coarse pixel being divided exactly into four fine resolution pixels, this method would result in an exact replica of the coarse model. This thereby sacrifices the additional smoothing regularization that interpolation can offer in return for guaranteed accuracy of the interpolation.

Unfortunately, as stated previously, our grids do not align perfectly. They are only approximately doubled in each dimension, and as such, scaling up with injection will lead to small errors. This is compounded when we consider that the segmentation is distinct at each resolution, and as such errors will occur along edges if we up-sample naively. Thankfully, in this case, the segmentation also provides our solution.

By comparing coarse and fine segmentations, we can identify pixels where standard image rescaling will not accurately predict the shape of the object’s edges. We can then process these identified sources of error separately, so that minimal artefacts are produced. In this case, we identify pixels that could cause artefacts, and set their values to the mean of the non-zero values of similarly classified pixels in the 3×3 neighbourhood of the pixel in question. This seeks to introduce minimal

smoothing, and only in locations where we have no other meaningful data about the value that this pixel should take.

6.4.3.3 Nearest Neighbour with Local Averaging

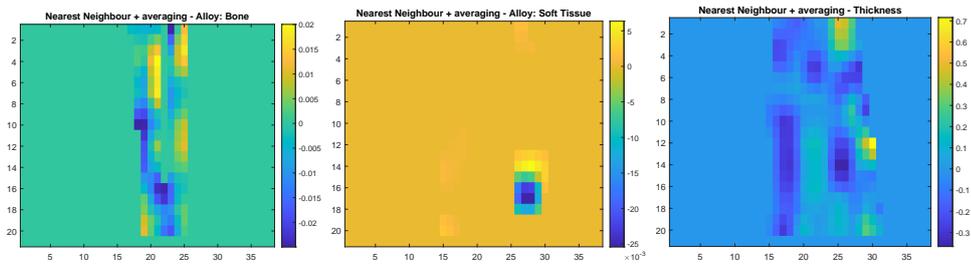


Figure 6.11: Thickness and alloy update from Figure 6.8, re-sampled using an injection followed by segmentation-guided local averaging.

In this method, we try to combine the accuracy of the injection method described in 6.4.3.2 with the regularization that smoothing via 6.4.3.1 provides. This method takes the same, injection-based interpolation that was described in 6.4.3.2, then applies a masked averaging filter to that interpolated estimate, which leads to a final estimate which is smoothed only in the directions that are reasonable, given the information that the segmentation provides.

After the injection-based interpolation has been classified, every pixel within each segmentation-based component of the model (thickness, bone-alloy and soft-tissue-alloy) is assigned the value of the mean of its 3×3 neighbourhood, masked via the segmentation. This will lead to inward blurring of the bone region (for example), but not outward blurring. Changes applied within this region will be smoothed out only within that region, and will not bleed out into the surrounding area. Equally (and more importantly), the zero values that fill the areas outside the region of interest will not bleed inwards, reducing the magnitude of the changes along the edges.

This is distinct from the bilinear method from 6.4.3.1, as the values that are assigned to the pixels in the re-sampled image are *unweighted*, in comparison to

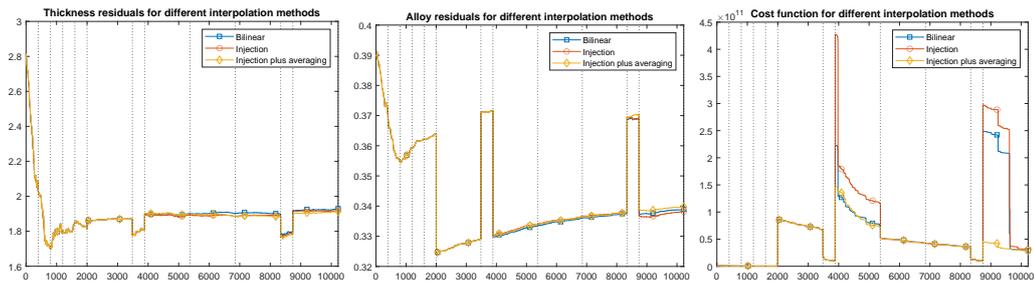


Figure 6.12: Frobenius norm over thickness residuals (left) and alloy residuals (middle), alongside cost function for each interpolation method (right), over the course of many iterations at different resolutions. Dotted vertical lines signify changes in resolutions.

`bilinear`'s *weighted* average.

6.4.3.4 Comparison of Results

Figure 6.12 allows the various interpolation methods to be directly compared against each other, by iteratively coarsening and refining a model, reaching both coarsest and finest resolution twice within a single run of the iterative optimisation algorithm.

The significant takeaway from this figure is that the interpolation methods used to translate coarse data to finer resolutions has only a minor effect on the Frobenius norms over thickness and alloy residuals. This is an expected result; The difference in the values of the models will appear only along edges, and will be a relatively small change. That said, Injection appears to be most effective at reflecting true alloy residuals, while injection plus averaging is most accurate for thickness. These differences are so small however, that it is easy to attribute these successes and failures to noise.

In the cost function however, injection plus averaging is significantly better at preventing sudden spikes in value immediately following coarsening and refinement of the spatial model. This implies that it is more accurate to the expected distribution of values in the model, as high cost function values signify that some error that disagrees with domain knowledge has been introduced. It also removes the

need for the algorithm to work to correct the spikes (which is what occurs for the other two interpolation methods), thereby reducing the total number of iterations needed.

6.5 Paths Through Resolutions

With the resolutions that are viable for use in the algorithm defined, and methods laid out for transforming a spatial model from resolution to resolution, we can now consider the way in which we actually want to make use of these available down-sample factors. We have introduced coarser resolutions than the standard DS23 we worked with in Chapter 5 for the speed of convergence they can provide, as well as the potential they have to improve the solution, by enforcing smoothness constraints on the model (in comparison to updating the model at only the finest-possible resolution). This thereby acts as a regularizer for the model. We start out the algorithm in the coarse resolutions, so that less time is taken to resolve a greater number of the final parameters, as opposed to the time taken if all parameters had to be worked out at the finest-possible resolution. Furthermore, considering the expectations on the smoothness laid out in 5.2.1, we can see that coarse resolutions help to preserve overall smoothness. Small areas of significant noise are impossible to create when only large patches of the image are being altered at any given time. Beyond the idea that we will start coarse, and gradually refine, how do we traverse through the available resolutions?

6.5.1 V-Cycles

V-cycles (Stewart, 2014; Braess, 2007) come from multi-grid research, and formalise the process we have begun to outline in the previous section, where coarse grids are used as a fast, inexpensive intermediary between more expensive, fine grids. As we gradually traverse down layers of resolution, from coarse to fine, we try to

identify situations in which the refinement is not sufficiently improving the quality of our estimation. In these situations, a V-cycle can be carried out. In these V-cycles, the coarsening and refining operators described in subsections 6.4.1 and 6.4.2, respectively, are employed, and the spatial model is translated from the current, fine resolution through each level that is coarser than that, to the coarsest resolution, then refining back to the start point. At each of these resolutions iterations of optimization are carried out, in proportion to the number of effective parameters that can be altered at that resolution.

We have observed that occasionally models will be degraded by overly-coarse refinement, leading to a worse spatial model post-V-cycle. We therefore also introduce a process we call blacklisting, where coarse grids are removed from the set that a V-cycle can access, in response to model degradation. In the event that the algorithm detects significant deterioration of the quality of the simulated image the coarsest, non-blacklisted grid in the set of all grids is made inaccessible, so that further V-cycles will not reach that resolution. If significant degradation occurs more than once, then more than one grid will be blacklisted and made inaccessible. When one or more grids have been blacklisted, a V-cycle will coarsen the model through each non-blacklisted grid coarser than the current, then gradually refine to the grid it began at.

To illustrate this, we can imagine a setup in which the model is intentionally degraded enough to result in blacklisting at every possible opportunity, as is seen in Figure 6.13. This will lead to a sequence of V-cycles, each reaching fewer coarse grids than the previous, and therefore also each containing fewer total iterations. Eventually, only the current grid will remain in the non-blacklisted set, and any subsequent V-cycles will not contain any changes in resolution.

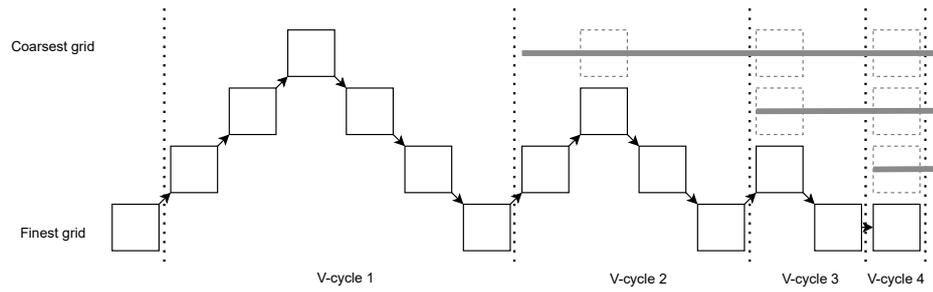


Figure 6.13: Repeated blacklisting and V-cycles, starting from a fine grid. We can see that after each V-cycle, a further coarse grid is blacklisted, reducing the number of grids available in future, and reducing the length of each subsequent V-cycle.

6.5.2 Full Multi-grid

With the concept of the V-cycle established, we use these to produce a full multi-grid (FMG)-like scheme, in which V-cycles and what we call single cycles (where only a single grid is iterated on) are sequentially executed. A hand-crafted and tuned condition set is used to determine the resolution that will be chosen at any given point in this pipeline, by selecting either a V-cycle or a single cycle at the current or next-finest grid. This conditional algorithm seeks to utilise V-cycles as part of a gradual refinement of the spatial model, for optimal speed as well as final accuracy (Kaltenbacher, 2001).

We define four cycle cases, in which either a V-cycle or single cycle will be carried out, leading to coarsening or refining of the model. These cases determine how the algorithm proceeds, in response to the amount by which the simulated X-ray's accuracy has improved during the previous cycle case. These cycle cases, and the improvement factor that determines them, have been found through experimentation. During a cycle, iterations of optimization are performed on the spatial model at one or many resolutions. The number of iterations in a cycle is dependent on the resolutions used by that case.

The four possible cycle cases are listed below, along with a qualitative description of the level of improvement that should trigger that cycle case. Presented in this way, the theoretical justification for the cases becomes clear.

1. Significant improvement has occurred at the current resolution. This informs us that the current resolution is working well, and has not yet converged. An additional cycle will be performed at this resolution.
2. Some improvement has occurred, but it is less significant than the first case. This tells us that complete convergence has not occurred yet, but is likely approaching. We trigger a V-cycle to try to resolve any large patches that could be slowing the progress at a fine resolution.
3. No significant improvement or deterioration. We take this as an indicator that the current resolution has resolved all it can, and refine by one level to begin process at the next-finest resolution.
4. Significant deterioration has occurred. This informs us that continued work on this resolution is no-longer beneficial. In this case, we blacklist the coarsest resolution available in the scheme, and perform a V-cycle. This seeks to avoid any setbacks that could be being introduced in case 2, and reset the algorithm to a minor extent.

These qualitative cases are selected according to a quantitative improvement factor, which is a measure of how much the simulated X-ray image has improved in the previous cycle. It is calculated by dividing the current image residual, $|I - L(\mathbf{t}, \boldsymbol{\mu})|$ by the image residual prior to the most recent cycle. Thus, a smaller improvement factor (< 1) indicates that the residual has decreased, and our estimate is better. A residual of 1 would indicate no change, and a factor > 1 tells us our estimate is becoming less accurate. This quantitative improvement factor γ is defined as follows:

$$\gamma = \frac{|I - L(\mathbf{t}, \boldsymbol{\mu})_C|}{|I - L(\mathbf{t}, \boldsymbol{\mu})_{C-1}|}$$

Where C is an integer denoting the current cycle, and $L(\mathbf{t}, \boldsymbol{\mu})_C$ denotes the spatial model at the end of cycle C .

We experimented with different values for the selection thresholds for each of the four cycle states, and ultimately selected the following for the four states

1. $\gamma < 0.8$
2. $0.8 \leq \gamma < 0.92$
3. $0.92 \leq \gamma < 1.05$
4. $1.05 \leq \gamma$

The values selected here are intended to reflect the qualitative thresholds for improvement described when we introduced our cycle cases. This selection criteria means that V-cycles can be implemented when they are most needed (either when the model has degraded and need to be reset, or when improvement is slowing and could be pushed further by faster alterations on coarser resolutions), and single-cycles can be carried out when they are most needed (When the current resolution is performing well, and when the current resolution has stagnated and needs to be increase).

6.5.3 Results

We have presented our ideas for the techniques that can be used for moving through grids already in this section. Here we will justify our decision to use the full multi-grid described in Subsection 6.5.2, by comparing the model reconstruction capability of FMG against just a single grid and a gradually increasing scheme.

6.5.3.1 Single Grid

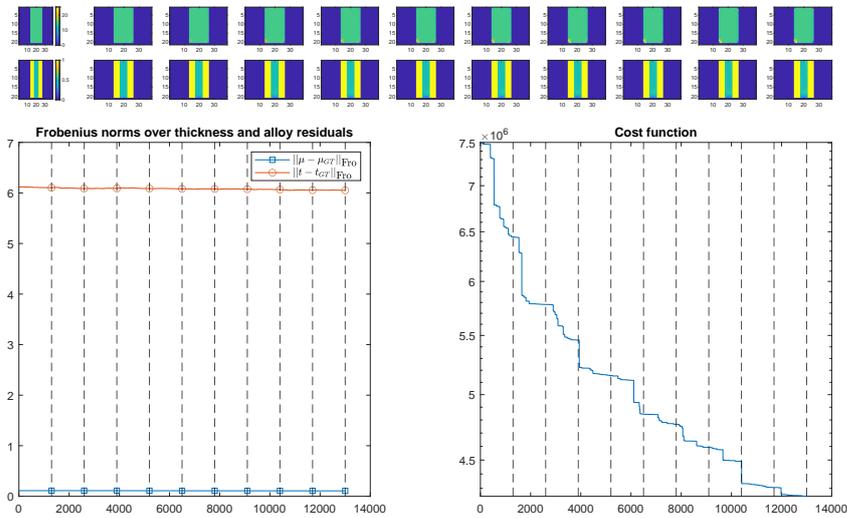


Figure 6.14: Convergence in the thickness and alloy residual norms (left) and cost function J (right), reconstructing the M2AP at just the fine-grid level without grid changes. Only minimal convergence can be seen over the residual norms due to the small impact of fine-grid changes.

First, we show the convergence at just DS45 in Figure 6.14. Unsurprisingly, only a very small amount of convergence is observed in the residual norms over thickness and alloy. When limited to just a fine grid such as this one, the changes that can be applied to the model can only have a small impact on the cost function, particularly if the model is significantly different to the ground truth (such as just after initialization).

Furthermore, the anti-smoothness penalty incurred by alterations is increased at the fine-grid level, as will be covered in Section 6.6. Because of this, alterations that make a substantial change to the initially smooth spatial model are significantly less likely to reduce the overall cost function, and therefore less likely to be accepted as beneficial changes.

Despite this, we do see an ideal convergence trend in the cost function. Only small changes can be applied to the model, but these changes are leading to the optimization that we want.

6.5.3.2 Refining Grids

In Figure 6.15, we present the success of a multi-grid method that only increases onto finer grids. Because we initialise the optimization at the coarsest pseudo-pixel resolution (where all parameters vary in unison), we observe very rapid convergence in both the thickness residual norm and the cost function J over the first ~ 10 iterations. In these iterations, global updates to the initial thickness, soft-tissue alloy and bone alloy are being trialled and accepted where they reduce J . Because the initial state is significantly different to the ground truth, a high proportion of these trialled changes will improve the initially-inaccurate spatial model, and thus be accepted.

At iteration ~ 3100 , we see the model refined onto the finest grid. At this point, we see a sharp increase in the cost function J , and more minor positive and negative changes to the thickness and alloy residual norms, respectively. It is essential to observe here that the sharp change to the cost function that coincides with a resolution change does not correspond to an equivalent deterioration of the quality of the model. Instead, this rapid increase is the result of the increased number of parameters in the model, which the smoothness regularizer is being applied to. Because the model is larger, the total number of contributions from the smoothness regularizer is greater. This should in theory be a predictable increase, proportional to the total number of parameters in the model, but our attempts at normalization have not resulted in a more continuous value.

Instead we emphasise that the selected descent algorithm (described in Section 5.3.1) only permits updates that reduce the cost function, and that the final state of the optimization will always be the finest grid reached during the reconstruction, thus the absolute value of the cost function is not a significant measure of success, but a continually reducing cost function is.

6.5.3.2. Refining Grids

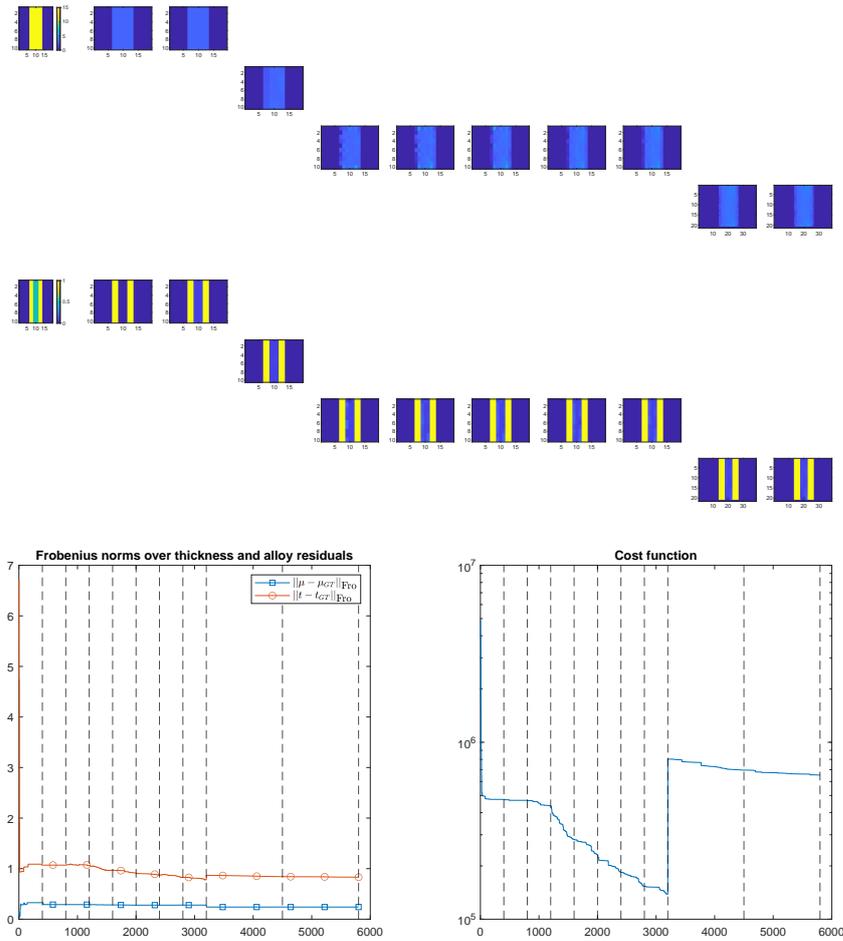


Figure 6.15: Convergence in the thickness and alloy residual norms (left) and cost function J (right), reconstructing the M2AP at multiple grid levels, sequentially refining from coarse to fine. The first few iterations make a sharp reduction in both thickness residuals and cost function. The sharp increase in cost function when refining onto the finest grid (at approximately iteration 3100) comes from an issue normalizing the cost function at different grid levels, and does *not* represent deterioration of the model.

6.5.3.3 Sequential Coarsening and Refinement

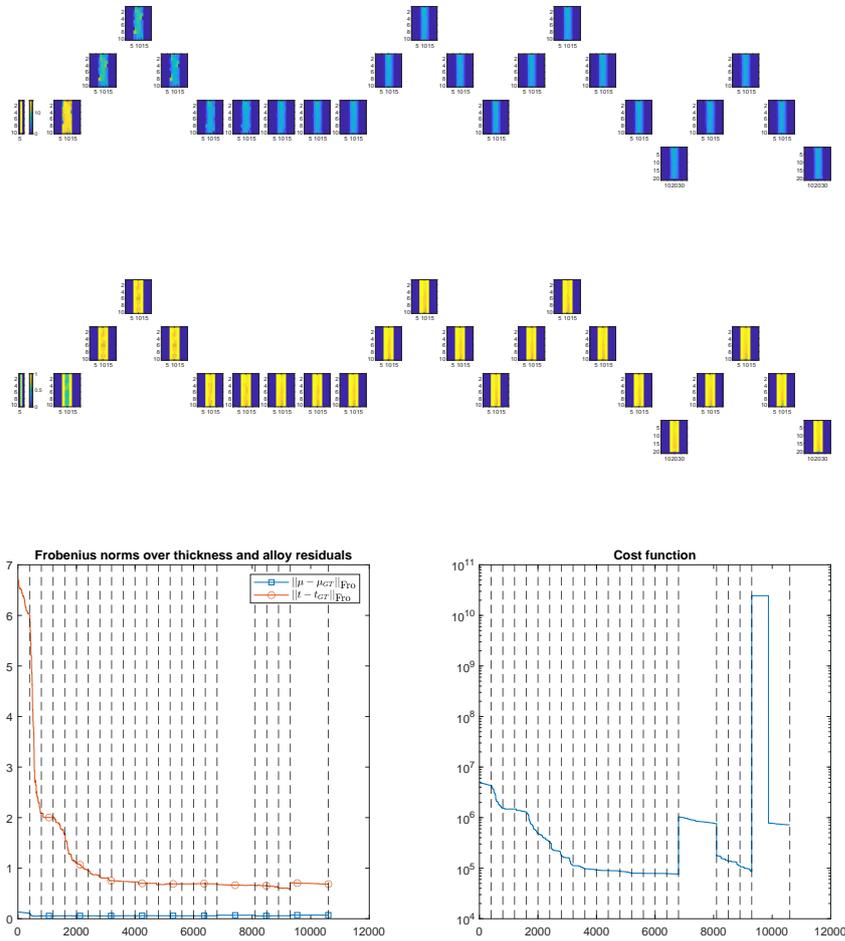


Figure 6.16: Convergence in the thickness and alloy residual norms (left) and cost function J (right), reconstructing the M2AP at multiple grid levels, where refinement and v-cycles of coarsening followed by refinement are implemented in response to relative changes in the cost function. The initial reduction in the thickness and alloy norms is slower than in Figure 6.15, but the convergence over the whole reconstruction run is smoother, and reaches a lower residual norm in both thickness and alloy.

In Figure 6.16 we present an example run of the multi-grid algorithm that we ultimately selected for implementation in our work, described in 6.5.2. This run shows many examples of V-cycles (described in Subsection 6.5.1), where the model is iteratively coarsen and then re-refined, to regularize and escape local minima in the cost function.

Note here that the model is initialised *not* at the coarsest resolution but at the coarsest full-simulated resolution, i.e. the coarsest grid that is not one of the pseudo-pixel resolution described Section 6.3.1. This is to prevent the rapid convergence into particularly common local minima such as the one reached in the first iterations of Figure 6.15.

6.6 Resolution-dependencies in the Regularizers

The regularizers that were defined in Chapter 5 were developed for use at a single, pre-determined resolution. As such, minor alterations need to be made to some of them, to ensure their effectiveness over a range of resolutions.

The smoothness regularizer and the continuity regularizer are both designed to penalise a sharp change in the value of thickness or alloy from one pixel to another. When we produce a model on a single grid the weightings of these regularizers simply need to be tuned appropriately through a process of trial and error. When appropriately tuned in this way, the regularizers neither under- nor over-smooth the model, and a continuous and natural-looking shape is reached as the algorithm's estimate of the model's thickness and alloy parameters. However, when the same model is being processed at many resolutions, the impact on the overall smoothness of the model following a change in the value of a single noisy pixel is variable. The appropriate change can be identified by considering that the negative impact on smoothness of a single parameter being altered by a given amount is inversely proportional to the area of the image that parameter represents. That is to say, a small area that is significantly different to the rest of the model is less smooth than a large significantly different area. Therefore, we make the regularizers' strength proportional to the number of pixels in the object we are attempting to resolve. Phrased differently, the regularizers are less strong on coarser images.

For the segmentation regularizer, the benefit in pushing alloy values towards their predicted state (that is, the soft-tissue/non-soft-tissue state predicted by the seg-

mentation) is equally valid at all scales. If an alloy parameter has been predicted as being 100% soft-tissue by the regularizer, in most cases changing the resolution of the model will not change this prediction. However, we also know that the objects we wish to image are imperfectly aligned with the grids we analyse them on, and that those grids do not align with each other at different resolutions. The edges of the anatomies, and the bones within those anatomies, will not follow the grid lines of our model, and the grid lines at one resolution will not be in the same places at different resolutions. All of this signifies that some alloy parameters will be trying to describe a region of the model that straddles a material boundary. On a coarse grid, a parameter may be trying to describe the alloy of a region that contains some amount of bone alloy and some amount of soft-tissue, and when this model is translated to a finer grid, those two regions become better (but still imperfectly) separated. On that coarse grid, we can estimate that one predicted segmentation-derived state is more accurate than the other, but there will be an inaccuracy associated with this. Therefore, we also know that the accuracy of the segmentation is inherently lower on the coarser grids than on the finer. As such, we make the segmentation regularizer less strong on coarser grids, to reflect the uncertainty of the segmentation at these coarse resolutions. This is implemented by multiplying a constant weighting of the regularizer by the square-root of the number of scattering pixels in the image. We choose this multiplication factor on the grounds that the uncertainty introduced by the imperfect grid alignment is inversely proportional to the side-length of the model, as opposed to the area of the model.

The physicality regularizer, for penalizing values outside the valid range is equally valid at all resolutions, and is unaffected by the grids and their alignment. As such, its weighting is invariant with resolution.

This leads to the following, slightly modified forms of equations 5.2,5.3 and 5.5:

$$\begin{aligned}
 J(\mathbf{t}, \boldsymbol{\mu})_{\text{Smoothed_multi-scale}} &= N\alpha_{D1} \frac{1 - \text{edge}(I_{\mathbf{t}})}{2} \|\Delta \mathbf{t}\|_{fro}^2 \\
 &+ N\alpha_{D2} \frac{1 - \text{edge}(I_{\boldsymbol{\mu}})}{2} \|\Delta \boldsymbol{\mu}\|_{fro}^2
 \end{aligned} \tag{6.1}$$

(described in pseudocode in Appendix A.10),

$$J(\mathbf{t}, \boldsymbol{\mu})_{\text{Segmentation_multi-scale}} = \sqrt{N}\alpha_C \left(\|\text{seg}_1 \odot (1 - \boldsymbol{\mu})\|_{fro} + \|\text{seg}_2 \odot [\boldsymbol{\mu} \geq 1]\|_{fro} \right) \tag{6.2}$$

(described in pseudocode in Appendix A.8),

$$J(\mathbf{t}, \boldsymbol{\mu})_{\text{Continuity_multi-scale}} = N\alpha_E \left(\|\Delta_h \mathbf{t}\|_{fro} + \|\Delta_h \boldsymbol{\mu}\|_{fro} \right) \tag{6.3}$$

(described in pseudocode in Appendix A.2).

Where N denotes the number of scattering pixels in the down-sampled image, i.e. the number of pixels in the down-sampled X-ray that are not open beam.

As the physicality regularizer defined in Eq. 5.4 is not resolution-dependent, it remains un-altered. Thus, the final cost function that is implemented and optimised over to best-reconstruct a thickness-alloy model is

$$\begin{aligned}
 J(\mathbf{t}, \boldsymbol{\mu})_{\text{Regularized}} &= \|I - L(\mathbf{t}, \boldsymbol{\mu})\|_h \\
 &+ J(\mathbf{t}, \boldsymbol{\mu})_{\text{Smoothed_multi-scale}} \\
 &+ J(\mathbf{t}, \boldsymbol{\mu})_{\text{Segmentation_multi-scale}} \\
 &+ J(\mathbf{t}, \boldsymbol{\mu})_{\text{Physicality}} \\
 &+ J(\mathbf{t}, \boldsymbol{\mu})_{\text{Continuity_multi-scale}}
 \end{aligned} \tag{6.4}$$

(implemented as pseudocode in Appendix A.3), or

$$\begin{aligned}
 J(\mathbf{t}, \boldsymbol{\mu})_{\text{Regularized}} = & \|I - L(\mathbf{t}, \boldsymbol{\mu})\|_h & (6.5) \\
 + & N\alpha_{D1} \frac{1 - \text{edge}(I_{\mathbf{t}})}{2} \|\Delta \mathbf{t}\|_{fro}^2 \\
 + & N\alpha_{D2} \frac{1 - \text{edge}(I_{\boldsymbol{\mu}})}{2} \|\Delta \boldsymbol{\mu}\|_{fro}^2 \\
 + & \sqrt{N}\alpha_C \left(\|seg_1 \odot (1 - \boldsymbol{\mu})\|_{fro} \|seg_2 \odot [\boldsymbol{\mu} \geq 1]\|_{fro} \right) \\
 + & \alpha_B \left(\|\min\{\mathbf{t}, 0\} + \max\{0, \mathbf{t} - \mathbf{t}_{\max}\}\|_{fro} \|\min\{\boldsymbol{\mu}, 0\} + \max\{0, \boldsymbol{\mu} - 1\}\|_{fro} \right) \\
 + & N\alpha_E \left(\|\Delta_h \mathbf{t}\|_{fro} + \|\Delta_h \boldsymbol{\mu}\|_{fro} \right).
 \end{aligned}$$

6.7 Conclusion

In this chapter we have described the final element to our algorithm, a method for efficient production of spatial models using many grids. We have described why these additional grids are beneficial, and the two ways that coarser grids can reduce the time needed to resolve an accurate spatial model. We then described how additional grids to those produced by the Geant4 simulation can be constructed, and why this can further reduce the time taken to develop an accurate model.

Following that, we described how spatial models constructed on these grids are translated between them, using custom operators for coarsening and refining. We described multiple methods for doing this that were considered and tested, as well as the final method that was settled upon and why. These operators were then combined into cycles of refining and coarsening, which are used in sequence to produce our spatial models. V-cycles, in particular, and how that form part of a full multi-grid method for our model production were described.

Finally we discussed how the regularizers described in Chapter 5 needed to be altered, to perform well on all grids, instead of only a single one.

Validation on Natural Anatomies

In this chapter we include results demonstrating the effectiveness of the algorithm that we have described in this thesis by testing them on realistic medical X-ray scans. We will compare the performance on a range of different anatomies, and ultimately demonstrate the success of our proposed techniques. The anatomies examined in this chapter comprise of X-ray scans of a full-body phantom, designed to accurately replicate human form using synthetic materials, as well as scans of a cadaver, that were taken by staff at IBEX Innovations.

7.1 Experimental Setup

For all of the models described in this section, the grid was initialised not in its coarsest state, but in the coarsest real grid (i.e. the coarsest grid that is not a pseudo-pixel grid). While it is the coarsest pseudo-pixel grid (*DSInf*) that leads to the fastest convergence, our testing found that initialising the grid at a resolution where the curved shape could begin to be resolved led to the best results. In all cases this was quickly followed by coarsening and the rapid convergence of the pseudo-pixel grids, and thus we see the fastest rate of convergence normally occurs in the second or third cycle of the algorithm.

The weighting factors from equations 6.2, 6.1, 5.4 and 6.3, denoted by α_B , α_C , α_{D1} ,

α_{D2} and α_E are given consistent values throughout the experiments described in this chapter. These values have been selected empirically, and gradually refined to the values tabulated below over many experiments refining the algorithm.

Weighting Factor	Value
α_B	1×10^{12}
α_C	1×10^9
α_{D1}	0.2
α_{D2}	0.2
α_E	1×10^7

All of the experiments discussed in this chapter were left to run for 10 cycles of optimization, with a variable number of iterations per cycle. The batch size for these experiments, which is the number of parameters that are adjusted in a given iteration is 1, as this led to the best results during testing. Furthermore, the parameters of the model are always updated between every evaluation, which corresponds to a minibatch size of 1.

7.1.1 Hardware Details and Execution Time

All of the experiments presented in this chapter were carried out on a Windows 10 machine, with an Intel i7-7700 CPU and 32 gigabytes of RAM. All of the code was written in MATLAB and executed in either MATLAB 2020a or MATLAB 2022b.

The execution time for any given experiment of the ones below was typically in the order of hours, though this was variable, because of the varying number of iterations and x, y dimensions of the images being reconstructed.

7.2 Quantitative Analysis - M2AP

As the M2AP is the only imaged object for which we have ground truth data, it is also the only one for which we can draw meaningful quantitative analysis. The

results displayed below demonstrate that despite being initialised with an initial estimate that is only informed by the result of the segmentation, our algorithm was able to significantly reduce the difference between predicted alloys and thicknesses and their ground truth values.

For clarification, when we say that the initial estimate is only informed by the segmentation, what is meant is that the initial values for thickness and alloy from which the estimate converges are naive guesses based only on the range of values accepted by the simulator, and the prediction of bone and soft-tissue regions from the segmentation network. As thickness values from 0cm to 30cm are accepted by the simulator, we initialise the thickness values at all points in the object at 15cm. For regions predicted to be soft tissue, we initialise the alloys at 1, in line with the expected 100% soft tissue. For regions where the segmentation has predicted a combination of bone and soft-tissue, the alloy values are initialised at 0.5, the mid-point of the valid range.

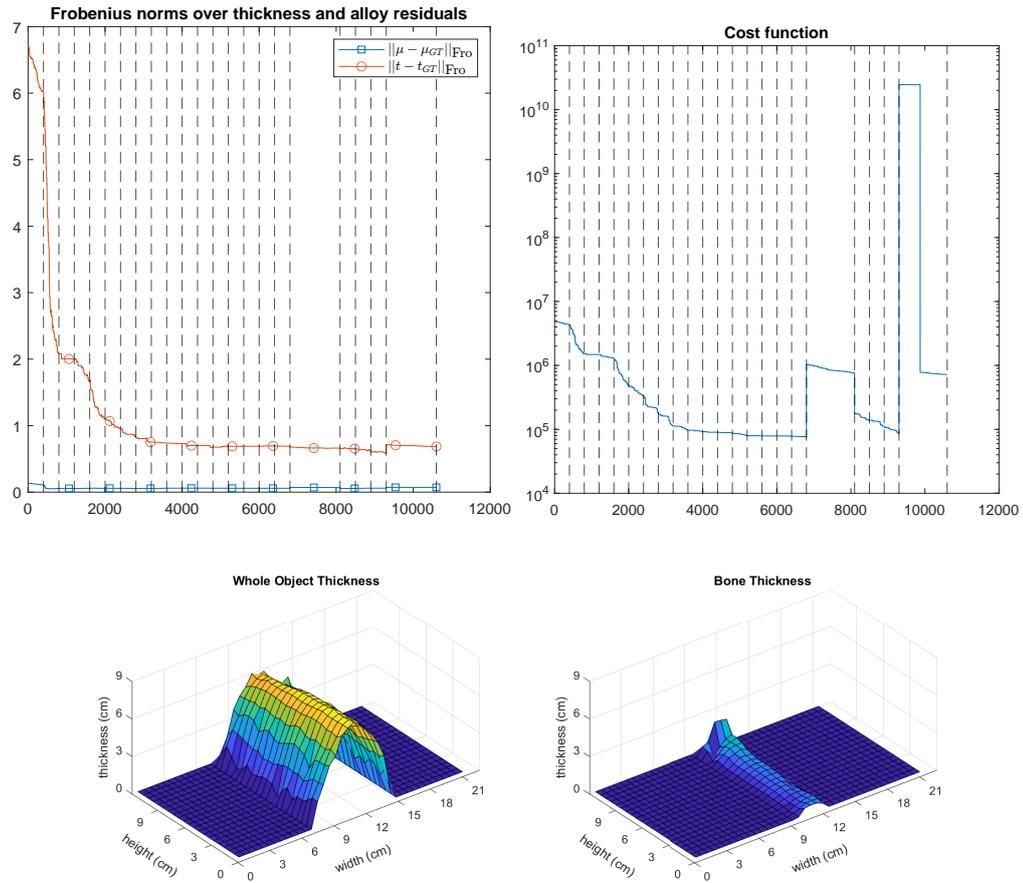


Figure 7.1: Frobenius norm over thickness any alloy residuals against ground-truth data (upper-left), cost value (upper-right), final estimated thickness (lower-left) and bone thickness $((1 - \mu) * \mathbf{t})$ (lower-right) for the M2AP, initialised at 15cm thickness and alloy guided by the segmentation.

Figure 7.1 shows the variation in Frobenius norms of the differences between predicted and ground-truth thicknesses and alloys, as well as the cost function over the course of optimization, and the final reconstructed object thickness and bone thickness. Bone thickness here is calculated by multiplying the bone proportion of the alloy with the total object thickness. I.e. a alloy of 0.5 (50% soft-tissue) multiplied by an object thickness of 10cm implies a bone thickness of 5cm.

From Figure 7.1 we can see that the Frobenius norm over difference between predicted and true object thickness has decreased from an initial Frobenius norm of 6.71 to a final value of 0.69. We find that the average final difference between ground

truth and predicted thickness is 26%, and the average final difference between ground truth alloy and predicted alloy is 25%. The reconstructed alloys have a mean difference of 0.12 in the region segmented as bone, and a mean difference of < 0.01 in the region classified as soft-tissue. The reconstructed thickness has a mean difference from ground truth of 0.998cm.

7.3 Full Body Phantom

In this section we include a series of scans of small sections of the limbs of a full body human X-ray phantom. In contrast to more complex anatomies we will present later in this chapter, these are most similar to the M2AP that we have tested the algorithm against. Here we include the reconstructed surfaces from scans of the upper and lower arms and legs of the full body phantom. Alongside each of these surfaces we present the path through the resolutions that was taken to reach the final surface, the cost function J against number of iterations, and the norms of the residuals between our material reconstructions and the reconstructions produced by IBEX Innovations.

For clarification, we reinforce here that the X-ray scans discussed in this section have come from an artificial model of the human anatomy, known as a phantom, designed for the purpose of testing and training related to X-ray hardware and software. We will only be considering scans of the limbs of this phantom, as we have access to more accurate cadaver data for more complex regions of the anatomy, which will be discussed in Section 7.4.

Furthermore, our decision to consider limb scans first is guided by the testing that have been carried out in previous chapters. Up to now, the algorithm has been tested using the M2AP, an artificial phantom that seeks to simply represent the anatomy of a human upper-arm. The M2AP however is an oversimplification, as it exhibits only very small amounts of variation in thickness and bone density across its surface. This simplicity makes it very easy to gather ground-truth data for.

We therefore opt to begin the validation of the algorithm over regions of artificial anatomy that closely resemble the one used for development, as we can expect reasonable performance on these based on our testing.

7.3.1 Upper Arm

The first set of reconstructions we present come from the phantom's upper arm, as this has the highest degree of similarity with the M2AP model we have tested our algorithm on during development. Because of this, we expect to see good performance on this region of the anatomy.

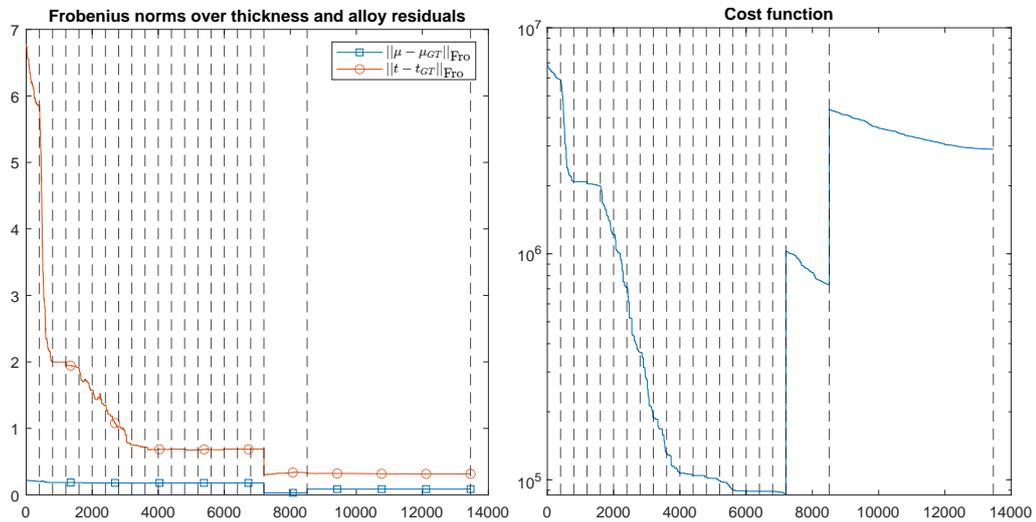


Figure 7.2: Frobenius norms over residuals between reconstructed thickness and alloy values and the reconstructions obtained by IBEX Innovations (left), and the cost function per iteration during the iterative reconstruction process. (right) Vertical dashed lines represent points where grid resolution was able to change.

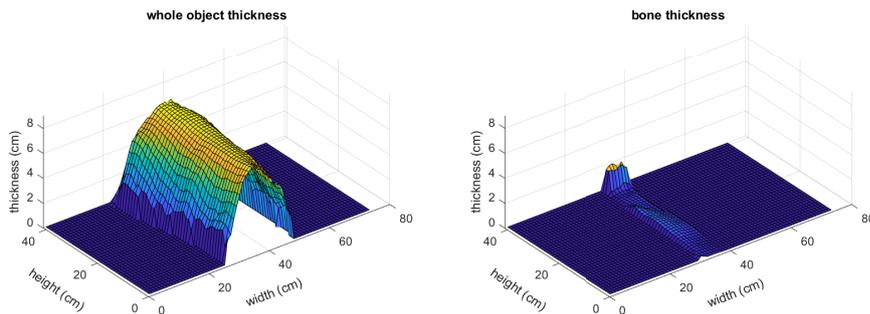


Figure 7.3: Final estimated thickness (left) and bone thickness ($(1 - \mu) * \mathbf{t}$) (right) for the upper arm phantom, initialised at 15cm thickness and alloy guided by the segmentation.

Each of the vertical dashed lines in Figure 7.2 represents one of the points in the iterative process where the conditions for grid change was checked. (I.e. these vertical dashed lines indicate points where the algorithm selected to either remain in the current grid, or change). This does not necessarily mean that the grid did change, only that these were points at which a change *could* change. Notably, in the latter half of the iterations, we can see that the model is being constructed on two grids that are finer than those used in the earlier half. Because these grids are finer, more iterations are required per grid, and thus the spaces between these vertical dashed lines increases.

We also see that the cost function is not consistent during these grid changes. When the model is reconstructed on a finer grid, the cost function corresponding to that model — despite only small changes introduced during that iteration — is significantly greater. Despite efforts to compensate for this through normalisation, this has persisted. It is our assumption that the sharp changes incurred here are due to the changes to the smoothness regularizer when the total number of voxels is altered. The smoothness regularizer (discussed in Subsection 5.2.1) has a additive component for each voxel in our model, thus the contribution changes when the grid size changes, and the model is made up of more/fewer voxels. However, the

smoothness contribution resists normalization because it is also deactivated along the edges. This rate of deactivation — the number of voxels that are not making a contribution to the smoothness penalty — also varies with the grid size, but proportional to the square root of the number of voxels.

Analysing the reconstructed surfaces shown in Figure 7.3, we can see that the overall shape of both object thickness and bone thickness match our expectations for the anatomy. The most notable exception to this is a sharp ridge at one end of the bone, where the bone thickness rapidly increases for a single voxel row. This occurs where the smoothness regularization has been deactivated along the edges of the model/image, and as such these voxels introduce no penalty for being non-smooth. While this introduces an artefact, the artefact is minor compared to the negative impact of enforcing smoothness along the edge of the image. This recurs in other scans (particularly of the cadaver wrist in Subsection 7.4.3), but as the change is relatively minor and localised only to a small region at the periphery of the image, we have determined it does not warrant concern.

7.3.2 Lower Arm

The lower arm region of the same phantom, covering the section between the elbow and wrist, presents a new challenge to our optimiser. Where all previous examples used to test our algorithm have been comprised of a single, approximately cylindrical bone, centred within an approximately cylindrical mass of soft-tissue, the lower arm introduces a far more common feature of real anatomies: multiple bones.

Here we find two bones, each of which is similar to the kind considered in development in previous chapters, and an overall object shape and thickness similar to the upper arm and the M2AP. The reconstructed bone surface $((1 - \mu) * \mathbf{t})$ shown in Figure 7.4 shows that these two bones have been successfully reconstructed separately from each other, but the the model has not managed to accurately capture the cylindrical nature of each bone, instead describing the two bones as getting

thicker towards their combined centre point. This effect is most prominent in the centre of the model, and thus could be caused in-part due to the same effect seen in the overall object thickness.

Additionally, we see that the overall object thickness is noticeably thicker at the centre than at all the edges, creating a overall surface that is closer to a hemisphere than a pure semi-cylinder, as we have seen previously. This is not entirely incorrect — The lower arm does thin along its length, creating an effect that one end is thinner than the other, but the thinning at both ends appears to be an error. This could be caused by the imaging set-up, or could be the effect of a failure of the smoothness regularizer along the edge, as described in the previous section.

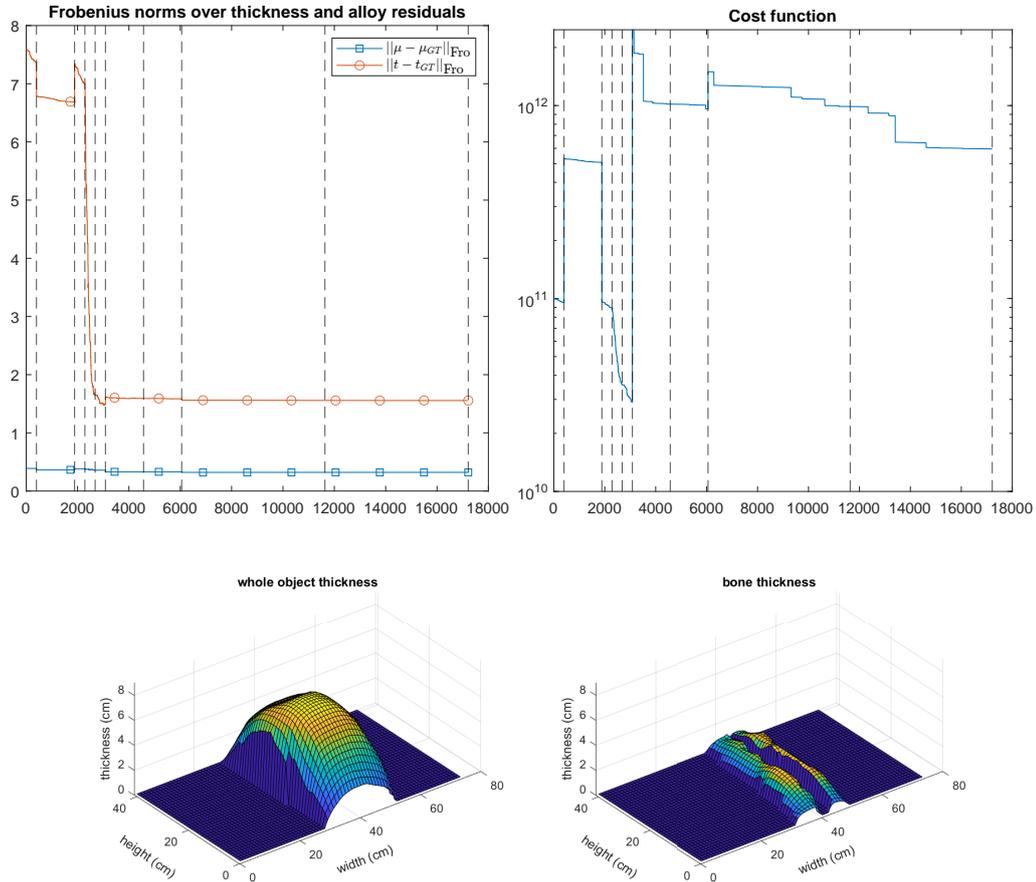


Figure 7.4: Frobenius norm over thickness any alloy residuals against gold-standard data (upper-left), cost value (upper-right), final estimated thickness (lower-left) and bone thickness $((1 - \boldsymbol{\mu}) * \mathbf{t})$ (lower-right) for the lower arm phantom, initialised at 15cm thickness and alloy guided by the segmentation.

7.3.3 Upper Leg

As with the upper arm from Subsection 7.3.1, the upper leg contains only a single bone. The primary challenge presented here is that we are modelling a larger region of the anatomy, after testing on a simple phantom of the arm. The principal challenge for the optimiser here is to reconstruct a surface that is on a different scale to the examples considered previously.

Of the phantom studies, this is where the algorithm's performance is worst. The residual norms shown in Figure 7.5 show that the overall thickness of the reconstructed model is inaccurate by an average of 2cm when compared to the reconstruction from IBEX Innovations. Visually inspecting the model, this makes sense. From our intuition and the cross section of the imaged object, we expect the surface for this part of the anatomy to have a noticeably greater thickness than for the arm regions of the previous two subsections. However, the surface is reconstructed at approximately 8cm, similar to that of the arms. This could present a substantial limitation to the algorithm. For limbs in particular, it would be possible to repeatedly converge to a generic thickness value around the 8cm mark, and be approximately correct for any limb scan. More detailed data to test against (such as CT scans) would prove or disprove this, but scans of more varied anatomies (such as those presented in Section 7.4) can also help us assess this potential limitation.

The graph of the cost function reflects that the algorithm has struggled to make an accurate reconstruction. The grid has iterated through several V-cycles, as it does in the event of stagnating improvement or deterioration.

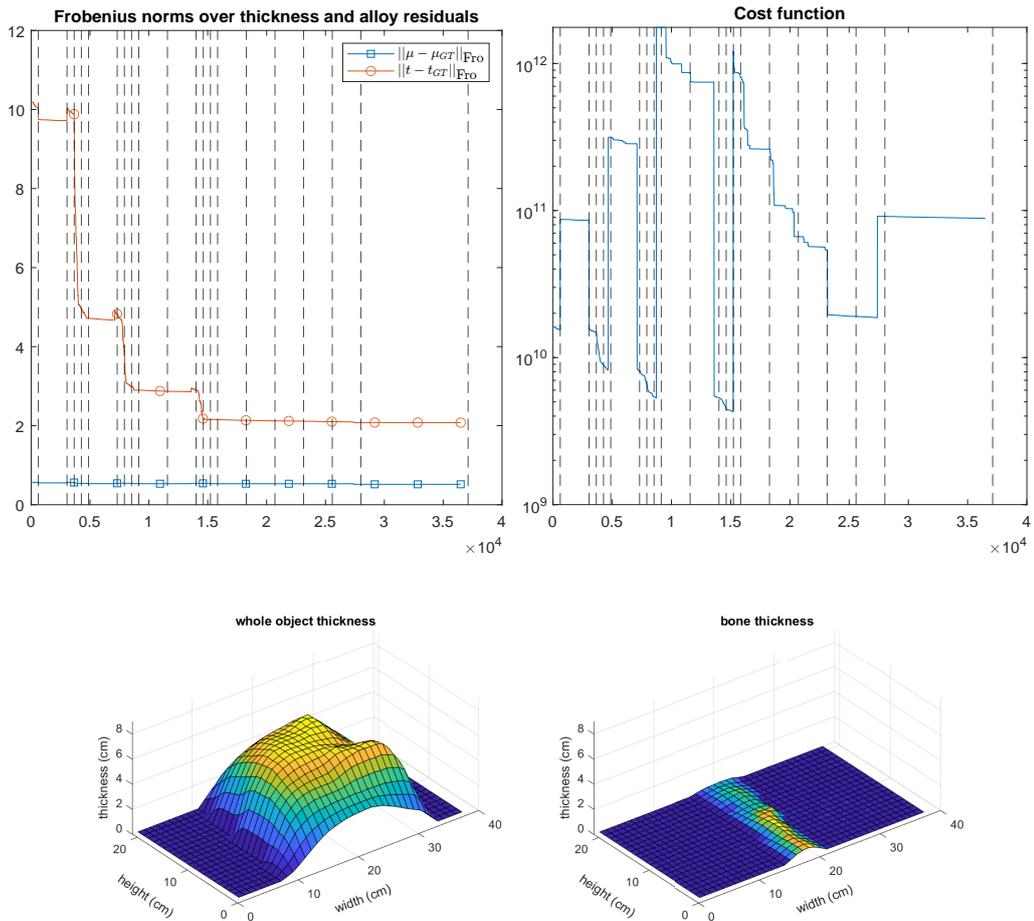


Figure 7.5: Frobenius norm over thickness any alloy residuals against gold-standard data (upper-left), cost value (upper-right), final estimated thickness (lower-left) and bone thickness $((1 - \mu) * t)$ (lower-right) for the upper leg phantom, initialised at 15cm thickness and alloy guided by the segmentation.

7.3.4 Lower Leg

Finally, we present scans of the lower leg region. This could be considered to be the most challenging of the studies in this section, as it presents the challenges of both multiple bones, as in 7.3.2, and the thicker anatomy, as seen in 7.3.3

Despite the intrinsic challenges, Figure 7.6 shows good reconstruction of both the overall shape of the anatomy, as well as the bones within. The object thickness does not decrease at the top and bottom edges, and is thicker along its entire length

than in other tests, as we would expect. This accuracy is reinforced by the norms over the residuals, which are approximately 1cm away from the IBEX estimate, on average.

On the bone reconstruction, we again see that the optimiser has correctly reconstructed one bone as thicker than the other. Furthermore, the thicker bone displays a gradual increase in thickness towards one end, as is typical for this region of the leg.

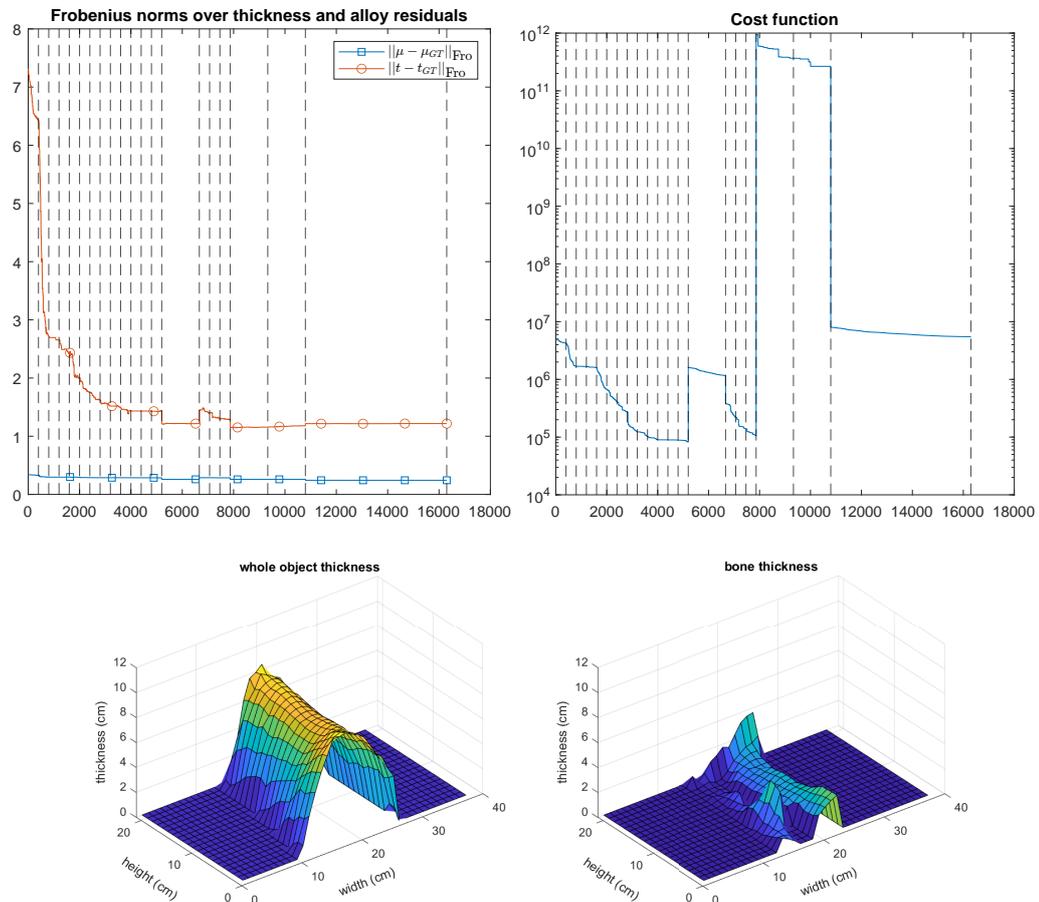


Figure 7.6: Frobenius norm over thickness any alloy residuals against gold-standard data (upper-left), cost value (upper-right), final estimated thickness (lower-left) and bone thickness $((1 - \mu) * t)$ (lower-right) for the lower leg phantom, initialised at 15cm thickness and alloy guided by the segmentation.

7.4 Cadaver

In this section, we present reconstruction of more complex regions of human anatomy, using X-ray scans of a human cadaver. Without access to costly CT scanning technology, we present these results without ground truth or ground truth-like information such as a reconstruction to compare against courtesy of IBEX Innovations. Instead, we seek to evaluate the algorithm’s performance by considering its ability to scatter-correct these scans. Because these are generally scans of regions of the body that induce more scatter than limbs do, successful scatter correction will have a more noticeable effect, compared to the correction that will be achieved on scans such as in Section 7.3.

We also clarify here that the scans in this section do not resemble the simple geometry of the M2AP. These are regions of the body that contain many bones of varying sizes, as well as a range of tissues, sometimes overlapping each other multiple times in the scan. It is this complexity that leads to additional scatter, as well as additional need for its removal.

7.4.1 Skull

The skull is the first part of the anatomy we have attempted to reconstruct which is connected to the rest of the body at only one end. As such, this is the first reconstruction where we would expect the object thickness to taper off to zero on more than just the two parallel sides of the object. Further, this is the first case where we encounter multiple bones in distinct regions in the anatomy, as the top of the cadaver’s collarbones are present in the scan. Finally, the skull itself displays complex changes in thickness across its surface, and is highly different from the cylinders all of the previous test cases could be approximated as. As such, for a complex 3D object such as this, the 2.5D reconstruction produced by our model is going to be less insightful for visually inspecting the quality of the reconstruction,

compared to the limbs presented in the previous section.

Nonetheless, visually inspecting the surfaces in Figure 7.7, we can see that the optimiser has succeeded in some of these reconstructions, and failed in others. Firstly, the overall object thickness looks to be, generally, a successful estimation. The shape of the head tapers off around its edges in the way that we would expect, and the body becomes thicker below the neck as it covers the shoulders and top of the torso.

With regard to the reconstruction of the bone thickness, we can see that the collarbones have been successfully reconstructed as a distinct region of bone from the skull and neck. However, the thickness values estimated for the skull itself does not hold up so well to visual inspection. The reconstruction indicates that the volume of the head is almost entirely filled by the bone of the skull, and while it is appropriate that would indeed fill most of this volume, it is not solid bone. The skull, particularly at the top, is a thin shell of bone, filled with tissues such as the brain. Thus, the bone thickness should be considerably lower than the approx. 10cm estimated. Furthermore, there is no indication in the reconstruction of hollow regions for the eye sockets, which would have been a very encouraging result to see.

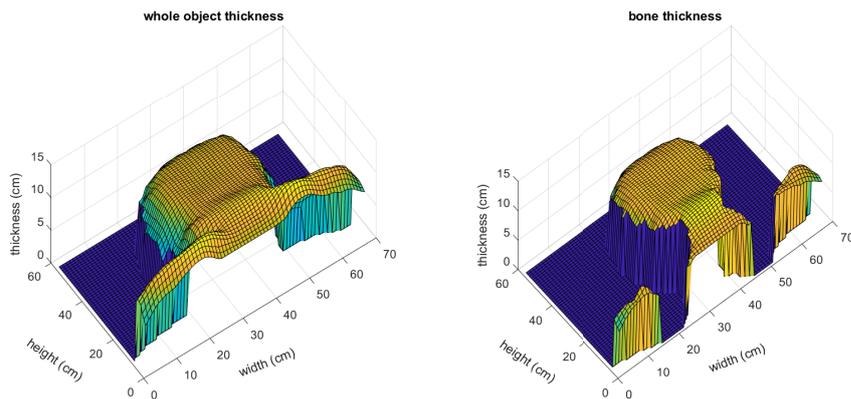


Figure 7.7: Reconstructed surfaces of overall object thickness (left) and bone thickness (right) of the cadaver skull.

7.4.2 Shoulder

Moving on to the visual inspection of the reconstructed surfaces for the scan of the shoulder shown in Figure 7.8, we again have positive and negative aspects to discuss. As with the previous study, the overall object thickness matches our expectations, remaining generally flat with drop-off along the top of the arm, above the collarbone and around the neck. There is a concerning low point towards the centre of the torso, at the corner of the model, which is no doubt incorrect, but should have a minor impact on the scatter correction due to it being at the edge of the composition.

The principal challenge of this example is the number of overlapping bones, such as the ribs, that cross over each other in the X-ray scan composition, in a way that cannot be accounted for by the algorithm in its current state. This limitation is explicitly discussed later, in Subsection 8.1.2. In a perfect reconstruction, we would expect to see several sharp changes in the bone thickness reconstruction, where two bones overlap each other, and the thickness is suddenly doubled. Instead, we impose a penalty on just such behaviour, encouraging the reconstructed surface to be smooth, so as to make the problem at hand solvable, as was discussed in Subsection 4.3.2 of Chapter 4 and Section 5.2 of Chapter 5.

Furthermore, it is in images such as this one that the highly simplified, two-material alloy abstraction of real tissues breaks down. In the chest we expect to find a range of soft-tissue densities, as a result of organs such as the heart and lungs. modelling these as the same soft-tissue analogue in arms and legs is inaccurate, but a necessary simplification for the algorithm we have developed.

Finally, this criss-crossing of bones, combined with the possibility of bones being masked by highly absorbing soft-tissues means that this region is one where we expect that the segmentation is more likely to fail. This further reduces the expected quality of the model in this region

Taking these serious limitations into account, the performance of the bone recon-

struction is surprisingly good. We see a thicker volume of bone in the regions where we would expect the most overlap, along the spine and outermost edge of the ribs. The thickness then decreases into the shoulder and along the upper arm, to approach similar estimates as for the upper arm study in Subsection 7.3.1. Finally, we see the neck curve off on either side, showing the the optimiser has recognised this region is distinct from the collarbones just below, which have a near constant thickness right up to the edge of the scan’s composition.

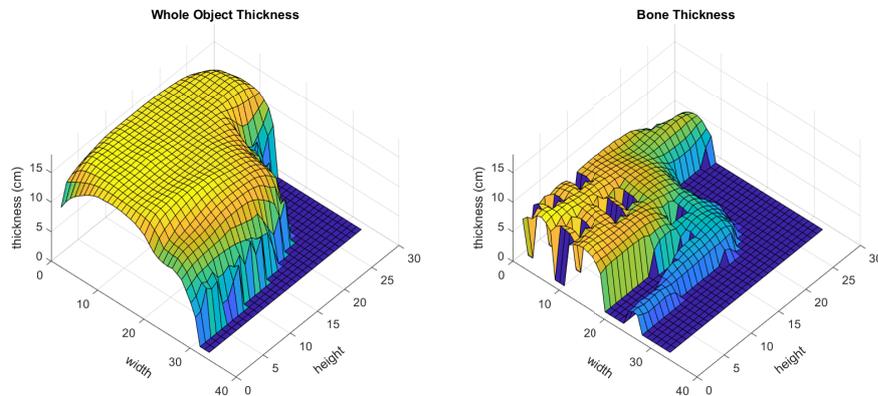


Figure 7.8: Reconstructed surfaces of overall object thickness (left) and bone thickness (right) of the cadaver shoulder.

7.4.3 Wrist

The final cadaver scan we discuss is of the wrist. This composition covers the entire hand, as well as a portion of the lower arm, similar to the regions reconstructed in Subsection 7.3.2. As such, the presents an interesting opportunity for analysis, as part of the reconstruction should match to previously observed results, while the remainder of the scan is completely new to the optimiser. The specific challenge of this reconstruction is the hand, as it is made up of many small bones, most of which are considerably thinner than anything encountered so far.

On immediate inspection, the most obvious issue with the reconstruction shown in Figure 7.9 is the sharp peak in one of the corners of the bone thickness (and, to a less extent, a corresponding peak at the same point in the object thickness.) As

has been observed in other studies, this is likely due to the decision to deactivate the smoothness regularizer along the edges of the model. As was discussed in 5.2.1, the thickness and bone alloy models are penalised so as to encourage smoothness, except along the edges. As such, in the corner there is no smoothness consideration for the value of the model, and we can expect to see sharp peaks more often than along the edges. This is a known issue, but the impact should be minor given that it affects only a single value at the furthest periphery of the image.

One encouraging aspect of this that we do observe is that the voxels around this corner peak have not been smoothed towards incorrect values as well. This is good evidence for why we chose to implement these domain-knowledge as regularizers instead of hard-coding them in such a way that smoothness (for example) must always be enforced. If smoothness had been universally enforced, then the voxels around this one would have been smoothed into incorrect values as well, producing a higher image residuals and a worse model. Instead, in this region, the voxel values are only smoothed if it does not lead to a substantial decrease in model quality, and thus smoothing is not attempted in a case such as this.

Finally, we note that the object thickness and bone thickness both exhibit the expected gradual decrease from elbow to fingertip, taking accurate values lower than in previous test studies. Furthermore, in the object thickness reconstruction, the spaces between the fingers have been identified, with the thickness rising and falling to capture the fingers and spaces between them.

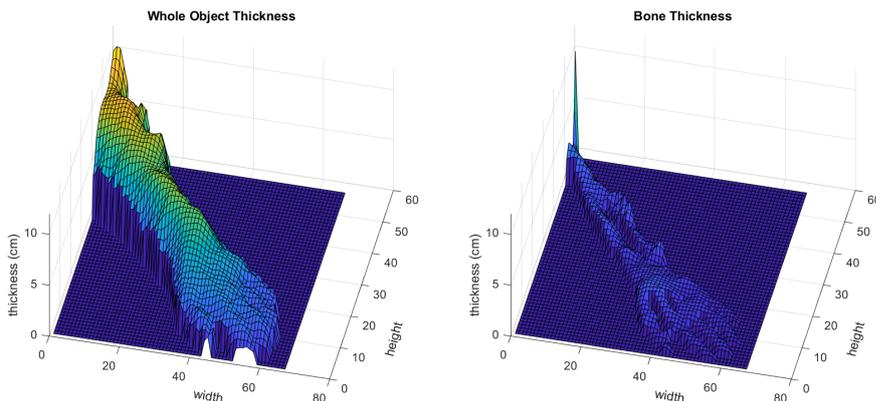


Figure 7.9: Reconstructed surfaces of overall object thickness (left) and bone thickness (right) of the cadaver wrist.

As previously stated, the absence of high-quality ground-truth data limits our ability to quantify the success of our materials estimation algorithm. Access to technical diagrams of the phantoms studied in Section 7.3, or CT measurements corresponding to the X-rays provided by IBEX would have facilitated these quantifications. If given access to such data, the logic next step in this experimental process is to calculate error on our reconstructed models, and compare this against the image residuals. Furthermore, the total number of parameters correctly estimated to within a given margin could be evaluated and used to quantify the success of the reconstruction on each dataset.

7.5 Scatter Correction

To finish this chapter, we present the results of scatter correction using the spatial models presented in the previous sections. These figures seek to demonstrate the benefit that such reconstruction can have on X-ray scan readability. For each anatomical region presented so far in this chapter, we will show a side-by-side comparison of the X-ray scan before and after scatter correction. For each of these scans, the images presented will be contrast-windowed to make the information as readable as possible. This means that the intensity values that pure back and white

correspond to are selected to only span the range with valuable information. This is shown via a corresponding histogram, showing only the range being displayed in the image presented. This point is reinforced in the following subsection, where an unwindowed image is presented for comparison.

7.5.1 Scatter Correction of Phantom Limb Scans

The scatter corrected images from the Full Body Phantom study presented in 7.3 are presented below. The most important thing to highlight here is that for relatively small body parts such as these, we do not expect to see a significant scattering effect, and as such the scatter correction impact may only be minimal. While the changes to the image are not always clear, the histograms are included to show that the effect on the distribution of intensities is significant in all cases. This change is generally to reduce the difference in intensity between the soft tissue region and the bone, such that the two can be seen more clearly in a single contrast window.

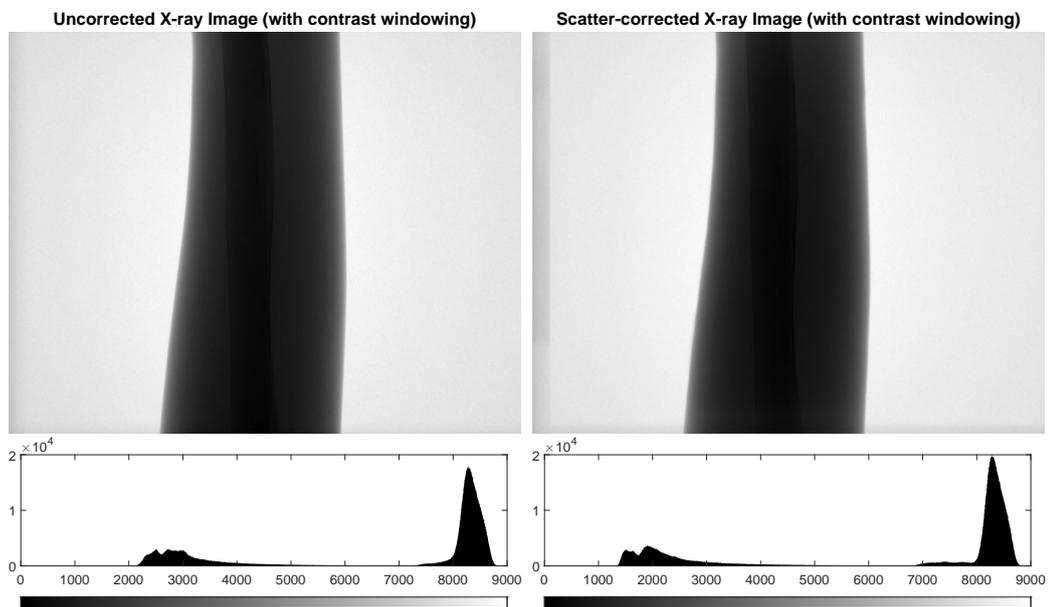


Figure 7.10: Uncorrected (left) and scatter-corrected (right) X-ray scans of the phantom upper arm and corresponding histograms, *without* contrast windowing to improve readability. It is expected that the reader will struggle to discern visible differences between the images, particularly if the thesis is viewed in print. This figure serves to demonstrate the importance of contrast windowing.

Figure 7.10 shows the uncorrected (left) and scatter-corrected (right) scan of the upper arm presented in Subsection 7.3.1, without any contrast windowing. We do not expect the reader to be able to meaningfully distinguish between these two images (particularly if this thesis is being viewed in print). This is partly due to the minimal impact of scatter, as is the case with all results in this subsection, but mostly due to the selected contrast window (or lack thereof) including irrelevant information, such as the details of the open beam region on either side of the phantom, and the collimated region on the left edge of the images.

This collimated region is of interest to us for analysis, as it demonstrates a recurring issue with the existing simulation process. In the uncorrected image on the left, we can see a gradient into a darker shade on the left edge of the image, indicating that a metal plate has been positioned at the edge of the detector area, to reduce unnecessary radiation dose to the patient. This is called collimation, and it occurs frequently in X-ray imaging. The segmentation described in Section 2.3 has identified this collimation as a material that is distinct from bone or soft tissue, and as such will be treated differently by the X-ray simulation process. However, because the collimator actually only occupies a very small area of the full-resolution image, but the segmentation has been created at a reduced resolution, the region that the collimator occupies has been over-estimated. This leads to a significant artifact along the scatter-corrected image's edge, where intensity is reduced because of the (inaccurate) presence of a collimator. In this instance, the artifact has no impact on the quality of the output (see Figure 7.11), as the viewer is not interested in an open-beam region far from the anatomy being imaged, but there are cases where this collimator crosses the object being imaged, leading to such an artifact impacting the quality of the X-ray scan.

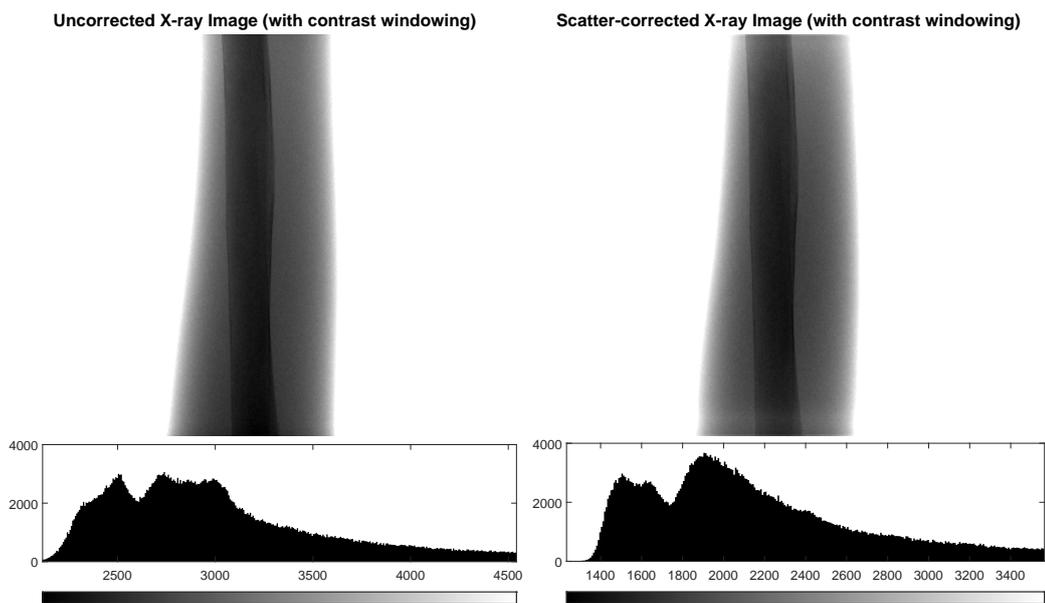


Figure 7.11: Uncorrected (left) and scatter-corrected (right) X-ray scans of the phantom upper arm and corresponding histograms, with contrast windowing to improve readability.

Figure 7.11 shows the same X-ray scan as Figure 7.10, with contrast windowing applied to increase intensity resolution on the regions of interest. As anticipated, the effect of scatter correction is negligible for such a low-scattering region of the anatomy. This is a result we see repeated on all scans in this subsection. Nonetheless, the histograms show that some change to the image has been applied to the image. While it is difficult to draw meaningful conclusions from the histograms, it is encouraging to see that in a simple case such as this one (one narrow, approximately cylindrical region of anatomy, with a single bone within) the histogram's two peaks have become more defined.

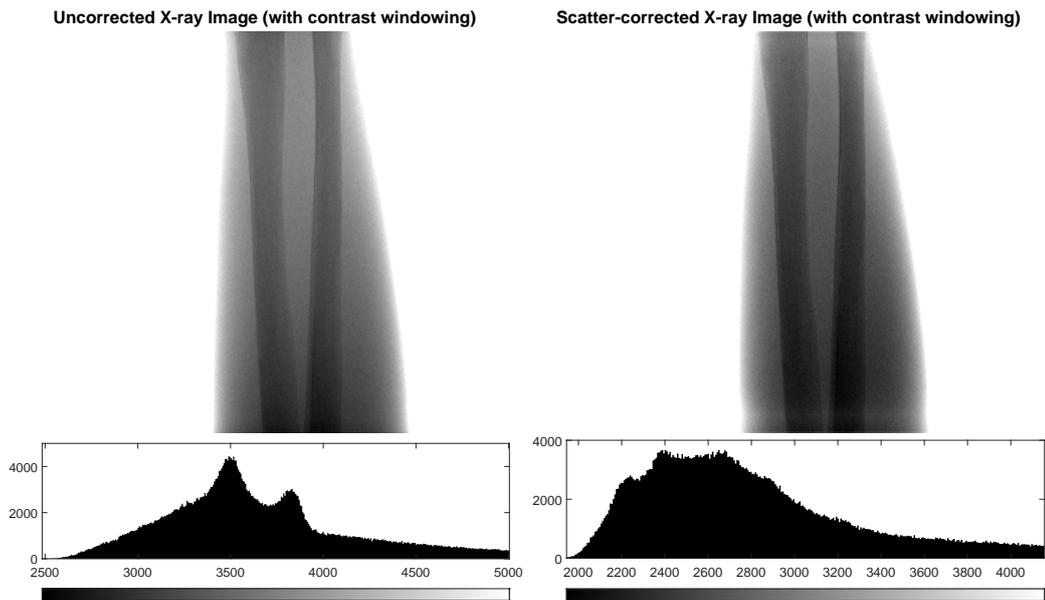


Figure 7.12: Uncorrected (left) and scatter-corrected (right) X-ray scans of the phantom lower arm and corresponding histograms, with contrast windowing to improve readability.

On the scans of the lower arm phantom, contrast between the bones and the tissue around them appears to have increased, but in this instance it is not clear from the histograms whether the two distinct peaks in the uncorrected image have become a single peak in the scatter-corrected because the distance between them has reduced, or just that a greater number of pixels now have an intensity level between these peaks.

This can be investigated further by producing similar histograms after masking the images with the segmentation. Such histograms make clear how the pixels labelled bone and soft-tissue have changed in intensity relative to each other.

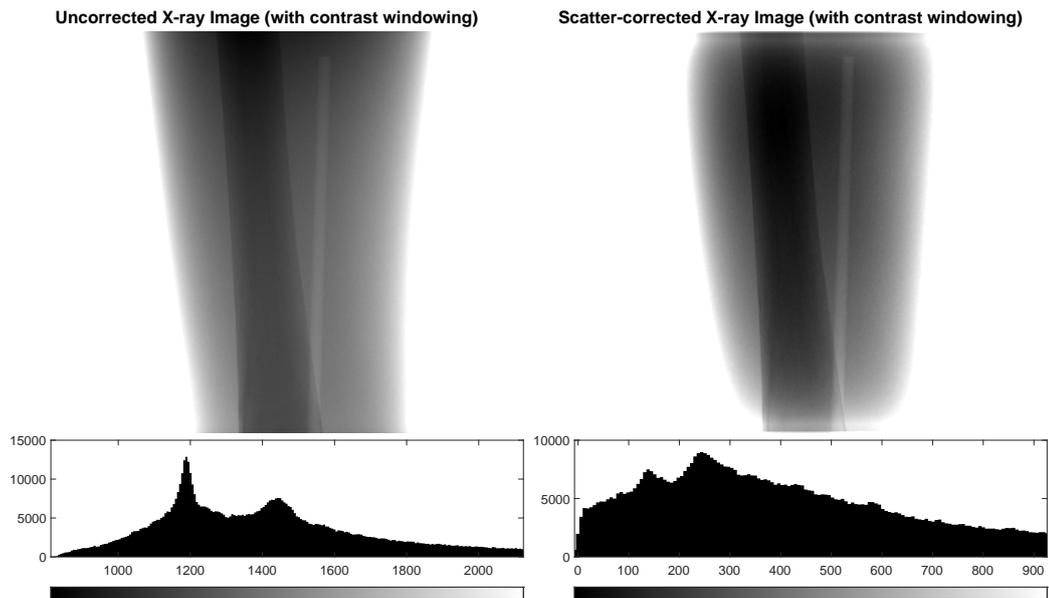


Figure 7.13: Uncorrected (left) and scatter-corrected (right) X-ray scans of the phantom upper leg and corresponding histograms, with contrast windowing to improve readability.

The scans of the upper leg begin to approach a degree of thickness substantial enough for meaningful scatter correction. In Figure 7.13 it can be seen that the scatter removal has increased the range of intensities which the soft-tissue region pixels occupy. This has led to a case where the edges of the object disappear when windowed to most clearly show the difference between the bone and soft tissue (note that the edges are still clear in a differently windowed image.)

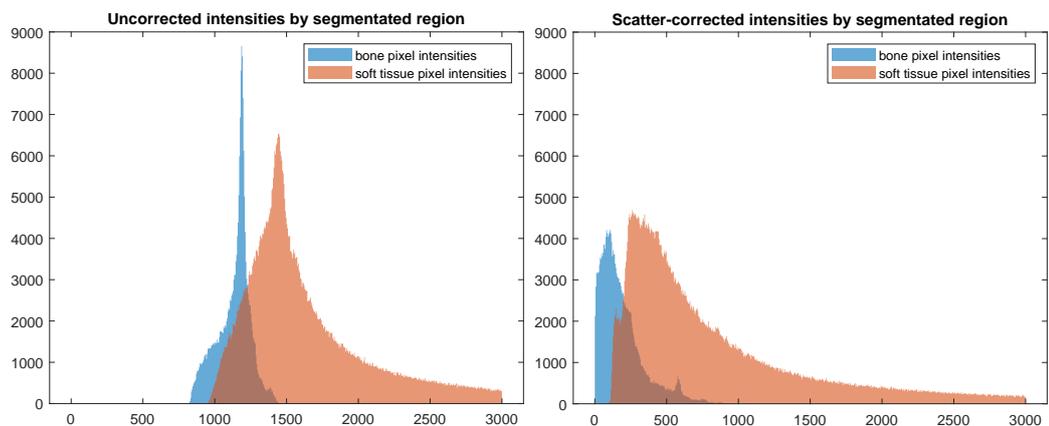


Figure 7.14: Histograms of the uncorrected (left) and scatter-corrected (right) upper leg scan shown in Figure 7.13, separated according to the segmentation of the original image into bone, soft-tissue and open-beam regions.

The histograms in Figure 7.13 show that the two distinct peaks have broadened — in particular the darker, lower-intensity peak corresponding to the bone — suggesting that the intensity range of the bone has increased. This is confirmed by Figure 7.14, which separates the histograms in Figure 7.13 according to the segmentation of the image. Thus we can compare the histograms of pixels classified as bone against those classified as soft-tissue. Figure 7.14 makes it clear that the intensity peaks of the bone and soft-tissue have been broadened and brought closer together by the scatter-correction process, implying to a reduction in contrast between the bone and the soft-tissue. It is also worth noting though that the darkest soft-tissue pixels in the uncorrected image are darker than most of the bone pixels, whereas in the scatter-corrected case, the darkest soft-tissue pixels are no darker than the intensity at which the bone pixels peak. This implies that the two regions should be easier to discern, as it is the darkest soft-tissue pixels which will occupy the region on the border between bone and soft-tissue.

Looking at the uncorrected scan in Figure 7.13 (left), we can visually confirm this. the darkest soft-tissue pixels can be found along the upper edge of the image. In this region, it is harder to pick out the bone than in the same region in the scatter-corrected image (right).

Figure 7.15 highlights a region of interest on the upper leg scans, to give the reader a clearer view of the impact of scatter correction on contrast. Here we focus our contrast windowing on a small region of the images, and control the range of the contrast window to maintain a constant intensity resolution. In the resulting images, visual inspection indicates better contrast between bone and soft tissue, and slightly worse contrast between soft tissue and an unmodelled third material, which could be an internal air gap.

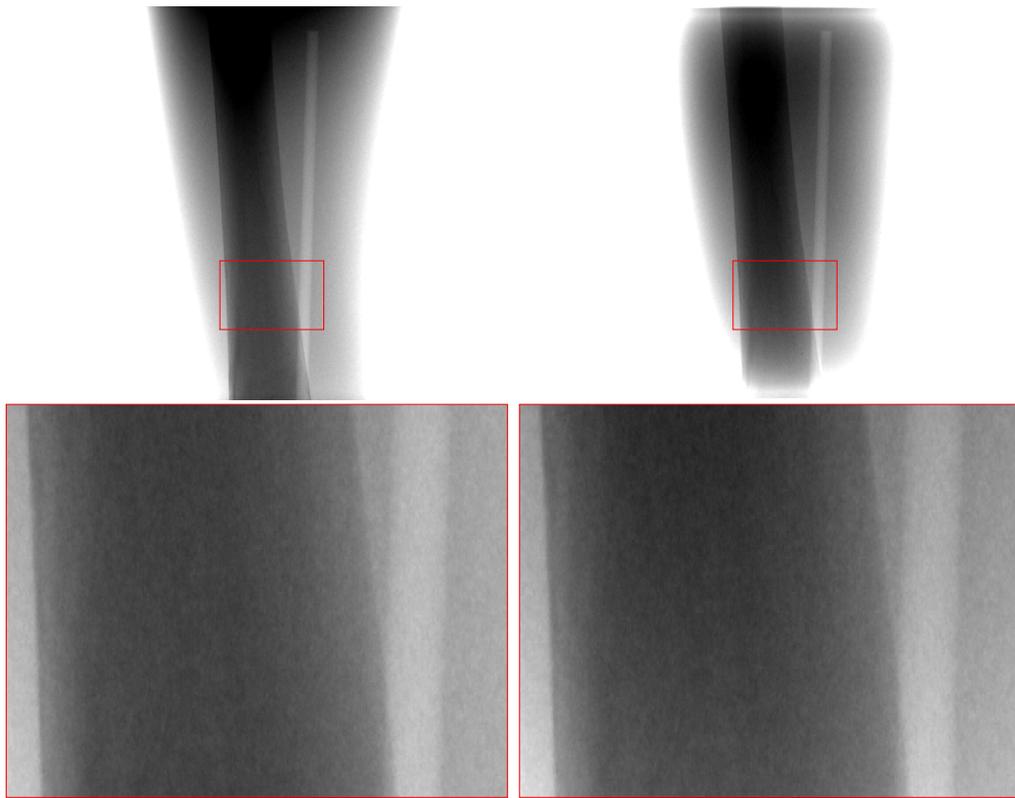


Figure 7.15: Highlighted region of interest on the uncorrected (left) and corrected (right) upper leg scan. In this figure, contrast windows for both uncorrected and corrected scans have been set to have equal range, but centred at different intensities.

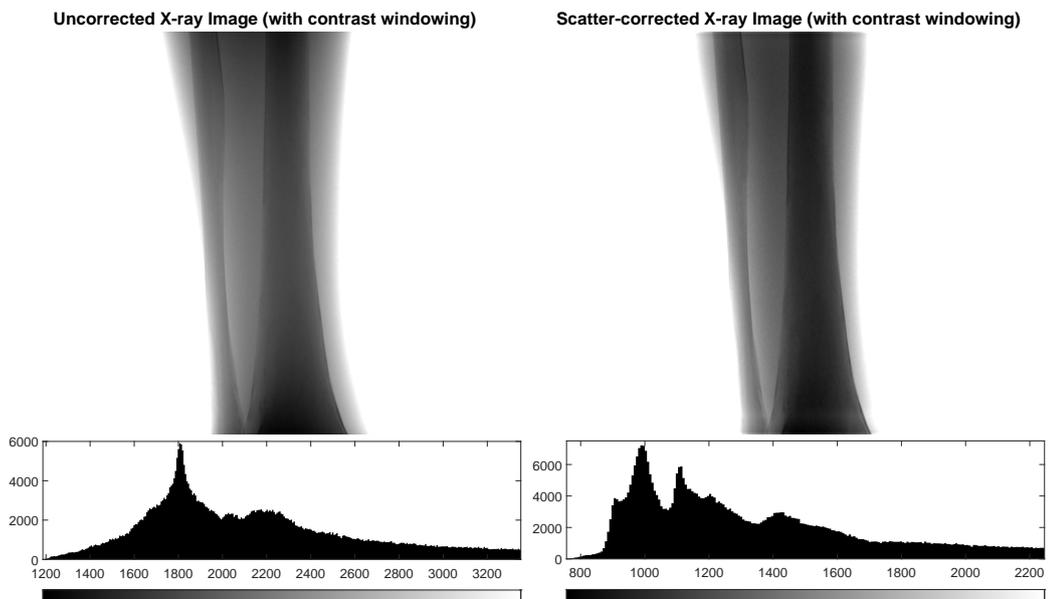


Figure 7.16: Uncorrected (left) and scatter-corrected (right) X-ray scans of the phantom lower leg and corresponding histograms, with contrast windowing to improve readability.

The lower leg scans presented in Figure 7.16 present broadly the same story as the scans already discussed in this subsection. Of note here is that the histogram for the scatter-corrected image present *more* distinct peaks than those in the uncorrected. For an anatomy such as the lower leg, where we see more than one bone present in the image and the bones have very different thicknesses, we could expect each bone to have its own distinct peak.

7.5.2 Scatter Correction of Cadaver Scans

Finally, we present in this subsection the original and scatter-corrected X-ray scans of the cadaver anatomies reconstructed in Section 7.4. In many ways, this can be viewed as a true validation of the algorithm developed. This is the closest to a real-world use-case as is applicable to the work developed, and thus success on these images is a hallmark of a successful anatomy reconstruction and, by extension, scatter correction.

In the images in this section, we cannot expect to see the same clearly defined peaks for bone and soft tissue. This is because the anatomy regions in this section are typically composed of many bones, of a wide range of sizes. Thus, each bone could have a peak that it contributes, which could have a broad intensity spectrum if that bone also varies in thickness within the image.

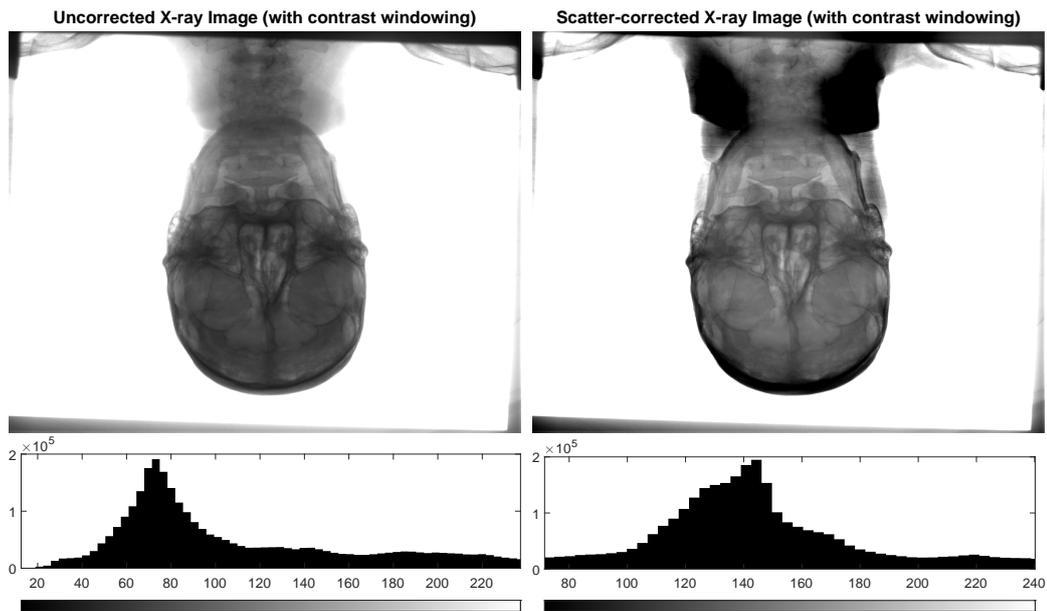


Figure 7.17: Uncorrected (left) and scatter-corrected (right) X-ray scans of the cadaver skull and corresponding histograms, with contrast windowing to improve readability. Note that the histogram binning in this figure and the equivalent histograms in Figures 7.18 and 7.20 is chosen arbitrarily. These images are arrays of float values, and as such there is not a binning that corresponds to the precision of the images. Instead, we opt to use the same histogram binning across all histograms in this chapter.

The cadaver skull X-ray scan, presented in Figure 7.17, is one of the cases where we expect to see a noticeable difference between the corrected and uncorrected images. From the scans presented, it is clear that the intensity resolution of the skull itself has increased significantly, and the structure of the skull can be more clearly discerned in the scatter-corrected image. There is a noticeable difference between the images in the dark region around the neck in the scatter-corrected image, above the skull in the composition. A corresponding dark region can be seen in the uncorrected image, but in the uncorrected scan this neck tissue is only dark *relative* to the other soft-tissue in the scan, which is light enough to be outside the visible intensity range and just rendered as white. In the corrected scan, the soft-tissue have been brought closer in intensity to the range of the bone, such that a more significant portion of the soft tissue is visible in a contrast window optimised for bone.

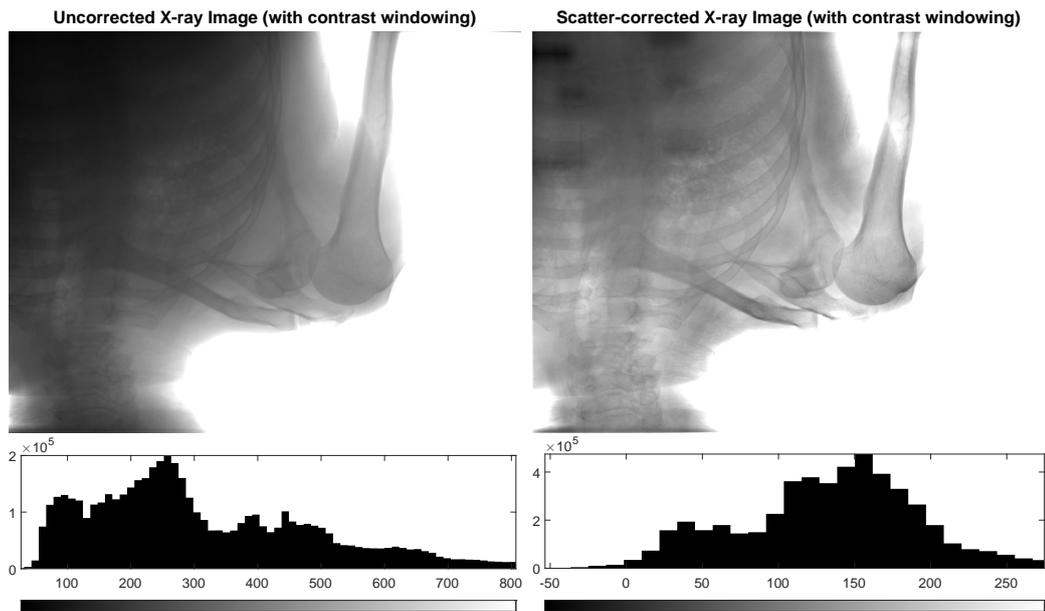


Figure 7.18: Uncorrected (left) and scatter-corrected (right) X-ray scans of the cadaver shoulder and corresponding histograms, with contrast windowing to improve readability. To reiterate from the caption of Figure 7.17, the low precision in the histograms does not indicate low precision in the images.

The shoulder scan presented in Figure 7.18 is the case in which we expect to see the most significant effect of scatter correction. Images such as this, where we are viewing a wide angle of a thick region of the body, contain more scattered radiation due to the greater volume of scattering material. Thus, it is a highly encouraging result to see the substantial impact of scatter correction in the scan on the right in Figure 7.18.

The most significant effect of scatter correction on this scan has been to bring the whole image into a narrower intensity range, such that a single, high intensity-resolution window can be used to see the majority of the torso *and* the arm, without the chest becoming too murky to read. While a second (or even third) contrast window could be used to retrieve comparable detail on the darker regions of the chest in the uncorrected scan, the scatter correction allows for almost all of the anatomy to be viewed clearly in a single contrast window. This is, perhaps, the most meaningful product of the research presented in this thesis. The readability of the image has been improved, in a way that requires less input from the viewer

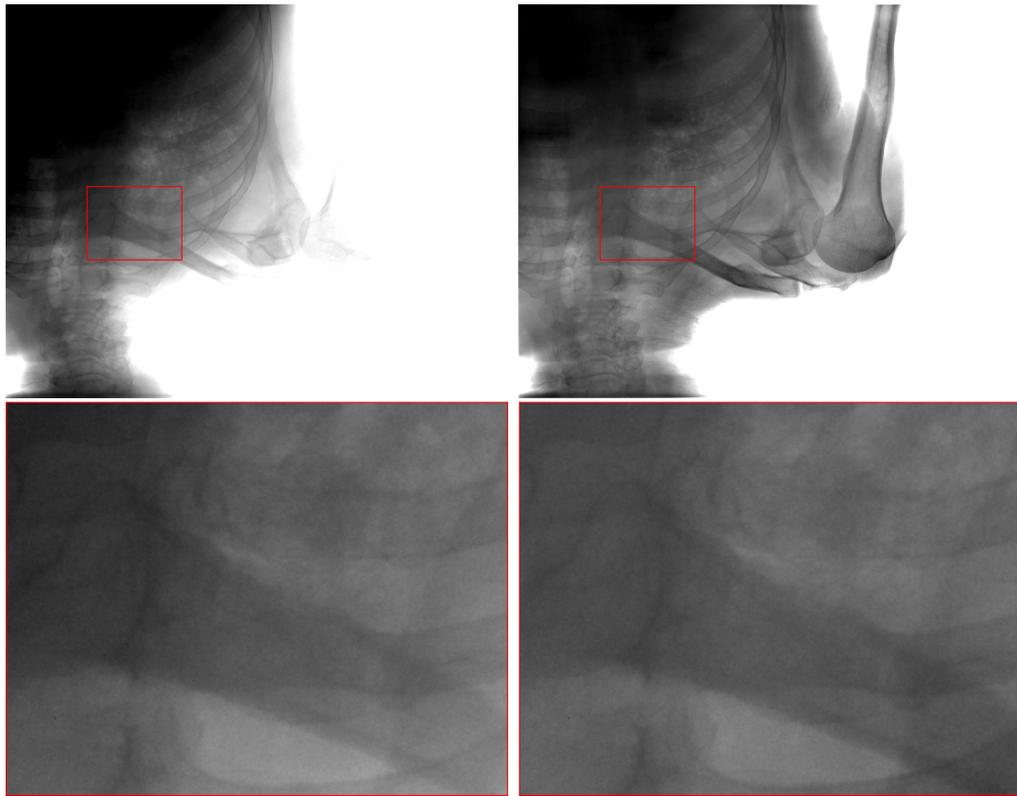


Figure 7.19: Highlighted region of interest on the uncorrected (left) and corrected (right) shoulder scan. In this figure, contrast windows for both uncorrected and corrected scans have been set to have equal range, but centred at different intensities.

to read multiple parts of a single, highly complex anatomy.

This is all despite the fact that the shoulder is also a case which poses known challenges to our material reconstruction algorithm, as was discussed in Subsection 7.4.2. To reiterate here, we know that there will be many bones that overlap in the field of view of the X-ray, in a way that is actively penalised from being represented in the reconstructed model. Furthermore, we know that the composition contains many different organs, and thus many different tissue densities. This makes estimating scatter with a Monte Carlo simulation based on just two materials (bone analogue, and a general soft-tissue analogue) a known over-simplification.

Figure 7.19 highlights the impact of scatter correction on the shoulder by presenting a smaller region of interest at a larger scale.

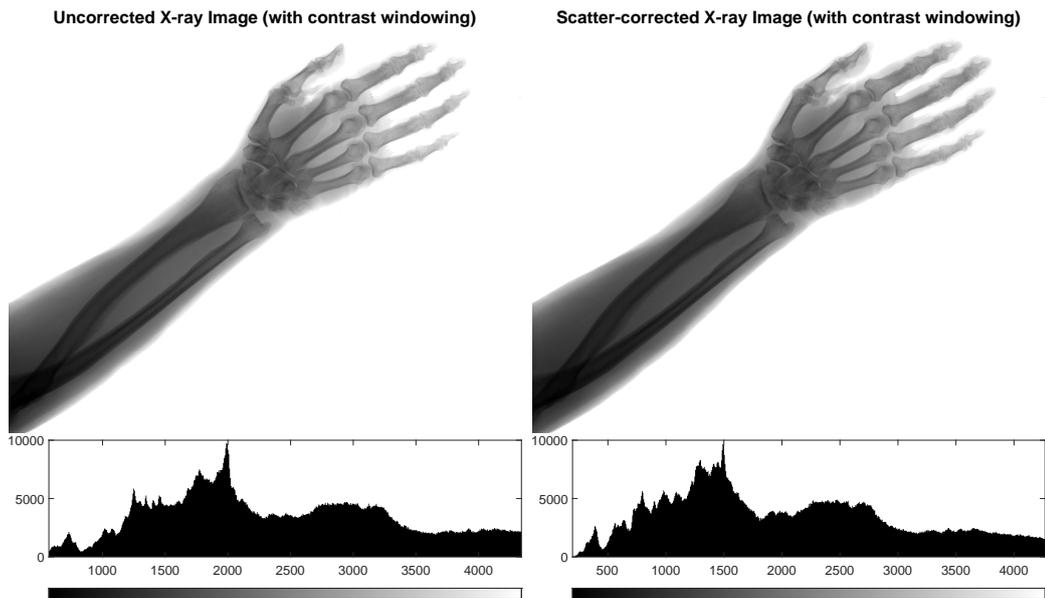


Figure 7.20: Uncorrected (left) and scatter-corrected (right) X-ray scans of the cadaver wrist and corresponding histograms, with contrast windowing to improve readability.

Finally, in Figure 7.20 we present the wrist scan, a case in which we expect to see only slight improvements from scatter correction. What we do observe however, is a similar effect to what was discussed on the shoulder scan, that the whole anatomy has been brought into a narrower intensity spectrum, and thus can all be viewed more clearly in a single contrast window. This is most noticeable at the tips of the fingers, which become too light to be visible in the uncorrected image, but both bone and soft tissue become visible in the scatter-corrected image, without reducing readability of the rest of the object.

Conclusion

In this thesis we have demonstrated the application of multi-grid regularized descent methods to the ill-posed problem of material reconstruction from medical X-ray images. This reconstructed 2.5D model can then be used to produce scatter-corrected X-ray images.

This required us to develop novel, physics-based regularization functions that allow us to constrain the ill-posed problem of single-view material reconstruction. With these regularizers there is sufficient domain knowledge about the problem encoded into the optimiser, that the limited data in a single X-ray image can be used to produce a coarse model of the anatomy.

Recognising the significant computational cost associated with estimating these thickness and alloy parameters at high resolution, we developed a multi-grid method for the reconstruction, such that coarse-grid estimates of thickness, alloy could be quickly produced, then used as a starting point for progressively finer reconstruction processes. We took into account how such multi-grid algorithm could lead to cause the optimization to fall into local minima and create challenging artefacts. To reduce the impact of this, we developed and implemented a grid traversal algorithm that can coarsen and refine the model, in response to the rate of change to our cost function. As the rate of optimization slows and stagnation begins to occur, the grid is able to either refine for better quality, or V-cycle through coarse

grids back to fine, to regularize the model and escape minima with more significant model alterations. This grid traversal allows us to balance increased speed with good accuracy.

Finally, we demonstrated the effectiveness of all of the developed steps on real X-ray images of phantoms and cadavers, and showed meaningful results in terms of both the 3D reconstruction of the anatomies and scatter correction of the native-resolution images.

Reflecting upon the work carried out during the course of this research, the value of the exploratory work described in Chapter 4 cannot be overstated. It was this work that allowed us to make rapid progress towards our goal by simplifying the problem into something easy to understand and rapid to test. Three of the four regularization functions that were part of the final algorithm were developed for this simplified model and scattering process. In hindsight, a different approach to the multi-grid techniques could have proved more valuable in the long run. It was possible that, instead of requiring pre-computed databases for each grid-level that we want to model, we instead only required a single, high-quality set of pre-computed databases, that we then down-sampled at run time. By doing this instead, a wider range of grid sizes could have been simulated and modelled, as well as simulating a greater number of grids within this range. However, this was not possible at the time of development due to the workload increase it would have involved, relative to the additional capability it would have provided.

8.1 Future Work

With all the ideas considered within the scope of this project discussed and evaluated, we take some time here to recommend research directions that could follow this work. These are ideas that were deemed outside the scope of the research due to the time required to implement them and the needs of the industry partner,

but nonetheless can be beneficial for further enhancing the output or reducing the compute-time required.

8.1.1 Better ML Informed Prior

First, we consider here the prior that is derived from the segmentation, from which the algorithm begins optimizing. In the absence of other information, we begin from a state that is perfectly flat, with the bone and soft-tissue regions assigned different alloy values, and the same thickness. This constant thickness has generally been set to the mid-point of the available values (typically $t = 15cm$), and the bone-region has been set to the midpoint of allowed alloys ($\mu = 0.5$). While this has been effective for reducing the number of iterations required to make a good estimate of the model (when compared to a prior that does not make use of the information in the segmentation), we can imagine ways to further improve this. Novel research has shown advances in the field of depth/surface estimation from single images, and thus we can imagine a second neural network which is trained to predict a reasonably accurate prior (i.e. something closer to the ground truth than the one used here), from which the algorithm can converge.

Naturally, the downside to this is that we risk introducing a bias to the final result of the convergence, which we cannot easily justify to end-users, and is based of the trained predictions of the neural network.

8.1.2 Predicting Bone Overlap

In many of the X-ray scans we have tried to model, we encounter regions where two or more bones overlap each other. At these areas, we expect to see sharp discontinuities in alloy, as the thickness of the object is unchanged, but the proportion of bone suddenly doubles or halves. To account for this better, it could be worthwhile to retrain the segmentation network to do one of two things. either:

- Attempt to segment crossover points as an additional class, or
- Attempt to segment each bone independently, such that crossover points would be regions with multiple classification labels

Creating an additional class would allow us to identify the boundaries along which sharp changes in alloy are expected, and reduce the strength of the smoothness regularizer along these boundaries, in much the same way as we do along bone and object edges, see Section 5.2.1. Alternatively, segmenting each bone as its own object would give us a better understanding of how the bones lie above/below each other, and could provide a way to estimate the amount by which the alloy should change along those crossing points. This could then be introduced as a further regularizer to ensure accurate alloy estimation at these bone overlaps.

8.1.3 Additional Material Databases

When resolving models of the most complex regions of the anatomy, such as chest and shoulder images, the limited materials databases begin to be insufficient. Particularly in the human chest, we encounter a range of tissues of different densities, and attempting to abstract these with just generic analogues for bone and soft-tissue reduces the accuracy of the models, and by extension simulated X-ray images our model is able to produce. Introducing new materials has a significant computational cost associated with it, as we now introduce a further dimension to the parameter space the the optimiser must search, but in select cases it is feasible that adding new materials to our database, and limiting when and where they can be used, would be beneficial to the algorithm.

8.2 Final Remarks

The research presented in this thesis seeks to add to the growing body of work around 3D reconstruction and scatter correction for medical X-ray imaging. Most

significantly, we hope to have shown that through careful mathematical formulation of domain knowledge, single-shot reconstruction of ill-posed data is possible. While the reconstruction shown in this research is imperfect — estimating aspects of a 3D model as opposed to a full 3D reconstruction — we have shown how it can facilitate creation of valuable outputs. Further, this reconstruction method stands to benefit from traditional efficiency-increasing techniques in optimisation, such as multi-grid methods.

Bibliography

- J. Adler and O. Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, nov 2017. doi: 10.1088/1361-6420/aa9581. URL <https://doi.org/10.1088/1361-6420/aa9581>.
- S. Agostinelli, J. Allison, K. a. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand, et al. Geant4—a simulation toolkit. *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 506(3):250–303, 2003.
- Z. Alnewaini, E. Langer, P. Schaber, M. David, D. Kretz, V. Steil, and J. Hesser. Real-time, ray casting-based scatter dose estimation for c-arm x-ray system. *Journal of Applied Clinical Medical Physics*, 18(2):144–153, 2017. doi: <https://doi.org/10.1002/acm2.12036>. URL <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/acm2.12036>.
- S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019. doi: 10.1017/S0962492919000059.
- G. Astfalk. *Applications on advanced architecture computers*. Society for Industrial and Applied Mathematics, 1996. ISBN 9780898713688. URL

<https://portal-igpublish-com.ezphost.dur.ac.uk/iglibrary/search/SIAMB0000227.html>.

D. O. Baguer, J. Leuschner, and M. Schmidt. Computed tomography reconstruction using deep image prior and learned reconstruction methods. *Inverse Problems*, 36(9):094004, 2020.

A. B. Bakushinsky, M. Y. Kokurin, and A. Smirnova. *Iterative methods for ill-posed problems*. de Gruyter, 2010.

M. Bertram, J. Wiegert, and G. Rose. Scatter correction for cone-beam computed tomography using simulated object models. In M. J. Flynn and J. Hsieh, editors, *Medical Imaging 2006: Physics of Medical Imaging*, volume 6142, pages 462 – 473. International Society for Optics and Photonics, SPIE, 2006. doi: 10.1117/12.651027. URL <https://doi.org/10.1117/12.651027>.

M. Bilal and M. Arif. Multivariate regression cnn for ill-posed inverse reconstruction of satellite images. *Procedia Computer Science*, 163:312–318, 2019. ISSN 1877-0509. doi: <https://doi.org/10.1016/j.procs.2019.12.113>. URL <https://www.sciencedirect.com/science/article/pii/S1877050919321520>. 16th Learning and Technology Conference 2019 Artificial Intelligence and Machine Learning: Embedding the Intelligence.

J. Birdi, A. Repetti, and Y. Wiaux. A regularized tri-linear approach for optical interferometric imaging. *Monthly Notices of the Royal Astronomical Society*, 468(1):1142–1155, Feb 2017. ISSN 1365-2966. doi: 10.1093/mnras/stx415. URL <http://dx.doi.org/10.1093/mnras/stx415>.

D. Braess. *Multigrid Methods*, page 225–277. Cambridge University Press, 3 edition, 2007. doi: 10.1017/CBO9780511618635.008.

J. Bullock, C. Cuesta-Lázaro, and A. Quera-Bofarull. Xnet: A convolutional neural network (cnn) implementation for medical x-ray image segmentation suitable for small datasets. In *Medical Imaging 2019: Biomedical Applications in Molecular*,

- Structural, and Functional Imaging*, volume 10953, page 109531Z. International Society for Optics and Photonics, 2019.
- P. Bühlmann and S. van de Geer. *Statistics for High-Dimensional Data : Methods, Theory and Applications*. Springer Series in Statistics. Springer Berlin Heidelberg, Berlin, Heidelberg, 1. Aufl. edition, 2011. ISBN 9783642201929.
- J. Cowling, G. Gibson, N. Loxley, P. Scott, P. White, K. Robson, B. Lopez, et al. Novel x-ray detector technology for quantitative material information in digital radiography. European Congress of Radiology-ECR 2016, 2016.
- C. Crockett, D. Hong, I. Y. Chun, and J. A. Fessler. Incorporating handcrafted filters in convolutional analysis operator learning for ill-posed inverse problems. In *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 316–320, 2019. doi: 10.1109/CAMSAP45676.2019.9022669.
- M. J. Darby, D. Barron, and R. Hyland. *Oxford handbook of medical imaging*. OUP Oxford, 2011.
- O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao. Optimal distributed online prediction using mini-batches, 2012.
- A. P. Dhawan. *Medical image analysis*, volume 31. John Wiley & Sons, 2011.
- F. di Franco, A. Sarno, G. Mettivier, A. Hernandez, K. Bliznakova, J. Boone, and P. Russo. Geant4 monte carlo simulations for virtual clinical trials in breast x-ray imaging: Proof of concept. *Physica Medica*, 74:133–142, 2020. ISSN 1120-1797. doi: <https://doi.org/10.1016/j.ejmp.2020.05.007>. URL <https://www.sciencedirect.com/science/article/pii/S112017972030123X>.
- J. Edmonds. *Greedy Algorithms*, page 225–250. Cambridge University Press, 2008. doi: 10.1017/CBO9780511808241.018.

- H. Engl and C. Groetsch. *Inverse and Ill-Posed Problems*. Notes and reports in mathematics in science and engineering. Elsevier Science, 2014. ISBN 9781483272658. URL <https://books.google.co.uk/books?id=6pHOBQAAQBAJ>.
- H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.
- M. Ghassemi, L. Oakden-Rayner, and A. L. Beam. The false hope of current approaches to explainable artificial intelligence in health care. *The Lancet Digital Health*, 3(11):e745–e750, 2021.
- D. Gilton, G. Ongie, and R. Willett. Model adaptation for inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 7:661–674, 2021.
- Y. Gu, J. Zhang, and P. Xue. Multi-grid nonlocal techniques for x-ray scatter correction. In E. D. Angelini and B. A. Landman, editors, *Medical Imaging 2018: Image Processing*, volume 10574, pages 427 – 433. International Society for Optics and Photonics, SPIE, 2018. doi: 10.1117/12.2293347. URL <https://doi.org/10.1117/12.2293347>.
- D. C. Hansen, G. Landry, F. Kamp, M. Li, C. Belka, K. Parodi, and C. Kurz. Scatternet: A convolutional neural network for cone-beam ct intensity correction. *Medical Physics*, 45(11):4916–4926, 2018. doi: 10.1002/mp.13175. URL <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/mp.13175>.
- A. Hauptmann, J. Adler, S. Arridge, and O. Öktem. Multi-scale learned iterative reconstruction. *IEEE transactions on computational imaging*, 6:843–856, 2020.
- P. Henzler, V. Rasche, T. Ropinski, and T. Ritschel. Single-image tomography: 3d volumes from 2d cranial x-rays. In *Computer Graphics Forum*, volume 37, pages 377–388. Wiley Online Library, 2018.
- M. H. Hesamian, W. Jia, X. He, and P. Kennedy. Deep learning techniques for medical image segmentation: achievements and challenges. *Journal of digital imaging*, 32(4):582–596, 2019.

- C. M. Heyer, J. Thüring, S. P. Lemburg, N. Kreddig, M. Hasenbring, M. Dohna, and V. Nicolas. Anxiety of patients undergoing ct imaging—an underestimated problem? *Academic Radiology*, 22(1):105–112, 2015. ISSN 1076-6332. doi: <https://doi.org/10.1016/j.acra.2014.07.014>. URL <https://www.sciencedirect.com/science/article/pii/S1076633214002980>.
- J. Hsieh. *Computed tomography: principles, design, artifacts, and recent advances*, volume 114. SPIE press, 2003.
- J. H. Hubbell and S. M. Seltzer. Tables of x-ray mass attenuation coefficients and mass energy-absorption coefficients 1 keV to 20 MeV for elements Z= 1 to 92 and 48 additional substances of dosimetric interest. Technical report, National Inst. of Standards and Technology-PL, Gaithersburg, MD (United . . . , 1995).
- W. Huda, J. A. Seibert, K. Ogden, E. Gingold, and R. Schaetzing. Scatter removal grids | radiology | suny upstate medical university, 2008. URL <https://www.upstate.edu/radiology/education/rsna/radiography/scattergrid.php>.
- P. Jain and P. Kar. Non-convex optimization for machine learning. *arXiv preprint arXiv:1712.07897*, 2017.
- D. Jha, M. A. Riegler, D. Johansen, P. Halvorsen, and H. D. Johansen. Doubleu-net: A deep convolutional neural network for medical image segmentation. In *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*, pages 558–564. IEEE, 2020.
- S. I. Kabanikhin. *Inverse and ill-posed problems*. de Gruyter, 2011.
- B. Kaltenbacher. On the regularizing properties of a full multigrid method for ill-posed problems. *Inverse Problems*, 17(4):767–788, jul 2001. doi: 10.1088/0266-5611/17/4/313. URL <https://doi.org/10.1088/0266-5611/17/4/313>.
- G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

- Y. Kasten, D. Doktofsky, and I. Kovler. End-to-end convolutional neural network for 3d reconstruction of knee bones from bi-planar x-ray images, 2020. URL <https://arxiv.org/abs/2004.00871>.
- C. A. Kelsey. The physics of radiology, 4th edition edited by h. e. johns and j. r. cunningham. *Medical Physics*, 11(5):731–732, 1984. doi: <https://doi.org/10.1118/1.595545>. URL <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.595545>.
- S. Khirirat, H. R. Feyzmahdavian, and M. Johansson. Mini-batch gradient descent: Faster convergence under data sparsity. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2880–2887, 2017. doi: [10.1109/CDC.2017.8264077](https://doi.org/10.1109/CDC.2017.8264077).
- S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. doi: [10.1126/science.220.4598.671](https://doi.org/10.1126/science.220.4598.671). URL <https://www.science.org/doi/abs/10.1126/science.220.4598.671>.
- H. Latha and R. R. Sahay. Joint estimation of depth map and focus image in sff: an optimization framework by sparsity approach. In *2020 International Conference on Communication and Signal Processing (ICCSP)*, pages 0656–0660. IEEE, 2020.
- M. M. Lavrent'ev, S. M. Zerkal, and O. E. Trofimov. *Computer modelling in tomography and ill-posed problems*. De Gruyter, 2014.
- H. Lee and J. Lee. A deep learning-based scatter correction of simulated x-ray images. *Electronics*, 8(9), 2019. ISSN 2079-9292. doi: [10.3390/electronics8090944](https://doi.org/10.3390/electronics8090944). URL <https://www.mdpi.com/2079-9292/8/9/944>.
- J. Lee and J. Chen. A single scatter model for x-ray ct energy spectrum estimation and polychromatic reconstruction. *IEEE Transactions on Medical Imaging*, 34(6):1403–1413, June 2015. ISSN 0278-0062. doi: [10.1109/TMI.2015.2395438](https://doi.org/10.1109/TMI.2015.2395438).

- A. B. Levy. *The basics of practical optimization*, volume 114. Society for Industrial and Applied Mathematics, 2009. ISBN 9780898716795. URL <https://portal-igpublish-com.ezphost.dur.ac.uk/iglibrary/search/SIAMB0000382.html>.
- H. Li, J. Schwab, S. Antholzer, and M. Haltmeier. Nett: Solving inverse problems with deep neural networks. *Inverse Problems*, 36(6):065005, 2020.
- W. Liao and Q. Ji. Learning bayesian network parameters under incomplete data with domain knowledge. *Pattern Recognition*, 42(11):3046–3056, 2009. ISSN 0031-3203. doi: <https://doi.org/10.1016/j.patcog.2009.04.006>. URL <https://www.sciencedirect.com/science/article/pii/S0031320309001472>.
- H. W. Loh, C. P. Ooi, S. Seoni, P. D. Barua, F. Molinari, and U. R. Acharya. Application of explainable artificial intelligence for healthcare: A systematic review of the last decade (2011–2022). *Computer Methods and Programs in Biomedicine*, page 107161, 2022.
- B. Lopez et al. *A Bayes Linear Approach to Making Inferences from X-rays*. PhD thesis, Durham University, 2018.
- S. Lunz, O. Öktem, and C.-B. Schönlieb. Adversarial regularizers in inverse problems, 2019.
- C. Maaß, E. Meyer, and M. Kachelrieß. Exact dual energy material decomposition from inconsistent rays (mdir). *Medical Physics*, 38(2):691–700, 2011. doi: <https://doi.org/10.1118/1.3533686>. URL <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.3533686>.
- A. K. Maier, C. Syben, B. Stimpel, T. Würfl, M. Hoffmann, F. Schebesch, W. Fu, L. Mill, L. Kling, and S. Christiansen. Learning with known operators reduces maximum error bounds. *Nature machine intelligence*, 1(8):373–380, 2019a.
- J. Maier, Y. Berker, S. Sawall, and M. Kachelrieß. Deep scatter estimation (dse): feasibility of using a deep convolutional neural network for real-time x-ray scatter

- prediction in cone-beam ct. In *Medical imaging 2018: physics of medical imaging*, volume 10573, page 105731L. International Society for Optics and Photonics, 2018.
- J. Maier, E. Eulig, T. Vöth, M. Knaup, J. Kuntz, S. Sawall, and M. Kachelrieß. Real-time scatter estimation for medical ct using the deep scatter estimation: Method and robustness analysis with respect to different anatomies, dose levels, tube voltages, and data truncation. *Medical physics*, 46(1):238–249, 2019b.
- N. Mail, D. J. Moseley, J. H. Siewerdsen, and D. A. Jaffray. The influence of bowtie filtration on cone-beam ct image quality. *Medical Physics*, 36(1):22–32, 2009. doi: 10.1118/1.3017470. URL <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.3017470>.
- MathWorks. imresize documentation. URL <https://uk.mathworks.com/help/matlab/ref/imresize.html>.
- D. Mentrup, S. Jockel, B. Menser, and U. Neitzel. Iterative scatter correction for grid-less bedside chest radiography: performance for a chest phantom. *Radiation protection dosimetry*, 169(1-4):308–312, 2016.
- M. Meyer, W. A. Kalender, and Y. Kyriakou. A fast and pragmatic approach for scatter correction in flat-detector CT using elliptic modeling and iterative optimization. *Physics in Medicine and Biology*, 55(1):99–120, dec 2009. doi: 10.1088/0031-9155/55/1/007. URL <https://doi.org/10.1088/0031-9155/55/1/007>.
- A. Mileto, L. S. Guimaraes, C. H. McCollough, J. G. Fletcher, and L. Yu. State of the art in abdominal ct: the limits of iterative reconstruction algorithms. *Radiology*, 293(3):491–503, 2019.
- D. Minh, H. X. Wang, Y. F. Li, and T. N. Nguyen. Explainable artificial intelligence: a comprehensive review. *Artificial Intelligence Review*, pages 1–66, 2022.

- C. Moore, G. Avery, S. Balcam, L. Needler, A. Swift, A. Beavis, and J. Saunderson. Use of a digitally reconstructed radiograph-based computer simulation for the optimisation of chest radiographic techniques for computed radiography imaging systems. *The British journal of radiology*, 85(1017):e630–e639, 2012.
- G. Nemhauser, A. Rinnooy Kan, and M. Todd. Preface. In *Optimization*, volume 1 of *Handbooks in Operations Research and Management Science*, pages v–ix. Elsevier, 1989. doi: [https://doi.org/10.1016/S0927-0507\(89\)01001-7](https://doi.org/10.1016/S0927-0507(89)01001-7). URL <https://www.sciencedirect.com/science/article/pii/S0927050789010017>.
- H. Nguyen, T. Tran, Y. Wang, and Z. Wang. Three-dimensional shape reconstruction from single-shot speckle image using deep convolutional neural networks. *Optics and Lasers in Engineering*, 143:106639, 2021. ISSN 0143-8166. doi: <https://doi.org/10.1016/j.optlaseng.2021.106639>. URL <https://www.sciencedirect.com/science/article/pii/S0143816621001093>.
- C. Niu, J. Li, and K. Xu. Im2struct: Recovering 3d shape structure from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4521–4529, 2018.
- S. Niu, Y. Gao, Z. Bian, J. Huang, W. Chen, G. Yu, Z. Liang, and J. Ma. Sparse-view x-ray ct reconstruction via total generalized variation regularization. *Physics in Medicine & Biology*, 59(12):2997, 2014.
- A. Norouzi, M. S. M. Rahim, A. Altameem, T. Saba, A. E. Rad, A. Rehman, and M. Uddin. Medical image segmentation methods, algorithms, and applications. *IETE Technical Review*, 31(3):199–213, 2014. doi: 10.1080/02564602.2014.906861. URL <https://doi.org/10.1080/02564602.2014.906861>.
- J. Persliden and G. A. Carlsson. Scatter rejection by air gaps in diagnostic radiology. calculations using a monte carlo collision density method and consideration of molecular interference in coherent scattering. *Physics in Medicine*

- and Biology*, 42(1):155–175, jan 1997. doi: 10.1088/0031-9155/42/1/011. URL <https://doi.org/10.1088%2F0031-9155%2F42%2F1%2F011>.
- M. Piperno, M.-P. H. Le Graverand, T. Conrozier, M. Bochu, P. Mathieu, and E. Vignon. Quantificative evaluation of joint space width in femorotibial osteoarthritis: comparison of three radiographic views. *Osteoarthritis and Cartilage*, 6(4):252–259, 1998. ISSN 1063-4584. doi: <https://doi.org/10.1053/joca.1998.0118>. URL <https://www.sciencedirect.com/science/article/pii/S1063458498901182>.
- A. Prügel-Bennett. *Monte Carlo*, page 45–58. Cambridge University Press, 2020. doi: 10.1017/9781108635349.004.
- R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 2015. ISBN 9781400873173. doi: doi:10.1515/9781400873173. URL <https://doi.org/10.1515/9781400873173>.
- W. C. Röntgen. *On a New Kind of Rays*. Phys.-med. Gesellschaft, 1895.
- M. Sayed, J. Gibson, J. Watson, V. Prisacariu, M. Firman, and C. Godard. Simple-recon: 3d reconstruction without 3d convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.
- O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen. Variational methods in imaging. 2009.
- P. Scott and E. Krastev. 2d x-ray inspection with materials and thickness identification. *SMT Surface Mount Technology Magazine*, 32(6):70–79, 2017.
- S. Shalev-Shwartz and S. Ben-David. *Stochastic Gradient Descent*, page 150–166. Cambridge University Press, 2014. doi: 10.1017/CBO9781107298019.015.
- L. Shen, W. Zhao, and L. Xing. Patient-specific reconstruction of volumetric computed tomography images from a single projection view via deep learning. *Nature biomedical engineering*, 3(11):880–888, 2019.

- P. M. Silva and J. B. Florindo. Using down-sampling for multiscale analysis of texture images. *Pattern Recognition Letters*, 125:411–417, 2019.
- A. Sisniega, W. Zbijewski, A. Badal, I. S. Kyprianou, J. W. Stayman, J. J. Vaquero, and J. H. Siewerdsen. Monte carlo study of the effects of system geometry and antiscatter grids on cone-beam ct scatter distributions. *Medical Physics*, 40(5):051915, 2013. doi: 10.1118/1.4801895. URL <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.4801895>.
- W. E. Snyder and H. Qi. *Mathematical morphology*, page 144–180. Cambridge University Press, 2004. doi: 10.1017/CBO9781139168229.008.
- R. Speller. Tissue analysis using x-ray scattering. *X-Ray Spectrometry*, 28(4):224–250, 1999.
- U. Stankovic, L. S. Ploeger, M. van Herk, and J.-J. Sonke. Optimal combination of anti-scatter grids and software correction for cbct imaging. *Medical Physics*, 44(9):4437–4451, 2017. doi: 10.1002/mp.12385. URL <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/mp.12385>.
- J. M. Stewart. *Case study: multigrid*, page 184–204. Cambridge University Press, 2014. doi: 10.1017/CBO9781107447875.010.
- K. Stüben. An introduction to algebraic multigrid. *Multigrid*, pages 413–532, 2001.
- Z. Tan, J. Li, H. Tao, S. Li, and Y. Hu. Xctnet: Reconstruction network of volumetric images from a single x-ray image. *Computerized Medical Imaging and Graphics*, 98:102067, 2022. ISSN 0895-6111. doi: <https://doi.org/10.1016/j.compmedimag.2022.102067>. URL <https://www.sciencedirect.com/science/article/pii/S0895611122000404>.
- H. Targett, D. Hutchinson, R. Hartley, R. McWilliam, B. Lopez, B. Crone, and S. Bonner. Enhanced visualization of mobile chest x-ray images in the intensive care setting using software scatter correction. *Acta Radiologica*, 2022.

- P. Teodorescu, N.-D. Stanescu, and N. Pandrea. *Numerical analysis with applications in mechanics and engineering*. John Wiley & Sons, 2013.
- U. Trottenberg. *Multigrid*. Academic Press, San Diego, Calif. ; London, 2001. ISBN 012701070x.
- D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018.
- F. Verhaegen, A. E. Nahum, S. V. de Putte, and Y. Namito. Monte carlo modelling of radiotherapy kV x-ray units. *Physics in Medicine and Biology*, 44(7):1767–1789, jan 1999. doi: 10.1088/0031-9155/44/7/315. URL <https://doi.org/10.1088/0031-9155/44/7/315>.
- B. Wall and D. Hart. Revised radiation doses for typical x-ray examinations. report on a recent review of doses to patients from medical x-ray examinations in the uk by nrpb. national radiological protection board. *The British journal of radiology*, 70(833):437–439, 1997.
- G. Wang, W. Li, M. A. Zuluaga, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin, et al. Interactive medical image segmentation using deep learning with image-specific fine tuning. *IEEE transactions on medical imaging*, 37(7):1562–1573, 2018.
- Y. Watanabe and C. Constantinou. *Phantom Materials in Radiology*. American Cancer Society, 2006. ISBN 9780471732877. doi: <https://doi.org/10.1002/0471732877.emd201>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/0471732877.emd201>.
- T. Würfl, M. Hoffmann, V. Christlein, K. Breininger, Y. Huang, M. Unberath, and A. K. Maier. Deep learning computed tomography: Learning projection-domain weights from image domain in limited angle problems. *IEEE transactions on Medical Imaging*, 37(6):1454–1463, 2018.

- M. Xian, J. Huang, Y. Zhang, and X. Tang. Multiple-domain knowledge based mrf model for tumor segmentation in breast ultrasound images. In *2012 19th IEEE International Conference on Image Processing*, pages 2021–2024, 2012. doi: 10.1109/ICIP.2012.6467286.
- W. Yan, Y. Wang, S. Gu, L. Huang, F. Yan, L. Xia, and Q. Tao. The domain shift problem of medical image segmentation and vendor-adaptation by unet-gan. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 623–631. Springer, 2019.
- Y. Yang and P. Perdikaris. Physics-informed deep generative models, 2018.
- W. Yao and K. W. Leszczynski. An analytical approach to estimating the first order x-ray scatter in heterogeneous medium. *Medical Physics*, 36(7):3145–3156, 2009. doi: <https://doi.org/10.1118/1.3152114>. URL <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.3152114>.
- Y. Zhang. 2. image segmentation. In *Image Analysis*, pages 33–68. De Gruyter, 2017.
- Y. Zhang, S. Miao, T. Mansi, and R. Liao. Unsupervised x-ray image segmentation with task driven generative adversarial networks. *Medical Image Analysis*, 62:101664, 2020. ISSN 1361-8415. doi: <https://doi.org/10.1016/j.media.2020.101664>. URL <https://www.sciencedirect.com/science/article/pii/S136184152030030X>.
- A. Zhao, G. Balakrishnan, F. Durand, J. V. Guttag, and A. V. Dalca. Data augmentation using learned transformations for one-shot medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8543–8553, 2019.
- W. Zhao, D. Vernekohl, J. Zhu, L. Wang, and L. Xing. A model-based scatter artifacts correction for cone beam CT. *Medical Physics*, 43(4):1736–1753, Apr 2016. doi: 10.1118/1.4943796.

Pseudocode

This is the appendix, containing pseudocode for the functions described in the main body of the thesis.

A.1 Bi-directional Greedy Descent

This is the algorithm that refines the spatial mode, by applying the changes created in CreateDelta (A.4), measuring the change to the cost function and preserving those changes that reduced cost.

Algorithm 1 Descent Algorithm

```

Result: [L1_Alloy_contribution, L1_Thickness_contribution]
J = FindJ(spatial_model, precomputed_databases)
if currentRes is pseudo-pixel resolution then
    | // if we are at a pseudo-pixel resolution, reduce the number of
    |   iterations by the same factor as resolution has been decreased
    |   num_iterations = number of scatter centres / pixels combined to pseudo-pixel
else
    | num_iterations = number of scatter centres
end
// if num_iterations < 400, increase it to 400 so that some
  convergence can be observed at very coarse resolutions
num_iterations = max(num_iterations*2, 400)
for Iteration in num_iterations do
    | if Iteration = 1 then
    |   | initial J = J
    |   end
    |   forward delta = CreateDelta(currentRes, minibatchsize, segmentation)
    |   backward delta = -1 * forward delta
    |   forward gradient = FindDeltaJ(J, forward delta, spatial_model, precomputed
    |     databases)
    |   backward gradient = FindDeltaJ(J, backward delta, spatial_model, precom-
    |     puted databases)
    |   // remove gradient values where cost function has increased
    |   forward gradient[forward gradient > 0] = 0
    |   backward gradient[backward gradient > 0] = 0
    |   // where backward delta reduces cost function more than forward
    |     delta, accept those changes, and vice-versa
    |   update(backward gradient < forward gradient) = backward delta(backward
    |     gradient < forward gradient)
    |   update(forward gradient < backward gradient) = forward delta(forward gradi-
    |     ent < backward gradient)
    |   spatial_model.Thickness = spatial_model.Thickness + update.Thickness
    |   spatial_model.Alloy = spatial_model.Alloy + update.Alloy
    |   J = FindJ(spatial_model, precomputed_databases)
    |   final J = J
    |   RETURN spatial_model, initial J, final J
end

```

A.2 Continuity regularizer

This function is the equivalent of Equation 6.3. The addition of the result of this function to the cost function regularizes the model according to the expected rates of change of thickness and alloy across the image.

Algorithm 2 L1 Regularizer

```
Result: [L1_Alloy_contribution, L1_Thickness_contribution]
// Evaluate L1/continuity regularizer
initialization
alpha_E = initial alpha_E * Number of scatter centres
Filter_A = [-1, 1, 0]
Filter_B = [0, -1, 1]
Thickness_L1_A_X = convolve(SpatialModel.Thickness, Filter_A)
Thickness_L1_A_X = absolute value(Thickness_L1_A_X)
Thickness_L1_A_Y = convolve(SpatialModel.Thickness, Filter_A.Transpose)
Thickness_L1_A_Y = absolute value(Thickness_L1_A_Y)
Thickness_L1_B_X = convolve(SpatialModel.Thickness, Filter_B)
Thickness_L1_B_X = absolute value(Thickness_L1_B_X)
Thickness_L1_B_Y = convolve(SpatialModel.Thickness, Filter_B.Transpose)
Thickness_L1_B_Y = absolute value(Thickness_L1_B_Y)
Alloy_L1_A_X = convolve(SpatialModel.Alloy, Filter_A)
Alloy_L1_A_X = absolute value(Alloy_L1_A_X)
Alloy_L1_A_Y = convolve(SpatialModel.Alloy, Filter_A.Transpose)
Alloy_L1_A_Y = absolute value(Alloy_L1_A_Y)
Alloy_L1_B_X = convolve(SpatialModel.Alloy, Filter_B)
Alloy_L1_B_X = absolute value(Alloy_L1_B_X)
Alloy_L1_B_Y = convolve(SpatialModel.Alloy, Filter_B.Transpose)
Alloy_L1_B_Y = absolute value(Alloy_L1_B_Y)
Thickness_L1_X = element-wise maximum(Thickness_L1_A_X, Thick-
ness_L1_B_X)
Thickness_L1_Y = element-wise maximum(Thickness_L1_A_Y, Thick-
ness_L1_B_Y)
Alloy_L1_X = element-wise maximum(Alloy_L1_A_X, Alloy_L1_B_X)
Alloy_L1_Y = element-wise maximum(Alloy_L1_A_Y, Alloy_L1_B_Y)
Thickness_L1 = Thickness_L1_X + Thickness_L1_Y
Alloy_L1 = Alloy_L1_X + Alloy_L1_Y
RETURN [Thickness_L1, Alloy_L1]
```

A.3 Cost function which the gradient descent aims to minimise

The following evaluates the cost function for a given distribution of thicknesses and alloys. This is the sum of Equations 5.1, 5.4, 6.1, 6.2 and 6.3.

`spatial_model` stores the current thickness and alloy estimates.

Algorithm 3 Cost Function

Result: J, I_Guess,...

```

// Evaluate cost function J for a given geometry
initialization
J = 0
// Create a scatter estimate for the current geometry
Simulated_X-ray = EstimateScatter(spatial_model)
if GDflag then
    // GDflag included for testing purposes, toggles contribution of
    image norm differences to J
    fro = Frobenius norm((Raw_X-ray - Simulated_X-ray) * Scatter_Centres)
    Lmax = max((Raw_X-ray - Simulated_X-ray) * Scatter_Centres)
    Combined_Norm = (theta * fro2) + ((1 - theta) * Lmax2)
    J = J + Combined_Norm
end
if smoothflag then
    // Toggles contribution of smoothness regularizer to J
    [SmoothAlloyContribution, SmoothThicknessContribution] = SmoothnessReg-
    ulariser(spatial_model, Segmentation)
    J = J + SmoothAlloyContribution + SmoothThicknessContribution
end
if segflag then
    // Toggles contribution of segmentation regularizer to J
    [SegBoneContribution, SegSTContribution] = SegmentationRegular-
    izer(spatial_model, Segmentation)
    J = J + SegBoneContribution + SegSTContribution
end
if physflag then
    // Toggles contribution of regularizer over physically allowed
    values to J
    alloy_physicality = (min(0, spatial_model.Alloy)) + (max(0, spa-
    tial_model.Alloy - 1))
    thickness_physicality = (min(0, spatial_model.Thickness)) + (max(0, spa-
    tial_model.Thickness - 30))
    if max(abs(physicality)) > 0 then
        | J = J - (alphaB * Norm(alloy_physicality))
    end
    if max(abs(physicality)) > 0 then
        | J = J - (alphaB * Norm(thickness_physicality))
    end
end
if continuityflag then
    // Toggles contribution of continuity regularizer to J
    [L1_AlloyCntribution, L1_ThicknessContribution] =
    L1Regularizer(spatial_model)
    J = J + L1_AlloyContribution + L1_ThicknessContribution
end
RETURN J

```

A.4 Perturbation to material estimate

This function is used to generate a Delta, an array which can be added to the current material estimation to measure gradients in the cost function.

ScatterCentres is a Boolean array with the same shape as the sub-sampled image. It contains True values for every index that contains scattering material such as bone or soft tissue. Changes to the thickness and alloy values outside of these scatter centres would have no effect on the cost function, so only scattering pixels can be perturbed by delta.

In the final line, the values of delta that will be applied to the thickness estimates are multiplied by 30, since the valid range of alloy values is 0 - 1 and the valid range of thickness values is 0 - 30.

Algorithm 4 CreateDelta

```

Result: delta
// Create array Delta which will perturb the geometry for
  calculating gradient
initialization
[X, Y] = size(ScatterCentres)
delta = zeros(X, Y, 2)
// validIndices are all non-zero entries, which are valid to perturb
validIndices = find(ScatterCentres==1)
validIndices_rand = validIndices(randperm(length(validIndices)))
if Current Resolution = Inf then
  // Currently at the Inf pseudo-pixel resolution, same
  modification should be applied to each parameter within a
  region.
  minibatch_size = min(3, minibatch_size)
  for pseudo-pixel to modify in minibatch do
    if pseudo-pixel to modify = Thickness then
      | delta.Thickness = delta.Thickness + modification
    end
    if pseudo-pixel to modify = Bone Alloys then
      | delta.Alloy[segmentation = bone] = delta.Alloy[segmentation = bone] +
      | modification
    end
    if pseudo-pixel to modify = Soft Tissue Alloys then
      | delta.Alloy[segmentation = soft tissue] = delta.Alloy[segmentation =
      | soft tissue] + modification
    end
  end
  RETURN delta
end
if Current Resolution = Pseudo-pixel resolution then
  for Pseudo-pixel to modify in minibatch do
    | delta[pseudo-pixel to modify] = delta[pseudo-pixel to modify] + modifica-
    | tion
  end
  RETURN delta
end
if Current Resolution is NOT pseudo-pixel resolution then
  | delta[pixel to modify] = delta[pixel to modify] + modification
  RETURN delta
end

```

A.5 Selecting the next cycle case

The following begins a cycle with a resolution determined by the recent improvement factor. This is the function responsible for determining which resolution should be selected at the end of a given cycle.

Algorithm 5 SelectCycleCase

```

Result: spatial_model
// Select and execute the appropriate cycle case based on recent
  improvement
initialization
// Define the improvement factor thresholds for each case
Case1Limit = 0.8
Case2Limit = 0.92
Case3Limit = 1.05
if First Iteration then
  | // This is the initialization condition, so that the first cycle
  |   is one pass at the lowest resolution
  |   improvementFactor = Case1Limit*0.9
else
  |   improvementFactor = LatestImprovement/PreviousImprovement
end
if improvementFactor < Case1Limit then
  |   cycleCase = 1
  |   TemporaryImprovement = LatestImprovement
  |   [spatial_model, PreviousImprovement, LatestImprovement...] = SingleCycle()
else if improvementFactor < Case2Limit then
  |   CycleCase = 2
  |   [spatial_model, PreviousImprovement, LatestImprovement...] = VCycle()
  |   TemporaryImprovement = PreviousImprovement
else if improvementFactor < Case3Limit then
  |   cycleCase = 3
  |   if find(resolutions == currentRes) == size(resolutions,2) then
  |     | currentRes = resolutions(end)
  |   else
  |     | currentRes = resolutions(find(resolutions == currentRes)+1)
  |   end
  |   TemporaryImprovement = LatestImprovement
  |   [spatial_model, PreviousImprovement, LatestImprovement...] = SingleCycle()
else
  |   cycleCase = 4
  |   if size(resolutions,2) > 1 then
  |     | if currentRes == Resolutions(1) then
  |       | currentRes = resolutions(find(resolutions == currentRes)+1)
  |     end
  |     | resolutions = resolutions(:,2:end)
  |   end
  |   [spatial_model, PreviousImprovement, LatestImprovement...] = VCycle()
  |   TemporaryImprovement = PreviousImprovement
end
PreviousImprovement = TemporaryImprovement

```

A.6 Finding the gradient of J numerically for the current material estimation

Finds the numerical gradient of J with respect to a given parameter by making a small change to that parameter and measuring the change it causes.

Algorithm 6 Find delta J

Data: *spatial_model, Delta*

Result: *Gradient*

spatial_model_Temp = spatial_model

spatial_model_Temp.Thickness = TI_Model.Thickness + Delta.Thickness

spatial_model_Temp.Alloy = TI_Model.Thickness + Delta.Alloy

J_Star = FindJ(spatial_model_Temp)

RETURN J_Star - J

A.7 Re-scale a thickness alloy model

This function will re-scale a thickness-alloy model of anatomy, from an initial resolution to a new, given resolution. This function additionally assumes that if a thickness-alloy model is being down-sampled then that is taking place during a V-cycle, and it can thus be assumed that the model will be up-sampled to the initial resolution later in the execution. This function therefore also stores the model at each change, and when re-up-sampling this historical data is used to produce a best estimate.

Algorithm 7 Rescale Spatial Model

Result: RescaledModel

```
// Change the resolution of a geometry estimate, carrying forward
  previous work
initialization
if increasing in V-cycle then
  // We're in a V-cycle and increasing resolution
  PreviousTargetModel = RecordedModels[currentRes, ResolutionCounter - 1]
  PreviousSourceModel = RescaleModel(PreviousTargetModel, SourceResolution)
  // Replicate previous downsampling method
  LowResWorkDone = Model - PreviousSourceModel
  HighResWorkDone = resize(LowResWorkDone, TargetResolution)
  RescaledModel = PreviousTargetModel + HighResWorkDone
else
  // InitiateImage creates an estimate according to the current
  rules for creating an initial estimate.
  coarsePriorSpatialModel = initiateImage(Segmentation, coarseXY)
  GDWorkDone.Thickness = SpatialModel.Thickness - coarsePriorSpatialM-
  odel.Thickness
  // Work done by optimization
  GDWorkDone.Alloy = SpatialModel.Alloy - coarsePriorSpatialModel.Alloy
  newPriorSpatialModel = initiateImage(Segmentation, newXY)
  RescaledModel.Thickness = newPriorSpatialModel.Thickness + imres-
  ize(GDWorkDone.Thickness, newXY)
  RescaledModel.Alloy = newPriorSpatialModel.Alloy + imres-
  ize(GDWorkDone.Alloy, newXY)
end
```

A.8 Segmentation regularizer

This function is the equivalent of Equation 6.2. The addition of the result of this function to the cost function regularizes the model according to the expected alloy values from the segmentation of the image into regions of bone and soft-tissue.

Algorithm 8 Segmentation Regularizer

Result: SegBoneCont, SegSTCont

```
// Find the contribution to J from the segmentation
alphaC = initialAlphaC * square root(number of scatter centres)
// incorrectBone and Alloy are all the parameters that diverge from
  the predicted value
IncorrectBone = (SegmentedImage == Bone) .* (1 - spatial_model.Alloy)
IncorrectST = (SegmentedImage == Soft-Tissue). * (spatial_model.Alloy >= 1)
// take Frobenius norm, multiply by weighting
SegBoneCont = alphaC * Frobenius norm(IncorrectBone)
SegSTCont = alphaC * Frobenius norm(IncorrectST)
RETURN [SegBoneCont, SegSTCont]
```

A.9 Cycle of optimization at a single resolution

Optimization in our algorithm occurs over a series of cycles, which can include iterations at many resolutions, or at just one. This function executes a cycle of optimization at a single resolution, which has been determined by the function described in Section A.5.

Algorithm 9 SingleCycle

Result: spatial_model, rhn, rh0 ...

```
// Perform one cycle of iterative optimization at the given  
resolution.
```

```
initialization
```

```
if currentRes precomputed not loaded then
```

```
  | load precomputed databases for currentRes
```

```
end
```

```
spatial_model = rescale_spatial_model(spatial_model, currentRes)
```

```
if batchsize==0 then
```

```
  | batchsize = number of scatter centres
```

```
else if batchsize<1 then
```

```
  | batchsize = round(number of scatter centres * batchsize)
```

```
end
```

```
[spatial_model, initial J, final J] = DescentAlgorithm(spatial_model, currentRes,  
precomputed databases)
```

A.10 Smoothness regularizer

This function is the equivalent of Equation 6.1. The addition of the result of this function to the cost function regularizes the thickness-alloy model according to our expectation on the smoothness of organic anatomies.

Algorithm 10 Smoothness Regularizer

Result: SmoothAlloyCont, SmoothThicknessCont, ...

```
// Find the contribution to J from smoothness
initialization
alphadxi = initial_alphadxi * number of scatter centres
// convolve thickness and alloy with smoothness filters in X and Y
smoothFilter = [-1 , 2 , -1 ; -4 , 8 , -4 ; -1 , 2 , -1]
ThicknessSmoothnessY = convolve(spatial_model.Thickness, smoothFilter)
ThicknessSmoothnessX = convolve(spatial_model.Thickness, smoothFilter.transpose)
AlloySmoothnessY = convolve(spatial_model.Alloy, smoothFilter)
AlloySmoothnessX = convolve(spatial_model.Alloy, smoothFilter.transpos)
// Identify edges in the segmentation
edgesInX = edge detect along X axis applied to Segmentation
edgesInY = edge detect along Y axis applied to Segmentation
// blurred object edges is a blurred version of the bone-air
// transition edges, and these are less strongly enforced
blurred object edges = Gaussian filter(edge(segmentation = open beam))
// Total Alloy smoothness is sum of X an Y convolutions multiplied
// by (1-edges)
AlloySmoothness = ((AlloySmoothnessY+AlloySmoothnessX) * (1-edgesInX) * (1-edgesInY))
// Total Thickness smoothness is sum of X an Y convolutions
// multiplied by (1-blurred object edges)
ThicknessSmoothness = ((ThicknessSmoothnessY+ThicknessSmoothnessX)*(1 - blurred object edges))
// contribution to J is half of squared Frobenius norm of these
// smoothness images
smoothAlloyContribution = (0.5*(Frobenius norm(AlloySmoothness)2))
smoothThicknessContribution = (0.5*(Frobenius norm(ThicknessSmoothness)2))
SmoothAlloyCont = (alphadxi*0.5*(Frobenius norm(AlloySmoothness)2))
SmoothThicknessCont = (alphadxi*0.5*(Frobenius norm(ThicknessSmoothness)2))
RETURN [SmoothAlloyCont, SmoothThicknessCont]
```

A.11 Cycle of optimization across many resolutions (V-cycle)

Cycle from current resolution, through each currently permitted coarser resolution, then refine back to current resolution again. At each resolution, iterations of optimization will be carried out.

Algorithm 11 V Cycle

```
Result: spatial_model, initial J, final J
// Perform a pass of all resolutions coarser than the current, then
  increase to current resolution.
initialization
// Start at the resolution below current (unless current is lowest
  allowed), then return to current.
for resolution in  $-\max(\text{currentRes} - 2, 0):1:\text{currentRes} - 1$  do
  // prepData loads all of the required prerequisite data from IBEX
  files
  if precomputed databases at this resolution have not been loaded then
    | load this resolution precomputed database
  end
  // Rescale Geometry to preserve work done
  spatial_model = rescale_spatial_model(spatial_model, resolution)
  if batchsize==0 then
    | // batchsize = 0 indicates all parameters should be selected,
    |   i.e. the batch size should be the maximum possible
    |   batchsize = number of scatter centres
  else if batchsize<1 then
    | //  $0 < \text{batchsize} < 1$  indicates batchsize should be that fraction
    |   of all parameters, i.e. batchsize = 0.5 means the batch
    |   should contain 50% of all possible parameters
    |   batchsize = round(number of scatter centres * batchsize)
  end
  [spatial_model, initial J, final J] = DescentAlgorithm(spatial_model, resolu-
  tion, precomputed databases)
end
```

Colophon

This thesis is based on a template developed by Matthew Townson and Andrew Reeves. It was typeset with L^AT_EX 2_ε. It was created using the *memoir* package, maintained by Lars Madsen, with the *madsen* chapter style. The font used is Latin Modern, derived from fonts designed by Donald E. Knuth.