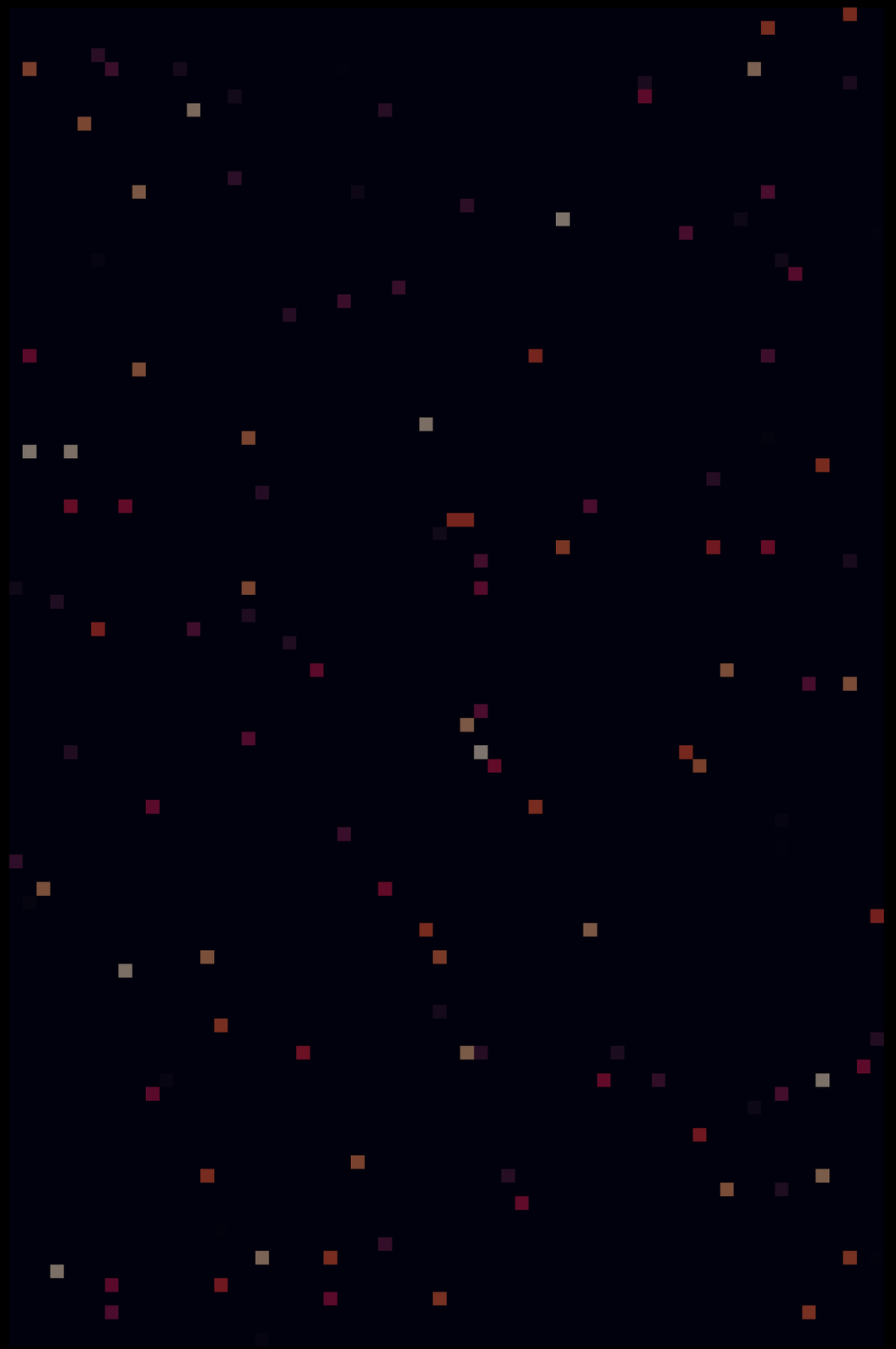
SPARSITY-BASED ALGORITHMS FOR INVERSE PROBLEMS

Poulami Somanya Ganguly

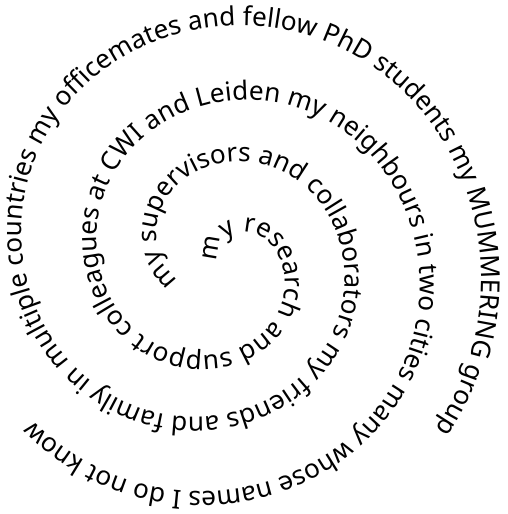


SPARSITY-BASED ALGORITHMS FOR INVERSE PROBLEMS

Poulami Somanya Ganguly

Acknowledgements

By setting the direction and goals of my projects, by working alongside and thinking together, by making sure I was fed and looked after, by ensuring my workstation and office were functional, by taking my mind off research with their company, by answering my queries without irritation, by making conferences and workshops enjoyable, by checking up on me throughout the pandemic,



made this PhD work possible. To those among them reading this, thank you.

Sparsity-based Algorithms for Inverse Problems

Proefschrift

ter verkrijging van

de graad van doctor aan de Universiteit Leiden,

op gezag van rector magnificus prof.dr.ir. H. Bijl,

volgens besluit van het college voor promoties

te verdedigen op donderdag 8 december 2022

klokke 12:30 uur

door

Poulami Somanya Ganguly

geboren te Kolkata, India

in 1992

Promotores:
Prof.dr. K.J. Batenburg
Prof.dr. H.J. Hupkes

Co-promotor:
Dr. F. Lucka (Centrum Wiskunde & Informatica)

Promotiecommissie:
Prof.dr. F.A.J. de Haas
Prof.dr. A. Doelman
Prof.dr. C.B. Schönlieb (University of Cambridge)
Prof.dr. A.B. Dahl (Technical University of Denmark)
Dr. D.M. Pelt

The research presented in this dissertation was carried out at Centrum Wiskunde & Informatica (CWI) and Leiden University.

Financial support was provided by the European Union's Horizon 2020 Research and Innovation programme under the Marie Sklodowska-Curie grant agreement no. 765604.

Contents

Chapter 1

Introduction

In this chapter we give a general introduction to the field of inverse problems and algorithmic approaches to solving such problems in a few application areas. We also introduce the reader to notions of sparsity and show how sparsity is used to tackle the various research questions investigated in this thesis.

1.1 Background

Scientific questions can be broadly divided into two kinds. The first kind seeks to question the effects of a set of causal factors while the second seeks to determine the causal factors given the effects. In this thesis we deal with the latter type of questions.

We shall restrict ourselves to situations where the effects are observations or measurements of a physical system, and the causal factors are certain variables that characterize the system. One common starting point in this case is to construct a simplified representation or physical model of the system.

An example of this process of model building is the physical theory of the interaction of light with matter. Such a theory enables us to calculate, among other things, the interaction of X-rays passing through a three-dimensional object. The measurements from this system are images – two-dimensional snapshots of the X-ray beam after it emerges from the object. These snapshots can be obtained using an X-ray detection system and compared against our prediction from the physical theory. It turns out that in this case our predictions match the experimental measurements well, thus validating the correctness of our theory.

The problem described above is a *direct* or *forward* problem, where we predict the effects given causes and a reliable model. Complementary to this problem is the *inverse problem*, where we want to infer the physical properties of the 3D object, in particular its capacity to interact with X rays, using a set of 2D images. An illustration of both problems is shown in Figure 1.1. It turns out that the inverse problem of reconstruction brings about a different set of challenges to the forward problem of projection, and in order to solve the former problem we must make further assumptions about the 3D object.

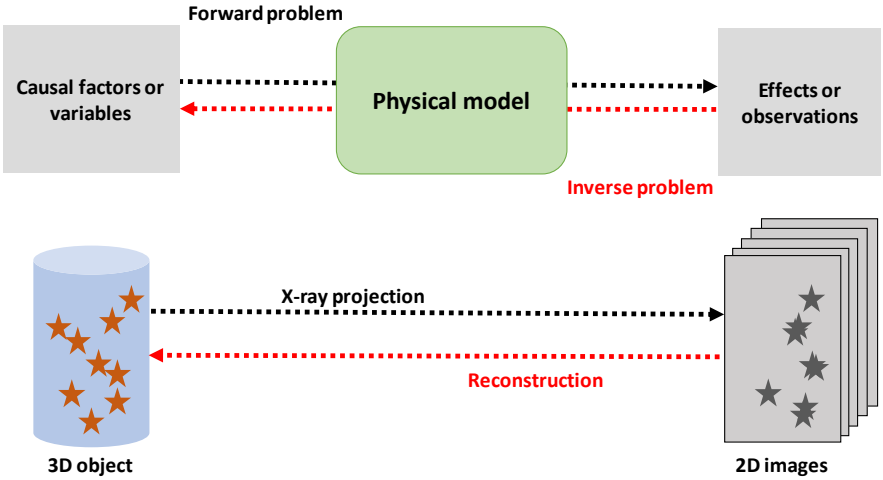


Figure 1.1: A forward problem is one where we predict the effects of a set of causal factors given a physical model of the system. An inverse problem is one where we invert this process. An example of a forward problem is the calculation of a 2D X-ray projection of a 3D object; the inversion of a set of such images to infer the 3D object is the inverse problem of reconstruction.

One such assumption that is central to the work in this thesis comes from the notion of sparsity. Broadly speaking, sparsity is the assumption that only a small set of variables or causal factors is sufficient to explain the measurements of a system. In the example of X-ray reconstruction above, sparsity could mean that the object is made of a small set of discrete constituents. Using this as prior knowledge of our 3D object makes the inversion procedure much more reliable. Stated differently, assuming sparsity enables us to limit our search for causal factors to a small set.

In this thesis we study several application areas where notions of sparsity yield practical algorithms for inverse problems. In the next section we first introduce these application areas and discuss the forward problems therein. Next we present the mathematical framework of inverse problems and discuss ways to include sparsity in this framework. In the penultimate section, we return to our application areas and show how practical algorithms can be designed to tackle sparse inverse problems in these areas. Finally, we present four research questions that are investigated in the following chapters of this thesis, and provide a brief abstract of our methods and contributions.

1.2 Application areas

In this section we introduce three application areas that were studied in this thesis, and give some mathematical background to these areas.

1.2.1 X-ray computed tomography

X-ray computed tomography (CT) is a powerful method to visualize and obtain quantitative information about the inside of an object non-destructively. X-ray CT is widely used in medical settings for diagnostic purposes [1], in materials science for studying structural changes in materials [2] and in cultural heritage for probing the construction of art objects [3].

In this imaging modality, an X-ray beam is used to generate projection images of an object of interest. The flux of the X-ray beam changes as it passes through the object according to the Beer–Lambert law:

$$I = I_0 \, e^{-\int_0^l \mu(z)\, dz}\,, \tag{1.1}$$

where I_0 is the flux of the incident X-ray beam, I is the flux after the beam has passed through a distance l inside the object and μ is the attenuation coefficient that denotes the capacity of the materials in the object to absorb X rays. Dividing both sides of (1.1) by I_0 and taking the logarithm, we arrive at the linear projection model of X-ray CT:

$$\log\left(\frac{I}{I_0}\right) = -\int_0^l \mu(z)\, dz \tag{1.2}$$

Many different experimental setups are used for X-ray CT depending on the application, but in most setups the basic acquisition strategy consists of rotating the sample with respect to the incident X-ray beam to acquire measurements along several projection angles. The emergent X-ray beam after absorption by the sample is detected using a detection system. In this thesis we focus on *parallel-beam* CT, where the distance between X-ray source and object is large enough to approximate the incident rays as being parallel to each other. This is the setup shown in Figure 1.2. Using (1.2) the forward projection of a 2D object $f(x, y)$ taken along a projection angle θ, $P_\theta(t)$, can then be written as

$$P_\theta(t) := \mathcal{R}(f) = -\iint_{\mathbb{R}^2} f(x, y)\, \delta(x\cos\theta + y\sin\theta - t)\, dx\, dy\,, \tag{1.3}$$

where the function $f : \mathbb{R}^2 \to \mathbb{R}$ is a finite integrable function with bounded support describing the attenuation of the object. Note that $\mu(z)$ in (1.2) is equivalent to the function $f(x, y)$ evaluated on the line given by $x\cos\theta + y\sin\theta = t$. $\mathcal{R}(f)$ is known as the Radon transform of the function $f(x, y)$, and δ denotes the delta function. Using (1.3) a set of projections $P_\theta(t)$, $\theta \in [0, \pi)$ can be acquired and rearranged to give a *sinogram*. In Figure 1.2, we show a popular analytical object – known as the Shepp-Logan phantom – along with its sinogram.

The tomographic reconstruction problem refers to the inversion of (1.3) to yield a suitable function $f(x, y)$ from a set of measurements $P_\theta(t)$, $\theta \in [0, \pi)$. In Section 1.4 we shall return to this inverse problem and discuss it in more detail.

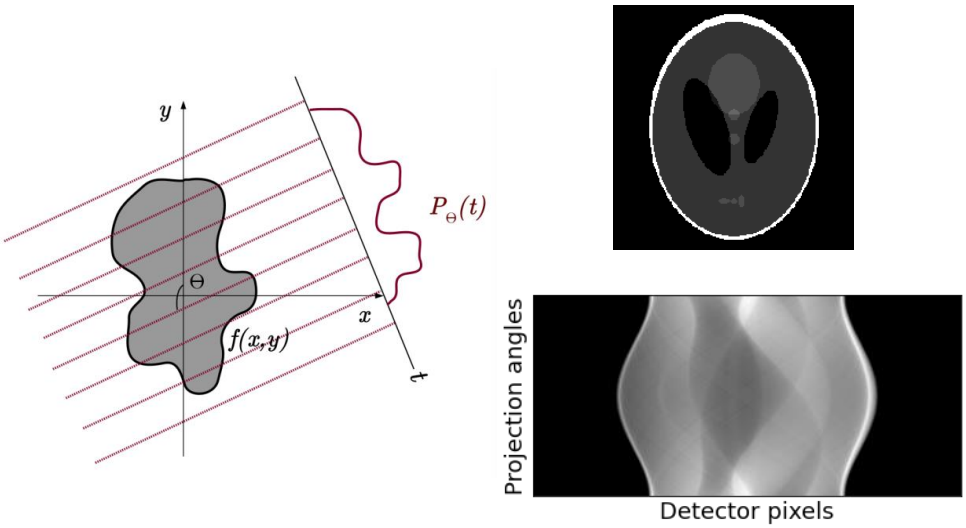


Figure 1.2: Parallel-beam X-ray CT (left), Shepp–Logan phantom (top right) and its sinogram (bottom right).

1.2.2 Electron tomography

The next application area of relevance to this thesis is electron tomography (ET). ET is the method of choice for imaging nanoparticles and biological macromolecules at atomic or near-atomic resolutions. Imaging with an electron beam allows for much higher resolutions compared to X-ray imaging because of the shorter wavelength of electrons [4], [5].

Images in ET are generated by passing a focused electron beam through a sample. In one common modality, each projection image is generated in transmission electron microscopy (TEM) mode. In this mode, the whole sample is irradiated with the incident electron beam at the same time and the transmitted electron beam is detected. Alternatively, the electron beam can be focused to scan the sample one small area at a time. This mode is known as scanning transmission electron microscopy (STEM). To obtain images at different projection angles, the sample is tilted with respect to the beam. For thin samples, the linear projection model (1.2) holds for each image in the tilt series. Similar to X-ray CT, the inverse problem in ET consists of inverting the forward model to infer the structure of nanoparticles and macromolecules from their projections.

In this thesis, we present methods for two different applications of ET. These are atomic-resolution ET and cryoET. In Section 1.4, we focus on each of these areas separately and state the inverse problems we investigated for each.

1.2.3 Vascular network formation

The final application area we study in this thesis is of relevance to developmental and cancer biology. Vasculogenesis is the process by which a primitive circulatory system is generated in vertebrates. Following the generation of a primitive network, new blood vessels arise by sprouting and expanding, in a process known as angiogenesis. Angiogenesis also occurs in certain types of cancer, where it contributes to tumour maintenance and metastasis.

Understanding how individual cells organize to form mature vascular networks is a long-standing question. In particular, the contribution of cell–cell interactions and environmental cues are still a topic of research. One way to investigate the conditions for vascular network formation is using computer simulations, where different cell–cell interactions and environmental cues can be prescribed and the resulting long-term dynamics can be studied. Simulation studies are particularly effective because all the parameters of a chosen model can be adjusted and different parameter regimes, which may not be easy to probe in experimental studies, are easily simulated.

Different simulation paradigms have been used in the literature to simulate vascular network generation. One paradigm is cellular Potts model, a lattice-based simulation where cells are represented as patches of interacting spins. A complementary paradigm is a lattice-free particle-based model, where each cell is represented by an ellipse and is assumed to interact with all other cells in a prescribed neighbourhood. The forward model of this particle-based paradigm is given by a Langevin equation:

$$\frac{d\boldsymbol{v}_i}{dt} = \frac{1}{m_i}\Big(-\tau \boldsymbol{v}_i + \sum_{j\neq i} \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|} F_{ij} + \boldsymbol{\eta}\Big), \quad \boldsymbol{v}_i = \frac{d\boldsymbol{x}_i}{dt}, \tag{1.4}$$

where $\boldsymbol{x}_i$ denotes the position of cell i at time t, $\boldsymbol{v}_i$ is the velocity of cell i, m_i is the mass of cell i, τ is the so-called damping constant, F_{ij} is the pairwise interaction between cells i and j, and $\boldsymbol{\eta}$ is a stochastic noise term.

Using the above equation, the long-time dynamics of cells can be simulated for different parameter values. The steady-state solutions can then be used to determine the parameter regions – and hence conditions – for network formation.

An alternative approach is to start directly from experimental observations of network formation and infer the interaction terms and parameter values in (1.4). In Section 1.4, we shall return to this inverse problem and describe our methods to tackle it.

1.3 Inverse problems

In the previous section, we described several application areas and the forward problems that arise in each of them. We also mentioned briefly the inverse to these forward problems. In this section we discuss the mathematical framework of inverse problems and describe ways to solve such problems reliably.

Inverse problems arise in many areas of science and engineering, where the goal is to infer specific variables given measurements of a system and a reliable physical model [6].

Mathematically, this translates to inferring x from data y, where the two are related by the following equation:

$$A(x) = y, \tag{1.5}$$

where A is the forward model.

One of the examples described above is that of tomographic reconstruction. The forward problem of tomography is the projection of a 3D object along a set of projection angles, while the inverse is combining the information from a set of projections to obtain a reconstruction. In this example, the forward problem has a well-defined solution but the inverse problem does not.

One way to understand the difference between forward and inverse problems is the notion of *well-posedness*. A mathematical problem is said to be well-posed if its solution exists for arbitrary data (existence) and is unique (uniqueness). Additionally, the solution must depend continuously on the data such that small changes in the data result in correspondingly small changes in the solution (stability). Problems that do not meet these conditions are known as *ill-posed*.

Some physical intuition regarding the ill-posedness of inverse problems is obtained by using the idea of entropy from the second law of thermodynamics and information theory. Forward problems are generally those that describe physical phenomena oriented along the cause–effect sequence [6]. The cause–effect sequence is determined by the second law of thermodynamics, which posits an increase in total entropy in the direction of time evolution. This means that the solution to a direct problem has lower "information content" than the data. The opposite is true for an inverse problem, where data with lower information content must be used to infer unknowns with higher information content.

The ill-posedness of inverse problems – specifically the fact that small variations in the data (caused, for e.g. , by measurement noise) lead to large variations in the solution – makes it difficult to obtain a meaningful solution to an inverse problem. This is addressed by using prior knowledge about the physical system being studied. The mathematical theory that deals with this is called regularization.

An illustration of regularization is provided by the use of Tikhonov regularization in X-ray CT. The discrete formulation of the X-ray CT problem is given by:

$$\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{y}, \tag{1.6}$$

where $\boldsymbol{A}$ is the linear forward operator which amounts to the discretized version of the Radon transform (1.3), $\boldsymbol{y}$ is a vector of discrete projection data and $\boldsymbol{x}$ is the unknown discretized reconstruction. The reconstruction problem can then be stated as an optimization problem where we seek to minimize the difference with respect to projection data. The least-squares solution to the discrete reconstruction problem is

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimize}} \quad \|\boldsymbol{y} - \boldsymbol{A}\,\boldsymbol{x}\|_2^2 \tag{1.7}$$

An example reconstruction of the Shepp-Logan phantom using a least-squares strategy is shown in Figure 1.3. A common way to regularize this problem is to minimize not just the discrepancy with respect to the projection data but also the energy of the solution, defined

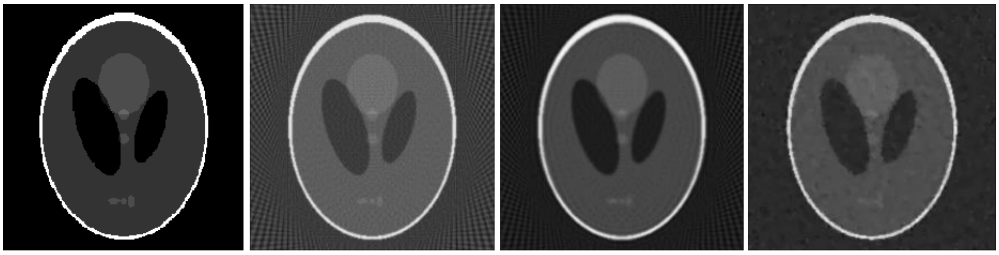


Figure 1.3: Tomographic reconstructions of the Shepp–Logan phantom using filtered back-projection (FBP), the simultaneous iterative reconstruction technique (SIRT) and total variation (TV) minimization. SIRT solves a least-squares problem with added preconditioning; TV solves a regularized least-squares problem that penalizes large gradients in the reconstructed image. All reconstructions were performed using the Astra Toolbox [7] and the Operator Discretization Library (ODL) [8].

as its ℓ^2-norm. This regularization, known as Tikhonov regularization, then amounts to the optimization problem:

$$\underset{\boldsymbol{x} \in \mathbb{R}^d}{\text{minimize}} \quad \|\boldsymbol{y} - \boldsymbol{A}\,\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_2^2, \tag{1.8}$$

where $\lambda > 0$ – the regularization parameter – adjusts the relative weighting of the two terms in the optimization objective.

1.3.1 Sparse inverse problems

The example of regularization shown above penalizes the energy of the solution. In the last three decades, a different form of prior knowledge has been shown to be a powerful technique for solving a host of inverse problems [9], [10]. This prior knowledge relates to the *sparsity* of the unknown vector $\boldsymbol{x}$. One notion of sparsity is given by the number of nonzero elements of the vector $\boldsymbol{x}$, which is called the ℓ^0 "norm". The ℓ^0 "norm" of a vector $\boldsymbol{x} \in \mathbb{R}^d$ is given by

$$|\boldsymbol{x}|_0 := \sum_{i=1}^{d} |x_i|^0, \tag{1.9}$$

and the vector $\boldsymbol{x}$ is said to be s-sparse if

$$|\boldsymbol{x}|_0 \leq s. \tag{1.10}$$

A sparse inverse problem is one where we look for the sparsest solution that explains the observed data. Mathematically, we can state a sparse inverse problem as a constrained optimization problem where the goal is to

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimize}} \quad |\boldsymbol{x}|_0 \qquad \text{subject to} \quad \boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{y}. \tag{1.11}$$

In some scenarios a reformulation of the optimization problem may be more appropriate. For example, we may choose to relax the exact equality in the constraint to account for measurement noise

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimize}} \quad |\boldsymbol{x}|_0 \qquad \text{subject to} \quad \|\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{y}\|_2^2 \leq \epsilon. \tag{1.12}$$

Or, we could switch the objective function with the constraint:

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimize}} \quad \|\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{y}\|_2^2 \qquad \text{subject to} \quad |\boldsymbol{x}|_0 \leq s. \tag{1.13}$$

Objective and constraint functions may also be added to result in an optimization problem analogous to Tikhonov regularization (1.8):

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimize}} \quad \|\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda|\boldsymbol{x}|_0. \tag{1.14}$$

The ℓ^0 term in the above formulations makes the optimization problem nonconvex, and thus sensitive to initialization. A convex surrogate is achieved by replacing the ℓ^0 term with the ℓ^1 norm of $\boldsymbol{x}$. The convex surrogate of (1.14) is

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimize}} \quad \|\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1. \tag{1.15}$$

The convex formulation above can be solved with guarantees on existence and convergence of the solution. However, nonconvex optimization methods, such as greedy pursuit and simulated annealing, have also been used to solve the ℓ^0 minimization problem directly [11].

In many applications, tomographic reconstruction being one of them, the unknown reconstruction is not sparse per se, but can be sparsely coded in a suitable orthonormal basis. For e.g., images can be assumed to be piecewise constant, which implies sparsity in the space of gradients. This results in the total variation (TV) regularization method that results in surprisingly good reconstructions even for heavily undersampled data:

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimize}} \quad \|\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\||\nabla\boldsymbol{x}|\|_1. \tag{1.16}$$

Reconstruction of the Shepp–Logan phantom using TV regularization is shown in Figure 1.3.

In some applications like in atomic-resolution ET, discretization of the reconstruction $\boldsymbol{x}$ as an image may not be the most suitable. In situations where $\boldsymbol{x}$ is a set of unknown cardinality (e.g. the atomic coordinates in a nanoparticle, with an unknown number of atoms), the optimization problem (1.15) is no longer convex in the space of atomic coordinates. A convex formulation in such situations is achieved by lifting the problem to the vector space of measures [12].

1.4 Computational solution of inverse problems using sparsity

In this section, we return to the application areas introduced in Section 1.2 and present ways to include sparsity in the design of efficient algorithms. At the end of each subsection, we summarise our use of sparsity in that particular problem in a red box.

1.4.1 Sparse design of implementation-adapted filters for X-ray CT

The discretized tomographic reconstruction problem is the estimation of $\boldsymbol{x}$ from equation (1.6). There exist several methods to obtain an estimate of the reconstruction $\boldsymbol{x}$ given data $\boldsymbol{y}$. Direct methods that use discretized inversions of the continuous Radon transform (1.3) are some of the most popular methods due to their speed. An exact inversion of the continuous Radon transform is possible and leads to the following inversion formula for the function $f(x, y)$:

$$f(x,y) = \int_0^{\pi} q_\theta(x\cos\theta + y\sin\theta)\,d\theta\,, \quad q_\theta(t) = \int_{\infty}^{-\infty} |u|\hat{P}_\theta(u)e^{2\pi i u t}\,du, \tag{1.17}$$

where $|u|$ is known as the ramp filter in Fourier space and $\hat{P}_\theta$ denotes projection data in Fourier space. Filtered backprojection (FBP), a real-space direct method, computes a discretized version of the above inversion formula, such that

$$f(x,y) \approx \sum_{\theta\in\Theta}\sum_{\tau\in T} P_\theta(\tau)h_\theta(x\cos\theta + y\sin\theta - \tau),$$

where h is a discretized filter in real space. Equivalently, starting from the algebraic equation (1.6), the FBP reconstruction $\tilde{\boldsymbol{x}}_{\text{FBP}}$ of projection data $\boldsymbol{y}$ is given by

$$\tilde{\boldsymbol{x}}_{\text{FBP}} = \boldsymbol{A}^T(\boldsymbol{y} * \boldsymbol{h}) = \boldsymbol{A}^T\boldsymbol{C}_h\,\boldsymbol{y}, \tag{1.18}$$

where $\boldsymbol{C}_h$ denotes convolution with filter $\boldsymbol{h}$ and $\boldsymbol{A}^T$ is known as the backprojection operator.

In Fourier-space direct methods such as GridRec, both filtering and backprojection are performed in Fourier space, after which a fast Fourier transform (FFT) is used to convert the Fourier-space reconstruction to a real-space reconstruction. In addition, Fourier-space methods often use a windowing function to improve the accuracy of interpolation in Fourier space [13].

An important point to note is that, although the problem of inversion is well defined in the continuous setting, the discretized reconstruction formula (1.18) depends on the choice of discretization and interpolation. These choices are usually implementation-specific, which means that they differ across the various available open-source software implementations of direct algorithms, and contribute to quantitative differences between reconstructions from each implementation.

Direct methods usually result in poor reconstructions when noise in the data is high or data have been collected over a limited angular range. For such data, methods that solve the linear least-squares problem (1.7) iteratively are better. One popular iterative method is SIRT, which solves the linear least-squares problem with additional preconditioning and, optionally, non-negative constraints. For large data, a major limiting factor to the practical application of iterative methods is that the computation time required for reconstructing is much larger than the time required by direct methods.

A class of filter-optimization methods seek to augment direct methods with some of the advantages of iterative methods without compromising on the speed of computation. In such methods, the filter in direct algorithms ($\boldsymbol{h}$ in (1.18)) is learned from the data $\boldsymbol{y}$, following which they can be used on-the-fly with direct methods in place of standard hand-crafted filters. Filter learning using a minimum-residual approach has been performed for FBP [14] as well as the Feldkamp–Davis–Kress (FDK) algorithm [15], which generalizes FBP to cone-beam tomography setups.

A minimum-residual filter for data $\boldsymbol{y}$ can be computed by solving the following optimization problem:

$$\underset{\boldsymbol{h}}{\text{minimize}} \quad \|\boldsymbol{y} - \boldsymbol{A}\,\boldsymbol{A}^T\boldsymbol{C}_{\boldsymbol{y}}\,\boldsymbol{h}\|_2^2. \tag{1.19}$$

The minimum-residual filter $\boldsymbol{h}^* := \sum h_i^*\,\boldsymbol{b}_i$ is expressed as a linear combination of basis vectors $\boldsymbol{b}_i$. Sparse design of the filter is possible by choosing an appropriate basis such that only a few filter coefficients have to be learned. A binned basis with exponentially wider bins away from the centre of the detector array was first proposed for FBP [14] and later used for filter computation for FDK [15] without loss of reconstruction accuracy. Such an exponential binning basis translates to a basis of linear combinations of cosines in Fourier space.

In Chapter 2 we use such sparse-basis filters to tackle the problem of reproducibility in synchrotron tomography. Hardware and software vary across synchrotrons, and the results of experiments performed by users at different facilities are often not directly comparable with each other. We focus on the image reconstruction block in the synchrotron tomography pipeline, where differences between discretization and interpolation in various software packages play a role in enhancing differences between experimental results. We show that minimum-residual filters can improve the similarity between reconstructions (of synthetic and real data) obtained from several open-source implementations of direct algorithms, and contribute to a more reproducible reconstruction block in the synchrotron pipeline.

Sparsity is used to limit the number of filter coefficients of minimum-residual filters.

1.4.2 Grid-free, sparse reconstruction of nanocrystal defects

The goal of atomic-resolution electron tomography is to get a precise quantitative picture of a nanocrystal down to the atomic scale. To probe at such high resolutions, optimizing both the acquisition of projection images and the reconstruction of projection data is required.

One approach to atomic-resolution reconstruction is based on discrete tomography. In this approach, atoms are assumed to lie on a regular lattice and the measured projections are considered as atom counts along lattice lines. A key advantage of this approach is its ability to exploit the constraints induced by the discrete domain and range of the image. As a consequence, a small number of projection angles (typically less than 5) can already lead to an accurate reconstruction [16], [17]. A key drawback of the discrete lattice assumption is that in many interesting cases the atoms do not lie on a perfect lattice due to defects in the crystal structure or interfaces between different crystal lattices.

As an alternative, it has been demonstrated that a more conventional tomographic series consisting of hundreds of projections of a nanocrystal can be acquired in certain cases. An image of the nanocrystal is then reconstructed using sparsity-based reconstruction techniques on a continuous model of the tomography problem, typically solving the problem (1.16). This approach does not depend on the lattice structure and allows one to reconstruct defects and interfaces [18]. As a downside, the number of required projections is large and to accurately model the atom positions the reconstruction must be represented on a high-resolution pixel grid resulting in a large-scale computational problem. More importantly, increasing the resolution of the pixel grid in order to capture defects results in a much more ill-posed problem.

For the atomic-resolution reconstruction problem, a canonical discretization is provided not by an arbitrarily imposed pixel grid but by the spatial coordinates of atoms in the nanoparticle. Optimizing over a set of atom coordinates is nonconvex; a convex formulation proposed in the context of single-molecule localization microscopy [12] involves mapping the problem to the space of measures. In the space of measures, a set of atoms can be represented as a positive measure $\mu := \sum_{i=1}^{N_{\text{atoms}}} w_i \delta_{\boldsymbol{x}_i}$, with $\boldsymbol{x}_i$ being the spatial coordinates of atom i and $w_i \geq 0$ denoting weights that scale the intensity of atom i in projection data. Reconstructing the correct measure means solving

$$\underset{\mu}{\text{minimize}} \quad \|\Phi\,\mu - y\|_2^2, \tag{1.20}$$

where the forward model $\Phi\,\mu$ maps the measure to data y, such that

$$\Phi\,\mu := \mathcal{R}\Big(\sum_{i=1}^{N_{\text{atoms}}} w_i (G * \delta_{\boldsymbol{x}_i}) \Big),$$

where $\mathcal{R}$ is the continuous Radon transform and G denotes a known shape function. Sparsity can be included in the optimization problem in a number of ways: one way is by adding a term that minimizes the ℓ^1-norm of the weights $\{w_i\}$, another is by using a Frank–Wolfe-type algorithm [19] where the objective is minimized iteratively and only one atom is added to the support of the measure μ at each iteration.

In Chapter 3, we investigate grid-free algorithms to solve the reconstruction problem above. We demonstrate the advantages of using a grid-free approach to traditional grid-based reconstruction algorithms. We also show that including physical priors relevant to the problem – in this case, the potential energy of the atomic configuration – can help to resolve configurations with greater accuracy, especially in situations where the projection data are not enough to determine a unique atomic configuration.

Atomic configurations are modelled as sparse measures, whose support is the locations of atomic centres in continuous space.

1.4.3 Grid-free tilt-series alignment in cryoET

The goal of cryoET is to study the structure of biological macromolecules, such as proteins, in their native cellular context. Aspects of cryoET that distinguish it from other CT setups are as follows. Firstly, the geometry of the experimental system limits the extent to which the sample can be tilted. Moreover, the increase in apparent sample thickness with increasing tilt allows projection images to only be acquired for a limited angular range in cryoET, usually in $[-60°, 60°]$, resulting in a *missing wedge* of information that is not available during reconstruction [20]. Secondly, cryoET samples are dose-sensitive, which limits the total dose during acquisition and leads to very noisy projection images when a large number are acquired. Thirdly, the sample undergoes local and global movements during the acquisition procedure, making it difficult to reconstruct with a constant sample assumption.

Local deformation of the sample induced by the electron beam is a key resolution-limiting factor in cryoET. One way to align the tilt series is by using high-contrast gold beads as markers and modelling the deformation of markers using prior knowledge on sample deformation.

Extending the formalism of the previous section, the deforming marker configuration in cryoET can be mapped to a measure $\mu := \sum_{i=1}^{N_{\text{markers}}} w_i \delta_{\boldsymbol{x}_i}$ and the projection data at time t can be modelled using a forward model given by

$$\Phi_t \, \mu := \sum_{i=1}^{N_{\text{markers}}} w_i \, (G * \mathcal{R}) \, \delta_{\boldsymbol{x}_i + \boldsymbol{D}_t(\boldsymbol{P}, \boldsymbol{x}_i)}, \tag{1.21}$$

where $\boldsymbol{D}_t(\boldsymbol{P}, \boldsymbol{x}_i)$ denotes a deformation field with parameters $\boldsymbol{P}$. Tilt-series alignment then amounts to optimizing over the deformation parameters, marker locations and weights, and number of markers N_{markers}:

$$\underset{w_i, \boldsymbol{x}_i, \boldsymbol{D}_t, N_{\text{markers}}}{\text{minimize}} \quad \sum_{t=0}^{T} \left\| y_t - \sum_{i=1}^{N_{\text{markers}}} w_i \, (G * \mathcal{R}) \, \delta_{\boldsymbol{x}_i + \boldsymbol{D}_t(\boldsymbol{P}, \boldsymbol{x}_i)} \right\|_2^2. \tag{1.22}$$

We tackle this problem in Chapter 4 of this thesis, and show that our grid-free formulation allows the recovery of deformation parameters in synthetic and real data accurately despite the absence of labelled marker data as in existing approaches.

Gold-bead markers are modelled as sparse measures that deform over time according to a parametrized deformation field.

1.4.4 Cell–cell interaction learning for vascular network formation

In the final application area studied in this thesis, we look at the problem of inferring cell–cell interactions that are necessary for vascular network formation. To do this we adopt a method called Sparse Identification of Nonlinear Dynamics (SINDy) that has been shown to recover dynamical equations from time-series data [21].

The SINDy approach is applicable to ordinary differential equations of the type:

$$\dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x}), \tag{1.23}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ denotes the system state at a certain time and $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^n$ is a vector field that defines the dynamics of the system. Given measurements of $\boldsymbol{x}$ at a discrete set of time-points $\mathcal{T}$, and using computed values for $\dot{\boldsymbol{x}}$ at these time-points, the goal of SINDy is to recover the functional form of $\boldsymbol{g}$ from a library of functions. To do this, SINDy solves the following optimization problem:

$$\underset{\boldsymbol{\xi}\in\mathbb{R}^K}{\text{minimize}} \quad \|\dot{\boldsymbol{x}} - \Theta(\boldsymbol{x})\boldsymbol{\xi}\|_2^2 + \lambda\|\boldsymbol{\xi}\|_1 \,, \tag{1.24}$$

where $\Theta(\boldsymbol{x})$ denotes the library functions evaluated at the data points and $\boldsymbol{\xi}$ is the vector of coefficients that weights the library terms. Thus, SINDy optimizes for a sparse set of library terms that describes the measurements of a dynamical system.

SINDy has been used to infer the dynamics of simulated and real data for a variety of canonical systems exhibiting nonlinear dynamics [21]. Moreover, extensions of the SINDy approach have been used to investigate several problems of biological relevance. Two important examples are learning stochastic differential equations [22] and implicit ordinary differential equations describing biological networks [23].

In our case, the vascular network formation process is described by the dynamical equation (1.4) in particle-based simulations. In the overdamped regime, this translates to a form for $\boldsymbol{g}$ given by

$$g_i(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) := \sum_{j\in\mathcal{N}_i} \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|} F_{ij} \,, \quad i = 1, \ldots, n_p \tag{1.25}$$

$$F_{ij} := \Phi(\boldsymbol{x}_i, \boldsymbol{x}_j, \gamma_i, \gamma_j), \tag{1.26}$$

where we parametrize the force between cell pairs as a function Φ of cell locations $\boldsymbol{x}_i$ and orientations γ_i. The learning problem then amounts to learning a form for these cell–cell interactions F_{ij} from a library of functions.

In Chapter 5 we provide details of how this can be done, and apply our learning approach to simulation studies of vascular network formation. Our method can be extended to recover similar interaction terms from experimental data, and enables the discovery of effective equations from observations of a few variables.

Sparsity is used to constrain the number of terms in the inferred pairwise interaction between cells.

1.5 Research questions

To close this introductory chapter, we present the four research questions that were investigated in this thesis. Each of these research questions is presented on a separate page and is dealt with in a separate chapter. Here we provide a brief abstract of our method and main contributions, along with a representative illustration.

Research question 1. *Can sparse-basis minimum-residual filters be used to improve reproducibility in the synchrotron CT pipeline?*

In Chapter 2, we propose a filter-learning approach that reduces the quantitative differences between reconstructions obtained from popular open-source implementations. These differences are a result of differing software conventions for discretization and interpolation. We show that optimizing the filter in real-space and Fourier-space direct algorithms reduces such differences, resulting in fewer differences also in post-processing results. We apply our method to real data acquired at the synchrotron to validate the usability of our approach.

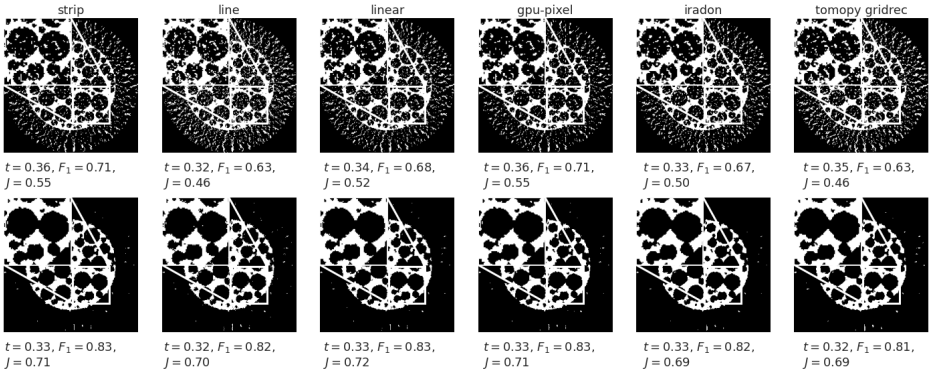


Figure 1.4: Differences between post-processing results after thresholding with Otsu's method. The top row shows thresholded reconstructions obtained using different back-projector implementations and a standard Shepp-Logan filter; Otsu thresholds t, F_1 scores and Jaccard indices are given for each image. The bottom rows shows thresholded reconstructions obtained using our implementation-adapted filters. Both qualitatively and quantitatively these results are more similar to each other than those in the top row.

Research question 2. *Can grid-free sparse reconstruction approaches infer the locations of defects in nanocrystals from very few projections?*

In Chapter 3, we turn to the atomic-resolution ET problem and propose a grid-free sparse optimization approach to tackle it. We also show how adding a physical potential energy term to the optimization objective helps to resolve atomic configurations from only two or three projections. We compare the performance of our method with that of existing grid-based methods such as SIRT and FISTA, as well as with that of nonconvex techniques like simulated annealing.

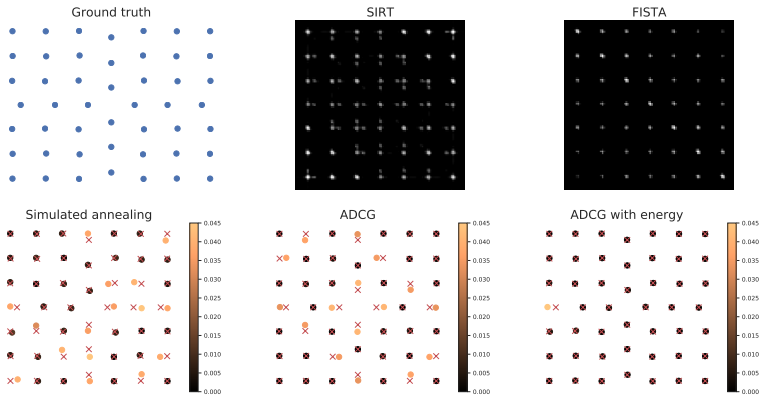
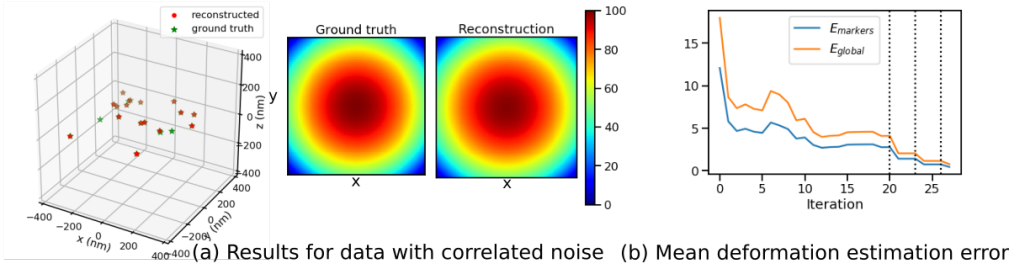


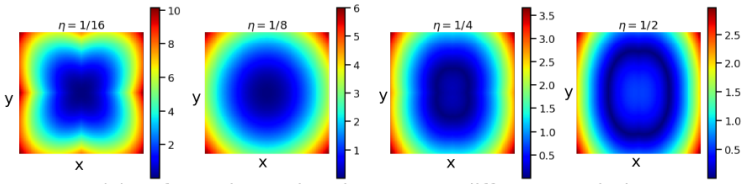
Figure 1.5: Reconstructions of a vacancy defect from three projections. For the simulated annealing, ADCG and ADCG with energy reconstructions, atoms are coloured according to their Euclidean distance from the ground truth. Ground truth positions are marked with red crosses.

Research question 3. *Can we extend grid-free sparse optimization methods to infer deformation parameters for cryoET alignment?*

In Chapter 4, we extend and adapt a grid-free algorithm to infer both locations and deformation parameters of gold markers in cryoET. We use globally supported parametrized deformation fields based on previous experimental studies to model beam-induced sample motion. The parameters of this model and marker locations are simultaneously inferred from our method, without the need for labelled marker data in each projection. We apply our method to TEM simulations as well as real data of gold beads on ice, and show that our method can estimate deformation fields in a host of noise and model mismatch settings.



(a) Results for data with correlated noise (b) Mean deformation estimation error



(c) Deformation estimation error at different resolutions

Figure 1.6: Inference of marker locations and deformation parameters from simulated TEM data with correlated noise. (a) Reconstructed and ground truth marker locations (left), and reconstructed and ground truth deformation fields in the direction of the electron beam (right). (b) Deformation estimation error as a function of iterations. (c) Deformation estimation errors in the beam direction

Research question 4. *Can sparse equation learning recover cell–cell interactions from simulated time-series data of vascular network formation?*

In Chapter 5, we adapt a sparse equation-learning approach to infer which pairwise interaction terms contribute to vascular network formation. We run particle-based simulations of network formation to generate cell trajectories over time. We formulate the time evolution of the system to be given by an overdamped Langevin equation with force terms that correspond to the pairwise interactions between cells. These force terms are then inferred from the cell trajectory data from a library of plausible forces.

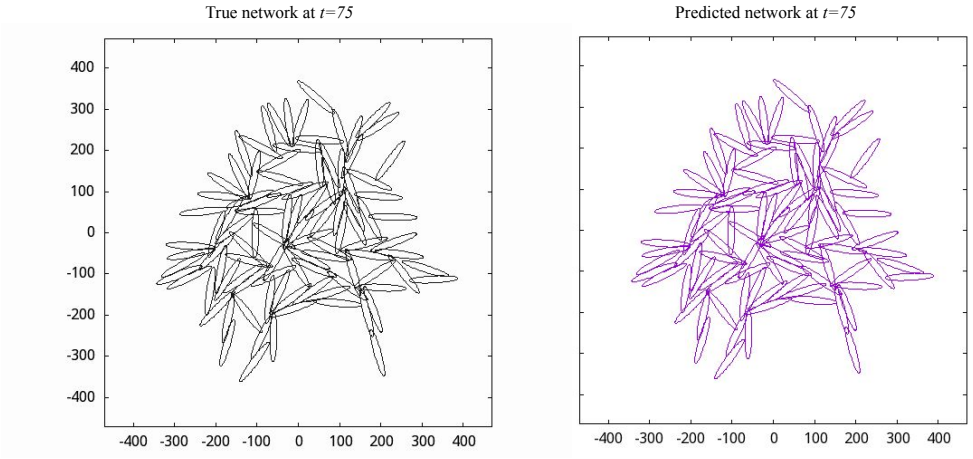


Figure 1.7: True and inferred vascular networks using 100 elongated cells.

Chapter 2

Implementation-adapted filters for synchrotron tomography

2.1 Introduction

In several scientific disciplines, such as materials science, biomedicine and engineering, a quantitative three-dimensional representation of a sample of interest is crucial for characterizing and understanding the underlying system [24]–[27]. Such a representation can be obtained with the experimental technique of computerized tomography (CT). In this approach, a penetrating beam, such as X-rays, is used to obtain projection images of a sample at various angles. These projections are then combined by using a computational algorithm to give a 3D reconstruction [28], [29].

Different tomographic setups are used in various practical settings. Our focus here is on tomography performed with a *parallel-beam* X-ray source at synchrotrons. Synchrotrons provide a powerful source of X-rays for imaging, enabling a broad range of high-resolution and high-speed tomographic imaging techniques [30]–[32].

A typical tomography experiment at the synchrotron can be described by a pipeline consisting of several sequential steps (see Fig. 2.1). First, a sample is prepared according to the experiment and imaging setup requirements. Then, the imaging system is aligned [33], and a series of projection images of the sample are acquired [34]. These data are then processed for calibration, contrast improvement (e.g. phase retrieval [35]) or removal of undesirable artefacts like rings or stripes [36]. Following pre-processing, the data are fed into a reconstruction software package that makes use of one or more standard algorithms to compute a 3D reconstruction [37], [38]. The reconstruction volumes can then be further post-processed and analysed [39], [40] to obtain parameter estimates of the system being

This chapter is based on:

Improving reproducibility in synchrotron tomography using implementation-adapted filters. *P. S. Ganguly, D. M. Pelt, D. Gürsoy, F. de Carlo, and K. J. Batenburg*. Journal of Synchrotron Radiation 28, no. 5, 2021.

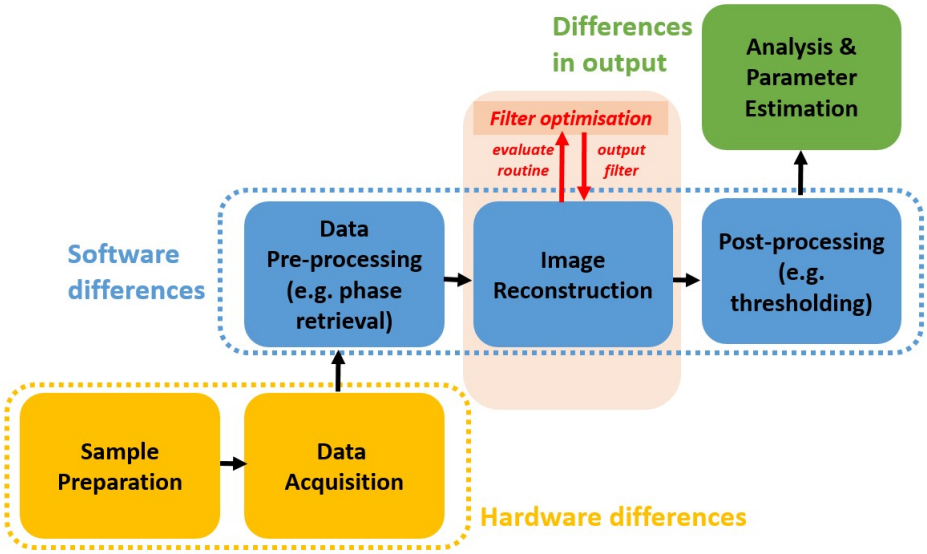


Figure 2.1: Schematic representation of a typical tomography pipeline at synchrotrons. Hardware differences play an important role during sample preparation and data acquisition. Software differences affect image pre-processing, reconstruction and post-processing. Together these lead to differences in the output of analysis and parameter estimation studies. In this chapter we propose a filter optimization method that works as a wrap-around routine on the reconstruction block. Our method only requires evaluations of the reconstruction routine and does not require any internal coding. The output of our method is a filter that can be used in the reconstruction block for more reproducible reconstructions.

studied. In some cases, systematic imperfections in the data can also be corrected by post-processing reconstructions. For example, ring artefacts, which are commonly observed in synchrotron data, can be corrected before or after reconstruction [37].

At various synchrotron facilities in the world, the pipeline described above is implemented using different instruments, protocols and methods specific for each facility [41]. These differences are on the level of both hardware and software. Dissimilarities in the characteristics of the used X-ray source and detection system, including camera, visible light objective and scintillator screen, lead to differences in the acquired data. The differences in the data are then further compounded by variations in processing and reconstruction software, resulting in differences in voxel or pixel intensities, and eventually in variations in the output of post-processing and analysis routines.

For users, such differences pose several challenges. First, it is difficult to ensure that results and conclusions obtained from experiments at one facility are comparable and consistent with experiments from another facility. Second, other researchers seeking to reproduce the results of a previous work with their own software might not be able to do so, even if they have access to raw data. In [41], the authors report quantitative

differences at various stages of the pipeline when scanning the same object at different synchrotrons. Reproducibility and the ability to verify experimental findings is crucial for ascertaining the reliability of scientific results. Therefore, in order to ensure reproducibility for the synchrotron pipeline, it is important to quantify and mitigate differences in the acquired, processed and reconstructed data.

Hardware and software vary across synchrotrons for a number of reasons. Each synchrotron uses a pipeline that is optimized for its specific characteristics. In addition, legacy considerations play a role in the choice of components. Because of the variations across synchrotrons, any successful strategy for creating reproducible results must take this diversity into account. Ideally, the choices for specific implementations of each block in the synchrotron pipeline in Fig 2.1 should not influence the final results of a tomography experiment. Following this strategy, each block can be optimized for reproducibility independently from the rest of the pipeline.

In this chapter, we focus on improving the reproducibility of the reconstruction block in the pipeline. In most synchrotrons, fast analytical methods such as filtered backprojection (FBP) [29] and Gridrec [42] are the most commonly-used algorithms for reconstruction. This is primarily because such algorithms are fast and work out-of-the-box without parameter tuning. These algorithms give accurate reconstructions when the projection data are well-sampled, such as in microCT beamlines where thousands of projections can be acquired in a relatively short time.

Several open-source software packages for synchrotron tomography reconstruction are available, such as TomoPy, the ASTRA toolbox and scikit-image [37], [43], [44]. Usually, an in-house implementation of FBP or Gridrec, or one of the open-source software packages is used for reconstruction. Each of these implementations contains a *filtering* step that is applied to the projection data as part of the reconstruction. Filtering influences characteristics, such as noise and smoothness, of the reconstructed volume. A sample-independent, pre-defined filter is generally used for reconstruction. Some filters used in this step have tunable parameters, but these are often tuned on-the-fly and are not recorded in metadata.

Reconstructions in analytical algorithms are obtained by inversion of the Radon transform [45]. Although this inversion is well-defined mathematically in a continuous setting, software implementations invariably have to work in a discretized space. In software implementations, the measurements as well as the reconstructed volume are *discrete*. In a discretized space, inversion of the Radon transform often translates to a *backprojection* step, which makes use of a discretized *projection kernel* to simulate the intersection between the scanned object and X rays [46]. The backprojection operation can also be performed directly using interpolations in Fourier space [29].

Different choices of discretization and interpolation, in projection kernels and filters, are possible. These choices lead to quantitative differences between the reconstructions obtained from different software implementations. A simple example of this effect is shown in Fig. 2.2, where we consider a phantom of pixel size 33×33 and data along 8 projection angles uniformly sampled in $[0, \pi)$. We compare reconstructions of the same data using two different projection kernels and two different filtering methods. In both instances, the image to be reconstructed contains a single bright pixel at the centre of the field-of-view. The *sinogram* of such an image (i.e. the combined projection data for the

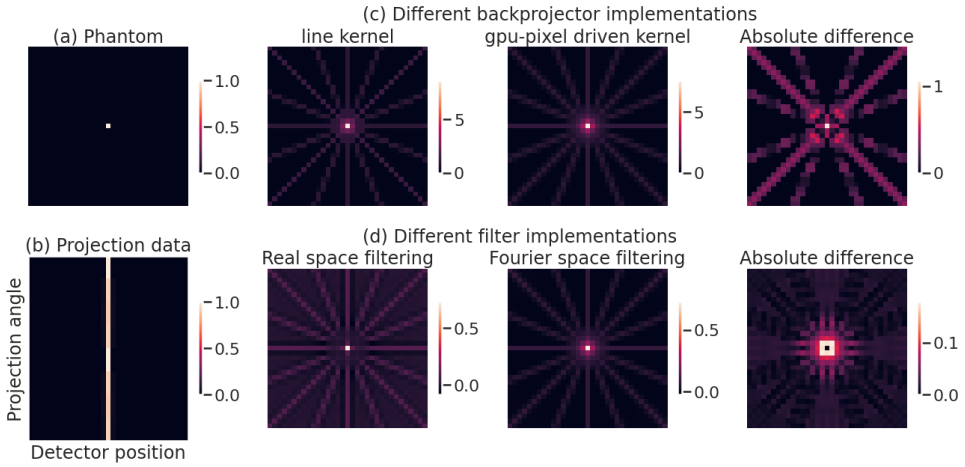


Figure 2.2: Differences in reconstruction due to differences in backprojector and filter implementations. (a) a 33×33 phantom with one bright pixel, (b) sinogram of the phantom (computed using a strip kernel from the ASTRA toolbox), (c) differences in (unfiltered) backprojection when using different backprojectors: (*left to right*) backprojection using a CPU line kernel from the ASTRA toolbox, backprojection using a GPU pixel-driven kernel from the ASTRA toolbox, absolute difference between the two backprojections. (d) differences in reconstruction when using different filtering routines in FBP with the `gpu-pixel` kernel as backprojector: (*left to right*) reconstruction using filtering in real space with the Ram-Lak filter, reconstruction using the ramp filter in Fourier space, absolute difference between the two reconstructions.

full range of angles) was computed using a CPU strip kernel projector from the ASTRA toolbox [43]. Backprojections of this projection data using two other projectors - a CPU line projection kernel and a pixel-driven kernel implemented on a graphics processing unit (GPU) - show significant, radially-symmetric differences. These differences are dependent on the number of projection angles used, and are highly structured, unlike differences due to random noise. We also observe structured differences between reconstructions when the same projection kernel (`gpu-pixel`) is used after different filtering operations in real and Fourier space. This example highlights the impact of discretization and interpolation choices on the final reconstruction obtained from identical raw data.

Our main contribution in this chapter is a heuristic approach that can improve reproducibility in reconstructions. Our method consists of optimizing the filter used in different software implementations of reconstruction methods. We call such optimized filters *implementation-adapted filters*. The computation of our filters does not require knowledge of the underlying software implementation of the reconstruction algorithm. Instead, a wrapper routine around any black-box implementation can be used for filter computation. Once computed, these filters can be applied with the reconstruction software like any other standard filter.

Our chapter is organized as follows. In Section 5.2, we formulate the reconstruction

problem mathematically and discuss the effect of different software implementations. In Section 2.3, we describe our algorithm for computing implementation-adapted filters. Numerical experiments described in Sections 2.4 and 4.5 demonstrate use cases for our filters on simulated and real data. Finally, we discuss extensions to the current work in Section 5.5 and conclude our chapter in Section 4.6. Our open-source Python code for computing implementation-adapted filters is available on GitHub (https://github.com/poulamisganguly/impl-adapted-filters).

2.2 Background

2.2.1 Continuous reconstruction

Consider an object described by a two-dimensional attenuation function $f : \mathbb{R}^2 \to \mathbb{R}$. Mathematically, the tomographic projections of the object can be modelled by the Radon transform, $\mathcal{R}(f)$. The Radon transform is the line integral of f along parametrized lines $l_{\theta,t} = \{(x, y) \in \mathbb{R}^2 \,|\, x\cos\theta + y\sin\theta = t\}$, where θ is the projection angle and t is the distance along the detector. Projection data $p_\theta(t)$ along an angle θ are thus given by

$$p_\theta(t) = \mathcal{R}(f) = \iint_{\mathbb{R}^2} f(x, y)\delta(x\cos\theta + y\sin\theta - t)dxdy. \tag{2.1}$$

The goal of tomographic reconstruction is to obtain the function $f(x, y)$ given the projections $p_\theta(t)$ for various angles $\theta \in \Theta$. One way to achieve this is by direct inversion of the Radon transform. Given a complete angular sampling in $[0, \pi)$, the Radon transform can be inverted giving the following relation [29]

$$f(x, y) = \int_0^\pi \left(\int_{-\infty}^{\infty} \tilde{P}_\theta(\omega)|\omega| e^{2\pi i\omega(x\cos\theta + y\sin\theta)} d\omega \right) d\theta, \tag{2.2}$$

where $\tilde{P}_\theta(\omega)$ denotes the Fourier transform of the projection data $p_\theta(t)$ and multiplication by the absolute value of the frequency $|\omega|$ denotes filtering with the so-called ramp filter.

For noiseless and complete data, the Radon inversion formula (2.2) provides a perfect analytical reconstruction of the function $f(x, y)$ from its projections. However, in practice, tomographic projections are obtained on a *discretized* detector, consisting of individual pixels, and for a finite set of projection angles. Additionally, the reconstruction volume must be discretized in order to represent it on a computer. Therefore, in practical applications, a discretized version of (2.2) is used to obtain reconstructions.

2.2.2 Discrete reconstruction

Discretization of the reconstruction problem yields the following equation for the discrete reconstruction $r(x_d, y_d)$:

$$r(x_d, y_d) = \sum_{\theta_d \in \Theta} \sum_{t_d \in T} h(t_d) P_{\theta_d}(x_d\cos\theta_d + y_d\sin\theta_d - t_d), \tag{2.3}$$

where (x_d, y_d), θ_d and t_d denote discretized reconstruction pixels, angles and detector positions, respectively, and $h(t_d)$ is a discrete real-space filter. This inversion formula is known as the filtered backprojection (FBP) algorithm.

The FBP equation (2.3) can be written algebraically as the composition of two matrix operations: filtering and backprojection. Filtering denotes convolution in real space (or, correspondingly, multiplication in Fourier space) with a discrete filter. Backprojection consists of a series of interpolation and numerical integration steps to sum contributions from different projection angles. These discretized operations can be implemented in a number of different ways and different software implementations often make use of different choices for discretization and interpolation. Consequently, the reconstruction obtained from a particular implementation is dependent on these choices. The reconstruction $\boldsymbol{r}_I$ from an implementation I can thus be written as

$$\boldsymbol{r}_I(\boldsymbol{h}, \boldsymbol{p}) = \boldsymbol{W}_I^T \boldsymbol{M}_I(\boldsymbol{h}, \boldsymbol{p}), \tag{2.4}$$

where $\boldsymbol{W}_I^T$ is the backprojector and $\boldsymbol{M}_I(\cdot, \cdot)$ is the (linear) filtering operation associated with implementation I. We denote the discrete filter by $\boldsymbol{h}$.

In the following subsection, we discuss some common choices for projection and filtering operators in software implementations of analytical algorithms.

2.2.3 Differences in projectors and filtering

In order to discretize the Radon transform, we must choose a suitable discretization of the reconstruction volume, a discretization of the incoming ray and an appropriate numerical integration scheme. All these choices contribute to differences in different backprojectors $\boldsymbol{W}_I^T$ in (2.4).

Voxels (or pixels in 2D) in the reconstruction volume can be considered either to have a finite size or to be spikes of infinitesimal size. Similarly, a ray can be discretized to have finite width (i.e. a strip) or have zero width (i.e. a line). The numerical integration scheme chosen might be piecewise constant, piecewise linear or continuous. All of these different choices have given rise to different software implementations of backprojectors [46]. There exist different categorizations of backprojectors in the literature; for example, the `linear` kernel in the ASTRA toolbox is referred to as the slice-interpolated scheme in [47] and the `strip` kernel is referred to as the box-beam integrated scheme in the same work. In this chapter, we designate different backprojectors with the terms used in the software package where they have been implemented.

In addition to the choices mentioned above, backprojectors have also been optimized for the processing units on which they are used. For this reason, backprojectors that are optimized to be implemented on graphics processing units (GPUs) might be different from those that are implemented on a CPU due to speed considerations. In particular, GPUs provide hardware interpolation that is extremely fast, but can also be of limited accuracy compared to standard floating point operations.

So far, we have discussed real space backprojectors. Fourier-domain algorithms such as Gridrec [42] use backprojectors that operate in the Fourier domain. These operators are generally faster than real-space operators, and are therefore particularly suited for accelerating iterative algorithms [48]. Unlike real space backprojectors, Fourier-space

backprojectors perform interpolation in the Fourier domain. As this might lead to non-local errors in the reconstruction, an additional filtering step is performed to improve the accuracy of the interpolation.

Apart from differences in backprojectors, different implementations also vary in the way they perform the filtering operation in analytical algorithms. Filtering can be performed as a convolution in real space or as a multiplication in Fourier space. Real space filtering implementations can differ from each other in computational conventions, for example by the type of padding used [13] to extend the signal at the boundary of the detector. Moreover, the zero-frequency filter component is treated in different ways between implementations. For example, the Gridrec implementation in TomoPy sets the zero-frequency component of the filter to zero.

2.3 Implementation-adapted filters

We now present the main contribution of our chapter. In order to mitigate the differences between implementations discussed in the previous section, we propose to specifically tune the filter $\boldsymbol{h}$ for each implemented analytical algorithm. In the following, we describe an optimization scheme for the filter, which helps us to reduce the differences between reconstructions from various implementations.

We optimize the filter by minimizing the ℓ^2 difference with respect to the projection data $\boldsymbol{p}$. This can be stated as the following optimization problem over filters $\boldsymbol{h}$:

$$\boldsymbol{h}_I^* = \arg\min_{\boldsymbol{h}} \|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{r}_I(\boldsymbol{h}, \boldsymbol{p})\|_2^2, \tag{2.5}$$

where $\boldsymbol{r}_I$ is the reconstruction from implementation I. Note that the forward projector $\boldsymbol{W}$ used above is chosen as a fixed operator in our method (the same for each implementation for which the filter is optimized) and does not have to be the transpose of the implementation-specific backprojection operator $\boldsymbol{W}_I^T$. In order to improve stability and take additional prior knowledge of the scanned object into account, a regularization term can be added to the objective in (2.5).

The solution to the optimization problem above is a implementation-adapted filter $\boldsymbol{h}_I^*$. Once the filter has been computed, it can be used in (2.4) to give an optimized reconstruction:

$$\boldsymbol{r}_I^* = \boldsymbol{W}_I^T \boldsymbol{M}_I(\boldsymbol{h}_I^*, \boldsymbol{p}).$$

Out of all reconstructions that an implemented algorithm can produce for a given dataset $\boldsymbol{p}$ by varying the filter, this reconstruction, $\boldsymbol{r}_I^*$, is the one that results in the smallest residual error. Such filters are known as minimum-residual filters and have previously been proposed to improve reconstructions of real-space analytical algorithms in low-dose settings [14], [15].

Our implementation-adapted filters are thus minimum-residual filters that have been optimized to each implementation I. The main difference between the previous works [14], [15] and our present study is that we use a fixed forward operator in our optimization problem, which is not necessarily the transpose of the backprojection operator. More importantly, our goal in this chapter is not the improvement of reconstruction accuracy,

but the reduction of differences in reconstruction between various software implementations.

We hypothesize that such minimum-residual reconstructions obtained using different implementations are closer (quantitatively more similar) to each other than reconstructions obtained using standard filters. As an example for motivating this choice, let's take an implementation of an analytical algorithm from both TomoPy and the ASTRA toolbox. Given a certain dataset, changing the reconstruction filter results in different reconstructed images, each with a different residual error. Even though the implementations used by TomoPy and ASTRA are fixed, the freedom in choosing a filter gives us an opportunity to reduce the difference between reconstructions from both implementations. Tuning the filter is a way to *optimize* the reconstruction according to user-selected quality criteria. Choosing the *minimum-residual* reconstruction for each implementation results in reconstructions that are the *closest possible* to each other in terms of data misfit. Closeness in data misfit, under convexity assumptions, indicates closeness in pixel intensity values of reconstruction images. Hence, the minimum-residual reconstructions for the two implementations are closer to each other than reconstructions with standard filters offered by the implementations.

To compute the optimized filter (2.5), we use the fact that the reconstruction $\boldsymbol{r}_I(\boldsymbol{h}, \boldsymbol{p})$ of data $\boldsymbol{p}$ obtained from an implementation of FBP or Gridrec is *linear* in the filter $\boldsymbol{h}$. This means that we can write the reconstruction as

$$\boldsymbol{r}_I(\boldsymbol{h}, \boldsymbol{p}) = \boldsymbol{R}_I(\boldsymbol{p})\boldsymbol{h},$$

where $\boldsymbol{R}_I(\boldsymbol{p})$ is the reconstruction matrix of implementation I given projection data $\boldsymbol{p}$. Thus, the optimization problem (2.5) becomes

$$\boldsymbol{h}_I^* = \arg\min_{\boldsymbol{h}} \|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{R}_I(\boldsymbol{p})\boldsymbol{h}\|_2^2 =: \arg\min_{\boldsymbol{h}} \|\boldsymbol{p} - \boldsymbol{F}_I(\boldsymbol{p})\boldsymbol{h}\|_2^2 \tag{2.6}$$

The matrix $\boldsymbol{F}_I(\boldsymbol{p})$ has dimensions $N_p \times N_f$, where N_p is the size of projection data and N_f is the number of filter components. For a filter that is independent of projection angle, the number of filter components, N_f, is equal to the number of discrete detector pixels, N_d. The projection size $N_p := N_d N_\theta$, where N_θ is the number of projection angles. $\boldsymbol{F}_I(\boldsymbol{p})$ can be constructed explicitly by assuming a basis for filter components. A canonical basis can be formed using N_d unit vectors $\{\boldsymbol{e}_i, i = 1, 2, \ldots, N_d\}$, such that

$$\boldsymbol{e}_1 = \begin{pmatrix} 1 \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix}, \quad \boldsymbol{e}_2 = \begin{pmatrix} 0 \\ 1 \\ . \\ . \\ . \\ 0 \end{pmatrix}, \quad \ldots \quad \boldsymbol{e}_{N_d} = \begin{pmatrix} 0 \\ 0 \\ . \\ . \\ . \\ 1 \end{pmatrix}.$$

Using these basis filters, each column of $\boldsymbol{F}_I(\boldsymbol{p})$ can be computed by reconstructing $\boldsymbol{p}$ using the implementation I, followed by forward projection with $\boldsymbol{W}$:

$$\boldsymbol{f}_j = \boldsymbol{W}\boldsymbol{r}_I(\boldsymbol{e}_j, \boldsymbol{p}), \quad j \in \{1, 2, \ldots, N_d\}$$
$$\boldsymbol{F}_I(\boldsymbol{p}) = \begin{pmatrix} \boldsymbol{f}_1 & \boldsymbol{f}_2 & \boldsymbol{f}_3 & \ldots & \boldsymbol{f}_{N_d} \end{pmatrix}$$

We can then substitute for $\boldsymbol{F}_I(\boldsymbol{p})$ in (2.6) and solve for the optimized filter $\boldsymbol{h}_I^*$. Note that our method only requires *evaluations* of the implementation I by using it as a black-box routine to compute the reconstructions $\boldsymbol{r}_I(\boldsymbol{e}_j, \boldsymbol{p})$ above. In other words, no knowledge of the implementation I or any internal coding is required.

If we expand the filter in a basis of unit vectors, $\mathcal{O}(N_p)$ reconstructions using the implementation I and $\mathcal{O}(N_p)$ forward projections with $\boldsymbol{W}$ must be performed for filter optimization. In contrast, the complexity of a standard FBP reconstruction is of the order of a single backprojection. Choosing a smaller set of suitable basis functions would result in a reduction in the number of operations for filter optimization and, consequently, faster filter computations. One way to do this is by exponential binning [14].

The idea of exponential binning is to assume that the real-space filter is a piecewise constant function with N_b bins, where $N_b < N_d$. The bin width w_i, for $i = 1, 2, \ldots, N_b$, is assumed to increase in an exponential fashion away from the centre of the detector, such that:

$$w_i = \begin{cases} 1, & |i| < N_l \\ 2^{|i|-N_l}, & |i| \geq N_l \end{cases}, \tag{2.7}$$

where N_l is the number of large bins with width 1. Exponential binning is inspired by the observation that standard filters used in tomographic reconstruction, such as the Ram-Lak filter, are peaked at the centre of the detector and decay to zero relatively quickly towards the edges. Binning results in a reduction of free filter components from N_d to N_b. Moreover, despite the reduction in components, it does not typically result in a significant change in reconstruction quality [14].

The pseudocode for our filter computation method is shown in Algorithm 1. Here we give further details of the routines used in the algorithm. The `filter` routine performs filtering in the Fourier domain, which is equivalent to multiplication by the filter followed by an inverse Fourier transform. The $\texttt{reconstruct}_I$ routine calls the function for reconstruction in implementation I with the internal filtering disabled. Finally, the `lstsq` routine calls a standard linear least squares solver in NumPy [49] to compute filter coefficients.

Algorithm 1 Implementation-adapted filter computation

procedure Compute filter($\boldsymbol{p}$, I, $\boldsymbol{W}$):
    Create filter basis: $\mathcal{B} := \{b_1, b_2, \ldots, b_{N_b}\}$
    Compute columns of $\boldsymbol{F}_I(\boldsymbol{p})$:
    for $\boldsymbol{b}_j \in \mathcal{B}$ **do**
        Filter data with basis filter: $\boldsymbol{q} \leftarrow \texttt{filter}(\boldsymbol{p}, \boldsymbol{b}_j)$
        Reconstruct filtered projection with I: $\boldsymbol{r} \leftarrow \texttt{reconstruct}_I(\boldsymbol{q})$
        Forward project reconstruction $\boldsymbol{f}_j \leftarrow \texttt{flatten}(\boldsymbol{W}\boldsymbol{r})$
    end for
    Linear least squares fitting of filter coefficients: $\boldsymbol{c} \leftarrow \texttt{lstsq}(\boldsymbol{F}_I(\boldsymbol{p}), \boldsymbol{p})$
    Return filter: $\boldsymbol{h}^* \leftarrow \sum_{j=1}^{N_b} c_j \boldsymbol{b}_j$
end procedure

Once a filter $\boldsymbol{h}^*$ is computed, we can store it in memory, either as a filter in Fourier

space or as a filter in real space after computing the Fourier transform of $\boldsymbol{h}^*$. Using the filter with a black-box software package involves calling the `filter` routine with the data and the computed filter as arguments, followed by one call of the $\texttt{reconstruct}_I$ routine in a chosen algorithm (with its internal filtering disabled). Thus, the complexity of a reconstruction using a computed implementation-adapted filter is the same as that of a reconstruction run using a standard filter.

In the following sections, we describe numerical experiments and the results of filter optimization on reconstructions.

2.4 Data and metrics

We performed a range of numerical experiments on real and simulated data to quantitatively assess (i) the effect of our proposed optimized filters on the variations between reconstructions from different implementations; (ii) the behaviour and dependence of our proposed filters on acquisition characteristics such as noise and sparse angular sampling; and (iii) the effect of our proposed filters on post-processing steps following the reconstruction block in Fig 2.1. In this section, we describe the software implementations used, data generation steps and the metric used to quantify intra-set variability of reconstructions.

2.4.1 Software implementations of analytical algorithms

We optimized filters to commonly used software implementations of FBP and Gridrec. For FBP, we considered different projector implementations in the ASTRA toolbox [43] as well as the `iradon` backprojection function in scikit-image [44]. These implementations use different choices of volume and ray discretization as well as numerical integration schemes. From the ASTRA toolbox, we considered projectors implemented on the CPU (`strip`, `line` and `linear`) as well as a pixel-driven kernel on the GPU (`gpu-pixel`, called `cuda` in the ASTRA toolbox). For Fourier-space methods, we considered the Gridrec implementation in TomoPy. We used the ASTRA `strip` kernel as the forward projector $\boldsymbol{W}$ in (2.5) during filter computations.

2.4.2 Projection data

We performed experiments with both simulated and real data. Both data consisted of projections acquired in a parallel-beam geometry along a complete angular range in $[0, \pi)$.

Simulated foam phantom data

Simulated data of foam-like phantoms were generated using the foam_ct_phantom package in Python. This package generates 3D volumes of foam-like phantoms by removing, at random, a pre-specified number of non-overlapping spheres from a cylinder of a given material [50]. The simulated phantoms are representative of real foam samples used in tomographic experiments and are challenging to reconstruct due to the presence of features at different length scales. At the same time, the phantoms are amenable to experimentation as data in different acquisition settings can be easily generated. Slices of one such

phantom, which we used for the experiments in this chapter, are shown in Fig. 2.3 and Fig. 2.5.

Ray tracing through the volume is used to generate projection data from a 3D foam phantom. To simulate real-world experimental setups, where detector pixels have a finite area, ray supersampling can be used. This amounts to averaging the contribution of n neighbouring rays within a single pixel, where n is called the supersampling factor.

For our experiments, we generated a 3D foam with 1000 non-overlapping spheres with varying radii. A parallel beam projection geometry, in line with synchrotron setups, was used to generate projection data. We used ray supersampling with a supersampling factor of 4, and each 2D projection was discretized on a pixel grid of size 256×256. We varied the number of projection angles, N_θ, in our experiments in order to determine the effect of sparse sampling ranges on our filters.

Poisson noise was added to noiseless data by using the `astra.add_noise_to_sino` function in the ASTRA toolbox [43]. This function requires the user to specify a value for the photon flux I_0. In an image corrupted with Poisson noise, each pixel intensity value k is drawn from a Poisson distribution

$$f_{\text{Pois}}(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!},$$

with $\lambda \propto I_0$. High photon counts (and high values of λ) correspond to low noise settings. All noise realizations in our experiments were generated with a pre-specified random seed.

Real data of shale

In order to validate the applicability of our method to real data, we performed numerical experiments using microCT data of the Round-Robin shale sample N1 from the tomographic data repository Tomobank [51]. We used data acquired at the Advanced Photon Source (APS) for our experiments. The Round-Robin datasets were acquired for characterizing the porosity and microstructures of shale, and the same sample has been imaged at different synchrotrons (using the same experimental settings) for comparison of results [41]. The dataset we used was acquired with a 10x objective lens and had an effective pixel size of approximately 0.7μm. Each projection in the dataset had pixel dimensions 2048×2048, and data were acquired over 1500 projection angles. In order to simulate sparse angular range settings, we removed projections at intervals of $m = 2, 3, 4, 5$ and 10 from the complete data.

2.4.3 Quantitative metrics

Reconstructions of a 3D volume from parallel beam data can be done slice-wise, because data in different slices (along the rotation axis) are independent of each other in a parallel beam geometry. Therefore, all our quantitative metrics were computed on individual slices. Reconstructed slices of the simulated foam phantom were discretized on a pixel grid of size 256×256. Reconstruction slices of the Round-Robin dataset were discretized on a pixel grid of size 2048×2048. All CPU reconstructions were performed on an Intel(R)

Core(TM) i7-8700K CPU with 12 cores. GPU reconstructions were performed on a single Nvidia GeForce GTX 1070 Ti GPU with CUDA version 10.0.

We were interested in comparing the similarity between reconstructions in a *set* of images, without having a reference reconstruction. We quantified the intra-set variability between reconstruction slices obtained from different implementations using the pixelwise standard deviation between these. For a set of reconstruction slices $\{\boldsymbol{r}_I, I \in \mathcal{I}\}$ obtained using different implementations I, the standard deviation of a pixel j is given by:

$$\sigma_j = \sqrt{\frac{1}{N_I} \sum_{I \in \mathcal{I}} \Big((r_I)_j - \bar{r}_j \Big)^2}; \qquad \bar{r}_j = \frac{1}{N_I} \sum_{I \in \mathcal{I}} (r_I)_j, \tag{2.8}$$

where $(r_I)_j$ is the intensity value of pixel j in reconstruction $\boldsymbol{r}_I$ and N_I is the total number of implementations.

In our experiments, we reconstructed the same data using our set of implementations $\{I \in \mathcal{I}\}$, by using the Ram-Lak filter and the Shepp-Logan filter as defined in different packages, and then by using filters $\{\boldsymbol{h}_I^*, I \in \mathcal{I}\}$ (2.5) that were optimized to those implementations. As a result, we achieved three sets of reconstructions: one set using the Ram-Lak filter, a second set using the Shepp-Logan filter and a third set using the implementation-adapted filters. We computed the pixelwise standard deviation (2.8) over slices for all sets.

The mean standard deviation of a slice S (with dimensions $N \times N$) is defined as the mean of pixelwise standard deviations in that slice:

$$\bar{\sigma}^S = \frac{1}{N^2} \sum_{j \in J^S} \sigma_j, \tag{2.9}$$

where J^S is the list of pixels in slice S.

In addition to the mean, the histogram of standard deviations (2.8) provides important information about the distribution of standard deviation values in a slice. The *mode* of this histogram is the value of standard deviation that occurs most, and the tail of the histogram indicates the number of large standard deviations observed. For reconstructions that are more similar to each other, we would expect the histogram to be peaked at a value close to 0 and have a small tail.

In order to quantify the difference between a reconstruction slice and the ground truth (in experiments where a ground truth was available), we used the root mean squared error (RMSE) given by

$$\text{RMSE}(\boldsymbol{r}_I) = \sqrt{\frac{1}{N^2} \sum (\boldsymbol{r}_I - \boldsymbol{r}_{gt})^2}, \tag{2.10}$$

where $\boldsymbol{r}_{gt}$ is the ground truth reconstruction. For a set of reconstructions we used the squared bias defined below to quantify the difference with respect to the ground truth:

$$\Big(\text{bias}(\{\boldsymbol{r}_I, I \in \mathcal{I}\}) \Big)^2 = \Big(\bar{\boldsymbol{r}} - \boldsymbol{r}_{gt} \Big)^2, \tag{2.11}$$

where $\bar{\boldsymbol{r}} := \sum_{I \in \mathcal{I}} \frac{1}{N_I} \boldsymbol{r}_I$ is the mean over the set of reconstructions. The squared bias, similar to the standard deviation in (2.8) is a pixelwise measure. The mean squared bias over a slice S is obtained by taking the mean of (2.11) over all pixels in the slice.

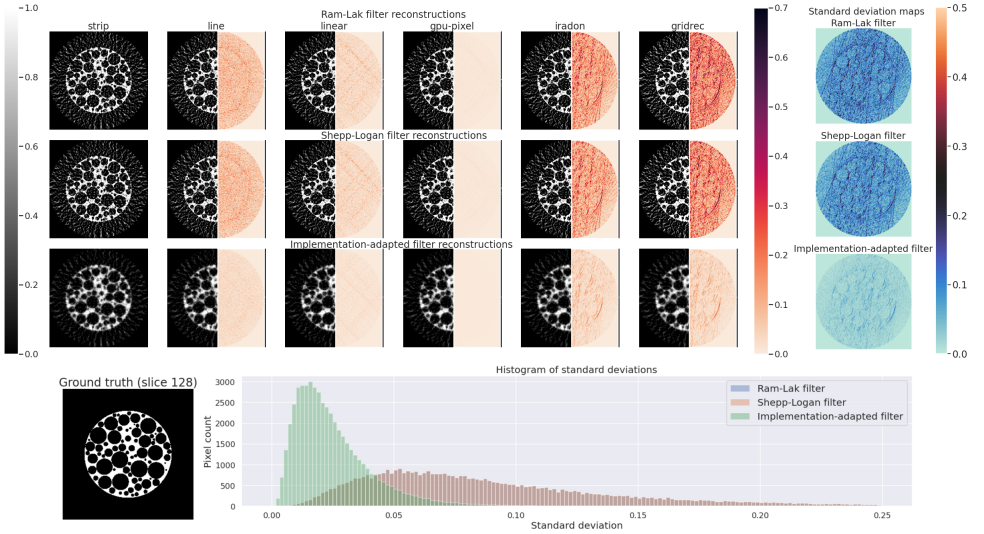


Figure 2.3: Reduction in intra-set variability between reconstructions of simulated foam data ($N_\theta = 32$, no noise) by using implementation-adapted filters. (*top three rows*) Reconstructions of the central slice (slice no. 128) of a foam phantom. To highlight intra-set discrepancies we show the absolute difference with respect to the corresponding `strip` kernel reconstructions in the right half of each image. The rightmost column shows pixelwise standard deviation σ in each set. (*bottom row, left*) Ground truth foam phantom slice. (*right*) Histograms of standard deviations σ for all three sets. The Ram-Lak filter and Shepp-Logan filter histograms overlap.

In our experiments, we also quantify the effect of filter optimization on later post-processing steps after reconstruction. To do this, we threshold a set of reconstructions using Otsu's method [52], which picks a single threshold to maximize the variance in intensity between binary classes. To quantify the accuracy of the resulting segmentations and to compare the similarity in a set we used two standard metrics for segmentation analysis: the F_1 score and the Jaccard index. The F_1 score takes into account false positives (fp), true positives (tp) and false negatives (fn) in binary segmentation and is given by:

$$F_1 = \frac{\mathrm{tp}}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})}. \tag{2.12}$$

The Jaccard index is the ratio between the intersection and union of two sets A and B. In our case, one set is the segmented binary image and the other set is the binary ground truth image:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{2.13}$$

2.5 Numerical experiments and results

In this section, we give details of our numerical experiments and discuss their results.

2.5.1 Foam phantom data

Reduction in differences between reconstructions

Fig. 2.3 shows the central (ground truth) slice of the foam phantom. Data along $N_\theta = 32$ angles were reconstructed using all implementations using the Ram-Lak filter, the Shepp-Logan filter and our implementation-adapted filters. Reconstructions using the various filters are shown in Fig. 2.3. In order to highlight intra-set variability, we include heatmaps showing the absolute difference with respect to one (`strip`) reconstruction. Upon visual inspection, we see that discrepancies between reconstructions are smaller in the set obtained using implementation-adapted filters. An interesting point to note is that the Gridrec and `iradon` reconstructions show the largest differences from the ASTRA `strip` kernel reconstruction in both sets. This suggests that differences between different software packages are greater than differences between different projectors in the same software package.

To further investigate intra-set variability, we use pixelwise standard deviation maps for all sets of reconstructions. We observe that higher values of standard deviation are observed when using the Ram-Lak and Shepp-Logan filters. This indicates that quantitative differences between these reconstructions were more pronounced. In contrast, reconstructions using our implementation-adapted filters were more similar, resulting in low pixelwise standard deviations. Furthermore, the mode of the histogram of standard deviations (in the central slice) is shifted closer to zero for reconstructions with our filters, and the tail of the histogram is shorter. This highlights the fact that the *maximum* standard deviation between reconstructions with our filters is smaller than the maximum standard deviation in reconstructions with the Shepp-Logan or Ram-Lak filters.

Dependence of filters on noise and sparse angular sampling

We consider the effect of noise and sparse sampling on our filters. For the central slice of the foam phantom shown in Fig. 2.3, we generated data by varying the number of projection angles N_θ and the photon flux I_0. For each of these settings, we computed the mean standard deviation (2.9) between reconstruction slices. Our results are shown in Fig. 2.4. For all noise and angular sampling settings, the mean standard deviation in the slice was reduced by using implementation-adapted filters, with the difference being particularly prominent for noisy and smaller angular sampling settings. Shepp-Logan filter reconstructions had smaller mean standard deviation compared with Ram-Lak filter reconstructions, except in situations where many angles ($N_\theta \geq 256$) were used. In the high angle regime, reconstructions using the Ram-Lak filter have a relatively small number of artefacts and improvements due to filter optimization are modest.

We also quantified the mean squared bias and the mean RMSE with respect to the ground truth for this slice. From these plots, we observe that reconstructions using implementation-adapted filters have lower mean squared bias and mean RMSE compared

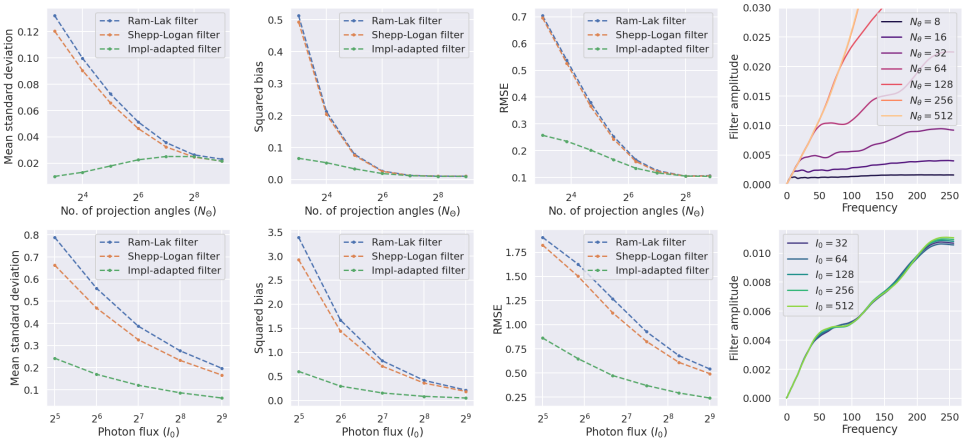


Figure 2.4: Implementation-adapted filters for noisy and sparsely sampled data. (*top, left to right*) Mean standard deviations $\bar{\sigma}^S$ for slice $S = 128$ as a function of the number of projection angles N_θ, mean value of the squared bias, mean value of RMSE with respect to the ground truth slice, and optimized filters in Fourier space. (*bottom, left to right*) Mean standard deviations in $S = 128$ as a function of photon flux I_0 (higher values of I_0 correspond to lower noise levels) using $N_\theta = 64$, mean value of the squared bias, mean value of RMSE with respect to the ground truth slice, and optimized filters in Fourier space.

with those for reconstructions with standard filters. High noise (low I_0) and sparse angular sampling settings result in an increase in bias and RMSE for all filter types. However, the increase is sharper for the Shepp-Logan and Ram-Lak filters than for our implementation-adapted filters. For every noise setting, the Ram-Lak filter results in the worst reconstructions in terms of bias and RMSE. Although both bias and RMSE increase as the number of projection angles is reduced in the noise-free setting, we observe a reduction in mean standard deviation for reconstructions using implementation-adapted filters. This suggests that in spite of a reduction in mean standard deviation due to effective suppression of high frequencies, the reconstructions produced by our implementation-adapted filters in this regime are incapable of mitigating the large number of low-angle artefacts. In effect, these settings show a limit where optimization of a linear filter is not sufficient for good reconstructions, and intra-set homogeneity is achieved at the expense of an increase in bias and RMSE.

In addition, we also show the shapes of the filters (computed for the `strip` kernel in the ASTRA toolbox) as a function of noise and angular sampling. As the number of projection angles is increased, the shape of implementation-adapted filters approaches that of the ramp filter. In these regimes, reconstructions obtained using the Ram-Lak filter and the Shepp-Logan filter are nearly identical in terms of bias and RMSE. For different noise settings, the filters only vary at certain frequencies. It is possible that these frequencies are indicative of the main features in the foam phantom slice used.

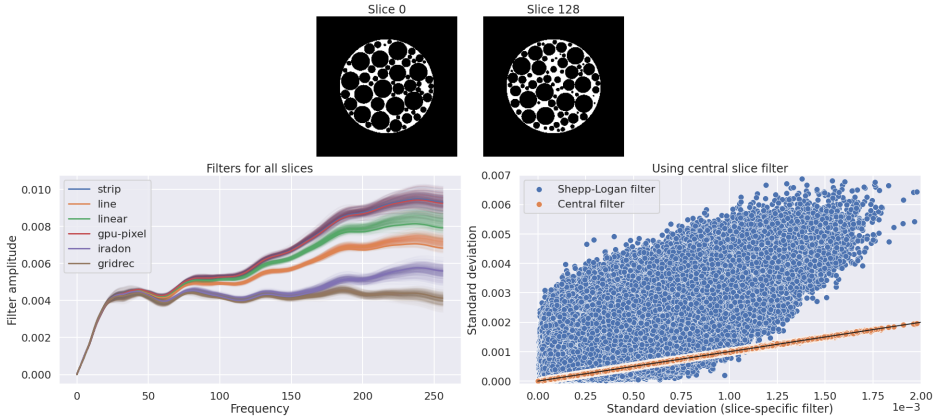


Figure 2.5: Variation of filters with projection data. (*top*) Two slices of a simulated foam phantom with differences in features. (*bottom left*) Implementation-adapted filters for all slices of the foam phantom (slice-specific filters). Central slice (slice no. 128) filters for each implementation are indicated with bold lines. (*bottom right*) Scatter plot of pixelwise standard deviations σ using slice-specific filters, the central slice filter and the Shepp-Logan filter. Standard deviations using the central slice filter are almost the same as those using slice-specific filters (orange dots). These points lie on a straight line (shown in black) with slope ~ 1 and intercept ~ 0. In contrast, standard deviations using the Shepp-Logan filter are higher than those using slice-specific filters (blue dots) for most pixels.

Variation of filters with projection data

In order to understand how our filters change with changes in the data, we computed filters for all slices of our simulated foam phantom. Two such slices are shown in Fig. 2.5. These slices, although visually similar, have different features. Implementation-adapted filters for all 256 slices of the foam phantom are shown in Fig. 2.5.

In order to study the applicability of the central slice filter to other slices, we performed the following experiment. First, we reconstructed all slices using the slice-specific filters, i.e. filters that had been optimized for *each individual slice* using different implementations. Next, we reconstructed all slices with the central slice filter. As a baseline, all slices were also reconstructed using the Shepp-Logan filter. Pixelwise standard deviations (2.8) were computed for all pixels in the foam phantom volume for the three cases. The scatter plot in Fig. 2.5 shows that the pixelwise standard deviations with the central slice filter are nearly the same as those with the slice-specific filters. In fact, these points lie on a line with slope nearly equal to one. This indicates that using the central slice filter results in an equivalent reduction in differences between reconstructions as slice-specific filters. In contrast, the pixelwise standard deviations using the Shepp-Logan filter are, for a majority of pixels, larger than those obtained using slice-specific filters. This suggests that, for a majority of pixels in the reconstruction volume, smaller values of standard deviation are observed after filter optimization.

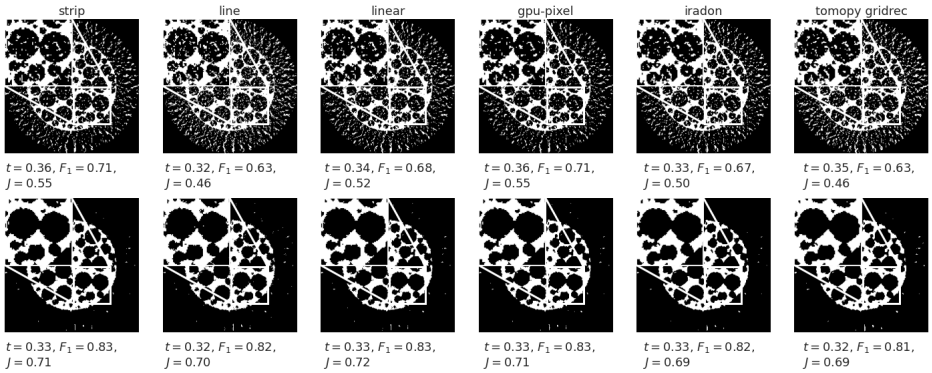


Figure 2.6: Differences after thresholding using Otsu's method. Reconstructions shown in Fig. 2.3 were used as input to the thresholding routine. (*top row*) Thresholded reconstructions obtained using different backprojector implementations and the Shepp-Logan filter. Corresponding Otsu thresholds t, F_1 scores and Jaccard indices are given for each image. (*bottom row*) Thresholded reconstructions obtained using implementation-adapted filters.

Our experiment suggests that using the central slice filter for all slices of the foam phantom results in an equivalent reduction in standard deviation as slice-specific filters. This paves the way to fast application of such filters in a real dataset. An implementation-adapted filter computed for one slice of such a dataset could be reused with all other slices with no additional computational cost, just like any of the standard filters in a software package.

Reduction in differences after thresholding

We investigated the effect of our filters on the results of a simple post-processing step. We reconstructed data ($N_\theta = 32$, no noise) from the central slice of the foam phantom and used Otsu's method in `scikit-image` [44] to threshold reconstruction slices from different implementations. In Fig. 2.6, we show two sets of thresholded reconstructions, one obtained using the Shepp-Logan filter and the other obtained using our implementation-adapted filters. We show values for the Otsu threshold t, the F_1 score with respect to the ground truth slice and the Jaccard index in the figure. We used routines in `scikit-learn` [53] to compute all segmentation metrics. For the set of Shepp-Logan filter reconstructions, the ranges of threshold values (0.32-0.36), F_1 scores (0.63-0.71) and Jaccard indices (0.46-0.55) were larger than the corresponding ranges for the implementation-adapted filter reconstructions. For the latter set, the Otsu threshold varied between 0.32 and 0.33 for all reconstructions. The F_1 scores were between 0.81 and 0.83, and the Jaccard indices were in the range of 0.69-0.72. Upon visual inspection of the zoomed-in insets we find greater differences between thresholded reconstructions in the set of Shepp-Logan filter reconstructions. These results suggest that post-processing steps such as segmentation may be rendered more reproducible and amenable to automation if reconstructions are obtained using implementation-adapted filters.

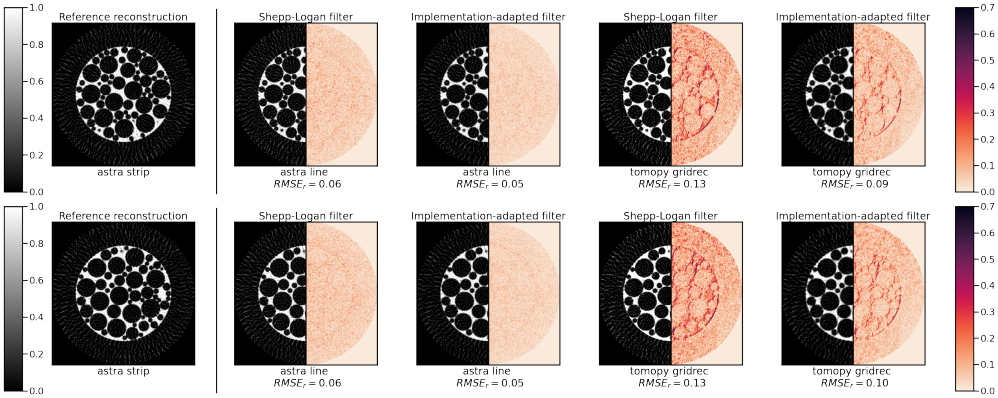


Figure 2.7: Filter optimization using a reference reconstruction. (*top row*) Filters optimized to a `strip` kernel reconstruction (*top row, left*). (*top row*) Reconstructions before and after filter optimization using the ASTRA `line` kernel and Gridrec. Right half of each image shows absolute difference with the reference reconstruction. RMSE values with respect to the reference are also shown. (*bottom row*) Reconstructions of a different (test) slice using the filters obtained for the slice in the top row. Pixelwise absolute difference and RMSE using implementation-adapted filters are smaller in both cases.

Optimizing to a reference reconstruction

Although we focus on filter optimization in sinogram space in this chapter, a related optimization problem is one where reconstruction results from different implementations are optimized to a reference reconstruction. This type of optimization might be useful when the result of one specific implementation is preferred due to its superior accuracy and when the exact settings used with this algorithm are unknown.

In some cases, high-quality reconstructions might be computed with an unknown (possibly in-house) software package during the experiment by expert beamline scientists. When users reconstruct this data later at their home institutes, it might not be possible to use the same software packages with identical settings. Our approach would enable users to reduce the difference between their reconstructions and the high-quality reference reconstructions.

Optimization in reconstruction space can be performed by modifying the objective in (2.5):

$$\boldsymbol{h}_I^* = \arg\min_{\boldsymbol{h}} \|\boldsymbol{r}_{\text{ref}} - \boldsymbol{r}_I(\boldsymbol{h}, \boldsymbol{p})\|_2^2, \tag{2.14}$$

where $\boldsymbol{r}_{\text{ref}}$ is the reference reconstruction.

To illustrate filter optimization in reconstruction space, we performed the following experiment. Using the `strip` kernel reconstruction (with the Shepp-Logan filter) as a reference, we computed optimized filters for two other implementations (ASTRA `line` kernel and TomoPy Gridrec) for reconstructing the central slice of the foam phantom. Subsequently, we reconstructed the sinogram with the Shepp-Logan filter and our filters. These reconstructions are shown in the top row of Fig. 2.7. To quantify similarity with

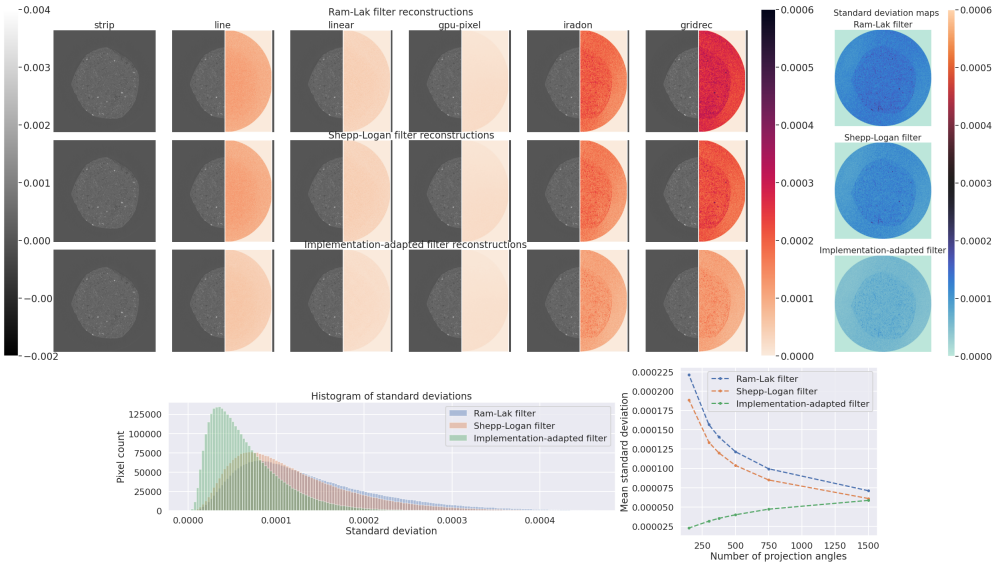


Figure 2.8: Reduction in differences between reconstructions of the Round-Robin dataset (slice no. 896). (*top three rows*) Slice reconstructions using different implementations. Reconstructions were performed by discarding every second projection from the full dataset. The right half of the images show absolute differences with the corresponding `strip` kernel reconstruction in each set. The rightmost column shows pixelwise standard deviations in each set. (*bottom row, left*) Histograms of standard deviation for all three types of filters. (*right*) Mean standard deviations $\bar{\sigma}^S$ in slice $S = 896$ for different numbers of projection angles.

the reference reconstruction, we computed the pixelwise absolute difference between each reconstruction and the reference as well as the RMSE using the reference as ground truth, which we denote as RMSE_r. For both `line` and Gridrec backprojectors, optimizing the filter to a reference reconstruction reduced the RMSE_r and absolute difference. As a further test, we applied the filters computed for this slice to a different slice of the foam phantom, which did not have any overlaps with the slice used to compute the filters. For this test slice, we again observed the reduction in RMSE_r and absolute error, suggesting that our filters were able to bring the resulting reconstructions closer to the reference reconstruction.

2.5.2 Round-Robin data

Fig. 2.8 shows the results of our method on the central slice (slice no. 896) of the Round-Robin dataset N1. These reconstructions were performed by discarding every second projection from the entire dataset. From the heatmaps of absolute difference with respect to the `strip` kernel reconstruction, we observe that intra-set differences are reduced by using implementation-adapted filters. This is further shown by the pixelwise standard

deviation maps. Standard deviations between reconstructions using the Ram-Lak and Shepp-Logan filters are larger than those between reconstructions using implementation-adapted filters. Similar to the distributions in Fig. 2.3, we see that our implementation-adapted filters are able to shift the mode of the histogram of standard deviations towards zero and to reduce the number of large standard deviations in the slice. We also observe that the Ram-Lak filter reconstructions show higher standard deviations than the Shepp-Logan filter reconstructions.

We also studied the effect of the number of projections used on the mean standard deviation (2.9) in this slice. To do this, we performed experiments with the whole dataset and also with parts of the data, where every $2, 3, 4, 5$ and 10 projections were discarded. For each instance, the data were reconstructed using the Ram-Lak filter, the Shepp-Logan filter and our implementation-adapted filters. The plot of mean standard deviations is shown in Fig. 2.8. For all projection numbers, filter optimization reduced the mean standard deviation in the slice. The difference was smaller for higher projection numbers, indicating that our filters are especially useful in improving reproducibility of reconstructions when the number of projection angles is small. In practice, data along few angles may be acquired to reduce the X-ray dose on a sample or to speed up acquisition when the sample is evolving over time.

2.6 Discussion

In this chapter, we presented a method to improve the reproducibility of reconstructions in the synchrotron pipeline. Our method uses an optimization problem over filters to reduce differences between reconstructions from various software implementations of commonly-used algorithms.

The objective function that was used in our optimization problem was the ℓ^2-distance between the forward projection of the obtained reconstruction and the given projection data. This choice was motivated by the fact that ground truth reconstructions are generally not available in real-world experiments. However, it is possible to formulate a similar (and related) problem in reconstruction space, by using the ℓ^2-distance between the reconstruction from a given software package and a reference reconstruction as the objective to be minimized. The solution to such an optimization procedure is a shift-invariant blurring kernel in reconstruction space. The implementation-adapted filters presented in this chapter can thus be viewed as a linear transformation of the projection data that results in an automatic selection of shift-invariant blurring of reconstructions.

Our work here can be extended to optimize other pre-processing and post-processing steps in the synchrotron pipeline. An important example is phase retrieval, which can be formulated in terms of a filtering operation [35]. This filter can be optimized similarly in order to improve reproducibility.

One limitation of our method is that we optimize to the data available. This optimization can lead to undesired solutions in the presence of outliers in the data, such as zingers or ring artefacts. Reconstructions of data corrupted with zingers (randomly placed very bright pixels in the sinogram) are shown in Fig. 2.9. In this example we see that the FBP reconstruction using the ASTRA `strip` kernel and the Shepp-Logan filter shows less

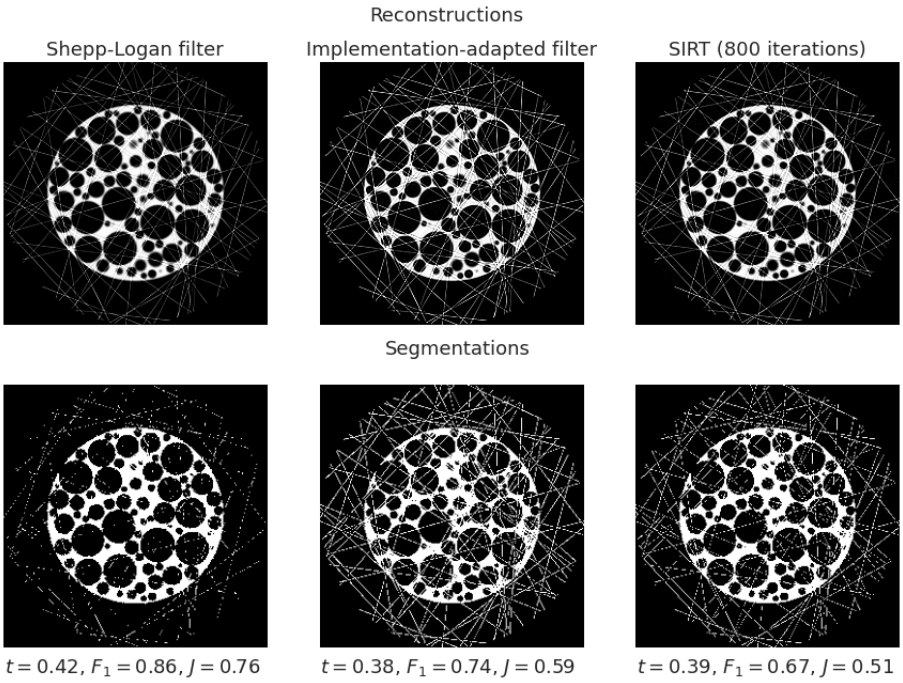


Figure 2.9: Reconstructions of data corrupted with zingers showing an example where the Shepp-Logan filter reconstruction and corresponding segmentation are better than those using an implementation-adapted filter or an iterative method (SIRT). (*top row*) Reconstructions of data from slice 128 ($N_\theta = 512$, no noise) corrupted with zingers. Zingers are more prominent in the reconstruction using an implementation-adapted filter and in the SIRT reconstruction (after 800 iterations). (*bottom row*) Segmentations using Otsu's method of all three reconstructions. The Otsu threshold, F_1 score and Jaccard index for each image is given below.

prominent zingers than the reconstruction using an implementation-adapted filter. This is because the optimized filter preserves the zingers in the data whereas the unoptimized FBP reconstruction is independent of them. Other methods, such as the simultaneous iterative reconstruction technique (SIRT), which iteratively minimize the data misfit also give similar, poor reconstructions. One way to improve iterative reconstruction methods is to use regularization, which can be achieved either by early stopping or by the inclusion of an explicit regularization term in the objective function to be minimized. Analogous techniques can be used for our filter optimization problem (2.5) to ensure greater robustness to outliers.

Although we have demonstrated the reusability of our filters for similar data, these filters are dependent on the noise statistics and angular sampling in the acquired projections. One way to improve the generalizability of filters would be to simultaneously optimize to more than one dataset. This idea has been explored in [54], [55] using shallow neural

networks.

Another promising direction is provided by deep learning-based methods, which have been applied to improve tomographic image reconstruction in a number of ways [56]. Supervised deep learning approaches can be used to learn a (non-linear) mapping from input reconstructions to a reference reconstruction. However, such approaches generally require large amounts of paired training data (input and reference reconstructions). When insufficient training pairs are available, various unsupervised approaches, such as the Deep Image Prior method proposed in [57], are more suitable. For a quantitative comparison of various popular deep learning-based reconstruction methods, we refer the reader to [58].

Apart from software solutions for image reconstruction, which have been the focus of this chapter, improving reproducibility throughout the synchrotron pipeline requires hardware adjustments to the blocks in Fig 2.1. Scanning the same sample twice under the same experimental conditions leads to small fluctuations in the data due to stochastic noise and drifts during the scanning process. In addition, beam-sensitive samples might deform due to irradiation. Such changes lead to differences in reconstructions that are similar to the differences due to software implementations, albeit less structured than those shown in Fig. 2.2. To improve hardware reproducibility, controlled phantom experiments might be performed to address differences in data acquisition. Finally, software and hardware solutions can be effectively linked by using approaches like reinforcement learning for experimental design and control [59], [60]. Such creative solutions might provide an efficient way for synchrotron users to perform reproducible experiments in the future.

2.7 Conclusion

In this chapter, we proposed a filter optimization method to improve reproducibility of tomographic reconstructions at synchrotrons. These implementation-adapted filters can be computed for any black-box software implementation by using only evaluations of the corresponding reconstruction routine. We numerically demonstrated the properties of and use cases for such filters. In both real and simulated data, our implementation-adapted filters reduced the standard deviation between reconstructions from various software implementations of reconstruction algorithms. The reduction in standard deviation was especially evident when the data were noisy or sparsely sampled.

Our filter optimization technique can be used to reduce the effect of differences in discretization and interpolation in commonly-used software packages and is a key building block towards improving reproducibility throughout the synchrotron pipeline. We make available the open-source Python code for our method, allowing synchrotron users to obtain reconstructions that are more comparable and reproducible.

Chapter 3

Sparse grid-free reconstruction of nanocrystal defects

3.1 Introduction

Electron tomography is a powerful technique for resolving the interior of nanomaterials. After preparing a microscopic sample, a series of projection images (so called tilt-series) is acquired by rotating the specimen in the electron microscope, acquiring data from a range of angles. In recent years, electron tomography has been successfully applied to reconstruct the 3D positions of the individual atoms in nanocrystalline materials [61]–[63].

Since the first demonstration of atomic resolution tomography of nanocrystals in 2010 by discrete tomography [64], a range of tomographic acquisition techniques and reconstruction algorithms have been applied to reconstruct nanocrystals of increasing complexity. In the discrete tomography approach, atoms are assumed to lie on a regular lattice and the measured projections can be considered as atom counts along lattice lines. A key advantage of this approach is its ability to exploit the constraints induced by the discrete domain and range of the image. As a consequence, a small number of projection angles (typically less than 5) can already lead to an accurate reconstruction [16], [17]. The theoretical properties of the discrete reconstruction problem have been studied extensively with results on algorithm complexity, uniqueness, and stability [65]–[67]. A key drawback of the discrete lattice assumption when considering real-world applications to nanocrystal data is that in many interesting cases the atoms do not lie on a perfect lattice due to defects in the crystal structure or interfaces between different crystal lattices. In such cases the atoms do not project perfectly into columns, forming a mismatch with the discrete tomography model.

As an alternative, it has been demonstrated that a more conventional tomographic

This chapter is based on:

Atomic Super-resolution Tomography. *P. S. Ganguly, F. Lucka, H. J. Hupkes, K. J. Batenburg*. International Workshop on Combinatorial Image Analysis. Springer, Cham, pp. 45-61, 2020.

series consisting of hundreds of projections of a nanocrystal can be acquired in certain cases. An image of the nanocrystal is then reconstructed using sparsity based reconstruction techniques on a continuous model of the tomography problem. This approach does not depend on the lattice structure and allows one to reconstruct defects and interfaces [18]. As a downside, the number of required projections is large and to accurately model the atom positions the reconstruction must be represented on a high-resolution pixel grid resulting in a large-scale computational problem. This raises the question if a reconstruction problem can be defined that fills the gap between these two extremes and can exploit the discrete nature of the lattice structure while at the same time allowing for continuous deviations of atom positions from the perfect lattice.

In this chapter we propose a model for the atomic resolution tomography problem that combines these two characteristics. Inspired by the algorithm proposed in [12], our model is based on representing the crystal image as a superposition of delta functions with continuous coordinates and exploiting sparsity of the image to reduce the number of required projections. We show that by incorporating a physical model for the potential energy of the atomic configuration, the reconstruction results can be further improved.

3.2 Problem setting

In this section we formulate a mathematical model of the atomic resolution tomography problem and discuss several approaches to solve it. Some of these approaches assume that the atom locations are restricted to a perfect grid, the *crystal lattice*, which corresponds to only one possible local minimum of the potential energy of the atomic configuration. To overcome certain limitations of this assumption, we propose an alternative formulation where the atom locations are allowed to vary continuously and an explicit model of the potential energy of their configuration is used to regularize the image reconstruction.

An atomic configuration is characterized by a positive measure μ on a bounded subset X of $\mathbb{R}^d$. We denote the space of such measures by $\mathcal{M}(X)$. The measure represents the *electron density*, which is the probability that an electron is present at a given location. The electron density around an atomic configuration is highest in regions where atoms are present. In electron tomography, electron density is probed by irradiating a sample with a beam of electrons. The beam undergoes absorption and scattering due to its interactions with the electrons of the atomic configuration. The transmitted or scattered signal can then be used to form an image. The Radon transform provides a simplified mathematical model of this ray-based image formation process. For $d = 2$, the Radon transform $\mathcal{R}\mu$ can be expressed as integrals taken over straight rays

$$\mathcal{R}[\mu](r,\theta) := \int_{l(r,\theta)} d\mu, \tag{3.1}$$

$$l(r,\theta) = \{(x_1,x_2) \in \mathbb{R}^2 \,|\, x_1\cos\theta + x_2\sin\theta = r\}, \tag{3.2}$$

where we parametrized the rays by the projection angle θ and the distance on the detector r. The corresponding inverse problem is to recover μ from noisy observations of $y = \mathcal{R}\mu + \varepsilon$. One way to formulate a solution to this problem is via the following optimization over the

space of measures:

$$\underset{\mu \in \mathcal{M}(X)}{\text{minimize}} \quad \|\mathcal{R}\mu - y\|_2^2, \tag{3.3}$$

which is an infinite dimensional non-negative linear least-squares problem. In the following, we will introduce a series of discretizations of this optimization problem. Numerical schemes to solve them will be discussed in Section 3.3.

In situations where we only have access to data from a few projection angles, introducing a suitable discretization of (3.3) is essential for obtaining a stable reconstruction. One way to achieve this is to restrict the atom locations to a spatial grid with n nodes, $\mathbf{x}_{i=1}^n$, and model their interaction zone with the electron beam by a Gaussian with known shape G. The atom centres are then delta peaks $\delta_{\mathbf{x}_i}$ on the gridded image domain. The Gaussian convolution of atom centres can be viewed as the "blurring" produced by thermal motion of atoms. In fact, it is known from lattice vibration theory that, for large configurations, the probability density function of an atom around its equilibrium position is a Gaussian, whose width depends on temperature, dimensionality and interatomic forces [68]. The discretized measure μ can then be written as

$$\mu_{\text{grid}} = \sum_{i=1}^{n} w_i (G * \delta_{\mathbf{x}_i}), \tag{3.4}$$

where n is the total number of grid points and weights $w_i \geq 0$ were introduced to indicate confidence in the presence or absence of an atom at grid location i. If we insert (3.4) in (3.3) and introduce the forward projection of a single atom as $\psi_i \coloneqq \mathcal{R}(G * \delta_{\mathbf{x}_i})$ we get

$$\|\mathcal{R}\mu_{\text{grid}} - y\|_2^2 = \|\mathcal{R}\sum_{i=1}^{n} w_i (G * \delta_{\mathbf{x}_i}) - y\|_2^2 = \\ \|\sum_{i=1}^{n} w_i \mathcal{R}(G * \delta_{\mathbf{x}_i}) - y\|_2^2 =: \|\sum_{i=1}^{n} \psi_i w_i - y\|_2^2 =: \|\Psi w - y\|_2^2$$

The corresponding optimization problem is given by

$$\underset{w \in \mathbb{R}_+^n}{\text{minimize}} \quad \|\Psi w - y\|_2^2 \quad , \tag{3.5}$$

which is a finite dimensional linear non-negative least squares problem.

The choice of the computational grid in (3.4) is unfortunately not trivial. Only in certain situations, one can assume that all atoms lie on a lattice of known grid size and orientation, and directly match this lattice with the computational grid. In general, one needs to pick a computational grid of much smaller grid size. With the data y given, the grid admits multiple solutions of (3.5) and most efficient computational schemes tend to pick a blurred, artefact-ridden solution with many non-zero weights far from the true, underlying μ, as we will demonstrate in Section 3.4. To obtain a better reconstruction, one can choose to add *sparsity constraints* which embed our physical *a priori* knowledge that μ originates from a discrete configuration of atoms. In our model (3.4), this corresponds

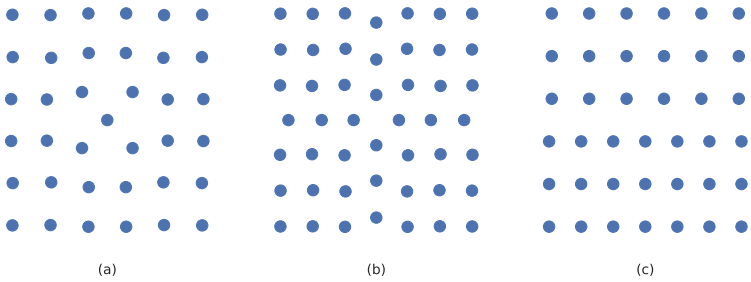


Figure 3.1: Atomic configurations with (a) an interstitial point defect, (b) a vacancy and (c) an edge dislocation.

to a $w \in \mathbb{R}^n_+$ with few non-zeros entries. To obtain such a sparse solution we can add a constraint on ℓ^0 norm of the weights to the optimization problem:

$$\begin{aligned} \underset{w \in \mathbb{R}^n_+}{\text{minimize}} \quad & \|\Psi w - y\|_2^2 \\ \text{subject to} \quad & |w|_0 \leq K. \end{aligned}$$

However, this problem is NP-hard [69]. A approximate solution can be found by replacing the ℓ^0 norm with the ℓ^1 norm and adding it to the objective function:

$$\underset{w \in \mathbb{R}^n_+}{\text{minimize}} \quad \|\Psi w - y\|_2^2 + \lambda \|w\|_1, \tag{3.6}$$

where λ is the relative weight of the sparsity-inducing term. This particular choice of formulation is not always best and alternative formulations of the same problem exist [69].

For atomic configurations where only one type of atom is present, the weights can be considered to be one where an atom is present and zero everywhere else. This corresponds to discretizing the range of the reconstructed image. The fully discrete optimization problem then becomes:

$$\underset{w \in \{0,1\}^n}{\text{minimize}} \quad \|\Psi w - y\|_2^2. \tag{3.7}$$

With image range discretization, a constraint on the number of atoms is typically no longer needed because adding an additional atom with weight 1 after all atoms have been found leads to an increase in the objective function.

Although the optimization problems (3.5), (3.6) and (3.7) allow for the recovery of atomic configurations without solving (3.3), all of them rely crucially on discretization of the domain of the reconstructed image, i.e. the assumption that atoms lie on a grid. However, this assumption is not always true. In particular, atomic configurations often contain defects where atom positions deviate from the perfect lattice. Fig. 3.1 shows examples of common lattice defects. In order to resolve these defects correctly, the image domain must be discretized to higher resolutions, i.e. the grid of possible atom positions

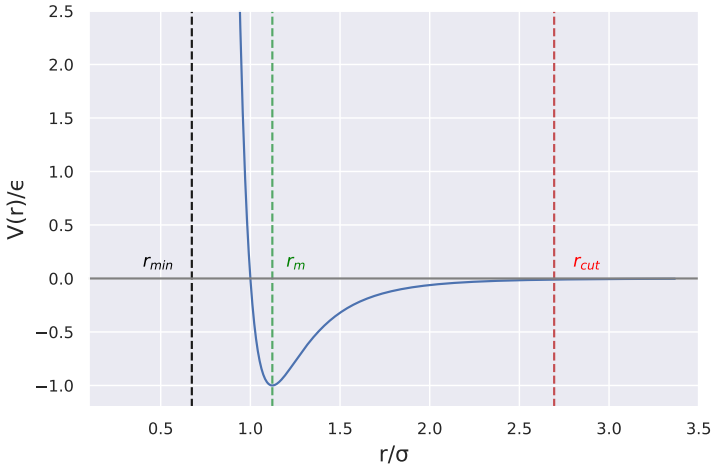


Figure 3.2: The normalized Lennard-Jones pair potential as a function of normalized interatomic separation.

must be made finer. This introduces two main problems: First, making the grid finer for the same data makes the inverse problem more ill-posed. Second, the computational time increases significantly even for modestly sized configurations.

In order to overcome these difficulties, we revisit (3.4) and remove the requirement for $\mathbf{x}_i$ to lie on a grid. The projection of a single atom now becomes a function of its location $\mathbf{x} \in \mathbb{R}^d$, $\psi(\mathbf{x}) := \mathcal{R}(G * \delta_{\mathbf{x}})$. We keep the image range discretization introduced above by requiring $w_i \in \{0, 1\}$. Now, (3.7) becomes

$$\underset{\mathbf{x}\in X^n, w\in\{0,1\}^n}{\text{minimize}} \quad \left\| \sum_{i=1}^{n} w_i \psi(\mathbf{x}_i) - y \right\|_2^2. \tag{3.8}$$

The minimization over $\mathbf{x}$ is a non-linear, non-convex least-squares problem which has been studied extensively in the context of mathematical super-resolution [12], [70], [71]. In these works, efficient algorithms are derived from relating it back to the infinite dimensional linear least-squares problem on the space of measures (3.3). For instance, for applications such as fluorescence microscopy [12] and ultrasound imaging [72], an alternating descent conditional gradient (ADCG) algorithm has been proposed, which we will revisit in the next section. Compared to these works, we have a more complicated non-local and under-determined inverse problem and the minimization over w adds a combinatorial, discrete flavor to (3.8). To further tailor it to our specific application, we will incorporate physical *a priori* knowledge about atomic configurations of crystalline solids by adding a functional formed by the atomic interaction potentials. This will act as a regularization of the underlying under-determined inverse problem.

3.2.1 Potential energy of the atomic configuration

The total energy of an atomic configuration is the sum of its potential energy and kinetic energy. As we consider only static configurations, the kinetic energy of the configuration is zero and the total energy is equal to the potential energy. In order to compute the potential energy of the atomic configuration, we must prescribe the interaction between atoms. In this chapter, we use the Lennard-Jones pair potential, which is a simplified model of interatomic interactions. The Lennard-Jones potential V_{LJ} as a function of interatomic separation r is given by [73]

$$V_{\text{LJ}}(r) = \begin{cases} 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right], & r < r_{\text{cut}} \\ 0, & r \geq r_{\text{cut}} \end{cases} \tag{3.9}$$

where ϵ is the depth of the potential well and σ is the interatomic separation at which the potential is zero. The separation at which the potential reaches its minimum is given by $r_m = 2^{1/6}\sigma$. The parameter r_{cut} denotes a cut-off separation beyond which the potential is inactive. Fig. 3.2 shows the form of the the Lennard-Jones pair potential as a function of interatomic separation. The potential energy of the atomic configuration is computed by summing over the pairwise interaction between all pairs of atoms

$$V_{\text{tot}}(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N) = \sum_{i>j} V_{\text{LJ}}(\mathbf{x}_i - \mathbf{x}_j). \tag{3.10}$$

Adding this energy to the objective in (3.8) leads to

$$\underset{\mathbf{x}\in\mathcal{C}, w\in\{0,1\}^n}{\text{minimize}} \quad \left\| \sum_{i=1}^{n} w_i \psi(\mathbf{x}_i) - y \right\|_2^2 + \alpha V_{\text{tot}}(\mathbf{x}). \tag{3.11}$$

The regularization parameter, α, adjusts the relative weight of the energy term, so that by tuning it we are able to move between atomic configurations that are data-optimal and those that are energy-optimal. The constraint set $\mathcal{C} \subset X^n$ is defined by a minimum distance $r_{\min}$, such that $|\mathbf{x}_i - \mathbf{x}_j| > r_{\min}$, $\forall i > j$. The minimum distance, $r_{\min}$, is chosen to be smaller than the optimal interatomic separation r_m and allows us to set α to 0 and still avoid configurations where atoms are placed exactly at the same position. For small separations, the energy is dominated by the $\left(\frac{\sigma}{r}\right)^{12}$ term and increases sharply for separations less than r_m. Thus, for non-zero α, configurations where atoms have a separation less than r_m are highly unlikely.

3.3 Algorithms

In this section we discuss several algorithms to solve the optimization problems introduced in Section 3.2.

3.3.1 Projected gradient descent

The non-negative least-squares problem (3.5) can be solved with a simple iterative first-order optimization scheme. At each step of the algorithm, the next iterate is computed by moving in the direction of the negative gradient of the objective function. Non-negativity of the weights is enforced by projecting negative iterates to zero. Mathematically, each iterate is given by

$$w^{k+1} = \prod_{+}\Big(w^k + t\Psi^T(\Psi w^k - y)\Big), \tag{3.12}$$

where t is the step size and the projection operator is given by

$$\prod_{+}(\cdot) = \max(\,\cdot\,, 0). \tag{3.13}$$

In the numerical experiments in Section 3.4, we used the SIRT algorithm [29] as implemented in the tomographic reconstruction library ASTRA [7], which is based on a minor modification of the iteration described above.

3.3.2 Proximal gradient descent

If we add the non-smooth ℓ^1 regularizer and obtain problem (3.6), we need to extend (3.12) to a proximal gradient scheme [74]

$$w^{k+1} = \mathrm{prox}_h\Big(w^k + t\Psi^T(\Psi w^k - y)\Big), \tag{3.14}$$

where the projection operator (3.13) is replaced by the proximal operator of the convex function

$$h(x) := \begin{cases} \lambda\|x\|_1 & x \geq 0 \\ 0 & \text{elsewhere} \end{cases}, \tag{3.15}$$

which is given by the non-negative soft-thresholding operator

$$\mathrm{prox}_h(x) = \begin{cases} x - \lambda, & x \geq \lambda \\ 0, & \text{elsewhere} \end{cases}.$$

In the numerical experiments in Section 3.4, we used the fast iterative soft-thresholding algorithm (FISTA) [75] as implemented in the Python library ODL [8], which is based on a slight modification of the iteration described above.

3.3.3 Simulated annealing

For solving the fully discrete problem (3.7), we used a simulated annealing algorithm as shown in Algorithm 2, which consists of two subsequent accept-reject steps carried out with respect to the same inverse temperature parameter β. In the first one, the algorithm tries to add a new atom to the existing configuration. In the second one, the atom

Algorithm 2 Discrete simulated annealing

while $\beta < \beta_{\text{max}}$ **do**
  Select new atom location: $\tilde{w}^k \in \arg\min_{k \in \mathcal{C}} \Psi w^k - y$
  Add new atom to current configuration: $\tilde{w}^{k+1} \leftarrow \{w^k, \tilde{w}^k\}$
  Accept new configuration with a certain probability:
  if $\beta \|\Psi \tilde{w}^{k+1} - y\|_2^2 < \beta \|\Psi w^k - y\|_2^2$ **then**
    $w^{k+1} \leftarrow \tilde{w}^{k+1}$
  else
    Generate random number: $t \in \text{rand}[0, 1)$
    if $t < e^{-\beta \|\Psi \tilde{w}^{k+1} - y\|_2^2} / e^{-\beta \|\Psi w^k - y\|_2^2}$ **then**
      $w^{k+1} \leftarrow \tilde{w}^{k+1}$
    end if
  end if
  Move atom: $w^{k+1} \leftarrow$ random move(w^{k+1})
  Run acceptance steps 5–13
  Increase β
end while

locations are perturbed locally. As β is increased towards β_{max}, fewer new configurations are accepted and the algorithm converges to a minimum.

In the atom adding step at each iteration k, the algorithm tries to add an atom at one of the grid location i where the residual $\Psi w^k - y$ is minimal (this corresponds to flipping w_i^k from 0 to 1 in (3.7)). The allowed grid locations belong to a constraint set $\mathcal{C}$, such that no two atoms are closer than a pre-specified minimum distance $r_{\min}$. To perturb the atom positions locally, the algorithm selects an atom at random and moves it to one of its 4 nearest neighbor locations at random.

Algorithm 3 ADCG with energy

for $k = 1 : k_{\text{max}}$ **do**
  Compute next atom in grid g:
  $\mathbf{x}_{\text{new}} \in \arg\min_{\mathbf{x}_{\text{new}} \in g, (\mathbf{x}^k, \mathbf{x}_{\text{new}}) \in \mathcal{C}} \| \sum_{i=1}^k \psi(\mathbf{x}_i) - y + \psi(\mathbf{x}_{\text{new}}) \| + \alpha V_{\text{tot}}(\mathbf{x}^k, \mathbf{x}_{\text{new}})$
  Update support: $\mathbf{x}^{k+1} \leftarrow \{\mathbf{x}^k, \mathbf{x}_{\text{new}}\}$
  Locally move atoms:
  $\mathbf{x}^{k+1} \leftarrow \min_{\mathbf{x} \in X} \|\Psi \mu(\mathbf{x}^{k+1}) - y\|_2^2 + \alpha V(\mathbf{x}^{k+1})$
  Break if objective function is increasing:
  if $\|\Psi \mu(\mathbf{x}^{k+1}) - y\|_2^2 + \alpha V(\mathbf{x}^{k+1}) > \|\Psi \mu(\mathbf{x}^k) - y\|_2^2 + \alpha V(\mathbf{x}^k)$ **then** break
  end if
end for

3.3.4 ADCG with energy

Variants of the Frank-Wolfe algorithm (or conditional gradient method) [19], [76] have been proposed for solving problems of the form (3.8) [72], [77] without discrete constraints for w and are commonly known as alternating descent conditional gradient (ADCG) schemes (see [78] for an analysis specific to multidimensional sparse inverse problems). Here, we modify the ADCG scheme to

1. incorporate binary constraints on w
2. handle the singularities of the atomic interaction potentials
3. avoid local minima resulting from poor initializations

The complete algorithm is shown in Algorithm 3. Essentially, the scheme also alternates between adding a new atom to the current configuration and optimizing the positions of the atoms.

In the first step, the image domain is coarsely gridded and the objective function after adding an atom at each location is computed. Locations closer to existing atoms than $r_{\min}$ are excluded. In the second step, the atom coordinates are optimized by a continuous local optimization method. Here, the Nelder-Mead method [79] implemented in SciPy [80] was used.

A continuation strategy is used to avoid problems resuling from poor initilizations: Algorithm 3 is run for increasing values of α, starting from $\alpha = 0$. The reconstruction obtained at the end of a run is used as initialization for the next. In the following section, we demonstrate the effect of increasing α on the reconstructions obtained and discuss how an optimal α was selected. In the following section, we refer to Algorithm 3 as "ADCG" when used for $\alpha = 0$ and as "ADCG with energy" otherwise.

3.4 Numerical experiments

We conducted numerical experiments by creating 2D atomic configurations with defects and using the algorithms discussed in Section 3.3 to resolve atom positions. In this section we describe how the ground truth configurations were generated and projected, and compare the reconstruction results of different algorithms.

3.4.1 Ground truth configurations

We generated ground truth configurations using the molecular dynamics software HOOMD-blue [81], [82]. We created perfect square lattices and then induced defects by adding or removing atoms. The resulting configuration was then relaxed to an energy minimum using the FIRE energy minimizer [83] to give the configurations shown in Fig. 3.1. The following parameter values were used in (3.9) for specifying the Lennard-Jones pair potential between atoms.

Defect type	ϵ	σ	r_{cut}
Interstitial defect	0.4	0.15	0.4
Vacancy	0.4	0.14	0.4
Edge dislocation	0.4	0.13	0.17

3.4.2 Discretized projection data

We generated two 1D projections for each ground truth atomic configuration at projection angles $\theta = 0°, 90°$. As discussed in Section 3.2, the projection of a single atom centre is given by a Gaussian convolution followed by the Radon transform. The Radon transform of a Gaussian is also a Gaussian. Therefore, we interchanged the two operations in the forward transform to speed up the computations. The sum over individual projections of atom centres was used as the total (noise-free) projection. Using the Radon transform in (3.1), each atom centre was projected onto a 1D detector, following which it was convolved with a 1D Gaussian of the form $G(z) = e^{-(z-z_0)/\varsigma^2}$, where z_0 is the position of the atom centre on the detector and ς controls the width of the Gaussian. Finally, the continuous projection was sampled at a fixed number of points to give rise to a *discrete* projection. For our experiments, the ς of the Gaussian function was taken to be equal to the discretization of the detector given by the detector pixel size d. Both were taken to be 0.01.

3.4.3 Discretization of the reconstruction volume

For SIRT, FISTA and simulated annealing (described in subsections 3.3.1, 3.3.2 and 3.3.3, respectively), each dimension of the reconstruction area was discretized using the detector pixel size d. Therefore, there were $1/d \times 1/d$ grid points in total.

Gridding is required for our variant of ADCG (subsection 3.3.4, Algorithm 3) at the atom adding step. We found that a coarse discretization, with less than $1/9^{\text{th}}$ the number of grid points, was already sufficient.

3.4.4 Comparison between reconstructions

The reconstructions obtained with the different algorithms are shown in Fig. 3.3. For each reconstruction, data from only two projections were used. Note that two projections is far from sufficient for determining the correct atomic configuration and several different configurations have the same data discrepancy.

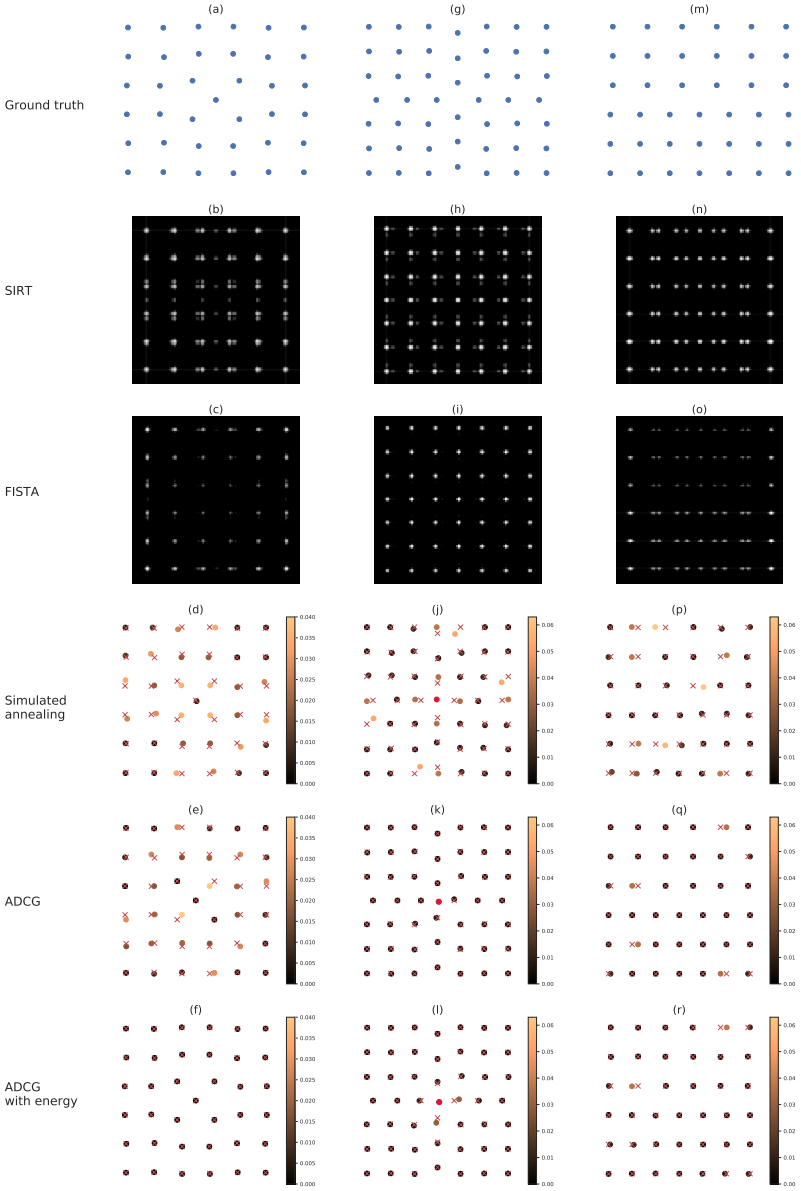


Figure 3.3: Reconstructions of atomic configurations with (a)–(f) an interstitial point defect, (g)–(l) a vacancy and (m)–(r) an edge dislocation from two projections. For the simulated annealing, ADCG and ADCG with energy reconstructions, atoms are coloured according to their Euclidean distance from the ground truth. The ground truth positions are marked with red crosses. In (j)–(l) an extra atom (shown in red) was present in the reconstructions but not in the ground truth.

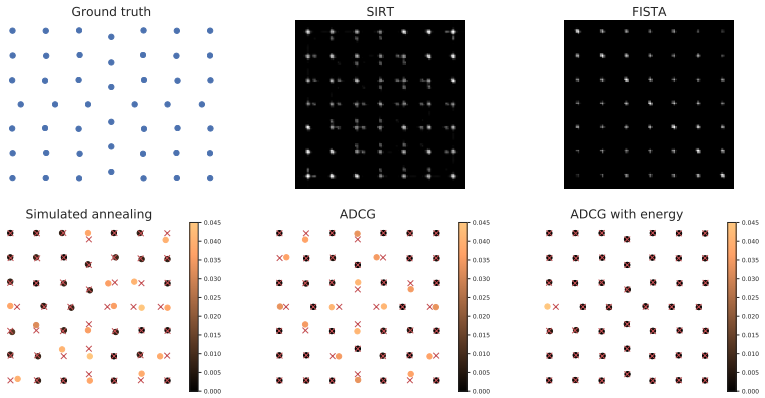


Figure 3.4: Reconstructions of a vacancy defect from three projections. For the simulated annealing, ADCG and ADCG with energy reconstructions, atoms are coloured according to their Euclidean distance from the ground truth. Ground truth positions are marked with red crosses.

In the SIRT reconstructions, atom positions were blurred out and none of the defects were resolved. In all cases, the number of intensity peaks was also different from the true number of atoms. Although FISTA reconstructions, which include sparsity constraints on the weights, were less blurry, atoms still occupied more than one pixel. For both these algorithms, additional heuristic post-processing is required to output atom locations. In the edge dislocation case, both algorithms gave rise to a configuration with many more atoms than were present in the ground truth.

The discrete simulated annealing algorithm performed better for all configurations. For the interstitial point defect and edge dislocation, the number of atoms in the reconstruction matched that in the ground truth. The positions of most atoms, however, were not resolved correctly. Moreover, the resolution, like in previous algorithms, was limited to the resolution on the detector. We ran the simulated annealing algorithm for comparable times as the ADCG algorithms and picked the solution with the least data discrepancy.

Already the ADCG algorithm for $\alpha = 0$ performed far better than all the previous algorithms. For the configurations with an interstitial point defect and edge dislocation, all but a few atom locations were identified correctly. For the configuration with a vacancy, all atoms were correctly placed. However, an additional atom at the centre of the configuration was placed incorrectly.

Adding the potential energy (ADCG with energy) helps to resolve atom positions that were not identified with $\alpha = 0$. For the interstitial point defect and edge dislocation, these reconstructions were the closest to the ground truth. Adding the energy to the

	Interstitial defect		Vacancy (3 projs.)		Edge dislocation	
	Number of atoms	Mean distance	Number of atoms	Mean distance	Number of atoms	Mean distance
Ground truth	37	0.0000	48	0.0000	39	0.0000
SIRT	36	–	49	–	66	–
FISTA	36	–	49	–	66	–
Simulated annealing	37	0.0184	48	0.0164	39	0.0159
ADCG	37	0.0138	48	0.0130	39	0.0049
ADCG with energy	37	0.0018	48	0.0024	39	0.0048

Table 3.1: Number of atoms and mean Euclidean distance from ground truth atoms for reconstructions obtained with different algorithms. Thresholding was used to compute the number of atoms detected in the SIRT and FISTA reconstructions.

configuration with a vacancy moved the atoms near the defect further apart but was not able to correct for the extra atom placed. For this case, we performed an additional experiment with three projections at 0°, 45° and 90°. These results are shown in Fig. 3.4. Taking projections at different angles (e.g. 0°, 22.5° and 90°) did not improve results. The defect was still not resolved in the SIRT and FISTA reconstructions. However, the number of atoms in the simulated annealing, ADCG and ADCG with energy reconstructions was correct. Once again, the reconstruction obtained with our algorithm was closer to the ground truth than all other reconstructions, with all but one atom placed correctly. Reconstructions with 3 projections for the interstitial point defect and edge dislocation were not significantly different from those with 2 projections. In Table 3.1, we report the number of atoms detected and (where applicable) the mean Euclidean distance of atoms from the ground truth. Note that for computing the mean distance, we required that the number of atoms detected in the reconstruction matched that in the ground truth. Thresholding with a pre-defined minimum distance between peaks was used to detect atoms in the SIRT and FISTA reconstructions.

3.4.5 Effect of adding energy to the optimization

To resolve atom positions using Algorithm 3, the contribution of the potential energy was increased gradually by increasing α. In Fig. 3.5, we show the effect of adding energy to the optimization problem. For $\alpha = 0$, an initial guess for the true configuration was obtained. This configuration, though data optimal, was not the ground truth. A quantitative measure of this mismatch is the Euclidean distance between the reconstructed atom locations and those in the ground truth. As α was increased, the reconstructions evolved from being data-optimal to being energy-optimal. At a certain value of α, the Euclidean distance between reconstructed and ground truth atom locations decreased to zero. Increasing α beyond this point led to a large increase in the data discrepancy

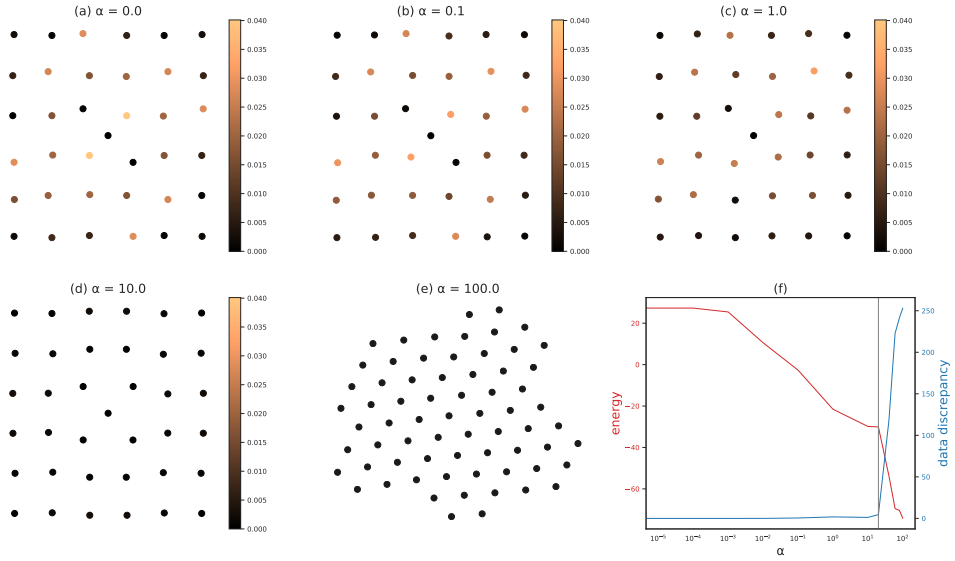


Figure 3.5: (a)-(d): Increasing the weighting of the energy term from $\alpha = 0.0$ to $\alpha = 10.0$ helps to resolve the correct atomic configuration. The reconstructed atoms are coloured according to their Euclidean distance from the atoms in the ground truth. (e) At high values of α, the reconstructions have a high data discrepancy and correspond to one of the global minima of the potential energy. (f) From the plots of potential energy and data discrepancy, an optimal value of α (indicated by the grey line) is selected. Increasing α beyond this optimal value leads to a large increase in the data discrepancy due to addition of more atoms.

term due to the addition of more atoms. For very high values of α, the configurations obtained were essentially global minima of the potential energy, such as the honeycomb configuration in Fig. 3.5(e) for $\alpha = 100.0$. An optimal value of the regularization parameter was selected by increasing α to the point at which more atoms were added to the configuration and a jump in the data discrepancy was observed.

3.5 Discussion

The results of our numerical experiments demonstrate that algorithms like ADCG, which do not rely on domain discretization, are better at resolving the defects in the atomic configurations shown in Fig. 3.1. Moreover, the output from ADCG is a *list of coordinates* and not an image like that of SIRT or FISTA, which requires further post-processing steps to derive the atom locations. Direct access to coordinates can be particularly useful

because further analysis, such as strain calculations, often require atom positions as input.

Adding the potential energy of the atomic configuration to the optimization problem resulted in reconstructions that were closer to the ground truth. One challenge of the proposed approach (with or without adding the potential energy) is that the resulting optimization problem is a non-convex function of the atom locations. The numerical methods we presented are not intentionally designed to escape local minima and are therefore sensitive to their initialization. To improve this, one important extension would be to also remove atoms from the current configuration, which might make it possible to resolve the vacancy defect in Fig. 3.3 with two projections. More generally, one would need to include suitable features of global optimization algorithms [84] that do not compromise ADCG's computational efficiency (note that we could have adapted simulated annealing to solve (3.11) but using a cooling schedule slow enough to prevent getting trapped in local minima quickly becomes practically infeasible). A related problem is to characterize local and global minimizers of (3.11) to understand which configurations can be uniquely recovered by this approach and which cannot. To process experimental data, it may furthermore be important to analyze the impact of the error caused by the approximate nature of the mathematical models used for data acquisition ($\mathcal{R}, G$) and atomic interaction (V_{LJ}).

3.6 Conclusions

In this chapter we proposed a novel discrete tomography approach in which the locations of atoms are allowed to vary continuously and their interaction potentials are modeled explicitly. We showed in proof-of-concept numerical studies that such an approach can be better at resolving crystalline defects than image domain discretized or fully discrete algorithms. Furthermore, in situations where atom locations are desired, this approach provides access to the quantity of interest without any additional post-processing. For future work, we will extend our numerical studies on this atomic super-resolution approach to larger-sized scenarios in 3D, featuring realistic measurement noise, acquisition geometries, more suitable and accurate physical interaction potentials and different atom types. This will require additional computational effort to scale up our algorithm and will then allow us to work on real electron tomography data of nanocrystals.

Chapter 4

Grid-free marker-based alignment in cryo-electron tomography

4.1 Introduction

Cryo-electron tomography (cryoET) is a powerful imaging technique to resolve the structures of biomolecules and cellular components *in situ* using an electron microscope [5]. In recent years, advancements in detector technology and image processing methods have greatly improved the resolution of structure determination routines using cryoET, down to near-atomic resolution [85].

A typical cryoET workflow consists of tilt-series acquisition, tilt-series alignment and reconstruction, followed by post-processing steps such as per-particle reconstruction refinement, segmentation and sub-tomogram averaging [86], [87].

The image formation process in cryoET is as follows. A frozen sample is inserted into a transmission electron microscope (TEM) where it is irradiated with an electron beam, and the resulting transmitted beam lands on the camera to form a TEM image. For biological samples, the observed image contrast is mainly phase contrast because such samples are made up of light materials and thus are weak scatterers [88]. In contrast, gold markers are strong scatterers and show clear image contrast even under low-dose acquisition conditions. In order to obtain a tomographic *tilt series* (i.e. a series of projection images for consecutive angles), images of the sample are acquired at different view angles by tilting the sample with respect to the electron beam.

Aspects of cryoET that distinguish it from other CT setups are as follows. Firstly, the

This chapter is based on:

SparseAlign: A Grid-Free Algorithm for Automatic Marker Localization and Deformation Estimation in Cryo-Electron Tomography. *P. S. Ganguly, F. Lucka, H. Kohr, E. Franken, H. J. Hupkes and K. J. Batenburg.* IEEE Transactions on Computational Imaging, vol. 8, pp. 651-665, 2022.

geometry of the experimental system limits the extent to which the sample can be tilted. Moreover, the increase in apparent sample thickness with increasing tilt allows projection images to only be acquired for a limited angular range in cryoET, usually in $[-60°, 60°]$, resulting in a *missing wedge* of information that is not available during reconstruction [20]. Secondly, cryoET samples are dose-sensitive, which limits the total dose during acquisition and leads to very noisy projection images when a large number are acquired. Thirdly, the sample undergoes local and global movements during the acquisition procedure, making it difficult to reconstruct with a constant sample assumption. For a detailed discussion on the mathematics of electron tomography we refer the reader to [89].

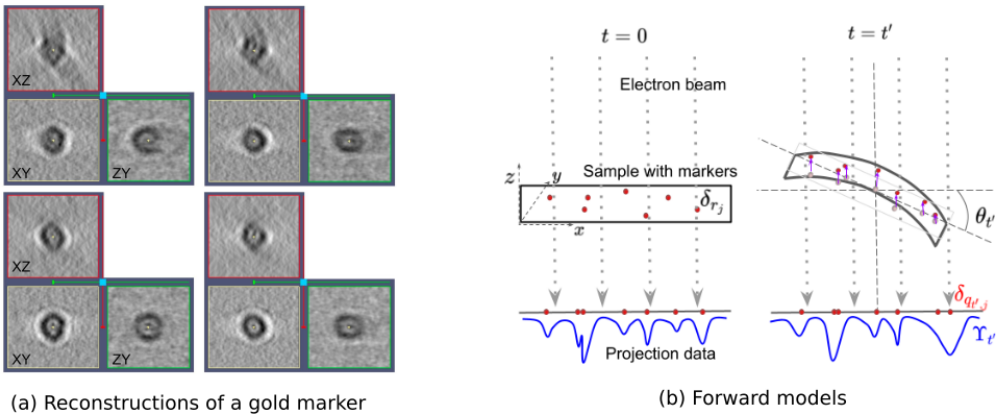
The acquired tomographic tilt series must be corrected for global and local sample motion during tilt-series acquisition [90]. Types of global motion include rotations and shifts of the sample with respect to the field-of-view (FoV) captured by the camera. Local motion includes sample deformation induced by the electron beam. In addition, a build up of surface charges due to irradiation can lead to apparent sample motion due to a microlensing effect [91]. When not corrected, sample motion leads to blurred reconstructions and poor resolution of the biological structures extracted by further post-processing [92]. *Tilt-series alignment*, the process of figuring out geometric relationships between projections in the tilt series, provides a way to correct for these effects so that the highest possible resolution can be achieved in subsequent reconstructions.

Beam-induced local sample deformation is a crucial limiting factor in high-resolution cryoET studies [93]. In particular, as shown in Fig. 4.1(a), compensation of local motion during alignment leads to sharper reconstructions and thus more reliable structure determination. In [93], the authors propose a method to extend currently used alignment methods with a sample deformation term that takes into account local sample motion induced by the electron beam. It has previously been observed that cryoET samples undergo "doming" motion, where the sample exhibits an upward deformation perpendicular to the sample plane (Fig. 4.1(b)). The authors of [93] model this motion using polynomial surfaces with coefficients that can be estimated as part of a minimization scheme. In addition to global shifts and rotations, the parameters of the doming model are fitted by solving a non-linear least-squares problem.

One of the drawbacks of the doming model approach is that it requires labelled marker locations in the tilt series as input, where the same marker has to be identified in all tilt images such that its locations can be connected to a trace. Markers are usually identified and traced in tilt-series images by template matching, a procedure that is prone to errors when the signal-to-noise ratio in tilt images is low, when markers cluster together or when they overlap in projection while being separate in 3D [90]. Other, state-of-the-art approaches in local sample deformation correction such as emClarity [94] and M [95] rely on detecting features from reconstructed tomograms and using these as fiducials, and are computationally expensive.

An additional disadvantage of the doming model method is the large number of parameters that must be estimated because no additional prior information on the deformation field is incorporated. Without smoothness constraints on the time evolution of the deformation field, the model allows deformation parameters to vary freely over the tilt series and does not penalize unphysical deformations.

Though not always appropriate, smoothness constraints on local sample motion are



(a) Reconstructions of a gold marker

(b) Forward models

Figure 4.1: (a) Reconstructions of a gold bead marker using (top two rows) standard alignment without sample deformation compensation and (bottom two rows) with sample deformation compensation. Images reproduced with permission from [93]. (b) Forward models used in SparseAlign and the doming model method. At $t = 0$ the sample with markers is not deformed. Projected marker locations (red dots) are convolved with a known shape function to yield projection data (blue line). As the sample is tilted, it undergoes doming deformation. At time $t = t'$, the change in marker locations caused by doming (purple upward arrows) leads to a change in the projection data.

reasonable in the context of continuous-tilt cryoET (CTT) data collection, where thousands of very noisy projection images are captured continuously while the stage is tilted with a constant rotation speed [96]. This allows for a reduction in the number of doming model parameters.

We propose extensions to the doming model approach that make it possible to align tilt-series images without labelling markers in the tilt series. Taking inspiration from algorithms proposed in the context of single-molecule localization microscopy [12], we use a continuous formulation of the marker localization problem, which enables us to formulate an image-based loss and identify marker locations with a localization precision greater than the pixel spacing of the acquired tilt-series data. We equip the localization scheme with an additional deformation estimation routine and solve for the parameters of the doming model.

In addition, we incorporate a polynomial time dependence of the deformation field, which assumes smoothness of the local sample motion after global motion correction. This assumption is motivated by the fact that local sample motion is the result of positive-charge accumulation on the sample due to irradiation with a high-energy electron beam [92], [97]. As charge accumulation happens continuously and smoothly over the acquisition time, we can assume that local sample motion is also smooth. This assumption helps us reduce the number of deformation parameters by orders of magnitude. An important aspect of our approach, however, is that it is independent of the choice of deformation field parametrization.

To validate our proposed method, we apply it to simulated data in 2D and 3D as well as experimental data containing gold markers on ice. As the main focus of our chapter is on testing the properties and robustness of our proposed method, we focus on simulation studies with ground-truth marker locations and deformation fields. In experimental studies, we restrict ourselves to data of gold markers on ice to disentangle the marker localization and deformation estimation problem from the later image reconstruction problem. We study the robustness of our approach with respect to noise, forward model mismatch and deformation model mismatch. We show that we are able to estimate deformation fields and marker locations with similar accuracy as the doming model approach without the need for labelled marker data, and that our method estimates deformation parameters accurately despite model mismatch.

This chapter is structured as follows. In Section 4.2, we review the mathematical formulation of the alignment problem and discuss a unifying framework for solving it. We derive the doming model approach in [93] as *one* possible choice of alignment method. We also present the main contribution of our chapter: a method that localizes markers and estimates deformation fields without marker labelling. In Section 4.3, we give details of the optimization techniques used to solve our extended problem. In Section 5.4, we describe the numerical experiments performed, and discuss our results on 2D and 3D simulated data as well as experimental data in Section 4.5. We end our chapter with a critical discussion of our approach and point to possible extensions in Section 4.6.

4.2 Mathematical formulation

We consider an initial sample $u_0(\rho)$, with $\rho \in \Omega \subset \mathbb{R}^d$ ($d = 2, 3$ for simulated data and $d = 3$ for experimental data), which consists of two distinct components with non-overlapping supports:

$$u_0(\rho) = u_0^m(\rho) + u_0^s(\rho), \tag{4.1}$$

where $u_0^m(\rho)$ represents markers and $u_0^s(\rho)$ represents the biological sample in the background.

This initial sample deforms over time, in the sense

$$u_t(\rho) = u_0(\rho - D_t(P)(\rho)) =: \mathcal{W}_{D_t(P)} u_0(\rho), \tag{4.2}$$

where $D_t(P, \rho) : \mathcal{P} \times \Omega \to \mathbb{R}^d$ is a time- and space-dependent deformation field parametrized by global parameters $P \in \mathcal{P}$. The action of this deformation field can be represented by a linear warping operator $\mathcal{W}_{D_t(P)}$. The global deformation parameters couple the reconstruction problems for individual markers. Later in this section we discuss appropriate parametrizations for the deformation field.

Projection data Ψ_t of the deforming configuration are generated by applying the continuous Radon transform to $u_t(\rho)$:

$$\Psi_t = \mathcal{R}_{\theta_t} u_t(\rho) = \mathcal{R}_{\theta_t} \mathcal{W}_{D_t(P)} \left(u_0^m + u_0^s\right), \tag{4.3}$$

where θ_t is the projection angle and the Radon transform for $d = 2$ is defined as a line

integral over rays:

$$\mathcal{R}_{\theta_t}[u](s) = \int_{l(s,\theta_t)} u(\rho)\, d\rho$$
$$l(s,\theta_t) = \{(x,y) \in \mathbb{R}^2 \,|\, x\cos\theta_t + y\sin\theta_t = s\}.$$

Projection in 3D for a parallel beam geometry, as in the case for cryoET, can be decomposed into a series of 2D projections [45].

The full tomographic data, obtained over discrete time points $t \in \{t_0, t_1, \ldots, t_T\}$ is a stack of individual projections:

$$\Psi := \begin{bmatrix} \Psi_0 \\ \Psi_1 \\ \cdots \\ \Psi_T \end{bmatrix} = \begin{bmatrix} \mathcal{R}_{\theta_1}\mathcal{W}_{D_0(P)} \\ \mathcal{R}_{\theta_2}\mathcal{W}_{D_1(P)} \\ \cdots \\ \mathcal{R}_{\theta_T}\mathcal{W}_{D_T(P)} \end{bmatrix} (u_0^m + u_0^s). \tag{4.4}$$

Solving the set of equations (4.4) when all the variables - u_0^m, u_0^s and D_t - are unknown amounts to solving a joint image reconstruction and alignment problem. Most approaches for solving the joint problem alternate between solving (4.4) for one of the three variables while keeping the others fixed. In such schemes, determining a good order for these updates is crucial.

As markers are designed to have a significantly higher contrast compared to the sample, we can often obtain reasonable first estimates for the marker configuration u_0^m and deformations D_t while ignoring the sample contribution. This corresponds to solving (4.4) by setting $u_0^s = 0$.

One way to parametrize the initial marker configuration u_0^m is to represent it using the continuous locations of markers at $t = 0$. Here we represent a single marker as a delta function at the location of its centre convolved with a fixed, known shape function; the marker configuration is then a sum of convolved delta functions in $\Omega \subset \mathbb{R}^d$:

$$u_0^m(x) = \sum_{j=1}^{M} \Big(G * \delta_{r_j}(\rho)\Big), \tag{4.5}$$

where r_j are the initial marker locations, M is the total number of markers and G is a known shape function, for instance a Gaussian.

For parallel beam projection, Theorem 1.2 in [45] states that:

$$\mathcal{R}_\theta(G * \delta_{r_j}(\rho)) = (\mathcal{R}_\theta G) * \big(\mathcal{R}_\theta \delta_{r_j}(\rho)\big) =: G_\theta^p * \big(\mathcal{R}_\theta \delta_{r_j}(\rho)\big)\,. \tag{4.6}$$

Furthermore, the Radon transform of a delta function is a delta function in projection space:

$$\mathcal{R}_\theta \delta_{r_j}(\rho) = \delta_{A_\theta r_j}(s), \tag{4.7}$$

where $A_\theta \in \mathbb{R}^{(d-1)\times d}$ is a projection matrix that maps marker locations in configuration space to locations in projection space. We denote the resulting projected marker locations by $q_j := A_\theta r_j$.

We can assume that in contrast to the sample, markers are displaced over time, not deformed. Furthermore, when variations in the global deformation field D_t over the area covered by a marker are small, we can make the following approximation by commuting the deformation operator with convolution with the shape function:

$$\begin{aligned}\mathcal{W}_{D_t(P)}(G * \delta_r)(\rho) &= (G * \delta_r)(\rho - D_t(P,\rho)) \\ &\approx G * \delta_r(\rho - D_t(P,\rho) = G * \delta_{r+D_t(P,\rho)}(\rho).\end{aligned}$$

Thus, the deformed marker configuration is given by:

$$\mathcal{W}_{D_t} u_0^m(x) \approx \sum_{j=1}^{M} \Big(G * \delta_{r_j + D_t(P,r_j)}(\rho) \Big). \tag{4.8}$$

This assumption is accurate when the support of G is small and the deformation $D_t(P,\rho)$ is smooth over the support of $(G * \delta_{r_j})$. Setting $u_0^s = 0$ and inserting the ansatz above into (4.3) yields

$$\begin{aligned}\Psi_t = \mathcal{R}_{\theta_t}\mathcal{W}_{D_t(P)}u_0^m &\approx \sum_{j=1}^{M} \Big(G^p_{\theta_t} * \delta_{A_{\theta_t}(r_j + D_t(P,r_j))} \Big) \\ &= \sum_{j=1}^{M} \Big(G^p_{\theta_t} * \delta_{q_{t,j}} \Big),\end{aligned} \tag{4.9}$$

where

$$q_{t,j} = A_{\theta_t}(r_j + D_t(P, r_j)). \tag{4.10}$$

Using equation (4.9) amounts to localizing markers by matching their projection data $\Psi_t \in \mathbb{R}^{(N_\theta \times N_d)}$ (in 2D), where N_θ is the number of projection angles and N_d is the discretization of the detector plane. A schematic of this forward model is shown in Fig. 4.1(b), where we indicate 1D projected data with blue lines.

In [93], the authors use projected marker locations over time as the input instead of image data (indicated with red dots in Fig. 4.1(b)) and use the following optimization problem for deformation estimation and marker localization:

$$\underset{r_j, P}{\text{minimize}} \quad \sum_{t=0}^{T}\sum_{j=1}^{M} \Big\| \Big(\tilde{q}_{t,j} - A_{\theta_t}(r_j + D_t(P, r_j)) \Big) \Big\|_2^2. \tag{4.11}$$

Such an approach assumes that we can identify the projected marker locations $\tilde{q}_{t,j}$ directly, despite convolution with $G^p_{\theta_t}$. Here and elsewhere, we use symbols with a tilde (e.g. $\tilde{q}_{t,j}$) to denote measured data and symbols without a tilde (e.g. $q_{t,j}$) to denote model predictions.

Comparing equations (4.9) and (4.10), we find that for each t the dimensions of 2D data for (4.10) are $d \times M$ and those of the data for (4.9) are $N_\theta \times N_d$. Typical values for d, M, N_θ and N_d are $3, 20, 100$ and 4096, respectively, such that $d \times M = 3 \times 20$ and $N_\theta \times N_d = 100 \times 4096$, the latter being approximately 6000 times the former. Thus, (4.10) is a much lower-dimensional problem. Furthermore, the deformation field can be

extracted from (4.10) in a more direct fashion as it directly describes the corresponding projected marker displacement, not the change in the projection image caused by it.

However, identifying markers robustly from data is not a trivial problem [90]. It involves solving an optimization problem of the form:

$$\underset{q_{t,j}}{\text{minimize}} \sum_t \|\tilde{\Psi}_t - \sum_j (G^p_{\theta_t} * \delta_{q_{t,j}})\|_2^2.$$

Marker labelling is generally performed using normalized cross-correlation-based schemes or template matching algorithms. Such methods are error-prone when projection data are noisy or when gold beads are occluded or cluster together in projection data. In such situations, users must manually annotate markers, or manually inspect and correct for incorrect and failed detection in one or more images in the tilt series. This manual intervention leads to time-consuming and subjective labelling.

To avoid solving the marker identification problem, we take a step back and start directly from (4.9). We solve for marker locations and the deformation field in a least-squares sense. In addition, we do not assume that we know the number of markers beforehand. The resulting optimization problem is as follows:

$$\underset{r_j,P,M}{\text{minimize}} \quad \sum_{t=0}^{T} \left\| \tilde{\Psi}_t - \sum_{j=1}^{M} \left(G^p_{\theta_t} * \delta_{A_{\theta_t}(r_j + D_t(P, r_j))} \right) \right\|_2^2. \tag{4.12}$$

The optimization problem above assumes a model for the markers, uses an image-based loss and does not need labelled marker locations like the problem in (4.11). In the following section, we discuss optimization schemes for solving (4.12).

The deformation field D_t can be represented using different basis functions. If one uses localized basis functions, e.g. the B-spline basis functions often used in non-rigid image registration, one either needs a sufficiently dense sampling of the domain with markers or include suitable regularization constraints [98]. Global basis functions that are supported in the entire domain will only lead to a compact, low-dimensional description of the deformation field with sufficient accuracy if they are chosen based on *a priori* knowledge about the sample deformation.

In this chapter, we use the global basis functions proposed in [93], where the beam-induced sample deformation is modeled with a set of polynomial surfaces. The parametrized sample deformation $D_t(P, r_j) := [D_{t,x}, D_{t,y}, D_{t,z}]$ is modelled with polynomials in (x, y, z) such that the deformation in each direction is given by

$$D_{t,k}(r, P) = \sum_{\substack{\alpha,\beta,\gamma \geq 0 \\ \alpha+\beta+\gamma \leq d_p}} \left(P_{\alpha\beta\gamma}(t) \right)_k x^\gamma y^\beta z^\alpha, \quad k \in \{x, y, z\}, \tag{4.13}$$

where $P_{\alpha\beta\gamma}$ are the coefficients of the polynomial and d_p is the degree of the polynomial. In [93], these polynomials are allowed to vary freely over the tilt series, resulting in a large number of free parameters. In 3D, we must estimate 18 parameters for each tilt for a quadratic deformation model, which amounts to thousands of parameters when the number of tilts is high. One way to reduce the number of parameters, used in [93], is

by assuming that the deformation field is constant along the depth (z direction) of the sample. with $\frac{(d_p+2)(d_p+1)}{2}$ free parameters.

To further reduce the number of free parameters, we introduce a temporal dependence in (4.13), which reduces the number of parameters from 18 for each tilt to 18 for the entire tilt series, assuming a quadratic deformation model. Our time-dependent deformation field is given by:

$$D_{t,k}(r,P) = \sum_{\zeta=1}^{d_t} \sum_{\substack{\alpha,\beta,\gamma\geq 0 \\ \alpha+\beta+\gamma\leq d_p}} \left(P_{\alpha\beta\gamma\zeta}\right)_k x^\alpha y^\beta z^\gamma t^\zeta, t \in [0,1]. \tag{4.14}$$

As we reconstruct the first image, there is no way to recover a zeroth order deformation in time. For simplicity, we consider linear time dependence in our experiments, which amounts to setting $d_t = 1$.

Our method is independent of the choice of parametrization of the deformation field. Other parametrizations, which take advantage of the possible symmetries of the deformation field or additional understanding of the physics underlying the sample behaviour, could also be suitable choices.

4.3 Optimization

In [12], [72], [99], convex approximations to the minimization problem (4.12) have been devised by mapping the problem onto the space of measures $\mathcal{M}(\Omega)$. We interpret the marker configuration as a measure $\mu := \sum_{j=1}^{M} w_j \delta_{r_j} \in \mathcal{M}(\Omega)$, where the weights w_j are introduced as a means of relaxing the optimization problem (4.12). The weights determine the relative "importance" of the markers and, as we show later, can be used to remove candidate markers that do not contribute significantly to the data. Mapping the problem to measure space enables us to express the forward operation shown in (4.9) in terms of a linear operator, $\Phi_t : \mathcal{M}(\Omega) \to \mathbb{R}^{N_d}$:

$$\Psi_t = \sum_{j=1}^{M} w_j \left(G^p_{\theta_t} * \delta_{q_{t,j}}\right) =: \Phi_t \mu, \qquad \Psi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \dots \\ \Phi_T \end{bmatrix} \mu =: \Phi\mu \tag{4.15}$$

The minimization problem (4.12) can then be rewritten as the following problem in the space of measures, where the loss is convex in the measure μ:

$$\underset{\mu\in\mathcal{M}(\Omega)}{\text{minimize}} \quad \ell(\Phi\mu - \tilde{\Psi}), \qquad \ell(\cdot) := \|\cdot\|_2^2 \tag{4.16}$$

In [12], the authors devised an effective numerical scheme for solving infinite-dimensional convex problems of the type shown above by using a variant of the conditional gradient or Frank-Wolfe method [19]. They also showed that interleaving the convex Frank-Wolfe iterations with nonconvex local optimization steps improved the convergence of the algorithm. This algorithm, known as the alternating descent conditional gradient (ADCG)

method, has been subsequently extended for and applied to a range of application areas [12], [72], [99].

In this chapter, we adapt the ADCG algorithm to solve the marker localization and deformation estimation problems simultaneously. To do this, we perform the Frank-Wolfe iterations as-is but modify the block coordinate descent routine to include an additional deformation estimation step. At each iteration of the algorithm, we place a new marker at a candidate initial location by solving a linearized approximation of our optimization problem. Then, we solve a linear optimization problem to obtain estimates for the weights of all current markers. Local optimization routines are used to solve for the parameters for the deformation field and to refine the marker support in a bounded region. Our modified ADCG routine, which we call SparseAlign, is shown in Algorithm 4. Below we describe each step in our method in detail.

Algorithm 4 SparseAlign

for $n = 1 : n_{\text{max}}$ **do**

1) Compute current residual: $\varrho_n \leftarrow \Phi\mu_n - \tilde{\Psi}$

2) Find next marker: $r_n^* \leftarrow \arg\min_{r \in \text{grid}} \langle \nabla \ell(\varrho_n), \Psi(r) \rangle$

3) Update support: $\boldsymbol{r}_{n+1} \leftarrow [\boldsymbol{r}_n, r_n^*]$

4) Block coordinate descent:

Repeat:

(a) Compute weights:
$w_{n+1} \leftarrow \arg\min_w \ell(\Phi\mu_{n+1} - \tilde{\Psi})$

(b) Prune support:
$(w_{n+1}, \boldsymbol{r}_{n+1}) \leftarrow \texttt{prune}(w_{n+1}, \boldsymbol{r}_{n+1})$

(c) Fit deformation parameters:
$P_{n+1} \leftarrow \arg\min_{P \in \mathcal{P}} \ell(\Phi\mu_{n+1} - \tilde{\Psi})$

(d) Improve support:
$\boldsymbol{r}_{n+1} \leftarrow \arg\min_{\boldsymbol{r} \in \mathcal{C}} \ell(\Phi\mu_{n+1} - \tilde{\Psi})$

end for

Adding candidate marker locations We use the conditional gradient method to obtain candidate marker locations in steps 2-3. The conditional gradient or Frank-Wolfe method [19] can be used to solve constrained optimization problems of the type $\text{minimize}_{x \in C} f(x)$ iteratively, where C is a convex set. The first step in each iteration is to minimize a linearized version of the loss within a specified domain. The linear approximation to a function $f(x)$ at x_k is given by

$$f_{\text{lin}}(s) = f(x_k) + \langle \nabla f(x_k), s - x_k \rangle.$$

Minimizing $f_{\text{lin}}(s)$ over a domain $\mathcal{D}_s$ thus amounts to solving

$$\underset{s \in \mathcal{D}_s}{\text{minimize}} \quad \langle \nabla f(x_k), s - x_k \rangle.$$

Using our forward model (4.15) and the loss function in (4.16), we can compute that the linear minimization step at iteration n is the following optimization problem over

measures $s \in \mathcal{M}_s(\Omega) \subset \mathcal{M}(\Omega)$

$$\underset{s \in \mathcal{M}_s(\Omega)}{\text{minimize}} \quad \langle \nabla \ell(\varrho_n), \Phi s \rangle, \tag{4.17}$$

where $\varrho_n := \Phi \mu_n - \tilde{\Psi}$ is the residual at iteration n.

An optimal solution of the above problem is the addition a single new marker with positive weight to the current support of μ_n. This ensures that, at iteration n of the algorithm the measure μ is supported at n points. Adding only one location at a time has been shown to give the sparsest possible solution [12].

Practically, we solve (4.17) by gridding the domain of marker locations coarsely. The contribution of a single marker at each grid point, r_{grid}, is computed for a current guess of deformation parameters:

$$\psi(r_{\text{grid}}) = \begin{bmatrix} G^p_{\theta_t} * \delta_{A_{\theta_1}(r_{\text{grid}} + D_1(r_{\text{grid}}))} \\ G^p_{\theta_t} * \delta_{A_{\theta_2}(r_{\text{grid}} + D_2(r_{\text{grid}}))} \\ \cdots \\ G^p_{\theta_t} * \delta_{A_{\theta_T}(r_{\text{grid}} + D_T(r_{\text{grid}}))} \end{bmatrix}$$

Then, the inner product of the current residual with the forward projection of a marker located at each grid location is calculated. The grid location r^*_{grid} with the smallest inner product with the residual is chosen as the next candidate location:

$$r^*_{\text{grid}} = \arg\min_{r \in \text{grid}} \quad \langle \nabla \ell(\varrho_n), \psi(r) \rangle. \tag{4.18}$$

Optimizing weights Once we have optimized for marker locations, we can optimize the weights of each marker as shown in steps 4(a)-(b). Note that the model (4.15) depends linearly on the weights w_j, $j \in \{1, 2, \ldots, M\}$. Thus, with the number of markers, marker locations and deformation parameters fixed, the weights w_j can be estimated by solving the following linear least-squares problem

$$\underset{w \in [0,1]^n}{\text{minimize}} \quad \|\ell(\Phi \mu_n - \tilde{\Psi})\|_2^2. \tag{4.19}$$

All weights w_j are constrained to lie in $[0, 1]$ and represent the relative importance of marker contributions to the data. Markers with weights close to zero can be removed by an additional `prune` routine that removes all markers with a weight lower than a predefined threshold. In some cases an additional `prune` routine can be used to remove markers with small weights at the end of a full algorithm run. This further ensures that the solution obtained is the sparsest possible marker configuration required to explain the data $\tilde{\Psi}$.

Refining initial marker locations At each iteration, we perform the nonconvex local optimization step shown in 4(d) to refine our estimates for the initial marker locations. This step was first proposed in [12] as a way to speed up convergence of the conditional gradient method.

Refining the support of the current measure μ_n without changing the number of markers ensures that markers are moved off the grid locations used in steps 2-3. It also

imparts some of the rapid local convergence qualities of nonconvex optimization [12]. In our implementation, we use the L-BFGS-B algorithm to perform local optimization over initial marker locations.

Estimating deformation parameters The optimization problem behind step 4(c) is given by

$$\underset{P\in\mathcal{P}}{\text{minimize}} \quad \sum_{t=0}^{T}\left\|\tilde{\Psi}_t - \sum_{j=1}^{M} w_j\left(G^p_{\theta_t} * \delta_{A_{\theta_t}(r_j+D_t(r_j,P))}\right)\right\|_2^2, \tag{4.20}$$

which is a difficult nonconvex problem that is often studied in the context of image correspondence problems such as image registration or optical flow estimation [100]. We use L-BFGS-B initialized at the current P_n to compute a local update P_{n+1} for the parameters of the deformation field.

Coarse-to-fine scheme for large data One of the challenges of solving (4.20) is that the objective function is flat if the forward projection of the current marker configuration and the data do not share the same support, and gradient-based optimization schemes such as L-BFGS-B have a hard time locating a minima. This easily happens for small objects, such as markers, embedded in large projection images. The remedy is typically to smooth both images with a Gaussian, compute a deformation field on the smoothed problem, and use the solution of the smoothed problem to initialize the optimization of the original problem.

Gaussian smoothing followed by downsampling removes high image frequencies and one starts matching only the low frequencies. For noisy data, downsampling has the additional advantage of denoising the data. Furthermore, for large experimental data, where each tilt image has pixel dimensions 4096×4096, warm-starting the optimization at high resolutions with good initial values ensures that not many expensive iterations have to be performed.

For realistic simulation data and experimental data, we use a coarse-to-fine scheme where the marker localization and deformation estimation problem is solved at successively finer resolutions using the results at the coarser resolutions as initialization.

At full resolution, we generate the forward projection of a single marker using (4.6) followed by sampling on a spatial grid X_f with N_d grid points. Thus, the discretized forward projection of the full marker configuration can be written as

$$\Psi_t = \sum_j w_j S^f \mathcal{G}_{(q_{t,j},\tau_f)}, \tag{4.21}$$

where S^f is the sampling operator associated with the spatial grid X_f and $\mathcal{G}_{(q_{t,j},\tau_f)}$ is a Gaussian centred at $q_{t,j}$ with standard deviation τ_f.

For obtaining measured data at coarse resolutions, we downsampled the full-resolution measured data $\tilde{\Psi}_t$ at each time after Gaussian convolution to prevent aliasing artefacts [101]. Thus, the coarse-resolution data were given by $\tilde{\Psi}^c_t := \mathcal{H}^c(\mathcal{G}_{\tau_a} * \tilde{\Psi}_t)$, where $\mathcal{H}^c$ is a downsampling operator associated with a coarse grid X_c and $\mathcal{G}_{\tau_a}$ is an anti-aliasing Gaussian. For integer downsampling factors $\eta := |X_c|/|X_f|$, $\mathcal{H}^c$ only keeps pixels separated by η in the coarse-resolution image.

We approximated matching forward projection data Ψ_t^c directly from marker locations using our forward model (4.9) by sampling the Gaussian-convolved projected marker locations on the coarse grid X_c:

$$\Psi_t^c = \sum_j w_j S^c \mathcal{G}_{(q_{t,j},\tau_f)}, \tag{4.22}$$

where S^c is the sampling operator associated with the coarse grid X_c.

4.4 Numerical experiments

In this section we describe our experiments with simulated and real data. Implementation notes with details of software packages used are provided in Section 4.4.6 of the Supplementary Materials.

4.4.1 Illustrative 2D example

Ground truth We used a simple simulated sample to elucidate properties of our algorithm in 2D. The FoV was taken to be $[-L/2, L/2]$ along both axes, with the canonical length scale $L = 1$. The ground truth sample consisted of 10 gold bead markers confined to a thin rectangular region: $x \in [-2L/5, 2L/5], z \in [-L/10, L/10]$. We chose this geometry for our 2D sample to mimic the geometry of experimental cryoET samples.

For simplicity, we considered deformation field components to be zero along the horizontal (x) direction. In the vertical (z) direction, we assumed the deformation to be given by a quadratic polynomial of x and z:

$$D_{t,z}(r, P) = (P_0 + P_1 x + P_2 z + P_3 x^2 + P_4 z^2 + P_5 xz)t =: D_{1,z}t, \tag{4.23}$$

with $P_0 = 0\ L$, $P_1 = P_2 = -1$, $P_3 = P_4 = P_5 = -1\ L^{-1}$, and t taking values in $[0, 1]$

Projection data We generated projection data using the forward model in (4.15) over a set of discrete projection angles $\theta \in [-70°, 70°)$, $N_\theta = 20$. Practically, we computed the continuous Radon transform of each marker, followed by a continuous 1D Gaussian convolution in projection space. The Gaussian-convolved projection was then discretized on a detector grid with $N_d = 64$. At each projection angle, the projection was then a 1D profile. All the projections were rearranged in a sinogram with dimensions $N_\theta \times N_d$.

For comparison, we also generated input data for the doming model method in [93]. These data were the projected locations of each marker over the same series of projection angles.

4.4.2 Simulated 3D examples

Ground truth We used a 3D configuration of markers to test the robustness of our method to noise and to mismatches in the forward model. We used 20 randomly placed

markers in a thin region in 3D with dimensions 819.2 nm $\times$ 819.2 nm $\times$ 100.0 nm. The sample used was the same as that described in 4.4.3.

We considered deformation field components to be non-zero only along the z direction; this component was then given by:

$$D_z(x, y, z, t) = (P_0 + P_1 x^2 + P_2 y^2)t, \tag{4.24}$$

with $P_0 = 200$ nm, $P_1 = P_2 = -100$ nm^{-1}, and t taking values in $[0, 1]$.

Projection data We generated projection data along 140 equispaced projection angles in $[-70°, 70°]$ using a Gaussian with standard deviation 15nm as the shape function of individual markers. Each projection image was discretized on a 64×64 pixel grid.

To convert the intensities in these generated images to meaningful electron counts, we used that the expected electron count in any pixel is given by $I = I_0 e^{-V_{\text{abs}} C \times \delta x}$, where I_0 is the incoming electron count, V_{abs} is the absorption potential of gold nanoparticles ($5.39V$ for a $300keV$ electron beam, treating the gold as amorphous), C is the interaction constant ($0.00653V^{-1}nm^{-1}$ at $300keV$) and δx is the path length travelled by electrons through a gold marker. This path length is equal to the product of the diameter of the gold bead, which we take to be 15nm, and the intensity in our generated images. For our experiments, we generated data with $I_0 = 2^n, n \in \{6, 7, 8, 10, 12, 14\}$.

Gaussian noise To test the properties of our approach for noisy data, we performed experiments with data corrupted with additive Gaussian noise, such that

$$\Psi_{\text{noisy}} = \Psi_{\text{clean}} + \mathcal{N}(0, \sigma^2_{\text{noise}}),$$

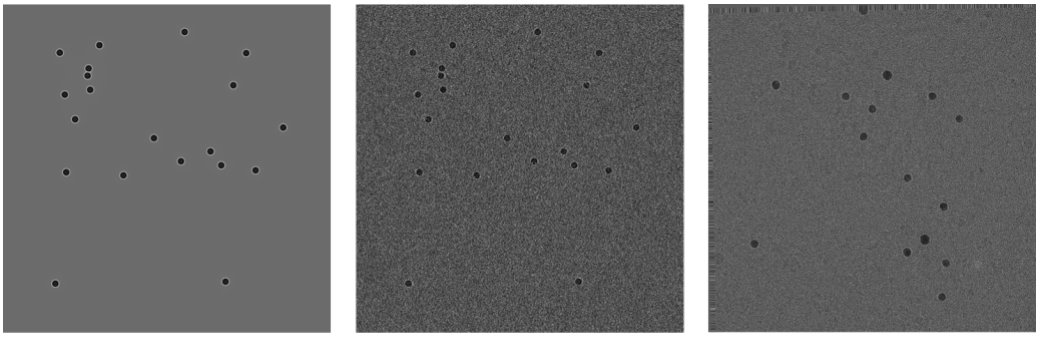
where Ψ_{clean} are the data scaled to physical electron counts and σ^2_{noise} is the variance of the noise added.

We performed experiments using $\sigma^2 = 2^n$, $n \in \{7, 8, 10, 12, 14\}$. For each noise setting, multiple independent experiments were performed and the results were averaged to obtain mean values for the metrics. Each independent experiment was initialized with a with a different random seed.

Poisson noise We also generated a series of Poisson noise-corrupted data by varying the electron count per pixel per frame, I_0. For $I_0 = 2^n$, $n \in \{6, 8, 10, 12, 13, 14\}$, we generated Poisson-distributed electron counts at each pixel using:

$$\Psi_{\text{noisy}} = \text{Poi}(\Psi_{\text{clean}}), \tag{4.25}$$

where Ψ_{clean} are the data scaled to physical electron counts and $\text{Poi}(\cdot)$ denotes a Poisson random variable. The Poisson-noise data were generated to have comparable signal-to-noise ratios as those of the Gaussian-noise data. For each noise instance, we performed multiple independent experiments with different random seeds and averaged over the obtained metrics.



(a) Noise-free simulated tilt image

(b) Noisy simulated tilt image

(c) Experimental tilt image

Figure 4.2: Example tilt images generated using TEM-simulator (a) without noise and (b) with added correlated noise; (c) an experimental TEM image showing gold beads on vitrified ice.

4.4.3 Realistic TEM simulations

We used the TEM-simulator software [102] to generate physically plausible simulations of TEM images from a specification of a 3D sample (see example projection images in Fig. 4.2(a) and (b)). To simplify matters, the sample consisted purely of gold particles in vacuum, thus disregarding the ice buffer and other sample structures. The purpose of this numerical experiment was to test our algorithm in situations where its forward model did not match the one used for data generation. In particular, the explicit assumption of Gaussian shape of gold particles and the implicit assumption of additive uncorrelated noise characteristics were violated.

The test sample consisted of 20 gold particles of 15nm diameter, randomly distributed in a slab of dimensions $819.2\text{nm} \times 819.2\text{nm} \times 100.0\text{nm}$ in x, y, z space. Over time, this sample was simulated to undergo a deformation described by the vector field

$$D_z(x, y, z, t) = (P_0 + P_1 x^2 + P_2 y^2)t, \quad D_x = D_y \equiv 0 \tag{4.26}$$

with $P_0 = 200$ nm, $P_1 = P_2 = -100$ nm^{-1}, and t taking values in $[0, 1]$. This amount of deformation (200 nm at $x = y = 0,\ t = 1$) is an exaggerated version of a doming motion observed in practice. The large amplitude was chosen to make the effects under investigation easier to observe.

Assuming constant tilt speed, the time t was mapped to a tilt angle θ according to $\theta_i = -70^\circ + t_i \cdot 140^\circ,\ t_i = \frac{i}{140}, i = 0, \ldots, 140$. At each tilt angle, a projection image was simulated according to the weak phase object approximation model [88], taking the contrast transfer function (CTF) of the optical system into account (see [102] for details). We used electrostatic potential values of $V = 0$ for vacuum and $V = (29.87 + \imath \cdot 5.39)$ Volt for (amorphous) gold. The CTF parameters were chosen as $\Delta z = 8\ \mu$m (defocus), $C_C = 2.7$ mm (chromatic aberration) and $C_S = 2.7$ mm (spherical aberration).

The size of each projection image was chosen equal to the $x - y$ dimensions of the sample, subdivided into $(N_x, N_y) = (512, 512)$ pixels, each of size 1.6 nm. Simulated

data were generated with 8x binning, with the full resolution pixel size equal to 0.19 nm. Binning was performed because of computational convenience.

Noiseless data The noiseless images generated by TEM-Simulator correspond to probability densities of detecting an electron at a given location in the detector plane. Therefore, scaling with the average number of incoming electrons per pixel area results in each pixel value representing the expected number of electrons measured in that pixel, also referred to as "infinite dose" case.

Noise generation In a real experiment, a finite number of electrons interacts with the sample and is detected at the camera. This process was modeled with a Poisson random variable $\mathrm{Poi}(\lambda_k)$ per pixel, where the parameter $\lambda_k = I_0 \Psi_k$ equals the intensity of the $k-$th pixel in the scaled noiseless image. This noise model applies to a perfect counting camera. However, cameras operating in integration mode have a nontrivial point spread function because charge from one incident electron can leak into neighboring pixels, triggering multiple detection events. Furthermore, signal and noise transfer vary with spatial frequency. These two effects are characterized by the MTF (modulation transfer function) and DQE (detective quantum efficiency) of the camera and lead to signal blur and noise correlation [88]. The noisy images in these numerical experiments made use of this model.

Pre-processing for noisy data For data with correlated Poisson noise, we performed the following pre-processing steps. First, we used noiseless data to perform segmentation with Otsu's method [52]. We obtained a mask for the markers in the tilt series from this segmentation procedure, which we used to compute average background and marker intensities in the noisy tilt series. Second, we shifted the range of the noisy data by subtracting its minimum value and applied the Anscombe transform to our shifted data. Our forward model (4.15) assumes that the intensity in the background of a projection image is mean zero with constant variance and the intensity at gold beads is mean one with constant variance. The variance of data with Poisson noise varies with the mean, and thus differs from the assumption in our forward model. To reduce the discrepancy between our model assumptions and the simulated data, we used the Anscombe transform

$$\text{Anscombe}(\tilde{\Psi}) := 2\sqrt{\tilde{\Psi} + 3/8}$$

as a variance-stabilizing transformation to obtain data with an approximately constant variance and standard deviation [103]. Finally, we subtracted the average background intensity and divided by the average bead intensity in the data.

4.4.4 Experimental data

For our experimental data we used a sample with gold beads as the only prominent features. We deposited 20nm gold particles on a lacey carbon grid, which was plunge-frozen in liquid ethane using a Thermo Scientific Vitrobot. An example tilt image is shown in Fig. 4.2(c).

We acquired a tomographic tilt series using the Thermo Scientific Tomography 5.5 software package on a Thermo Scientific Titan Krios electron microscope equipped with a Thermo Scientific Falcon 3EC camera. An area in a hole with 15 gold beads was selected. A magnification of 37000x was chosen for a pixel size of 1.949 and a field of view of 800 nm. The sample was tilted from -60 to +60 degrees with a tilt step of 2 degrees. Each image in the tilt series had an electron dose of 0.198 $e^-/^2$.

Cross-correlation-based global alignment Projection images were globally shift-aligned using the cross-correlation-based routine in Thermo Scientific Inspect3D.

Data pre-processing Not all projections were globally aligned correctly using the cross-correlation-based alignment routine. We inspected the tilt series visually for any misaligned projections and removed these. This resulted in a total of 27 projections that were then used for estimating local sample deformation. Next, we deleted 256 pixels from each of the four borders of the tilt series images to get rid of missing image data added by the cross-correlation-based alignment routine. Only one marker, near the top edge of the tilt series images, was discarded because of edge removal. As we expected correlated Poisson noise in these data, we applied the Anscombe transform to the raw tilt series to obtain data with approximately constant variance. After applying the Anscombe transform, we subtracted the mean of the tilt series; because most pixels were background pixels, this ensured that the average background intensity was close to 0. Finally, all tilt series pixels were divided by the average marker intensity to ensure that, in accordance with our forward model, the markers had an average intensity of approximately 1. To determine the average bead intensity in experimental data, we inspected the tilt series visually and used the average intensity in three small square regions around three beads.

4.4.5 Evaluation criteria

To quantify the accuracy of our estimated deformation fields with respect to the ground truth, where available, we used the following evaluation criteria. First, the estimated and ground truth deformation parameters were used to compute the deformation field at $t = 1$ on a gridded FoV of dimensions 1000×1000 (for 2D) and $1000 \times 1000 \times 1000$ (for 3D), using equation (4.23). Next, the vectorial difference between estimated and ground truth deformation fields at $t = 1$ was computed at each grid point:

$$E(r_{\text{grid}}) = \|D^{\text{gt}}_{1,z}(r_{\text{grid}}) - D^{\text{est}}_{1,z}(r_{\text{grid}})\|_2^2 \tag{4.27}$$

This deformation estimation error was averaged over the whole grid to obtain the global deformation estimation error and averaged only at the ground-truth marker locations to obtain the deformation estimation error at markers:

$$E_{\text{global}} = \frac{1}{N_{\text{grid}}} \sum_{\text{grid}} E(r_{\text{grid}}) \tag{4.28}$$

$$E_{\text{markers}} = \frac{1}{M} \sum_{j=1}^{M} E(r_j) \tag{4.29}$$

where $N_{\text{grid}} = 10^9$ for 3D and $N_{\text{grid}} = 10^6$ for 2D.

4.4.6 Implementation details

We implemented SparseAlign in Python 3.6 and used several Python packages for each subroutine. Marker location refinement was performed using automatic differentiation routines in `Autograd` [104] and the L-BFGS-B method implemented in `SciPy` [105]. The L-BFGS-B routine in `SciPy` was also used for deformation estimation.

We generated Gaussian noise and Poisson noise for 3D simulated data using `NumPy` [49]. Segmentation using Otsu's method was performed for 3D simulated data using pre-defined functions in `scikit-image` [44].

All experiments were performed using JupyterLab notebooks [106] on an Intel(R) Core(TM) i7-8700K CPU with 12 cores. Example code for 2D alignment using SparseAlign is available in the repository https://github.com/poulamisganguly/SparseAlign/.

4.5 Results

SparseAlign adds markers with small displacements first In Fig. 4.3(a) and (b), we show how SparseAlign localizes markers. At each iteration, markers are added by solving the linearized problem (4.18) on a coarse grid. We show the values of the objective function at each grid location in Fig. 4.3(a). The first marker added is a marker close to the centre of the field of view, where the displacement of markers is smallest. This corresponds with the fact that all deformation parameters are set to zero for the first iteration. After the first iteration, when we start optimizing for the deformation parameters, markers that show larger displacements are added. In Fig. 4.3(b), we show two examples of marker location refinement. The two plots on the left show marker addition and refinement at iteration 3; a new marker, indicated with a red star, is added at a grid location. Local optimization then allows us to move this marker as well as all currently placed markers (blue plus signs) off the grid and closer to the ground truth locations (green crosses). The two plots on the right show another step of marker addition and local optimization at iteration 7. In both cases, local optimization helps to improve the solution close to the region where the new marker is added. We indicate this region with a red rectangle in the plots.

SparseAlign's image-based loss is not convex with respect to deformation parameters In Fig. 4.3(c), we plot the image-based loss in (4.12) as a function of each deformation parameter separately, while holding other parameters and marker locations fixed at their respective ground truth values. For comparison we also plot the marker-based loss in (4.11). Finally, each plot is normalized with a different normalization constant, equal to the maximum value of the loss for that parameter. For each parameter, the marker-based loss is a near-perfect quadratic function with a minimum at the ground truth parameter value. The image-based loss function shares the same minima but differs from the marker-based loss at higher parameter values. In general, the image-based loss function is only convex in a small region around the global minimum. As we move away from the

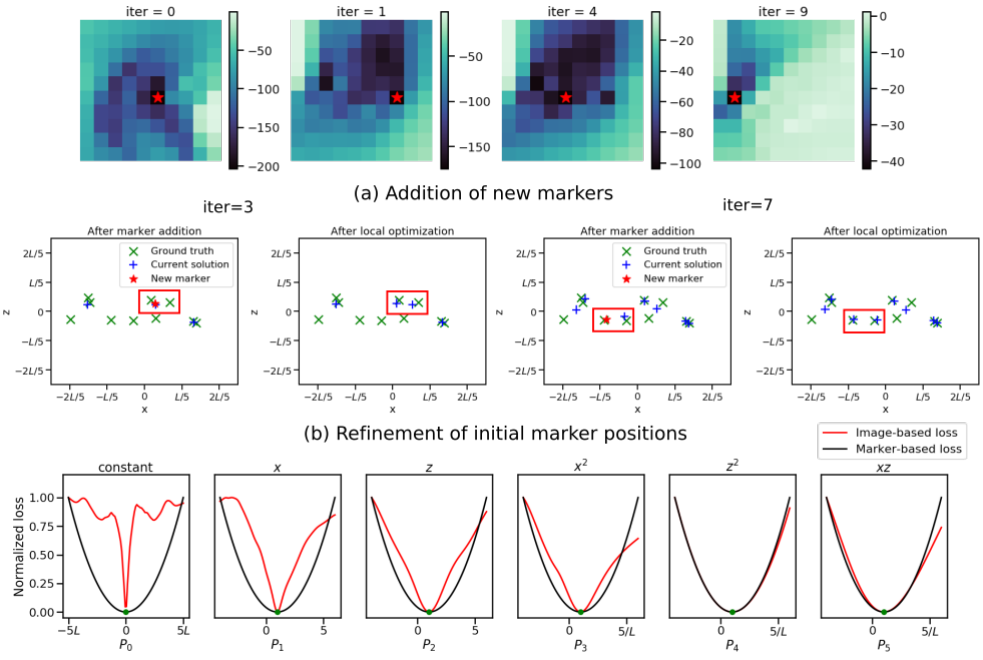


Figure 4.3: Three steps in SparseAlign. (a) Addition of new markers is performed on a coarse grid using the optimization problem (4.18). The grid location with the smallest pixel intensity in the heatmap is chosen as the next candidate location, which is indicated with a red star. (b) Refinement of initial marker locations is performed using L-BFGS-B. The two leftmost plots show one step of marker addition followed by local optimization; the two rightmost plots show another step of marker addition and local optimization. In both cases, after addition of a new marker (red star), local optimization ensures that all current markers (blue plus signs) are brought closer to the ground truth locations (green crosses). We indicate the areas where this improvement is clearest with red rectangles. (c) Sensitivity of the marker-based loss (black line) used in the doming model approach and our image-based loss (red line) to changes in deformation parameter values. For each plot, the loss was normalized independently with respect to its maximum value.

minimum, the loss function increases for each parameter until, at large parameter values, markers move out of the field of view and the loss shows other minima (as in the plot for P_0) or flattens and dips (as in the plots for P_1 through P_3). Gradient-based schemes can thus get caught in local minima if parameter values are very far away from the true minimum at initialization.

SparseAlign estimates deformation parameters with an accuracy comparable to that of the doming model In Fig. 4.4 we illustrate the differences between the doming model optimization used in [93] and our method. We use the simple 2D sample shown in

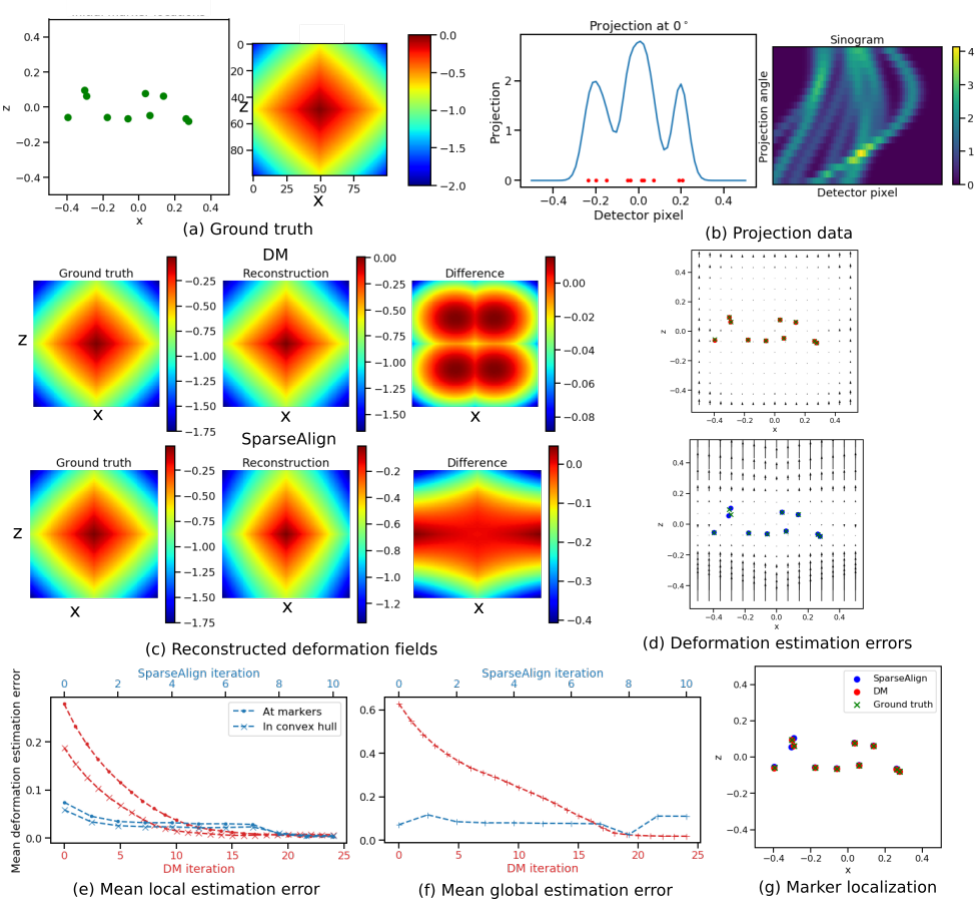


Figure 4.4: Marker localization and deformation estimation using SpaseAlign and the doming model method (DM). (a) Ground truth initial marker locations and deformation field component along the z-axis at $t = t_1$, $D_{1,z}$. (b) Input data for DM are the projected marker locations indicated with red dots. Projection data for SparseAlign at 0° is a 1D profile that is a superposition of Gaussians; we indicate this data in blue. The full sinogram data is a stack of projections taken along tilt angles in [-60°, 60°). (c) Reconstructed deformation fields using DM and SparseAlign. In both cases, errors are largest at the boundaries of the field of view (FoV), where no markers are present. (d) Deformation estimation error (4.27) obtained using DM and SparseAlign. Errors are comparable in the convex hull of markers; errors outside the convex hull are larger when using SparseAlign. (e)-(f) Mean local and global deformation estimation errors (4.28)-(4.29) as a function of DM and SparseAlign iterations. (g) Localized initial marker locations using SparseAlign (blue circles) and DM (red circles) overlaid with the ground truth marker locations (green crosses).

Fig. 4.3 with a quadratic deformation field along the vertical (z) direction.

Input data for the doming model ('DM') optimization are indicated with red dots in Fig. 4.4(b); projection data for SparseAlign is a 1D profile indicated with a blue line. The set of line profiles can be rearranged to give a sinogram for the SparseAlign data.

In Fig. 4.4(c), we show the reconstructed deformation fields obtained using the two methods. In Fig. 4.4(d), we illustrate the vectorial deformation field error (4.27) in both cases. We observe that the error in the convex hull of the markers is comparable using both methods. This is true despite the fact that our method does not need labelled marker locations and minimizes a more complicated image-based loss function. In regions without markers, our method shows larger errors. This is an indication of the greater ill-posedness of our deformation estimation problem (4.20).

In Fig. 4.4(e-f), we compare mean deformation estimation errors (4.29) and (4.28) for both methods at the ground truth marker locations and in the entire FoV. Mean deformation estimation errors at marker locations are comparable for both methods although the global mean error is higher for SparseAlign. The larger global error, however, is not significant because the major contribution comes from boundaries where no sample is present. Marker localization using SparseAlign and DM gives comparable results, as illustrated in Fig. 4.4(g).

Deformation estimation accuracy reduces almost linearly for additive Gaussian noise In Fig. 4.5, we perform a quantitative analysis of the robustness of our method with respect to noise in projection data. The ground truth marker configuration and deformation field are shown in Fig. 4.5(a). We used different noise settings to probe the properties of our method for data corrupted with Gaussian and Poisson noise, and for each noise level we performed 100 independent experiments by randomizing both the initial marker locations as well as using different noise realizations. The mean deformation estimation error plots for Gaussian noise show an almost linear decrease in deformation estimation accuracy for increasing signal-to-noise ratio (SNR, given by the standard deviation of the Gaussian noise). Moreover the spread of the distribution narrows for high SNRs, indicating that there are fewer catastrophic failure cases for deformation estimation.

The dependence of deformation estimation error on noise is more complicated in the case of Poisson noise. As shown in the plots in Fig. 4.5(c), we do not see a linear dependence as in the case of Gaussian noise. The difference in accuracy between deformation estimation results for low and high electron counts is also smaller. This suggests that the mismatch between Poisson noise data and data generated from our forward model is greater than the mismatch in the case of comparable Gaussian noise.

Model mismatch does not affect deformation estimation significantly We used physically plausible TEM simulations to generate data where the forward model of SparseAlign did not match the data generation model.

In these data, the shape function of a gold bead marker is not a Gaussian. In Fig. 4.6(a), we show the profile of a marker in projection data generated using the TEM-simulator package [102] and the profile of a marker using our forward model. We assumed that the size of gold bead markers and the pixel size of projection images are known, so that the

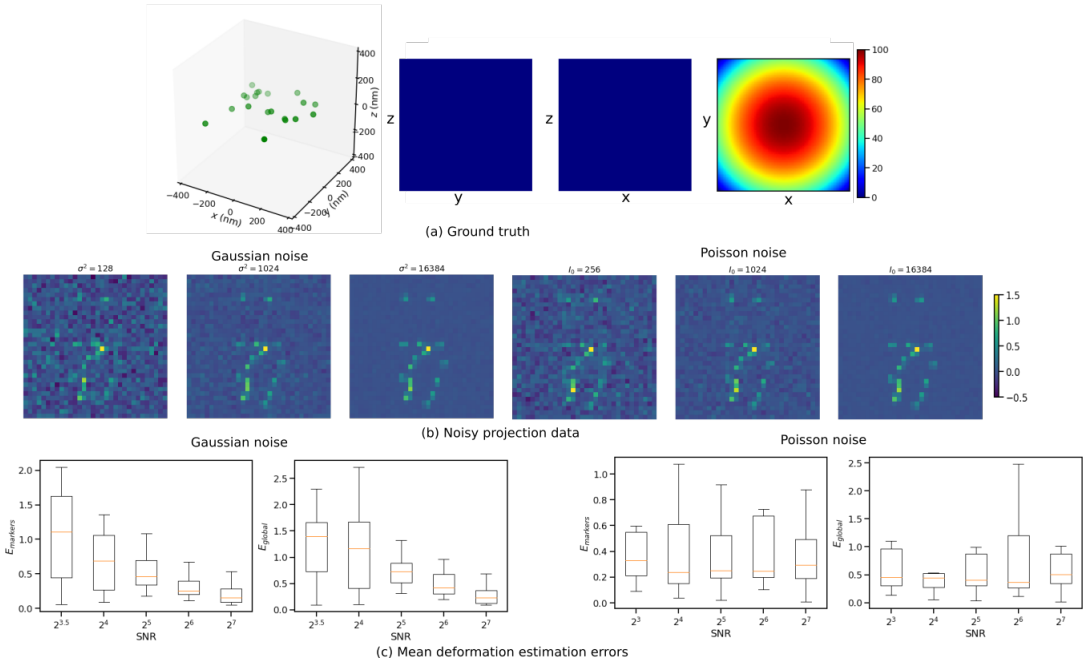


Figure 4.5: Deformation estimation in 3D with Gaussian and Poisson noise-corrupted data. (a) Ground truth configuration of markers (left) and ground truth deformation field in nm (right). (b) Projection image at 0° with different Gaussian noise and Poisson noise settings. The variance of Gaussian noise (σ^2) and the photon flux (I_0) were chosen to simulate comparable Gaussian noise and Poisson noise realizations. (c) Deformation field estimation errors as a function of iteration at markers (E_{markers}) and in the entire field-of-view (E_{global}) for various Gaussian and Poisson noise settings.

width of the Gaussian can be computed.

We used binned simulated data, as detailed in 4.4.3, for these experiments. In Fig. 4.6(b), we show results on marker localization and deformation estimation using noiseless data. The ground truth marker configuration and deformation field are the same as those shown in Fig. 4.5(a). The results we show in Fig. 4.6(b) are those obtained at the final step of a coarse-to-fine scheme, where we solved for marker localizations and deformation parameters at increasing resolutions using downsampling factors $\eta = 1/16, 1/8, 1/4, 1/2$. The final result of such a scheme shows a good qualitative match between reconstructed and ground truth marker locations and deformation fields. We stopped at $\eta = 1/2$ because the effect of model mismatch, which we discuss in the next paragraph, is greatest at high resolutions. Moreover, our current implementation is unable to handle very large data sizes, an area we plan to improve in a future work. Nevertheless, our results indicate a good qualitative match between ground truth and estimated deformation fields, suggesting that the absence of higher-resolution data might not impact deformation estimation for the cases considered.

In Fig. 4.6(c), we show the effect of model mismatch at different resolutions using plots of the difference between our forward projected reconstructed markers and the observed data. We see that the effect of model mismatch is most pronounced at the finest resolutions. This indicates why using a coarse-to-fine scheme, where we obtain initial guesses for marker locations and deformation parameters by solving the problem in a coarse resolution first, leads to reasonable results despite the difference in forward models.

We plot mean deformation estimation errors (4.29) and (4.28) for each iteration in Fig. 4.6(d). Jumps in resolution are indicated with dotted lines. Here we observe that the maximum reduction in deformation estimation error is achieved at the coarsest resolution. The initial guesses obtained are then refined subsequently at each finer resolution. The stopping criterion we used to jump to a higher resolution was to check whether the absolute difference in loss at each new iteration was greater than a pre-set tolerance value (here, 10^{-6}).

Finally, in Fig. 4.6(e), we illustrate the deformation estimation error (4.27) at each resolution. Here we observe that, at the coarsest resolution, the error is already small near the centre of the FoV, where a majority of markers is present. At higher resolutions, the refinement in deformation parameters ensures smaller errors at the boundaries and indicates improvements in the values of estimated parameters.

Marker localization is poor for data with correlated Poisson noise In Fig. 4.7, we show results of our method on data with realistic markers and realistic correlated noise using the ground truth marker configuration and deformation field in Fig. 4.5.

We observe that marker localization for correlated noise-corrupted data is poorer than that for noiseless data (shown in Fig. 4.6). At the end of a coarse-to-fine scheme, two markers are not localized and a few markers with small weights are added to the reconstruction domain. These small weighted markers were removed with a further thresholding step, where markers with weights less than 0.1 were discarded. Improving marker localization might need changes to the forward model used, an aspect that needs further research; however, in our experiments, marker localization did not have a significant effect on deformation estimation accuracy, as seen from the reconstructed deformation field shown in Fig. 4.7(a).

In Fig. 4.7(b), we show plots of mean deformation estimation errors. Note that the same stopping criterion as that used for noiseless data ensured that more iterations were performed at finer resolutions for data with realistic noise.

In Fig. 4.7(c), we plot the deformation estimation error at different resolutions. Comparing these plots with those for noiseless data in Fig. 4.6, we see that the errors at the boundaries are higher for noisy data, which is most clearly observed at the coarse resolutions.

Deformation estimation is limited by the model basis We performed experiments with realistic 3D simulated data where the ground truth deformation field along the z direction contained cubic terms in x and y in addition to the quadratic terms in (4.26).

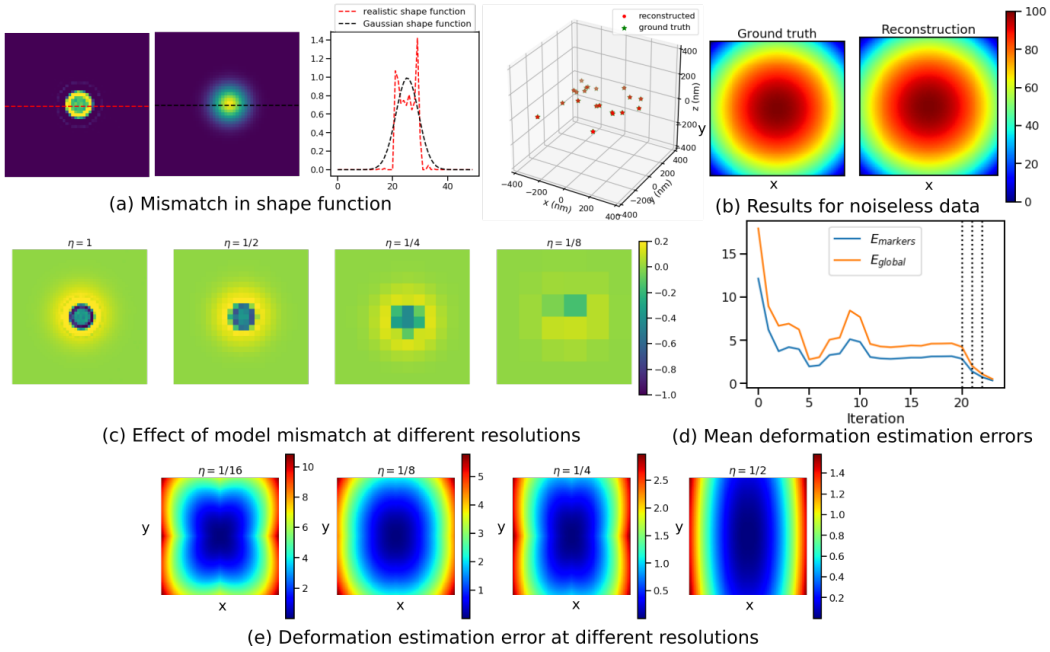


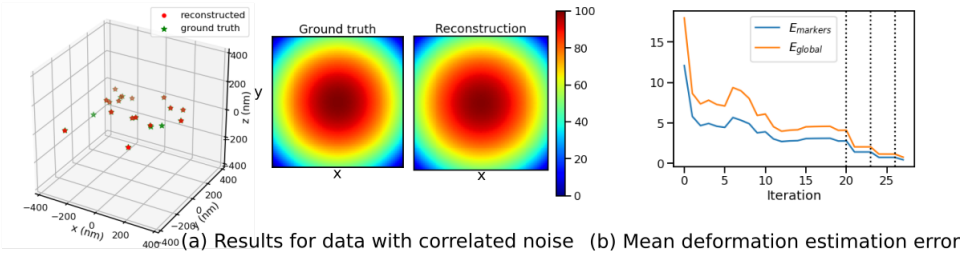
Figure 4.6: (a) Mismatch in shape function. (left) 2D projection of a single marker generated using the TEM simulator. (centre) Projection of a Gaussian marker used in our forward model. (right) Profiles of both shape functions. (b) Marker localization results (left) and deformation estimation results in nm (right) for noiseless realistic data. (c) Difference between forward projected marker locations and observed data (a small region around a single marker is shown). The difference due to model mismatch is largest at the fine resolutions. (d) Mean deformation estimation error at ground truth marker locations and in the entire FoV for different iterations. Resolution changes in the coarse-to-fine scheme are indicated with black dotted lines. (d) Absolute error of estimated deformation field with respect to the ground truth at different values of the downsampling factor η.

The ground truth deformation field used in these experiments was given by:

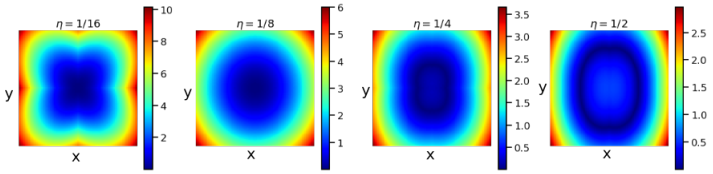
$$D_z(x, y, z, t) = (P_0 + P_1 x^2 + P_2 y^2 + P_3 xy^2 + P_4 x^2 y)t \tag{4.30}$$

with $P_0 = 200$ nm, $P_1 = P_2 = -50$ nm^{-1}, $P_3 = P_4 = 25$ nm^{-2}. Although the ground truth contained cubic terms, we restricted the deformation terms used in our forward model to be quadratic in x and y. We performed experiments for both noiseless data and data corrupted with correlated Poisson noise. For both noiseless and noisy data, our algorithm was able to identify the quadratic terms in the deformation field (Fig. 4.8(a-b)). As there were no cubic terms in the forward model, the reconstructed deformation fields did not contain any cubic components. The effect of this mismatch is greatest at the two corners of the FoV where the contribution of cubic terms was the highest.

When we included cubic terms in the forward model, we found that both marker



(a) Results for data with correlated noise (b) Mean deformation estimation error



(c) Deformation estimation error at different resolutions

Figure 4.7: Results on realistic marker data with correlated noise. (a) Marker localization (left) and deformation estimation (right) results on data corrupted with realistic correlated noise. Deformation values shown in nm. (b) Mean deformation estimation error at ground truth marker locations and in the entire field-of-view for each iteration. Resolution changes in the coarse-to-fine scheme used to solve for marker locations and deformation parameters are indicated with black dotted lines. (c) Absolute deformation estimation error (along z) with respect to the ground truth at different values of the downsampling factor η.

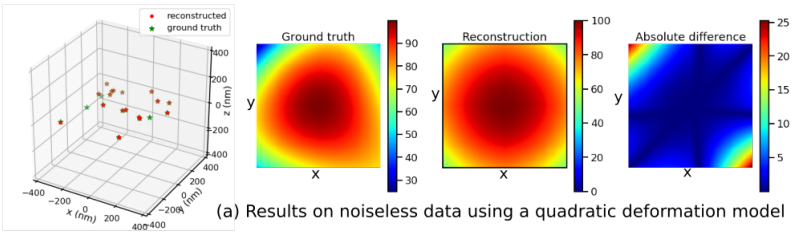
localization and deformation estimation improved as both quadratic and cubic terms were now estimated. The recovered deformation field in Fig. 4.6(c) is much closer to the ground truth. These results indicate that the accuracy of SparseAlign is limited by the basis used for deformation modelling.

SparseAlign locates markers reasonably in experimental data We used an experimental dataset of gold beads embedded in ice to test the applicability of our method to experimental datasets. We used a coarse-to-fine scheme with downsampling factors $\eta = 1/128, 1/64, 1/32, 1/16, 1/8$ to localize gold bead markers and estimate the deformation field. We show an example tilt image in Fig. 4.9(a) and the same image at different downsampling factors in Fig. 4.9(b).

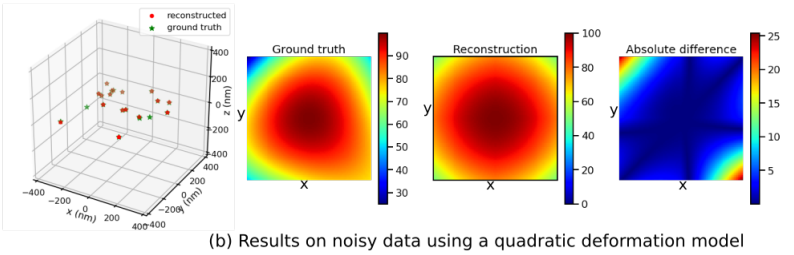
Using a coarse-to-fine scheme we were able to localize several, but not all, markers. In Fig. 4.9(c), we show our marker localization results. We thresholded the localized markers according to their reconstructed weights. Here we show 15 markers with the highest weights. We estimated deformation along the z direction using a quadratic model:

$$D_{t,z}(r, P) = (P_0 + P_1 x + P_2 y + P_3 x^2 + P_4 y^2 + P_5 xy)t \tag{4.31}$$

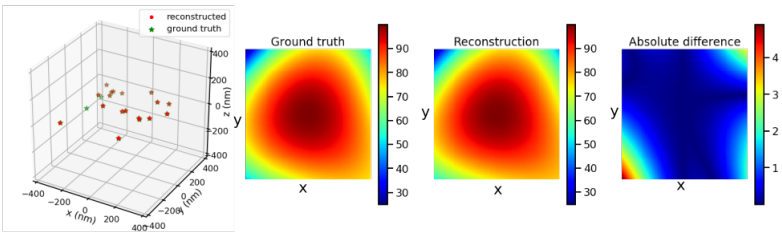
Additionally, we set the x and y components of the deformation field to zero. It is probable that our assumed deformation field was insufficient to model sample deformation in the experimental data.



(a) Results on noiseless data using a quadratic deformation model



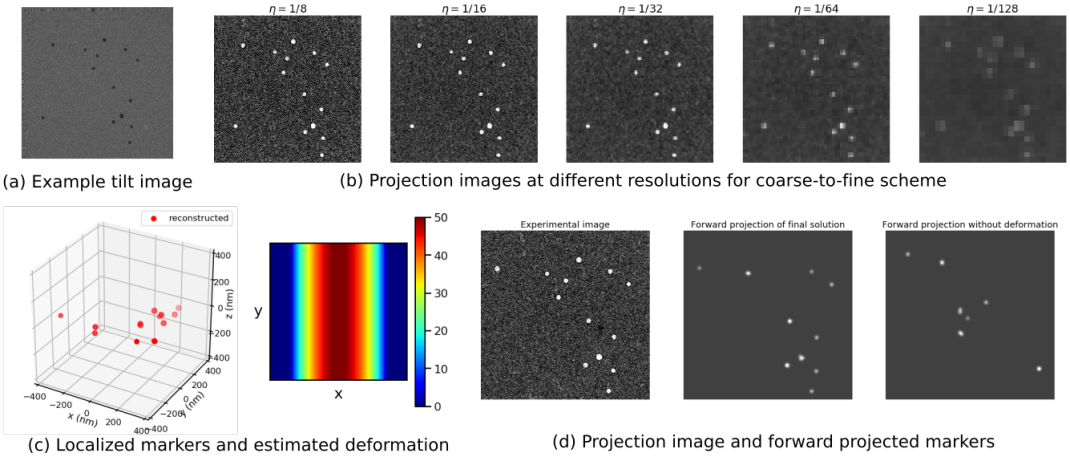
(b) Results on noisy data using a quadratic deformation model



(c) Results on noiseless data using a cubic deformation model

Figure 4.8: Alignment using a mismatched deformation model. Marker localization and deformation estimation for (a) noiseless and (b) noisy data using a quadratic deformation field model and a cubic deformation field as ground truth. (c) Marker localization and deformation field estimation using a cubic deformation field model for noiseless data. All deformation values shown in nm.

Our algorithm predicted a deformation field that is quadratic in x but constant in y, a model that could not be experimentally validated. Plugging the estimated deformation field and marker locations into our forward model, we computed the forward projection shown in Fig. 4.9(d). Comparing this image to the data, we see that not all markers have been localized correctly, but at least one marker was localized in each of location with a cluster of markers. Markers throughout the FoV were localized; this suggests that the deformation estimation routine did not do worse for certain spatial regions. Moreover, mismatch in the shapes of actual markers and the Gaussian used in our forward model did not hinder the localization of most markers. Using localized marker locations and setting deformation to zero leads to projection images that are qualitatively different from the experimental data (Fig. 4.9(d)).



(a) Example tilt image

(b) Projection images at different resolutions for coarse-to-fine scheme

(c) Localized markers and estimated deformation

(d) Projection image and forward projected markers

Figure 4.9: Results on experimental data. (a) A raw projection image in the acquired tilt series. (b) One image from pre-processed data used for deformation estimation and marker localization with downsampling factor $\eta = 1/8, 1/16, 1/32, 1/64, 1/128$. (c) Localized markers (left) and estimated deformation along z (in nm). (d) One experimental projection image downsampled by $\eta = 1/8$ (left), forward projection of localized markers with estimated deformation field (centre) and forward projection of markers with deformation field set to zero (right).

4.6 Conclusion and discussion

Marker-based alignment is a important step for reconstruction improvement in cryoET. We have developed a mathematical approach called SparseAlign for modeling beam-induced local sample motion. In contrast to current approaches, our method does not need labelled marker locations, and directly uses projection data to localize markers and solve for the parameters of a polynomial deformation model. As a consequence, our method is more suited to data with low signal-to-noise ratios where markers cannot be reliably identified. The deformation fields estimated using our method can be used in a subsequent routine to compute a motion-compensated sample reconstruction.

Despite solving a more ill-posed problem for deformation estimation, SparseAlign localizes markers and estimates deformation parameters with an accuracy comparable to that of the doming model approach. Moreover, SparseAlign estimates deformation accurately even when the forward model for markers shows discrepancies with respect to marker projections in observed data.

The image-based loss (4.12) in this chapter can be improved upon by using a more canonical loss as the objective function for marker localization and deformation estimation. Unlike the ℓ^2 loss used in this chapter, the Wasserstein loss measures distances between distributions and has non-zero gradients even when the supports of the ground truth and current solution do not overlap [107].

The application of our approach to experimental data is limited by the deformation

model used. One way to choose the most suitable, sparse basis for deformation modelling is to optimize over a library of basis functions using the data-driven approach in [21].

In this chapter, we have ignored the image contrast of the biological sample while estimating deformation parameters. Ideally, our approach would be the first step in an iterative scheme where we alternate between sample reconstruction and tilt-series alignment, taking both sample and marker contributions into account during deformation estimation.

Chapter 5

Learning cell–cell interactions for vascular network formation

5.1 Introduction

In many real-world applications, it is important to model dynamical equations that best describe the system studied. Dynamical equations may be constructed from first principles; the heat equation in physics is one such example. However, in other scenarios where first-principles methods may be insufficient or lacking, dynamical equations can be learned from data on the time evolution of a system.

A recent approach [108] formulates the discovery of dynamical equations as a sparse inverse problem. In this approach – known as Sparse Identification of Nonlinear Dynamics (SINDy) – the unknown dynamical equation is expressed as a linear combination of library functions, and a sparse combination of these functions able to explain the time evolution of the system is sought.

SINDy has been used to infer the dynamics of simulated and real data for a variety of canonical systems exhibiting nonlinear dynamics. In this chapter we adapt it to study vascular network formation in vertebrates.

Vascular network formation is the generation of a blood vessel network from cells that are initially separate. This process is responsible for the generation of a circulatory system during morphogenesis in vertebrates. The first step of this process is vasculogenesis, where a primary network is created. This network then sprouts and expands, in a process termed angiogenesis. Angiogenesis is also observed in cancer tumours, where it helps tumour maintenance and metastasis.

How endothelial cells organize to form a vascular network is still an open question. It has been proposed [109] that two main contributing factors are: 1) the intrinsic ability of

This chapter is based on:

Learning cell–cell interactions for vascular network formation. *P. S. Ganguly, K. A. E. Keijzer, D. Chen, T.M. Vergroesen, R. M. H. Merks and H. J. Hupkes.* (in preparation)

cells to form networks, and 2) environmental cues. The effects of both these factors have been studied using experimental and simulation studies of network formation. Although there have been extensive experimental investigations of angiogenesis [110]–[112], simulation studies are particularly effective in understanding how the interplay between different biological ingredients leads to network formation. This is because all the parameters of a simulated model can be adjusted and different parameter regimes, which may not be easy to probe in experimental studies, are easily simulated.

Different simulation paradigms have been used in the literature to study vascular network formation: one example is a lattice-free, particle-based approach [113], and another is the lattice-based cellular Potts model (CPM) [109].

The forward problem of network formation consists of modelling the cellular system using a Hamiltonian or a differential equation, followed by obtaining solutions that correspond to the steady state or have the lowest energy. However, it is not always clear which model is most suitable and which parameter regions are the most promising for observing network formation behaviours. Moreover, the correspondence between different simulation models is also not clear. For e.g. it is unknown whether there exist effective equations for stochastic Hamiltonian-based models like CPM.

In this chapter, we adapt the SINDy method to learn effective equations for vascular network formation directly from cell trajectories. In particular, we parametrize the pairwise interaction between cells instead of the vector field in our differential equation. This ensures that the number of parameters we learn remains the same despite an increase in the system size. A related work to ours is [114] where the authors adapt the SINDy framework for stochastic differential equations and parametrize the potential instead of the force vector. However, [114] considers only single particle systems in low dimensions, while we consider systems of many particles. Another related line of research is that that of learning force fields for molecular dynamics [115], where the task is to fit the energy of an atomic configuration obtained by solving the electronic Schroedinger equation. Starting with [116], the approach used is that of decomposing the energy into a sum of terms, one for each atom, and parametrizing each contribution via a neural network. While the idea of sharing parameters across particles is similar to our approach, the task in force fields parametrization is different from ours. Further, our optimization problem is similar to that of SINDy and has the advantage of being a convex optimization, while that of [116] is non-convex.

In this chapter, we focus on proof-of-concept studies, where ground-truth effective equations are available, in order to validate our approach and perform systematic numerical studies of the effect of system size, function library size and noise (Gaussian and stochastic) on the the accuracy of recovery. Our work is an important stepping stone towards applying such an approach to experimental data, where effective equations are unknown, or to other modelling paradigms like CPM, in order to find a correspondence between different simulation strategies. Effective differential equations are amenable to analysis and are much easier to simulate than cell models, thus providing much-needed analytical insight into biological systems.

This chapter is organized as follows. In Section 5.2 we review the SINDy method and provide some background on simulation methods for vascular network formation. In Section 5.3 we detail our method, which adapts the SINDy approach to learn pairwise

interactions. We give details of our numerical experiments and results in Section 5.4, and point to limitations and extensions in Section 5.5.

5.2 Background

5.2.1 SINDy

We consider the following ODE:

$$\dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x}), \tag{5.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ denotes the system state at a certain time and $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^n$ is a vector field that defines the dynamics of the system.

We only have data at discrete time points $\mathcal{T} := \{t_1, \ldots, t_m\}$, which we denote as X:

$$X := \begin{pmatrix} \boldsymbol{x}(t_1) \\ \boldsymbol{x}(t_2) \\ \vdots \\ \boldsymbol{x}(t_m) \end{pmatrix} = \begin{pmatrix} x_1(t_1) & x_2(t_1) & \ldots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \ldots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \ldots & x_n(t_m) \end{pmatrix}. \tag{5.2}$$

From X we can also approximate the time derivatives at $\mathcal{T}$, which we call $\dot{X}$. We shall use central differences

$$\dot{X}_{ij} := \frac{x_i(t_{j+1}) - x_i(t_{j-1})}{t_{j+1} - t_{j-1}}, \tag{5.3}$$

or forward differences

$$\dot{X}_{ij} := \frac{x_i(t_{j+1}) - x_i(t_j)}{t_{j+1} - t_j}, \tag{5.4}$$

depending on the application.

Learning problem The goal of SINDy is to learn the form of the function $\boldsymbol{g}$ from a library of basis functions, given data points X and $\dot{X}$.

First we define the library of K basis functions $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$, such that $\boldsymbol{\theta}_p : \mathbb{R}^n \to \mathbb{R}^n$. The unknown function $\boldsymbol{g}$ is approximated by a linear combination of these basis functions.

We evaluate the functions $\boldsymbol{\theta}_p$ at data points X by writing

$$\Theta(X) := \begin{pmatrix} \boldsymbol{\theta}_1(X) & \boldsymbol{\theta}_2(X) & \ldots & \boldsymbol{\theta}_K(X) \end{pmatrix}, \tag{5.5}$$

where

$$\boldsymbol{\theta}_p(X) = \begin{pmatrix} \boldsymbol{\theta}_p(\boldsymbol{x}(t_1)) \\ \boldsymbol{\theta}_p(\boldsymbol{x}(t_2)) \\ \vdots \\ \boldsymbol{\theta}_p(\boldsymbol{x}(t_m)) \end{pmatrix}.$$

We formulate the recovery of the function $\boldsymbol{g}$ as the following linear least-squares problem:

$$\underset{\boldsymbol{\xi} \in \mathbb{R}^K}{\text{minimize}} \quad \left\| \dot{X} - \Theta(X)\boldsymbol{\xi} \right\|_2^2. \tag{5.6}$$

Inducing sparsity Sparsity of the learnable coefficients is a regularization method used in machine learning to prevent overfitting, namely the fact that the model fits very well the training data but generalizes poorly to unseen data – in our case, to unseen time points. One way to induce sparsity in the coefficients is by solving the following optimization problem that has an ℓ^1 penalty:

$$\underset{\boldsymbol{\xi}\in\mathbb{R}^K}{\text{minimize}} \quad \left\|\dot{X} - \Theta(X)\boldsymbol{\xi}\right\|_2^2 + \alpha\left\|\boldsymbol{\xi}\right\|_1, \tag{5.7}$$

The above problem can be solved using LASSO. For large system sizes, LASSO is known to be computationally expensive and a sequentially thresholded least-squares (STLSQ) algorithm has been used in the literature as an alternative [108].

5.2.2 Particle-based model of vascular network formation

In this section we review the particle-based simulation paradigm that has been used in the literature to study vascular network formation.

This method was originally used to demonstrate that cell elongation and mutual attraction between endothelial cells was indeed sufficient for producing vascular networks [113], a claim that was first made using cellular Potts model (CPM) simulations [109].

In this lattice-free paradigm, each cell is represented with a particle that interacts with other particles in a predefined neighbourhood. The time evolution of the system is modelled with a Langevin equation:

$$\frac{d\boldsymbol{v}_i}{dt} = \frac{1}{m_i}\Big(-\tau\boldsymbol{v}_i + \sum_{j\neq i}\frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}F_{ij} + \boldsymbol{\eta}\Big), \quad \boldsymbol{v}_i = \frac{d\boldsymbol{x}_i}{dt}, \tag{5.8}$$

where τ is the damping constant, F_{ij} is the pairwise interaction between cells and the last term is a stochastic noise term with correlation function

$$\mathbb{E}(\eta_a(t)\eta_b(t')) \propto \delta_{ab}\delta(t-t')\,. \tag{5.9}$$

The pairwise interaction F_{ij} is modelled with a short-range repulsive term and a long-range attractive term:

$$F_{ij} := \lambda_r A_r - \lambda_a A_a, \tag{5.10}$$

where A_r is the area of overlap between the smaller repulsive ellipses and A_a is the overlap between attractive ellipses (see Figure 5.1 (c)), and λ_r and λ_a are constants. The areas of overlap are usually computed in a Cartesian coordinate system and are functions of the locations, eccentricities and orientations of ellipses. We discuss this in more detail in the following section.

Without loss of generality we can set $m_i = 1$, so that the discrete time evolution, using forward differences, is:

$$\boldsymbol{a}_i(t+\Delta t) = -\tau\boldsymbol{v}_i(t) + \sum_{j\neq i}\frac{\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)}{\|\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)\|}F_{ij}(t) + N_v\beta_v(t)\Delta t^{-0.5} \tag{5.11}$$

$$\boldsymbol{v}_i(t+\Delta t) = \boldsymbol{v}_i(t) + \boldsymbol{a}_i(t+\Delta t)\Delta t \tag{5.12}$$

$$\boldsymbol{x}_i(t+\Delta t) = \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t+\Delta t)\Delta t\,. \tag{5.13}$$

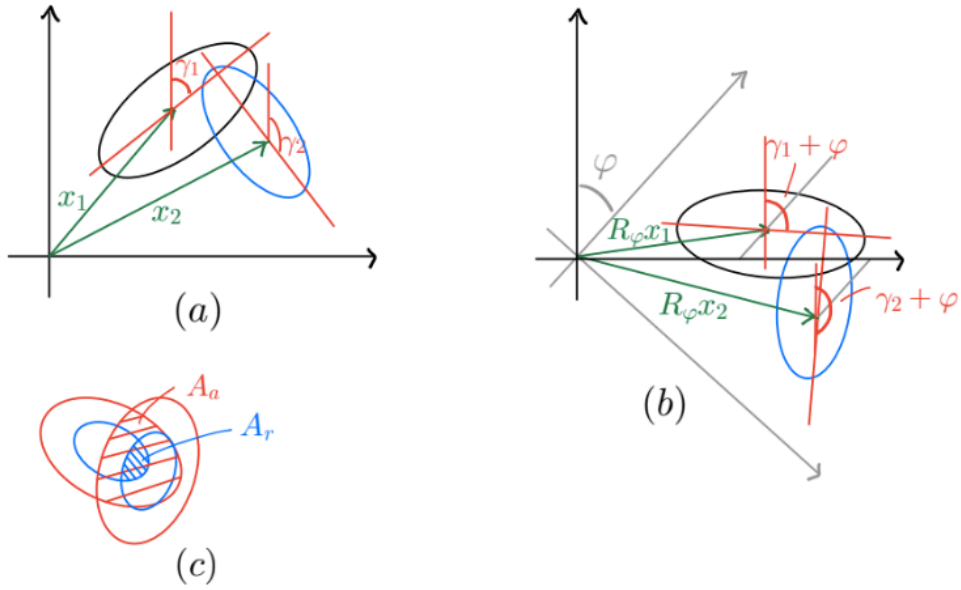


Figure 5.1: (a) The two ellipses model two cells, labelled 1 and 2. x_1, x_2 stand for the coordinates of the centers of the ellipses and γ_1, γ_2 for the angles the axis of the ellipses form with the y axis. (b) A global rotation of φ of the system. (c) Inner area of overlap A_r and outer area of overlap A_a.

Here we introduced the noise amplitude N_v and the Gaussian random vector β_v. In the overdamped regime, where the acceleration is negligible, setting $\tau = 1$, the discrete time evolution of the system reduces to

$$\boldsymbol{x}_i(t + \Delta t) - \boldsymbol{x}_i(t) = \Delta t \sum_{j \neq i} \frac{\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)}{\|\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)\|} F_{ij}(t) + N_v \beta_v(t) \sqrt{\Delta t}\,. \tag{5.14}$$

In addition to vectorial noise modulated by the amplitude N_v, the particle-based simulations make use of angular noise. This corresponds to random changes in the orientation of cells. A change in orientation of cell i is accepted with a turn probability

$$\Pi_i = \min\Big\{1, \exp\Big(\frac{1}{N_a} \sum_{j \neq i} F_{ij} - \sum_{i \neq j} F'_{ij}\Big)\Big\}, \tag{5.15}$$

where N_a is the angular noise amplitude, and F'_{ij} is the interaction between cells i and j if the orientation change is accepted.

In the following section, we show how we apply the method reviewed in Section 5.2.1 to the vascular network formation problem, and how this formulation leads us to discover cell–cell interactions from cell trajectories.

5.3 SINDy for pairwise interaction discovery

We now look at particle and lattice systems whose dynamics is governed by an interaction force between constituents. We first discuss particle systems, which is the primary focus of this chapter, and then comment on how to adapt the framework to lattice systems. In the vascular network formation problem, each of the particles represents a cell with coordinates $\boldsymbol{x}_i \in \mathbb{R}^d$, where d is the dimensionality of the problem. Then the number of variables is $n = d \times n_p$, where n_p denotes the number of particles.

We assume that $d = 2$ and that the dynamics of the system is given by (5.1) with

$$g_i(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) := \sum_{j \in \mathcal{N}_i} \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|} F_{ij}\,, \quad i = 1, \ldots, n_p \tag{5.16}$$

$$F_{ij} := \Phi(\boldsymbol{x}_i, \boldsymbol{x}_j, \gamma_i, \gamma_j), \tag{5.17}$$

where $\mathcal{N}_i$ is the set of particles that particle i interacts with, and γ_i denotes the angle that the i-th ellipse forms with y axis, see Figure 5.1 (a). Each ellipse is determined by $(\boldsymbol{x}_i, \gamma_i)$ and the two axes lengths which are assumed to be fixed for all cells, and therefore omitted from Φ. At this point Φ is a generic function and represents the interaction between the two ellipses. As such it should not change if we translate or rotate both ellipses w.r.t the origin. Translation by a vector $\boldsymbol{a}$ acts as $(\boldsymbol{x}_i, \gamma_i) \mapsto (\boldsymbol{x}_i + \boldsymbol{a}, \gamma_i)$. Rotation by an angle φ acts as $(\boldsymbol{x}_i, \gamma_i) \mapsto (R_\varphi \boldsymbol{x}_i, \gamma_i + \varphi)$, where R_φ is the 2×2 rotation matrix, see Figure 5.1 (b). Imposing translation invariance leads to

$$\Phi(\boldsymbol{x}_i, \boldsymbol{x}_j, \gamma_i, \gamma_j) = \Phi(\boldsymbol{x}_i + \boldsymbol{a}, \boldsymbol{x}_j + \boldsymbol{a}, \gamma_i, \gamma_j) \tag{5.18}$$

whose solution is $\Phi(\boldsymbol{x}_i - \boldsymbol{x}_j, \gamma_i, \gamma_j)$. Imposing rotation invariance leads to

$$\Phi(\boldsymbol{x}_i - \boldsymbol{x}_j, \gamma_i, \gamma_j) = \Phi(R_\varphi(\boldsymbol{x}_i - \boldsymbol{x}_j), \gamma_i + \varphi, \gamma_j + \varphi)\,, \quad \forall \varphi \in [0, 2\pi)\,. \tag{5.19}$$

First, we note that the following is invariant: $\Phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|, \gamma_i - \gamma_j)$. However, this is too restrictive, as it satisfies the more general symmetry $\Phi(\boldsymbol{x}_i - \boldsymbol{x}_j, \gamma_i, \gamma_j) = \Phi(R_\varphi(\boldsymbol{x}_i - \boldsymbol{x}_j), \gamma_i + \varphi', \gamma_j + \varphi')$ even for $\varphi \neq \varphi'$. To simplify the parametrization we follow [113] and add a dependency on the areas of overlap, so that:

$$F_{ij} = \Phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|, \gamma_i - \gamma_j, A_{a,ij}, A_{r,ij})\,, \tag{5.20}$$

where A_a, A_r are as in (5.10). While these areas of overlap can be computed from $\boldsymbol{x}_i - \boldsymbol{x}_j$ and γ_i, γ_j, their expression is complicated and no simple analytical form is known [113]. We also note that this function is periodic in the second argument with period 2π.

We want to recover the function $\Phi : \mathbb{R}_+ \times [0, 2\pi) \times \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}$ given the trajectories of cells over time encoded in the matrix X of size $m \times n$, where n is the number of variables and m is the number of time samples. We shall now adapt the formalism described in Section 5.2.1 to this problem.

As a first step, we write down a set of basis functions $\{f_p(r, \gamma, a, b)\}_{p=1}^K$ to parametrize the unknown function $\Phi(r, \gamma, a, b)$ appearing in equation (5.20). These correspond to the following $\boldsymbol{\theta}_p$ in the formalism of Section 5.2.1:

$$(\boldsymbol{\theta}_p(\boldsymbol{x}))_i = \sum_{j \in \mathcal{N}_i} \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|} f_p(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|, \gamma_i - \gamma_j, A_{a,ij}, A_{r,ij})\,. \tag{5.21}$$

Then we can plug in these values for $\boldsymbol{\theta}_p$ in equation (5.6) and solve the least square problem. The solution $\boldsymbol{\xi}$ will then describe the function Φ as:

$$\Phi(r, \gamma, a, b) = \sum_{p=1}^{K} \xi_p f_p(r, \gamma, a, b)\,. \tag{5.22}$$

If we take $\mathcal{N}_i$ in (5.16) to be the set of $n_p - 1$ points $j \neq i$, this implies that all particles interact with each other. To restrict particle interaction to within a neighbourhood, we can define a critical radius of interaction r_c, such that $f_p(r, \gamma, a, b) = 0$ if $r > r_c \ \forall p$. In the experiments shown later in the chapter, we do not learn the dynamics of the cell orientation parameters γ; instead, we treat them as known inputs.

5.4 Numerical experiments and results

5.4.1 1D lattice system

We can use the formulation of (5.16) for lattice systems as well by assigning the indices i to points on the lattice. For example in 1D, $i = 1, \ldots, n$ are the points on a line. $\boldsymbol{x}$ is then a field with values $\boldsymbol{x}_i$ at site i. In the case of lattice systems, we take the range of values $\mathcal{N}_i$ to be the neighbours on the grid. For example in 1D, $\mathcal{N}_i = \{i-1, i+1\}$ describes nearest-neighbour interactions.

In this section, we first describe our experiments on recovering the pairwise interaction between harmonic oscillators on a 1D lattice with nearest-neighbour interactions. The displacement of the ith particle is given by $x_i(t)$. We generated particle trajectories by evolving the system in the overdamped regime:

$$\dot{x}_i(t) = \sum_{j=i-1,i+1} \frac{x_i - x_j}{r_{ij}} F_{ij}, \quad F_{ij} := -k(r_{ij} - \rho), \tag{5.23}$$

where $r_{ij} = |x_i - x_j|$ and ρ is an offset. The initial configuration of oscillators and F_{ij} are shown in Figure 5.2, where we take $k = 2.0$, $\rho = 1.0$.

We integrated the dynamical equation numerically to obtain a matrix X for a discrete set of time points $\mathcal{T} = \{t_1, \ldots, t_m\}$. The matrix of time derivatives $\dot{X}$ was obtained using equation (5.3).

As the pairwise interaction between particles is a function of r_{ij}, we chose library functions that were polynomials of r_{ij}:

$$f_p(r) = r^p, \quad p \in \{0, 1, \ldots, K\} \tag{5.24}$$

and used these to solve the LASSO problem (5.7).

As a first experiment, we show how to determine the regularization parameter α in equation (5.7). For fixed K, n and m, we use LASSO to infer the parameters $\boldsymbol{\xi}$ for different values of α (see Figure 5.2). We choose the optimum α to be the one with the minimum number of non-zero terms in $\boldsymbol{\xi}$ for which the coefficient of determination

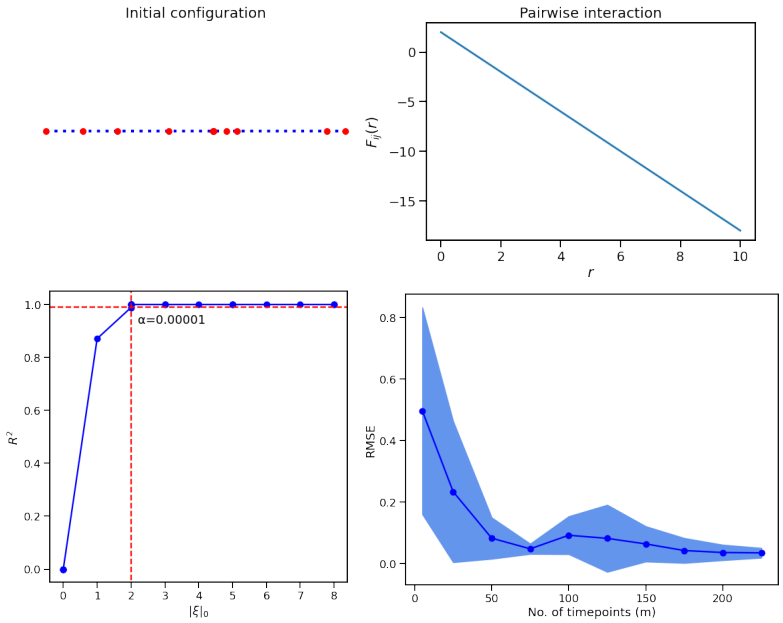


Figure 5.2: (*top left*) Initial configuration of the 1D lattice system with oscillators shown in red and connecting springs shown in blue; (*top right*) ground-truth pairwise interaction F_{ij} as a function of separation distance r. (*bottom left*) Plot of R^2 coefficient with respect to the number of non-zero parameters ξ for different values of the regularization parameter α at fixed $K = 10$, $m = 3$, $n = 1024$. The optimum regularization parameter, $\alpha = 10^{-5}$, is chosen such that $R^2 \geq 0.99$ for the least number of non-zero parameters. (*bottom right*) Plot of RMSE with respect to the number of timepoints m for noisy measurement data with $\sigma = 0.1$. The blue dots show mean values and the ribbons show standard deviations computed over 10 randomised noise seeds.

satisfies $R^2 \geq 0.99$, where:

$$R^2 = 1 - \frac{\|\dot{X} - \Theta(X)\boldsymbol{\xi}\|_2^2}{\|\dot{X} - \frac{1}{mn}\sum_{ij}\dot{X}_{ij}\|_2^2}. \tag{5.25}$$

In the absence of measurement noise, we can infer the correct coefficients for arbitrary K and n with as little as $m = 3$ timepoints (when time derivatives are computed using the central difference scheme (5.3)) and $\Delta t = 0.001\,\frac{1}{k}$.

As the next experiment, we investigate the effect of measurement noise on inference accuracy. In the most general setting, measurement noise affects both X and $\dot{X}$, the latter being numerical derivatives of the former. Applying SINDy to such data typically leads to large errors in the inferred parameters [108]. Instead as in [108], we choose to restrict measurement noise to observed values of $\dot{X}$. This translates to the forward problem:

$$\dot{X} = \Theta(X)\boldsymbol{\xi} + \eta\,, \tag{5.26}$$

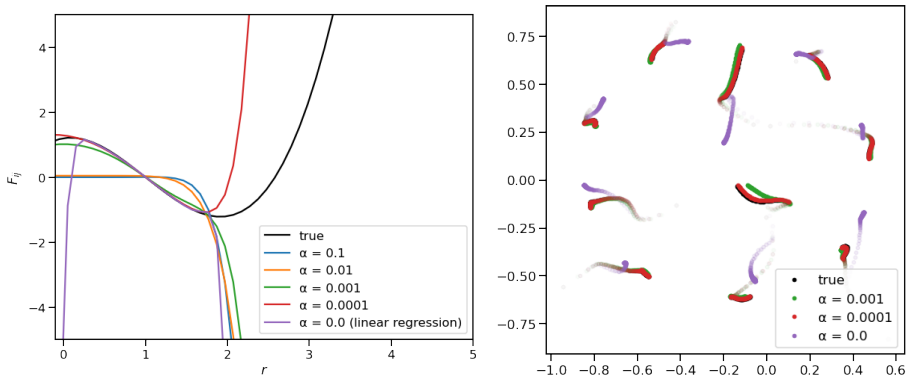


Figure 5.3: (*left*) Predicted interactions for various values of the regularization parameter α; in all inference experiments a library with $K = 11$ polynomial terms in r was used. (*right*) Ground truth trajectories (in black) overlaid with predicted trajectories for $n = 100$, $m = 100$; more transparent points are earlier in time.

where $\eta \sim \mathcal{N}(0, \sigma \mathbf{1})$.

We inferred parameters $\boldsymbol{\xi}$ for noisy measurement data using $\sigma = 0.1$ times the range of $\dot{X}$. We computed inference accuracy using the root mean squared error (RMSE) of the inferred parameters $\boldsymbol{\xi}_{\text{inf}}$ with respect to the ground truth $\boldsymbol{\xi}_{\text{gt}}$:

$$\text{RMSE} = \|\boldsymbol{\xi}_{\text{inf}} - \boldsymbol{\xi}_{\text{gt}}\|_2 \,. \tag{5.27}$$

In Figure 5.2, we observe that the RMSE is high for a small number of timepoints m and declines as m is increased.

5.4.2 2D particle system

Next we turn to a particle system in 2D, where each particle interacts with all others. This latter system brings us closer to the vascular network system, where 2D cell–cell interactions are at play.

For this system, we pick a cubic function to describe the ground-truth interaction between cells:

$$\dot{\boldsymbol{x}}(t) = \sum_{j \neq i} \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{r_{ij}} F_{ij}, \quad F_{ij} := k_1(r_{ij} - \rho)^3 - k_2(r_{ij} - \rho), \tag{5.28}$$

with $k_1 = 0.8$, $k_2 = 2.0$, $\rho = 1.0$. The inter-particle separation r_{ij} is now given by the Euclidean distance between particles i and j: $r_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$.

We performed simulations with $n = 10$ particles by integrating the above equation for $m = 100$ time points with time interval equal to $k_2/10$.

We inferred pairwise interactions between the particles by generating a library of polynomial terms (5.24) with $K = 11$. Using LASSO with regularization parameter α, we get different solutions for the inferred interaction in this case (shown in Figure 5.3).

High values of α lead to pairwise interactions where the cubic nature of the ground-truth function is not captured at all. Reducing α activates more and more terms in the function library. For $\alpha = 0.0$, where we effectively solve the linear least-squares problem (5.6), we get a poorer estimation for the interaction. The predicted trajectories overlaid on the ground-truth trajectories for $\alpha = 0.0001$ show a close match. This indicates that the part of the interaction that is not matched in this setting does not play a role in the data. This is not surprising as the part of the interaction that is not matched corresponds to the asymptotically increasing part of the cubic function in (5.28), and particles that experience this large force show exploding trajectories (x approaching infinity). Such particles were not included in the data in our simulations as such exploding trajectories are unphysical and unlikely to occur in a real experiment.

5.4.3 Particle-based simulations of vascular network formation

Finally we apply our method of interaction learning to simulated data of vascular network formation. For data generation in this part we used the particle-based simulation method described in Section 5.2.2, which has an open-source implementation in C++ [113].

We performed simulations with $n = 100$ elongated cells with fixed orientations. The ground-truth interaction between cells was given by equation (5.10) with $\lambda_r = 0.02$ and $\lambda_a = 0.0006$. We evolved the system using the discretized Langevin equation (5.11) for $m = 100$ time steps with time interval $\Delta t = 1.0$. The damping factor τ was set to 1.0 to simulate overdamped dynamics. To simulate vectorial stochastic noise in the locations of cells, we performed a series of simulations by modulating the noise amplitude N_v in equation (5.11). A network generated with the particle-based simulation method using noise amplitude $N_v = 0.0$ is shown in the top row of Figure 5.4.

Using our method, we then inferred cell–cell interaction terms from a library of $K = 15$ terms. The library terms used were polynomial functions of the areas of overlap A_r and A_a as well as those of the separation distance r. We also used two trigonometric terms for the relative orientation between cells γ. The full library used was:

$$\begin{aligned} f_1 = 1.0,\, f_2 = A_r,\, f_3 = A_a,\, f_4 = A_r^2,\, f_5 = A_a^2, \\ f_6 = A_r^3,\, f_7 = A_a^3,\, f_8 = A_r^4,\, f_9 = A_a^4,\, f_{10} = \cos(\gamma), \\ f_{11} = \sin(\gamma),\, f_{12} = r^1,\, f_{13} = r^2,\, f_{14} = r^3,\, f_{15} = r^4\,. \end{aligned}$$

In Figure 5.4, we plot inferred networks for noise amplitude $N_v = 0.0$. These networks were obtained by using the coefficients of the inferred terms as input to the particle-based simulations and integrating forward in time. The global structure of the inferred networks is qualitatively similar to that of the true networks. To quantify the similarity between networks at a given timepoint, we defined the deviation of the inferred network from the true network as

$$\epsilon = \frac{1}{n_p} \sum_{i=1}^{n_p} \|\boldsymbol{x}_i^{\text{inf}} - \boldsymbol{x}_i^{\text{gt}}\|_2, \tag{5.29}$$

where $\boldsymbol{x}_i^{\text{inf}}$ denotes the position of cell i in the inferred network and $\boldsymbol{x}_i^{\text{gt}}$ denotes its position in the ground truth.

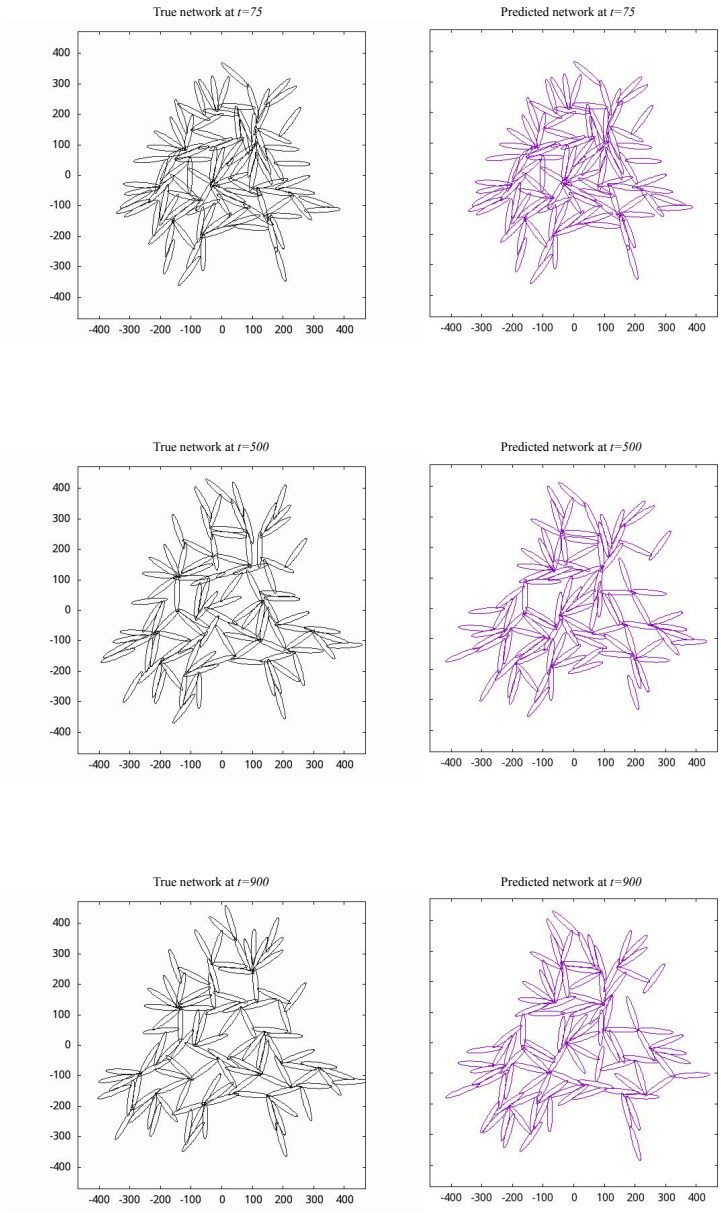


Figure 5.4: True and inferred networks of a particle-based simulation of angiogenesis using 100 elongated cells. Pairwise interactions were inferred for $N_v = 0.0$ using $m = 100$, $n = 100$; inferred networks were obtained by using the inferred interactions as input to an open-source C++ particle-based simulation code [113].

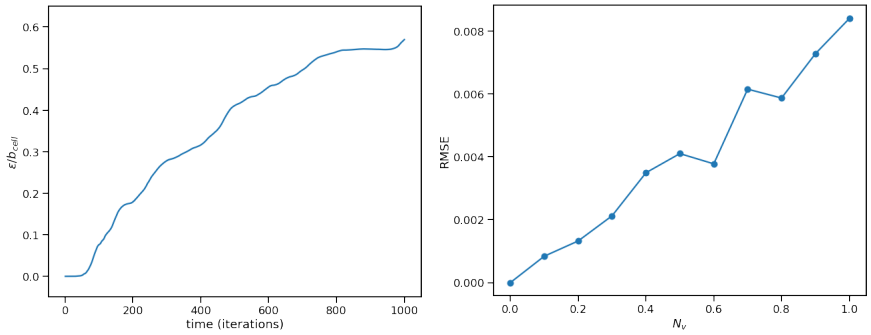


Figure 5.5: (*left*) Plot showing deviation of inferred network from the true network as a function of iterations for $N_v = 0.0$; (*right*) RMSE with respect to the stochastic noise amplitude N_v. For all inference experiments, we used a library with $K = 15$ terms, $m = 100$ and $n = 100$.

In Figure 5.5, we plot this deviation normalised by the major diameter of cells (b_{cell}, which is taken to be constant) as a function of iterations. The deviation is almost zero for iteration numbers lesser than 50 and is less than 10% of the cell diameter for the first 100 iterations. This indicates a good match with the data used for inference, given that we used the first 100 iterations for inferring pairwise interactions. The inferred network deviates from the true network to a greater extent for the next iterations; this is expected as small deviations in the earlier time points accumulate to larger differences later on. Qualitatively, we observe greater differences between the networks at $t = 500$ than between those at $t = 75$ in Figure 5.4. Interestingly, for longer iteration times (greater than 800), the deviation flattens out; this could be the result of the two networks (true and inferred) reaching separate steady states. Note, however, that the largest deviation in networks is still quite small – lesser than one cell diameter. For comparison, the field-of-view in the plots in Fig 5.4 is slightly greater than $8\, b_{\text{cell}}$.

In Figure 5.5, we also plot the RMSE (5.27) as a function of the noise amplitude N_v. For the noiseless simulation ($N_v = 0.0$), we were able to estimate the coefficients with high accuracy. The RMSE increased almost linearly for increasing amplitudes N_v of stochastic noise. Qualitatively, this observation is similar to one reported in [114], where the authors study homogeneous diffusion in the presence of thermal noise and report an increase in the percentage of function libraries that result in the correct solution with decreasing noise. In our experiments, adding stochastic noise did not have a large effect on inference accuracy, as evidenced by an increase in RMSE of less than 1%. This suggests that our deterministic method performs reasonably for the amounts of stochastic noise in such simulations.

5.5 Discussion and conclusions

In this chapter we discussed a method to learn pairwise interactions between cells from their trajectories. We adapted an existing equation learning method, SINDy, to our problem and demonstrated our approach on simulated lattice and particle data. On 1D lattice data we demonstrated the effect of Gaussian measurement noise on inference accuracy and presented a way to choose the optimum sparsity level by tuning the regularization parameter α. On 2D particle data, we further demonstrated the effect of the parameter α on the learned interaction and showed that parts of the interaction that are not matched correspond to specific regions that are not sampled in the data. On particle-based simulations of angiogenesis, we presented results on learning the interaction between elongated cells, and showed how the accuracy of inference degrades with stochastic noise. In the following, we briefly discuss how to apply our method to cellular Potts model (CPM) simulations.

The CPM is another simulation paradigm that has been used to elucidate mechanisms of vascular network formation. In particular, it was used to show that cell elongation was crucial to network generation [109], a claim that is supported by experimental observations. The CPM uses lattice spins to simulate biological cells. Each cell is a patch of identical spins, while the intercellular spaces are modelled by patches of the opposite spin. The interaction between neighbouring spins is used to generate an effective Hamiltonian, whose ground state is reached by performing Monte Carlo steps. To learn a CPM, we would use a library of Hamiltonian terms and coefficients. The observed data, analogous to the data obtained from particle-based simulations, would be the centres of mass and orientations of whole cells, which in the case of CPM correspond to patches of spins or Potts domains.

Applying our method to CPM is a stepping stone to inferring effective equations from experimental wet-lab data. This would enable a complementary approach to angiogenesis simulations, and pave the way to directly learning interactions that lead to network formation.

Chapter 6

Conclusion

In this thesis we investigated ways to leverage sparsity in the design of practical algorithms for various inverse problems. The inverse problems we focused on arose in quite different application areas, with each being a topic of intensive research in its own right. The methods presented in this thesis, while being tailored to each application, also share some overarching similarities in design and implementation. This, we believe, indicates the importance of developing mathematical tools that can be applied to more than one practical problem.

In this concluding chapter, we summarize the contributions of this thesis and point to some future research directions.

In Chapter 2 we presented a filter-optimization method to improve reproducibility of reconstructions for synchrotron tomography. Our method used sparsity in the design of optimal filters. By using the fact that many standard real-space filters taper off to zero at the detector boundaries, we were able to reduce the number of filter coefficients that need to be computed. These sparse-basis filters, when optimized to various implementations of direct reconstruction algorithms, were shown to result in reconstructions with fewer differences than those that were obtained with standard filters. Our work in this chapter is a stepping stone towards a more reproducible synchrotron pipeline, which will require both hardware and software modifications.

In Chapter 3, we demonstrated the use of sparsity in reconstructing atomic defects. We built on existing ideas of grid-free sparse optimization to propose a more canonical discretization of the atomic-resolution reconstruction problem. This discretization did away with the need for reconstructing on a voxel grid. Instead, we modelled atomic configurations as sparse measures, allowing for continuous deviations of atomic locations. We showed how, coupled with physical prior knowledge on the potential energy of atomic configurations, our grid-free method is able to reconstruct common lattice defects with very few projections. We demonstrated the power of our approach in proof-of-concept numerical studies, and proposed further modifications that would make our algorithm applicable to real data.

We extended our grid-free sparse optimization method to investigate marker-based alignment for cryoET in Chapter 4. Here we modelled marker configurations as deforming

measures and used similar ideas to those first developed in Chapter 3 to solve for marker locations and deformations. We applied our approach to synthetic data as well as real data of markers embedded in ice. Our numerical experiments showed that this approach was able to localize markers without the need for the user to label markers in projection data, a cumbersome and error-prone pre-processing task that is needed for existing methods. Our approach is flexible and allows for different models of sample deformation and marker shapes.

In Chapter 5, we used sparsity to recover pairwise interactions that lead to network formation in vertebrates. Our work builds on existing literature on nonlinear equation learning, where a sparse combination of library terms is learnt for time-series data. We used particle-based simulations of angiogenesis to generate time-series data of interacting cells, and were able to recover the relevant interaction terms that led to the formation of networks. Our work is a stepping stone to learning interaction terms in settings where these are not evident, such as other simulation paradigms like the cellular Potts model and experimental data of vascular network formation from endothelial cells.

The work in this thesis shows how sparsity can be used both implicitly and explicitly. Examples of the former include choosing sparse filter basis functions and making certain algorithmic choices, such as adding only one atom to the current solution at each iteration. An explicit way to include sparsity while solving inverse problems is to include an ℓ^1 regularization term in the objective, for example when inferring pairwise interactions between cells.

One promising paradigm developed in recent years is to parametrize the regularizer with a neural network. In the case of tomographic imaging, such learned regularizers [117], [118] have been shown to outperform methods with hand-crafted regularization terms. This is part of a wider interest in the application of data-driven approaches to inverse problems [56], [119]. Although learned approaches might be superior to several classical approaches that enforce sparsity explicitly, ideas of sparsity are also important for improving the efficiency and robustness of deep-learning methods. One example is the use of sparsity to reduce the complexity of deep neural networks by pruning network weights. This has been shown to result in more generalizable networks that also use less resources to train [120]. Another example is using sparsity implicitly in choosing an appropriate discretization for the studied system. This approach is similar to our work on reconstructing nanocrystal defects, and has recently been used to study the problem of atomic-resolution cryo-electron microscopy of proteins [121]. Such examples indicate the continued relevance of sparsity-based approaches in designing efficient algorithms for inverse problems.

Bibliography

[1] W. A. Kalender, "X-ray computed tomography," *Physics in Medicine & Biology*, vol. 51, no. 13, R29, 2006.

[2] P. J. Withers, C. Bouman, S. Carmignato, *et al.*, "X-ray computed tomography," *Nature Reviews Methods Primers*, vol. 1, no. 1, pp. 1–21, 2021.

[3] M. Morigi, F. Casali, M. Bettuzzi, R. Brancaccio, and V. d'Errico, "Application of X-ray computed tomography to cultural heritage diagnostics," *Applied Physics A*, vol. 100, no. 3, pp. 653–661, 2010.

[4] M. Weyland and P. A. Midgley, "Electron tomography," *Materials Today*, vol. 7, no. 12, pp. 32–40, 2004.

[5] M. Turk and W. Baumeister, "The promise and the challenges of cryo-electron tomography," *FEBS letters*, vol. 594, no. 20, pp. 3243–3261, 2020.

[6] M. Bertero, P. Boccacci, and C. De Mol, *Introduction to inverse problems in imaging*. CRC press, 2021.

[7] W. van Aarle, W. J. Palenstijn, J. De Beenhouwer, *et al.*, "The ASTRA toolbox: A platform for advanced algorithm development in electron tomography," *Ultramicroscopy*, vol. 157, pp. 35–47, 2015.

[8] J. Adler, H. Kohr, and O. Oktem, "Operator discretization library (ODL)," *Software available from https://github. com/odlgroup/odl*, 2017.

[9] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[10] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[11] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010.

[12] N. Boyd, G. Schiebinger, and B. Recht, "The alternating descent conditional gradient method for sparse inverse problems," *SIAM Journal on Optimization*, vol. 27, no. 2, pp. 616–639, 2017.

[13] F. Marone and M. Stampanoni, "Regridding reconstruction algorithm for real-time tomographic imaging," *Journal of synchrotron radiation*, vol. 19, no. 6, pp. 1029–1037, 2012.

[14] D. M. Pelt and K. J. Batenburg, "Improving filtered backprojection reconstruction by data-dependent filtering," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4750–4762, 2014.

[15] M. J. Lagerwerf, W. J. Palenstijn, H. Kohr, and K. J. Batenburg, "Automated FDK-filter selection for cone-beam CT in research environments," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 739–748, 2020.

[16] K. J. Batenburg, "A network flow algorithm for reconstructing binary images from discrete X-rays," *Journal of Mathematical Imaging and Vision*, vol. 27, no. 2, pp. 175–191, 2006.

[17] K. J. Batenburg and J. Sijbers, "Generic iterative subset algorithms for discrete tomography," *Discrete Applied Mathematics*, vol. 157, no. 3, pp. 438–451, 2009.

[18] B. Goris, J. De Beenhouwer, A. De Backer, *et al.*, "Measuring lattice strain in three dimensions through electron microscopy," *Nano letters*, vol. 15, no. 10, pp. 6996–7001, 2015.

[19] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval research logistics quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

[20] T. Bendory, A. Bartesaghi, and A. Singer, "Single-particle cryo-electron microscopy: Mathematical theory, computational challenges, and opportunities," *IEEE signal processing magazine*, vol. 37, no. 2, pp. 58–76, 2020.

[21] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the national academy of sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[22] L. Boninsegna, F. Nüske, and C. Clementi, "Sparse learning of stochastic dynamical equations," *The Journal of chemical physics*, vol. 148, no. 24, p. 241 723, 2018.

[23] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Inferring biological networks by sparse identification of nonlinear dynamics," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 2, no. 1, pp. 52–63, 2016.

[24] F. Fusseis, X. Xiao, C. Schrank, and F. De Carlo, "A brief guide to synchrotron radiation-based microtomography in (structural) geology and rock mechanics," *Journal of Structural Geology*, vol. 65, pp. 1–16, 2014.

[25] Y. Luo, S. Wu, Y. Hu, and Y. Fu, "Cracking evolution behaviors of lightweight materials based on in situ synchrotron X-ray tomography: A review," *Frontiers of Mechanical Engineering*, vol. 13, no. 4, pp. 461–481, 2018.

[26] P. A. Midgley and R. E. Dunin-Borkowski, "Electron tomography and holography in materials science," *Nature materials*, vol. 8, no. 4, pp. 271–280, 2009.

[27] G. D. Rubin, "Computed tomography: Revolutionizing the practice of medicine for 40 years," *Radiology*, vol. 273, no. 2S, S45–S74, 2014.

[28] T. M. Buzug, "Computed tomography," in *Springer Handbook of Medical Technology*, Springer, 2011, pp. 311–342.

[29] A. C. Kak, M. Slaney, and G. Wang, "Principles of computerized tomographic imaging," *Medical Physics*, vol. 29, no. 1, pp. 107–107, 2002.

[30] A. Thompson, J. Llacer, L. C. Finman, *et al.*, "Computed tomography using synchrotron radiation," *Nuclear Instruments and Methods in Physics Research*, vol. 222, no. 1-2, pp. 319–323, 1984.

[31] F. De Carlo, X. Xiao, and B. Tieman, "X-ray tomography system, automation, and remote access at beamline 2-BM of the Advanced Photon Source," in *Developments in X-ray Tomography V*, International Society for Optics and Photonics, vol. 6318, 2006, 63180K.

[32] S. R. Stock, *Microcomputed tomography: methodology and applications*. CRC press, 2019.

[33] X. Yang, F. De Carlo, C. Phatak, and D. Gürsoy, "A convolutional neural network approach to calibrating the rotation axis for X-ray computed tomography," *Journal of Synchrotron Radiation*, vol. 24, no. 2, pp. 469–475, 2017.

[34] C. Hintermüller, F. Marone, A. Isenegger, and M. Stampanoni, "Image processing pipeline for synchrotron-radiation-based tomographic microscopy," *Journal of synchrotron radiation*, vol. 17, no. 4, pp. 550–559, 2010.

[35] D. Paganin, S. C. Mayo, T. E. Gureyev, P. R. Miller, and S. W. Wilkins, "Simultaneous phase and amplitude extraction from a single defocused image of a homogeneous object," *Journal of microscopy*, vol. 206, no. 1, pp. 33–40, 2002.

[36] L. Massimi, F. Brun, M. Fratini, I. Bukreeva, and A. Cedola, "An improved ring removal procedure for in-line X-ray phase contrast tomography," *Physics in Medicine & Biology*, vol. 63, no. 4, p. 045 007, 2018.

[37] D. Gürsoy, F. De Carlo, X. Xiao, and C. Jacobsen, "TomoPy: A framework for the analysis of synchrotron tomographic data," *Journal of synchrotron radiation*, vol. 21, no. 5, pp. 1188–1193, 2014.

[38] D. M. Pelt, D. Gürsoy, W. J. Palenstijn, J. Sijbers, F. De Carlo, and K. J. Batenburg, "Integration of TomoPy and the ASTRA toolbox for advanced processing and reconstruction of tomographic synchrotron data," *Journal of synchrotron radiation*, vol. 23, no. 3, pp. 842–849, 2016.

[39] M. Salomé, F. Peyrin, P. Cloetens, *et al.*, "A synchrotron radiation microtomography system for the analysis of trabecular bone samples," *Medical Physics*, vol. 26, no. 10, pp. 2194–2204, 1999.

[40] M. Bührer, H. Xu, J. Eller, J. Sijbers, M. Stampanoni, and F. Marone, "Unveiling water dynamics in fuel cells from time-resolved tomographic microscopy data," *Scientific Reports*, vol. 10, no. 1, pp. 1–15, 2020.

[41] W. Kanitpanyacharoen, D. Y. Parkinson, F. De Carlo, *et al.*, "A comparative study of X-ray tomographic microscopy on shales at different synchrotron facilities: ALS, APS and SLS," *Journal of synchrotron radiation*, vol. 20, no. 1, pp. 172–180, 2013.

[42] B. A. Dowd, G. H. Campbell, R. B. Marr, *et al.*, "Developments in synchrotron X-ray computed microtomography at the National Synchrotron Light Source," in *Developments in X-ray Tomography II*, International Society for Optics and Photonics, vol. 3772, 1999, pp. 224–236.

[43] W. J. Palenstijn, K. J. Batenburg, and J. Sijbers, "The ASTRA tomography toolbox," in *13th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE*, vol. 2013, 2013, pp. 1139–1145.

[44] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, *et al.*, "Scikit-image: Image processing in Python," *PeerJ*, vol. 2, e453, 2014.

[45] F. Natterer, *The mathematics of computerized tomography*. SIAM, 2001.

[46] K. J. Batenburg, P. C. Hansen, and J. S. Jorgensen, "Discretization models and the system matrix," in *Scientific Computing for Computed Tomography*, P. C. Hansen, J. S. Jorgensen, and W. R. B. Lionheart, Eds., in press, 2021, ch. 8.

[47] F. Xu and K. Mueller, "A comparative study of popular interpolation and integration methods for use in computed tomography," in *3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro, 2006.*, IEEE, 2006, pp. 1252–1255.

[48] F. Arcadu, M. Stampanoni, and F. Marone, "Fast gridding projectors for analytical and iterative tomographic reconstruction of differential phase contrast data," *Optics Express*, vol. 24, no. 13, pp. 14 748–14 764, 2016.

[49] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020.

[50] D. M. Pelt, K. J. Batenburg, and J. A. Sethian, "Improving tomographic reconstruction from limited data using mixed-scale dense convolutional neural networks," *Journal of Imaging*, vol. 4, no. 11, p. 128, 2018.

[51] F. De Carlo, D. Gürsoy, D. J. Ching, *et al.*, "TomoBank: A tomographic data repository for computational X-ray science," *Measurement Science and Technology*, vol. 29, no. 3, p. 034 004, 2018.

[52] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[54] D. M. Pelt and K. J. Batenburg, "Fast tomographic reconstruction from limited data using artificial neural networks," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5238–5251, 2013.

[55] M. J. Lagerwerf, D. M. Pelt, W. J. Palenstijn, and K. J. Batenburg, "A computationally efficient reconstruction algorithm for circular cone-beam computed tomography using shallow neural networks," *Journal of Imaging*, vol. 6, no. 12, p. 135, 2020.

[56] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb, "Solving inverse problems using data-driven models," *Acta Numerica*, vol. 28, pp. 1–174, 2019.

[57] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9446–9454.

[58] J. Leuschner, M. Schmidt, P. S. Ganguly, *et al.*, "Quantitative comparison of deep learning-based image reconstruction methods for low-dose and sparse-angle CT applications," *Journal of Imaging*, vol. 7, no. 3, 2021.

[59] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 253–279, 2019.

[60] V. Kain, S. Hirlander, B. Goddard, *et al.*, "Sample-efficient reinforcement learning for CERN accelerator control," *Physical Review Accelerators and Beams*, vol. 23, no. 12, p. 124 801, 2020.

[61] D. Van Dyck, J. R. Jinschek, and F.-R. Chen, "'Big Bang' tomography as a new route to atomic-resolution electron tomography," *Nature*, vol. 486, no. 7402, pp. 243–246, 2012.

[62] C.-C. Chen, C. Zhu, E. R. White, *et al.*, "Three-dimensional imaging of dislocations in a nanoparticle at atomic resolution," *Nature*, vol. 496, no. 7443, pp. 74–77, 2013.

[63] P. Rez and M. M. Treacy, "Three-dimensional imaging of dislocations," *Nature*, vol. 503, no. 7476, E1–E1, 2013.

[64] S. Van Aert, K. J. Batenburg, M. D. Rossell, R. Erni, and G. Van Tendeloo, "Three-dimensional atomic imaging of crystalline nanoparticles," *Nature*, vol. 470, no. 7334, p. 374, 2011.

[65] R. J. Gardner and P. Gritzmann, "Discrete tomography: Determination of finite sets by X-rays," *Transactions of the American Mathematical Society*, vol. 349, no. 6, pp. 2271–2295, 1997.

[66] M. Baake, C. Huck, P. Gritzmann, B. Langfeld, and K. Lord, "Discrete tomography of planar model sets," *Acta Crystallographica A*, vol. 62, no. 6, pp. 419–433, 2006.

[67] A. Alpers and P. Gritzmann, "On stability, error correction, and noise compensation in discrete tomography," *SIAM Journal on Discrete Mathematics*, vol. 20, no. 1, pp. 227–239, 2006.

[68] E. W. Montroll, "Theory of the vibration of simple cubic lattices with nearest neighbor interactions," in *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, Univ of California Press, vol. 3, 1956, pp. 209–246.

[69] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*, ser. Applied and Numerical Harmonic Analysis. Birkhäuser/Springer, New York, 2013, pp. xviii+625.

[70] E. J. Candès and C. Fernandez-Granda, "Towards a mathematical theory of super-resolution," *Communications on pure and applied Mathematics*, vol. 67, no. 6, pp. 906–956, 2014.

[71] K. Bredies and H. K. Pikkarainen, "Inverse problems in spaces of measures," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 19, no. 1, pp. 190–218, 2013.

[72] G. S. Alberti, H. Ammari, F. Romero, and T. Wintz, "Dynamic spike superresolution and applications to ultrafast ultrasound imaging," *SIAM Journal on Imaging Sciences*, vol. 12, no. 3, pp. 1501–1527, 2019.

[73] D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*. Elsevier, 2001, vol. 1.

[74] N. Parikh, S. Boyd, *et al.*, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[75] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[76] M. Jaggi, "Revisiting Frank-Wolfe: Projection-free sparse convex optimization.," in *ICML (1)*, 2013, pp. 427–435.

[77] Q. Denoyelle, V. Duval, G. Peyré, and E. Soubies, "The sliding Frank-Wolfe algorithm and its application to super-resolution microscopy," *Inverse Problems*, 2019.

[78] C. Poon and G. Peyré, "Multidimensional sparse super-resolution," *SIAM Journal on Mathematical Analysis*, vol. 51, no. 1, pp. 1–44, 2019.

[79] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.

[80] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, "SciPy 1.0–fundamental algorithms for scientific computing in Python," *arXiv preprint arXiv:1907.10121*, 2019.

[81] J. A. Anderson, C. D. Lorenz, and A. Travesset, "General purpose molecular dynamics simulations fully implemented on graphics processing units," *Journal of computational physics*, vol. 227, no. 10, pp. 5342–5359, 2008.

[82] J. Glaser, T. D. Nguyen, J. A. Anderson, *et al.*, "Strong scaling of general-purpose molecular dynamics simulations on GPUs," *Computer Physics Communications*, vol. 192, pp. 97–107, 2015.

[83] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch, "Structural relaxation made simple," *Physical review letters*, vol. 97, no. 17, p. 170 201, 2006.

[84] P. M. Pardalos, A. Zhigljavsky, and J. Žilinskas, *Advances in stochastic and deterministic global optimization*. Springer, 2016.

[85] R. I. Koning, A. J. Koster, and T. H. Sharp, "Advances in cryo-electron tomography for biology and medicine," *Annals of Anatomy - Anatomischer Anzeiger*, vol. 217, pp. 82–96, 2018.

[86] M. Chen, J. M. Bell, X. Shi, S. Y. Sun, Z. Wang, and S. J. Ludtke, "A complete data processing workflow for cryo-ET and subtomogram averaging," *Nature methods*, vol. 16, no. 11, pp. 1161–1168, 2019.

[87] E. Pyle and G. Zanetti, "Current data processing strategies for cryo-electron tomography and subtomogram averaging," *Biochemical Journal*, vol. 478, no. 10, pp. 1827–1845, 2021.

[88] M. Vulović, R. B. Ravelli, L. J. van Vliet, *et al.*, "Image formation modeling in cryo-electron microscopy," *Journal of structural biology*, vol. 183, no. 1, pp. 19–32, 2013.

[89] O. Öktem, *Mathematics of electron tomography*. 2015, pp. 937–1031, QC 20160218.

[90] F. Amat, D. Castaño-Díez, A. Lawrence, F. Moussavi, H. Winkler, and M. Horowitz, "Alignment of cryo-electron tomography datasets," *Methods in enzymology*, vol. 482, pp. 343–67, Dec. 2010.

[91] C. J. Russo and R. Henderson, "Charge accumulation in electron cryomicroscopy," *Ultramicroscopy*, vol. 187, pp. 43–49, 2018.

[92] A. F. Brilot, J. Z. Chen, A. Cheng, *et al.*, "Beam-induced motion of vitrified specimen on holey carbon film," *Journal of structural biology*, vol. 177, no. 3, pp. 630–637, 2012.

[93] J.-J. Fernandez, S. Li, T. A. Bharat, and D. A. Agard, "Cryo-tomography tilt-series alignment with consideration of the beam-induced sample motion," *Journal of structural biology*, vol. 202, no. 3, pp. 200–209, 2018.

[94] B. A. Himes and P. Zhang, "emClarity: Software for high-resolution cryo-electron tomography and subtomogram averaging," *Nature methods*, vol. 15, no. 11, pp. 955–961, 2018.

[95] D. Tegunov, L. Xue, C. Dienemann, P. Cramer, and J. Mahamid, "Multi-particle cryo-EM refinement with M visualizes ribosome-antibiotic complex at 3.5 Å in cells," *Nature Methods*, vol. 18, no. 2, pp. 186–193, 2021.

[96] G. Chreifi, S. Chen, L. A. Metskas, M. Kaplan, and G. J. Jensen, "Rapid tilt-series acquisition for electron cryotomography," *Journal of structural biology*, vol. 205, no. 2, pp. 163–169, 2019.

[97] S. Q. Zheng, E. Palovcak, J.-P. Armache, K. A. Verba, Y. Cheng, and D. A. Agard, "MotionCor2: Anisotropic correction of beam-induced motion for improved cryo-electron microscopy," *Nature methods*, vol. 14, no. 4, pp. 331–332, 2017.

[98] J. Modersitzki, *Numerical methods for image registration*. OUP Oxford, 2003.

[99] P. S. Ganguly, F. Lucka, H. J. Hupkes, and K. J. Batenburg, "Atomic super-resolution tomography," *arXiv preprint arXiv:2002.00710*, 2020.

[100] G. D. Evangelidis and E. Z. Psarakis, "Parametric image alignment using enhanced correlation coefficient maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1858–1865, 2008.

[101] F. J. Harris, *Multirate signal processing for communication systems*. River Publishers, 2021.

[102] H. Rullgård, L.-G. Öfverstedt, S. Masich, B. Daneholt, and O. Öktem, "Simulation of transmission electron microscope images of biological specimens," *Journal of microscopy*, vol. 243, no. 3, pp. 234–256, 2011.

[103] F. J. Anscombe, "The transformation of Poisson, binomial and negative-binomial data," *Biometrika*, vol. 35, no. 3/4, pp. 246–254, 1948.

[104] D. Maclaurin, D. Duvenaud, and R. P. Adams, "Autograd: Effortless gradients in NumPy," in *ICML 2015 AutoML workshop*, vol. 238, 2015, p. 5.

[105] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, "SciPy 1.0: Fundamental algorithms for scientific computing in python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.

[106] T. Kluyver, B. Ragan-Kelley, F. Pérez, *et al.*, *Jupyter Notebooks–a publishing format for reproducible computational workflows.* 2016, vol. 2016.

[107] S. Kolouri, S. R. Park, M. Thorpe, D. Slepcev, and G. K. Rohde, "Optimal mass transport: Signal processing and machine-learning applications," *IEEE signal processing magazine*, vol. 34, no. 4, pp. 43–59, 2017.

[108] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[109] R. M. Merks, S. V. Brodsky, M. S. Goligorksy, S. A. Newman, and J. A. Glazier, "Cell elongation is key to in silico replication of in vitro vasculogenesis and subsequent remodeling," *Developmental Biology*, vol. 289, no. 1, pp. 44–54, 2006.

[110] P. Carmeliet, "Angiogenesis in life, disease and medicine," *Nature*, vol. 438, no. 7070, pp. 932–936, 2005.

[111] F. De Smet, I. Segura, K. De Bock, P. J. Hohensinner, and P. Carmeliet, "Mechanisms of vessel branching: Filopodia on endothelial tip cells lead the way," *Arteriosclerosis, thrombosis, and vascular biology*, vol. 29, no. 5, pp. 639–649, 2009.

[112] J. Folkman, "Angiogenesis: An organizing principle for drug discovery?" *Nature reviews Drug discovery*, vol. 6, no. 4, pp. 273–286, 2007.

[113] D. Palachanis, A. Szabó, and R. M. Merks, "Particle-based simulation of ellipse-shaped particle aggregation as a model for vascular network formation," *Computational Particle Mechanics*, vol. 2, no. 4, pp. 371–379, 2015.

[114] L. Boninsegna, F. Nüske, and C. Clementi, "Sparse learning of stochastic dynamical equations," *The Journal of chemical physics*, vol. 148, no. 24, p. 241 723, 2018.

[115] F. Noé, A. Tkatchenko, K.-R. Müller, and C. Clementi, "Machine learning for molecular simulation," *Annual Review of Physical Chemistry*, vol. 71, no. 1, pp. 361–390, 2020, PMID: 32092281.

[116] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," *Physical review letters*, vol. 98, no. 14, p. 146 401, 2007.

[117] S. Mukherjee, S. Dittmer, Z. Shumaylov, S. Lunz, O. Öktem, and C.-B. Schönlieb, "Learned convex regularizers for inverse problems," *arXiv preprint arXiv:2008.02839*, 2020.

[118] S. Lunz, O. Öktem, and C.-B. Schönlieb, "Adversarial regularizers in inverse problems," *Advances in neural information processing systems*, vol. 31, 2018.

[119] S. Ravishankar, J. C. Ye, and J. A. Fessler, "Image reconstruction: From sparsity to data-adaptive methods and machine learning," *Proceedings of the IEEE*, vol. 108, no. 1, pp. 86–109, 2019.

[120] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste, "Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks.," *J. Mach. Learn. Res.*, vol. 22, no. 241, pp. 1–124, 2021.

[121] E. D. Zhong, A. Lerer, J. H. Davis, and B. Berger, "Exploring generative atomic models in cryo-EM reconstruction," *arXiv preprint arXiv:2107.01331*, 2021.

List of publications

Publications that are part of this dissertation:

- Atomic Super-resolution Tomography. *P. S. Ganguly, F. Lucka, H. J. Hupkes, K. J. Batenburg.* International Workshop on Combinatorial Image Analysis. Springer, Cham, pp. 45-61, 2020.
- Improving reproducibility in synchrotron tomography using implementation-adapted filters. *P. S. Ganguly, D. M. Pelt, D. Gürsoy, F. de Carlo, and K. J. Batenburg.* Journal of Synchrotron Radiation 28, no. 5, 2021.
- SparseAlign: A Grid-Free Algorithm for Automatic Marker Localization and Deformation Estimation in Cryo-Electron Tomography. *P. S. Ganguly, F. Lucka, H. Kohr, E. Franken, H. J. Hupkes and K. J. Batenburg.* IEEE Transactions on Computational Imaging, vol. 8, pp. 651-665, 2022.

Publications that are not part of this dissertation:

- Quantitative comparison of deep learning-based image reconstruction methods for low-dose and sparse-angle CT applications. *J. Leuschner, M. Schmidt, P. S. Ganguly, V. Andriiashen, S. B. Coban, A. Denker, D. Bauer, A. Hadjifaradji, K. J. Batenburg, P. Maass and M. van Eijnatten.* Journal of Imaging, 7(3), 44, 2021.
- Parallel-beam X-ray CT datasets of apples with internal defects and label balancing for machine learning. *S. B. Coban, V. Andriiashen, P. S. Ganguly, M. van Eijnatten, K. J. Batenburg.* arXiv:2012.13346, 2020.

Samenvatting

This chapter contains a summary of the thesis in Dutch and English. Thanks to Adriaan Graas for his extensive help with the summary in Dutch.

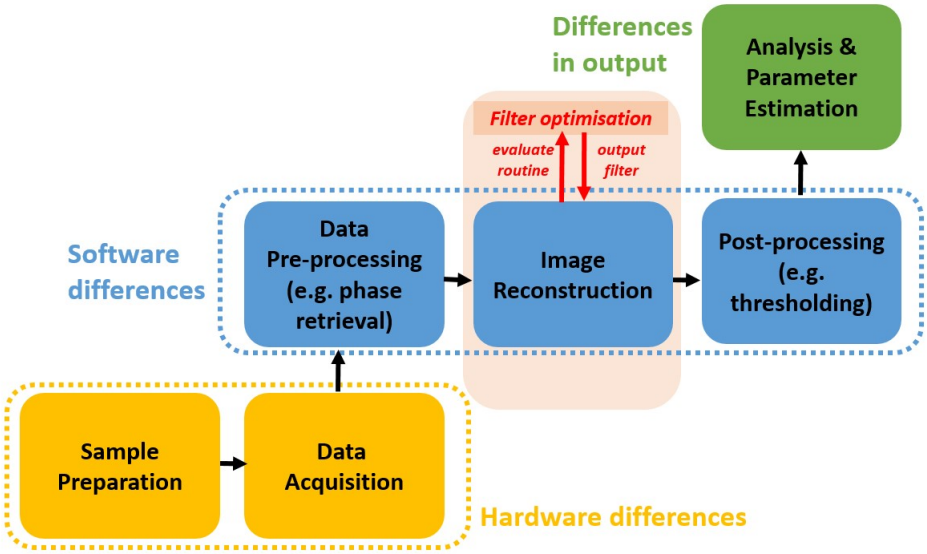
Inverse problemen zijn problemen waarbij we de waarden van bepaalde parameters van een systeem willen schatten, gegeven een aantal waarnemingen van het systeem. Dergelijke problemen komen veelvuldig voor in verschillende gebieden van wetenschap en techniek. Inverse problemen zijn vaak lastig oplosbaar, wat betekent dat de waarnemingen niet op unieke wijze de te schatten parameters kunnen bepalen. Om dergelijke problemen op te lossen moeten we daarom gebruik maken van extra kennis die beschikbaar is over het systeem in kwestie. Een voorbeeld daarvan wordt gegeven door het begrip "ijlheid".

Met "ijlheid" (engels: *sparsity*) wordt bedoeld dat de oplossing van het inverse probleem kan worden uitgedrukt als een combinatie van slechts enkele termen. De ijlheid van een oplossing kan expliciet of impliciet worden bewerkstelligt. Een expliciete manier is door het minimaliseren van het aantal niet-nul termen in de oplossing. Een impliciete manier is, bijvoorbeeld, door een aanpassing te maken in het algoritme dat gebruikt wordt om tot de oplossing te komen.

In dit proefschrift hebben we vier verschillende inverse problemen uit vier verschillende toepassingsgebieden bestudeerd. Per geval laten we zien hoe ideeën over ijle problemen toegepast kunnen worden om tot effectieve algoritmes te komen, en problemen uit de toepassing op te lossen.

In hoofdstuk 2 hebben we het probleem van reproduceerbaarheid in synchrotron tomografie bestudeerd. Hardware en software kunnen sterk verschillen tussen synchrotrons onderling, en de resultaten van experimenten die uitgevoerd worden door gebruikers van verschillende faciliteiten zijn dus niet zonder meer met elkaar te vergelijken. Om die reden hebben we een filter-optimalisatie algoritme ontwikkeld dat het reconstructiedeel van de synchrotron pijplijn verbetert (zie figuur 1). Onze filters zijn uitgedrukt in een ijle basis, wat betekent dat er maar een klein aantal filtercomponenten berekend hoeft te worden.

In hoofdstuk 3 hebben we onze aandacht gericht op de reconstructie van nanokristallen uit een laag aantal projecties, m.b.v. elektronentomografie. Atomic-resolution tomography (een beeldvormende methode voor atomen) van nanokristallen is in het verleden al aangetoond, maar de tot dusver gebruikte algoritmes berustten op de veronderstelling dat atomen op een rooster liggen. Deze aanname maakte het moeilijk om kristaldefecten te reconstrueren, wat juist één van de meest interessante kwaliteiten is van zulke beelden. Wij hebben hiervoor een rastervrij algoritme ontworpen dat in staat was om veelvoorko-

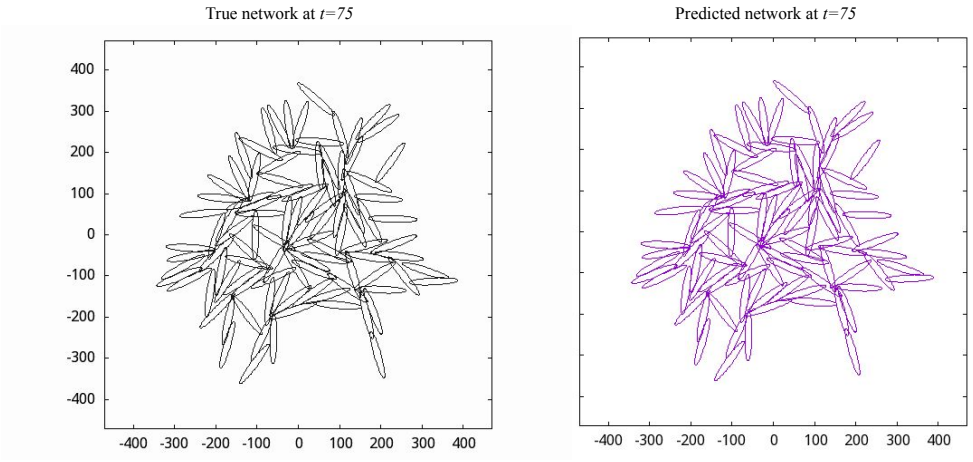


Figuur 1: Schematische voorstelling van een typische tomografie-pijplijn bij synchrotrons. Onze filter-optimalisatie methode is een routine die het reconstructiedeel omvat. De output van onze methode is een filter dat in het reconstructiedeel gebruikt kan worden voor reproduceerbare reconstructies.

mende kristaldefecten te reconstrueren uit gesimuleerde data. We introduceerden ijlheid in de ruimte van atomen door bij elke iteratie slechts één atoom toe te voegen aan de atoomconfiguratie van dat moment. We toonden daarnaast aan dat het meenemen van voorkennis over de potentiële energie van de atoomconfiguratie de nauwkeurigheid van de reconstructie verbetert.

We hebben onze methode voor atomic-resolution tomography verder uitgebreid in hoofdstuk 4, waar we een nieuwe methode voorstellen voor marker-gebaseerde uitlijning in cryo-elektronentomografie. Cryo-elektronentomografie is de methode bij uitstek om de structuur van biologische macromoleculen, zoals eiwitten, in hun oorspronkelijke cellulaire omgeving te bepalen. Uitlijning van projectiebeelden (de zogenaamde tilt-reeks) is een cruciale stap om de resolutie van de uiteindelijke structuren te verbeteren. Bij cryo-elektronentomografie is het bijzonder uitdagend om de plaatselijke vervorming van het monster ten gevolge van de bestraling met de elektronenbundel te corrigeren. Normaal gesproken wordt de uitlijning uitgevoerd met behulp van contrasterende gouden deeltjes als markers, door de deeltjes te volgen in de tilt-reeks. Dit is echter een lastige, tijdrovende en foutgevoelige taak, vooral wanneer de tilt-reeks ruis vertoont. Wij hebben een uitlijningsmethode voorgesteld waarbij het volgen van de markers niet nodig is; in plaats daarvan gebruikt onze methode een model voor de markers en wordt zowel de lokalisatie van de markers als de schatting van de vervorming tegelijkertijd uitgevoerd. We hebben deze methode toegepast op zowel gesimuleerde als echte data, en vergeleken met een

state-of-the-art beeldveranderingsmethode.



Figuur 2: Ware en afgeleide vasculaire netwerken met 100 langgerekte cellen.

In ons laatste hoofdstuk, hoofdstuk 5, onderzochten we de vorming van vasculaire netwerken – i.e., een nieuw bloedvatenstelsel – bij gewervelde dieren. Het ontspruiten en uitbreiden van nieuwe bloedvaten uit een primitief netwerk gebeurt zowel tijdens de ontwikkeling als bij bepaalde kankertypes, waar in dit laatste geval het proces bijdraagt tot het behoud van de tumor en metastasering. Hoe individuele cellen zich organiseren om vasculaire netwerken te vormen wordt nog niet goed begrepen. Eén manier om dit proces te bestuderen is door netwerkvorming op de computer te simuleren met behulp van handmatig ingestelde cel-cel interacties en omgevingsfactoren. Dergelijke simulaties zijn bijzonder nuttig gebleken voor het verhelderen van de minimale condities die nodig zijn voor netwerkvorming. Een aanvullende aanpak is om de cel-cel interacties direct af te leiden uit experimentele studies over netwerkvorming. Voor dit laatste stelden we een aanpak voor waarbij we paarsgewijze interacties tussen cellen leren uit tijdreeksdata. We pasten deze aanpak toe op proof-of-concept experimenten, en toonden aan dat onze methode in staat is om relevante interacties tussen cellen te leren uit een database van mogelijke interactietermen, wat resulteerde in een goede overeenkomst tussen het werkelijke netwerk en het netwerk voorspeld door onze methode (zie figuur 2). In de toekomst hopen we deze methode uit te breiden naar andere simulaties waar zulke interactietermen niet vanzelfsprekend zijn en, tenslotte, naar experimentele data van netwerk-vormende cellen.

Ondanks de verschillen tussen de toepassingsgebieden die in dit proefschrift zijn bestudeerd, hebben we laten zien dat vergelijkbare optimalisatietechnieken op basis van ijlheid gebruikt kunnen worden om verschillende problemen aan te pakken. Dit weerspiegelt het feit dat – zelfs als specifieke eigenschappen van problemen sterk verschillen – er vaak

een onderliggende wiskundige overeenkomst is, en dat die door toegepaste wiskundigen gebruikt kan worden voor het ontwerp van effectieve oplossingen.

English Summary

Inverse problems are problems where we want to estimate the values of certain parameters of a system given observations of the system. Such problems occur in several areas of science and engineering. Inverse problems are often ill-posed, which means that the observations of the system do not uniquely define the parameters we seek to estimate. In order to solve such problems, therefore, we need to make use of additional knowledge about the system at hand. One such prior information is given by the notion of sparsity.

Sparsity refers to the knowledge that the solution to the inverse problem can be expressed as a combination of a few terms. The sparsity of a solution can be controlled explicitly or implicitly. An explicit way to induce sparsity is to minimize the number of non-zero terms in the solution. Implicit use of sparsity can be made, for e.g., by making adjustments to the algorithm used to arrive at the solution.

In this thesis we studied four different inverse problems in four different application areas and showed how ideas of sparsity can be used in each case to design effective algorithms to solve such problems.

In Chapter 2, we studied the problem of reproducibility in synchrotron tomography. Hardware and software vary across synchrotrons, and the results of experiments performed by users at different facilities are not readily comparable with each other. We proposed a filter optimization approach to improve the reconstruction block in the synchrotron pipeline (see Figure 1). Our filters are expressed in a sparse basis, which means that not many filter components have to be computed.

In Chapter 3, we turned to the problem of reconstructing nanocrystals from a few projections using electron tomography. Atomic-resolution tomography of nanocrystals has been demonstrated in the past; however, the algorithms used relied on the assumption that atoms lie on a grid. This assumption made it hard to reconstruct crystal defects, which are often *the* feature of interest in such samples. We devised a grid-free algorithm to reconstruct crystal defects that was able to reconstruct common defects from simulated data. We induced sparsity in the space of atoms by adding only one atom to the current atomic configuration at each iteration. We also showed that making use of physical prior knowledge on the potential energy of the atomic configuration improved reconstruction accuracy.

We extended our method for atomic-resolution tomography to propose a new method for marker-based alignment in cryo-electron tomography in Chapter 4. Cryo-electron tomography is the method of choice for resolving the structure of biological macromolecules, such as proteins, in their native cellular environment. Alignment of projection images (known as a tilt-series) is a crucial step to increasing the resolution of the final structures. In cryo-electron tomography, *local* deformation of the sample due to irradiation with the electron beam is particularly challenging to correct. Usually, alignment is done by using high-contrast gold beads as markers, whose positions are tracked across the tilt-series. However, tracking markers is a difficult, time-consuming and error-prone task,

especially when tilt-series are noisy. We proposed an alignment method *without* the need for marker tracking; instead, our method used a model for the markers and performed marker localization and deformation estimation simultaneously. We applied this method to both simulated and real data, and compared it against a state-of-the-art method for deformation estimation.

In our final chapter, Chapter 5, we investigated the problem of vascular network formation – the formation of a new circulatory system – in vertebrates. Sprouting and expansion of new blood vessels from a primitive network occurs both during development and in certain types of cancer, where this process contributes to tumour maintenance and metastasis. How individual cells self-organize to form vascular networks is poorly understood. One way to study this process it to simulate network formation on a computer using hand-crafted cell–cell interactions and environmental cues. Such simulations have proven to be extremely useful in elucidating the minimal interactions required for network formation. A complementary approach is to infer cell–cell interactions directly from experimental studies of network formation. To that end, we proposed a sparse optimization approach to learn pairwise interactions between cells from time-series data on cells. We applied this approach to proof-of-concept experiments and showed that our method is able to learn the relevant interactions between cells from a library of possible interaction terms, which resulted in a good match between the actual network and the network predicted by our method (see Figure 2). In the future, we hope to extend this method to other simulations where such interaction terms are not evident and, finally, to experimental data of network-forming cells.

Despite the differences between the application areas studied in this thesis, we showed that similar optimization techniques based on sparsity can be used to tackle each problem. This reflects the fact that, although the specifics of each problem vary vastly from the next one, there is an underlying mathematical similarity between the problems that can be used by applied mathematicians to design effective ways to solve them.

Curriculum Vitae

Poulami Somanya Ganguly was born and brought up in Kolkata, India, and attended high school at Loreto Day School Dharamtala and La Martiniere for Girls. She went on to study physics at St Stephen's College, Delhi, and was awarded a BSc (Hons) degree in 2013. In the summers of 2011 and 2012, she conducted biophysics research at the Bose Institute in Kolkata and the National Centre of Biological Sciences in Bangalore. Following her bachelor's studies, she moved to Köln, Germany, for a Master's degree (2015) in theoretical physics at the Bonn Cologne Graduate School. In 2017, she was awarded an MPhil in theoretical and computational biophysics by University College London; her MPhil research on the mechanics of morphogenesis was carried out at the Francis Crick Institute. In 2018 she began her doctoral research on inverse problems at the Centrum Wiskunde & Informatica (CWI) in Amsterdam and Leiden University, as part of the Marie-Sklowdowska Curie Innovative Training Network MUMMERING. She made extended research visits to the Paul Scherrer Institute (Villigen, Switzerland) and Thermo Fisher Scientific (Eindhoven, The Netherlands) as part of her PhD, and was twice awarded travel grants by the Society of Industrial and Applied Mathematics (SIAM) to present her work at the SIAM Imaging Science conference.

Propositions

Propositions accompanying the thesis "**Sparsity-based algorithms for inverse problems**".

1. Reconstruction results from different implementations of fast algorithms can be made more quantitatively similar to each other without knowledge of the underlying details of each implementation (Chapter 2).

2. Pixel- and voxel-based discretization are not always the most appropriate discretizations of an imaging problem, especially when physical or chemical knowledge of the system dictates otherwise (Chapter 3).

3. Model assumptions built into inference algorithms may not always hold in practice. However, practical heuristics are often able to steer algorithms to a good solution (Chapters 3 and 4).

4. The dynamics of multicellular biological systems can be modelled using a partial differential equation with a few terms. Learning-based methods that make use of the sparsity of terms can recover this dynamics from limited data (Chapter 5).

5. New algorithms for solving inverse problems often rely on the revival of older optimization techniques, such as the Frank-Wolfe method, and illustration of their applicability to a larger class of problems (*Revisiting Frank-Wolfe: Projection-free sparse convex optimization*, Jaggi, 2013).

6. Proper utilization of existing prior knowledge, such as the use of atomic models, is crucial for designing effective learning-based methods that minimize the need for additional error-prone steps (*Exploring generative atomic models in cryo-EM reconstruction*, Zhong et al., 2021).

7. Simulation paradigms such as the TEM-simulator provide a useful way to test the properties and shortcomings of an inference algorithm on close-to-realistic data and allow for small changes in simulation settings not easily accessible to experiments (*Simulation of transmission electron microscope images of biological specimens*, Rullgård et al., 2011).

8. Evaluating the performance of state-of-the-art algorithms on benchmark datasets must go hand in hand with interrogating current practices around dataset creation and splitting (*Parallel-beam X-ray CT datasets of apples with internal defects and label balancing for machine learning*, Coban et al., 2020).

9. Understanding why a method fails to give a desired result can be more insightful than a demonstration of cases where it succeeds; scientific "results" that question and debunk are much rarer – and therefore deserve more importance – than those that showcase improved performance.

10. There is no better way for a teacher to understand and correct their own misconceptions about a topic than conversing and interacting with a group of students.

Poulami Somanya Ganguly,
Amsterdam; October 28, 2022.