Automatic and efficient tomographic reconstruction algorithms

Lagerwerf, M.J.

Chapter 1

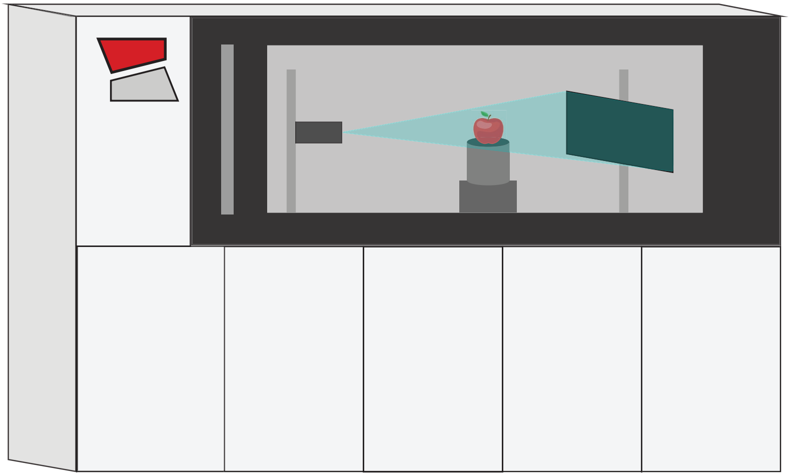
Introduction

Tomography deals with imaging the interior of an object without destroying it. It is a useful tool in many applications in science, industry and medicine. In tomography a penetrating wave is used to measure projection images of an object along different directions. These projection images are then used to determine the interior of the object, through a *tomographic reconstruction method* [KS01; Nat01; Her09].

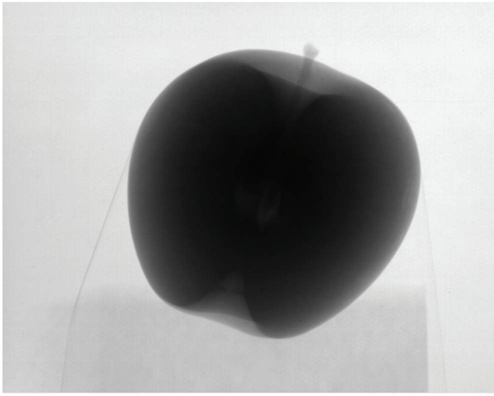
A popular type of tomography is *computed tomography* (CT), where X-rays are used as the penetrating waves to measure the projection images. Within CT imaging there many different types of scanners, such as medical CT scanners, μ-CT scanners (laboratory setup) and synchotron facilities. All these scanners follow a similar principle. A *source* generates X-rays which penetrate the measured object. Inside this object the X-rays are attenuated through interaction with the object. The amount of attenuation depends on the material properties of the object and the energy distribution of the X-rays. After interacting with the object the intensity of the X-ray beams is measured with a *detector* forming a projection image, see **Figure 1.1**a. This process is repeated for several source positions and detector positions[1] and all the resulting projection images combine into the *measured projection data*. This measured projection data is then used as the input for a CT specific reconstruction method that computes the interior of the object **Figure 1.1**b.

The problem of computing the interior of the object from the measured projection data is called the *reconstruction problem*. The difficulty of the reconstruction problem depends on the amount of information available and the uncertainty in the measured projection data, *e.g.*, the number of projection images and the noise

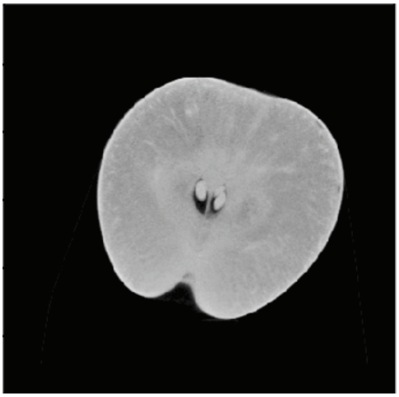
[1] In some applications the source and detector are fixed and the object rotates. This is equivalent to rotating the source and detector position.



(a) Illustration of a CT scanner.



(b) Projection image.



(c) 2D slice of a 3D reconstruction.

Figure 1.1: Examples of a CT scanner, a projection image and a reconstruction. (a) Illustration of a CT scanner. The source in the CT scanner generates X-rays penetrating the apple and the detector measures the intensity of the X-ray beam after interacting with the apple. (b) These measurements form a projection image. (c) From a collection of projection images a reconstruction can be computed showing the interior of the apple.

levels in the projection images. Consequently, the effectiveness of a reconstruction method depends on the amount of information and uncertainty in the measured projection data and how the reconstruction method handles this (lack of) information and uncertainty. For example, direct inversion reconstruction methods are

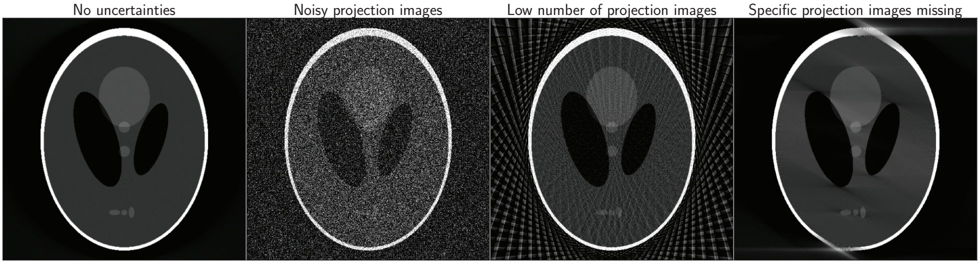


Figure 1.2: Reconstructions using a direct inversion reconstruction method (Filtered backprojection) with a different amount of information or uncertainties in the measured projection data. From left to right: 360 projection images with no noise (close to ground truth), 360 projection images with high noise levels (uncertain data), 36 projection images with no noise (insufficient data), the first 300 projection images from the left case (insufficient data).

derived with the assumption that there is enough information and no noise in the measured projection data and they may fail to compute accurate reconstructions when these conditions are not met. This is illustrated in **Figure 1.2**.

CT imaging is used in a broad spectrum of applications, such as industrial quality control [GUV11], materials sciences [Die+14; Bul+16] and medical imaging [For+02; GKT17]. However, the CT scans in these applications are typically processed *offline*, *i.e.*, the reconstructions are computed after scans are acquired. If instead the reconstruction can be computed while the scan is acquired, *i.e.*, in real-time or *online*, the operator of the CT scan can react to insights gained from the reconstruction. For example, practical problems related to the acquisition parameters — *e.g.*, misaligned source or detector, incorrect center of rotation, or incorrect field of view — could be fixed while scanning. In industrial quality control manual inspection could be replaced by real-time scanning and automatic removal from the assembly line. Furthermore, dynamic processes could be scanned and followed as they occur. Consider for example a dynamic process where external parameters such as pressure or heat are applied to an object. By following these processes in real-time the operator can adjust the parameters when specific events occur, such as the small cracks due to pressure or overheating.

In real-time CT imaging, there are limitations on the scanning time — *i.e.*, the scans should be acquired real-time — and on the reconstruction process — *i.e.*, computing the reconstructions should be real-time. This means for the scanning process that the exposure time per projection image should be low (this will lead to high noise levels) and the number of projection images should be small (this leads to limited information). Consequently, this means for the reconstruction method that it should be able to compute an accurate reconstruction from projection data with limited amount of information and uncertain data in a short amount of time. This example shows that depending on the restrictions introduced by the application a different type of reconstruction method will be most effective.

The research presented in this thesis follows a plug-and-play strategy, *i.e.*, we focus on practical limitations of existing reconstruction methods and develop new strategies to make these methods more effective or easier to use. This strategy is suggested in [PSV09], because they observe that many promising reconstruction methods are not used in practice due to limited consideration on how to effectively apply these new reconstruction methods in practice.

Chapter 2 and **Chapter 3** of this thesis focus on developing mathematical frameworks to pick the correct parameter for reconstruction methods. It is often not clear how to choose the correct parameter and in some cases even the effect of the parameter choice is not clear. This means that in practice the parameter choice becomes a process of trial-and-error requiring manual tuning of the parameters and a good understanding of the reconstruction method. We develop a framework for streamlining the process of picking the correct regularization parameter for variational methods in **Chapter 2**. The idea of this framework is to efficiently compute approximations of reconstructions through pixel-wise interpolation for a broad range of regularization parameters. These approximations are then used to (visually) determine the correct regularization parameter. In **Chapter 3** we formulate an optimization problem which can be used to automatically compute a filter that is adapted to the measured projection data for the FDK algorithm. We show that these computed filters achieve similar performance as optimally smoothed standard filters.

In **Chapter 4** and **Chapter 5** we focus on developing reconstruction methods for 3D CT imaging applications where both reconstruction time and scanning time are a constraint, such as the examples discussed before. The challenge with the combination of these restrictions lies in balancing the reconstruction time and the reconstruction accuracy. This is because reconstruction methods that can accurately reconstruct measured projection data containing noise or a low number of projection data generally do not satisfy the reconstruction time constraints and vice versa. Therefore, we adapt existing filtered-backprojection reconstruction methods — which are known for their short reconstruction times — to improve their reconstruction accuracy for noisy projection data and data with a low number of projection images, while maintaining their computational efficiency. Specifically, we expand upon the Neural Network filtered-backprojection (NN-FBP) algorithm [PB13]. The NN-FBP algorithm is an adaptation of the standard FBP algorithm where a machine learning component is added to greatly improve the reconstruction accuracy of the algorithm. In **Chapter 4** we extend the NN-FBP algorithm to the FDK algorithm and show that this is possible for any linear FBP-type method. In **Chapter 5** we fit the NN-FBP algorithm to a real-time quasi-3D reconstruction framework (RECAST3D) [Buu+18] and replace the supervised learning strategy with a semi-supervised learning strategy proposed in [HPB20]. This leads to the Noise2Filter (N2F) algorithm, a reconstruction method that can be trained on the fly and can compute arbitrarily oriented 2D slices of a 3D volume in real-time.

In this thesis we will use simulated and experimental data. We consider three scanning geometries: parallel beam, fan beam and circular cone-beam. The experimental fan and cone-beam data was acquired using the custom-built and highly flexible FleX-ray CT scanner, developed by XRE NV and located at CWI [Cob+20], and the experimental parallel beam data was taken from the public TomoBank repository [De +18]. More specifically, the fuel cell data used for the TomoChallenge, which was acquired at the TOMCAT beamline at the Swiss Light Source (Paul Scherrer Institut, Switzerland).

In the remainder of this chapter we give a mathematical description — continuous and discrete — of the tomographic reconstruction problem and introduce several reconstruction methods used throughout this thesis.

1.1 Tomographic reconstruction problem

This section gives a mathematical introduction to the idealized tomographic reconstruction problem. Specifically, we introduce the continuous formulation and the scanning geometries in **Section 1.1.1**, the discrete formulation of the reconstruction problem in **Section 1.1.2**, and discuss the Radon transform and the Ray transform in **Section 1.1.3**. Note that all following chapters are based on self-contained articles, each containing a separate introduction. Therefore, we give a more general introduction here.

1.1.1 Continuous formulation

We model the scanned object as a function $f : \mathbb{R}^n \to \mathbb{R}$ in the image function space X with $n \in \{2, 3\}$ and $f(x)$ representing the attenuation coefficient of the scanned object at position $x \in \mathbb{R}^n$. The photon count $I(l)$ for (monochromatic) X-rays traversing the object along the line l (called *X-ray line* from this point onwards) can be expressed in terms of the emitted photon count I_0 at the source and the

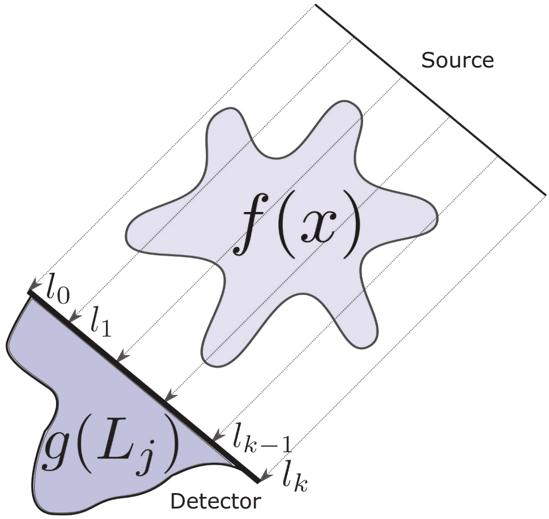


Figure 1.3: Schematic representation of a projection image for a 2D parallel beam CT scanning geometry. The dotted arrows represent the X-ray lines. We define the projection image $g(L_j)$ as the image formed by the values $g(l_i)$ with $l_i \in L_j$ the measured X-ray lines for a fixed source and detector position.

attenuation coefficient function f using the Beer-Lambert law:

$$I(l) = I_0 e^{-\int_l f(x)dx}, \tag{1.1}$$

where we assume that f is bounded with compact support.

We can simplify (1.1) by rearranging the terms and taking the logarithm:

$$-\log\left(\frac{I(l)}{I_0}\right) = \int_l f(x)dx. \tag{1.2}$$

This is the linearized photon count along the X-ray line l, which we will denote by $g(l)$. Representing the projection data by this function $g \in Y$ enables us to formulate the linear forward operator $K : X \to Y$, with Y the projection data function space. More specifically,

$$g(l) = -\log\left(\frac{I(l)}{I_0}\right), \qquad (Kf)(l) = \int_l f(x)dx. \tag{1.3}$$

Up till now we have only considered one X-ray line l, however, the projection data is known for a (possibly infinite) set of measured X-ray lines l.

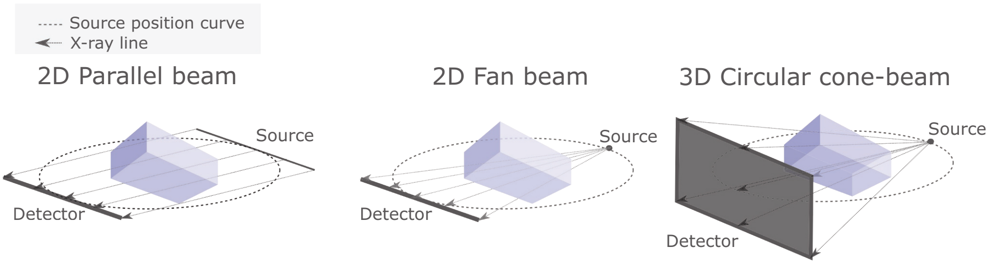


Figure 1.4: Different scanning geometries. In the 2D parallel beam and 2D fan beam scanning geometries, each projection image is a 1D image and the reconstruction is a 2D slice of the measured object, whereas in the 3D circular cone-beam geometry each projection image is a 2D image and the reconstruction is a 3D image of the measured object.

Let us define the set of measured X-ray lines for a fixed source and detector position as L_j. The projection image j is the image formed by the linearized photon counts $g(l_i)$ along the X-ray lines $l_i \in L_j$, which we denote (with a small abuse of notation) by $g(L_j)$. In a similar fashion we define L as the set of all measured X-ray lines for all considered source and detector positions and $g(L)$ then forms the projection data.

We can now formulate the tomographic reconstruction problem. Given the forward operator $K : X \to Y$ and projection data $g \in Y$ find a function $f \in X$ that satisfies:

$$g(l) = (Kf)(l), \qquad \text{for all } l \in L. \tag{1.4}$$

Note that this is an idealized version of the reconstruction problem as we assume there are no practical problems, such as measurement noise, photon scattering or misalignment of the source and the detector.

The set L is determined by the position and properties of the detector — *e.g.*, the shape, number of detector pixels and the physical size of the detector — and the position and properties of the X-ray source — *e.g.*, multiple X-ray sources emitting parallel beams, or a point source emitting X-rays in all directions. These properties form the *scanning geometry*. Common examples of scanning geometries are parallel beam, fan beam for 2D tomographic problems and circular cone-beam for 3D tomographic problems. Schematic representations of these scanning geometries are given in **Figure 1.4** for a fixed source and detector position.

An essential operator for the tomographic reconstruction problem is the adjoint $K^* : Y \to X$, or *backprojection operator*, of the forward operator K. This operator is defined through the following condition:

$$\langle Kf, g\rangle_Y = \langle f, K^* g\rangle_X, \qquad \text{for all } f \in X, g \in Y. \tag{1.5}$$

The backprojection operator is used in many theoretical derivations and reconstruction algorithms [Nat01]. In **Section 1.1.3** we derive the explicit forms for the backprojection operators.

1.1.2 Discrete formulation

In practice only a finite number of X-ray lines can be measured, *i.e.*, the number of elements of the set L is finite. Therefore, it is natural to consider the projection data as a finite dimensional vector $\mathbf{y} \in \mathbb{R}^M$ with each element relating to the linearized measured photon count along an X-ray line l, and M the number of X-ray lines in L. For the discretization of the scanned object we assume that our object is contained in a rectangular box and discretize this box in a number of elements, called *pixels* or *voxels*, for 2D or 3D objects, respectively. We define a vector $\mathbf{x} \in \mathbb{R}^N$, where the elements of the vector correspond to the attenuation coefficient on the elements of the discretized box and let N denote the number of elements in this box.

A natural way of relating $\mathbf{y}_i$ — the i-th element of $\mathbf{y}$ — to the vector $\mathbf{x}$ is by approximating a line integral through the discretized box over the line l_i by a weighted sum over the elements of $\mathbf{x}$. More specifically, given the weight vector $w_i \in \mathbb{R}^N$ with elements $w_{ij} \in \mathbb{R}$, we have the relation:

$$\mathbf{y}_i = \sum_{j=1}^{N} w_{ij}\mathbf{x}_j = w_i^T \mathbf{x}. \tag{1.6}$$

The implementation choice of approximating the line integral leads to different weight vectors $w_i \in \mathbb{R}^N$. Moreover, the approximation can be adapted to attain desirable numerical properties or achieve better performance [Aar+15].

If we consider the matrix $W \in \mathbb{R}^{M\times N}$ with rows w_i, we can relate the vector $\mathbf{x}$ to $\mathbf{y}$ and formulate the discrete tomographic reconstruction problem: given measured projection data $\mathbf{y} \in \mathbb{R}^M$ find a reconstruction $\mathbf{x} \in \mathbb{R}^N$ such that the following holds approximately:

$$W\mathbf{x} = \mathbf{y}. \tag{1.7}$$

This matrix W is often referred to as the *projection matrix*.

Analogously to the continuous formulation, the adjoint[2] of W is the backprojection operator W^T, which is essential in many reconstruction methods.

[2] As W is real-valued, the adjoint is equivalent to the transpose.

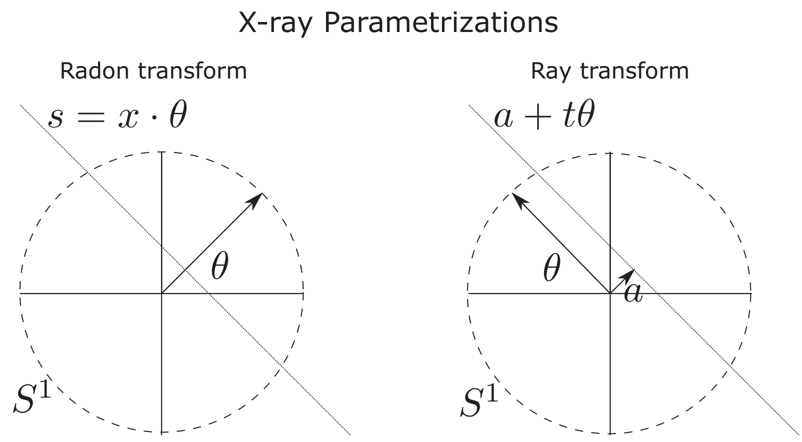


Figure 1.5: Illustrations of the X-ray parametrizations for the Radon transform and the Ray transform in $\mathbb{R}^2$.

Although we did not consider practical problems in these idealized formulations of the reconstruction problems, they are often present in reality. These practical problems introduce additional uncertainties to the reconstruction problem. Specifically, the reconstruction problem might not have a solution, it might have multiple solutions, or the solution varies heavily with respect to changes in the projection data. Therefore, it is important to develop reconstruction methods that take these challenges into consideration.

1.1.3 The Radon and Ray transform

In this section we focus on the explicit expressions for the Radon and the Ray transform and derive the adjoint for both. Note that most of this section goes beyond the scope of the main chapters of this thesis.

The Radon transform integrates a function f on $\mathbb{R}^n$ over hyperplanes. More specifically, given a hyperplane $\{x \in \mathbb{R}^n | s = x \cdot \theta\}$, with $\theta \in S^{n-1}, s \in \mathbb{R}$ and S^{n-1} the unit sphere in $\mathbb{R}^n$, the Radon transform is:

$$(Rf)(\theta, s) = \int_{\mathbb{R}^n} \delta(s - x \cdot \theta) f(x) dx \tag{1.8}$$

Since hyperplanes in $\mathbb{R}^2$ are just lines we can conclude that the Radon transform is in fact identical to the forward operator K from (1.3) for $\mathbb{R}^2$. For $\mathbb{R}^3$, however, any hyperplane is a plane, showing us that the Radon transform does not fit the forward model for the 3D reconstruction problem.

For the Ray transform a more general line parametrization is used, see **Figure 1.5** (right). Given a direction $\theta \in S^{n-1}$ of a line in $\mathbb{R}^n$, we define the hyperplane $\theta^\perp$ in $\mathbb{R}^n$, which is the hyperplane orthogonal to the vector θ, *i.e.*,

$$\theta^\perp := \left\{ a \in \mathbb{R}^n |\, a \cdot \theta = 0, \theta \in S^{n-1} \right\}. \tag{1.9}$$

The parametrization of the line then becomes for an $a \in \theta^\perp$ and $\theta \in S^{n-1}$:

$$l(\theta, a) := \{ a + t\theta | t \in \mathbb{R} \}. \tag{1.10}$$

Note that this parametrization can be used in $\mathbb{R}^n$ for $n \geq 2$. If we subsitute this parametrization in the right-hand side of (1.4) we get the explicit expression for the Ray transform:

$$(Pf)(\theta, a) = \int_{\mathbb{R}} f(a + t\theta) dt. \tag{1.11}$$

To conclude this section, we derive the expressions for the adjoint operators related to the Radon and the Ray transform using (1.5). Following [Nat01] we take X and Y to be a Schwartz space $\mathscr{S}(\cdot)$ on the domain of f and g, respectively.

For the Radon transform we use the definition of the Dirac δ function to get:

$$\langle Rf, g \rangle_{\mathscr{S}(S^{n-1} \times \mathbb{R})} = \int_{S^{n-1}} \int_{\mathbb{R}} \int_{\mathbb{R}^n} \delta(s - x \cdot \theta) f(x) g(\theta, s) dx ds d\theta, \tag{1.12}$$

$$= \int_{\mathbb{R}^n} \int_{S^{n-1}} f(x) g(\theta, x \cdot \theta) d\theta dx = \langle R^* g, f \rangle_{\mathscr{S}(\mathbb{R}^n)}, \tag{1.13}$$

$$(R^* g)(x) = \int_{S^{n-1}} g(\theta, x \cdot \theta) d\theta. \tag{1.14}$$

In a similar fashion we derive the adjoint operator for the Ray transform. Note that in this case the domain of g differs from the Radon transform, *i.e.*, $g \in \mathscr{S}^2(S^{n-1} \times \theta^\perp)$.

In this derivation we substitute $x = a + t\theta$, which implies[3] $t = x \cdot \theta$, *i.e.*,

$$\langle Pf, g\rangle_{\mathscr{S}(\mathscr{S}(S^{n-1}\times\theta^{\perp}))} = \int_{S^{n-1}}\int_{\theta^{\perp}}\int_{\mathbb{R}} f(a+t\theta)g(\theta,a)dt\,da\,d\theta, \tag{1.15}$$

$$= \int_{\mathbb{R}^n}\int_{S^{n-1}} f(x)g(\theta, x-(x\cdot\theta)\theta)d\theta\,dx, \tag{1.16}$$

$$= \int_{\mathbb{R}^n}\int_{S^{n-1}} f(x)g(\theta, E_\theta(x))d\theta\,dx = \langle P^* g, f\rangle_{\mathbb{R}^n}, \tag{1.17}$$

$$(P^*g)(x) = \int_{S^{n-1}} g(\theta, E_\theta(x))d\theta, \tag{1.18}$$

with $E_\theta(x) = x-(x\cdot\theta)\theta$ the orthogonal line projection of x onto $\theta^{\perp}$.

If we compare the adjoint operators for the Radon ($n = 2$) and Ray transform, we observe similar behavior. Specifically, the adjoint operator computes the integral of the measured projection data over all X-ray lines that contain the point x. In practice this means that the adjoint, or backprojection, *smears out* the measured projection data over the reconstruction volume.

Further details about the Radon, Ray transform are given in [Nat01].

1.2 Reconstruction methods

In this thesis many different reconstruction methods are used. In this section we give an introduction to these methods. We mainly consider the discretized versions and omit detailed derivations.

Reconstruction methods can roughly be subdivided in three categories: (1) Direct methods, which are often based on an (approximate) closed form inverse of the operator K, (2) Iterative methods, which solve the tomographic reconstruction problem by using an iterative optimization scheme, (3) Machine learning methods, which use a data-driven approach to remove artifacts from reconstructions or improve existing reconstruction methods.

1.2.1 Direct methods

Direct reconstruction methods are closed form and are often designed for a particular scanning geometry, limiting their general use. Examples of direct methods are GridRec [OSu85], Katsevich [Kat03], the filtered backprojection algorithm [Nat01], and the Feldkamp-Davis-Kress algorithm [FDK84].

[3] This equality is found by taking the inner product with θ of $x = a + t\theta$ on both sides and using $a \in \theta^{\perp}$

Filtered backprojection algorithm

The filtered backprojection (FBP) algorithm is derived for the 2D tomographic reconstruction problem using properties of the Radon transform. Specifically, consider **Theorem 2.3** from [Nat01]. This theorem states that, given functions $f, V \in X$ and $g, v \in Y$ that satisfy $g = Rf$ and $V = R^* v$ — with R and R^* the Radon transform and its backprojection operator, respectively — the following holds true:

$$V * f = R^*(v * g). \tag{1.19}$$

with $*$ denoting a two-dimensional convolution over $\mathbb{R}^2$ on the left-hand side and a one-dimensional convolution over $\mathbb{R}$ along the detector width on the right-hand side.

By taking v to be the inverse Fourier transform ($\mathscr{F}^{-1}$) of the absolute value of the frequencies in the Fourier domain, *i.e.*, $v = \mathscr{F}^{-1}\{|\xi|\}$, one can show that V is the Dirac δ function. This means that for this choice of v the left-hand side of (1.19) simplifies to f and the right-hand side is the expression used for the Filtered Backprojection (FBP) algorithm. This function v is called the *ramp filter* and with an additional cut-off function it is referred to as the *Ram-Lak filter* [RL71]. Note that if g does not contain noise and is available for all possible measured X-ray lines (1.19) is an exact inversion of the Radon transform.

Similar to (1.19) we can formulate a discrete expression used in the FBP algorithm in terms of $\mathbf{x}, \mathbf{y}, W$ and the discretized ramp filter $\mathbf{h}_{\mathrm{r}}$:

$$\mathbf{x}_{\mathrm{FBP}} = W^T(\mathbf{y} * \mathbf{h}_{\mathrm{r}})_{1D}, \tag{1.20}$$

with the convolution $(\cdot * \cdot)_{1D}$ applied in one dimension over the detector width.

The FBP algorithm can be applied to the 2D parallel beam and fan beam scanning geometry. Moreover, it can be applied to the 3D parallel beam and 3D fan beam scanning geometries as these can be considered a stack of their respective 2D reconstruction problems.

Feldkamp-Davis-Kress algorithm

The Feldkamp-Davis-Kress (FDK) algorithm [FDK84] is a reconstruction algorithm for the circular cone-beam scanning geometry. This geometry does not satisfy the Tuy-Kirrilov condition [Tuy83] meaning that it inherently has insufficient information for unique inversion. Therefore, instead of directly inverting the Ray transform for this geometry, the authors propose considering the cone-beam projection data as a stack of fan beam data and using the FBP algorithm for fan beam data in combination with a reweighting of the data that aims to compensate

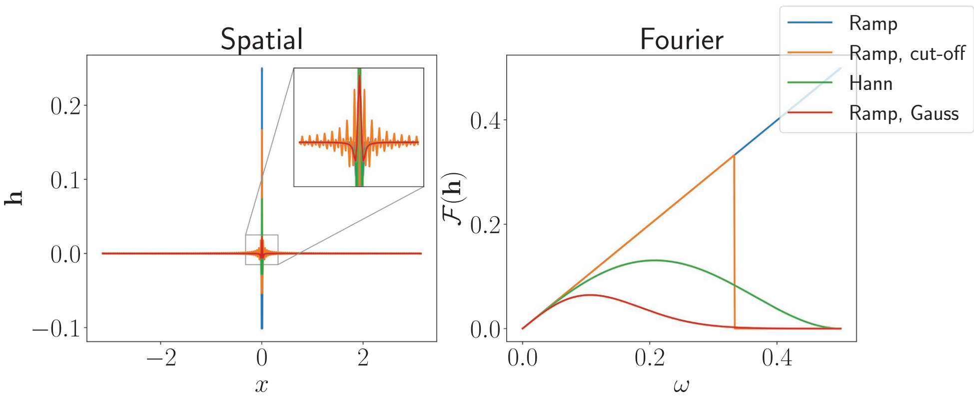


Figure 1.6: Examples of standard filters and adapted standard filters.

for the mismatch between this assumption and the actual cone-beam geometry. The expression used in the FDK algorithm in terms of $\mathbf{x}, \mathbf{y}, W$ and $\mathbf{h}_\mathrm{r}$ is:

$$\mathbf{x}_{\mathrm{FDK}} = W^T(r(\mathbf{y}) * \mathbf{h}_\mathrm{r})_{1D}. \tag{1.21}$$

with $r(\cdot)$ the reweighting operator. Note that W^T is the backprojection operator related to the 3D cone-beam geometry and not the backprojection operator related to the 2D fan beam geometry.

Filter adaptation for FBP-type methods

The ramp filter follows from a theoretical result based on the assumption that there is no noise in the measured projection data. However, this is often not the case in practice, therefore, filter adaptations to the ramp filter have been suggested, such as the Shepp-Logan filter, the Cosine filter, and the Hann filter. These filters put less emphasis on higher frequencies to reduce the noise in the reconstruction. This strategy can be taken further by applying smoothing filters to these filters — such as Gaussian or Binomial filters — or cutting the higher frequencies of by setting their contributions to zero. Examples of these standard and adapted filters are given in **Figure 1.6** and examples of reconstructions of noisy projection data is shown in **Figure 1.7**.

1.2.2 Iterative methods

Iterative methods aim to solve the tomographic reconstruction problem by refining the solution over a number of iterations. Note that these methods often do not rely

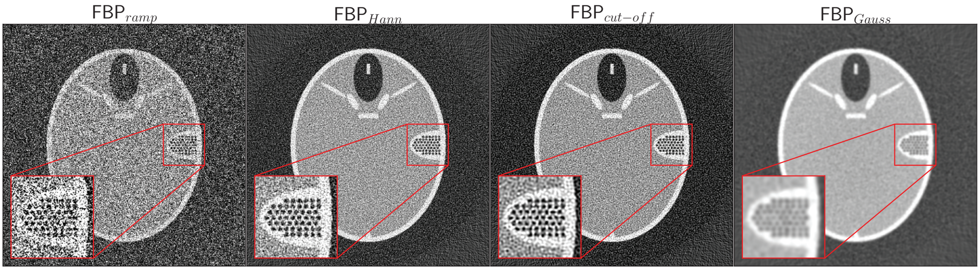


Figure 1.7: Examples of reconstructions with standard filters and adapted standard filters applied to projection data with high noise levels. We see that the filters with less emphasis on the high frequencies contain less noise at the cost of smoother edges. These reconstructions were computed within a second.

on specific properties of the forward operator except for being a linear operator, meaning that these methods can be applied to reconstruction problems with any scanning geometry. There are many different iterative methods, *e.g.*, SIRT [VV90], (S)ART [Kac37; GBH70; AK84], ICD [Wat94], and variational methods [ROF92; GHO99; BKP10; Goc16]. We highlight SIRT and variational methods as these will be used throughout the thesis.

SIRT

The SIRT algorithm [VV90] is a common iterative method for CT reconstruction. An iteration in the SIRT algorithm is defined as follows:

$$\mathbf{x} \leftarrow \mathbf{x} + \omega C W^T R(\mathbf{y} - W\mathbf{x}). \tag{1.22}$$

with $\omega \in \mathbb{R}$ an optional relaxation parameter and

$$C = \operatorname{diag}(\mathbf{c}), \qquad c_j^{-1} = \sum_i W_{ij}, \tag{1.23}$$

$$R = \operatorname{diag}(\mathbf{r}), \qquad r_i^{-1} = \sum_j W_{ij}. \tag{1.24}$$

Simple prior information, such as non-negativity, can be used to improve the results of the SIRT algorithm. The update with this non-negativity constraint becomes:

$$\mathbf{x} \leftarrow \max(\mathbf{x} + \omega C W^T R(\mathbf{y} - W\mathbf{x}), 0). \tag{1.25}$$

We will refer to SIRT with a non-negativity constraint as SIRT$^+$.

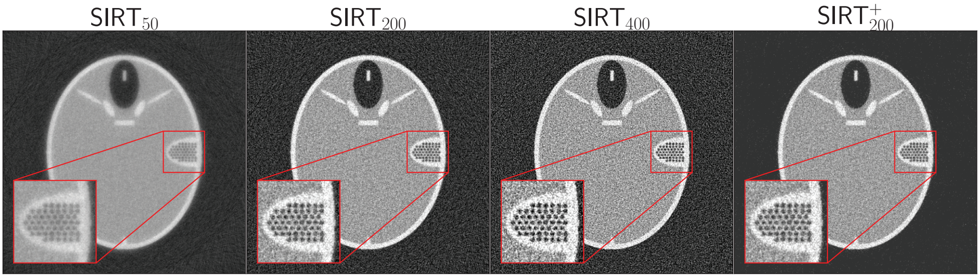


Figure 1.8: Examples of reconstructions with the SIRT algorithm applied to projection data with high noise levels. We see that fewer iterations lead to less noise, but also lower contrast and blurrier edges in comparison to higher iteration reconstructions. Moreover, we observe that the background for SIRT with non-negativity is almost noiseless. These reconstructions are more than a hundred times slower than the earlier shown FBP reconstructions.

In **Figure 1.8** we show example reconstructions of the SIRT algorithm. Here the subscript indicates the number of iterations that were computed.

Note that the SIRT algorithm is closely related to gradient descent applied to the standard least squares problem related to (1.7). By taking $C = R = Id$ instead of the above definition the iteration coincides with an gradient descent iteration applied to the least squares problem.

Variational methods

Variational methods are a class of iterative methods where the reconstruction is the solution to a minimization problem related to the tomographic reconstruction problem. A general (discrete) formulation for the minimization problem is

$$\mathbf{x}_{\mathrm{VM},\lambda} = \underset{\mathbf{x}}{\operatorname{argmin}} \{\mathscr{D}(W\mathbf{x},\mathbf{y}) + \lambda\mathscr{R}(\mathbf{x})\}\,. \tag{1.26}$$

with $\mathscr{D}$ the data fidelity term, $\mathscr{R}$ the regularizer and $\lambda > 0$ the regularization parameter. The data fidelity measures the distance between the data and the forward projection of the reconstruction $W\mathbf{x}$. A common choice is simply the squared difference $\mathscr{D}(W\mathbf{x},\mathbf{y}) = \|W\mathbf{x}-\mathbf{y}\|_2^2$. The regularizer promotes certain properties of the reconstruction $\mathbf{x}$, for example taking $\mathscr{R}(\mathbf{x}) = \|\,|\nabla\mathbf{x}|\,\|_2$ promotes piece-wise constant reconstructions, because $\mathscr{R}(x)$ is large for $\mathbf{x}$ with a large gradient. The regularization parameter λ balances the data fidelity and the regularizer, *i.e.*, taking λ close to zero emphasises the data fidelity and taking λ large emphasises the

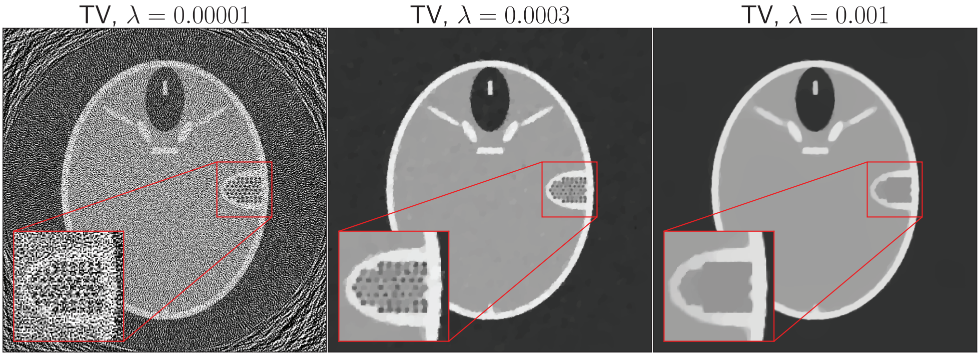


Figure 1.9: Examples of reconstructions with TV regularization applied to projection data with high noise levels. We see that taking λ too low will lead to noisy reconstructions and taking λ too high will lead to over-regularized reconstructions. These reconstructions were computed within a minute.

Method	Data fidelity	Regularizer
Tikhonov regularization	$\|\|W\mathbf{x}-\mathbf{y}\|\|_2^2$	$\|\|\mathbf{x}\|\|_2^2$
Sobolev regularization	$\|\|W\mathbf{x}-\mathbf{y}\|\|_2^2$	$\|\| \|\nabla\mathbf{x}\| \|\|_2^2$
Total Variation (TV) regularization	$\|\|W\mathbf{x}-\mathbf{y}\|\|_2^2$	$\|\| \|\nabla\mathbf{x}\| \|\|_1$

Table 1.1: Variational methods with the corresponding terms for the data fidelity and the regularizer.

regularizer. Examples of the effect of λ on the TV regularization reconstructions are shown in **Figure 1.9** for noisy projection data.

Some common variational methods with the corresponding choices for data fidelity and regularizer are given in **Table 1.1**. Depending on the choices for $\mathscr{D}$ and $\mathscr{R}$ the properties of (1.26) differ and different optimization schemes are needed. For example, gradient descent can be used for optimizing Tikhonov and Sobolev regularization, whereas for TV regularization FISTA [BT09] or PDHG [CP11] schemes are needed. The choice for regularization parameter λ depends on many different properties of the reconstruction problem and the variational method.

1.2.3 Machine learning methods

Using *machine learning* methods is an emerging approach in CT imaging [Wan+18] and has shown promising results for many applications within the development of

CT reconstruction methods [KMY17].

One well established strategy is training a network to remove the artifacts from the output of a standard reconstruction method. This is often called *post-processing* [RFB15; KMY17; PS18; Jin+17]. The promise of these methods is aided by the fact that the post-processing problem can be viewed as a classic image enhancement problem — *e.g.*, denoising, inpainting, or deblurring — for which many effective machine learning methods have already been developed [SLD17; PCC18; Zha+17]. We will introduce the general post-processing strategy below, because this strategy is used in the main chapters of this thesis.

Another strategy is incorporating machine learning components in existing reconstruction methods. Examples of these are variational networks [Kob+17; Ham+18], plug and play priors [VBW13; REM17; RS18] and learned regularizers [LÖS18; Muk+20]; all introduce a machine learning component to various variational methods. Additionally, in [SLX+16; AÖ18; WKL19] a network is proposed that learns an iterative scheme. Lastly, for direct methods the Neural Network Filtered-backprojection (NN-FBP) [PB13] and Neural Network FDK algorithms (**Chapter 4**) were developed.

Post-processing

In general the idea is to find an image-to-image mapping that can remove artifacts from a reconstruction. This mapping is found by defining a set of possible functions — *i.e.*, fixing a neural network architecture — and determining the best functions from this set by determining the best possible mapping on problems for which we know the answer — *i.e.*, using *supervised learning* [HTF09] to determine the network parameters. The network architectures used for post-processing are often *convolutional neural networks* (CNNs), which means that the set of possible functions is a concatenation of convolutions where the weights of the convolution are learned.

More specifically, given a network architecture CNN_Θ with *trainable parameters* Θ and a *training set* containing T *training pairs*, where a training pair consists of an input reconstruction with artifacts $\mathbf{x}_{i,rec}$ and a target reconstructions $\mathbf{x}_{i,target}$ without artifacts. We can train the network CNN_Θ by finding the $\Theta^\star$ that minimizes the *loss function*:

$$\sum_{i=1}^{T} ||\mathrm{CNN}_\Theta(\mathbf{x}_{i,\, rec}) - \mathbf{x}_{i,target}||_2^2. \tag{1.27}$$

Using the trained network $\mathrm{CNN}_{\Theta^\star}$ we can remove artifacts from a reconstruction

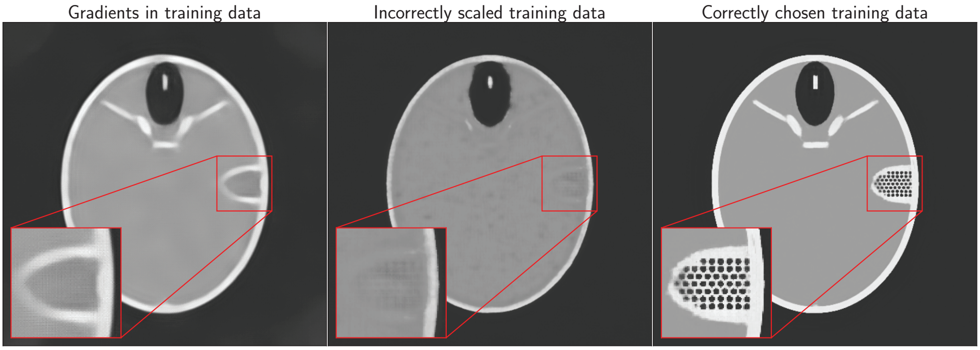


Figure 1.10: Examples of MSD networks trained on different training data applied to the same FBP reconstruction with high noise levels. All training data had the same noise levels. (Left) Training data contained gradients in the measured objects. (Middle) Training data contained piece-wise constant measured objects, but with a different scaling. (Right) Training data contained piece-wise constant objects with the correct scaling. Training a network took roughly 6 hours and computing a reconstruction took roughly a second.

$\mathbf{x}_{\text{rec}}$ similar to the input reconstructions in the training set:

$$\mathbf{x}_{\text{post-process}} = \text{CNN}_{\Theta^\star}(\mathbf{x}_{rec}). \tag{1.28}$$

One can vary the network architecture, loss function, training procedure, training data and corresponding hyper parameters and all these choices will lead to post-processing methods with different properties.

Examples of reconstructions using different types of training data are shown in **Figure 1.10**. We see that the networks are sensitive to changes in the training data.

1.2.4 The process of computing a reconstruction

In this section we will discuss the process of computing a reconstruction and the influence of reconstruction parameters on the ease of use of a reconstruction method.

From the introduction of the reconstruction methods we observe that all reconstruction methods have some kind of set of reconstruction parameters. For example: for the direct methods a filter has to be chosen, for iterative optimization schemes the number of iterations and the step-size parameter have to be set, the choice for regularization parameter is key for variational methods, and for

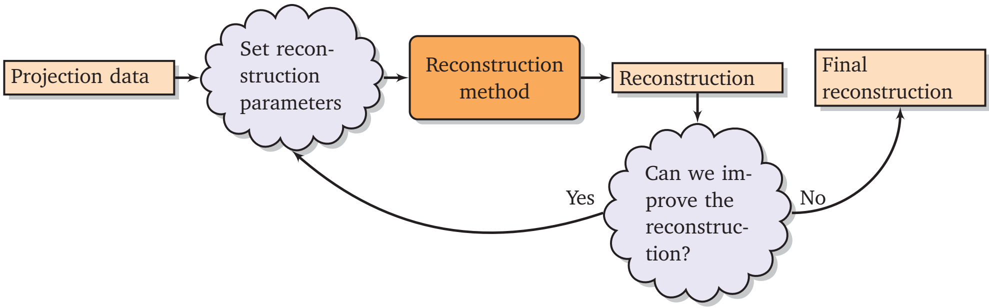


Figure 1.11: Schematic representation of the process of computing a reconstruction. The more time consuming it is to set the reconstruction parameters and compute the reconstruction, the more cumbersome it is to pick an almost-optimal set of reconstruction parameters.

machine learning methods the network architecture needs to be determined and the corresponding weights $\Theta^\star$ have to be learned for the chosen training data. Moreover, we have seen in the previous sections that the choice of these reconstruction parameters can strongly influence the accuracy of the reconstruction method. Following this reasoning we give a schematic representation of the process of computing a reconstruction in **Figure 1.11**. From this representation we can conclude that the harder it is to pick a suitable set of reconstruction parameters — *e.g.*, due to the number of possible choices or the time it takes to set the reconstruction parameters and compute a reconstruction — the more cumbersome the reconstruction process becomes.

If we now compare the reconstruction accuracy for different reconstruction methods — see **Figure 1.12** — we see that TV regularization and FBP + MSD reconstructions produce the most accurate results. However, recall from **Figure 1.9** and **Figure 1.10** that these methods are also the most time consuming methods and that the accuracy of these methods strongly depends on the choice of reconstruction parameters. Consequently, these methods are harder and more involved to use, especially for users with limited experience. This reiterates the importance of developing methods that are easy to use and perform similar to state-of-the-art methods or methods that improve the ease of use of an existing reconstruction method.

1.3 Outline of the thesis

This thesis is structured as follows: chapters 2 through 5 are based on self-contained research articles. Although these chapters have been edited slightly, they can be

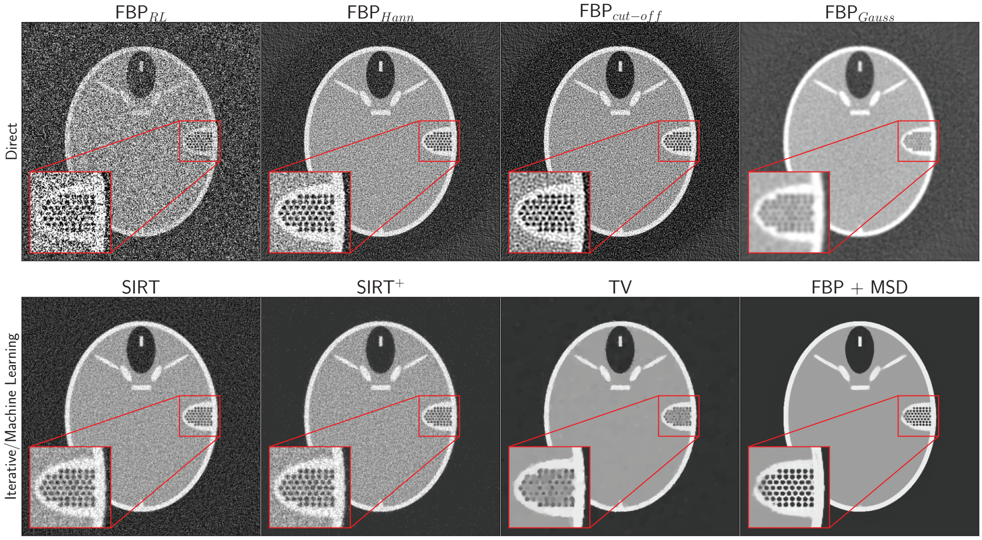


Figure 1.12: Comparison of different reconstruction methods applied to projection data with high noise levels. The reconstructions shown here are the 'optimal' reconstructions shown earlier in the section.

read independently. In **Chapter 2** we develop a framework for streamlining the process of picking the correct regularization parameter for variational methods. In **Chapter 3** we formulate an automatic algorithm for computing a data-dependent filter for the FDK algorithm. The NN-FBP algorithm is extended to the FDK algorithm in **Chapter 4** and combined with the Noise2Inverse training strategy in the RECAST3D framework in **Chapter 5**. A conclusion and outlook is given in **Chapter 6**, followed by the bibliography, a layman summary in Dutch and my Curriculum Vitae.