Automatic and efficient tomographic reconstruction algorithms

Lagerwerf, M.J.

Chapter 3

Automated FDK-Filter selection for cone-beam Computed Tomography

3.1 Introduction

Research environments in academia nowadays have cone-beam (micro-)CT systems that are used for imaging the 3D interior structure of highly diverse objects. These systems may be shared by many users, each studying their own type of objects and their own questions they would like to answer based on the interior structure. As an example, one can think of a natural history department where various fossils, meteor fragments, plant remains, insects, and a variety of other objects are all scanned using the same system. Similarly, industrial research labs use micro-CT to analyze their products ranging from detergents to dairy products and packaging materials, all using the same CT system. For each new scan, the settings of the scan (number of angles, dose, energy level, etc.) are chosen by the user, often based on how much time is available or the dose sensitivity of the sample.

The Feldkamp-Davis-Kress algorithm (FDK) is the most common reconstruction method used in laboratory circular cone-beam CT systems. It is well known that

This chapter is based on:

Automated FDK-Filter Selection for Cone-Beam CT in Research Environments. *MJ Lagerwerf, WJ Palenstijn, H Kohr, KJ Batenburg.* IEEE Transactions on Computational Imaging (Volume: 6), pp. 739–748, 2020.

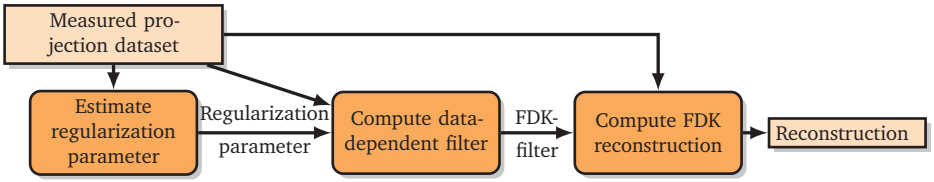


Figure 3.1: Schematic view of the proposed approach. Given a measured projection dataset with a certain geometrical setup, we estimate a regularization parameter and compute an FDK-filter that yields accurate results for common automated tasks (such as segmentation by global thresholding, porosity quantification).

optimizing the filter, also referred to as filter kernel, in the FDK algorithm to the characteristics of the scan (number of angles, dose, cone angle, etc.) can improve the accuracy of the FDK algorithm [Hsi+09; Rus17] (see Ch. 3.4.2 and 35.4.3.3, respectively). For high-throughput CT systems designed for a specific application (e.g. medical CT-scanners, dental CBCT scanners) the scanner comes with a set of proprietary pre-optimized filter [Com; Pla] that are chosen through a predefined protocol or by the user. In contrast, the broad variety in scans made in research scanners (many different objects with many different scan settings) require the user to manually select the parameters of the filter on a case-by-case basis, requiring specific expertise and time-consuming intervention from the user, or otherwise resulting in sub-optimal image quality.

Several studies have been made on how to compute such filters in an automated way, based on the geometrical parameters of the scanning process. In [GMD06; Nie+12] the authors exploit the tomosynthesis geometry to compute an acquisition-dependent filter. Alternatively, one can use the fact that the backprojection and filtering step are interchangeable in the FBP algorithm — for the parallel beam geometry — to optimize filters to approximate an iterative reconstruction method [BP12; Zen12; PB13], or to fit towards a specific scanner [Kun+07].

For the parallel beam geometry, a more general strategy for determining filters is proposed in [PB14], where a filter is computed that minimizes the residual error of the FBP reconstruction in the least squares sense. So far, this general concept has not been introduced in cone-beam tomography as the algorithm for computing the filter does not scale well to the 3D case of cone-beam tomography, where the full 3D volume must be taken into account.

In this chapter we present a computationally efficient and automated method to compute an FDK-filter for a given measured projection dataset that is optimal with respect to an objectively defined quality criterion based on the ℓ^2-norm of the difference between the measured projection data and the computed projections

of the reconstructed volume (**Figure 3.1**). Since this criterion is often referred to as the minimum residual, we will refer to our filters as Minimum Residual (MR) filters. We show that for a variety of objects, scan settings (number of angles and noise levels), and tasks (porosity quantification, threshold-based segmentation), the MR filters computed by our approach yield accurate results in terms of several different metrics (e.g. MAE, SSIM, MTF, which we will define later). In contrast, using the same manually tuned filter in all scenarios only yields accurate results for some of the cases, regardless of the particular choice of filter.

This chapter is structured as follows. In **Section 3.2** we introduce our method and describe how it allows for fully-automatic and efficient computation of the MR filter. In **Section 3.3** we describe how a set of experiments was carried out to investigate the behavior of our method under various scanning conditions, using both simulated and real experimental data. The results of these experiments are presented in **Section 3.4**. Conclusions are drawn in **Section 3.5**.

3.2 Method

3.2.1 Filter optimization problem

The 3D tomographic reconstruction problem can be modeled by a system of linear equations

$$W\mathbf{x} = \mathbf{y}, \tag{3.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is a vector containing the voxel gray values, $\mathbf{y} \in \mathbb{R}^m$ is a vector containing the measured projection data, and $W \in \mathbb{R}^{m \times n}$ is a discretized version of the *forward model*, i.e. the Radon transform for parallel beam tomography and the cone-beam transform for cone-beam tomography. In this chapter we focus exclusively on the circular cone-beam geometry, where the object rotates with respect to a point source and a planar detector, acquiring 2D cone-beam projections. For the sake of simplicity we assume that the volume consists of $n = N \times N \times N$ voxels and the detector consists of $2N \times N$ pixels. We denote the number of angles with N_a, so we have $m = N_a \times 2N \times N$.

The FDK algorithm [FDK84] is an extension of the well-known Filtered Backprojection algorithm that approximately solves (3.1) for the circular cone-beam geometry. For each projection angle, it applies a *reweighting* step, that corrects for some of the geometrical properties of the cone-beam transform, a *filtering* step, that filters the projections line-by-line by convolving the data with a *filter*, and a *backprojection* step that transfers the filtered projection into the image volume

domain. Using the notation of (3.1), the FDK algorithm is given by

$$\mathrm{FDK}(\mathbf{y}, \mathbf{h}) = W^T(\mathbf{h} * r(\mathbf{y}))_{1D}, \tag{3.2}$$

with W^T the transpose of W, known as the *backprojection operator*, $\mathbf{h} \in \mathbb{R}^{2N}$ a one-dimensional filter, r the reweighting operator, and $((\mathbf{h} * r(\mathbf{y}))_{1D})$ the discrete convolution between the data and the filter. While the standard Ram-Lak filter corresponds to the analytical derivation of the reconstruction problem, a variety of filters are used in practice for reaching a trade-off between artifacts, noise, sharpness of the reconstruction, and other application-specific image properties. The key contribution of this chapter is to propose a computationally efficient numerical algorithm for computing a filter for a specific combination of scanned object, geometrical parameters of the cone-beam acquisition, number of angles, and noise level. The aim is to devise an approach that provides decent quality results across a broad range of scenarios, such that the same automated approach can be used to compute FDK reconstructions, yielding high quality results in all cases.

A problem in automatically optimizing the FDK filter is that without access to a high quality reference image of the scanned object, defining reliable quality metrics is not straightforward. To solve this problem, we introduce a criterion that is not based directly on the reconstructed image, but instead on the *consistency* of the FDK-image with respect to the measured projection data $\mathbf{y}$, measured by simulating the projections of the FDK reconstruction and comparing these to the measured projections. Specifically, we select the filter as the minimizer of this cost function:

$$\mathbf{h}^\star = \operatorname*{argmin}_{\mathbf{h}} \|W(\mathrm{FDK}(\mathbf{y}, \mathbf{h})) - \mathbf{y}\|_2^2 + \lambda \|\mathbf{h}\|_2^2 \tag{3.3}$$

The first term corresponds to the *residual* of the FDK-reconstruction for a given filter $\mathbf{h}$, computed by applying the cone-beam transform to the FDK-reconstruction and comparing to the measured data $\mathbf{y}$. This term will be low if the filter results in an FDK-reconstruction that is consistent with the measured projections. Using a detector of size $2N \times N$ ensures that the convolution of the projected object with the filter is fully supported on the detector discretization domain and the influence of all filter coefficients on the reconstructed image is taken properly into account. The second term is a Tikhonov-type regularization term that penalizes filters with coefficients that have large absolute value. The regularization parameter λ determines the relative weight of this term. Note that this objective function only incorporates data specific information and general regularization, it does not include task or problem specific assumptions. In computing these filters we

are minimizing the residual, therefore we will refer to these computed filters as Minimum Residual (MR) filters.

We also evaluated more sophisticated regularizers, such as a Tikhonov-type term with the reconstructed image or the gradient of the reconstructed image, but we have not included them in this chapter as they lead to similar filters with similar performance as the proposed method.

We point out that the FDK algorithm is a *bilinear* operator with respect to the projection data $\mathbf{y}$ and the filter $\mathbf{h}$. This implies that for fixed projection data $\mathbf{y}$, the output FDK$(\mathbf{y}, \mathbf{h})$ of the FDK algorithm can be considered as a matrix-vector product $F_{\mathbf{y}}\mathbf{h}$. Consequently, the minimization problem in (3.3) is a linear least-squares problem for which the solution corresponds to the solution of the normal equations[1]:

$$(F_{\mathbf{y}}^T W^T W F_{\mathbf{y}} + \lambda I_{2N})\mathbf{h} = F_{\mathbf{y}}^T W^T \mathbf{y}, \tag{3.4}$$

with $I_{2N} \in \mathbb{R}^{2N \times 2N}$ the identity matrix. In the next subsection we will discuss how (3.4) can be solved accurately and efficiently.

3.2.2 Computational aspects

Note that the matrix inverse problem in (3.4) is small enough to solve directly once an explicit representation is available of the matrix $M = F_{\mathbf{y}}^T W^T W F_{\mathbf{y}}$. This matrix is square and its size equals the number of entries in the FDK-filter $\mathbf{h}$. However, computing the matrix M explicitly is not straightforward as it involves much larger matrices W and $F_{\mathbf{y}}$ and their transposes, which are too large to be represented explicitly. Instead, we compute the columns M_j of the matrix M individually, by evaluating the following expression $2N$ times:

$$M_j = M\mathbf{e}_j = (F_{\mathbf{y}}^T W^T W F_{\mathbf{y}})\mathbf{e}_j, \tag{3.5}$$

with $\mathbf{e}_j \in \mathbb{R}^{2N}$ a unit vector with all entries equal to zero except for the j^{th} element.

The computation for each element M_j involves a forward and backprojection, which can still impose a high computational load if the filter has many coefficients. Therefore, to reduce the number of filter coefficients, the filter is represented with respect to a small set of basis functions as $\mathbf{h} = E\mathbf{h}_e$ on an exponentially binned grid similar to [PB14]. As a result, we have $N_e \approx \log(N)$, with N_e the number of elements of $\mathbf{h}_e$. Details about this approximation are discussed in **Appendix 3.6.1**.

The algorithm to compute a MR filter is summarized in **Algorithm 2**. Lastly, we observe that the computational effort of computing a MR filter is roughly $2N_e$ forward and backward projections and lies mainly in the computation of the matrix M.

[1] Consider the first order optimality conditions of (3.3) and rearrange the terms to get (3.4).

Algorithm 2 Computing a MR filter

1: **for** $j = \{0, 1, 2, .., N_e - 1\}$ **do**
2: $M_j = \left(E^T F_{\mathbf{y}}^T W^T W F_{\mathbf{y}} E\right) \mathbf{e}_j$
3: Compute: $\mathbf{h}_e^\star = \left(M + \lambda I_{N_e}\right)^{-1} E^T F_{\mathbf{y}}^T W^T \mathbf{y}$

3.2.3 Regularization parameter

The final component of our automated approach for computing the FDK-filter is to determine a suitable value for the regularization parameter λ. Finding an optimal value for this parameter is in general not possible as it requires knowledge of the ground truth as well as detailed modelling of the application-specific quality criteria. We therefore aim for a computationally efficient heuristic that yields decent results across a broad range of imaging scenarios.

Our strategy involves three consecutive components:

- **Computing a low-noise reference reconstruction at strongly reduced spatial resolution**. We compute a low-resolution reconstruction, where the volume as well as the projection data are down-sized by a factor of 4 in each dimension (so, 4×4×4 for the volume and 4×4 for each projection). In this small reconstruction problem the signal-to-noise ratio is much higher and the number of angles is much larger relative to the size of the volume compared to the full problem. To compute the reference reconstruction, 200 iterations of the iterative SIRT algorithm are used, with a nonnegativity constraint applied in each iteration.

- **Optimizing the regularization parameter for the low-resolution problem using the noise characteristics of the full problem**. We subsample the high resolution data by taking every 4^{th} pixel in the detector width and height and compute the matrix M_{LR} with this subsampled low resolution data and compute reconstructions $\mathbf{x}_{LR}^\lambda$ for several values of λ. Note that by subsampling we do not reduce the noise levels, making the noise characteristics of the low resolution problem similar to the original high resolution problem. We choose the regularization parameter for which the difference with the reference reconstruction is minimal in the ℓ^1-norm. This heuristic choice of norm works well for our experiments. Other norms could also be used.

  Regularization parameters can vary strongly per problem. Therefore, we scale the parameter to account for the influence of the operators W, $F_{\mathbf{y}}$ and

E:

$$\lambda = \tilde{\lambda} \left\| W_{LR} F_{\mathbf{y},LR} E_{LR} \right\| = \tilde{\lambda} \sqrt{\|M_{LR}\|}. \tag{3.6}$$

Moreover, we consider a two step process for optimizing the parameter. First, we optimize over a broad and coarse grid, more specifically a logarithmically scaled grid spanning from 10^{-6} to 10 with 8 grid points. Second, we optimize over a fine grid around the optimal parameter from the first grid.

- **Scaling the regularization parameter to the full-resolution case**. The regularization parameter λ computed for the low-resolution problem cannot be used directly in the high-resolution problem. To account for the scaling differences between the problems, we apply the following conversion formula (*cf.* (3.6)):

$$\lambda_{HR} = \lambda_{LR} \frac{\left\| W F_{\mathbf{y}} E \right\|}{\left\| W_{LR} F_{\mathbf{y}_{LR}} E_{LR} \right\|} = \lambda_{LR} \frac{\sqrt{\|M\|}}{\sqrt{\|M_{LR}\|}}. \tag{3.7}$$

Once the regularization parameter λ and corresponding filter $\mathbf{h}$ have been computed, this filter can be used in the FDK algorithm to compute a reconstruction of the high-resolution data.

3.3 Experiments

We performed a series of experiments to assess the properties of our proposed MR filters. The goal of our experiments is twofold; (1) Compare the accuracy of the FDK results for MR filters to manually selected filters for the experiments, (2) Investigate the capability of MR filters to automatically adapt to a variety of objects, scan settings and tasks.

To achieve this we consider the following four scenarios:

- **Reconstruction problems with common data deficiencies.** We vary the noise levels, number of projection angles in the different input data and cone angle and compare the results from the computed filters to the manually tuned filters.

- **Task specific reconstruction problems.** We compute reconstructions of a *foam phantom* similar to the one used in [PBS18] and use these to do segmentations and compute the pore size distribution of this phantom. For the segmentations we use Otsu's global thresholding method [Ots79]. Details about the computation of the pore size distribution are given in **Appendix 3.6.4**.

- **Filter comparison and analysis.** We compare the computed filters and their respective *Modulation Transfer Functions* (MTF) to the manually selected filters. Details about the implementation of the MTF are given in **Appendix 3.6.5**.

- **Examples with experimental data.** We show results for two experimental datasets using the computed filters and compare the reconstructions to manually selected filters and the gold standard reconstruction (see **Section 3.3.1**).

The results are compared to several manually selected filters and analytic FDK filters. As manually selected filters we consider the family of filters which combine a low-pass filter with an analytic filter, *e.g.*, the Shepp-Logan filter and a Gaussian filter. The set of selected filters are chosen such that they are close to the optimal quality metrics for each of the conducted experiments. We point out that a manually selected filter from this set might actually yield sub-optimal results for experiments other than it was selected for, which is exactly why an automated and deterministic approach for computing filters can be beneficial.

Details on how the filters are combined, the definitions of the low-pass filters and how these filters are selected are given in **Appendix 3.6.2**.

3.3.1 Data

Simulated data

In **Figure 3.2** we show the FORBILD head phantom [LB], which is used in the simulated data experiments. Note that with the chosen scanning conditions we focus on the high contrast details and we do not expect to resolve the low contrast objects in this head phantom.

For our simulated data experiments we take $N = 1024$, which means that a reconstruction and the data are respectively defined on a 1024^3 equidistant voxel grid and a 2048×1024 equidistant detector grid per projection angle. To limit the influence of the inverse crime, we generate[2] a phantom with $N = 1536$, and forward project this phantom to a data space with detector size 3072×1536. Then interpolate the data per projection angle to a 2048×1024 detector grid and use this as input data. We set the source radius to 10 times the physical size of the phantom, resulting in a cone angle of 5.7 degrees. Poisson noise is applied before linearizing the data, *i.e.*,

$$\mathbf{y} = -\log\left(\frac{I_{\text{noise}}}{I_0}\right), \qquad I_{\text{noise}} \sim \text{Pois}(I), \tag{3.8}$$

[2]We also generated phantoms with higher resolutions, but did not observe noticeable differences. Hence we chose for a sampling factor of 1.5, to limit computational and memory constraints.

with I the noise-free photon count and I_0 the emitted photon count. Higher I_0 implies a higher dose and therefore less noise in the data.

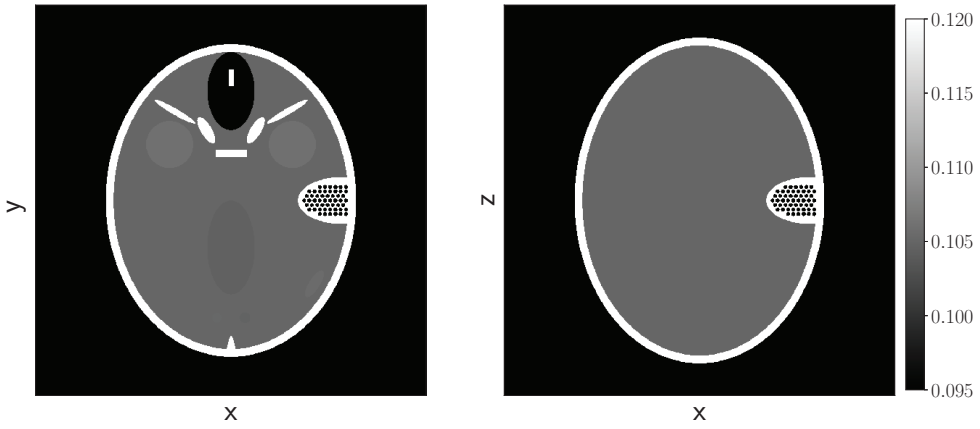


Figure 3.2: Two-dimensional slices $z = 0$ (Left) and $y = 0$ (Right) of the FORBILD head phantom [LB]. Note that the gray value scaling is different from the reconstructions further in the chapter. This is done to highlight the low contrast objects in the center, top, and bottom of the phantom. The phantom is continuously defined and sampled on a chosen grid.

Experimental data

We acquired an experimental dataset of a pomegranate using the custom-built and highly flexible FleX-ray CT scanner, developed by XRE NV and located at CWI. This scanner has a flat panel detector with 1943×1535 pixels and a physical size of 145.34×114.82 mm. The datasets contain 500 equidistantly spaced projections over a full circle. The distance from the center of rotation to the detector was set to 109 mm and the source radius to 590 mm. The scans were performed with a tube voltage of 70 kV. The high-dose scan was collected with a tube power of 45 W and an exposure time of 500 ms. The low-dose scan was collected with a tube power of 20 W and an exposure time of 100 ms. These datasets are available at Zenodo [CLB18].

In **Figure 3.3** the *gold standard* reconstruction of the experimental data is shown, obtained by computing a SIRT reconstruction with 300 iterations of the high-dose dataset with 500 equidistant projection angles, where we set the voxels in the background below a certain threshold equal to zero.

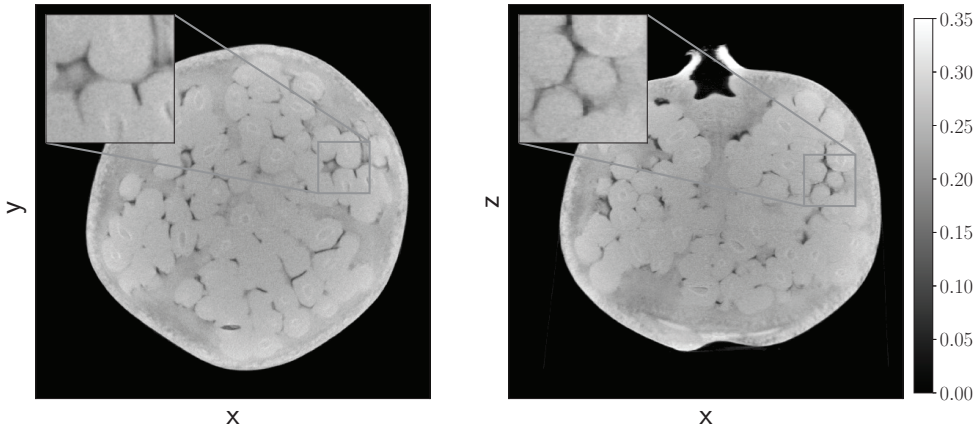


Figure 3.3: The $z = 0$ (Left) and $y = z$ (Right) slice of the gold standard reconstruction of the pomegranate dataset. The projection data is acquired using the FleX-ray scanner located at the CWI.

3.3.2 Quantitative measures

To quantify the accuracy of the reconstructions we consider two measures, the mean absolute error (MAE) and the structural similarity index (SSIM). The MAE and the SSIM compare the reconstructed image $\mathbf{x}_r$ to the phantom image $\mathbf{x}_p$. The MAE is defined as

$$\mathrm{MAE}(\mathbf{x}_r, \mathbf{x}_p) = \frac{\left\|\mathbf{x}_r - \mathbf{x}_p\right\|_1}{\left\|\mathbf{x}_p\right\|_1}, \tag{3.9}$$

The SSIM [Wan+04] is implemented based on the scikit-image 0.13.1 [Wal+14] package, where all the constants are set to default and the filter is uniform with a width of 19 pixels.

For experimental data there is no ground truth image available. Therefore, we will use the high quality gold standard reconstruction as reference image $\mathbf{x}_p$.

Lastly, we are only interested in how the methods perform on the object itself and not on the background. Therefore, we only consider the reconstructed object and roughly $0.2N$ pixels away from the object. Here the position of the object is determined in the ground truth or gold standard reconstruction.

3.3.3 Implementation

All the methods are implemented using Python 3.6.2, Numpy 1.12.1 [WCV11], ODL [AKÖ17] and PyFFTW 0.10.4 [FJ05], and the forward- and backprojection

are implemented on the GPU using the ASTRA-toolbox [Van+16], which provides a collection of high-performance building blocks for tomography algorithm development. Here, for performance reasons, the forward projection is not the exact adjoint of the backward projection and vice versa. Our implementation of this method is available on Github [Laga].

3.4 Results and discussion

3.4.1 Common data deficiencies

Sparse view and noisy data

Figure 3.4a and **Figure 3.4b** show the MAE and SSIM for reconstructions with a varying number of projection angles and varying noise levels, respectively. We observe that the MR filters are close to the optimal manually selected filter (see **Appendix 3.6.2** for details about manually tuned filters) for all considered cases with respect to the quantitative measures. Moreover, we observe that the manually selected filters are only optimal for a certain range of cases. This is illustrated in **Figure 3.5** where reconstructions of noisy data are shown with several filters.

Varying cone angles

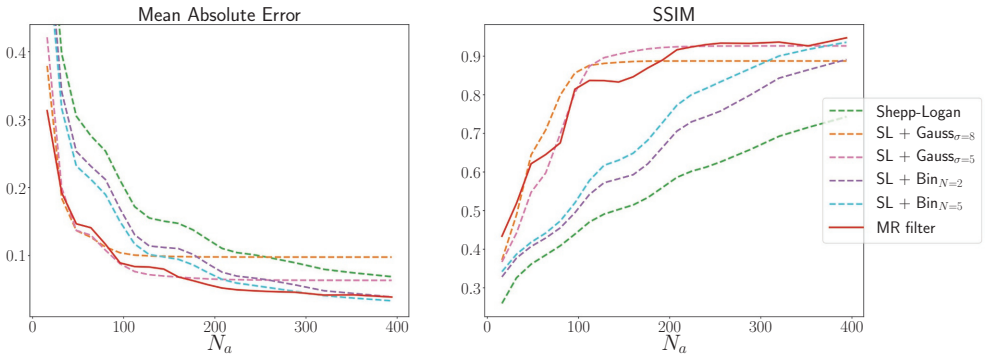
So far we have considered a relatively small cone angle of 5.7 degrees. In this section we show the effect of varying the cone angle. **Figure 3.6** shows the MAE and SSIM for a range of cone angles.

We observe that all the reconstruction methods react similarly to the change in cone angle with unchanged relative performance. **Figure 3.7** illustrates the effect of a large cone angle with strong artifacts at the top and bottom of the reconstruction. The effect of these artifacts are also reflected in the quantitative measures.

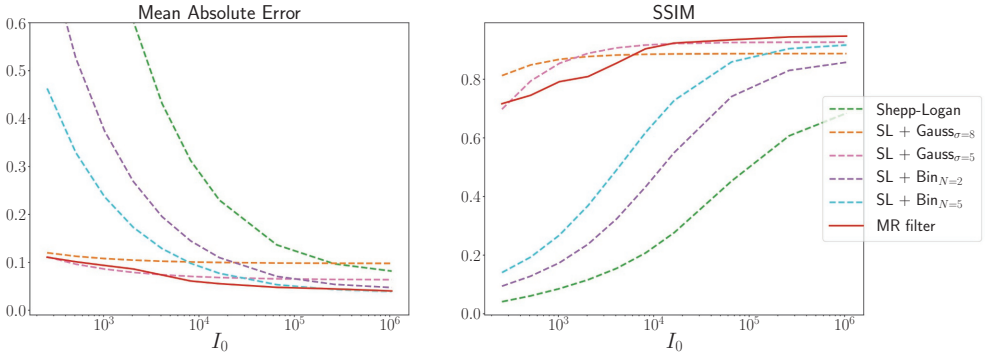
3.4.2 Task based problems

In this section we test the performance of MR filters for specific tasks, namely segmentation and porosity computation. **Figure 3.8** shows the percentage of misqualified voxels for an Otsu global thresholding segmentation [Ots79] for several filters and varying noise levels.

We observe that the less smoothing filters lead to a lower percentage of misqualified voxels. Looking at **Figure 3.9** we observe that the MR filter still leads to a less noisy reconstruction, but the smaller pores are lost in the segmentation.



(a) Varying number of projections angles.



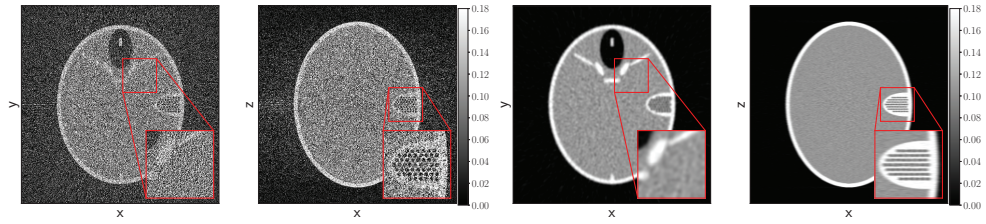
(b) Varying emitted photon counts I_0.

Figure 3.4: Comparing quantitative measures of FDK reconstructions with varying filters. The data is generated from the FORBILD head phantom.

Looking at **Figure 3.9** we observe that reconstructions with the MR filter contain less noise than the reconstructions with SL + Bin$_{N=5}$ filter, which is in line with the experiments in **Section 3.4.1**. Additionally, we see that, although the SL + Bin$_{N=2}$ has a lower percentage of misqualified voxels in this case, it still contains noise in the object and false positives in the background.

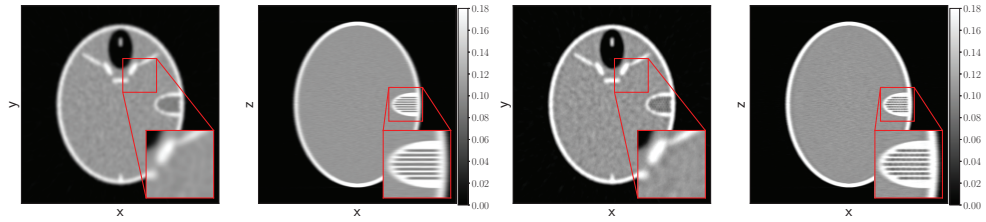
From these segmentations we can also count the pores inside the phantom. A pore is defined as an open space in the segmented volume and in **Figure 3.10** we show the pore size distributions for several noise levels. First of all, we note that the segmentations with the SL + Bin$_{N=2}$ filter for $I_0 = 256$ contain too much noise to compute a sensible pore distribution. Second, we observe that the MR filter underestimates the number of pores, whereas the filters with binomial smoothing tend to overestimate the number of pores.

Considering these observations we can conclude that, although MR filters do



(a) FDK reconstruction with Shepp-Logan filter.

(b) FDK reconstruction with a SL + Gauss$_{\sigma=5}$ filter.



(c) FDK reconstruction with a SL + Gauss$_{\sigma=8}$ filter.

(d) FDK reconstruction with MR filter.

Figure 3.5: Reconstructions from the FORBILD phantom; the dataset has 360 equidistant projection angles and emitted photon count $I_0 = 256$.

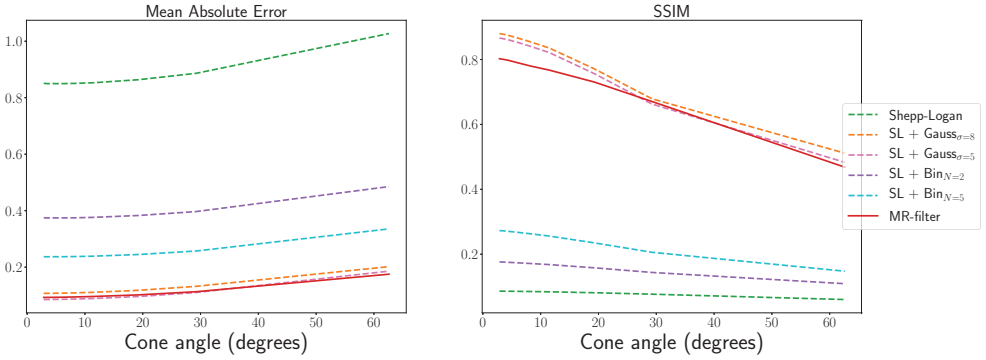


Figure 3.6: Reconstruction quality measures computed from the FORBILD phantoms with 360 equidistant projection angles, emitted photon count $I_0 = 1024$, and varying cone angles.

not lead to the best segmentations or pore size distributions, they still perform robustly in all the considered cases.

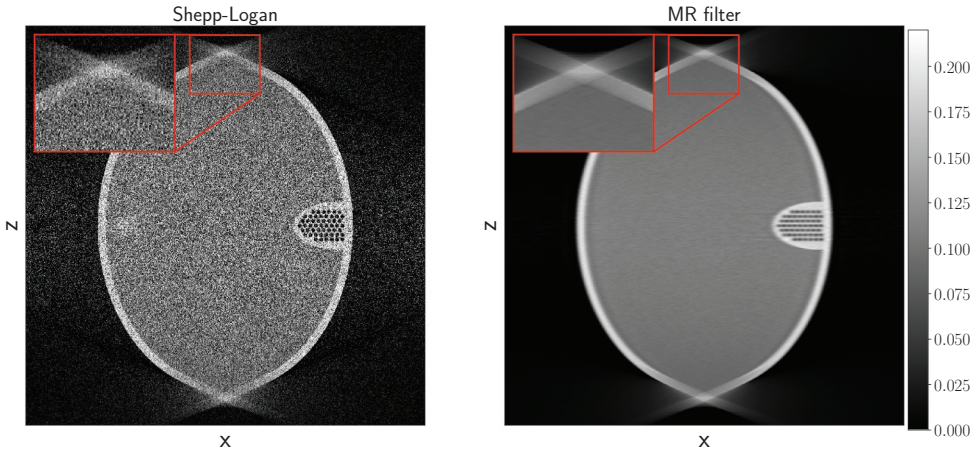


Figure 3.7: Reconstructions using Shepp-Logan filter (Left) and MR filter (Right) computed on the FORBILD phantom with 360 equidistant projection angles, cone angle of 62.6 degrees and an emitted photon count $I_0 = 1024$.

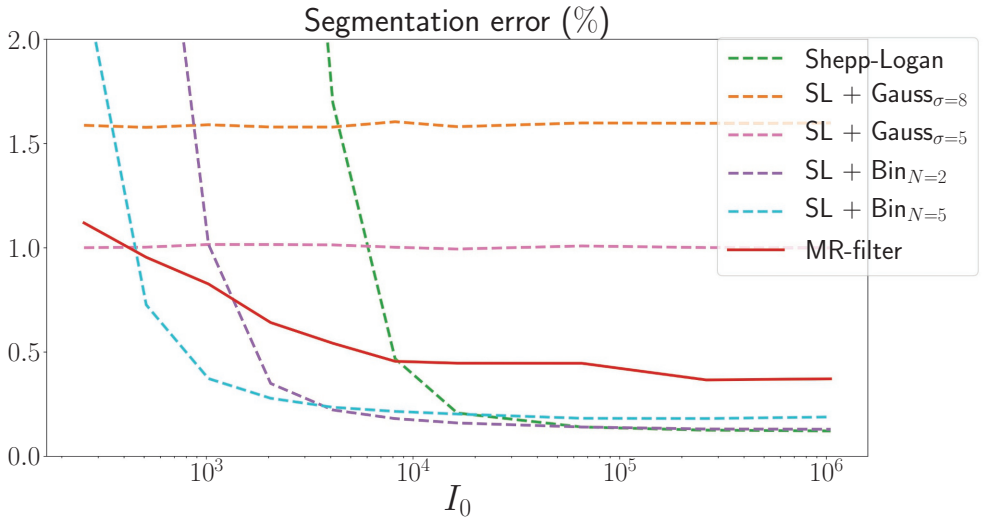


Figure 3.8: Segmentation errors for various filters. The projection data is simulated with varying emitted photon counts I_0 from a foam phantom.

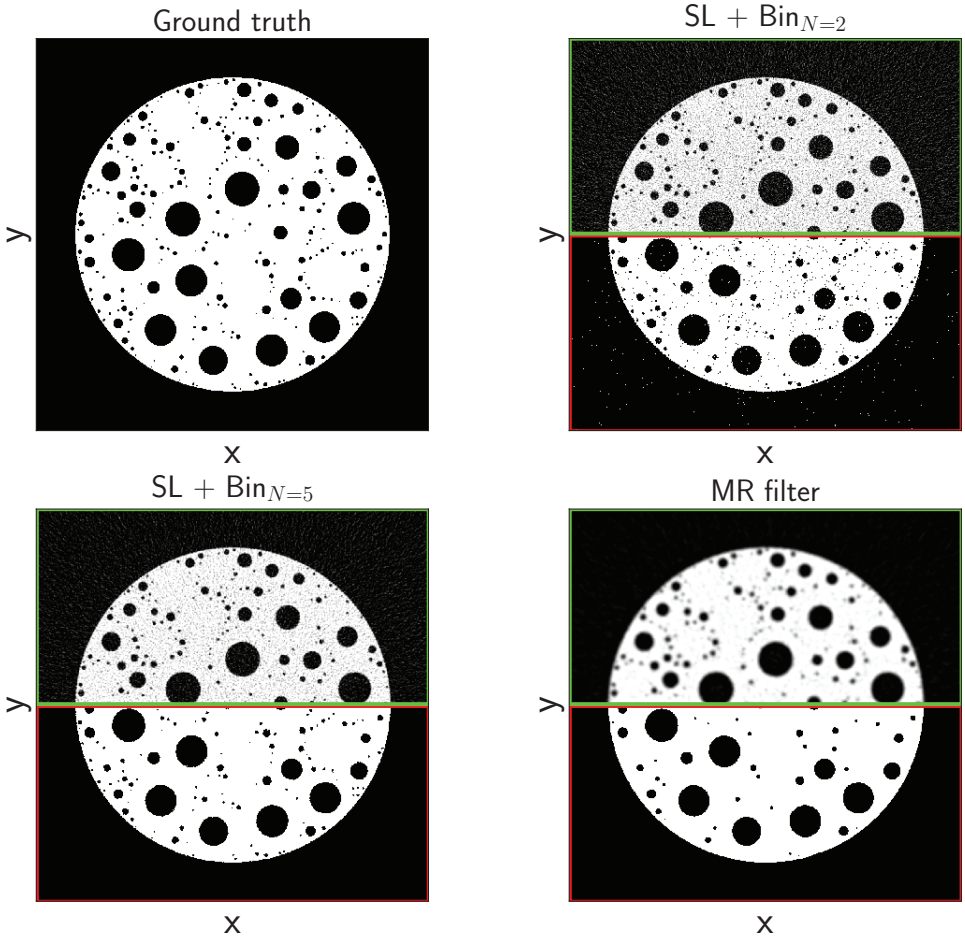


Figure 3.9: Reconstructions (top half, green border) and segmentations (bottom half, red border) of projection data from a foam phantom with emitted photon count $I_0 = 2048$ and 360 projection angles.

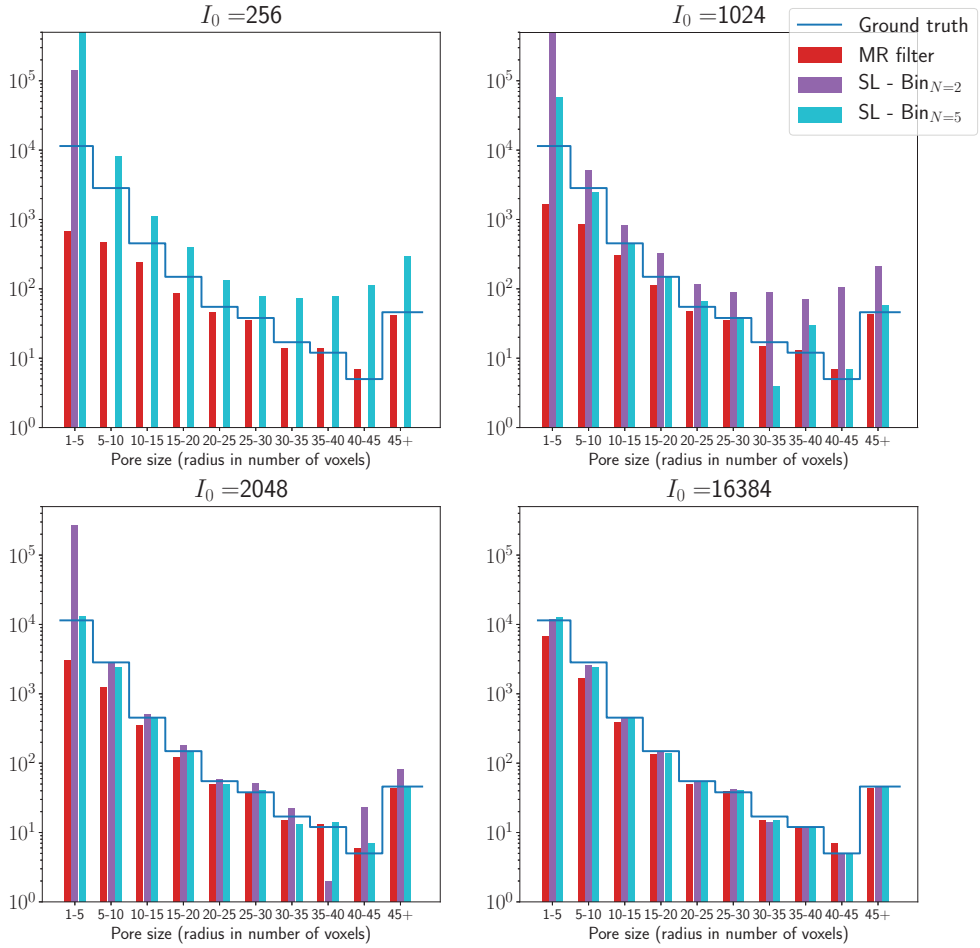


Figure 3.10: Pore size distributions for three different filters and the ground truth distribution. The size of a pore is defined as the radius of the pore, which is expressed in a number of voxels. Furthermore, the pores are grouped into bins with similar radii (5 voxel intervals). The projection data is simulated with varying emitted photon counts from the foam phantom.

3.4.3 Filter analysis

In this section we analyse the MR filters. In **Figure 3.11** we show the Fourier representation of several types of filters, namely: analytic, manually tuned, and MR filters. We show MR filters from four different cases. Comparing the filters we observe that the shape of the MR filters is different from the manually selected filters, *i.e.*, they have a relatively lower amplitude and a longer tail. From the MTFs, which are shown in **Figure 3.12**, we also observe a slight difference between the three types of filters. The MTFs of the analytic and manually selected filters all have a similar gradual drop in frequencies, whereas the MTFs of the MR filters have a steeper drop and a longer tail compared to the SL + Gauss filters, which were the filters that led to the most comparable results for the (NOI, 1) and (NOI, 2) scenarios.

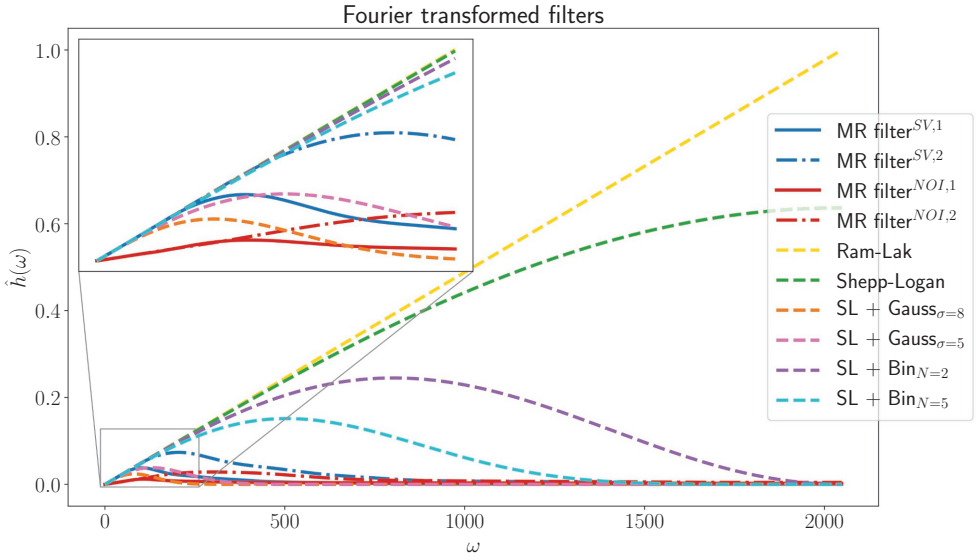


Figure 3.11: Fourier representations of various filters. Here the MR filters denoted with (SV, 1) and (SV, 2), are computed on the FORBILD head phantom with 96 equidistant and 192 projection angles, respectively. The MR filters denoted with (NOI, 1) and (NOI, 2) are computed on the FORBILD phantom with 360 equidistant projection angles and emitted photon count $I_0 = 256$ and $I_0 = 16384$, respectively.

Lastly, we consider the image bias introduced by the MR filters. The MR filters use the FDK algorithm to reconstruct the data. Therefore, their reconstructions will suffer similar limitations as the reconstructions with standard filters. In **Figure 3.13** we show the cross section along the x-axis of averaged reconstructions computed

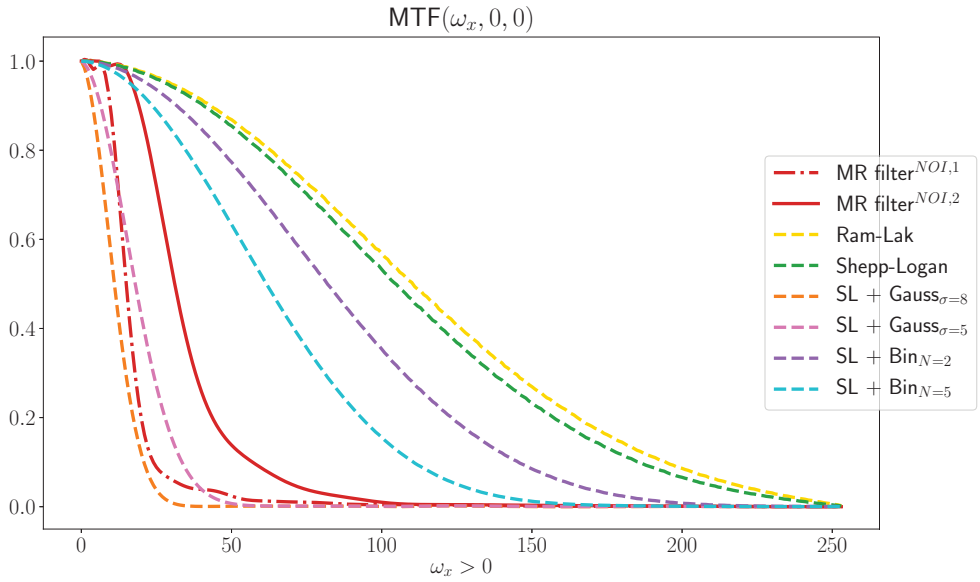


Figure 3.12: Modulation Transfer Functions of several filters in the positive ω_x direction. The MR filters are computed on datasets with 360 equidistant projection angles of the FORBILD phantom and emitted photon counts $I_0 = 256$ (NOI, 1) and $I_0 = 16384$ (NOI, 2).

over 500 random noise instantiations and their standard deviations of 3 phantoms: a cylinder with constant density, a cylinder with a ramp in its density and the FORBILD phantom. We clearly see the bias-variance trade-off between the more smoothing filters and the Ram-Lak filter. Additionally, we see that the MR filters adapt to the data and even try to fit the edges, at the cost of overshooting around these edges.

3.4.4 Experimental data

In **Table 3.1** the quantitative measures for the experimental data with respect to the gold standard reconstruction (recall **Section 3.4.4** and **Figure 3.3**). First we observe that the MR filters have a lower MAE than the other filters. In **Figure 3.14** we see that the intensity of the MR filter reconstruction is closer to the gold standard reconstruction, which explains the difference in MAE. This difference is less prominent in the SSIM, because the SSIM is designed to be less influenced by scaling differences. Lastly, we observe that the reconstruction with the MR filter for the low-dose dataset still contains noise, which explains the relatively low SSIM.

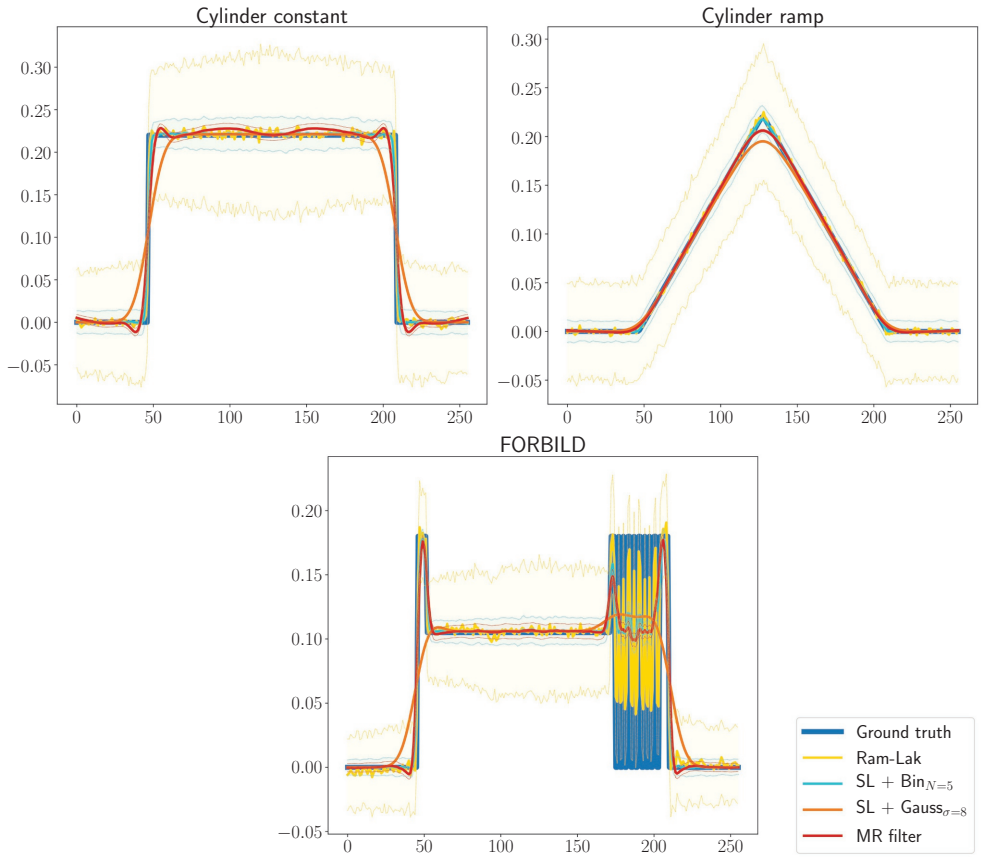
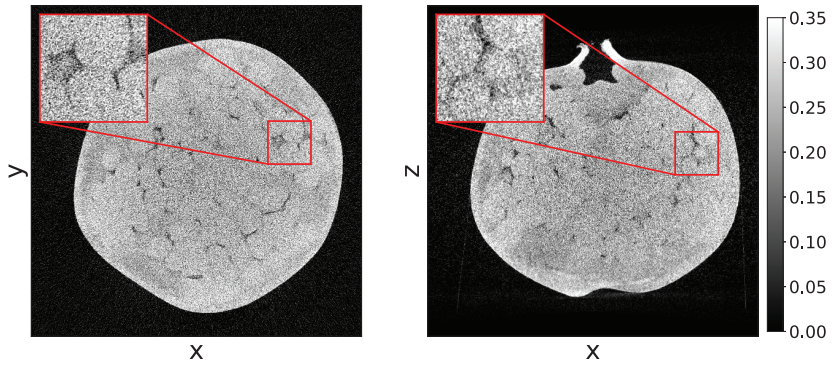
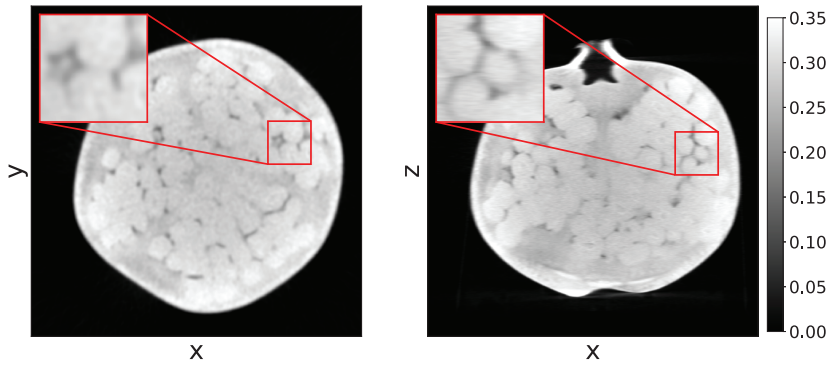


Figure 3.13: Average reconstructions and standard deviation of the x-axis for a (Left) constant cylinder phantom (Middle) ramp cylinder phantom (Right) FORBILD phantom. The average and standard deviation are computed over 500 reconstructions with 256^3 voxels of input data with different initialisations of emitted photon count $I_0 = 256$ and 360 projection angles.

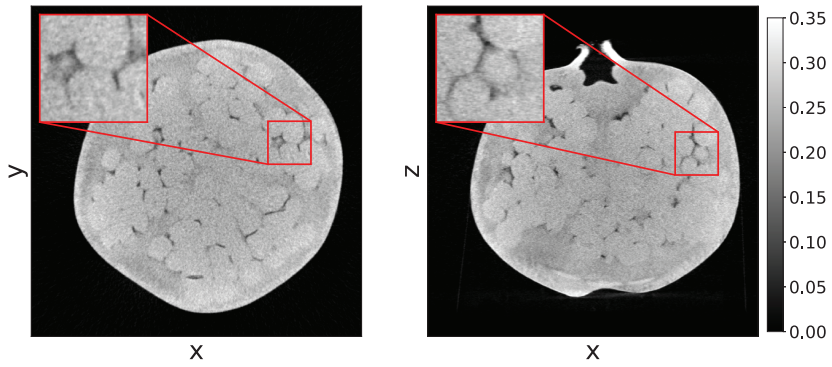
This indicates that the estimation for the regularization parameter is relatively low, which is most likely due to noise in the low resolution high quality reconstruction used to determine this parameter.



(a) FDK reconstruction with Shepp-Logan filter.



(b) FDK reconstruction with SL - Gauss$_{\sigma=8}$ filter.



(c) FDK reconstruction with MR filter.

Figure 3.14: Reconstructions of the low-dose Pomegranate dataset with 500 equidistant projection angles.

Table 3.1: Reconstruction quality measures from the high- and low-dose Pomegranate1 dataset with 64 and 500 equidistant projection angles (Left & Right), respectively.

High dose, 64 projection angles		
Method	MAE	SSIM
Shepp-Logan	0.2290	0.1789
SL + $\text{Bin}_{N=2}$	0.2839	0.1948
SL + $\text{Bin}_{N=5}$	0.2620	0.2350
SL + $\text{Gauss}_{\sigma=5}$	0.2112	0.5365
SL + $\text{Gauss}_{\sigma=8}$	0.2037	0.6304
MR filter	0.0711	0.6500

Low dose, 500 projection angles		
Method	MAE	SSIM
Shepp-Logan	0.2426	0.1799
SL + $\text{Bin}_{N=2}$	0.2794	0.2147
SL + $\text{Bin}_{N=5}$	0.2513	0.2745
SL + $\text{Gauss}_{\sigma=5}$	0.2144	0.6983
SL + $\text{Gauss}_{\sigma=8}$	0.2152	0.7294
MR filter	0.0636	0.5934

3.5 Conclusions and outlook

We have proposed a computationally efficient and automated method to compute a FDK-filter for a given imaging scenario (scanned object, number of angles, dose) that is optimal with respect to an objectively defined quality criterion. For cone-beam CT scanners used in research environments, where many different objects are scanned and parameters are often varied, our method can be used to automatically determine a filter that yields accurate results across a range of scanning conditions and tasks to be performed.

The experimental results demonstrate that for a variety of objects, scan settings (number of angles and noise levels), and tasks (porosity quantification, threshold-based segmentation), the MR filters computed by our approach are not task or problem specific and yield accurate results in terms of several different metrics that are comparable to manually selected filters.

Although the computational cost of computing an MR-filter is substantially lower than running an iterative reconstruction algorithm, it is much higher than the computational cost of FDK with a fixed filter. When carrying out batches of scans for similar objects, one can reuse an MR-filter computed for a particular object to further bring down the computational cost.

3.6 Appendices

3.6.1 Filter approximation

We want to design an expansion operator E to reduce the dimension needed to describe a filter. **Figure 3.15** shows five examples of analytic FDK filters as

presented in [Nat01; Buz08]. From the figure we see that the filters are symmetric and have most information on a fine grid around the origin.

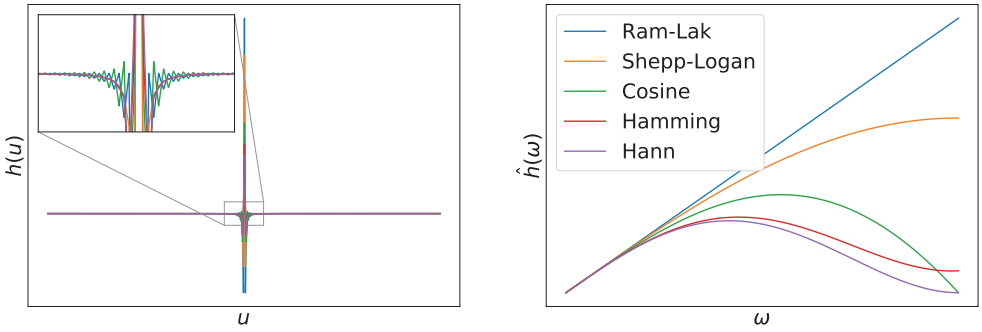


Figure 3.15: Standard filters in the spatial (left) and Fourier (right) domain.

To emulate this behavior, we introduce a re-sampling scheme as used in [BK06; PB13; PB14]. We do this by defining the discretised filter space $\mathbb{D}_u$ to be a uniform grid with $2N_u$ points on the interval $[-D_u, D_u]$ and grid points u_j. Now we re-sample these points into bins β_i with length:

$$d_i = \begin{cases} \frac{\Delta_u}{2}, & \text{for } i = 0, \\ \Delta_u, & \text{for } 0 < i < b, \\ 2^{b-i}\Delta_u, & \text{for } i \geq b, \end{cases} \tag{3.10}$$

with $\Delta_u = \frac{D_u}{N_u}$, the length of a single detector pixel. Here the indices i denote the place of the bin with respect to the central bin, *i.e.*, $\beta_0 = [0, d_0]$ and $\beta_1 = (d_0, d_0 + d_1]$, etc. This binning strategy results in a grid with uniform spacing around the origin and, depending on the binning parameter b, an exponentially coarsening grid for the outer regions. Instead of piece-wise constant basis functions as were used in [BK06; PB13; PB14], we will use piece-wise linear basis functions. This is because we observed that filters computed with these basis functions behave regularly in Fourier space. We define these functions as such:

$$\phi^{\text{lin}}_{i,u} = \begin{cases} \frac{s_i - u_j}{d_{i-1}}, & \text{for } \left|u_j\right| \in \beta_{i-1}, \\ \frac{u_j - s_i}{d_i}, & \text{for } \left|u_j\right| \in \beta_i, \end{cases} \tag{3.11}$$

where s_i is the boundary value of the i^{th} bin, such that $\beta_i = (s_i, s_{i+1}]$.

Spatial representations of filters created with piece-wise constant and linear basis functions are shown in **Figure 3.16**. we use the binning parameter $b = 2$, which strongly limits the computational effort of computing a filter. In experiments

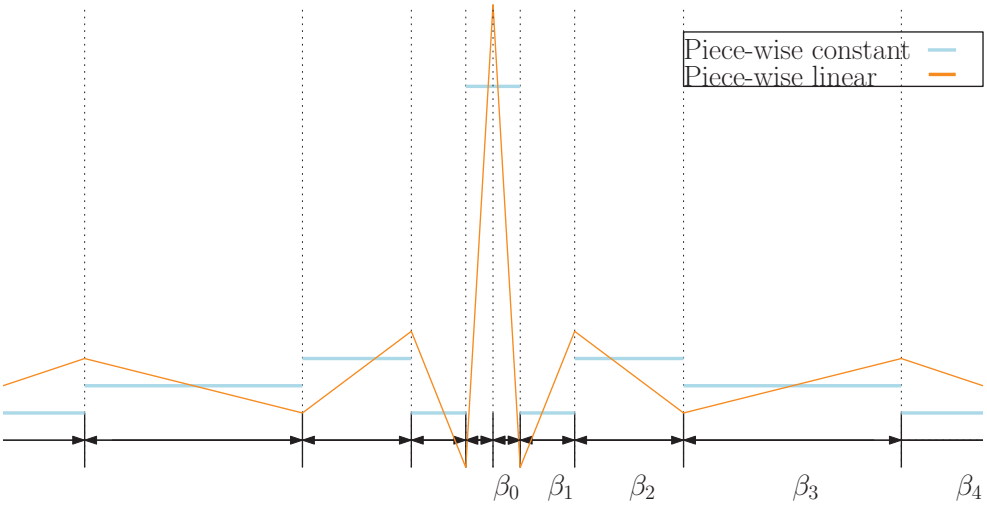


Figure 3.16: Examples of exponentially binned filters with piece-wise constant and linear basis functions. Binning parameter $b = 2$.

we observed that selecting a binning parameter greater than 2 does not improve the quality of the reconstructions substantially.

3.6.2 Filters selected for comparison

Manually selected filters for the FDK algorithm are often analytic filters combined with low-pass filters. These filters are typically manually adapted to the scan conditions. We selected several of such filters as reference methods for our results. We consider two types of low-pass filters.

Binomional filters are defined as:

$$\text{Bin}_N = \underbrace{([1\ 1] \otimes [1\ 1] \otimes \cdots \otimes [1\ 1])}_{N \text{ times}} / 2^{N+1}, \qquad \text{with } N \in \mathbb{N}_{>0}, \tag{3.12}$$

and N the order of the binomial filter.

Gaussian filters are defined as:

$$\text{Gauss}_{\sigma,j} = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(j-N_u/2)^2}{2\sigma^2}}, \qquad \text{with } \sigma > 0, \tag{3.13}$$

with $\text{Gauss}_{\sigma,j}$ the j^{th} element of $\text{Gauss}_\sigma \in \mathbb{R}^{N_u}$ and σ the standard deviation of the Gaussian in pixels.

We combine the filters by convolving the low-pass and analytic filters in the spatial domain and cutting them off. These filters are specified by first referencing

the analytic filter, then the low-pass filter and lastly the value of the parameter related to this low-pass filter.

For comparing the results of our proposed MR filter with a set of low pass filters combined with analytic filters (as defined above), we carried out a search over the parameter space of these filters (specifically $N \in \{1, ..., 8\}$ and $\sigma \in \{1, ..., 10\}$) and selected four filter settings such that for each of the experiments reported in the chapter, at least one of the selected filters has quality metrics close to the optimum across the parameter space (with respect to MAE, SSIM, and task specific metrics). The selected filters are the SL + $\text{Gauss}_{\sigma=5,8}$ filters and the SL + $\text{Bin}_{N=2,5}$ filters.

3.6.3 Methods used in experiments

3.6.4 Pore size distribution

We derive the pore size distribution from the cumulative pore size distribution of a segmentation. The cumulative pore distribution is computed through inverting the segmentation, which leads to pores being particles, and step-wise erosion of these resulting particles. By controlling the number of voxels that we erode, we know how many pores there are with a certain radius (this radius is expressed in the number of voxels).

3.6.5 Modulation transfer function

The *modulation transfer function* (MTF) of a reconstruction method is defined as the magnitude of the Fourier transform of its *point spread function* (PSF). However, due to aliasing, computing the MTF directly from the PSF is unstable. Therefore, instead of measuring the PSF directly, we measure the *edge spread function* (ESF) and compute its derivative, which coincides with the PSF. Following the ASTM standard [AST13], we consider a homogeneous cylinder, with its axis on the z-axis, as the measured object. Since the FDK algorithm and the object are radially symmetric, every ray from the center of the cylinder to the edge of the reconstruction domain is a realization of the ESF. Defining ϕ to be the angle that this ray makes with the x-axis we can compute the average ESF over all ϕ, which we denote by $\mathbb{E}_\phi$ (we limit ourselves to the z=0 plane). Next we compute the gradient of this average ESF and use it to compute the MTF acting in the z=0 plane.

In mathematical terms, the MTF in the z=0 plane, or w.l.o.g. the x-direction, due to the radial symmetry, related to a filter $\mathbf{h}$ in the FDK algorithm is defined as

follows:

$$\mathrm{MTF}_{\mathbf{h}}(\omega_x) = \left|\mathscr{F}_{1D}\left\{\mathrm{PSF}_{\mathbf{h}}\right\}(\omega_x, 0, 0)\right|, \tag{3.14}$$

$$= \left|\mathscr{F}_{1D}\left\{\tfrac{d}{dx}\mathrm{ESF}_{\mathbf{h}}\right\}(\omega_x, 0, 0)\right|, \tag{3.15}$$

$$\approx \left|\mathscr{F}_{1D}\left\{\tfrac{d}{dx}\left(\mathbb{E}_{\phi}\left[F(\mathbf{y}_{\mathscr{C}}, \mathbf{h})_{z=0}\right]\right)\right\}(\omega_x, 0, 0)\right| \tag{3.16}$$

where $\mathbf{y}_{\mathscr{C}}$ is the cone-beam projection data of the aforementioned cylinder.