Ultra-fast reconstruction using Fourier domain filters

Poulami Somanya Ganguly, Daniel Pelt, Joost Batenburg
CWI, Amsterdam & Leiden University

6 July 2020

SIAM Imaging Science

Table of contents

1. Algebraic filters

   Fast real-space FBP reconstruction

2. Fourier-domain filters

   Ultra-fast FFT-based reconstruction for high angle counts

3. Implementation-specific filters

   Mitigate the influence of less accurate but fast backprojection operators

Introduction

Iterative reconstruction

Tomographic projection $\boldsymbol{p}$ is given by

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p} \tag{1}$$

Iterative reconstruction

Tomographic projection $\boldsymbol{p}$ is given by

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p} \tag{1}$$

To recover $\boldsymbol{x}$ solve the following optimisation problem:

$$\boldsymbol{x}_r = \arg\min_{\boldsymbol{x}} \|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}\|_2^2 \tag{2}$$

Iterative reconstruction

Tomographic projection $\boldsymbol{p}$ is given by

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p} \tag{1}$$

To recover $\boldsymbol{x}$ solve the following optimisation problem:

$$\boldsymbol{x}_r = \arg\min_{\boldsymbol{x}} \|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}\|_2^2 \tag{2}$$

Kaczmarz: $\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} - \dfrac{\boldsymbol{p}_i - \boldsymbol{W}_i \boldsymbol{x}^{(k-1)}}{\boldsymbol{W}_i^T \boldsymbol{W}_i} \boldsymbol{W}_i^T$

Iterative reconstruction

Tomographic projection $\boldsymbol{p}$ is given by

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p} \tag{1}$$

To recover $\boldsymbol{x}$ solve the following optimisation problem:

$$\boldsymbol{x}_r = \arg\min_{\boldsymbol{x}} \|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}\|_2^2 \tag{2}$$

Kaczmarz: $\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} - \dfrac{\boldsymbol{p}_i - \boldsymbol{W}_i\boldsymbol{x}^{(k-1)}}{\boldsymbol{W}_i^T\boldsymbol{W}_i}\boldsymbol{W}_i^T$

SIRT: $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{C}\boldsymbol{W}^T\boldsymbol{R}(\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}^{(k)})$

Iterative reconstruction

Tomographic projection $\boldsymbol{p}$ is given by

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p} \tag{1}$$

To recover $\boldsymbol{x}$ solve the following optimisation problem:

$$\boldsymbol{x}_r = \arg\min_{\boldsymbol{x}} \|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}\|_2^2 \tag{2}$$

Kaczmarz: $\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} - \frac{\boldsymbol{p}_i - \boldsymbol{W}_i\boldsymbol{x}^{(k-1)}}{\boldsymbol{W}_i^T\boldsymbol{W}_i}\boldsymbol{W}_i^T$

SIRT: $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{C}\boldsymbol{W}^T\boldsymbol{R}(\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}^{(k)})$

+ Robust w.r.t. noise and limited angular range

– Computationally expensive

Iterative reconstruction

$$\boldsymbol{x}^{(k+1)} = \underbrace{(\mathbf{1} - \boldsymbol{C}\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W})}_{\boldsymbol{A}}\boldsymbol{x}^{(k)} + \underbrace{\boldsymbol{C}\boldsymbol{W}^T\boldsymbol{R}}_{\boldsymbol{B}}\boldsymbol{p}$$

$$= \begin{pmatrix} \mathbf{1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}^k \begin{pmatrix} \mathbf{0} \\ \boldsymbol{p} \end{pmatrix}$$

Iterative reconstruction

$$\boldsymbol{x}^{(k+1)} = \underbrace{(\boldsymbol{1} - \boldsymbol{C}\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W})}_{\boldsymbol{A}}\boldsymbol{x}^{(k)} + \underbrace{\boldsymbol{C}\boldsymbol{W}^T\boldsymbol{R}}_{\boldsymbol{B}}\boldsymbol{p}$$

$$= \begin{pmatrix} \boldsymbol{1} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{1} \end{pmatrix}^k \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{p} \end{pmatrix}$$

Any *linear* iterative method (e.g., Kaczmarz, SIRT) can be written as

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{S}_k \boldsymbol{p}, \tag{3}$$

Iterative reconstruction

$$\boldsymbol{x}^{(k+1)} = \underbrace{(\mathbf{1} - \boldsymbol{C}\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W})}_{\boldsymbol{A}}\boldsymbol{x}^{(k)} + \underbrace{\boldsymbol{C}\boldsymbol{W}^T\boldsymbol{R}}_{\boldsymbol{B}}\boldsymbol{p}$$
$$= \begin{pmatrix} \mathbf{1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}^k \begin{pmatrix} \mathbf{0} \\ \boldsymbol{p} \end{pmatrix}$$

Any *linear* iterative method (e.g., Kaczmarz, SIRT) can be written as

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{S}_k\boldsymbol{p}, \tag{3}$$

Analytical methods (e.g. filtered backprojection) are also linear operations on data $\boldsymbol{p}$

Iterative reconstruction

$$\boldsymbol{x}^{(k+1)} = \underbrace{(\mathbf{1} - \boldsymbol{C}\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W})}_{\boldsymbol{A}}\boldsymbol{x}^{(k)} + \underbrace{\boldsymbol{C}\boldsymbol{W}^T\boldsymbol{R}}_{\boldsymbol{B}}\boldsymbol{p}$$

$$= \begin{pmatrix} \mathbf{1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}^k \begin{pmatrix} \mathbf{0} \\ \boldsymbol{p} \end{pmatrix}$$

Any *linear* iterative method (e.g., Kaczmarz, SIRT) can be written as

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{S}_k\boldsymbol{p}, \tag{3}$$

Analytical methods (e.g. filtered backprojection) are also linear operations on data $\boldsymbol{p}$

Can we approximate iterative reconstruction results with *faster* analytical methods?

Algebraic filters

Filter optimisation

An analytical method $\mathcal{A}$ such as filtered backprojection (FBP) can be written as

$$\begin{aligned}\mathcal{A}(\boldsymbol{h}, \boldsymbol{p}) &= \boldsymbol{W}_{\mathcal{A}}^{T} \boldsymbol{C}_{\boldsymbol{h}} \boldsymbol{p} \\ &= \boldsymbol{W}_{\mathcal{A}}^{T} \boldsymbol{C}_{\boldsymbol{p}} \boldsymbol{h},\end{aligned}$$

where $\boldsymbol{h}$ is a filter

Filter optimisation

An analytical method $\mathcal{A}$ such as filtered backprojection (FBP) can be written as

$$\begin{aligned}\mathcal{A}(\boldsymbol{h}, \boldsymbol{p}) &= \boldsymbol{W}_{\mathcal{A}}^{T} \boldsymbol{C}_{\boldsymbol{h}} \boldsymbol{p} \\ &= \boldsymbol{W}_{\mathcal{A}}^{T} \boldsymbol{C}_{\boldsymbol{p}} \boldsymbol{h},\end{aligned}$$

where $\boldsymbol{h}$ is a filter

We can optimise this filter by solving an optimisation problem similar to that in iterative methods

$$\boldsymbol{h}^{*} = \arg\min_{\boldsymbol{h}} \|\boldsymbol{p} - \boldsymbol{W}\mathcal{A}(\boldsymbol{h}, \boldsymbol{p})\|_2^2$$

$$\boldsymbol{h}^{*} = \arg\min_{\boldsymbol{h}} \|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{W}_{\mathcal{A}}^{T} \boldsymbol{C}_{\boldsymbol{p}} \boldsymbol{h}\|_2^2 \tag{4}$$

Filter optimisation

An analytical method $\mathcal{A}$ such as filtered backprojection (FBP) can be written as

$$\begin{aligned}\mathcal{A}(\boldsymbol{h}, \boldsymbol{p}) &= \boldsymbol{W}_{\mathcal{A}}^{T} \boldsymbol{C}_{\boldsymbol{h}} \boldsymbol{p} \\ &= \boldsymbol{W}_{\mathcal{A}}^{T} \boldsymbol{C}_{\boldsymbol{p}} \boldsymbol{h},\end{aligned}$$

where $\boldsymbol{h}$ is a filter

We can optimise this filter by solving an optimisation problem similar to that in iterative methods

$$\boldsymbol{h}^* = \arg\min_{\boldsymbol{h}} \|\boldsymbol{p} - \boldsymbol{W}\mathcal{A}(\boldsymbol{h}, \boldsymbol{p})\|_2^2$$

$$\boldsymbol{h}^* = \arg\min_{\boldsymbol{h}} \|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{W}_{\mathcal{A}}^{T} \boldsymbol{C}_{\boldsymbol{p}} \boldsymbol{h}\|_2^2 \tag{4}$$

h^* is the **minimum-residual** filter

Advantages

Typical size of projection data is $N_\theta N_d \sim 10^6$
Reconstruction volume size is $N^2 \sim 10^6$

Advantages

Typical size of projection data is $N_\theta N_d \sim 10^6$
Reconstruction volume size is $N^2 \sim 10^6$

- Filter size is $N_d \sim 10^3$

Advantages

Typical size of projection data is $N_\theta N_d \sim 10^6$
Reconstruction volume size is $N^2 \sim 10^6$

- Filter size is $N_d \sim 10^3$
- Filter dimension can be further reduced to $\mathcal{O}(\log N_d)$

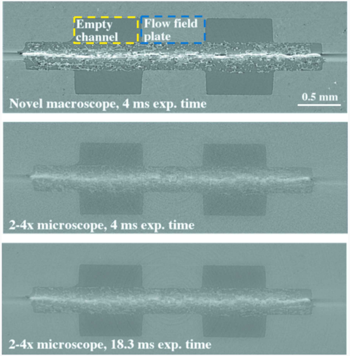
Advantages

Typical size of projection data is $N_\theta N_d \sim 10^6$
Reconstruction volume size is $N^2 \sim 10^6$

- Filter size is $N_d \sim 10^3$
- Filter dimension can be further reduced to $\mathcal{O}(\log N_d)$
- Filter can be reused for similar noise experimental geometries and noise statistics

High-throughput applications

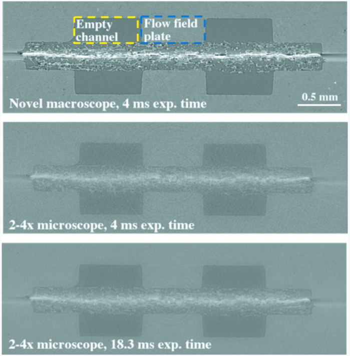
Time-resolved experiments
(tomography at synchrotrons)



Reconstructions of a fuel cell
(Bührer et al., 2019)

High-throughput applications

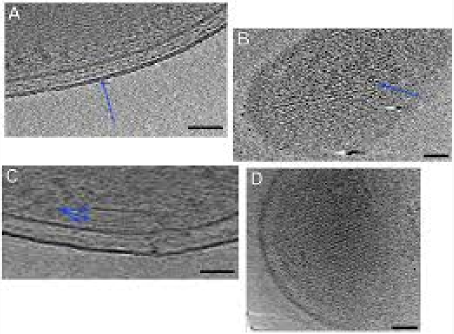
Time-resolved experiments
(tomography at synchrotrons)



Reconstructions of a fuel cell
(Bührer et al., 2019)

Data rates $\sim$ 10GB s^{-1} $\rightarrow$

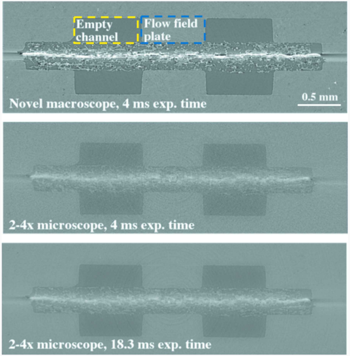
Continuous acquisition
(continuous-tilt electron tomography)



Tomograms of *Bdellovibrio bacteriovorus*
(Chreifi et al., 2019)

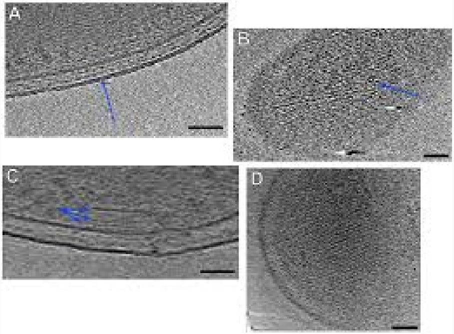
High-throughput applications

Time-resolved experiments
(tomography at synchrotrons)



Reconstructions of a fuel cell
(Bührer et al., 2019)

Continuous acquisition
(continuous-tilt electron tomography)

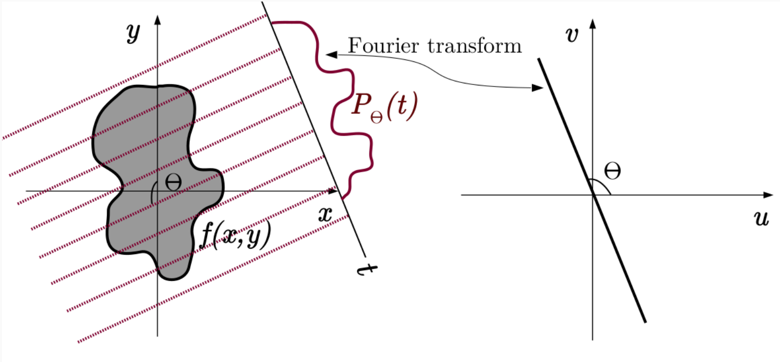


Tomograms of *Bdellovibrio bacteriovorus*
(Chreifi et al., 2019)

Data rates $\sim$ 10GB s^{-1} $\rightarrow$ Can we make reconstructions even faster?

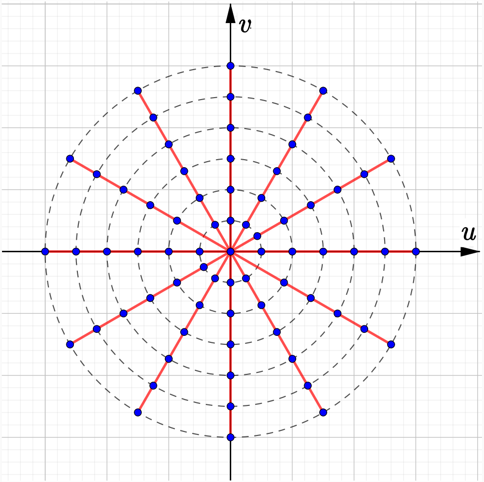
Fourier-domain filters

Fourier slice theorem



$$\tilde{P}_\theta(\omega) = \tilde{F}(\omega\cos\theta, \omega\sin\theta)$$

Fourier slice theorem



Fourier domain algorithms

Marone, F., and M. Stampanoni. "Regridding reconstruction algorithm for real-time tomographic imaging." (2012)

Fourier domain algorithms

Marone, F., and M. Stampanoni. "Regridding reconstruction algorithm for real-time tomographic imaging." (2012)

Steps in Gridrec

- filtering in Fourier space

Fourier domain algorithms

Marone, F., and M. Stampanoni. "Regridding reconstruction algorithm for real-time tomographic imaging." (2012)

Steps in Gridrec

- filtering in Fourier space
- convolution with windowing function

Fourier domain algorithms

Marone, F., and M. Stampanoni. "Regridding reconstruction algorithm for real-time tomographic imaging." (2012)

Steps in Gridrec

- filtering in Fourier space
- convolution with windowing function
- interpolation in Fourier space

Fourier domain algorithms

Marone, F., and M. Stampanoni. "Regridding reconstruction algorithm for real-time tomographic imaging." (2012)

Steps in Gridrec

- filtering in Fourier space
- convolution with windowing function
- interpolation in Fourier space
- 2D inverse FFT

Fourier domain algorithms

Marone, F., and M. Stampanoni. "Regridding reconstruction algorithm for real-time tomographic imaging." (2012)

Steps in Gridrec

- filtering in Fourier space
- convolution with windowing function
- interpolation in Fourier space
- 2D inverse FFT
- deapodisation

Fourier domain algorithms

Marone, F., and M. Stampanoni. "Regridding reconstruction algorithm for real-time tomographic imaging." (2012)

Steps in Gridrec

- filtering in Fourier space
- convolution with windowing function
- interpolation in Fourier space
- 2D inverse FFT
- deapodisation

FBP backprojection: $\mathcal{O}(N^2 N_\theta)$
Gridrec 2D FFT: $\mathcal{O}(N^2 \log N)$

Fourier domain algorithms

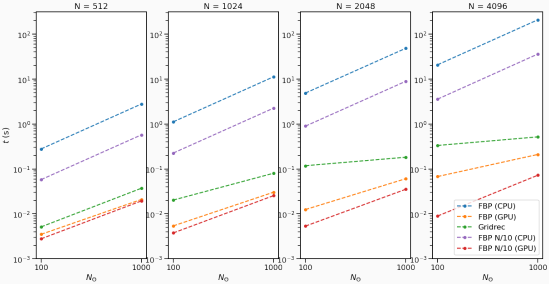
Marone, F., and M. Stampanoni. "Regridding reconstruction algorithm for real-time tomographic imaging." (2012)

Steps in Gridrec

- filtering in Fourier space
- convolution with windowing function
- interpolation in Fourier space
- 2D inverse FFT
- deapodisation

FBP backprojection: $\mathcal{O}(N^2 N_\theta)$
Gridrec 2D FFT: $\mathcal{O}(N^2 \log N)$

Reconstruction times for N_θ projection angles



Fourier domain algorithms

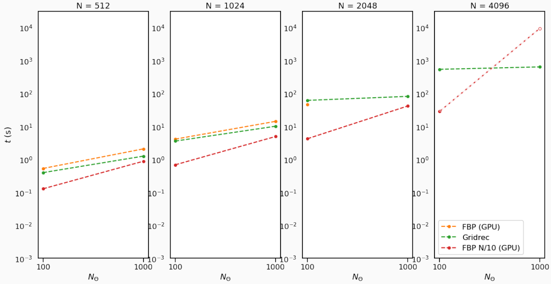
Marone, F., and M. Stampanoni. "Regridding reconstruction algorithm for real-time tomographic imaging." (2012)

Steps in Gridrec

- filtering in Fourier space
- convolution with windowing function
- interpolation in Fourier space
- 2D inverse FFT
- deapodisation

FBP backprojection: $\mathcal{O}(N^2 N_\theta)$
Gridrec 2D FFT: $\mathcal{O}(N^2 \log N)$

Reconstruction times for N_θ projection angles



Filter computation

- Compute minimum-residual filter

$$\boldsymbol{h}^* = \arg\min_{\boldsymbol{h}} \|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{W}^T \boldsymbol{C_p}\boldsymbol{h}\|_2^2$$

  using projectors in Fourier domain

Filter computation

- Compute minimum-residual filter

$$\boldsymbol{h}^* = \arg\min_{\boldsymbol{h}} \|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{W}^T \boldsymbol{C}_{\boldsymbol{p}}\boldsymbol{h}\|_2^2$$

  using projectors in Fourier domain
- Speed up filter computation by expanding in suitable basis

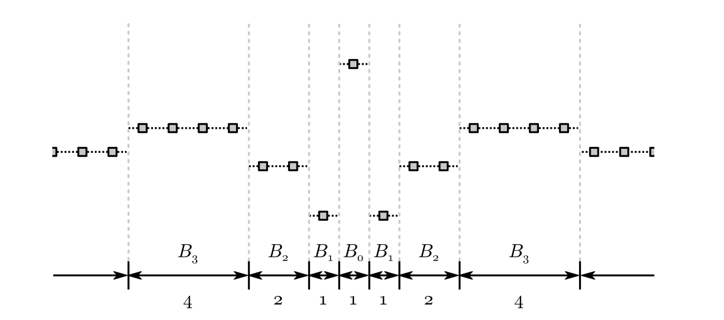
Filter computation

- Compute minimum-residual filter

$$\boldsymbol{h}^* = \arg\min_{\boldsymbol{h}} \|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{W}^T \boldsymbol{C_p}\boldsymbol{h}\|_2^2$$

using projectors in Fourier domain

- Speed up filter computation by expanding in suitable basis



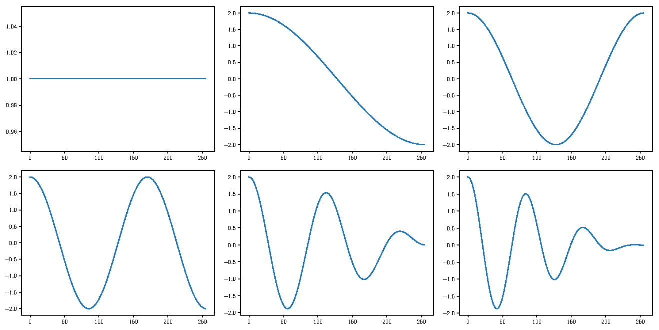
Filter computation

- Compute minimum-residual filter

$$\boldsymbol{h}^* = \arg\min_{\boldsymbol{h}} \|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{W}^T \boldsymbol{C}_{\boldsymbol{p}} \boldsymbol{h}\|_2^2$$

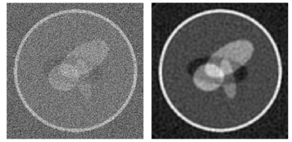
  using projectors in Fourier domain

- Speed up filter computation by expanding in suitable basis



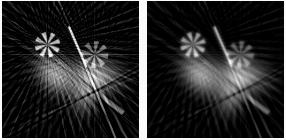
Results — 1: phantom

Noisy sinogram



ramp filter our filter

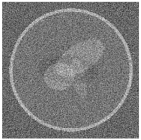
Sinogram with few angles

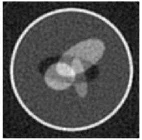


ramp filter our filter

Results — 1: phantom

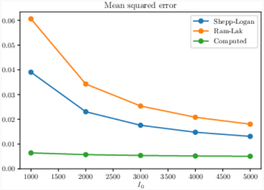
Noisy sinogram

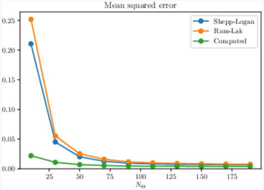




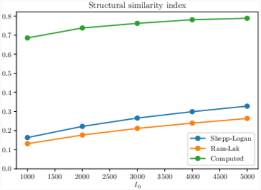
ramp filter   our filter

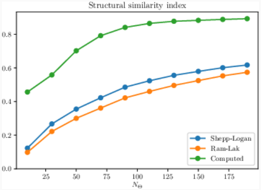
Mean squared error



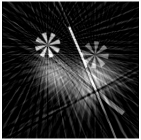


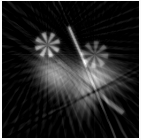
Structural similarity





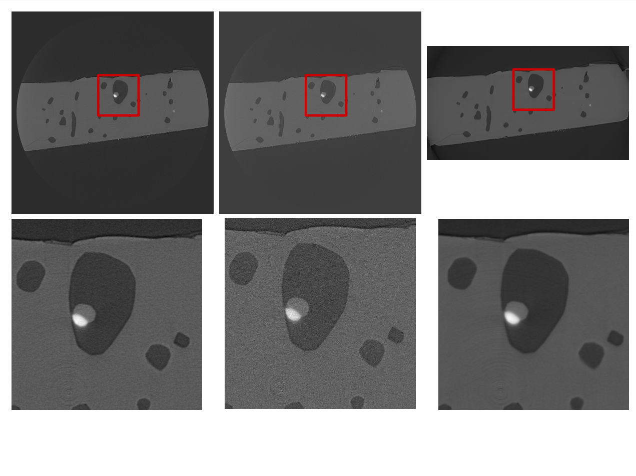
Sinogram with few angles





ramp filter   our filter

Results — 2: synchrotron data



(*left to right*) Gridrec reconstructions using a minimum-residual filter, the Parzen filter, and additional phase retrieval

High performance implementations

- Different implementations of the same algorithm result in *quantitatively* different reconstructions

High performance implementations

- Different implementations of the same algorithm result in *quantitatively* different reconstructions
- Differences arise due to choice of discretisation and/or interpolation

High performance implementations

- Different implementations of the same algorithm result in *quantitatively* different reconstructions
- Differences arise due to choice of discretisation and/or interpolation
- Mismatch between different backprojectors and real (physical) forward operator

High performance implementations

- Different implementations of the same algorithm result in *quantitatively* different reconstructions
- Differences arise due to choice of discretisation and/or interpolation
- Mismatch between different backprojectors and real (physical) forward operator

$\rightarrow$ Compute minimum-residual filters for standard implementations

High performance implementations

- Different implementations of the same algorithm result in *quantitatively* different reconstructions
- Differences arise due to choice of discretisation and/or interpolation
- Mismatch between different backprojectors and real (physical) forward operator

$\rightarrow$ Compute minimum-residual filters for standard implementations
! Does not require knowledge of the implementation

Implementation-specific filters

Results

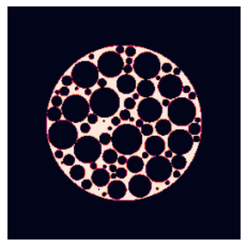
Minimum-residual filters *reduce* the mismatch between operators

Results

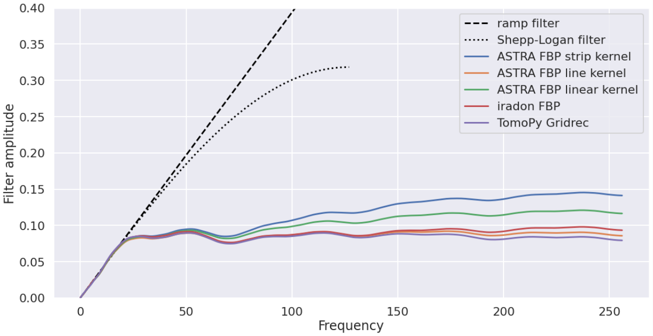
Minimum-residual filters *reduce* the mismatch between operators
$\Rightarrow$ more quantitatively similar reconstructions

Results

Minimum-residual filters *reduce* the mismatch between operators
$\Rightarrow$ more quantitatively similar reconstructions



Foam phantom



Minimum-residual filter shapes

Results

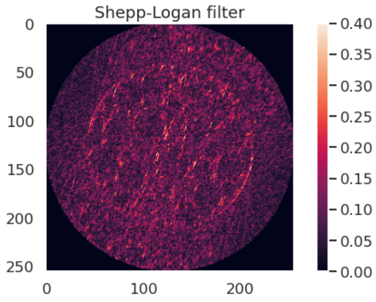
Pixelwise standard deviation in reconstructions

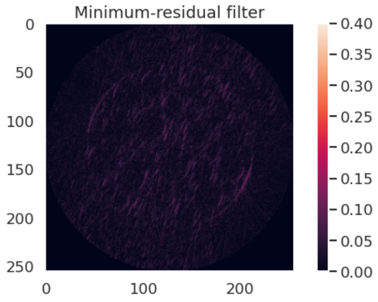
$$\mathrm{std}\{\boldsymbol{x}_r^{\text{strip}}, \boldsymbol{x}_r^{\text{line}}, \boldsymbol{x}_r^{\text{linear}}, \boldsymbol{x}_r^{\text{iradon}}, \boldsymbol{x}_r^{\text{gridrec}}\}$$

Results

Pixelwise standard deviation in reconstructions

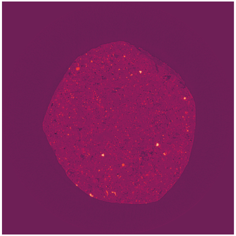
$$\text{std}\{\boldsymbol{x}_r^{\text{strip}}, \boldsymbol{x}_r^{\text{line}}, \boldsymbol{x}_r^{\text{linear}}, \boldsymbol{x}_r^{\text{iradon}}, \boldsymbol{x}_r^{\text{gridrec}}\}$$





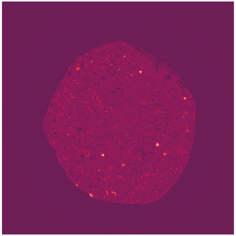
Results

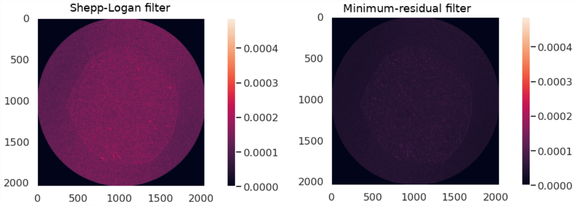
Round-robin dataset from Tomobank

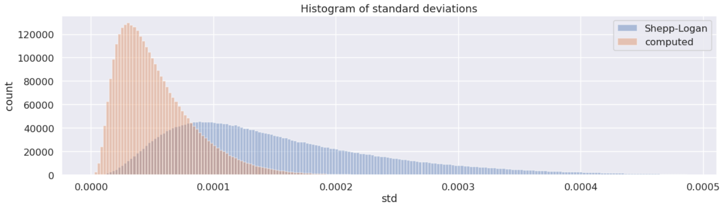


Results

Round-robin dataset from Tomobank







Conclusions

Summary and outlook

- Analytical algorithms despite their inability to handle imperfect data are **widely used** in practice because they are **fast**
- One way to **improve reconstruction quality** of these algorithms is by computing a **minimum-residual filter** for the available data
- The filter, once computed, can be **reused** for experimental setups where the projection geometry and noise levels are similar

Summary and outlook

- Analytical algorithms despite their inability to handle imperfect data are **widely used** in practice because they are **fast**
- One way to **improve reconstruction quality** of these algorithms is by computing a **minimum-residual filter** for the available data
- The filter, once computed, can be **reused** for experimental setups where the projection geometry and noise levels are similar
- Implementation-specific filters **reduce the mismatch** between forward and backprojection operators
- This leads to more **quantitatively similar** reconstructions

Summary and outlook

- Analytical algorithms despite their inability to handle imperfect data are **widely used** in practice because they are **fast**
- One way to **improve reconstruction quality** of these algorithms is by computing a **minimum-residual filter** for the available data
- The filter, once computed, can be **reused** for experimental setups where the projection geometry and noise levels are similar
- Implementation-specific filters **reduce the mismatch** between forward and backprojection operators
- This leads to more **quantitatively similar** reconstructions

- Learn filter by optimising to more than one dataset
- More general approach for learning corrections to the backprojection operator from data

Thank you for your attention!