Edge-Preserving Computed Tomography (CT) with Uncertain View Angles

Per Christian Hansen

DTU Compute, Technical University of Denmark

Joint work with

Johnathan M. Bardsley – Univ. of Montana, USA

Yiqiu Dong & Nicolai A. B. Riis – DTU, Denmark

Felipe Uribe – LUT Univ., Finland

F. Uribe, J.M. Bardsley, Y. Dong, P.C. Hansen, & N.A.B. Riis, *A hybrid Gibbs sampler for edge-preserving tomographic reconstruction with uncertain angles*, SIAM/ASA J. UQ, 10 (2022), pp. 1293–1320, doi 10.1137/21M1412268.

sites.dtu.dk/cuqi — cuqi-dtu.github.io/CUQIpy

This work is part of the project **C**omputational **U**ncertainty **Q**uantification for **I**nverse problems, which is funded by The Villum Foundation.

- A collaborative effort to develop a mathematical, statistical and computational framework for UQ.
- We released the first version of our software package:
- Some applications
  - X-ray computed tomography: industrial inspection, materials science.
  - Electrical impedance tomography (EIT) and hybrid EIT.
  - Inverse problems in tokamak plasma physics.
  - Dynamical models in drug kinetics.

Overview of This Talk

Prelude

- X-ray CT model
- Uncertain view angles

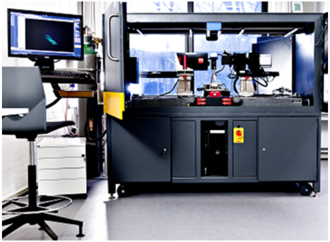
Fugue

- The joint problem of image reconstruction and view angle estimation
- A Bayesian framework
- A hybrid Gibbs sampler
- Some implementation details
- Numerical results

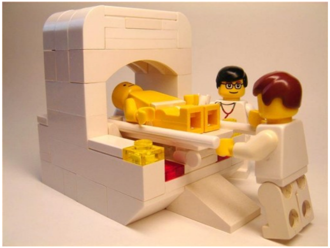
Coda

- Conclusions

Applications of X-Ray Computed Tomography (CT)

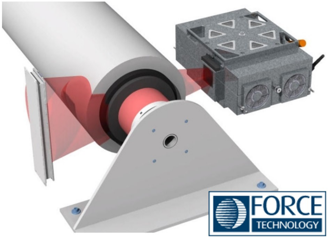


Lab scanner



Medical scanner

Synchrotron

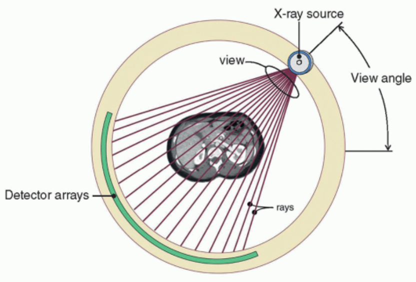


Industrial inspection

The Principles of X-Ray CT

The Principle

Send X-rays through the object at different *view angles*, and measure their attenuation.



Inverse problem: reconstruct an image of the object from the data.

Lambert–Beer law $\rightarrow$ attenuation of an X-ray through the object f is a line integral:

$$b_i = \int_{\text{ray}_i} f(x,y)\, d\ell \ ,$$

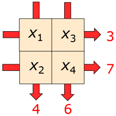
$$f = \text{attenuation coef.}$$

A discrete version:

$$\boxed{\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}}$$

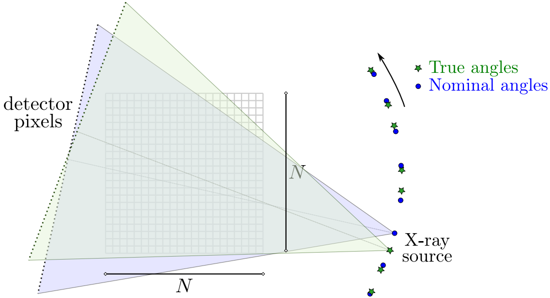
$\boldsymbol{A} \sim$ measurement geometry,

$\boldsymbol{x} \sim$ reconstruction, $\quad \boldsymbol{b} \sim$ data.



$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 3 \\ 7 \\ 4 \\ 6 \end{pmatrix}$$

Uncertain View Angles



Each position of the X-ray source is defined by a corresponding view angle θ_i.

The **true view angles** θ_i^{tru} may differ from the assumed **nominal view angles** θ_i^{nom}.

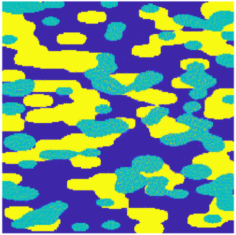
- The model for the measured data is $\boldsymbol{b} = \boldsymbol{A}(\boldsymbol{\theta}^{\text{tru}})\,\boldsymbol{x} + \boldsymbol{e}$, where $\boldsymbol{e}$ is the measurement noise, $\boldsymbol{x}$ represents the image, and $\boldsymbol{A}(\boldsymbol{\theta}^{\text{tru}})$ is the forward model defined for the *unknown* true angles.
- A "naive" (and potentially inferior) reconstruction uses the matrix $\boldsymbol{A}(\boldsymbol{\theta}^{\text{nom}})$ based on the nominal angles.

The Need for Handling Uncertain Angles

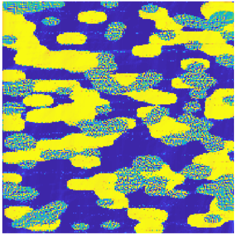
A simple example generated with the AIR Tools II MATLAB package:

```
N = 200;
theta_nom  = 3:3:180;
theta_true = theta_nom + 0.1*randn(size(theta_nom));
A_nom  = paralleltomo(N,theta_nom );
A_true = paralleltomo(N,theta_true);
X = phantomgallery('threephases',N); x = X(:); b = A_true*x;
options.lbound = 0; options.ubound = 1;
x_nom  = kaczmarz(A_nom, b,200,[],options);
x_true = kaczmarz(A_true,b,200,[],options);
```

Reconstr. with true angles



Reconstr. with nominal angles



Dealing with Uncertain Angles, No UQ (references in our papers)

Two-stage methods – first estimate the angles ("angle recovery," "alignment reconstitution"), then reconstruct with potential error propagation.

- Cross-correlation of projections of simple objects/phantoms, e.g., with few particles or spheres.
- Obtain information about the object's apparent movement through the use of markers.
- Use methods from computer vision.

Joint methods – estimate the true angles and the reconstruction via a single non-convex optimization problem.

- Solve the joint problem via an alternating variable-projection scheme.
- Solve the joint optimization problem in a Bayesian setting.
- Use a Bayesian approach based on a mixture framework.

Our method uses a Bayesian approach that solves the joint problem and *provides UQ* for both the reconstruction and the view angles.

Formulation of the Joint Problem

Notation:

- The vector $\boldsymbol{b}$ holds the measured data.
- The vectors $\boldsymbol{x}$ and $\boldsymbol{\theta}$ hold the image pixels and the view angles.
- The matrix $\boldsymbol{A}(\boldsymbol{\theta})$ represents the CT forward model for view angles $\boldsymbol{\theta}$.

The inverse problem is linear in $\boldsymbol{x}$ and nonlinar in $\boldsymbol{\theta}$:

$$\boxed{\text{find } (\boldsymbol{x}, \boldsymbol{\theta}) \text{ such that } \boldsymbol{b} = \boldsymbol{A}(\boldsymbol{\theta})\, \boldsymbol{x} + \boldsymbol{e}}$$

Here $\boldsymbol{e}$ represents additive measurement noise. The noise is log-Poisson; we approximate it with a Gaussian

$$\boldsymbol{e} \sim \mathcal{N}(0, \lambda \boldsymbol{I}) \ .$$

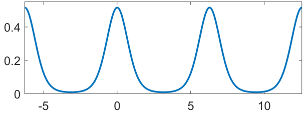
We want to be able to reconstruct edges in the image, since edges often carry the most important information, e.g., about defects or tumors.

The Bayesian Inverse Problem

We formulate a Bayesian inverse problem with a likelihood that involves both $\boldsymbol{x}$ and $\boldsymbol{\theta}$:

$$\pi_{\text{pos}}(\boldsymbol{x}, \boldsymbol{\theta}) \;\propto\; \pi_{\text{lik}}(\boldsymbol{b} \,|\, \boldsymbol{x}, \boldsymbol{\theta})\, \pi_{\text{pri}}(\boldsymbol{x})\, \pi_{\text{pri}}(\boldsymbol{\theta})\ .$$

- As mentioned, $\pi_{\text{lik}}(\boldsymbol{b} \,|\, \boldsymbol{x}, \boldsymbol{\theta})$ is a Gaussian.
- For $\pi_{\text{pri}}(\boldsymbol{x})$ we use a Laplace distribution of the differences of neighbour pixels. This enables the desired sharp edges in the image; it has connections to total variation (TV) regularization.
- For $\pi_{\text{pri}}(\boldsymbol{\theta})$ we use the von Mises distribution, i.e., a *periodic* normal distribution $\propto \exp(\kappa \cos(x))$.



Hyperparameters ("uninformative"):

- λ in the Gaussian likelihood,
- δ in the Laplace-difference prior for $\boldsymbol{x}$,
- κ in the von Mises prior for $\boldsymbol{\theta}$,
- exponential distributions $\pi_{\text{hyp}}(\cdot) = \beta \exp(-\beta\, \cdot)$ with $\beta = 10^{-4}$.

The Posterior

$$\pi_{\text{pos}}(\boldsymbol{x}, \boldsymbol{\theta}, \lambda, \delta, \kappa) \propto \pi_{\text{lik}}(\boldsymbol{b} \,|\, \boldsymbol{x}, \boldsymbol{\theta}, \lambda) \times \pi_{\text{pri}}(\boldsymbol{x} \,|\, \delta) \times \pi_{\text{pri}}(\boldsymbol{\theta} \,|\, \kappa) \times \pi_{\text{hyp}}(\lambda) \times \pi_{\text{hyp}}(\delta) \times \pi_{\text{hyp}}(\kappa)$$

with $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{\theta} \in \mathbb{R}^p$ and

$$\pi_{\text{lik}}(\boldsymbol{b} \,|\, \boldsymbol{x}, \boldsymbol{\theta}, \lambda) = \left(\frac{\lambda}{2\pi}\right)^{m/2} \exp\left(-\frac{\lambda}{2}\|\boldsymbol{A}(\boldsymbol{\theta})\,\boldsymbol{x} - \boldsymbol{b}\|_2^2\right) \qquad \text{(Gaussian)}$$

$$\pi_{\text{pri}}(\boldsymbol{x} \,|\, \delta) = \left(\frac{\delta}{2}\right)^{n} \exp\left(-\delta(\|(\boldsymbol{I} \otimes \boldsymbol{D})\,\boldsymbol{x}\|_1 + \|(\boldsymbol{D} \otimes \boldsymbol{I})\,\boldsymbol{x}\|_1)\right) \qquad \text{(“Laplace difference”)}$$

$$\pi_{\text{pri}}(\boldsymbol{\theta} \,|\, \kappa) = \left(\frac{1}{2\pi I_0(\kappa)}\right)^{p} \exp\left(\kappa \sum_{i=1}^{p} \cos(\theta_i - \theta_i^{\text{nom}})\right) \qquad \text{(von Mises)}$$

in which $\boldsymbol{I}$ = identity matrix and $\boldsymbol{D} = \text{bidiag}(-1, 1)$.

Conditional Densities for the Posterior

$$\pi_{\text{pos}}(\boldsymbol{x},\boldsymbol{\theta},\lambda,\delta,\kappa) = \lambda^{m/2}\,\delta^{n}\,I_0(\kappa)^{-p}\exp\Big(-\tfrac{\lambda}{2}\|\boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x}-\boldsymbol{b}\|_2^2 -\delta\big(\|(\boldsymbol{I}\otimes\boldsymbol{D})\boldsymbol{x}\|_1+\|(\boldsymbol{D}\otimes\boldsymbol{I})\boldsymbol{x}\|_1\big)+\kappa\sum_{i=1}^{p}\cos(\theta_i-\theta_i^{\text{nom}})-\beta\lambda-\beta\delta-\beta\kappa\Big)$$

Conditional densities:

$$\pi_1(\boldsymbol{x}\,|\,\boldsymbol{\theta},\lambda,\delta) \propto \exp\Big(-\tfrac{\lambda}{2}\|\boldsymbol{A}(\boldsymbol{\theta})\,\boldsymbol{x}-\boldsymbol{b}\|_2^2 - \delta\big(\|(\boldsymbol{I}\otimes\boldsymbol{D})\,\boldsymbol{x}\|_1 + \|(\boldsymbol{D}\otimes\boldsymbol{I})\,\boldsymbol{x}\|_1\big)\Big)$$

$$\pi_2(\boldsymbol{\theta}\,|\,\boldsymbol{x},\lambda,\kappa) \propto \exp\left(-\tfrac{\lambda}{2}\|\boldsymbol{A}(\boldsymbol{\theta})\,\boldsymbol{x}-\boldsymbol{b}\|_2^2 + \kappa\sum_{i=1}^{p}\cos(\theta_i-\theta_i^{\text{nom}})\right)$$

$$\pi_3(\lambda\,|\,\boldsymbol{x},\boldsymbol{\theta}) \propto \lambda^{m/2}\exp\left(-\lambda\left(\frac{1}{2}\|\boldsymbol{A}(\boldsymbol{\theta})\,\boldsymbol{x}-\boldsymbol{b}\|_2^2+\beta\right)\right)$$

$$\pi_4(\delta\,|\,\boldsymbol{x}) \propto \delta^{n}\exp\big(-\delta\,\big(\|(\boldsymbol{I}\otimes\boldsymbol{D})\,\boldsymbol{x}\|_1 + \|(\boldsymbol{D}\otimes\boldsymbol{I})\,\boldsymbol{x}\|_1+\beta\big)\big)$$

$$\pi_5(\kappa\,|\,\boldsymbol{\theta}) \propto I_0(\kappa)^{-p}\exp\left(-\kappa\left(\sum_{i=1}^{p}\cos(\theta_i-\theta_i^{\text{nom}})+\beta\right)\right)$$

Hybrid Gibbs Sampler → Different Samplers for Each Conditional

Initial states $\boldsymbol{x}^{(0)}, \boldsymbol{\theta}^{(0)}, \lambda^{(0)}, \delta^{(0)}, \kappa^{(0)}$

For $j = 1, 2, \ldots, N_{\text{samp}}$

Sample image pixels

$$\boldsymbol{x}^{(j)} \sim \pi_1\left(\cdot \,|\, \boldsymbol{\theta}^{(j-1)}, \lambda^{(j-1)}, \delta^{(j-1)}\right)$$

Sample view angles

$$\boldsymbol{\theta}^{(j)} \sim \pi_2\left(\cdot \,|\, \boldsymbol{x}^{(j)}, \lambda^{(j-1)}, \kappa^{(j-1)}\right)$$

Sample hyperparameters

$$\lambda^{(j)} \sim \pi_3\left(\cdot \,|\, \boldsymbol{x}^{(j)}, \boldsymbol{\theta}^{(j)}\right)$$

$$\delta^{(j)} \sim \pi_4\left(\cdot \,|\, \boldsymbol{x}^{(j)}\right)$$

$$\kappa^{(j)} \sim \pi_5\left(\cdot \,|\, \boldsymbol{\theta}^{(j)}\right)$$

End

Our implementation draws on several existing methods – our contribution is to bring them together and make them work.

The Main Challenge: How to Work With π_1

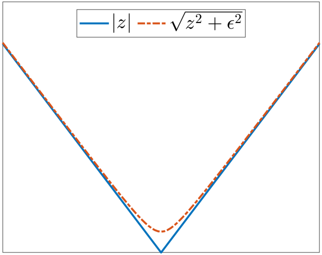
$$\pi_1(\boldsymbol{x}, \boldsymbol{\theta}, \lambda, \delta, \kappa) \propto \exp\Big(-\tfrac{\lambda}{2}\|\boldsymbol{A}(\boldsymbol{\theta})\,\boldsymbol{x} - \boldsymbol{b}\|_2^2 - \delta\big(\|(\boldsymbol{I}\otimes\boldsymbol{D})\,\boldsymbol{x}\|_1 + \|(\boldsymbol{D}\otimes\boldsymbol{I})\,\boldsymbol{x}\|_1\big)\Big)$$

Linear and large-scale in $\boldsymbol{x}$

- Use an iterative solver: CGLS (conjugate gradients for least squares problems)

Non-differentiable due to $\|\cdot\|_1$

- Introduce the usual smoothing (known from, say, TV) $\longrightarrow$



Nonlinear in $\boldsymbol{\theta}$.

- Wikipedia: *Laplace's approximation* fits an un-normalised Gaussian approximation to a (twice differentiable) un-normalised target density.

More About Laplace's Approximation

Approximate $\pi_1(\boldsymbol{x})$ by a *Gaussian* density $\pi_G(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{H}^{-1})$ with $\boldsymbol{\mu}$ = MAP estimator of π_1 and $\boldsymbol{H}$ = approximate Hessian of $-\log \pi_1$:

$$
\begin{aligned}
\boldsymbol{\mu}(\boldsymbol{x}) &= \lambda\, \boldsymbol{H}^{-1}(\boldsymbol{x})\, \boldsymbol{A}(\boldsymbol{\theta})^\top \boldsymbol{b} \\
\boldsymbol{H}(\boldsymbol{x}) &= \lambda\, \boldsymbol{A}(\boldsymbol{\theta})^\top \boldsymbol{A}(\boldsymbol{\theta}) + \delta \Big((\boldsymbol{I} \otimes \boldsymbol{D})^\top \boldsymbol{W}_1(\boldsymbol{x}) (\boldsymbol{I} \otimes \boldsymbol{D}) + \\
&\qquad (\boldsymbol{D} \otimes \boldsymbol{I})^\top \boldsymbol{W}_2(\boldsymbol{x}) (\boldsymbol{D} \otimes \boldsymbol{I}) \quad \Big) \\
\boldsymbol{W}_1(\boldsymbol{x}) &= \operatorname{diag}\Big[\big\{ ((\boldsymbol{I} \otimes \boldsymbol{D})\boldsymbol{x})^2 + \varepsilon^2 \big\}^{-1/2} \Big] \\
\boldsymbol{W}_2(\boldsymbol{x}) &= \operatorname{diag}\Big[\big\{ ((\boldsymbol{D} \otimes \boldsymbol{I})\boldsymbol{x})^2 + \varepsilon^2 \big\}^{-1/2} \Big]
\end{aligned}
$$

For details see, e.g., (Bardsley, 2018, §4.3.1).

Much easier to work with – but the Gaussian approximation $\pi_G(\boldsymbol{x})$ misses the heavy tails of $\pi_1(\boldsymbol{x})$ and hence our uncertainties are imprecise.

We now develop a so-called *horseshoe prior* (Uribe, Dong, H, 2023), also based on a Gaussian approximation, that can better handle heavy tails.

How To Sample From π_G?

In principle we can use Metropolis-Hastings to sample from π_G; but we only produce one proposal which is always accepted. This is conceptually identical to the unadjusted Langevin algorithm (ULA).

We thus obtain $\boldsymbol{x}^{(j)}$ by solving the following linear least squares problem with a random perturbation of the right-hand side:

$$\min_{\boldsymbol{x}} \left\| \begin{bmatrix} \sqrt{\lambda}\, \boldsymbol{A}(\boldsymbol{\theta}^{(j-1)}) \\ \sqrt{\delta}\, \boldsymbol{W}_1(\boldsymbol{x}^{(j-1)})^{1/2}\, (\boldsymbol{I} \otimes \boldsymbol{D}) \\ \sqrt{\delta}\, \boldsymbol{W}_2(\boldsymbol{x}^{j-1)})^{1/2}\, (\boldsymbol{D} \otimes \boldsymbol{I}) \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \sqrt{\lambda}\boldsymbol{b} + \boldsymbol{\xi}_0 \\ \boldsymbol{\xi}_1 \\ \boldsymbol{\xi}_2 \end{bmatrix} \right\|_2$$

where $\boldsymbol{\xi}_i \sim \mathcal{N}(0, \boldsymbol{I})$ for $i = 0, 1, 2$.

This done by means of the CGLS iterative method.

- Each iteration involves one matrix-vector multiplication with $\boldsymbol{A}(\boldsymbol{\theta}^{(j-1)})$ and one with its transpose.
- We found experimentally that 10 iterations are sufficient.

How to Sample π_2?

We introduce a partitioning of the matrix and the rhs, where each block $\boldsymbol{A}(\theta_i)$ and $\boldsymbol{b}(\theta_i)$ correspond to the ith view angle.

We use a single-component Metropolis algorithm with componentwise updates applied to

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}(\theta_1) \\ \boldsymbol{A}(\theta_2) \\ \vdots \\ \boldsymbol{A}(\theta_p) \end{bmatrix} \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}(\theta_1) \\ \boldsymbol{b}(\theta_2) \\ \vdots \\ \boldsymbol{b}(\theta_p) \end{bmatrix}$$

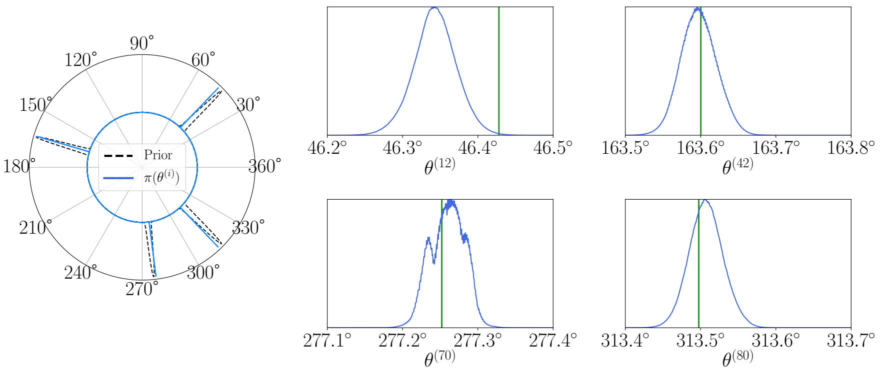
$$\pi_2(\boldsymbol{\theta} \,|\, \boldsymbol{x}, \lambda, \kappa) \propto \prod_{i=1}^{p} \exp\left(-\tfrac{\lambda}{2} \|\boldsymbol{A}(\theta_i)\,\boldsymbol{x} - \boldsymbol{b}(\theta_i)\|_2^2 + \kappa \cos(\theta_i - \theta_i^{\mathrm{nom}})\right).$$

Given $\boldsymbol{\theta}^{[0]} = \boldsymbol{\theta}^{(j-1)}$ from the Gibbs sampler, we perform 20 burn-in cycles:

$$\begin{aligned}
\theta_1^{[k+])} &\sim \pi_2\left(\boldsymbol{\theta} \,|\, \boldsymbol{x}, \lambda, \kappa, \left[\theta_2^{[k]}, \theta_3^{[k]}, \ldots, \theta_p^{[k]}\right]\right), \\
\theta_2^{[k+1]} &\sim \pi_2\left(\boldsymbol{\theta} \,|\, \boldsymbol{x}, \lambda, \kappa, \left[\theta_1^{[k+1]}, \theta_3^{[k]}, \ldots, \theta_p^{[k]}\right]\right), \\
&\vdots \\
\theta_p^{[k+1]} &\sim \pi_2\left(\boldsymbol{\theta} \,|\, \boldsymbol{x}, \lambda, \kappa, \left[\theta_1^{[k+1]}, \theta_2^{[k+1]}, \ldots, \theta_{p-1}^{[k+])}\right]\right).
\end{aligned}$$

This produces the next sample $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^{[20]}$ in our hybrid Gibbs sampler.

Illustration of Working with π_2



Left: von Mises prior with the respective component densities.

Right: zoom-in on selected component densities; the true angles are shown as solid green lines.

In this example, the values of $\boldsymbol{x}$, λ and κ are assumed known.

Sampling the Hyperpriors

▷ The conditional density π_3 can be written in closed form

$$\pi_3(\lambda \,|\, \boldsymbol{x}, \boldsymbol{\theta}) = \frac{\omega^\tau}{\Gamma(\tau)}\, \lambda^{\tau-1} \exp(-\omega\, \lambda).$$

with $\tau = \frac{m}{2} + 1$ and $\omega = \frac{1}{2}\|\boldsymbol{A}(\boldsymbol{\theta})\, \boldsymbol{x} - \boldsymbol{b}\|_2^2 + \beta$

▷ We approximate the conditional density π_4 by a distribution written in closed form

$$\tilde{\pi}_4(\delta \,|\, \boldsymbol{x}) \approx \frac{\varpi^\nu}{\Gamma(\nu)}\, \delta^{\nu-1} \exp(-\varpi\, \delta)$$

with $\nu = n + 1$ and
$\varpi = \boldsymbol{x}^\top \big((\boldsymbol{I} \otimes \boldsymbol{D})^\top \boldsymbol{W}_1(\boldsymbol{x})(\boldsymbol{I} \otimes \boldsymbol{D}) + (\boldsymbol{D} \otimes \boldsymbol{I})^\top \boldsymbol{W}_2(\boldsymbol{x})(\boldsymbol{D} \otimes \boldsymbol{I}) \big) \boldsymbol{x} + \beta.$

▷ Sampling of the conditional density $\pi_5(\kappa \,|\, \boldsymbol{\theta})$ is done by a standard random-walk Metropolis algorithm.

Numerical Experiments

Software: our Python package CUQIpy + ASTRA package for CT models.

We compare with the CT-VAE method (Riis et al., 2020) that computes the MAP estimate (no UQ) through

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{C}_{\boldsymbol{\nu}}(\boldsymbol{A}(\theta)\,\boldsymbol{x} - \boldsymbol{b} + \boldsymbol{\mu}_{\boldsymbol{\nu}})\|_2^2 + \gamma \mathrm{TV}(\boldsymbol{x}),$$

where $\mathrm{TV}(\boldsymbol{x})$ denotes Total Variation regularization and γ = regularization parameter. In this model, $\boldsymbol{\nu}$ = measurement noise + model discrepancy, with mean $\boldsymbol{\mu}_{\boldsymbol{\nu}}$ and covariance matrix $(\boldsymbol{C}_{\boldsymbol{\nu}}^\top \boldsymbol{C}_{\boldsymbol{\nu}})^{-1}$.

We generate noisy data

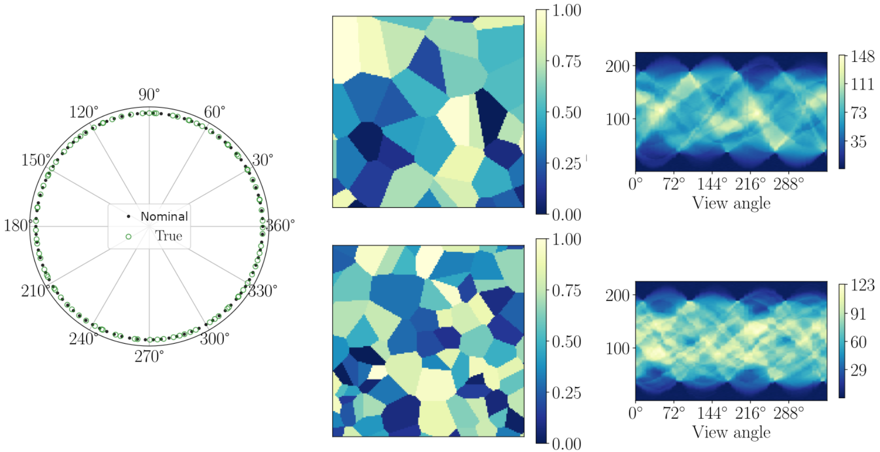
$$\boldsymbol{b} = \boldsymbol{A}(\theta^{\mathrm{tru}})\,\boldsymbol{x}^{\mathrm{tru}} + \boldsymbol{e}, \qquad \boldsymbol{e} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$$

with noise level $\sigma = 0.01\|\boldsymbol{A}(\theta^{\mathrm{tru}})\,\boldsymbol{x}^{\mathrm{tru}}\|_2/\sqrt{m}$.

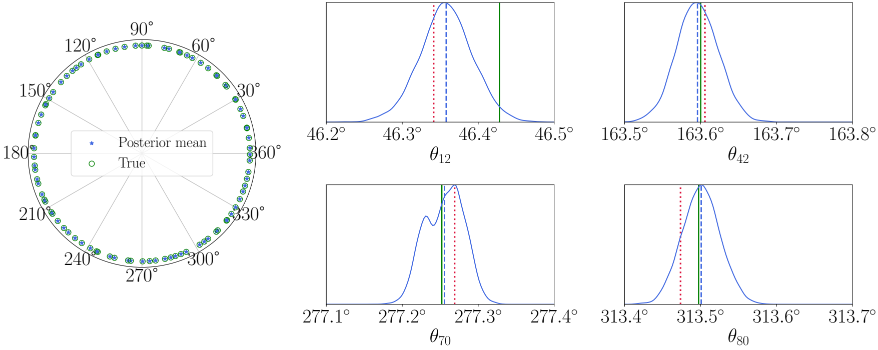
The Fanbeam CT Problem

Image size $150 \times 150 \rightarrow n = 22\,500$ pixels. Detector with 225 pixels and $p = 90$ view angles $0°, 4°, 8°, \ldots, 356° \rightarrow m = 20\,250$ measurements.

Phantoms “grains” and “ppower” from AIR Tools II.



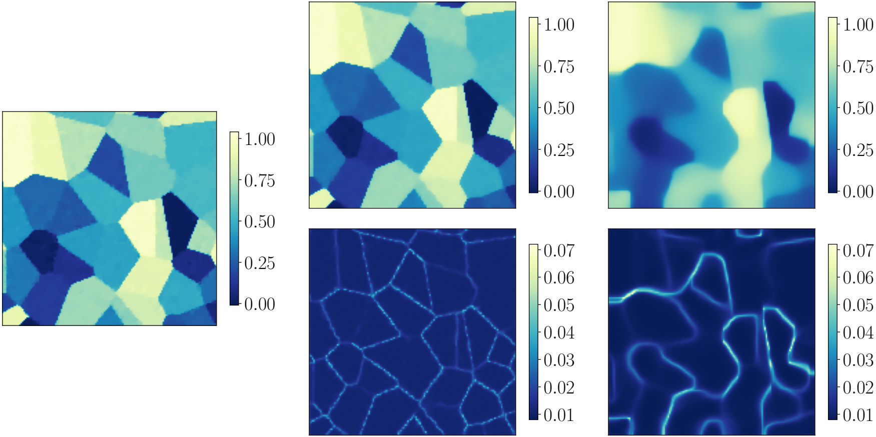
Estimated View Angles



Left: posterior mean and true view angles θ_i.

Right: zoom-in on component densities for selected θ_i. True angles are shown as solid green lines, posterior mean angles as dashed blue lines, and the angles estimated by the CT-VAE method are shown as dotted red lines.

Estimated Reconstruction of Phantom with 50 Grains

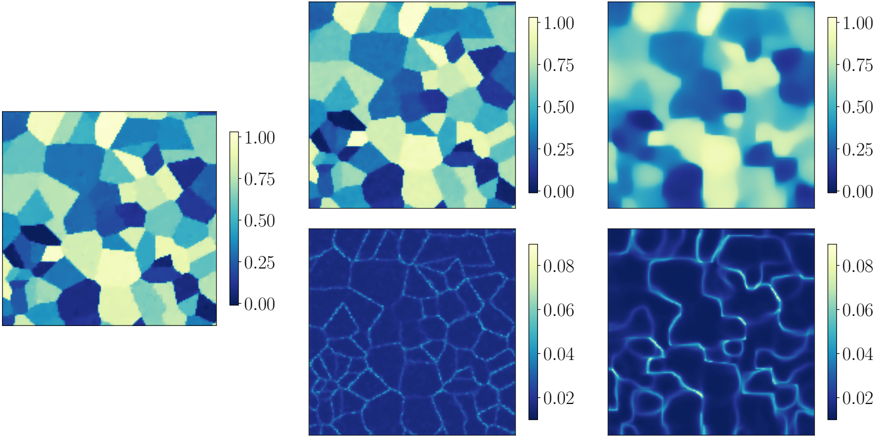


Left: MAP estimate from the CT-VAE method.

Middle: posterior mean and standard deviation from our method.

Right: ditto from hybrid Gibbs sampler with fixed nominal angles.

Estimated Reconstruction of Phantom with 100 Grains

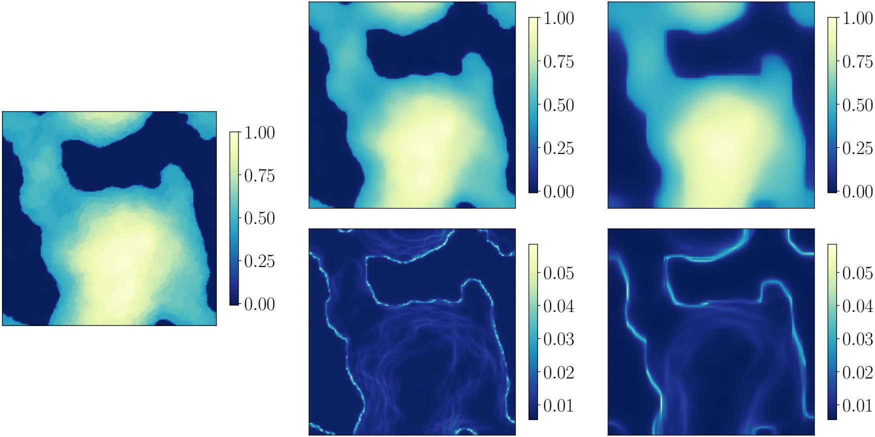


Left: MAP estimate from the CT-VAE method.

Middle: posterior mean and standard deviation from our method.

Right: ditto from hybrid Gibbs sampler with fixed nominal angles.

Estimated Reconstruction of Sparse Phantom



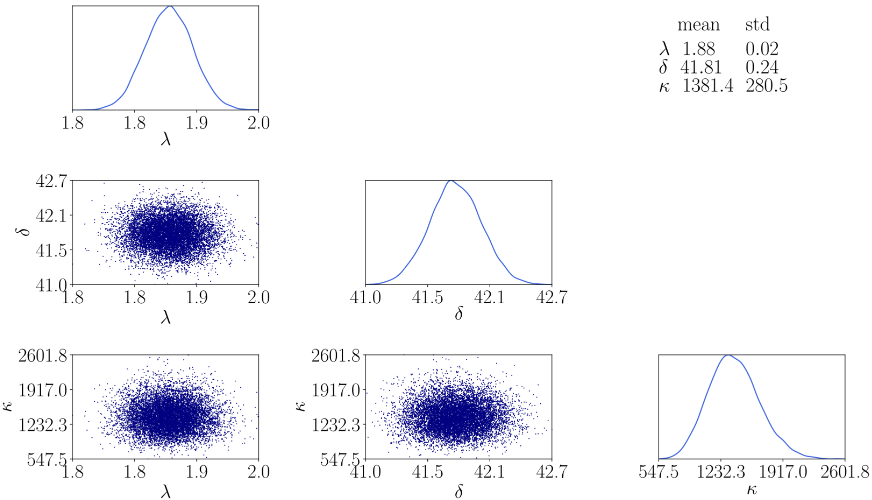
Left: MAP estimate from the CT-VAE method.

Middle: posterior mean and standard deviation from our method.

Right: ditto from hybrid Gibbs sampler with fixed nominal angles.

Posterior Hyperparameters – Sparse Phantom

	mean	std
λ	1.88	0.02
δ	41.81	0.24
κ	1381.4	280.5



Samples and estimated densities of λ (noise parameter), δ (Laplace-diff. parameter), and κ (von Mises parameter).

Coda – Conclusions

In short

- In many CT problems we need to correct for uncertain view angles.
- The Bayesian framework allows us to solve the joint problem . . .
- and perform UQ on both the reconstruction and the angles.
- Our numerical results confirm the applicability of our method.

Challenges

- Deriving an efficient sampler is challenging:
  - high dimension problem for the image $\boldsymbol{x}$,
  - nonlinear problem for the view angles $\boldsymbol{\theta}$.
- The Gaussian approximation misses the heavy tails of the posterior; our new *horseshoe prior* seeks to circumvent with this issue.
- A framework for other large-scale inverse problems with uncertain parameters – what are good preconditioners for CGLS?
- How to incorporate the framework in our software package CUQIpy?