EFFICIENT MUON TOMOGRAPHY IMAGE RECONSTRUCTION USING GRADIENT DESCENT OPTIMIZATION

Jean-Marco Alameddine, Felix Sattler, Angel Bueno Rodriguez, Maximilian Pérez Prada, Maurice Stephan, and Sarah Barnes for the German Aerospace Center

DLR

DLR contributions to Muographers

Data generation:

"B2X - Seamless integration of complex 3D scenes with Monte Carlo frameworks"
Felix Sattler

Reconstruction:

"Efficient Muon Tomography Image Reconstruction Using Gradient Descent Optimization"
This talk

Introduce a statistical reconstruction method that is:

- Accurate
- Flexible
- Computationally effective

Application / Data analysis:

"Simulation study on high-resolution muon tomography of a spent nuclear fuel storage cask"
Maximilian Pérez Prada (Wednesday, 13:00)

"Anomaly Detection in Operational Muon Cargo Inspection with Real Scan Data"
Angel Bueno Rodriguez (Wednesday, 13:45)

Gradient-descent Reconstruction

Idea: Formulate a likelihood $\mathcal{L}(D|\vec{\lambda})$ which quantifies how good a model $\vec{\lambda}$ is in agreement with the given data D

- The model $\vec{\lambda}$ is given by the voxel density map (this is to be optimized)
- The data is given by the muon measurements for each event (these are fixed)

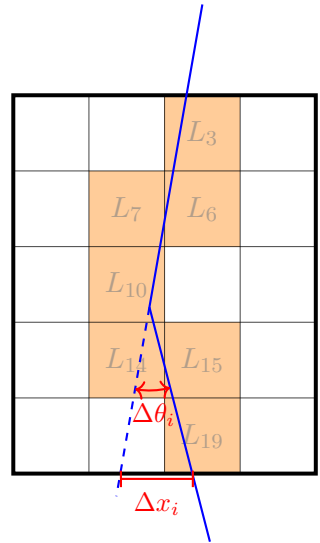
$\Rightarrow$ Find optimal $\vec{\lambda}$ by maximizing the likelihood $\mathcal{L}$

Note: The formulation of the problem follows the definitions given by Schultz, L. J. et al. in "Statistical Reconstruction for Cosmic Ray Muon Tomography." *IEEE Transactions on Image Processing* 16 (2007): 1985-1993

Gradient-descent Reconstruction – Defining Data

Given the ingoing and outgoing muon positions and directions, we calculate:

1. Angular change $\Delta\theta_i$ and displacement Δx_i of track
   - → Data vector $\mathbf{D_i} = (\Delta\theta_i, \Delta x_i)^T$
2. Distances L_j covered in each voxel j
   - → We need to make assumptions about the muon track! (e.g. straight-line, PoCA-path, etc.)
   - → This geometrical information is calculated once for each muon i and stored in weight matrices $\mathbf{W_{i,j}}$



Gradient-descent – Likelihood for Single Muon Event

$$\mathcal{L}(D_i|\lambda) = \frac{1}{2\pi\sqrt{|\Sigma_i|}} \exp\left(-\frac{1}{2} D_i^T \Sigma_i^{-1} D_i\right)$$

with

- $D_i = (\Delta\theta_i, \Delta x_i)^T \leftarrow$ Data vector for muon i
- $\Sigma_i = (p_0/p_i)^2 \sum_{j:\text{voxels}} (\lambda_j W_{i,j}) \leftarrow$ Covariance matrix
- $W_{i,j} \leftarrow$ weight matrix (geometric information from voxel tracing)
- $\lambda_j \leftarrow$ Scattering density in voxel j
- $p_i \leftarrow$ Momentum of muon i

Gradient-descent – Likelihood for Multiple Muon Events

Combining information from several muon events i (which are independent) and using information in three-dimensions (where x and y are independent)

$$\mathcal{L}(D|\lambda) = \prod_{i:\ \text{muons}} \mathcal{L}(D_{i,x}|\lambda) \cdot \mathcal{L}(D_{i,y}|\lambda)$$

leads to the overall negative log-likelihood

$$-\log(\mathcal{L}) = \sum_i \left(\log(|\Sigma_i|) + \tfrac{1}{2} D_{i,x}^T \Sigma_i^{-1} D_{i,x} + \tfrac{1}{2} D_{i,y}^T \Sigma_i^{-1} D_{i,y}\right)$$

Definition of Likelihood for Multiple Muon Events

This negative log-likelihood

$$-\log(\mathcal{L}) = \sum_i \left(\log(|\Sigma_i|) + \tfrac{1}{2} D_{i,x}^T \Sigma_i^{-1} D_{i,x} + \tfrac{1}{2} D_{i,y}^T \Sigma_i^{-1} D_{i,y}\right)$$

quantifies how well our **scattering density values** λ_j describe our **given data** $D_i = (\Delta\theta_i, \Delta x_i)^T$ under the given **assumption about the tracing**.

We now optimize the λ_j by maximizing the likelihood $\mathcal{L}$, i.e., by minimizing the negative log-likelihood $-\log(\mathcal{L})$

Likelihood Maximization – Efficient Implementation

- Implementation of the algorithm in PyTorch
  - → Prominent features from ML directly available
  - → Allows for performant calculations on GPU
- Likelihood optimization performed via automatic differentiation
  - → Quick and uncertainty-free calculations of gradients
  - → Natively available in PyTorch (gradients are calculated on-the-fly)
- Tracing of muon paths performed in C++, with parallelization
- Batch-wise calculation possible for large datasets
  - → Processing of data in batches in case memory consumption becomes too high
  - → Tracing results can be stored intermediately

Example: Contraband in Industrial Boiler

NEW ZEALAND / CRIME

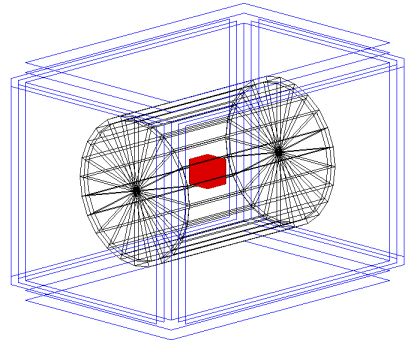
Seven facing charges after 190kg of cocaine found in boiler from Ecuador

7:01 pm on 15 November 2022

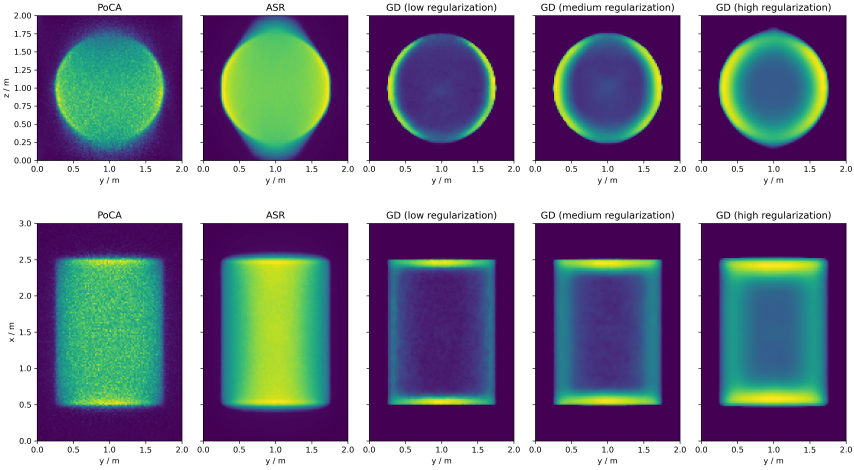
Share this

Example: Contraband in Industrial Boiler

- Simple boiler simulation with GEANT4
  - → Boiler made of **stainless steel** with 5 cm walls
  - → $(25 \times 25 \times 25)$ cm^3 block of **cocaine** hidden at the center ($\approx$20 kg)
- Muons simulated with EcoMug
  - → 60 000 000 muons injected on a half-spherical surface
  - → Corresponding to a measurement time of $\approx$4 h
- Detectors plates positioned at all sides
  - → Assuming availability of perfect position and momentum information



Industrial Boiler – Reconstruction Results



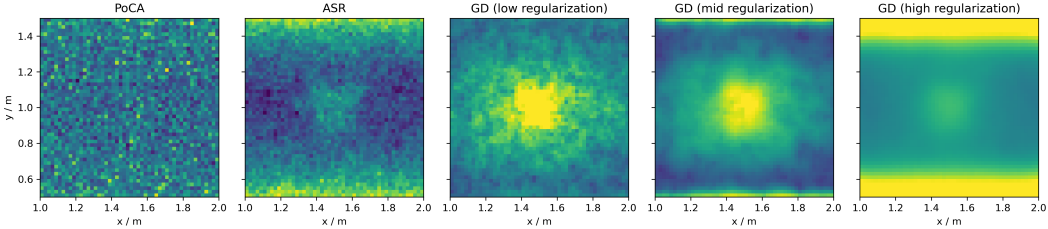
Side view (y, z)

Top view (x, y)

Voxel size: 2 cm

Three different levels of regularization
for Gradient Descent ($\mathcal{L}' = \mathcal{L} + \mathcal{L}_{\text{reg}}$)

Industrial Boiler – Region of Interest

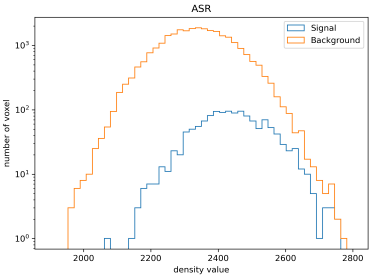


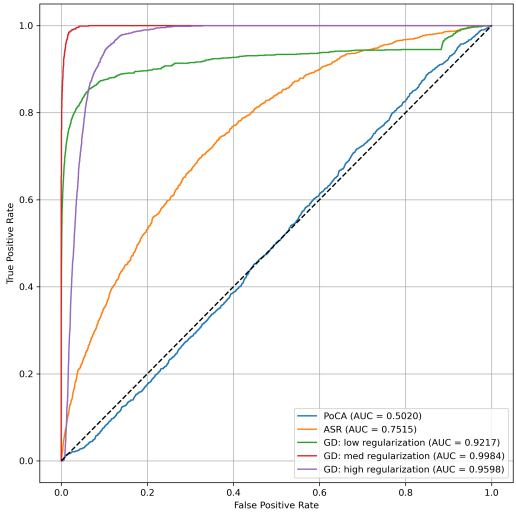
Signal-to-noise ratio (SNR)

PoCA	ASR	Gradient Descent		
		Low	Medium	High
−0.036	0.944	3.056	3.380	2.102

⇒ Qualitatively and quantitatively: Better contraband discrimination

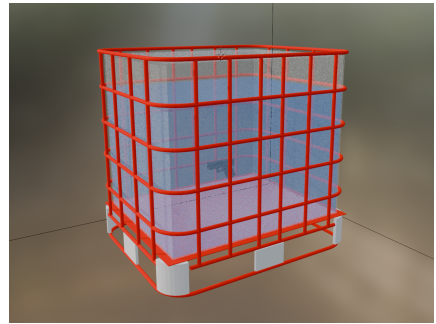
Industrial Boiler – Region of Interest





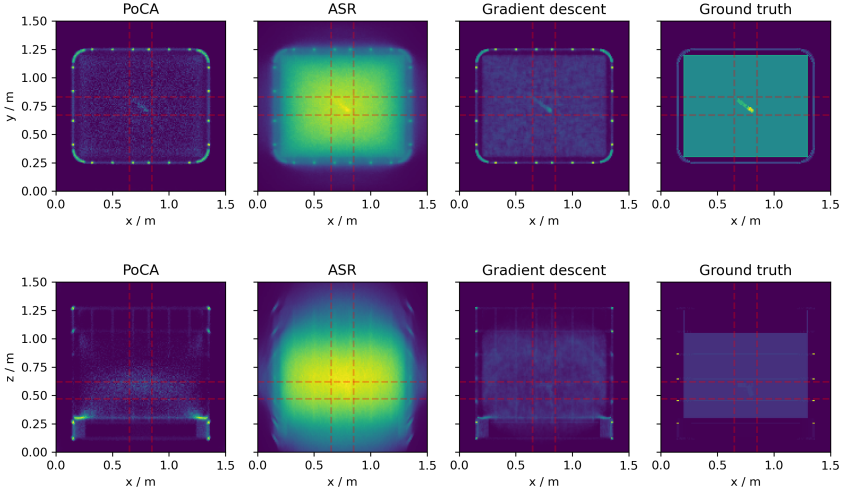
Example: Contraband in IBC

- IBC simulation with GEANT4 and B2G4
  - → See previous talk by Felix Sattler
  - → Realistic model of **IBC filled with water**
  - → **Hidden pistol** made out of stainless steel
- Muons simulated with EcoMug
  - → 42 000 000 muons injected on a half-spherical surface
  - → Corresponding to a measurement time of ≈4 h
- Detectors plates positioned at all sides
  - → Assuming availability of perfect position and momentum information



Silent Border

Contraband in IBC – Reconstruction Results

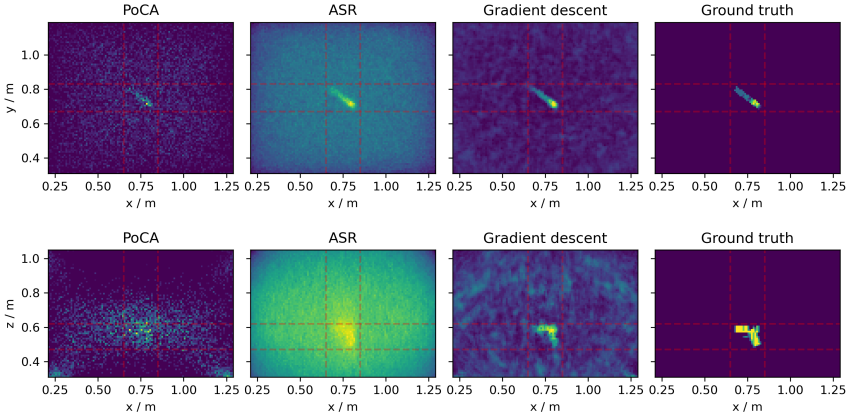


Top view (x, y)

Side view (x, z)

Voxel size: 1 cm

Contraband in IBC – Region of Interest



Top view (x, y)

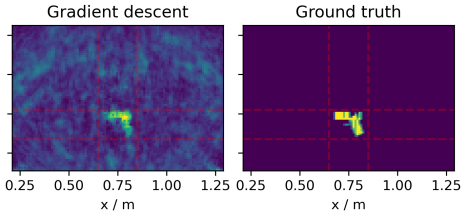
Side view (x, z)

Signal-to-noise ratio (SNR)

PoCA	ASR	GD
2.025	4.339	11.516

Summary & Outlook

- Presented the theoretical foundation of a gradient descent reconstruction algorithm
- Introduced a PyTorch implementation, utilizing GPU computation and parallelization
- Algorithm shows improved contraband detection in simulation studies
- Validation of the algorithm on real data ongoing
- Exploiting the explicit likelihood structure of the method: $-\log(\mathcal{L}) \rightarrow -\log(\mathcal{L}) + p(\lambda)$



APPENDIX

Gradient-descent – Likelihood for Multiple Muon Events

$$
\begin{aligned}
-\log(\mathcal{L}) &= -\log\left(\prod_i \mathcal{L}(D_{i,x}|\lambda)\cdot\mathcal{L}(D_{i,y}|\lambda)\right) \\
&= -\sum_i \left[\log\left(\mathcal{L}(D_{i,x}|\lambda)\right)\right) + \left(\log\left(\mathcal{L}(D_{i,y}|\lambda)\right)\right] \\
&= -\sum_i \left[\log\left(\frac{1}{2\pi\sqrt{|\Sigma_i|}}\exp\left(-\frac{1}{2}D_{i,x}^T\Sigma_i^{-1}D_{i,x}\right)\right) + \right. \\
&\left. \log\left(\frac{1}{2\pi\sqrt{|\Sigma_i|}}\exp\left(-\frac{1}{2}D_{i,y}^T\Sigma_i^{-1}D_{i,y}\right)\right)\right] \\
&= \sum_i \left(\log\left(|\Sigma_i|\right) + \frac{1}{2}D_{i,x}^T\Sigma_i^{-1}D_{i,x} + \frac{1}{2}D_{i,y}^T\Sigma_i^{-1}D_{i,y}\right)
\end{aligned}
$$

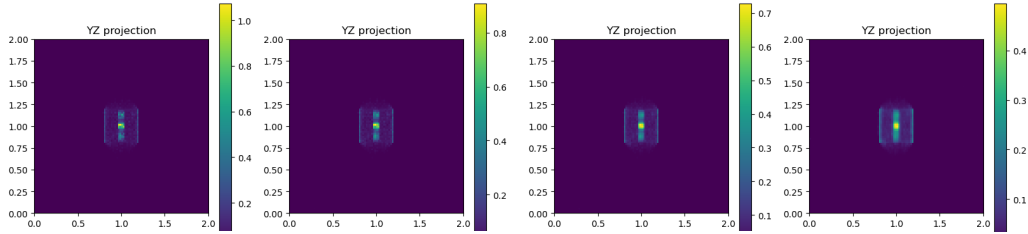
Benchmarking

	CPU peak RAM	GPU peak VRAM	Execution time (wall time)
PoCA	≈5.3 GB	≈2.2 GB	0:43.87 min
ASR	≈41.8 GB	≈17.5 GB	4:45.57 min
Grad. des.	≈14.6 GB	≈39.9 GB	7:04.57 min

Benchmarking results for Industrial Boiler Example
(1 500 000 voxels, 60 000 000 muons)

Regularization – Laplace Regularization

$$\mathcal{L}' = \mathcal{L} + \mathcal{L}_{\text{reg}}$$



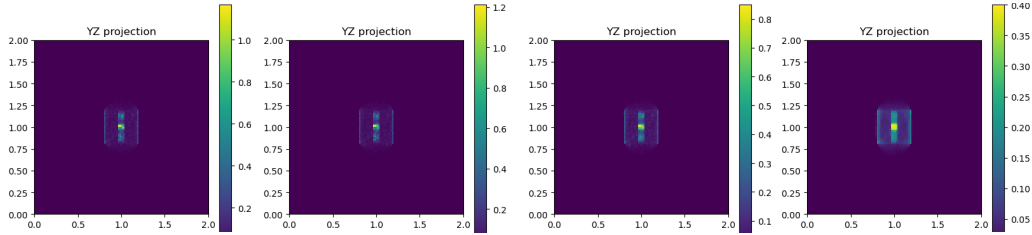
$c_{\text{Laplace}} = 10^1$ $c_{\text{Laplace}} = 10^2$ $c_{\text{Laplace}} = 10^3$ $c_{\text{Laplace}} = 10^4$

$$\mathcal{L}_{\text{reg}} = c_{\text{Laplace}} \cdot \frac{1}{N} \sum_{\text{voxel}} ((x_{i+1,j,k} - x_{i,j,k}) + (x_{i-1,j,k} - x_{i,j,k}) + (x_{i,j+1,k} - x_{i,j,k}) + (x_{i,j-1,k} - x_{i,j,k}) + (x_{i,j,k+1} - x_{i,j,k}) + (x_{i,j,k-1} - x_{i,j,k}))^2$$

Penalize sharp local changes (curvature)

Regularization – Total Variation Regularization with Huber Loss

$$\mathcal{L}' = \mathcal{L} + \mathcal{L}_{\text{reg}}$$



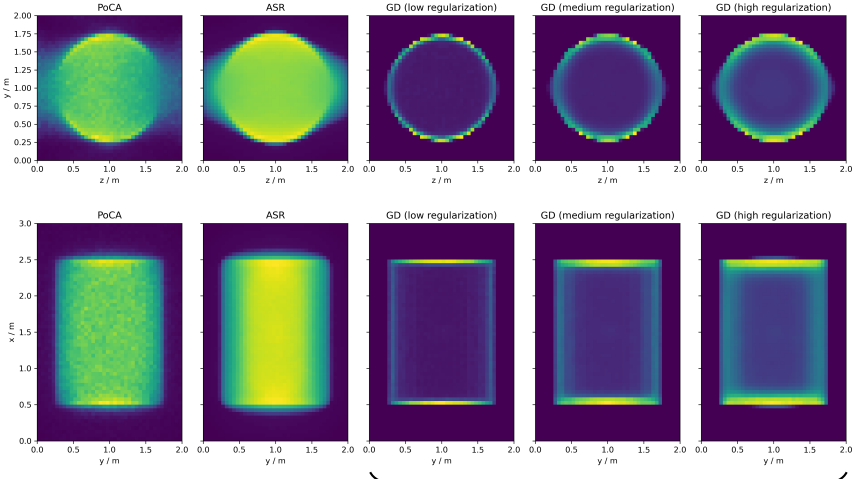
$c_{\text{Huber}} = 10^3$ $c_{\text{Huber}} = 10^4$ $c_{\text{Huber}} = 10^5$ $c_{\text{Huber}} = 10^6$

$$\mathcal{L}_{\text{reg}} = c_{\text{Huber}} \cdot \sum_{\text{voxel}} \phi\left(\sqrt{(x_{i,j,k} - x_{i-1,j,k})^2 + (x_{i,j,k} - x_{i,j-1,k})^2 + (x_{i,j,k} - x_{i,j,k-1})^2}\right)$$

with $\phi(x) = \begin{cases} 0.5g^2, & g \leq \delta \\ \delta(g - 0.5\delta), & g > \delta \end{cases}$

Penalize gradients while preserving sharp edges

Industrial Boiler – Reconstruction Results (5 cm voxel size)



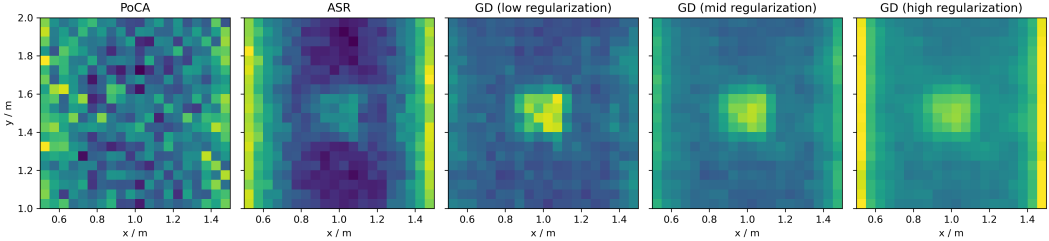
Side view (y, z)

Top view (x, y)

Voxel size: 5 cm

Three different levels of regularization
for Gradient Descent ($\mathcal{L}' = \mathcal{L} + \mathcal{L}_{\text{reg}}$)

Industrial Boiler – Region of Interest (5 cm voxel size)



Signal-to-noise ratio (SNR)

PoCA	ASR	Gradient Descent		
		Low	Medium	High
−0.192	1.807	5.8	6.859	4.406

⇒ Qualitatively and quantitatively: Better contraband discrimination

Imprint

Topic: **Efficient Muon Tomography Image Reconstruction Using Gradient Descent Optimization**

Date: 02. June 2026

Author: Jean-Marco Alameddine

Institute: Institute for the Protection of Maritime Infrastructures