Journal of
Applied Physics

RESEARCH ARTICLE | OCTOBER 13 2025

A comprehensive framework toward the seamless integration of muon reconstruction algorithms with machine learning

Special Collection: Muography: Discoveries, Innovations, and Applications

F. A. Sattler ✉ ; J. M. Alameddine ; Á. Bueno Rodríguez ; M. Stephan ; S. Barnes

J. Appl. Phys. 138, 144904 (2025)
https://doi.org/10.1063/5.0288348

Articles You May Be Interested In

Pillar embedding visualization for muon-scattering tomography

J. Appl. Phys. (October 2025)

Revealing secondary particle signatures in muography based on the point of closest approach algorithm

J. Appl. Phys. (January 2026)

A generalized muon trajectory estimation algorithm with energy loss for application to muon tomography

J. Appl. Phys. (March 2018)

AIP Publishing

Journal of
Applied Physics

ARTICLE

pubs.aip.org/aip/jap

A comprehensive framework toward the seamless integration of muon reconstruction algorithms with machine learning

Cite as: J. Appl. Phys. **138**, 144904 (2025); doi: 10.1063/5.0288348
Submitted: 30 June 2025 · Accepted: 22 September 2025 ·
Published Online: 13 October 2025

F. A. Sattler,[a] J. M. Alameddine, Á. Bueno Rodríguez, M. Stephan, and S. Barnes

AFFILIATIONS

German Aerospace Center (DLR), Institute for the Protection of Maritime Infrastructures, Fischkai 1, 27572 Bremerhaven, Germany

Note: This paper is part of the Special Topic on Muography: Discoveries, Innovations, and Applications.

[a] **Author to whom correspondence should be addressed:** felix.sattler@dlr.de

ABSTRACT

Muon-scattering tomography (MST) utilizes naturally occurring cosmic-ray muons to reveal the three-dimensional composition of concealed volumes, such as cargo containers in the maritime domain, reducing the need for artificial radiation sources. The reconstruction methods of current state-of-the-art systems rely on geometry-based approaches, such as the Point of Closest Approach (PoCA) algorithm, whose strong heuristics blur fine structures and introduce high-frequency noise. Statistical Expectation–Maximization (EM) reconstruction methods can recover these lost details but are traditionally ruled out for real-time application given their high computational and numerical demands. We introduce a comprehensive framework for MST reconstruction in PyTorch, including traditional and fast, but inaccurate geometry-based methods, as well as a highly optimized EM solver within a single, end-to-end differentiable pipeline. Using parallelism and graphics processing unit (GPU) acceleration, our framework overcomes the aforementioned computational obstacles. As a benchmark, the EM solver is tested on several MST scenarios generated with Geant4. Image quality metrics shows its superiority over traditional reconstruction algorithms, while retaining a per-iteration latency of 0.8 s at a 1 cm voxel resolution on standard GPUs.

I. INTRODUCTION

Maritime infrastructures form the backbone of modern society, enabling global trade, transportation, and resource distribution. However, these infrastructures face increasing threats from terrorism, crime, and natural disasters. Ensuring their protection and monitoring is crucial for border control and overall maritime security. Effective maritime situational awareness is essential to anticipate and respond to these challenges. Key to this effort is the integration of automation and intelligent systems to improve situational awareness, providing meaningful information to aid operators in decision-making.

Cosmic-ray tomography, first described by Borozdin *et al.* in the context of surveillance,[1] has emerged as a promising technique for enhancing this maritime situational awareness. By leveraging naturally occurring high-energy muons, muon-scattering tomography (MST) enables non-invasive and penetrating imaging of structures. Through the combination of at least two detectors on opposing sides of an object to be imaged, the incoming as well as outgoing position and direction of a muon, a so called track, can be recorded. The observed displacement in position and corresponding angle of scatter are used to infer the properties of the concealed materials. Since muons are naturally occurring particles that pose no threat to organisms, this approach is particularly suitable for monitoring maritime infrastructures and detecting concealed threats.

Complex scenarios, such as border control, demand robust 3D reconstruction techniques that make use of the available statistics generated by recording muon tracks and ideally incorporate strong priors to aid optimization. For this, several technical challenges must be addressed. These include robustness to noise, scalability to large datasets, and the need for high-performance implementations suitable for real-time applications. The integration of modern algorithmic tools—such as automatic differentiation and graphics processing unit (GPU) acceleration—also allows for new optimization strategies and learning-based approaches. For example, combining muon tomography with machine-learning techniques can enable data-driven models that learn from large datasets while benefiting from end-to-end differentiability and hardware acceleration. This work, therefore, introduces a comprehensive and scalable framework for 3D muon tomography.

The main contributions of this work are

- Differentiable, GPU-accelerated implementations of 3D reconstruction methods for MST, including an Expectation–Maximization (EM) algorithm, geometry-based approaches, and trainable 3D volumetric filters.
- Integration with modern machine-learning ecosystems, such as PyTorch,[2] enabling seamless experimentation and rapid prototyping of reconstruction methods.
- Interactive 3D data exploration tools that simplify the understanding and visualization of complex 3D reconstructed data.

By addressing these challenges, this work aims to establish a comprehensive and scalable framework for cosmic-ray tomography. The integration of advanced reconstruction techniques with machine learning will contribute to more effective and robust monitoring solutions.

II. RELATED WORKS

Two classical and widely used geometry-based algorithms for MST are the Point of Closest Approach (PoCA)[3] and Angle Statistics Reconstruction (ASR).[4] PoCA assumes that the most probable scattering location of a muon lies at the point where the incoming and outgoing trajectories are closest to each other and assigns the scattering angle to a single voxel in 3D space. ASR, on the other hand, uses statistical distributions of scattering angles across multiple voxels to infer high-density regions. While both methods are computationally efficient, they do not exploit prior knowledge about the scanned scene or incorporate statistical models of the measurement process.

A significant advancement beyond purely geometric approaches was introduced by Schultz *et al.*,[5] who proposed an Expectation–Maximization (EM) algorithm that incorporates measured scattering statistics (using the displacement and angular change of the muon track) and allows for the inclusion of prior information (such as those coming from geometry-based methods) into the reconstruction process. This marked a shift toward more probabilistic and model-based reconstruction techniques in MST. Building on this, Riggi *et al.*[6] conducted a comparative study evaluating several reconstruction methods, including a refined EM algorithm, specifically for detecting high-Z materials inside shipping containers. Their work demonstrated the potential of statistical methods for enhanced material discrimination. However, the focus was primarily on high-Z threats (e.g., nuclear materials), while broader border security applications must also address low-Z materials, such as powders and liquid substances.

Barnes *et al.*[7] explored the application of muon tomography for detecting a wider range of illicit substances, highlighting its utility in comprehensive border security settings. Their work underscored the need for general-purpose and adaptable reconstruction techniques.

Recently, machine learning has been applied to MST reconstruction, showing promising improvements in image quality and reconstruction speed.[8,9] A differentiable pipeline has also been demonstrated by Strong *et al.*[10] to optimize detector setups. These methods, often based on supervised learning, leverage curated training datasets to learn complex mappings from muon tracks to material distributions or sensor configurations. While effective, such approaches require substantial amounts of labeled data, which may not always be available or generalizable across different environments.

Consequently, there remains a strong need for reliable, unsupervised 3D reconstruction methods, such as the EM algorithm. Beyond its standalone utility, a scalable and robust EM implementation can serve as a valuable prior or initialization strategy when training more complex AI models. As outlined by Schultz *et al.*,[5] their method is a global optimization technique that utilizes all available statistics in a scene and, therefore, comes with increased computational complexity and memory demands. Specifically, the traditional EM method relates every muon track M with every voxel N, thus scaling with a memory complexity of $\mathcal{O}(MN)$. Even with a short exposure time and coarse grid resolution, the algorithm becomes prohibitively expensive in terms of memory. Riggi *et al.*[6] circumvent this by using a hash map (a data structure that maps keys to values using a hash function to determine an index into an array of buckets), storing only a sparse subset of interactions. While reducing the memory complexity, hash maps are hard to access in parallel,[11] therefore strongly limiting the effectiveness of parallel computations.

We, therefore, propose a novel, sparse implementation of the EM algorithm that works in a parallel manner on the GPU together with optimized geometry-based methods. The implementation of the EM algorithm and the corresponding methods, together with efficient implementations of the traditional PoCA and ASR algorithms, are combined into a consistent framework for PyTorch that enables end-to-end learning through automatic differentiation. This allows for efficient benchmarking of various algorithms for MST in the context of maritime situational awareness and paves the way for the development of more complex reconstruction methods.

III. METHODS

This section describes the methodology for developing the proposed framework for MST. We divide our contributions into three stages that will be discussed subsequently: Initialization, processing, and visualization. Each stage leverages modular, high-performance algorithms designed for interchangeability with existing data processing workflows, with an emphasis on GPU-accelerated computation. The reconstruction process itself offers statistical methodologies and filtering techniques that support automatic differentiation, facilitating seamless integration with contemporary machine-learning frameworks.

A. Initialization

Utilizing naturally occurring muons leads to sparse and noisy statistics. Due to the complex scattering processes that muons undergo inside materials, the reconstruction of the original muon path is a high-dimensional problem. It is, therefore, crucial to initialize any optimization algorithm with a well-informed prior to reduce computational load and help convergence. Especially for the

EM algorithm, that iteratively optimizes a non-convex likelihood function without a closed-form solution,[5] the initial parameter choice can significantly influence convergence behavior.

After recording enough muon tracks, the reconstruction begins by discretizing the entire volume of interest (VOI) into a regular $N_x \times N_y \times N_z$ voxel grid. Each voxel is initialized with an initial scattering density estimate that is either based on user-defined values or the output of a geometry-based reconstruction method. Schultz *et al.*[5] show that the initialization with PoCA improves convergence.

To allow users to benchmark different reconstruction approaches, we implement two geometry-based methods, PoCA and ASR. Both versions are written in PyTorch and can, therefore, make use of its automatic differentiation engine and GPU acceleration. These fast implementations enable experimentation with different initialization of the EM optimization procedure, where PoCA may serve either as a standalone reconstruction or as an initialization prior that aids EM convergence. Reconstruction with PoCA can be solved analytically and in parallel for all muon events: After converting the muon tracks into 3D PoCA-positions and subsequently into voxel indices, the per-voxel scattering statistics are reduced into a single per-voxel scattering value. For that, calculating either the average, minimum, or maximum scattering value is currently supported. Saving all values per voxel without reducing them, to allow for subsequent quantile or median filtering, is also supported, but only sequentially. For ASR, we extend this implementation by first computing the angles between the incoming and outgoing muon tracks. Since ASR works over a neighborhood, defined by a user-controllable factor (see Stapleton *et al.*[4]), we first compute a sparse distance matrix that acts as a mask to include relevant voxels. Columns correspond to muon events, while rows correspond to voxels. Next, we compute the total contribution per voxel by multiplying the rows of the mask with the scattering angles for all voxels and collapse the rows using another average reduction. Due to the inclusion of statistics over a neighborhood, ASR reconstruction yields a smoother map that mitigates the reconstruction noise. Note that both PoCA and ASR can also be used as a standalone reconstruction algorithm within the provided framework.

Both algorithms are implemented via PyTorch, which allows access to high-performance GPU-accelerated calculations via CUDA. Furthermore, calculations are performed batch-wise, utilizing parallelism to efficiently populate millions of voxels.

B. Reconstruction and filtering

After initialization, the populated voxel volume can be optimized using our novel EM implementation and additionally post-processed using volumetric filters to reduce noise. A core component of the original EM algorithm is the computation of a path response matrix[12] that weighs each voxel in a muon track. Computing this matrix in a dense way leads to the worst-case memory complexity of $\mathcal{O}(MN)$, as described in Sec. II. To allow parallelization on the GPU and handle large scenes efficiently, we implemented the computation of the path response matrix using a custom C++ back-end, which is accessed from the Python code via the pybind11 library.[13] Implementing the numerically expensive voxel path tracing in C++ allows for compilation-time optimizations of the code and trivial multi-threading, severely improving the execution speed. In essence, the tracing algorithm is based on the Hierarchical Digital Differential Analyzer (HDDA) algorithm,[14] widely used in computer graphics to traverse volumetric structures. The implementation supports straight-line muon tracing, connecting muon entry and exit points of the VOI directly, as well as single scattering (as it is computed with PoCA), where the muon traverses a piecewise-linear trajectory. Due to the modular structure of the framework, arbitrary tracing functions may be implemented in the future, such as splines or more complex primitives[15] approximated by piecewise linear rays that are chained together and traced with the HDAA. The implemented EM algorithm also supports the addition of muon momentum estimation by assigning a muon momentum directly to each event to aid convergence.

The estimated, sparse path response matrix is subsequently used to perform parallel optimization on the GPU in PyTorch, minimizing execution times per iteration and allowing scalability. Since the EM algorithm requires the constant computation of matrix inverses, we also employ several techniques proposed in the implementation by Riggi *et al.*[6] to improve numerical stability as well as to condition the EM for larger problems and regularize outliers.

Additionally, optimized, learnable volumetric filters, such as Gaussian smoothing, median filtering, or 3D convolution functions, are integrated into the framework. These functions play a crucial role in post-processing and regularization, improving the overall quality of the reconstructed data by reducing noise and enhancing feature resolution. This filtering approach ensures that the reconstructed images are both accurate and interpretable, simplifying practical applications in maritime security and other scientific domains.

C. Visualization

To allow interactive data exploration, we have implemented a web-based visualization front-end in JavaScript that renders volumetric MST reconstructions in real-time, as shown in Fig. 1. The interface provides fine-grained controls to enhance the inspection of the reconstructed volume, including voxel grid opacity, point cloud size, density minimum and maximum thresholds, and color map selection. These controls enable users to filter, rescale, and contrast-enhance the 3D scattering density maps in real time. We further support ground-truth geometries imported via the Blender-to-Geant4 (B2G4) framework[16] or single 3D mesh models, as they can be overlaid as semi-transparent meshes, enabling pixel-accurate, side-by-side comparisons between reconstructed volumes and expected objects. For more advanced methods that employ supervised data and thus offer a ground-truth, this tool can offer a better understanding of the reconstructed output and cross-check the impact of metrics on the visual quality of a reconstruction. Especially for the application of MST in border security deployments, these features allow operators to precisely inspect the results and enhance the detection capabilities of security inspection operations.

IV. SIMULATION SETUP

A. Dataset creation

In order to evaluate the effectiveness and scalability of our framework, we have conducted a series of extensive studies on the

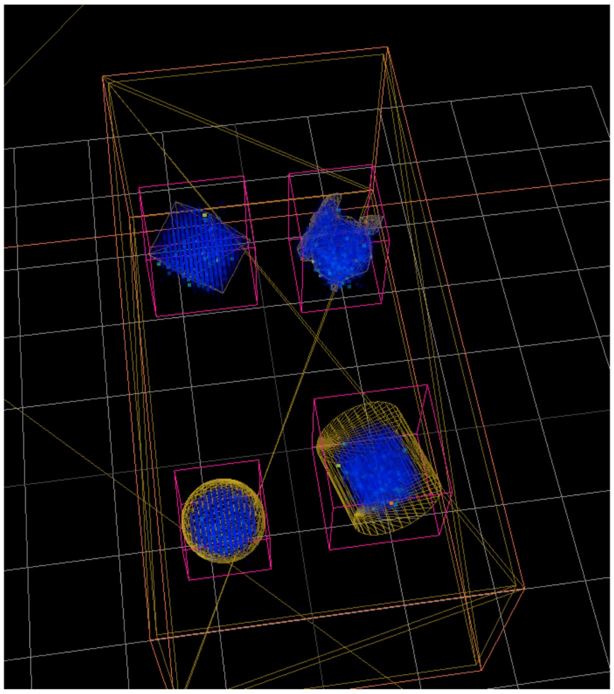

FIG. 1. Exemplary reconstruction results displayed in the visualization front-end provided with the framework. Displayed is a simple, simulated scene with reconstructed points (blue) as well as original 3D meshes (yellow) and bounding boxes (pink) to aid the understanding of the quality of the reconstruction.

described implementations of PoCA, ASR, and the custom EM algorithm. A set of synthetic scenes with increasing geometric complexity and diverse material compositions was generated in Geant4[17] under ideal detector conditions. Scene construction followed a two-step workflow: (i) individual cubes, nested cubes, and elongated cubes were arranged using B2G4 and (ii) the resulting 3D polygonal geometries were exported directly into Geant4 for Monte Carlo particle simulation. Muon tracks were sampled from the Cosmic-ray Shower Library (CRY)[18] with a fixed number of 1.5×10^7 simulated events to ensure fair algorithmic comparison.

Figure 2 illustrates the nine simulation configurations: Six simple configurations each contain one of the following test materials: Iridium (Ir), sodium (Na), lead (Pb), tantalum (Ta), yttrium (Y), and zinc (Zn). These were chosen to give a broad spectrum of varying material densities to estimate algorithmic performance on a varying grade of sparsity. To approach more realistic scenarios, three scenes with additional complexity are used: These scenes deliberately pair materials of comparable density but contrasting radiation length (SameX_0), nest different densities (ASR-Test) to challenge the reconstruction methods, and test spatial resolution (ThreeCubes). The background of all the scenes is set as air. The world size for all scenes is set to $2 \times 2 \times 2\mathrm{m}^3$ centered around the origin, leading to grid sizes between 10^3 and 400^3 voxels.

B. Performance metrics and evaluation

We assess reconstruction quality across our implementations of PoCA, ASR, and EM while evaluating scalability and computational efficiency of our novel sparse and parallel EM implementation. Simulated experiments were performed on the nine Geant4 scenes described in Sec. IV A. For quantitative comparison, we employ three standard image quality metrics: the Peak Signal-to-Noise Ratio (PSNR), the Structural Similarity Index Measure (SSIM),[19] and the Learned Perceptual Image Patch Similarity (LPIPS).[20] Together, these complementary metrics offer a quantitative view of reconstruction accuracy, structural integrity, and perceptual quality. Estimated densities are normalized with respect to the maximum estimated value to allow comparison with the ground-truth. For normalized, estimated density per voxel i_1 and corresponding ground truth i_2, the PSNR is computed as

$$\mathrm{PSNR}(i_1, i_2) = 10 \cdot \log_{10}\left(\frac{L^2}{\mathrm{MSE}(i_1, i_2)}\right), \tag{1}$$

with MSE as the mean squared error between estimated reconstruction i_1 and ground-truth i_2, and L as the maximum possible voxel value (meaning the highest value in the respective ground-truth). Note that the PSNR quantifies the voxel-wise error relative to the ground-truth scattering density map, with higher PSNR values corresponding to fewer voxel-wise errors. The SSIM is defined as

$$\mathrm{SSIM}(i_1, i_2) = \frac{(2\mu_{i_1}\mu_{i_2} + C_1)(2\sigma_{i_1 i_2} + C_2)}{\left(\mu_{i_1}^2 + \mu_{i_2}^2 + C_1\right)\left(\sigma_{i_1}^2 + \sigma_{i_2}^2 + C_2\right)}, \tag{2}$$

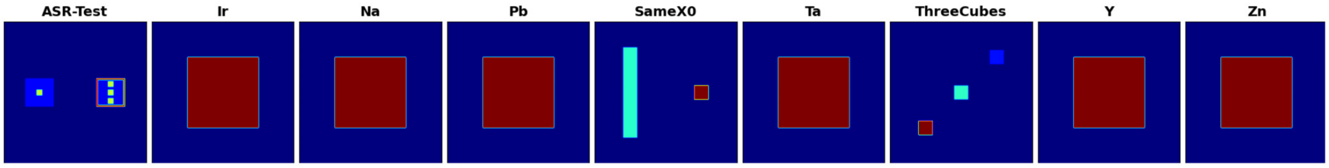

FIG. 2. Ground-truth of the XZ-projection for the different scenarios, where normalized density is mapped using a color map, representing different materials. For the ASR-Test, the borders of the two small boxes are made of iron and are filled with water, while the cubes within these boxes are made of tungsten. For SameX_0, the object on the left is made of titanium, while the object on the right is made of lead. For ThreeCubes, the cubes are made of tungsten (bottom-left), iron (middle), and aluminum (top-right). For all scenarios, the background is air. Best viewed digitally.

where μ_1 and μ_2 denote the local means of i_1 and i_2, σ_1^2 and σ_2^2 represent their local variances (contrast), and σ_{12} is the local covariance that captures shared structural information. The stabilizing constants $C_1 = (k_1 L)^2$ and $C_2 = (k_2 L)^2$, with $k_1 = 0.01$ and $k_2 = 0.03$ used here, prevent division by zero when μ_1, μ_2, σ_1^2, or σ_2^2 are very small. By jointly evaluating the local mean, contrast, and structure, SSIM offers a perceptually aligned measure of reconstruction quality that is especially valuable for assessing MST results beyond simple voxel-wise error metrics. We complement PSNR and SSIM with LPIPS, which compares deep-network feature maps instead of raw voxel values, as given by

$$\mathrm{LPIPS}(i_1, i_2) = \sum_{\ell} w_\ell \, \| \phi_\ell(i_1) - \phi_\ell(i_2) \|_2^2, \tag{3}$$

with the feature-map (activation tensor) $\phi_\ell(\cdot)$ extracted from layer ℓ when the input is forwarded through the network.[20] Note that these standard neural networks encode rich, hierarchical visual feature vectors, including edges, textures, and object parts that correlate well with human perception, with lower LPIPS scores indicating that i_1 is perceptually closer to i_2. Since all the metrics work on 2D instead of 3D data, we slice the 3D volume along the Y axis according to the given grid resolution and average over the slices.

In addition to evaluating image quality, we conduct a scalability study for the EM algorithm to measure execution time, memory consumption, and voxel count as a function of input size. This analysis is crucial for ensuring that our approach remains feasible for large-scale real-world applications. Furthermore, we conduct a multi-resolution study to investigate the effect of varying resolutions on convergence behavior. This qualitative evaluation allows us to assess the trade-offs between the computational cost and reconstruction accuracy, providing insights into optimal parameter configurations for various use cases.

V. RESULTS

A. Quantitative comparison

This section presents a comparative analysis of the different algorithms across all simulated scenes. Figure 3 displays the results for ASR, EM, and PoCA (from top to bottom), with each column, from left to right, representing one of the simulated scenes detailed in Sec. IV A. For each algorithm and scene, the corresponding performance metrics are shown below; the boldface values beneath each image mark the best score for that scene.

In numerical terms, the EM reconstruction achieves very high perceptual fidelity across all scenes, attaining the top SSIM score in eight scenes and the lowest LPIPS score in the same eight scenes, reaching a minimum of 0.23 in the ThreeCubes scene. Although its PSNR score is only outright best in the high-Z cases (Ir, Pb, and Ta), it stays close to PoCA elsewhere, for instance, 22.40–22.83 dB in the ASR-Test scene and 26.70–27.77 dB in the ThreeCubes scene.

While the standard reconstruction methods achieve good results, we observe that their perceptual quality does not match

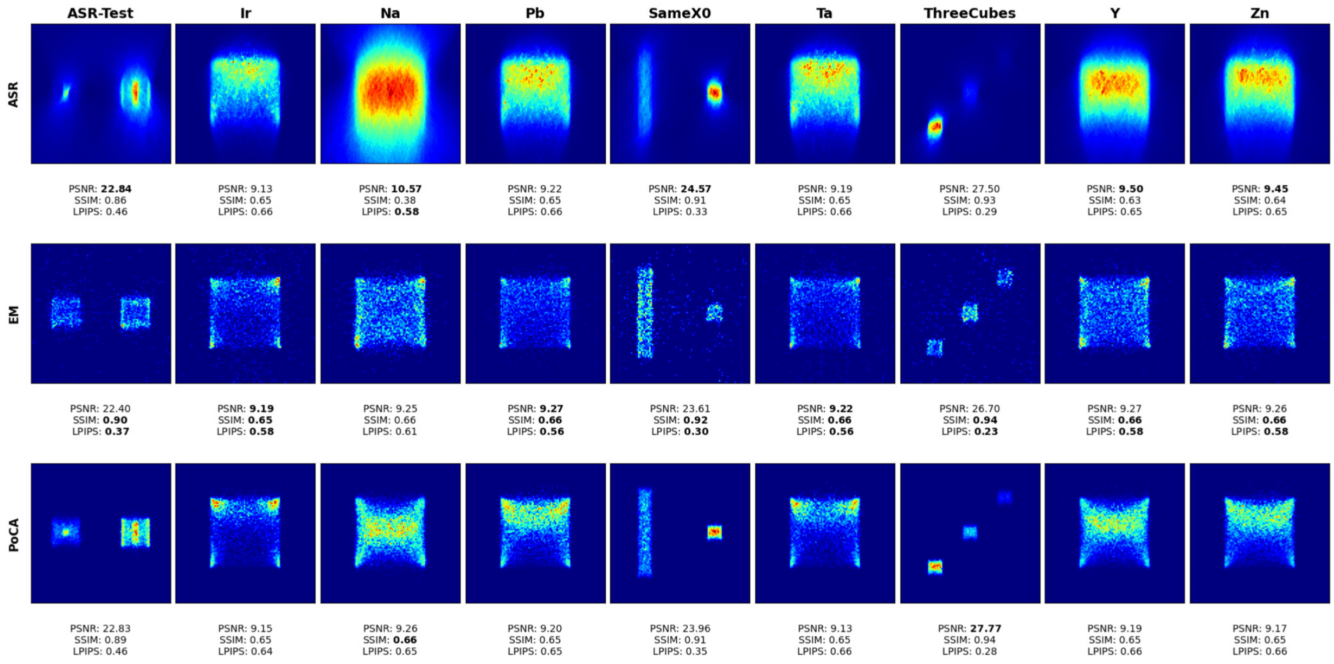

FIG. 3. Comparison of the reconstruction results for the different algorithms. The results are arranged row-wise for ASR, EM, and PoCA (from top to bottom) and column-wise by test scene (from left to right). Each image is accompanied by the corresponding performance metrics below, where boldface values indicate the best-performing algorithm for each scene. For comparison with the ground-truth, see also Fig. 2. Best viewed digitally.

TABLE I. Benchmarking results of the EM algorithm, performed with different voxel sizes. The simulation was conducted with ≈3.7×10^5 hits, 1000 iterations, and has been initialized with PoCA.

Voxel size (cm)	20	10	5	2	1	0.5
Voxel count	1×10^3	8×10^3	64×10^4	1×10^6	8×10^6	64×10^6
Weight matrix (MB)	82	182	387	1010	2054	4147
Initialization time (s)	1.2	2.2	4.7	11.1	22.5	44.3
Avg. time per iteration (s)	0.15	0.15	0.19	0.4	0.8	1.6

that of the EM implementation, and their geometric assumptions are reflected in the metrics. PoCA highlights the edges and, thanks to this, achieves the highest PSNR scores in high-contrast settings: 27.77 dB in the ThreeCubes scene and 23.96 dB in the Same X_0 scene. However, this enhancement introduces artifacts that raise its LPIPS score in other scenes compared to EM algorithm results. Furthermore, ASR produces smooth maps that improve the PSNR score in low-density materials, for instance, 10.57 dB in Na and 9.50 dB in Y. However, this smoothing degrades the structure, as evidenced by the SSIM score dropping to 0.38 for the Na scene, as well as ASR failing to fully resolve the ASR-Test scene.

A noticeable effect beyond the metrics that can be derived from Fig. 3 is that the geometrical assumptions of each algorithm are visible. The PoCA assumption of a single scattering point introduces high-frequency artifacts. It is unable to fully capture the shape of the cube, concentrating on areas where the dispersion is greater. ASR, by contrast, distributes the inferred scattering continuously along the straight muon path, averaging the signal and yielding a noticeably smoother density map. However, this introduces a characteristic “shadow” visible in all reconstructions. The EM algorithm employs a likelihood-based statistical model that iteratively refines voxel densities to maximize the joint probability of all muon trajectories; this update suppresses counting noise and

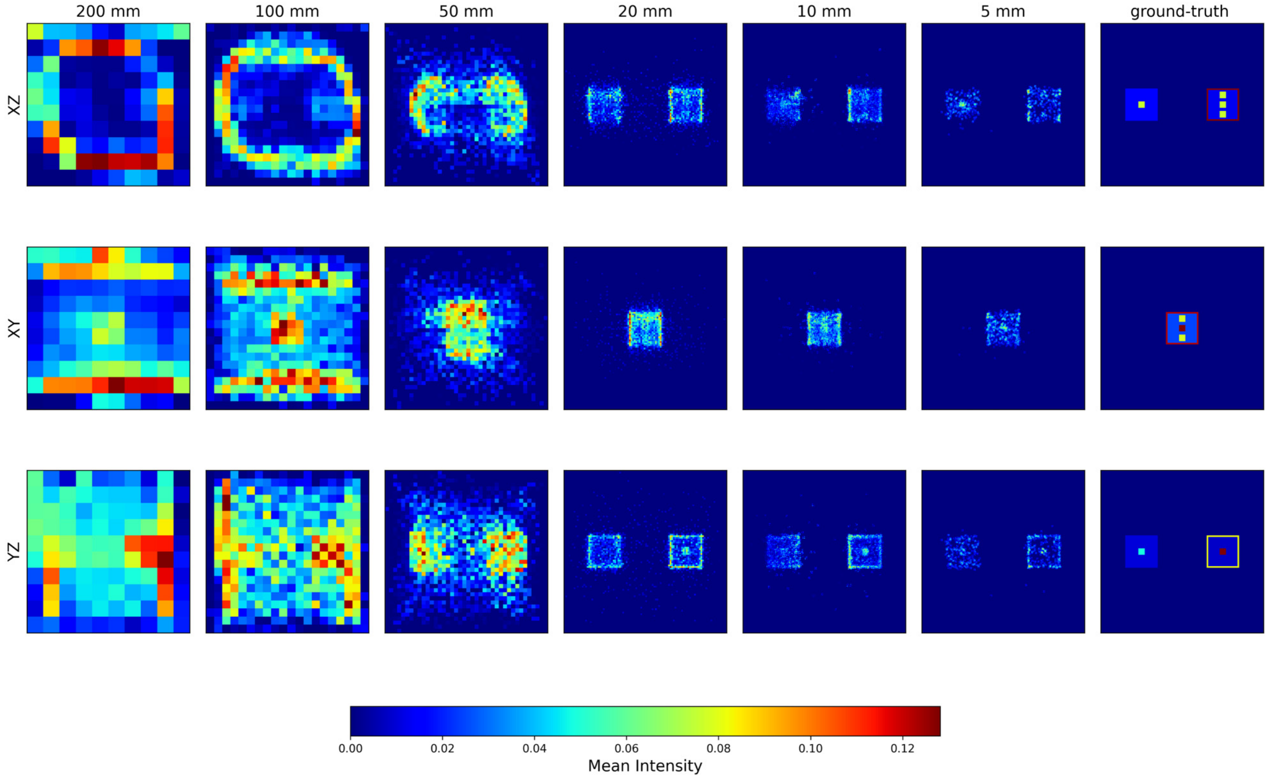

FIG. 4. Reconstruction results of the EM algorithm for the ASR-Test scene for different voxel sizes. The first six columns represent the results for the different voxel sizes, as indicated on the top, while the last column shows the ground-truth. The three rows represent the different projections of the scene, i.e., the XZ-projection, the XY-projection, and the YZ-projection.

compensates for path overlap. As a result, EM preserves shape and contour, resulting in high SSIM scores and low LPIPS scores. However, the same statistical filtering attenuates local amplitude variations; hence, its PSNR score is not always the maximum: it sacrifices some point accuracy to gain structural fidelity. An example of this can be seen in the SameX_0 and ASR-Test scenes where visually PoCA seems to deliver a cleaner result. When comparing this to the EM, it can be seen that despite higher noise in the EM, the density is more uniformly distributed without the "hot center" (the red concentration inside the solid), which, in downstream tasks, such as material classification, would lead to errors. With better initialization or pre-learned priors, these effects can be mitigated—however, these efforts are beyond the scope of this work but will be a topic for future studies.

B. Scalability

In this section, we present a scalability study of the implemented EM algorithm, examining its performance with varying voxel sizes. Table I presents the benchmarking results for the EM algorithm obtained with varying voxel resolutions, from 20 cm to 5 mm, using 1000 iterations and initialized with the PoCA reconstruction. We keep the number of muon events constant across all resolutions. By halving the voxel size, the number of voxels increases by a factor of eight (cubic law), which means that we look at scenarios ranging from 1×10^3 voxels up to 64×10^6 voxels. In parallel, the weight matrix, which stores the geometric information from the voxel-tracing for every muon event and every voxel, grows from 82 MB to 4.1 GB. The initialization time, dominated by the construction of the weight matrix, scales from 1.2 to 44.3 s. The average time per iteration also increases, but more moderately, thanks to GPU parallelism: from 0.15 s at 20 cm to 1.6 s at 5 mm. Overall, the table indicates that $1-2$ cm resolutions remain computationally affordable, with a consumption of $\approx$1 GB of memory and times per iteration of less than 1 s, whereas going down to 5 mm requires several BG of memory and significantly longer time for initialization. However, such voxel sizes are only required for use cases where millimeter localization is critical. Nevertheless, the proposed workflow demonstrates that while the voxel counts increase exponentially, the time and memory complexity only increase linearly by a factor of $\approx$2.

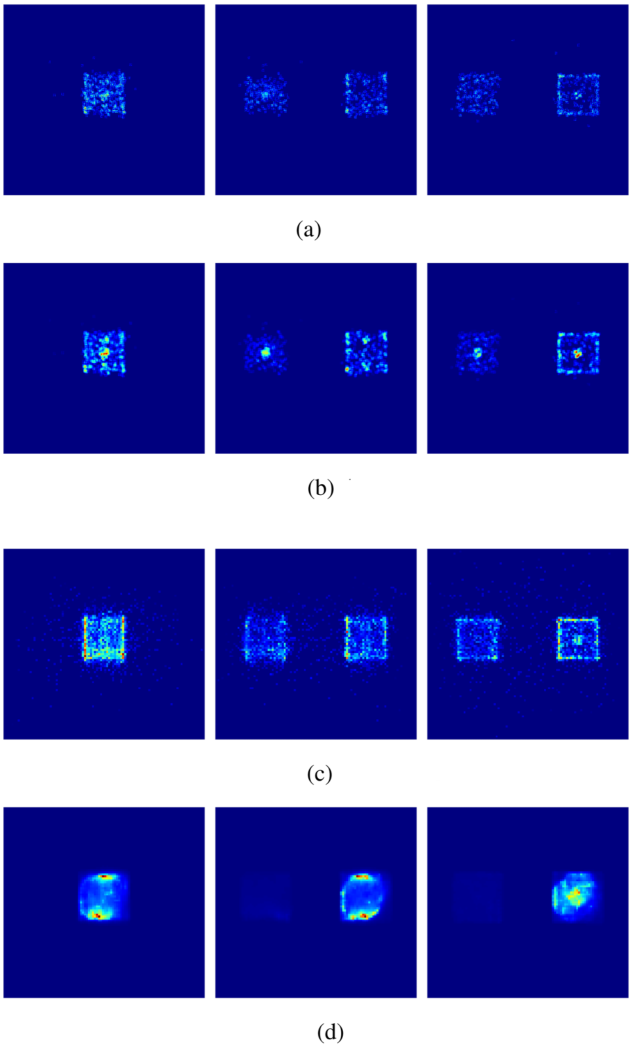
(a)
(b)
(c)
(d)

FIG. 5. Four different reconstructions using the EM implementation of the same ASR-Test scene used previously in Fig. 4. Columns show the XZ-, XY-, and YZ-projections, respectively. Two different voxel sizes have been tested, each with and without pre-processing the PoCA points with a 3D volumetric Gaussian kernel. Colors indicate normalized density with all four scenes normalized to their respective maximum value. (a) 5 mm voxel size, PoCA initialized; (b) 5 mm voxel size, PoCA initialized, pre-processed with Gaussian kenrel; (c) 20 mm voxel size, PoCA initialized; and (d) 20 mm voxel size, PoCA initialized, pre-processed with Gaussian kernel.

C. Effects of voxel resolution

Figure 4 shows the reconstruction results of the EM algorithm for the ASR-Test scene, but for different voxel sizes varying between 5 and 200 mm. Note that for all voxel sizes, the number of iterations remains identical, which allows us to draw conclusions about the convergence behavior of the algorithm. While the voxel size of 200 mm represents the extreme case of using a few voxels with a large muon-per-voxel statistic, the 5 mm case represents the opposite, i.e., a large number of voxels with a small muon-per-voxel statistic. We do not perform any form of averaging or neighborhood sampling but reconstruct with the statistics available. At 200 and 100 mm, the sampling is so coarse that the objects are reduced to low-frequency spots, and the central cavity of the cube is barely detectable. At 50 mm, the general contours of the cubes appear; yet, the walls remain discontinuous. The interior, albeit hollow, is rather noisy, and the concealed structures are not entirely resolvable. A voxel resolution of 20 mm is decisive: The four walls and the inner cubes are clearly distinguishable, although random grain persists. Moving to a resolution of 10 mm sharpens the edges, and noise decreases. Further refinement to a 5 mm resolution appears to provide no additional information; instead, details get lost due to the lack of muon-per-voxel statistics and additional sparsity. At a resolution of 10 mm, the geometry of the cubes becomes reasonably well captured, though noticeable deviations from the ground truth remain visible. Further refinement to 5 mm does not yield substantial improvements due to limited muon-per-voxel statistics. Thus, voxel sizes around 10 mm represent a practical balance between fidelity and computational efficiency. Millimeter-scale resolutions are only justified when sub-centimeter localization is explicitly required, in combination with a statistical count sufficient to maintain an acceptable signal-to-noise ratio.

Another interesting effect of voxel resolution can be seen in Fig. 5. Shown are the two previously described reconstructions of the ASR-Test scene performed with our EM implementation at 5 and 20 mm resolution. The volume is initialized with PoCA points for faster convergence. In addition, a $3 \times 3 \times 3$ Gaussian filter kernel is applied during initialization for both scenes, effectively distributing the PoCA values over a neighborhood. It is visible that at a resolution of 5 mm, sparsity of the initial PoCA points is reduced as values are distributed over a neighborhood by the application of the filter, leading to a qualitatively better and more refined reconstruction. Moving to 20 mm resolution, it becomes clear that the same filtering biases the initialization and the reconstruction becomes overly blurred. This ties in with the previous observations that a meaningful reconstruction is closely related to the initial distribution and that a balance between sparsity and variance per voxel is important.

VI. CONCLUSION AND FUTURE WORK

This study introduces an end-to-end reconstruction framework for muon-scattering tomography (MST), including a highly optimized EM solver, which provides geometric accuracy, perceptual quality, and reasonable requirements for computational resources. It builds a critical foundation for practical, near real-time MST in maritime and border security applications. We have demonstrated that it is possible to achieve both high geometric fidelity and operational efficiency within realistic hardware constraints. This aligns closely with the increasing demand for automated, scalable, and accurate monitoring systems as outlined earlier.

Our implementation of the EM algorithm features a custom back-end for fast muon-path tracing within the VOI. In maritime environments, where large and often complex structures need to be continuously monitored, the ability to perform high-resolution 3D reconstruction in near real-time is crucial. The modularity of our PyTorch-based framework supports seamless integration with existing data acquisition and simulation pipelines, including B2G4 and Geant4. Standard reconstruction methods, such as PoCA, ASR, and EM, can be freely interchanged and incorporated. As all implemented algorithms support automatic differentiation and GPU acceleration, the combination of our framework with novel machine-learning pipelines is made possible.

A set of test scenes with varying levels of complexity and materials has been created in B2G4 and simulated using Geant4. The EM implementation has been benchmarked against PoCA and ASR: Our implementation of EM reduces per-iteration latency to 0.8 s for a voxel size of 1 cm while keeping memory footprints within 1 GB. Even at 5 mm resolution, it stays below 1.6 s per iteration with a memory consumption lower than 5 GB. Furthermore, the integrated EM solver maintains superior perceptual fidelity compared to PoCA and ASR reconstructions, as shown by comparing the results using PSNR, SSIM, and LPIPS scores as metrics.

Future work will focus on developing better priors to incorporate pre-learned, non-linear information to improve the initialization of the algorithm. By using PyTorch and its automatic differentiation engine to implement the algorithm, we can replace the formulation of expectation–maximization with other optimization procedures. Furthermore, our implementation can be used in end-to-end learning pipelines where the EM can be used as an intermediate step between input data and downstream tasks, such as material classification, without breaking the automatic differentiation graph. This enables complex learning tasks through the EM implementation. To allow for even more complex scenes with larger voxel counts, adaptive resolution using hierarchical spatial data structures can be explored. This will simultaneously improve execution time too. Scalability and robustness require leveraging synthetic data pipelines (e.g., B2G4) that mimic real acquisition conditions to enable large-scale, noise-aware reconstruction studies and the development of methods that isolate relevant signals from background contributions. In a next step, real-world experiments must be conducted to evaluate noise behavior, robustness to detector alignment, and environmental uncertainty. To generate a better understanding of the generalization of our framework, we participate in an ongoing research effort focusing on monitoring nuclear materials. We hope that the public release of our code will also motivate further generalization in the academic community.

In conclusion, this work establishes a practical and extensible baseline for near real-time MST, paving the way for intelligent security systems in maritime and border environments.

ACKNOWLEDGMENTS

This research was funded by the SilentBorder project under Grant Agreement ID 101021812 of the European Union's Horizon 2020 research and innovation program.

We would like to acknowledge a parallel research effort creating open-source tools for muon-scattering tomography to which this work is closely linked: https://github.com/maximelagrange/muograph.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

F. A. Sattler: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **J. M. Alameddine:** Formal analysis (equal); Methodology (equal); Software (equal); Writing – review & editing (equal). **Á Bueno Rodríguez:** Conceptualization (equal); Formal analysis (equal); Methodology (equal); Writing –

review & editing (equal). **M. Stephan:** Supervision (equal); Validation (equal); Writing – review & editing (equal). **S. Barnes:** Funding acquisition (equal); Supervision (equal); Validation (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request. An implementation of the framework is openly available in GitHub at https://github.com/DLR-MI/MuonEM, Ref. 21.

APPENDIX: HIGH-RESOLUTION DATA ANALYSIS

For a better understanding, we provide our results and ground-truth combined into a single high-resolution graphic in Fig. 6.

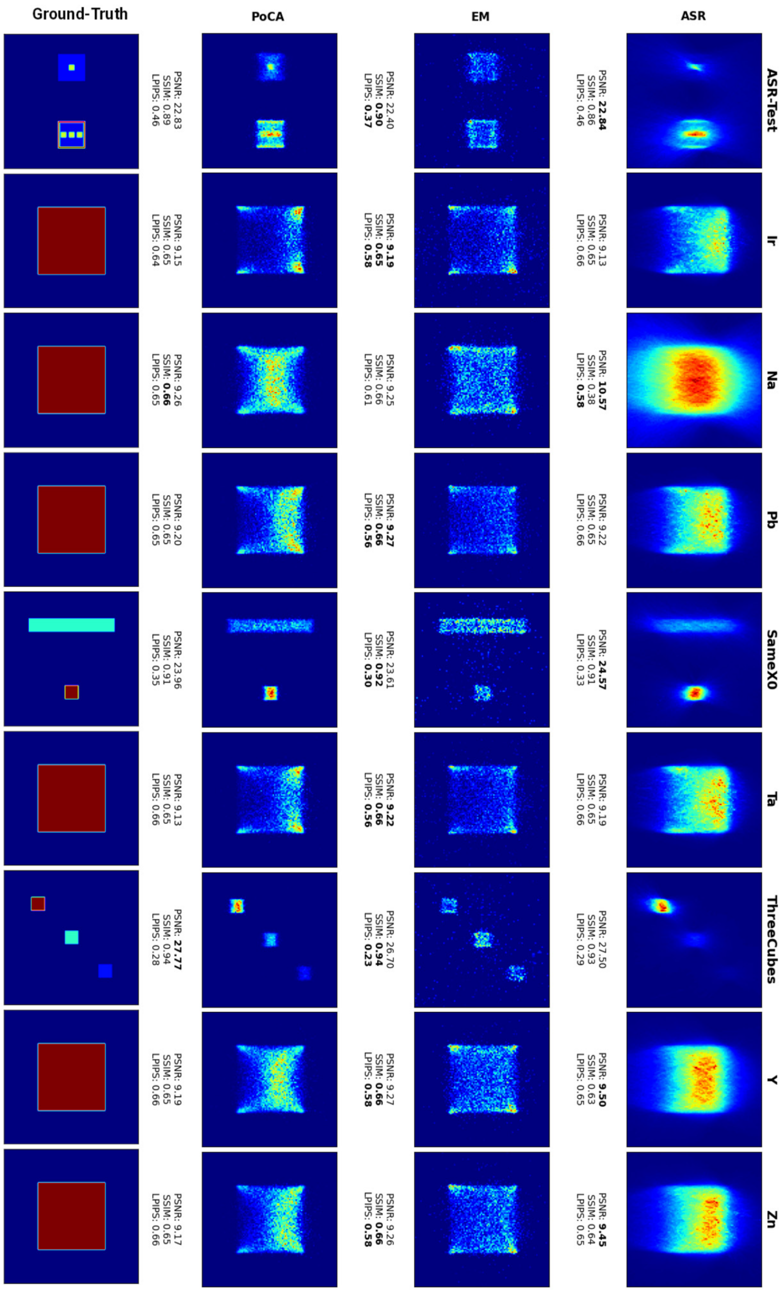

FIG. 6. Joined ground-truth and comparison of reconstruction results for the different algorithms. The results are arranged row-wise for Ground-truth, ASR, EM, and PoCA (from top to bottom) and column-wise by a test scene (from left to right). Each image is accompanied by the corresponding performance metrics below, where boldface values indicate the best-performing algorithm for each scene. See also Fig. 3.

REFERENCES

[1] K. Borozdin, G. Hogan, C. Morris, W. Priedhorsky, A. Saunders, L. Schultz, and M. Teasdale, "Radiographic imaging with cosmic-ray muons," Nature **422**, 277 (2003).

[2] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (Curran Associates, Inc., Red Hook, NY, 2019).

[3] L. Schultz, "Cosmic ray muon radiography," Ph.D. thesis (Portland State University, 2003).

[4] M. Stapleton, J. Burns, S. Quillin, and C. Steer, "Angle statistics reconstruction: A robust reconstruction algorithm for muon scattering tomography," J. Instrum. **9**, P11019 (2014).

[5] L. J. Schultz, G. S. Blanpied, K. N. Borozdin, A. M. Fraser, N. W. Hengartner, A. V. Klimenko, C. L. Morris, C. Orum, and M. J. Sossong, "Statistical reconstruction for cosmic ray muon tomography," IEEE Trans. Image Process. **16**, 1985–1993 (2007).

[6] S. Riggi, V. Antonuccio-Delogu, M. Bandieramonte, U. Becciani, A. Costa, P. La Rocca, P. Massimino, C. Petta, C. Pistagna, F. Riggi, E. Sciacca, and F. Vitello, "Muon tomography imaging algorithms for nuclear threat detection inside large volume containers with the muon portal detector," Nucl. Instrum. Methods Phys. Res. A **728**, 59–68 (2013).

[7] S. Barnes, A. Georgadze, A. Giammanco, M. Kiisk, V. Kudryavtsev, M. Lagrange, and O. Pinto, "Cosmic-ray tomography for border security," Instruments **7**, 13 (2023).

[8] B. Lefevre, J. Vogel, H. Gomez, D. Attié, L. Gallego, P. Gonzales, B. Lesage, P. Mas, and D. Pomarède, "3D reconstruction of a nuclear reactor by muon tomography: Structure validation and anomaly detection," PRX Energy **4**, 013002 (2025).

[9] L. Pezzotti, D. Cifarelli, D. Corradetti, J. P. Costa, G. Gabrielli, L. Galante, A. Gallerati, I. Gnesi, A. Jouve, and A. Marrani, "A new method for structural diagnostics with muon tomography and deep learning," J. Instrum. **20**, P06034 (2025).

[10] G. C. Strong, M. Lagrange, A. Orio, A. Bordignon, F. Bury, T. Dorigo, A. Giammanco, M. Heikal, J. Kieseler, M. Lamparth *et al.*, "TomOpt: Differential optimisation for task-and constraint-aware design of particle detectors in the context of muon tomography," Mach. Learn.: Sci. Technol. **5**, 035002 (2024).

[11] T. Maier, P. Sanders, and R. Dementiev, "Concurrent hash tables: Fast and general(?)!," ACM Trans. Parallel Comput. **5**, 16 (2019).

[12] G. Wang, L. Schultz, and J. Qi, "Statistical image reconstruction for muon tomography using a Gaussian scale mixture model," IEEE Trans. Nucl. Sci. **56**, 2480–2486 (2009).

[13] W. Jakob, H. Schreiner, J. Rhinelander, R. W. Grosse-Kunstleve, D. Moldovan, I. Smirnov, A. Gokaslan, Y. Jadoul, A. Huebl, B. Staletic, S. Izmailov, E. Cousineau, D. Spicuzza, M. Carlstrom, B. Merry, b pass, A. Lee, S. Corlay, L. A. Burns, Dan, X. Pan, bennorth, T. Houliston, S. Lyskov, R. Haschke, jbarlow, gentlegiantJGC, and M. Šimáček (2025). "pybind/pybind11: Version 3.0.1," Zenodo. https://doi.org/10.5281/zenodo.16929811

[14] K. Museth, "Hierarchical digital differential analyzer for efficient ray-marching in OpenVDB," in *ACM SIGGRAPH 2014 Talks, SIGGRAPH '14* (Association for Computing Machinery, New York, 2014).

[15] C. Benton, N. Smith, S. Quillin, and C. Steer, "Most probable trajectory of a muon in a scattering medium, when input and output trajectories are known," Nucl. Instrum. Methods Phys. Res. A **693**, 154–159 (2012).

[16] A. B. Rodriguez, F. Sattler, M. P. Prada, M. Stephan, and S. Barnes, "B2G4: A synthetic data pipeline for the integration of blender models in Geant4 simulation toolkit," J. Adv. Instrum. Sci. **2024**, 476.

[17] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, and G. Barrand *et al.*, "GEANT4—A simulation toolkit," Nucl. Instrum. Methods Phys. Res. A **506**, 250–303 (2003).

[18] C. Hagmann, D. Lange, and D. Wright, "Cosmic-ray shower generator (CRY) for Monte Carlo transport codes," in *2007 IEEE Nuclear Science Symposium Conference Record* (IEEE, 2007), Vol. 2, pp. 1143–1146.

[19] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Trans. Image Process. **13**, 600–612 (2004).

[20] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2018), pp. 586–595.

[21] "MuonEM" (2025). Github repository, available at https://github.com/DLR-MI/MuonEM.