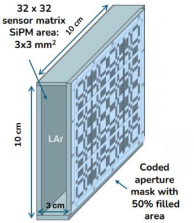
Reconstruction with Coded Aperture masks updates

Valentina Cicero

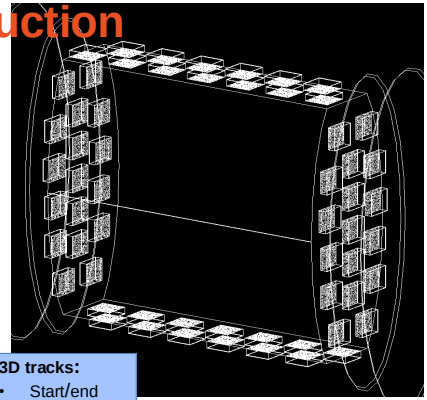
DUNE Italia meeting

29/10/2024

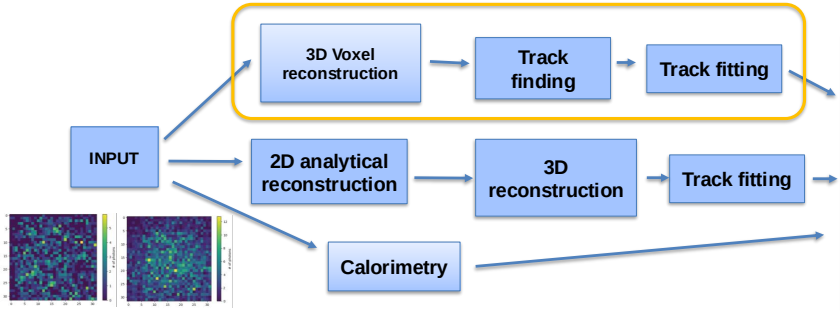
Coded aperture mask system reconstruction



- Coded aperture mask with random pattern, 50 % of filled area
- Camera thickness : 3 cm
- 60 cameras:
  - 17 on each side
  - 13 on top/bottom rows



Reconstruction chain



3D voxel reconstruction algorithm

- Directly reconstructs in 3D the initial photon source distribution in a segmented volume (voxels).
- Maximizes the Likelihood that an initial distribution density can produce the observed data

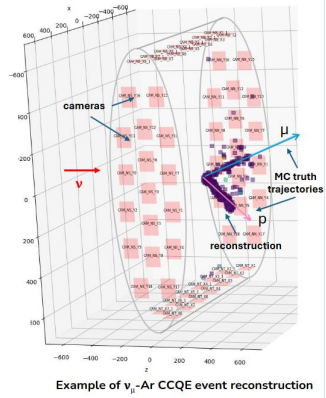
$$\lambda_j^{k+1} = \frac{\lambda_j^k}{\sum_s p(j,s)} \cdot \sum_s \frac{H_s \cdot p(j,s)}{\sum_j p(j,s) \cdot \lambda_j^k}$$

H_s: number of detected photons by sensor s

λ_j unknown photon emission in voxel j

$p(j,s)$: probability of a photon that originated in voxel j is detected by sensor s

k : iteration number

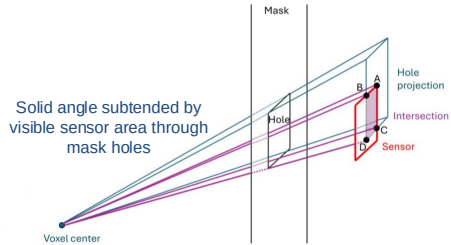


Example of ν_μ-Ar CCQE event reconstruction

*p (j,s) = P geometry (j,s) * P LAr (j,s) * P sensor (s)*

Photon absorption in LAr
Depends on distance between voxel and sensor

SiPM PDE



Updates since last DUNE Italia meeting

Over the past year, we focused on:

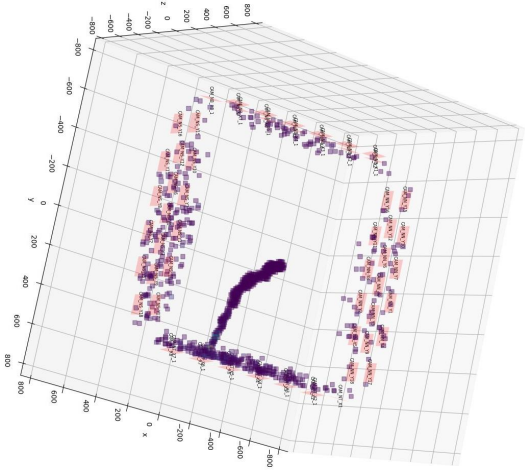
- Reconstruction improvements that did not require a complete rewrite of the software → Corrections to the calculation of weights
- Development of software and algorithms for the analysis of the reconstruction output

Optimization studies to make execution faster are currently less of a priority:

- DUNE software framework is not yet well defined, especially how GPUs will be integrated.

Algorithm weight corrections

- Reconstruction artifacts in voxels close to cameras (d < 6 cm) that decrease the size of the effective fiducial volume for reconstruction.
- Approximation of the solid angle computation is not valid at close distance, which leads to an underestimation of the weights



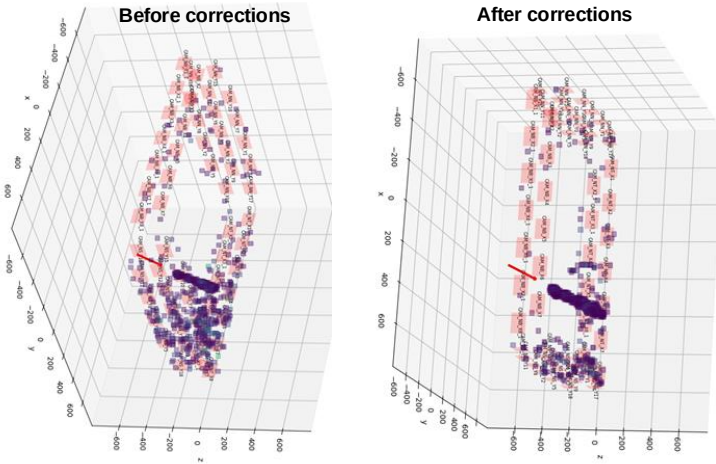
Correction to solid angle computation:

- Subdivide each voxel in 1 mm side mini-voxels
- Compute solid angle subtended by each mini-voxel centers
- Store the average solid angle for a voxel

$$\Omega_{corr} = \frac{1}{N} \sum_{i=1}^{i=N} \Omega_i$$

- Increased computing time, but weights are computed only once per geometry
- Same storage size (~ 120 GB with 60 cameras and 12 mm voxels)

Reconstruction with weight correction

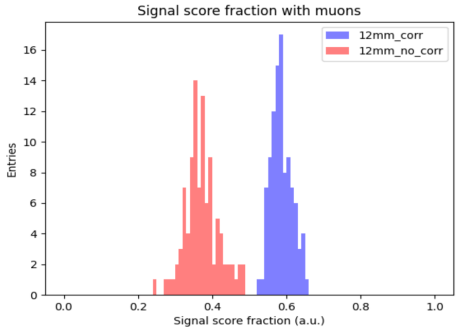


- Example of a reconstructed muon event crossing GRAIN along z
- voxels with score > 200 a.u.

comparison metrics:

$$\frac{\Sigma\lambda_j\ (d \leq R\ )}{\Sigma\lambda_{j'}\ (d > R)}$$

- λ_j voxel score
- d voxel center distance from particle trajectory
- R = 6 cm



F. Mei

Mask reconstruction tools repository

Baltig repository: *https://baltig.infn.it/dune/sand-optical/tools*

Name	Last update
analysis	1 month ago
display	1 month ago
edepsim	3 months ago
gdml_maker	7 months ago

Tools for reconstruction analysis

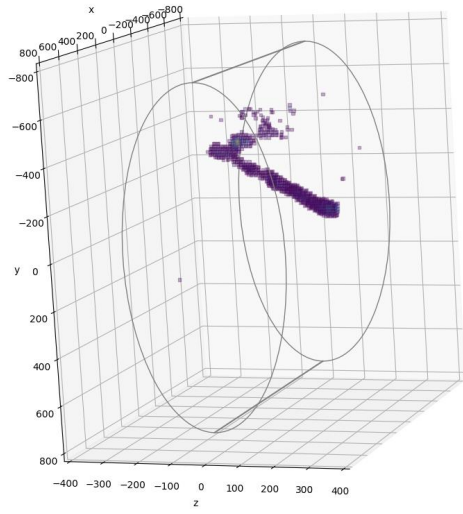
- Plot reconstruction and edeps-sim
- Edep-sim event selection before optical simulation
- Mask camera gdml maker

Written in Python with PyROOT

Reconstruction analysis tools

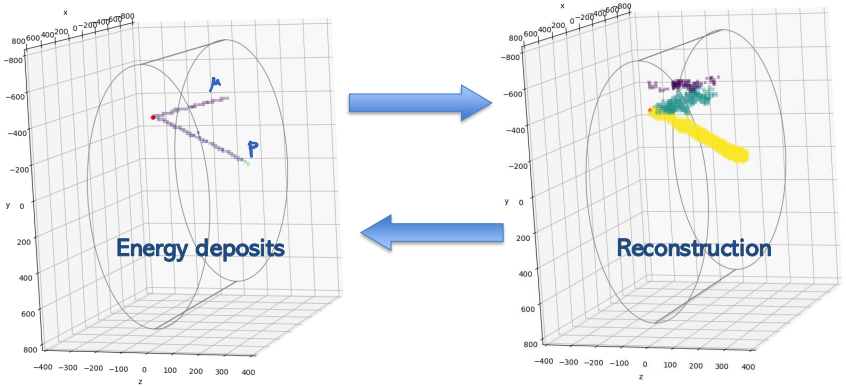
The RecoData class contains GRAIN 3D reconstruction data and geometry information:

- Data as voxels (3D array) or array of points (voxel centers) and ampitudes
- Several methods and algorithms are already implemented:
  - Voxel amplitude transformation: thresholding, histogram equalization
  - clustering algorithms to remove noise
  - Geometrical transformations: Projection to planes, Principal Component Analysis
- Simple visualization in matplolib



MCtruth – reconstruction matching

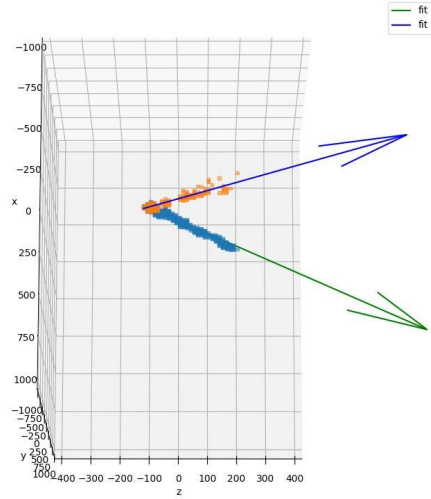
- EDEPosits class for wrapping edepsim output with primaries and trajectory information and a voxelized representation of energy deposits.
- Algorithms to find correspondence between reconstruction and mc-truth (and vice-versa).



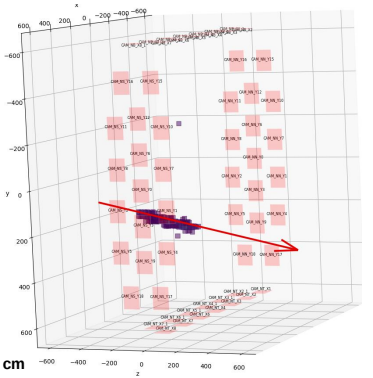
Track finding and fitting

- Implemented a **Hough transform** algorithms for straight lines in 3D
  - Modified using voxel scores as weights in the accumulator
  - Can be improved: low efficiency when a track is short or has a low score

- Simple **linear fit** of voxel blobs based on minimization on sum of squared distance, with voxel scores as weights



Simulated events: muon sample



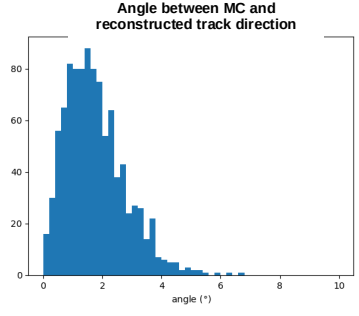
Simulated sample:

- 1k muons crossing GRAIN along z
- Origin position: ([-30, + 30], [-30, + 30], -50) cm
- Direction: θ = [160, 180], φ = [0, 360]
- Energy = (1 ± 0.3) GeV

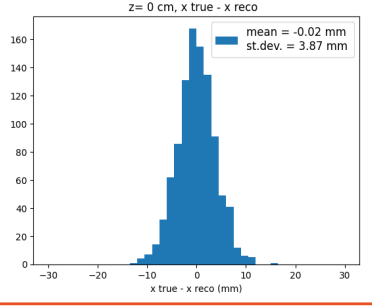
Reconstruction:

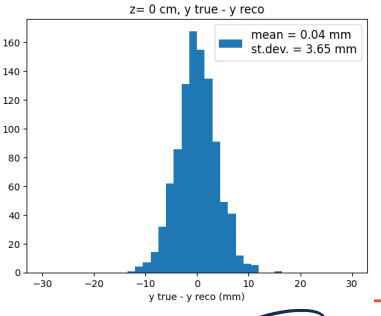
- Voxels size = 18 mm
- Iterations = 200

Analysis using Hough transform and no prior information on the number of tracks



Track position residuals at z = 0 cm





Simulated events: $\nu_\mu - Ar$ CCQES sample

- Events selected with vertex inside reconstruction fiducial volume
- Reconstruction in progress:
  - ~140 events/day
  - ~ 1k reconstructed so far
- use MC truth for voxel clusterization first, evaluate track finding next
- Define some performance figure of merit for a comparison with lenses

Conclusions

This year:

- Corrections to algorithm weights -> improvement of reconstruction
- Focus on building tools for analyzing data
- Simulating CCQES events in GRAIN for TDR

Next steps:

- Evaluate reconstruction performance for GRAIN Physics case
- test extension of MLEM algorithm with Ordered Subsets (OSEM) for faster reconstruction