Chapter 1

Software and Computing

1.1 Overview

This chapter outlines the technical design of the offline computing system and introduces the software that will be used to simulate, reduce and reconstruct ProtoDUNE-SP data.

The data rate and volume will substantial over the relatively short run and this drives many design choices for the offline computing system. The system provides resources necessary for data distribution, processing, and analysis on the grid[1].

All raw data will be saved to tape after being transferred to central CERN and FNAL computing facilities (Section 1.2.2). During this, the metadata and storage locations for all raw data files will be captured in a file catalog system.

A small portion of the data will be immediately processed for data quality monitoring (DQM) purposes (see Section 1.3). The processing steps on the full data sample (see Section 1.4.2) include ADC-level corrections, potential excess-noise filtering, signal processing, reduction, calibration, reconstruction, summarizing and final user analysis. Multiple passes through this chain will be required for final results, as calibrations, algorithms, and summary definitions are expected to evolve. Details on the software framework, event simulation and reconstruction are covered in the final sections.

1.2 Data storage and management system

1.2.1 Data characteristics

It is the TPC data which drives the requirements for the raw ProtoDUNE-SP data. The *ProtoDUNE-SP Data Scenarios* spreadsheet [2] provides details on these above numbers and a few alternative running conditions. Table 1.1 summarizes the nominal these estimates.

Parameter	estimate
In-spill trigger rate	25 Hz
Avg. trigger rate	10 Hz
Channels	15,360
Readout time	5 ms
Compression	4×
Compressed event	60 MByte
Instantaneous rate	1.5 GByte/sec
Average rate	600 MByte/sec
Total triggers	52 M
Total volume	3 PB

Table 1.1: Estimates of nominal raw data parameters driving the design for the raw data storage and management.

The average trigger rate over the entire beam cycle assumes one out-of-spill trigger from the cosmic-μ trigger system is acquired for every in-spill trigger due to the beam. The assumed compression factor can be achieved even with similar levels of excess noise as experience in the first year of MicroBooNE running [**?**]. If no excess noise is experienced, as expected then compression factor of 6 – 8 is expected. A data reduction scheme is described in Section 1.4.1.

1.2.2 Raw data flow

A conceptual diagram of the raw data flow is presented in Figure 1.1. It reflects the central role of the CERN storage service *EOS* in the raw data management scheme. Long-term experience has been gained by the LHC experiments and EOS has proven to be performant and reliable. EOS serves as the staging area from which the data are committed to CASTOR (a hierarchical storage management system developed at CERN) and from which data are transmitted to a number of endpoints including principal data centers such as Fermilab and others. It is also used to provide input to DQM and will be available for personal ad-hoc analyses.

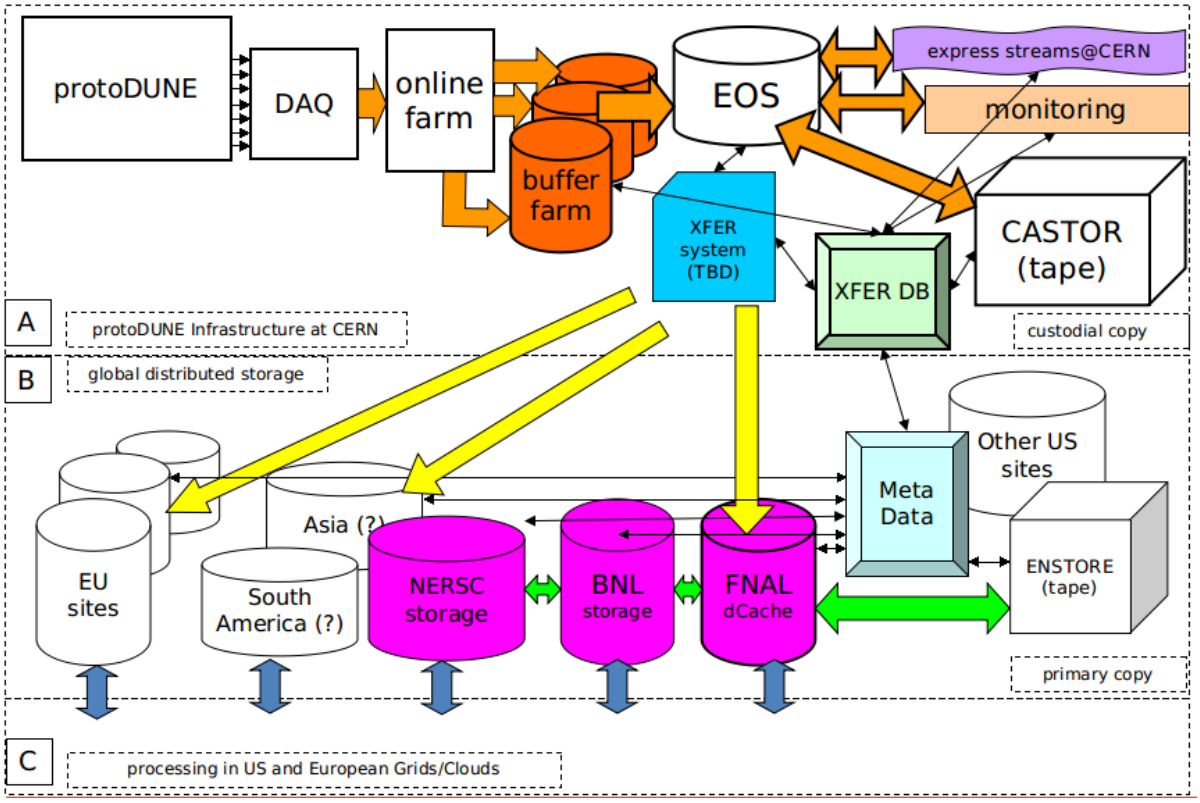

Figure 1.1: Conceptual diagram of the flow of raw data in ProtoDUNE-SP

fig:raw_

1.3 Prompt Processing for Data Quality Monitoring

ocessing

sec:daq_online_monitoring

As described in Section ??, the first point at which data quality monitoring occurs in directly inside the DAQ Online Monitoring (OM). The DAQ computing cluster hardware is relatively high performance and has access to the full, high rate data stream. As such it is ideal for monitoring algorithms which require small amount of CPU and a large fraction of the data.

On the other end of this spectrum, some monitoring algorithms have large CPU requirements but produce meaningful feedback on relatively little data. Running these algorithms on commodity cluster hardware is more cost effective. To manage these jobs a special purpose system called the "protoDUNE prompt processing system" (p3s) is developed. Unlike traditional batch systems it limits maximum latency to provide results at the cost of 100% data throughput. The p3s is portable to many native batch systems, user configurable to allow directed graphs of many jobs to execute in parallel making efficient use of the available hardware resources.

The prompt processing is expected to sample about 1% of the most immediate data just after it is saved by the DAQ and available to the hardware on which it runs. An initial estimate finds that at least 300 dedicated cores will be required to achieve that. This estimate must be refined as a comprehensive list of monitoring algorithms is developed.

The prompt jobs are developed in the familiar form of offline software modules to LArSoft. The processing is expected to include algorithms from signal processing through to full reconstruction.

Due to the sharing of underlying Art framework in both prompt processing jobs and DAQ OM modules it will be easy to migrate algorithms between the two context in the case of computer hardware resource constraints.

1.4 Production processing

-offline

The second major user of the raw data is the production processing. It will make several passes of 100% of the raw data over time as algorithms improve. It has two major stages, a data reduction and an event reconstruction which finally feeds into user analysis.

1.4.1 Data reduction

atareduc

The initial steps of production processing can be factored out into a specific stage in order to allow a vastly reduced data set to be produced which retains all signal information. This data reduction stage consists of about 20% of the total production processing CPU time.

It consists of four major steps. First, it applies any ADC-level corrections and then a software noise filter if excess noise is encountered. These first two steps alone will allow a $6\times -8\times$ compression factor due to reduction of information entropy due to removal of excess noise. The third step applies signal processing. This crucially deconvolves the detector field response functions from the waveforms and identifies the signal regions of interest (signal-ROI). In particular, this allows the otherwise small induction-plane signals to become unipolar and rise above the inherent noise floor. By definition, the signal-ROI selection rejects portions of waveforms which are indistinguishable from noise and thus carry no signal information. It is this selection that provides the bulk of the reduction. Finally, the oversampling of the data is no longer required and is removed while keeping safely below the Nyquist limit and the data is packed, compressed and saved to file. This reduction targets only the TPC data and all other raw fragments are simply copied to the output file

The reduced TPC data is more than 400 times smaller than the raw input. This translates to reducing the 3 PB data described in table 1.1 to only 6.5 TB. This relatively small data set can then be more easily transferred to the remaining, high-CPU part of the production processing chain and it facilitates sharing of the data in general. Independent of the output of the data reduction, the full raw data will be still archived to tap at CERN and FNAL as described in section 1.2.2 so that as data reduction techniques improve they can be rerun and if any special studies require it, they may have access to the full raw waveforms.

tab:goldi

sec:raw_concept

1.4.2 Reconstruction processing

-process

Starting with the signal-ROI there are two basic approaches to reconstruction which are described in detail in the following sections. The first starts with fitting multiple Gaussian distributions to

the waveforms and the second to retaining their binned structure.

1.5 The LArSoft framework for simulation and reconstruction

:larsoft

LArSoft [3] is a suite of tools for simulating and reconstructing data collected from LArTPC detectors. It is built on the *art* [4] event-processing framework. The main features of the *art* framework are its configurability by human-readable and editable control files (that use the Fermilab Hierarchical Control Language (*FHiCL*)), and the scheduling of program module execution. The modules are of five types: event sources, filters, data-product producers, analyzers, and output. Common utilities that can be accessed by any program module at any time are called *services*.

The *art* framework defines the input/output structure of ROOT-formatted files using TTrees to store the data, metadata, and provenance information. The provenance information consists of the contents of the FHiCL documents used to steer the processing of the job that created the output file, and those of input files and parents.

The *art* framework's division of the simulation and reconstruction jobs into modular pieces allows multiple developers to contribute to an effort, and to test their ideas in isolation before integrating them into a larger system. Because the data read in from an event is placed in read-only memory, analyzers can program with confidence that upstream algorithms cannot alter the data, but must produce additional data products which can later be processed or written out.

The LArSoft suite provides the interface to the event generators and Geant4 [5] for simulation of the passage of particles through the detector, the details of which are described in Section 1.6, and event reconstruction, the details of which are presented in Section 1.7.

The *art* framework and LArSoft source code are publicly available, and pre-built versions are provided [6] for supported versions of Linux and Mac OS X. Tools for compiling the framework and applications are also provided, along with all of the required dependencies, including the gnu C++ compiler. The versions of the software and its dependencies are managed by the UNIX Product Support (UPS) system, which allows easy version selection, setup and configuration of the LArSoft environment on computers with previous versions already installed, such as those at Fermilab.

LArSoft is under rapid development by both the core LArSoft team and by contributors from participating experiments: ArgoNeuT, LArIAT, MicroBooNE, SBND, and DUNE. Within DUNE, the LArTPC near-detector option, both ProtoDUNE detectors, and the Far Detector are clients and contributors of LArSoft.

1.6 Event simulation

Two kinds of events must be simulated in the ProtoDUNE-SP detector geometry: beam events and cosmic-ray events. Beam events are generated using a dedicated particle gun generator that has as input parameterizations of the flux and the beam profile parameters. The upstream beam instrumentation devices – wire chambers, Cherenkov counters, and time-of-flight-counters – are fully simulated as described in Section **??**. Cosmic-ray events are simulated either with the CRY [7, 8] event generator or CORSIKA [9]. Neutrino scattering events are simulated using GENIE [10]; while these events will be very rare in the detector, the extrapolation of the performance of the ProtoDUNE-SP detector to the FD will require simulating them. A dedicated generator in LArSoft simulates radionuclide decay products which can be overlaid on other events.

The detector geometry is coded in GDML files [5] that are generated by the GeGeDe [11] geometry system. These files contain the locations, sizes, shapes, and material content of the detector components, the active liquid argon volume, and the surrounding materials, such as the field cage, the beam windows, the cryostat the supporting structure, and the experimental hall. These external features will impact the distributions of cosmic-ray particles impinging on the active detector. The channels and volumes are numbered and named in the GDML files, with conventions followed by the LArSoft simulation code.

The active volume of the detector is divided into cubes 300 μm on a side, called *voxels*. GEANT4 tracks particles through the argon with each step ending on a voxel boundary. This allows the simulation of small-scale physics processes, such as delta-ray emission and showering, at a level of detail smaller than the intrinsic resolution of the detector. While GEANT4 calculates the energy deposited by each particle for each step, the simulation of ionization and scintillation-photon emission is performed using one of two algorithms in LArSoft: a dedicated parameterization that depends on the electric field in the liquid argon and the ionization density [12], or NEST [13], which is tuned to previous noble-liquid experimental results and introduces an anti-correlation between the photon yield and the ionization electron yield for each step.

An alternate simulation based on FLUKA [14, 15, 16] is being interfaced into LArSoft. FLUKA provides a MC simulation of neutrino-nucleus interactions as well as detailed modeling of particles traversing the detector, configurably replacing the functionality of both GENIE and GEANT in the simulation. The availability of alternatives allows for better flexibility in tuning the models to the data, as well as a basis for estimating systematic uncertainty.

The average specific energy loss for a minimum-ionizing particle (MIP) is approximately 2.12 MeV/cm. The W-value for ionization is 23.6 eV per electron-ion pair, and the W-value for scintillation is 19.5 eV per photon, resulting in tens of thousands of drifting electrons and photons per cm of charged-particle track in the detector. It is impractical to simulate the paths of each of these electrons and photons using GEANT4, and therefore computational techniques are incorporated into LArSoft to achieve a high simulation speed while preserving accuracy. The electrons are propagated by LArSoft-specific tools, including a tool that integrates over the distributions created by longitudinal and transverse diffusion, and a geometry tool that indicates the wire locations that record charge, assuming uniform wire spacing.

The effect of charge loss due to attachment of electrons to impurities (i.e., the effect of the electron lifetime), is implemented in this step. Effects due to space charge are simulated using a smoothly parameterized map of distortions in the apparent position components (x, y, z) of the charge deposits as functions of the true position. This map can be made using SPaCE [17], a program that traces particle trajectories in LAr based on the electric field calculated using Poisson's equation, a given space-charge density map, and the boundary conditions provided by the cathodes, anodes, and field cages. It is anticipated that the space-charge distortions in ProtoDUNE-SP may be as large as 20 cm [17].

Photon propagation is simulated using a library that contains the probabilities of observing a photon emitted at a particular point in space by a particular photon detector. Here, the space is divided into cubical voxels 6 cm on a side, and the library is indexed by photon detector element.

The simulated arrival times and charge amounts on each wire, and of each set of photons arriving at each photon detector, along with the identity of the particles generating them, are stored in the simulation output file for use in determining the performance of the downstream reconstruction algorithms. These charge depositions and photons are inputs to the detector response functions – the field response and the electronics response are convoluted with the true arrival times to make simulated waveforms. The detector field response functions are simulated using GARFIELD [18], but they will be validated with real data, as the simulation contains oversimplifications, such as inadequate modeling of induction signals.

The electronics gain is applied so that the simulated signals match the expected responses. Simulated noise is then added, and the result is quantized to reproduce the behavior of a 12-bit ADC, including realistic pedestals and saturation. A similar process is followed to simulate the response of the photon detectors, given the arrival times of the photons. Functionality exists within LArSoft to overlay MC-simulated particles with raw digits in the data in order to simulate pileup of cosmics and other beam interaction particles. The simulated raw digits are then written to compressed ROOT files for further analysis.

1.7 Event reconstruction algorithms and performance

1.7.1 Reconstruction

The interpretation of the data from LArTPC detectors has proven challenging, largely due to the wealth of information provided in each event by the detector, but also due to the high rate of multiple scattering and particle interactions, as well as the projection of 3D information onto a discretized 2D space of readout ADC counts on wires as functions of time. The flexibility of the *art*/LArSoft framework allows for multiple approaches for reconstructing and analyzing the data, and for the use of different approaches depending on the targeted physics deliverable.

For current large LArTPC detectors, noise filtering is applied to improve the signal-to-noise ratio. A large component of the noise in existing LArTPC experiments is from coherent sources – sources

that affect many neighboring wires and/or neighboring readout channels (channels from different planes may be interleaved in the front-end electronics). The contribution from coherent noise to a measured ADC value on a channel can be estimated from the data on nearby channels in the same time slice, and subtracted out. However, this procedure reduces the signal as well as the noise, in a manner that depends on the angle of the track or shower with respect to the drift field.

Procedures that first identify signal hits and protect them from distortion [19] are under study. With software noise filtering, MicroBooNE [20] has achieved excellent noise levels, consistent with expectations, based on the design specification of the cold electronics. In MicroBooNE, various sources of noise have been identified and hardware upgrades are ongoing to eliminate them. Once noise has been removed, signals will be processed to recover the ionization charge.

Hits are identified by seeking deconvoluted signals exceeding thresholds that are adjusted to minimize the creation of false noise hits while preserving the true signal hits. The standard LArSoft hit finder fits Gaussian functions to the deconvoluted signals, and saves the times, widths, and amplitudes of the Gaussians. In addition, it saves the sum of the ADC readings in the time windows corresponding to the hits; since a Gaussian function is not always representative of the charge arrival distribution, the resolution of the calorimetry is improved by summing the ADC counts.

The hits are associated with DAQ channels not wire segments, since, due to the wrapping of the induction-plane wires in the APAs, the location of the charge deposition contributing to the hit is ambiguous. Because the wire angle is chosen such that each induction wire intersects each collection-plane wire at most one time, only two views are needed in order to identify hits and resolve ambiguities. A separate LArSoft module compares the hits in the collection and induction views to resolve ambiguities.

Physics analyses are most sensitive with a full 3D reconstruction of the event – the primary vertex (if there is one), the tracks and the showers. ProtoDUNE-SP is implementing several approaches to achieve this. Once hits are identified on wire segments, 2D reconstruction identifies clusters and tracks in each view separately, and 3D hypotheses for the event are constructed by comparing the 2D clusters in the separate planes. The 2D clustering algorithms currently in use are the Blurred Clustering Algorithm [21], LineCluster [22], and TrajCluster [23].

1.7.2 Performance

Algorithm performance metrics are efficiency, purity, and completeness. The efficiency of the algorithm is defined as the fraction of true particles that match reconstructed objects within the bounds of pre-specified criteria, such as matching position and length and the type of object expected. The purity of the reconstructed object is defined as the fraction of hits (or charge) included in that object that truly came from the matched particle, divided by the total number of hits (or charge) included in the reconstructed cluster. The completeness is defined as the number of true hits that are found in a cluster, track or shower expressed as a fraction of total true hits in that object.

Particle-level and event-level performance metrics include particle identification and misidentifica-

tion rates, shower energy resolution, and energy scale offsets.

EMShower

The EMShower package [24] takes the output of the Blurred Clustering Algorithm and produces energies, angles, and start positions for 3D showers, as well as the dE/dx in the initial part of the shower. Identifying events with two showers consistent with $\pi^0 \to \gamma\gamma$ decays allows for an *in situ* calibration of the electromagnetic energy scale as well as the performance of shower identification and reconstruction for photons that are produced inside the detector. A distribution of reconstructed π^0 masses in Monte Carlo is shown in Figure 1.2.

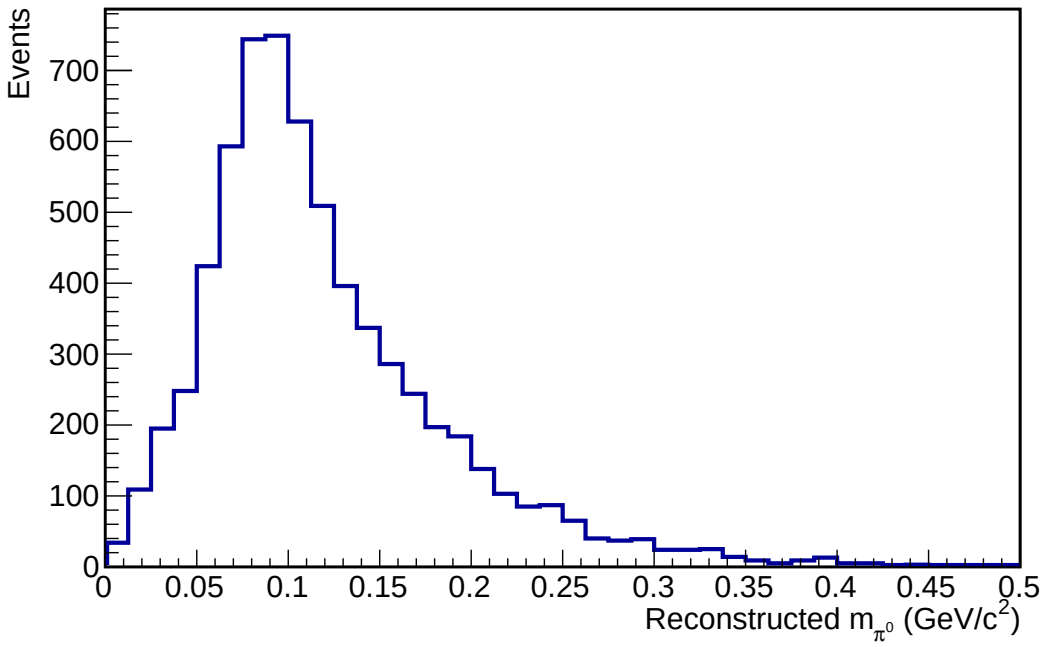

Figure 1.2: The reconstructed invariant masses of π^0 candidates in Monte Carlo using the Blurred-Cluster and EMShower algorithms.

PANDORA

The reconstruction framework PANDORA [25] also works by building up a 3D picture from 2D reconstructed objects. PANDORA is a flexible framework developed for International Linear Collider (ILC) detector simulation, and provides a convenient way to develop algorithms for reconstructing particles. In all, more than 80 algorithms, each targeting a specific topology, have been incorporated into PANDORA to date. PANDORA allows multiple reconstruction passes through the data. Different criteria for clustering hits into tracks and showers may be applied when seeking to remove cosmic rays than when identifying signal events. PANDORA follows a process that clusters hits in 2D, reconstructs vertices in 3D, reconstructs tracks in 3D, reconstructs showers in 3D, performs a *mop-up* step in 2D and 3D, and finally performs full event-building in 3D.

Plots of the efficiency, the completeness, and the difference between the true and reconstructed track lengths for single muons with momentum between 300 MeV and 5 GeV in the ProtoDUNE-SP geometry are shown in Figure 1.3. The PANDORA algorithm performs very well, although a small inefficiency occurs in the current algorithm's ability to match track segments from one APA's drift volume to another.

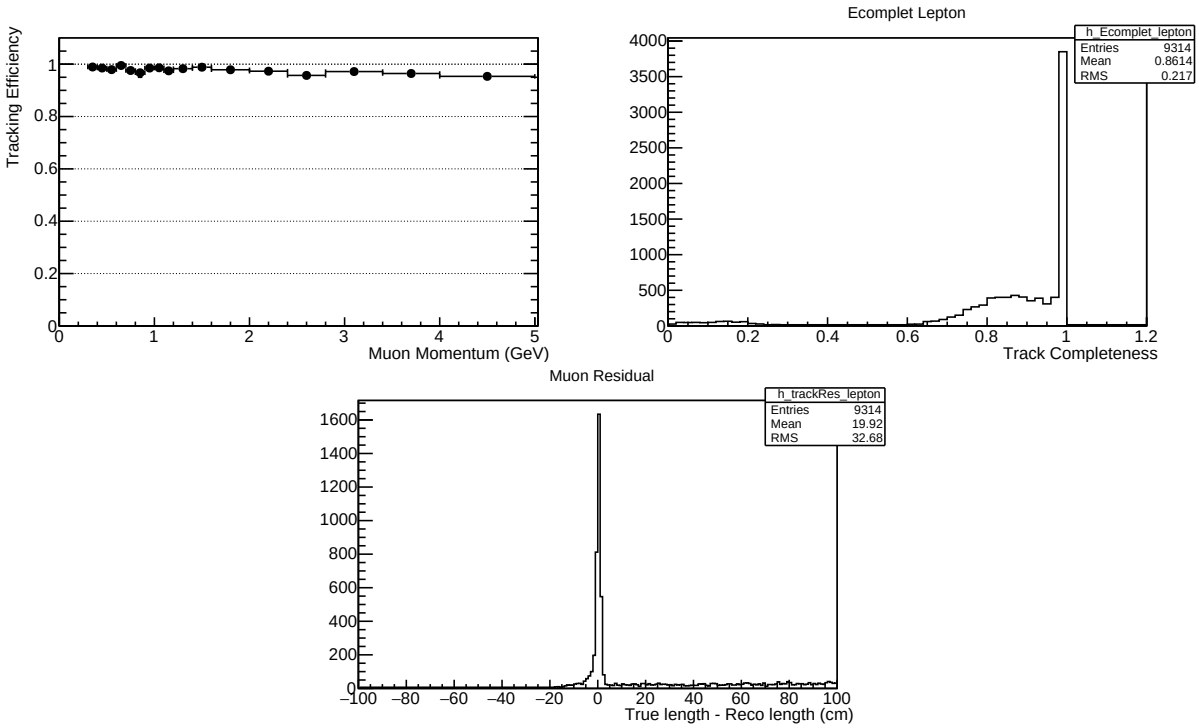

Figure 1.3: Performance of the PANDORA reconstruction algorithm for single muons with momentum between 300 MeV/c and 5 GeV/c. The top left-hand figure shows the tracking efficiency as a function of muon momentum, the top right-hand figure shows the distribution of tracking completeness, and the bottom figure shows the distribution of the difference between the reconstructed muon track length and the true track length.

PMA

Another approach to 3D reconstruction in LArTPC detectors is referred to as the *Projection Matching Algorithm (PMA)* [26]. PMA was primarily developed as a technique for 3D reconstruction of individual particle trajectories (trajectory fits) [26] Instead of building up a 3D hypothesis from 2D clusters, it starts with the 3D hypothesis and compares the 2D projection of the predicted trajectory of a particle with the observed data. Association of hits between the 2D planes is not needed in this approach, improving its performance in problematic cases, such as isochronous and short tracks.

PMA can take as input the output from different pattern recognition algorithms, from LineCluster [22] to WireCell (described below). Because these 2D algorithms are run on each 2D projection independently, and because of detector defects, clusters from particles may be broken into several smaller pieces, fractions of 2D clusters may be missing, and clusters obtained from complementary

projections are not guaranteed to cover corresponding sections of trajectories. Such behavior is expected since ambiguous trajectories can be resolved only if the information from multiple 2D projections is used. PMA performs higher-level pattern recognition using as input clustering information from all projections in order to search for the best matching combinations of clusters. The algorithm also attempts to correct hit-to-cluster assignments using properties of 3D reconstructed objects.

Plots of the efficiency, the completeness, and the difference between the true and reconstructed track lengths for single muons with momentum between 300 MeV and 5 GeV in the ProtoDUNE-SP geometry are shown in Figure 1.4.

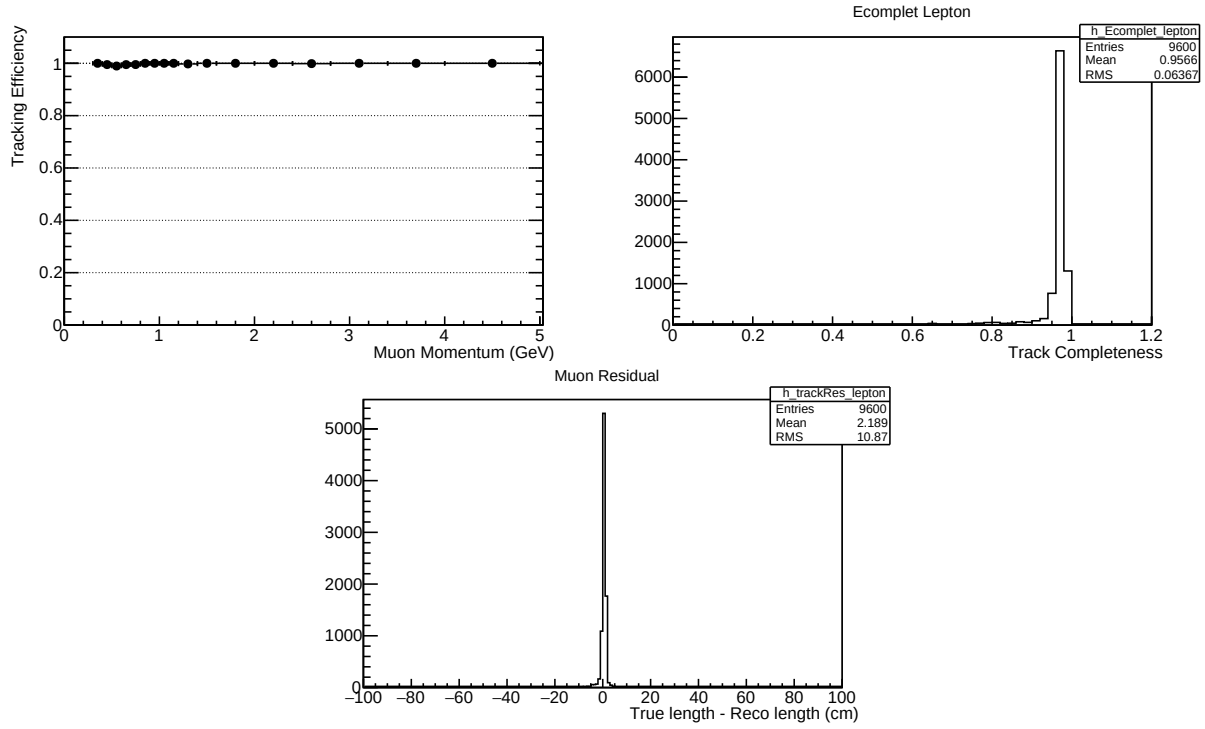

Figure 1.4: Performance of the PMA reconstruction algorithm for single muons with momentum between 300 MeV/c and 5 GeV/c. The left-hand figure shows the tracking efficiency as a function of muon momentum, the middle figure shows the distribution of tracking completeness, and the right-hand figure shows the distribution of the difference between the reconstructed muon track length and the true track length.

PMA has been used successfully in ProtoDUNE-SP to reconstruct simulated beam particles. In order to illustrate the performance of the entire reconstruction chain, Figure 1.5 shows the spatial resolution of the interaction vertex with neutral pion production appearing in the 2-GeV/c π^+ sample. The resolution is found to be 0.6 cm in this study. A similar resolution is obtained for the reconstruction of inelastic interaction vertices in the 2-GeV/c proton sample.

Figures 1.6 and 1.7 show examples of reconstruction of a 2-GeV/c proton in the test beam and cosmic-ray muons, respectively.

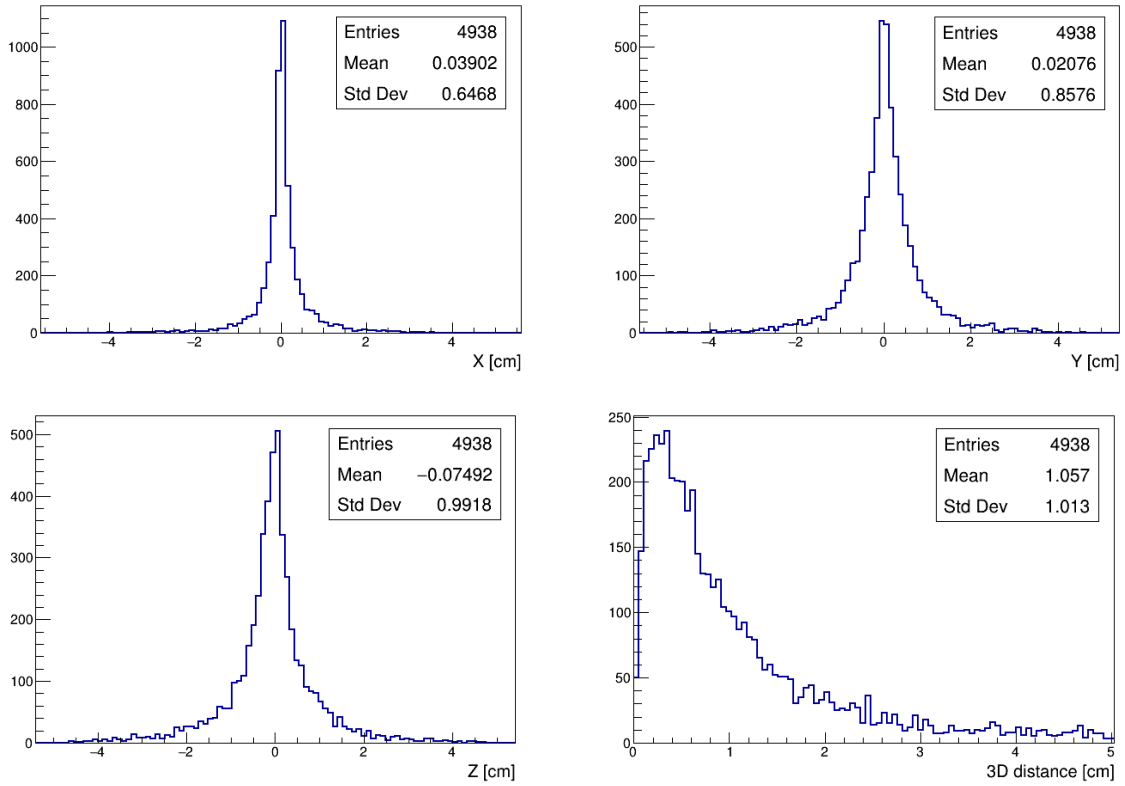

Figure 1.5: Vertex position resolution in cm in x, y, and z and 3D for the inelastic interaction of charged pions on liquid argon nuclei in events in which a π^0 is produced, in ProtoDUNE-SP, using the PMA algorithm.

fig:PMAp

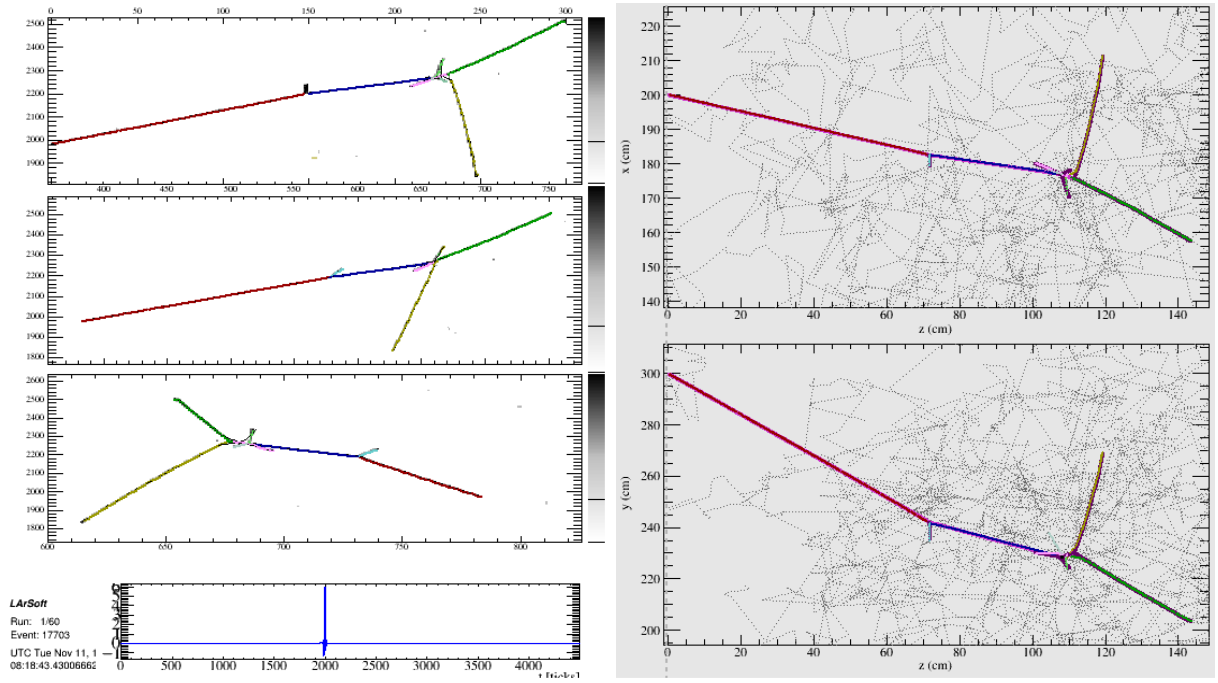

Figure 1.6: Example of reconstructed event of simulated proton with initial momentum 2 GeV/c (reconstruction algorithms: gaushit, Line Cluster and PMA).

fig:PMAp

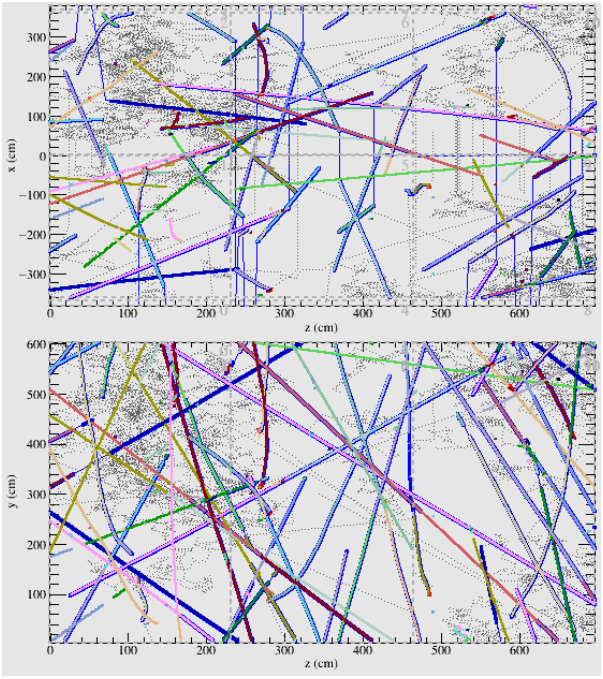

Figure 1.7: Example of reconstructed cosmic muons using gaushit, Line Cluster and PMA. fig:PMAc

WireCell

WireCell wire-cell [27], a new reconstruction method under development, adopts a very different approach from the aforementioned algorithms . Instead of performing pattern recognition directly on each of the 2D views (drift time versus wire number), the first step of the WireCell reconstruction is to perform 3D imaging with time, geometry, and charge information. The algorithm takes advantage of this information to suppress the effects of electronic noise; Hits from different wire planes arriving in different time slices cannot be associated with each other.

cannot be associated because it's impossible, or are not allowed to be associated and the 3D imaging process accomplishes this? Please clarify

(The definition of *hit* is based on signal strength after charge extraction in a 2-μs time slice.) Hits from wires that do not cross in a region consistent with the charge deposition cannot be associated with each other. Hits from different wire planes with different signal strengths are unlikely to be associated with each other. Use of the charge information takes advantage of the fact that in a LArTPC with induction planes, each of the wire planes, in principle, detects the same ionization electrons as the other planes. Figure 1.8 shows an example of the improvement of WireCell 3D imaging over the more traditional approach. fig:quality

The suppression of the electronic noise comes at the cost of more sensitivity to hit inefficiencies from dead channels or the signal processing steps. Since the track and shower hypotheses are not used, the 3D imaging works for any event topology. Pattern recognition is needed to identify the content of these 3D images. Figure 1.9 shows the performance of the currently available 3D pattern recognition in WireCell. For the long track going close to parallel to the wire plane, the fig:tracking2

reconstructed track shows a zig-zag behavior. This is due to the current lack of a fine track-fitting algorithm that is expected to be added in the near future. Further developments of the WireCell pattern recognition algorithms are needed before meaningful physics quantities can be calculated.

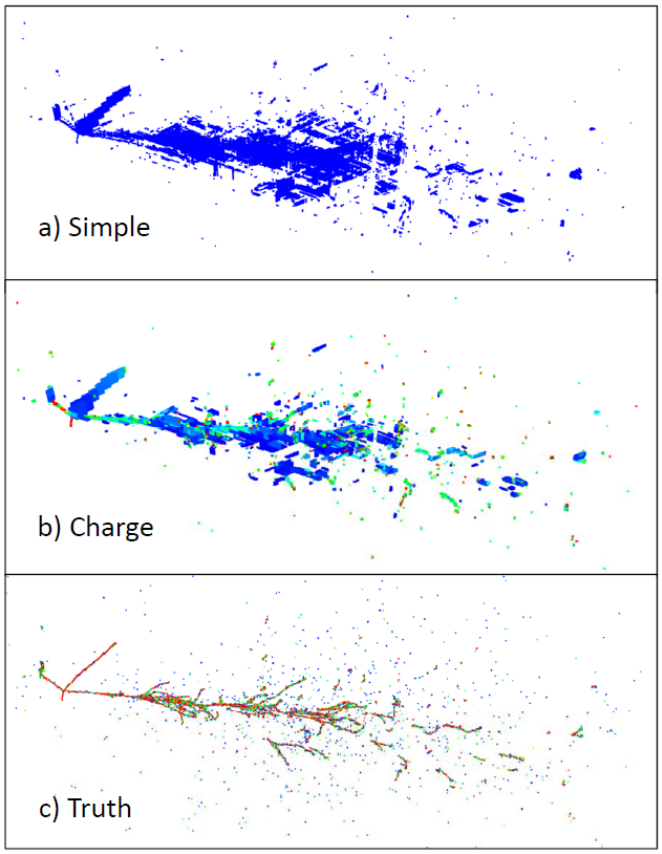

Figure 1.8: Comparison of imaging reconstruction qualities with and without the charge information. fig:qual

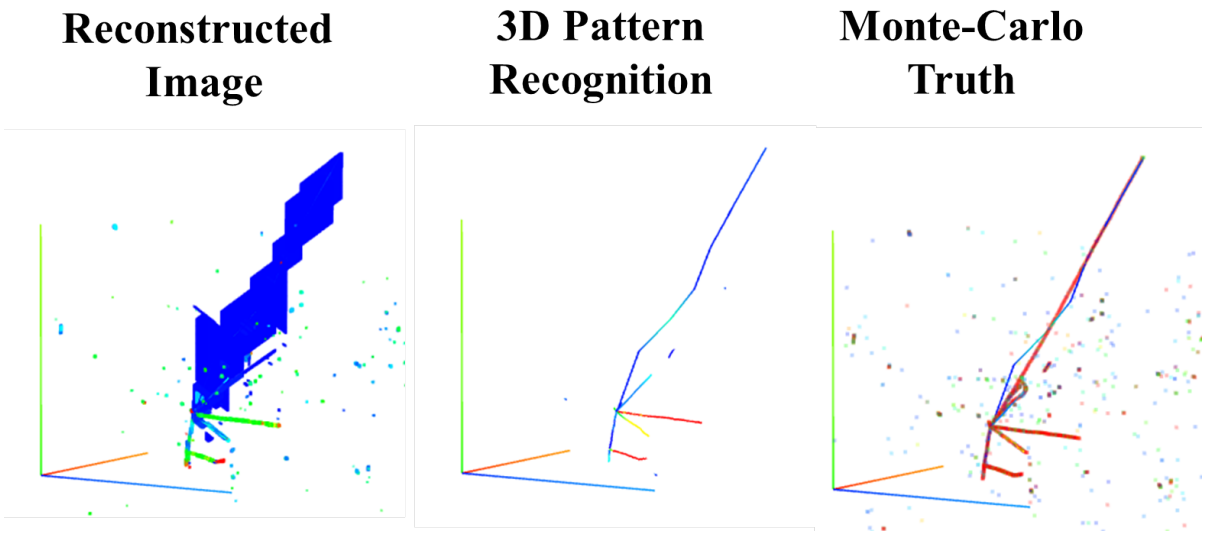

Figure 1.9: The reconstructed image is shown on the left panel for one neutrino interaction event. The image was passed through the 3D pattern recognition program with tracks identified (middle panel). The identified pattern is compared with Monte-Carlo truth in the right panel. The zig-zag line in the right panel is the identified track. More sophisticated track-fitting algorithms, to be added in the future, will improve the track reconstruction. fig:trac

References

[1] "Design of the Data Management System for the ProtoDUNE Experiment (DUNE doc-db 1212)." http://docs.dunescience.org:8080/cgi-bin/ShowDocument?docid=1212.

[2] "Data scenarios spreadsheet." DUNE DocDB 1086.

[3] "LArSoft Collaboration, Software for LArTPCs." http://larsoft.org.

[4] "The art Event Processing Framework." http://art.fnal.gov.

[5] S. Agostinelli *et al.*, "Geant4: A Simulation toolkit," *Nucl. Instrum. Meth.*, vol. A506, pp. 250–303, 2003.

[6] "Scientific Software for Relocatable UPS ." http://http://scisoft.fnal.gov/.

[7] Chris Hagmann, David Lange, Jerome Verbeke, Doug Wright, "Proton-induced Cosmic-ray Cascades in the Atmosphere," tech. rep. http://nuclear.llnl.gov/simulation/doc_cry_v1.7/cry.pdf.

[8] Chris Hagmann, David Lange, Doug Wright, "Monte Carlo Simulation of Proton-induced Cosmic-ray Cascades in the Atmosphere," tech. rep., 2012. UCRL-TM-229452.

[9] D. Heck, G. Schatz, T. Thouw, J. Knapp, and J. N. Capdevielle, "CORSIKA: A Monte Carlo code to simulate extensive air showers," 1998.

[10] C. Andreopoulos *et al.*, "The GENIE Neutrino Monte Carlo Generator," *Nucl. Instrum. Meth.*, vol. A614, pp. 87–104, 2010.

[11] "General Geometry Description (GGD)." https://github.com/brettviren/gegede.

[12] J. B. Birks, *The Theory and practice of scintillation counting.* 1964.

[13] M. Szydagis, N. Barry, K. Kazkaz, J. Mock, D. Stolp, M. Sweany, M. Tripathi, S. Uvarov, N. Walsh, and M. Woods, "NEST: A Comprehensive Model for Scintillation Yield in Liquid Xenon," *JINST*, vol. 6, p. P10002, 2011.

[14] G. Battistoni, T. T. Böhlen, F. Cerutti, P. W. Chin, L. S. Esposito, A. Fassò, A. Ferrari, A. Lechner, A. Empl, A. Mairani, A. Mereghetti, P. G. Ortega, J. Ranft, S. Roesler, P. R. Sala, V. Vlachoudis, and G. Smirnov, "Overview of the FLUKA code," *Annals of Nuclear Energy*, vol. 82, p. 10, 2015.

[15] A. Ferrari, P. R. Sala, A. Fasso, and J. Ranft, "FLUKA: A multi-particle transport code (Program version 2005)," 2005.

[16] G. Battistoni, P. R. Sala, M. Lantz, A. Ferrari, and G. Smirnov, "Neutrino interactions with FLUKA," *Acta Phys. Polon.*, vol. B40, pp. 2491–2505, 2009.

[17] M. Mooney, "The MicroBooNE Experiment and the Impact of Space Charge Effects," in *Meeting of the APS Division of Particles and Fields (DPF 2015) Ann Arbor, Michigan, USA, August 4-8, 2015*, 2015.

[18] `http://garfield.web.cern.ch/garfield/`.

[19] "Noise Characterization and Filtering in the MicroBooNE TPC." `http://www-microboone.fnal.gov/publications/publicnotes/MICROBOONE-NOTE-1016-PUB.pdf`.

[20] MicroBooNE noise filtering technot, doc-db:5730 `http://microboone-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=5730`.

[21] `http://larsoft.org/single-record/?pdb=110`.

[22] "The Cluster Crawler Suite Technical Manual." `http://microboone-docdb.fnal.gov/cgi-bin/ShowDocument?docid=2831`.

[23] "TrajCluster: A 2D Cluster Finder ." `https://cdcvs.fnal.gov/redmine/attachments/download/34265/TrajCluster%20Technical%20Note.pdf`.

[24] `http://larsoft.org/single-record/?pdb=113`.

[25] J. S. Marshall and M. A. Thomson, "The Pandora Software Development Kit for Pattern Recognition," *Eur. Phys. J.*, vol. C75, no. 9, p. 439, 2015.

[26] `http://larsoft.org/single-record/?pdb=102`.

[27] `http://www.phy.bnl.gov/wire-cell/`.