Reconstruction Software in DUNE

Tingjun Yang
FNAL

Nov 13, 2017

Introduction

- I will give an overview of the current reconstruction chain used by DUNE FD. A lot of the algorithms are shared with protoDUNE and other LArTPC experiments.
- You need to know how to setup larsoft/dunetpc. See previous tutorials. There are many wiki pages.
- I will focus on the TPC reconstruction. Alex Himmel will talk about the photon detector reconstruction.
- Outline
  - Overview of data products
  - Brief description of various reconstruction algorithms

Data products

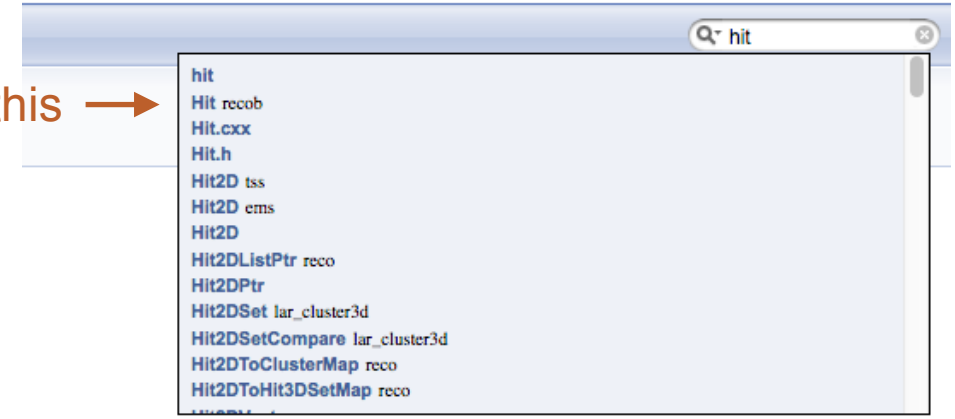
- raw::* - raw data
  - raw::RawDigit, raw::AuxDetDigit, raw::OpDetPulse, raw::OpDetWaveform, raw::Trigger, raw::BeamInfo, etc.
- recob::* - reconstructed information
  - recob::Wire, recob::Hit, recob::Cluster, recob::EndPoint2D, recob::Vertex, recob::PFParticle, recob::Track, recob::Shower, recob::OpHit, recob::OpFlash, etc.
- anab::* - information that is derived from reconstruction information
  - anab::Calorimetry, anab::ParticleID, anab::CosmicTag, anab::T0, etc.
- simb::* - simulation information
  - simb::MCTruth, simb::MCParticle, simb::MCFlux, etc.
- Associations - links between different data products
  - https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki/Use_associations

Get information from Doxygen

- http://nusoft.fnal.gov/larsoft/doxsvn/html/index.html

If you want to get information for recob::Hit, type hit here

Click this



http://nusoft.fnal.gov/larsoft/doxsvn/html/classrecob_1_1Hit.html

Get information from an art file

- lar -c **eventdump.fcl** /pnfs/dune/tape_backed/dunepro/mc/dune/full-reconstructed/01/89/47/00/prodgenie_atmnu_max_dune10kt_1x2x6_0_20171102T160222_merged0.root -n 1

ProcessName	ModuleLabel	InstanceLabel	ClassName	size
GenieGen....	generator.....	.....	std::vector<simb::GTruth>.....	...1
GenieGen....	TriggerResults.	.....	art::TriggerResults.....	...-
GenieGen....	generator.....	.....	std::vector<sim::BeamGateInfo>.....	...1
GenieGen....	rns.....	.....	std::vector<art::RNGsnapshot>.....	...0
GenieGen....	generator.....	.....	std::vector<simb::MCTruth>.....	...1
GenieGen....	generator.....	.....	art::Assns<simb::MCTruth,simb::MCFlux,void>.....	...1
GenieGen....	generator.....	.....	std::vector<simb::MCFlux>.....	...1
GenieGen....	generator.....	.....	art::Assns<simb::MCTruth,simb::GTruth,void>.....	...1
G4.....	largeant.....	.....	std::vector<sim::OpDetBacktrackerRecord>.....	.113
G4.....	rns.....	.....	std::vector<art::RNGsnapshot>.....	...2
G4.....	TriggerResults.	.....	art::TriggerResults.....	...-
G4.....	largeant.....	.....	std::vector<simb::MCParticle>.....	.391
G4.....	largeant.....	.....	std::vector<sim::AuxDetSimChannel>.....	...0
G4.....	largeant.....	.....	art::Assns<simb::MCTruth,simb::MCParticle,void>.....	.391
G4.....	largeant.....	.....	std::vector<sim::SimChannel>.....	1181
G4.....	largeant.....	.....	std::vector<sim::SimPhotonsLite>.....	.113
Detsim.....	TriggerResults.	.....	art::TriggerResults.....	...-
Detsim.....	opdigi.....	.....	std::vector<raw::OpDetWaveform>.....	.618
Detsim.....	daq.....	.....	std::vector<raw::RawDigit>.....	4510
Detsim.....	rns.....	.....	std::vector<art::RNGsnapshot>.....	...1
Reco.....	TriggerResults.	.....	art::TriggerResults.....	...-
Reco.....	trajcluster....	.....	std::vector<recob::Vertex>.....	...6
Reco.....	trajcluster....	.....	std::vector<anab::CosmicTag>.....	...0
Reco.....	pmtrajfit.....	kink.....	std::vector<recob::Vertex>.....	...0
Reco.....	pandora.....	.....	std::vector<recob::PCAxis>.....	...1
Reco.....	pmtrack.....	.....	std::vector<recob::Vertex>.....	...9
Reco.....	pandoracalo....	.....	art::Assns<recob::Track,anab::Calorimetry,void>.....	..24
Reco.....	pandora.....	.....	art::Assns<recob::PFParticle,recob::SpacePoint,void>.....	1224
Reco.....	pmtrackpid.....	.....	art::Assns<recob::Track,anab::ParticleID,void>.....	..36
Reco.....	pmtrajfit.....	.....	std::vector<recob::Vertex>.....	...0
Reco.....	pandora.....	.....	std::vector<recob::Vertex>.....	..11
Reco.....	pmtrack.....	.....	art::Assns<recob::PFParticle,recob::Vertex,void>.....	..18
Reco.....	pmtracktccalo..	.....	std::vector<anab::Calorimetry>.....	..45
Reco.....	pandoracalo....	.....	std::vector<anab::Calorimetry>.....	..24
Reco.....	pmtracktccalo..	.....	art::Assns<recob::Track,anab::Calorimetry,void>.....	..45
Reco.....	hitfd.....	.....	art::Assns<recob::Wire,recob::Hit,void>.....	1928
Reco.....	pmtrack.....	kink.....	art::Assns<recob::Track,recob::Vertex,void>.....	...0
Reco.....	pandora.....	.....	art::Assns<recob::PFParticle,recob::Vertex,void>.....	..11
Reco.....	pandora.....	.....	art::Assns<recob::PFParticle,recob::Seed,void>.....	.773
Reco.....	pmtracktc.....	node.....	std::vector<recob::Vertex>.....	...0

FD reconstruction fcl file

- https://cdcvs.fnal.gov/redmine/projects/dunetpc/repository/revisions/develop/entry/fcl/dunefd/reco/standard_reco_dune10kt.fcl
- srcs/dunetpc/fcl/dunefd/reco/standard_reco_dune10kt.fcl

```
#include "services_dune.fcl"
#include "hitfindermodules_dune.fcl"
#include "cluster_dune.fcl"
#include "trackfindermodules_dune.fcl"
#include "pandoramodules_dune.fcl"
#include "calorimetry_dune10kt.fcl"
#include "particleid.fcl"
#include "mctruth0matching.fcl"
#include "t0reco.fcl"
#include "opticaldetectormodules_dune.fcl"
#include "trackshowerhits.fcl"
#include "showerfindermodules_dune.fcl"
#include "emshower3d.fcl"
#include "imagepatternalgs.fcl"

#include "tools_dune.fcl"

process_name: Reco
```

```
pmtrack:                @local::dunefd_pmalgtrackmaker
pmtrackcalo:            @local::dune10kt_calomc
pmtrackpid:             @local::standard_chi2pid
```

```
reco: [ rns,
        #optical hits and flashes
        ophit, opflash,
        #TPC wire signals
        caldata,
        #hit reco
        gaushit,
        #disambiguation
        hitfd,
        #cluster reco
        linecluster, trajcluster,
        #track shower split
        trkshowersplit,
        #pandora
        pandora, #pandora stitcher goes here
        pandoracalo, pandorapid,
        #pmatrack
        pmtrack, pmtrackcalo, pmtrackpid,
        pmtrajfit, pmtrajfitcalo, pmtrajfitpid,
        pmtracktc, pmtracktccalo, pmtracktcpid,
        pmtrajfittc, pmtrajfittccalo, pmtrajfittcpid,
        #shower reconstruction
        blurredcluster, emtrkmichelid, emshower#, emshower3d, mergeemshower3d
        ]
```

- srcs/dunetpc/dune/TrackFinderDUNE/trackfindermodules_dune.fcl

```
#include "trackfindermodules.fcl"

BEGIN_PROLOG
```

```
dunefd_pmalgtrackmaker:                        @local::standard_pmalgtrackmaker
dunefd_pmalgtrackmaker.HitModuleLabel:         "linecluster"
dunefd_pmalgtrackmaker.ClusterModuleLabel:     "linecluster"
dunefd_pmalgtrackmaker.PMAlgTracking.FlipToBeam:          true
dunefd_pmalgtrackmaker.PMAlgTracking.AutoFlip_dQdx:       false
dunefd_pmalgtrackmaker.PMAlgTracking.RunVertexing:        true
dunefd_pmalgtrackmaker.ProjectionMatchingAlg.HitWeightZ: 1.0
dunefd_pmalgtrackmaker.ProjectionMatchingAlg.HitWeightV: 1.0
dunefd_pmalgtrackmaker.ProjectionMatchingAlg.HitWeightU: 1.0
```

- srcs/larreco/larreco/TrackFinder/trackfindermodules.fcl

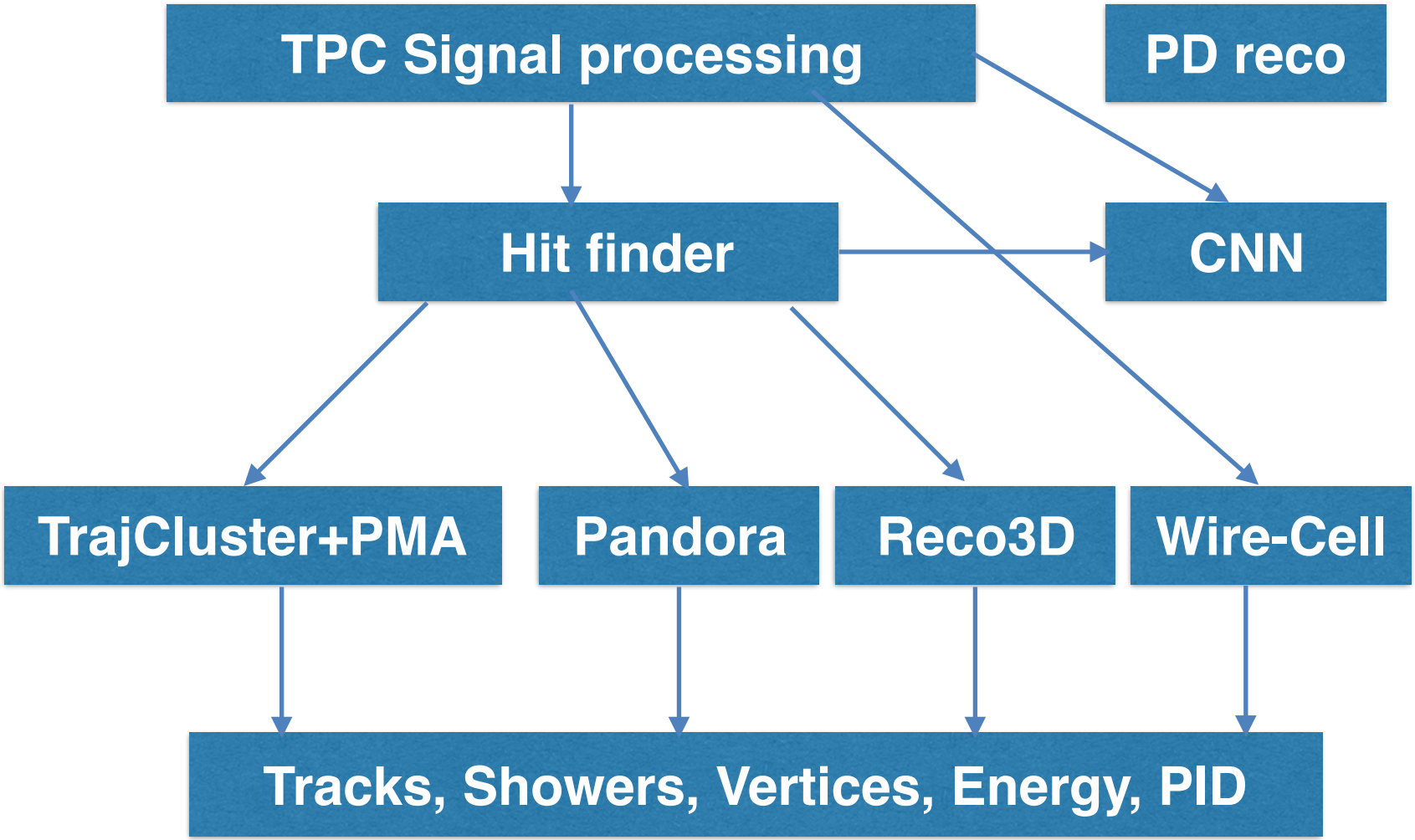
```
standard_pmalgtrackmaker:
{
  module_type:            "PMAlgTrackMaker"
  ProjectionMatchingAlg:  @local::standard_projectionmatchingalg
  PMAlgTracking:          @local::standard_pmalgtracker
  PMAlgCosmicTagging:     @local::standard_pmalgtagger
  PMAlgVertexing:         @local::standard_pmavertexalg
  PMAlgStitching:         @local::standard_pmastitchalg
                          #
  SaveOnlyBranchingVtx:   false       # use true to save only vertices interconnecting many tracks, otherwise
                                      # vertex is added to the front of each track
  SavePmaNodes:           false       # save track nodes (only for algorithm development purposes)
                                      #
  HitModuleLabel:         "hits"      # unclustered hits are used for tracks validation
  WireModuleLabel:        "caldata"   # deconvoluted adc is also used for tracks validation

  ClusterModuleLabel:     "cluster"   # cluster module label, these clusters are used for track building
  EmClusterModuleLabel:   ""          # EM-like clusters, will be excluded from tracking if provided
}
```

- srcs/larreco/larreco/TrackFinder/PMAlgTrackMaker_module.cc

Reconstruction Chain

DUNE-doc-1689-v3



TPC Signal Formation

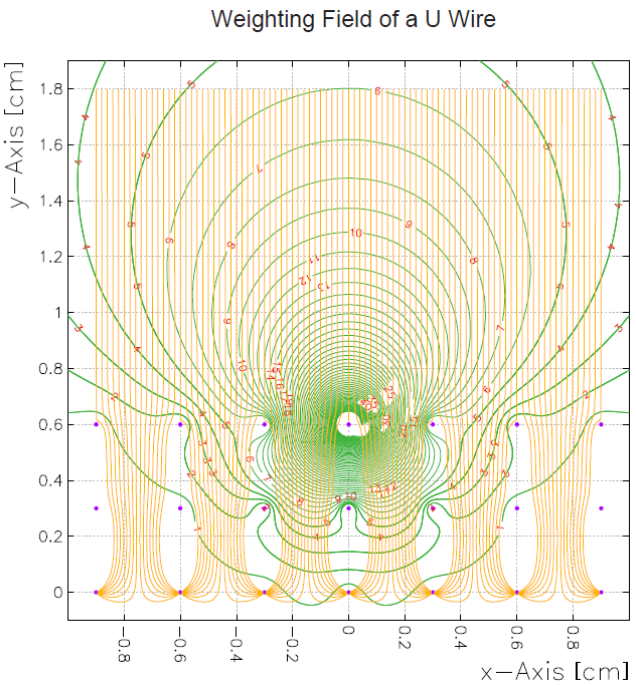
Shockley–Ramo theorem

$$i = -q \cdot \vec{E}_w \cdot \vec{v}_q$$

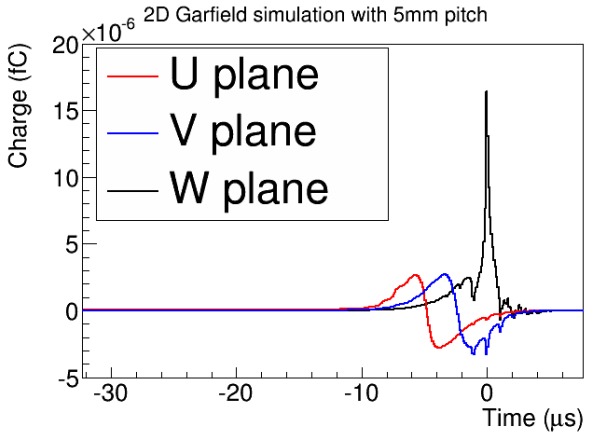
v_q: velocity

E_w: weighting field

q: charge



Garfield simulation of weighting field

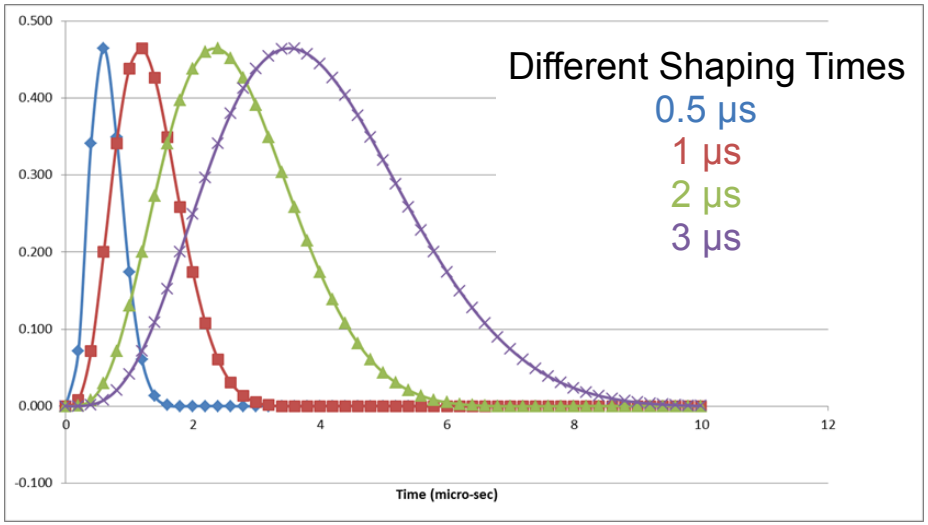


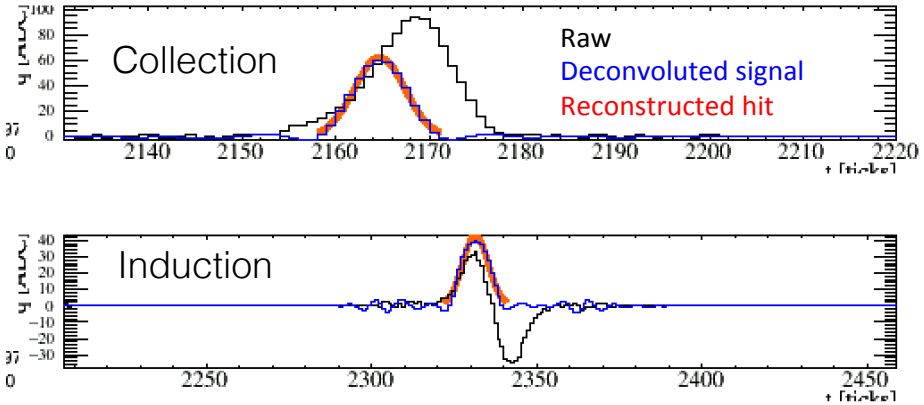
Garfield simulation of field responses

- More details: http://www-microboone.fnal.gov/publications/publicnotes/MICROBOONE-NOTE-1017-PUB.pdf

Signal Processing

BNL SPICE ASIC simulation

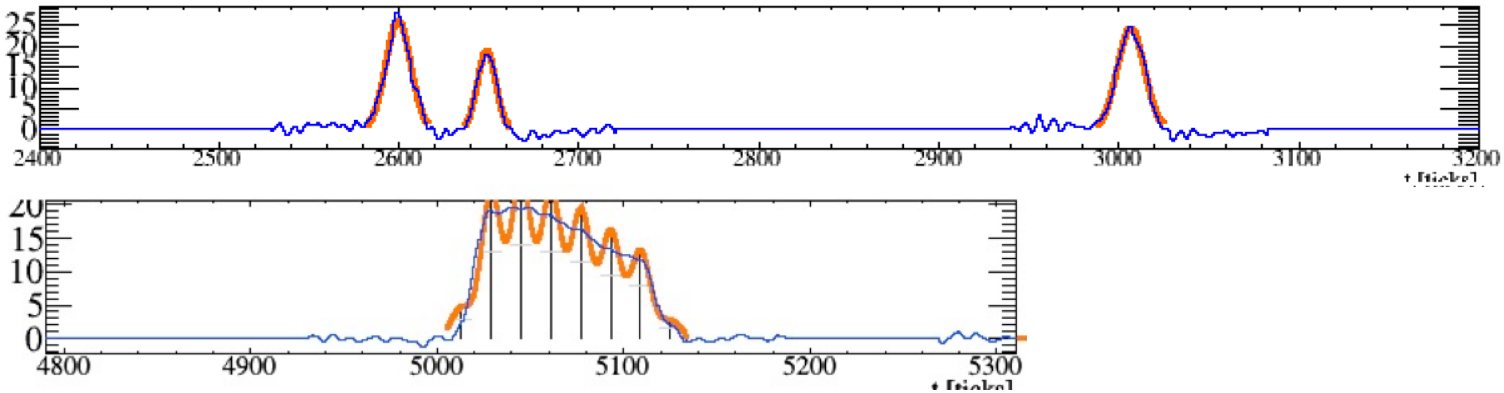




- Filter noise and remove effects of field response and electronics response through deconvolution.
- It is important to take into account induced signals from far away electrons to recover ionization charge.
- Input: raw::RawDigit
- Output: recob::Wire
  - One per channel/wire
  - std::vector< float > Signal () - ADC per tick
  - size_t NSignal () - number of ticks

Hit Finder

- Default hit finder in LArSoft is GausHitFinder.
- Start from deconvoluted signals on wire and define areas above threshold known as “pulses”.
- Once a pulse is found, a “n” Gaussian hypothesis is applied where “n” is defined by the number of peaks initially identified within the pulse.



- Input: recob::Wire
- Output: recob::Hit
  - float PeakTime () - Time of the signal peak, in tick units.
  - float Integral () - Charge
  - raw::ChannelID_t Channel () - ID of the readout channel

3D reconstruction

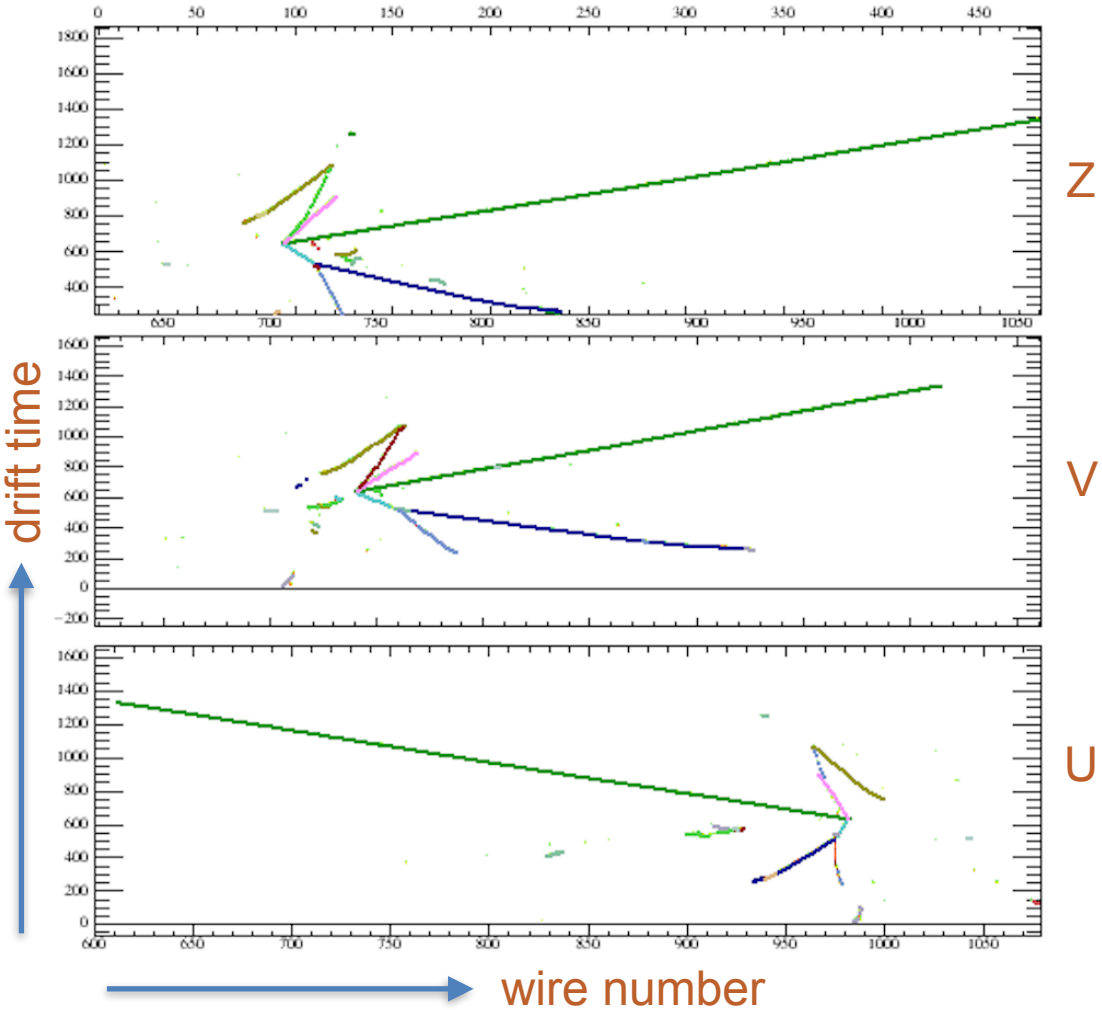
- First discuss the 2D matching approach.

	2D cluster reco	3D matching (based on time)	Track/shower maker
Data products	recob::Cluster	recob::PFParticle	recob::Track recob::Shower recob::Vertex
Algorithms	(LineCluster) TrajCluster Pandora BlurredCluster	TrajCluster Pandora PMA	Pandora PMA KalmanTrack EMShower TrajCluster(shower only)

TrajCluster

- TrajCluster: 2nd generation LineCluster (ClusterCrawler).
- Trajectory (cluster) reconstruction starts from construction of a short “seed” of nearby trajectory points (TPs).
- Additional nearby TPs are attached to the leading edge of the trajectory if they are similar to the TPs already attached to it.
  - Using impact parameter, hit width and hit charge.
- Momentum estimate using Multiple Coulomb Scattering (MCSMom) for each trajectory
  - An valuable parameter for pattern recognition, tagging muons & showers
- Use low-momentum trajectories to form shower-like clusters.
- Input: recob::Hit
- Output: recob::Cluster, recob::EndPoint2D, recob::PFParticle, recob::Vertex, recob::Shower, etc.

One example



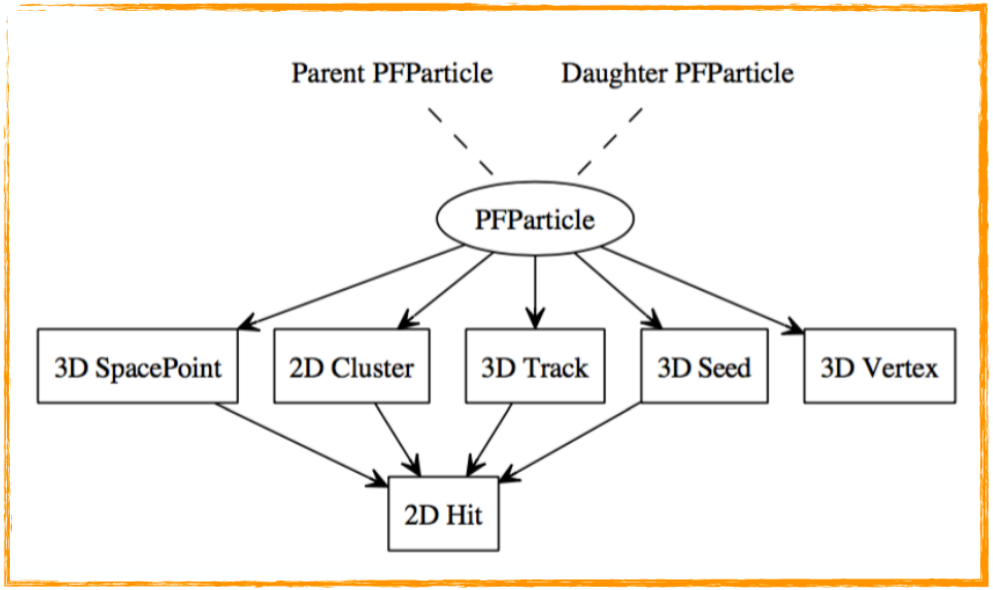
Different colors represents different clusters.
Same color between views indicates matched clusters (PFParticle).

More in https://cdcvs.fnal.gov/redmine/documents/1026

Pandora

https://github.com/PandoraPFA

- Pandora is a well-established tool for fine-grain pattern recognition in high energy physics (future linear collider, LHC, LArTPCs)
- Supports multi-algorithm approach to automated and optimized pattern recognition.
- Takes hits as input. Outputs PFParticles with hierarchies
  - Reconstruct both track-like and shower-like PFParticles for further track/shower reconstruction.

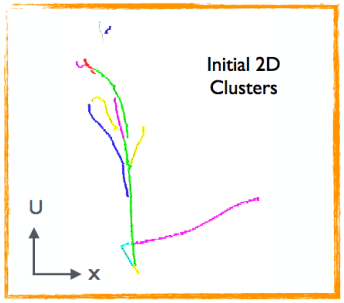


- Input: recob::Hit
- Output: recob::Cluster, recob::EndPoint2D, recob::PFParticle, recob::Vertex, recob::SpacePoint, recob::Track, recob::Shower, etc.

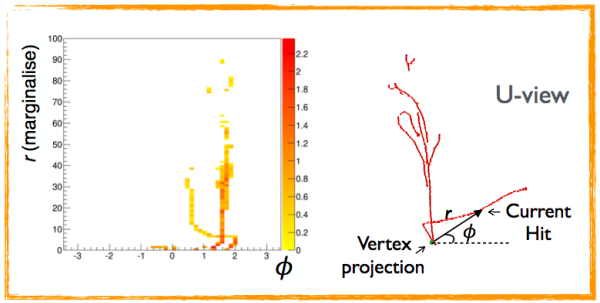
Two Chains of Algorithms

- PandoraCosmic:
  - more strongly track-oriented; showers assumed to be delta rays, added as daughters of the muons; muon vertices at track high-y coordinate.
- PandoraNu:
  - more careful to find interaction vertex and to protect particles emerging from vertex. Careful treatment to address track/shower tension.

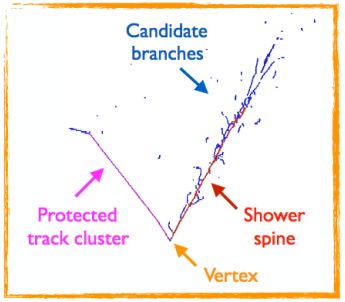
2D reconstruction

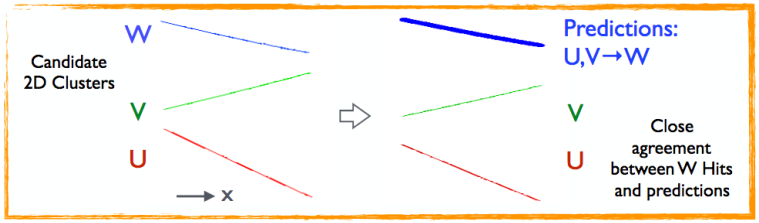


Vertex reconstruction



Shower reconstruction

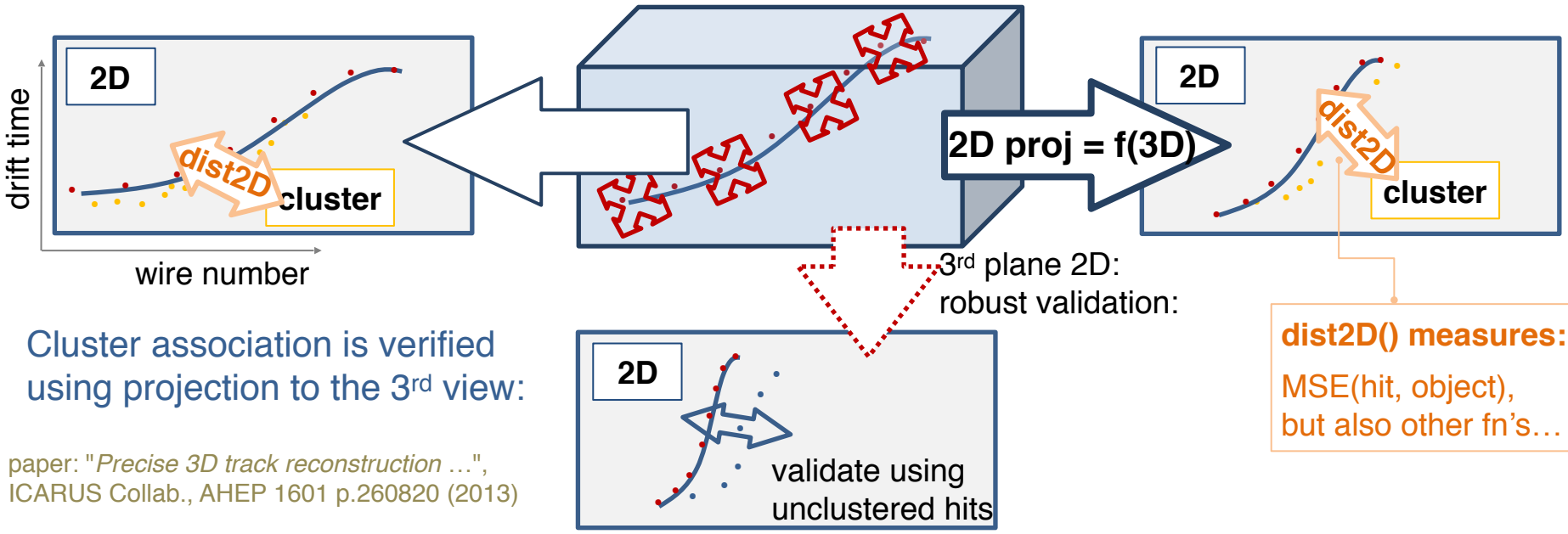




3D track matching

arXiv:1506.05348
arXiv:1708.03135

Projection Matching Algorithm (PMA)



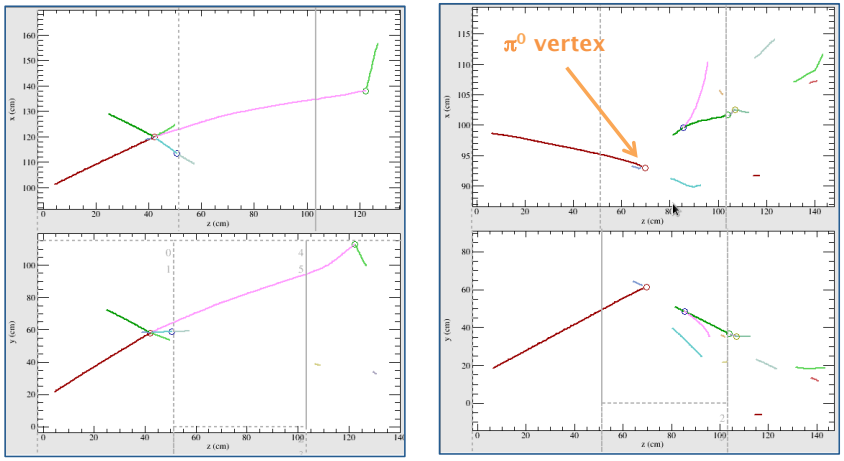
Cluster association is verified using projection to the 3rd view:

paper: "*Precise 3D track reconstruction …*", ICARUS Collab., AHEP 1601 p.260820 (2013)

dist2D() measures:

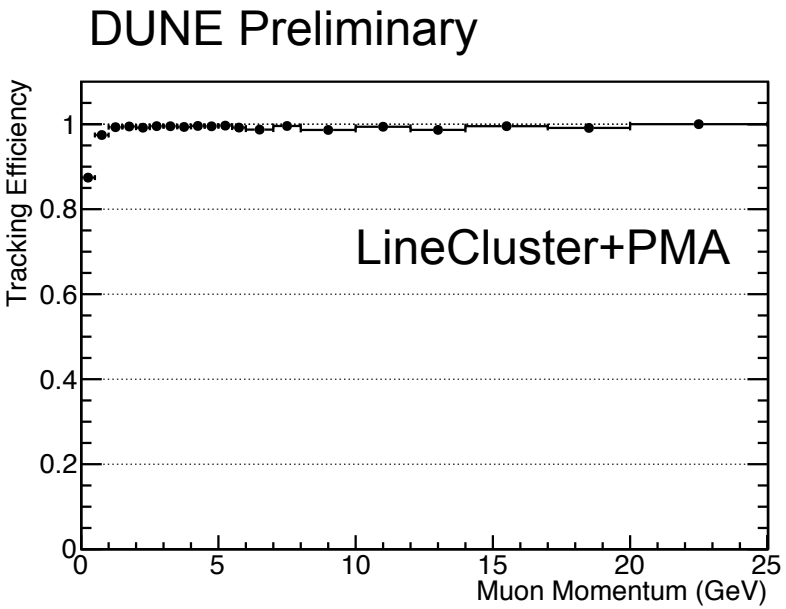
MSE(hit, object), but also other fn's…

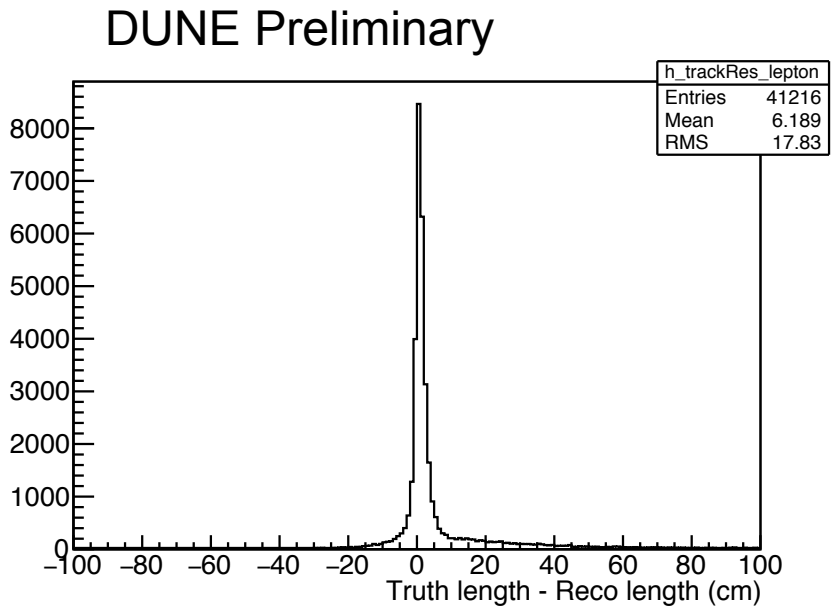
- Instead of building 3D object by matching 2D hits between different views, build 3D object by minimizing distance the object's 2D projection to 2D hits.
- Fit vertex and reoptimize tracks.



Reconstructed event examples: π^- @ 2GeV/c, 35t geometry, vertices indicated with circles, red track: incident particle.

Tracking Efficiency



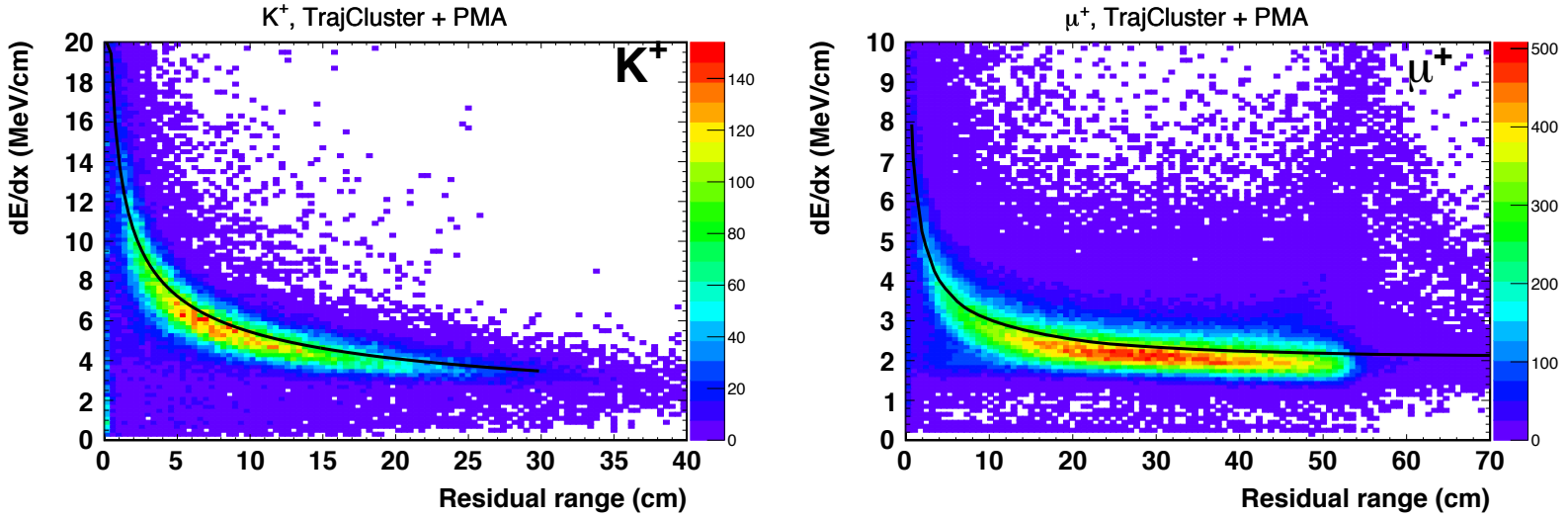


- The efficiency to reconstruct muons in the neutrino beam is very high.
- The reconstructed track length matches true length very well.

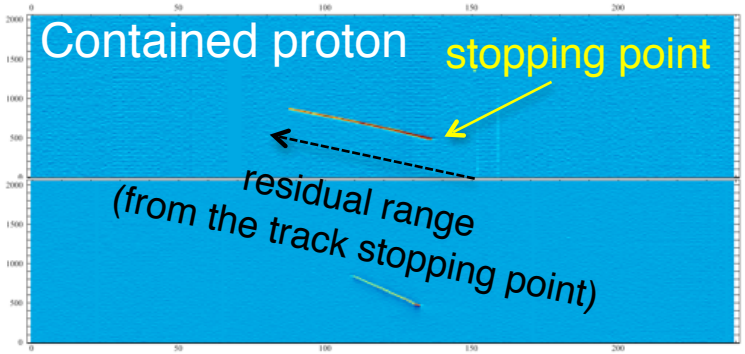
Calorimetry reconstruction

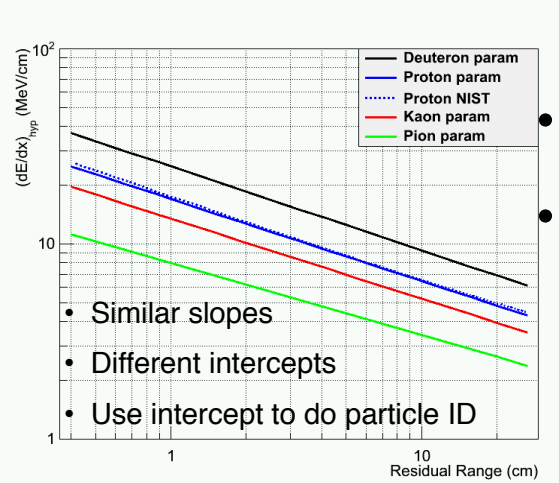
- Convert dQ/dx to dE/dx after correcting for electron lifetime and recombination effects - crucial for particle identification in LArTPC.
- Input: recob::Track
- Output: anab::Calorimetry
  - const std::vector< double > & dEdx ()
  - const std::vector< double > & ResidualRange ()

p->nubar+K



Particle Identification

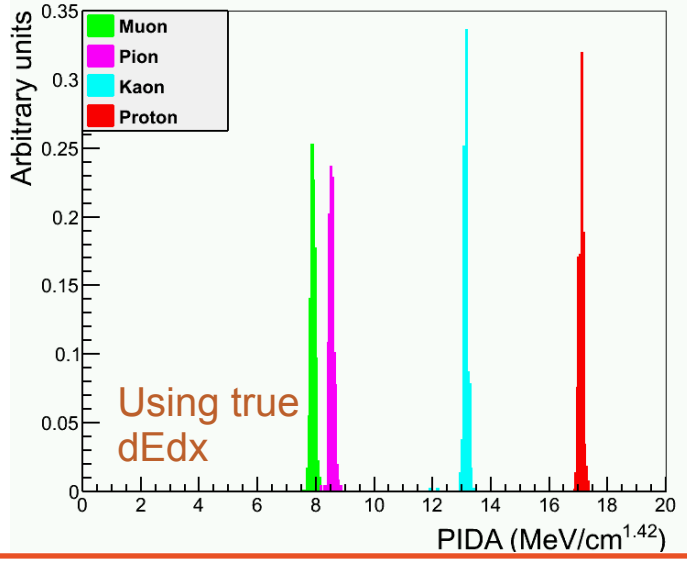




- Input: anab::Calorimetry
- Output: anab::ParticleID

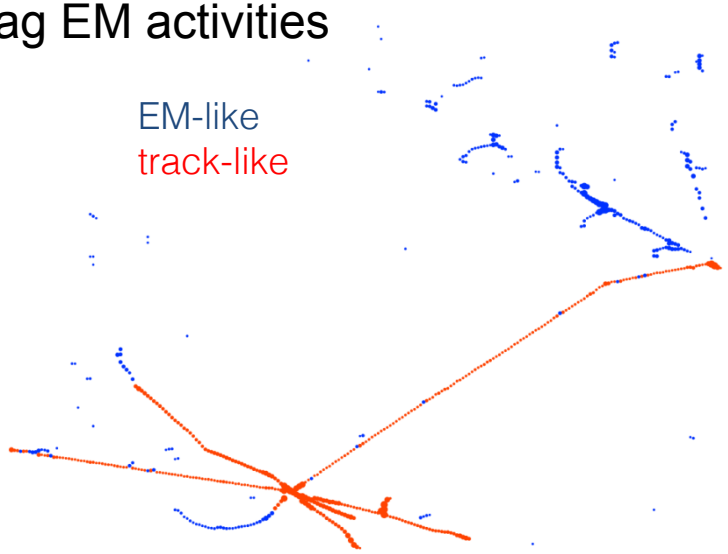
- Calorimetry information provide powerful particle ID.
- <Ai> for each track
- Ai = $(dE/dx)_{calo}$ $R^{0.42}$
  - R is residual range

arXiv:1306.1712

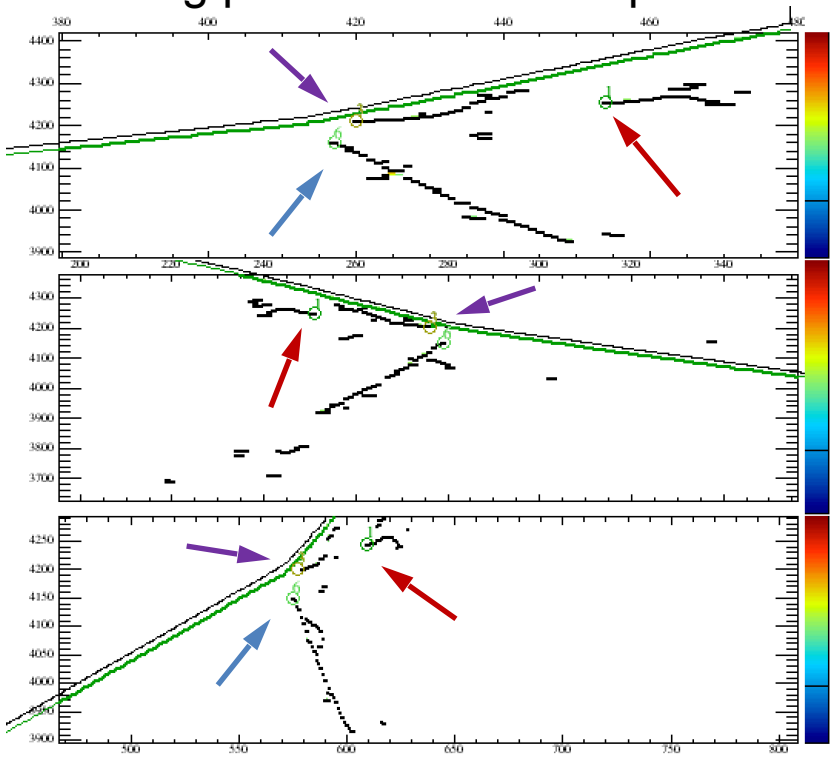


CNN in LArTPC reconstruction

Tag EM activities

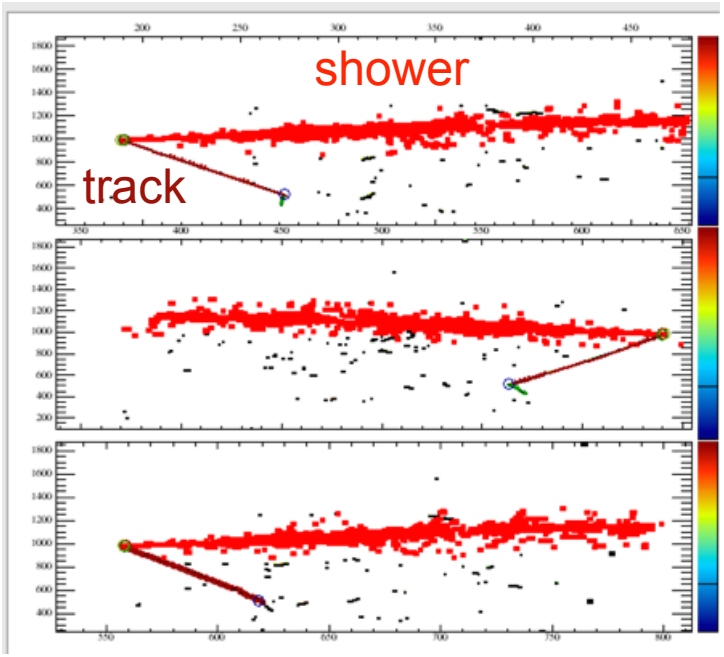


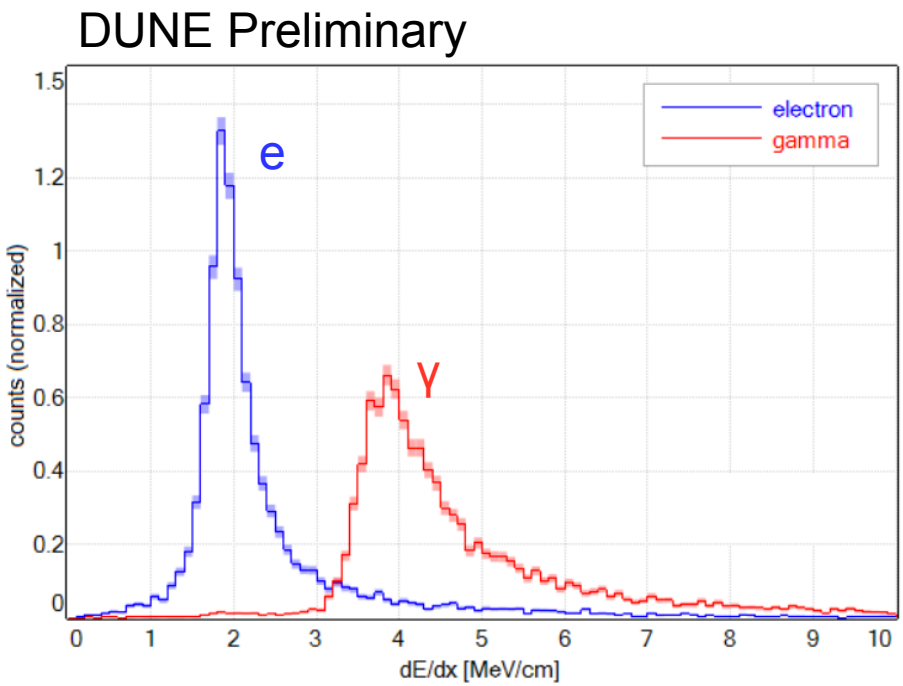
Tag photon conversion points



- ProtoDUNE team is pioneering “pixel level” feature labeling.
- Tensorflow is recently included in the LArSoft package suite.
- Input: recob::Wire, recob::Hit, recob::Cluster
- Output: anab::FeatureVector

Shower Reconstruction

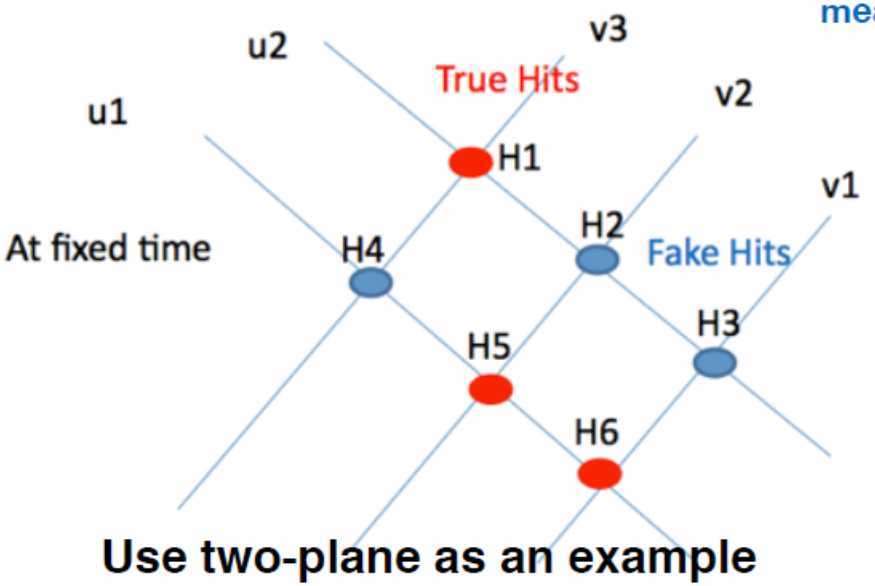




- Current shower reconstruction in DUNE:
  - Pandora pattern recognition
  - CNN track/shower ID
  - EMShower (shower maker)
- Average dE/dx of the first 1 cm of shower provides information for e/gamma separation.

Wire-Cell Reconstruction

- 3D Tomographic reconstruction
- Reduce degeneracy by using the charge information: same charge in a voxel is measured 3 times by wires on the three wire planes



measured charges on wires

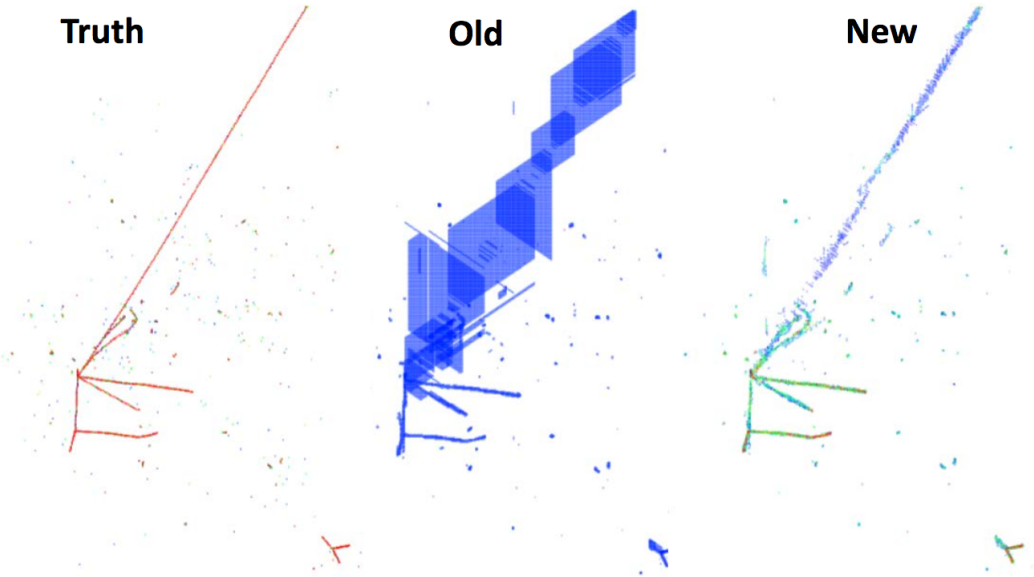
true charges to be solved

$$\begin{pmatrix} u1 \\ u2 \\ v1 \\ v2 \\ v3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & a & a & a \\ a & a & a & 0 & 0 & 0 \\ 0 & 0 & a & 0 & 0 & a \\ 0 & a & 0 & 0 & a & 0 \\ a & 0 & 0 & a & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \\ H_5 \\ H_6 \end{pmatrix}$$

Matrix determined by geometry

(a=1)

Compressed sensing in wirecell

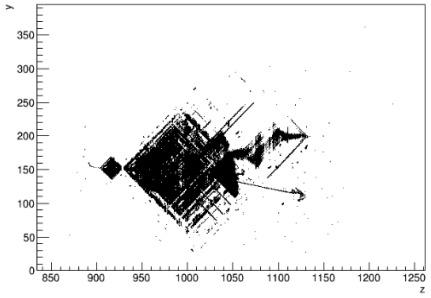


- L1 regularization - most physics signal is zero (sparse)
- More accurate reconstructed information
- Huge improvement on speed: a few hours -> half a minute
- Good progress on Wire Cell Toolkit and LArSoft integration

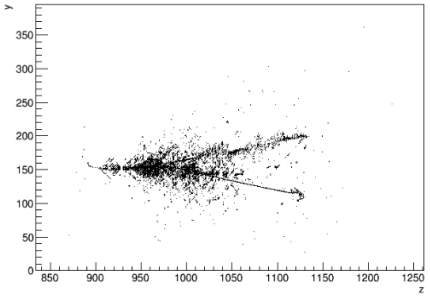
Converting 2D hits to 3D points (SpacePointSolver)

- A new method to convert 2D hits to 3D points
  - Related to WireCell, but independent work
- Problem is to distribute collection hit charge over the coincident induction wires to minimize errors on induction wire predictions
  - Regularization helps 3D imagine reconstruction

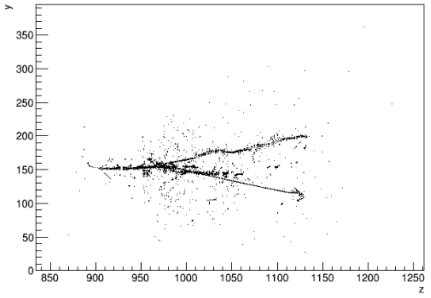
Hit matching



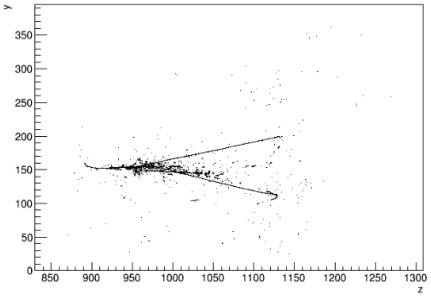
Unregularized



Regularized



True



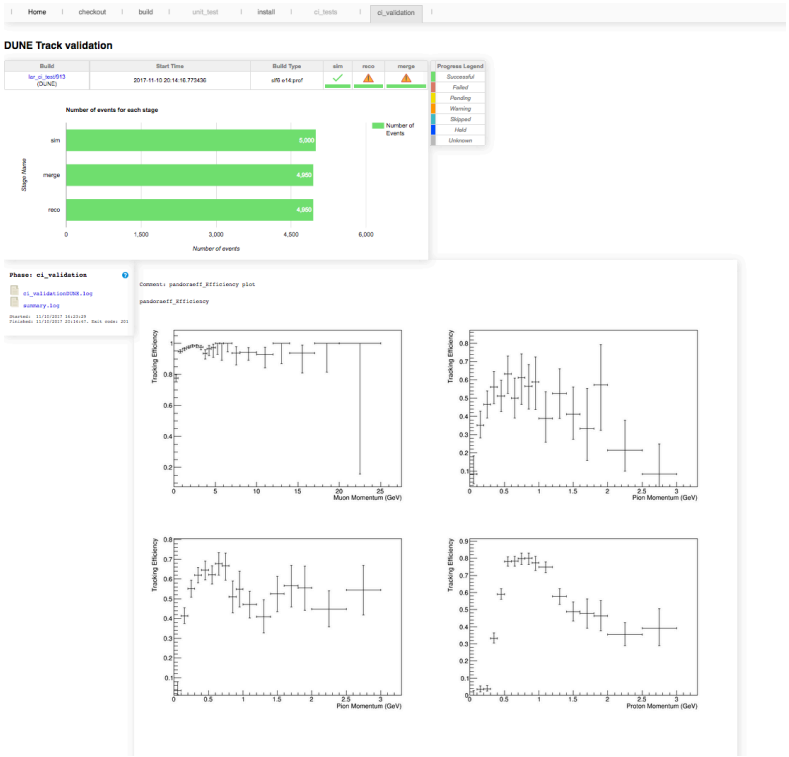
- O (10s)/event for the FD neutrino sample
- Input: recob::Hit
- Output: recob:SpacePoint

Other reconstruction/analysis algorithms

- Neutrino energy reconstruction
  - Works well for nue CC and numu CC events (DUNE-doc-2278)
  - Information saved in ntuples.
  - We need to create data products to save information in art files - action item for this week.
- CVN neutrino selection
  - Provides CVN scores for nueCC, numuCC, nutauCC, NC events.
  - Gives best sensitivity for oscillation analysis.
  - We need to create data products to save information in art files - action item for this week.

CI validation

- http://dbweb6.fnal.gov:8080/TestCI/app/ns:dune/build_detail/phase_details?build_id=lar_ci_test/913&platform=Linux%202.6.32-696.1.1.el6.x86_64&phase=ci_validation&buildtype=slf6%20e14:prof
- Automatically submit jobs to run standard neutrino simulation, reconstruction and evaluation of tracking efficiency for different reconstruction algorithms and particles.
- Plan to include validation of shower reconstruction as well.



Conclusions

- After many years of dedicated efforts, the event reconstruction in liquid argon TPC has reached an advanced level.
- There is a fully automatic reconstruction chain in DUNE to do signal processing, pattern recognition, track and shower reconstruction and particle identification.
- The common framework LArSoft has boosted the reconstruction development.