Projection Matching (PMA)

Dorota Stefan, Robert Sulej

ProtoDUNEs Science Workshop

June 29

Outline

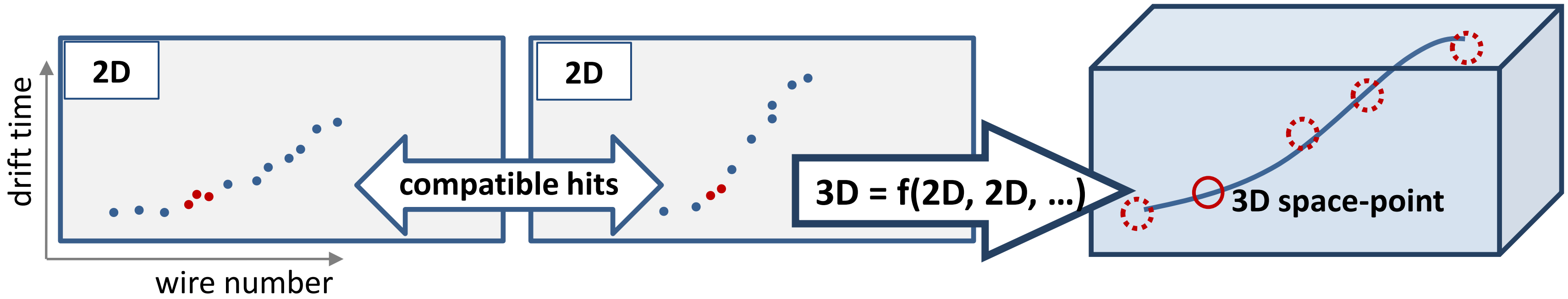
- How *Projection Matching Algorithm* works?
- Possible reconstruction chains using PMA in LArSoft framework.

PMA reconstruction of:

- trajectory and dE/dx
- track–vertex structure
- shower direction

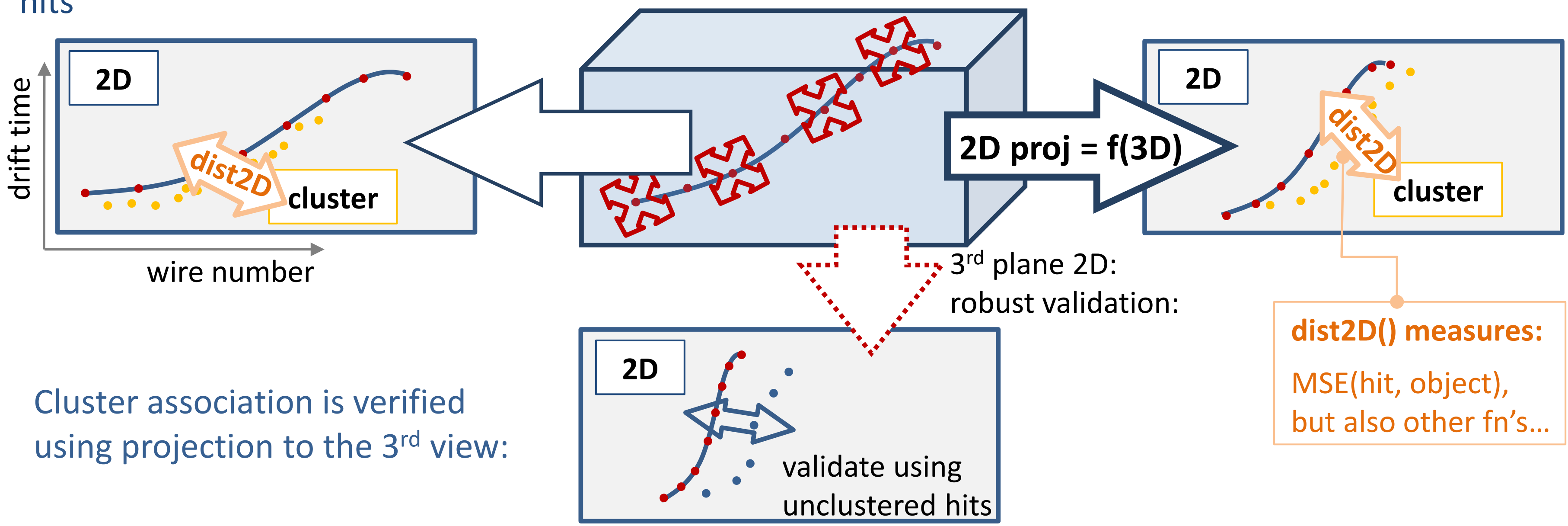
PMA 3D approach: what is different from other approaches

usual: *n* compatible 2D hits → 3D space-point → fit 3D trajectories to space-points



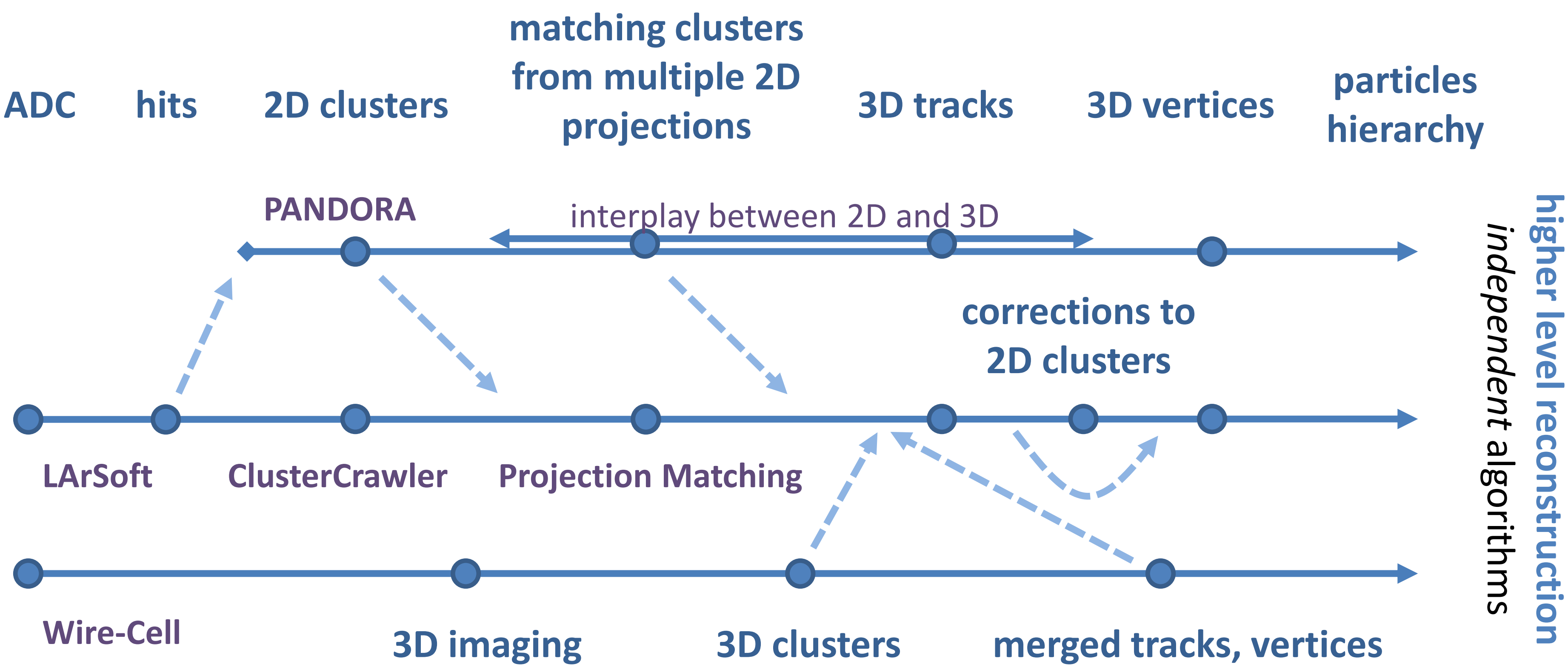
other way around: 3D object(s) ↔ hits or raw ADC in multiple 2D views

build 3D (***single track*** or ***full track structures***) to minimize distance the object's 2D projections to 2D hits



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

PMA place in the reconstruction chain



- **input: pattern recognition at various stages, best if *shower-like* / *track-like* tagged**
- **output: 3D particle hierarchy (3D tracks + vertices, associations to 2D clusters and hits)**
  - hierarchy stored in PFParticle objects in LArSoft
  - can fit single tracks, or cascade directions with or w/o fixed vertex position
  - dQ/dx sequence for each view

PMA features

- **hit – hit association is not needed**, each 2D hit has its own 3D position on the trajectory, it is ***independent*** from hits in other projections
  - reconstruction can use **2 or 3 views**; even sections with only 1 view are still useful (in case of e.g. dificult track orientation, hit or cluster inefficiency, hardware problems, …)
  - hit charge is positioned along the trajectory without bias of association to a hit(s) in complementary view -> very good dE/dx

- **full 3D objects are driven directly by 2D information;** no intermediate step with 3D hits/points to be refitted again into tracks in 3D space

- the optimization can take into account also 3D points: vertices, feature points, …, it is now used to support **wire-plane-parallel** orientation: clean endpoints (or entry/exit points) of tracks are detected

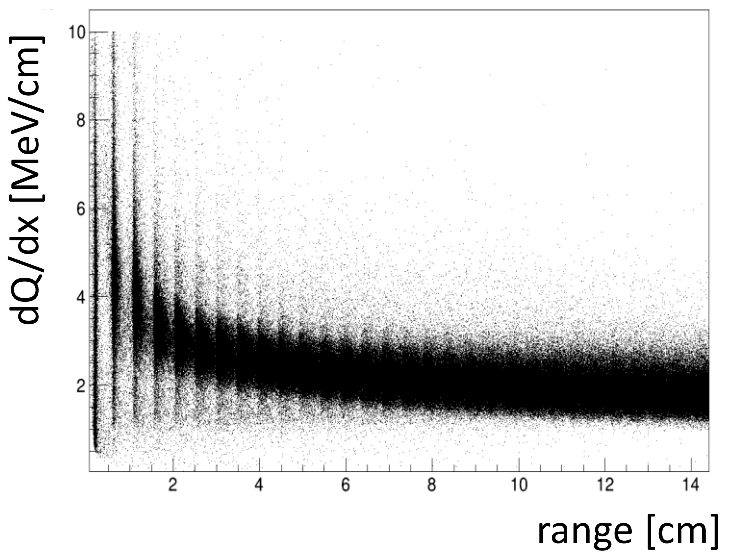
- **basic idea can be widely extended** to many aspects of reconstruction – next slides

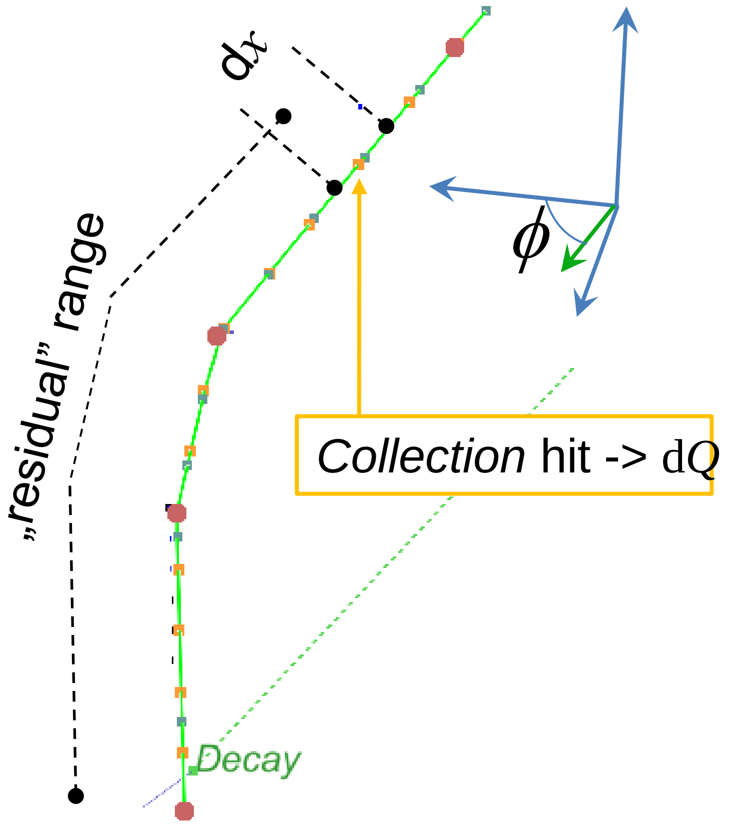
- **distortions (space charge, nonuniform E field) can be accomodated** in the 3D→2D projection function used during the optimzation if displacement mapping known

dE/dx output from *Projection Matching*

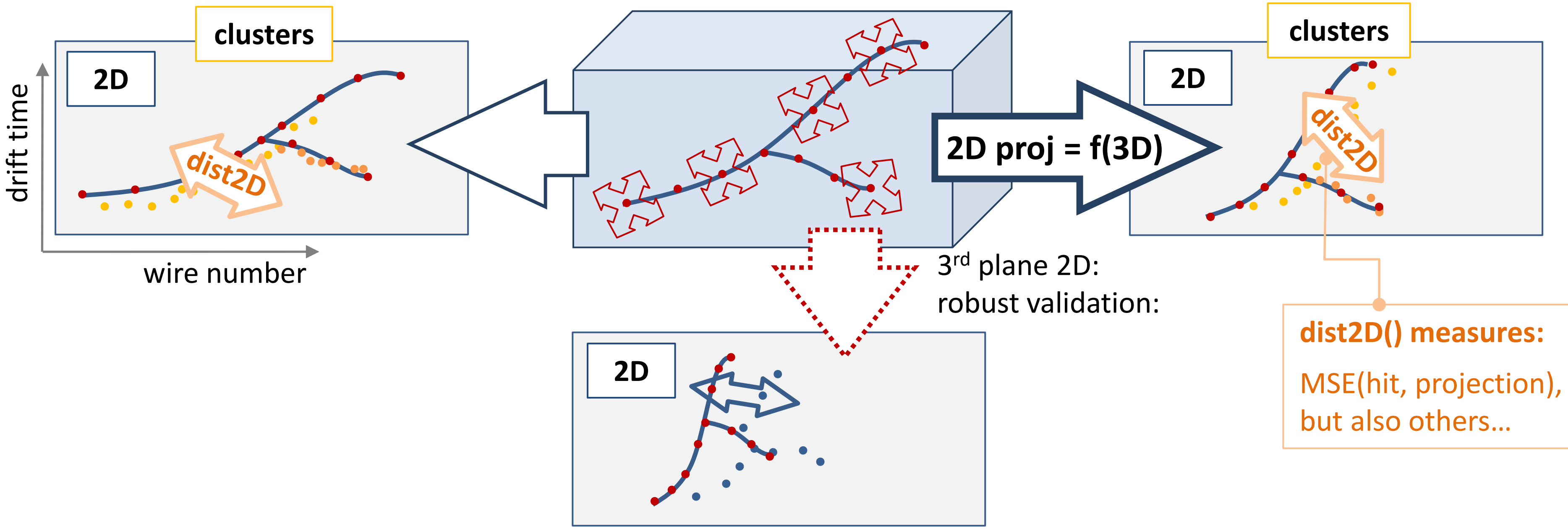
- *Collection* hit projected to 3D
- *Induction* hit projected to 3D
- reco track nodes
- reco track

dQ/dx vs range of reconstructed muons in protoDUNE





How vertices are reconstructed

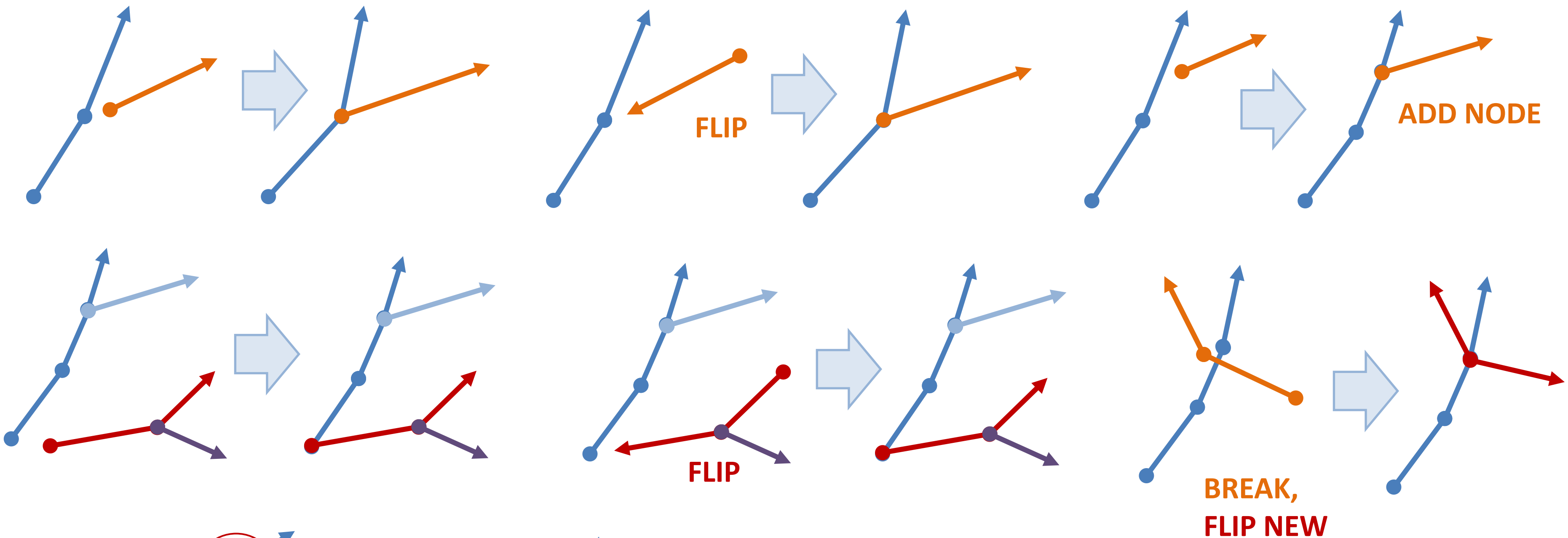


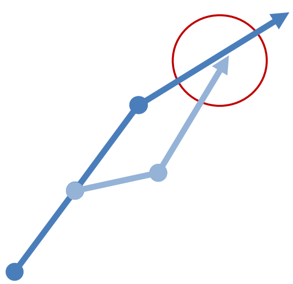
Optimization can be applied to complex structures of tracks interconnected with vertices:

- Build, extend, stitch isolated track first
- Find candidates for track intersections and points where multiple tracks meet
- Join tracks in vertex candidates, reoptimize all tracks together
- **local information from multiple tracks → fit the best position of vertex and track directions**
- Can further improve 2D pattern recognition output
- But needs implementation of complex manipulations on track merging / splitting / …

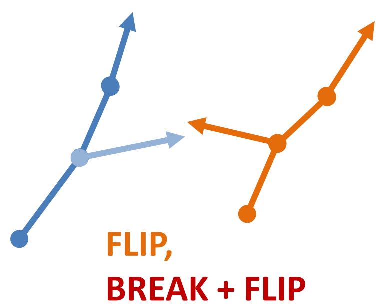
Vertices + trajectories grammar

some basic examples of all possibilities





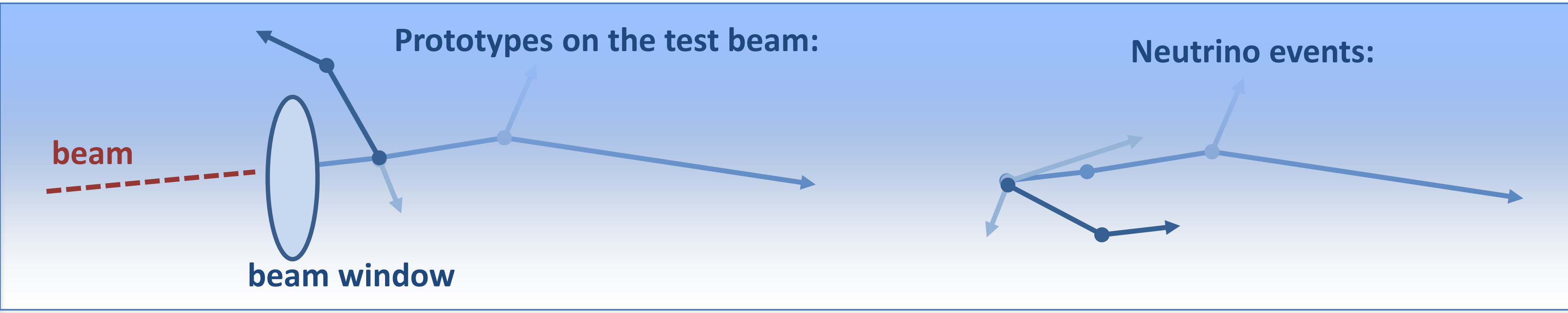
loops **not allowed,**
only *tree* structures

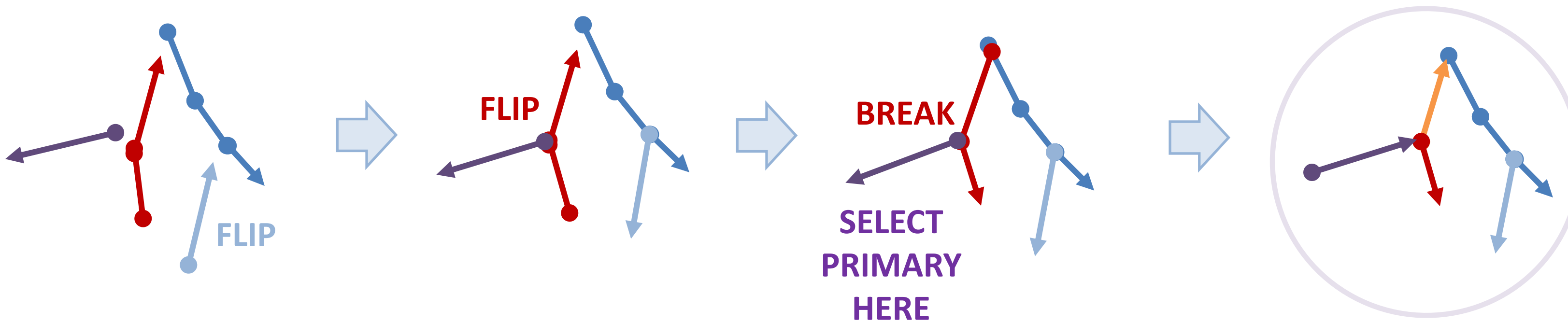


not implemented now,
=> vertex missed

If reoptimization of new structure fails, then initial configuration is recovered.

Where is the primary vertex?

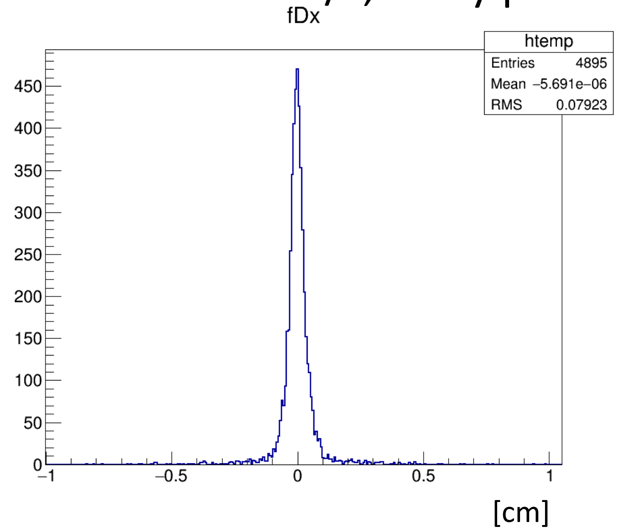


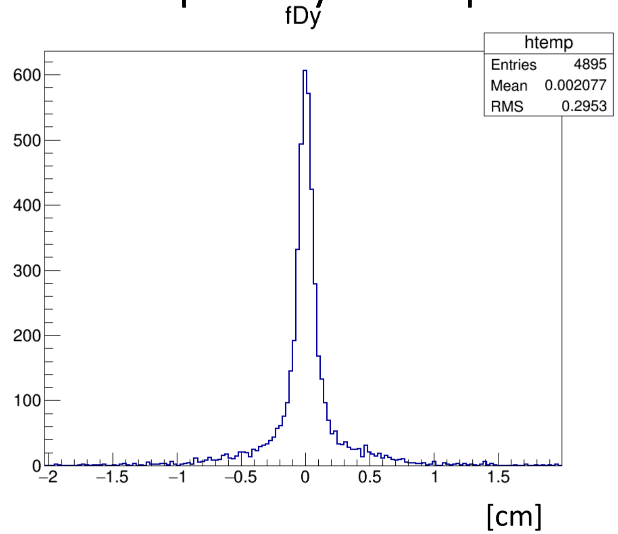


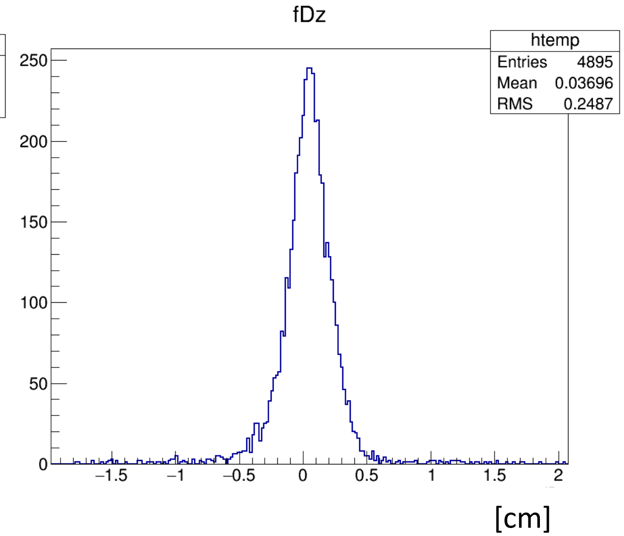
- may be trivial: most upstream vertex for ν beam, but this fails often,
- test beam events: known beam entry position, direction → select best matching particle,
- small events (proton decay, low energy neutrino): detect stoppers by dE/dx, adjust directions in the track system accordingly,
- large events: <u>to be developped</u>, primary vertex selection by track orientation and energy at reconstructed vertices.

Vertex reconstruction resolution

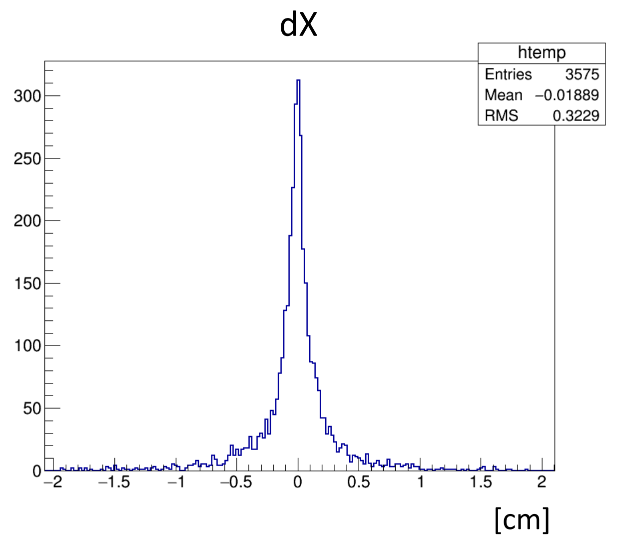
Protons 2GeV/c; entry position of the primary beam particle

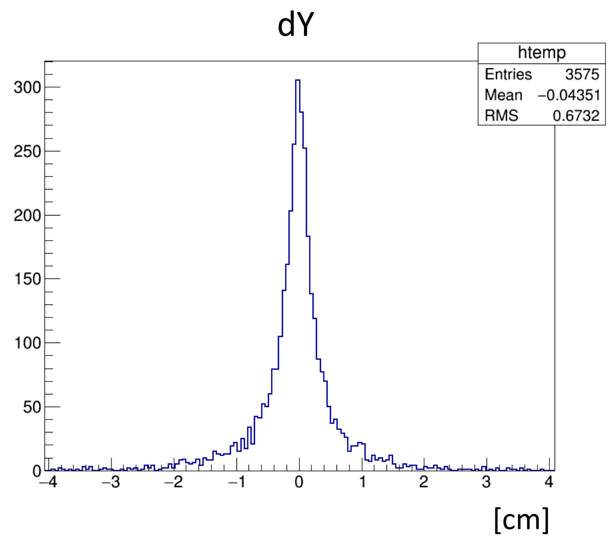


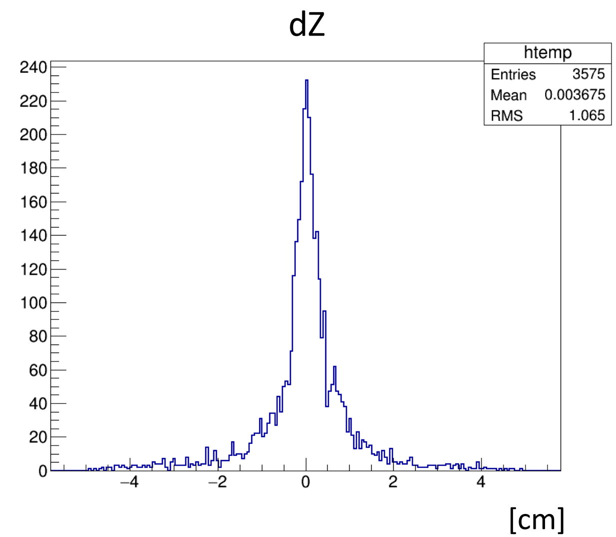




Protons 2GeV/c; first inelastic vertex position, if > daughter with E_k > 50 MeV



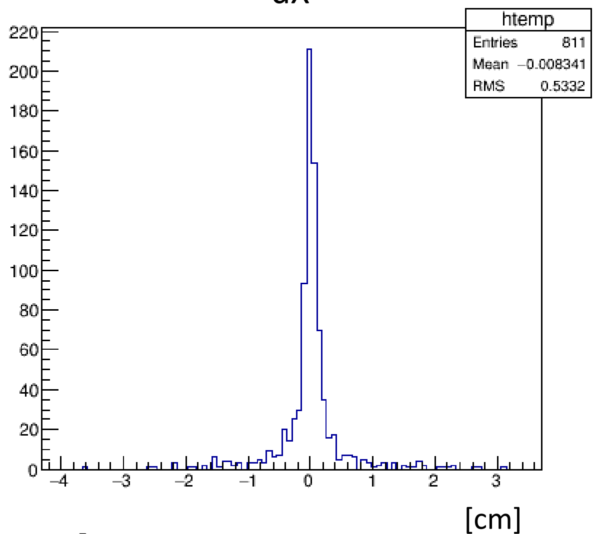




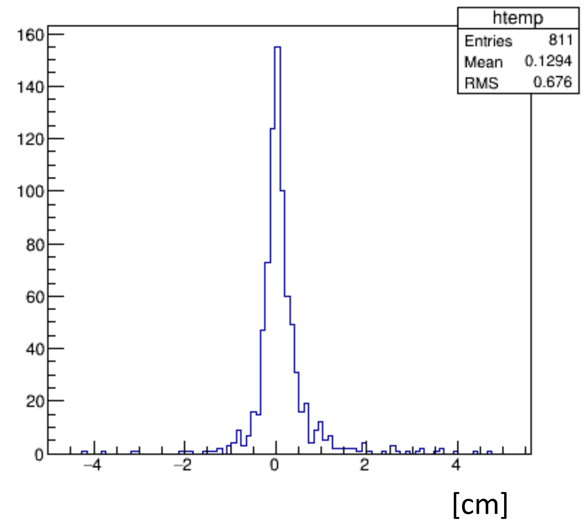
Vertex reconstruction resolution

π^0 in the hadronic system

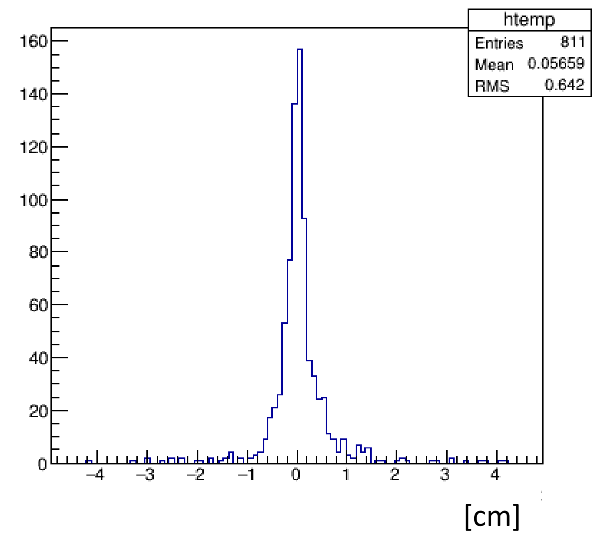
dX



dY

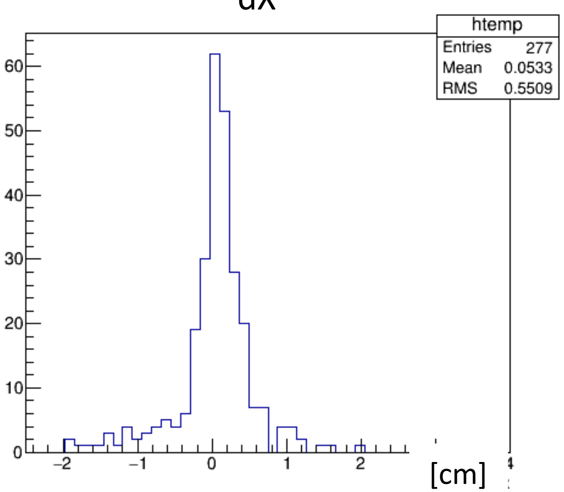


dZ

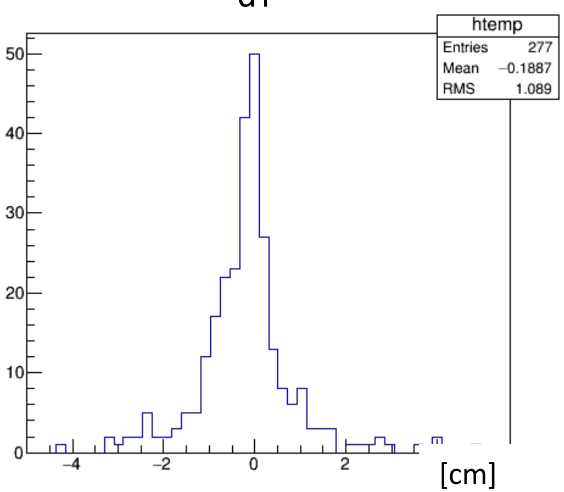


$\mu \rightarrow e$ decay

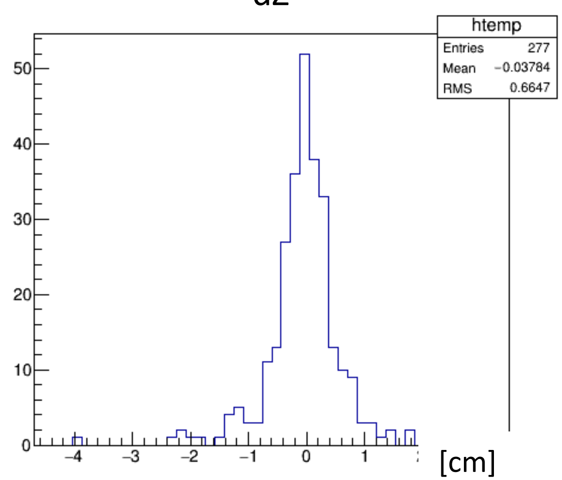
dX



dY

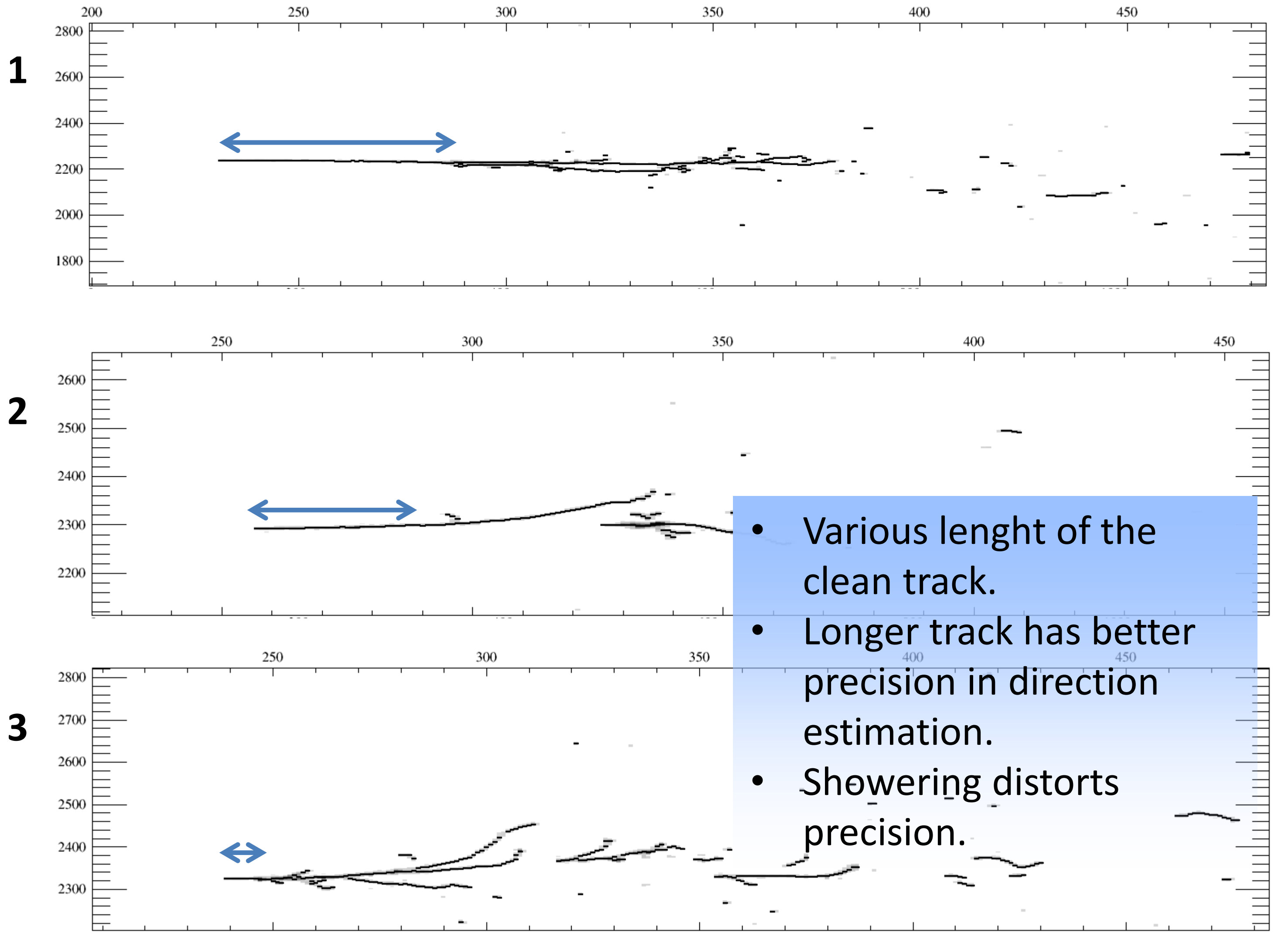


dZ



Showers

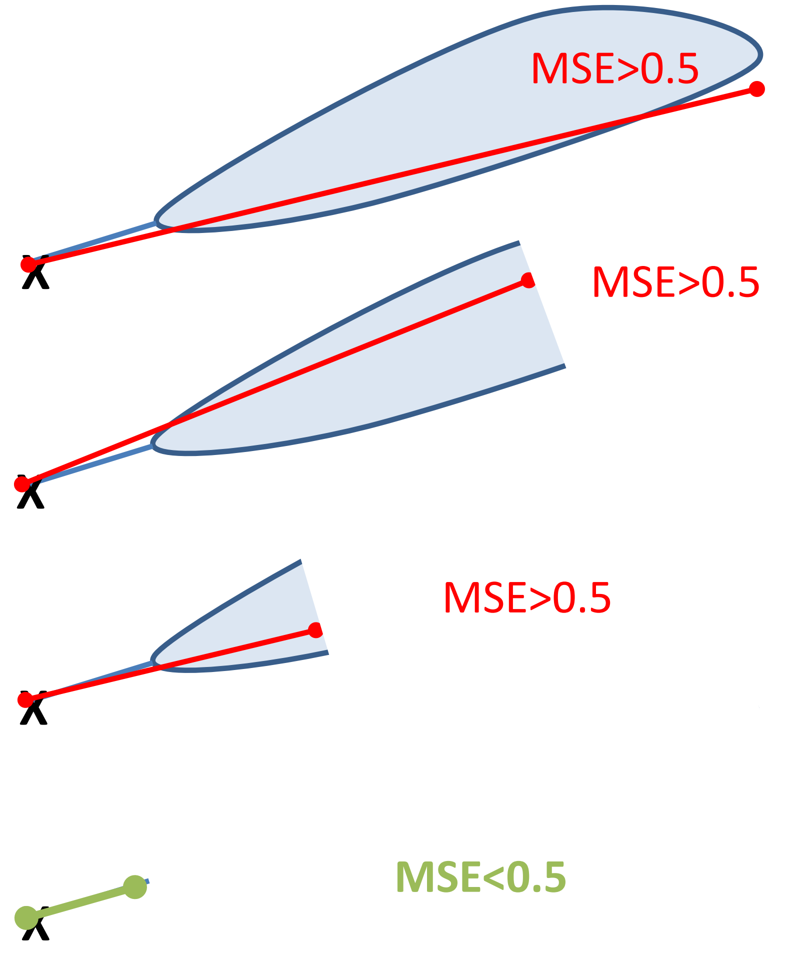
Electrons, the same direction generated



- Various lenght of the clean track.
- Longer track has better precision in direction estimation.
- Showering distorts precision.

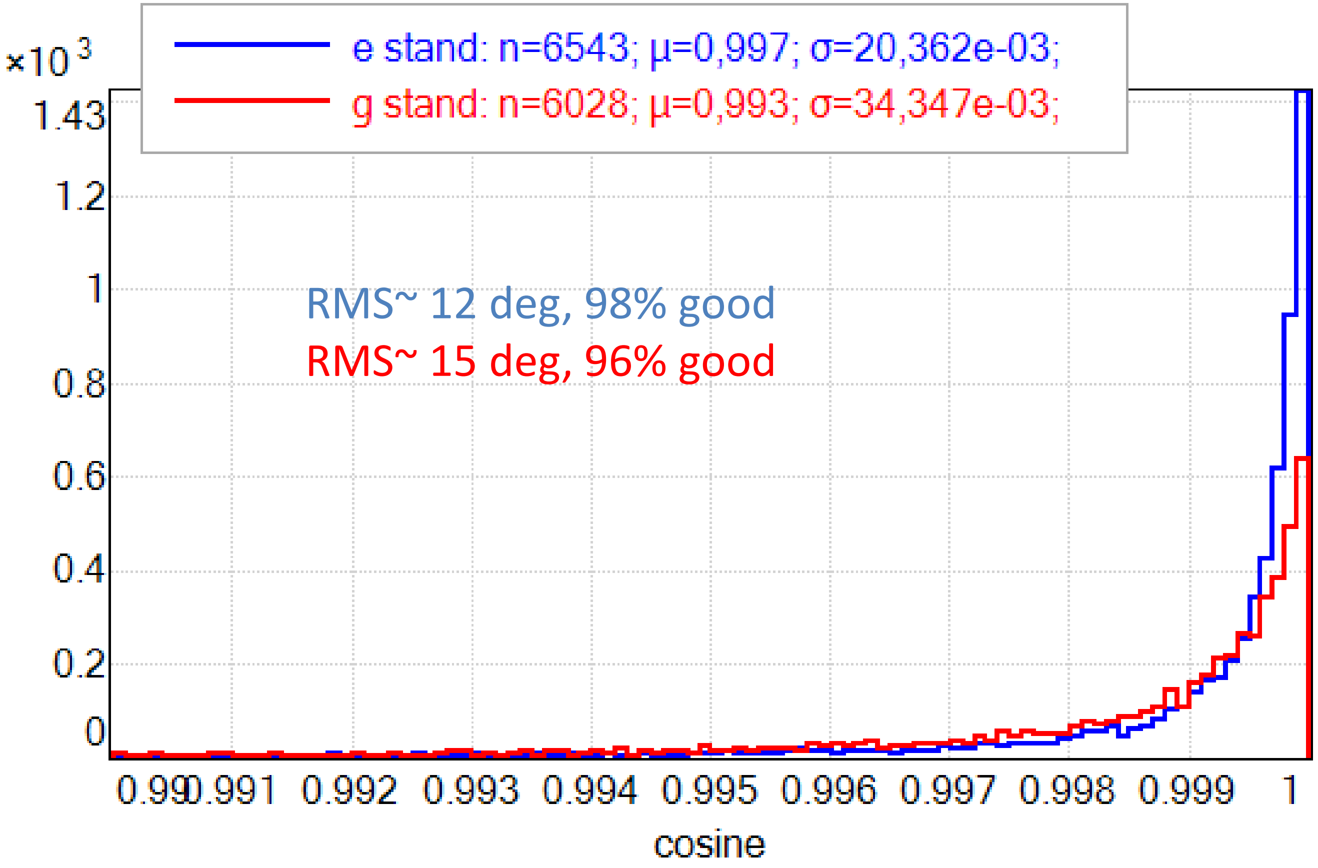
Shower direction reconstruction

1. Input: pattern recognition provides PFParticle tagged as EM shower, with associated clusters and vertex.
2. Compact group of hits is selected in each view, small & isolated fragments are not used.
3. PMA segment is fitted to all hits from the shower; one endpoint is fixed at the vertex.
4. Hits with high range from vertex are removed, segment is reoptimized until MSE is low enough or min segment length is reached.
5. MSE allows to verify if direction is correctly reconstructed.

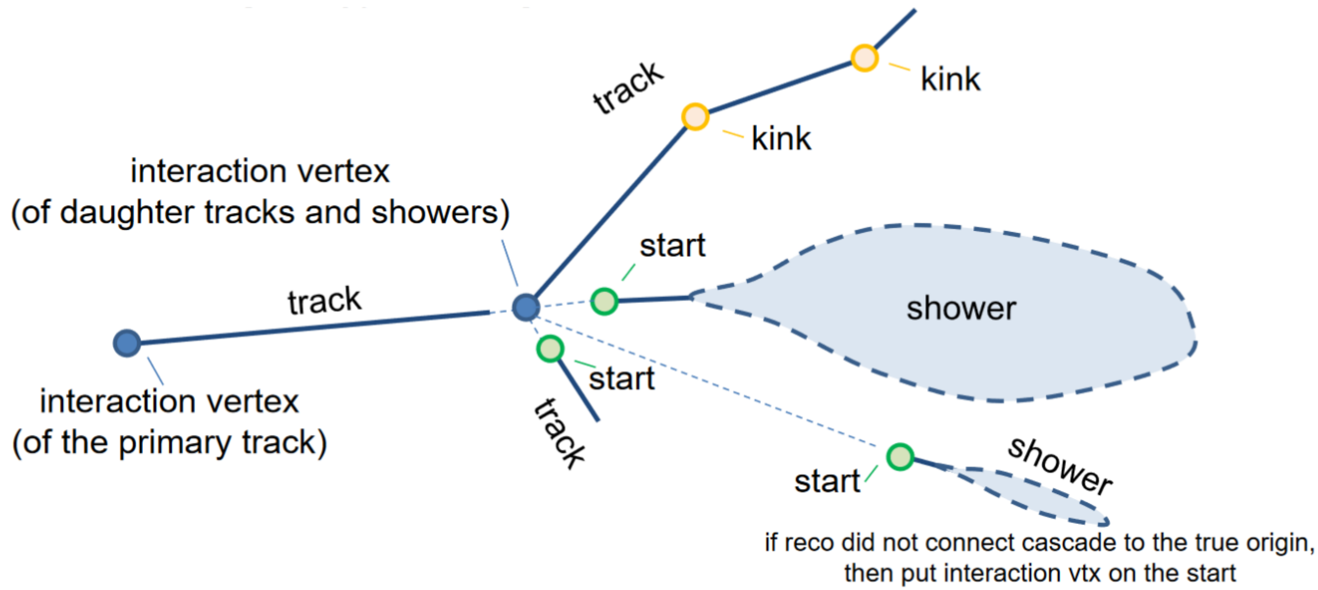


Note: Various shower orientation can prevent segment from correct reconstruction (parallel to the wire plane, initial part of the cascade in a shadow of developed cascade).

Shower direction reconstruction



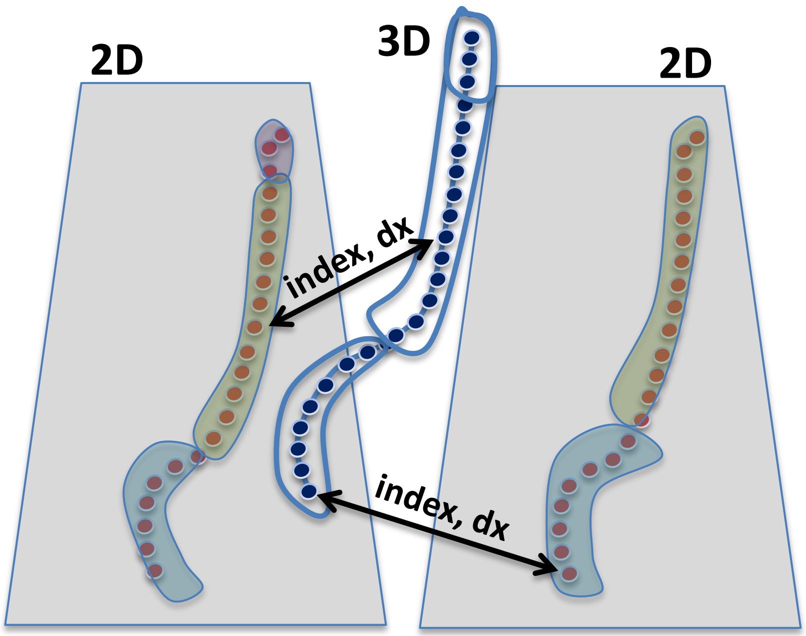
Full reconstructed structure

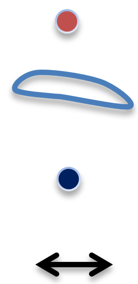


Define a convention for data products made by PMA → help navigation in analysis codes

- **only one interaction vertex** assigned to each PFParticle / Track:
  meaning: interaction point, where particle was created according to the reconstruction,
- **kinks and any other features** saved in accordingly named collections (*art*) and assinged to Tracks using the same instance name for associations,
- **still there is no strict „LArSoft policy"** for such topology descriptions.

Associations: singe trajectory

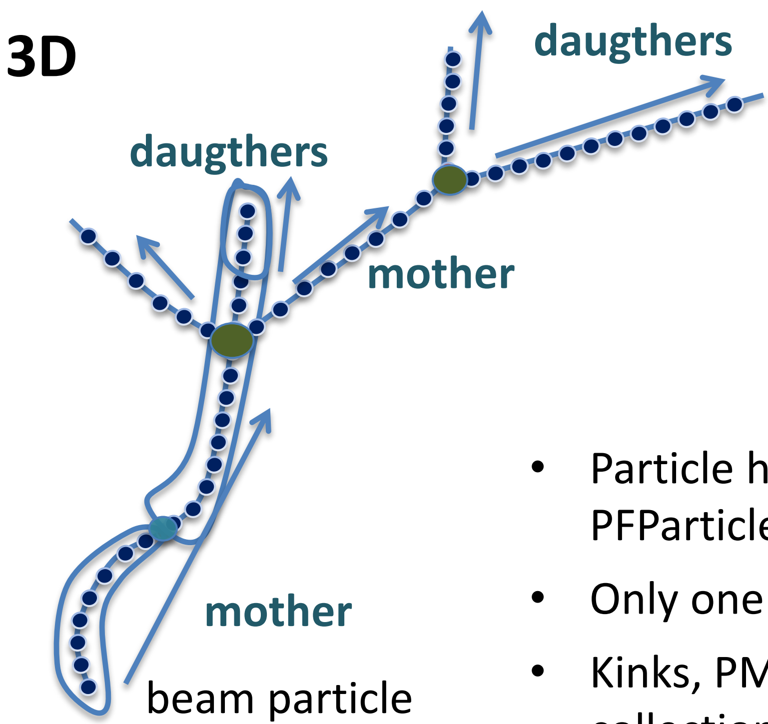


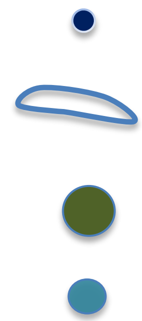


- each 2D hit is associated to its own 3D point
- metadata helps sorting hits along trajectory (index) and find dx seen by hit in its 2D view

```
// howto get dQ/dx from the track:
art::Handle< std::vector<recob::Track> > trkListHandle;
bool hasTrk = evt.getByLabel(trackingModuleLabel, trkListHandle);
art::FindManyP< recob::Hit, recob::TrackHitMeta > hitFromTrk(trkListHandle, evt, trackingModuleLabel);
// ...loop over tracks
        auto vhit = hitFromTrk.at(id);              // get vector of hits for track with id
        auto vmeta = hitFromTrk.data(id);           // get vector of metadata associated with hit-track assns
        // ...loop over hits
                size_t view = vhit[h]->WireID().Plane;        // get 2D plane of hit
                size_t idx = vmeta[h]->Index();               // get index of this hit along the trajectory
                double dx = vmeta[h]->Dx();                   // dx seen by the hit in its 2D plane
                double tdrift = vhit[h]->PeakTime();          // peak time of the hit
                double dqadc = vhit[h]->Integral();           // dq of this hit
```

Associations: track-vertex structure





3D space points

2D clusters

interaction vertices

kinks on trajectories

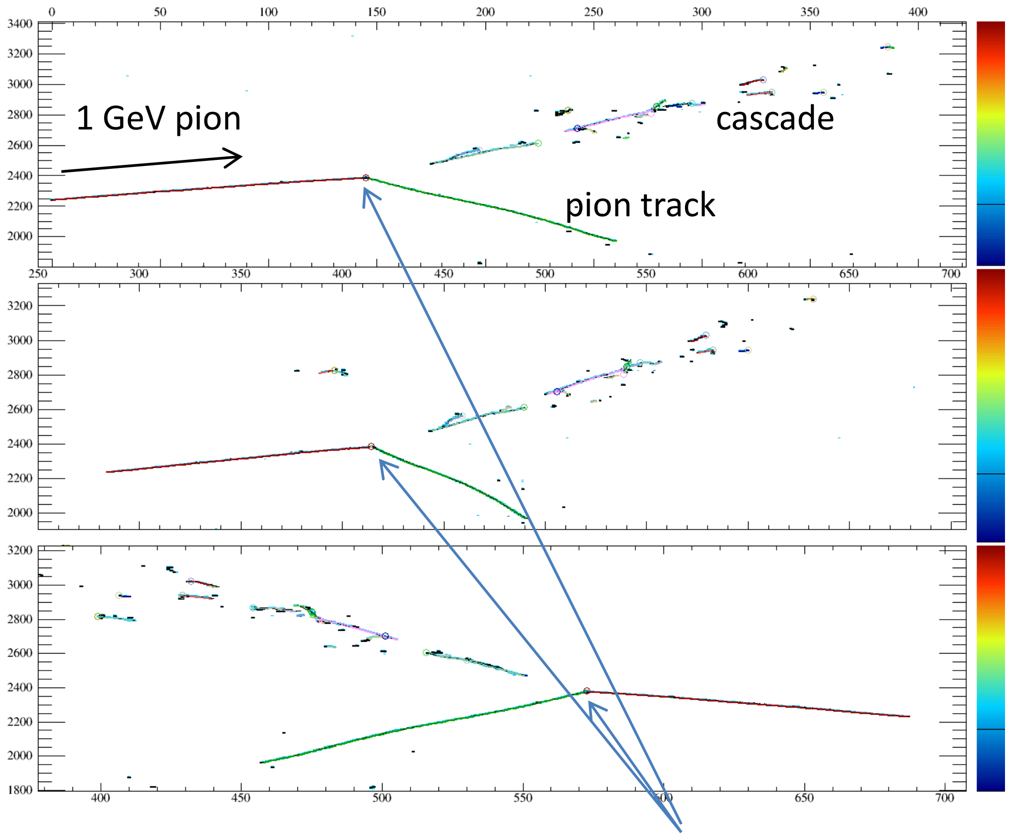
PFParticles

- Particle hierarchy (mother→daughters) described with PFParticles.
- Only one interaction vertex assigned to each PFParticle.
- Kinks, PMA nodes: saved as vertices / associated in named collections.

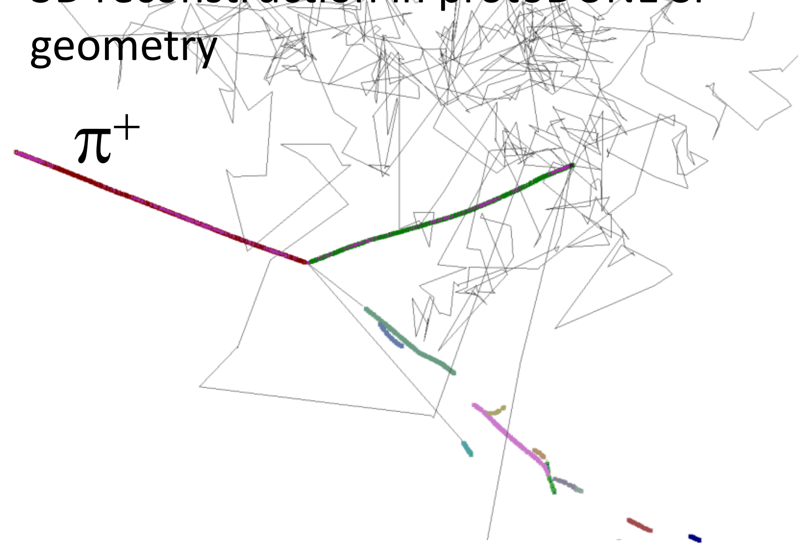
```
// howto get associations of higher level structures (this is valid today, but may change according to LArSoft policies)
art::Handle< std::vector<recob::PFParticle> > pfpListHandle;
art::Handle< std::vector<recob::Track> > trkListHandle;
bool hasPfp = evt.getByLabel(trackingModuleLabel, pfpListHandle);
bool hasTrk = evt.getByLabel(trackingModuleLabel, trkListHandle);
art::FindManyP< recob::PFParticle > pfpFromTrk(trkListHandle, evt, trackingModuleLabel);          // track-pfparticle assns
art::FindManyP< recob::Vertex > vtxFromPfp(pfpListHandle, evt, trackingModuleLabel);              // interaction vertices
art::FindManyP< recob::Vertex > vtxFromPfp(pfpListHandle, evt, art.::InputTag(trackingModuleLabel, „kink”));   // trajectory kinks
```

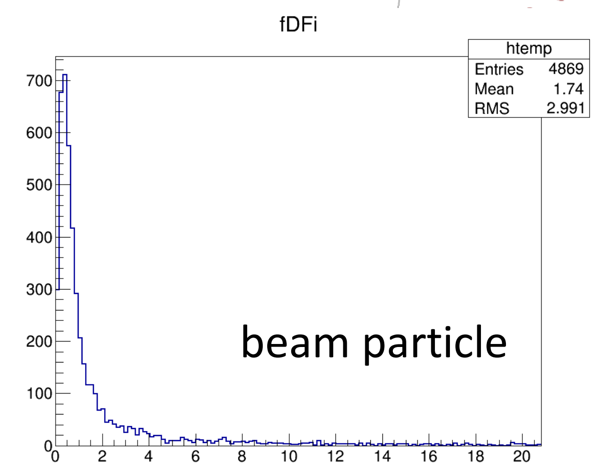
Topology description, example

2D reconstruction:
cluster crawler



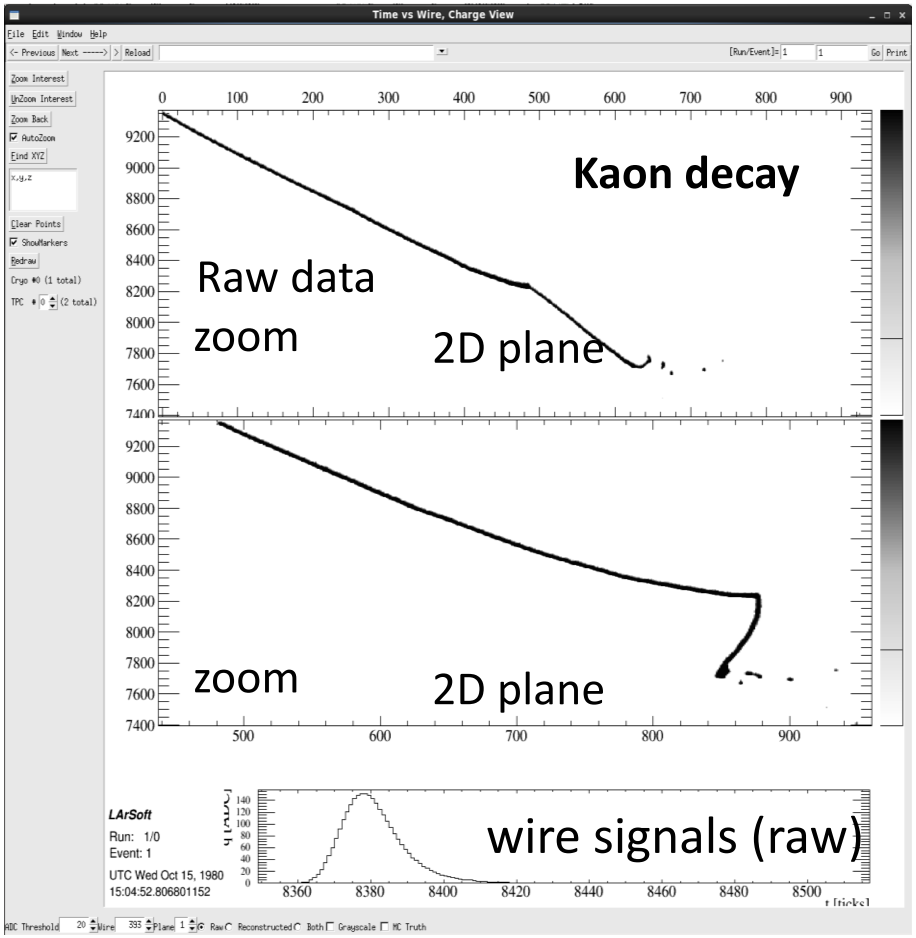
3D reconstruction in protoDUNE SP geometry

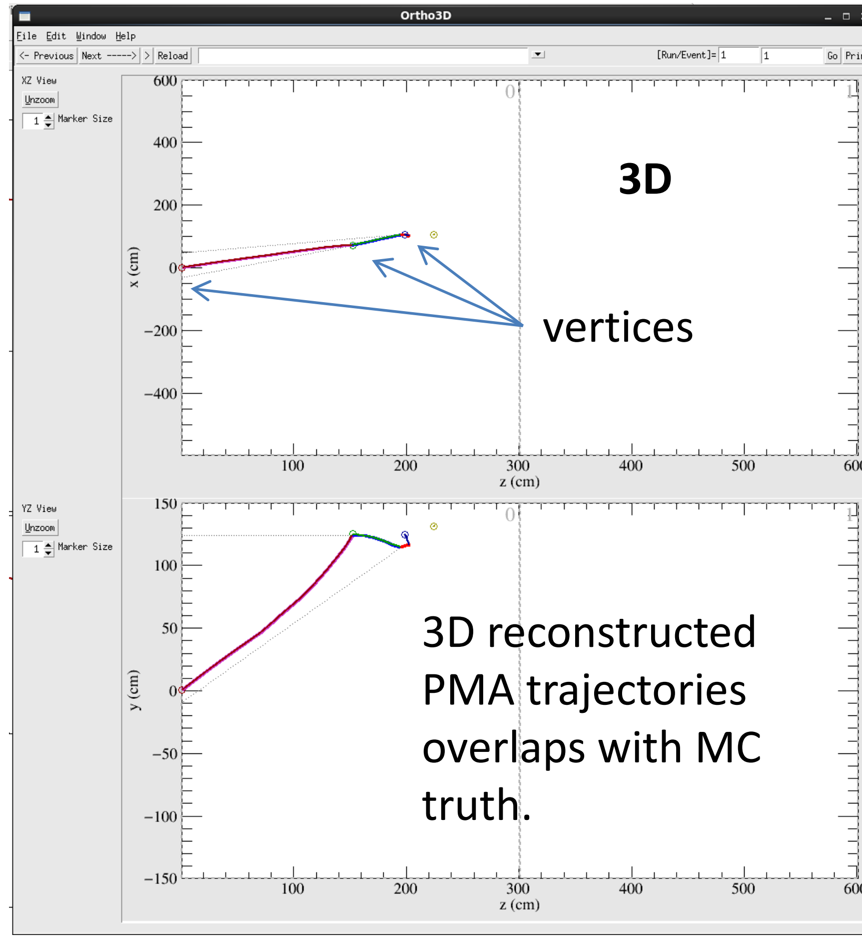




Reconstruction in dual phase detector

```
reco: [ rns, caldata, fasthit, gaushit, cchit, linecluster, pmtrack ]
```

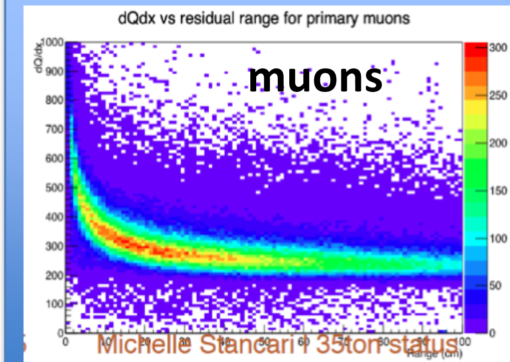


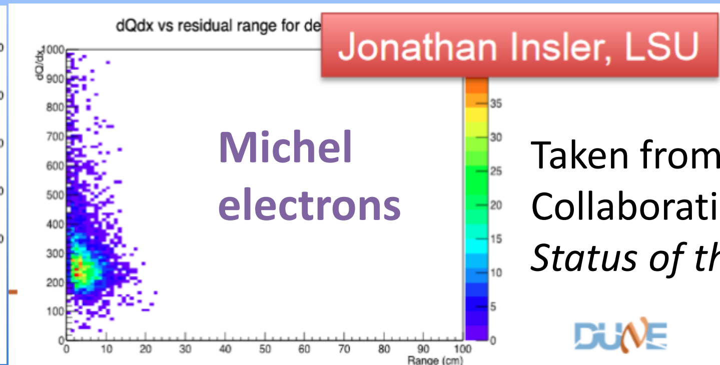


Event display for dual phase: **evd_dunefddp.fcl**

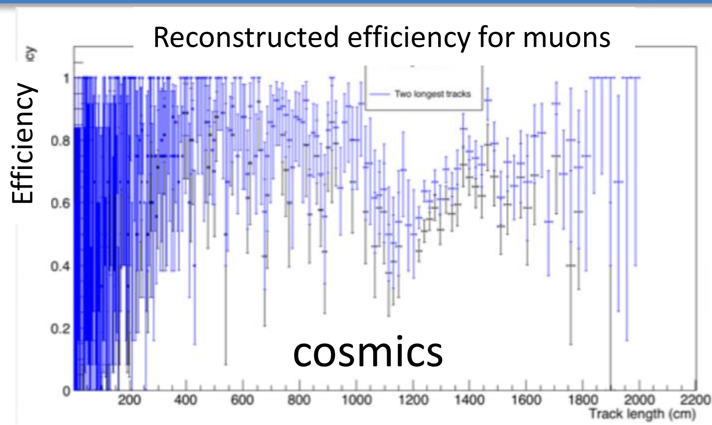
PMA is used in many physic analysis

35t



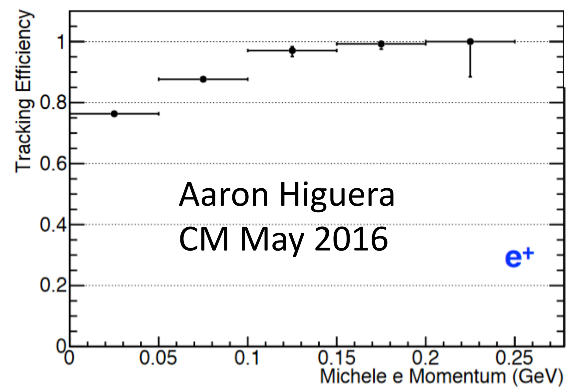


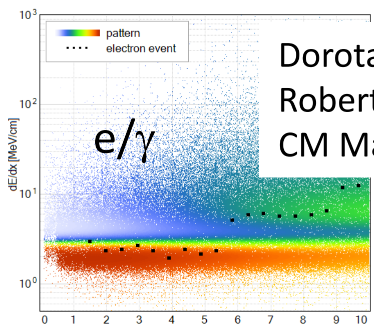
Taken from Michelle Stancari presentation on Collaboration meeting (september 2015) FNAL *Status of the 35t Prototype*



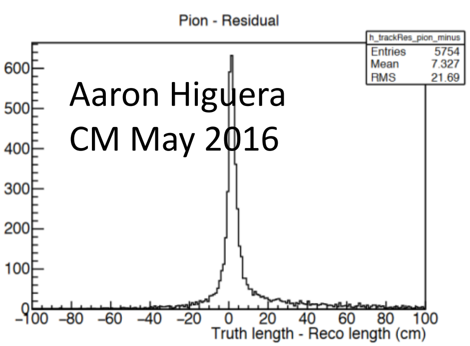
Matt Robinson,
Vitaly Kudryavtsev
Karl Warburton,
CM May 2016

FD

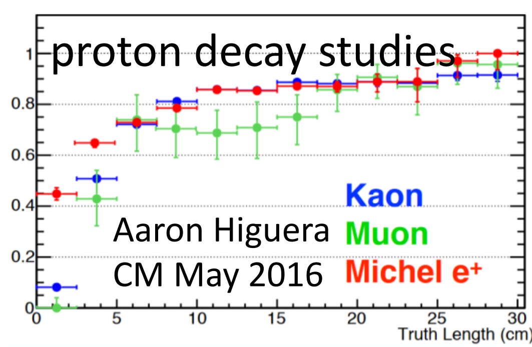




Dorota Stefan
Robert Sulej
CM May 2016



FD



Thank you!