Status and Future of Shower Reconstruction

7/14/15

Corey Adams, Yale

Goal

- Teach you what the current shower reconstruction algorithm is doing, and point out some issues.
- Suggest some low hanging improvements that are easy to reach quickly, as well as some long term features that shower reconstruction needs to be ready for analysis

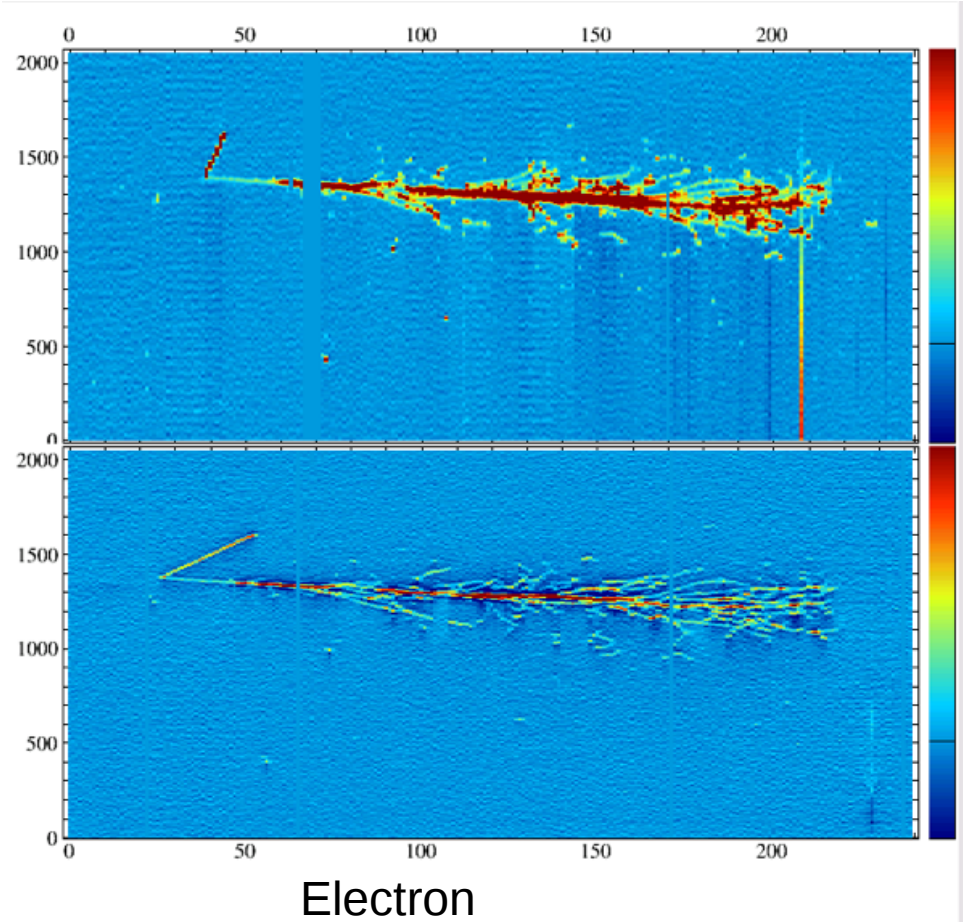
Shower reco

- Take 2 or more clusters in at least 2 planes and turn it into a 3D object known as a shower:
  - Need 2 different projections to localize it in 3D
  - Shower definition as a C++ object here:
    - larlite
    - Larsoft
- Currently shower reconstruction is pretty much contained in larlite and periodically ported to larsoft. Not a problem, really, but for state-of-the-art look in larlite.

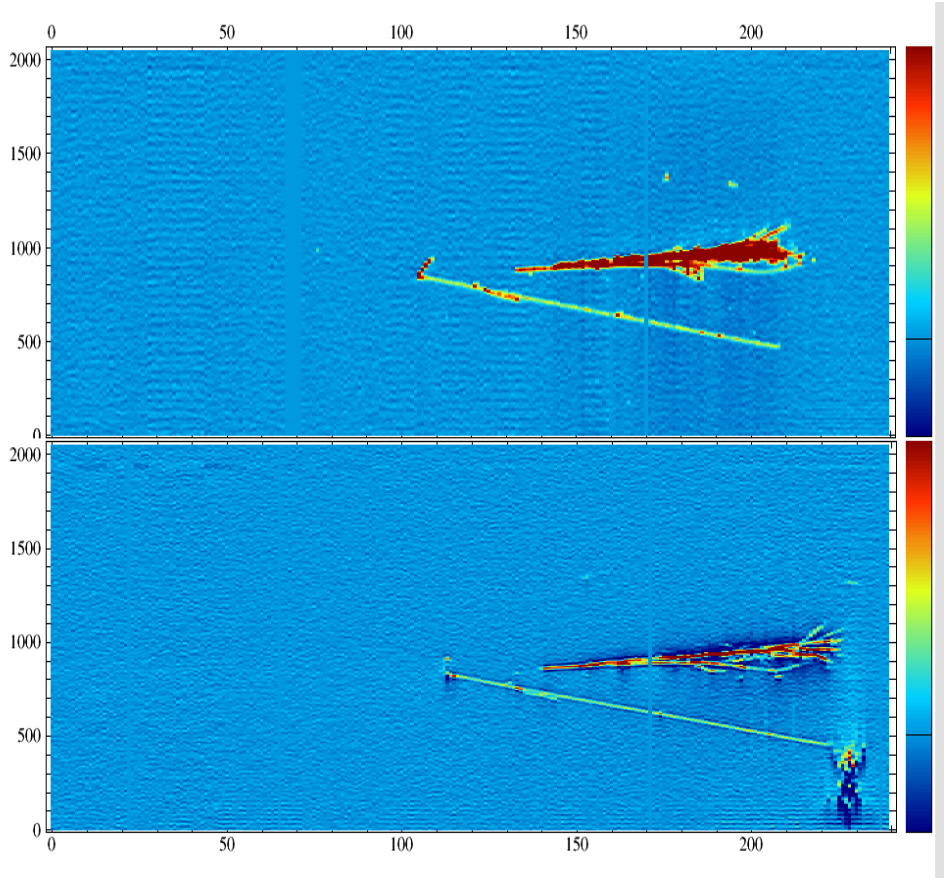
What is a “Shower”

- 3 Dimensional C++ object that is defined by:
  - **Its origin/start point (in 3D)**
  - **Its direction (in 3D)**
  - **Its energy deposition at the start of the shower**
  - And also: it's length, the angle at which it opens, the energy it deposits, and the uncertainty on some parameters.
- A shower should be associated (in the art/larsoft/larlite sense) with the clusters it was made from, and it should be able to go through a calorimetry algorithm to determine it's particle ID, but that is not part of the actual shower.
  - A Shower is a geometrical object equivalent to a track as opposed to a ParticleID Object.

Examples of Showers from ArgoNeuT



Electron



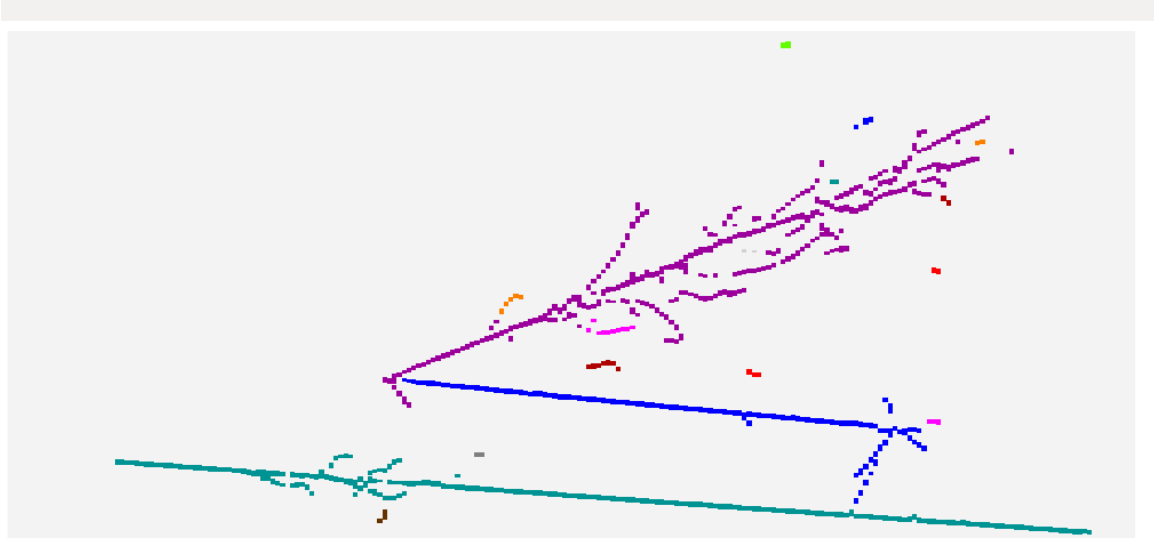
Photon

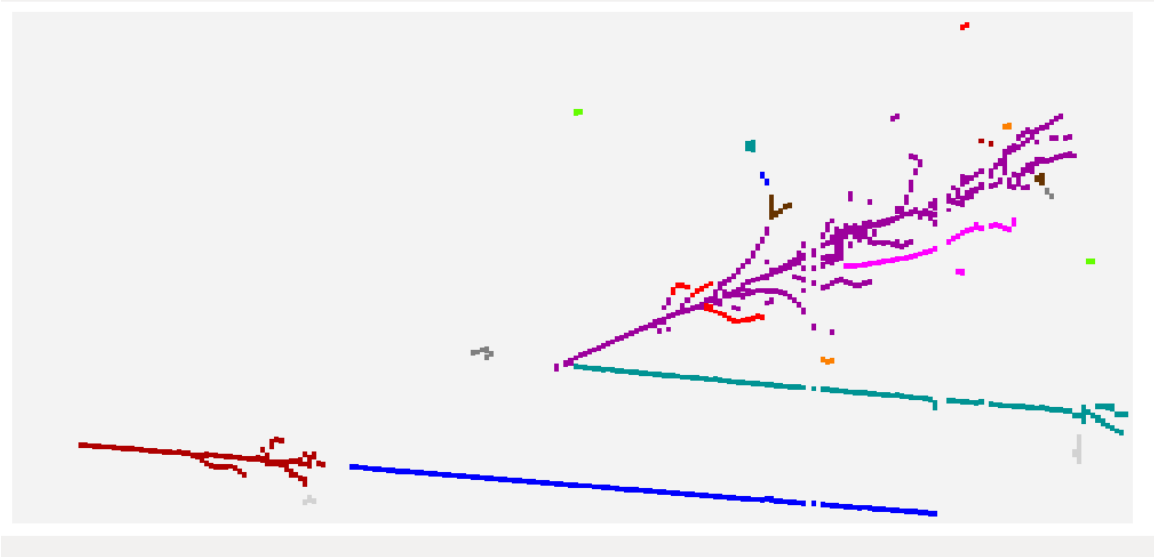
To really achieve physics goals, we need to get the 3D information about these objects!

Step -1 of Shower Reconstruction: Reco up to Clustering

- Need deconvolution, hit finding, clustering at the very minimum before shower reconstruction.
  - Having things like track finding or pandora could be very nice, too.
- Clustering is a very difficult task, and though it remains on the “ToDo” list it isn't the subject today.

Good Clustering IS Possible





- Regardless of the clustering scheme used, I'm confident that clustering will be possible.
- How do we extract showers with an algorithm?

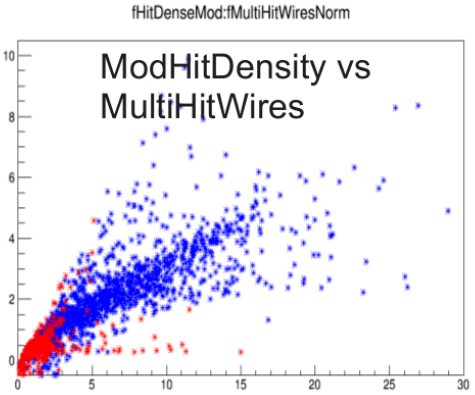
Step 0: Shower Identification

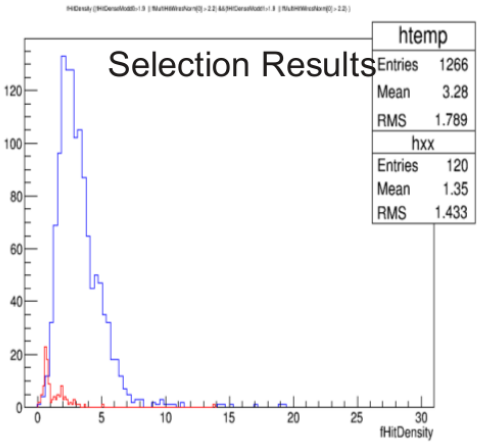
- How can a computer distinguish between showers and tracks in the data? Looking at clusters, there are a number of variables that can be used to separate them. Pick your favorite machine learning technique and go to it.
  - In past analyses hand scanning has also been used for shower reconstruction. We will **need** to move away from this eventually (SBND will have tens of thousands of events!!)
- Been an open topic for some time:
  - Shower Reco Retreat (Andrzej)
  - Shower Reco Overview (Andrzej)

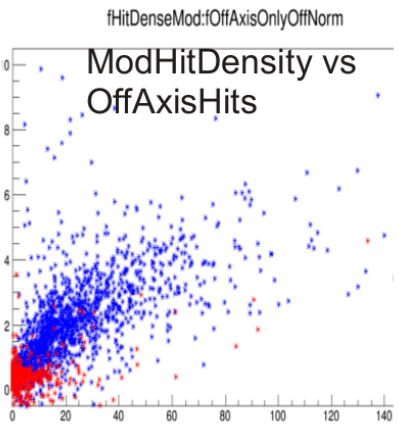
Slide from Andrzej, 3/2014

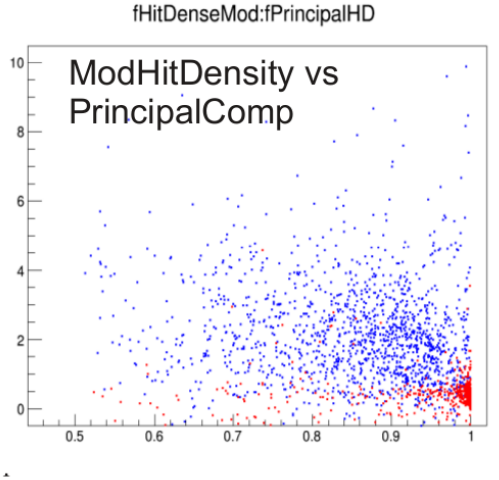
Combining the above methods together.

- They are probably strongly correlated, but it might make sense to try a TMVA analysis?
- They are now plugged in to a ShowerSelectorFilter.
- `ClusterParamsAlg::isShower(...)`
- Here applied cuts of ModHitDensity > 1.9 || MultihitWire > 2.1 (to be inclusive). Require 2 showerlike clusters in event.
- This gave ~88% efficiency for Showers that had at least 1 hit and 8% tracks kept (these are cuts and variables optimized for ArgoNeuT).
- The 8% tracks are sometimes legitimately kept. More on that in the backup slides.

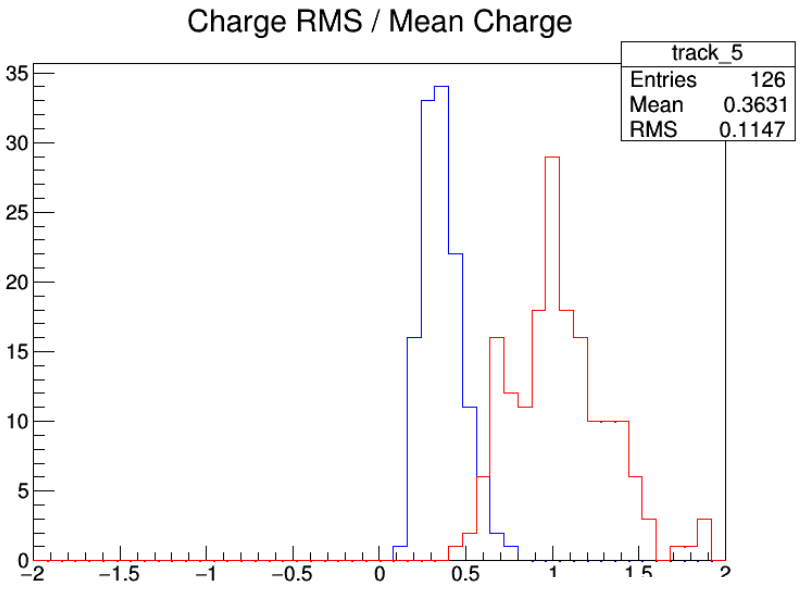


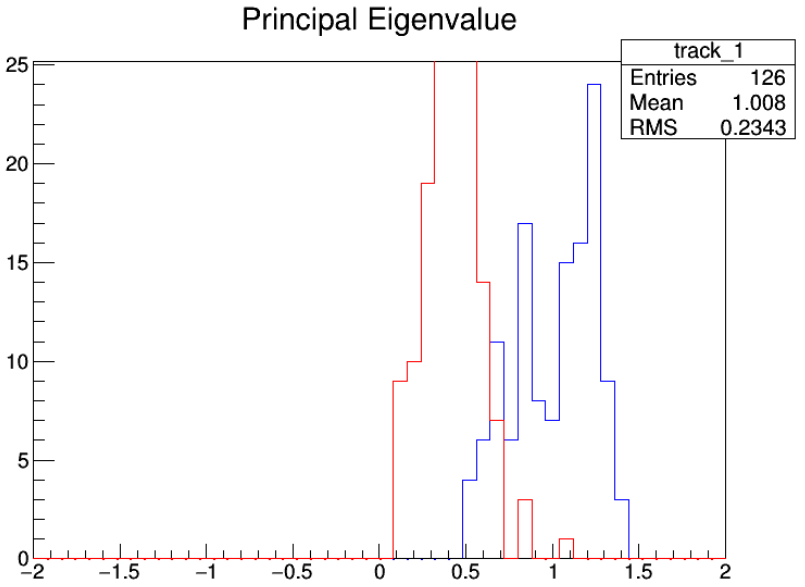


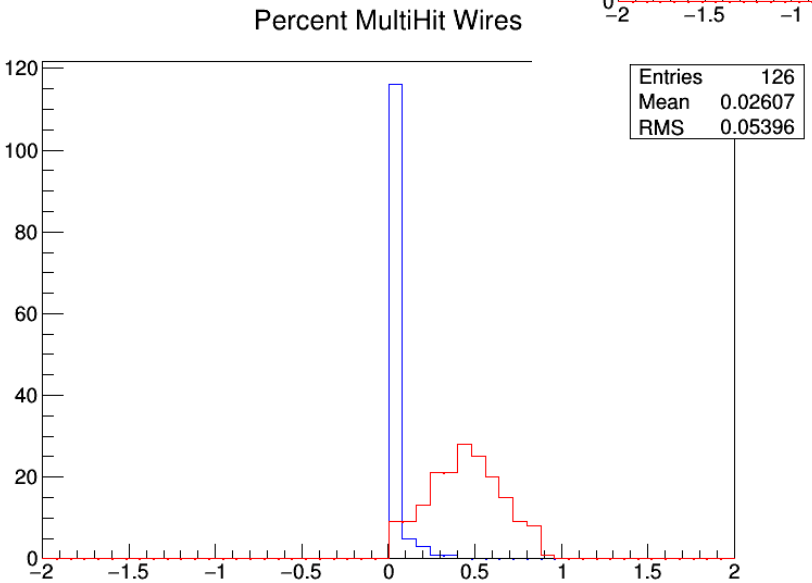




Development of ClusterParamsAlg was for shower identification







TRACKS

SHOWERS

+ Other discriminants not shown

Shower Reco Proper

- Current Strategy
  1) Pick the best two planes by the longest clusters along the wires.
  2) Use the opening angle in 2D of the “best” plane as the 3D opening angle
  3) Calculate the 3D axis using a complicated formula (next slide), only use 2 planes
  4) Calculate the start point (3D) using the 2D start points of 2 planes
  5) Calculate dE/dx for each plane
     1) This has hardcoded calorimetry constants, unclear and bizarre catches for bad behavior.

Step 1: 3D Axis
Reconstructing the 3D angles
(in 3-plane and 2-plane geometries)

- Original 2-plane formulas from Ornella and Maddalena (all of this (2D and 3D) is in GeometryUtilities)
- 0 index denotes the vertical plane (if used) or the plane with the negative angle wrt to vertical (Coll in ArgoNeut)

$$\phi = \arctan\left(n/l\right)$$

$$\theta = \arccos\frac{m}{\sqrt{\left(l^2+m^2+n^2\right)}}$$

2 Planes

$$l = sgn\left(t_{end} - t_{start}\right)$$

$$m = \frac{1}{2\sin\alpha}\left(\frac{1}{\Omega_0} - \frac{1}{\Omega_1}\right)$$

$$n = \frac{1}{2\cos\alpha}\left(\frac{1}{\Omega_0} + \frac{1}{\Omega_1}\right)$$

α_0 = 0 if one of the two planes is vertical otherwise it is the smaller of the two
Ω is the tangent of the 2D angle
l,m,n are basically the x,y,z Coords, respectively.

3 Planes

$$l = sgn\left(t_{end} - t_{start}\right)$$

$$m = \frac{1}{2\sin\alpha_1}\left(\frac{\cos\alpha_1}{\Omega_0\cos\alpha_0} - \frac{1}{\Omega_1} + !\left|sgn\left(\alpha_0\right)\right|\left(\frac{\cos\alpha_1}{\cos\alpha_0\,\Omega_0} - \frac{1}{\Omega_1}\right)\right)$$

$$n = \frac{1}{2\cos\alpha_0}\left(\frac{1}{\Omega_0} + \frac{1}{\Omega_1} + !\left|sgn\left(\alpha_0\right)\right|\left(\frac{1}{\Omega_0} - \frac{1}{\Omega_1}\right)\right)$$

If you don't have any idea how this works, welcome to the club!

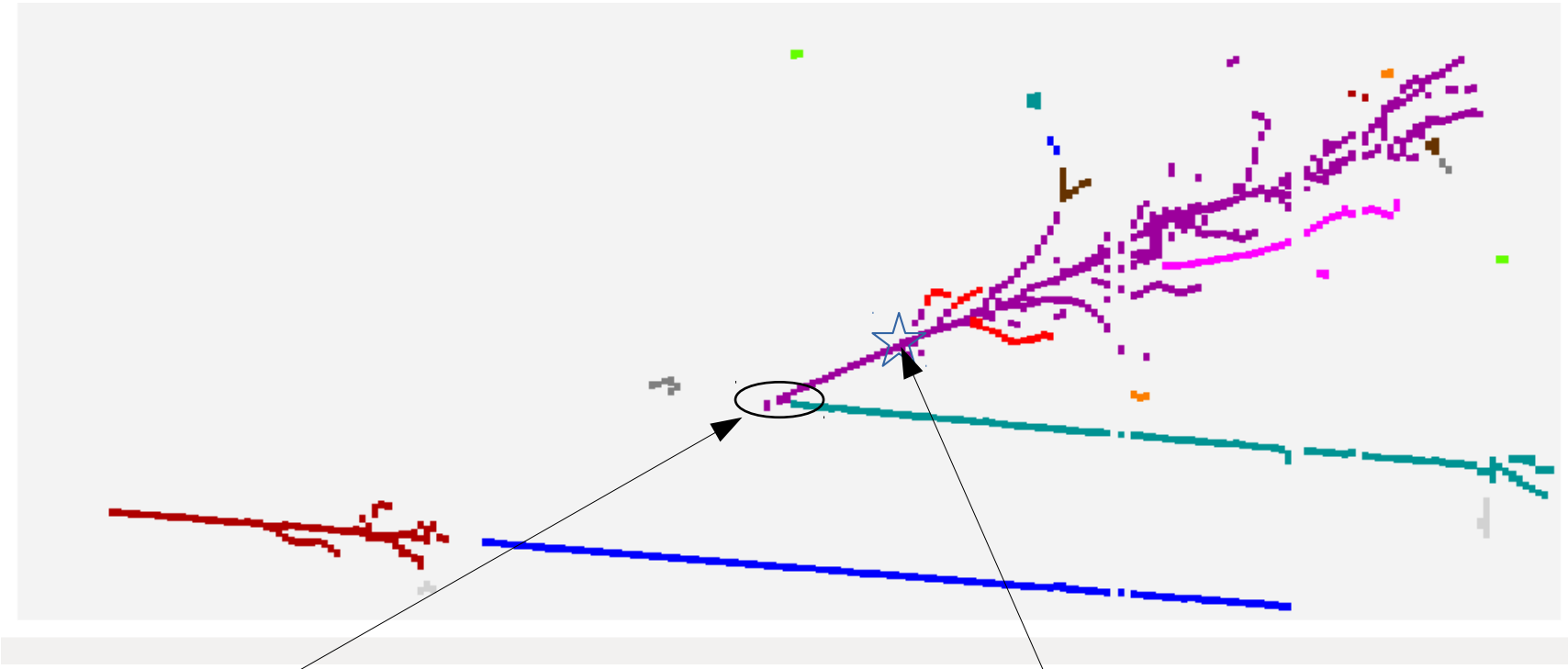
Proposed Strategy

1. Find 3D axis of shower.
   (a)The 3D axis of the whole shower and of the start of the shower might not be identical.
2. Refine 2D start points by looking at the charge per hit: goal is to remove any stray vertex hits from the shower that might mess up dQ/dx
3. Determine 3D start point.
4. Get dQ/dx (not yet dE/dx).
   (a)Should have dedicated, experiment (or even analysis) specific code for finding dE/dX from dQ/dx.
5. Get other parameters like length, total charge deposited, opening angle, etc.

1. 3D Axis

- Proposal: throw out calculation and use an iterative scheme and every available plane.
  - Start with an educated guess (might still use the complicated formula?) to pick a 3D axis.
  - Project the axis into each plane and calculate how much the axis is off for each shower
  - Feedback the errors in each plane to the 3D axis and adjust it in the correct directions.
  - Repeat, hopefully decreasing the adjustment each time, until the 3D axis converges to a best fit for the shower planes
    - Can favor planes by having different adjustment weights per plane, etc.
- The 3D axis might be slightly different to the axis needed for finding the dQ/dx later, but the same algorithm could be recylced for that as well (with a restricted hit list)
- Track and seed finding is quite good – we probably don't want to reinvent the wheel but rather take advantage of the work that is already done!

2. Refine start points



Need to make sure the hits at the start are all from the shower and not other particles at the vertex

The point at which showering starts is important and should be found for determining how much of the shower to use for dQ/dx calculation

3. 3D Start Point

- With a good 3D axis, the 3D Start point had better be roughly along the axis!
  - Can make a guess along the axis, project into planes, find how to adjust, and iterate until the start point is a best fit for all three planes.
- Important to make cross checks to verify that the start point makes sense in each plane, if possible.

4. dQ/dx

- Correct dQ/dx is essential to doing physics with electromagnetic showers!
- The “natural unit” for dx is the wire spacing because that is the unit of charge measurement, but we need to project the wire spacing into 3D
  - Find the points along the 3D axis that line up with each hit wanted for dQ/dx measurement and iteratively improve them until the step size (in 3D) is found.
- In the past, dQ/dx was measured over a set distance for all showers
  - Abandon this one size fits all approach! For each shower, determine where the “showering” behavior starts and measure dQ/dx up to that point
- Find dQ/dx for each plane, if possible. Conversion to dE/dx should be done independent of dQ/dx.

5. Other Parameters

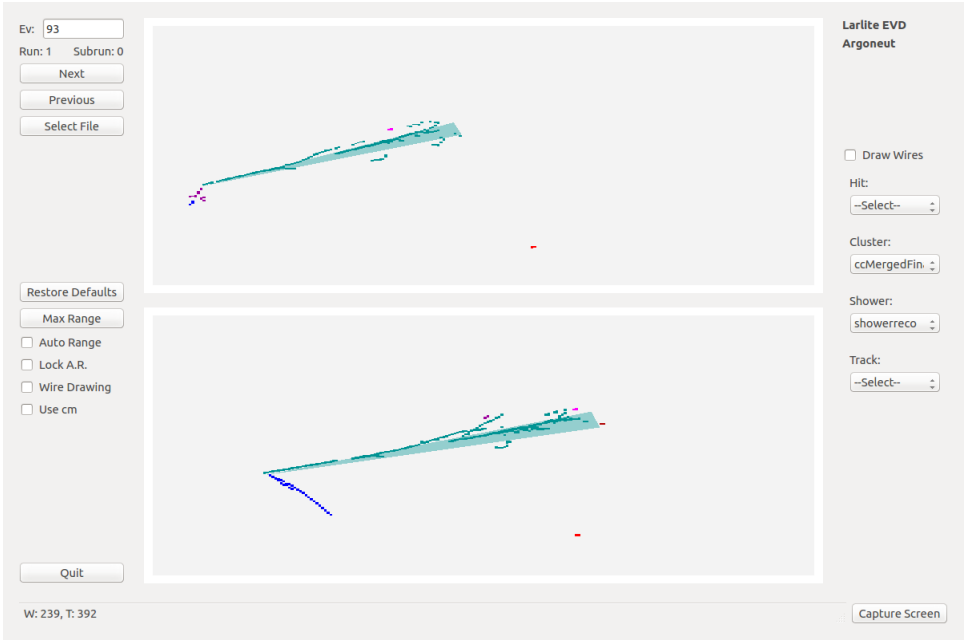
- Determine length, opening angle, etc.
  - Every parameter needs to be checked by projecting it back into the 2D planes and iterating until the parameter converges!
- Also, we should be able to quantify the uncertainty on many of the parameters:
  - Start point uncertainty could be the average difference of the 3D-Projected-to-2D point vs. cluster start point in each plane, etc.

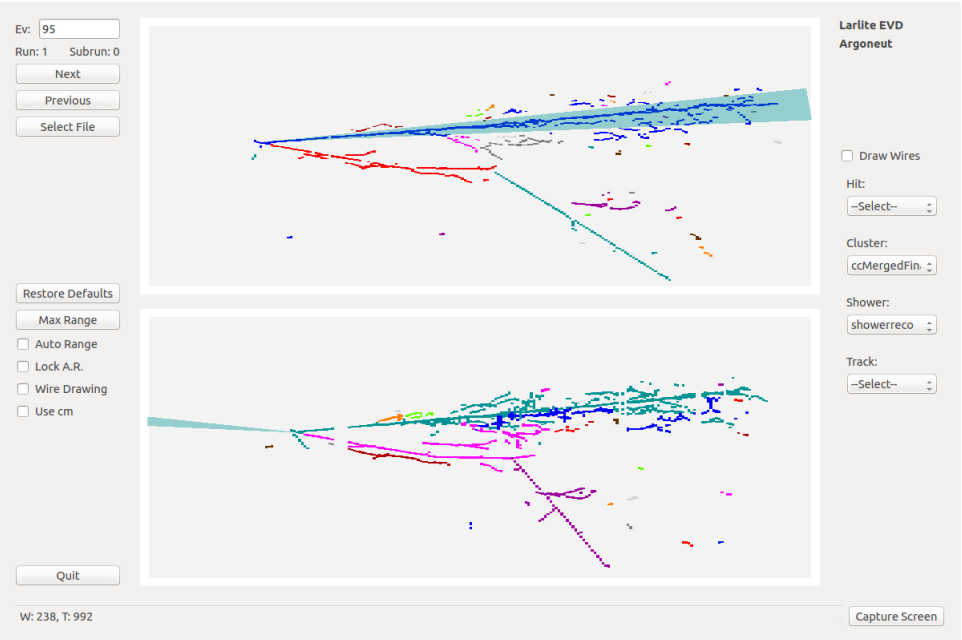
General Comments about Development

- The best (IMO) way to do this is with modular design, where each step is it's own function(s) inside of a shower reconstruction algorithm.
  - If you're familiar with clusterParamsAlg, I think we should follow a similar format.
- To make an experiment specific algorithm, for example, a user could inherit the general algorithm and override **only the functions** they need to change.
  - Gives flexibility and also easy access to the core functions in a custom algorithm.
    - For example, 3D axis finding might be the same everywhere but dE/dx calculation might not be!
    - Start point refining will probably be different depending on the clustering algorithm
    - Etc...

Specific comments about development

- Work in larlite, on the branch “shower_dev”
  - Already pushed to github, check out and dive in!
- The larlite evd will draw clusters and the projection of showers from 3D into 2D, so you can see when things are working and when they are not...





Geometry Utilities

- The current shower reco uses the GeometryUtilities class.
- That class is difficult to maintain for development, and finding the function a user needs is very difficult.
- Shower reconstruction should develop, in it's own class, a utilities module that is clear, well documented, and easy to extend.
  - Branch out entirely, rather than try to change GeometryUtilities, to prevent breaking other areas of code.
- Need to be able to convert a point/line/angle/segment from 3D to 2D, or 2D to 3D, etc.
  - For consistency and to prevent errors, we should enforce units of choice (cm and radians?)

Summary

- I've shown the way shower reconstruction currently works and proposed one possible way to redo it.
- This is just one person's opinion! This is a big project with a lot of very interested people, so let's be sure we're happy with the way it's going.