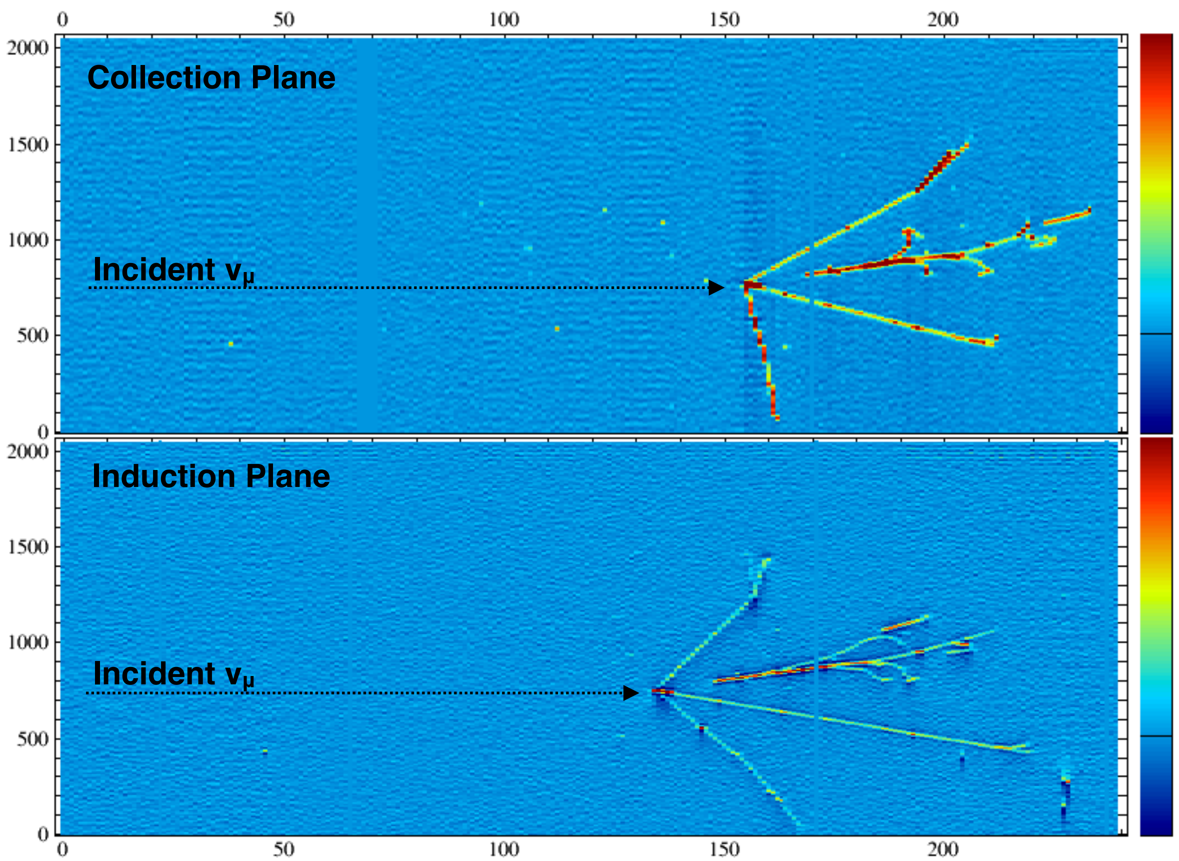
Pattern Recognition in Liquid Argon TPCs for Neutrino Experiments

Tracy Usher
SLAC
Speaking on behalf of a lot of hardworking people
on MicroBooNE and ELBNF

Connecting the Dots 2015
February 10, 2015

What's New is Old?



An image from the ArgoNeuT Experiment
(see http://t962.fnal.gov/Images.html)

Liquid Argon Time Projection Chambers are this millennium's bubble chambers…

This time the output is a digital image!

The challenge is to automatically reconstruct the events - no human scanners!

Some important differences to your typical collider experiment:

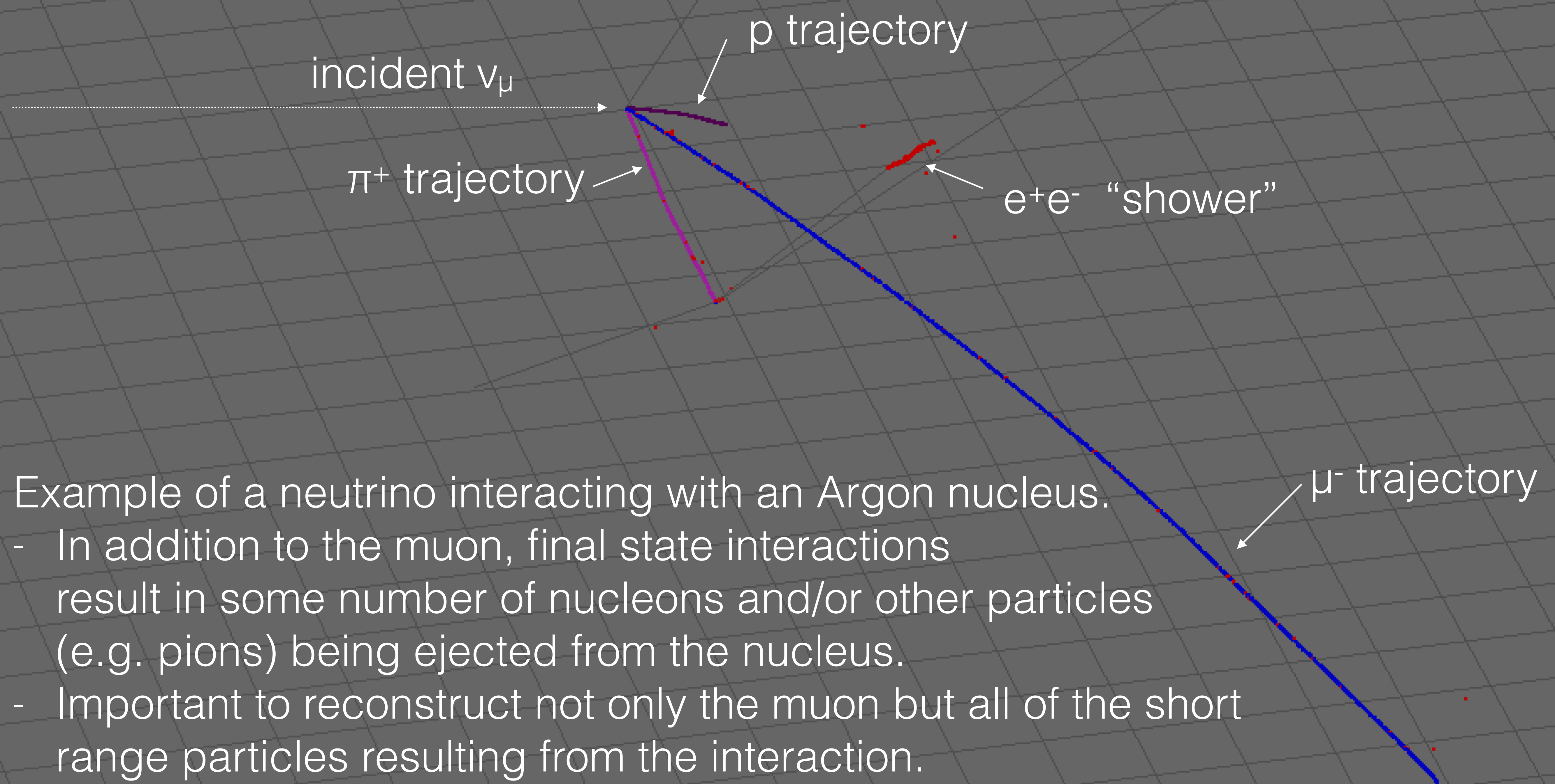
1. The rates are low — ~8 Hz — take all beam gated events
2. The number of tracks to find are much lower - typically ~15-20 per event
3. NOT sparsely sampled detector, can have 1000's hits per track
4. There is no well defined IP, the entire detector is the target
5. Typically no magnetic field - all the tracks are "straight"
6. It is still a surprisingly difficult problem, though extremely interesting

And still fertile ground for new ideas!

Liquid Argon Time Projection Chamber

The Promise: 3D imaging of the event

3D View of simulated ν_μ + Ar -> μ + p + π^+



Example of a neutrino interacting with an Argon nucleus.

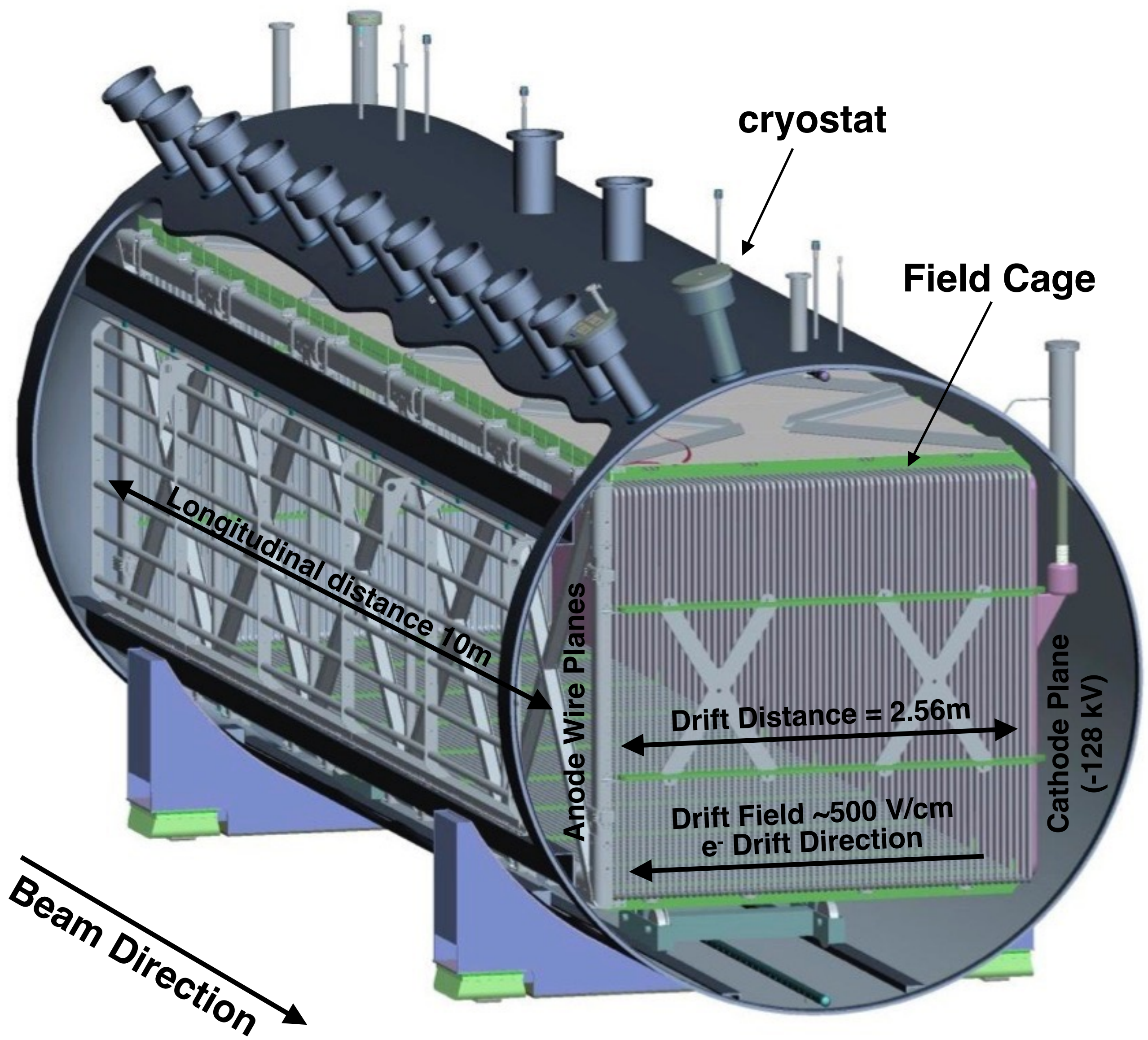
- In addition to the muon, final state interactions result in some number of nucleons and/or other particles (e.g. pions) being ejected from the nucleus.
- Important to reconstruct not only the muon but all of the short range particles resulting from the interaction.

3D imaging should allow you to "see" all the pieces of the interaction

Liquid Argon Time Projection Chamber

Current Detectors Follow Similar Design Concepts

The MicroBooNE Detector



Total volume ~170 tons
Active volume ~80 tons

Induction planes at ±60° to vertical
Induction planes each have 2399 wires, Collection plane has 3456 wires

Basic idea:

1. Fill a cryostat with liquid argon
2. Insert a field cage to create a drift field
3. Collect charge on anode wires
    1. Wire gives position along beam direction
    2. Drift time gives horizontal position transverse to the beam
    3. Induction planes in front of collection with wires at angles to give stereo information - gives vertical position transverse to the beam

Drift of e^- over ~3 m has been demonstrated

- sets the scale for the transverse dimensions of the detector
- Larger volume detectors simply insert more field cages, increase length, etc.
    - ICARUS 600 ton inserts two TPC's go get ~480 ton active volume
    - ELBNF uses multiple TPC's (and wrapped anode planes) to achieve kTon scale

I will use MicroBooNE as the example but I claim (without proving) that the reconstruction is/can be essentially identical for all currently planned LAr TPC's

The General Plan for What Follows

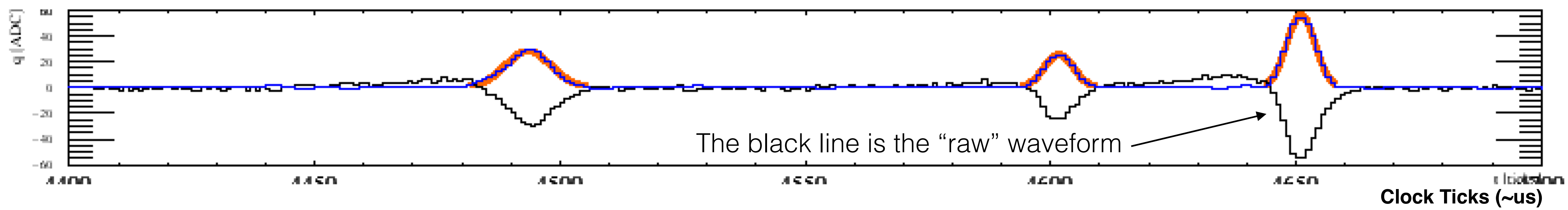
- My assumption is that the LAr TPC Tracking problem is new to the bulk of the people sitting in this room
- My goal is to give a very broad overview of the basic problems
    - Start by highlighting a few “extra” challenges
    - Then briefly contrast two approaches to Track finding to illustrate some issues
- There is much I won’t mention
    - Shower reconstruction (a whole topic to itself!)
    - Important effects but which should be second order to the above
        - Electron lifetime
        - Space Charge effects
        - Electric field distortions
        - Higher order field response effects
    - Event reconstruction
    - etc.

Basic Hit Finding

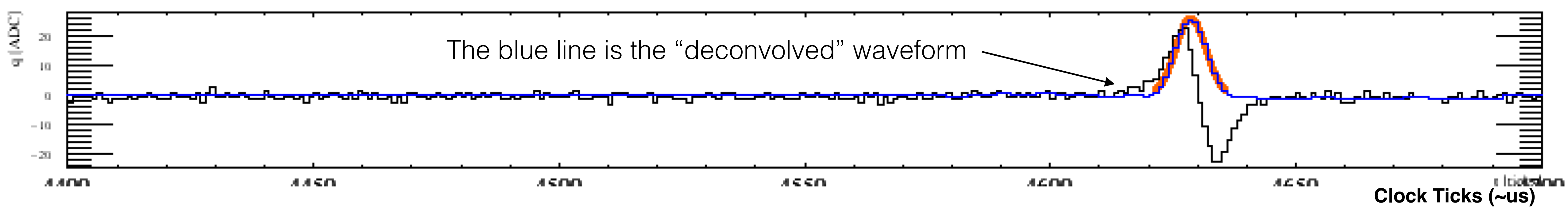
By way of explaining challenge #1

The input “raw” waveforms are processed to “wire data” and then passed to a hit finder which expects to see gaussian shaped pulses for each of the hits

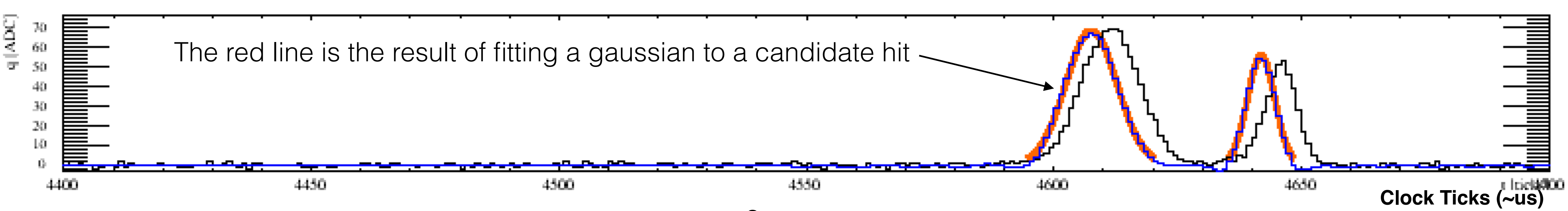
The U plane (1st Induction Plane) does not have a fully bipolar shape (no guard plane)



The V plane (2nd induction plane) is a bipolar pulse



The W plane (collection plane) is a unipolar pulse

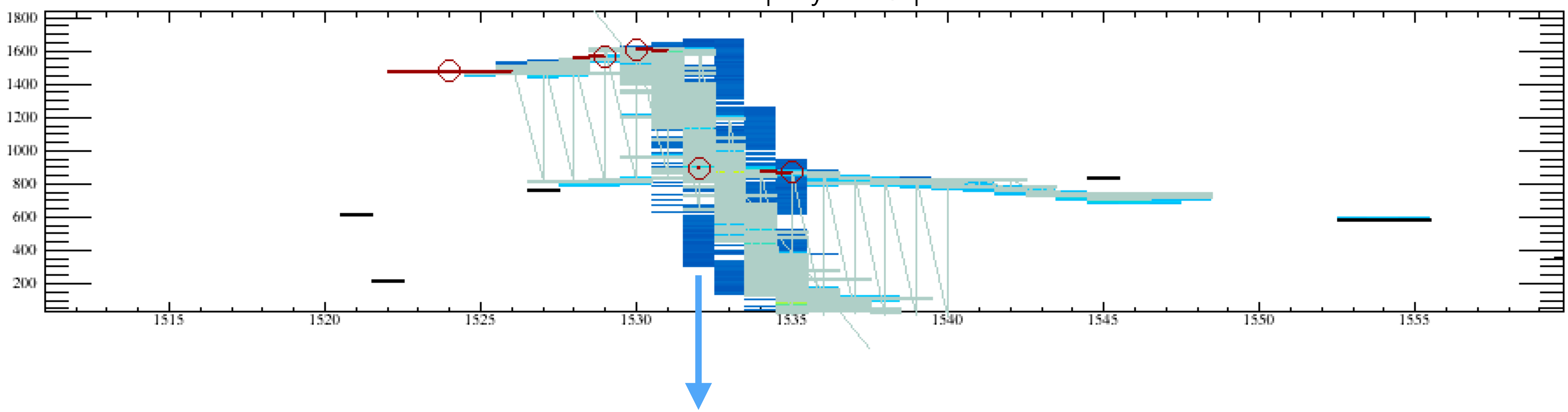


Example Hit Finding Issue

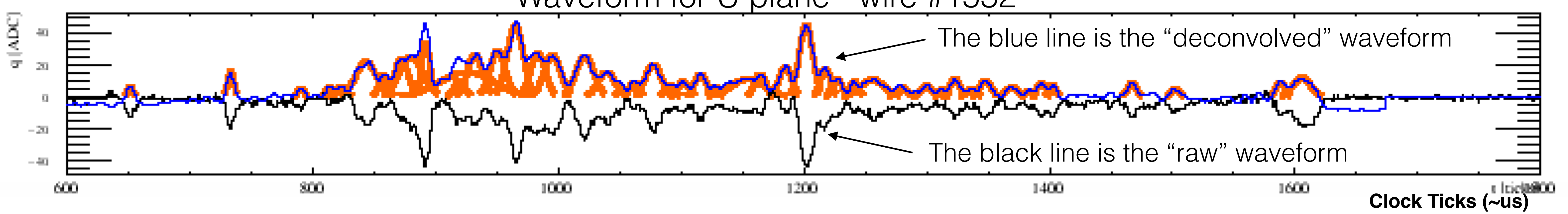
Challenge #1: “magic” track angles that impact the hit finding

The #1 issue facing hit finding occurs when tracks travel ~parallel to the drift direction
#2 occurs when tracks run ~parallel to a wire and with a large angle to the plane of the wires

2D Waveform display for U plane



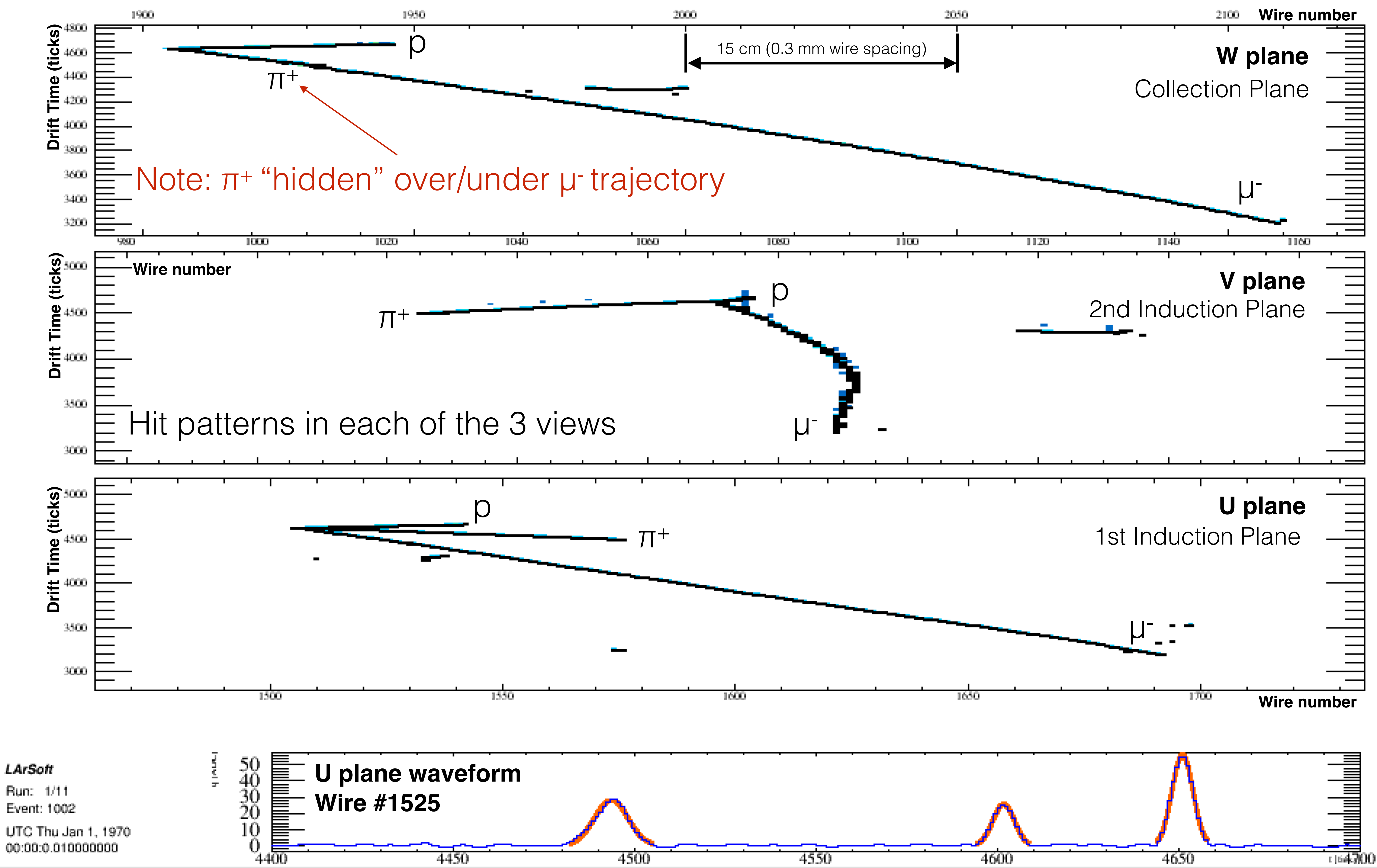
Waveform for U plane - wire #1532



As can be seen, the waveform does not necessarily represent the trajectory of the particle, the resulting waveform can have many peaks, it can also have many valleys which can create “holes” in the trajectory.

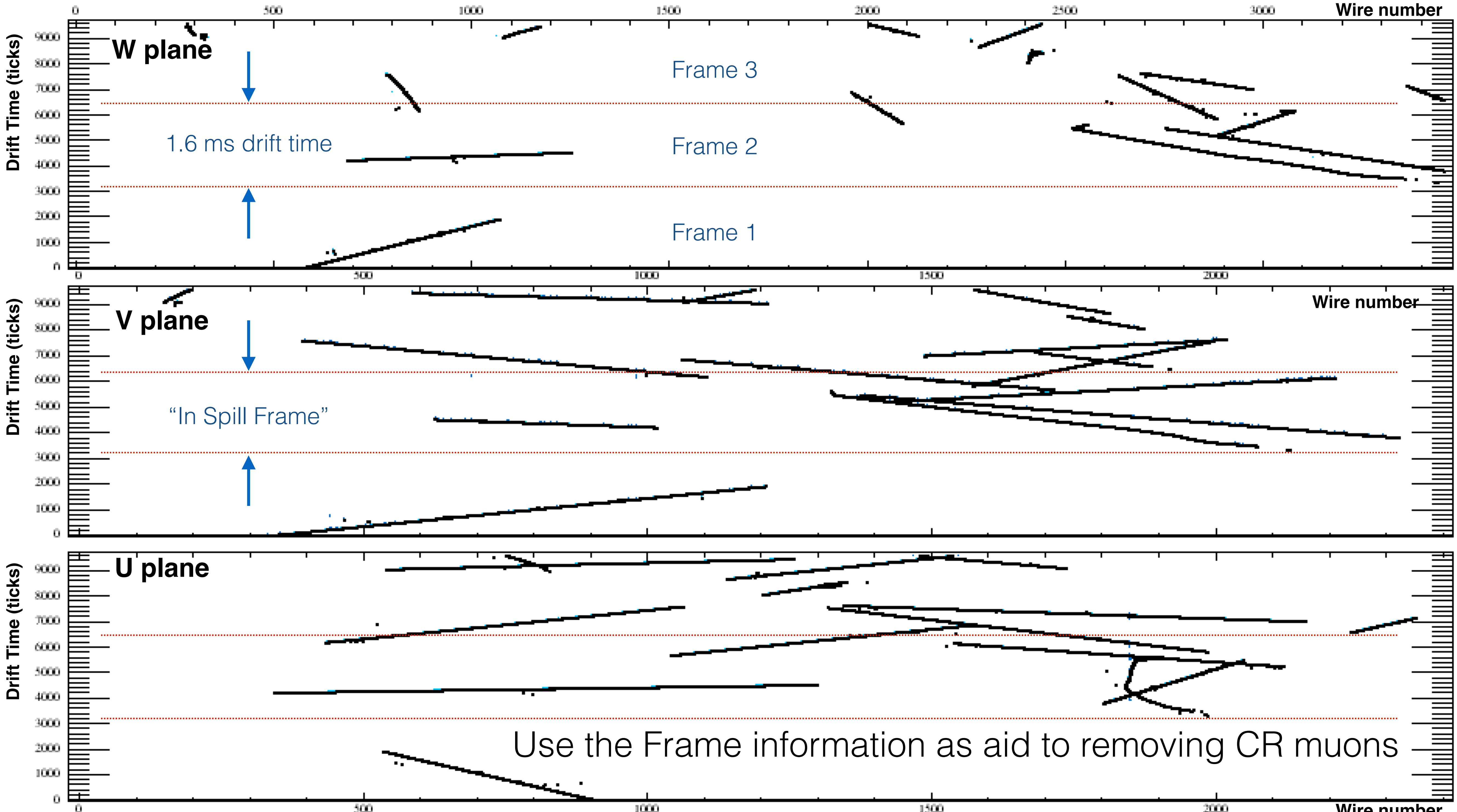
Event In 2D Post Hit Finding

(Revisit nice neutrino event from slide 4)



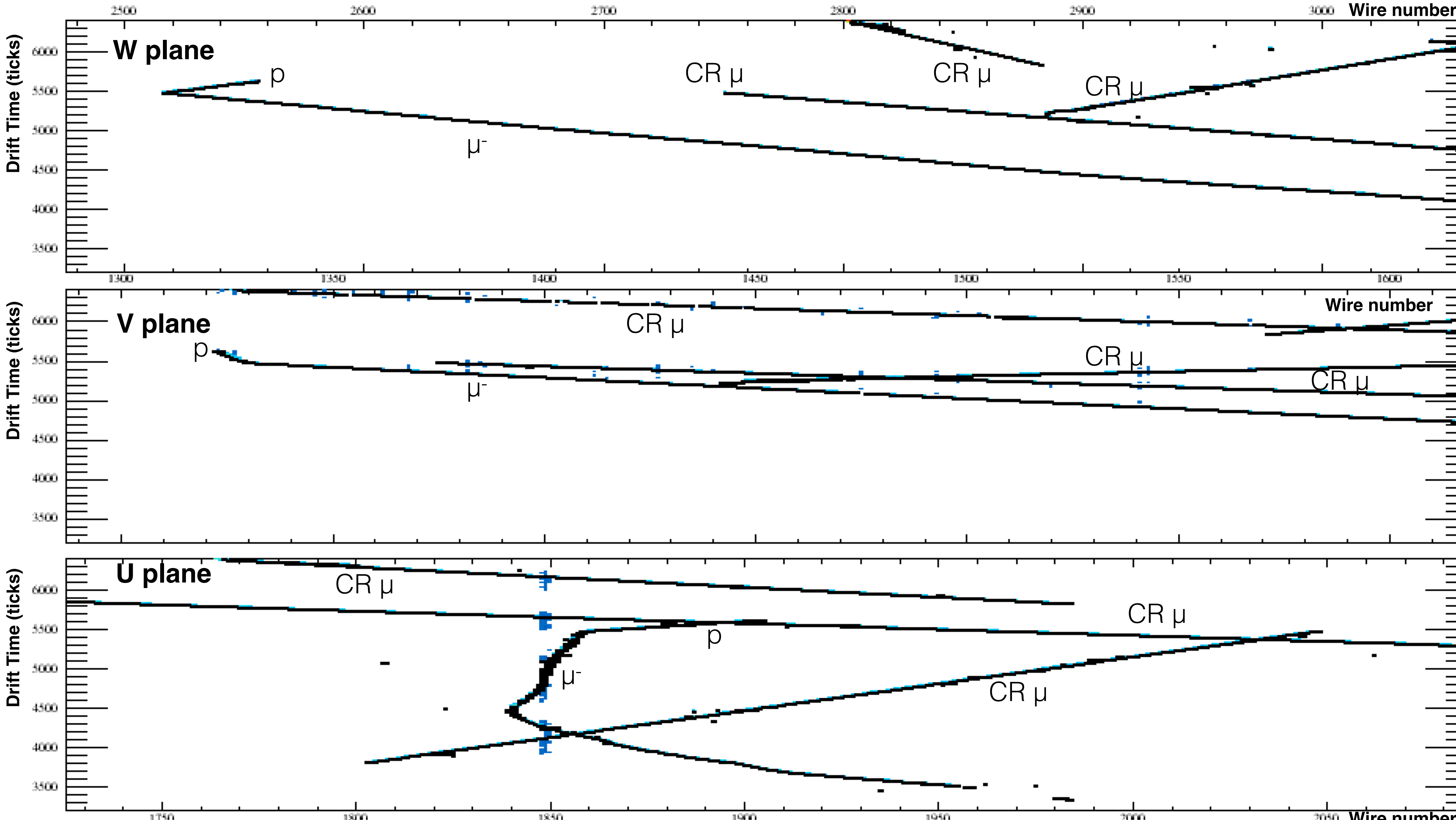
Challenge #2: Cosmic Rays

MicroBooNE (and ICARUS after move to Fermilab) are operating at the surface and will endure significant Cosmic Ray Backgrounds



Challenge #2: Cosmic Rays

Zooming in on previous event
NOTE: Scale changes in both drift time and wire



Pattern Recognition Approaches

- Primary goals
    - Return lists of 2D hits representing tracks which can be fit
        - Note that we will let the track fitter determine the order of the hits
    - Provide enough information to eliminate the “easy” CR muons
- Two approaches to contrast
    - Do initial pattern recognition in each of the 2D views, then do feature matching between views to create 3D picture of event
    - From 2D hits, create all “consistent” 3D space points and perform pattern recognition - rely on pattern recognition to disambiguate where necessary
- Common themes between the approaches
    - Group hits into logically associated sets
    - Search for straight line segments
        - Can link straight line segments to account for curvature due to MCS

Track Finding Method I

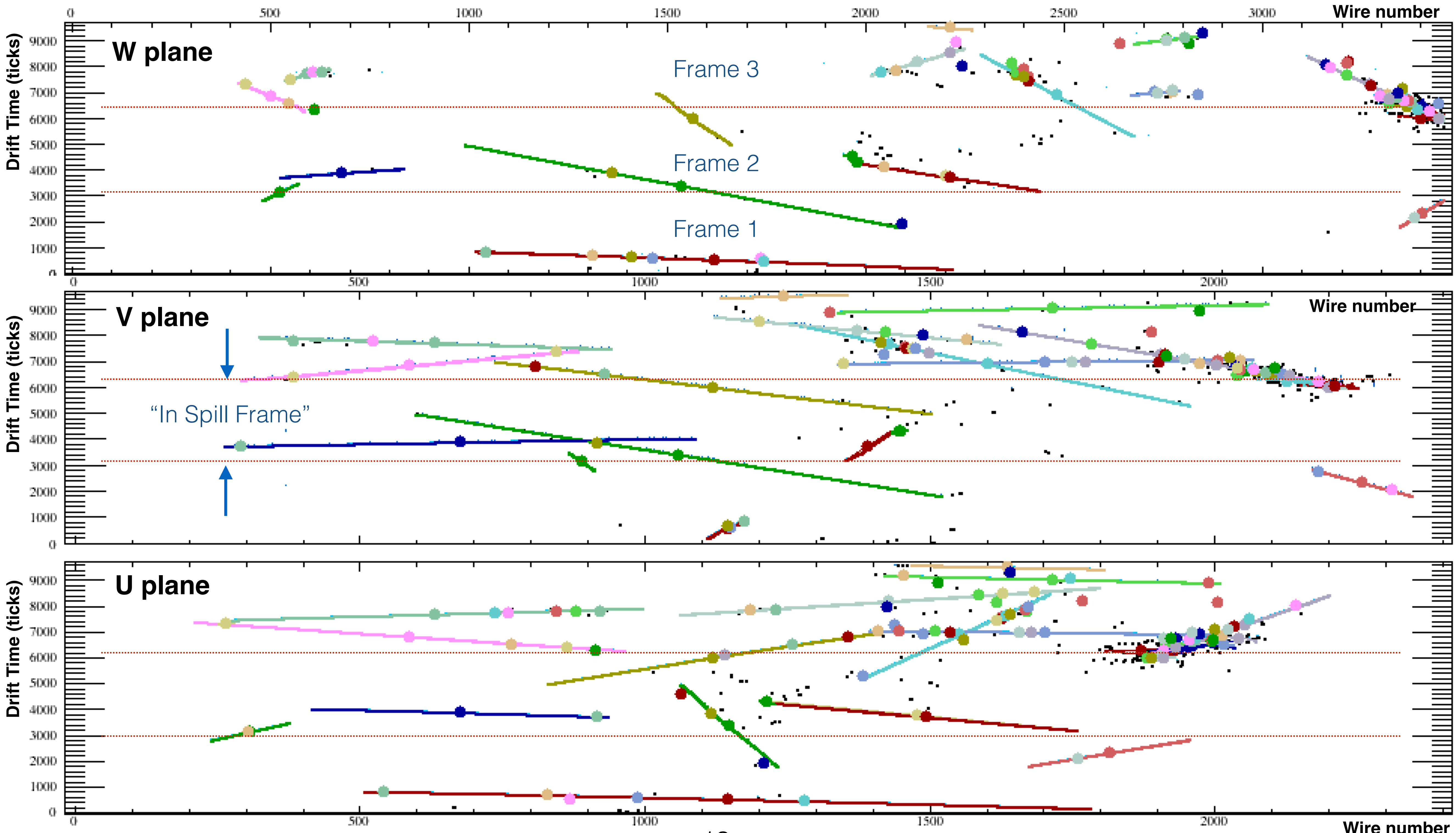
2D to 3D Approach - rather simplified to fit on one page

- Treat each 2D view independently
  - Group hits into associated sets
  - Find straight line segments to start candidate tracks
    - Use a “sliding line fit” to add hits - takes into account the slow curvature resulting from multiple coulomb scattering
  - Accommodate “kinks” as either track endpoints or track crossings
  - End result is a series of track candidates with endpoints (and possible vertices)
- Match the features in 2D to make the 3D result
  - Most powerful tool: match timing of endpoints
  - Must be prepared to split or merge to handle special cases
    - For example, tracks “hidden” in a view
      - Can try to use deposited charge information in these situations
      - Must allow hit sharing in these situations as well
- End result
  - Matched lists of 2D hits in each view which can be sent to the fitter
  - Can try to make 3D space points (but this can be problematic)

2D to 3D Example

Clearly able to resolve tracks in all 3 views!

2D Hits associated to the same 3D “particle” appear in the same color



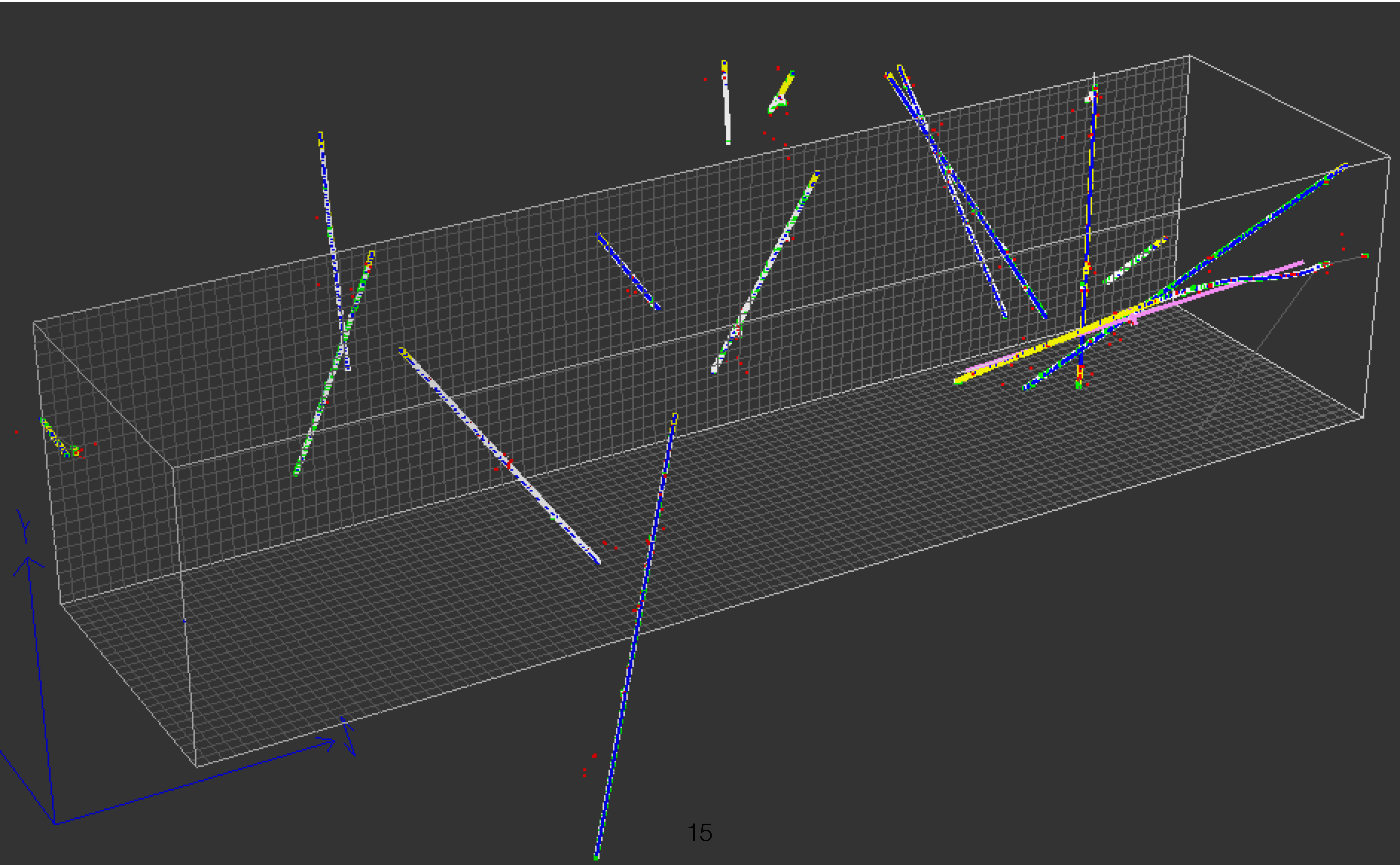
Track Finding Method II

3D Approach

- In general the idea is to create 3D space points by matching 2D hits from each of the views and then input these to a 3D pattern recognition algorithm…
- Wait!
  - You just said the 2D to 3D approach works fine
  - Why do you want to do something different?

Simplicity: Everything Separates in 3D

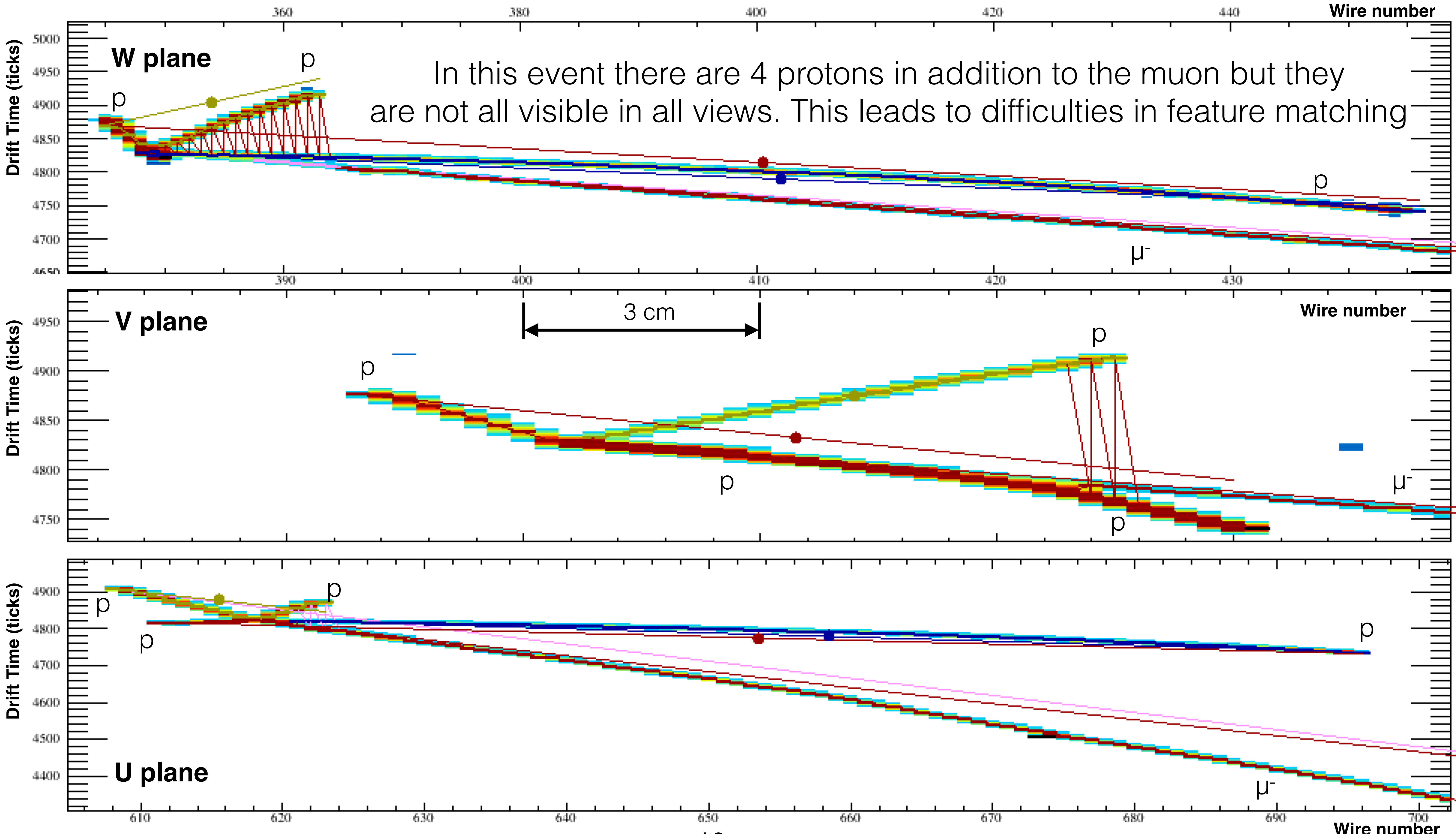
The event shown in 2D on slide 10



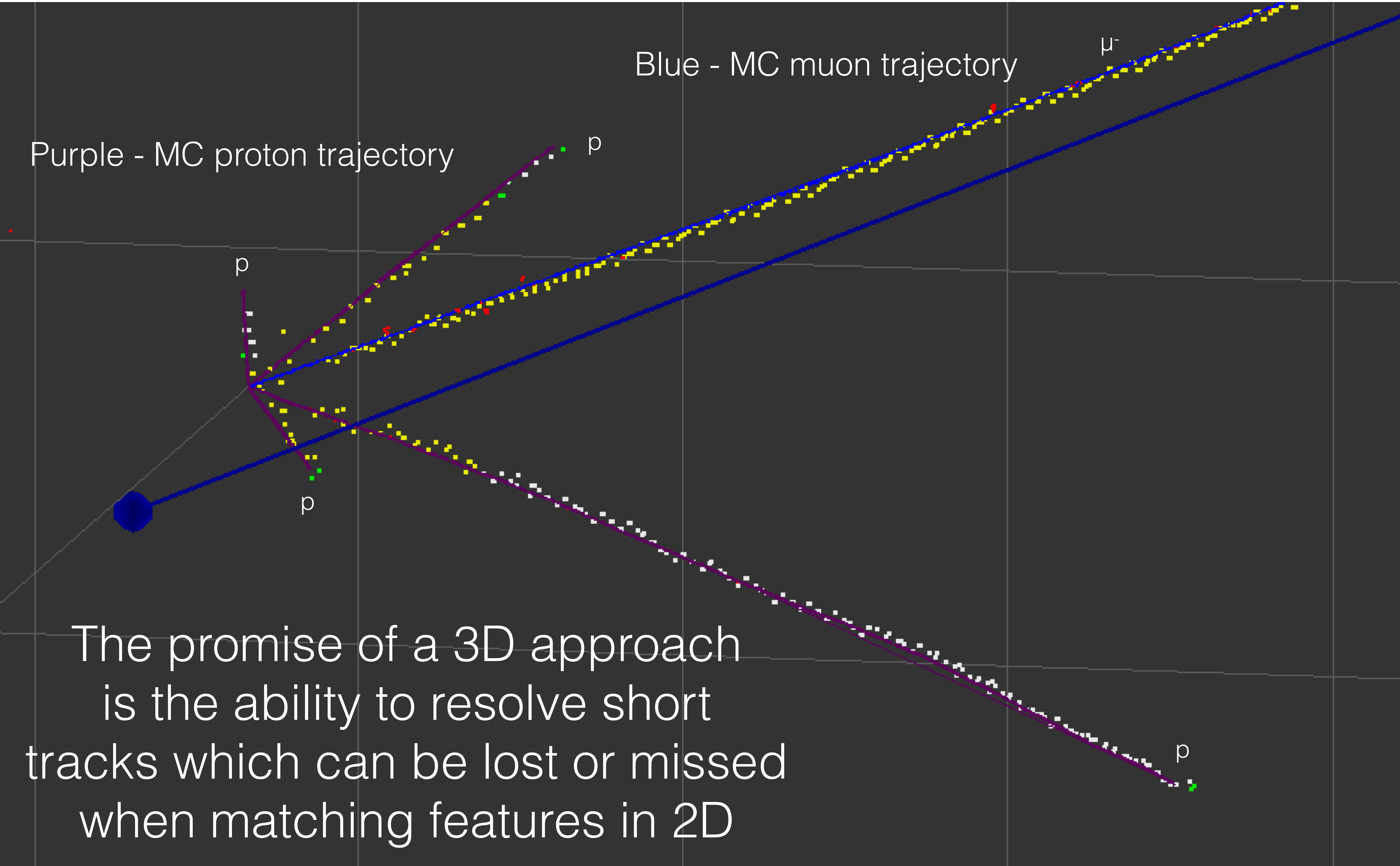
Resolve 2D Feature Matching Problems?

2D to 3D Approach

Feature matching across views can struggle for certain topologies



Prongs Become Visible in 3D



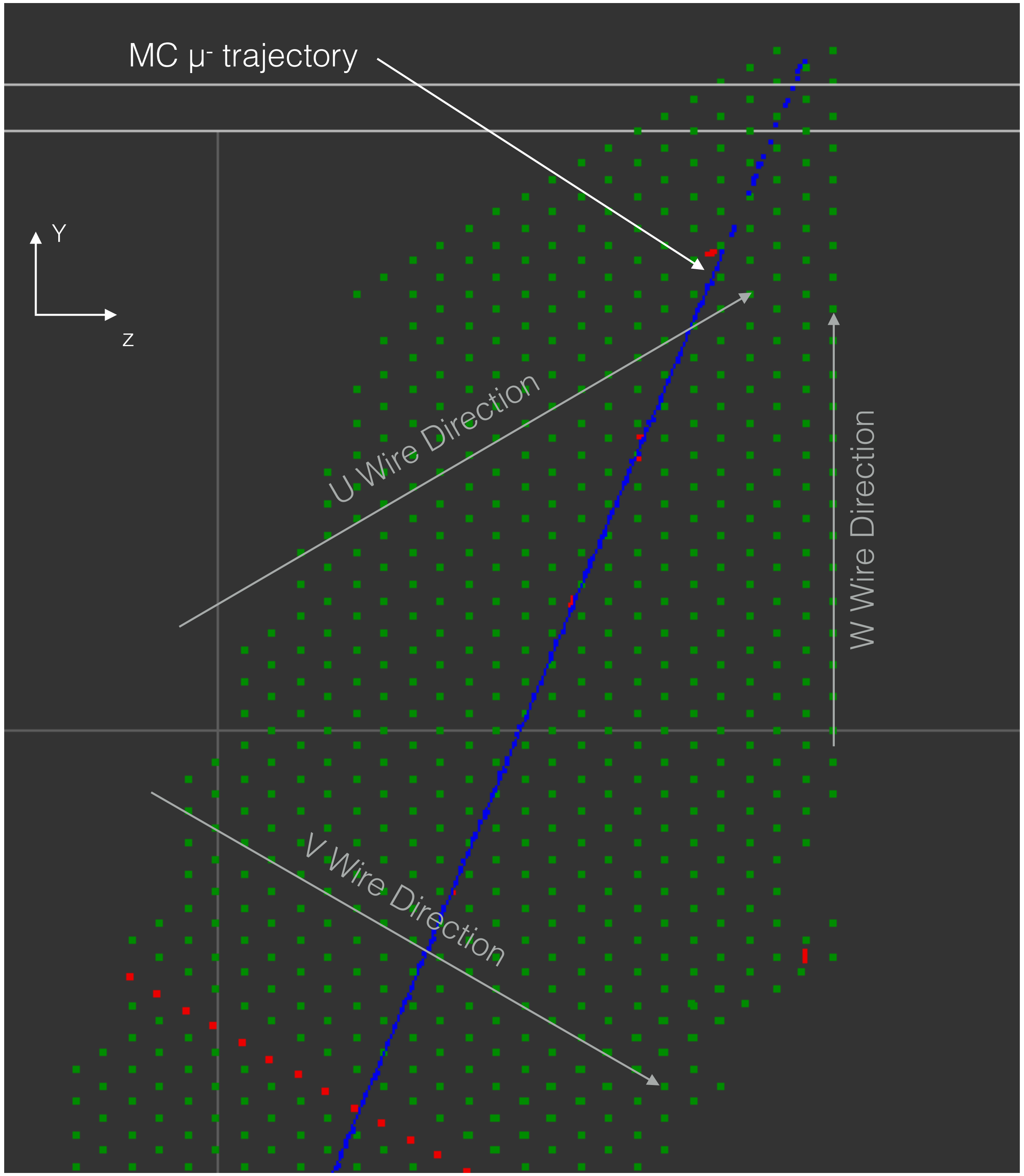
The promise of a 3D approach is the ability to resolve short tracks which can be lost or missed when matching features in 2D

2D Hits to 3D Space Points

- Combine hits in each of the 3 views to form Space Points
  - Obviously, we don't a priori know which 2D hits belong together
  - Criteria is that:
    - The wires for candidate hits to be combined "intersect"
    - The drift times for the hits are "consistent" within the resolution of the hit finding - this is the real constraint
- Creating Space Points is a balance
  - Too tight constraint above results in low efficiency
    - Broken and missing tracks
  - Too loose constraints above results in combinatoric issues
    - Tracks become artificially linked
    - Memory and CPU time issues
      - 20k 2D hits can mushroom to > 1M 3D hits easily
- The balance:
  - Space points made from 2D hit combinations in each of 3 views - hit "triplets"
  - Space points made from 2D hit combinations in 2 views only in special cases - hit "pairs"

- Will always have ambiguous hits
- 3D Pattern Recognition must be prepared to deal with this

Challenge #3: Ambiguous Space Points



Each dot represents a 3D Space Point

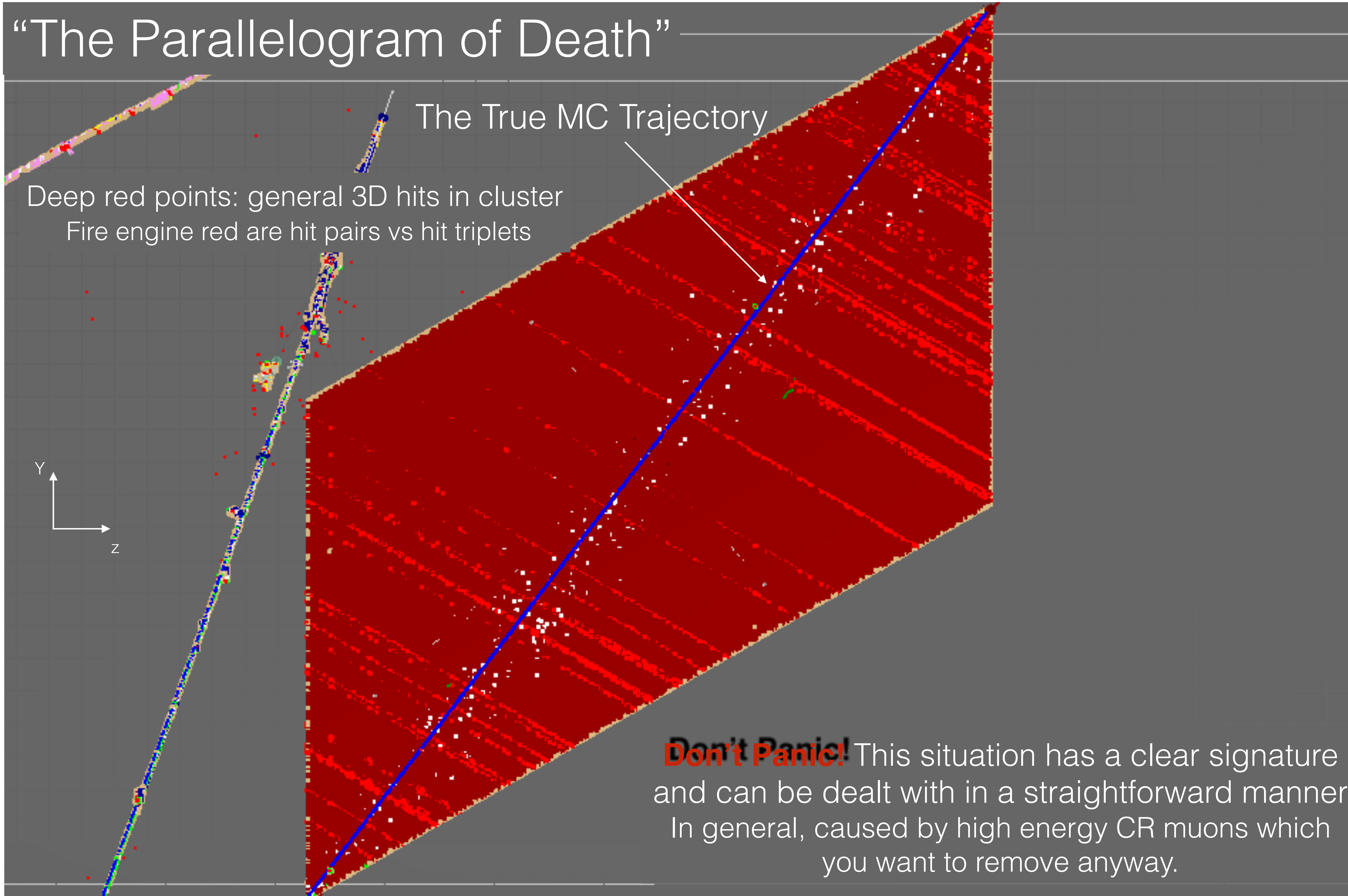
Green dots: 3D Space Points made from 1 2D hit from each of the three views - a hit “triplet”

Red dots: 3D Space Points made from a 2D hit from each of 2 views where there is a “valid reason” for the hit to be missing in the 3rd view

In general, getting the “correct” 2D to 3D hit association is a difficult problem - for certain track trajectories, even the the kalman filter struggles…

3D Hit Ambiguity - the worst case

Track running parallel to the plane containing the sense wires

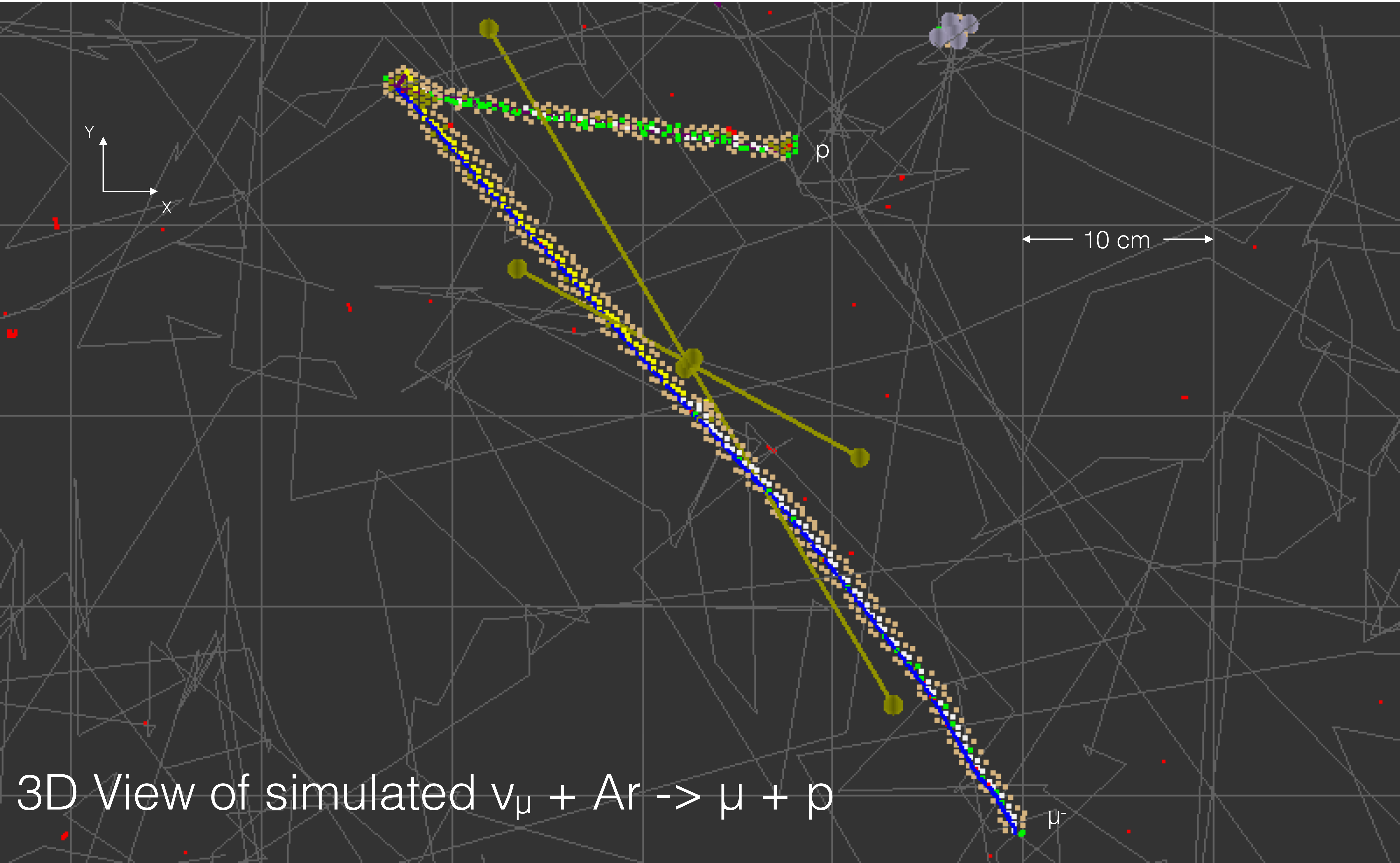


3D Track Finding Approach

The one slide summary

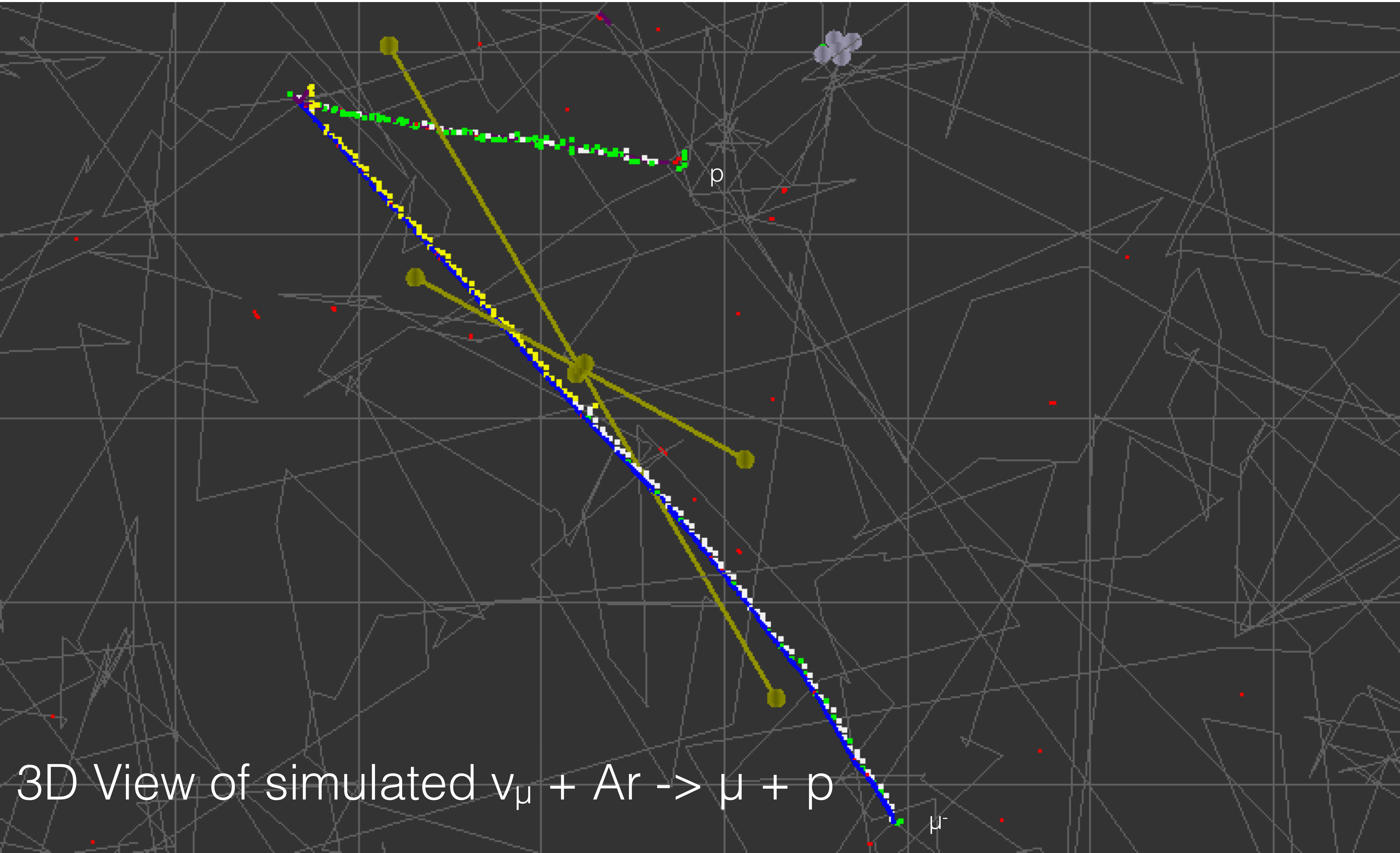
- Adapt some very basic principles from 3D imaging:
  - Treat your 3D Space Points as a point cloud
  - Use a density based clustering algorithm to associate hits
  - “Skeletonize” the resulting clusters to find the central trajectory
  - Use a Principal Components Analysis to find the axes of spread
  - Project 3D Space Points to plane of maximum spread
  - Use a Hough Transform to associate 3D hits which fall on straight line segments
  - Match/Merge segments to get candidate tracks
- Note that for long, straight clusters (e.g. CR muons) the Principal Components Analysis is all you need
  - The bulk of the hits lay on the same line segment
  - The primary axis gives good starting direction for track fit

Example with all 3D Hits

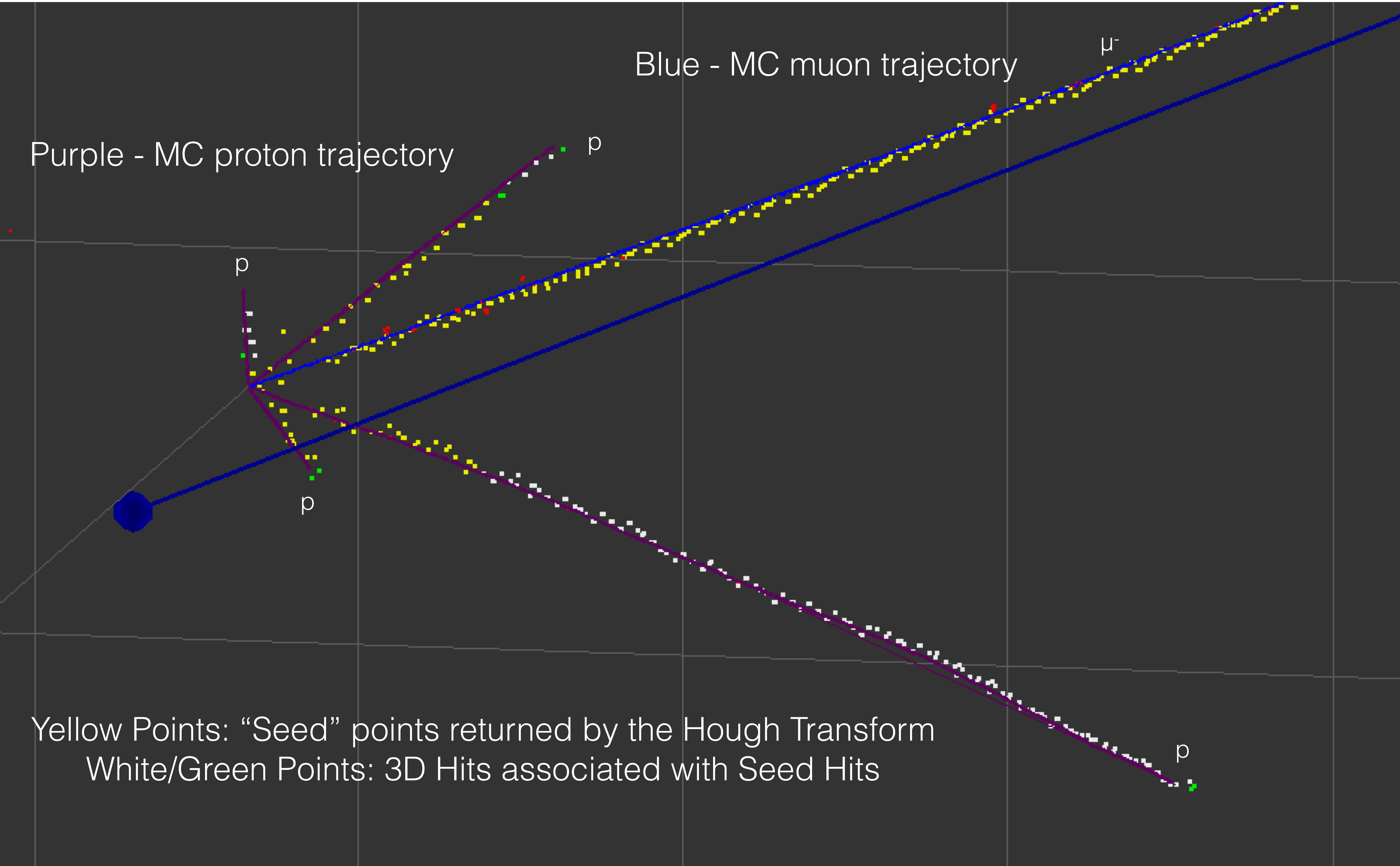


3D View of simulated ν_μ + Ar -> μ + p

… And with Skeleton Hits Only



Revisit the Motivating Event



Comparing the Two Approaches

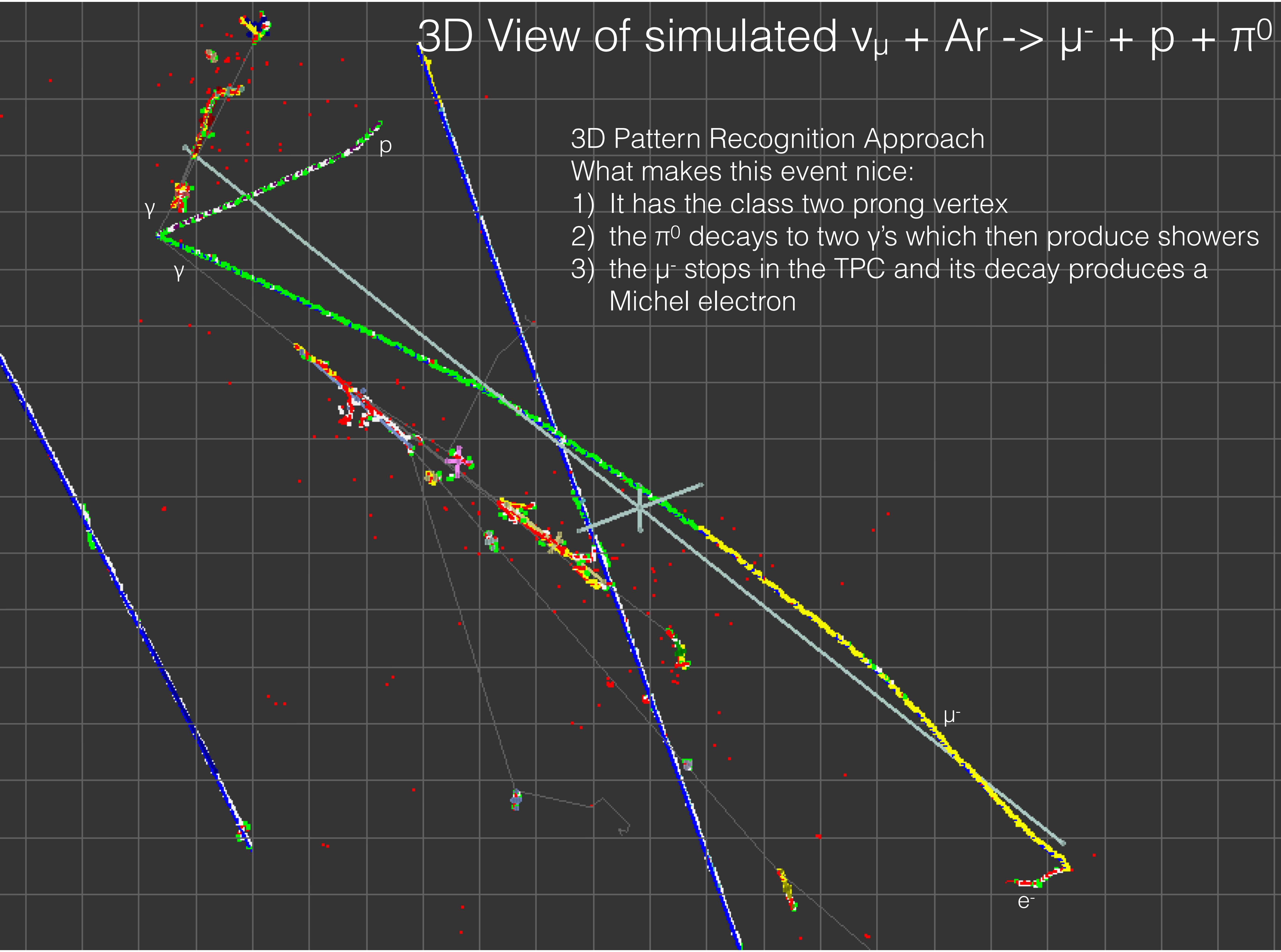
- 2D to 3D approach
  - Is less sensitive to missing hits than the pure 3D approach
    - In principle, can even miss an entire view and still get a 3D trajectory
  - Hard parts here are in resolving certain classes of crossing tracks in a given 2D view and then matching 2D views to form 3D objects
  - Has had more focus - is further developed at this stage
- Pure 3D approach
  - Has ability to resolve situations that easily confuse 2D to 3D, typically near the neutrino interaction vertex in multi-prong events
  - Primary issue here is dealing with the significant number of ambiguous 3D space points
  - Is very sensitive to missing hits and currently does not work if a view is missing
  - Surprisingly, is significantly faster than 2D to 3D approach
- Both Approaches
  - have good efficiency for finding tracks with lengths more than ~6-8 cm (>60 D hits)
- Best of all worlds
  - Can afford to run both and then let a later stage choose the "better" solution

3D View of simulated ν_μ + Ar -> μ^- + p + π^0

3D Pattern Recognition Approach

What makes this event nice:

1) It has the class two prong vertex
2) the π^0 decays to two γ's which then produce showers
3) the μ^- stops in the TPC and its decay produces a Michel electron



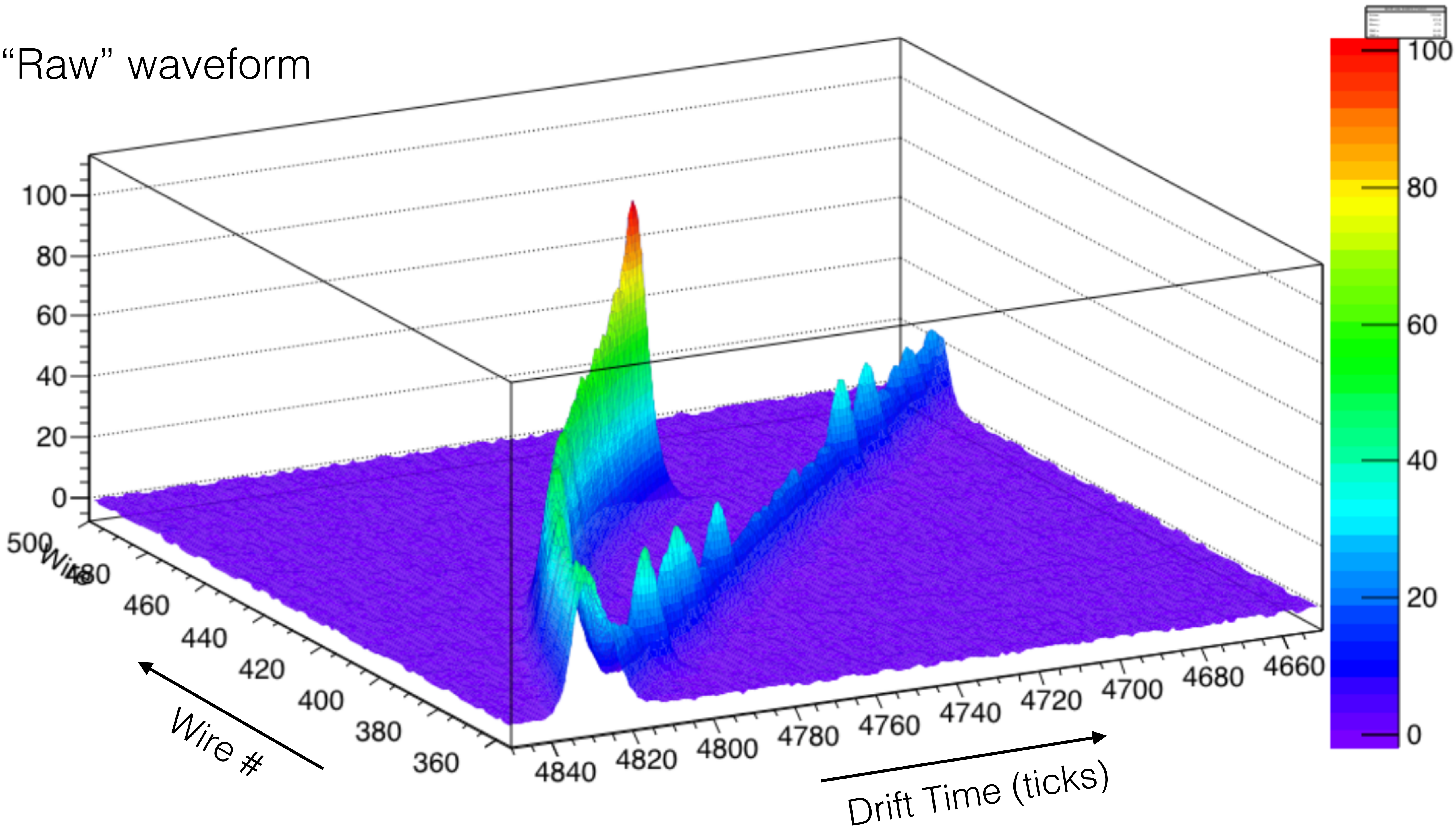
Where Do We Stand

- Definitely making great progress with the goal of “automatic” reconstruction of events from Liquid Argon TPC’s
    - Can reliably find tracks in the MicroBooNE TPC
        - Efficiency quickly approaches 1 for lengths > ~8 cm
        - Good enough to get started for tracks > 3-4 cm
        - Work still to do on the shortest segments
    - Have the ability to easily remove the bulk of the CR background
        - Tracks are easily found, those with hits “out of time” are easily removed
- In good shape going forward with two reliable and complementary approaches
    - “Ready” for first data from MicroBooNE TPC in just a few months!
- But…. is this the best way to do this?

An Even Better Approach?

We are not yet fully exploiting the TPC as an imaging detector

"Raw" waveform



Note that looking at waveforms in this manner can resolve the issue raised in slide 7 with "magic" track angles in the TPC…