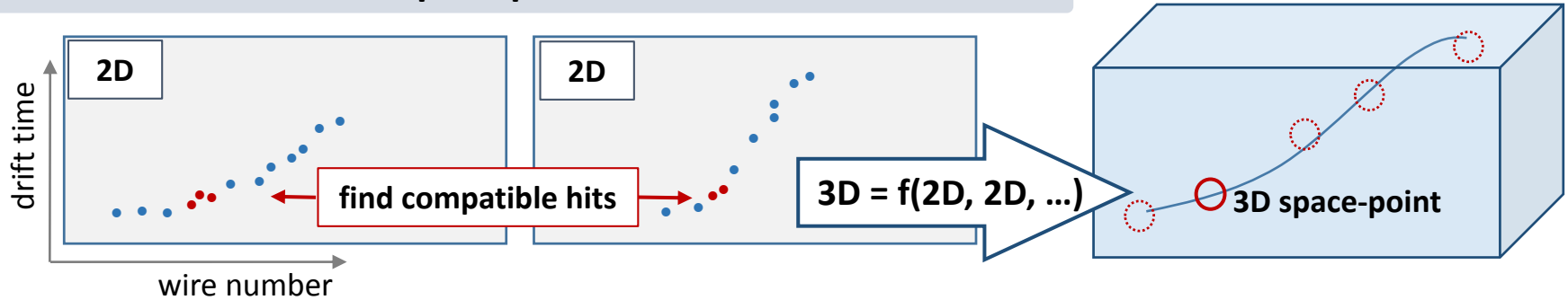# Projection Matching Algorithm for track 3D reconstruction
- LArSoft implementation

D. Stefan, R. Sulej
NCNR Warsaw
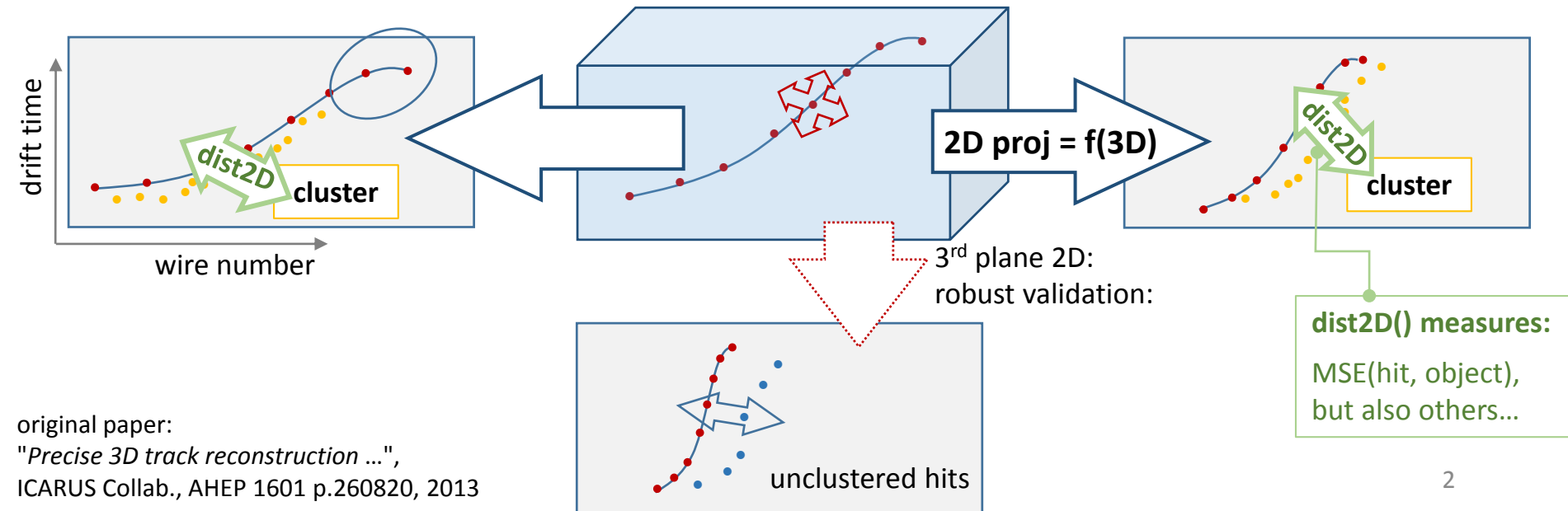
Another approach to build 3D things in TPC

usual: n **2D hits -> 3D space point -> 3D tracks, etc.**

2D

drift time

find compatible hits

wire number

3D = f(2D, 2D, ...)

3D space-point

up side down: *Projection Matching Algorithm*

work in 3D (on *single tracks* or *full track structures*) to match 2D projections to hits

drift time

dist2D

cluster

wire number

2D proj = f(3D)

dist2D

cluster

3rd plane 2D:
robust validation:

unclustered hits

dist2D() measures:

MSE(hit, object),
but also others...

2

Algorithm features

- **no explicit hit-to-hit associations between 2D planes**
- **simultaneous use of information from all planes**
- **3D objects driven by 2D parts, not only isolated points**
  - individual 2D planes can have some missing information (due to dificult track orientation, hit/cluster inefficiency, hardware, …)
- 3D optimization can take into account also 3D points: vertices, feature points, … - that were found with another algorithms

Typical use

- associate two clusters -> optimize 3D -> validate in the 3rd 2D view
- initial clusters pair do not need to be precisely corresponding
- grow / complete the track by adding compatible clusters
- **stich tracks, find and conect full 3D structures -> reoptimize 3D**
- do analysis: initial directions, track dE/dx, PID, energy, …

PMA in LArSoft

- **PMA engine in: larreco/RecoAlg/PMAlg/***

- **Algorithm interface class:** larreco/RecoAlg/**ProjectionMatchingAlg**.h&cxx

  - few basic functions to create, extend and validate tracks

  - few basic parameters to controll algorithm

  - more to be added (to expose settings used in pma::Track3D)

    - weights used to combine information from different planes

    - weights assigned to 3D points from other algorithms

    - functionality for freezing track nodes (shower reco needs this)

    - …

- **Module to create tracks from clusters:**
  larreco/TrackFinder/PMAlgTrackMaker_module.cc

  - very basic logic to loop over clusters, first quick example and test of the algorithm implementation

  - loop starts from the largest cluster (any plane), finds best matching cluster by drift time span (any other plane), validates track (if 3rd plane available)

  - **many other logics possible** – we'll try, and we encourage others as well

First attempts   (standard linecuster used as input)

Long, high energy muons crossing the detector, 5mu/event, large sample:

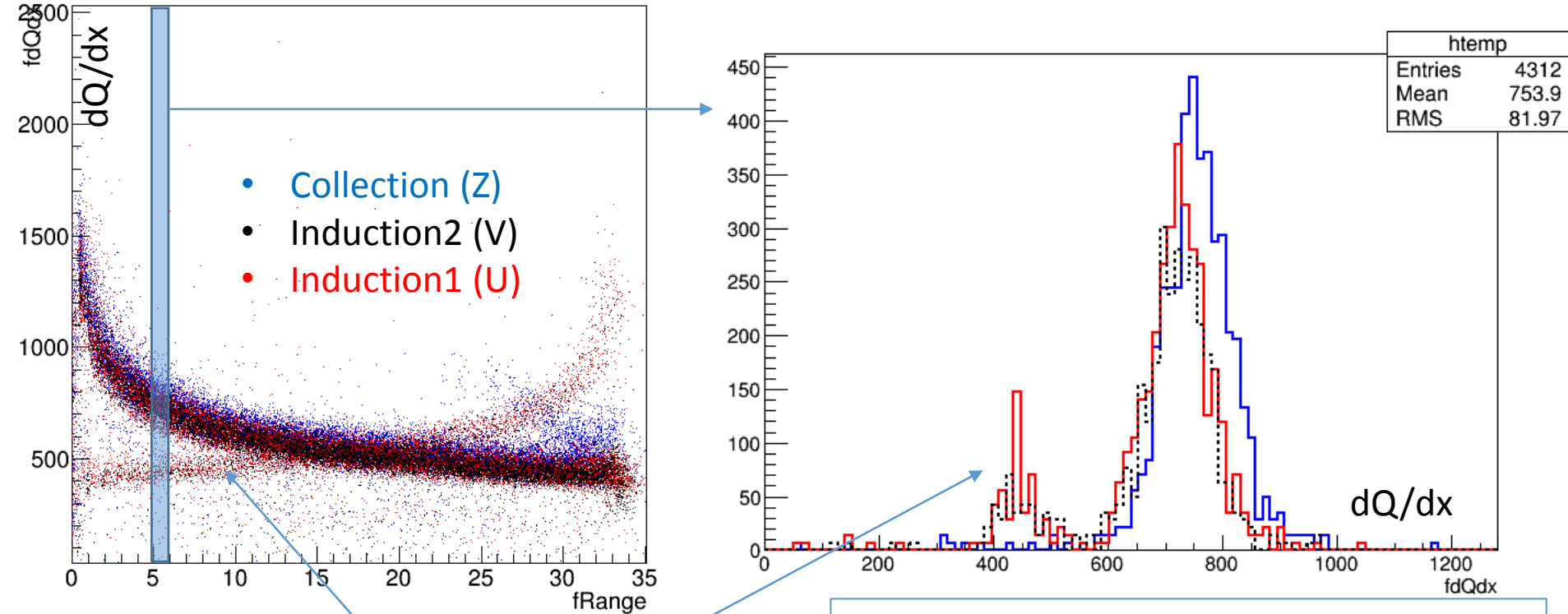MC   reco
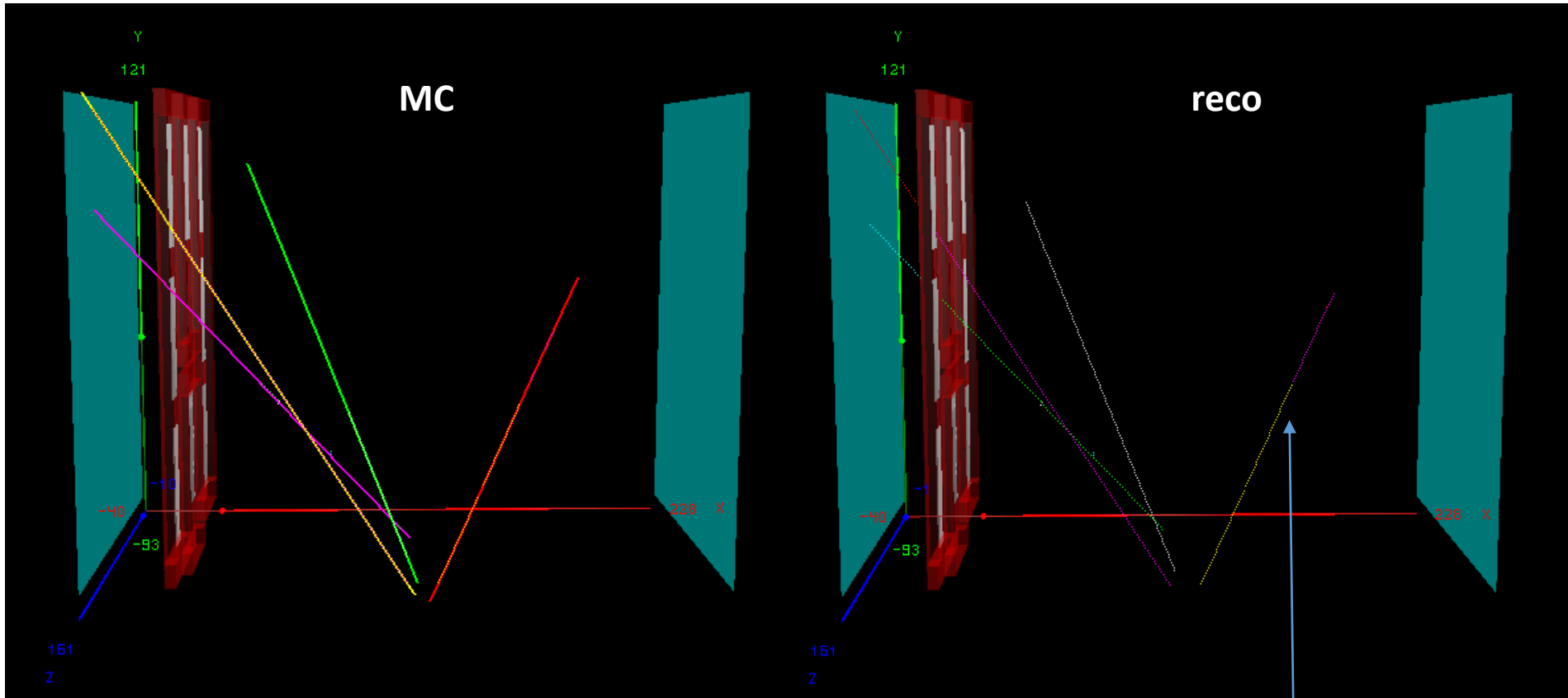
no stitching yet

Systematic efficiency measure needed, of course.
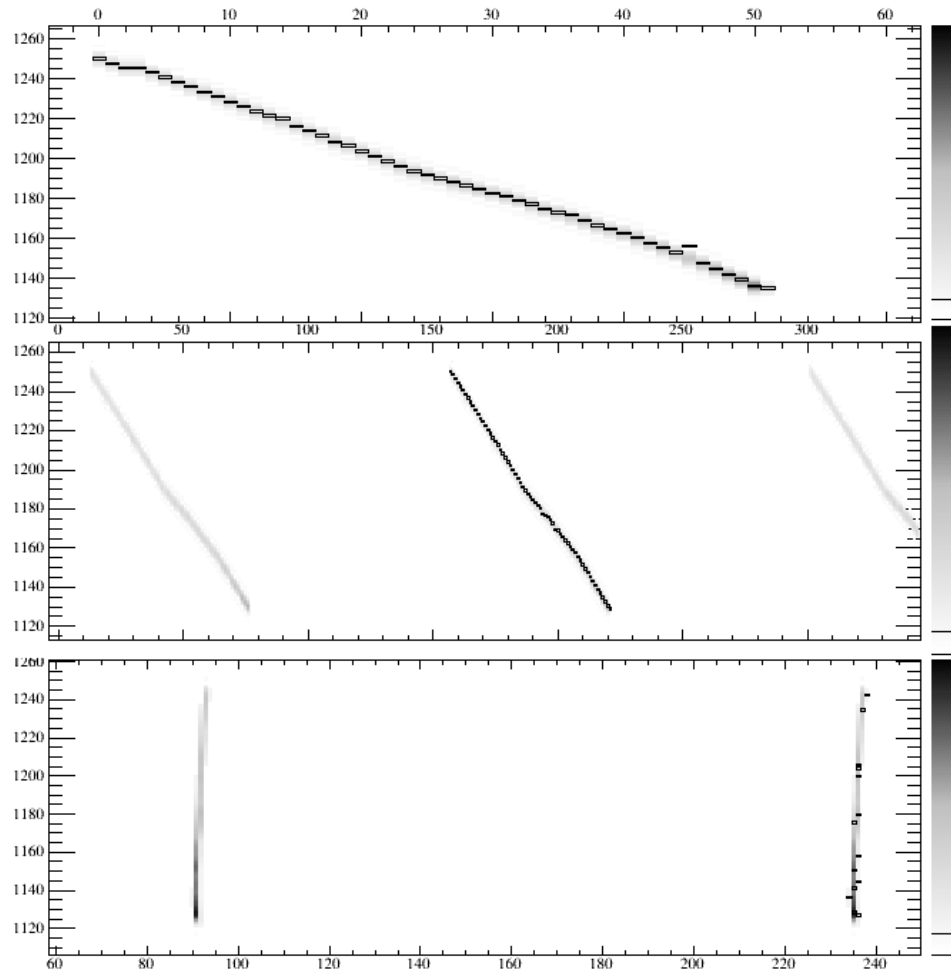
First attempts (standard linecuster used as input)

Single, low energy protons (700MeV/c, ~30cm), dQ/dx reconstruction:

- Collection (Z)
- Induction2 (V)
- Induction1 (U)

htemp	
Entries	4312
Mean	753.9
RMS	81.97

dQ/dx in Induction planes lower than in Collection (Tingjun says it is known behaviour).

Automatic flip of the track direction failed for a few protons with short projection in Collection, to be corrected with Induction views.
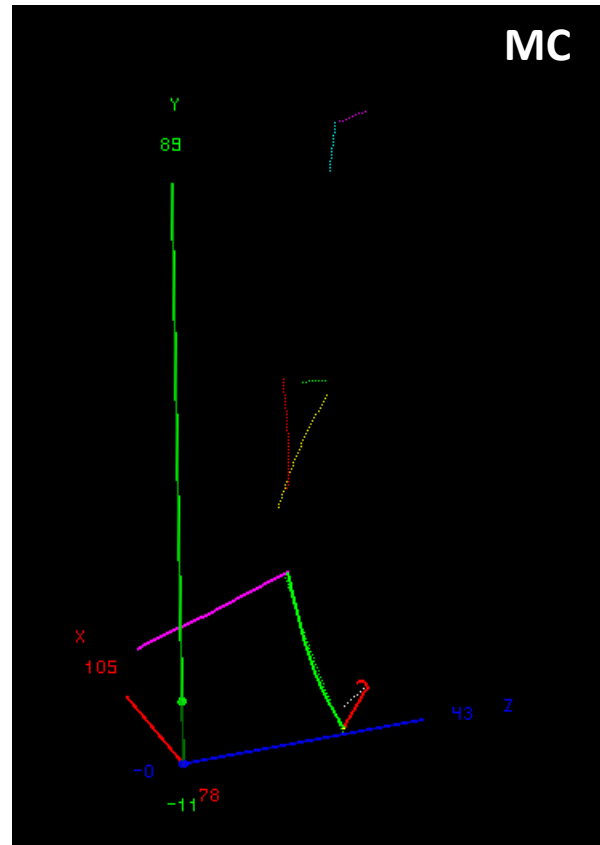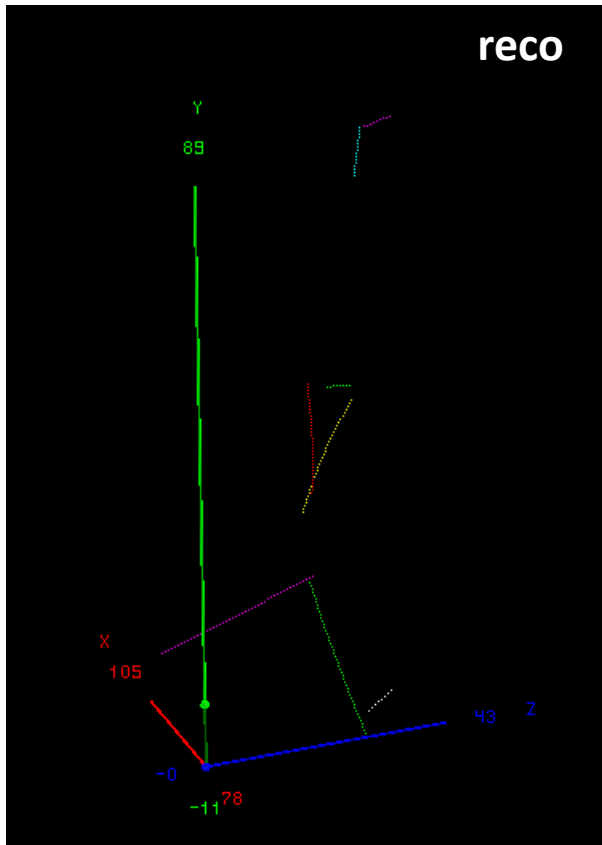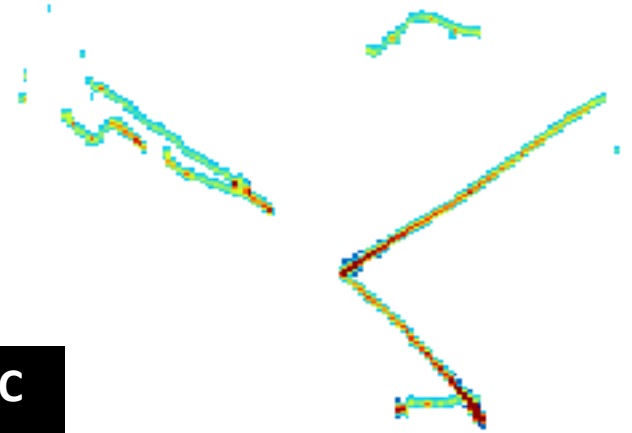
Hit reconstruction to be tuned:

- Hit peak time delays due to deconvolution to be optimized

- Try to improve params of hit reconstruction for tracks ~parallel to the drift direction.

LArSoft

Run: 1/0
Event: 56

UTC Fri Aug 1, 198(
08:46:33.212287264
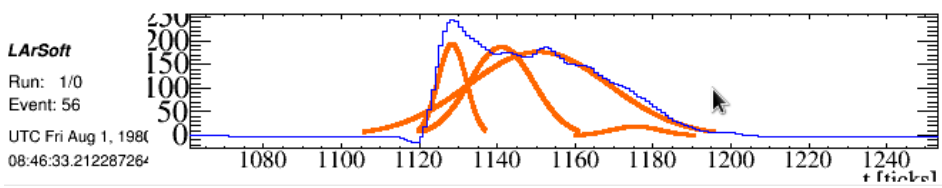
First attempts   (standard linecuster used as input)

Decaying Kaon, just example of a few-track event:

Summary

- **many thanks to Tingjun!!!**

- **it is a pleasure to work in LArSoft environment**

- **basic algorithm is up and running**

- **efficiency measures to be applied (many are for sure ready!)**

- our first goal is shower initial direction -> more functionality for this purpose is coming soon

- validation of 3D in the 3rd plane gives potential to apply the algorithm without disambiguation in „wrapped" planes

- any comments and suggestions are very welcome, we would be also glad to help those interested in using the algorithm

Thank you