

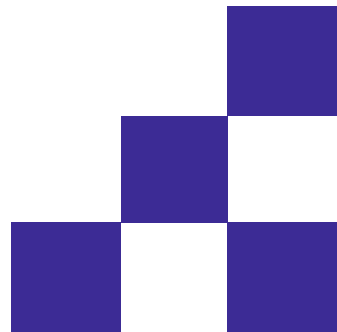
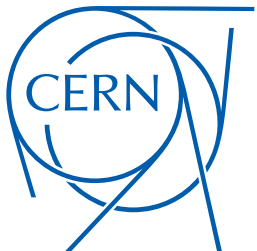


# PMA Track Stitching and Associated Updates

Leigh Whitehead

LArSoft Coordination Meeting

28/02/17



# Introduction

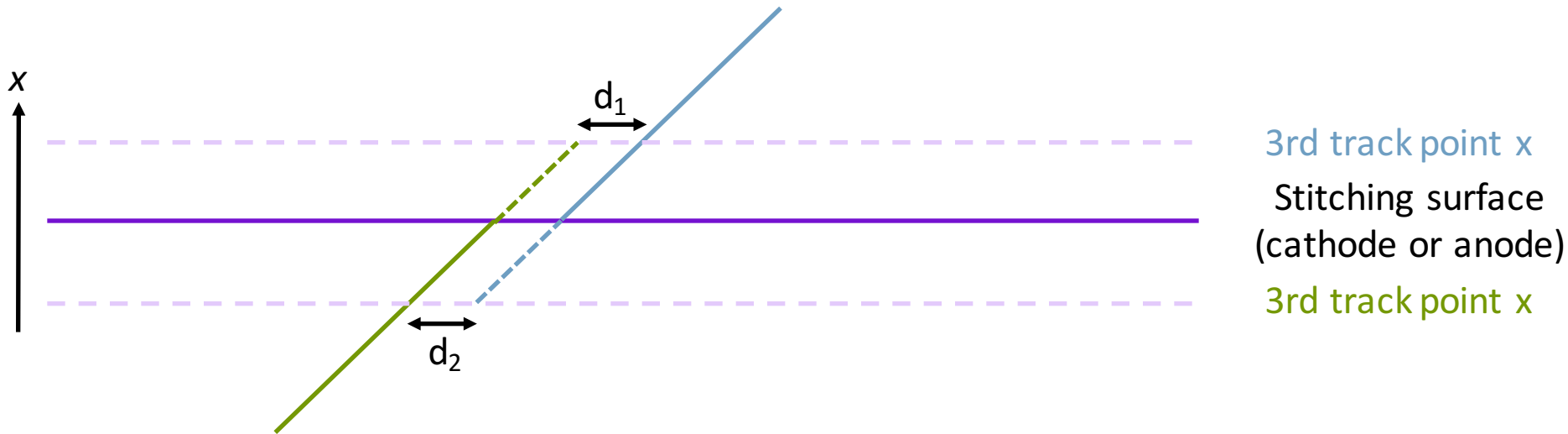
- We need to be able to stitch tracks together across the CPAs and APAs.
  - Generally important in order to have a single particle represented by a single object after the reconstruction.
  - In protoDUNE SP, this is also particularly useful since a track that crosses the CPA gives us a well-defined T0.
- I will give an overview of the method used, along with some examples.
- Finally, I will cover some associated updates that Robert implemented in order to fully integrate this work

# Methodology

- Take a slightly brute-force approach, but make attempts to reduce the dimensionality of the problem.
- Take a track, and shift it in  $x$  (this is perfectly valid since we don't know what the  $x$ -coordinate really is) to either the cathode or the anode.
- Iterate over all other tracks, shifting them to the same cathode or anode (if geometrically possible), and try to match the tracks together.
  - Matching attempted between all 4 combinations of track ends.

# Matching Approach

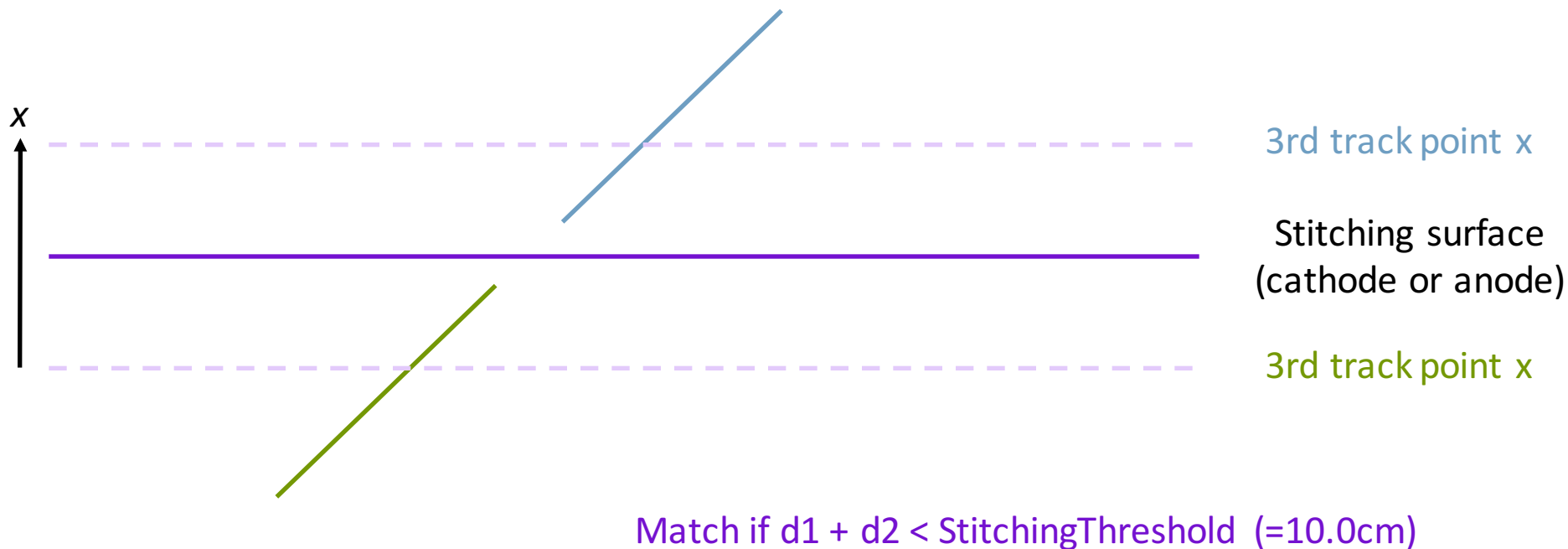
- Extrapolate from the 3<sup>rd</sup> final node on each track to the x-coordinate of the 3<sup>rd</sup> node point on the other.
  - Try to mitigate any odd end effects. ← Actually a tunable parameter: NodesFromEnd (= 2)
  - Initial shift to move tracks to the stitching surface



Whilst  $d_1$  and  $d_2$  are shown here in 2D, it is done in 3D in the code.

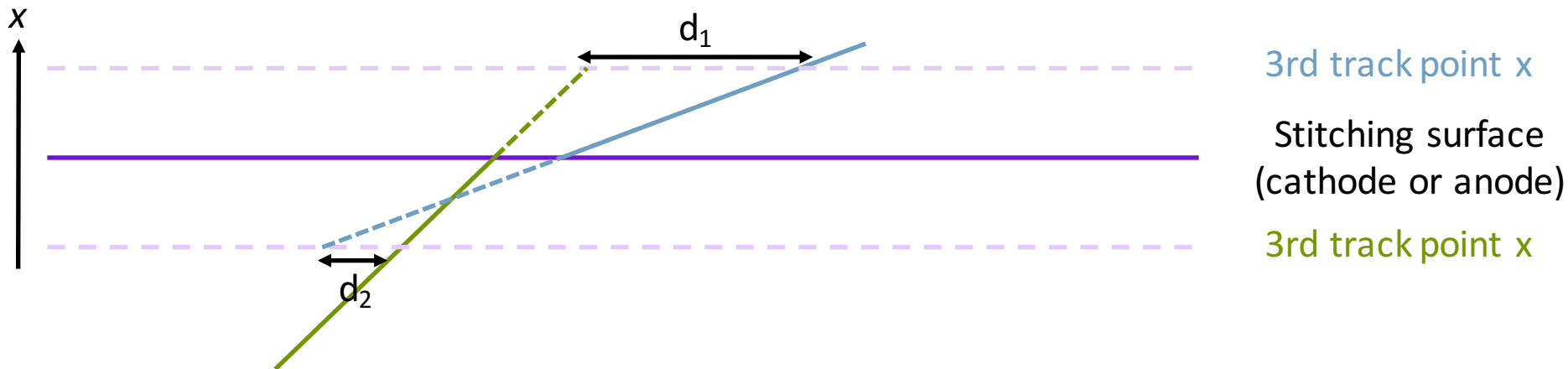
# Matching Approach

- Extrapolate from the 3<sup>rd</sup> final point on each track to the x-coordinate of the 3<sup>rd</sup> final point on the other.
  - Vary the x-shift within a few centimetres to minimise  $d_1 + d_2$
  - A good match has small  $d_1 + d_2$  (in this picture, actually zero)



# Matching Approach

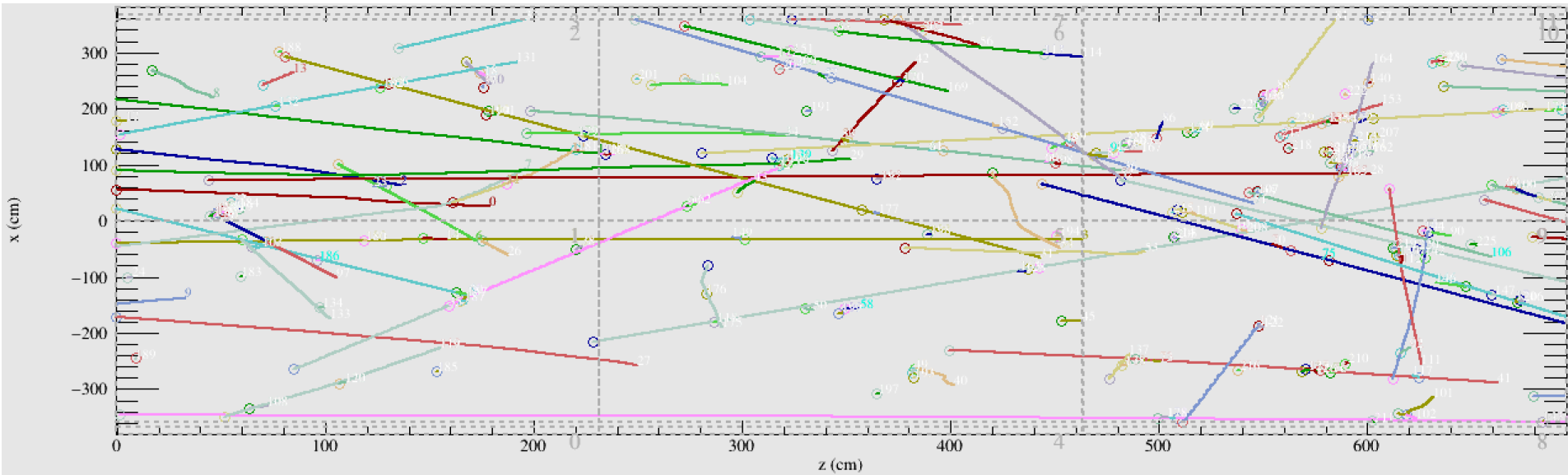
- Extrapolate from the 3<sup>rd</sup> final point on each track to the x-coordinate of the 3<sup>rd</sup> final point on the other.
  - Vary the x-shift within a few centimetres to minimise  $d_1 + d_2$
  - A poor match has large  $d_1 + d_2$



Match if  $d_1 + d_2 < \text{StitchingThreshold}$  (=10.0cm)

# Stitching Example - ProtoDUNE

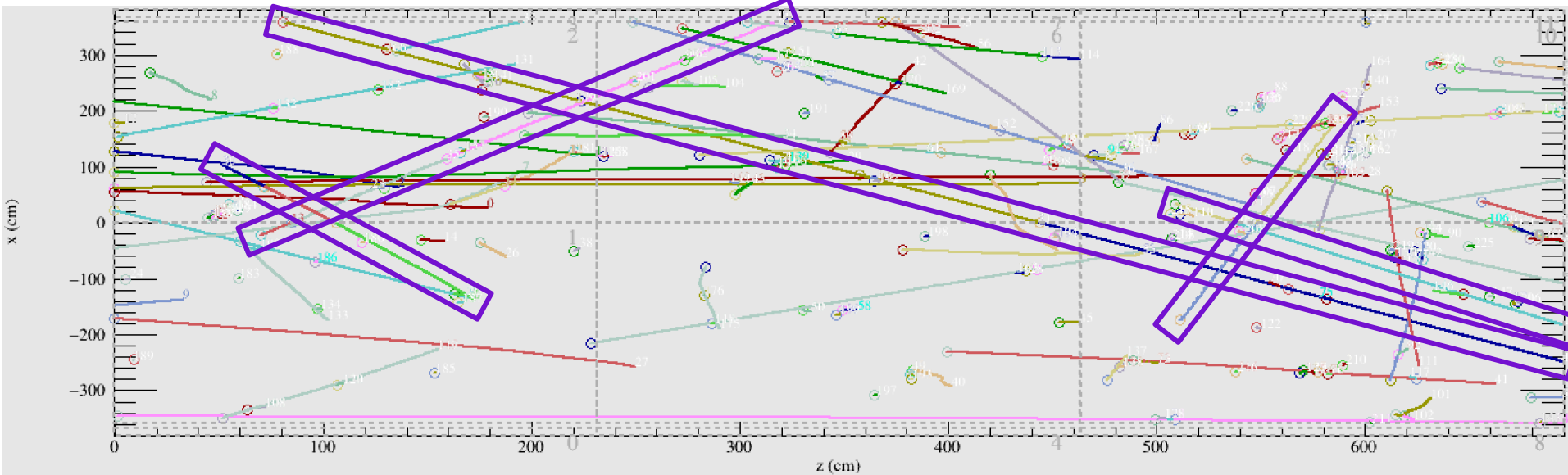
- Example of a beam + cosmics event from protoDUNE SP.



- There are five tracks here broken across the cathode.
  - Easiest to see by toggling between this slide and the next.

# Stitching Example - ProtoDUNE

- Example of a beam + cosmic event from protoDUNE SP.

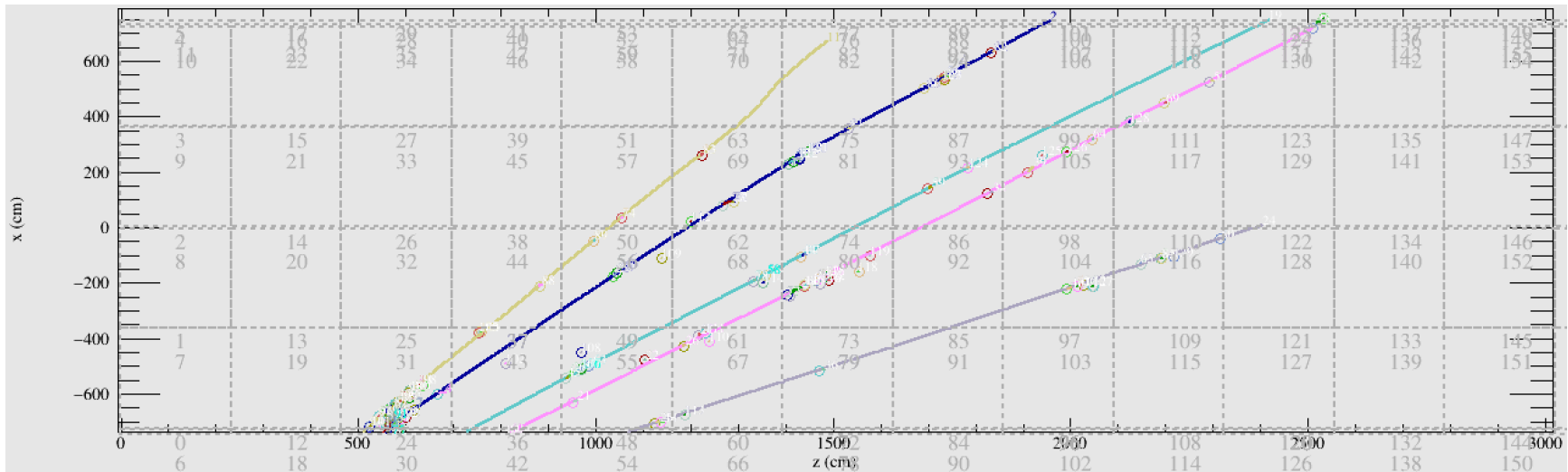


- After stitching these tracks are moved into the correct positions.
  - NB: different track colours either side of the cathode as since this example was from before the updates from Robert (see slides 11 and 12 ).



# Far Detector Example

- After all of the changes, we now have continuous tracks across multiple TPCs.

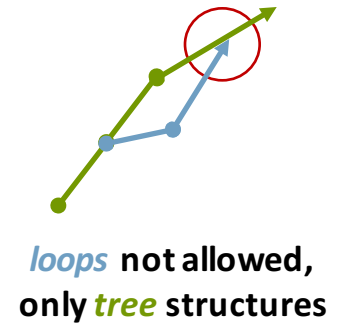
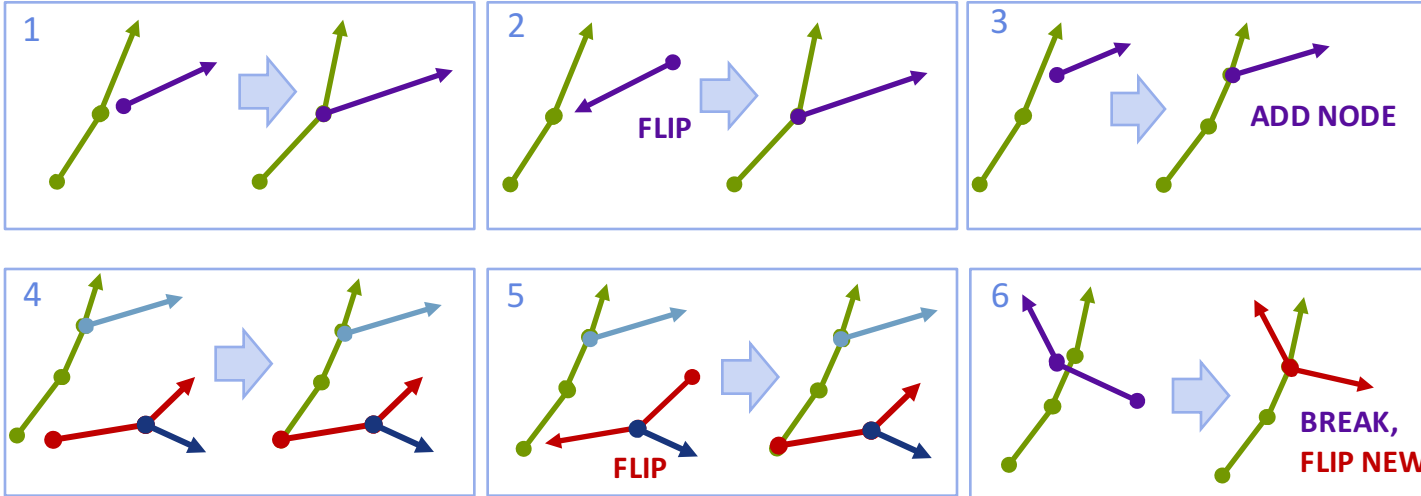


- Most tracks here are stitched 3 times, and the single colour of each track shows each is formed from a single recob::track object.

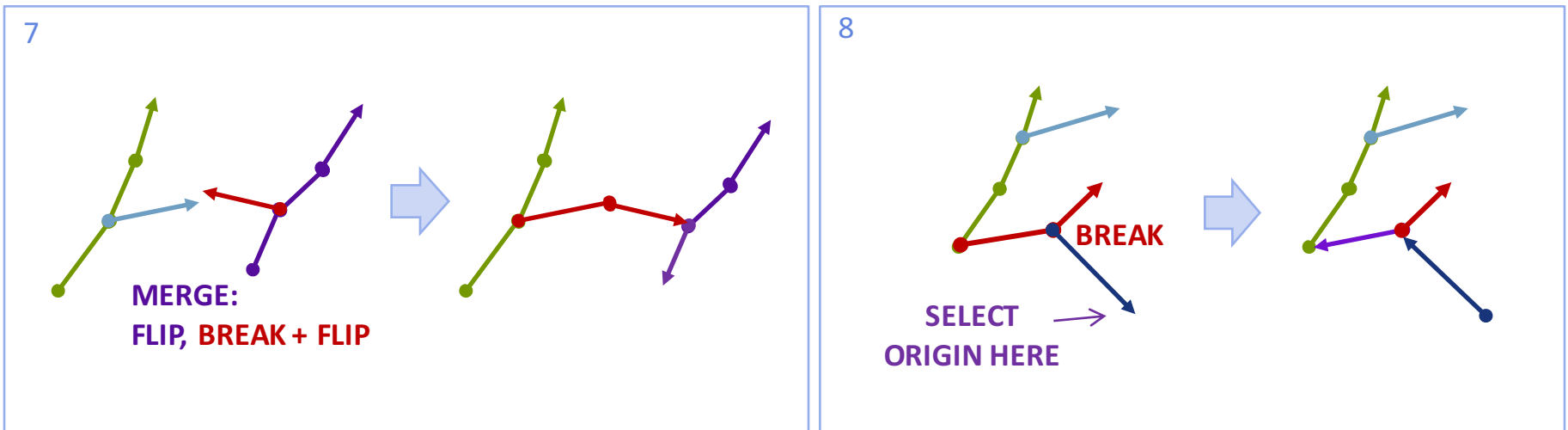
# Code Implementation

- New algorithm added to larreco:
  - larreco/larreco/RecoAlg/PMAAlgStitching.cxx / .h
  - larreco/larreco/RecoAlg/pmastitchalg.fcl
- Updates to existing code:
  - larreco/larreco/TrackFinder/PMAAlgTrackMaker\_module.cc
    - Added parameters for the stitching config.
  - larreco/larreco/TrackFinder/trackfindermodules.fcl
    - Added reference to pmastitchalg.fcl
  - larreco/larreco/TrackFinder/trackfinderalgorithms.fcl
    - Set PMA defaults for MatchT0in[CPA/APA]Crossing to false for now.
  - larreco/larreco/RecoAlg/PMAAlgTracking.cxx / .h
    - Stitcher added and replaces matchCoLinearAnyT0 function.
    - Added new parameter “MatchT0inCPACrossing” to partner the existing “MatchT0inAPACrossing”.

# Vertexing in PMA

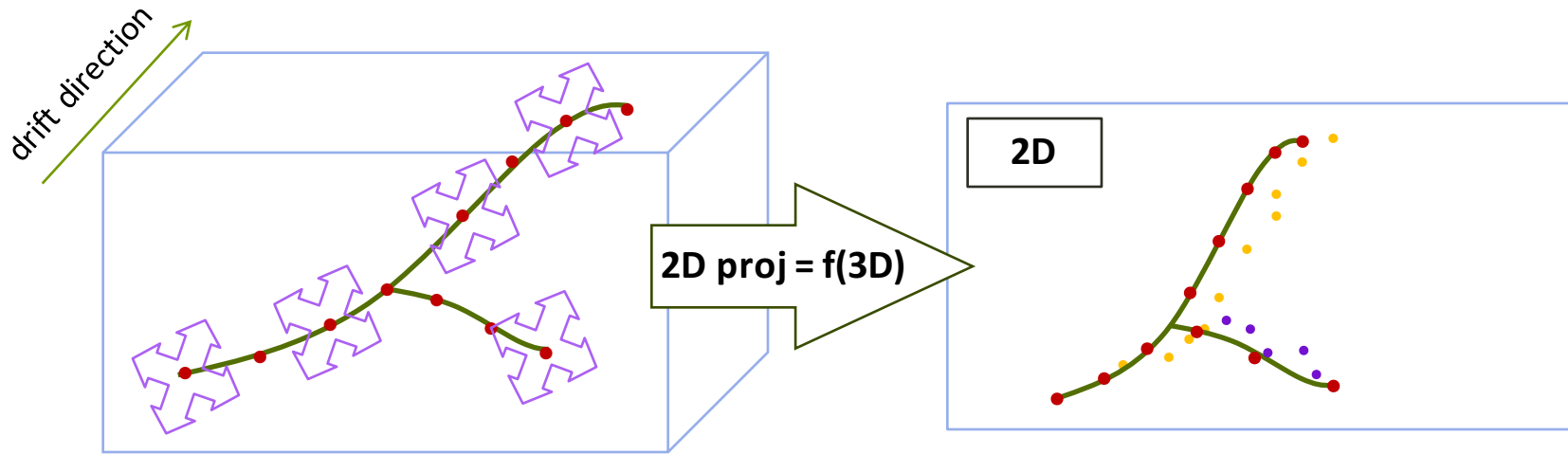


Missing operations implemented, allowing cases like these:

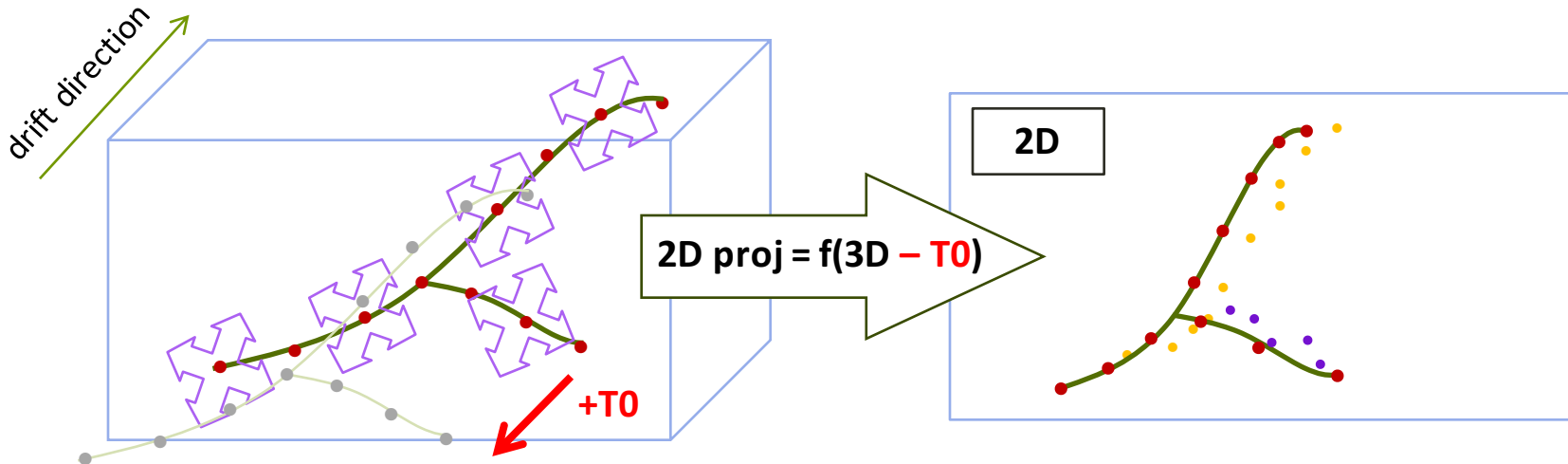


Feature enabled in merges during stitching and in dE/dx based auto-flip.

# PMA (re)optimisation with T0 shift



Track-vertex structures with any  $T_0 \neq 0$  can be optimized/re-optimised to match 2D projections (negligible impact on code speed)



# Summary

- We have an algorithm to stitch together PMA tracks across CPAs and APAs.
  - Have tested on simulation of protoDUNE SP, the FD, and 35t.
- Implementation lives in larreco.
  - The base .fcl file has the stitching in both CPA and APA disabled by default.
  - I imagine this will just be overwritten by experiment specific fcl files, since 35t has no CPA boundaries and protoDUNE SP has no APA boundaries.
- Output from pma will look different due to vertexing changes.
  - Vito will produce new references for the standard tests.
- Branch was merged in for the v06\_26\_00 release.