Somewhat-frozen TMS track finding

UNIVERSITY of ROCHESTER

Clarence Wret

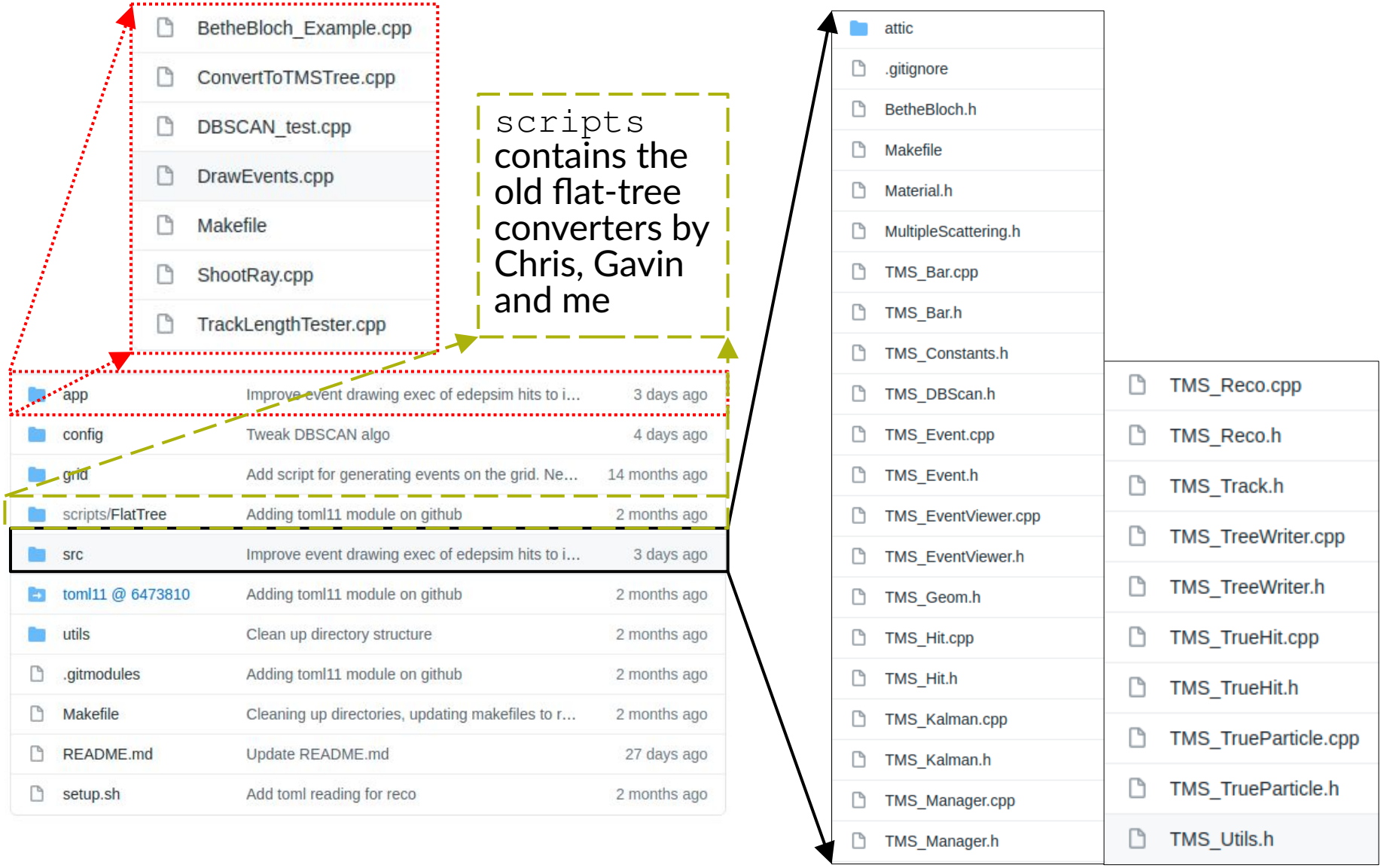
DUNE TMS meeting
July 26 2021

Introduction

- Haven't given a talk on the TMS data products and reco in a while, and it's reached good maturity
- Have cleaned up repo signficiantly, with decent documentation and commenting
- TOML config files and main manager, output tree controlled by tree-writer class, "event display" handler class does pdfs
- Have also somewhat-optimised track finding algorithms
- Essentially, all is in a pretty stable state now

Repository structure

- Nothing too unfamiliar I hope: `app`, `config`, `grid`, `src`, `utils`, `scripts`



Configurations

- Most settings configured with toml
- Controlled by a manager singleton (`TMS_Manager`)

```
# Recon parameter constants
[Recon]
  # Minimum number of hits to consider reconstructing the event
  MinHits = 10

  # Track reco; Hough, AStar, DBSCAN
  #TrackMethod = "Hough"
  TrackMethod = "AStar"
  #TrackMethod = "DBSCAN"

  # Run clustering algorithm (DBSCAN) after track finding?
  Clustering = true

  [Recon.DBSCAN]
    # The minimum points for the DBSCAN clustering algorithm
    MinPoints = 3
    # How far away can a bar/plane combination be to include in a cluster
    Epsilon = 2.5

  [Recon.Hough]
    # The highest number of transform we can run
    MaxTrans = 10
    # Intercepts
    MinInter = -40.0E3
    MaxInter = 40.0E3
    # Slopes
    MinSlope = -2.0
    MaxSlope = 2.0
    # What factor of hits of total do we need to cover to count a successful Hough line finding
    HitMult = 0.2

  [Recon.AStar]
    #CostMetric = "Manhattan"
    #CostMetric = "Euclidean"
    CostMetric = "DetectorZ"
    #CostMetric = "DetectorNotZ"
    IsGreedy = false
```

Track finder method

Run clustering after track finding?

DBSCAN

Specific algorithm settings

Hough

A*

To add new setting:
1) Add new line in config
2) Add corresponding setter/getter in `TMS_Manager`

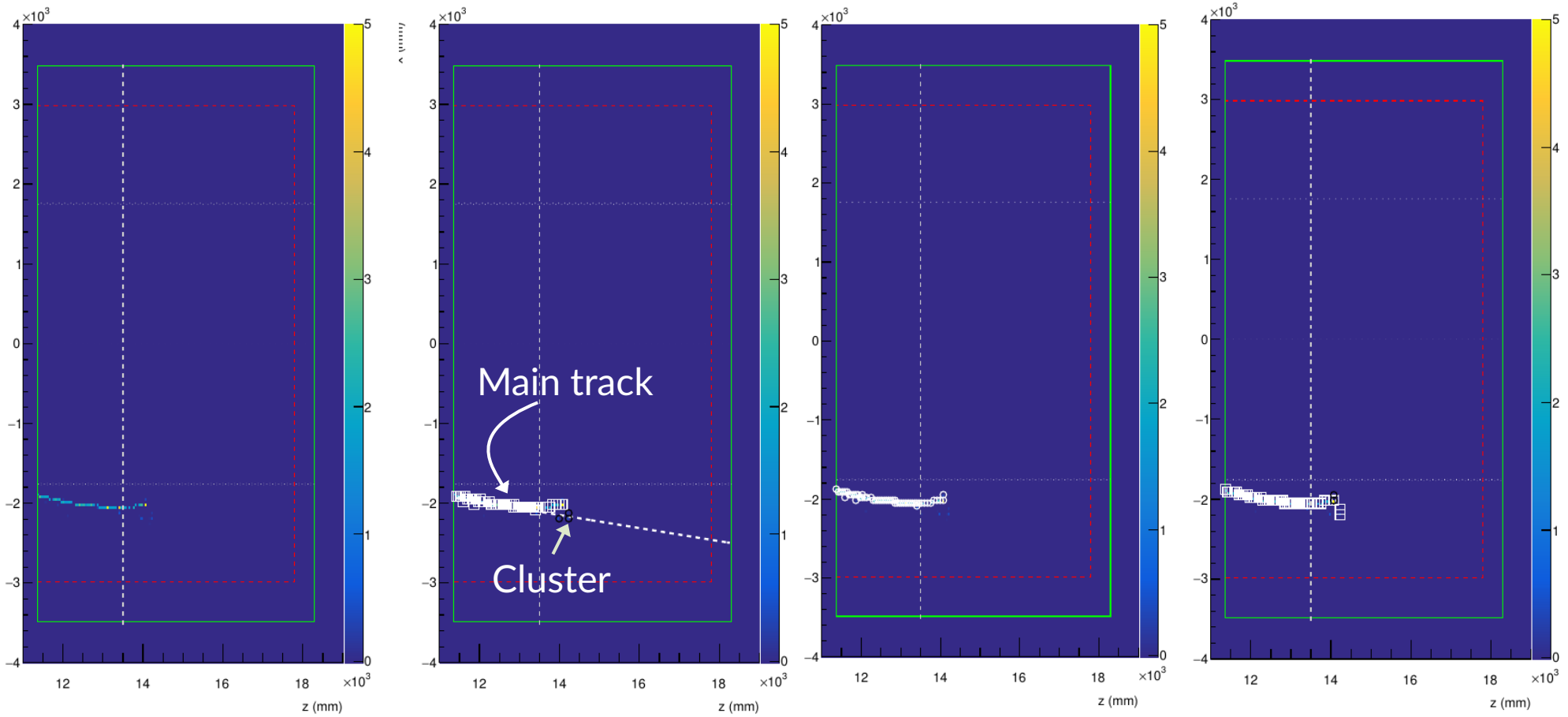
Algorithms

- Linear Hough transform (track)
  - Can run on most upstream section or entire detector
  - Performs very well for multiple tracks
- A* (track)
  - Could do with some improvements on selecting good start and end points
  - Can get confused with multiple tracks
  - Can run on Hough transform hits to find fastest path from start to end point
- DBSCAN (track, cluster)
  - Mostly wrote to bundle up hits after the track finder
  - Turns out it's also quite OK at finding tracks, as long as they're not **too** close

Algorithms

- Most of the time, TMS events are very straight forward to reconstruct for all algorithms

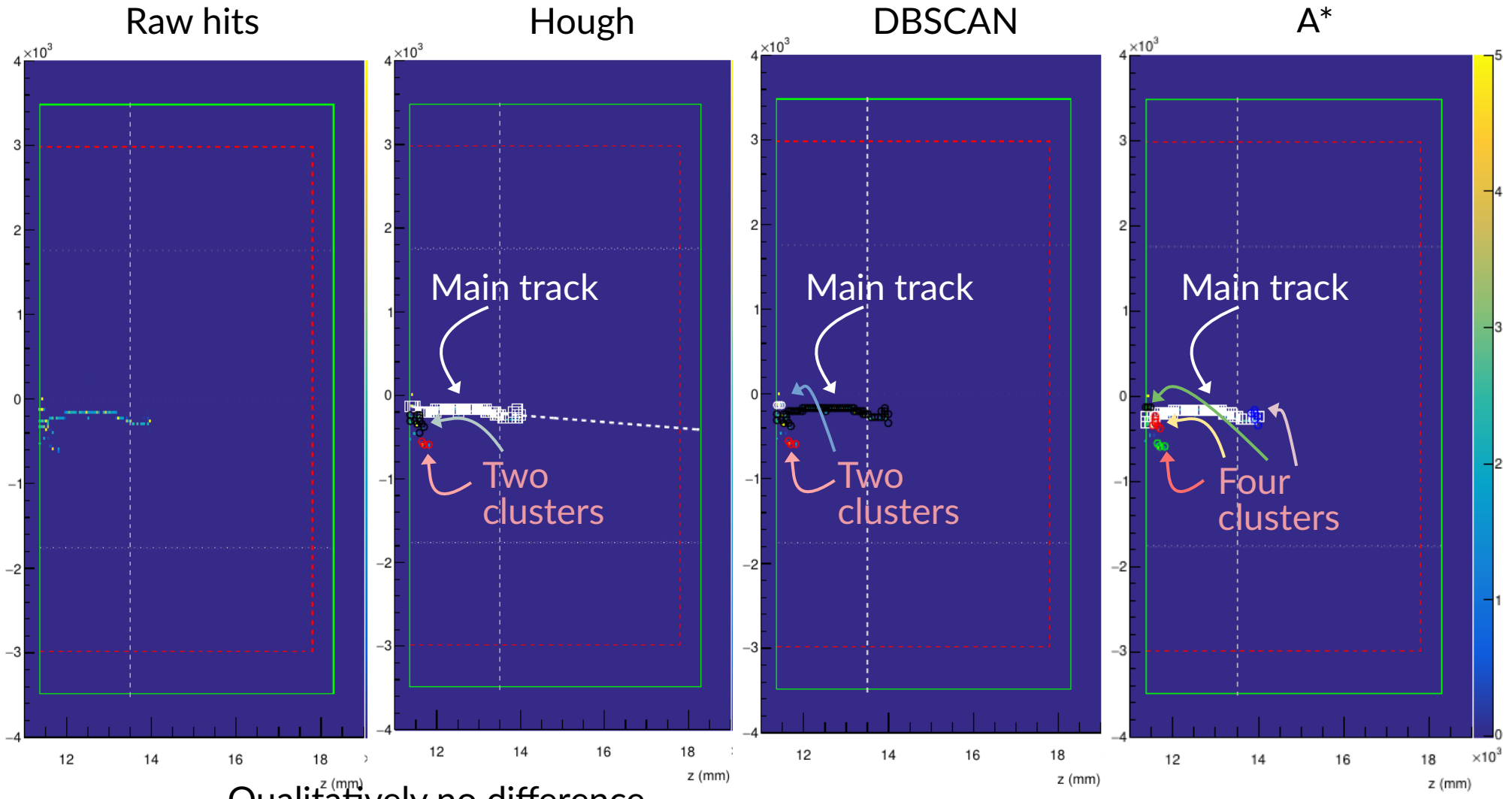
Raw hits | Hough | DBSCAN | A*



No difficulties reconstructing these events
End points and “greediness” of algorithm are the only differences

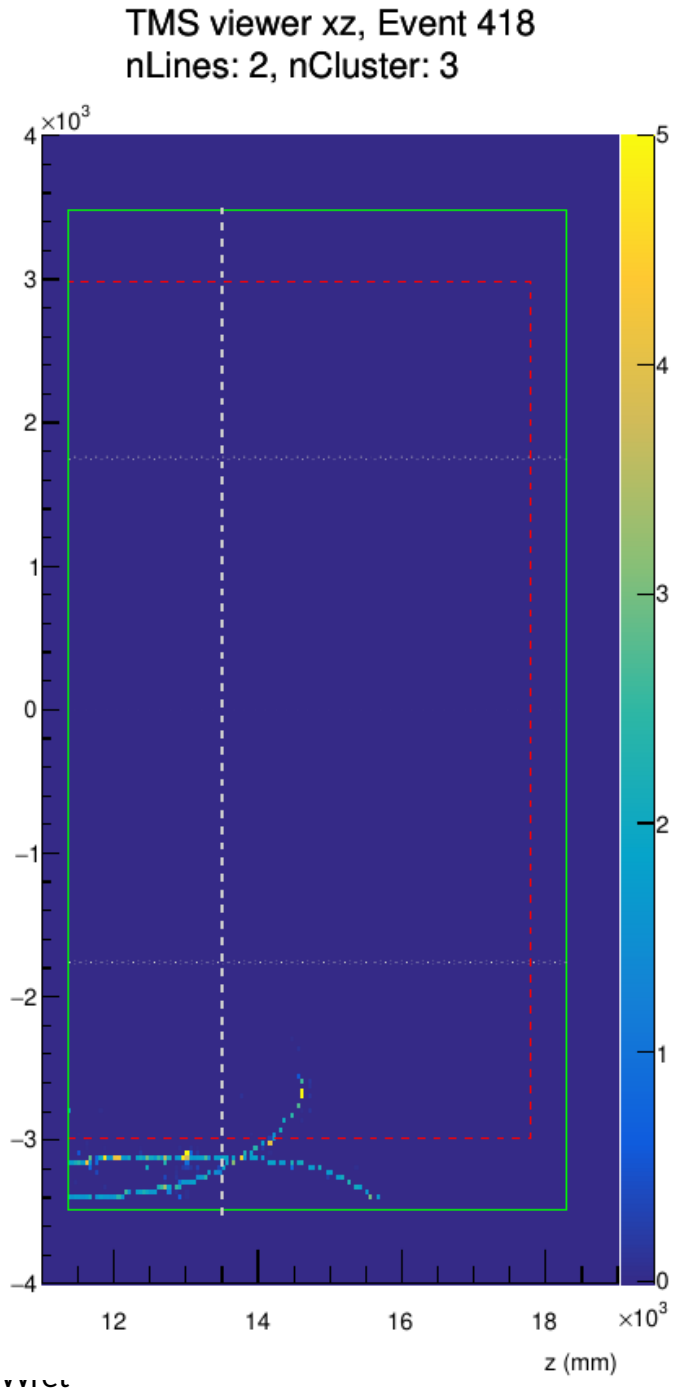
Algorithms

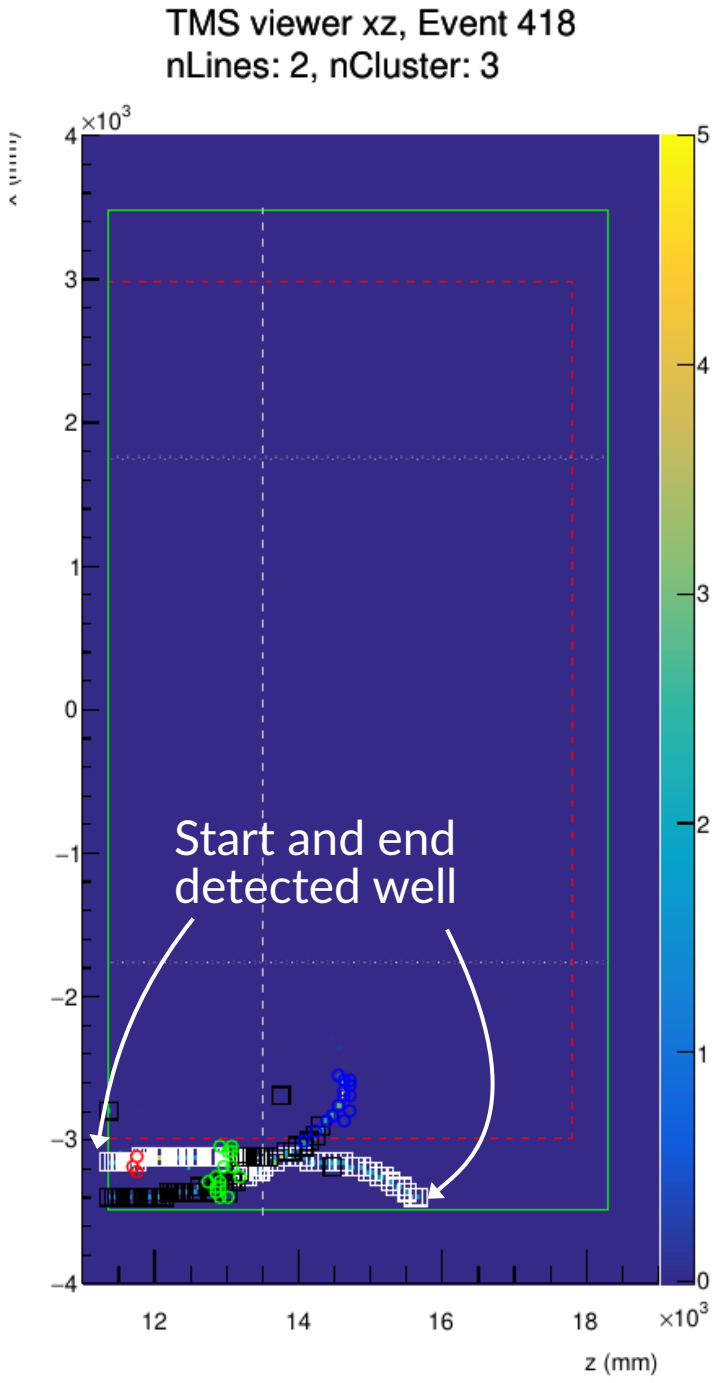
- Most of the time, TMS events are very straight forward to reconstruct for all algorithms



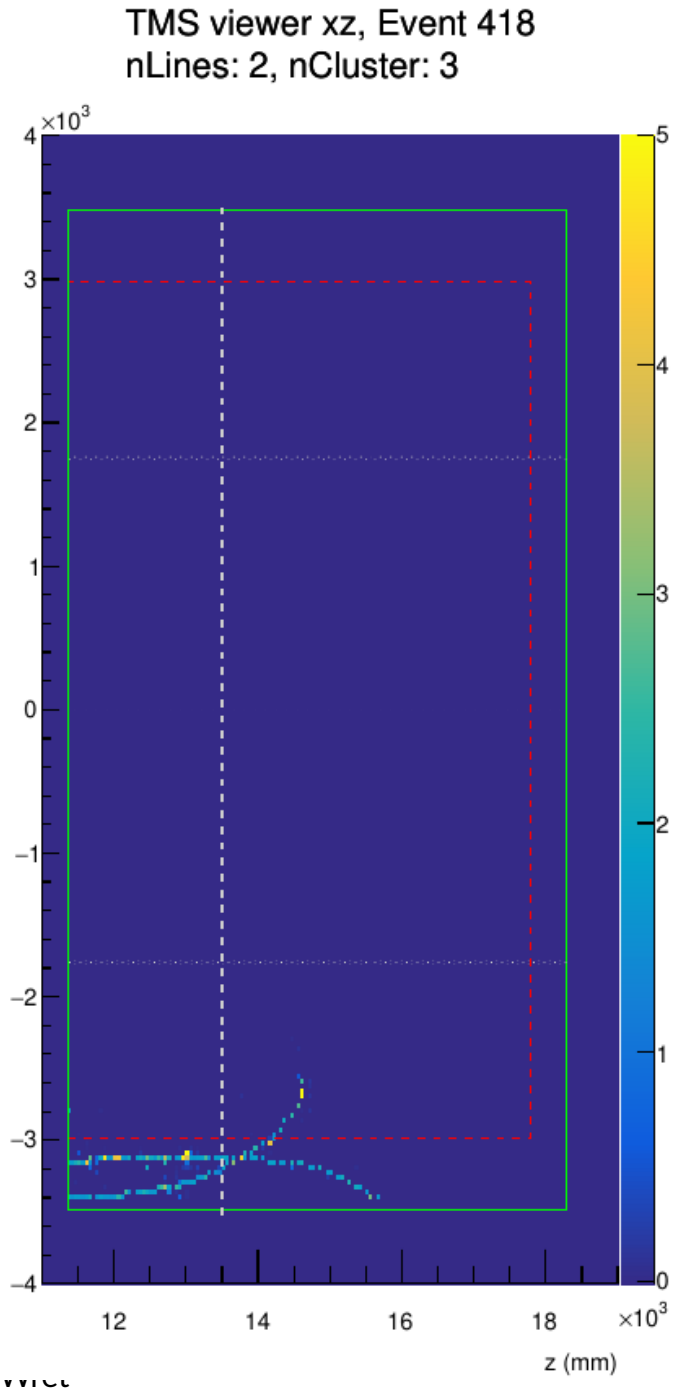
Qualitatively no difference
Start and end-points similar, direction vector the same.
DBSCAN may need A* run on main track to clean up

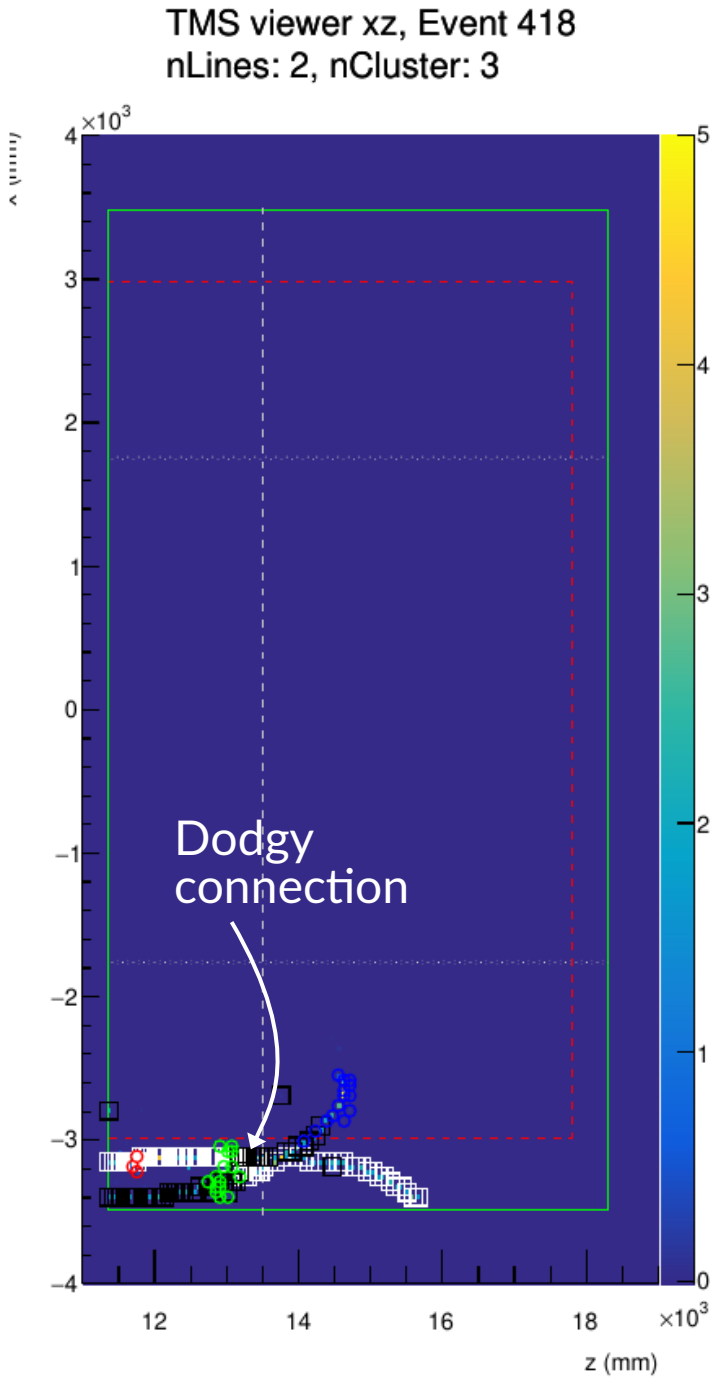
Failure modes



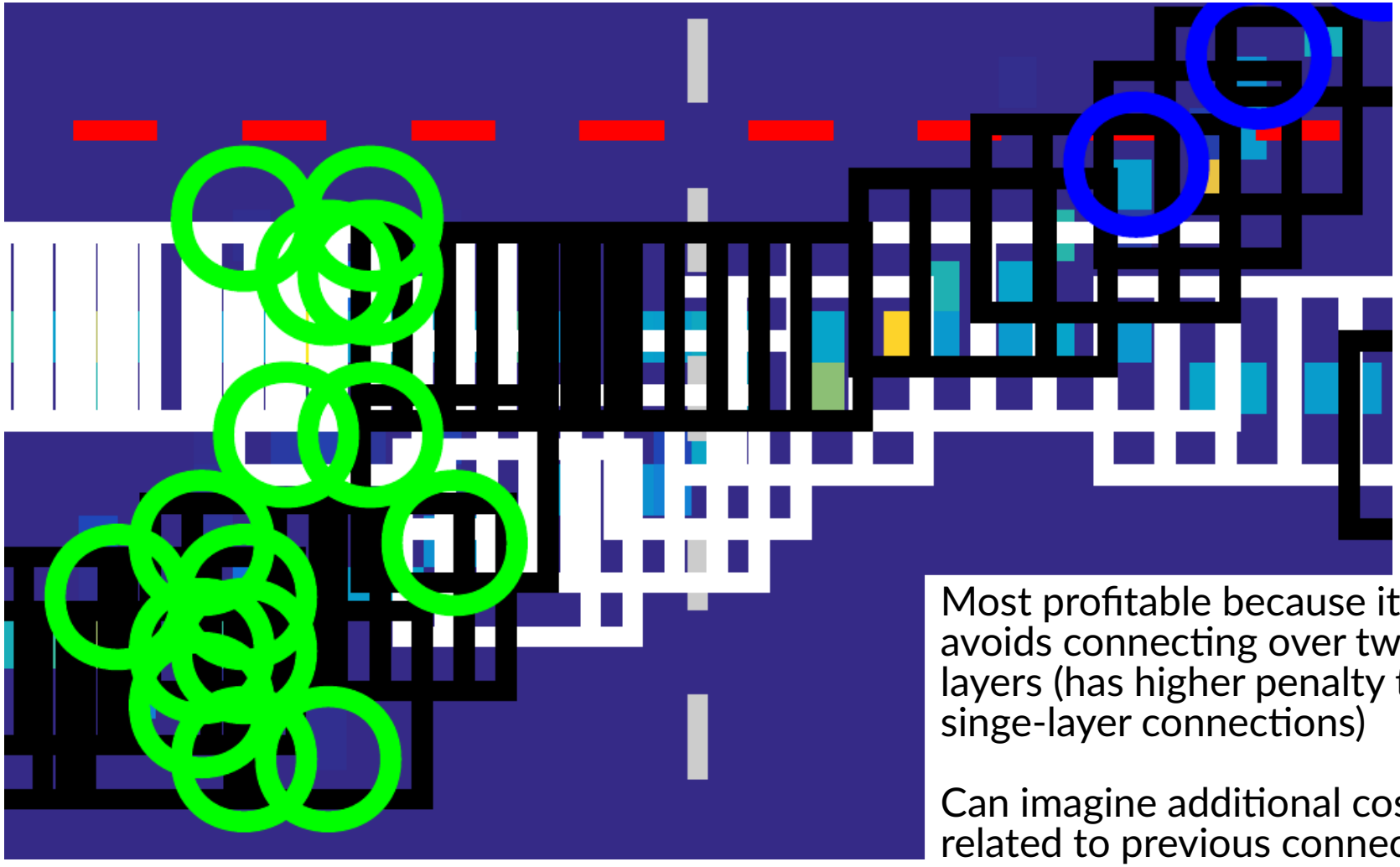


Failure modes





Failure modes

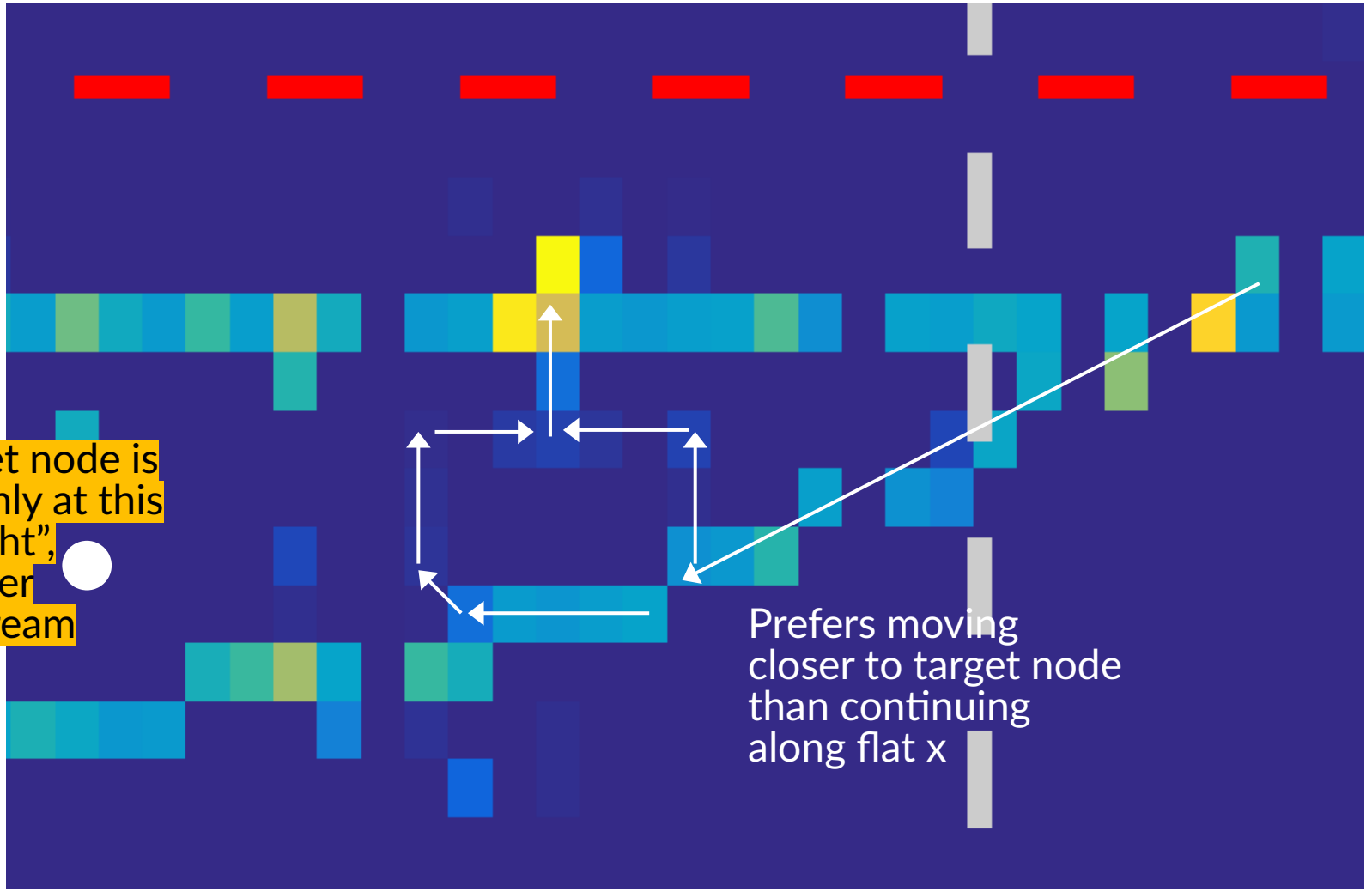


Most profitable because it avoids connecting over two layers (has higher penalty than singe-layer connections)

Can imagine additional cost related to previous connection

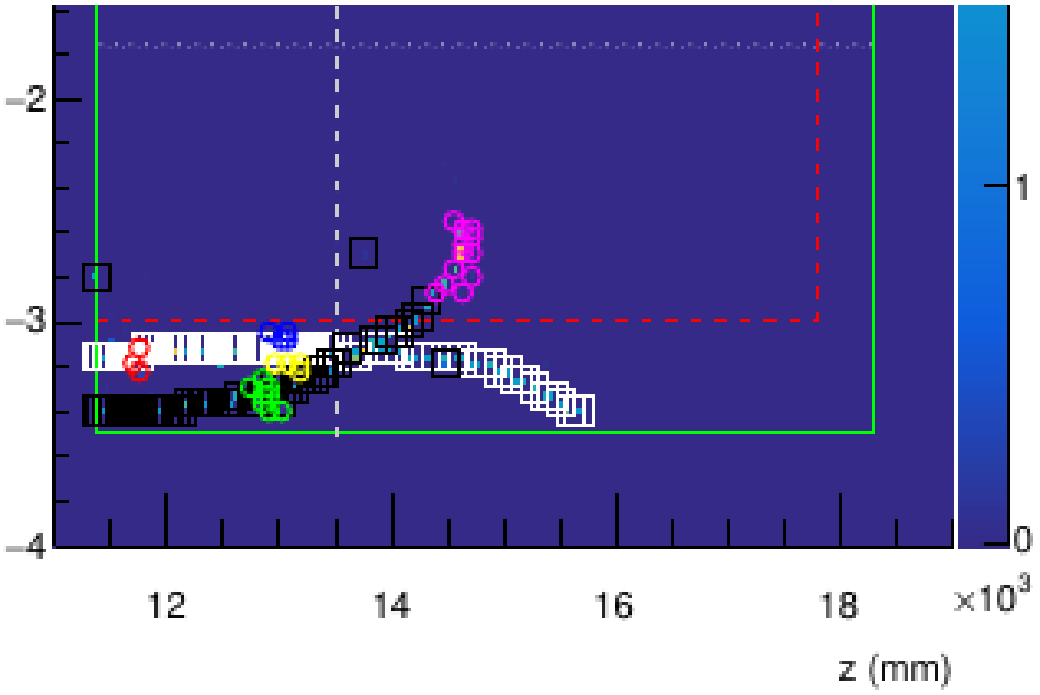
Or simply penalise upwards connections more than sideways connections

Failure modes



Failure modes

- Use Euclidean distance instead of “Manhattan” (Δx + Δy), and vertical moves are double cost to horizontal moves
- Then works mostly well!

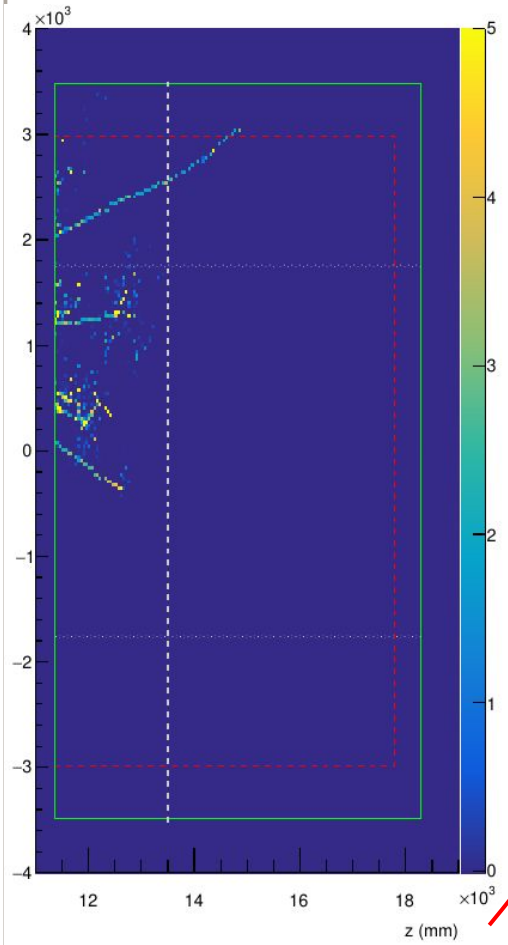


- Also noticed if **only** use the plane distance (z-only) in heurisitic, performance improves

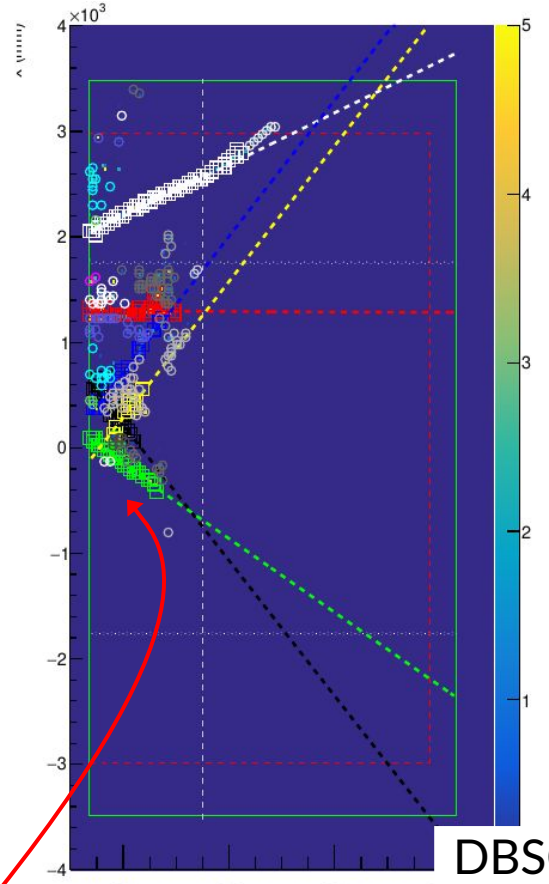
Failure modes

- “high multiplicity events”

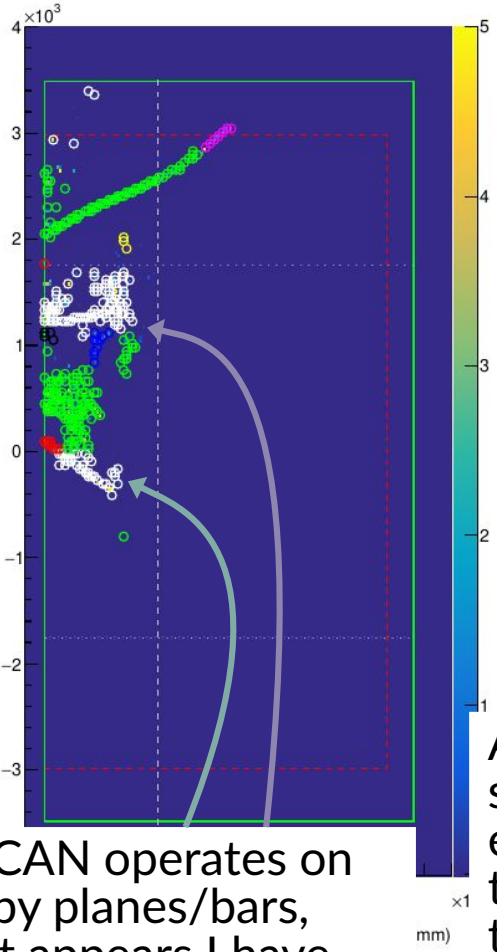
Raw hits



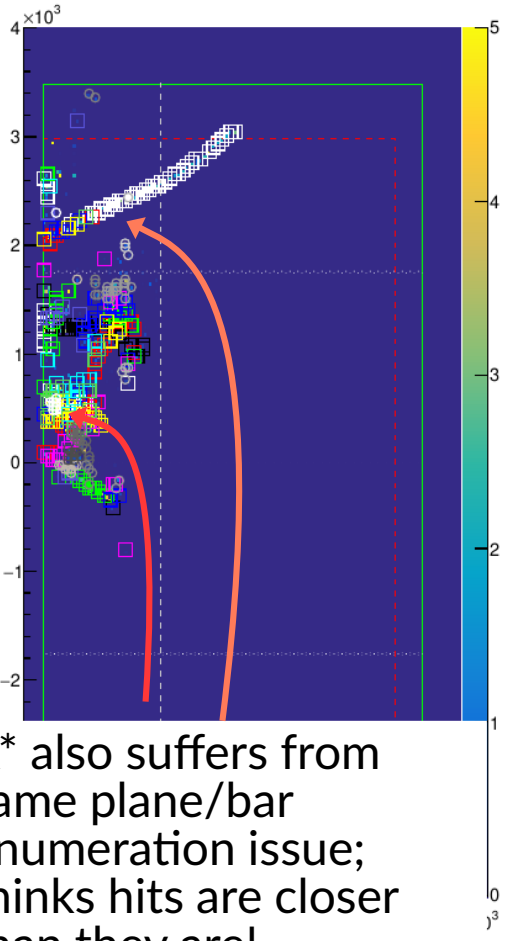
Hough



DBSCAN



A*



Hough transform scales very well and finds all three tracks

DBSCAN operates on nearby planes/bars, and it appears I have the numbering wrong from the geometry: merges hits far from each other!

A* also suffers from same plane/bar enumeration issue; thinks hits are closer than they are!

Applications

- General converters for edep-sim files
  - Display hits in LAr and TMS
  - Run the full reconstruction, printing to pdf and writing a ROOT file with track and cluster candidates
- Track finder testing applications
- Geometry tools, needed for Kalman filter
  - Path length calculator using ROOT geometry
  - What material is between point x and y, etc

Output format

- Currently separated into Truth and Reco

```
root [1] _file0->ls()
TFile**         neutrino.1.edep_LineCandidates.root
 TFile*         neutrino.1.edep_LineCandidates.root
  KEY: TTree    Truth_Info;11   Truth_Info
  KEY: TTree    Truth_Info;10   Truth_Info
  KEY: TTree    Line_Candidates;3       Line_Candidates
  KEY: TTree    Line_Candidates;2       Line_Candidates
```

```
root [2] Truth_Info->Print()
******************************************************************************
*Tree    :Truth_Info: Truth_Info                                             *
*Entries :    10000 : Total =         1655370 bytes  File  Size =     853479 *
*        :          : Tree compression factor =   1.94                       *
******************************************************************************
*Br    0 :MuonP4    : MuonP4[4]/D                                            *
*Entries :    10000 : Total  Size=     321412 bytes  File Size  =     235771 *
*Baskets :       11 : Basket Size=      40448 bytes  Compression=   1.36     *
*.............................................................................*
*Br    1 :Muon_Vertex : Muon_Vertex[4]/D                                      *
*Entries :    10000 : Total  Size=     321487 bytes  File Size  =     189897 *
*Baskets :       11 : Basket Size=      40448 bytes  Compression=   1.69     *
*.............................................................................*
*Br    2 :nParticles : nParticles/I                                           *
*Entries :    10000 : Total  Size=      41355 bytes  File Size  =      10951 *
*Baskets :       10 : Basket Size=      12288 bytes  Compression=   3.73     *
*.............................................................................*
*Br    3 :Interaction : string                                                *
*Entries :    10000 : Total  Size=     566497 bytes  File Size  =      85455 *
*Baskets :       11 : Basket Size=      65024 bytes  Compression=   6.62     *
*.............................................................................*
*Br    4 :EventNo   : EventNo/I                                               *
*Entries :    10000 : Total  Size=      41313 bytes  File Size  =      15247 *
*Baskets :       10 : Basket Size=      12288 bytes  Compression=   2.68     *
*.............................................................................*
*Br    5 :NeutrinoPDG : NeutrinoPDG/I                                         *
*Entries :    10000 : Total  Size=      41369 bytes  File Size  =       3412 *
*Baskets :       10 : Basket Size=      12288 bytes  Compression=  11.97     *
*.............................................................................*
*Br    6 :NeutrinoP4 : NeutrinoP4[4]/D                                        *
*Entries :    10000 : Total  Size=     321472 bytes  File Size  =     311233 *
*Baskets :       11 : Basket Size=      40448 bytes  Compression=   1.03     *
*.............................................................................*
```

```
root [3] Line_Candidates->Print()
******************************************************************************
*Tree    :Line_Candidates: Line_Candidates                                   *
*Entries :     2272 : Total =          391563 bytes  File  Size =     155392 *
*        :          : Tree compression factor =   2.50                       *
******************************************************************************
*Br    0 :nLines    : nLines/I                                               *
*Entries :     2272 : Total  Size=       9829 bytes  File Size  =       1557 *
*Baskets :        3 : Basket Size=      12288 bytes  Compression=   6.00     *
*.............................................................................*
*Br    1 :EventNo   : EventNo/I                                              *
*Entries :     2272 : Total  Size=       9836 bytes  File Size  =       4329 *
*Baskets :        3 : Basket Size=      12288 bytes  Compression=   2.16     *
*.............................................................................*
*Br    2 :Slope     : Slope[nLines]/D                                        *
*Entries :     2272 : Total  Size=      37853 bytes  File Size  =      15593 *
*Baskets :        3 : Basket Size=      24576 bytes  Compression=   2.39     *
*.............................................................................*
*Br    3 :Intercept : Intercept[nLines]/D                                     *
*Entries :     2272 : Total  Size=      37881 bytes  File Size  =      13867 *
*Baskets :        3 : Basket Size=      24576 bytes  Compression=   2.69     *
*.............................................................................*
*Br    4 :DirectionZ : DirectionZ[nLines]/D                                   *
*Entries :     2272 : Total  Size=      37888 bytes  File Size  =      17642 *
*Baskets :        3 : Basket Size=      25088 bytes  Compression=   2.11     *
*.............................................................................*
*Br    5 :DirectionX : DirectionX[nLines]/D                                   *
*Entries :     2272 : Total  Size=      37888 bytes  File Size  =      18831 *
*Baskets :        3 : Basket Size=      25088 bytes  Compression=   1.98     *
*.............................................................................*
*Br    6 :FirstHoughHit : FirstHoughHit[nLines][2]/D                          *
*Entries :     2272 : Total  Size=      65918 bytes  File Size  =      20546 *
*Baskets :        4 : Basket Size=      37376 bytes  Compression=   3.18     *
*.............................................................................*
*Br    7 :LastHoughHit : LastHoughHit[nLines][2]/D                            *
*Entries :     2272 : Total  Size=      65910 bytes  File Size  =      22176 *
*Baskets :        4 : Basket Size=      37376 bytes  Compression=   2.94     *
*.............................................................................*
*Br    8 :FirstHoughPlane : FirstHoughPlane[nLines]/I                         *
*Entries :     2272 : Total  Size=      23956 bytes  File Size  =       7755 *
*Baskets :        3 : Basket Size=      18432 bytes  Compression=   3.01     *
*.............................................................................*
```

- Truth directly from edep-sim or GENIE pass-through
- Reco quantities being added as we go along, e.g. track quality, where does track start/end, track length, etc
- Probably also want to save the actual hits

What's left?

- Small optimisations in track finder from geometry, e.g. making sure I understand module/plane numbering
- Track length estimator from track finder
- Kalman filter gets track length estimator as seed
  - The hiccough here is **always** track propagation routines, notably in inhomogenous magnetic fields with energy loss
  - If you know of any good packages, please let me know
  - For now, proceeding based on ROOT's particle propagator for the event display
- Looking at a proper (TM) event display implementation using Eve in ROOT
- Add variables to output tree as the LAr+TMS groups continue to work on the matching

What's left?

- Should then have a fully-fledged tool to relatively easy whip up the reco for any geometries
- Lots of parameters could be tuned against lots of metrics, but I won't have time for these detailed studies
- Performance? Currently below 0.1s/event for full conversion and reconstruction
  - But have done no profiling whatsoever

Thanks