

QuadVtx vertex reconstruction

larsoft coordination meeting
11 Feb, 2020

Chris Backhouse
University College London

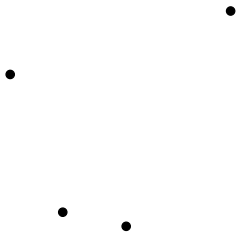
Introduction

- ▶ Independent vertex can be useful as input to other algorithms
- ▶ Presenting a very simple hit-based approach
- ▶ Used DUNE MCC11 FD ν_μ files

`/pnfs/dune/tape.backed/dunepro/mcc11/protodune/mc/full-reconstructed/*/nu_dune10kt_1x2x6*gen.g4_detsim_reco.root`
(despite misleading directory name)

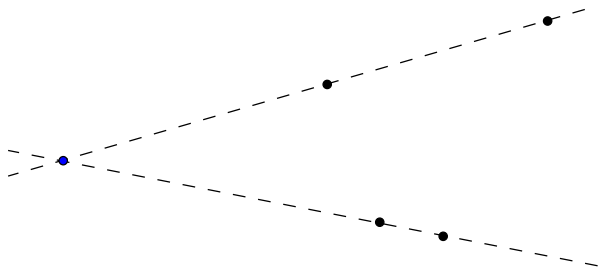
- ▶ Input is all `hitfd RecoHits` in the event (these are disambiguated with a real algorithm)

Algorithm 1



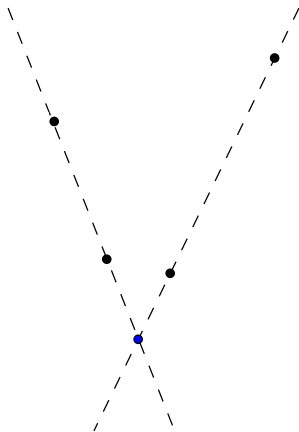
- ▶ Inspired by two-point Hough Transform
- ▶ Two points define a line, four can define the intersection of two lines

Algorithm I



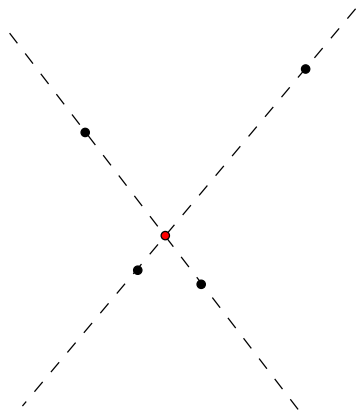
- ▶ Inspired by two-point Hough Transform
- ▶ Two points define a line, four can define the intersection of two lines

Algorithm I



- ▶ Inspired by two-point Hough Transform
- ▶ Two points define a line, four can define the intersection of two lines
- ▶ Permutations give different valid “votes” for the vertex position

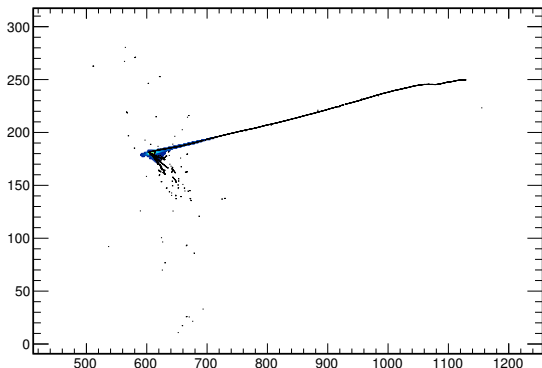
Algorithm I



- ▶ Inspired by two-point Hough Transform
- ▶ Two points define a line, four can define the intersection of two lines
- ▶ Permutations give different valid “votes” for the vertex position
- ▶ Combinations that cause a “track” to straddle the vertex are skipped

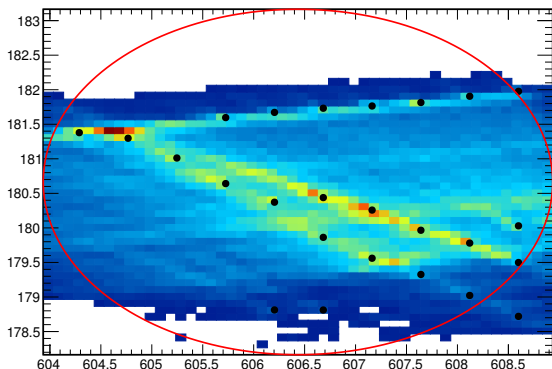
Algorithm II

- ▶ Computationally intractable to try all combinations of four points
- ▶ Restrict to random subset of 10,000,000 quadruplets
- ▶ Fill map in each of the three views (dropped bins $< 5\%$ of peak here)
- ▶ Find 3D point with highest sum when projected into the three maps



Algorithm II

- ▶ Computationally intractable to try all combinations of four points
- ▶ Restrict to random subset of 10,000,000 quadruplets
- ▶ Fill map in each of the three views (dropped bins $<5\%$ of peak here)
- ▶ Find 3D point with highest sum when projected into the three maps
- ▶ Zoom in to mm resolution and repeat



FAQs

Shouldn't you do error propagation?

- ▶ Could take a ~ 5 mm error on each point, compute the error on the implied vertex, and fill a gaussian into the accumulation space
- ▶ That's a lot more computation time per quadruplet
- ▶ If the points really do jitter around I should get the same aggregate smearing over my large ensemble

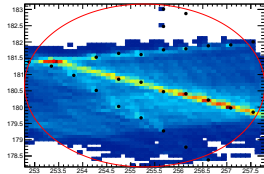
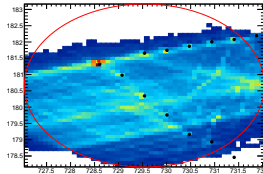
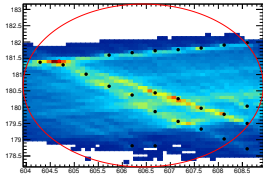
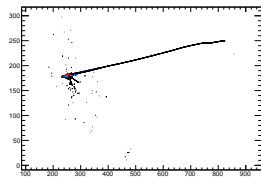
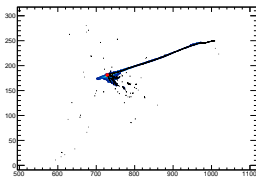
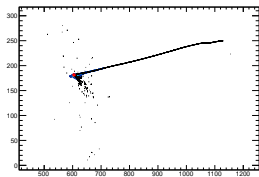
Shouldn't you weight by the energy of the hits?

- ▶ Has a nice property that splitting a hit into two smaller hits in the same place leaves the map unchanged. Better way to treat showers?
- ▶ I tried this and it didn't work so well in practice, not sure why

Some quadruplet topologies are unlikely to point to a vertex

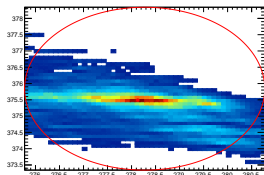
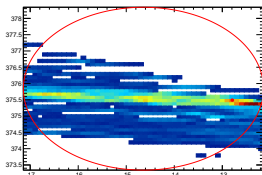
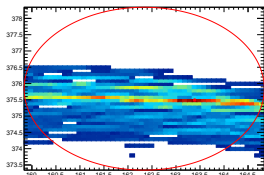
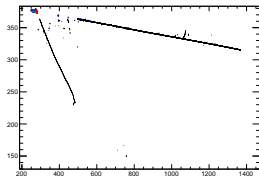
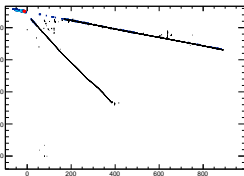
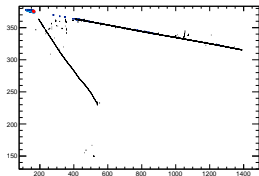
- ▶ I didn't want to introduce too much model dependence
- ▶ Open to exploring simple cuts for this if they help

Examples



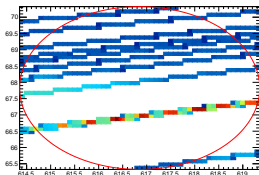
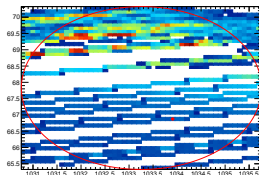
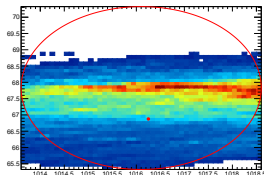
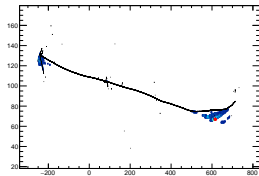
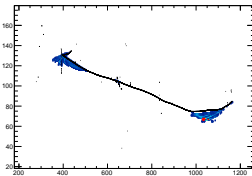
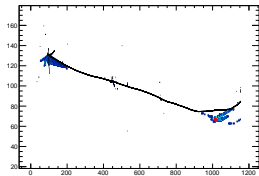
- ▶ Same event as earlier, but in all three views

Examples



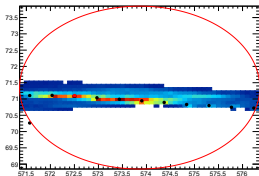
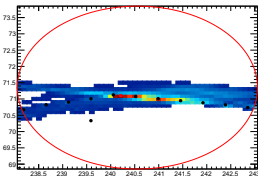
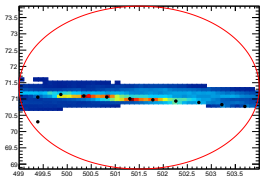
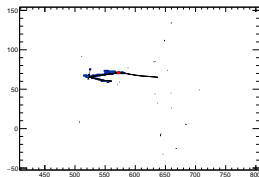
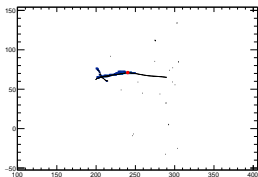
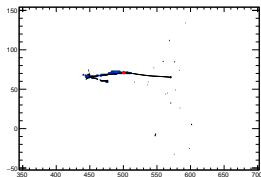
- ▶ Vertex separated from tracks is well-reconstructed

Examples



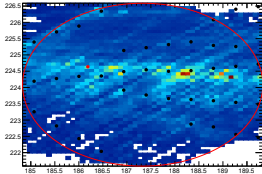
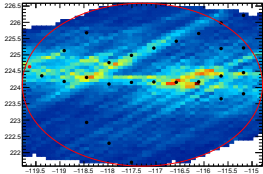
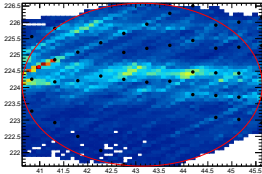
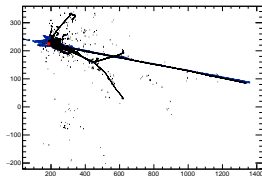
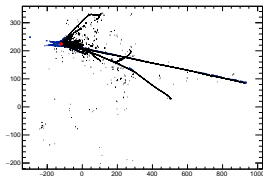
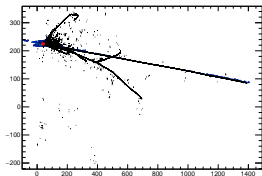
- ▶ Longer track segments point here than to the true vertex :(

Examples



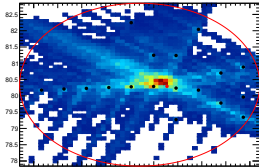
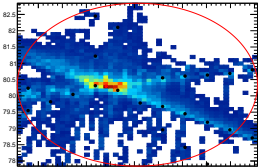
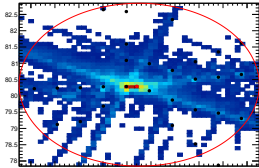
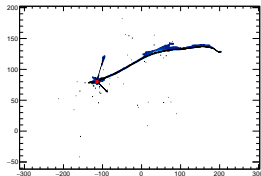
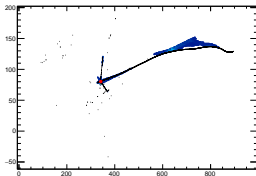
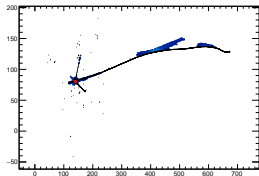
- Find a secondary vertex. I now require the vertex to be in the upstream 1/4 of the event. This is the only directional bias anywhere

Examples



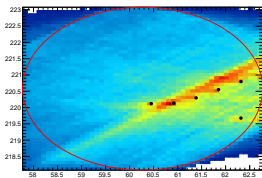
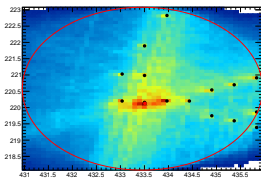
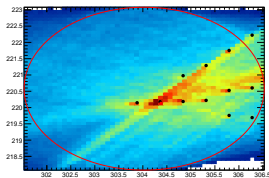
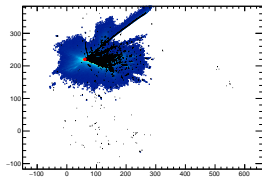
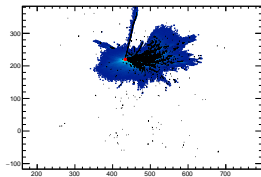
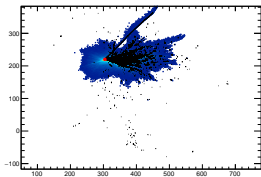
- More complex event – almost right

Examples



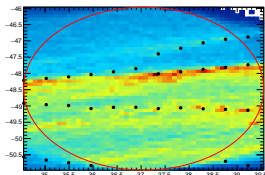
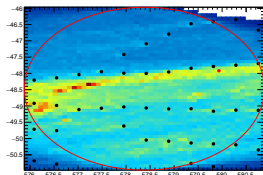
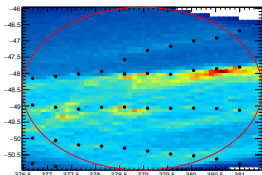
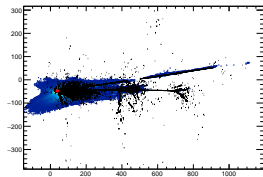
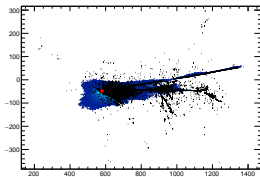
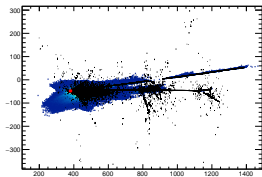
- More complex event – good, avoids secondary vertex

Examples



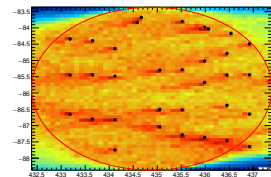
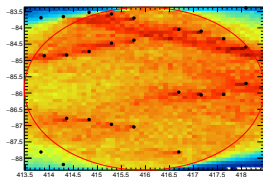
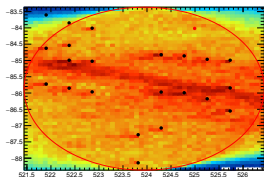
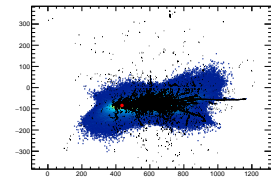
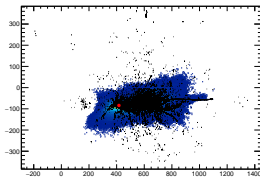
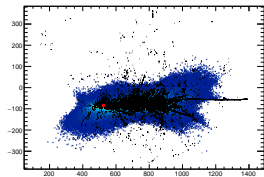
- Very busy events can also work well

Examples



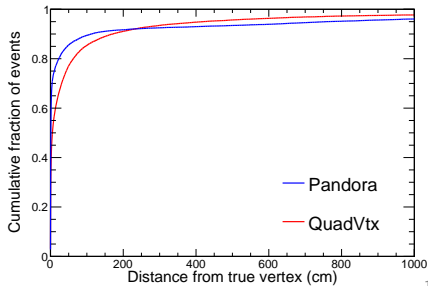
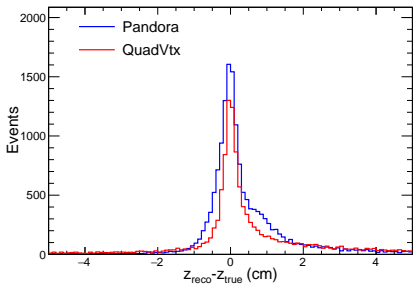
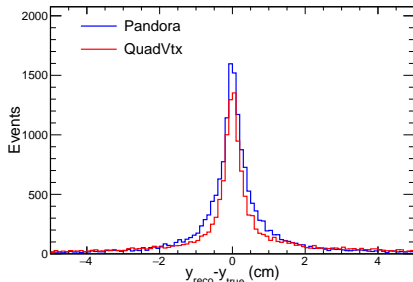
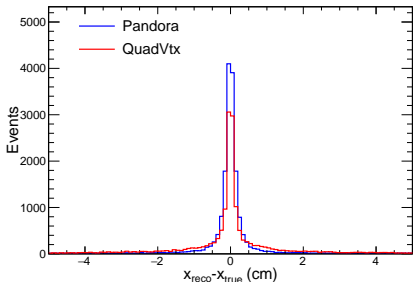
- ▶ Very busy events can also work well

Examples

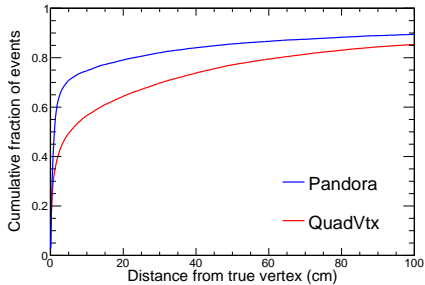
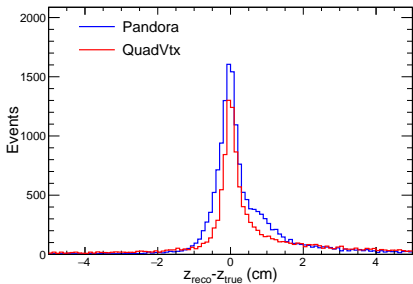
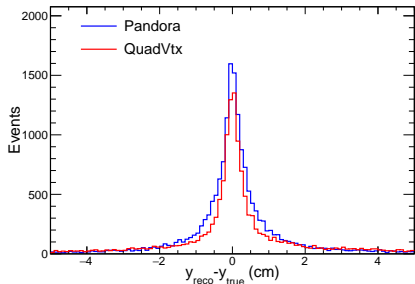
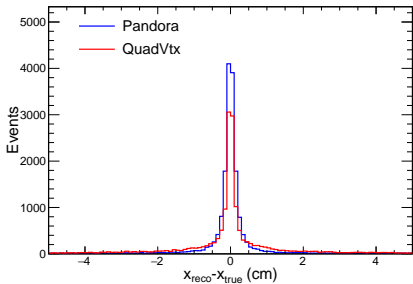


- ▶ Very busy events can also work well – this one is a bit downstream

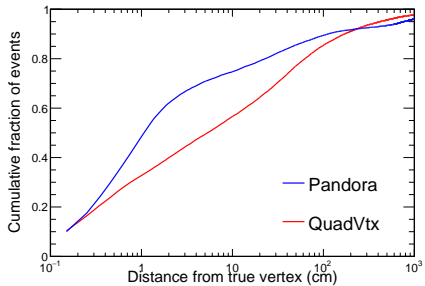
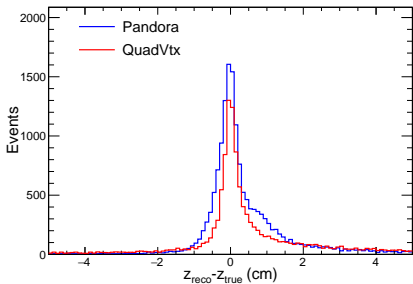
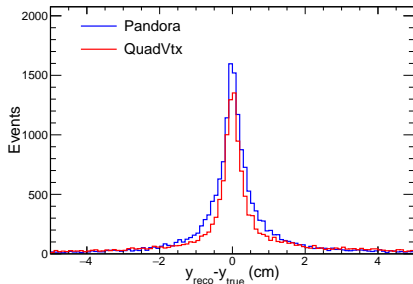
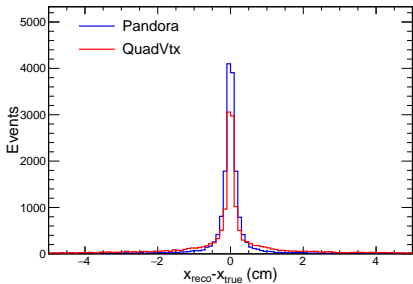
Performance



Performance



Performance



Conclusions

- ▶ Pandora does a little better
- ▶ But QuadVtx has fewer errors $>2m\dots$

- ▶ Achieved with a very simple and fast procedure

- ▶ Pull request <https://github.com/LArSoft/larreco/pull/4>