Setting Trajectory Point Index in Calorimetry

Justin Hugon

Louisiana State University

LarSoft Coordination Meeting

2019-03-26

Introduction

- **calo::Calorimetry module takes hit charge values in ADC and converts them to dQ/dx and dE/dx**
- **In the process it discards some hits for various reasons**
- **It then saves dQ/dx, dE/dx, pitch, x, y, z for each hit into vectors in anab::Calorimetry**
- **Can’t match back these values to hits because don’t know which hits were discarded**

Justin Hugon
Louisiana State University

Trajectory Point Index

- **anab::Calorimetry has a place for a vector of tpIndex to resolve the ambiguity**
- **calo::Calorimetry just needs to be modified to store the tpIndex**
- **I did this and pushed the feature/jhugon_caloTPIndex branch**

Justin Hugon
Louisiana State University

Which Index?

My understanding is that the hit to trajectory point mapping is trivially linear, so I'm mapping entries in recob::Calorimetry to recob::Hits

When recob::TrackHitMeta is absent:

- **For a track: tpIndex is the index of the hit in the track's art::FindManyP<recob::Hit> entry**
- **So the indexing of hits is simple**

When recob::TrackHitMeta is present:

- **For a track: tpIndex is the index returned from a track hit's recob::TrackHitMeta::Index()**

Justin Hugon
Louisiana State University

Conclusions

- I modified calo::Calorimetry to fill the anab::Calorimetry tpIndex in branch feature/jhugon_caloTPIndex
- It's now possible to identify the wire and channel a calorimetry entry corresponds to
  - Can check the calibration/response of each channel using Calorimetry
  - Easier to tell in e.g. a pion cross-section if a track missed a wire
- It would be great for this to be in LArSoft in time for the next ProtoDUNE production campaign

Justin Hugon
Louisiana State University