Proposed Updates To LArSoft: LArEventDisplay

LArReco

Tracy Usher

LArSoft Coordination Meeting

January 15, 2019

Executive Summary

- Proposed update to LArEventDisplay covers three general areas
  - Updates to support visualization of 3D pattern recognition algorithms
    - Most updates in this category centering around the 3D event display
  - Some early attempts to simplify the event display
    - Pushing pieces of the code out to sub-algorithms
  - A few other simple updates
- Proposed update to LArReco: Primarily Cluster3D code
  - Recent updates allow it to work with general LArTPC geometries
    - MicroBooNE, ICARUS and ProtoDUNE now demonstrated
  - Also can accept Space Point input from alternate sources
    - e.g. SpacePointSolver

Motivation

- Have been slowly developing the 3D clustering over a long period of time
- Meanwhile, the SLAC group has recently grown dramatically
  - In particular have one member working on ProtoDUNE, now interested in doing some data analysis
- Idea is to use the 3D clustering as an “event slicer” to associate 2D hits to common pieces of a given event
  - Primary target will be the Projection Matching Algorithm
- To facilitate this program need to move the Cluster3D updates off a feature branch and into develop to get tagged
- Have been maintaining a feature branch for the event display to support the above work, need/want to merge that as well

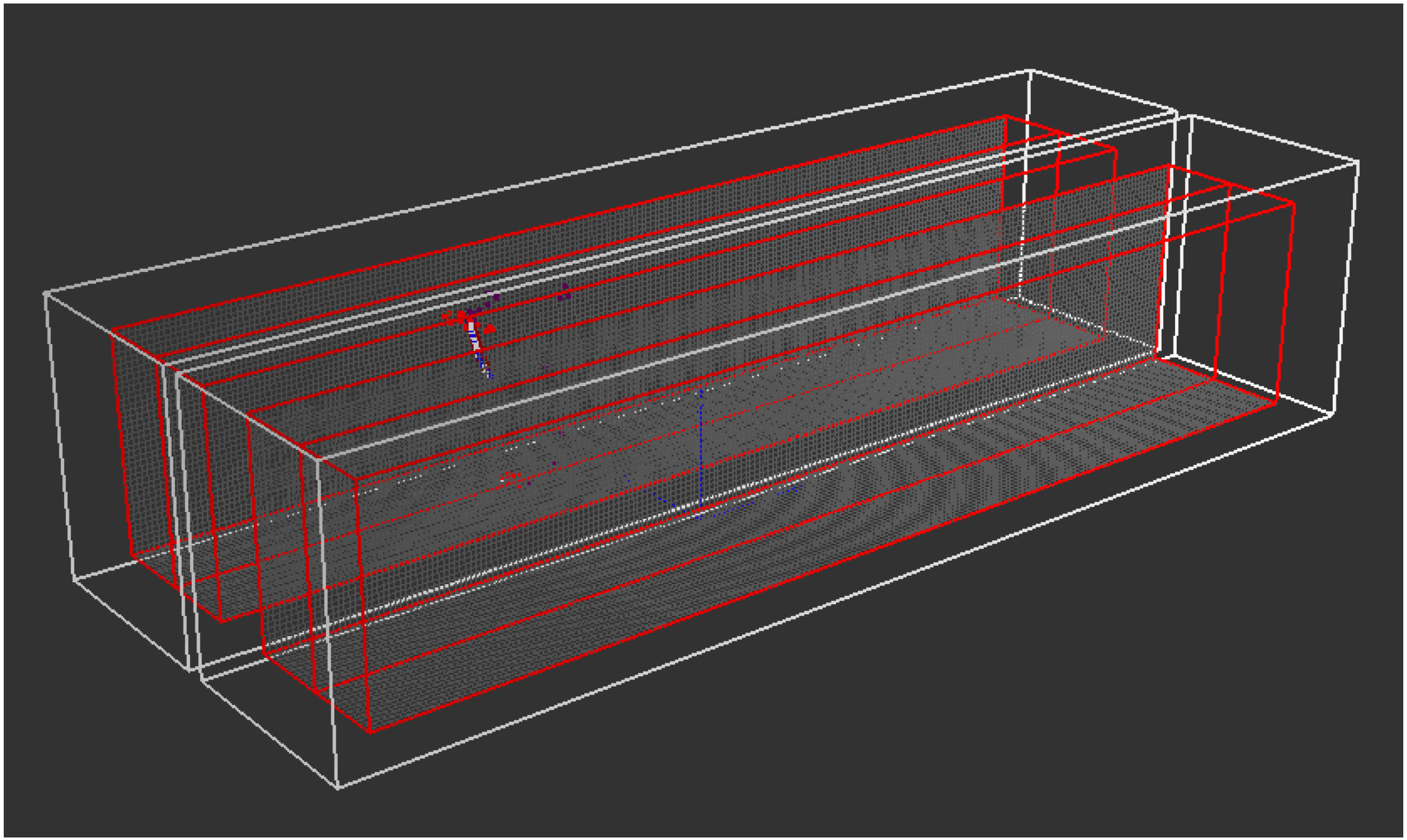
Event Display “Simplification”

- Most generally agree that the existing event display code can be challenging to understand when one wants to make an update
  - Difficult to understand how to make “simple” changes
- LArSoft v07 series appeared to present path to simplification
  - Art tools could be used to interchange various components
- Some time ago challenged myself that when making updates to the event display would attempt to use tools…
  - Mostly for the 3D display
  - Though have rethought this in the light of a new display coming soon!
- Will try to describe what was done so far
  - Will go a little bit out of order…

3D Geometry Drawers

- Experiments can implement custom 3D event display drawers in a separate tool
  - Generic tool will faithfully draw the outline of the geometry
  - Custom tool can include custom fhicl parameters if wanted
    - e.g. can draw “bad channels” based on channel status DB
- Currently four drawing tools available:
  - “Default” drawer, ICARUS, MicroBooNE and ProtoDUNE
  - See the “ExptDrawers” subfolder of “EventDisplay” folder
- To select a custom drawer:
  - services.EvdLayoutOptions.Experiment3DDrawer: @local::protodune_drawer

Example: ICARUS

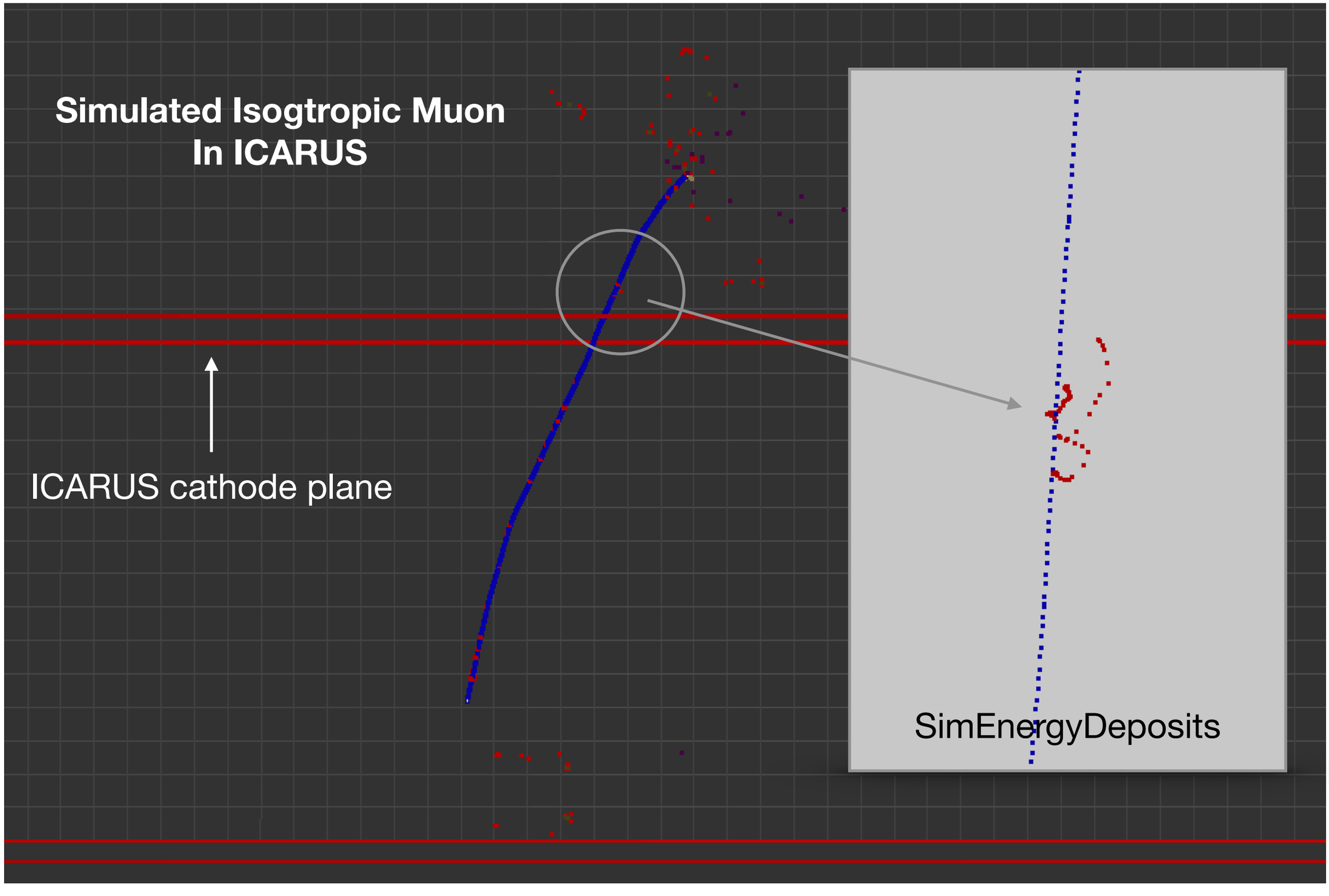


Simulation Truth Information

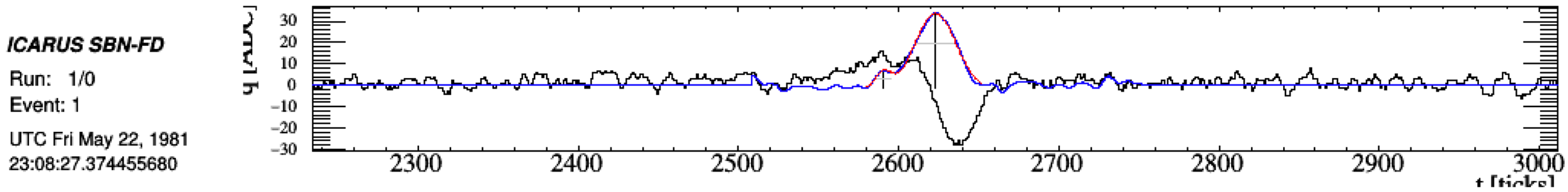
- Recently a new 3D truth data object available: SimEnergyDeposit
- Extracted the 3D drawing section from “SimulationDrawer” and created two 3D simulation drawing tools
  - DrawSimEnergyDeposit3D_tool - based on SimEnergyDeposit objects
  - DrawLArVoxel3D_tool - based on SimChannel objects
  - See “SimDrawers” subfolder of “EventDisplay”
- Activating one or the other (or both):
  - services.SimulationDrawingOptions.ShowSimChannelInfo: true
  - services.SimulationDrawingOptions.ShowSimEnergyInfo: true

Example 3D Truth Info

SimEnergyDeposits



Waveform Drawers



- Module “TQPad” draws the waveform plus hits part of the display
  - Similar but separate code to draw single phase vs dual phase reconstructed hits makes module somewhat cumbersome
- Started the task of breaking the drawing of hits into art tools
  - DrawGausHits_tool
    - Allows drawing of hits from multiple sources in different colors
    - Draws hit center plus width, allows a “floating” baseline, etc.
  - DrawSkewHits_tool
    - Intended to be for dual phase but not yet implemented since I don’t have easy way to test
    - Have left this part of the display code in TQPad
- See the “wfHitDrawers” subfolder

Cluster3D - Brief Reminder

- Cluster3D is a package that ultimately will attempt to do basic pattern recognition in 3D using Space Points
- Currently, the steps are:
  - Form Space Points from allowed combinations of 2D hits
  - Cluster adjacent 3D Space Points to form initial clusters
  - Merge clusters that “obviously” belong together
  - Do basic shape analysis
    - Principal Components Analysis to get primary axis of cluster and its spread
    - Build 2D convex hull based on projection to plane of maximum spread of PCA
  - After this can begin to do path finding, etc.
    - Before this stage the output is very effectively “Sliced” and can be sent to one of the 2D pattern recognition algorithms which can run more efficiently on the reduced sets of hits

For more info see this more detailed presentation at the Calibration/Reconstruction Workshop

Major Updates Since Last Release

- Attempt to be 3D Space Point generic
  - Can build own Space Points from internal builder or accept Space Points from outside builder
    - Currently only SpacePointSolver but hope eventually WireCell
- Space Point Quality Metric
  - Simplified the clustering and merging steps
    - Tight requirement to put points in clusters
    - Merging algorithm prefers under merging to over merging
- Significant effort on path finding
  - Recursive algorithm based on breaking at convex hull defect points
- Initial implementation of candidate vertex identification

Other Updates

- GausHitFinder
  - Main module - removed unused fhicl variable definitions
  - Underlying tools - update to the “morphological filter” which is used by ICARUS but not by any other experiment (yet).
- SpacePointSolver
  - Built a tool interface to “read hits” in order to handle the special case of the ICARUS geometry
    - Standard tool which handles all the cases SpacePointSolver was originally built for
    - ICARUS specific tool to handle the horizontal induction wires
  - Should be no differences in the operation of the module other than the tool usage
    - Immortal words: “it works for me”

Back to Event Display

- In order to visualize the Cluster3D work, including new features in the 3D event display:
  - Option to draw Space Points in a “heat map” scheme based on the quality metric - “better” hits more red, “worse” hits more blue
  - Principal Components Axes
  - “Edges” to draw the convex hull
  - Associated candidate vertices
  - etc.
- Some examples to follow

Simulated Muon in ICARUS

Principal Components Axes

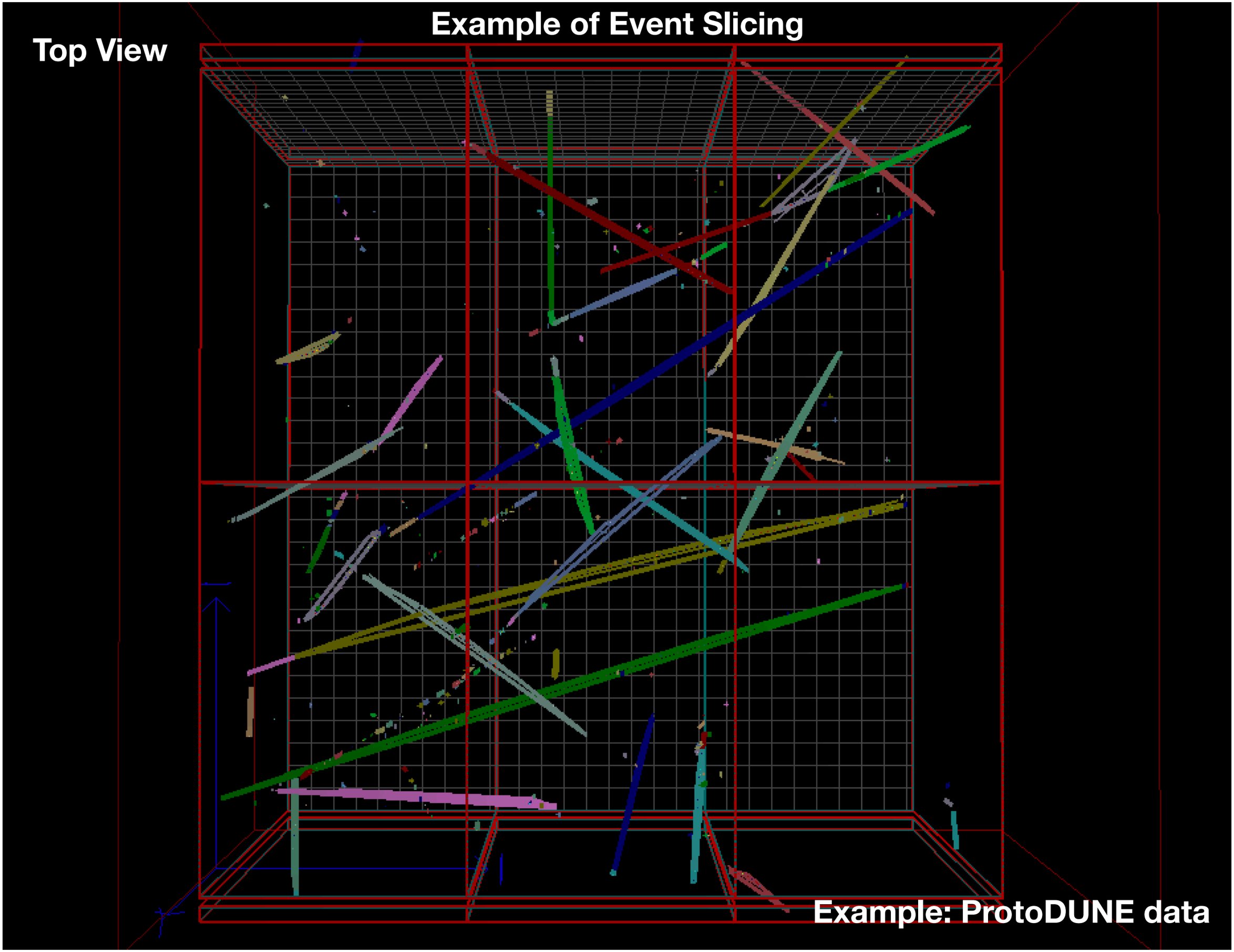
(Rotated to look down on the plane of maximum spread)

Large yellow balls are kink points on the hull - Taken as candidate vertex points

Project points to plane of maximum spread by PCA, Then compute the convex hull

Example of Event Slicing

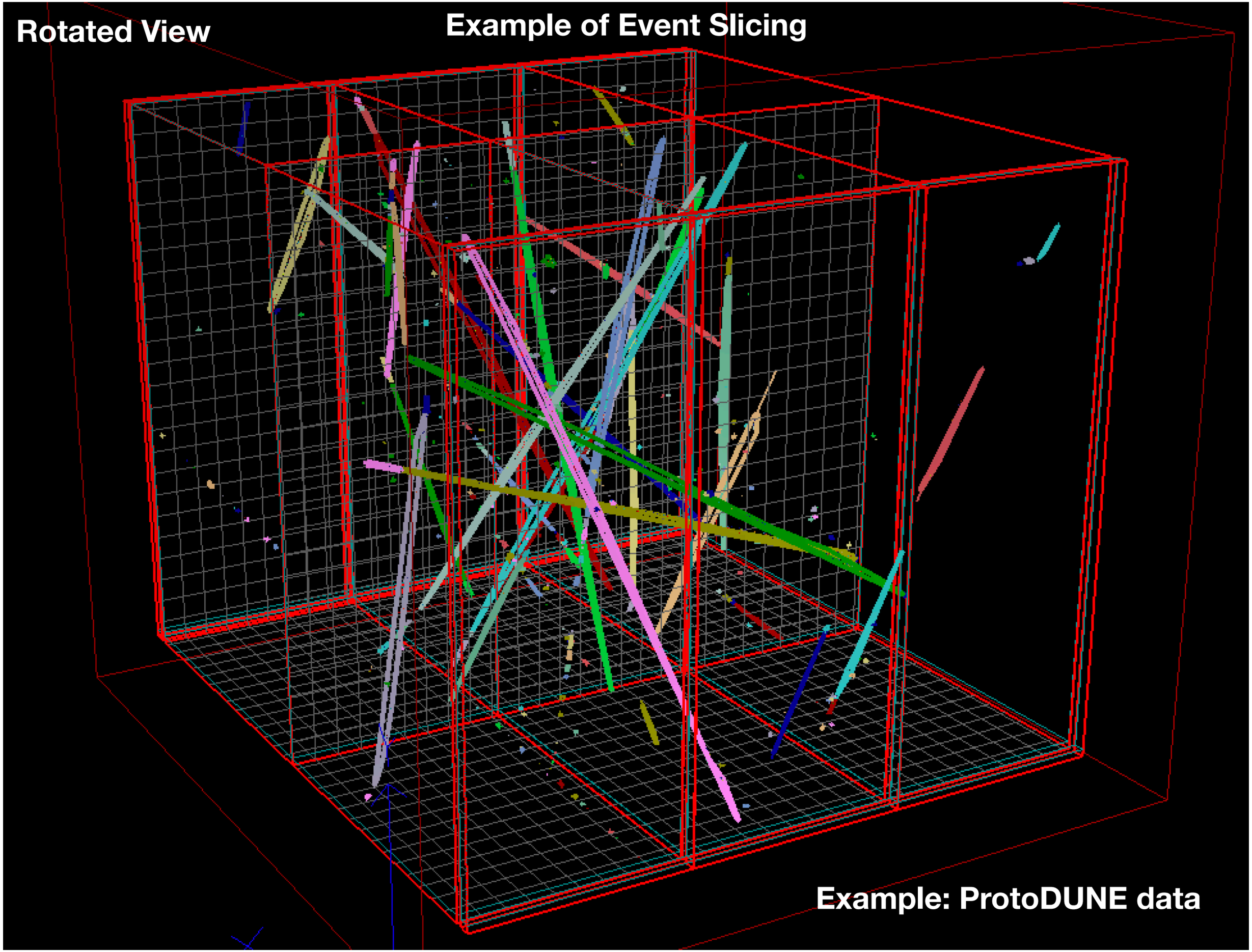
Top View



Example: ProtoDUNE data

Rotated View

Example of Event Slicing



Proposed Updates

- Merge larreco feature branch:
  - feature/usher_cluster3dspacepoints
- Merge lareventdisplay feature branch:
  - feature/usher_edgedrawingoptions
- Both branches have been continually rebased to the develop branch
  - Last merge was to larsoft v08_03_00
- Both will bring in useful new features
  - Cluster3D which can be used as a “slicing” algorithm
  - Event display with new 3D capabilities plus, hopefully, some organization improvements