Stony Brook University

T2K

The SuperFGD Reconstruction

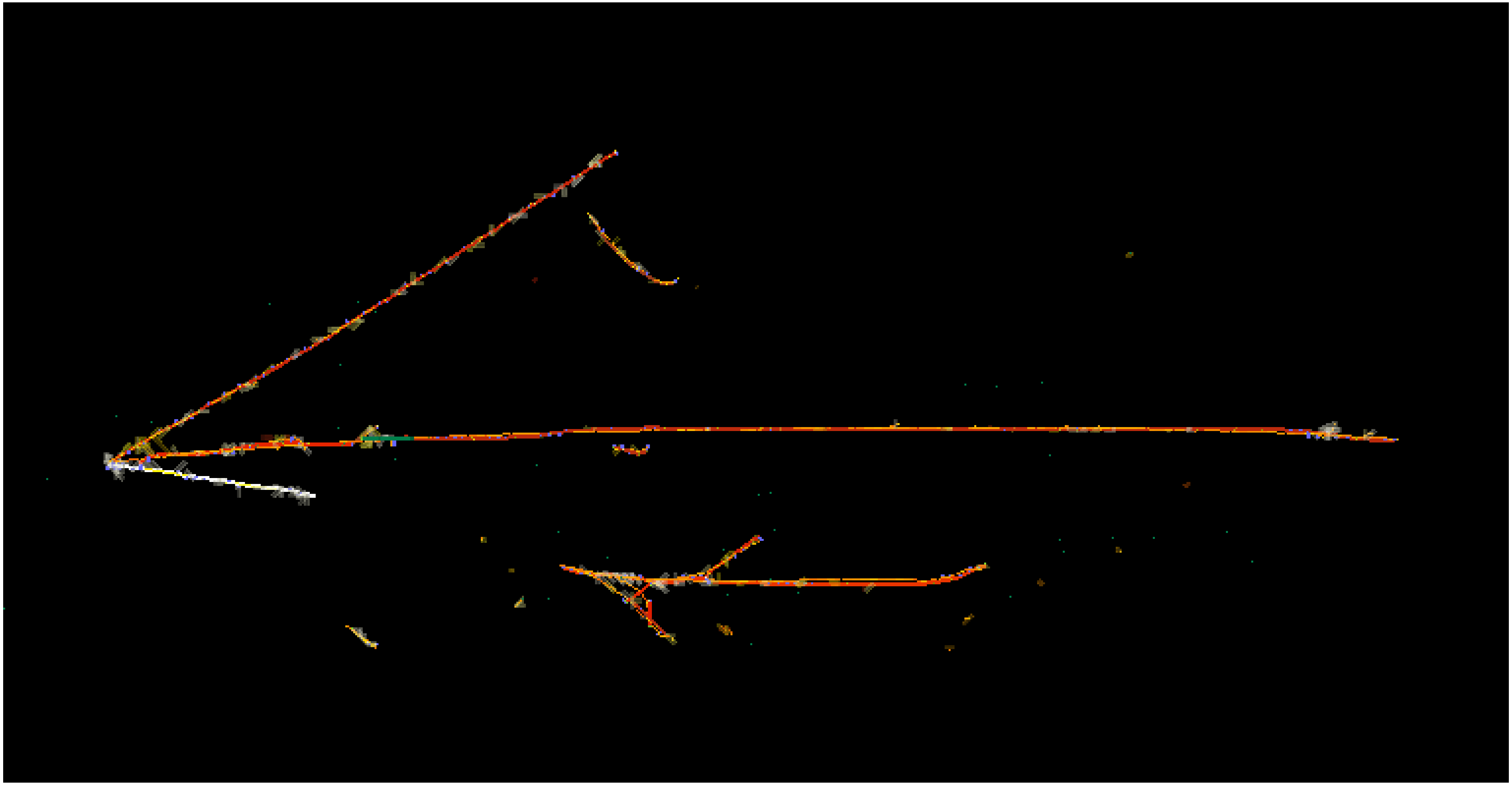
Overview of What is Being Done for the Super FGD

Clark McGrew
Stony Brook Univ.
(almost, but not quite, for the SFG Reconstruction Group)

- ND280 Reconstruction Basics
  - Phase 1: Sergey Martynenko’s cube recon
  - Phase 2: Preliminary superFGD reconstruction (moving to phase 3)
  - Phase 3: Trials with larger samples of neutrino interactions
- Plans for 3DST

True Energy Deposit for

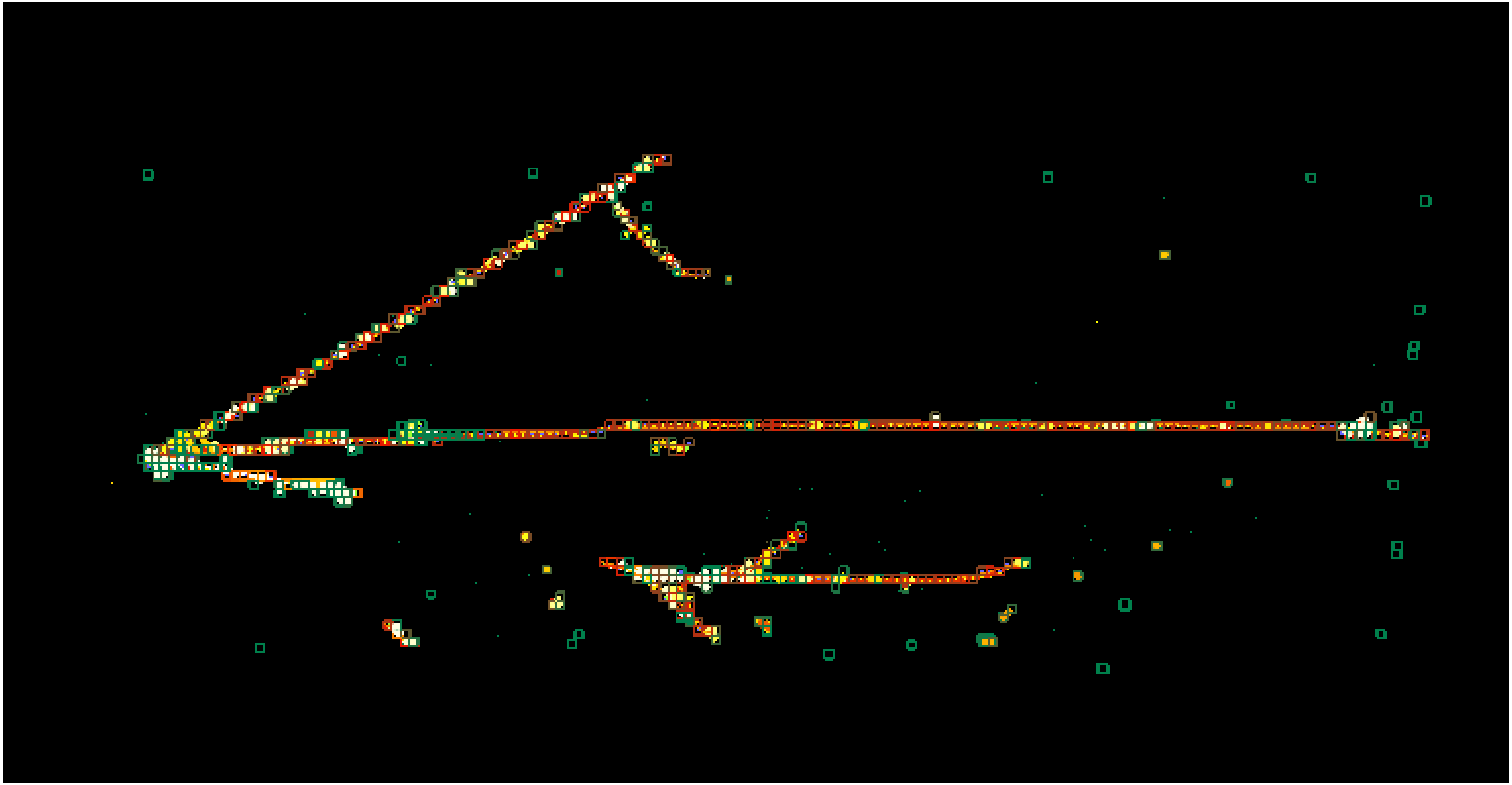
700 MeV μ^- , 300 MeV π^+, 200 MeV p, 400 MeV π^0, 50 MeV p and 60 MeV p (Kinetic Energy)



Typical reconstruction “development” event. Chosen because topology tests the reconstruction

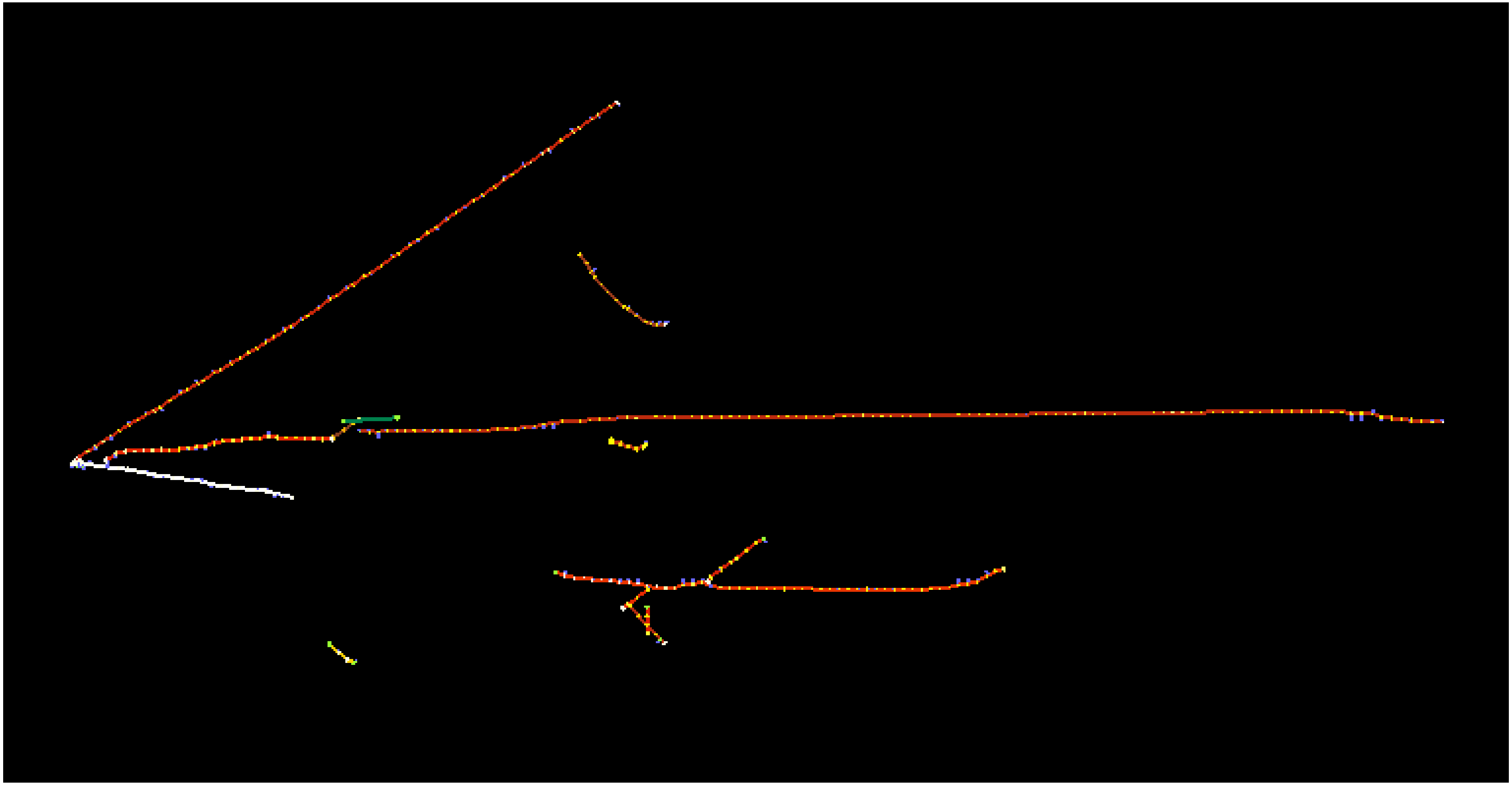
Reconstructed 3D Hits for

700 MeV μ^-, 300 MeV π^+, 200 MeV p, 400 MeV π^0, 50 MeV p and 60 MeV p (Kinetic Energy)



Reconstructed Tracks for

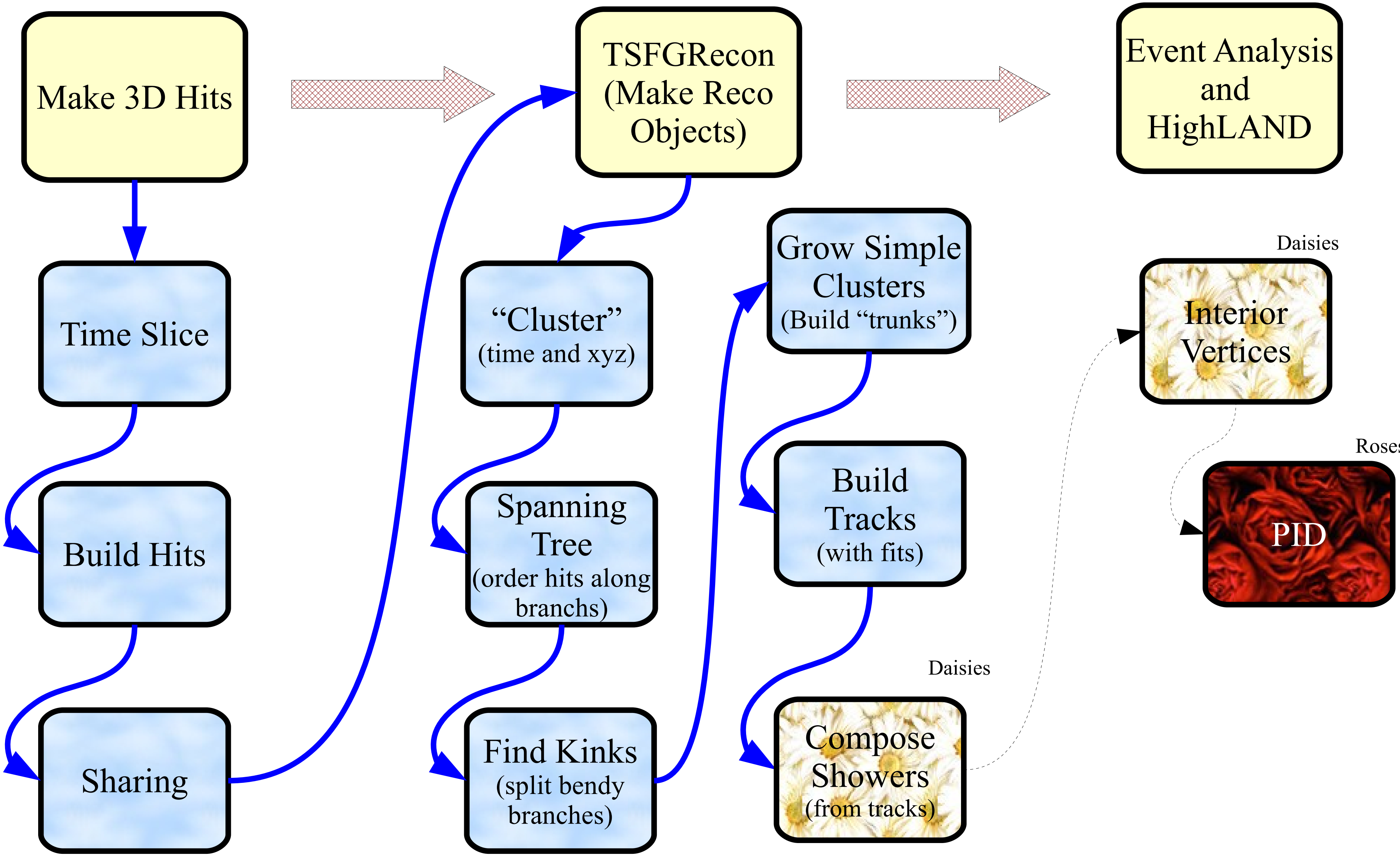
700 MeV μ^- , 300 MeV π^+, 200 MeV p, 400 MeV π^0, 50 MeV p and 60 MeV p
(Kinetic Energy)



The SuperFGD ND280 Chain

- The SuperFGD Reconstruction is developed inside of the T2K ND280 infrastructure
- Key components for today
  - ND280MC (the intellectual ancestor of edep-sim)
    - The detailed SFG (plus the rest of the upgrade) geometry is defined.
    - Geometry is will be updated with as-built drawings.
  - ElecSim
    - The SFG is simulated with "all of the important features"
      - Cube, and fiber response
      - Light transport and MPPC response
      - Electronics simulation including thresholds, resolutions and digitzation
    - Tuned to the beam test MIPS and protons
  - Reconstruction
    - Hit formation, pattern recognition, track fitting

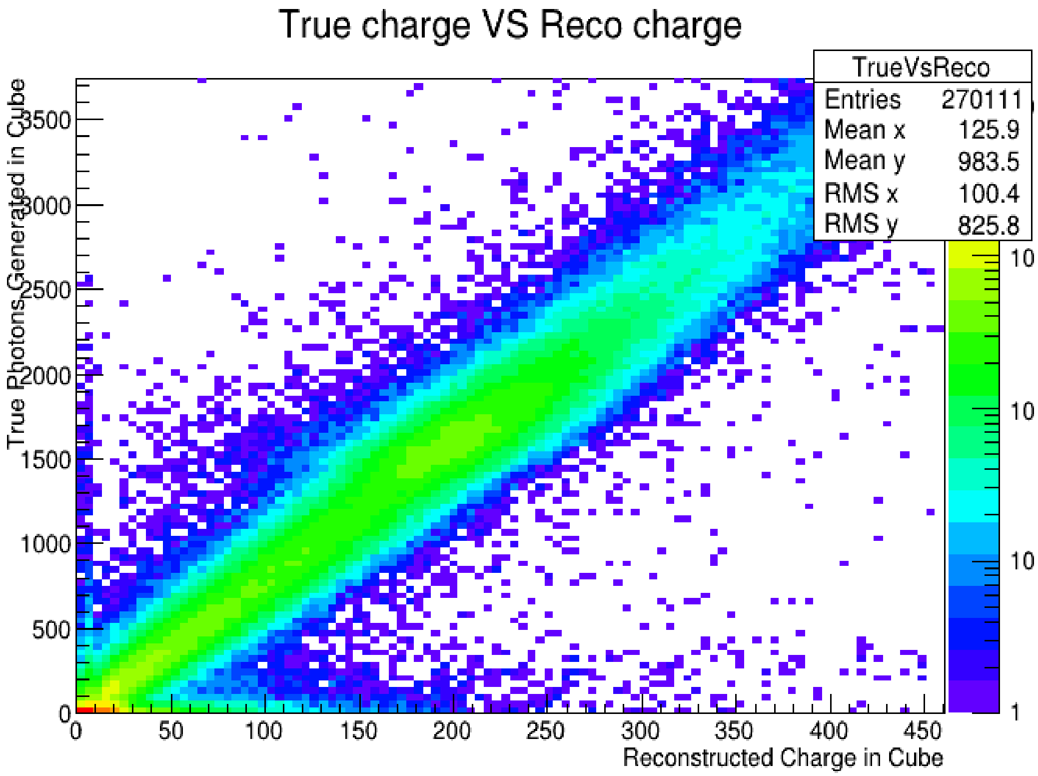
The Flow of the SFG Reconstruction

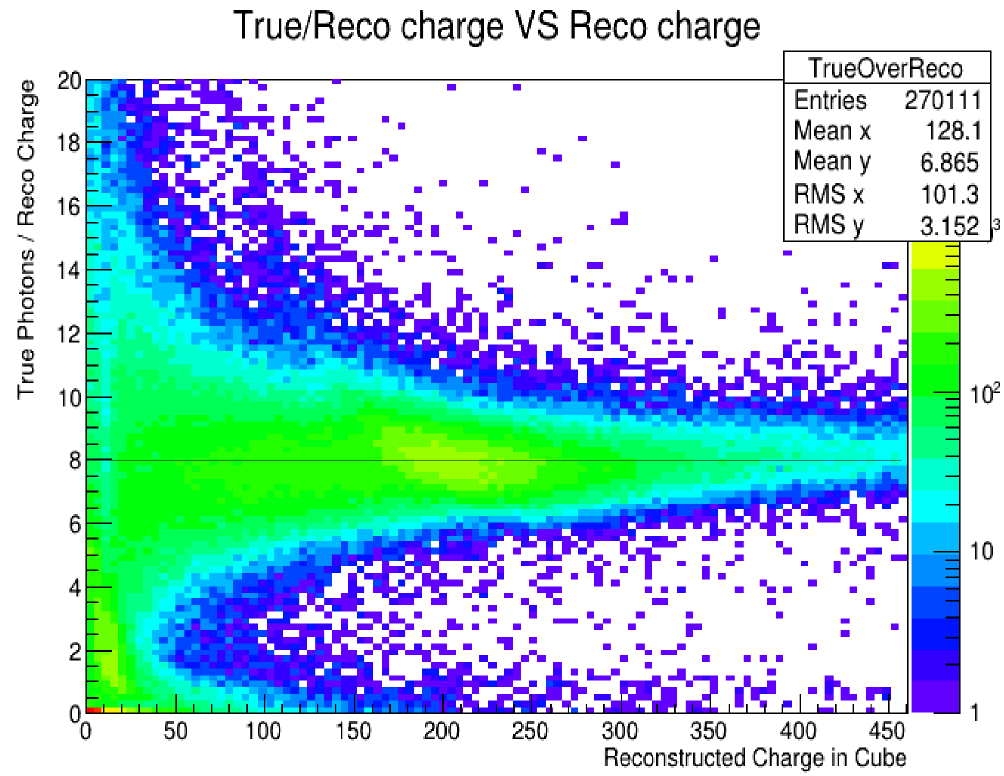


Charge Sharing

- This is a maximum likelihood fit to determine the expected charge generated in each cube
- The likelihood definition
  - Physics Factors
    - Sum of the charges from all cubes on a fiber, corrected for attenuation, should equal the measured MPPC charge
      - This is calculated assuming a Poisson probability for each measurement
    - Negative charges from a cube are forbidden.
  - Bayesian Factors (to break degeneracies)
    - The entropy of the set of cube charges is maximized.
      - This very slightly prefers that the cubes all have the same charge
      - The "entropy" is based on "$\sum Q \ln Q$" (but, approximated for speed).
    - This is a "maximum entropy criteria" similar to what is often used in image processing to guarantee the likelihood is non-degenerate.
      - Without this criteria, we can have cases where there are more cubes than fibers giving a degenerate likelihood.

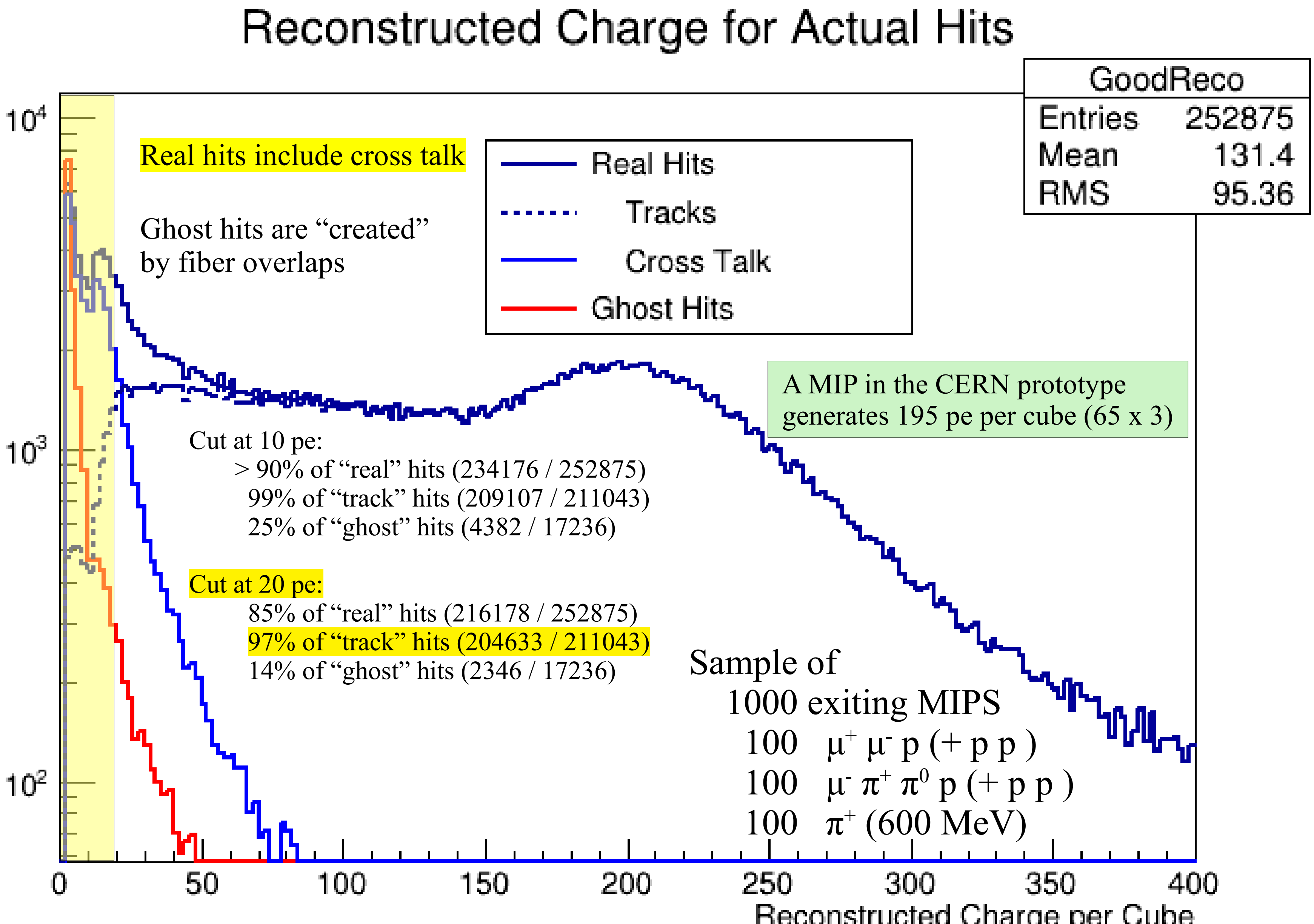
True vs Reconstructed Charge





- Truth: The number of photons generated in the fibers inside each cube
  - Half are lost because the fiber is not mirrored.
  - PDE is 25%, and MPPC starts to saturate at high light levels.
- Reconstructed: The reconstructed number of photoelectrons generated in each cube. It's approximately the sum of the measured photoelectrons from each fiber, corrected for attenuation.

Reconstructed Charge

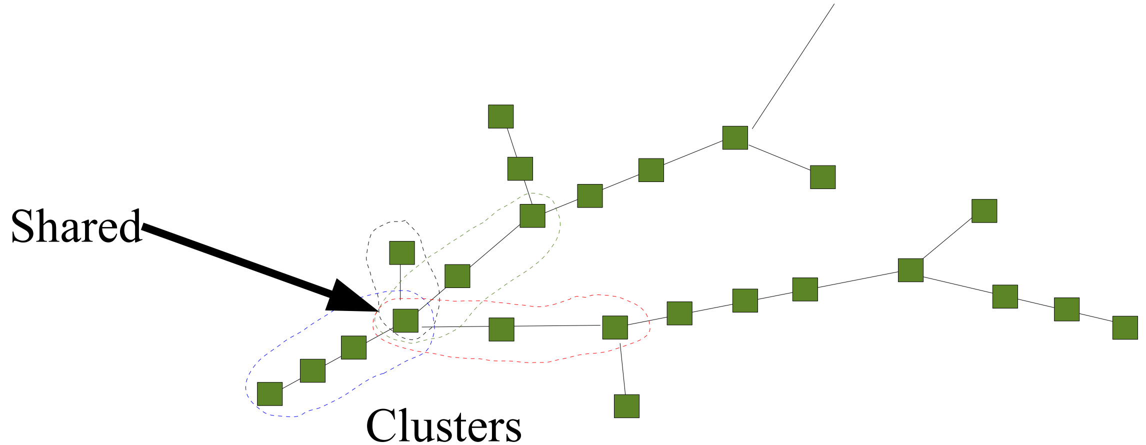


Pattern Recognition (Charged Particles)

- Pattern recognition is applied to each group simply connected hits
  - i.e. The result of DBScan and TimeSlice after charge sharing
- Steps
  - Order the hits using a minimal spanning tree
    - Uniquely defines the neighbors of each hit
  - Split hits in the chains (an ordered set of hits)
    - Split at branches
    - Split at kinks
  - Turn long chains of hits into track segments
  - Combine track segments into tracks
    - Fitting is done based on a 400 MeV/c muon hypothesis
    - Fitting accommodates "kinks" in tracks
  - Combine candidate cross-talk hits with tracks (not available yet)
- Resulting collection of tracks and clusters are ready to be analyzed by global reconstruction, or used in a local particle identification.
  - Tracks include dEdX information as well as total energy deposition
  - Clusters include total energy deposition

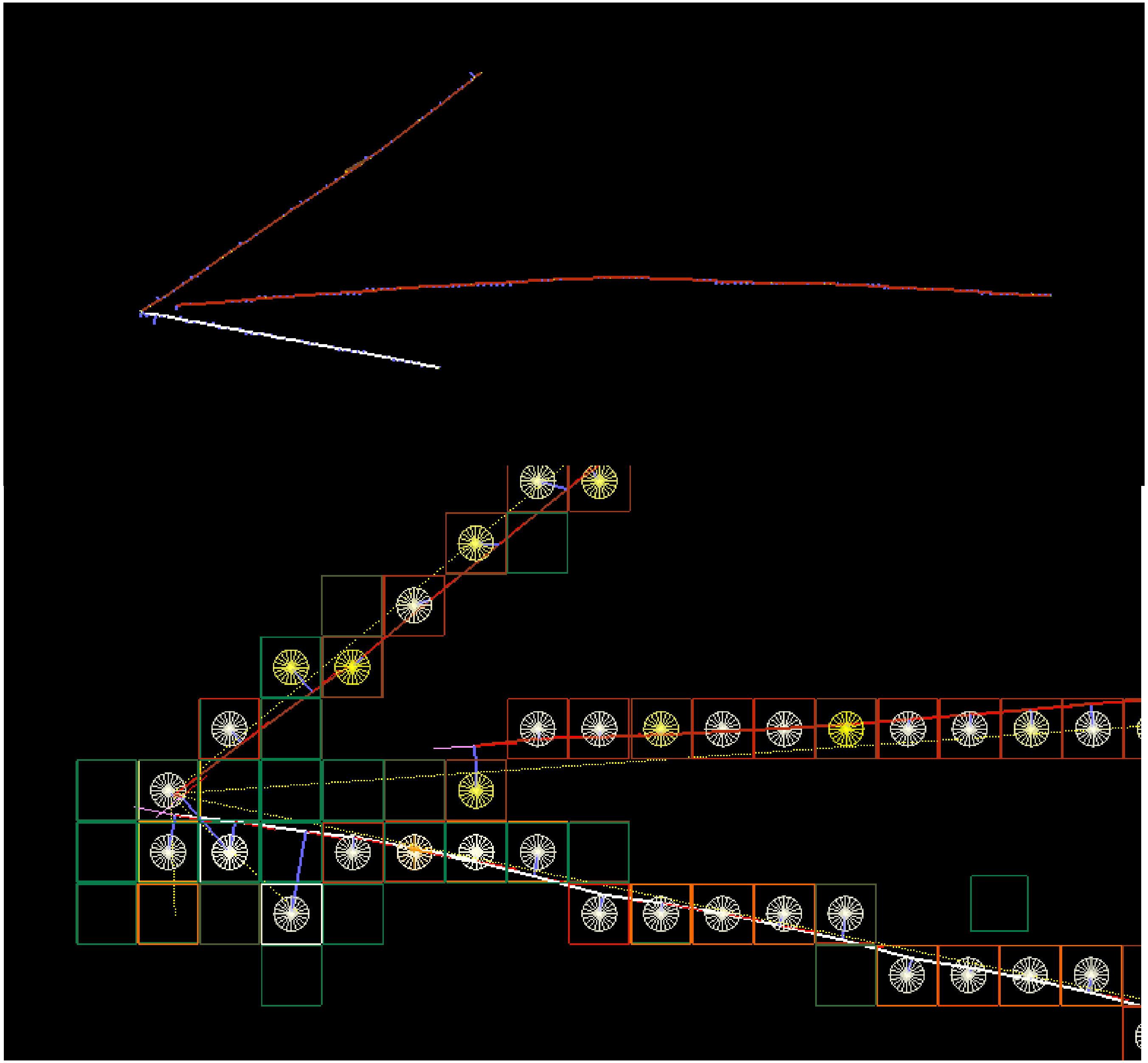
Minimal Spanning Tree Clustering

- The order of the hits in the cluster is significant
- The tree is built from branches, and each branch is a sequence of hits
  - Each cluster represents one branch
  - The hits in the cluster are ordered from “first” to “last”
    - Choice between which is “first” and which is “last” is arbitrary
      - Upstream is “first”, downstream is “last” (mostly...)
- The “first” and “last” hits in a cluster are usually shared with another cluster
  - This is how the connections between the branches has been saved



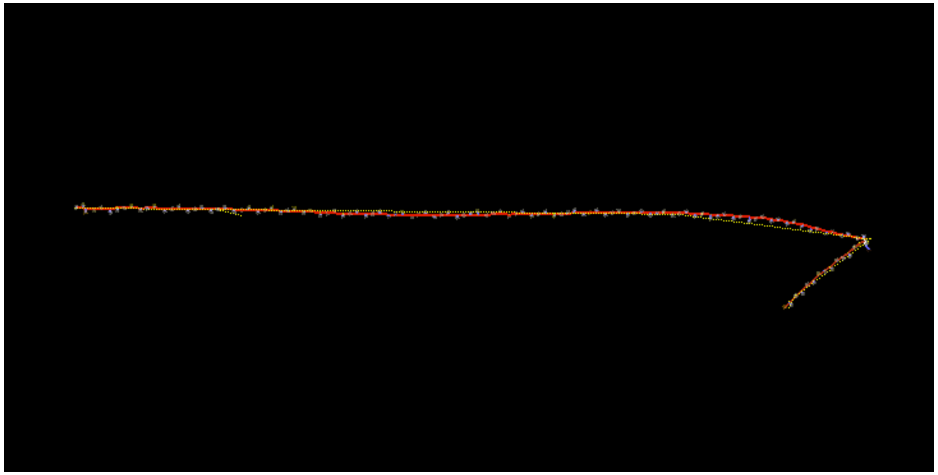
Example Reconstruction Result

- Kinetic Energy
  - mu- : 700 MeV
  - mu+ : 300 MeV
  - p : 300 MeV
  - p : 50 MeV
  - p : 60 MeV
- Top shows just tracks
  - Color set by reconstructed dEdX
- Bottom shows hits and near the vertex
- Clusters not shown

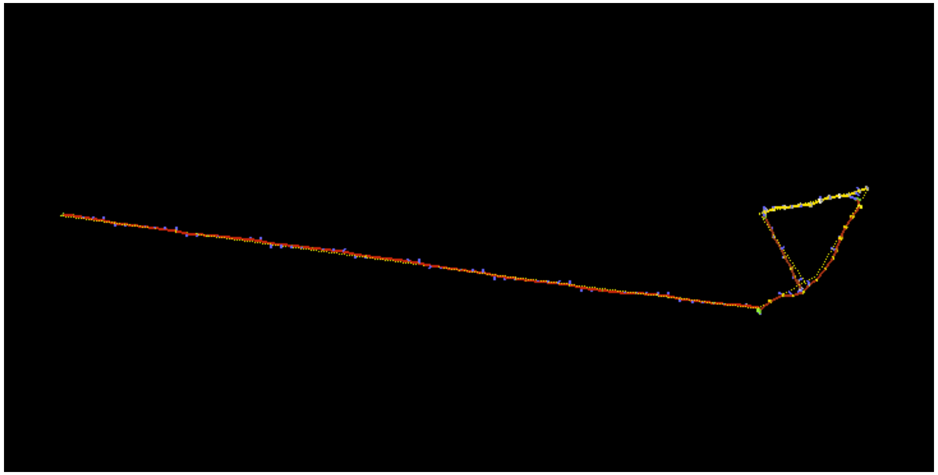


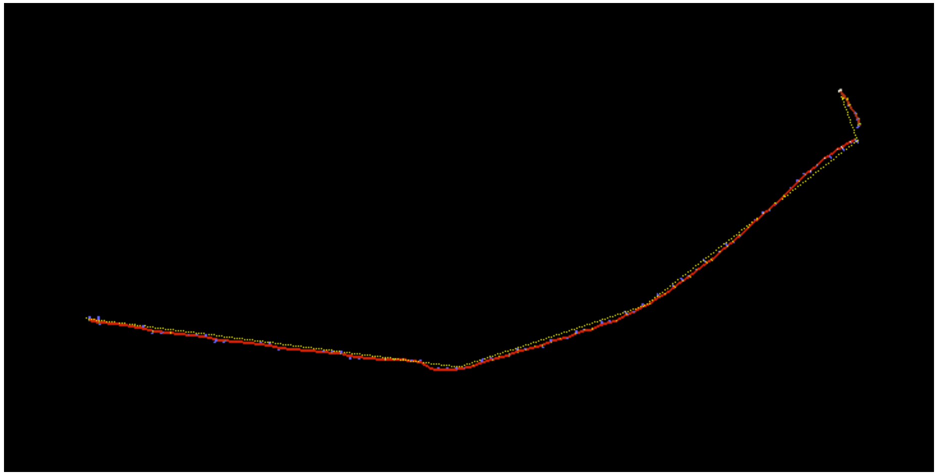
First Four 640 MeV π^+ Particle Gun Events as Reconstructed

Decay at rest to muon (electron is visible)

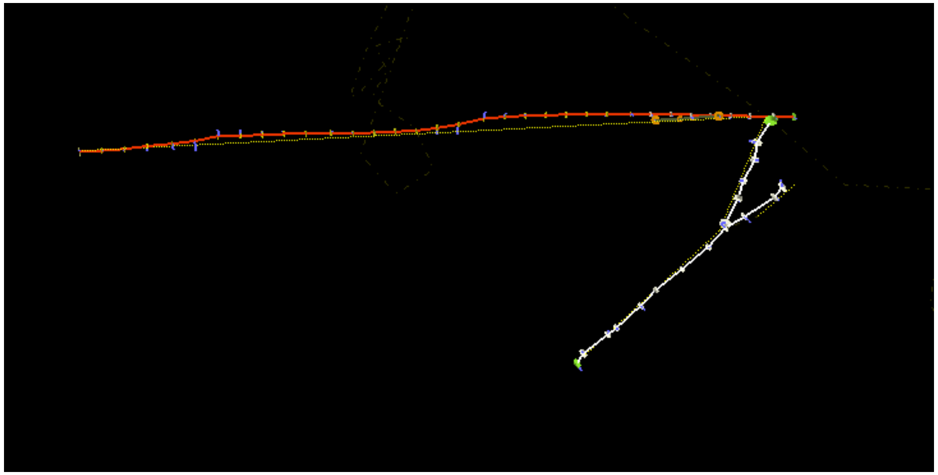


Hard secondary scatter followed by decay at rest



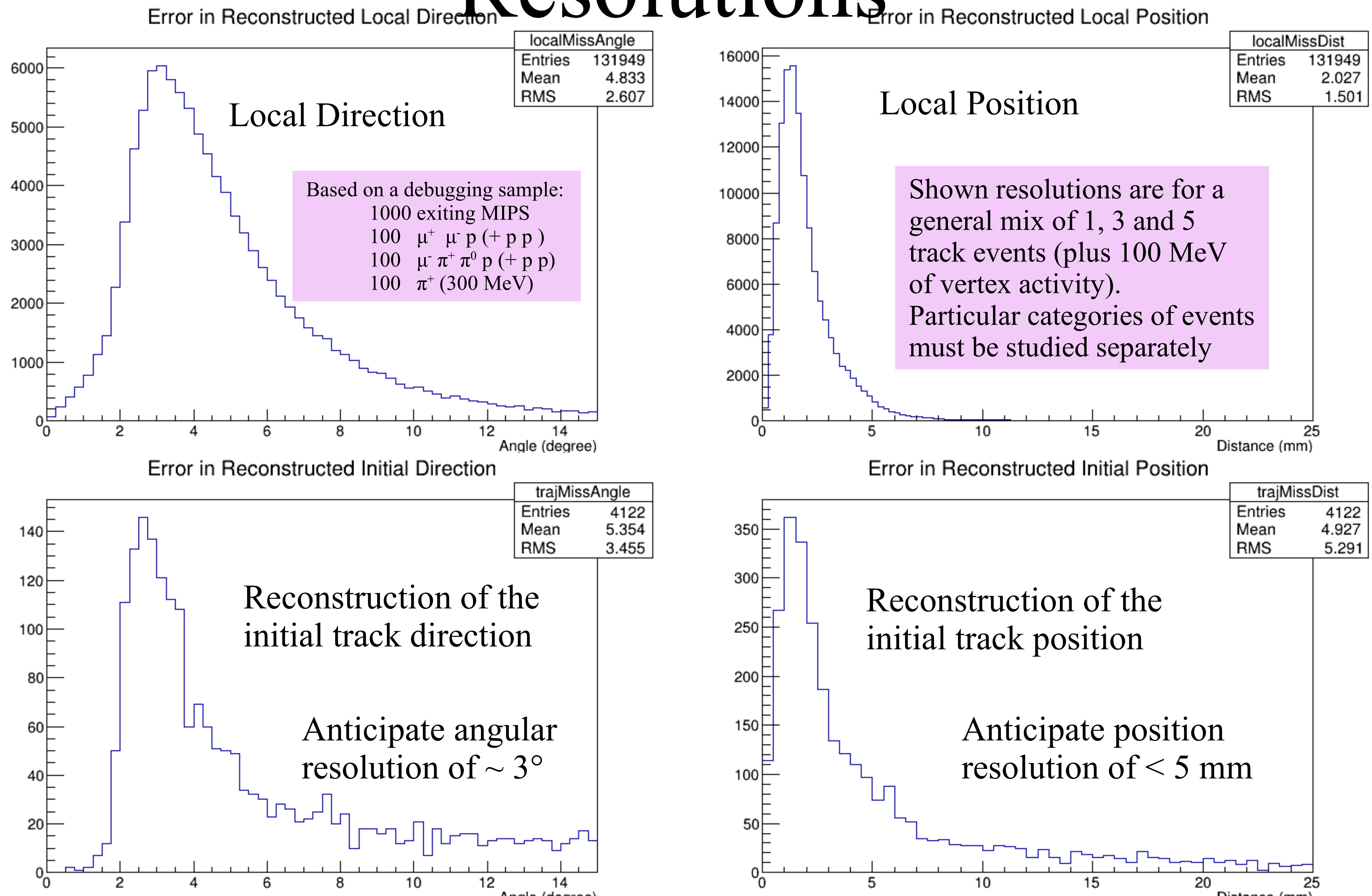


Hard scatter followed by decay in flight



Hard scatter to $\pi \rightarrow \pi$ (250 MeV) $\rightarrow$ p + p

Very Preliminary All Track Resolutions



Includes all tracks, including secondaries, delta-rays, decays (1900 primary, and 2222 secondary tracks).

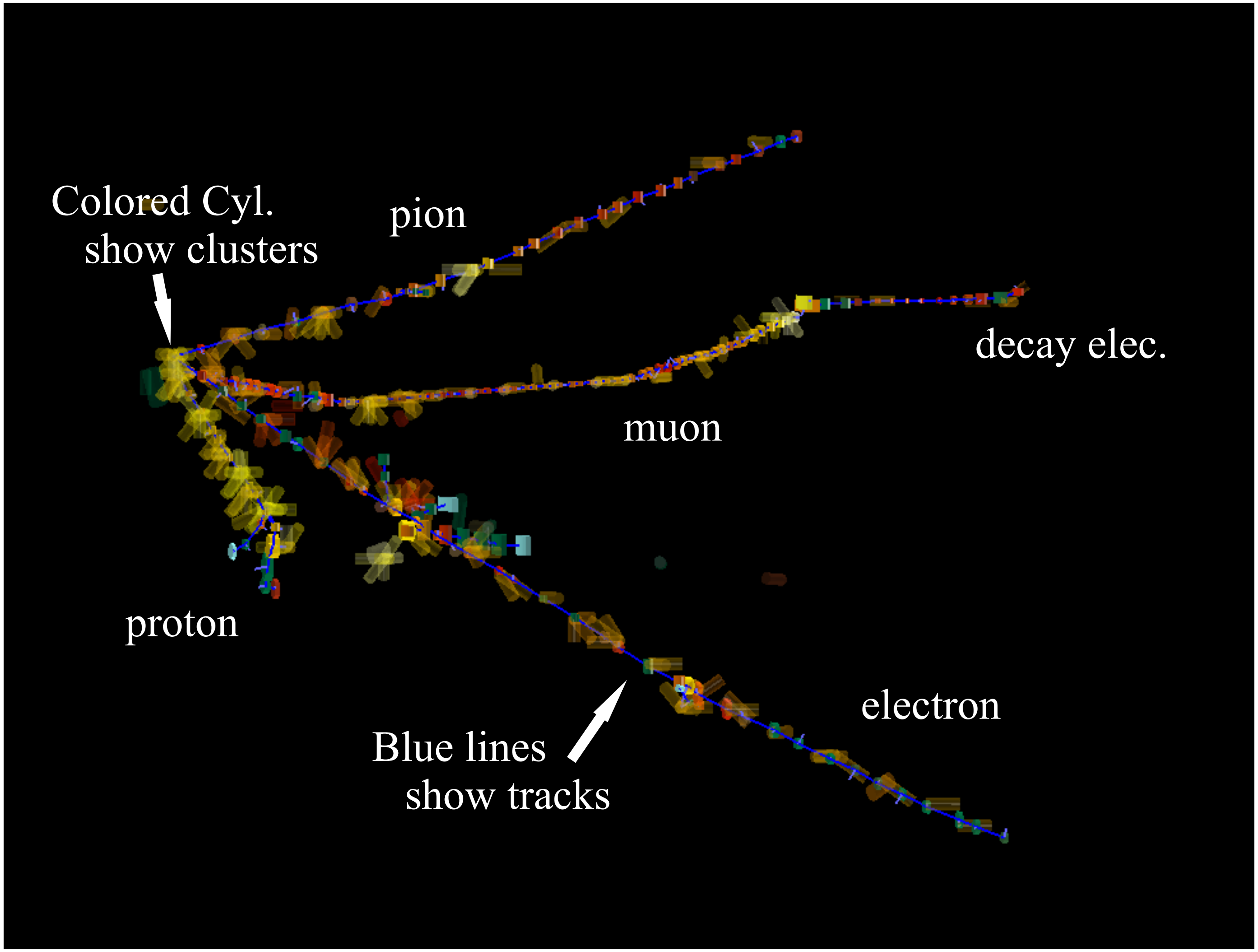
Summary

- The baseline SuperFGD reconstruction is ready[1] for use in physics studies
  - Products
    - Tracks with fitted results
      - Position, direction, length, “dE/dX” vs position on track, total energy deposition, path in detector
      - Minimum track length is 4 cubes
    - Clusters
      - Small groups of hits isolated in space and time.
      - Position, direction, shape, total energy deposition
  - Lots of work to be done
    - Efficiency studies
    - Benchmark physics signature studies
- Techniques are directly applicable to any cube based detector
  - 3DST is the overgrown twin of the SuperFGD

[1] Professional driver on a closed track. Your mileage may vary. Past performance does not indicate future results. Unintended uses may result in injury. Do not ingest Avoid contact with eyes. This is not a toy. Keep away from children. Please handle safely Use at your own risk. Danger, danger Will Robinson.

Backup Slides

Example Particle Bomb with Tracks



Color is roughly dEdX

Concepts in the ND280 Reconstruction

- The reconstruction objects. These are used to describe all of the information (*and everything has to fit into one of these*).
  - Everything coming out of the reconstruction is a
    - **<u>THit</u>** → This is a detector hit, generally produced by the calibration
      - TReconHit – This is a "reconstructed" hit. It is a "cheater's" cluster for use when a cluster is too expensive. Conceptually it's halfway between a hit and cluster.
    - **<u>TReconCluster</u>** → (a position) This is a group of (one or more) hits that is summarized as a "blob" (derived from TReconBase)
    - **<u>TReconShower</u>** → (a position and direction) An object that represents stuff in a "cone" (derived from TReconBase)
    - **<u>TReconTrack</u>** → (a path) This represents a path through the detector (derived from TReconBase)
      - Points along the path are represented by **<u>TReconNode</u>** with the "position" and a "measurement" (this is a simplification)
    - **<u>TReconPID</u>** → (id. information) This is an object that is based on a track, shower, or cluster. It has particle identification information attached. Don't assume a PID is a track! (derived from TReconBase)
    - **<u>TReconVertex</u>** → (a position) This represents a vertex with tracks, showers, and/or clusters. attached (derived from TReconBase)

Structure of The ND280 Reconstruction

- The ND280 reconstruction is built from "TAlgorithm" objects
- A TAlgorithm →
  - A "function" that does something (fitting, clustering, PID, &c)
    - Accepts one TAlgorithmResult object
    - Returns one TAlgorithmResult object
  - An algorithm can use other algorithms
- A TAlgorithmResult → The product of a TAlgorithm.
  - [optional] One or more THitSelection objects
    - A THitSelection object holds hits.
    - Convention: The selection name for hits used in the algorithm is "used". It should be the last selection added to the result.
  - [optional] One or more TReconObjectContainer objects
    - These hold the clusters, showers, tracks, particles and vertices
    - Convention: The container name for the results of the algorithm is "final". It should be the last container added to the result.
  - [optional] One or more sub-objects
    - including other TAlgorithmResult objects

Time Slicing and DBScan

- Time Slicing: Collect hits from one interaction
  - Break into MPPC hits into groups separated by a > 100 ns gaps.
    - Algorithm not optimal when there is noise (a fixed time window is better)
    - Should be optimized, but for now, the analysis isn't very sensitive to it.
    - Notice that electrons from muon decay are "just another track"
- 3D Hit Building:
  - Assign a "cube" to every possible three fiber combination
    - Hits are built by "time slice", so MPPC hits in different time slices are not considered
    - Charge is assigned to each cube based on event topology
    - Time is not reliably assigned (yet) because more thought is needed about how to handle ambiguities
      - Times of each fiber are available in the 3D hit.
- DBScan:
  - Use the geometric distance between cubes
  - Growth criteria
    - More than one neighbor within 1.6 cm (i.e. collect all simply connected cubes)

Minimum Spanning Tree Definition

- Vertices: The hits
  - Each cube is taken as a separate vertex
- Edges: The connections between the hits
  - Initially → every cube is connect by an edge
  - Finally → only the edges in the minimal spanning tree are kept
  - Edge Weight: The edge weight is defined as
    - The geometric distance between the cubes
      - Alternatively, the "chess-board jumps" between cubes.
    - Plus, a charge factor which makes low charge cubes "further apart"
      - e.g. 1 mm × exp(- (Q1 + Q2))
  - This produces an (almost) uniquely defined MST
    - Possible degeneracy when two cubes have identical charges
- Clusters are built for the "branches" of the tree.
  - Break at every junction (i.e. a cube with more than two edges attached)
  - Break at every kink
    - Cesar's definition: central cube more than 2cm (tunable) from line between end point cubes
  - Hits in the cluster are ordered from first to last
    - First and last hits will overlap with hits in other clusters

Cluster Growth

- The track segments found by the minimal spanning tree are combined into longer segments
- Possibly combine clusters sharing a single hit (i.e. they are neighbors)
  - Choose best pair of cluster based on
    - Product of the number of hits in each cluster (prefer more)
    - Product of the minimum charge in each cluster (prefer larger)
    - Change in χ^2 to a line ($\Delta\chi^2 = \chi^2_{1+2} - \chi^2_1 - \chi^2_2$)
  - If best pair meets the goodness criteria, combine
- Continue until there are no more candidates that can be combined.

Sequential Importance Resampling (SIR) Particle Filter (as a track fit)

- Particle filters and Kalman filters are both implementations of the general Sequential Bayesian Filter concept.
  - A particle filter is philosophically similar to an MCMC, and uses a large number of weighted "samples" to describe the fit posterior.
- General procedure:
  1) Choose a lot of samples from an assumed prior distribution
     a) e.g. choose 1000 points inside the first cube with directions toward the second cube. If you have enough measurements, the prior won't matter.
  2) Update the distribution by sequentially adding measurements
     a) Propagate each sample to the next measurement. Include the effect of multiple scattering, magnetic field, etc
     b) Weight each sample based on the likelihood of the measurement (the weighted samples describe the posterior based on the measurement).
     c) Calculate the sample average and covariance and use as the track state
     d) Repeat
  3) Resample the posterior when lots of samples have a "zero" weight.
  4) Finally: Use forward/backward smoothing to find the final state at each node.
- See TSFGStochTrackFitter for a working example using the SimpleSIR template.

The Stochastic Track Fitter

- Implemented by TSFGStochTrackFit
  - Based on the SimpleSIR template implementation.
- Uses a simple Gaussian likelihood based on the closest approach to the cluster position. Ignores the cluster time and charge.
- State Propagation: Move to the closest approach to the next measurement
  - Update the direction assuming Gaussian multiple scattering
    - Scattering assumes a 500 MeV/c muon (to be controlled by the parameters file).
  - Update the position assuming linear propagation (ignore curvature)
    - Apply effect of multiple scattering.
  - If measurement is missed by a large amount, assume a hard scatter occurred.
  - Do not update
    - Track energy deposition (calculated after the fit)
    - Track curvature (calculated based on the sagita, but currently ignored)
- Forward/Backward Filtering: Straight combination of the states from fitting in the forward direction, and the backward direction
  - Deweight states that are influenced by the choice of prior
- Underestimates uncertainty for long straight tracks due to discreet cubes.

Combine Short into Long Tracks (TSFGGrowTracks)

- Combine pairs of tracks that have a good match between the end points and directions.
- General algorithm:
  - Queue the tracks based on the number of nodes
    - Longest track first
    - Number of nodes is equivalent to the number of hits
  - For the front of the queue (longest first), compare with remaining tracks.
    - Don't match if the ends are far apart (default > 5 cm)
    - Don't match if the directions are consistent with a kink (default: > 30°)
    - Do match if the χ^2 between the end states is good enough (default: $\chi^2 < 16$)
    - Do match if the changes in direction and projected "miss" distance are small (default: < 10° and < 15 mm)
      - This manages the case where the fitted covariance is underestimated due to the discreet cube positions.
  - When a match is found
    - Remove both tracks from the queue
    - Combine the tracks and put the combined track at the front of the queue
      - It will be the longest track
  - If no match is found
    - Put the track into the output object container.

Merge Crosstalk Hits with Tracks

- Implemented in TSFGMergeXTalk
- Basic Algorithm
  - For all hits that have not been reconstructed into a track
    - Find the highest charge (immediate) neighbor that is part of a track
      - Only look along the the fibers that go through the hit
    - Combine the hit into the track associated with the neighbor
  - Cluster any remaining hits by position and time
    - Us DBScan and require 1 neighbor within 2 cubes to expand cluster
- Produces new tracks so that the input tracks are not modified.
  - Input tracks will have one hit per node
    Output tracks may have more than one hit per node.

Results of the Reconstruction

- Products
  - Tracks
  - Clusters
- Track objects contain
  - Position and Direction at each end of the track
  - The position, direction, and local charge collected along the track path
  - The covariance information for each parameter
  - Threshold for a track is 4 cubes
- Cluster objects contain
  - The Position of the cluster centroid
  - The moments of the charge distribution
  - The covariance information for each parameter
  - Threshold for a cluster is 1 cube (non-overlapping)