Hit-Based Kalman Filter Update

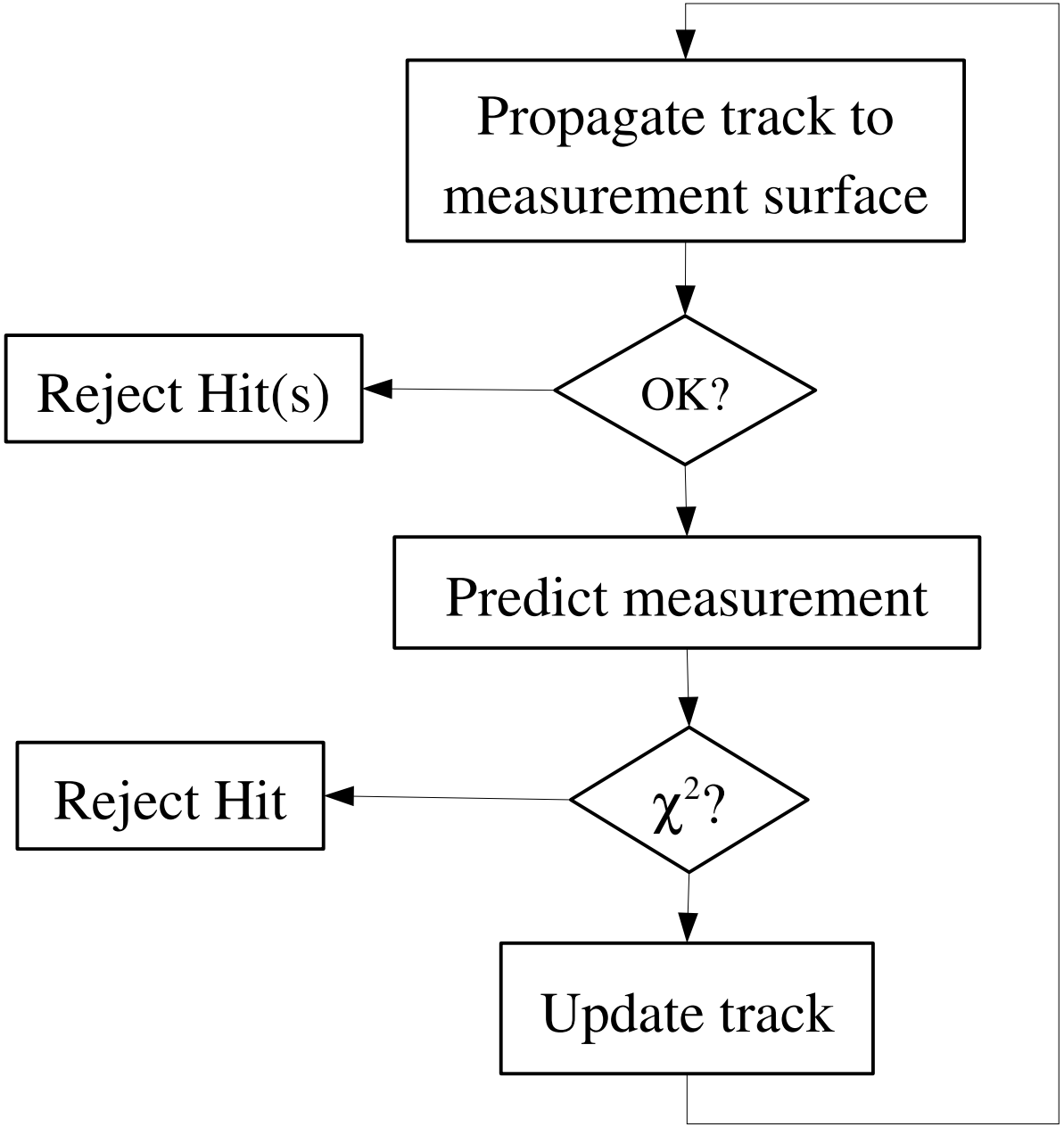
LArSoft Tracking Meeting
May 16, 2012

H. Greenlee

Outline

- Unidirectional Kalman fit.
- Bidirectional Kalman fit (smoothing).
- Using Kalman Filter for pattern recognition (seeding).
- Special problems for LAr TPC geometry.
  - Starting and stability problem.
  - Tracks parallel to wire planes.
  - Tracks parallel to wire in yz plane.
- TrackFinder modules.
- Examples.

Unidirectional Kalman Fit



Propagation

- Given track state vector **x** and error matrix C on initial surface S, calculate state vector and error matrix on distant surface S'.
  - In general mapping of state vector is some arbitrary nonlinear function (also nonlinear for straight tracks in LAr TPC).

$$\mathbf{x}' = f(\mathbf{x})$$

  - Propagation can be linearized with respect to some known reference trajectory ($x_0 \rightarrow x'_0$) in terms of propagation matrix P.

$$\mathbf{x}' = \mathbf{x}'_0 + P(\mathbf{x}-\mathbf{x}_0)$$

  - Propagation of error matrix depends on matrix P and noise matrix Q (Q depends on multiple scattering and energy loss fluctuations).

$$C' = PCP^T + Q$$

Prediction

- Given a track state vector $\mathbf{x}$ and error matrix C, calculate a predicted measurement vector $\mathbf{m}_p$, and prediction error T.
  - In general, preduction function can be nonlinear.

    $$\mathbf{m}_p = h(\mathbf{x})$$

  - Prediction function can be linearized with respect to some reference prediction ($\mathbf{x}_0 \rightarrow \mathbf{m}_{p0}$) in terms of matrix H.

    $$\mathbf{m}_p = \mathbf{m}_{p0} + H(\mathbf{x}\text{-}\mathbf{x}_0)$$

  - Error of prediction.

    $$T = HCH^T$$

Measurement and Residuals

- Actual measurements consist of a measurement vector *m* and error matrix *V*.
- From measurement and prediction, define residual **r** and error matrix *R*.

$$\mathbf{r} = \mathbf{m} - \mathbf{m}_p$$

$$R = T + V$$

- Incremental chisquare can be calculated from residual.

$$\chi^2 = \mathbf{r}^T R^{-1} \mathbf{r}$$

Track Update

- Track state vector **x** and error matrix C are updated by a correction that is proportional to the residual vector **r.**

$$\mathbf{x}' = \mathbf{x} + K\mathbf{r}$$

$$C' = (1 - KH)\, C$$
$$= (1 - KH)\, C\, (1 - KH)^{\mathrm{T}} + KVK^{T}$$

  - where K (Kalman gain matrix) is

$$K = CH^{T}R^{-1}$$

Real Tracks and Hits in LArSoft

- Measurement surfaces defined by wire and drift velocity.
- Track state vector:

$$\mathbf{x} = (x, v, w, \mathrm{d}x/\mathrm{d}w, \mathrm{d}v/\mathrm{d}w, 1/p)^T$$

  - Propagation is nonlinear because of *dw* in denominator of parameters 3 and 4.
- Hit measurement (prediction) vector and *H*-matrix are trivial.

$$\mathbf{m} = (x)$$

$$H = (1, 0, 0, 0, 0)$$

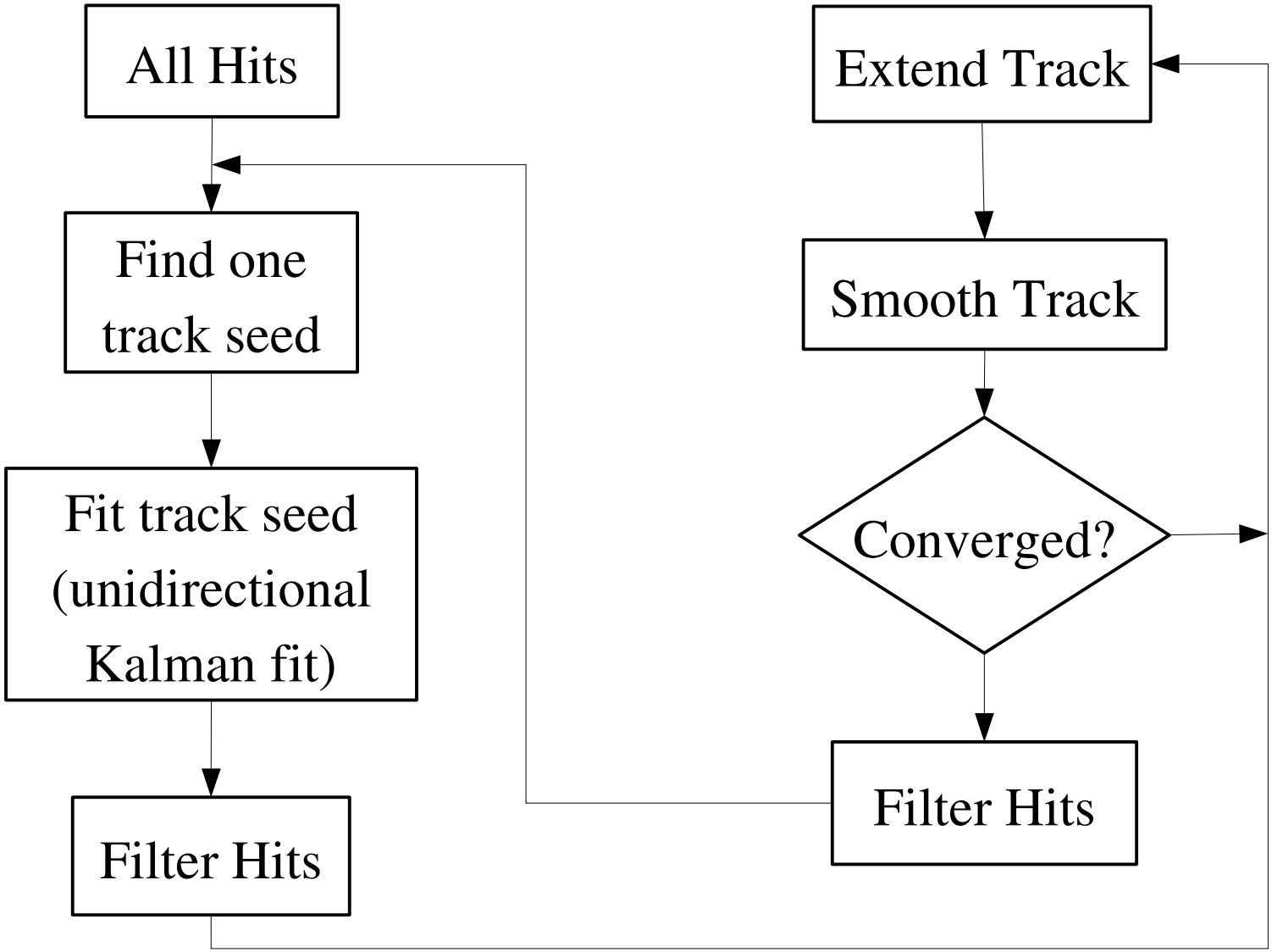
Kalman Smoothing

- Unidirectional Kalman fit produces optimal estimate of track state only at final measurement surface.
- "Smoothing" means following a unidirectional fit with a unidirectional fit in the opposite direction.
  - Track state estimate from forward and reverse fit can be combined to produce optimal estimate at each measurement surface.
  - Smoothing can be iterated to remove bias caused by initial guess.

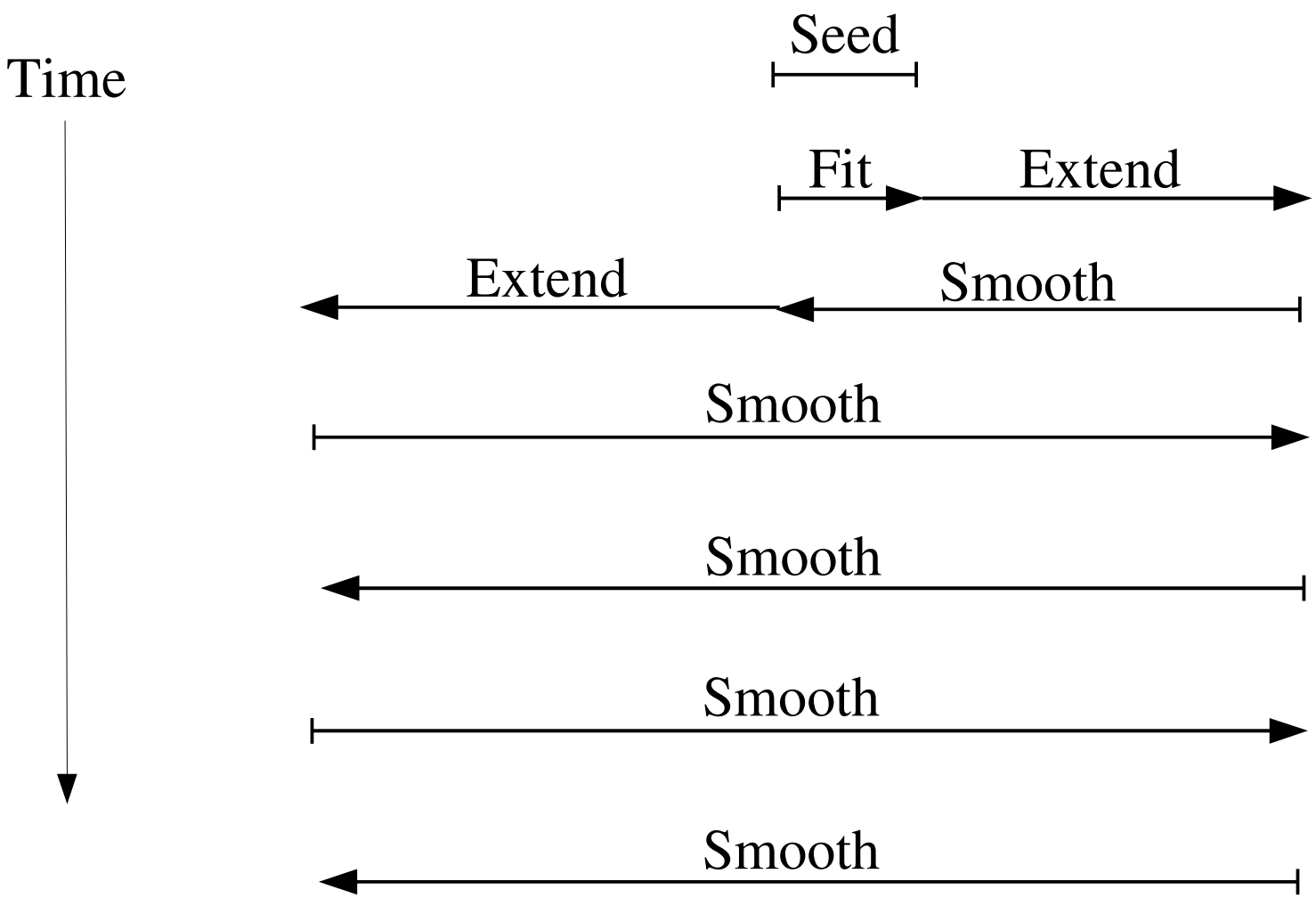
Using Kalman Filter for Pattern Recognition

- Because of its ability to easily calculate incremental chisquare, the Kalman filter can be used to select compatible hits (or the best hit) an any measurement surface.
- Track extension (Kalman filter):
  - Works just like unidirectional Kalman fit, except it tries to extend an established track by selecting the best hit (among all hits in the event) on a given surface.
- Combine Kalman filter with Ben J.'s seeding algorithm.

Seeding Algorithm



Seeding Algorithm



Fit Startup and Stability

- Kalman fit and track extension usually work fine for an established track with well-estimted parameters.
- Starting fit for a brand new track (seed) poses some difficulty.
  - Track parameters can be pulled to crazy values, false minimum.
  - I have found that false chisquare minimums can be eliminated or greatly reduced by using the linear approximation of the propagation equation, where the reference track come from external knowledge of track trajectory (seed parameters or mc truth).
- I have also found that fit stability is improved by only updating track on parallel surfaces.
  - Updating equation modified by $H \rightarrow HP$

Difficult Situations

- Track parallel to wire planes.
  - Kalman filter seed algorithm can usually handle this, provided seed and track can be established in non-parallel region.
- Track parallel to wire in *yz* plane.
  - Kalman filter seed algorithm can handle this situation too.
  - Track is mostly fit using the two non-parallel views (provided there are three views).
  - Main thing is to ensure that track is not screwed up by parallel view hits.

Hit-Based Kalman Filter Modules

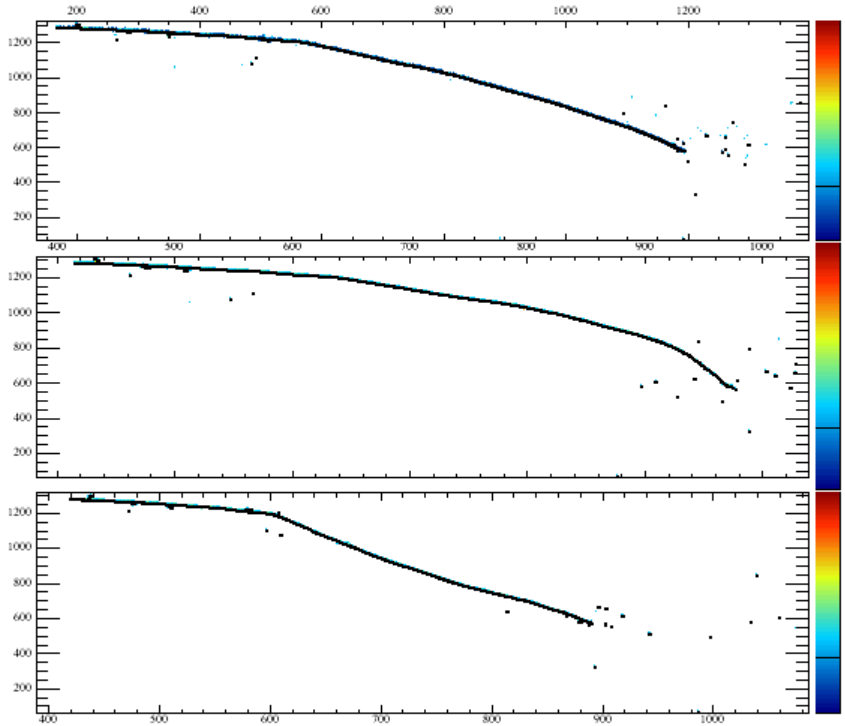
- TrackKalmanCheater.
  - Iterative fit + smooth algorithm.
  - Used mc truth to identify Hits and for initial reference track.
  - Mainly a warm up exercise for following module.
- Track3DKalmanHit.
  - Seed, then iterative extend + smooth algorithm.
  - Does not use mc truth.
- Both modules:
  - Make Track objects.
  - Make SpacePoints from adjacent Hits as a side effect.

Testing Track3DKalmanHit

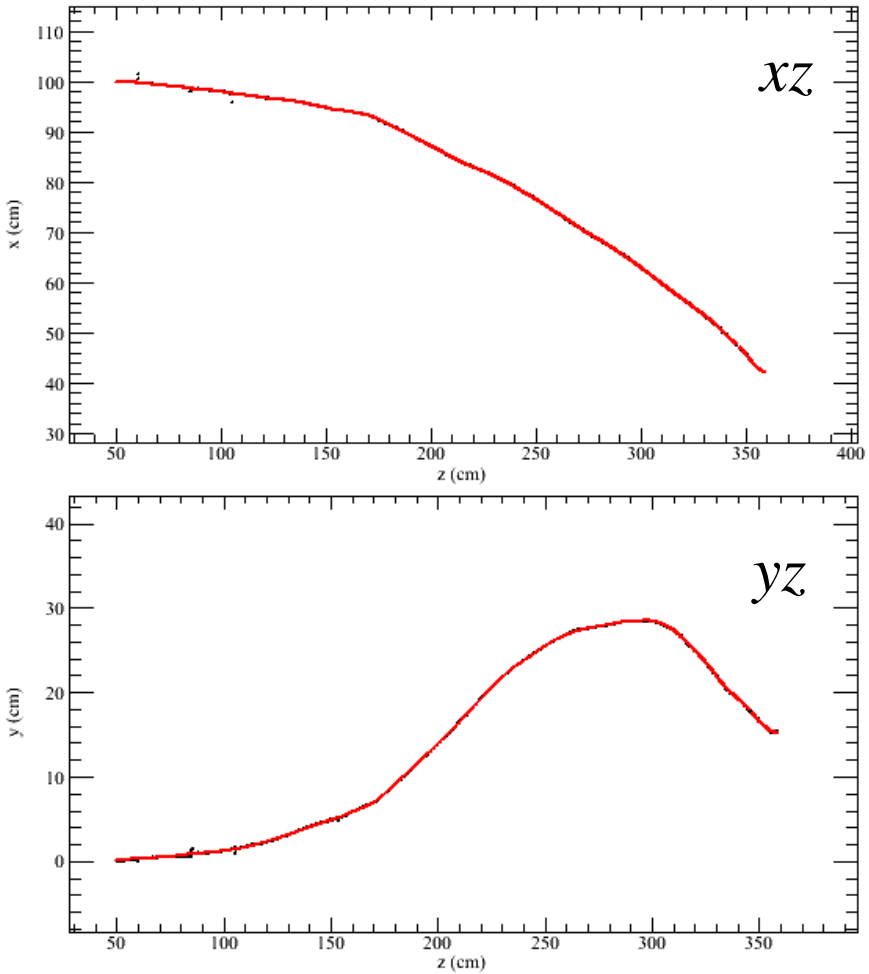
- Tested Track3DKalmanHit on various kinds of mc samples.
  - Single muons: $\theta_{xz} = 10 \pm 5°$, $\theta_{yz} = 0 \pm 5°$ (easy tracks).
  - Single muons: $\theta_{xz} = 0 \pm 5°$, $\theta_{yz} = 0 \pm 5°$ (parallel to wire plane).
  - Single muons: $\theta_{xz} = 0 \pm 5°$, $\theta_{yz} = 30 \pm 5°$ (along wire).
  - Genie ν_μ CC.
- On following slides:
  - Black points on 3D/ortho view: truth space points (for visual reference).
  - Color points on 3D/ortho view: Trajectory points in reconstructed tracks.

An Easy Track

Time-Wire View

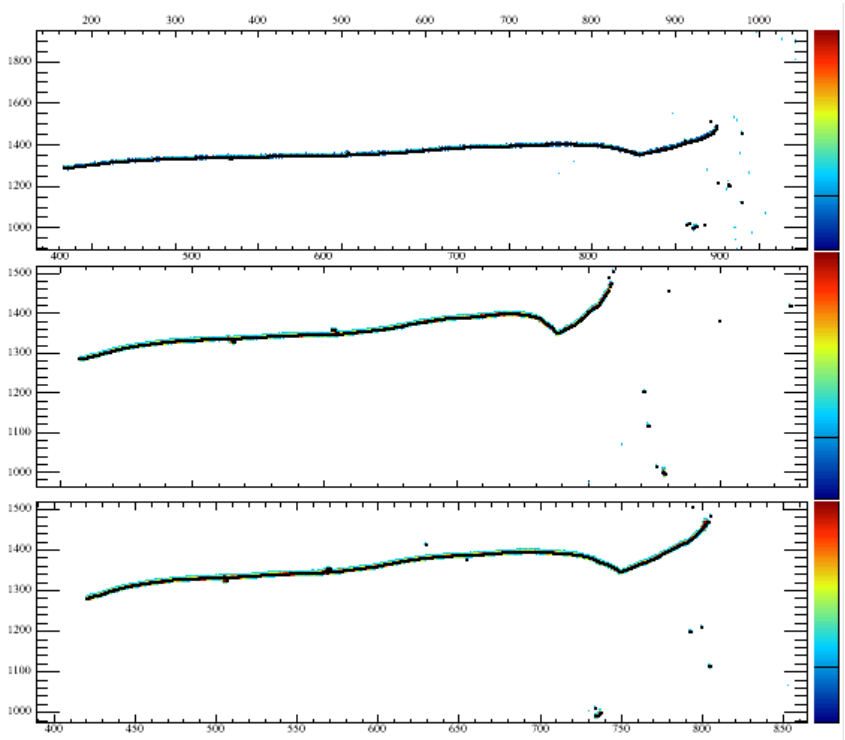


Ortho 3D View

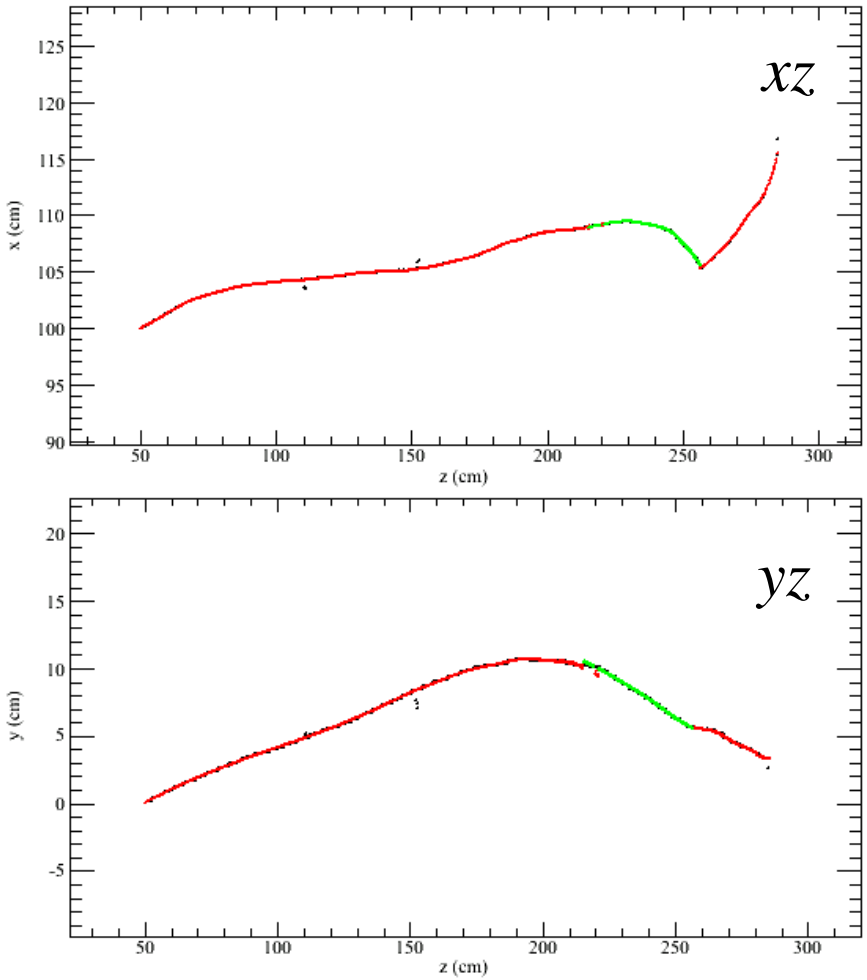


A Partial Reconstruction Failure

Time-Wire View

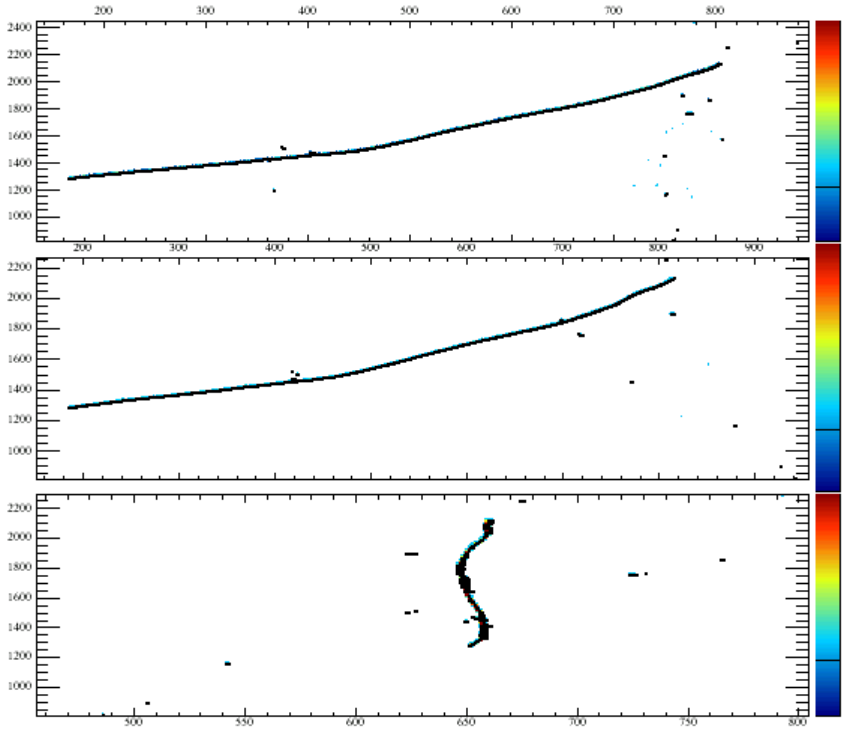


Ortho 3D View

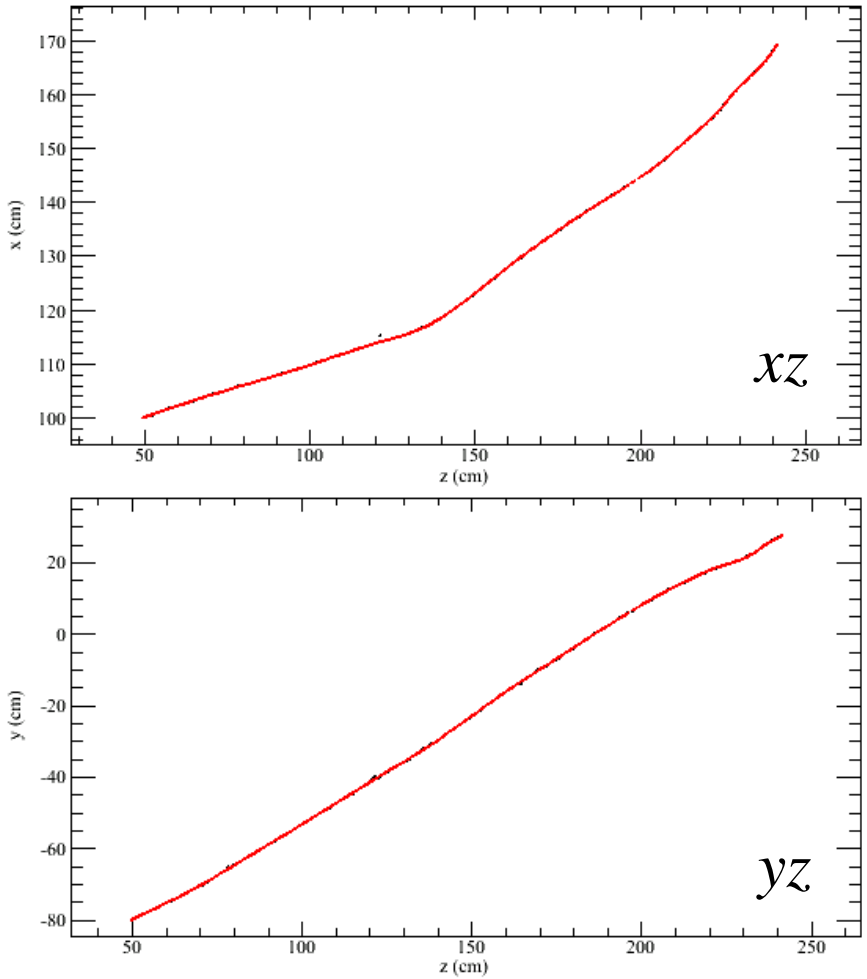


A Difficult Track

Time-Wire View

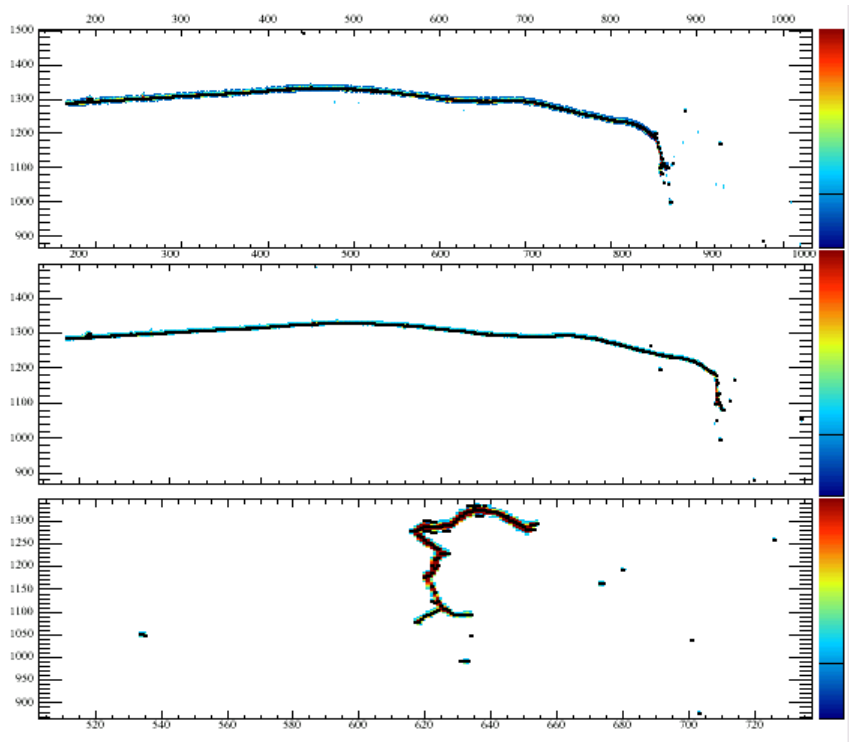


Ortho 3D View

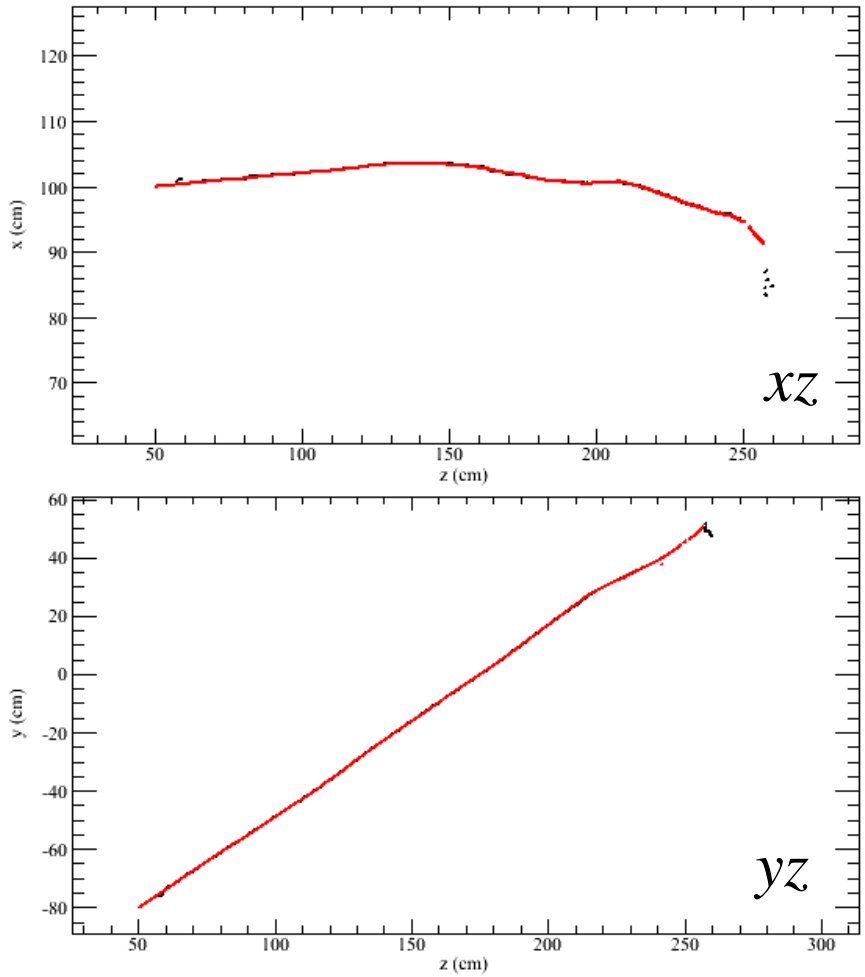


Another Difficult Track

Time-Wire View

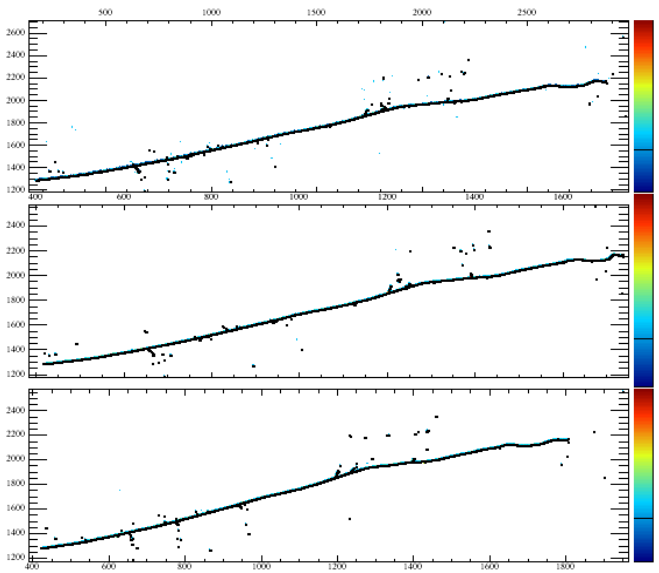


Ortho 3D View

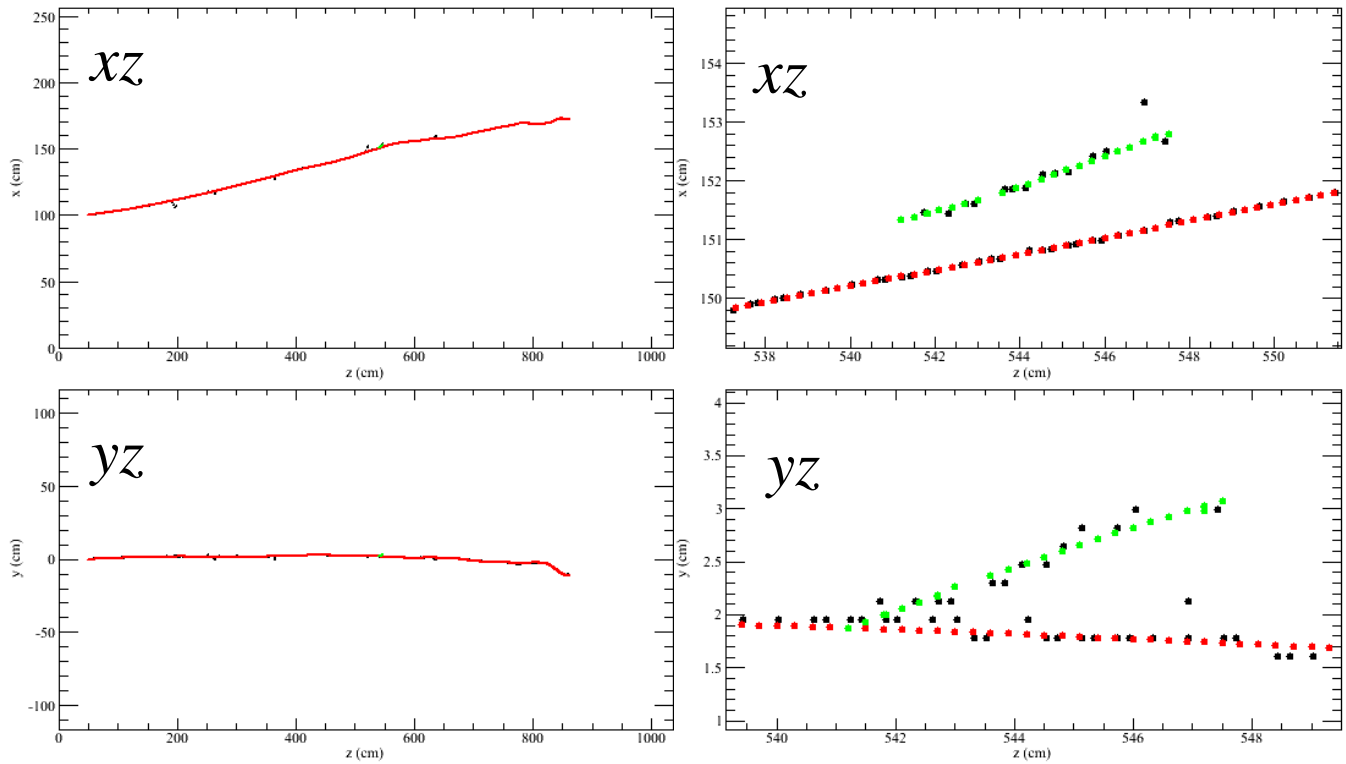


A Reconstructed Delta Ray

Time-Wire View

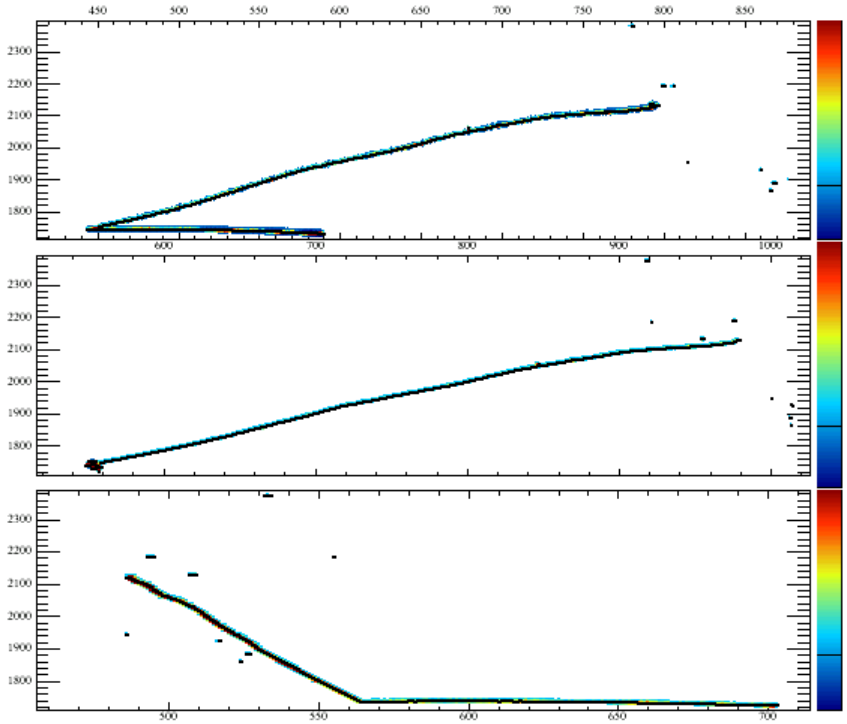


Ortho 3D View

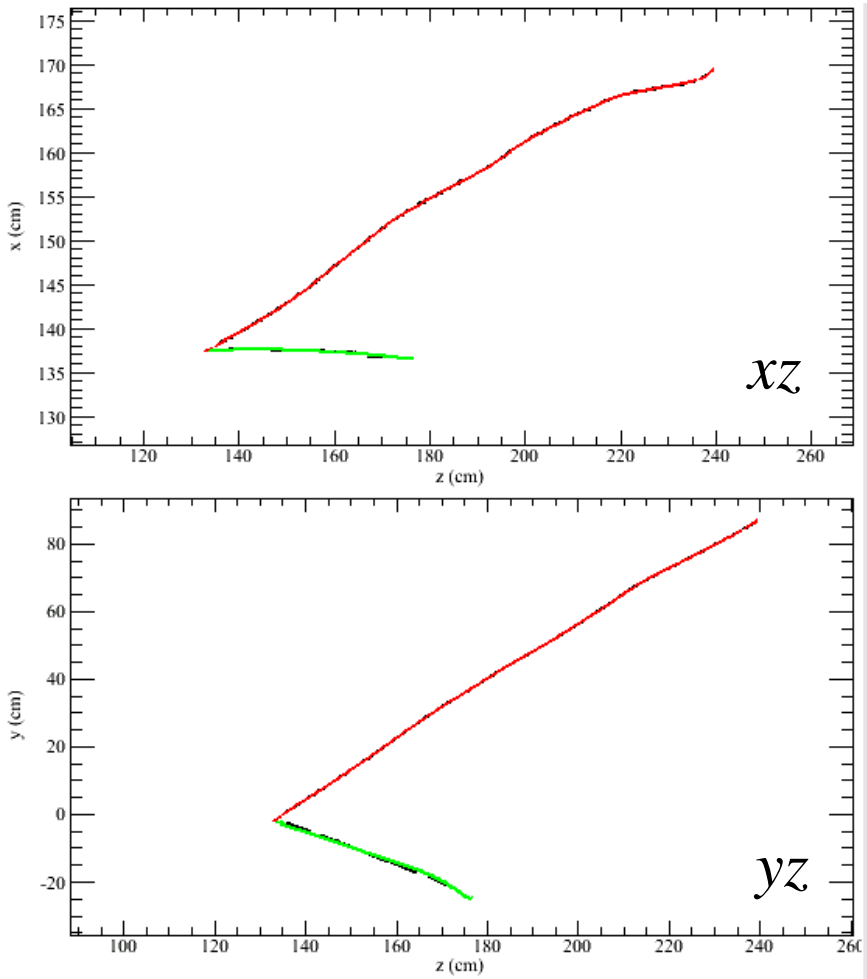


Genie Event with Two Tracks

Time-Wire View

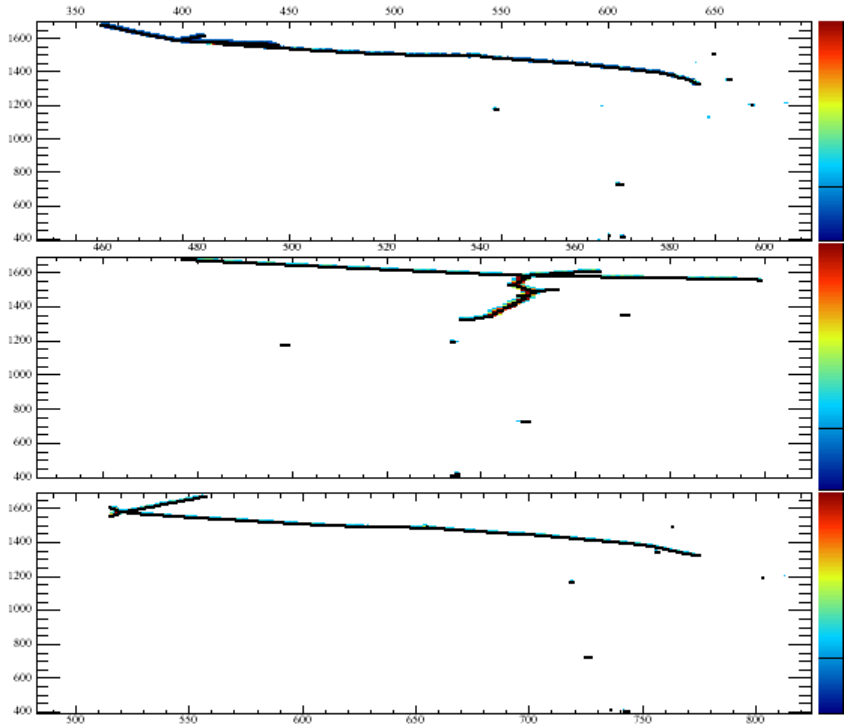


Ortho 3D View

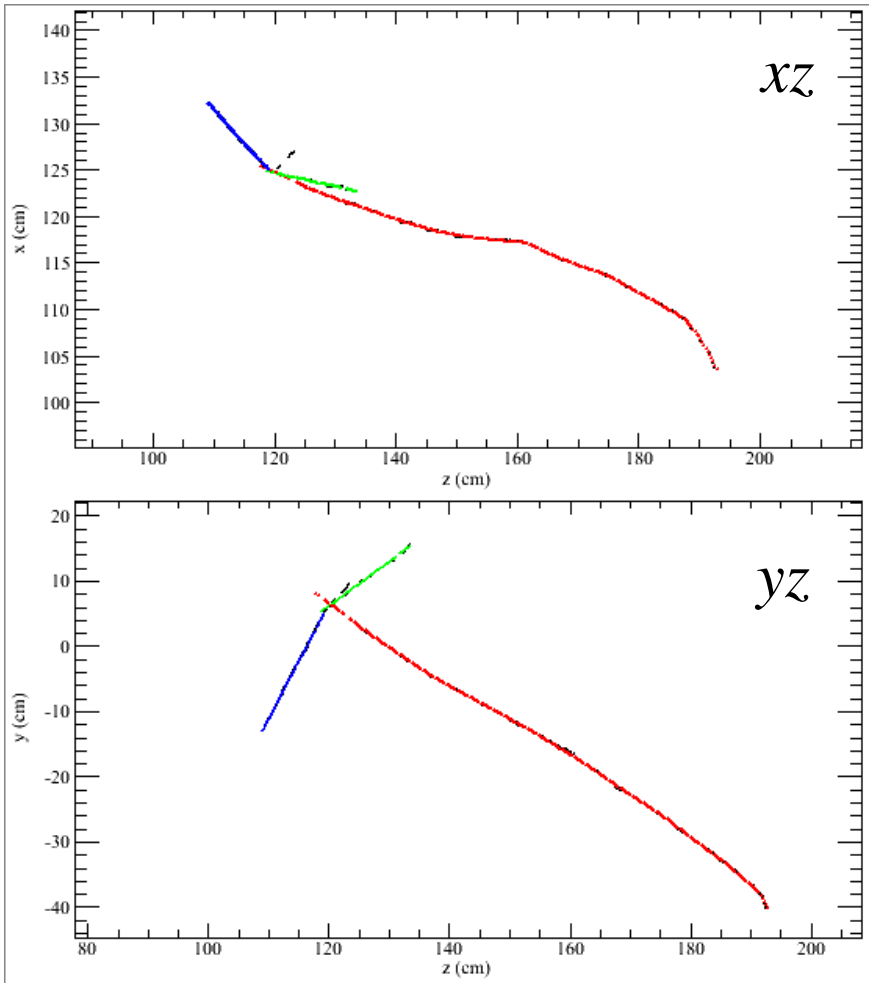


Genie Event with Three Tracks

Time-Wire View

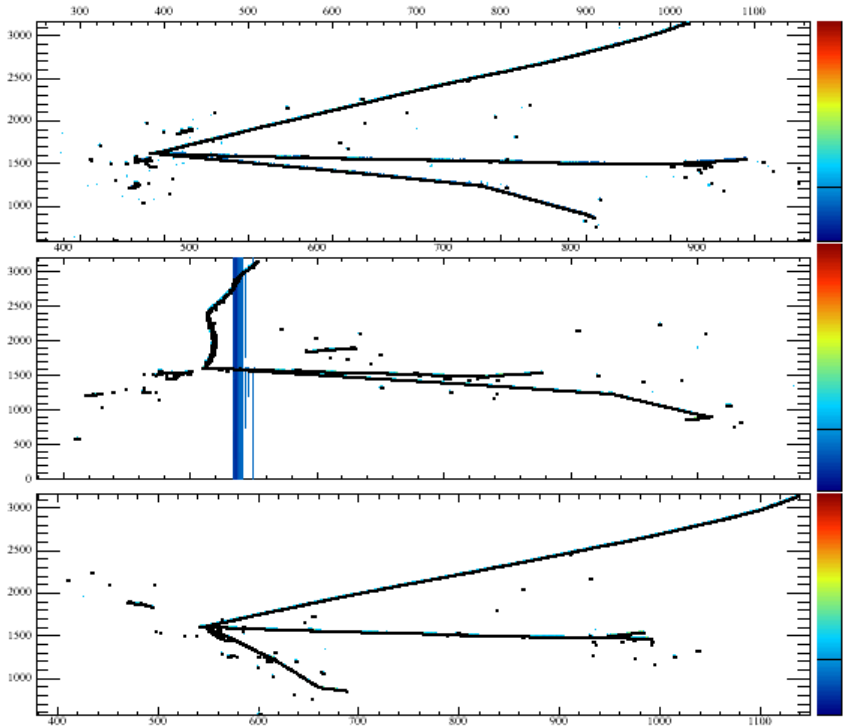


Ortho 3D View

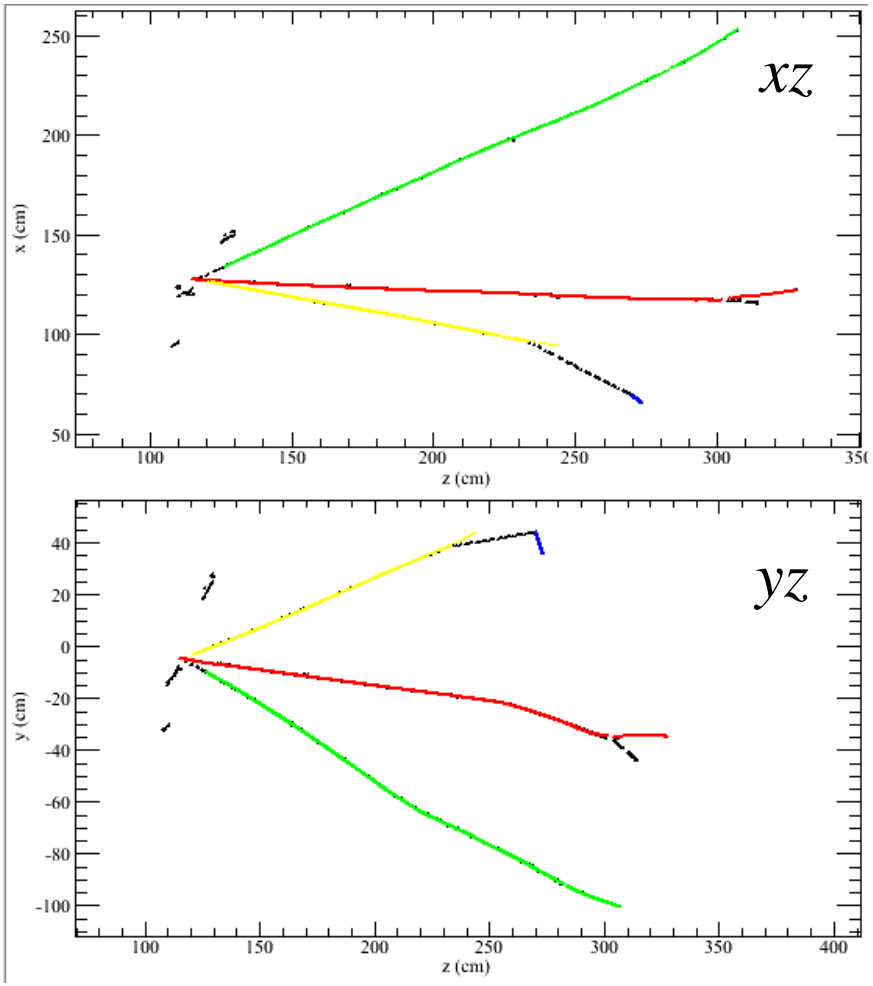


Genie Event with Multiple Tracks

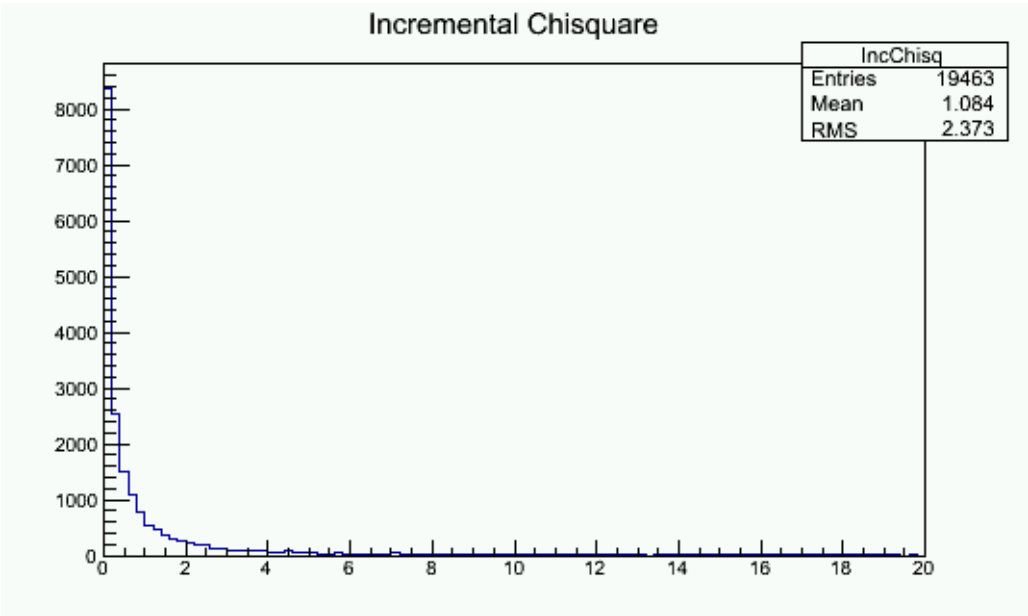
Time-Wire View



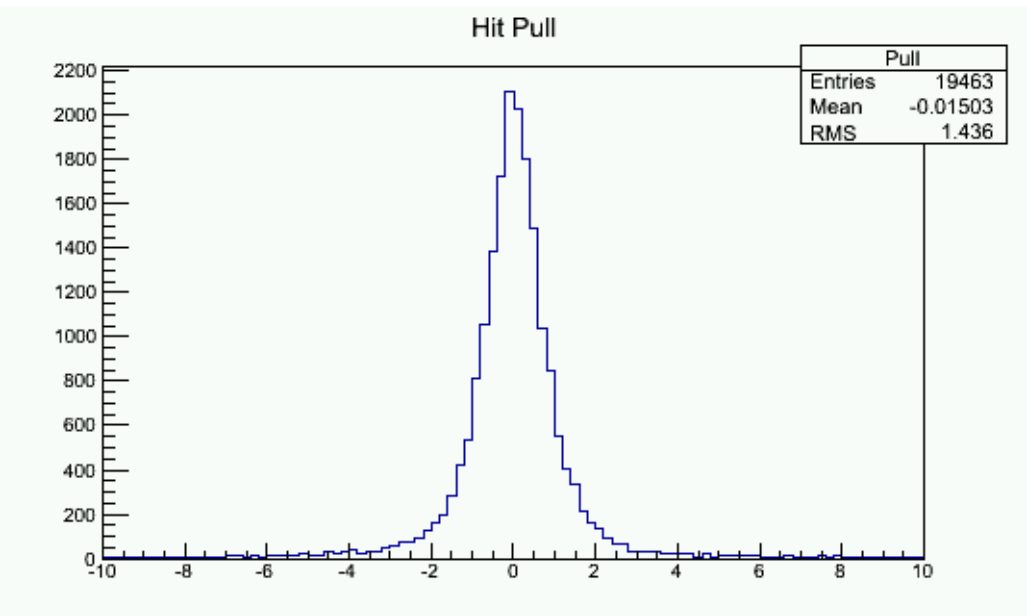
Ortho 3D View



Incremental Chisquare and Pull (per Hit)



$\Delta\chi^2$



Pull

- Made using truth momentum (KalmanTrackCheater).
- Hit error set to (one tick) / sqrt(12).

What is Missing / Still Needed

- Full information about track is not yet stored in reconstructed Track object.
  - Full 5x5 error matrix, pointing error.
- Momentum determination (current Kalman filter doesn't do it).
- Continuing improvements in robustness, reliability, tuning.
  - Short tracks, track stubs (perhaps an entirely different algorithm).
  - Very flat tracks.
  - Track splitting and merging (gaps in tracks).
  - Improvements in seeding.