Thoughts on Space Points and 3D Reconstruction in LarSoft

LarSoft Meeting

Apr. 20, 2011

H. Greenlee

Reconstruction of Space Points from Hits

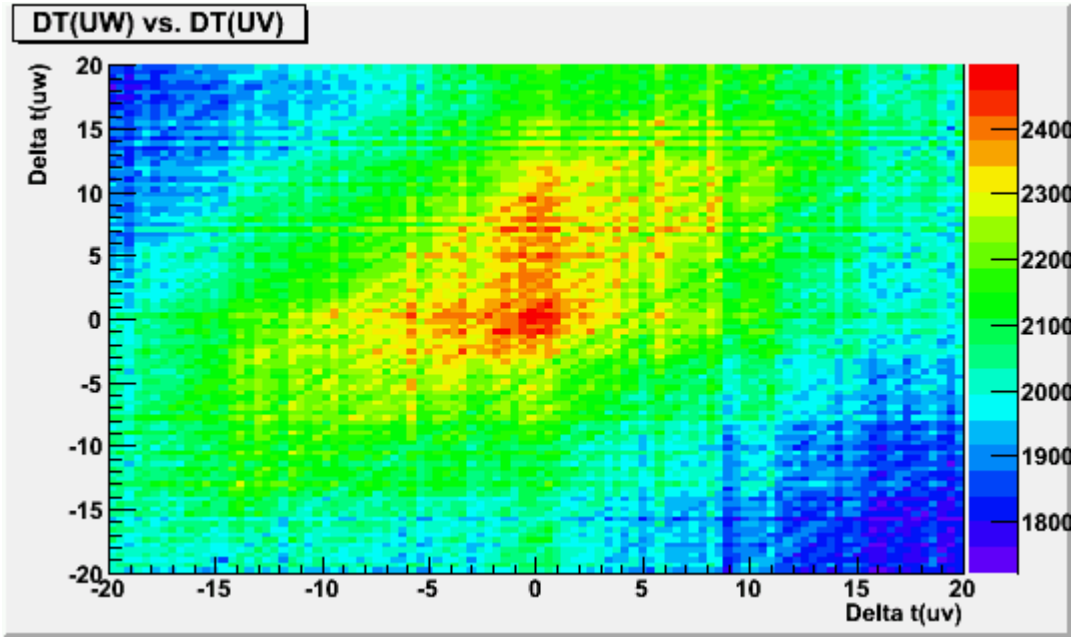
- With two views, two hits (t_1,u), (t_2,v) will make a 3D space point with one constraint.
  - $t_1 - t_2 = 0.$
- With three views, three hits (t_1,u), (t_2,v), (t_3,w) will make a 3D space point with three constraints.
  - $t_1 - t_2 = 0.$
  - $t_1 - t_3 = 0.$
  - $\Delta S = \sin(\theta_v - \theta_w)u + \sin(\theta_w - \theta_u)v + \sin(\theta_u - \theta_v)w = 0.$

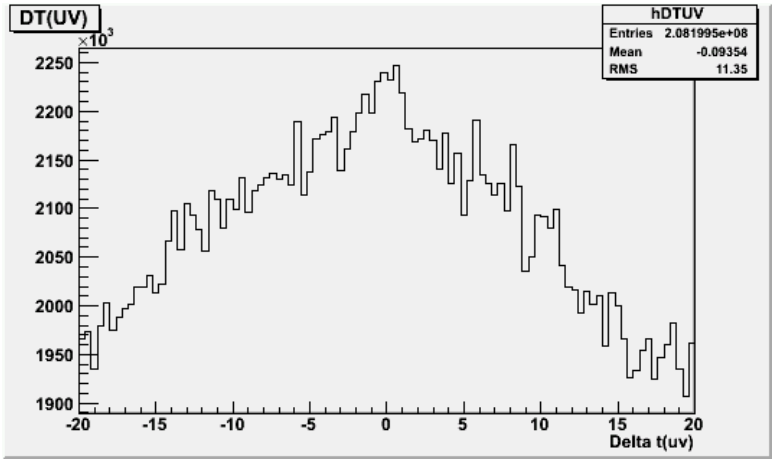
Reconstruction of Space Points (cont.)

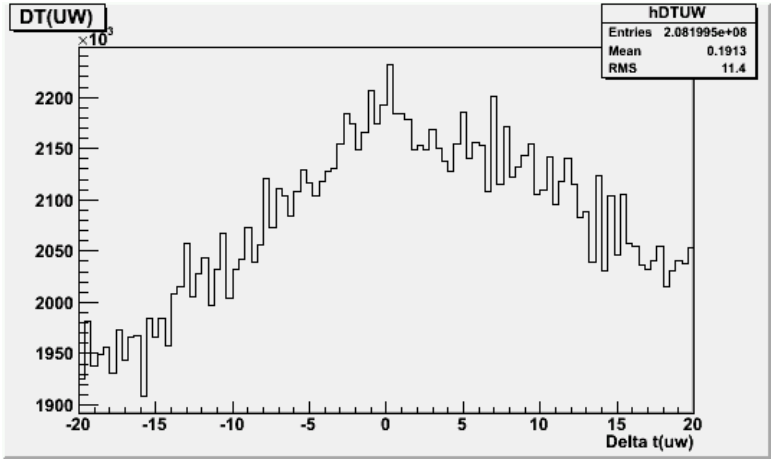
- The space point reconstruction problem is totally different for two views and three views.
- There is no hope of reconstructing space points before clustering from two views for an event that is more complicated than a single track.
- There is some hope of reconstructing space points before clustering from three views.
  - On following slides I tried this using Microboone geometry for 10 events.

Time Coincidence – All Hits

$\Delta t(uw)$ vs. $\Delta t(uv)$

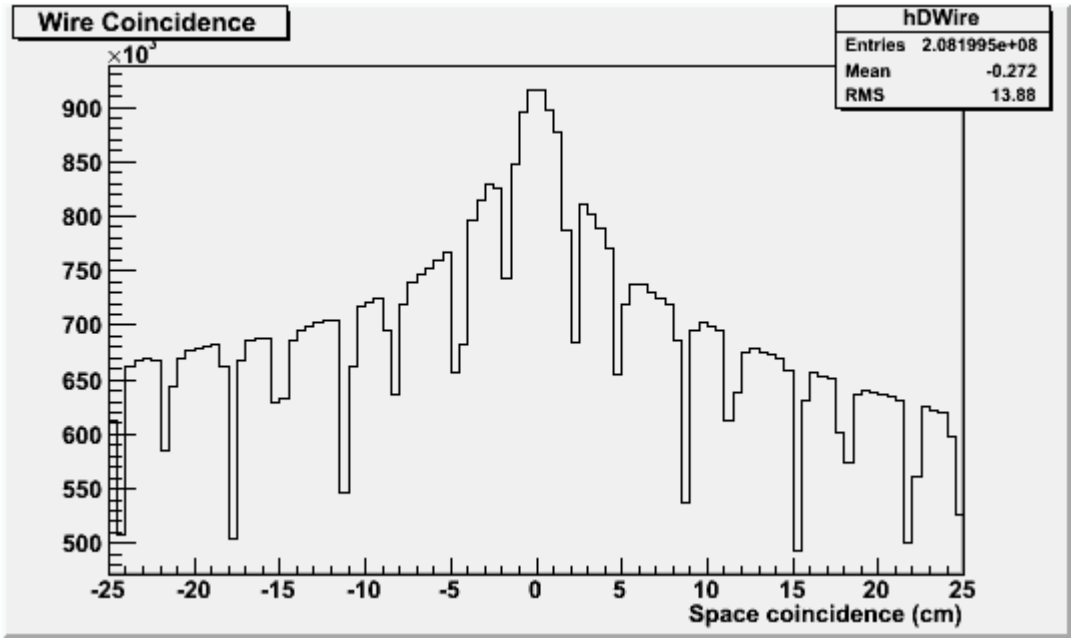






Space Coincidence

- Includes all hits in bounds of plot on previous slide, i.e.
  - $\Delta t(uv) < 20$ ticks.
  - $\Delta t(uw) < 20$ ticks.

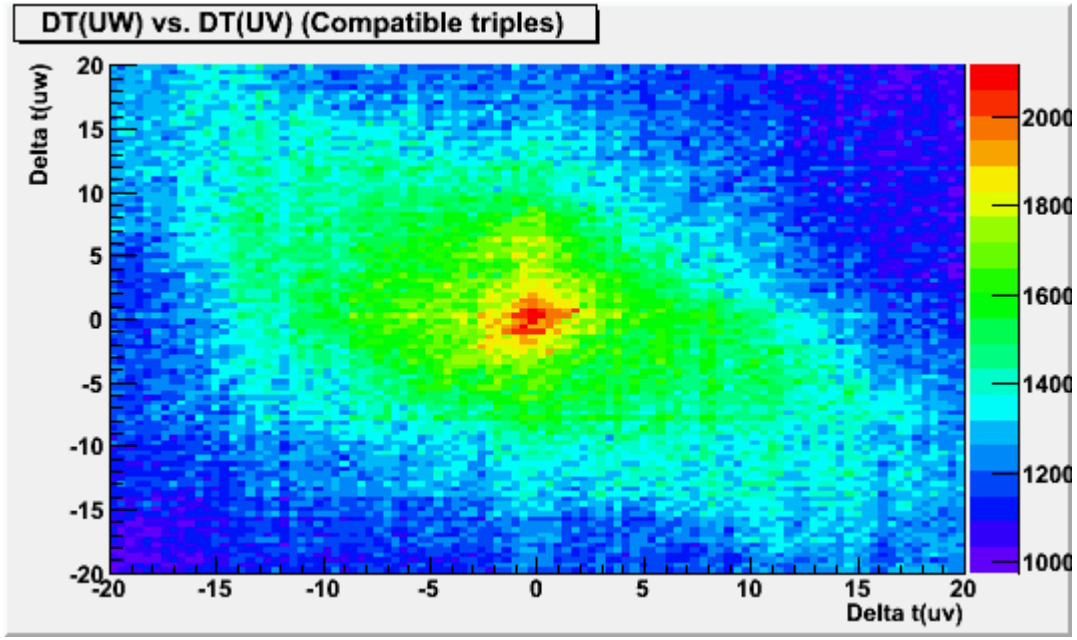


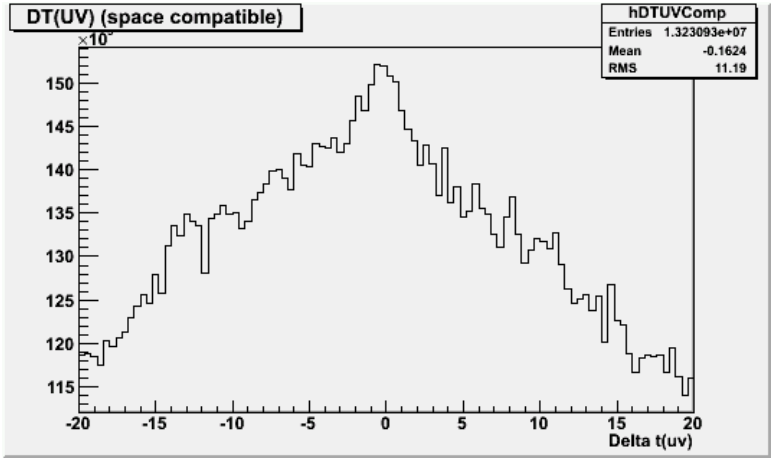
ΔS (cm)

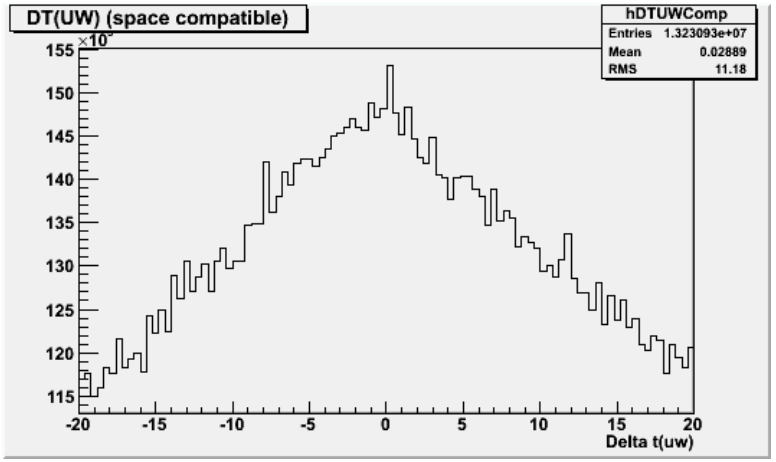
Time Coincidence – Spatially Coincident Hits

$\Delta S < 4$ cm

$\Delta t(uw)$ vs. $\Delta t(uv)$

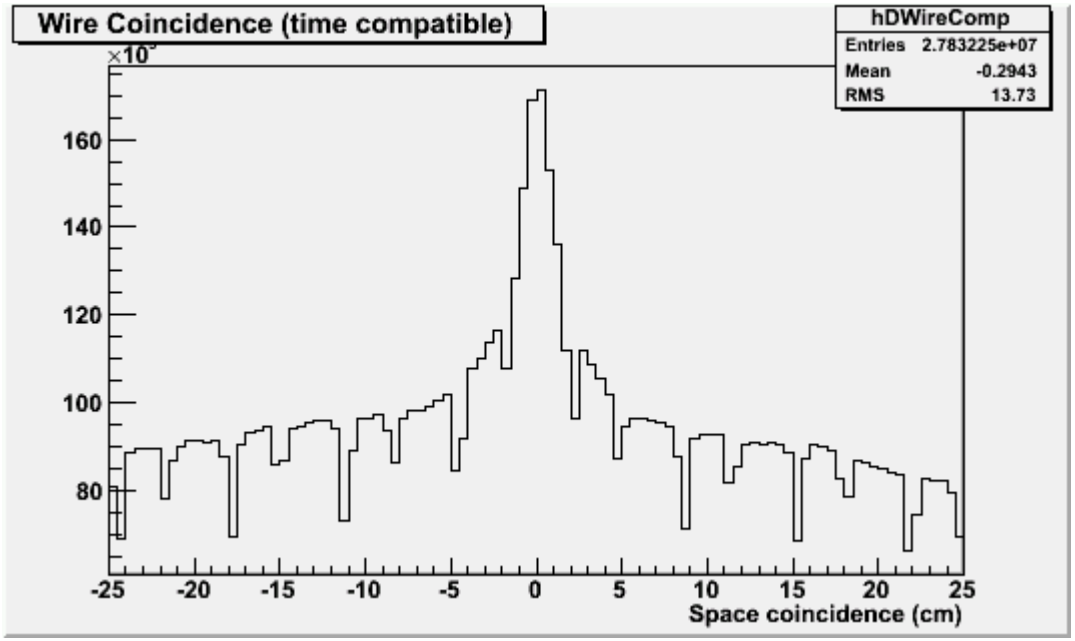






Space Coincidence II

- Includes all hits with tighter time coincidence cuts.
  - $\Delta t(uv) < 7$ ticks.
  - $\Delta t(uw) < 7$ ticks.



ΔS (cm)

Space Points Before Clustering Conclusions

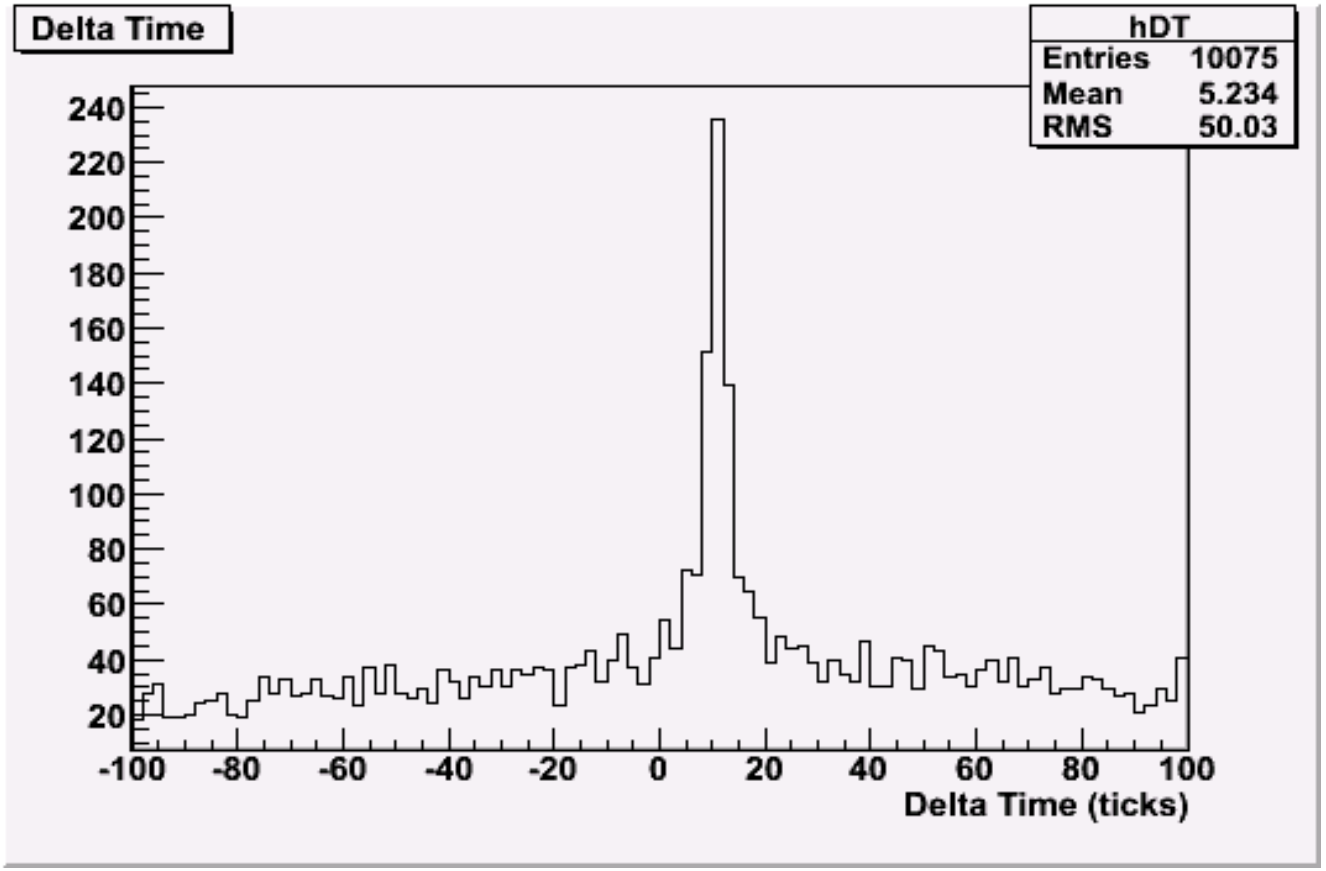
- Space points before clustering still has significant combinatoric background, but can achieve a high enough purity to be interesting.
  - Due to presence of nearby real hits, it is not obvious that one expects a true peak above background.
  - There is probably room to tighten the cuts further, not sure how much. Not obvious how to tune.
    - Is there mc truth information down to voxel level?

3D Cluster Reconstruction

- In the case of two views, 3D reconstruction must necessarily come after 2D pattern recognition in the two views.
  - Matching of reconstructed objects will mainly rely on finding time-coincident features of reconstructed objects between the two views.
  - Current 3D track matching (Track3Dreco) relies on finding track segments with two time-coincident endpoints.
- On following page, time difference between Harris endpoints.

Harris Endpoint Time Difference

- Time difference of Harris endpoints found in collection and induction view (genie events). Not corrected for drift time between wire planes.



Matching Endpoints

- Matching endpoints clearly shows a signal, but one probably needs to match multiple features per cluster (like Track3Dreco).
- For the above to make sense, there needs to be an improved cluster splitting/merging algorithm to make clusters more compact and more planar.
  - Existing modules HoughLineFinder and LineMerger do a decent job of cluster splitting/merging for track-like clusters.
  - Is argoneut satisified with finding track-like clusters?

Some Other Ideas for Improvements

- Correct hits for drift time between planes at hit reconstruction level so each module that matches views doesn't have to do it independently.
  - I couldn't figure out how to determine the drift time between planes from geometry or LAr properties services. This information is hard-coded in Track3Dreco.
- Rotating the event display by 90° (so that views are stacked on the time axis) would make it much easier to spot time-coincident features by eye.
- Add more qualitative information to endpoints (e.g. to distinguish vertices from stopping points).