Pattern Recognition Confusion Update

Doug Roberts

University of Maryland

Overview

- Goal is to be able to model effects that would be present if we ran an actual pattern recognition algorithm in FastSim without actually taking the time to do it.
  - Tracks stealing hits from other tracks
  - Hits that are too far from the track not being included
- What this WON'T do is find fake tracks from random hits
- Want to do this is a detector-model independent way and have the ability to include hits from various background sources

Procedure

- During the Hit Merging process, which runs prior to track reconstruction, SimHits that are close to another SimHit but not so close that they end up getting merged together are flagged as potential PatRec candidates
- We don't know the track resolution at this point so we just create this list of candidates and attach it to the SimHit
- Track reconstruction is now done in a momentum order from highest to lowest
  - This assumes that we would do a better job of PatRec for high *p* tracks
  - The entire list of PacSimTrks is now sorted; not sure if this will affect anyone else?
- During track reconstruction (PacSimTrk becomes TrkRecoTrk) these extra SimHits are used to create extra TrkHitOnTrks (HoTs) but they are initially inactive
  - Should be a no-op w.r.t. the track fit results and I believe it is

Procedure (cont.)

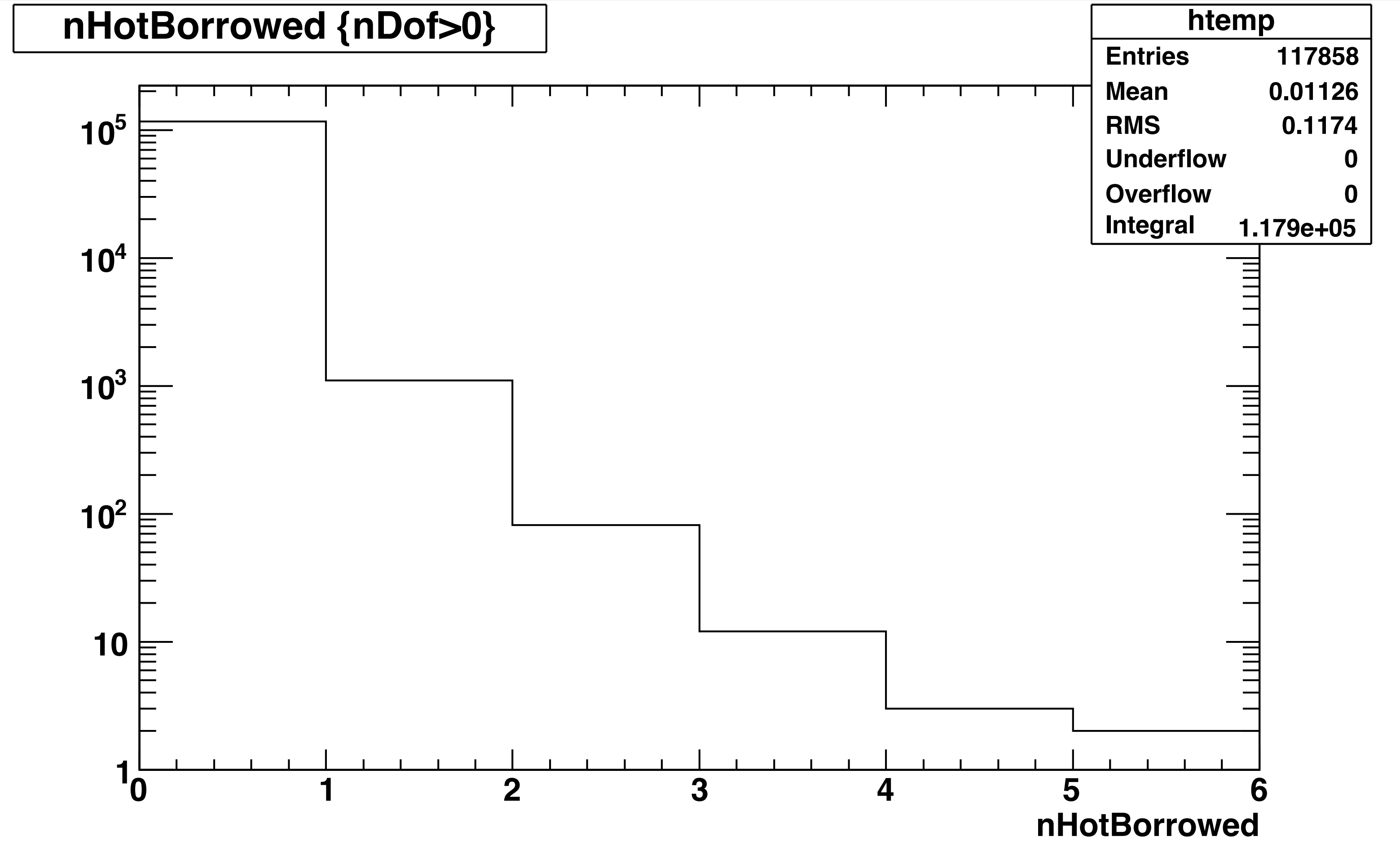
- The track is then fit as usual, in both directions
  - This is now the default
  - It allows us to know the full track parameters and error matrix at every point along the track
- After the initial track fit, we iterate over the active (original) HoTs and see if there are any PatRec candidates to be considered
  - Requires going back to the SimHit (more on that later)
- If there are, we get the unbiased residual for the original HoT and its error
  - The error defines a PatRec "road"
  - The "road" actually used swaps out the original HoT's hit resolution for the PatRec candidate HoT's hit resolution

Procedure (cont.)

- For each PatRec candidate HoT, we grab it's residual
  - Since it is on the track, even though inactive, the residual can be determined
  - The residual will be biased, however, by the presence of the original HoT that is active and on the track
- Compute a χ^2-like quantity: $\text{resid}^2/\text{road}^2$
- Find the PatRec candidate with the smallest χ^2
- If this χ^2 is less than 2× (arbitrary) the original HoT's χ^2, swap them, i.e. make the original HoT inactive and the other HoT active
  - Can make this swapping criteria softer to generate more swaps
- Had to be careful handling hits in pixel detectors
  - Two views are really tied together and each view is a separate HoT
  - Don't want to swap out just one view and χ^2 needed to be computed for both views
  - Had to create some mechanism to link the two HoTs together
- After iteration over all active HoTs, refit the track if necessary

Procedure (cont.)

- Once this hit-swapping is done, the hits get "weeded"
  - Iteratively remove (deactivate) HoTs with large contribution to the track χ^2 and refit
  - Can remove up to 5 HoTs (settable in
    - Might change this to 10
- After the track gets "weeded", go through and "lock" all of the active HoTs on the track so they can't be used be subsequent tracks
  - Really locks the SimHit that generated the HoT because one SimHit can be used to create many HoTs



Number of “Stolen” HoTs on a Track

Small effect with no background. Only about 1% of the tracks end up with HoTs that weren’t originally from that SimTrack.

SimHit-to-TrkHitOnTrk Map

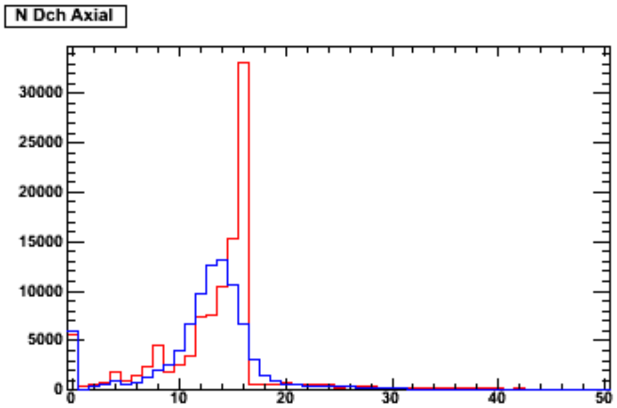
- In order to get this all to work, I created a class, PacTrk/PacTrkSimToHotMap, that keeps track of which SimHit created which HoT and vice versa.
- It also hold maps for
  - Pixel HoT pairs
  - PacSimTrack <-> TrkRecoTrk
- Works by having the maps as static private data members
- Just in case anyone else would like to use this info…

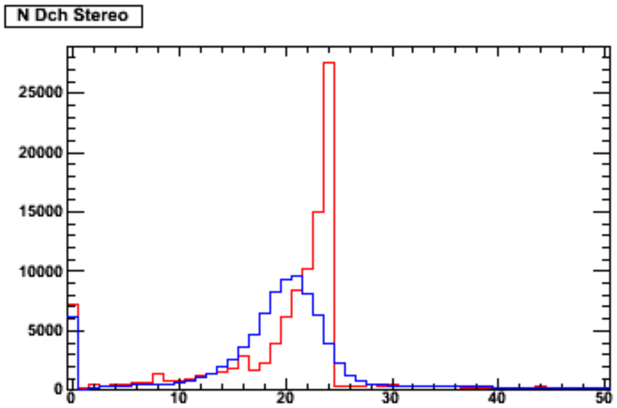
Status

- Code is pretty much finished. Will probably require some tuning, especially once backgrounds are added
- I will commit once I update my V0.2.0 release with the latest patches. I was waiting for the dust to settle on the patches.
- One comment: With the Hit Merging and this PatRec confusion we are adding some tails to the hit resolutions. Something like these tails is probably already in the resolution model (it was tuned prior to the introduction of the merging code). It might be necessary to go back and re-tune the resolution models to make sure we aren't double-counting

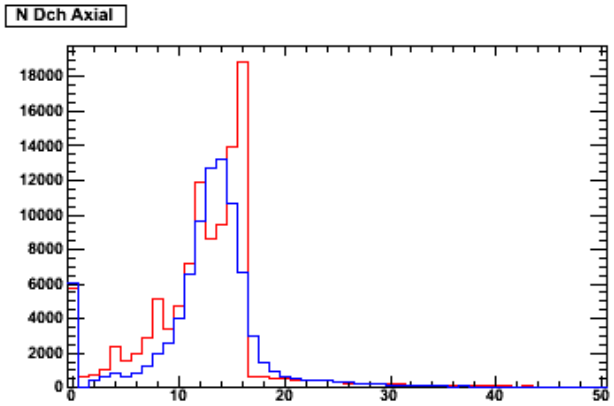
Comment on DCH Hit Efficiency

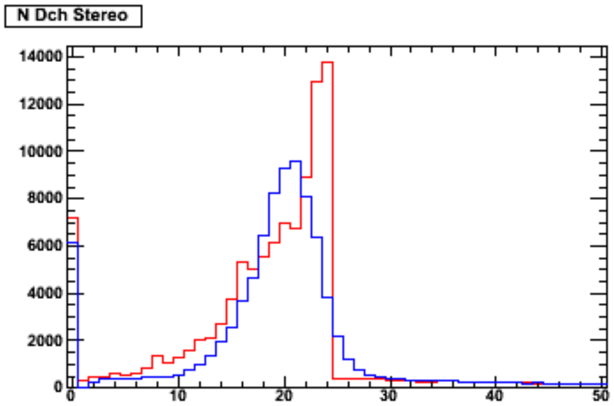
99% Eff.





97% Eff.


N Dch Axial


N Dch Stereo