

Hit Merging and PatRec Confusion in Fastsim

Doug Roberts
University of Maryland

Overview

- Goal is to add effects from merging of nearby hits and mistakes in pattern recognition into **fastsim** without actually doing full hit simulation and reconstruction or doing a full pattern recognition
- Design should work with all detector types, although some detector-specific implementation may be needed
- Should not be tuned to a specific geometry or assumptions regarding detector technology
- Should be able to deal with simulated background hits should they become available to **fastsim**

Hit Merging

- The basic idea is to emulate the effects of GEANT hit → digitization → hit reconstruction without having to create digis or do real hit reconstruction
- Hits that are close enough to each other to be reconstructed as a single hit will be merged into one hit with a position between the two original hits
 - “Close” defined by resolution of detector
- Only one track should be able to use the merged hit
 - Question: Is this really the model we want to use?

Hit Merging

- Because the merging decision only depends on the hit resolution, as opposed to the track resolution, it seemed natural to do hit merging after simulation but prior to reconstruction and fitting.
- But, this hit merging module could flag hits that should be considered for pattern recognition confusion...

Pattern Recognition Confusion

- The goal here is to simulate effects of imperfect pattern recognition in a detector implementation non-specific way
- Quality of pattern recognition should ideally depend on the resolution of the track. For example:
 - Project a seed track's full error matrix on to a detector element
 - Search for candidate hits within a road defined by this error projection
 - Choose hit candidates to add to the track

PatRec Confusion

- Because these effects are driven more by track resolution, it seems natural to put this in after reconstruction when the full track fit is available
- Hopefully only affects a small fraction of tracks so doing a re-fit would not be too costly
- Some book-keeping issues, especially if we don't want to allow tracks to share hits
 - Again, that choice makes an assumption about future full patrec capabilities

Progress

- Currently, I have written the hit merging module, `PmcMergeHits`, that runs between `PmcSimulate` and `PmcReconstruct`
- At the moment, it only deals with track hits
 - Shouldn't be too hard to expand to other detector types
 - Should be trivial to include background hits if they become available
- Still need to think some more about the implementation of the patrec confusion
 - Mostly how to do the book-keeping

PmcMergeHits

- Does a loop over all SimTracks and their SimHits
- Puts them in a sorted table (**multimap**) keyed by the detector element and measurement type
- Loops over hits in the same detector
 - Hits from same gTrack aren't considered for merging (overlaps)
- Compares hits' separation in the direction of measurement
- Constructs a χ^2 based on measurement resolution to see if the hits should be merged

PmcMergeHits

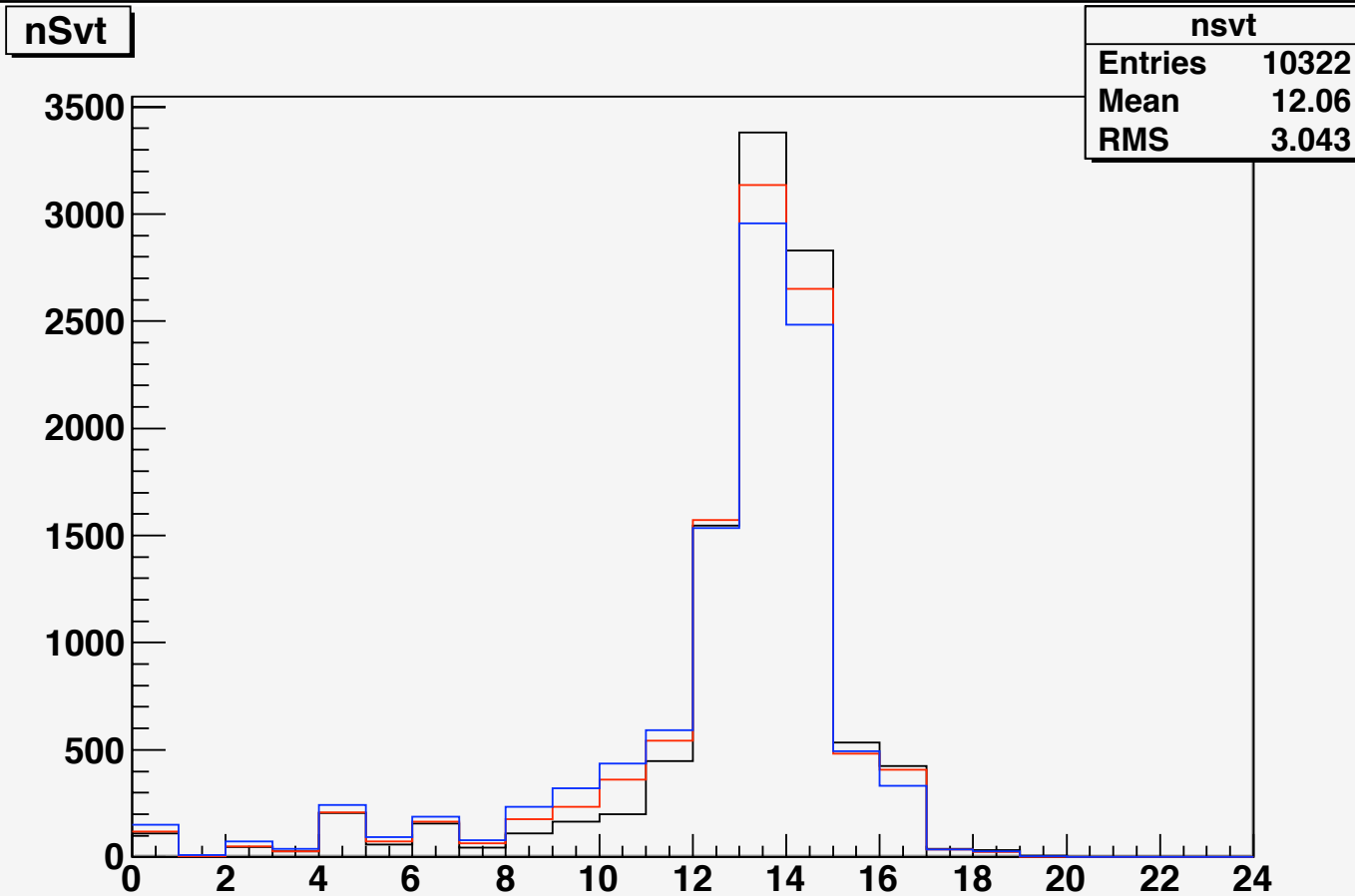
- Currently makes two χ^2 comparisons:
 - A loose cut to flag hits for later patrec confusion consideration
 - A tighter cut for the actual merging
- If a pair passes the tight χ^2 cut, a new position is determined at the mid-point along the direction of measurement (assumes resolution of two hits are the same; could be modified)
- One SimHit gets its position updated to the new position
 - Design question: This throws away the original position?
- The other SimHit gets flagged as “Don’t Use in Fit” for reconstruction
- The decision is simply based on the keeping the hit on the track with the largest momentum. Somewhat arbitrary, but assumes that patrec will have the highest success with high-p tracks
 - Open to other suggestions for this choice.
 - Could keep the modified hit on both tracks?

Some details...

- Hits in pixel detectors must pass the tight χ^2 cut in both views in order to be merged
- Hits in double-sided Si strip detectors must pass the χ^2 cut in one view but be within some distance in the other view
 - Tries to model the fact that the readout will have some granularity.
 - Currently set this distance arbitrarily to 5cm
 - This is roughly the size of a Si wafer
 - Note that this isn't fixed in the geometry but moves with the hits
 - Does not create fake hits that could arise from z- ϕ hit mixing. Hits are still basically space-points.
- Wire chamber hits with just one view straight forward

Tests

- To test the hit merging, I ran with two rather loose cuts on χ^2
 - $\chi^2 < 10$ ($\sim 3 \sigma$)
 - $\chi^2 < 100$ (10σ)
- Neither really realistic, but I expect the effect to be small and just wanted to see something
- For reference
 - With $\chi^2 < 10$, about 0.6% of hits get merged
 - With $\chi^2 < 100$, about 1.5% of hits get merged
- So far just looked at some basic things...



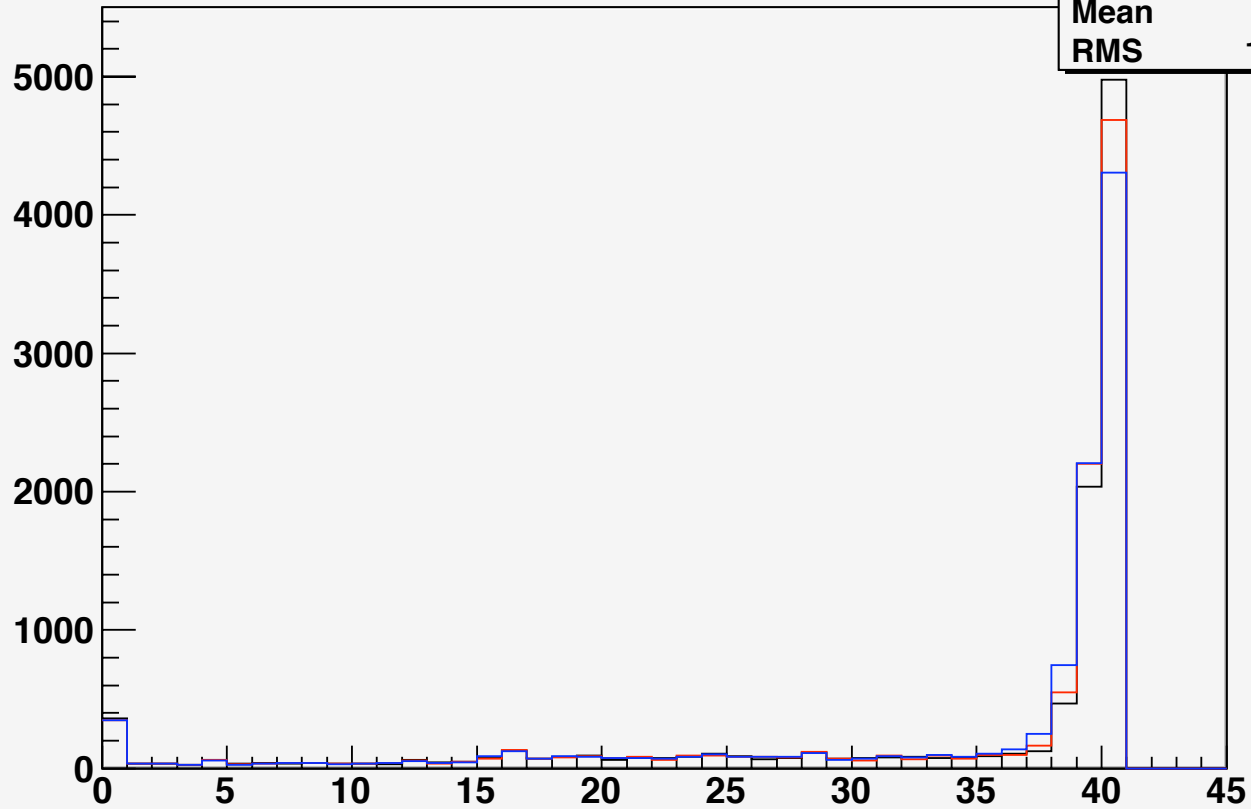
Number of hits in the Silicon

Black: No hit merging

Red: Hit merging with $\chi^2 < 10$ criteria

Blue: Hit merging with $\chi^2 < 100$ criteria

nDch



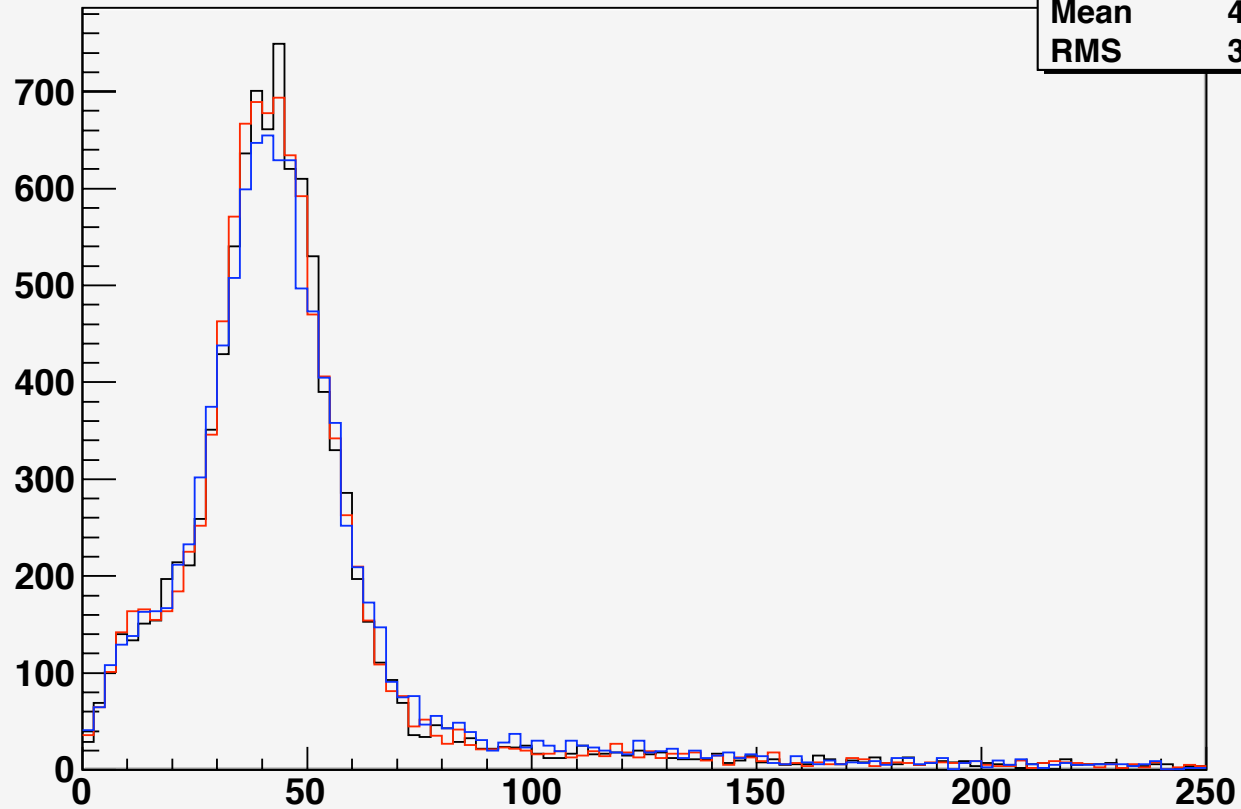
Number of hits in the drift chamber

Black: No hit merging

Red: Hit merging with $\chi^2 < 10$ criteria

Blue: Hit merging with $\chi^2 < 100$ criteria

Track χ^2



Track fit χ^2

Black: No hit merging

Red: Hit merging with $\chi^2 < 10$ criteria

Blue: Hit merging with $\chi^2 < 100$ criteria

Still To Do

- Hit merging in tracking not a huge effect
 - Could become bigger with background
- Currently really slow, but this should be fixable
- More validation
- Need to expand to other detector types
 - Would like to talk to some sub-system experts to see what would make the most sense
- Work on patrec confusion
 - This will probably be bigger effect
 - Should have happened post-reco but pre- β cand creation (I think)
 - Could move the Hit Merging to post-reco as well