Improvement on DT Segments reconstruction

Stefano Lacaprara

INFN LNL

DT-DPG

CERN, 21-Feb-2008

Outline

The problem

- Uncorrect reconstruction of 2D segments (mainly in θ view)
- The direction is sometime plain wrong, as a result of uncorrect pattern recongition

  - Will report status of my understanding and possible cure tried and to be tried
  - GREN data (run 30625) CMSSW 175
  - *caveat*: still no quantitative results, have problem in running on large sample (yet)

- The current pattern recongition actually finds both the correct and the ghost segment (and many more)
- The problem is not in pattern reco!
- Sometime the ghost suppression kills the correct segment and keep the uncorrect.
- For short segment (3 hits) the chance that a wrong segment has a χ^2 smaller than the right one is sometime non negligible.

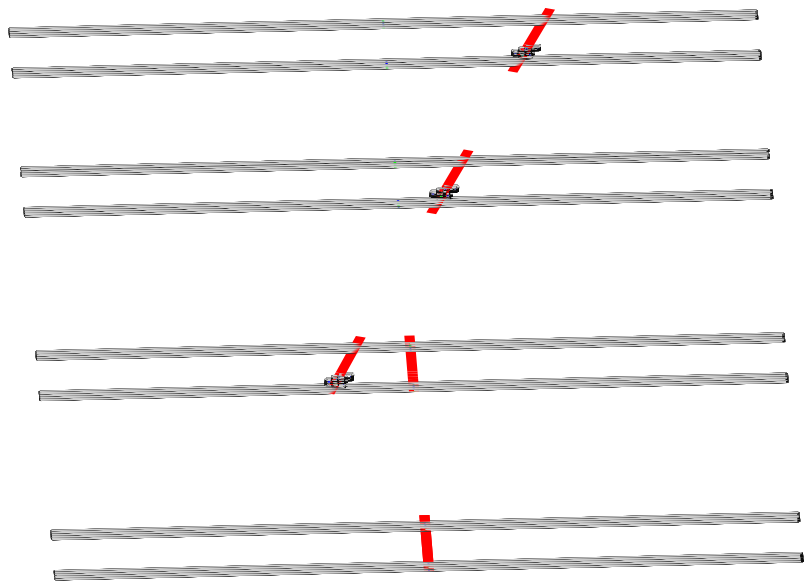
- It's possible to tune the ghost suppression and/or conflict solving (two segment shares a hit with opposite left/right choice)
- Ideally, loosening the request should results in more segments including the right one
- Unfortunately, even allowing just 1 conflicting hits in two segments allows many more segment to be retained
- don't have quantitative numbers, but I've seen events where a SL with 4 hits in 4 layers, well aligned and very clean, have up to 4 segments produced instead of one. The situation is much worse in case of delta rays or showering
- this proliferation of segments has to be solved at some higher level (track reco) with significant increase of combinatorics and timing.

Possible Solution

Try a reconstruction using simultaneously more than one station in order to increase the lever arm

- I've written a prototype for this kind of reconstruction
- reusing the same pattern recognition and ghost suppression of single SL reconstruction
- *Just* start with hits from more than a station
- Code is ready, not (yet) in CVS
- Still problems to be addressed before releasing

Possible solution



Issues

- Potentially is slower than the actual one, since the combinatorics is higher
- tuning of errors is more difficult: we cannot ignore contribution from long extrapolation and multiple scattering, as is safe to do in intra-SL extrapolation
- understanding that an hit is compatible or not with a segment hypotesis is more complex
- which SL should we try to reconstruct togheter? Now starting with same sector (for the prototype) but it's clearly not enough
- for $p - p$ interaction different wheel has to be considered.
- Will try ± 1 wheel and ± 1 sector, but that means a sizeable fraction of the whole detector
- how to avoid multiple reconstruction while covering whole detector?

Issues: II

- I do want to use these *"long segments"* for 4D segments as well as for StandAlone reco
- To do that, the segments have to be limited to a chamber
- how to split a long segment to smaller ones, one in each SL while retaining the direction information of the long one?
- create one small segment for each SL which has hits contributing to the long one, with just the hits in that SL but position and direction of the long segment
- problem: we can have *e.g.* 2 or even just 1 hit in a SL: can we have segments with 1 or 2 hits?

Other ideas

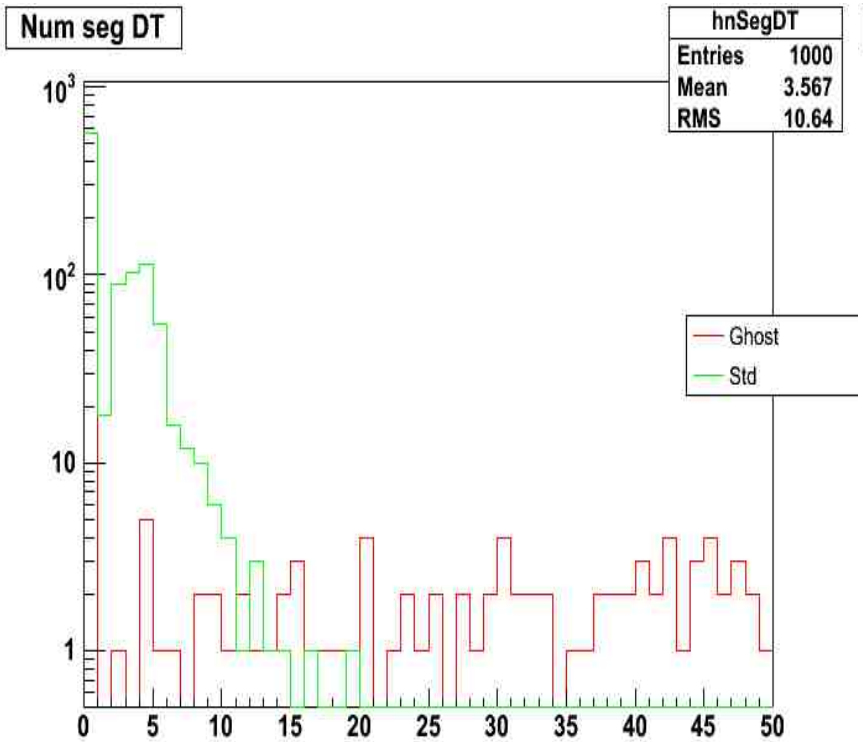
- Other possibilities is to do the reconstruction in just one SL at a time (as we do now), but use the information from other chamber to *bias* the segment reconstruction toward the *right* one.
- Clusterize (*á la* calorimetry) the hits in each θ SL, to get a centroid and a width.
- For each SL, use its hits plus the clusters from other close SL
- then build the segment retaining only the real hits
- advantages: limited increase of combinatorics (many hits, few clusters), long lever arm
- issues: as before, which other chamber should we consider? Need for a *chamber* pattern reco?
- In general: what about overlap region?

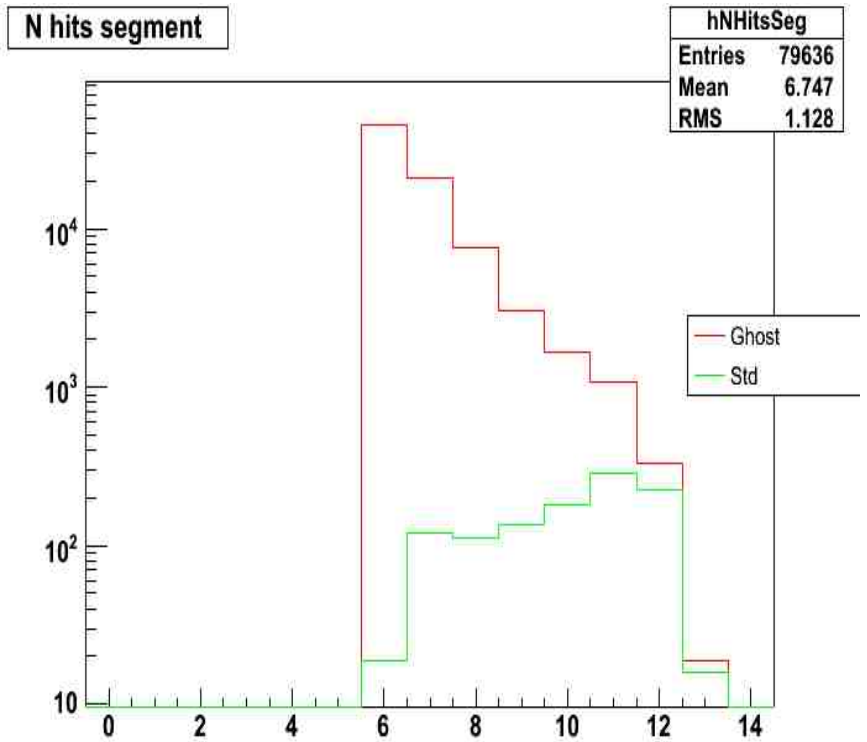
Outline

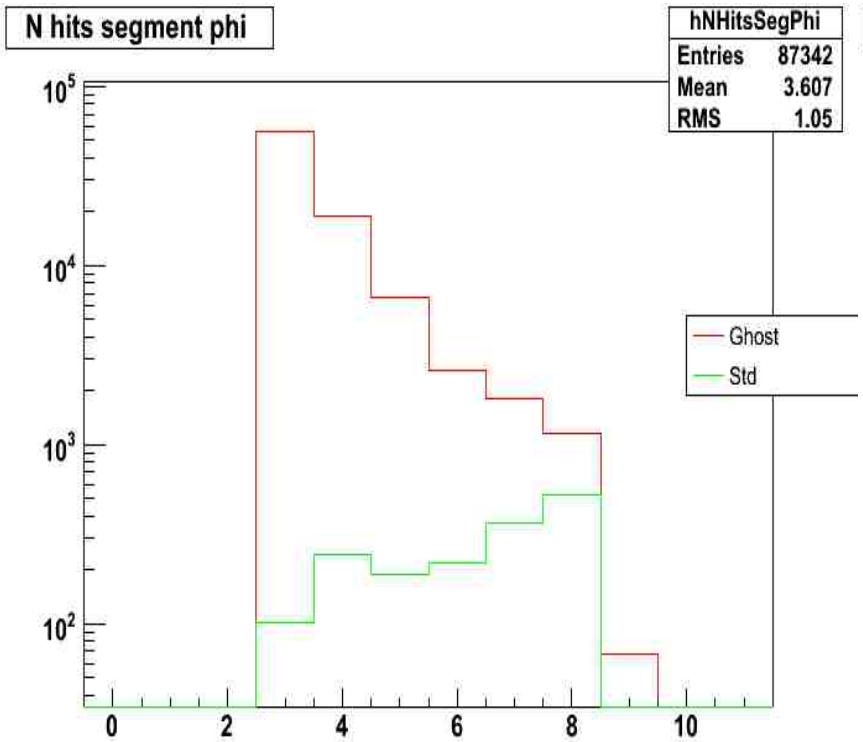
Need for Speed

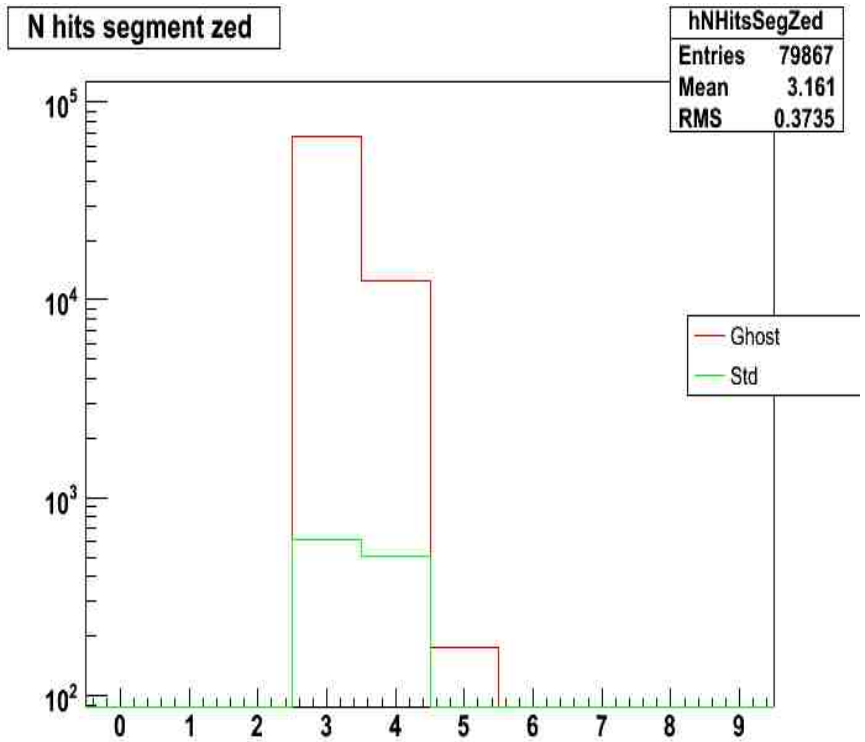
Some issues raised lately about timing of segment reconstruction. How to improve things.

- First optimization of the code it try to reduce the number of segments tried.
- we do reconstruct in mean $N = 1.7$ segments per event (forget about absolute number, the important is the ratio)
- but we also create $N = 3.57$ segments most of which we reject during ghost suppression
- and we also try many more hits combination which are rejected because are too bad (too few hits and poor χ^2)
- It's difficult to reduce the ghost, we can do something to optimize the latter item


Num seg DT
hnSegDT
Entries 1000
Mean 3.567
RMS 10.64
Ghost
Std
10^3
10^2
10
1
0 5 10 15 20 25 30 35 40 45 50


N hits segment
hNHitsSeg
Entries 79636
Mean 6.747
RMS 1.128
Ghost
Std
10^4
10^3
10^2
10
0 2 4 6 8 10 12 14


N hits segment phi
hNHitsSegPhi
Entries 87342
Mean 3.607
RMS 1.05
Ghost
Std
10^5
10^4
10^3
10^2
0 2 4 6 8 10


N hits segment zed
hNHitsSegZed
Entries 79867
Mean 3.161
RMS 0.3735
Ghost
Std
10^5
10^4
10^3
10^2
0 1 2 3 4 5 6 7 8 9

- now we construct the hits collection using (almost) full combinatorics.
- we do not considered hits not compatible with initial segment hypotesis (good)
- but we first build a segment (fit included) and then we check that it was already tried and throw it away (very bad!)
- this happen quite often
  - eg typical clean situation with 4 hits in 4 layers
  - start with hits in layer $1 + 4$ and get them all
  - start with hits in layer $1 + 3$ and get them all again
- I've introduced a check to recognize the same hits pattern is already tried and stop immediately.
- Some more test then I'll commit

- other optimization are possible
  - start pattern reco only if hits in 3 different layers are present (not done now, bad!)
  - consider only hits which are "not so far away" (*e.g.* ± 2 wires): already done, good. Can be tuned
  - cache the extrapolation of segment hypotesis to each layer? Just linear extrap, not clear if it is worth
  - ...
- More than any other thing we need a serious profiling, to understand where we spend out time
- I've done something (`cachegrind`) but had not time to analyze results in details. Help is very welcome!

Outline

Conclusion

Reconstruction improvement

- trying a multi SL reconstruction in θ
- code prototype ready, need to tune and analyze results
- some isseus already found

Speed

- trying some trivial yet possible effective optimization
- reduce the number of segments tried
- need proper time profiling

Proposal

Set up a dedicated task force to coordinate and address these issues: I'm volunteering for leading/contributing.

Results

MTCC Run 4446 (DT trigger with $B = 0$ T)

hitsMult

Results

MTCC Run 4398 (RPC trigger with $B = 0$ T)

hitsMult