Status of Imperial tasks

Paul Dauncey

Imperial tasks

- Simulation updates and production - Paul
- Bad pixels, configuration and threshold – Paul
- Efficiency, 2D method – Paul

- Simulation
  - Nothing done since last meeting ☹
- Bad pixels
  - Have first order list of bad pixels for each run
  - Some simple software to handle the information
- 2D efficiency
  - Efficiency vs impact position of track relative to pixel
  - Some basic results on this

Bad pixels

- Based on masking, bad configuration columns, bad pedestals
  - Does not include full memory flagging
- Complicated by sensor configuration having only destructive readback
  - Only know configuration is bad after run finishes
  - Need to find pad pixels before analysis job
  - If run crashes, no check at all
- Selection of bad pixels/columns/sensors
  - Pixel: masked, trim=0, or wrong on readback
  - Column: >100 pixels with trim=0, trim=31 or wrong on readback
  - Sensor: no runEnd readback
- Results stored in files in data/pxl/
  - One file per run and layer; single bit per pixel
  - E.g. Run447790Layer0.pxl

Using the bad/good pixel lists

- Do an rsync from the Imperial data area to get the data/pxl/ directory
- Define the objects to contain the lists

```
MpsGoodPixels mgp[6];
```

- At runStart, read in the list files

```
for(unsigned layer(0);layer<6;layer++) {
   mgp[layer].readRunLayer(runNumber,layer);
}
```

- Find number of good pixels in a layer

```
unsigned gn=mgp[layer].goodNumber();
```

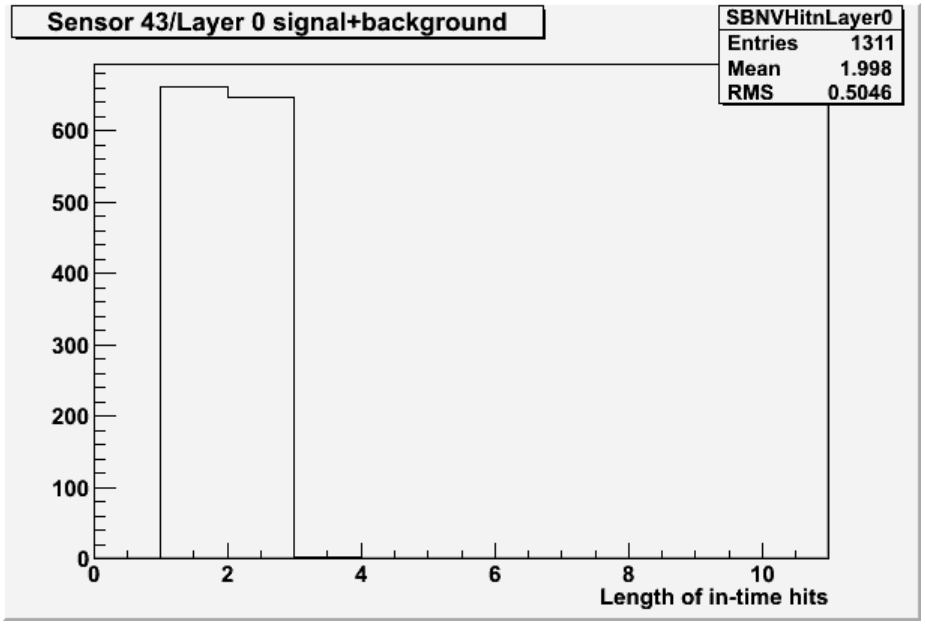
- For any pixel x<168 and y<168

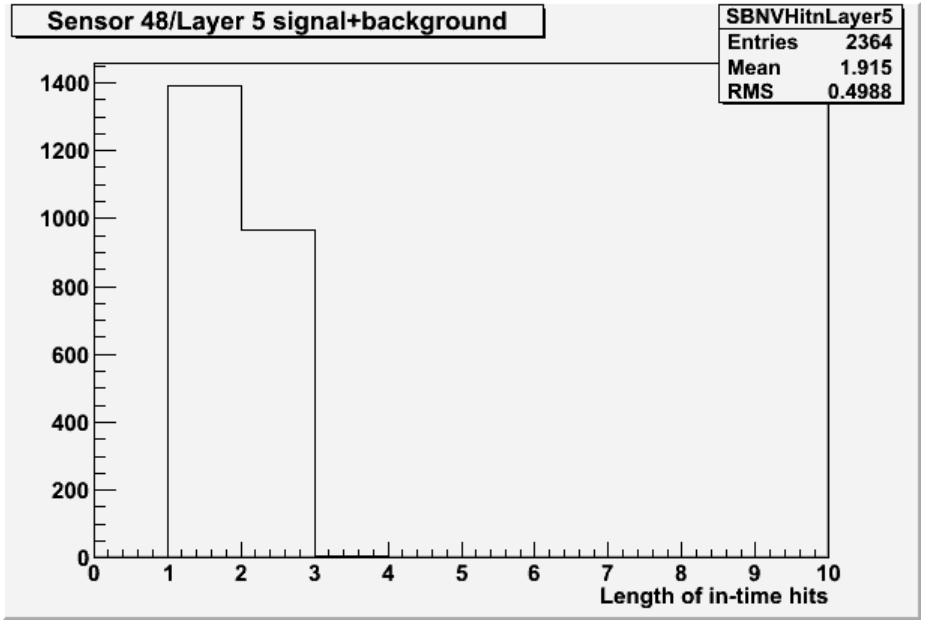
```
if(mgp[layer].good(x,y)) {
   // Use for analysis
```

- Check daquser/inc/mps/MpsGoodPixels.hh for other useful methods

Aside on monostable lengths

- Checked number of contiguous hits in time for each pixel
  - See high rate of pixels with more than one hit
  - Disagrees with Benedict's study; needs to be cross-checked



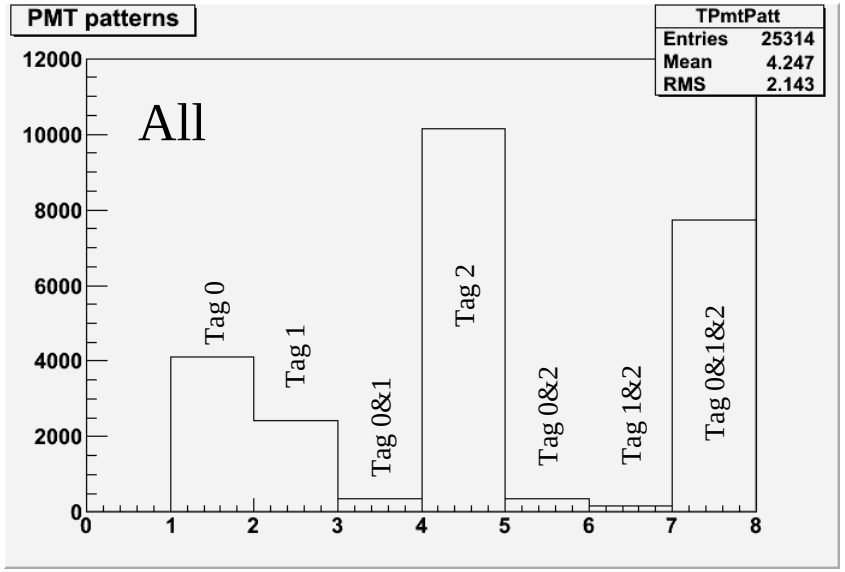


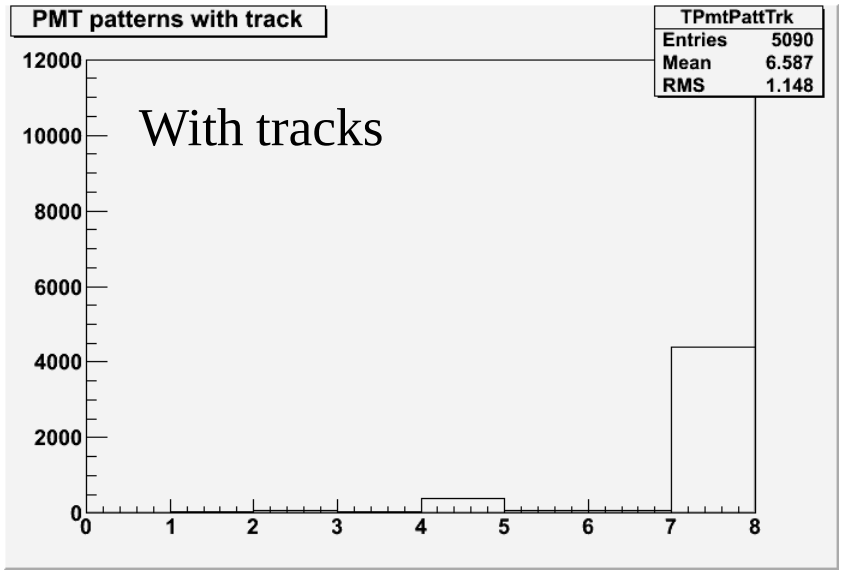
- N.B. Quick check so not systematic
  - Only checked for hits within ±1 of PMT time
  - Contiguous hits could be longer

2D efficiency

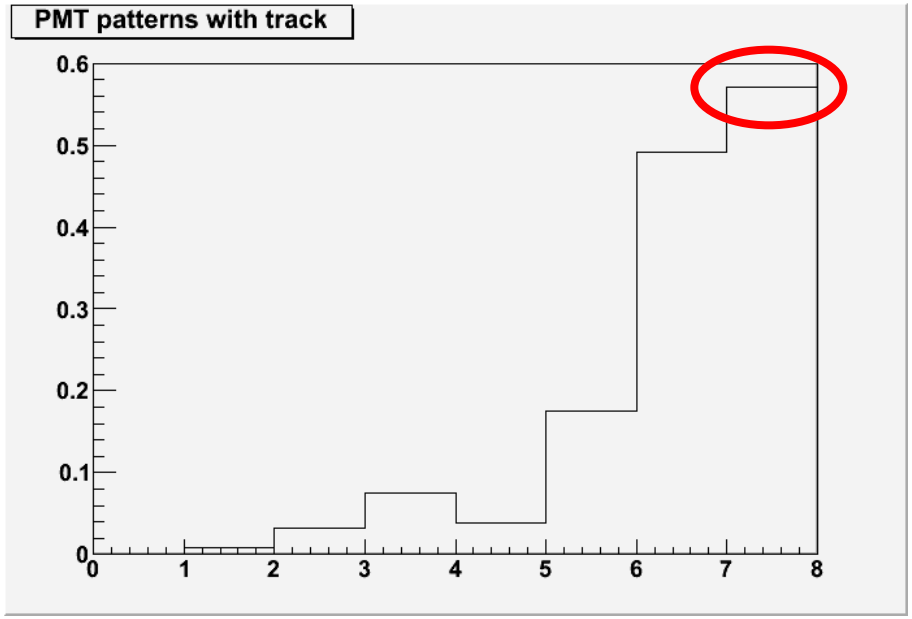
- Basic concept
  - Form a track from all layers except the one under study
  - Project track onto layer under study
  - Find position of track projection relative to each good pixel (within 7×7)
  - Plot number of hits in good pixel as a function of position
  - Divide by track position plot to get efficiency as a function of position
- Need to check track quality
  - Badly reconstructed track will not project to right position on sensor
  - Gives artificial inefficiency; not yet tackled this
- Construct "best" track from all combinations of all hit in all layers
  - All tracks required to have at least 3 layers and fit χ^2 probability > 0.1
  - Always pick track with highest number of layers
  - Pick highest probability if multiple tracks with highest number of layers
- Repeat selection for all tracks with each layer excluded in turn
  - Used for efficiency estimate

Tracks dependence on PMT hits





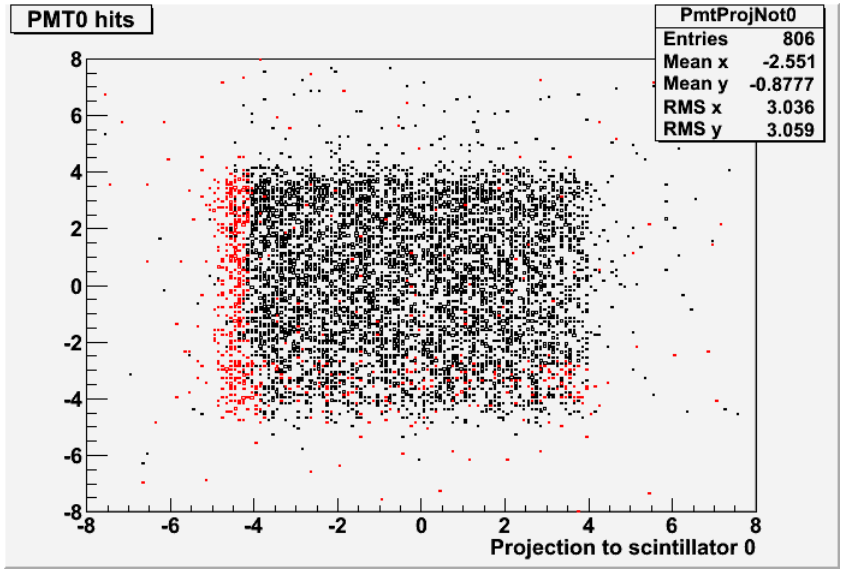
- PMT output clean; use all singles and coincidences to try to find tracks

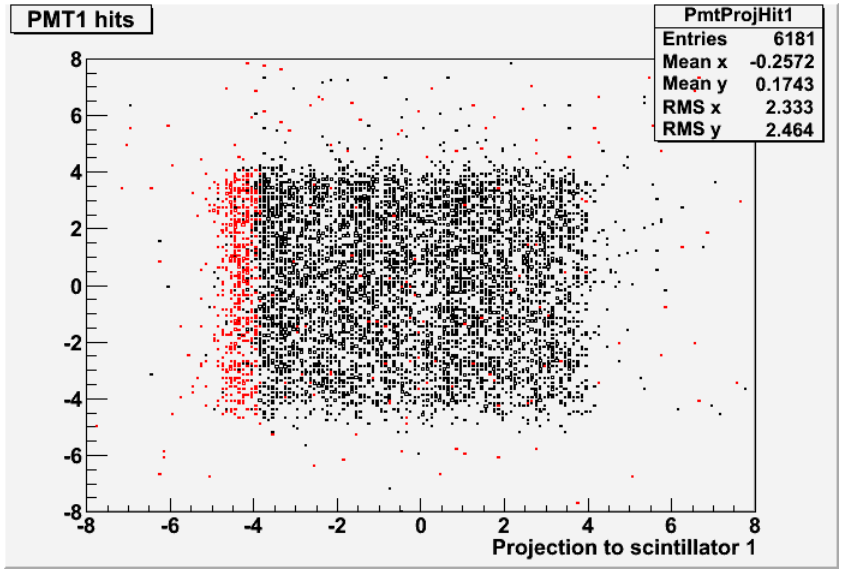


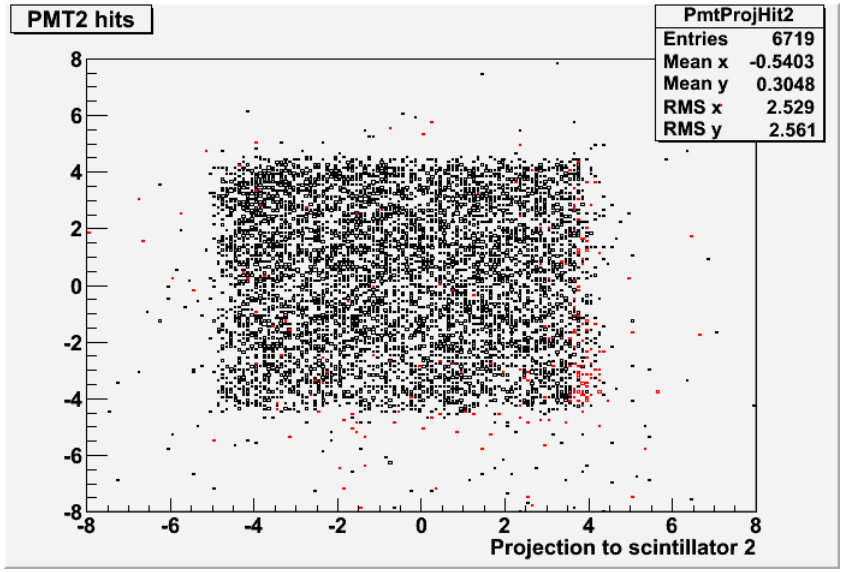
~58%

- Probability of finding a track given a particular pattern of scintillator hits

Scintillator positions

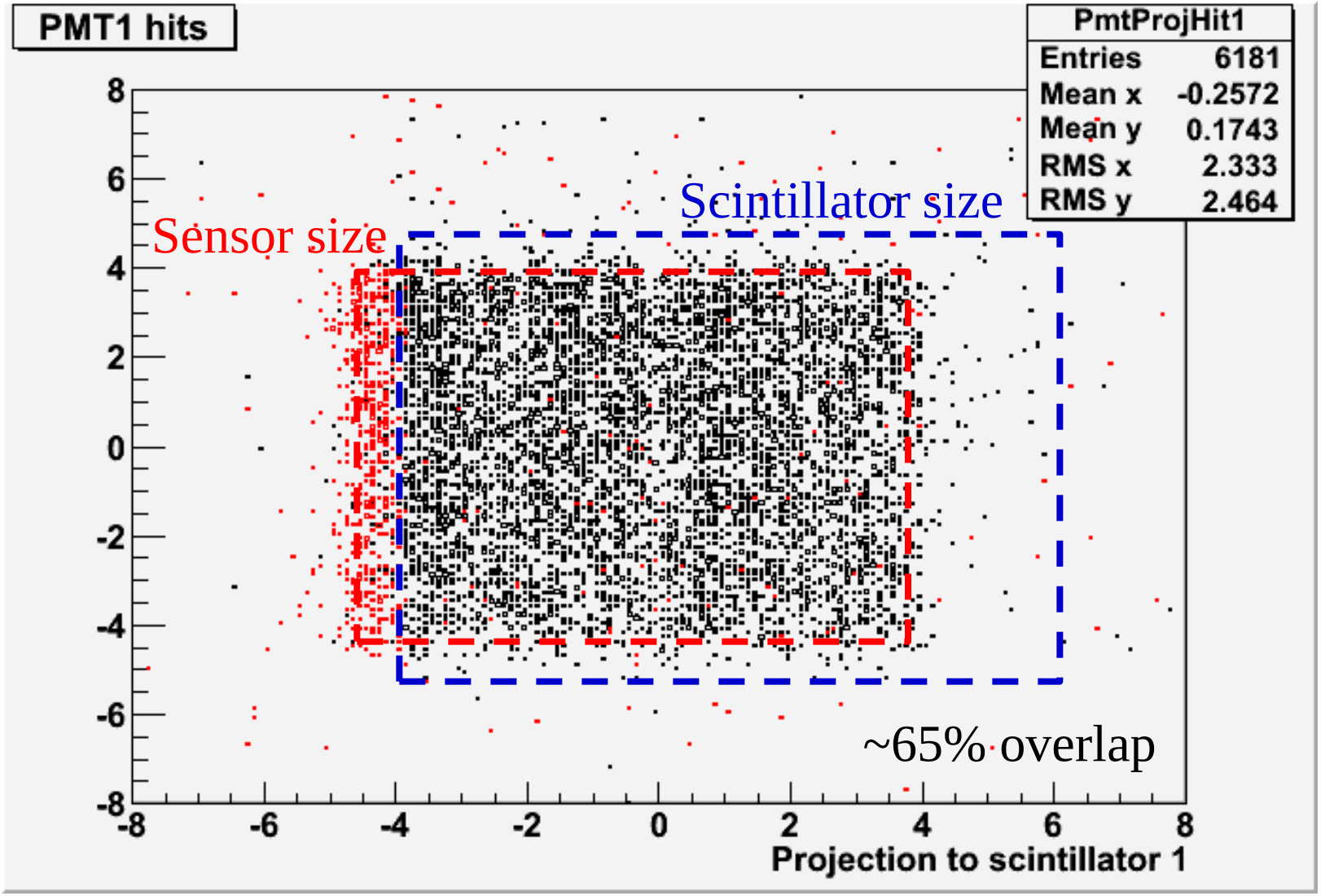




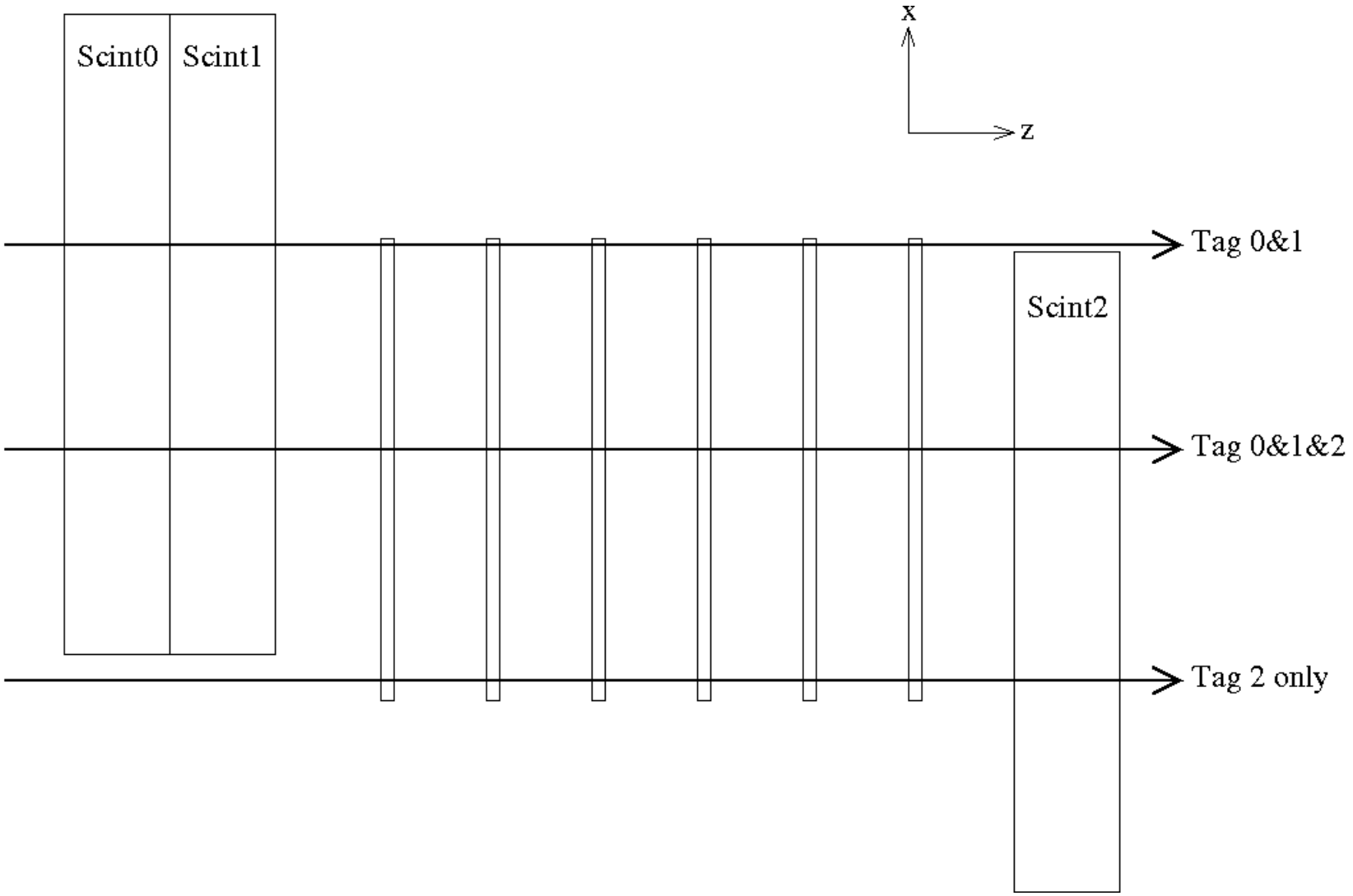


- Project tracks into plane of scintillators
- Black points are if scintillator gives tagging hit, red if no tagging hit
- Clear edge of scintillators 0 and 1
- Possible edge of scintillator 2

Scintillator-sensor overlap

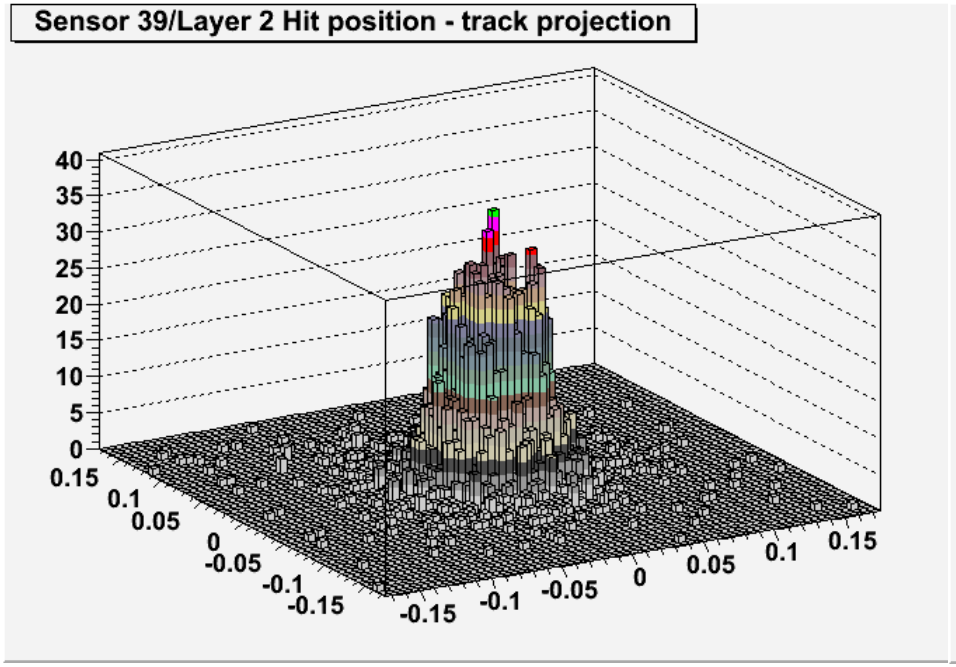


Cartoon of scintillator geometry

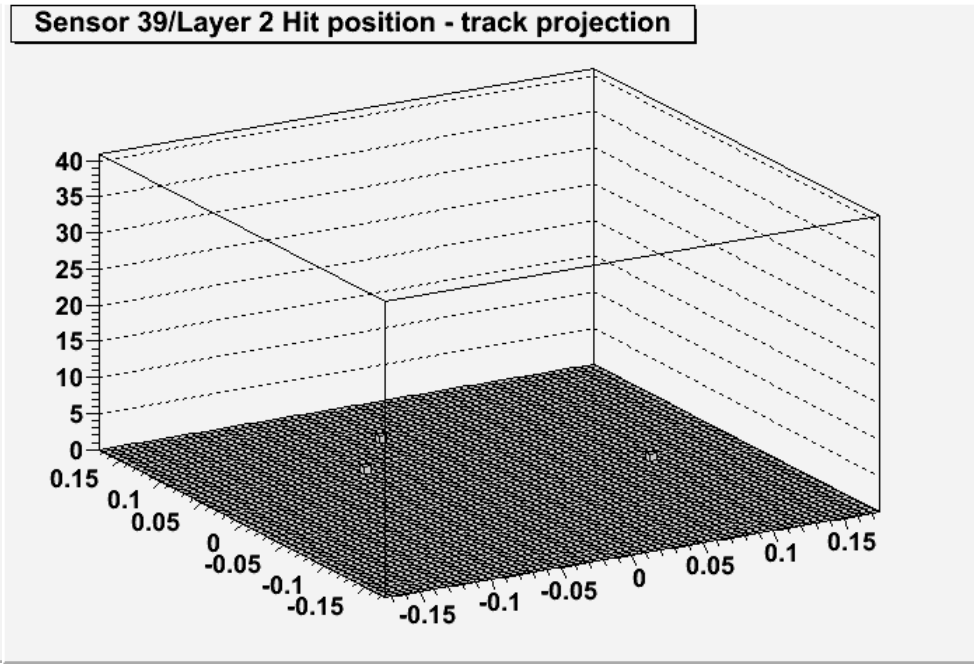


Hits in and out of time

- Typical run 447794, threshold 170
  - Number of sensor hits as a function of track position w.r.t. pixel centre
  - Plot is for 7×7 pixel array = ±175μm

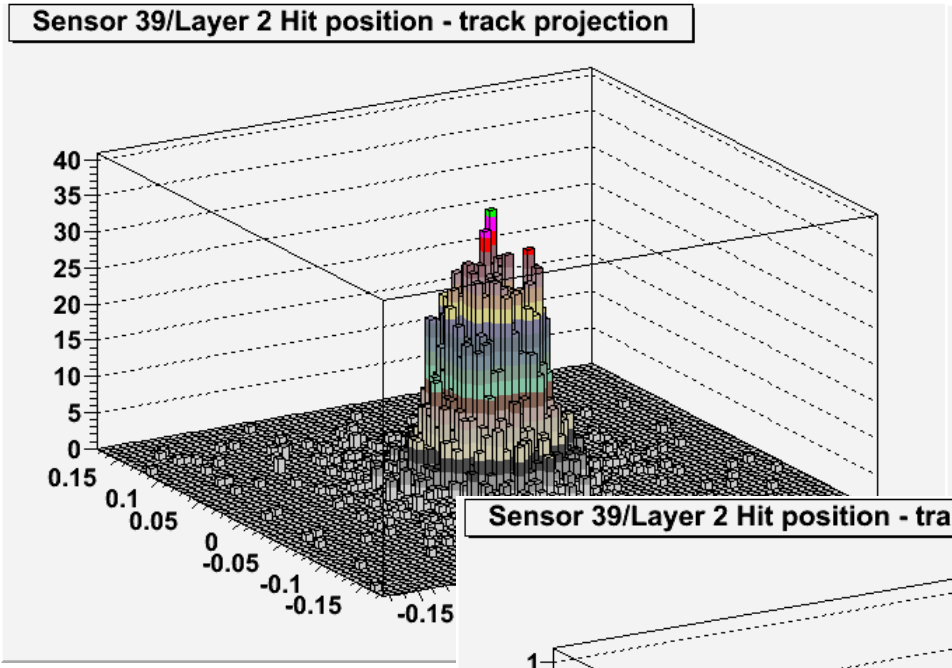


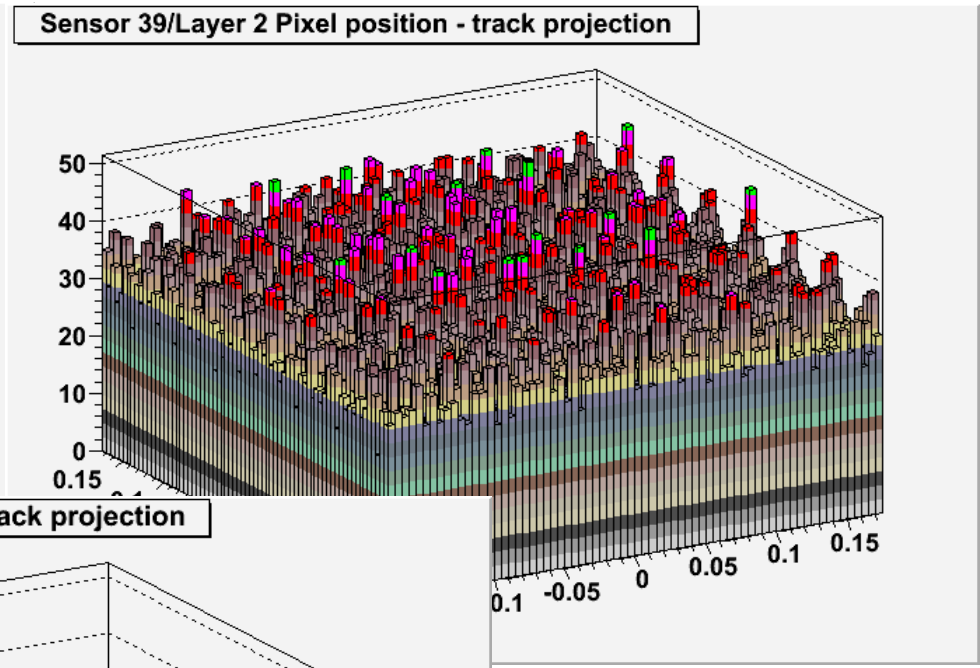
In-time

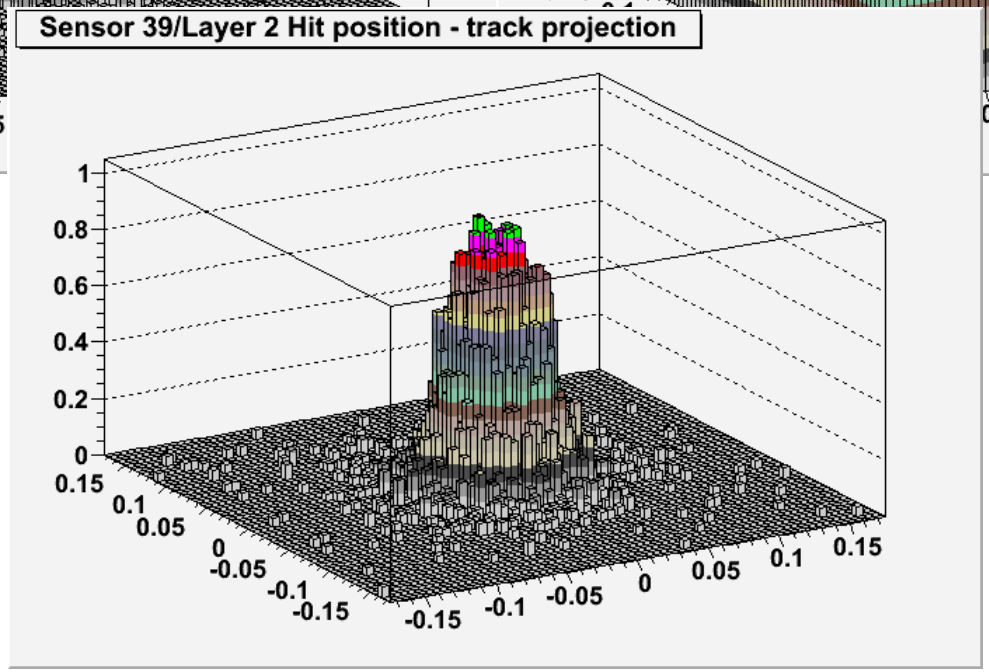


Out-of-time

Track distribution

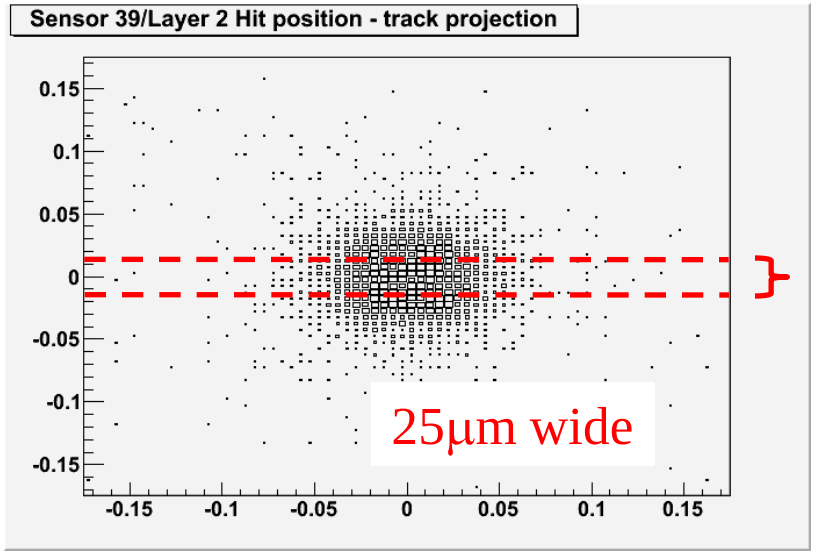




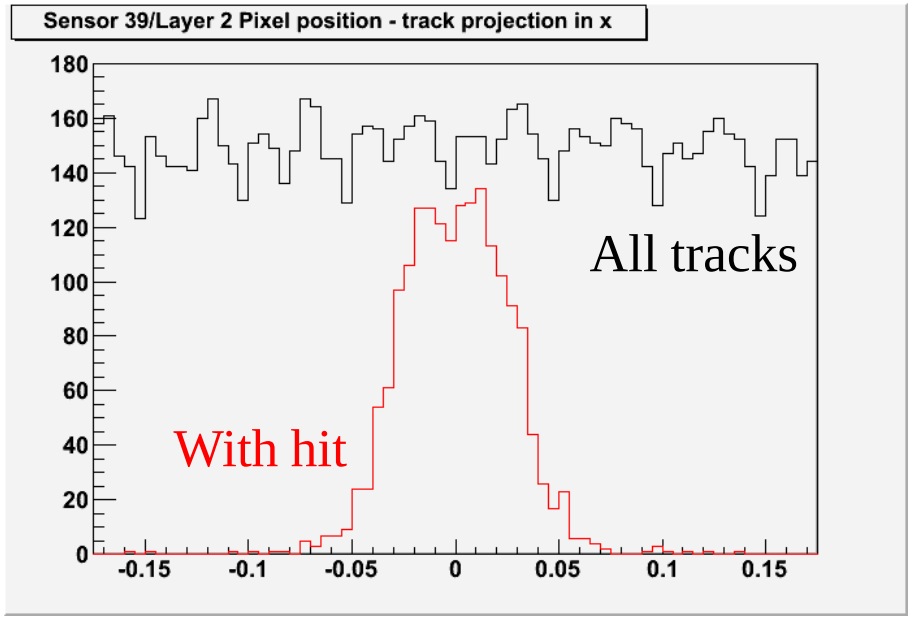


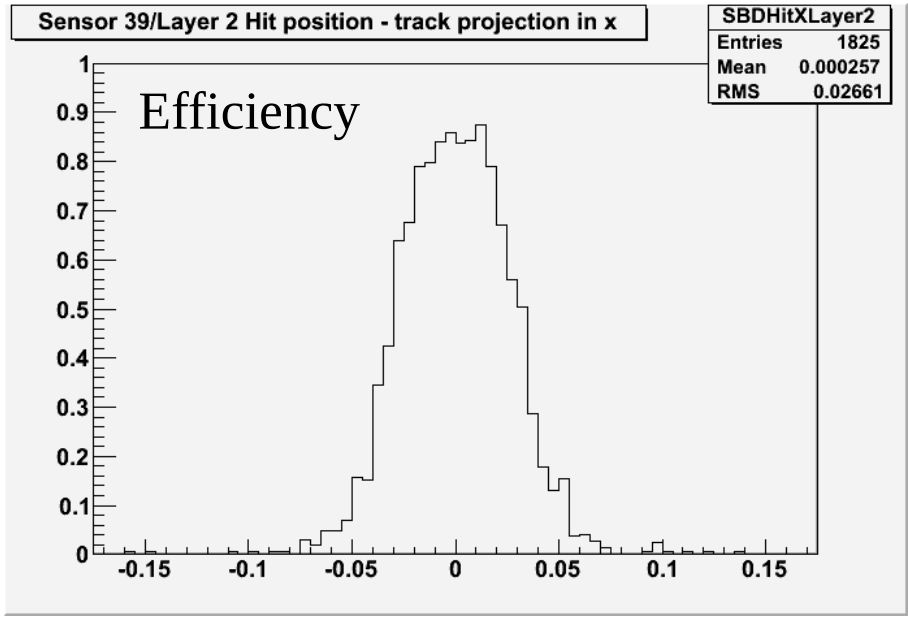
Division gives efficiency

Projections in x and y



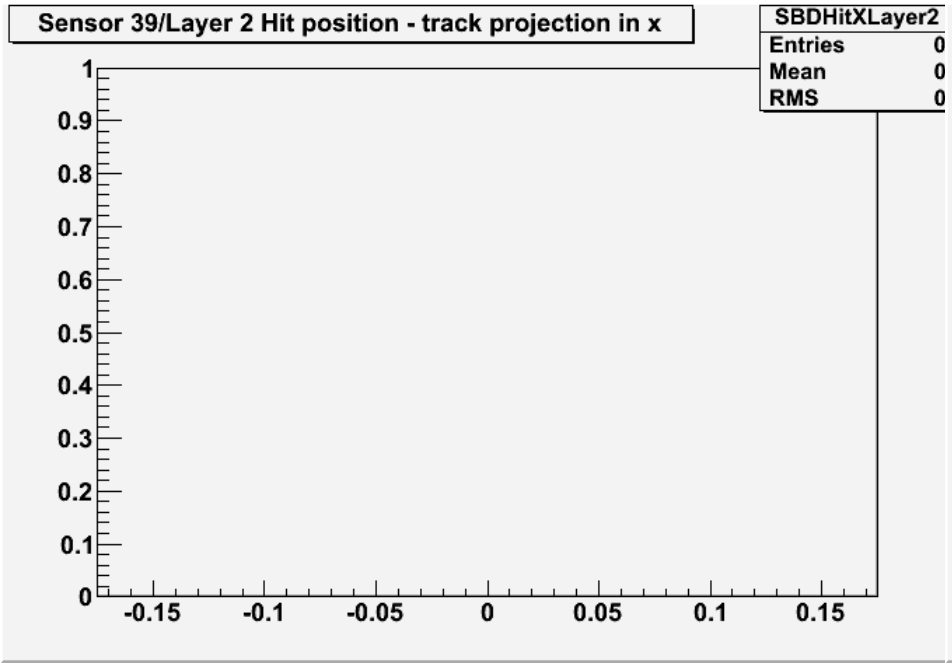
Keep

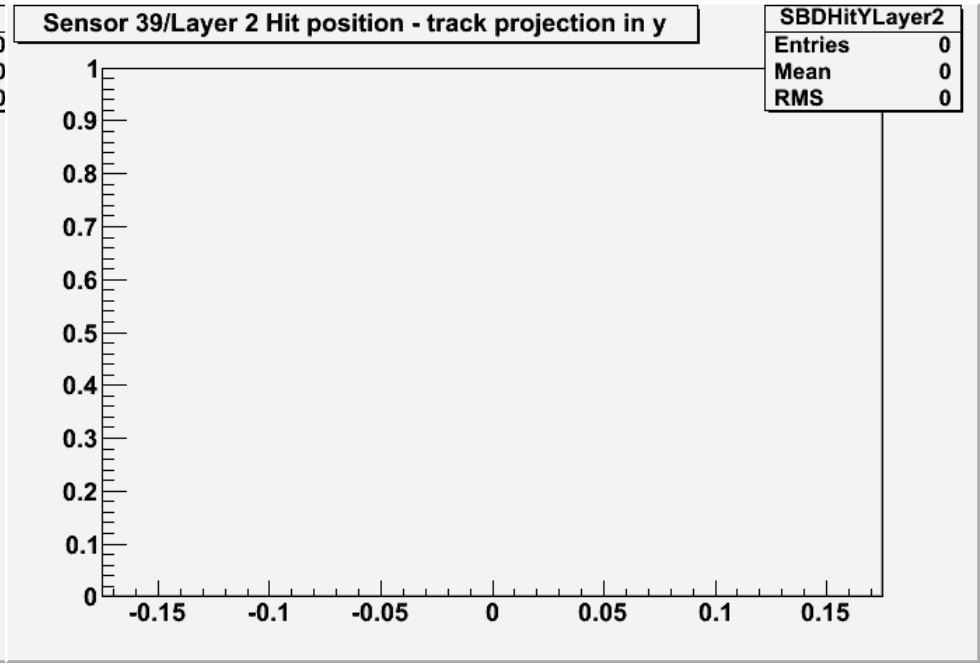




Projections in x and y vs threshold

- Run 447790, threshold 130

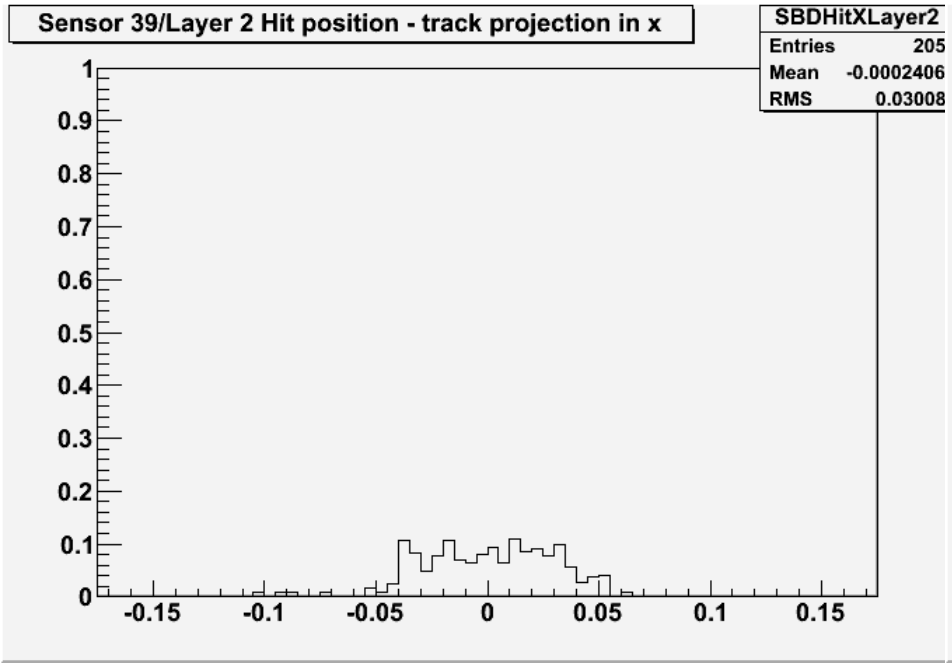


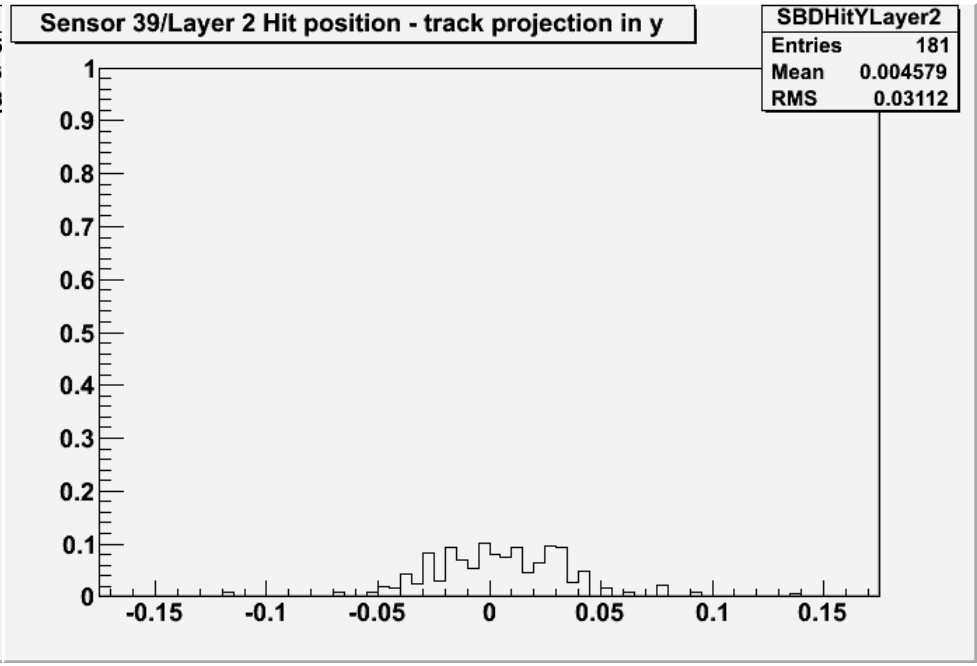


No hits at all!

Projections in x and y vs threshold

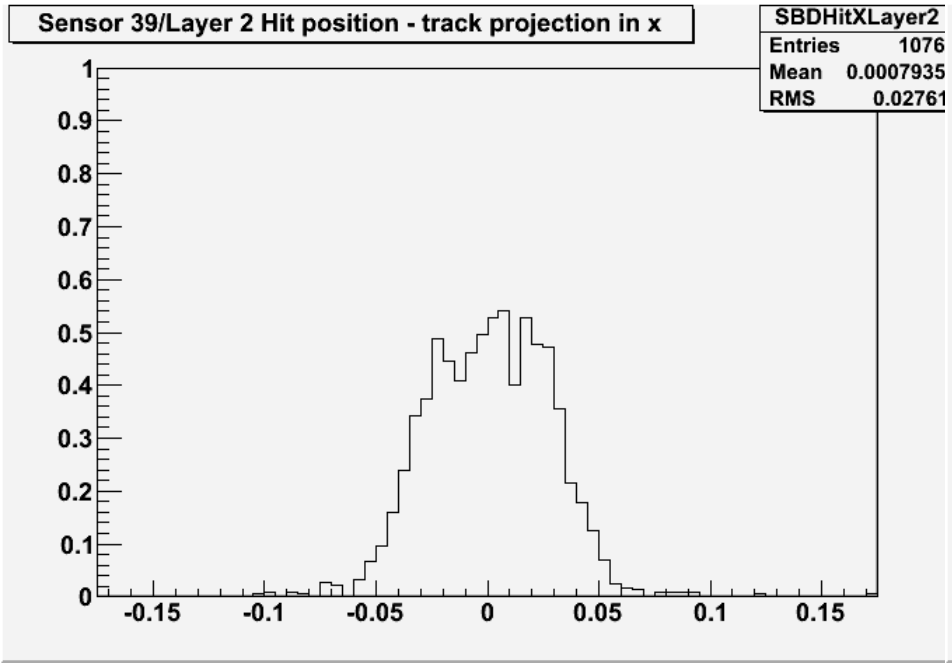
- Run 447789, threshold 140

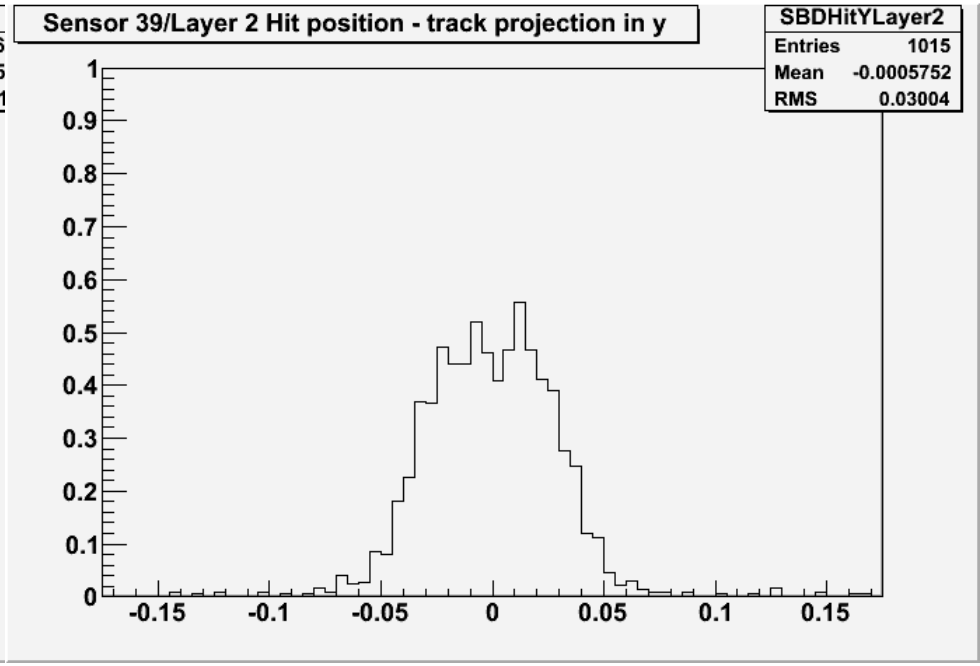




Projections in x and y vs threshold

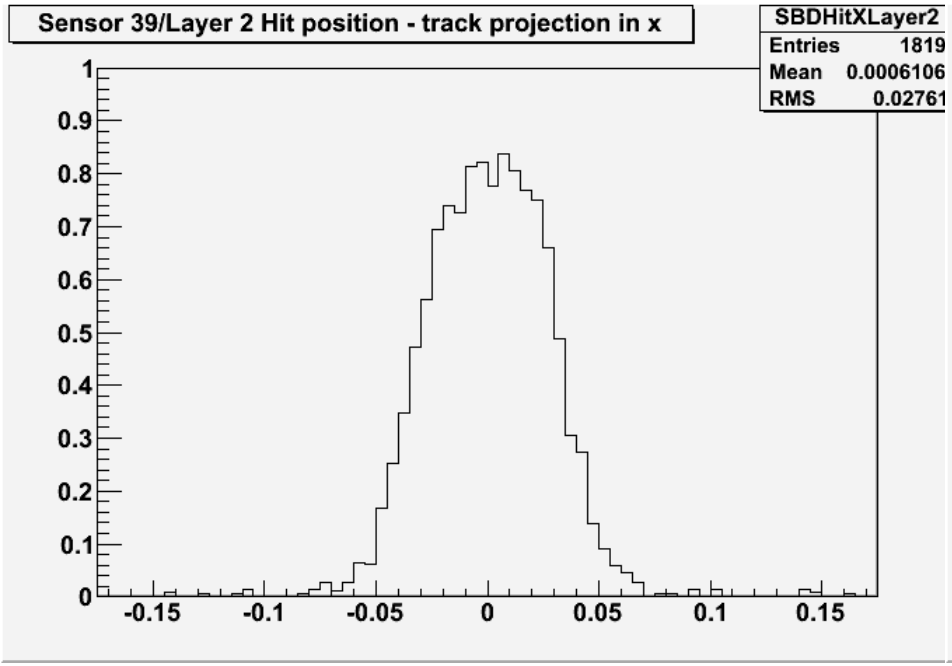
- Run 447788, threshold 150

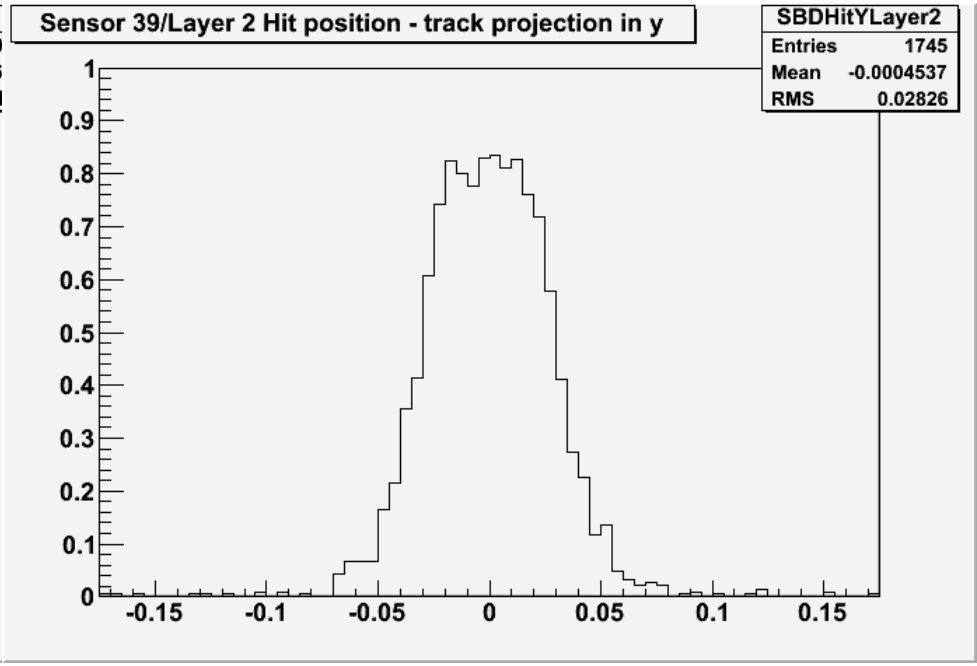




Projections in x and y vs threshold

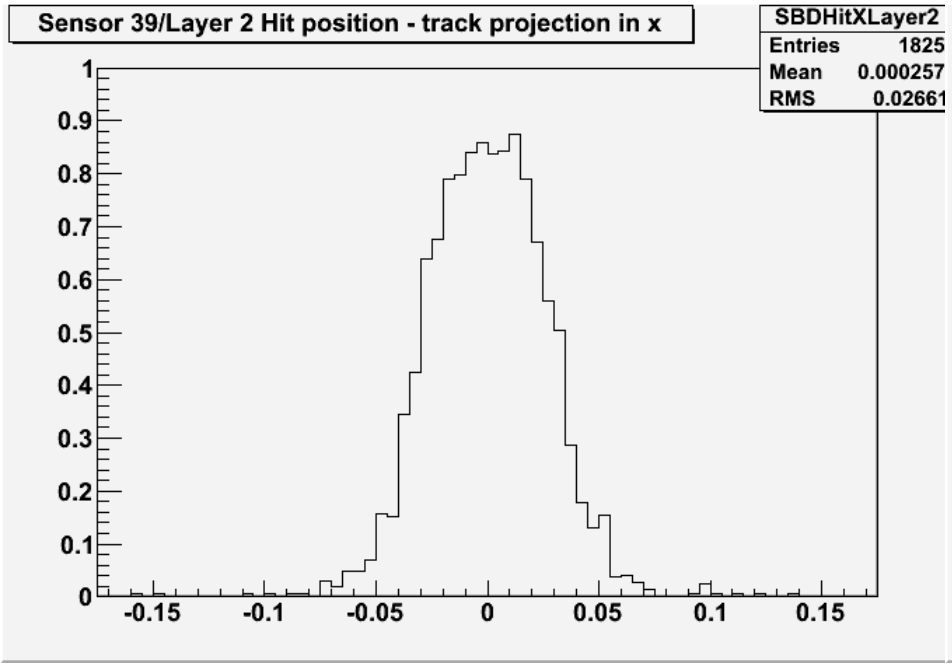
- Run 447787, threshold 160

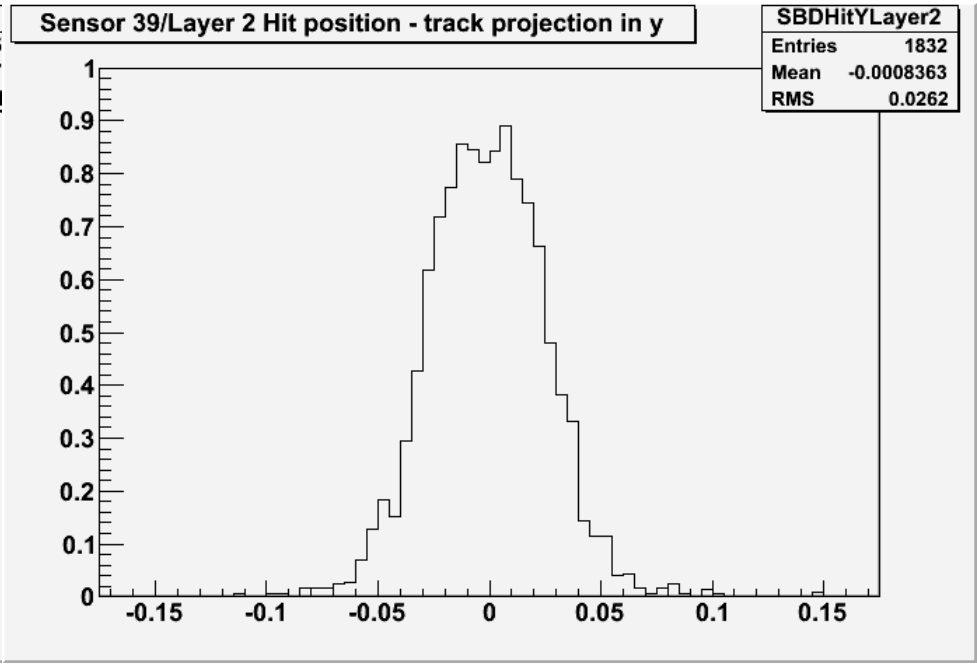




Projections in x and y vs threshold

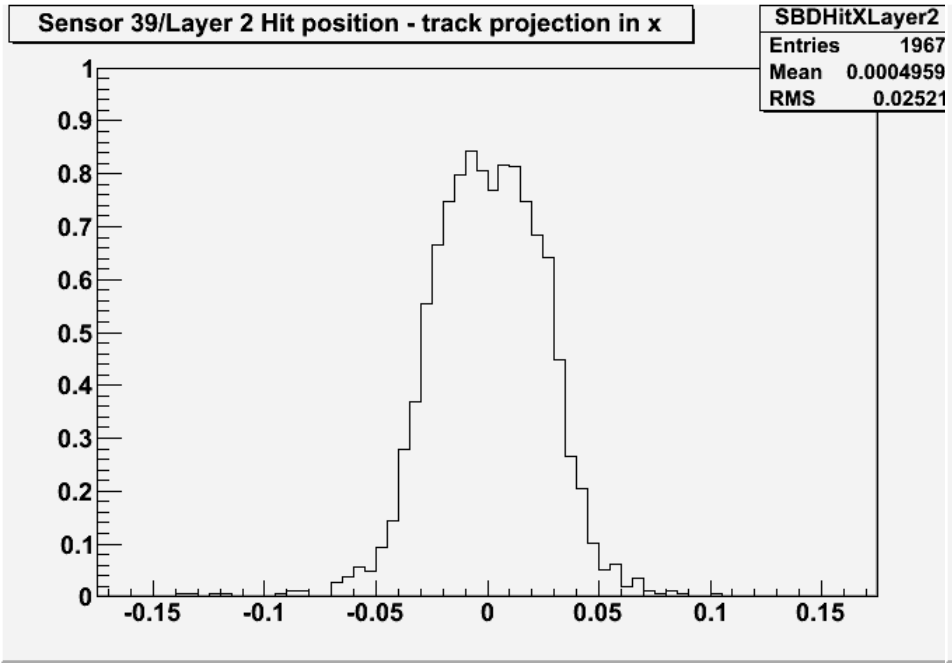
- Run 447794, threshold 170

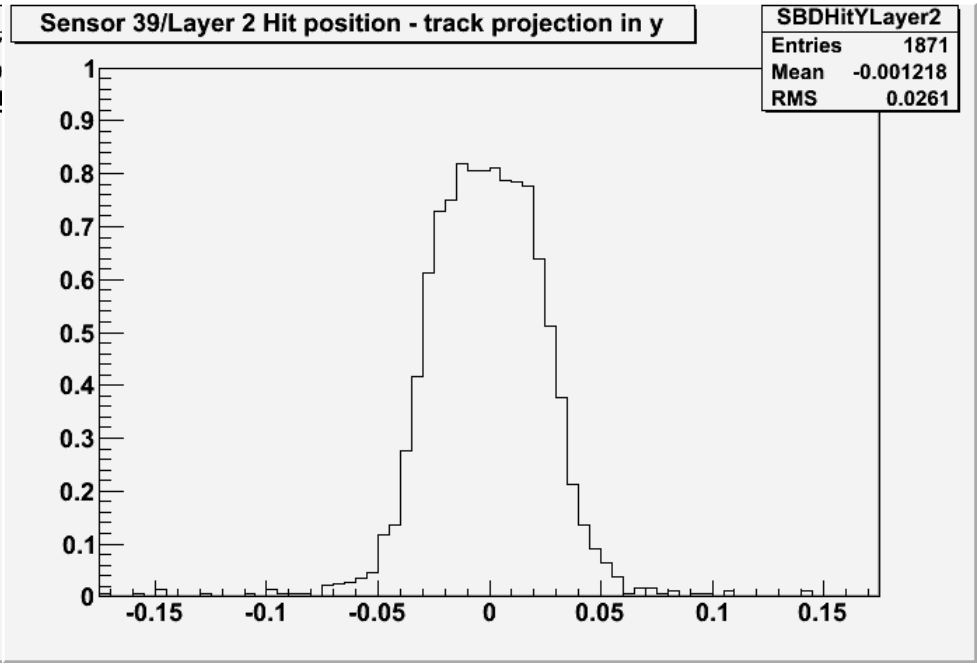




Projections in x and y vs threshold

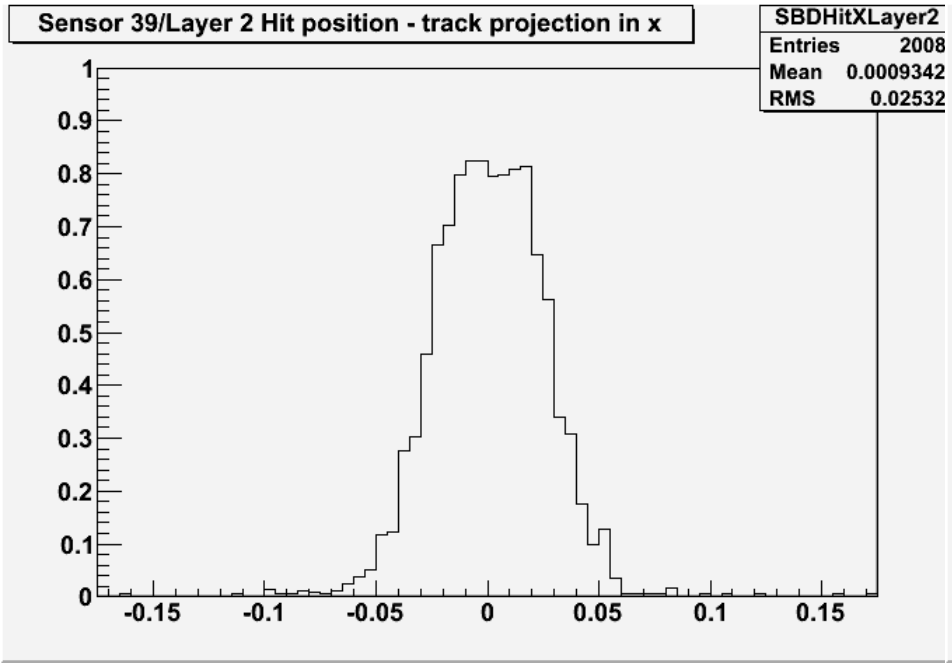
- Run 447793, threshold 180

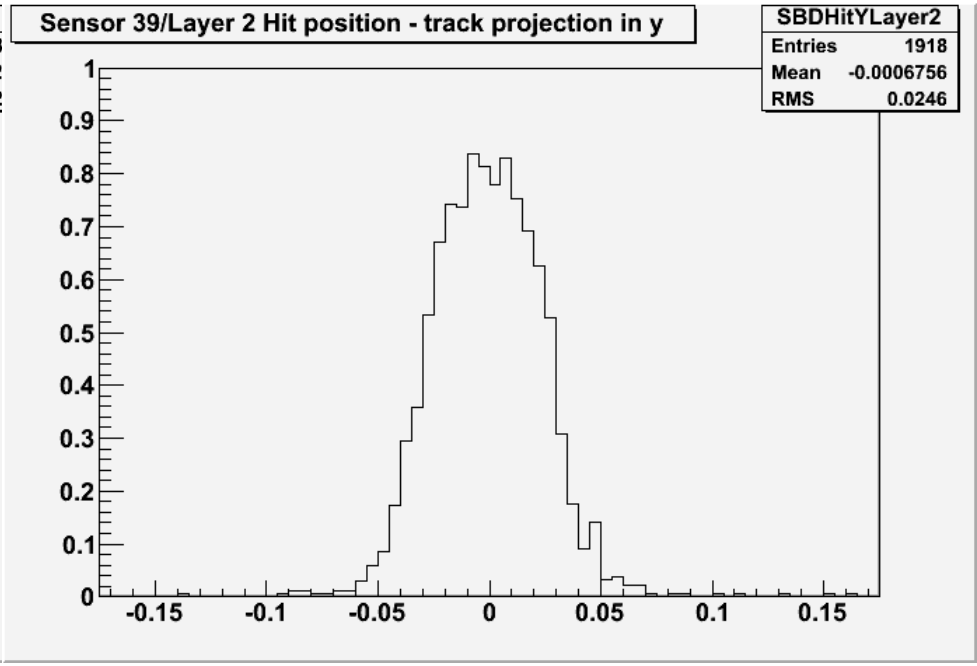




Projections in x and y vs threshold

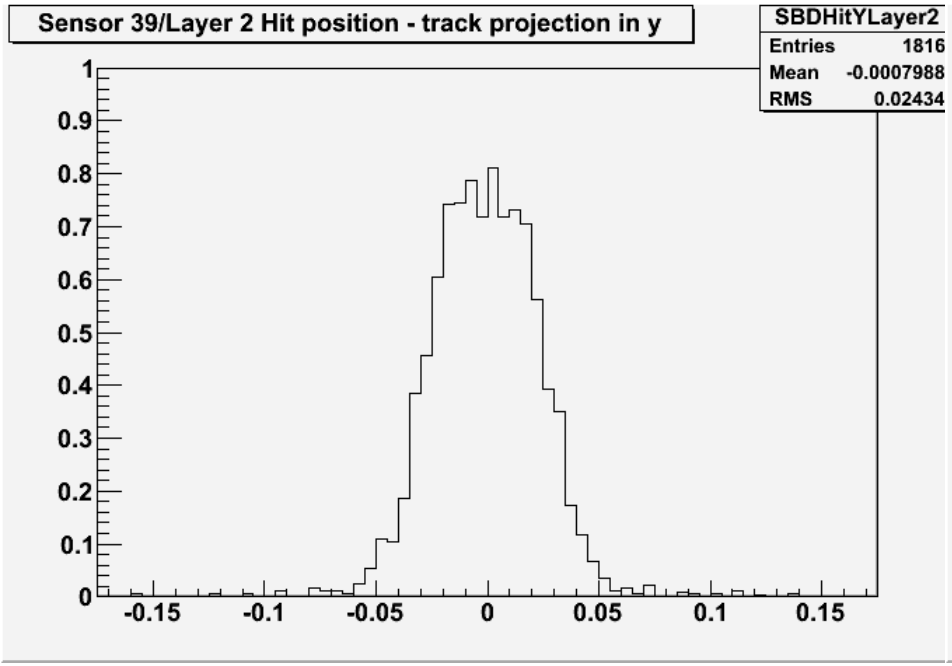
- Run 447792, threshold 190

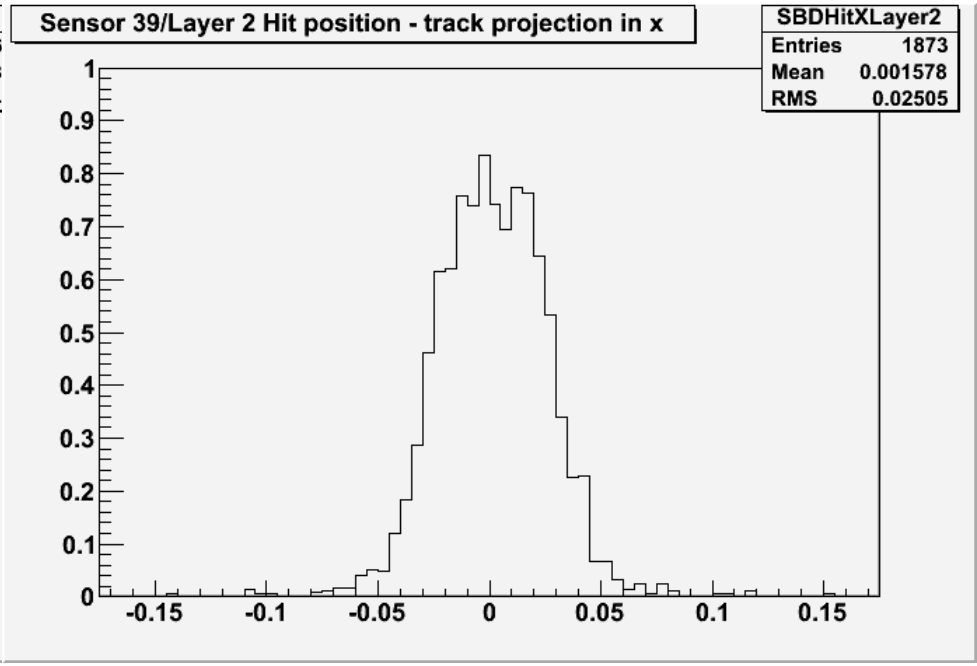




Projections in x and y vs threshold

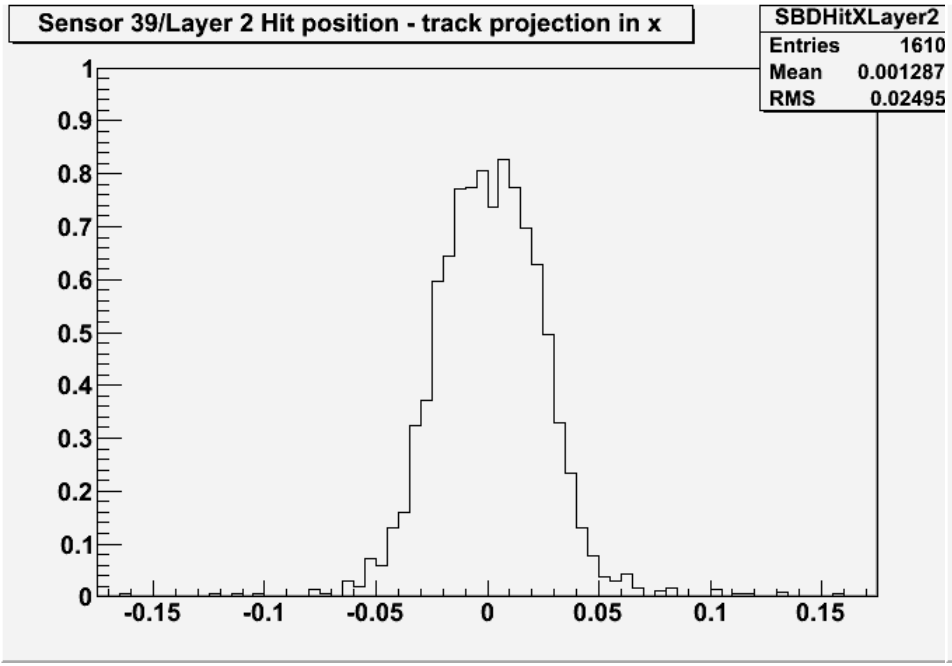
- Run 447791, threshold 200

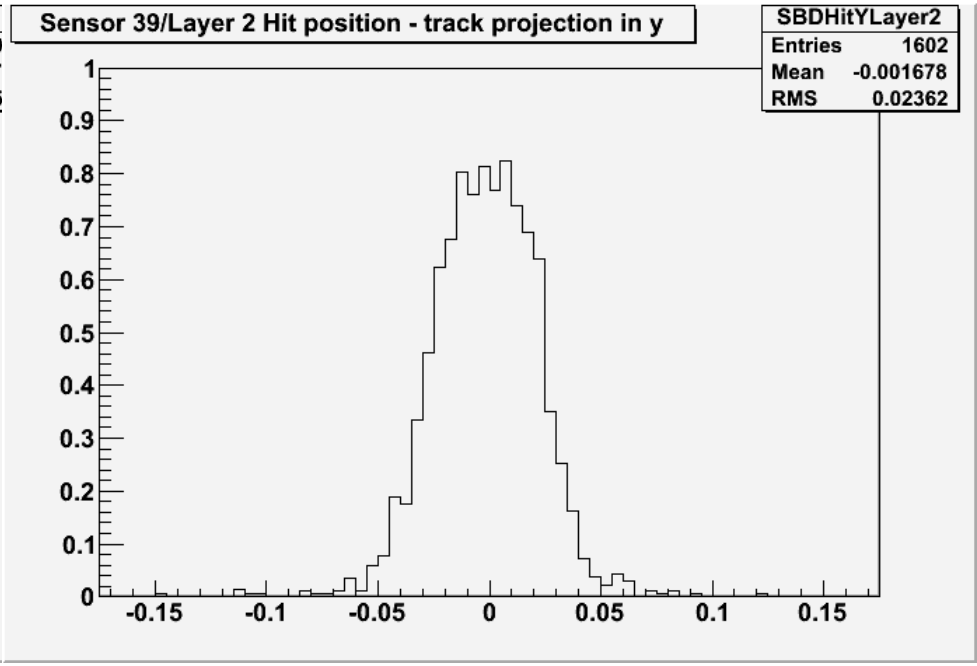




Projections in x and y vs threshold

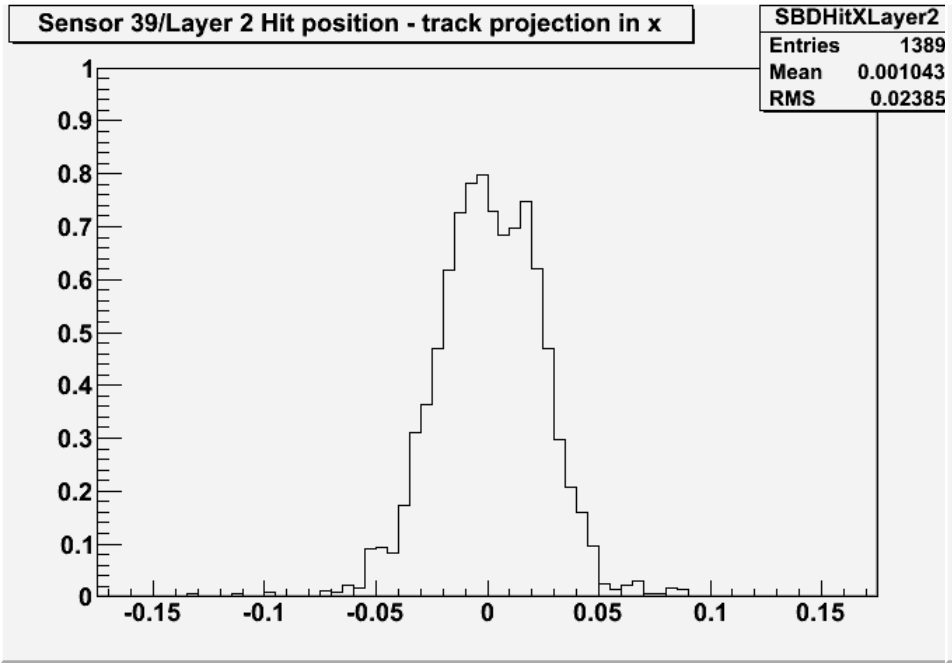
- Run 447952, threshold 210

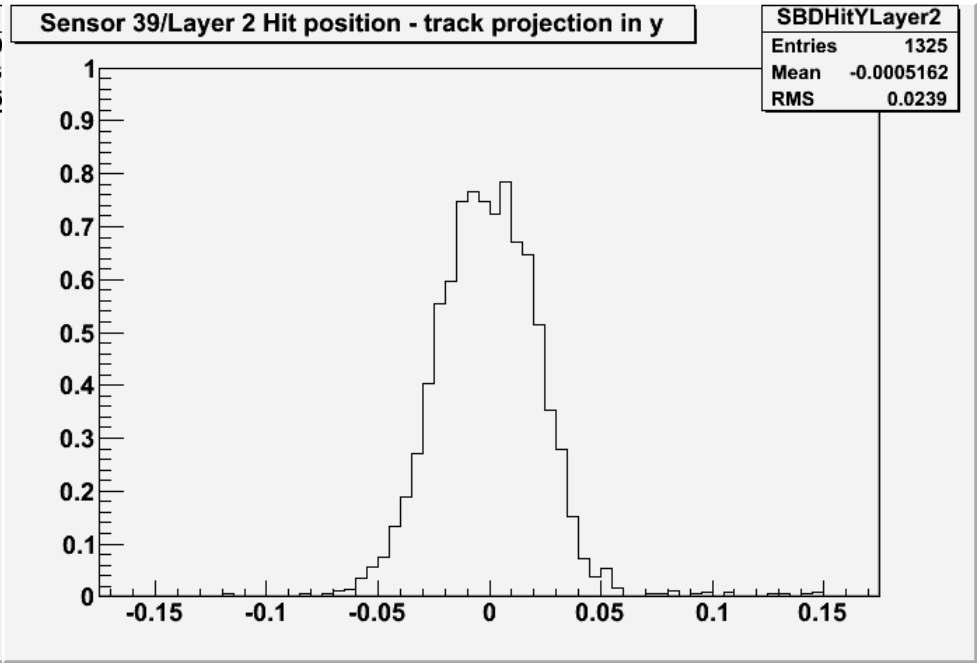




Projections in x and y vs threshold

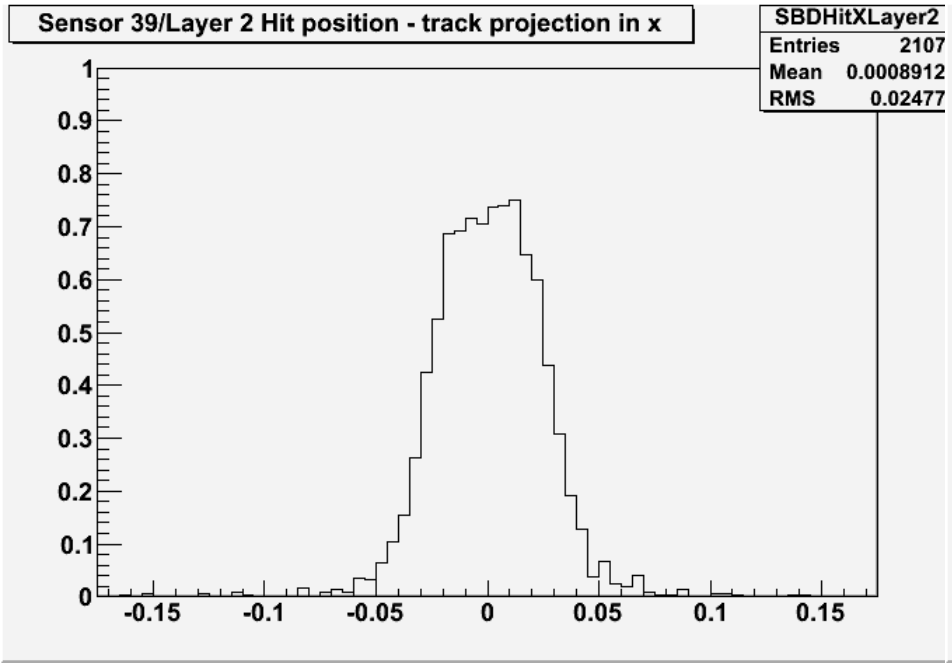
- Run 447954, threshold 220

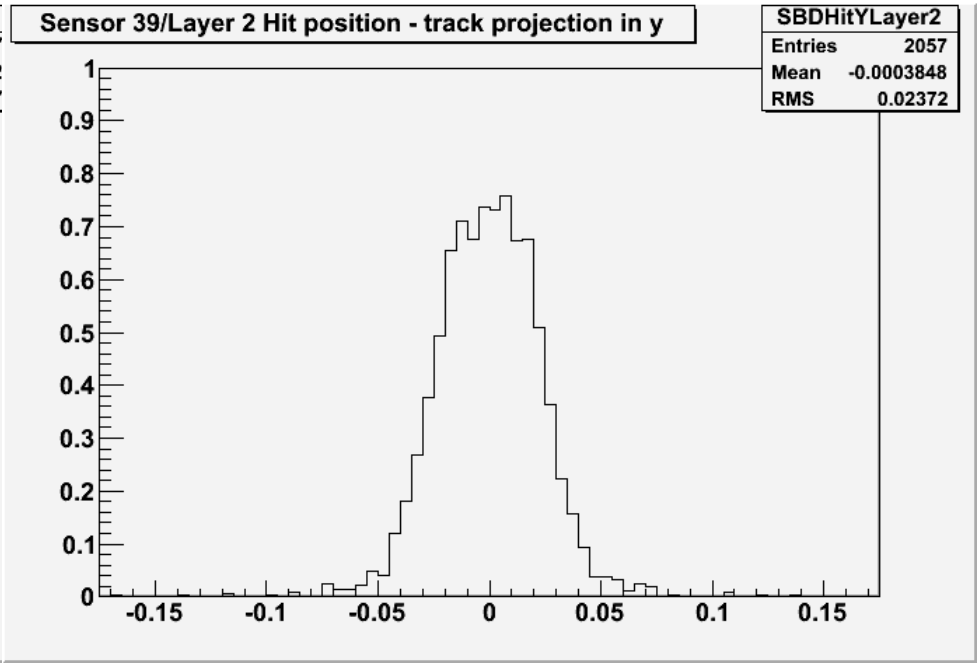




Projections in x and y vs threshold

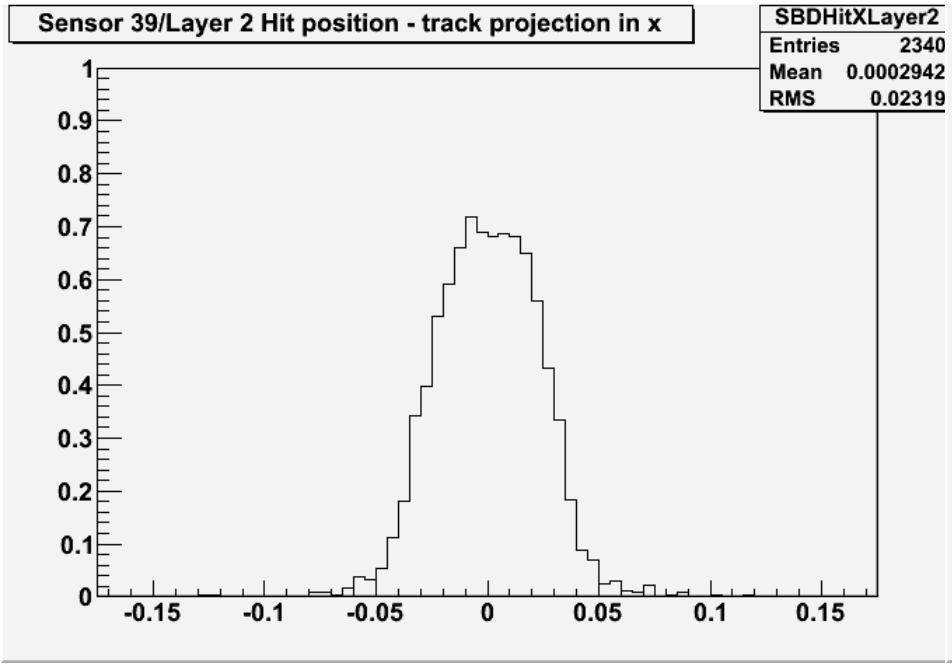
- Run 447956, threshold 230

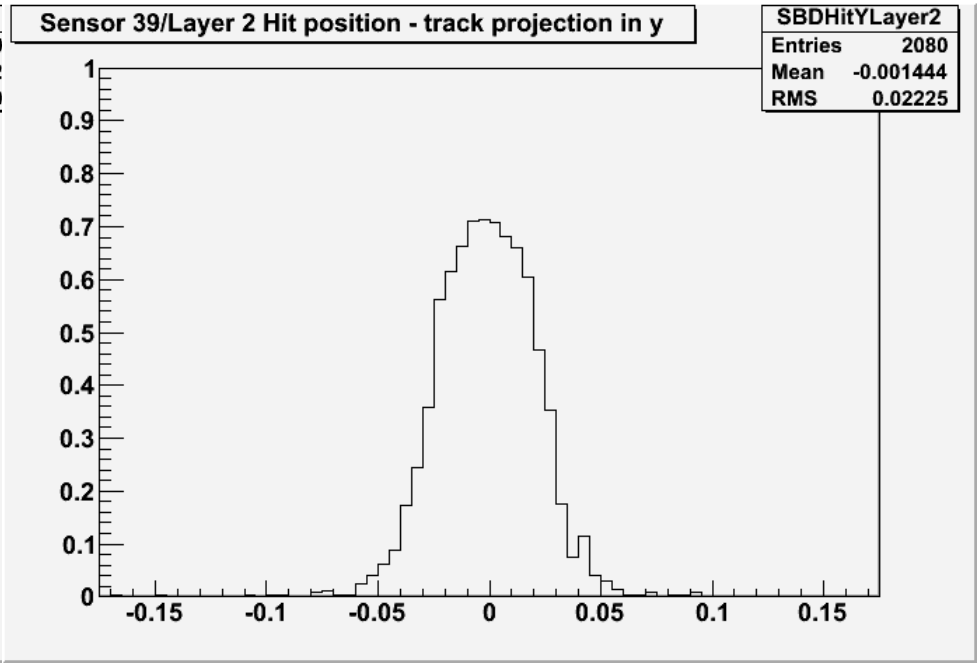




Projections in x and y vs threshold

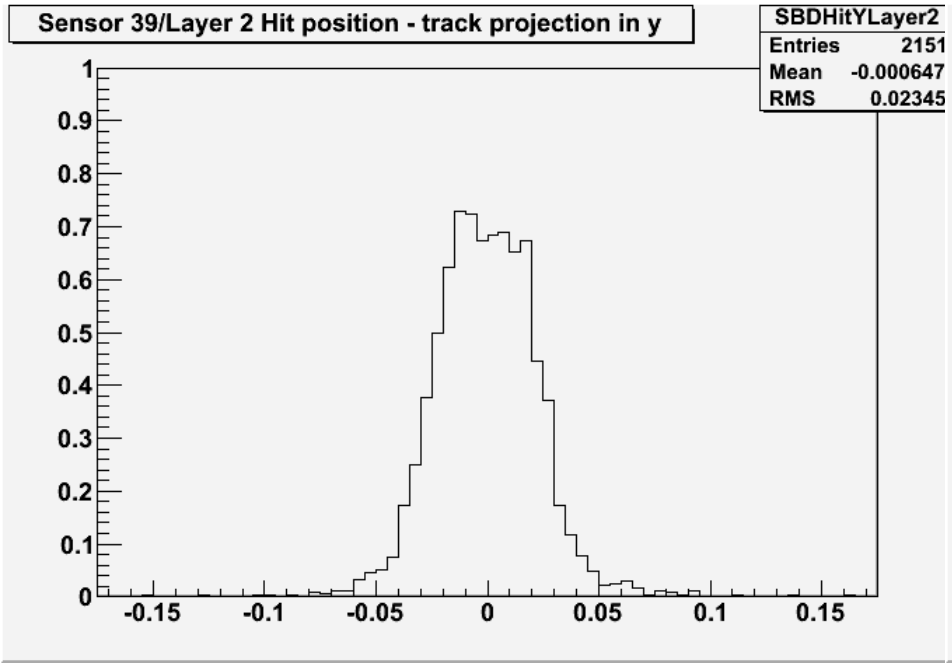
- Run 447958, threshold 240

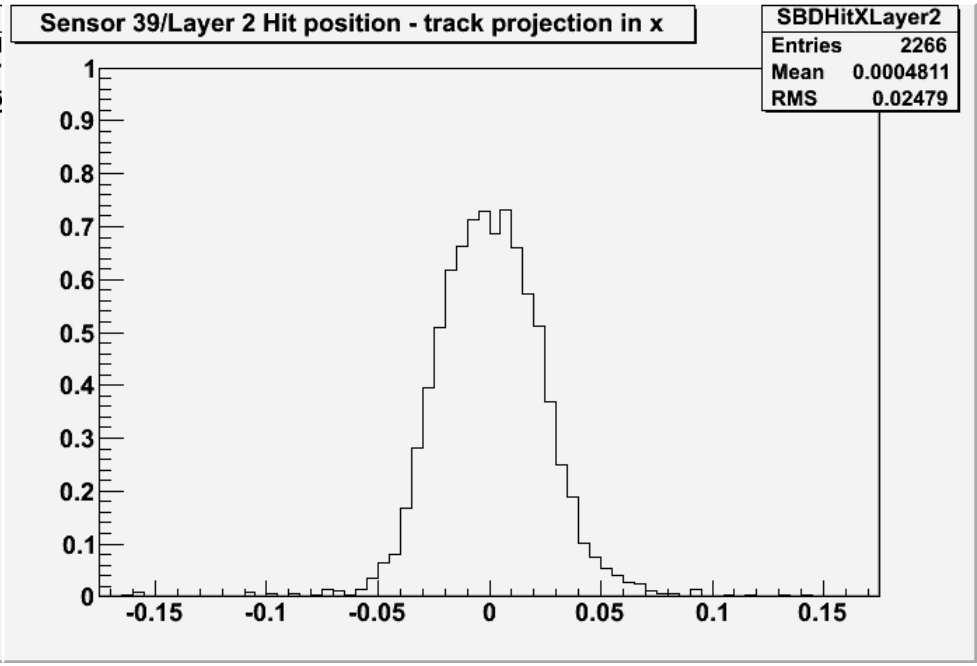




Projections in x and y vs threshold

- Run 447960, threshold 250





Conclusions

- Major part of previous (apparent) inefficiency due to scintillator/sensor overlap
  - Overlap is ~65% with first two scintillators
- Need to find a method to measure bad track rate
  - Not yet started
- Efficiency does not monotonically decrease with threshold
  - Low thresholds have very low efficiencies
  - Presumably due to memory filling
- Efficiency does not fall off fast at high thresholds
  - Did not take data above 250 TU
  - Insufficient range for our studies?