Pixel Efficiency Update

Janos Karancsi, Viktor Veszpremi,
ATOMKI Debrecen

2010 Data processed

- Datasets:
  - /MinimumBias/Commissioning10-Jun14thReReco_v1/RECO
  - /MinimumBias/Run2010A-Nov4ReReco_v1/RECO
  - /MinimumBias/Run2010B-Nov4ReReco_v1/RECO

- CMSSW version:
  - CMSSW_3_6_1_patch4 (for Commissioning 10)
  - CMSSW_3_8_6 for 2010A and 2010B

- Lumi selection from JSON files:
  - Cert_132440-137028_7TeV_June14thReReco_Collisions10_JSON.txt

  (Only runs above 135 000 were selected)
  - Cert_136033-149442_7TeV_Nov4ReReco_Collisions10_JSON.txt

Event and track selection

- Event Selection (vertex)
  - Number of vertices ≥ 1, where $|z| < 15cm, \quad N_{dof} > 4, \quad |\rho| < 2.0$
- Track Selection for „generalTracks” in BPix
  - p_T >1 GeV
  - Track consistent with primary vertex (|dz| < 0.1 cm, |d0| < 0.01 cm)
  - Number of strip hits > 10
  - Valid hit conditions (listed on next slide)
- Fiducial region selection (varies by module type)
  - Avoid module overlaps and edges
  - Ensure that propagated track falls on the right module in Layer 1

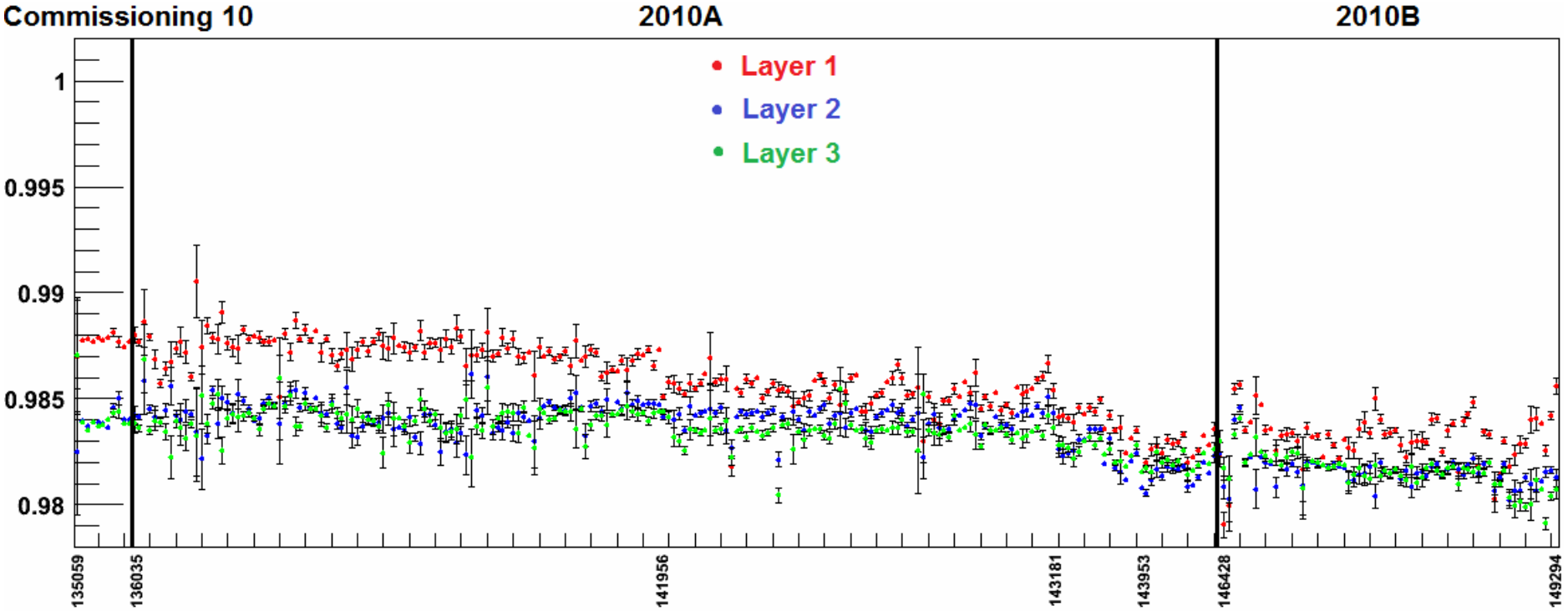
Definition of efficiency

- Hits are required on the „other" layers or disks in order to remove bias due to pixel seeding
  - for Layer 1 : on Layer 2+3, Layer 2 + Disk 1, Disk 1 +2
  - Layer 2: Layer 1+3, Layer 1 + Disk 1
  - Layer 3: Layer 1+2
  - Disk 1 : Layer 1 +Disk 2, Layer 2 + Disk 2, Layer 3 + Disk 2
  - Disk 2: Layer 1 + Disk 1

- RecHit Efficiency Definition:
  - **Eff = $N_{\text{valid hits}}$ / ($N_{\text{valid hits}}$ + $N_{\text{missing hits}}$)**
    where both valid and missing hits come from track reconstruction, therefore **it is affected by the cluster matching efficiency of the official tracking**
  - Layer 1 definition:
    Eff = $N_{\text{valid propagated hits}}$ / ($N_{\text{valid propagated hits}}$ + $N_{\text{missing propagated hits}}$)
    where hits are propagated from valid Layer 2 hits onto Layer 1

Efficiencies in 2010

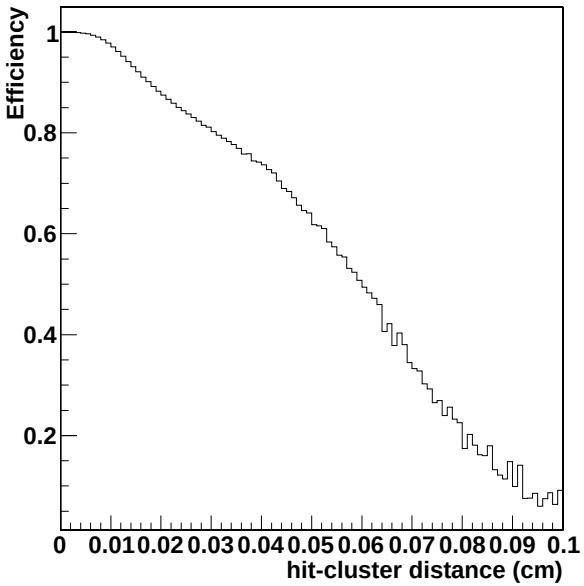
- Efficiency overall is measured ~1 % lower than it was in early 2010. We see a drop when changing to CMSSW 3_6_X (exact reason is not yet known)
- Likely suspect is how clusters are attached to tracks. Tried to fix it as follows

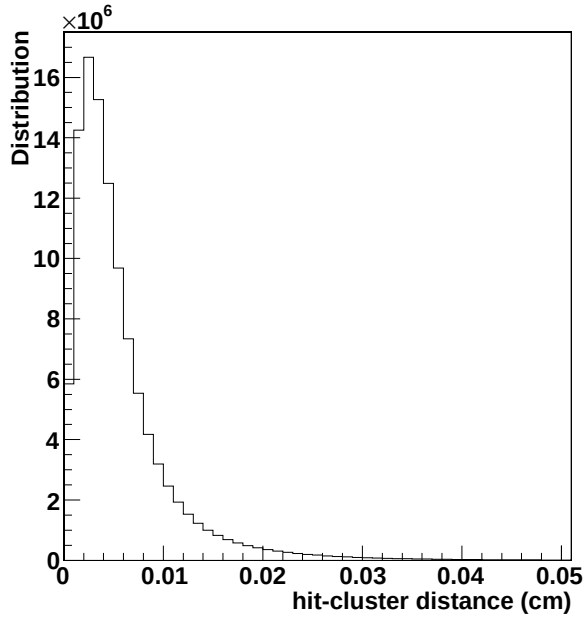


Tracking hit – cluster matching

- Fraction of valid hits to all hits as a function of distance to nearest cluster (plot on the left) per module on Layer 2
  - Hits pass the selections described above (its track has hits on both Layer 1 and 3)
  - A sizable fraction of hits are missing but have a cluster within 100-300 microns!

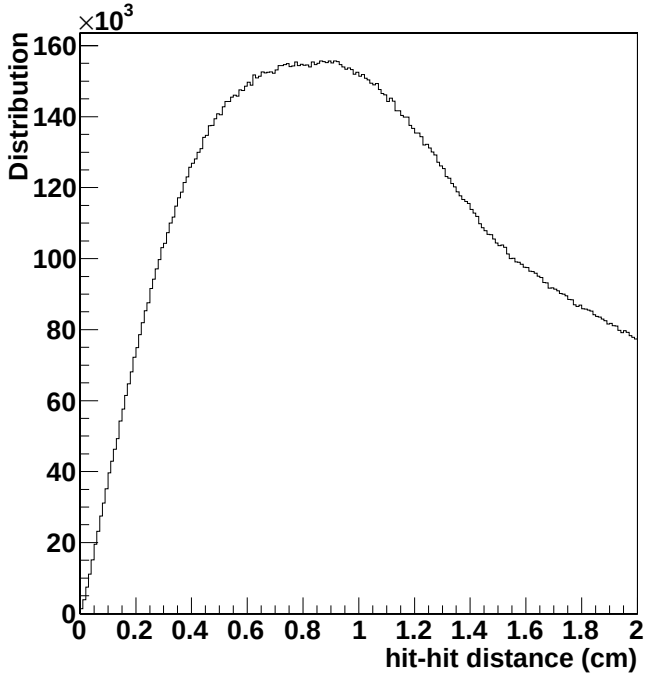
- Nearest cluster (if exists) is within 500 microns for 99.9% of the hits (plot on the right)
  - Would lose 0.1% of hits (with cluster on the same module) if we considered hits valid only when having clusters within 500 microns – efficiency underestimated at most by 0.1%





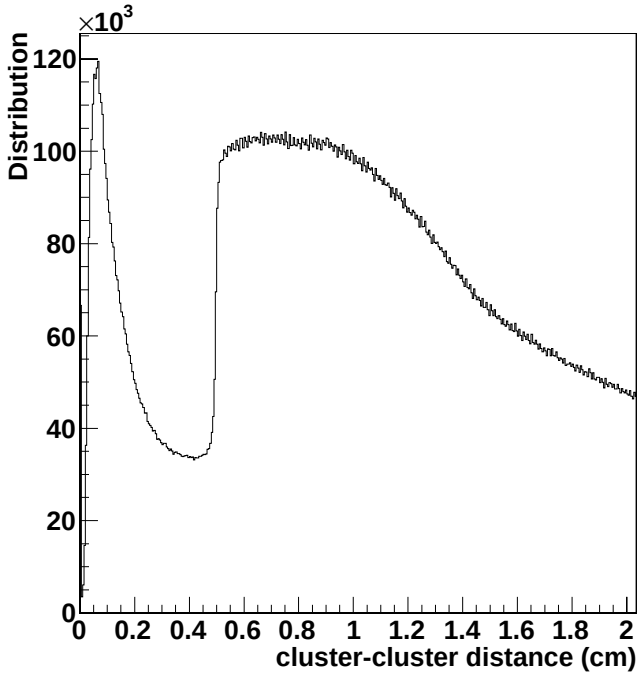
Tracking hit – hit separation

- But how many fake valid hits would be counted?
  - Plotted nearest hit-hit separation on the same module
  - Note: shape suggests no connection between hit-pairs (geometric probability)
  - ~0.3% of hits have another hit within 1000 microns - could produce a fake valid hit
- Tried removing hits which have another hit nearby on the same module
  - Minimum hit separation: 5 mm (arbitrary choice) – do we remove all particle pairs?



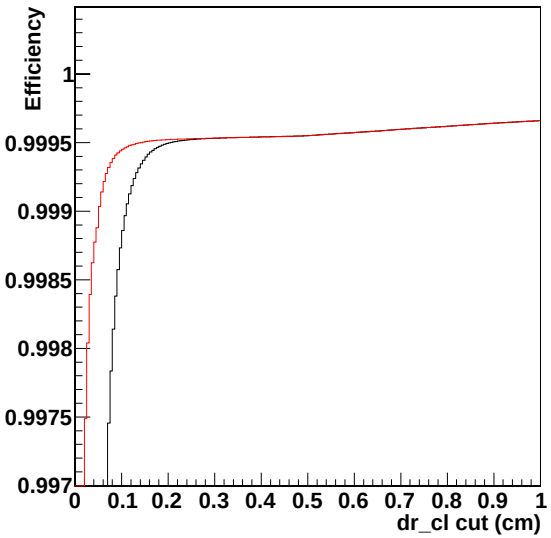
Cluster – cluster separation

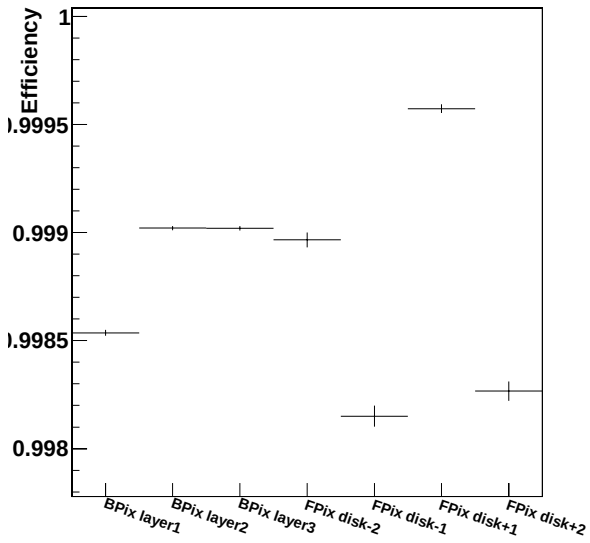
- Checked nearest second cluster, to see if we removed every hit-pairs on the same module
  - See effect of cutting on hit-hit separation, but a third of clusters remain (either they are on neighboring modules or have no tracks)
  - Peak at 600 microns (1-2% of all hits) are clusters that should also belong to the hit (split clusters?) assuming geometric probability
  - Based on the hit-separation plot, we estimate ~0.15% of associated clusters have another track with a cluster within 500 microns



New definition of efficiency

- Efficiency: keep every originally valid hit, and turn a missing hit valid if it has cluster within 500 microns
  - Fully efficient with smaller cluster search distance (red curve, first plot) than when valid hit is decided purely on cluster distance (black curve)
  - Cluster mathing distance is arbitrary for now, but it should be as small as possible
  - We should avoid cutting hard on hit separation if we want to measure efficiency losses due to high occupancy (in pp collisions it is not needed)
- Overall layer efficiency (on the right) is accurate within 0.1%



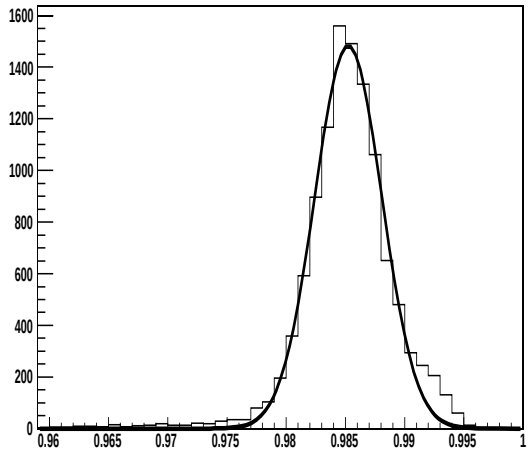


ROC efficiencies

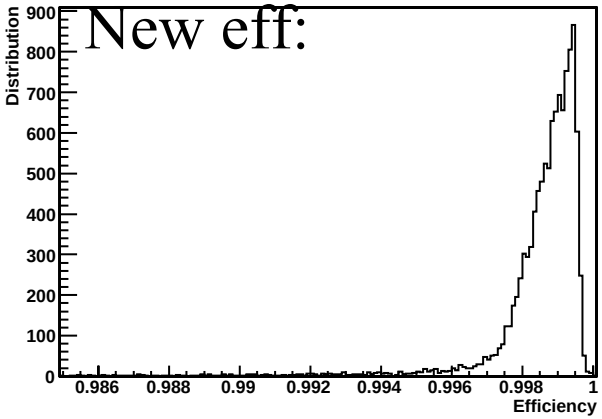
- Distribution of efficiencies computed for each ROC
  - Oldeff: 98.5% +/- 0.3
  - Neweff: 99.9% +/- 0.15%
  - Max error on efficiency calculation with new method is ~0.1%

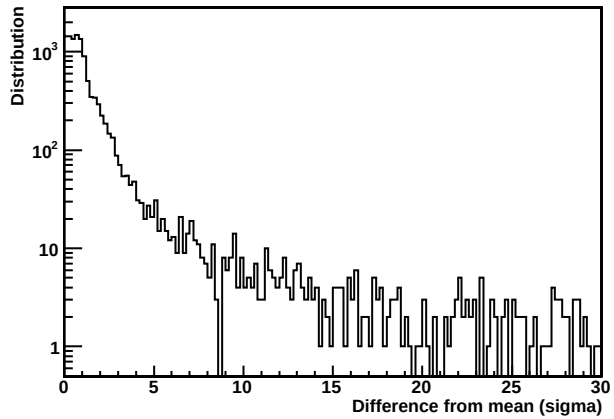
- Different performances between ROCs started to be visible?

Old eff:



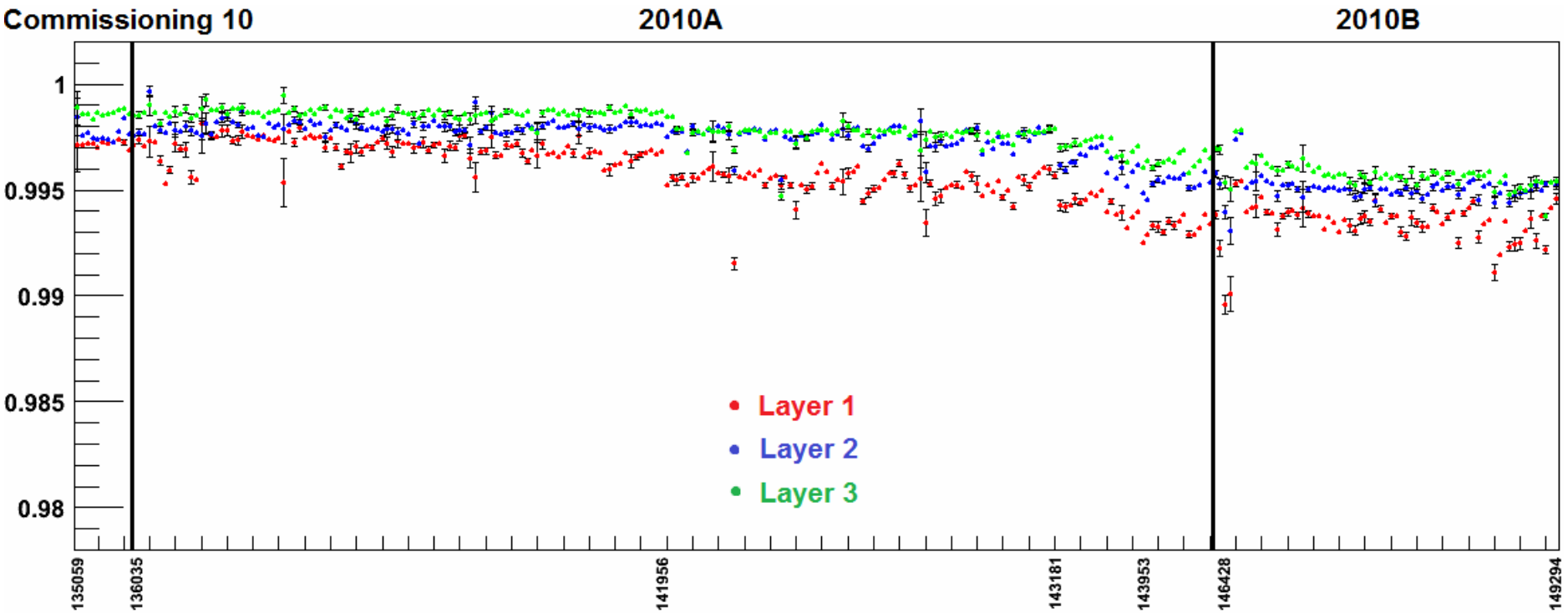
New eff:



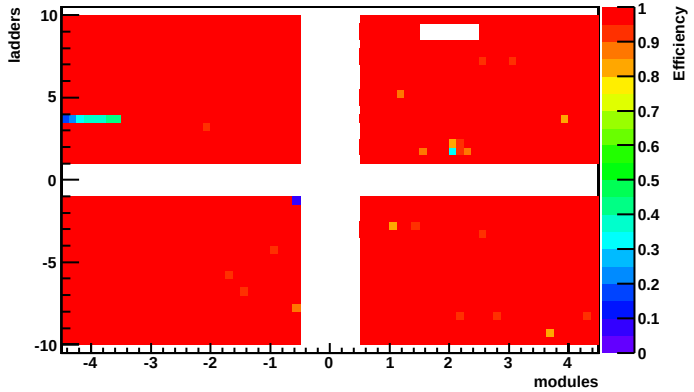


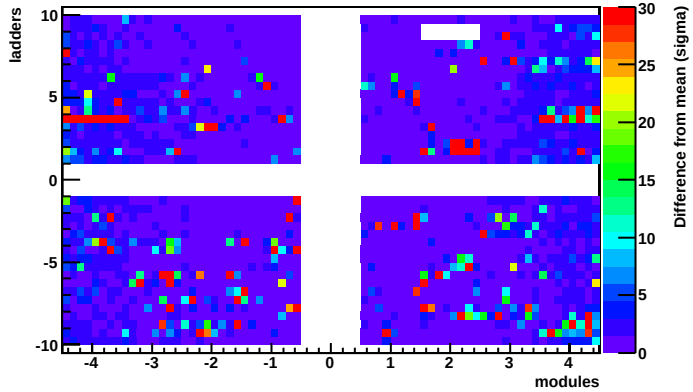
Efficiencies in 2010 (new)

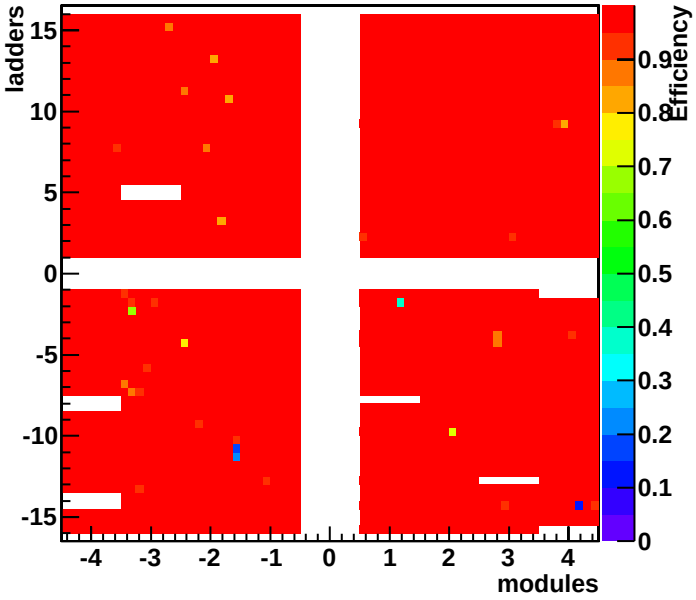
- 1% drop experienced starting with CMSSW 3_6_X is recovered
- Overall pattern similar to old efficiency – efficiency loss as function of luminosity is visible
- Inner layers experience consistently larger efficiency loss

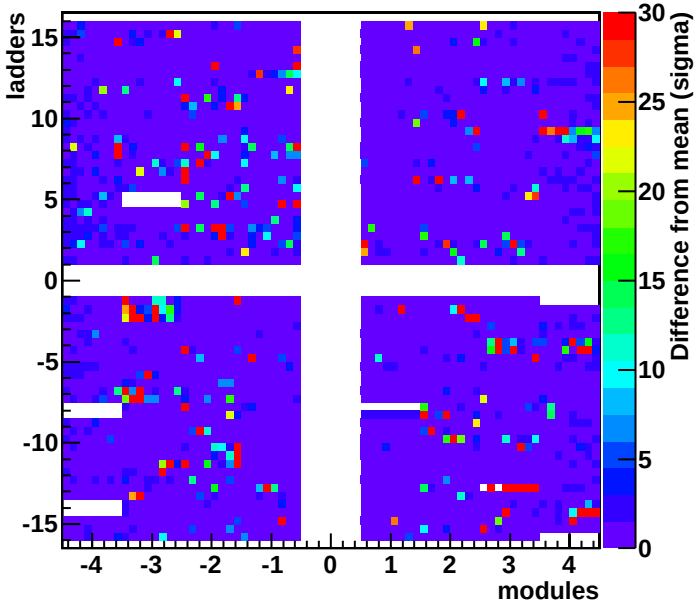


Efficiency (new) map Layer 1 and 2





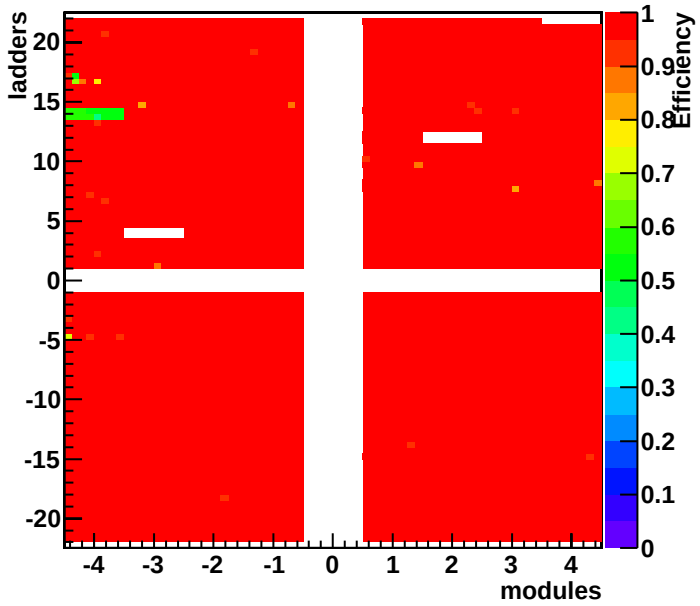




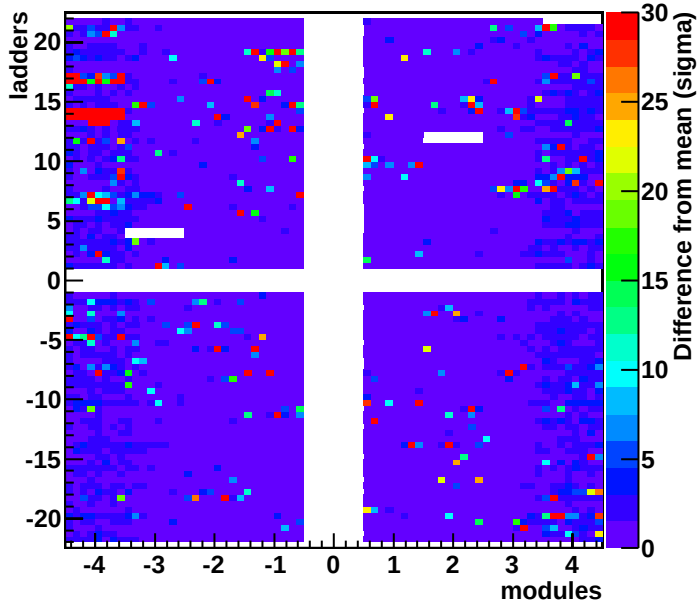
ROC efficiencies

Difference from mean ROC efficiency

Efficiency (new) map Layer 3



ROC efficiencies



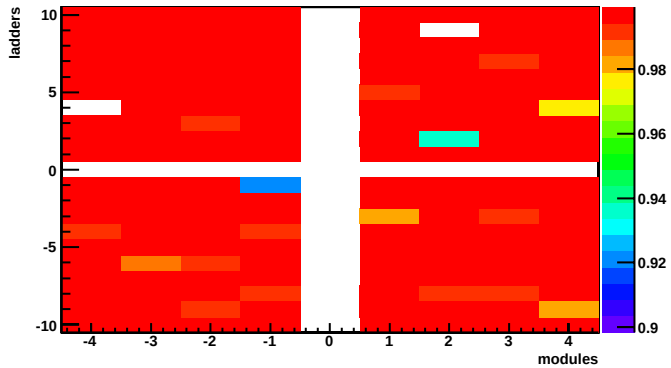
Difference from mean ROC efficiency

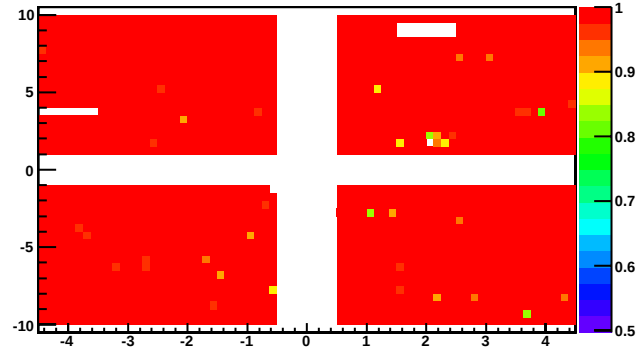
Summary and plans

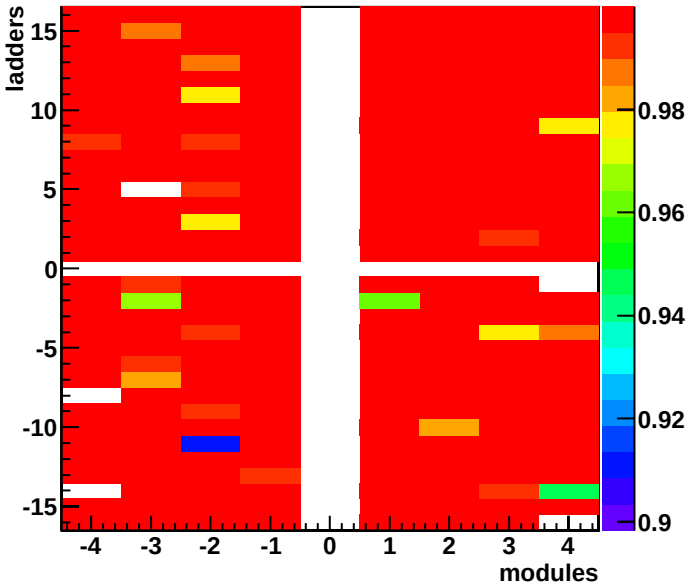
- Source of main efficiency loss in finding clusters to tracks is in the official tracking, this loss is removed from efficiency calculation
- Measured the efficiency in BPix for the entire 2010 running with ~0.1% accuracy
- FPix needs a similar study, we are planning to do it
- Efficiency of outlier ROCs are significantly different from the average
- We have compiled a list of inefficient ROCs in BPix

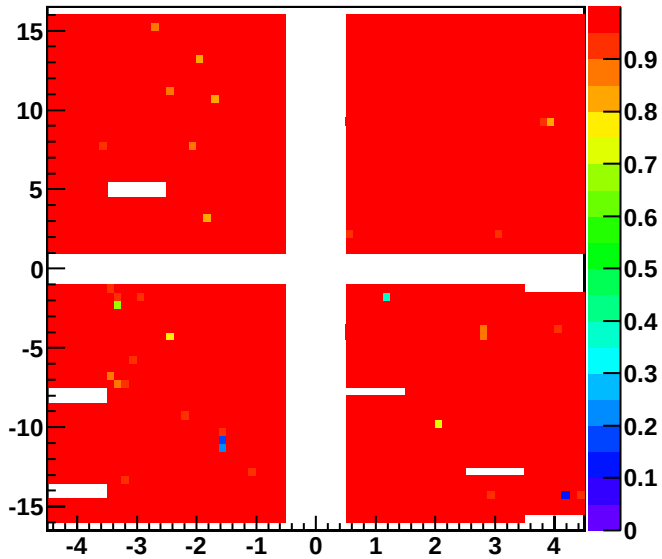

CMS

Efficiency (new) map Layer 1 and 2









Efficiency (new) map Layer 3