Digitization and hit reconstruction for Silicon Tracker in MarlinReco[1]

Sergey Shulga [2], Tatiana Ilicheva[3]

JINR, Dubna, Russia
GSU, Gomel, Belarus

LCWS07
30 May – 3 June 2007
Hamburg, Germany

[1] This work is supported by BMBF(Germany)

[2] shulga@mail.desy.de, [3] ilicheva@mail.desy.de

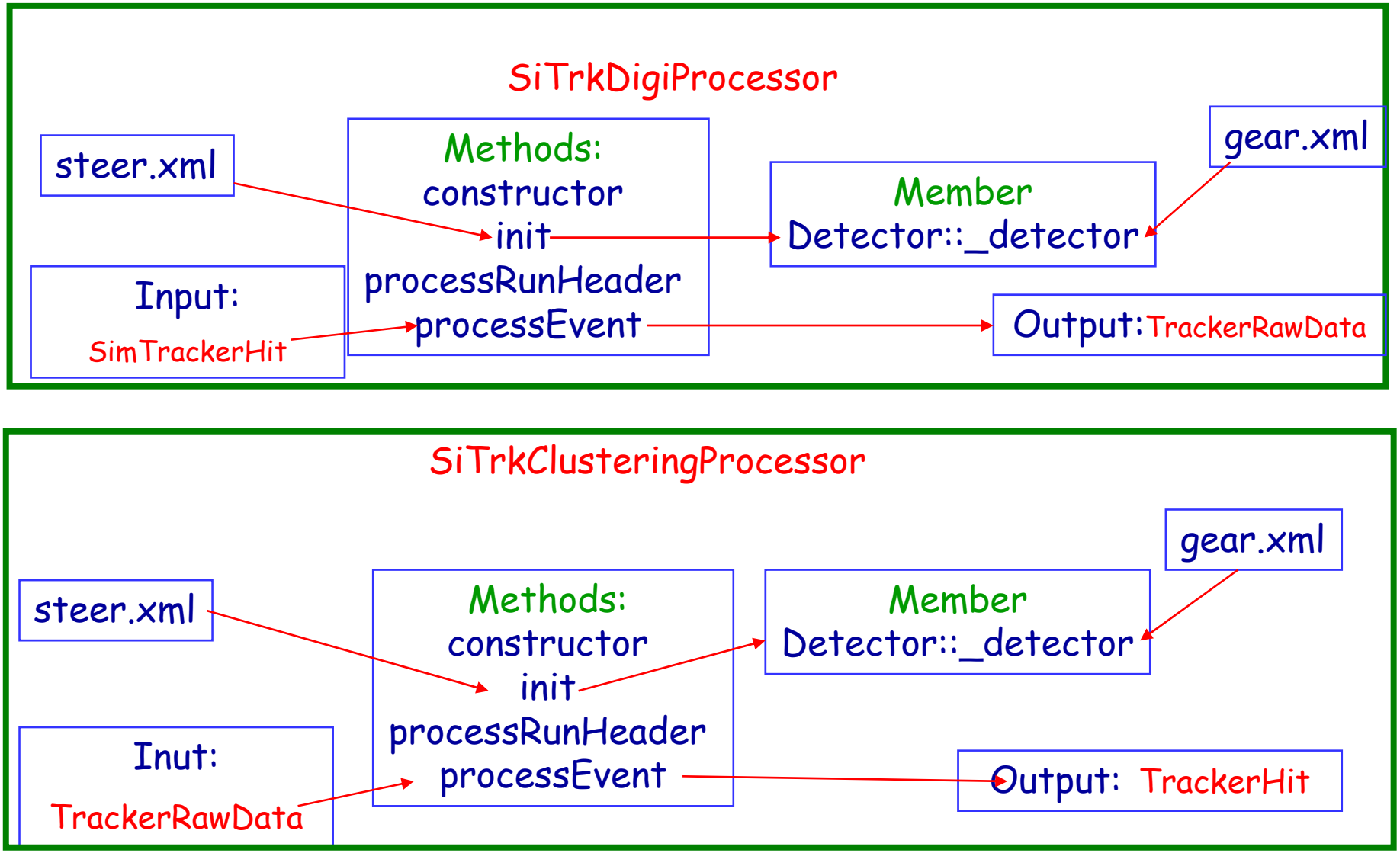
Introduction

Status of MarlinReco implementation of:

- processor for digitization of Si tracker:
  - * design : classes Detector and DetUnit,
  - * digitizer : parameters, input and output;

- clustering processor:
  - * clusterizer : parameters, input and output;
  - * pixel clustering algorithm
  - * cluster parameter estimation
  - * validation of reconstructed hits

Design of package: Processors



Design of package: Detector and DetUnit

Class Detector is container of layers and DetUnit's

Main method of Detector performs initialization
of DetUnit by using GEAR xml-information.

Abstract base class DetUnit is
container of sim/raw/rec and temporary hits.

DetUnit can read/write standard LCIO sim/raw/rec hits.

Detector contains digitizer and clusterizer.

The object to be digitized/clusterized is DetUnit.

There are logical reasons to include
classes Detector and DetUnit in GEAR

Digitizer and clusterizer: references

Notes:

- S.Cucciarelli, D.Kotlinsky, T.Todorov, CMS Note 2002/049
- S.Cucciarelli, D.Kotlinsky, CMS IN 2004/014

Presentations:

- D.Kotlinski, Pixel Software Workshop, 11-15/01/07 (CMS)
- G.Giurgiu, P.Maksimovich, M.Swartz, Offline Pixel Meeting, 05/02/07 (CMS)
- G.Giorgiu, Pixel Workshop, 01/12/07 (CMS)
- D.Kotlinski, Pixel Software meeting, 19/09/06 (CMS)

Codes

taken from CMS software and **adapted** in LCIO/MarlinReco framework

---

See also our talk at
ILC Software and Tools Workshop LAL-Orsay, May 3, 2007

http://ilcagenda.linearcollider.org/conferenceDisplay.py?confId=1446

Digitizer (summary)

Parameters	default
• Pixel sizes	0,150 x 0,150 mm
• DetUnit thickness	0,282 mm
• Ionization segment length	0.01 mm
• Angle of Lorenz drift	$\tan\Theta_{Lorenz} = 0.106 * B$, B = 4 Tesla
• $\sigma_{diffusion}$ for drift length 0.3 mm	0.007 [mm]
• Fired cluster widths: $\sigma_X, \sigma_Y = f(\Theta_{Lorenz}, \sigma_{diffusion})$	$[3*\sigma_X, 3*\sigma_Y]$,
• RMS of gaussian distribution of pixel noise	500 electrons
• Pixel threshold in units of noise RMS for pixels 0,150 x 0,150 [mm]	4 (2000 electrons)
• Efficiency for single pixel	99%
• Efficiency for pixel double column	99%
• Readout Chip efficiency	99.75%
• Readout Chip sizes (in units of pixels)	20 x 52

Digitizer input: DetUnit with collection of sim.hits (TrackerSimHit) in event
and geometrical information: pixel X,Y sizes, thickness,
and number of pixels in DetUnit along X (row) and Y (column).

Digitizer output: DetUnit with collection of fired pixels which are collected in
map < int channel, Amplitude amp > _signal, where channel is packed 2-dimensional pixel number,
Amplitude contains total charge (in electrons)
from all sim.hits in event and vector of contributing sim.hits.

Clusterizer

The clusterization is performed on a matrix with size
equals to the size of the pixel detector.
Each cell contains the ADC count of the corresponding pixel
with ADC > pixelThreshold * noiseRMS.

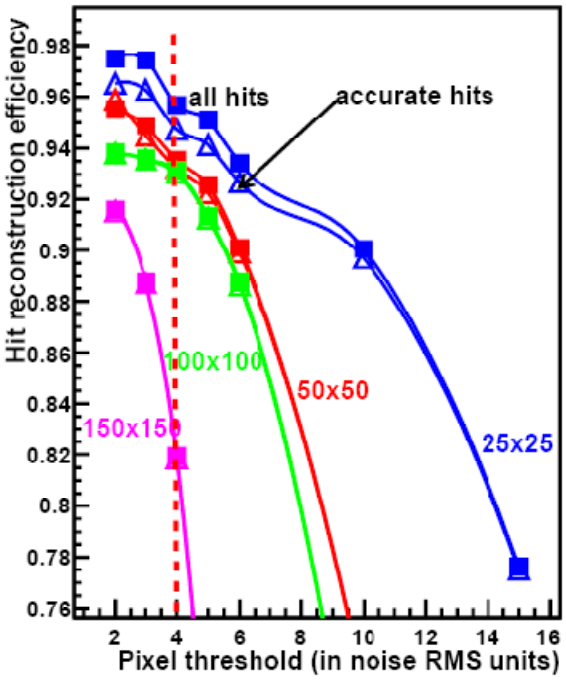
The search of cluster starts from seed pixels
containing ADC > seedThreshold * noiseRMS.

Clusters are set of neighbour pixels
including pixels which touched by corners
with total cluster ADC > clusterThreshold * noiseRMS

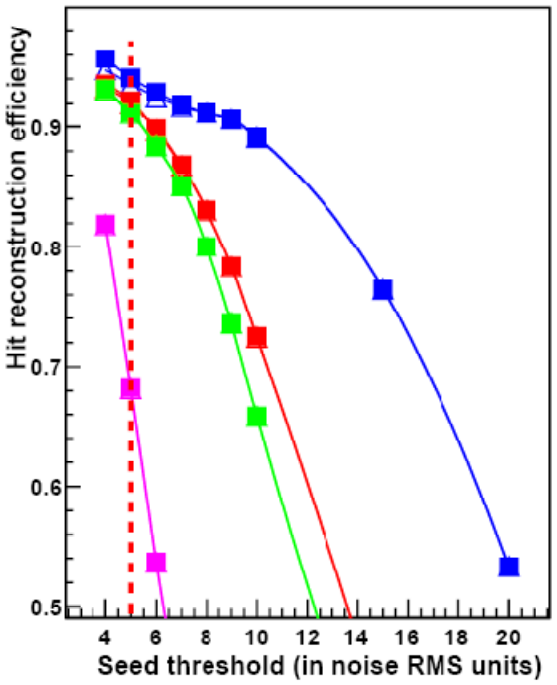
Clusterizer input: DetUnit with collection of raw.hit (TrackerRawData) in event
and geometrical information: number of pixels in DetUnit along X (row) and Y (column).

Clusterizer output: DetUnit with collection of reconstructed hits (TrackerHit)

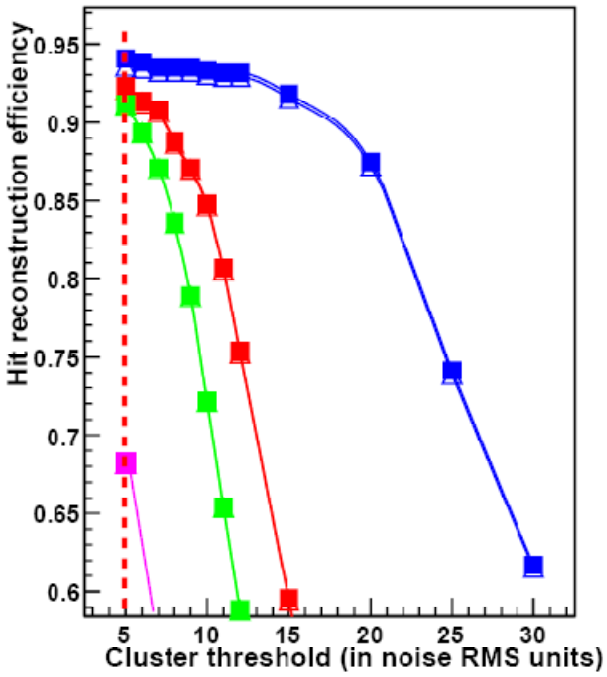
Thresholds



seedThreshold = 0,
clusterThreshold = 0



pixelThreshold = 4,
clusterThreshold = 0



pixelThreshold = 4,
seedThreshold = 5

- Noises is switch off to find efficiencies
- blue – pixels 25x25 microns, noiseRMS = 100 electrons
  red – 50x50, noiseRMS = 150 electrons
  green – 100x100, noiseRMS = 250 electrons
  magenta – 150x150, noiseRMS = 500 electrons
- thickness of sensitive area = 0,037 mm ("vxd_00")

Set for next study:
PixelThreshold = 4
SeedThreshold = 5
ClusterThreshold = 6
(in units of noise RMS)

Used events

- Mokka06-01
- subdetector "vxd_00":

Layer N	Distance to sensitive part, mm	Ladder length, mm	Ladder sensitive thickness, mm
1	15,78	50	0,037
2	27,28	125	0,037
3	38,28	125	0,037
4	49,28	125	0,037
5	60,28	125	0,037

- MC events: Pythia6.410, MSEL = 6, $e^+e^- \to t\bar{t} \to X$,
  c.m.s. energy = 500 GeV

Hit reconstruction efficiency

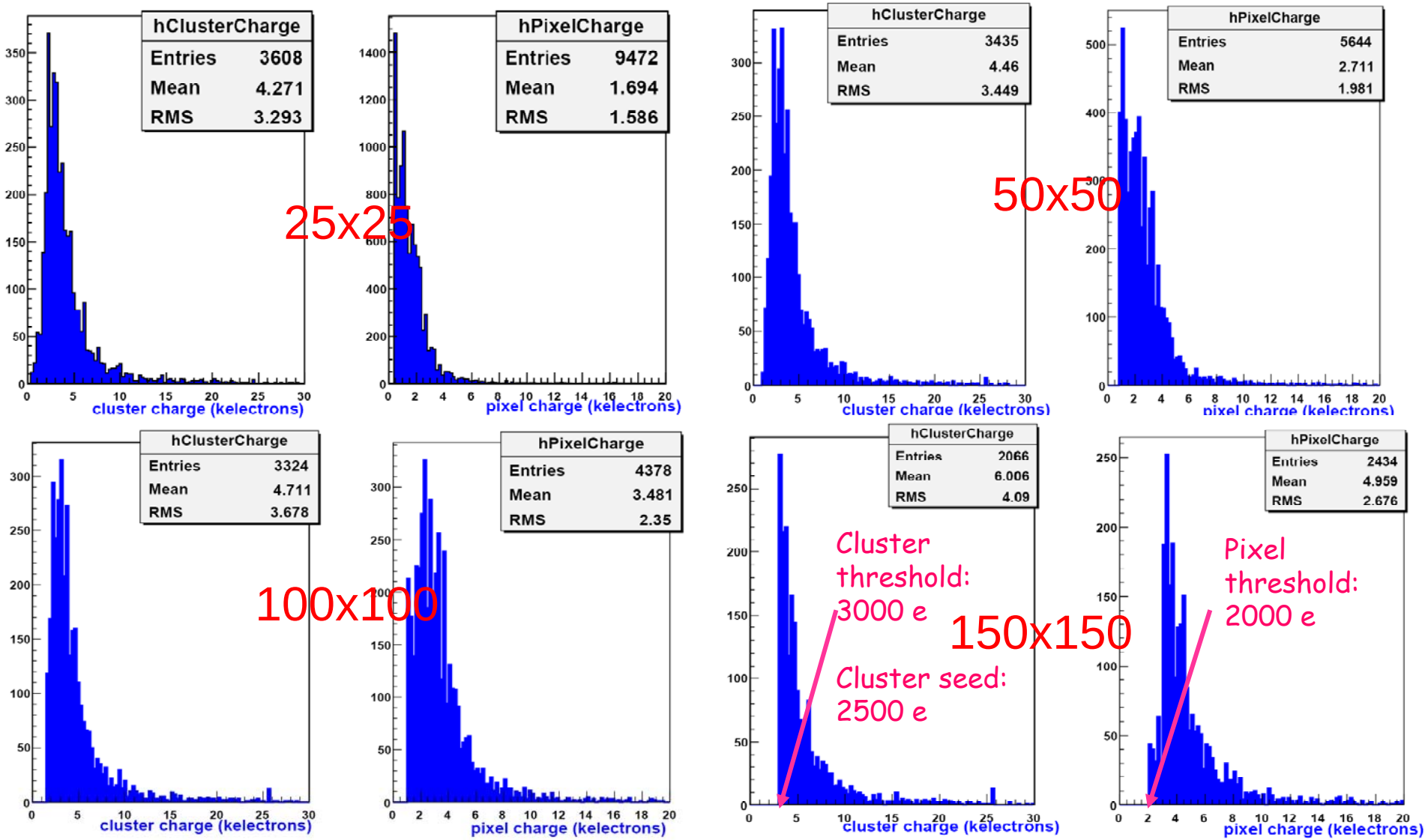
Pixel size, μm^2	All rec.Hits/sim.Hits	True [1] recHits/sim.Hits	False rec.Hits /rec.Hits
25x25	0,943	0,935	0,0086
50x50	0,898	0,896	0,0015
100x100	0,869	0,868	0,0006
150x150	0,540	0,539	0,001

- Subdetector "vxd_00"
- Noises is switch on
- Inefficiencies are applyed to kill some pixels, double columns of pixels, readout chips

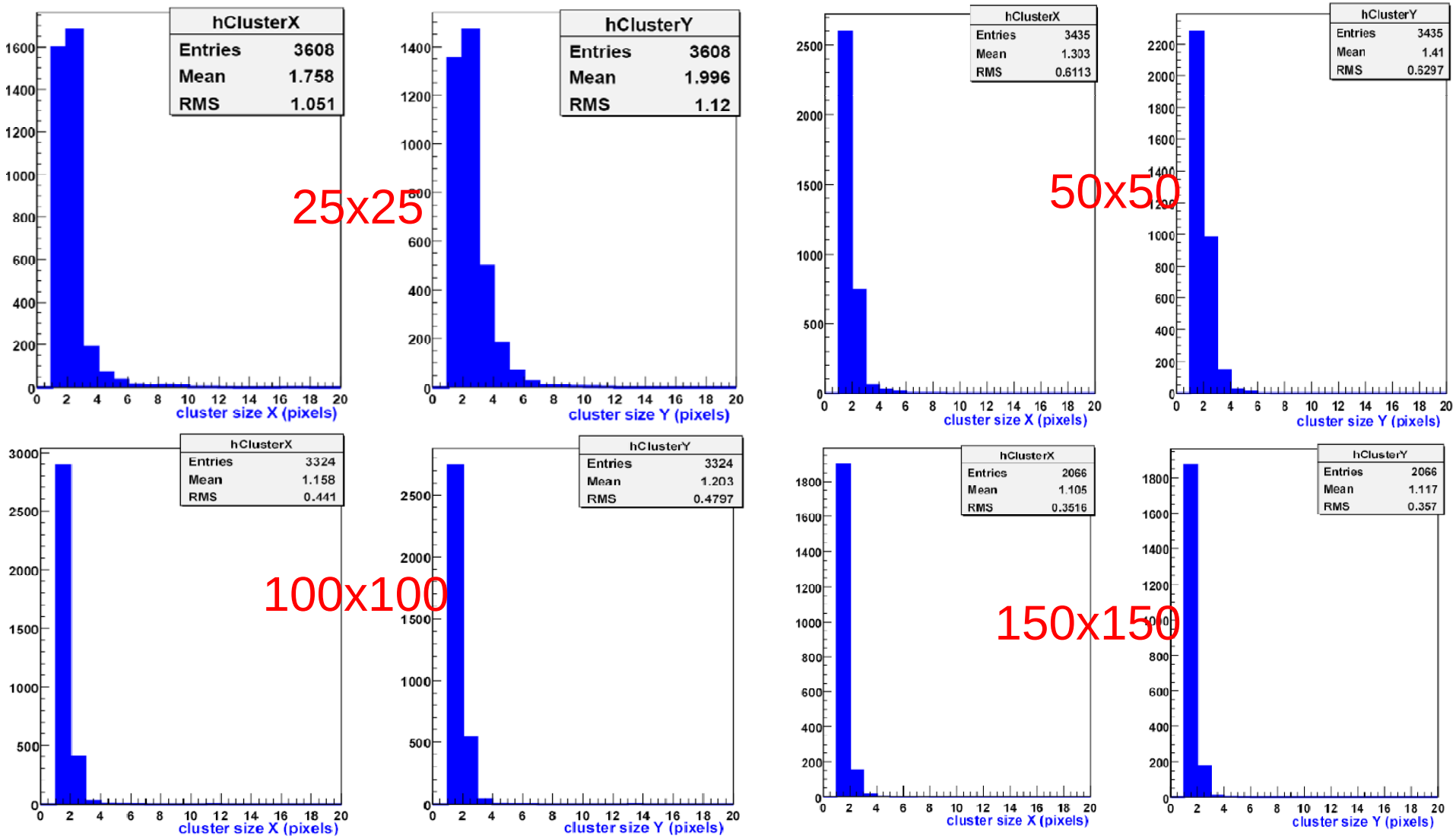
[1] True rec.hits are in distance less then 1 pixel size to sim.hit position

PixelThreshold = 4
SeedThreshold = 5
ClusterThreshold = 6
(in units of noise RMS)

Charge


hClusterCharge
Entries 3608
Mean 4.271
RMS 3.293
cluster charge (kelectrons)
hPixelCharge
Entries 9472
Mean 1.694
RMS 1.586
pixel charge (kelectrons)
25x25
hClusterCharge
Entries 3435
Mean 4.46
RMS 3.449
cluster charge (kelectrons)
hPixelCharge
Entries 5644
Mean 2.711
RMS 1.981
pixel charge (kelectrons)
50x50
hClusterCharge
Entries 3324
Mean 4.711
RMS 3.678
cluster charge (kelectrons)
hPixelCharge
Entries 4378
Mean 3.481
RMS 2.35
pixel charge (kelectrons)
100x100
hClusterCharge
Entries 2066
Mean 6.006
RMS 4.09
cluster charge (kelectrons)
Cluster threshold: 3000 e
Cluster seed: 2500 e
150x150
hPixelCharge
Entries 2434
Mean 4.959
RMS 2.676
pixel charge (kelectrons)
Pixel threshold: 2000 e

Cluster size



Cluster Parameter Estimation

- Fully documented in CMS note 2004/014 by S. Cucciarelli and D. Kotlinski
- Uses information from the first and last pixels in the cluster to determine the hit position:

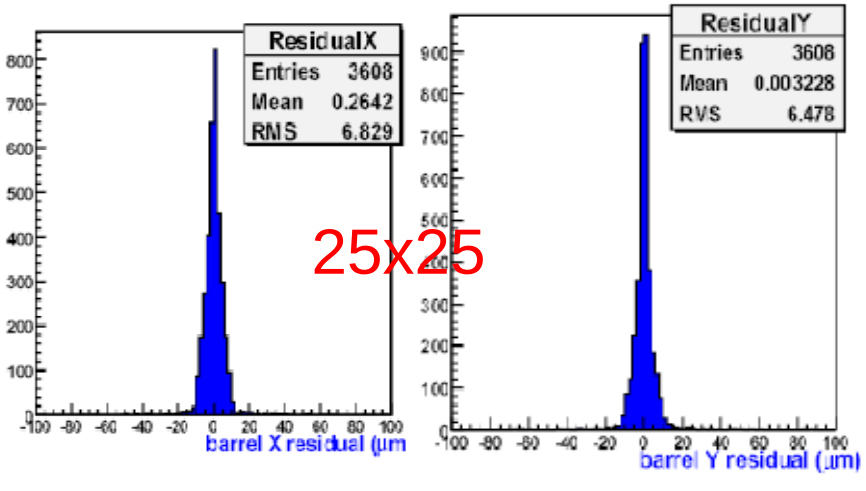
$$x = x_C + \frac{q_{last} - q_{first}}{2(q_{last} + q_{first})} \left| W - W_{inner} \right| - LorentzShift / 2$$

- x_c - geometrical x center
- $q_{first/last}$ – charge of first/last pixel in cluster
- W – charge width = sensor width × tan(α) + LorentzShift
- W_{inner} – inner pixels length

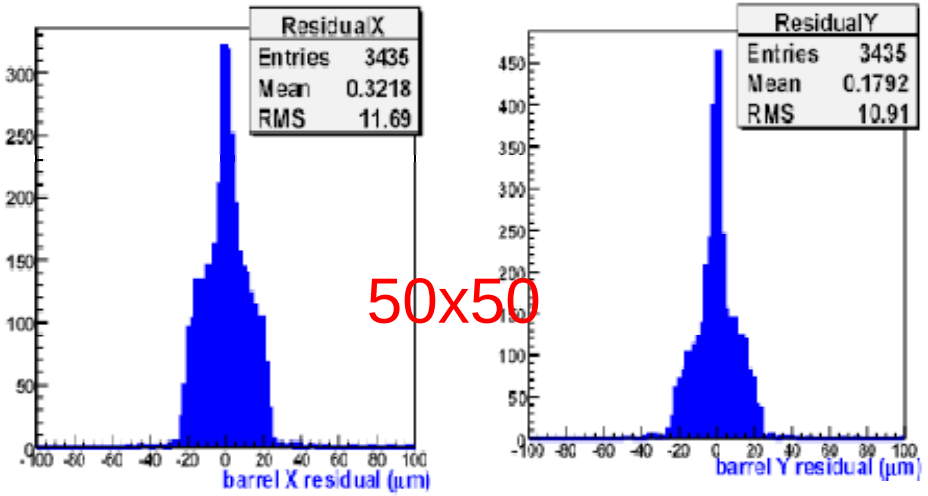
- Initially, incident angle α is approximately calculated assuming the track originates from detector center and goes through module center

Slide taken from talk
G.Giurgiu, CMS Pixel Workshop, 01/12/2007

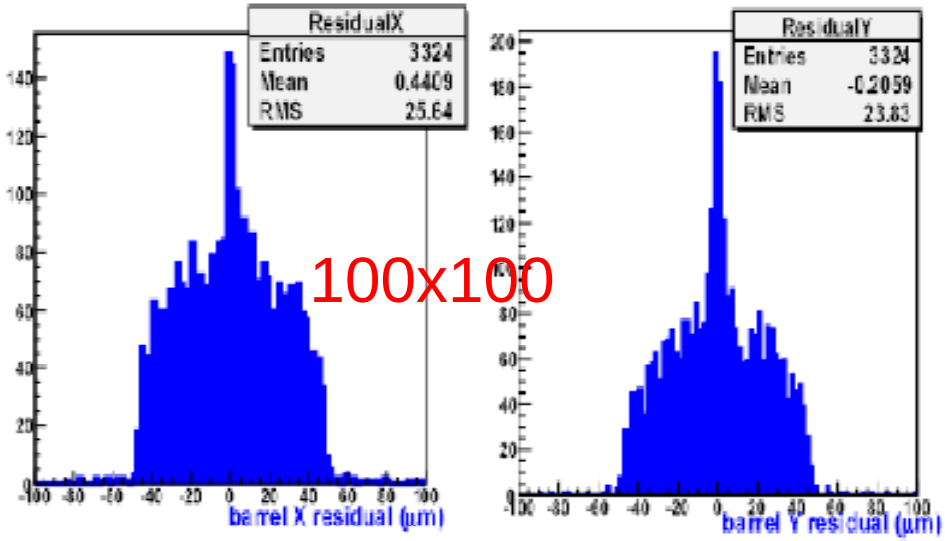
X, Y residuals



25x25



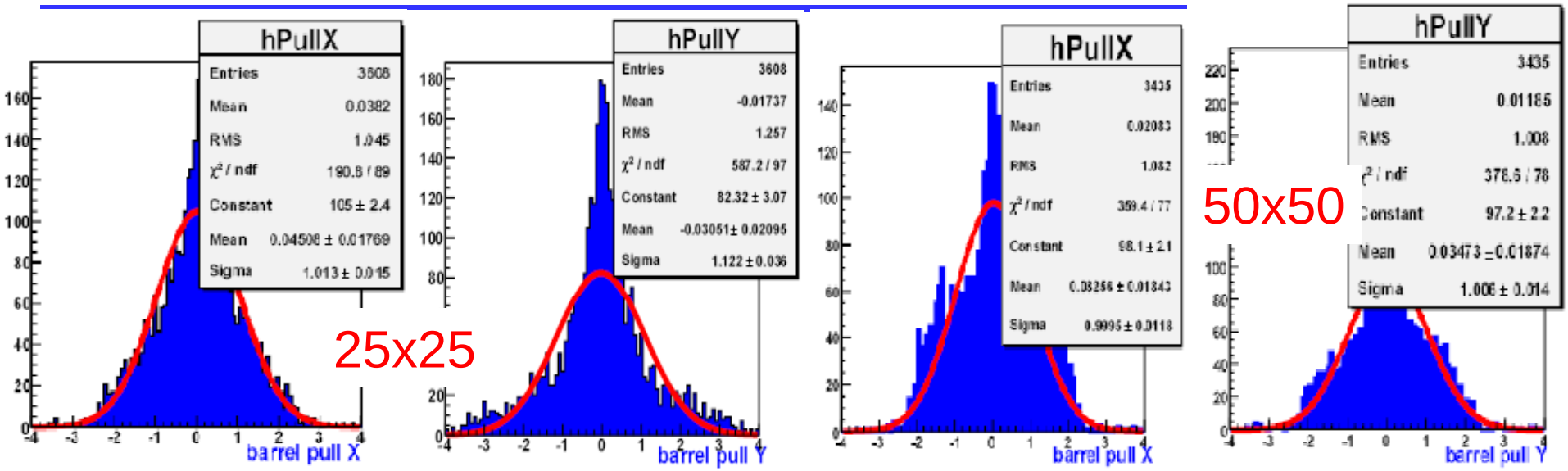
50x50



100x100

Residual = distance w.r.t. the closest simHit

Errors and pulls



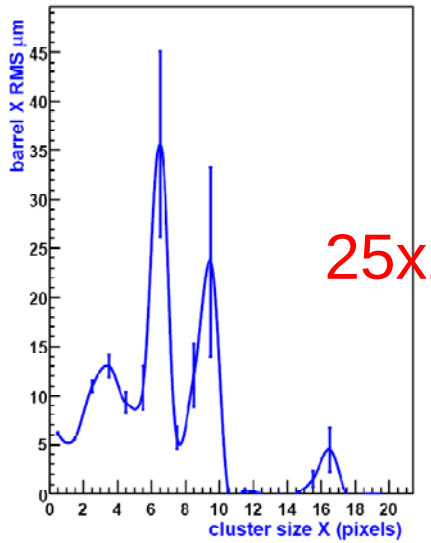
pull = residual/ σ

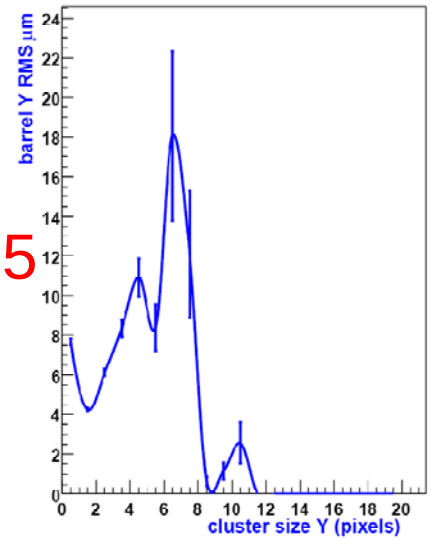
Constant errors are used in pulls.

Task to improve the error estimators

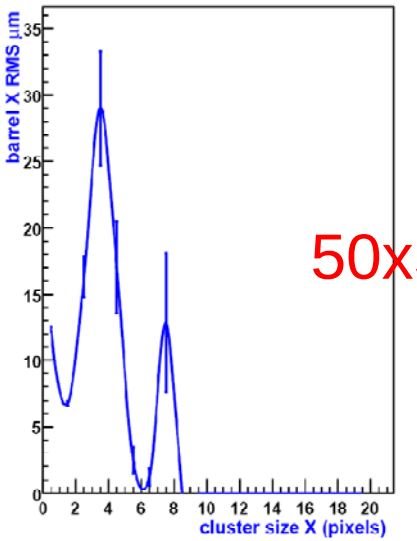
μm^2		$\sigma, \mu m$	RMS, μm
25x25	X	4,14	6,83
	y	3,02	6,48
50x50	X	10,31	11,69
	y	10,42	10,91
100x100	X	22,65	25,64
	y	22,54	23,83
150x150	X	36,11	39,97
	y	36,29	38,53

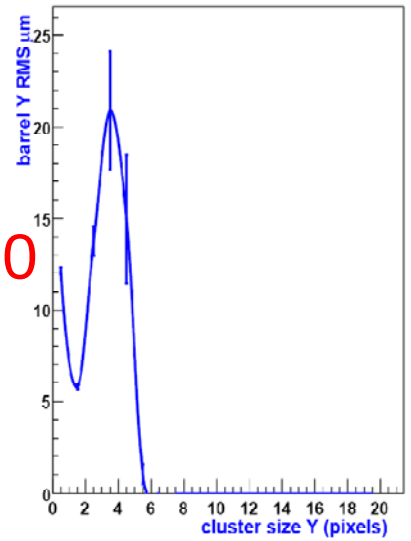
X, Y resolution vs. cluster size



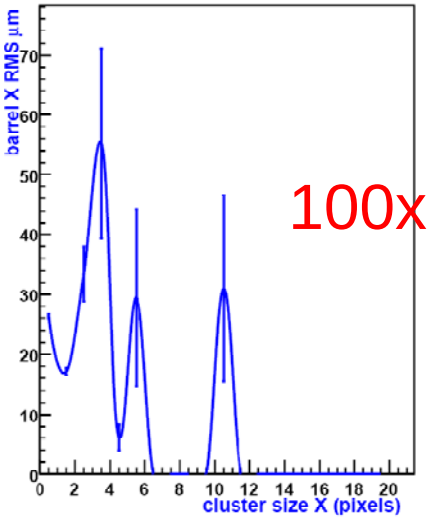


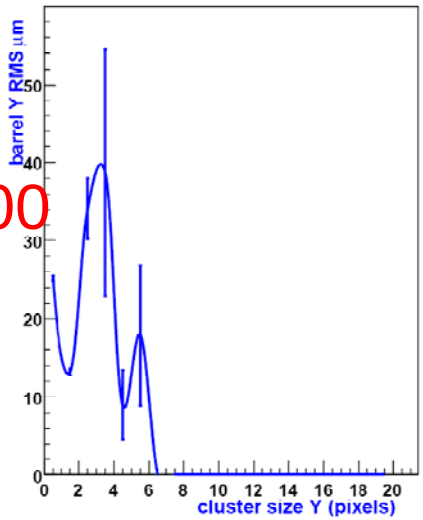
25x25





50x50



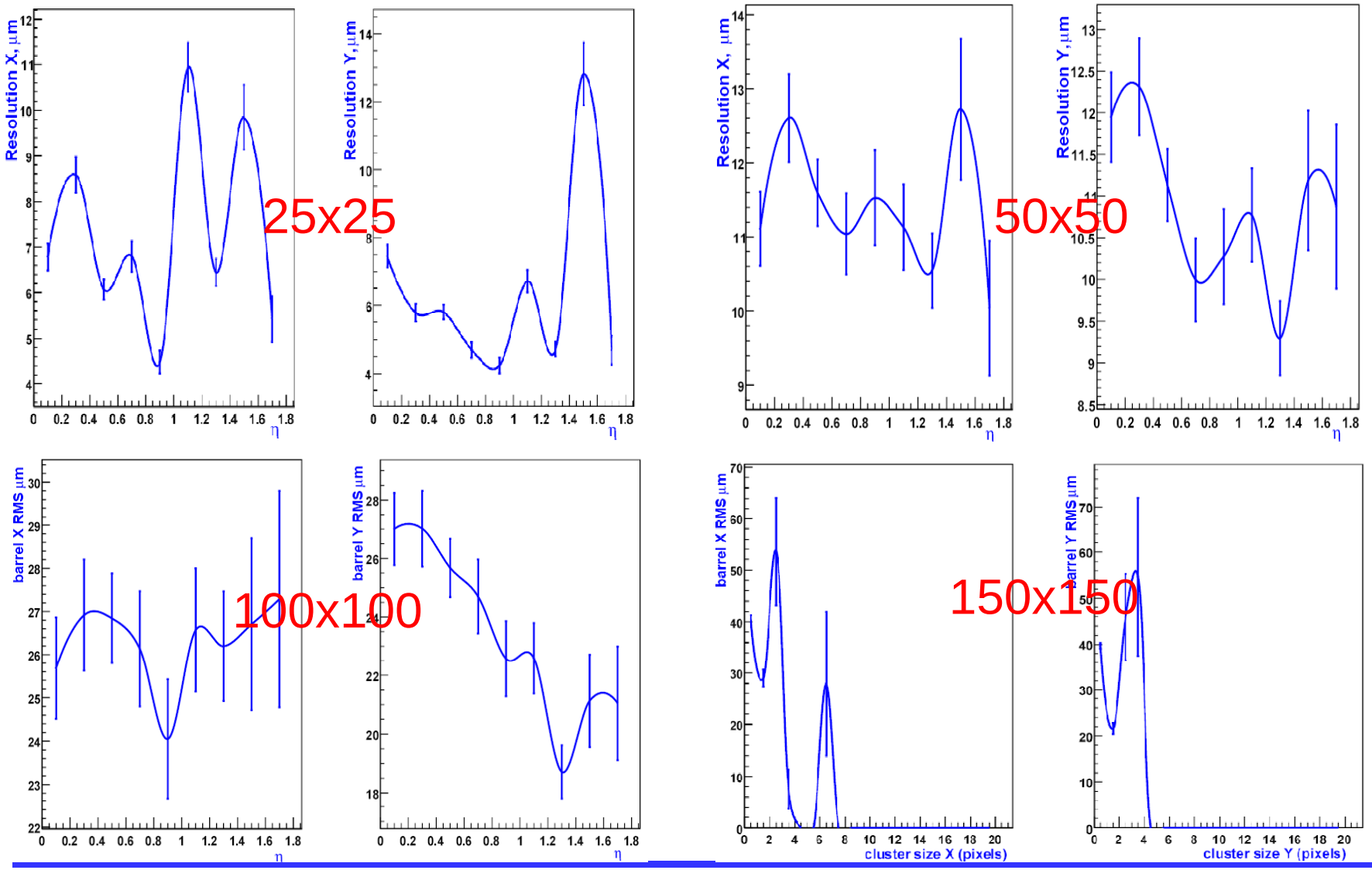


100x100

Resolution = RMS of residual.

Hit resolution depends on cluster sizes and polar angle of hit

X, Y Resolution vs. Eta (barrel)



RecHit validation

- current version of clustering processor validates all recHits obtained by using approximate hit angles from module position;

Tasks

- determination of recHit position by using incidence angles of reconstructed track
- recHit validation by using parameters of reconstructed tracks;
- errors estimation by using errors dependences on cluster sizes and on polar angle.

Summary

- Classes Detector, DetUnit and BarrelDetUnit are developed to use at digitization and clustering processors;
- Pixel digitizer for rectangular det. units is based on CMS Software and implemented in LCIO/MarlinReco framework;
- Pixel clusterizer for rectangular det. units including standard Cluster Parameter Estimator (CPE) is based on CMS Software and implemented in LCIO/MarlinReco framework;
- Pixel, seed and cluster thresholds are investigated
- Hit reconstruction efficiency is studied
- Validation plots for reconstructed hits are presented.

Plans

- Study hit reconstruction vs. noise, inefficiency, threshold parameters
- Improving recHit position estimators
- Development of error estimators
- Pixel and strip DetUnit classes for FTD layers
- Strip digitizers and clusterizers for FTD and SIT
- Performance study together with track finding processors.