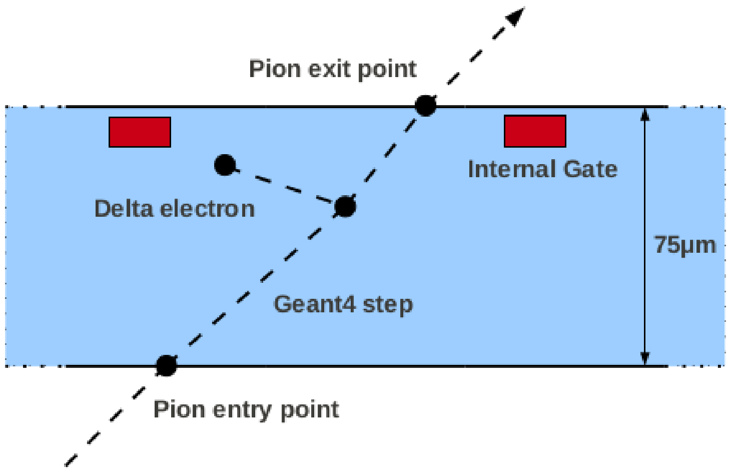
New ideas on PXD hit reconstruction and calibration from beam data

B. Schwenker

University of Göttingen

PXD hit reconstruction

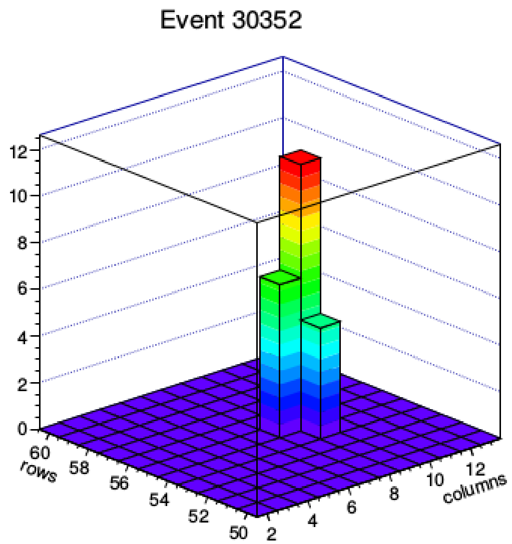
Particle hits sensor



Digitization

[all processes in sensor and readout]

Pixel raw data



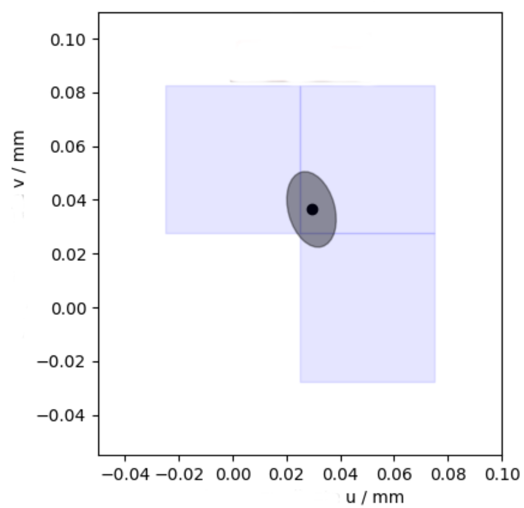
Clustering

[group neighboring pixels]

Hit reconstruction

[find confidence ellipse for intersection]

Pixel Hit

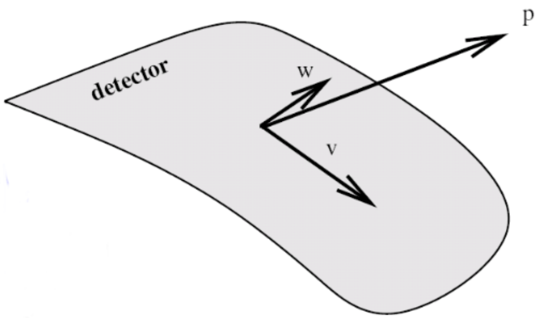


Track fitting [Kalman filter]

[Use pixel hit to improve estimate of track state]

Pixel hits and the Kalman filter

5D track state on sensor midplane:



$$x = (\tan\theta^u, \tan\theta^v, u^x, v^x, q/p)^T$$

2D pixel hit coordinate + covariance matrix:

$$m = (u^m, v^m)$$

$$V = \mathrm{Cov}\,(u^m - u^x, v^m - v^x) = \begin{pmatrix} V_{uu} & V_{uv} \\ V_{vu} & V_{vv} \end{pmatrix}$$

Improve predicted track state using pixel hit from sensor k:

$$S_k = H P_k^- H^T + V_k$$
$$K_k = P_k^- H^T S_k^{-1}$$
$$\bar{x}_k = \bar{x}_k^- + K_k[m_k - H\bar{x}_k^-]$$
$$P_k = P_k^- - K_k S_k K_k^T$$

:- The Kalman filter needs unbiased hit coordinates

:- and consistent (not too large and not too big) hit covariance matrix

:- The Kalman filters does not tell us how to get these numbers.

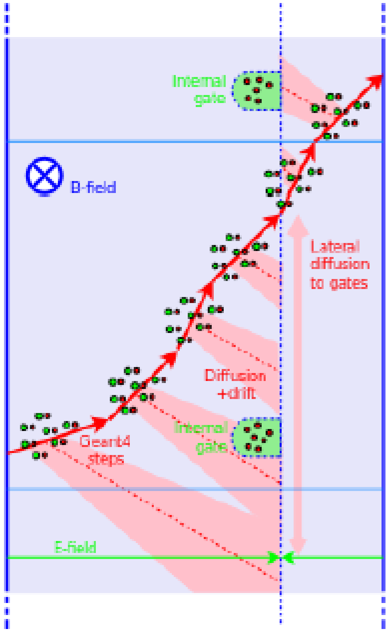
Looking for some guidance

:- We have our digitizers:

Detector response = random numbers + detector physics

Energy loss straggling, Lorentz effect, drift + diffusion, el. Noise, ADC ...

:- One can formalize this idea using recursive Bayesian filters



[PXDDigitizer]

Given a cluster c_k at plane k, the ‘filtered’ distribution for the track State can be computed by Bayes rule:

$$p(x_k|c_{1:k}) = \frac{1}{Z_k} p(c_k|x_k) p(x_k|c_{1:k-1})$$

Filtered distribution

Predicted distribution, using clusters on past sensors.

Measurement model (cond. Pdf) = digitizer

Bayesian cluster shape filter

:- The ‘typical’ tracking scenario:

→ The KF predicted track state has imprecise information on the intersection point (relative to precision of the pixel hit)

→ The KF contains precise information on momentum and incidence angles into sensor.

:- Hit reconstruction can be conditioned on ‘beam’ condition data from KF

$$\beta_k = (\tan\theta_k^u, \tan\theta_k^v, q/p_k)^T$$ → [already available on master (P. Kodys)]

:- We can compute cluster moments from measurement model (in principle)

$$m_k(c_k|\beta_k) = \int Hx_k p(c_k|x_k) du_k^x dv_k^x$$ → [input to KF for track fitting]

$$V_k(c_k|\beta_k) = \int (Hx_k - m_k(c_k|\beta_k)) (Hx_k - m_k(c_k|\beta_k))^T p(c_k|x_k) du_k^x dv_k^x.$$

Bayesian cluster shape filter

:- This looks infeasible, but we have discrete translation symmetry to our help

Shift cluster by m,n pixel units

$$p\left(c|x\right) = p\left(c'|x'\right)$$

$$c' = \mathbf{T}(m,n)c = \{vc_i + m, uc_i + n, s_i\}_{i=0,..,n}$$

$$x' = \mathbf{T}(m,n)x = \left(\tan\theta^u, \tan\theta^v, u^x + nP_u, v^x + mP_v, q/p\right)^T$$

Shift intersection by m,n pixel pitches

:- This will only hold for well designed and well calibrated detector → other topic ;)

:- In case symmetry holds, we only need cluster moments for a much smaller subset of clusters called shapes.

Shape == Cluster with min(ucells) = 0 && min(vcells=0)

Training data and bootstrapping

:- We can do all computations from sufficiently large training data for some beam condition.

$$D(\beta) = \{\bar{x}_i^-; P_i^-; c_i\}$$

Pred. track state
+ covariance

Cluster caused by
track.

:- Training data can originate from real experiment or generated from simulation

TrueHits + related Digits | ‘fitted’ track states + close-by Digits

:- Number of tracks in training data should not be too large (<1Mio).

:- The PXD uses 8bit ADC codes → the number of shapes is too large

:- In order to reduce the number of shapes, we need some sort of ‘shape clustering’

Digital labels and their moments

:- One very robust shape clustering is simply ignoring the signals → digital labels

$$l_{\mathrm{D}}(s) = \texttt{'-'.join('V:'+str(d[0])+'.U:'+str(d[1]) for d in } s \texttt{)}$$

Here a label is really a string literal. For example: V:0.U:0 == one digit cluster

:- The number of digital labels is typically rather small (<<100) for a given beam condition

Label probability

$$p(l|\beta) = \frac{|D(\beta, l)|}{|D(\beta)|}$$

Label hit coordinate

Shifts from cluster to shape

$$o(l|\beta) = \frac{1}{|D(\beta, l)|} \sum_{i \in D(\beta,l)} H\bar{x}_i^- - \mathbf{F}(c_i)$$

#tracks / labels in data

Label covariance matrix

$$V(l|\beta) = \frac{1}{|D(\beta, l)| - 1} \sum_{i \in D(\beta,l)} (H\bar{x}_i^- - \mathbf{F}(c_i) - o(l|\beta))(H\bar{x}_i^- - o(l|\beta))^T - A(\beta)$$

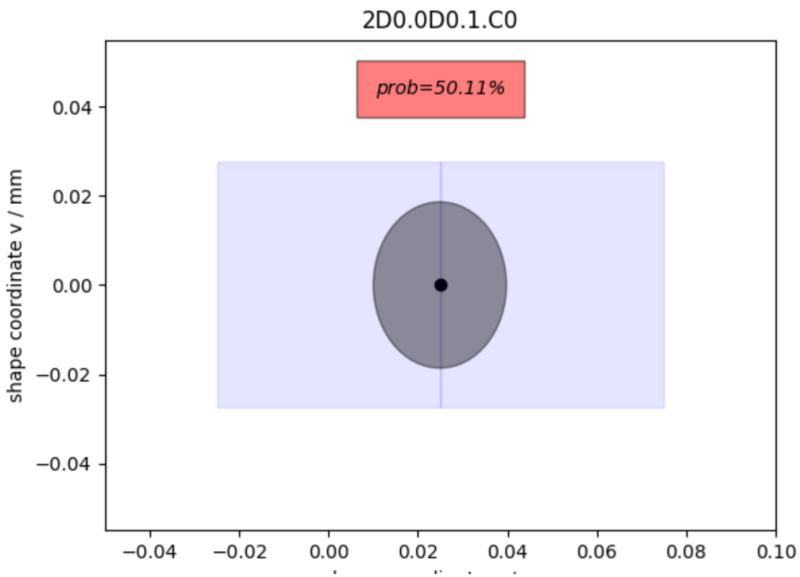
Some examples

:- We take parameters from angular scan in a PXD test beam as reference

:- 4GeV electrons, 200k single track events, B=0T
:- PXDDigitizer parameters: (Pixelkind 55x50um^2)

ADCFineMode	: False
Gq	: 0.77nA/e
SourceBorder	: 6.3um
DrainBorder	: 6.3um
ClearBorder	: 4.2um
El. Noise	: 150e
ChargeThreshold	: 5ADU

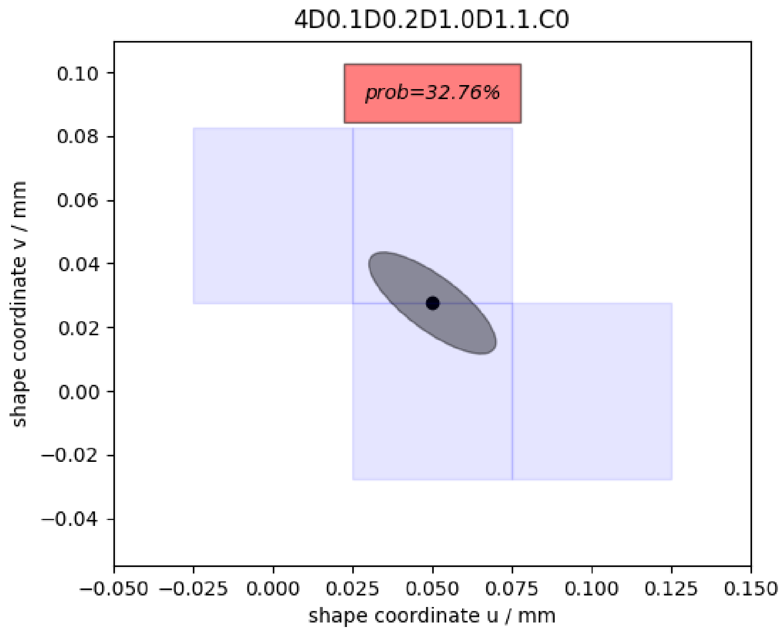
Some examples



:- Sim data for test beam situation (theta=90° / phi=60°)

:- ~50% of all digital labels are like that

:- Bayesian filter gives positions and 2x2 covariance matrix



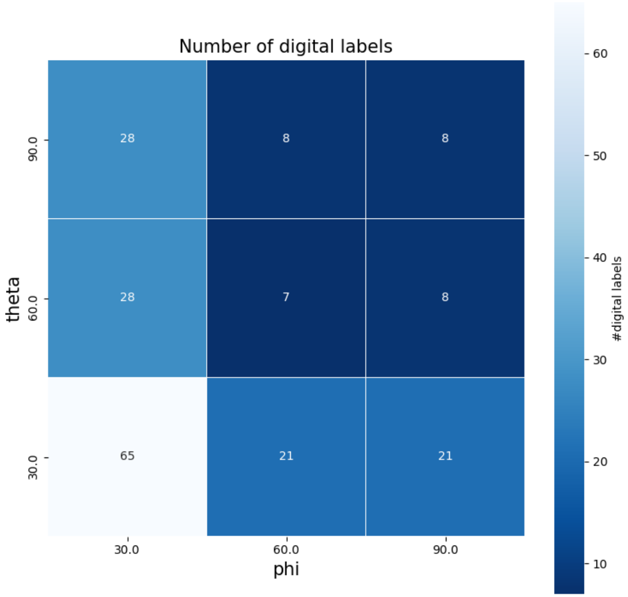
:- Sim data for larger incidence angles (theta=60° / phi=30°)

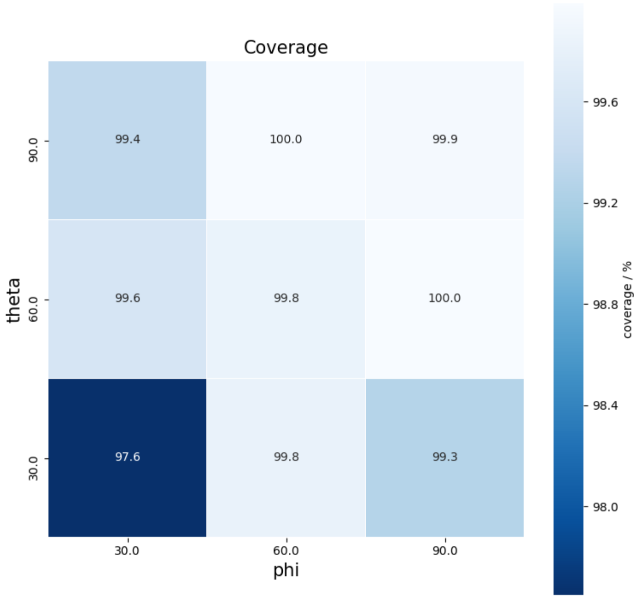
:- most important single label (~33%)

:- Remember: Estimate of UV correlation Based on:

- :- Geometry of firing pixel cells
- :- Conditioned on incidence angles

Overview: Results from digital labels



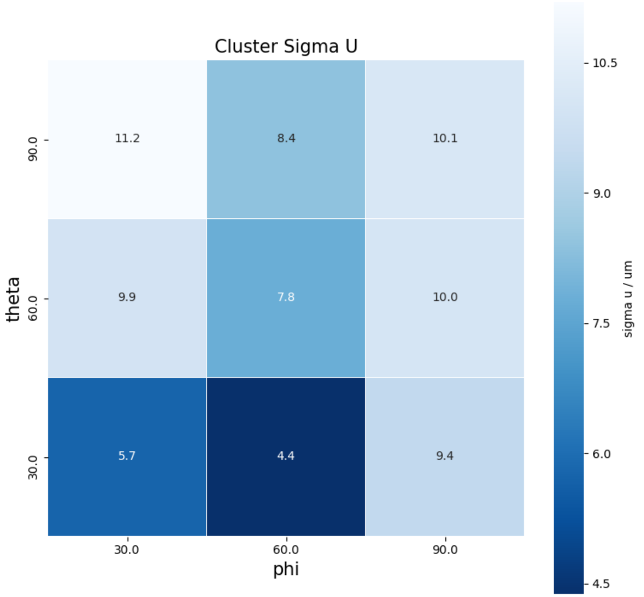


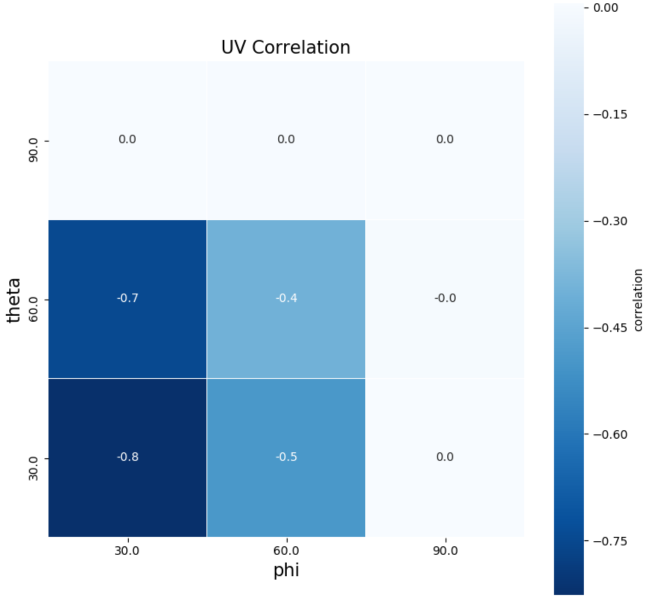
:- Number of labels grows with incidence angles into sensor

:- Require >200 to accept label and estimate corrections

:- Coverage = Prob to find correction given some cluster

Overview: Results from digital labels





:- Weighted average of cluster sigmaU over all digital labels

:- Weight = Label probability

:- Average uv correlations when both incidence angles non zero

:- Correlations significant for certain beam conditions.

How to incorporate digit signals?

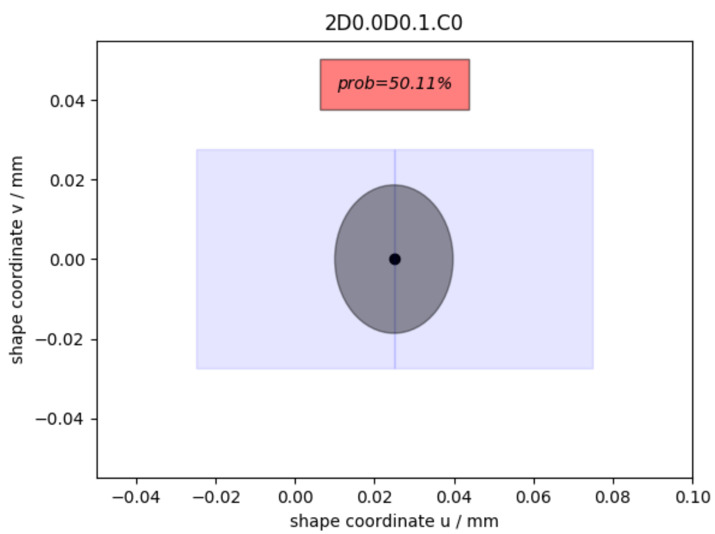
:- Digital labels provide useful clusters of shape, but sometimes too big.

→ too many shapes in digital label → significant loss of resolution

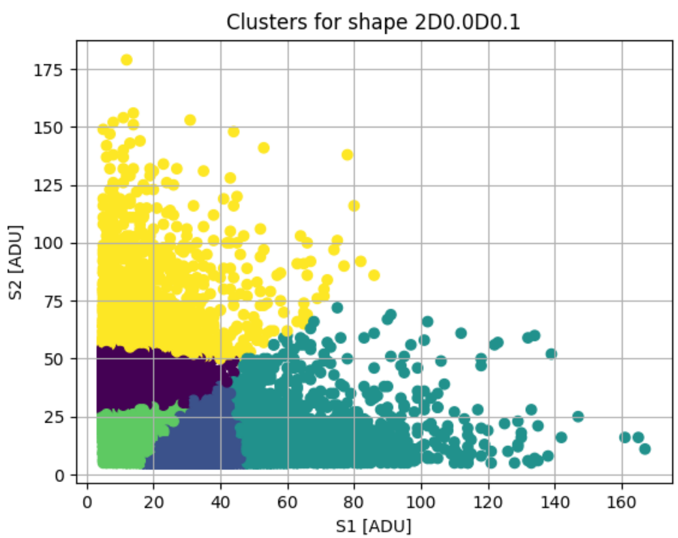
:- Idea: further sub division of shapes inside the same digital label

→ for example using k-means clustering

:- Example: ‘2u’ cluster at theta=90° / phi=60°



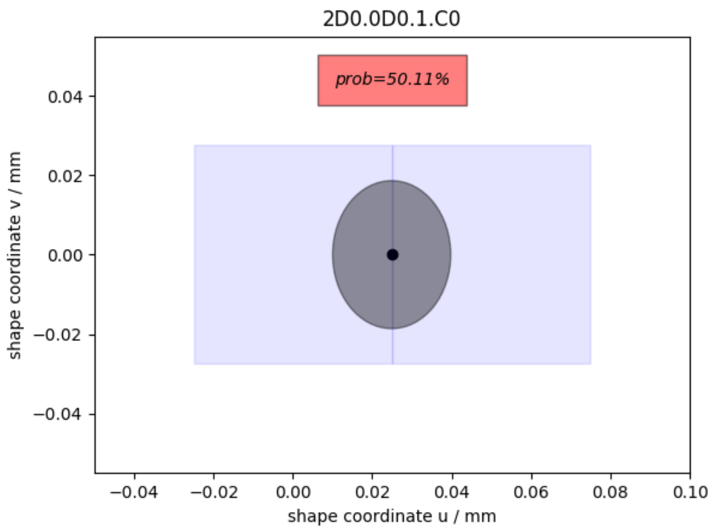
k=5

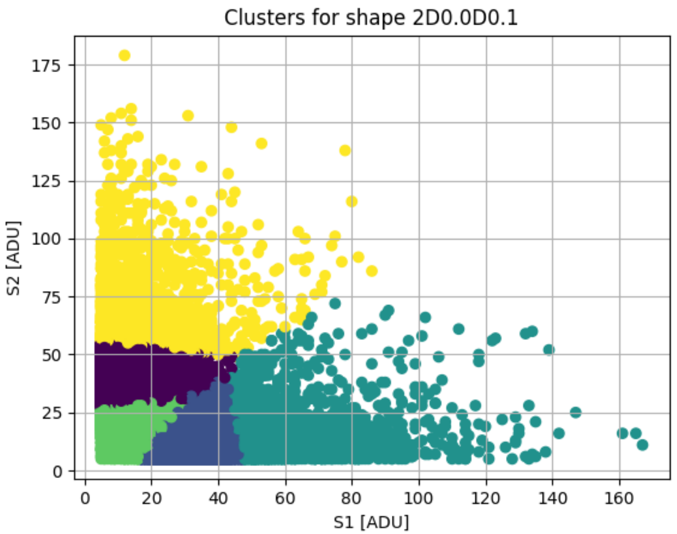


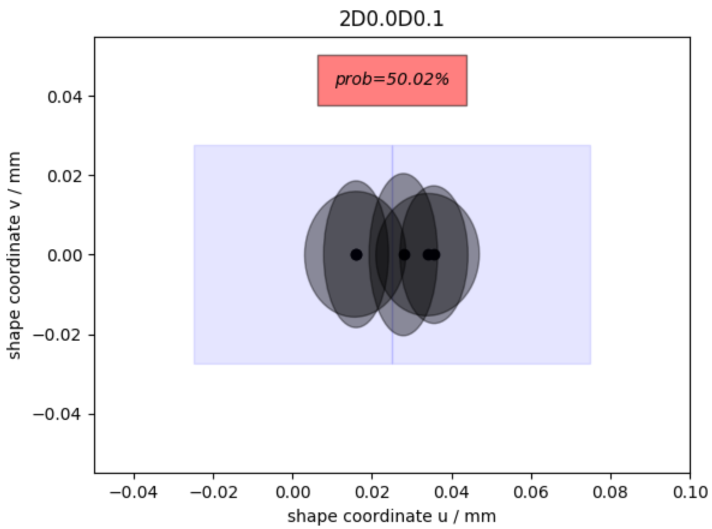
How to incorporate digit signals?

k=1

k=1





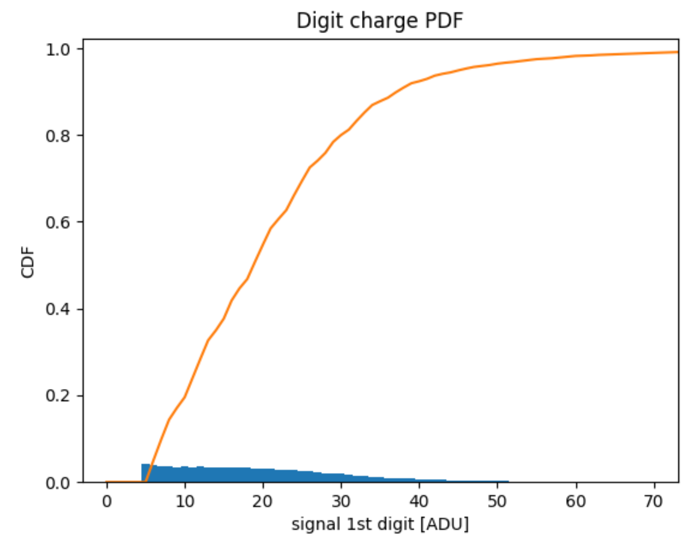


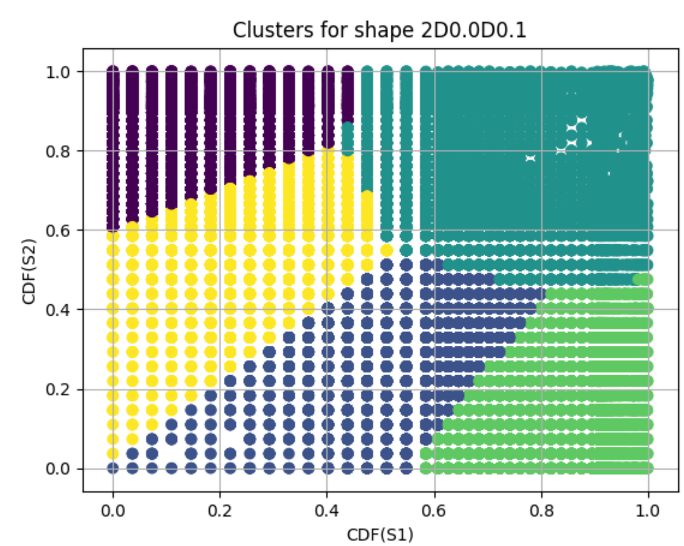
:- Consider now the results of
K-means clustering as labels

Some pit falls of K-means

K means works best when density of points is constant → we have Landau tails

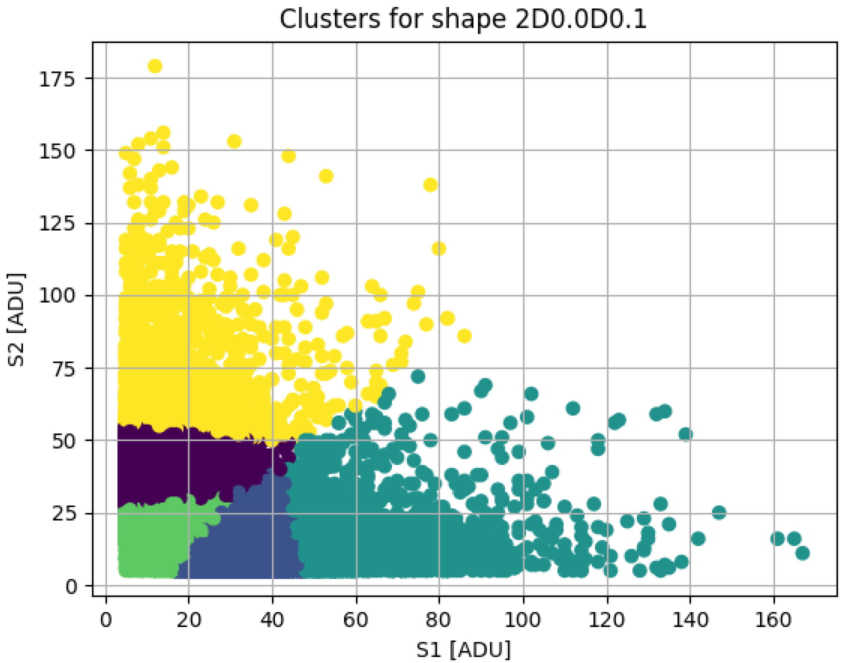
→ Transform digit signals before clustering
→ Not fully implemented yet.



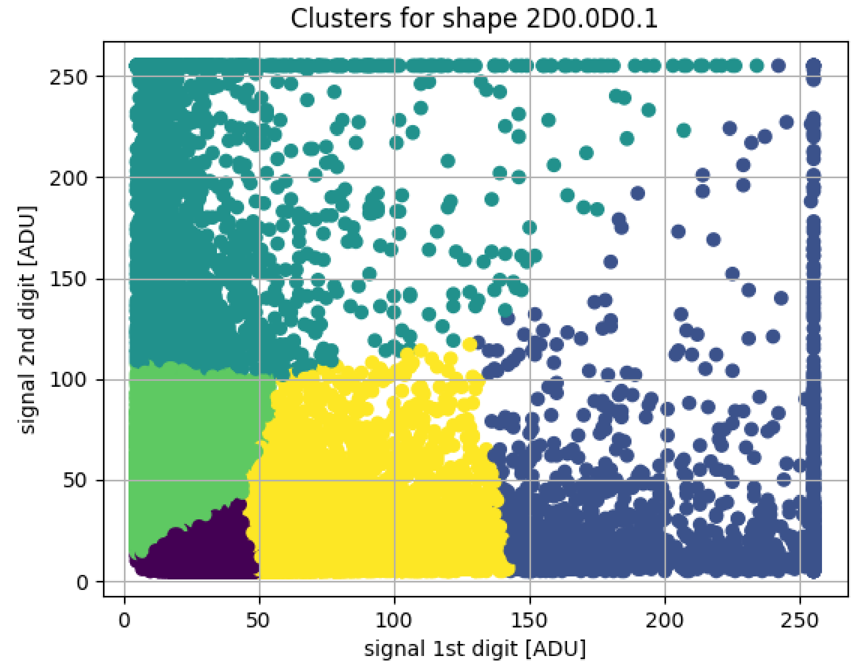


Some artefacts in simulation

TB data (Nov 15)



Basf2 simulation



:- basf2 simulation tends to produce too many
very large signals

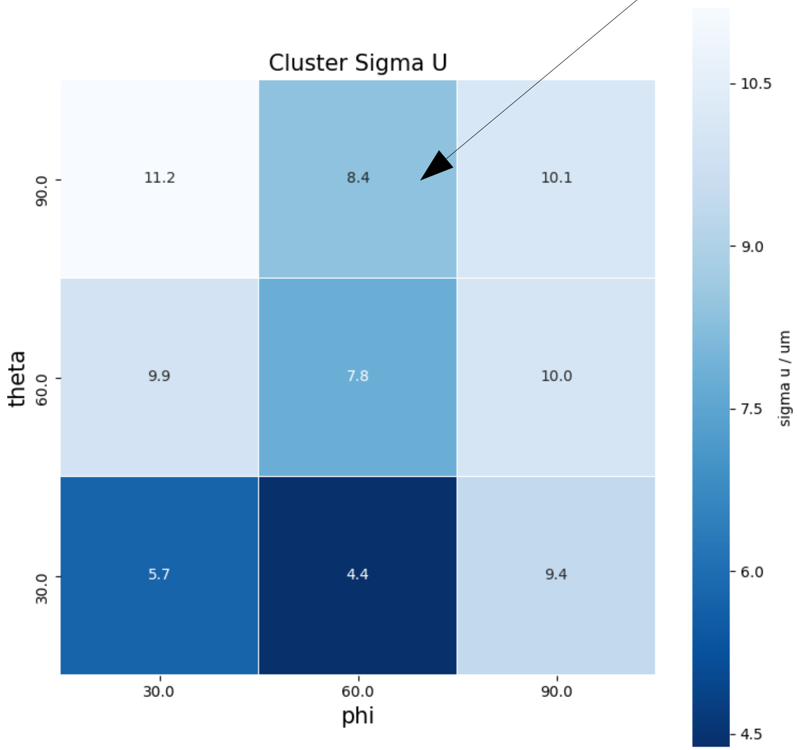
:- probably happens when PXDSimHits are produced ...

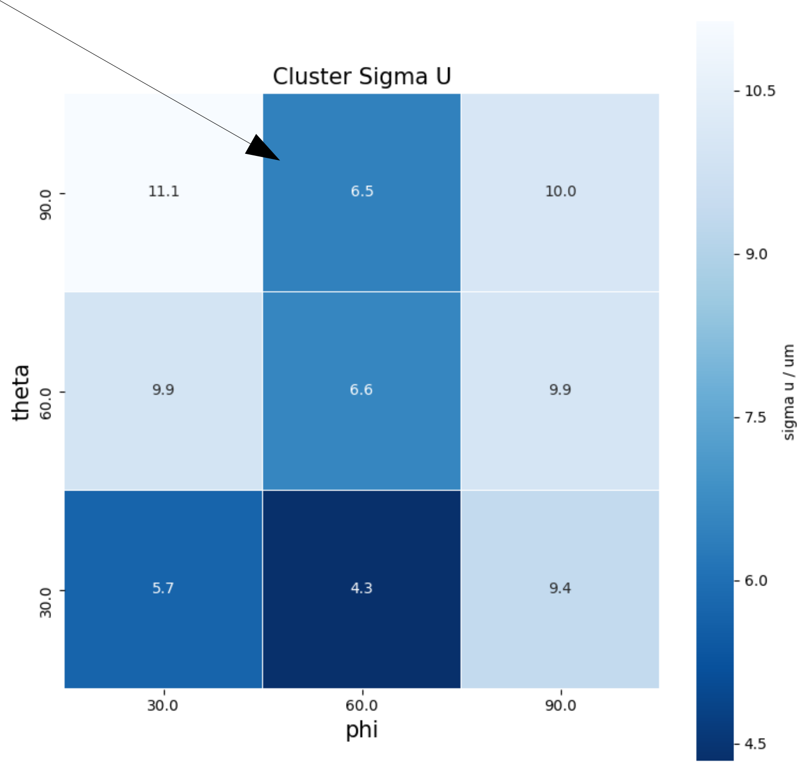
Improvements from K means (using signals directly)

K = 1

Improvements visible
(but in many cases simpel clustering is unreliable ...)

K = 5





Summary & Conclusion

:- Presented new approach for hit reconstruction in pixel (strip) detectors

- estimates full 2x2 covariance matrix
- training on real data and simulation possible
- no ‘heuristics’ needed; instead method is data driven

:- Some aspects still need a bit of work

- shape clustering directly with K means is not ideal way
- pre-processing needed: normalize signals before clustering
- different clustering methods other then K means (???)

:- Full blown implementation in pxd sw needs to be considered

- Current cluster shape correction by P. Kodys works differently

PXD calibration from beam data

:- PXDDigitizer parameters:

ADCFineMode	: False
Gq	: 0.77nA/e
SourceBorder	: 6.3um
DrainBorder	: 6.3um
ClearBorder	: 4.2um
El. Noise	: 150e
ChargeThreshold	: 5ADU

:- All of these parameters affect cluster shapes (→ hit reconstruction)

:- Need a data driven way to estimate these paramters from beam data

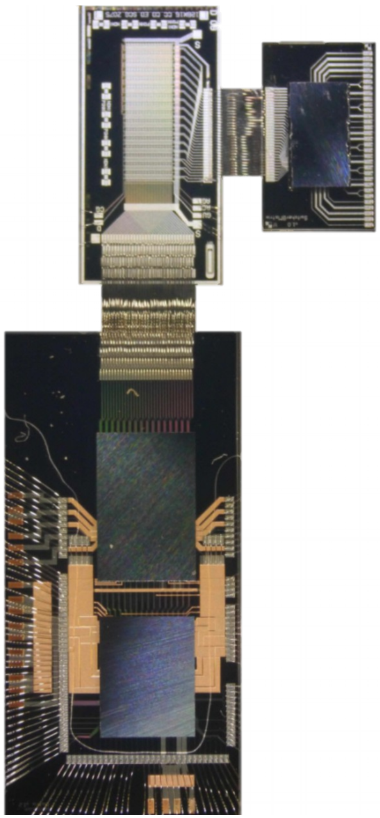
:- Tweak parameters q until label probabilities from reference data (from experiment) and simulated data match:

$$M(q) = \sum_i \sum_l |p(l|\beta_i) - p'(l|\beta_i, q)|^2$$

:- Initial implementation working and tested with beam data from Nov. 15

Backup

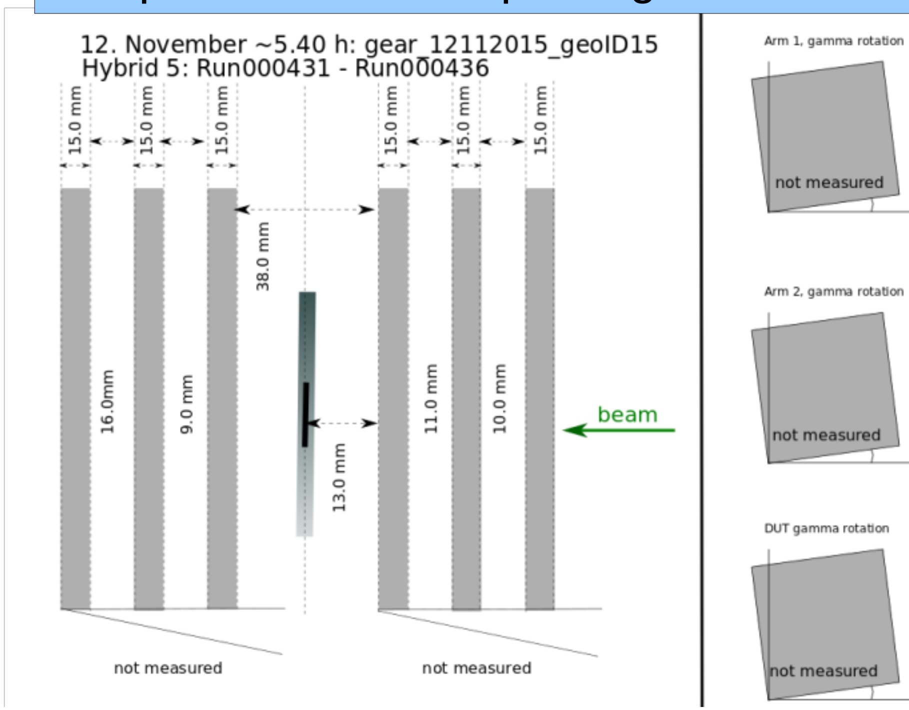
Small PXD9 @ DESY (Nov. 2015)

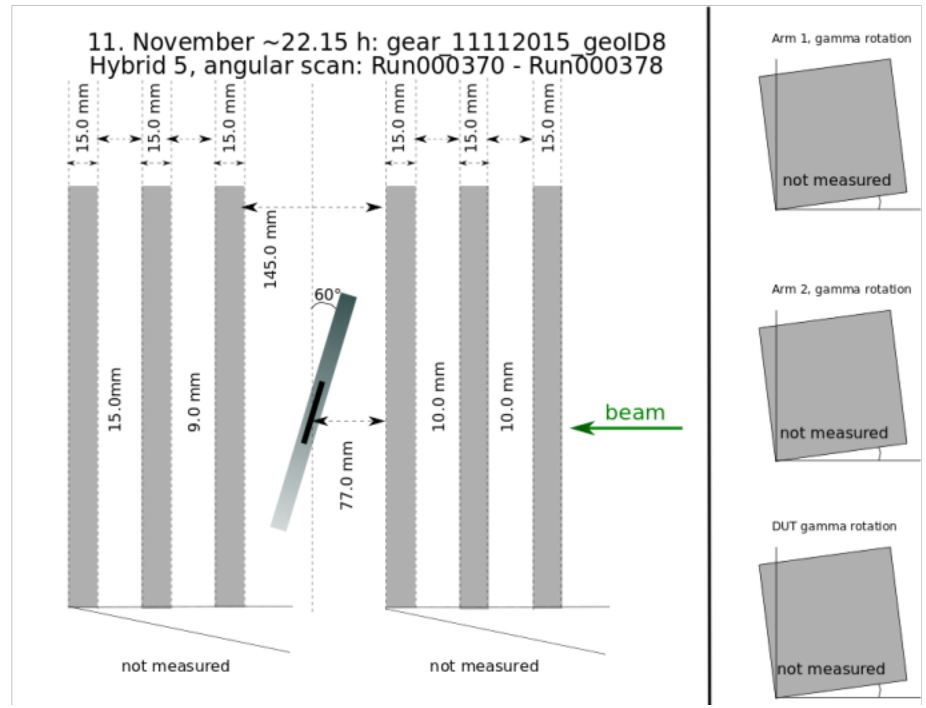


- First Belle II type matrix in a test beam with EUDET telescope
- Called Hybrid5 (H5)
- PXD9 small Belle II type matrix
  - Pixel pitch: 50x55 μm^2 (→ layer 1 PXD)
  - Gate length: 5µm (→ like PXD)
  - thin gate oxide (→ like PXD)
- Still a very valuable data set
  - High resolution telescope (in-pixel study)
  - High statistics: Millions of (precise) tracks matched to PXD cluster
  - Angular scan: Tilt of PXD sensor against beam (up to 60 degree)

Telescope geometries

https://docs.google.com/spreadsheets/d/1Ob5KCRMYuoHW5TROI7iMACItBA29Jw7i2kWqMmwhCbA/edit?pli=1#gid=491395880





:- small distances to keep tel. interpolation error small.

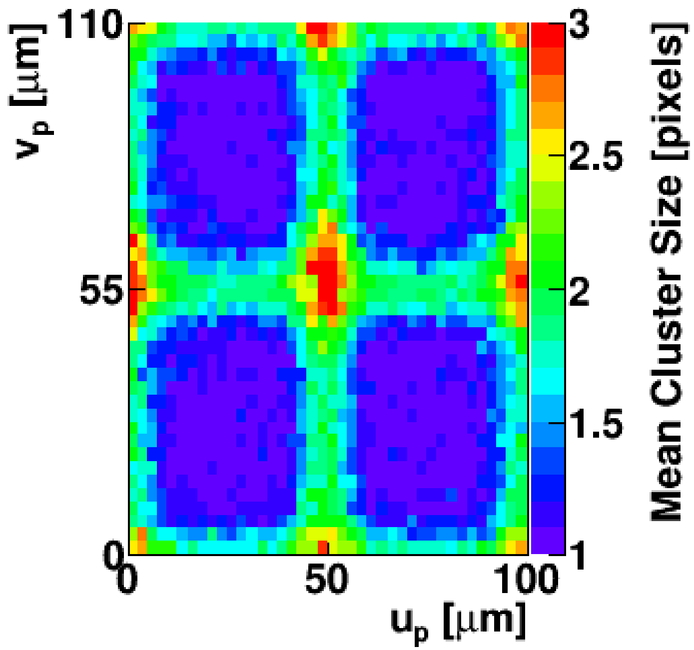
:- Hybrid 5 mechanics a bit bulky → larger distances to PXD

:- Rotating Hybrid 5 implies moving arms away and increases material.

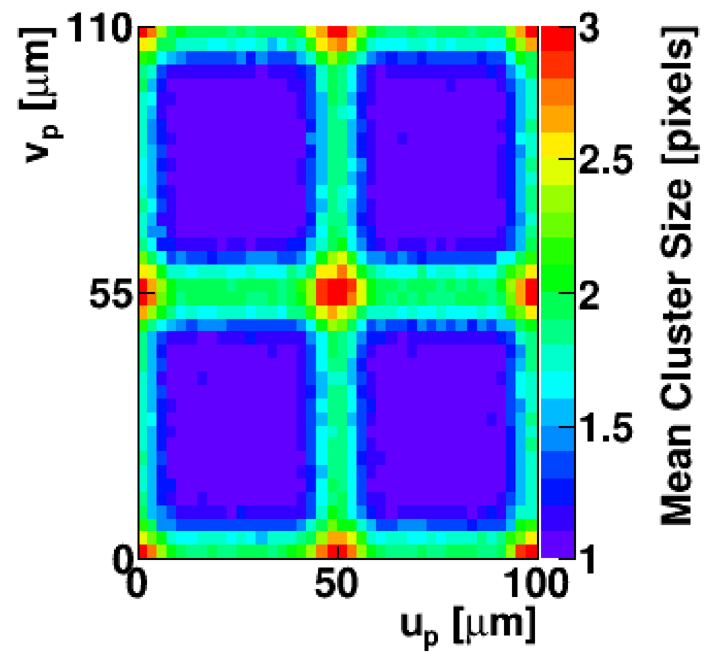
:- Different distances for all angles, still interpolation errors @ PXD grows

H5: Inter pixel charge sharing

Small PXD9 in test beam



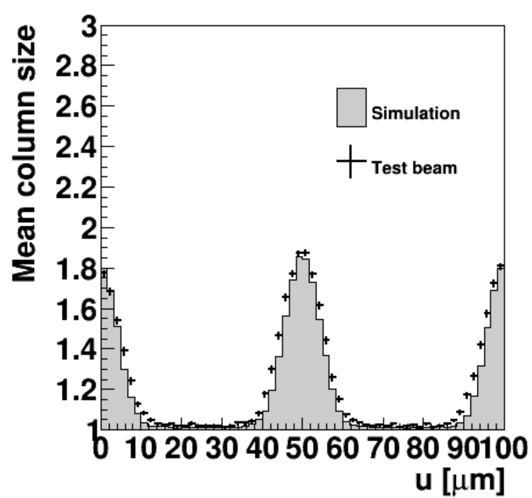
"Tuned" PXD9 Digitizer



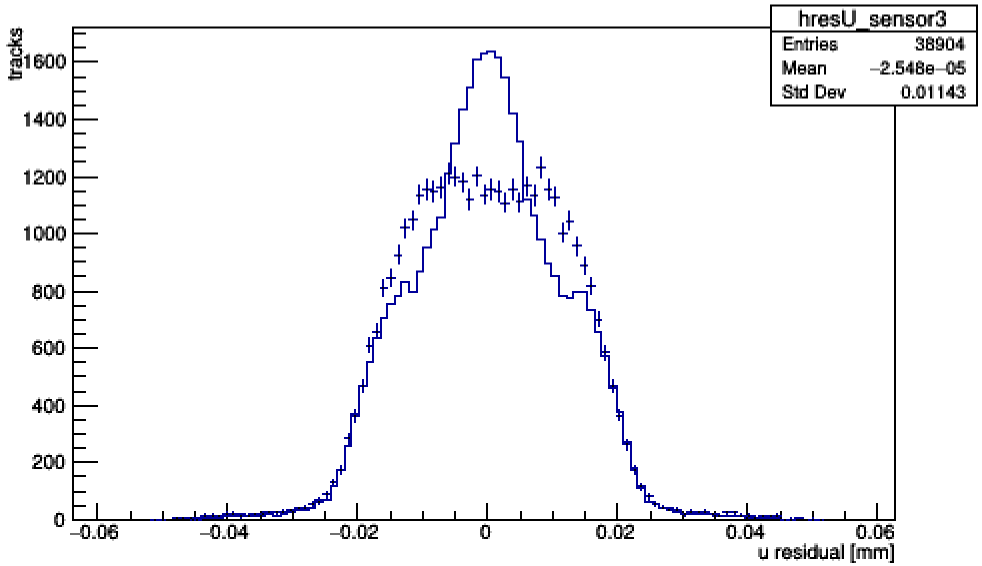
Summary of "tuned" digitizer parameters PXD9 50x55:

:- Charge sharing region between rows: ~12um
:- Charge sharing region between columns: ~12um

Expected resolution for two row cluster ~3.5um



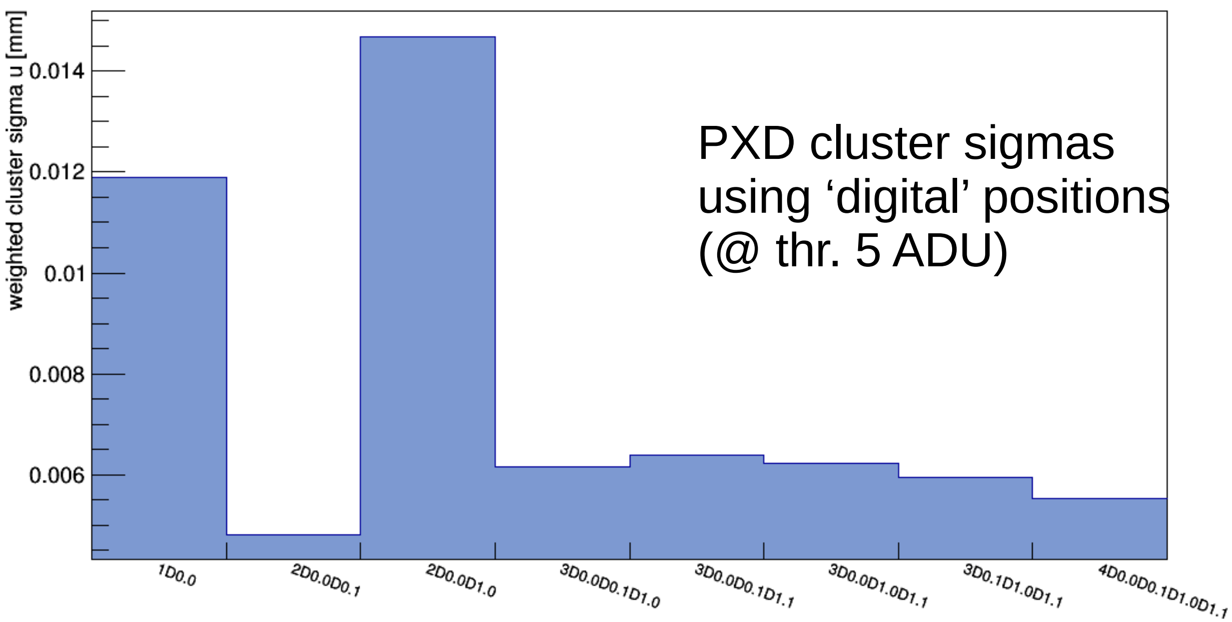
H5: Residuals at perp. incidence



:- compare u residuals using different position reconstructions (PXD)
→ center-of-gravity (crosses)
→ digital (solid line)

:- ‘Digital’: using same method as for M26 sensors (hit thr. 5ADU)

:- Cog performs worse than digital
→ charge sharing restricted to ~10um region between pixels
→ true for close to perp. incidence



:- Cluster sigmas obtained after subtracting tel. Interpolation error

:- double column cluster have sigma ~5um.

:- single pixel cluster ~12um