ROI Algorithm and Test-Beam Results

Giulia Casarosa
INFN - Sezione di Pisa

First *Belle II* Italian Collaboration Meeting

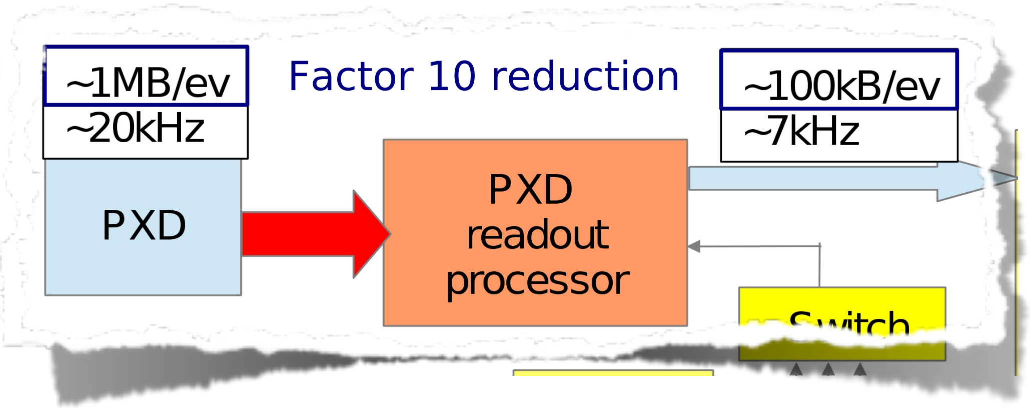
Rome, 9th ~ 10th June 2014

outline

- Region of Interest (ROI) finding in *Belle II*
  - motivation
  - algorithm
  - current expected performances
- VXD Test-Beam @ DESY
  - the setup
  - the results
- Conclusions

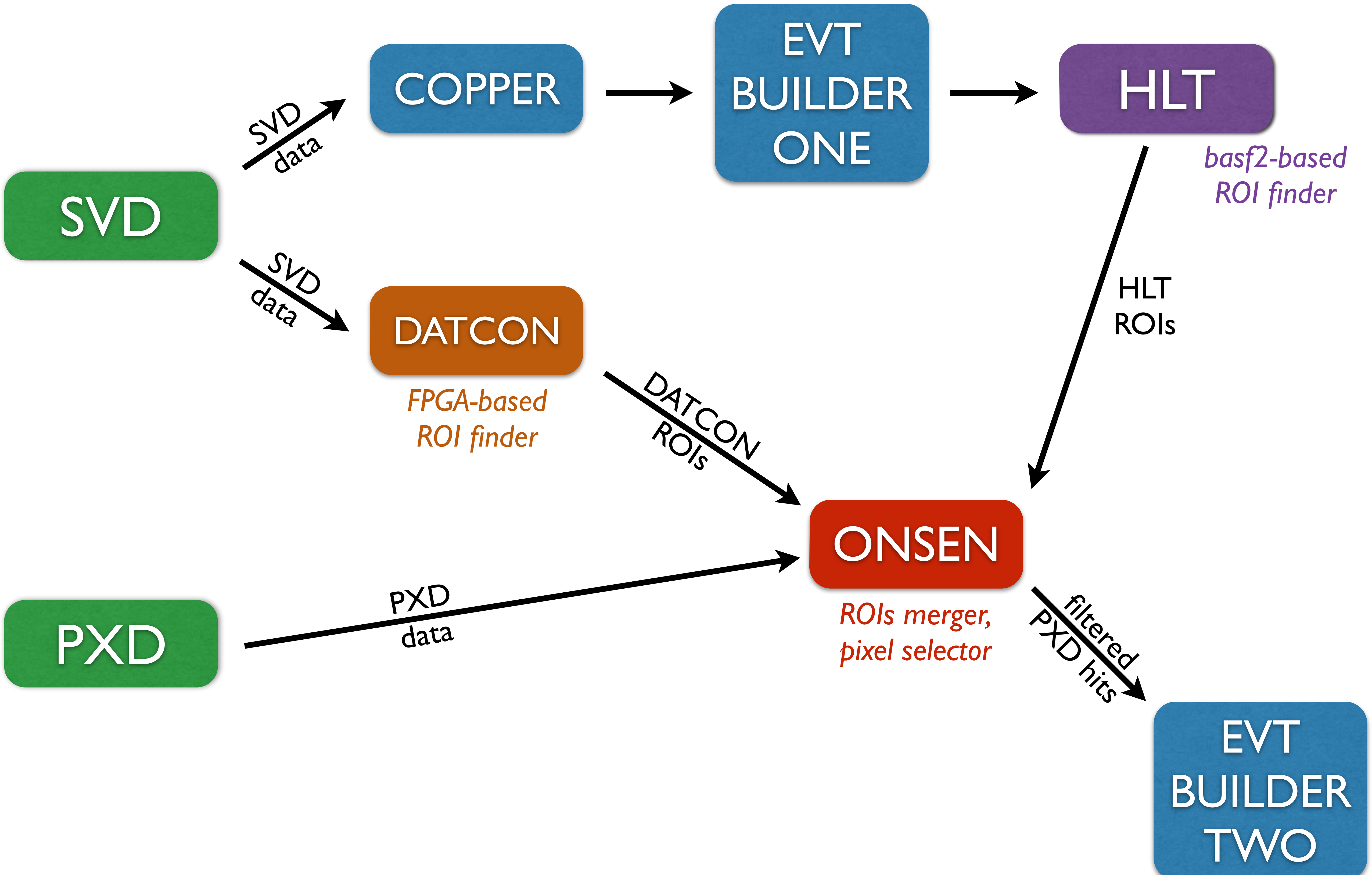
Region Of Interest Finding Motivation

- a very large amount of PXD data is expected at full luminosity
  - ~2M pixels with a ~3% occupancy, 10kHz trigger rate → 20Gb/s of PXD data
- need to reduce the total amount of PXD data to be stored, roughly by an order of magnitude



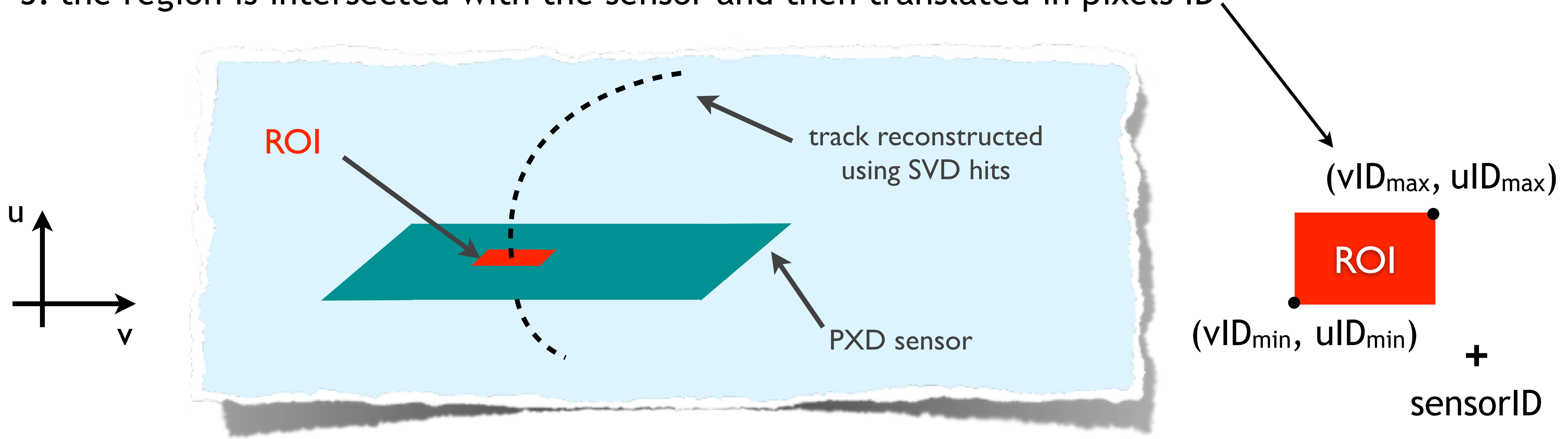
- this operation is done between the Event Builder 1 (all detectors except PXD) stage and Event Builder 2 stage (all detectors + PXD), with inputs from High Level Trigger (HLT)
- two parallel implementation of PXD data reduction are foreseen:
  - software-based solution
  - hardware-based solution exploiting FPGAs

ROIs General Picture



Software-Based ROI Finding Algorithm

1. pattern recognition performed with SVD hits only:
   - ★ TrackCand list produced by *VXDTF* (or *MCTrackFinder* for testing purposes only)
2. fit the TrackCand using the standard kalman filter (genfit) and produce a Track
   - ★ the fit is done in both directions: first inward, then outward
3. the Track is extrapolated on each of the 40 planes containing a PXD sensor
   - ★ obtain an extrapolation point on the plane and the associated statistical errors σ_{stat}
4. a rectangular region is defined given σ_{stat}, a systematic error σ_{syst} and a total number of $\sigma = \mathrm{sqrt}(\sigma^2_{stat} + \sigma^2_{syst})$ in each direction u,v
5. the region is intersected with the sensor and then translated in pixels ID



Definition of the Figures of Merit

➡ Definition of efficiency for PXD Data Reduction:

$$\varepsilon = \frac{\text{\# PXDDigits inside a ROI}}{\text{total \# PXDDigits of TrackCand}}$$

inefficiencies of the pattern recognition are factorized!!

➡ Definition of data reduction factor:

$$r = \frac{\text{< \# pixels in ROI/event >}}{\text{250*768 pixels/module * 40 modules}}$$

➡ execution time: we run on the HLT, we need to be fast: benchmark = 1ms/track

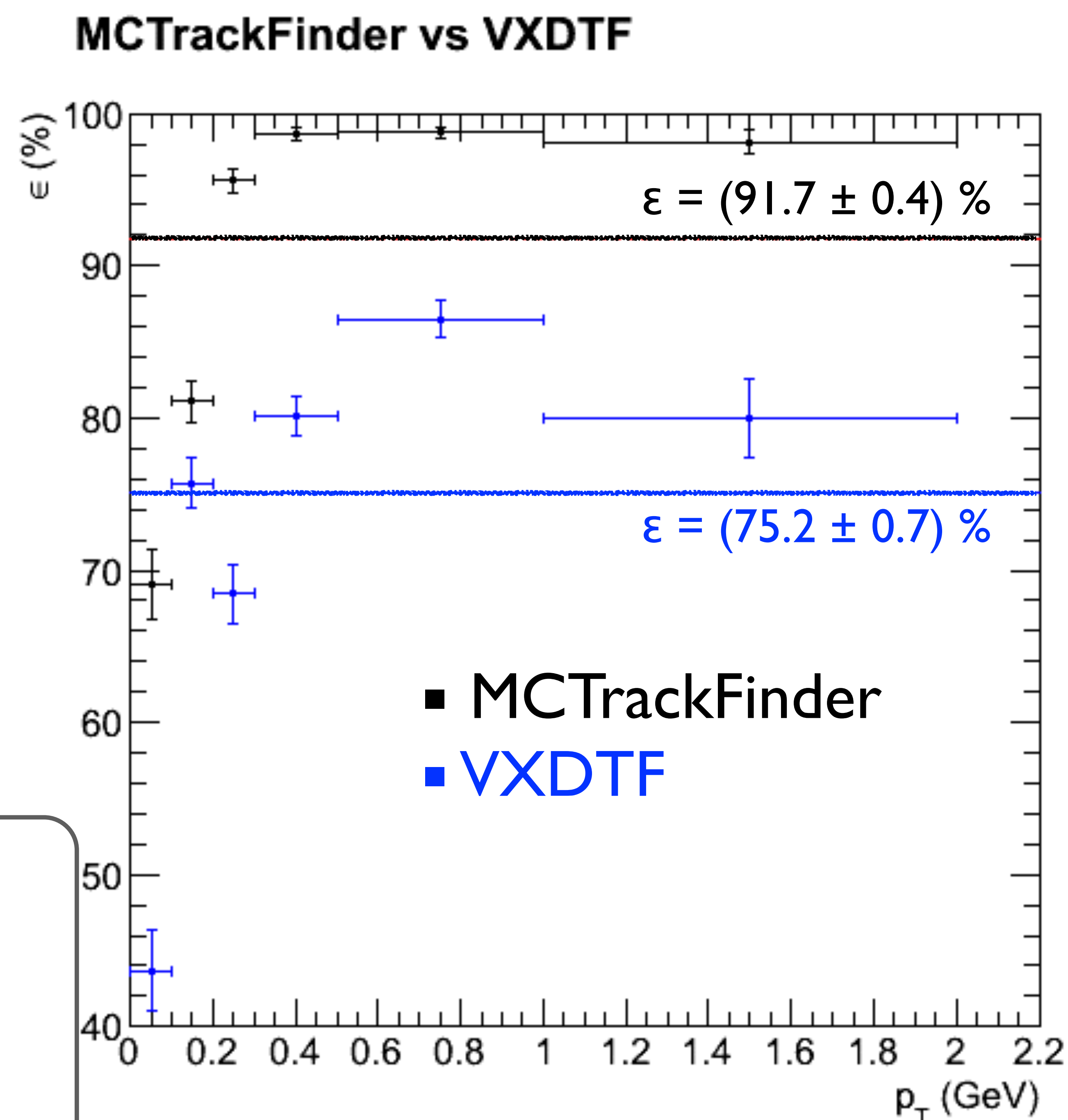
ROI Finding Efficiency

- ➡ We compare the efficiency obtained using the VXDTF with the MCTrackFinder:
  - VXDTF: official track finder
  - MCTrackFinder: uses the *true* hits
- ➡ Efficiency with VXDTF = (75.2±0.7)%
- ➡ Efficiency with the MCTF = (91.7 ± 0.4) %
- ➡ In both cases inefficiency mostly due to failures in fitting the track and finding an intercept with the sensor planes (see next slide)

$$\varepsilon = \frac{\text{\# PXDDigits inside a ROI}}{\text{total \# PXDDigits of TrackCand}}$$

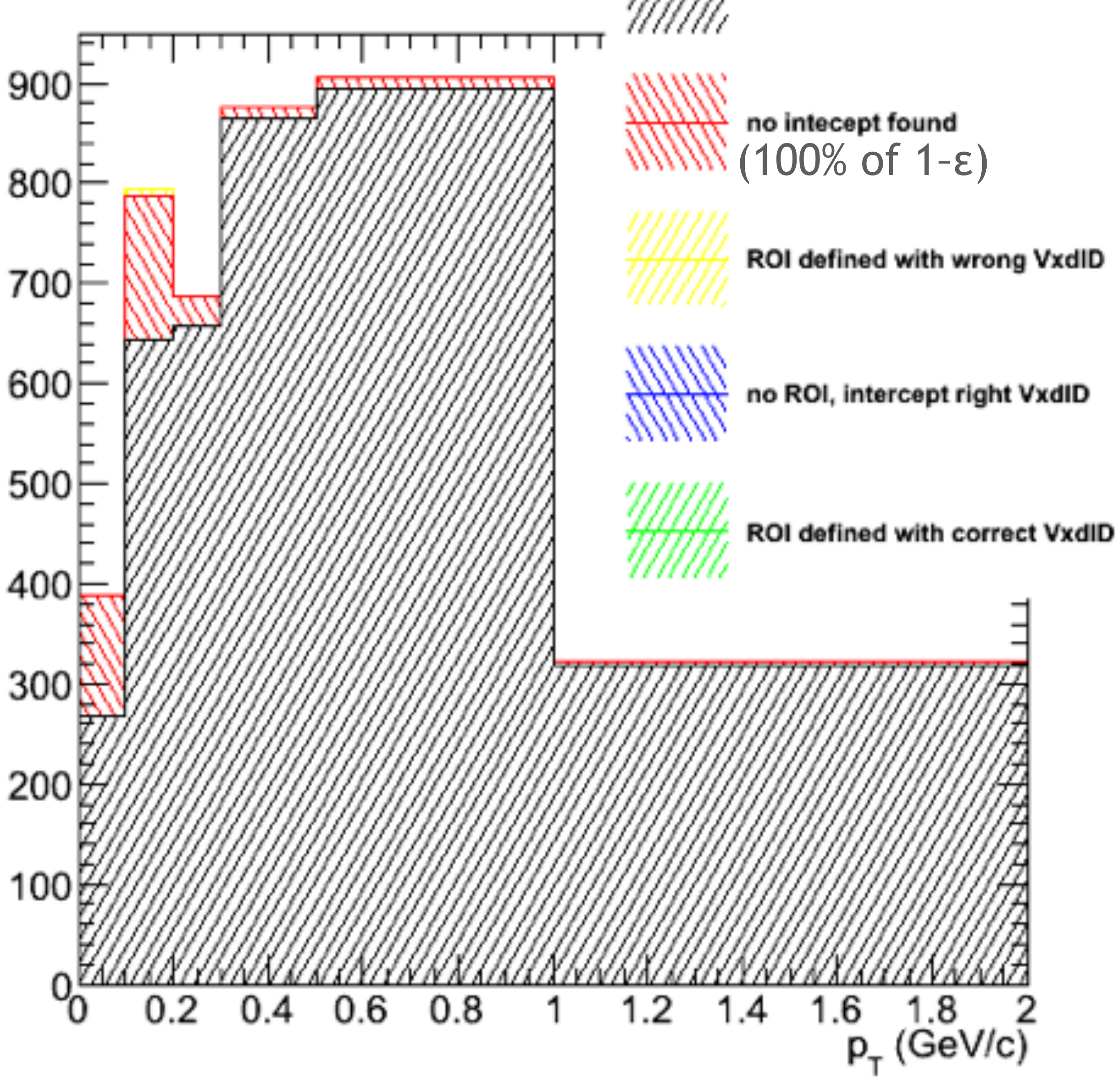
inefficiencies of the pattern recognition are factorized,

but the TrackCand quality is not!

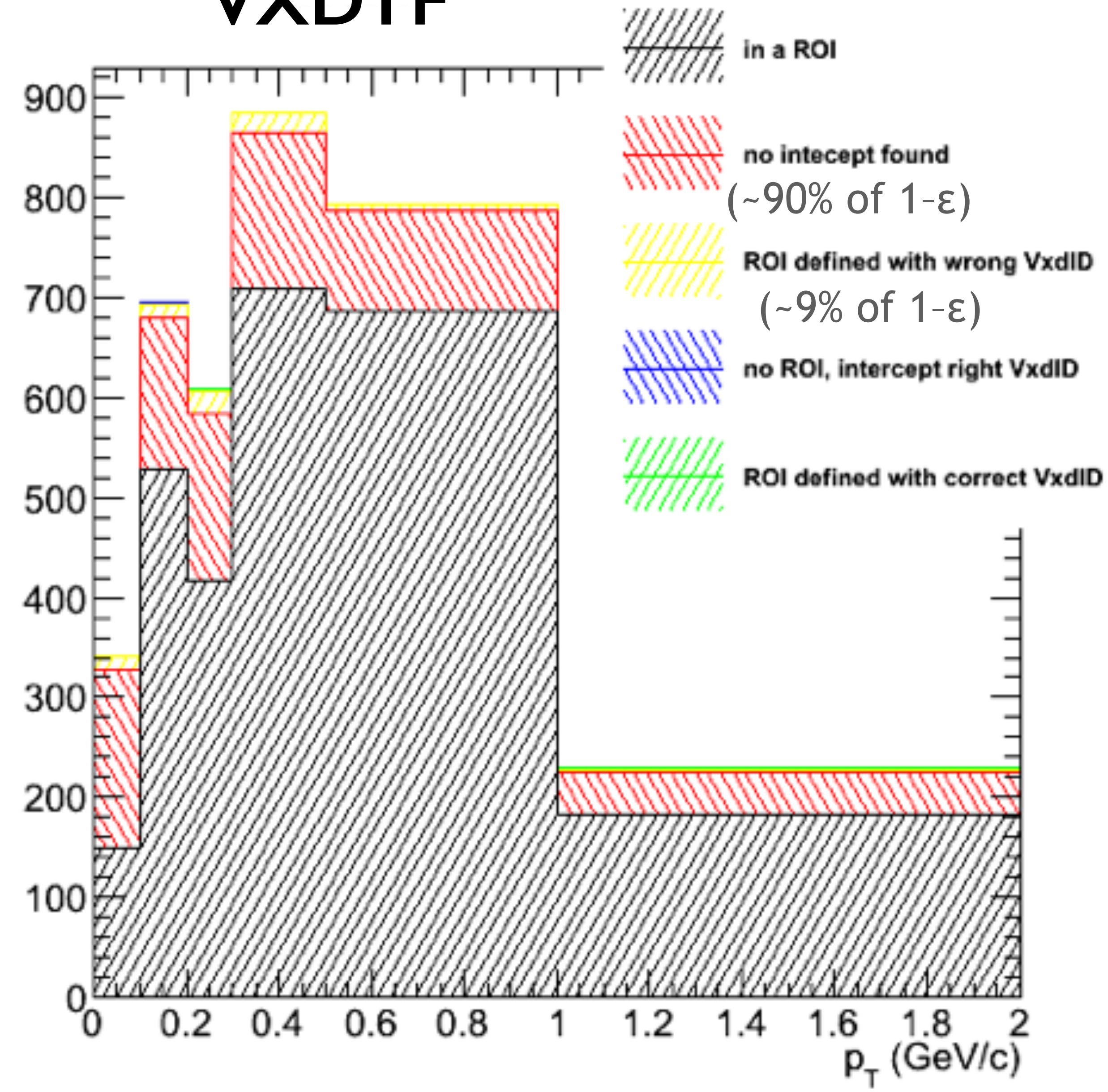


PXDDigits classification

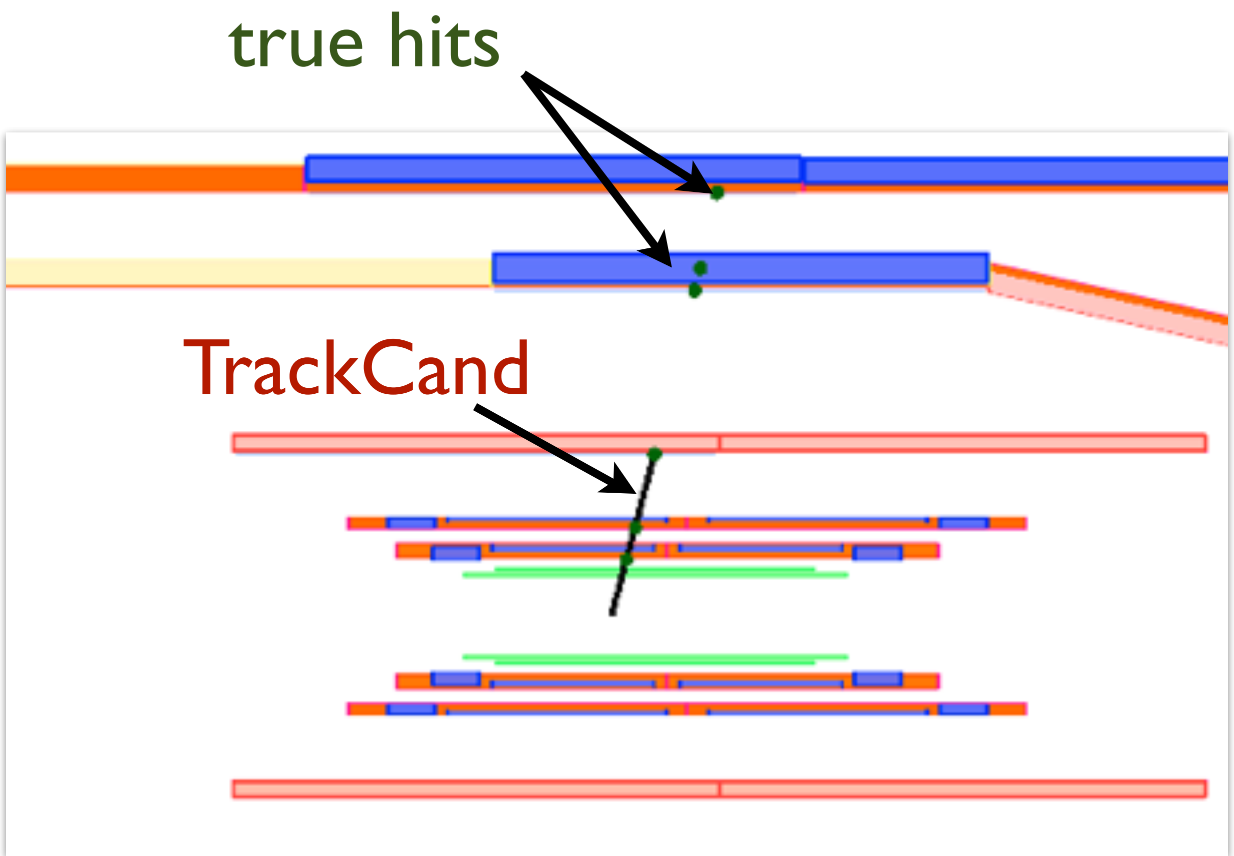
MCTrackFinder

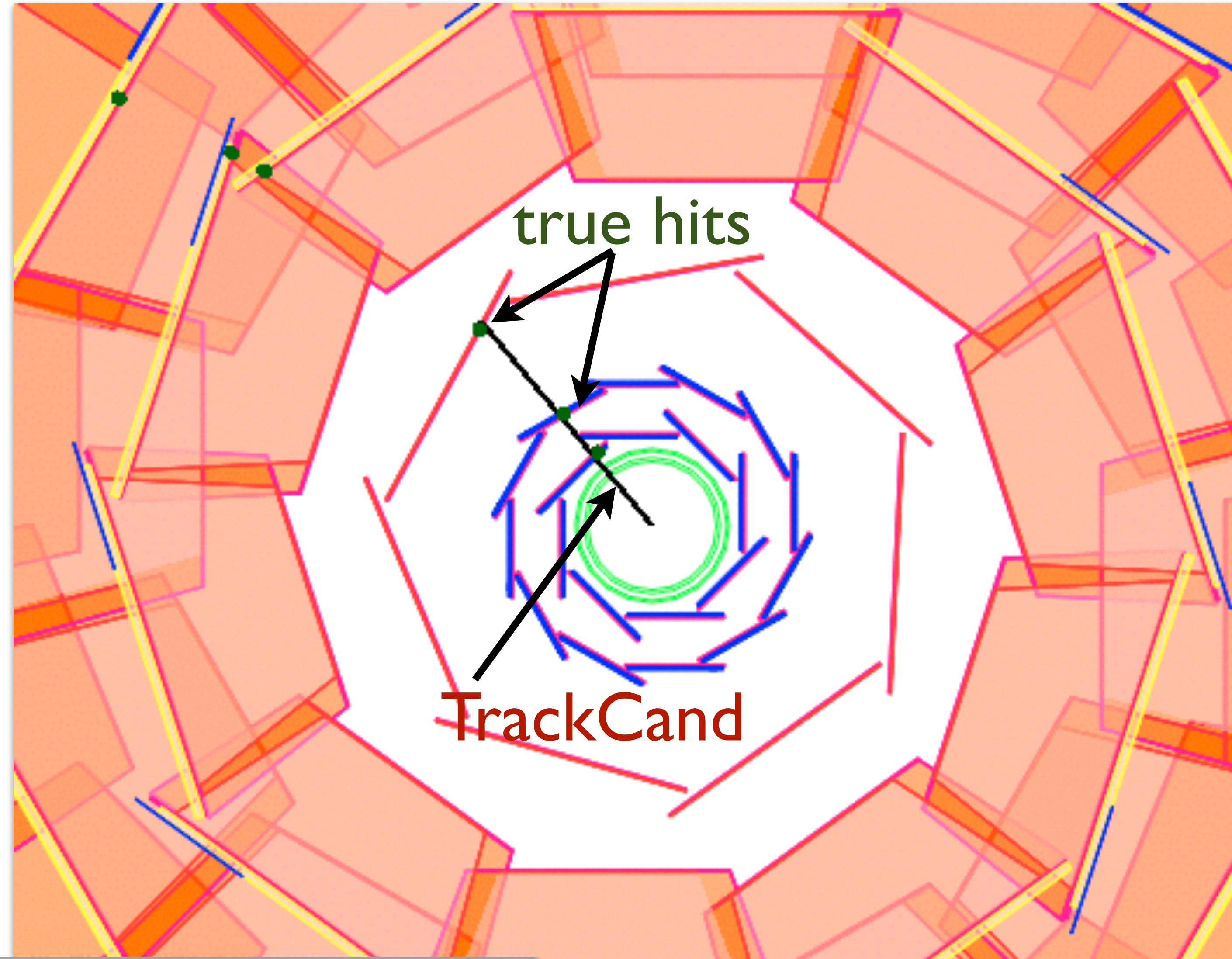


VXDTF



Inefficiency due to *Bad Track Status*



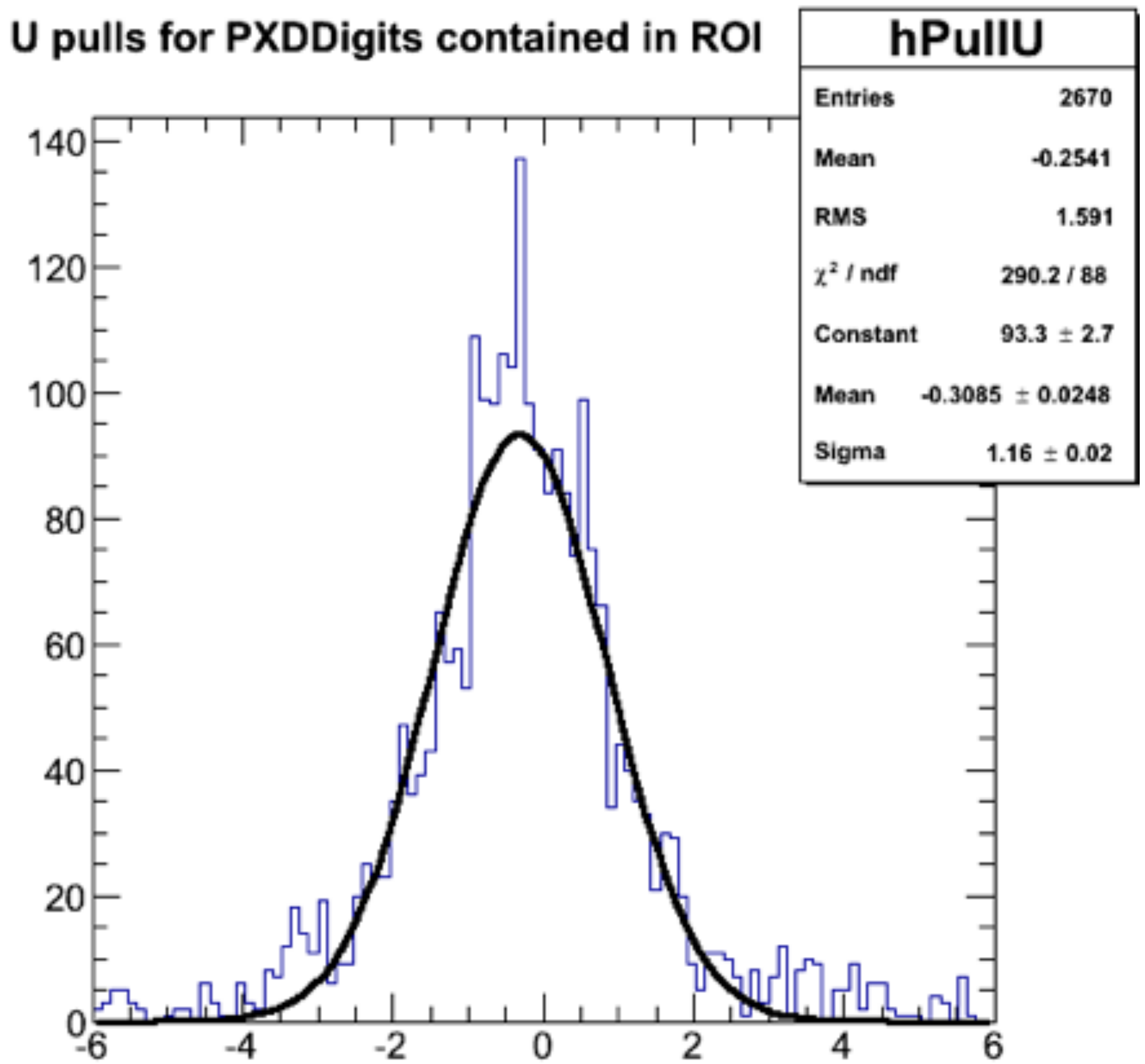


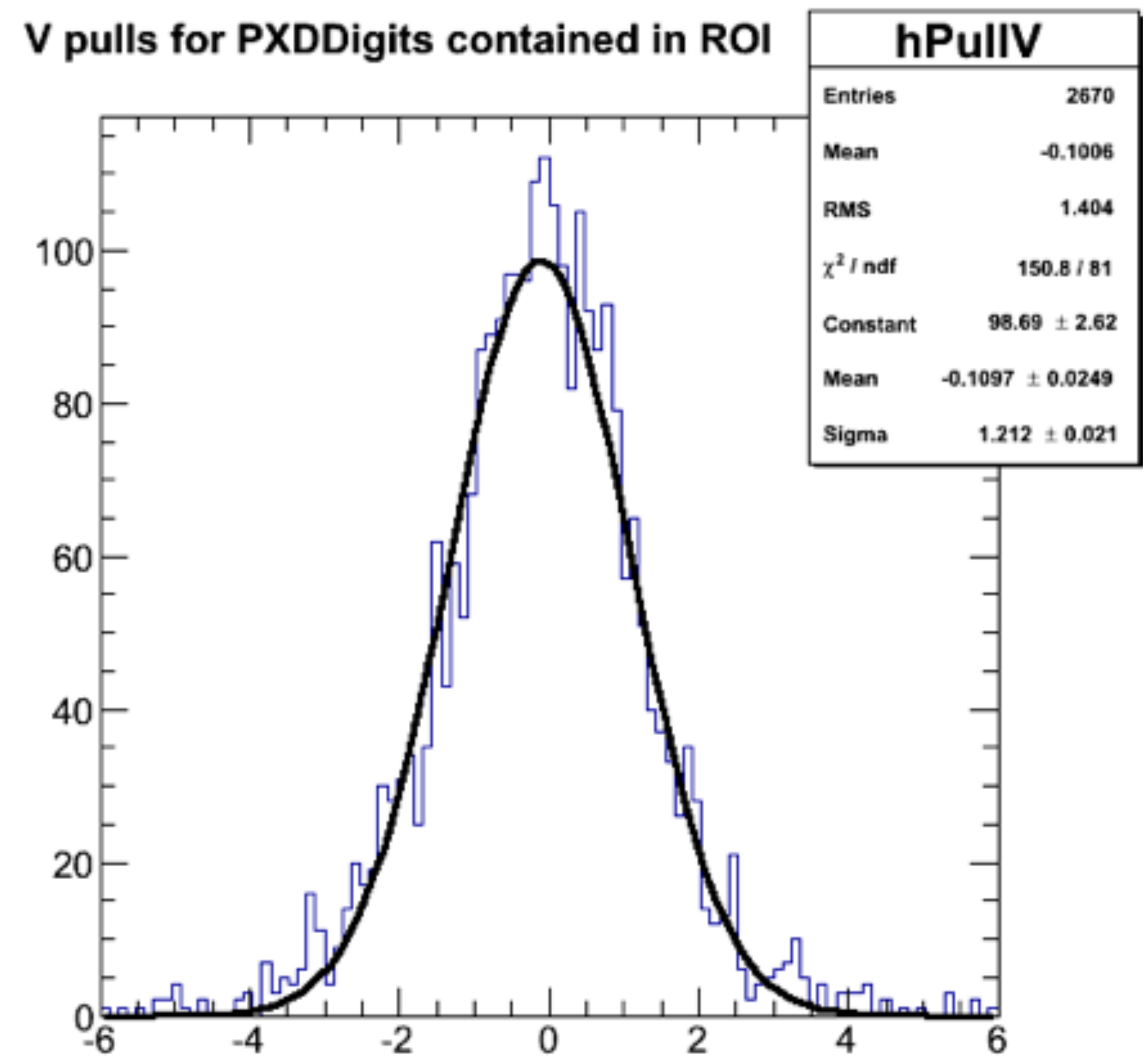

```
GFException thrown with excString:
RKTrackRep::RKutta ==> Do not get closer to plane!
in line: 1230 in file: /home/buildbot/externals/v00-04-01/src/genfit/RKTrackRep/RKTrackRep.cxx
with fa
[WARNIN
```

```
GFException thrown with excString:
RKTrackRep::Extrap ==> maximum number of iterations exceeded
in line: 934 in file: /home/buildbot/externals/v00-04-01/src/genfit/RKTrackRep/RKTrackRep.cxx
with f
[WARNI
```

```
GFException thrown with excString:
RKTrackRep::RKutta ==> momentum too low: 2.56996 MeV
in line: 1134 in file: /home/buildbot/externals/v00-04-01/src/genfit/RKTrackRep/RKTrackRep.cxx
with fatal flag 0
[WARNING] bad track status  { module: PXDDataReduction }
```

The Pulls of the Track Intercept





$$\text{pull} = \frac{\text{intercept} - \text{pixel center}}{\text{stat error on intercept}}$$

- ➡ U (V) Pulls are negatively biased by 30% (11%) of the statistical error
- ➡ the statistical errors are underestimated by ~15-20% (understood, due to our definition of pull)

ROI Modules Status

- ➡ Efficiency needs to be improved, in particular for low p_T tracks
- ➡ Data Reduction Factor estimated ~1%, fully satisfactory
- ➡ Execution Time needs to be reduced
- ➡ Tracking code is in continuos development
  - improvements in track finders
  - improvements in track fitting
- ➡ The ROI module will be re-designed to exploit the powerful tools that are being developed and to match all the requirements

Expected Performance on Beam Test Geometry

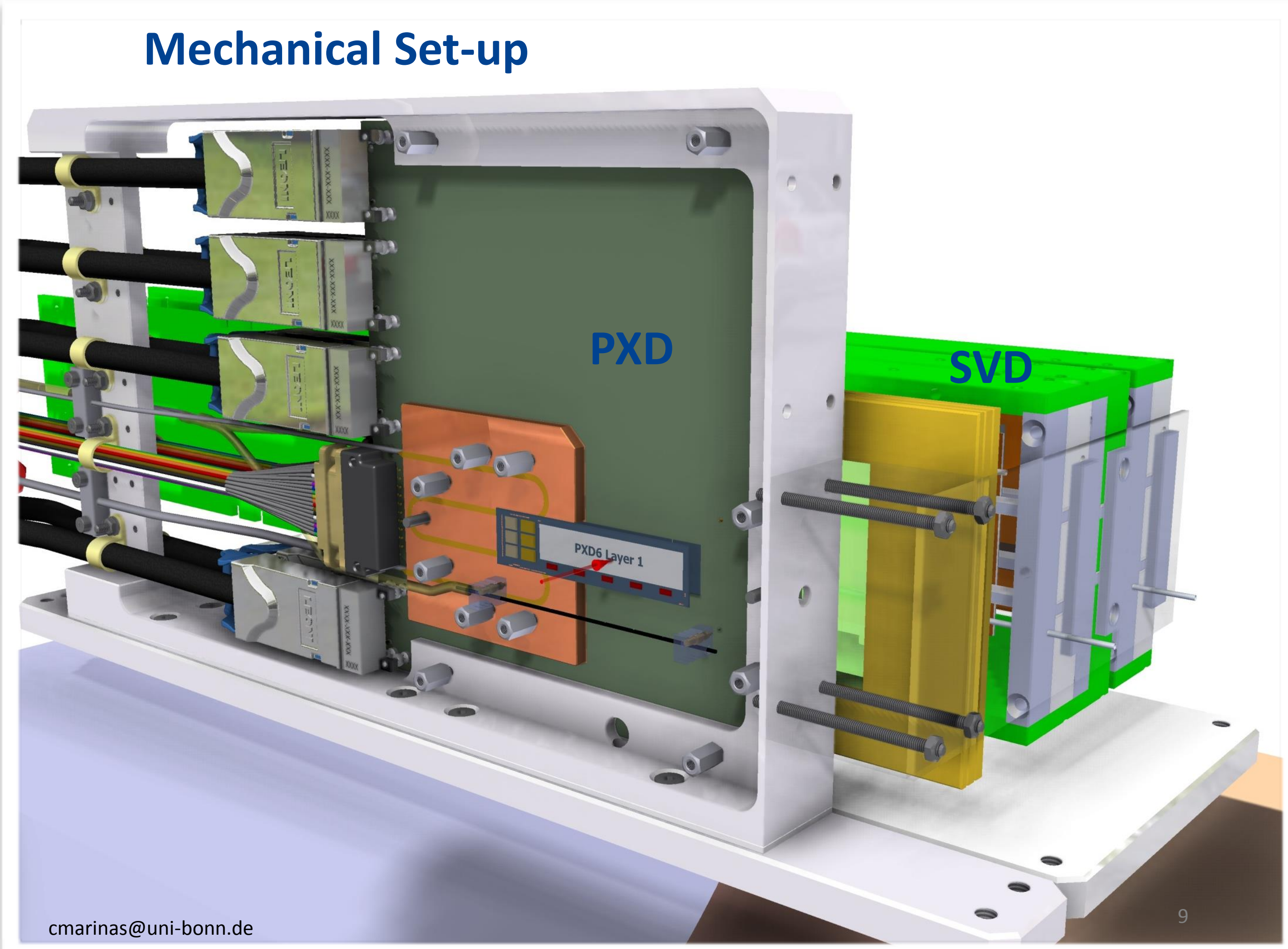
- We simulate 100 events using particleGun (2GeV e^-, no beam divergence) and use the *VXDTF* as *pattern recognition*:
  - ~0.45 tracks/event (0.32 with MCTrackFinder)
  - ~1.9 PXDDigits/track (3.1 with MCTrackFinder)
- Efficiency = (97.5 ± 0.7)% ((99.2±0.4)% with MCTrackFinder)
- Data Reduction Factor = 0.01%
- Execution time = 5 ms/track (in debug mode!)

the code is usable in the test beam environment

VXD test beam

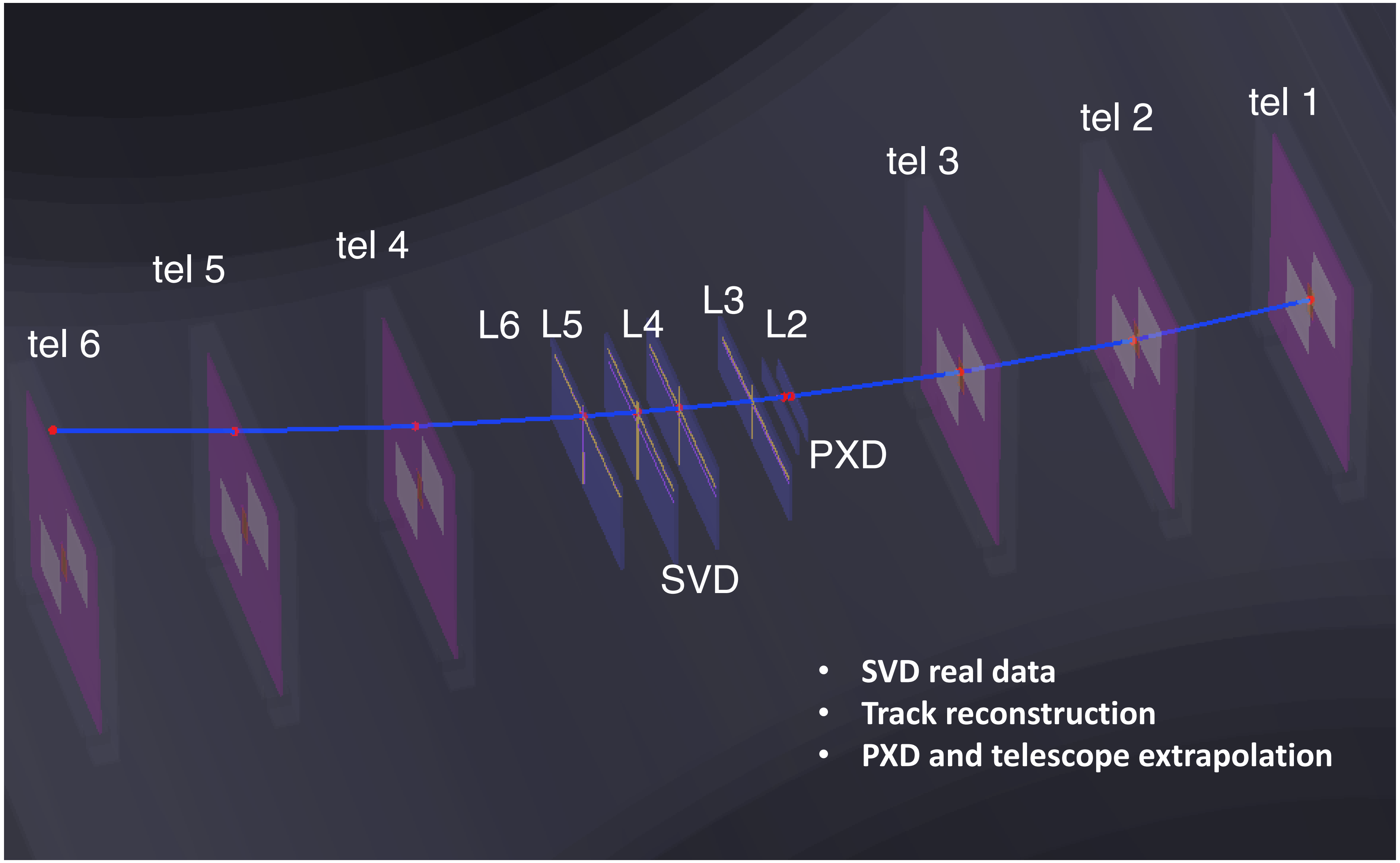
Test-Beam Introduction

- VXD common test beam in January 2014 (4 weeks @ DESY)
- Small sector close to final prototype detectors and ASICs + telescope:
  - 1 PXD half ladder + 4 SVD single module layers
  - EUDET telescope (3+3 layers)
- Complete VXD readout chain: HLT, monitoring, event building, PocketDAQ
- Illumination with (up to) 6 GeV e^- under soleoidal magnetic field
- Alignment, pattern recognition, online tracking, ROIs, …



GOAL = system integration test

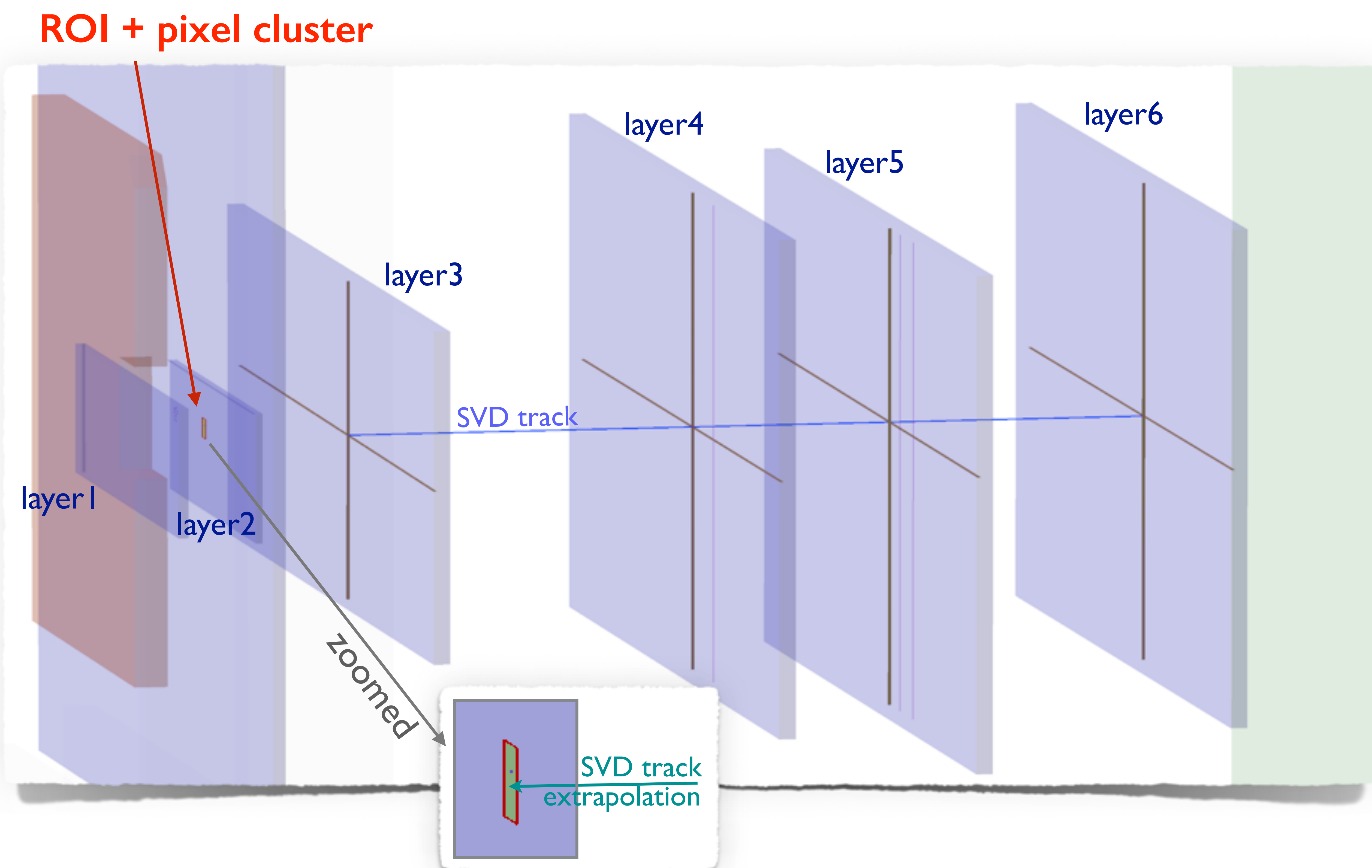
Event Display



The Golden Run for HLT ROIs

- ➡ preliminary results shown here are taken from run 642:
  - no magnetic field
  - 5 mm Al plate in front of PXD (~10% X_0)
  - SVD aligned with the first set of alignment parameters
  - PXD not aligned
    - we manually shifted the intercept position to correct for misalignment, estimating the shift using the previous run (625)
  - ROIs parameters:
    - σ_{syst} = 200 μm in both directions
    - N_σ = 10
    - maximum width = 5 mm in both directions
    - statistical error of extrapolation:
      - $\langle\sigma_{stat}(U)\rangle$ = 27 μm (RMS = 31 μm)
      - $\langle\sigma_{stat}(V)\rangle$ = 16 μm (RMS = 50 μm)

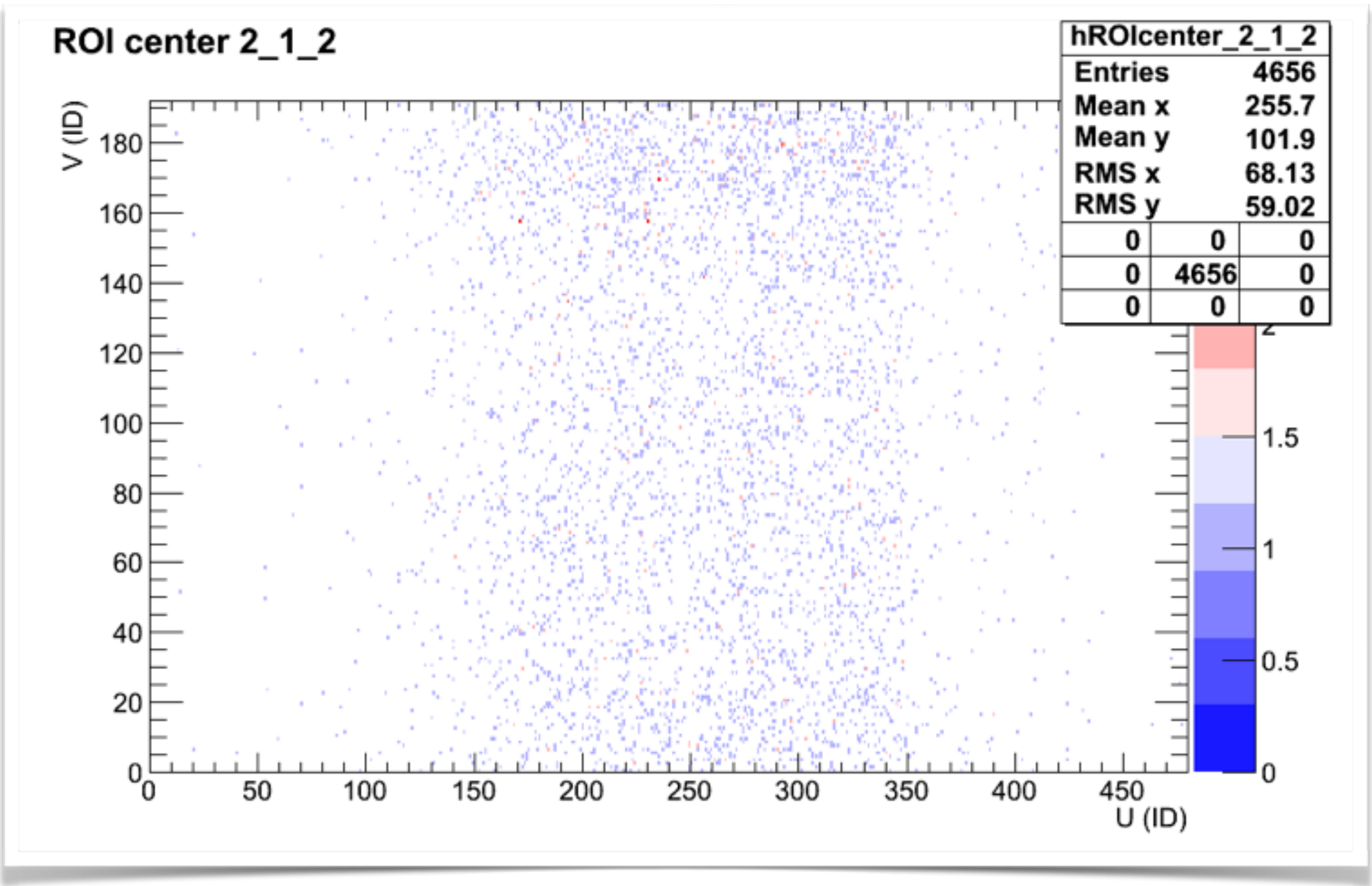
Event 573 of Run 642



PRELIMINARY

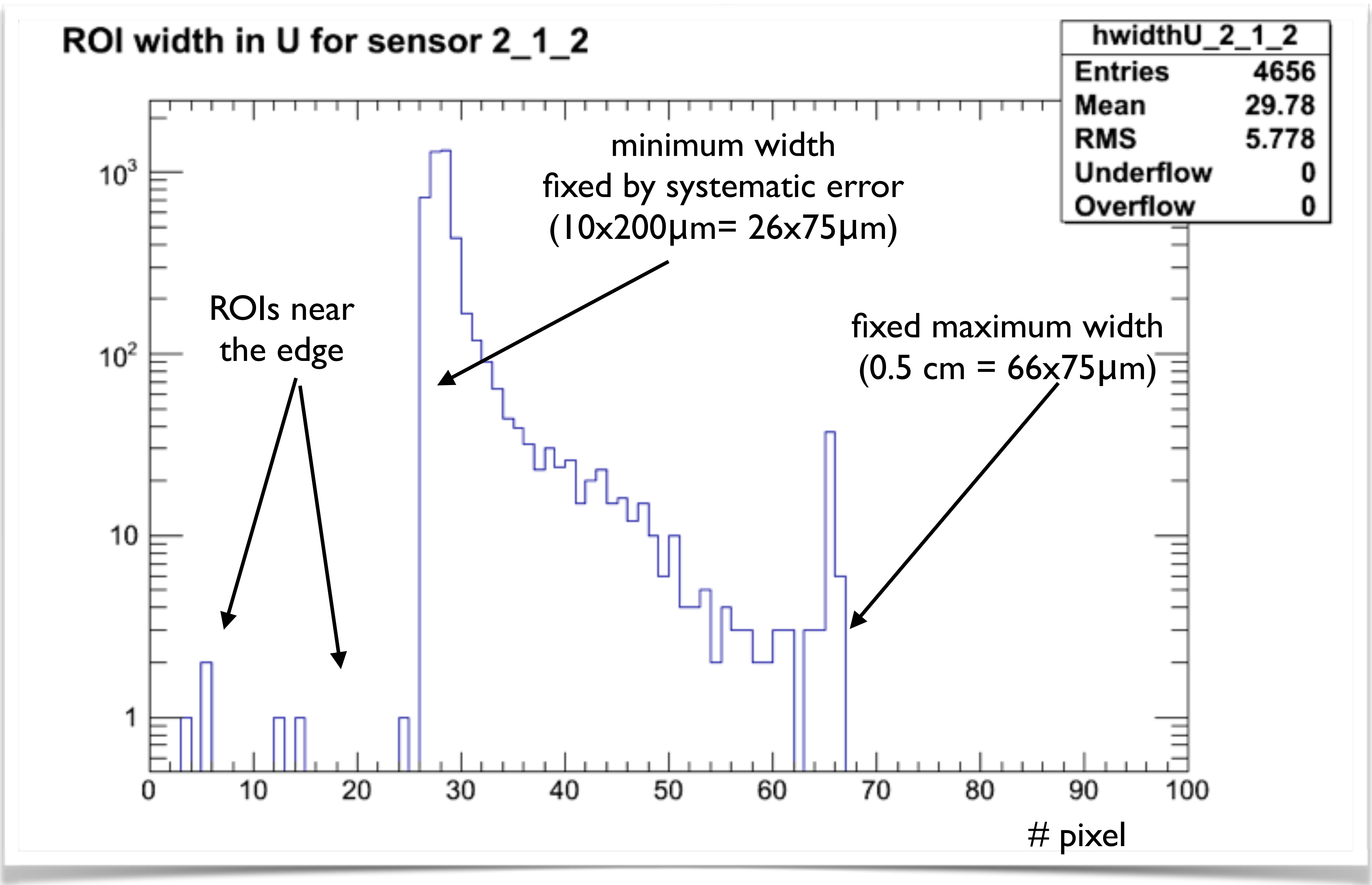
ROIs Position on the PXD sensor

➡ ROIs are centered in U (long side) and spread over the V direction:



ROIs Width along U (long side)

➡ average width along U is 30 pixels = 2.25 mm

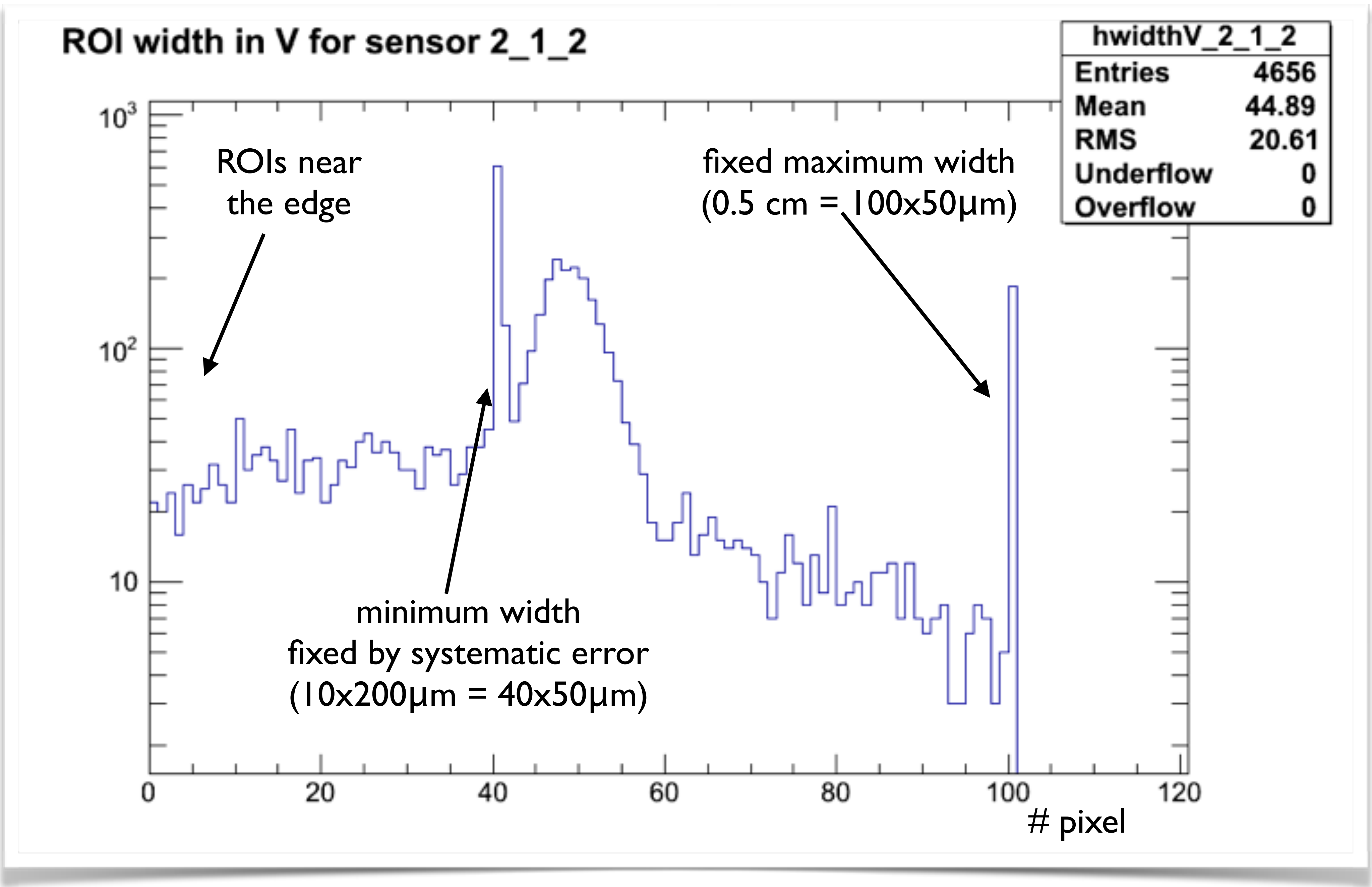


PRELIMINARY

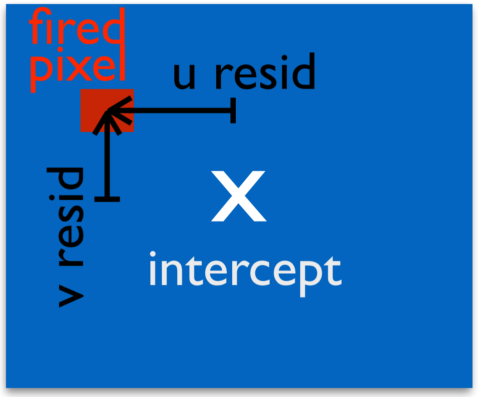
run642

ROIs Width along V (short side)

➡ average width along V is 45 pixels = 2.25 mm

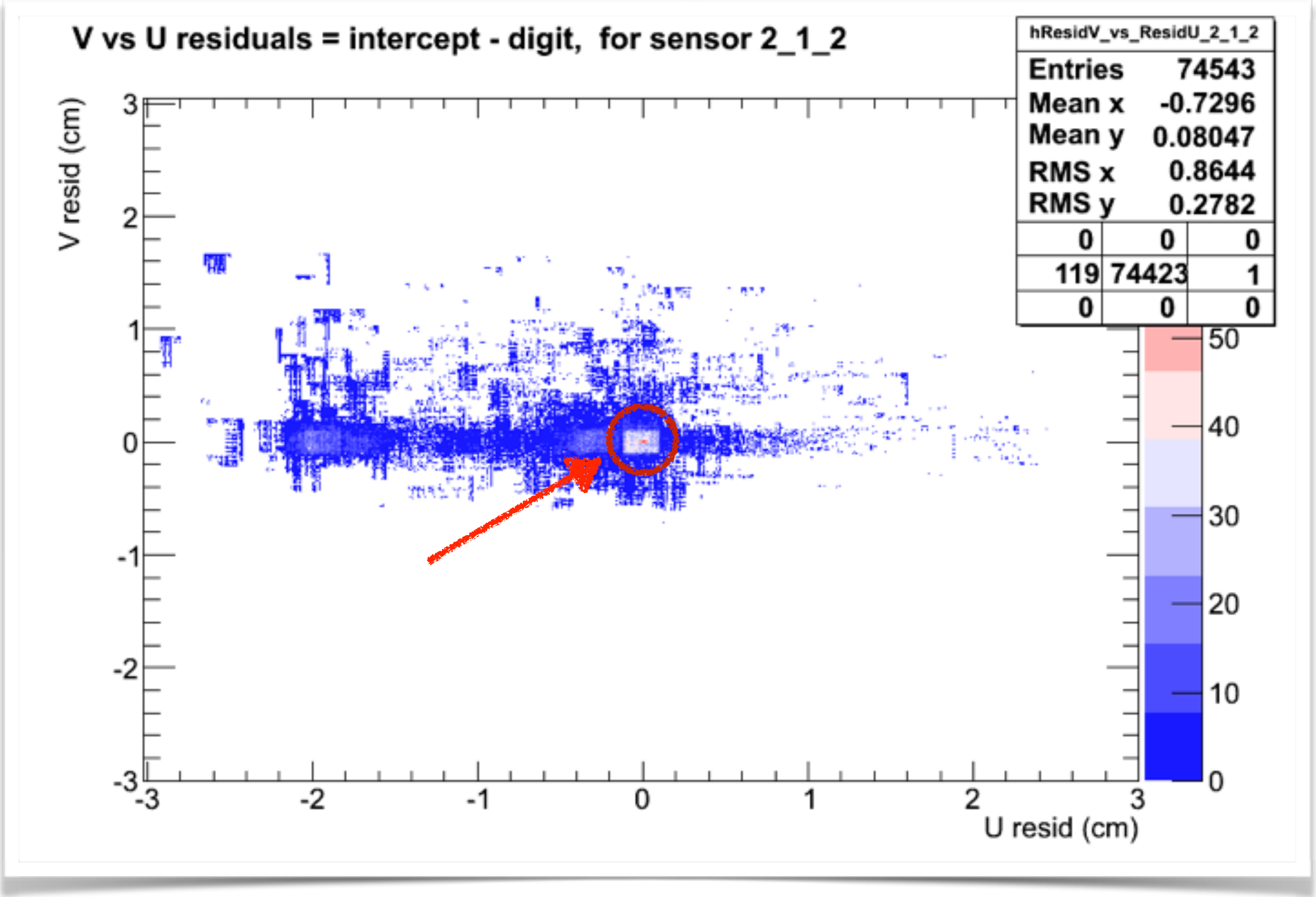


2D Residuals

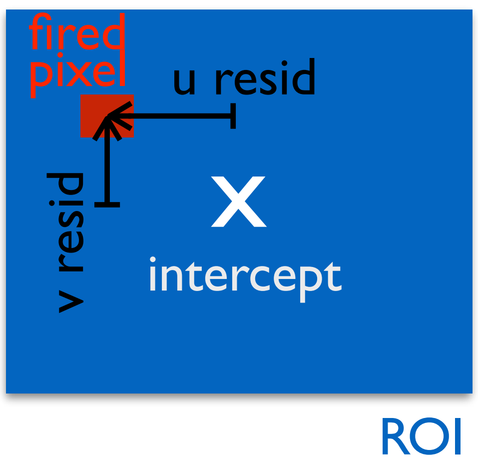


ROI

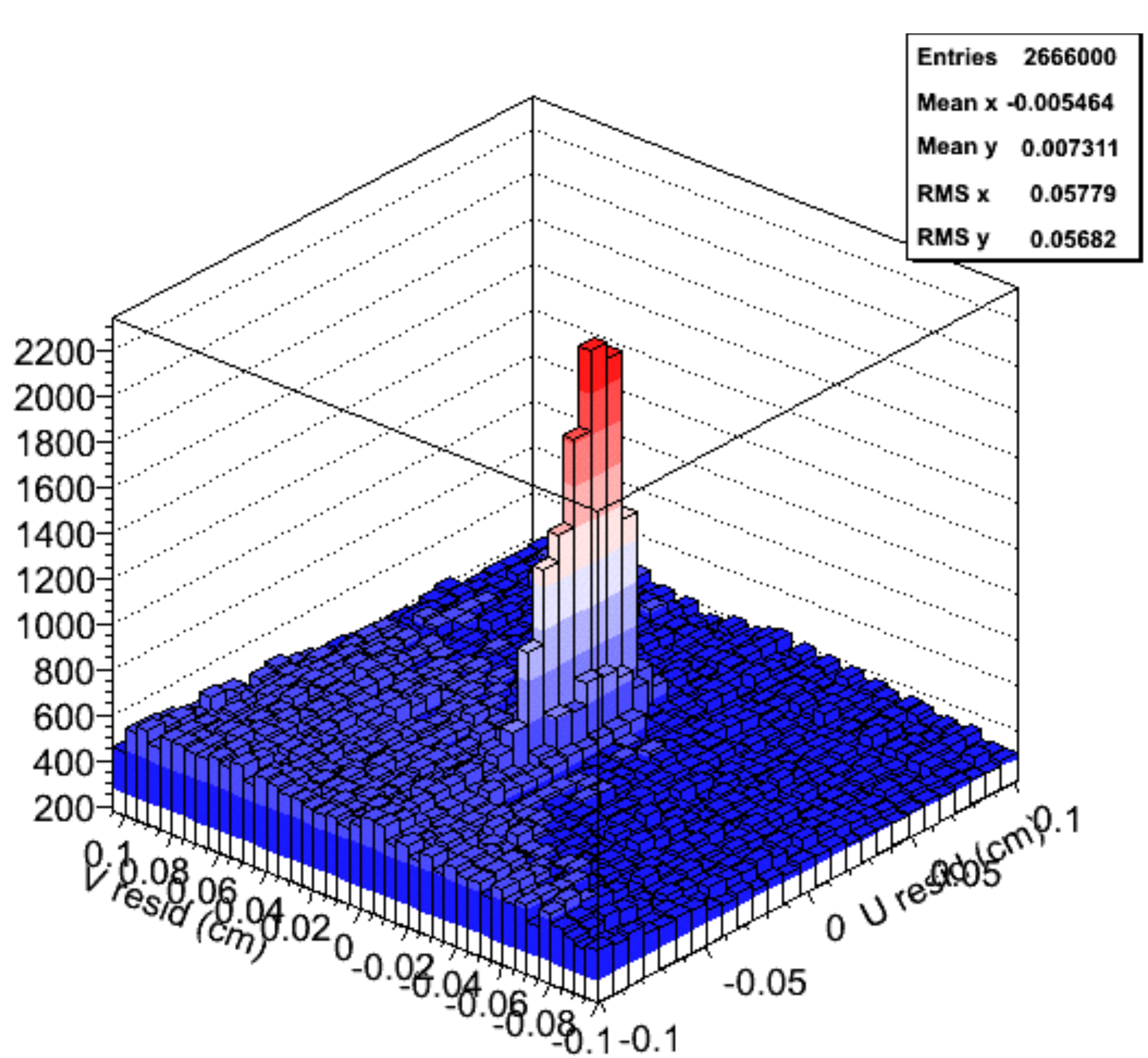
➡ shadow of HTL ROIs + DATCON ROIs spread over the sensor



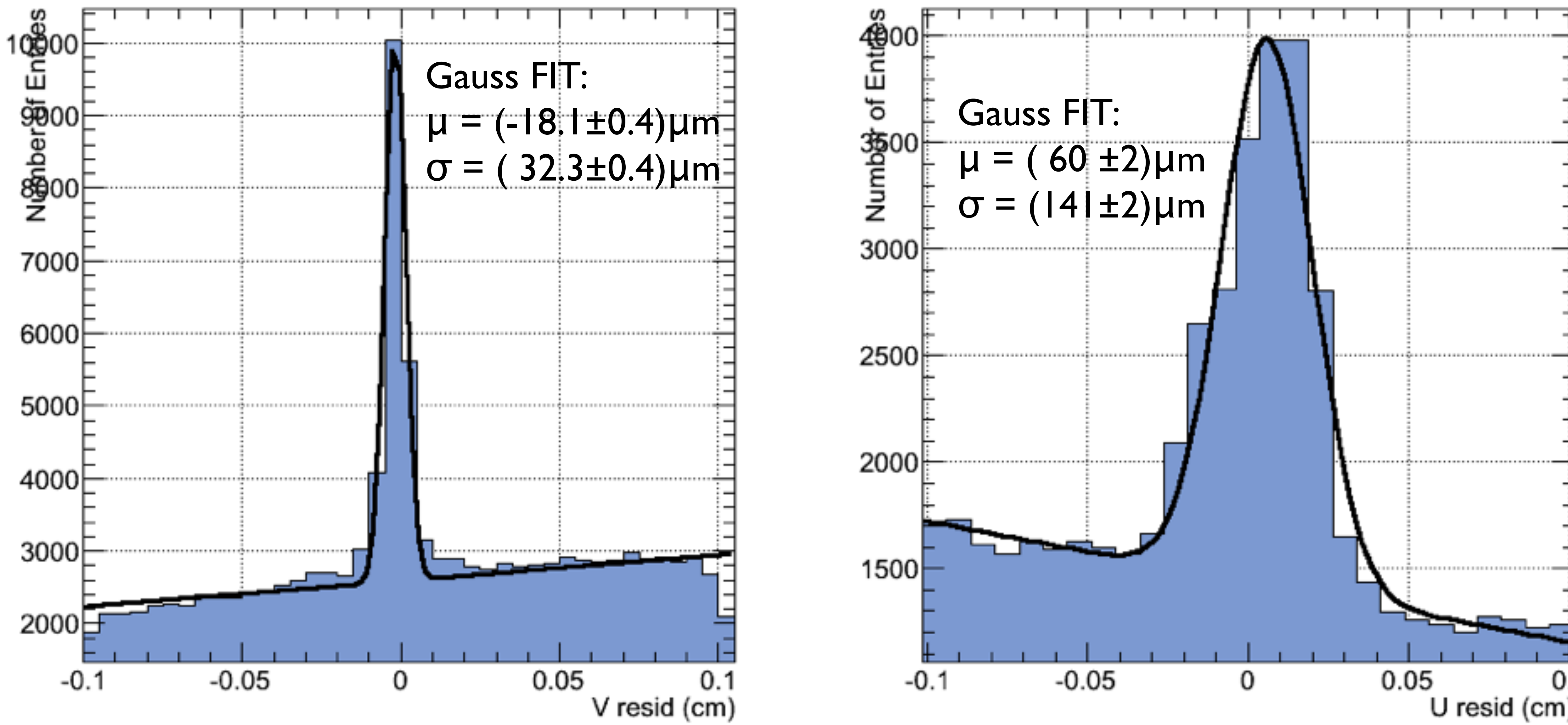
2D Residual Peak Zoomed in



➡ a clear peak of signal pixels is shown:



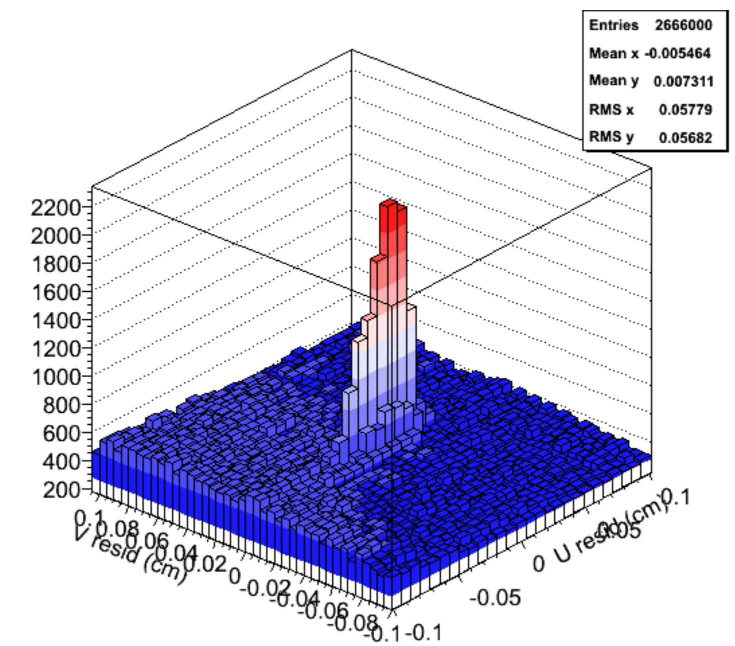
Projected Residuals



- PXD ladder was *manually* aligned, minimal residual misalignment not affecting ROIs finding
- Extrapolation resolution ~30μm on short side (V) and ~140μm on long side (U), difference due to SVD strip pitch difference

Conclusions

- ROI Finding is of crucial importance in *Belle II*
- At present, our module is based on an pattern recognition module and a track-fit function whose performance influence the ROI finding efficiency
- Improvements in execution time and efficiency are needed and will come in the future, exploiting the powerful features being developed by the tracking group
- The entire ROI chain has been successfully tested during test beam:
  - the most important result is already achieved!



Thank You!