Development of the digitization framework

H. Cao, A. Paramonov, and others

09/08/2026

Development of the digitization framework for FCC-ee experiments

- FCC-ee experiments are aiming to carry out a number of measurements and searches with uncertainties as low as 1e-5 … 1e-6
- The Trigger and Data Acquisition system should comply the requirements of the physics program.
  - The data losses need to be negligible. For example, the on-detector buffer overflow needs to negligible and accounted for.
  - Triggering may be avoided or it needs to be done with extremely low systematic uncertainties.
- Therefore, the FCC community needs tools to evaluate the data rates, experimental signatures, event sizes, and
  - The experimental signatures are instrumental to explore on-detector (on-edge) AI/ML data processing.
- Outputs of the detectors need to be digitized and stored with realistic granularity.
  - As an example, outputs of the vertex detector are stores as space points, not individual pixels.

Vertex Detector Digitization

- We started the development with the vertex detector given its high output bandwidth and challenging cable routing.
- Our code is Based on the DetailedDigitizer package from Lyon group (thanks to Jessy Daniel, Gaelle Boudoul, and others): https://indico.cern.ch/event/1516157/contributions/6445378/attachments/3064438/5420120/GB_Brookhaven_2025.pdf
- The original code creates space points after the digitization but the individual pixels are not stored.
  - Lorenz forces and diffusion are taken into account
  - Unfortunately, signal time is not evaluated at the moment.
  - A space-point is calculated as ionization-weighted average of individual pixels in a cluster.
- The space points are used for track reconstruction and resolution calculations.

Vertex Detector Digitization

- We have updated the code to store the individual pixels in a separate branch.
- We store 3D coordinates (x,y,z), local coordinates (row, column), energy loss, time, and CellID.
- We need to add associations to the generated particles

```
root [1] events->Scan("VXDTrackerPixels.position.x:VXDTrackerPixels.position.y:VXDTrackerPixels.position.z:VXDTrackerPixels.du:VXDTrackerPixels.dv")
************************************************************************************
*
*    Row    * Instance * VXDTracke * VXDTracke * VXDTracke * VXDTracke * VXDTracke *
************************************************************************************
*
*       0 *        0 * -28.90408 * 51.141826 * 44.325000 *      246 *     1773 *
*       0 *        1 * -28.92717 * 51.132259 * 44.325000 *      247 *     1773 *
*       0 *        2 * -4.285361 * -12.57746 * -1.700000 *      -11 *      -68 *
*       0 *        3 * -4.262264 * -12.58703 * -1.700000 *      -10 *      -68 *
*       0 *        4 * -4.262264 * -12.58703 * -1.675000 *      -10 *      -67 *
*       0 *        5 * -4.239167 * -12.59659 * -1.675000 *       -9 *      -67 *
*       0 *        6 * -4.239167 * -12.59659 * -1.650000 *       -9 *      -66 *
*       0 *        7 * -4.216070 * -12.60616 * -1.650000 *       -8 *      -66 *
```

Code base

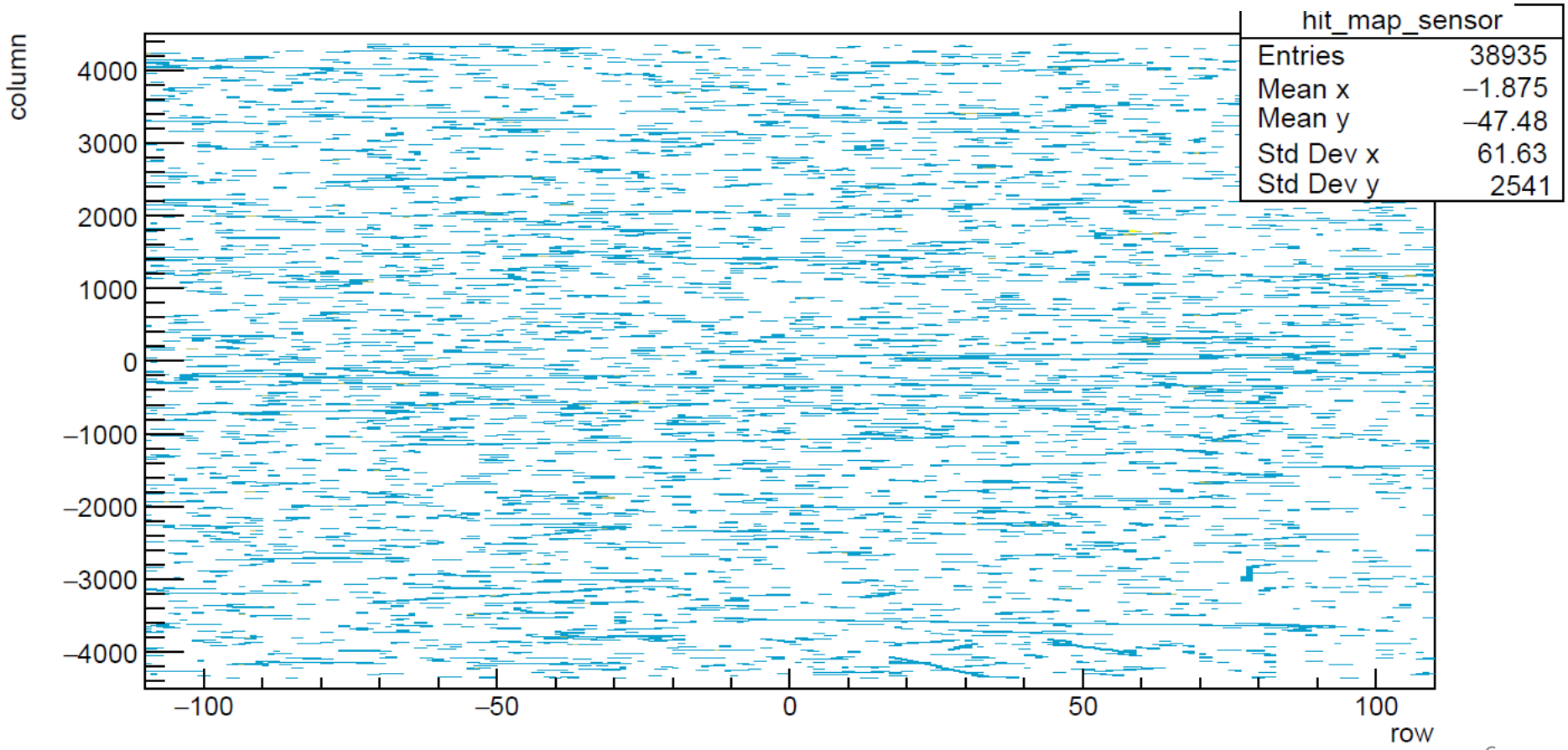
- The code is in GIT:
  - https://gitlab.cern.ch/caoh/k4rectracker
  - https://gitlab.cern.ch/caoh/cld_config
- The existing python (in cld_config) scripts are for CLD.
- We will prepare the scripts for the other detectors.
- The code runs with k4hep. For example:

```
k4run CLDReconstruction.py --truthTracking --trackingOnly --inputFiles
../../Step1_edm4hep*.root \
                --outputBasename Step2 \
                -n -1
```

- The instructions are in the README file in k4rectracker
- The code should eventually be included into the main reconstruction branch.
- It would also be useful to use individual pixels for track fitting

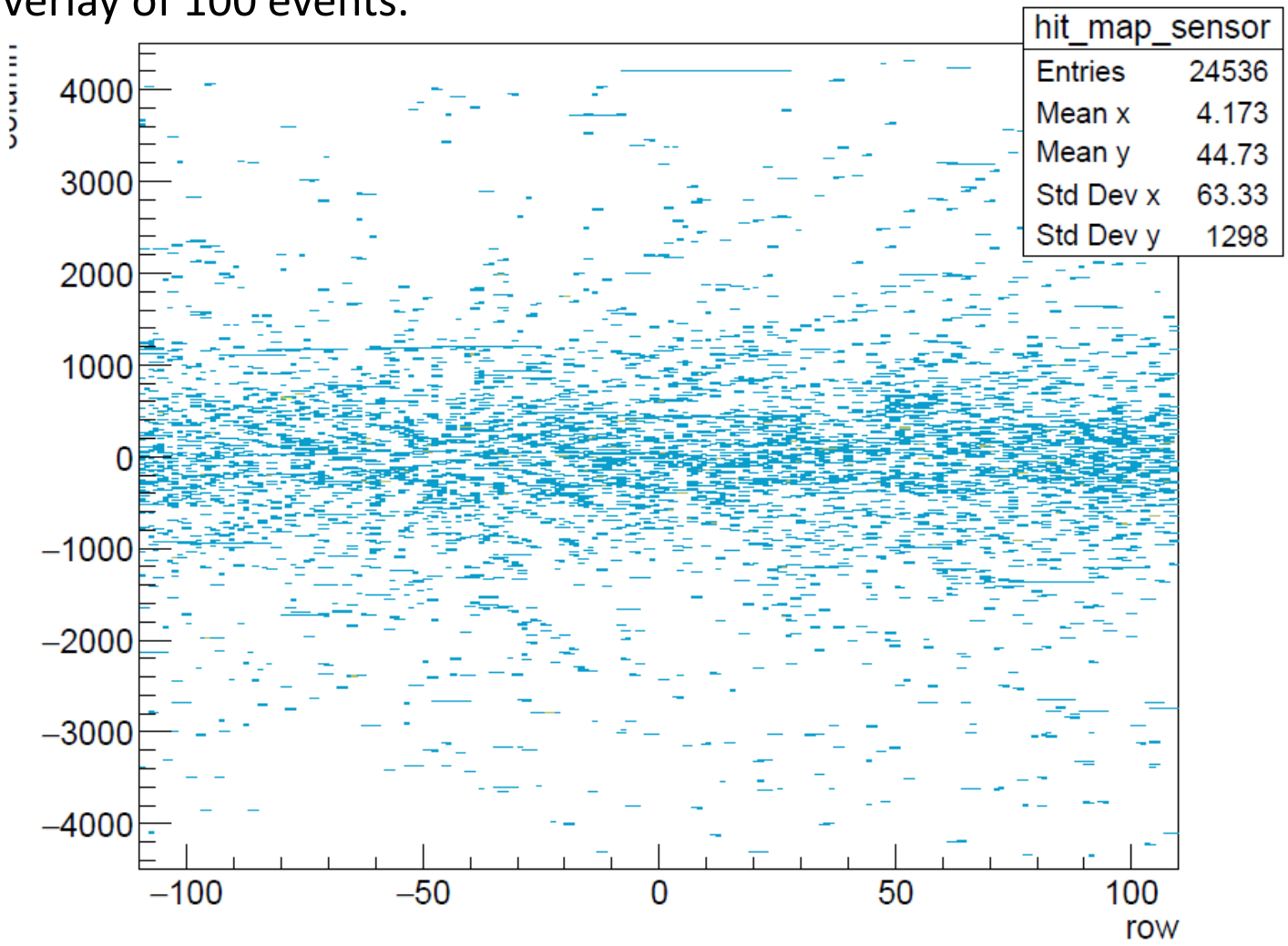
Validation results

Beam backgrounds. Thanks to the MIT group for sharing the BB simulations.
The sensor is 218 mm x 5.5 mm. 25 x 25 um^2 pixels.
The image is overlay of 100 events.



Validation results

ZH. Z decays to muons and H decays inclusively
Overlay of 100 events.



What's next?

- We have had a look at the barrel part of the vertex detector.
- It would be useful to validate the endcaps.
- Then we can move to development of the flexible digitization code for the outer layers of the tracker (drift chamber, strip sensors, and so on.)
- We will collaborate with the experts closely