Digitization (and MPGD Simulation) in EICrecon

Yann Bedfer

CEA/DPhN Saclay

17 February 2025

Present Workflow

- `ddsim` → collections of **simulated hits** = `edm4hep:SimTrackerHit`

  $= (\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}, \vec{\boldsymbol{P}}, \boldsymbol{E_{dep}}, \boldsymbol{t},$ **cellID, pathLength**. . .)

  Example of collections: `MPGDBarrelHits`, `OuterMPGDBarrelHits`, `BackwardMPGDEndcapHits` . . .

  → Parallel (*same # of entries*) collections of index into `edm4hep:MCParticle`

- `eicrecon`'s **digitization** → collections of **raw hits** = `edm4eic::RawTrackerHit`

  $= (\boldsymbol{iQ}, \boldsymbol{iT},$ **cellID**)

  Example of collections: `MPGDBarrelRawHits`, `OuterMPGDBarrelRawHits` . . .

  → Parallel collections for X-reference SimHit ↔ RawHit

  - `SiliconTrackerDigi` (*all but CyMBaL and Outer, as of now*):

    1 SimHit → **1** RawHit , **cellID** = **cellID**

  - `MPGDTrackerDigi` (*CyMBaL, Outer, as of now*):

    1 SimHit → **2** RawHits, **cellID's** ≠ **cellID**

  `MPGDTrackerDigi` ***not only digitization***, *but* ***also simulation.***

  Eventually: ***1*** *SimHit* → ***2*** ***clusters*** *of RawHits.*

  # of RawHits per SimHit still impacted by ***Merging****, see infra.*

- `eicrecon`'s **reconstruction** → collections of **reconstructed hits** `edm4eic::TrackerHit`

  $=$ (**position,positionError**, $\boldsymbol{E_{dep}}, \boldsymbol{\Delta E_{dep}}, \boldsymbol{t}, \boldsymbol{\Delta t},$ **cellID**)

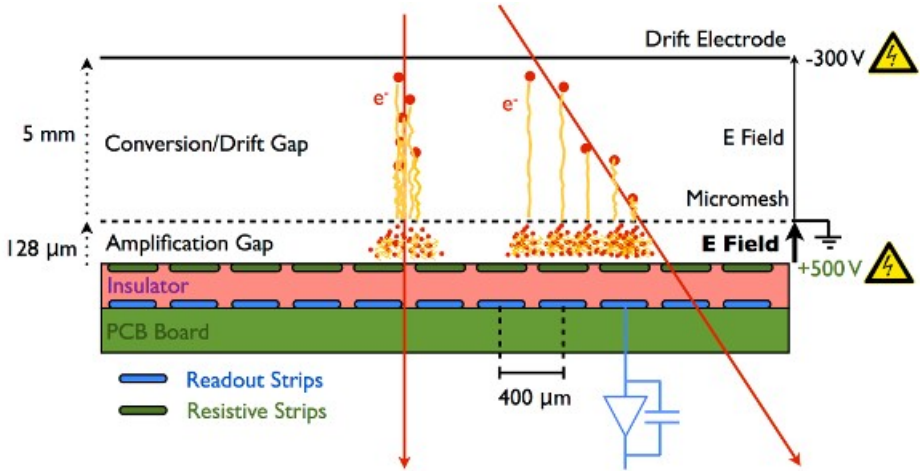
Digitization

- `SiliconTrackerDigi` ($SiTD$) and **part of** `MPGDTrackerDigi` (MTD).
- Digitization = Simulate FE electronics.
  - Digitization proper: $float \rightarrow int$.
  - **Discrimination**: cut on $\boldsymbol{E_{dep}}$ (`config::threshold`).
  - **Smearing Time** (`config::timeResolution`), but **no Energy Smearing**.
  - **Merging** SimHits falling on same **cellID** ($SiTD$) or strip **cellID** (MTD).

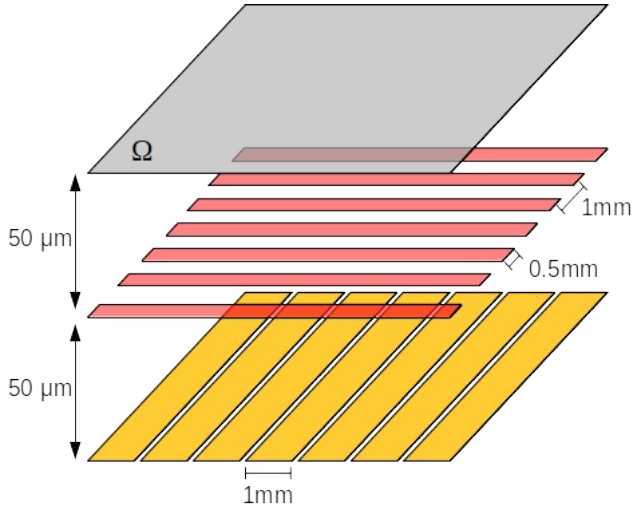
- Issue: Threshold **discrimination** applied **too early**, see EICrecon #1722.
  - **Discrimination first**, then **Merging**.
    $\Rightarrow$ Prevents merging of small energy from helping pass threshold.
  - **Merging first**, then **Discrimination**. **Time assigned** to Merged Hit**?**
    Earliest? $\Rightarrow$ Sensitivity to whatever accidental background.
    Largest $\boldsymbol{E_{dep}}$? Not fully satisfactory.
  - **Simon Gardner** (Glasgow U.) claims he has a step-by-step idea,
    that he will present to the **Track-Recon WC** at a forthcoming meeting (**Feb.20**).
  - My idea: shouldn't we have a `config::deadTime`?
    Does this imply inheritance of deadTime across time frames?
  - To what level of detail do we want to go?
    Ultimately, could be the amplitude sampling level,
    w/ an algorithm finding peaks in the superposition of signal shapes.

MPGDTrackerDigi

- Current `MPGDTrackerDigi`
  - **Re-segmentation**: Overwriting what's done in simulation. → Strips instead of Pixels.
  - **Merging**: Based on newly segmented strip **cellID**.
  - **For the rest**: **exact same** as `SiliconTrackerDigi`.
- Planned `MPGDTrackerDigi`



Class12 Micromegas: 1D-strip Readout



Charge Sharing ⇒ 2D-strip Readout

- **One I**onization × **A**mplification $\xrightarrow[\textbf{Sharing}]{\textbf{Charge}}$ **Two proto** Hits, along strips $\boldsymbol{u} = \boldsymbol{p}, \boldsymbol{n}$

  Amplitude/Time **S**mearing $= \boldsymbol{S}(\textbf{I}\times\textbf{A}) + \boldsymbol{S}(\textbf{ChSh}|\boldsymbol{u})) + \boldsymbol{S}(\textbf{F}\text{ront}\textbf{E}\text{nd})$.

  Do we need the **F**ront**E**nd term? Or negligible?
- **Charge Spreading** (*can only be a posteriori*): **proto** Hit → **cluster** of Hits.

Separate Simulation from Digitization?

- Put forward by **Dmitry Kalinkin**: so as to have a **unique digitization method**.
  I.e.: Factorize `MPGDTrackerSimulation` × `SiliconTrackerDigi`.
  `MPGDTrackerSimulation` **input collection** of SimHits, **output new collection** of SimHits.
- Problem: Any **specificity of MPGDs** preventing this?
  - Is FE electronics same for the two coordinates (*including parameters*)?
    If not, **output two collections**, one per strip coordinate, w/ **each its own parameters** `config`?
  - Amplitude smearing at **F**ront**E**nd? (*not in current* `SiliconTrackerDigi`)
  - **Merging** and **Charge Sharing/Spreading**:
    - **One track** at a finite angle strikes **two adjacent pixels** but **one strip** (*if angle along the strip*):
      → Two SimHits but **one I**onization×**A**mplification.
      ⇒ The two SimHits have to be **Merged before Charge Sharing/Spreading**.
    - **Two tracks** falling on a **same** strip at some distance:
      → **Two I**onization×**A**mplification's. ⇒ **Charge Sharing/Spreading first**.

    ⇒ Need **two Merging** steps.
    But could be one in `MPGDTrackerSimulation` and one in `SiliconTrackerDigi`.
  - **Conclusion: No showstopper**. But some amount of work. . .
    Do we still insist for **F**ront**E**nd Amplitude Smearing in `SiliconTrackerDigi`?

MPGD Simulation (*as opposed to digitization*)

- Already agreed(*t.b.c.*): based on **Parametrization** of **BeamTests**. To what level of detail?
  - Presently (*in progress*): two-hit clusters, w/ **Gaussian** smearing (150 μm), **truncated** so as not to exceed the closest neighbour or the edge.
  - In the **longer term?** Keeping in mind we have to **amass BeamTests accordingly**.
- **One-dimensional** parametrizations: random draw cluster-size, then draw position based on resolution. **Oversimplistic**.
- **Minimal** program:

  Two-dimensional: resolution as a f(cluster-size).

  Amplitude and Time correlations also according to BeamTests.

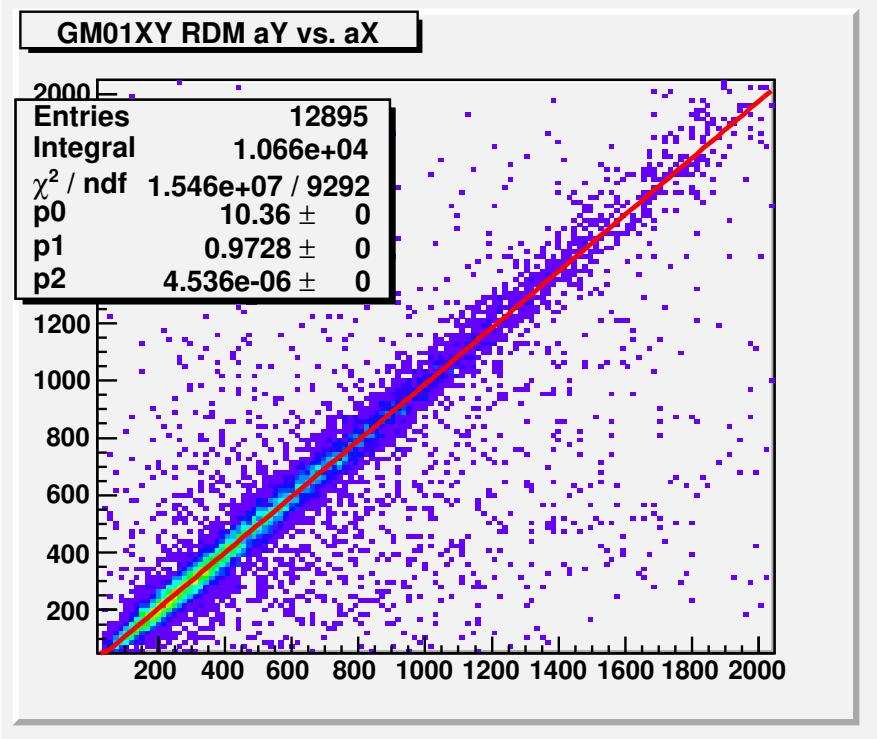
  (*Simulation of the random factor in* **Charge Sharing**).
- **Angle** (*across strips*):
  - `ddsim` does distribute $\boldsymbol{E_{dep}}$ as a **f(pathLength)**. Overwrite but **recycle** part of the info. How?

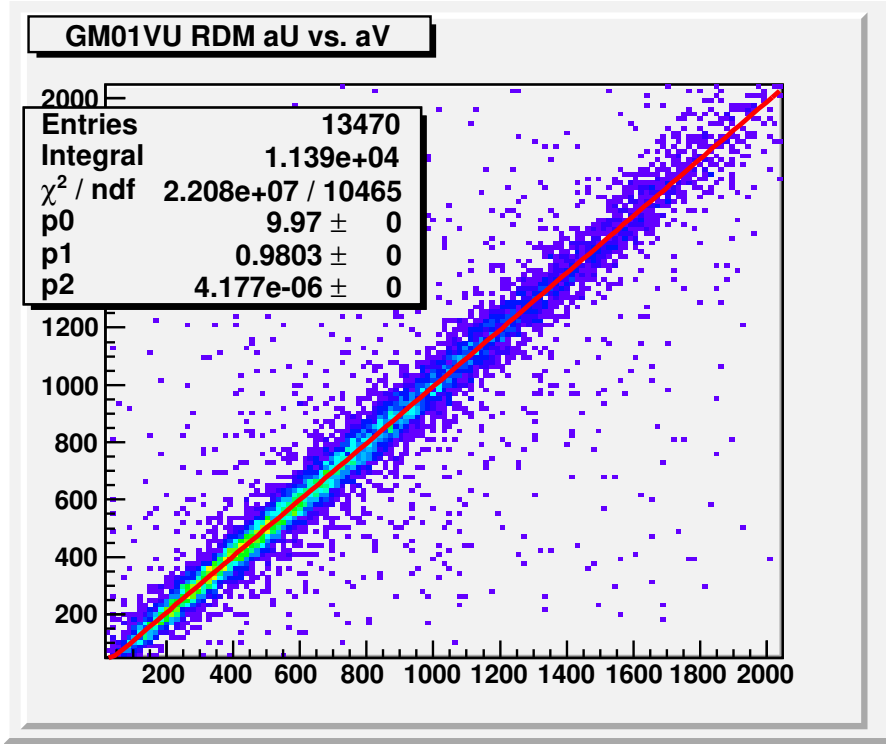
    Instead, brute **force three-dimensional**: resolution as a f(cluster-size × angle)?
  - Timing: simulate **arrival time as a f(depth)** (*depth in conversion gap*): micro-TPC.
- **Timing** of hits from cluster **wings**.
- **Edge effect**: on the edge, truncated cluster.
- Impact of **Spacers**.

Backup

Amplitude Correlation

- Can be **exploited in Tracking**.
- As an **example**: amplitude correlation obtained in **COMPASS GEMs** (*after some calibration*):





Class definitions (in /opt/local/include)

- edm4hep:SimTrackerHit = (X, Y, Z, $\vec{P}$, E_{dep}, t, **cellID**, **pathLength**. . .)

  std::uint64_t **cellID**;
  float **EDep**; // energy deposited [GeV].
  float **time**; // time in lab frame [ns].
  float **pathLength**; // path length in sensitive material
  std::int32_t **quality**; // quality bit flag.
  ::edm4hep::Vector3d **position**; // position [mm].
  ::edm4hep::Vector3f **momentum**; // 3-momentum [GeV]

- edm4eic::RawTrackerHit = (iQ, iT, **cellID**)

  std::uint64_t **cellID**;
  std::int32_t **charge**; // ADC value
  std::int32_t **timeStamp**; // TDC value.

- edm4eic::TrackerHit = (**position**,**positionError**, E_{dep}, ΔE_{dep}, t, Δt, **cellID**)

  std::uint64_t **cellID**;
  ::edm4hep::Vector3f **position**; // Hit (cell) position [mm]
  ::edm4eic::CovDiag3f **positionError**; // Covariance Matrix
  float **time**; // Hit time [ns]
  float **timeError**; // Error on the time
  float **edep**; // Energy deposit in this hit [GeV]
  float **edepError**; // Error on the energy deposit [GeV]

(*Note:* ***cellID*** *comment omitted, because misleading imho.*)