FACULTY OF PHYSICS
UNIVERSITY OF WARSAW

ILD-TPC with pixel readout - simulation and reconstruction status

Jan Klamka
University of Warsaw
jan.klamka@fuw.edu.pl

Motivation

First due to ongoing **long-lived particle** search analysis

- **Standard** vs. **all-silicon ILD** design comparison was part of the analysis

  → would be interesting to compare with another design

- **Low-p_T** region under study in the analysis - "missing hits" problem in the pad-based readout had a significant impact

  → improvement expected for the **pixel readout**

...turns out not so easy

Technical status

What do we have from Kees Ligtenberg?

- detector models in lcgeo – **55 µm** or **990 µm** sensitive volume thickness
- modifications/patches in: DD4hep, KalTest, DDKalTest, MarlinTrk, MarlinTrkProcessors, MarlinUtil, MarlinDD4hep, MarlinReco and also ILDPerformance (for analysis/tests)
- **dedicated tracking** nested in the Clupatra processor

Technical status

What do we have from Kees Ligtenberg?

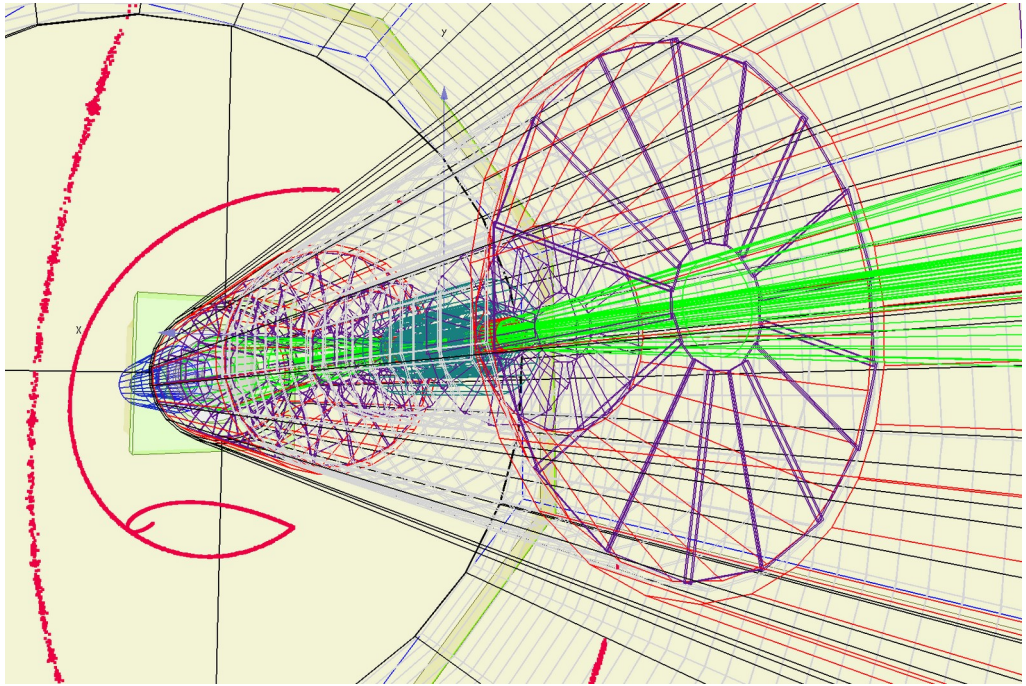
- detector models in lcgeo – **55 µm** or **990 µm** sensitive volume thickness
- modifications/patches in: DD4hep, KalTest, DDKalTest, MarlinTrk, MarlinTrkProcessors, MarlinUtil, MarlinDD4hep, MarlinReco and also ILDPerformance (for analysis/tests)
- **dedicated tracking** nested in the Clupatra processor

But...

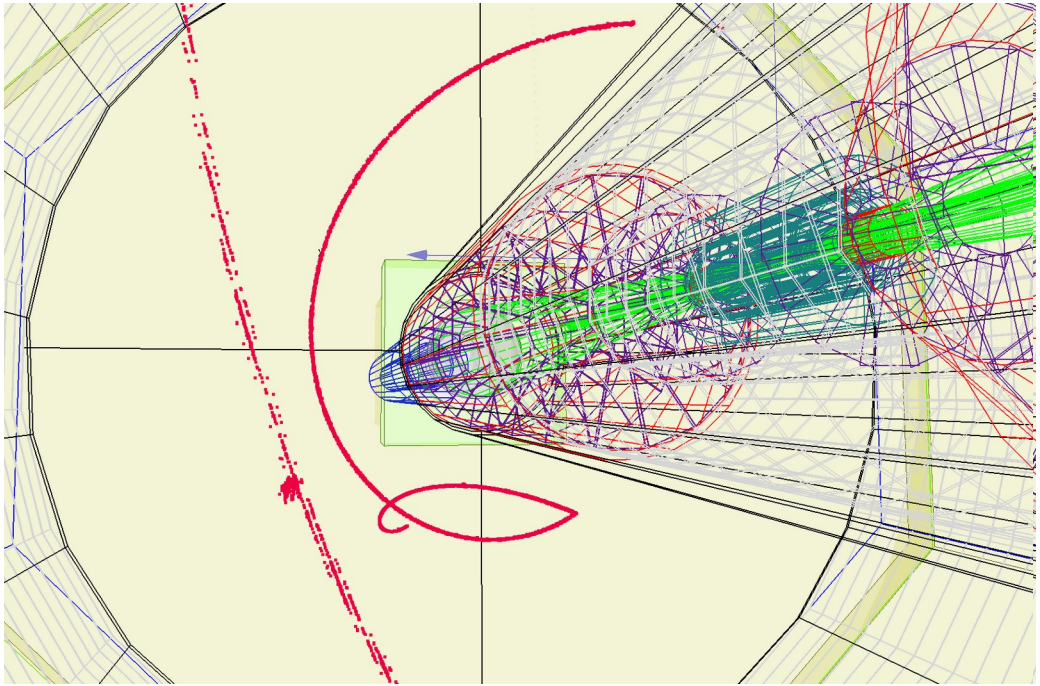
- based on **ILCSoft v01-19-02** (?), I only have an archive with the above packages from Peter Kluit
- no documentation, tracking of changes, etc.

→ Managed to run it with ILCSoft v02-02-03 and DD4hep v01-11-02 with Kees' additions
→ Using **55 µm** version – 990 µm is based on interpolation and gave strange results
→ I rely mostly on stuff left in ILDPerformance (**parsed xml steering files**, some output samples) and the PhD thesis

Digitisation



55 µm sensitive volumes



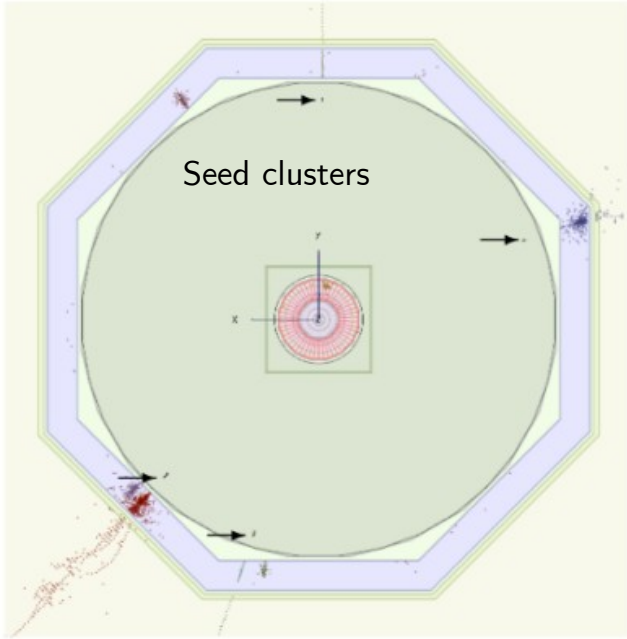
990 µm sensitive volumes

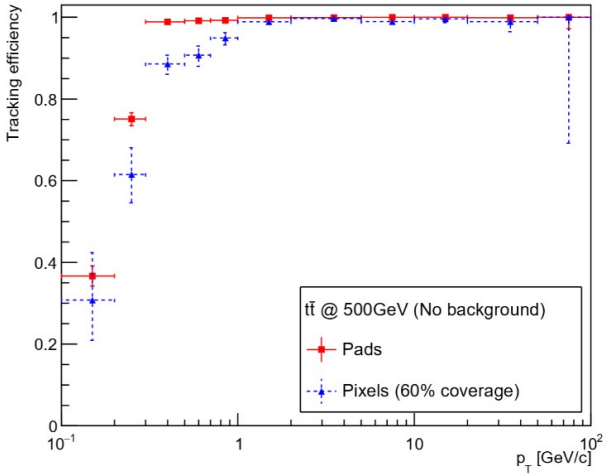
Pattern recognition (KL PhD thesis)

- Seed cluster finding in **ranges of pixel rows** going inwards

  → Fill hits into bins in Φ angle, take bin with most hits and its neighbours
- Init. track with a **straight line** through average hit position to the IP
- After additional cuts fit the seed with Kalman filter

  → add and fit more hits going inwards (and outwards if possible)





- "The difference between a pad readout and a pixel readout is almost exclusively due to **differences in the pattern recognition**."
- More focus needed for the **lower p_T**

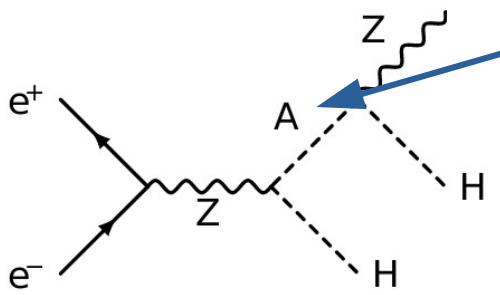
Case study: long-lived particles

- Computation time huge for pixel-TPC event simulation & reco.

→ LLP decays serve as a general case and good tradeoff between event complexity and computation time

- We're only interested in tracks inside the TPC anyway

For tests, the same sample as in LLP analysis was used – Inert Doublet Model with very small mass splitting between scalars



Long-lived, with $c\tau = 1\,\mathrm{m}$

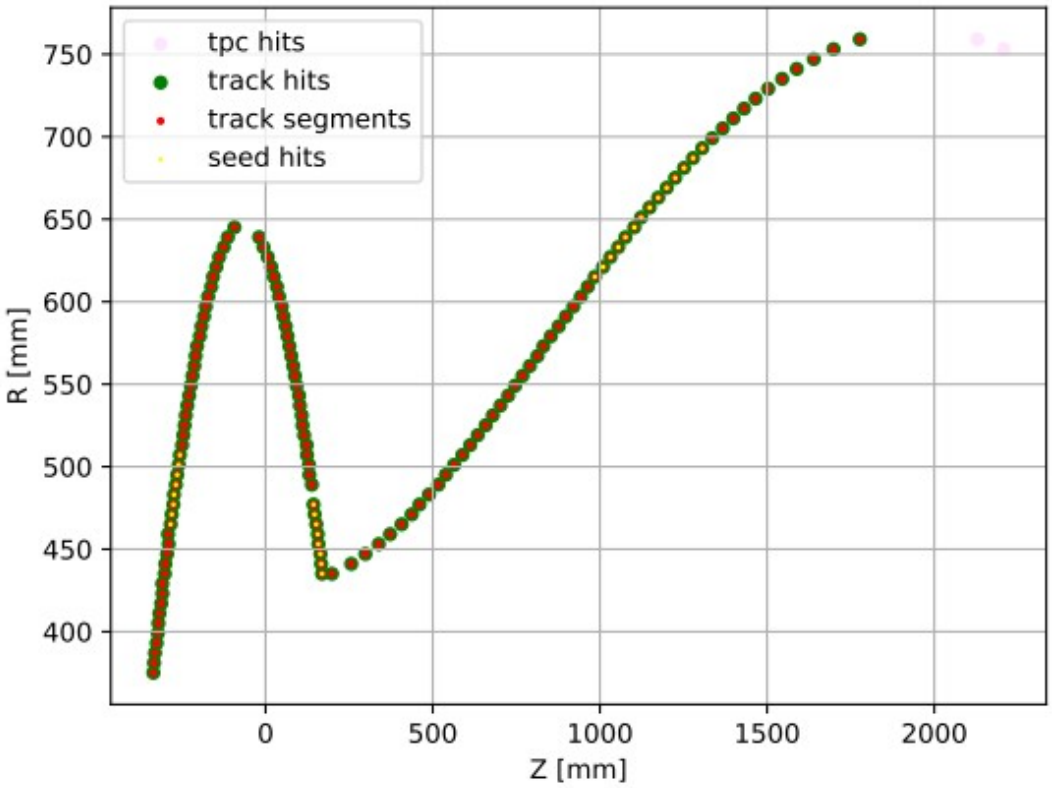
$m_A - m_H = 1\,\mathrm{GeV}$

- H stable and invisible
- $Z^* \to \mu\mu$

Provides the low pT we want to focus on

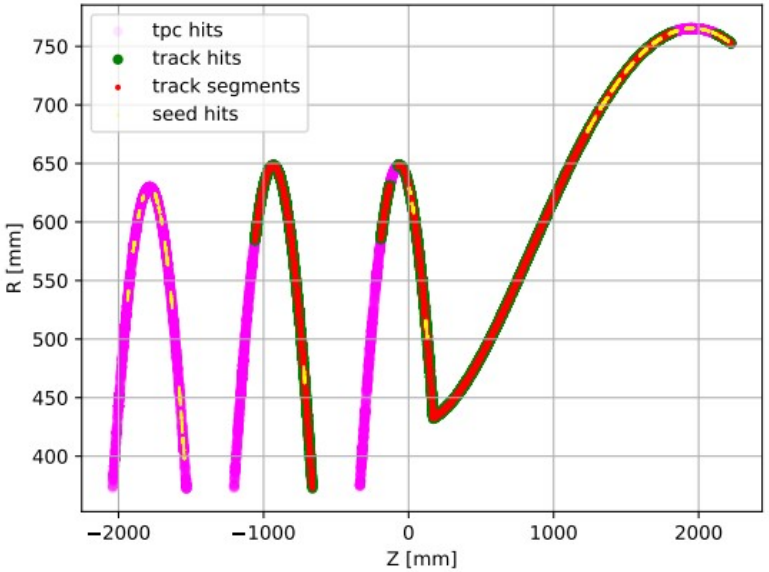
Clupatra algorithm (standard ILD)

- Nearest-neighbour (NN) clustering for seed finding, also in pad row ranges going inwards
- Fit and extend the seeds inwards and backwards using Kalman filter
  → track segments
- Re-cluster in leftover hits and force into clusters based on pad-row multiplicity
- Merge track segments (curlers separately)



Clupatra + pixel TPC

Kees' "phi-binning transform"

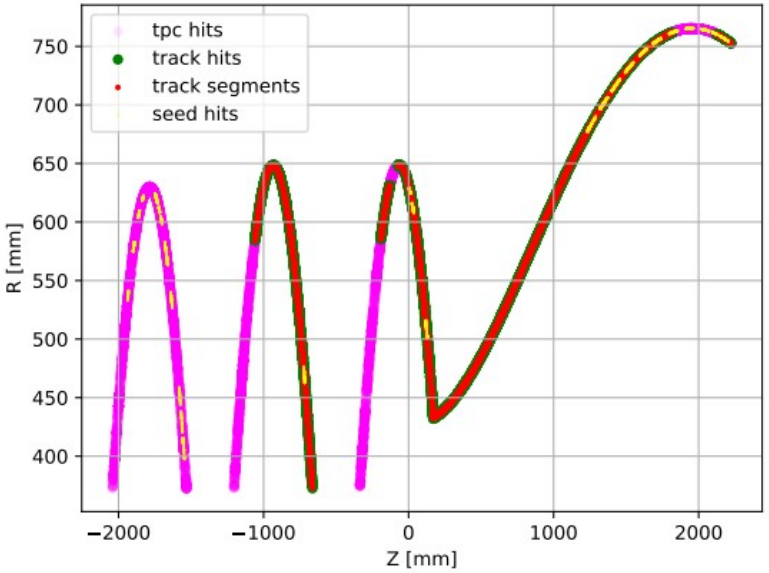


Note: parameters taken from a parsed Marlin xml steering file

- Too many seeds and output tracks
- Problem with extending the seeds

Clupatra + pixel TPC

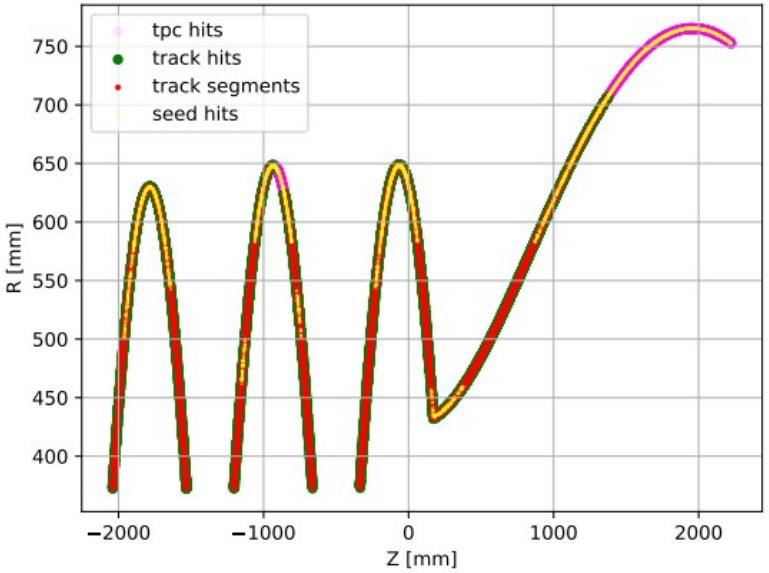
Kees' "phi-binning transform"



Note: parameters taken from a parsed Marlin xml steering file

- Too many seeds and output tracks
- Problem with extending the seeds

NN-clustering with the same settings



Based on Clupatra with some necessary technical changes for the pixel. In addition

- decreased errors of initial helix
- if adding hits inward fails, try backwards first

→ still too many seeds, they look unclean

Can we improve?

The main problem: extending the seeds does not work as expected

- Seeds found by NN-clustering seem not very clean / uniform
- Hit density for a track varies more in the pixel-TPC

  → maybe change clustering algorithm?

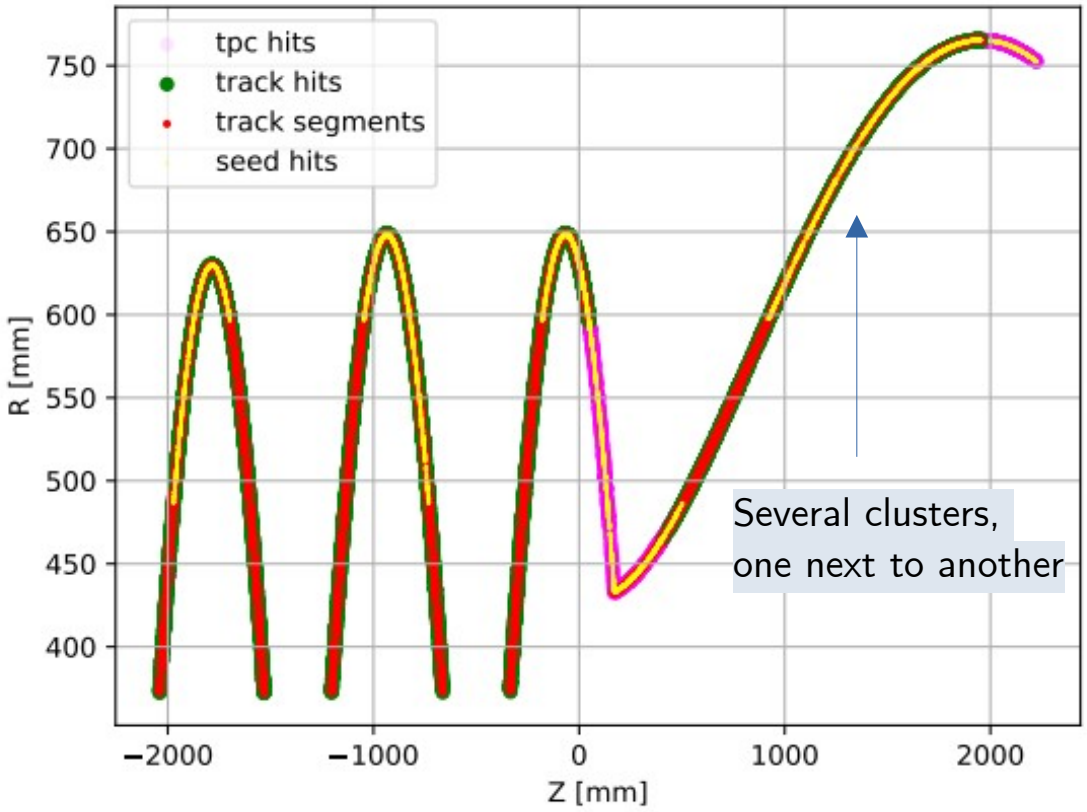
- HDBScan is a clustering algorithm based on metric that accounts for density/sparsity
- There is an implementation in C++ under MIT license based purely on STL library

Improvement attempt - HDBScan

Seed clusters look cleaner and more robust

However,

- Still too many seeds (due to the problems with extending them)
- Memory-related issues in events with more hits



Summary

- Some effort made to revive the model of ILD design with the TPC readout based on pixels
- Technically challenging, with potential issues related to versions compatibility
- Digitised output using model with 55 µm pixel size seems reasonable
- Improvement needed for tracking

- Tests of tracking performed on events with muons from displaced vertex, as a tradeoff between complexity and computation time
- Results for the approach similar to "standard" don't look repelling, but far from good
- Any input more than welcome