Track Reconstruction in MPD's TPC Using Acts (A Common Tracking Software)

Alexander Kamkin[1,2] (kamkin@ispras.ru), Pavel Belecky[2]

[1]Plekhanov Russian University of Economics (PRUE)
[2]Ivannikov Institute for System Programming of the Russian Academy of Sciences (ISPRAS)

JINR, Dubna | November 9, 2022

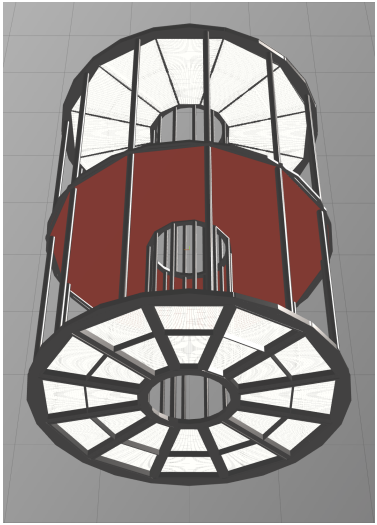
Acts Common Tracking Software

- **Experiment-independent** toolkit for particle track reconstruction
- Implemented in **modern C++** (`-std=c++17`) w/ minimal dependencies
- Result of redesign/refactoring of **ATLAS** tracking software (Athena)
- Designed for **multi-threaded** data processing (thread safe)

- https://acts.readthedocs.io
- https://github.com/acts-project/acts/

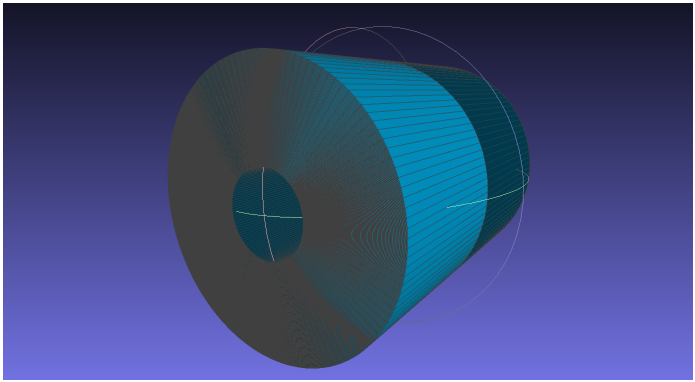
Track Reconstruction Steps

- **Digitization**
  - ⇒ Measurements (bound to sensor surfaces)
- **Space point making**
  - ⇒ 3D points (in global coordinates)
- **Track seeding**
  - ⇒ Seeds / proto tracks (3-point tracklets)
- **Track parameters estimation**
  - ⇒ Initial track parameters $\langle l_1, l_2, \phi, \theta, \frac{q}{p}, t \rangle$
- **Track finding**
  - ⇒ Trajectories (arrays of measurements)



Digitization: Virtual Sensor Surfaces in TPC

Measurements in Acts are bound to surfaces ⇒ **virtual sensors** should be introduced

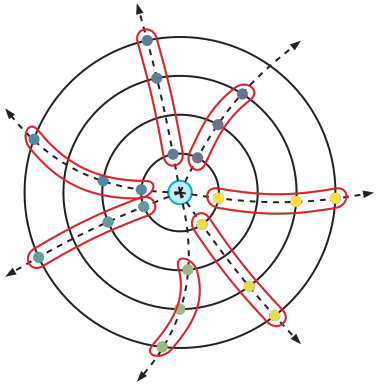


[*Idea*] J.D. Osborn, A.D. Frawley, J. Huang, S. Lee, H.P. Da Costa, M. Peters, C. Pinkenburg, C. Roland, H. Yu. *Implementation of ACTS into sPHENIX Track Reconstruction*. Computing and Software for Big Science, 2021.

Track Seeding

- Seed is a space-point triplet
- Enumeration of space point triplets
- Filtering triplets w/ *xy* and *rz* filters

Parameter	Value
$[z^0_{min}, z^0_{max}]$	[-30 cm, 30 cm]
$\mathrm{ctg}_{max}\theta = (\frac{\Delta z}{\Delta r})_{max}$	2.0
$[\Delta r_{min}, \Delta r_{max}]$	[1 cm, 6 cm]
Δz_{max}	15 cm
B_z	0.5 T
$Impact_{max}$	3 cm

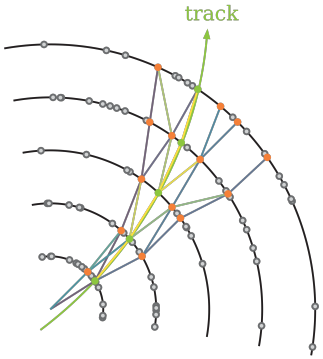


https://acts.readthedocs.io

Track Finding

- Track is a sequence of measurements
- Combinatorial Kalman filter (CKF)
- Branching for points on the same surface

Parameter	Value
$PropagationSteps_{max}$	1000
$EnergyLoss$	*true*
$Smoothing$	*true*
$N_{max}PerSurface$	5
χ^2_{max}	30
$Resolve\{Material, Sensitive\}$	*true*

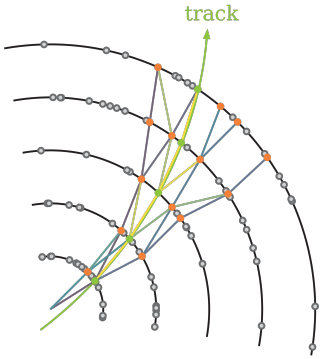


https://acts.readthedocs.io

Ambiguity Resolution

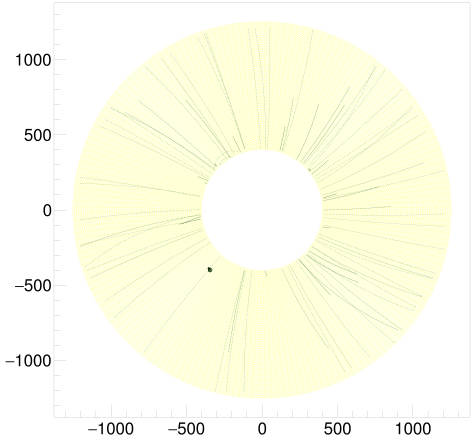
- CKF $\Rightarrow$ a tree-like set of track candidates
- *Acts has no general solution for filtering*
- We use a naïve approach (subject to change)
  - Long tracks take precedence over short ones
  - Track should contain certain % of new points
  - ... and a certain number of new points in a row

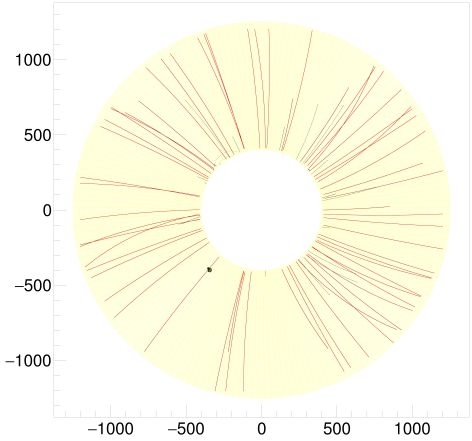
Parameter	Value
TrackMinLength	4
NewHitRatio	0.25
NewHitInRow	3



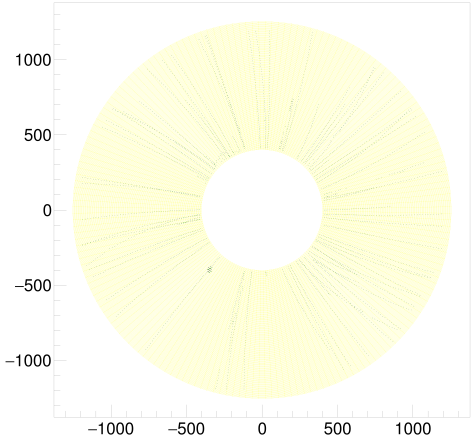
https://acts.readthedocs.io

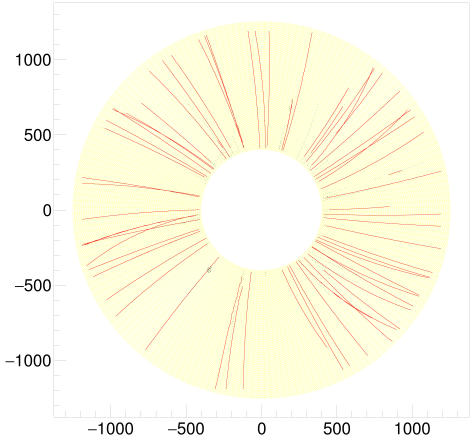
Example 1: Simple Event | Raw Simulation Points



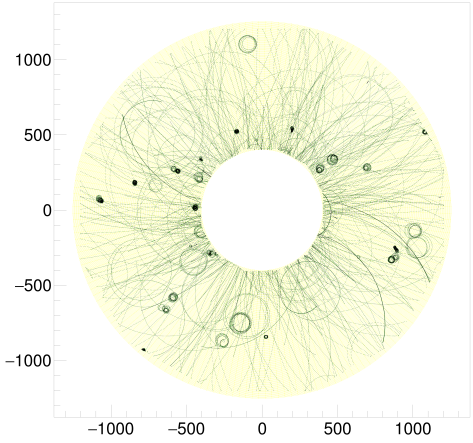


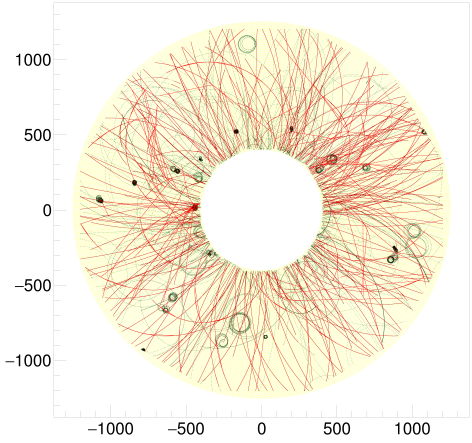
Example 2: Simple Event | Simulated TPC Hits



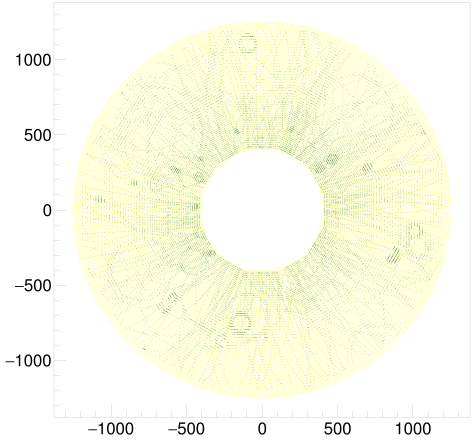


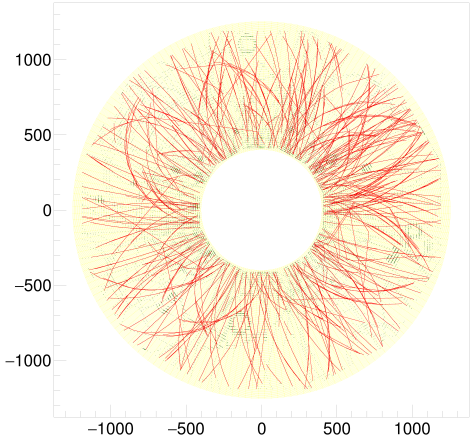
Example 3: More Complex Event | Raw Simulation Points





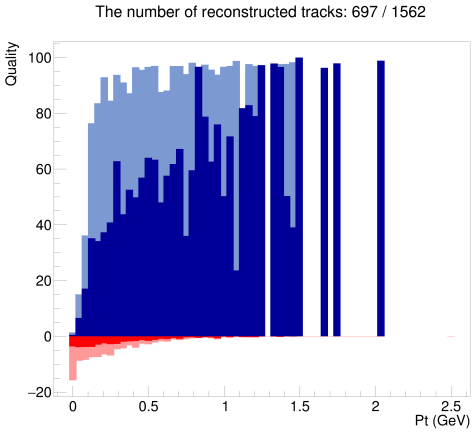
Example 4: More Complex Event | Simulated TPC Hits



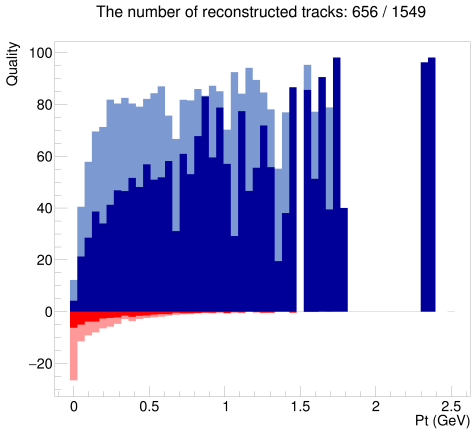


Tracking Efficiency (*work in progress*)

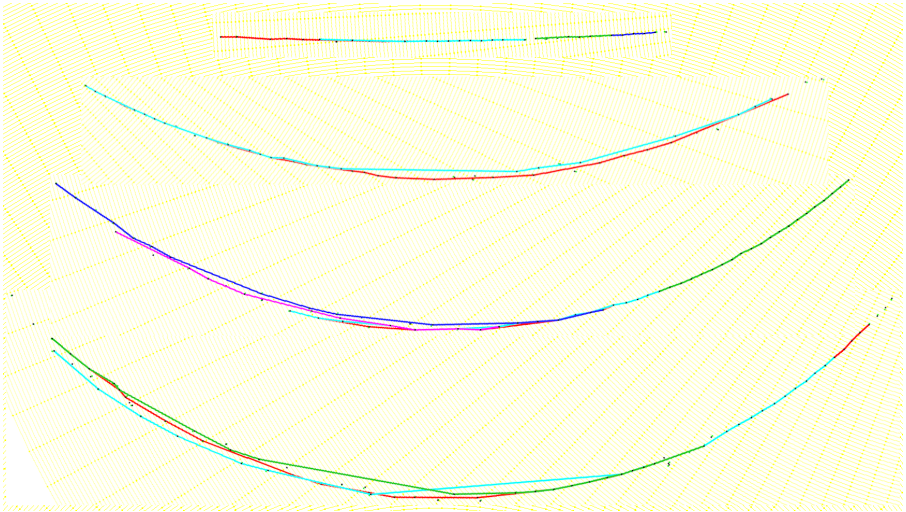
Raw points (high resolution)



TPC hits (realistic resolution)



Low-Quality for Some Tracks $\sim$ Piecewise Recognition



Further Research

- **Bringing virtual sensors in line w/ real pads geometry**
  - Adequate branching in Kalman filter during track finding
- **Ambiguity resolution and track merging techniques**
  - Many factors: track lengths, χ^2, shared hits, "holes", etc.
- **Deeper study of the track finding efficiency**
  - Identifying causes of "bad" cases, more test data, etc.
- **Comparison w/ the existing track finding module**
  - Need to think about comparison metrics...

Thank You!