



Tracking pattern recognition developments

Yue Shi Lai

LBL NSD

Detector-1 Tracking WG, Aug 04 2022

Seeding for Detector-1

- Essential ingredient of a tracking pipeline
- Even in detector study, needed for realistic performance
 - Efficiency, fake rate, multiple reconstruction, etc.
 - Beam condition (synchrotron radiation, beam gas interaction)
- Code now included in DD4hep/Juggler grew out of ATHENA
 - Had its share of ACTS issues and bugs due to the high field strength and tracker size
 - Only recently discovered all ACTS issues resulting in a Juggler plugin with the needed workarounds
 - Now works for efficiency studies with $p > 100 \text{ MeV}/c$ and $|\eta| < 4$
- Parallel effort from ECCE
 - By Sebastian Tapia (Iowa State Univ.), supervised by Joe Osborn (ORNL)
 - Never committed to ECCE repository (*private* GitHub)
 - Likely never encountered the ACTS issues discovered with ATHENA/DD4Hep/Juggler
 - The lower field and tracker size allowed a set of parameters for $p > 1 \text{ GeV}/c$ and $|\eta| < 3$, for the ECCE performance plot
 - “used [...] huge search windows”

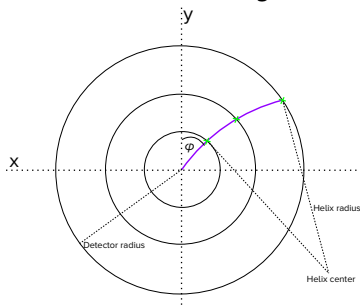
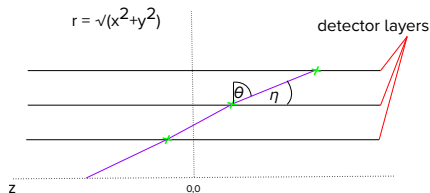
Seeding Code Landscape

- ACTS (ATLAS)
- Several CMS-derived, public code
 - FKDTree <https://github.com/felicepantaleo/FKDTree>
 - ⇒ Abandoned (last commit March 2018)
 - ⇒ Undocumented
 - TrickTrack <https://github.com/HEP-SF/TrickTrack>
 - ⇒ Semi-abandoned (last commit Feb. 2020)
 - ⇒ Obscure (hard to find via Google)
 - Close to what is used with CMS Run-3
- Documentation, development, responsiveness of developer clearly favor choosing ACTS
- From TrackML: Not trivial to beat traditional seeder, possibly even less so without the LHC-like multiplicity
 - ⇒ Unlikely to be priority until further in the detector development

ACTS Seeding

Figures from <https://acts.readthedocs.io/en/latest/core/seeding.html>

- Triplet generation (not limited to 3 layers, and works for forward configuration)



- Triple-loop filtering based on reasonable curvature, compatible η , backprojection to the vertex z range, etc.
- Double-loop filtering, weighting seeds based on mergeability of seed groups (ideally merge to seeds with 5 space points), backprojection to the lowest vertex z and ρ possible
- Single-loop filtering to only retain the highest quality N seeds per middle space point

— Particle path within detector
+ Particle measurement on detector surface

Implementation Detail

- A plug-in named TrackParamACTSSeeding, near drop-in replacement for TrackParamTruthInit
- Input is the raw hits (seeding is done mostly in global coordinate, with one exception)
 - Local coordinate “measurements” currently do not work, there seem to be issues in global-to-local transform (material description or geometry issues?)
- Reuses the source links from Kalman filter, and will reorder internally (source links and hits can have different order)
- Differences vs. how ATLAS runs seeding:
 - Seeds are pre-filtered to 1 seed/track
 - ATLAS would reconstruct (using CKF) all seeds and then filter

Bugs, etc. with ACTS

- The code underwent 4 significant versions to track compatibility-breaking ACTS interface changes around versions 12, 15, 19
- Studies made with this plugin with ATHENA also contributed to the discovery or fix of at least two bugs in the ACTS code
 - For very forward tracking, the number of η bins can go to 0, then a difficult-to-trace segfault (masked by other code before 12, then “suddenly appeared”)
 - ⇒ Fixed around ACTS 15
 - For very forward tracking, low-ish min. p_T , and high magnetic field, the number of ϕ bins can become NaN (NaN converts on x86 to $2^{64} - 1$ for `size_t/uint64_t`, causing STL `bad_alloc`)
 - ⇒ How to fix is not very clear, related to prior performance issues, but ACTS developer notified
- Some of problems took long time and require intimate knowledge of ACTS to track down, e.g. forward low p inefficiency was caused by an

```
if ((bCoef * bCoef) > (1 + aCoef * aCoef) /  
    m_config.minHelixDiameter2) {...}
```

- There is an older version already merged into Juggler
 - Good performance for $p > 2 \text{ GeV}/c$ throughout the acceptance
 - Good performance for $p < 2 \text{ GeV}/c$ up to $|\eta| < 2$
- A version with good *efficiency* for all p and $|\eta| < 4$ is being tested and will be submitted to Juggler
 - Performance is reasonable for efficiency, but not Δp for $p < 2 \text{ GeV}/c$ (offset towards $+\Delta p$)
 - Also the $|z|$ vertex compatibility has to be a few 100 mm (otherwise inefficient)
 - ⇒ Hints at potential material description/geometry issues
- Not really optimized (yet) for user experience, none of the parameters are not Python configurable, requires editing of the C++ code and recompile

How to Use

```
File Edit Options Buffers Tools Python Help
[Icons] Save Undo [Icons]

from Configurables import Jug_Digi_SimTrackerHitsCollector as SimTrackerHitsCollector
from Configurables import Jug_Digi_SiliconTrackerDigi as TrackerDigi
from Configurables import Jug_Reco_TrackerHitReconstruction as TrackerHitReconstruction
from Configurables import Jug_Reco_TrackingHitsCollector2 as TrackingHitsCollector
from Configurables import Jug_Reco_TrackerSourceLinker as TrackerSourceLinker
from Configurables import Jug_Reco_TrackParamTruthInit as TrackParamTruthInit
from Configurables import Jug_Reco_TrackParamClusterInit as TrackParamClusterInit
from Configurables import Jug_Reco_TrackParamVertexClusterInit as TrackParamVertexClusterInit
from Configurables import Jug_Reco_TrackParamACTSSeeding as TrackParamACTSSeeding
from Configurables import Jug_Reco_TrackFindingAlgorithm as TrackFindingAlgorithm
from Configurables import Jug_Reco_ParticlesFromTrackFit as ParticlesFromTrackFit
#from Configurables import Jug_Reco_TrajectoryFromTrackFit as TrajectoryFromTrackFit
from Configurables import Jug_Reco_SimpleClustering as SimpleClustering

algorithms = [ ]

-:~%- track_reconstruction.py 16% L52 Git:master (Python AC company EIDoc)
```

```
File Edit Options Buffers Tools Python Help
[Icons] Save Undo [Icons]

outputSourceLinks="TrackSourceLinks",
outputMeasurements="TrackMeasurements")
algorithms.append( sourceLinker )

acts_seed trk_init = TrackParamACTSSeeding('acts_seed trk_init',
inputSourceLinks = sourceLinker.outputSourceLinks,
inputMeasurements = sourceLinker.outputMeasurements,
inputHitCollection = trk_hit_col.trackingHits,
outputInitialTrackParameters = 'InitTrackParamsSeeding',
OutputLevel = DEBUG)
algorithms.append(acts_seed trk_init)

## Track param init
truth_trk_init = TrackParamTruthInit('truth_trk_init',
inputMCParticles="MCParticles",
outputInitialTrackParameters="InitTrackParams")
algorithms.append( truth_trk_init )

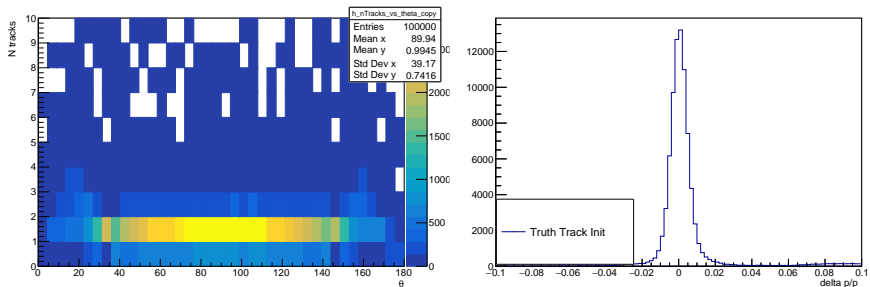
# Tracking algorithms
trk_find_alg = TrackFindingAlgorithm('trk_find_alg',
inputSourceLinks = sourceLinker.outputSourceLinks,
inputMeasurements = sourceLinker.outputMeasurements,
inputInitialTrackParameters="InitTrackParamsSeeding",
outputTrajectories="Trajectories")
algorithms.append( trk_find_alg )

parts_from_fit = ParticlesFromTrackFit('parts_from_fit',
inputTrajectories="trajectories",
outputParticles="reconstructedParticles",
outputTrackParameters="outputTrackParameters")
algorithms.append( parts_from_fit )

-:~%- track_reconstruction.py 81% L234 Git:master (Python AC company EIDoc)
```

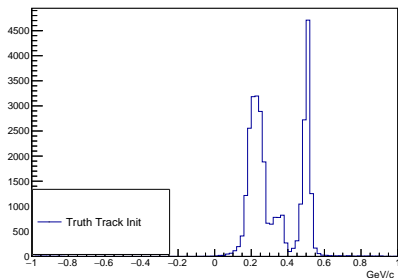
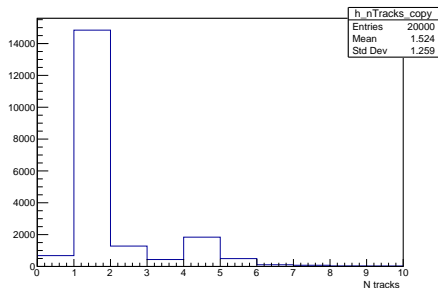
- Edit e.g. reconstruction_benchmarks/benchmarks/tracking/options/
/track_reconstruction.py
- Import Configurables.Jug_Reco_TrackParamACTSSeeding as something
more weildy
- Connect input and name output as e.g. 'InitTrackParamsSeeding'
- Change the input to the (inappropriately named)TrackFindingAlgorithm
(fitting and not really track *finding*) to 'InitTrackParamsSeeding'

Performance



- ATHENA tracking reconstruction benchmark (slightly) modified to give all θ and a 2D N_{track} vs. θ (175° is $\eta \approx 3.1$)
- $N_{\text{track}} = 0$ is inefficient, $N_{\text{track}} = 1$ is single seed, $N_{\text{track}} > 1$ is multiple seed
- Still ATHENA/with a 3 T field, 1–2 GeV/c π^+ (the most critical region)
- Ignore “Truth Track init” (someone hardcoded it into the ATHENA tracking benchmark)
- ACTS recommendation is to use cuts on fully reconstructed tracks to remove multiple reconstruction, but performance appear good enough to be without

Performance



- Select on $2 < \eta < 2.5$
- Again, ignore “Truth Track init” (someone hardcoded it into the ATHENA tracking benchmark)
- Highly efficient, $\epsilon > 0.95$
- Issues remain with Δp

Way Forward

- Track Δp performance hints at potential material description or geometry in DD4hep/Juggler, that are not fixable from the seeding code side
 - ⇒ Likely not a show-stopper for background studies with efficiency/fake rate only
- ACTS developers encourages move towards a new orthogonal range search seed finder
 - Main advantage is bin-less, and avoids all the ACTS bugs discovered so far with binned spacepoint group
 - But not clear how well-tested for low p_T and forward, as we experienced sofar with the ACTS test coverage
 - ⇒ Likely not a priority until after background studies