Future Development Prospects

Noemi Calace - noemi.calace@cern.ch
Paul Gessinger, Andreas Salzburger

ACTS Developers Workshop
8 November 2023

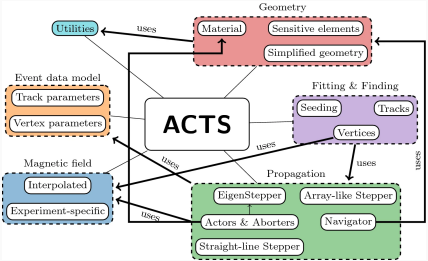
General Introduction

ACTS is becoming a well established tool for track reconstruction for many experiments

- More and more crucial to provide a consistent and reliable set of tools

Several components available in the ACTS repository and covered in this contribution:

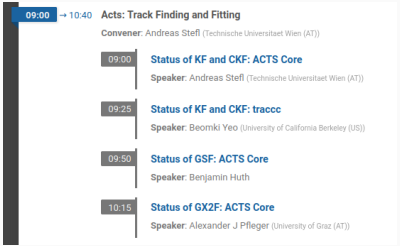
- Geometry, Propagation and Navigation
- Track Finding and Fitting
- Vertex Finding and Fitting
- Contextual geometry handling
- Event data model, track parameterisation, algebra plugin, ...

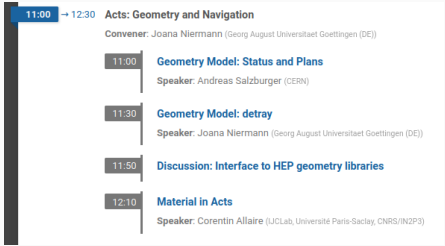


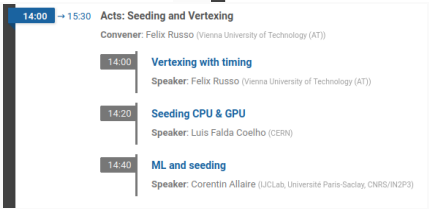
Workshop Goal: Gathering client feedback to assess whether their requirements have been fulfilled and identifying areas for improvement.

General Introduction

A lot of details on the several topics also in tomorrow's sessions:



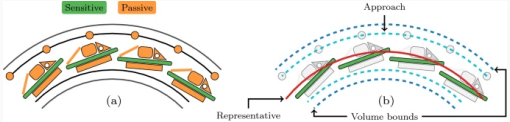




Geometry and Navigation component

The geometry represent a fundamental component for track reconstruction.

Detector description translated into ACTS tracking geometry from different sources e.g. DD4Hep, gdml, TGeo, etc.



New geometry model, based on detray experience, has been worked on (Proposal presented last year):

- Geometry implementation without layers (the concept stays), only volumes exist
- Local volume navigation is performed via a delegate optimised for the given environment

This model is already exercised by some experiments, and the (new) navigator is almost ready for extensive validation: see Andreas' contribution

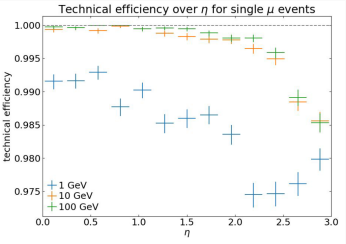
Concern: How to phase out the legacy implementation and help experiments to migrate to new detector and navigation model

Pattern Recognition Component

Continues to attract a lot of interest from various experiments:

- Seed Finding and Combinatorial Kalman Filter (CKF): ATLAS, ALICE, NA60+, ePIC, FASER, LDMX, and many more: Friday's contributions

Several studies (e.g. "Performance evaluation for ODD using ACTS", CTD 2023) have proven the necessity of improving the implementation of the CKF, extending its capabilities.
Development Proposal: Providing another implementation of CKF "reverting" the relation with the propagation



- Other implementations of pattern recognition algorithms: Hough transform seeder and Exa.TrkX Plugin

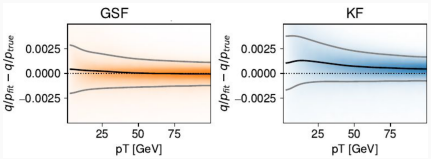
Development Proposal: Expanding pattern recognition methods to include other detector types. Are experiments interested in something specific and willing to contribute?

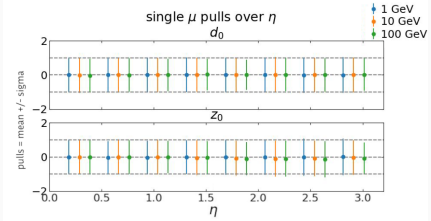
Fitters

Several Fitter implementations available: Tomorrow's contributions

- Implementation of Kalman Fitter and Gaussian Sum Fitter extensively validated along the years
  - Results with the ODD are very promising: "Performance evaluation for ODD using ACTS", CTD 2023
    Question: Have experiments used them? How do they perform? Is tuning necessary?
- Global χ^2 Fitter is being implemented
  - Building up all relevant components to the fitter, still some work before deploying

Development Proposal: Implementing Deterministic Annealing Fitter. Is this something needed from experiments? E.g. for drift chambers.



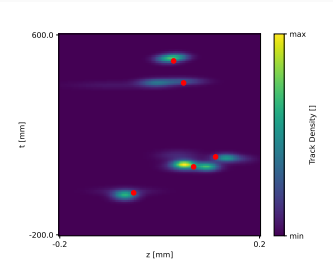


Vertex Reconstruction

Several implementations of Primary Vertex Reconstruction, also used in production by experiment (e.g. ATLAS Track Reconstruction for LHC Run 3)

Vertex Finders: Iterative Vertex F., Adaptive Multi-Vertex F., Z Vertex F., Grid Density Vertex F., Track Density Vertex F., ...
Vertex Fitters: Full Billoir Vertex Fitter, Adaptive Multi-Vertex Fitter

Work is currently done to validate the handling of time component in vertex reconstruction: Tomorrow's contribution



→ Several developments can be foreseen in this area depending on the experiment's interests

Development Proposals:

- Generalisation of the Track linearisation using propagator to harmonise primary and secondary vertex reconstruction (the latter to be implemented) in common kernel
- Including kinematic constraints in Vertex Reconstruction (e.g. mass of the decaying particle) to improve robustness of fit convergence

Tracking Geometry Alignment

- Geometry context is used to account for changes in event conditions, and update e.g. transform associated to the surfaces. It is passed to all geometry-dependent algorithms in ACTS, so that:
  - Algorithms can access the right version of geometry
  - Alignment algorithm can create or update a version of geometry
- Basic ingredients are present: Geometry context, track parameters, measurements, derivatives w.r.t. track parameters and alignment parameters.
  - Ready to get in: Add derivatives with alignment parameters as rotation around local axes, as well as rotations around bigger structures (staves, layers, volumes)
- Potential for exciting developments, yet the current situation is progressing slowly and would benefit from increased involvement and contributions.

  Development Proposals:
  - Interfacing with Millepede
  - Implementing incremental testing workflow
  - End-to-end machine learning approach being investigated

Event Data Model

- Development of high-level Track EDM well underway
- Examples framework largely migrated at this point

Development Proposals for wider ecosystem integration:

- Extend EDM4hep integration
  - currently can write Tracks + Track States, but lose relationships like truth particles, hits, etc.
  - Ideally: want to be able to round-trip sim-hits and truth particles
- More robust integration with PODIO
  - TrackContainer exercise showed custom PODIO EDM for persistency is promising
  - This would enable full information preserving IO without complicated ntuple readers/writers

Additional Topics

- ACTS depends on Eigen math library for linear algebra

  Development Proposal: Encapsulation of algebra operation back-end (as done in traccc project). Can be used for library bench-marking in realistic applications as well as dealing with unmaintained libraries.

- Track parameterisation: most suitable for collider-type experiments, and not e.g. for fixed-target and telescope setups

  Development Proposal: Evaluating implementation of compile-time definition of track parameters $\rightarrow$ In some parts of the code we assume the particular definition of track parameters, e.g. ϕ, θ.

Conclusion and Discussion

- ACTS has achieved significant progress along the years
  - All components are in very active development: Geometry and Navigation, Track Finding, Track Fitting, Vertexing, ...
  - However, some areas would benefit from further development, and some have still missing elements

You input is crucial!

- Experiments using ACTS are encouraged to provide valuable feedback
- Collaboration between developers and clients is essential to improve and expand ACTS capabilities for the benefit of the HEP community

.. so, let's start!