LUXE Geometry using DD4hep Plugin

Yufeng Wang
Mar 21, 2025

HELMHOLTZ

DESY.

Motivation

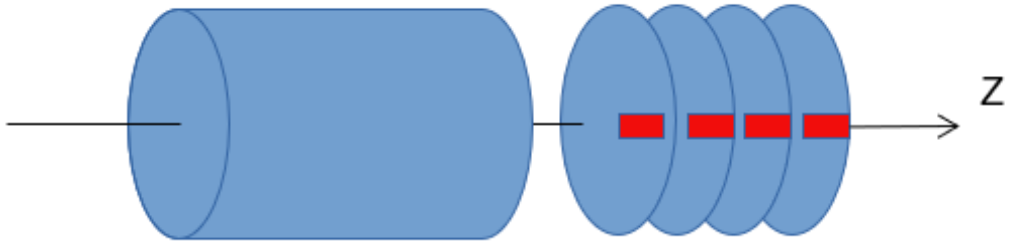
- Update the current MarlinACTSTracking (based on a very old version of Acts 13, effectively a copy&adaption from MuonCollider software) for LUXE tracking

- Apply geometry loading, DD4hep plugin for ACTS, etc

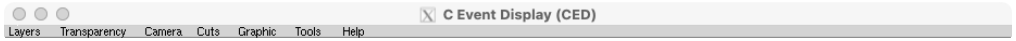
Mainly Yee's work

LUXE geometry and simulation

- Geometry
  - 4 “endcap” disk layer
    - each has 2 layers
      - each has 9 modules (1 overlap)
  - Parameters defined in LUXETrackerAsEndcap.xml, translated into LUXETrackerAsEndcap.root through TGeo in ACTS v13

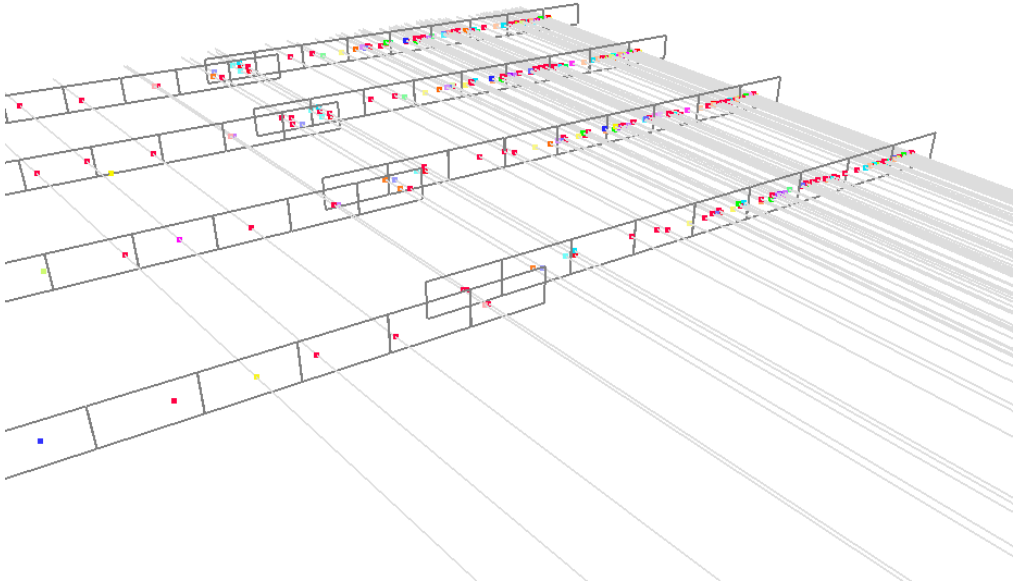


(↑ from Michal’s report)



- Simulation
  - (Irrelevant to the ACTS updates)

```
ddsim --compactFile LUXETrackerAsEndcap.xml
--numberOfEvents 1
--enableGun
--gun.multiplicity 100
--gun.particle e+
--outputFile positrons_100_sim.slcio
```



Major changes w.r.t. Acts 13 version

Mainly taken from branch porting_v32 of MuonColliderSof/ACTSTracking.

- New configuration parameters for seeding
- New way to get range of elements with requested geoID
- New GridFinder using R & Z parameters
- Particle hypothesis (muon, positron, …) for seeding.
- New classes: VectorTrackContainer, VectorMultiTrajectory, XXXContext…
- Updates in some calibration & SeedSpacePoint functions
- Updates in Surface, SourceLinkAccessor
- …
- In short, it works with ACTS 32 now. checking seeding & tracking efficiency.
  - Built 18 surfaces *4 =72 layers (in total); reconstructed 365 hits → 275 ACTS space points.

Geometry building procedure (old)

ACTSTrackingProc.cxx
ACTSProcBase.cxx

trackinggeometry() is used to find **surface** in the main tracking code.
Then, **surface** (containing **geometryID**) is used to:

- Convert LCIO hits to ACTS hits
- Create sourcelinks between measurement and hit
- Get track parameters once seeds are found
- (hits & tracks back to LCIO collection)

---

trackingGeometry() is build through **buildDetector()** in the base code.

trackingGeometryBuilder::Config tgConfig → cylinderGeometryBuilder

- materialDecorator (_matFile.json)
- trackingVolumeBuilders
  - CylinderVolumeBuilder ← layerBuilder ← Acts::TGeoLayerBuilder ← **TGeoManager**

TGeoMenager reads from _tgeoFile (ROOT).

Geometry building procedure (new)

ACTSTrackingProc.cxx
ACTSProcBase.cxx

trackinggeometry() is used to find **surface** in the main tracking code.
Then, **surface** (containing **geometryID**) is used to:

- Convert LCIO hits to ACTS hits
- Create sourcelinks between measurement and hit
- Get track parameters once seeds are found
- (hits & tracks back to LCIO collection)

---

trackingGeometry() is build through **buildDetector()** in the base code.

trackingGeometryBuilder::Config tgConfig → cylinderGeometryBuilder

- materialDecorator (_matFile.json)
- trackingVolumeBuilders
  - CylinderVolumeBuilder ← layerBuilder ← Acts::TGeoLayerBuilder **← DD4hep Detector**

dd4hep::Detector& dd4hepDetector = dd4hep::Detector::getInstance();

- DD4hep description (xml)
- Detector constructor (cxx)

_trackingGeometry = geometryService->trackingGeometry(); **← DD4hepPlugin**

Surface visualization

```
// visualization global geometry
Acts::ObjVisualization3D geoVis;

Acts::ViewConfig sConfig;
sConfig.triangulate = true;
sConfig.color = {200, 200, 200}; // grey

trackingGeometry()->visitSurfaces([&](const Acts::Surface* surface) {
  if (surface) {
    Acts::GeometryView3D::drawSurface(
        geoVis,
        *surface,
        geometryContext(),
        Acts::Transform3::Identity(),
        sConfig
    );
  }
});
```

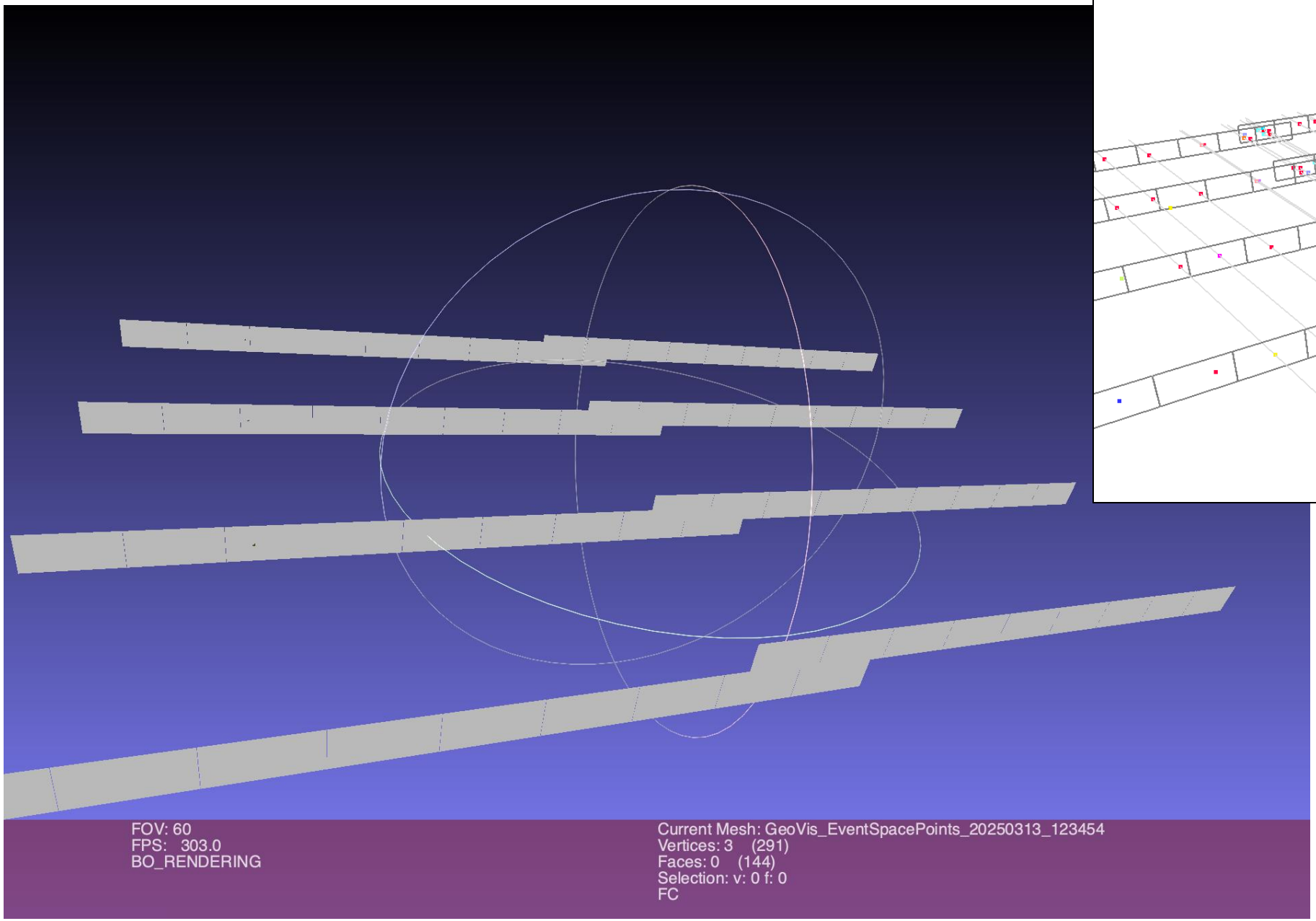
Traverse all ACTS surfaces

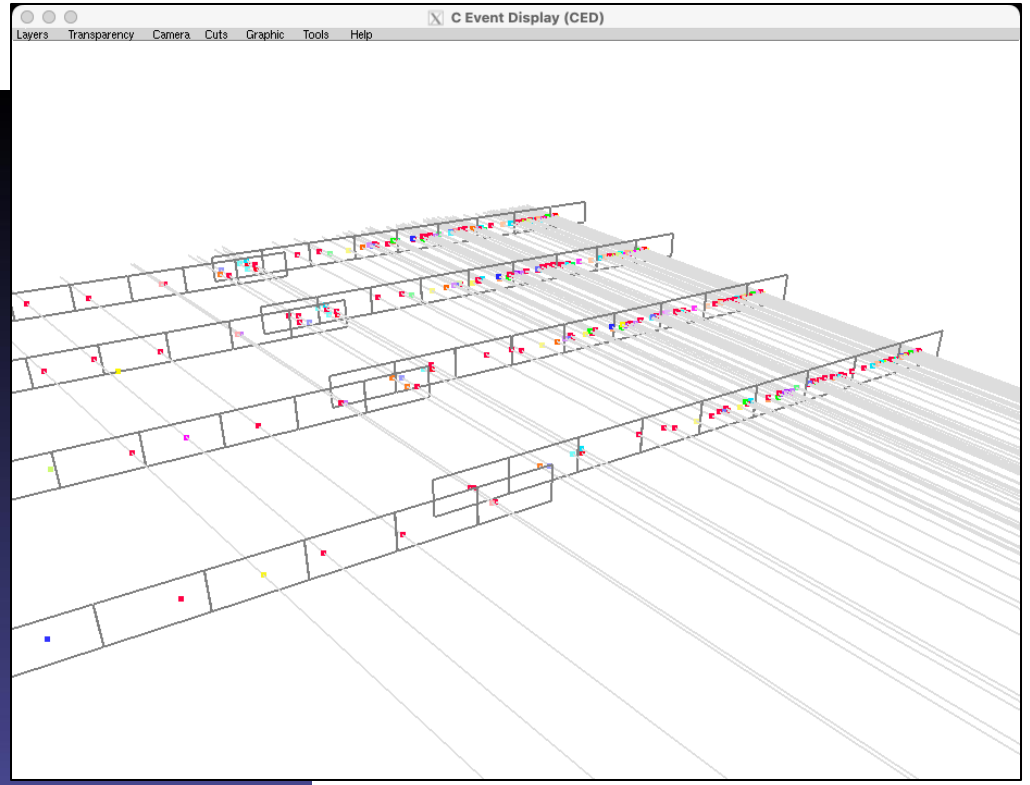
trackingGeometry & geometryContext built by cylinderGeometryBuilder

Saved an .obj file

- Cleaning up the codes to separate the geometry translation part
- visualize space points → still tiny. It's not a event display though, might not be necessary.

ACTS surfaces





detector

Back up