Detector Material Mapping with the Acts Project

Gabriel Farrugia
Supervisor: Paul Gessinger

August 2019

1 Introduction

Acts (A Common Tracking Software)[1] is a project whose main goal is to provide high-level track reconstruction modules that can be used for any tracking detector. One of the main modules used for modelling the structure of a detector is the Geometry module. This contains classes describing the subdivision of a detector's geometry into volumes with different layers, each with their specific geometric structure and bounds. The Material module is used for describing how material is distributed across different surfaces on the detector. This is required for taking into account the effect of different layers of the detector on particle tracks during particle propagation (such as multiple scattering and energy loss). For track reconstruction, a simplified geometry is used to describe each component of the detector and material is mapped into bins on thin "material layers".

2 Generating Material Maps

The first step of the project involved the generation of material maps using hard-coded detector specifications and the comparison of material tracks (particle tracks with material descriptions at relevant intervals) for the original and mapped material. As can be seen in Figure 1, the material distributions for the original and mapped material are almost identical, indicating that material maps (without binning) are still faithful to the original representation and can be used interchangeably.

After binning, the material distribution is seen to change slightly, depending on the number of bins chosen. For our purposes, an equidistant binning strategy with the same number of bins across both binning dimensions was chosen. This meant that a single parameter (number of bins in one dimension) could be varied which would determine the overall quality of the mapping. Figure 2 shows the comparison of different binning strategies ($n \times n$ bins where $n \in \{1, 20, 40, 100\}$). As can be seen, with an increasing number of bins we get decreasing amounts of material. This was attributed to the fact that the number of particles being propagated was kept constant, even with an increasing number of bins (i.e. finer subdivisions of each surface). Finer binning configurations were ending up with more empty bins (bins with no material), hence lowering the average material for certain values of η.

Increasing the number of particles propagated through the material layers reduced the number of empty bins up to a certain limit, where the number of empty bins could not be reduced further. This indicated the presence of a certain hole in the particle propagation coverage of the detector. At this point it was decided to create a visualization tool for material maps so that the material in each bin, on each layer, could be viewed and explored. This would enable us to find the location of the empty bins and identify the section which was not covered during particle propagation.

3 The PlyWriter Plugin

PLY (Polygon File Format) is a format used to store 3 dimensional objects represented as a number of flat polygons. By dividing individual surfaces into a number of polygons, we get an approximate representation of the surface (approximate as rounded surfaces such as cylinders will be divided into a number of smaller flat surfaces). By dividing each individual surface according to its binning specification we get one polygon per bin, each of which can be coloured to reflect the material contained in that bin.

[1] https://gitlab.cern.ch/acts/acts-core

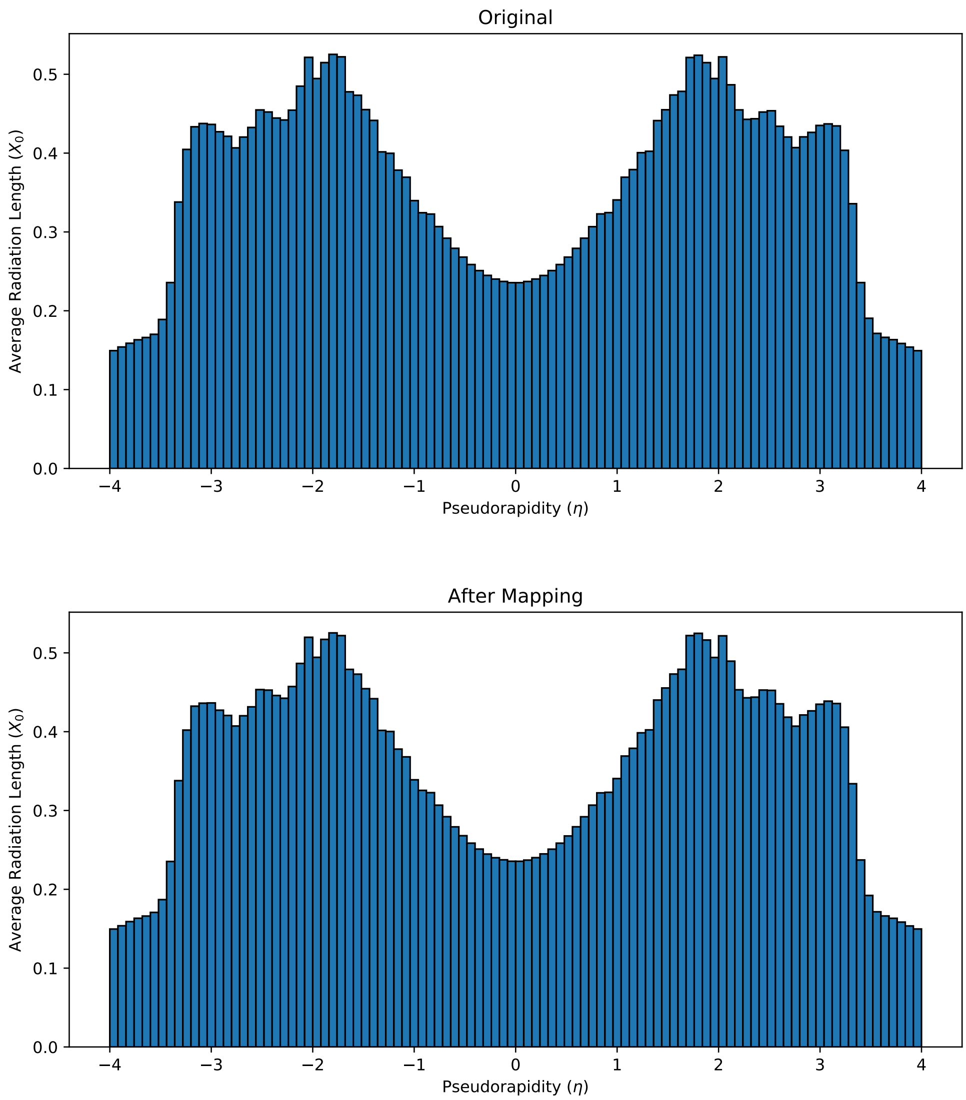


Figure 1: Comparison of material distribution using the original material description and the mapped material

3.1 Design

The `PlyWriter` class (inheriting from `FW::IWriter`) was created with the purpose of recursively exploring the geometry and writing out the necessary surfaces. On finding any `Acts::Surface`, a check is made to find the shape of the object which should be drawn (currently only cylinders, disks and planes are implemented since these are the only shapes present in the geometry used). The number of bins on the surface (in each dimension) is then extracted and an iterative procedure is carried out where the position of the four vertices of each bin is determined and the bin's average radiation length (X_0) is extracted and used to get a colour. Colours are representative of the material properties, i.e. a scale is used which is directly proportional to the material's X_0. For better distinction and identification of empty bins, a binary colouring scheme was initially used.

3.2 Application

The `PlyWriter` was used to generate models describing the material content of each bin on each material layer in the detector. Figure 5 illustrates the resulting models, generated with a binary option for colours where presence of material is indicated by green and absence is indicated by black. This highlights the black region around the beam line which seems to be growing in a conical shape from the origin, which indicates a problem with the pseudorapidity range for propagated tracks. Looking further into this and taking into account the required pseudorapidity values for

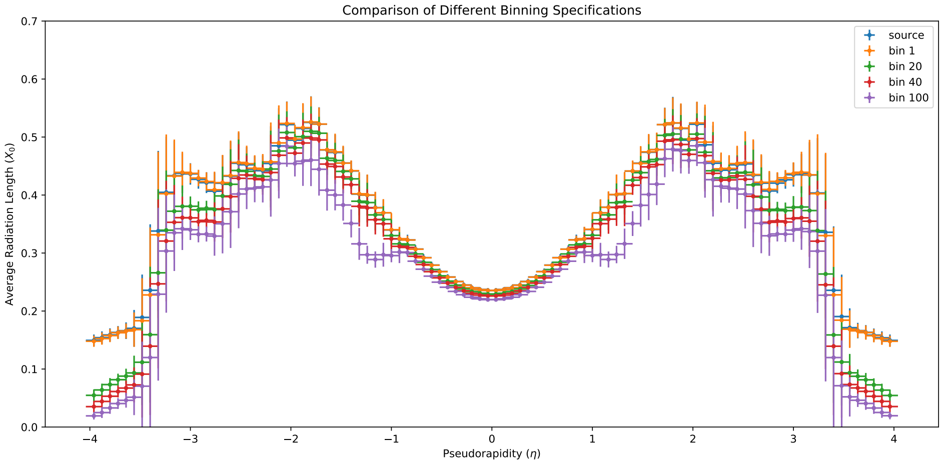


Figure 2: The effect of number of bins on material distribution

full coverage, it was confirmed that the problem lied in the default setting for pseudorapidity range (which was originally $(-4, 4)$ but should be changed to $(-6, 6)$).

3.3 Usage

Given a valid `FW::Sequencer` instance, as well as a `Acts::TrackingGeometry` instance and the program arguments contained in a `boost::program_options::variables_map`, a `PlyWriter` instance can be added to the writers. The `ACTFWPlyPlugin` should also be marked for linking in the relevant `CMakeLists.txt` file.

```
...
boost::program_options::variables_map vm = ... ;
...
FW::Sequencer sequencer = ... ;
...
Acts::TrackingGeometry *tGeo = ... ;
...
// Add the .ply file writer
FW::Ply::PlyWriter::Config config = FW::Options::readPlyWriterConfig(vm, *tGeo);
FW::Ply::PlyWriter tgPlyWriter = std::make_shared<FW::Ply::PlyWriter>(config);
sequencer.addWriter(tgPlyWriter);
```

4 Concluding Remarks

The scope of the project was originally to find a good binning configuration such that using material maps derived from said binning configuration produced propagation results similar to the original material. However, throughout the lifetime of the project, the need for additional tools to further explore material maps gave rise to the new goal of generating visualizations for the material layers of a detector. This was achieved by creating the `PlyWriter` plugin, which is a simple approach to generating three dimensional representations of material layers.

5 Future Work

- Find a way to generalize the `PlyWriter`'s functionality for any surface (without requiring the use of a dynamic cast to a specific surface type).
- Implement per-surface colouring for better material distinction on individual surfaces.
- Add in functionality for non-equidistant binning.

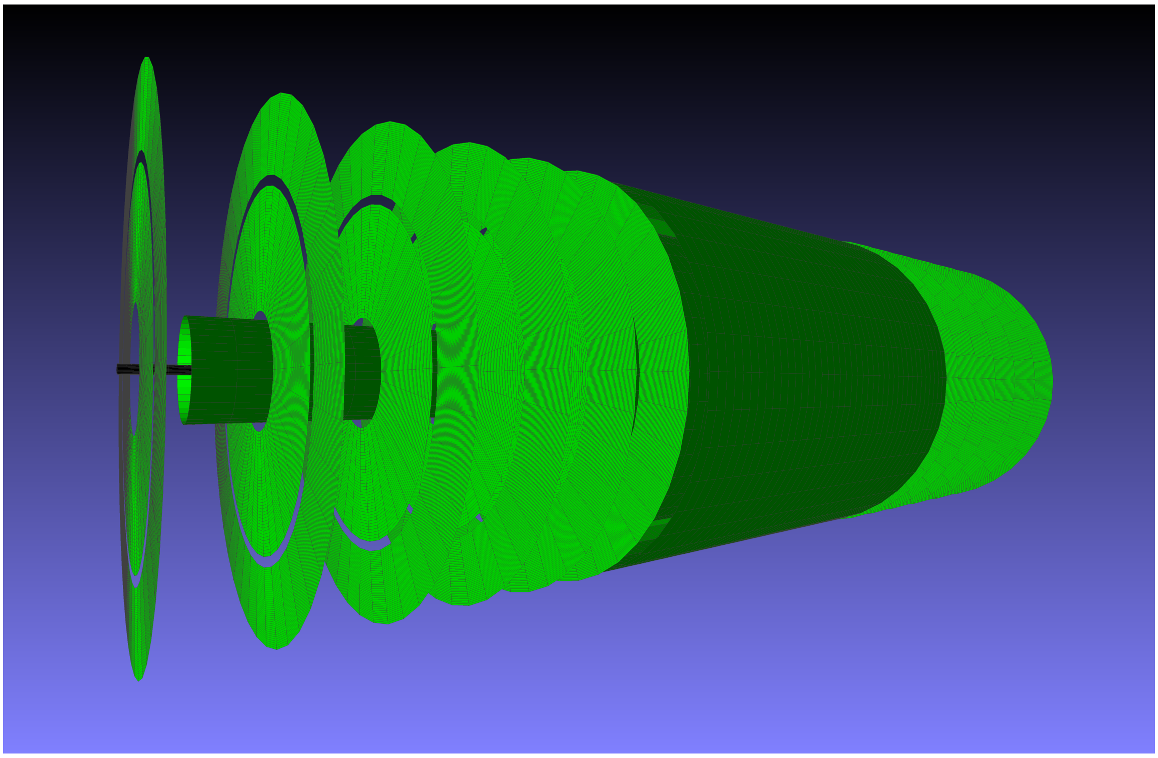


Figure 3: Initial output with a quadrilateral per bin and colour reflecting the presence of a material (black = no material, green = material present)

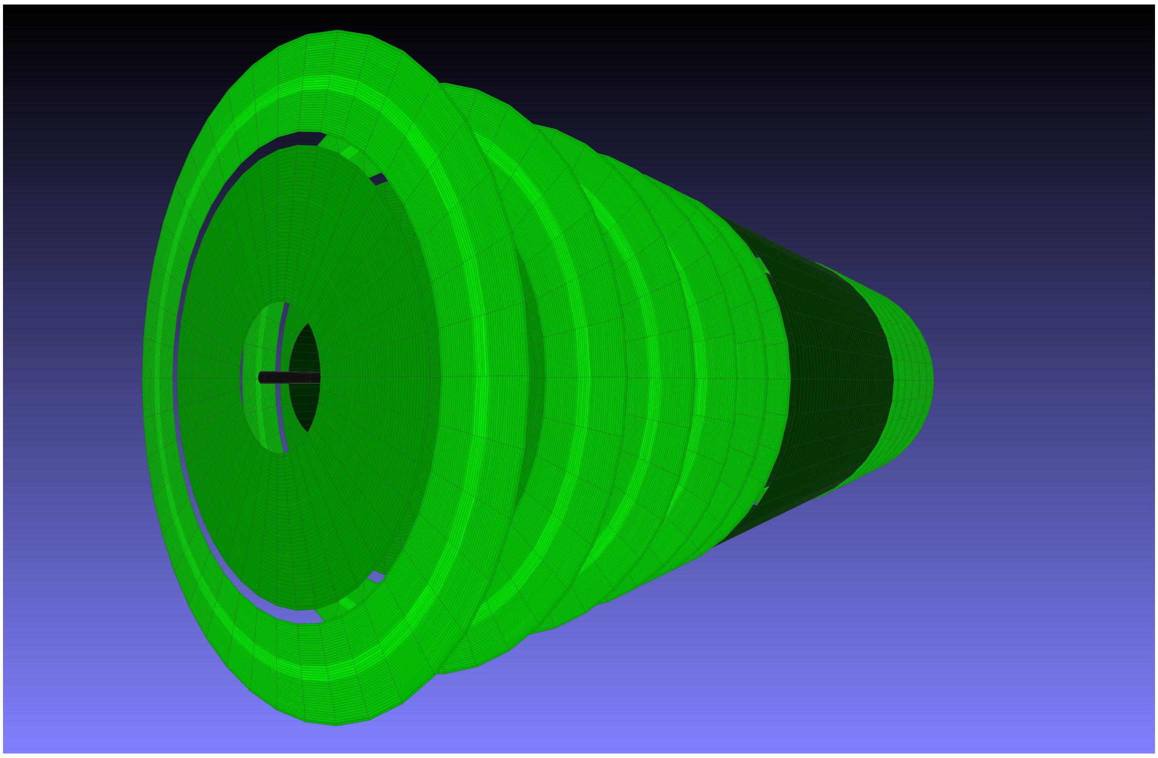
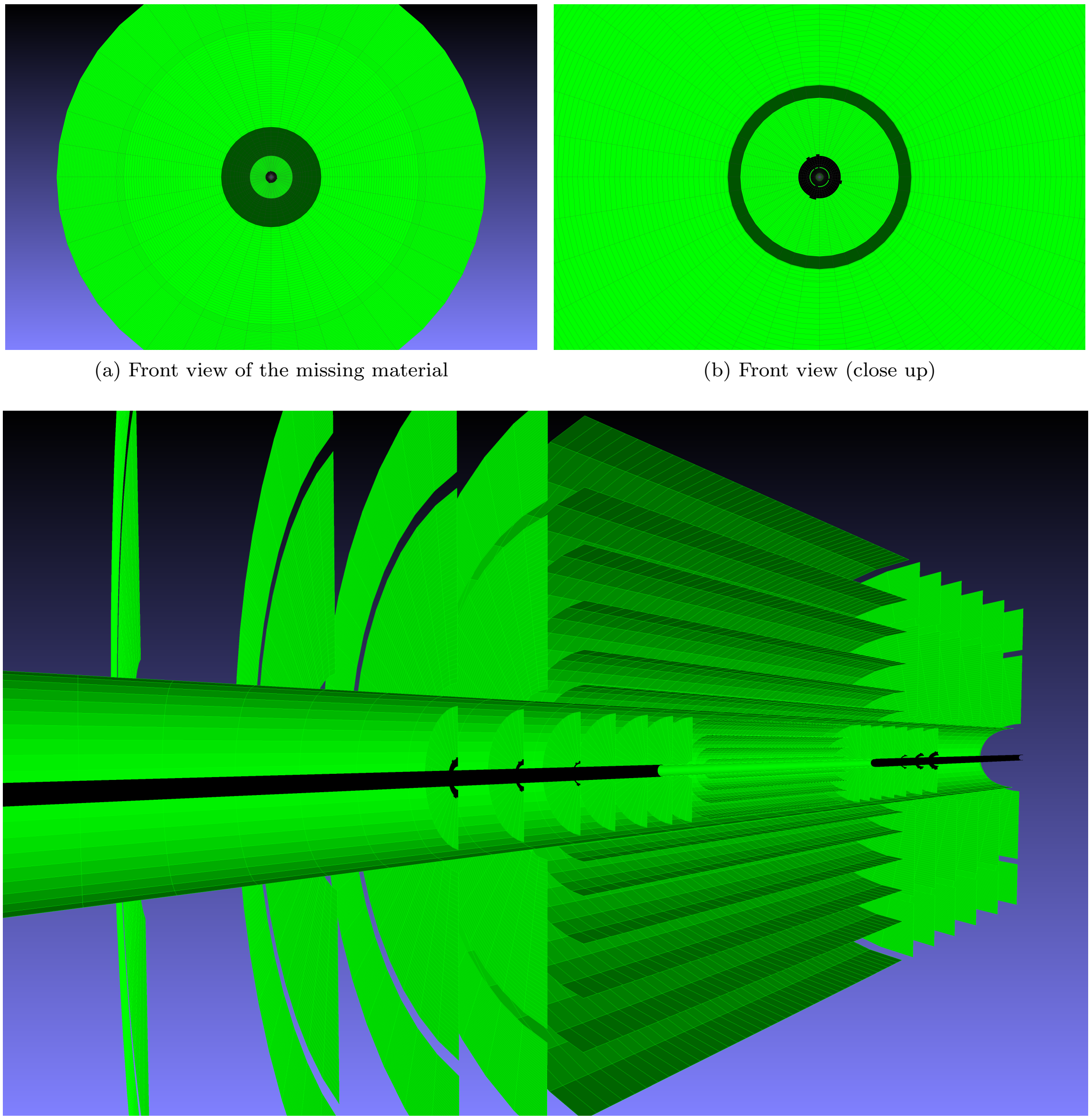


Figure 4: Final output with colour indicating material X_0 on each bin



(a) Front view of the missing material

(b) Front view (close up)

(c) Lateral view of the missing material (sliced)

Figure 5: Finding the bins with missing material