Simulating a Particle Tracker

Collaborative Software Development

Group F

Nils Hellerhoff
Benedikt Poggel
Julian Titze
Akshath Wikramanayake

Contents

Part 1

- The reconstruction algorithm
- Final results and testing
- Implementation of the updated Requirements

The Reconstruction Algorithm

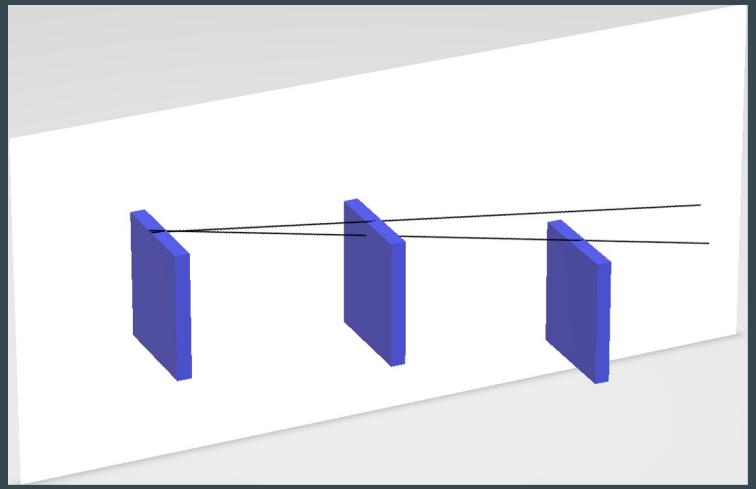
Uses pixel data to calculate four extremum gradients

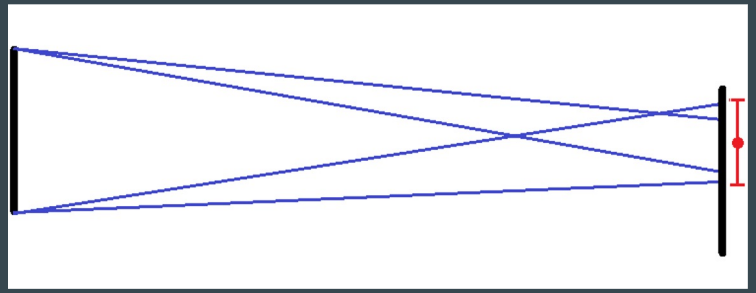
- Top-to-top
- Top-to-bottom
- Bottom-to-top
- Bottom-to-bottom

Successively narrows down gradients with each hit

Implement on the yz-plane and xz-plane to get x, y coordinates of approximate hit locations

Use singular value decomposition to obtain the reconstructed particle track





Final Results and Testing

Compare true hit location with intersection points between reconstructed vector and detectors

Errors are considerably smaller than the side length of a pixel

Errors at each detector appear to be uniform

Small standard deviation

Distance between true hit and reconstructed hit (μm) (n = 1000)					
	Detector 1	Detector 2	Detector 3	Detector 4	Detector 5
Average	39.330	23.095	21.601	35.120	52.785
Standard dev	14.985	10.838	10.646	18.039	28.307

```
Created a particle with angles phi=-83.24692860022121 and theta=5.399984474838126

Detector hits:
[1033.  718.]
[1038.  671.]
[1044.  624.]
[1050.  577.]
[1055.  530.]

Particle vector is:
[ 0.01106622 -0.09345514  0.99556199]


Reconstructed direction vector is:
[ 0.01067732 -0.09358211  0.99555431]

passing through point:
[ 0.44291667 -3.755      40.        ]


The true hit locations are:
[[ 0.33346666 -2.81615221 30.        ]
 [ 0.38904443 -3.28551091 35.        ]
 [ 0.44462221 -3.75486961 40.        ]
 [ 0.50019999 -4.22422831 45.        ]
 [ 0.55577776 -4.69358701 50.        ]]

The reconstructed hit locations are:
[[ 0.33566667 -2.815      30.        ]
 [ 0.38929167 -3.285      35.        ]
 [ 0.44291667 -3.755      40.        ]
 [ 0.49654167 -4.225      45.        ]
 [ 0.55016667 -4.695      50.        ]]

The errors in the hits at each detector are:
[0.00248347 0.00056758 0.00171052 0.00373882 0.00578627]
```

Updated Requirements

Source properties (uniform vs. normal distribution) can be chosen when calling the particle class

The specific covariance matrix can be chosen

Tracks from the source with a normal distribution do not always hit all the detectors

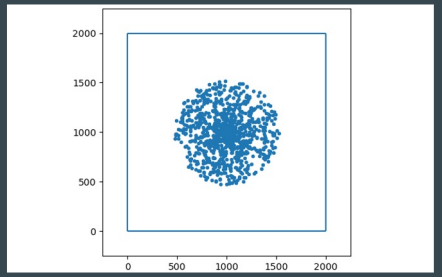
Exception raised when a track does not intersect all detectors

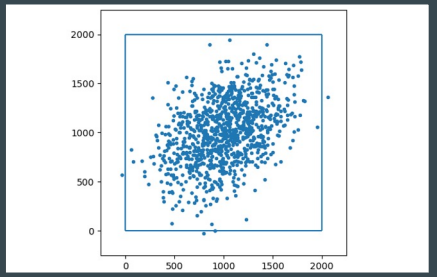
Further analysis should be done to determine whether tracks with less pixel data are precise enough to be useful

Different particle source properties

We now learned that the property of the particle source are different than expected: The particle flux is not uniform. When measured in a plane perpendicular to the nominal flight direction in a distance of 10 cm, we get a normal distribution centered at the nominal flight axis with the covariance matrix

$$C = \begin{pmatrix} 1.2 & 0.5 \\ 0.5 & 1.2 \end{pmatrix} \text{cm}^2$$






```
Created a particle with angles phi=103.98322703795202 and theta=16.582430803664845

Detector hits:
[ 784. 1866.]
Traceback (most recent call last):
  File "main.py", line 45, in <module>
    print(d.detector_hit_no_event(p))
  File "C:\Users\aksha\Downloads\groupf-Reconstruction-output\groupf-Reconstruction-output\detector.py", line 51, in detector_hit_no_e
    pixel_id = self.pixel_from_pos(hit)
  File "C:\Users\aksha\Downloads\groupf-Reconstruction-output\groupf-Reconstruction-output\detector.py", line 32, in pixel_from_pos
    raise Exception("Position outside of sensor")
Exception: Position outside of sensor
```