Overview of the Integration of the General Broken Lines Algorithm within EUTelescope

Beam Telescopes and Test Beams Workshop
2015

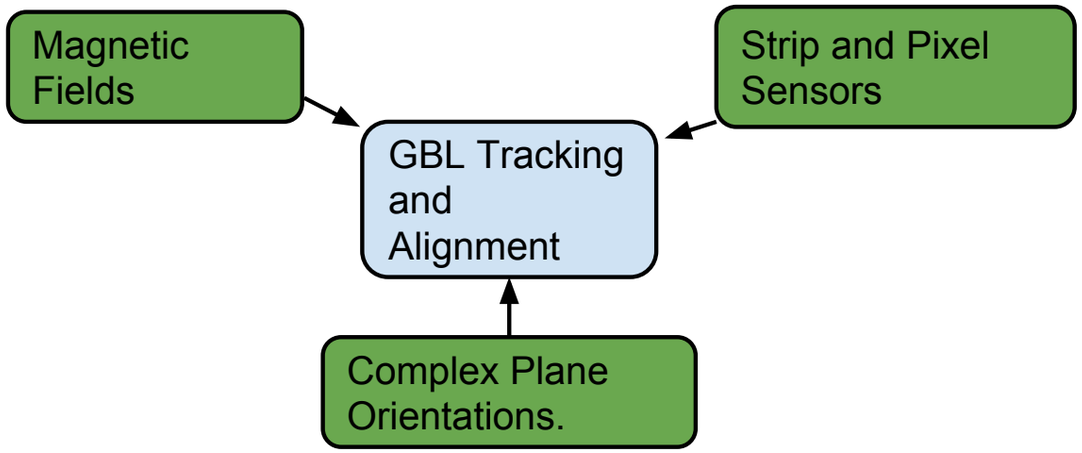
Alexander Morton

Introduction

- General Broken Lines (GBL) algorithm written by Claus Kleinwort
    - Thanks to all those involved
- Daniel Pitzl, advocate for its use with telescope track reconstruction
    - Created first working example
- Denys Lontkovskyi began implementation within EUTelescope framework
- Work continued on by myself and Igor Rubinskiy
- All results shown today can be run at jobsub/examples/GBL
- Work goes beyond updating tracking software but also extends to alignment procedure
- Past workshops can be viewed to further understanding
    - https://indico.desy.de/contributionDisplay.py?sessionId=4&contribId=25&confId=10296

Requirements from Tracking and Alignment

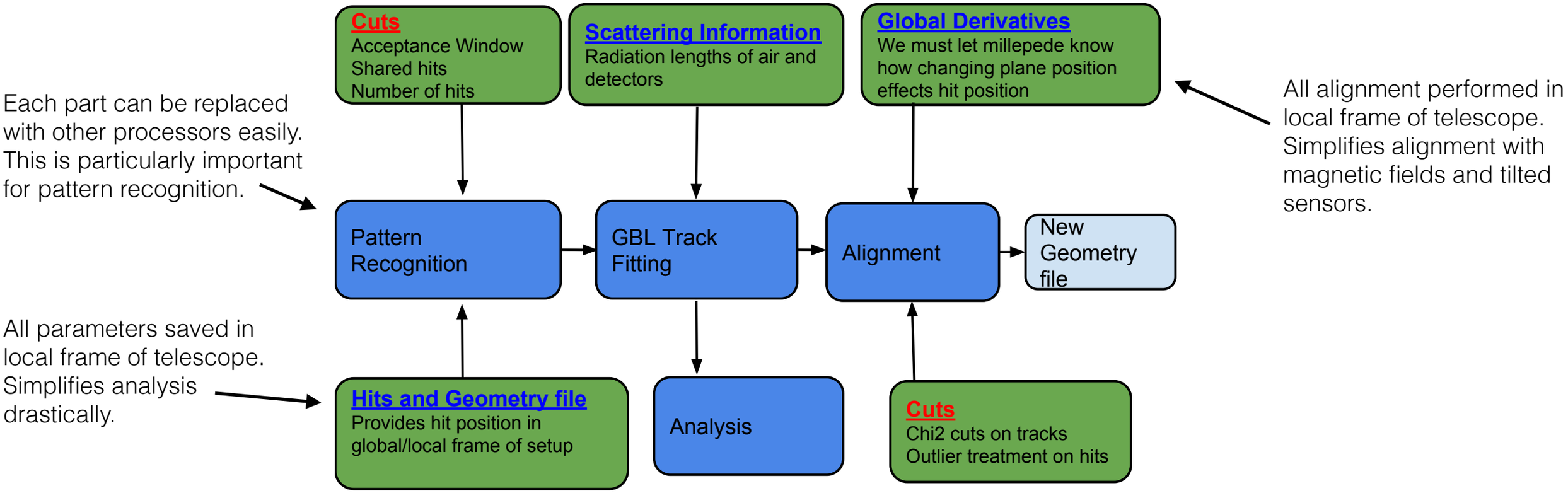
- Many interesting studies require a versatile track fitter and alignment procedure



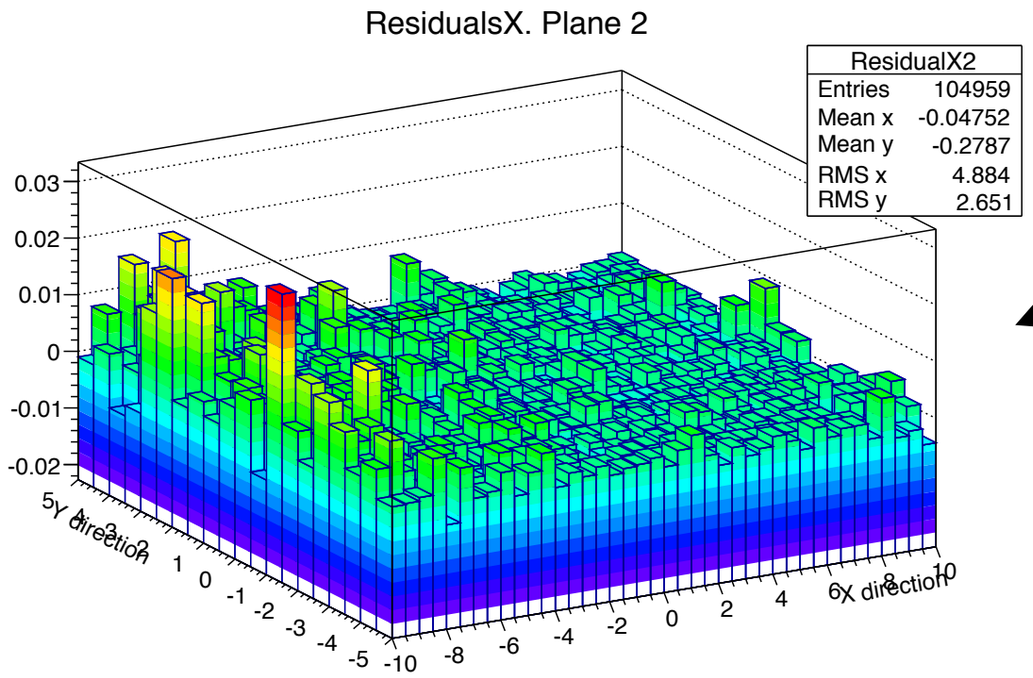
- We want to design software which is as plug and play as possible
- Allows a single analysis code to be written once for a multiple setups and DUTs
- Use past analyses as an example for future work
- To achieve all these goals we use a combination of the General Broken Lines (GBL) for track fitting and millepede 2 to align the mimosa and DUT planes

Tracking and Alignment Overview

- Pattern recognition based on predicted kinematics of beam particles provide hits to fit track
- The GBL algorithm needs many input parameters to have the optimum track fit



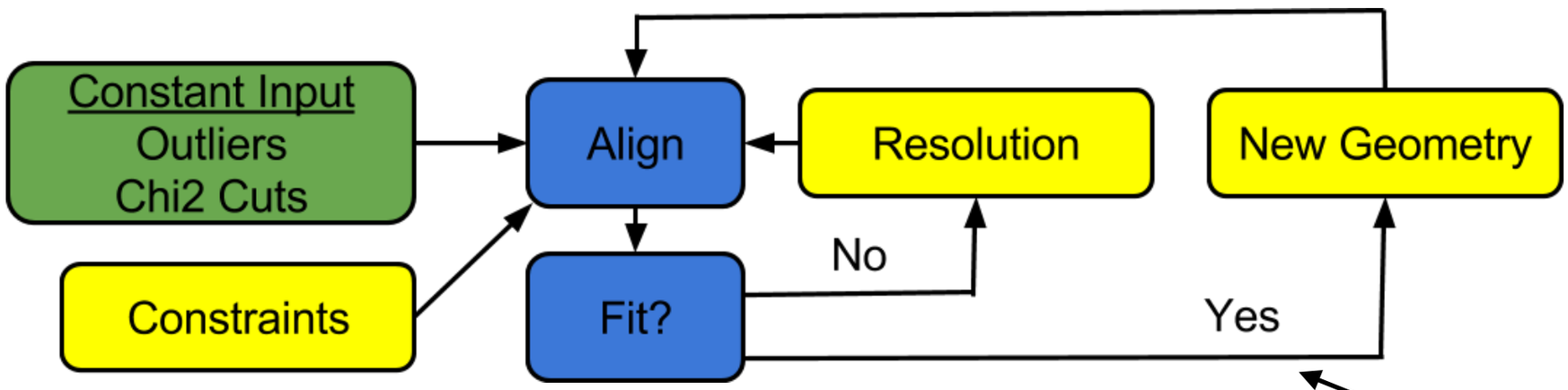
- Analysis code will work for any DUT using tracks produced by GBL processor



We can look at how the residual of tracks varies with position on sensor. This example is one of the mimosa sensors with residual in X direction.

Iterative Alignment

- To find the correct alignment parameters can be a difficult process
  - Multiple steps needed to constrain different degrees of freedom
  - Estimated resolution of detectors depends on initial misalignment
  - Noisy tracks must be removed during process
  - Outlier treatment essential to deal with single noisy planes
- Millepede 2 combined with an iterative alignment procedure is used to automate the process
  - Outlier treatment and noisy tracks dealt within Millepede 2
  - Different degrees of freedom freed with each step
  - Each step outputs results and geometry file for easy debugging

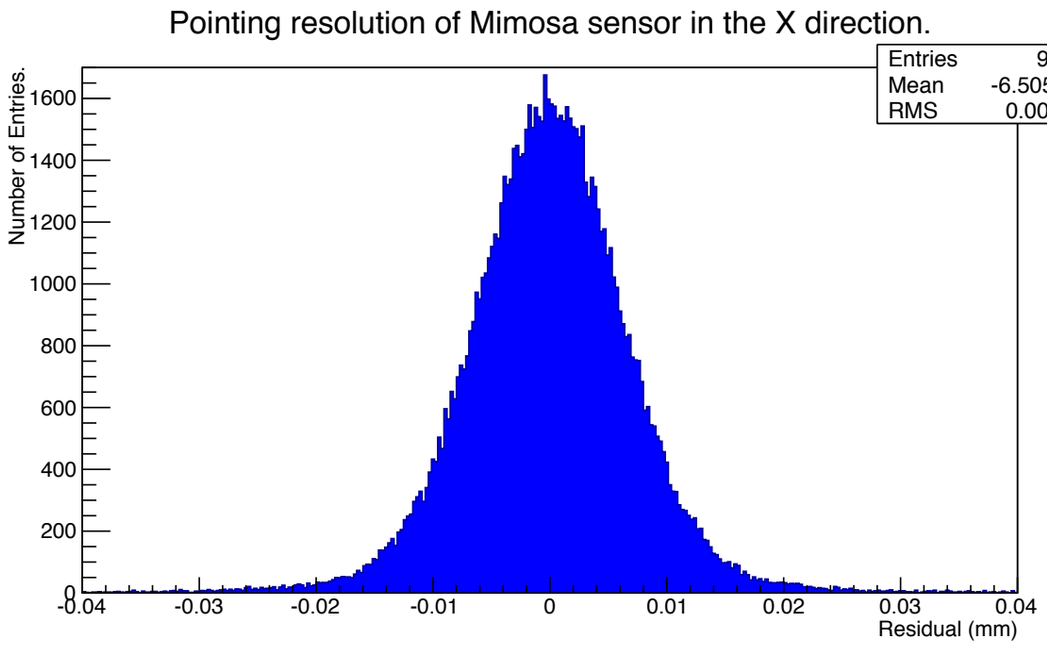


The Iterative Alignment Procedure in a nutshell.
Yellow input indicates an initial value which will change upon each iteration.

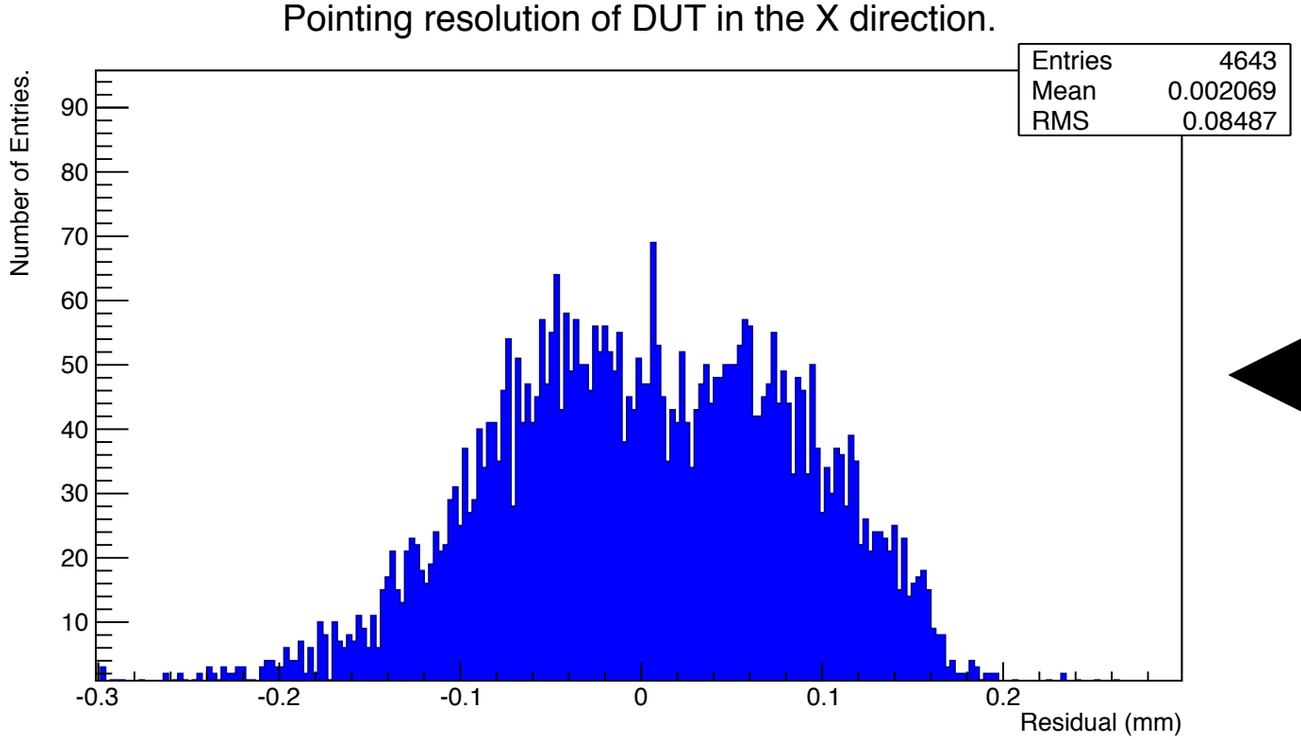
Each step produces histograms and gear file.
This is very useful for debugging.

Pointing Resolutions

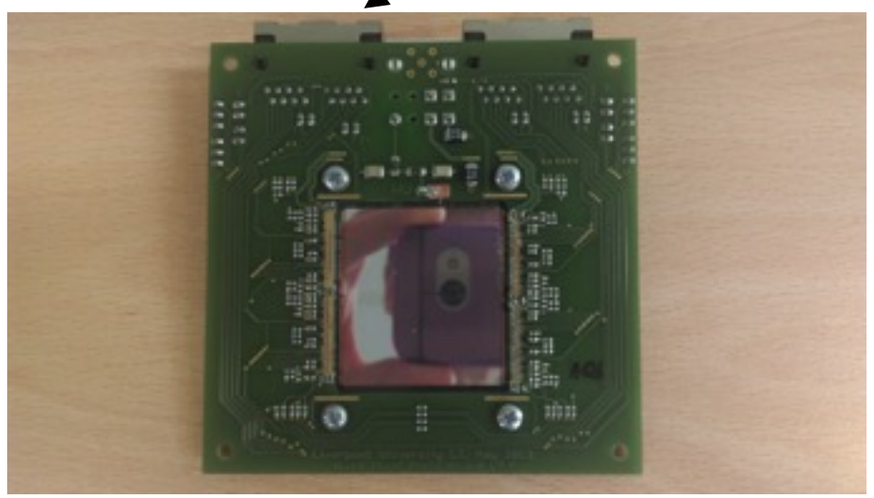
- After alignment procedure we can produce tracks
- Pointing resolution useful to have a first look at the accuracy of fit
  - Sometimes more illuminating than chi2 distribution



Sensor 2 (Centre).
Residual slightly larger than expected for geometry setup.
See Igor Rubinskiy's talk to see expected resolution.
We use a large distance between planes in z direction.
Model for scattering wrong?
Incomplete alignment of weak modes?

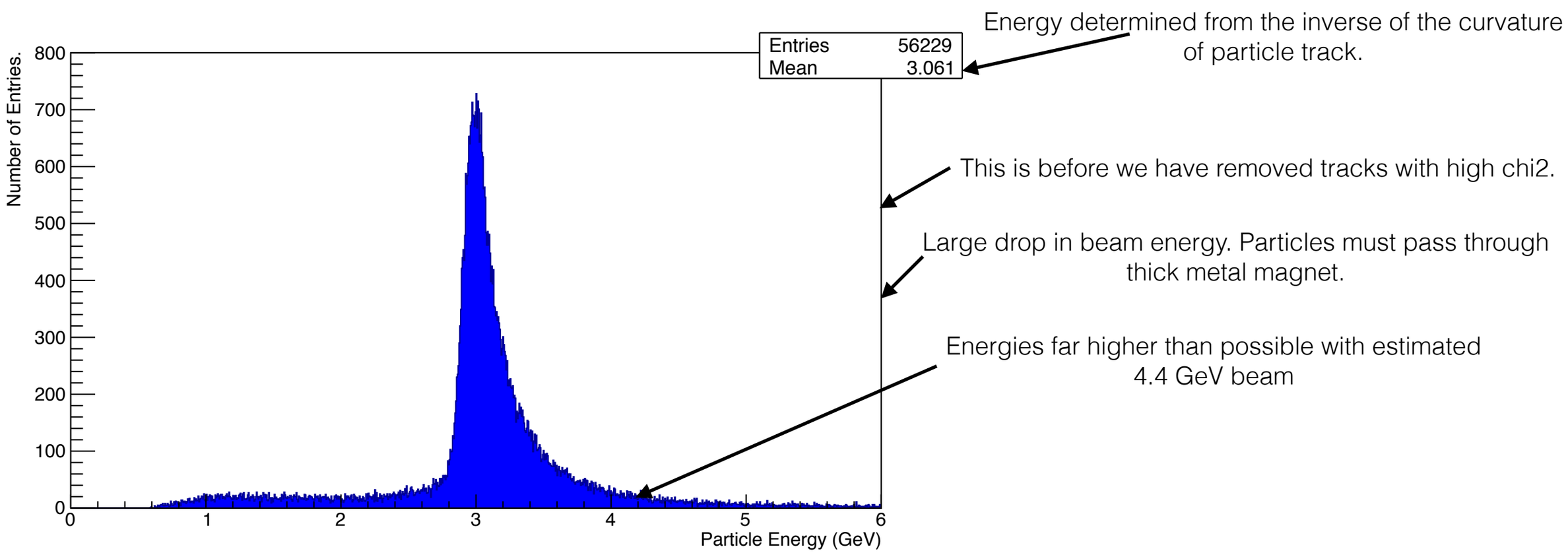


DUT 250x50 micron
Data taken by Glasgow group
See Kenny Wraight's talk.
X direction is 250 micron width



Analyses
Magnetic Fields and Particle Energies

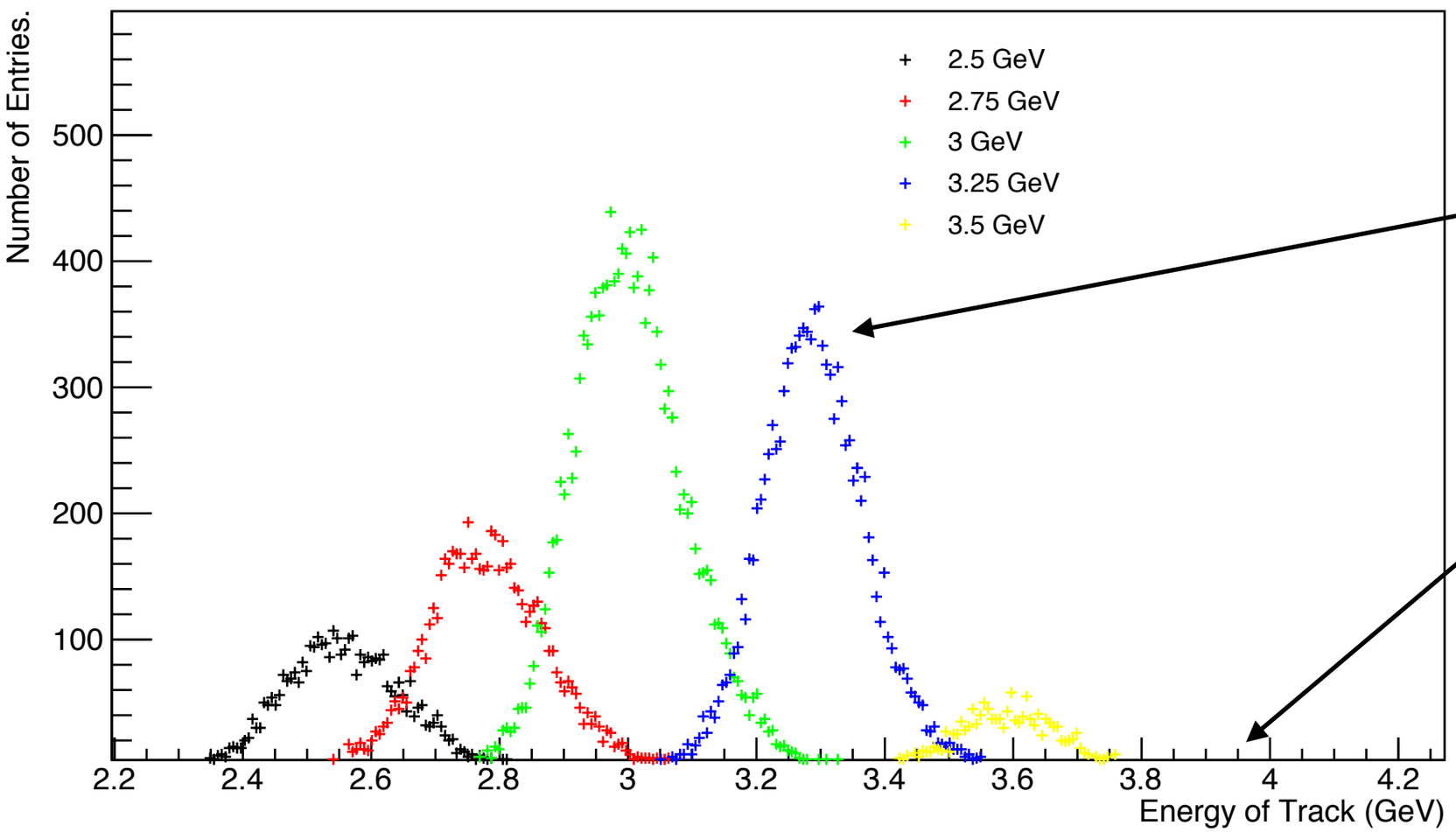
- After first initial checks the real fun begins
  - Analyses related to peoples questions and interests
- We fit for the particle energy using GBL directly assuming a specific magnetic field strength
- Initial magnetic field strength assumed to be 1T
- Energy of beam will have an obvious deviation from the initial beam estimation
  - Natural spectrum of beam energy
  - Initial scattering before the beam reaches the detector
- Pattern recognition has no cut on window size to allow all possible track energies to pass



Magnetic Fields and Particle Energies

- Pattern recognition needs some initial guess to look for hits to form tracks from
  - This will bias found tracks to a particular energy
  - Inaccurate pattern recognition will pass noisy tracks to fit
- We want to keep the bias of tracks and noise to a minimum
  - Open search window to full
  - Perform Chi2 cut to remove noise tracks
- After we cut tracks with chi2/ndf > 5 we see more realistic results
  - However the results is still preliminary.

Energy Distribution with Different Initial Beam Energies

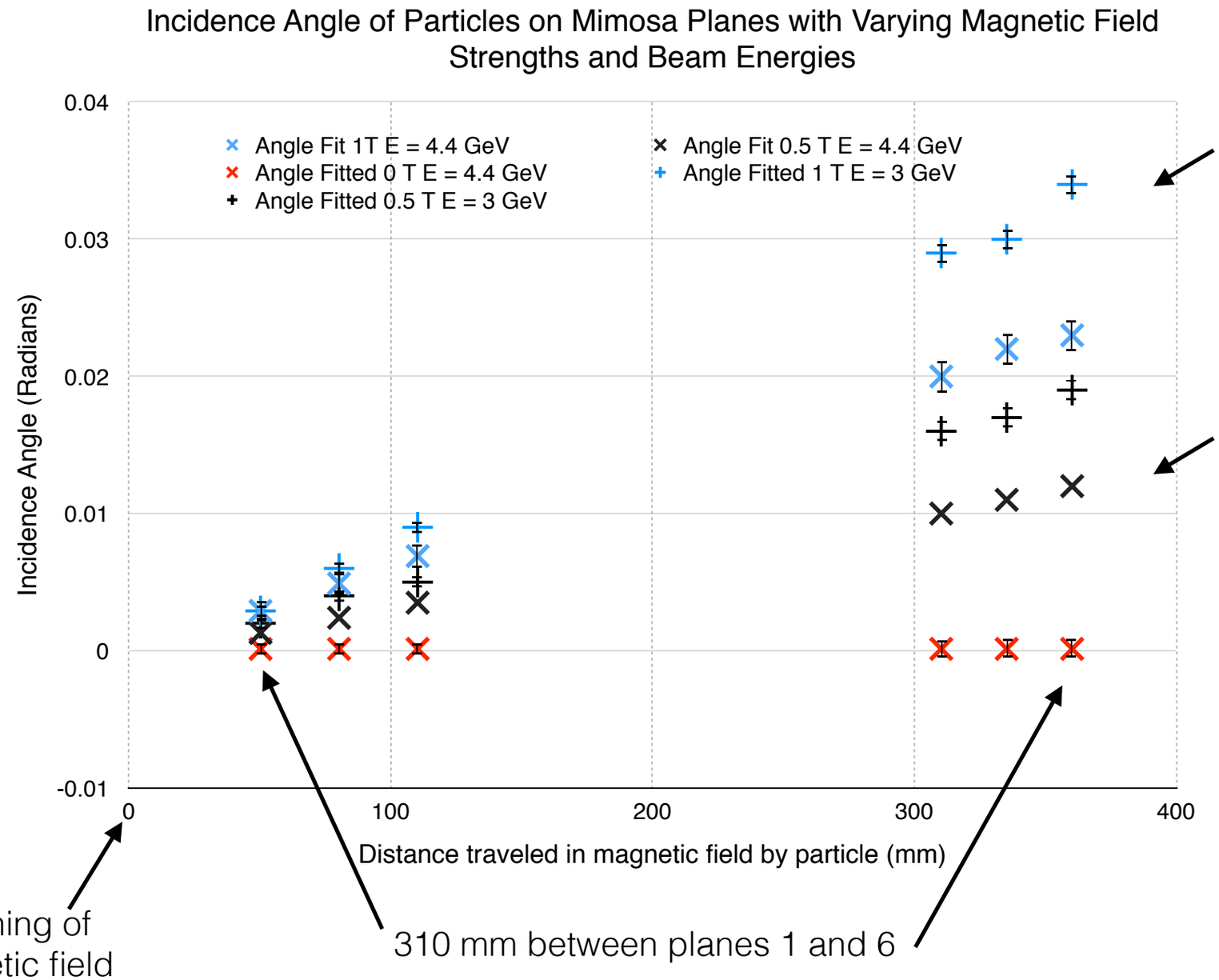


Each initial input energy to pattern recognition produces a different peak. Must check if these are different tracks!

No tracks now found at higher or lower energies.

Track Parameters (Incidence Angles)

- Track parameters can be determined at any point in space specified by the geometry file
  - Positions (x,y)
  - Incidence angles (x,y)
  - Curvature (Given by beam energy as before)
- We can look to see how magnetic fields affect the incidence angle on the mimosa planes



This is useful for Eda Yildirim's work concerning Lorentz Angle Measurements (LAM).

Observe how the incidence angle is effected by varying magnetic field and beam energy for the same geometric setup.

Beginning of magnetic field

310 mm between planes 1 and 6

How to Run in a Nutshell

- All the analyses discussed here can be found within the EUTelescope jobsub/examples/GBL folder
  - Follow the readme file to run the example with no DUT
- The new track fitting and alignment procedure changes slightly the conversional way most people track fit and align
  - Differences due to the separation of processors

Go into this file to set variables that change as we perform the alignment. All other variables within config file as usual.

This file simply runs pattern recognition and GBLTrack for you.

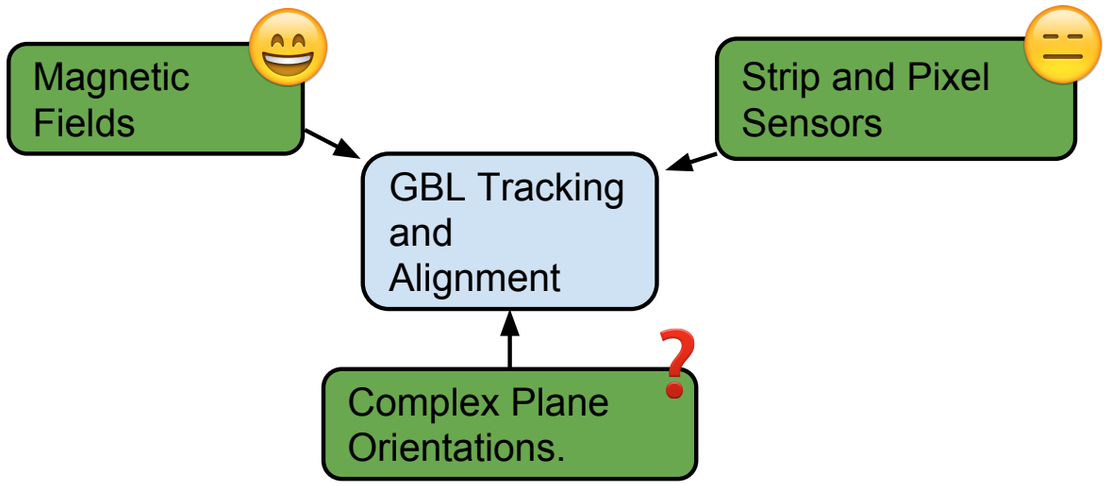
```
[amorton@ppepc181 noDUTExample]$ ls
config  geometry  noDUTIterativeAlignment.sh  output  patRecAndTrackFitUsingConfigurationFile.sh  runlist  steering
[amorton@ppepc181 noDUTExample]$
```

Found within each example in GBL

- Strip sensors and complex pixel geometries a work in progress so examples not complete

Conclusion

- The GBL track fitting algorithm has been successfully integrated with EUTelescope
- New alignment procedure based on GBL and Millepede2 also complete
- Working examples can be found and you can add your analysis to these examples
- Developments still ongoing
  - Strip sensors
  - Complex pixel geometries
- Any further developments to code to suit your needs welcome
- I am happy to assist any new analysis work with problems and how we can include this into the EUTelescope framework
- During the tutorial I will be here to answer any questions



Thank You!

“These are my backup slides”

–Alexander Morton