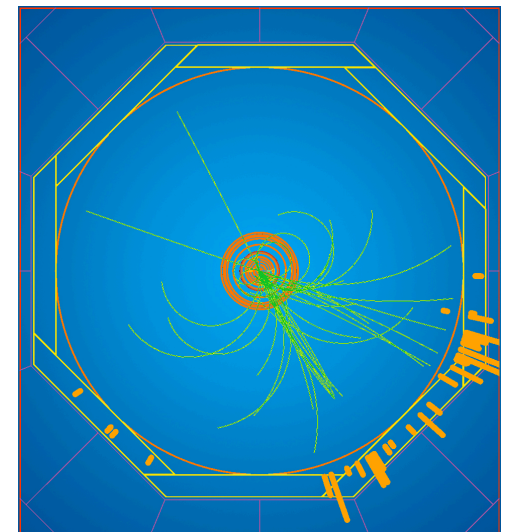


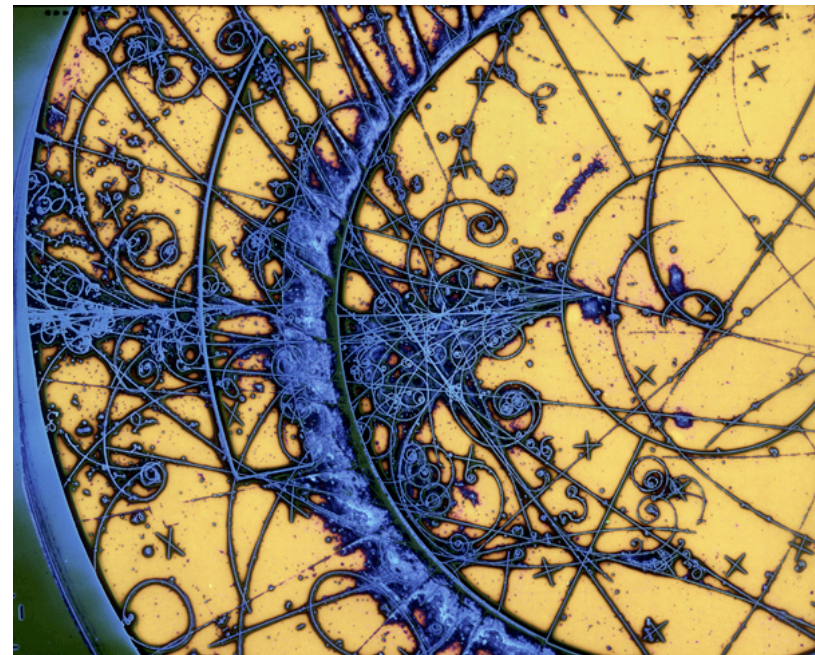
Status of Tracking Task in WP2

Steve Aplin
DESY

AIDA 1st Annual Meeting at DESY
28th March 2012



- Reminder of Requirements
- Status of Current Activities
- Outlook



Current Tracking Activities in LC Studies

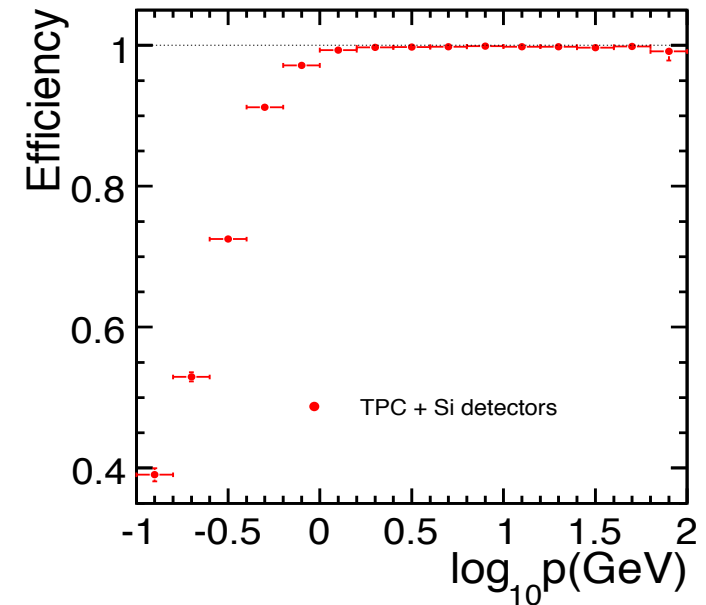
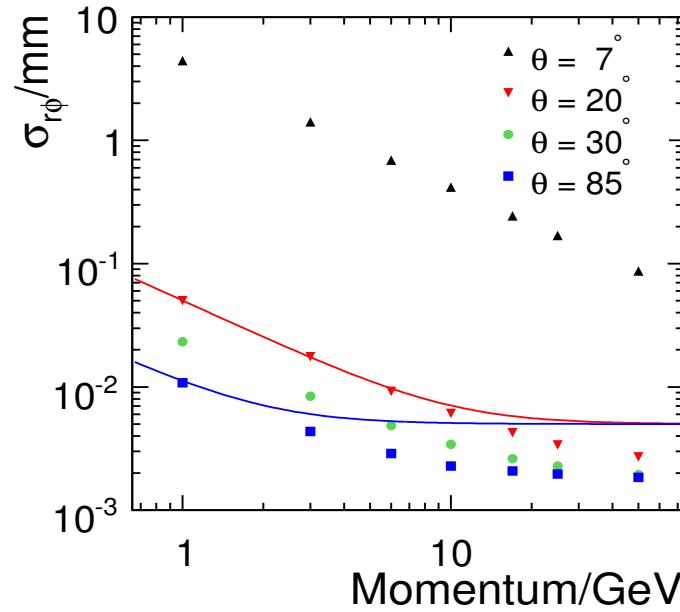
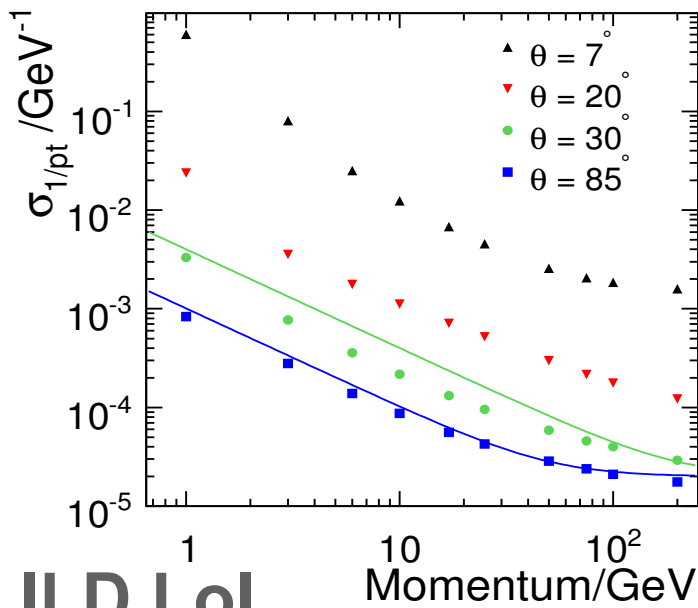
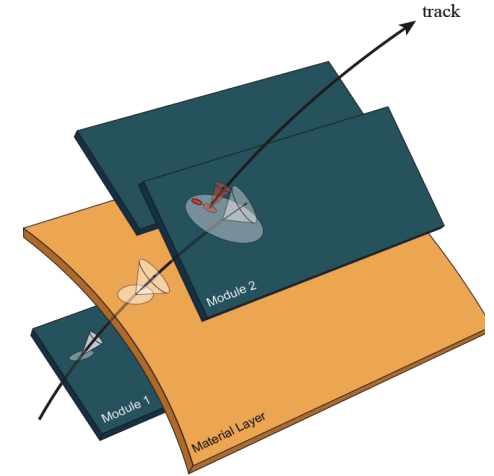
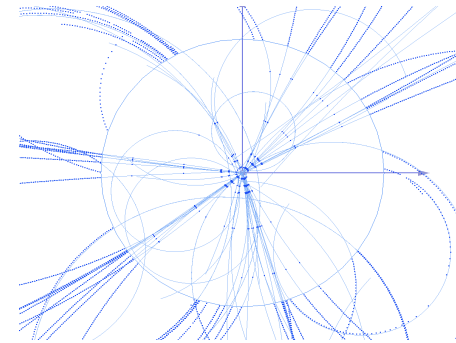
- The next stage in ILC activities for the **Detector Baseline Document**, calls for dedicated studies into the impact of accelerator induced **background**, as well as **more accurate detector descriptions**.
- For this we have had to replace the tracking software previously used for our studies.

ILD Track Reconstruction

Full pattern recognition

Stand-alone track finding in both Inner Silicon Trackers and TPC

Kalman Filter Track Fitting



Tracking Task in Aida WP2

- Provide a toolkit which includes the necessary building blocks needed to put together a realistic track reconstruction.
- Software design tailored to R&D studies for the planning stage of detector design.
- Joint activity between DESY and OEAW.
 - DESY: F. Gaede, S. Aplin
 - OEAW: W. Mitarof, R. Glattauer

Milestones and Deliverables

Deliverables

- **D2.4, M12: Software design for tracking toolkit.**
- D2.8, M38: Software toolkit with tracking algorithms.

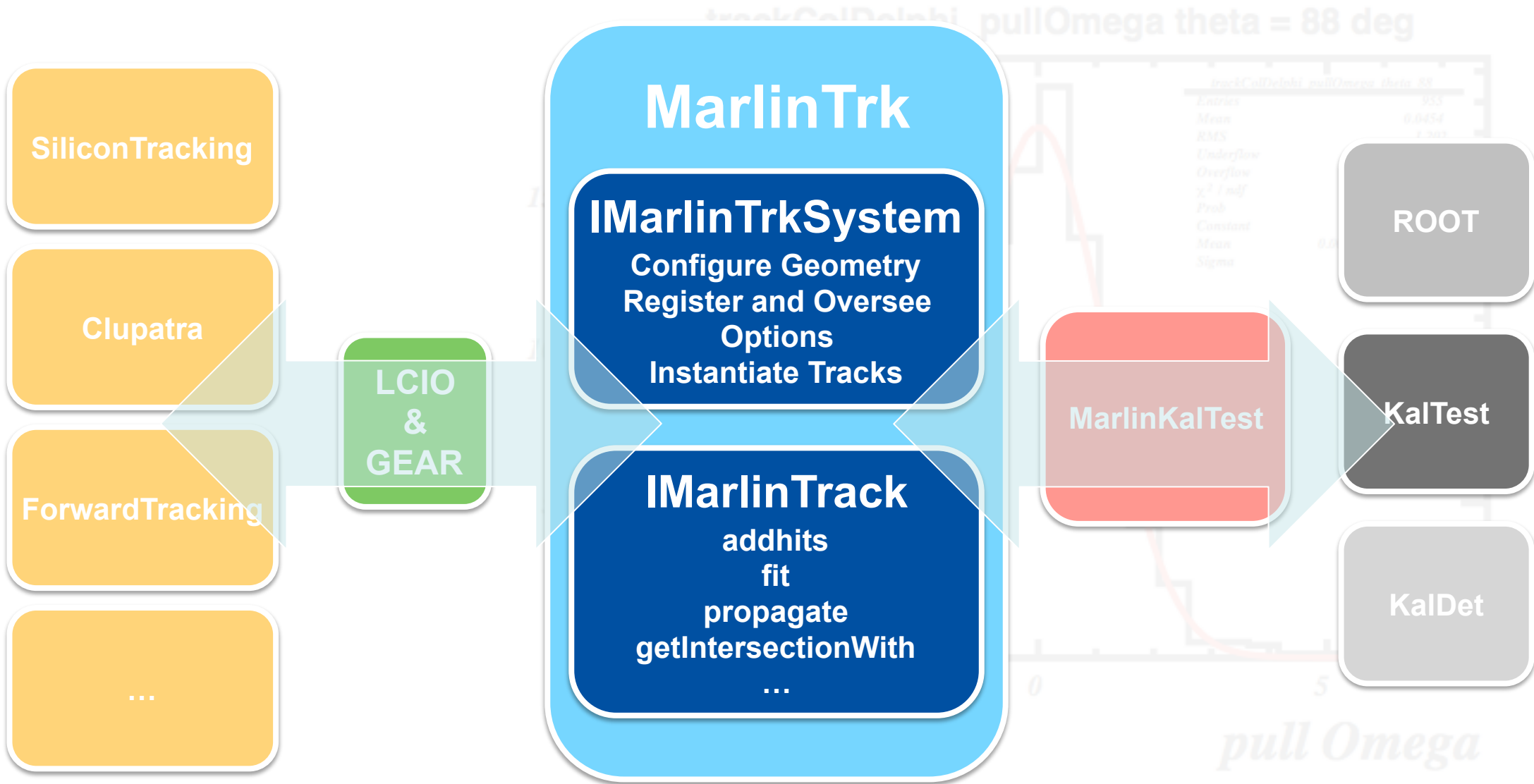
Milestones

- **MS11, M18: Running prototype of tracking toolkit including some algorithms. Application to ILD-TPC simulation.**
- MS14, M44: Integration of tracking toolkit into LC software framework. Validation of physics performance.

Three Main Areas of Activity

- Toolkit Framework – Interface for track fitting
DESY and OEAW
- Nearest-Neighbor Clustering for TPC Pattern Recognition
DESY
- Forward Track Reconstruction using Cellular Automata
OEAW

MarlinTrk KalTest Implementation



KalTest

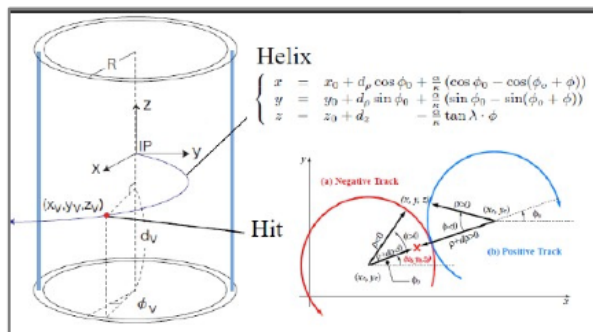
Kalman Filter fitting library (Keisuke Fuji et al)

Based on Root

Structured in sub-libraries

- geomlib -- geometry
- kallib -- Kalman filter
- kaltracklib -- Kalman tracker
- utils -- utilities

Built into one libKalTest.so



Example 2 : Tracking in HEP Experiments

$$a_k = \begin{pmatrix} d_\rho \\ \phi_0 \\ \kappa \\ d_z \\ \tan \lambda \end{pmatrix}_k$$

Helix parameter vector at (k)

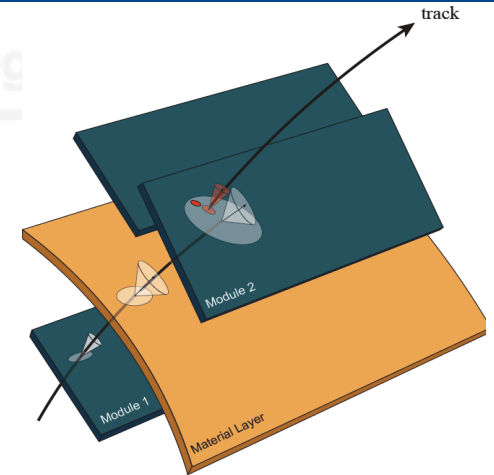
Multiple scattering between ($k-1$) and (k)

$$w_{k-1}$$

Measured hit point at (k)

$$m_k$$

random detector noise

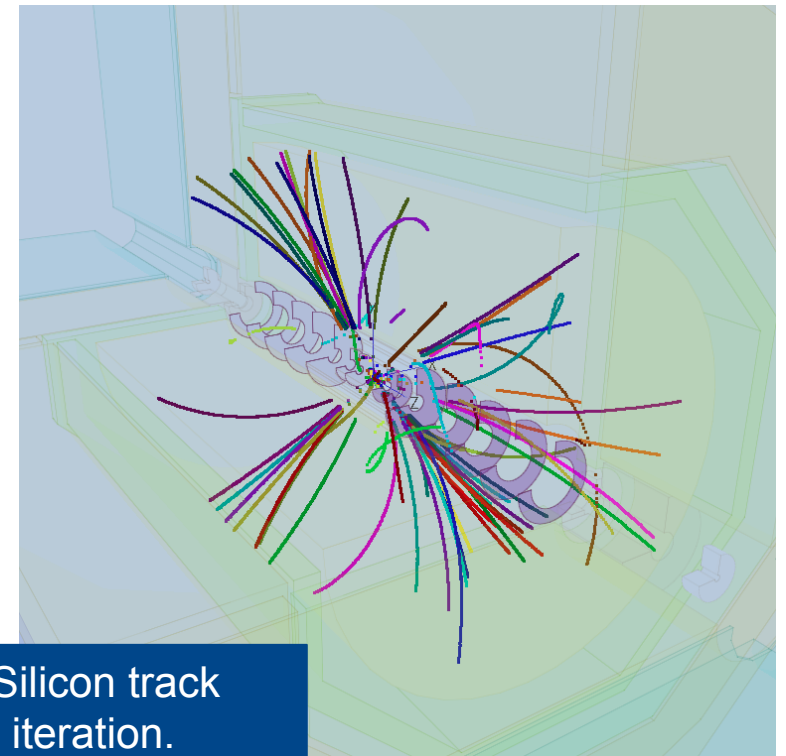
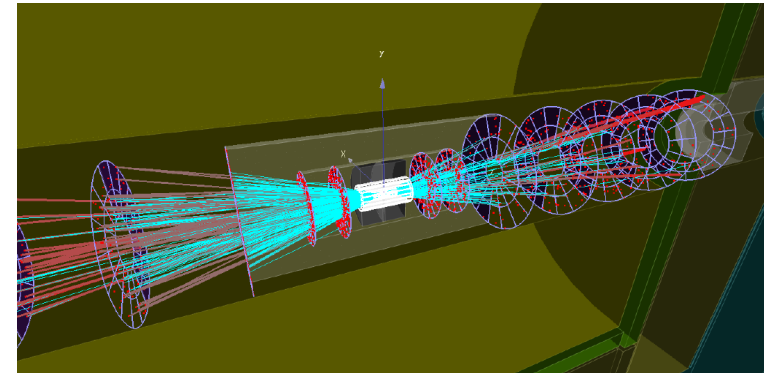
$$\epsilon_k$$


User needs to define their detector classes
(**KalDet**)

- TVMeasLayer: meas. layer, coord. to track state transformation
- TVDetector: position of measurement layers and material properties

Complete ILD Track Reconstruction

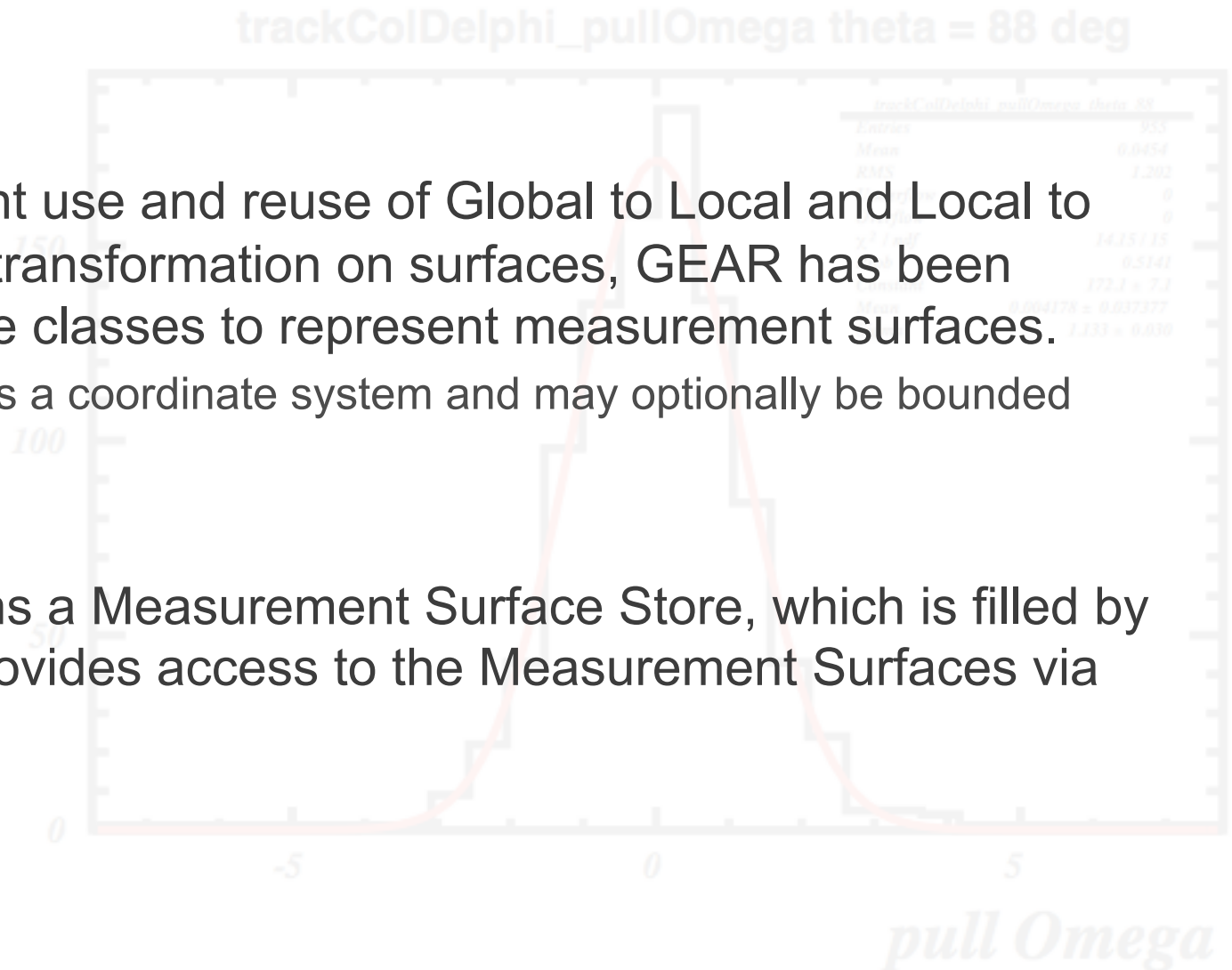
- First completed version of the ILD track reconstruction released.
- Includes new realistic sensor based silicon designs.
- Supports both pixel readout and stereo strip readout, including space-point creation from strips.
- Stand-Alone Track Reconstruction in both TPC and Silicon Trackers.



ttbar event @ 500 GeV reconstructed using stand alone TPC and Silicon track reconstruction which are then combined into full tracks in a final iteration.

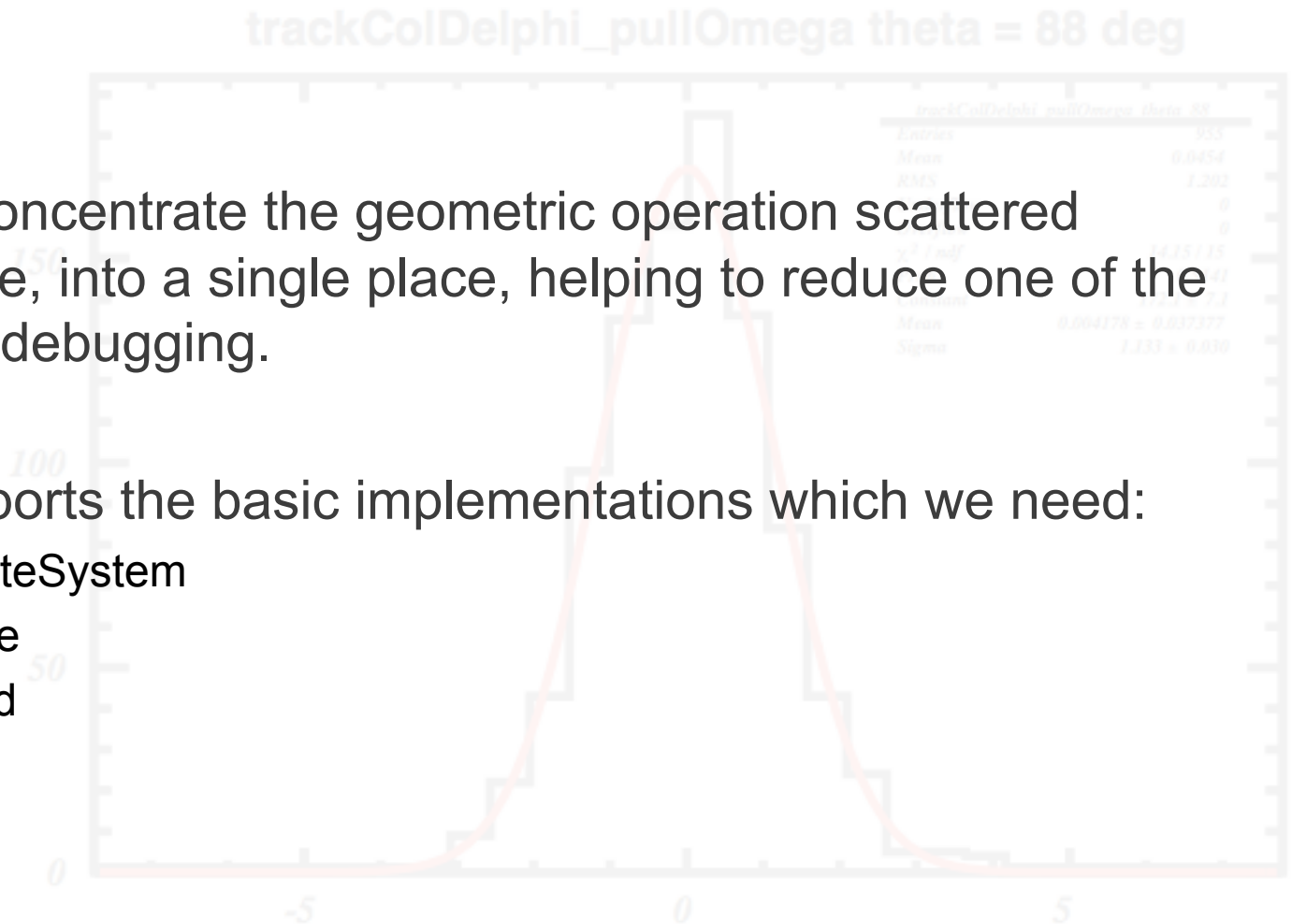
Core Software GEAR

- To facilitate efficient use and reuse of Global to Local and Local to Global coordinate transformation on surfaces, GEAR has been extended to include classes to represent measurement surfaces.
 - A surface contains a coordinate system and may optionally be bounded
- GEAR also contains a Measurement Surface Store, which is filled by client code, and provides access to the Measurement Surfaces via their ID.



Core Software GEAR

- This allows us to concentrate the geometric operation scattered throughout the code, into a single place, helping to reduce one of the biggest sources of debugging.
- Currently only supports the basic implementations which we need:
 - CartesianCoordinateSystem
 - BoundaryRectangle
 - BoundaryTrapezoid



Really, this is a temporary solution to address current needs and this should be replaced by DD4Hep

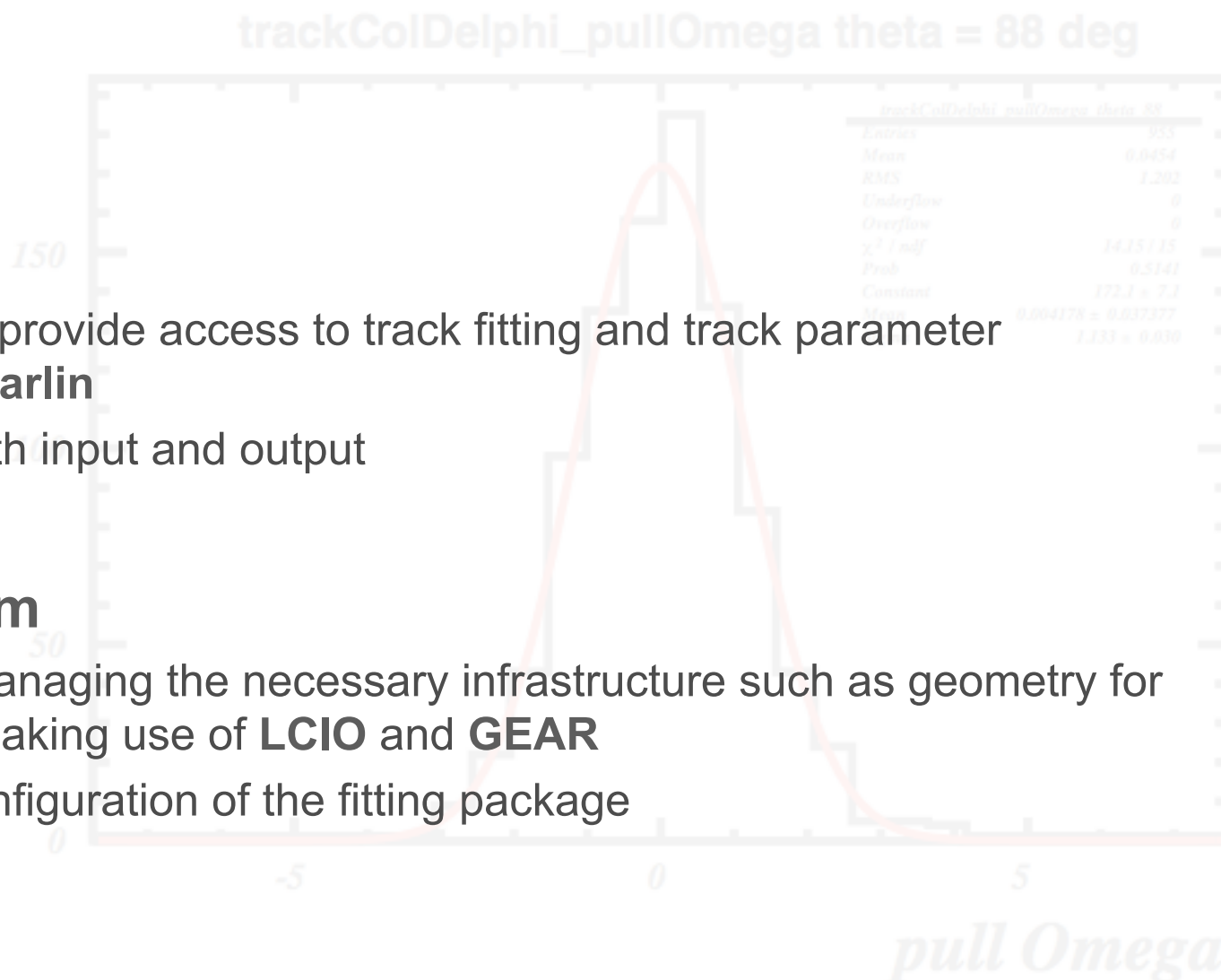
IMarlinTrack and IMarlinTrkSystem

- **IMarlinTrack**

- interface class to provide access to track fitting and track parameter manipulation in **Marlin**
- uses **LCIO** for both input and output

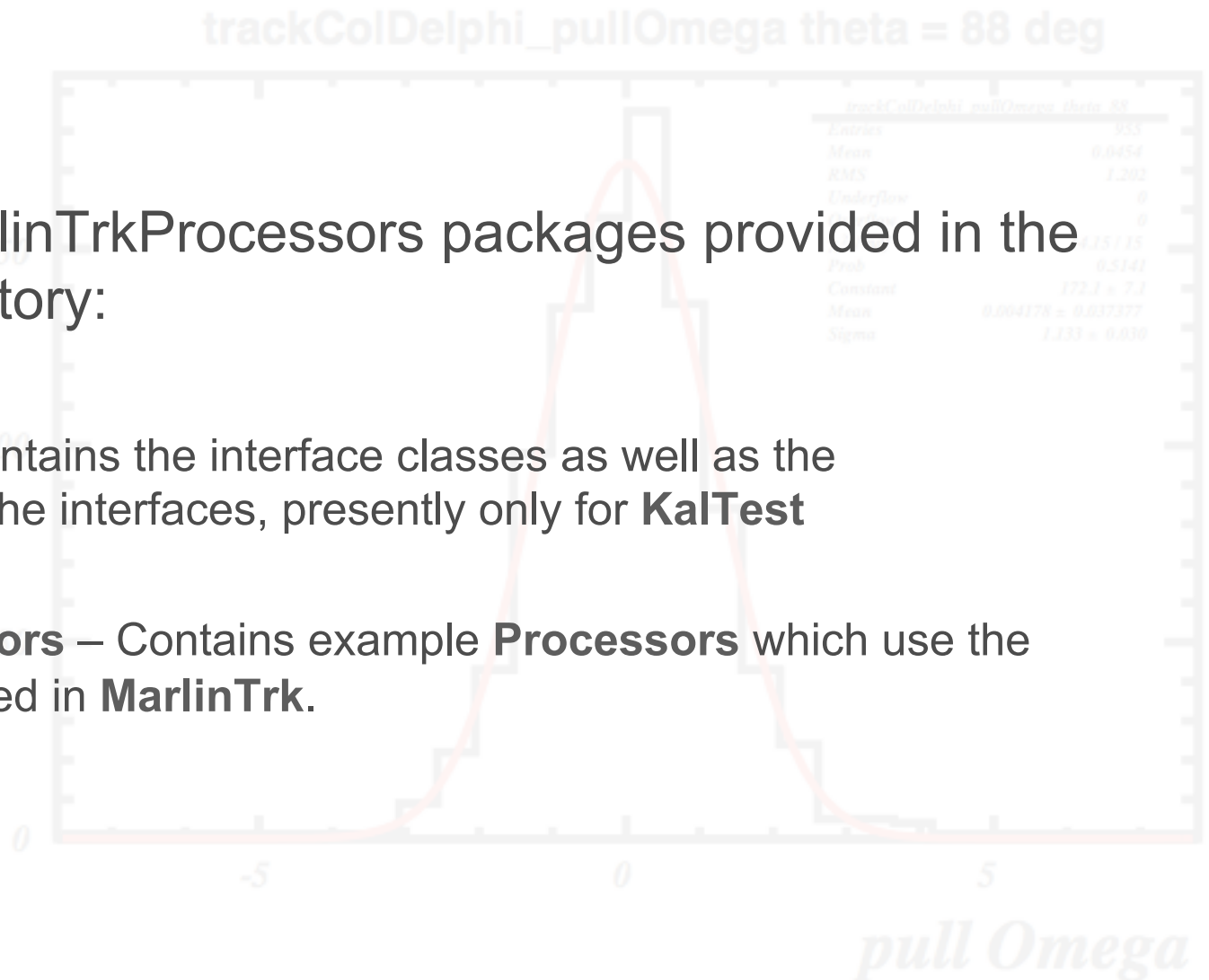
- **IMarlinTrkSystem**

- responsible for managing the necessary infrastructure such as geometry for the track fitting, making use of **LCIO** and **GEAR**
- controlling the configuration of the fitting package

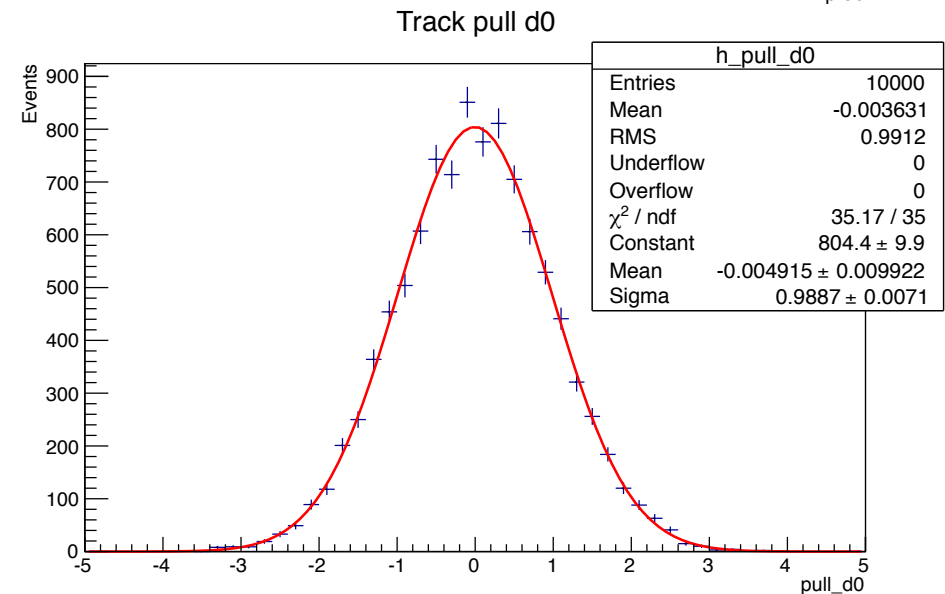
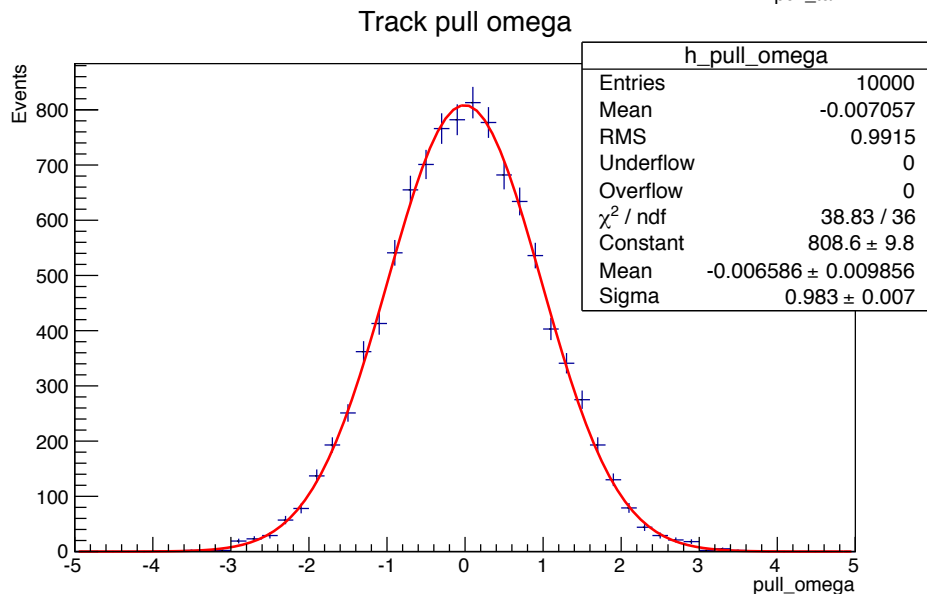
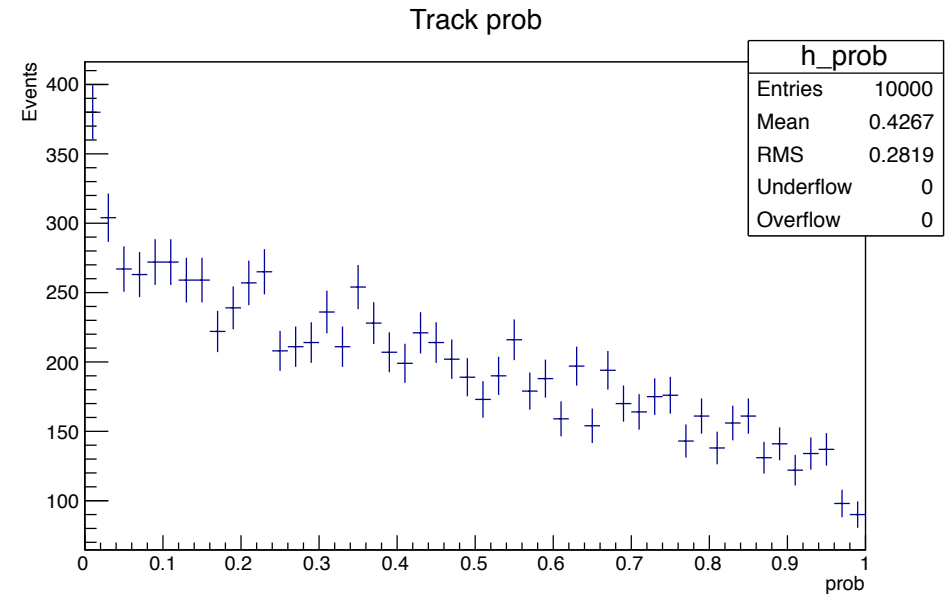
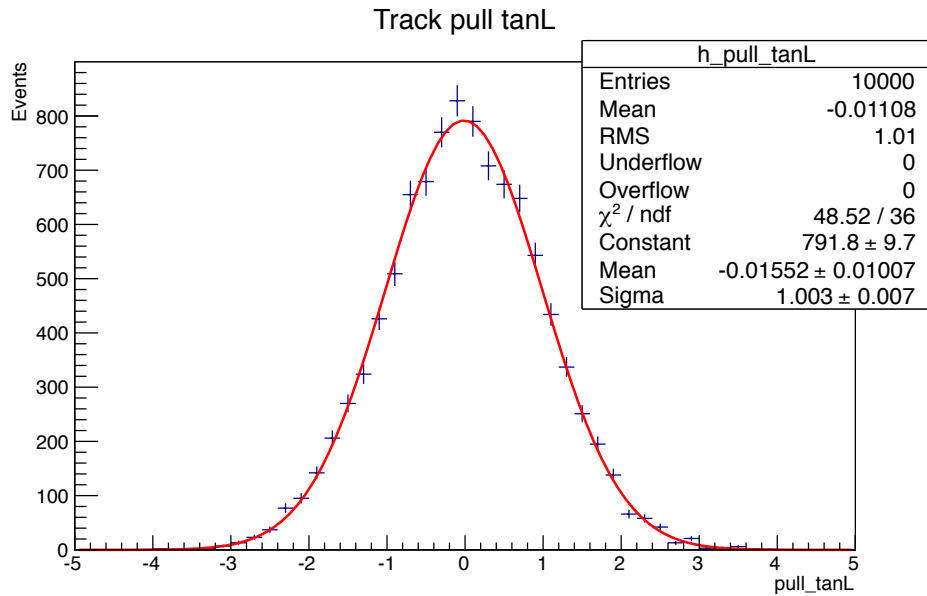


IMarlinTrack and IMarlinTrkSystem

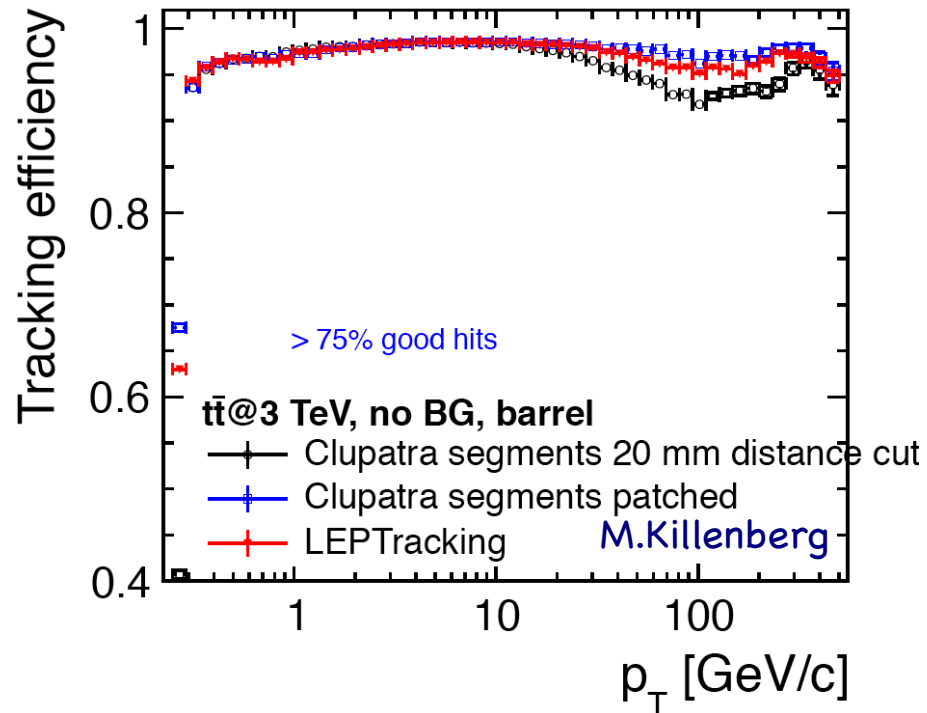
- MarlinTrk and MarlinTrkProcessors packages provided in the iLCSoft svn repository:
 - **MarlinTrk** – this contains the interface classes as well as the implementation of the interfaces, presently only for **KalTest**
 - **MarlinTrkProcessors** – Contains example **Processors** which use the functionality provided in **MarlinTrk**.



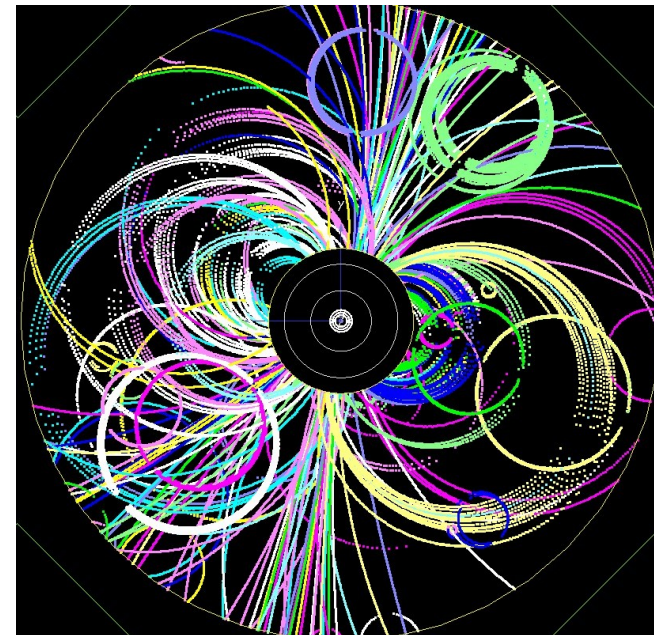
Single Track Studies using Stereo Strips



TPC Reconstruction – Clupatra @ 3TeV



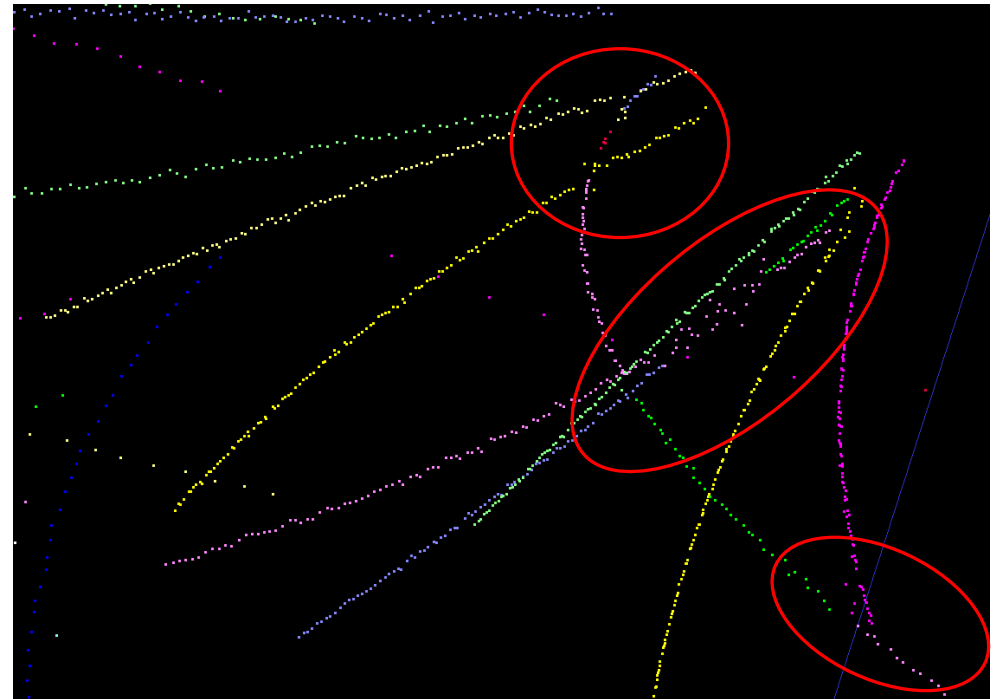
- Tracking efficiency improved over previous tracking software
- Modified track seeding to cope with higher energies (3TeV)



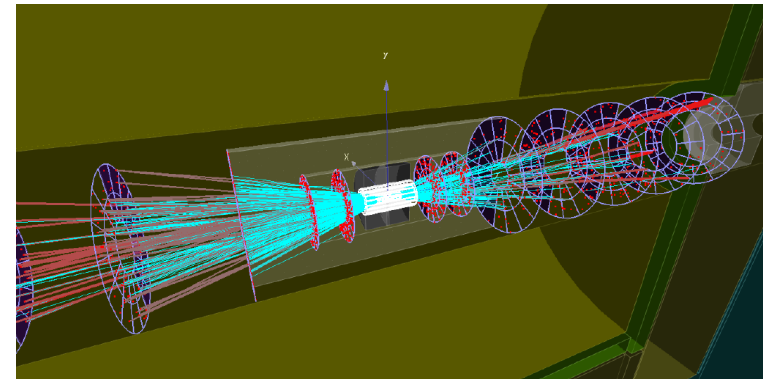
- Example events showing no loss of tracks, even in dense jets, where very close by tracks had been seen to suffer from losses due to merging.

TPC Reconstruction – Clupatra @ 3TeV

- Improving efficiency in dense forward jets, here the ratio of the distance between hits from the same track and that from neighboring tracks is less favourable.
- Also effort is currently being invested into merging of split tracks.



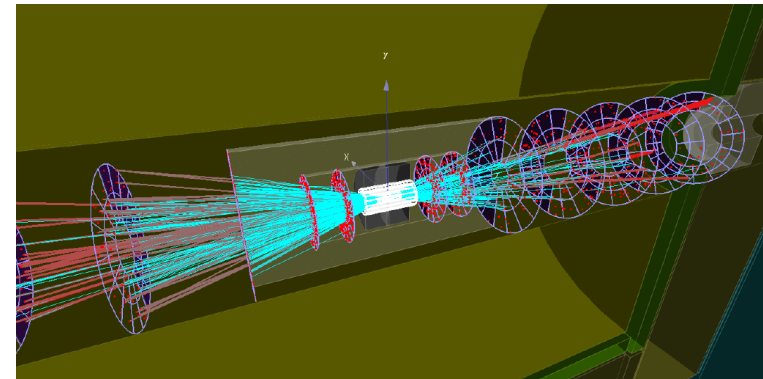
Track Reconstruction – Forward



- Best combination of Forward tracks with the those from other areas currently being studied.
- Next step: look into implementation of new ideas targeted at the overlay of background e.g. : multilayer perceptron, use of sector specific values for the Cellular Automaton, etc.
- The Hopfield Neural Network may be not even be necessary in most cases, as most conflicting tracks can be separated into groups of completely mutual exclusive tracks → just pick the best. Needs further investigation.

Track Reconstruction – Forward

The Package is currently being adapted so that the core functionality is decoupled from the ILD specifics

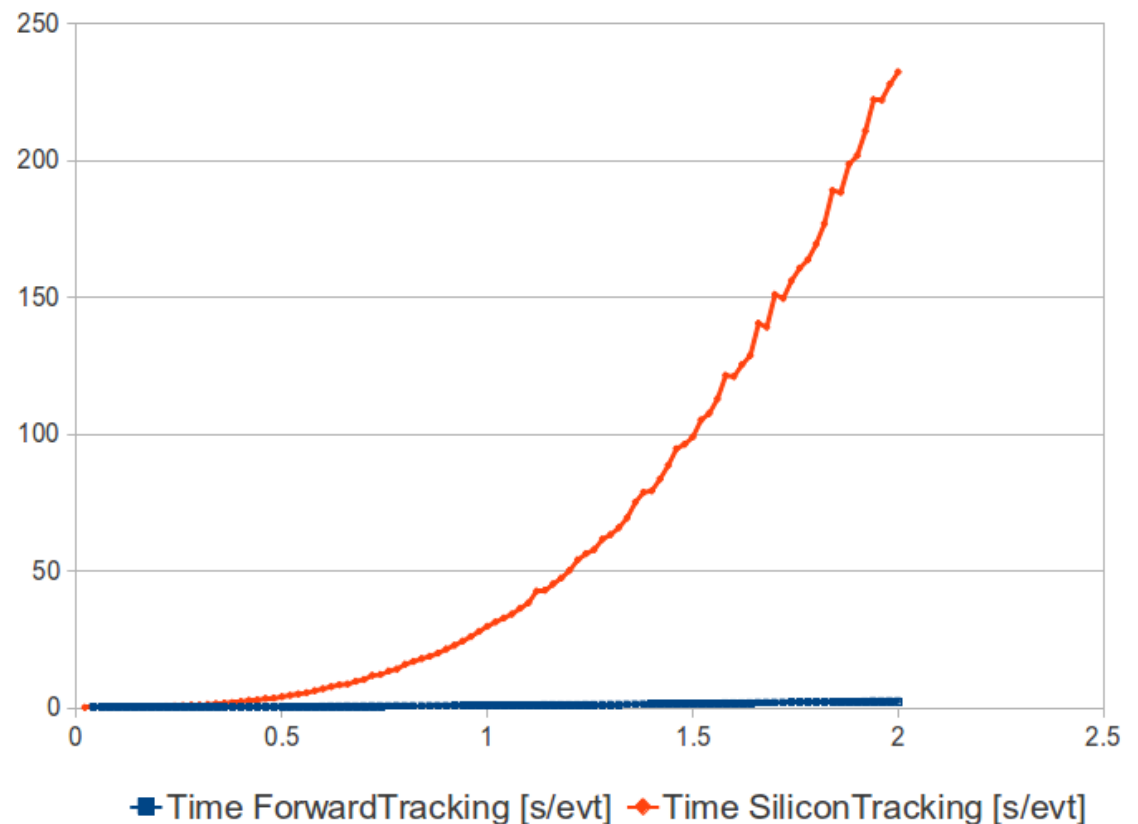


- Best combination of Forward tracks with the those from other areas currently being studied.
- Next step: look into implementation of new ideas targeted at the overlay of background e.g. : multilayer perceptron, use of sector specific values for the Cellular Automaton, etc.
- The Hopfield Neural Network may be not even be necessary in most cases, as most conflicting tracks can be separated into groups of completely mutual exclusive tracks → just pick the best. Needs further investigation.

Track Reconstruction – Forward

- Processing time behaves well (but well enough?)
- Comparison with previous Silicon Tracking shows large improvement against higher background.
- Question: Need to consider against more realistic background?

Speed and Efficiency



proc. time vs multiplicity scaled to LOI background

Summary and Outlook

- First full version of the new ILD track reconstruction software released. Validation on-going.
- The deliverable D2.4, M12: Software design for tracking toolkit, has been delayed by 3 months to allow time for the evaluation of the new ILD track reconstruction to feed into it.
- We will continue to prototype ideas in the iLCSoft framework, which can then be implemented within the Tracking Toolkit as it develops.
- Manpower reduction likely at OEAW, at least in the short-term

