EUDET-Memo-2010-12

EUTelescope.
Offline track reconstruction and DUT analysis software.

I. Rubinskiy*

December 1, 2010

Abstract

The final EUDET beam test telescope was equipped with Mimosa-26 sensors. the large pixel number per sensor along with higher readout rate lead to a dramatic increase of the beam test data to be analyzed. The EUTelescope offline track reconstruction optimization is discussed.

*DESY, Hamburg, Germany

1 Introduction

The final EUDET beam test telescope was equipped with Mimosa-26 sensors with more then 660K pixels on a sensor and being read at 110 us per frame. The data analysis software called EUTelescope [1] has been tuned for the previous version of the telescope (demonstrator telescope) and therefore had to be optimized. In this note the data analysis steps have been revised.

2 Data analysis steps

The EUTelescope data analysis is realized in terms of modules which are called Processors. This term determines an event-wise processing unit with it's initialization ("init()"), core ("processEvent()"), and finalization ("end()") methods. The core is executed once per every event, while the initialization and finalization methods - once per execution (a run or a group of runs). Fig. 1 shows a general data analysis sequence.

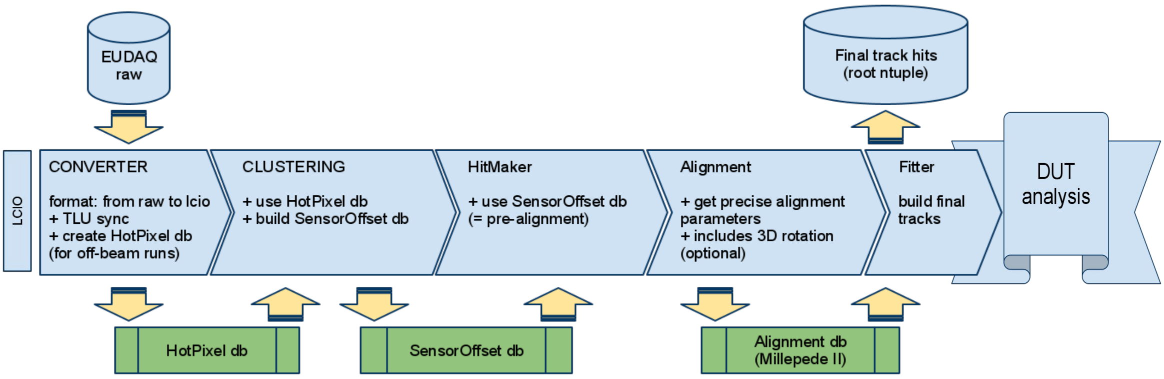

Figure 1: The EUTelescope data analysis steps.

2.1 Initial data conversion

The EUTelescope library is based on the ILCSoft framework and requires the input data to be stored in the "lcio" format. For this reason the data should be converted from the native EUDAQ format into the internal ILCSoft "lcio" data format. Identification of hot or noisy pixels can be performed optionally. This option (if taken) should be performed on a special run when there was no beam in the telescope. This a so called off-beam or a dry run.

2.1.1 Identification of hot pixels

During the data conversion each sensor can be scanned for a set of hot pixels, these are the pixels which fire more often then the others. The default values set for this step are: the scan is performed only in the first 10K events, pixels are identified as hot only if they fire more often then in 1% of all events. On the average for Mimosa-26 sensors operated at threshold S/N=10 this gives $\sim$ 600 hot pixels per sensor, which is in other terms only 0.001 of the sensor active area. Implementation of the hot pixel suppression in the clustering step the number of clusters is reduced by a factor 1.5-2. Thus the hot pixel suppression turned out to be also a valuable improvement in terms of is CPU time consumption in all subsequent steps.

2.2 Clustering

Clustering is an algorithm which defines that group of pixels created by one hit. There are basically only two ways to identify a cluster. The first one is a Fixed Frame Clustering algorithm, and the second - Sparse Clustering algorithm (two versions).

2.2.1 Fixed Frame Clustering algorithm (DFF)

Fixed Frame Clustering algorithm requires a maximum width and height of a cluster to be a priori known or guessed. A typical cluster size settings are 3x3 and 5x5. In some specific beam tests a sensor can be tilted in respect to the beam. In this case one of the cluster dimensions can be become quite large with a maximum width (or height) reaching the sensor width (or height).
The algorithm initially being a simple clone of the one used for analog sensors analysis suffered from inefficiencies in the vector and map management based on the event basis. The major source of inefficiency was hidden in the hot pixel suppression methods. This was resolved by introducing an external hot pixel map as discussed in Sec. 2.1.1. The tests showed a CPU performance improvement by a factor $\sim$ 10.

2.2.2 Sparse Clustering algorithm (SP,SP2)

There are two versions of the algorithm based on a nearest neighbour search. The first one (SP) requires at least one adjacent neighbour to have the signal to be above some predefined limit in order to merge the two pixels into a cluster. The second one (SP2) requires the distance to the next pixel (not necessarily direct neighbour) to be less then a predefined value.
The original version of the algorithm implementations (SP and SP2) was missing the hot pixel suppression, which has been implemented for consistency with DFF clustering. In the current version of the EUTelescope library the SP2 algorithm is recommended as the default one due to a more general approach for cluster finding and superior CPU wise performance.

2.3 Correlator

The Correlator Processor should be run together with the Clustering Processor as a recommendation. The Correlator in its essence is a simple histogram builder. Correlation plots belong to the most important set of histograms in the library.

A new option allows one to calculate and dump into a db file the sensor geometrical offset in X and Y directions. The offset values are calculated in respect to first sensor in the telescope. The offset values are treated as initial alignment of the sensors and are used (see 2.4 as corrections to the cluster coordinates in the telescope frame.

The Correlator can be ran also with the HitMaker Processor. In this case the correlation plots are used only as control plots.

2.4 Cluster coordinates in the telescope frame (HitMaker).

The cluster coordinates are initially given only in the local sensor frame in terms of column and row numbers. The transformation from the local (sensor) frame into the global (telescope) frame is done via the telescope description stored in the geometry steering files. The GEAR library responsible for this has been extended with the methods reading and retrieving values which define sensor rotations in the global frame. The HitMaker Processor has been supplied the proper methods to apply sensor rotations respectively.

2.5 Alignment

The sensor position can be determined with even higher precision within a strict mathematical model implemented in the Millepede II package.

The mathematical model has been extended to allow also full 3D rotations. The 3D angle rotations allow one to test sensors at any arbitrary tilt in respect to the beam. A good number of 10K tracks has been seen to give alignment precision at sub micron level.

2.6 Track fit

The hits from well aligned sensors can be used to be tracks candidates. Due to ever remaining noise in the sensors the combinatorics of possible hit combinations can significantly slow down the track search.

A set of tighter conditions on a track search has been implemented. The algorithm by which a track candidate is being formed incorporates a track segment estimate of its intersection with the nearest plane (or next one of no hit found), which is very much similar to the well known Kalman Filter approach. Unlike in the usual Kalman Filter method the track candidate is formed by collecting hits only in one direction, along the z axis.

The overall CPU wise performance improved by 2-4 orders of magnitude depending on the algorithm settings.

2.7 DUT analysis

The DUT analysis has become more straightforward and intuitive [3].

3 Conclusion

The EUTelescope library is a part of the ILCSoft package. The improvements in the library structure and individual elements have been presented in this memo. The improvements include: new approach of hot pixel suppression, clustering algorithms optimization, full Euler angle rotations in the geometry and alignment for the tilted sensor beam test data analysis, cone-based hit pruning for the track fitter. One of the main improvements is the sensor pre-alignment which makes the fine alignment with Millepede II fully automatic. The overall CPU wise performance of the library significantly improved.
During the last year of operations the EUTelescope library had went through 8 releases. The documentation has been placed in the ILCSoft portal [4, 5]..

Acknowledgement

This work is supported by the Commission of the European Communities under the 6^{th} Framework Programme "Structuring the European Research Area", contract number RII3-026126.

References

[1] A.Bulgheroni, T.Klimkovich, P.Roloff and A.F.Zarnecki, EUTelescope: tracking software, EUDET-Memo-2007-20.

[2] A.F.Zarnecki,Analytical track fitting: performance and new options, EUDET-MEMO-2010-014.

[3] V.Libov, EUDET-MEMO-2010-015.

[4] http://ilcsoft.desy.de/portal

[5] http://ilcsoft.desy.de/portal/software_packages/eutelescope/