EUDET-Memo-2010-14

Analytical track fitting: performance and new options

A.F.Żarnecki*

November 29, 2010

Abstract

The JRA1 group within the EUDET project developed a high resolution, low material pixel telescope to characterize new devices. Analytical track fitting method taking into account multiple scattering in the telescope sensors was implemented in the EUDET software framework to allow for highest track fitting accuracy. The algorithm has been recently tested with the new data taken at CERN with final telescope sensors. Modifications to the fitting algorithm and new options were implemented to improve code performance.

*Faculty of Physics, University of Warsaw; e-mail: zarnecki@fuw.edu.pl

1 Introduction

For low energy running, track displacement due to scattering of beam particles in telescope sensors, DUT or box windows, can be of the order of micrometers, comparable with the sensor position resolution. Therefor, it is crucial to take multiple scattering into account in the track finding algorithm. Following the analytical approach described in [1], dedicated track fitting method was developed and implemented in the Marlin processor called **EUTelTestFitter**.
The analytical approach was based on few simplifying assumptions:

- all telescope planes are parallel to each other
- the incoming beam is perpendicular to the telescope planes and has a small angular spread
- particle scattering angles in subsequent telescope layers are also small
- thicknesses of all material layers are very small compared to the distances between planes
- particle energy losses in telescope layers can be neglected

With these assumptions the problem of finding track position in each telescope plane by searching minimum of χ^2 function can be reduced to solving of a matrix equation. Moreover, track fitting can be performed separately in XZ and YZ planes (where Z is defined along the beam axis direction). The track fitting procedure was described in details in [2]. The fitting algorithm considers all possible track hypotheses. This would be very time consuming, especially at high hit multiplicities. However, if hypotheses are properly ordered, the procedure can be significantly optimized. This is based on the observation that hypothesis can be rejected without χ^2 calculation, if any subset of hits resulted in χ^2 value above the threshold.
To improve algorithm performance even further possibility of using beam direction constraint was added. With beam constraint track hypotheses can be rejected already based on the first two hits. The other method to improve algorithm performance was to allow same fit to be used in more than one track. With this assumption one has to check each track hypothesis only once. This option resulted in significant performance improvement, especially for samples with high track multiplicities. The algorithm, tested both on low energy (DESY) and high energy (CERN) test data taken in 2007 proved to be very fast, efficient and flexible [3].

2 New performance tests

Since summer 2009 EUDET telescope has been equipped with MIMOSA 26 sensors with digital readout. Data taken at CERN in June 2010, with 6 active sensor planes, were used to test EUTelTestFitter performance.

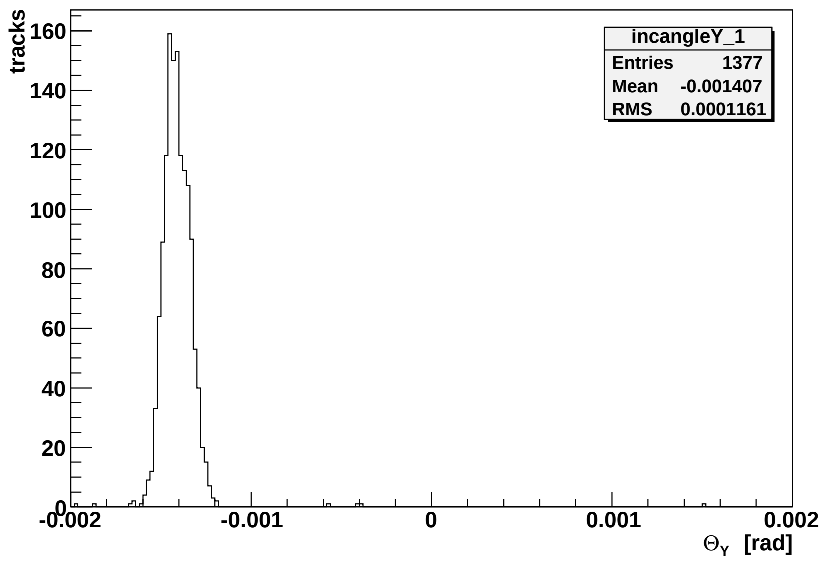

Figure 1: Vertical incident angle for particles entering EUDET telescope, as obtained from the track fit.

It turned out that the code performance deteriorated significantly in the new setup. The main reason was the necessity to switch off the two optimizing options mentioned above. The beam constraint (UseBeamConstraint option) could not be used because the beam was not perpendicular to the sensor planes. Significant beam tilt, much greater than the angular beam spread is clearly visible in Figure 1. The option allowing a hit to be used in more than one track (AllowAmbiguousHits option) also could not be used as it resulted in a large number of additional “ghost” tracks (number of fitted tracks was more than doubled with this option). With digital sensor readout large pixel clusters tend to be splitted into two (or more) sensor hits. This is clearly visible in Figure 2 where spacial separation between two hits in the same sensor is plotted. Clear tendency to have pairs of hits separated by less than $50\mu m$ is seen. If all hit combinations passing χ^2 cut are accepted then two adjacent hits (created by cluster splitting) result in two overlapping track hypothesis. This effect can be reduced by applying a tighter χ^2 cut, but then the track finding efficiency is significantly reduced.

3 Algorithm improvements

3.1 Beam constraint

To restore the possibility to use beam constraint in the fit two new control card options were added, which allow to set the estimated beam slope (BeamSlopeX and BeamSlopeY options). After setting the beam slope based on the analysis of a small subsample of events (and using other means to improve performance, eg. restricting analysis to full

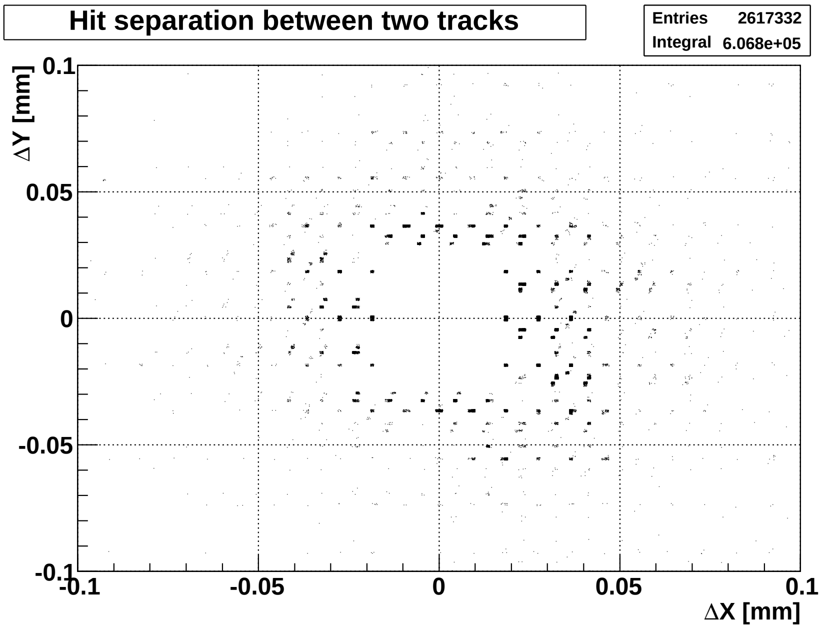

Figure 2: Spacial separation between two hits reconstructed in the same sensor. Note that both vertical and horizontal plot scales are much smaller than the sensor size to demonstrate short range correlations between hits.

tracks only), full analysis of the whole sample can be performed with significant CPU time gain. Shown in Figure 3 are the track finding efficiency (relative to the fit without beam constraint) and CPU time per event as a function of the assumed angular spread of the beam (used in a fit constraint). With proper choice of the beam spread algorithm speed can be improved by a factor of 20 without the significant loss of efficiency.

3.2 Ambiguous hits

Without the AllowAmbiguousHits option the fitting was much more time consuming as it was performed in iterative mode. In each pass the track with the lowest χ^2 value was looked for. If accepted (below the cut), the hits belonging to this track were removed from the data table and the whole track finding algorithm was repeated.
To improve track finding performance the new algorithm logic was implemented. Track fitting is now performed in two steps. In the first step all track hypothesis are considered and all tracks with χ^2 value below the cut are stored a dedicated map. The χ^2 value is used to sort the map and in the second step we loop over all selected tracks starting from those with smallest χ^2. The track is accepted if the number of hits already used (in tracks with lower χ^2 values) does not exceed the number given by MaximumAmbiguousHits parameter (new option). The advantage of the new solution is that it can be used also when planes with missing hits are allowed in the track (as set by AllowMissingHits option), while in the old implementation performance was improved only for full tracks

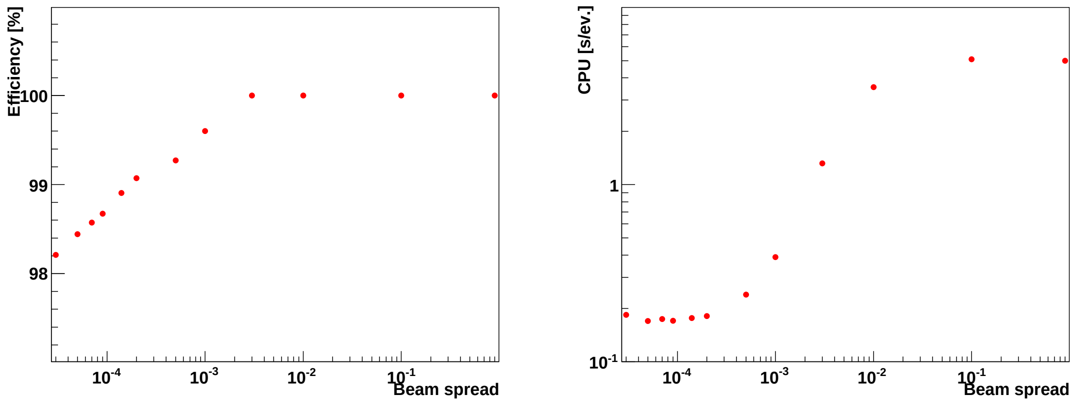

Figure 3: Track finding efficiency relative to the fit without beam constraint (left) and an average fitting CPU time per event (right) as a function of the angular spread of the beam assumed in the fit. Tracks with 4 to 6 fired planes (up to 2 missing hits) are considered (AllowMissingHits parameter set to 2).

(i.e. when AllowMissingHits was set to 0). In the new implementation of the algorithm track selection procedure (loop over all track possibilities) is run only once independently on the 'AllowAmbiguousHits' flag. The performance is significantly improved even if we do not allow same hit to be used more than once (i.e. when MaximumAmbiguousHits is set to 0). It was checked that setting MaximumAmbiguousHits to 1 or 2 does not increase the number of fitted tracks significantly (1 and 82 additional tracks were fitted in a sample of 1000 events i.e. about 3000 tracks, respectively). As can be seen in Figure 4 reduction of the processing time almost by a factor of 4 was obtained.

3.3 Track slope

Additional preselection of track candidates was implemented to improve track finding performance. This approach was suggested and implemented in the old code by Igor Rubinski. In the new version this part of code was completely rewritten to make it consistent with the logic of the track finding algorithm. This resulted in an additional performance gain.

The idea of using preselection cuts (UseSlope option) is based on the observation that some track hypothesis can be rejected even without calculating the χ^2 value. When using the beam constraint (UseBeamConstraint option) one can calculate, from the position measured in the first telescope layer and the beam slope, the expected hit positions in remaining telescope layers. If the measured position for any hit differs by more than the value of SlopeDistanceMax parameter from the expected position, analysis of this track hypothesis is stopped. Additional preselection cut can be also applied based on the estimated track scattering angle in telescope layer. If the scattering angle, as calculated from the hit positions in the subsequent layers, is greater than the assumed

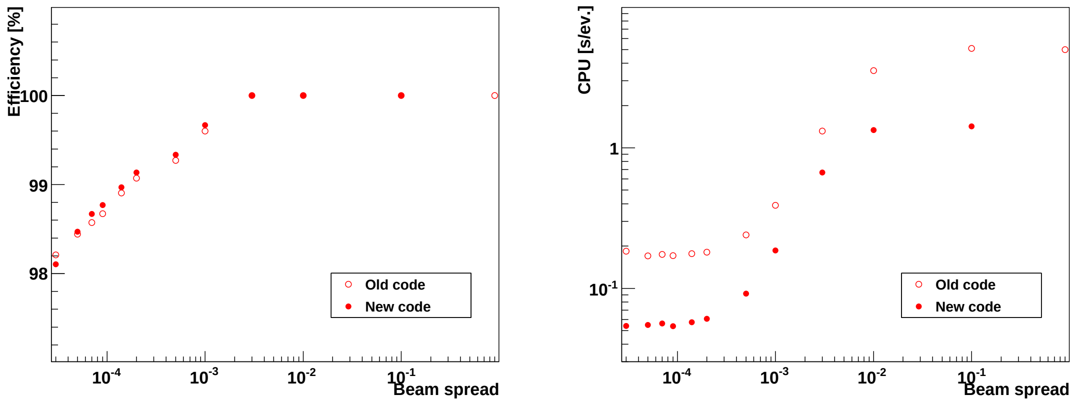

Figure 4: Track finding efficiency relative to the fit without beam constraint (left) and an average fitting CPU time per event (right) as a function of the angular spread of the beam assumed in the fit. With the new fitting algorithm implementation (filled points) significant performance gain is obtained with almost the same (or even slightly higher) track finding efficiency as in the old code (open symbols).

cut (SlopeXLimit and SlopeYLimit parameters) the track is not considered. Although the significant gain in time can be obtained (factor 5-10) one has to be careful, as to tight preselection cuts can significantly reduce track finding efficiency, as shown in Figure 5. One has to stress that the preselection criteria described above correspond to the position uncertainty term and multiple scattering term in the χ^2 formula (see [1] for details). The preselection cuts should be sufficiently loose not to reject tracks passing the final χ^2 cut. Otherwise the position errors resulting from the fit will not be reliable.

4 Conclusions

Analysis of EUDET telescope data taken with final digital sensor indicated the need for code improvements. Large parts of the fitting code were rewritten resulting in significant performance improvement, especially for tracks with missing hits (see Table 1). Additional options were also added to make code more flexible. New preselection options should be used with care.

Acknowledgement

This work is partially supported by the Polish Ministry of Science and Higher Education, project number EUDET/217/2006 (2006-2009) and by the Commission of the European Communities under the 6^{th} Framework Programme "Structuring the European Research Area", contract number RII3-026126.

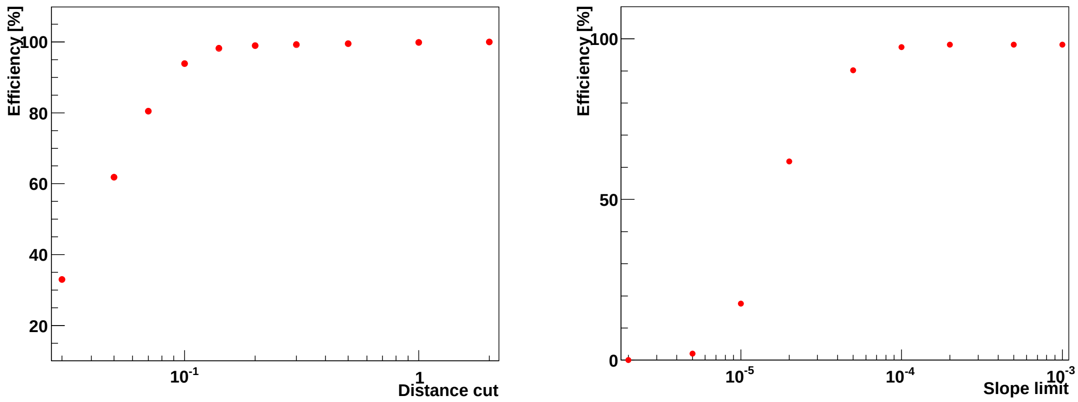

Figure 5: Track finding efficiency relative to the fit without preselection cuts for additional cut based on the estimated hit distance to the track (left) and on the estimated scattering angle (right).

CPU [s/event]	full tracks only	tracks with missing hits
Old version	0.38	5.0
New version	0.19	1.4
+ UseBeamConstraint	0.01	0.06
+ UseSlope	0.002	0.008

Table 1: Measured CPU time per event (Core 2 Duo laptop, 1.6GHz, 2GB RAM) for run 10233 with average of 20 hits per plane, 3 tracks per event. Performance of old code version is compared with new versions without and with beam constraint and preselection cuts. Track fitting results for tracks without missing hits (full tracks) as well as with missing hits (with AllowSkipHits and AllowMissingHits parameters set to 2) are shown.

Appendix: EUTelTestFitter options

Below, parameters defining performance of the algorithm are described. Some suggestion for the optimal choice of parameters are given at the end of the section.

A.1 Main algorithm parameters

Following parameters define input and output from the processor.

InputCollectionName Name of the input TrackerHit collection

OutputTrackCollectionName Name of the output Track collection (main output of the processor).

InputHitsInTrack Flag for storing input (measured, corrected for alignment) hits together with the track.

CorrectedHitCollectionName Name of the collection for storing corrected particle positions in telescope planes (hits), i.e. positions after alignment corrections, as used in the fit

OutputHitsInTrack Flag for storing output (fitted) hits together with the track. Input and output hits can be distinguished by looking into hit type (type $\leq$ 31 for measured hits, type $\geq$ 32 for fitted).

OutputHitCollectionName Name of the output collection of fitted particle positions in telescope planes (hits)

DebugEventCount Print out debug and information messages only for one out of given number of events. If zero, no debug information is printed.

HistoInfoFileName Name of the histogram information file. Using this file histogram parameters can be changed without recompiling the code.

Ebeam Beam energy in [GeV], needed to estimate multiple scattering.

A.2 Geometry description parameters

Following parameters can be used to adjust geometry description.

SkipLayerIDs Ids of layers which are described in GEAR but should not be included in the fit. Can be used to remove layers in front of and behind the telescope, which do not influence the fit, but can slow down the algorithm (increase fit matrix size).

PassiveLayerIDs Ids of layers which are described as active layers in GEAR but should be treated as passive in the fit (their data should be ignored).

AlignLayerIDs Ids of layers for which alignment corrections should be applied

AlignLayerShiftX Shifts in X, which should be applied to correct alignment of these layers.

AlignLayerShiftY Shifts in Y, which should be applied to correct alignment of these layers.

AlignLayerRotZ Rotation around Z (beam) axis, which should be applied to correct alignment of these layers.

WindowLayerIDs Ids of layers for which position cuts are defined. Only hits inside the defined "window" are accepted

WindowMinX Lower window edge in X

WindowMaxX Upper window edge in X

WindowMinY Lower window edge in Y

WindowMaxY Upper window edge in Y

MaskLayerIDs Ids of layers for which position cuts are defined. Only hits outside the defined "mask" are accepted

MaskMinX Lower window edge in X

MaskMaxX Upper window edge in X

MaskMinY Lower window edge in Y

MaskMaxY Upper window edge in Y

A.3 Fit performance parameters

Following parameters decide about fit performance in terms of selected track quality and constraints used. The choice of these options can significantly affect code performance, see comments below.

MaxPlaneHits Maximum number of hits considered per plane. Should be large (default is 100) for high track finding efficiency even in events with high hit multiplicities (but have a look at performance issues below). For low values algorithm runs much faster (reduced number of hit hypothesis), but the efficiency is lower. In the current version limitation coming from numerical precision (numbering of fit hypothesis) is no longer relevant.

Chi2Max Maximum χ^2 for accepted track fit.

SearchMultipleTracks Flag for searching multiple tracks in events with multiple hits. If false, only best (lowest χ^2) track is taken.

AllowAmbiguousHits Allow same hit to be used in more than one track. In the new implementation this option does not improve the performance (new optimization method works independently). This option can now be used also when missing hits are allowed (AllowMissingHits > 0)

MaximumAmbiguousHits Number of hits than can be shared by two track (when AllowAmbiguousHits option is set)

UseNominalResolution Flag for using nominal sensor resolution (as given in geometry description) instead of hit position errors. Improves tracking performance.

UseDUT Flag for including DUT measurement in the track fit.

UseBeamConstraint Flag for using beam direction constraint in the fit. Can improve the fit, if beam angular spread is small. Improves track searching for multiple hits.

BeamSpread Assumed angular spread of the beam [rad]

BeamSlopeX Assumed beam direction tilt in X-Z plane [rad]

BeamSlopeY Assumed beam direction tilt in Y-Z plane [rad]

AllowMissingHits Allowed number of hits missing in the track (sensor planes without hits or with hits removed from given track)

MissingHitPenalty "Penalty" added to track χ^2 for each missing hit (when no hit is left in active layer).

AllowSkipHits Allowed number of hits removed from the track (because of large χ^2 contribution)

SkipHitPenalty "Penalty" added to track χ^2 for each hit removed from the track because of large χ^2 contribution.

UseSlope Use additional cuts, based on measured particle position in the first layer and the expected track direction (beam slope given when UseBeamConstraint option is used; otherwise beam is assumed to be perpendicular to the sensor) to constraint number of considered hit combinations. Also the track slope change when passing the layer can be constrained. As the cuts are based on the measured positions (and not the fitted ones) this should only be used as preselection (cuts should not be too tight) - final selection should be done based on χ^2

SlopeXLimit Limit on track slope change when passing sensor layer (in X direction)

SlopeYLimit Limit on track slope change when passing sensor layer (in Y direction)

SlopeDistanceMax Maximum hit distance from the expected position, used for hit preselection (see above).

A.4 Performance issues

As described above, if multiple hits are found in telescope layers or hit rejection is allowed, the algorithm checks all hits selection possibilities (all track hypothesis). This task is optimized to a large extent, but still track finding can be slow for large multiplicities. Here are some suggestions on how to improve performance.

- Remove addition layers from geometry description. At high energies, when multiple scattering can be neglected, only active telescope planes and DUTs are relevant. At low energies you can still remove planes which are in front of the first active layer and behind the last one. You can also consider removing thin passive layers inside the telescope. Number of layers determines the order of matrix equation which has to be solved for each track.

- Use nominal plane resolutions instead of cluster position errors (set UseNominalResolution to true). For full tracks (without missing hits) matrix inversion is done only once and not for each track hypothesis.

- Use beam constraint (set UseBeamConstraint to true), even if beam spread is large. With beam constraint first two hits are sufficient to recognize bad track hypothesis. Without beam constraint at least 3 hits are needed. If the beam direction is not exactly perpendicular to telescope layers, beam tilt can be taken into account by setting parameters BeamSlopeX and BeamSlopeY.

- Use track preselection based on slope (set UseSlope to true). This helps a lot especially when the beam is well collimated and the energy is high (scattering in telescope planes small, so preselection parameters SlopeDistanceMax SlopeXLimit and SlopeYLimit can be set to small values)

- Do not allow for missing hits (set AllowMissingHits to 0). This reduces number of fit hypothesis and improves fit performance.

- Limit number of hits per plane. This should **not** be done by using MaxPlaneHits parameter, as it would bias plane efficiency calculation. Best way is to define position window in each active plane (see geometry description parameters above).

References

[1] A.F. Żarnecki and P. Nieżurawski, “EUDET telescope geometry and resolution studies”, EUDET-Report-2007-01, arXiv:physics/0703058.

[2] A. Bulgheroni, T. Klimkovich, P. Roloff and A.F. Żarnecki, “EUTelescope: tracking software”, EUDET-Memo-2007-20.

[3] A.F. Żarnecki and P. Nieżurawski,”Status of the analysis”, presented at the EUDET JRA1 meeting in Hamburg, January 2008.
`http://hep.fuw.edu.pl/u/zarnecki/talks/afz_jra1_jan08.pdf`