LHCb note 2007-020

Performance of the track matching

M. Needham
CERN

J. van Tilburg
Physik-Institut der Universität Zürich

March 12, 2007

Abstract

The procedure and the performance of the track matching algorithm at the time of DC'06 is described. The event-weighted efficiency is 79.3 % for all long tracks increasing to 86 % for tracks with $p > 5$ GeV. An approach to tune the matching algorithm with real data is presented and a discussion on future improvements to the algorithm given.

1 Introduction

In this note the performance of the track matching algorithm at the time of the DC' 06 data challenge [1] is described. This is one of two approaches developed to find tracks that traverse the entire LHCb spectrometer from the VELO to the T stations. An alternative approach the so-called 'Forward Tracking', is described elsewhere [2]. The track matching algorithm reconstructs long tracks by identifying good combinations of VELO and T seeds that have been found using stand-alone algorithms [3, 4].

This note is structured as follows. First the algorithm is described. This is followed by a discussion of its current performance. Then, strategies for tuning the algorithm with real data are discussed and suggestions are given for improving the performance of the algorithm. Finally, the current settings for the algorithm parameters are described in an appendix.

2 Algorithm description

The algorithm proceeds as follows. First 'good' T seeds are selected. This is done by requiring that:

- The momentum is greater than 2 GeV.
- The χ^2/ndof is less than 50.
- The likelihood of the seed [3] is less than -30.

Each selected T seed is then extrapolated to $z = 830$ mm using a 5^{th} order Runga-Kutta method [5]. The VELO tracks are transported to the same plane using a linear extrapolator. Using the track angles provided by the VELO track and the momentum provided by the seed the transverse momentum is calculated and required to be more than 80 MeV. If this is the case a matching χ^2 is calculated as:

$$\chi^2_{\text{match}} = (\vec{x}_{\text{VELO}} - \vec{x}_{\text{T}})^T (C_{\text{VELO}} + C_{\text{T}})^{-1} ((\vec{x}_{\text{VELO}} - \vec{x}_{\text{T}}) \quad ,$$

where $\vec{x}_{\text{VELO}}$ and $\vec{x}_{\text{T}}$ are the track parameters of the VELO and T seeds at $z = 830$ mm and C_{VELO} and C_{T} are the corresponding covariance matrices. If this value is less than a predefined cut the combination of VELO and T seeds is stored in a temporary list of valid track candidates. After all valid combinations of T seeds and VELO tracks have been made this list is sorted by increasing χ^2_{match}. At this stage there are two possibilities:

- All combinations are considered as valid track candidates.
- Only the combination of a given T seed and VELO seed with the lowest χ^2_{match} is kept.

Currently the algorithm is configured to run in the second way.

The effect of the material between the end of the T stations and the matching position is ignored in this procedure to reduce the processing time of the algorithm. Since the uncertainties on the track parameters are dominated by the effect of multiple scattering, the χ^2_{match} values are consequently overestimated. Therefore, to get reasonable efficiencies the value of χ^2_{match} has to be set to rather high values. Since low momentum particles are more effected by multiple scattering this also means the performance of the algorithm for low momentum particles is consequently reduced.

2.1 Adding TT hits

In the final step of the matching procedure, the corresponding TT hits are added to each matched track. The trajectory through the TT stations is estimated by extrapolating the VELO track to each TT layer, using the momentum from the T seed. Due to multiple scattering in RICH 1, the predicted trajectory may deviate from true trajectory. However, this deviation is approximately equal for all TT hits belonging to the same particle. The algorithm exploits this idea by searching for *groups* of TT hits having approximately the same distance from the predicted trajectory.

In the search, only one measurement per TT layer is allowed. This means that a group consist of maximally four TT hits. Only hits which have a distance smaller than 3 mm are considered in the search. The hits in a group are not allowed to differ in distance by more than 1 mm (in the same station) or 2 mm (in different stations). When two or more hits in the same layer are compatible with the group, a separate group is created for each hit. Furthermore, a group should have at least three TT hits.

When a matched track has more than one group of TT hits, only the one with the smallest quality is selected. The quality of a group is defined as

$$q^2 = \bar{d}^{\,2} + w^2_{\mathrm{spread}} \times s^2_d \quad , \tag{1}$$

where $\bar{d}$ is the mean distance, w_{spread} is a weight factor, and s_d is the rms spread of the distances. The weight factor is tuned such that the TT hit efficiency is the highest. Studies on DC '04 data in [6] indicate that optimal performance is achieved when a value is chosen at $w_{\mathrm{spread}} = 7$.

3 Performance

The performance of the algorithm has been studied using data generated for the DC' 06 production. Four data samples were used:

- A sample of 25000 $B^+ \to D^0 K^+$ events generated at the default LHCb luminosity of 2×10^{32} cm^{-2}s^{-1}.
- A sample of 2000 $B_d \to J/\psi(\mu^+\mu^-)K_S(\pi^+\pi^-)$ events generated at the default LHCb luminosity of 2×10^{32} cm^{-2}s^{-1}.
- A sample of 4000 $B_d \to J/\psi(e^+e^-)K_S(\pi^+\pi^-)$ events generated at the default LHCb luminosity of 2×10^{32} cm^{-2}s^{-1}
- A sample of 1000 $B_d \to J/\psi(\mu^+\mu^-)K_S(\pi^+\pi^-)$ events generated at a luminosity of 5×10^{32} cm^{-2}s^{-1}.

The majority of results were obtained with the first sample. From the context it should be clear when this is not the case. The definitions of efficiency and ghost rate are given in [7]. The following efficiencies are determined with respect to those particles which are reconstructible as long tracks.

The performance of the algorithm depends on the value of the χ^2_{match} that is used. In Fig. 1 the efficiency versus the ghost rate is plotted for various values of this variable. From this plot it can be seen that a cut at $\chi^2_{\mathrm{match}} = 1000$ gives a reasonable performance. Using this value an event-weighted efficiency of 79.3 % is obtained [1]. The following results are obtained with a cut at $\chi^2_{\mathrm{match}} = 1000$.

The efficiency of the algorithm depends quite strongly on the track momenta. This can be seen in Fig. 2 where the efficiency is plotted as a function of the track momentum. For tracks with $p > 5$ GeV an efficiency of ~ 86 % is found. Below 5 GeV the efficiency falls rapidly. This reflects the fact that low momentum tracks are penalized in the matching procedure because the

---

[1] The corresponding track-weighted efficiency would be 77.7 %.

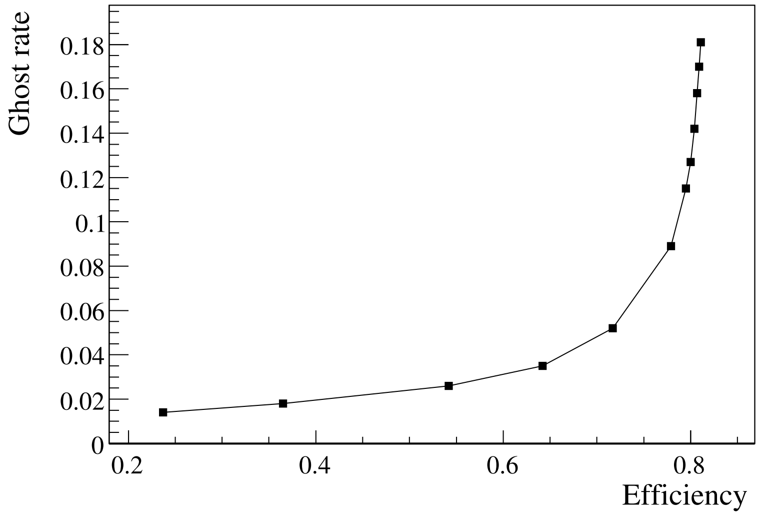

Figure 1: Track finding efficiency versus ghost rate for various cuts on the χ^2_{match}. The points from left to right correspond to cuts at 5, 20, 50, 100, 200, 500, 800, 1000, 1200, 1500, 1800 and 2100.

effect of multiple scattering is ignored. In Fig. 3 the dependence of the efficiency on the pseudorapidity of the track is given. Within the LHCb acceptance the efficiency is mostly flat. However, there is a dip at $\eta \sim 4.3$. This is attributed to the material of the 25 mrad conical section of the beam pipe which lies within the acceptance of the detector.

The efficiency for reconstructing tracks that originate from B decays has also been investigated. The results are summarized in Table 1. For the case of muons from $B_d \to J/\psi(\mu^+\mu^-)K_S(\pi^+\pi^-)$ the performance is compara-

Track type	$\overline{p}$ (GeV)	Track efficiency (%)
$\mu^\pm$ from $B_d \to J/\psi(\mu^+\mu^-)K_S(\pi^+\pi^-)$	33	83.1 ± 0.7
$e^\pm$ from $B_d \to J/\psi(e^+e^-)K_S(\pi^+\pi^-)$	34	80.6 ± 0.6
$\pi^\pm$ from $B_d \to J/\psi(\mu^+\mu^-)K_S(\pi^+\pi^-)$	12	56.7 ± 2.5

Table 1: Efficiencies for reconstructing tracks from specific B final states. The column labeled final state efficiency refers to the efficiency for reconstructing both tracks.

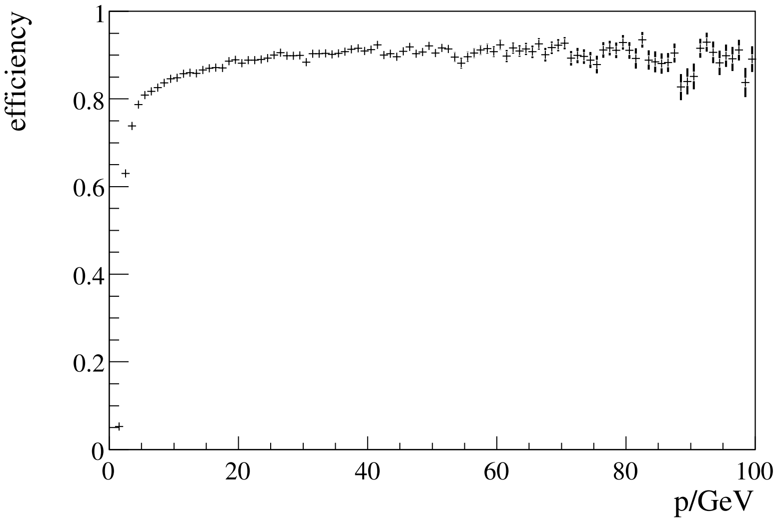

Figure 2: Track finding efficiency as a function of the track momentum.

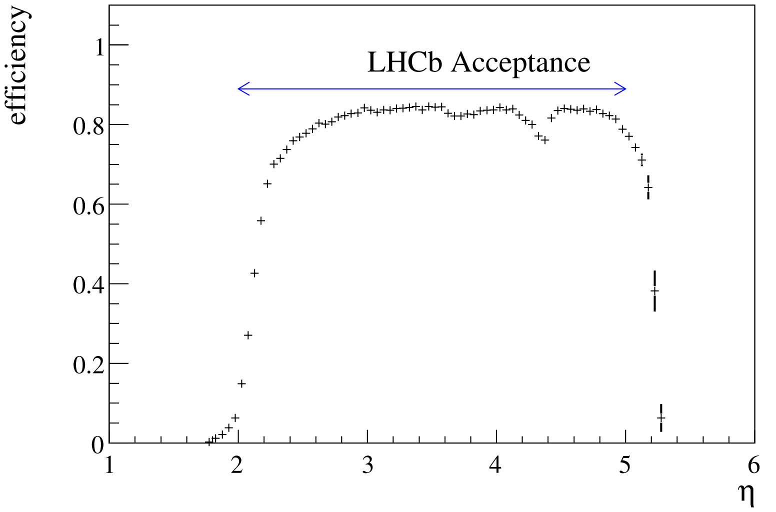

Figure 3: Track finding efficiency as a function of the track pseudorapidity η.

ble to that of the inclusive track sample. The performance for pions from $B_d \rightarrow J/\psi(\mu^+\mu^-)K_S(\pi^+\pi^-)$ is poor. There is only a 56.7 % probability to reconstruct a track from this source. This is explained two factors. First, the VELO track finding performance degrades for particles that originate far from the interaction point [4]. If the VELO tracking had the same efficiency as for the inclusive track sample an efficiency of 66 % would be expected. Second, tracks from this source tend to be quite soft. Therefore, the efficiency is further reduced by the poor performance of the matching for low momentum tracks.

An event-weighted ghost rate of 13.7 % is found[2] with the default value of the χ^2_{match} cut. Since this value is stored in the **Track** class [8] it is possible to reduce the ghost rate at a later stage — though at the expense of some loss in efficiency. In Fig. 4 the distributions for four different variables are compared for real and ghost tracks. The four variables are:

- The weighted number of measurements on the track defined as:

  $$n_{meas} = n_{VELO} + n_{TT} + n_{IT} + 0.5 \times n_{OT} \quad ,$$

  where the weight of 0.5 takes accounts of the fact that the OT gives on average twice as many measurements per track as the IT.
- The χ^2/ndof.
- The track pseudorapidity.
- The track's transverse momentum.

Compared to real tracks ghost tracks have less measurements and a worse χ^2/ndof. In addition, they tend to lie at high η and also around $\eta = 4.3$ [3]. Finally, it can be seen that ghost tracks have on average a lower p_T than real tracks. Either one or a combination of these variables could also be used to reduce the ghost rate.

The efficiency obtained with the algorithm is far from 100 %. The reasons for this are as follows. First, only the best combination of a T seed with a VELO seed is selected as a valid track candidate. Since for incorrect matches the distribution of χ^2_{match} is flat it can be that the best combination is not the correct one. If all combinations with $\chi^2_{\mathrm{match}} < 1000$ are kept an efficiency of 84.9 % is found. However, the event-weighted ghost rate increases dramatically to 40.0 %. The second reason is that the cut on χ^2_{match} at 1000 removes already 1.4 % of good combinations. Finally, the cut

---

[2] The track-weighted ghost rate is 16.6 %.

[3] This effect is also attributed to the 25 mrad cone of the beam pipe.

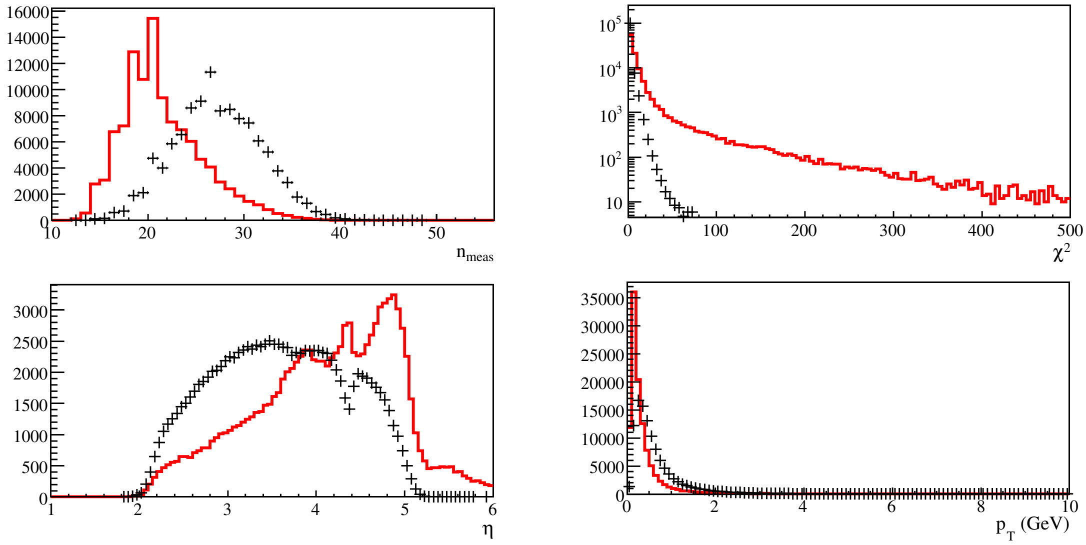

Figure 4: Comparison of the properties of real (points) and ghost tracks (line). The four variables considered are: number of measurements, χ^2/ndof, η, p_T.

on the track momentum also removes some good combinations. The effect of loosening these requirements is summarized in Table 2. It can be seen that at best an efficiency of 86.8 % can be achieved. This is close to the maximum that can be expected given the efficiency of the T seeding and VELO seeding algorithms. However, the corresponding ghost rate of 55.7 % is unacceptably high.

Conditions	Efficiency (%)	Ghost rate (%)
All combinations	84.9	40.8
+ relax χ^2_{match} cut to 2000	86.3	55.7
+ relax p cut to 1 GeV	86.8	55.8

Table 2: Efficiencies and ghost rates when relaxing the cuts.

The performance as a function of the number of visible interactions as defined in [9] has been investigated. Fig. 5 shows the dependence of the efficiency and ghost rate on this quantity. For each additional visible interaction in the detector the efficiency decreases by 2.4 % whilst the ghost rate increases by 4.6 %. The performance with data generated at a luminosity of

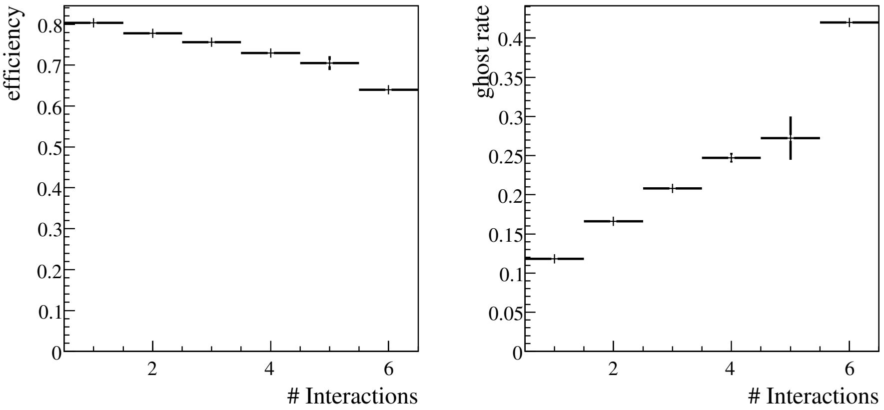

Figure 5: Efficiency (left) and ghost rate (right) versus the number of visible interactions.

5×10^{32} cm^{-2}s^{-1} has also been studied. In this case an efficiency of 77.5 % and a ghost rate of 16.3% is found. It should be noted that if only the number of visible interactions in the event spill affects the performance of the track reconstruction, then efficiencies and ghost rates for an arbitrary luminosity can be derived directly from Fig. 5. At a luminosity of 5×10^{32} cm^{-2}s^{-1}, on average there are two visible interactions per B event. From Fig. 5 the corresponding efficiency is 77.6 % and the ghost rate 16.4 % — in good agreement with the observed values at a luminosity of 5×10^{32} cm^{-2}s^{-1}. At higher luminosities this extrapolation will at some point break down due to increased spillover that further increases occupancies and detector dead time.

Finally, it should be noted that a luminosity of 2×10^{32} cm^{-2}s^{-1} the algorithm takes 75 ms per event on a 2 GHz Intel Centrino processor. At 5×10^{32} cm^{-2}s^{-1} the time per event increases to 105 ms.

4 Tuning the algorithm with data

All the studies presented in this note are made with simulated data where the performance can be judged against Monte Carlo truth. In reality the performance has to be understood without this information. The most important parameter to tune in the case of the track matching is χ^2_{match}. With data, one would like to choose the working point that optimizes the efficiency versus the ghost rate as in Fig. 1. Therefore, estimators that are correlated to the efficiency and ghost rate that can be determined from data alone are needed.

With data the relative efficiency of the reconstruction algorithm can be checked by studying the yield of J/ψ or K_S as a function of the cut on χ^2_{match}. Such an approach has been used in the HERA-B experiment to tune their equivalent of the matching algorithm [10]. In this case it was also found that as the cut loosened the background under the K_S peak increased. They argue that this is due to the increase in the number of ghost tracks produced by the matching. However, it could be that the increase is simply due to having more good matches. Another possibility to evaluate the ghost rate is to assume that it is related to the number of times a T-seed can be used to make a valid match. Then, simply plotting the J/ψ or K_S yield versus the number of valid combinations will give an indication of the optimal value of χ^2_{match}.

A first study of these ideas has been made using the sample of 2000 $B_d \rightarrow J/\psi(\mu^+\mu^-)K_S(\pi^+\pi^-)$ events. To obtain a measure of the J/ψ yield the number of events was counted where both tracks from the J/ψ were reconstructed[4]. Figure 6 shows the J/ψ yield versus the χ^2_{match} cut whilst Fig.7 shows the J/ψ yield versus the number of reconstructed tracks. From these plots it would be concluded that a cut on χ^2_{match} at around 1000 (the same value chosen from the studies with Monte Carlo) optimizes the J/ψ yield whilst keeping the ghost rate low. It should also be noted that tracks from K_S maybe a better choice to use in the tuning procedure as they have a momentum spectra that is more comparable to a 'generic' track sample. However, the current efficiency for finding such tracks in the VELO is low which may introduce some other bias.

5 Route Map for Development of the Algorithm

There is clearly room to improve the performance of the algorithm. From the studies that have been presented in this note two weak points of the algorithm are apparent:

---

[4] This simplification was made to save time. In a proper study the J/ψ yield should be estimated by running the standard selection algorithm. However, given that the J/ψ gives a clear signal with little background it is expected that the full study will similar results.

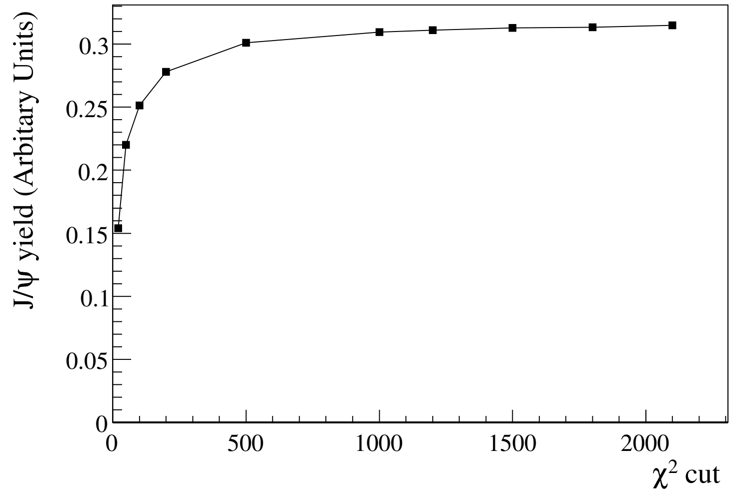

Figure 6: J/ψ yield versus χ^2_{match} cut.

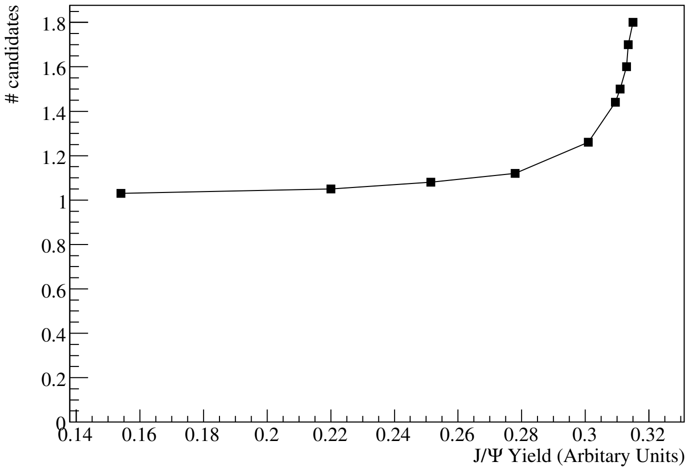

Figure 7: Number of times a T-seed is matched versus J/ψ yield. The points from left to right correspond to values of χ^2_{match} of 20, 50, 100 ,200, 500, 1000, 1200, 1500, 1800, 2100.

- Taking only the best combination of a T seed and VELO track leads to a sizable inefficiency.
- Ignoring the material in the matching algorithm means large cuts on

the χ^2_{match} are needed to reach high efficiency.

A high efficiency together with more intuitive values for the χ^2_{match} cut can be achieved by using an extrapolator that takes accounts of the material in the detector together with running the algorithm in a mode that takes all valid matches below the χ^2_{match} cut. However, there is a price to pay. First, the current mechanism for accessing the detector material, the transport service is slow. Using an extrapolator that accesses the material in the detector the CPU time of the algorithm increases by a factor of three [5]. In addition, by taking all combinations the ghost rate increases dramatically. There are several possible ways to reduce the ghost rate. First, after the matching a further cleaning of the candidates could be performed based on the reconstructed properties of the candidate track (c.f. Fig. 4). As in the current algorithm tracks could be ranked according to some criteria. If the ranking were based on the χ^2_{match} there would be no difference to the current algorithm. However, by deferring the decision, more information becomes available — for example, the total number of hits on the track including those in the TT station and the χ^2/ndof of the full track fit. One possibility would be to rank the tracks by the number of hits and χ^2_{match}. Alternatively, if a fast fit of the tracks is available the ranking could be based on the number of hits and the χ^2/ndof of the fit. In addition, two VELO seeds should be allowed to share the same T seed. This allows both tracks originating from a photon conversion to be reconstructed. One of the main focuses of further studies would be to determine the criteria that optimally reduce the ghost rate.

Another intrinsic problem of the matching procedure is that it requires a reliable estimate of the input covariance matrices. This is currently not the case for the simple fit of the VELO seeds and it is also not the case for the T seeds as their extrapolation to the VELO region does not account for multiple scattering. A simple solution could be to use one fixed covariance matrix for all candidate tracks. This approach would also make the matching algorithm less sensitive to seeds with large errors in their covariance matrix as they tend to decrease the χ^2_{match}.

Finally, it should be noted that there is room to improve the speed of the algorithm. By re-organizing the code such that the number of operations carried out in loops is minimized it should be possible to gain at least a factor of three in speed.

---

[5] Work is ongoing to improve the speed of the transport service.

References

[1] Gauss 25r7, Boole v12r10, Brunel 30r14, XmlDDDB 30r14.

[2] O. Callot and S. Menzemer. Performance of the forward tracking. LHCb-note 2007-015.

[3] R. Forty M. Needham. Standalone Track Reconstruction in the T-Stations. LHCb-note 2007-022.

[4] D. Hutchcroft *et al.* Velo Pattern Recognition. LHCb-note 2007-013.

[5] A. Spiridinov. Optimized Integration of the Equations of Motion of a Particle in the HERA-B Magnet. *HERA-B note*, 98-133.

[6] J. van Tilburg. *Track reconstruction and simulation in LHCb.* PhD thesis, Vrije Universiteit Amsterdam, 2005.

[7] M. Needham. Combined Long Tracking Performance. LHCb-Note 2007-019.

[8] J. Hernando and E. Rodrigues. Tracking Event Model. LHCb-note 2007-007.

[9] The LHCb Collaboration. Reoptimized Detector Design and Performance. CERN/LHCC LHCC-2003-030.

[10] A. Spiridinov. Tracking in the High Rate Enviroment of the Hera-B detector. *Nuclear Instruments and Methods A*, 566:154–156, 2006.

A Algorithms and Tools

class: TrackMatchVELOSeed

Derived from: GaudiAlgorithm

Purpose: Top Level Algorithm that drives the matching

JobOptions Parameters:

Property	**Description**	**Default**
InputVeloTracks	location of input VELO track container	TrackLocation::Velo
InputSeedTracks	location of seed track container	TrackLocation::Tsa
Chi2MatchingCut	Cut on the χ^2 of the matching	1000
AllCombinations	Take all combinations below the χ^2 cut	false
ptCut	Cut on the p_T of a match combination	80 MeV
MomentumCut	cut on the T seed momentum	2 GeV
VeloXCut	Maximum error on x of VELO track	0.6 mm
VeloYCut	Maximum error on y of VELO track	0.6 mm
VeloTxCut	Maximum error on tx of VELO track	8×10^{-4}
VeloTyCut	Maximum error on ty of VELO track	8×10^{-4}
SeedXCut	Maximum error on x of Seed track	150 mm
SeedYCut	Maximum error on y of Seed track	30 mm
SeedTxCut	Maximum error on tx of Seed track	0.1
SeedTyCut	Maximum error on ty of Seed track	0.01
MatchAtZPosition	z Matching position	830 mm
VariableZ	Match at a variable z	false
VarZParameters	Parameters for variable z matching	—
AddTTClusters	Add clusters in TT to the track	true
AddMeasurements	Add measurements to the track	false
Chi2SeedCut	Cut on the χ^2 of the seed	50
LikCut	Cut on the likelihood of the seed	-30
ExtrapolatorVelo	VELO Extrapolator Name	TrackLinearExtrapolator
ExtrapolatorSeed	T-seed Extrapolator Name	TrackHerabExtrapolator
TTClusterToolName	Tool to add TT hits	AddTTClusterTool

class: AddTTClusterTool

Derived from: GaudiTool, IAddTTClusterTool, IIncidentListener

Purpose: Tool to add TT hits to match candidates

JobOptions Parameters:

Property	**Description**	**Default**
TTClusterCut	Window for collecting hits	3 mm
MinTTHits	Number of TT hits	3
InterStationCut	Max distance 2 hits from different stations	2.0 mm
IntraStationCut	Max distance of 2 hits within a station	1.0 mm
SpreadWeight	Add description	7.0
AddLHCbIDs	Add LHCbIDs only to track	true
AddMeasurements	Add measurements	false
Extrapolator	Extrapolator type	TrackHerabExtrapolator
TTGeometryPath	path to DetectorElement	DeSTDetLocation::location(”TT”)
TTClusterPositionTool	TT cluster tool	STOfflinePosition
yTol	tolerance in y	20.0 mm

class: TrackMatch

Derived from: —

Purpose: Working class to describe a match track candidate

class: TTCandidate

Derived from: —

Purpose: Working class to describe a candidate cluster of TT hits