LHCB/2001–109
Tracking
3 October 2001

Track Seeding

Roger Forty

CERN, Geneva

Abstract

An algorithm for track seeding is described, the first step of pattern recognition in the LHCb main tracking system. This involves the search for track segments that are close to straight lines in the set of tracking stations positioned downstream of the spectrometer dipole magnet, in a region of low magnetic field. The performance of the algorithm is discussed.

1 Introduction

The current design of the LHCb spectrometer is shown in Fig. 1 [1]. The tracking system is divided into stations, each consisting of a number of layers with varying orientation of the sensitive elements. In the present layout, there are nine such stations, distributed between the vertex detector and RICH-2. Four stations (T6–T9), the furthest downstream,[1] are denoted the seeding stations. They are in a region of relatively low magnetic field—the fringe-field from the dipole has a mean value of about 300 gauss there—so the tracks are close to straight lines, aiding pattern recognition.

The standard strategy for pattern recognition in the main tracking system therefore starts with a search for track "seeds" in the seeding region. These track seeds are then followed upstream into the magnet, towards the vertex detector, in a second step [2]. An alternative approach would be to use as seeds the track segments found in the Vertex Detector, for which the pattern recognition has already been studied as they form a part of the early trigger levels of LHCb [3]. The Vertex Detector is also in a low-field region, and the occupancy is low there. However, following the track candidates found in the Vertex Detector downstream, into the magnet, looks difficult in the current setup: the magnetic field turns on fast (in particular due to the shielding plate that serves to reduce the field in the region of RICH-1), so extrapolating from the vertex region to T3 is not easy.

Furthermore, only the "physics" tracks would be found by such a downstream track search, i.e. those which really originate in the vertex region. For the current spectrometer design a substantial fraction (more than half) of the tracks reaching RICH-2 are secondary particles produced in the material of the beam-pipe and upstream detector components. While they are background tracks for physics analysis, they play an important role in the RICH pattern recognition, as the understanding of the observed pattern of detected photons is improved if the tracks that contribute are reconstructed. For such tracks, pattern recognition in the seeding region will be necessary. They will also be relevant for the analysis of other downstream detectors, the calorimeters and muon system.

It is likely that the final tracking analysis will include both upstream and downstream passes, to maximise efficiency. If the downstream following of tracks into the magnetic field turns out to be too difficult, an alternative approach (which may have application in the Level-2 trigger) would be to match track segments found in the vertex detector and seeding region, measuring their momentum from the p_{T} kick [4]. In this case, pattern recognition in the seeding region will still be necessary.[2]

The algorithm that is described here is not presented as the the best possible solution

[1] Downstream refers to the direction of increasing z coordinate in the experiment, for which the origin is at the nominal interaction point and z increases towards the muon detector; y is vertical and x completes the right-handed coordinate system.

[2] Yet another approach to downstream track seeding has recently been investigated: matching the Vertex Detector track candidates to each hit in the seeding region in turn, and using the resulting constraints on the track parameters to search for other hits on the track in the seeding region; finally the track–hit pairing that gives the best overall track fit is selected. This looks promising, but will be described elsewhere.

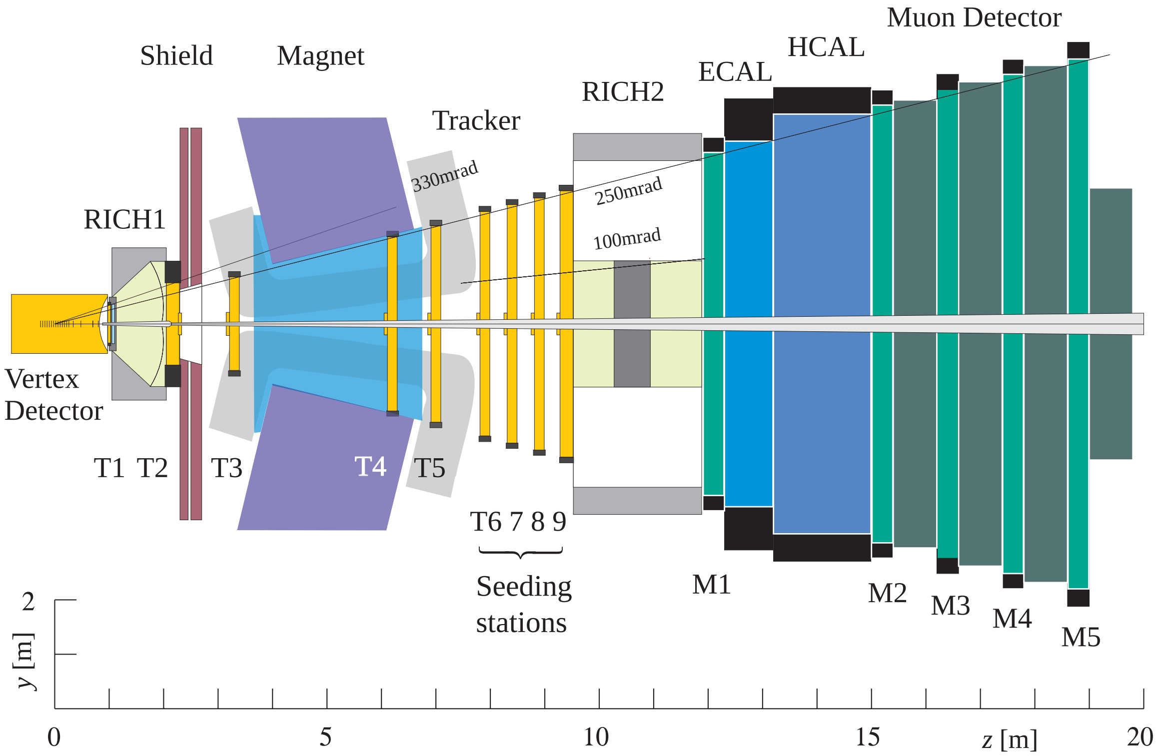

Figure 1: Schematic layout of the LHCb spectrometer, viewed from the side (the non-bend plane of the magnet); the tracking stations are marked as T1–9, with the seeding stations indicated.

to the pattern recognition problem. Rather, it was initiated at a time when no seeding algorithm existed, and issues such as the space required for the seeding stations needed to be addressed. As such, it should be considered as a first attempt, against which other ideas can be tested. The major problem that arises is that of combinatorics: if one tries all possible combinations of hits to make candidate tracks, the processing time of an event quickly becomes too long. For example, a brute force method might be considered as follows: working in the (x, z) projection of the Outer Tracker, join all pairs of hits in stations T7 and T9 with a straight line, open a road of ± 1 cm around the extrapolation of that line (well matched to the cell size), and count the number of hits inside that road. For physics tracks, of which there are about 30 per event, 99% have 12 or more hits when counted this way.[3] But, on average, there are 25,000 other pairings that also satisfy this requirement per event! To avoid such unwieldy combinatorics, the approach followed here is to first search for track segments (referred to as "stubs") in each station separately; those stubs are then linked to form seed candidates.

The simulation used for the studies presented here is described in the following section, followed by a description of the algorithm. The implementation in software is then discussed, and finally performance of the algorithm applied to simulated events is presented.

[3]If a track gives a hit in each layer it would have 16 hits.

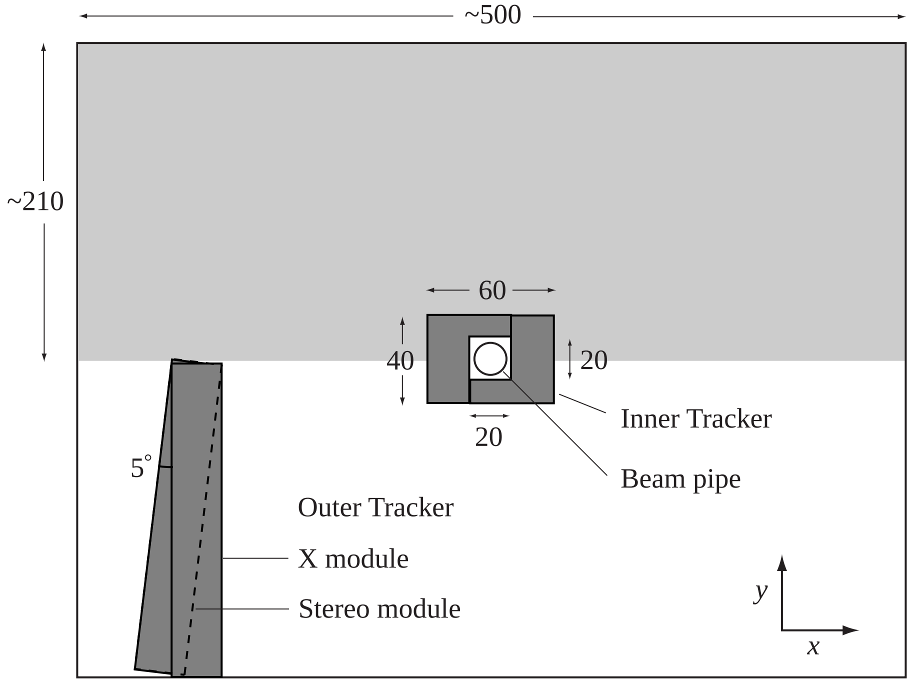

Figure 2: Schematic view of the layout of the seeding stations (transverse to the beam axis, not to scale). The Outer Tracker is divided into upper and lower sectors—the upper sector is lightly shaded—and each sector is constructed from X and stereo modules, as illustrated in the lower half; the Inner Tracker is made up of two L-shaped sectors in this simulation, as described in the text. For stations T6–T9 the Outer Tracker dimensions are adjusted in the simulation to follow the 300×250 mrad acceptance, whilst the Inner Tracker dimensions (shown in cm) are unchanged.

2 Simulation

The algorithm described here has been tuned on the recently-implemented detailed simulation of the Inner and Outer Tracker detectors. This represented a significant change compared to the earlier, simpler, description of the detectors, in which the stations were treated as boxes with equally spaced detector layers inside. Now the module sub-structure of the Outer Tracker is implemented, with 64 straws per module, each with 5.0 mm active diameter on a 5.25 mm pitch. The hit efficiency is modelled as described in Ref. [1], with an average of 97% across the active diameter of each cell for isolated tracks, but single-hit electronics so that if more than one track cross the cell in a given bunch-crossing only the track passing closest to the wire gives a hit. The spatial resolution is taken to be 200 μm. The OT modules are grouped into sectors above and below the plane $y = 0$. The structure along z in a station is of four double-layers (each with staggered straws) with wire orientation X, U, V, X respectively, where X denotes vertical wires (measuring the x coordinate) and U, V denote stereo wires with a $\pm 5°$ angle relative to the X layers. See Ref. [1] for more detail.

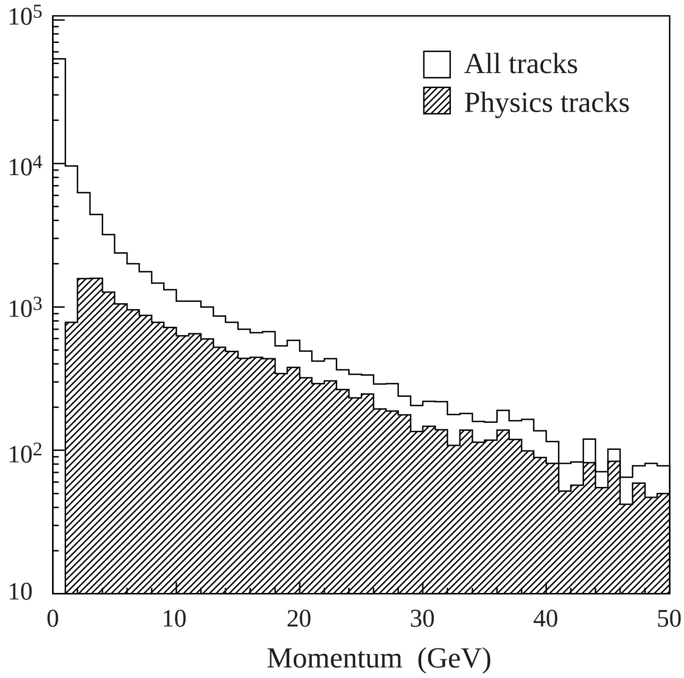

Figure 3: Momentum distribution of tracks that give hits in the seeding stations: for all tracks (open histogram), and for the physics quality tracks (hatched).

The Inner Tracker is made up of silicon wafers, individually described in the simulation. However, for the purpose of the algorithm they are grouped together into L-shaped sectors, two per station, on the left- and right-hand sides of the beam pipe. The boundary between inner and outer tracker is a rectangular shape $60 \times 40\,\mathrm{cm}^2$ $(x \times y)$ centred on the beam axis, in this simulation, as illustrated in Fig. 2. The latest modification to the geometry, with a cross-shaped boundary, has not yet been implemented. In principle, that change should reduce the occupancy somewhat in the Outer Tracker, and should lead to an improvement in the seeding performance. The Inner Tracker strip pitch is $235\,\mu$m, and the resolution $90\,\mu$m. Each station has the same X, U, V, X structure as the Outer Tracker (but with single layers rather than double).

Each OT station has a radiation length of 2% X_0 in this simulation, and the beam pipe is the baseline Al/Be alloy. Pile-up is accounted for, at the nominal luminosity of $2 \times 10^{32}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$. Spill-over is not switched on for the performance study presented below: it will primarily affect the Outer Tracker due to the 50 ns time window required for the read out, corresponding to two bunch crossings. This will give out-of-time hits that in principle could be recognised by their anomalous reconstructed drift-time pattern along the out-of-time track, but this analysis has not yet been performed. Just treating the spill-over hits as background, their effect is expected to be similar to that of pile-up, a reduction of seeding efficiency at the percent level.

The data-set used here is 500 $\mathrm{B}^0 \to \pi^+\pi^-$ events, passed through the full GEANT simulation, generated using SICBMC version 245 [5]. All tracks in the events are studied,

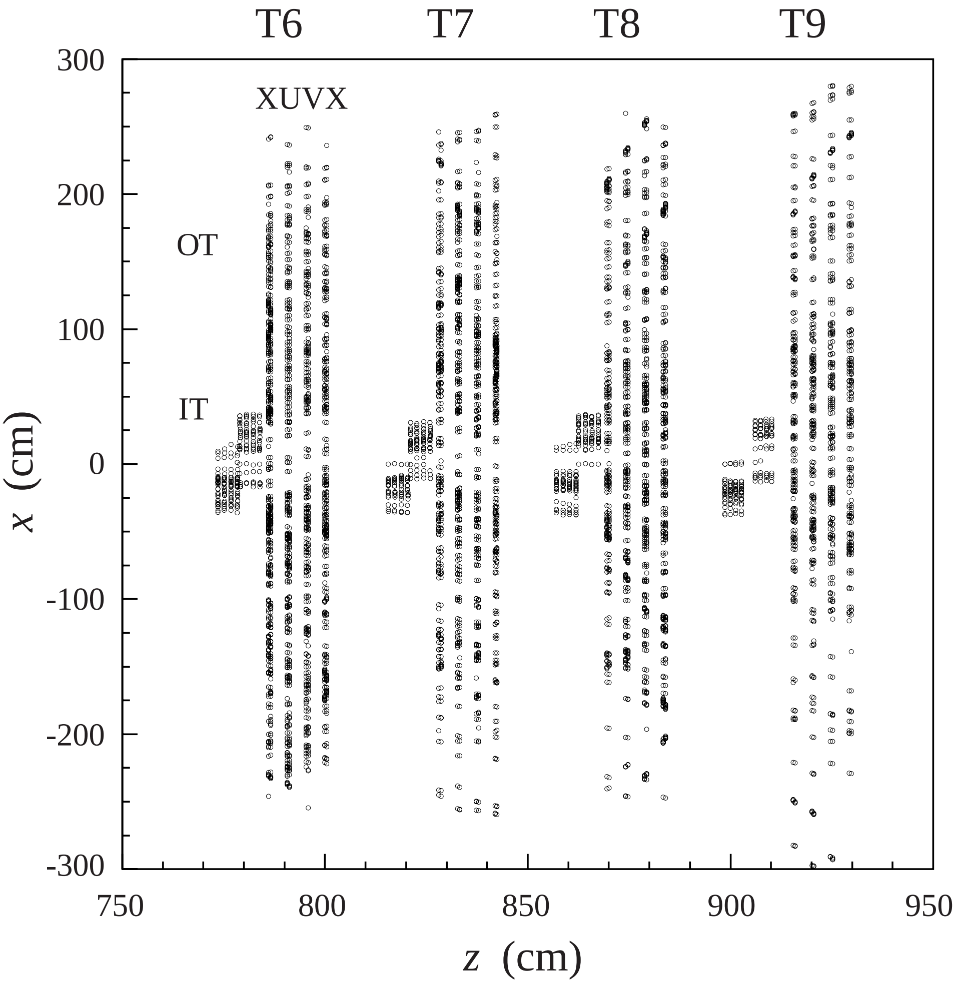

Figure 4: Hits in the seeding stations for a typical event, in the (x, z) projection. The hits in the stereo layers (the second and third layer in each group of four) are shown at the midpoint of the wire (OT) or strip (IT) that is hit. The separation along z of the two IT sectors in each station is evident; the slight difference in positioning for the IT layers of T9 is due to the provision of extra Y layers in that station in this simulation, which are not used here.

so little dependence is expected on the b-decay mode that is chosen. "Physics-quality" tracks are selected as originating from the vertex detector and passing through both T6 and T9; on average there are 28 such tracks per event, compared to a total track multiplicity of about 150 (giving at least one hit in a seeding station). The momentum distribution for these tracks is shown in Fig. 3; the physics tracks all have momentum greater than 1 GeV due to their requirement of passing through the magnet. The hits in the seeding stations for a typical event are shown in Fig. 4.

3 Seeding algorithm

The seeding algorithm starts with a search for track stubs in each station. These are then linked together to form chains of stubs, that are fitted with a parabola in the (x, z) projection to form seed candidates. The hits in the stereo layers are then used to search for a straight line in the (y, z) projection, to complete the seed tracks. These steps of the algorithm are described in more detail in this section.

3.1 Stub search

The X, U, V, X layout of layers within each station gives a pair of X layers (or double-layers in the case of the Outer Tracker) which are at either end of the station, separated by about 5 cm (14 cm) for the IT (OT). They give a reasonable lever-arm to form a stub, i.e. a straight-line track segment in the station. The stub search therefore starts in the (x, z) projection, joining pairs of hits from the X layers of the station, and requiring that the magnitude of the slope of the line joining them is less than 1.0 rad. This stub search is performed separately in each sector of the tracking stations: the sectors are defined by splitting each station into upper and lower halves (relative to the beam axis) for the OT, and into left and right L-shaped sectors for the IT; there are thus 16 sectors in total (two OT and two IT for each of the four seeding stations). Stubs have to be found within a given sector: if the track crosses between two sectors at that station, the stub finding will be less efficient there, but this is a small effect.[4]

The resulting stub candidate has to be validated using other hits, and the validation is applied differently for the OT and IT. For the OT, due to the double-layer structure, a track giving a hit in each layer would have four X hits in each station. The stub candidate is validated by requiring at least one further X hit lies close to the line joining the first pair (within 0.8 mm). For this, the left/right (L/R) ambiguity is taken into account: since for the straw chamber only the distance of the hit from the wire is reconstructed, for a given track direction this gives an ambiguity as to whether the true hit is on the left- or right-hand side of the wire. Both ambiguities are taken for each hit in the search for stub candidates, but the requirement that the third hit should be close to the line joining the first two resolves the ambiguity in most cases. This is illustrated in Fig. 5.

For the Inner Tracker this approach to stub validation cannot be applied, as the layers are single and therefore only two X hits are expected from a track in the station. Instead, the stereo layers U and V are used. From the interpolated x coordinate of the stub candidate at each stereo layer, the stereo hits in that layer can be converted to y measurements, under the assumption that they originated from the track that produced the X hits on the stub. For example, the u coordinate is given by

$$u = x \cos\theta + y \sin\theta \; , \tag{1}$$

[4]Tracks rarely cross between lower and upper OT sectors in this region, so the only noticeable effect is a small decrease of efficiency in the cross-over region between IT and OT, to be weighed against the gain of a large reduction in combinatorics.

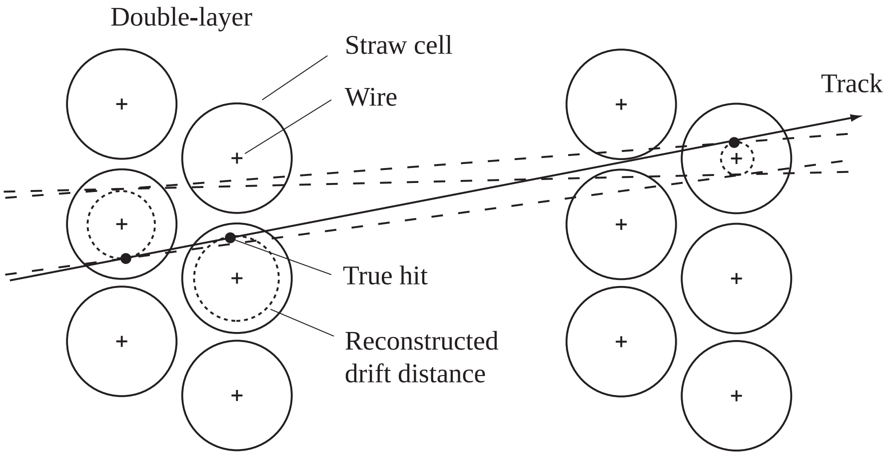

Figure 5: Schematic view of the two double-layers of straw cells in the X layers of an Outer Tracker station (not to scale); the reconstructed drift distances are indicated by dotted circles around the wires which are hit by a track. Candidate tracks that could be constructed from the left-right ambiguities of the hits in the first and last layer are shown as dashed lines; the hit in the second layer resolves the ambiguities.

where $\theta = 5°$ is the stereo angle of the U-layer strips, and hence

$$y = \frac{u}{\sin\theta} - \frac{x}{\tan\theta} \ , \tag{2}$$

where x is the interpolated coordinate at that layer. The resulting y coordinate is required to lie within the physical boundaries of the IT sector concerned, and a pair of such stereo hits is required to be found from the U and V layers respectively, with a cut applied on the corresponding magnitude of the slope in the (y, z) projection: it should be less than 0.3 rad.

At present, this information from the stereo layers of the IT is simply used to validate the stub found in the (x, z) projection, and discarded after validation has been successful. In principle, it could be used to provide a fully three-dimensional stub candidate within the station. However, this approach (particularly when applied to the OT) leads to heavy combinatorics, as there are often more than one pair of stereo hits that can validate a stub. Instead, the validated stubs in the (x, z) projection are first linked together between stations, which is a powerful way of selecting true stubs. The stereo information is then returned to after the linking has been performed, allowing a straight line to be searched for in the (y, z) projection for *all* layers in the seeding region together, which helps in pattern recognition. The disadvantage of this approach is that searching for seed candidates first in X and then in the stereo view gives a final efficiency which is a product of the separate efficiencies of the two steps. Perhaps by incorporating the stereo information at the stub level the overall efficiency could be increased: this remains an avenue for future investigation.

The validated stubs are recorded from all sectors, in the IT and OT. All combinations satisfying the stub selection criteria are kept, even if they share hits with other stub candidates. There are typically a few hundred stubs found in each sector. Subsequent stages of the algorithm act on these stubs, combining them to form seed candidates, irrespective of their origin. In this way tracks which pass only through the OT, only through the IT, or cross between them both, are found at the same time.

3.2 Stub linking

The stubs are linked together to form chains of up to four stubs, one from each station. This is achieved by calculating the predicted x coordinate and slope t_x at a given station from the extrapolation of a stub, and then comparing the values of these parameters for all stubs in that station and applying matching cuts. For example, for OT stubs in neighbouring stations, the x coordinate of the stub at one station must be within 4 mm of the predicted position from the stub in the other station, using a straight line extrapolation, and the difference in stub slopes must be less than 30 mrad. Whilst the magnetic field in the seeding region is low, it is not negligible, so tracks are not straight lines in the (x, z) projection (the bend plane). For example, a 1 GeV track deviates from a straight line by ± 2 mm (depending on its charge) between stations T8 and T9, a large effect compared to the tracker resolution. The cuts applied when linking stubs are wide enough to allow for some curvature of the track. The chain of linked stubs is not required to be continuous: if a stub is missed for a track that passed through a given station but is found in the neighbouring stations, they can still be linked.

A least-squares fit of a parabola is made to the chain of linked stubs, fitting directly to the list of hits that have been assigned to the stubs. A search is then made for extra hits that can be assigned to the seed candidate: using the parameters of the fitted parabola, the position of the track candidate is interpolated at each X layer for which a hit had not been found, and the nearest hit in that layer is added to the list if its distance from the track is within a cut. This recuperates hits from stations that failed to satisfy the stub validation criteria, despite belonging to the track (for example, a track through an OT sector that only gave two X hits, instead of the three required for stub formation). A final parabolic fit is made to all assigned hits, for each chain of linked stubs, to form seed candidates.

The next step is to select the best seed candidate, which is done by taking that candidate with the most assigned hits, as long as the χ^2 of its fit is acceptable; the candidate with the lowest χ^2 is chosen if more than one have the same number of hits. Other seed candidates are not permitted to share hits with the selected track, so after the selection has been made all other candidates are checked to see if they share any of the hits on the selected track, and if so they are refitted excluding those hits (unless their total number of hits is reduced below three, in which case the candidate is dropped). The next best candidate is then selected, and this procedure is iterated until no further candidates remain.

This procedure is applied to each chain of linked stubs in turn, starting with the four-link chains, then three- and two-link chains; then the sequence is repeated for a second iteration with looser linking cuts (increased by a factor of two). On average 140 candidates are selected per event.

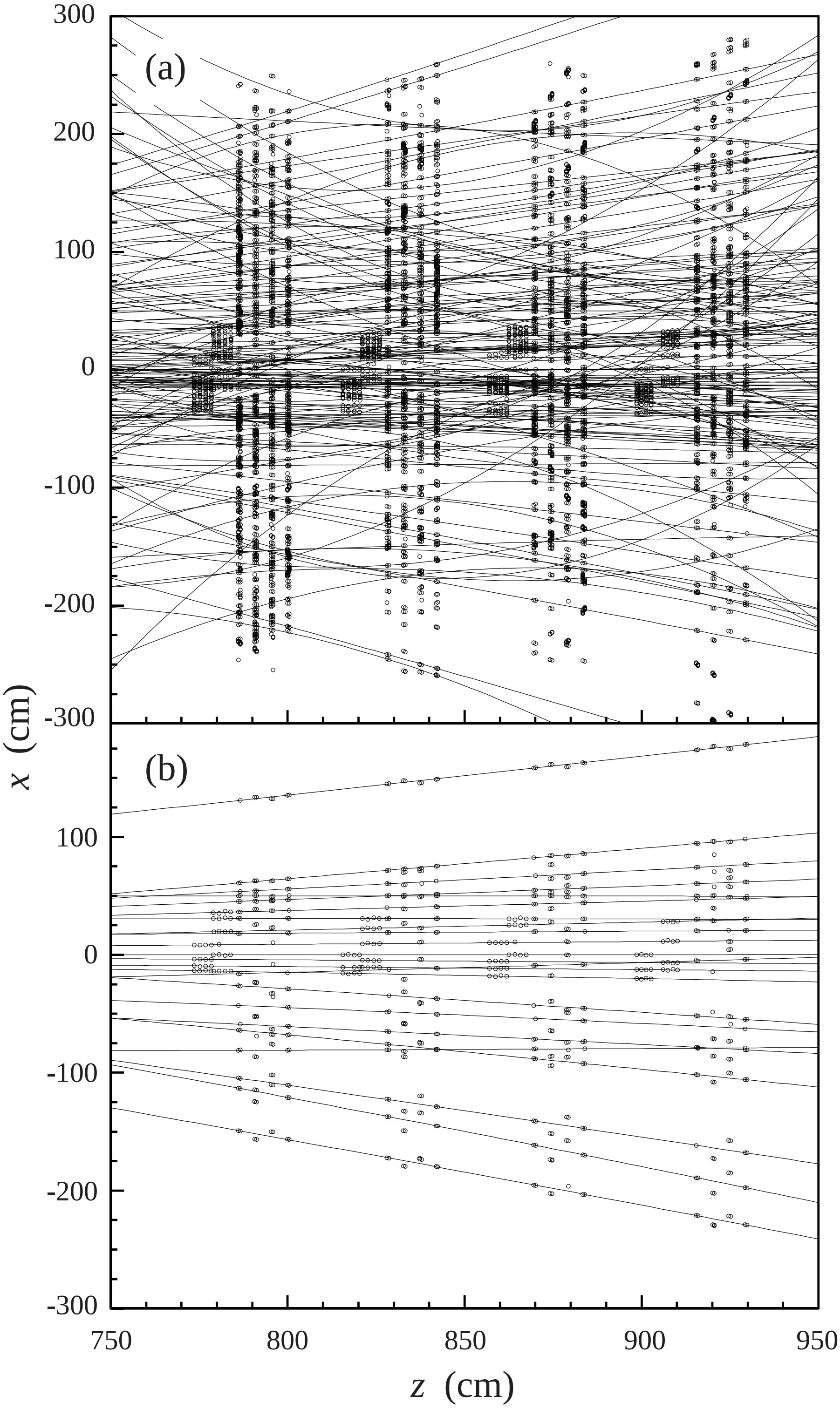

Figure 6: The same event as Fig. 4: (a) with all the found seed candidates superimposed; (b) with only the hits that originated from physics tracks displayed, along with the seeds that are successfully matched to those tracks.

3.3 Stereo search

Finally, the search for hits belonging to the seed candidates is made in the (y, z) projection, using the stereo layers. The hits in the U and V layers are converted to y measurements as described above, on the assumption that they originated from the candidate in question. A straight line is then searched for amongst those hits that are within the physical limits of the sector, taking a pair of hits from the first and last layers and adding hits from intervening layers if they lie within a cut from the interpolated line. A least-squares fit to a straight line is then made to the assigned hits. This procedure is repeated for other pairings of hits, and the line with the most hits and lowest χ^2 is selected, as for the fit in the other projection. By starting at the two extremes of the seeding region for the initial pair of hits, and working inwards, not all pairs have to be tested, and the procedure is reasonably fast. Hits from the selected line are flagged to avoid them being used for later track candidates, with the stereo search being performed starting with the best candidate from the (x, z) projection.

The seed candidates that are found for the event that was shown in Fig. 4 are shown superimposed on Fig. 6 (a). The subset of the hits which originate from physics tracks are shown in Fig. 6 (b), along with the seeds that have been successfully matched to those tracks. The seeds shown here have not been fit to the hits from the physics tracks alone, but rather have been found successfully amongst the rather crowded full event: in this case, all physics tracks were reconstructed. Note that the stereo layer hits can be offset from the track, depending on the position of the track in the other projection: the offset for U and V layers should have equal magnitude but opposite direction. A zoom on the region close to the beam pipe for T6 and T7 of the same event is shown in Fig. 7.

4 Implementation

The tracking software environment TRAIL has been used for these studies, running in the GAUDI framework [1]. The seeding algorithm is written in FORTRAN,[5] called from C^{++} code[6] that fills arrays with information about each hit in the seeding region: (x, y, z) of the centre of the wire or strip that is hit, the stereo angle θ, a measured radius r and its uncertainty σ_r. The radius r stores different information for the Inner and Outer Tracker hits: for the Inner Tracker it is simply the position of the reconstructed hit in the coordinate measured by that layer (i.e. the centre of a cluster when clustering of neighbouring hit strips has been performed); for the Outer Tracker, it is the radius of the reconstructed hit position from the wire. The sign of σ_r is flipped to label Inner Tracker hits.

FORTRAN was used for the algorithm for reasons of efficiency: speed of getting a working prototype, that is, due to my lack of experience with C^{++}. However, there may be an issue for the speed of the eventual algorithm, as the number of nested loops is

[5] `seed.F`, $\sim$ 1800 lines of code (including some comments).

[6] `TrSeedEvent.cpp`

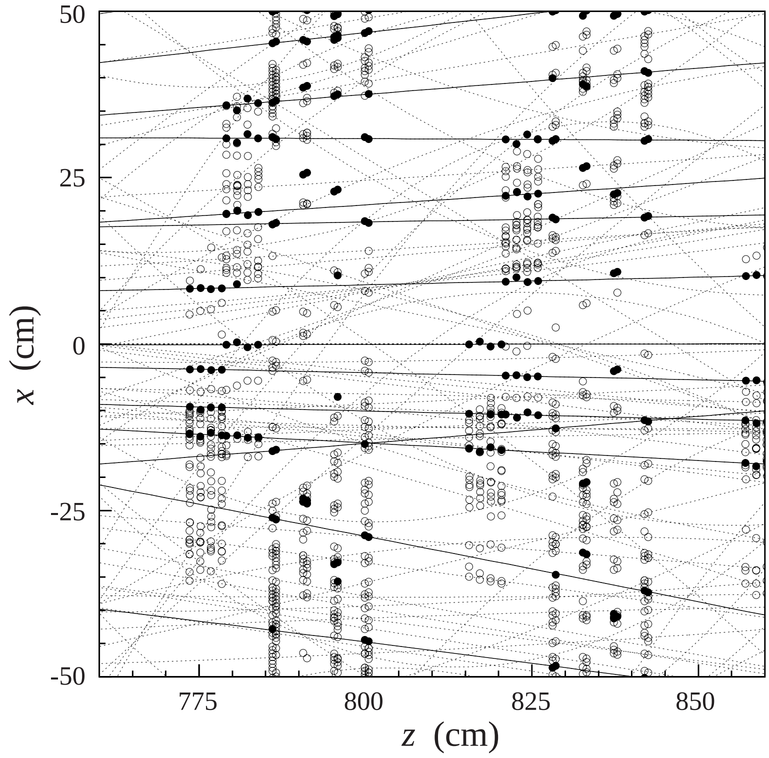

Figure 7: Zoom on the region close to the beam pipe for T6 and T7 of the event in Fig. 6. The full points indicate hits that originate from physics tracks, and the solid lines the seed candidates that have been successfully matched to those tracks; the open points and dotted lines indicate the other hits and seed candidates.

substantial (12 levels of nesting in the OT stub finding, at the last count). Fast access to the hit information, ordered into the logical subdivisions of the tracking system, is crucial. The current algorithm takes about 5 seconds per event, running interactively on LXPLUS.[7] This is no accident, as there is an overhead of about a minute when running a job (due to loading of the magnet field map or whatever), so a test job run on 20 events takes two minutes, a time that I am happy to sit and wait for. Modifications to the algorithm which took an order of magnitude longer would have been too painful when running test jobs, and were therefore reconsidered. Thus the time per event should not be taken too seriously when considering applications in the trigger, for example, where the requirement on processing time is much more stringent. There are surely many tricks available to increase the speed, which have not yet been explored: for example, ordering the hits in each layer with respect to their measured coordinate should give a substantial increase in speed for the search that is frequently made for the nearest hit to a given value of that coordinate.

The geometry of the tracking stations is stored in a separate file, included in sub-

[7]Pentium III processor running at $\sim 600\,$MHz, with corresponding power of ~ 12 SI95 units.

routines as required.[8] In particular the positions of the layers along z are listed: this information is used to classify the hits into suitable subdivisions of the detector, for the stub searches to be performed on. The relevant information is currently stored in various places (CDF files and inline code) for the different subdetectors, and it simplifies life to have local access to them—but at the cost of having to modify that geometry file to follow any changes made in the simulation. Eventually a centrally-located store of all such geometry information would be better.

The output from the seeding algorithm is a list of seed candidates. For each, an array of parameters is provided, including the track state at a given z coordinate: $(x, t_x, q/p, y, t_y)$, where t_x and t_y are the slopes in the (x, z) and (y, z) projections, and q/p is the charge divided by momentum of the candidate, estimated from the measured curvature. The covariance matrix for these parameters is also returned, constructed from the covariance matrices of the separate fits in the (x, z) and (y, z) projections, assuming that they are uncorrelated: it is a 5×5 matrix of form

$$\begin{pmatrix} C_x & 0 \\ 0 & C_y \end{pmatrix} \tag{3}$$

where C_x is the 3×3 covariance matrix from the (x, z) parabolic fit, and C_y is the 2×2 matrix from the (y, z) linear fit. Finally a list is provided of the hits that have been assigned to each candidate. These values are returned to the C^{++} steering routine, which then creates track candidate objects for use by later algorithms (such as the track following).

A separate "monitoring" algorithm has been created (again in FORTRAN)[9] to enable the performance of the seeding algorithm to be checked. This simple routine is passed the truth information for the tracks in the event (including the list of hits that truly belonged to each track in the simulation), along with the list of hit assignments returned by the seeding algorithm. A comparison of these lists allows the performance to be quantified.

5 Performance

The seed candidates are matched to true tracks by comparing the hits that they share (ignoring the L/R ambiguity for OT hits). The number of correct X hits on the seed candidate that shares the most of the true track's hits is shown in Fig. 8, versus (a) the number of hits on the true track and (b) the number of hits on the reconstructed track; the same distributions are shown for the stereo hits in (c) and (d).

The track is defined to be successfully found if greater than 70% of the hits on a seed candidate originate from that track. For this stage of the pattern recognition, no requirement is made on the "hit efficiency", i.e. that the seed candidate should also have more than a certain fraction of the total number of hits from the track: such a requirement is applied later after the seed candidate has been followed through the full tracking

[8] `seed_geometry.inc`
[9] `seed_monitor.F`

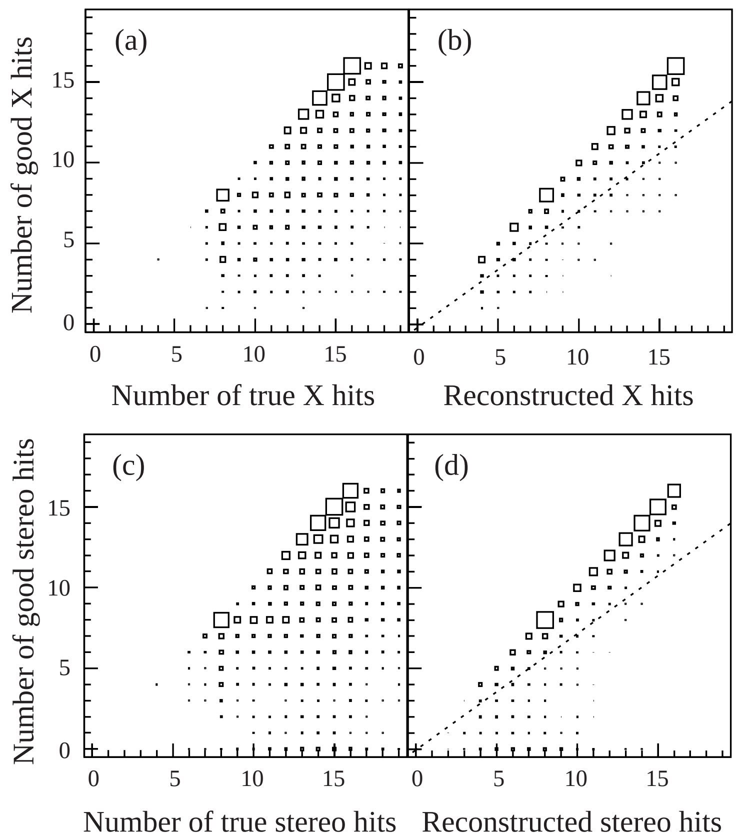

Figure 8: (a) Number of good X hits on a seed candidate (i.e. the number of hits in the (x, z) projection that truly came from the track that is best matched to the seed) versus the true number of X hits on that track. (b) Number of good X hits on a seed candidate versus the total number of X hits assigned to that seed candidate: the requirement of $> 70\%$ hit purity for a track to be successfully reconstructed is indicated by the dotted line. (c, d) The same for the stereo projection. These plots are made for physics quality tracks, and the bin content is proportional to the area of the box.

system [2]. The important point for the seed is that is should accurately reproduce the track's parameters, but need not include all of its hits as any that have been missed can be picked up later. To ensure that the track is found successfully in both projections, the 70% requirement is applied to both X and stereo hits separately.

With this definition, the reconstruction efficiency for "physics quality" tracks is 95% for the algorithm described in this note. The dependence of the efficiency on momentum is shown in Fig. 9: it is largely independent of momentum, apart from dropping significantly for tracks with $p < 3$ GeV. The efficiency for the 20 (4) tracks per event passing only through the OT (IT) is 95% (97%); the 4 tracks per event that cross over between the two tracking detectors have a lower efficiency of 93%. This accounts for the residual variation with momentum, as the slightly lower efficiency around 10 GeV is in the region most populated by cross-over tracks.

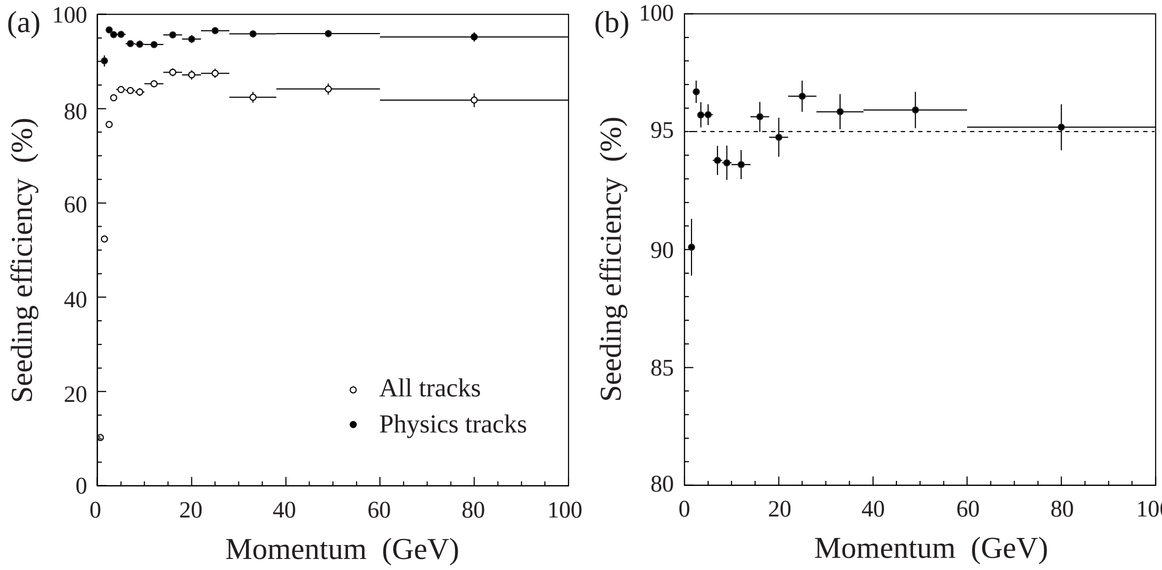

Figure 9: Efficiency of seed finding as a function of momentum: (a) for all tracks (open points) and physics tracks (full points); (b) for physics tracks on an expanded scale.

The ghost rate—the number of reconstructed seeds that fail the requirements for successful matching to a track, divided by the total number of tracks—is about 18% for the full track sample. A relevant definition of ghost rate for the physics track sample is the number of reconstructed seeds that are best matched to a physics track, but fail the requirements for successful matching, divided by the total number of physics tracks. This is 14%, but can be reduced to 10% without loss of efficiency by requiring that the total number of hits on the reconstructed seed is greater than seven. The distribution of this total number of hits is significantly different for ghost tracks compared to successfully matched seeds, as shown in Fig. 10 (a) and (b), so if a lower ghost rate is required for a physics analysis this can easily be achieved by tightening the cut on the total number of hits, with only modest loss of efficiency: see Fig. 10 (c). The algorithm has been developed focussing more on efficiency than ghost rate, trying to achieve the highest efficiency for physics tracks in a bearable amount of processing time. If low ghost rate is considered important, more sophisticated requirements could be applied to achieve it.

There is a third class of reconstructed seeds, those which pass the requirements for successful matching to a track, but are not the *best* matched seed to that track (where best is defined as the seed that shares the most hits with the track). Such seeds are referred to as clones, as their track parameters will be very close to that of the seed that is best matched to the track: for example, a track with 16 hits in the seeding region might be reconstructed as two separate segments, one with 10 hits and a clone with 6 hits. Another source of clones is a modification of the algorithm that was made after noticing that the efficiency of IT only tracks is higher if the seeding algorithm is run on the IT hits alone: this is therefore provisionally implemented as a first pass, before running the algorithm on both IT and OT hits together. A dedicated study could be made to remove copies of

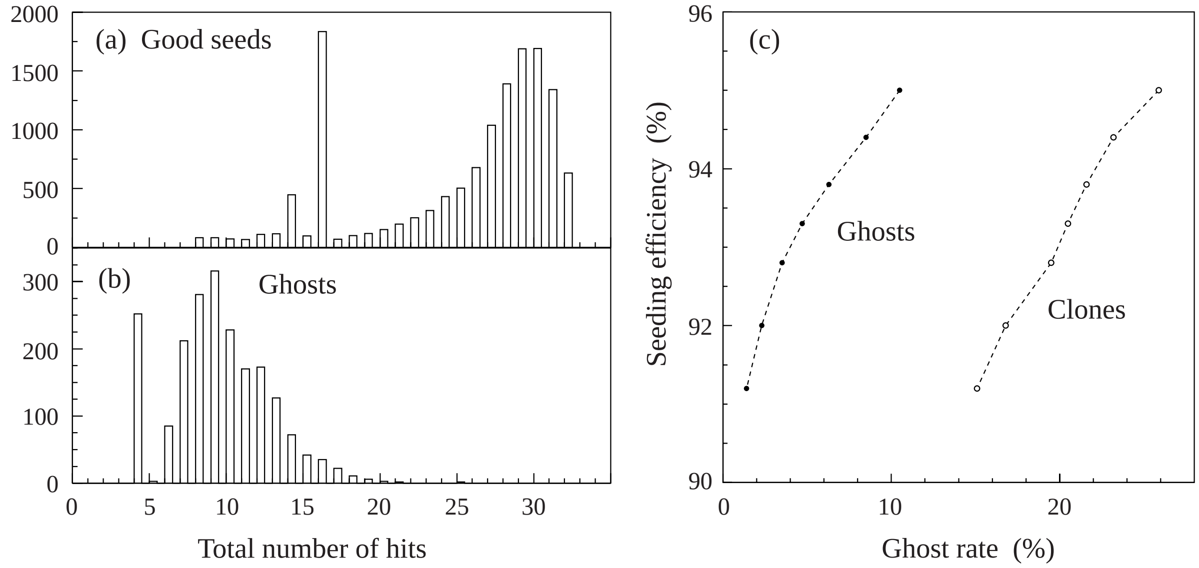

Figure 10: (a) Total number of hits (X plus stereo) on successfully matched seed candidates: the highly populated bin at 16 hits is from IT tracks, the peak around 30 from OT tracks; (b) the same for ghost tracks. (c) Variation of the seeding efficiency and ghost rate (full points) as the cut on the number of hits on the track is varied: the point with highest efficiency is for > 7 hits, and each successive point represents an increase in this requirement by one hit; the clone rate is also shown (open points). These plots are made for physics quality tracks.

seeds found in both passes, but this has not yet been done. The current clone rate is 26%, reducing with the cut on the number of total hits on the seed, as shown in Fig. 10 (c), although not as dramatically as the ghost rate.

The dependence of the efficiency on the hit multiplicity has been studied, i.e. the dependence on the total number of hits in the tracker. The result is shown in Fig. 11, separately for the IT and OT, for tracks that pass only through the detector considered. As can be seen there is a clear linear dependence for both detectors. In the limit of low multiplicity the efficiency is compatible with 100% for both, and the decrease with increasing multiplicity has a slope of 1.2% per 1000 hits in the OT and 0.4% per 1000 hits in the IT. For very high multiplicities there is some indication that the decrease of efficiency becomes more severe than a simple linear behaviour.

For tracks which have not been successfully matched to a seed candidate, the impact point in the seeding region is shown in Fig. 12. As can be seen, a significant fraction (about half) of the missed tracks pass close to the boundary between upper and lower OT sectors. This is understood, due to the structure of the stereo modules in this region: they have the same rectangular shape as X modules, but tilted by the stereo angle of 5° (see Fig. 2). As a result, a "corner" of the stereo module projects into the other side of the beam pipe ($y > 0$ for modules below the pipe, and vice versa). Since the search for seed stubs is only made in the sectors above and below the pipe separately, this leads to

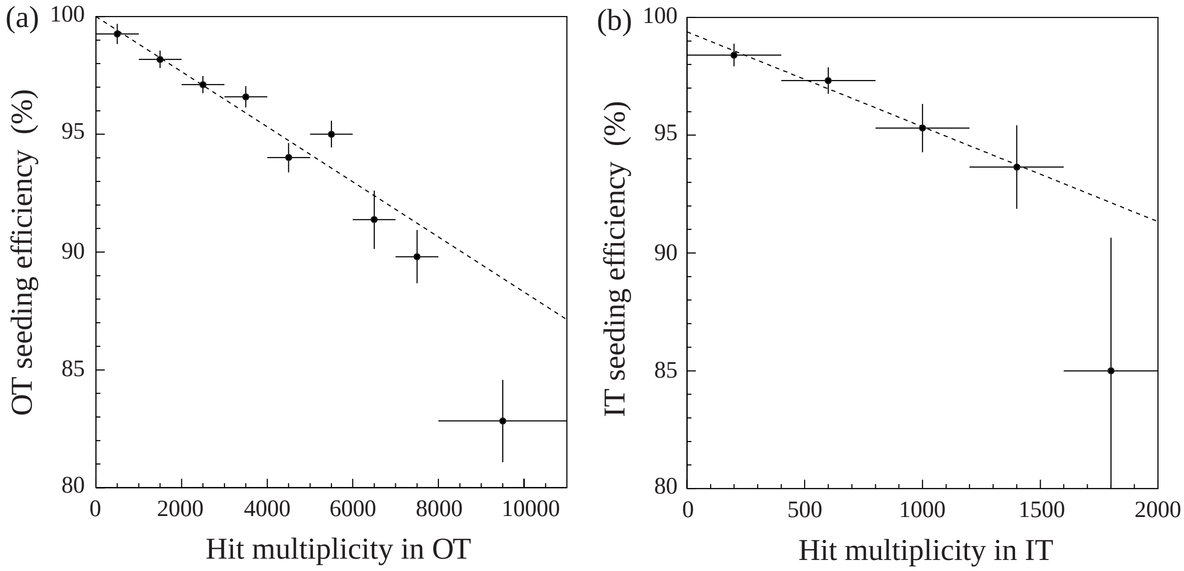

Figure 11: Efficiency of seed finding as a function of total hit multiplicity: (a) for tracks passing through the Outer Tracker only, versus the number of hits in the OT; (b) for tracks passing through the Inner Tracker only, versus the number of hits in the IT.

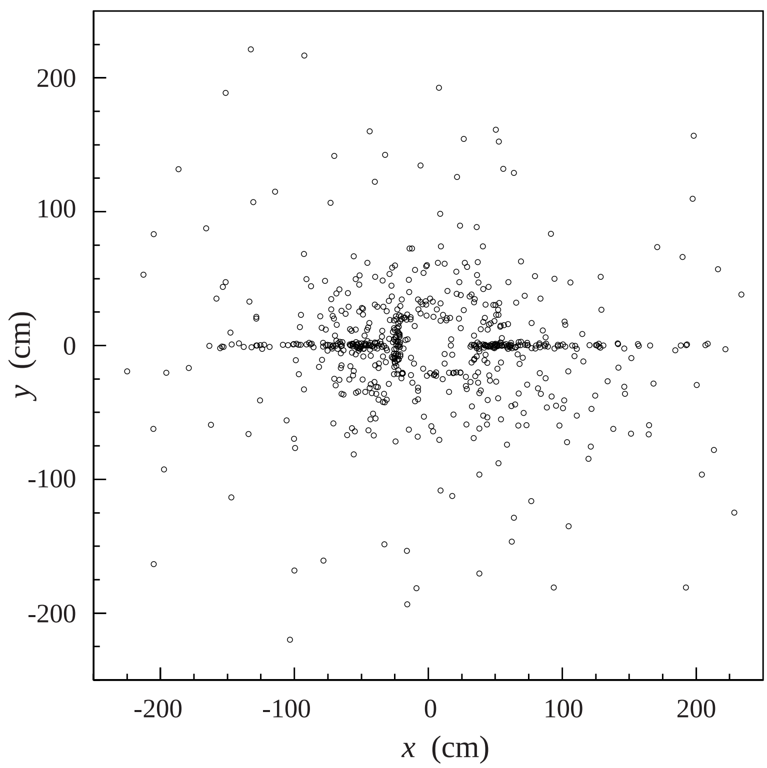

Figure 12: Impact point of true tracks that have not been successfully matched to seed candidates, in the transverse plane at z-coordinate in the middle of the seeding region.

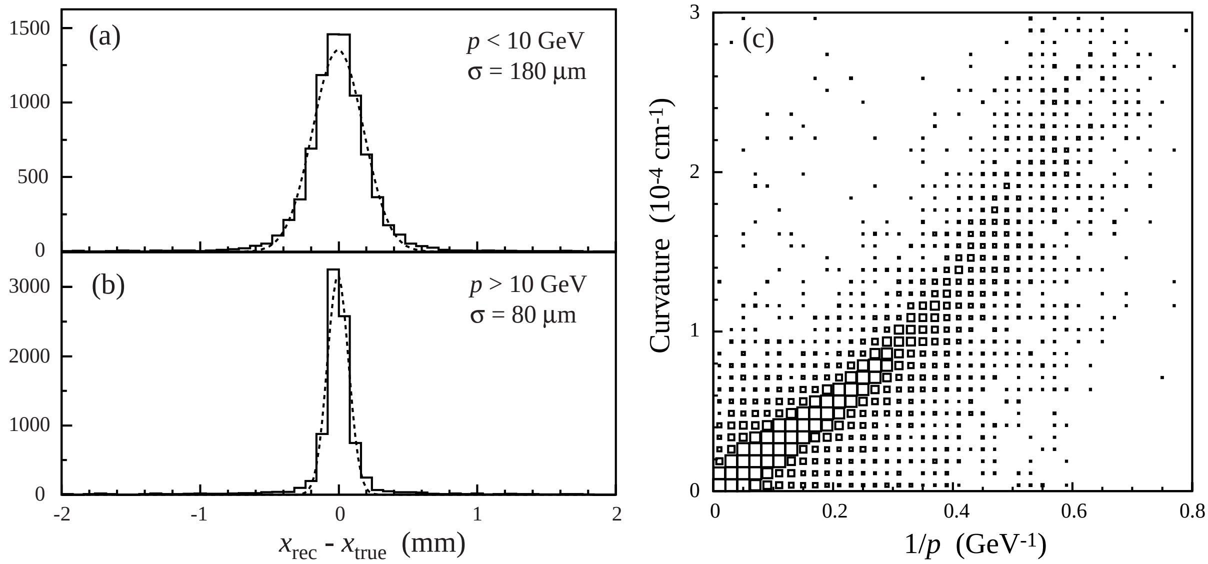

Figure 13: (a) Difference between the reconstructed and true x coordinate for momentum $p < 10\,\text{GeV}$, (b) $p > 10\,\text{GeV}$; the sigma of a Gaussian fit to each distribution is indicated. (c) Reconstructed magnitude of the curvature of seed candidates versus their true inverse momentum.

Momentum	x	y	t_x	t_y
$<$ 10 GeV	180 μm	1.8 mm	1.0 mrad	2.2 mrad
$>$ 10 GeV	80 μm	0.8 mm	0.4 mrad	0.9 mrad

Table 1: Resolution on the track parameters of the reconstructed seeds, at $z = 930\,\text{cm}$, for two bins in momentum.

increased inefficiency in the overlap region.

The resolution measured for the x coordinate of tracks, at the exit from T9, is shown in Fig. 13. The resolution improves with momentum, so is shown for two ranges in (a) and (b). The corresponding resolutions for this and the other track parameters are given in Table 1. The curvature parameter that is measured in the parabolic fit to the seed candidates is closely related to the remaining track parameter, the inverse momentum, as shown in Fig. 13 (c). After suitable calibration, a core resolution of $\Delta p/p \sim 20\%$ can be extracted.

6 Conclusion

An algorithm for pattern recognition in the seeding stations of LHCb has been described. Without much tuning, it provides an efficiency for recognising "physics quality" tracks (that originate from the vertex region) of about 95%, with little dependence on momentum. The "clone" rate for these candidates (i.e. seeds that are matched to the same true track) is about 26%, but should easily be reduced by a second step comparing the track parameters of the seeds that have been found. The "ghost" rate (i.e. seeds that are not matched to a true track) is about 10%. This could be further reduced if necessary by track quality cuts, at some cost to the efficiency.

The separate efficiencies for tracks passing through the Outer Tracker alone, the Inner Tracker alone, or crossing between OT and IT, are 95%, 97% and 93% respectively. Some causes of the remaining inefficiency have been identified, such as the treatment of the stereo layers for tracks close to the boundary between upper and lower OT sectors. More careful treatment of tracks passing close to the boundary regions between sectors should increase the overall efficiency for physics tracks to 97% or higher, but each further improvement comes at the cost of increasing complexity in the algorithm. The efficiency for *all* tracks (including those which do not originate from the vertex region) is about 85% for $p > 3\,\mathrm{GeV}$, falling for lower momenta. As the cuts have been set concentrating on physics tracks, this efficiency could be improved by a second pass, with adjusted cuts (for example, allowing tracks with larger angles).

A significant dependence is found of the seeding efficiency with hit multiplicity in the tracking detectors. If the rate of background hits from secondary tracks could be reduced, by reduction of the material that physics tracks pass through before reaching the seeding region, there would be a clear benefit for the seeding performance.

Acknowledgements

It is a pleasure to thank Matthew Needham for his help in setting up the environment in which I could run the algorithm described here; he also generated much of the data used in the tracking studies. Thanks also to the other members of the Tracking and Particle ID Task Forces for feedback.

References

[1] LHCb Outer Tracker Technical Design Report, CERN/LHCC 2001–024, 14 September 2001.

[2] R. Hierck, "Track following in LHCb", LHCb 2001–112.

[3] LHCb Vertex Locator Technical Design Report, CERN/LHCC 2001–011, 31 May 2001.

[4] J. van Tilburg, "Matching VELO tracks with seeding tracks", LHCb 2001–103.

[5] `http://lhcb-comp.web.cern.ch/lhcb-comp/SICB/html/sicbug.html`