LHCb note 2007-022

Standalone Track Reconstruction in the T-stations

R. Forty and M. Needham
CERN

March 21, 2007

Abstract

An algorithm for fast and efficient tracking in the T-stations is described together with its performance in the DC06 data challenge. An efficiency of 94–95 % is achieved for tracks with momenta above 2 GeV for a ghost rate of 13 %.

1 Introduction

In this note the performance of an algorithm to reconstruct the trajectories of particles traversing the tracking stations located after the LHCb spectrometer magnet is described. From previous studies this is known to be a challenging task for several reasons [1, 2]. These stations are located in the fringe field of the magnet. Consequently, the trajectory cannot be assumed to be linear in the (x,z) projection. Instead a more complicated parabolic track model must be used. In addition, many low energy secondary particles are generated in secondary interactions within the detector. This results in events where the detector occupancy in some regions is 40 % or more. Therefore, care has to be taken to ensure that the computation time does not become prohibitively high.

The seeding algorithm described here is a development of a FORTRAN based algorithm described in an earlier note [2]. That was a stub-based algorithm, i.e. as a first step short track segments referred to as stubs were searched for in each tracking station. The structure of the stations is (x,u,v,x), where x represents a tracking layer with approximately vertical detection elements and u,v are layers with $\theta = \pm 5^\circ$ stereo angle. For the outer tracker these are double-layers of straws, while for the inner tracker they are single layers of silicon strips. A stub therefore consists of a minimum of four hits, one in each (double-)layer, giving a precise measurement of the x-coordinate of the track, and its slope in the (x,z) projection, plus a less precise measurement of its y-coordinate (since the precision is reduced by $1/\sin\theta \sim 11.5$) and its slope in the (y,z) projection (for which the precision is in addition reduced by the limited lever arm between the u and v layers, compared to the x layers). The original algorithm searched for such track stubs in the Inner and Outer Tracker stations, and then linked them together to form track candidates, by placing cuts on the distance in space and angle of the stubs. This worked well in the spectrometer of that vintage, i.e. LHCb-classic rather than LHCb-light. However, during the reoptimization [3] the number of tracking stations in the seeding region after the magnet was reduced from four to three. This reduced the redundancy in the stub-based approach, with the result that it was difficult to maintain an efficiency of ~ 95 % or better.

The problem is that in the high occupancy environment of the Outer Tracker, there is a non-negligible probability of missing a hit in one of the layers required for stub reconstruction due to the single hit electronics and detector dead-time. With the current detector layout with three seeding stations, this leads to about 85 % efficiency for finding long tracks, if they are required to have reconstructed stubs in at least two stations. The efficiency can be pushed up to over 95 % if one accepts track candidates with only one stub, but the ghost rate for such candidates is unacceptable. One is therefore driven toward an algorithm where single stub candidates are verified by extrapolating them into the neighbouring stations to look for additional hits. This suggests that it would be better to treat the three stations as a whole, from the start, searching for track candidates according to their hit distribution throughout the twelve tracking layers of the full tracker. This approach is referred to as working in projection, as the x layers are searched first to find an (x,z) projection of the track, before converting the stereo layer information to y-coordinate measurements and searching for the (y,z) projection. This forms the core of the algorithm that is described in this note.

However, although moving to a global, projection-based approach can achieve higher efficiency, it comes with at a cost: the attractive feature of the stub approach that the stations can be searched independently is lost. In particular, when the stations have not been fully aligned at the start of data taking, it will probably be easier to find track segments in individual stations rather than in the full tracker. For this reason, a stub-based approach can still have some application, in particular for alignment studies where the highest efficiency is not so important. Also, for specific final states such as the search for K_S decay products, the flexibility offered by a stub-based approach may be important. For now, the stub-based approach is maintained for a second pass, to pick up tracks that cross over between Inner and Outer Tracker. Following the first pass of projection-based search, the unused hits in the Inner Tracker are used to form stubs. They are then linked together, and any left over are extrapolated into the Outer Tracker to add hits. The stub-based part of the algorithm is also described in more detail below.

2 Algorithm Description

The algorithm proceeds as follows:

Data Preparation The input data is picked up from the transient event store and converted to working objects. During this phase hot-spots within the detector where the local occupancy is very high are searched for and removed.

Projection Search A projection based search is made for track candidates in both the Inner and Outer Tracker.

Stub Search Unused hits in the Inner Tracker are used to build space-points. These are then linked to make track candidates. Finally, any remaining unused space-points are extended into the Outer Tracker.

Output The final list of track candidates is converted to instances of the standard LHCb **Track** class [4] and registered in the transient event store.

These steps are described in more detail in the following sections.

2.1 Data preparation

In previous studies [5] it has been observed that there are localized regions in the T stations where the fraction of hit channels is high. Such ‘hot-spots’ give rise to combinatorial problems. Therefore, in the first stage of the algorithm such regions are searched for and the corresponding hits discarded.

In the case of the Outer Tracker two types of hot-spots are removed. The first topology is steep or curling tracks that give rise to strings of adjacent hit straws (Fig. 1). Such a topology is identified and removed by a simple clustering algorithm. Currently, if more than six adjacent straws are hit they are discarded.

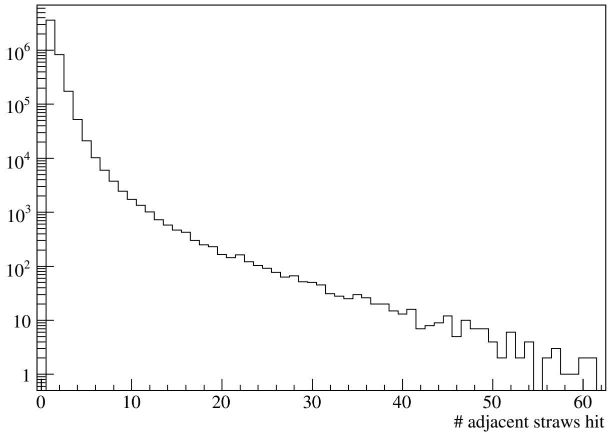

Figure 1: Number of adjacent Outer Tracker straws hit.

The second topology removed is modules where the fraction of straws hits is large (Fig. 2). Hits from modules where the occupancy is more than 40% are discarded.

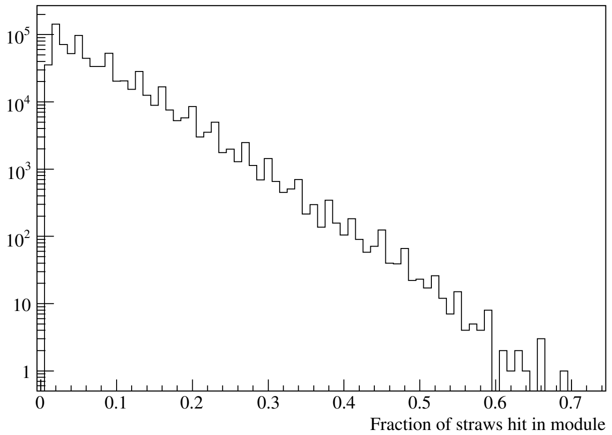

Figure 2: Outer Tracker module occupancy.

Such hot-spots also occur in the Inner Tracker. However, in this case the impact on the pattern recognition is less pronounced due to the higher granularity of the detector. If more than 32 strips in a front-end chip are above threshold the corresponding clusters are removed. These cuts remove 10 % of the total T-station data but increase the speed of the algorithm by 50 % without effecting the reconstruction efficiency.

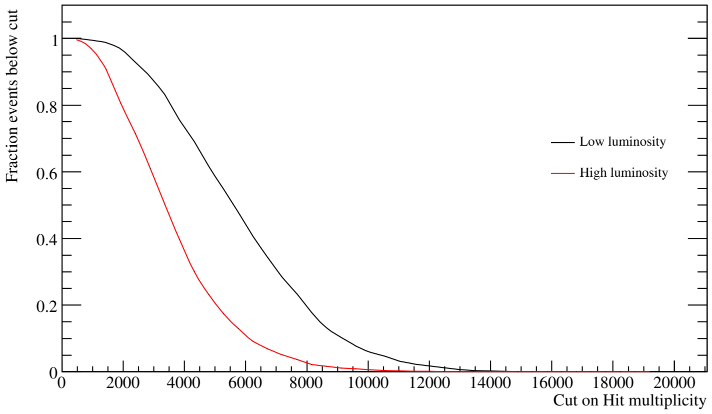

Figure 3: Fraction of events removed by a cut on the total hit multiplicity in the T-stations. Low luminosity corresponds to 2×10^{32} cm^{-2}s^{-1} and high luminosity to 5×10^{32} cm^{-2}s^{-1}.

Finally, it is chosen to apply a cut on the maximum hit multiplicity. If the total number of hits in the T-stations exceeds 10000 then the seeding algorithm is not executed. From Fig. 3 it can be seen that at a luminosity of 2×10^{32} cm^{-2}s^{-1} this cut removes 0.5 % of events. At higher luminosities the effect of this cut becomes more pronounced: at 5×10^{32} cm^{-2}s^{-1} ∼6 % of events are rejected.

2.2 Projection Search

The projection search is performed using hits in each of five logical "sectors" of the T-stations, as illustrated in Fig. 4. These are defined as the upper and lower halves of the Outer Tracker, the upper and lower boxes of the Inner Tracker, and finally a sector which combines hits from the left and right boxes of the Inner Tracker. The tracking layers from all three stations are combined in each of these sectors. Since the magnetic field is predominantly vertical, tracks tend to be swept from side to side, i.e. along x but not along y. Tracks rarely cross from the upper half to the lower half of the OT, or the upper box to the lower box of the IT. Therefore, the search is made separately in each of those sectors, to reduce the combinatorics. Since the IT boxes have some overlap both between each other, and with the OT halves, there are some tracks which cross between sectors, which will be particularly useful for alignment purposes. Those that are not found by the projection search, because they leave insufficient hits in a single sector, are picked up by the stub search described in the next section.

Within each sector, the projection search starts with the x hits from all three stations,

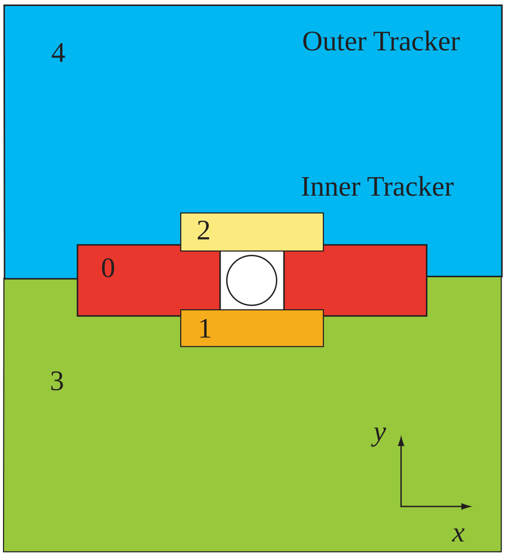

Figure 4: Layout of the tracking sectors in the OT and IT (not to scale).

searching for parabolic trajectories in the (x,z) projection. In this and the following sections this will be described for the OT; a similar approach is taken for the IT although it is simplified somewhat by the lack of left-right ambiguity: the drift time in the OT straws measures a radius from the wire, which when coupled to an assumed direction of the track leads to an ambiguity as to whether the track passed on the left or right side of the wire; this effect is not present in the silicon IT. The trajectory candidates are selected by a series of steps, first taking pairs of hits to define a road, then selecting a third hit to define a parabola, and so on. The philosophy that is followed is to apply relatively loose cuts at each of these steps in order to maintain high efficiency. However, applying cuts at each step, as more information is added, helps to reduce the large combinatorial background. At the end of the algorithm (after the stereo search has been made) a likelihood is calculated for each remaining candidate, which acts as a discriminant between signal (long tracks) and background (ghosts). Seed candidates are selected for output in order of decreasing likelihood, and after each one has been selected its clone candidates are killed. This continues down to a limit on the likelihood value, which can then be adjusted to give the optimal compromise between high efficiency and high purity (low ghost rate). Although the same structure is used for the IT search, a different set of cuts is applied for its sectors. The cuts have been tuned to find long (i.e. tracks that give hits in the VELO as well as the T stations) and downstream tracks, but the algorithm also finds short tracks, typically secondaries produced from the interaction of primary tracks in the spectrometer material, with a reasonable efficiency. These can be useful in the pattern recognition for RICH2.

X search

The X search in the OT starts by filling a vector with the x hits[1] in each double layer, and sorting them in order of increasing x coordinate (defined by the wire position, at the middle of the wire). Pairs of hits in the first and last station are used to define a straight line in (x,z)[2], and the pair is kept if the slope of that line s_x satisfies $|s_x| < 0.8$, and $|x/3125\,\mathrm{mm} - s_x| < 0.7$ (to take into account the correlation between slope and x observed for long tracks). A cut is also made on the p_T kick angle (discussed below): $|\Delta\theta| < 0.7$, where $\Delta\theta = \tan^{-1} s_x - \tan^{-1}(x_1 - s_x(z_1 - z_0)/z_0)$, the coordinates of the first hit is (x_1,z_1), and $z_0 = 5.3\,\mathrm{m}$ is the approximate centre of the magnet. All hits within the window of $\pm 10\,\mathrm{mm}$ around that line candidate are selected for further study, with a requirement that there should be at least five hits found (in addition to the two that formed the line candidate) out of ten that would expected for a track passing through the full set of OT x layers, if they all give a hit (see Fig. 5 (a)). At least one of those hits must be found in station T2. At this point the drift time has not been taken into account, so the window size is of the order of the drift cell size (5 mm straws). Some care has been taken to optimize the code for this part of the X search, as it is called many times: in particular, the ordering of hits in their vectors is profited from, by only looking at the neighbouring hits to the ones that defined the line.

Next a hit is chosen from amongst those selected in T2, and the three hits are used to define a parabola. The L/R ambiguity of each of those three hits is then tested, opening a 1 mm window around the parabola, adjusted to pass through the hit when the drift time has been taken into account, taking each ambiguity in turn. The number of other hits in the same station that fall within the new window is counted, amongst those selected as being within the previous window, allowing them to take each L/R ambiguity in turn (see Fig. 5 (b)). The ambiguities that maximize the number of selected hits are retained. This approach reduces the number of L/R combinations to be tested, by neglecting the effect on the parabolic parameterization in other stations when the ambiguity of a hit is changed. A least-squares fit of a parabola is then made to the selected hits, taking into account their chosen ambiguities. The fit is iterated, allowing the L/R ambiguity of each hit to change sign at each iteration, if it moves the point closer to the fitted line; if any point remains more than $3\,\sigma$ from the fitted line after all L/R ambiguities have been optimized, the most outlying point is removed and the iteration continues[3]. Typically the

[1] In practice, the vector is filled with pointers to **SeedHit** objects [6], that are filled with the information that is needed for each hit from the "clusters" provided by the Tsa framework. This was a matter of convenience, allowing extra information to be added to the **SeedHit** as required; in the longer term, the **SeedHit** and **Tsa::Cluster** classes could be merged.

[2] At this point the tilt of the chambers is ignored, so the x used here is strictly speaking the coordinate defined by the tilted wires. This tilt has a size of 3.6 mrad, as the local beam axis of the LHC is tilted downward by that amount, while the T station modules (and thus the x wires) are vertical.

[3] The uncertainty on the hit position is taken as $\sigma = 200\,\mu\mathrm{m}/\cos\theta$ for the OT, with $\cos\theta = 1/\sqrt{1+s_x^2}$,

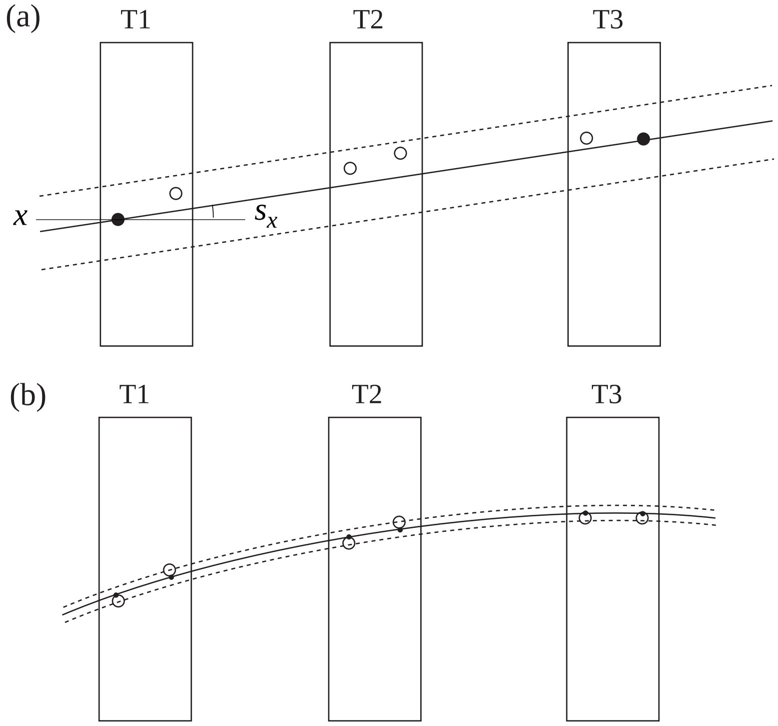

Figure 5: Search for track candidates in the X projection: (a) opening a window around a straight line between two hits in T1 and T3, where the cuts are indicated by dashed lines; (b) opening a window around a parabolic trajectory, after adding a hit in T2.

fit converges after a few iterations, because the L/R ambiguity choice and application of a window during the hit selection has already limited the outliers. The resulting X-search seed candidate is required to have at least seven hits, have an x-coordinate satisfying $|x_0| < 4.5\,\mathrm{m}$ and slope satisfying $|s_{x0}| < 0.8$ at $z = 0$, and a curvature term t_x which is consistent with the p_T kick angle discussed above: $|5.3 \times 10^4 t_x - \Delta\theta| < 0.8$.

The information contained in each accepted X-search seed candidate is stored in an object, which is then added to a vector of such candidates to be passed to the stereo search.[4] As a final step, the hits that have been used on a successfully selected candidate are tagged so

where θ is the angle of the track in the (x,z) projection.

[4] The vector contains pointers to **SeedTrack** objects, that contain information such as the number of x hits on the candidate, the parabolic parameters, etc, as well as a vector of objects that hold the information for each hit on the candidate. The latter objects are **SeedPnt**, which is basically a pointer to the relevant **SeedHit** along with a choice of L/R ambiguity. Whereas there is only one **SeedHit** created from each **Tsa::Cluster**, there can be many **SeedPnt**s pointing to a given **SeedHit**, if a given hit is shared between more than one seed candidate.

that they are not used in a similar combination for subsequent candidates. This removes most of the clones that would otherwise be found, since a good track typically has four x hits in each station and would be found many times over if such precautions were not taken.

Stereo search

The stereo search starts with the selection of stereo-layer hits that are compatible with the X candidate, within the sector of the Tracker that is under consideration. This involves converting each stereo hit to a y coordinate measurement, using the knowledge of the (x,z) trajectory from the X search, and then checking whether that y coordinate is within the physical boundaries of the hit channel (i.e. within the length of the wire, for an OT hit). During the conversion of stereo hit to y coordinate, the tilt of the T stations is taken into account.

The stereo search then proceeds by searching for a straight line trajectory amongst the selected hits, in a similar manner to the X search (but without the complication of the parabolic parameterization). Pairs of hits in the first and last station are taken to define a line candidate, with a cuts applied to its slope s_y such that $|s_y - y_1/z_1| < 0.1$ where (y_1,z_1) are the coordinates of the first hit. All hits within the window of $\pm 100\,\mathrm{mm}$ around that line are selected for further study, with a requirement that there should be at least 6 hits found (in addition to the two that formed the line candidate) out of ten that would expected for a track passing through the full set of OT stereo layers, if they all give a hit. Using just those selected hits, the four possible combinations of L/R ambiguity are then tested for the initial pair of hits, and the combination that gives the largest number of hits within a $\pm 10\,\mathrm{mm}$ window chosen, when the ambiguities of all the other selected hits have been tried. A straight line is then fitted to the set of hits, taking into account the preferred L/R ambiguity for each hit. The fit is iterated, in a similar manner to the parabolic fit of the X search, but with a looser cut on outliers $(4\,\sigma)$ to account for the additional uncertainty introduced by the conversion of stereo hit to y coordinates using the fitted (x,z) trajectory. The resulting candidate is required to have at least five stereo hits, and 15 hits in total including the x hits. Only the stereo candidate with the largest number of hits is kept. Candidates from the X search that that fail these requirements are not deleted, to avoid the need to reorder the vector that holds them, but instead a flag is set within the seed candidate object to indicate that it should no longer be used.

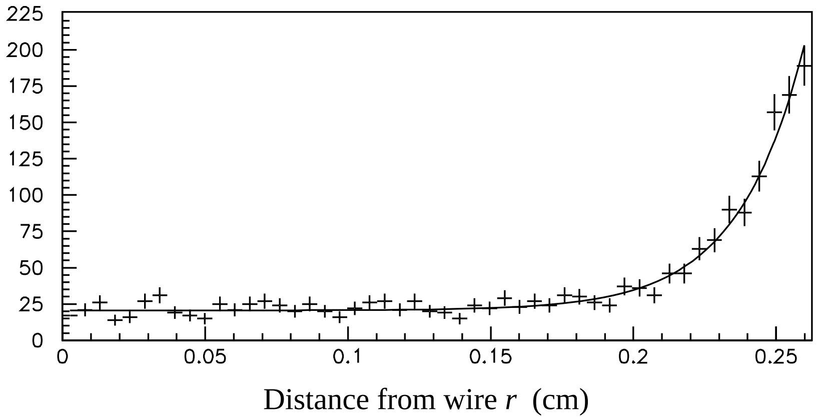

Figure 6: Histogram of the distance of the seed candidate trajectory from the wire, for OT cells that did not give a hit; the parametrization used in the likelihood calculation is superimposed.

Likelihood determination

A likelihood is calculated for each of the surviving candidates. Originally a simple sum of the χ^2 from the parabolic and linear fits to the candidate was used as a discriminant, but this penalizes tracks with more hits. One can try to combine the number of hits and the χ^2 in a composite discriminant. However, better performance was found by building a likelihood based on whether the track is *expected* to give a hit in each layer. The inefficiency of a single cell of the OT increases toward the wall of the straw, and a layer is of course insensitive to tracks that pass through the gaps between two cells (0.25 mm compared to the 5 mm cell diameter). A likelihood is calculated for each tracking layer where the candidate does not have a hit, although the fitted trajectory of the candidate passes through an active region of the tracker. It has a larger value if the trajectory passes through a less efficient region of the cell, as shown in Fig. 6. The product of these likelihoods is taken, and combined with a term that accounts for the goodness of fit of the hits that have been assigned to the candidate: this is taken as the probability of χ^2 for the parabolic and linear fits, calculated for the appropriate number of degrees of freedom. Finally a binomial counting term is added to the likelihood, according to the number of hits on the candidate compared to the number expected, taking into account the expected average efficiency. The natural logarithm of the combined likelihood value is then used as a discriminant to select good quality tracks.

The likelihood calculation is performed with the help of a tool provided by the Tsa framework [6], which for a given trajectory returns a list of the expected hits on that candidate, along with the distance of the trajectory from the wire for each of those hits.

Before this tool is used, a correction needs to be made to the parabolic fit parameters, as they were calculated in the tilted frame of the x wires. The parabola is refitted, after first adjusting the z-coordinate of the x hits, according to the known y coordinate taken from the linear fit of the candidate. After this refitting procedure the candidate's parameterization is in the standard LHCb coordinate system. The likelihood contribution for missing hits is determined by sampling a normalized probability density function that was filled from a reference sample of tracks. This is parametrized as a function of the radius r from the nearest wire, with value $\mathcal{L} = 0.261 + \exp(5.1\,r - 11.87)$. It was found that fluctuations of the probability of χ^2, when added to the likelihood calculated for missing hits, led to a degradation of performance of the track selection. As a result, a scaling factor was introduced, de-weighting the contribution from the probability of χ^2, for which a value of three was found empirically to give the best performance. The contribution to the log-likelihood of the counting term for the number of observed hits n compared to the number expected N is

$$\ln \mathcal{L} = \ln \frac{N!}{n!(N-n)!} + n \ln \epsilon + (N-n)\ \ln(1-\epsilon)\ ,$$

where ϵ is the average effective efficiency. This is found in the simulation to be 0.90 for the OT, 0.99 for the IT. Note that for the OT this is significantly lower than the single cell efficiency, and includes the effect of missing hits in the high occupancy environment due to more than one track passing through a cell.

The seed candidates are sorted according to their (log-)likelihood, and the one with the highest likelihood selected. Within each of the hits on that seed, a pointer is stored to indicate that the hit has been used for that selected seed. The next highest likelihood candidate is then checked to see whether it shares more than three hits with the first, in which case it is discarded, otherwise it is selected, and so on. At each step, the selected candidate should not share more than three hits with any of the previously selected ones. This continues until there are no remaining candidates, or the log-likelihood falls below a cut, which can be tuned to adjust the efficiency versus ghost rate as discussed below. This approach of using the likelihood as a discriminant to decide which tracks should be selected lends itself to wider application. If the tracks that are of interest all come from the interaction region, for example, a contribution can be added to the likelihood favouring those candidates with track parameters that point to that region. The efficiency of reconstructing such tracks will typically be higher with this approach than if a simple cut is applied on the pointing criterion, when evaluated for the same ghost rate.

2.3 Stub Search

After the selection of seed candidates has been carried out using the projection search for each tracking sector, a second pass is made using a stub search algorithm. This uses hits in the IT which have not been assigned to any selected candidate from the projection search. Track stubs are searched for in each of the IT boxes, for each of the stations. To form a stub, a hit is required from each of the four layers of the IT. Pairs of hits in the two x layers are used to define a straight line segment in the (x,z) projection, with cuts applied on the slope: $|s_x| < 0.5$, $|x_1/3125\,\mathrm{mm} - s_x| < 0.3$ (see Fig. 7). This parametrization is then used to convert the stereo hits in that box to y measurements, in a similar manner to that described above for the projection search, ensuring that the resulting coordinates are within the physical boundaries of the strips. Pairs of stereo hits in the u and v layers are required to give a straight line segment in the (y,z) projection that satisfies $|s_y| < 0.25$, and for which the slope points towards the interaction point, which is ensured by requiring $|y_2 - y_1 - (z_2 - z_1)\, y_1/z_1| < 3\,\mathrm{mm}$. No clone suppression is needed in this case, as the occupancy is rather low in the IT compared to the OT, and anyway many hits have been removed by the first pass. However, this stub search finds segments of those tracks which do not stay within a given IT sector, either because they cross to another IT sector or to the OT at one of the other T stations.[5]

The next step is to link stubs together to make seed track candidates. This starts by taking a stub from the first and second stations, from any IT sector, and calculating the distance between them in both projections at the midpoint between the two stations. If $|\Delta x| < 3\,\mathrm{mm}$, $|\Delta y| < 3\,\mathrm{mm}$ and $|\Delta s_x| < 30\,\mathrm{mrad}$ then the two stubs are marked as being linked together, and a similar test is made to search for a linked stub in the third station. Stubs from the second and third stations are also tested to see if they can be linked. Once pairs or triplets of stubs have been linked together, fits are made to the hits that have been used to make those stubs, of a parabola in the (x,z) projection and a straight line in the (y,z) projection, in order to form a seed candidate. As a final step, for stubs that cannot be linked to any others, a check is made to see whether they come from a track that crossed from IT to OT, by extending the stub into the OT and searching for associated hits. This is done by searching for hits consistent with a parabolic trajectory, amongst the x hits in the OT that are within a window from the extrapolated stub. Then a search is made for a linear trajectory using the stereo hits of the same OT sector, this time consistent with a line joining the interaction point to the stub y coordinate (the slope information from the stub alone is not sufficiently precise in the (y,z) projection).

[5] The stub candidates are saved in **SeedStub** objects.

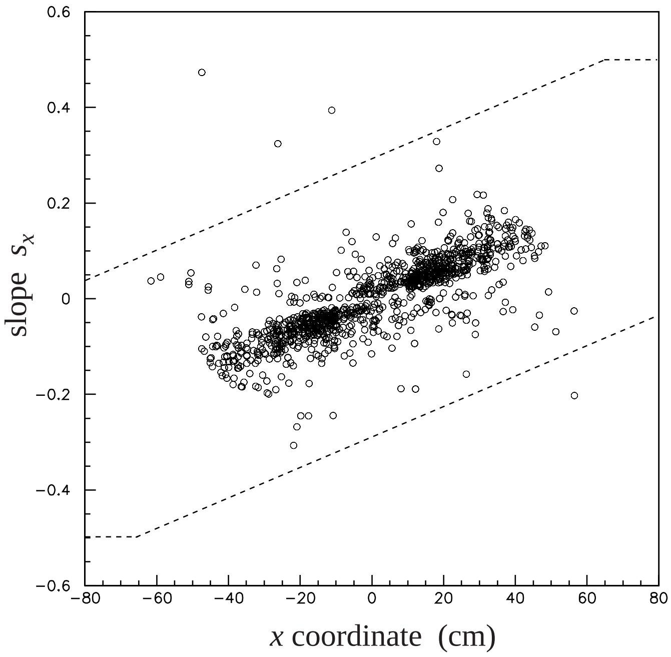

Figure 7: Correlation between slope and x coordinate for long tracks passing through the IT; the cuts applied to select stub candidates are indicated by the dashed lines.

2.4 Output

The final stage of the algorithm is to convert the working **SeedTrack** objects to the standard **Track** class [4]. This is done by the **TsaSeedTrackCnv** algorithm. The main part of this algorithm is related to format translation. In addition, this algorithm provides an improved estimate of the particle momentum using a p_{T}-kick method [7]. In this method the effect of the magnetic field is approximated by a single kick at the centre of the magnet (Fig. 8). The particles' momentum can then be estimated from:

$$\Delta p_x = p_{x,f} - p_{x,i} = p\left(\frac{t_{x,f}}{\sqrt{1+t_{x,f}^2+t_{y,f}^2}} - \frac{t_{x,i}}{\sqrt{1+t_{x,i}^2+t_{y,i}^2}}\right) = q\int|\vec{dl}\times\vec{B}|_x$$

where $t_{x,f}$ and $t_{y,f}$ are the slopes of the T seeds. They are known from the parameters of the **SeedTrack** and are evaluated at T1. The total integrated field and the centre of the magnet (that is to say the z at which the integrated field equals half its total value) are estimated using the **TrackPtKick** tool. Using this class for tracks in the T-acceptance a core momentum resolution of 1.6% is achieved — with some tail, the RMS of the distribution is around 5 %. This is a sizeable improvement on estimate coming from

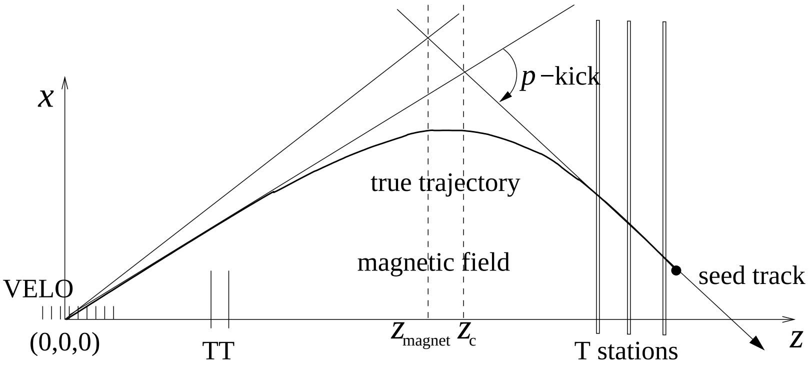

Figure 8: Schematic drawing of the p_T-kick method (Not to scale).

the track curvature in the T stations of 24 %. A possible objection to this procedure is that it is biased for tracks not originating from the primary vertex. For tracks produced before $z = 3$ m this procedure gives a core momentum resolution of 3 % with an RMS of 8 %. This is still considerably more accurate than the estimate coming from the track curvature. Therefore, it is concluded that this procedure is also optimal for these tracks.

The container of **Tracks** is registered on the transient event store and fitted using a Kalman filter [8]. The Kalman filter is initialized with the track parameters provided by the T-seed. In addition, an initial estimate of the covariance matrix is needed. In general the initial estimates of the uncertainties on the track parameters are set to large values to give a non-informative prior [9]. If this procedure is followed for the momentum parameter then during the fit the information on the momentum given by the p_T-kick method is lost. The fit will converge to the value that gives the best estimate of the curvature within the T-stations. To retain the information on the momentum provided by the p_T-kick the error on $\delta p/p$ is set 4% at the start of the fit. This value is chosen because it was found to give optimal performance for the track matching algorithm. From the above discussion it seems a reasonable compromise between obtaining good momentum resolution for particles that originated from the primary vertex region on one hand and being unbiased for particles that originated after this point. After the fit a momentum resolution of 3.7 % is found.

3 Definitions

Before discussing the performance it is necessary to define performance indicators. The seeding algorithm finds tracks from three sources. The first class is particles from the primary vertex that have hits in the VELO. A particle is defined to be in the 'Long Track' acceptance if it satisfies the following criteria:

- The particle momentum at its production vertex is more than 1 GeV.
- Three reconstructed clusters in the r sensors of the VELO.
- Three reconstructed clusters in the ϕ detectors of the VELO.
- A reconstructed x and stereo hit in each of the tracking stations T1-T3.
- It does not interact hadronically before the end of the T stations.

This is the most important class of tracks for physics and the majority of the results given in this note will be for this acceptance definition. The second class is tracks from particles produced in secondary interactions or $\mathrm{K_S}$ decays after the VELO. A particle is said to be in the 'Downstream Acceptance' if:

- The particle momentum at its production vertex is more than 1 GeV.
- The particle is not reconstructible within the VELO.
- A reconstructed x and stereo hit in each of the tracking stations T1-T3.
- A reconstructed hit in each of TTa and TTb.
- It does not interact hadronically before the end of the T stations.

The final class is all particles that can reasonably be expected to be reconstructed by the algorithm. A particle is said to be in the 'T acceptance' if:

- The particle momentum at its production vertex is more than 100 MeV.
- A reconstructed x and stereo hit in each of the tracking stations T1-T3.
- It does not interact hadronically before the end of the T stations.

In each case the track reconstruction efficiency is then given by:

$$\text{efficiency} = \text{N(accepted } \cap \text{ track reconstructed)/N(accepted)}$$

For the efficiency calculation all particles except electrons satisfying the above acceptance criteria are used regardless of their origin. Electrons are excluded because the majority originate in secondary interactions such as photon conversions. These have little physics interest but are more difficult to reconstruct due to subsequent bremsstrahlung in the detector material.

To determine whether a Monte Carlo has been reconstructed an association algorithm is needed. A track is said to be related to a true particle (**MCParticle**) if more than 70 % of the clusters in the VELO come from that particle and more than 70 % of the hits in the seeding region also come from that particle.

The other important indicator of the tracking performance is the ghost rate. The is defined as:

ghost rate = N(rec tracks not related to a **MCParticle**)/N(rec tracks)

Both the efficiency and the ghost rate can be calculated in two ways. The first is to calculate these quantities on an event-by-event basis ('event weighted'). If values for the whole event sample are required the averages of the resulting distributions are used. The alternative is simply to calculate the efficiency and ghost rate on the whole sample of tracks ignoring which event the track came from ('track weighted'). Since there are large event-to-event fluctuations in the case of the ghost rate the first method is preferred and will be used for the majority of the results presented in this note.

4 Performance

The performance of the algorithm has been studied using data generated for the DC06 production [10]. Four data samples were used:

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

The majority of results were obtained with the first sample. From the context it should be clear when other samples were used.

With the default settings of the algorithm an event weighted efficiency of 92.3 % is found for particles in the Long acceptance [6]. Fig 9 shows the efficiency as a function of momentum. It can be seen that above momenta of 2 GeV the efficiency plateaus at around 95 %.

[6] The track weighted efficiency would be 91.6 %.

Below 2 GeV the efficiency of the algorithm falls rapidly. Another source of inefficiency is the 7 cm region around $y = 0$ cm in the Outer Tracker where the detector is only 50 % efficient. If this region is excluded the efficiency at high momentum increases to 96 %. The efficiency for reconstructing all particles in the T acceptance as defined in Section 3 is 78%.

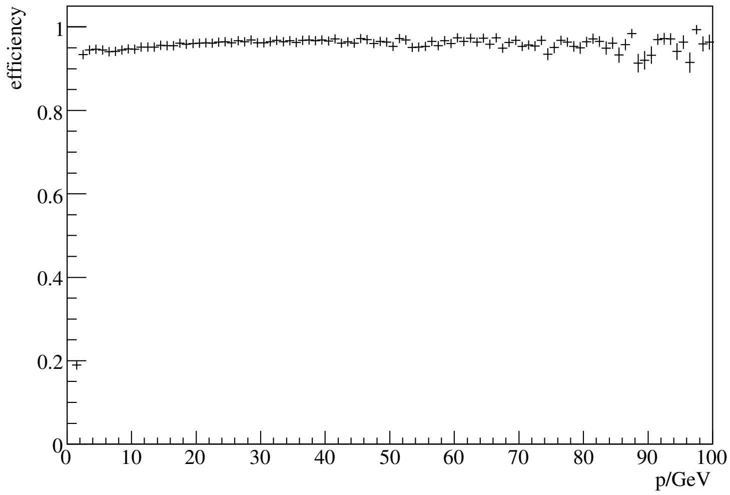

Figure 9: Track weighted efficiency as a function of particle momentum.

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

Track type	$\overline{p}$/GeV	Track efficiency
$\mu^\pm$ from $B_d \to J/\psi(\mu^+\mu^-)K_S(\pi^+\pi^-)$	33	96 ± 0.4
$e^\pm$ from $B_d \to J/\psi(e^+e^-)K_S(\pi^+\pi^-)$	34	94 ± 0.3
$\pi^\pm$ from $B_d \to J/\psi(\mu^+\mu^-)K_S(\pi^+\pi^-)$ in Long acceptance	12	91 ± 2
$\pi^\pm$ from $B_d \to J/\psi(\mu^+\mu^-)K_S(\pi^+\pi^-)$ in Downstream acceptance	14	94 ± 1

Table 1: Efficiencies for reconstructing tracks from specific B final states.

and electrons from $B_d \to J/\psi(e^+e^-)K_S(\pi^+\pi^-)$ a performance comparable to that obtained with the inclusive track sample is found. For the electron case the performance is slightly worse reflecting the fact that bremsstrahlung in the material of the detector makes them harder to reconstruct. The performance for pions from $B_d \to J/\psi(\mu^+\mu^-)K_S(\pi^+\pi^-)$ where the pion gives sufficient hits in the VELO to be reconstructible as a Long Tracks is slightly worse. This may be partially explained by the lower momenta of these tracks. The performance for pions from $B_d \to J/\psi(\mu^+\mu^-)K_S(\pi^+\pi^-)$ where the pions are reconstructible as Downstream tracks is comparable to that obtained with the inclusive track sample.

The event weighted ghost rate is 13 % [7]. If a higher purity is needed the ghost rate can be reduced, at some cost in efficiency, by cutting harder on the likelihood variable. Fig. 10 shows the efficiency versus ghost rate for various cuts on this variable. It can be seen from this plot that ghost rate can be reduced by a factor of almost two for a loss in efficiency of $\sim$ 1 %. It should be noted that since in the standard LHCb software no link to the Monte Carlo truth is stored for hits coming from spills other than the event one, if a track from a previous spill is reconstructed it will be classified as ghost. In dedicated studies it has been found that such tracks give (relatively) 10 % of the observed ghost rate.

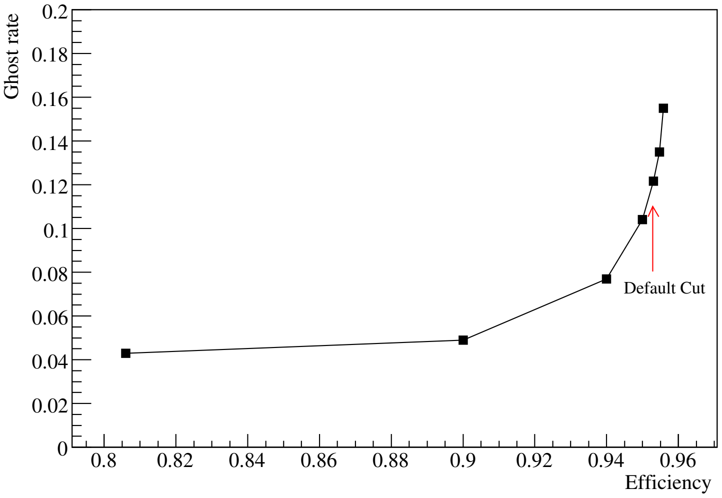

Figure 10: Efficiency for tracks with p > 2 GeV versus ghost rate for various cuts on the Likelihood. From left to right the cut values are −15 , −20, −25, −28, −30, −32 and −35.

The performance as a function of the number of visible interactions as defined in [3] has been investigated. Fig. 11 shows the dependence of the efficiency and ghost rate on

[7] The corresponding track weighted ghost is 21.3 %

this quantity. It can be seen that the dependence of the efficiency on the number of visible interactions is quite weak. For each additional visible interaction in the detector the efficiency decreases by ~ 1.2 %. The ghost rate shows a strong dependence on the number of visible interactions in the detector. For each additional interaction the ghost rate increases by ~ 10 %.

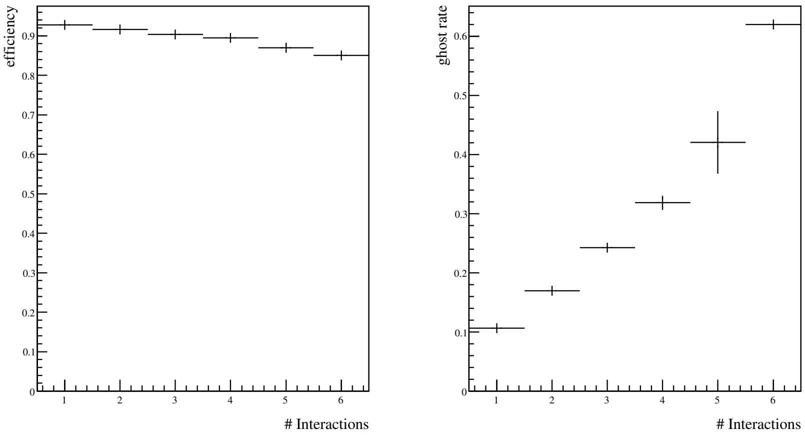

Figure 11: Efficiency (left) and ghost rate (right) versus the number of visible interactions.

In addition, the performance with data generated at a higher luminosity of 5×10^{32} cm^{-2}s^{-1} has been studied. In this case an efficiency of 91% and a ghost rate of 22 % is found. If only the number of visible interactions in the event spill effects the performance of the track reconstruction then efficiencies and ghost rates for an arbitrary luminosity can be derived directly from Fig. 11. At a luminosity of 5×10^{32} cm^{-2}s^{-1}, on average there are two visible interactions per B event. From Fig. 11 the efficiency in this case would be expected to be 91.6 % and the ghost rate 22 % — in agreement with the observed values. At higher luminosities this extrapolation will break down due to increased spillover that further increases occupancies and detector dead-time.

Finally, the CPU performance of the algorithm has been evaluated. On a 2 GHz Intel Centrino machine the algorithm runs in a time of 220 ms per event using the standard LHCb compilation options. Fig. 12 shows the time per event versus the total number of hits in the T-stations. The observed behaviour can be parameterized as:

$$t = 4.7 \times 10^{-12} \times N^{2.93} \text{ [s]}.$$

The power law dependence with an exponent between two and three is expected given the track finding approach used [11]. A 15 % improvement in speed is observed if the

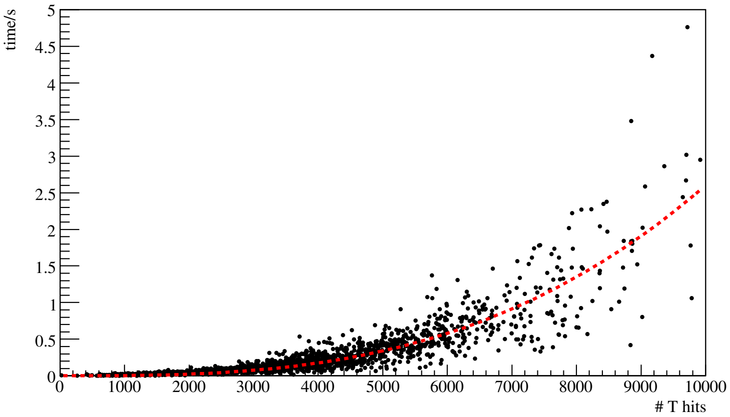

Figure 12: Algorithm time versus the total T station multiplicity. The dashed line is the parameterization given in the text.

compiler flags that allow the use of the SSE registers are enabled [8].

5 Strategy for tuning 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. Given the relative complexity of the seeding algorithm even when real data is available simulated data will certainly be used to tune the algorithm. Therefore, it is important that the Monte Carlo is tuned to reflect the efficiencies, resolutions and noise rates observed in the data. However, this should be complemented with indicators of the performance that can be derived from directly from the data.

To estimate the efficiency it is proposed to use the method similar to that used by the Hera-B collaboration [12]. By linking Velo tracks to calorimeter clusters or L0 muon candidates the denominator for an efficiency calculation can be estimated. An efficiency can then be easily estimated by seeing how many of these tracks are found by the seeding. A direct determination of the ghost rate from the data seems harder. Instead, the ghost rate will be estimated by comparing distributions such as the likelihood or the number of hits on a track between data and Monte Carlo. If it is assumed that the Monte Carlo at least models the shape of these distributions correctly for real and ghost tracks then the ghost rate can be estimated.

[8] The gcc compiler flags: -msse2 -mfpmath=sse.

It is clear from the previous discussions that work is needed to demonstrate that the efficiency can be determined in the manner suggested and to ensure that all necessary monitoring tools are in place before the start of data taking.

6 Summary

In this note an algorithm for finding tracks in the T stations has been described in detail. An efficiency of 94–95 % has been achieved for tracks with momenta above 2 GeV for a ghost rate of 13 %. The algorithm runs in around 220 ms event. The latter is a sizeable improvement on previous implementations of the algorithm and has been achieved despite the large increase in T-station occupancy seen in DC' 06 compared to DC '04 [13, 14]. Thus, the feasibility of fast track reconstruction in the T-stations has been demonstrated.

The studies that have been presented here should not be considered as representing the definitive performance of the algorithm, rather they represent a snapshot of the performance at the start of the DC 06 reconstruction phase. Since that time further studies have led to improvements in the algorithm mainly related to the ghost rate and speed. These studies will be documented in a subsequent note [15].

References

[1] M. Needham. Tsa: Fast and Efficient reconstruction for the Inner Tracker. LHCb-Note 2004-075.

[2] R. Forty. Track Seeding. LHCb-Note 2001-109.

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

[4] E. Rodrigues J. Hernando Morata. Tracking event model. LHCb-note 2007-007.

[5] M. Needham. A Simple clustering algorithm for the Outer Tracker. LHCb-Note 2001-149.

[6] M. Needham. The Tsa Reconstruction Framework. LHCb-Note 2007-037.

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

[8] E. Rodrigues . The LHCb Kalman Fit. LHCb-note 2007-014.

[9] M. Merk *et al.* Performance of the LHCb OO Track Fitting Software. LHCb-note 2000-086.

[10] Gauss v25r7, Boole v12r10, Brunel v30r14, XmlDDDB v30r14.

[11] R. Frühwirth *et al. Data Analysis +Techniques for High-Energy Physics*. Cambridge University Press, 2000.

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

[13] M. Needham. ST Occupancies and Clustering. LHCb-note 2007-024.

[14] J. Amoraal. OT Simulation. LHCb-note 2007-018.

[15] R. Forty and M. Needham. Updated Performance of the T seeding. LHCb-note 2007-023.