LHCb-PUB-2014-002
March 27, 2014

The Seeding tracking algorithm for a scintillating fibre detector at LHCb

Y. Amhis[1], O. Callot[1], M. De Cian[2], T. Nikodem[2], F. Polci[3]

[1]*LAL, Université Paris-Sud CNRS/IN2P3, Orsay, France*
[2]*Physikalishches Institue, Ruprecht-Karls-Universitat Heidelberg, Heidelberg, Germany*
[3]*LPNHE, Université Pierre et Marie Curie, Université Paris Diderot, CNRS/IN2P3, Paris, France*

Abstract

The project of the LHCb upgraded detector foresees the presence of a Scintillating Fiber Tracker. This document describes the algorithm used for reconstructing standalone tracks in the SciFi, called *Seeding*. This algorithm is crucial for reconstructing tracks generated by long lived particles such as K^0_S. The main performances on simulated samples for running conditions expected in future data taking after the upgrade, namely a luminosity larger then $\mathcal{L} = 2 \times 10^{33}$ cm^{-2}s^{-1}, are also discussed.

1 Introduction

The LHCb detector shown in Fig. 1 is a single-arm forward spectrometer covering a pseudo-rapidity range $2 < \eta < 5$, designed to study particles containing a b or a c quark. This detector will be upgraded and it is foreseen to be operational in 2018. While the main structure will be maintained various sub detectors will be replaced. Around the interaction point there is the Vertex Locator (VELO), followed by a silicon strip detector upstream of the magnet (UT) and a *Scintillating Fibre* detector downstream of the magnet (FT). To increase the mean free path of the particles and reduce multiple scattering it was decided to not equip the detector with tracking stations inside the magnet.

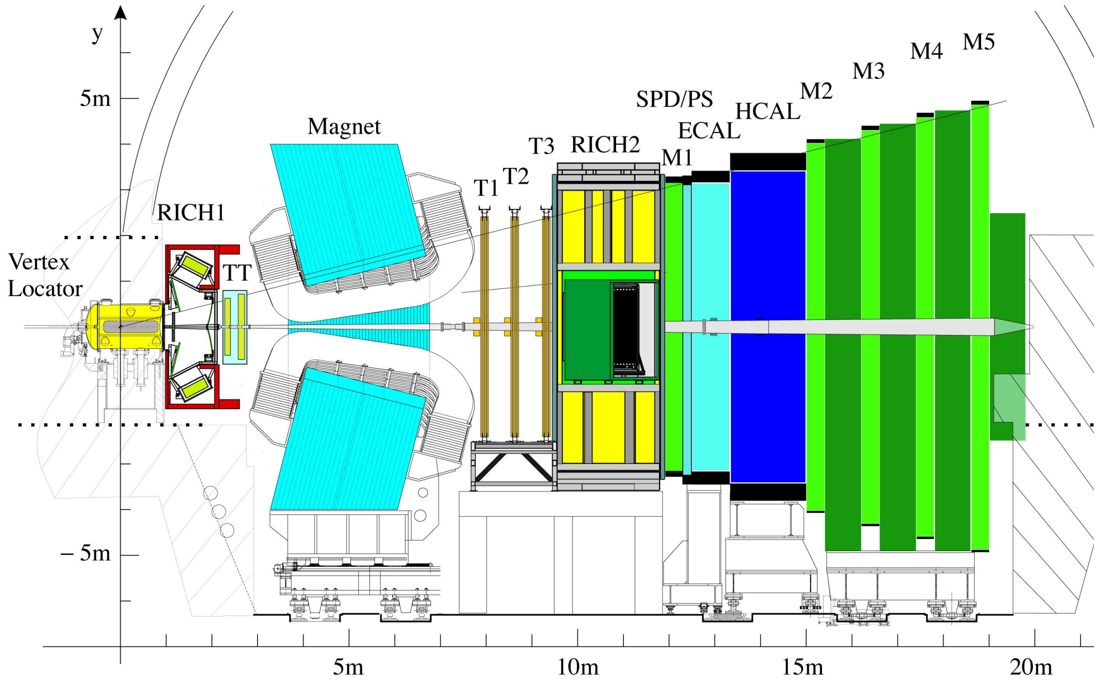

Figure 1: Schematic view of the current LHCb detector. In the upgrade the TT will be replaced by the UT, and the T stations (T1-T3) will be equipped with a *Scintillating Fibre* detector (FT).

The *Scintillating Fibre* detector downstream of the magnet is made of three tracking stations, T1, T2, T3 as shown in Fig. 2 where the active material is a mat of scintillating fibers. The detector geometry is described in details on Ref. [1]. Each station is composed of four layers (X-U-V-X). In the first and fourth layer of each station, fibers are vertically oriented to give the x coordinate of the track, *x-layers*, while in the second and third layers the fibers form a $\pm 5°$ angle with the y axis of the laboratory frame, thus providing the u and v *stereo* coordinates. The effects of radiation damage corresponding to 50 fb^{-1} of data-taking are simulated in the events used to evaluate the performances of the *Scintillating Fibre*. For these conditions, the expected efficiency to detect a single hit is $\sim$ 99% [2]. Additionally about 0.8% of the area will be dead regions, decreasing this efficiency even further. Concerning pattern recognition a 98% probability to detect a single hit means, that only 78.5% of all tracks will leave all possible 12 hits inside the detector. However, if one allows for two missing hits, it would be nevertheless possible to reconstruct 99.8% of all tracks. The single hit resolution is $\sim 70\,\mu m$ depending on the number of fibres combined to a cluster. This is sufficient, as the limiting factor for the momentum resolution is multiple scattering. Nevertheless, the good spatial resolution is an advantage in patter recognition for efficient track finding and reducing the number of wrong combinatorics.

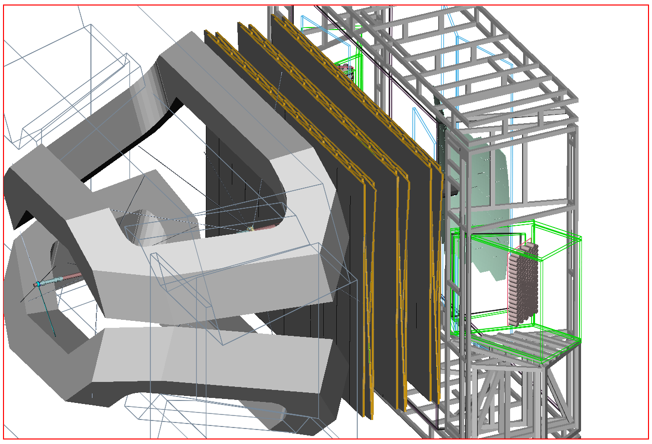

Figure 2: Schematic view of the *Scintillating Fibre Tracker*.

In the first stage of the reconstruction of the tracks so-called the pattern recognition algorithms are run to find a collection of hits associated to the flight path of charged particle through the detector. The two most important pattern recognition algorithms using information from the *Scintillating Fibre* in the LHCb track reconstruction are the *Forward Tracking* [3] and the *Seeding*. As will be discussed later, the parameters of these algorithms are optimised to maximise the track reconstruction efficiency while keeping low the processing time and the rate of fake tracks. Following the pattern recognition, other algorithms are ran, such as the track fitting using Kalman Filters[1] or the clone killing. This note focuses on the description of the *Seeding* pattern recognition algorithm. While the *Forward Tracking* uses as an input tracks found in the VELO, the *Seeding* relies exclusively on information form the *Scintillating Fibre* detector. This algorithm is specifically suited for reconstructing tracks (T-Track as shown in Figure 3) generated by the decay products of particles, like K^0_S, decaying after the VELO.

2 The *Seeding* algorithm

The prototype algorithm *PrSeedingXLayers* relies on a similar logic to what was developed for current LHCb detector, namely *PatSeeding*. A complete description of *PatSeeding* can

[1]The pattern recognition stage itself comprises only a simplified fit of the hits which gives a (pseudo-)χ^2 value as a quality criterion and a momentum estimate.

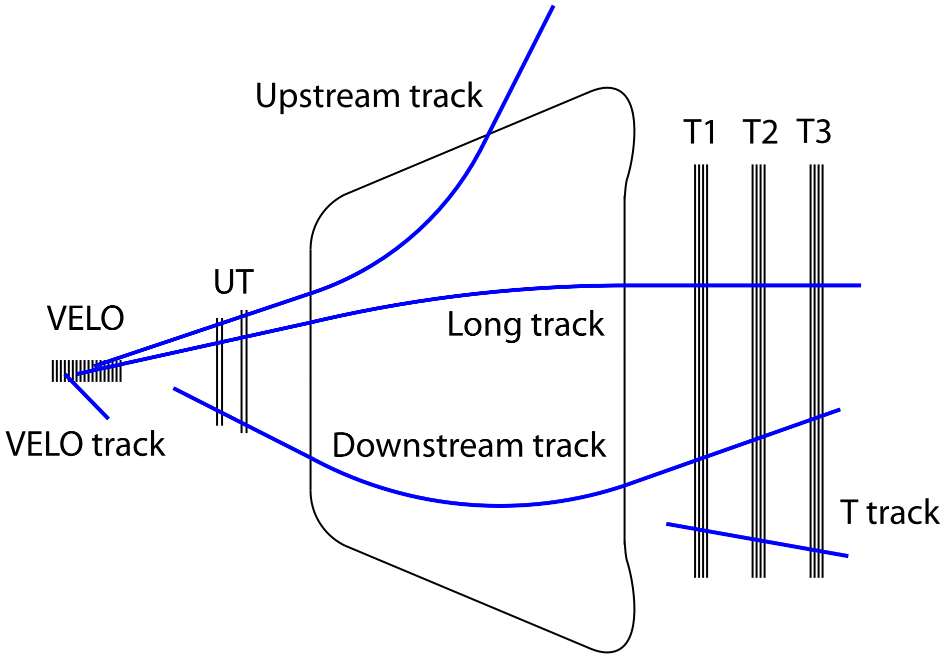

Figure 3: Graphic representation of the different types of tracks in LHCb . The T tracks, having hits only in the *Scintillating Fibre Tracker*, are reconstructed by the *Seeding* algorithm.

be found in Ref. [5] and [6]. The individual steps are briefly summarized in the following.

2.1 Track model and track fit

In the absence of magnetic field, multiple scattering and energy loss the tracks would follow straight lines. While decreasing along the $z-$axis, the LHCb magnet field has impact primarily in the $x-z$ trajectory, while its influence is almost negligible in the y direction. For these reasons, the *Seeding* relies on a straight line description of the trajectory in $y-z$ direction and a parabolic model is used for the $x-z$ projection. For the purpose of identifying the tracks, each layer of the detector is divided in an upper $(y>0)$ and a lower $(y<0)$ zone, numbered from 0 to 23 as shown on Figure 4. The even numbers identify the lower zones, odd numbers identify the upper zones. Since most of the tracks do not cross the $y=0$ plane, only hits from the upper (lower) zones are combined to produce track candidates. This already reduce by half the number of hits to be processed when searching for a track and therefore reduces the processing time.

The track search is done in two steps, first the hits are searched for in the *x-layers* then in the *u/v-layer*. The details of the implementation of each step is described in the following. The list of parameters used to tune the algorithm can be found in Table 1.

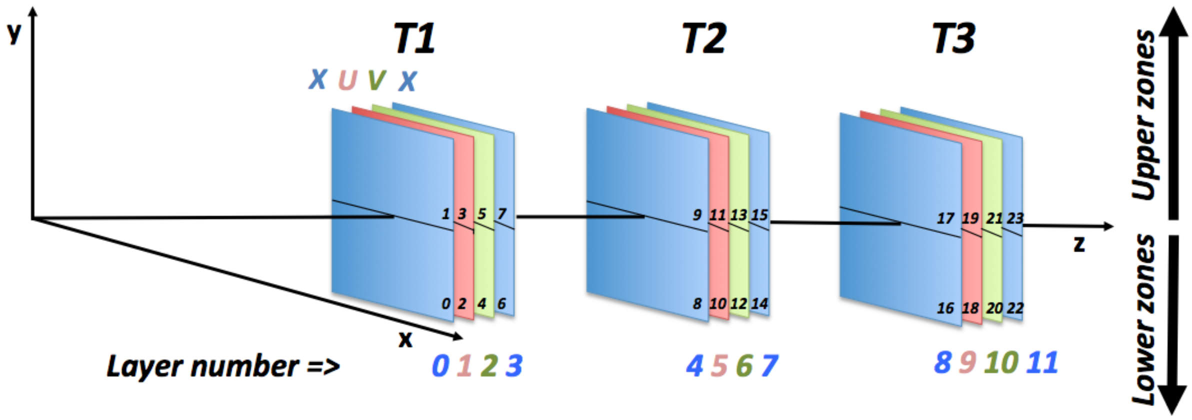

Figure 4: The detector layout and the numbering scheme used by the algorithm. Layers are numbered from 0 to 11. Each layer is divided in an upper and lower part, numbered from 0 to 23: even numbers identify lower zones, odd numbers identify upper zones.

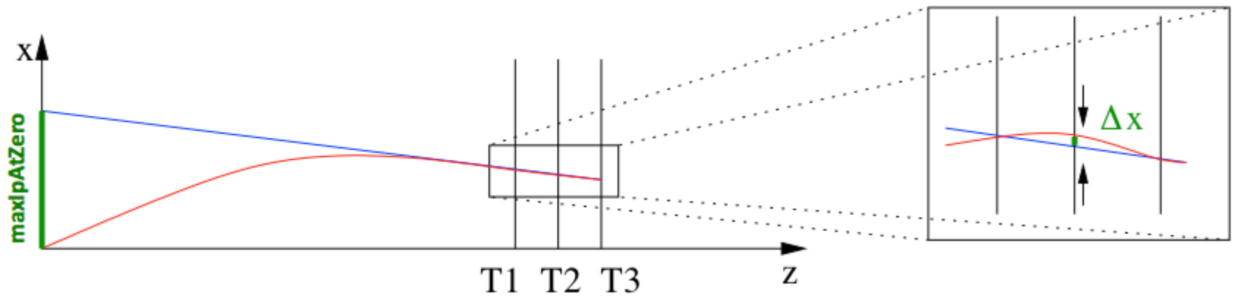

Figure 5: A simplified graphic representation of the search for $x - z$ projections of tracks in the *Seeding*.

Option name	Default Value
`maxIpAtZero`	5000 mm
`TolXInf`	0.5 mm
`TolXSup`	8.0 mm
`minXPlanes`	5
`maxChi2PerDoF`	4.0
`maxChi2InTrack`	5.5
`TolTySlope`	0.015 mm
`TolTyOffset`	0.002

Table 1: Default values of the parameters used in the *Seeding* algorithm.

2.2 Collecting hits in *x-layers*

The first step of the *Seeding* is a search for track candidates using only hits in the *x-layers* Since no inputs tracks from the VELO are used, an assumption on the origin of the track is needed. For this purpose a maximum impact parameter of the track at $z = 0$ (`maxIPAtZero`) is defined as illustrated in Figure 5. Then, for any hit in the first *x-layer* of coordinates (x_f, z_f), the maximum impact parameter is used to identify the maximum and minimum straight lines connecting the hit to the $z = 0$ plane. These straight lines, enveloping by definition any possible track candidate, are expressed by the equations:

$$x = \frac{z}{z_f} x_f \pm \mathtt{maxIPAtZero}(1 - \frac{z}{z_f} x_f) \tag{1}$$

From these equations the range $[minX_l; maxX_l]$, inside which searching for hits in the last *x-layer*, gets defined:

$$minX_l = \frac{z_l}{z_f} x_f - \mathtt{maxIPAtZero}(1 - \frac{z_l}{z_f} x_f) \tag{2}$$

$$maxX_l = \frac{z_l}{z_f} x_f + \mathtt{maxIPAtZero}(1 - \frac{z_l}{z_f} x_f) \tag{3}$$

For any hit of the last layer inside this range, the slope t_x and the constant term x_0 of the straight line connecting it to the hit in the first layer are evaluated:

$$t_x = \frac{x_l - x_f}{z_l - z_f} \tag{4}$$

$$x_0 = x_f - z_f t_x \tag{5}$$

and used to calculate the expected x-coordinate (x_P) of the track in each *x-layer* of z-coordinate z_{zone}:

$$x_P = z_{zone} t_x + x_0 \tag{6}$$

At this point, all hits inside a tolerance interval defined as $[x_P - \mathtt{TolXInf}, x_P + \mathtt{TolXSup}]$ for each x-layer in between the first and the last layers are collected. They are considered as potential hits of the track. At this point, a third hit in one of the *x-layers* of the second station is searched for within a interval $[x_P - \texttt{TolXInf}, x_P + 2 \cdot |t_x| \cdot \texttt{TolXSup} + 1.5]$. The dependence on t_x was introduced as a corresponding behaviour was observed on simulated data. The x and z positions of these three hits define a parabola, whose defining equation is calculated. This procedure is repeated for each hit which is found within the tolerance in the *x-layers* of the second station. For each parabola, hits are now searched within a search window of $[x_{para} + 2 \cdot |t_x| + 0.5, x_{para} - 2 \cdot |t_x| - 0.5]$ in the remaining *x-layers* in all stations, excluding the ones already used for the hits on the parabola. x_{para} stands for the predicted x-position of the hit following the parabola. Only one hit is added to the candidate in each layer. If more than one hit is found inside the search window, only the one with the least distance to x_{para} is taken.

The procedure is repeated three times: one assuming layer 0 as the first *x-layers* with an hit and layer 11 as the last *x-layer* with an hit; then assuming layer 3 as first *x-layer* and still layer 11 as last *x-layer*; finally assuming layer 0 as first *x-layer* and layer 8 as last *x-layer* (see Figure 4 for the layers numbering scheme). This allows to take into account cases where there are no hits in the layers 0 or 11.[2] As this procedure creates many clones, identical candidates are removed.

At this point the sets of hits in the *x-layer* that could have been left by a particle passing through the detector have been obtained. A fit is performed to each set of hits, using a polynomial of second order. Indeed, the track is expected to bend due to the residual magnetic field acting in the space where the tracking stations are located. If the fit is not successful, the worst hit is removed and a new fit is repeated until it becomes successful. If no successful fit is obtained, the set of hits is discarded. Otherwise, a track candidate has been obtained and, if it has a minimal number of hits (`minXPlanes`) and the χ^2 of the fit is less than a χ^2_{max}, it is kept. If some tracks share more than two hits, they are considered as clones and a procedure to choose the best among them is applied: the one having more hits is chosen, but if the clones have even the same number of hits, the one with higher χ^2 is discarded. The search of hits in the *x-layers* is the most time consuming step and takes about 88% of the *Seeding* time processing.

2.3 Adding stereo hits

The track candidates obtained from the procedure described in the previous section are just projections of the tracks in the $x - z$ plane, since only hits in the *x-layer* have been used. To get the whole track path in the space, the hits from the *stereo* layers must be added.

For this purpose, in each *u/v-layer* the algorithm calculates first of all the predicted x coordinate of the track candidate at the z of the layer, called x_P. A tolerance value defined by the product of the *stereo* angle α and the y-length of the detector (2500 mm) plus a safety margin (200 mm) is computed and used to define the interval $[x_P - 2700\alpha; x_P + 2700\alpha]$: only the hits in the *u/v-layer* whose x_{hit} belongs to this interval are retained.

[2]The third hit in the parabola does then not necessarily need to be in the second station.

Then each of the *stereo* hits in this interval is kept if the angle $\beta = \frac{x_{hit} - x_P}{z\alpha}$ is greater than 0.005 rad (lower than -0.005 rad) for the y-positive (y-negative) part of the layer (see figure: 6). This requirements translates into a y tolerance respect to the predicted hit, depending through β on the z of the layer.

Finally an additional requirement is applied: the difference between any β of a *stereo* hit surviving the previous selections and the β of the first hit β_{first} must be less than a tolerance value $T_y = \texttt{TolTyOffset} + \texttt{TolTySlope} \times |\beta_{first}|$.

Once the hits have been selected, the fit is performed again, this time using also the *stereo* hits, and the same criteria as before (worst *stereo* hit removal for failed fits, clones removal), are applied to select the best track candidate.

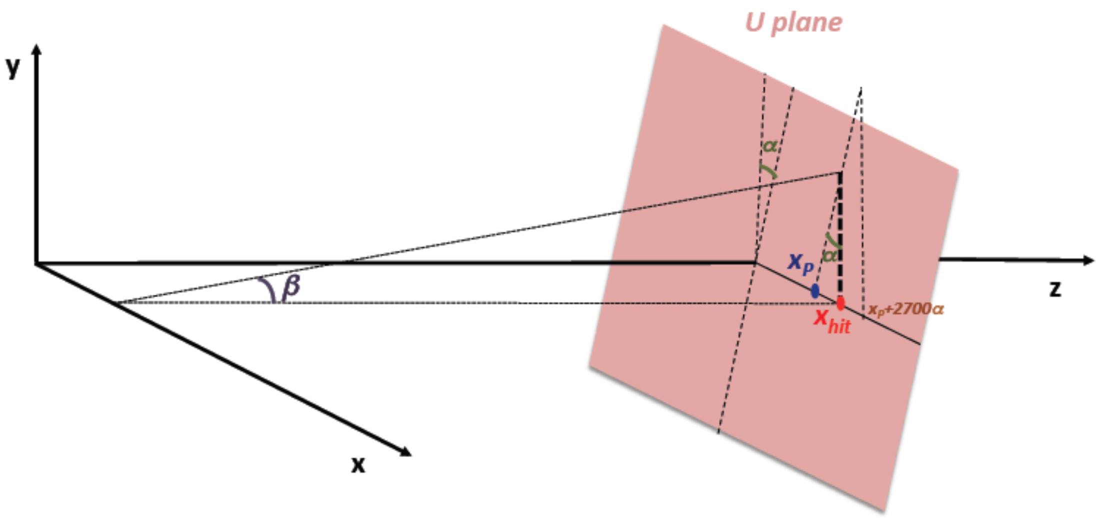

Figure 6: A sketch of the constraint on the selection of the *stereo* hits.

2.4 Converting the hit collections to LHCb tracks

In the LHCb track model, a track produced by a charged particle going through the LHCb detector can be described by five parameters. They are the position (x, y), the track slopes $(t_x = dx/dz$, $t_y = dy/dz)$ and the momentum at a given z- position. After a collection of hits is identified, the track candidates are passed to `PrGeometryTool`. The main purpose of this algorithm is to evaluate the momentum of the tracks. The computation of the momentum is done using an empirical method which models the behavior of the tracks in the LHCb magnetic field. It assumes that the tracks come from the interaction point $(x = 0, y = 0, z = 0)$. Finally, the track candidates are converted into `LHCbTrack`, which are the standard objects used in the LHCb reconstruction.

3 The *Seeding* algorithm performances

The *Seeding* algorithm has been tested on Monte Carlo samples in order to evaluate its performances. The LHCb upgraded detector is expected to work at higher instantaneous luminosity than today and so in worse pile-up conditions. Many samples have been used, simulating two different level of pile-up: one with $\nu = 3.8$ and one with $\nu = 7.6$, where ν is the mean number of primary vertices per bunch crossing. These samples correspond to instantaneous luminosities of 1 and 2 $\times 10^{33}$ cm^{-2}s^{-1} respectively. The algorithm has been executed using the same setup for all the analysed Monte Carlo samples, *i.e.* applying the same size of the search windows, the same quality cuts. The simulated decays were $B_s \to \phi\phi$ and $D^*(D^0(\to K^0_S\pi\pi)\pi)$. Samples of $B_s \to \phi\phi$ events have also been generated using the current detector with a level of pile-up corresponding to $\nu = 2$. We focus here on the most relevant tracks for physics analyses: `long track` and tracks produced in long-lived particle decays containing a strange quark for example K^0_S and Λ and leaving hits in both the *Upstream* tracker and the *Scintillating Fibre Tracker* (*strange* + UT + T). Additionally we quote the track efficiencies for tracks produced by *bottom* or *charm* hadrons.

Throughout the document the following definitions are used:

- *Reconstructible:* is a track having at least one hit in a x-layer and one hit in a *stereo*-layer in each of the three stations;
- *Reconstructed:* is a track matched successfully to a Monte Carlo particle;
- *Efficiency:* is the number of *reconstructed* tracks among those *reconstructible* divided by the number of *reconstructible* tracks;
- *Ghost track:* is a track which cannot be matched to any Monte Carlo particle;
- *Ghost rate:* the percentage of reconstructed tracks which are ghost tracks.

An important constraint for a pattern recognition algorithm is the processing time[3]. In the LHCb reconstruction, the *Seeding* can be executed stand-alone using all the hits produced in the detector. The current timing measurements are shown in T able 2.

The *ghost rate* of the *Seeding* algorithm for all the generated Monte Carlo samples are shown on Table 3. While the *ghost rate* is fairly similar for samples containing a *bottom* or a *charm* hadron decay, it increases almost linearly with the instantaneous luminosities. This is also shown on Figure 7 where the *ghost rate* is represented as function of the number

[3]The timing is measured on single core 1.3 times speed of a 2.8GHz Xeon. Depending on the machine, the timing can vary within a few percent but not significantly more.

Table 2: Time spent in the Seeding algorithm on simulated $B_s \to \phi\phi$ events.

	Current LHCb	Upgrade LHCb	
	$\nu = 2$	$\nu = 3.8$	$\nu = 7.6$
time [ms/event]	18	37	172

of primary vertices (PV) of the event, as we know that an higher instantaneous luminosity implies an higher average number of primary vertices per event. This behavior of the *ghost rate* is expected as the luminosity is correlated to the occupancy of the detector: an higher occupancy translates in an higher probability of creating fake tracks simply by random combination of hits. Never-the-less it should be reminded that at this stage of the reconstruction the Kalman Filter is not yet executed: selecting good quality fitted tracks actually allows to significantly reduce the *ghost rate*. The *ghost rate* is also shown on Figure 7 as function of the relevant kinematic variables like the momentum p, the pseudo-rapidity η and the angle ϕ of the tracks. The *ghost rate* is very large for low momentum tracks, but for tracks with a momentum greater of 10 GeV it tends to increase only slightly with the momentum. It can be observed also an higher rate of ghosts in the regions around $\phi = \pm\pi/2$, which correspond to the region $x = 0$ where the occupancy is higher. This large variation of ghost as function of ϕ might indicate the need for a further tuning of the algorithm, especially in the central part of the detector, where the occupancy is the largest.

The tracking efficiencies for various track categories are shown in Table 4 for both the current detector and the upgraded one. While the upgrade samples simulate higher instantaneous luminosities values compared to the current detector simulated samples, the first estimations of the efficiencies indicate that additional work is needed to reach similar performances. As expected, the *Seeding* efficiency slightly decreases as function of the instantaneous luminosity. This is confirmed by Figure 7 where the efficiency is shown as function of the primary vertices. In Figure 7 the *efficiency* is shown as function of the momentum p, the pseudo-rapidity η and the angle ϕ of the tracks.

Table 3: Pattern recognition performance parameters for the Seeding algorithm in the current and upgraded detector on simulated $B_s \to \phi\phi$ events.

	Current LHCb [%]	Upgrade LHCb [%]	
	$\nu = 2$	$\nu = 3.8$	$\nu = 7.6$
Ghost rate	5.2	7.4	19.6
Reconstruction efficiency			
long	96.1	85.3	82.6
long, $p > 5\,\mathrm{GeV}/c$	96.6	91.7	88.4
b-hadron daughters	96.9	89.3	87.6
b-hadron daughters, $p > 5\,\mathrm{GeV}/c$	97.2	92.4	90.4

Table 4: Pattern recognition performance parameters for the Seeding algorithm in the current and upgraded detector on a sample of simulated $D^* \to D^0(\to K^0_S\pi\pi)\pi$ events. The ghost rates are identical to the ones obtained on the $B_s \to \phi\phi$ sample.

	Current LHCb [%]	Upgrade LHCb [%]	
	$\nu = 2$	$\nu = 3.8$	$\nu = 7.6$
Reconstruction efficiency			
long	96.2	84.8	82.1
long, $p > 5\,\mathrm{GeV}/c$	96.6	91.5	88.1
strange daughter with UT (TT) hits	96.1	81.7	79.5
strange daughter with UT (TT) hits, $p > 5\,\mathrm{GeV}/c$	96.6	91.2	88.4
strange daughter with UT (TT) hits from B or D	96.4	84.3	82.9
strange daughter with UT (TT) hits from B or D, $p >$ 5 GeV/c	96.9	91.7	89.7
strange daughter with UT (TT) hits from B or D and not VELO reconstructible	96.4	85.3	83.7
strange daughter with UT (TT) hits from B or D , $p > 5\,\mathrm{GeV}/c$ and not VELO reconstructible	97.0	91.7	89.7

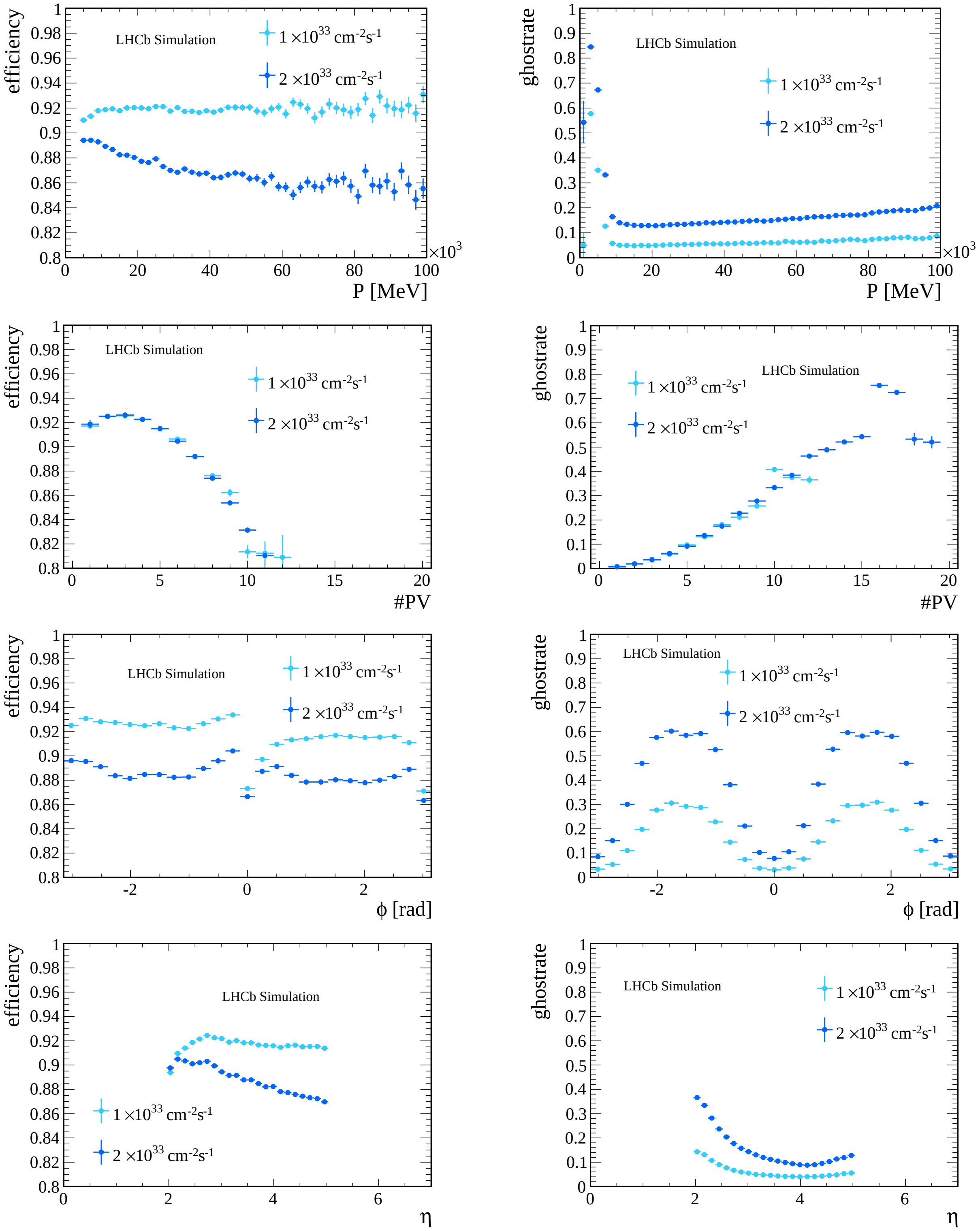

Figure 7: *Seeding* tracking efficiency (left) and *ghost rate* (right) for long tracks in bins of momentum, number of reconstructed primary vertices, ϕ and η in each row respectively.

4 Conclusion

The *Seeding* algorithm described in this note is a first implementation of a pattern recognition algorithm dedicated to the reconstruction of particles having no hits in the VELO. This algorithms relies on information provided be by the Scintillating Fiber detector, a tracking device foreseen for the LHCb upgrade. Although additional work is required to improve the efficiency of this algorithms for the various types of particles, it has been proved by a large set of detailed simulations of the detector under different running conditions, including data taking at an instantaneous luminosity up to $\mathcal{L} = 2 \times 10^{33}$ cm^{-2}s^{-1}, to satisfy the needs of the reconstruction in the expected high luminosity environment: a good reconstruction efficiency, a reasonable level of ghost rate and an acceptable time of execution. It has to be kept in mind that the performances shown in this note have to be taken as indications as they might change in the future according to the evolution of the reconstruction software.

References

[1] L. Delbuono, O.Gruenberg, D.Milanes "The Fiber Tracker description for the LHCb DDDB". LHCb-PUB-2014-004

[2] E. Cogneras et al., "The digitisation of the scintillating fibre detector". LHCb-PUB-2014-003

[3] Y.Amhis, T.Nikodem, M. De Cian, "Description and performance studies of the Foward Tracking algorithm for a scintillating fiber detector LHCb " LHCb-PUB-2014-001.

[4] R.Lindner, "Definition of the coordinate system". LHCB-C-EN-0001

[5] M.Schiller, "Track reconstruction and prompt K^0_S production at the LHCb experiment". CERN-THESIS-2011-078.

[6] O.Callot, M.Schiller, "PatSeeding - A Stand Alone Track Reconstruction Algorithm". LHCb-2008-042