Low-momentum track finding in Belle II

J. Lettenbichler, R. Frühwirth, M. Nadler, R. Glattauer
Institute of High Energy Physics, Austrian Academy of Sciences
Nikolsdorfer Gasse 18, 1050 Wien, Austria
E-mail: jakob.lettenbichler@assoc.oeaw.ac.at, rudolf.fruhwirth@oeaw.ac.at, moritz.nadler@gmx.de, robin.glattauer@assoc.oeaw.ac.at

Abstract. The Silicon Vertex Detector (SVD) of the Belle II experiment is a newly developed device with four measurement layers. Track finding in the SVD will be done both in conjunction with the Central Drift Chamber and in stand-alone mode. The reconstruction of very-low-momentum tracks in stand-alone mode is a big challenge, especially in view of the low redundancy and the large expected background. We describe an approach for track finding in this domain, where a cellular automaton and a Kalman filter is combined with a Hopfield network which finds an optimal subset of non-overlapping tracks. We present results on simulated data and evaluate them in terms of efficiency and purity.

1. Motivation

The Belle experiment was situated at the first generation B-factory KEKB in the KEK laboratories in Tsukuba, Japan. The KEKB storage ring is currently being upgraded to the new SuperKEKB collider, which will have a much larger instantaneous luminosity of $8 \times 10^{35} \text{cm}^{-2}\text{s}^{-1}$, about 40 times larger than its predecessor. In order to cope with this, the Belle detector is being upgraded, too, to Belle II. Concerning tracking, Belle II will feature a new vertex detector and a larger central drift chamber (CDC), resulting in higher vertex resolution and a broader range of reconstructed particle momenta down to $p_T \leq 50 \text{MeV}/c$. Reconstruction of low-momentum tracks will increase the signal efficiency in some interesting physics channels, such as the decay of an excited charged D meson to a neutral one in ground state and a charged pion with low momentum:

$$D^{*+} \rightarrow D^0 \pi^+.$$ 

The new vertex detector (VXD) of Belle II is designed to meet these requirements. It consists of two pixel layers (PXD) and four double-sided silicon strip layers (SVD). The design of the PXD requires on-line data reduction by defining regions of interests in the PXD that will be read-out, while the rest will be discarded.

Finding these regions of interest can be performed in stand-alone mode by a detailed analysis of the clusters in the pixel detector, but also by extrapolating tracks found in the SVD to the PXD. We have therefore started to develop a low-momentum track finder, the state of which is presented here. Its task is to reconstruct low-momentum tracks within the range of $\sim 50 \text{MeV}/c < p_T < 200 \text{MeV}/c$ in a high occupancy environment using only 3-4 hits. Tracks in this momentum range are highly sensitive to material effects such as multiple scattering and energy loss by ionization, and this has to be taken into account in the tuning of the algorithm.
Tracks with higher transverse momentum have sufficiently many hits in the surrounding central drift chamber and can be reconstructed there.

A topic closely related to low-momentum track finding is the track fitting in the VXD, which is described in another contribution to these proceedings [1].

2. The Belle II Silicon Vertex Detector

Part of the upgrade program is the replacement of the Belle Si tracker by a new one meeting the more challenging requirements of the increased physics rate and the higher machine background. The old four-layer Si detector using double-sided strip sensors will be replaced by two sub-detectors, which form together the vertex detector (VXD). The two innermost layers of the VXD are manufactured in pixel detector DEPFET technique (PXD), which allows higher vertex resolution and reduces the occupancy to manageable amounts. The PXD is enveloped by a four layer double-sided Si-strip detector (SVD), with a higher resolution than its predecessor and a slanted windmill design for better low-momentum track reconstruction. The VXD at Belle II will cover the same theta range as the SVD of the former Belle experiment, namely $17^\circ < \theta < 150^\circ$.

The other tracking detectors (SVD and CDC) will be used for on-line data reduction of the data stream produced by the PXD. High momentum tracks will be reconstructed using the CDC and SVD, while reconstruction of low-momentum tracks will have to rely on SVD data only. A schematic view of the vertex detector of Belle II is shown in Figure 1; its key features are summarized in Table 1. A more detailed description of Belle II and its physics program can be found in [2].

There will be a considerable amount of noise in the vertex detector. The double-sided sensors in the SVD will produce ghost hits if a sensor is crossed by more than track. Even more important, there will be background produced by the accelerator. The main sources of background hits are expected to be Bhabha scattering as the dominating beam-beam-scattering effect, and the Touschek effect as the dominating intra-beam-effect. Machine background will be most serious in the PXD layers.

Figure 1. Schematic view of the vertex detector of Belle II
Table 1. Some key features of the VXD. The resolution is taken from the software framework of Belle II, the rest from the Belle II TDR. The \( u \)-coordinate of the position measurement is orthogonal to the beam line, the \( z \)-coordinate is parallel to the beam line. Values are subject to change.

| Layer | Type               | Radius | Resolution \( u \) | Resolution \( z \) | Thickness |
|-------|--------------------|--------|---------------------|---------------------|-----------|
| 1     | PXD                | 13 mm  | 14.4 \( \mu m \)    | 15.9 \( \mu m \)    | 75 \( \mu m \) |
| 2     | PXD                | 22 mm  | 14.4 \( \mu m \)    | 24.5 \( \mu m \)    | 75 \( \mu m \) |
| 3     | SVD                | 38 mm  | 14.5 \( \mu m \)    | 45.1 \( \mu m \)    | 320 \( \mu m \) |
| 4     | SVD, w slanted part| 80 mm  | 21.7 \( \mu m \)    | 69.3 \( \mu m \)    | 320 \( \mu m \) |
| 5     | SVD, w slanted part| 115 mm | 21.7 \( \mu m \)    | 69.3 \( \mu m \)    | 320 \( \mu m \) |
| 6     | SVD, w slanted part| 140 mm | 21.7 \( \mu m \)    | 69.3 \( \mu m \)    | 320 \( \mu m \) |

3. Track finding strategy
The low-momentum track finder is currently developed as an off-line tool for the software framework of Belle II, \texttt{basf2}, but will be extended later to an on-line version following a similar design.

3.1. Global structure
A general problem of track finding techniques is that algorithms which result in a higher quality of the track candidates tend to be vulnerable to combinatorics. Because of this a multi-step-approach was chosen. It consists of a stepwise reduction of the combinatorial problem, thereby eventually allowing the use of sophisticated track finding algorithms. After applying several filter steps to reduce the combinatorics, a Cellular Automaton (CA) is used for further filtering and finding of track candidates. These track candidates are evaluated by a Kalman filter, which assigns a quality indicator to each of them. These are used by a Hopfield network for eliminating overlapping track candidates. The surviving ones define the set of final track candidates at this point. In the following subsections a detailed description of the various steps of the track finder is given. Figure 2 shows a schematic representation of the main parts forming the track finder.

3.2. Sectors
In the CA paradigm of track finding, track segments connecting two hits are regarded as cells (more details in Subsection 3.5). Since any combination of two hits located on two compatible layers could form such a cell, the number of cells would be too high for the CA. Discarding obviously bad combinations is therefore the most sensible way to go. To this end, each sensor is divided into sectors, and each hit is associated to a sector.

In order to suppress combinations of two hits lying in incompatible sectors, a map of compatible sectors is generated, using Monte Carlo generated tracks in the desired momentum range. Different sector maps can be generated for different momentum ranges. Each sector map is a lookup table that stores the physically valid combinations of sectors. Reducing the size of the sectors reduces the combinatorics, but increases the size of the sectors maps and therefore the memory load. Currently the sensors are divided into 4 sectors each.

The time required to assign the hits into sectors is proportional to the number of hits, so this is the fastest type of filter and is therefore applied at the beginning of each event. A useful side-effect is the possibility to chose tighter (sector-specific) cuts for the following filters, which reduces the number of allowed hit combinations again. Figure 3 illustrates the concept of the sector map and the associated reduction of combinatorics.
**Schematic view of the low momentum track finder in Belle II**

Unsorted hits from tracks, background, ghost coming from an event

- **Sector map** - 1-hit filter
  filters by set of compatible sectors, allows momentum dependent setups

- **Segment finder** - 2-hit filter
  filters by distance, min&max, including virtual Segment

- **Neighbour finder** - 3-hit filter
  filters by angle and Δ-distance min&max

- **Hopfield Network**
  uses QI’s to find best subset among overlapping TC’s

- **Clean TC’s**

  - The arrows represent a schematic interpretation of the possible number of combinations of hits at that point
  - Filters marked with an O use external information generated by simulation

---

**Figure 2.** Flow chart of the low-momentum track finder describing the currently implemented main parts.

**Figure 3.** The concept of sector maps, showing the difference between layer-wise and sector-wise cut-offs.
3.3. Segment finder
To construct the cells of the CA, two hits that lie in compatible sectors are combined to 2-hit-segments. The segment finder filters the possible combinations by various measures of distance (see Table 2), where lower and upper cut-offs are applied to get the minimal subset of useful combinations. As the efficiency of the CA increases with the number of compatible cells/segments forming a chain, the situation of typically having only 3-4 hits per track poses a challenge to the concept of the CA. As a remedy, a virtual segment is introduced that connects the interaction point with a hit in the innermost layer. Since the exact position of the interaction point at that point of event reconstruction is still unknown, a generalized guess is used. Although having only a rough guess, the existence of another cell in the line greatly improves the performance of the CA. The use of the virtual segment decreases with an increasing number of layers and is therefore best suited for setups with a small number of layers.

The number of cells grows with the second power of the number of compatible hits. However, the set of compatible hits is already reduced by the sector maps. The currently implemented filters used by the segment finder can be found in Table 2.

| name           | function                                                                 |
|----------------|--------------------------------------------------------------------------|
| dist3D         | measures the distance in 3D and filters the result by sector-dependent lower and upper cut-offs |
| distXY         | similar to distance3D, but uses only 2D (x − y-projection) information  |
| distZ          | similar again, but distance in z-direction is measured and normed to the total segment length in r − z-projection |
| normedDist3D   | distance3D/distanceXY, reduces the fringe effects which penalize tracks passing the sector near the border |

3.4. Neighbour finder
The next step is the neighbour finder, which uses the same principle as the segment finder by filtering hit combinations under chosen conditions. The neighbour finder searches for compatible pairs of segments connected by a common hit; such segments are called neighbours. To be accepted as neighbours, two connected segments are filtered by minimal/maximal angle and difference in length. Although this is a process of $3^{rd}$ order growth with the number of hits, the set of compatible hits is already greatly reduced at this point of the reconstruction process. Every cell having inner neighbours carries a list of these neighbours, which is needed for the CA. The filters used by the neighbour finder are described in Table 3.

3.5. Cellular automaton
Cellular automata are a convenient tool for pattern recognition when dealing with high occupancy and narrow time constraints [3, 4].

As indicated above, the cellular automaton assigns integer states to each segment. The states evolve in a discrete time process. In its typical implementation, all cells start with their initial state set to zero. In our implementation the neighbour finder assigns a state equal to 1 to all outer members of all neighbouring pairs, and the first iteration of the CA is skipped.

In each step of the time evolution, each cell scans its compatible inner neighbours, as found by the neighbour finder. If any inner neighbour of the current cell has the same state as the current one, the latter is scheduled for stepping up its state at the end of the time step. This
Table 3. Summary of the implemented filters used by the neighbour finder.

| name         | function                                                                 |
|--------------|--------------------------------------------------------------------------|
| angles3D     | measures the angle between two connected segments in 3D and filters the result by sector-dependent lower and upper cut-offs |
| anglesXY     | similar to angles3D, but uses only 2D ($x - y$-projection) information   |
| anglesRZ     | similar again, but angles are measured in $r - z$-projection             |
| ΔdistZ       | calculates the difference in distanceZ (see table 2) of both segments; only max cut-off is used |
| dist2IP      | calculates the radius and the center of the circle defined by the three hits of the two segments and checks the distance of the center to the interaction point; only max cut-off is used |

means that the states of the cells are checked independently and are updated simultaneously. This leads to an ever-changing neighbourhood for each cell after each time step and to chains of neighbouring segments with decreasing states. The time step is repeated until no cell changes its state any more. Finally, chains of neighbouring segments with decreasing consecutive states are stored as track candidates by the track candidate collector (TCC).

3.6. Track candidate filter

After collecting the track candidates, other filters of higher complexity can be applied. For instance, a track candidate can be checked for zig-zag patterns or kinks (represented by large changes in curvature). These filters discard physically improbable track candidates according to similar principles as the segment and the neighbour finder. The result is a set of physically plausible track candidates. The filters used by the track candidate filter are described in Table 4.

Table 4. Summary of the implemented filters used by the track candidate filter. Filters need track candidates with at least 4 hits/3 segments.

| name    | function                                                                 |
|---------|--------------------------------------------------------------------------|
| ΔPt     | calculates the curvature (function of transverse momentum) of the inner and the center segment (= 3 hits) and compares the value with the curvature of center and outer segment. |
| zigZag  | similar to ΔPt, but compares only the sign of the curvature. Alternating signs indicate bad track candidates. |

3.7. Track quality

The track candidates at this point are a set of typically partially overlapping candidates, some of which are real tracks, and some of which are ghost tracks. To find the best subset of non-overlapping track candidates, a quality indicator for each one is needed. It is planned to use a Kalman filter, which is known to be a statistically optimal track following algorithm [5, 6, 7]. In its intended version, a preliminary track fit is performed using initial guesses of the track parameters for each track candidate. The result for each track candidate is a $\chi^2$-value which can be converted into a probability that the track candidate is a real track. This probability will be used as a quality indicator (QI). The relatively small number of track candidates at this
stage allows using such a sophisticated and time-consuming technique. As the Kalman filter is not integrated yet into the track finder, the QI is emulated by using the track length as the quality attribute.

3.8. Hopfield network

The task of finding the best subset of non-overlapping track candidates is not trivial since there are two conflicting conditions describing the quality of a subset. In fact, real tracks can have worse QIs than ghost tracks or contaminated track candidates, which contain at least one extraneous hit produced by another track or by a background process. Therefore a trade-off between the highest number of chosen track candidates and the highest quality of the chosen track candidates has to be found. Although tracks with large QIs are preferred in principle, sometimes track candidates with slightly worse QIs are accepted when as a consequence more non-overlapping track candidates can be found.

To solve this combinatorial optimization problem, a neuronal network of the Hopfield type is used [8]. Track candidates are represented by neurons having a state comparable to the state of the CA, although the state of the neuron is not discrete, but continuous between zero and one. Two neurons are incompatible if their associated track candidates share hits, and are compatible if they don’t share hits. The compatibility is coded in the weight matrix of the network, and the activation thresholds depend on the QIs. The network follows the standard Hopfield dynamics with asynchronous updating in each time-step, where the neuron states are updated in random order. In order to increase the probability of finding the globally optimal solution of the problem, a thermal noise is included in the calculation of the transfer function and reduced after each time step (mean-field annealing). The network evolves until the changes of the neuronal states fall below a predefined threshold value. In the final state, neurons above a certain threshold are considered as active, and they form the desired subset of non-overlapping tracks, which are then stored for further use (track fitting and vertex finding and fitting).

4. An example

In Figures 4, 5 and 6 (see next page), the reconstruction of an event with 10 tracks having a $p_T$ between 70 and 100 MeV/c each is shown. Tracks having a $p_T > 70$ MeV/c reach the outermost layer of the SVD and therefore produce 4 hits per track when using only the SVD. To simplify the presentation, background and ghost hits are neglected. Curling tracks are not supported yet, but will be used in the final track finder. The figures show the detector and the tracks in the $r - \Phi$-plane.
(a) After the segment finder, 51 segments have been found.

(b) After the neighbour finder, 49 segments survived. The color of them represents the number of inner neighbours directly connected to the cell. black = no nb, yellow, = 1 nb, orange = 2 nbs.

(c) After the CA. The color of the cells represents the state of the cell. black = 0, yellow, = 1, orange = 2, red = 3, dark red = 4, violet = 5.

(d) After the track candidate collector. Some of the 14 track candidates are overlapping and therefore the neuronal network has to chose the optimal subset.

Figure 4. Segment finder (a) and neighbour finder (b).

Figure 5. Cellular automaton (c) and track candidate collector (d).
5. Results
To estimate the efficiency of the current version of the low-momentum track finder, two transverse momentum ranges and two detector setups have been examined, totaling four different scenarios. The following list presents the four scenarios, using setups with 3, 4, 5 and 6 layers. The filters activated for the respective scenario are listed as well. The 3- and 5-layer setup uses a sample of low-momentum tracks between 60 MeV/c and 70 MeV/c. The 3-layer sample contains only SVD hits, the 5-layer sample has hits from SVD and PXD. The 4- and 6-layer setup uses higher momentum tracks between 70 MeV/c and 100 MeV/c. The 4-layer sample contains only SVD hits, the 6-layer sample has hits from SVD and PXD.

- **Low: 60 – 70 MeV/c, 3 layers (w/o PXD)**
  filters activated: distZ, normedDist3D, ΔdistZ, anglesRZ, ΔPt, zigZag

- **Low: 60 – 70 MeV/c, 5 layers (with PXD)**
  filters activated: dist3D, distXY, distZ, normedDist3D, ΔdistZ, angles3D, anglesXY, anglesRZ, ΔPt, zigZag

- **High: 70 – 100 MeV/c, 4 layers (w/o PXD)**
  filters activated: distXY, distZ, normedDist3D, ΔdistZ, angles3D, anglesXY, anglesRZ, ΔPt, zigZag

- **High: 70 – 100 MeV/c, 6 layers (with PXD)**
  filters activated: dist3D, distXY, distZ, normedDist3D, ΔdistZ, angles3D, anglesXY, anglesRZ, ΔPt, zigZag

Each sample contains 1000 events with muon tracks in the theta range $20^\circ < \theta < 145^\circ$. No noise hits were generated. The Kalman filter was not yet available, so the Hopfield network was
disabled. Curling tracks are not supported yet and therefore truncated after leaving the SVD. Material effects such as multiple scattering and energy loss were activated, the production of secondary particles was deactivated.

The typical number of tracks per event in Belle II will be 10, therefore samples with 10 or 20 tracks per event were generated. The results are shown in Table 5. The efficiencies are the ones after the track candidate collector (TCC). The track finder works perfectly, as is to be expected under ideal conditions, with the exception of the 3-layer scenario, where the efficiency reaches only about 90%.

Table 5. Track finding efficiency of the low-momentum track finder under different conditions.

| Momentum range | # of layers | # of tracks | results post TCC | rec_{tot} |
|----------------|-------------|-------------|------------------|-----------|
|                | clean       | cont.       | lost             |           |
| Low            | 3           | 10 000      | 88.9% 0.48% 10.7% | 89.3%     |
|                | 20 000      | 88.2% 1.1% 10.8% | 89.2%     |
| Low            | 5           | 10 000      | 99.6% 0.1% 0.3%  | 99.7%     |
|                | 20 000      | 99.1% 0.3% 0.6% | 99.4%     |
| High           | 4           | 10 000      | 99.6% 0.1% 0.4%  | 99.6%     |
|                | 20 000      | 99.5% 0.1% 0.4% | 99.6%     |
| High           | 6           | 10 000      | 99.6% 0.1% 0.3%  | 99.7%     |
|                | 20 000      | 99.4% 0.2% 0.4% | 99.6%     |

6. Summary and outlook
To meet the requirements of the special situation of low-momentum track finding, a multi-step concept has been developed using powerful pre-filtering, a Cellular Automaton, a Kalman filter and a Hopfield network. The support of different momentum ranges using sector-maps allows fine-tuned tools especially for the region below 75 MeV/c. First tests show promising results, and therefore the concept is currently being implemented in the software framework of Belle II, basf2.

Important steps in the future will be the examination of the behavior of the track finder in high occupancy situations under realistic conditions and the combination with further event reconstruction steps like track fitting and vertex finding and fitting. The integration of the Kalman filter, support for curling tracks, and obligatory performance tuning will complete the low-momentum track finder for the framework. A follow-up project will convert the concept to an on-line version to be used for defining regions of interest in the PXD and thereby assisting data reduction.

References
[1] Nadler M and Frühwirth R to appear in 2012 Proceedings of CHEP 2012 Journal of Physics: Conference Series
[2] Abe T et al. October 2010 Belle II Technical Design Report Belle II collaboration, KEK http://arXiv.org/abs/1011.0352v1
[3] Abt I, Kisiel I, Masciocchi S and Emelyanov D 2002 Nucl.Instrum.Meth. A489 389–405
[4] Abt I, Emelyanov D, Gorbunov I and Kisiel I 2002 Nucl.Instrum.Meth. A490 546–558
[5] Frühwirth R 1987 Nucl.Instrum.Meth. A262 444–450
[6] Frühwirth R and Strandlie A 2011 Landolt-Börnstein New Series, Group I: Detectors for Particles and Radiation. Part 1: Principles and Methods vol 21B1 (Springer-Verlag Berlin Heidelberg) chap 4.3 Pattern Recognition and Reconstruction

[7] Strandlie A and Frühwirth R 2010 Rev. Mod. Phys. 82 1419–1458

[8] Frühwirth R 1993 Comput. Phys. Commun. 78 23–28