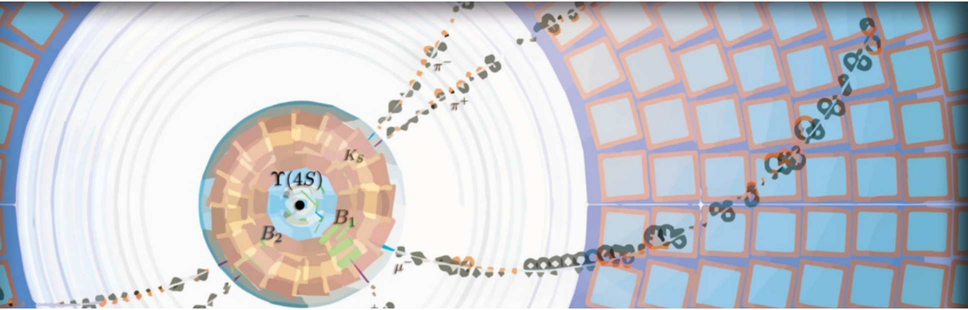
CDC cellular automaton track finding.

F2F Meeting - Vienna 2015



Oliver Frost
Deutsches Elektronen-Synchrotron (DESY)

2015-04-21

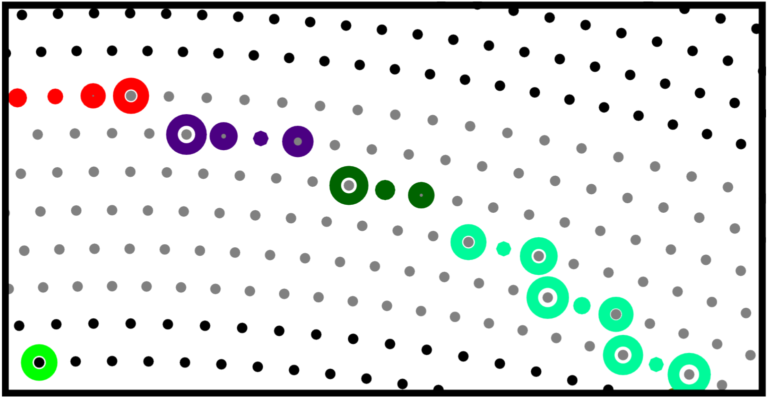
DESY HELMHOLTZ ASSOCIATION Belle II

Overview

- Hit inefficiencies
- Filters
- XT-relation with partial information

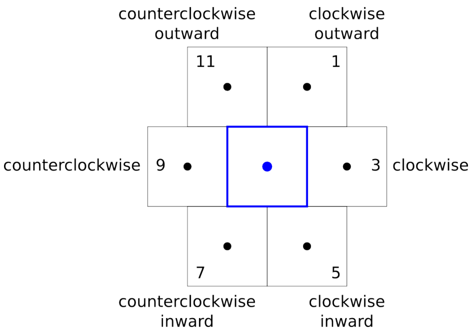
Hit inefficiencies

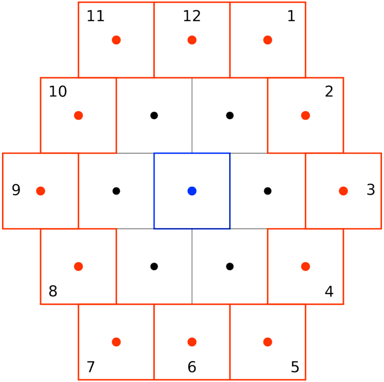
Hit inefficiencies splits clusters



- Shown above the cluster generated
- by expansion over the closest neighboring wires.
- The realistic xt-relation leads to gaps in the train of hits.
- Puts a lot of pressure on the accuracy in the segment generation stage
- (Frame choosen to be particularly bad)

Primary and secondary wire neighborhood





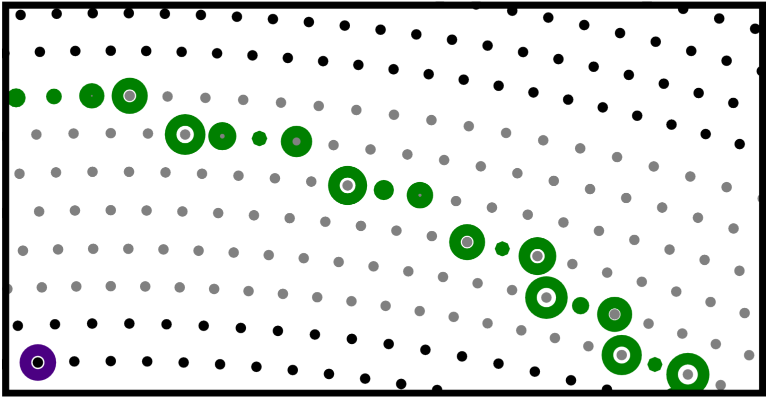
Primary neighborhood

- 6 neighbors at max
- Allows constrained to a superlayer

Secondary neighborhood

- Additional 12 neighbors at max

New concept superclusters



- Secondary wire neighborhood
  - bridges the hit inefficiencies.
  - generates bigger **superclusters**

How to us the secondary wire neighborhood ?

Creation of facets

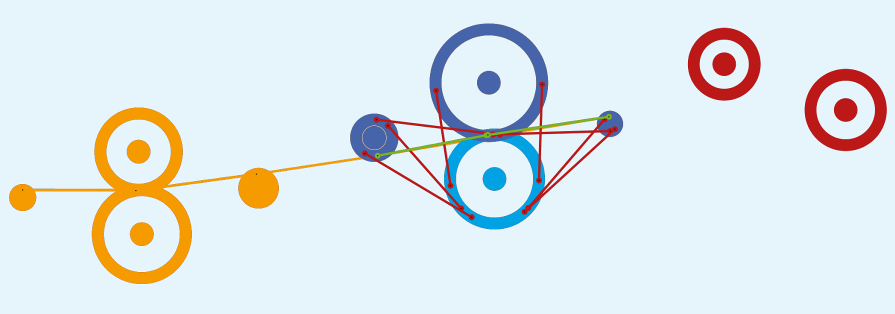
- The creation of the hit triplets may not use the (full) secondary neighborhood, because the combinatorics is to high.
- Partial secondary wire neighborhood could be activated by a *hit inefficiency marker* assigned during the supercluster generation on the suspicous wires.
- Accepting secondary neighbors leads to a wide varity of hit triples, which need adjusted filters.

Segment merging

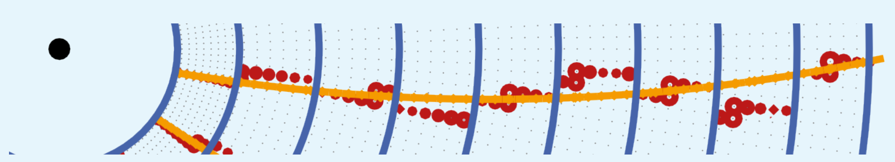
- Currently the supercluster only delimit the segments considered for merging.

Filters

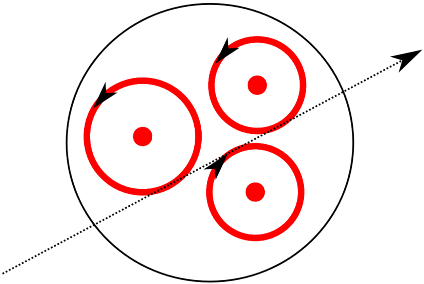
Build segments from individual hits in each super layer

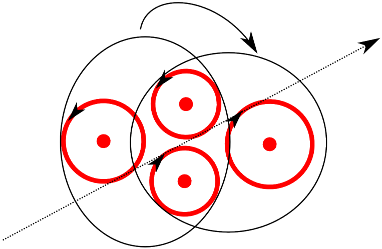


Build tracks from segments



Segment building stage - components of the graph



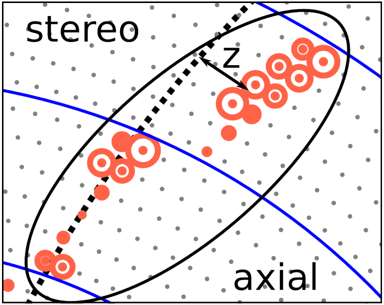


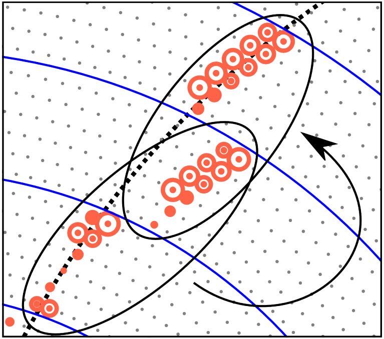
Vertices - θ_i

- Ordered triple of close by hits triangulating the suspected position
- Assumed right left passage hypotheses
- Three tangential trajectories

Edges - w_{ij}

- Neighboring triplets share two hits
- Filter to be adjusted





Vertices - θ_i

- Ordered pairs of segments axial $\leftrightarrow$ stereo
- Circle fit by Riemann technique + linear sz-fit
- $\rightarrow$ full helix with uncertainties

Edges - w_{ij}

- Neighboring segment pairs share one segment
- Filter to be adjusted

Filters

Segment creation stage

> Rejection of background clusters
> Acceptable hit tiplets : fitless and using tangential fits
> Acceptable hit triple neighbors
> Segment merging in superclusters

Track creation stage

> Acceptable segment pairs (triples) : fitless and using Riemann fits
> Acceptable segment pairs (triples)
> Track merging
> Pickup of leftover segments / hits

Not present algoritmically load

> Merging and pickup of hits have to be coded

Training sample and filter adjustments

Realistic XT-relation

- rendered all previous adjustement suboptimal

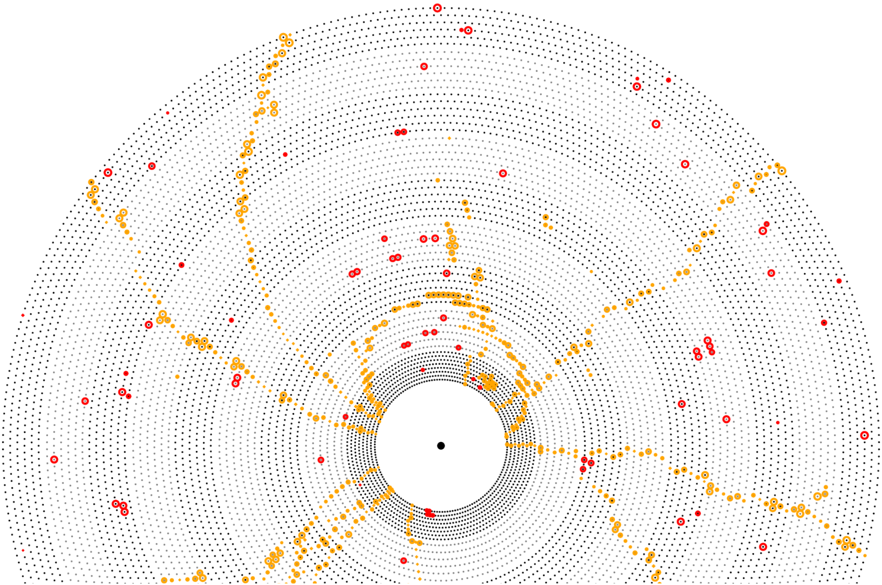
Readjustment of filters

- Every single one has to be adjusted in a supervised learning procedure comparing to Monte Carlo information.
- As much repeatable as possible in case the simulation, input hit set or desired output changes.

Recording filters

- record each instance passed to the filter
- writing a set of variables and the truth information to ROOT tree.
- Prove of concept for background cluster rejection.

Background detection on clusters



Supervised learning for filters by

Human neural networks aka. bachelors

- Person looking at histograms and figuring out cuts
- Learning procedure time consuming and only partial repeatable in fixed set of parameters.
- Implementation can be made fast
- Required where evaluations are frequent
- E.g. hit triplet creation and neighborhood construction.

TMVA

- Learning procedure repeatable
- Promising where decisions may consider many variables but comparibly few evaluations are needed.
- E.g. track merging, segment merging, track generation stage.

Modules reorganised

Developement modules

- Filters are fully replaceable and configurable at runtime.
- *Virtual dispatch used*
- SegmentFinderCDCFacetAutomatonDev for segment creation
- TrackFinderCDCSegmentPairAutomatonDev for track creation from segments
- TrackFinderCDCAutomatonDev for full execution

Production modules

- Fixed filters and fixed parameters to be used as is
- *No virtual dispatch*
- Currently for regular events (TrackFinderCDCAutomaton) and cosmic events (TrackFinderCDCCosmic).

Same code base

- Achived by template + virtual final.

XT-relation with partial information

Full parameterised xt-relation

Influential variables for the drift time reconstruction

1. TDC Count
2. Layer number
3. Right-left passage
4. Approach direction to wire (parameterised as α)
5. Transverse momentum
6. Z position of the closest approach
7. Polar angle of the trajectory
8. (Time of flight)

Information waste land

- At the beginning of track finding **no knowledge** on the actual trajectory is present.
- but track finding is rather sensitive to incorrect drift times (especially the cellular automaton finder in the frist stage)

XT-relation with partial information

Requirements of track finding

- The amount of aggregated information gradually increases in several steps.
  - Cellular automaton finder with distinct steps at facet creation, segment creation and fitting, associate axial and stereo segments.
  - Legendre finder with distinct steps at finding, 2d-fitting, association of stereo hits, 3d-fitting.
- Approximately fits of trajectory are utilized to extrapolate between close by hit groups.
- *Fit accuracy depends on good estimates of the drift length.*
- The latter is especially true for hit triples

Proposal

- Have estimates of drift length and variance based on part of the information, where unknown quantities have been marginalised.
- $\rightarrow$ **Drift length estimates should improve in parallel according to the amount of knowledge achieved.**

Desired Drift length functions

Influential variables for the drift time reconstruction

In the order in which they become available

1. TDC Count
2. Layer number (how big is the difference?)
3. Right-left passage (how big is the difference?)
4. Approach to wire α
5. Transverse momentum
6. Z position of the closest approach
7. Polar angle of the trajectory

> As a maximal wish each continuous subset 1.-3., 1.-4., 1.-5., 1.-6., 1.-7. would be translated by a function.

> **The question is what variable has the biggest impact here?**

> Some plots would be nice to qualitatively understand the severity of each of the variables.

Functions needed I

Definitely

- getDriftLength(tdc, wireID, rlInfo)

Most definitely

- getDriftLengthVariance(tdc, wireID, rlInfo)

If big difference

- getDriftLength(tdc, wireID, rlInfo, α)
- getDriftLengthVariance(tdc, wireID, rlInfo, α)

If big difference

- getDriftLength(tdc, wireID, rlInfo, α, p_t)
- getDriftLengthVariance(tdc, wireID, rlInfo, α, p_t)

Functions needed II

Definitely

- getDriftLength(tdc, wireID, rlInfo, α, p_t, z, θ)
- getDriftLengthVariance(tdc, wireID, rlInfo, α, p_t, z, θ)

Concerning Genfit

- Convenience interface for Genfit involving TVector3s is reasonable.
- However depending on the underlying implementation of the drift time function, it would be tedious to translate the above variables to TVector3s and back.

“Native” parameterization

- Expose the native parameterization as a function.
- *Helps in the understanding of the main influences.*

> To make an informed decision we need to understand the influence of each of the variables better.

Bucket list

- Implement recording filters
- Underpin them with a TMVA method
- Investigate a from of the XT-relation with partial information
- Come up with procedure to bridge hit inefficiency gaps