Low momentum track finding in the Silicon Vertex Detector of Belle 2

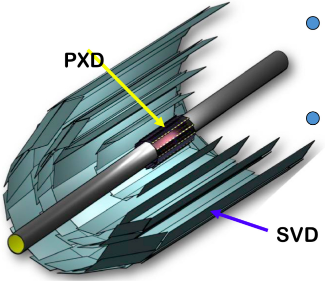
Jakob Lettenbichler, Rudolf Frühwirth

Institute of High Energy Physics
Austrian Academy of Sciences

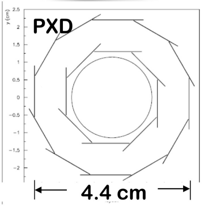
November 29, 2012

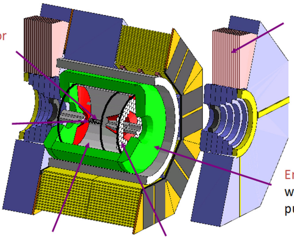
Belle II

Nano beam option: 1 cm radius of beam pipe



- **2 layer Si pixel detector (DEPFET technology) (R = 1.3, 2.2 cm) monolithic sensor thickness 50 μm (!), pixel size ~50 x 50 μm²** ← **„PXD“**
- **4 layer Si strip detector (DSSD) (R = 3.8, 8.0, 11.5, 14.0 cm)** ← **„SVD“**





1

Belle 2

Important for Track Finding

- new Si detector (windmill, slanted for small θ) for Si-only track finding
  - SVD: 4 layers (double sided strips $\rightarrow$ fast but ghost hits)
  - PXD: 2 layers pixel $\rightarrow$ slow but no ghosts, higher resolution)
- reconstruct low momenta ($p_T \geq 50\text{MeV}/c$) using 3-4 layers
- higher luminosity, 5x10^8 bunch-crossings/s, 30k events/s, 10 tracks each
- therefore higher background (Touschek, Bhabha scattering)

Table: *Resolution taken from the software framework, the rest from the TDR. The u-coordinate of position measurement is orthogonal and the z-coordinate is parallel to the beam line. Values are subject to change.*

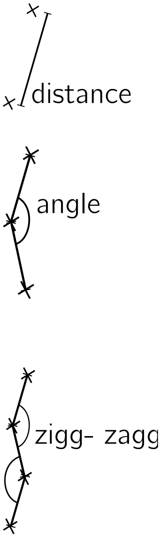
Layer	Type	Radius	Resolution u	Resolution z	Thickness
1	PXD	13 mm	14.4 µm	15.9 µm	75 µm
2	PXD	22 mm	14.4 µm	24.5 µm	75 µm
3	SVD	38 mm	14.5 µm	45.1 µm	320 µm
4	SVD, w slanted part	80 mm	21.7 µm	69.3 µm	320 µm
5	SVD, w slanted part	115 mm	21.7 µm	69.3 µm	320 µm
6	SVD, w slanted part	140 mm	21.7 µm	69.3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

Before we start...

Definitions

- *sector*: subunit of a sensor
- *friends*: two compatible sectors combined
- *segment*: two compatible hits combined
- *neighbours* (NBs): two compatible segments combined
- *hits*: 1u + 1v SVDCluster = 2D info
- *effifiency*: is 100%, when all tracks which produced at least 3 hits could be reconstructed
- a track is is successfully reconstructed through a TC, when at least 3 hits of the track have been found and less than 30% of the hits are garbage

Approach for reducing combinatorics



Schematic view of the low momentum track finder in Belle II

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

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

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

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

L *Cellular Automaton*
evolving states, includes TC-collector

O L *Post 4-hit filter*
filters by zigZag, ΔpT, ...

Kalman filter
calculates QI's

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*
*- Steps marked with an **L** cycle through several passes*

Filters

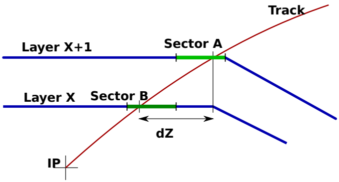
Motivation using filters:

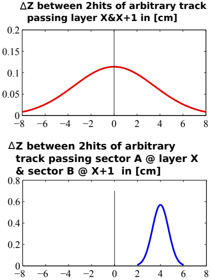
- single hits are combined to segments which form TC's when connected $\rightarrow$ combinatorial problem
- gradually filtering reduces combinatorics with increasing complexity
- using fastest filters first $\rightarrow$ starting with sectormap (see following slides) as a single hit-filter
- using typical CA-filters with min- and max-cutoffs (2-hit: hit-distance, 3-hit: angle of linked segments, 4-hit: zig-zag, Δp_T)
- quality of chosen cutoffs is essential:
  - to loose $\rightarrow$ more ghosts, slower
  - to narrow $\rightarrow$ lose real tracks
  - what about momentum dependency?

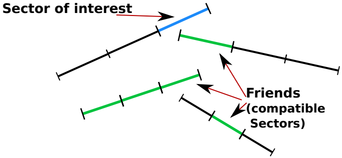
Sectors

Motivation using sectors:

- windmill structure and slanted sensors forbid simple layer-wise cutoffs → at least sensor-specific cutoffs needed
- better: subdividing sensors in sectors and storing friend-lists
- → allows customized cutoffs for filters to reduce combinatorics
- → allows multipass optimizing for different momenta and curling tracks







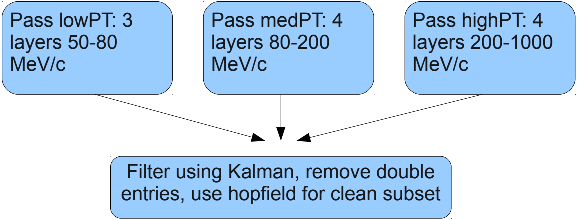
store info in a sectorMap


```xml
<sectorList_high_SVD>
  <!--This file contains the list of neighbouring sectors (friends) of the relevant sectors. Needed for the cellular automaton-->
  <aSector>
    <sectorOfInterest>41_34624_1</sectorOfInterest>
    <friendList>
      <aFriend>
        <FriendOfInterest>30_25664_6</FriendOfInterest>
        <filterList>
          <aFilter>
            <FilterOfInterest>deltaPt</FilterOfInterest>
            <quantiles>
              <Min>6.94055997883e-05</Min>
              <Max>153255002112.0</Max>
            </quantiles>
          </aFilter>
          <aFilter>
            <FilterOfInterest>angles3D</FilterOfInterest>
            <quantiles>
              <Min>0.0138595186807</Min>
              <Max>1.64301598306</Max>
            </quantiles>
          </aFilter>
          <aFilter>
            <FilterOfInterest>pT</FilterOfInterest>
            <quantiles>
              <Min>0.0101334603041</Min>
              <Max>1.35652917044</Max>
            </quantiles>
          </aFilter>
```

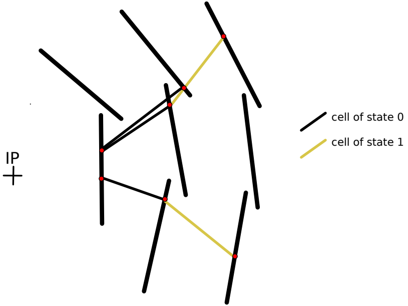
Multi pass support

Motivation using multi pass support

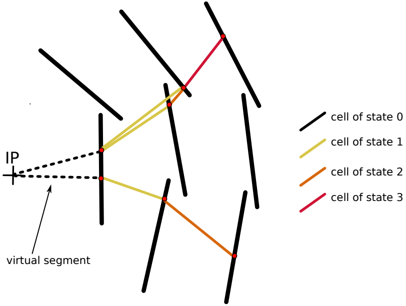
- allows several passes for same hits (of same event) using different sector setups and individual cutoffs
- number of passes arbitrary
- *missing:* hit removal and curling track pass, both will take some time for implementation



Adapting CA principle for 3-4 layers, virtual segment and sectors



basic concept of cells



extended concept using virtual segments attached to the IP and sectorMaps for segments in overlapping parts

Kalman filter and Hopfield network

Constructing and cleaning TCs

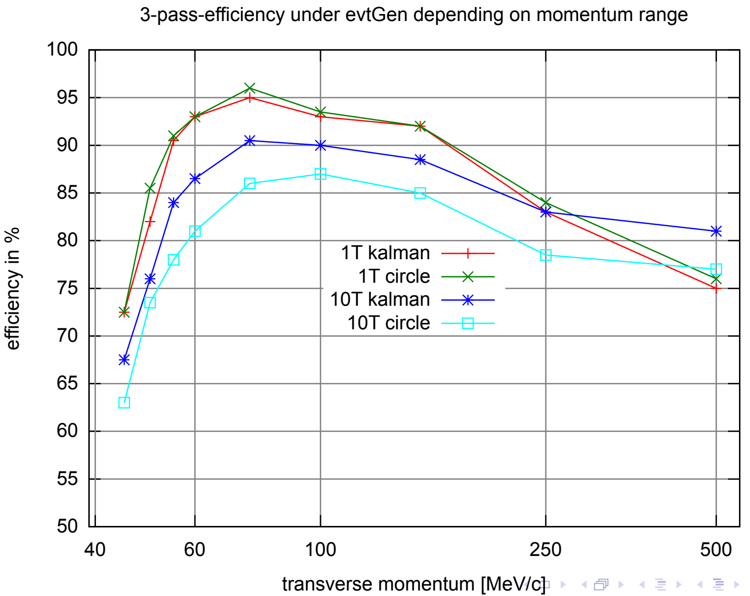
- New TCs start with a seed (cells with high states), grows inwards by attaching cells with decreasing value of state
- after the TC-collector (TCC), overlapping TCs are filtered to find clean subset
- post-TCC-Filter applies simple rules like zigg-zagg or Δp_T
- a Kalman filter calculates a quality index → probability that current TC is a real track
- a Hopfield network searches clean subset of overlapping TCs → uses QI's for decision-making

Test setup for preliminary results

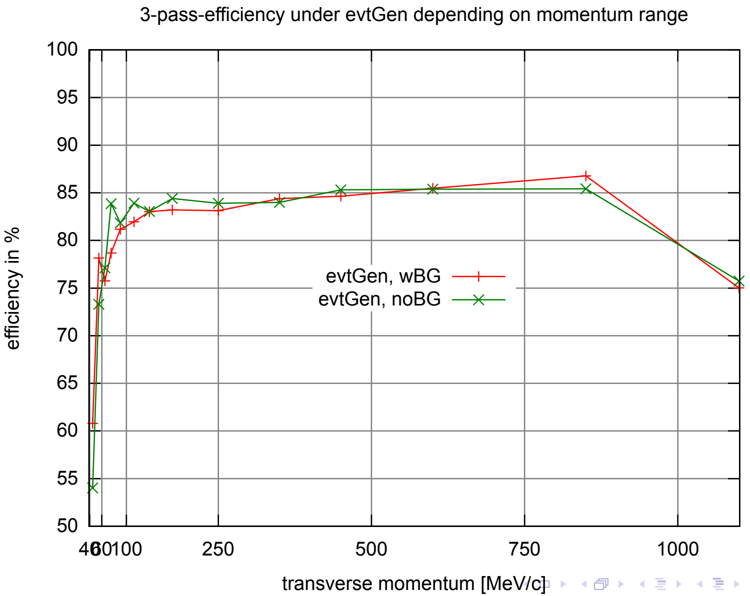
settings - beware alfa-release!

- pGun: 1track(T)/10T pions with $45\text{MeV}/c \leq p_T \leq 500\text{MeV}/c$ without (1T) and with BG (10T)
- in the following slides "low" means $60\text{MeV}/c$ (= 13cm circle radius), "high" means $500/c$ (= 100cm circle radius)
- evtGen: standard setting ($\Upsilon(4S)$) with and without BG. Efficiencies calculated for different momentum ranges
- all runs: SVD only
- 3 pass: low: 3 layers, $45\text{MeV}/c \leq p_T \leq 80\text{MeV}/c$, std: 4 layers, $75\text{MeV}/c \leq p_T \leq 200\text{MeV}/c$, high: 4 layers, $180\text{MeV}/c \leq p_T \leq 1200\text{MeV}/c$
- $16^\circ \leq \theta \leq 151^\circ$
- 6 sectors per sensor - u-boundaries (normed to sensor size): 0, 0.5, 1. v-boundaries: 0, 0.33, 0.67, 1

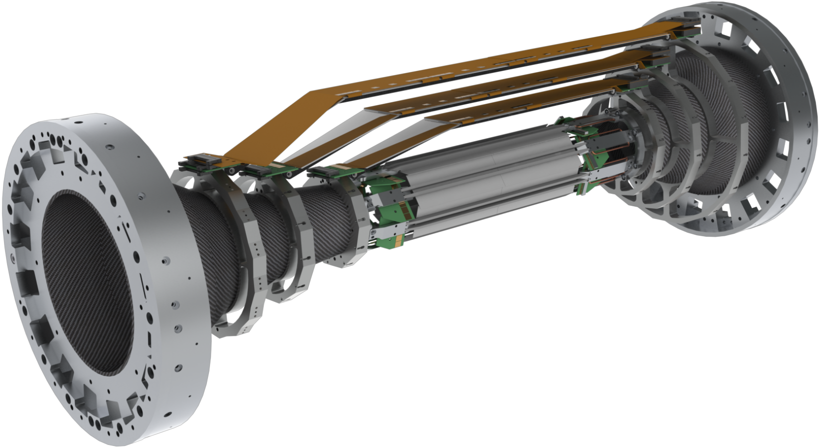
efficiencies w 1T and 10T pions



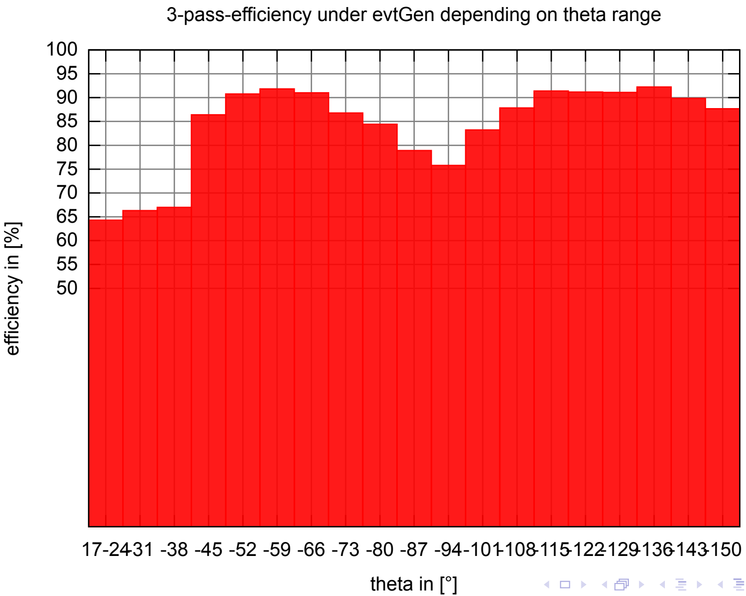
efficiencies with evtGen



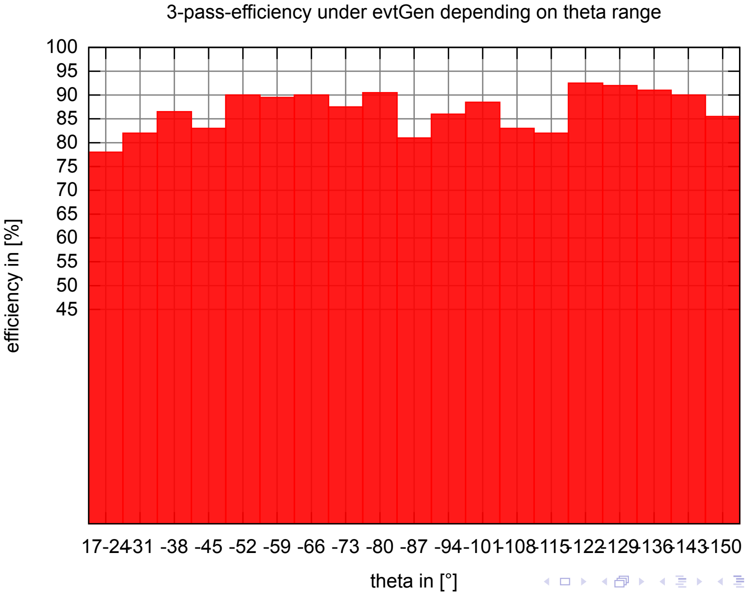
SVD - slanted parts



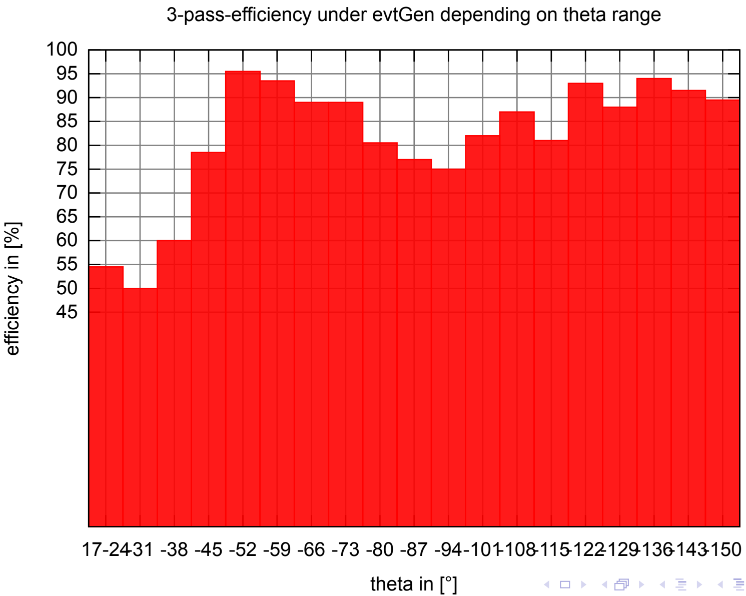
evtGen - dependency of θ



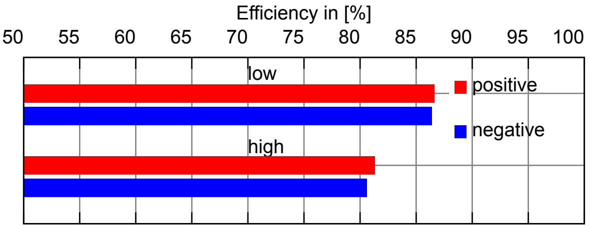
low p_T - dependency of θ, small effect in slanted parts



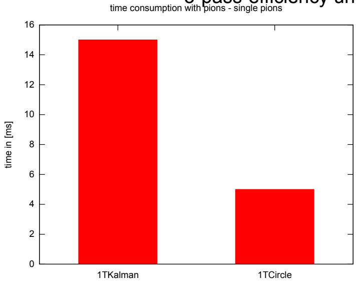
low p_T - dependency of θ, big effect in slanted parts

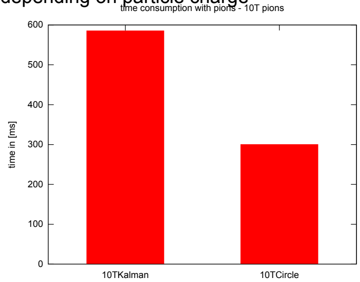


charge - dependency and time consumption

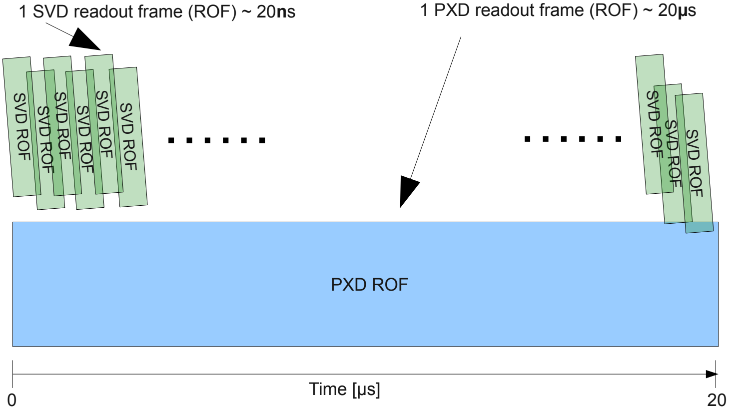


3-pass-efficiency under evtGen depending on particle charge





measured with my 7 year old notebook, 1,66GHz Core(1) Duo, debug mode



Online-specs

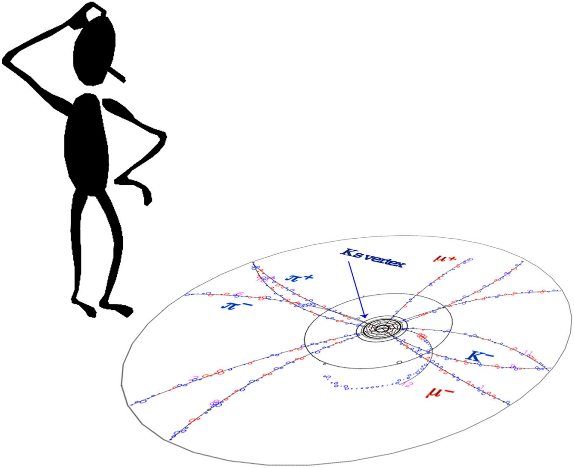
- PXD 1000 times slower than SVD and has rolling shutter readout
- roughly 5000 PXDclusters per ROF $\sim$ 25 real ones, rest: background
- low momentum TF needed for online data reduction

Outlook

ToDo

- checking PXD-performance under realistic event- and background conditions
- support curling tracks (idea: use an additional setup for curling tracks and combine afterwards)
- fine-tuning for sectormaps, filters and hit-removal (new student will do part of this work)
- developing online version (my future main project, starts 2013Q1)

that's all, folks!



Any suggestions, ideas or requests?

Jakob.Lettenbichler@oeaw.ac.at