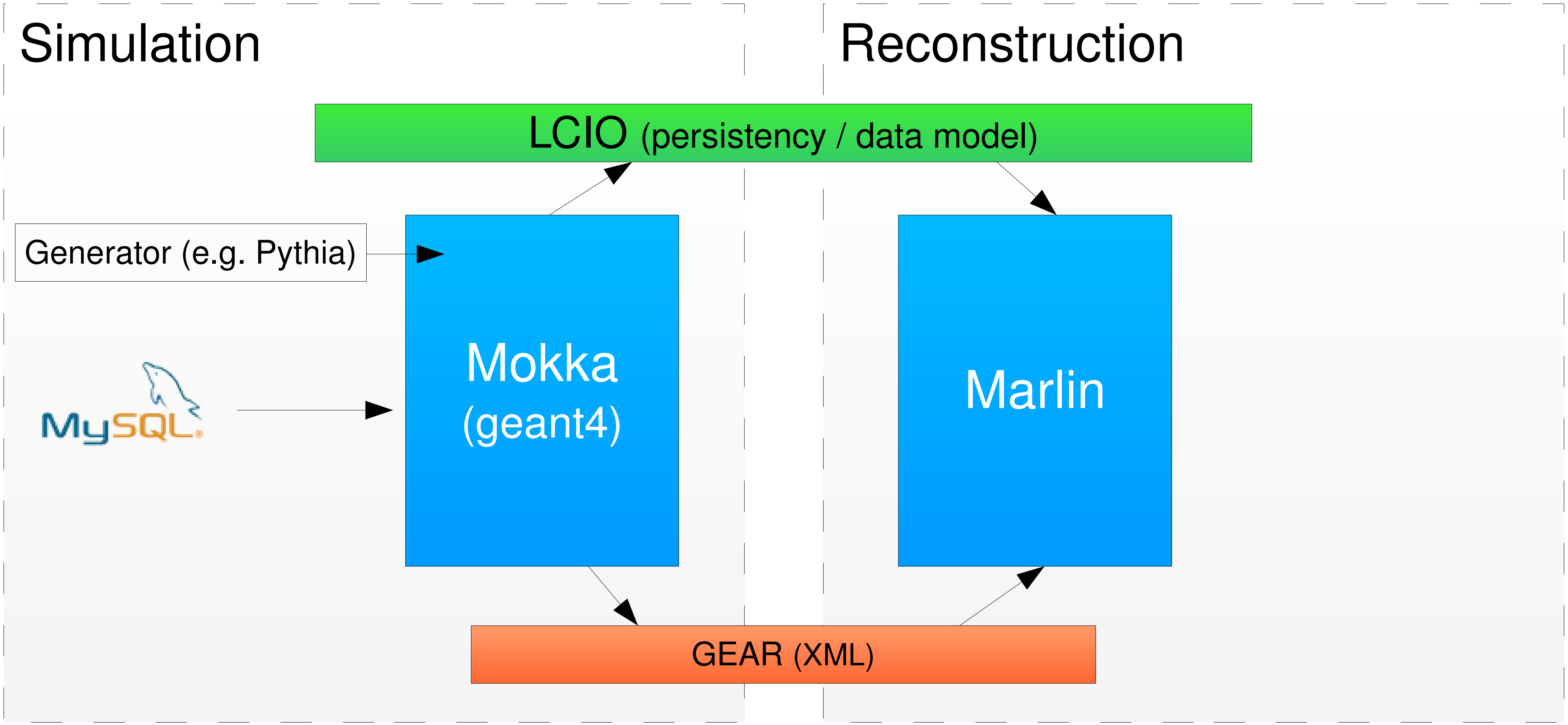
Simulation Studies of DEPFET Vertex Detector for SuperBelle Using ILC Software

Alexei Raspereza
Andreas Moll
Pit Vanhoefer

DEPFET Collaboration Meeting
Heidelberg 11.09.2008

- In April/May 2008 ILC group at MPI put forward initiative to join Belle/SuperBelle experiment
- The decision to participate in Belle/SuperBelle has been approved by MPI directorate
- Hardware contribution to SuperBelle project : DEPFET based vertex detector
- Near future activities
  - Contribution to physics analysis at Belle
  - Performance & optimization studies of silicon vertex detector for SuperBelle
- The latter endeavor builds upon mature and flexible ILC software which can be used to study performance of SuperBelle detector and optimize its design



Mokka is geant4 based framework for full detector simulation

LCIO is a persistency framework that defines a common data model

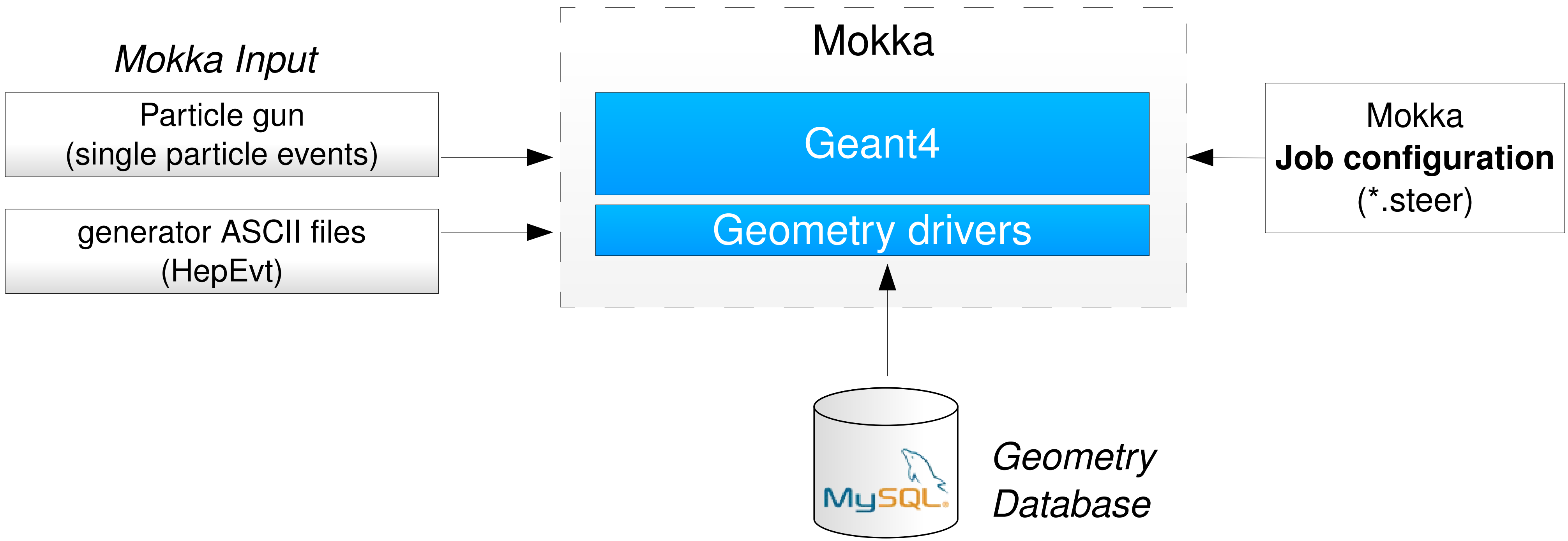
Marlin is modular C++ application framework based on LCIO

GEAR: one source of geometry. Mokka creates geometry xml files used in Marlin

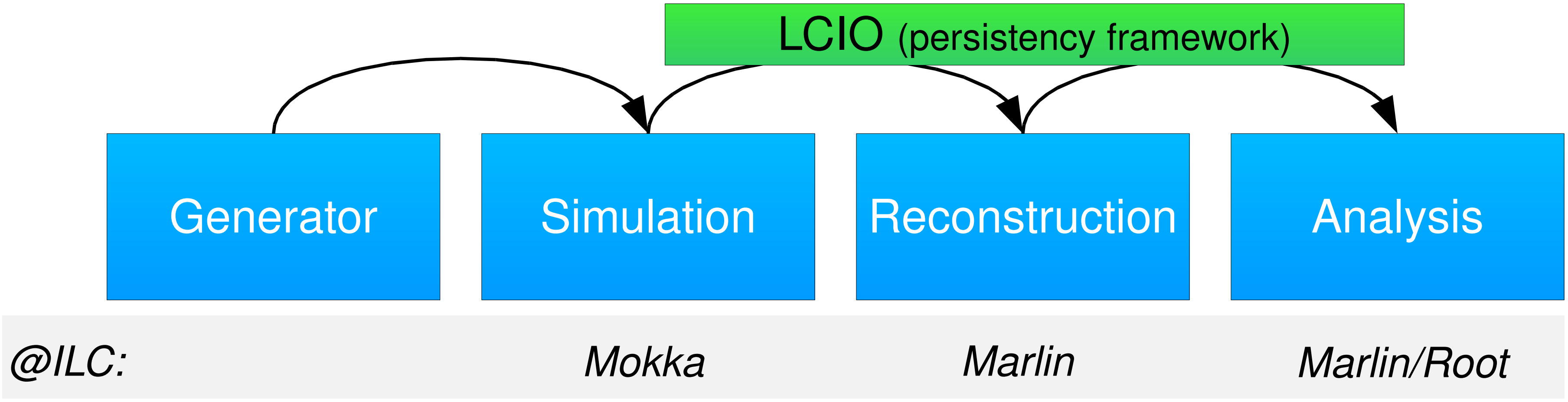
What is Mokka ?

- a **Geant4** based full detector simulation (written for the ILC project)
- developed at ecole polytechnique (P. Mora de Freitas)
- uses a **database** (MySQL) and C++ code (**drivers**) for the description of detector geometry
- Writes **LCIO** files
- abstract geometry layer (same geometry could be used for reconstruction and analysis)

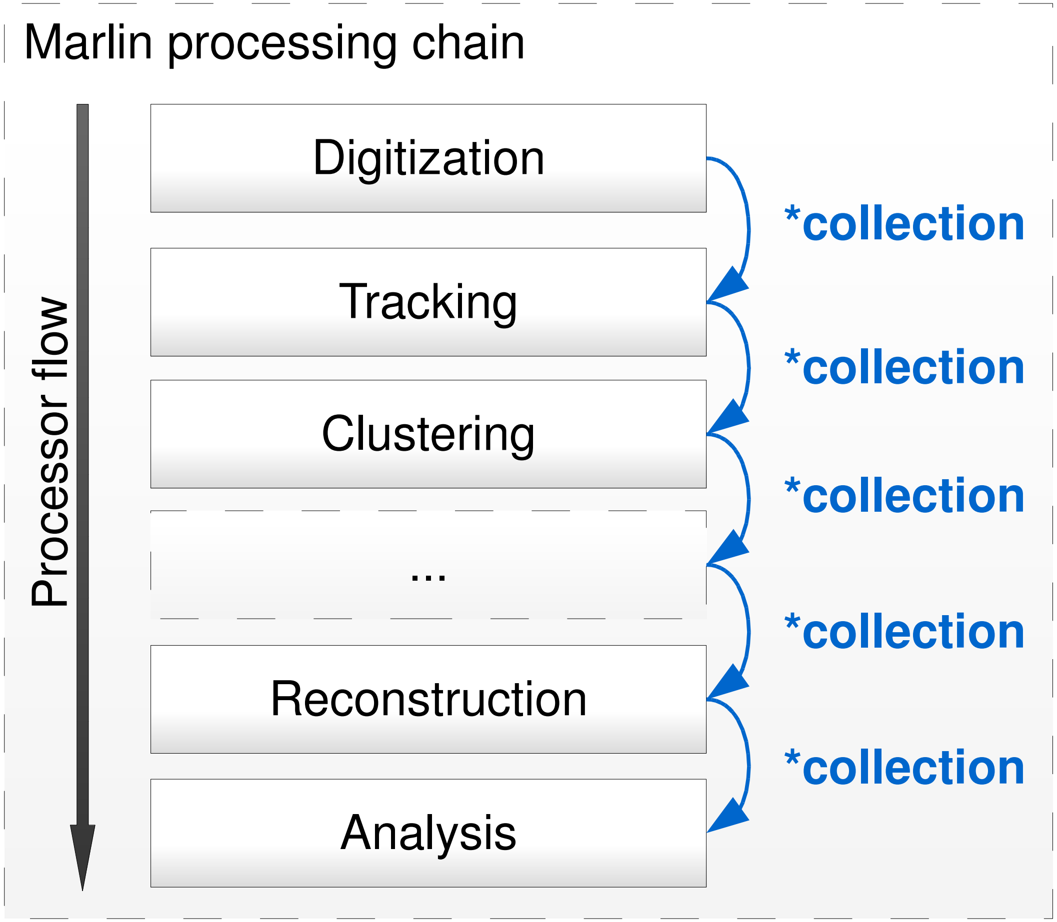
  → **Gear files**



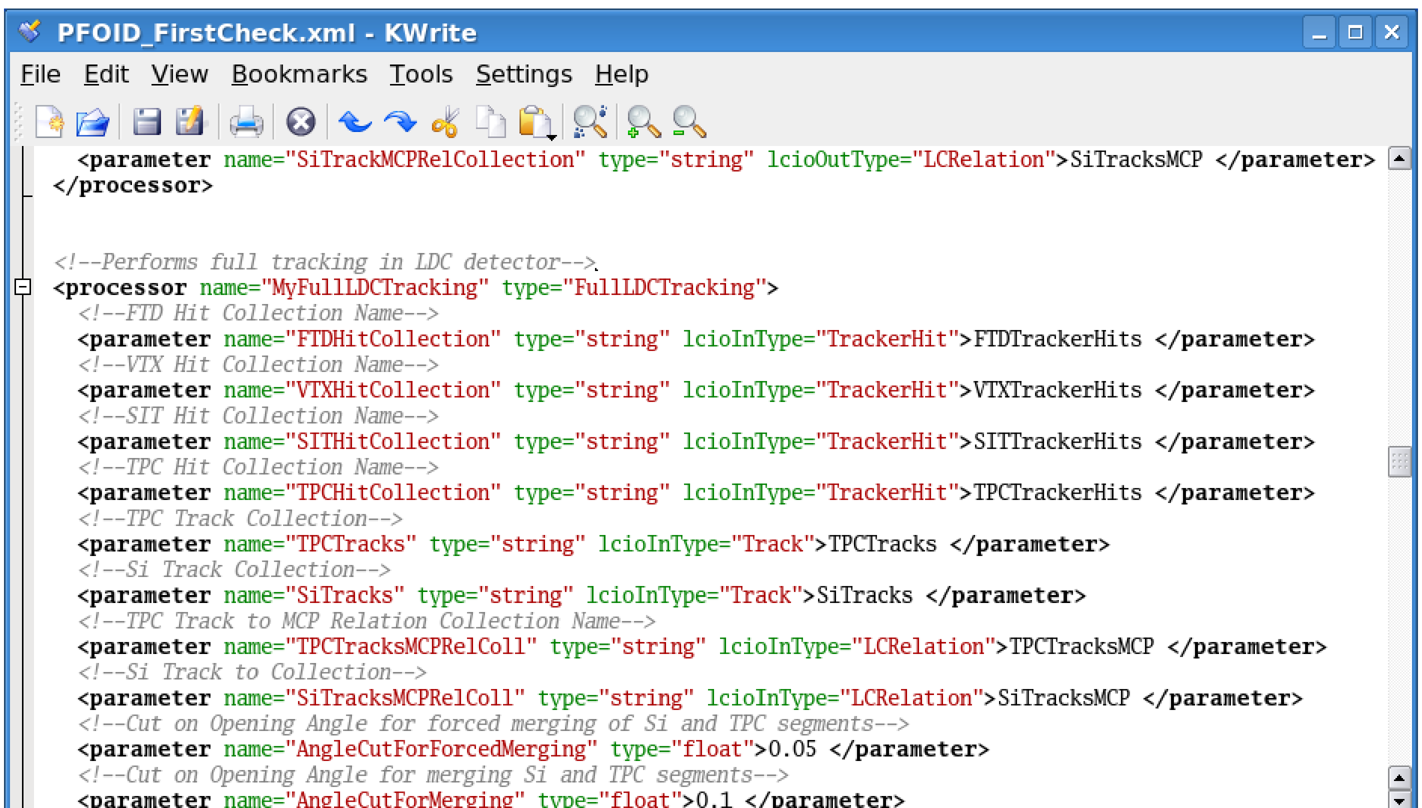
- LCIO is a persistency framework for linear collider simulation studies
- LCIO provides a common **event data model**
- LCIO event class serves as internal **data container** in Marlin
- data container is called **collection**
- LCIO used for simulation studies and analysis frameworks
- API (interface) for Java, C++, f77 exists



- Marlin is a modular and **detector independent** C++ application framework for the analysis of LCIO data.
- every computing task is implemented as a module (called **processor**).
- The LCIO event class serves as a container (**collection**) of the data, that is shipped between the processors.

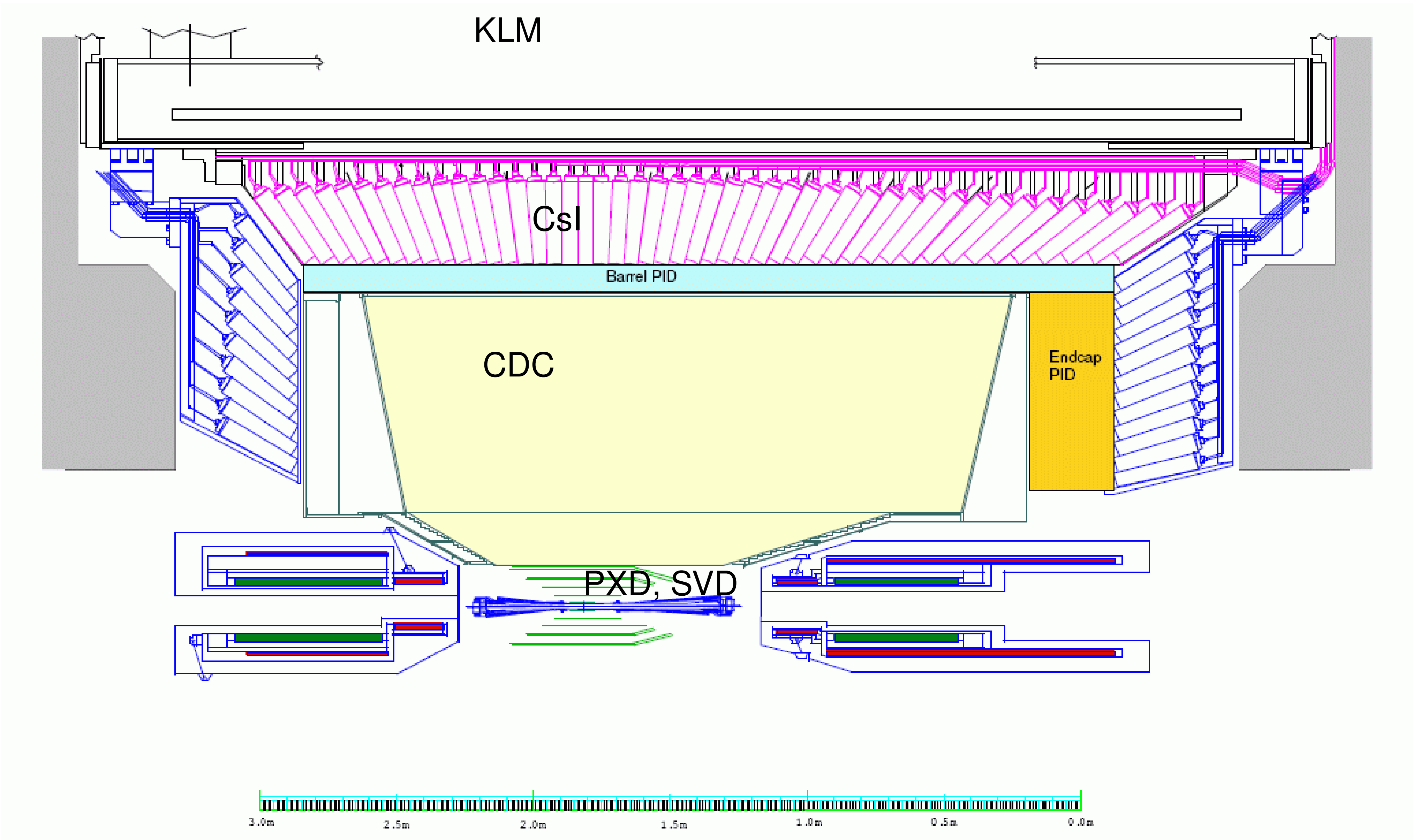


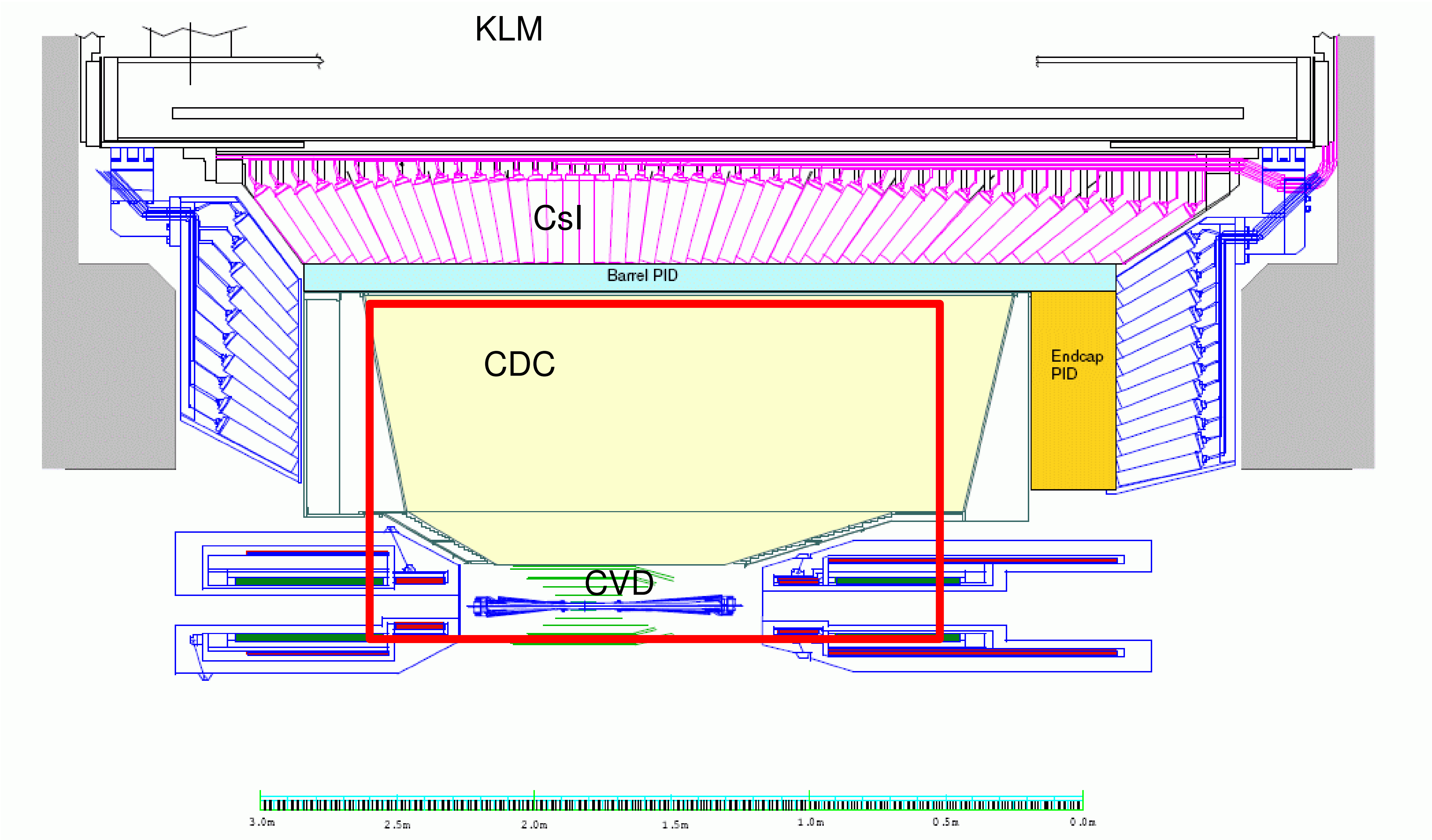
- Marlin allows to define the processors (and their order) that are executed at runtime in a simple, **XML** based, steering file (“Plug & Play”).
- Via the steering file, **named parameters** (string, float, int - single and arrays) are defined for every processor.
- Marlin performs consistency check of input/output collection types.
- Writing **own processors** (e.g. for Analysis) is simple, just implement defined callback functions, i.e. init(), processRunHeader(), processEvent(), end().



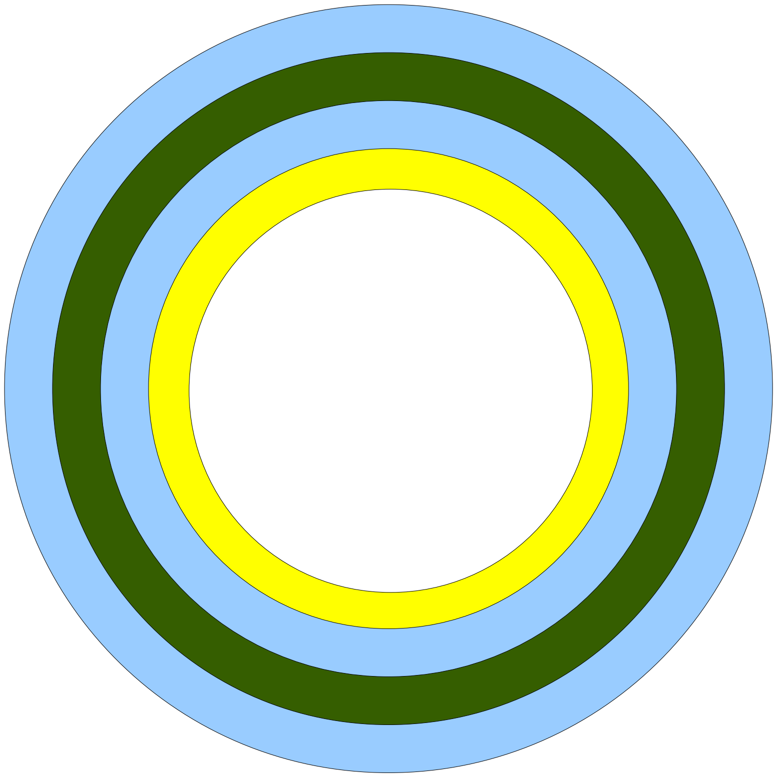

```xml
  <parameter name="SiTrackMCPRelCollection" type="string" lcioOutType="LCRelation">SiTracksMCP </parameter>
</processor>


<!--Performs full tracking in LDC detector-->
<processor name="MyFullLDCTracking" type="FullLDCTracking">
  <!--FTD Hit Collection Name-->
  <parameter name="FTDHitCollection" type="string" lcioInType="TrackerHit">FTDTrackerHits </parameter>
  <!--VTX Hit Collection Name-->
  <parameter name="VTXHitCollection" type="string" lcioInType="TrackerHit">VTXTrackerHits </parameter>
  <!--SIT Hit Collection Name-->
  <parameter name="SITHitCollection" type="string" lcioInType="TrackerHit">SITTrackerHits </parameter>
  <!--TPC Hit Collection Name-->
  <parameter name="TPCHitCollection" type="string" lcioInType="TrackerHit">TPCTrackerHits </parameter>
  <!--TPC Track Collection-->
  <parameter name="TPCTracks" type="string" lcioInType="Track">TPCTracks </parameter>
  <!--Si Track Collection-->
  <parameter name="SiTracks" type="string" lcioInType="Track">SiTracks </parameter>
  <!--TPC Track to MCP Relation Collection Name-->
  <parameter name="TPCTracksMCPRelColl" type="string" lcioInType="LCRelation">TPCTracksMCP </parameter>
  <!--Si Track to Collection-->
  <parameter name="SiTracksMCPRelColl" type="string" lcioInType="LCRelation">SiTracksMCP </parameter>
  <!--Cut on Opening Angle for forced merging of Si and TPC segments-->
  <parameter name="AngleCutForForcedMerging" type="float">0.05 </parameter>
  <!--Cut on Opening Angle for merging Si and TPC segments-->
  <parameter name="AngleCutForMerging" type="float">0.1 </parameter>
```


KLM
CsI
Barrel PID
CDC
Endcap PID
PXD, SVD
3.0m
2.5m
2.0m
1.5m
1.0m
0.5m
0.0m


KLM
CsI
Barrel PID
CDC
Endcap PID
CVD
3.0m
2.5m
2.0m
1.5m
1.0m
0.5m
0.0m

Beampipe

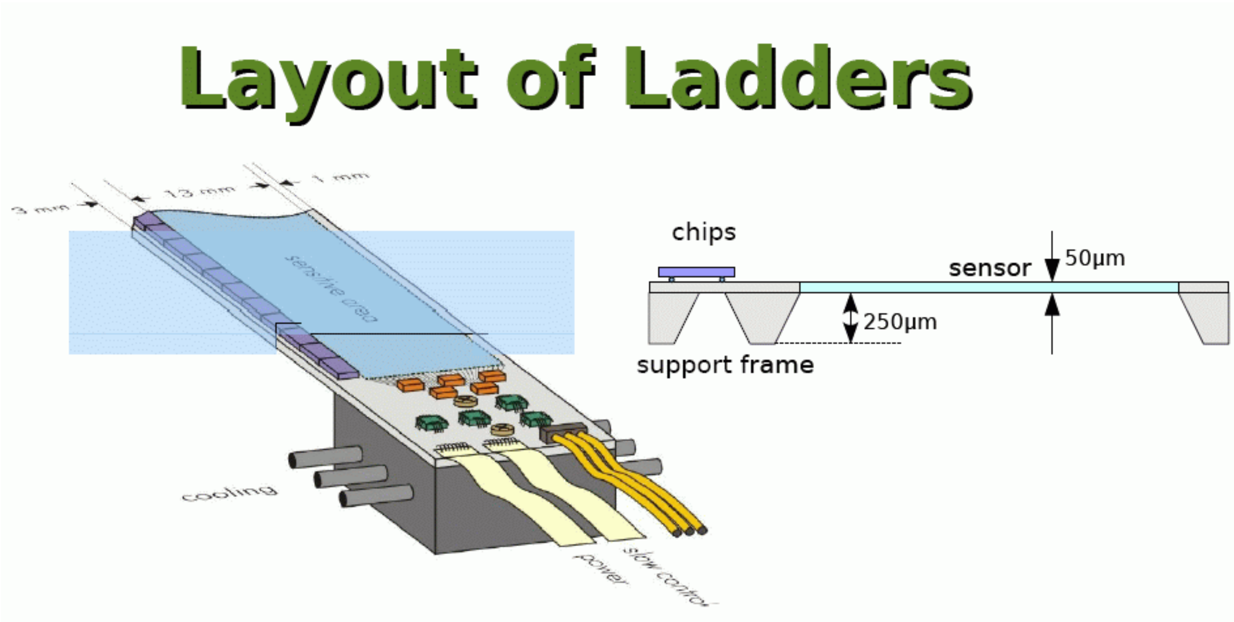


Beryllium walls

Cooling liquid (paraffin)

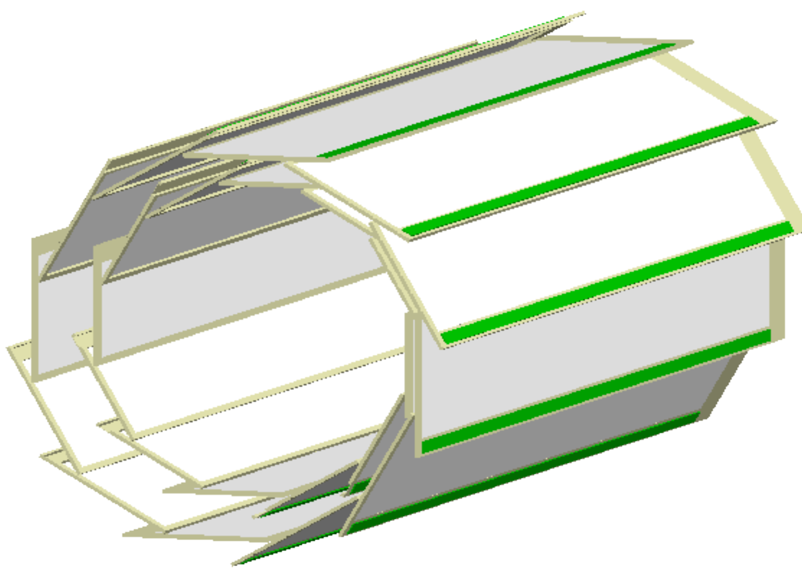
Gold coating (shielding against SR)

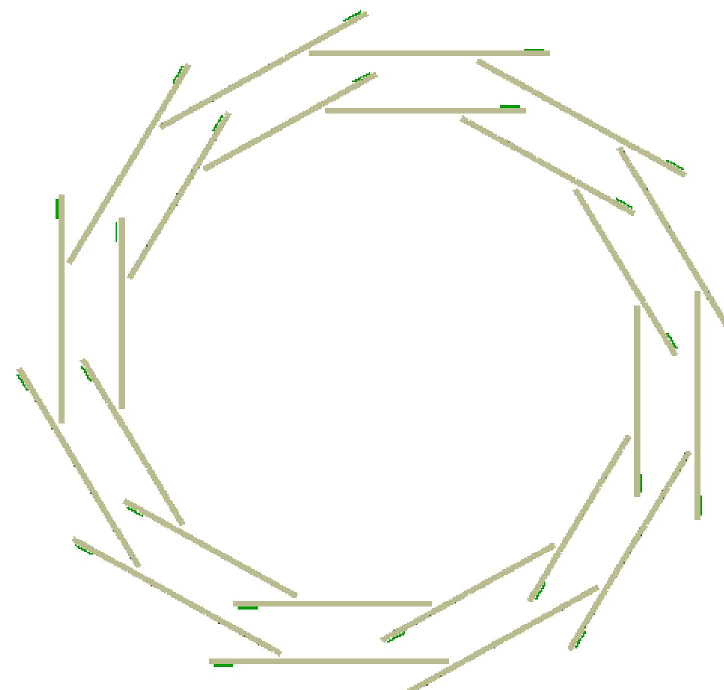
- Beam pipe adjacent to PXD is represented by cylinder
- Outer radius = 1.0 cm (1.5 cm in Belle)
- 4 layers
  - Be outer wall : 0.35 mm
  - Cooling liquid : 0.5 mm
  - Be inner wall : 0.6 mm
  - Gold coating : 10 μm
- Material budget : 0.66% X_0



⇐ first two layers of PXD (DEPFET)

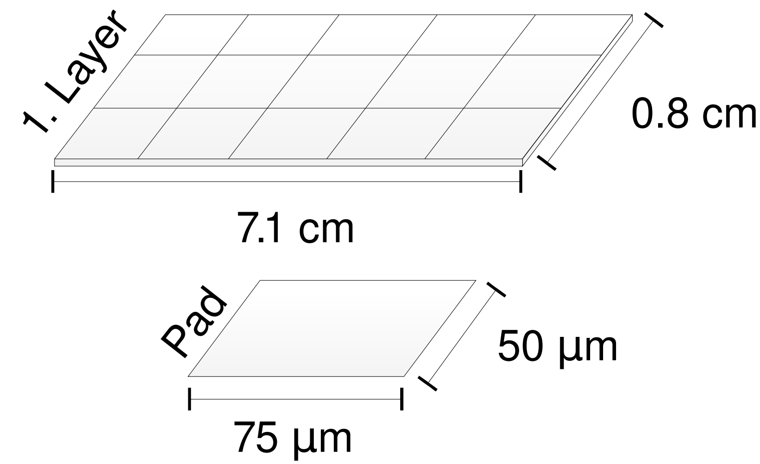
Detailed implementation
⇒ frames, electronic chips





Mokka implementation of SuperBelle Vertex Detector

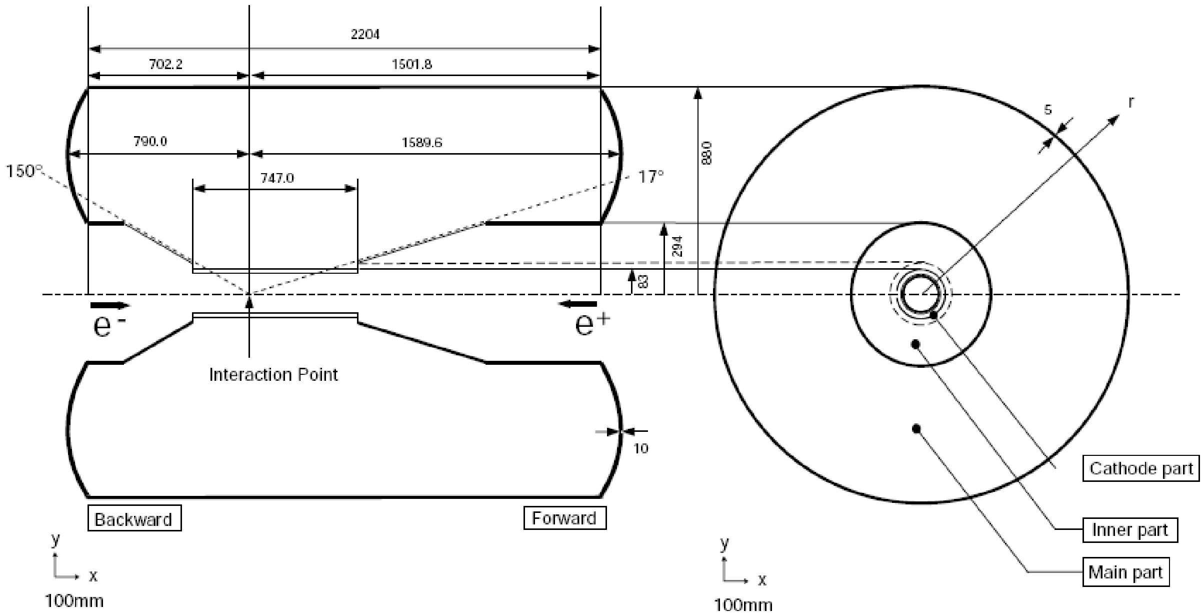
- New **database** (subdectector) in local MySQL was added.
- New **geometry driver** was written



Used values:

	#	r *(cm)*	**sensor** *(cmxcm)*	**# sensor** *in z*	**#ladders** *(around phi)*	**thickness** *(μm)*
Pixel	1	1.3	7.1x0.8	1	12	50
	2	1.6	8.4x1.0	1	12	50
Strips	3	4.5	8.0x2.8	3	12	300
	4	7.0	7.6x4.0	5	12	300
	5	10.0	9.0x2.8	5	24	300
	6	13.8	7.6x4.0	6	24	300

Simplified implementation of the four outer layers
⇒ each layer with strip readout represented by single Si plane
Strip size : $d(r\text{-}\phi) = 50\ \mu m$, $d(z) = 150\ \mu m$



Gas mixture:
He/C_2H_6 (50:50)
Al cage 5 mm thick

Parameters envisaged for superBelle

• Radius of inner boundary	150 mm
• Radius of outer boundary	1150 mm
• Radius of the innermost sense wire	172 mm
• Radius of the outermost sense wire	1120 mm
• Number of layers	58
• Effective radius of dE/dx measurements	978 mm

Simplified digitization procedure:

- G4 tracking

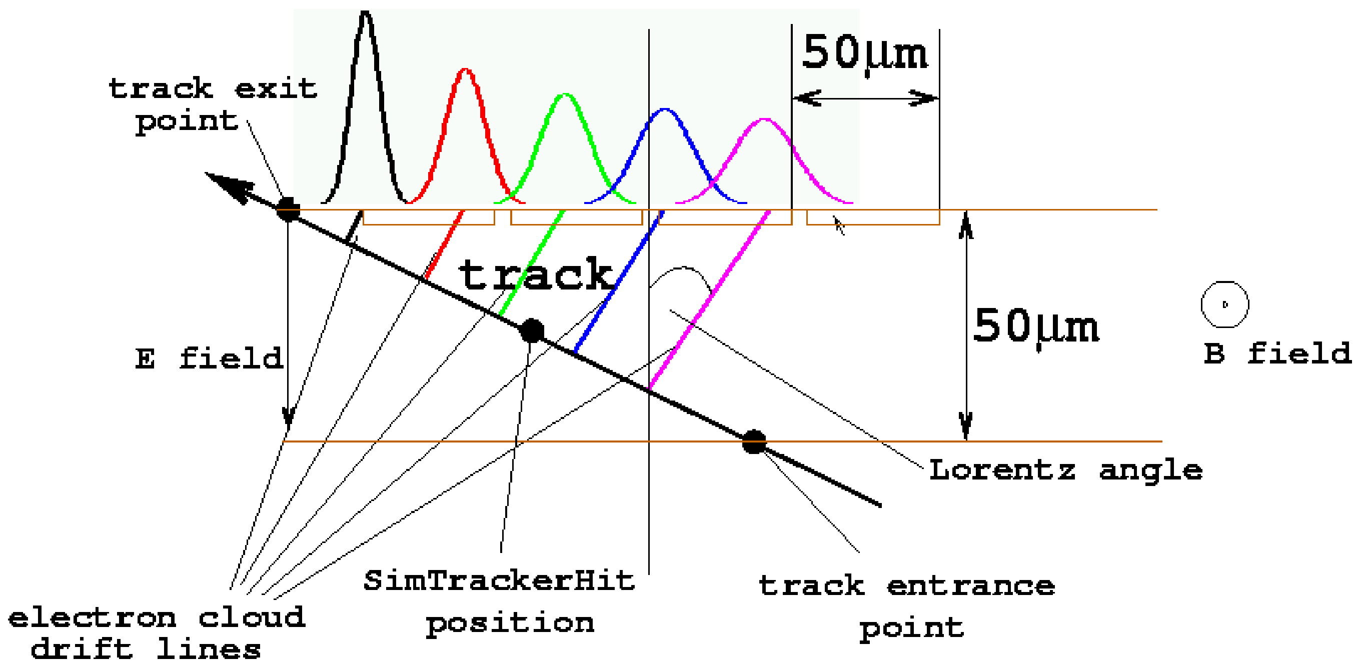
⇒ track intersection with measurement cylindrical layer

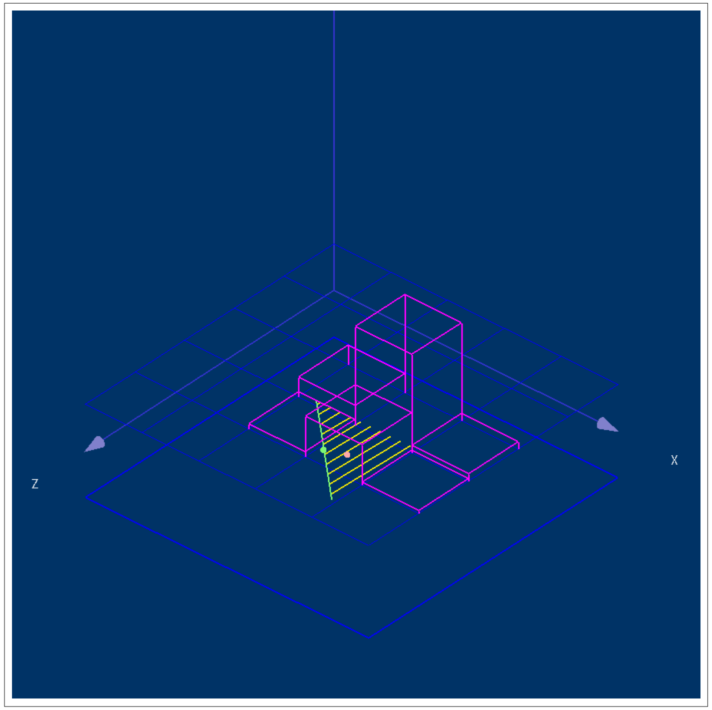
⇒ 3D point is smeared according to average spatial point resolutions as obtained in Belle

$\sigma(r\text{-}\phi)=130\mu m$

$\sigma(z)=800\mu m$

Simulation of DEPFET sensor response (**Digitization**)





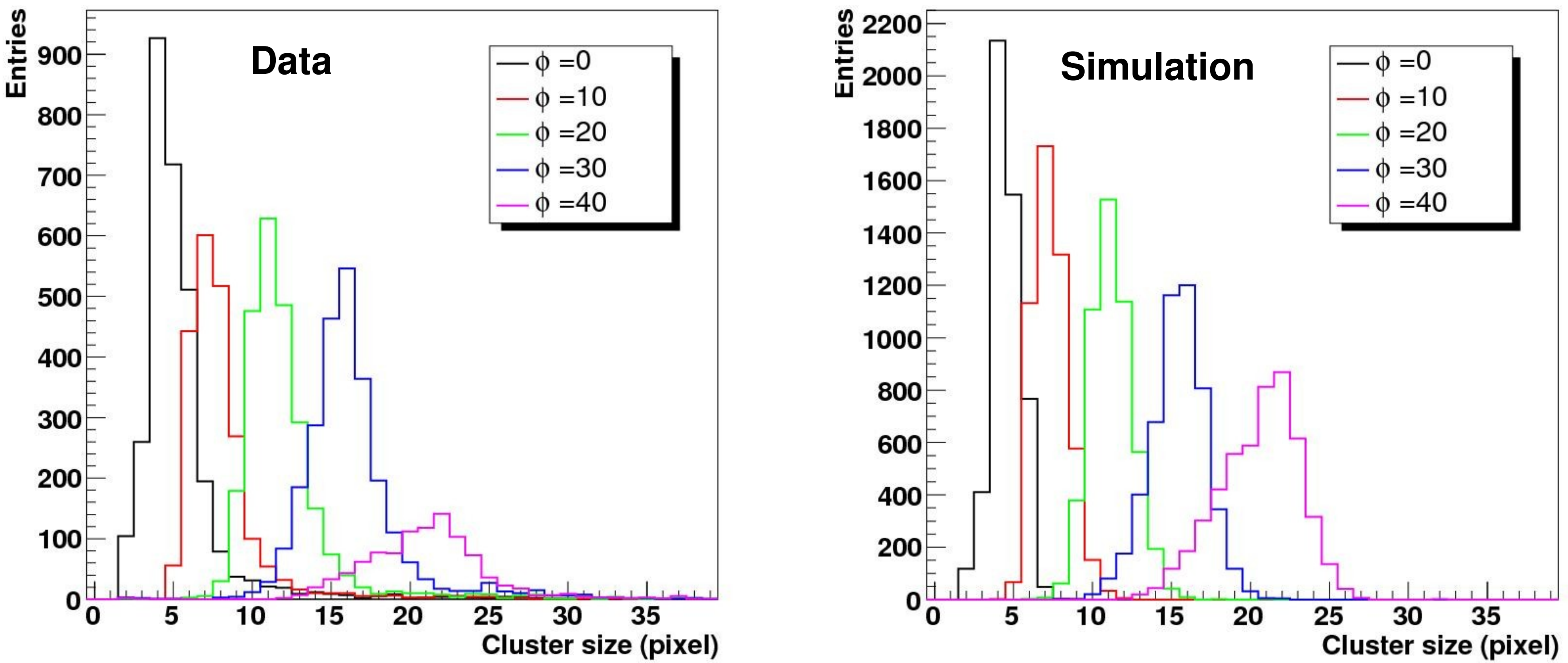
Effects taken into account

- Energy loss fluctuations along track path
- Electron cloud diffusion: Charge transport and sharing between neighboring pixels
- Lorentz shift in magnetic field
- Electronic noise (100/1200 e^- in pixels/strips)
- Ghost hits in case of strips

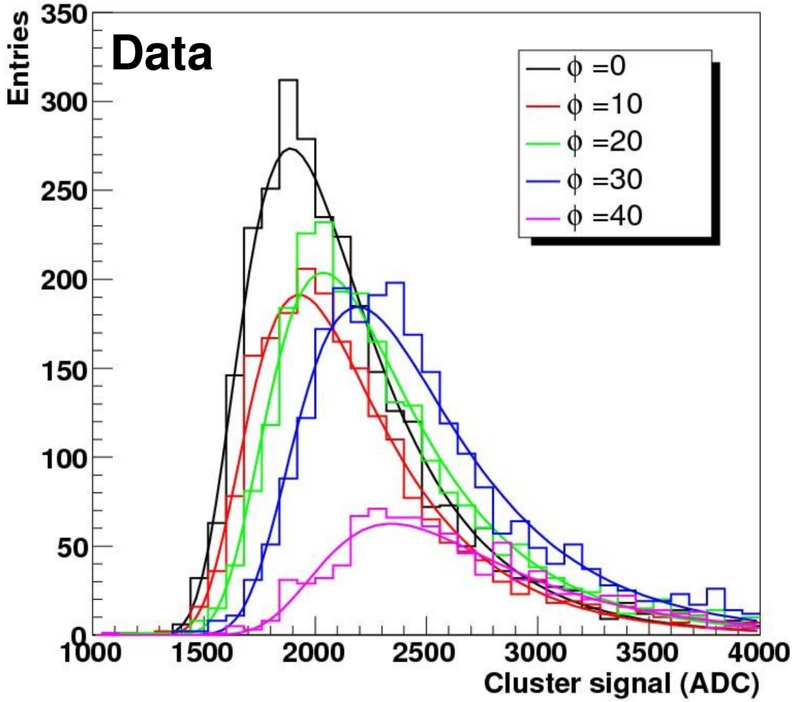
Comparison with data from DESY 2005 test beam (6 GeV electrons)

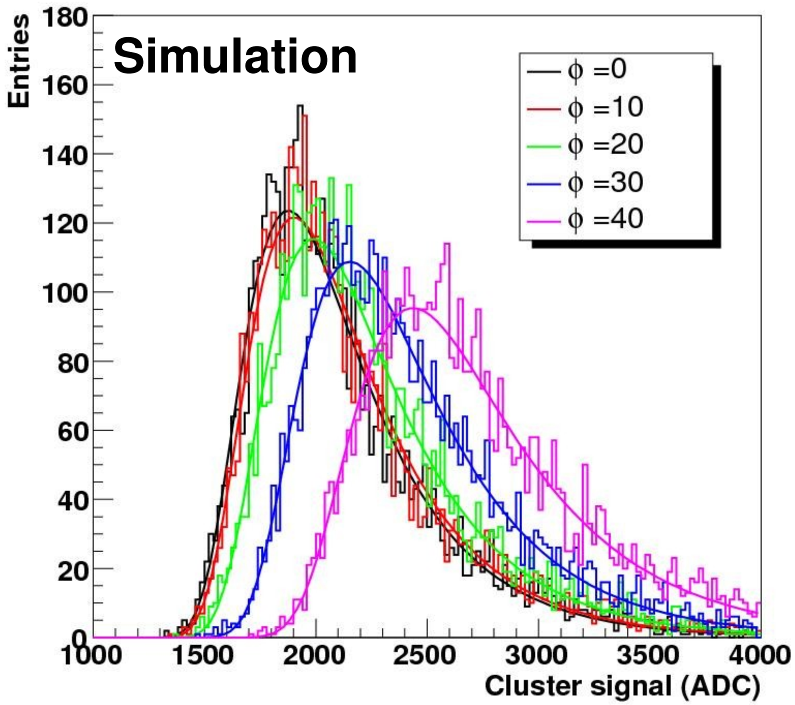
- DEPFET: thickness 450 µm, pixel size 36x22 µm^2
- no B field
- electronic noise set to 300 e$^-$
- Track incident angles from 0 to 40 degrees

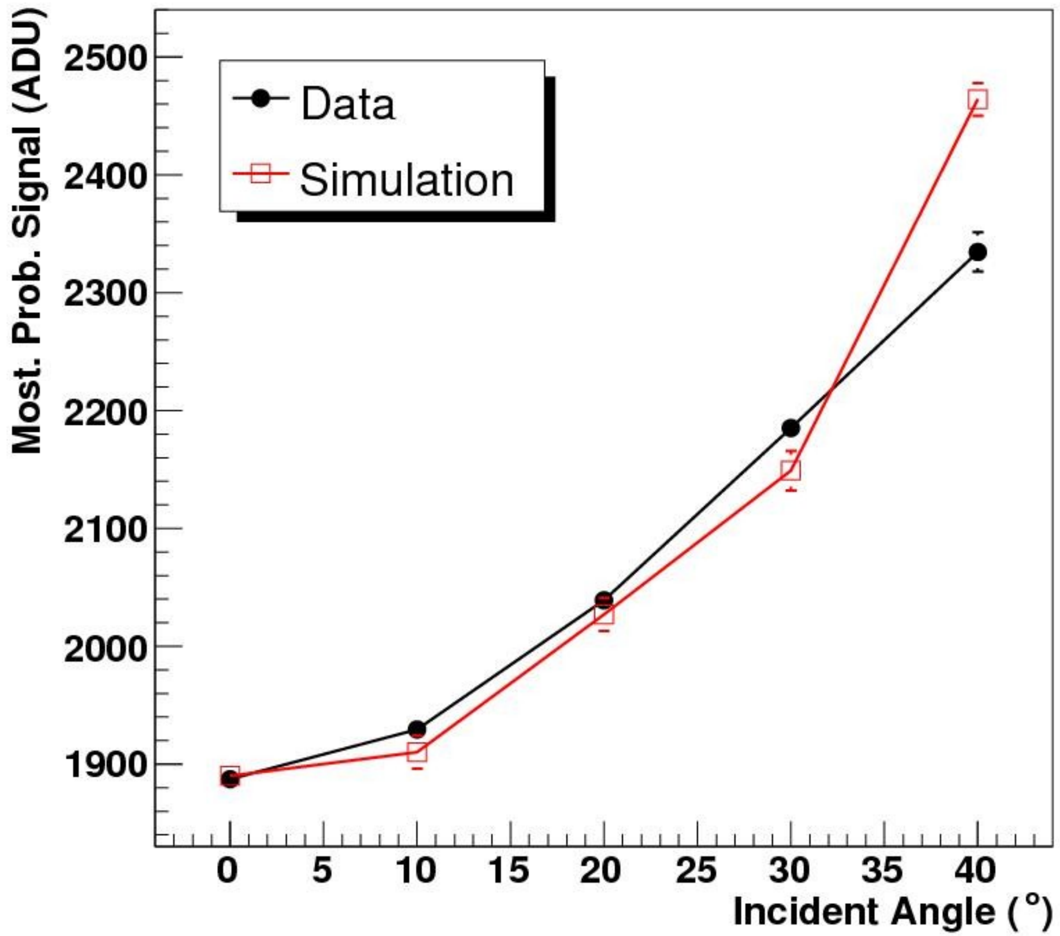
Cluster Size



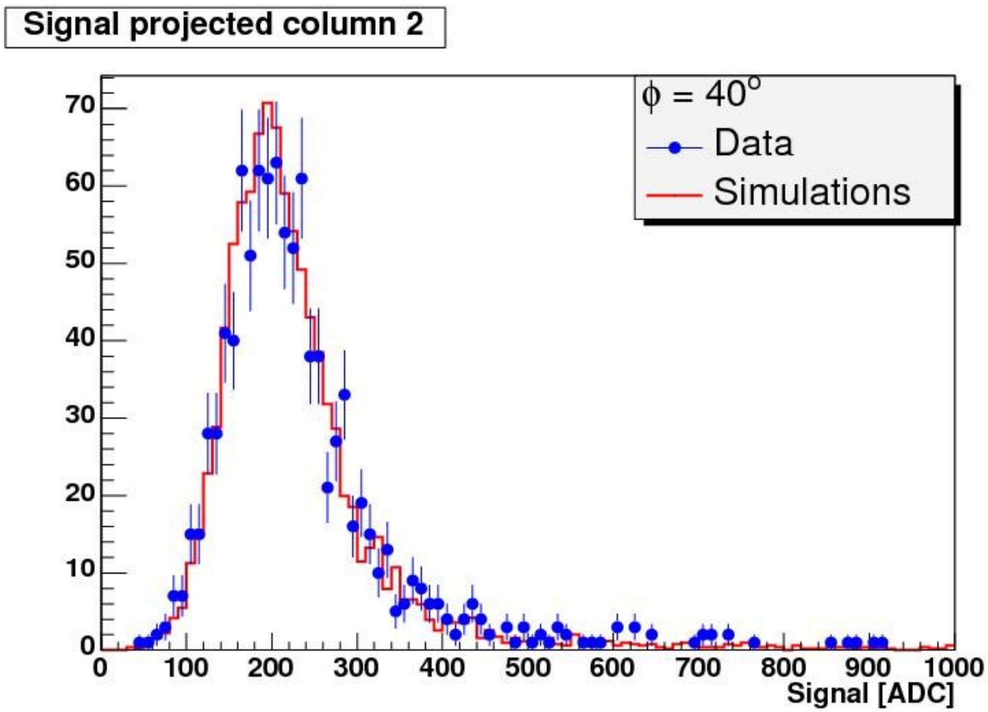
Validation with Test Beam data – Cluster Signal

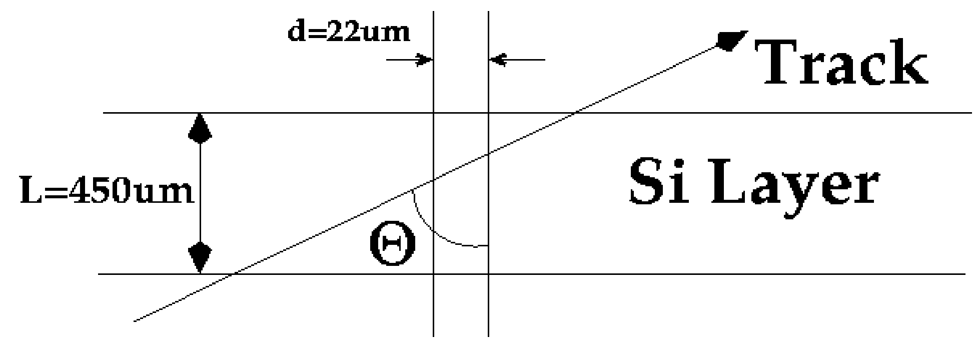




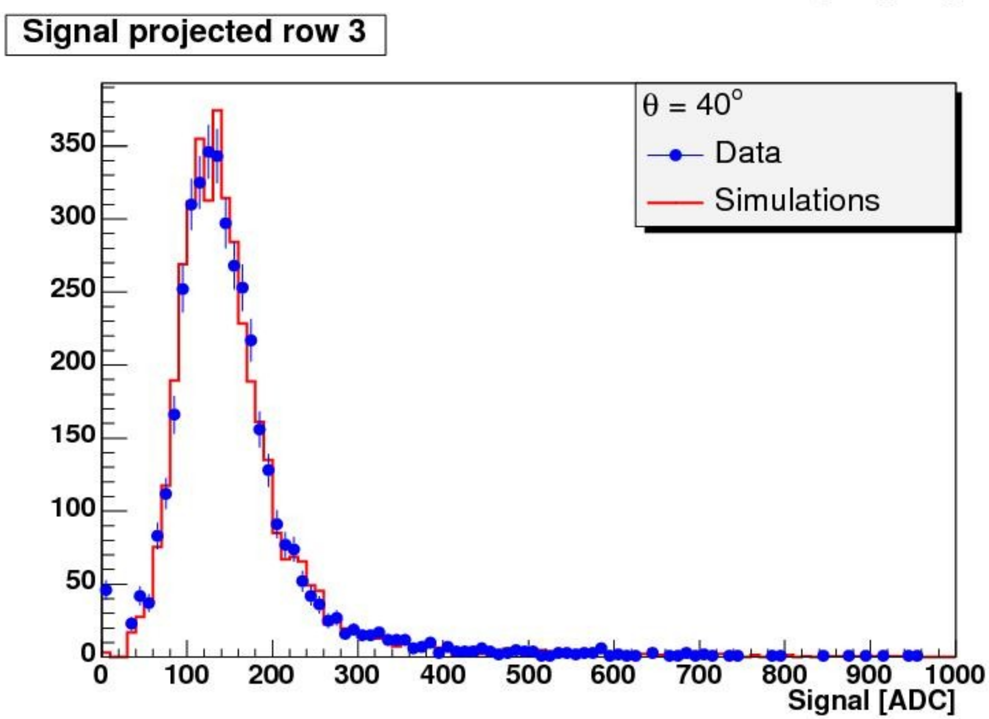


Good agreement of test beam data
and simulation

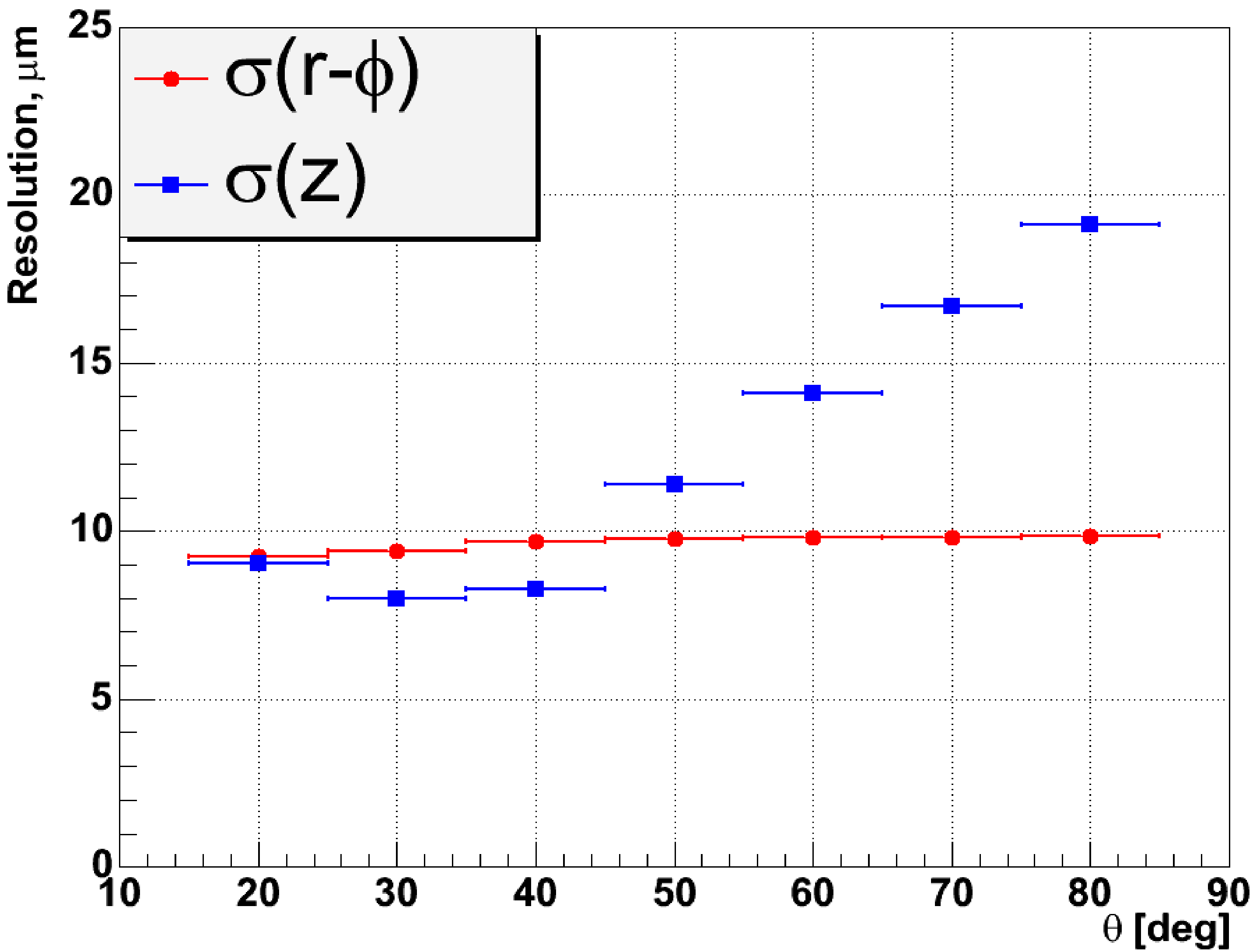




d	pixel size
Θ	incident angle

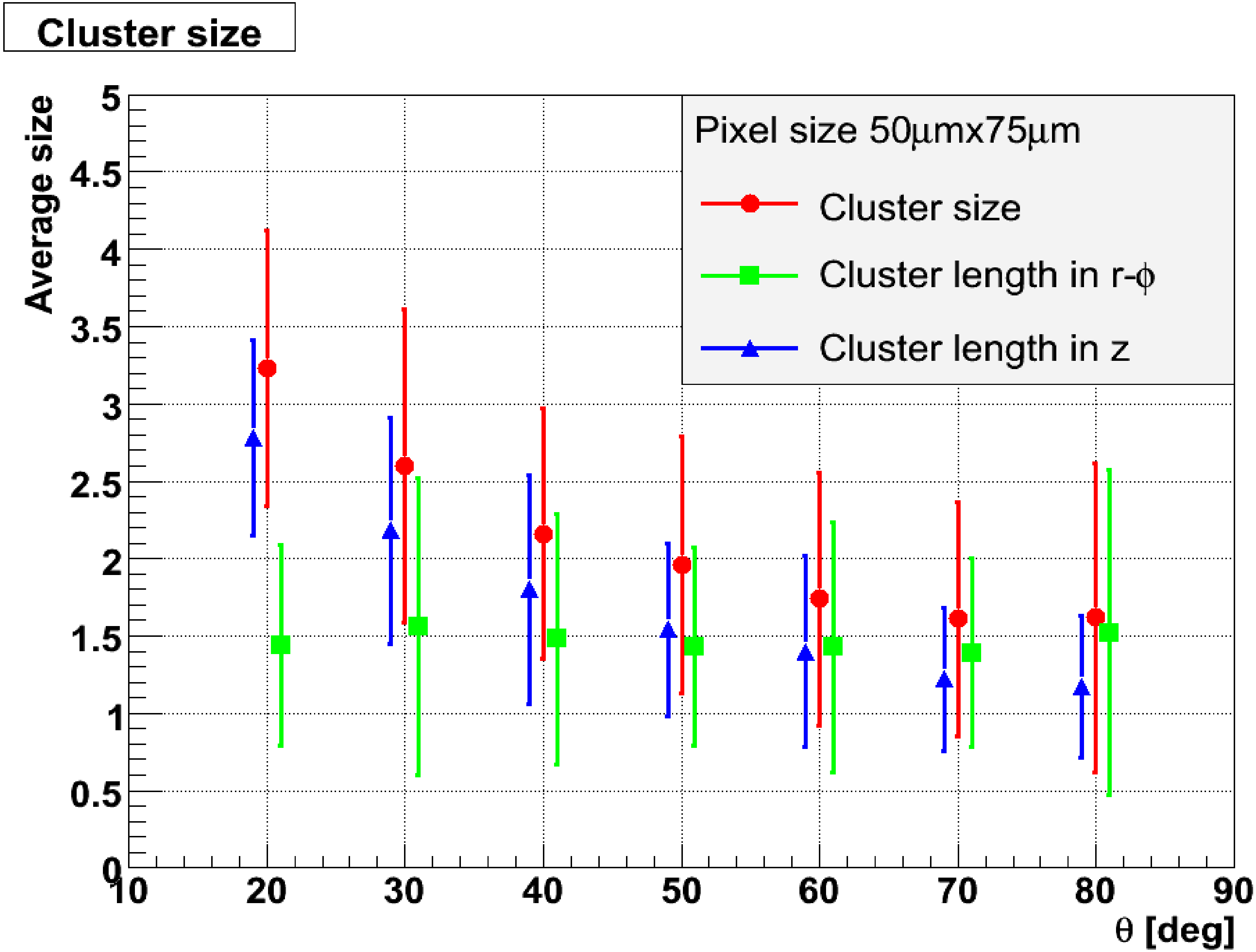


Simulation yields correct results also for thin sensors.



Hit position is calculated as a center of gravity of fired pixels in a given cluster.

Cut on pixel amplitude applied : $A > 2 \cdot \sigma_{noise}$


Cluster size
Average size
Pixel size 50µmx75µm
Cluster size
Cluster length in r-ϕ
Cluster length in z
θ [deg]
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
10
20
30
40
50
60
70
80
90

Sources of backgrounds at Belle/superBelle

1. Synchrotron radiation

2. Particle background (beam-gas interactions, intra-bunch scattering)

- Background rate estimate for Belle : $r_{BG} \approx 23500\ s^{-1} mm^{-2}$ (sBN/0007)
- Factor of 15 increase in background at superBelle.
- This does not include possible increase due to smaller distance to IP if beam pipe radius is reduced from 15 to 10 mm
- Safety margin : factor 30 is assumed $\Rightarrow r_{BG} \approx 700000\ s^{-1} mm^{-2}$

- Integration time for DEPFET sensors (ILC configuration) : **$50\mu s$**

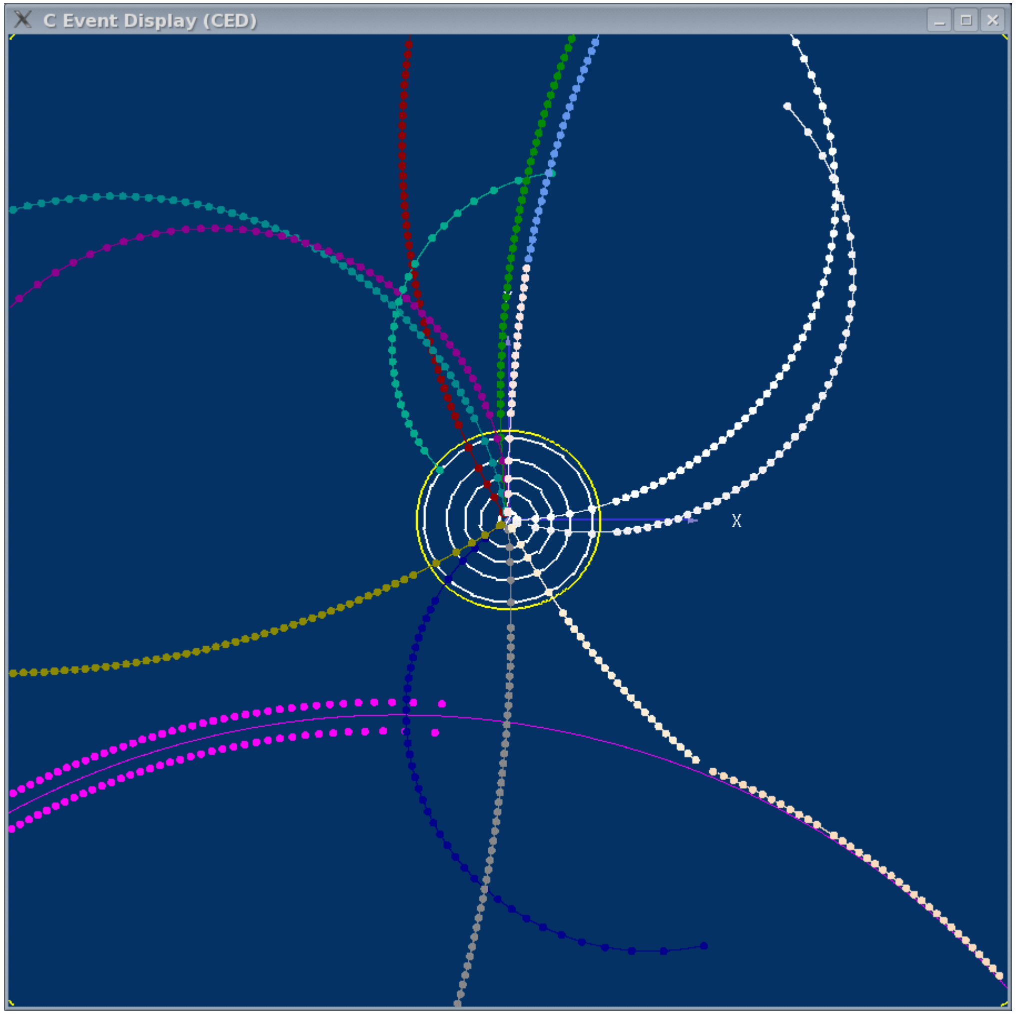
  $\tau \propto L_{ladder} / L_{pixel} \Rightarrow$ (SuperBelle configuration): **$12\mu s$**

$$Occupancy = r_{BG} \cdot \langle Cluster\_size \rangle \cdot Pixel\_area \cdot \tau \approx 6.5\%$$

- **Uniform distribution of background hits over ladder is assumed**
- **r_{BG} (second layer) = $0.5 \cdot r_{BG}$ (first layer), but readout twice slower**
- **backgrounds in the strip layers of SVD neglected**

Pattern recognition & track reconstruction procedure

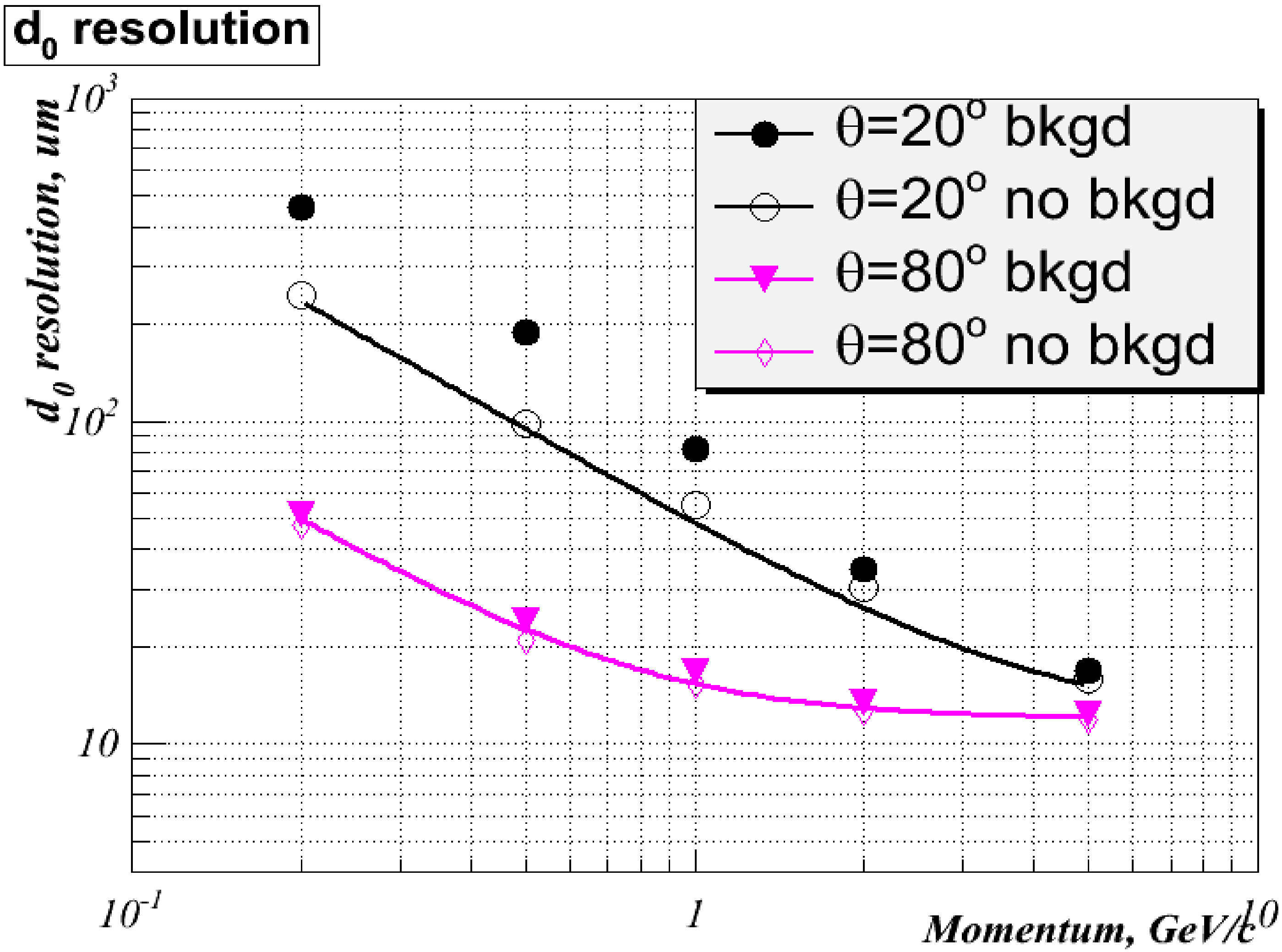
- Pattern recognition in CDC : Inward search for continuous hit patterns compatible with helix hypothesis
  - ILC code reused
- Fit of CDC tracks. Kalman filter
- Extrapolation of CDC tracks into vertex detector, picking up hits. Procedure starts from the outermost layer and continues inward. Track refit every time new hit is added to track
- Final track fit , quality cut on χ^2



$e^+e^- \rightarrow B^+B^-$ event

Tracking system of SuperBelle

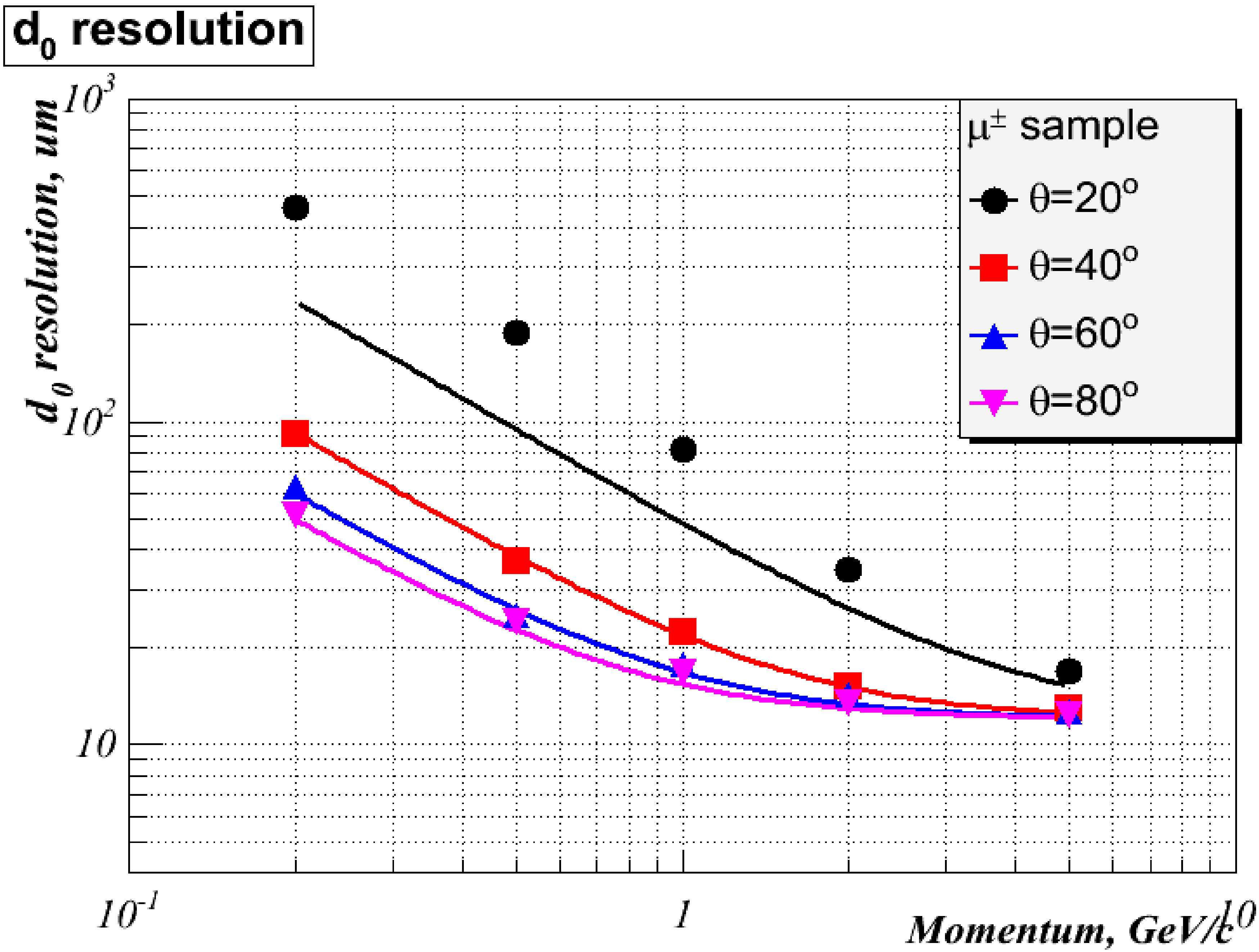
Performance studies with single particles (muons)

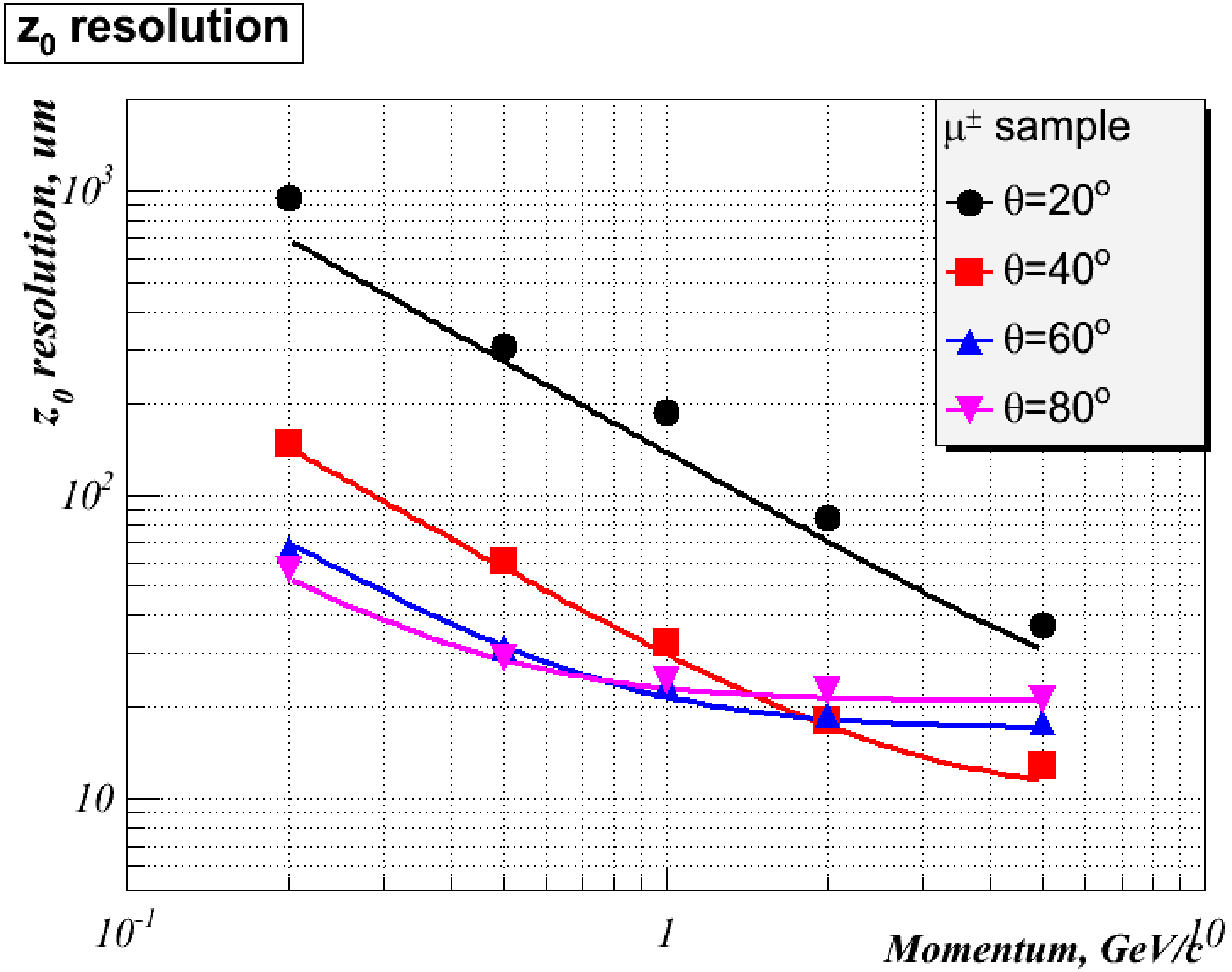


$$\text{Curves}: \sigma(d_0) = a \oplus b / p \cdot \sin^{3/2}\theta$$

$$a = 12\mu m$$

$$b = 13.6 MeV \cdot r_{beampipe} \cdot \sqrt{x/X_0}\,[1 + 0.038 \ln(x/X_0)] = 9.4\ \mu m\ GeV/c$$

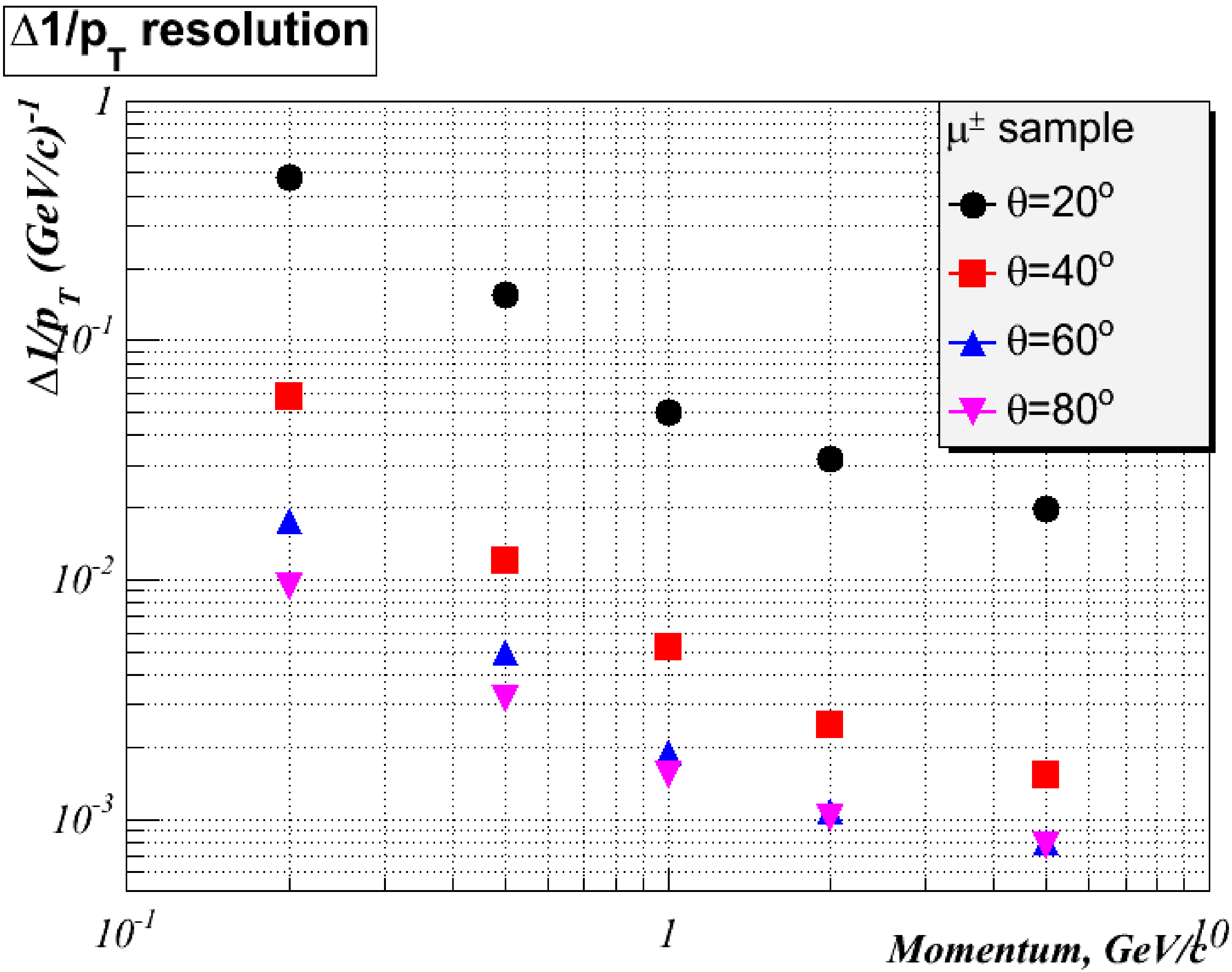

d$_0$ resolution
10^3
10^2
10
d_0 resolution, um
10^{-1}
1
10
Momentum, GeV/c
μ± sample
θ=20°
θ=40°
θ=60°
θ=80°

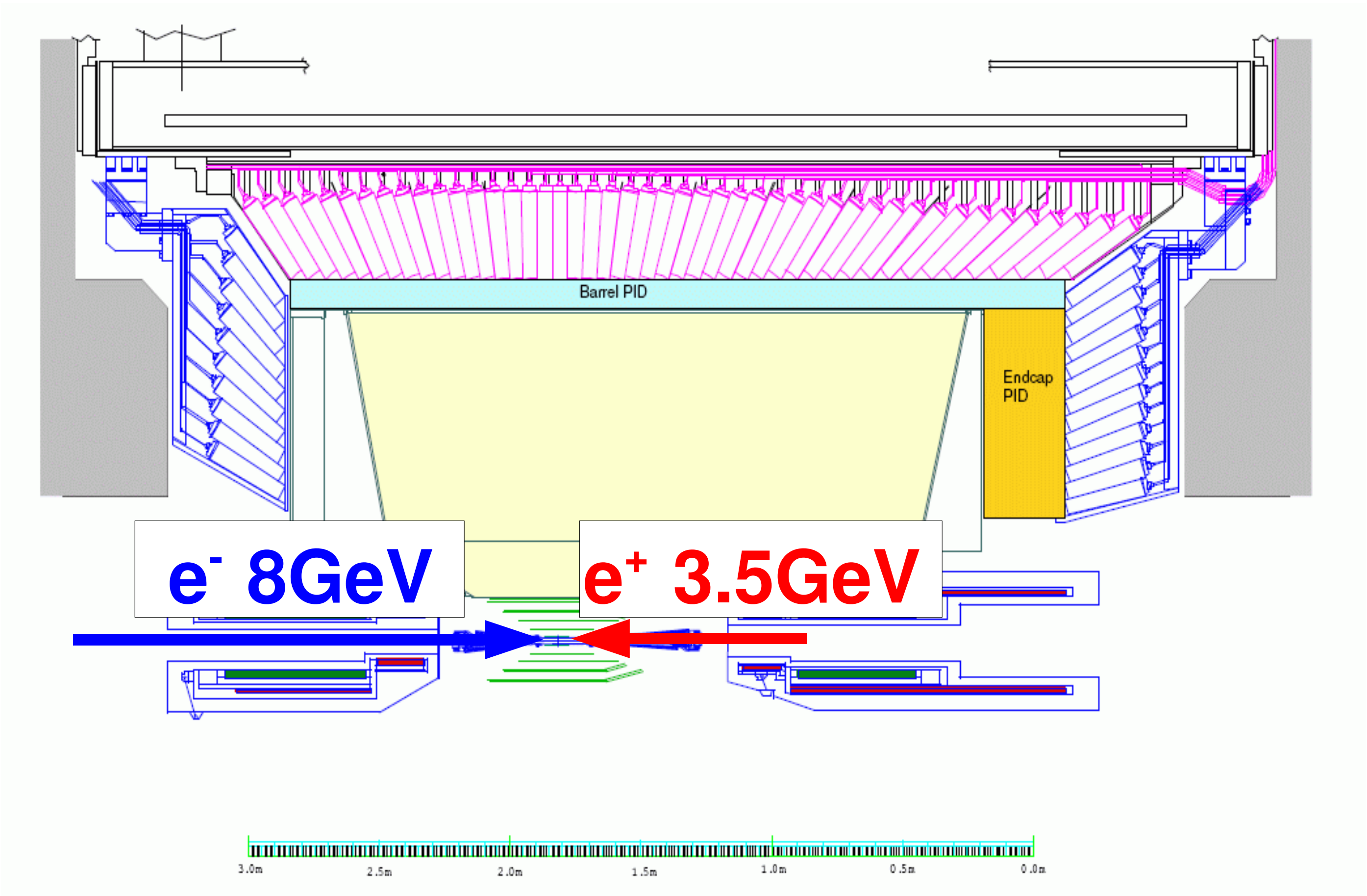


Curves : $\sigma(z_0) = a(\theta) \oplus b / p \cdot \sin^{5/2}\theta$

$a(20^\circ, 40^\circ, 60^\circ, 80^\circ)$ = 14, 10, 17, 21 μm

$a(\theta)$ dependence is explained by strong dependence of spatial point resolution in z on θ


Δ1/p_T resolution
Δ1/p_T (GeV/c)^-1
1
10^-1
10^-2
10^-3
10^-1
1
10
Momentum, GeV/c
μ± sample
θ=20°
θ=40°
θ=60°
θ=80°

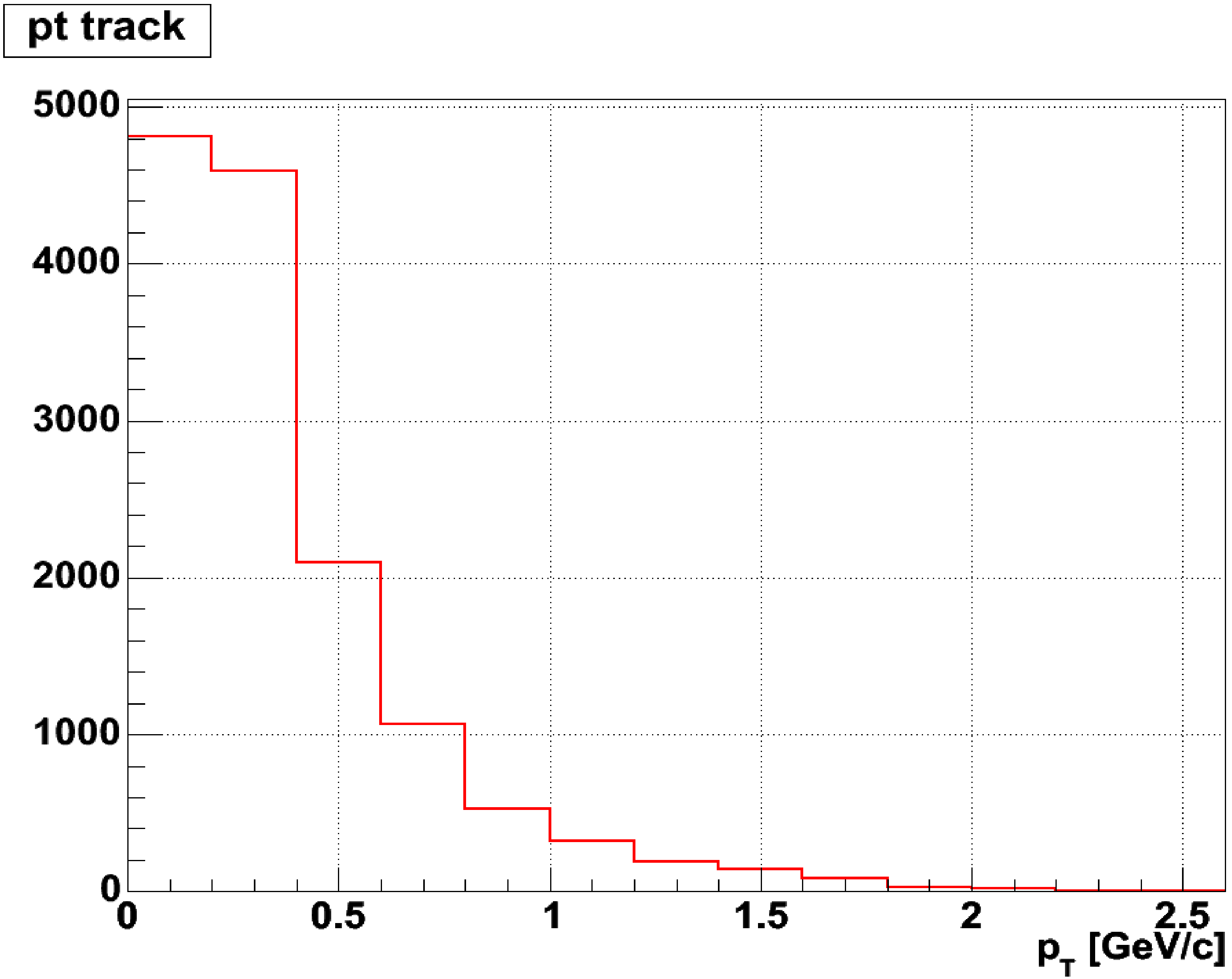


Signal sample : $e^+e^- \rightarrow B^0\overline{B}^0$ (B^+B^-)

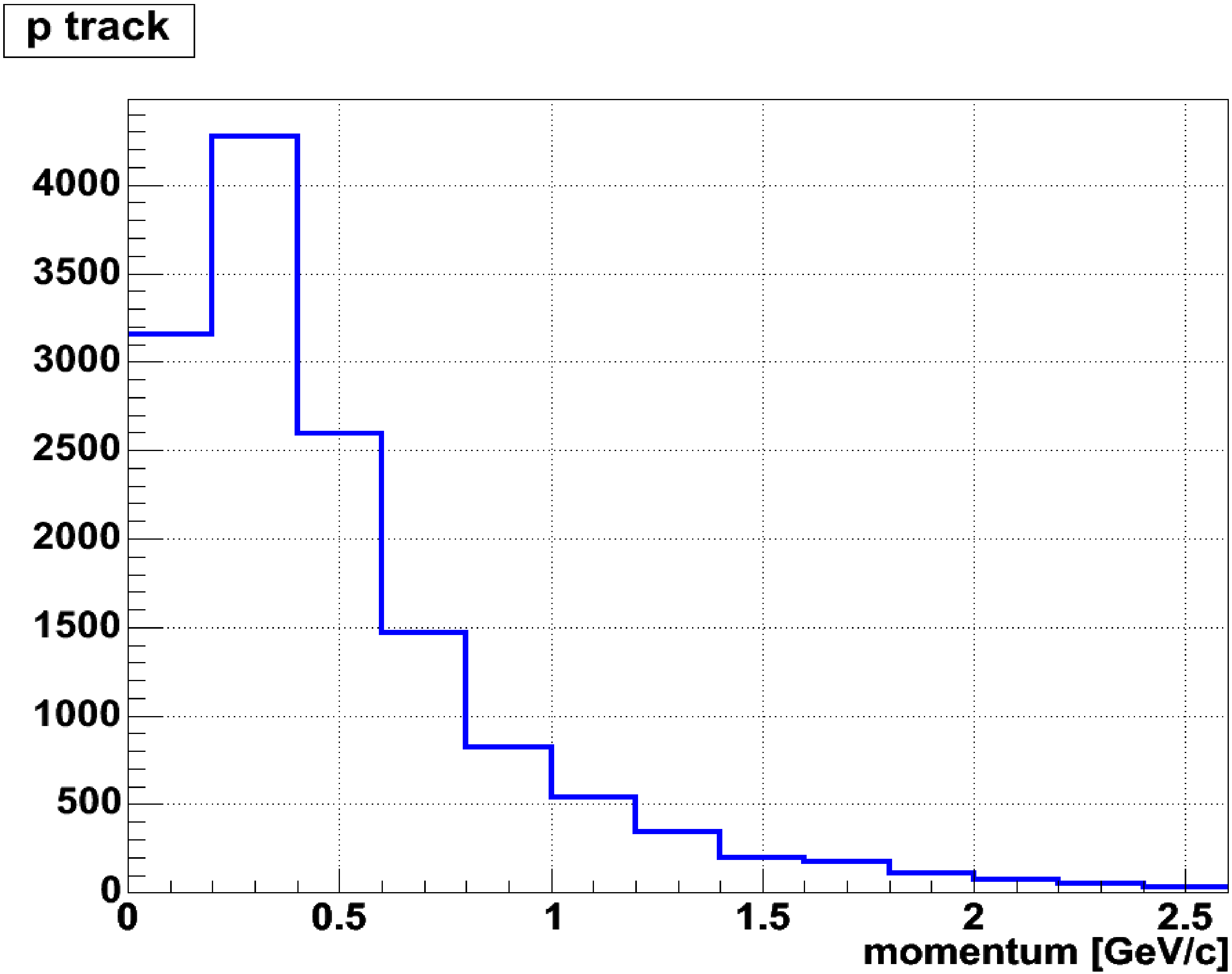
all decay modes of B's are on

Study performance in terms of track finding efficiency and fraction of tracks spoiled by bg hits

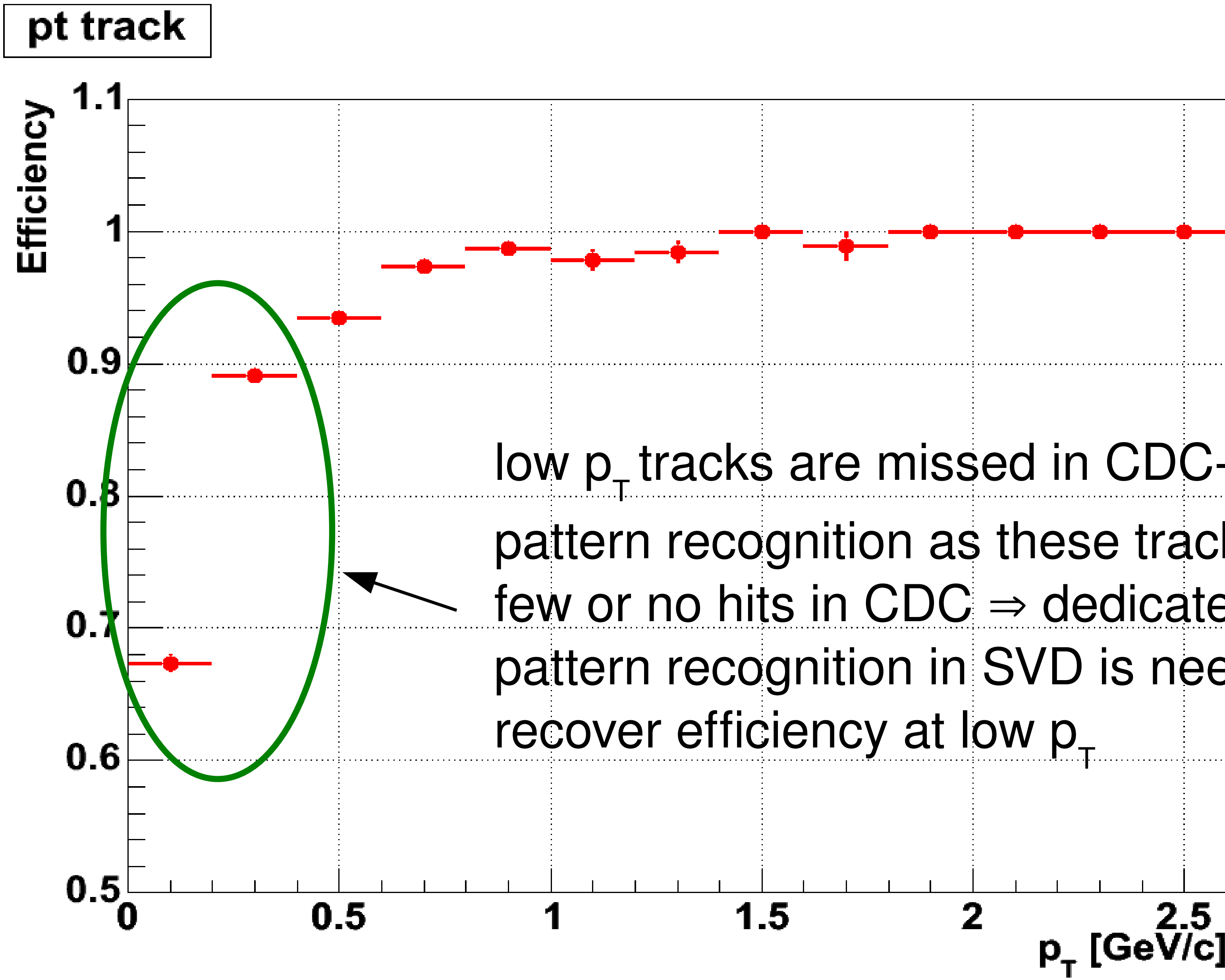
P_T spectrum of tracks in ee→BB sample



Momentum spectrum of tracks in ee→BB sample

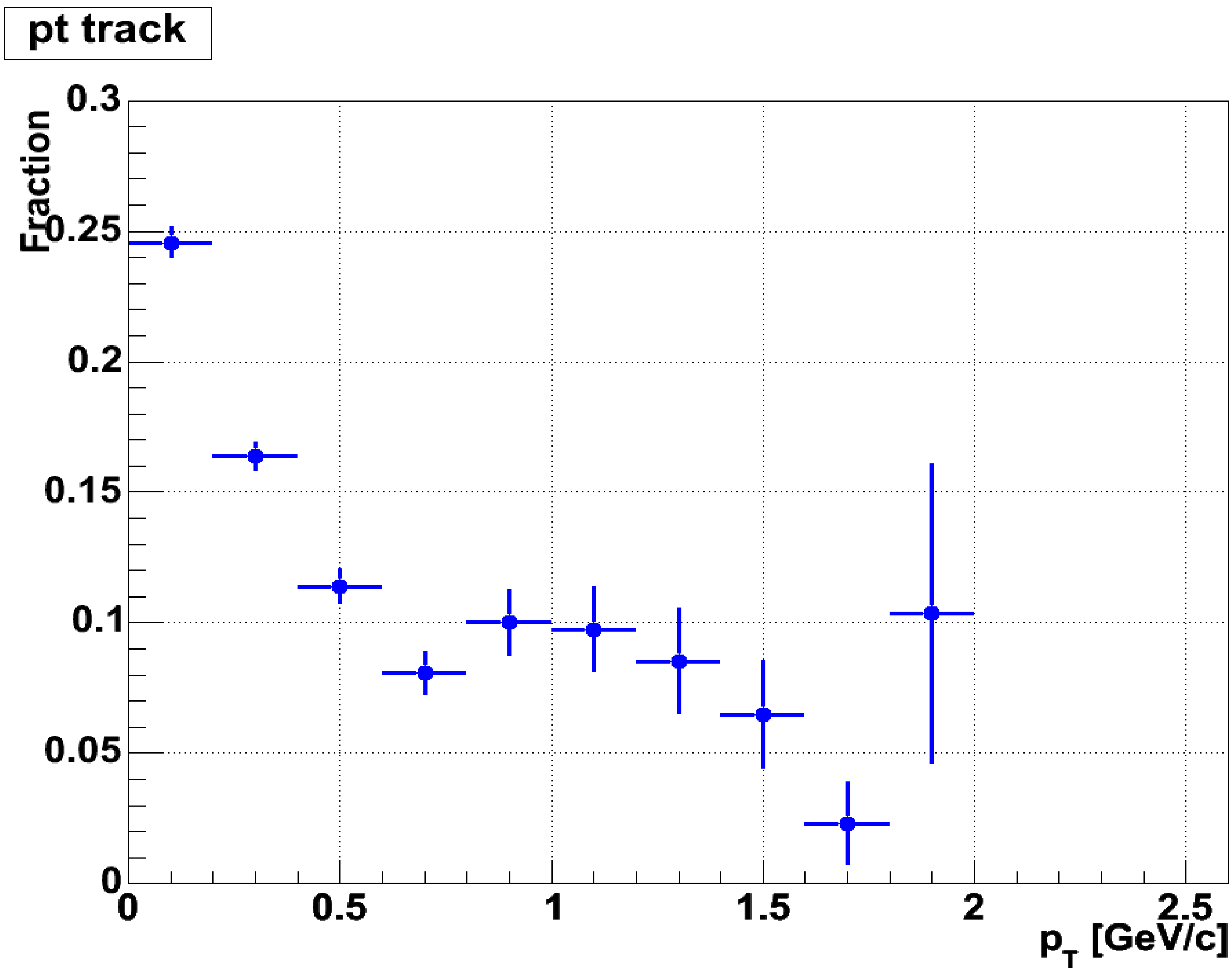


Track finding efficiency

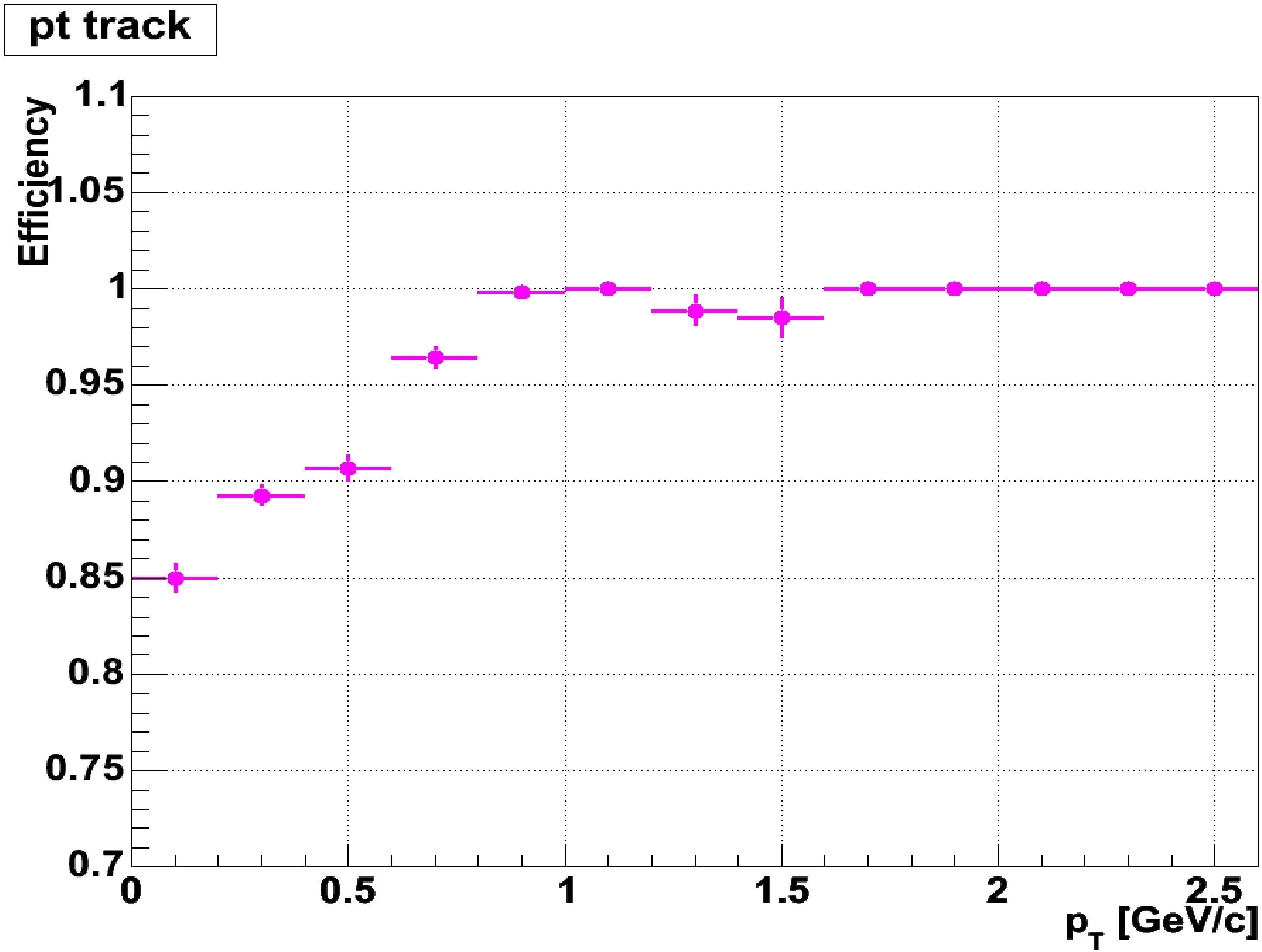


low p_T tracks are missed in CDC-driven pattern recognition as these tracks leave few or no hits in CDC ⇒ dedicated pattern recognition in SVD is needed to recover efficiency at low p_T

Fraction of tracks containing background hits



Silicon hits - CDC tracks association efficiency



Installation of Belle Software at MPI

- With joining Belle collaboration MPI group got access to Belle data
- Lack of experience and knowledge prevents us from immediate participation in physics studies ⇒ need “learning tools” to gain experience and knowledge about Belle software.
- With the help Vienna group (C. Schwanda), Belle software has been installed at MPI. This includes
  - libraries providing interface to PANTHER banks (Belle data model)
  - Detector simulation and event reconstruction tools
  - Belle data (limited statistics)
  - Tutorial examples, documentation
- Software is currently used by members of MPI group to understand and learn Belle analysis framework

Summary

- A software framework is being established to pursue performance and optimization studies of SuperBelle detector
- Initial simulation studies showed that DEPFET based vertex detector planned for SuperBelle is capable of operation under background conditions expected at SuperBelle, yielding good performance
- Belle analysis software has been installed at MPI with the help of Vienna group. Software serves as a "learning tool" for the members of newly established Belle group at MPI