Status of the LMD-Software (Luminosity-Monitoring-Det.)

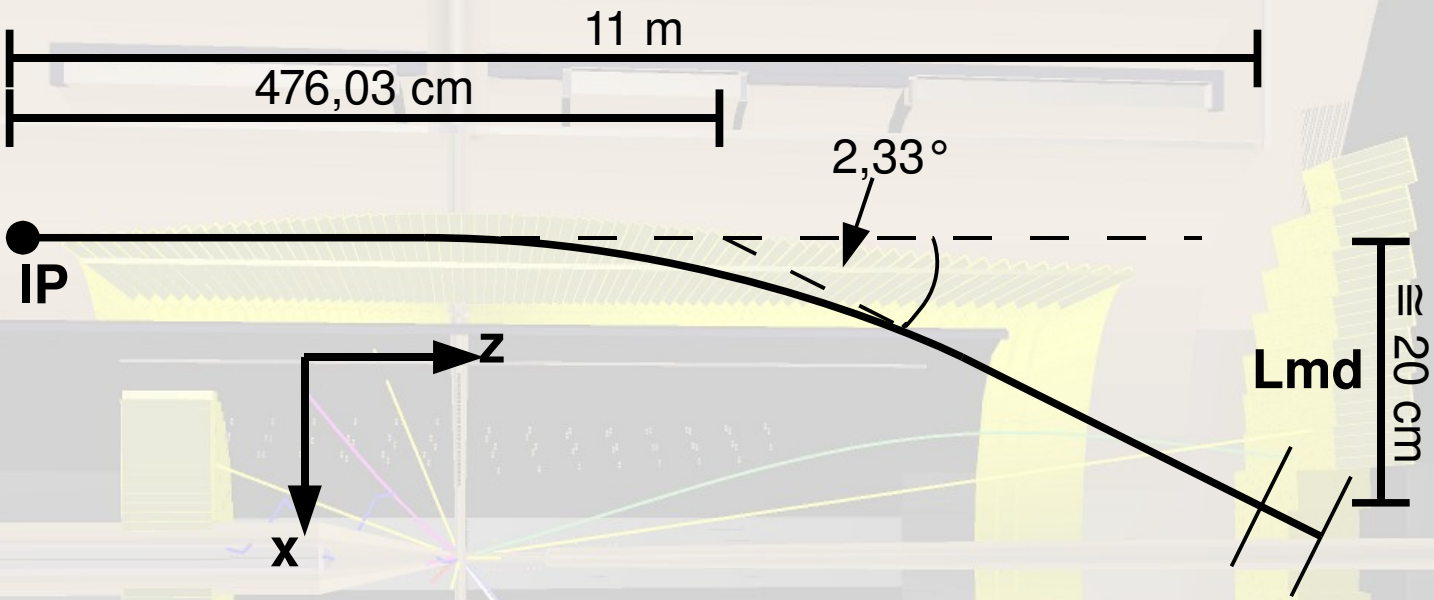
Mathias Michel

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

PANDA Collaboration Meeting

Aug 2010

The Luminosity Monitor



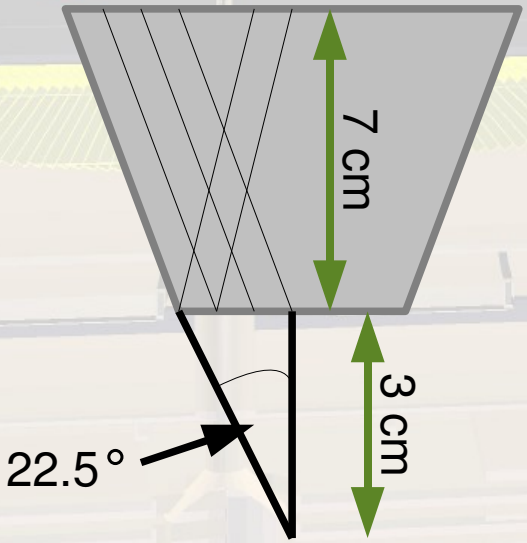
Example:

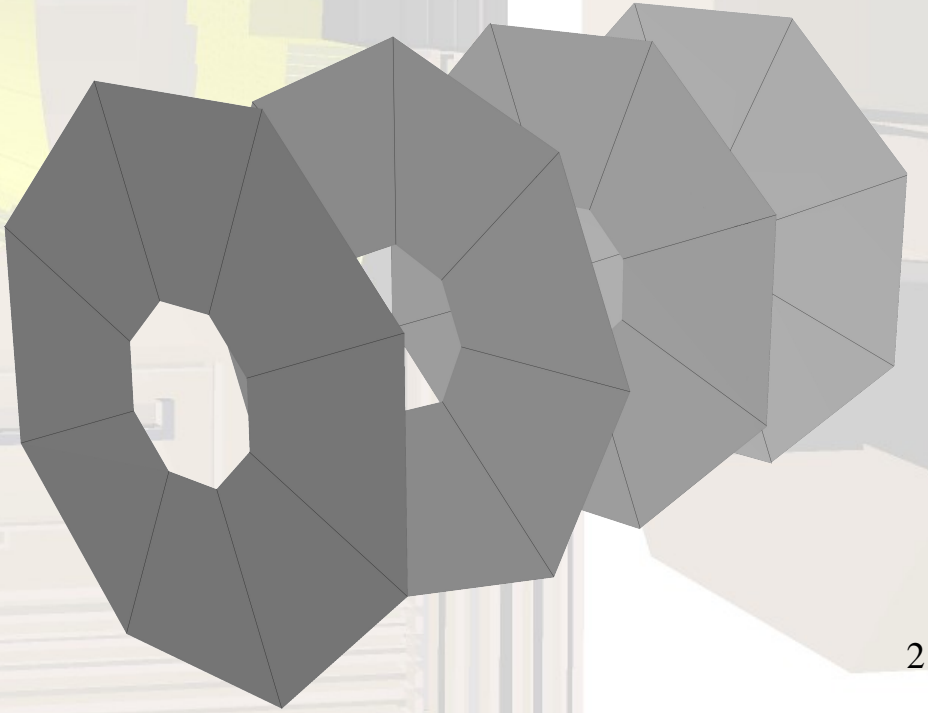
4 planes à 8 Si-trap. with 10 cm distance

thickness: 150 µm

stereo-angle 45°

pitch 50 µm





LMD-Software (/trunk/lmd)

Simulation
(same physics in every semiconductor-detector)

Track-Reconstruction
(Lumi outside magnetic fields: new algorithms!)

Hit-Reconstruction
(intrinsic: similar in every semicond.-det.)

1. PndLmdDetector

2. PndLmdStripHitProducer

3. PndLmdStripClusterTask

4. ..TrackFinderTask

5. ..LinFitTask

6. ..GeaneTask

⇒ full simulation and reconstruction chain available and working, macros and full-sim-script in /trunk/macro/lmd

SDS-based LMD

MVD
Micro-Vertex-Detector

generalized →

SDS
Semiconductor-Detector Software

derived ↓

LMD
Luminosity-Monitoring-Detector

⇒

Particle-Transportation
Digitization
Hit-Reconstruction

new: digitization with diffusion, developed by Tsito, not yet in svn

Track-Follower

Algorithm

1. sort hits by plane
2. create pseudo-tracks
3. extend to other planes (corridor)
   ➔ at least 3 hits: track-candidate found
4. 3D-Straight-Line fit of candidate

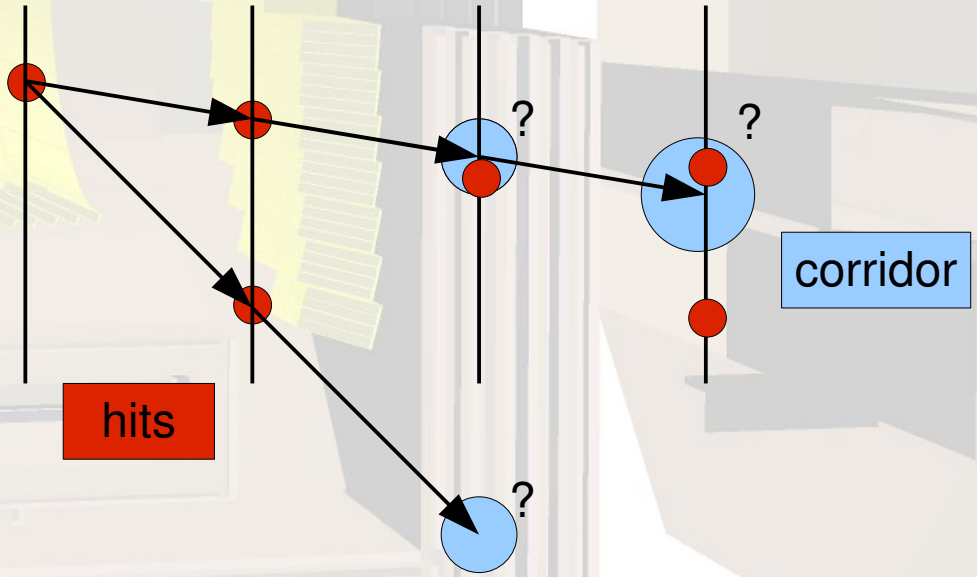
Characteristics

no restriction on number of hits

reconstruction of multiple tracks

corridor corresponds to scattering

run-time: $O(n^2)$



Hough-Transformation

Algorithm

1. transfer hits in (Θ,φ)-dualspace
2. heaps in dualspace:
   ➔ build Track-Candidates out of this hits
3. 3D-Straight-Line Fit of candidates

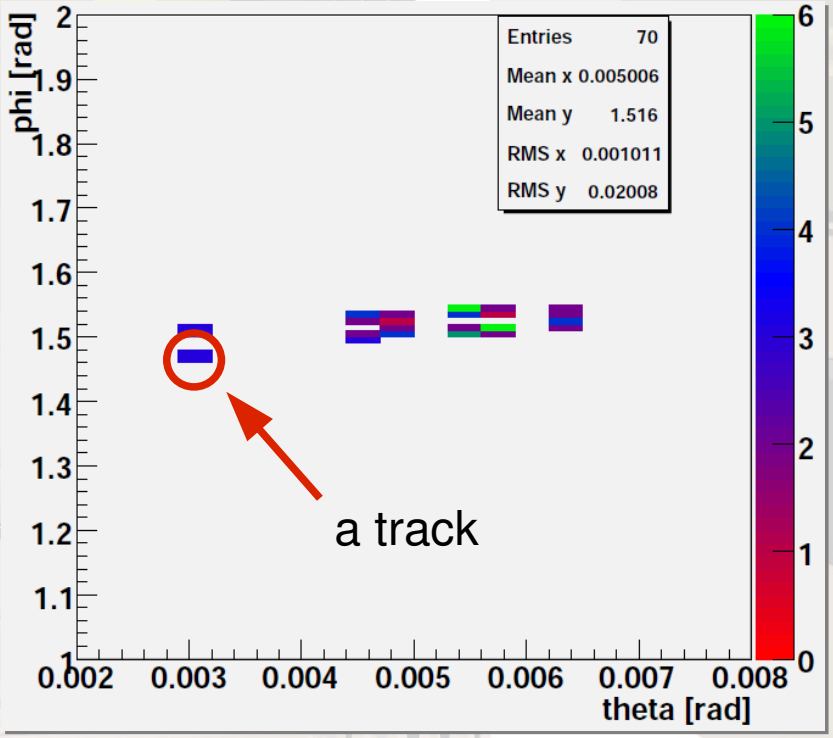
Characterisitcs

no restriction on number of hits

reconstruction of multiple tracks

run-time: O(n)

not yet in svn!



3D-Straight-Line Fit

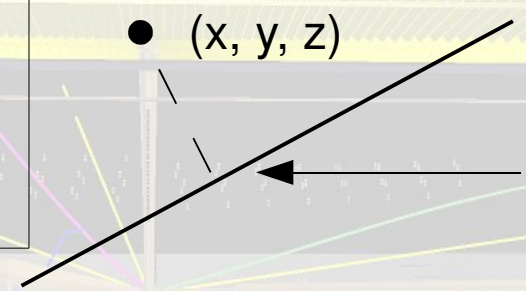
minimization of square-distance-sum
of track-points to a 3d-line

line:

$x = \mathbf{p_0} + \mathbf{p_1}\, t$

$y = \mathbf{p_2} + \mathbf{p_3}\, t$

$z = z_0 + t$



$$t_{min} = \frac{p_1 * (x - p_0) + p_3 * (y - p_2) + (z - z_0)}{p_1^2 + p_3^2 + 1}$$

$$X^2 = \sum \left(\frac{(x - (p_0 + p_1 * t_{min}))^2}{\sigma_x^2} + \frac{(y - (p_2 + p_3 * t_{min}))^2}{\sigma_y^2} + \frac{(z - (z_0 + t_{min}))^2}{\sigma_z^2} \right)$$

options:

- cut on χ^2
- take into account accuracy of hits
- additional weighting of hits

PndLinTrack

Container for Straight-Line-Trackfit

detName	Detector Name
p_x	Parameter of Fit
chi	ChiSquare of Fit
first	Id of first Hit
last	Id of last Hit
cand	Id of TrackCandidate

$$x = \mathbf{p_0} + \mathbf{p_1}\, t$$
$$y = \mathbf{p_2} + \mathbf{p_3}\, t$$
$$z = z_0 + t$$

Backtracking (Geane)

Input: $\vec{p} = |p_{beam}| * ||(p_1, p_3, 1)||$ in $\vec{x} = (p_0, p_2, z_0)$

Geane:

FairGeanePro::PropagateToPCA()

- uses same fieldmaps as Geant
- most effects are understood and under control

=> can be used for LMD-backtracking

Output: Θ and φ in point of closest approach (PCA) to IP

Summary

- simulation & hit-reconstruction based on SDS
- track-reconstruction for straight lines
- backtracking through magnetic fields via Geane

LMD-full-sim available in Pandaroot

Thank you for your attention!