LMD Tracking & Alignment in FairRoot

Stefan Pflüger

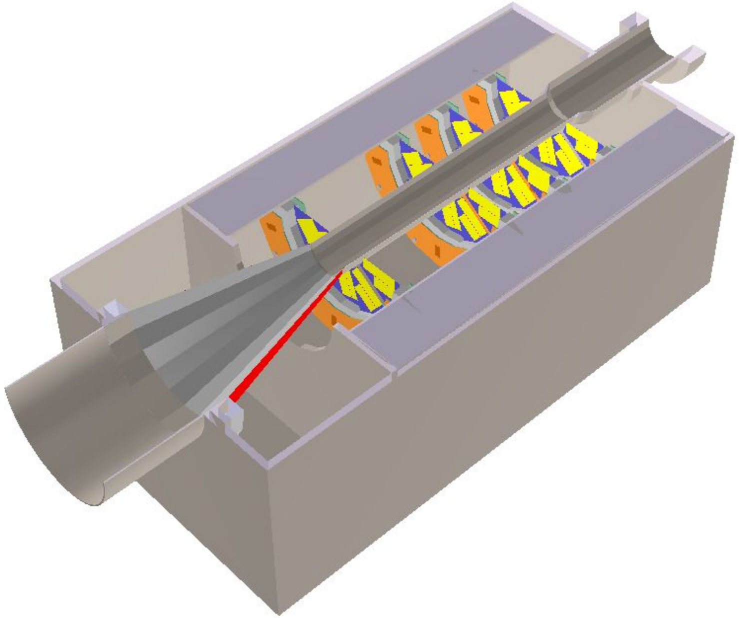
LMD Geometry & HVMAP Sensors

Used for PandaRoot simulations and reconstruction

Notes:

- Construction of the geometry was completely renewed and improved
- Unfortunately: The tracking performance plots of this talk use an older geometry
  (results are similar though)

Current Lmd Detector Geometry



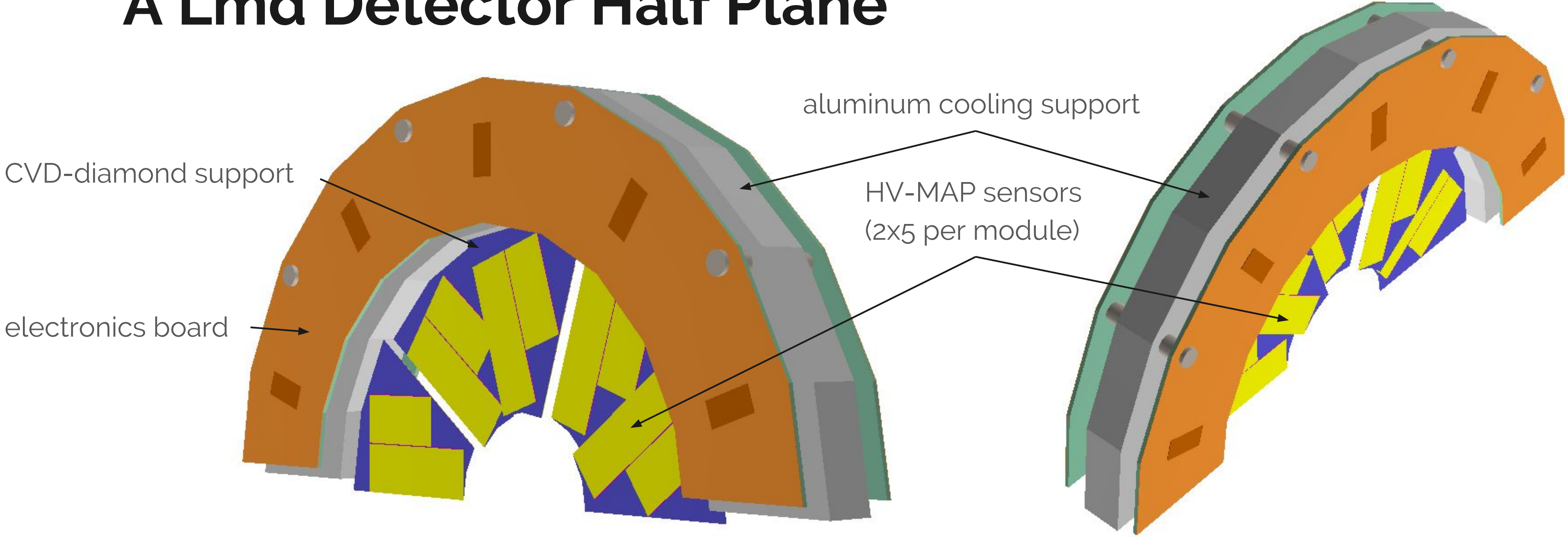
Overview of Lmd detector geometry:
(figure shows only lower half)

- steel box + reinforcements
- beam pipe + flanges + transition cone (red cone)
- detector planes (alltogether 4)

possible todos (currently we believe they are negligible):

- use a realistic vacuum in the steel box
- add more steel reinforcements and cooling equipment

A Lmd Detector Half Plane



front view of an lmd detector half plane

rear view of half plane

Track Reconstruction

Input: Lmd detector digis (event based)

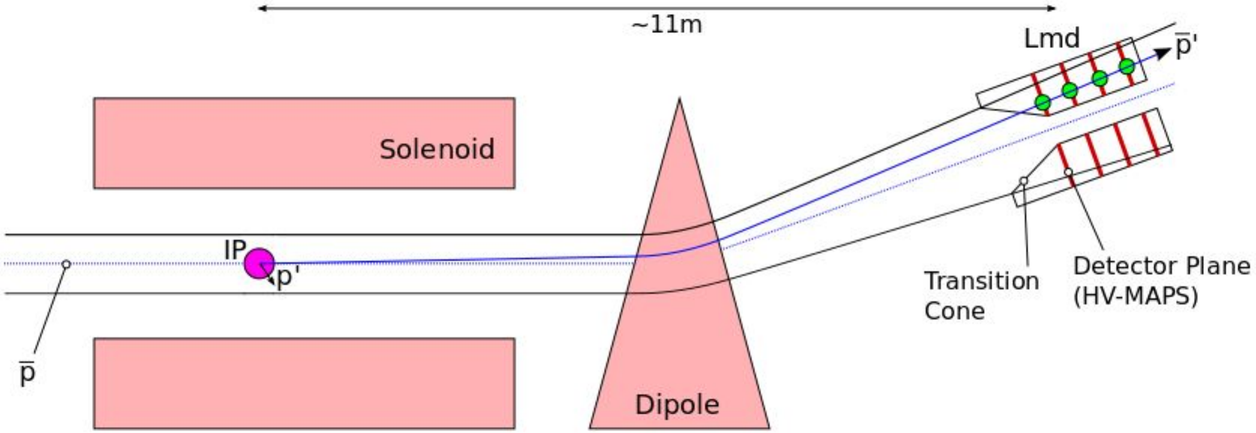
Output/Goal: Lmd track parameters

Notes:

- Fairroot/Pandaroot can simulate input (using Geant+ROOT)
- Performance results are using and **older** geometry

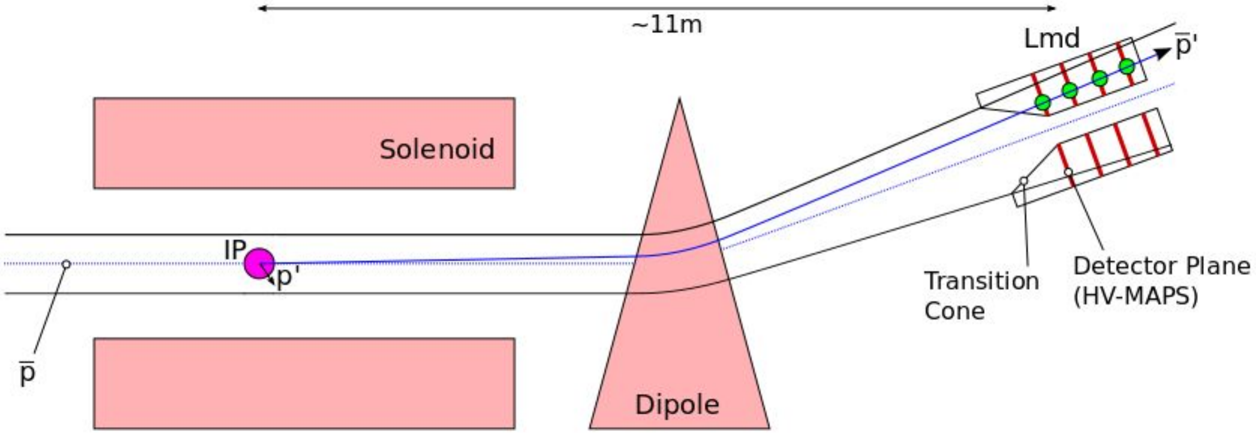
Track Reconstruction Steps

1. **Hit Reconstruction**
   - convert digi info to 3D point
   - hit clustering
   - optional: merging hits from different sides of modules



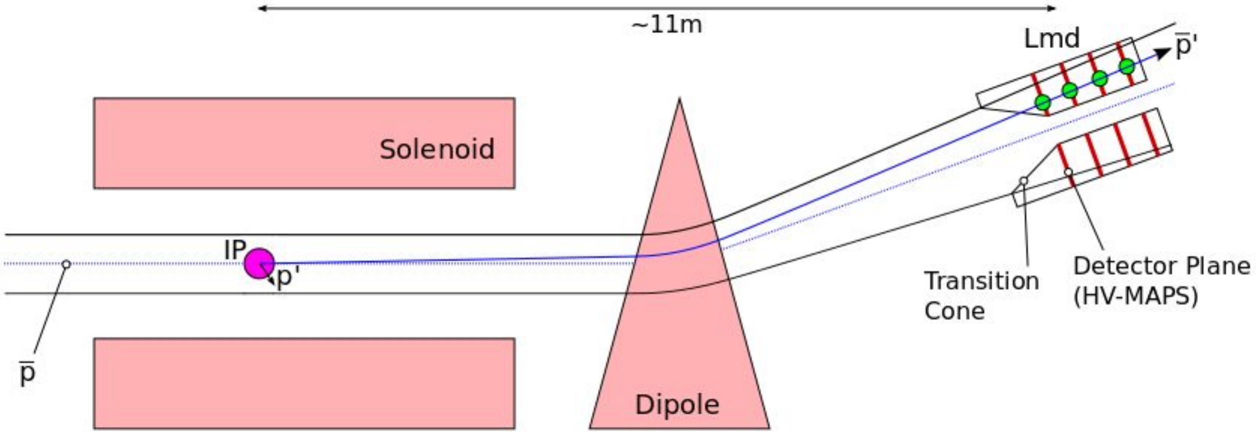
Track Reconstruction Steps

1. Hit Reconstruction
2. **Track Search**
    - find hits that make a meaningful track candidate
    - two methods available: cellular automaton (CA) track following



Track Reconstruction Steps

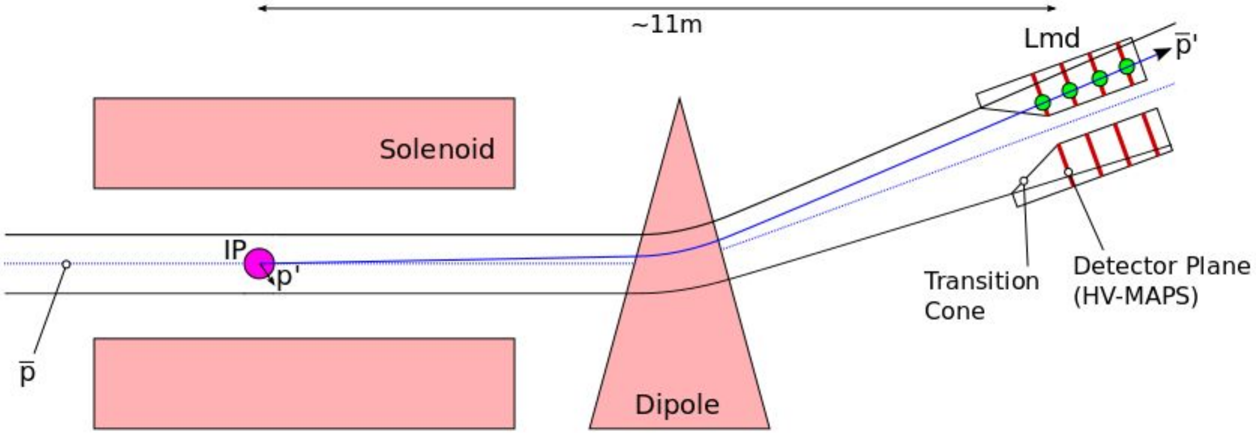
1. Hit Reconstruction
2. Track Search
3. **Track Fit**
   - fit track candidate to obtain precise track parameters + errors
   - using broken line model [1]



[1] Lutz, G., “Optimum Track Fitting in the Presence of Multiple Scattering” Nucl.Instrum.Meth.A273 (1988)

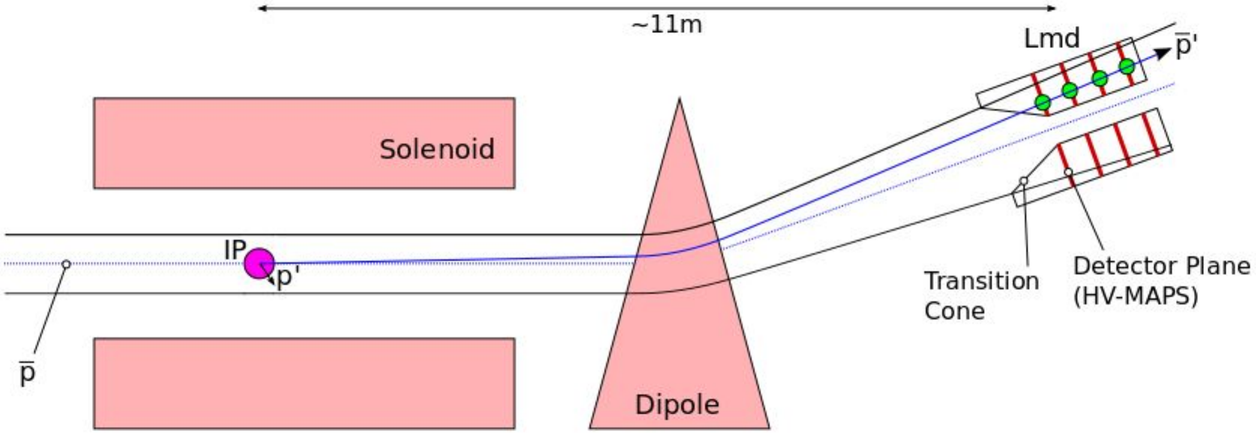
Track Reconstruction Steps

1. Hit Reconstruction
2. Track Search
3. Track Fit
4. **Track Filtering (1st)**
   - filter out "good" $\bar{p}$ tracks based on characteristic track parameters
   - using multivariate analysis (MVA)



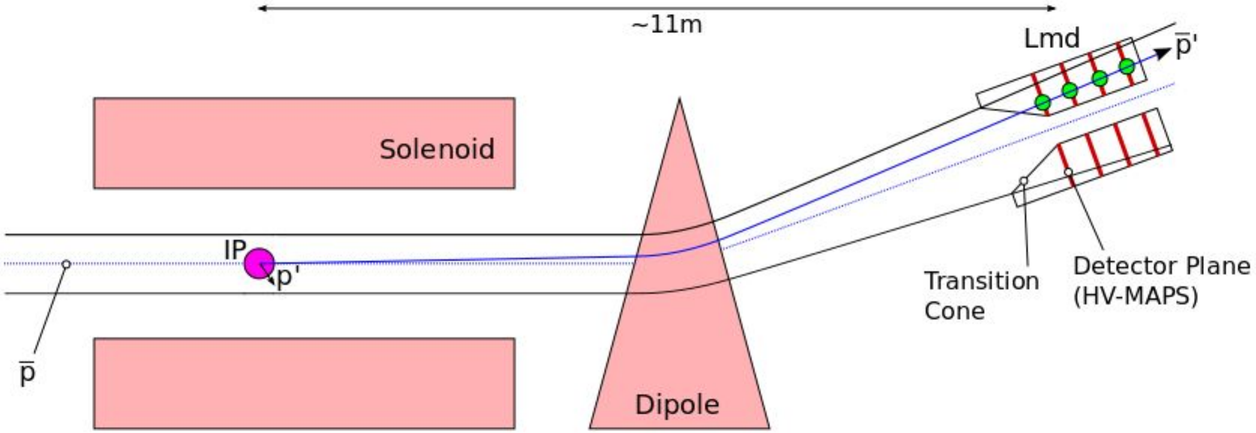
Track Reconstruction Steps

1. Hit Reconstruction
2. Track Search
3. Track Fit
4. Track Filtering (1st)
5. **IP Backtracking**
   - propagate tracks to IP (necessary for luminosity determination)
   - using Geane (Geant3) and point of closest approach (PCA) to IP



Track Reconstruction Steps

1. Hit Reconstruction
2. Track Search
3. Track Fit
4. Track Filtering (1st)
5. IP Backtracking
6. **Track Filtering (2nd)**
    - filter out “good” $\bar{p}$ tracks mainly based on track position info
    - deviation for background tracks, due to “wrong” momentum assumption

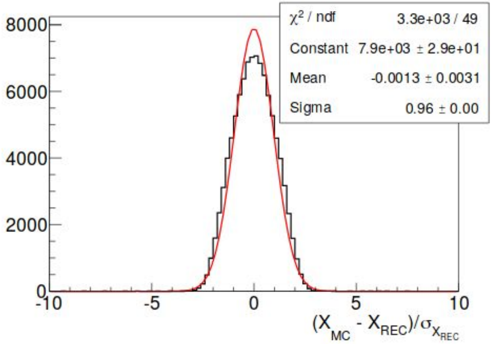


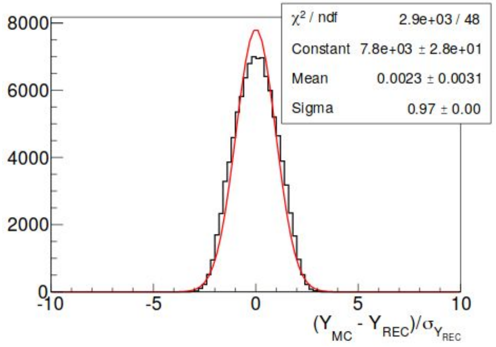
Track Parameter Resolutions

Beam Momentum	Track Parameter	weighted hits	broken lines
1.5 GeV/c	X_{start}, μm	14.12±0.02	14.03±0.02
	Y_{start}, μm	13.90±0.02	14.04±0.02
	P_x, keV	515±2	444±2
	P_y, keV	472±2	443±2
	P_z, keV	21±0.06	18±0.1
	θ, μrad	341±1	293±1
	ϕ, mrad	6.63±0.04	6.21±0.03
15 GeV/c	X_{start}, μm	13.86±0.02	13.89±0.03
	Y_{start}, μm	13.86±0.03	13.89±0.03
	P_x, keV	946±2	945±2
	P_y, keV	948±2	946±2
	P_z, keV	38.2±0.1	38.1±0.1
	θ, μrad	63.1± 0.2	63.1±0.2
	ϕ, mrad	1.58±0.01	1.58±0.01

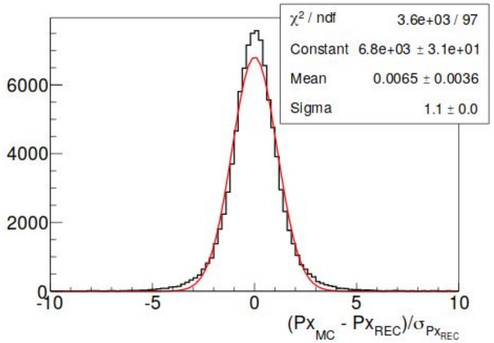
- track parameters at first LMD plane
- broken line fit is better at low momenta
  (multiple scattering)
-

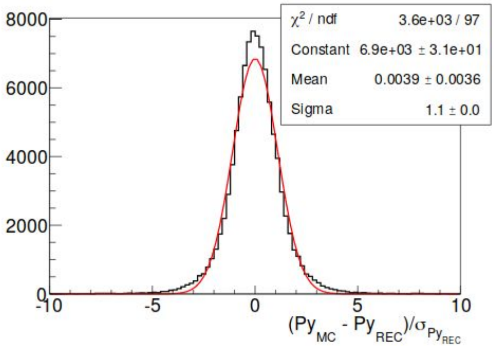
Broken Line Fit: Track Parameter Pulls

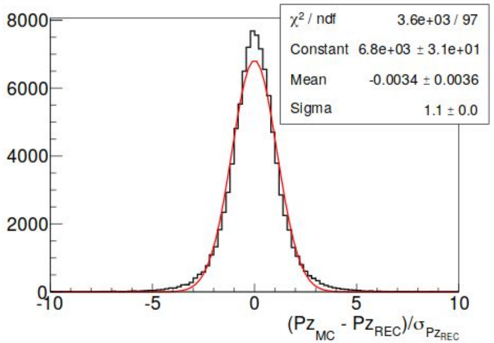




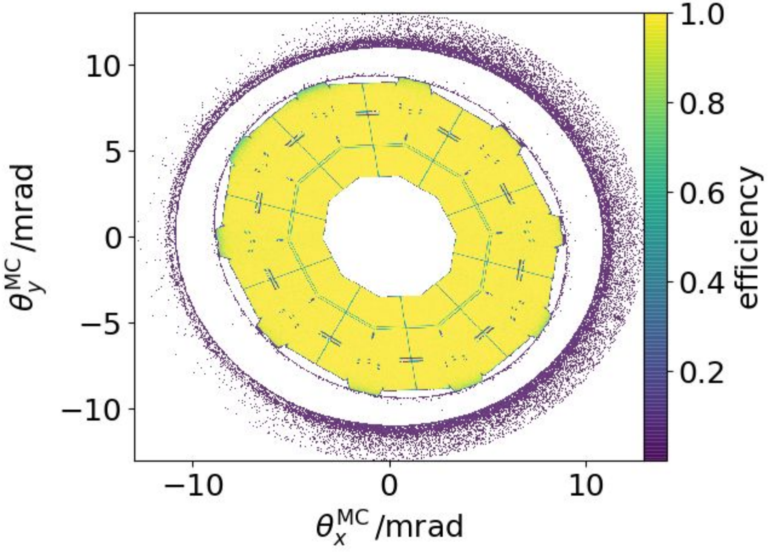
superior error estimation
(compared to simple straight line fit)



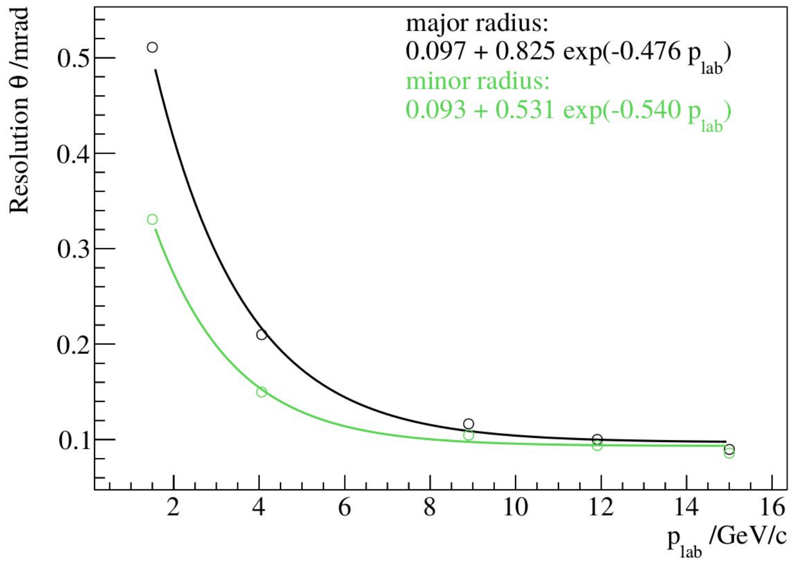




Full Angular Track Efficiency & Resolution



(@ 1.5 GeV/c antiproton beam momentum)



New Alignment Module in FairRoot

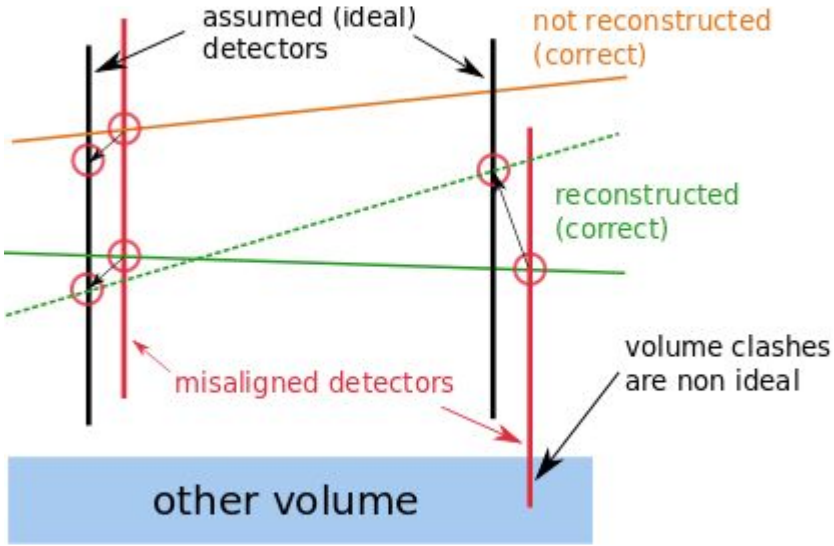
Under development!
FairRoot pull request
coming soon!

Current Status:

- done: implementation is complete
- now: performing some last tests
- next: pull request

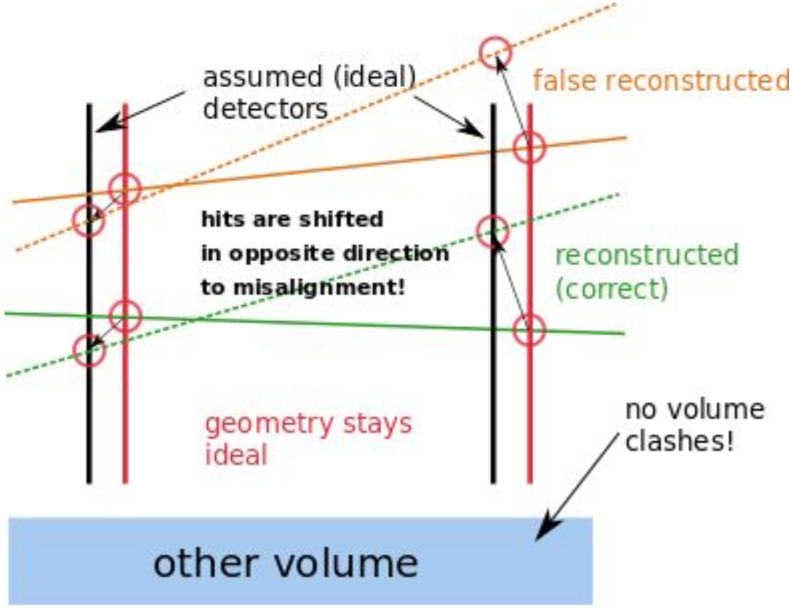
FairROOT alignment module

- all FAIR experiments get mis-/alignment features out of the box
- extremely simple to use (see example)
- use is optional (no modifications in macros necessary)
- two misalignment possibilities:
  - **geometry modification:**
    Pros: correct material budget + no fake induced hits
    Cons: volume clashes may appear
    -> shadows (hits not reconstructed)



FairROOT alignment module

- all FAIR experiments get mis-/alignment features out of the box
- extremely simple to use (see example)
- use is optional (no modifications in macros necessary)
- two misalignment possibilities
    - geometry modification
    - **hit transformation:**
      Pros: no simulation required (fast) + no shadows from volume clashes
      Cons: “fake” hits are reconstructed (individual detector groups have to filter out “fake” hits)



Example: Reco Macro

```
....
FairRunAna *fRun = new FairRunAna();
....

std::map<std::string, TGeoHMatrix> misalign_matrices = getMatrices();

bool invert_matrices(true);
fRun->AddAlignmentMatrices(misalign_matrices,  invert_matrices);

std::map<std::string, TGeoHMatrix> align_matrices = getMatrices();
fRun->AddAlignmentMatrices(align_matrices);
....
fRun->Init();
....
```

IMPORTANT: the alignment module calls have to be made before fRun->Init();

getMatrices(): dummy code for getting matrices. Currently LMD group generates these matrices separately. It is foreseen to have misalignment generating functionality in the alignment module

For misalignment via hit transformation , the inverse matrices have to be applied
(only used for misaligment studies)

Alignment matrices can also be stacked on top!
(for misalignment studies and alignment @ PANDA runtime)

The End

Thanks for your attention!