Linear Collider Track Reconstruction Tools

Frank Gaede, DESY
LCWS-2015
Whistler, Canada, 2.11-6.11.15

iLC SOft

Outline

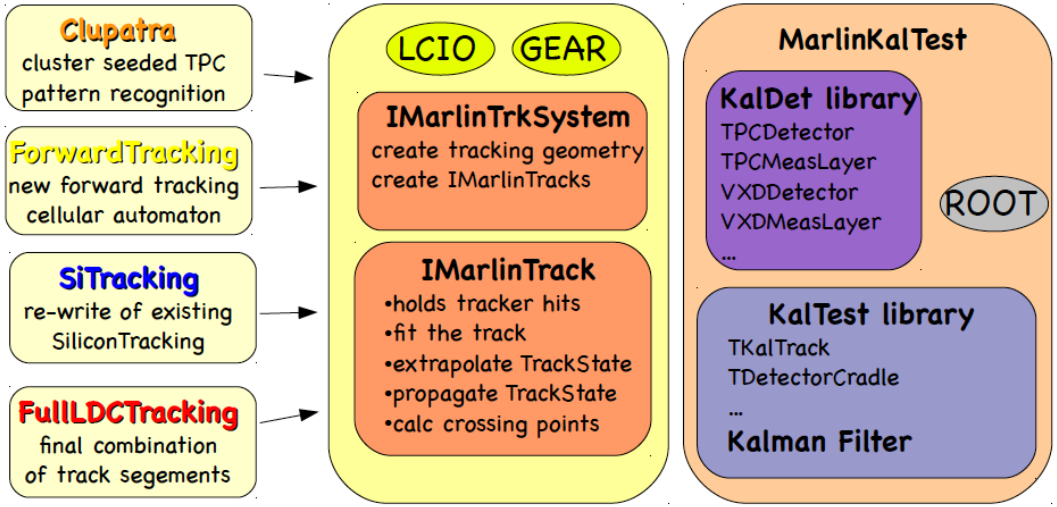
- Introduction
  - ILD tracking
- Geometry for tracking
- Core tracking tools
  - DDKalTest, aidaTT, MarlinTrk
- Latest developments in pattern recognition
- Summary Outlook

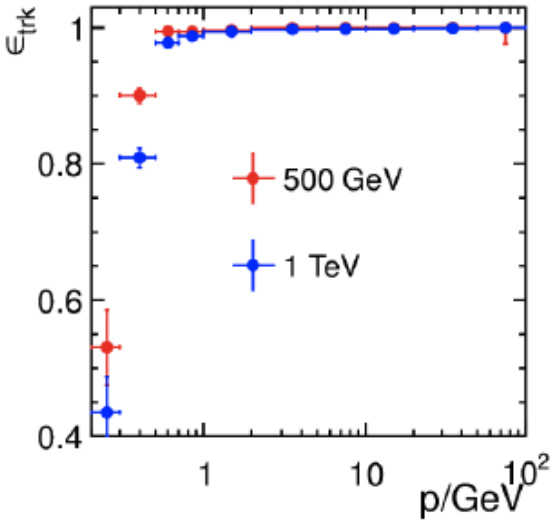
Introduction

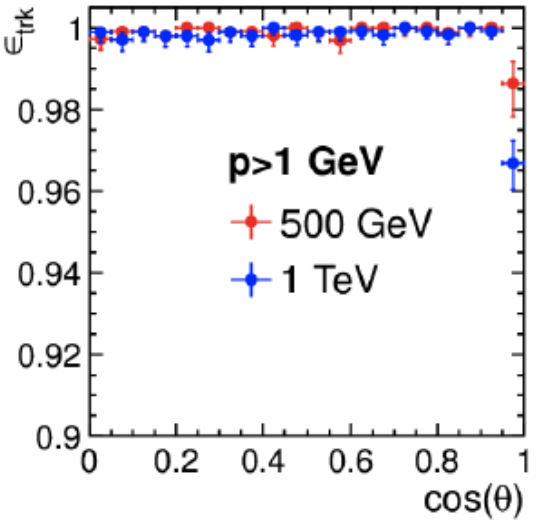
- developed new C++ tracking tools for the ILD DBD
  - to replace old F77 code from LEP
  - successfully used for ILD DB
- done in context of AIDA-WP2 project
- goal: eventually have a generic HEP tracking toolkit that could be shared by all LC detector concept groups
  - allowing to transparently use different fitting algorithms
  - provide toolkit for pattern recognition
  - have well defined and easy to use interface to detector geometry

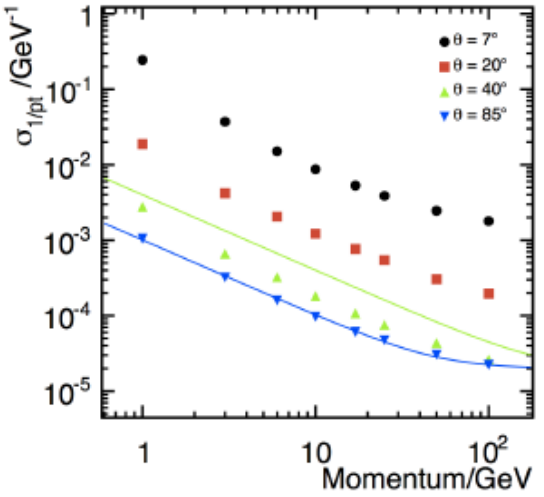
ILD track reconstruction

- KalTest Kalman filter (KEK)
- independent pattern recognition in TPC, Si, Fwd
- programmed against IMarlinTrk interface
- achieves performance goals for ILC





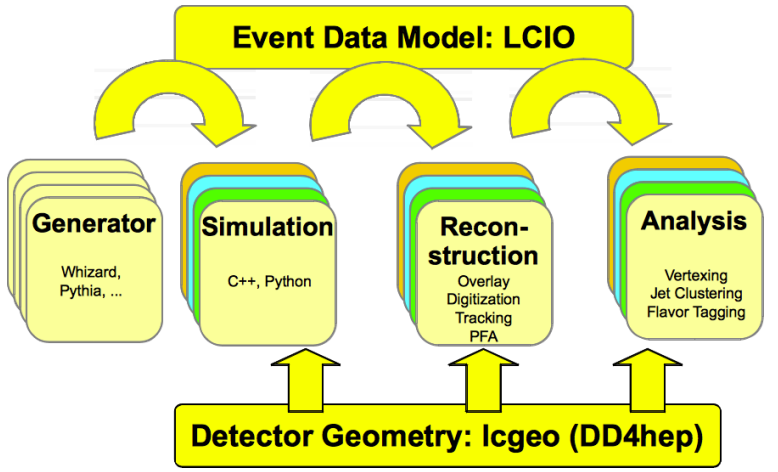


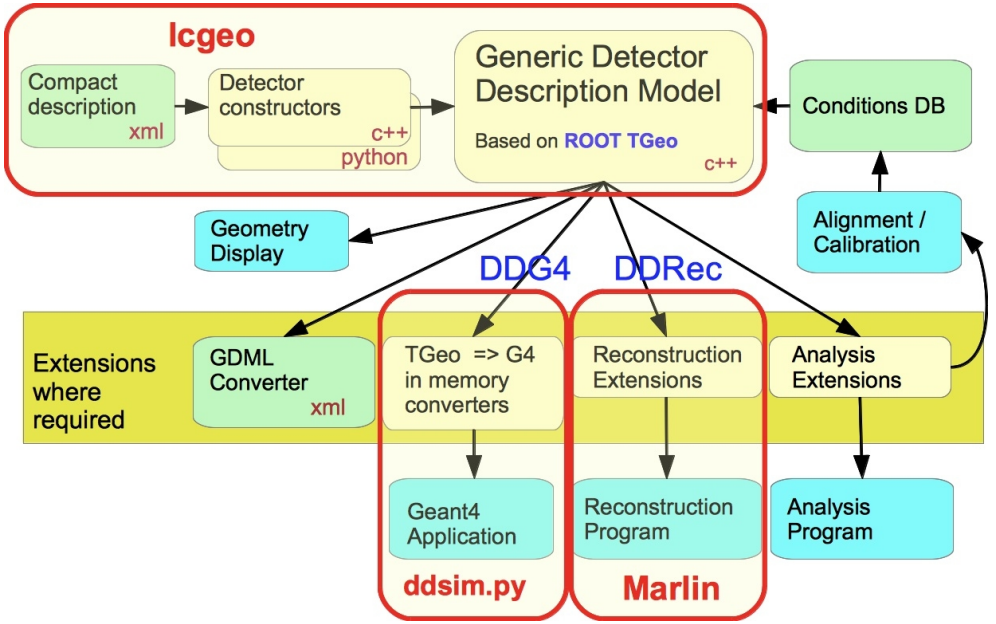


Why change a running system ?

- LC community is moving towards more common software tools
- decided to use the DD4hep geometry description (and simulation)
- excellent opportunity to make the tracking software
- more flexible and
- better maintainable

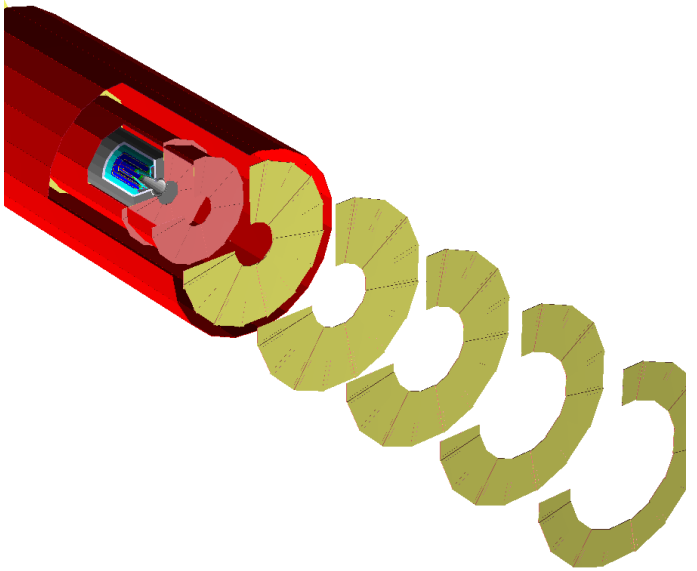
DDRec to replace GEAR

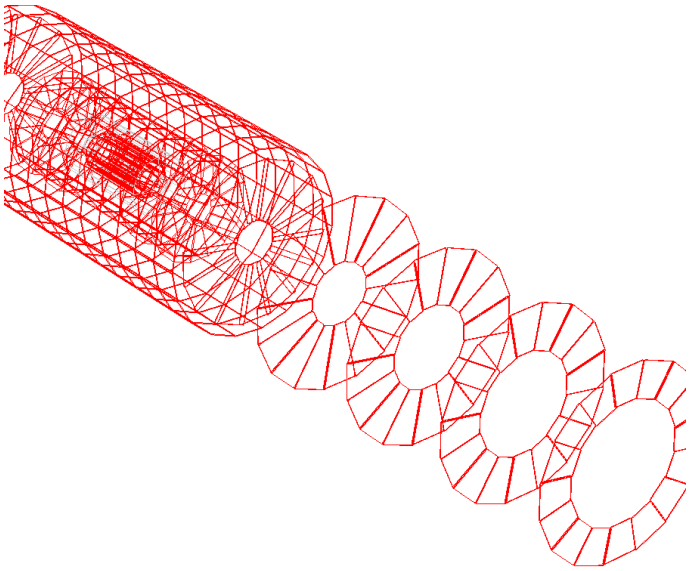




DDRec surfaces for tracking

- tracking needs special interface to geometry
- measurement and dead material surfaces (planar, cylindrical, conical)
- surfaces attached to volumes in detailed geometry model





- u,v, origin and normal
- inner and outer thicknesses and material properties
- local to global and global to local coordinate transforms:
- (x,y,z) ↔ (u,v)

automatic material averaging for surfaces

- material properties are averaged along normal of the surface
- along given thicknesses

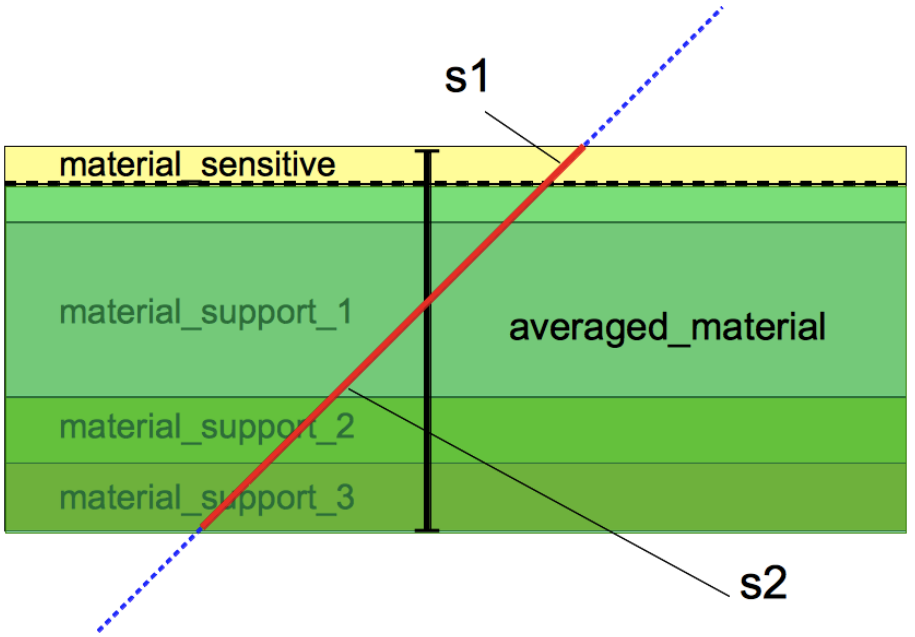
$$< A > = \left(\sum_i^N \rho_i t_i \right) / \left(\sum_i^N \rho_i \frac{t_i}{A_i} \right)$$

$$< Z > = \left(\sum_i^N \rho_i \frac{t_i Z_i}{A_i} \right) / \left(\sum_i^N \rho_i \frac{t_i}{A_i} \right)$$

$$< \rho > = \left(\sum_i^N \rho_i t_i \right) / \left(\sum_i^N t_i \right)$$

$$< X_0 > = \left(\sum_i^N t_i \right) / \left(\sum_i^N \frac{t_i}{X_{0i}} \right)$$

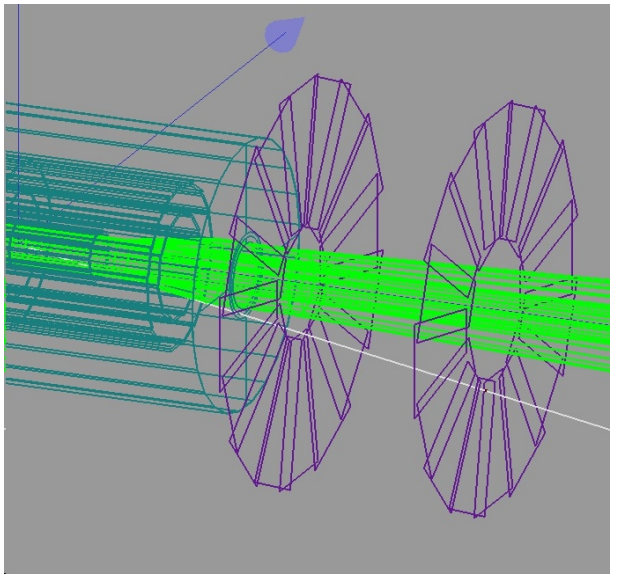
$$< \lambda > = \left(\sum_i^N t_i \right) / \left(\sum_i^N \frac{t_i}{\lambda} \right)$$



- roughly equivalent to individual materials for Bethe-Bloch
- identical for multiple scattering

DDKalTest

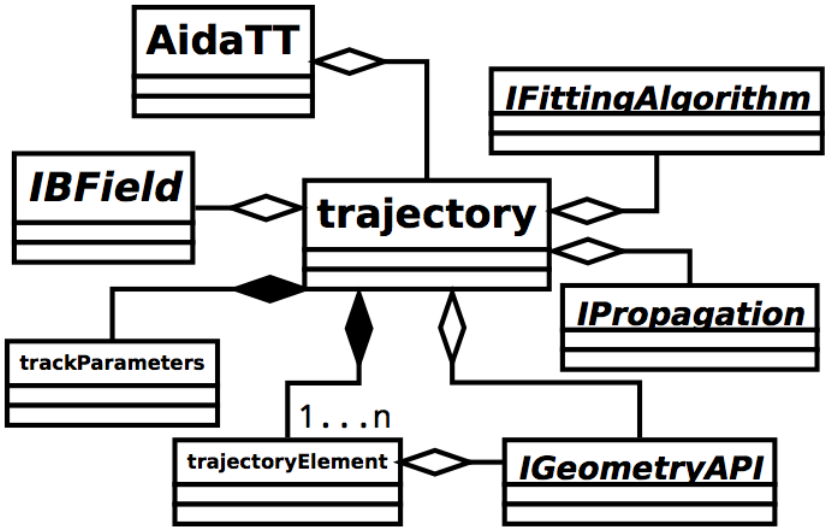
- new package that provides measurement surfaces needed by KalTest using DDRec::Surfaces:
- DDPlanarMeasLayer
  - 1D,2D Si-tracker - barrel/endcap
  - dead materials (endcaps)
- DDCylinderMeasLayer
  - 2D hits in TPC
  - supports (cryostat, field cage,...)
- DDConeMeasLayer
  - conical sections of beam pipe



with DDKalTest we can run the track fitting for every detector that has a DD4hep geometry description (and the surfaces added) !

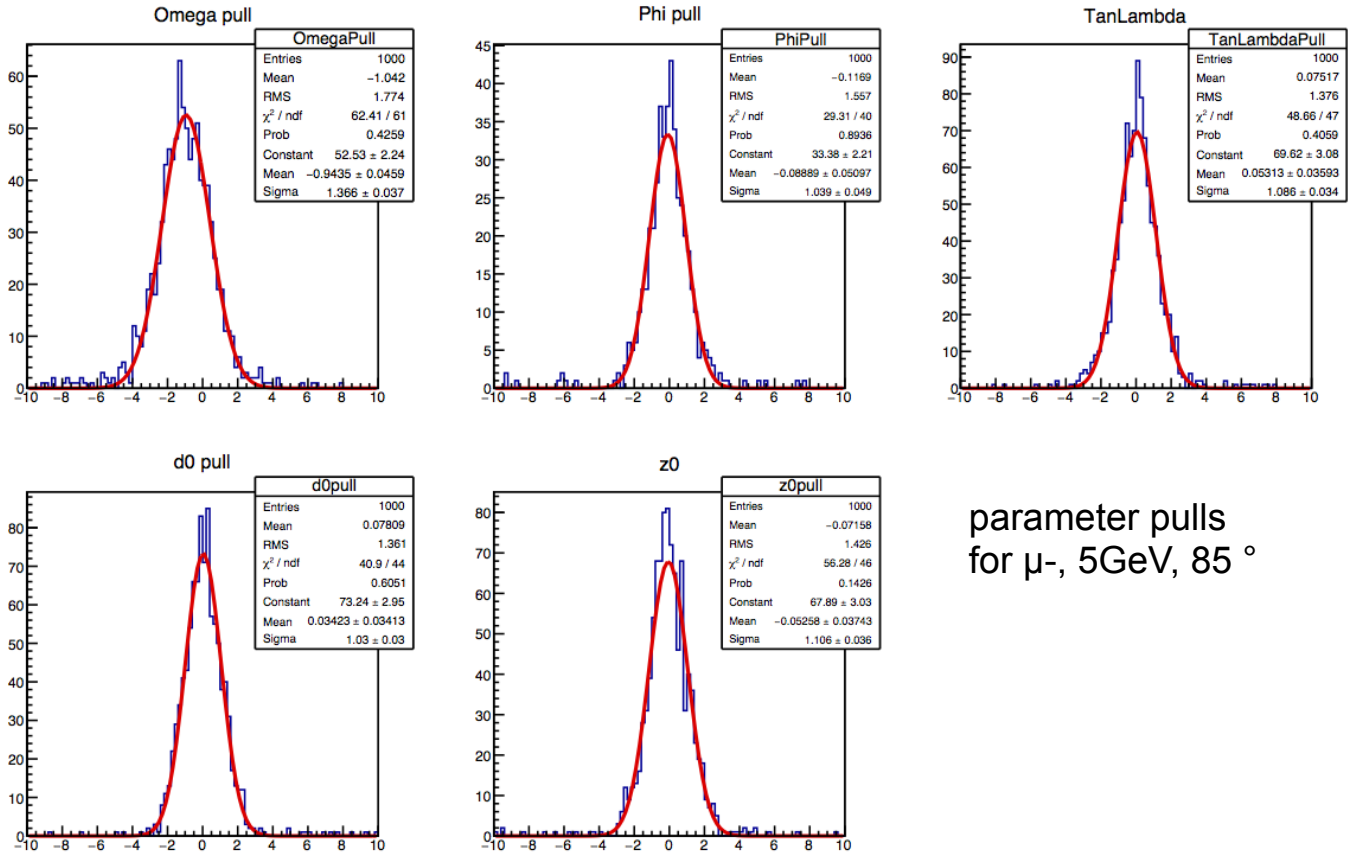
aidaTT

- generic tracking toolkit developed in AIDA-WP2
  - (C.Rosemann, Y.Voutsinas)
- can transparently use a Kalman Filter or the GeneralBrokenLines GBL
- GBL provides interface to Millipede alignment tool
- IGeometry interface uses DDRec::Surface

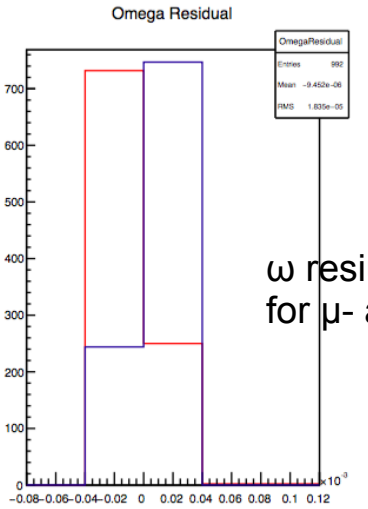


aidaTT - recent developments

Y.Voutsinas,FG



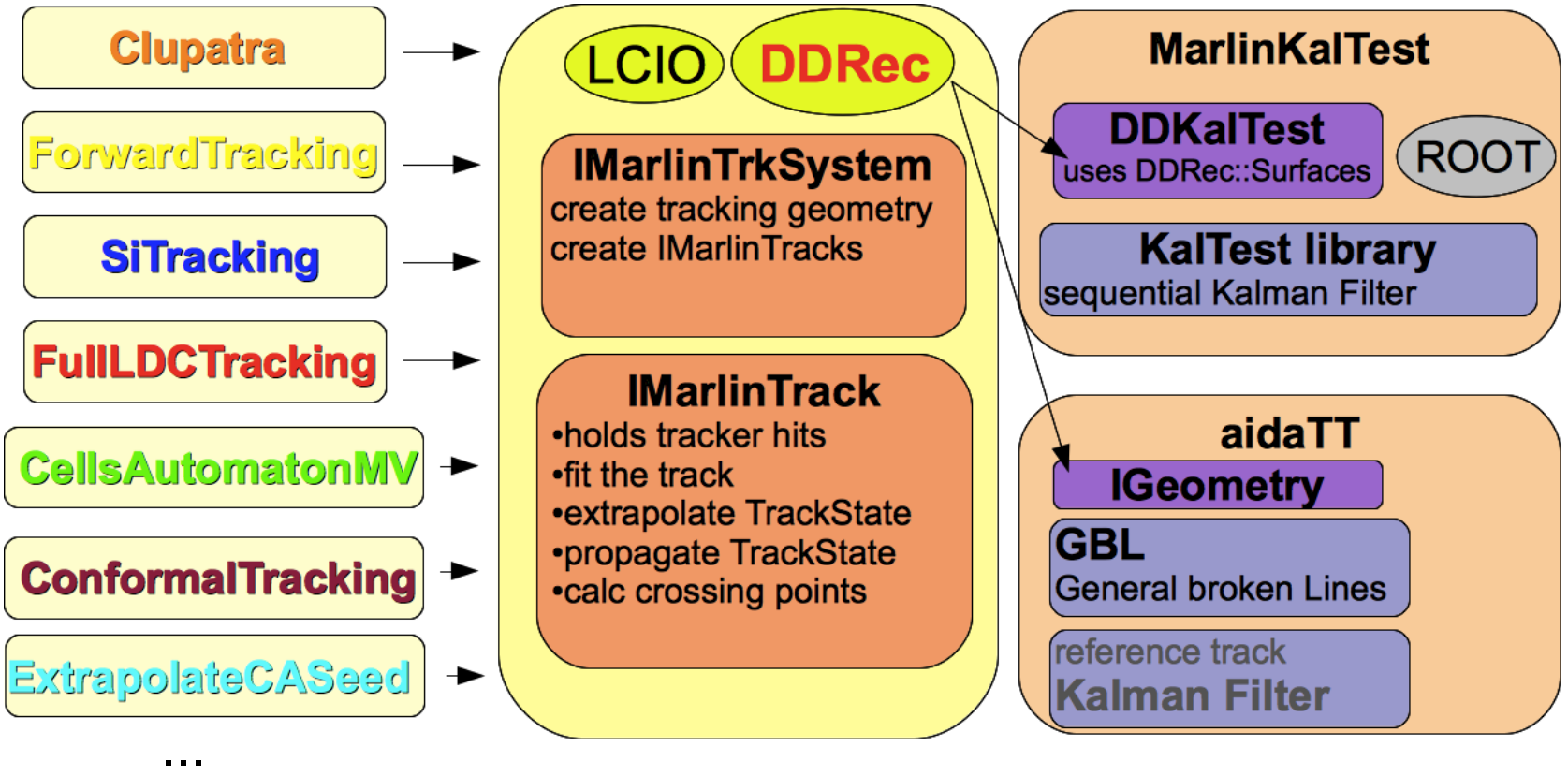
parameter pulls
for μ-, 5GeV, 85 °



ω resiudual
for μ- and μ+

- fixed a number of bugs
- implemented multiple scattering and energy loss
- still one issue w/ omega → study ongoing

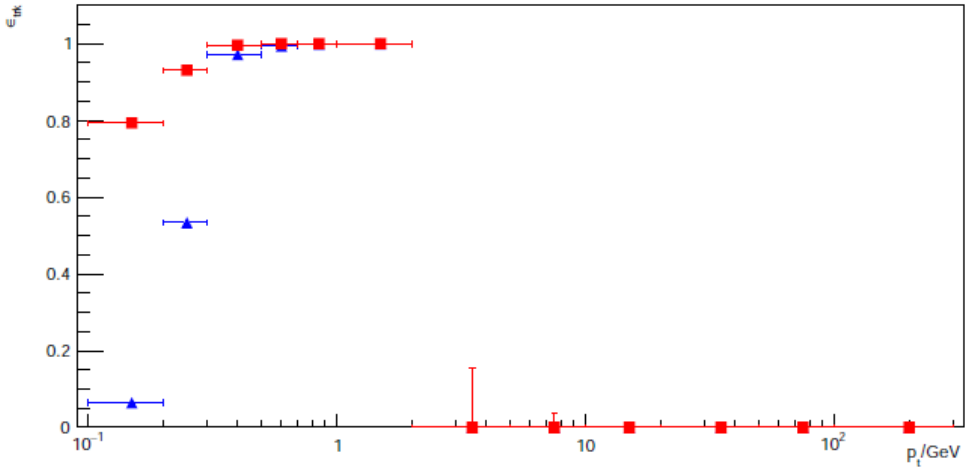
IMarlinTrk - LC Tracking Tools

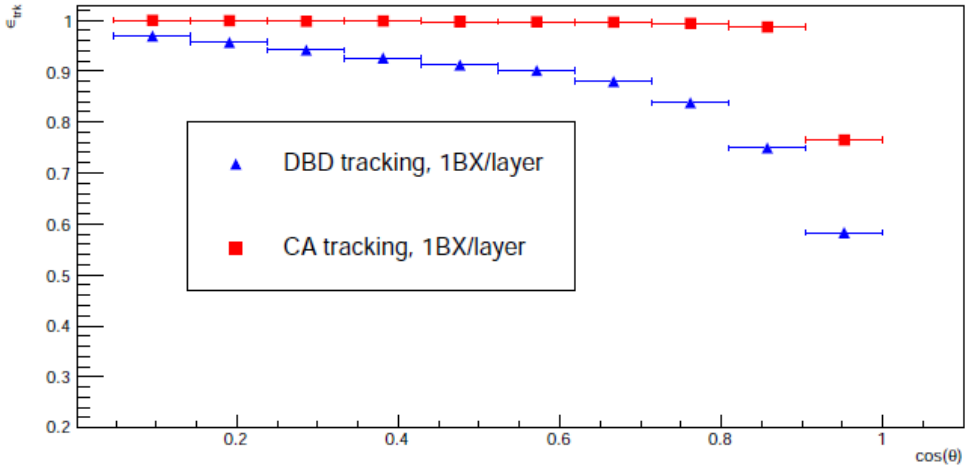


- IMarlinTrk interface implemented for aidaTT
- tracking tools are now fully compatible with DD4hep geometry
- new pattern recognition algorithms developed in parallel

CellsAutomatonMV

Y.Voutsinas

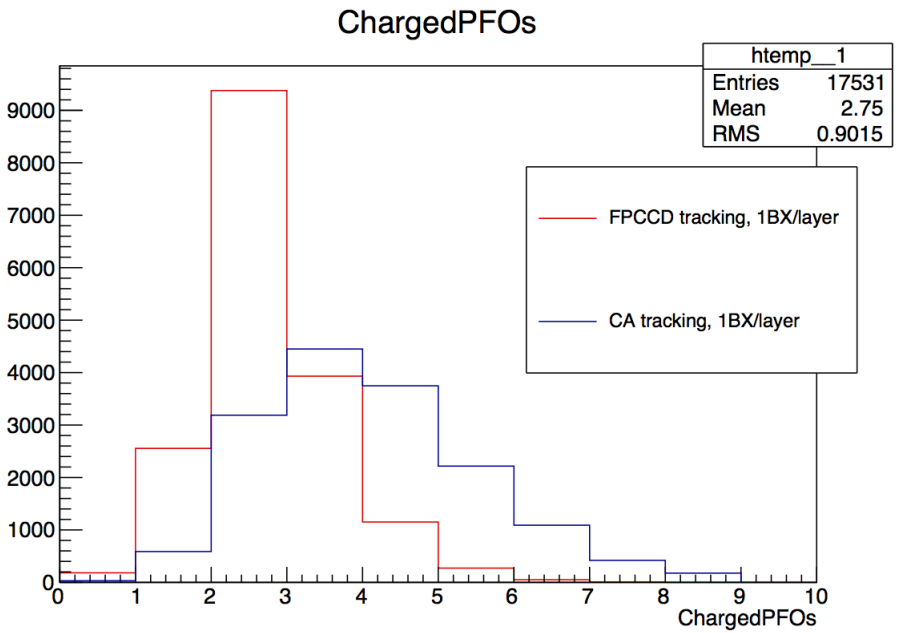




- standalone pattern recognition algorithm in VXD using mini-vectors and cellular automatons
- examine the low momentum performance in context of a light higgsino scenario
  - very few soft particles & missing energy in the final state
- compare to alternative algorithms:
  - DBD & FPCCD (require ≥ 1hit in SIT for seeding)
- => improved track finding efficiency

CellsAutomaton II

Y.Voutsinas



	Cons. CMOS		Ambitious CMOS	
layer	σ_{sp} (μm)	σ_{time} (μs)	σ_{sp} (μm)	σ_{time} (μs)
L1 / L2	4 / 4	4 / 4	3 / 3	1 / 1
L3 / L4	4 / 4	8 / 8	3 / 3	2 / 2
L5 / L6	4 / 4	8 / 8	3 / 3	2 / 2

- improved tracking efficiency reflected in particle reconstruction efficiency for light Higgsino analysis
- however standalone VXD tracking suffers from higher efficiency at low pt:
  - reconstructs more tracks from pair bg
  - different R&D scenarios under study
  - currently still get best performance for Higgsino mass using Ambitious CMOS with older FPCCTracking incl. SIT (single BX tagging)
- trade off between efficiency and purity folded with kinematics
- need to further investigate ...

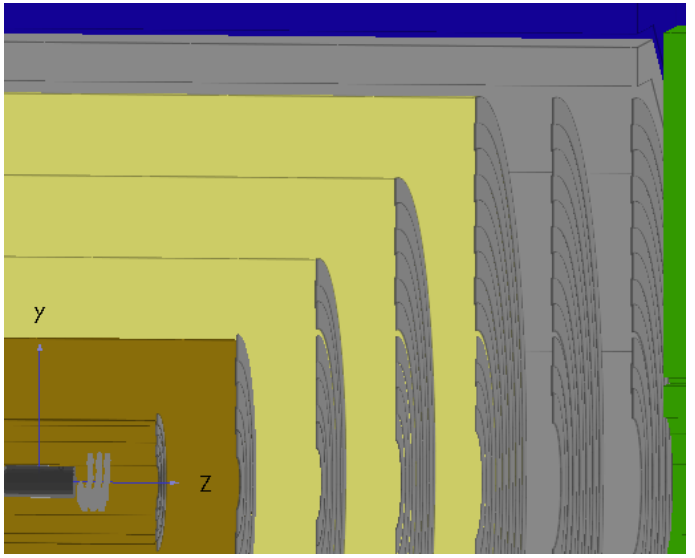
CA method for CLIC

R.Simoniello

- adapt and extend ILD patrec for the CLIC case
- find tracks according to helix trajectory
- different strategies in different subdetectors:
- tracks passing trough the vertex barrel use

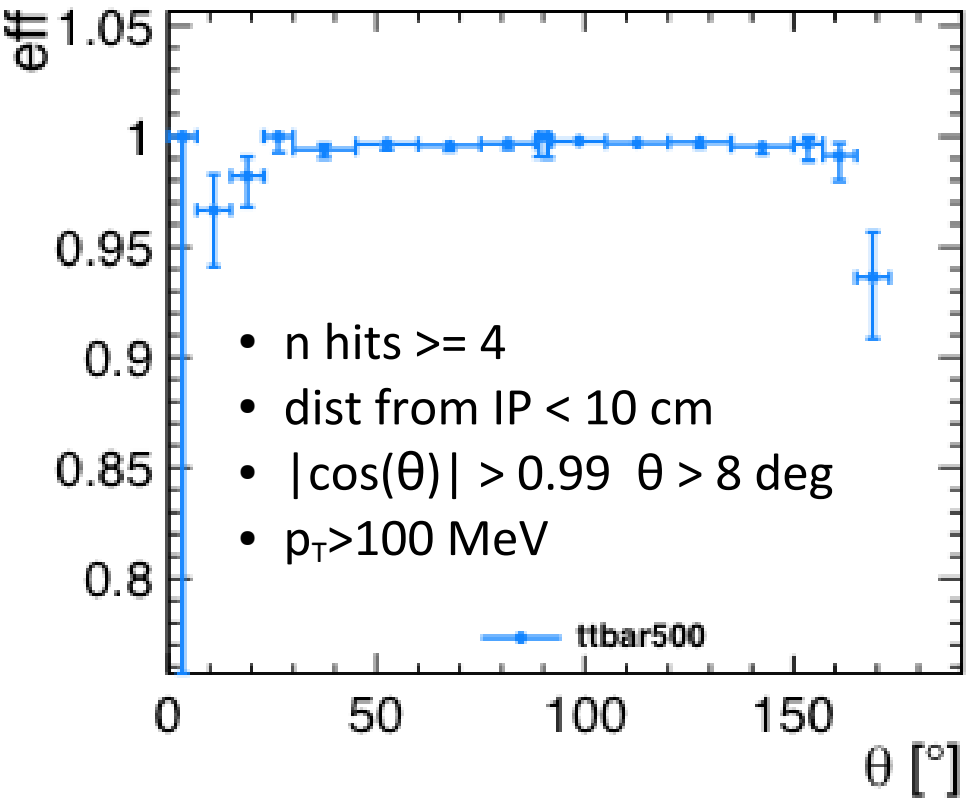
  mix of CA and Kalman Filter strategies
  - compute mini vectors in double layer vertex
  - run CA on MV → obtain vertex tracks
  - use vertex tracks as seed to track extrapolation to Inner and Outer Tracker (both in barrel and Endcap layers)
- tracks passing trough the vertex endcap use

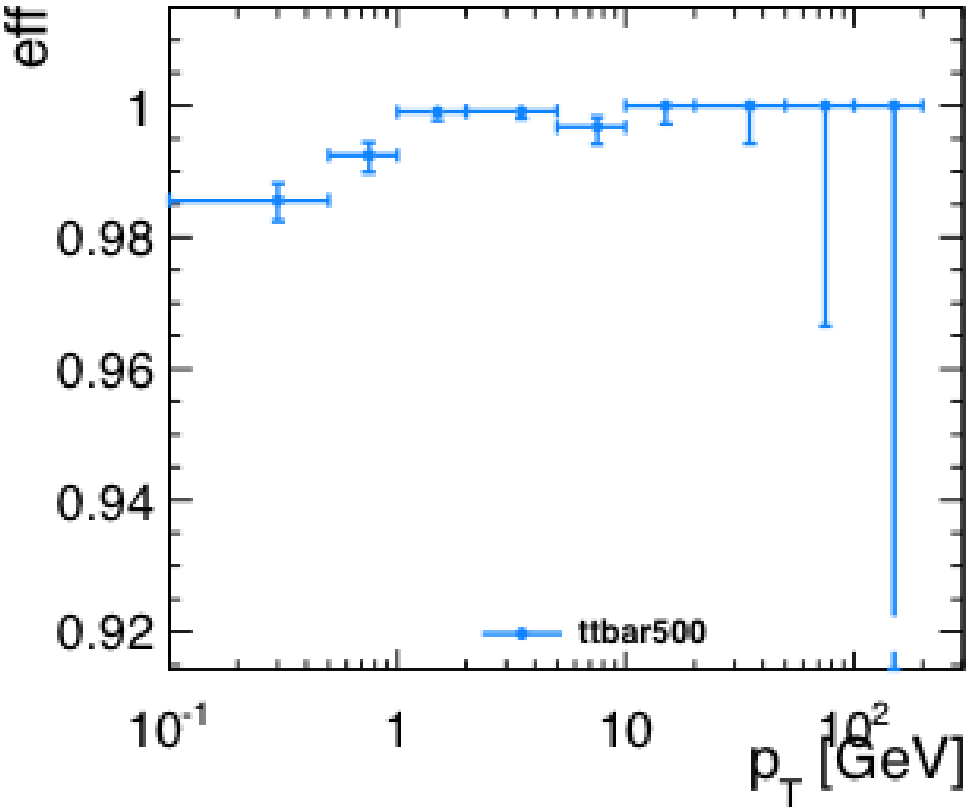
  pure CA strategy
  - run CA on vertex endcap hits, inner endcap hit, outer endcap hits
  - sectors definition (θ,φ,nlayer) to limit combinatorics



CA method for CLIC:results

R.Simoniello



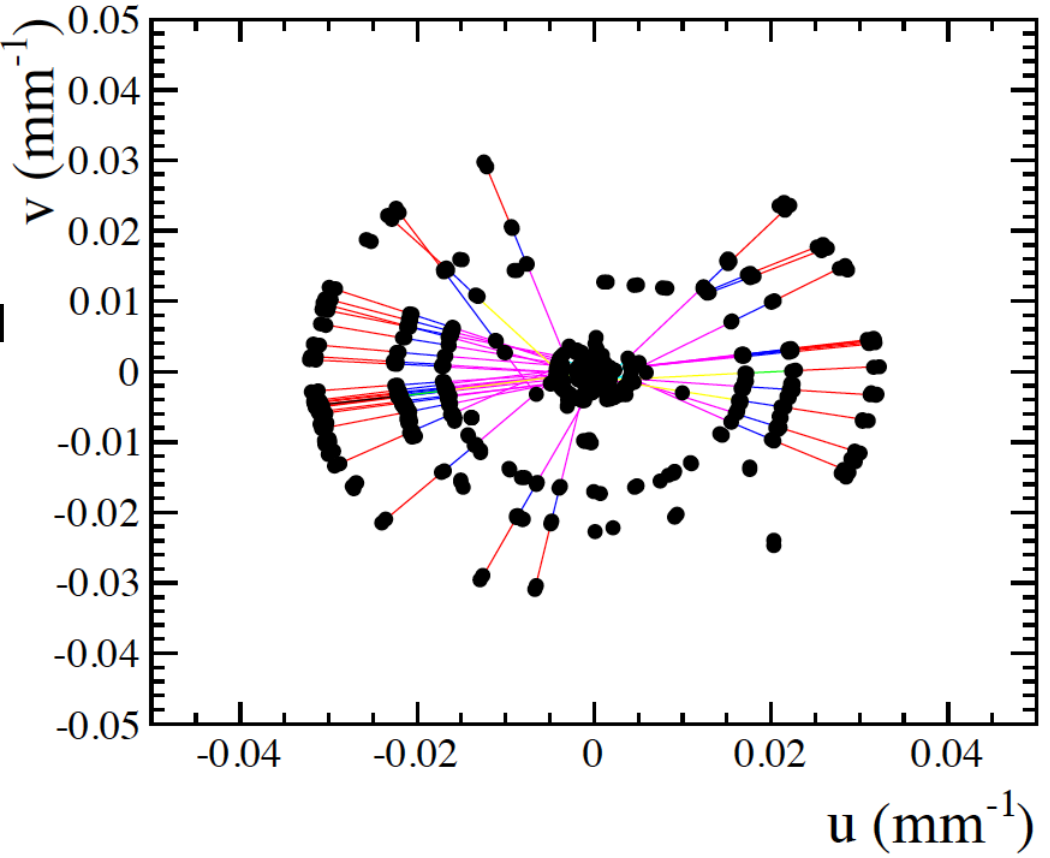


- achieve trk_eff > 98%
- no bg included so far → to be done

CLIC ConformalTracking

D.Hynds

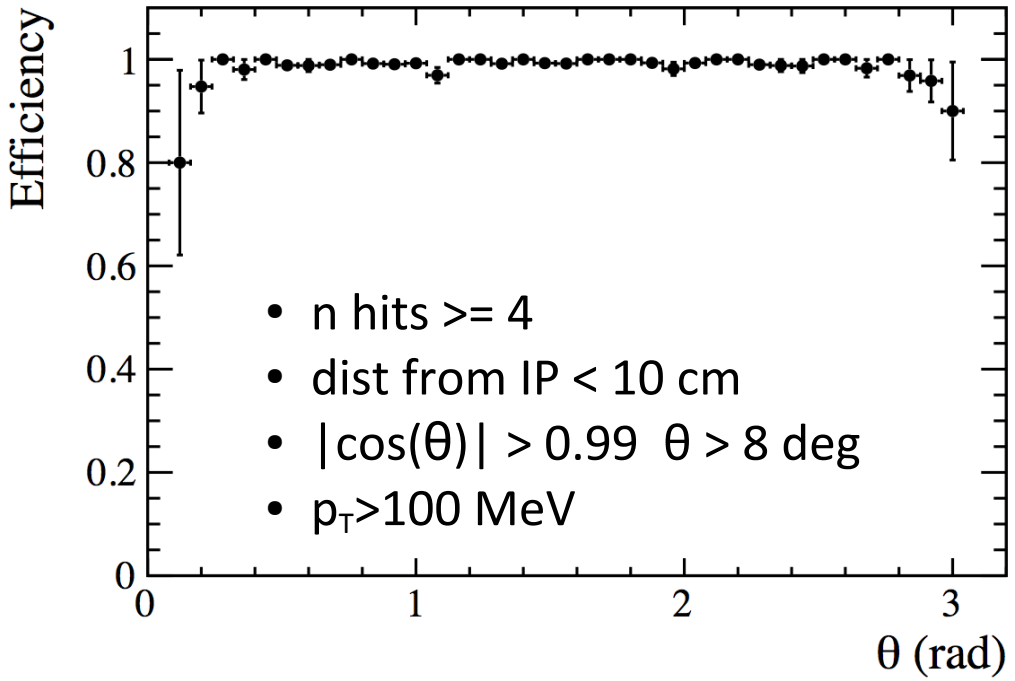
- apply conformal mapping to CLIC all Si-tracking
  - map x,y plane to u,v plane:
  - $u = x/r^2 , v = y/r^2 , r^2 = x^2 + y^2$
  - tracks (circles) from IP are mapped to straight lines
- run CA to find tracks in complete detector
- consistency criterion in z
- global method
- no geometry used !

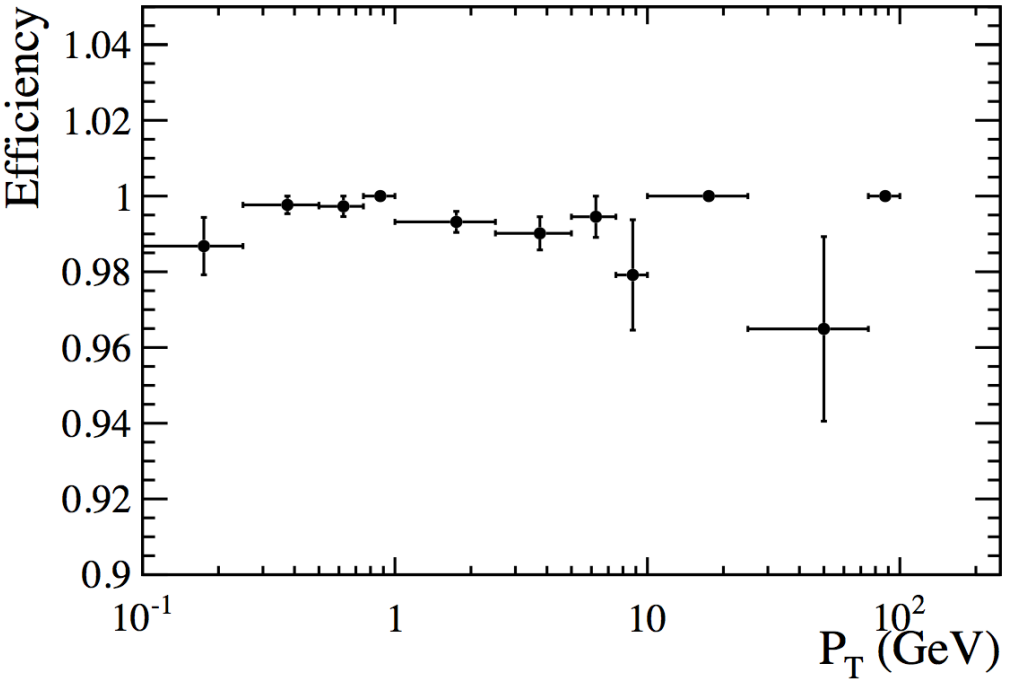


conformal mapping successfully used at Star and Alice

CLIC ConformalTracking - results

D.Hynds





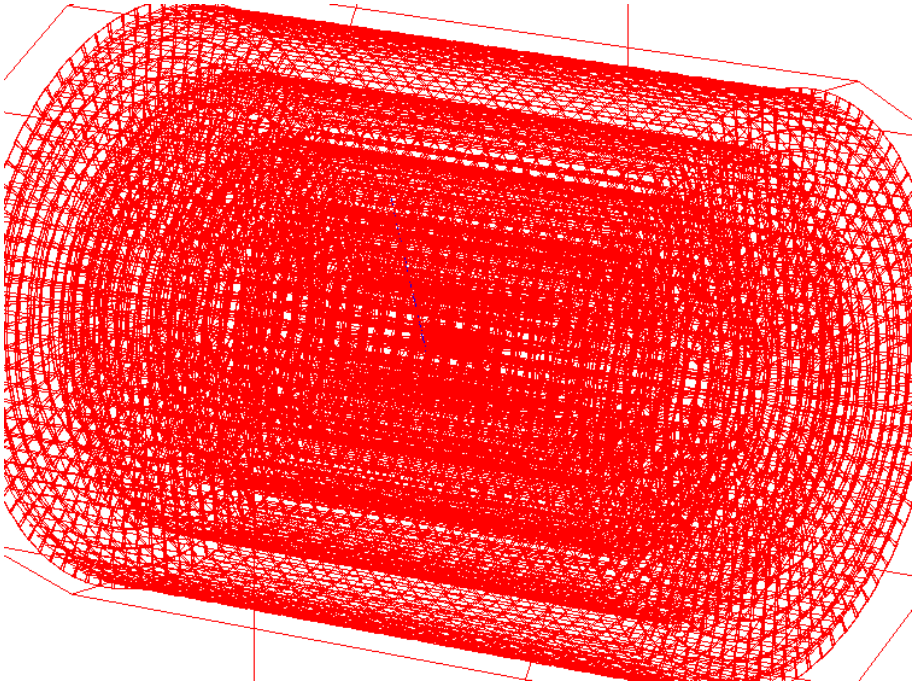
- also achieve trk_eff > 98%
- also no bg included so far

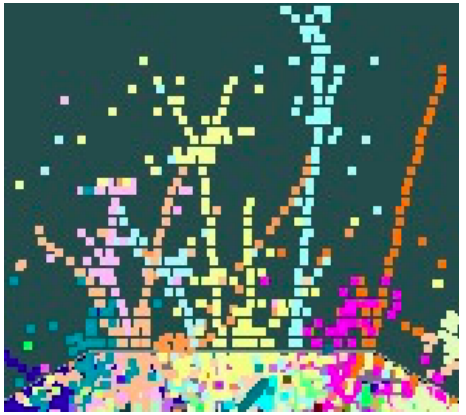
- CLIC will eventually use the method that gives best performance

see talk by Rosa on Thursday in joint Sim/Tracking session

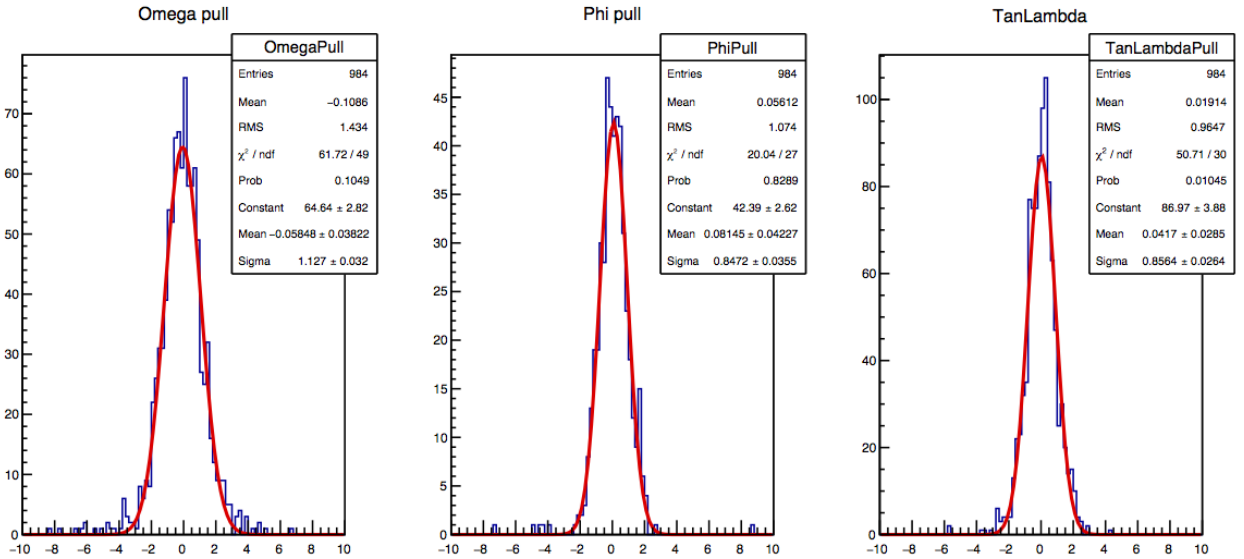
SiD in DD4hep/lcgeo

- implemented SiD model sidloi3 in lcgeo
- rather straight forward as DD4hep compact format originally based on SiD geomConverter
- tracking surfaces added automatically via plugin
- can directly run simulation with DDG4
- and try track fitting...

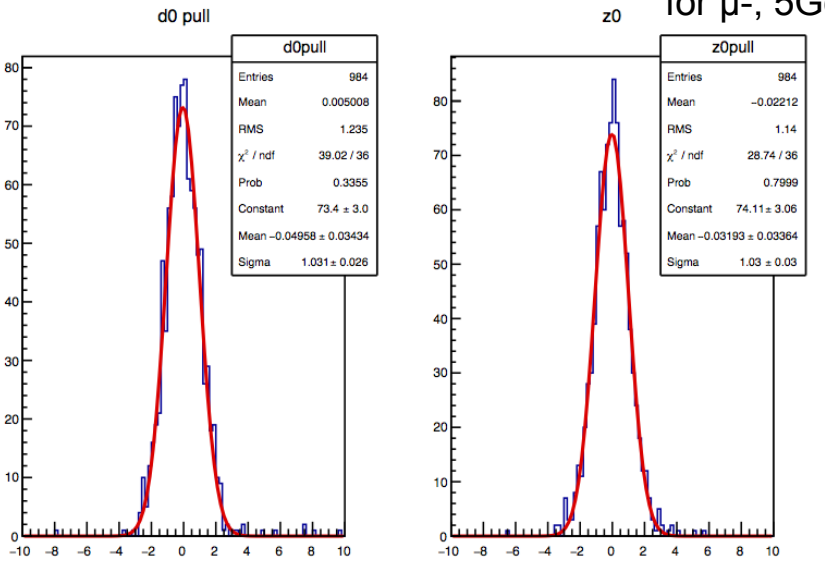


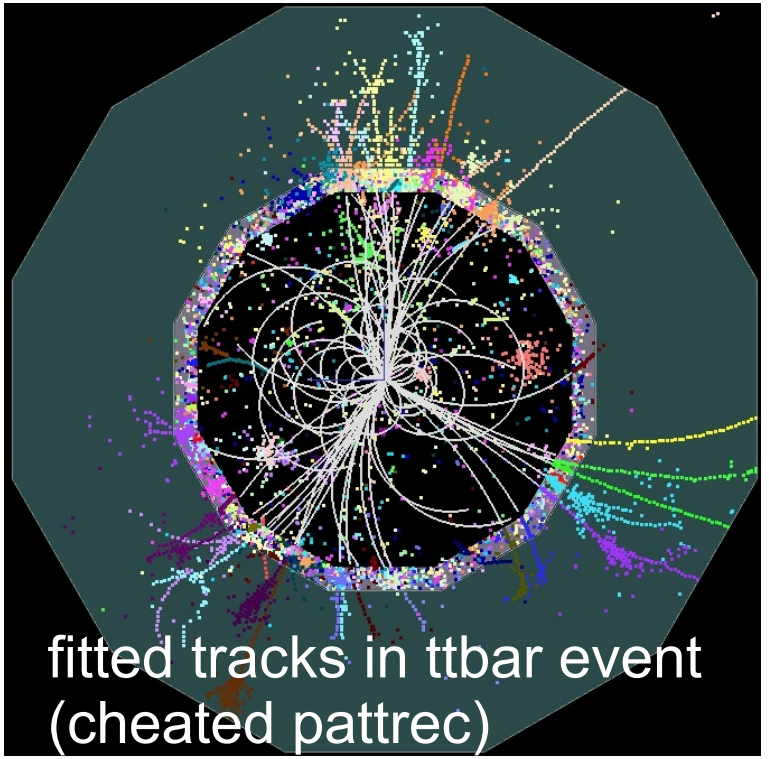


MarlinTrk tracking for SiD



parameter pulls
for μ-, 5GeV, 85 °





- proof of concept
  - further work needed
- demonstrates that tracking code really is generic

Summary & Outlook

- adapted existing ILD tracking to work with DD4hep
- generalized through use of DDRec::Surfaces
- new GBL fitting algorithm in MarlinTrk available
- code to be used by ILD, CLIC and possibly SiD

Outlook

- address remaining (minor) issues in core tools
- code needs to be made more robust for mass production
- start serious benchmarking (eff, perf, CPU,...)
- further improve pattern recognition
- study effects of background from pairs and $\gamma\gamma \rightarrow$ hadrons