Track reconstruction for the CLIC full silicon tracker

Frank Gaede°, Daniel Hynds*,
Rosa Simoniello*, Yorgos Voutsinas°

on behalf of the CLICdp collaboration

LCWS15, Whistler BC Canada
November 2-6, 2015

* CERN
° DESY

Introduction

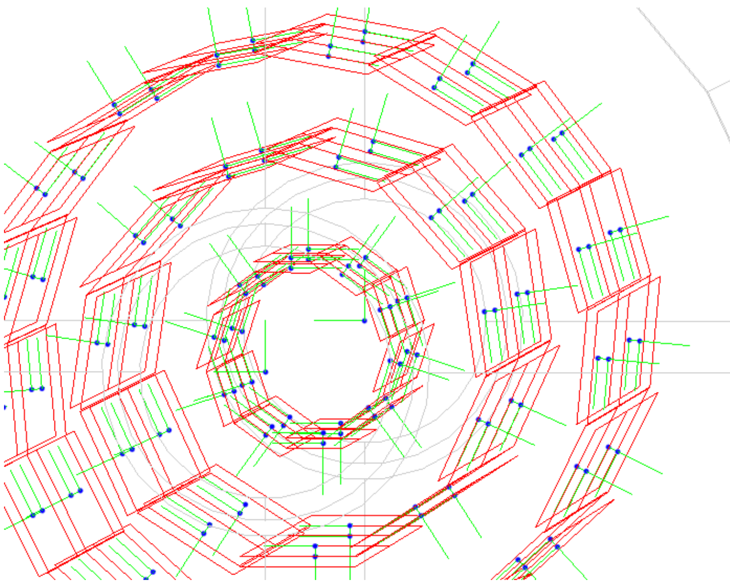
- Intense activity on the implementation of the simulation model and in the reconstruction code

N. Nikiforou's talk on Tuesday | **M. Petric's talk on Thursday**

- Developments in the full CLIC tracking chain:
  - Digitiser
  - Pattern recognition
  - Track fitting
- Tracking re-implementation → ***inside ILCSOFT***
- First results for physics events available

Motivation

- Tracking reimplementation to accommodate:
  - Kalman filter ⟷ **ILCSOFT** framework → **(DD)KalTest**
    - by Bo Li et al. possibility of reconstruction in not homogeneous B field → *to be tested for CLIC*
  - DD4hep interface ⟷
    - Use of **compact file** instead of GEAR
    - Tracking based on **surface** concept:
      - Local coordinate
        → follow particle along its trajectory
      - Material and thickness
        → compute effects of energy loss and multiple scattering

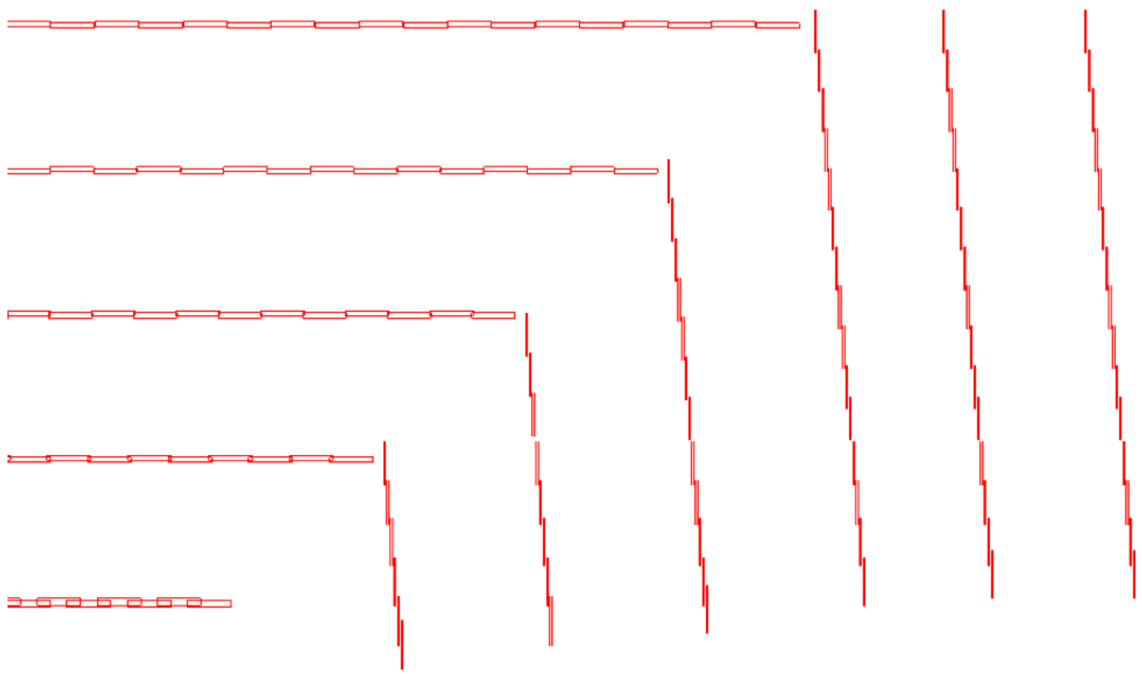


F. Gaede's talk on Tuesday

  - Optimise strategies for the CLIC full silicon detector ⟷ Subject of this talk → status report

The CLIC tracking system

- CLIC model used for these studies: **CLIC_o2_v02**
- **NOT** the most up-to-date → need of a stable version to develop tracking
- For current tracker version see **N. Nikiforou's talk on Monday**



Vertex:
- 3 barrel double layers
- 3 endcap double layers in spiral geometry

Tracker:
- 5 barrel layers
- 6 endcap disks (7 in the new model)

→ Full coverage: module overlap in Rϕ and z (under revision)

DIGITISER

Digitiser

- Digitizer → *gaussian smearing* for single point resolution
  - **Vertex** → pixels, resolution: **3µm x 3µm**
  - **Tracker** → elongated pixels, resolution: **7µm x length/√12**

	BARREL					ENDCAP	
	B1	**B2**	**B3**	**B4**	**B5**	**E in**	**E out**
Strip length [mm]	1	1	5	10	10	1	10
Resolution [µm]	300	300	1500	3000	3000	300	3000

- Plans:
  - Better handling of hit combinations
  - Take into account read out time → in presence of backgrounds it can give hit inefficiency
  - Charge sharing → move away from gaussian smearing

PATTERN RECOGNITION

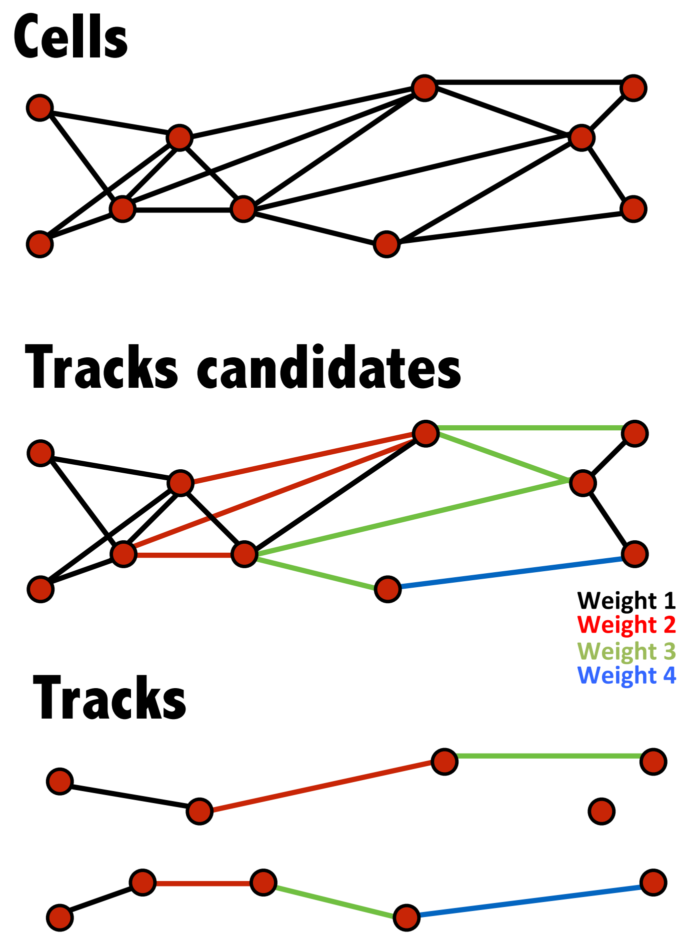
Pattern recognition for a full Si tracker

- 2 strategies under development
  - **Method 1** based on existing ILD code adapted for CLIC
  - **Method 2** based on conformal mapping
- Both included in **ILCSOFT**
  - Method 1 implemented in new processors in existing packages : MarlinTrkProcessor, ForwardTracking
  - Method 2 implemented in a new package: ConformalTracking
  - Steering file available in: ClicPerformance/examples/clicReconstruction.xml

  → *Under development* → *not yet recommended for users*
- In the end, only one will be kept for the final CLIC software chain and used for physics events production
- Both strategies based (at least partially) on *Cellular Automaton*

Cellular Automaton (CA)

- Based on the creation of cells:
  → a cell is a connection between two hits
  → every cell has a weight
- Local method → used local criteria (like angle between two cells) to connect cells
- Update cell weight: increase cell weight by 1 for each previous cell that passed criteria
- Track candidates: start from high weights and follow the connections
- Choose best tracks according to the χ^2 probability, number of tracker hits

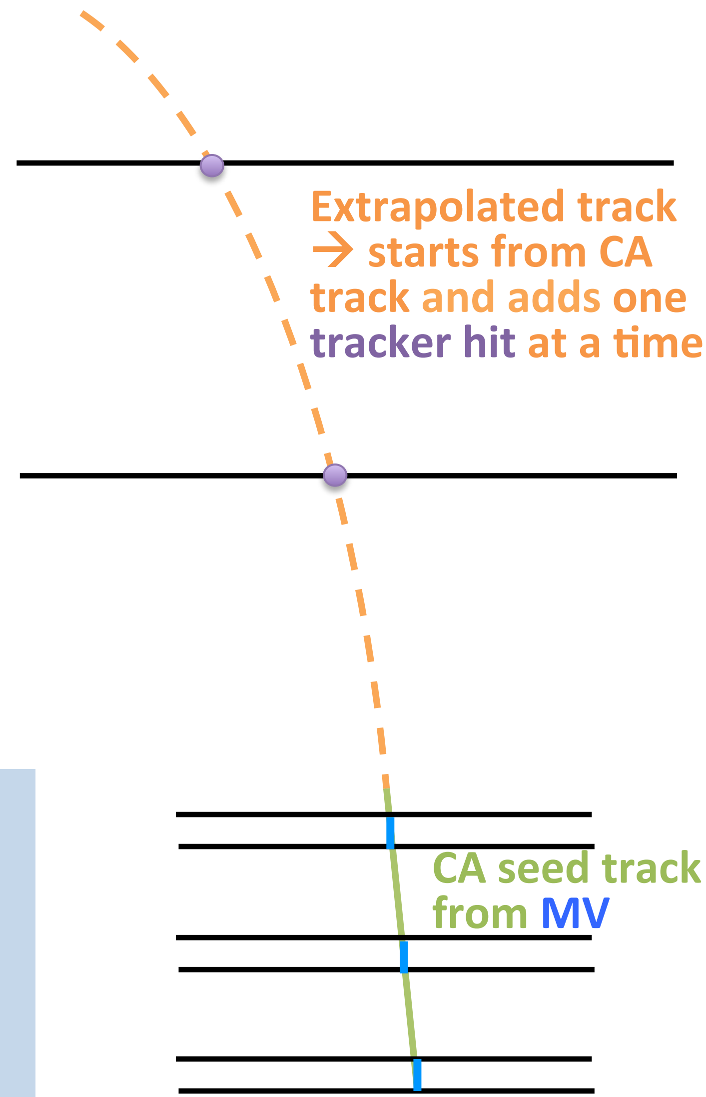


Method 1 – Overview

- From ILD software, adapt strategies for the full Si tracker case
  - Starting from existing ILD processors
  - Port to DD4hep, develop for the full Si tracker case
- Look for tracks according to *helical trajectory*
- Different strategies in different subdetectors:
  1. Tracks passing through the vertex barrel
     → *mix of CA and Kalman Filter strategies*
  2. Tracks passing through the vertex endcap
     → *pure CA strategy*
- Final combination of the tracks
  - Remove duplicate tracks
  - No need to combine track segments
    (full main tracker is considered for both cases)
  - Move to: TrackSubsetProcessor

Method 1 – part I

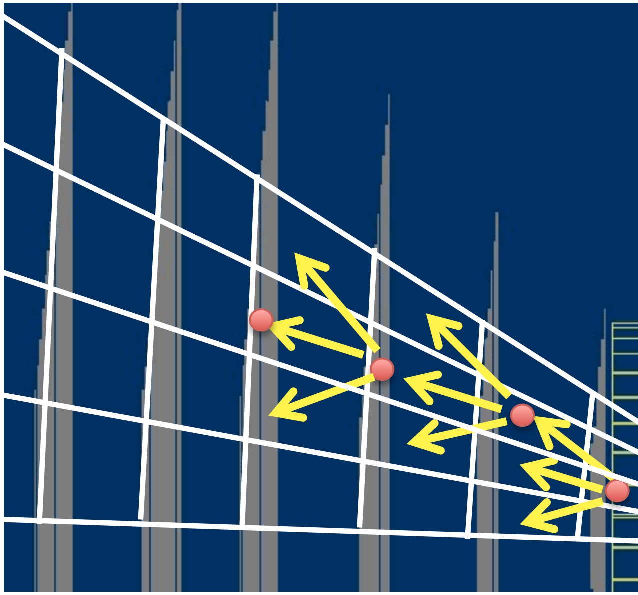
- For tracks passing through the vertex barrel
  1. Compute **mini vectors** (MV) exploit the double layer structure of the vertex barrel
  2. Run *CA on MV* → obtain vertex tracks
  3. Use vertex tracks as seed for track *extrapolation* to Inner and Outer Tracker (both in Barrel and Endcap layers)



Mini Vectors:
Create a mini vector out of 2 hits on adjacent layers (d < 5 mm) with $\delta\theta < 1°$
→ Reduce combinatorics in presence of background

Method 1 – part II

- For tracks passing through the vertex endcap
  - Based on ILD Forward Tracking, adapted to the CLIC case
    → i.e.: change in the sector definition
  - *Run CA on vertex endcap hits, inner endcap hits, outer endcap hits*
  - Sectors defined to limit combinatorics
    - Sector is a ($\Delta\phi$, $\Delta\theta$) region between two layers
    - Look for connections in neighbouring sectors
      (define step in ϕ, θ, nlayers internally to the processor)
  - Criteria defined in steering file to establish connections



- Sector and criteria definition used for both vertex detector and main tracker
  - Better to optimise sectors and criteria for each subdetector?

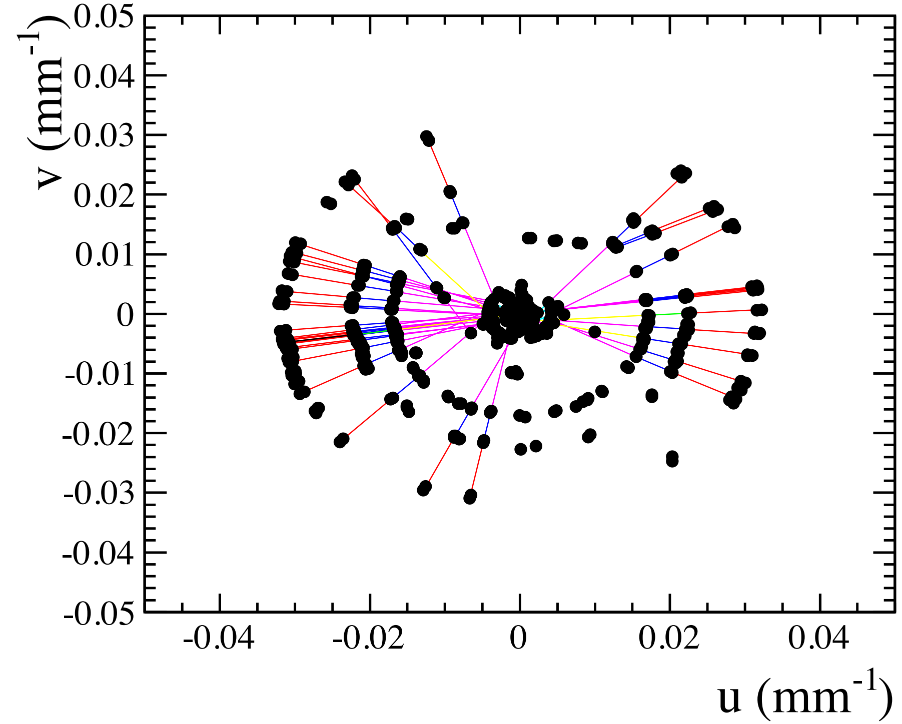
Method 2 – Overview

- Based on *conformal mapping* → coordinate transformation that preserves local angles
- Transformation of the (x,y)-plane in the (u,v)-plane

$$u = x/(x^2 + y^2)$$
$$v = y/(x^2 + y^2)$$

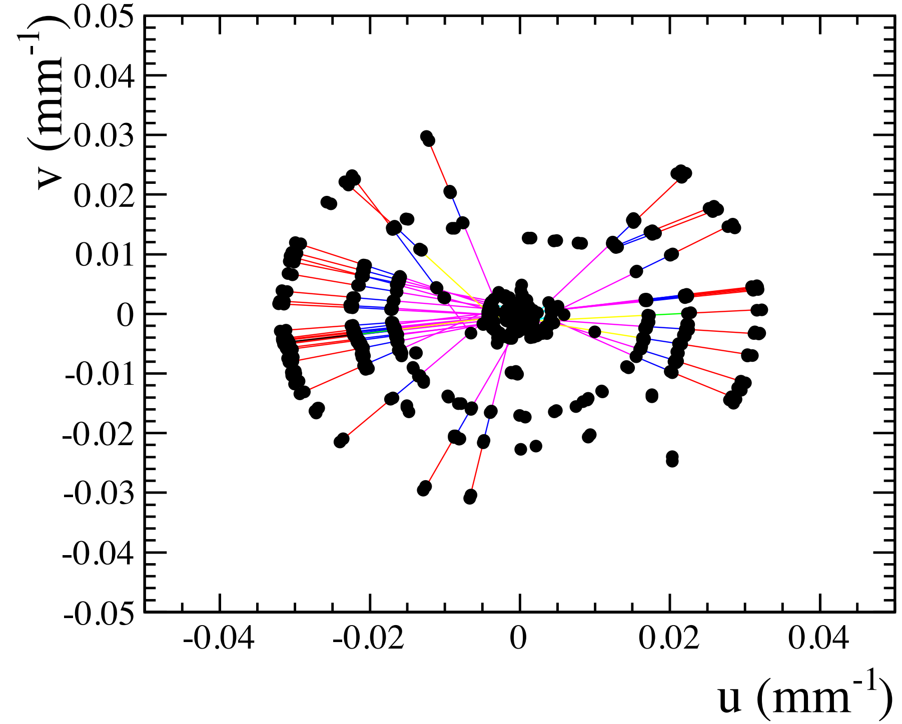
→ *Tracks are straight lines*

- *Run CA on all hit collections* → same approach in the full tracker system → no sub-track combination needed
- Method has been used by Star L3 trigger and the ALICE experiment



Method 2 - remarks

- In the (u,v)-plane vertex hits are the outermost while the tracker hits are the innermost
- High-p_T tracks point to the center
- *z information* also used, even if not visible in the projection in the (u,v)-plane → do not connect hits that are very distant in z
- *No geometry information* used (cells allowed between different layers → robust in case of missing hits)



Evaluation of performance

- Performance evaluation: track reconstruction *efficiency* and *purity*

$$\epsilon = \frac{N_{tracks}^{reconstructed}}{N_{tracks}^{reconstructable}} \qquad P = \frac{N_{tracks}^{hits}}{N_{associated\ MC}^{hits}}$$

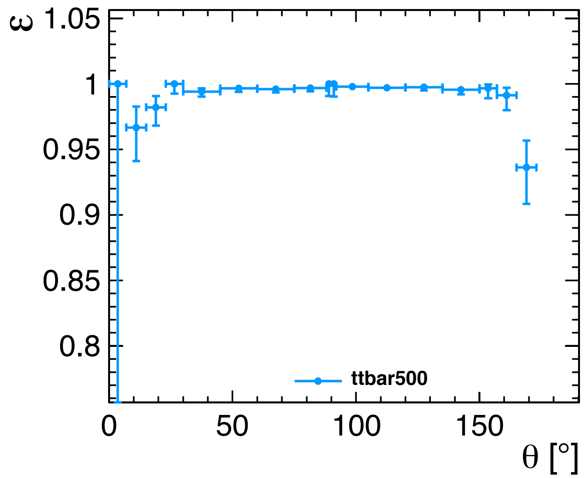
- A particle is considered *reconstructable* if:
  - It is charged and it leaves at list 4 hits in the full tracking system
  - Prompt particles: distance from the interaction point less then 10 cm
  - It is in tracker acceptance: $|\cos(\theta)| > 0.99 \rightarrow \theta > 8°$
  - It has $p_T > 100$ MeV
- Look at the performance in events without background overlay:
  - ttbar and di-jets events at 500 GeV
  - ttbar events at 3 TeV

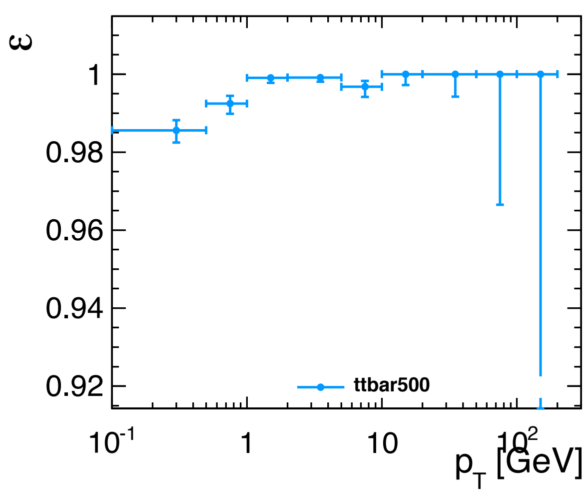
  → *Last week start production of background events in the new detector model to be overlayed with signal samples*

- *Computing time* will be also important but not considered at the moment

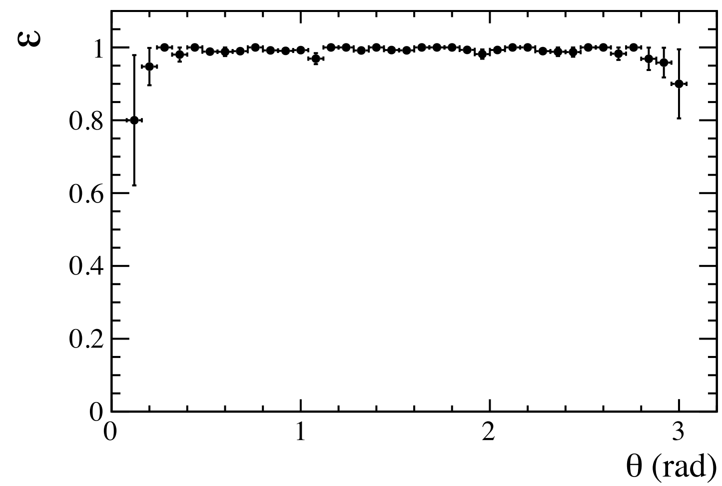
Results on ttbar events at 500 GeV

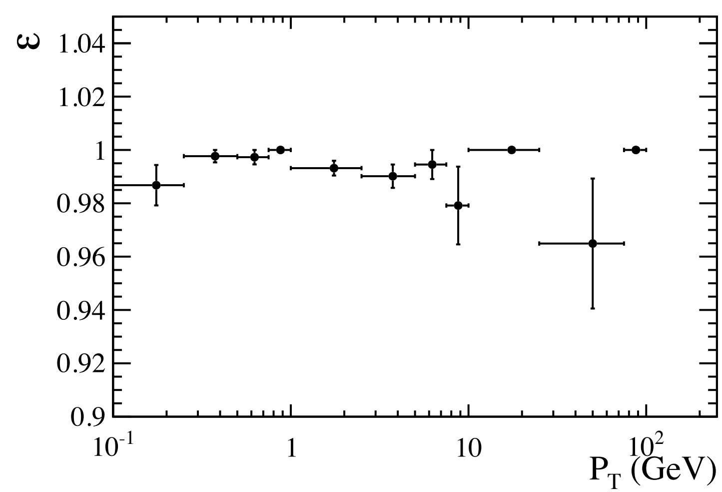
Method 1





Method 2

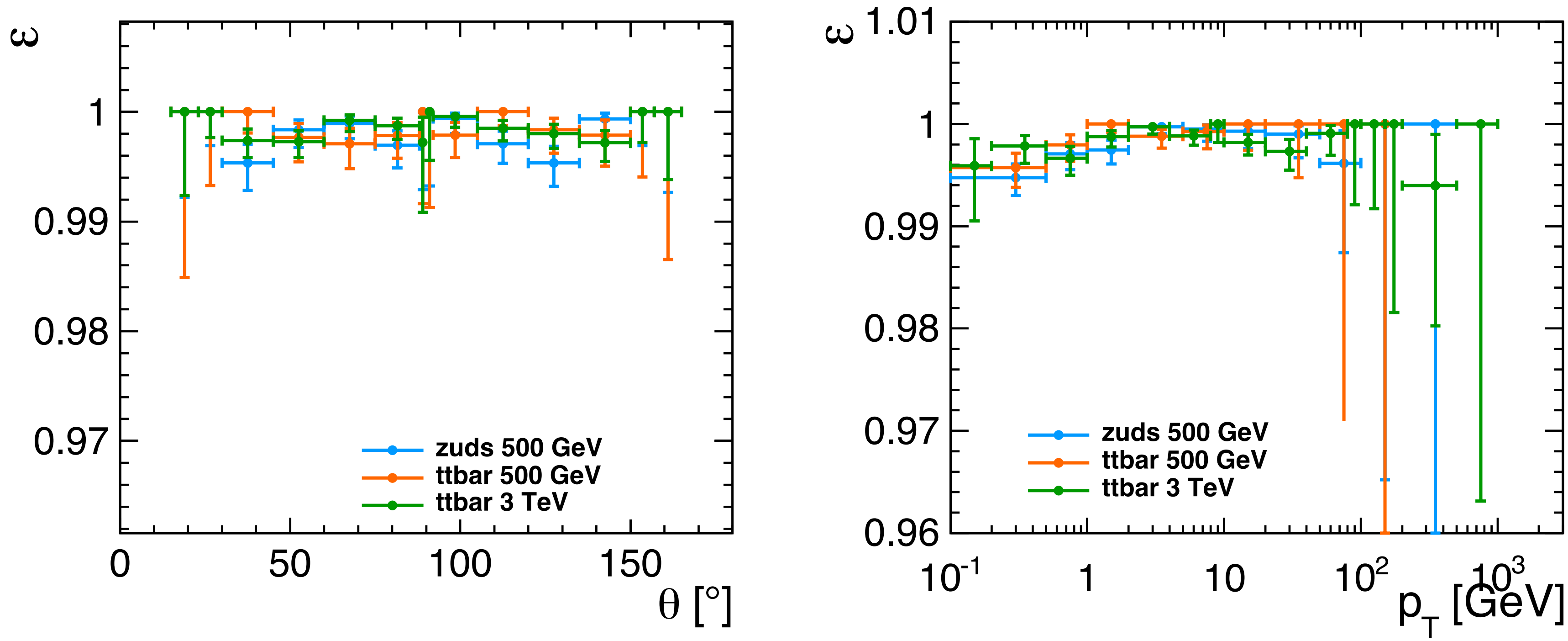




- Proof of concept for both methods ☺ – Similar results
- Next: increase statistics

Performance in different physics samples

Method 1 – CA in vertex barrel + extrapolation

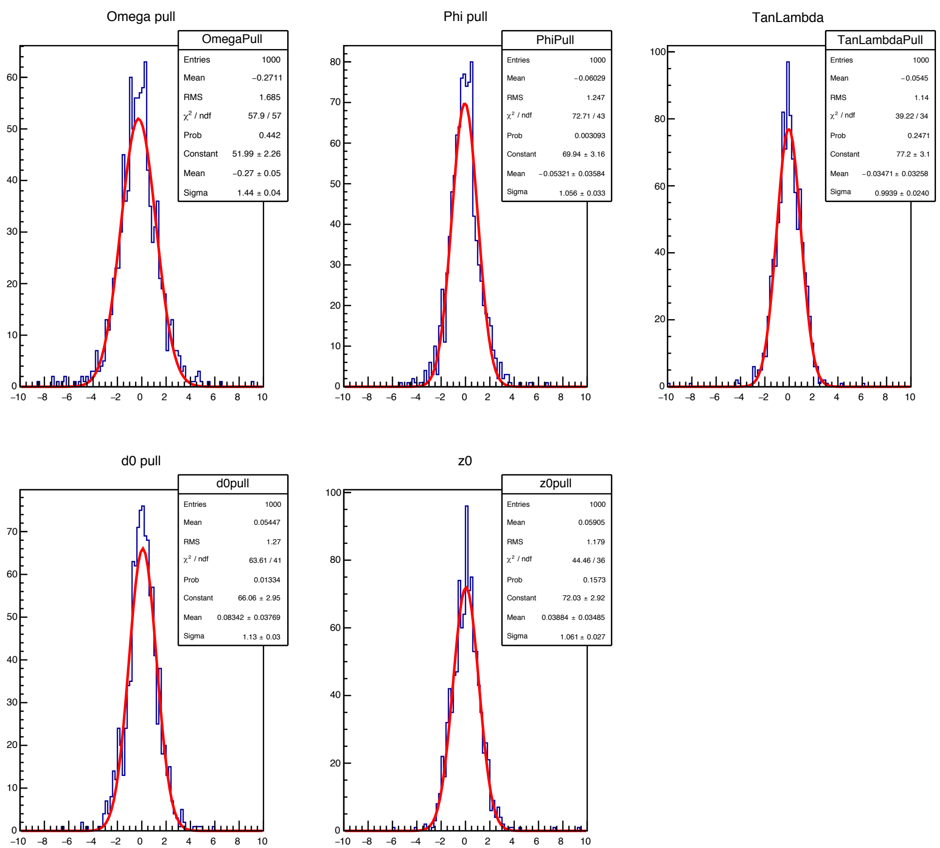


- Look only at tracks passing through the vertex barrel
- Performance looks consistent across different physics samples
- Results at 3 TeV, very preliminary → purity of the tracks not checked
- Substantial increase of processing time at 3 TeV

TRACK FIT

Track fit

- Use ILCSOFT Refit processor → fit implementation in AIDATT (based on GBL) tested – problem in the Ω pull to be understood (same for ILD)
- Decent pulls in the central region (θ = 85°) except for Ω
- In forward region, failure of fit even if track has high purity → to be investigated



Refit processor

	μ_{fit}	σ_{fit}
OmegaPull	-0.270 ± 0.047	1.440 ± 0.041
PhiPull	-0.053 ± 0.036	1.056 ± 0.033
TanLambdaPull	-0.035 ± 0.033	0.994 ± 0.024
d0pull	0.083 ± 0.038	1.130 ± 0.035
z0pull	0.039 ± 0.035	1.061 ± 0.027

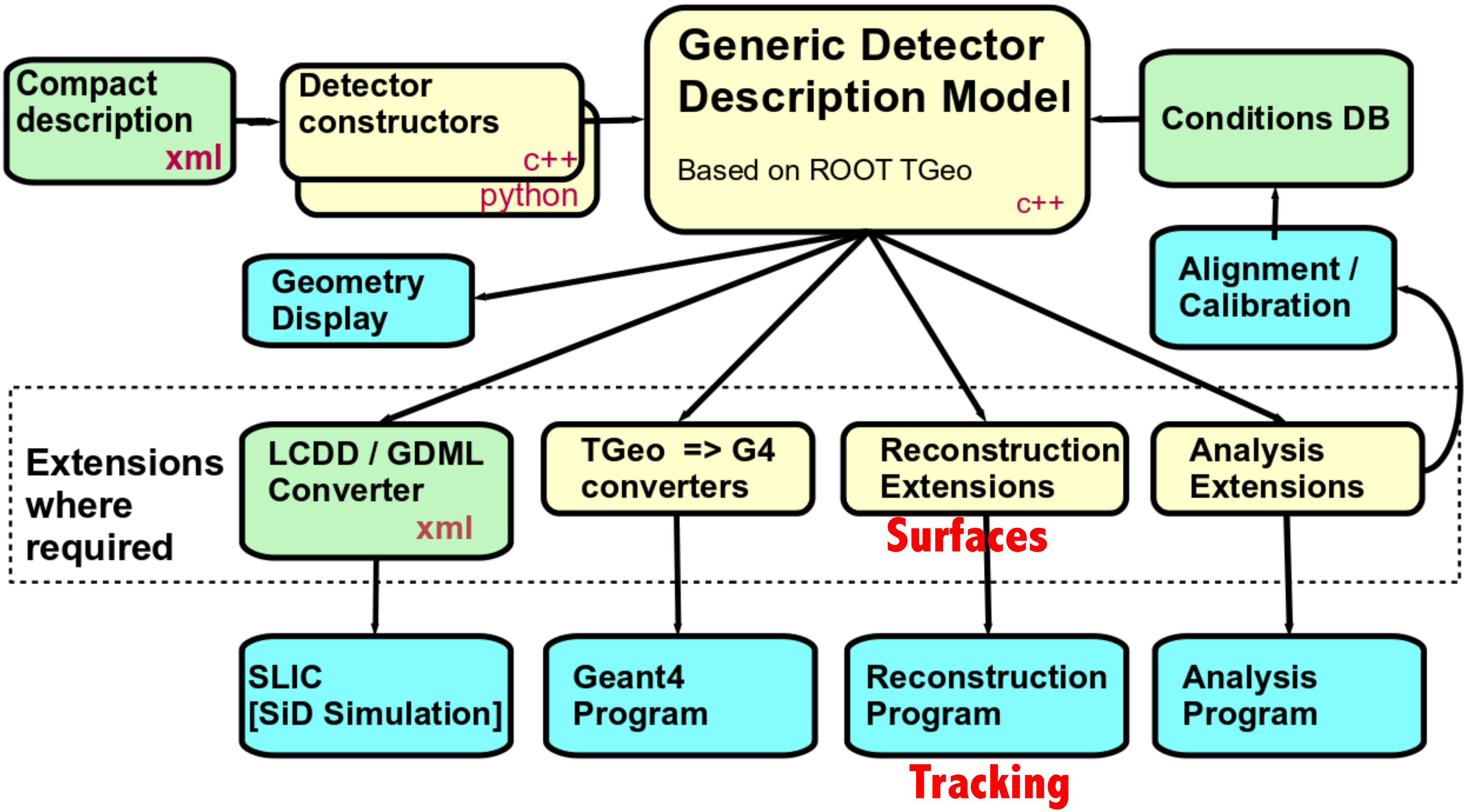
Refit in AIDATT with GBL

	μ_{fit}	σ_{fit}
OmegaPull	-0.379 ± 0.048	1.450 ± 0.038
PhiPull	-0.151 ± 0.040	1.220 ± 0.033
TanLambdaPull	-0.072 ± 0.037	1.135 ± 0.031
d0pull	0.105 ± 0.039	1.180 ± 0.031
z0pull	0.088 ± 0.038	1.121 ± 0.029

Conclusions and plans

- Significant progress in the pattern recognition
  - 2 methods under development for CLIC
- Work on-going but already encouraging performance
- Plans for pattern recognition:
  - Studies with background overlayed
  - Increase statistics → run on the grid
  - Studies with non-homogeneous magnetic field
- Other plans (in parallel to pattern recognition developments):
  - Improve realism in the digitiser
  - Improve track fit

BACK-UP


Compact description
xml
Detector constructors
c++
python
Generic Detector Description Model
Based on ROOT TGeo
c++
Conditions DB
Geometry Display
Alignment / Calibration
Extensions where required
LCDD / GDML Converter
xml
TGeo => G4 converters
Reconstruction Extensions
Analysis Extensions
Surfaces
SLIC [SiD Simulation]
Geant4 Program
Reconstruction Program
Analysis Program
Tracking

Perfomance with SiD tracking

C. Grefe's thesis