Tracking in ILD

9/10/2014
LCWS14, Belgrade, Serbia
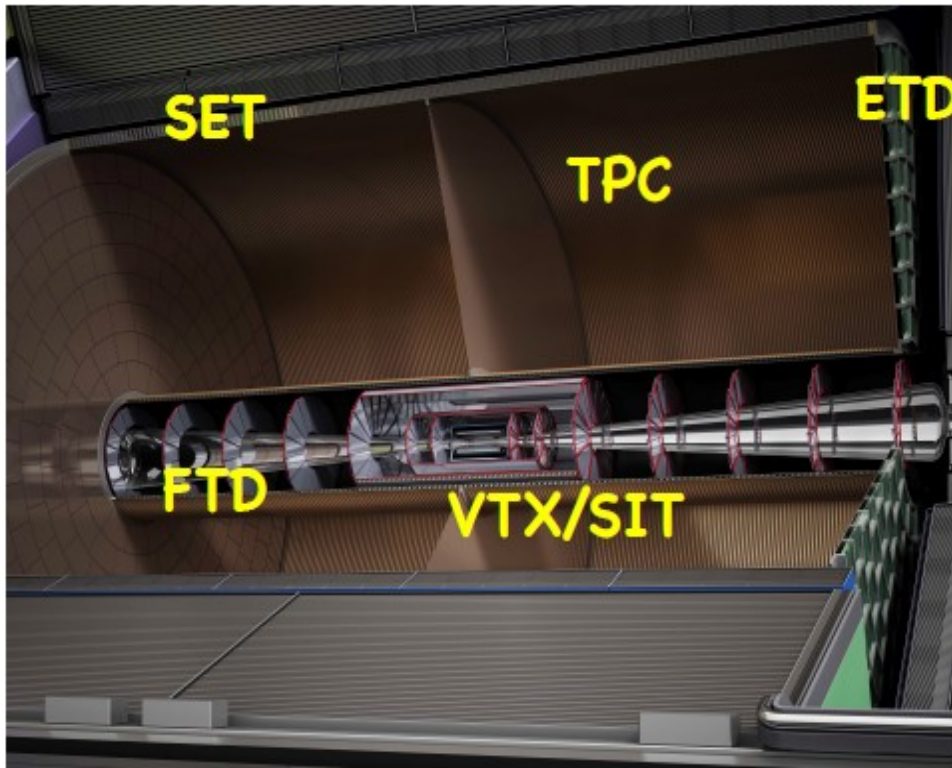
<u>Y. Voutsinas</u>
on behalf of the ILD tracking group

georgios.voutsinas@desy.de

Outline

- Overview of the ILD tracking & performance

- Recent progress in tracking

- Mini – vector VXD tracking

- Outlook – VXD studies

The ILD Tracking System

- Requirements

  - Momentum resolution

  $$\sigma_{1/p_T} \approx 2 \times 10^{-5} \text{ GeV}^{-1}$$

  - Impact parameter resolution

  $$\sigma_{r\phi} = 5 \ \mu\text{m} \oplus \frac{10}{p(\text{GeV}) \sin^{3/2}\theta} \ \mu\text{m}$$

- ILD Tracking system in simulation

  - Has a sophisticated G4 simulation model

  - Support structures, cabling, cooling infrastructure have been described in detail

| Detector	Point Resolution		
VTX	$\sigma_{r\phi,z}$	=	2.8μm (layer 1)
	$\sigma_{r\phi,z}$	=	6.0μm (layer 2)
	$\sigma_{r\phi,z}$	=	4.0μm (layers 3-6)
SIT	σ_{α_z}	=	7.0μm
	α_z	=	$\pm7.0°$ (angle with z-axis)
SET	σ_{α_z}	=	7.0μm
	α_z	=	$\pm7.0°$ (angle with z-axis)
FTD *Pixel*	σ_r	=	3.0μm first two discs
	$\sigma_{r\perp}$	=	3.0μm
FTD *Strip*	σ_{α_r}	=	7.0μm
	α_r	=	$\pm5.0°$ (angle with radial direction)
TPC	$\sigma_{r\phi}^2$	=	$(50^2 + 900^2 \sin^2\phi + ((25^2/22) \times (4T/B)^2 \sin\theta)(z/\text{cm})) \ \mu\text{m}^2$
	σ_z^2	=	$(400^2 + 80^2 \times (z/\text{cm})) \ \mu\text{m}^2$
	where ϕ and θ are the azimuthal and polar angle of the track direction		

3

Overview of the ILD Tracking Software

- Use ILCSoft core packages Marlin, LCIO, Gear
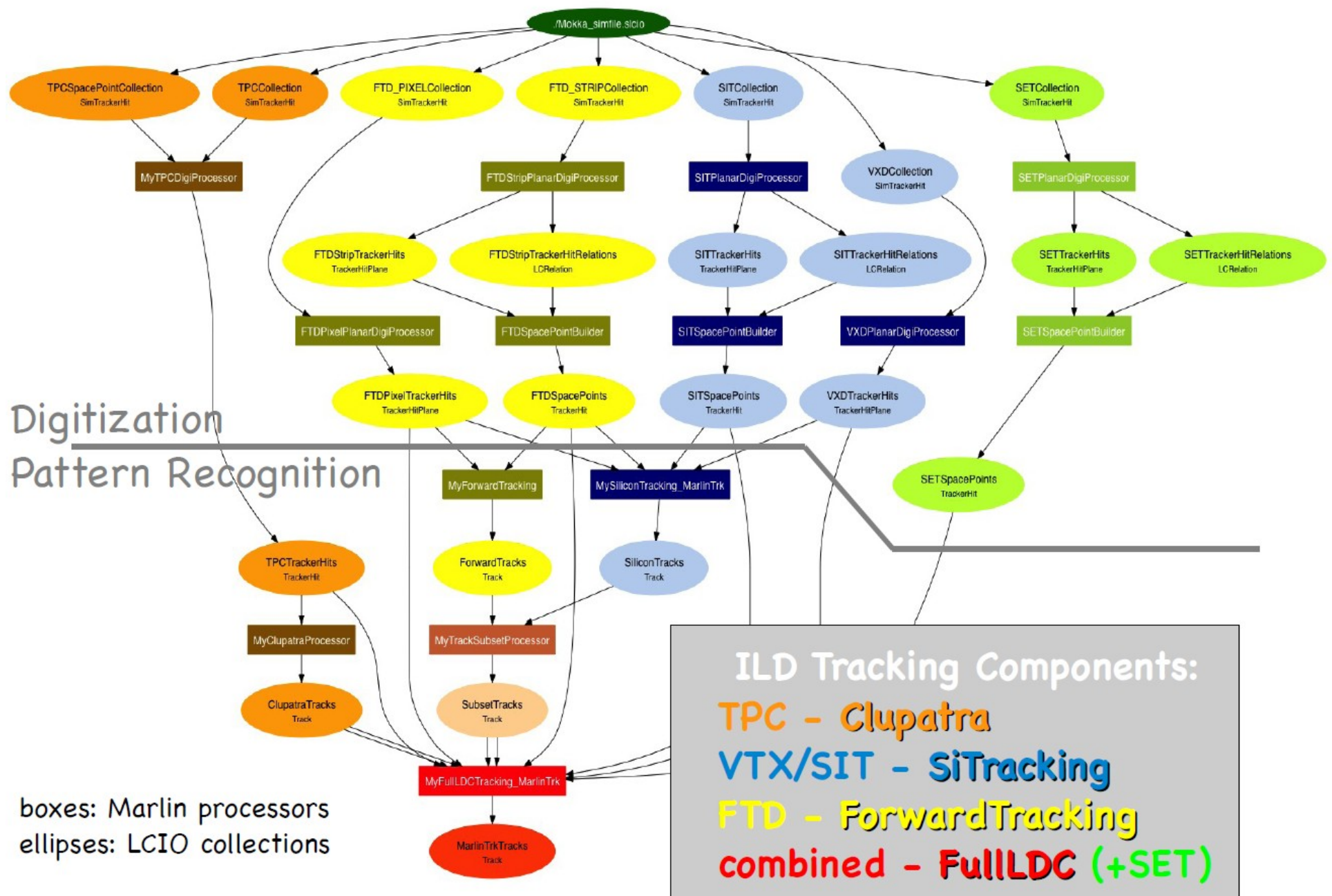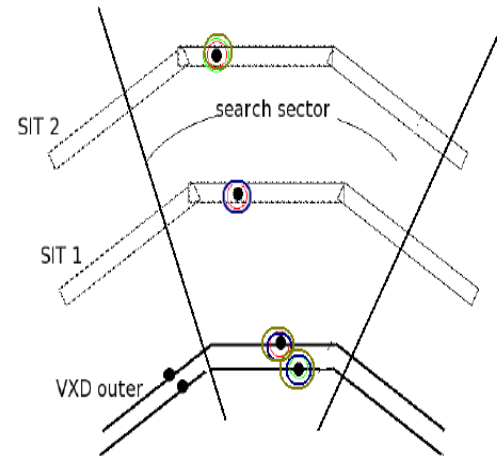
Pattern Recognition in ILD

- **Clupatra processor**
  - Form seeds using Nearest Neighbours hit clustering
  - Propagate seeds both inwards & outwards using Kalman fitter
  - Associate best matching hit
  - Update track state
  - So on...
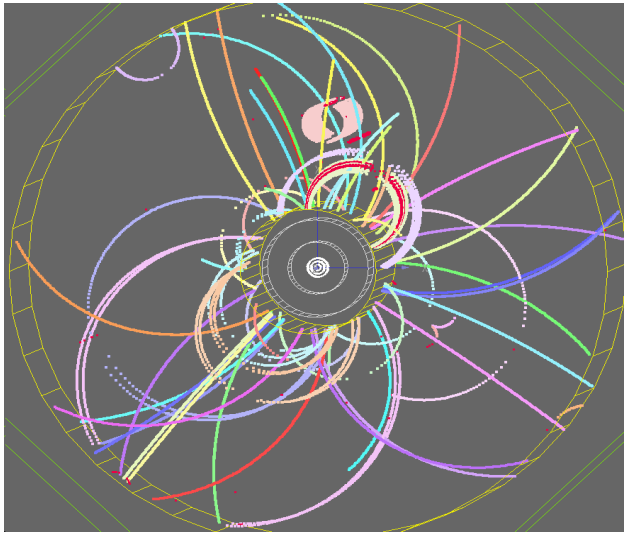
- **SiliconTracking**
  - Divide VXD – SIT into angular sectors
  - brute force triplet search in phi sectors based on a set of seed-layer-triplets
  - Fit a helix to the seed triplets
  - Follow the seed inwards – attach hits according to the distance from the helix
  - Refit with Kalman fitting
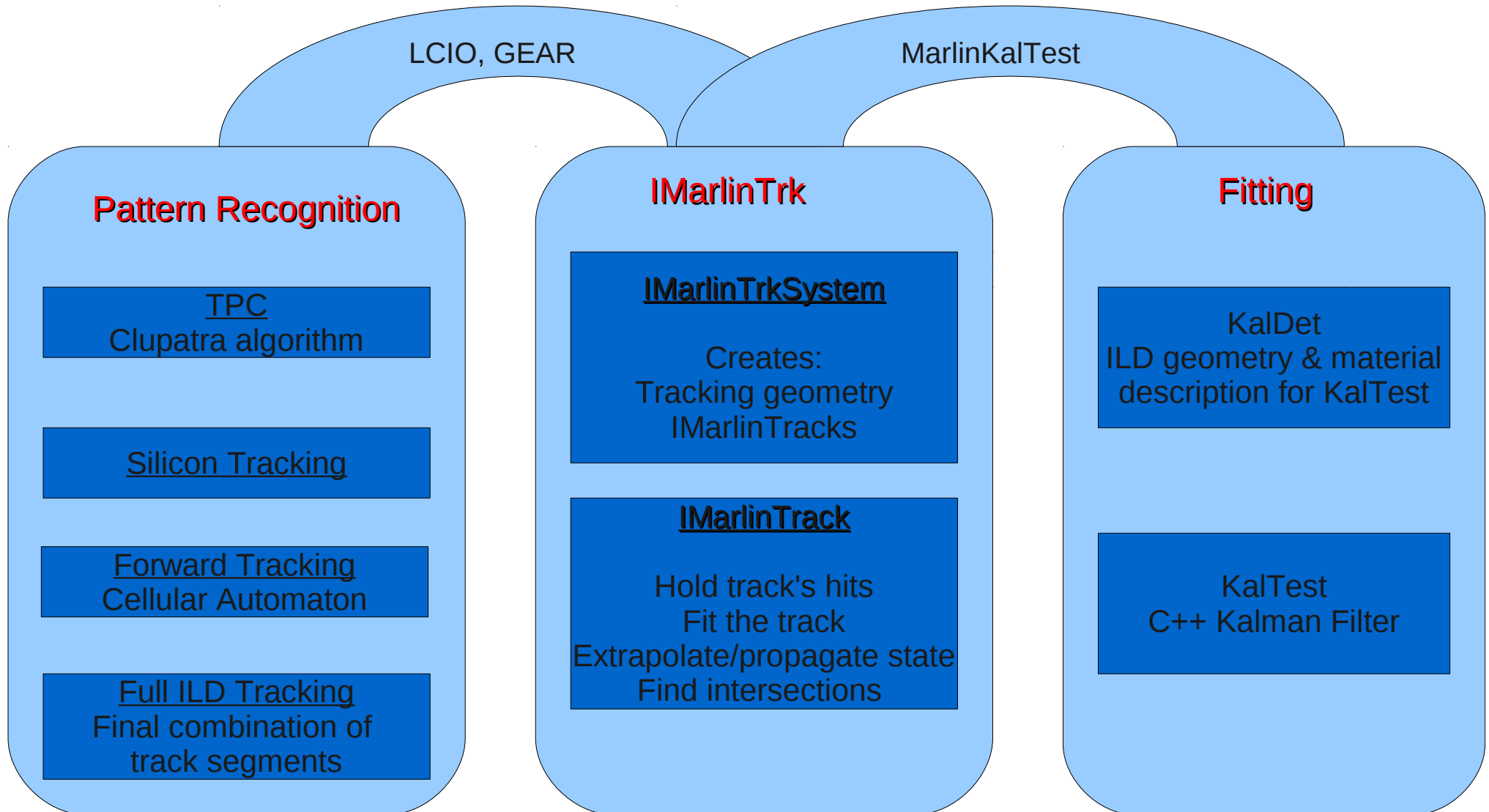

- **Forward Tracking**
  - Standalone tracking algorithm at FTD
  - Pattern recognition: Cellular automaton
  - Fitting: Kalman filter
  - Ambiguities resolution: Hopfield NN
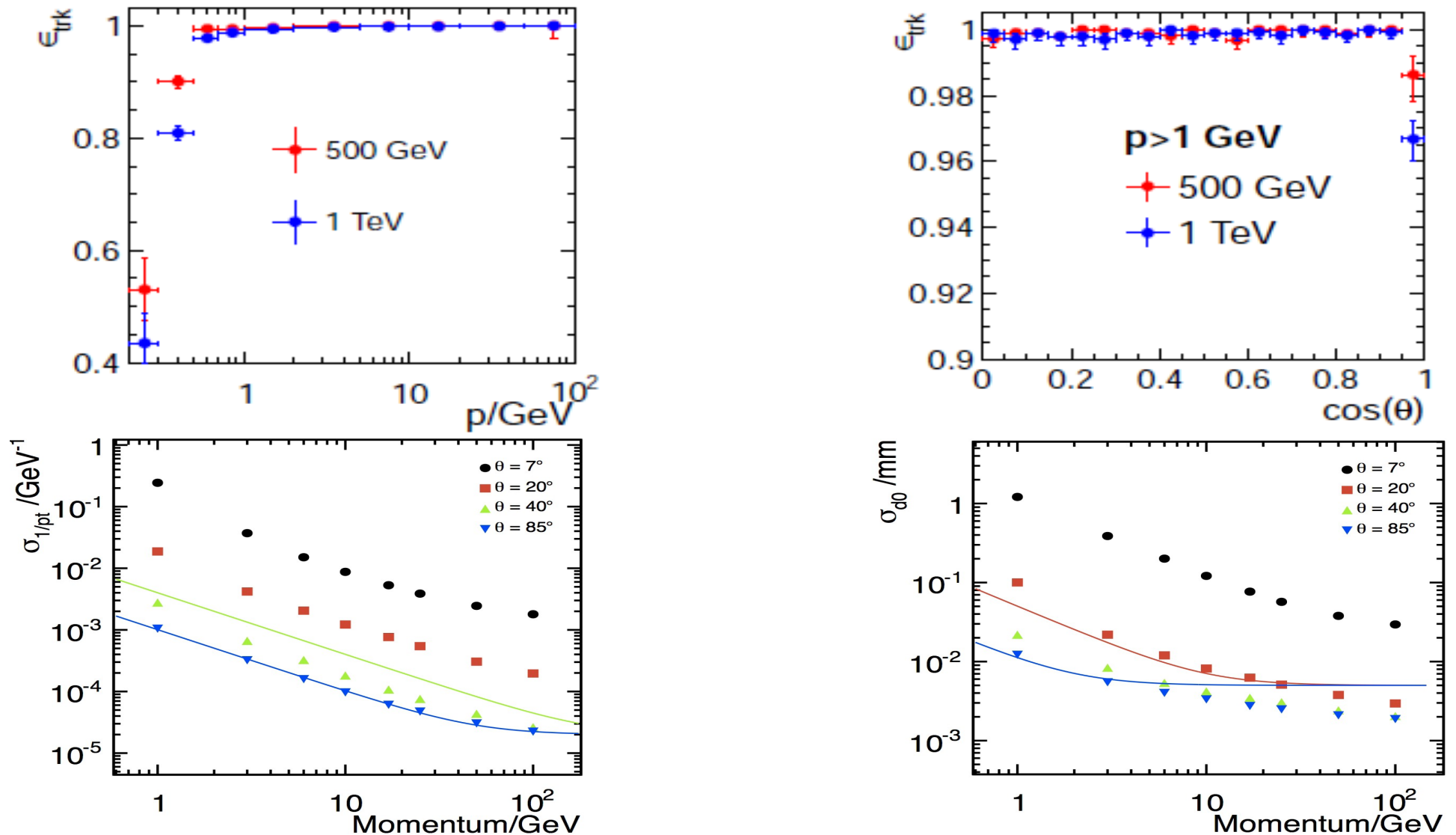
- **FullLDCTracking**
  - Combines track from TPC – FTD – Silicon tracking
  - Based on track parameter compatibility
  - Adding spurious leftover hits
  - Final track fit

5

IMarlinTrk Interface for Tracking

- Common API for developing tracking code

- Provides loose coupling between track finding & track fitting

LCIO, GEAR

MarlinKalTest

Pattern Recognition

IMarlinTrk

Fitting

TPC
Clupatra algorithm

Silicon Tracking

Forward Tracking
Cellular Automaton

Full ILD Tracking
Final combination of
track segments

IMarlinTrkSystem

Creates:
Tracking geometry
IMarlinTracks

IMarlinTrack

Hold track's hits
Fit the track
Extrapolate/propagate state
Find intersections

KalDet
ILD geometry & material
description for KalTest

KalTest
C++ Kalman Filter

Performance

Plots from DBD – ttbar sample, beam bkg included
* ~ 99.7 % eff., P ≥ 1 GeV, ≥ 99.8 %, cos(θ) < 0.95
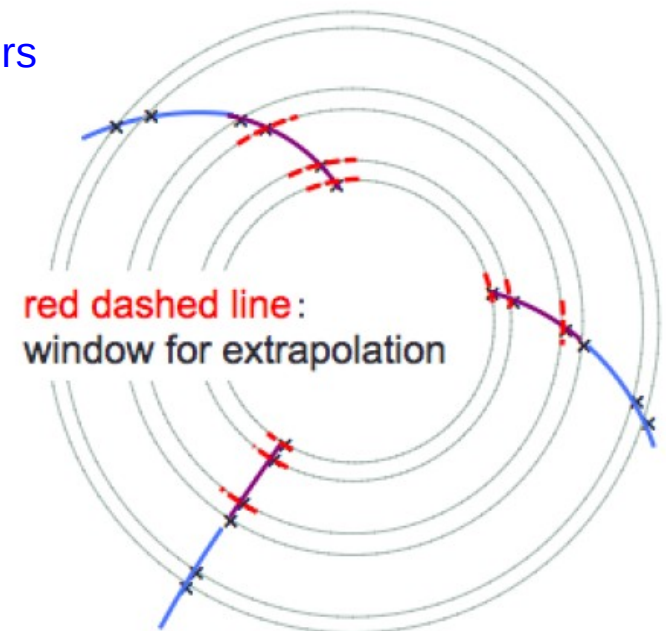* Achieve ILD goals in resolution

Progress in Tracking

- Pattern recognition in Si detectors

  - FPCCD tracking (Tatsuya Mori, Tohoku University)

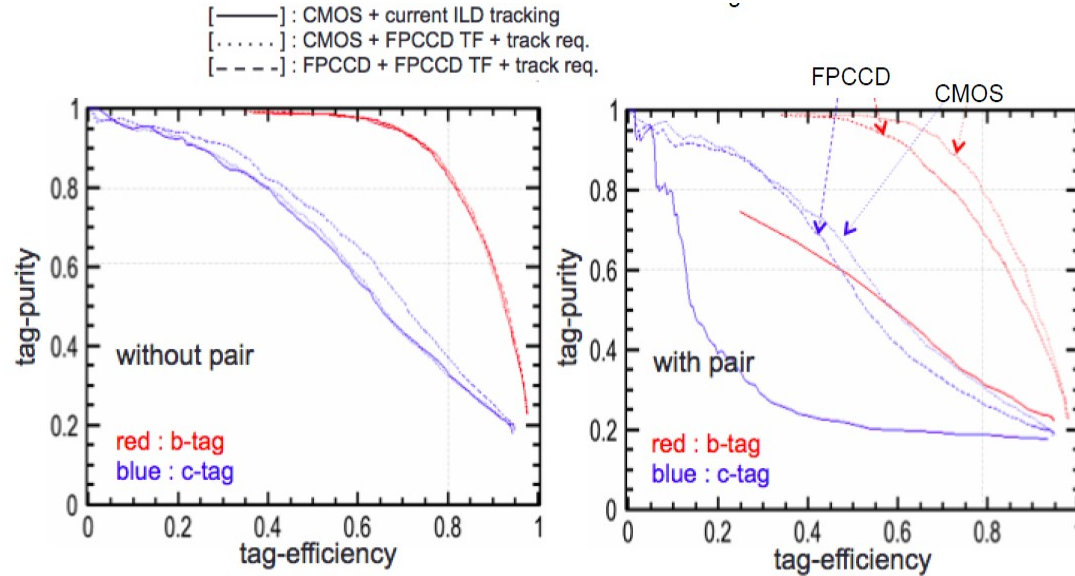 - Cellular automaton mini – vector tracking (DESY)

FPCCD Tracking

- Following the std silicon tracking philosophy

- Has improved the following crucial steps:

- Seed formation:

  - Angular sectors: φ width enough to generate seeds with minimum P_T 180 MeV

- Track extrapolation

  - Extrapolate seeds using Kalman filter instead of simple helix fit

    - More efficient for low P_T tracks, takes into account MSC

  - φ width for extrapolation flexible, defined by track parameters

    - It catches true hits and avoids most of bkg hits

red dashed line :
window for extrapolation

Striking improvement in Silicon tracking performance
in terms of efficiency, ghost rate, time
in the presence of pair bkg compared to std algorithm

FPCCD Tracking Flavour Tagging Performance

[———] : CMOS + current ILD tracking
[……] : CMOS + FPCCD TF + track req.
[– – –] : FPCCD + FPCCD TF + track req.

FPCCD   CMOS

without pair

red : b-tag
blue : c-tag

with pair

red : b-tag
blue : c-tag

- We examine 2k evts of Z → bb, cc, uds
  - One bunch train of pair bkg overlayed
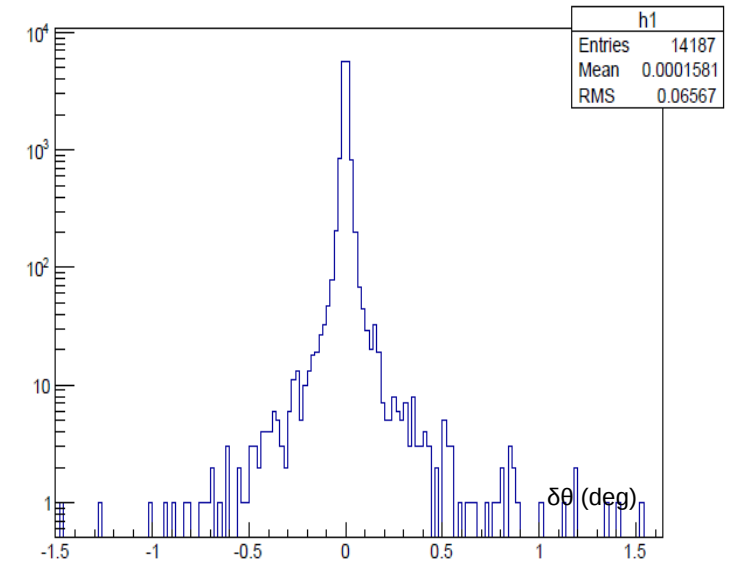- LCFIPlus trained with 14k evts
  - No pair bkg

Detector	Algorithm	Pairs	b – tag purity (%) (efficiency 80%)	c – tag purity (%) (efficiency 60%)
DBD	STD	No	82.8	56.4
DBD	STD	Yes	30.4	20.0
DBD	FPCCD	Yes	77.6	49.4
FPCCD	FPCCD	Yes	67.8	41.6

- Pair bkg degrades significantly the flavour tagging performance
- FPCCD track finder substantially improves the flavour tagging performance, compared to std silicon tracking algorithm

Results from J. Strube talk at AWLC14

Mini – Vector CA VXD Tracking

- Exploits the double sided ladder structure of VXD

- Up to now, has been applied in various CMOS VXD configurations (see table)

- Mini – vector formation

  1) Hits in adjacent layers (dist 2mm) with max distance 5mm

  2) Or $\delta\theta$ between hits in adjacent layers (cut can go up to 0.1^0)

- Divide VXD into θ, φ sectors

  ➢ Try to connect mini – vectors in neighbouring sectors using a cellular automaton algorithm (next slide)

ttbar, $\delta\theta$ of hits belonging to a MV based on MC info

layer	CMOS 1		CMOS 2		CMOS 3	
	σ_{sp} (μm)	σ_{time} (μs)	σ_{sp} (μm)	σ_{time} (μs)	σ_{sp} (μm)	σ_{time} (μs)
L1 / L2	3 / 6	50 / 2	5 / 5	8 / 8	3 / 5	50 / 8
L3 / L4	4 / 10	100 / 7	5 / 5	16 / 16	5 / 5	16 / 16
L5 / L6	4 / 10	100 / 7	5 / 5	16 / 16	5 / 5	16 / 16

11

Cellular Automaton Tools

- Core tools are already there for the FTD tracking

- Very flexible

  - Appealing to be used for pattern recognition in other detectors

  - See R. Glattauer Diploma thesis

http://www.hephy.at/fileadmin/user_upload/Publikationen/DiplomaThesis.pdf

MarlinTrkProc

MiniVector CA

KiTrack

Basic algos
Abstract classes
(hits, tracks, ...)
CA criteria

KiTrackMarlin

lcio / Marlin
implementation

- VXD & mini – vectors related definitions of KiTrack abstract classes have been created in KiTrackMarlin

- Set of criteria for mini – vector connections have been defined in KiTrack
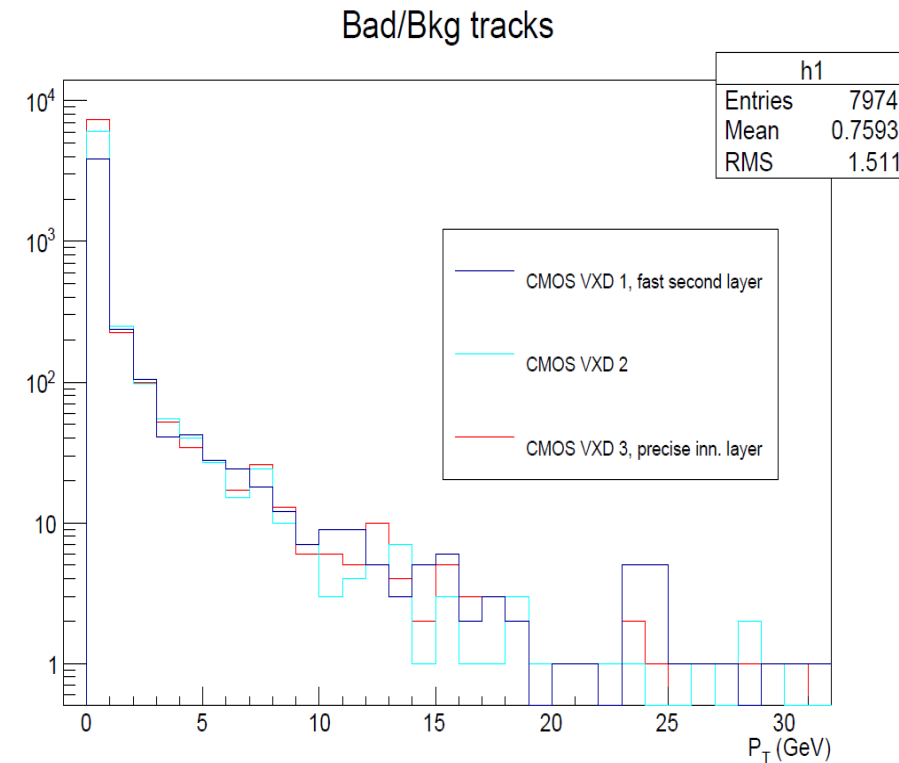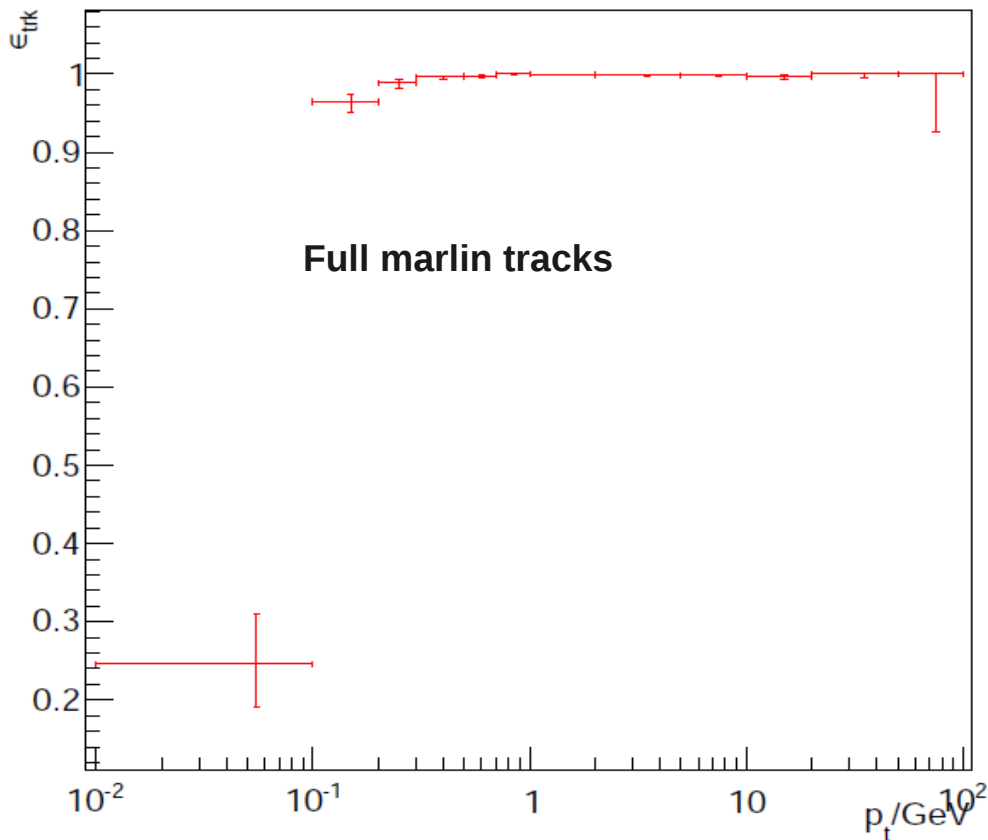
- Minor modifications in core tools
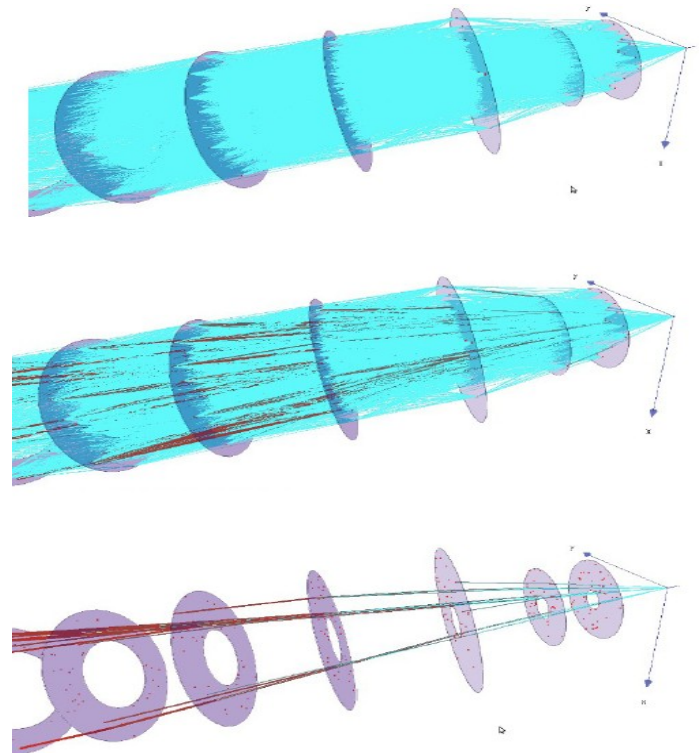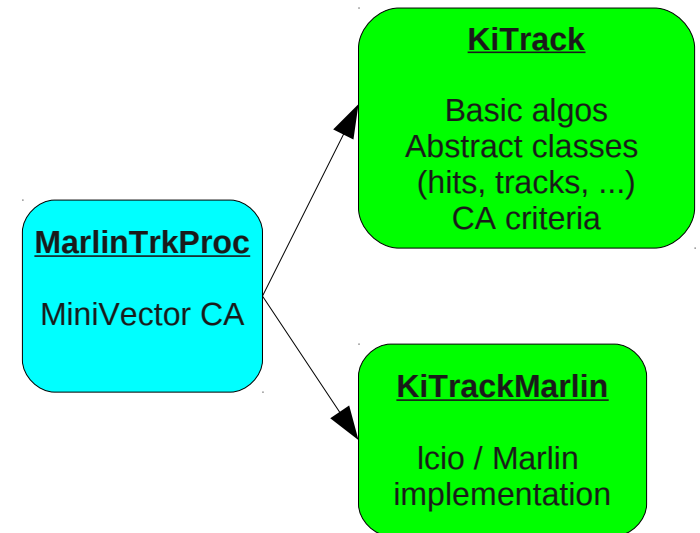
Performance

Sample: ttbar, $\sqrt{s}$ = 500 GeV, pair bkg overlayed

- Examined track sample

  - All charged tracks inside the geometrical acceptance of the VXD

  - 75% purity, ≥ 4 Silicon hits : track found

- "Bad" tracks

  - all tracks which does not correspond to a found MC particle

  - Could be pair bkg particles or combinatorics or misreconstructed tracks

	CMOS 1	CMOS 2	CMOS 3
Hits (x10³)	~120	~30	~100
Bad trks / evt	~56	~84	~100
Time / evt (s)	~15	~10	~100

Full marlin tracks

Bad/Bkg tracks

	h1	
Entries		7974
Mean		0.7593
RMS		1.511

CMOS VXD 1, fast second layer

CMOS VXD 2

CMOS VXD 3, precise inn. layer

Propagating to SIT

- VXD → mini – vectors, SIT → spacepoints

  - Complicate to integrate inside the same cellular automaton

  - Reconstruct track to the VXD or TPC

  - Use of IMarlinTrk tools to propagate and associate hits to the SIT
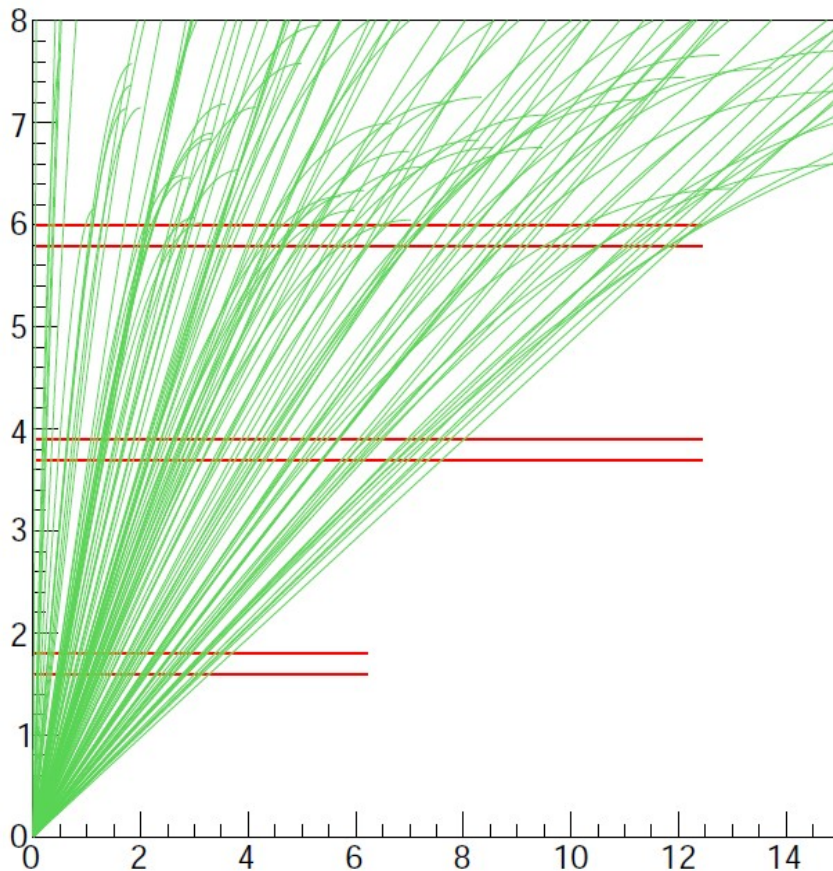
- Might be useful for rejection of "bad" tracks
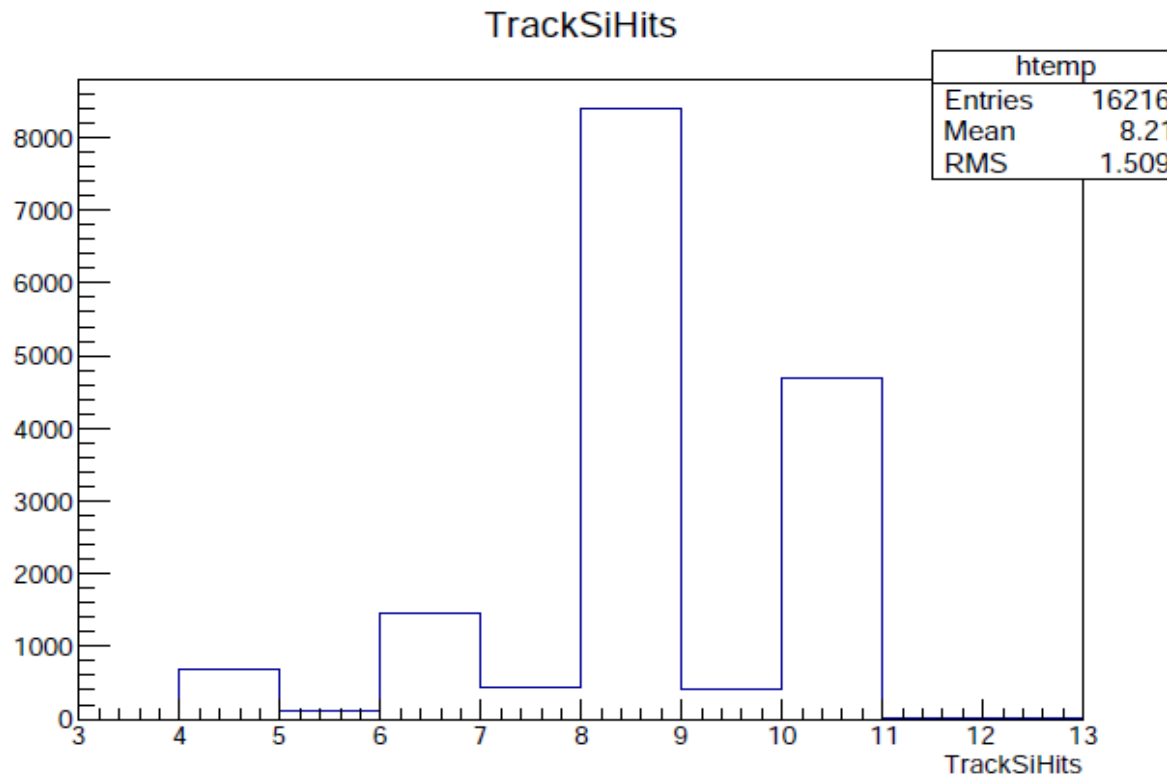
What about "bad" tracks

- Can we be efficient in low P_T tracks && suppress "bad" tracks in the same time?

- Where do they come from?

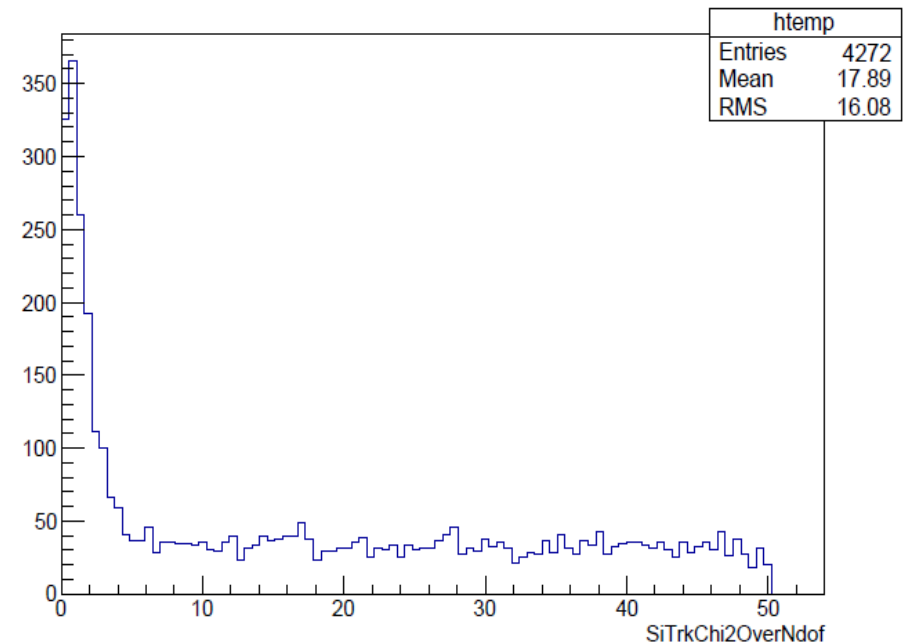- Pair bkg file

- Events: 60

- ~ 30 / evt with χ^2 / ndf < 10

Pair background in the VXD for 10 BX

From Armin Taenzer

SiTrkChi2OverNdof

| htemp	
Entries	4272
Mean	17.89
RMS	16.08

Pair bkg tracks, χ^2/ndf

15

Outlook

- 2 standalone Si tracking algorithms that show promising performance in the presence of the bkg

  ➢ Flavour tagging with mini – vector tracking need to be validated

- Select the most satisfactory one and proceed to VXD studies

- Questions

  ➢ Do we really need $\geq$ 3µm IP resolution?

  ➢ What is the optimal time resolution?

    ➢ Assume DBD – like model, where 2nd layer is optimised for time resolution

    ➢ Scan various values of time resolution of 2nd layer wrt to some indicative quantity (c-tagging?)

    ➢ See if we can reach a plateau, when further improvement gives marginal effect to performance

Backup

Pattern Recognition – Silicon

- Existing algoritms re-implemented to use MarlinTrk instead of old f77 fitter

  - Divide VXD – SIT into angular sectors

  - brute force triplet search in stereo angle sectors based on a set of seed-layer-triplets

  - Fit a helix to the seed triplets

  - Follow the seed helix inwards – attach hits according to the distance from the helix

  - Sorting – selection of track candidates

  - Attach remaining hits in the intermediate layers

  - Refit with Kalman fitter

- Doesn't appear to have optimal performance under realistic conditions

  - Machine induced bkg induces a huge combinatorial bkg

- Search for a new algorithm for pattern recognition

  - A cellular automaton algorithm based on mini-vectors (2-hit segments) seems to give promising results
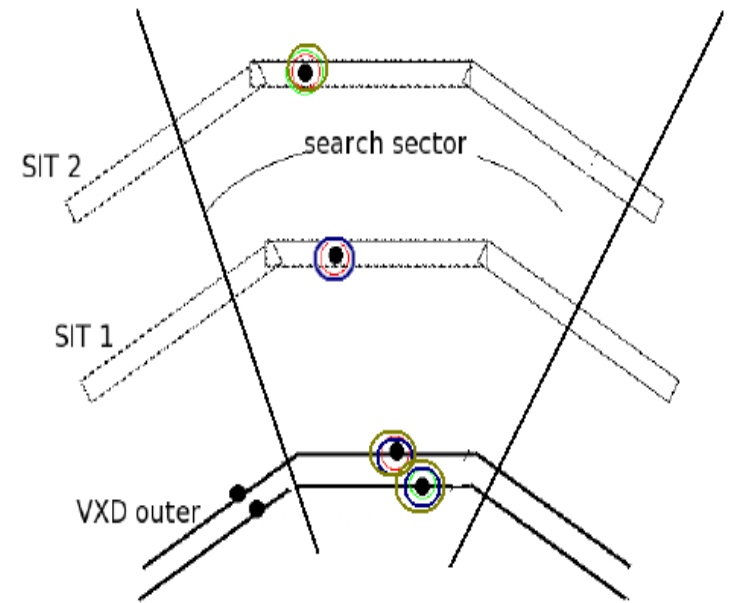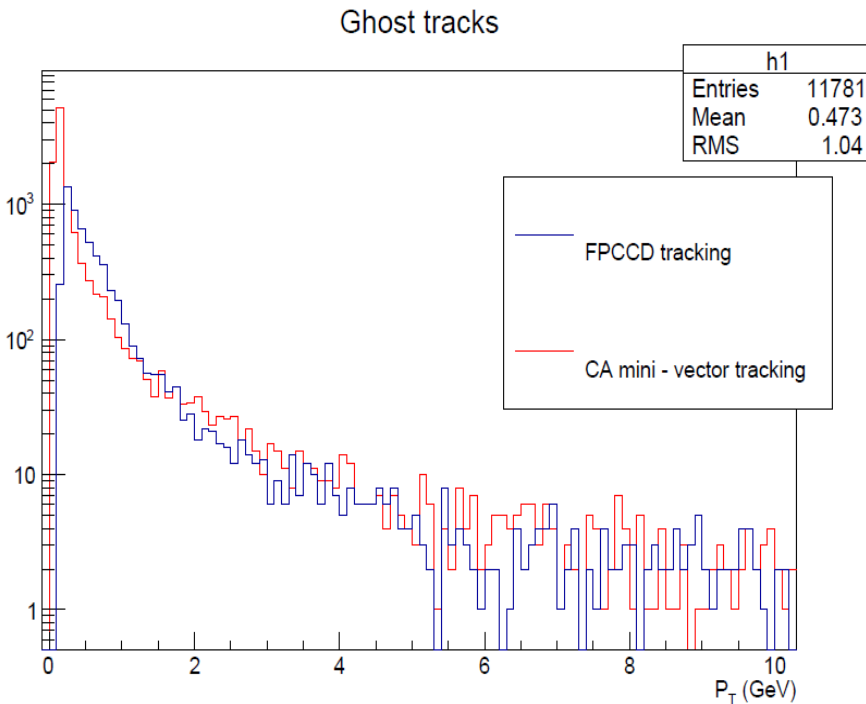
Pattern Recognition – TPC

⭐ **Pattern recognition activities in DESY: TPC & Silicon detectors**

- Clupatra processor – pat. rec. in TPC

  - Form seeds using Nearest Neighbours hit clustering

  - Propagate seeds using Kalman fitter

  - Associate best matching hit

  - Update track state

  - So on...

- Excellent track finding efficiency and CPU performance in realistic conditions

- Can be used for test beam data analysis as well
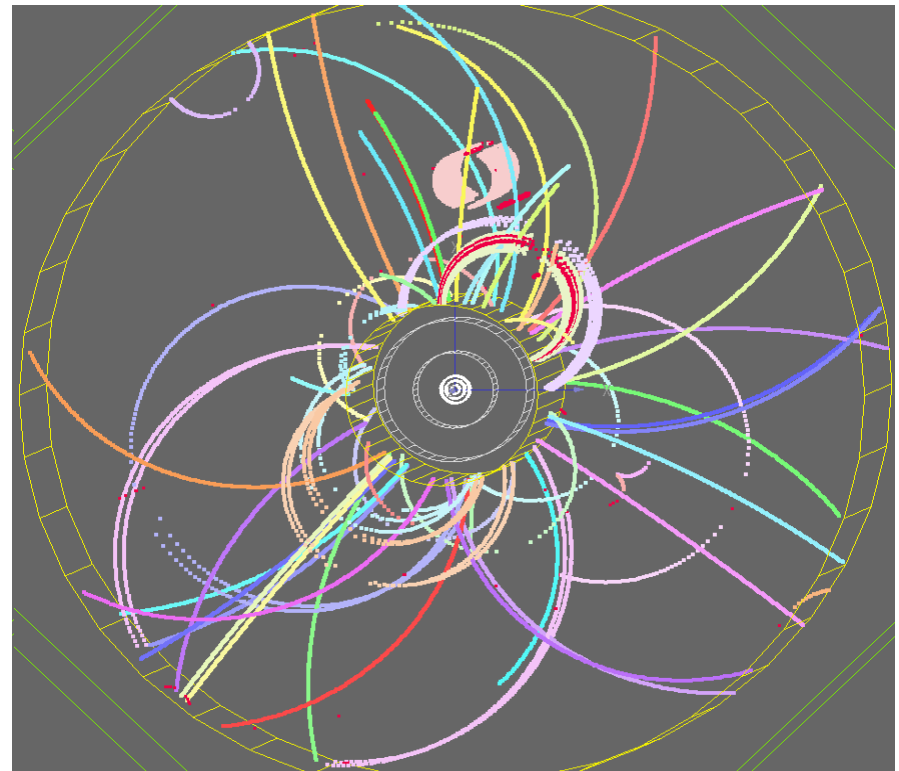
Comparison with FPCCD Tracking

- Examined track sample

  - All charged tracks inside the geometrical acceptance of the VXD

- Definition of found track

  - 75% purity, ≥ 4 hits

- "Bad" tracks

  - all tracks which does not correspond to a found MC particle

  - Could be pair bkg particles or combinatorics or misreconstructed tracks

- Time / evt

  - FPCCD: ~ 75 s

  - CA: ~ 15 s

Ghost tracks

h1	
Entries	11781
Mean	0.473
RMS	1.04

FPCCD tracking

CA mini - vector tracking

P_T (GeV)

ϵ_{trk}

FPCCD tracking

CA mini - vector tracking

p_t/GeV