Status of MarlinTPC

Jason Abernathy[1], Klaus Dehmelt[2], Ralf Diener[2], Jan Engels[2], Jim Hunt[3], Matthias Enno Janssen[2], Martin Killenberg[4], Thorsten Krautscheid[4], Astrid Münnich[5], Martin Ummenhofer[4], Adrian Vogel[2], Peter Wienemann[4], Simone Zimmermann[4]

[1]University of Victoria
[2]DESY
[3]Cornell University
[4]University of Bonn
[5]RWTH Aachen

LC-TPC Large Prototype Meeting
February 25, 2008
DESY, Germany

Introduction to MarlinTPC

- Common simulation, reconstruction, digitisation and analysis framework for LC-TPC
- Builds on top of LCIO, Marlin and other ilcsoft tools
- Versatility:
  - works for all TPCs that can be described by GEAR: prototypes, collider detectors, ...
  - works for GEM, Micromegas and wire amplification
  - works for pad and pixel readout
  - works for FADC and TDC based electronics
- Advantages:
  - easy comparability (algorithms, technologies, geometries, ...)
  - easier transferability from prototypes to full size detector in collider environment
  - high re-usability of code

universitätbonn

Overview

- Simulation
- Digitisation
- Reconstruction
- Analysis
- Calibration
- TPCCondData
- Tools
- Validation

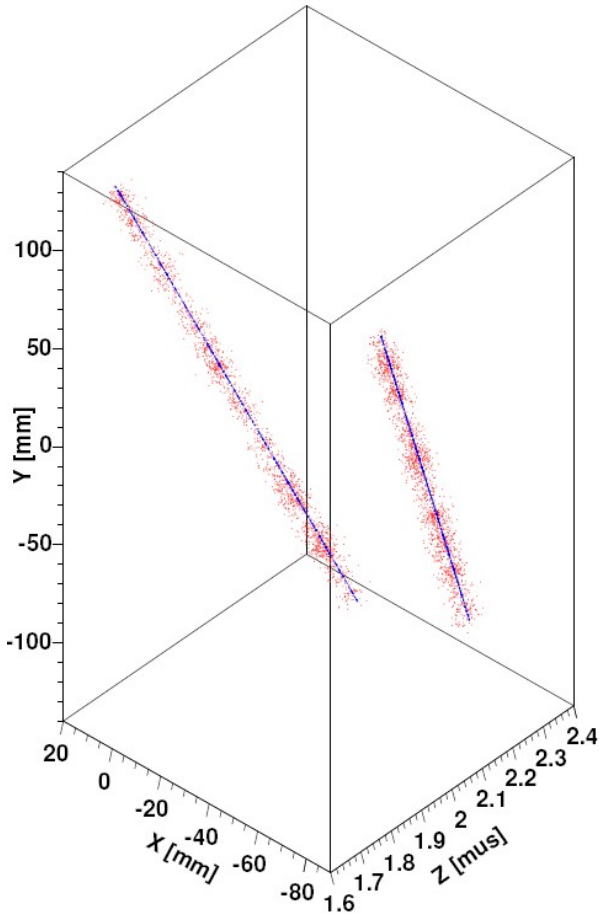
At present in repository:

- 33 processors from all categories in trunk
- > 800 commits in < 1 year
- 13 different authors

universitätbonn

Simulation

- Creates primary ionisation from a parametrisation of HEED simulation
- Parametrisation available for Ar-CO_2-CH_4 (93-2-5), Ar-CH_4 (90-10), Ar-CH_4 (95-5)
- Faster than full HEED simulation
- Correct treatment of delta electrons in magnetic fields



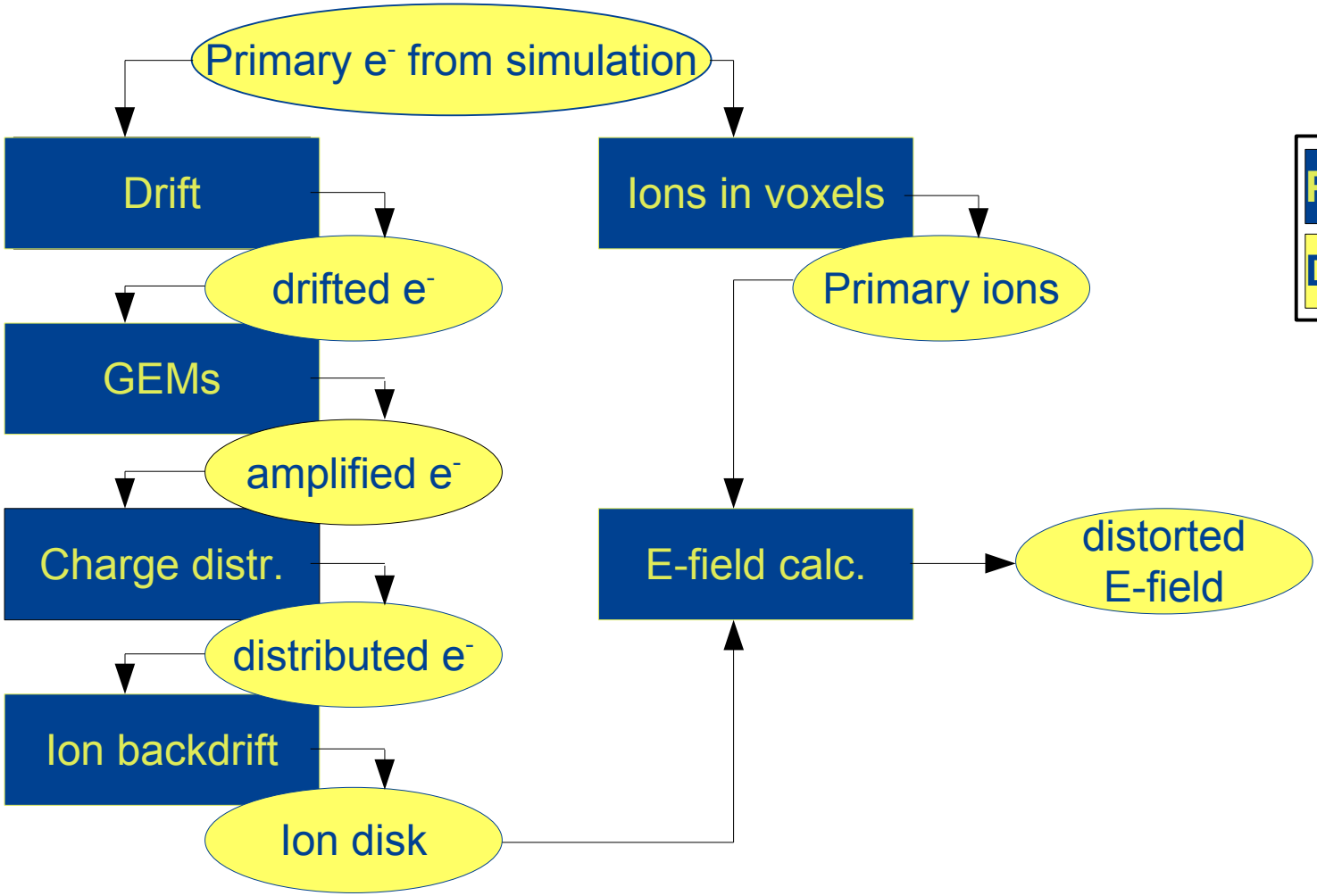
universitätbonn

Digitisation

- Simulates detector response to primary ionisation
- Reads primary charge, provides TPC raw data
- Takes ILC bunch structure properly into account
- So far only available for GEM amplification with FADC readout
- Rather detailed simulation which tracks individual electrons up to amplification process, includes many details (E-field distortions from ions, ...)
- A faster version working on Mokka hits is planned once important disturbing effects are known from detailed digitisation

universitätbonn

Digitisation flow chart



universitätbonn

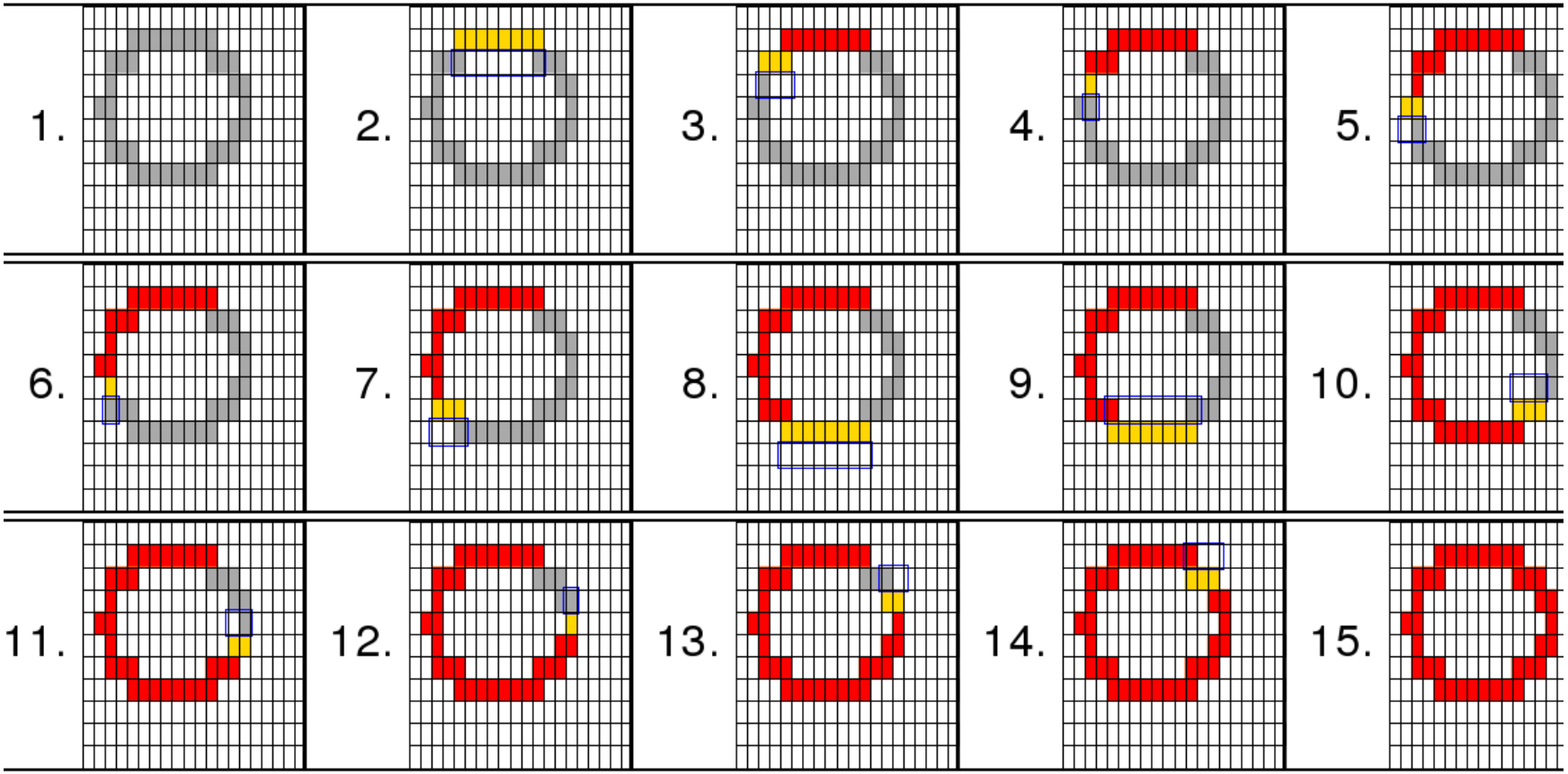
Reconstruction

Data Structure	Processor Name	Collection Name
`TrackerRawData`		TPCRawData
	TrackerRawDataToDataConverter	
`TrackerData`		TPCConvertedRawData
	PedestalSubtractor	
`TrackerData`		TPCData
	PulseFinder	
	ChannelMapper	
	CountsToPrimaryElectronsProcessor	
`TrackerPulse`		TPCPulses
	HitTrackFinderTopoProcessor	
`TrackerHit`		TPCHits
`Track`		TPCTrackCandidates
	TrackSeeder	
`Track`		TPCSeedTracks
	TrackFitterLikelihood	
`Track`		TPCTracks

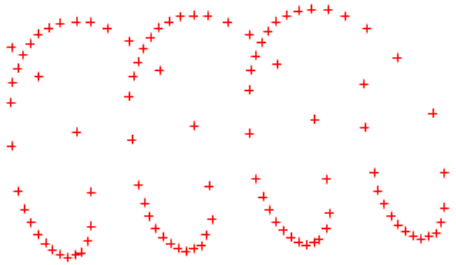
+ several correction processors

universitätbonn

Topological track finder



- Works in 3D without specific track hypothesis

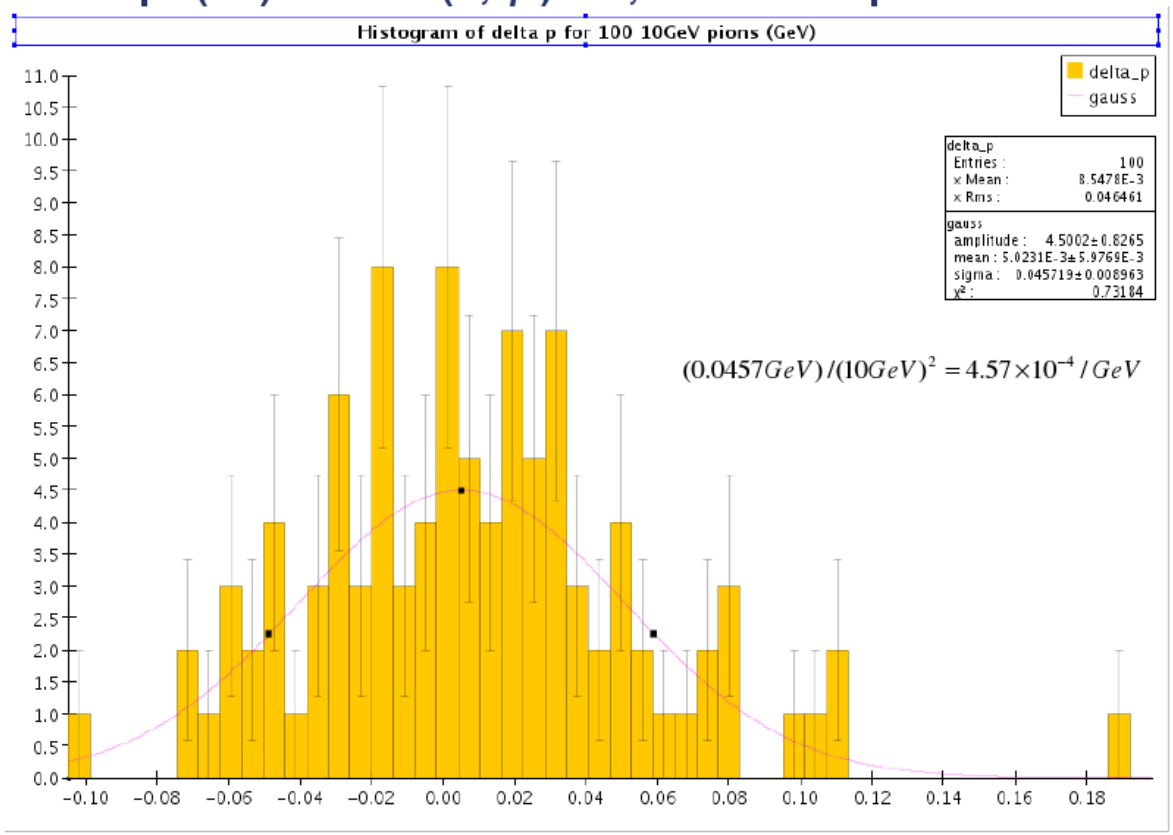


universitätbonn

Track fitter

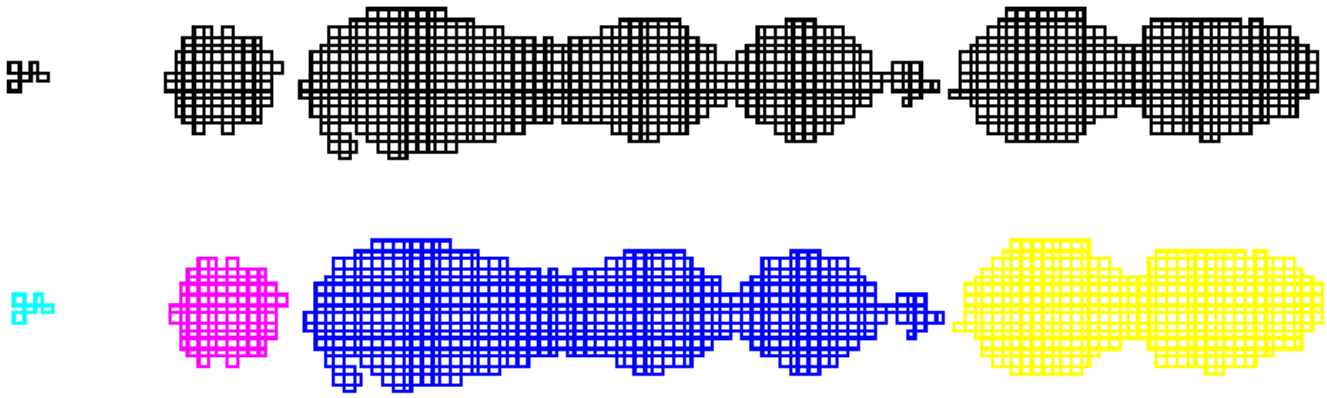
- Likelihood method implemented, performance not yet as expected
- χ^2 based fitter almost done

delta p (Ω) from (r,ϕ) fit, 10GeV pion

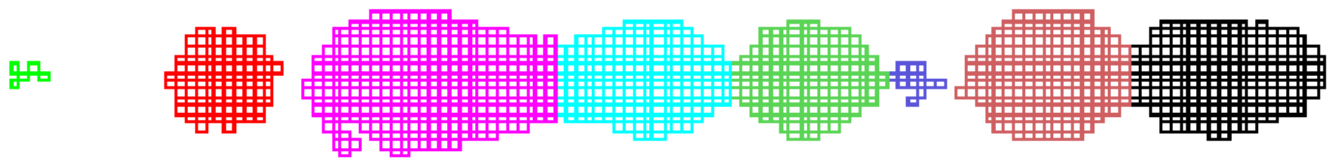


Reconstruction of TimePix data

- Zero-suppression
- Cluster finder: group all topologically connected pixels to clusters (works only for setup with GEMs)



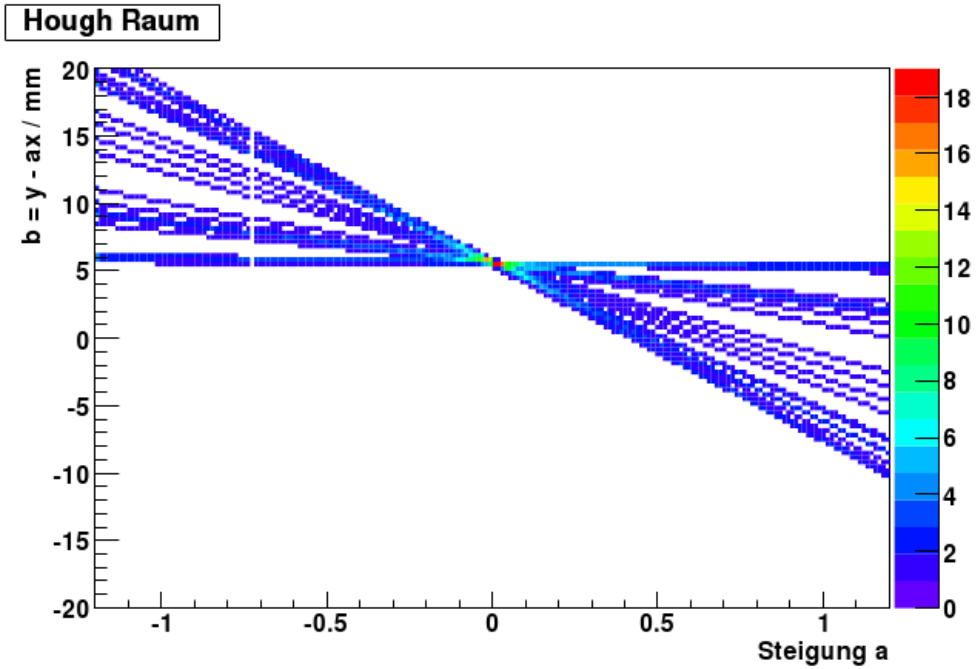
- Cluster separator: separates obviously distrinct cluster by projecting all pixels onto a straight line fit and cut at minima (only for GEM setups)



universitätbonn

Reconstruction of TimePix data

- Hit calculation: calculates centre-of-gravity of clusters (using charge info if available)
- Track finding: uses linear Hough transformation (every hit is a straight line in Hough space), intersection of tracks is estimate for track parameters



- Track fitting is shared with “normal” reconstruction

universitätbonn

Analysis

- First processor available producing residual plots
- Will extend list of processors producing figures of merit agreed on at first ILC TPC Analysis Jamboree
  - resolution from geometric mean of fits with and without test row
  - resolution using external reference track (hodoscope or MC truth)
  - resolution vs. drift distance
  - residuals vs. position on pad to check for biases
  - ...

universitätbonn

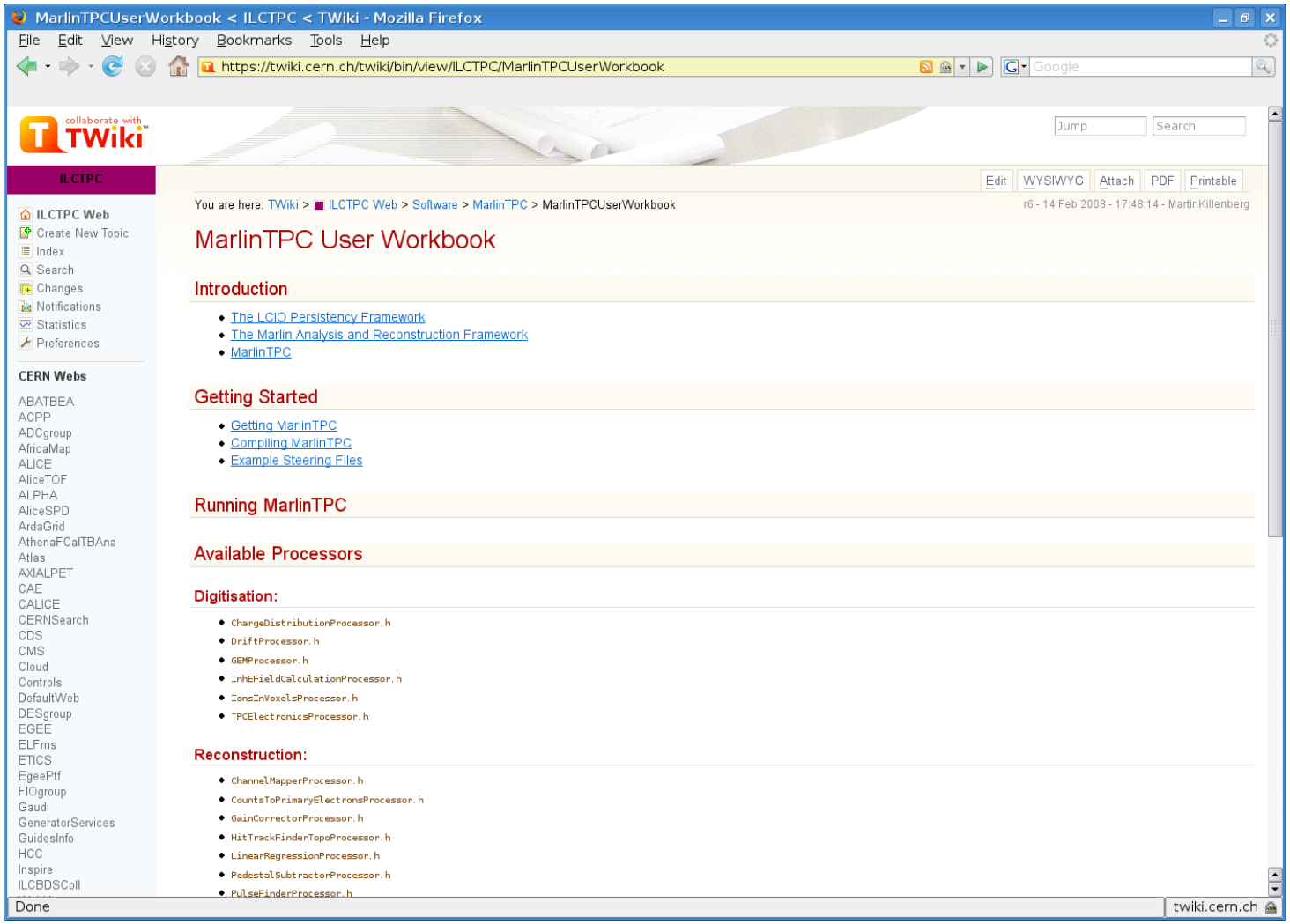
Installation procedure

- Since December 2007, MarlinTPC is integrated in `ilcinstall` (see `http://ilcsoft.desy.de`)
- Allows easy installation of MarlinTPC together with required other ilcsoft software
- For the impatients:
  Just type
  `svn co svn://pi.physik.uni-bonn.de/MarlinTPC/trunk`
  and have a look

universitätbonn

Wiki workbooks

- User workbook
- Developer workbook



universitätbonn

Summary

- Rapid MarlinTPC development during last year thanks to increasing number of developers
- Simulation, digitisation and reconstruction already in good shape for LP work
- MarlinTPC has become “working horse” for first small prototype analyses and for machine background study to study impact of ion disc on incoming tracks
- So far no work on implementation of calibration and alignment algorithms
- MarlinTPC on right track for LP

universitätbonn