MarlinTPC and Testbeam Software

Astrid Münnich

ECFA Meeting, 29. May 2013

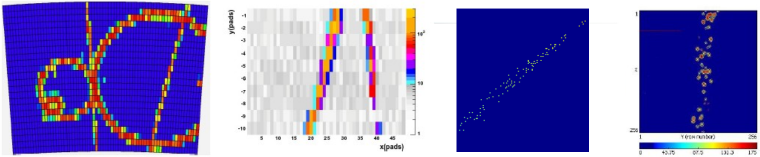
DESY, Germany

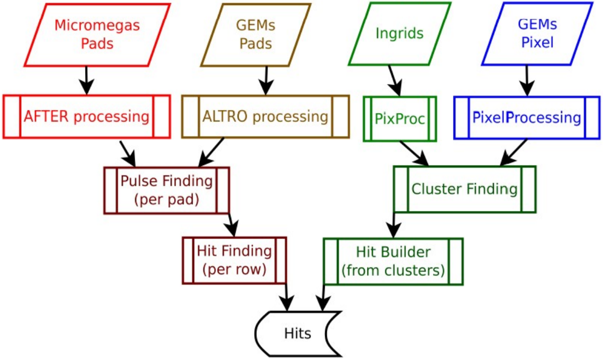
Overview

- MarlinTPC
- Geometry Description
- Grid & Conditions Data
- Tracking
- Testbeam Library

MarlinTPC

Reconstruction Framework for many different technologies:





Micromegas Pads

GEMs Pads

Ingrids

GEMs Pixel

AFTER processing

ALTRO processing

PixProc

PixelProcessing

Pulse Finding (per pad)

Cluster Finding

Hit Finding (per row)

Hit Builder (from clusters)

Hits

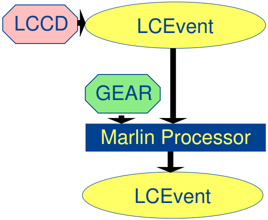
MarlinTPC

The packages structure:

- analysis
- calibration
- digitization
- examples
- general
- reconstruction
- simulation
- tools
- tpcconddata
- validation

$>$ 100 processors
$\approx$ 40 developer branches

Marlin Framework Principle:

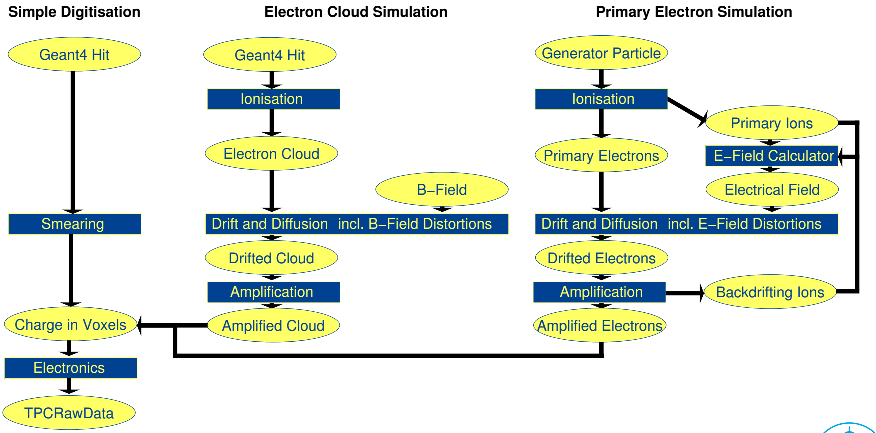


Information about MarlinTPC:
https://znwiki3.ifh.de/MarlinTPC/MarlinTPC

svn repository:
https://svnsrv.desy.de/desy/marlintpc/

Simulation & Digitisation

Different simulation approaches with different detail level:



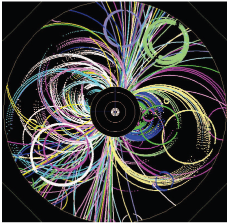
Geometry Description in GEAR

A realistic TPC end plate (EUDET LP) consists of multiple modules

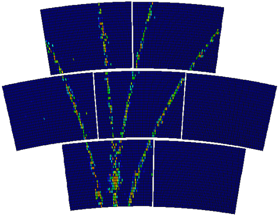
TPCModule

- Contains a pad layout implementation
- Pad plane implementation provides local coordinates
- Module has offset and angle to global coordinate system
- Accessing the pad plane through the module automatically provides correct global coordinates

ILC TPC: 1 big module



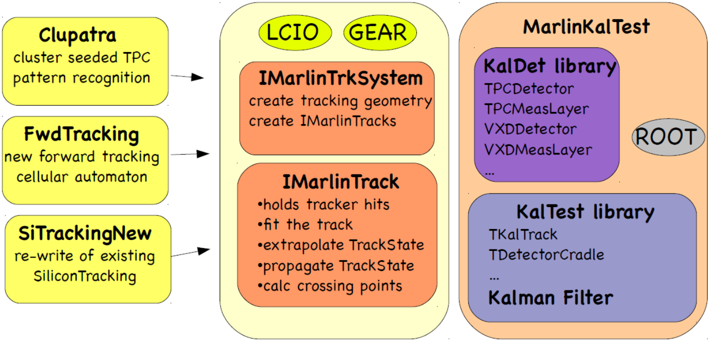
LCTPC: 7 modules



Tracking for ILD

Complex problem:

- Material ⇒ multiple scattering
- Correct error propagation
- Inhomogeneous fields



Clupatra
cluster seeded TPC
pattern recognition

FwdTracking
new forward tracking
cellular automaton

SiTrackingNew
re-write of existing
SiliconTracking

LCIO

GEAR

IMarlinTrkSystem
create tracking geometry
create IMarlinTracks

IMarlinTrack
•holds tracker hits
•fit the track
•extrapolate TrackState
•propagate TrackState
•calc crossing points

MarlinKalTest

KalDet library
TPCDetector
TPCMeasLayer
VXDDetector
VXDMeasLayer
...

ROOT

KalTest library
TKalTrack
TDetectorCradle
...
Kalman Filter

New: We can use the same packages Clupatra + MarlinTrk developed for full detector simulation also for the testbeam data

Testbeam Data on the Grid

Storage for raw data and converted LCIO data for all groups:
`srm://dcache-se-desy.desy.de/pnfs/desy.de/ilc/tpc/`

year	description/comments	electronics	RAW avail.	LCIO avail.
2010	Micromegas	T2K	NO	NO
2010	3 Asian GEM	ALTRO	YES	NO
2010	Octopuce (8 Ingrids)	TimePix+Muros	NO	NO
2011	DESY GridGEM	ALTRO	YES	YES
2012	6 Micromegas	int. AFTER	YES	YES
2012	DESY GridGEM	ALTRO	YES	YES
2012	Asian GEM	ALTRO	YES	YES
2013	7 Micromegas	AFTER	YES	YES
2013	DESY GridGEM	ALTRO	YES	YES

LCIO format is basis to exchange data!

Conditions Data

LCCD: The **L**inear **C**ollider **C**onditions **D**ata toolkit
User-defined data classes can be stored in LCIO files or a data base.
A Marlin processor provides this information during runtime:

- Channel mapping
- Channel quality
- Pedestals
- v_{drift} + Diffusion
- Gas conditions
- E-field settings
- B-field settings
- HV settings
- Field maps
- Electronics
- Calibration

Conditions data base available at DESY: `lctpcconddb01.desy.de`

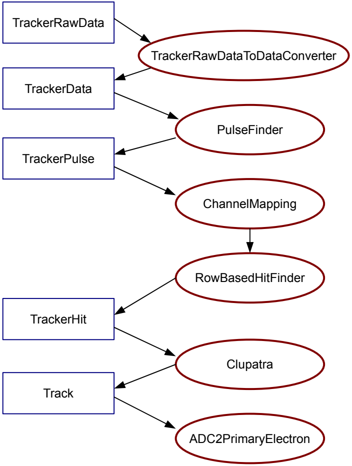
Reconstruction Example: Step 1

The first step is a pre-reconstruction:

1. Provide information from data base
2. Reconstruct pulses on each channel
3. Map hardware channel to GEAR pads
4. Reconstruct 3D space points
5. Tracking with clupatra
6. Obtain calibration factor to transform ADC counts to primary electrons

Now determine:

- Pulse time cut off according to cathode
- Pulse time shift according to anode position
- Other calibration info or corrections
- Fill conditions data base: drift velocity, diffusion, charge conversion etc...



TrackerRawData

TrackerRawDataToDataConverter

TrackerData

PulseFinder

TrackerPulse

ChannelMapping

RowBasedHitFinder

TrackerHit

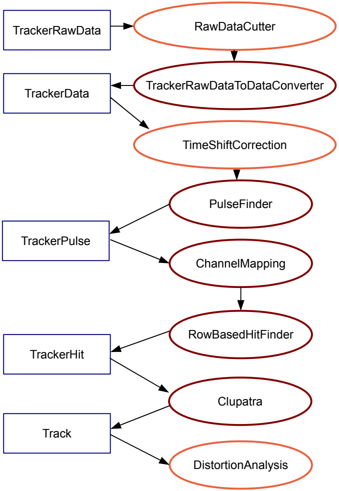
Clupatra

Track

ADC2PrimaryElectron

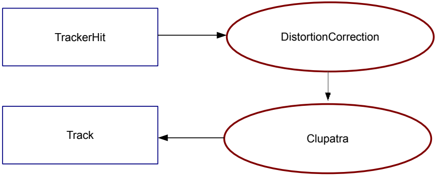
Reconstruction Example: Step 2

1. Apply timing cut
2. Apply time offset
3. Apply charge conversion during hit finding
4. Correct time offset of RCU
5. Do the tracking again
6. Calculate correction for field distortions



Reconstruction Example: Step 3

1. Correct distortions: Move hits accordingly along row and along z
2. Repeat tracking



Similar procedure for all module technologies.
Pulse reconstruction differs according to electronics.
Hit building is different for pads and pixel.

Status of MarlinTPC

- **Simulation:**
  different approaches available
- **Reconstruction:**
  pad based in reasonable shape, recent work on pulse finding
- **Tracking:**
  3 options: Clupatra, PathFinder, General broken lines
- **Corrections & Calibration:**
  work has started at DESY for gain calibration and correction of field distortion
- **Analysis:**
  not so much there

Other code might exist but not checked in yet.

Important:

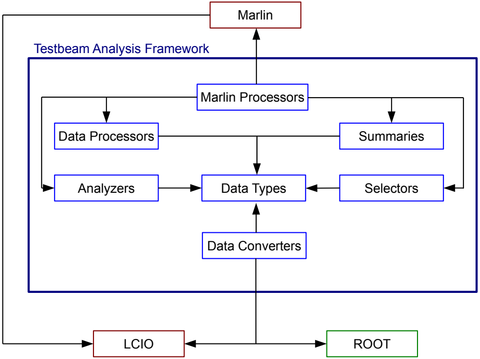
- Check in developments so that everybody can benefit.
- Also beneficial for the developer if other people try to use it.
- Help to keep the software up to date and avoid duplication of work!

Testbeam Software: The Framework

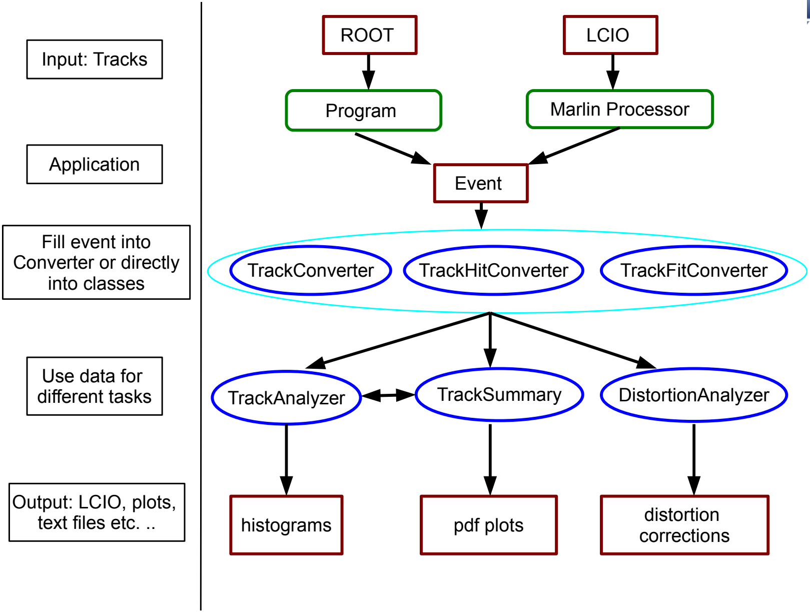
Depends on ILCSoft: LCIO, GEAR, MARLIN

Main Idea:

- Provide functionality in classes outside processors for more versatile use (e.g. different processors)
- Provide some fast tools for data quality checks during testbeam campaigns e.g. conversion to root for easy browsing and plotting



Testbeam Software: An Example



Input: Tracks

Application

Fill event into Converter or directly into classes

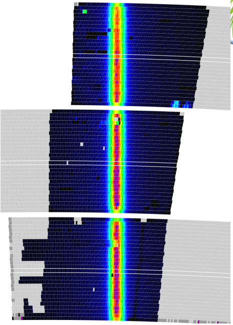
Use data for different tasks

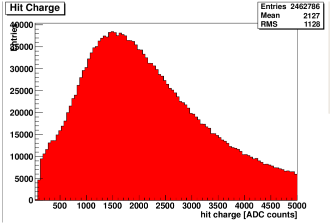
Output: LCIO, plots, text files etc. ..

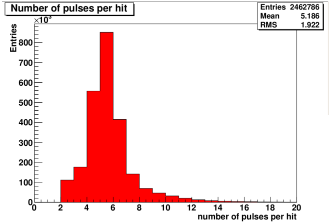
Testbeam Software: The Application

Python-based steering of all reconstruction steps:

- Automated configuration
- Creation and execution of Marlin xml files
- Makes summary/control plots for each step of the reconstruction (raw, pedestal, pulse, hit, track)
- Makes it easy to spot problems with the data: dead or noisy channels, not enough charge, etc.







Summary

MarlinTPC provides tools and reconstruction for the LCTPC collaboration.

Difficulty: Many different technologies → different needs
But: At a certain point reconstruction and problems become common.

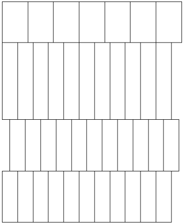
The basic reconstruction is available.
Real data needs: calibration and corrections
→ Developments have started.

Still some way to go in terms of software that is equipped to handle field maps and distortions.

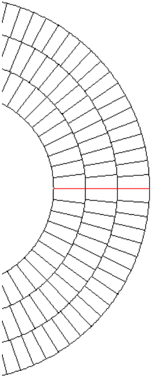
But also significant progress regarding tracking packages and towards completing the reconstruction chain for different technologies.

BACKUP

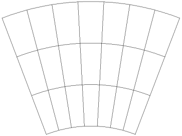
Pad Layouts in GEAR



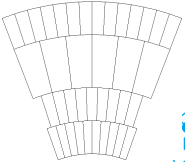
- **RectangularPadRowLayout**
  - Cartesian Geometry
  - All pads in one row are equal
  - Pad size and number of pads may vary from row to row



- **FixedPadSizeDiskLayout**
  - Polar Geometry
  - Complete circle
  - *All* pads have the same size



- **FixedPadAngleDiskLayout**
  - Polar Geometry
  - Segment of a circle
  - All pads have the same angle



- **VersatileDiskRowLayout**
  - Polar Geometry
  - Segment of a circle
  - All pads in one row are equal
  - Pad size and number of pads may vary from row to row