Digitisation in MarlinTPC

by Thorsten Krautscheid

University of Bonn

30. Januar 2008

Outline

Tasks

General Tasks

- processes MC data from (full) detector simulation (Geant4, Mokka...)
- simulates detector response
- provides realistic data samples

Processors and Helpers

Helper / Data classes

- VoxelTPC
- TPCVoxel
- EFieldVector

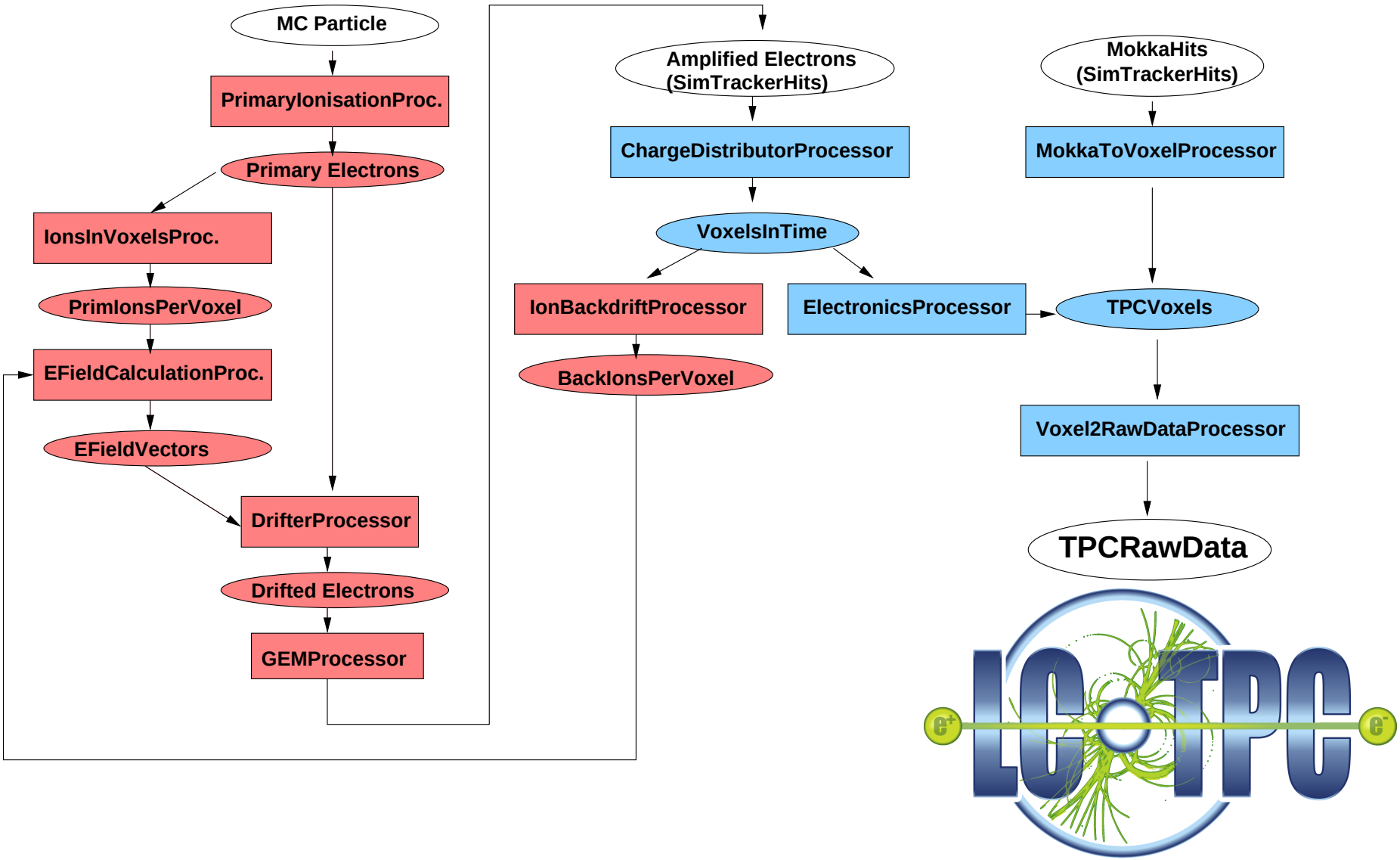
MarlinTPC Detector Simulation

- PrimaryIonisationProcessor
- IonsInVoxelProcessor
- DriftProcessor
- GEMProcessor
- IonBackdriftProcessor
- EFieldCalculationProcessor

MarlinTPC Digitisation

- ChargeDistributionProcessor
- TPCElectronicsProcessor
- MokkaToVoxelProcessor
- VoxelToRawDataProcessor

Structure


MC Particle
PrimaryIonisationProc.
Primary Electrons
IonsInVoxelsProc.
PrimIonsPerVoxel
EFieldCalculationProc.
EFieldVectors
DrifterProcessor
Drifted Electrons
GEMProcessor
Amplified Electrons (SimTrackerHits)
ChargeDistributorProcessor
VoxelsInTime
IonBackdriftProcessor
BackIonsPerVoxel
ElectronicsProcessor
MokkaHits (SimTrackerHits)
MokkaToVoxelProcessor
TPCVoxels
Voxel2RawDataProcessor
TPCRawData

Working Principle

Central helper class: VoxelTPC

- bins TPC in voxels:
  - binning in x,y by `gear::padIndex`
  - binning in z in arbitrary units
- fills/removes charge in/from TPCVoxel
- distributes charge according to longitudinal and transversal diffusion
- intrinsic pile-up of charges
- supports both geometries of the pad plane:
  - `gear::FixedPadSizeDiscLayout`
  - `gear::RectangularPadRowLayout`

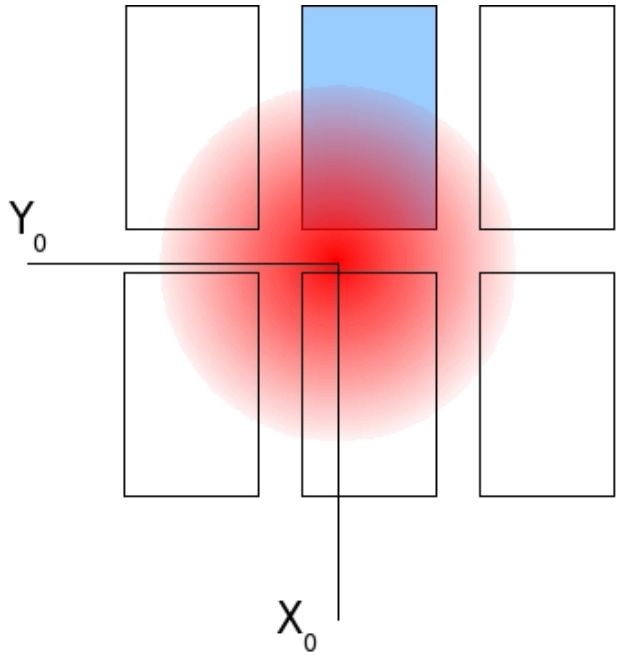
Container class: TPCVoxel

- derived from `LCIO::LCFixedObject`
- 3D bin of the TPC:
  - x,y defined by `gear::padIndex`
  - z by user defined unit (arbitrary)
- stores charges

Charge Distribution

Charge Distribution

- integration over a 2D gaussian
- rotation in case of circular pad layout

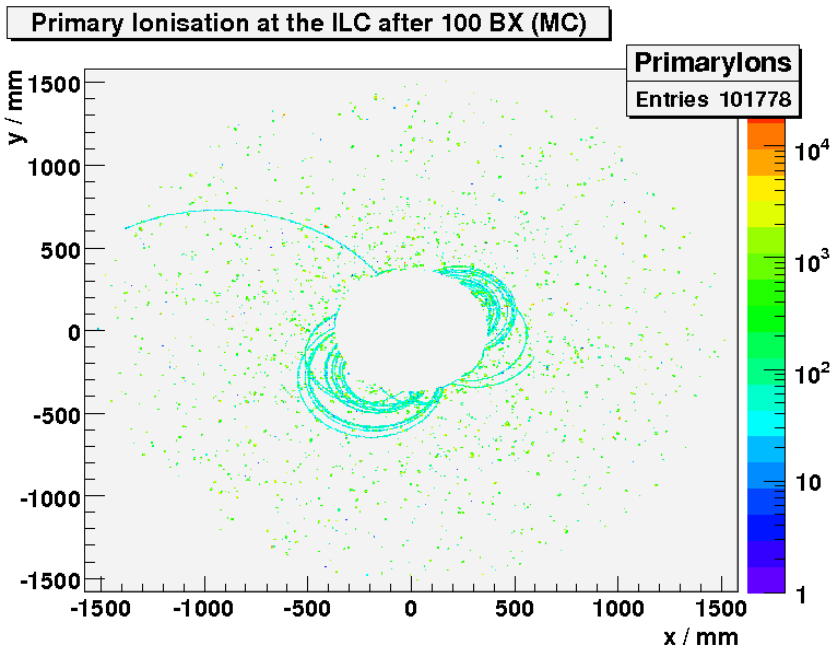


$$C(x,y) = Q_{hit} \cdot e^{\left(\frac{-(x-x_0)^2}{2\sigma^2}\right)} \cdot e^{\left(\frac{-(y-y_0)^2}{2\sigma^2}\right)}$$

$$Q_{pad} = \int_{x_{min}}^{x_{max}} \int_{y_{min}}^{y_{max}} C(x,y)\, dxdy$$

Processors using VoxelTPC:

- IonsInVoxelProcessor
- IonBackDriftProcessor
- ChargeDistributionProcessor
- MokkaToVoxelProcessor



Status and outlook

done:

- basic detector simulation
- implementation of VoxelTPC and TPCVoxel

to do:

- completion and implementation of the MokkaToVoxelProcessor
- completion and implementation of the VoxelToRawDataProcessor
- implementation of inhomogenious fields in DriftProcessor