The PXD Digitizer

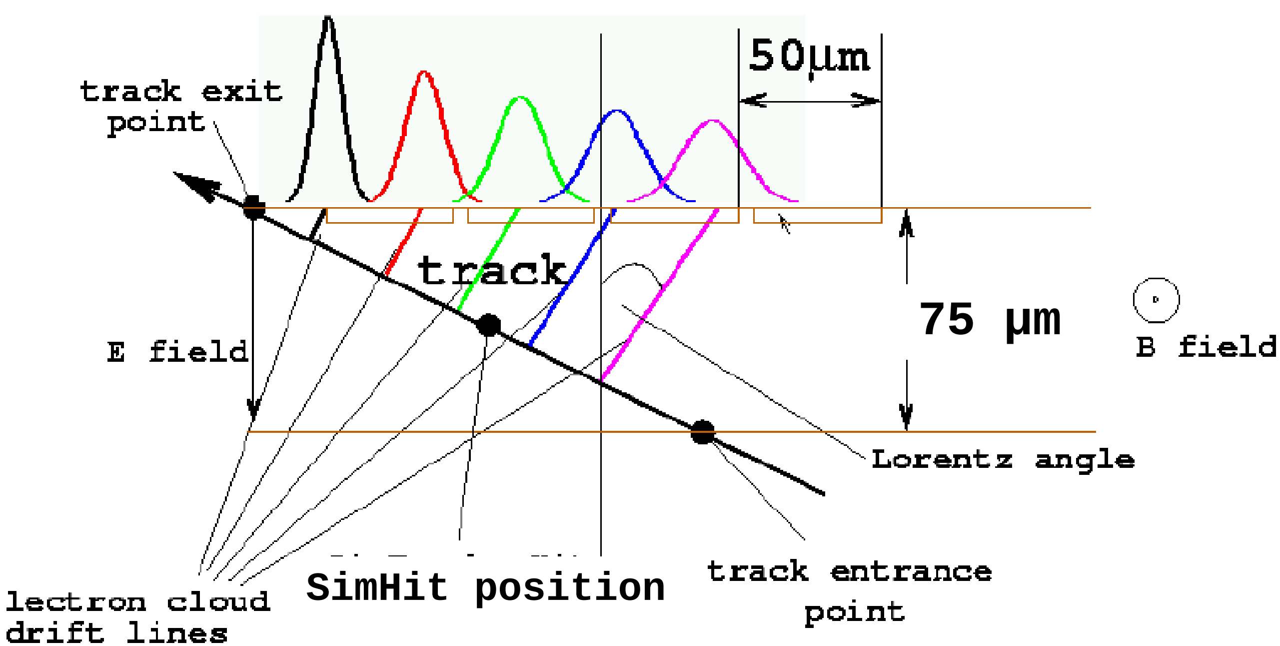
Peter Kvasnička
Institute of Particle and Nuclear Physics,
Charles University, Prague

Belle2 Framework/DAQ meeting, Munich,
21-23 February 2011

CU Prague

The PXD digitizer: What it does

- **For each simulated hit (PXDSimHit) obtained from Geant4**
  - **Track generation.** The SimHits are created where Geant4 detects a passage of a particle through an active detector. δ electrons are treated separately
  - **Ionization.** The track segment is divided into sub-segments (“ionization points”), whose charge is smeared according to the Landau distribution using the code borrowed from Geant4 (G4UniversalFluctuation). This provides substantially higher precision than in the generating “all-detector” Geant4 simulation.



The PXD digitizer: What it does (cont’d)

CU Prague

- **Hit reconstruction.** In the present implementation, digitization is coupled with hit reconstruction to avoid the storage of digits when they are not needed (the digits can however be saved if desired).
- **Clustering i**s based on seed and neighbor threshold values.
- **Hit position** is estimated separately in each coordinate, by center-of –gravity for cluster projections of 1 or 2 pixels, and analog head-tail (approximate the cluster by a uniform charge distribution consistent with pixel signals) for larger clusters.
- **Output** The output of the digitizer is

1. a collection of PXDDigits, that is, fired pixels stored serially with end-of-cluster marks. Each PXDDigit contains:
   - encoded pixel row and column number + end-of-cluster marks.
   - pixel signal in electrons or AD
2. a collection of PXDHits, with each reconstructed hit containing (apart from sensor identification)
   - position estimate (corrected for mean Lorentz shift) and its error covariance
   - total collected charge (analog in electrons or digital in ADU
   - A reference (relation) to SimHits that contributed.

CU Prague

The PXD digitizer: What it does (cont'd)

- **Drift and diffusion.** The ionization points are then drifted to the readout plane, Lorentz-shifted in the presence of a magnetic field and smeared by a Gaussian distribution to account for diffusion during the drift time.
- **Lateral diffusion.** Finally, the total charge of an ionization point is split into carrier groups of ~100 electrons. For each carrier group, a random walk in the readout plane is sampled until the internal gate region of a pixel cell is reached. Thus, we have a redistribution of the charge to neighbor pixels.
- **Digitization.** For each pixel electrode we now have a signal dependent on how many ionization points contributed.
  - **Background and noise.** The pixels are then populated by random electronic noise and background signal as appropriate.
  - **Analog-To-Digital converters** will be used in the data processing pipeline, Therefore, the analog signals are converted first into digital values.

CU Prague

Issues to discuss: Overview

- **Physics: Two issues appeared at the Bonn meeting**
  - Landau fluctuations: Odd behavior seen in energy spectra.
  - Handling of photons The digitizer (or, better, the SensitiveDetector class) ignores photons.
- **Code organization**
  - Separate the digitizer and the clusterizer: The original concept was not to store digits when they are not needed. But the digits and clustering algos will be important for some time now.
  - Calculation of resolutions. It is not possible (or is at least unreasonably costly) to calculate error matrices of individual reconstructed hits. Currently they are estimated based histograms acquired in calibration runs.
- **Technical issues (code optimization, storage)**
  - Storage of digits: The number of digits to process/store can be huge, so the storage and processing times have to be optimized.
  - Handling of photons The digitizer (or, better, the SensitiveDetector class) ignores photons.

Issues to discuss 1: Landau fluctuations

- **Current implementation:**
  - We have thin detectors, so Geant typically uses **a single step** to track a particle through.
  - At such large steps, the electromagnetic simulation is not precise.
  - The energy fluctuations are thus re-computed using basically the same formulas that Geant would use with smaller step (G4UniversalFluctuation)
  - There is a particle energy threshold, below which mean deopisited energy is used instead.
  - Parameters that control energy fluctuations:
    - `bool m_landauFluct` If set, internal Landau fluctuations will be used instead of those directly from Geant4.
    - `float m_landauBetaGammaCut` Below this beta*gamma factor, internal Landau fluctuations are not used (default 0.7)
    - `double m_prodThreshOnDeltaRays` Production threshold cut on delta electrons , same as in Geant4 (80keV ~ 0.05 mm).
    - `double m_segmentLength` Length of path segment - spatial precision of charge distribution simulation. 10 µm or less recommended.

CU Prague

Issues to discuss 1: Landau fluctuations

- **Proposal:**

  Dispose of internal fluctuations code and use smaller Geant4 steps in Si instead. Optimize at a later stage.

  +: Get rid of the internal fluctuation mechanism

  - : More expensive in terms of computation time.

  - : More complex, or at least different SensitiveDetector and digitizer code: we will have to digitize Geant4 steps.

- **Implementation:**
  - Geant step size in Si should be limited to ~10 µm or even less.
  - What shall be done:
    - Set limit on step size in UserLimits for Si sensitive volumes.
    - Implement a “limiting” transport process that will do nothing but apply the limit in SI sensitive volumes.

CU Prague

Issues to discuss 2: Handling of photons

- **Current implementation:**
  - The SensitiveDetector class does not generate hits for photons.
- **Tasks:**
  - Check that we have correct physics in Geant4
  - Adjust the SensitiveDetector class.

CU Prague

Issues to discuss 3: Code organization

- Separate the digitizer and the clusterizer: The original concept was not to store digits when they are not needed. But the digits and clustering algos will be important for some time now.
- This has to be done, for several reasons:
  - Clearer code organization (obvious, see above)
  - The use of MC particle / SimHit relations to digits will be cleaner if done using the basf2 mechanisms.

CU Prague

Issues to discuss 4: Technical: Storage of digits

- We need TObject-ized constant-size objects for efficient storage. Clusters are inherently of variable size, storage of pre-defined arrays of pixels is not optimal.

– **Proposal:**
  - – Store individual fired pixels with end-of-cluster flag set as a bit in the CellID. Danger: Can we be sure that the order is preserved in TClonesArray?
  - – We need an int to store CellID (encoded cell row and column), a codec class is provided.
  - – Using one bit as an end-of-cluster flag, we will have serialized storage of digits, relatively easy to read and process.
  - – Individual digits may have different relations to Mcparticles/SimHits, these will be combined in hit reconstruction.
  - – Do we accept multiple relations per reconstructed hit?

CU Prague

Issues to discuss 4: Technical: Data structures

- **The structure that stores digits in processing is**

```
typedef std::map<int, std::map<int, Digit*> > DigitsMap;
   // unique sensorID, unique pixelID, Digit
struct Digit {
  int cellIDZ;
  int cellIDRPhi;
  float charge;
  PXDSimHitMap pxdSimHitMap;
};
```

- This seems to be quite efficient overall, but it is awkward to navigate, for example when one searches nearest neighbors of a pixel – and that is the time-consuming step.

- **Proposal:**
  - Invest in a single 2D array (boost::MultiArray?), re-used for each sensor.

    +: Fast access, fast clearing of elements

    -: Larger memory footprint, some clearing costs.
  - Use sparse array storage, that is, a 1D array plus column and row index arrays.

    +: Memory efficient, relatively fast navigation

    -: Re-allocation costs, costly clearing of elements.

CU Prague

Schedule

– The items discussed here will be implemented after the first release of the digitizer appears (exception: storage of digits)

Thanks for your attention!