VXD Simulation and Reconstruction: Overview

Peter Kvasnička
Charles University in Prague

Belle II

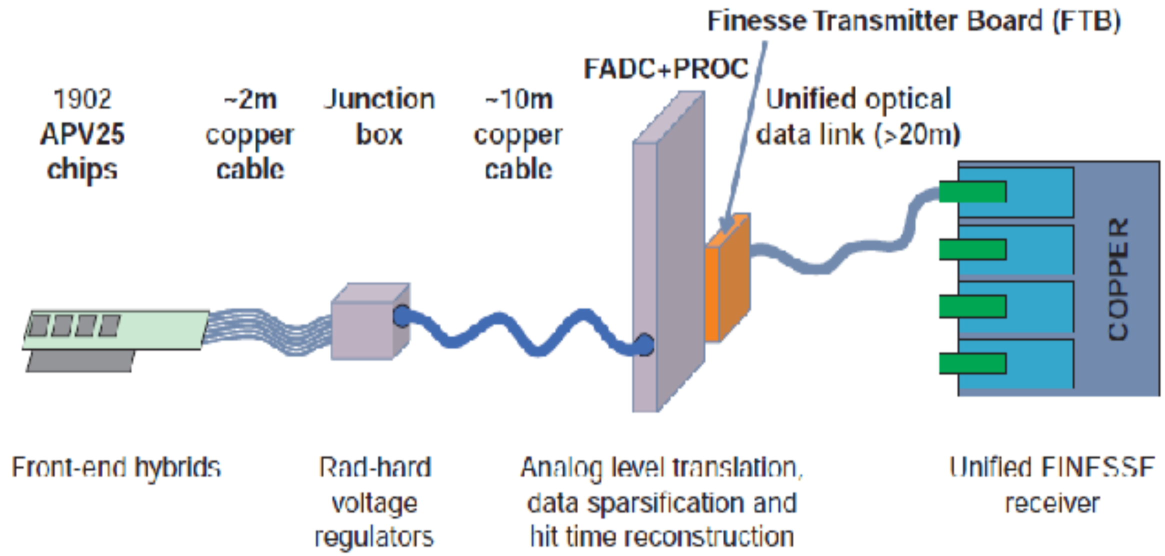
In this talk

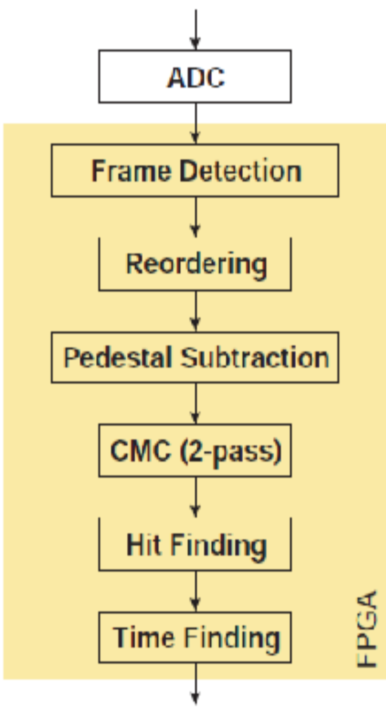
PXD and SVD

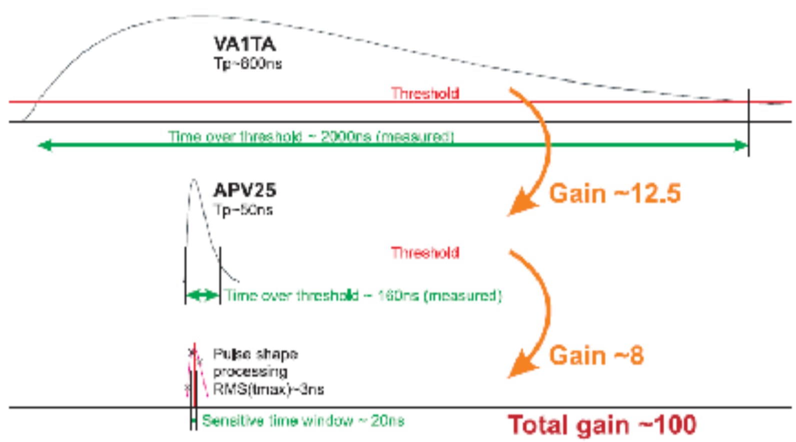
1. simulation
2. clustering
3. hit reconstruction

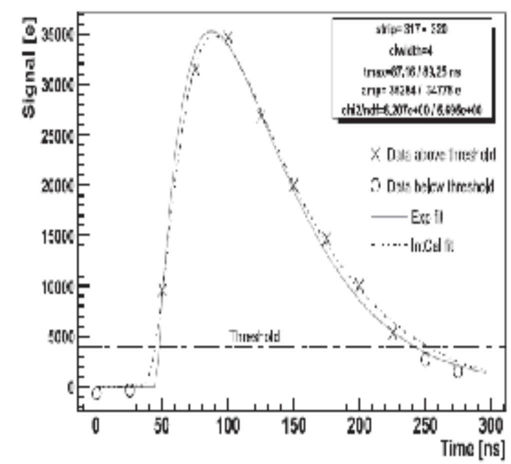
- what is implemented
- what is missing
- what for the testbeam

SVD and SVD DAQ

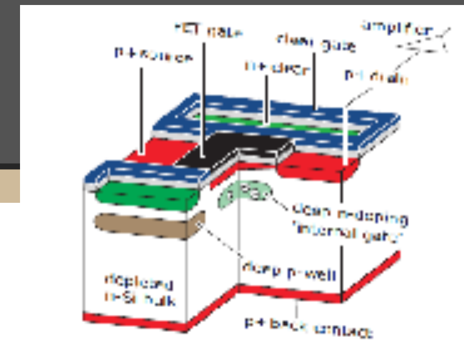


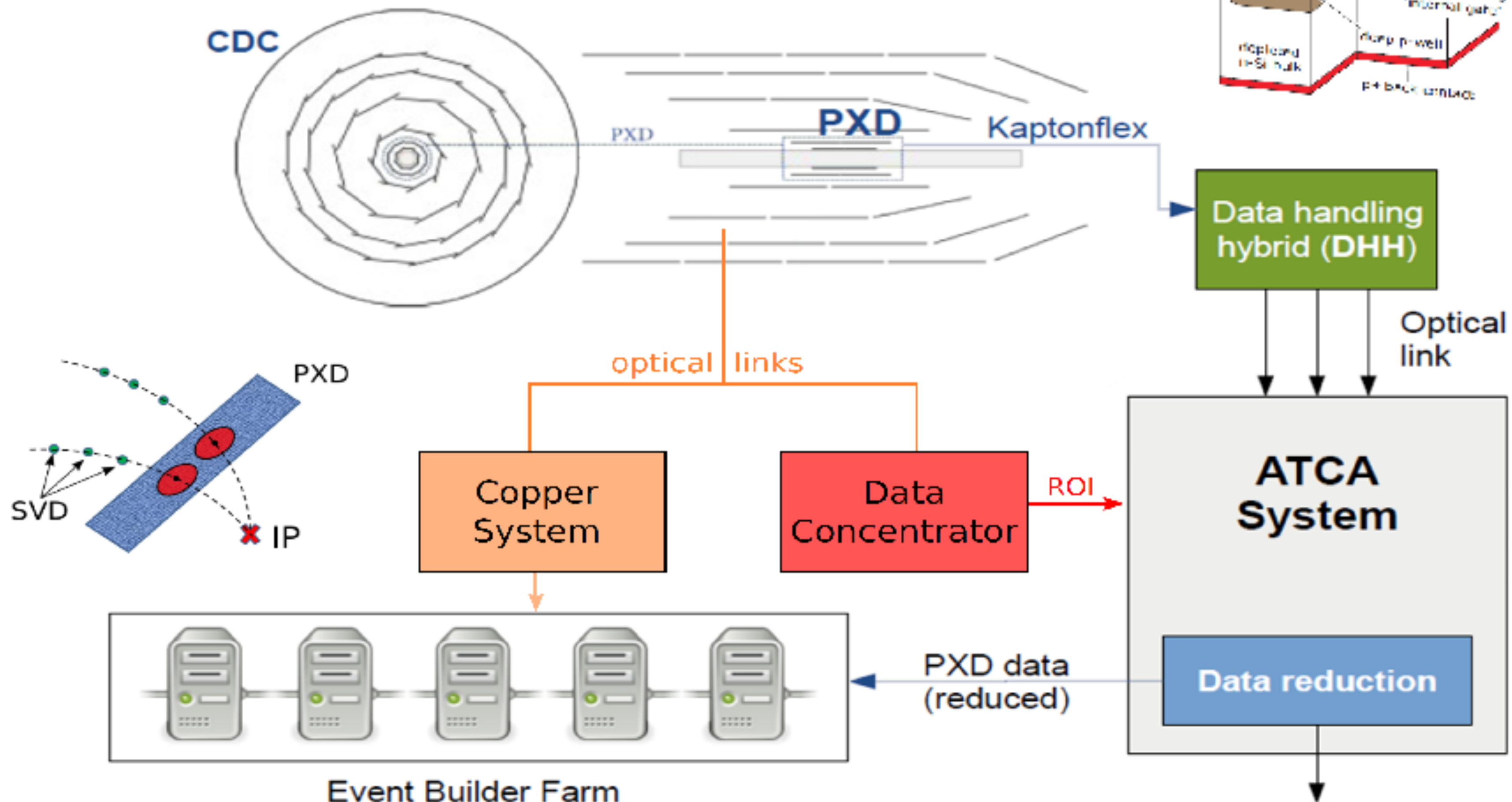


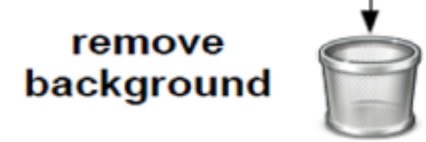




PXD and PXD DAQ

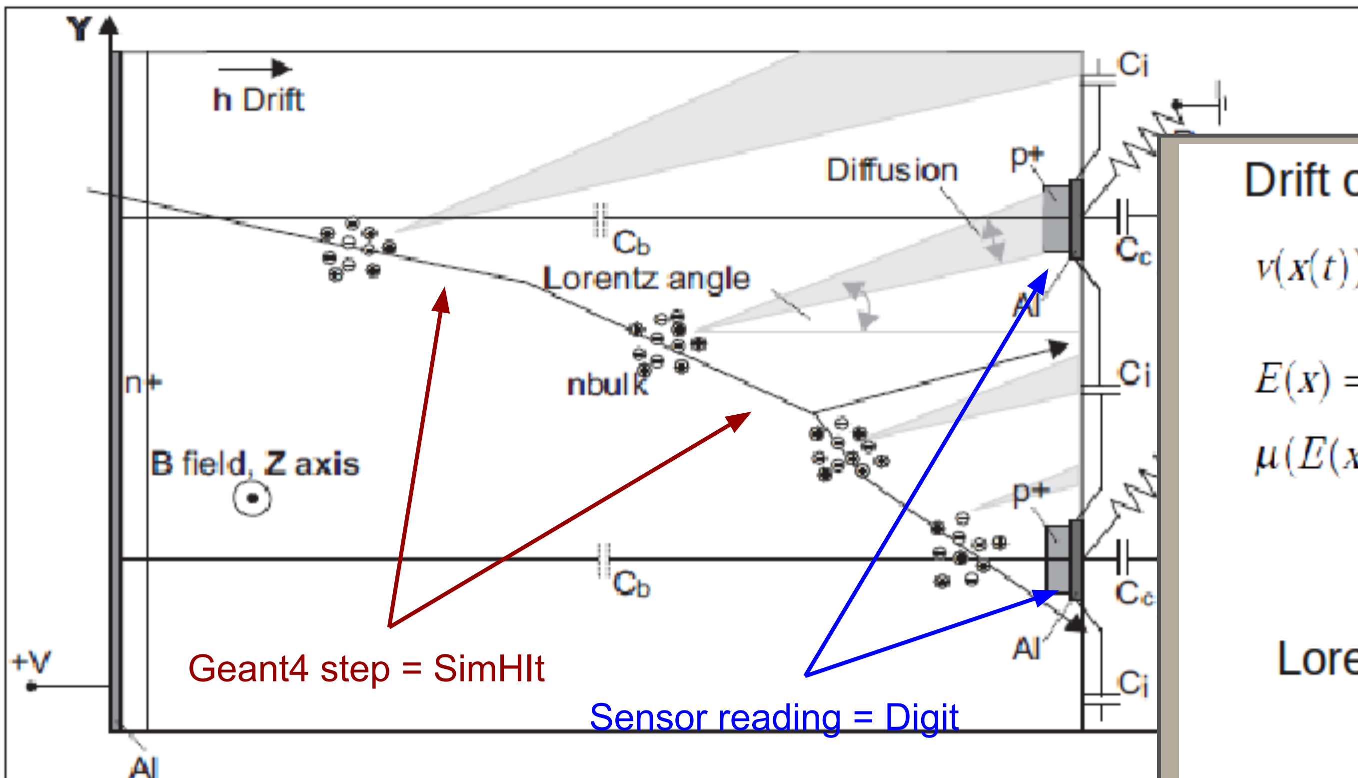






I could show a drawing with much more boxes here, but I like this old drawing that I borrowed from Michael Schnell (thanks!)

PXD and SVD simulation



Drift of electrons:

$$v(x(t)) \quad \mu E(x)$$

$$E(x) = -\left(\frac{V+V_{\mathrm{dep}}}{d} - \frac{2xV_{\mathrm{dep}}}{d^2}\right)$$

$$\mu(E(x),T) \quad \left(\frac{v_{\mathrm{m}}/E_{\mathrm{c}}}{(1+(E(x)/E_{\mathrm{c}})^{\beta})^{1/\beta}}\right)$$

Lorentz angle:

$$\tan(\vartheta_{\mathrm{L}}) = \frac{\int_x^d \mu(E(x))\, rB\, dx}{\int_x^d dx}$$

Hall factor

$$r = 1.13 + 0.0008(273 - T) \quad \text{for e}$$

Drift-to-target model:

- realistic for PXD
- fiction for SVD

Clustering

- Clusters formed from digits with signals over a fixed (zero-suppression) threshold
- (fast) 2D clustering used both for PXD (u,v) and SVD (u/v, time)
- SVDDigits are single APV samples - due to uncertainty about readout modes

PXD/SVD hit position reconstruction

- Simple expressions for hit positions and their uncertainties based on the classical Turchetta's* paper
  - Exception: "Unseen charge" method used to estimate errors in cluster sizes 1,2 for PXD, and 1 in SVD.
  - Cluster shape is taken into account when forming the error covariance matrix for PXD.
- Centre-of-gravity for cluster sizes <3, analog head-tail for larger sizes.
- So far:
  - no η correction implemented
  - no special analysis of large clusters,
  - no attempt to merge long clusters

---

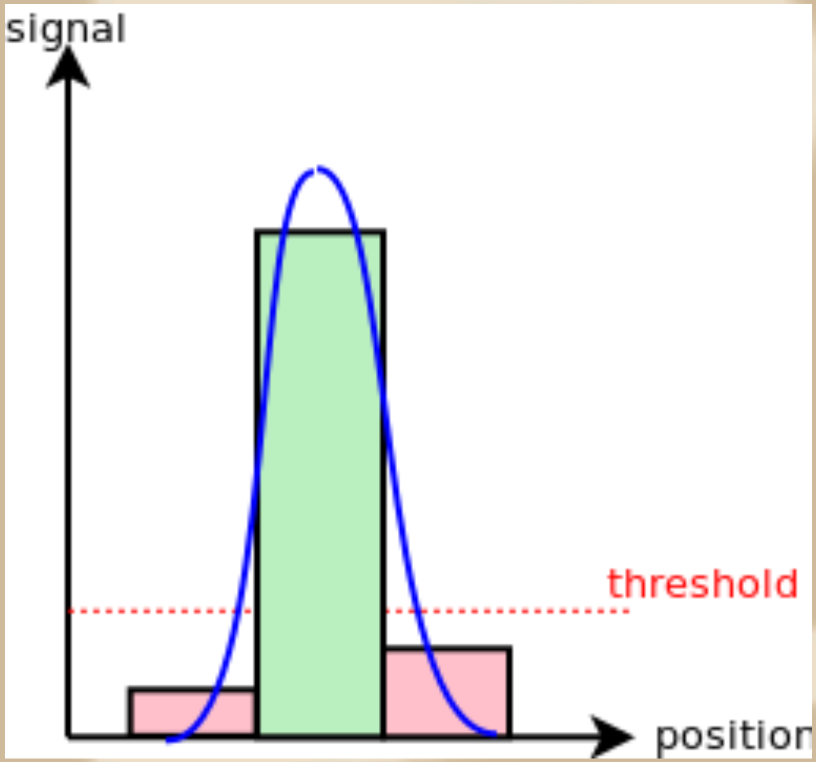
*Turchetta, R. : Spatial resolution of silicon microstrip detectors. Nuclear Instruments and Methods in Physics Research A335 (1993) 44-58

Unseen charge estimate

Motivation:

Estimate position error of a size 1 cluster.

- Position estimate is clear - the centre of the strip/pixel.
- The usual error estimate of pitch/sqrt(12) is wrong.
- We don't only have the one signal, we also have zero signals in neighbours.
- The higher the signal, the more sure we can be the actual position is near the centre of the pixel/strip.
- The same works to obtain better error estimates for size 2 clusters.



Hit reconstruction in the SVD

Hit position: u_c or v_c	Hit position error: σ_u or σ_v
Positions and their errors are calculated separately for u and v strips. There is no information on correlation. u_i strip position, S_i strip signal, S cluster charge, p strip pitch	
size in u = 1 Center of strip	$\sigma_u = p_u \frac{S_{thr}}{S+S_{thr}}$, S_{thr} - 0-suppression cut
size in u = 2 $u_c = \frac{S_1 u_1 + S_2 u_2}{S}$	$\sigma_u = a p_u \frac{S_e}{S}$, S_e - strip noise, $a = 1.4$
size in u > 2 $u_c = \frac{u_h+u_t}{2} + p_u \frac{S_h - S_t}{2S_0}$, $S_0 = \sum S_i$	$\sigma_u = \frac{p_u}{2}\left[2\left(\frac{S_{thr}}{S_0}\right)^2 + \frac{1}{2}\left(\frac{S_h}{S_0}\right)^2 + \frac{1}{2}\left(\frac{S_t}{S_0}\right)^2\right]^{1/2}$

The same formulas are used for v

Hit reconstruction in the PXD

Hit position: (u_c, v_c)	Hit position error $\begin{pmatrix} \sigma_u^2 & \rho\sigma_u\sigma_v \\ \rho\sigma_u\sigma_v & \sigma_v^2 \end{pmatrix}$

Positions and their errors are calculated separately from cluster projections to each direction. The correlation coefficient is calculated as

$$\rho = \frac{\sum_{pixels} S_i (u_i - u_c)(v_i - v_c)}{\left(\sum_{pixels} S_i [(u_i - u_c)^2 + \epsilon_u^2]\right)^{1/2} \left(\sum_{pixels} S_i [(v_i - v_c)^2 + \epsilon_v^2]\right)^{1/2}}$$

(u_i, v_i) pixel positions

S_i pixel signals

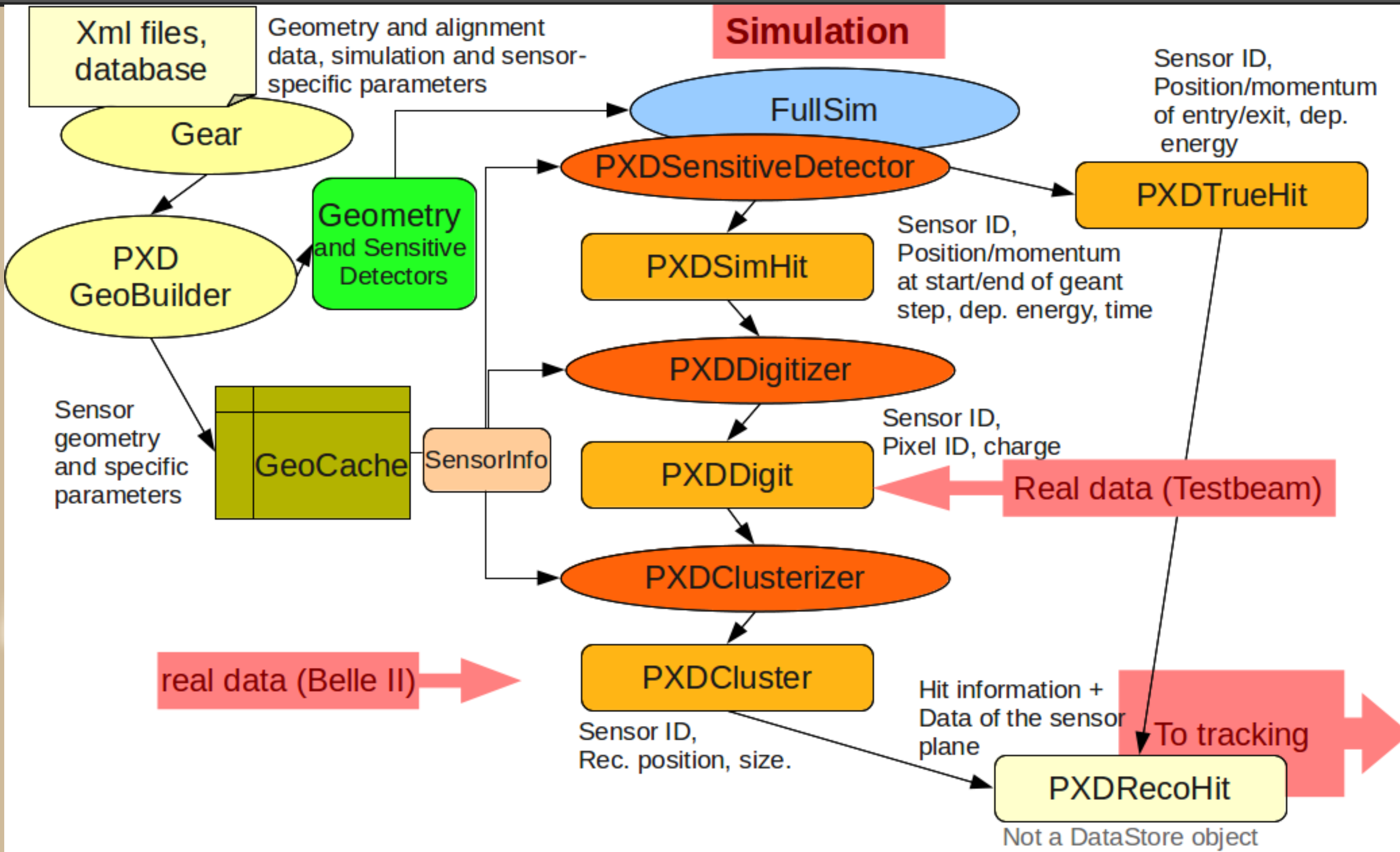
$\epsilon_u = \frac{p_u}{\sqrt{12}}$ in-pixel spread

$\epsilon_v = \frac{p_v}{\sqrt{12}}$ in-pixel spread

size in u = 1 Center of pixel	$\sigma_u = p_u \frac{(n_v + 2) S_{thr}}{S + (n_v + 3) S_{thr}}$ n_v cluster size in v; S_{thr} 0-supp. threshold
size in u = 2 $u_c = \frac{S_1 u_1 + S_2 u_2}{S}$	$\sigma_u = p_u \frac{(n_v + 2) S_{thr}}{S + (n_v + 3) S_{thr}}$ n_v cluster size in v; S_{thr} 0-supp. threshold
size in u > 2 $u_c = \frac{u_h + u_t}{2} + p_u \frac{S_h - S_t}{2 S_0}, \; S_0 = \sum S_i$	$\sigma_u = \frac{p_u}{2} \left[2\left(\frac{S_{thr}}{S_0}\right)^2 + \frac{1}{2}\left(\frac{S_h}{S_0}\right)^2 + \frac{1}{2}\left(\frac{S_t}{S_0}\right)^2 \right]^{1/2}$

The same formulas are used for v

PXD (and SVD) data processing in basf2: Overview



Data objects

SVDSimHit: Record of a single Geant4 step

```
/** ID of the sensor the energy was deposited in */
unsigned short m_sensorID;
/** PDG Code of the particle producing the Hit */
int m_pdg;
/** Theta angle of particle */
float m_theta;
/** Deposited energy in electrons */
float m_energyDep;
/** Time of energy deposition */
float m_globalTime;
/** Start point of energy deposition in local coordinates */
TVector3 m_posIn;
/** End point of energy deposition in local coordinates */
TVector3 m_posOut;
/** Momentum of particle at start of energy deposition */
TVector3 m_momIn;
```

SVDDigit: Record of a single strip signal sample

```
unsigned short m_sensorID; /**< Compressed sensor identifier.*/
bool m_isU;                /**< True if U, false if V. */
short m_cellID;            /**< Strip coordinate in pitch units. */
float m_cellPosition;      /**< Absolute strip position, -temporary-. */
float m_charge;            /**< Strip signal. */
double m_time;             /**< Time when the sampel was taken. */
```

SVDCluster: A hit reconstructed from strip signals

```
unsigned short m_sensorID; /**< Compressed sensor identifier.*/
bool m_isU;                /**< True if clusters of u-strips, otherwise false. */
float m_position;          /**< Absolute seed strip position in r-phi or z. */
double m_clsTime;          /**< Average waveform maximum time. */
double m_clsTimeSigma;     /**< Standard deviation of waveform maximum times.*/
float m_clsCharge;         /**< Deposited charge in electrons. */
float m_seedCharge;        /**< Cluster seed charge in electrons. */
unsigned short m_clsSize;  /**< Cluster size in pixels */
```

Data objects (cont'd)

VXDTrueHit: Record of a particle passage through a sensor

```
int m_sensorID;              /**< ID of the sensor */
float m_u;                   /**< Local u coordinate when crossing detector plane*/
float m_v;                   /**< Local v coordinate when crossing detector plane*/
float m_entryU;              /**< Local u coordinate when entering silicon */
float m_entryV;              /**< Local v coordinate when entering silicon */
float m_exitU;               /**< Local u coordinate when exiting silicon*/
float m_exitV;               /**< Local v coordinate when exiting silicon*/
float m_energyDep;           /**< Deposited energy while traversing sensor */
float m_globalTime;          /**< Global time. */
TVector3 m_momentum;         /**< momentum in local coordinates when crossing detector plane */
TVector3 m_entryMomentum;    /**< momentum in local coordinates when entering silicon */
TVector3 m_exitMomentum;     /**< momentum in local coordinates when exiting silicon */
```

VxdID: Unique sensor identifier for PXD and SVD

```
union {
      /** Unique id */
unsigned id: Bits;
      struct {
        /** Segment id */
unsigned segment: SegmentBits;
        /** Sensor id */
unsigned sensor: SensorBits;
        /** Ladder id */
unsigned ladder: LadderBits;
        /** Layer id */
unsigned layer: LayerBits;
      } parts /**< Struct to contain all id components */;
    } m_id; /**< Union to store the ID and all components in one go. */
  };
```

What is missing

PXD and SVD

1. Lorentz shift reconstruction (based on local magnetic fields and cluster shapes) - now only very simple versions - we have biases in hit positions. Will improve for the testbeam.
2. Treatment of long clusters - not a priority.

SVD

1. Hit time reconstruction - underway for the testbeam, now only maxima of waveforms.
2. Performance optimization:
   a. change order of processing cycles
   b. 6 samples per SVDDigit and 1D clustering (only after the testbeam)

Thank you for attention.