GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

ATLAS Pixel Testbeam Reconstruction and Analysis

Tobias Bisanz
University of Göttingen
II. Institute of Physics

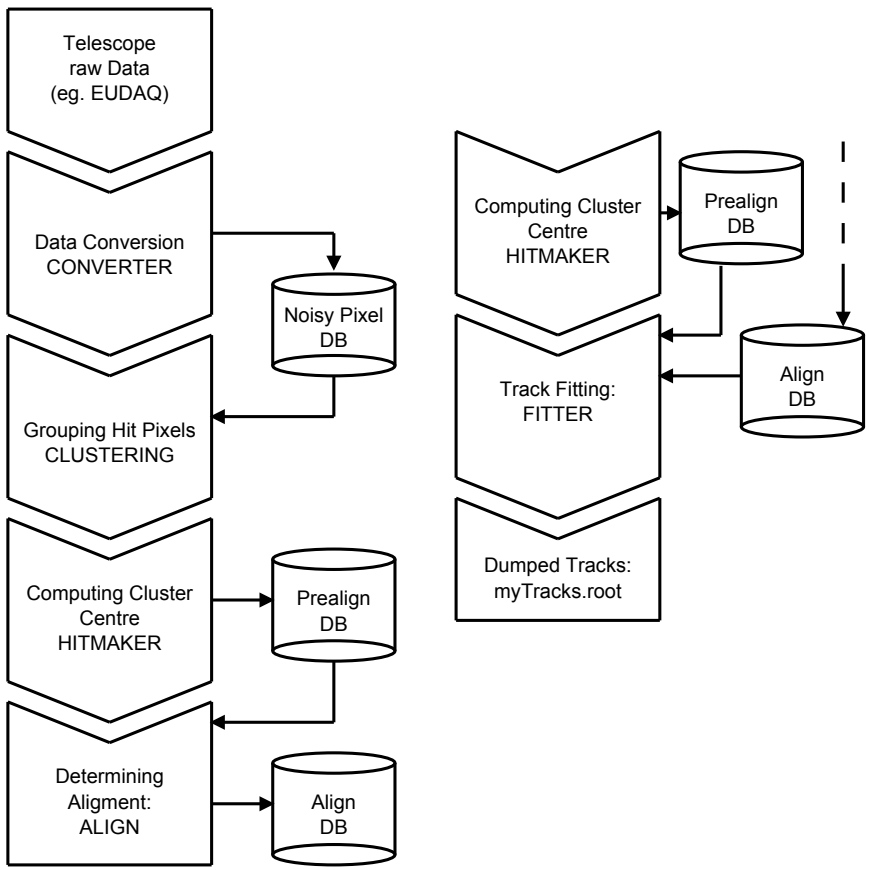
2015-20-01

Bundesministerium
für Bildung
und Forschung

Overview

- New EUTelescope Version
  - Overview
  - Changes
  - Outlook
- TBmonII
  - Overview
  - Verification

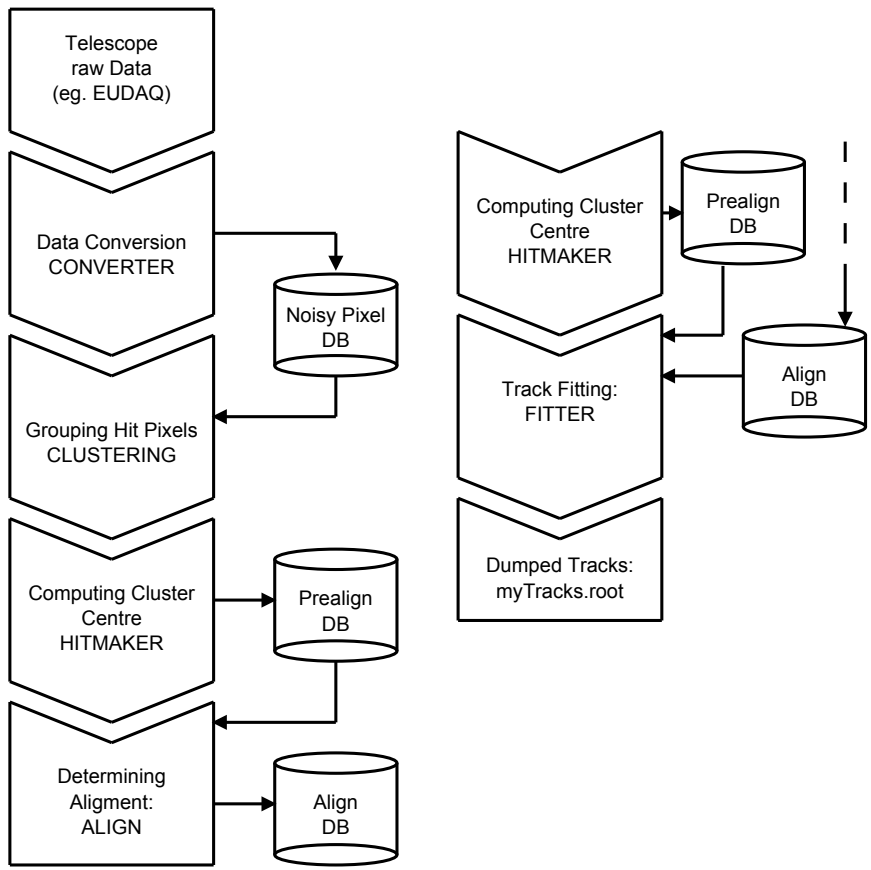
- Converter
  - Convert raw (telescope) data into LCIO
  - Create noisy pixel database (DB)
- Clustering
  - Group together hit pixels
  - Flag/remove clusters with noisy pixel
- Hitmaker
  - Obtain hit position from cluster
  - Determine uncertainty
  - Prealignment
  - (Transform into correct frame of reference)



EUTelescope

Overview II

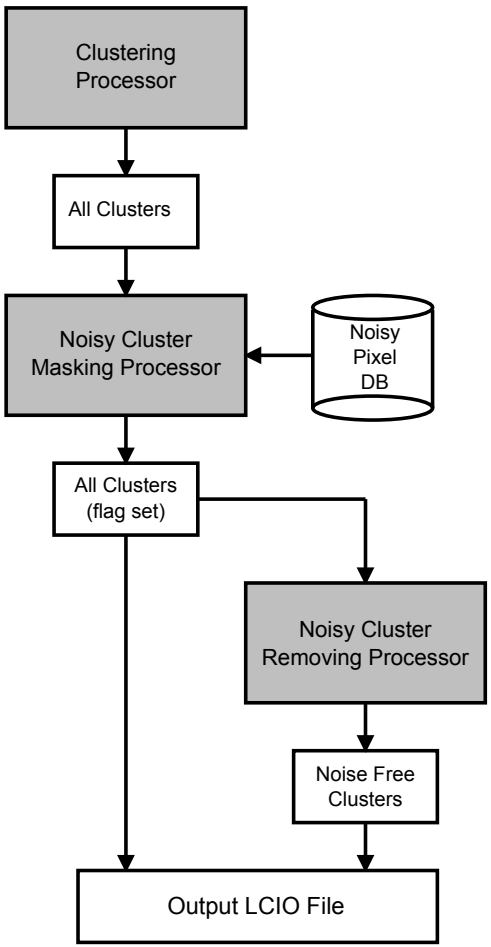
- Align
  - Apply prealignment
  - Using MillepedeII
  - Preliminary track fit and residuals
  - Results stored in alignment database
- Fitter
  - Apply prealignment and alignment
  - Track fitting processor
  - e.g. Deterministic Annealing Filter Fitter (DAF Fitter)
  - or General Broken Line Fitter (GBL Fitter)
  - Dump tracks into root file



Changes to EUTelescope

Previous Modifications

- New geometry description FW
  - ROOT TGeo
- New clustering processor
  - Exploits new geometry
- Modified noisy pixel treatment
- $\rightarrow$ c.f. my talk on the last Telescope WS

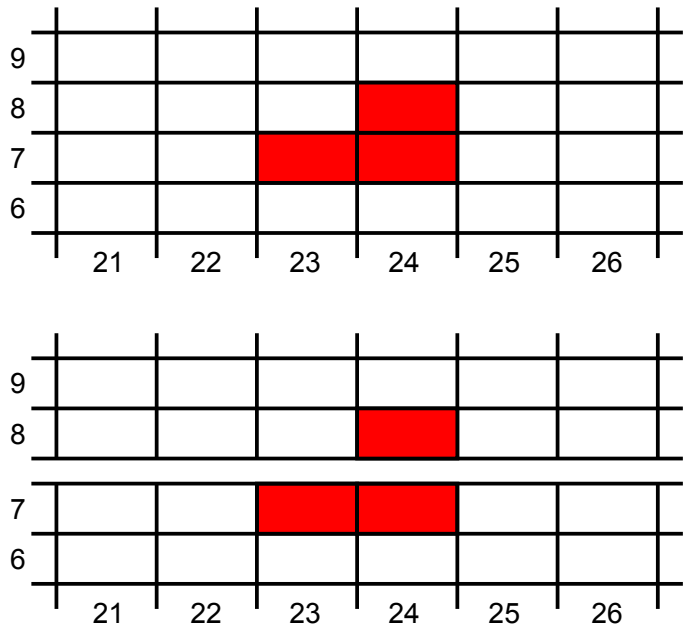


Geometric Clustering

`EUTelProcessorGeometricClustering`

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

- Similar to the previous one
- Only for ZS data
- Proximity requirement is now geometric
- Requires advanced pixel geometry description
- Slower ($\mathcal{O}(2)$)
- Idea: use `EUTelProcessorSparseClustering` and `EUTelProcessorGeometricClustering` in new ZS-data analysis together



Pixel Geometry Framework

`EUTelGenericPixGeoDescr`

- Interface class describing pixel layout
- Needs to be implemented by you
- Relies on ROOT's geometry libraries
- Can be cast from GEAR for simple layouts (default)
- Look into examples provided (Mimosa26, FEI4)
- Look into documentation @ github wiki
- More in tutorial session

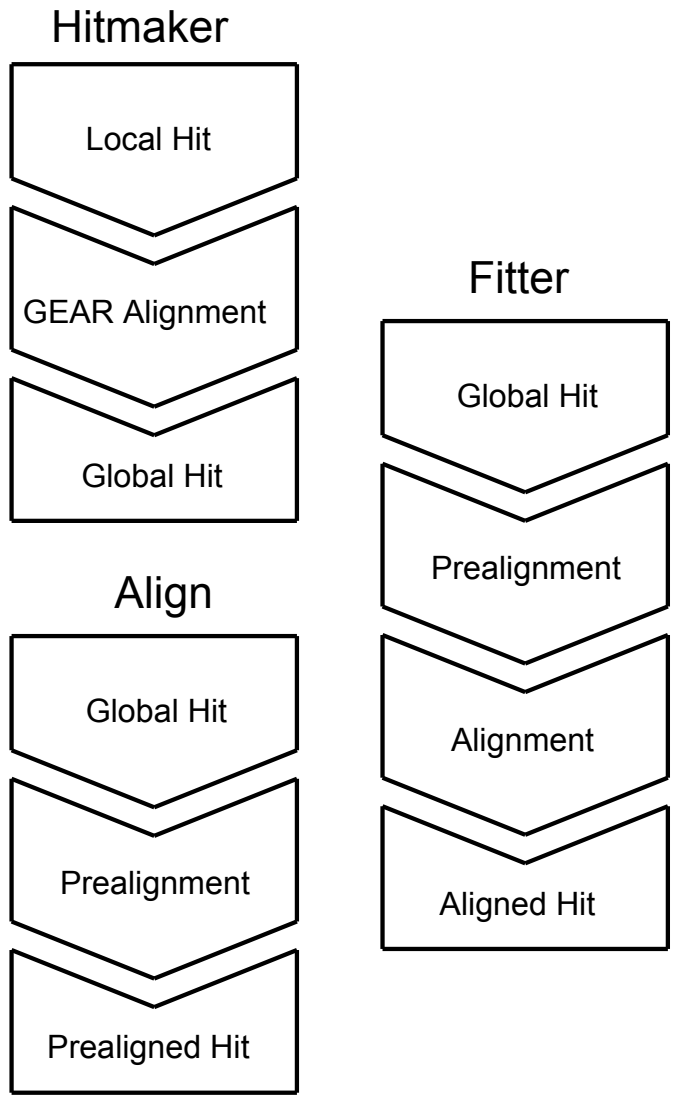
Interface

- Method to create a TGeo representation of your sensor
- Method to get TGeo node from pixel indices
- Method to get indices from node

Changes to EUTelescope

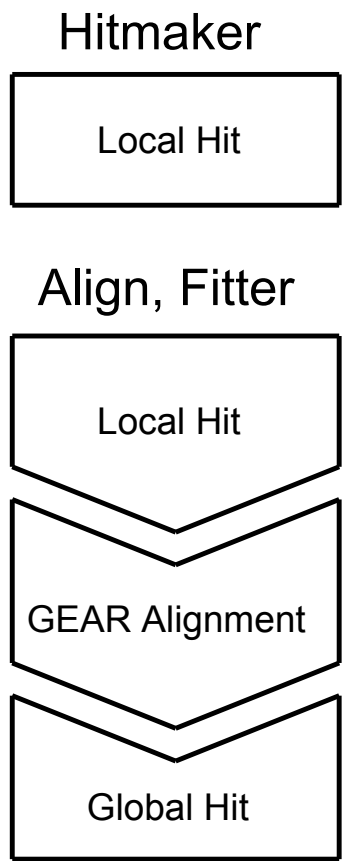
Recent Modifications

- Transition global↔local coordinates
  - Local is in the sensor f.o.r.
  - Global in the telescope f.o.r.
- Versions prior to 1.0 everything global
  - Hitmaker created global hits
- Global hits corrected
  - GEAR offsets
  - Prealignment
  - Alignment
- For every intermediate block on the right:
  - Rotations
  - Shifts



GEAR Based Alignment

- Remove multiple alignment steps
  - Faster
  - More transparent
  - Undoing alignment is easier
- Iteratively update GEAR file
- Store local hits
- In every reco step apply GEAR file

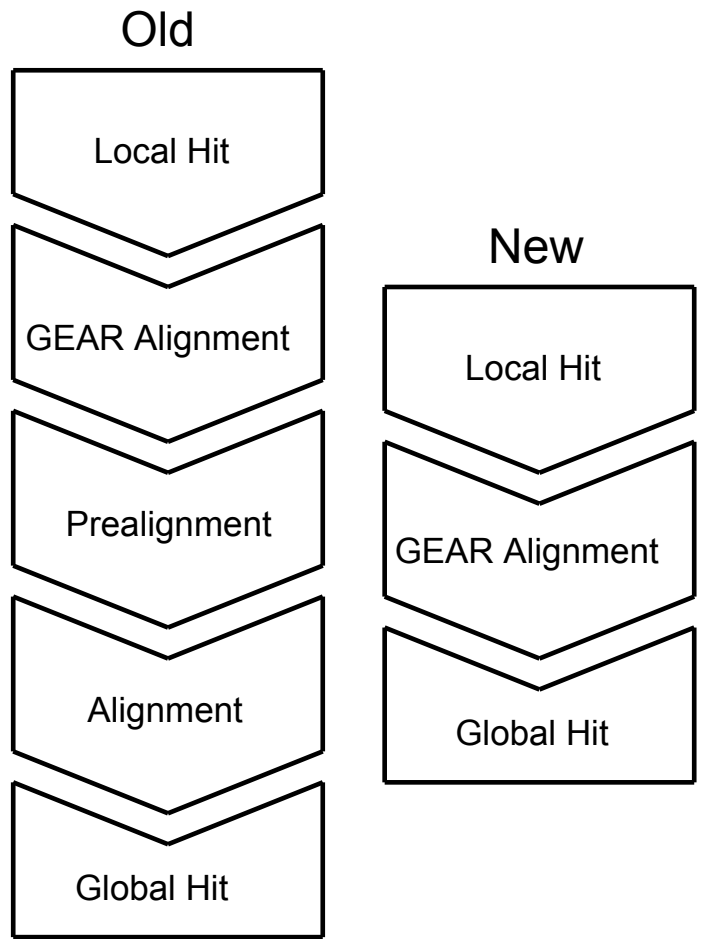


New Track Dumper

`EUTelAPIXTbTrackTuple`

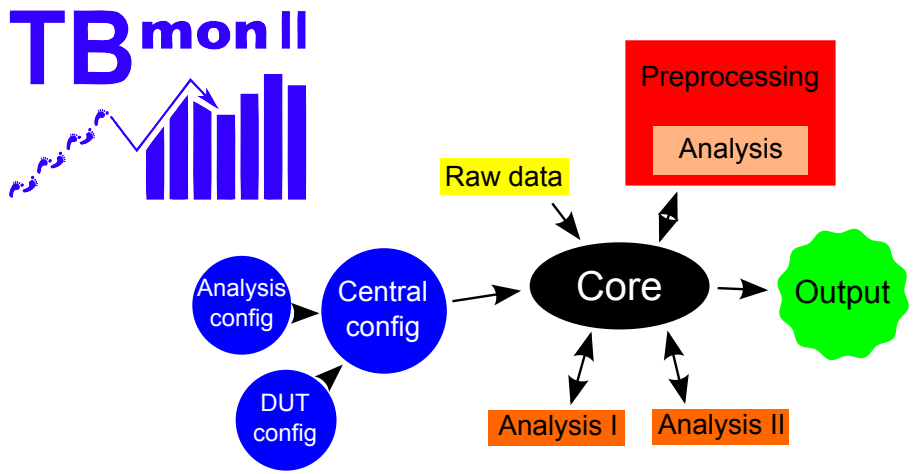
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

- Interface to analysis FW TBmonII
- Expects hits in local f.o.r.
  - Fitted hits are global
  - Undo alignment
  - Dumper does not undo alignment
- Processor to undo alignment needed
- Easy to do with new alignment scheme
  - Invert GEAR alignment and apply
  - Instead of three alignments



TBmonII

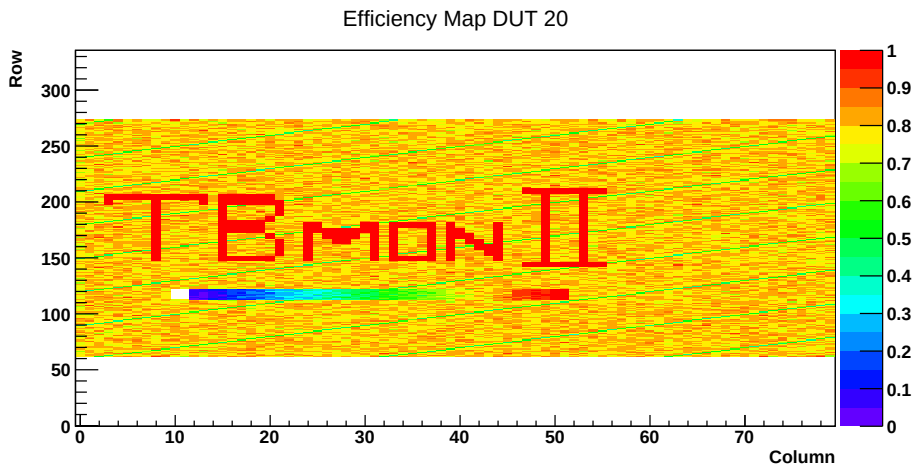
- Offline Analysis Tool
  - ATLAS Pixel default FW
  - Supports multiple analyses
- New version TBmonII
  - Better geometry support
  - Config files make recompiling unnecessary
  - Code fixes and clean-up
- Further Info
  - Lars Graber's talk last WS
  - Contact Marco Bomben
  - `http://bitbucket.org/TBmon2/tbmon2/`



TBmonII

Simple Straight Line Simulation

- Very simple python program
  - Simulates entirely straight tracks
  - One track per event
  - Telescope planes efficiency of 1
  - No noise
- Simulate two standard FE-I4 planes
  - $250 \times 50\,\mu m^2$ pixels
  - Reference plane efficiency of 1
  - Other plane 0.8 with pattern
- Verification of
  - Recent EUTelescope changes
  - EUTelescope $\leftrightarrow$ TBmonII interface
  - TBmonII analysis



GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

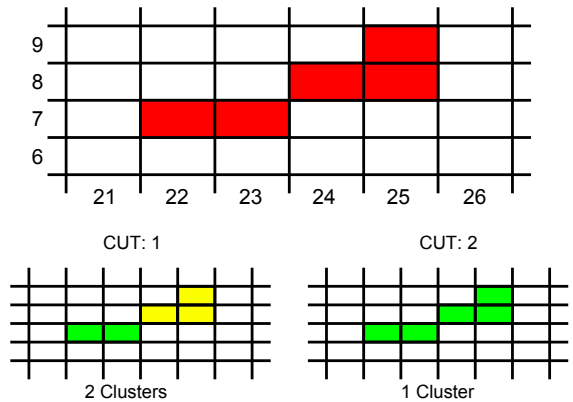
Thank you for your attention.

Backup

New Clustering

Modifications to the old clustering

- Old clustering still there and working
- But **we strongly discourage it**s use!
- If you have non-ZS-data, you have to introduce ZS for the new clustering
- New clustering processors do not care about noisy pixels → new noisy pixel management
- Small, versatile and easy to understand processor(s)



Sparse Cut Distance

Is now measured in distance squared (in units of pixel index, i.e. column, row), e.g. instead of specifying $\sqrt{2}$ for touching, you now have to set 2. This is also the default value, the variable which specifies this in the steering template is: `SparseMinDistanceSquared`.

Sparse Clustering

EUTelProcessorSparseClustering

- Standalone sparse clustering
- Does same as sparse clustering in old processor
- Without S/N cuts
- No noisy pixel information is used
- Very few input parameters (see right)
- Additional (optional) time cut
- On a side note: new pixel type(s) in EUTelescope
- No need for pedestal and noise values (e.g. no call of `EUTelAutoPedestalNoiseProcessor`)

Steering-template variables

For `EUTelProcessorSparseClustering`:

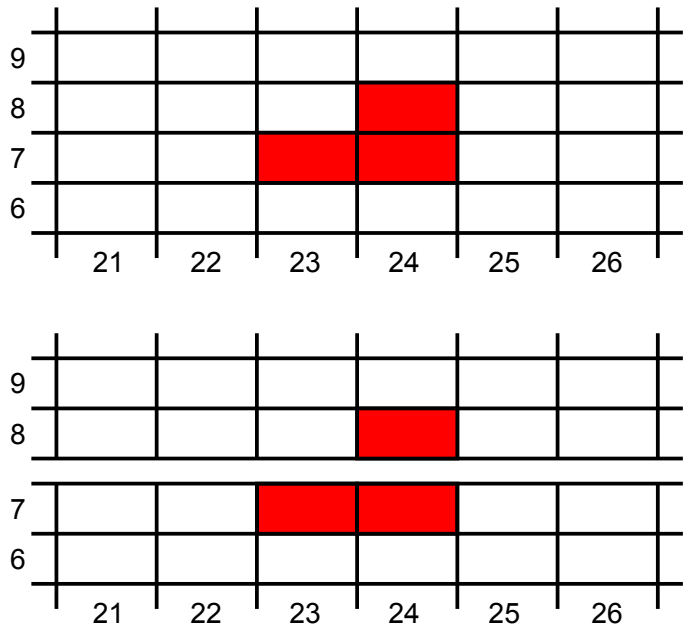
```
PulseCollectionName (string)
ZSDataCollectionName (string)
SparseMinDistanceSquared (int)
TCut (float)
```

Geometric Clustering

`EUTelProcessorGeometricClustering`

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

- Similar to the previous one
- Only for ZS data
- Proximity requirement is now geometric
- Requires advanced pixel geometry description
- Slower ($\mathcal{O}(2)$)
- Idea: use `EUTelProcessorSparseClustering` and `EUTelProcessorGeometricClustering` in new ZS-data analysis together



Pixel Geometry Framework

`EUTelGenericPixGeoDescr`

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

- Interface class describing pixel layout
- Needs to be implemented by you
- Relies on ROOT's geometry libraries
- Can be cast from GEAR for simple layouts (default)
- Look into examples provided (Mimosa26, FEI4)
- Look into documentation @ github wiki
- More in tutorial session

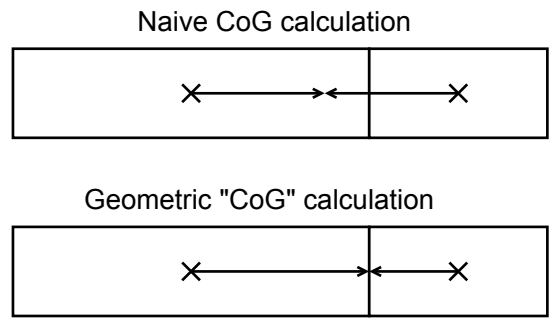
Under Construction

This feature is in an early stage, changes will happen!

Interface

- Method to create a TGeo representation of your sensor
- Method to get TGeo node from pixel indices
- Method to get indices from node

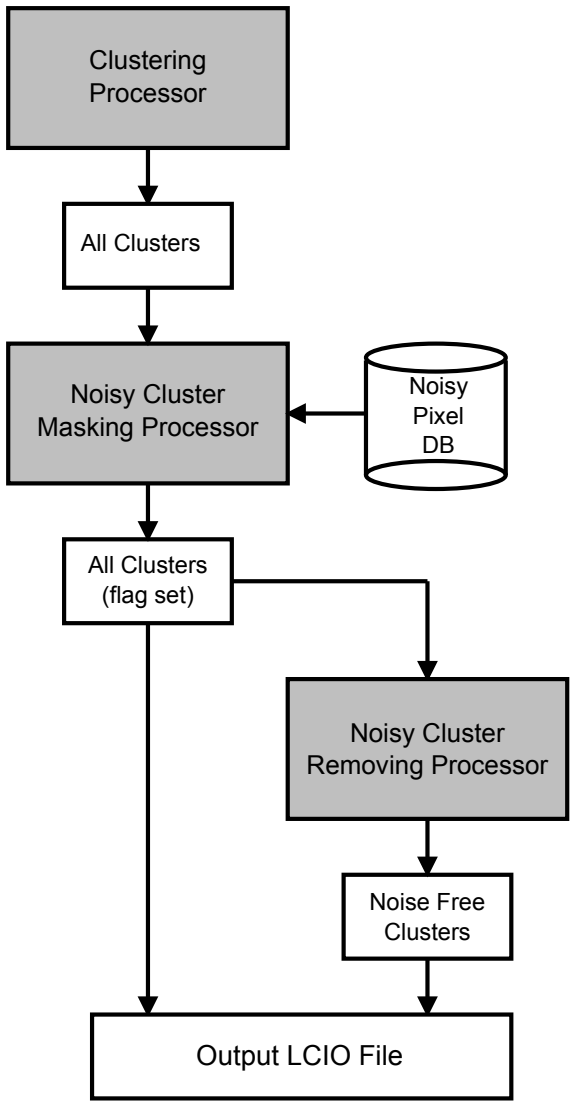
- Other advantage of new geometry description: differently sized pixels
- Example:
  ATLAS FEI4-double-chip modules
  - Prolonged pixels in centre
- Correct determination of pixel edges
- Pixel dimension stored in new `EUTelGeometricPixel` class



Noisy Pixel Treatment

The concept

- Not every processor should check if a cluster/hit contains a noisy pixel
- Most processors should never read noisy pixel DB
- Instead: small lightweight processor to process noisy pixels
- Manipulate data stream this way
- Example on right shows:
  - Clusters from clustering get marked
  - Noise free collection is derived from this



Noisy Pixel Treatment

The implementation

- Processor for setting the noisy property of clusters: `EUTelProcessorNoisyClusterMasker`
- Processor creating a new, noisy cluster free cluster collection: `EUTelProcessorNoisyClusterRemover`
- Reconstruction like this:

```
<processor name="LoadHotPixelDB"/>
<processor name="Clustering"/>
<processor name="NoisyCluMasker"/>
<processor name="NoisyCluRemover"/>
```

Steering-template variables

```
EUTelProcessorNoisyClusterMasker:
HotPixelCollectionName (string)
InputCollectionName (string)

EUTelProcessorNoisyClusterRemover:
InputCollectionName (string)
OutputCollectionName (string)
```

Results
For ATLAS FEI4-four-chip data

- ATLAS FEI4-four-chip modules feature prolonged pixels as well as dead area (bottom left)
- Thus makes use of all the new features introduced
- Hitmap of hit positions in the local frame of reference shown on the right