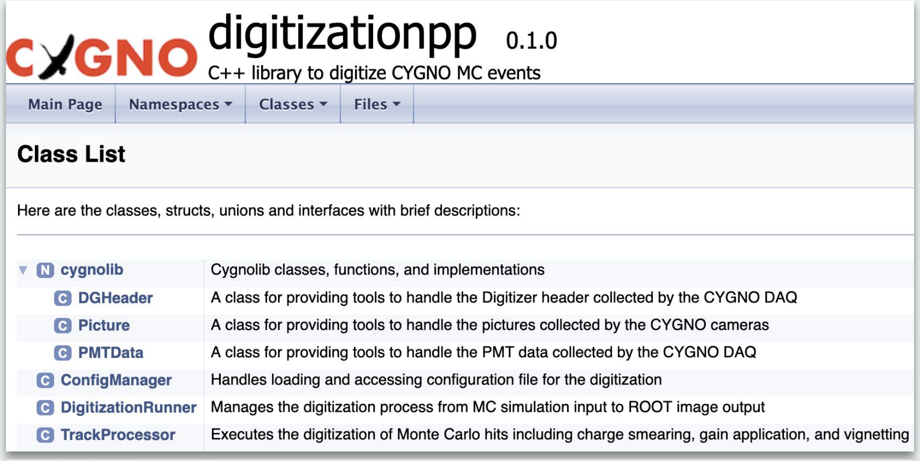
Optimization of digitization parameters

To-do list proposal



digitizationpp 0.1.0

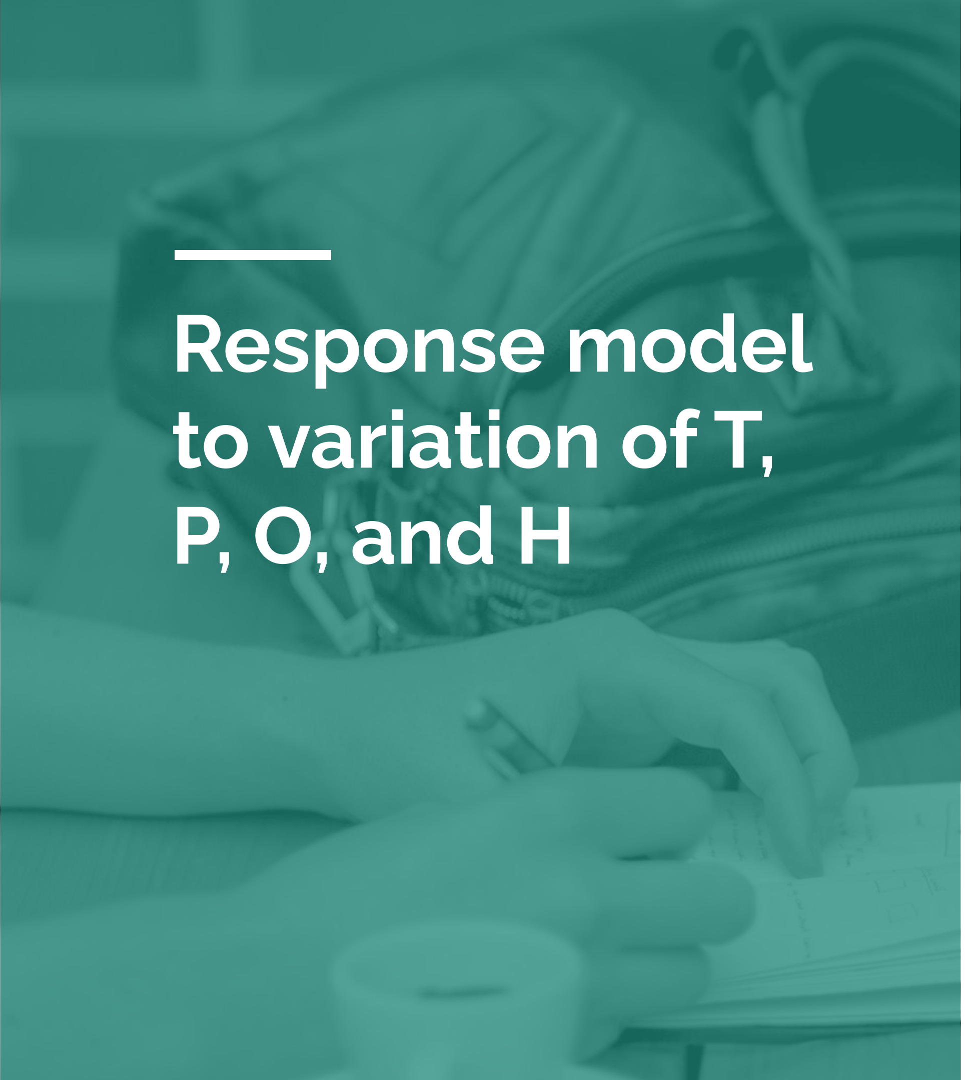
C++ library to digitize CYGNO MC events

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

▼ N cygnolib	Cygnolib classes, functions, and implementations
C DGHeader	A class for providing tools to handle the Digitizer header collected by the CYGNO DAQ
C Picture	A class for providing tools to handle the pictures collected by the CYGNO cameras
C PMTData	A class for providing tools to handle the PMT data collected by the CYGNO DAQ
C ConfigManager	Handles loading and accessing configuration file for the digitization
C DigitizationRunner	Manages the digitization process from MC simulation input to ROOT image output
C TrackProcessor	Executes the digitization of Monte Carlo hits including charge smearing, gain application, and vignetting

Stefano Piacentini - 04 / 09 / 2025



Response model to variation of T, P, O, and H

Goal: reproduce at the digitization level the response curves to 55Fe observed in daily calibrations

- Response curves depend on the conditions of temperature (**T**), pressure (**P**), oxygen conc. (**O**), and humidity (**H**).
- Crucial for a refined data-MC comparison.
- One of the missing steps in the digitization framework

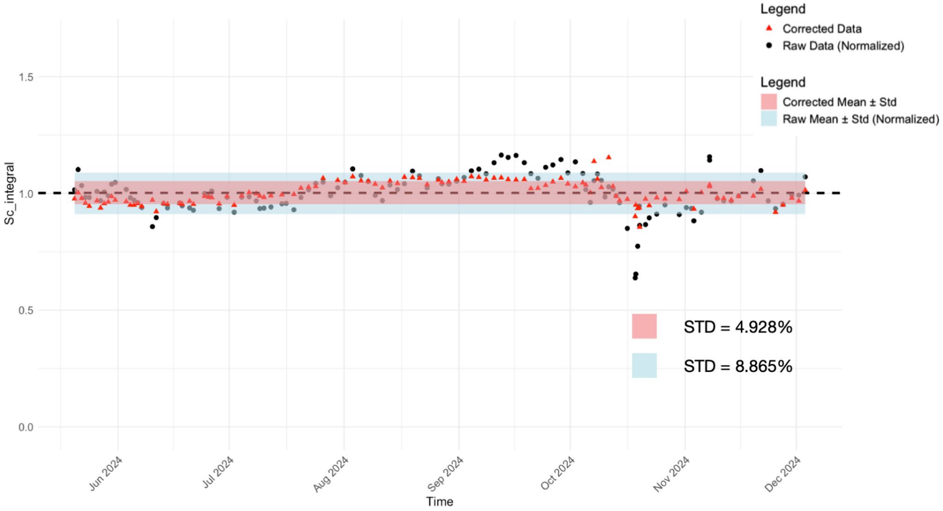
What we have today

- **Calibration module by Melba:** for each 55Fe position **i** in z: fitting procedure → a set of 5 parameters $\boldsymbol{\theta}(\mathbf{z_i})$ for each position. Note: based on calibration measurements → the convolution of all the detector response effects (electron attachment, charge quenching, light quenching, etc.)

- **Digitization code:** two parameters → the gain G of each GEM and the electron attachment lifetime λ

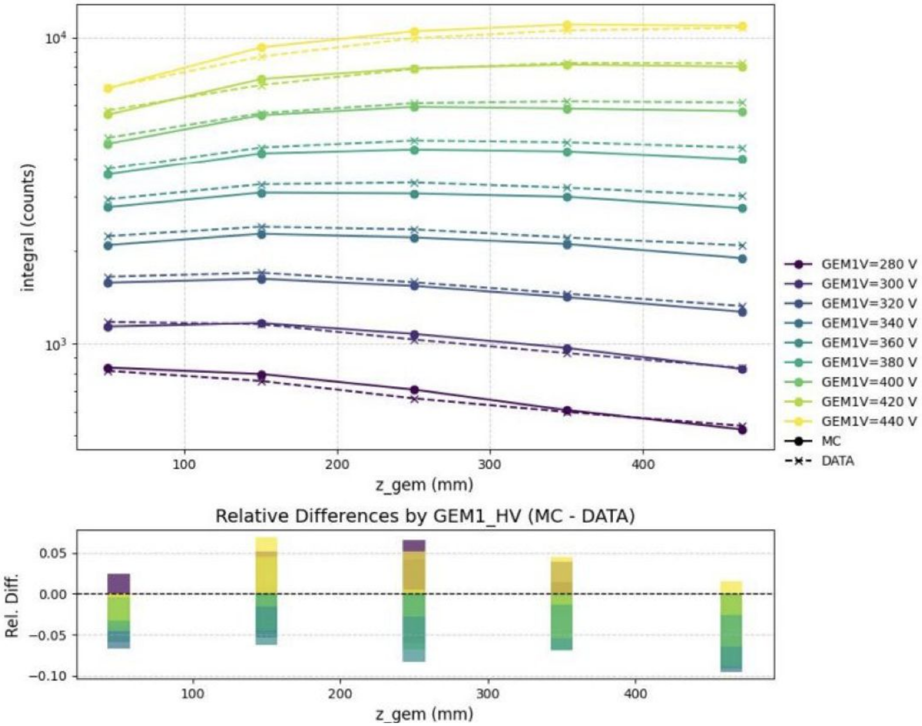
Melba's results at Step 3

The fitting procedure by Melba is able to predict the LY as function of T, P, O, and H with an accuracy below 5%



Pietro’s response curves

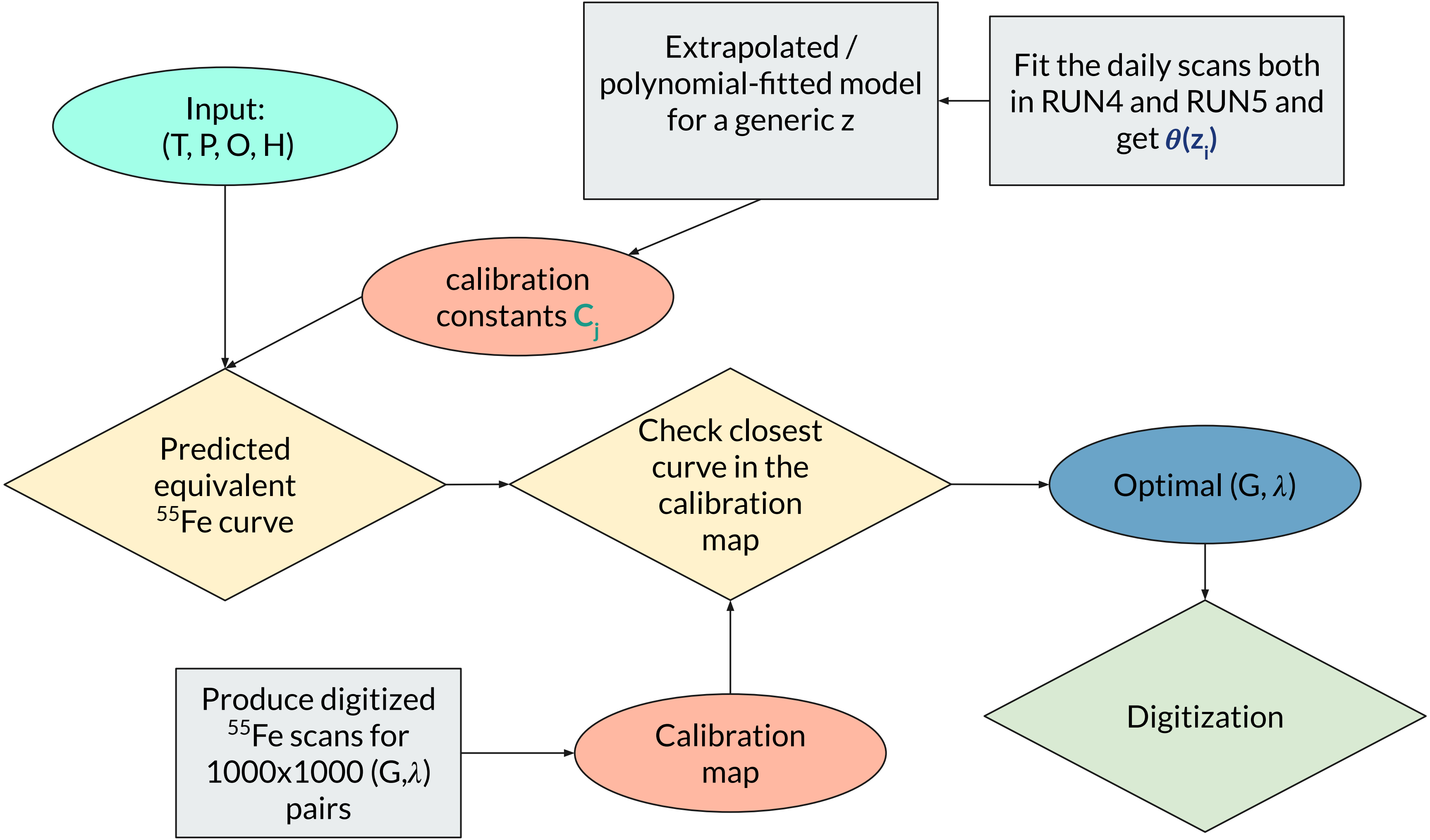
With opportune fine tuning we are able to find a pair (G, λ) to reproduce the observed data, but not stable with time (aka T, P, O, and H)



Suggested proposal

1. Implement Melba's model into the digitization code: for any input of (T, P, O, H) and for every z [e.g. interpolation / polynomial fit to $\theta(z_i)$], predict a ^{55}Fe curve via the set of calibration constants C_j produced by the calibration fit.
2. Prepare a calibration map to associate any set of (T, P, O, H) - defined by the user as an input to the digitization - to a pair of (G, λ) .

G	λ	^{55}Fe Step1	^{55}Fe Step2	^{55}Fe Step3	^{55}Fe Step4	^{55}Fe Step5
1.000	1.000	//	//	//	//	//
1.000	1.001	//	//	//	//	//



Input: (T, P, O, H)

Extrapolated / polynomial-fitted model for a generic z

Fit the daily scans both in RUN4 and RUN5 and get $\boldsymbol{\theta}(z_i)$

calibration constants C_j

Predicted equivalent ^{55}Fe curve

Check closest curve in the calibration map

Optimal (G, λ)

Produce digitized ^{55}Fe scans for 1000x1000 (G,λ) pairs

Calibration map

Digitization

Needs for maps production

- Automated way to send digitization jobs on cloud ✓
- Generate digitization files **without pedestals** → **reconstruction with random pedestals** **[missing, but recent work on it]**
- Automated way to reconstruct digitization output ✓
- Automated fitting procedure to assess the response curve for each (G, λ) pair **[missing]**
- Construction and historization of the map → table on cloud? **[missing]**

Needs for calibration constant evaluation

- Fitting procedure for ^{55}Fe prediction at each step ✓
- Extend what has been done by Melba for Step 3 and RUN5 to all the calibration steps and to RUN4 **[missing]**
- Construction and historization of calibration files → table on cloud? **[missing]**

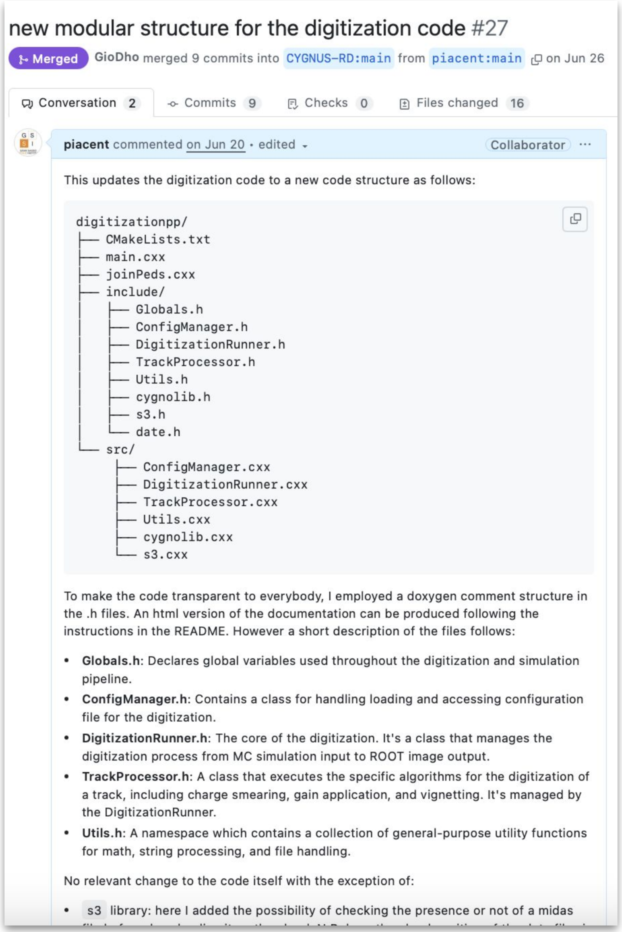
Needs for digitization code

- Implement a function in the digitization to reproduce the calibration model **[missing]**
- Implement a function to find the best (G,λ) on the basis of T, P, H, O input → what is the best way to compute the difference between two curves? Chi2? **[missing]**

- Ideal: extract (T,P,H,O) on the basis of input distribution (e.g. distribution in N days of data) event by event **[missing]**

Recent updates on digitization code

- Complete refurbishment of the code structure [pull request #27]



new modular structure for the digitization code #27

Merged
GioDho merged 9 commits into CYGNUS-RD:main from piacent:main on Jun 26
Conversation 2
Commits 9
Checks 0
Files changed 16
piacent commented on Jun 20 • edited
Collaborator

This updates the digitization code to a new code structure as follows:

```
digitizationpp/
├── CMakeLists.txt
├── main.cxx
├── joinPeds.cxx
├── include/
│   ├── Globals.h
│   ├── ConfigManager.h
│   ├── DigitizationRunner.h
│   ├── TrackProcessor.h
│   ├── Utils.h
│   ├── cygnolib.h
│   ├── s3.h
│   └── date.h
└── src/
     ├── ConfigManager.cxx
     ├── DigitizationRunner.cxx
     ├── TrackProcessor.cxx
     ├── Utils.cxx
     ├── cygnolib.cxx
     └── s3.cxx
```

To make the code transparent to everybody, I employed a doxygen comment structure in the .h files. An html version of the documentation can be produced following the instructions in the README. However a short description of the files follows:

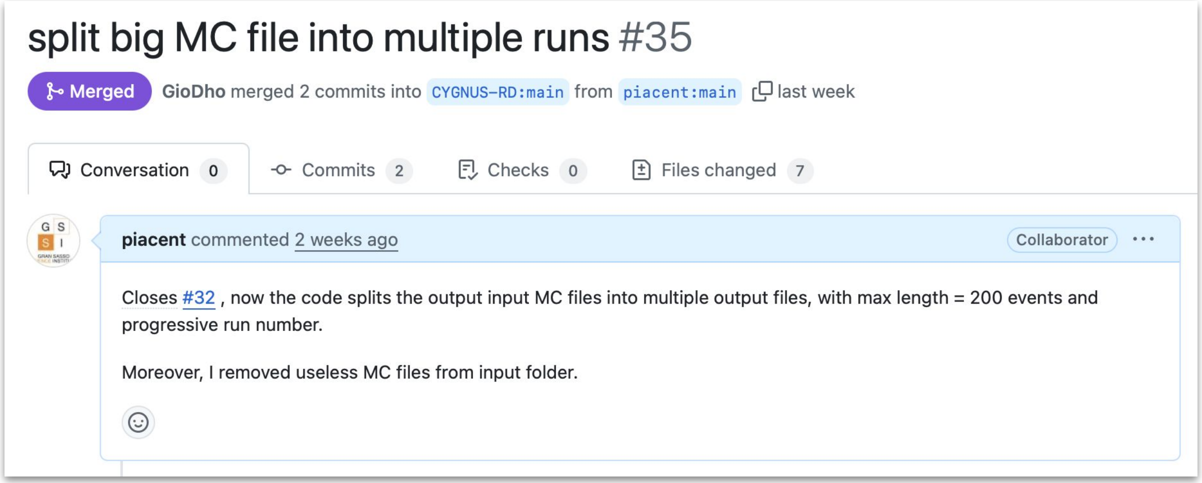
- **Globals.h**: Declares global variables used throughout the digitization and simulation pipeline.
- **ConfigManager.h**: Contains a class for handling loading and accessing configuration file for the digitization.
- **DigitizationRunner.h**: The core of the digitization. It's a class that manages the digitization process from MC simulation input to ROOT image output.
- **TrackProcessor.h**: A class that executes the specific algorithms for the digitization of a track, including charge smearing, gain application, and vignetting. It's managed by the DigitizationRunner.
- **Utils.h**: A namespace which contains a collection of general-purpose utility functions for math, string processing, and file handling.

No relevant change to the code itself with the exception of:

- `s3` library: here I added the possibility of checking the presence or not of a midas

Recent updates on digitization code

- Split large MC input files into smaller digitized files [pull request #35]



split big MC file into multiple runs #35

Merged GioDho merged 2 commits into `CYGNUS-RD:main` from `piacent:main` last week

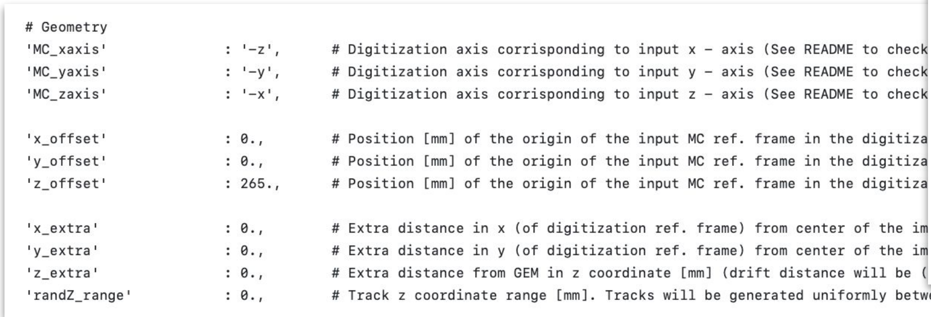
piacent commented 2 weeks ago

Closes #32 , now the code splits the output input MC files into multiple output files, with max length = 200 events and progressive run number.

Moreover, I removed useless MC files from input folder.

Recent updates on digitization code

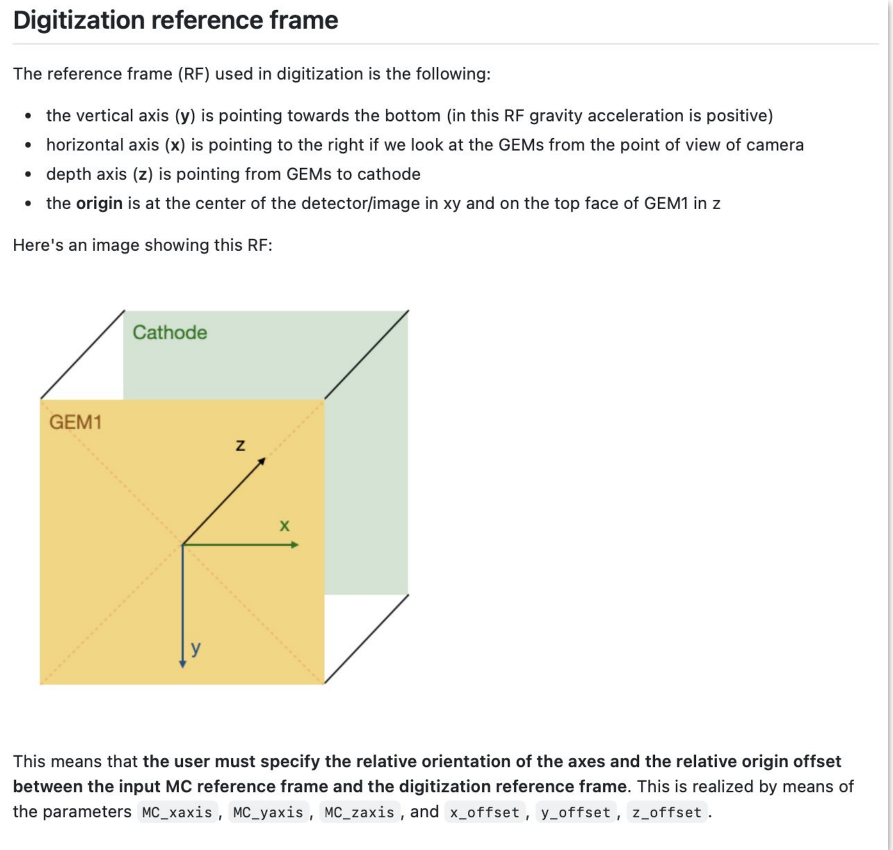
- User defined custom reference frame [pull requests #36, #37, #41]



```
# Geometry
'MC_xaxis'               : '-z',     # Digitization axis corrisponding to input x - axis (See README to check
'MC_yaxis'               : '-y',     # Digitization axis corrisponding to input y - axis (See README to check
'MC_zaxis'               : '-x',     # Digitization axis corrisponding to input z - axis (See README to check

'x_offset'               : 0.,       # Position [mm] of the origin of the input MC ref. frame in the digitiza
'y_offset'               : 0.,       # Position [mm] of the origin of the input MC ref. frame in the digitiza
'z_offset'               : 265.,     # Position [mm] of the origin of the input MC ref. frame in the digitiza

'x_extra'                : 0.,       # Extra distance in x (of digitization ref. frame) from center of the im
'y_extra'                : 0.,       # Extra distance in y (of digitization ref. frame) from center of the im
'z_extra'                : 0.,       # Extra distance from GEM in z coordinate [mm] (drift distance will be (
'randZ_range'            : 0.,       # Track z coordinate range [mm]. Tracks will be generated uniformly between 'z_hit+randZ_range/2' and 'z_hit-randZ_range/2'
```



Digitization reference frame

The reference frame (RF) used in digitization is the following:

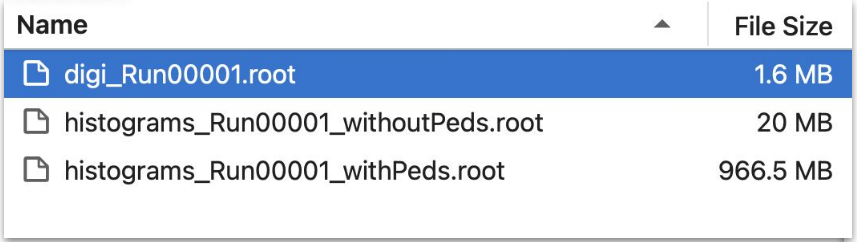
- the vertical axis (**y**) is pointing towards the bottom (in this RF gravity acceleration is positive)
- horizontal axis (**x**) is pointing to the right if we look at the GEMs from the point of view of camera
- depth axis (**z**) is pointing from GEMs to cathode
- the **origin** is at the center of the detector/image in xy and on the top face of GEM1 in z

Here's an image showing this RF:

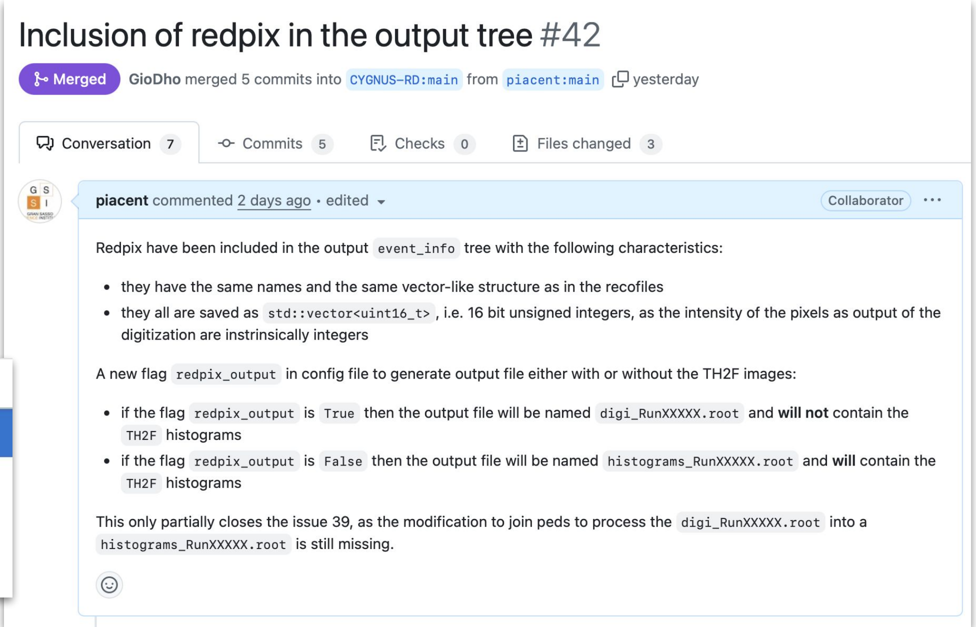
This means that **the user must specify the relative orientation of the axes and the relative origin offset between the input MC reference frame and the digitization reference frame**. This is realized by means of the parameters `MC_xaxis`, `MC_yaxis`, `MC_zaxis`, and `x_offset`, `y_offset`, `z_offset`.

Recent updates on digitization code

- Redpix in the output tree of digitization file [pull request #42]
- Possibility to not create TH2F images in the output file to save disk space [pull request #42]:



Name	File Size
digi_Run00001.root	1.6 MB
histograms_Run00001_withoutPeds.root	20 MB
histograms_Run00001_withPeds.root	966.5 MB



Inclusion of redpix in the output tree #42

Merged GioDho merged 5 commits into `CYGNUS-RD:main` from `piacent:main` yesterday

piacent commented 2 days ago • edited

Redpix have been included in the output `event_info` tree with the following characteristics:

- they have the same names and the same vector-like structure as in the recofiles
- they all are saved as `std::vector<uint16_t>`, i.e. 16 bit unsigned integers, as the intensity of the pixels as output of the digitization are instrinsically integers

A new flag `redpix_output` in config file to generate output file either with or without the TH2F images:

- if the flag `redpix_output` is `True` then the output file will be named `digi_RunXXXXX.root` and **will not** contain the `TH2F` histograms
- if the flag `redpix_output` is `False` then the output file will be named `histograms_RunXXXXX.root` and **will** contain the `TH2F` histograms

This only partially closes the issue 39, as the modification to join peds to process the `digi_RunXXXXX.root` into a `histograms_RunXXXXX.root` is still missing.

References

digitizationpp github page

Pietro’s recap presentation at the simulation meeting

Melba’s presentation at the analysis meeting

Melba’s presentation at the analysis meeting - last