Discussion on reconstruction status & plans

E. Di Marco

CYGNO reco and analysis meeting,
28 October 2021

Code branches

- On our GitHub repository https://github.com/CYGNUS-RD/reconstruction we have two active branches:

1. "lime21" (Default). This uses the reconstruction method used for LEMON and tuned a bit for the geometry/noise/light-yield of LIME

   - I left this as default because it is the only one with which we had at least once agreed

   - this is NOT OPTIMAL for LIME overground and 50ns camera aperture, because with the "ambient radioactivity" pileup tends to merge multiple long tracks

2. "autumn21" (not official-notagreed-unstable-forbraveharts). This uses the directional clustering by I. Pains to remove the long tracks, and a modified DBSCAN for the residual pixels

   - this works reasonably for LIME 50ms, it is under development, can be further tuned, it is slow, but usable (reconstrcuted ALL LIME data of April and July in one day @LNGS)

A reminder

Both "lime21" and "autumn21" use something complicatd to cope with pileup of cosmics to make the OVERGROUND data taken with LIME of any use.

- LIME – CYGNO – whatever underground will have a much smaller pileup, so none of these will be needed. Probably something as simple as naive DBSCAN or NNC are sufficient (Occam RAZOR !)

- Other prototypes have different background? Then maybe another approach can be more optimal

For example, for MANGO Giorgio changed the supercluster step from GAC (used in "lime21") to another.

- Note: branches are for free, so why not push that in the official repo? No Pull Request done so far.

Modularity

The main steps of the reconstruction are modular, and algorithms of each step can be modified w/o touching the others:

1. Noise suppression and filtering

2. clustering

3. evaluation of cluster shapes

4. energy corrections

5. reconstruction of the PMT waveform

6. tree dumping

- In particular: the clustering spits out a Collection of cluster objects. The following code is agnostic of the clustering method, so one can compute your preferred cluster shapes for any method w/o changing a line of code.

Integration

So far in any branch on github there are only the variables that I computed, because no pull-requests with new variables have been done.

Adding variables that are useful or might be useful is not dangerous: we have o(100) clusters / image, and adding few floats / cluster is ~for free.

- with LIME occupancy, trees are 4.7 kB / event => peanuts

- So I invite people that have developed variables to:

  - either resolve the possible conflicts with any of the existing branch and make a pull-request to integrate new variables

  - just push a new branch in github, so at least is in a central, collaboration-wide place for others to use it