Weekly meeting

Miscellaneous updates

Jérémie Decock

CEA Saclay - Irfu/SAp

September 7, 2016

Calibration

Fix the calibration implementation

Pyhessio bug fix

- Error was due to get_pedestal fonction in pyhessio/src/pyhessio.c
- Loopingt over pixels then channels instead looping over channels then pixels as required
- Fixed by Jean Jacquemier
- I checked on images
- See `https://github.com/cta-observatory/pyhessio/issues/34` (bug report)

Fix the calibration implementation

Calibration bug fix

Now it use the right channel, not only the first one...

```
85      peds, gains = get_mc_calibration_coeffs(tel_id)
86
87      peds_ch0 = peds[0]
88      peds_ch1 = peds[1]
89
90      calibrated_image = [ (adc0 - ped0) * gain0 if adc0
         < adc_treshold else (adc1 - ped1) * gain1
91                           for adc0, adc1, ped0, ped1,
       gain0, gain1
92                           in zip(adcs[0], adcs[1], peds
       [0], peds[1], gains[0], gains[1]) ]
```

listings/extract_crop_and_plot_all_astri_images.py

Miscellaneous notes

- Improve the script used to generate FITS files (Calibrated image + PE image): much faster
- Re-generate FITS files for ASTRI events for Gammas and Protons with the right pedestal and fixed calibration

Benchmark

Former benchmark method

The mean distance of normalized images

The error function $\mathcal{E}$ is given by:

$$\mathcal{E}(\hat{\mathbf{s}}, \mathbf{s}^*) = \text{mean}\left(\text{abs}\left(\varphi(\hat{\mathbf{s}}) - \varphi(\mathbf{s}^*)\right)\right)$$

Where:

- $\hat{\mathbf{s}}$ is the output image (the "cleaned" image) $\in \mathbb{R}^d$
- $\mathbf{s}^*$ is the reference image (the "clean" image) $\in \mathbb{R}^d$
- φ is a normalization function

$$\varphi(\mathbf{s}) = \frac{\mathbf{s} - \min(\mathbf{s})}{\max(\mathbf{s}) - \min(\mathbf{s})}$$

New benchmark method

The error function $\mathcal{E}$ is given by:

$$\mathcal{E}(\hat{\mathbf{s}}, \mathbf{s}^*) = (\mathcal{E}_1(\hat{\mathbf{s}}, \mathbf{s}^*), \mathcal{E}_2(\hat{\mathbf{s}}, \mathbf{s}^*))^T$$

$$\mathcal{E}_1(\hat{\mathbf{s}}, \mathbf{s}^*) = \text{mean}\left(\text{abs}\left(\frac{\hat{\mathbf{s}}}{\sum_i \hat{\mathbf{s}}_i} - \frac{\mathbf{s}^*}{\sum_i \mathbf{s}^*_i}\right)\right)$$

$$\mathcal{E}_2(\hat{\mathbf{s}}, \mathbf{s}^*) = \frac{\text{abs}\left(\sum_i \hat{\mathbf{s}}_i - \sum_i \mathbf{s}^*_i\right)}{\sum_i \mathbf{s}^*_i}$$

Where:

- $\hat{\mathbf{s}}$ is the output image (the "cleaned" image) $\in \mathbb{R}^d$
- $\mathbf{s}^*$ is the reference image (the "clean" image) $\in \mathbb{R}^d$
- i is the index of a PMT (i.e. of a pixel) within an image

Optimization

Work in progress...
I started with scipy but I will probably switch to my own implementation.

Library

We started to merge Tino's code with mine to make a consistent unified library: `https://github.com/jdhp-sap/data-pipeline-standalone-scripts` (the name and location of this library may change in the future as it's purpose slightly changed since this merge).

- Tino's scripts: `https://github.com/tino-michael/tino_cta` (reconstruction and classification)
- My scripts to manage simtel files: `https://github.com/jdhp-sap/snippets`
- My image cleaning scripts: `https://github.com/jdhp-sap/data-pipeline-standalone-scripts`

Conclusion

TODO

TODO

TODO

References I