Muon Calibration in CTApipe

CTA Calibration Meeting , 04/10/17

A. Mitchell, MPIK

cta
cherenkov telescope array

CTA Muon Calibration

- Telescope / Camera teams must:
  - Be able to trigger on incoming muons
  - Flag candidate muon events
  - i.e. perform muon preselection
- Telescope / Camera teams may:
  - Develop algorithms for internal use
  - Perform own simulations e.g. with degraded efficiency

- BUT: all telescopes will be calibrated by the same analysis and calibration chain.
- Muon software of teams beyond pre-selection will NOT be used in standard muon calibration → **ctapipe**

Muon Calibration Algorithms

- Many algorithms exist and are currently in use
- Ideally, one algorithm for all telescopes / cameras
- Which one should be used?
  - The one that performs the best!
- How to know? Compare algorithms on figures of merit in same software framework → ctapipe
- Recommend: If you think an algorithm could be *the one*, it should be implemented in ctapipe

Muon Calibration (reconstruction)

- Flagged candidate muon events enter. Two step procedure:
- Circle fitting
  - Chaudhuri-Kundu analytical solution → *implemented*
  - Chernov-Osokov analytical solution
  - Hough Transform
  - Karimäki chi-square fit
  - Taubin fit
  - Kasa method
- Intensity fitting
  - Chi-square fit to 1D profile
  - Chi-square fit to 2D image
  - Pixel-wise Log-likelihood fit to 2D image → *implemented*
- Per event fit parameters stored

Storage Classes: MuonRingParameter

- Container classes in io/containers.py

- After ring/circle fitting, store parameters:
  - run_id, event_id, tel_id, inputfile
  - ring_center_x, ring_center_y,
  - ring_radius
  - ring_phi, ring_inclination
  - ring_chi2_fit, ring_cov_matrix
  - ring_fit_method

Storage Classes: MuonRingParameter

- Container classes in io/containers.py

- After ring/circle fitting, store parameters:
  - run_id, event_id, tel_id, inputfile
  - ring_center_x, ring_center_y,
  - ring_radius
  - ring_phi, ring_inclination
  - ring_chi2_fit, ring_cov_matrix
  - ring_fit_method
- Example ring fitting:
  - image/muon/muon_ring_finder.py
  - ChaudhuriKunduRingFitter class

Storage Classes: MuonIntensityParameter

- Container classes in io/containers.py

- After intensity fitting, store parameters:
  - run_id, event_id, tel_id, inputfile
  - ring_completeness, ring_num_pixel, ring_size, off_ring_size,
  - ring_width, ring_time_width,
  - intensity_cov_matrix, COG_x, COG_y
  - impact_parameter, impact_parameter_chi2
  - impact_parameter_pos_x, impact_parameter_pos_y
  - optical_efficiency_muon, mask, prediction,
  - intensity_fit_method

Storage Classes: MuonIntensityParameter

- Container classes in io/containers.py

- After intensity fitting, store parameters:
  - run_id, event_id, tel_id, inputfile
  - ring_completeness, ring_num_pixel, ring_size, off_ring_size,
  - ring_width, ring_time_width,
  - intensity_cov_matrix, COG_x, COG_y
  - impact_parameter, impact_parameter_chi2
  - impact_parameter_pos_x, impact_parameter_pos_y
  - optical_efficiency_muon, mask, prediction,
  - intensity_fit_method
- Example intensity fitting:
  - image/muon/muon_integrator.py : MuonLineIntegrate class

Progress so far

- H.E.S.S. algorithm implemented (Chaudhuri-Kundu and 2D log-likelihood) with preliminary tests
- “muonmaster” branch no longer exists → has been merged into “master” branch

- For implementing more methods, **please create a new branch for each method**
  - Easier to keep track of and review changes
  - Cleaner approach, smaller steps, more frequent merging

- Muon_reconstruction example has been adopted into a tool (MuonDisplayerTool)

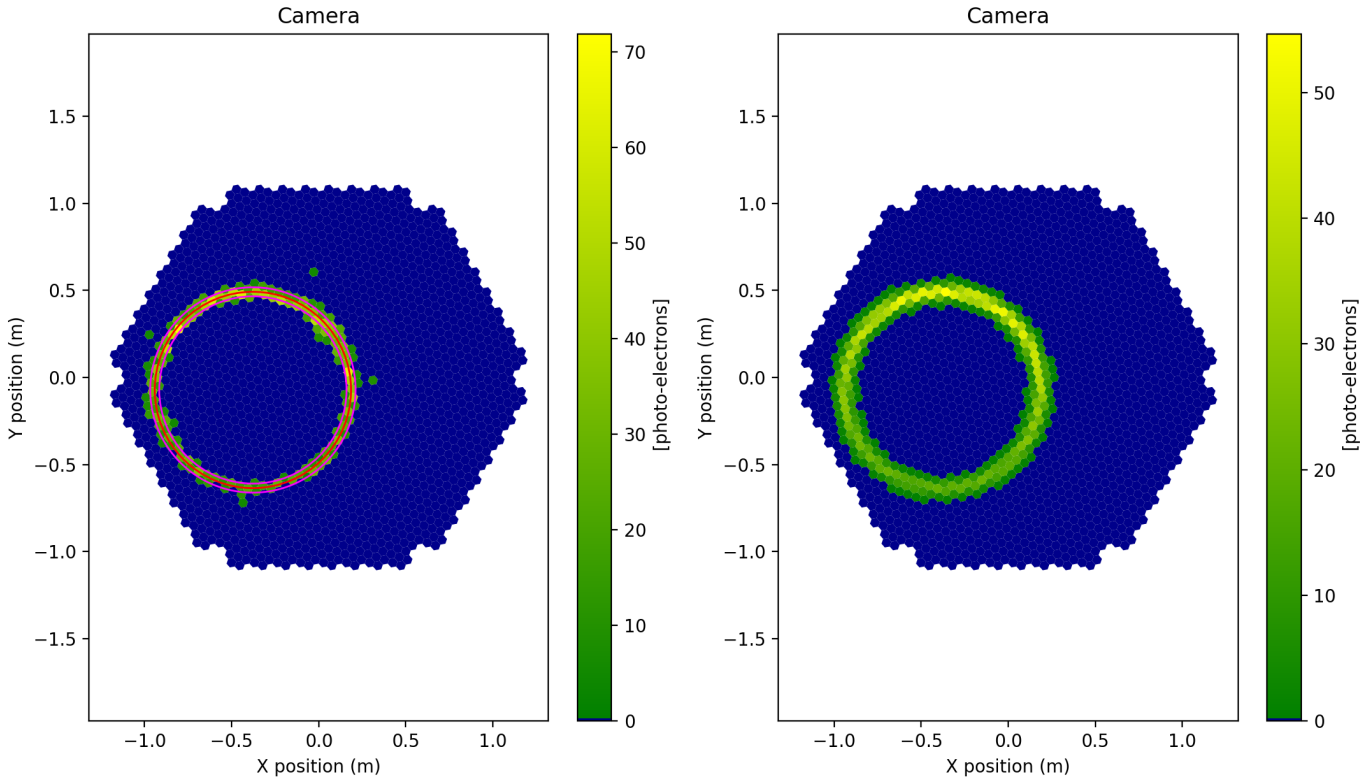
List of relevant files in ctapipe

- examples/muon_reconstruction.py
- image/muon/
  - muon_reco_functions.py
  - muon_ring_finder.py
  - muon_integrator.py
  - muon_diagnostic_plots.py
- & generic classes: intensity_fitter.py & ring_fitter.py

- MC muon simulation test files mst-fc and sst-dc at 100% efficiency in ctapipe-extra repository
  (can add/distribute more xSTs / efficiencies if there is interest → would need to as Karl Kosack)

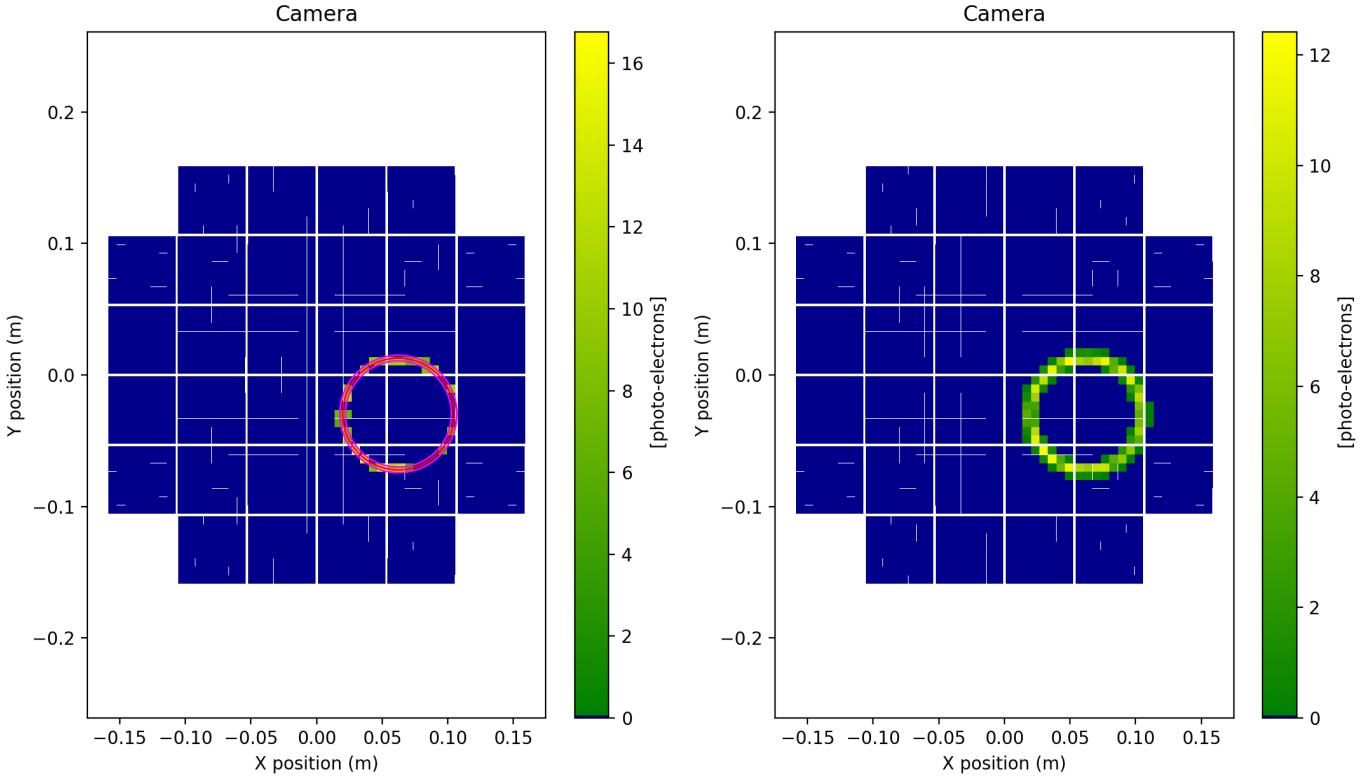
Muon Image Fitting

- Circle and width fit, intensity expectation



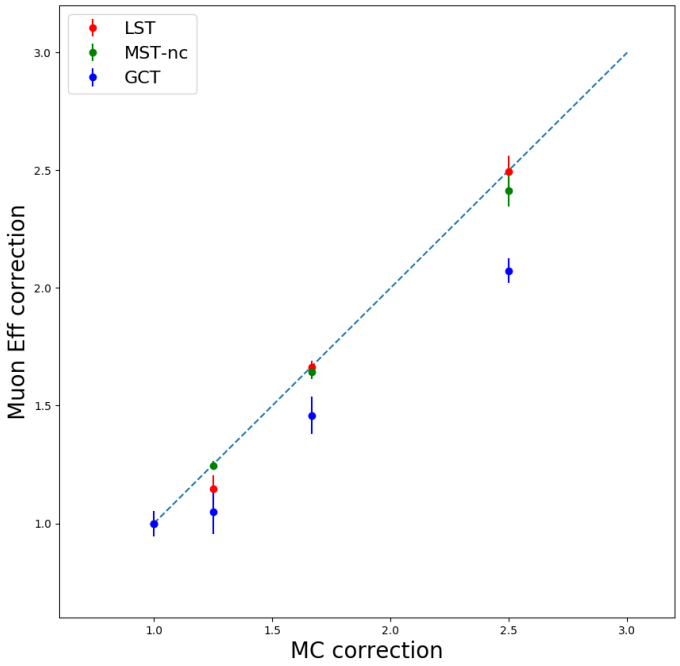
Muon Image Fitting

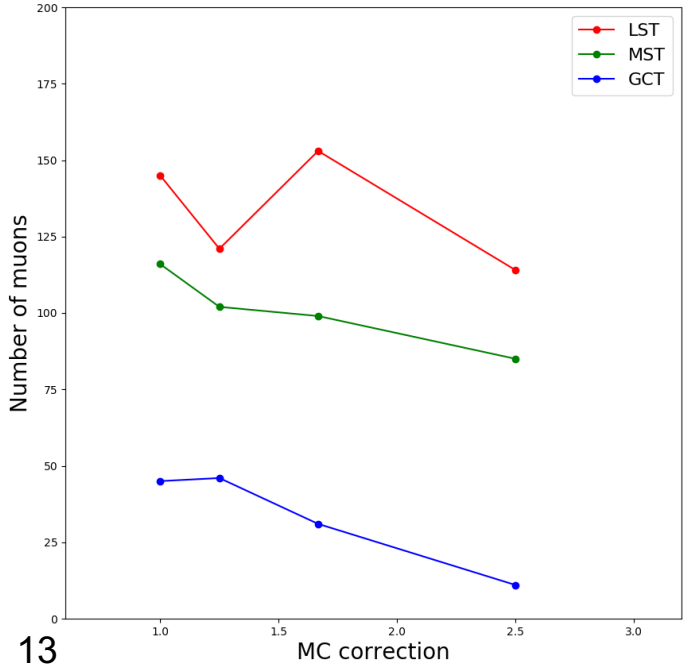
- Circle and width fit, intensity expectation



Figures of Merit – so far

- Gaussian-like efficiency distribution?
- Linearity of correction factor?
- Number of muons found?





Other Figures of Merit

- Independence of muon ring width / impact parameter from the muon optical efficiency
  - can be tested
- Robust performance against broken pixels
  - Needs dedicated study / extra MC
- Agreement between single muons and hadronic muons (e.g. proton simulations)
  - Can be tested with hadronic MC
- Independence of muon efficiency from observation position / environment
  - Probably needs data

Muon Calibration – to do

- Write more unit tests in ctapipe
  (throw warnings if code doesn't work as expected)
- Implement other algorithms
- Check and improve figures of merit
- Implement other applications of muons
  - Psf, camera flat-fielding, mirror calibration,
- Investigate other muon issues
  - Bias due to hadron / wavelength/ broken pixels…

Muon Calibration Paper - Suggestion

- For a technical study, compare the known methods of muon reconstruction and draw conclusions on “best” approach, e.g.
  - Most muons usable for calibration
  - Linearity with optical efficiency
  - Gaussianity of distribution
  - Dual-mirror telescopes performance
- To compare in as unbiased an approach as possible:
  - Implement all algorithms for comparison in ctapipe
  - Use same simulations for testing all algorithms
  - Follow standard input/output/storage class format

MAX-PLANCK-INSTITUT FÜR KERNPHYSIK

Thank you for your attention

Any Questions?

cta
cherenkov telescope array