The CTA pipepline prototype

Tino Michael
CEA Paris Saclay, Irfu/DAp

CTA Consortium Meeting
La Palma, 2017-11-06

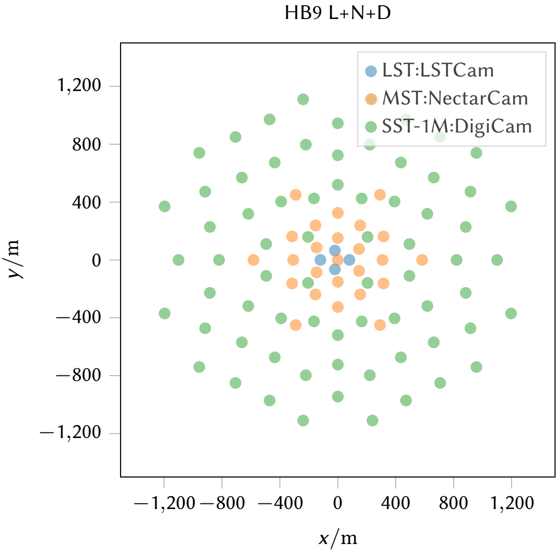
Introduction

the whole pipeline is up and running; cover here:

- image cleaning (tailcuts vs. wavelets)
- shower direction and impact reconstruction (purely geometric approach)
- h_{max} estimate (numerical minimisation in 3D)
- energy reconstruction (machine learning)
- event discrimination (machine learning)
- point-source sensitivity

- my github: `https://github.com/tino-michael/tino_cta`
- still needs external libraries for wavelet cleaning

Array Layout

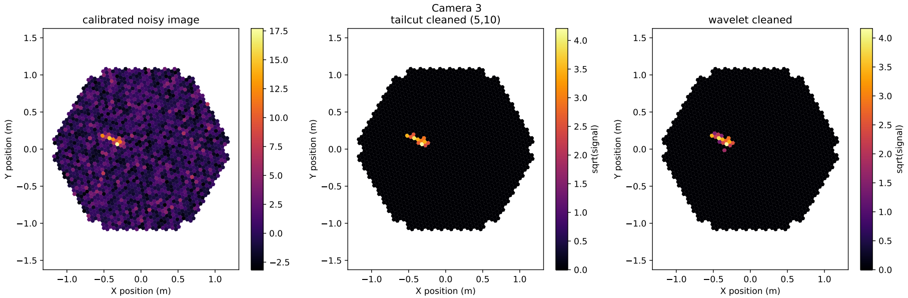


- until recently, only running on ASTRI mini-array
- now on full Paranal array:
- HB9 layout
  (LST + Nectar + DigiCam)
- pointing north at 20°
- on-axis gammas,
  diffuse protons and electrons

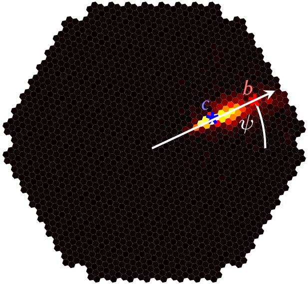
comparing two cleaning methods:

- two-step tailcuts (implemented in ctapipe)
- wavelet cleaning developed by Jérémie (to be merged into ctapipe)

- run the pipeline separately once for each cleaning method
- i.e. each cleaning does its own ML training, reconstruction, discrimination ...

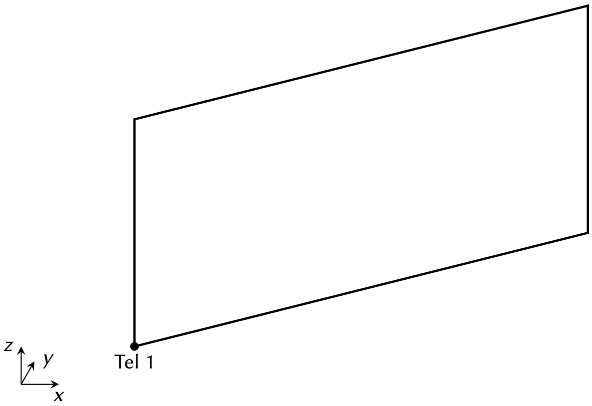


- Hillas parametrisation provides several parameters:
  - position of image core c
  - tilt of ellipsis ψ
- $\rightarrow$ arbitrary point b on shower axis
- note that every pixel on the camera corresponds to a direction in the sky
- b and c form a plane in which the shower lies



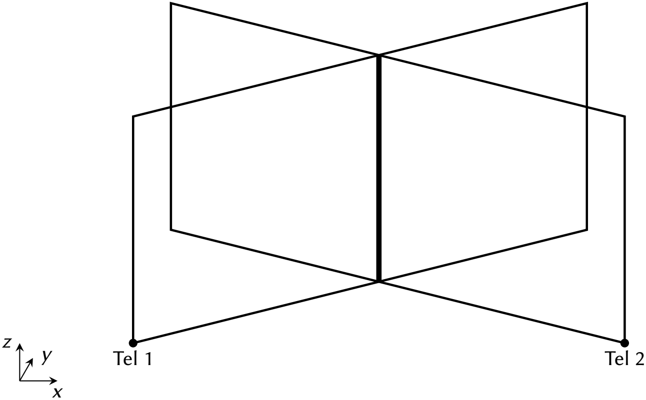
simple shower-reco – direction

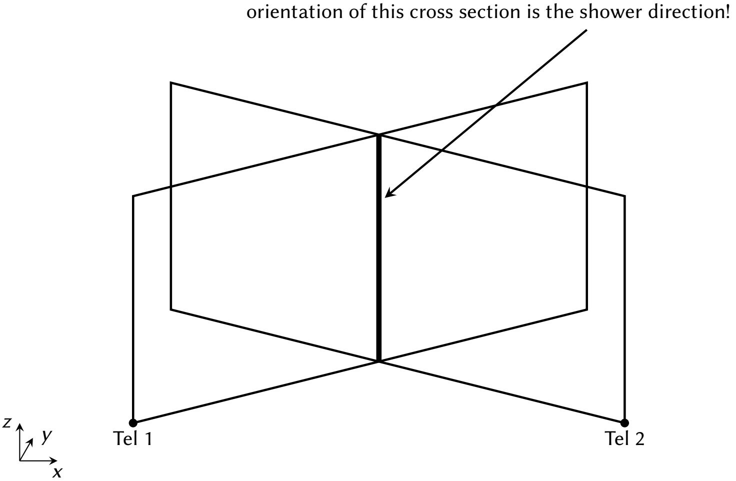
horizontal frame



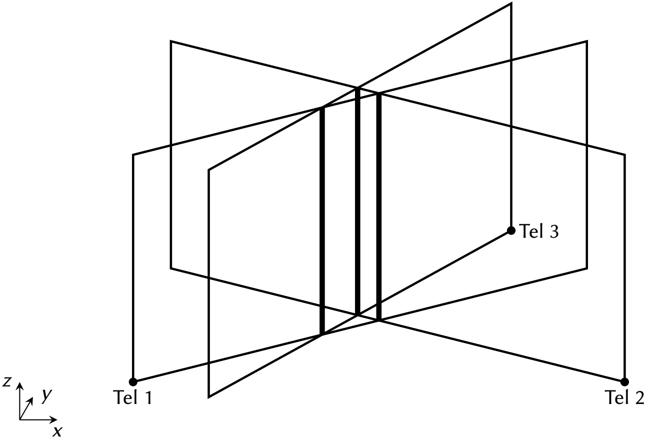
simple shower-reco – direction

horizontal frame




orientation of this cross section is the shower direction!
z
y
x
Tel 1
Tel 2

more cross sections means more direction estimators



- this cross section is perpendicular to the normal direction of both intersecting planes ($\vec{n} = \vec{b} \times \vec{c}$)
- $\rightarrow$ shower direction is $\vec{n}_1 \times \vec{n}_2$
- add up all cross products for weighted mean direction:

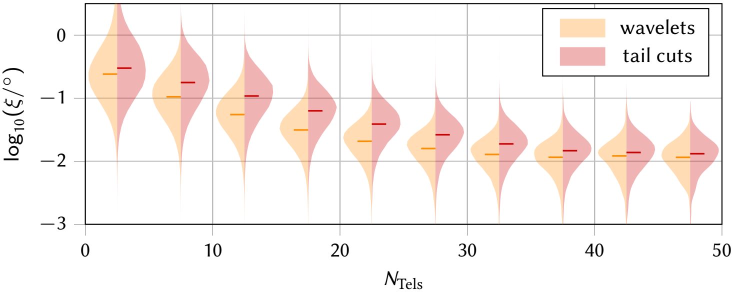
$$\vec{d}_\gamma = \sum_{i=1}^{N_\text{Tels}} \sum_{j=i+1}^{N_\text{Tels}} w_{ij} \cdot \vec{n}_i \times \vec{n}_j$$

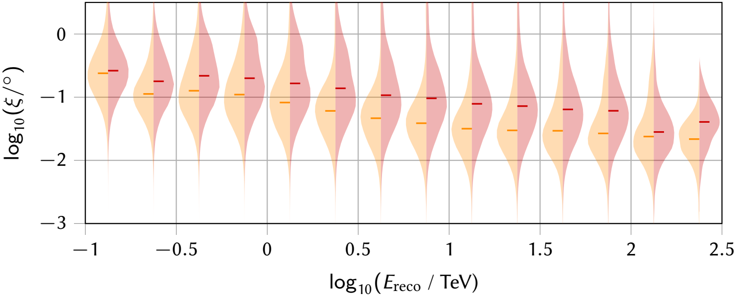
- w_{ij}: weight containing the total image intensity and eccentricity of the Hillas ellipsis
- note: $|\vec{n}_i \times \vec{n}_j| = |\vec{n}_i| \cdot |\vec{n}_j| \cdot \sin[\measuredangle(\vec{n}_i, \vec{n}_j)]$

  $\rightarrow$ automatically weights contributions according to the angle between intersecting planes

  $\rightarrow$ planes pairs crossing with more acute angle are weighted less

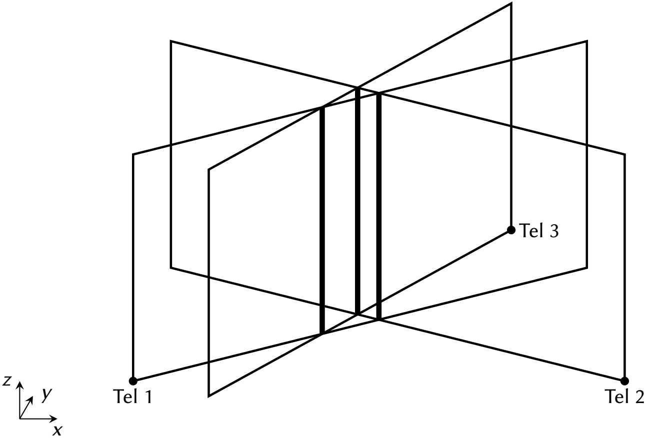
Shower Reconstruction – Direction
Angle between reconstructed and simulated direction



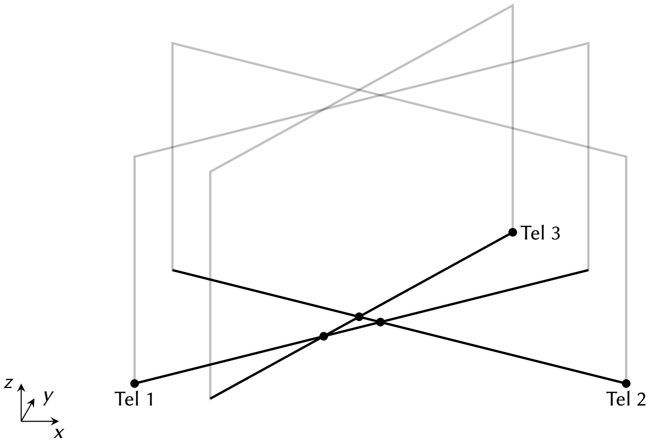


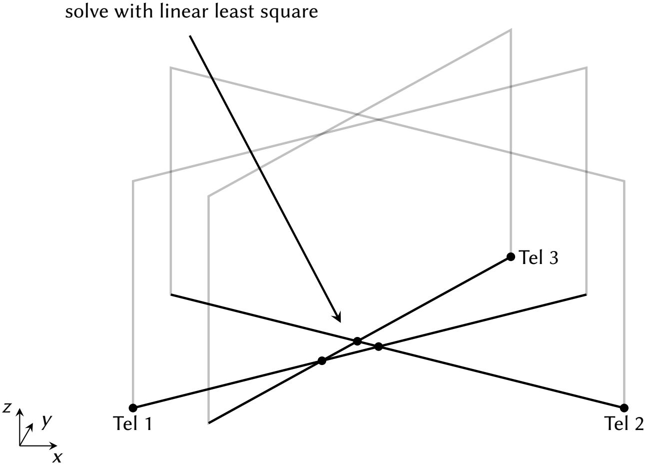
simple shower-reco – impact position

horizontal frame



simple shower-reco – impact position
horizontal frame




solve with linear least square
Tel 3
Tel 1
Tel 2
z
y
x

For any point $\vec{r}$ on a line i holds

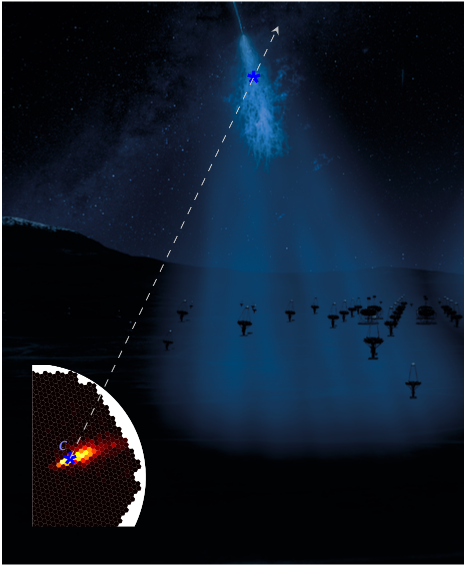
$$\vec{n}_i \cdot \vec{r} = \vec{n}_i \cdot \vec{p}_i = d_i$$

with n_i, the lines normal vector and $\vec{p}_i$, a fixed point on the line (e.g. the telescope position)
If $\vec{r}$ lies on several lines simultaneously, we can write:

$$\begin{pmatrix} n_1^x & n_1^y \\ \vdots & \vdots \\ n_m^x & n_m^y \end{pmatrix} \cdot \vec{r} = \begin{pmatrix} \vec{n}_1 \cdot \vec{p}_i \\ \vdots \\ \vec{n}_m \cdot \vec{p}_m \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_m \end{pmatrix}$$

or $\mathbf{A} \cdot \vec{r} = \vec{d}$.
If all lines i do not cross in one single point $\vec{r}$, there won't be a solution for this equation system.
The "optimal" solution can still be found with least linear squares:

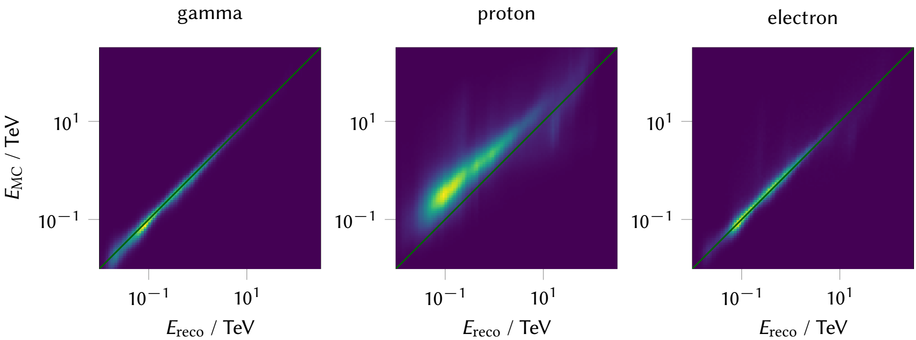
$$\vec{r}_{\chi^2} = (\mathbf{A}^{\mathrm{T}} \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^{\mathrm{T}} \cdot \vec{d}$$



- project the core position of the Hillas ellipsis (vector $\vec{c}$ from direction reco) as a line into the sky
- find the point that minimises the average distance to lines from all telescopes
- no unambiguous normal parametrisation of a line in $\mathbb{R}^3$ ☹
- need to use numeric minimiser after all

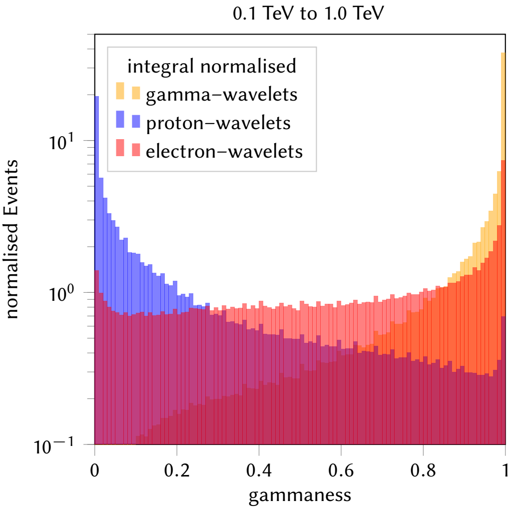
- train 1 Random Decision Forest for each telescope type
- follow a telescope-by-telescope approach
- training features include: Hillas length/width/higher moments, number of telescopes (per size), signal on telescope, summed signal on all triggered telescopes, distance between telescope, shower impact, error estimators, ...

- then, for a given shower event, let the Forest estimate the energy from every telescope separately and combine them into a single energy estimator



Next Stop: Event Classification

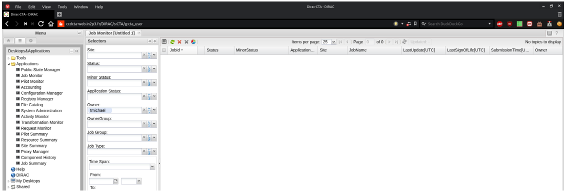
- Protons pose major background
- Event rate about 10^5 times above Photons
- Training Random Forest Classifier
- (virtually identical to energy estimation)

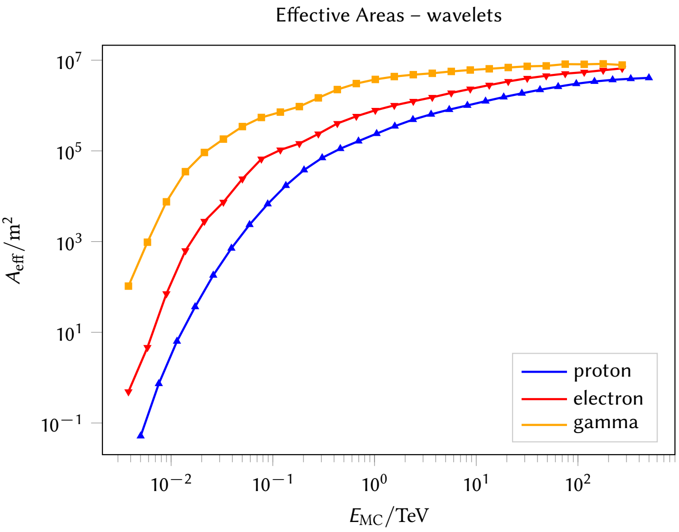


- all this runs with DIRAC on the GRID (including wavelet cleaning)
- lots of sweat, blood and tears to get it going
- Big thanks to Johan and Luisa to take care of the many tickets that I open all the time!

- processing a single setup (5k gamma and electron files, 40k proton files) takes about a week per cleaning mode

- take a look at my **dirac_submit.py** submit-script (handles also some book keeping)





Effective Areas – wavelets
10^7
10^5
10^3
10^1
10^{-1}
$A_{\text{eff}}/\text{m}^2$
10^{-2}
10^{-1}
10^0
10^1
10^2
E_{MC}/TeV
proton
electron
gamma

Summary

- data pipeline from reading of the MC files to IRFs and differential sensitivity fully implemented
- (almost) completely written in python
- many things unmentioned (pyhessio, camera calibration, ImPACT, muon reconstruction, ...)
- still "one or two" things to do

- image cleaning with wavelet outperforms two-stage tailcuts:
  better angular/position resolution, better acceptance, better sensitivity
- will show more of that tomorrow at ASWG session