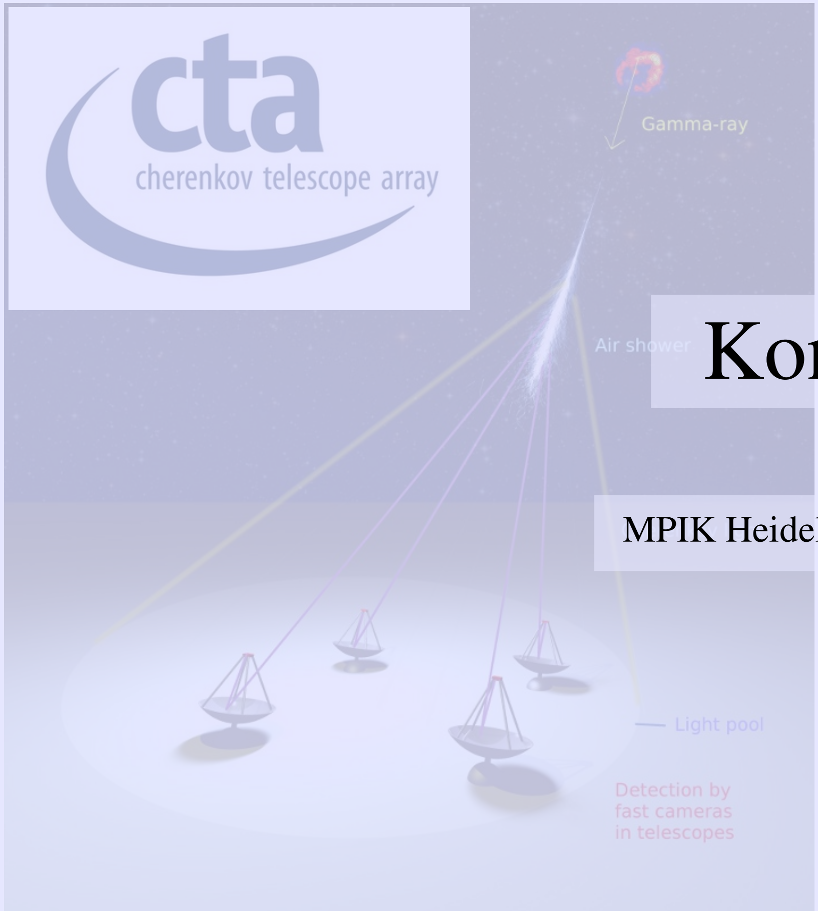
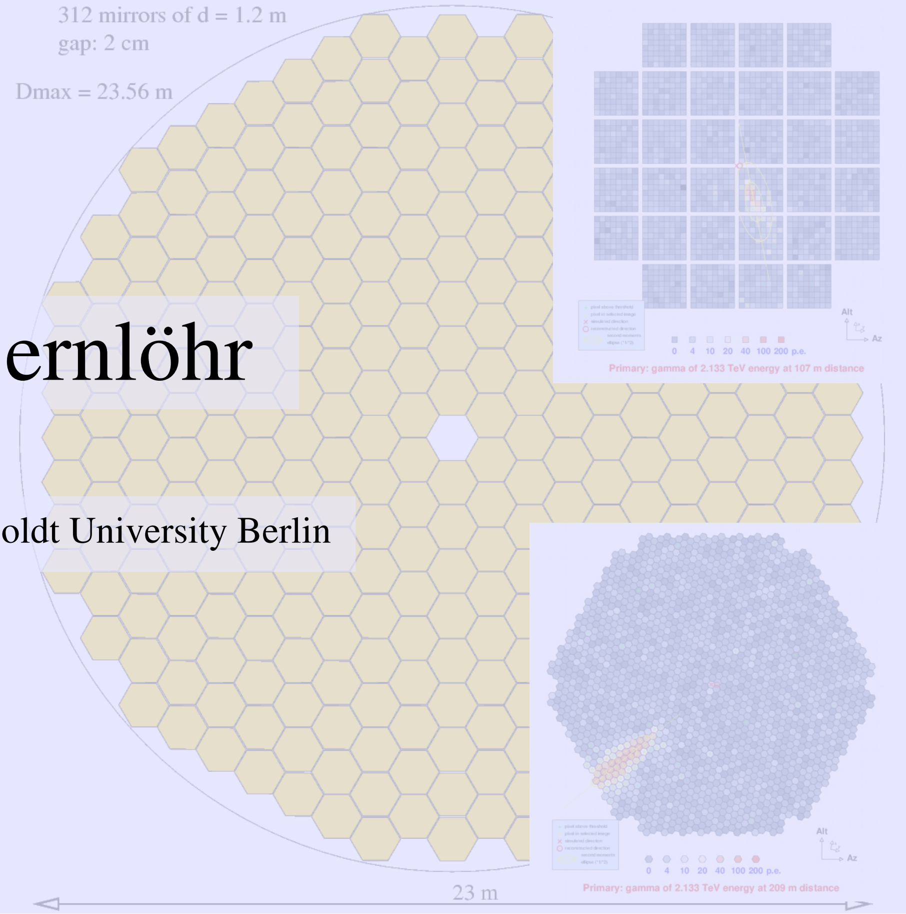
Tutorial 1: Installation and use of the CTA MC package



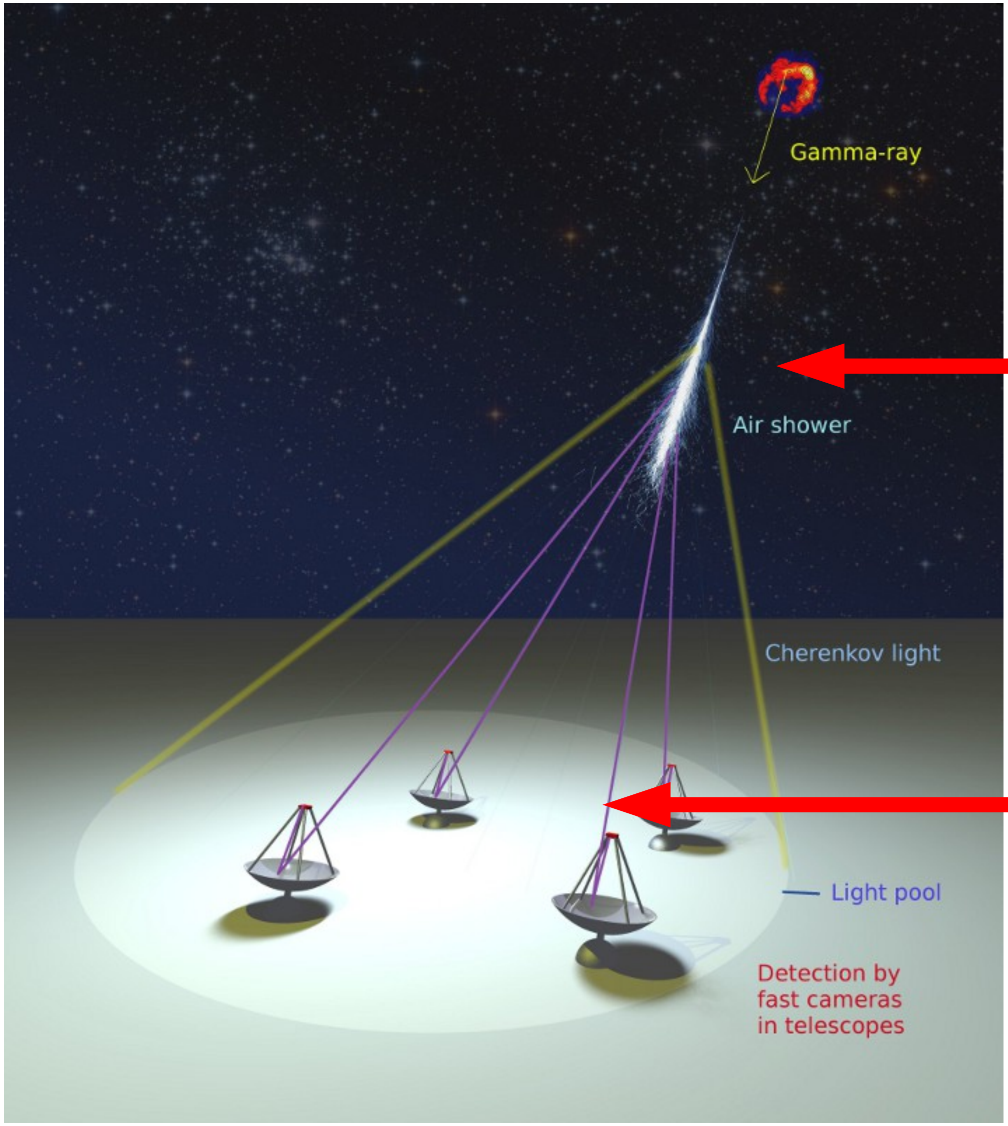
by

Konrad Bernlöhr

MPIK Heidelberg & Humboldt University Berlin


312 mirrors of d = 1.2 m
gap: 2 cm
Dmax = 23.56 m
0 4 10 20 40 100 200 p.e.
Alt
Az
Primary: gamma of 2.133 TeV energy at 107 m distance
0 4 10 20 40 100 200 p.e.
Alt
Az
Primary: gamma of 2.133 TeV energy at 209 m distance
23 m

CTA MC simulation steps



Shower simulation:

CORSIKA

Atmospheric transmission and telescope simulation:

sim_telarray

Goals of this tutorial

Show how to

- install the package,
- basic usage: prod-2 simulations,
- configuration files,
- check produced data,
- run NSB-only trigger simulations,
- optical ray-tracing.

Files for tutorial

- If you want to run the examples yourself, you will need a couple of files, e.g. from
  - http://www.mpi-hd.mpg.de/hfm/CTA/MC/Software/Testing/

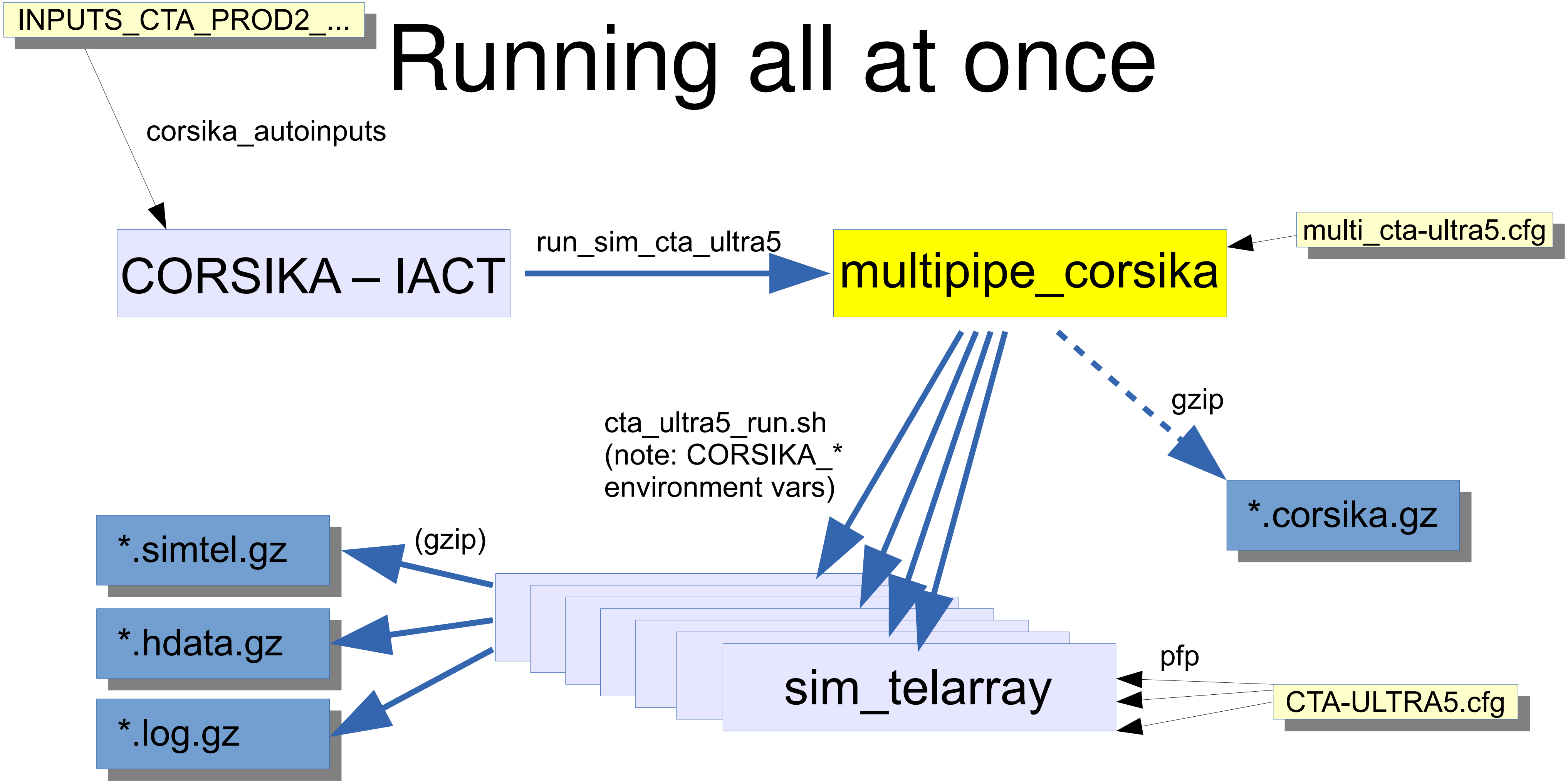
  including
  - corsika_simtelarray.tar.gz
  - gqsdat-II-03

  (In this meeting also available on USB stick.)
- You should have a look at the documentation at
  - http://www.mpi-hd.mpg.de/hfm/CTA/MC/Software/Doc/
- For those not in the meeting, short versions of the exercises will be available as videos.

Installation and a first prod2 run

- Unpack the source code package and create symlink to qgsdat-II-03 file:
  - tar zxvf ../corsika_simtelarray.tar.gz
  - ln -s ../qgsdat-II-03
- Use build_all script for automatic and consistent build of all required software (see 'build_all –help):
  - ./build_all prod2 qgs2
- Initial prod simulation (interrupt after a minute or so):
  - ./example14-gamma_N

Running all at once



INPUTS_CTA_PROD2_...

corsika_autoinputs

CORSIKA – IACT

run_sim_cta_ultra5

multipipe_corsika

multi_cta-ultra5.cfg

gzip

*.corsika.gz

cta_ultra5_run.sh
(note: CORSIKA_*
environment vars)

*.simtel.gz

(gzip)

*.hdata.gz

*.log.gz

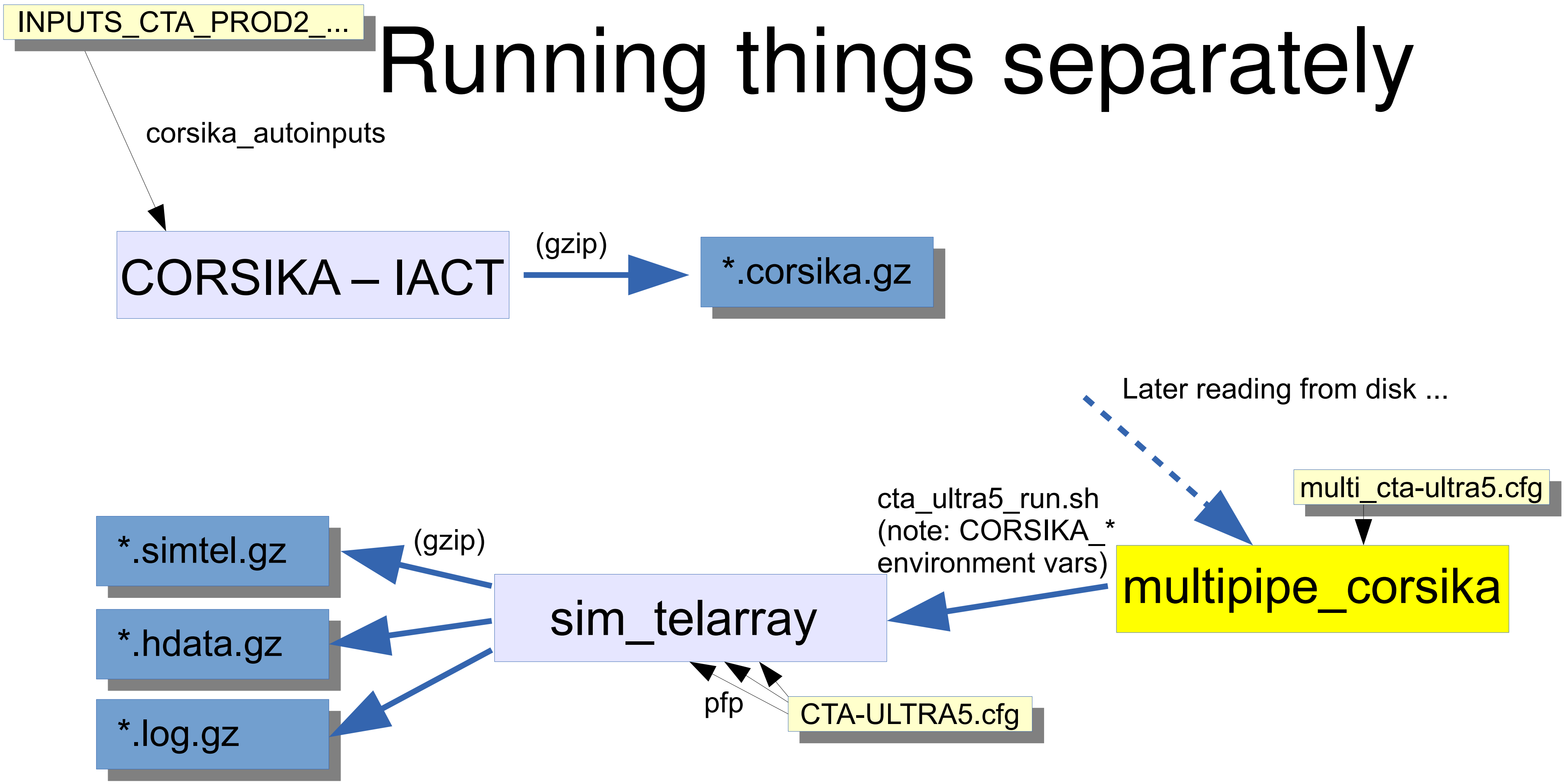
sim_telarray

pfp

CTA-ULTRA5.cfg

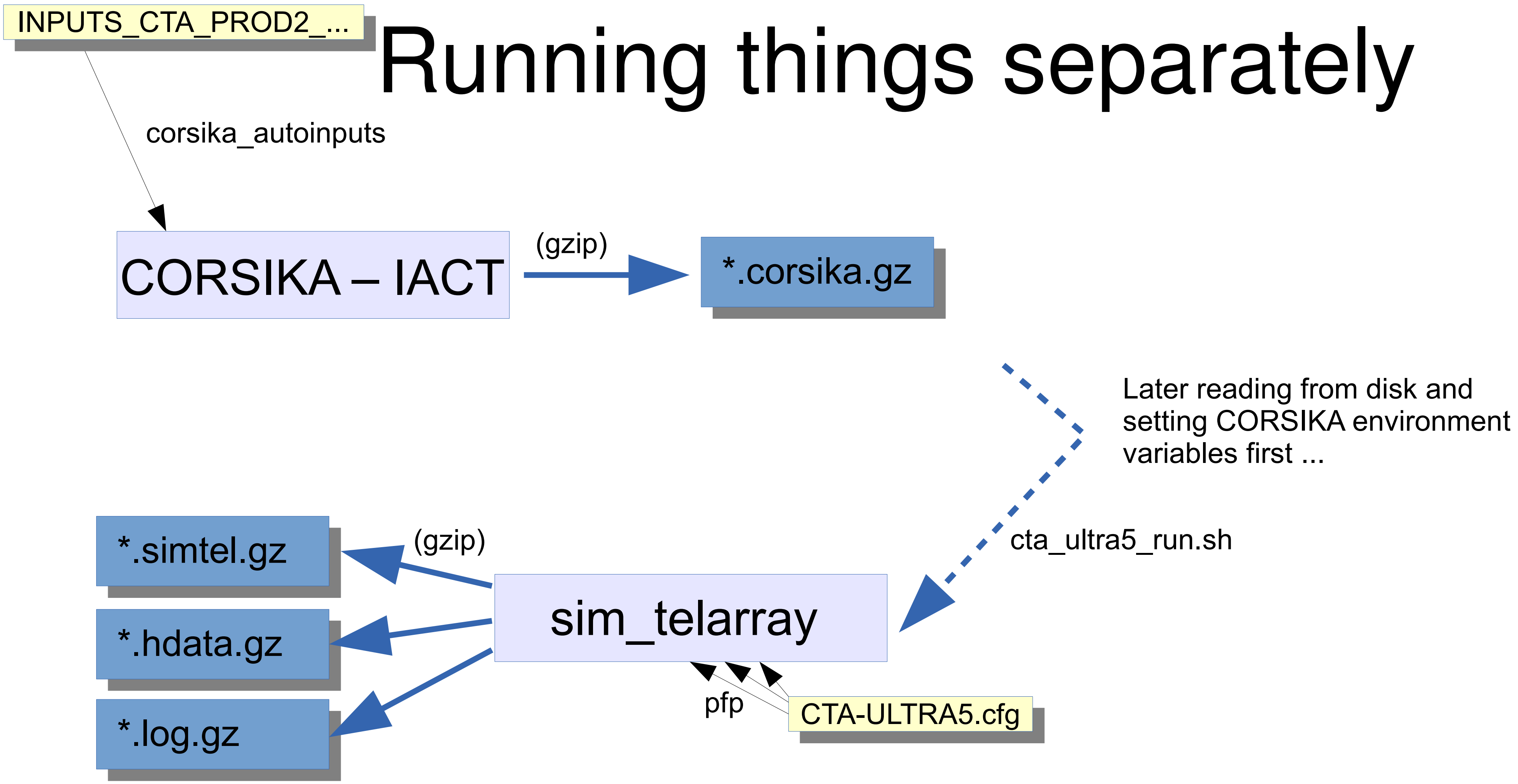
Failure of individual pipes are tolerated, as long as one pipe remains.
Programs are finished when no pipe is left.
To have processing in more sequential order, set
CORSIKA_MULTIPIPE_SEQUENTIAL environment variable.

Running things separately



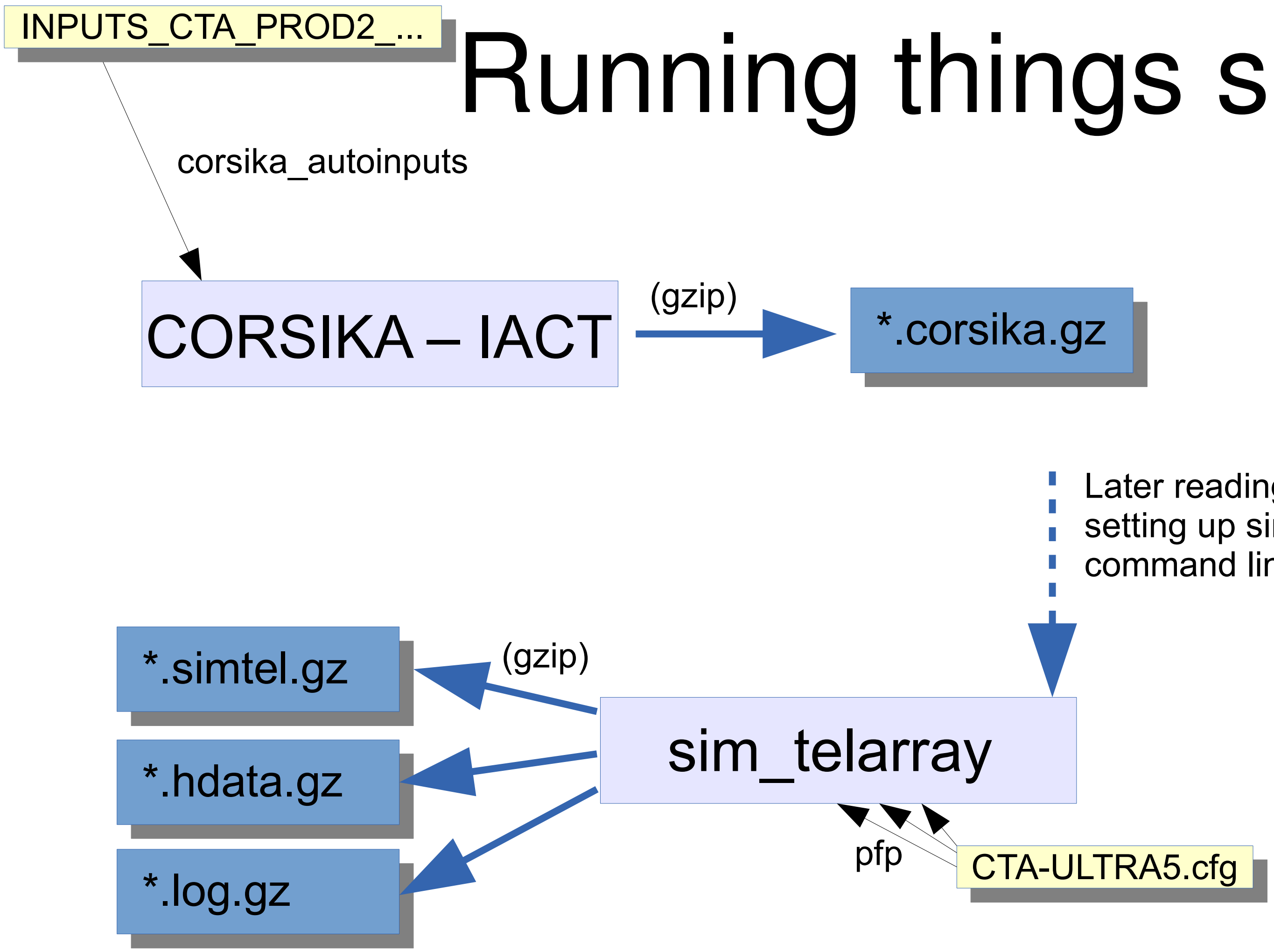
Failure of individual pipes are tolerated, as long as one pipe remains.
Programs are finished when no pipe is left.

Running things separately



Failure of individual pipes are tolerated, as long as one pipe remains. Programs are finished when no pipe is left.

Running things separately



Failure of individual pipes are tolerated, as long as one pipe remains.
Programs are finished when no pipe is left.

Configuration files

- CORSIKA inputs file:
  See CORSIKA_GUIDE69xx.pdf
  For TELFIL etc. see also ReadMe.pdf (iact-atmo)
- multipipe_corsika configuration file:
  All non-empty/comment lines executed in shell.
- sim_telarray configuration: See manual.
  Configuration preprocessed with 'pfp' (with some similarity to C preprocessor: #ifdef, #include, …).
  Command line typically set up with generic_run.sh (aliases: cta_ultra5_run.sh, …).

First data checks

- Checking structure and sanity of any eventio format data file:
  - statio → show statistics of different data block types
  - listio [-s] → shows order of data blocks [+ hierarchy]
- Seeing what is in the data
  - CORSIKA + sim_telarray data_files: read_cta (read_hess)
    - Use '-s' or '-S' to see data elements printed as text, also '-v'
    - Use '-p file.ps' for camera images (Ctrl-C after few seconds!)
  - histogram files: hdata2root, hdata2hbook, list_histograms

    → Tutorial 2

Special sim_telarray use cases

sim_telarray has a number of use cases without actual CORSIKA input (but note that event headers may be needed):

- NSB trigger simulations
- Ray tracing for optical PSF
- Ray tracing for shadowing
- Simulation of calibration data: pedestal, flat-field, single-p.e.

NSB trigger simulations

- Needs CORSIKA event headers but no Cherenkov photons: use dummy100000.corsika.gz file.
- Increase fadc_bins, fadc_sum_bins, disc_bins (recommended setting for all of them: 100), and
  - min_photons = 0
  - min_photoelectrons = 0
  - trigger_telescopes = 1
- Output data not needed (unless looking for NSB events classified as gammas):
  - Count 'has triggered' lines in log file.

NSB trigger simulations

- “Safe” threshold: single telescope 2*dark NSB trigger rate equal to CR single rate (assume 1.5*proton rate).
  - Not so safe that it can be used for everywhere along the Galactic plane.
- “Aggressive” threshold: stereo 2*dark NSB trigger rate equal to 10% of CR stereo rate
  - Problem of definition (depending on layout, …)
  - Although stereo NSB rate can be simulated directly, it would be more efficient to derive it from single NSB rate.

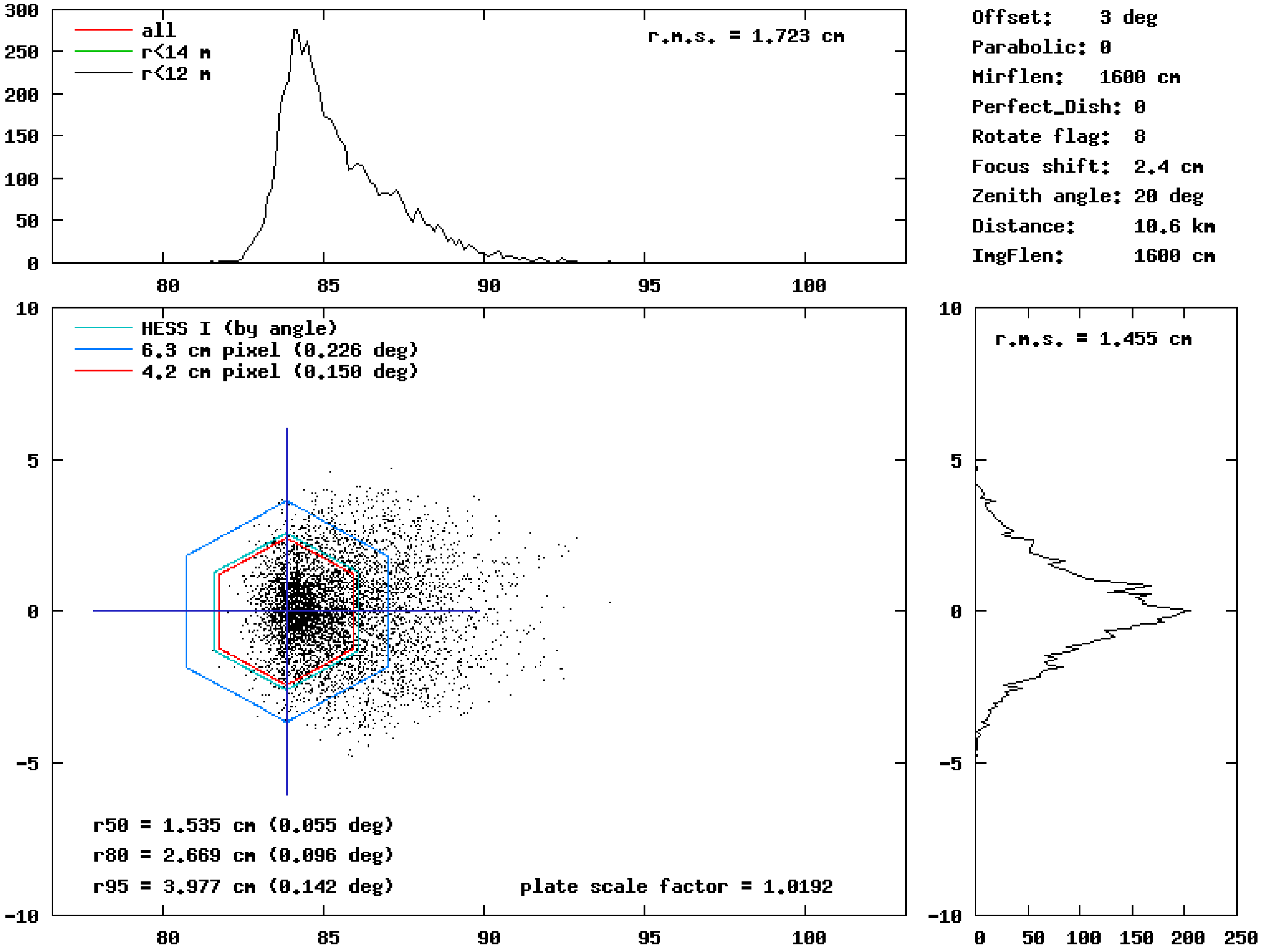
Ray-tracing for optical PSF

- Set up a light source ('star') file:

  star_at_z.dat: 0. 70 1.0 10.6

- Use it and produce ray-tracing output with
  - stars=star_at_z.dat
  - star_photons=10000
  - imaging_list=xt.lis
- Need single event CORSIKA file: dummy1.corsika.gz
- Use multiple runs to smooth over random mirror misalignments (output to 'xt.lis' is appended).
- Use 'rx' to get x% containment radius of PSF, ...

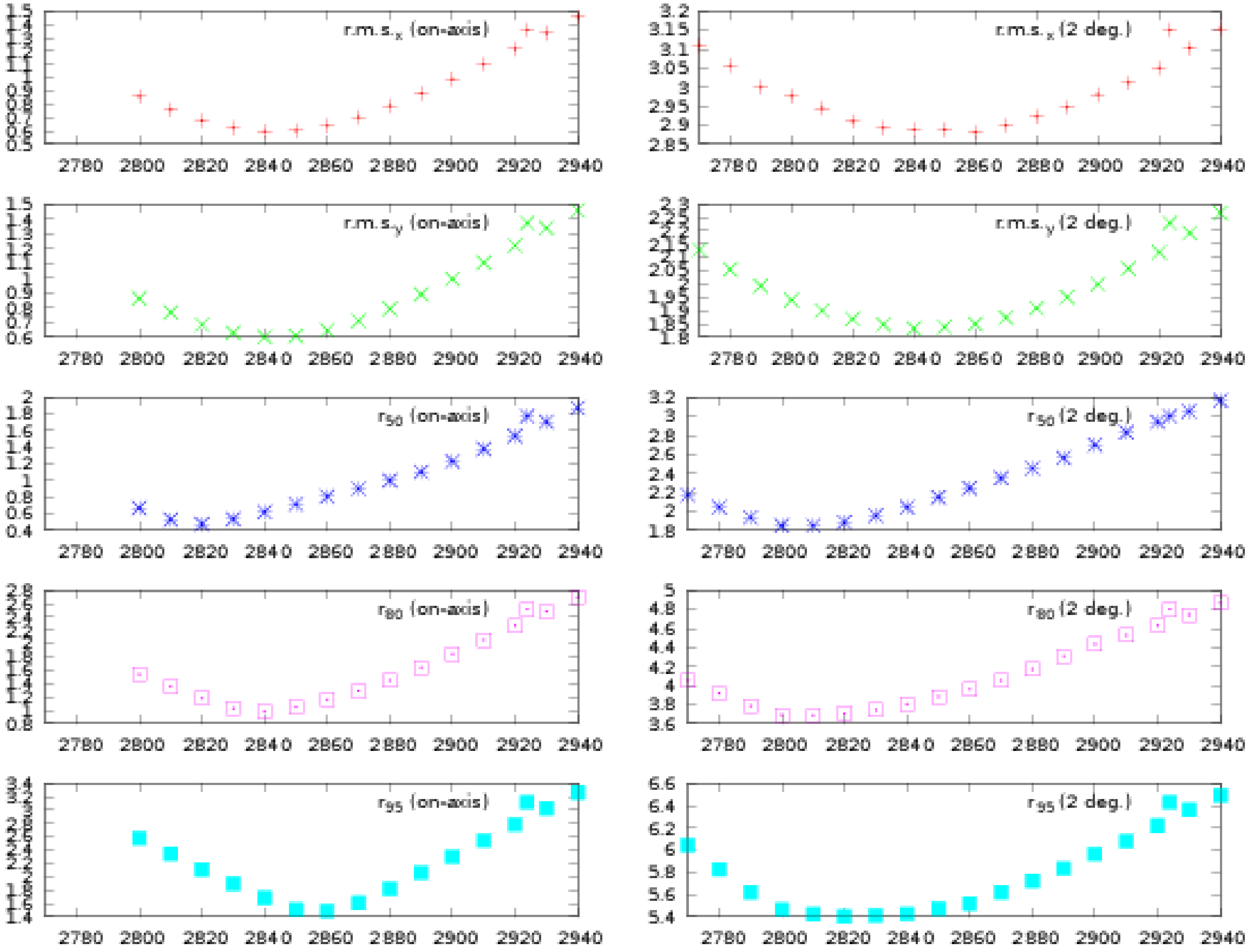
Ray-tracing for optical PSF



```
Offset:        3 deg
Parabolic: 0
Mirflen:       1600 cm
Perfect_Dish: 0
Rotate flag:    8
Focus shift:    2.4 cm
Zenith angle: 20 deg
Distance:       10.6 km
ImgFlen:        1600 cm
```

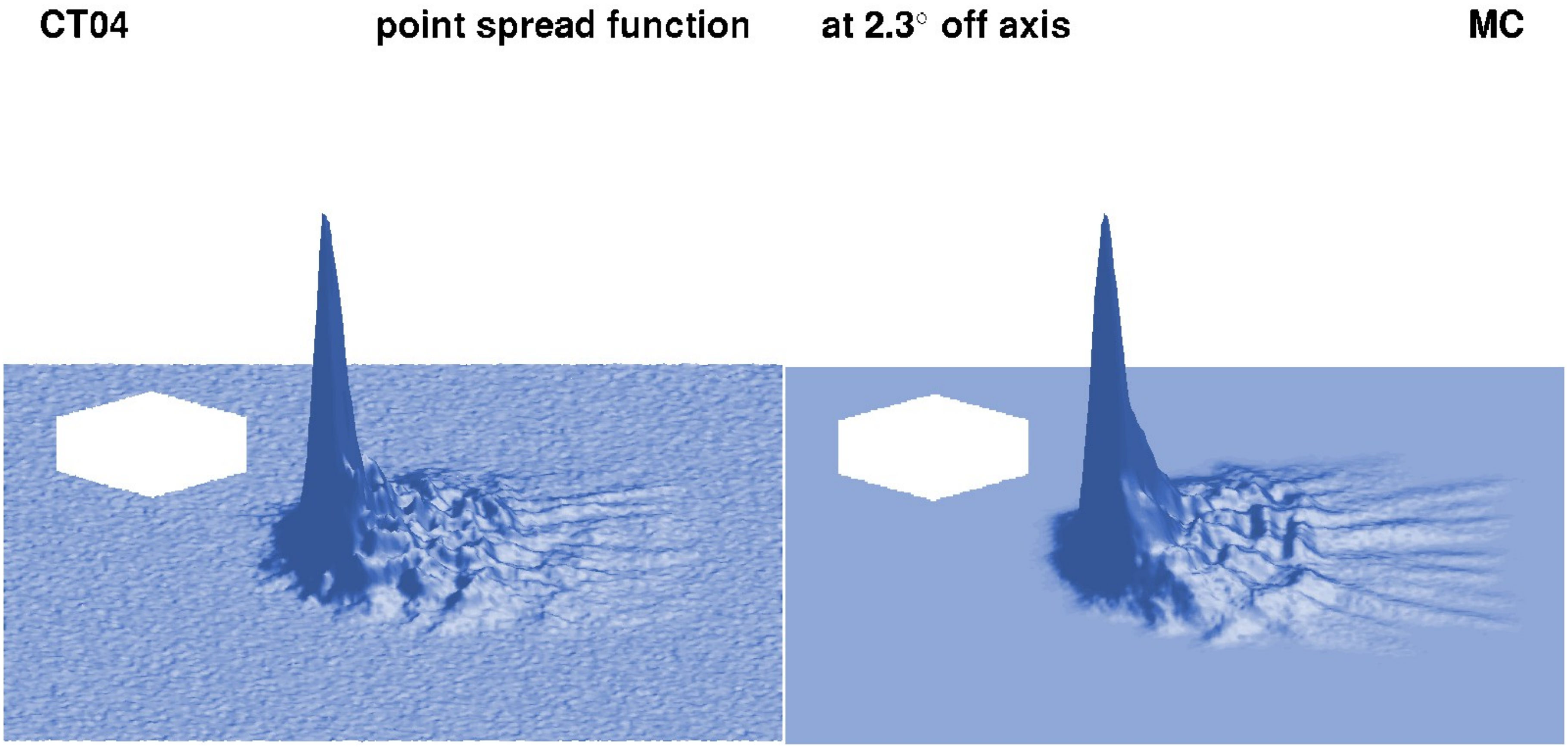
```
r50 = 1.535 cm (0.055 deg)
r80 = 2.669 cm (0.096 deg)
r95 = 3.977 cm (0.142 deg)          plate scale factor = 1.0192
```

Ray-tracing for optical PSF



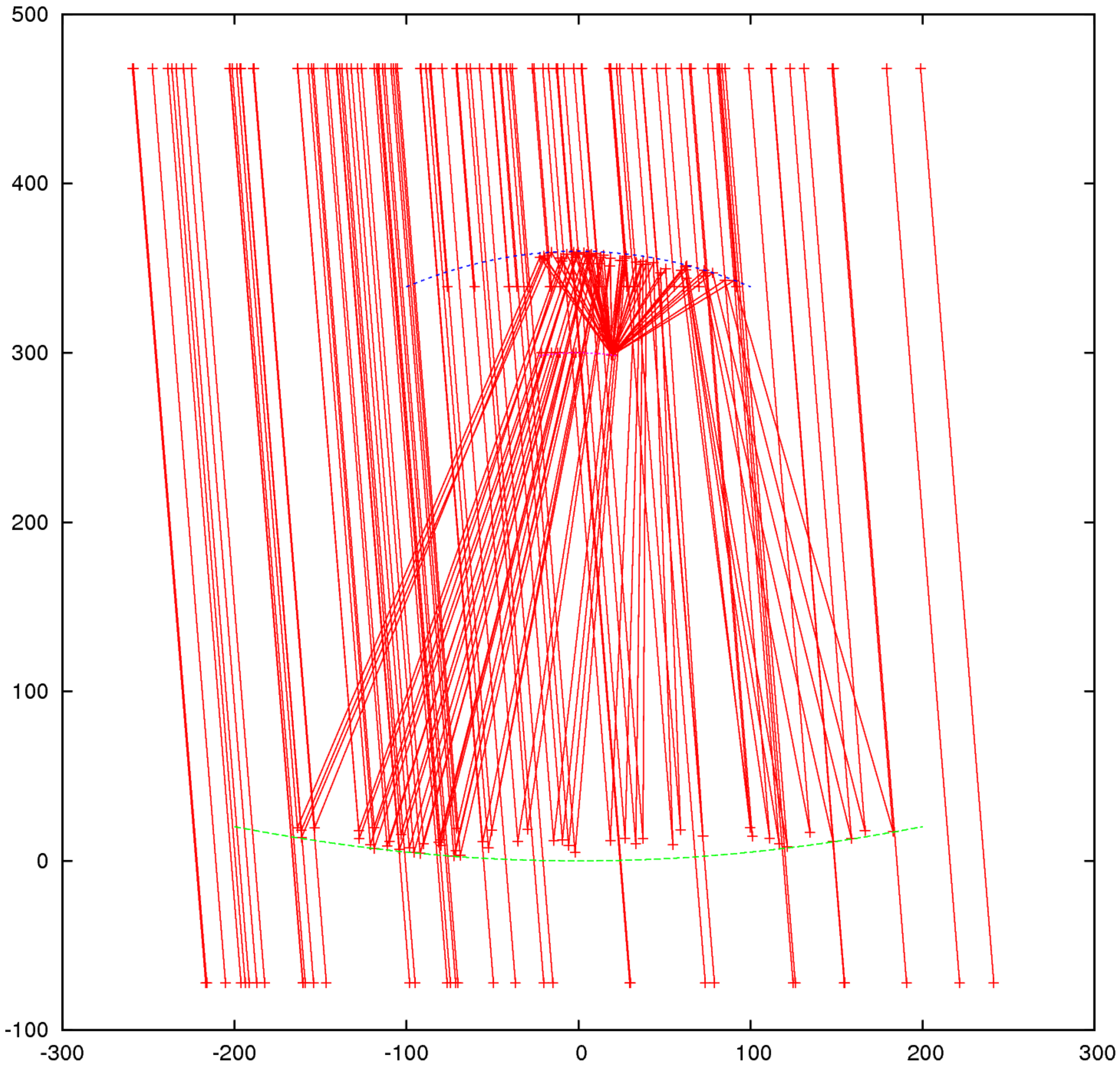
Comparison with measured PSF

H.E.S.S.



R. Cornils

Ray-tracing with dual mirror optics



Reflectors not segmented.

No shadowing of support structures implemented.

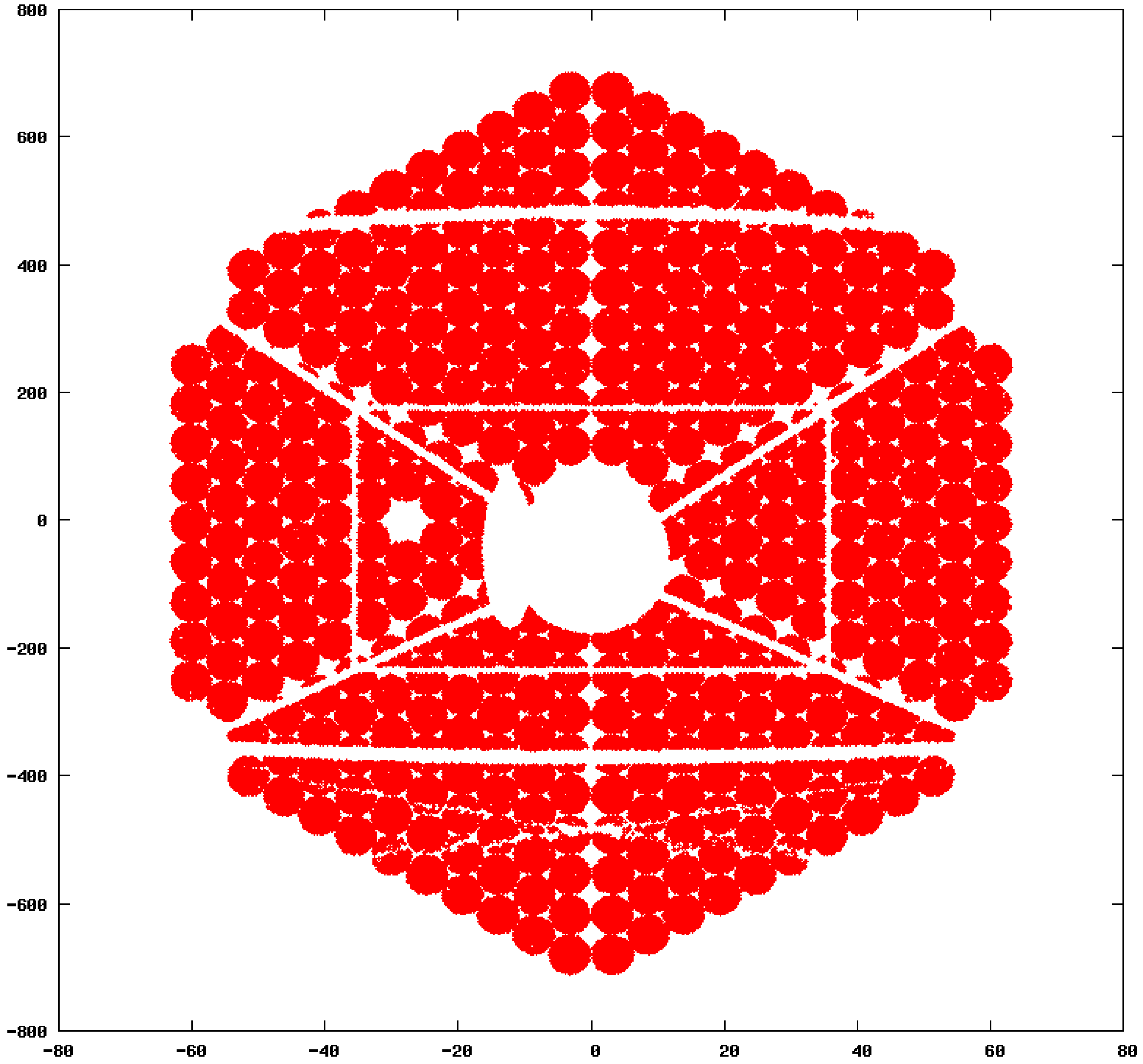
Ray-tracing for shadowing

- The ray-tracing in normal simulations only includes explicit shadowing of the camera body on incoming light:
  - camera_body_diameter = … % [cm]
- For more explicit shadowing compile with

  ```
  -DRAYTRACING_INTERSECT_RODS
  ```

  and include file with shadowing elements (cylinders)
  - masts_file = CTA-MST_masts.dat % not latest design!
  - camera_body_diameter = 0. % if explicit element used
- See impact points of lost photons etc.

Ray-tracing for shadowing



(H.E.S.S. telescope)

End of this tutorial session

- You should be able to
  - compile CORSIKA and sim_telarray now,
  - run simulations with the prepared configurations either
    - in single CORSIKA/multipipe_corsika/sim_telarray pipes or
    - split them into first running CORSIKA and then sim_telarray,
  - evaluate single-telescope random NSB trigger rates for a given telescope (and trigger) configuration,
  - use ray-tracing for PSF or shadowing.
- I am sure you still have questions!