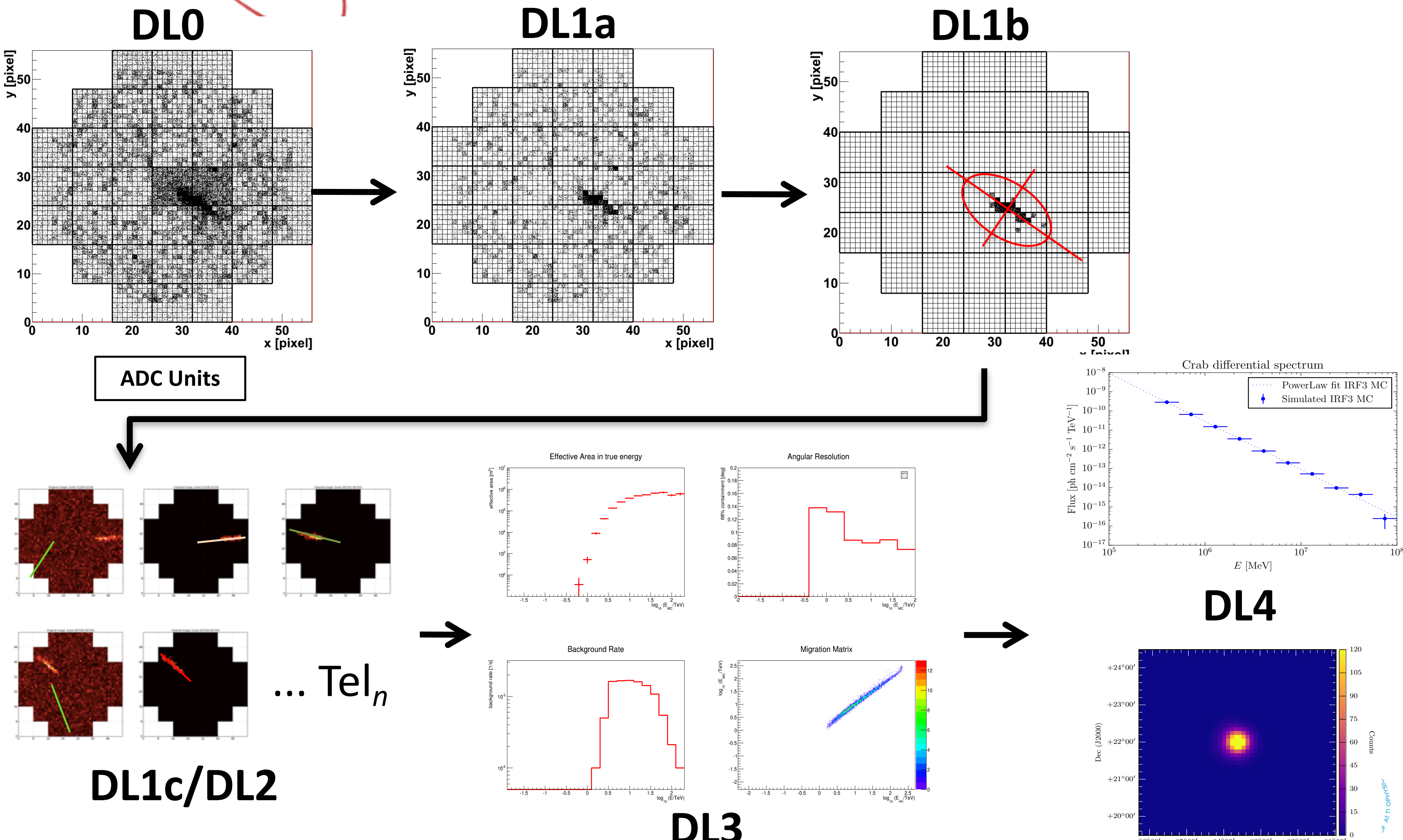
Low-power computing with ASTRI & CTA use cases

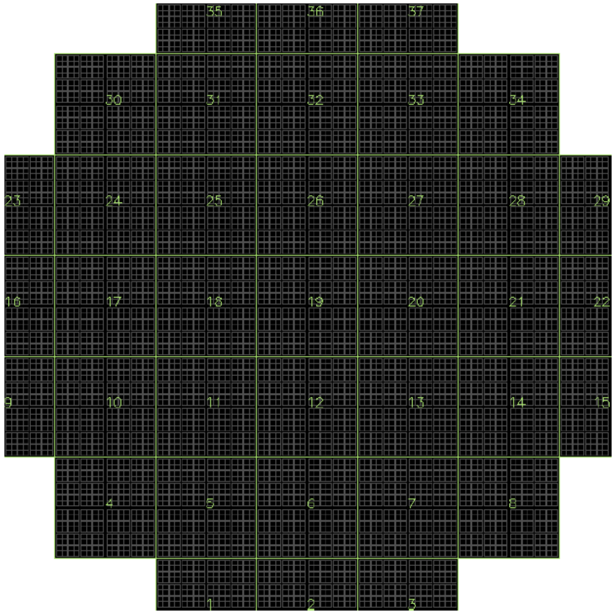
Presented by
D. Bastieri
GPU Research Center
Università di Padova
& INAF

for the CTA Consortium

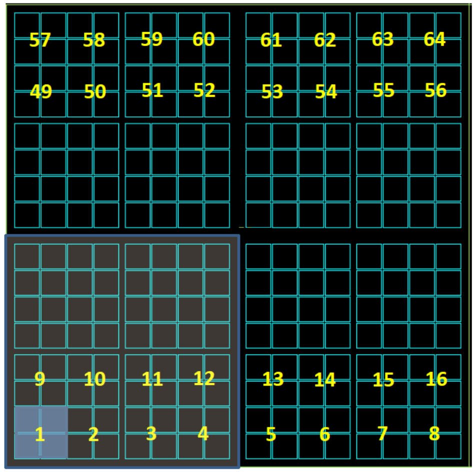
Main authors: A. Madonna, M. Urbani, G. Urso (Padova: Univ/INAF)
L.A. Antonelli, M. Mastropietro, S. Lombardi & F. Lucarelli (Roma: INAF/OAR + ASDC)

ASTRI SciSoft Data Flow





PDM pixels channels arrangement



The ASTRI SST-2M Prototype Camera

- Composed of 37 PDMs
- 2368 total pixels
  1984 connected pixels
- Camera hardware already integrates time slices
- DL0 data is made of HG ADC and LG ADC for each pixel

ASTRI Pixel-level algorithms easily express parallelism

- **Calibration**

  Essentially an *embarrassingly parallel*, Fused Multiply-Add operation (ASTRI camera outputs integrated ADC counts):

  PHE = ADC * coefficient + pedestal

  ```
  $0 = $0×$1+$2
  ```

- **Cleaning**

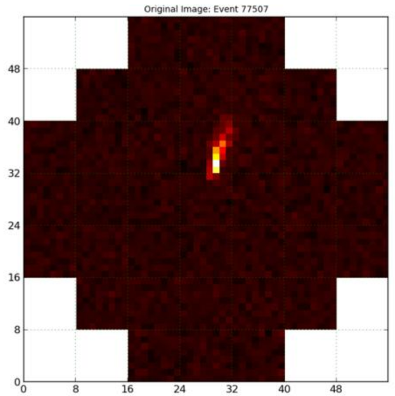
  Two pass cleaning (two threshold comparisons)

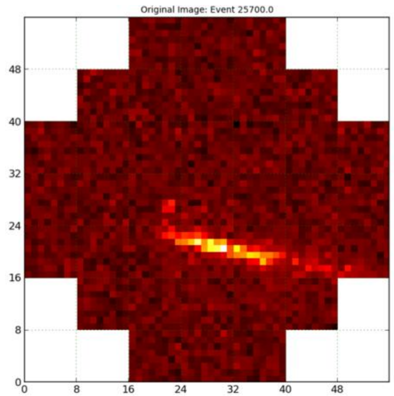
  Well suited to parallelism

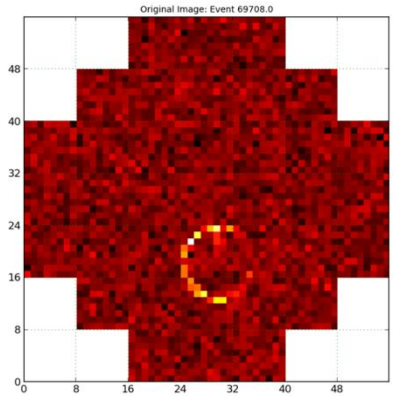
NVIDIA CUDA

NVIDIA GPU RESEARCH CENTER

Reference Test Case





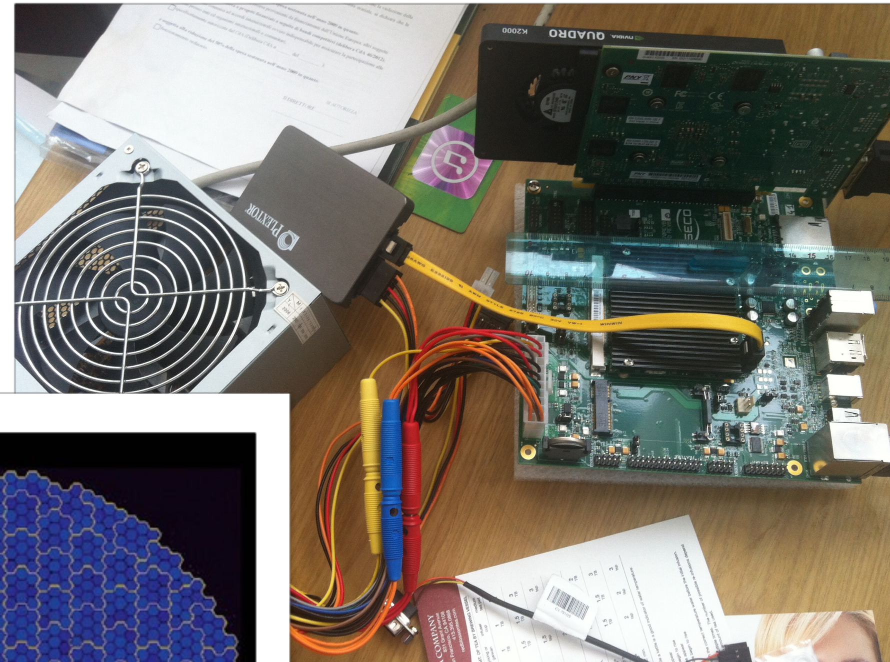


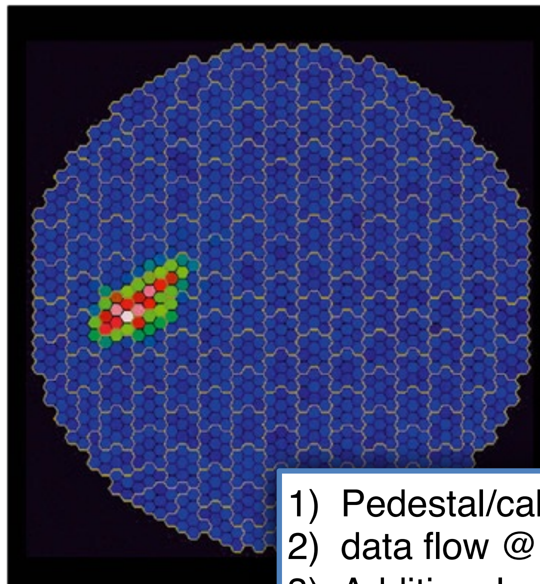
- 500MB (= 55049 events) of simulated DL0 “real data”
- ≈ 110s of nominal acquisition rate (500Hz)
- ≈ 55s of projected peak rate (1000Hz)
- ≈ 80.5% of events survives pruning with default settings
- Compliant with format and size agreed with camera hardware team

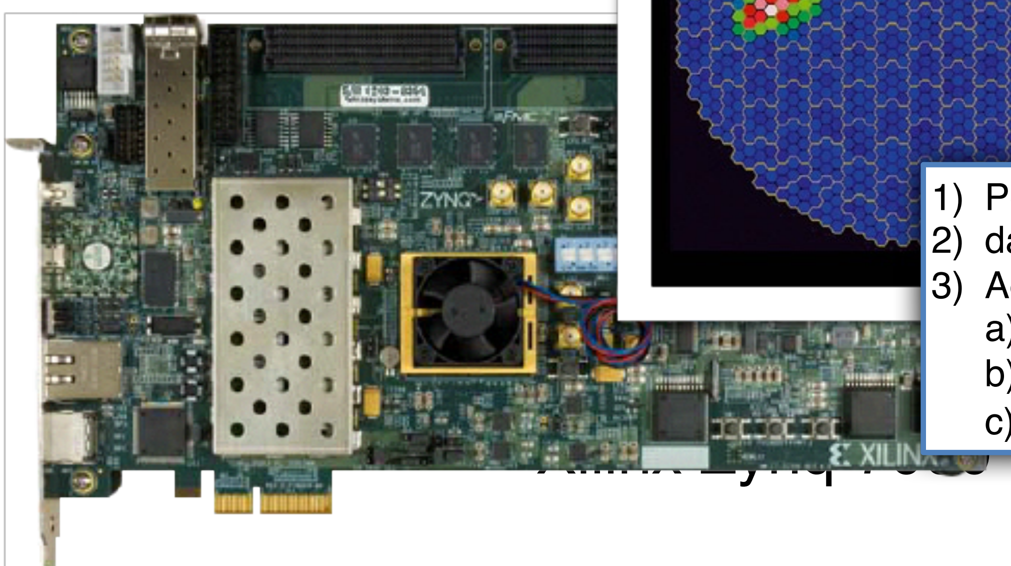
Data Crunching: recipe from OAR

1) Evaluate pedestal offsets from 2k random events
2) Real data input (2GB = 50 s on MAGIC II @200Hz)
3) Pedestal subtraction
4) signal integration via sliding window (short[] → int)
5) ADC counts (int) → (× calibration) → phe (float)
6) phe sorting/clustering/cleaning
7) evaluation of first 10 momenta
8) data output

D. Bastieri & S. Buson (UNIPD)
L.A. Antonelli, D. Gasparrini, S. Lombardi, F. Lucarelli & M



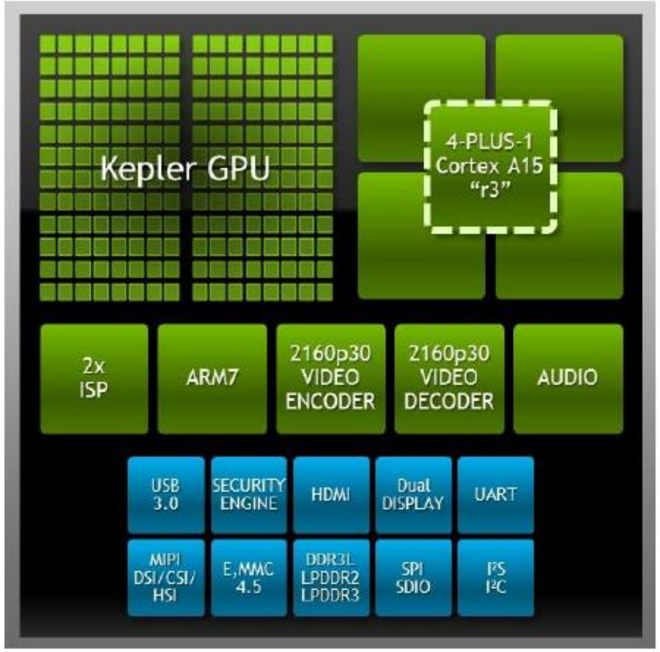




1) Pedestal/calibration feasible on ARM @5W.
2) data flow @1Gb/s, data processing @~2GB/min
3) Additional analysis:
   a) spawn it to GPU's cores (add 20W or <P> ~11W)
   b) filter through FPGA (?2W? <P> ~8W)
   c) try out Jetson-TK1 (SoC)

NVIDIA Jetson TK1

- Heterogeneous System-on-Chip
- CPU: Quad-core ARM A15
- GPU: Kepler architecture - 1 Multiprocessor
- RAM: 2GB (unified address memory)
- OS: Ubuntu 14.04 Linux for Tegra (L4T)
- CUDA 6.5
- I/O: SATA 3Gb/s HDD (no on-board eMMC)

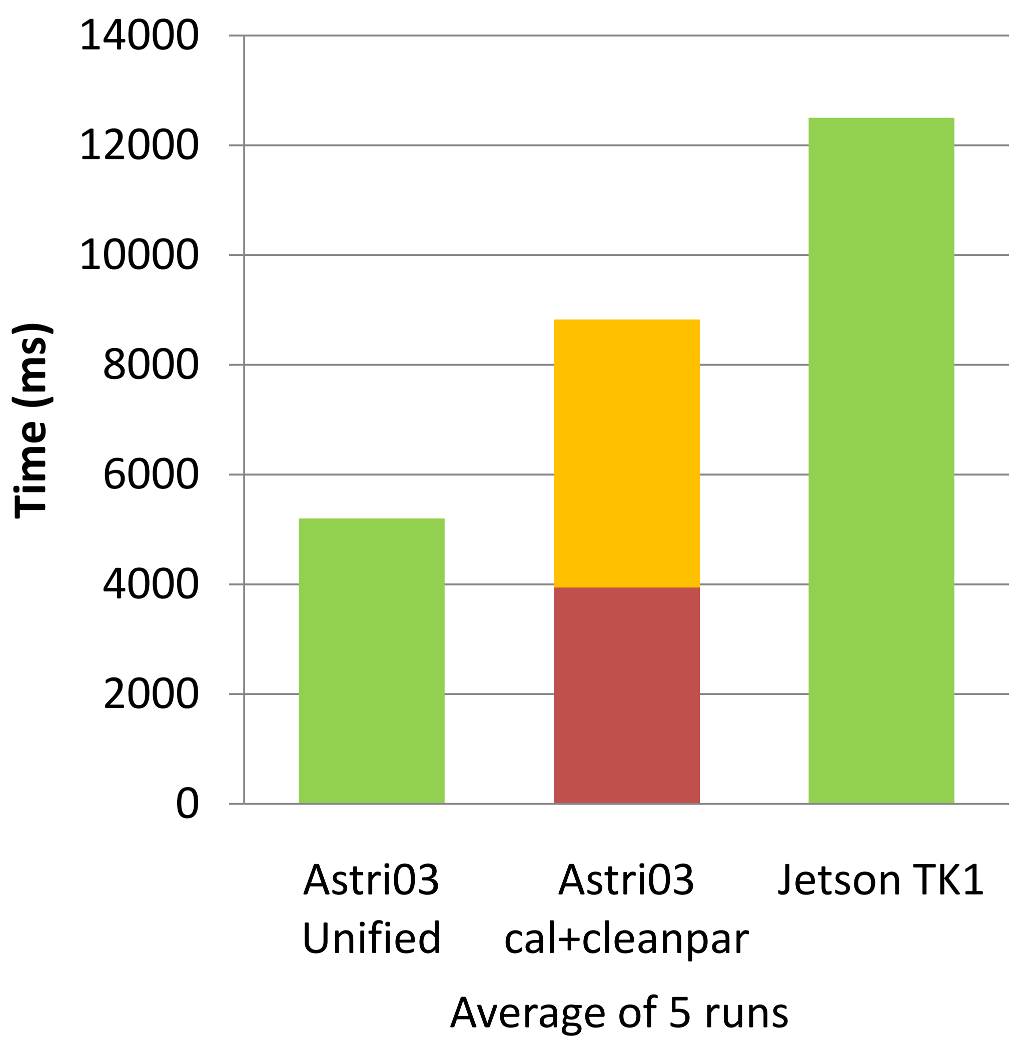


Average power consumption: < 10 W

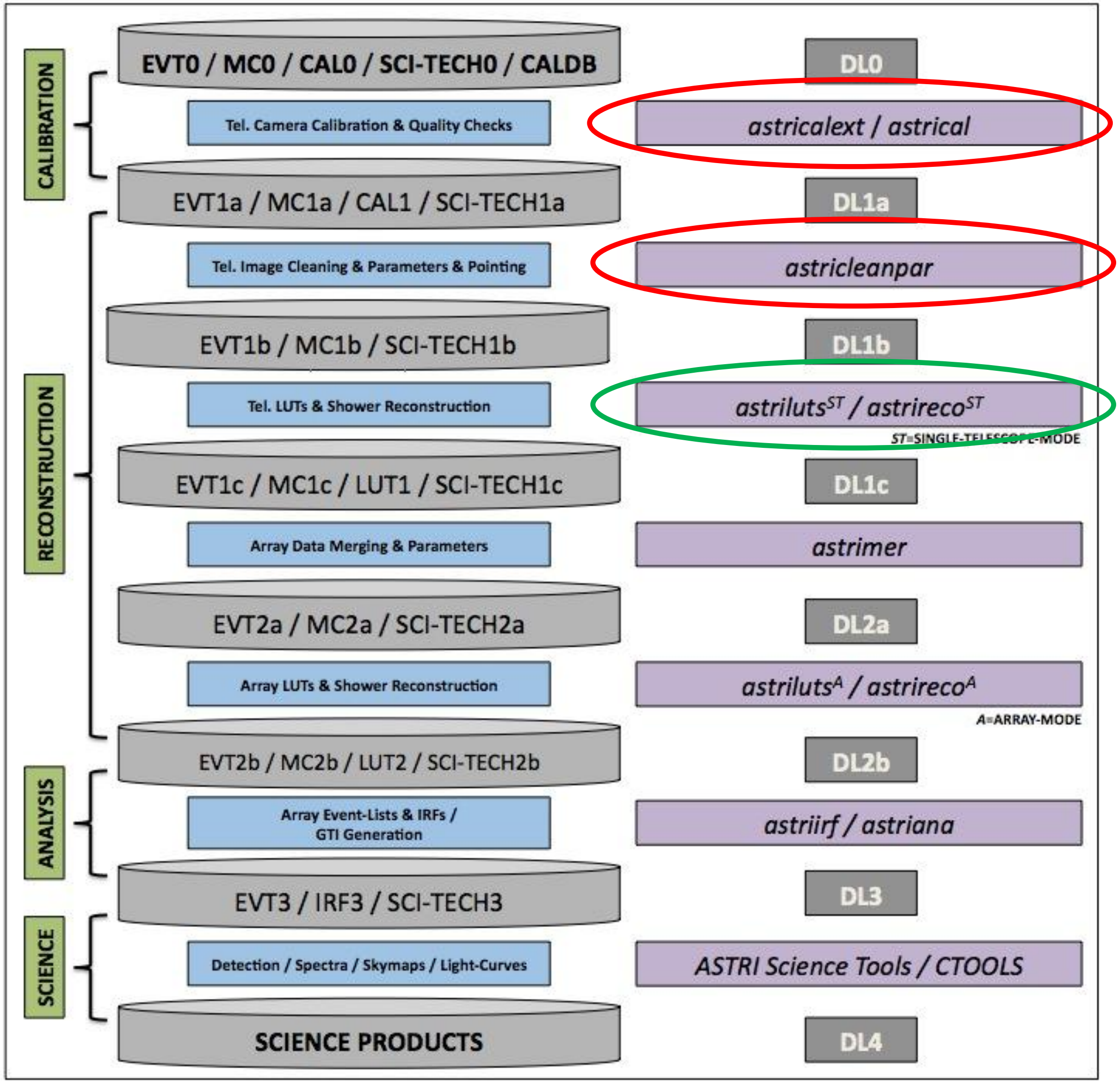
Low-level "Unified module"

- Performs calibration + cleaning + parameters computation in a single, tightly integrated program
- Direct processing from DL0 to DL1b (size-reduced telescope-wise data)
- 73x reduction in data size
- Minimizes disk transfer time

Low-power Unified module



- Processing from DL0 to DL1b (size-reduced telescope-wise data)
- All done in 12.5s:
  4400 evt/s
  > 4x peak acquisition rate
- 2.5x slower than server UM
  1.4x slower than separate modules
  30x less power
- Still plenty of time left for online analysis!



Lombardi, Antonelli, Bastieri et al.

Feed Jetson with other tasks

astrireco

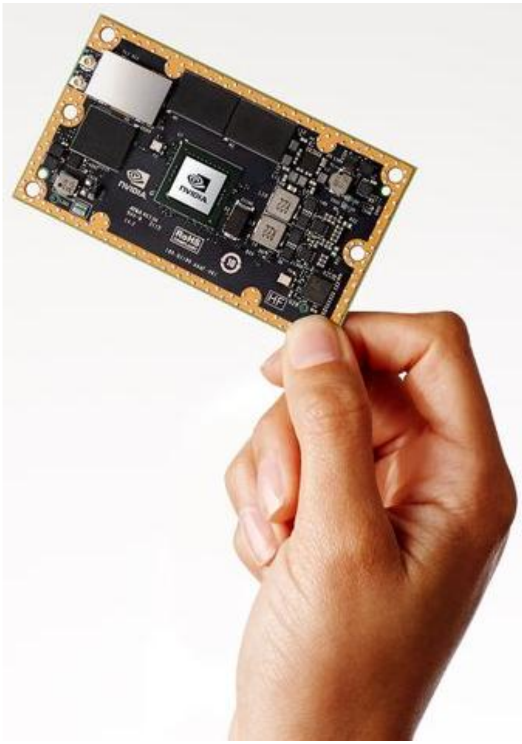
- Implements random forest application
- Loads pre-trained models (look up tables LUTs)
- Energy, direction and hadronness reconstruction

Execution time: 10s - 4 ARM core (using OpenMP)

Single telescope reconstruction pipeline

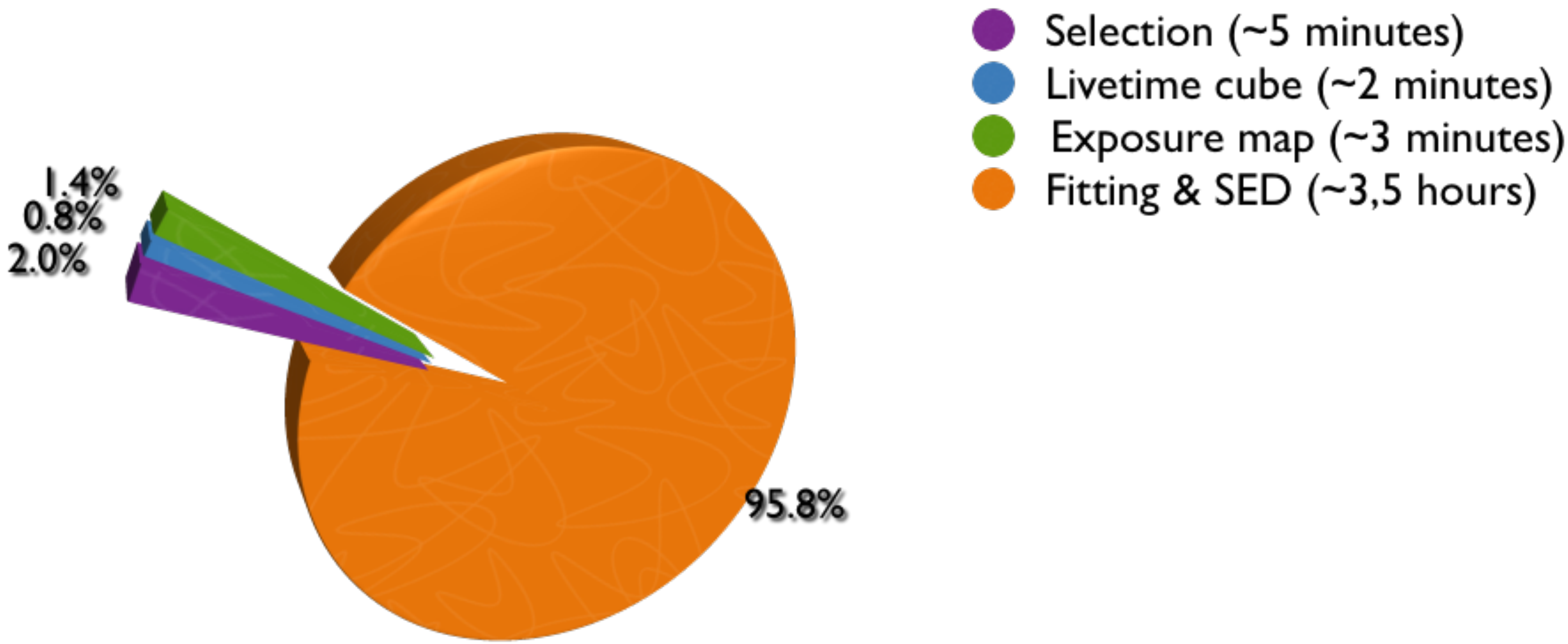
- Reduction + reconstruction = 22.5 s
- DL0 -> DL1c @ 2500 evt/s
- 2.5x of peak acquisition rate on embedded hardware

NVIDIA Jetson TX1



- Latest generation embedded module from NVIDIA (announced Nov. 11th 2015)
- Credit-card size, touted of same ≈10W consumption (max 15W)
- CPU: Quad-core ARM A57
- GPU: 256-core Maxwell arch (2 SMM multiprocessors)
- 4GB RAM, Gigabit Ethernet
- Devkit with carrier board: $600

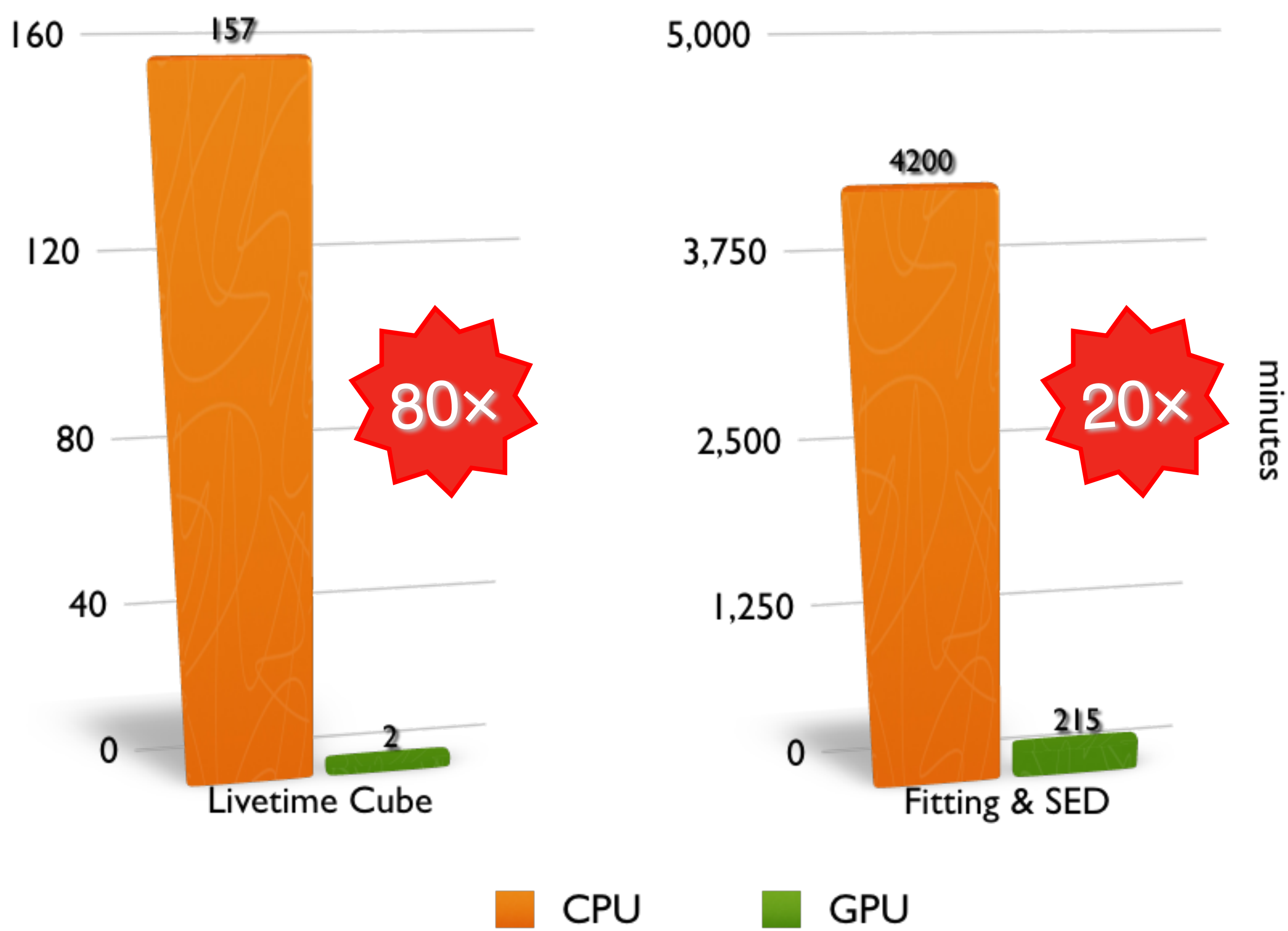
Computational Cost



from 3 days to **4 hours** per source for ~5 years worth of data

New Pipeline · NVIDIA S2050 · 3 GB RAM

Performance comparison



Maximum Likelihood Estimation on GPUs: Leveraging Dynamic Parallelism

M. Mastropietro[1], D. Bastieri[2,3], A. Pigato[2], A. Madonna[1,2],
S. Amerio[3], D. Lucchesi[3], L.A. Antonelli[1] & G. Lamanna[4]

1. *Rome Observatory, INAF, Rome, Italy*
2. *CUDA Research Center, University of Padova, Italy*
3. *Dept. Physics and Astronomy, Univ. Padova and INFN, Padova, Italy*
4. *LAPP, Laboratoire d'Annecy-le-Vieux de physique des particules, Annecy, France*

Maximum Likelihood Approach

- Parameters estimation through maximization
- Hypotheses testing through Wilks's theorem

Null hypothesis max likelihood, h parameters

$$\mathrm{TS} = -2\,\log\frac{\mathscr{L}_0}{\mathscr{L}} \underset{N\to\infty}{\rightarrow} \chi^2_{m-h}$$

non fixed parameters

Alternative hypothesis max likelihood, m parameters

Unbinned Likelihood

Poisson statistics: $p(n,\lambda) = \dfrac{\lambda^n e^{-\lambda}}{n!}$

Total number of predicted photons

$$\log\mathscr{L}(\{\alpha_k\}) = \sum_{i\in P}\log J(E,\vec{p};\{\alpha_k\}) - \Lambda_{tot}(\{\alpha_k\})$$

Set of bins with an observed photon

D. Bastieri - CTA Consortium Meeting, Warsaw, 24 September 2013 13/18

MLA &LMA

NVIDIA GPU RESEARCH CENTER

- Maximize the likelihood, given the data
- How to reduce CPU↔GPU data transfer?
- Levenberg-Marquardt vs. MINUIT
  see also de Naurois & Rolland
  arXiv:0907.2610
- Minimizer resident in GPU memory

Filtering Events

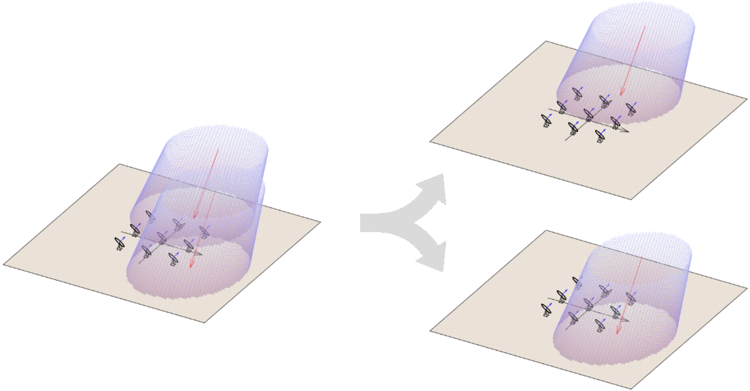
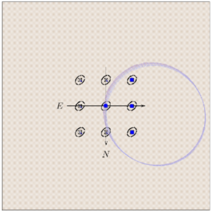
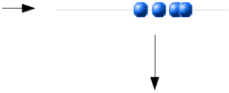


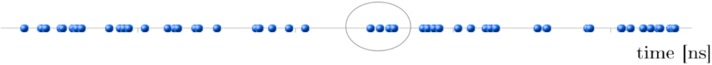
Figure 10: Schematic representation of the filtering concept.

Events

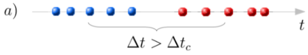


1	6	0	14065360
4	8	0	14208038
4	0	0	14208053
4	6	0	14208057
4	7	0	14208074
1	2	0	14392388

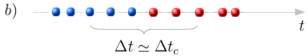




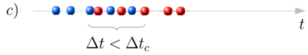
Now we are able to define a time relationship between two events:



a) *Far Events* if $\Delta t > \Delta t_c$,

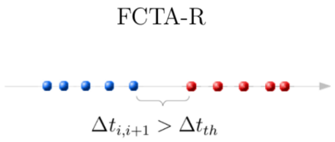


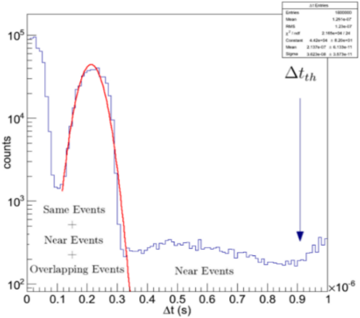
b) *Near Events* if $\Delta t \simeq \Delta t_c$

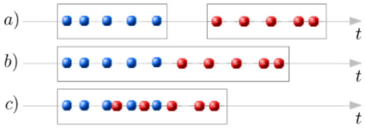


c) *Overlapping Events* if $\Delta t < \Delta t_c$.

FCTA-R: Minimum Large Threshold



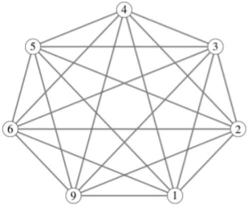




Distribution of Entries elapsed times $\Delta t_{i,i+1}$ in log scale.

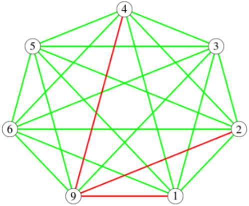
FCTA-N: Network Filter Algorithm

$$\Lambda = \begin{pmatrix} 0 & \lambda_{01} & \lambda_{02} & \cdots & \lambda_{0N} \\ \lambda_{10} & 0 & \lambda_{12} & \cdots & \lambda_{1N} \\ \lambda_{20} & \lambda_{21} & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_{N0} & \lambda_{N1} & \cdots & \cdots & 0 \end{pmatrix} \longrightarrow$$



We are able to map the Λ-matrix in a graph in wich the nodes are the Entries and the edges are the λ_{ij}. This kind of graph is called complete Network.

$$\Lambda = \begin{pmatrix} 0 & \lambda_{01} & \lambda_{02} & \cdots & \lambda_{0N} \\ \lambda_{10} & 0 & \lambda_{12} & \cdots & \lambda_{1N} \\ \lambda_{20} & \lambda_{21} & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_{N0} & \lambda_{N1} & \cdots & \cdots & 0 \end{pmatrix} \longrightarrow$$



Clustering Algorithm

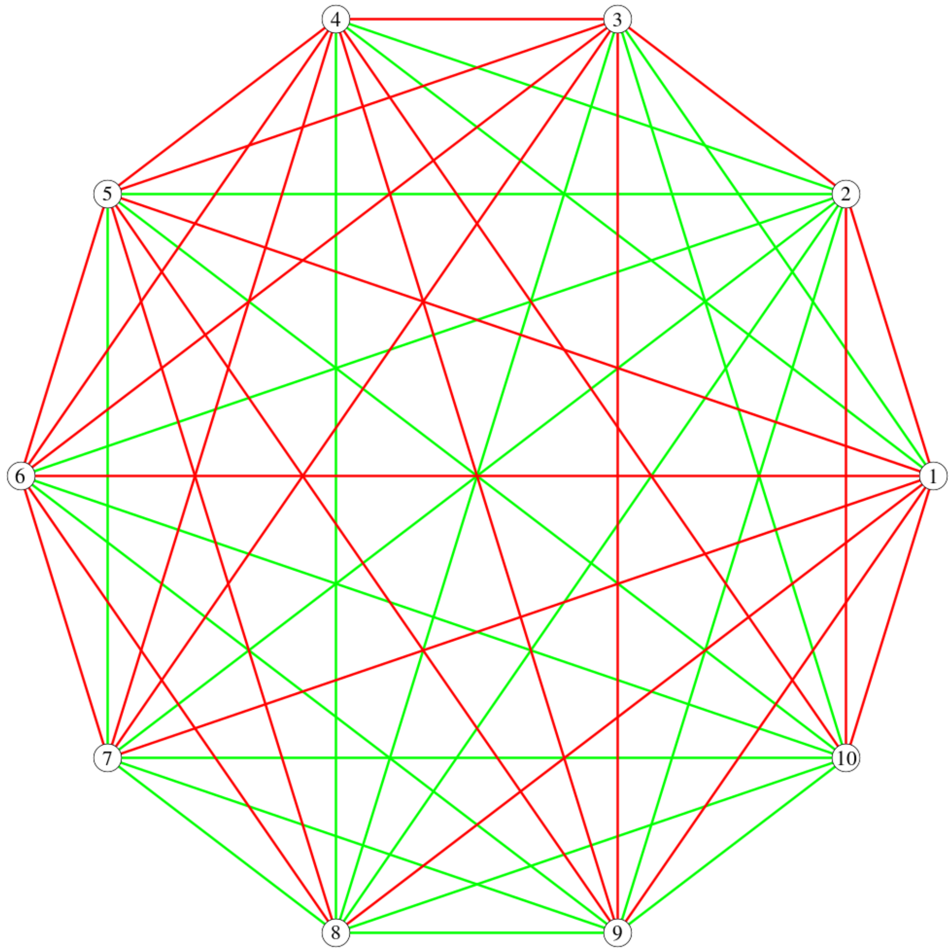


Figure 12: Images of a clustering process obtained with FCTA-N.

Clustering Algorithm

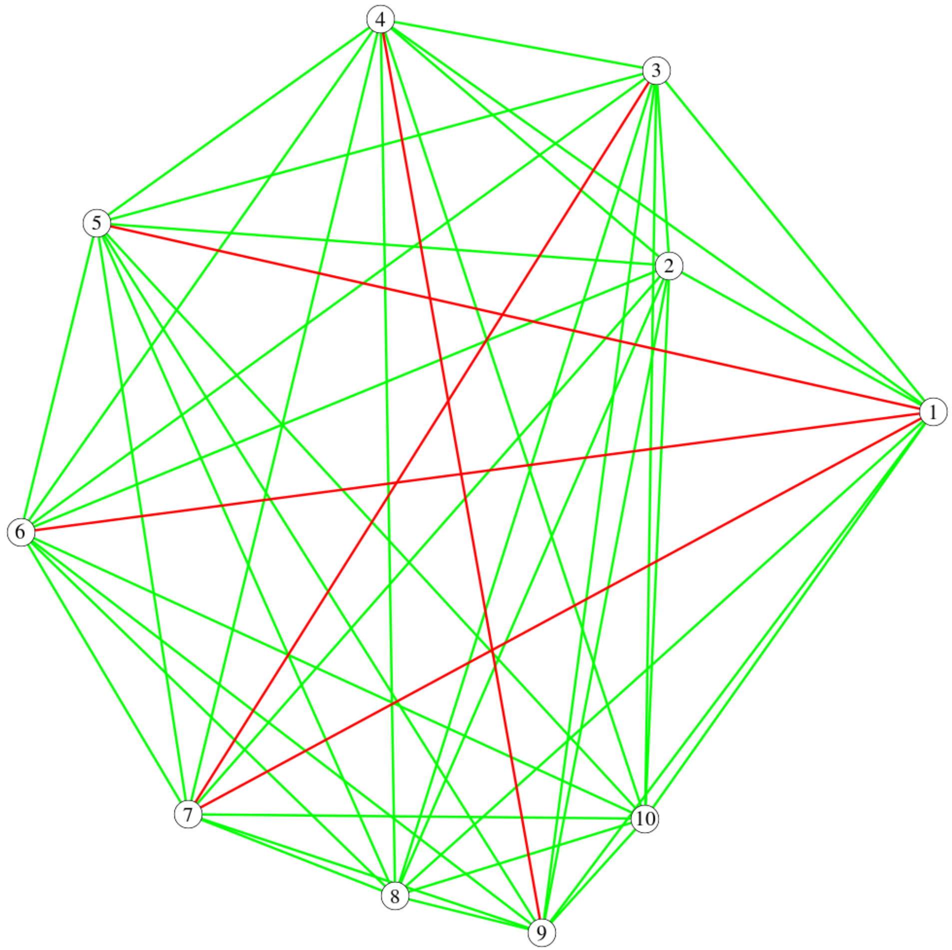


Figure 12: Images of a clustering process obtained with FCTA-N.

Clustering Algorithm

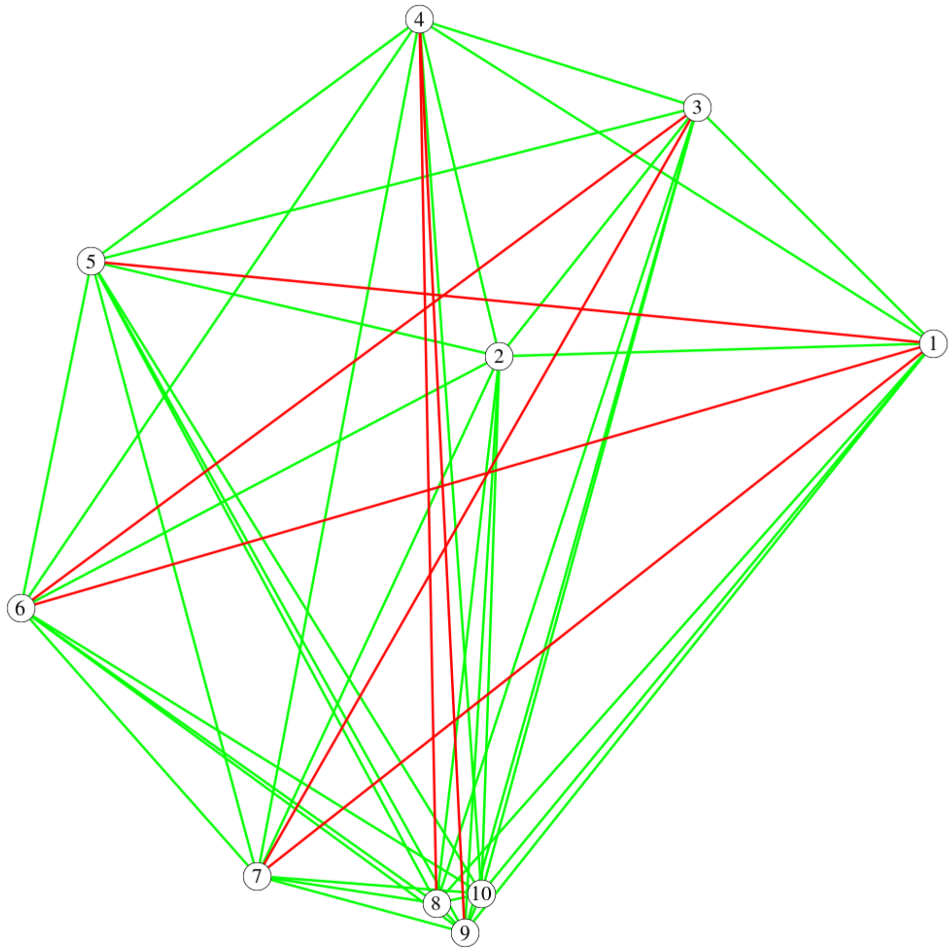


Figure 12: Images of a clustering process obtained with FCTA-N.

Clustering Algorithm

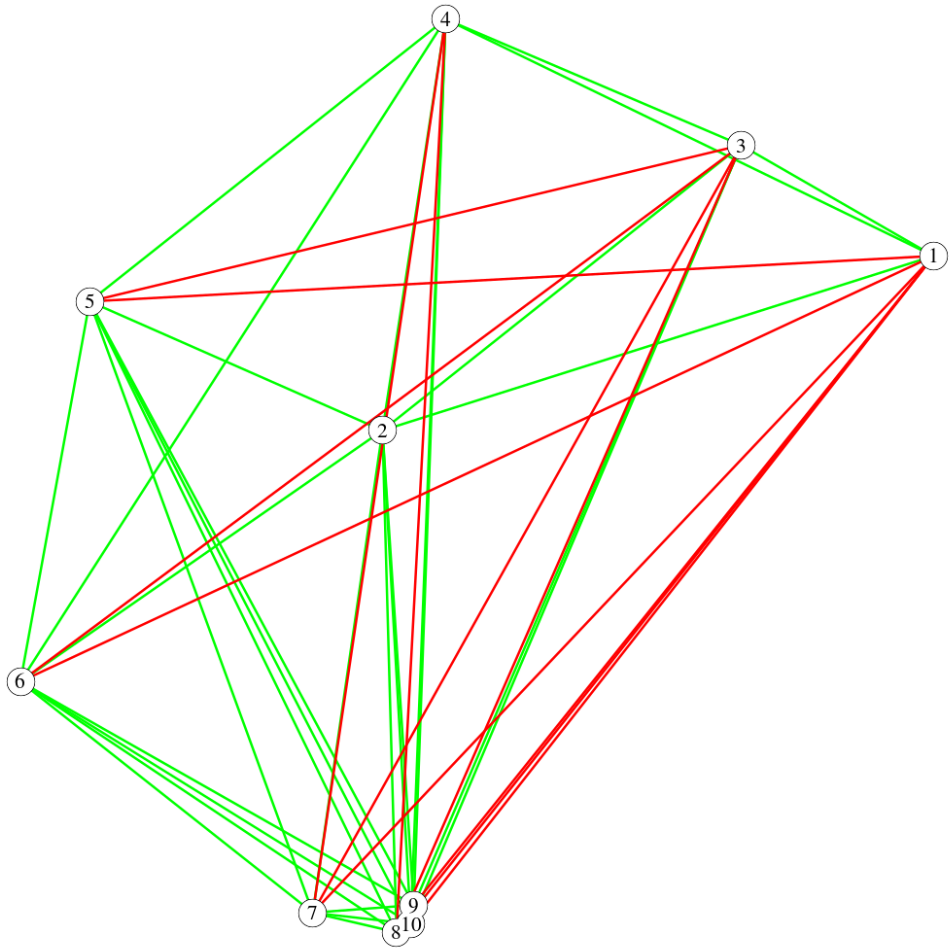


Figure 12: Images of a clustering process obtained with FCTA-N.

Clustering Algorithm

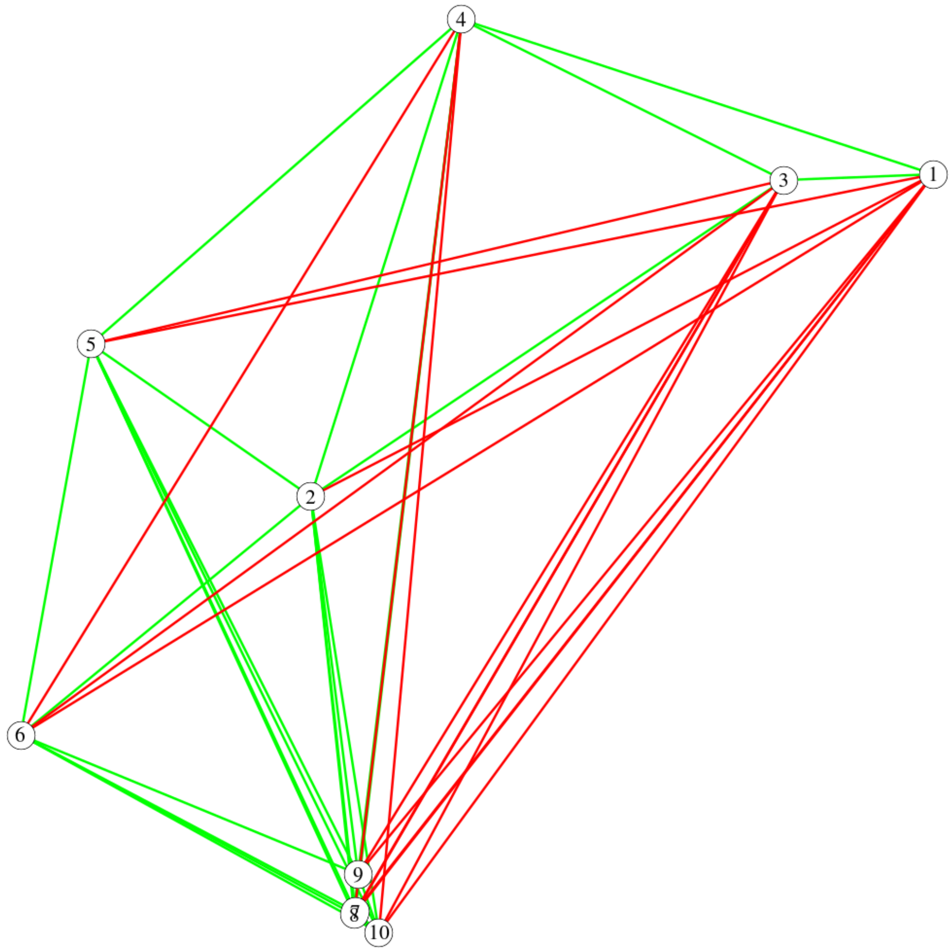


Figure 12: Images of a clustering process obtained with FCTA-N.

Clustering Algorithm

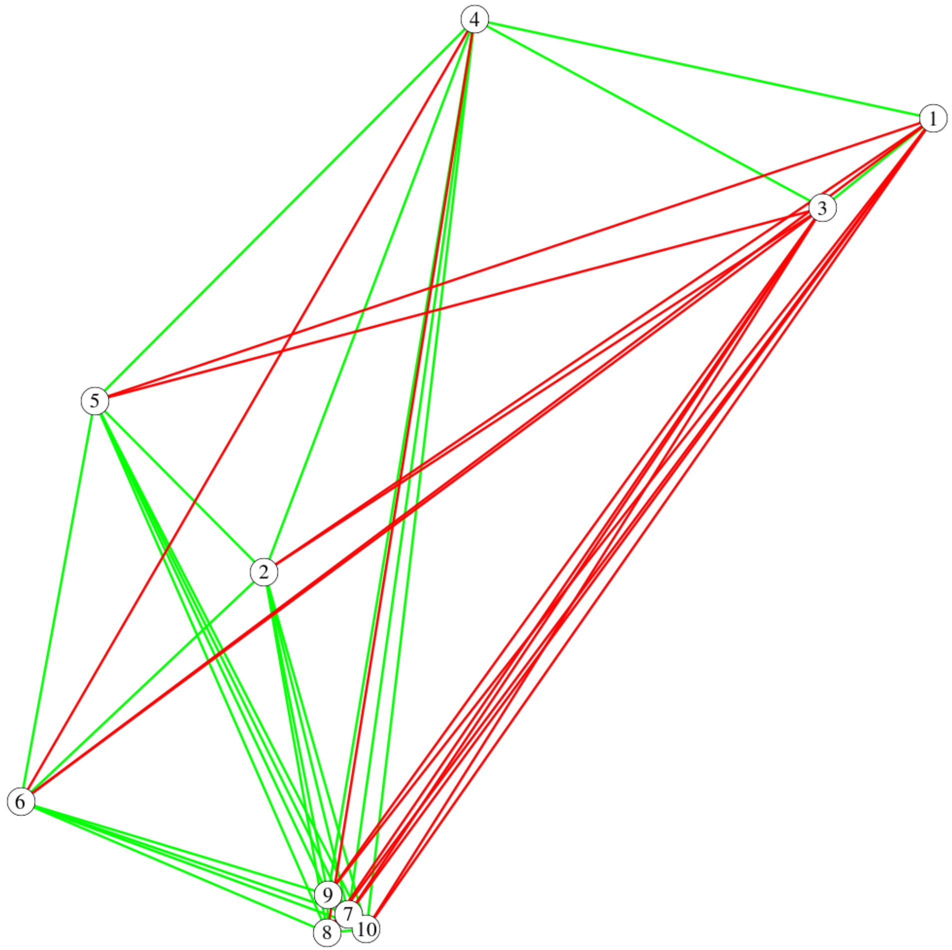


Figure 12: Images of a clustering process obtained with FCTA-N.

Clustering Algorithm

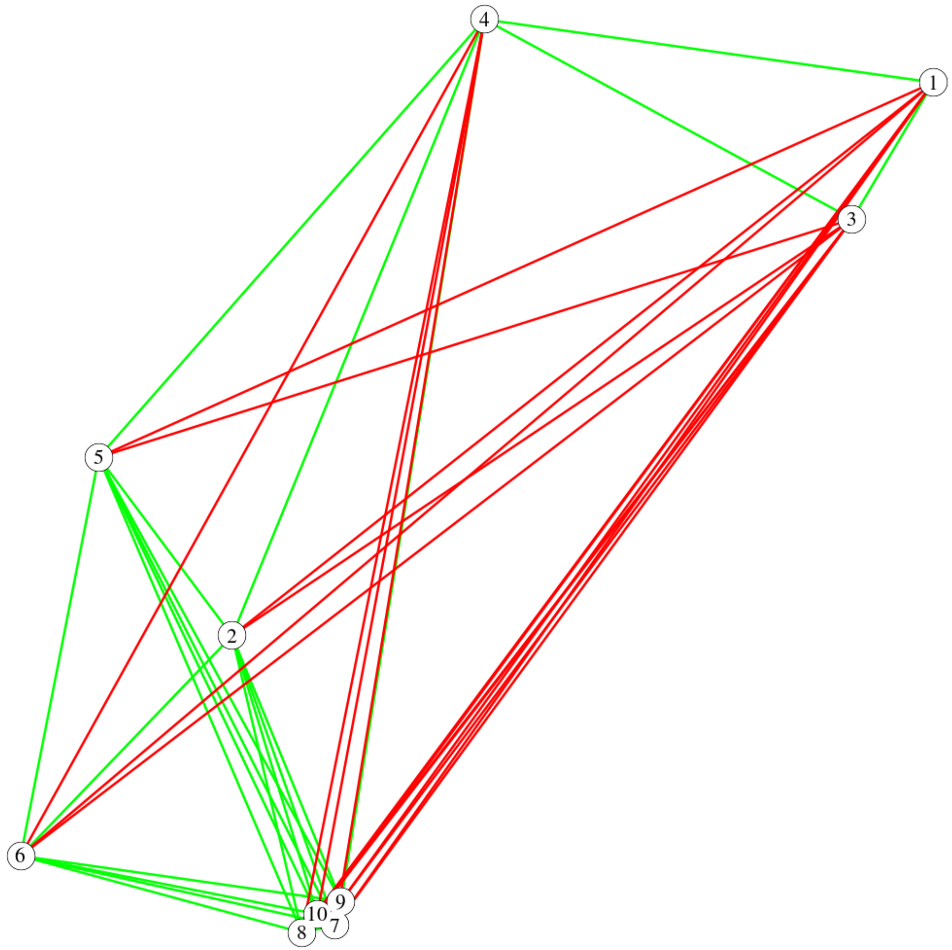


Figure 12: Images of a clustering process obtained with FCTA-N.

Clustering Algorithm

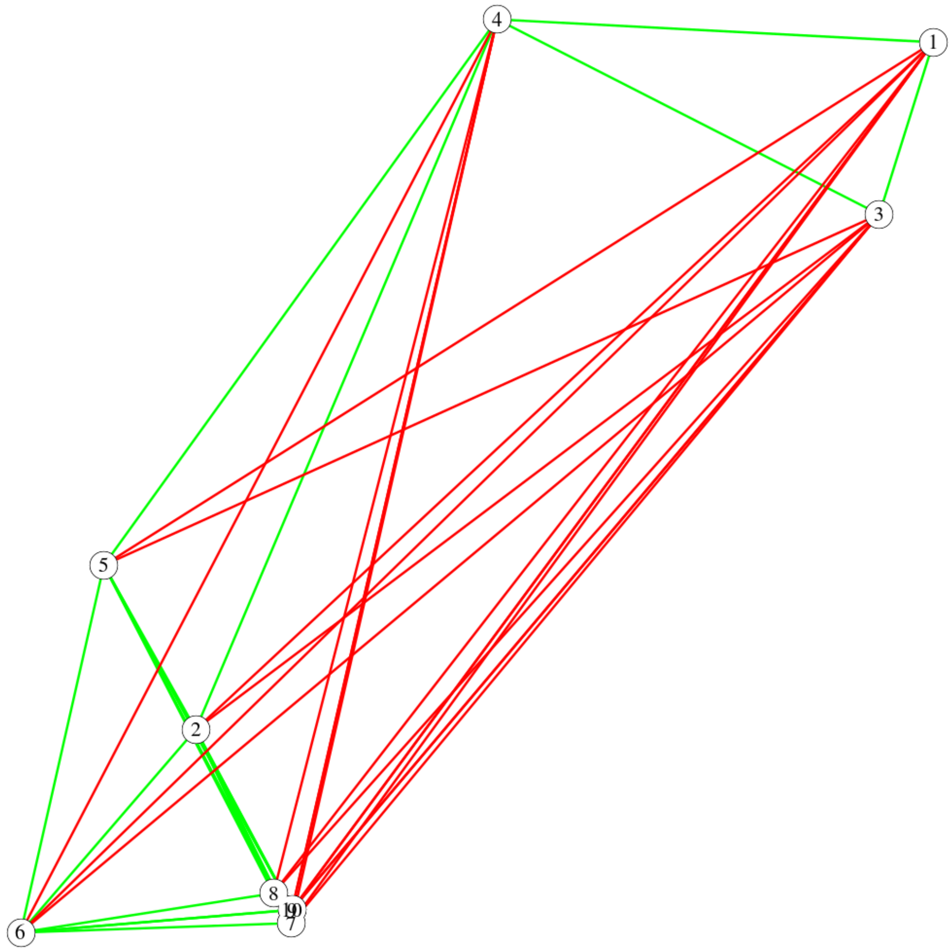


Figure 12: Images of a clustering process obtained with FCTA-N.

Clustering Algorithm

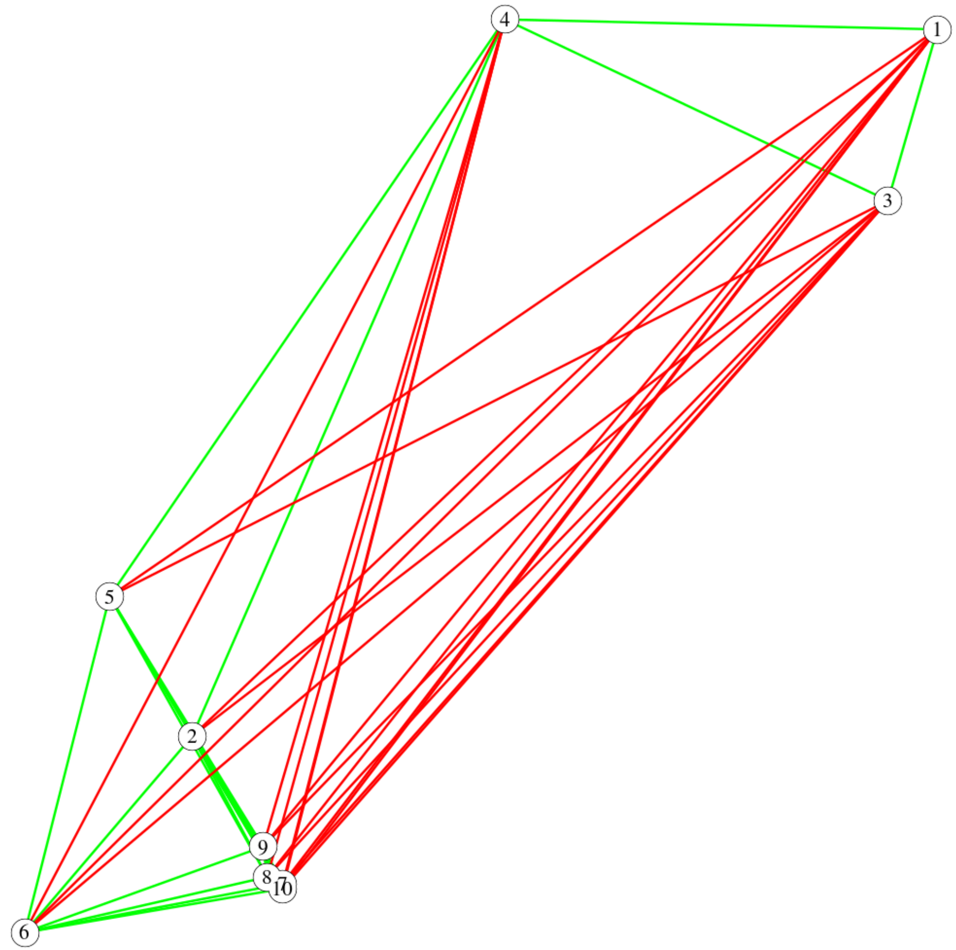


Figure 12: Images of a clustering process obtained with FCTA-N.

Clustering Algorithm

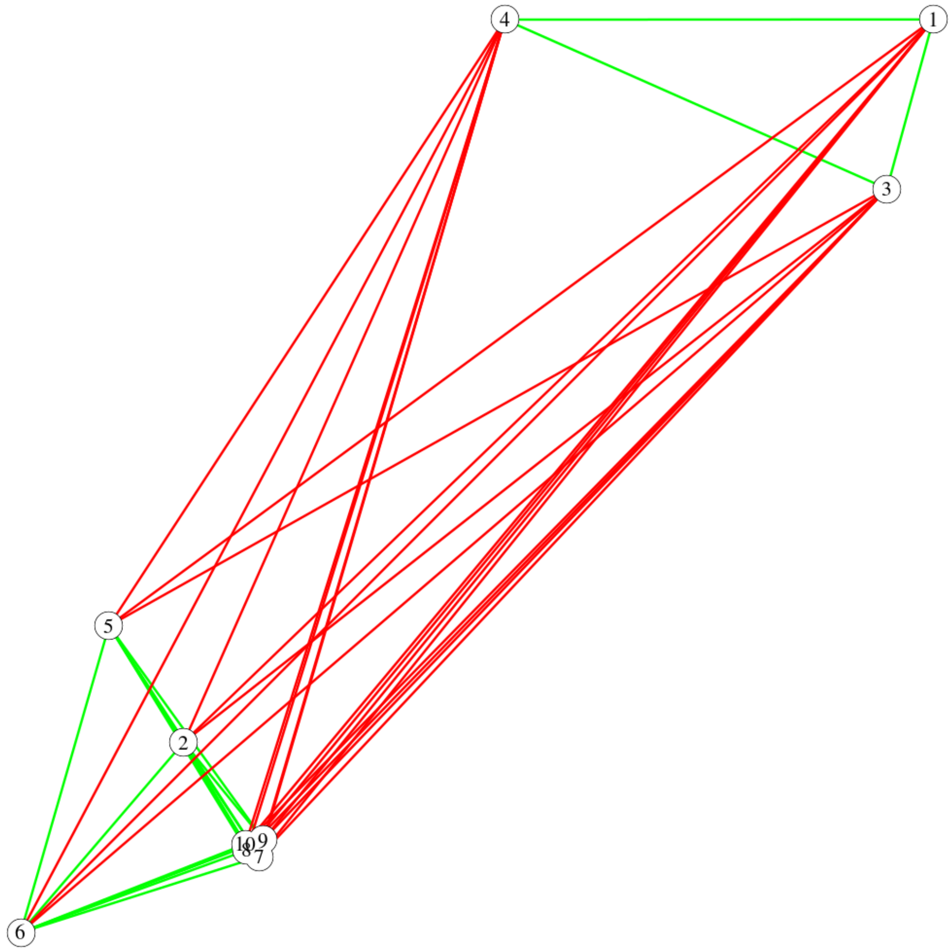


Figure 12: Images of a clustering process obtained with FCTA-N.

Clustering Algorithm

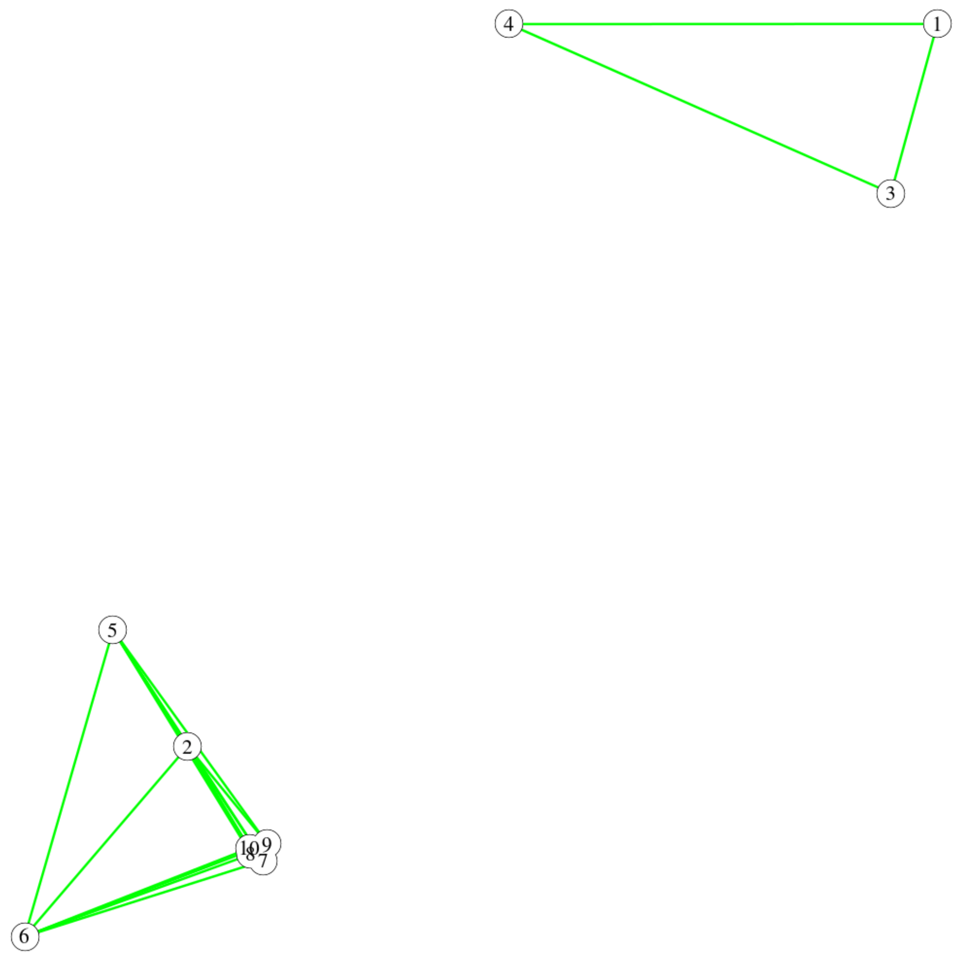
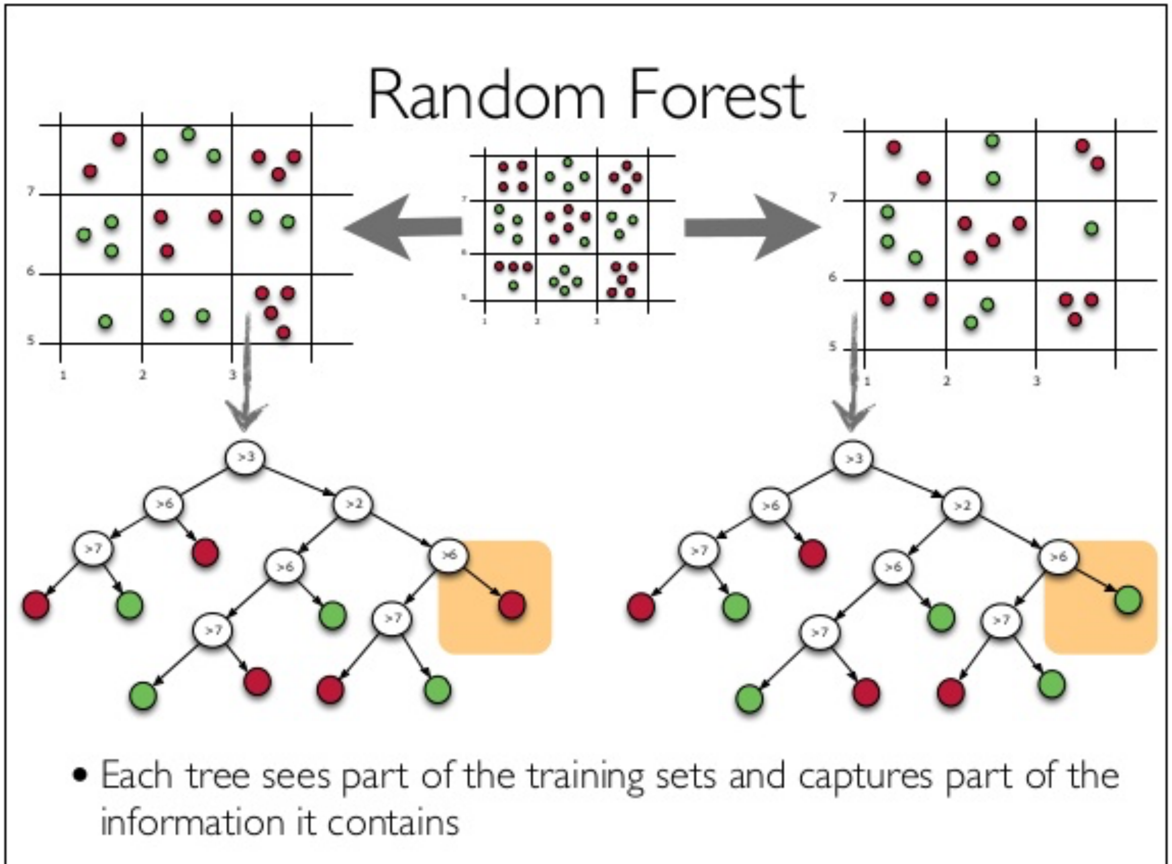


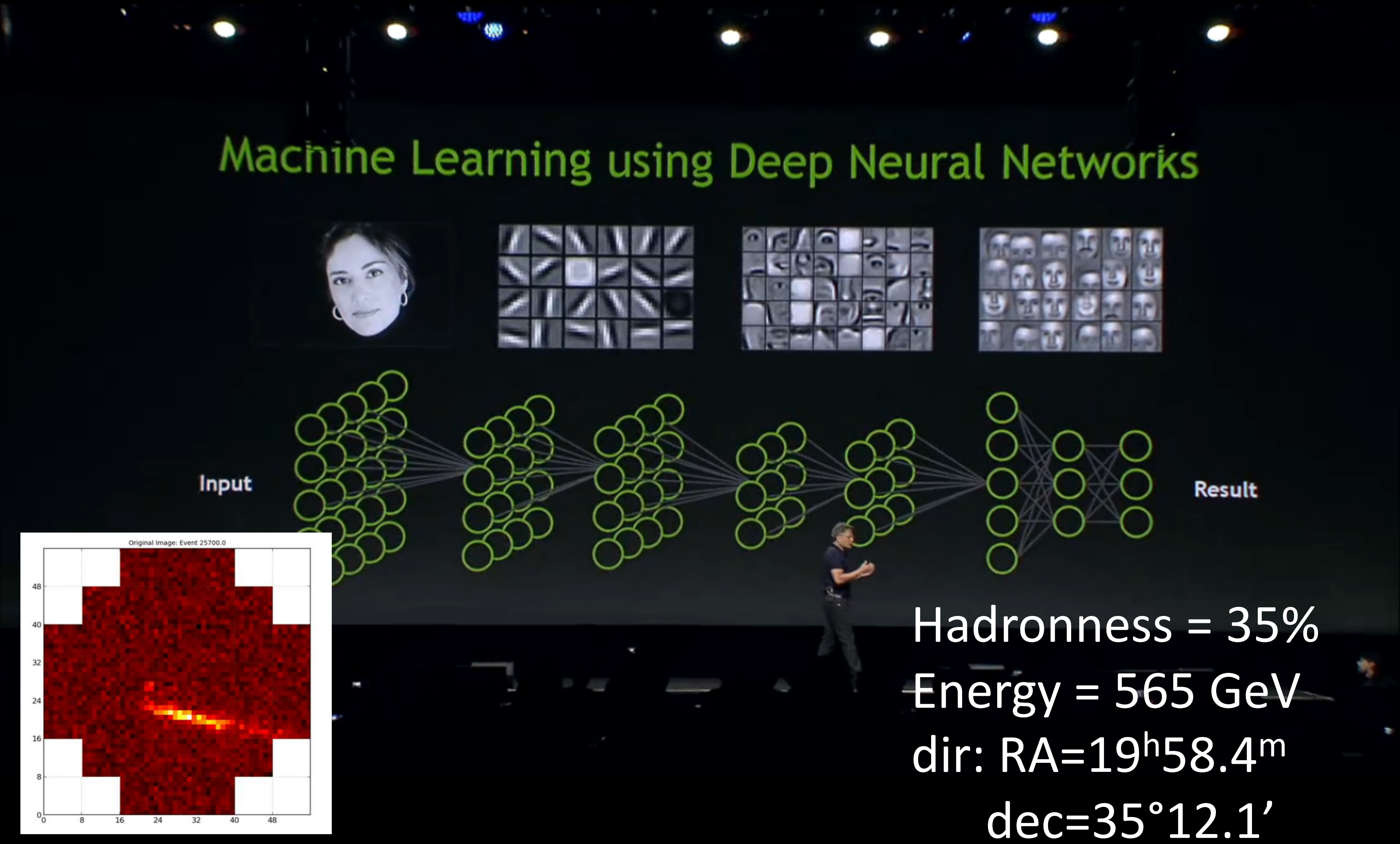
Figure 12: Images of a clustering process obtained with FCTA-N.

What's next?



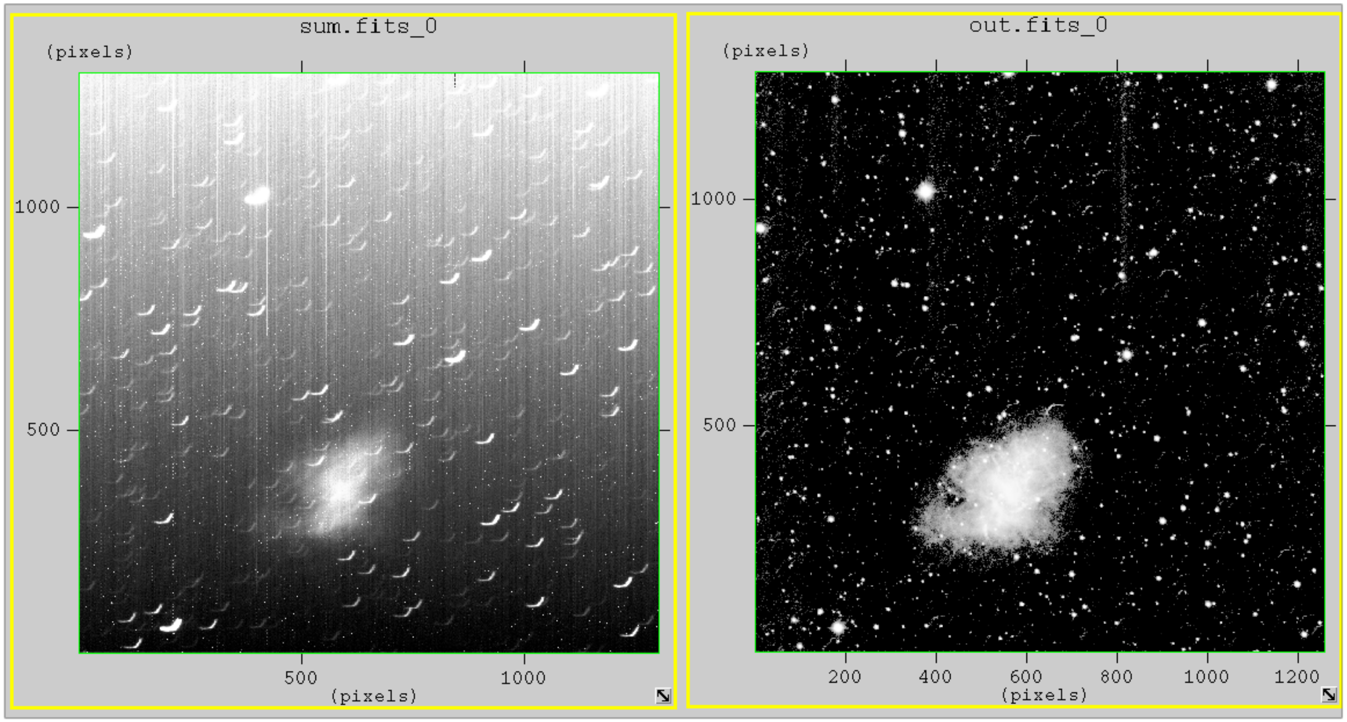
Hadronness
Energy estimation
Incoming direction

What’s next? DNN!



Hadronness = 35%
Energy = 565 GeV
dir: RA=$19^h58.4^m$
dec=35°12.1’

MaxEnt 1985
(6-mon proc)



MaxEnt
2014
(6-sec proc)

D. Bastieri & D. Costantin

2016 ICIOT-5GMT
GZ, China, November 28, 2016

Conclusion

- ASTRI, CTA, and Gamma-Ray Astronomy at large, are an optimal test-ground for Low-Power Computing and High-Throughput Computing.
- Gamma-Ray Astronomy from ground needs a lot of computing power
  - Mostly in the realm of HTC (calibration, cleaning, image momenta…)
  - Calibrations may be done with ARM
  - Calibrations may be done with FPGA
    (lower Watts, but worth the additional burden?)
  - Additional analyses are feasible on NVIDIA Jetson T∗1
  - Complete analysis chain working on NVIDIA Jetson TK1 (@2× max event rate)
  - Event builder: with a special concern about *purity*
- Where to go next for Gamma-Ray Astronomy?
  - Algorithms *select* the optimal hardware architecture
  - Data crunching: integration with FPGA and DSP
  - MLA: still trying to find a *resident* minimizer
  - What about DNN? Comparisons with Classification Trees started