Florian Ferreira
PhD student

COMPASS
Efficient GPU based AO simulations

3rd AO RTC Workshop

January 26 - 27th 2016
Observatoire de Paris

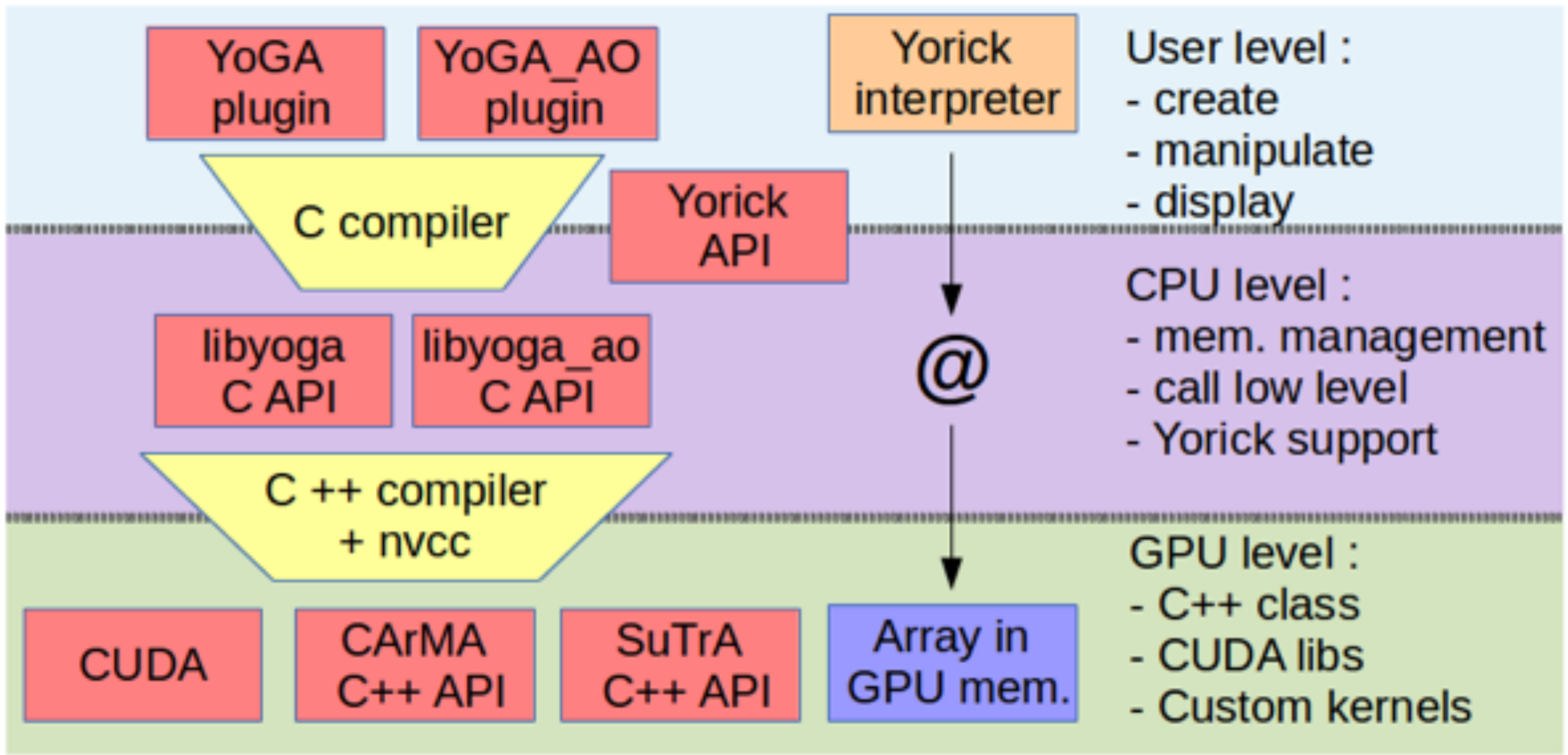
SUMMARY

- What is COMPASS ?
- Features
- Performance
- Further development

WHAT IS COMPASS ?

- COMPuting Platform for Adaptive opticS System
- End-to-end AO simulation platform
- GPU acceleration
- ELT scale

WHAT IS COMPASS ?



2 user interfaces:

Yorick: development stopped…

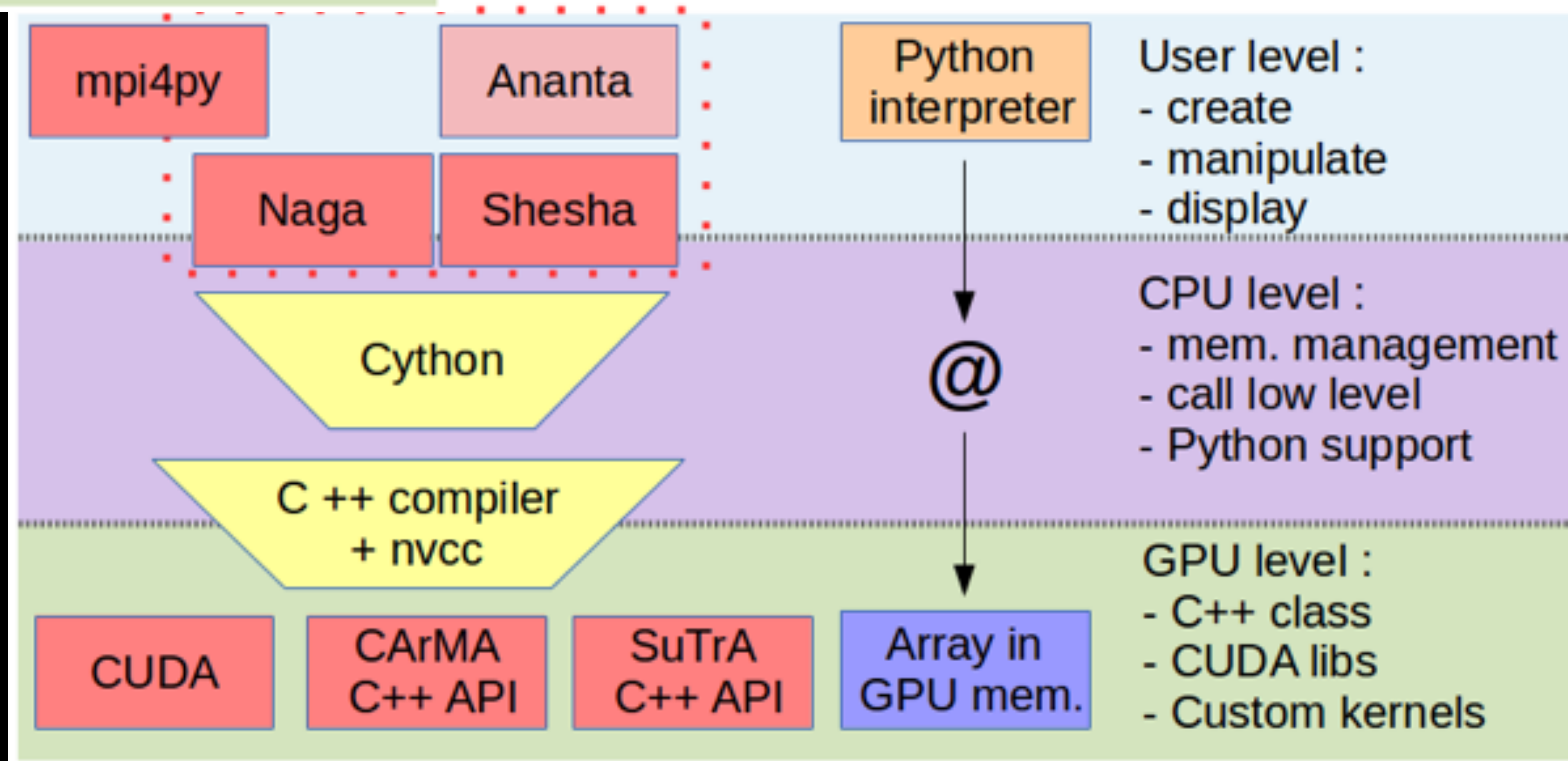
Python: …and translated in Python for long term maintenance plan

Main computations relies on GPU:

CArMA: C++ Api for Massively parallel Applications

SuTrA: Simulation Tool for Adaptive optics

Use optimized libraries such as CUBLAS, CUFFT, MAGMA…



SIMULATION PROCESS

User parameters file

CPU

Initializations

WFS

Atmosphere

Target

DMs

RTC

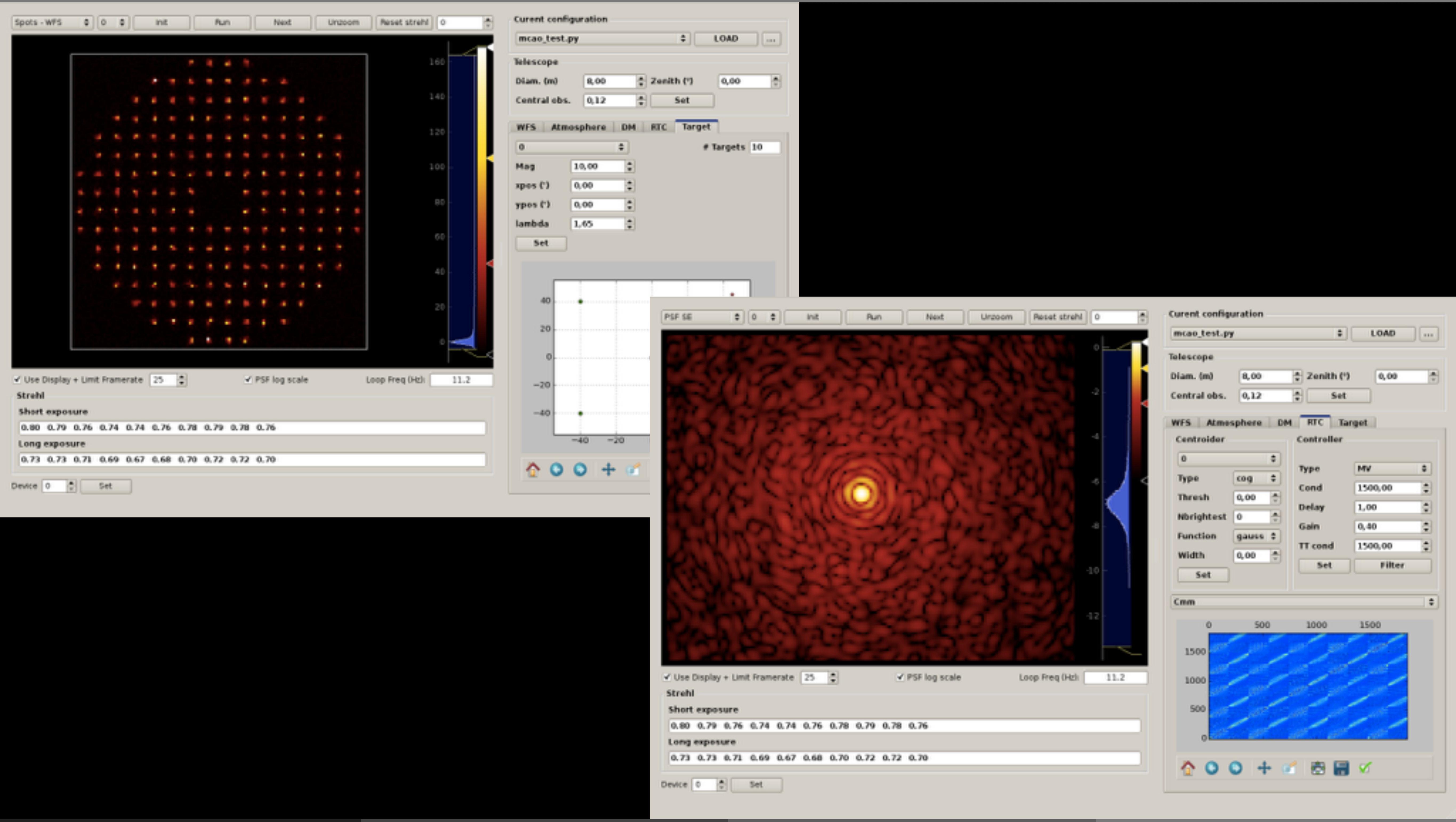
CPU + GPU allocations

AO Loop

- move atmos
- phase on WFS
- WFS image
- Centroids
- Commands
- DM shape

GPU

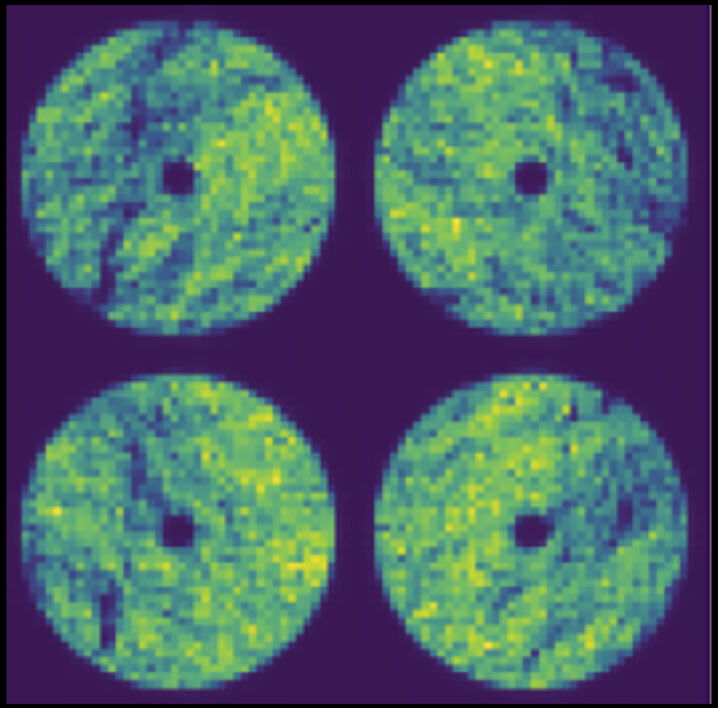
PYQTGRAPH GUI

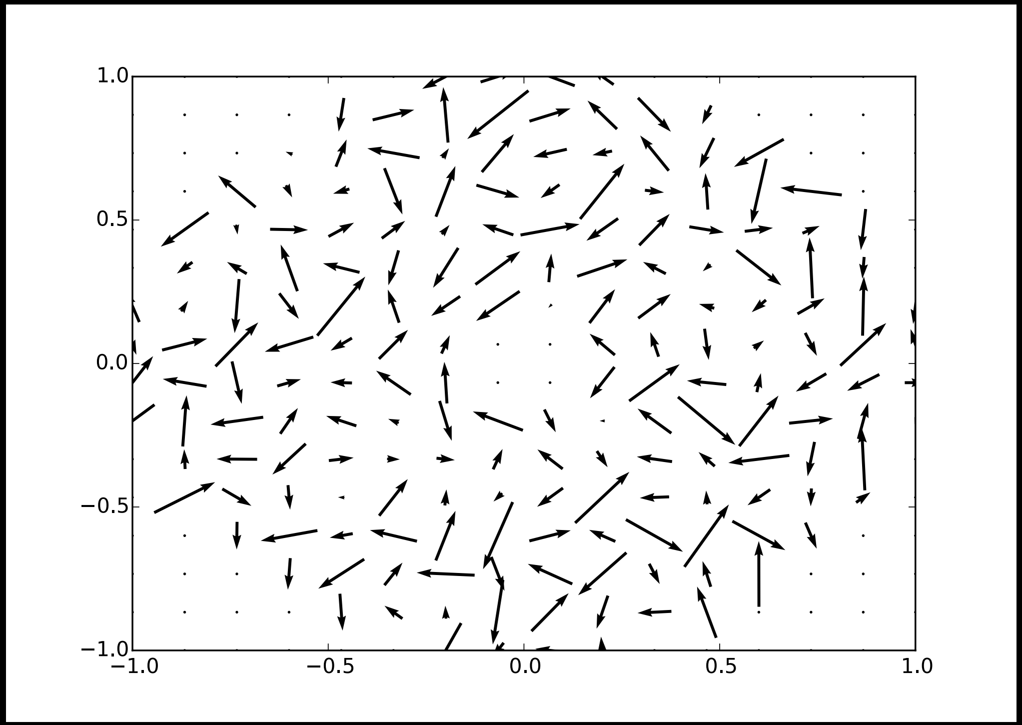


FEATURES

Wavefront Sensor models:

- Shack-Hartmann
- Pyramid
- Roof
- Laser Guide Star

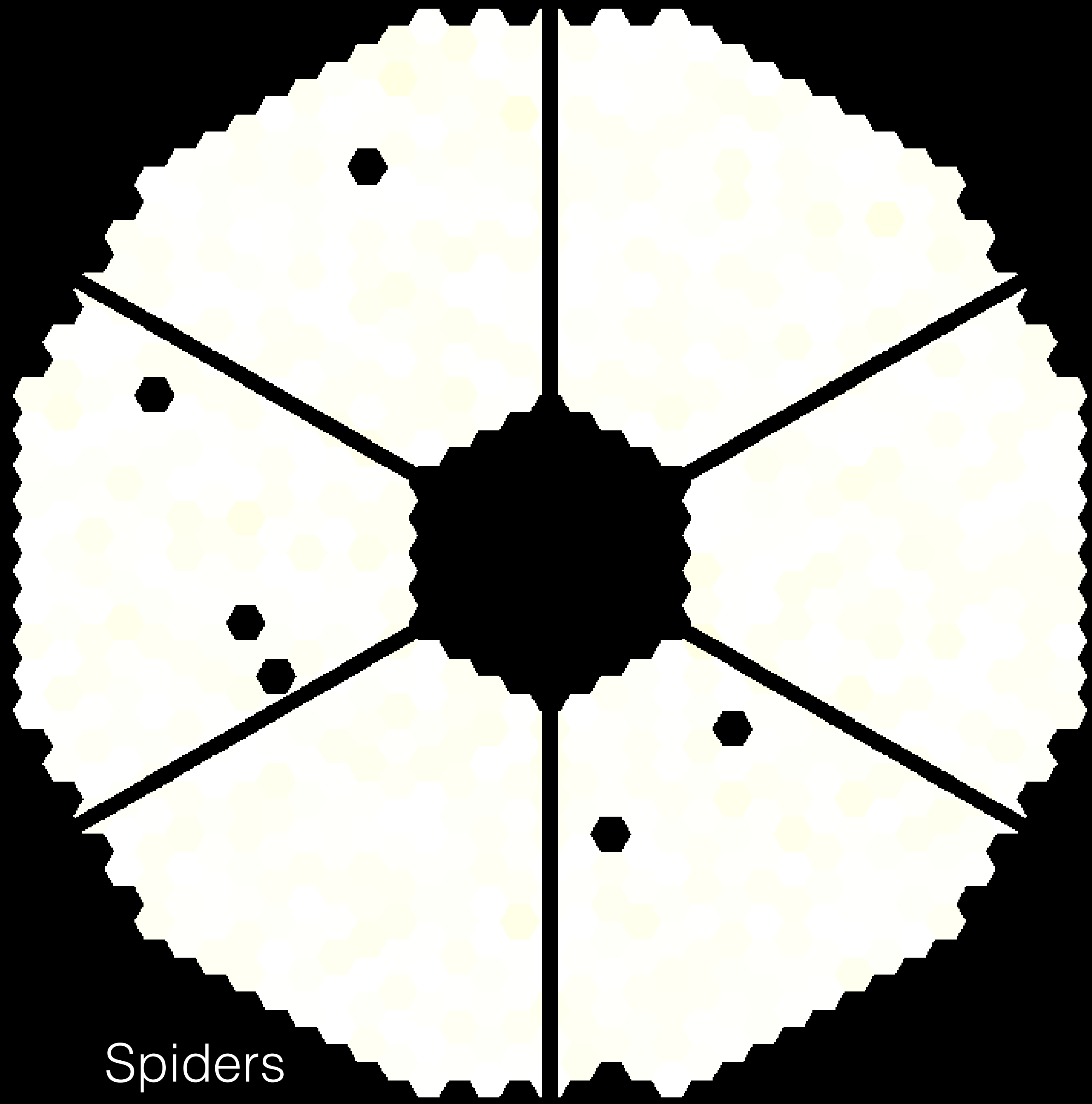




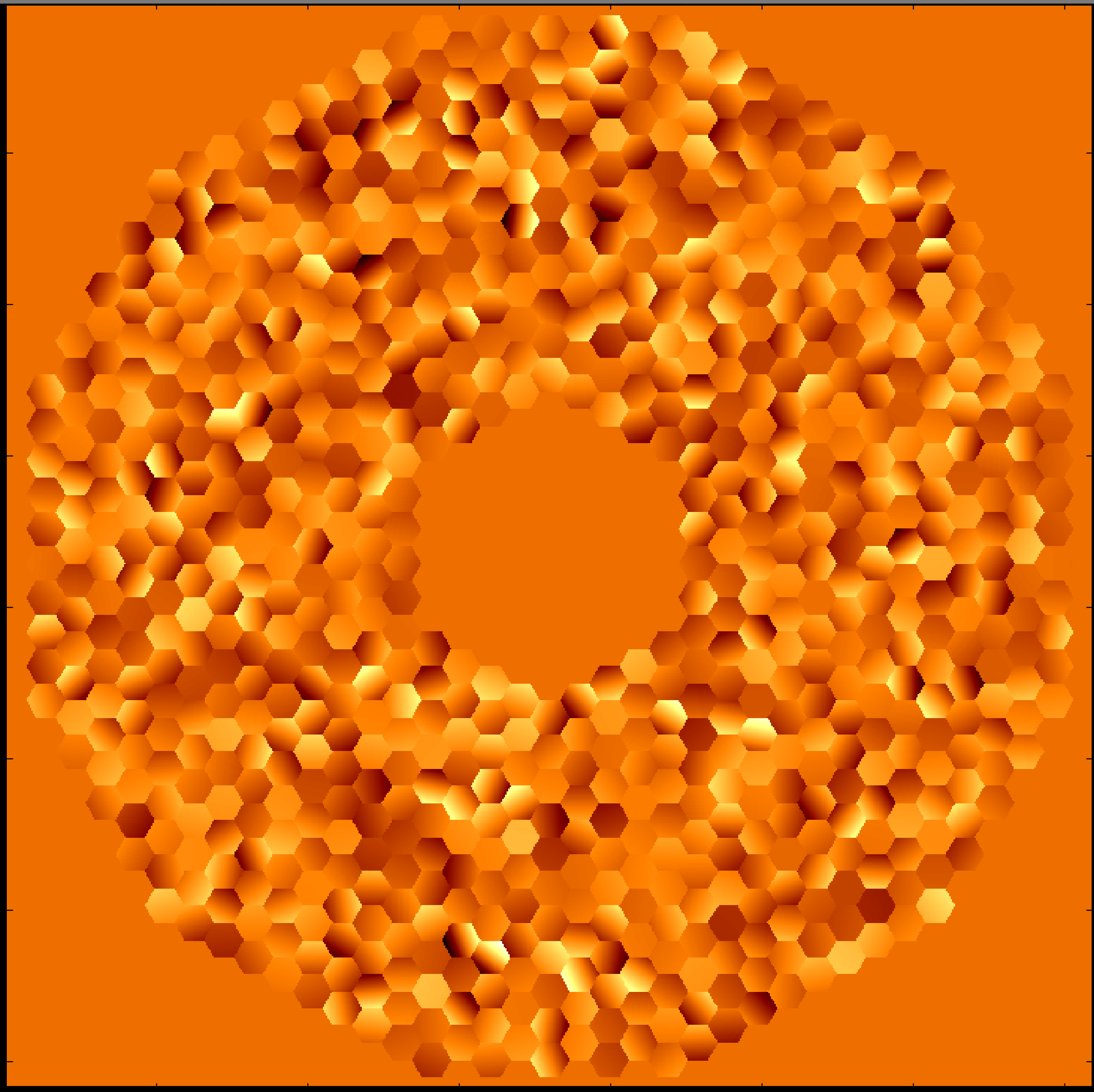
Centroiding methods:

- Center of gravity (cog)
- Thresholded cog
- Weighted cog
- Brightest pixels
- Correlation

E-ELT FEATURES

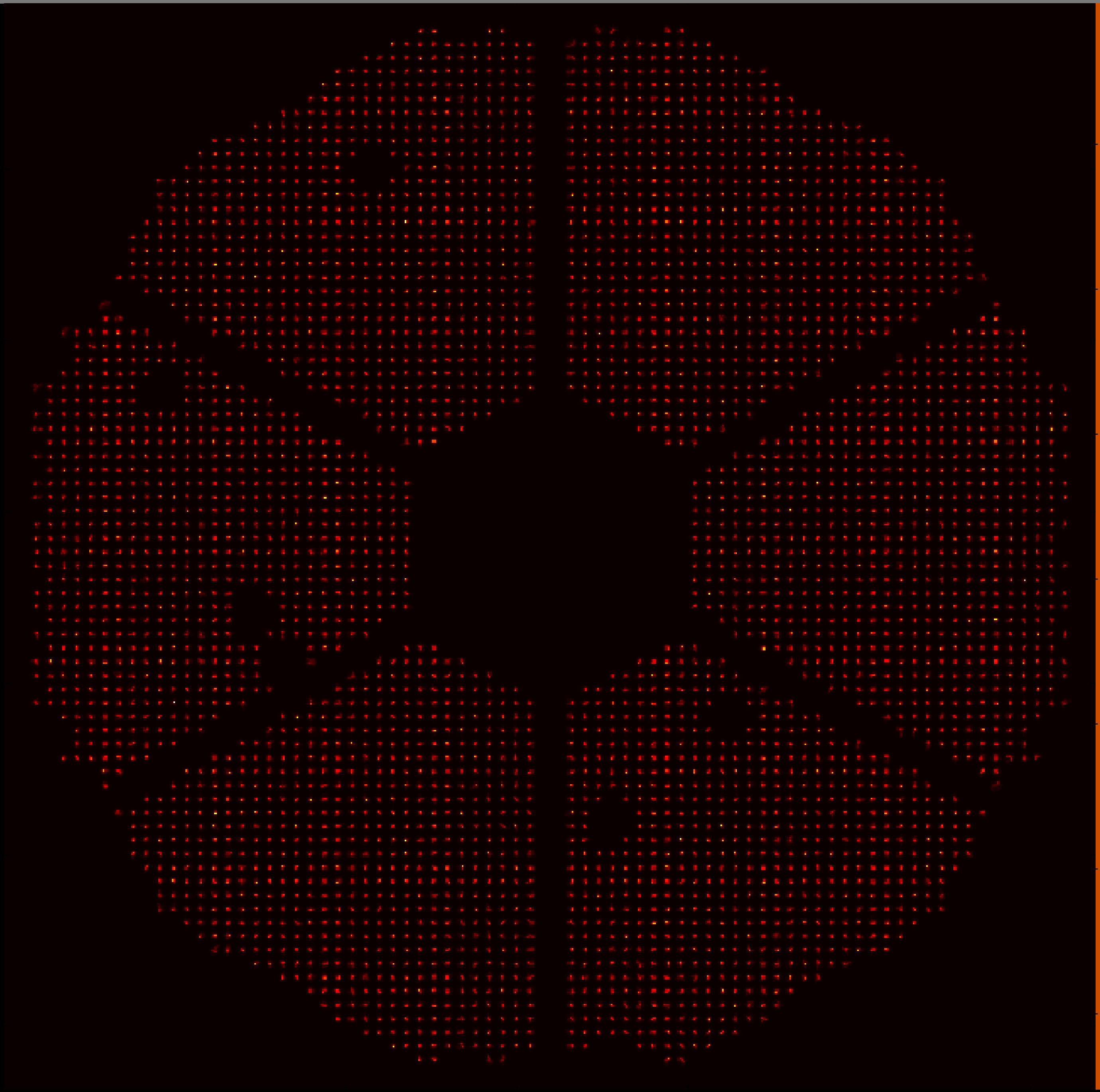


Spiders
Missing segments
Reflectivity

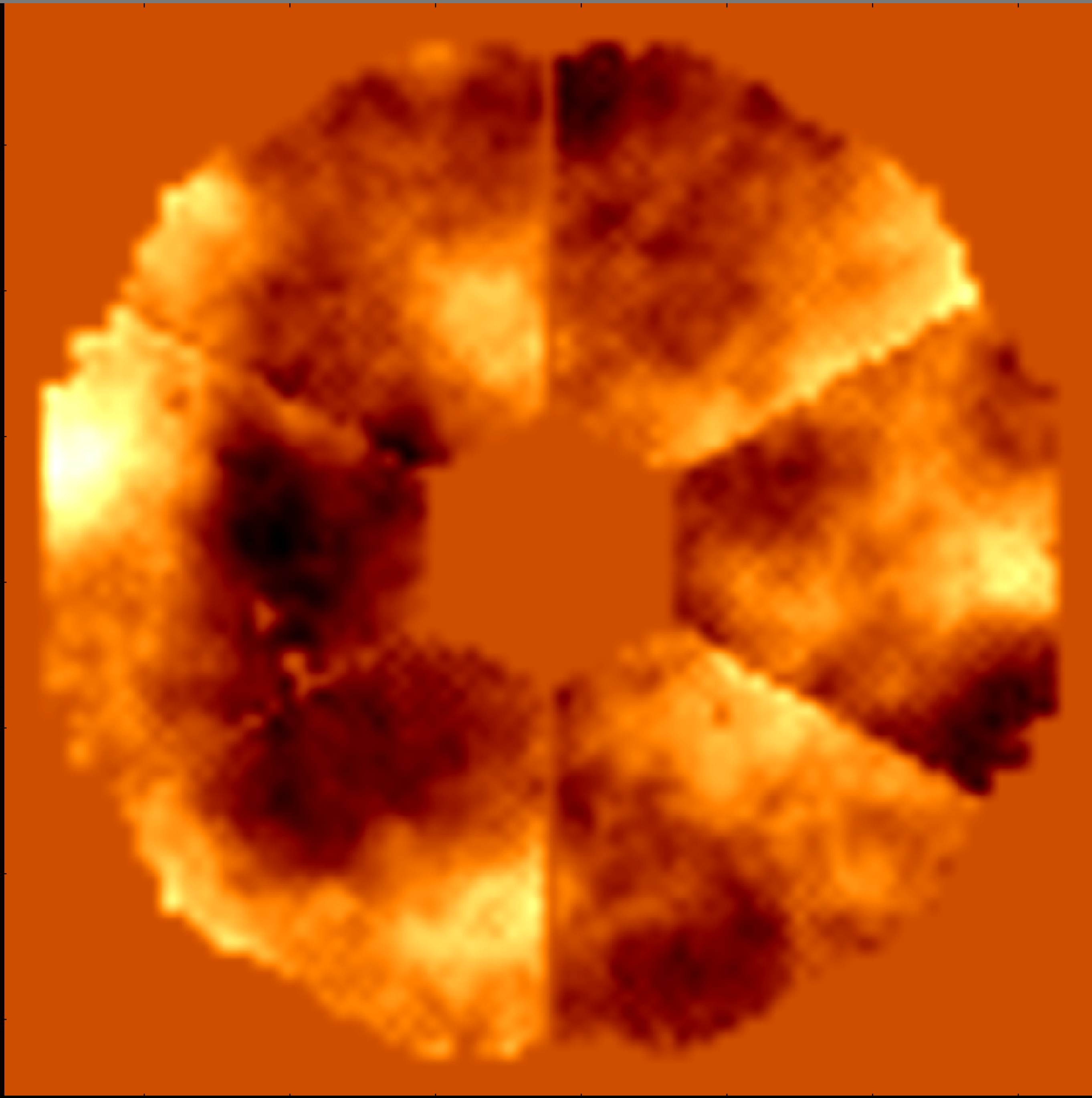


Phase aberrations

E-ELT FEATURES



WFS



DM

REAL-TIME CONTROLLERS

Least Square

$$R = (D^t D)^+ D^t$$

$$\vec{v}[k] = \vec{v}[k-1] - g\, R\, \vec{s}[k]$$

Modal optimization

Compute optimum modal gains from measurements

$$v[\vec{k}] = \vec{v}[k-1] - \vec{g}\, R\, \vec{s}[k]$$

Minimum variance

$$R = C_{\phi m}(C_{mm} + C_n)^{-1}$$

POLC:

$$\vec{s_{ol}}[k] = \vec{s}[k] - D\,(a\,\vec{v}[k-2] + (1-a)\,\vec{v}[k-1])$$

Commands:

$$\vec{v}[k] = (1-g)\vec{v}[k-1] + g\, R\, \vec{s_{ol}}[k]$$

REAL-TIME CONTROLLERS

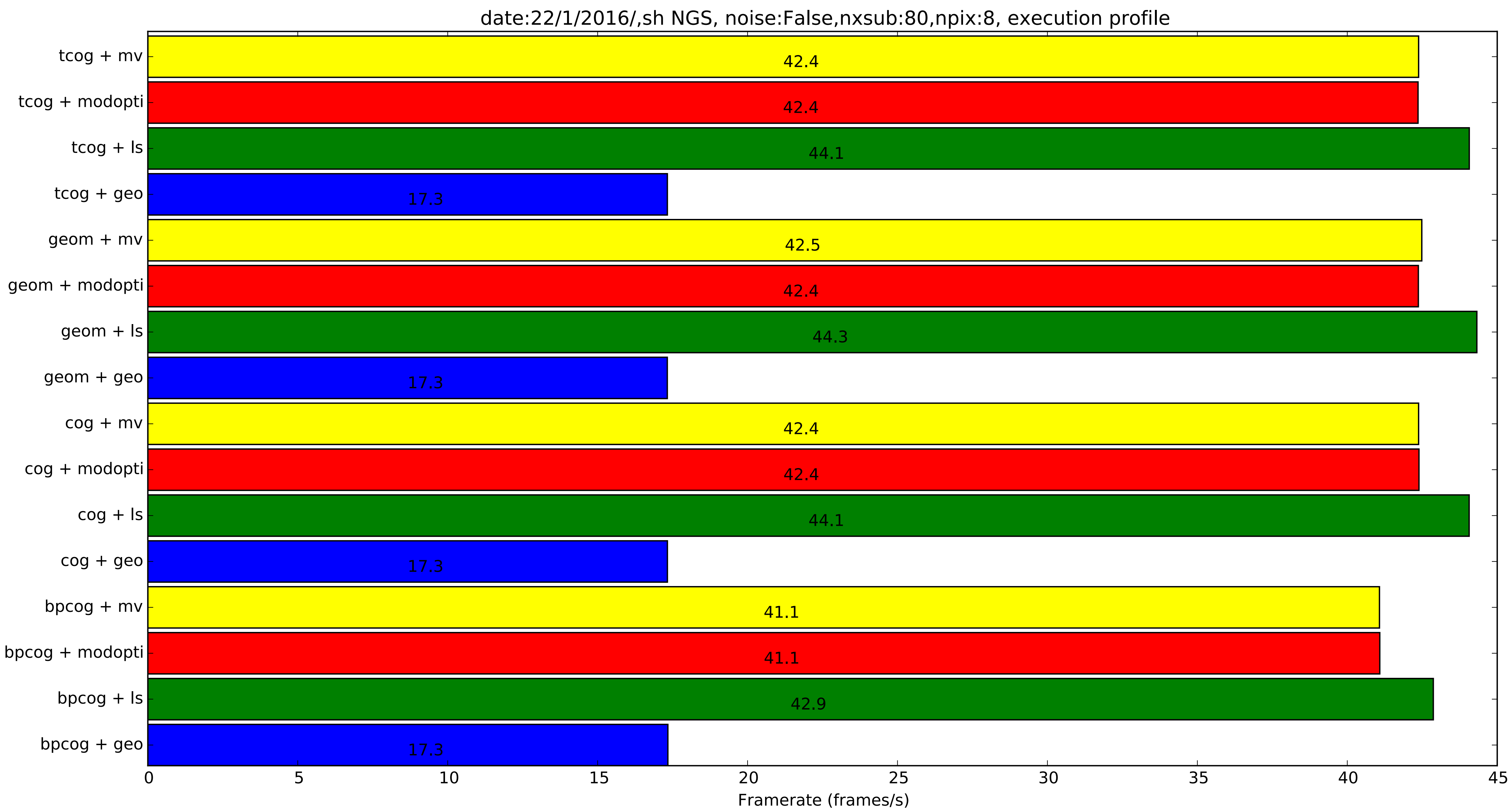
And also

- CuReD (Cumulative Reconstructor with Domain decomposition)
- « Geometric » : direct projection of the phase onto the DM

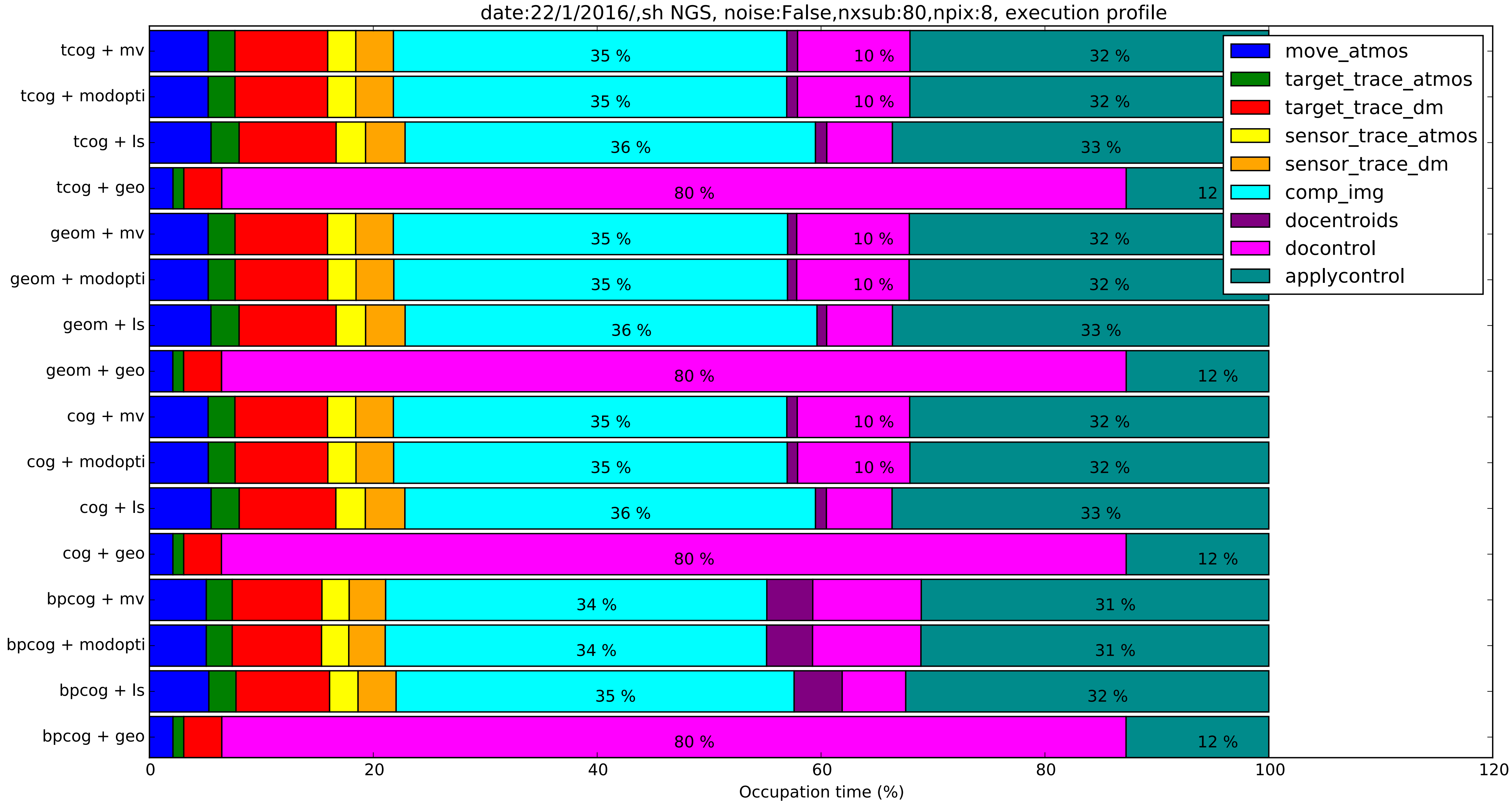
GPU optimized algorithms

- Real-time control operations are fully computed on GPU
- Hardware independent: tested on various Nvidia architectures
- « RTC ready »

SCAO PERFORMANCE



SCAO PERFORMANCE



SCAO PERFORMANCE

AO end-to-end simulation tools

- OCTOPUS: CPU-based, C-code
- YAO: CPU-based, C-code
- COMPASS: GPU-based, C++/CUDA code

Comparison

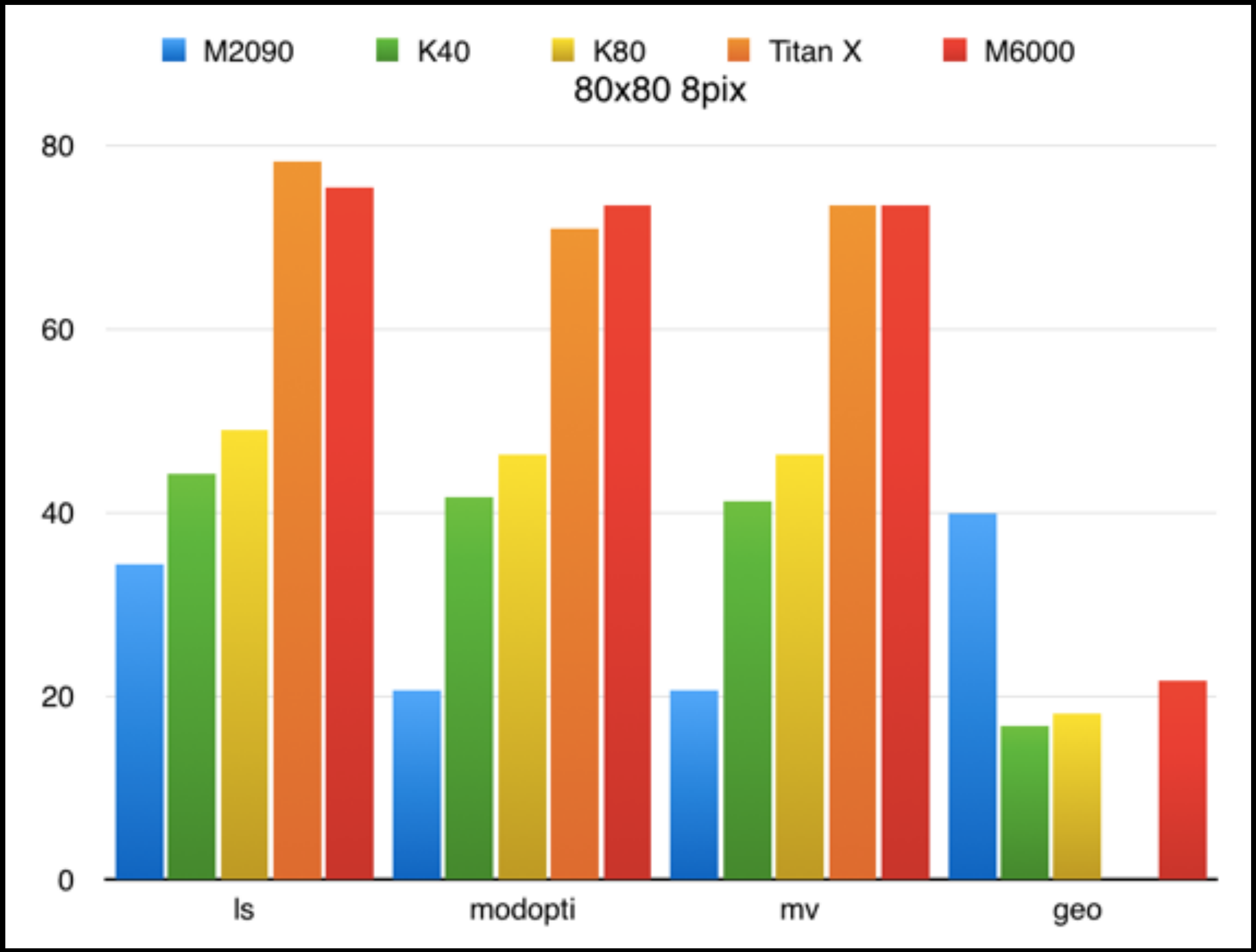
- ELT SCAO
- 38m, 38x38 subap
- LS reconstruction
- Tesla C2050

Tool	Init	Run	it/s
OCTOPUS	2.5 min	8.3 min	1
YAO	2 min	43 s	11.8
COMPASS	**36 s**	**5.2 s**	**96**

R. Wagner, A. Obereder, 2014

SCAO PERFORMANCE

Framerate



GPU	Arch	Cores
M2090	Fermi	512
K40	Kepler	2,880
K80	Kepler	2,496 *(x2)*
Titan X	Maxwell	3,072
M6000	Maxwell	3,072

https://en.wikipedia.org/wiki/List_of_Nvidia_graphics_processing_units

FURTHER DEVELOPMENTS

Error process

Computing on the fly a comprehensive error breakdown through the simulation

Multi-GPU distribution

Distribution over several GPUs and several nodes to reach ELT MCAO scale

FURTHER DEVELOPMENTS

SCAO MICADO

Numerical simulations for preliminary design studies

Green Flash

Develop and test RTC prototypes for ELT scale

Thank you !