

Beam holography

LOFAR2.0 development sprint review

Background

Calibration strategies for LOFAR2.0

- Aim:
 - Have (at least) **2 methods** for both **station** and **array** calibration
 - (previous meetings)
 - David McKenna discussed the classical station calibration method
 - Emma van der Wateren discussed imaging delays to assess the beamformer coherence

Background

Calibrate LOFAR's tied-array beams with holography

- Method described in **Salas, Brentjens, Bordenave, Oonk & Röttgering (2020, A&A)**
 - Determine **time delays between stations** in LOFAR core
 - LOFAR multibeaming allows to map voltage beam **faster** than a raster scan
 - ➔ Simultaneous & continuous beam calibration
- Same principles to **calibrate at station level** (Brentjens & Bordenave 2017: <https://weblectures.leidenuniv.nl/Mediasite/Play/b4f9b2beea6744138fd7ea4c540aeba51d?catalog=63487fe7-4485-43af-b2d9-de2c2ae2fb79>)

Background

Tied-array holography

- Determine **time delays between stations** in LOFAR core
- Two station sets:
- **Reference** station points at a bright compact source
- **Target** stations (array under test): multibeam at and around source

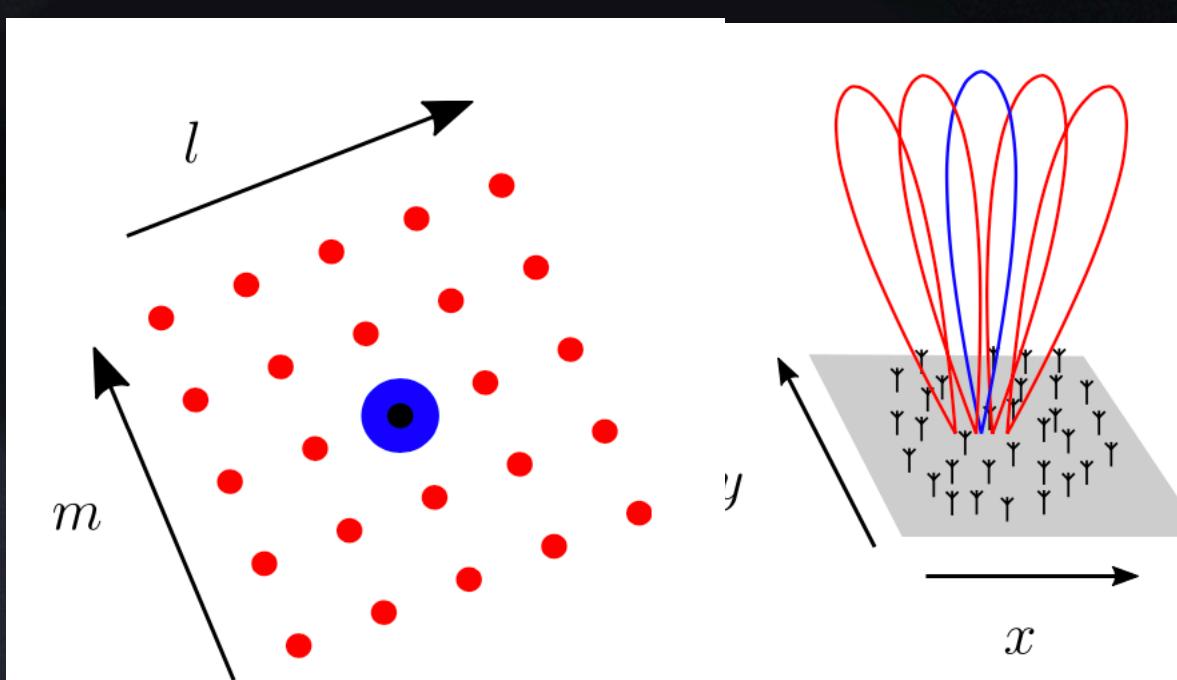


Figure 2. Sketch of holographic measurements for phased arrays. All beams can be formed at once in post-processing.

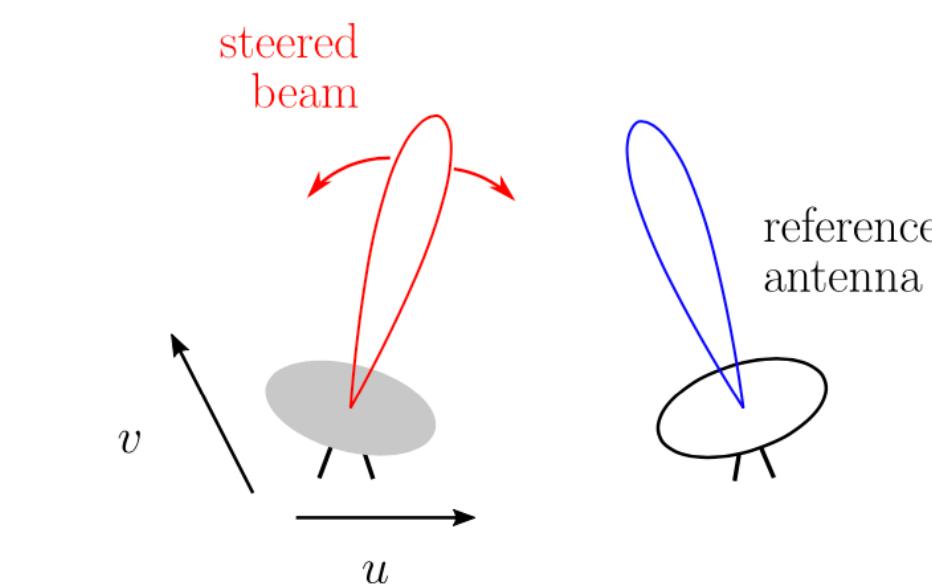
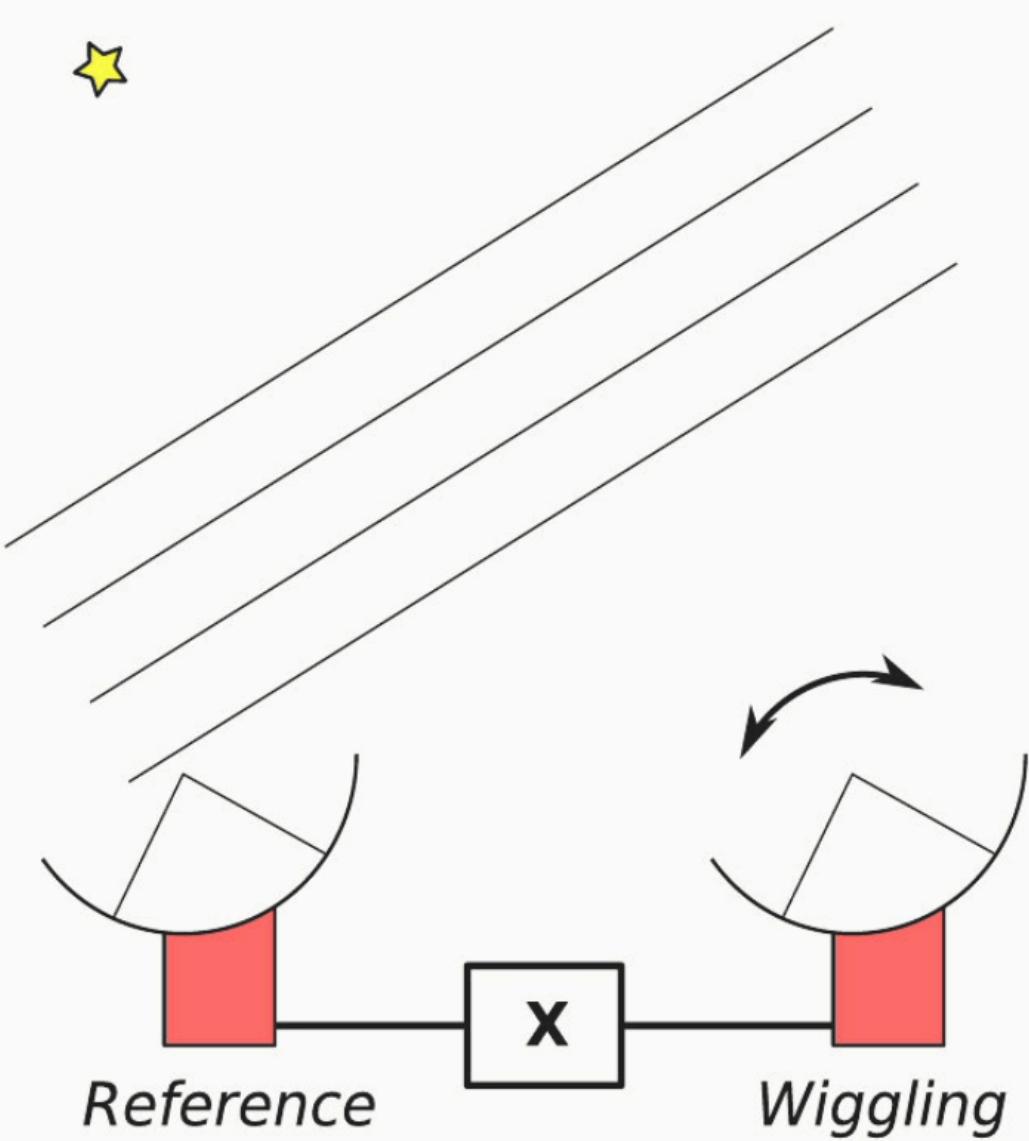


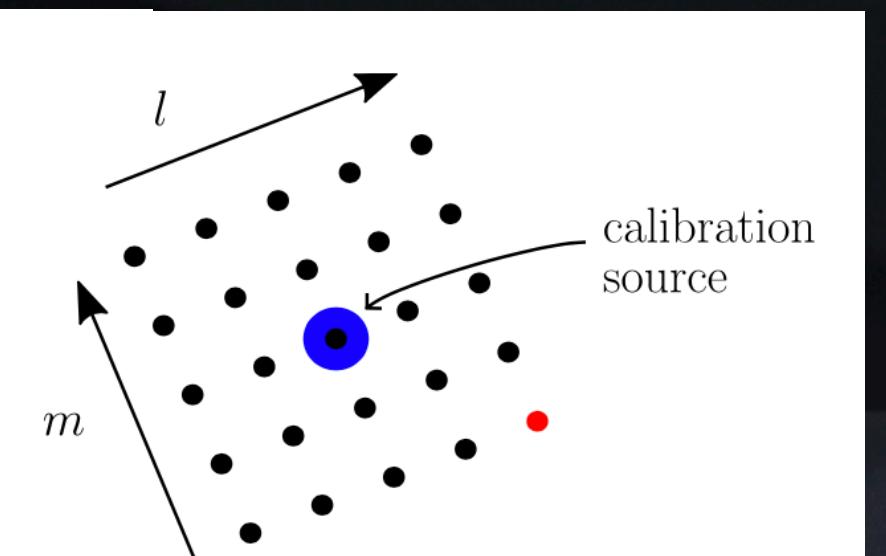
Figure 1. Sketch of the holographic measurement procedure for dish antennas. A separate reference antenna stays focused on the calibration source, while the AUT is adjusted to various lm -coordinates. The coordinate axes are drawn such that we are looking “down” on both the antennas and the sky.

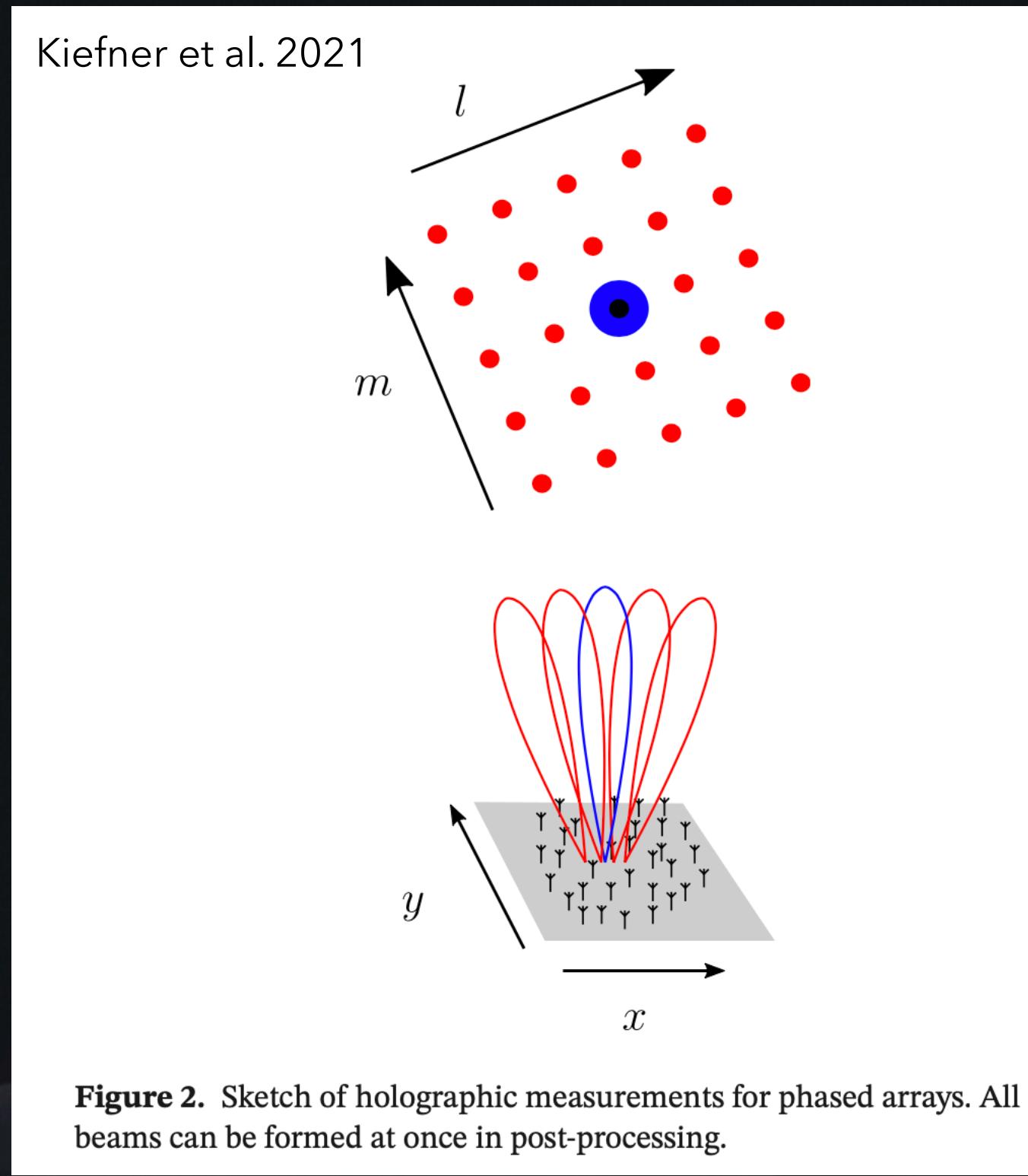
Kiefner et al. 2021



Brentjens & Bordenave (ASTRON & UW)

LOFAR HOLOG

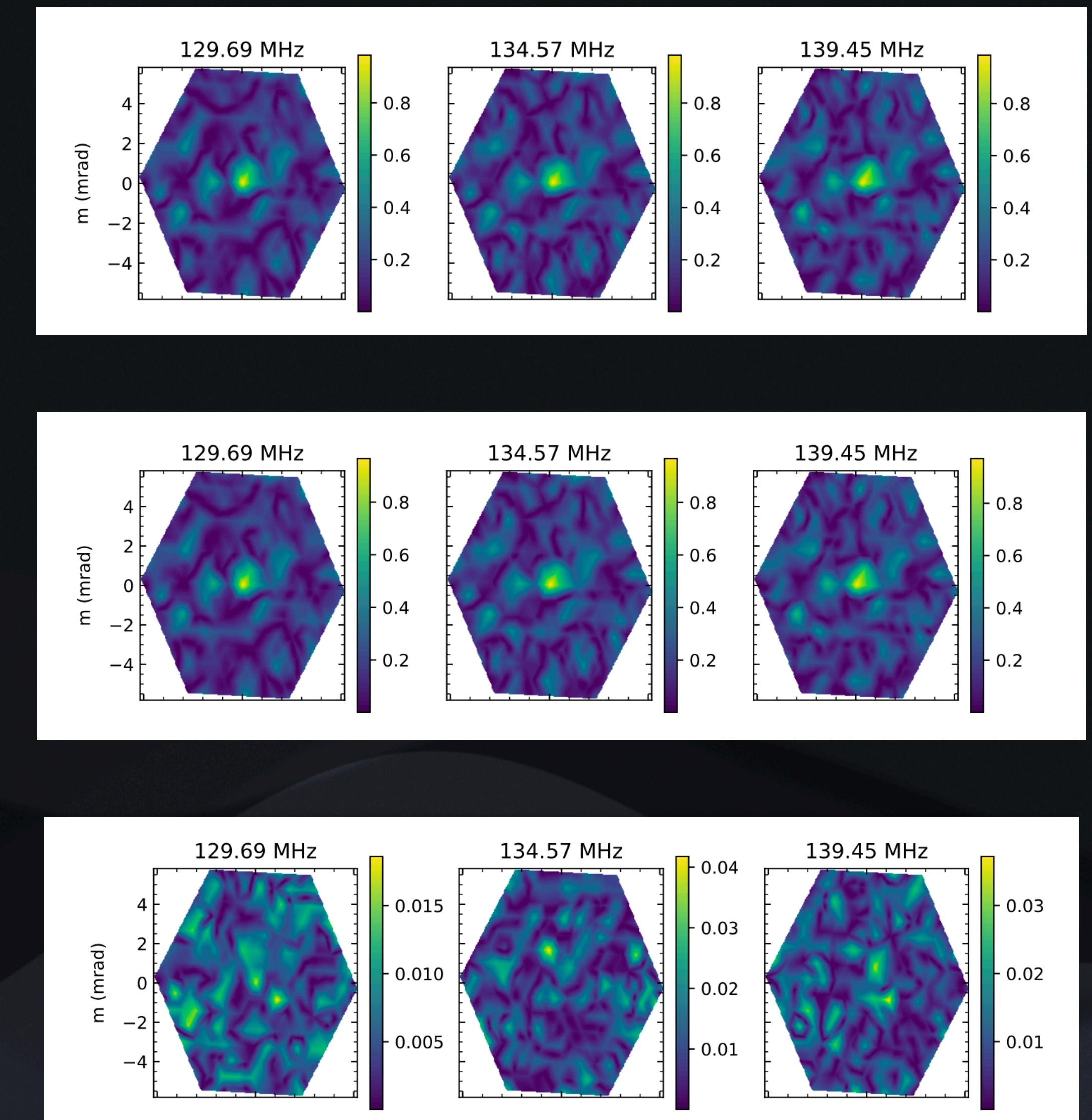




Observed beam

Model beam

Residual



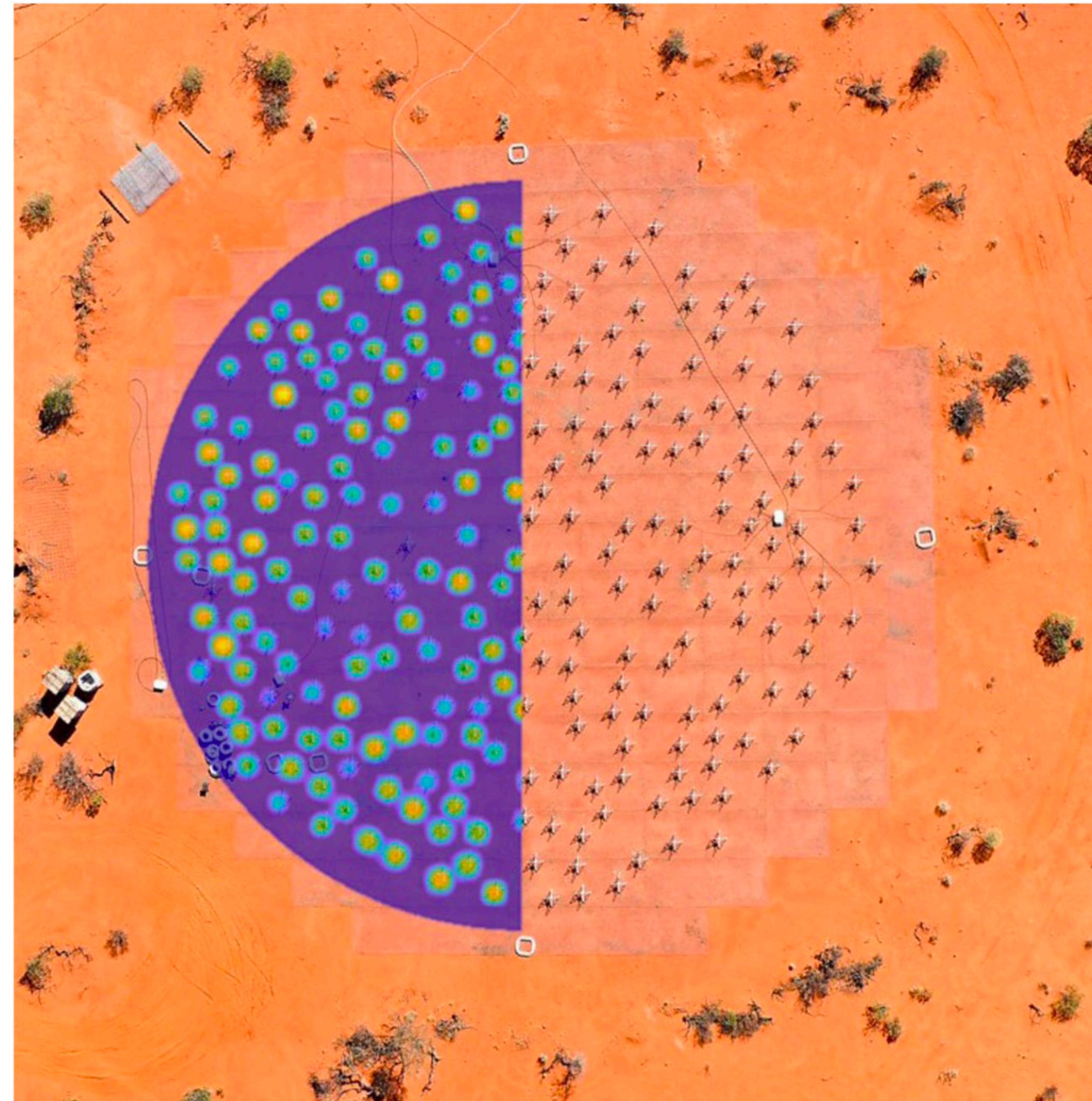


Figure 10. Overlay of an aperture image with an aerial drone photo of the EDA-2. The aperture image is partially transparent, and antenna dipoles can be seen in the locations of the peaks in the aperture as expected.

Background

Tied-array holography

- Aims here:
 - Make code user-friendly (eventually for LOFAR2.0 operators)
 - Make code modular (simplify updates in method / new test / modes)
 - Make code faster? (e.g. A. Gopinath's fast imaging to search for FRBs)

Update

Tied-array holography

- ***holog*** (P. Salas) (which relies lightly on ***taholog*** by M. Brentjens)
 1. Converted from Python 2 to 3, and containerized to singularity
 - Tested to reproduce results w/ Salas et al. original observations (L697741)
 2. Refactored code for **modular** runs
 - **Profiled** the code per step to evaluate bottlenecks
 - For a 60 second observation:
 - Single-threaded mode: ~**12 hours** ; Multi-threaded w/ 16 cores ~**2 hours 20 min**

Update

Tied-array holography

- After refactoring

```
method:serial, cuda, to_disk:True  
0:00:11.279658
```

```
method:serial, cuda, to_disk:False  
0:00:07.362257
```

```
method:serial, numpy, to_disk:True  
0:00:16.752776
```

```
method:serial, numpy, to_disk:False  
0:00:17.500211
```

```
method:serial, pyfftw, to_disk:True  
0:00:36.808636
```

```
method:serial, pyfftw, to_disk:False  
0:00:30.844438
```

```
method:multiprocessing, cuda, to_disk:True  
0:01:54.367784
```

```
method:multiprocessing, cuda, to_disk:False  
0:01:54.501390
```

```
method:multiprocessing, numpy, to_disk:True  
0:01:52.948554
```

```
method:multiprocessing, numpy, to_disk:False  
0:01:52.819608
```

```
method:multiprocessing, pyfftw, to_disk:True  
0:01:48.968451
```

```
method:multiprocessing, pyfftw, to_disk:False  
0:01:49.682640
```

Update

Tied-array holography

```
# FFT
```

```
time python main.py --parallel --no-use_gpu --no-use_pyfftw --to_disk --verbose  
--target_id 'L2036952' --reference_ids 'L2036944' --input_dir '/path_to/taholog/' --output_dir '/path_to/output/'  
--steps 'to_freq' >> ../taholog_L2036952_parallel_no-use_gpu_to_disk.out 2>&1
```

```
real    24m1.301s  
user    60m5.783s  
sys     176m4.354s
```

```
# Correlation
```

```
Singularity> time python main.py --parallel --no-use_numba --no-use_gpu --no-use_pyfftw --to_disk --verbose  
--target_id 'L2036952' --reference_ids 'L2036944' --input_dir '/path_to/taholog/' --output_dir '/path_to/output/'  
--steps 'xcorr' 'plot_beam' >> ../taholog_L2036952_xcorr_till_end.out 2>&1
```

```
real    36m36.055s  
user    294m41.879s  
sys     135m31.484s
```

```
# MAKE PLOT
```

```
Singularity> time python main.py --parallel --no-use_numba --no-use_gpu --no-use_pyfftw --to_disk --verbose  
--target_id 'L2036952' --reference_ids 'L2036944' --input_dir '/path_to/taholog/' --output_dir '/path_to/output/'  
--steps 'gencal' 'applycal' 'clip' 'average_t' 'to_uvhol' 'average_uvhol' 'solve_uvhol' 'order_sols' 'plot_report' >> ../taholog_L2036952_till_end.out 2>&1
```

```
real    2m17.000s  
user    8m30.428s  
sys     1m24.203s
```

Further development

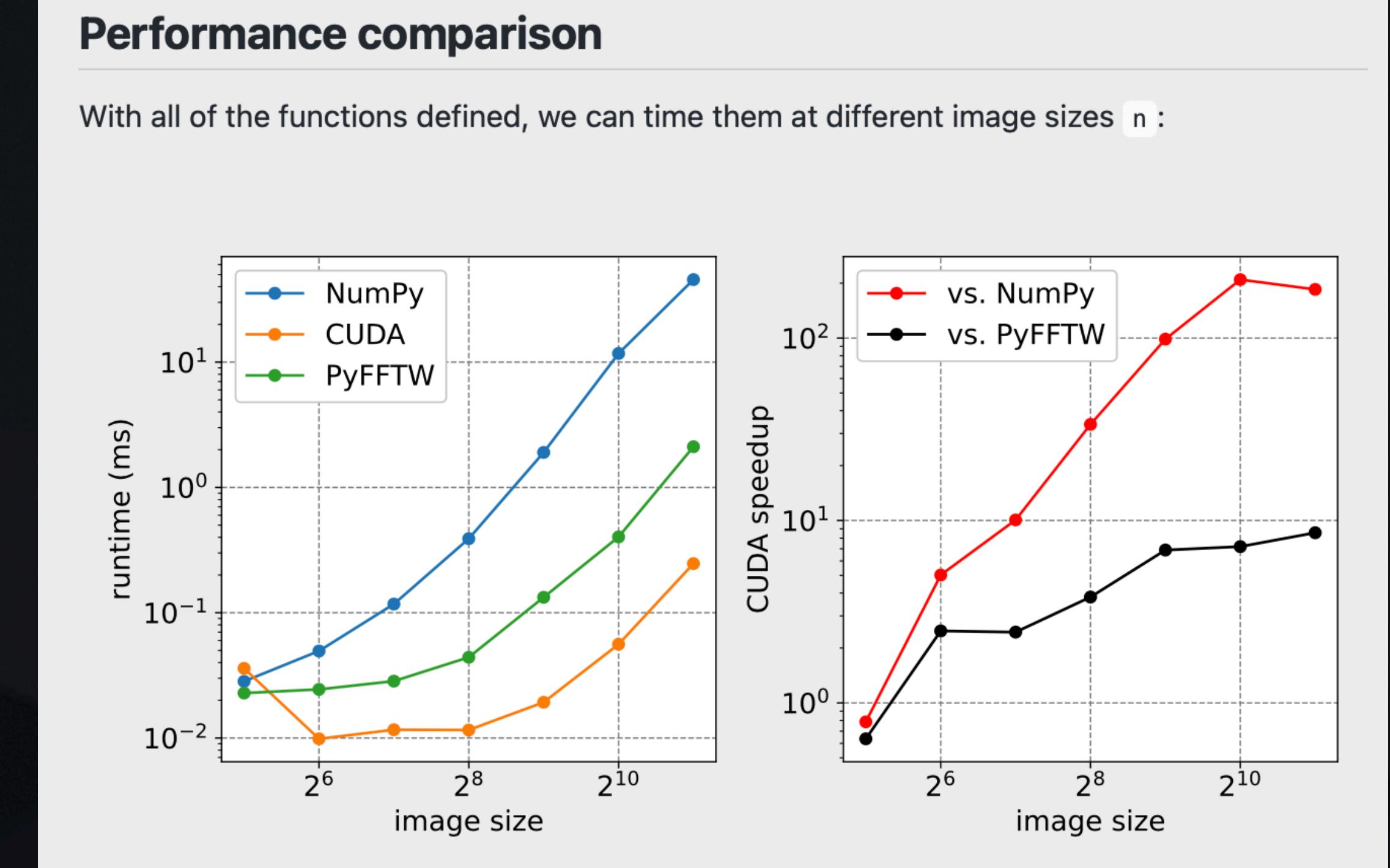
Updated observation strategy with TMSS: *updated data structure*

- Gathered recent observation through TMSS (w/ C. Bassa) to reproduce original observation
- Now stored into a single HDF5 file as different beams. The test station was recorded by itself.
- Long term, best to simply schedule a single observations that would include both the reference and test stations into a single file (and hence, need only to schedule a single observation, instead of two simultaneous observations).

Current and next steps

LOFAR2.0

- Finish fixes to new format
- Enable FFTs and correlation onto GPU
- Documentation & tests



johnaparker.com/blog/fft_2d_performance