A Cascading Beam Pattern Simulator for Multi-level Aperture Arrays

LIU, Lei

Shanghai Astronomical Observatory, Chinese Academy of Sciences

Oct. 23, 2024

IVTW2024, Haystack

Outline

- Introduction to Omni*UV*
- Cascading beam pattern simulation scheme
- Simulation examples

Interferometry simulation

中国科学院上海天文台
Shanghai Astronomical Observatory, CAS

- **Radio Interferometry Measurement Equation** (Smirnov 2011):

$$V_{pq} = Be^{-2\pi i(u_{pq}l+v_{pq}m+w_{pq}(n-1))}, \quad \boldsymbol{u}_{pq} = \boldsymbol{u}_p - \boldsymbol{u}_q,$$

✓ **OSKAR** (Dulwich et al. 2009)

- Interferometer and beamforming simulation dedicated to simulations of SKA aperture array
- Hierarchical structure: antenna field pattern within a station, station beam

✓ **pyuvsim** (Lamnan et al. 2019)

- Low frequency array
- Emphasis on accuracy and design clarity over efficiency and speed
- Neutral hydrogen study

✓ **RASCIL**

- SKA Science Data Processing (SDP)

✓ **CASA**

- ALMA, JVLA

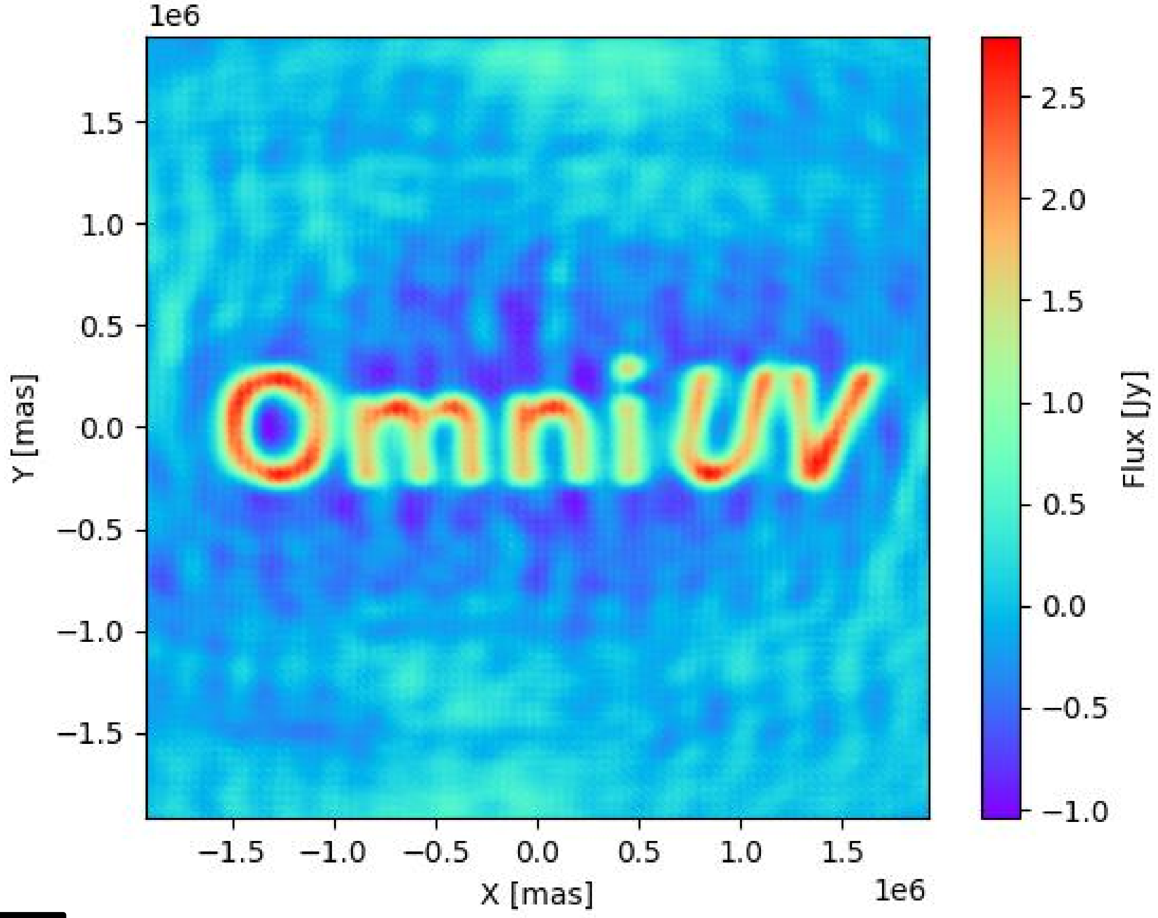
✓ **MeqTrees, MeqSilhouette, eth-imaging**

- ☐ **Dedicated to one project**
- ☐ **Full featured**
- ☐ **Speed**
- ☐ **Black box, difficult to extend**

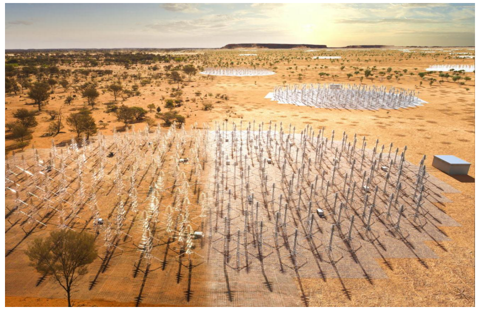
Omni*UV*: Omnipotent *UV*

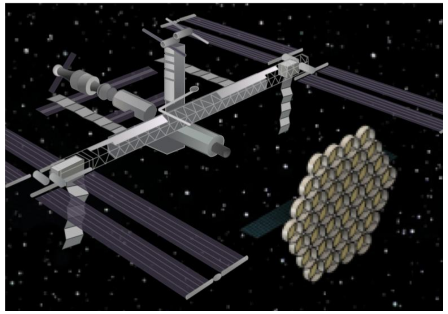
中国科学院上海天文台
Shanghai Astronomical Observatory, CAS

- ✓ Station trajectory calculation
  - from ground to space
- ✓ *UVW* calculation
  - Telescope availability
  - Elevation, separation
- ✓ Visibility simulation
  - System noise, antenna gain, ...
  - FFT, DFT
- ✓ Image/beam reconstruction
  - Small/wide field



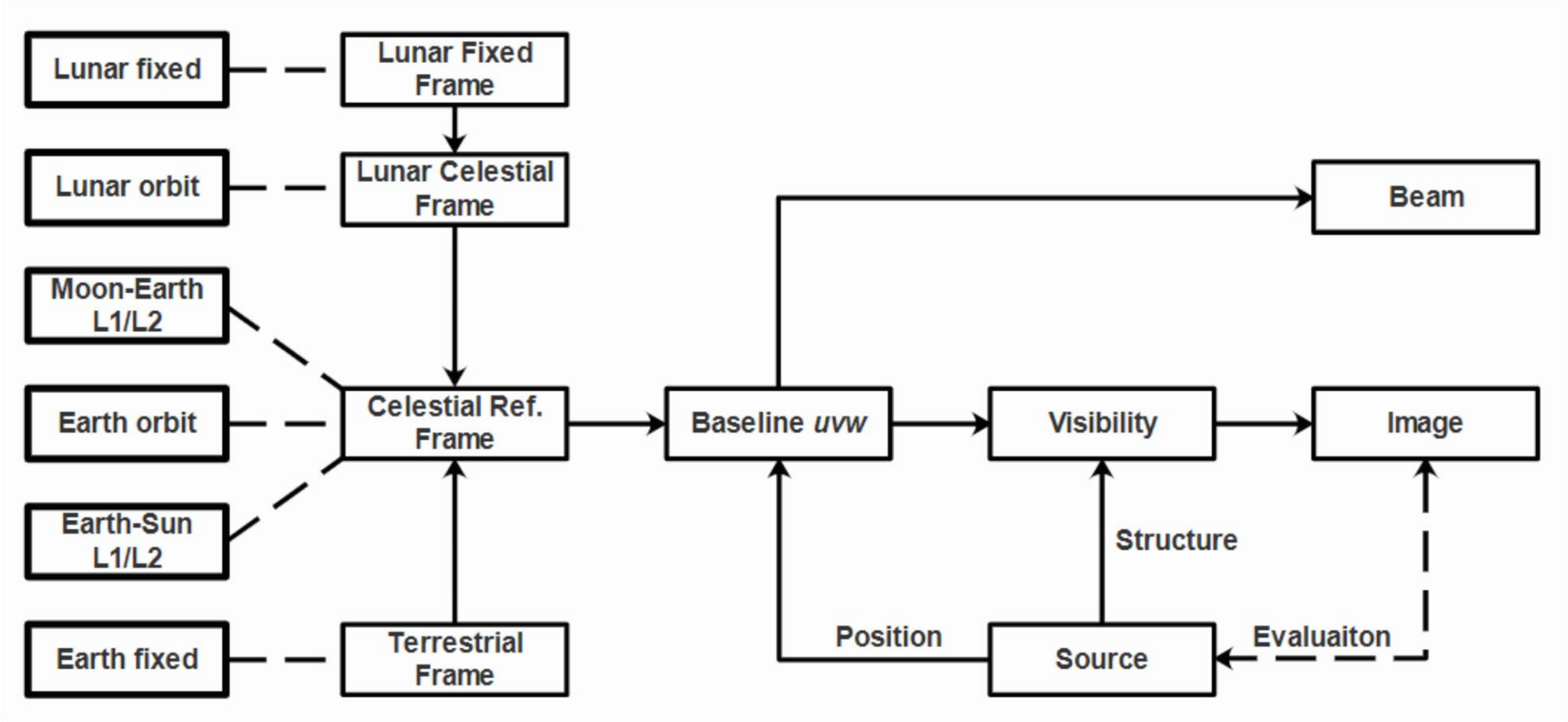
- ☐ **From aperture array to space VLBI**
- ☐ **Trajectory, uvw, vis., beam/image**
- ☐ **Python based, fully parallel, GPU**
- ☐ **Easy to extend (stations, features)**





Architecture & Implementation

中国科学院上海天文台
Shanghai Astronomical Observatory, CAS



Visibility calculation

$$V_k = \sum_i S_i e^{-j2\pi(u_k l_i + v_k m_i + w_k(n_i - 1))},$$

$$\sigma_{ij} = \frac{1}{\eta}\sqrt{\frac{\mathrm{SEFD}_i \times \mathrm{SEFD}_j}{2\,B\,T}},$$

$$V_{ij} = G_i\, G_j\, e^{-j(\phi_i - \phi_j)}(V_{0,ij} + \epsilon_{ij}),$$

Image reconstruction

$$S_i = \sum_k V_k\, e^{j2\pi(u_k l_i + v_k m_i + w_k(n_i - 1))}/N,$$

Wide field radio imaging

- ✓ w-stacking
- ✓ w-projection
- ✓ Discrete Fourier Transform (DFT)

Hardware acceleration

$$V_k = \sum_i S_i e^{-j2\pi(u_k l_i + v_k m_i + w_k(n_i - 1))},$$

```
__global__
void oskar_evaluate_jones_K_cudak_d(double2* restrict jones,
        const int num_sources, const double* restrict l,
        const double* restrict m, const double* restrict n,
        const int num_stations, const double* restrict u,
        const double* restrict v, const double* restrict w,
        const double wavenumber, const double* restrict source_filter,
        const double source_filter_min, const double source_filter_max)
{
    const int s = blockDim.x * blockIdx.x + threadIdx.x; /* Source index. */
    const int a = blockDim.y * blockIdx.y + threadIdx.y; /* Station index. */

    /* Cache source and station data from global memory. */
    __shared__ double l_[BLK_SOURCE], m_[BLK_SOURCE], n_[BLK_SOURCE];
    __shared__ double f_[BLK_SOURCE];
    __shared__ double u_[BLK_STATION], v_[BLK_STATION], w_[BLK_STATION];
    if (s < num_sources && threadIdx.y == 0)
    {
        l_[threadIdx.x] = l[s];
        m_[threadIdx.x] = m[s];
        n_[threadIdx.x] = n[s] - 1.0;
        f_[threadIdx.x] = source_filter[s];
    }
    if (a < num_stations && threadIdx.x == 0)
    {
        u_[threadIdx.y] = wavenumber * u[a];
        v_[threadIdx.y] = wavenumber * v[a];
        w_[threadIdx.y] = wavenumber * w[a];
    }
    __syncthreads();

    /* Compute the geometric phase of the source direction. */
    double2 weight = make_double2(0.0, 0.0);
    if (f_[threadIdx.x] > source_filter_min &&
            f_[threadIdx.x] <= source_filter_max)
    {
        double phase;
        phase =  u_[threadIdx.y] * l_[threadIdx.x];
        phase += v_[threadIdx.y] * m_[threadIdx.x];
        phase += w_[threadIdx.y] * n_[threadIdx.x];
        sincos(phase, &weight.y, &weight.x);
    }
```

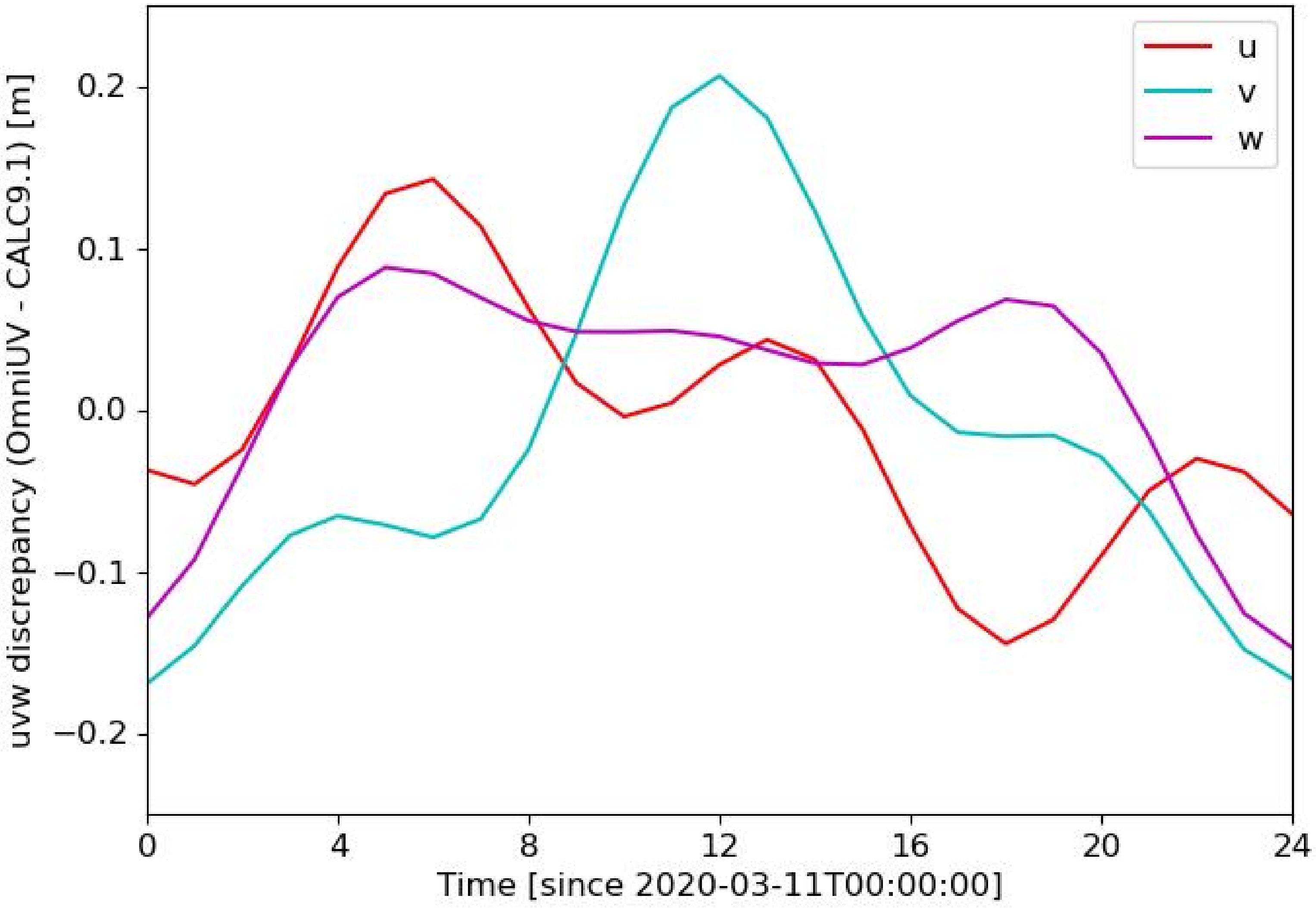
OSKAR (CUDA Kernel)

```
# t, uvw
        bl['uvw_m']         =   uvw_bl
# t
        bl['t']             =   self.ts[idt_bl]
# t, freq, uvw
        bl['uvw_wav']       =   bl['uvw_m'][:, np.newaxis, :] / \
            c_light.value * self.freqs[np.newaxis, :, np.newaxis]

        _lmn001             =   np.array([0, 0, 1])[np.newaxis, :]
# pixel, lmn
        lmn1                =   src.img.lmn - _lmn001
# pixel, t, freq
        lmn_uvw             =   np.einsum('hk,ijk -> hij', \
                                lmn1, bl['uvw_wav'])
        fringe              =   np.exp(2j * np.pi * lmn_uvw)
# fluxes: pixel
# vis: t, freq
        bl['vis']           =   np.einsum('h,hij -> ij', \
                                src.img.fluxes, fringe)
        bls.append(bl)
```

Omni*UV* (NumPy/CuPy/Torch, einsum)

Precision: *uvw* calculation



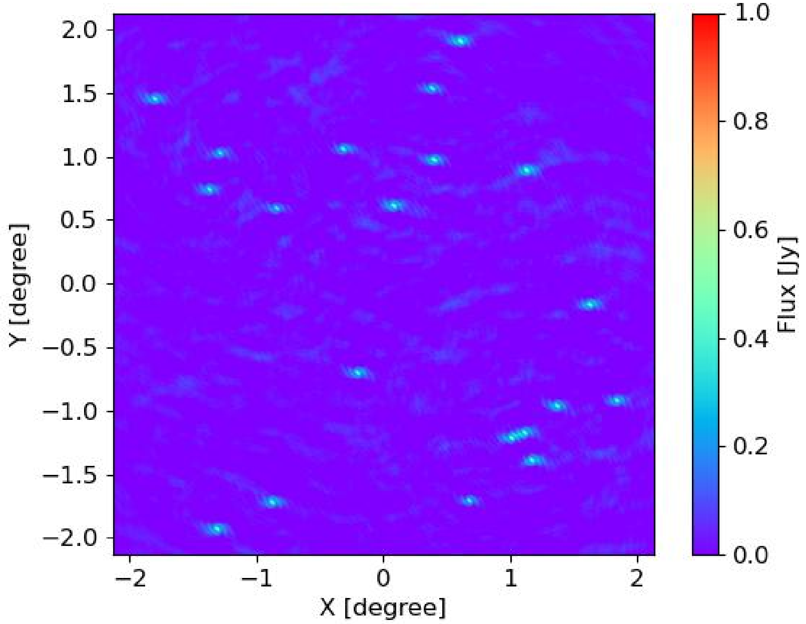
- Omni*UV* – CALC 9.1
- SKA-Mid location

EOPs:

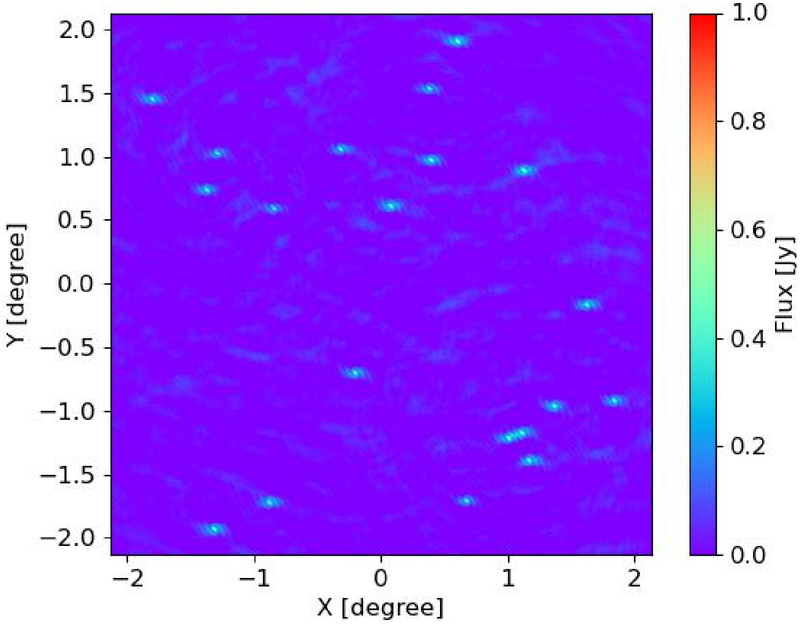
- TAI-UTC: 2 cm (typical value of 35 s)
- ut1-utc: 500m per 1s
- Polar motion: 10 m (30 m per arcsec)
- Precession & nutation: 10 km

Precision: image reconstruction

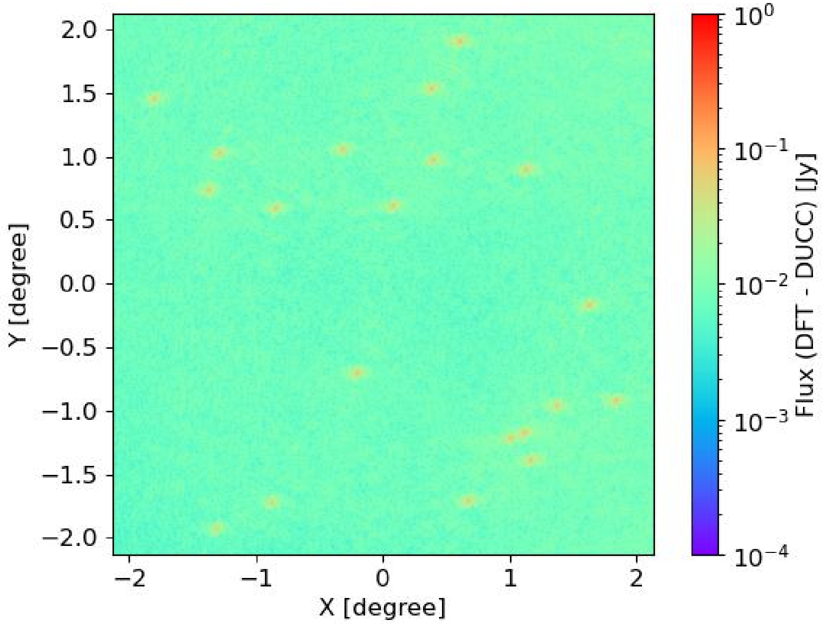
中国科学院上海天文台
Shanghai Astronomical Observatory, CAS



DFT



DUCC

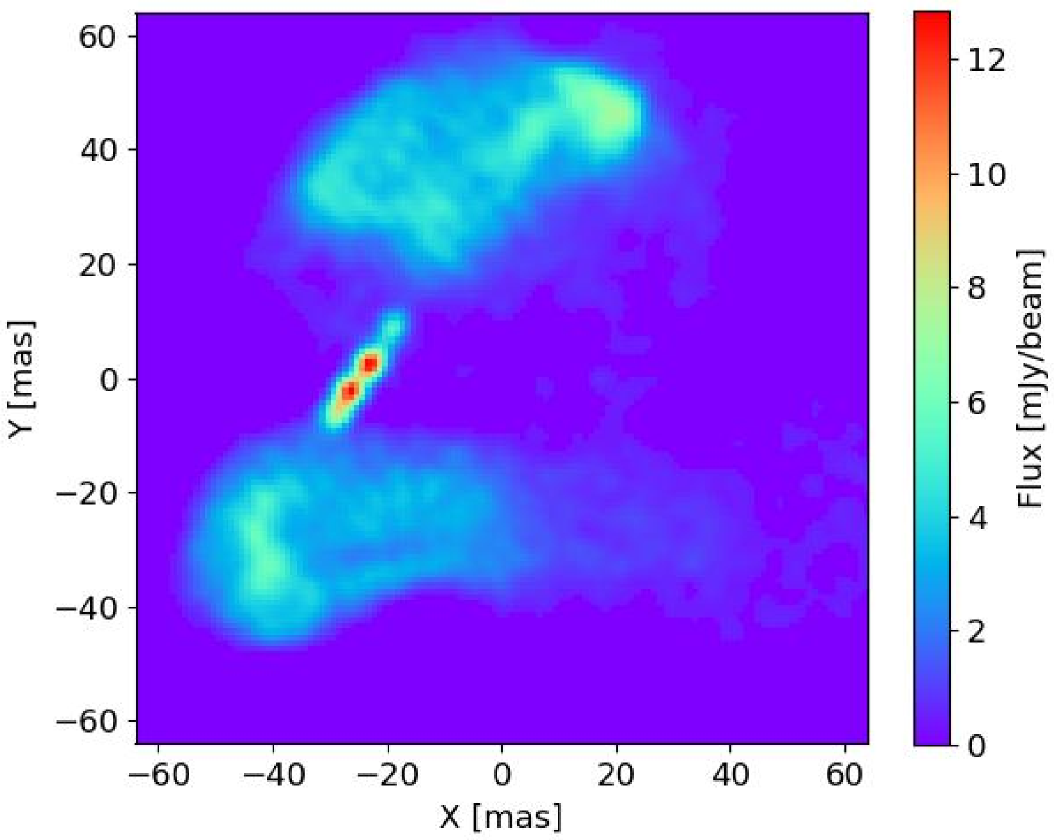


DFT - DUCC

DUCC

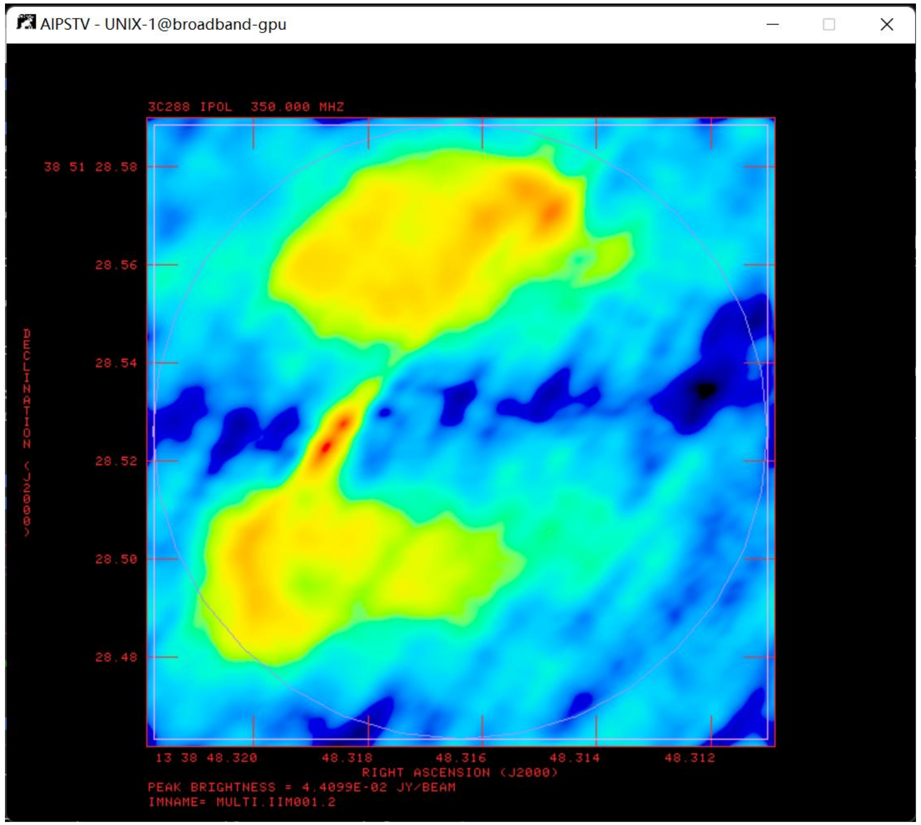
- Improved w-stacking (Arras et al. 2021)
- New gridding kernel based (Ye et al. 2020)
- https://gitlab.mpcdf.mpg.de/mtr/ducc

Export to FITS-IDI



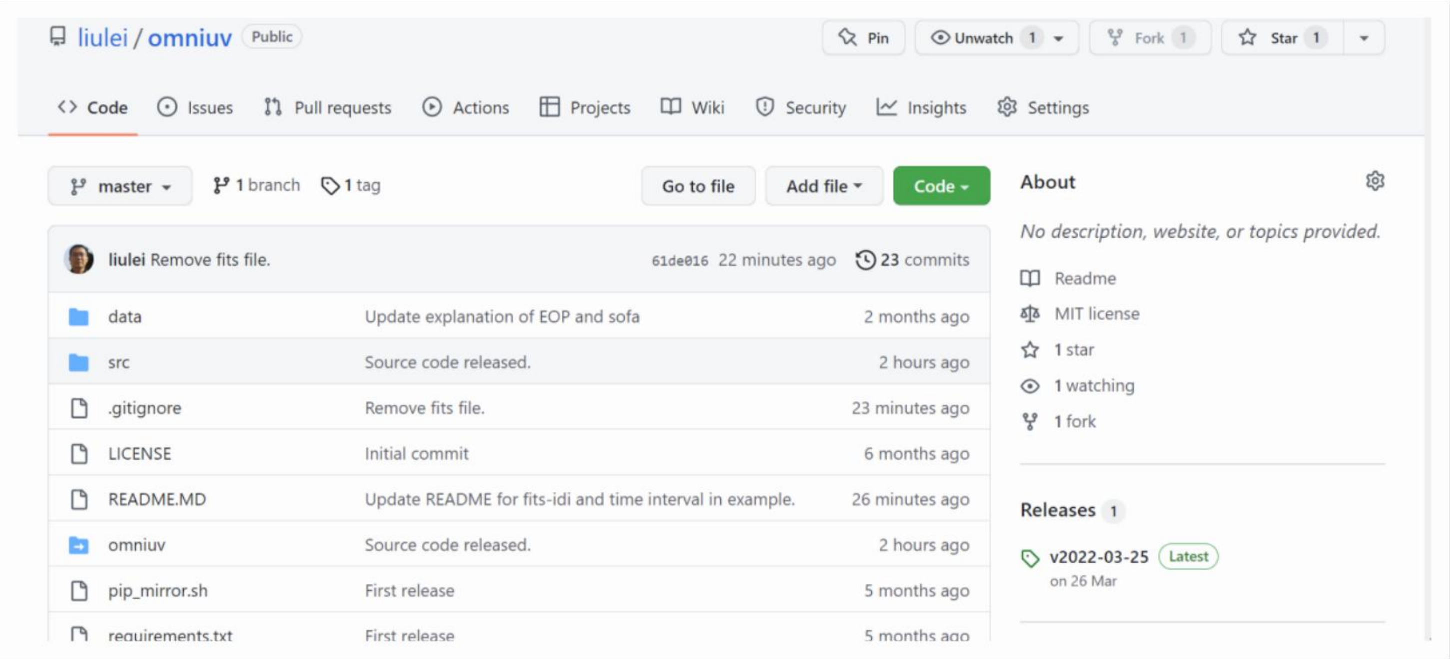
Source

Omni*UV*



AIPS

Omni*UV*: open source



- **Available on GitHub**
- **Codes, documents, examples**
- **Instrument effect**
- **FIT-IDI output**
- **Aperture array beam pattern**

- https://github.com/liulei/omniuv

The Astronomical Journal, 164:67 (9pp), 2022 August
https://doi.org/10.3847/1538-3881/ac77f0

CrossMark

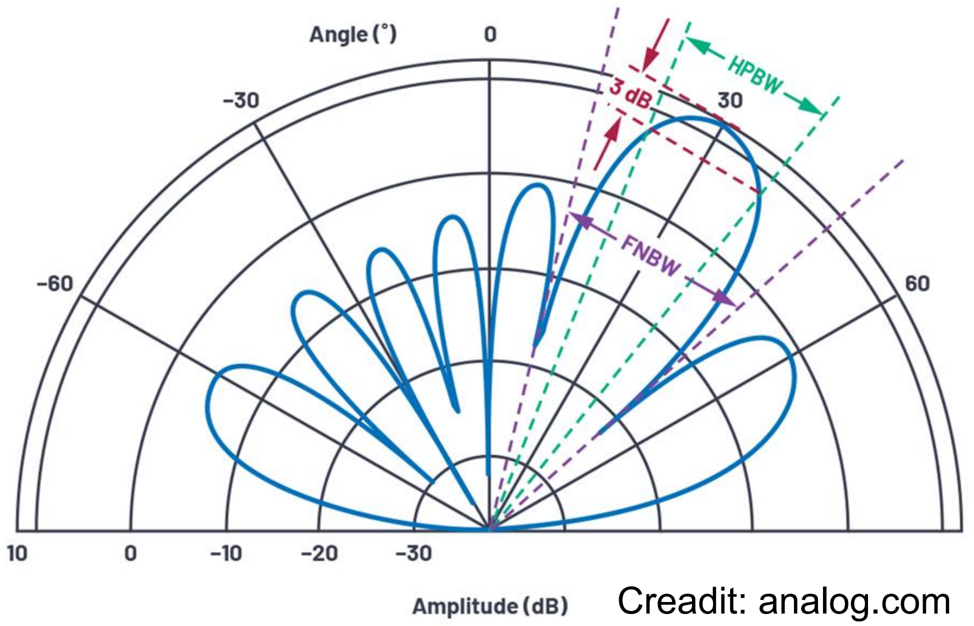
OmniUV: A Multipurpose Simulation Toolkit for VLBI Observation

Lei Liu[1,2,3], Weimin Zheng[1,2,3], Jian Fu[4], and Zhijun Xu[1]
[1] Shanghai Astronomical Observatory, Chinese Academy of Sciences, Shanghai 200030, People's Republic of China; liulei@shao.ac.cn
[2] National Basic Science Data Center, Beijing 100190, People's Republic of China
[3] Shanghai Key Laboratory of Space Navigation and Positioning Techniques, Shanghai 200030, People's Republic of China
[4] Key Laboratory for Research in Galaxies and Cosmology, Shanghai Astronomical Observatory, Chinese Academy of Sciences, 80 Nandan Rd., Shanghai, 200030, People's Republic of China
Received 2021 December 17; revised 2022 June 7; accepted 2022 June 9; published 2022 July 25

Cascading beam pattern simulation scheme

中国科学院上海天文台
Shanghai Astronomical Observatory, CAS

- Aperture array support
  - Wide field visibility simulation and imaging
  - Beam pattern calculation



Creadit: analog.com

- OSKAR (Dulwich et al. 2009)
  - Beamforming simulator
  - Hierarchical scheme
    - Station beam pattern

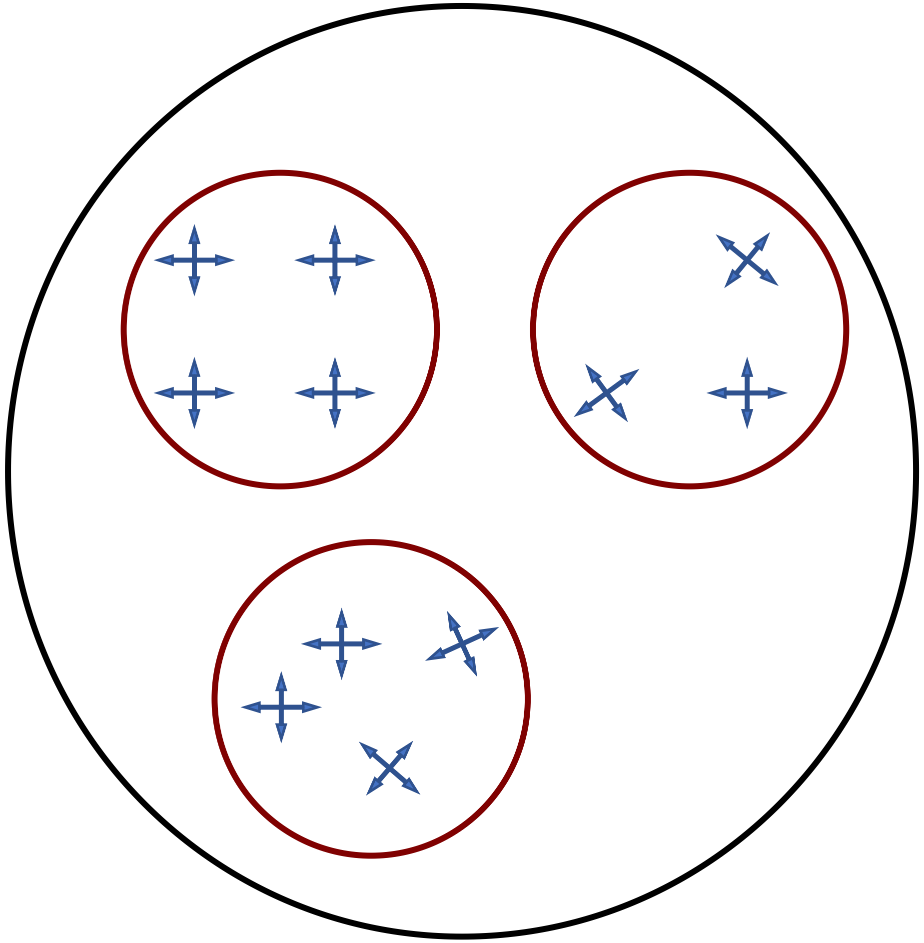
$$b_i = \sum_{j=1}^{N} W_{i,j} a_j$$

$$W = e^{i\psi}$$

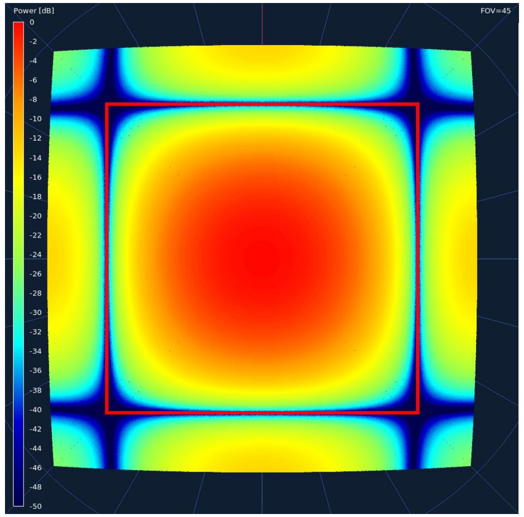
$$\psi = k \cos\theta \,(x \sin\phi + y \cos\phi)$$

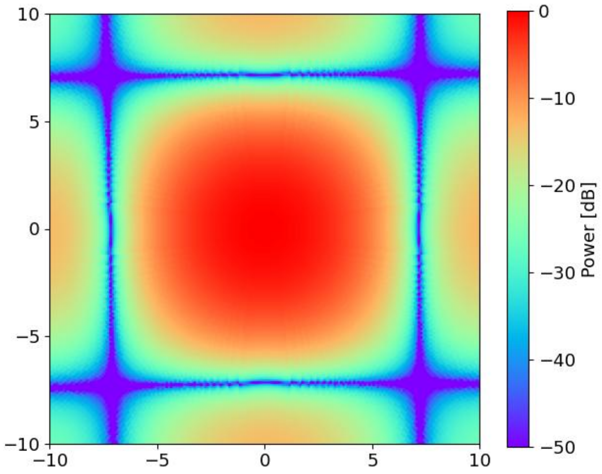
Cascading beam pattern simulation scheme

- Reference: Dulwich et al. (2009)
- High configurable
  - Uniform / directional
  - Beam forming for multi level arrays
    - Cascading structure
    - Identical class
    - Beam pattern is calculated based on the underlying level
- Support arbitrary levels of antenna/array configuration
- Integrated in the OminUV framework

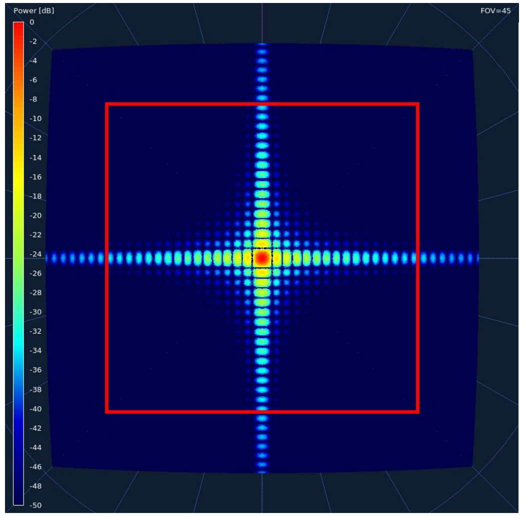


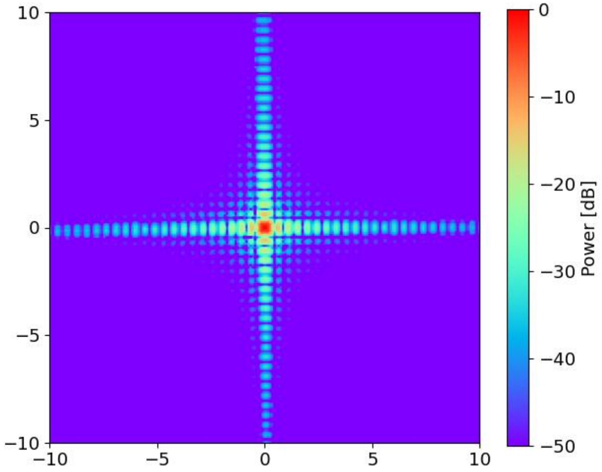
Beam pattern: comparison with OSKAR





Tile
- λ = 0.3m
- d = 0.5 λ
- 16 X 16 antennas



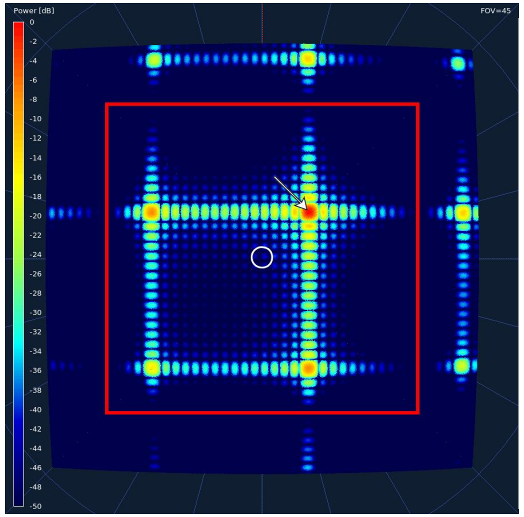


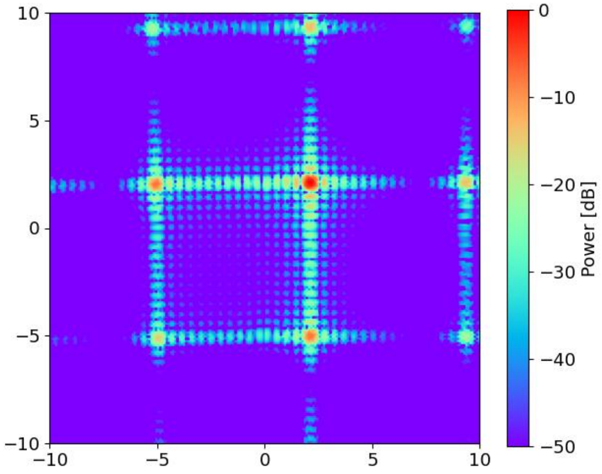
Station
- λ = 0.3m
- d = 0.5 λ X 16
- 16 X 16 packed tiles (65,536 antennas)

OSKAR **Omni*UV***

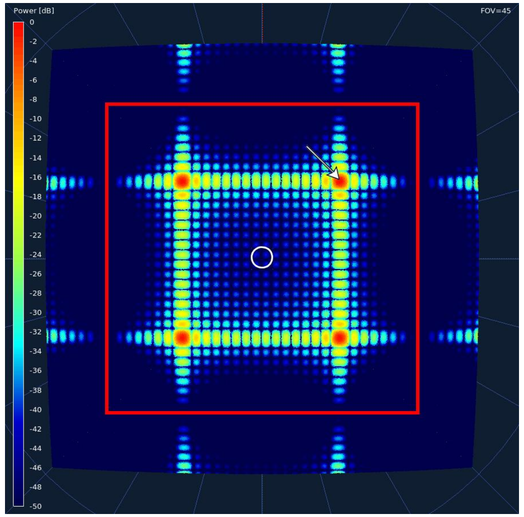
$\theta = 90°$

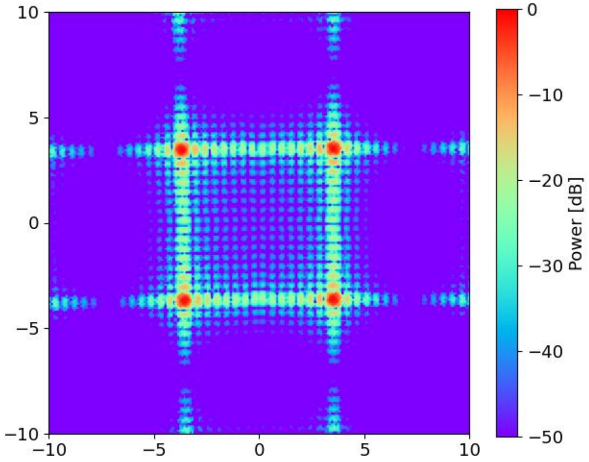
Beam pattern: comparison with OSKAR





$\varphi = 45°,\ \theta = 87°$





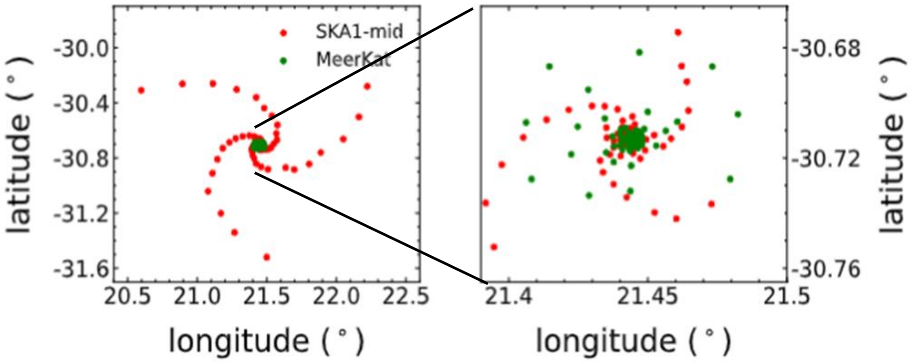
$\varphi = 45°,\ \theta = 85°$

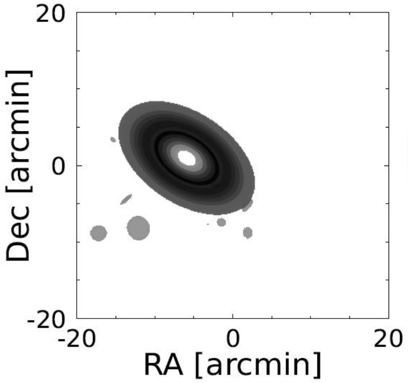
OSKAR **Omni*UV***

Simulation of HI gas with SKA-Mid

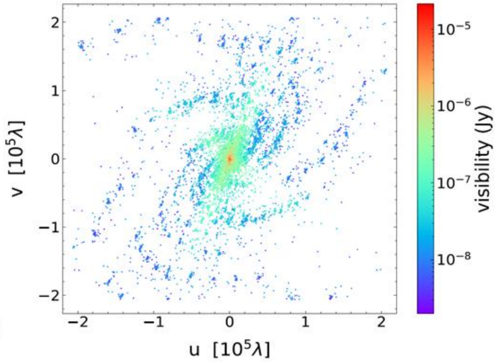
- Outputs of HI gas in galaxies from L-Galaxies and Illustris-TNG to construct mock HI flux.
- Simulate the observation of 21cm interferometer signals for SKA1-mid for HI gas, which helps for future observation of HI in nearby universe.
- Future work: FASTA, HI gas at high redshift, instrumental effects.

SKA-1 mid configurations

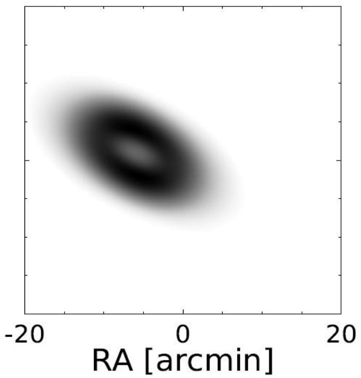




Input galaxy



uv distribution



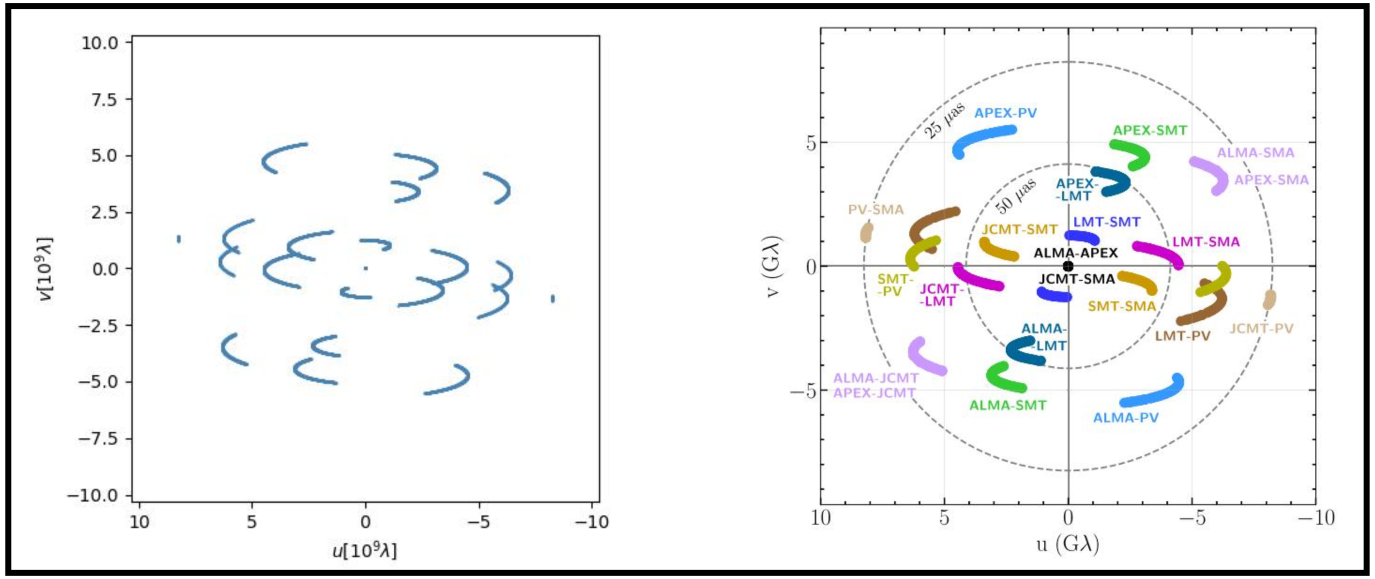
Output galaxy

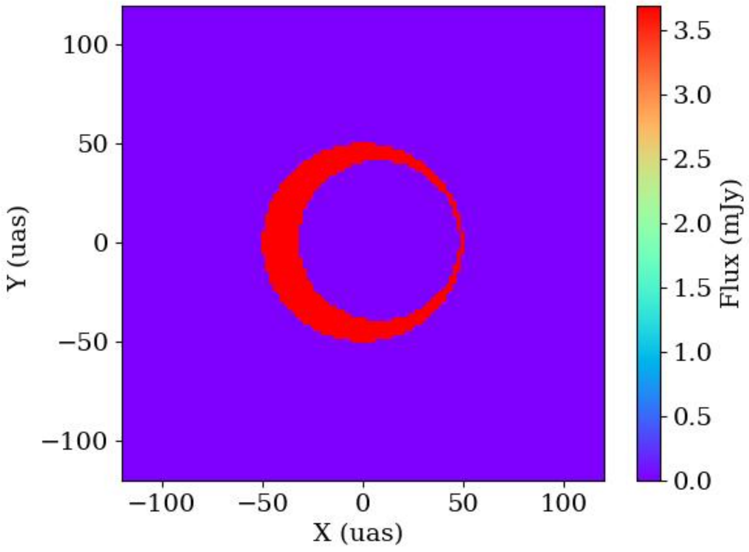
Simulation time	6 hours
Frequency	1.36 - 1.42 GHz
Angular res.	0.5 asec
Dec.	-30.7°

Credit: FU, Jian

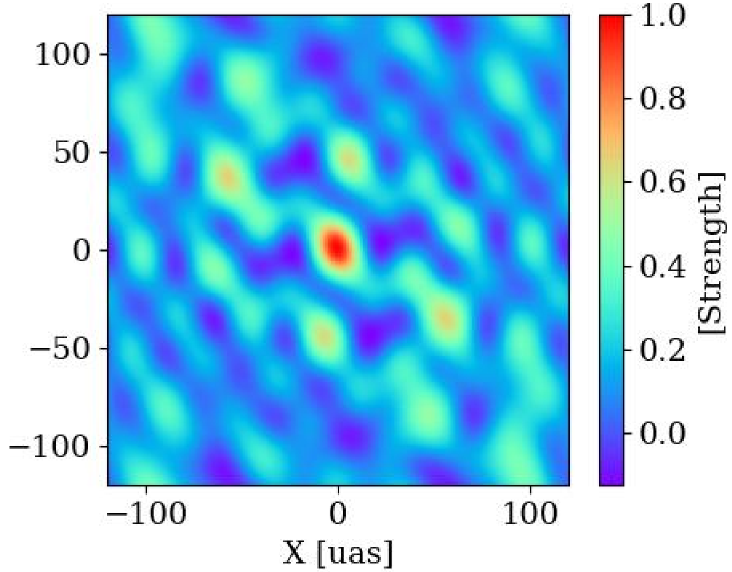
Simulation of BH crescent model

- Crescent model: eht-imaging
- Kamruddin & Dexter (2013)
- Frequency: 230 GHz
- Cellsize: 1.875 uas
- Visibility: OmniUV
- Image reconstruction: CASA

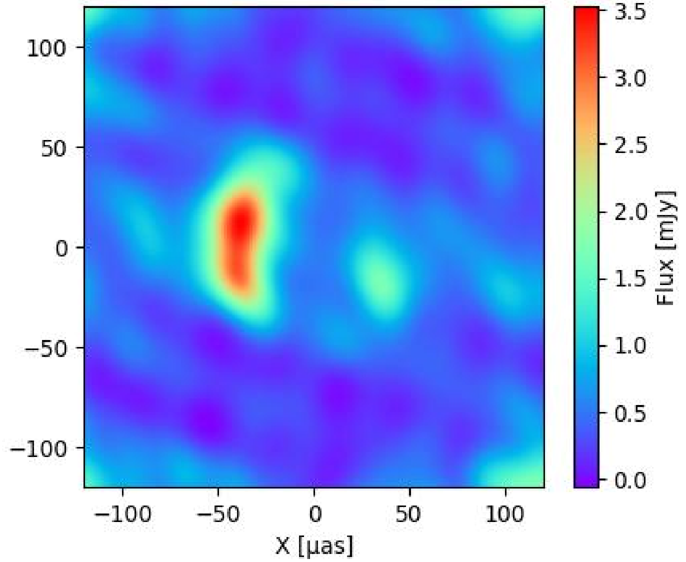




Input



Beam



Reconstructed

Credit:
Yu, Jia

Summary

中国科学院上海天文台
Shanghai Astronomical Observatory, CAS

OmniUV

- ✓ Multiple scales
- ✓ uvw, visibility, beam/image
- ✓ Easy to use/extend

Beam pattern simulator

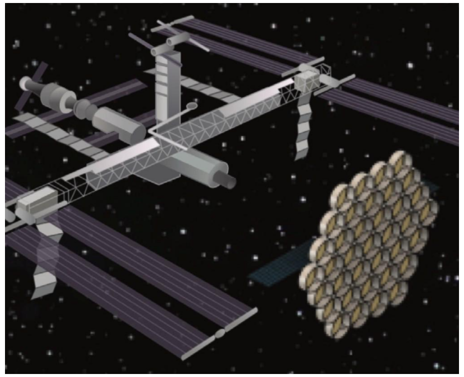
- ✓ Flexible antenna configuration
- ✓ Cascading structure
- ✓ OmniUV framework

https://github.com/liulei/omniuv

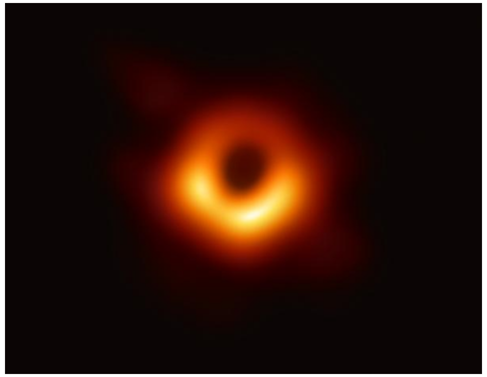
SKA

MeerKat

CVN



THEZA



Black hole

FAST Array