UNIVERSITY OF CAMBRIDGE

Lecture 8
Imaging the Universe with Radio Interferometers

Lecturer: Dr Haoyang Ye (hy297)

UNIVERSITY OF CAMBRIDGE

In the next one hour

- Join the quiz/discussion
- Ready to code in Python to solve a tiny problem (use QR code to download jupyter notebook)
- Ask questions loud if you need clarification
- You can discuss with each other

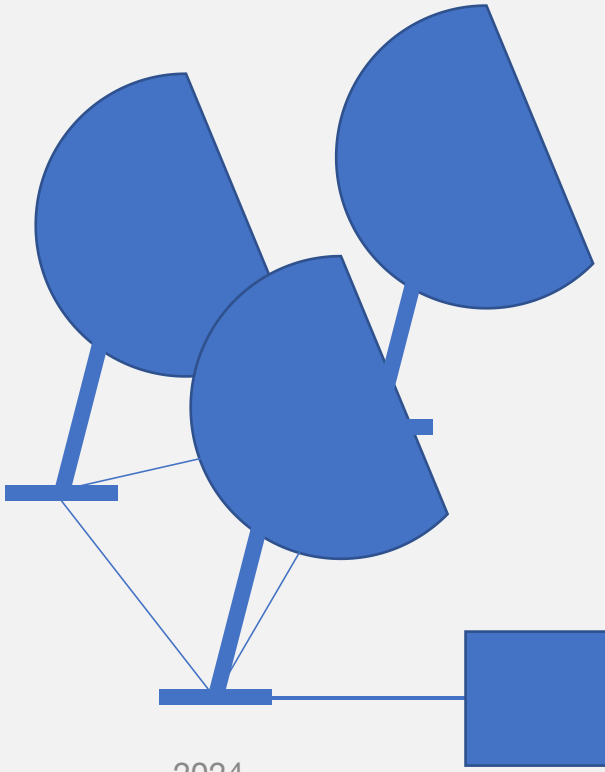
UNIVERSITY OF CAMBRIDGE

Overview

- Introduction to Big Data Radio Astronomy and Key Science Projects
- Instrument simulations and design tools
- Science Data Processing
    - o Lecture 7: Calibration of radio observations
    - o Lecture 8: Imaging techniques
    - o Lecture 9: Advanced imaging techniques
    - o Lecture 10: Time-domain radio astronomy
- Computing infrastructure
- Advanced ML and Bayesian methods for data analysis and science extraction

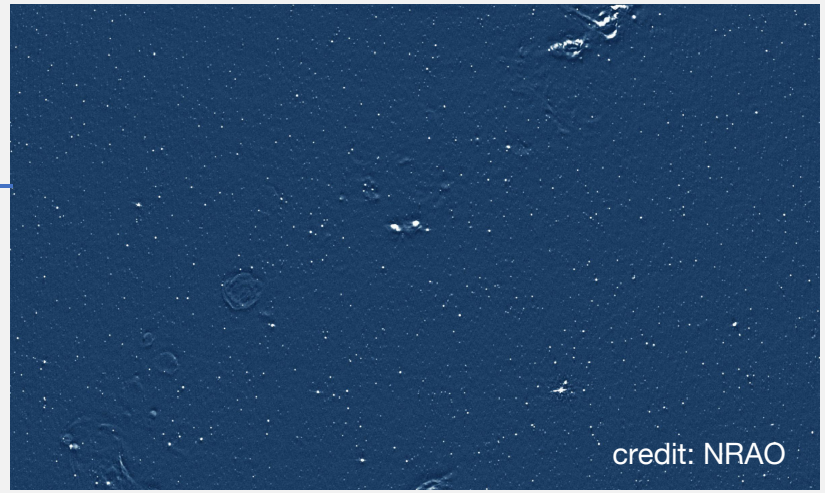
UNIVERSITY OF CAMBRIDGE

Today you will learn:



Visibility Data

Imaging



An image!!!

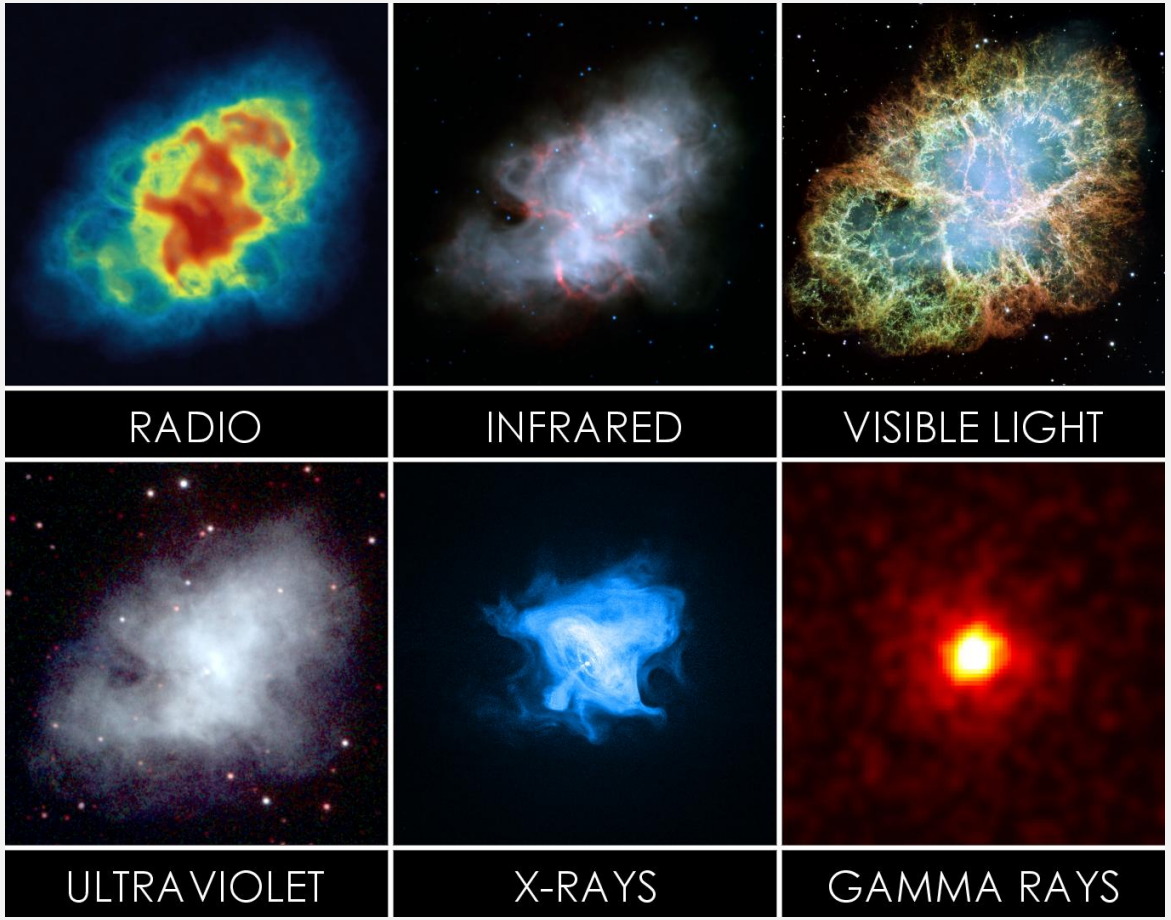
UNIVERSITY OF CAMBRIDGE

We love images!!!

Credit: NRAO

UNIVERSITY OF CAMBRIDGE

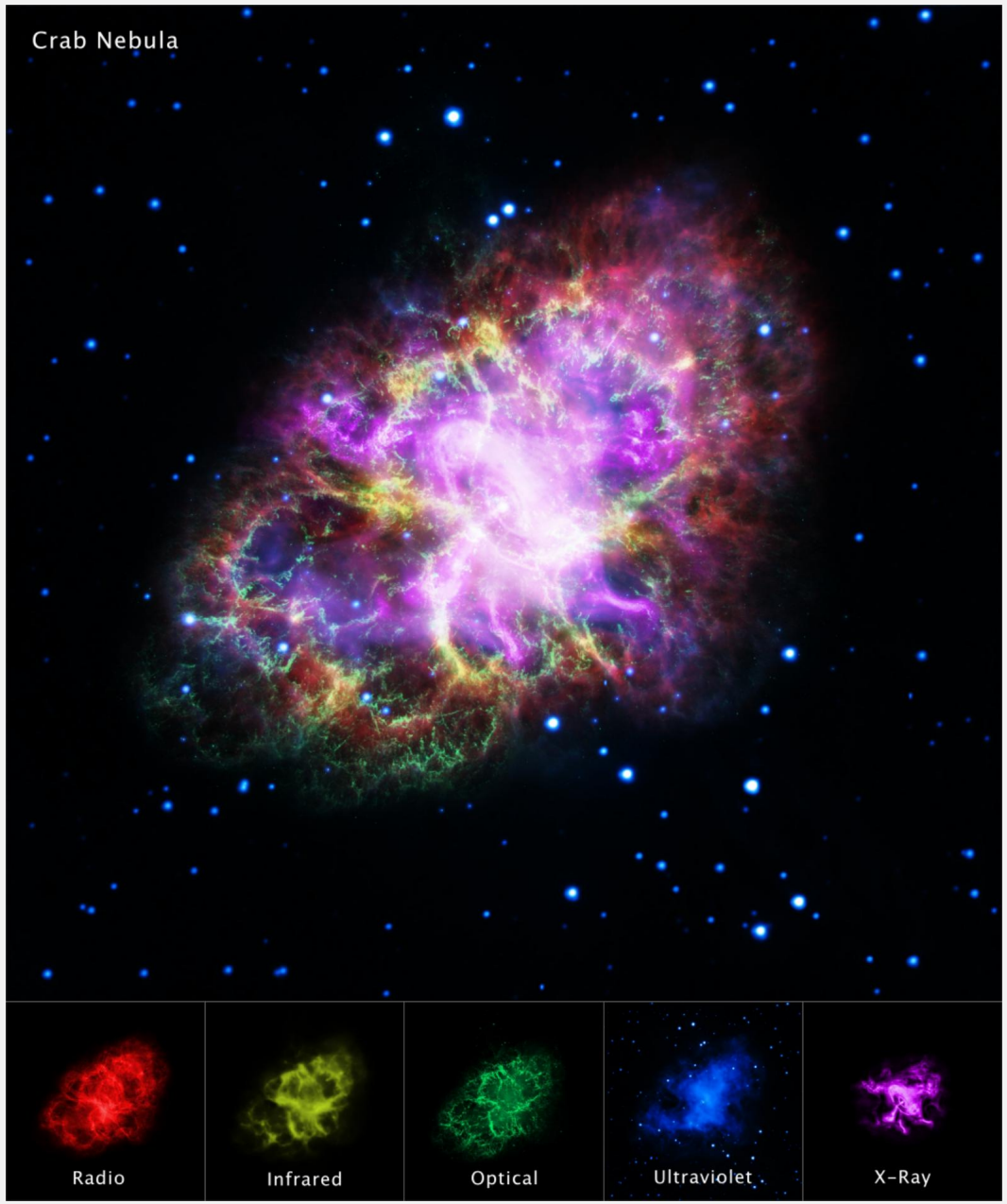
Crab Nebula imaged in Multiple Wavelengths



Credit: NRAO

UNIVERSITY OF CAMBRIDGE

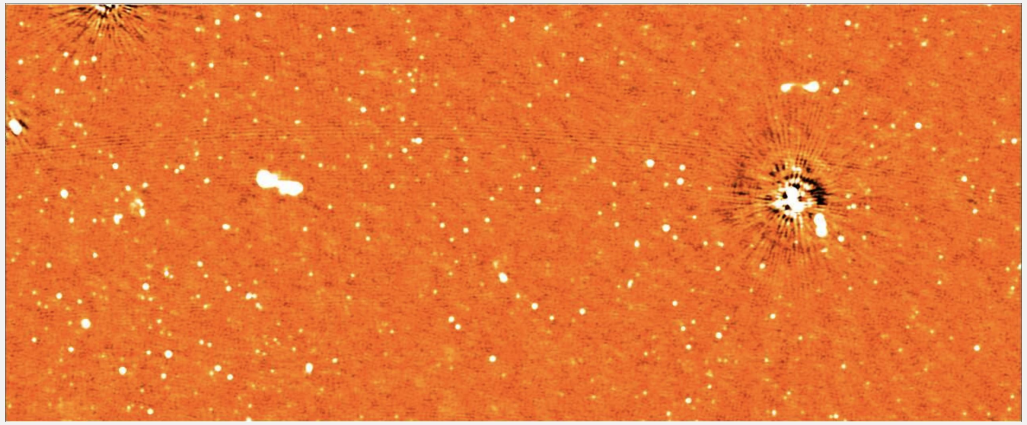
Crab Nebula imaged in Multiple Wavelengths

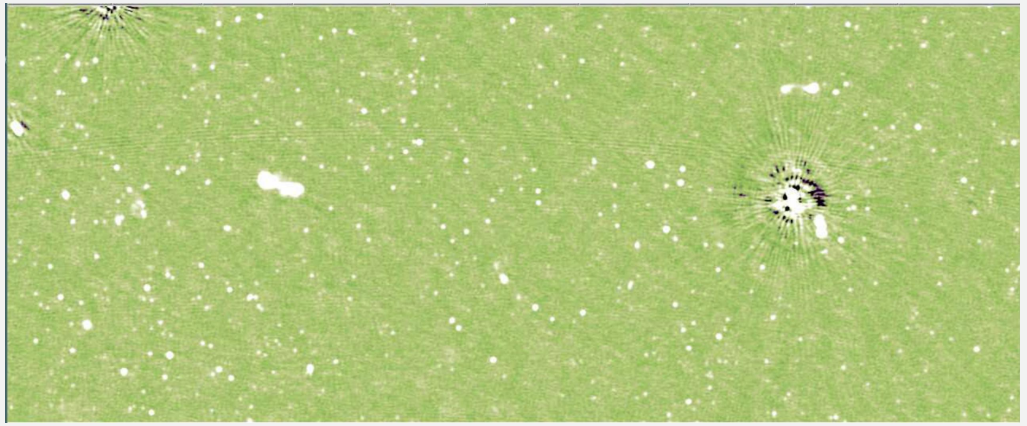


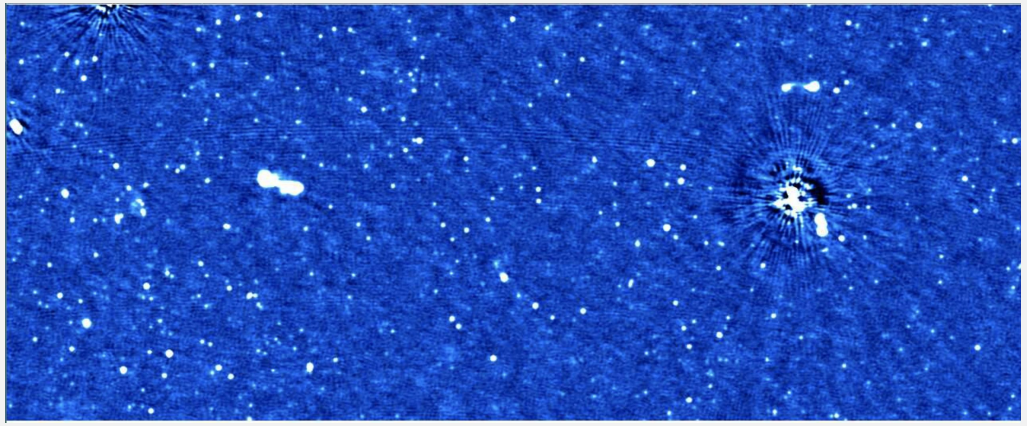
Credit: NRAO

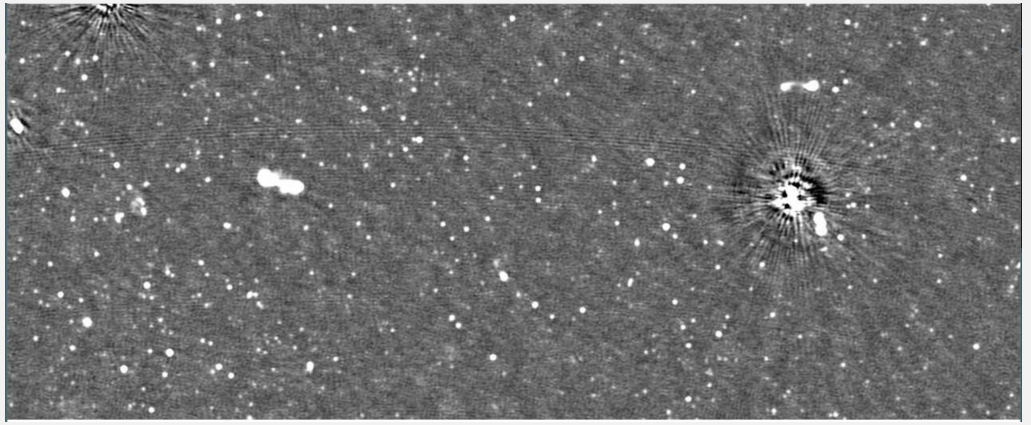
UNIVERSITY OF CAMBRIDGE

Images are not perfect









And you can choose different color schemes

UNIVERSITY OF CAMBRIDGE

Question Time

Why not perfect?

UNIVERSITY OF CAMBRIDGE

About me

- Bachelor: Electronic Engieering
- PhD: Astrophyscis - developing imaging and data anlaysis algorithms for radio interferometers
- Postdoc: worked with FAST*, LOFAR*, now working for SKA*

*FAST, LOFAR, SKA are radio telescope names

UNIVERSITY OF CAMBRIDGE

About you

UNIVERSITY OF CAMBRIDGE

Quiz Time

What was your major during your undergraduate studies?

Waiting for responses ...

Mentimeter

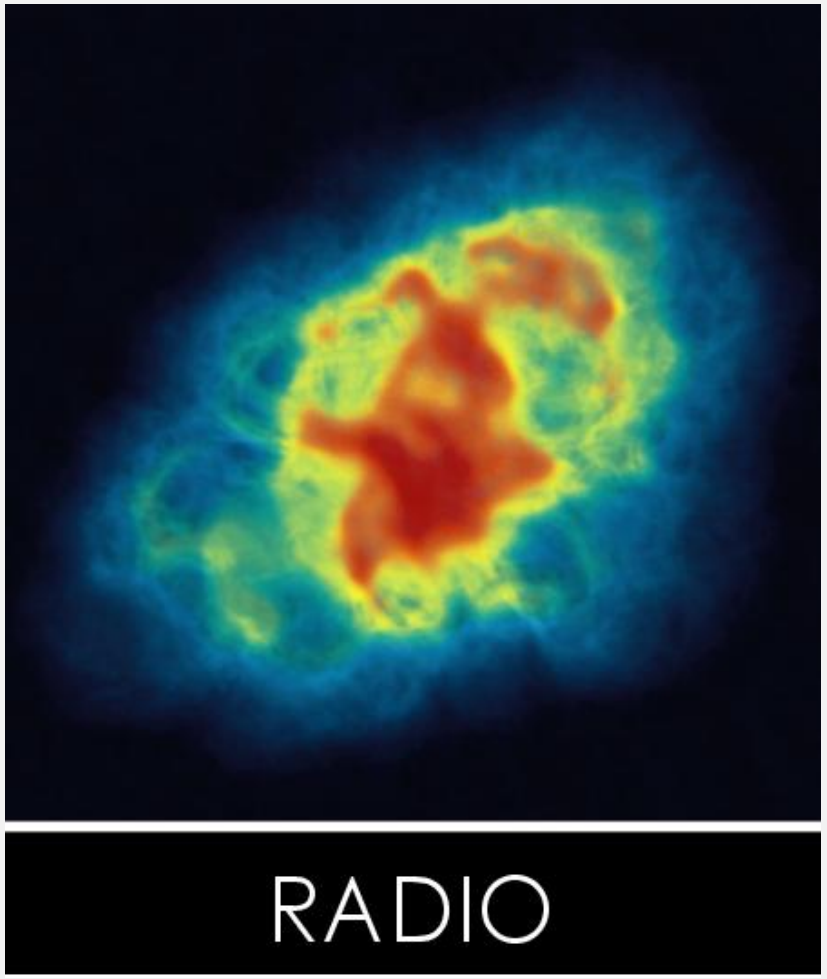
UNIVERSITY OF CAMBRIDGE

Quiz Time

Will I consider doing astronomy related jobs/studies?

I am applying	Maybe	Maybe not	No
0	0	0	0

Mentimeter



Credit: NRAO

Sky Brightness Distribution

$$I(\theta,\phi,\nu)$$

θ and ϕ are the angular coordinates (e.g., right ascension and declination in celestial coordinates)
ν is the frequency

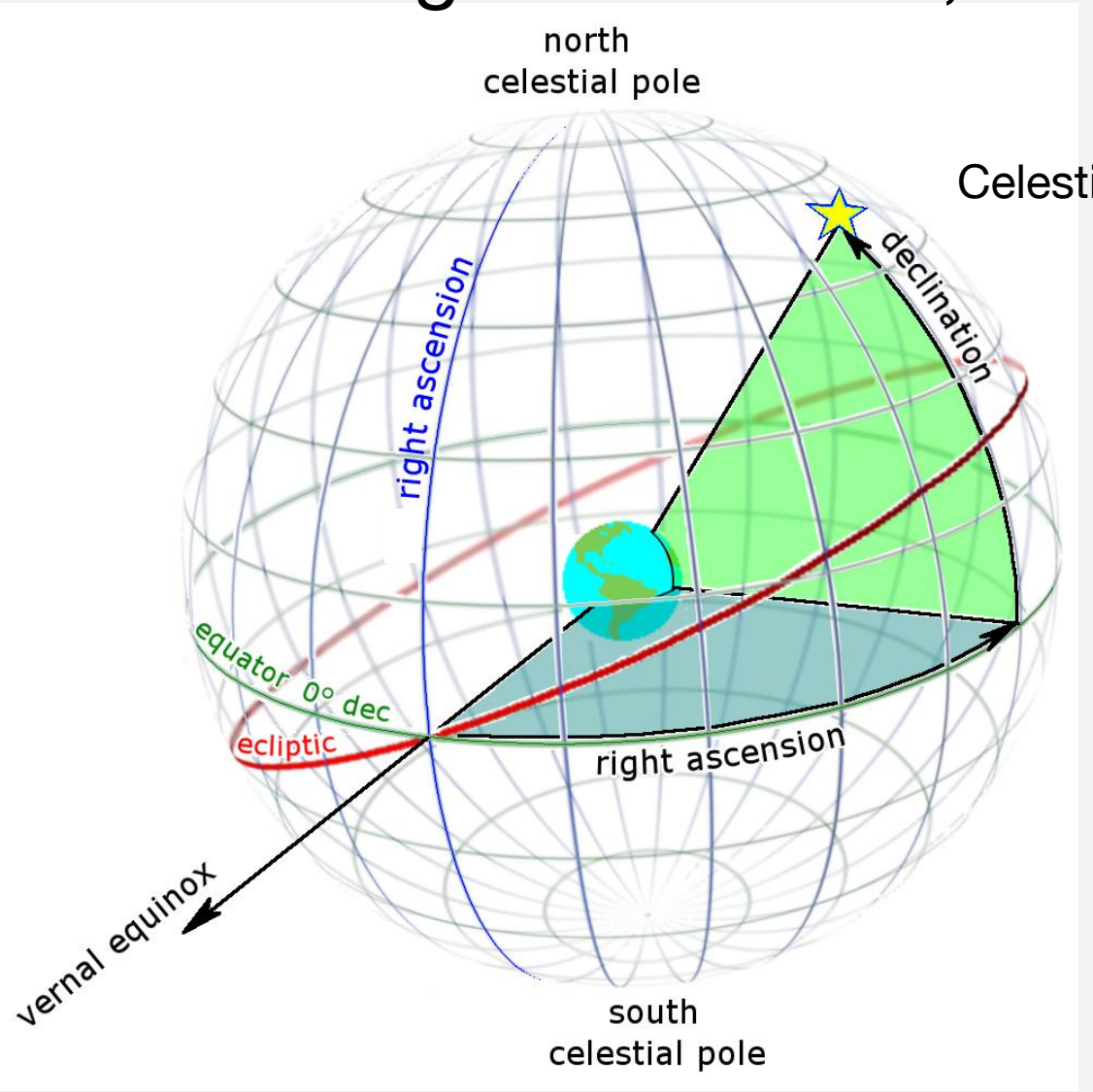
It describes how the intensity of radio emission varies across different directions in the sky at a specific wavelength or frequency.

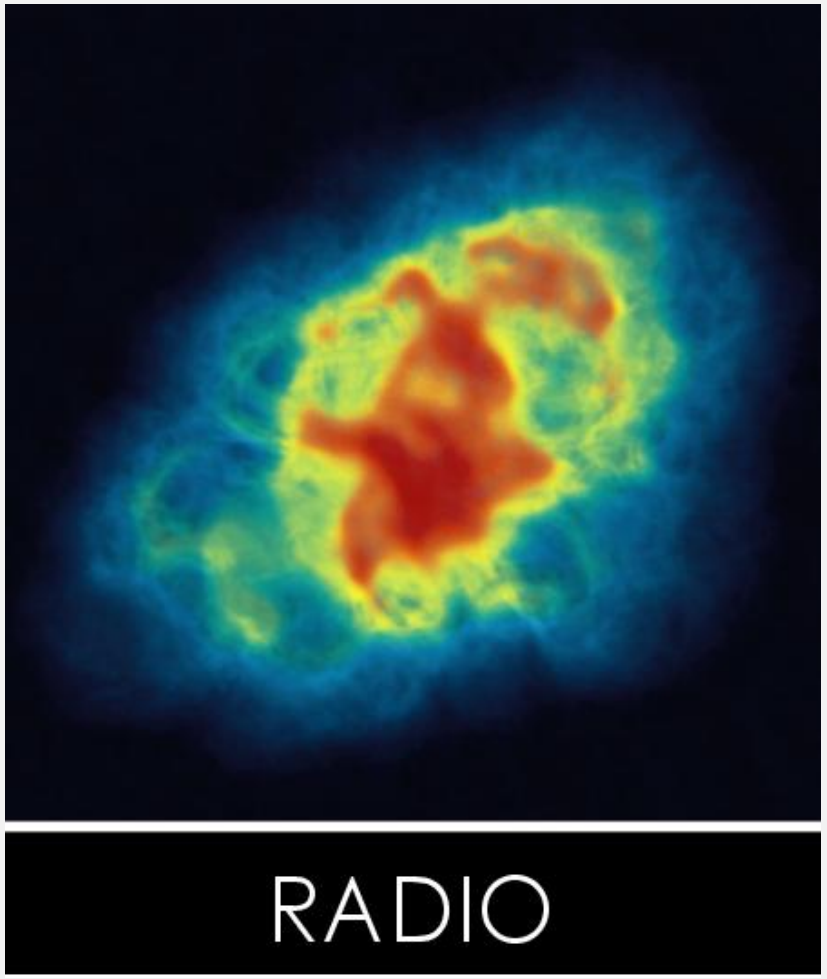
UNIVERSITY OF CAMBRIDGE

Sky Coordinates: Right Ascension, Declination

$$I(\theta,\phi)$$

- θ Right Ascension (RA) is the celestial equivalent of longitude.
- ϕ Declination (DEC) is the celestial equivalent of latitude.





Credit: NRAO

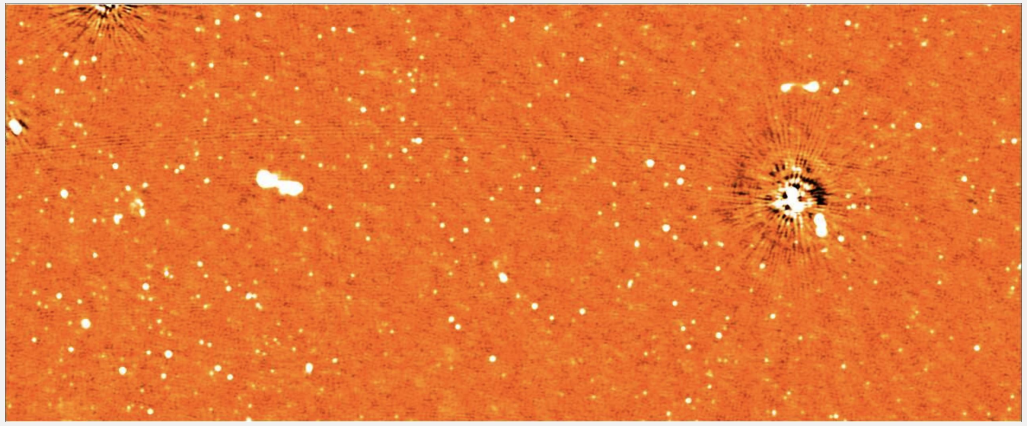
Sky Brightness Distribution

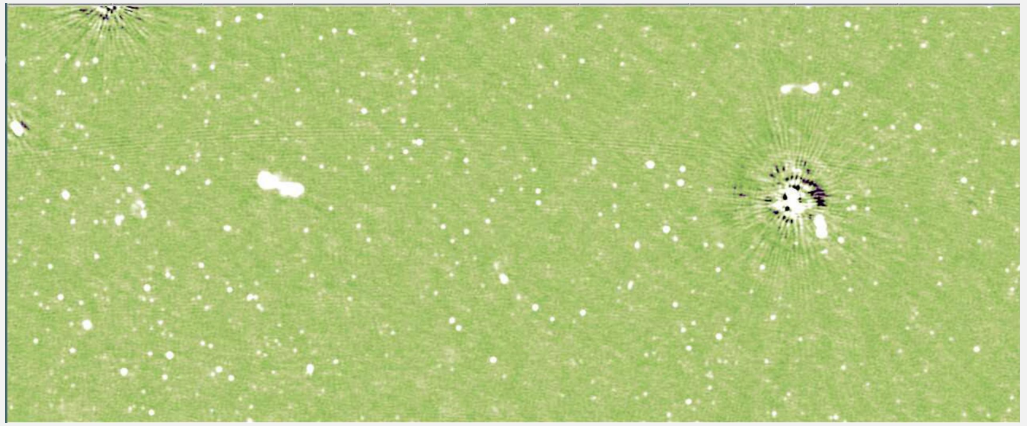
$$I(\theta,\phi,\nu)$$

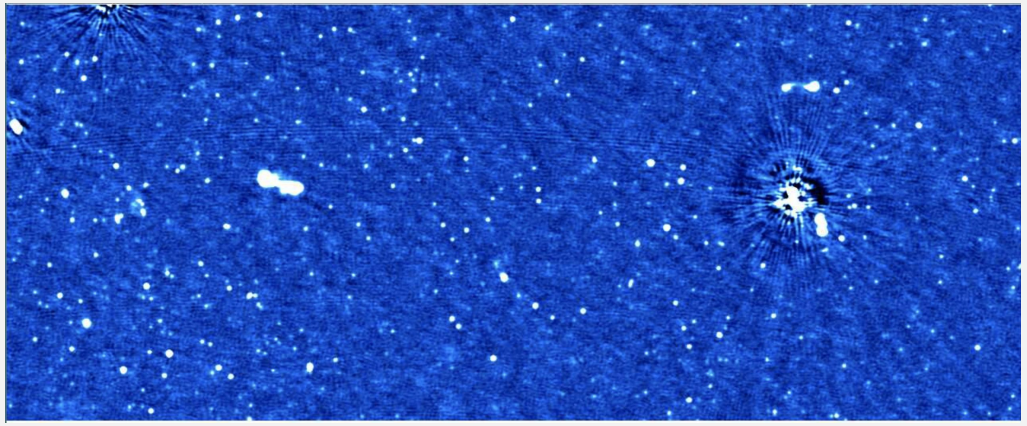
Which area has higher radio emission intensity?

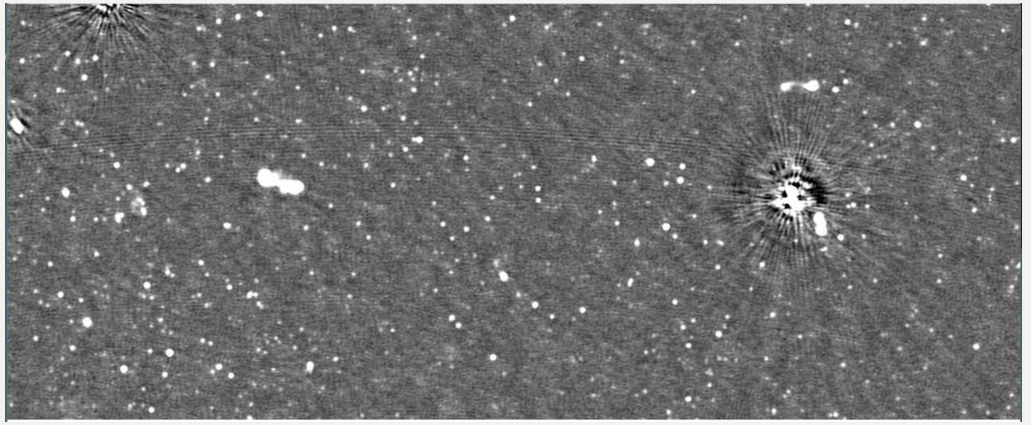
UNIVERSITY OF CAMBRIDGE

Display exmples





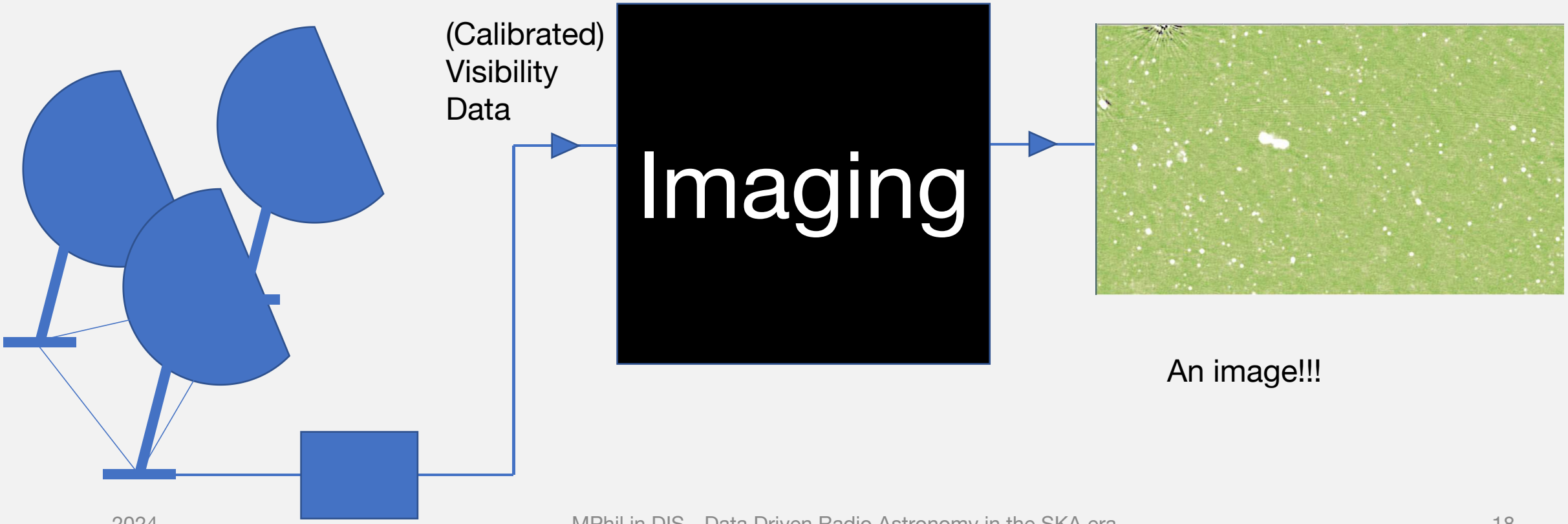




Which area has higher radio emission intensity?

UNIVERSITY OF CAMBRIDGE

But we start from visibility data!!!!



Visibility data $V(u,v,w)$ - read the data

$V(u,v,w)$ is usually stored inside a .ms file, short for"Measurement Set" .

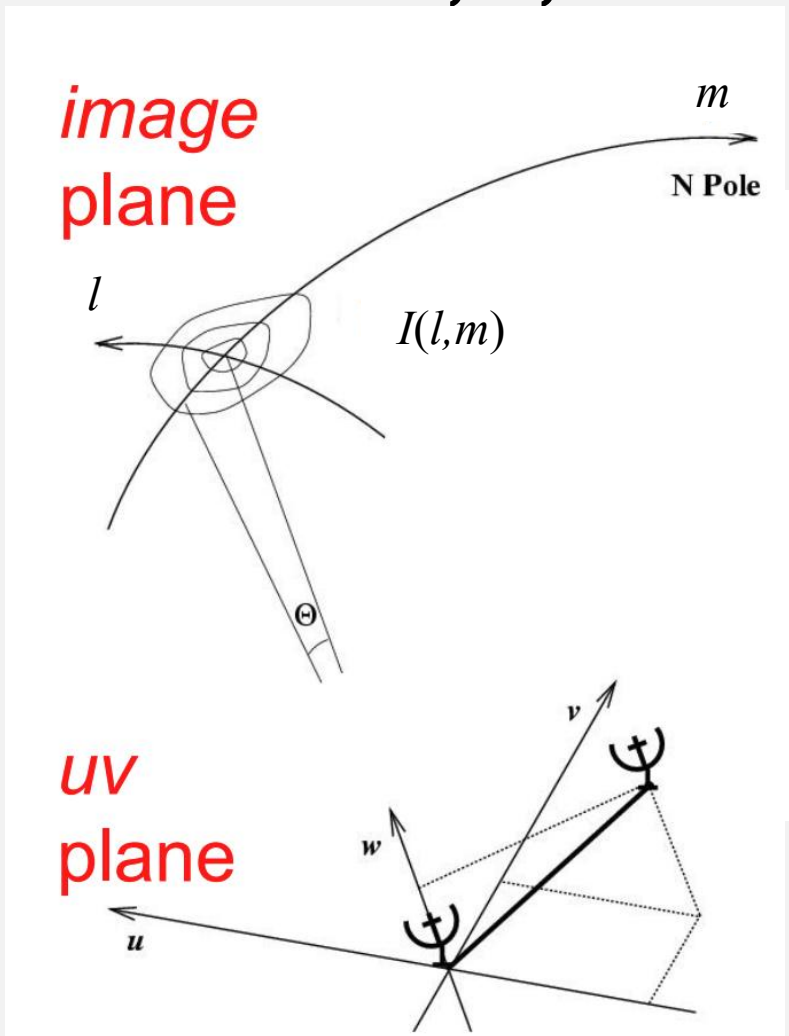
UNIVERSITY OF CAMBRIDGE

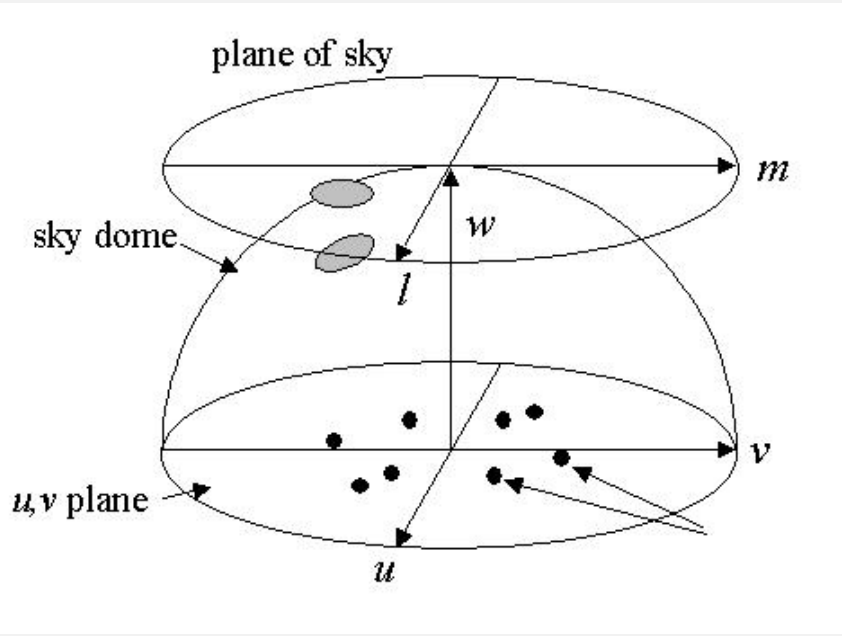
Baseline Coordinates: u,v,w

$$V(u,v,w)$$

???

$$I(l,m)$$





Check more about coordinates: https://science.nrao.edu/science/meetings/2018/16th-synthesis-imaging-workshop/talks/Perley_Geometry_new.pdf

UNIVERSITY OF CAMBRIDGE

Visibility data $V(u,v,w)$- read the data

$V(u,v,w)$ is usually stored inside a .ms file, short for"Measurement Set" .

```
>> import pyrap.tables as pt
>> MS = '/PATH/to/DATA.ms'  # measurement set
>> maintable = pt.table(MS)     # read measurement set
>> maintable.colnames()         # Check column names
```

```
>>> MS = '/home/hye/Desktop/complist_only1.vla.a.ms'
>>> maintable = pt.table(MS)
Successful readonly open of default-locked table /home/hye/Desktop/complist_only1.vla.a.ms: 24 columns, 28080 rows
>>> maintable.colnames()
['UVW', 'FLAG', 'FLAG_CATEGORY', 'WEIGHT', 'SIGMA', 'ANTENNA1', 'ANTENNA2', 'ARRAY_ID', 'DATA_DESC_ID', 'EXPOSURE', 'FEED1',
 'FEED2', 'FIELD_ID', 'FLAG_ROW', 'INTERVAL', 'OBSERVATION_ID', 'PROCESSOR_ID', 'SCAN_NUMBER', 'STATE_ID', 'TIME', 'TIME_CEN
TROID', 'DATA', 'MODEL_DATA', 'CORRECTED_DATA']
```

UNIVERSITY OF CAMBRIDGE

Visibility data $V(u,v,w)$- inspect the data

```
>> import pyrap.tables as pt
>> MS = '/PATH/to/DATA.ms'  # measurement set
>> maintable = pt.table(MS)     # read measurement set
>> maintable.colnames()         # Check column names
>> uvw_column = maintable.getcol("UVW")      # get coordinates u,v,w
>> udata, vdata, wdata=uvw_column[:,0], uvw_column[:,1], uvw_column[:,2]
```

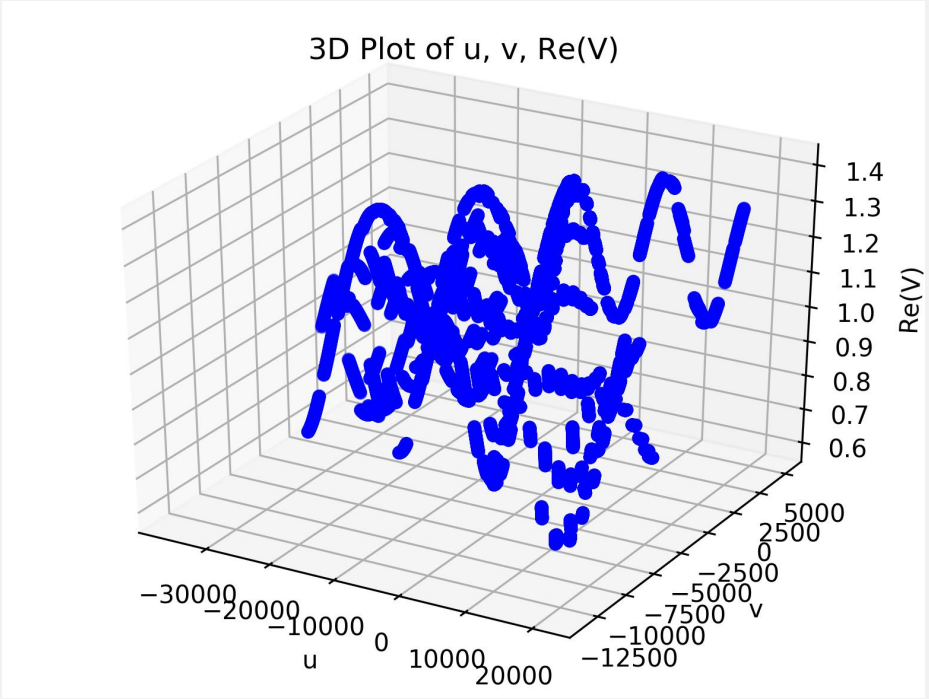
```
>>> uvw_column = maintable.getcol("UVW")
>>> vdata=uvw_column[:,1]
>>> udata=uvw_column[:,0]
>>> udata.shape
(28080,)
>>> vdata.shape
(28080,)
>>> udata
array([ 922.82439341, 2256.7422902 , 3954.86445702, ...,  346.07608855,
        724.5101658 ,  378.43407725])
```

UNIVERSITY OF CAMBRIDGE

Visibility data $V(u,v,w)$ - inspect the data

```
>> import pyrap.tables as pt
>> MS = '/PATH/to/DATA.ms'  # measurement set
>> maintable = pt.table(MS)     # read measurement set
>> maintable.colnames()         # Check column names
>> vis = maintable.getcol("DATA") # get visibility data
```

```
>>> vis = maintable.getcol("DATA")
>>> vis.shape
(28080, 1, 2)
>>> vis[:,:,0]
array([[1.3615973+2.0712614e-06j],
       [1.2061307+2.6077032e-07j],
       [1.0166397-9.7304583e-06j],
       ...,
       [1.3638289-5.6065619e-06j],
       [1.2508852-6.7204237e-06j],
       [1.3567369-5.9716403e-06j]], dtype=complex64)
```



UNIVERSITY OF CAMBRIDGE

Coding Time

Given that each pair of antennas (a baseline) generates one visibility data per frequency channel per second,

Do complete this Python function to calculates the total number of visibility data generated by an array of antennas.

UNIVERSITY OF CAMBRIDGE

Coding Time

Given that each pair of antennas (a baseline) generates one visibility data per frequency channel per second,

Do complete this Python function to calculates the total number of visibility data generated by an array of antennas.

```
def calculate_visibility_num(antennas, observation_time, frequency_channel):
    """
    Calculate the total number of visibility data generated by an array of antennas.

    Parameters:
        antennas (int): Number of antennas in the array.
        observation_time (int): Observation time in seconds.
        frequency_channel (int): Number of frequency channels.

    Returns:
        int: Total number of visibility data generated by the array.
    """
    # Calculate the number of pairs of antennas
    if antennas < 2:
        baseline_num = 0  # No baselines if there are less than 2 antennas
    else:
        baseline_num = ???????

    # Calculate the total visibility data
    visibility_num = baseline_num * observation_time * frequency_channel

    return visibility_num
```

UNIVERSITY OF CAMBRIDGE

Coding Time - unit test

Given that each pair of antennas (a baseline) generates one visibility data per frequency channel per second,

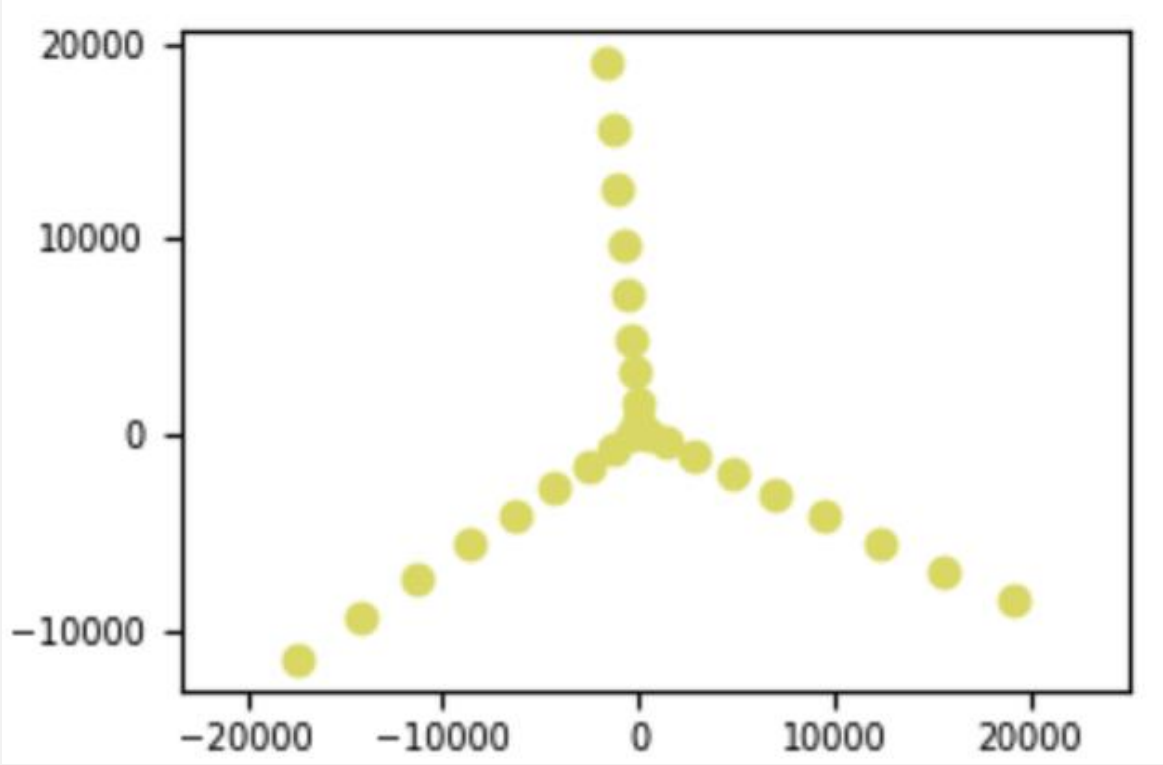
Do complete this Python function to calculates the total number of visibility data generated by an array of antennas.

```
import unittest

class TestCalculateVisibilityData(unittest.TestCase):

    def test_visibility_data_calculation(self):
        # Test case with 4 antennas, 60 seconds observation time, and frequency
channel 10
        result = calculate_visibility_num(4, 60, 10)
        self.assertEqual(result, 3600)  # Expected result: 6 pairs of antennas * 60
seconds * 10 frequency channels = 3600

        # Test case with 2 antennas, 30 seconds observation time, and frequency
channel 5
        result = calculate_visibility_num(2, 30, 5)
        self.assertEqual(result, 150)  # Expected result: 1 pair of antennas * 30 seconds
* 5 frequency channels = 150

        # Test case with 1 antenna, 10 seconds observation time, and frequency
channel 1
        result = calculate_visibility_num(1, 10, 1)
        self.assertEqual(result, 0)  # Expected result: No pairs of antennas
```

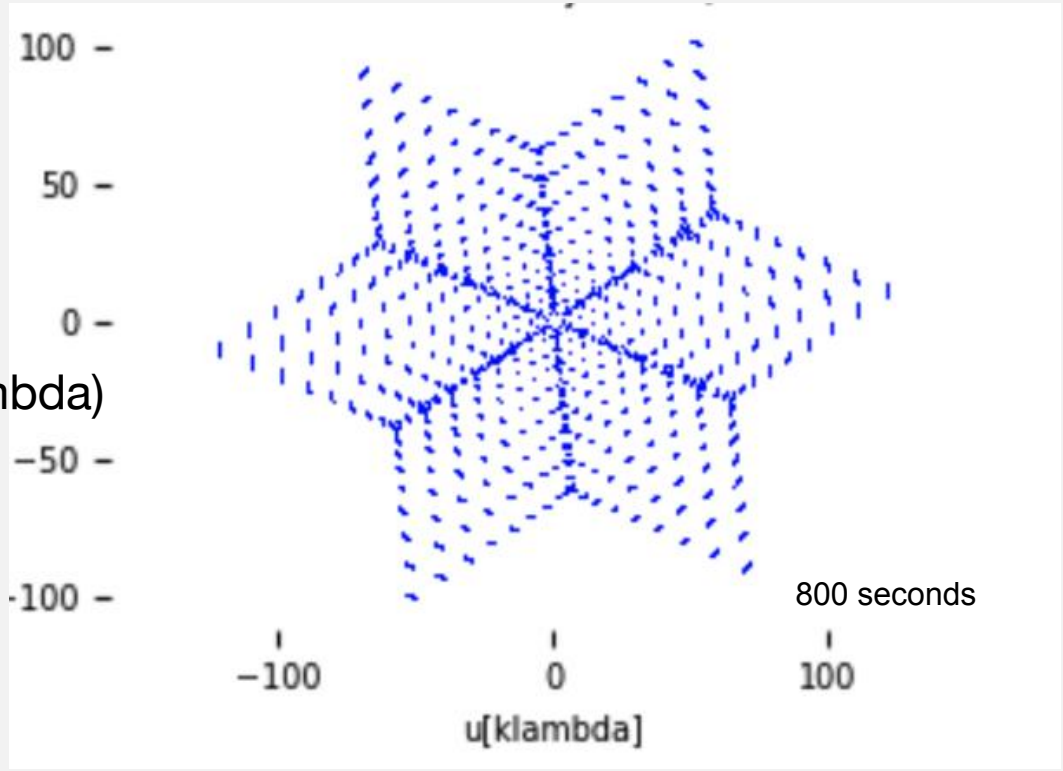
UNIVERSITY OF CAMBRIDGE

Visibility data $V(u,v,w)$ - plot the data

Antenna location



uv-coverage

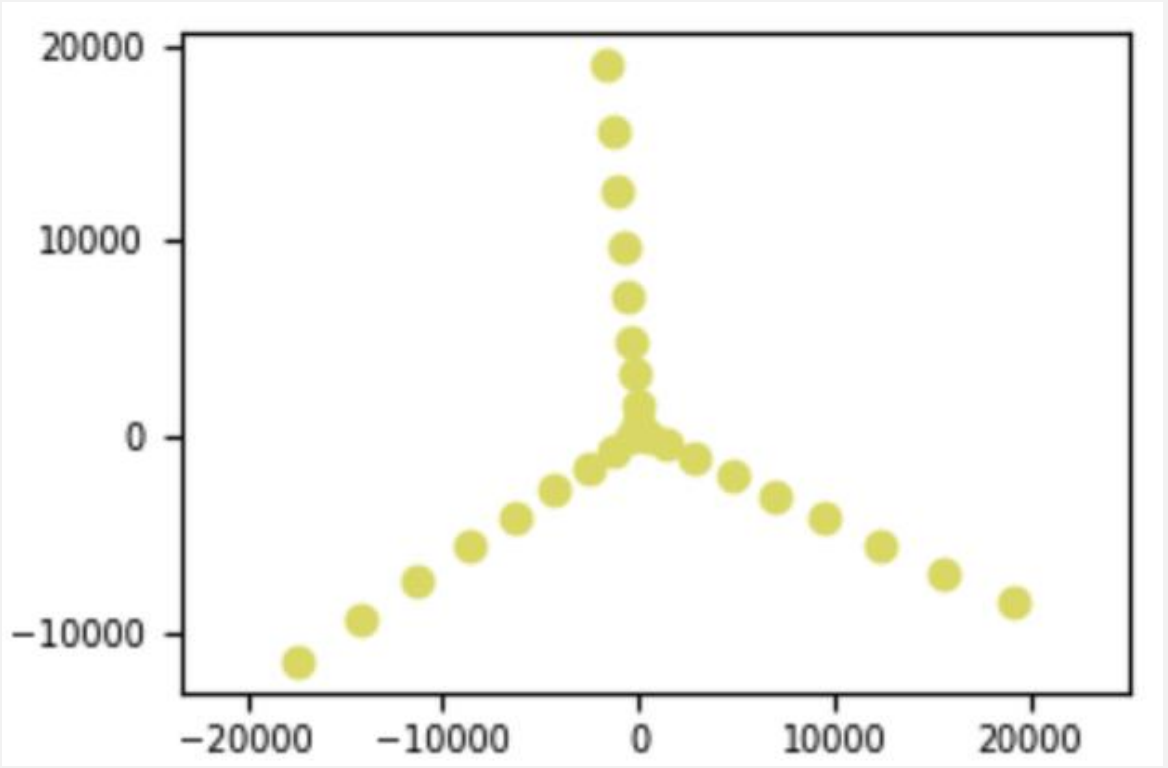


Generated using CASA

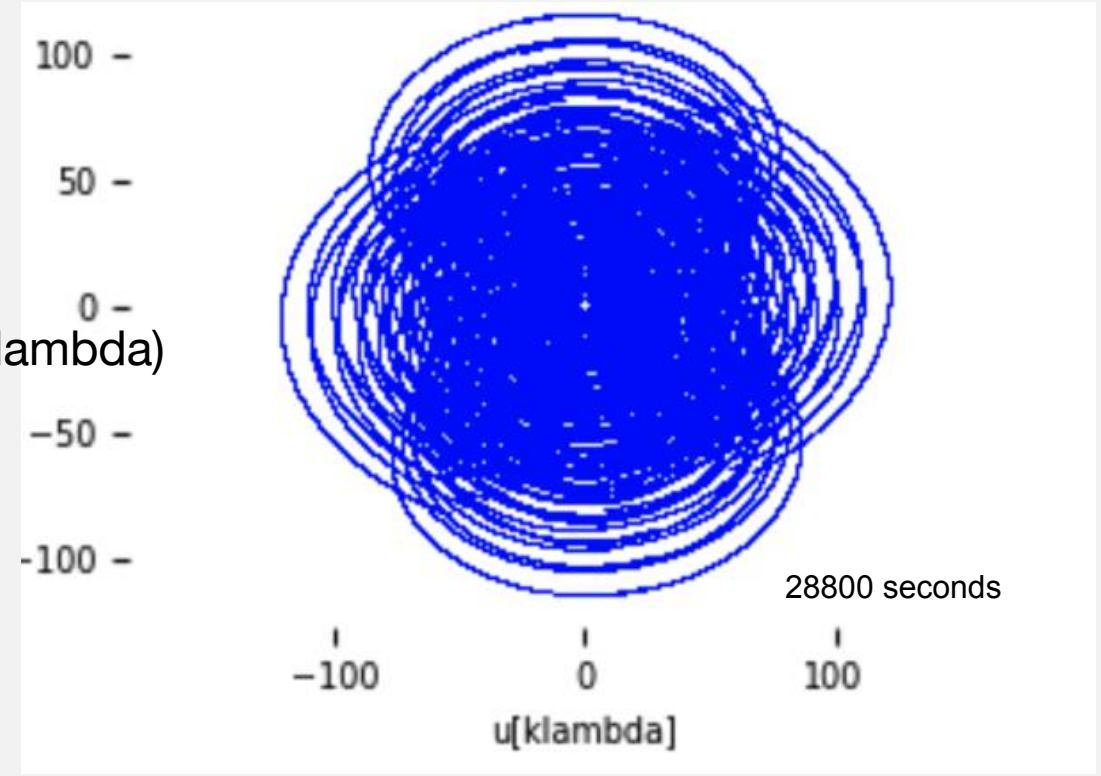
UNIVERSITY OF CAMBRIDGE

Visibility data $V(u,v,w)$ - plot the data

Antenna location



uv-coverage



Generated using CASA

UNIVERSITY OF CAMBRIDGE

Question Time

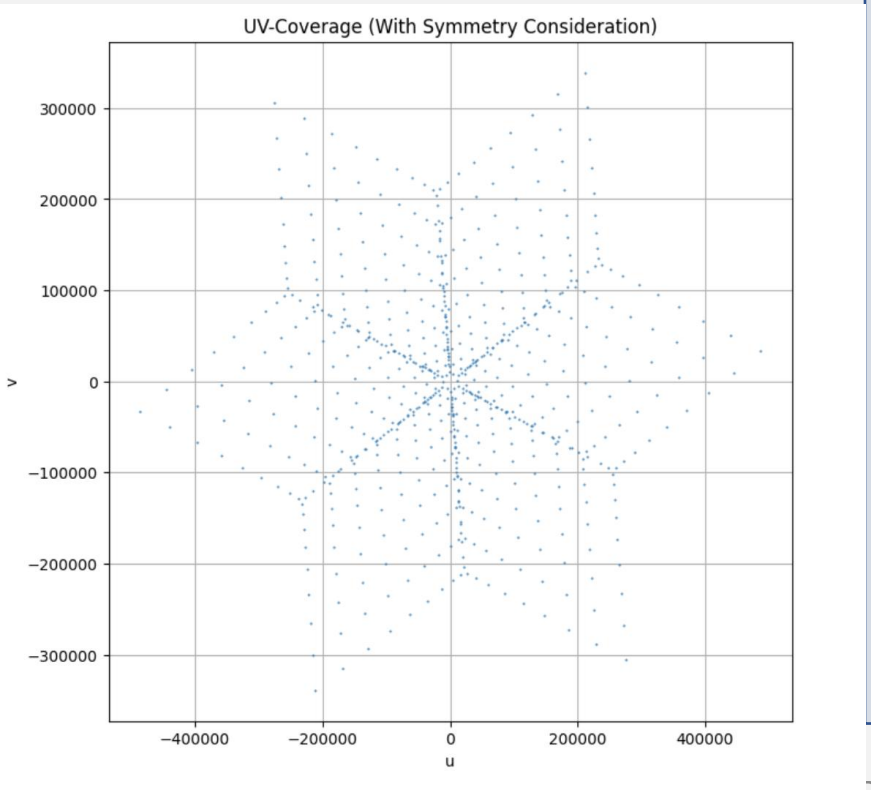
What changed?

UNIVERSITY OF CAMBRIDGE

Coding Time

Given a visibility data V(u,v,w),

Do plot uv-coverage.




```
# Load the data
Vis = np.loadtxt('onesource.csv', usecols=[0, 1, 2, 3, 4])
u = ????
v = ????

# Plot the uv-coverage
plt.figure(figsize=(8, 8))
plt.plot(u, v, '.', markersize=1)
plt.xlabel('u')
plt.ylabel('v')
plt.title('UV-Coverage (With Symmetry Consideration)')
plt.grid(True)
plt.show()
```

UNIVERSITY OF CAMBRIDGE

From the basics

UNIVERSITY OF CAMBRIDGE

Quiz

Is it a radio interferometer? Or not?

Frequency range:
12 to 18 GHz

Frequency range:
10 to 240 GHz

Frequency range:
119 to 374 THz

Frequency range:
12 to 999THz

UNIVERSITY OF CAMBRIDGE

Discussion

Is it a radio interferometer? Or not?

Frequency range:
12 to 18 GHz

AMI

Frequency range:
10 to 240 GHz

LOFAR
(low: 10-80 MHz
High: 120-
240MHz)

Frequency range:
119 to 374 THz

Hubble

Frequency range:
12 to 999THz

VLT

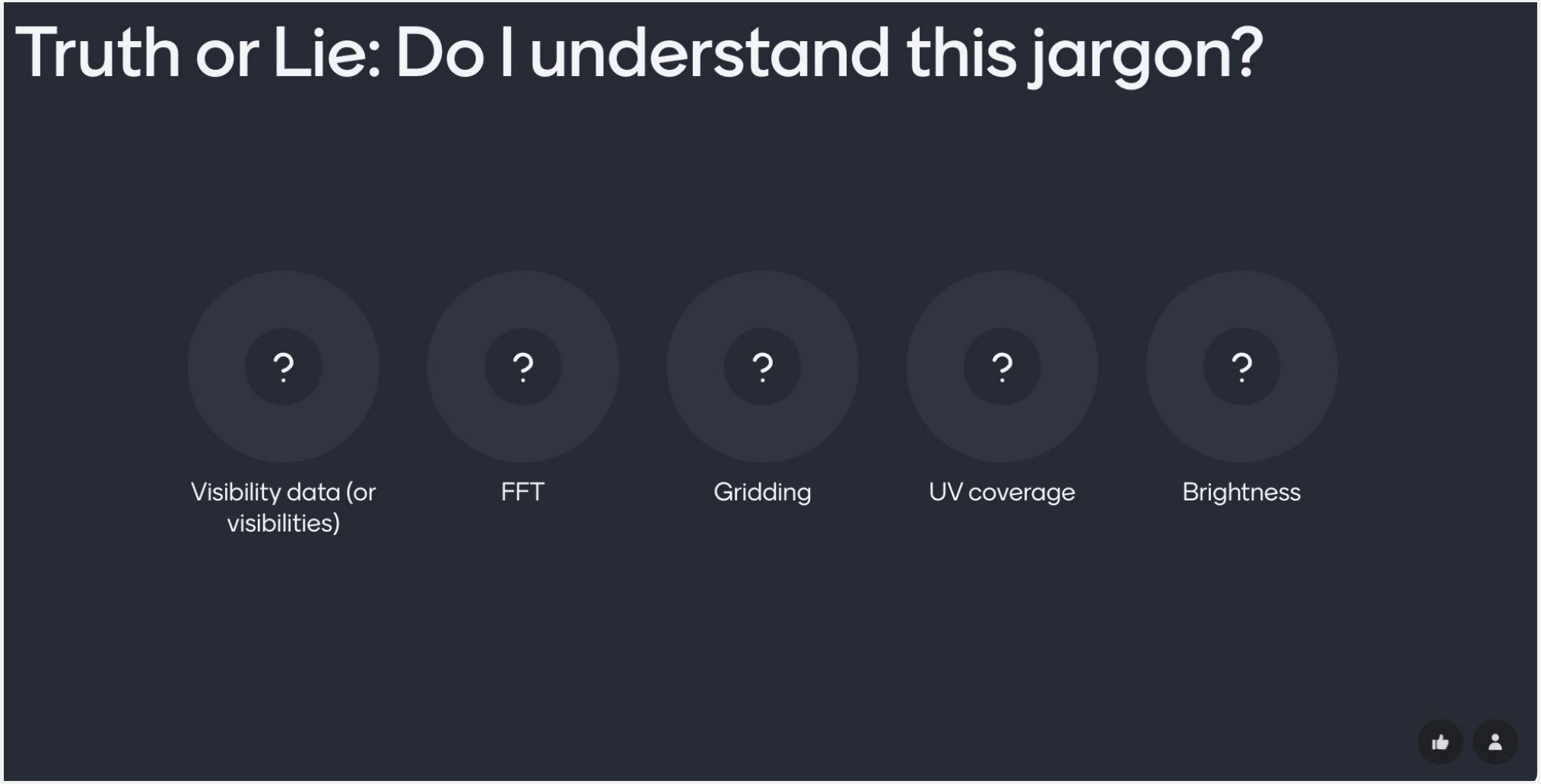
UNIVERSITY OF CAMBRIDGE

Discussion

To qualify as a radio interferometer, how many antennas must it possess?

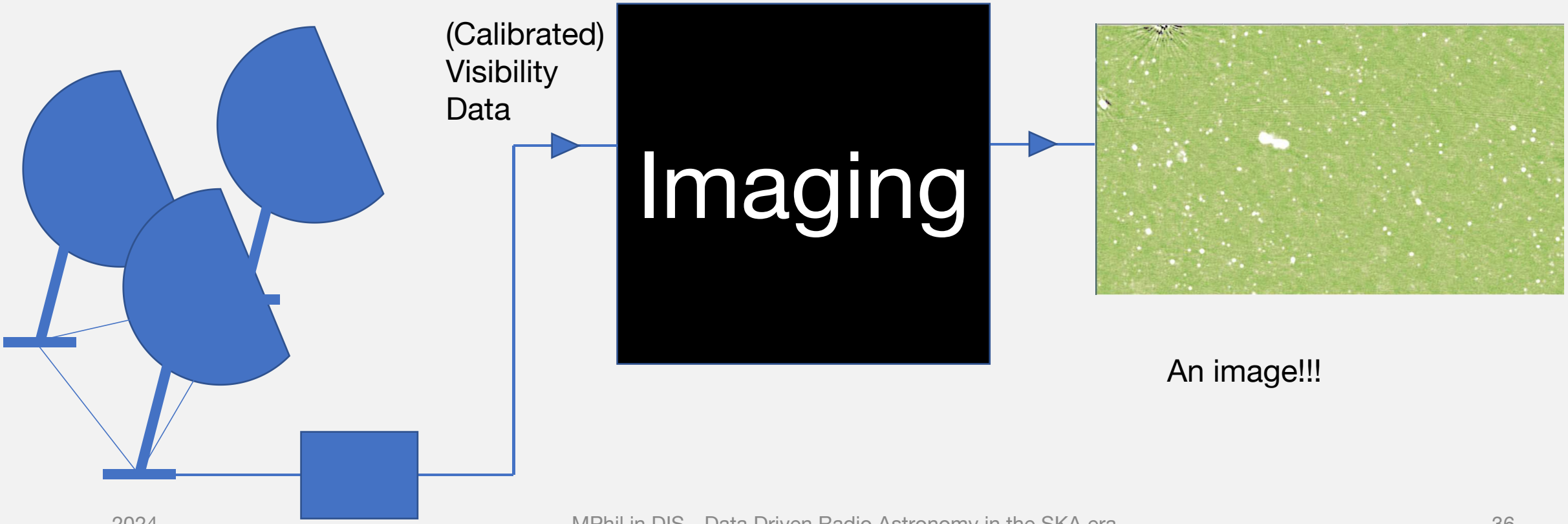
UNIVERSITY OF CAMBRIDGE

Discussion



UNIVERSITY OF CAMBRIDGE

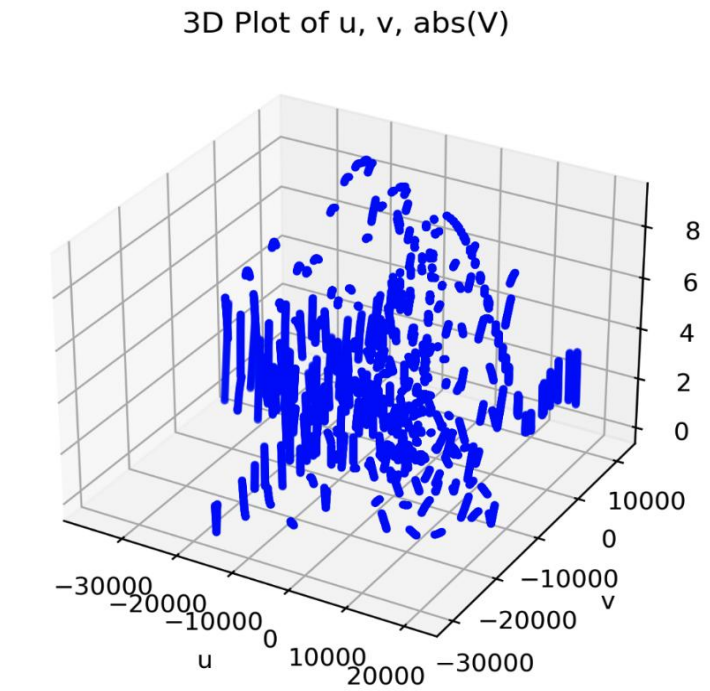
And how do we convert visibility data to an image???



UNIVERSITY OF CAMBRIDGE

How to convert visibility data into imaging?

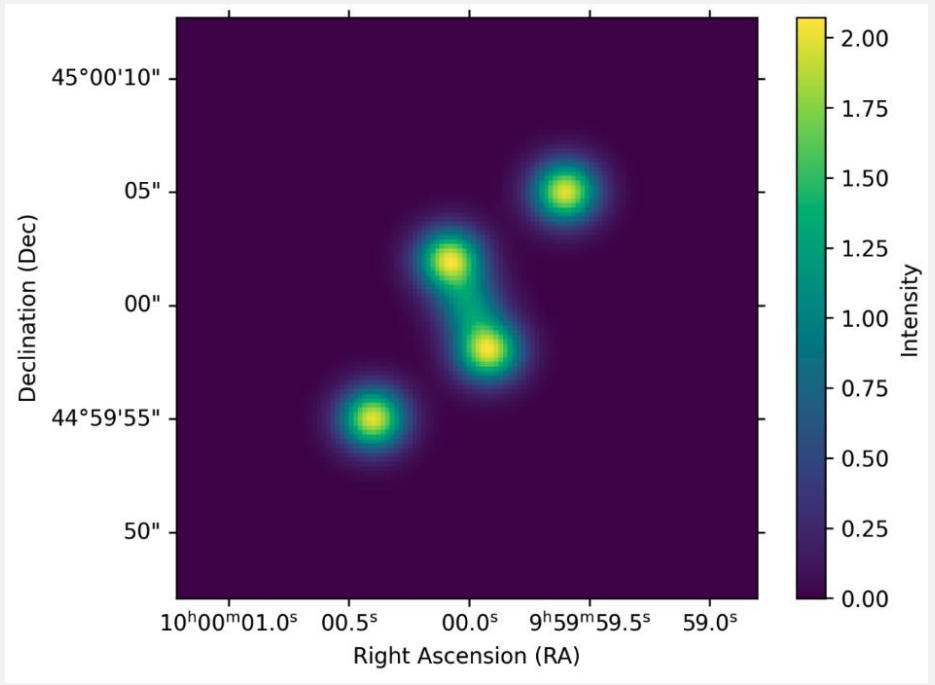
Visibilitiy Plane



Visibilitiy Data

$$V(u,v,w)$$

Image Plane



Sky Brightness Distribution

$$I(l,m)$$

UNIVERSITY OF CAMBRIDGE

How to convert visibility data into imaging?

$$V(u,v,w) = \int\int \frac{I(l,m)}{\sqrt{1-l^2-m^2}} e^{-2\pi i\left[ul+vm+w\left(\sqrt{1-l^2-m^2}-1\right)\right]} \mathrm{d}l\,\mathrm{d}m$$

Detailed explanation can be found at https://www.aspbooks.org/publications/6/11.pdf

UNIVERSITY OF CAMBRIDGE

How to convert visibility data into imaging?

$$V(u,v,w) = \int\int \frac{I(l,m)}{\sqrt{1-l^2-m^2}} e^{-2\pi i\left[ul+vm+w\left(\sqrt{1-l^2-m^2}-1\right)\right]} \mathrm{d}l\,\mathrm{d}m$$

$$\left(\sqrt{1-l^2-m^2}-1\right)w \approx 0$$

$$V(u,v,w) = \int\int I(l,m) e^{-2\pi i\,(ul+vm)}\, dl\, dm$$

UNIVERSITY OF CAMBRIDGE

How to convert visibility data into imaging?

$$\left(\sqrt{1-l^2-m^2}-1\right)w \approx 0$$

$$V(u,v) = \int\int I(l,m) e^{-2\pi i\,(ul+vm)}\, dl\, dm$$

$$I(l,m) = \int\int V(u,v)\, e^{2\pi i\,(ul+vm)}\, du\, dv$$

UNIVERSITY OF CAMBRIDGE

How to convert visibility data into imaging?

when the field of view is small...

$$\left(\sqrt{1-l^2-m^2}-1\right)w \approx -\frac{1}{2}(l^2+m^2)w \approx 0$$

If you are interested, figure it out and tell me how you reached this!

Or when w is negligible

$$\left(\sqrt{1-l^2-m^2}-1\right)w \approx 0$$

How to convert visibility data into imaging?

$$I(l,m) =$$

Inverse Fourier Transform of $V(u,v)$

*when the field of view is small...

or when w is negligible

How to convert visibility data into imaging?

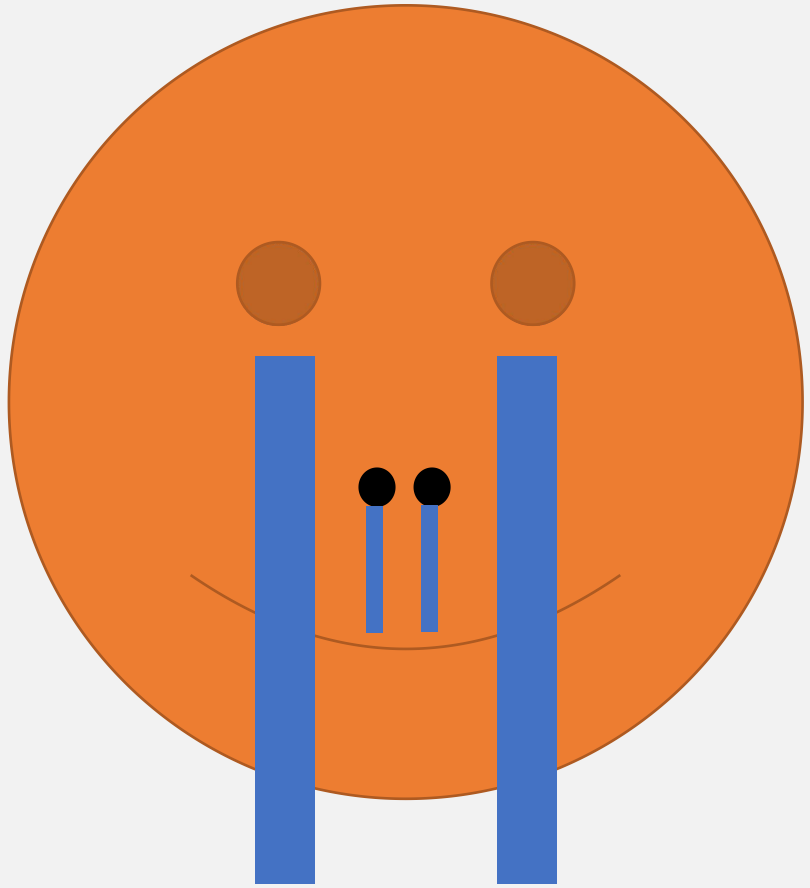
Perfect!!!!!

We just need one row of code!!!!

Hooray!!!!!

How to convert visibility data into imaging?

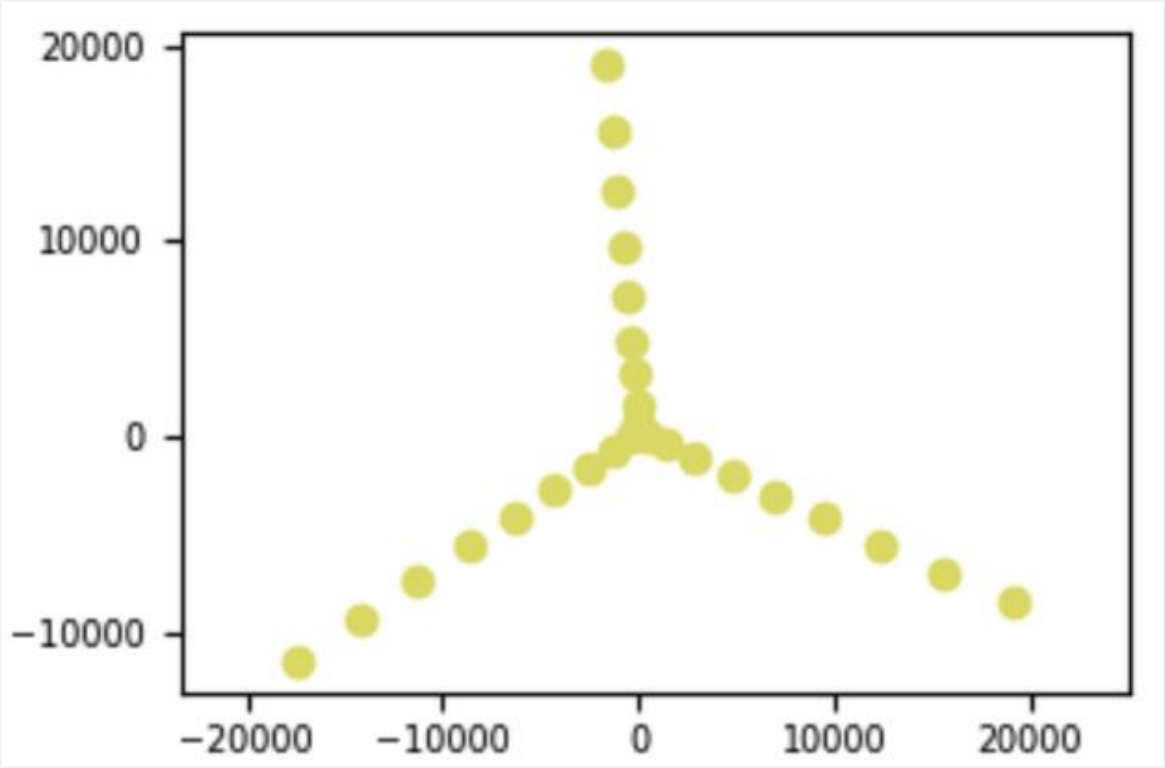
Wait.....



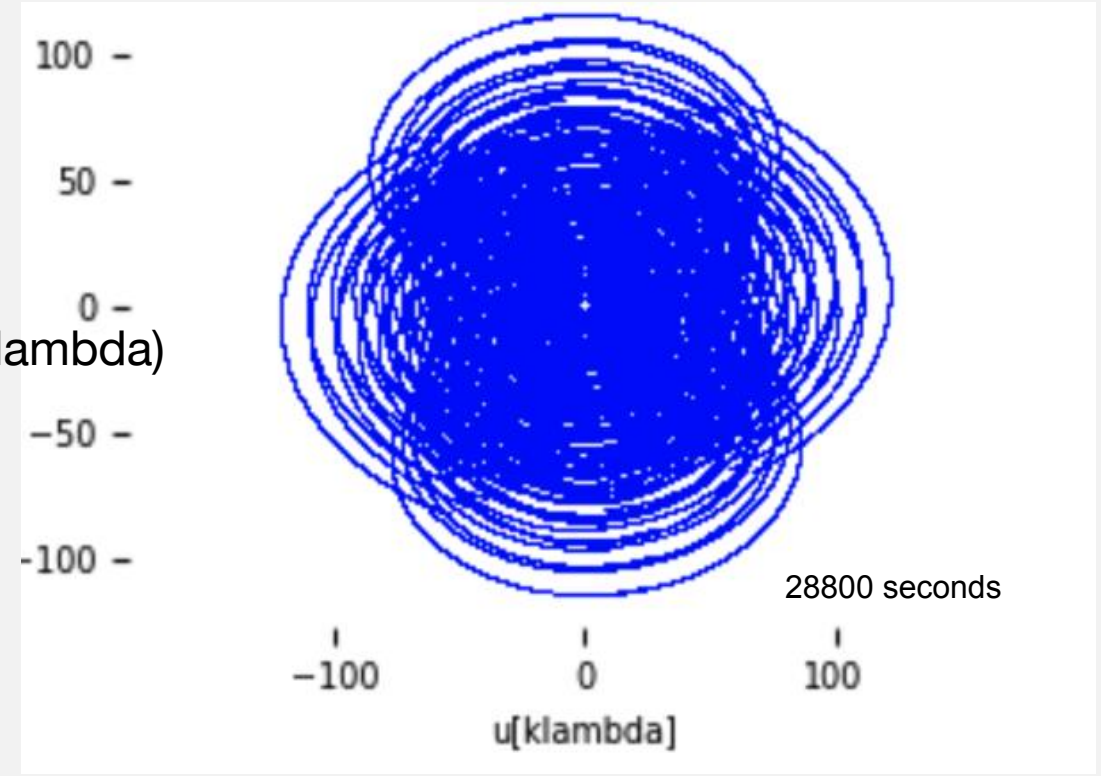
UNIVERSITY OF CAMBRIDGE

Visibility data $V(u,v,w)$ - plot the data

Antenna location



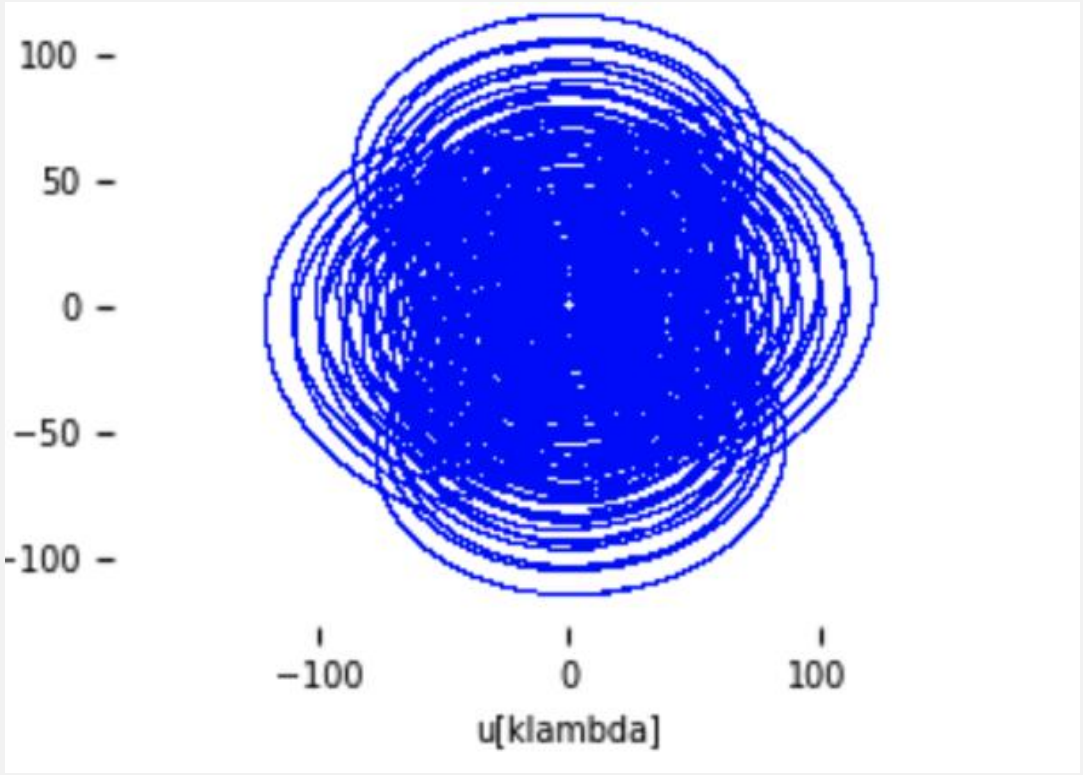
uv-coverage



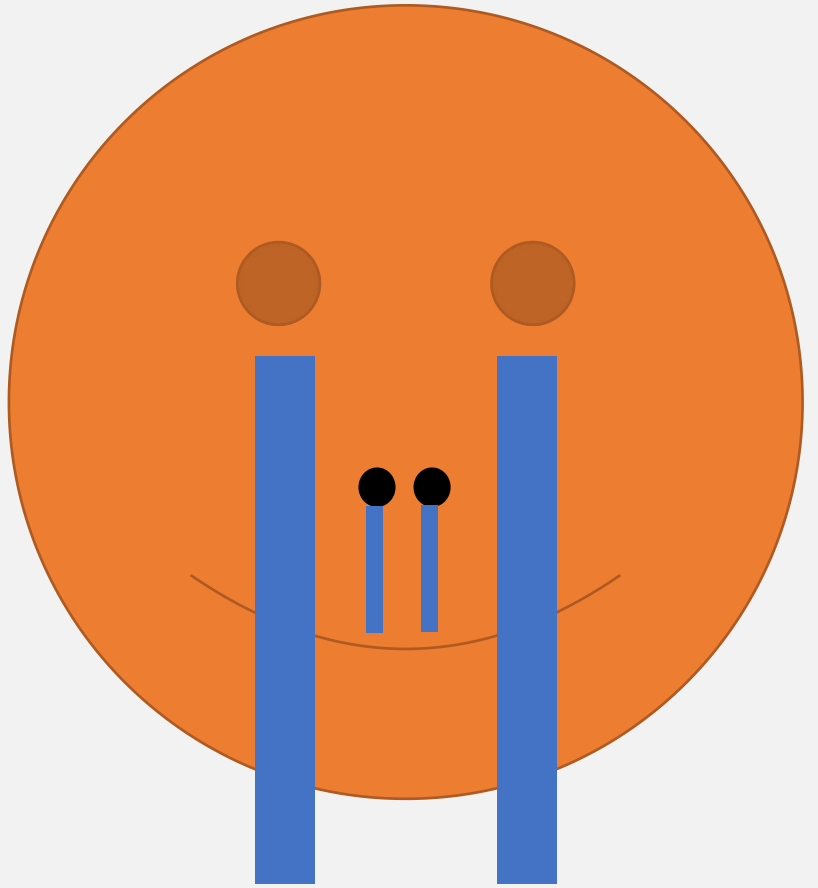
Generated using CASA

How to convert visibility data into imaging?

uv-coverage



We don’t have all uv values sampled!!!!!!!!



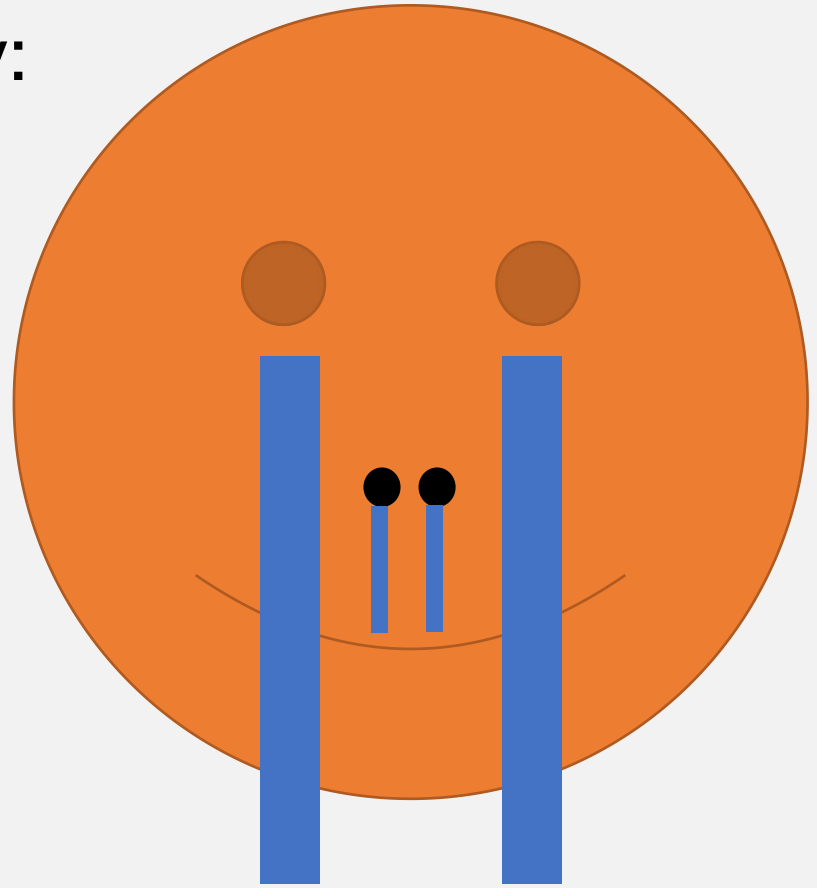
How to convert visibility data into imaging?

The visbility data we observed is actually:

$$V^s(u,v) = V(u,v)\,S(u,v)$$

$$= \sum_{k=1}^{M} \delta\left(u-u_k,\, v-v_k\right) V(u_k, v_k)$$

$$, \text{ where } S(u,v) = \sum_{k=1}^{M} \delta\left(u-u_k,\, v-v_k\right)$$

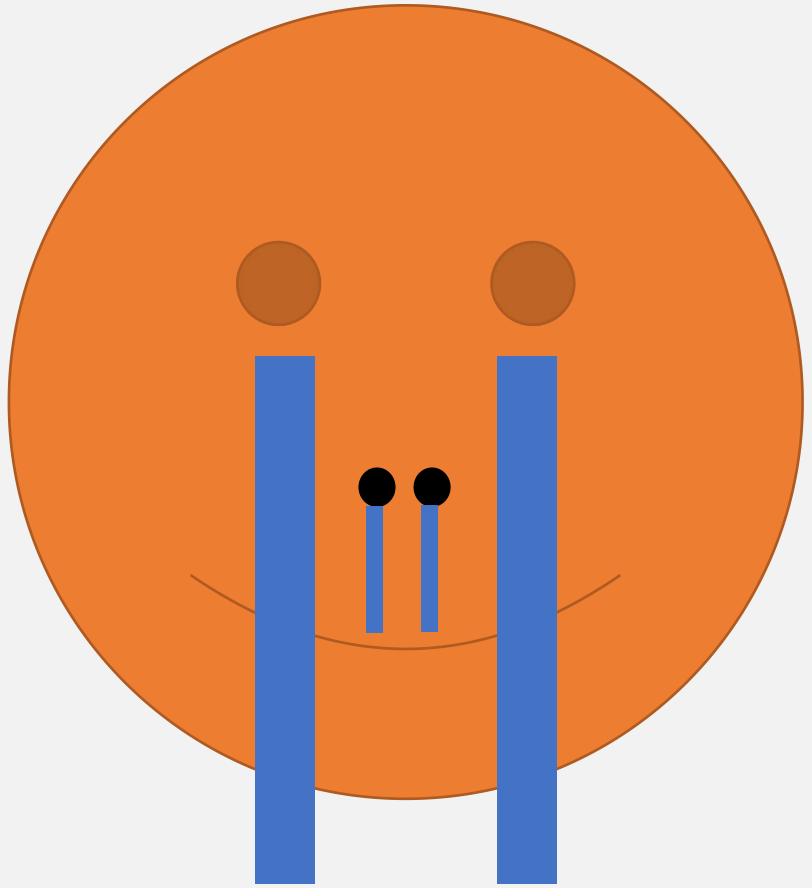


How to convert visibility data into imaging?

The Fourier Transform of

$$V(u,v)S(u,v)$$

would become???????



How to convert visibility data into imaging?

The Fourier Transform of

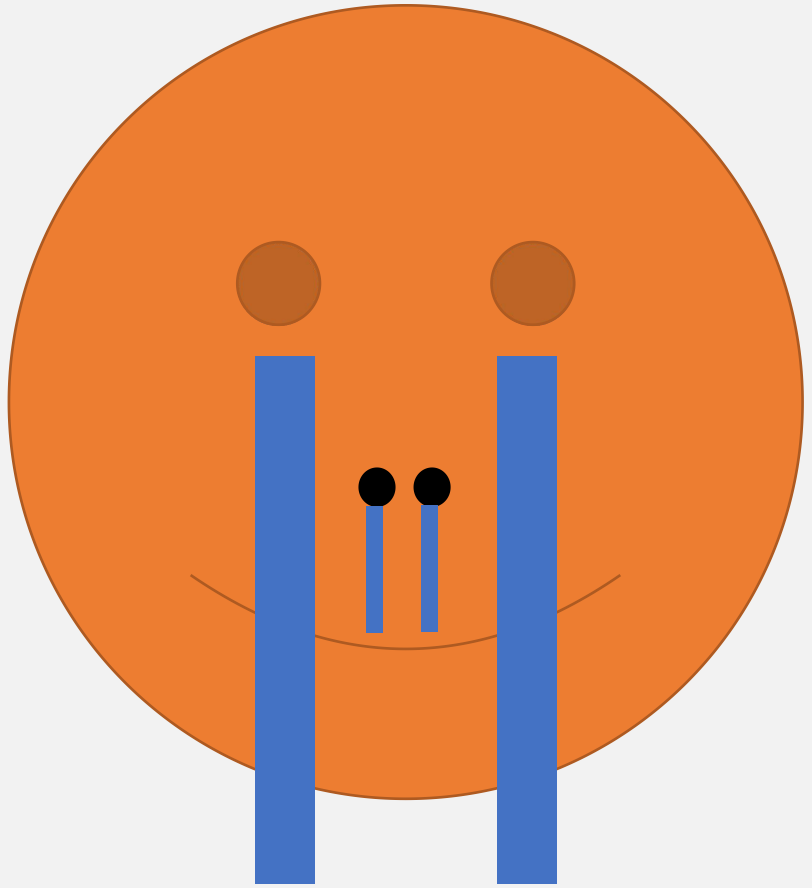
$$\mathcal{F}\left(V(u,v)S(u,v)\right) = \mathcal{F}\left(V(u,v)\right)^{*}\mathcal{F}\left(S(u,v)\right)$$

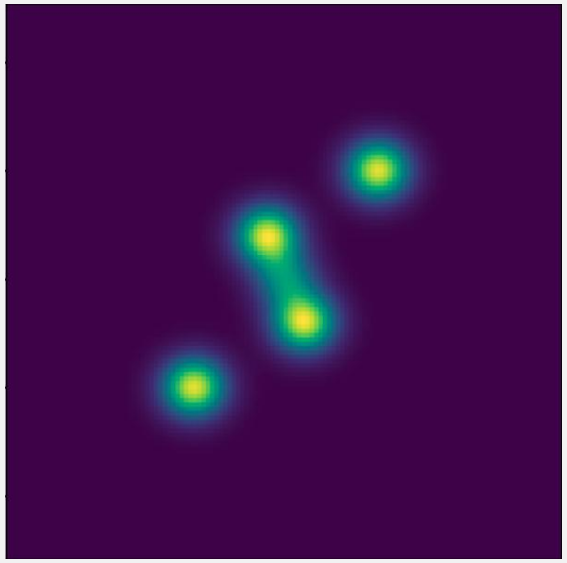
We want this!!!!

$$I_D(l,m) = I(l,m)\,^{*}B(l,m)$$

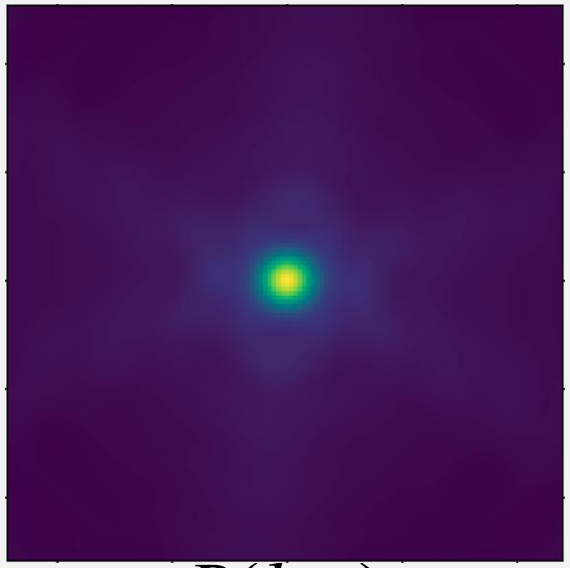
`Dirty image

Dirty beam

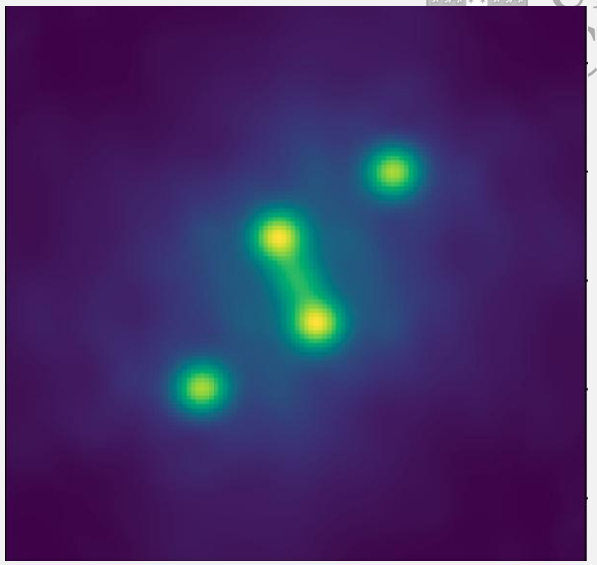




$I(l,m)$

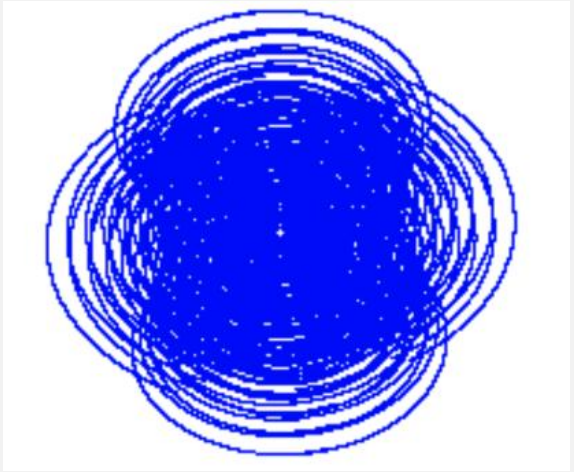


$B(l,m)$

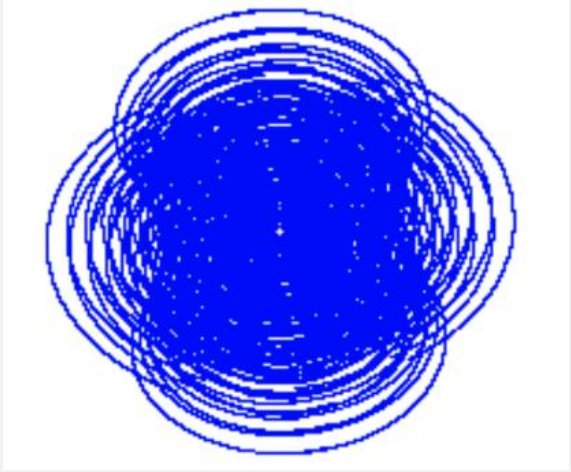


$B(l,m)*I(l,m)$

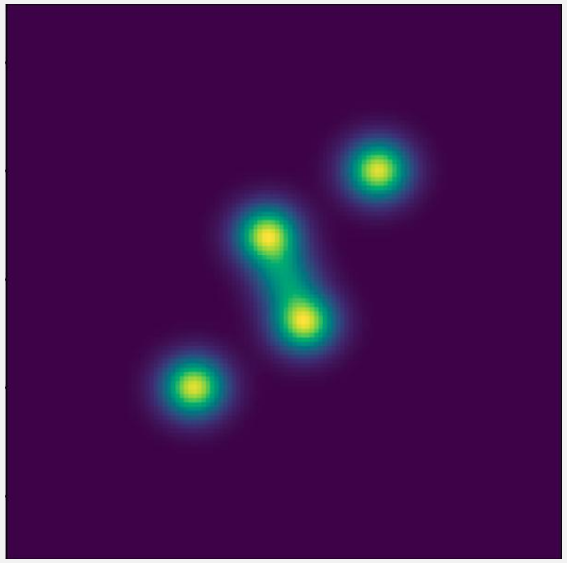
$V(u,v)$



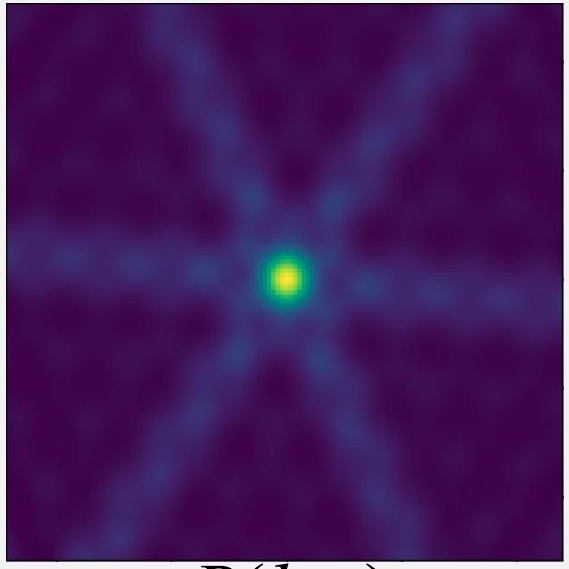
$S(u,v)$



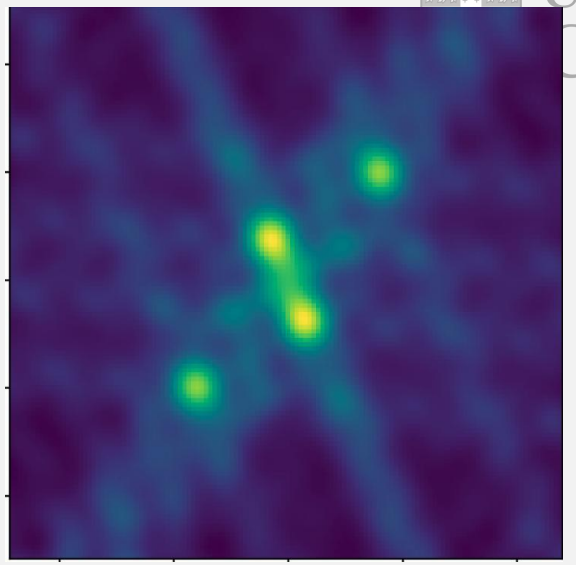
$S(u,v)V(u,v)$



$I(l,m)$

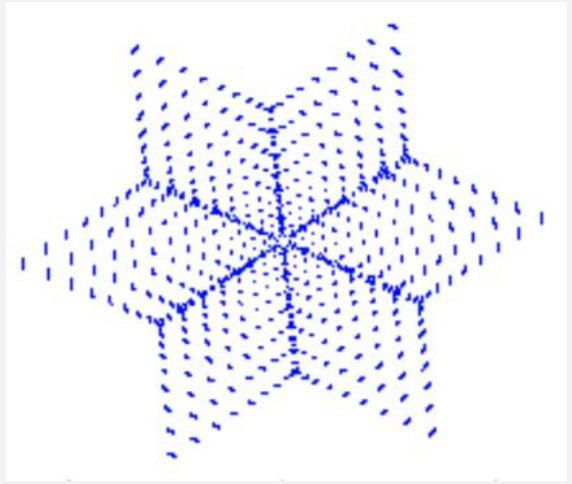


$B(l,m)$

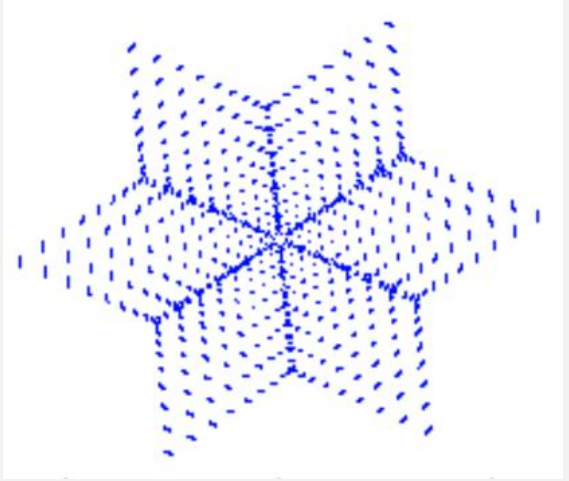


$B(l,m)*I(l,m)$

$V(u,v)$



$S(u,v)$



$S(u,v)V(u,v)$

How to convert visibility data into imaging?

Well, it is not that ideal, but I can at least get an image

UNIVERSITY OF CAMBRIDGE

Coding Time

Given the sampling function S(u,v,w),

Do image the dirty beam.

When the field of view is narrow, we have

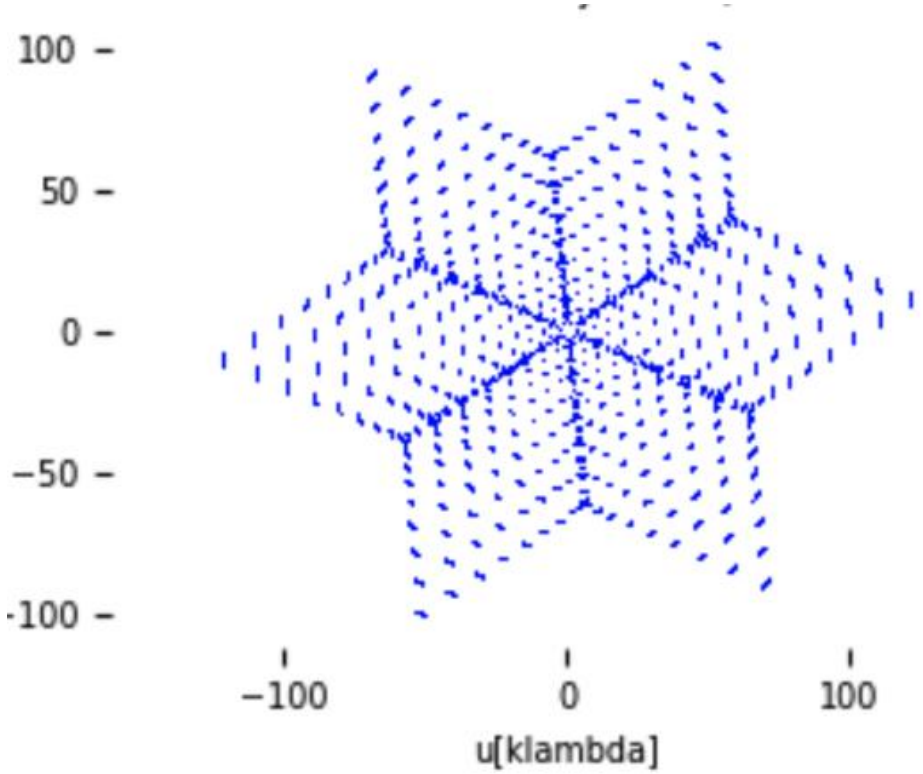
Dirty image: $I_D(\mathbf{x}) = \Re\left[\sum_{k=1}^{M} w_k V_k \exp\left(i2\pi \mathbf{u}_k \mathbf{x}\right)\right]$

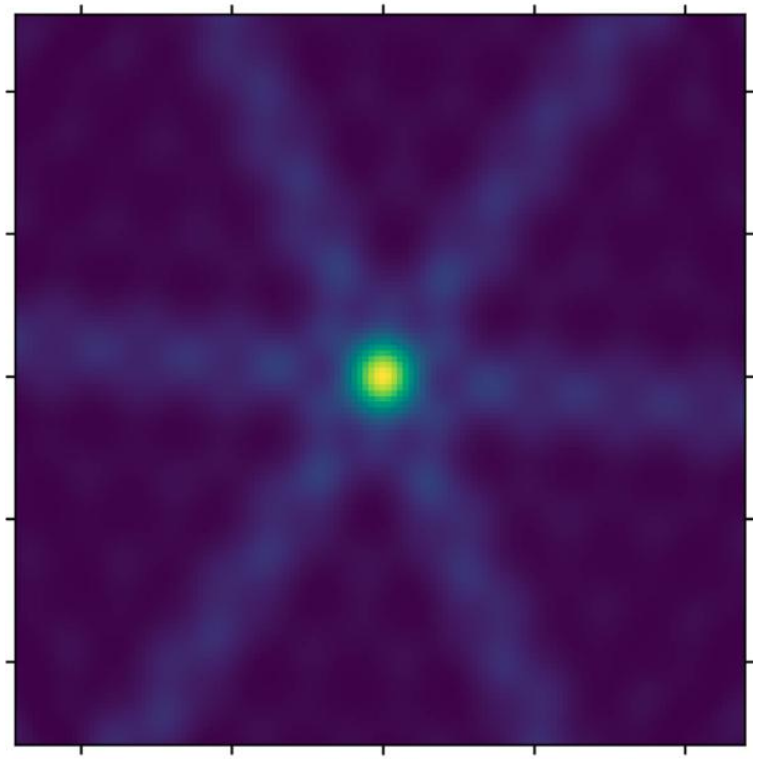
$\mathbf{u}_k = (u_k, v_k)$ and $\mathbf{x} = (x, y)$.

Dirty beam: $B(\mathbf{x}) = \Re\left[\sum_{k=1}^{M} w_k \exp\left(i2\pi \mathbf{u}_k \mathbf{x}\right)\right]$

UNIVERSITY OF CAMBRIDGE

Dirty Beam

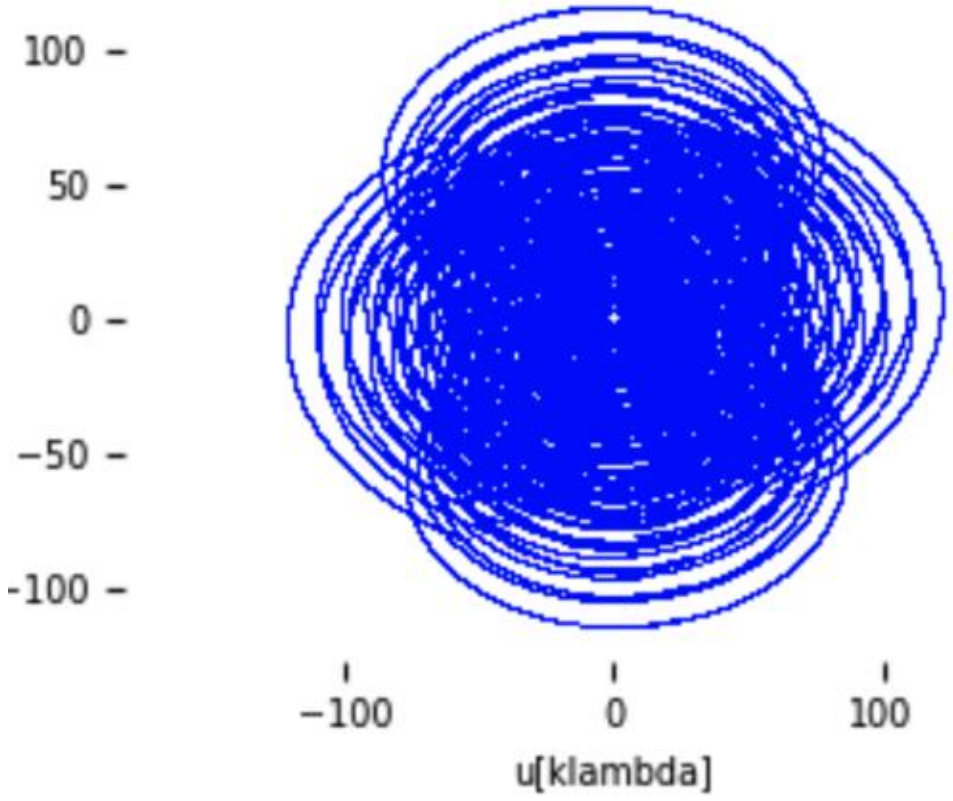




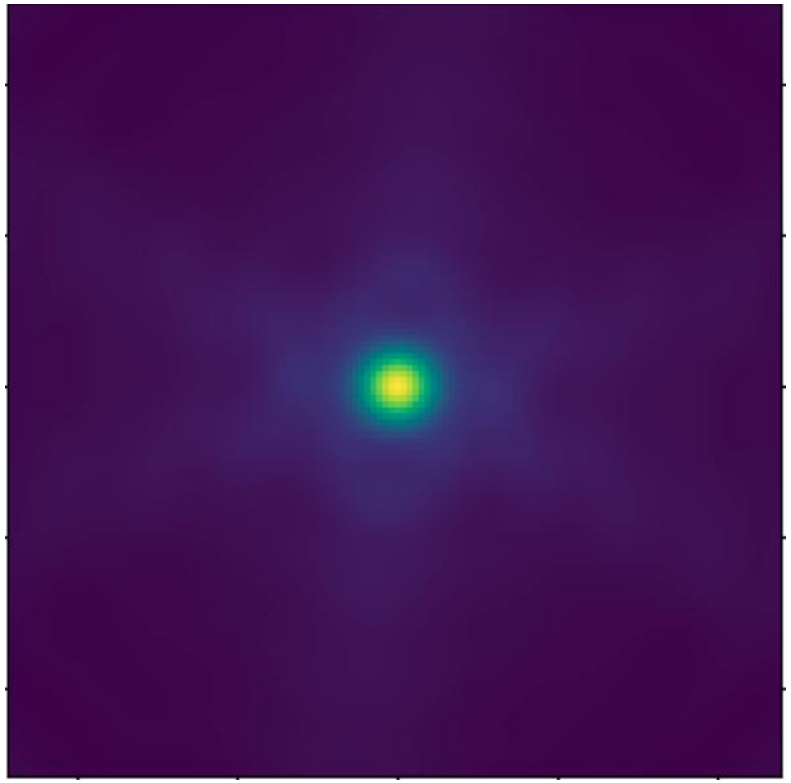
$S(u,v)$ $B(l,m)$

UNIVERSITY OF CAMBRIDGE

Dirty Beam



$S(u,v)$

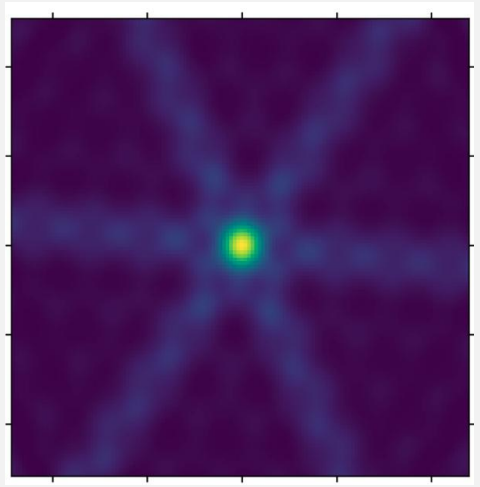


$B(l,m)$

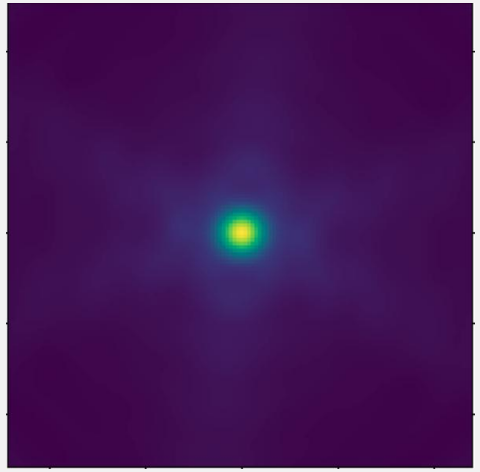
UNIVERSITY OF CAMBRIDGE

Dirty Beam

Which beam do you prefer? Why?



A



B

UNIVERSITY OF CAMBRIDGE

Dirty Beam

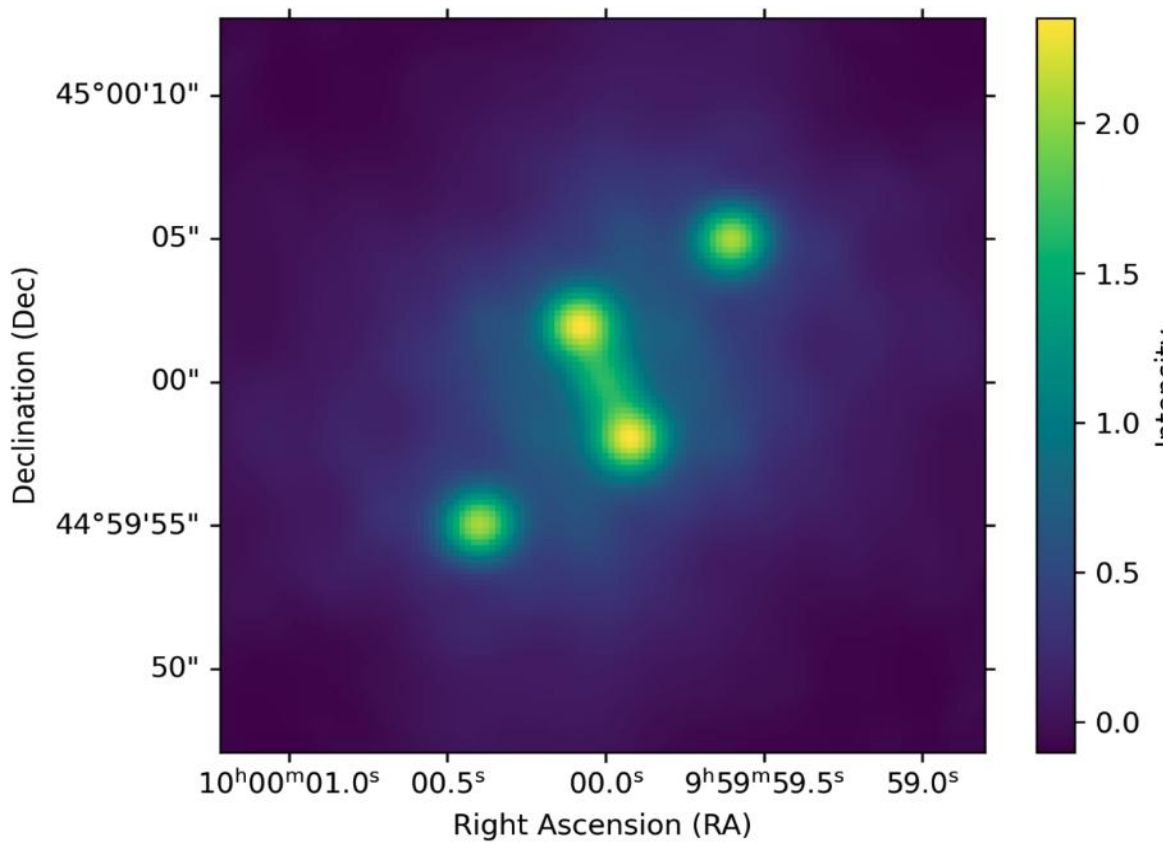
Therefore the layout of the antennas

$$S(u,v) = \sum_{k=1}^{M} \delta\left(u-u_k, v-v_k\right)$$

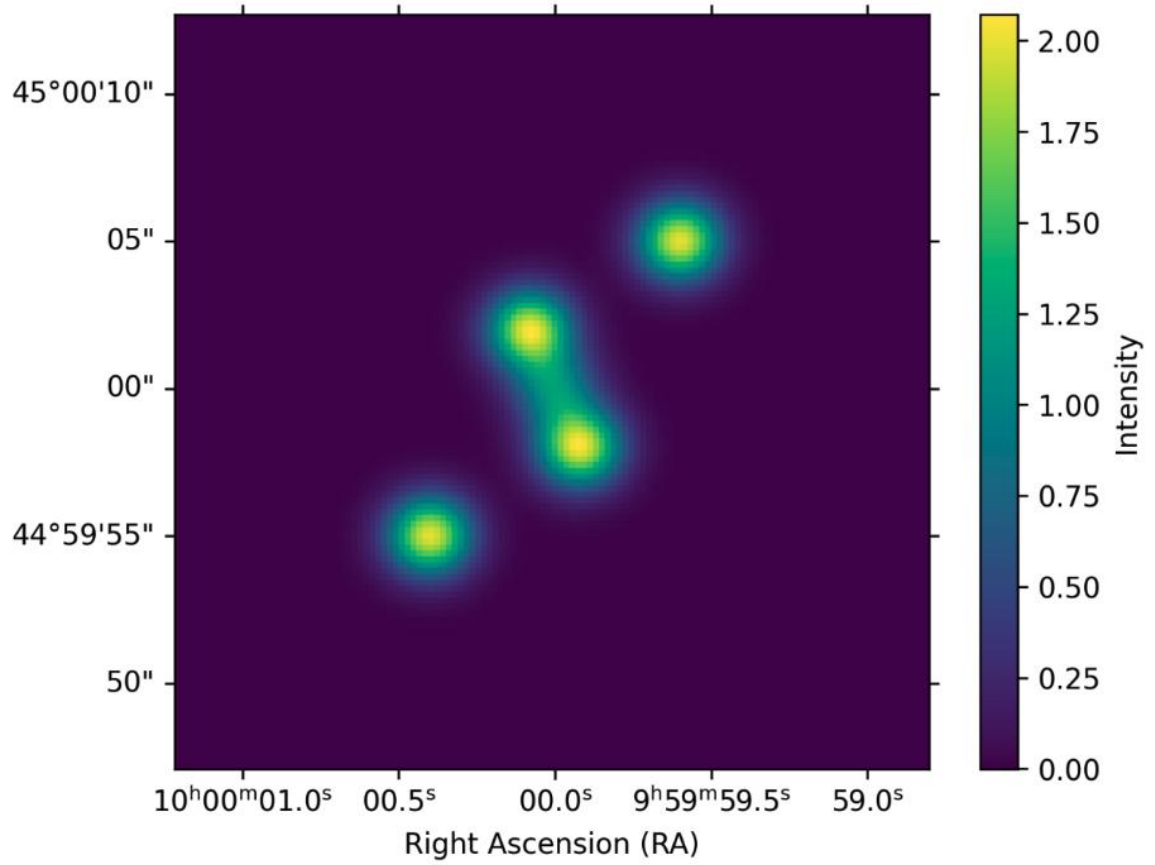
needs to be designed properly designed!

UNIVERSITY OF CAMBRIDGE

Dirty image VS Sky image



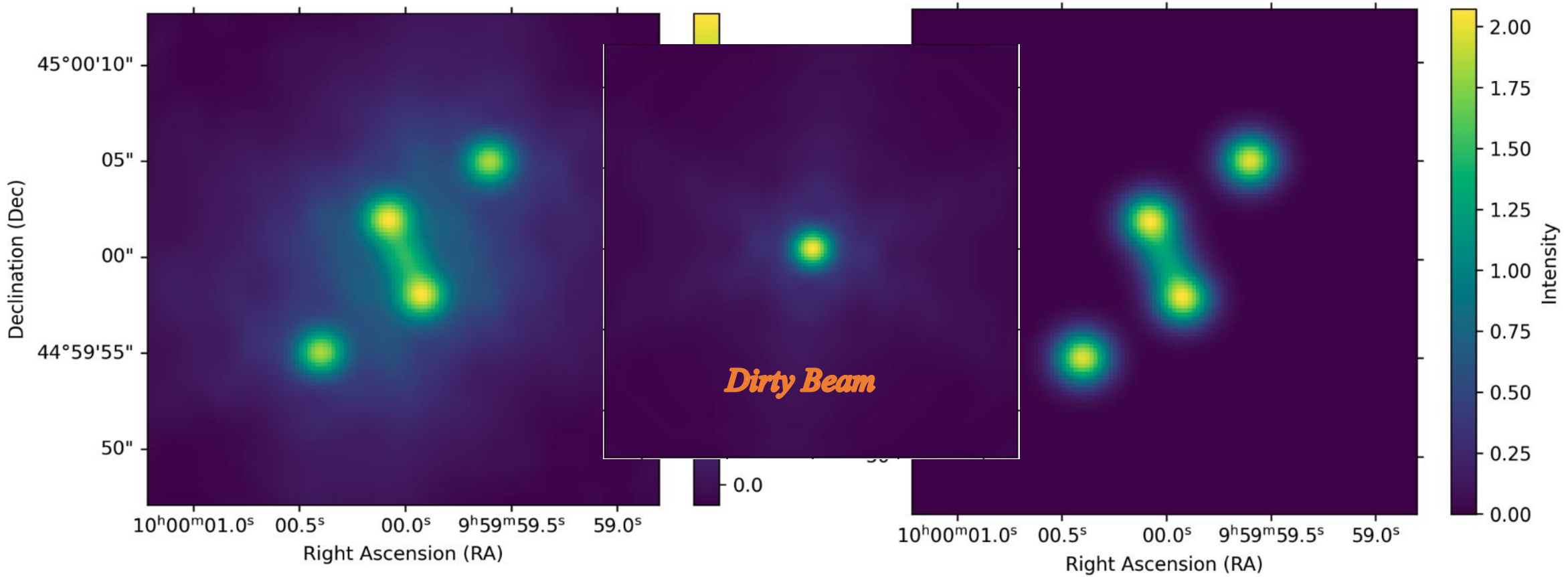
$I(l,m)*B(l,m)$



$I(l,m)$

UNIVERSITY OF CAMBRIDGE

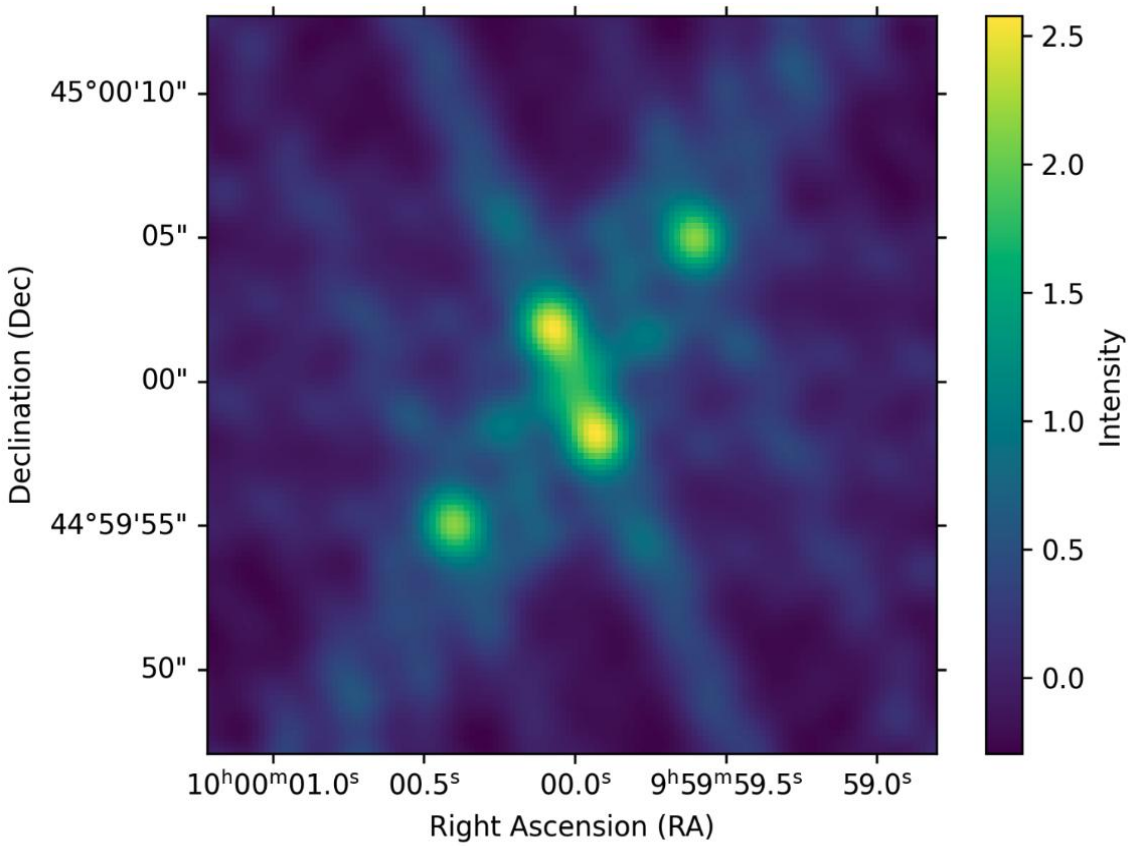
Dirty image VS Sky image



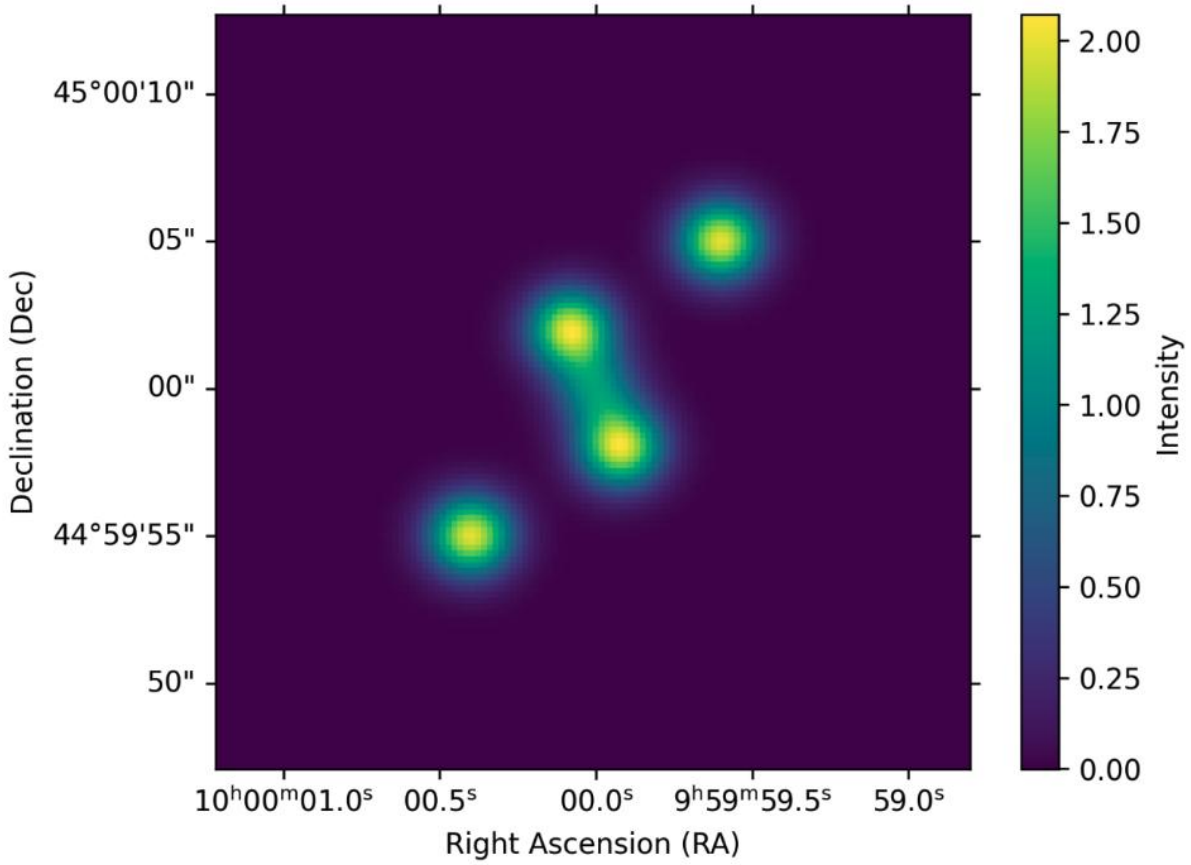
$I(l,m)*B(l,m)$ $I(l,m)$

UNIVERSITY OF CAMBRIDGE

Dirty image VS Sky image



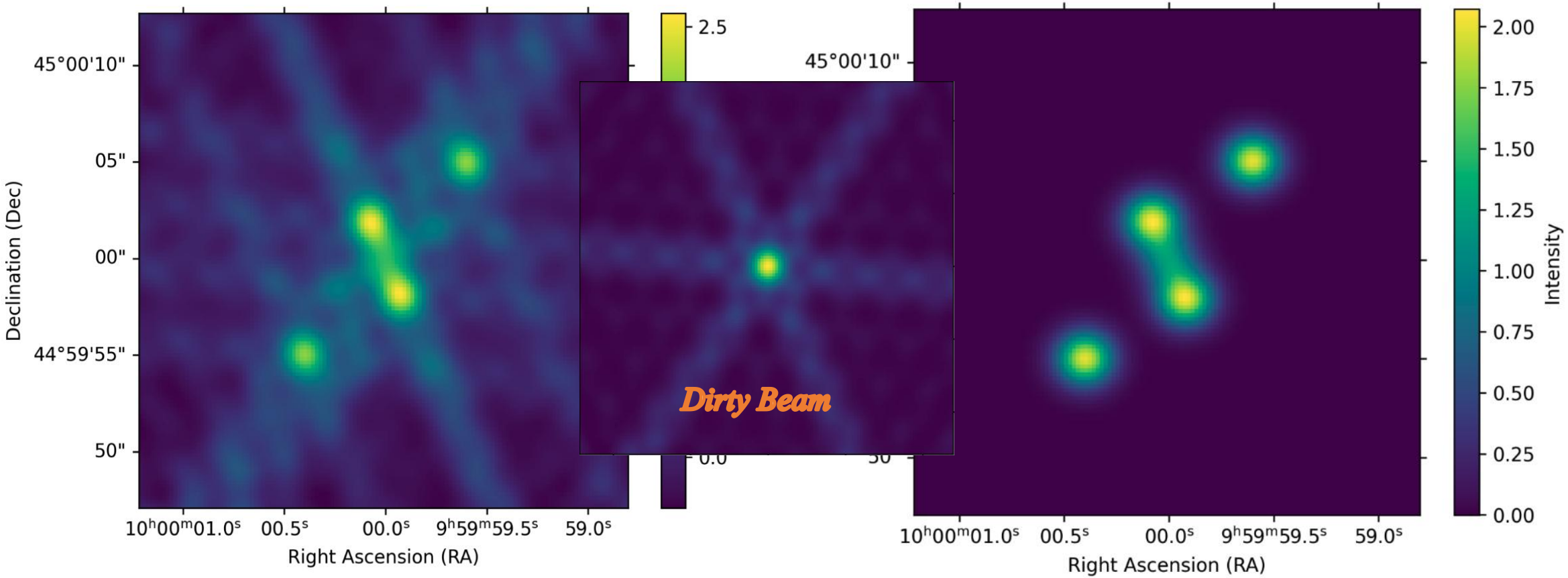
$I(l,m)*B(l,m)$



$I(l,m)$

UNIVERSITY OF CAMBRIDGE

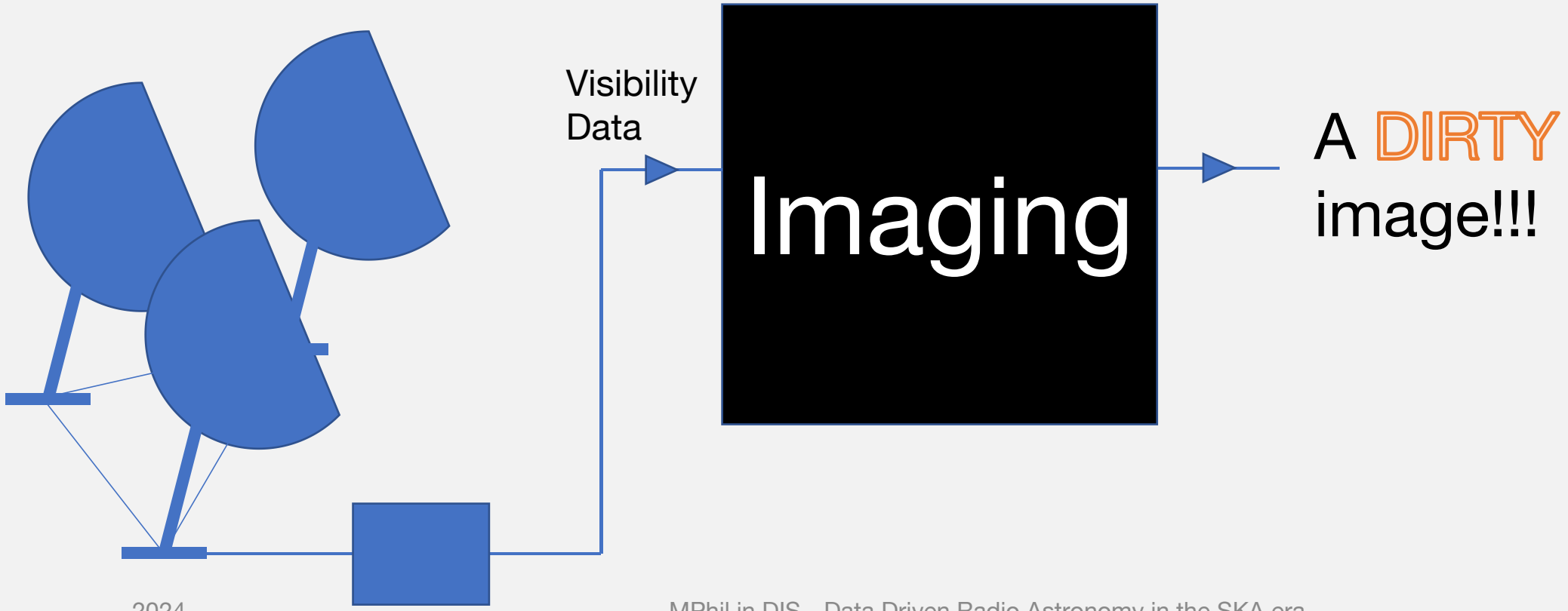
Dirty image VS Sky image



$I(l,m)*B(l,m)$ $I(l,m)$

UNIVERSITY OF CAMBRIDGE

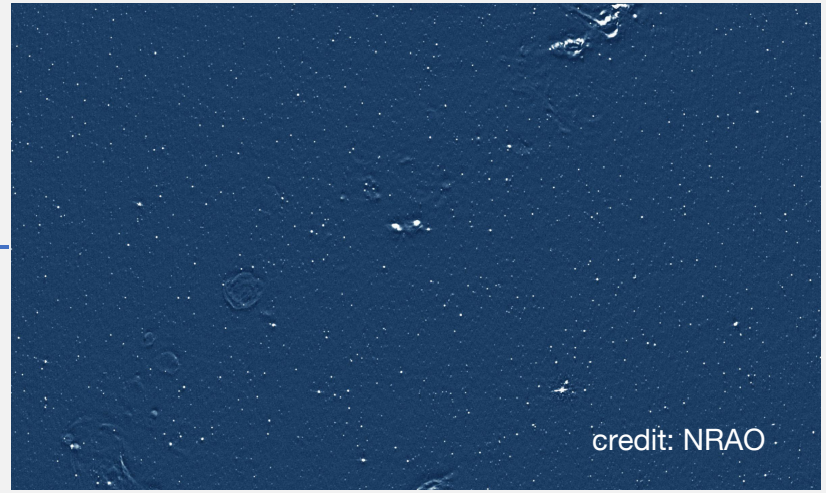
Today you have learned:



UNIVERSITY OF CAMBRIDGE

Deconvolution

Observed Visibility data → Inverse Fourier Transform → A DIRTY image!!!



→ A Deconvolved image!!!

UNIVERSITY OF CAMBRIDGE

Högbom CLEAN

Jan Arvid Högbom (born 3 October 1929) is a Swedish radio astronomer and astrophysicist. Högbom obtained his PhD in 1959 from the University of Cambridge with Martin Ryle.

1. Initialisation: Let D(x, y) represent the dirty image, where (x, y) are the spatial coordinates, and set the residual image R(x, y) equal to the dirty image: R(x, y) = D(x, y).
2. Identify Peaks: Search for the location (x_p, y_p) of the brightest pixel (peak) in the residual image
3. Extract Component: Subtract a (scaled) Gaussian component G(x, y) from the residual image at the location of the identified peak: R(x, y)<- R(x, y) - aG(x_p, y_p), a is the scaled peak intensity at (x_p, y_p)
4. Update Clean Component List: Add the extracted component to the clean component list
5. Iterate: Repeat steps 2-4 iteratively until one or more termination criteria are met.
6. Final Image Reconstruction: I(x, y) is reconstructed by combining the clean component list with the residual image

UNIVERSITY OF CAMBRIDGE

Högbom CLEAN

Jan Arvid Högbom (born 3 October 1929) is a Swedish radio astronomer and astrophysicist. Högbom obtained his PhD in 1959 from the University of Cambridge with Martin Ryle.

$$I_D(x,y) = I(x,y) * B(x,y)$$

$$= [I_1(x,y) + I_2(x,y) + \ldots + I_n(x,y)] * B(x,y)$$

$$= \sum_{\mathrm{i}=1}^{\mathrm{M}} I_i(x,y) * B(x,y)$$

$$R(x,y) \equiv I_D(x,y) \; - \; \alpha I_{CLEAN}(x,y) * B(x,y)$$

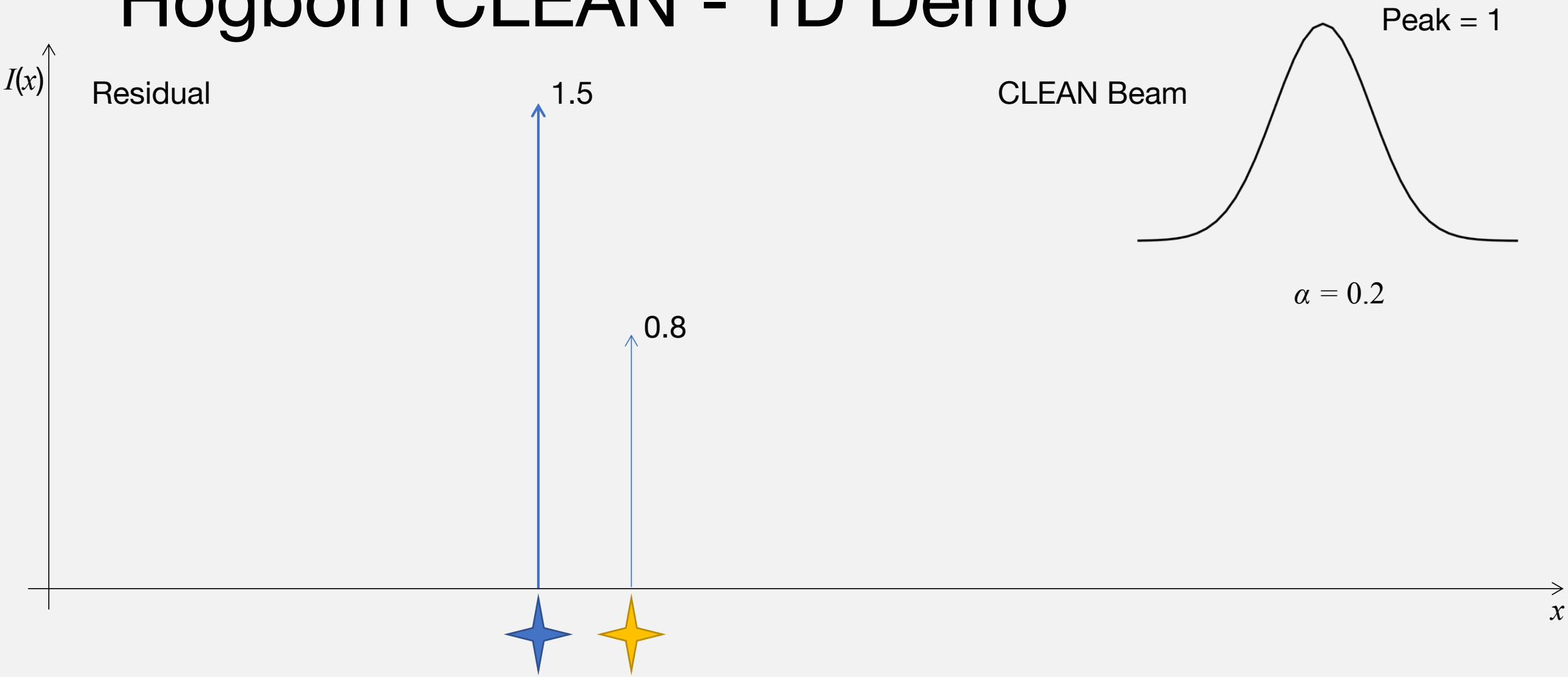
$$I_{reconstructed}(x,y) \; \equiv \; R(x,y) \; + \; I_{CLEAN}(x,y)$$

We usually don't use dirty beam B(x,y), but use a Gaussian for simplicity, which is called "CLEAN BEAM"

α is a scaling factor, so it takes several loops to subtract the highest intensity - do you know why?

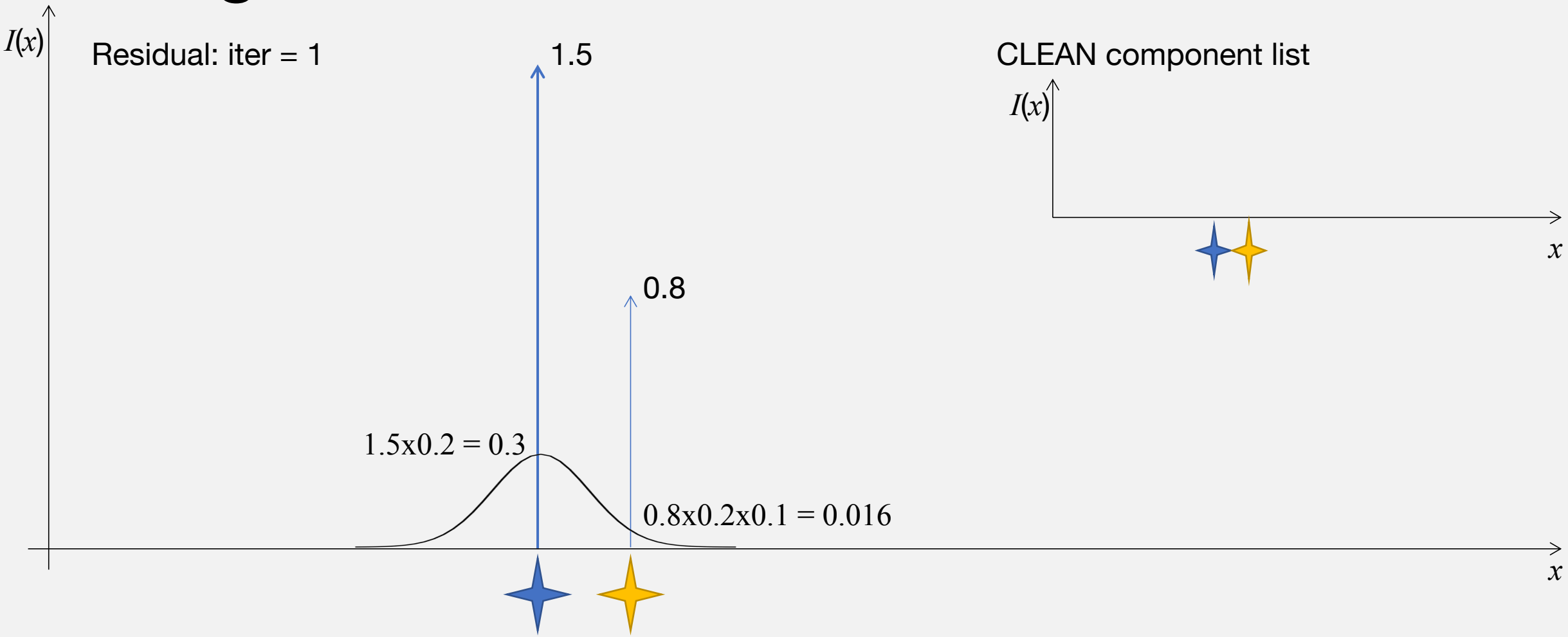
UNIVERSITY OF CAMBRIDGE

Högbom CLEAN - 1D Demo



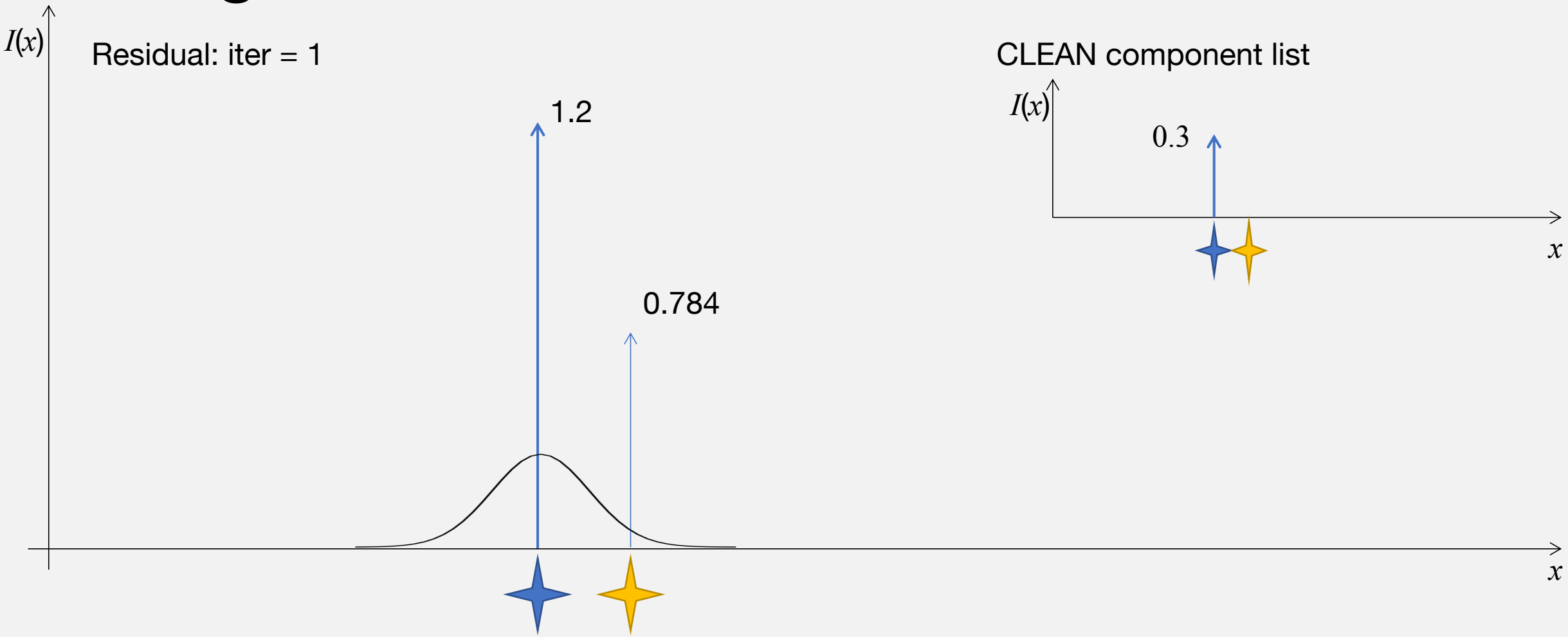
UNIVERSITY OF CAMBRIDGE

Högbom CLEAN - 1D Demo



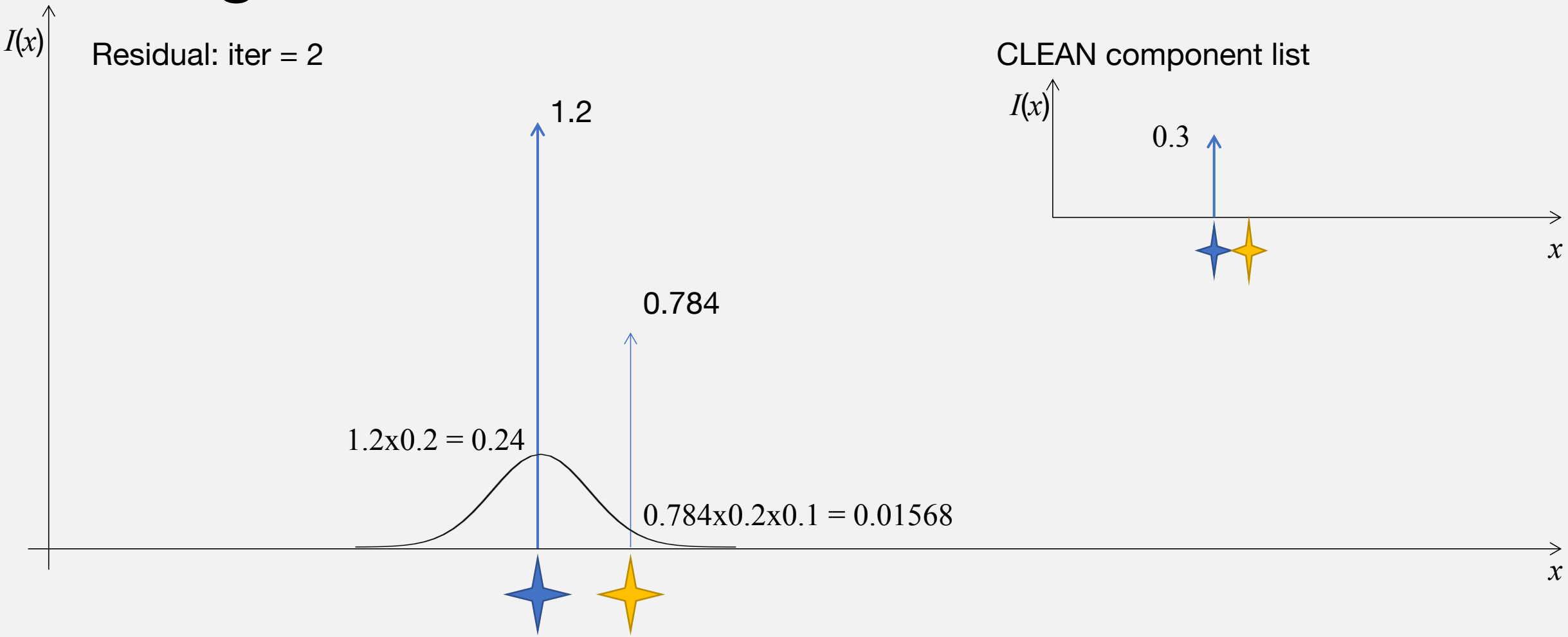
UNIVERSITY OF CAMBRIDGE

Högbom CLEAN - 1D Demo



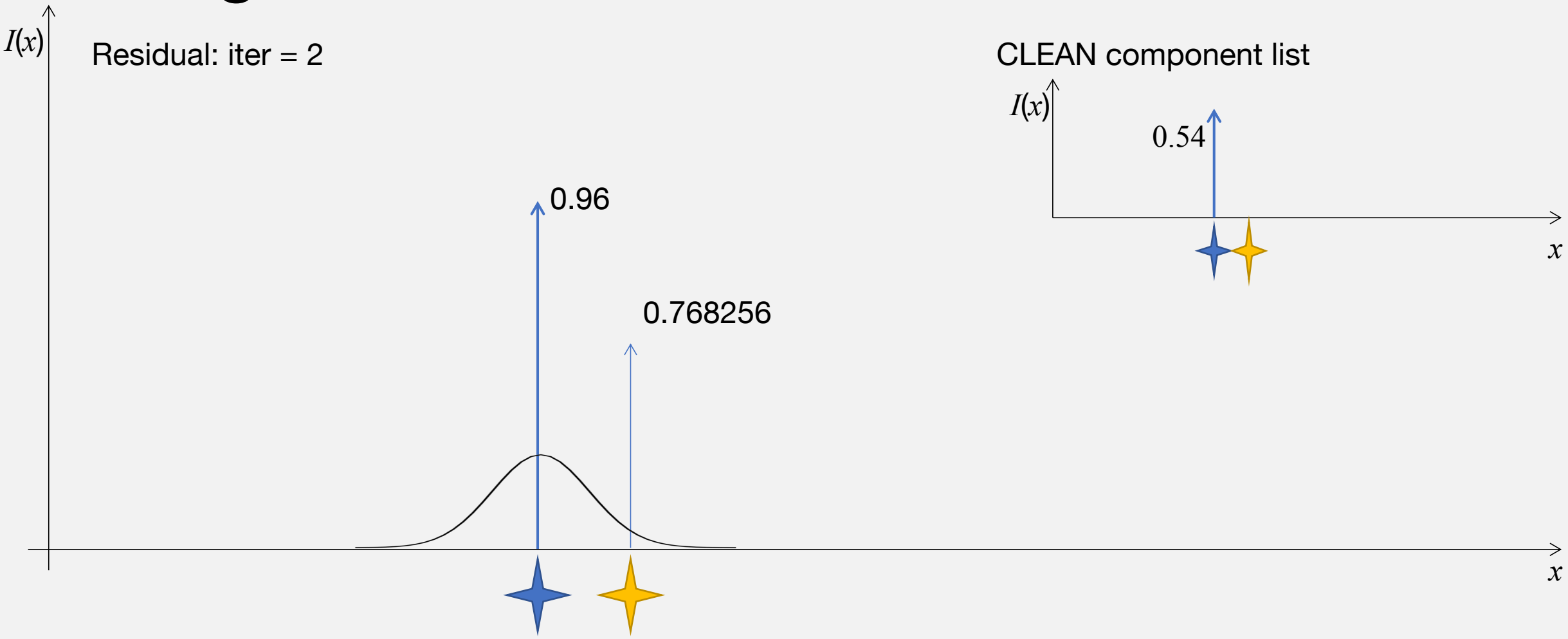
UNIVERSITY OF CAMBRIDGE

Högbom CLEAN - 1D Demo



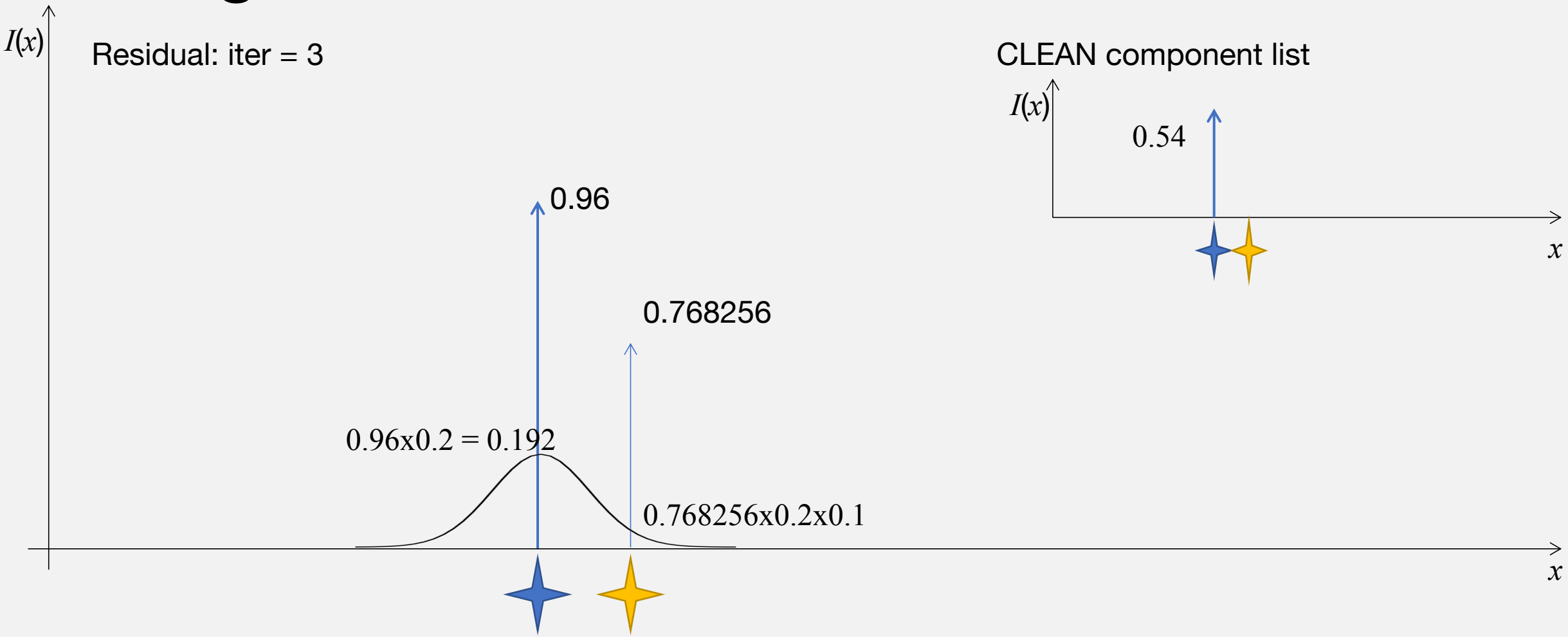
UNIVERSITY OF CAMBRIDGE

Högbom CLEAN - 1D Demo



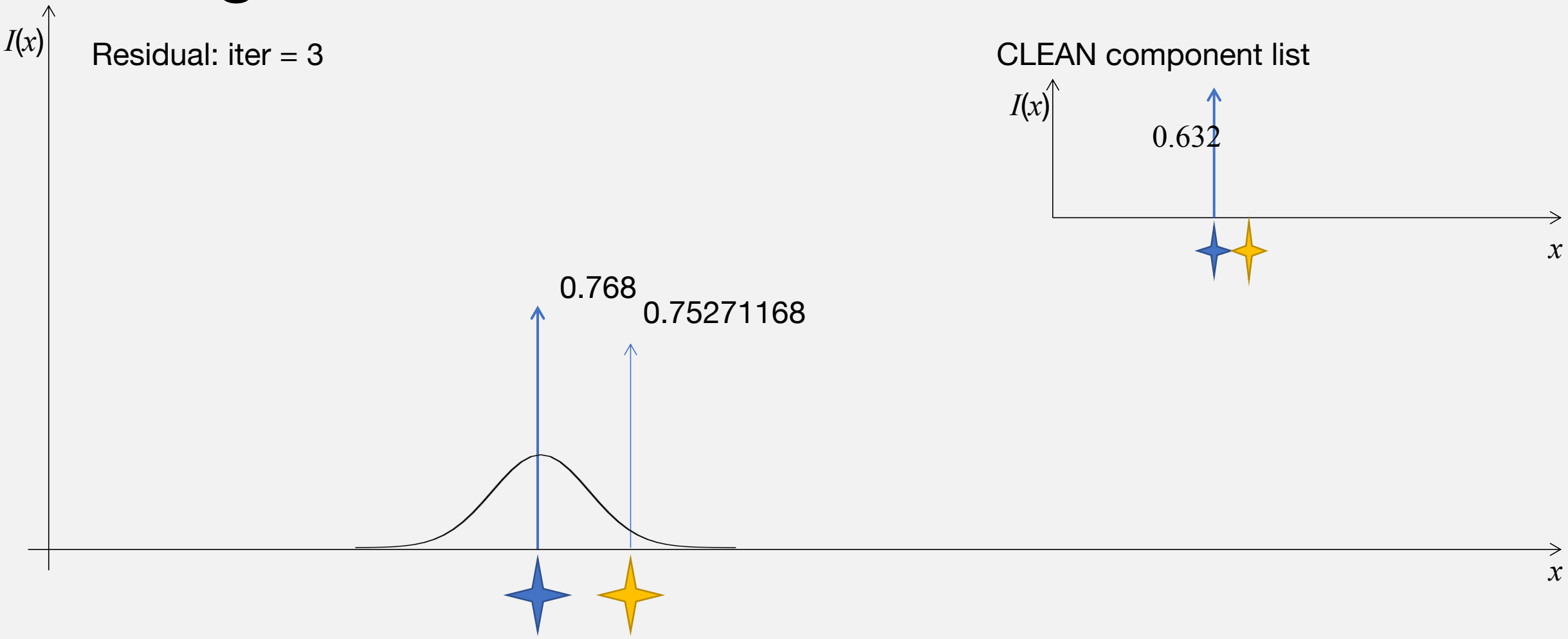
UNIVERSITY OF CAMBRIDGE

Högbom CLEAN - 1D Demo



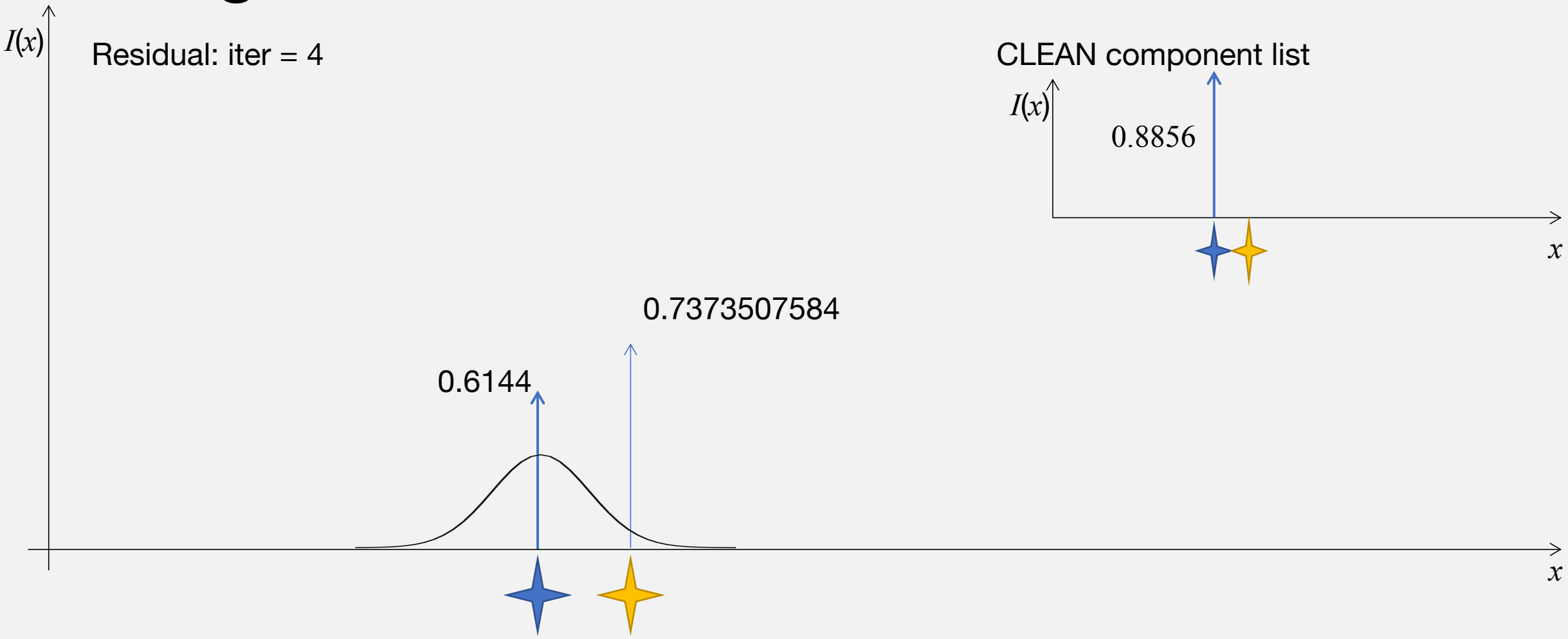
UNIVERSITY OF CAMBRIDGE

Högbom CLEAN - 1D Demo



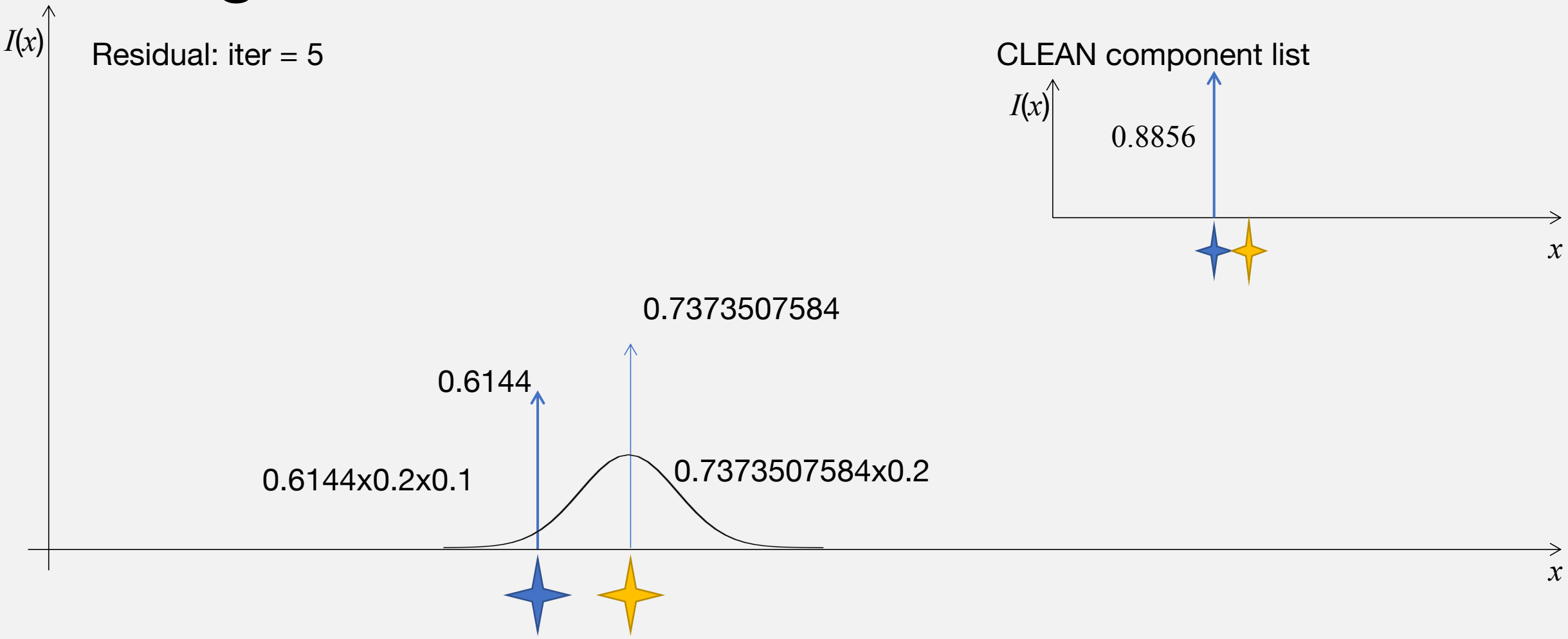
UNIVERSITY OF CAMBRIDGE

Högbom CLEAN - 1D Demo



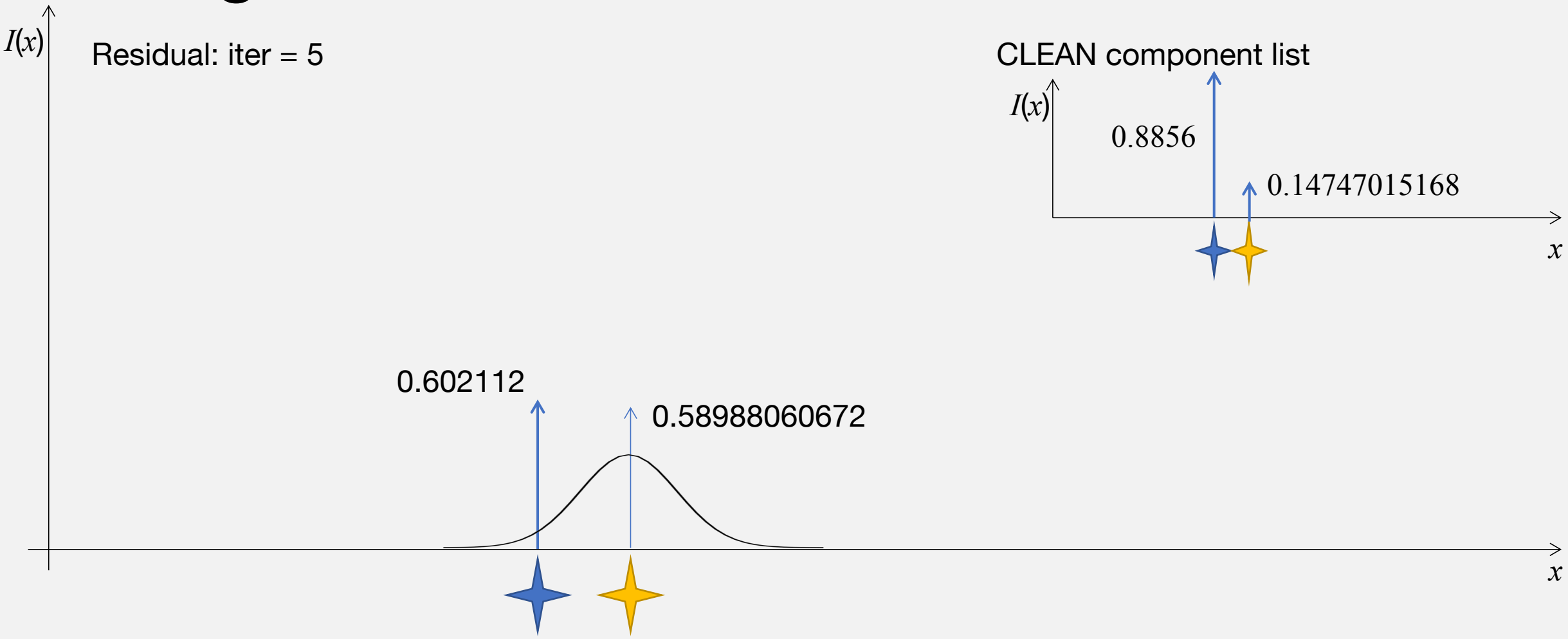
UNIVERSITY OF CAMBRIDGE

Högbom CLEAN - 1D Demo



UNIVERSITY OF CAMBRIDGE

Högbom CLEAN - 1D Demo

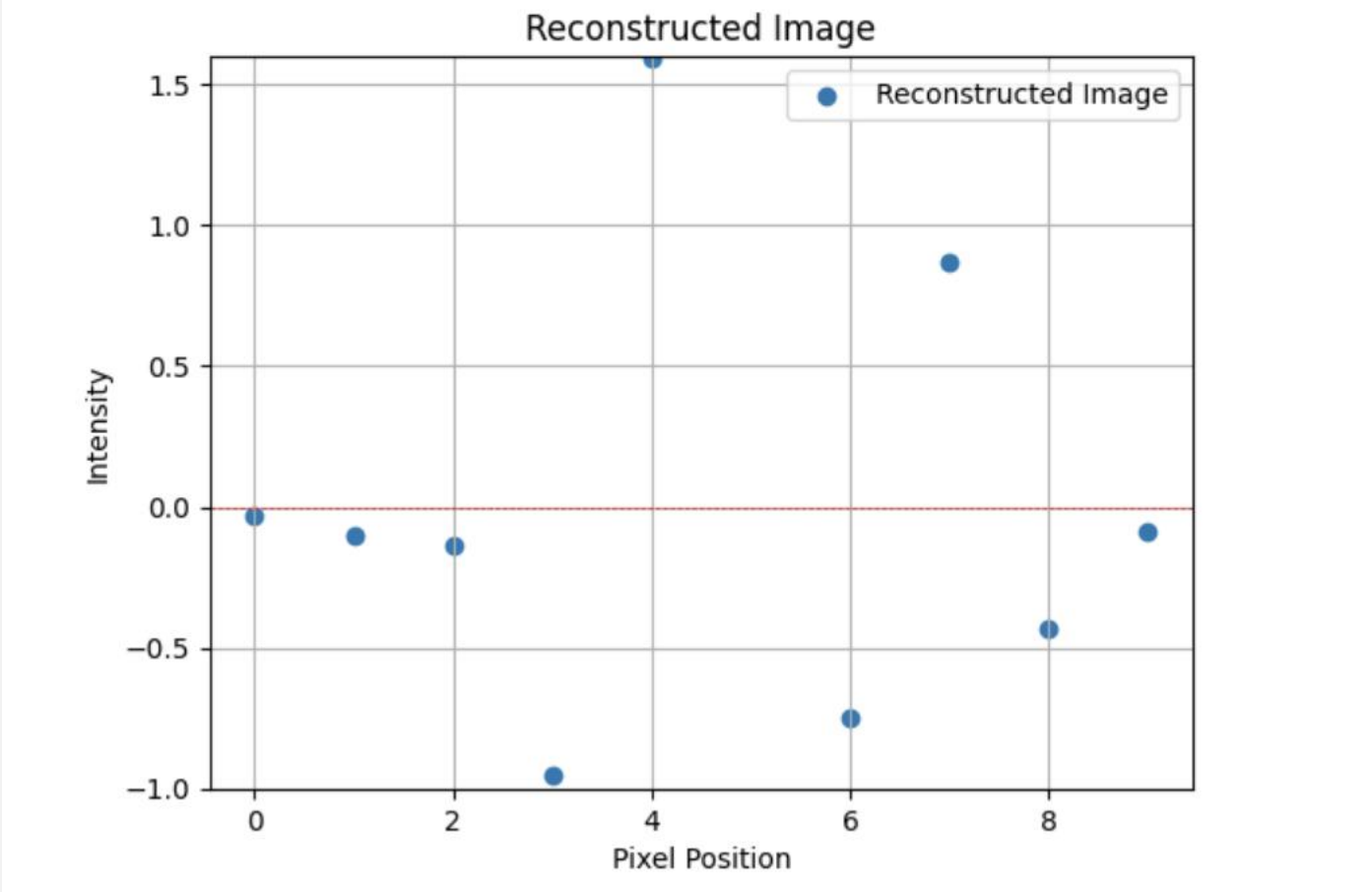


UNIVERSITY OF CAMBRIDGE

Coding Time

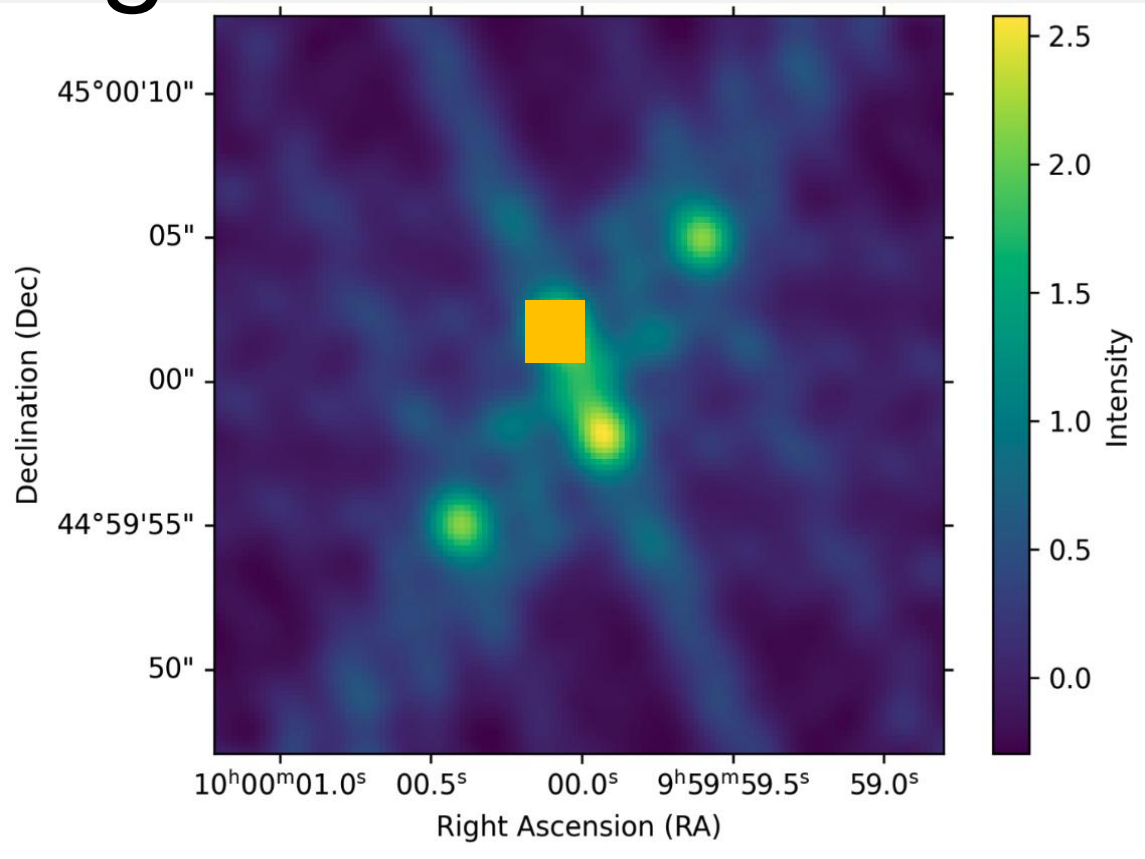
Given the 1-D Hogbom CLEAN algorithm example

Do complete the codes

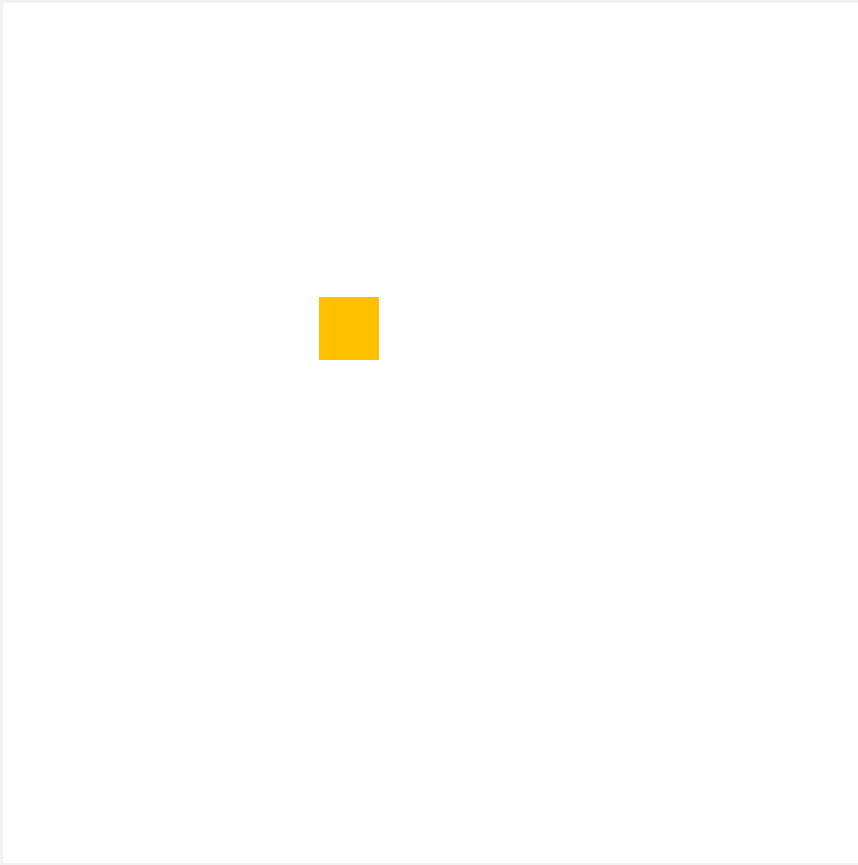


UNIVERSITY OF CAMBRIDGE

Högbom CLEAN



CLEAN Component list

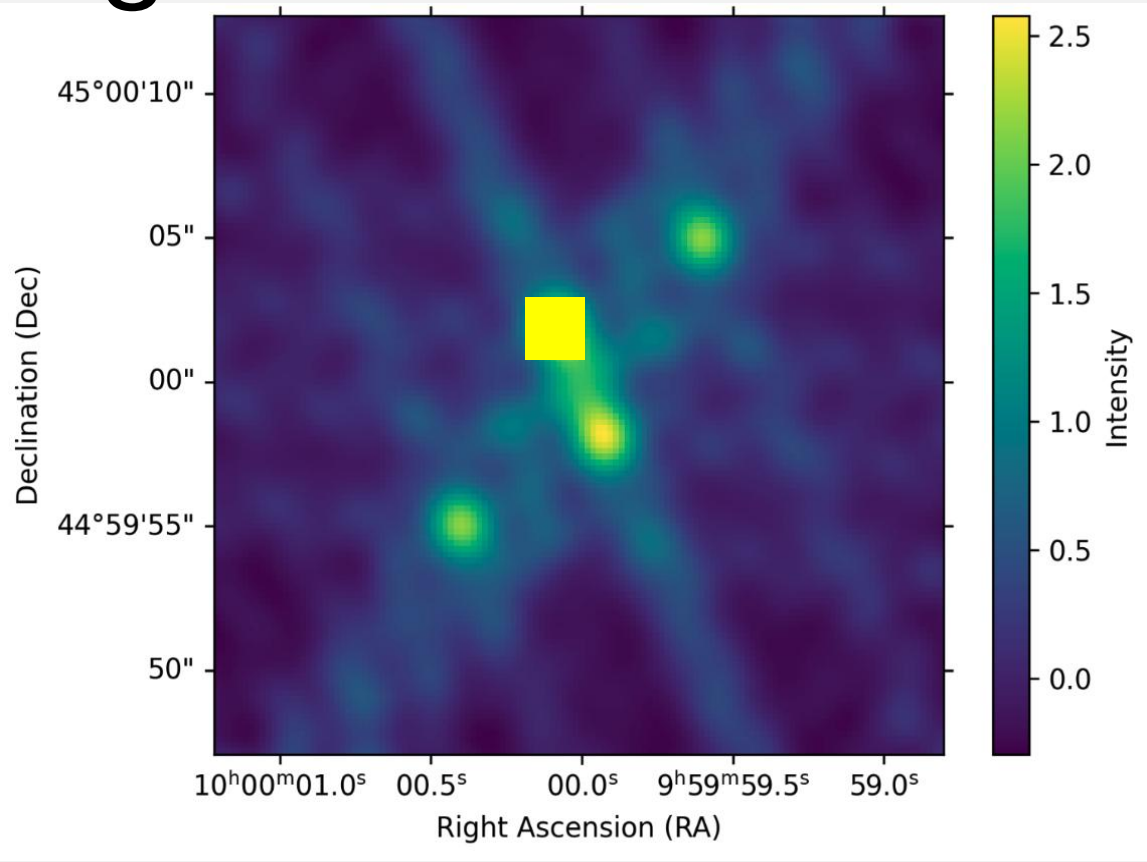


2. Identify Peak

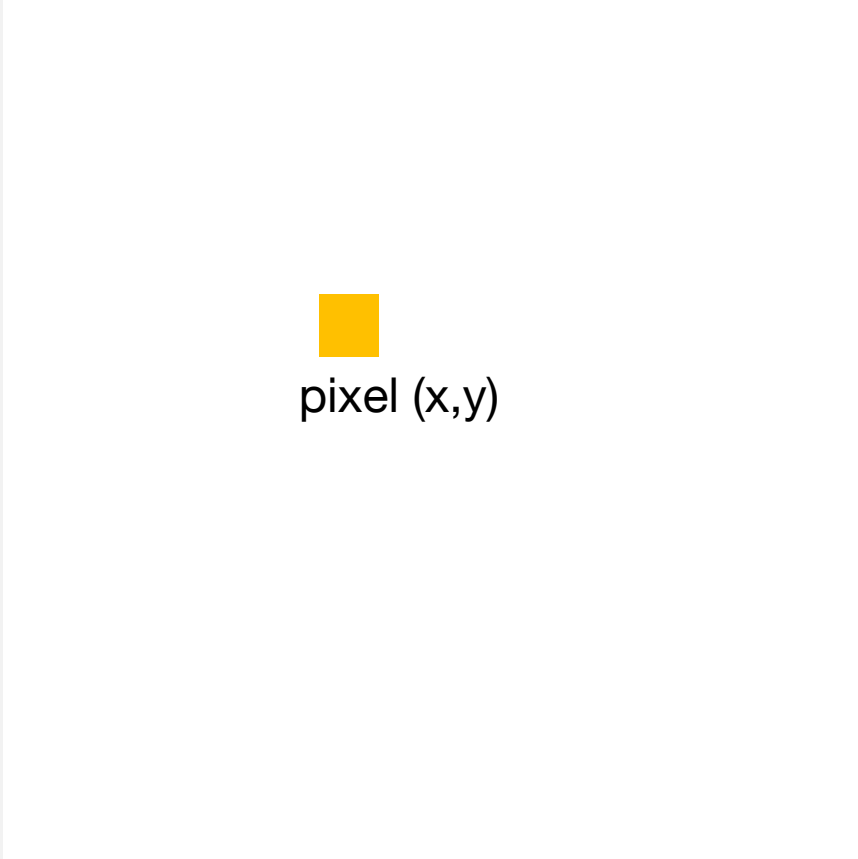
Find the brightes pixel !!!!

UNIVERSITY OF CAMBRIDGE

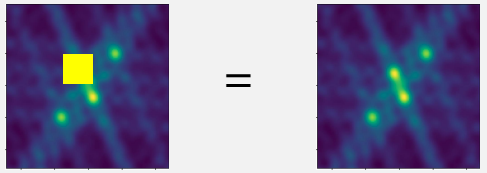
Högbom CLEAN



CLEAN Component list



3. Subtract Model

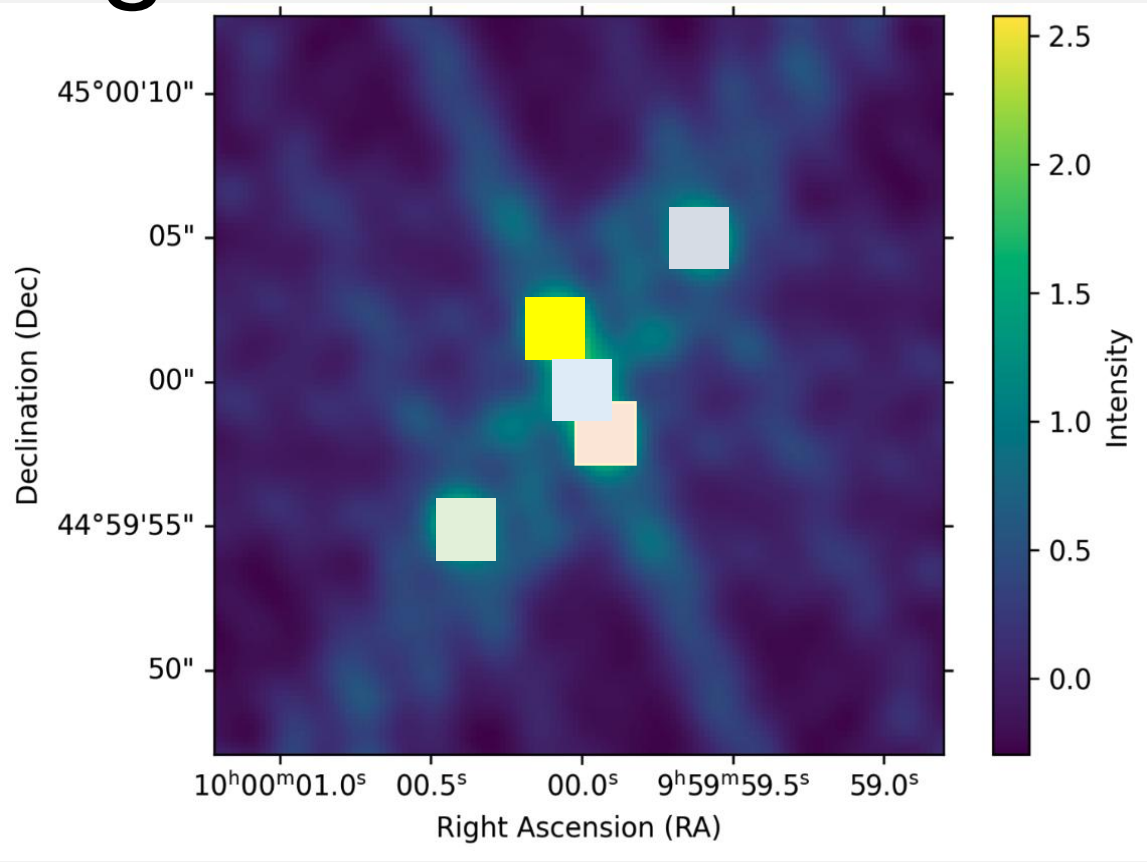


=

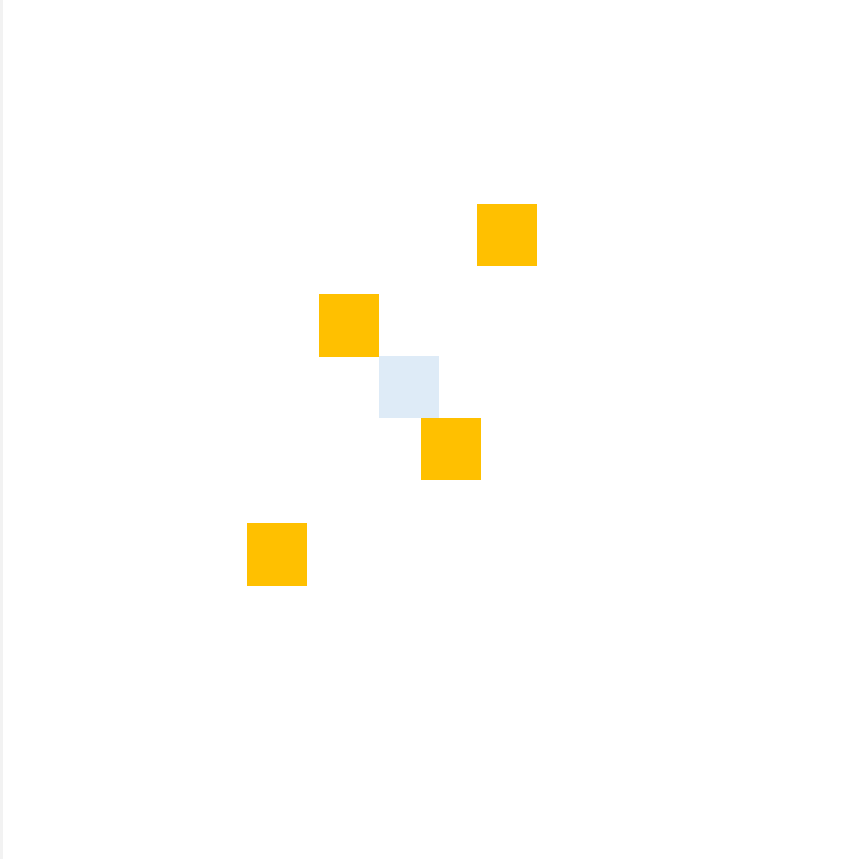
- Gaussian x scaling factor x intensity at pixel (x,y)

UNIVERSITY OF CAMBRIDGE

Högbom CLEAN



CLEAN Component list

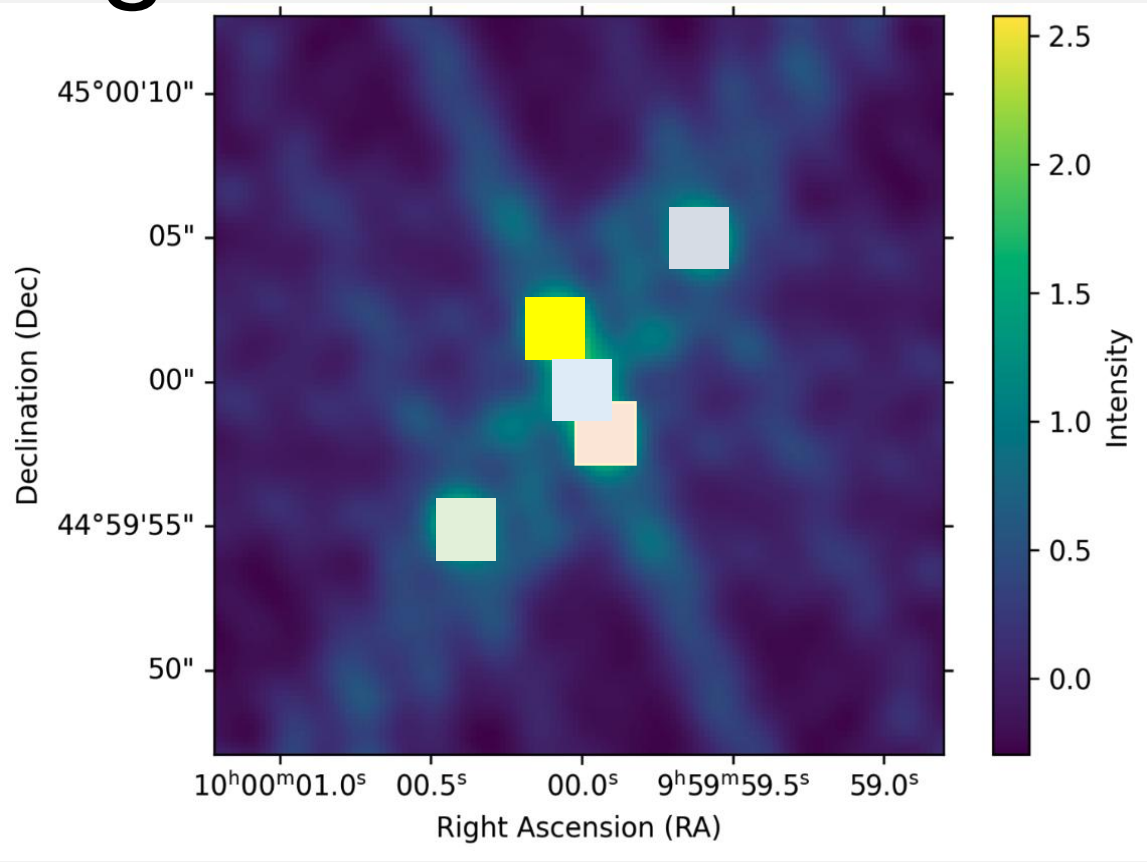


Repeat 2 and 3

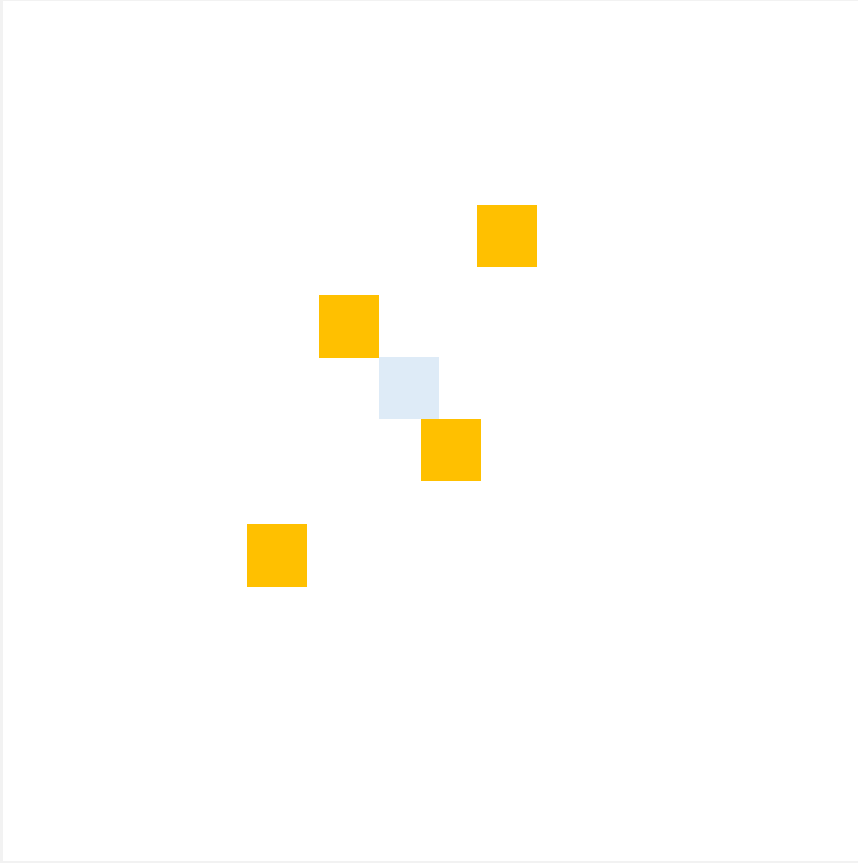
Residual image

UNIVERSITY OF CAMBRIDGE

Högbom CLEAN



CLEAN Component list



Iteration STOPs when

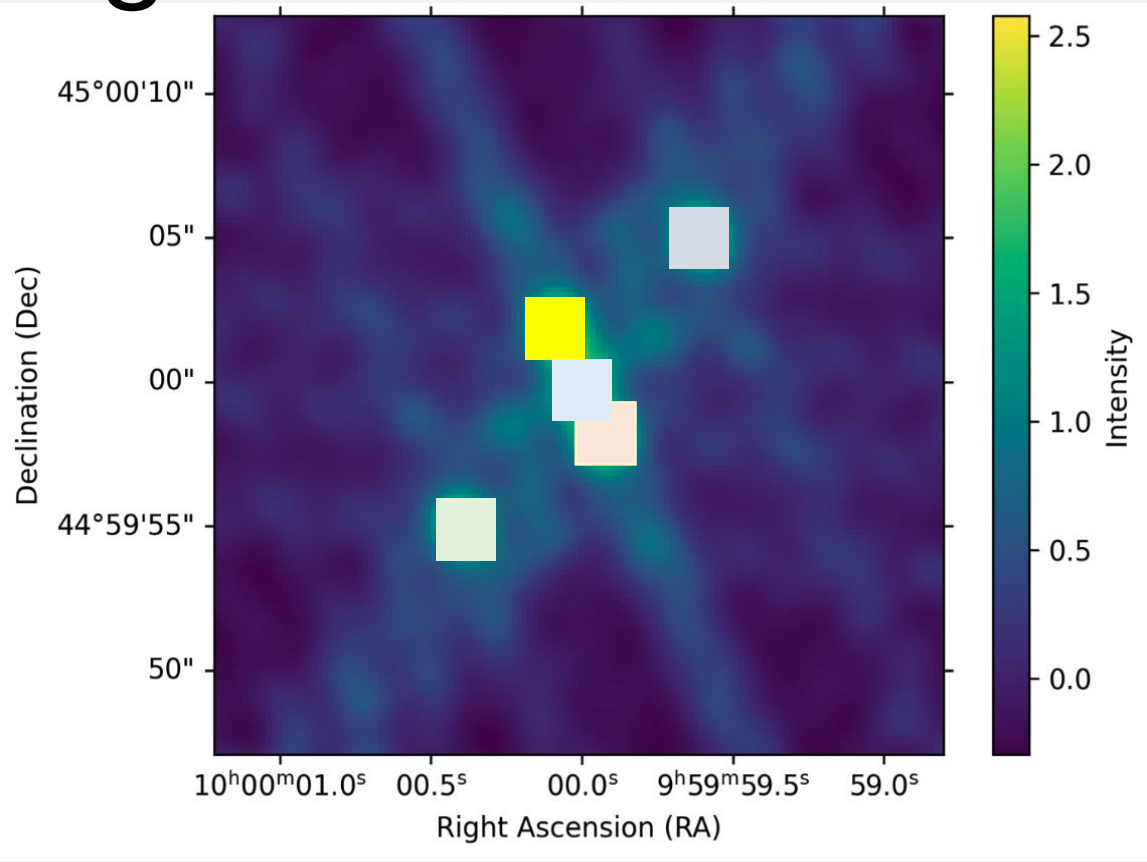
peak (Residual image) < given value or

≈

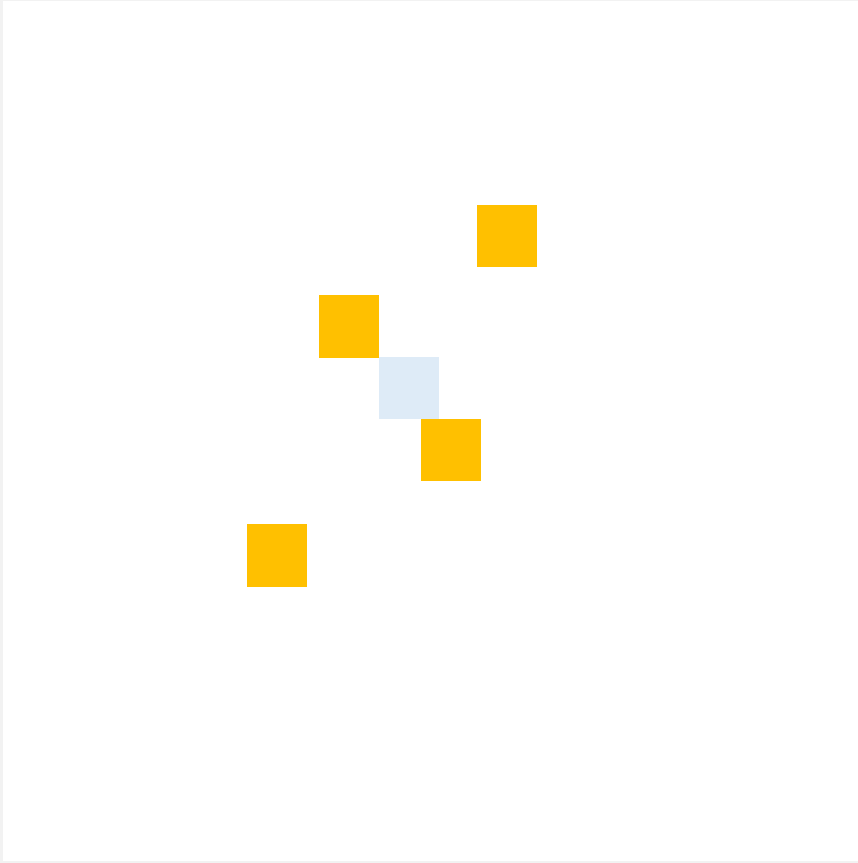
iteration cycle > given cycle number

UNIVERSITY OF CAMBRIDGE

Högbom CLEAN



CLEAN Component list



Final image =

Residual image + CLEAN component list

UNIVERSITY OF CAMBRIDGE

Coding Time

If my miniproject got selected, you will be able to try to write a full CLEAN algorithm in python :D

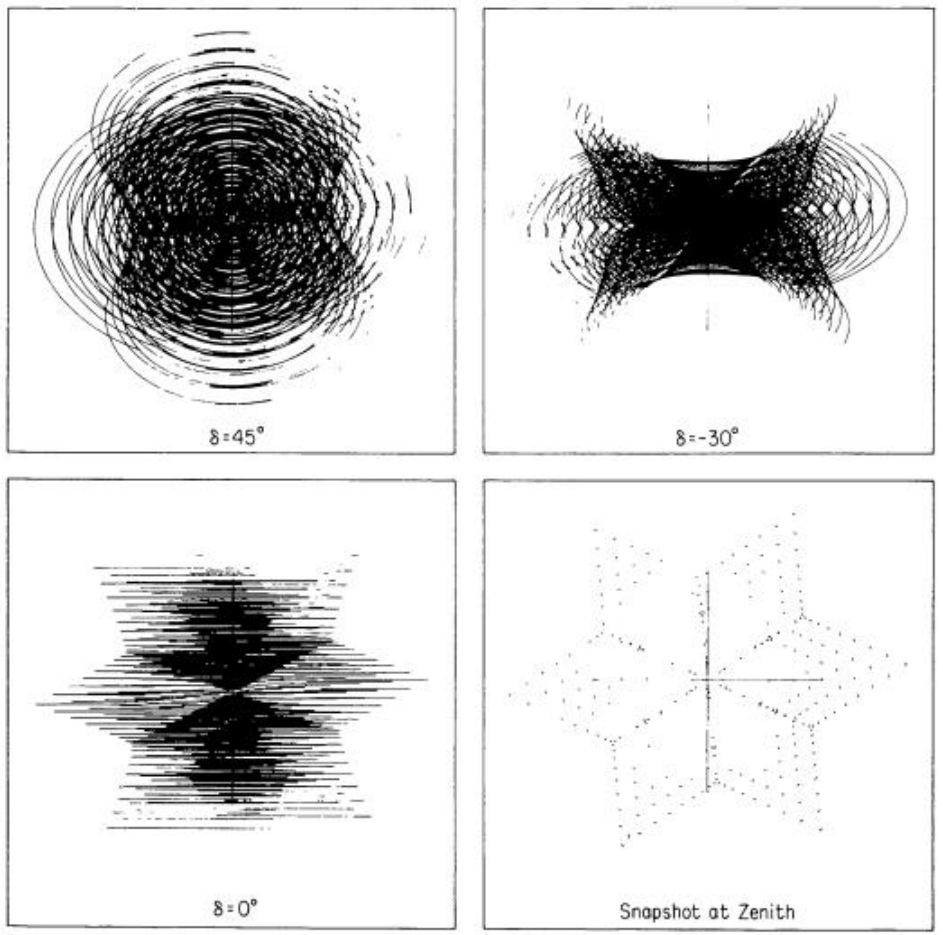
UNIVERSITY OF CAMBRIDGE

Reading

- Fourier Synthesis Imaging by Prof. Dale E. Gary: https://web.njit.edu/~gary/728/Lecture6.html
- Synthesis Imaging in Radio Astronomy: A Collection of Lectures from the Third NRAO Synthesis Imaging Summer School: https://www.aspbooks.org/a/volumes/table_of_contents/?book_id=118
- Imaging and Deconvolution by David J. Wilner: https://science.nrao.edu/science/meetings/2018/16th-synthesis-imaging-workshop/talks/Wilner_Imaging.pdf

UV-coverage

https://www.aspbooks.org/publications/6/83.pdf

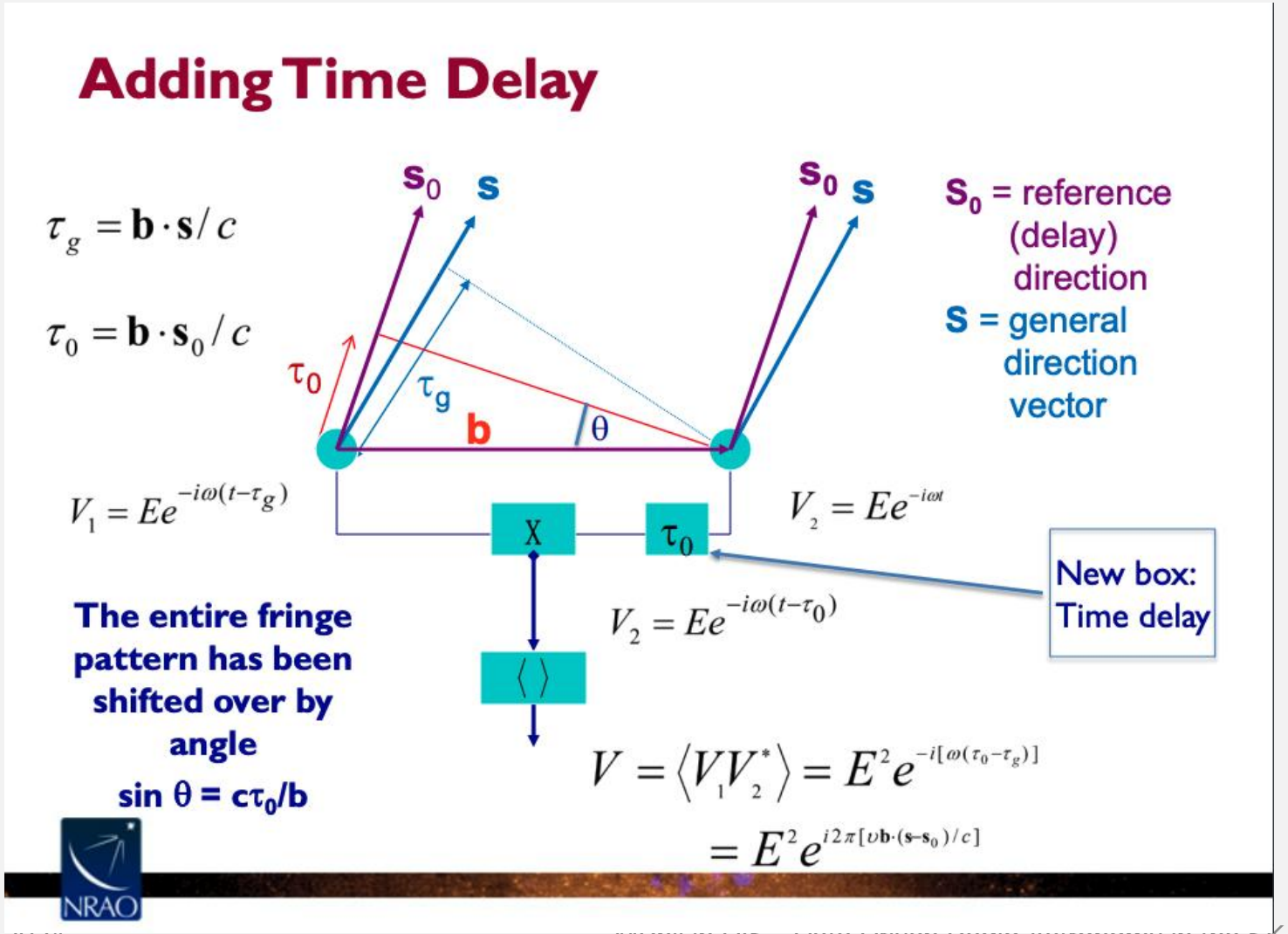


(b)

Figure 2–14. (a) The configuration of the 27 antennas of the VLA. **(b)** The transfer functions for four declinations with observing durations of $\pm 4^h$ for $\delta = 0°$ and $45°$, $\pm 3^h$ for $\delta = -30°$, and $\pm 5^m$ for the snapshot. [From Napier, Thompson and Ekers (© 1983 IEEE).]

UNIVERSITY OF CAMBRIDGE

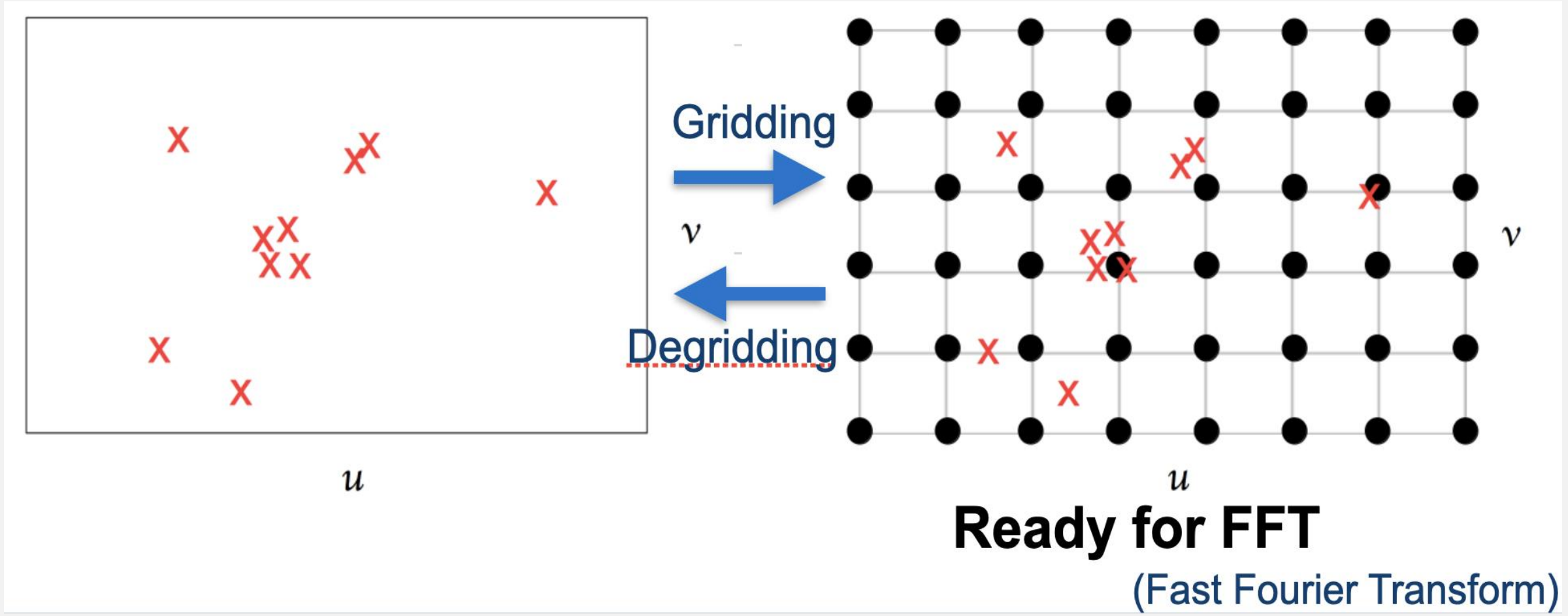
Correlation



https://science.nrao.edu/science/meetings/2018/16th-synthesis-imaging-workshop/talks/Perley_Advanced.pdf

UNIVERSITY OF CAMBRIDGE

GRIDDING



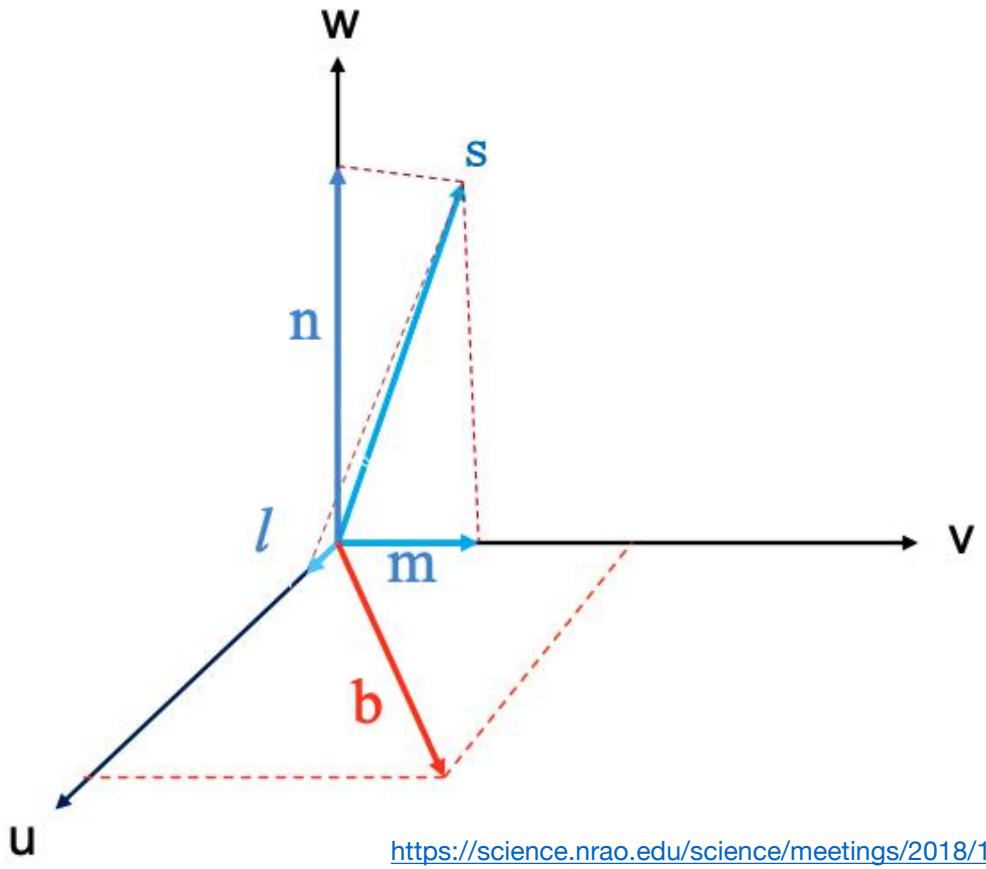
Direction Cosines – describing the source

The unit direction vector **s** is defined by its projections (l,m,n) on the (u,v,w) axes. These components are called the **Direction Cosines**.

$$l = \cos(\alpha)$$

$$m = \cos(\beta)$$

$$n = \cos(\theta) = \sqrt{1 - l^2 - m^2}$$



https://science.nrao.edu/science/meetings/2018/16th-synthesis-imaging-workshop/talks/Perley_Geometry_new.pdf

The angles, α, β, and θ are between the direction vector and the three axes.

UNIVERSITY OF CAMBRIDGE

Högbom CLEAN

Jan Arvid Högbom (born 3 October 1929) is a Swedish radio astronomer and astrophysicist. Högbom obtained his PhD in 1959 from the University of Cambridge with Martin Ryle.

1. Initialisation: Start with an empty image grid.

2. Identify Peak: Find the brightest pixel (peak) in the dirty image, which represents the location of a strong source.

3. Subtract Image: Subtract a scaled and shifted PSF (usually a Gaussian) centered at the location of the identified peak from the dirty image. This creates a residual image.

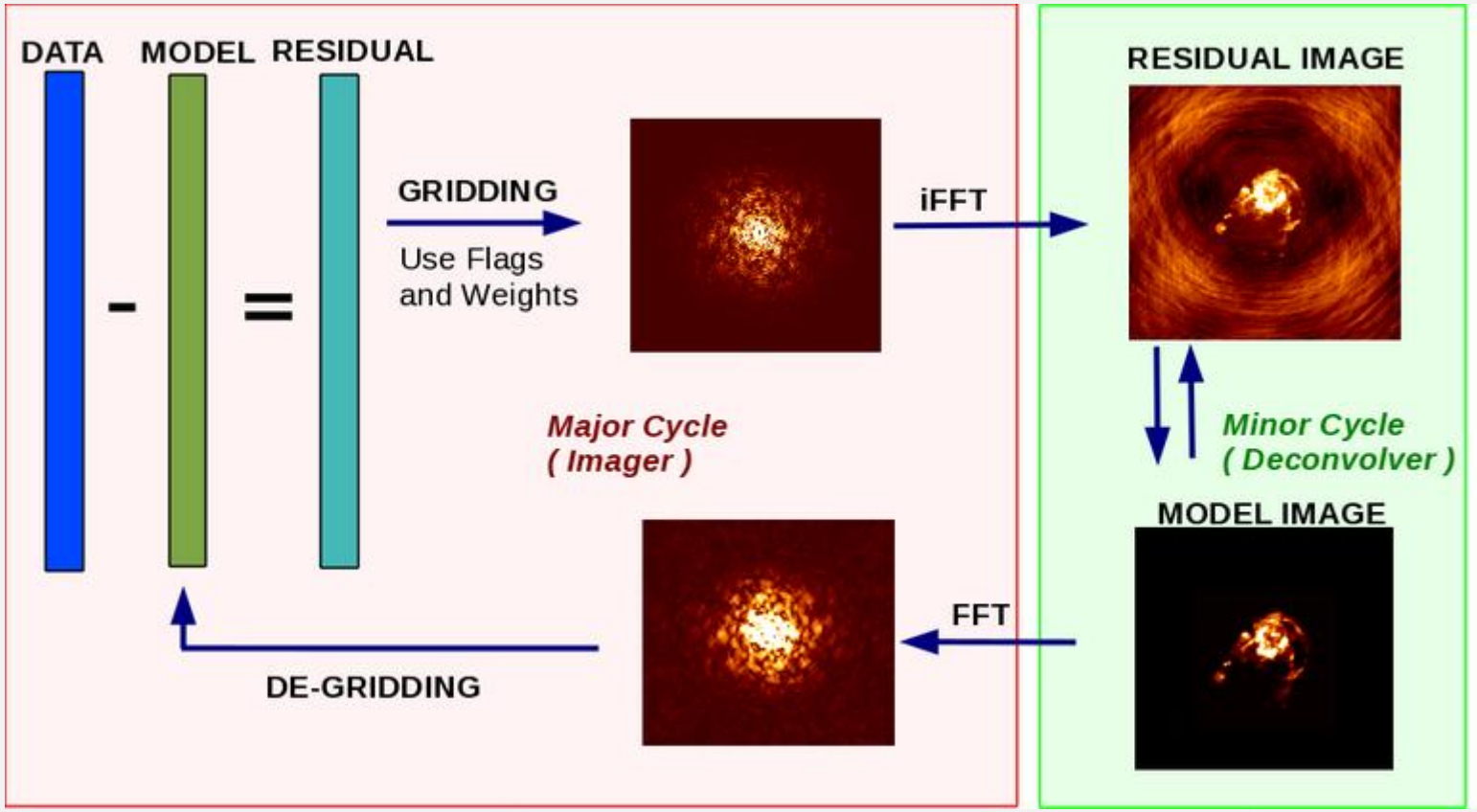
4. Update Model: Add the identified peak to the corresponding location in the model image.

5. Iterate: Repeat steps 2-4 until either a maximum number of iterations is reached or the residual image falls below a specified threshold level.

The final output is the model image + the residual image.

UNIVERSITY OF CAMBRIDGE

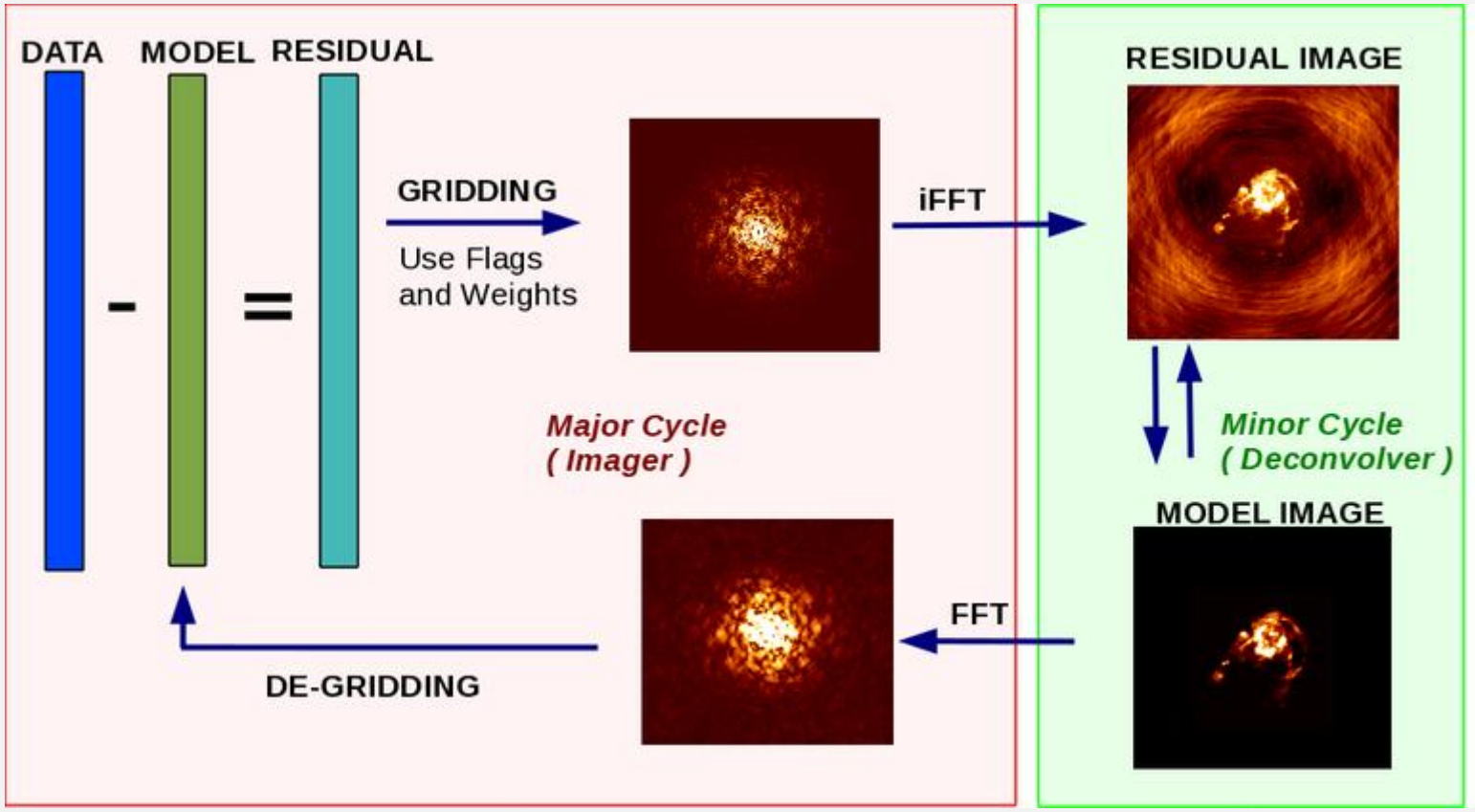
Schwab-Cotton CLEAN



Resource: https://casadocs.readthedocs.io/en/stable/notebooks/synthesis_imaging.html#Introduction

UNIVERSITY OF CAMBRIDGE

Maximum Entropy



Resource: https://casadocs.readthedocs.io/en/stable/notebooks/synthesis_imaging.html#Introduction