Imaging

Continuum & Spectral line

Sebastien Muller (Chalmers University of Technology, Nordic ARC node, Sweden)

European ALMA School 2026
Leiden - January 2026E

Who already did image/clean ALMA data?

What is/was the most difficult part to you?

Outlines

I- Interferometry to imaging

II- Imaging in practice

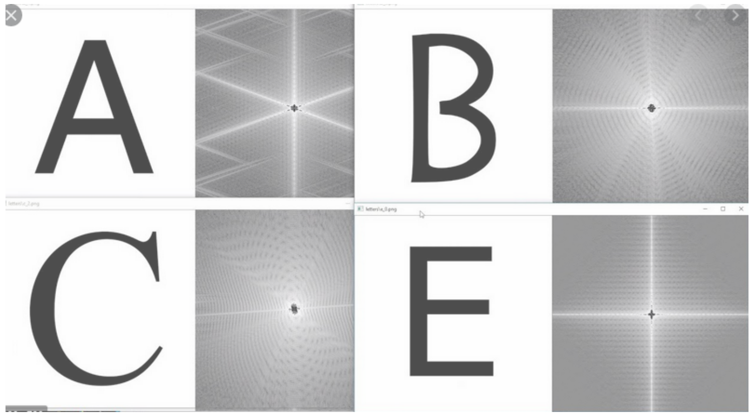
III- The short spacing issue

Part I

Interferometry to imaging

Jargon, jargon, jargon

An interferometer samples the
Fourier transform
of the source intensity distribution
I(x,y)
(modulated by the antenna primary beam P(x,y))

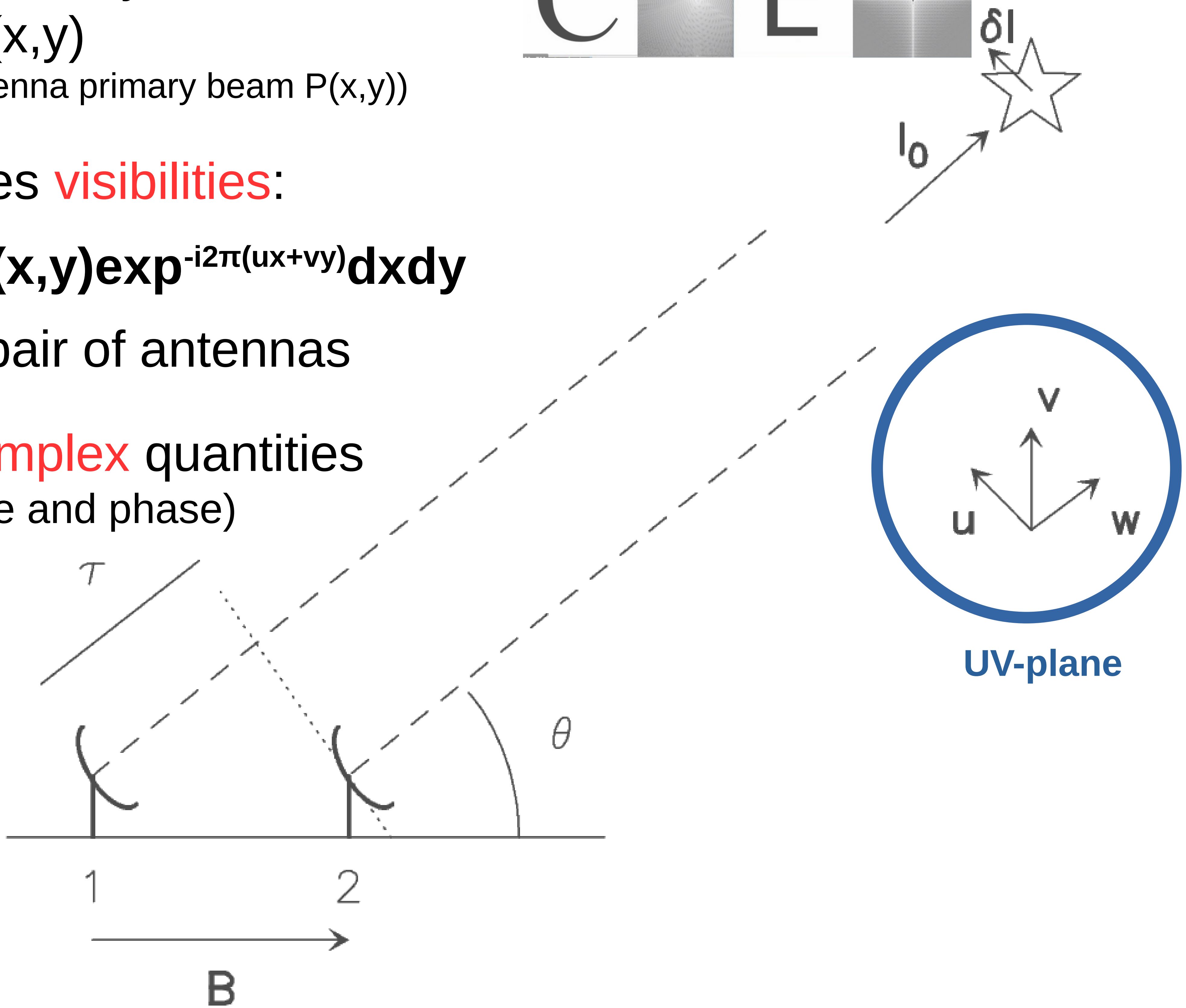


It measures visibilities:

$$\mathbf{V(u,v) = \iint P(x,y)I(x,y)exp^{-i2\pi(ux+vy)}dxdy}$$

sampled by pair of antennas

These are complex quantities
(amplitude and phase)



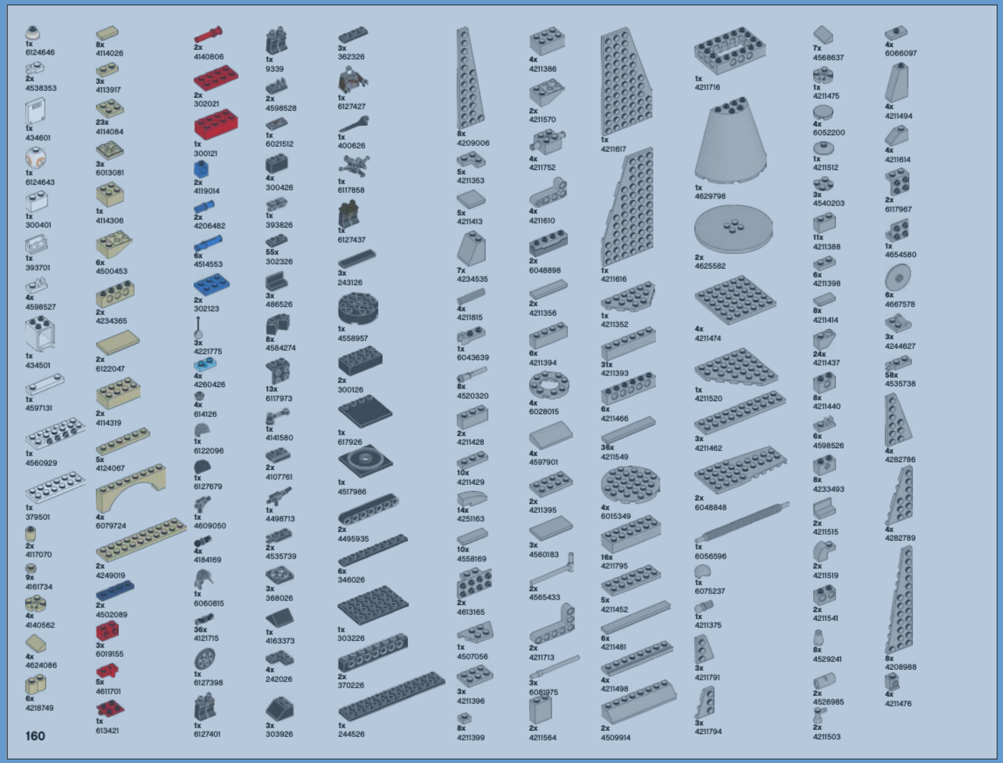
UV-plane

The Fourier transform decomposes an image into spatial frequencies (= break out into small and big pieces)

x,y

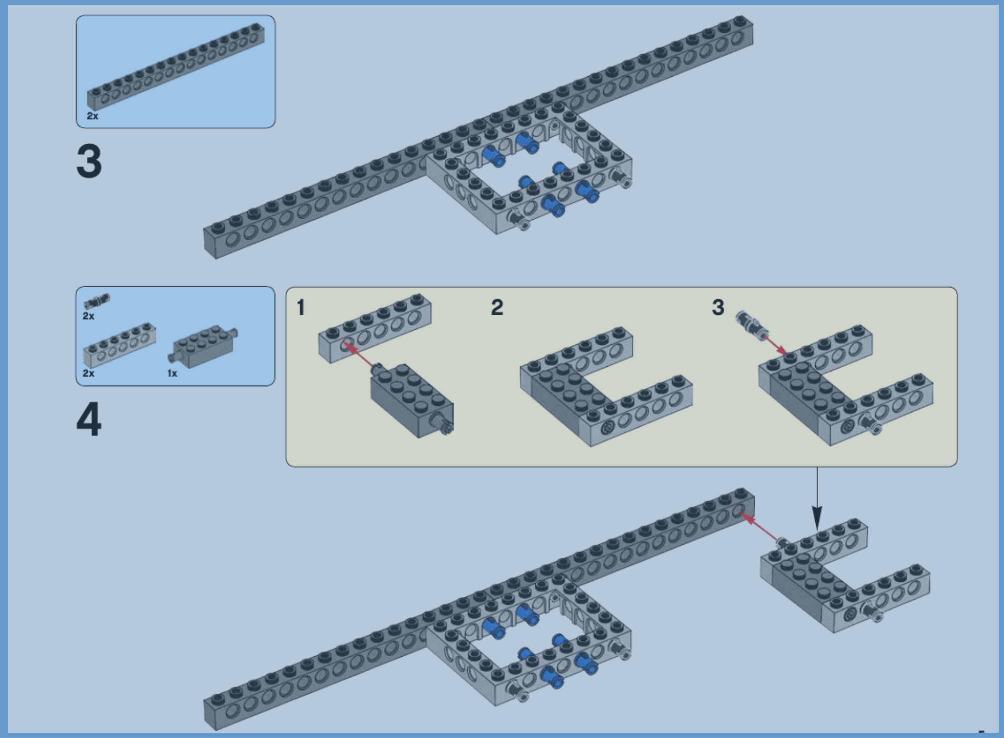
FT

Amplitude



u,v

Phase



$$V(u,v) = \iint I(x,y)\exp^{-i2\pi(ux+vy)}dxdy$$

The Fourier transform decomposes an image into spatial frequencies (= break out into small and big pieces)

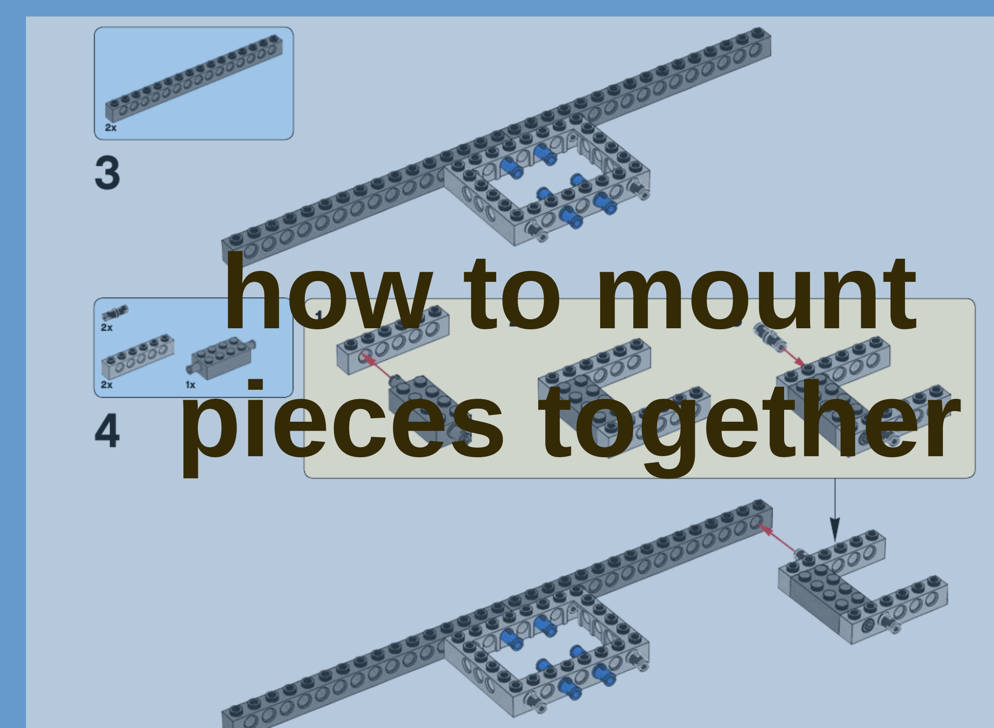
x,y

FT

Amplitude

u,v

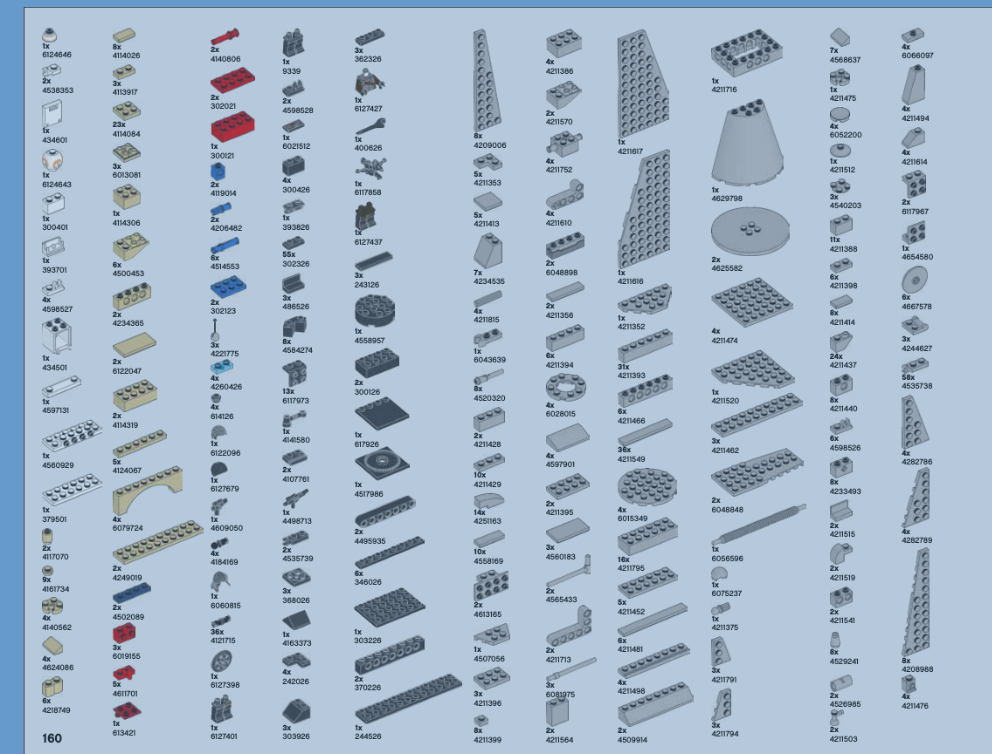
Phase



$$V(u,v) = \iint I(x,y)\exp^{-i2\pi(ux+vy)}dxdy$$

What we want is to reconstruct the image after the observations i.e., reassemble the pieces

Amplitude

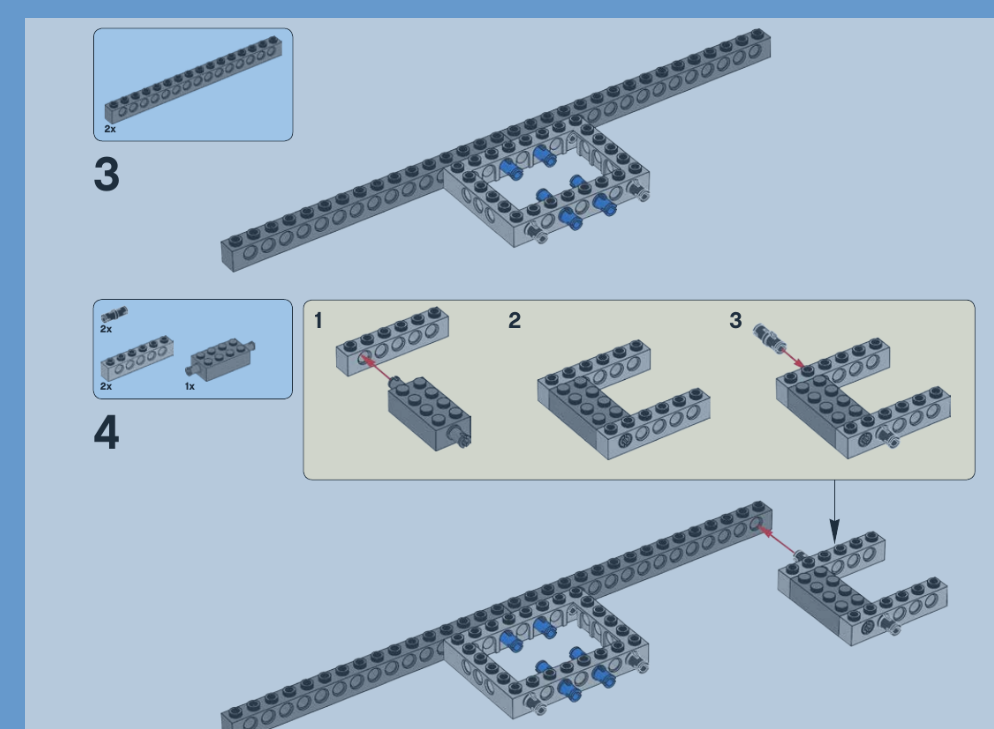


FT

x,y

u,v

Phase



$$I(x,y) = \iint V(u,v)\exp^{i2\pi(ux+vy)}dudv$$

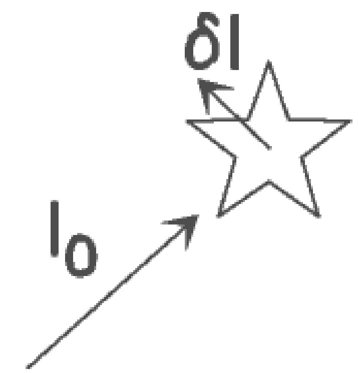
= Imaging

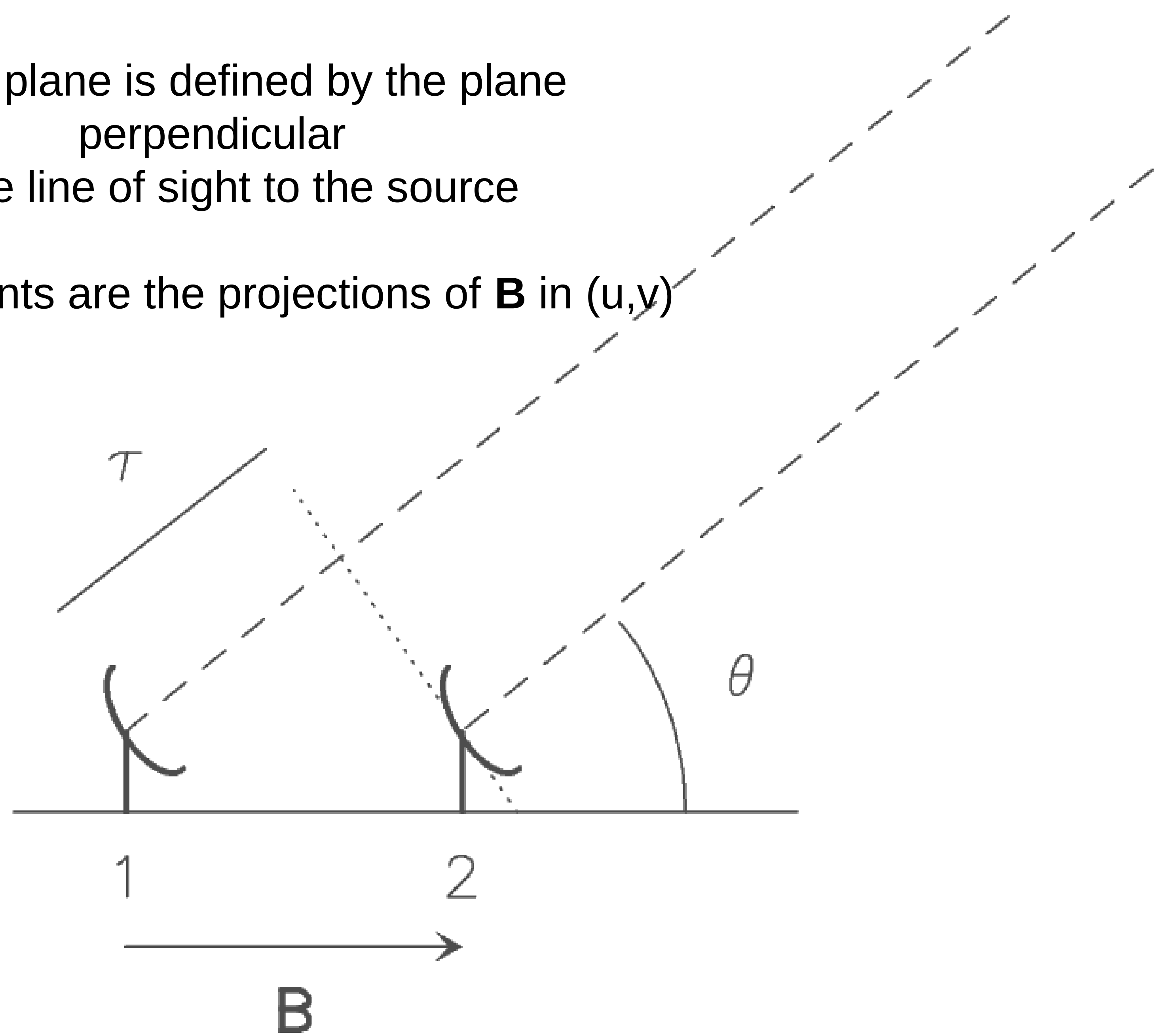
The uv plane

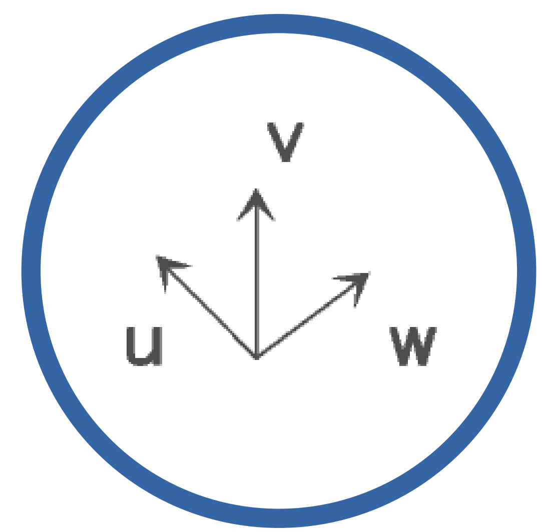
$$V(u,v) = \iint P(x,y)I(x,y)\exp^{-i2\pi(ux+vy)}dxdy$$

The uv plane is defined by the plane perpendicular
to the line of sight to the source

Sampled points are the projections of **B** in (u,v)





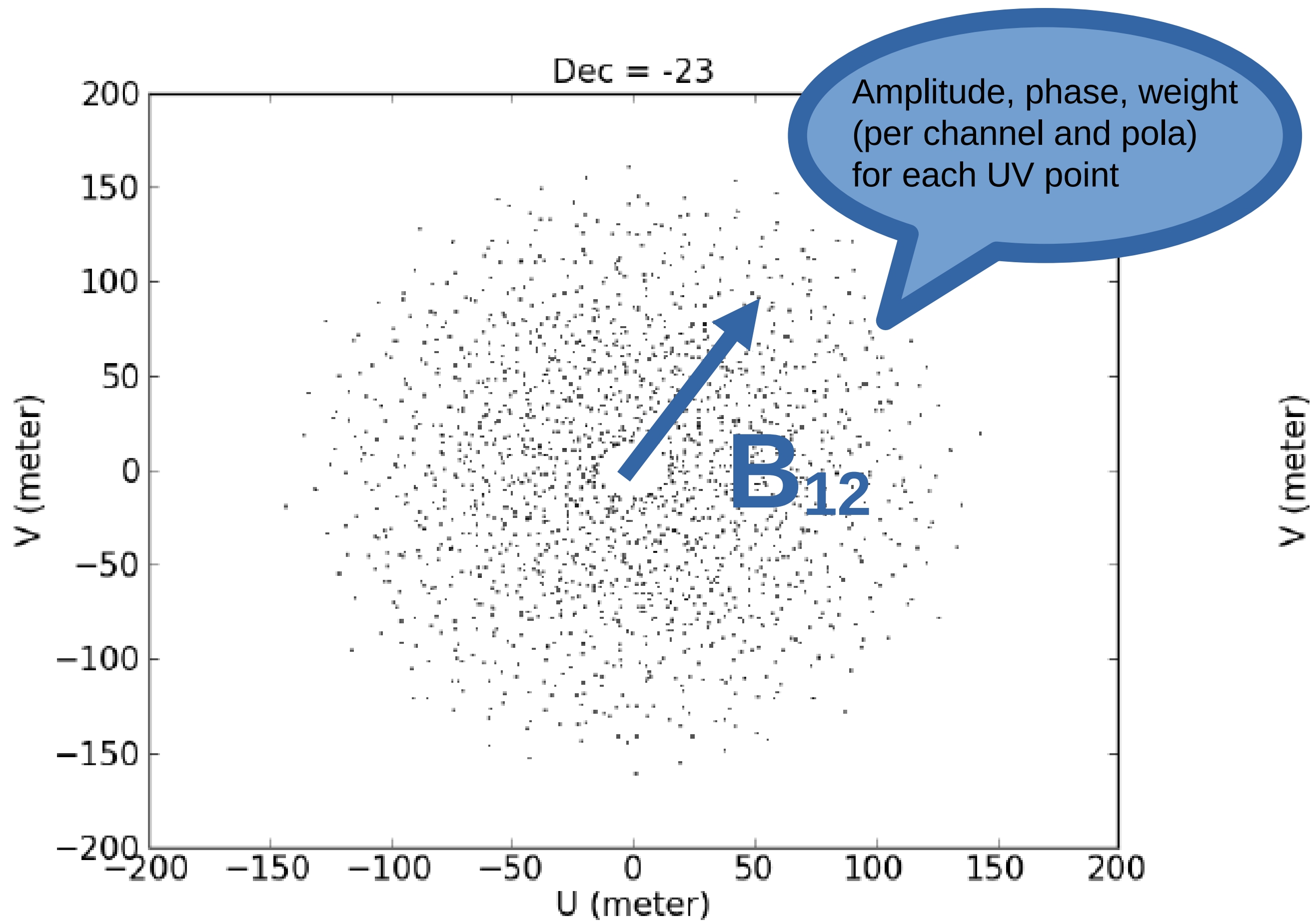


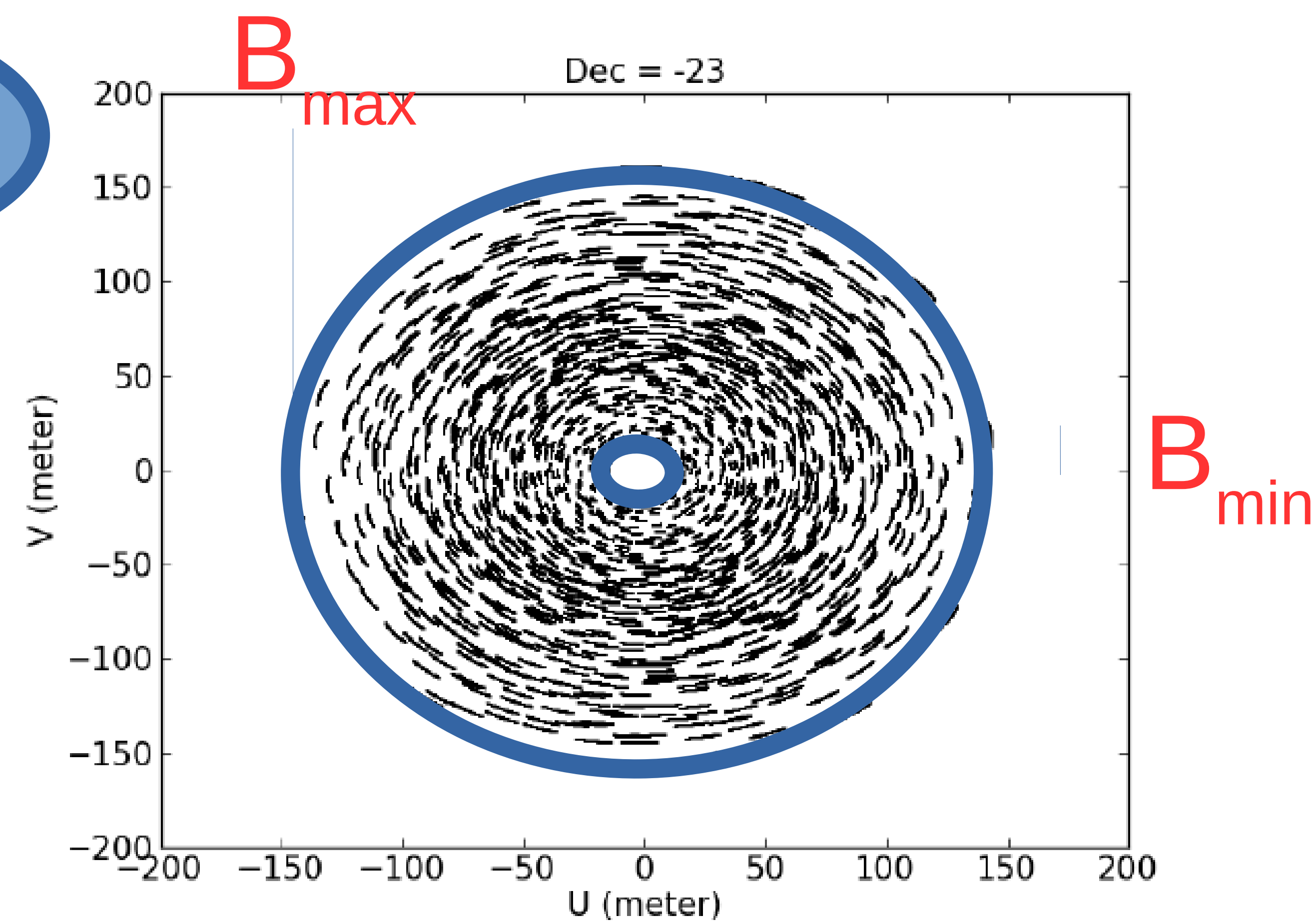
UV-plane

Delay = $\mathbf{B.L_0} / c$

u,v in wavelength unit
(or distance)

uv-plane coverage = sampling of spatial frequencies



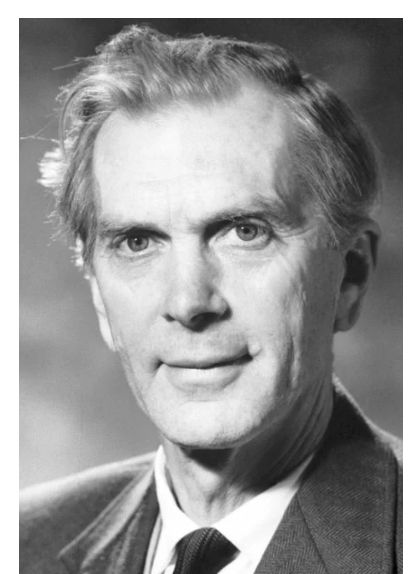


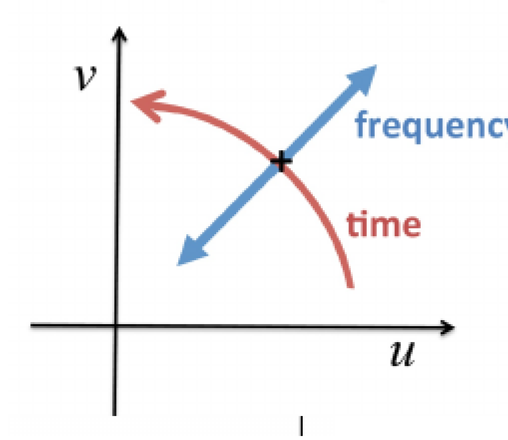
Integration: 1 min 1 h ("Eppur si muove!")

With many antennas as for the ALMA array,
even a snapshot observation fills in reasonably well
the uv-plane

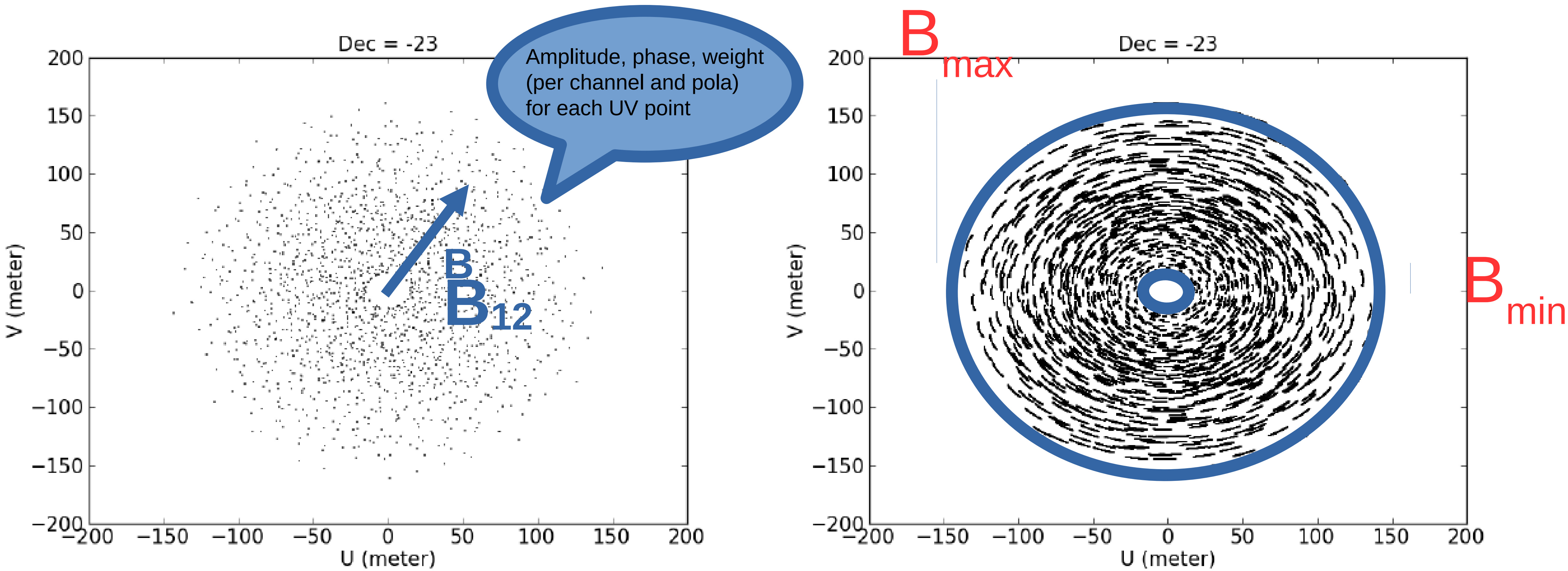
Aperture (super)synthesis is done by Earth rotation

[Sir Martin Ryle (1918-1984), Nobel Prize in Physics 1974]





uv-plane coverage = sampling of spatial frequencies



Integration: 1 min 1 h

But not all positions u,v are measured !!!

Incomplete sampling ... loss of information ... altering image reconstruction

B_{max} Longest baselines ... Resolution ~ λ/B_{max}

B_{min} Central hole in uv-plane ... Short spacing issue for extended sources

$$I(x,y) = \iint_{-\infty}^{+\infty} V(u,v)\exp^{i2\pi(ux+vy)}dudv$$

But we do not have all V(u,v) measurements :(

= Need some regridding of the measured visibilities onto a regular uv-grid prior to an inverse FFT

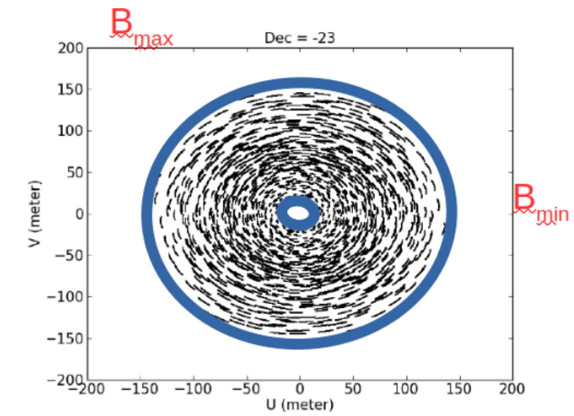
Sampling & Weighting

An interferometer can only measure a limited number of visibilities.

Let's define the sampling function **S(u,v)** of positions in the uv-plane:

S(u,v) = 1 where we get observations

S(u,v) = 0 everywhere else



We can also introduce an arbitrary weight function **W(u,v)**, and in total we can define:

$$I_W(x,y) = 1/P(x,y).\iint V(u,v).S(u,v).W(u,v).\exp^{i2\pi(ux+vy)}.dudv$$

$$I_w(x,y) = 1/P(x,y).\iint V(u,v).S(u,v).W(u,v).\exp^{i2\pi(ux+vy)}dudv$$

From the Fourier Transform properties: $FT(f.g) = F{**}G$

Let's define a new function:

$$D(x,y) = \iint S(u,v).W(u,v).\exp^{i2\pi(ux+vy)}dudv \quad (= FT^{-1}(SW))$$

depends on the observations

we get:

$$I_w = D ** (P.I_{sou}) + N$$

where we also introduce a noise function N

The “image” of the real source is degraded and with noise :(

Dirty beam (PSF)

$$\mathbf{I_w = {\color{red}D} ** (P.I_{sou}) + N}$$

- An interferometer produces an altered image of the source $I_{sou} \text{ --> } I_w$

- D is the image of a **point source** observed by the interferometer

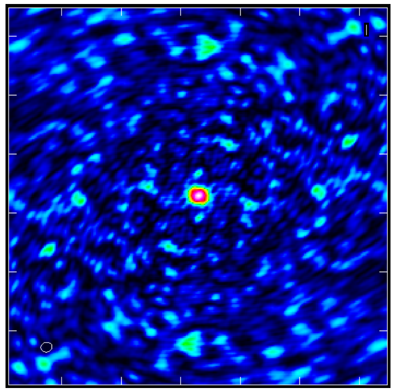
- In radio, D is called the dirty beam (Point Spread Function)

- $D = FT^{-1}(SW)$ … it depends on the uv-coverage and the weighting function

- Since $SW(0,0) = 0$ then $\iint D(x,y).dxdy = 0$

- By symmetry $SW(u,v) = SW(-u,-v)$, therefore $D(x,y) = D(-x,-y)$

D(x,y) has a peak at the center, with ***positive*** and ***negative*** sidelobes



Deconvolution

D is annoying ... we want to get rid of it
and recover $\mathbf{I_{sou}(x,y)}$ from the observations

as good as possible
(modulo missing information and noise)

This is called “Deconvolution”

Deconvolution: The “basic” Clean algorithm

Assumptions: -The source can be described as a collection of point sources (clean components)

-The sky is almost empty

0) Start: Residuals = Dirty image Model clean components (CC) = 0

1) Find the position of the maximum I_{max} in the residuals

2) There, subtract γI_{max}**D from residuals Add γI_{max} to the CC model

γ is a gain, usually 0.1-0.2

3) Back to 1) until a stopping criterion is met

(**noise threshold** in the residuals, number of iterations **niter**)

4) Convolve the clean components with the clean beam (Gaussian fit to the central peak)

== take into account the finite resolution

unit is now Jy/beam

5) Add the last residuals == take into account noise

Weighting

Noise for a given visibility: $\sigma(u,v) \propto J.T_{sys} / (\eta_Q \sqrt{(2\Delta\nu.t_{int})})$

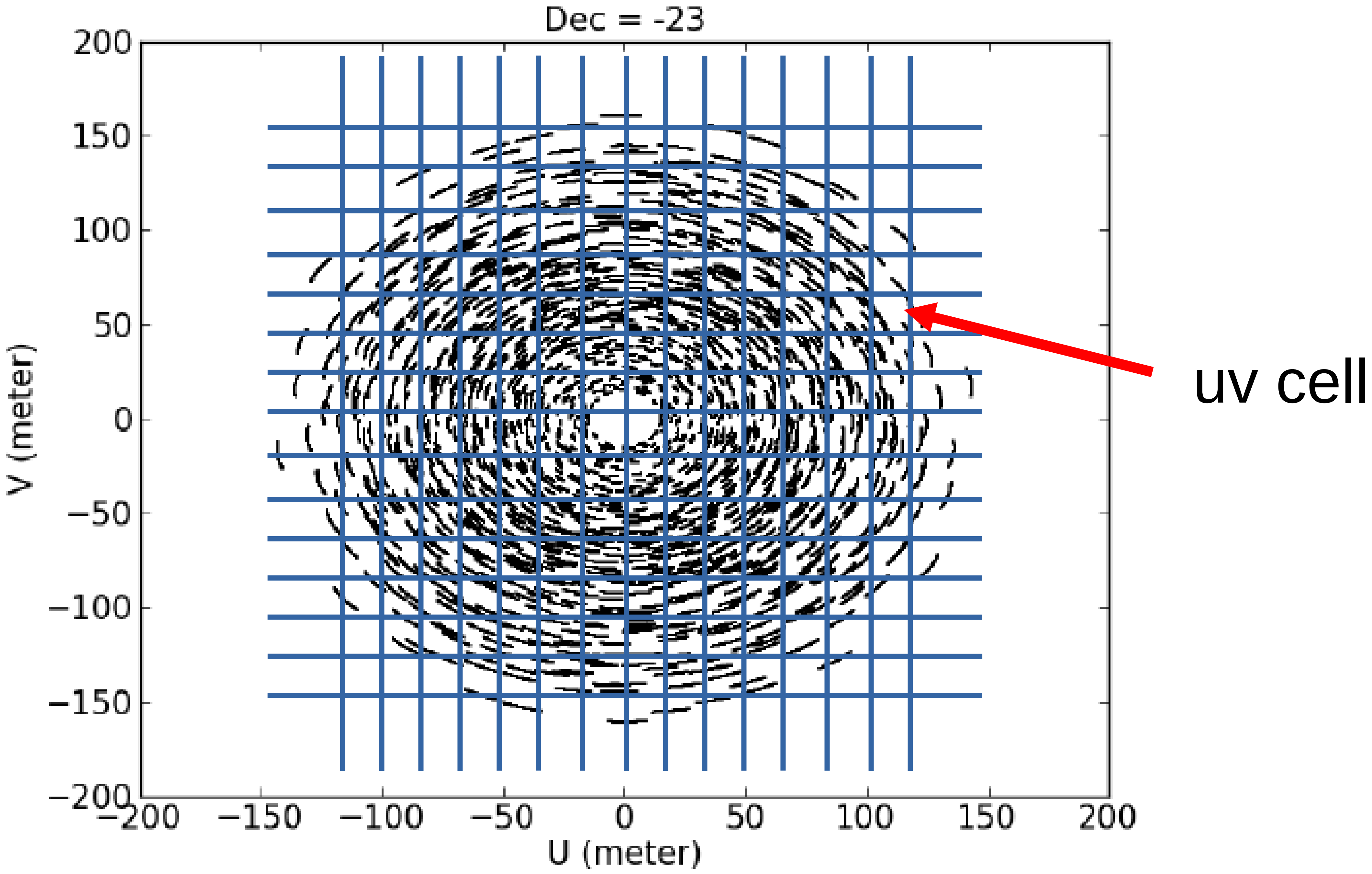
- J : antenna temperature to flux density $= 2k/(\eta_A A)$
- T_{sys} : system temperature
- η_Q : bit quantization (number of digits)
- $\Delta\nu$: channel bandwidth
- t_{int} : integration time

Weight: $\mathbf{w = 1 / \sigma^2}$

Weights carry the “confidence” in the data

Weighting is arbitrary

Gridding in the uv plane



Weighting: Natural & Uniform

Natural

Keep the original weights

$w_k = 1 / \sigma_k^2$

More weights on short baselines

Best sensitivity

But poorer beam shape and sidelobes

Uniform

Change the weights so that the the sum of weights in a uv cell $\Sigma(W.S)$ is a constant

(or 0, if no uv data in the cell)

More weights on long baselines

Better resolution, better sidelobes

But higher noise level

Weighting: Briggs

Natural

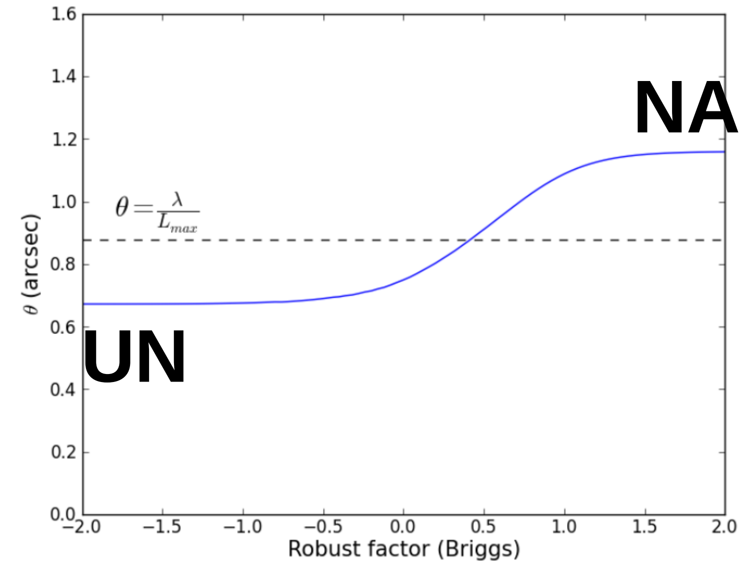
better rms | worse beam

Briggs (1995)

Offers a continuous bridge between NA and UN

Controlled by parameter 'Robust'

robust = -2 == close to Uniform
robust = +2 == close to Natural



tclean(..., weighting='briggs', robust=0.5)

offers a good trade-off between angular resolution, noise, and sidelobe levels

better beam | worse rms

Uniform

Weighting: tapering

Multiply the visibility weights by a Gaussian envelope:

$\exp(-(u^2+v^2) / \mathbf{t}^2))$ where t is the tapering size

Corresponds to smoothing the data in the image plane
(convolution with a Gaussian)

Kind of throw away some data ... not the best

Better to adjust the array configuration, when possible

```
tclean(..., uvtaper=['1.5arcsec', '1.0arcsec', '30deg'])     # [bmaj, bmin,pa]
```

Part II

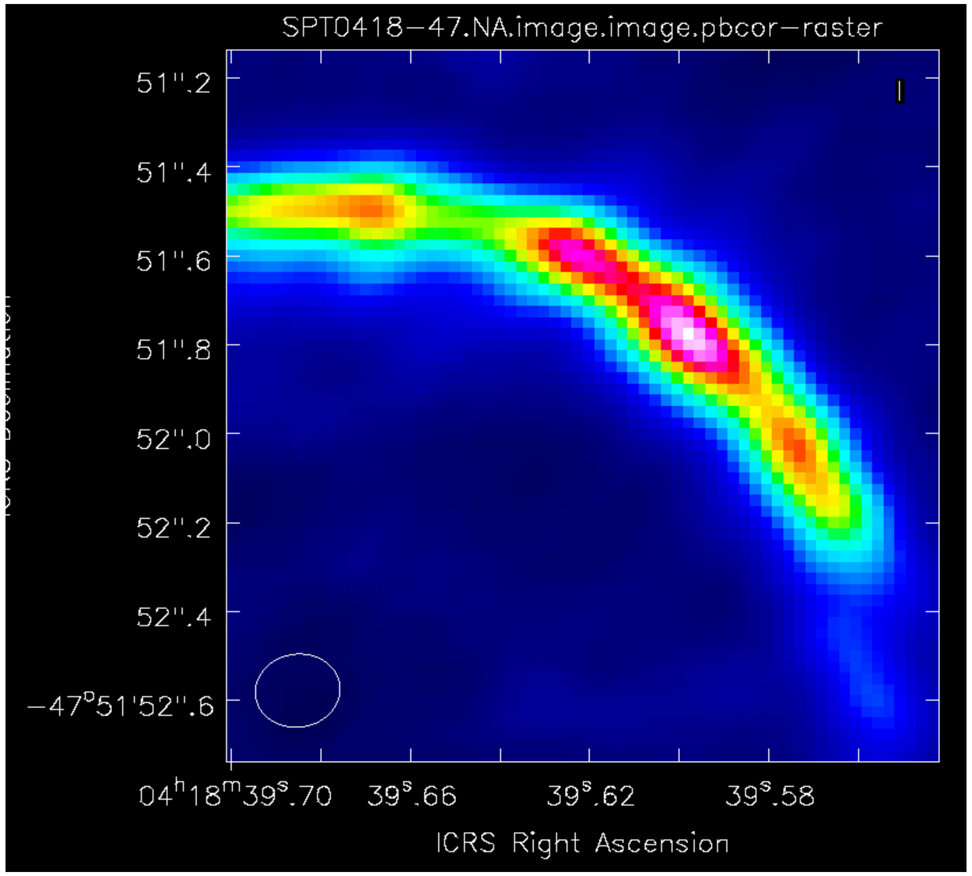
Imaging in practice

Implementation in CASA

A basic run of tclean

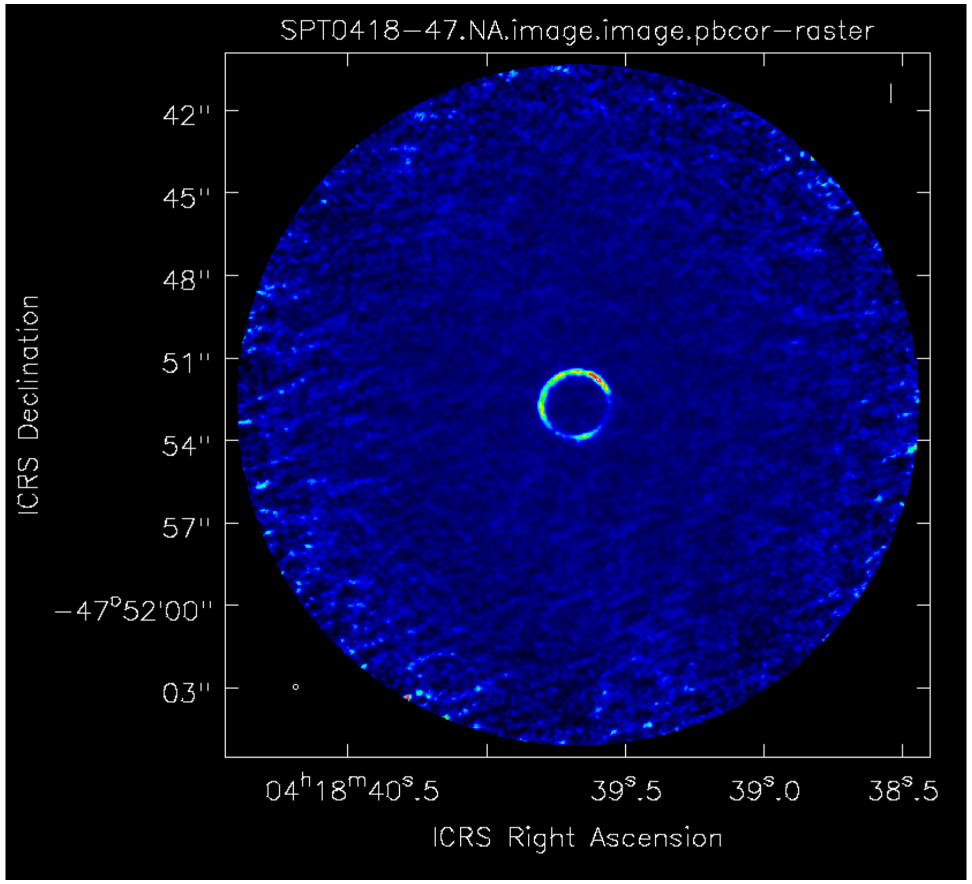
```
CASA>
imagename = ‘SPT0418-47_Cont_mfs_NA.image'
os.system('rm -rf '+imagename+'.*')

tclean(vis=’uid___A002_Xc1b25a_X71a6.ms’, # your calibrated measurement set
       spw='29,31',                              # spectral windows, see listobs
       imagename=imagename,
       datacolumn='corrected',                   # or ‘data’ if split before
       field= 'SPT0418-47',                      # source name, or id
       cell=’0.025arcsec’,                       # cell size typically 5-8 pixels/beam
       imsize=[1024,1024],                       # enough to cover the FOV
       mask=['circle[ [04h18m39.670000s, -47d51m52.70000s], 2.0arcsec]'],
       interactive=False,
       specmode='mfs',
       weighting='natural',
       niter=10000,
       threshold='0.1mJy'                # check sensitivity request, or theoretical noise
       )
```

tclean parameters: cell & imsize



cell: it should respect proper sampling compared to the synthesized beam width

In practice, 5--8 pixels per beam is a good choice



imsize: good practice to make at least a quick check of the full FOV for the first clean. Cover lambda/D (or mosaic size) where D is the antenna diameter.

CASA internal FFT routines prefers an even number of pixels, and factorizable by 2,3,5 only

Outputs of tclean

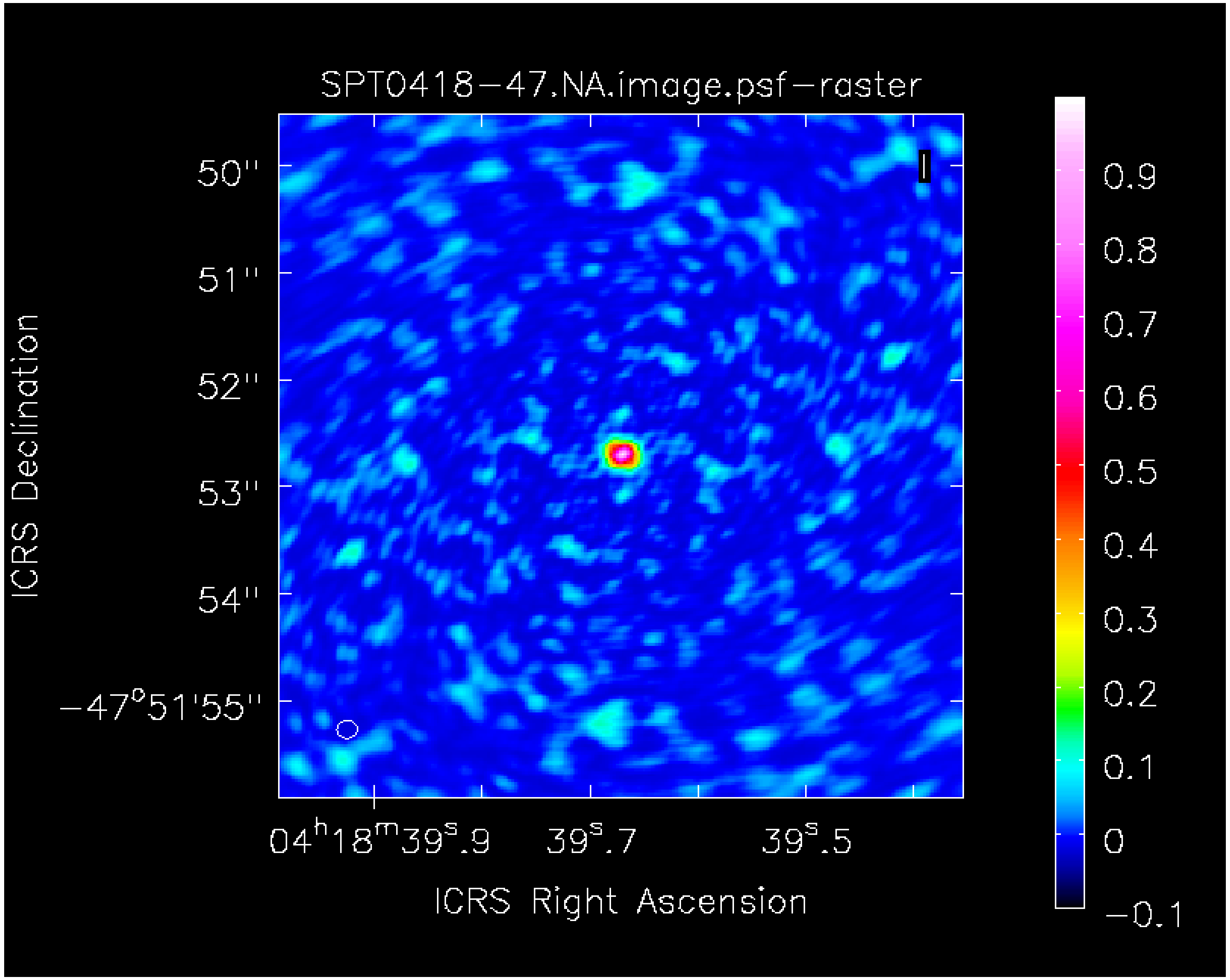
A tclean run would typically create all those CASA images:

- imagename.psf
- imagename.pb
- imagename.mask
- imagename.sumwt
- imagename.model
- imagename.residual
- imagename.image
- imagename.image.pbcor (if **pbcor=True** is set)

Run **tclean(...,niter=0)** to create a dirty map

Use **exportfits** to create FITS file from any of these CASA images.

Outputs of tclean: image.psf



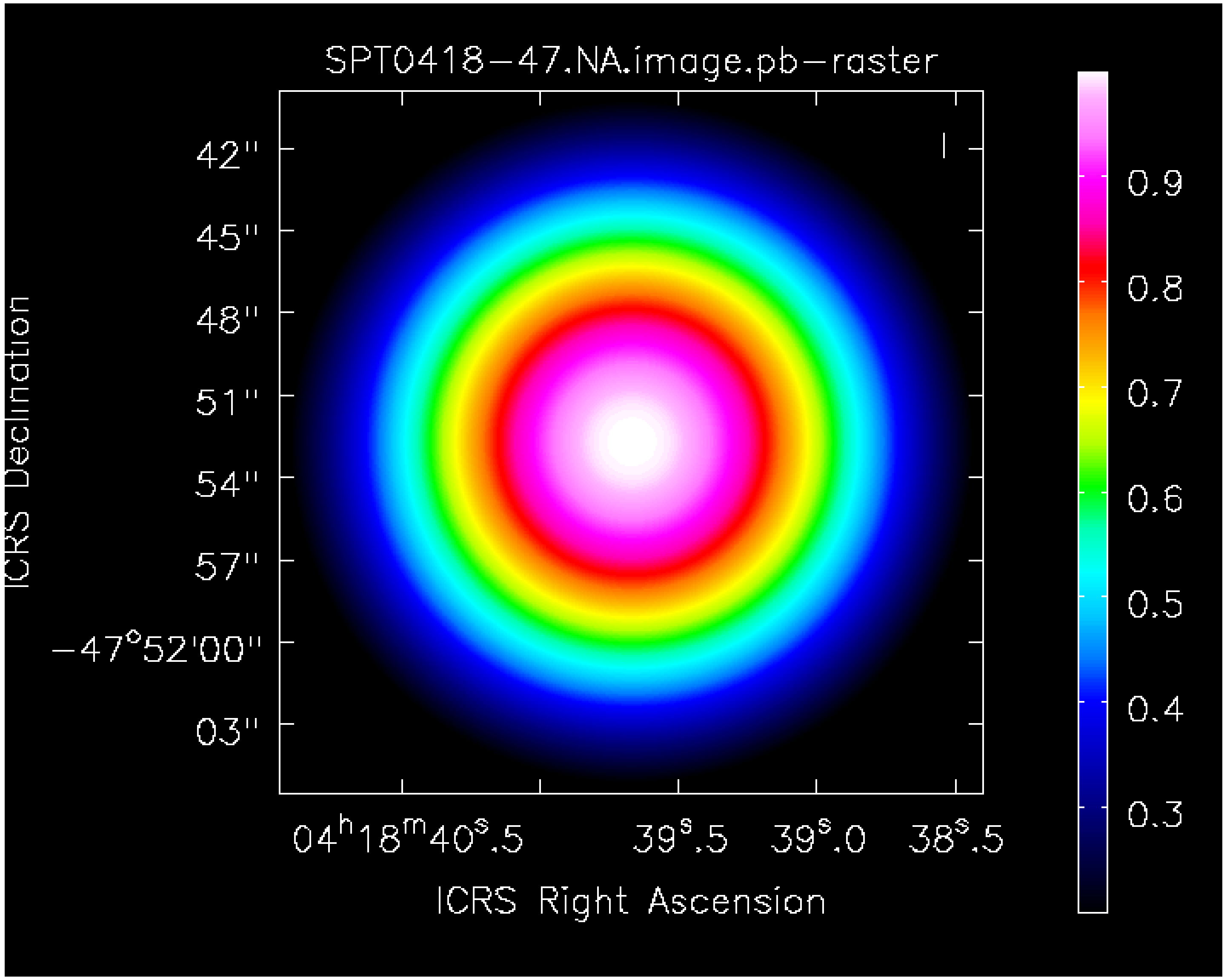
The dirty beam

(including sidelobes)

Normalized to peak =1

Central region fit to a Gaussian to get the clean beam

Outputs of tclean: image.pb

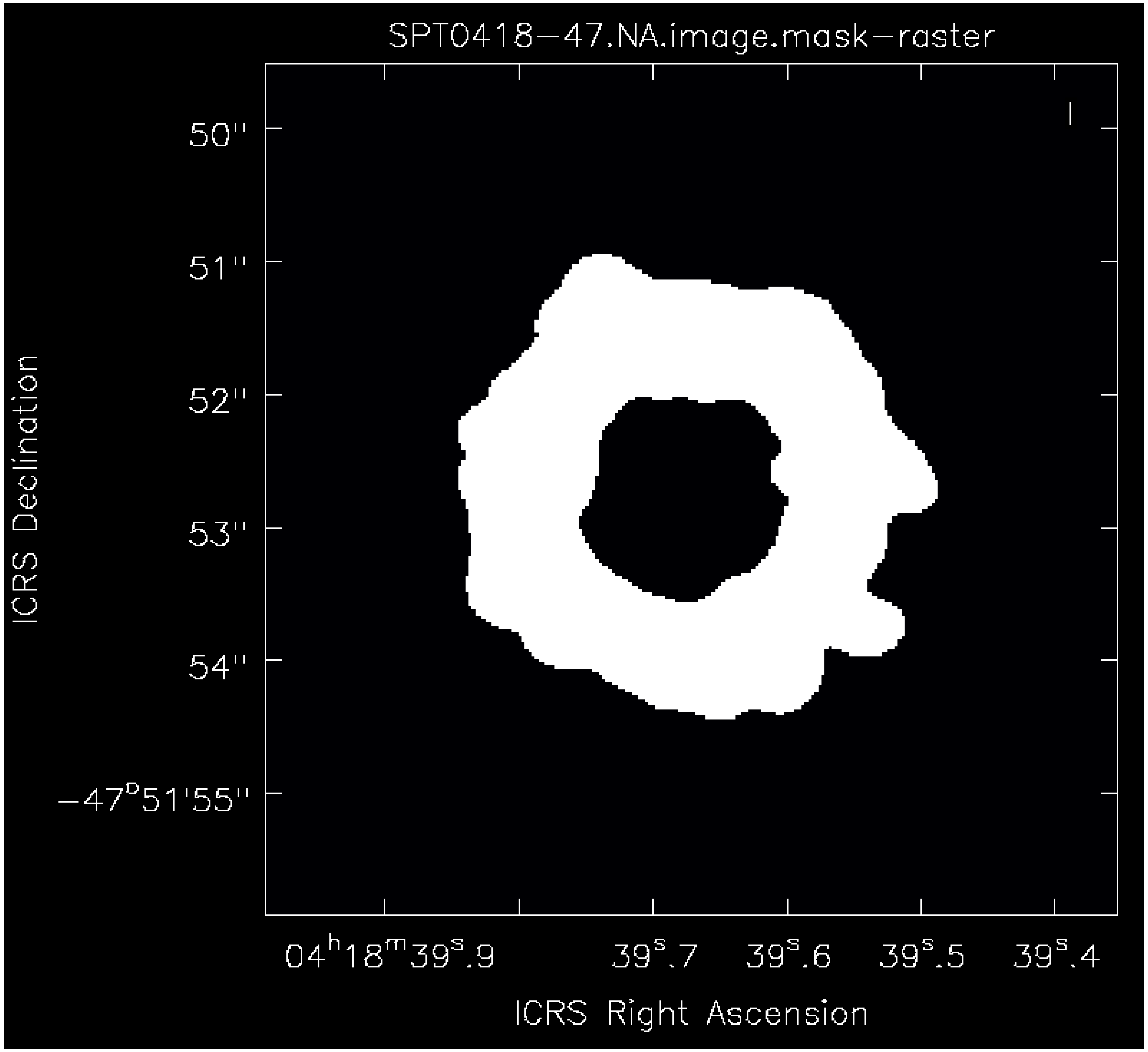


The normalized sensitivity map

(= antenna primary beam
for single pointing)

Region below the limit given by pblim (=0.2) is not used for primary beam correction

Outputs of tclean: image.mask

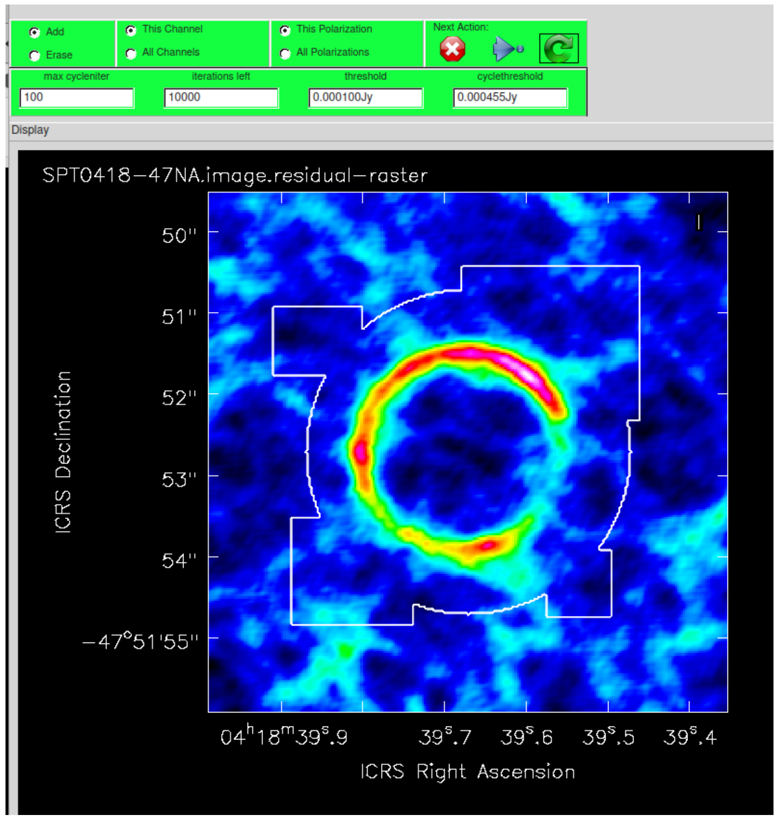


The mask

(imposed or from automasking)

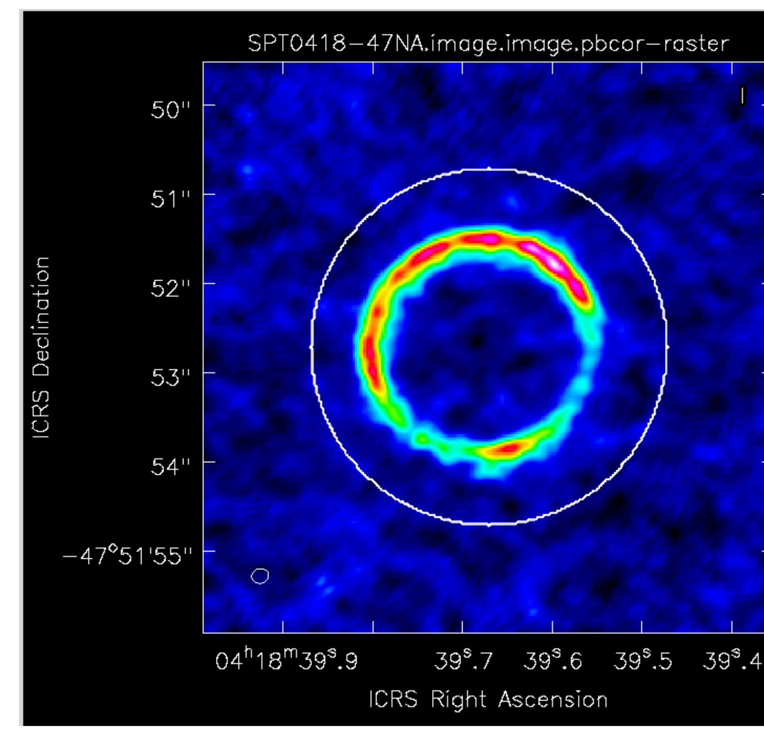
Clean components allowed to be within the white region

Masking



Edit the mask area manually (and iterate if needed):

tclean(…
interactive = True)

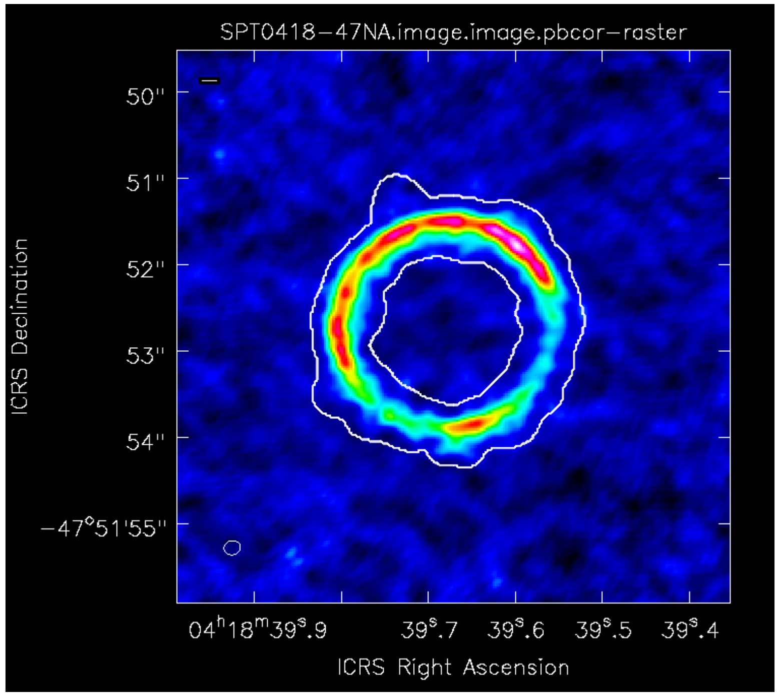


Or specifying a given mask:

tclean(…
interactive = False
mask = ['circle[[04h18m39.67s, -47d51m52.7s], 2.0arcsec]'])

Automasking

```
tclean( ...
        niter=10000,
        threshold='0.1mJy',
        interactive = False,
        # Automasking parameters below this line:
        usemask='auto-multithresh',
        noisethreshold=4.25,
        sidelobethreshold=2.0,
        lownoisethreshold=1.5,
        minbeamfrac=0.3,
        negativethreshold=0.0,
        fastnoise=False)
```

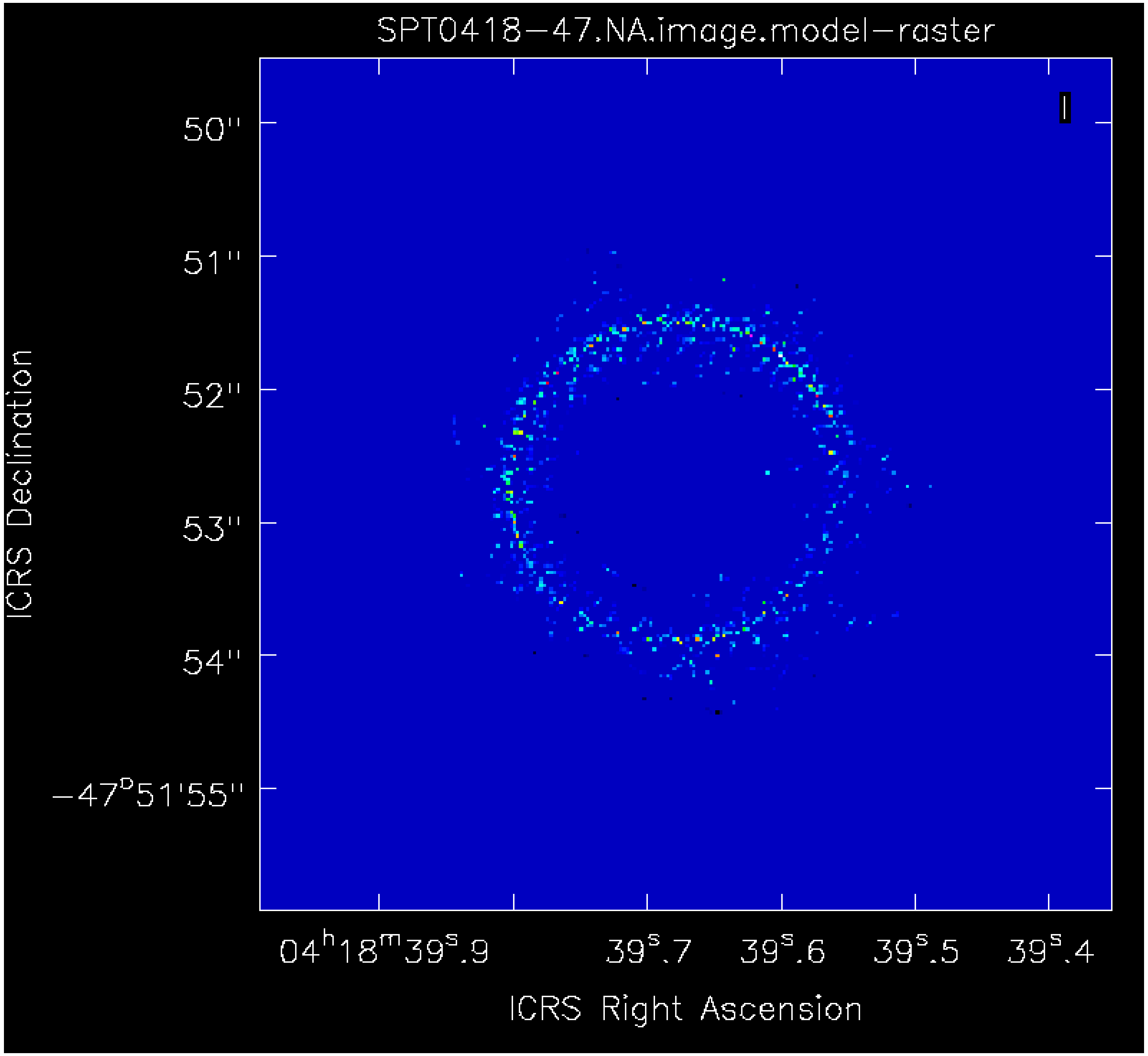


Recommendations for optimized automasking parameters under:

https://casaguides.nrao.edu/index.php?title=Automasking_Guide_CASA_6.5.4

Parameter	7m (continnum/ line)	12m (b75 < 300m)	12m (300m < b75 < 400m)	12m (b75 > 400m)	12m + 7m combined *TENTATIVE*
noisethreshold	5.0	**4.25**	5.0	5.0	4.25
sidelobethreshold	1.25	**2.0**	2.0	2.5	2.0
lownoisethreshold	2.0	**1.5**	1.5	1.5	1.5
minbeamfrac	0.1	**0.3**	0.3	0.3	0.3
negativethreshold	0.0	**0.0 (continuum)** **15.0 (line)**	0.0 (continuum) 7.0 (line)	0.0 (continuum) 7.0 (line)	0.0
fastnoise	False	**False**	False	True	False

Outputs of tclean: image.model



The model

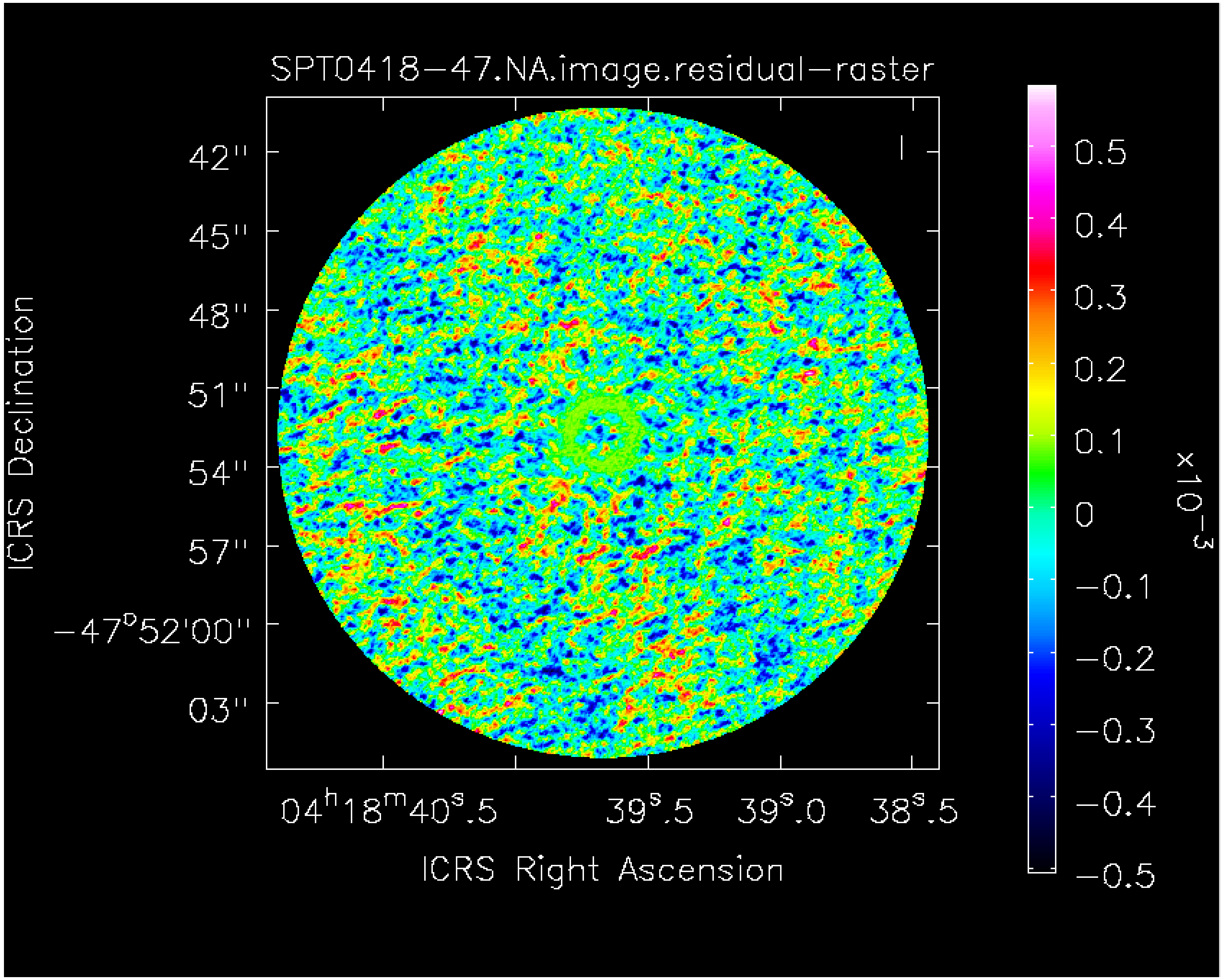
Clean components

= list of point sources

(not convolved)

Unit: Jy/pix

Outputs of tclean: image.residual

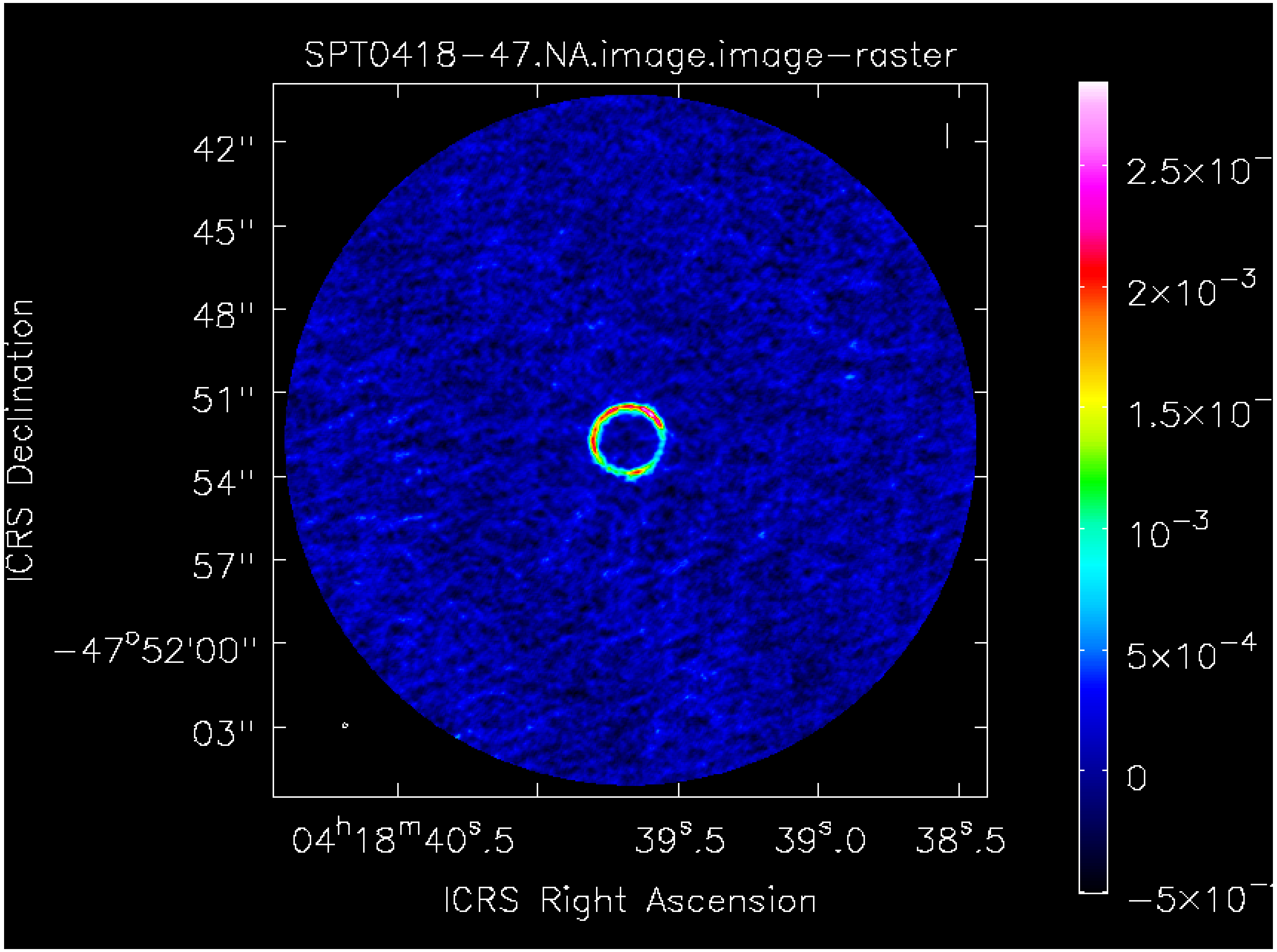


The residuals after last iteration of the Clean deconvolution

Check for potential systematics (bad clean, dynamic range issues, missing fluxes, bad data, ...)

Outputs of tclean: image.image

The clean image:



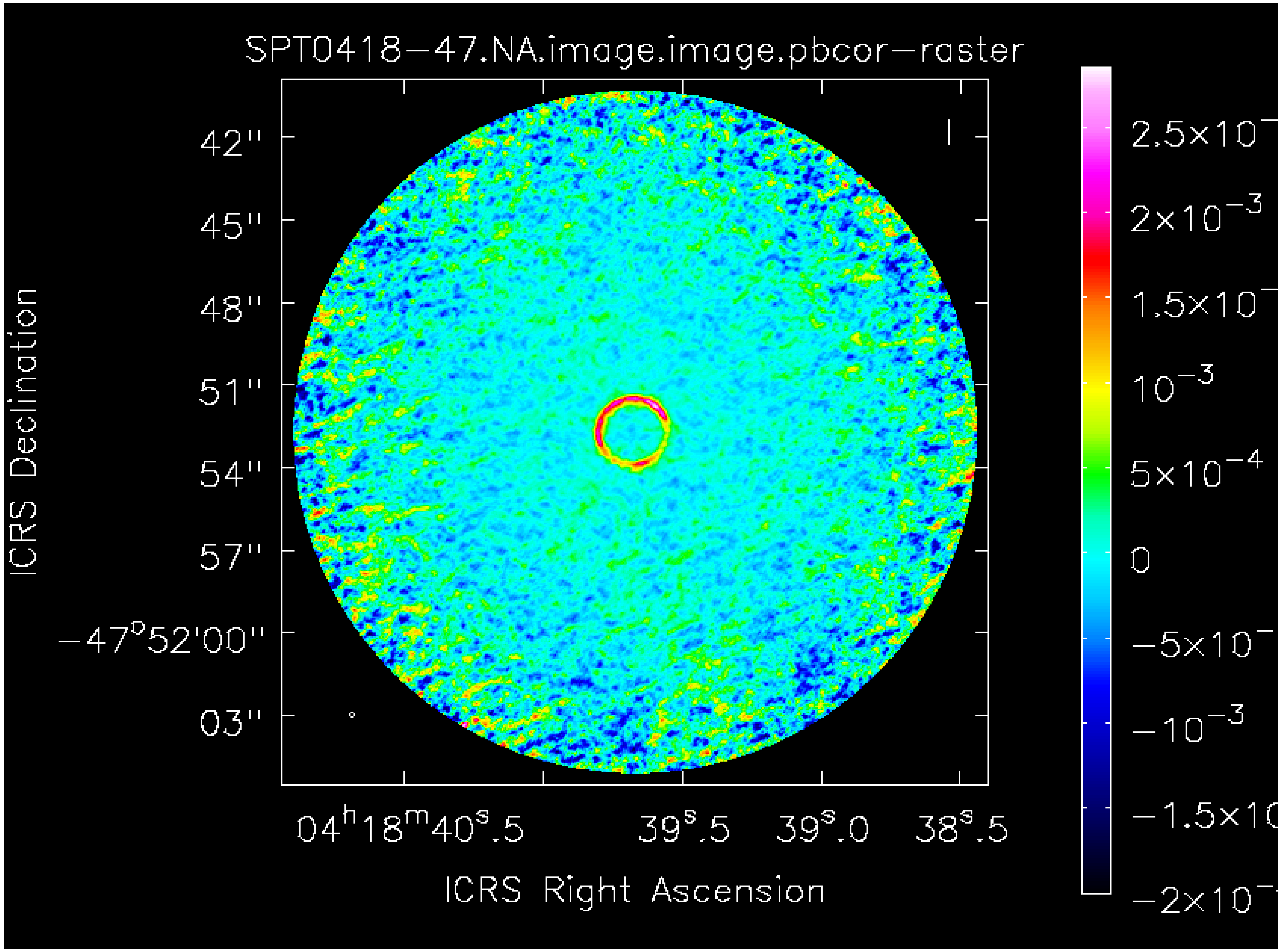
The clean model convolved with the clean beam
+
the final residuals

Unit: Jy/beam
!!!
Beam refers to the clean beam

Uniform noise across the field
but
!!!Flux not correct toward the edges

Outputs of tclean: image.image.pbcor

The primary-beam corrected clean image:



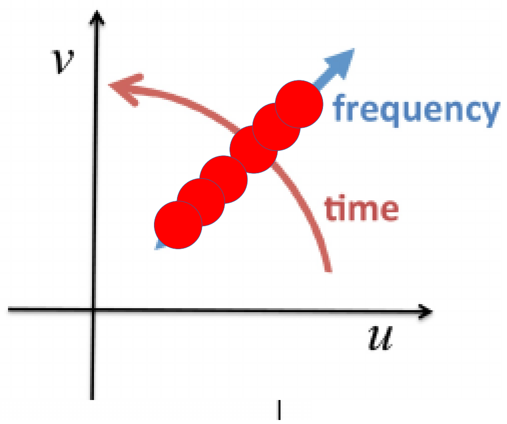
The clean model convolved with the clean beam
+
the final residuals

and
correction from primary beam

Unit: Jy/beam
Beam refers to the clean beam

!!!Non-uniform noise across the field
but
Fluxes corrected through the field

Specifics of continuum imaging



Continuum imaging, with only one output image channel

Multi-frequency synthesis (MFS)

Coordinates in the u,v plane depend on wavelength

Observing with multiple frequency channels
= better sampling of the uv-plane = better image fidelity

tclean(...
specmode = ‘mfs’)

Specifics of spectral line imaging

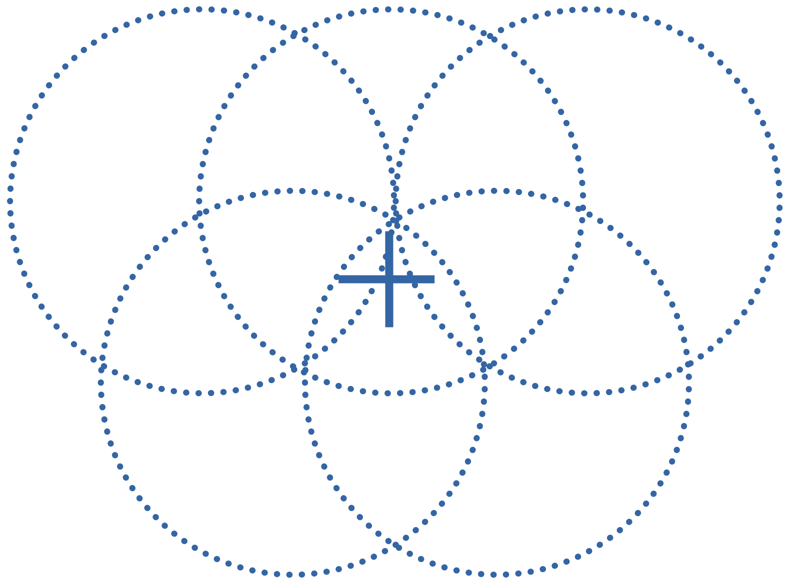
```
tclean(…
        specmode='cube',

        nchan=100,              # number of channels,
        start='-500km/s',       # channel number, velocity, or frequency of the first channel
        width='10km/s',         # channel width, by number of chan, velo, or freq

        restfreq = '115.271GHz',   # e.g., for CO(J=1-0)
        outframe='LSRK',
        ...)
```

- For line **Doppler tracking**, select **outframe** (TOPO, LSRK, BARY, ...)
  LSRK: Local Standard of Rest, kinematic
  = mean motion of material in the Milky Way in the neighborhood of the Sun

- Use **restoringbeam='common'** if you need same beam across entire cube

- **Continuum subtraction**
  Select line-free channels (not always easy)
  in Fourier plane, use the CASA task: uvcontsub
  in Image plane, use, e.g., STATCONT (Sanchez-Monge, 2018 A&A 609, 101)

Specifics for mosaics



For sources bigger than
~1/2 antenna primary beam field of view

```
tclean(...
        gridder = 'mosaic',    # instead of 'standard'
```

optional:

```
        phasecenter='ICRS 18.30.00 -21.11.00',
        mosweight = True)
```

Specify phasecenter if you want to center the mosaic
(default center is on first field, i.e., often at one corner of the mosaic)

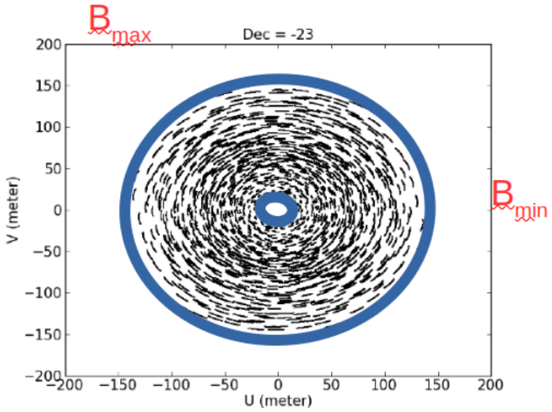
mosweight=True: weight each field in a mosaic independently
(better for poor uv-coverage or non-uniform sensitivity across the mosaic fields)
=False: calculate the weight density from the average uv distribution of all fields combined

Try multi-scale cleaning, array combination, if the source is extended

Part III

Short spacings

Problem of short spacings



$$V(u,v) = FT(I) = \iint I(x,y) \exp^{-i2\pi(ux+vy)} dx dy$$

#1 $V(0,0) = \iint I(x,y)\,dxdy$ = total flux of the source !

#2 No measurements for u,v inside a disk of radius B_{min}

= Loss of structures more extended than ~ λ/B_{min}

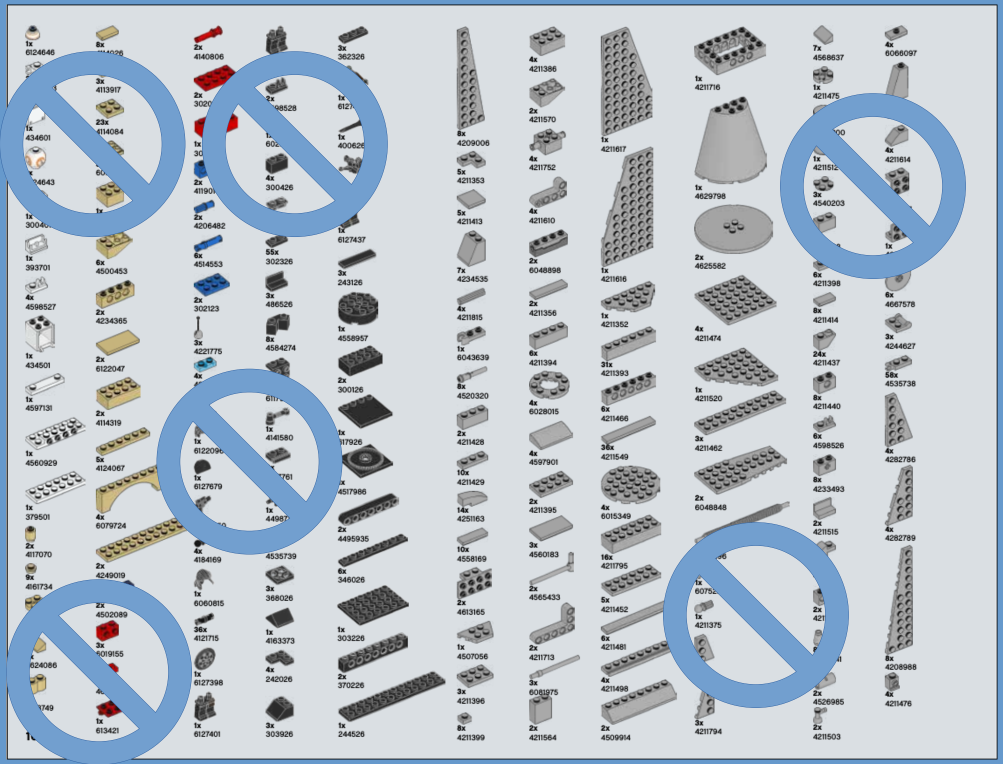
= Potential artefacts introduced into the reconstructed images

= Possible need for complementary data (smaller array, single dish)

Problem:

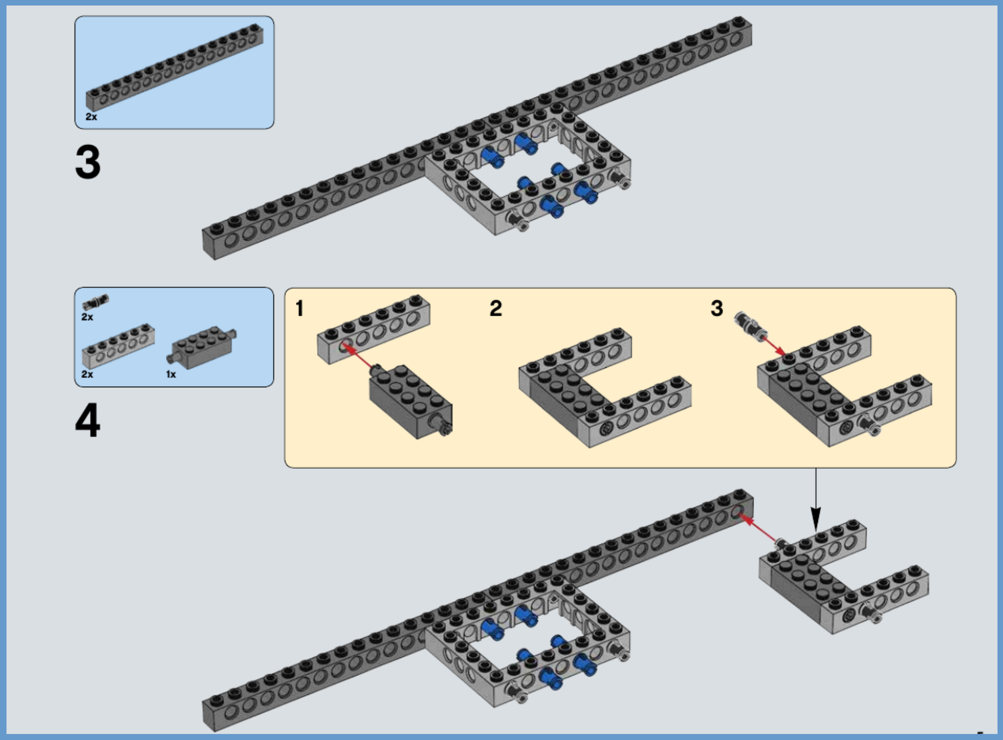
we do not have
all the pieces!
i.e., spatial frequencies

Amplitude

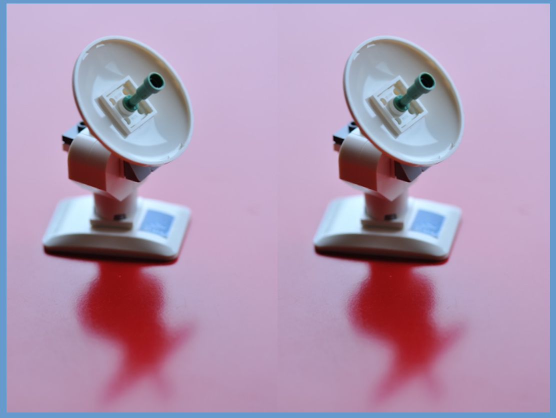


FT

Phase



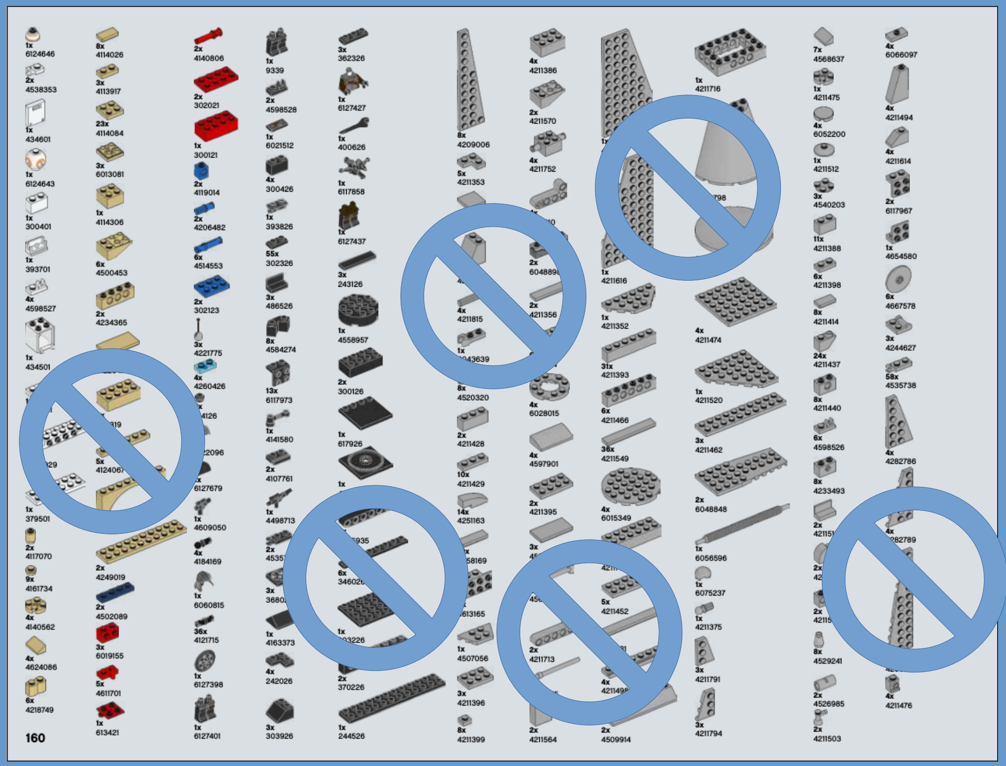
Compact configuration:
= not seeing small pieces
= low resolution



Problem:

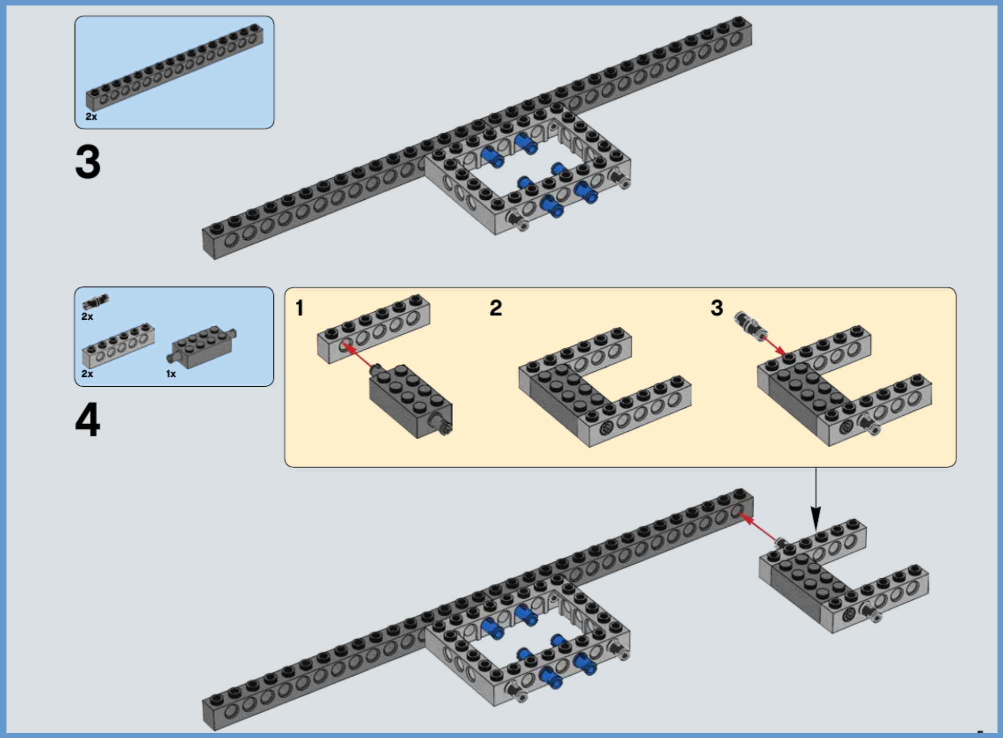
we do not have all the pieces! i.e., spatial frequencies

Amplitude



FT

Phase

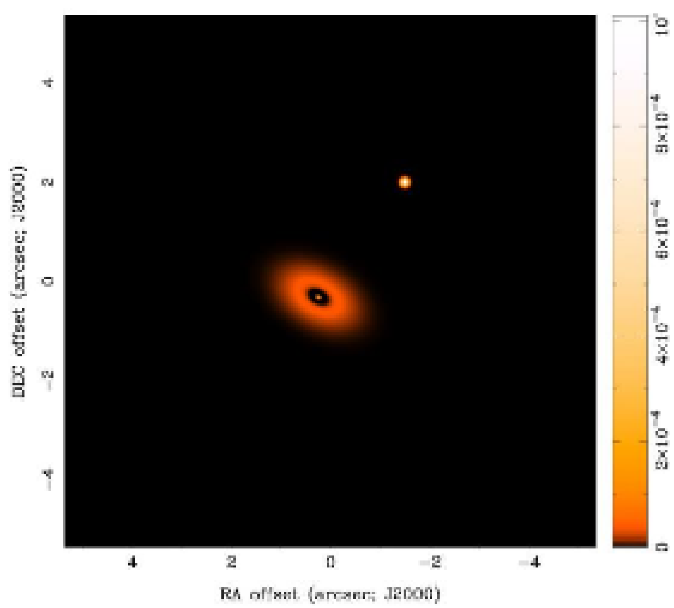


Extended configuration = loosing big pieces = not seeing large structures

The grain of salt, the onion ring,

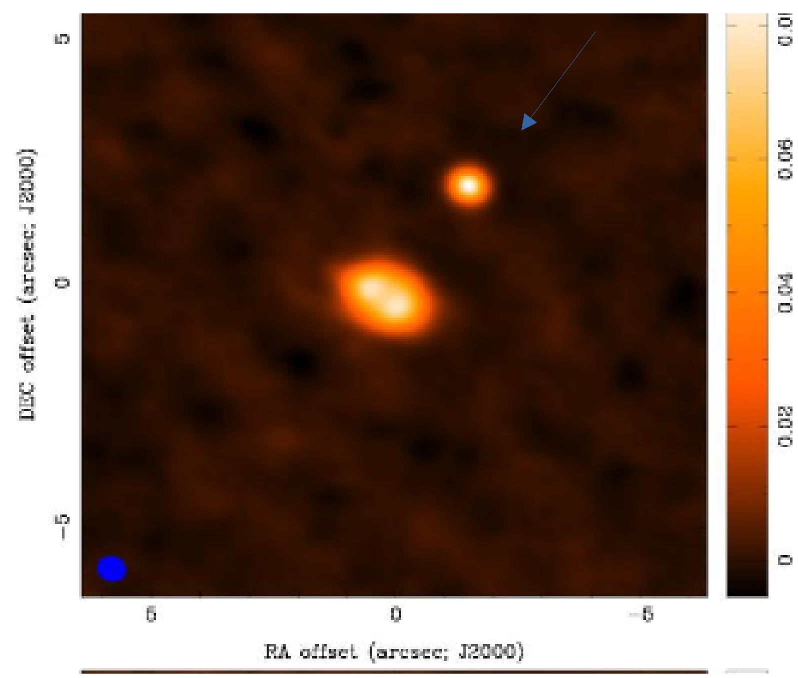
1

Model source

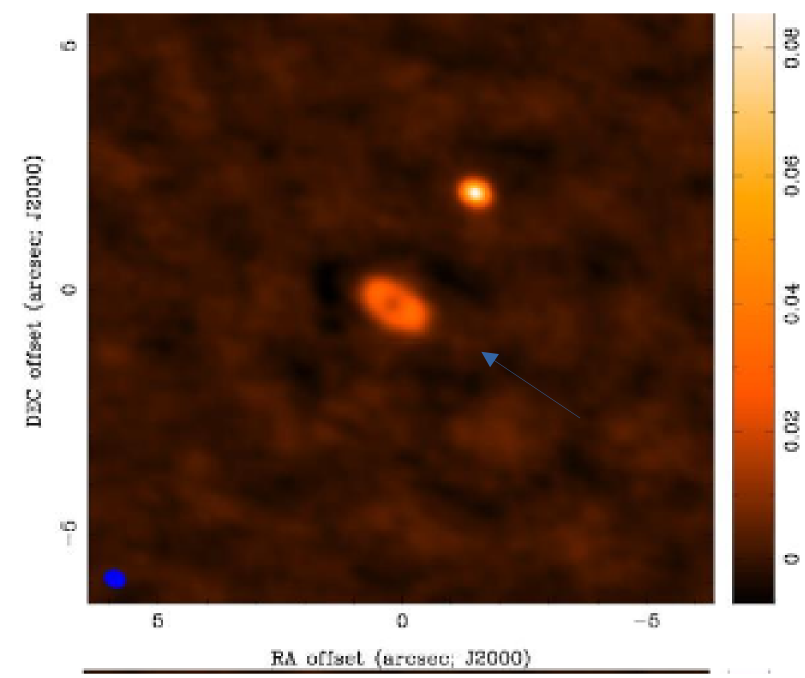


Compact configuration

and



Excluding short baselines

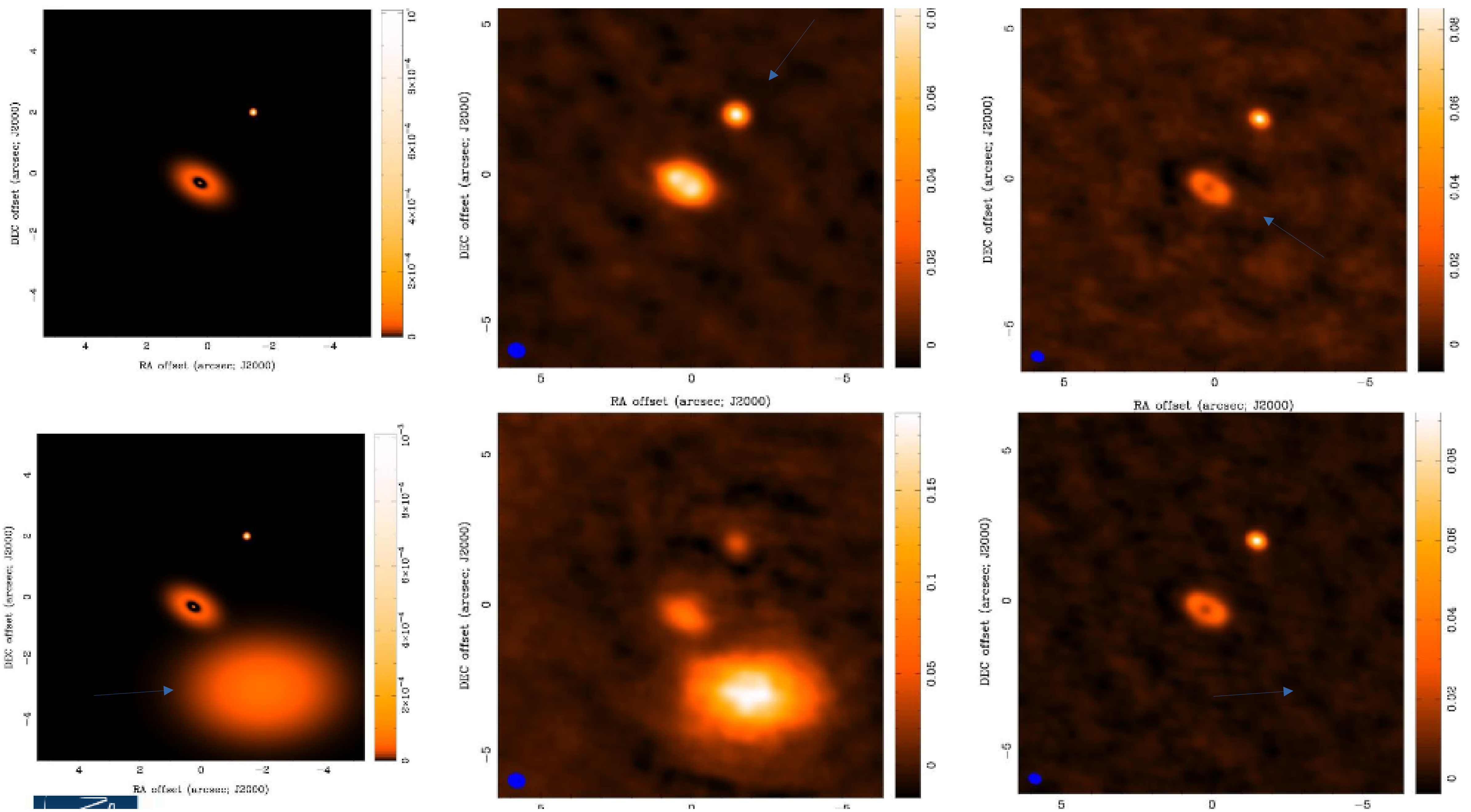


The grain of salt, the onion ring, and the potato

Model source | Compact configuration | and | Excluding short baselines

1

2



Good to do before/after imaging

- Check the quality of the visibilities (weblog, plotms, amplitudes, phases,...)
- Check the observations details (listobs: field names, spectral setup, …)
- Check the check source (if exists)

- Check the residuals
- Check beam resolution (~lambda/Bmax)
- Compare with previous images of the source
- Check potential missing flux issues
- ...

- Write and publish your paper

Imaging with an interferometer: a brief summary

- Interferometers sample the **Fourier transform** of the intensity distribution of the source/sky

- Limited number of antennas: partial coverage of source **spatial frequencies**

- The deconvolved image from an interferometer is a **model** compatible with the visibilities

- Remember the limits of interferometric observations

Quick quiz

- Should the position of the antennas be optimized to best sample the uv plane?

- Does an interferometer have problem to map extended sources?

- Can an interferometer ever measure the total flux of a source?

- Is the short spacing filtering always a problem?

Some references:

+ I-TRAIN video tutorials:

https://almascience.eso.org/tools/eu-arc-network/i-train

I-TRAIN #1: Imaging with the ALMA Pipeline
I-TRAIN #17: Introduction to tclean

+ ALMA science portal (ALMA Technical Handbook):

https://almascience.eso.org

+ Materials from previous schools (ERIS, IRAM, NRAO, ...)

+ CASA Guides
+ Automasking Guide CASA 6.5.4:
https://casaguides.nrao.edu/index.php?title=Automasking_Guide_CASA_6.5.4