ALMA Imaging

with CASA

Sheng-Jun Lin (ASIAA Postdoc Fellow/ARC Member),
Kuo-Song Wang (ASIAA), and ARC Taiwan node

ALMA imaging workshop @ ASIAA 2025/02/13

Goals of this talk

- Gain some intuition for interferometric observations
  - What is the visibility?
- Understand the imaging process
  - What is the CLEAN algorithm?
- Learn the `tclean` task in CASA
  - Hands-on session (this afternoon)

Interferometry basics

Single dish vs. Interferometry: What's the Beam about?

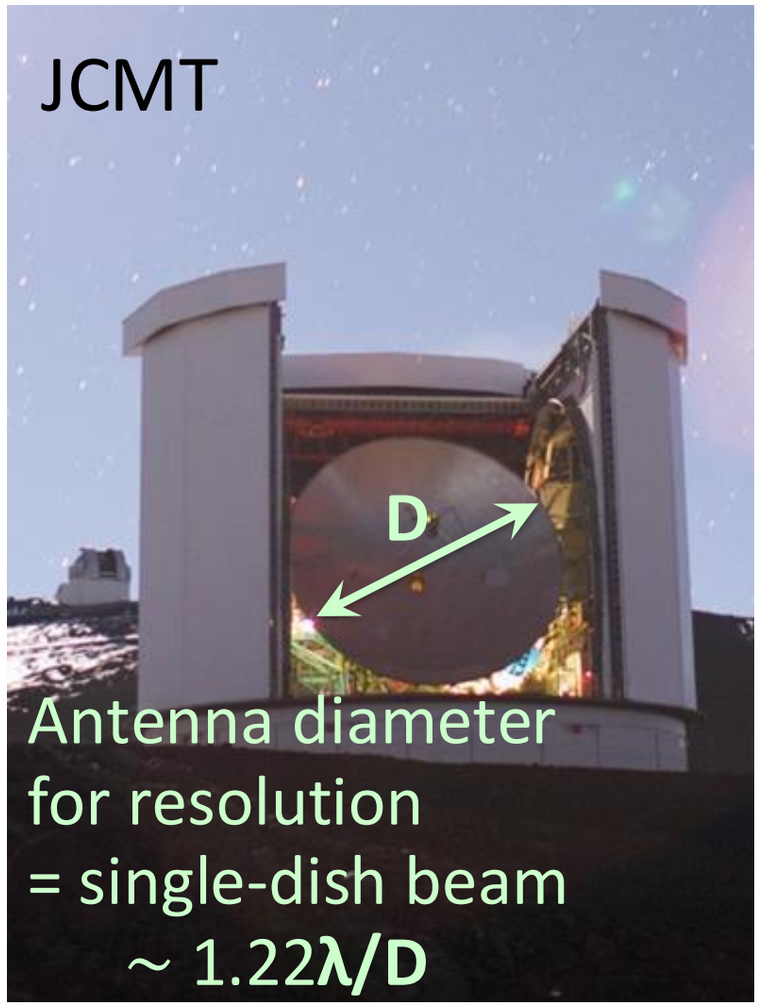
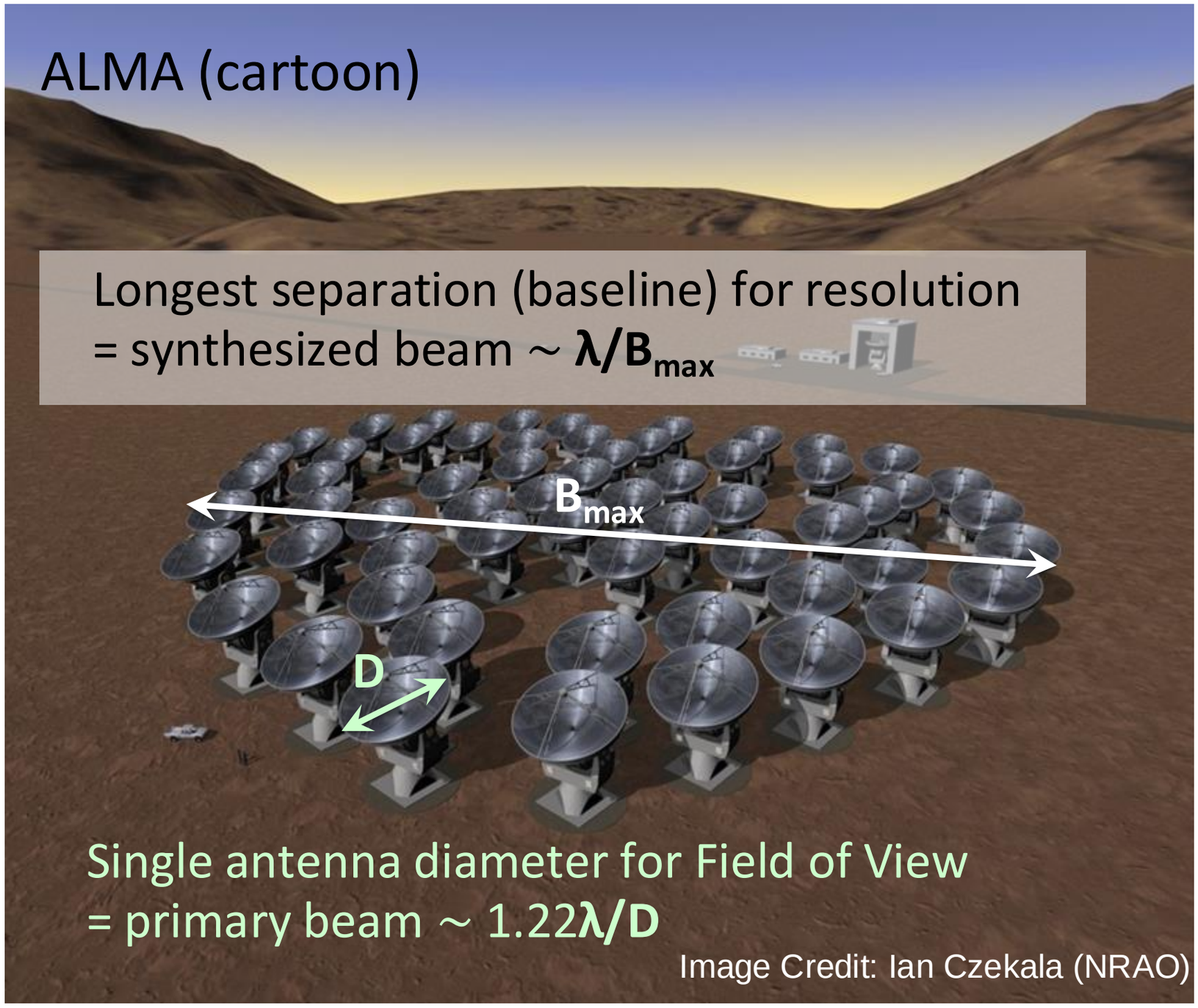


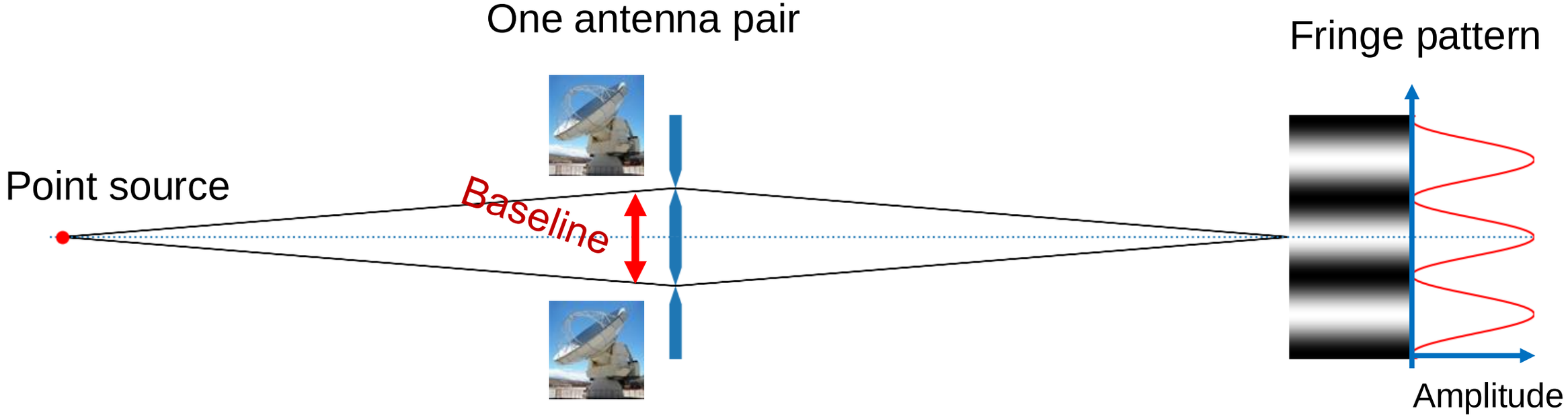
Image Credit: William Montgomerie (EAO)



Interferometry basics

Young’s double-slit experiment

An interferometer measures the **interference pattern (fringe)** observed by pairs of antennas.



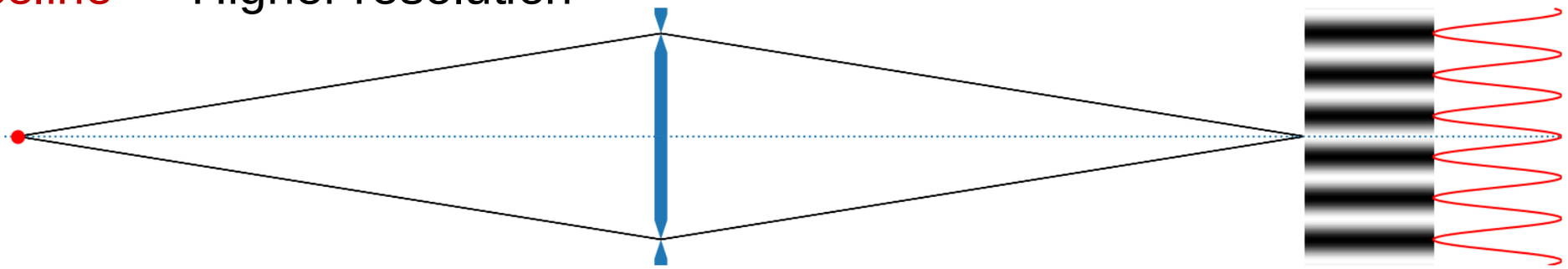
Code Credit: Oleg Smirnov (SKA SA)

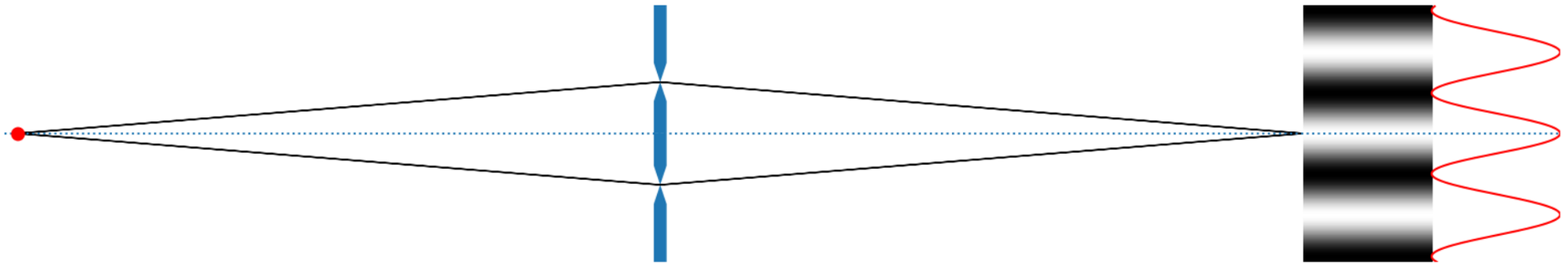
Interferometry basics

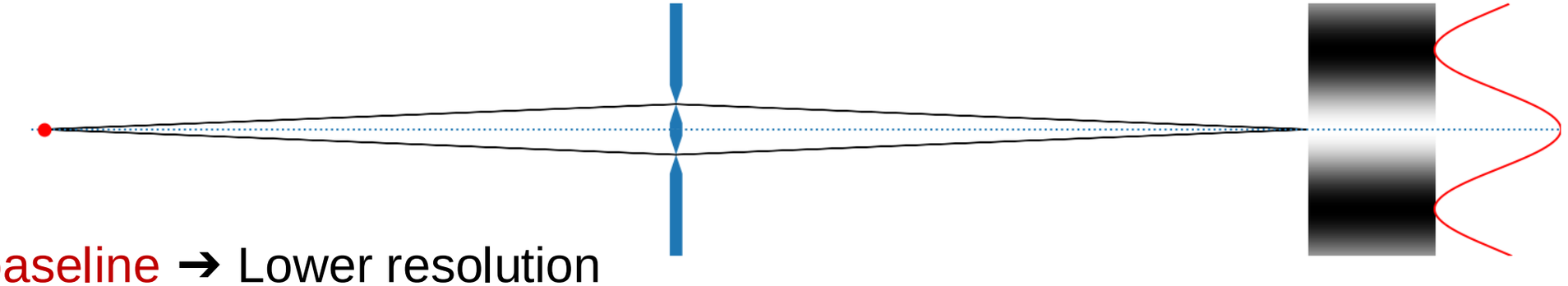
Young's double-slit experiment

Each baseline sees one Fourier component of the sky brightness with one **spatial frequency (resolution)**

Longer baseline ➔ Higher resolution







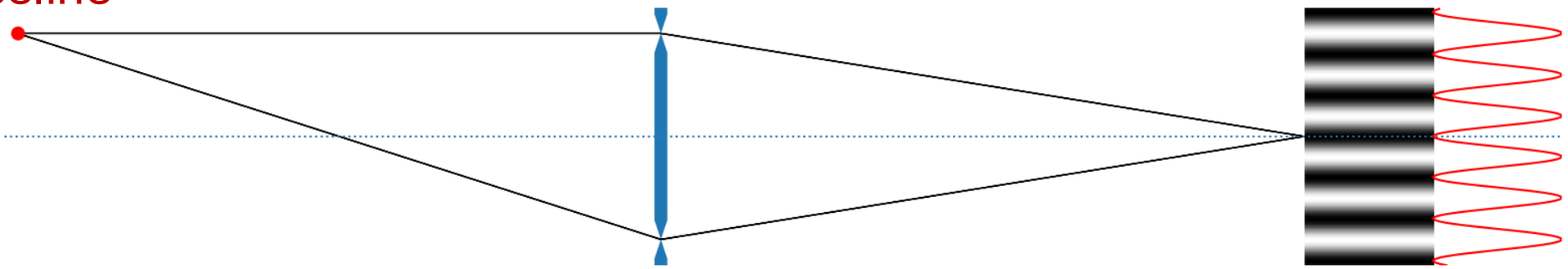
Shorter baseline ➔ Lower resolution

Interferometry basics

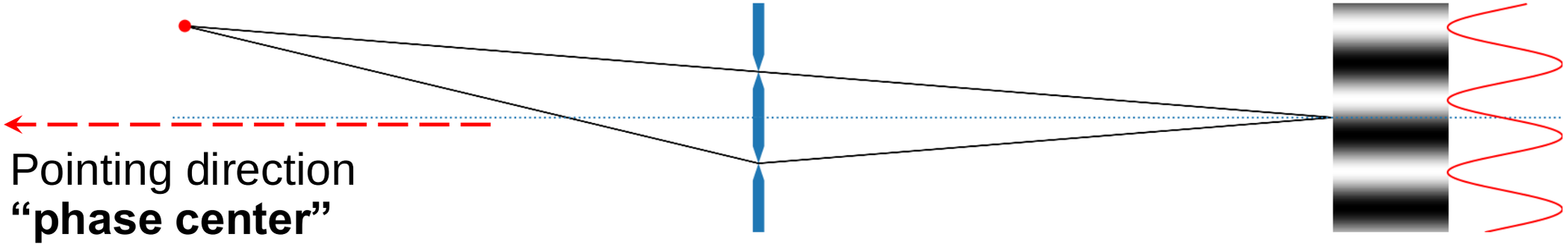
Young's double-slit experiment

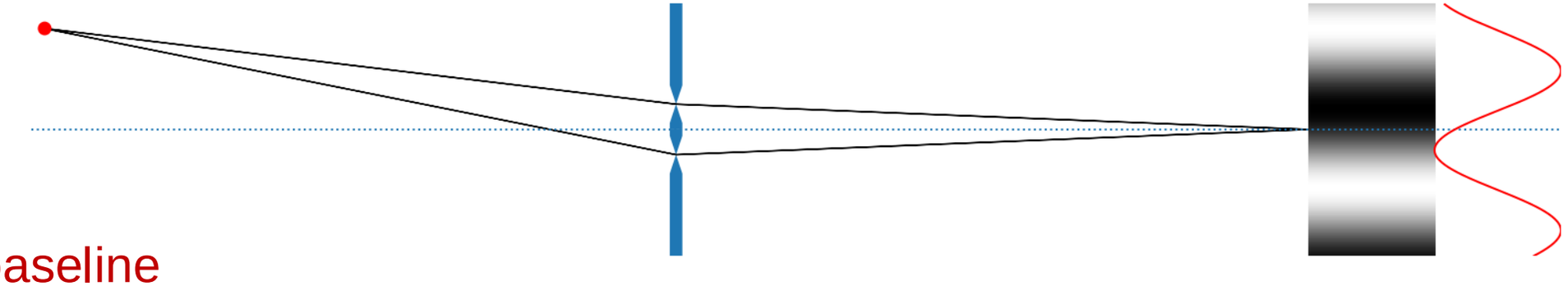
Same **source position** relative to the pointing direction results different **phases** seen by different antenna pairs (baselines)

Longer baseline



Offset point source





Shorter baseline

Visibility function

- "Visibility" was originally introduced to describe the contrast of the fringe pattern as $V=(I_{max}-I_{min})/(I_{max}+I_{min})$.

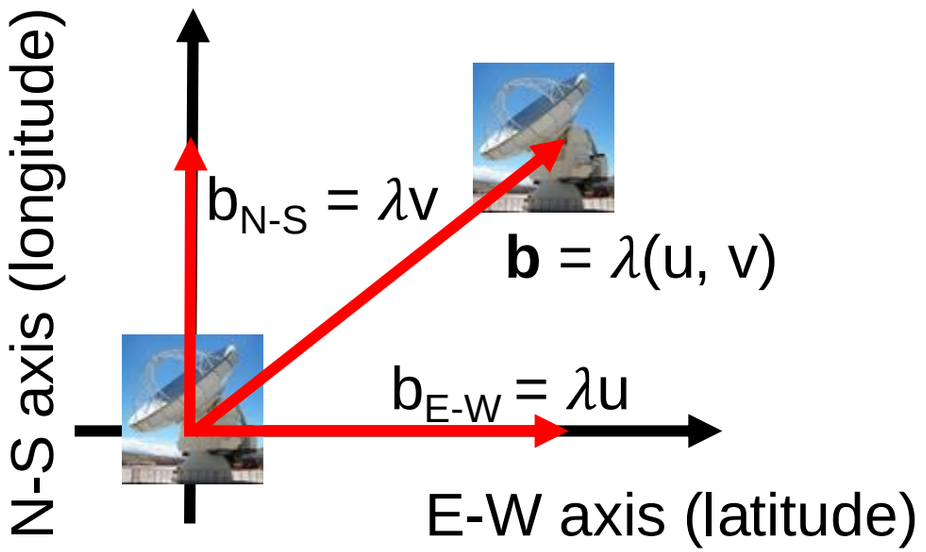
- Radio interferometry works with "complex visibilities"

  Let $u = b_{E-W}/\lambda$, and $v = b_{N-S}/\lambda$ be spatial frequencies.

  $$V(u,v) = V_M e^{i2\pi\phi_V} = V_M(\cos 2\pi\phi_V + i \sin 2\pi\phi_V).$$

  – Fringe amplitude → Visibility amplitude
  Fluxes from a spatial frequency (u, v)

  – Fringe phase → Visibility phase
  "*Location" of a spatial frequency (u, v)*



Visibility function

van Cittert-Zernike theorem

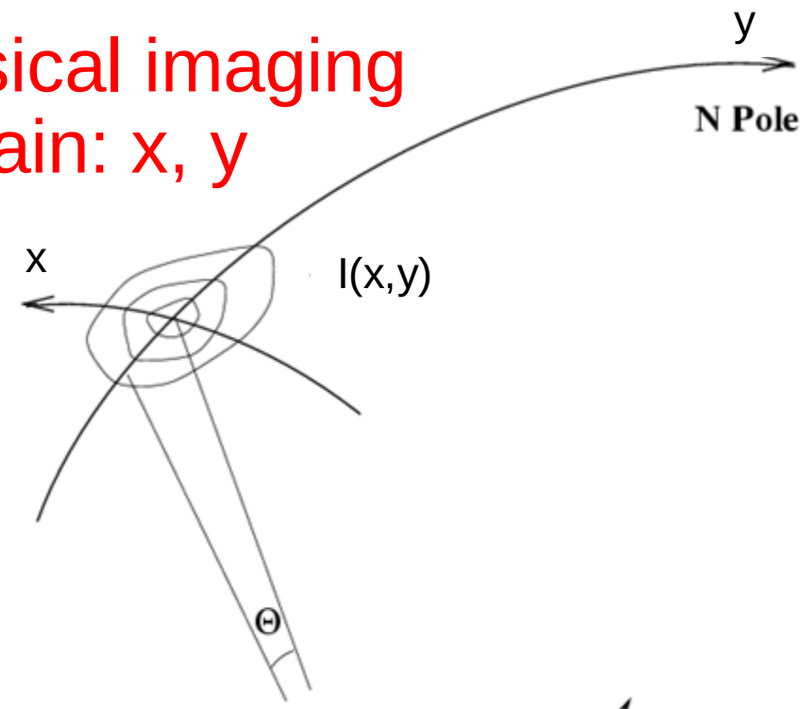
The interference pattern is directly related to the source brightness.
The **complex visibility, *V(u,v)*,** is the 2D Fourier transform of the **sky brightness, *I(x,y)*.**

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

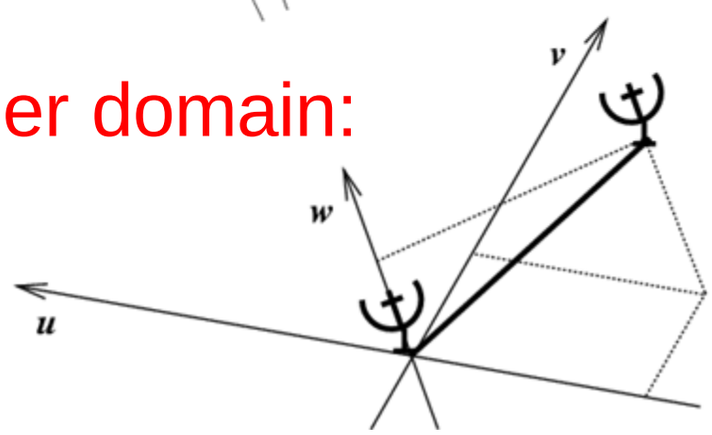
$$I(x,y) = \iint V(u,v) e^{+i2\pi(ux+vy)} dudv$$

u, v are E-W, N-S spatial frequencies [wavelengths]
x, y are E-W, N-S angles in the tangent plane [radians]

Physical imaging domain: x, y



Fourier domain: u, v

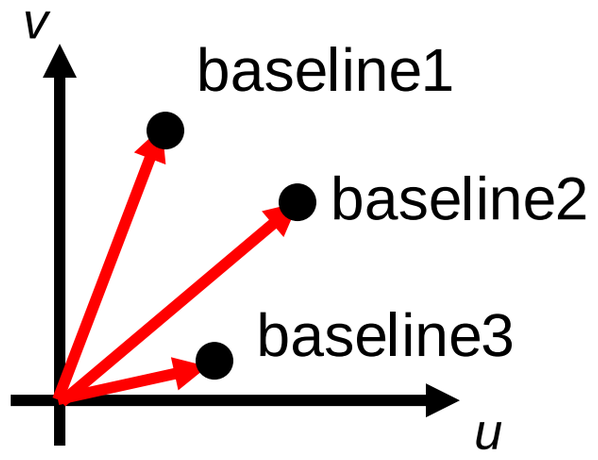


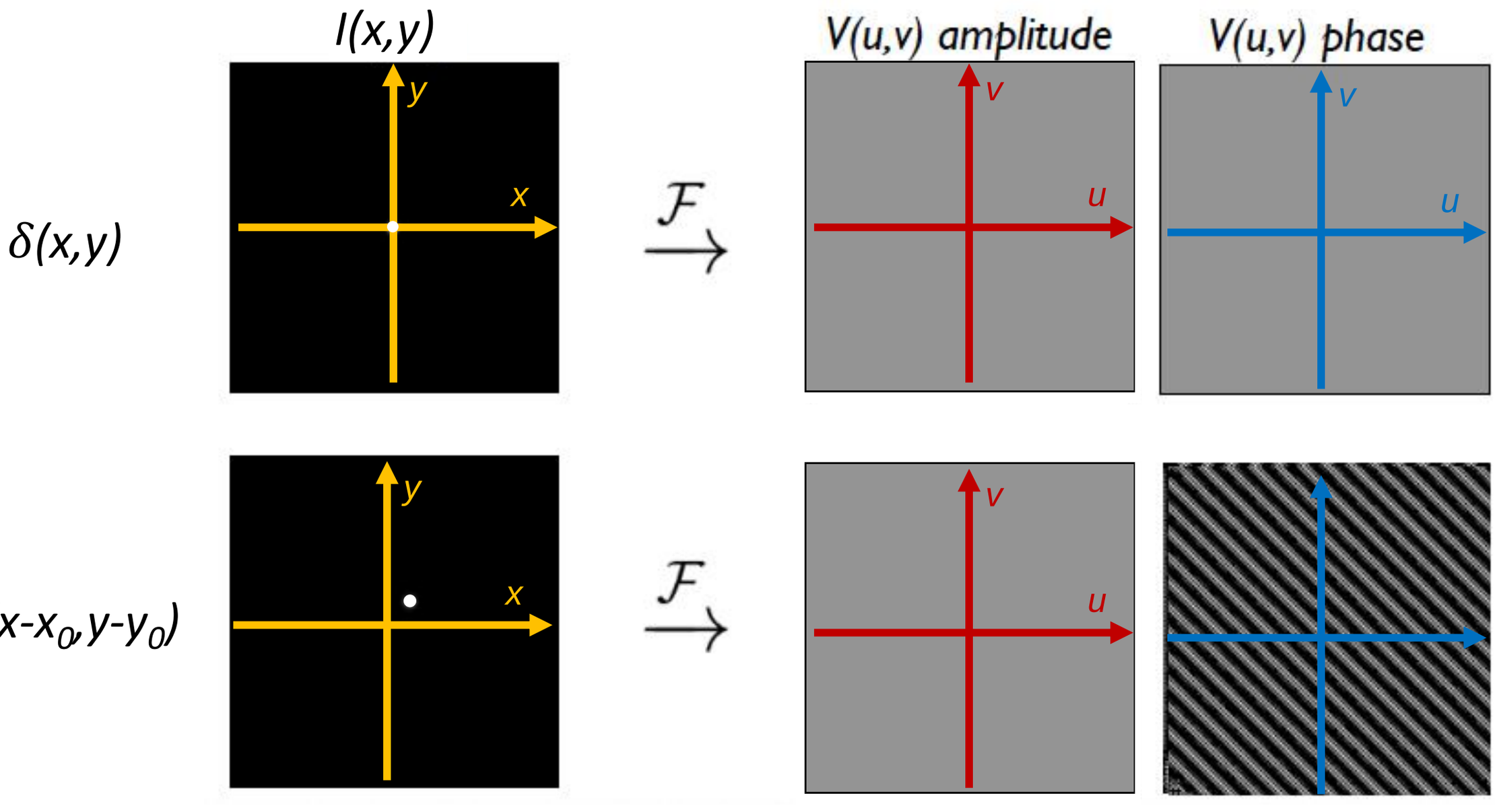
Fourier transform

2D case

$\mathcal{F}$ properties:

- $\mathcal{F}(\delta)$ = constant for amplitude



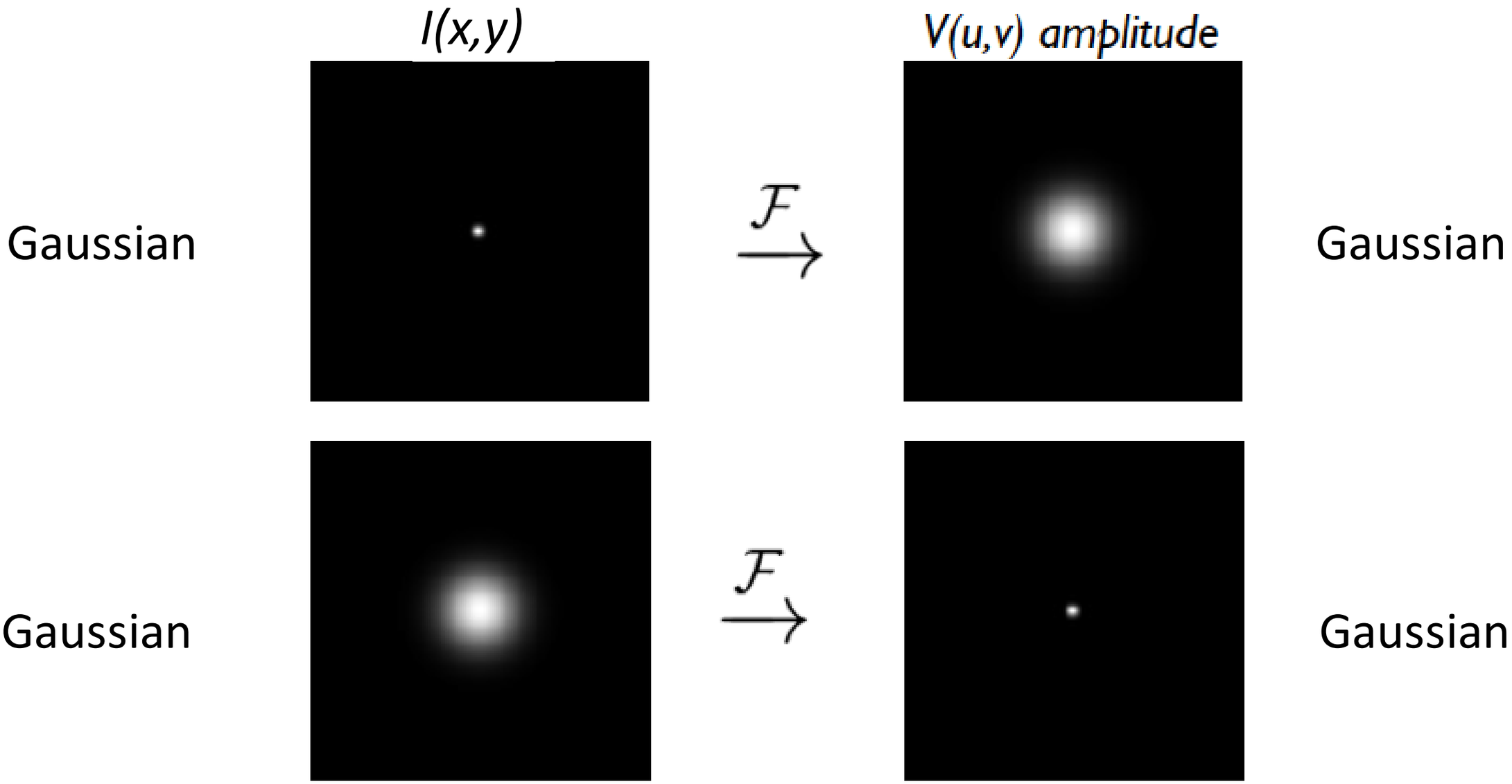


Fourier transform

2D case

$\mathcal{F}$ properties:

- $\mathcal{F}(Gaussian) = Gaussian$
- Narrow features transform into wide features and vice versa.

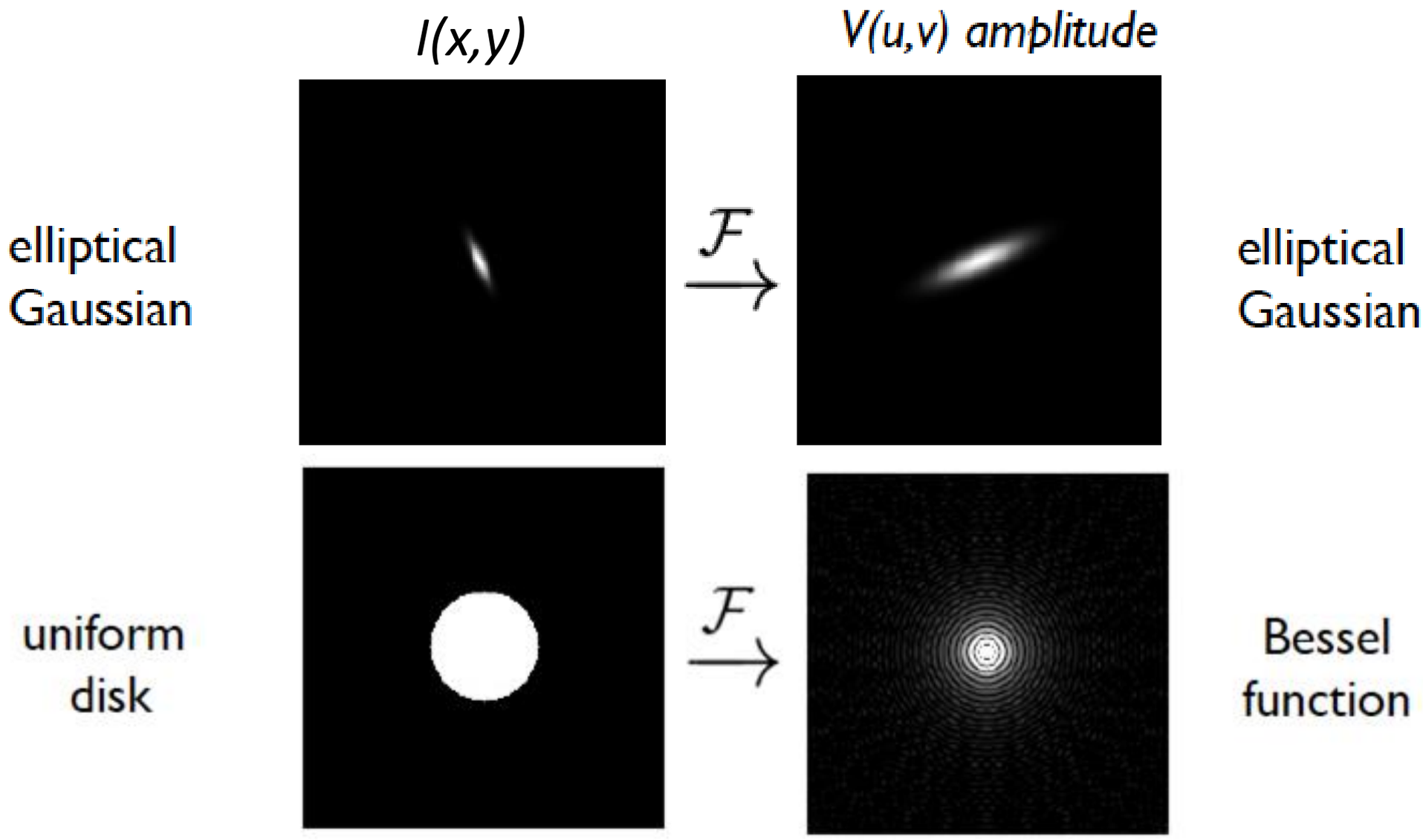


Fourier transform

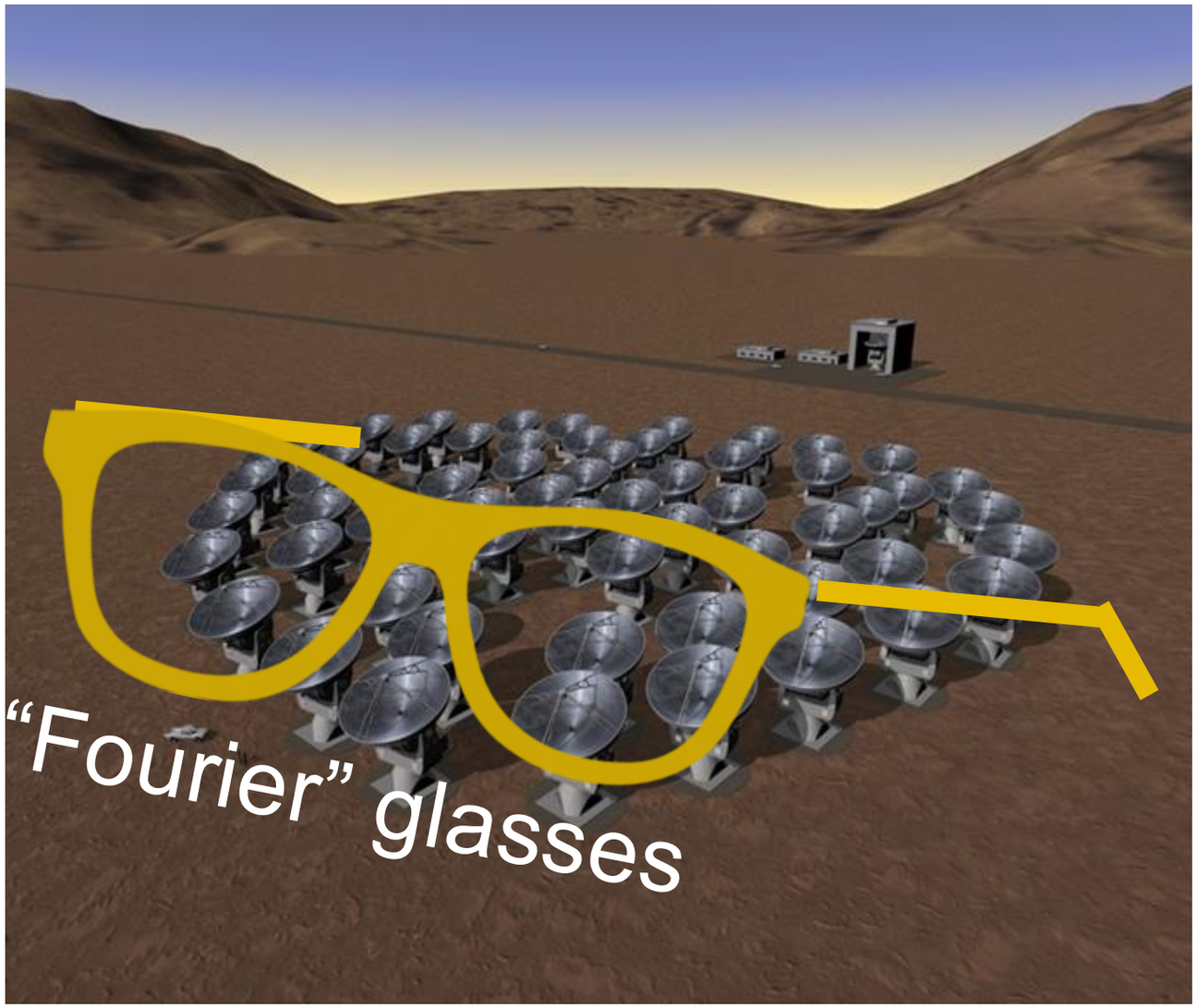
2D case

$\mathcal{F}$ properties:

- Sharp edges produce “ringing” features and vice versa.



Interferometric observations



So we have to think about our data in the Fourier domain!

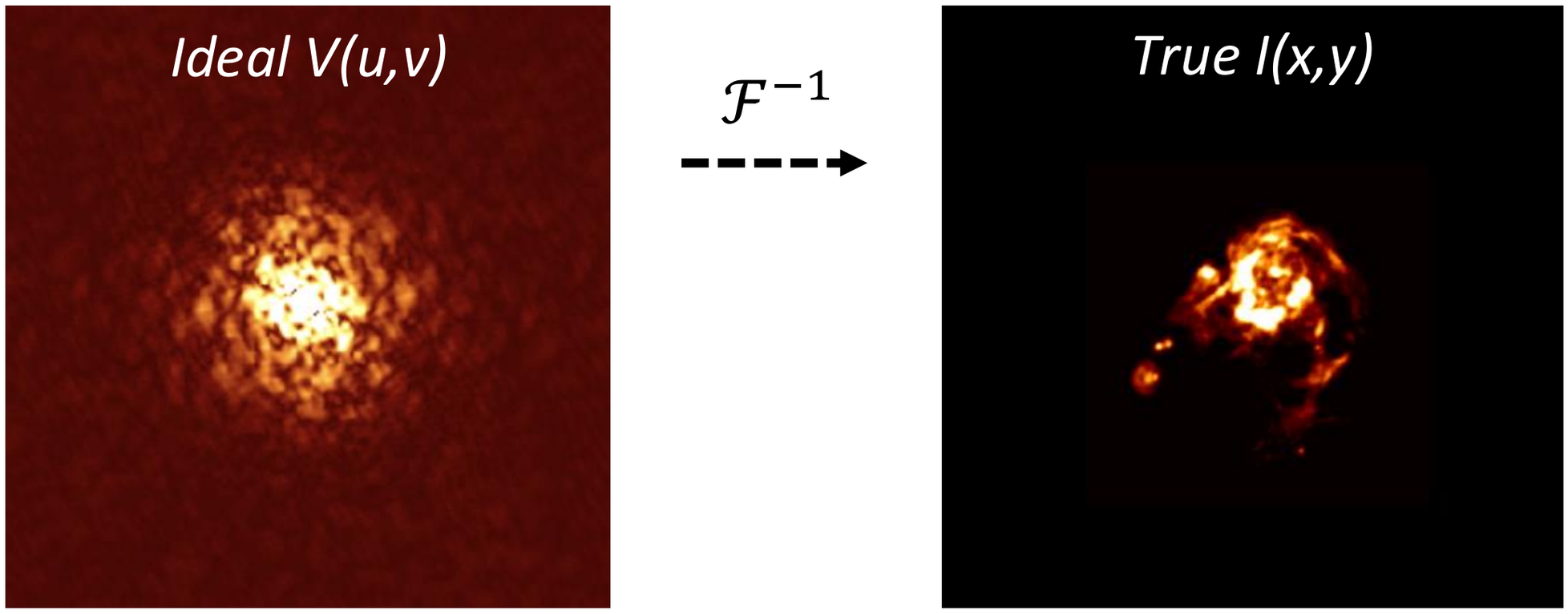
Goals of this talk

- Gain some intuition for interferometric observations
  - What is the visibility?
- Understand the imaging process
  - What is the CLEAN algorithm?
- Learn the `tclean` task in CASA
  - Hands-on session (this afternoon)

Interferometric observations

Imaging the visibility data

- CASA plotms task: View the visibility data
- CASA tclean task: Image the visibility data
- Ideally, we obtain the image by $\mathcal{F}^{-1}[V(u,v)] = I(x,y)$

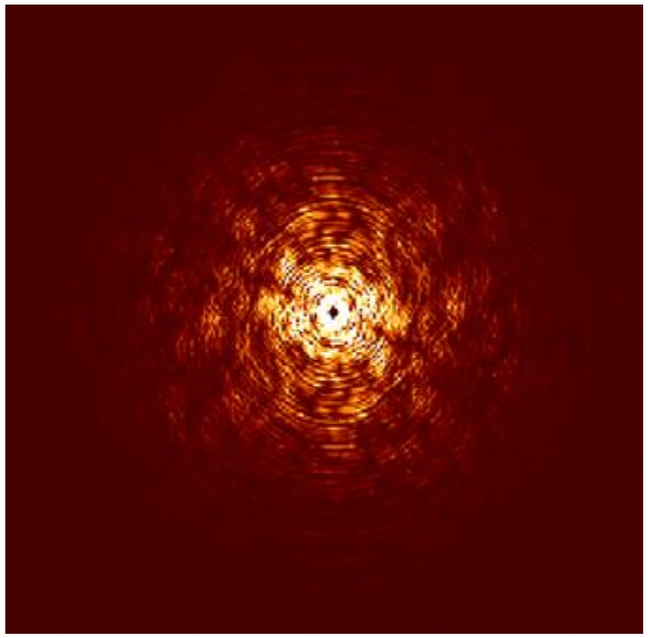


Interferometric observations

V(u, v) is not measured for all (u, v)…

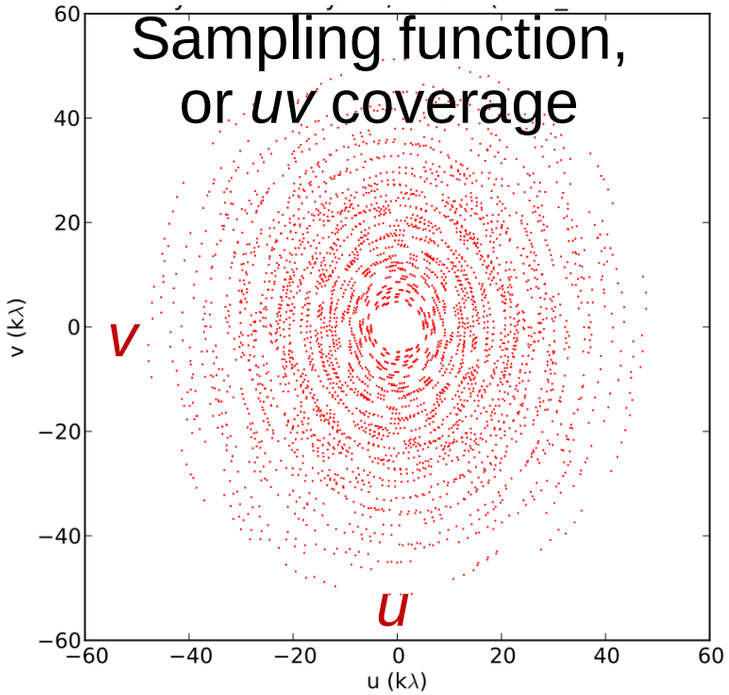
- Practically, $V_{\mathrm{obs}}(u,v) = S(u,v)V(u,v)$

Measured $V_{obs}(u,v)$



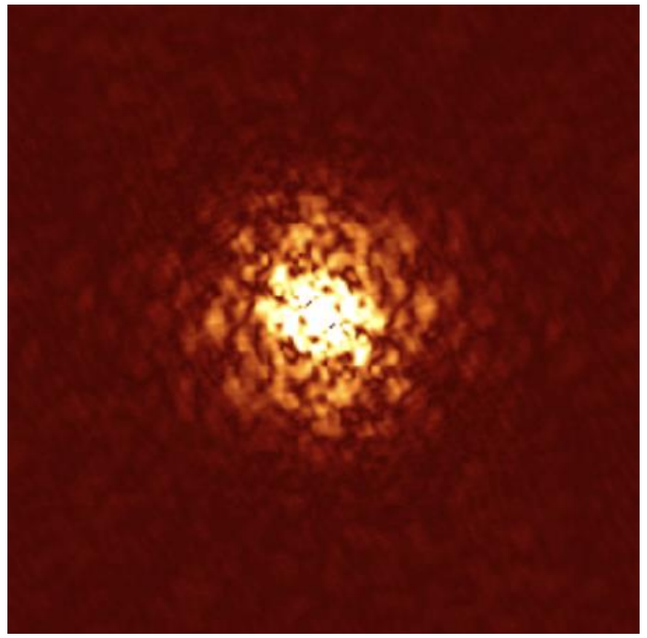
=

S(u,v)



×

Ideal V(u,v)

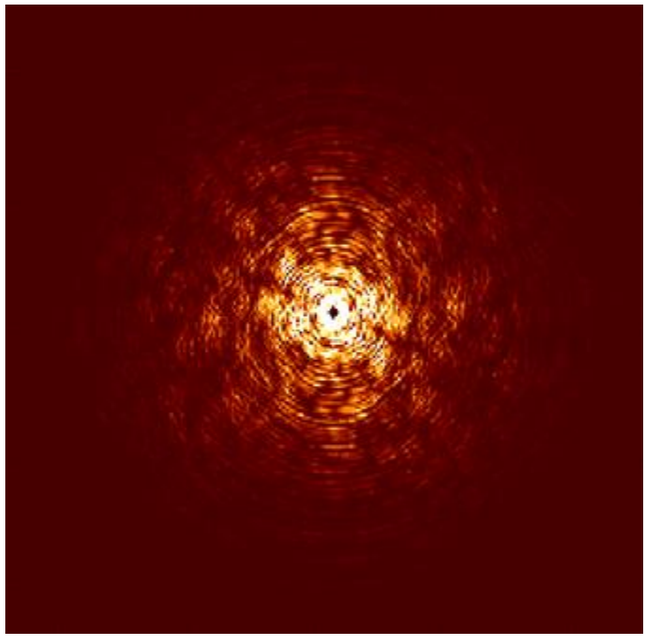


Interferometric observations

V(u, v) is not measured for all (u, v)…

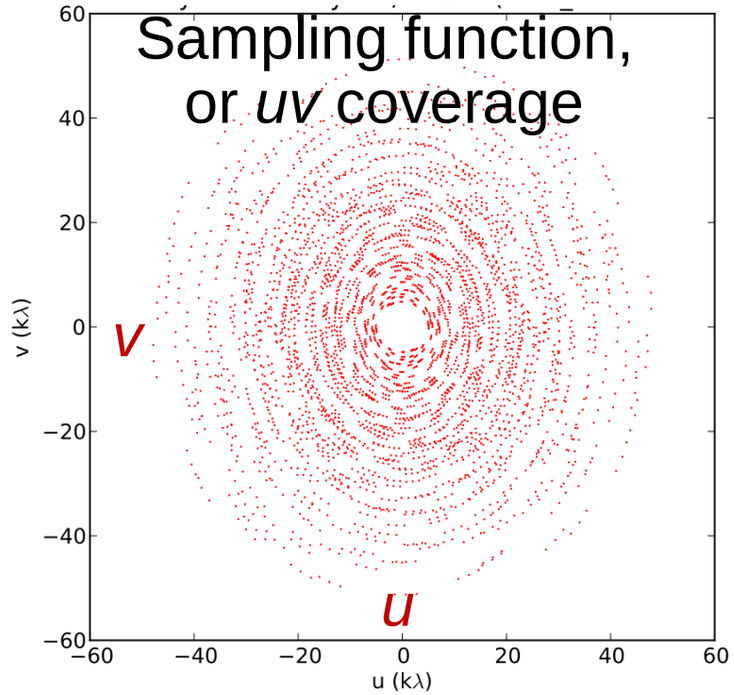
- Practically, $V_{\mathrm{obs}}(u,v) = S(u,v)V(u,v)$

Measured $V_{obs}(u,v)$



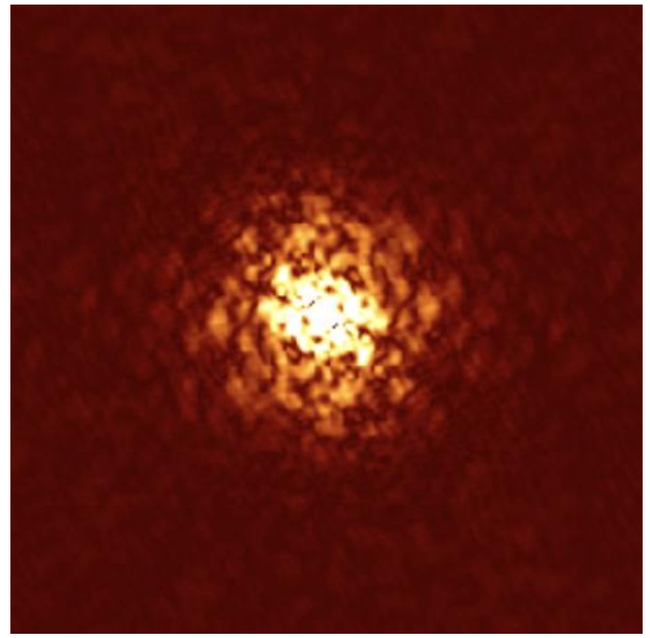
=

S(u,v)



×

Ideal V(u,v)



$$\Rightarrow \mathcal{F}^{-1}[V_{\mathrm{obs}}(u,v)] = \mathcal{F}^{-1}[S(u,v) \times V(u,v)]$$
$$= \mathcal{F}^{-1}[S(u,v)] * \mathcal{F}^{-1}[V(u,v)]$$

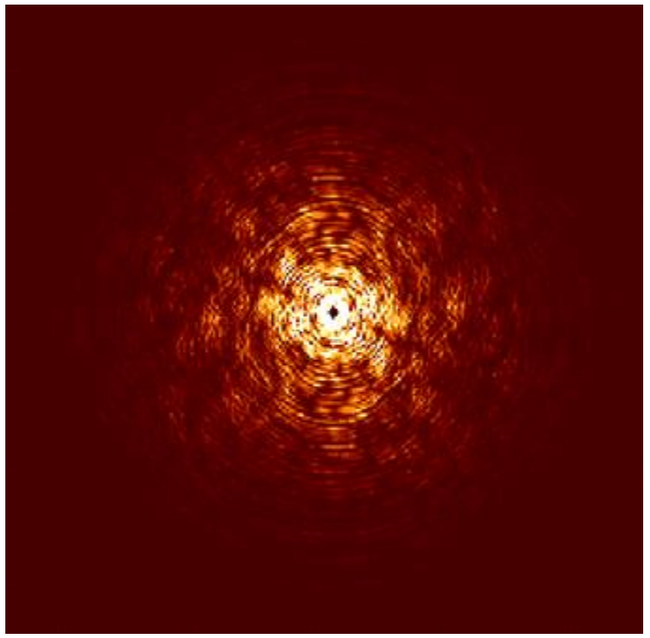
Convolution

True I(x,y)

S(u, v) is not sampled in a regular grid for V(u,v) either!

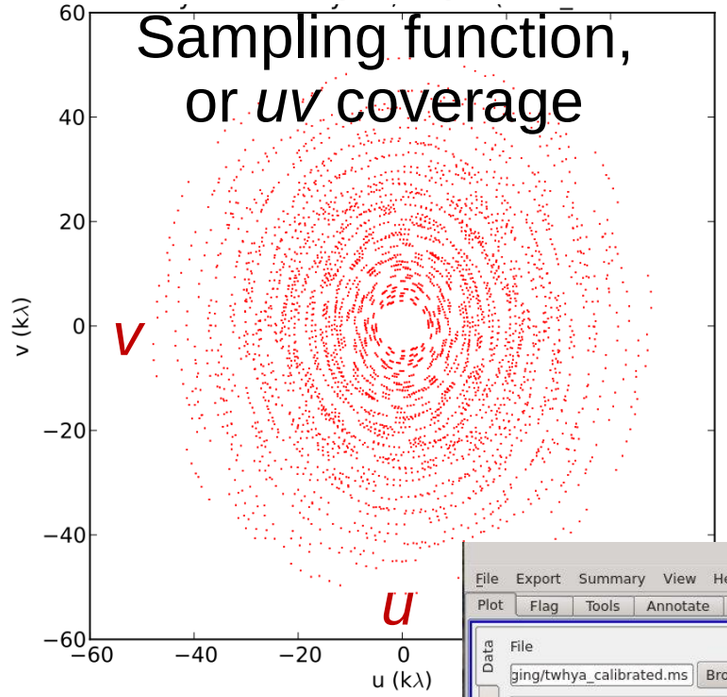
Interferometric observations

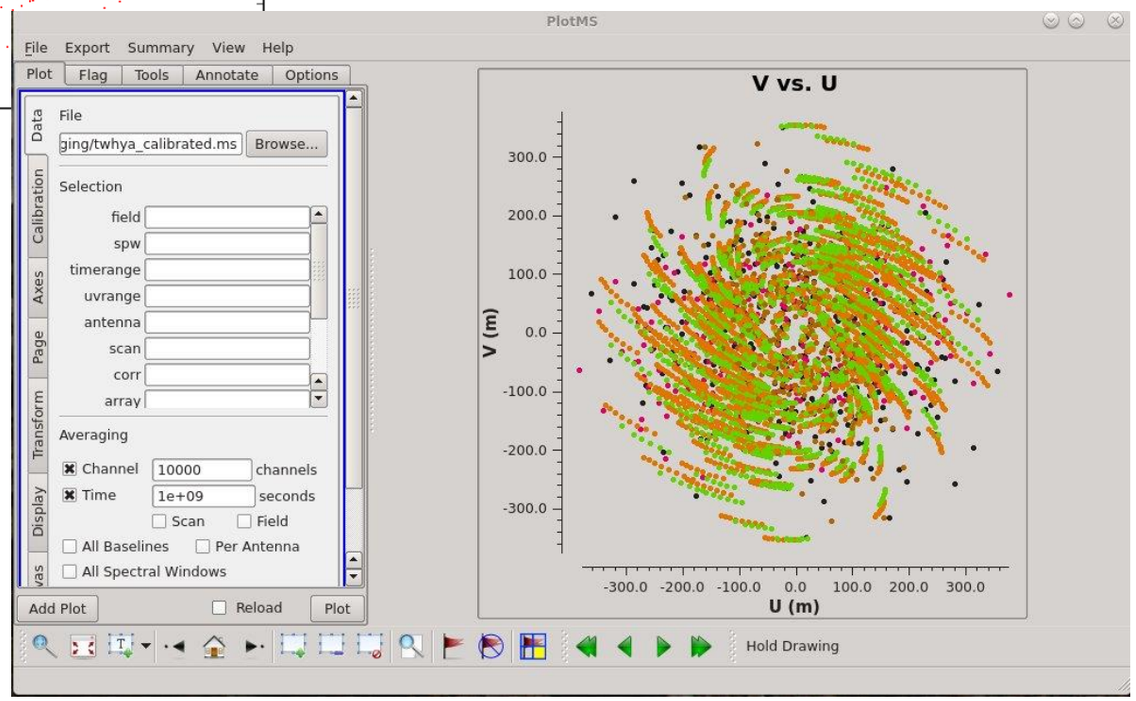
Measured $V_{obs}(u,v)$



S(u,v)

Sampling function,
or *uv* coverage





- CASA plotms task: View the visibility data

Imaging parameters

Handling the uneven visibility sampling

Fourier *uv* domain	Image *xy* domain

- Weighting $S(u,v)$ for re-sampling

  tclean/ "weighting" parameter

- Gridding $V_{\mathrm{obs}}(u,v)$ and $S(u,v)$ for FFT

  tclean/ "gridder" parameter

- Pixel size of the final image

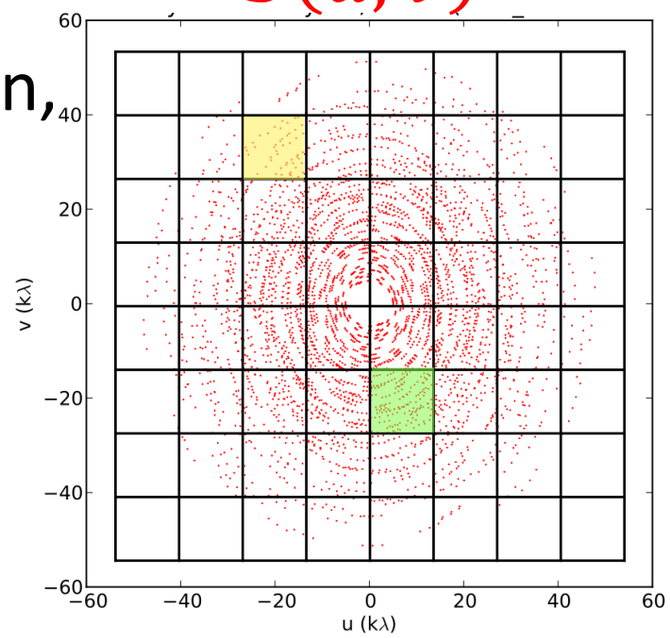
  tclean/ "cell" parameter

- Final image size

  tclean/ "imsize" parameter

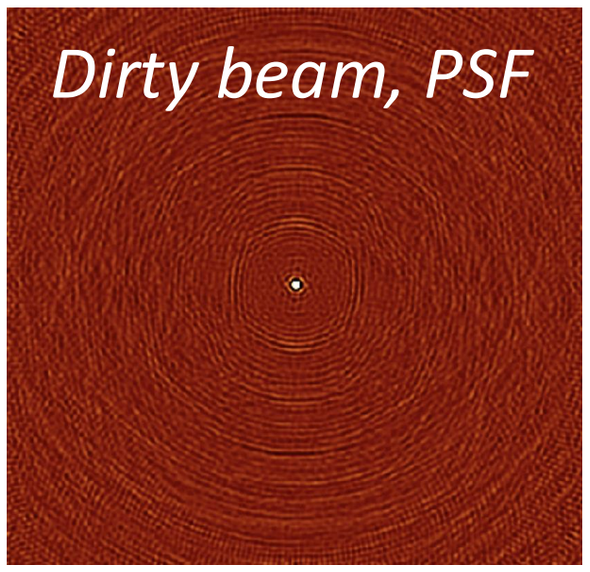
Sampling function, or *uv* coverage

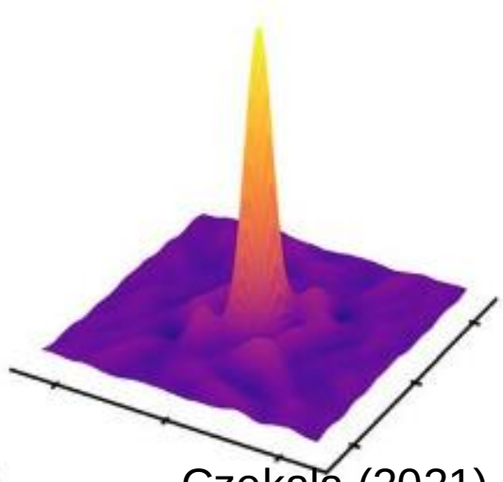
$S(u,v)$



$\mathcal{F}^{-1}$

$\mathcal{F}^{-1}[S(u,v)] = B_{dirty}(x,y)$





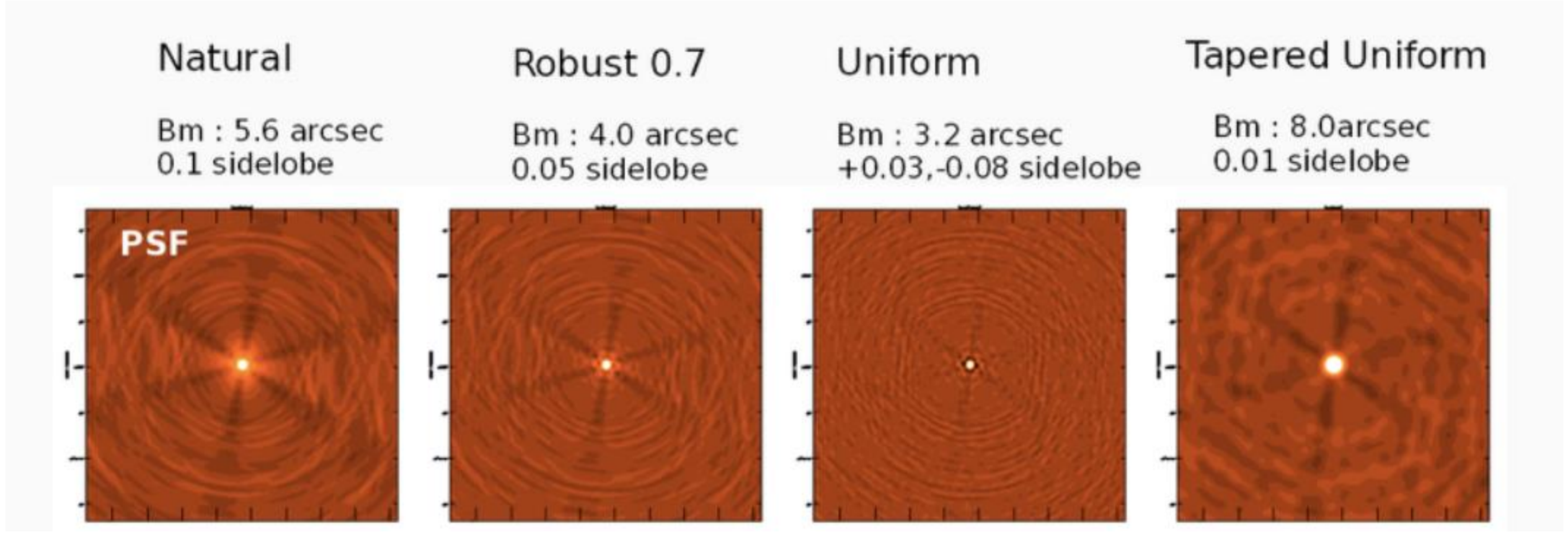
Czekala (2021)

Imaging parameters

tclean “weighting” parameter

Visibility weighting schemes

- Natural: Maximizes the sensitivity, degrades angular resolution
- Uniform: Best angular resolution, reduced point source sensitivity
- Robust: Smooth, tunable combination of Natural (robust=+2) & Uniform (-2)
- Taper: Similar to smoothing, degrades resolution

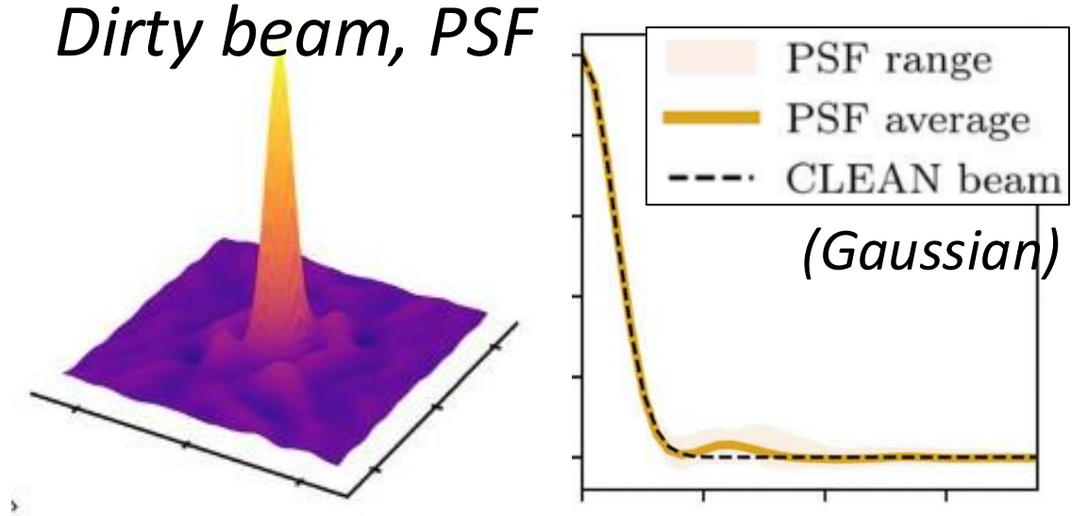


Imaging parameters

tclean “cell” and “imsize” parameters

- EM wave $\lambda = c/\nu$
- Pixel size “cell” satisfies the Nyquist sampling for the longest baseline
  - Beam $\theta \sim \frac{\lambda}{b_{max}}$
  - (>2) 3, 5, or 7 pixels across the **dirty beam** main lobe (=“**clean beam**”)

Dirty beam, PSF



(Gaussian)

Czekala (2021)

- Image size “imsize” ≥ 2 × Field of view (“primary beam”)
  - FOV $\sim 1.22\frac{\lambda}{D}$, where D is the dish diameter (12 m or 7 m)

From Visibility to Image: $\mathcal{F}^{-1}$

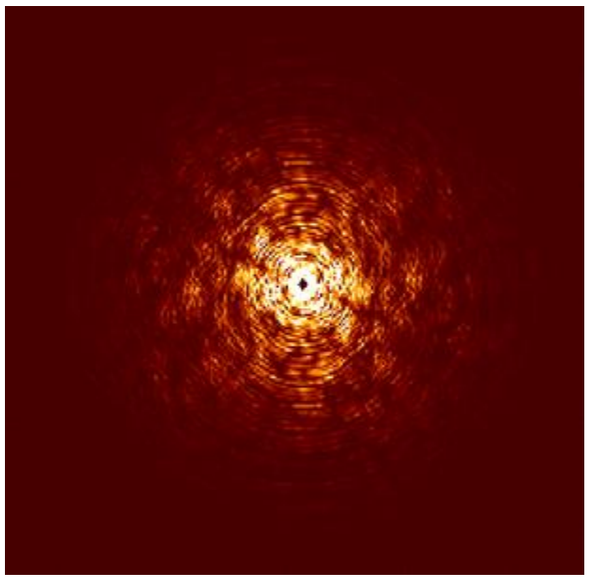
Measured $V_{obs}(u,v)$

S(u,v)

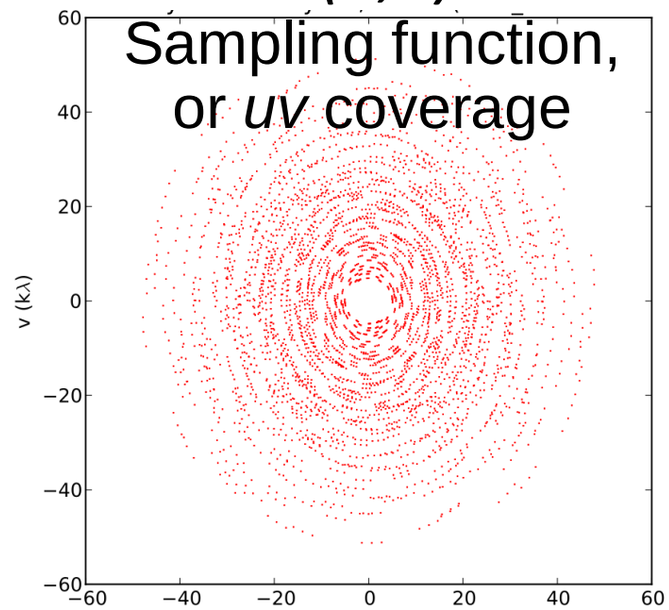
Sampling function, or *uv* coverage

Ideal V(u,v)

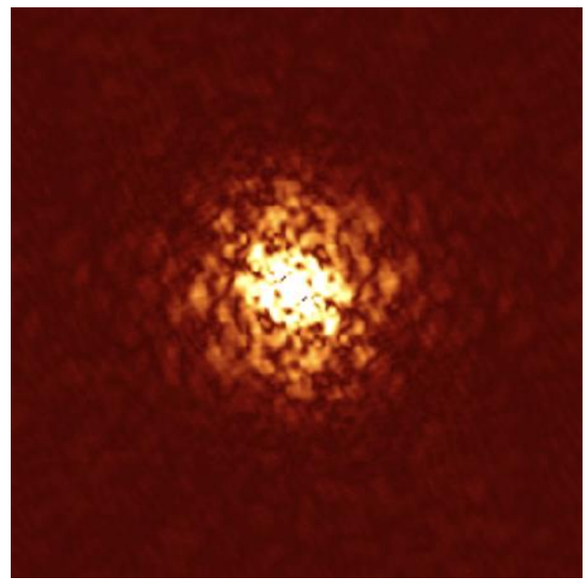
u, v domain:



=



×

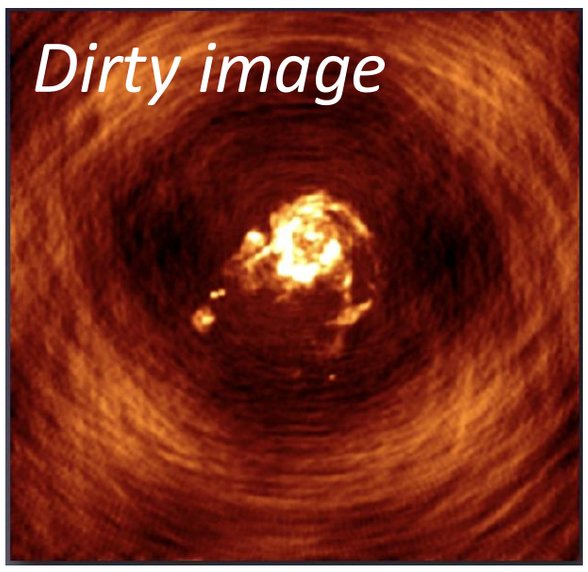


$\mathcal{F}^{-1}[V_{\mathrm{obs}}(u,v)] = I_{dirty}(x,y)$

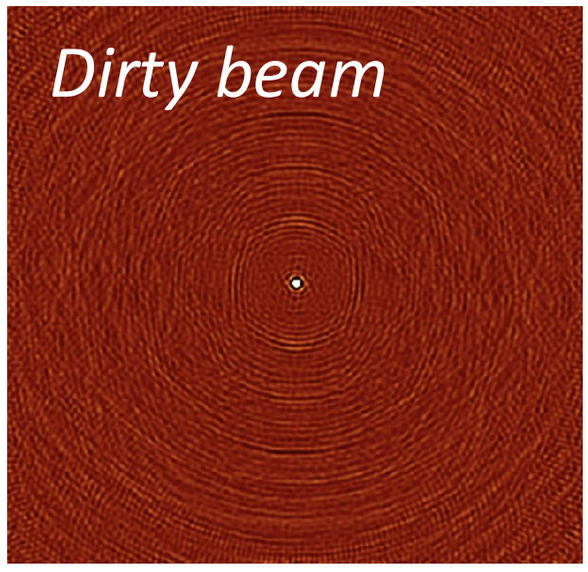
$\mathcal{F}^{-1}[S(u,v)] = B_{dirty}(x,y)$

$\mathcal{F}^{-1}[V(u,v)] = I_{true}(x,y)$

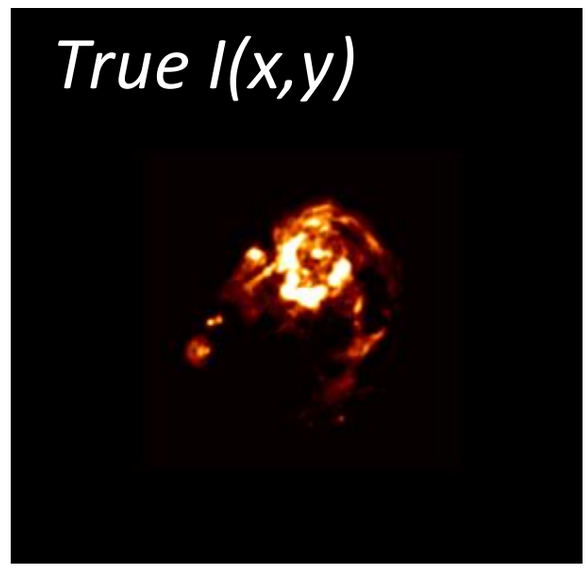
x, y domain:



=



∗

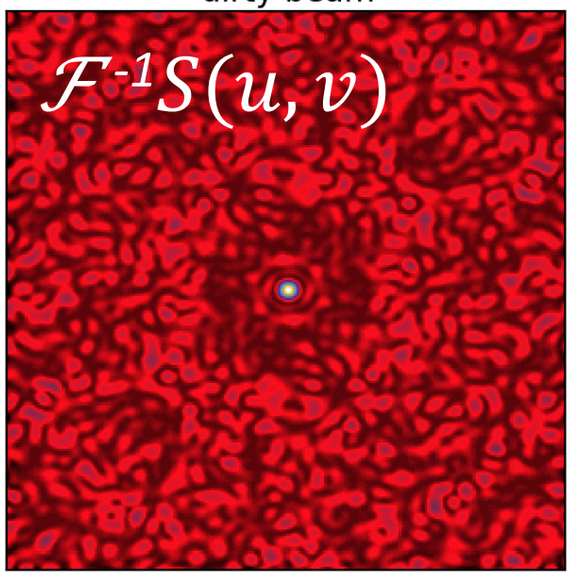


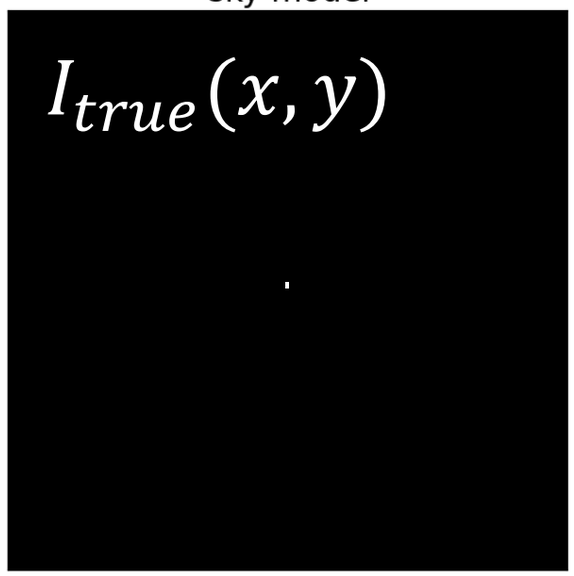
To recover *True I(x,y)*, we must **deconvolve** *Dirty beam* from *Dirty image*.

Simulated visibilities and images

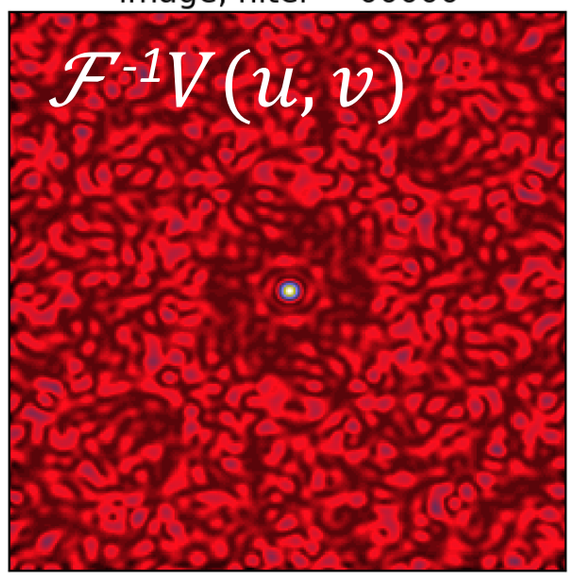
Point source

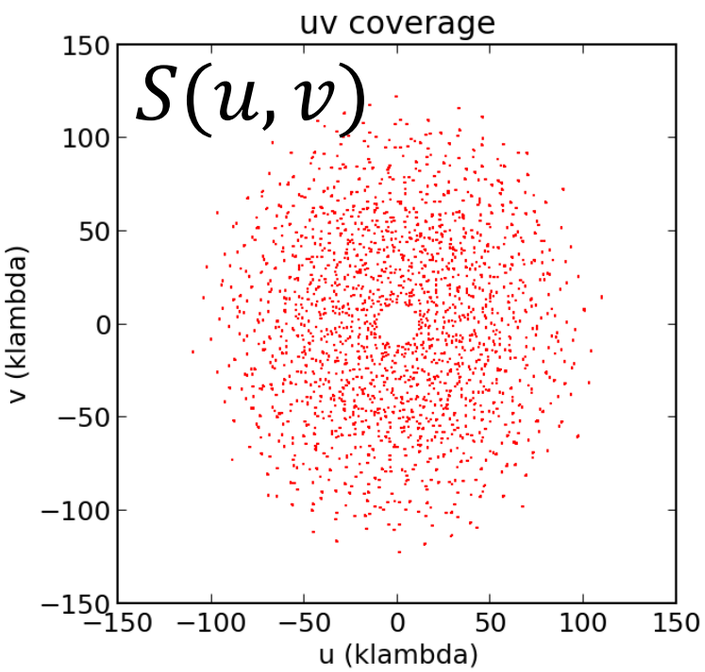
Dirty beam

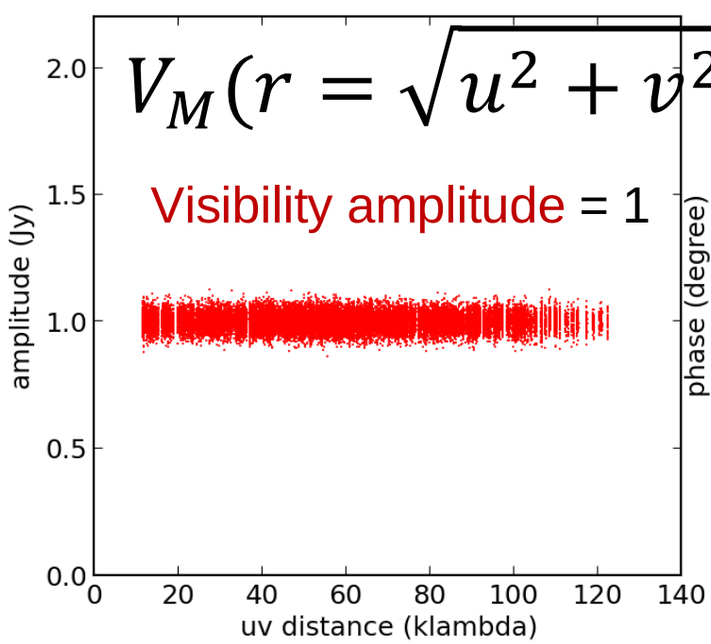


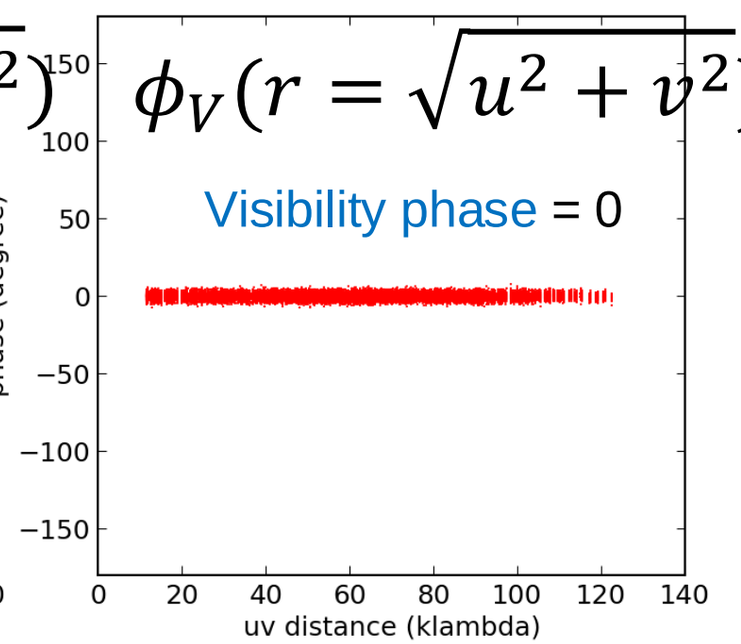


Dirty image







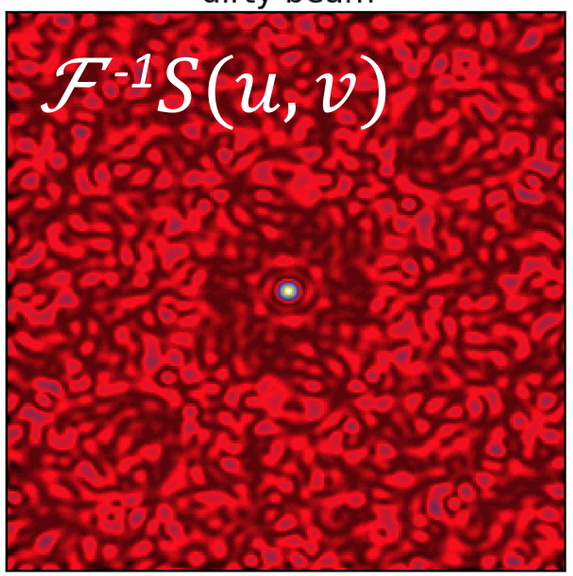


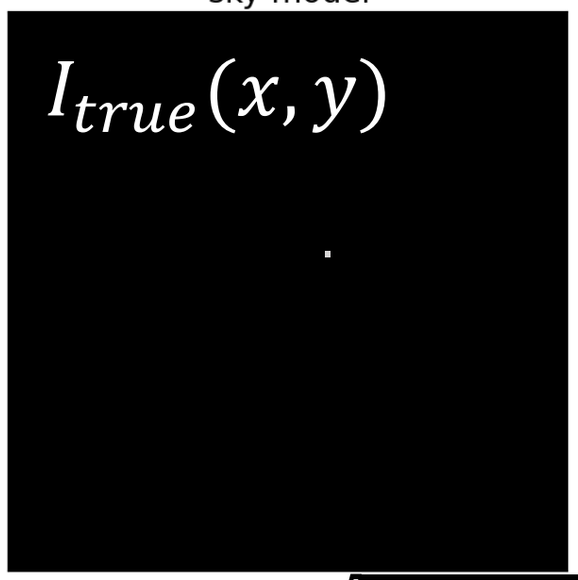
Credit: Kuo-Song Wang (ASIAA)

Simulated visibilities and images

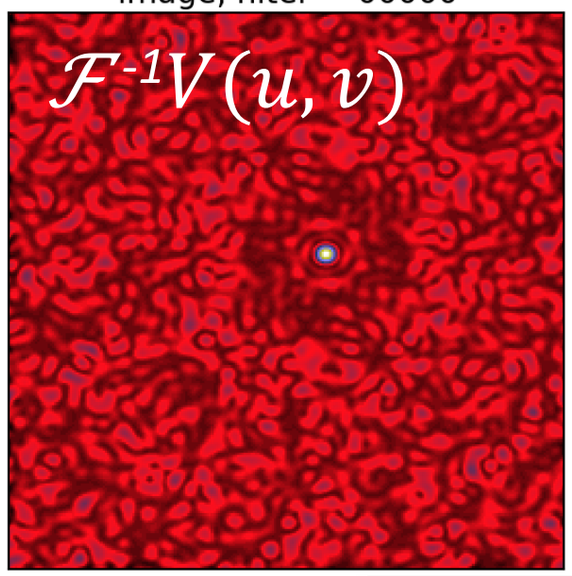
Point source (off-center)

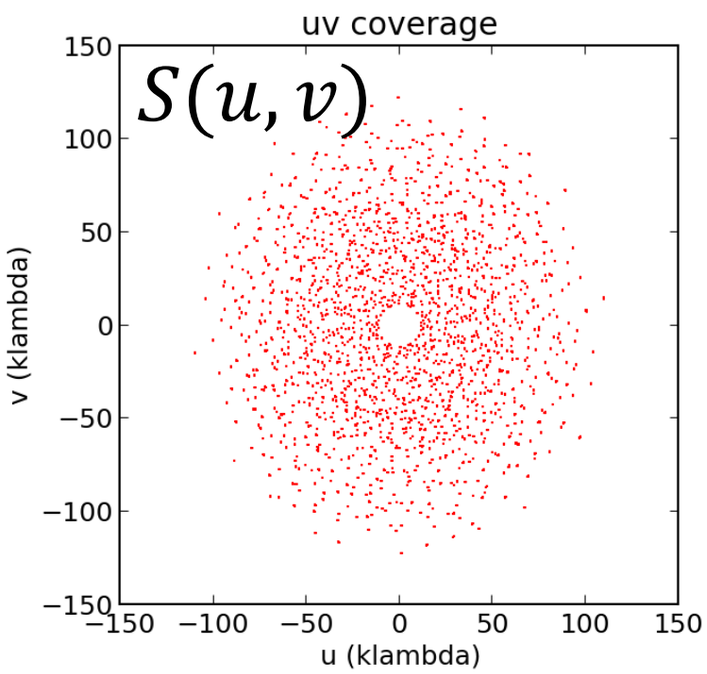
Dirty beam



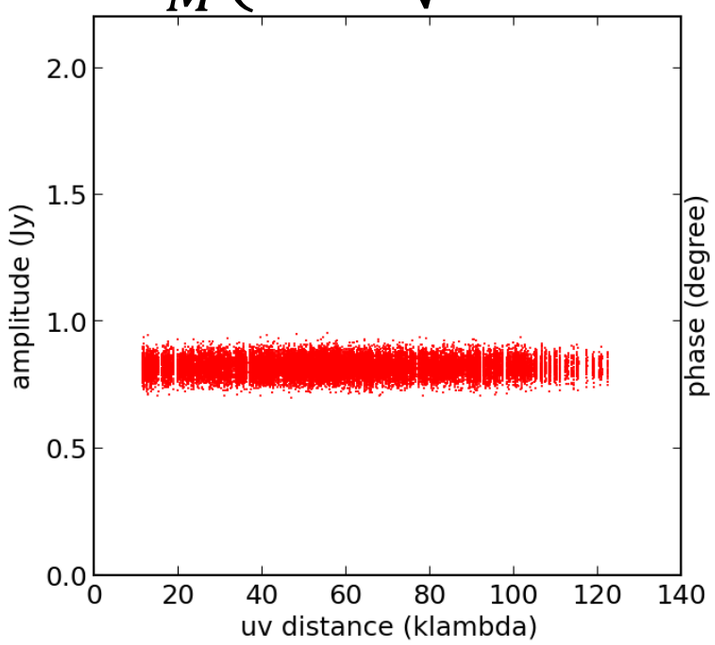


Dirty image

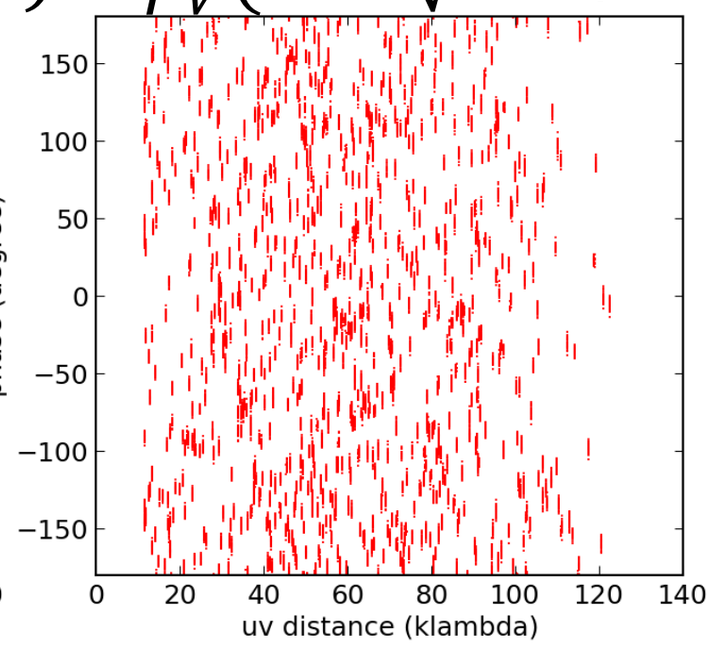




$V_M(r=\sqrt{u^2+v^2})$



$\phi_V(r=\sqrt{u^2+v^2})$

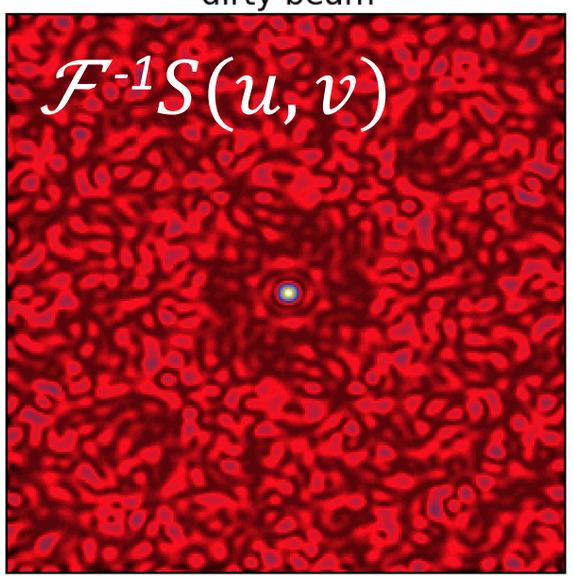


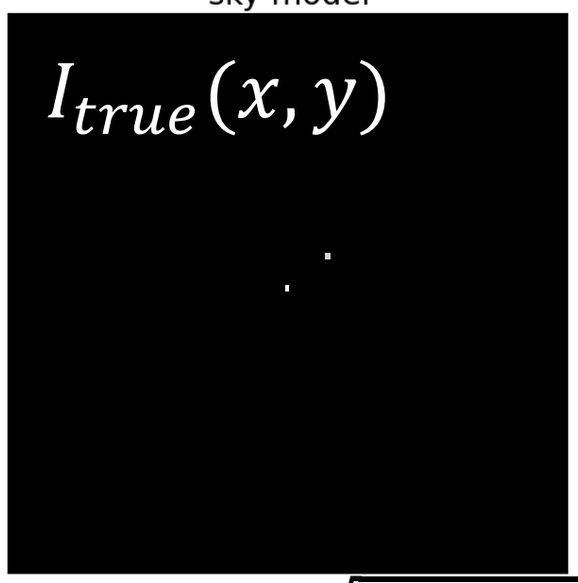
Credit: Kuo-Song Wang (ASIAA)

Simulated visibilities and images

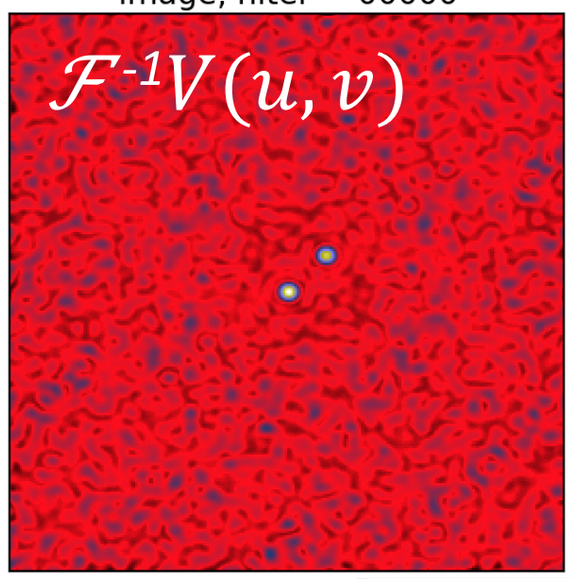
Double point sources

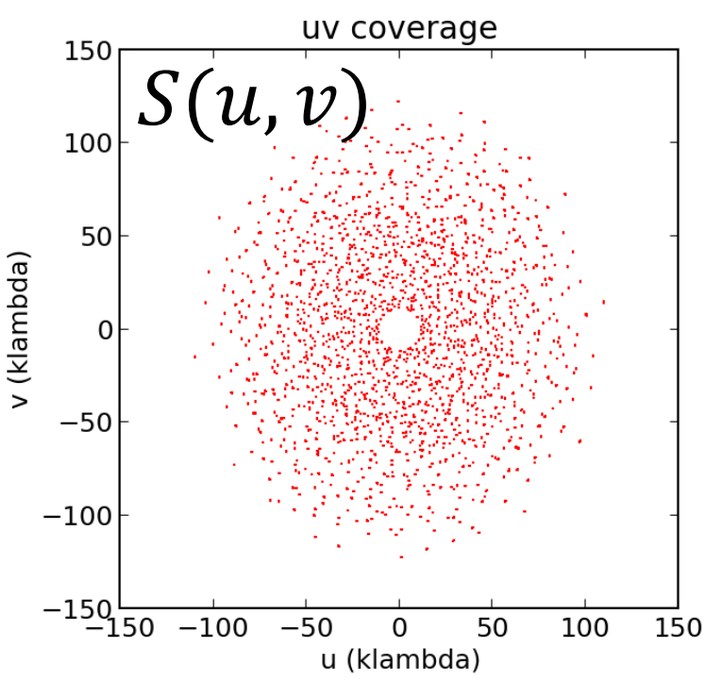
Dirty beam



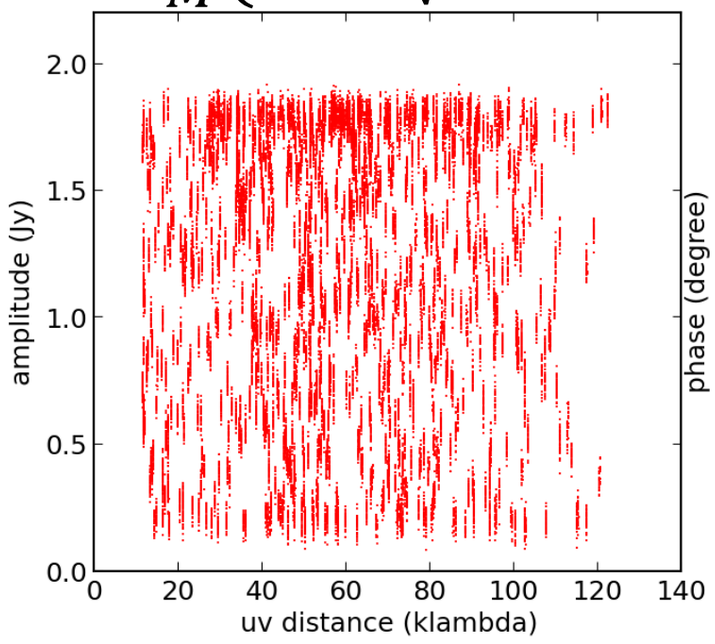


Dirty image

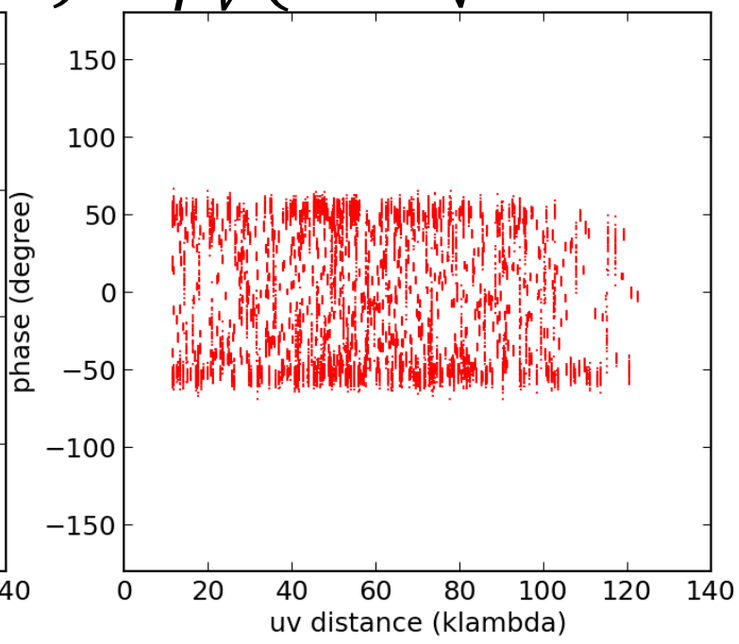




$V_M(r=\sqrt{u^2+v^2})$



$\phi_V(r=\sqrt{u^2+v^2})$

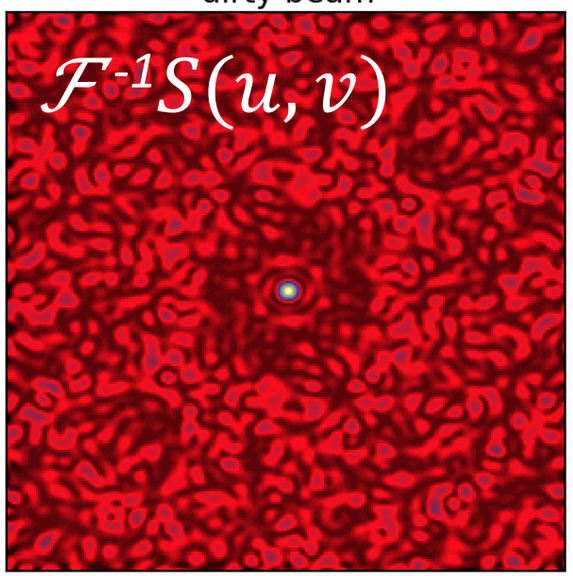


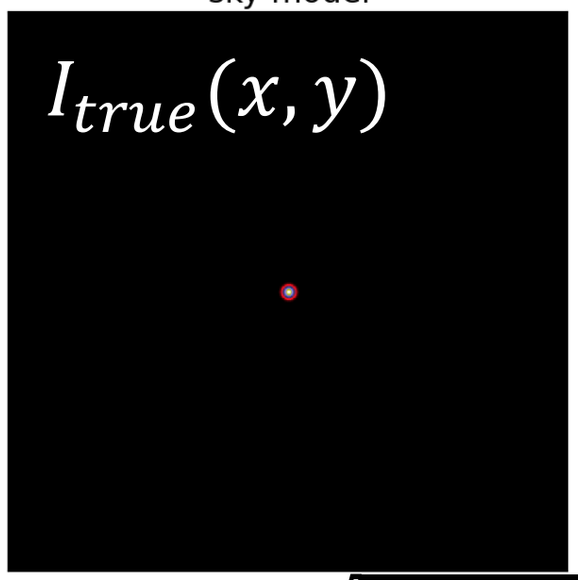
Credit: Kuo-Song Wang (ASIAA)

Simulated visibilities and images

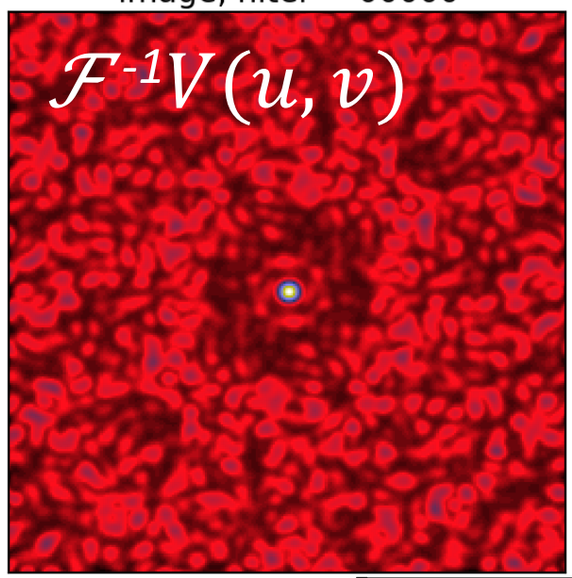
Small gaussian

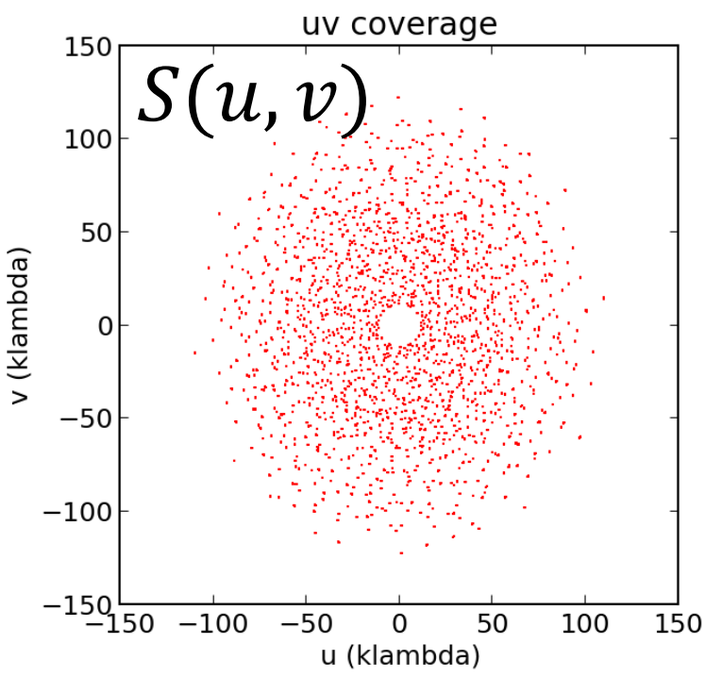
Dirty beam



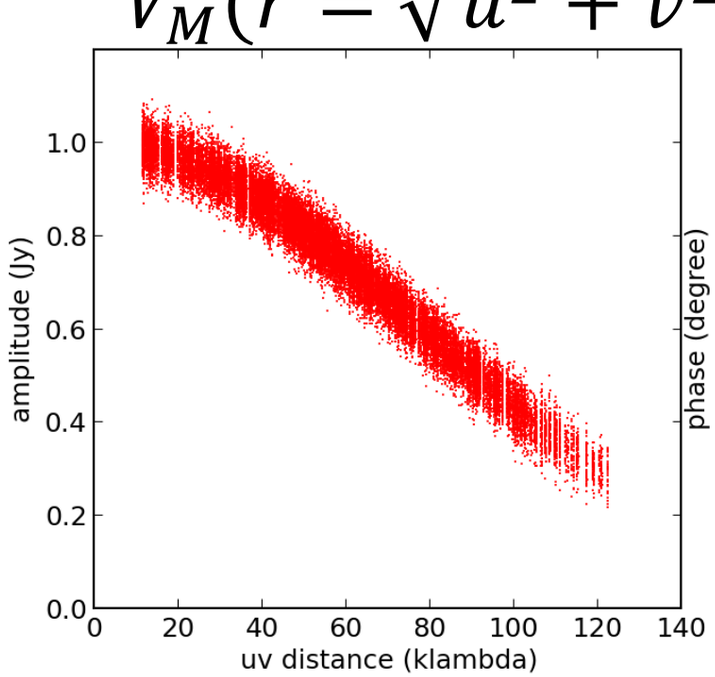


Dirty image

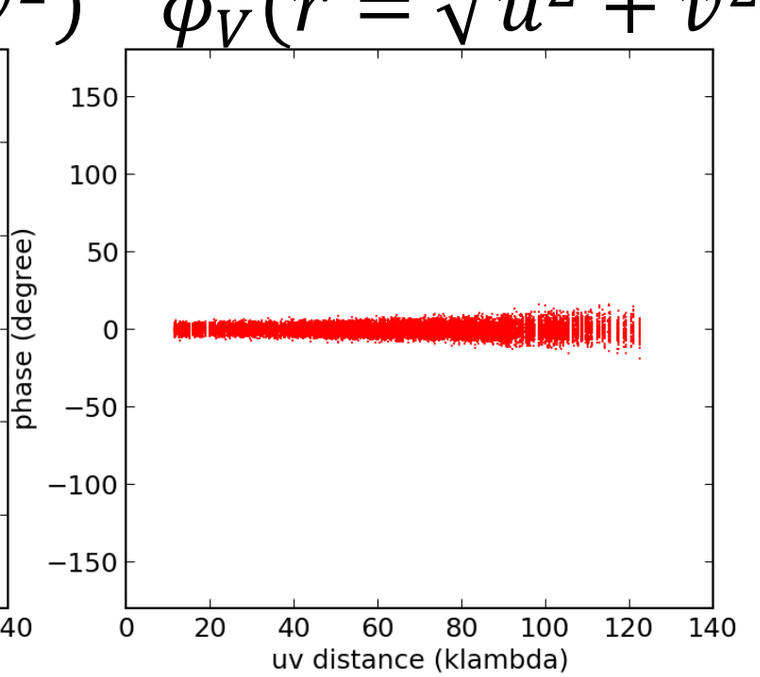




$V_M(r=\sqrt{u^2+v^2})$



$\phi_V(r=\sqrt{u^2+v^2})$

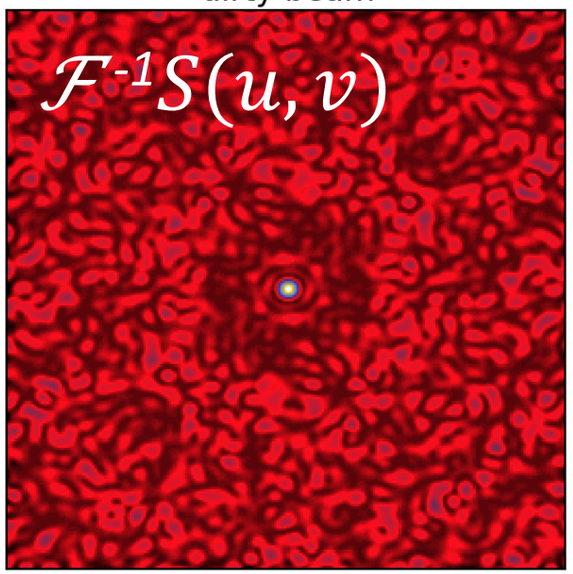


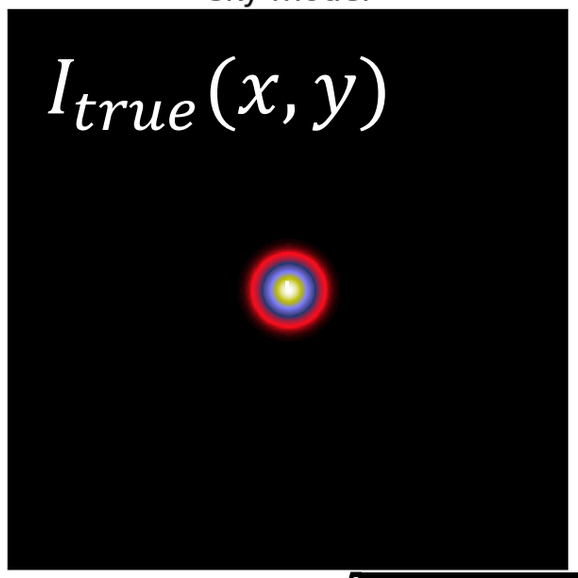
Credit: Kuo-Song Wang (ASIAA)

Simulated visibilities and images

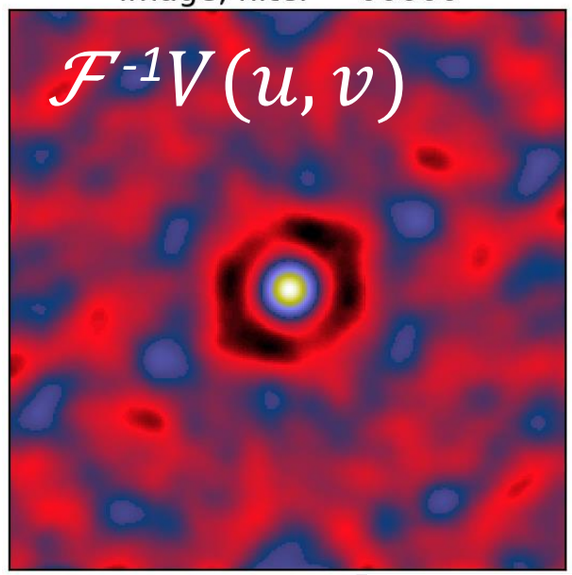
Large gaussian

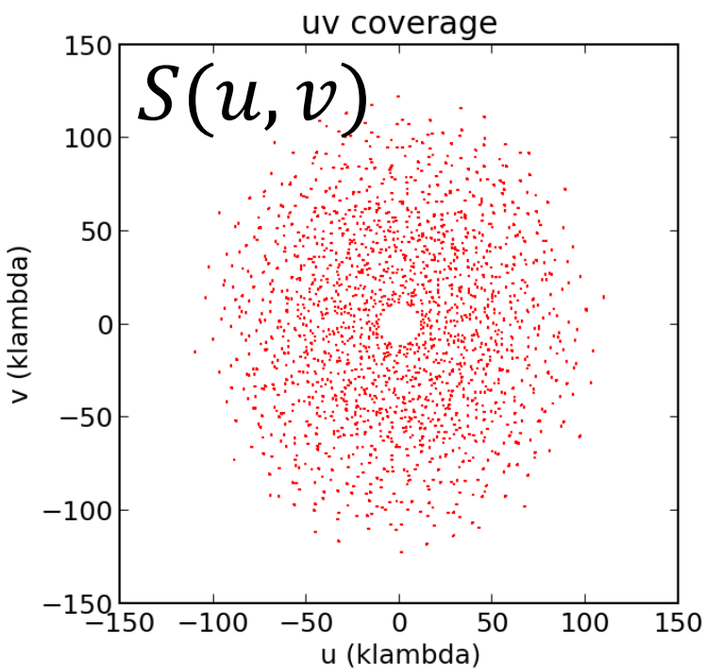
Dirty beam



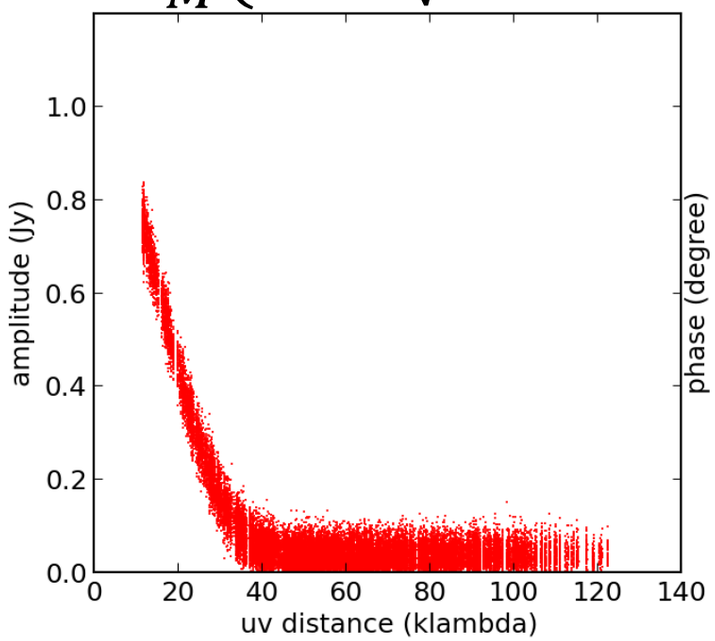


Dirty image

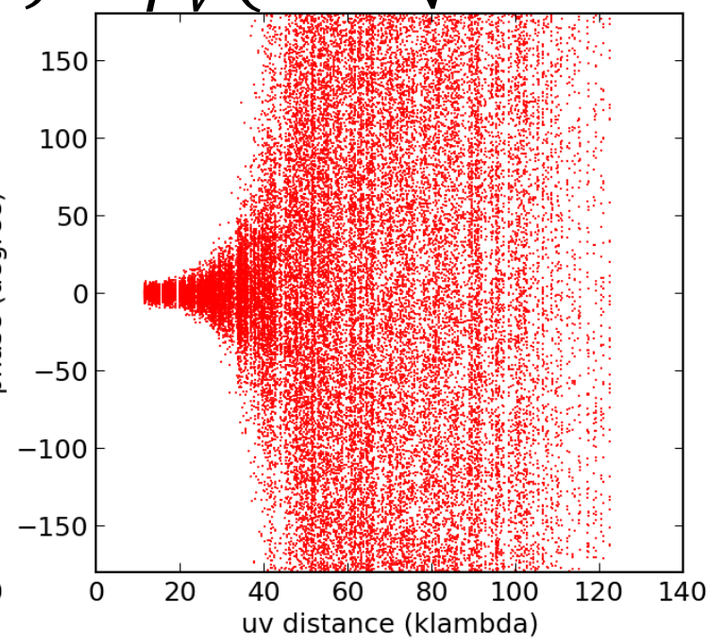




$V_M(r=\sqrt{u^2+v^2})$



$\phi_V(r=\sqrt{u^2+v^2})$

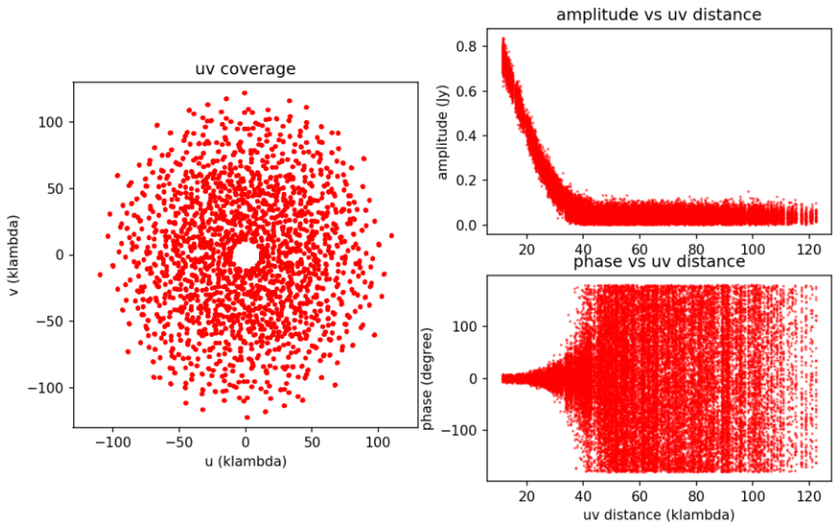


Credit: Kuo-Song Wang (ASIAA)

Deconvolution procedure

CLEAN algorithm

Inversion ($\mathcal{F}^{-1}$): Obtain *Dirty image*, $I_{dirty}(x, y)$.
Deconvolution: Perform on the image domain
Restoration: Obtain *Clean image*, $I_{clean}(x, y)$.

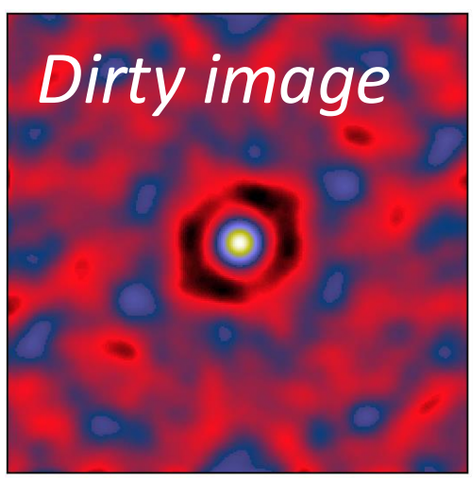


$$V_{\text{obs}}(u, v) = S(u, v)V(x, y)$$

$$V_{\text{obs}}(u, v) = S(u, v) \iint I_{\text{true}}(x, y) e^{-2\pi i(ux+vy)} dxdy$$

$\mathcal{F}^{-1}$

Approximation



$$I_{dirty}(x, y) \longrightarrow I_{clean}(x, y)$$

Deconvolution

CLEAN

Basic concepts

Högbom (1974): tclean/ “deconvolver=‘hogbom’ ”

Assumption: Target source consists of multiple point sources.

1. [**Inversion**] Compute dirty beam ($\mathcal{F}^{-1}[S(u,v)]$) and dirty map ($\mathcal{F}^{-1}[V(u,v)]$)

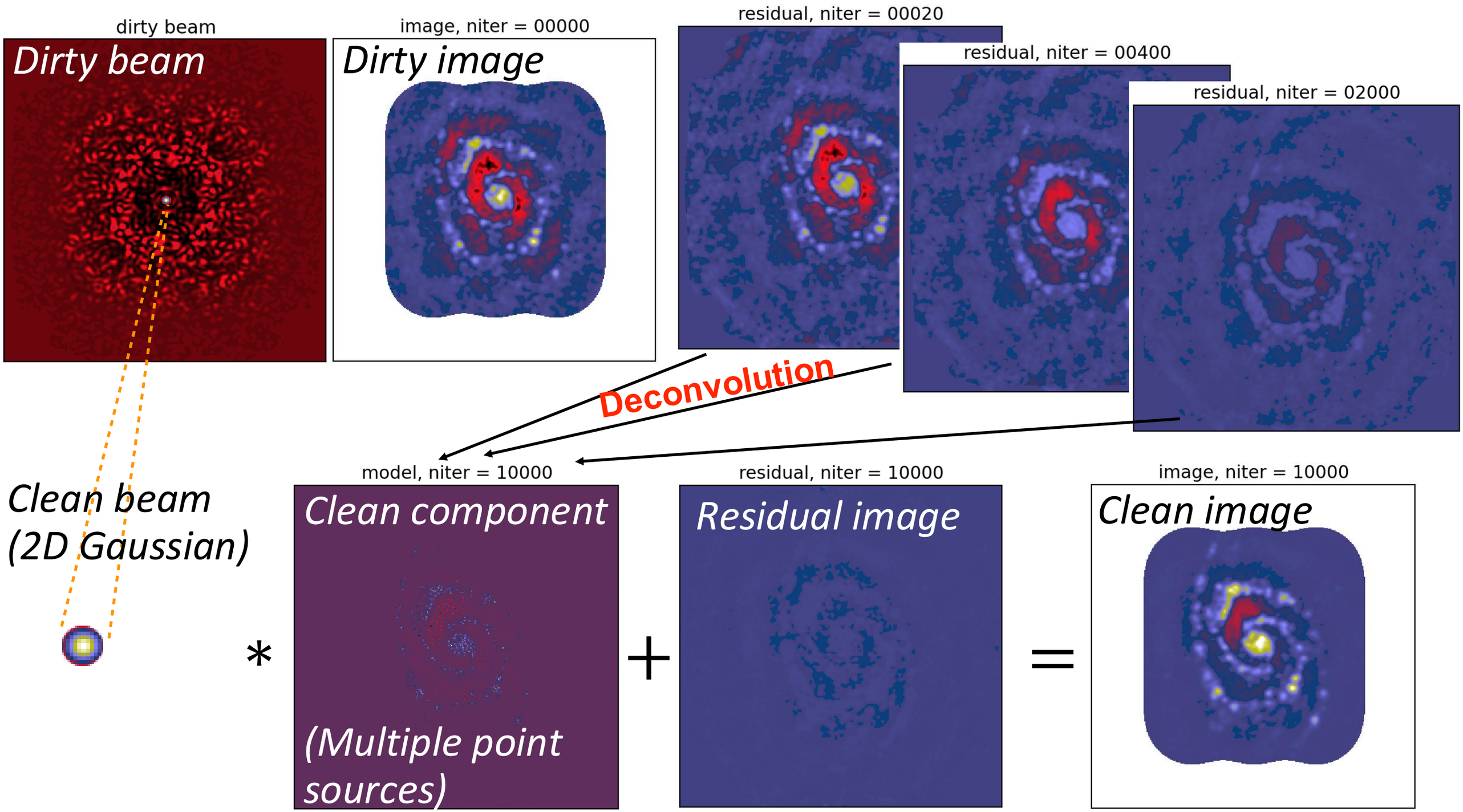
2. [**Deconvolution**] Subtract over the whole dirty map with a dirty beam pattern (multiplied by a factor, e.g., 0.1) which is centered at the point where the dirty map has its maximum of absolute value of intensity

3. Repeat step 2, each time replacing the dirty map by the remaining map (i.e., residual map) from the previous iteration

4. [**Restoration**] Once the iteration is satisfied by
   (a. tclean/ “niter”) reaching the iteration limit or
   (b. tclean/ “threshold”) that the brightness in the remaining map is lower than a threshold,
   generate the clean map by adding the final residual map with the clean component map convolved with the clean beam

CLEAN

As an iterative process

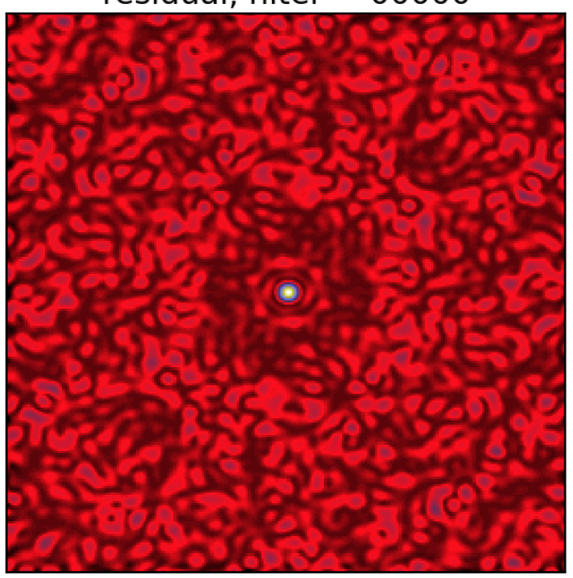


Credit: Kuo-Song Wang (ASIAA)

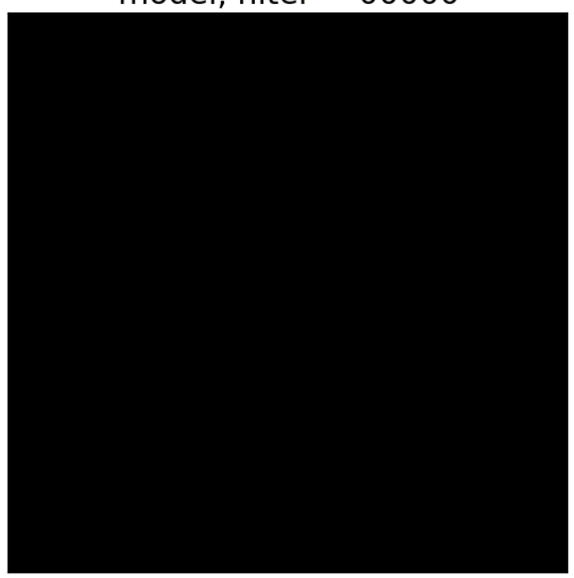
CLEAN examples

Point source

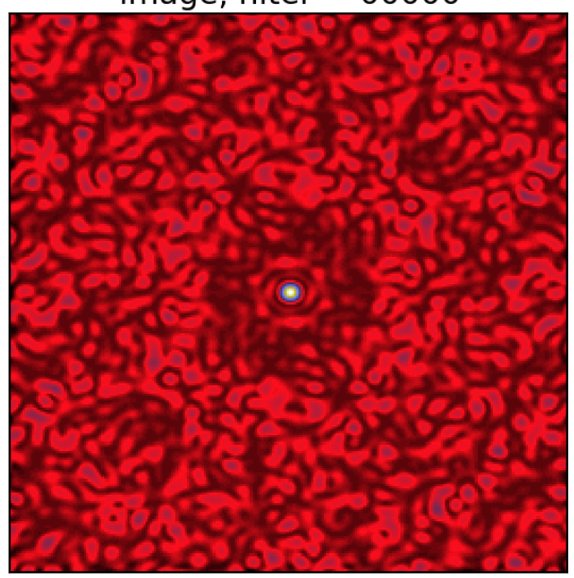
Residual image



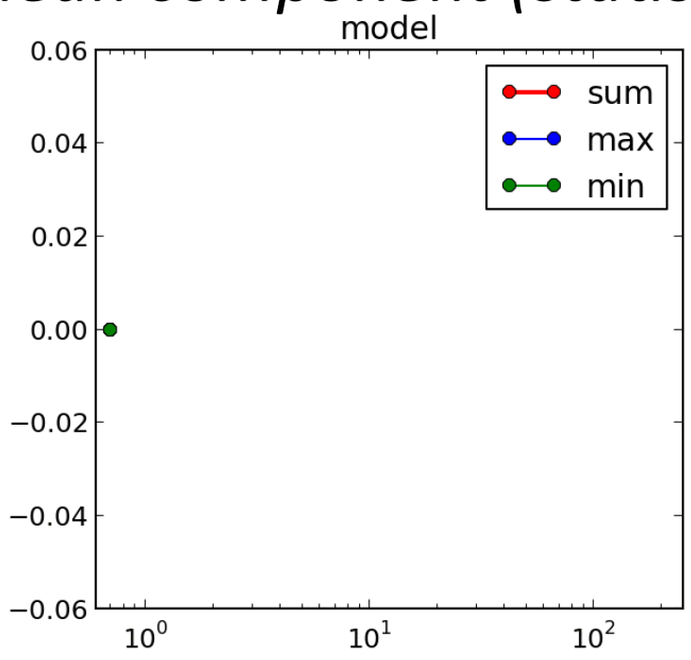
Clean component



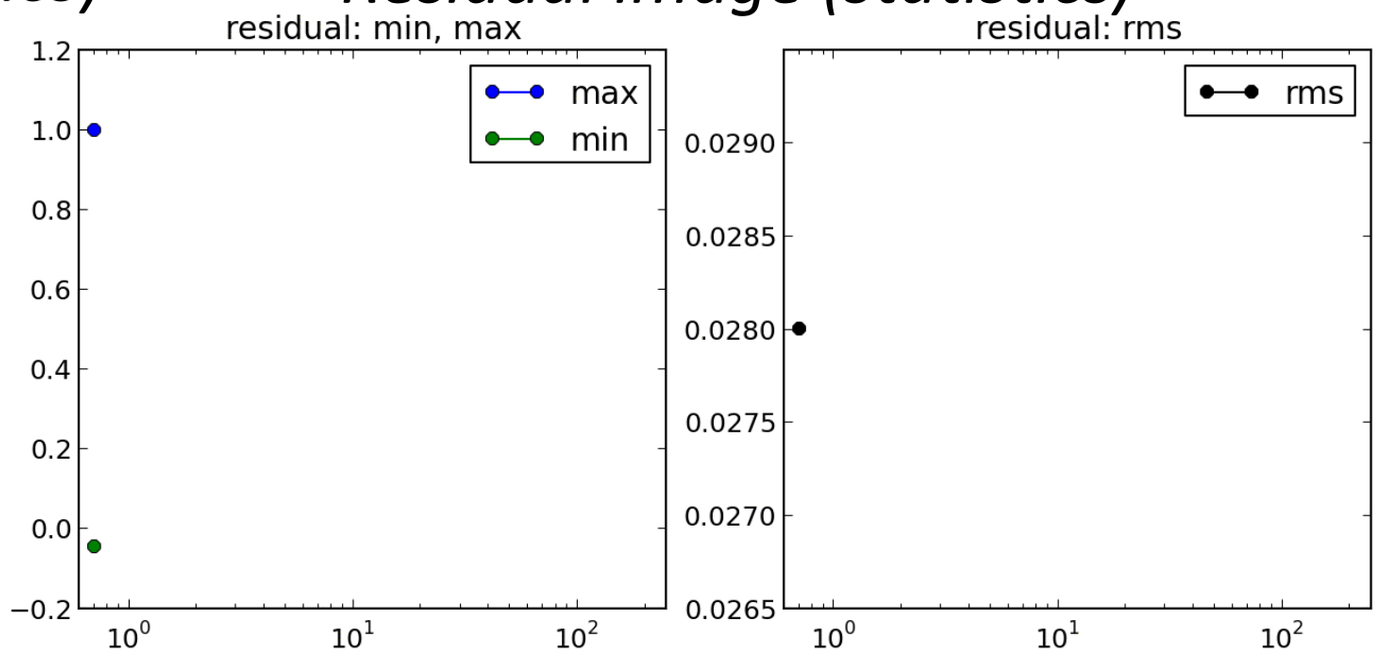
Clean image



Clean component (statistics)



Residual image (statistics)

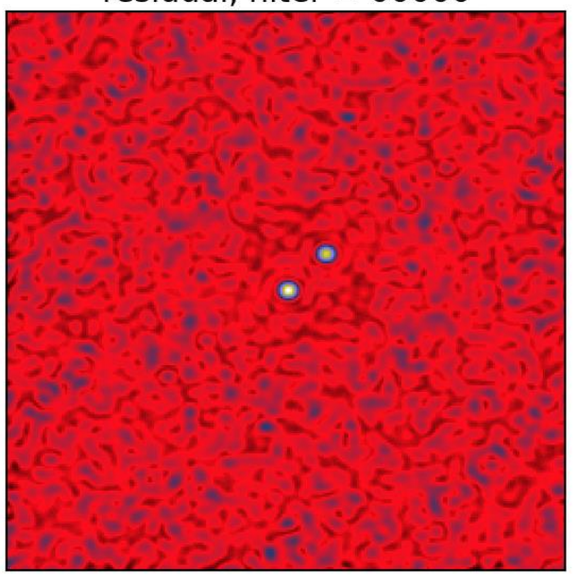


Credit: Kuo-Song Wang (ASIAA)

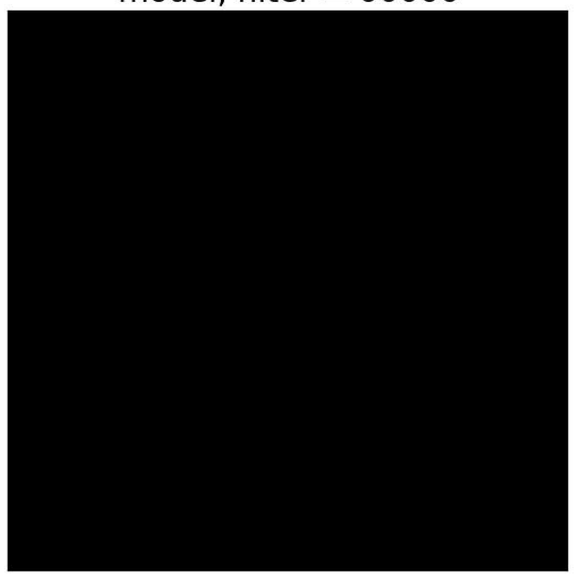
CLEAN examples

Double point sources

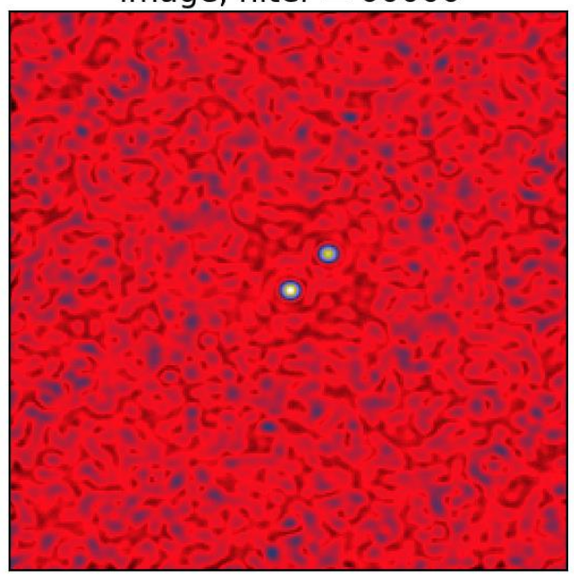
Residual image



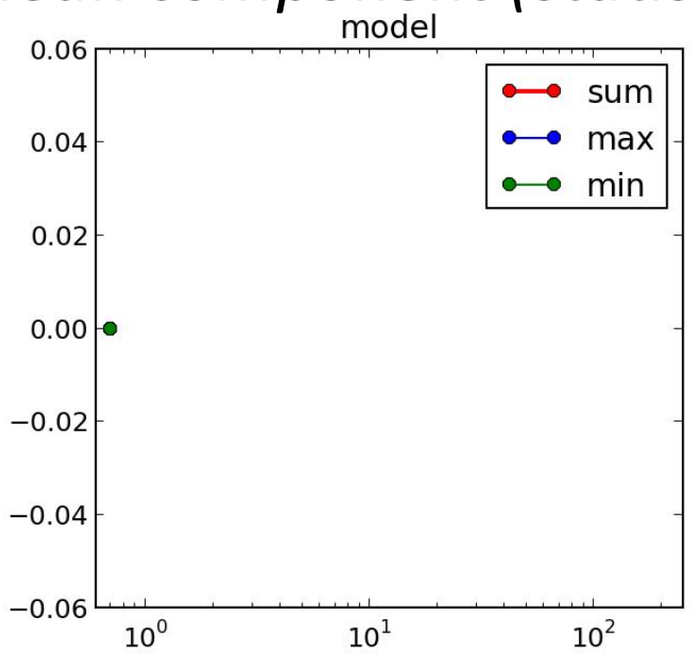
Clean component



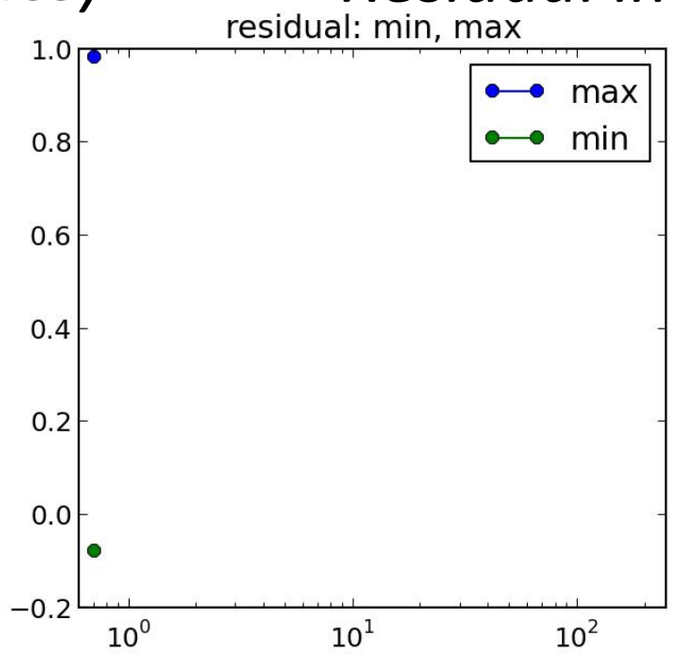
Clean image

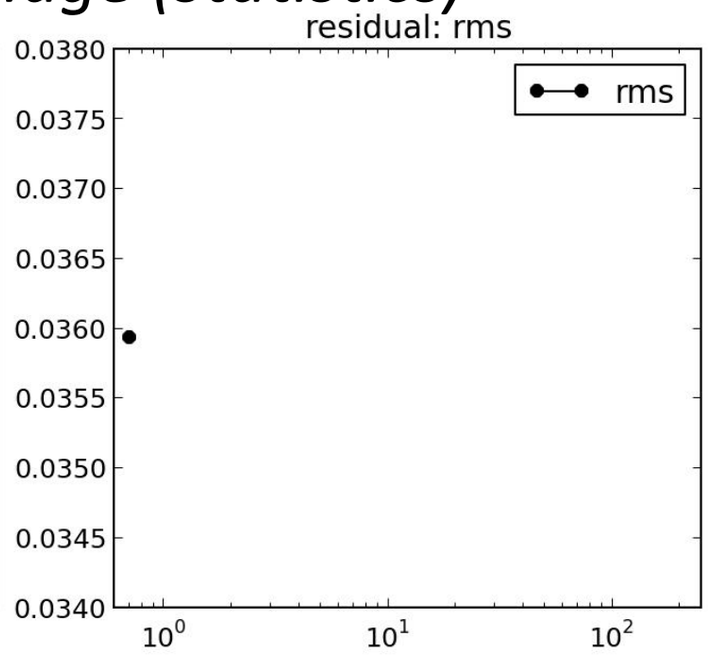


Clean component (statistics)



Residual image (statistics)



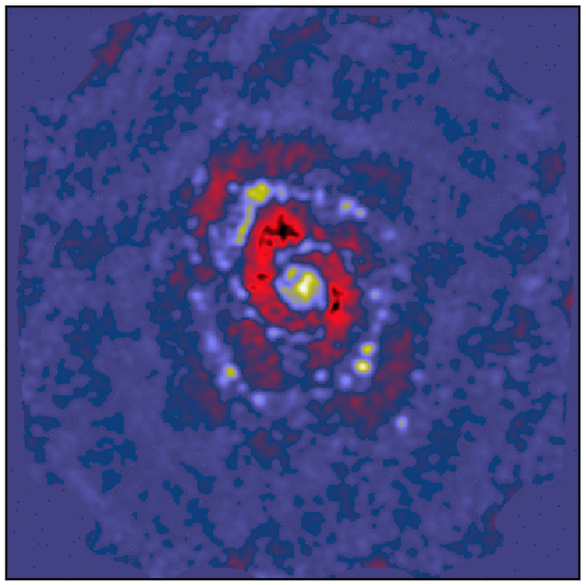


Credit: Kuo-Song Wang (ASIAA)

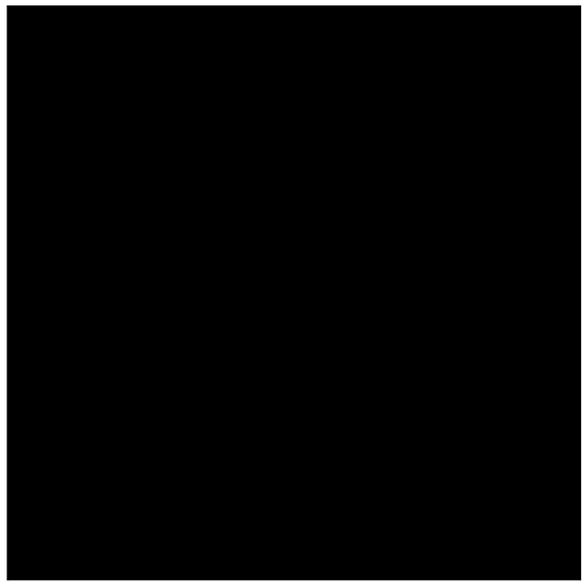
CLEAN examples

Galaxy

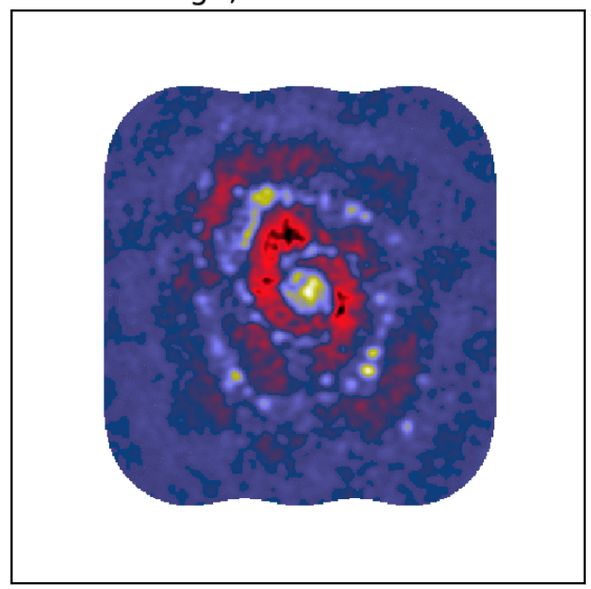
Residual image



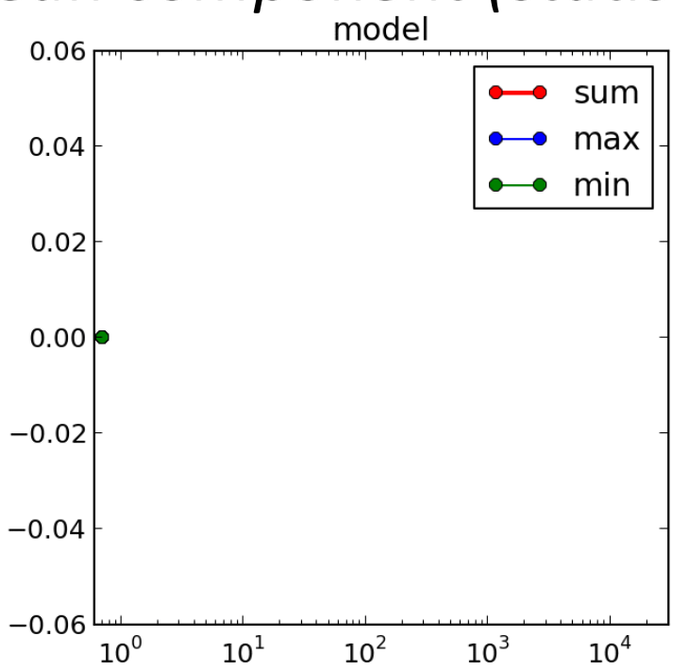
Clean component



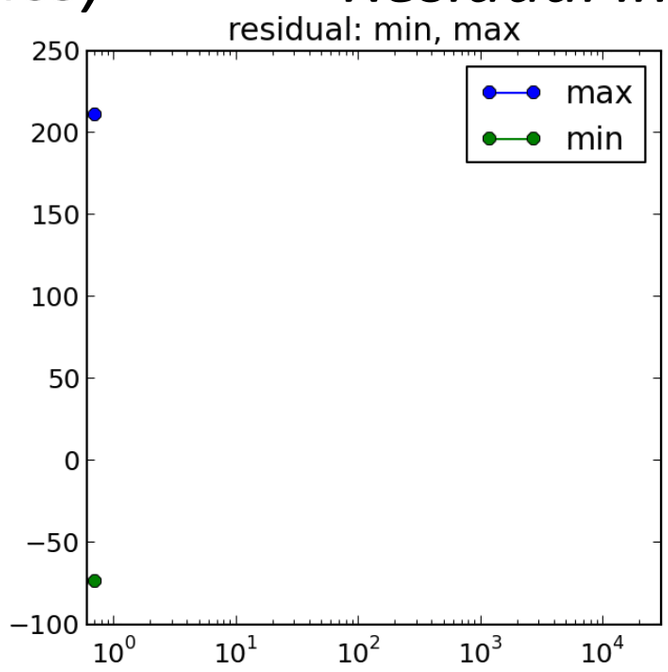
Clean image

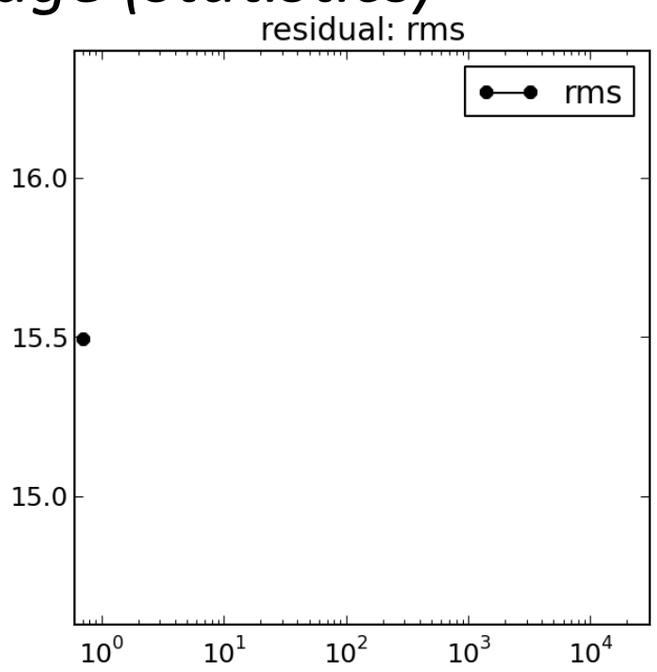


Clean component (statistics)



Residual image (statistics)





Credit: Kuo-Song Wang (ASIAA)

Terms associated with CLEAN

clean, dirty, beam, model, residual…

Dirty beam: point-source response (point spread function) of the data; $\mathcal{F}^{-1}$ of uv-coverage ($\mathcal{F}^{-1}[S(u,v)]$) and normalize the peak to unity

Dirty map: $\mathcal{F}^{-1}$ of visibilities ($\mathcal{F}^{-1}[V(u,v)]$)

Clean beam: (usually) a 2D Gaussian approximation of the main lobe of the dirty beam

Clean map: the sum of clean component map convoluted with clean beam and the residual map

Clean component (model) map: an assumable of point sources determined from CLEAN algorithm [Jy/pixel]

Residual map: a dirty map with clean components (convolved with dirty beam) removed, ideally noise-like at final stage of CLEAN

CASA tclean task

does

- weighting (visibility)
- gridding (visibility)
- normalization (visibility)
- $\mathcal{F}^{-1}$
- deconvolution (image)
- restoration (image)

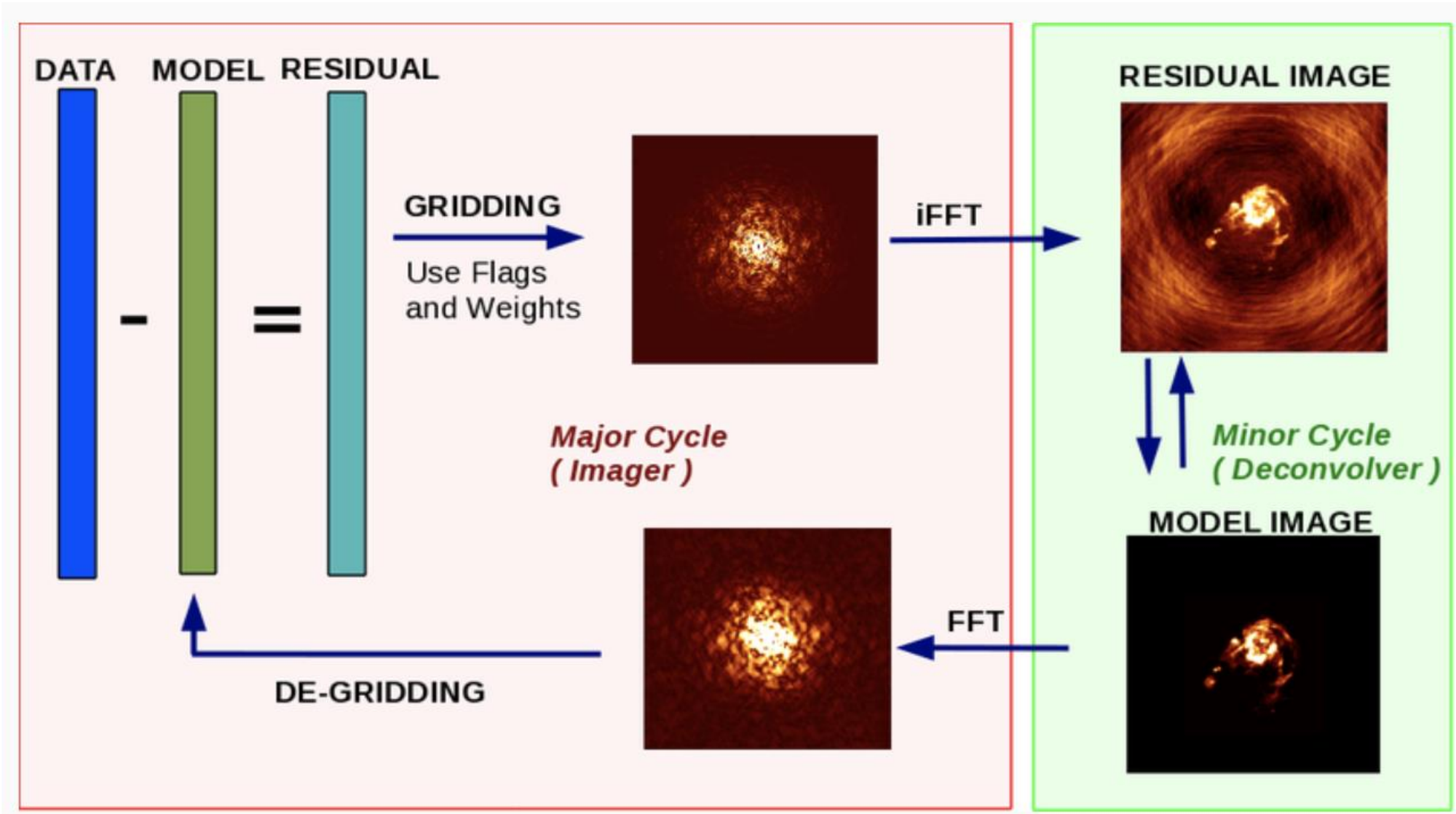
Basic Sequence of Imaging Logic:

Data : Calibrated visibilities, data weights, UV sampling function
Input : Algorithm and iteration controls (stopping threshold, loop gain,...)
Output : Model Image, Restored Image, Residual Image,...

Initialize the model image
Compute the point spread function
Compute the initial residual image
While (not reached global stopping criterion) /* Major Cycle */
{
  While (not reached minor-cycle stopping criterion) /* Minor Cycle */
  {
    Find the parameters of a new flux component
    Update the model and residual images
  }
  Use current model image to predict model visibilities
  Calculate residual visibilities (data - model)
  Compute a new residual image from residual visibilities
}
Convolve the final model image with the fitted beam and add to the residual image

Cotton-Schwab CLEAN algorithm

major cycle and minor cycle

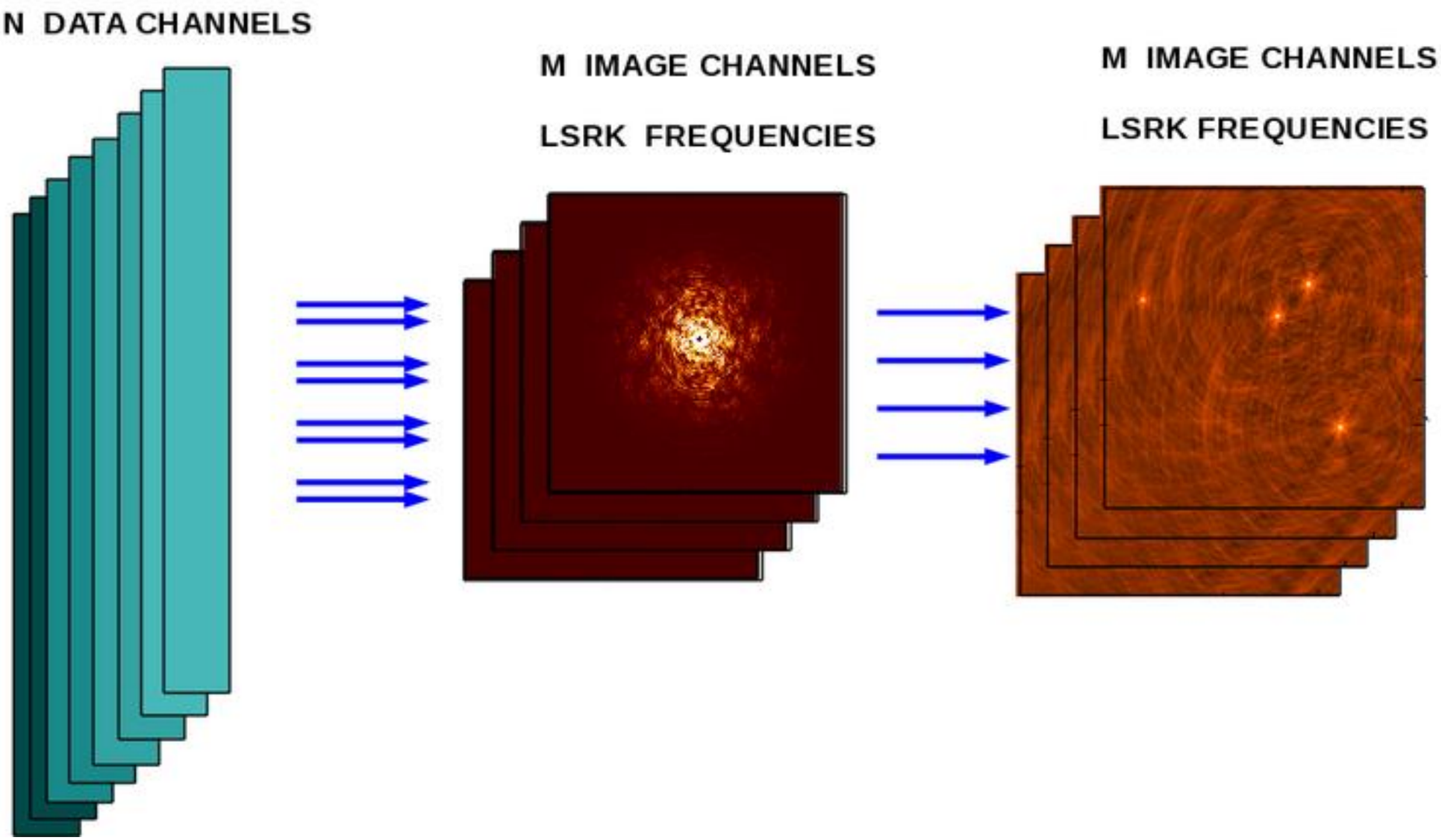


Iterative Image Reconstruction - Major and Minor Cycles

CASA tclean task

what tclean can do…

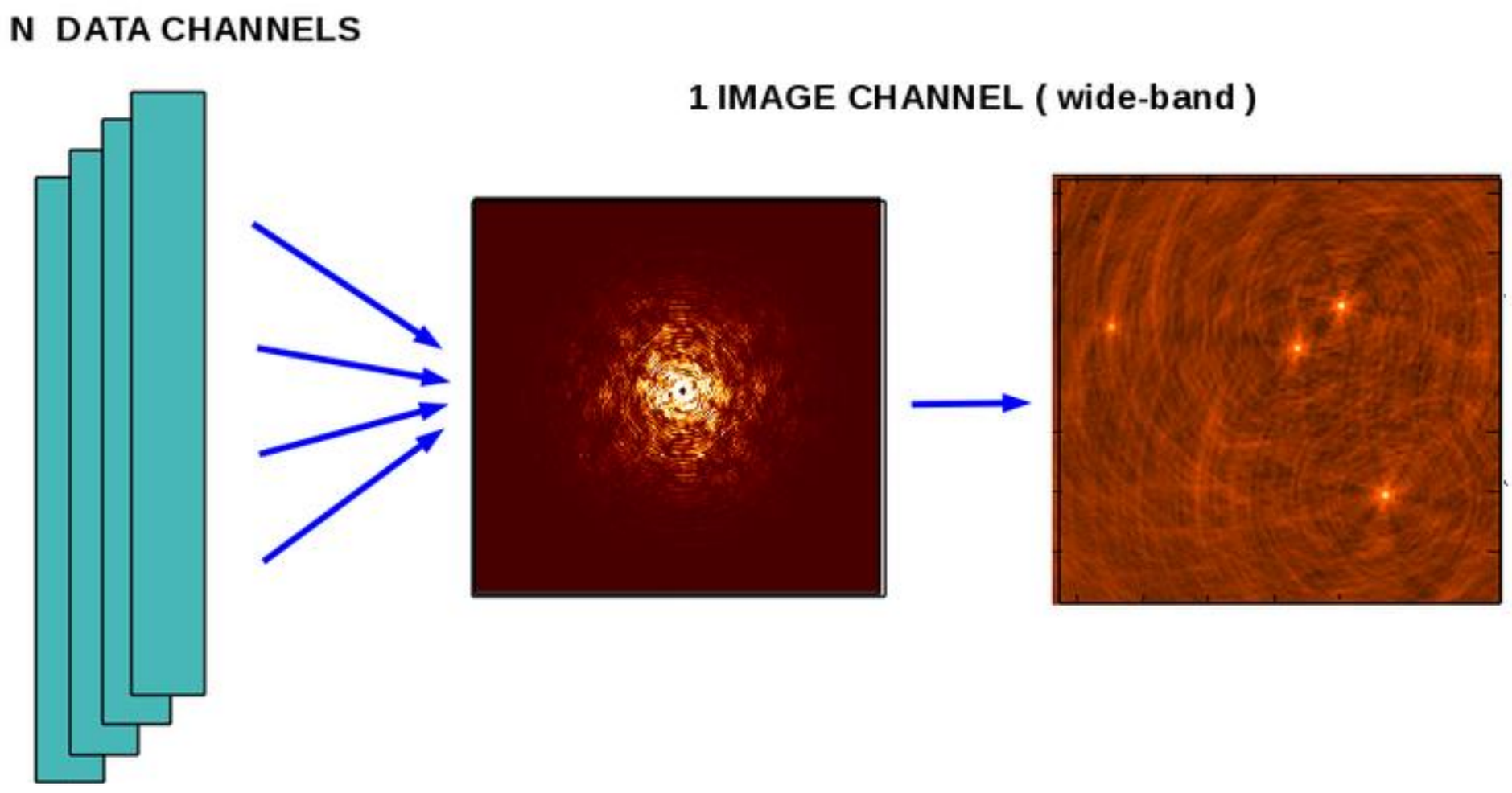
- Spectral cubes



CASA tclean task

what tclean can do…

- Multi Frequency Synthesis (MFS) - single wideband image



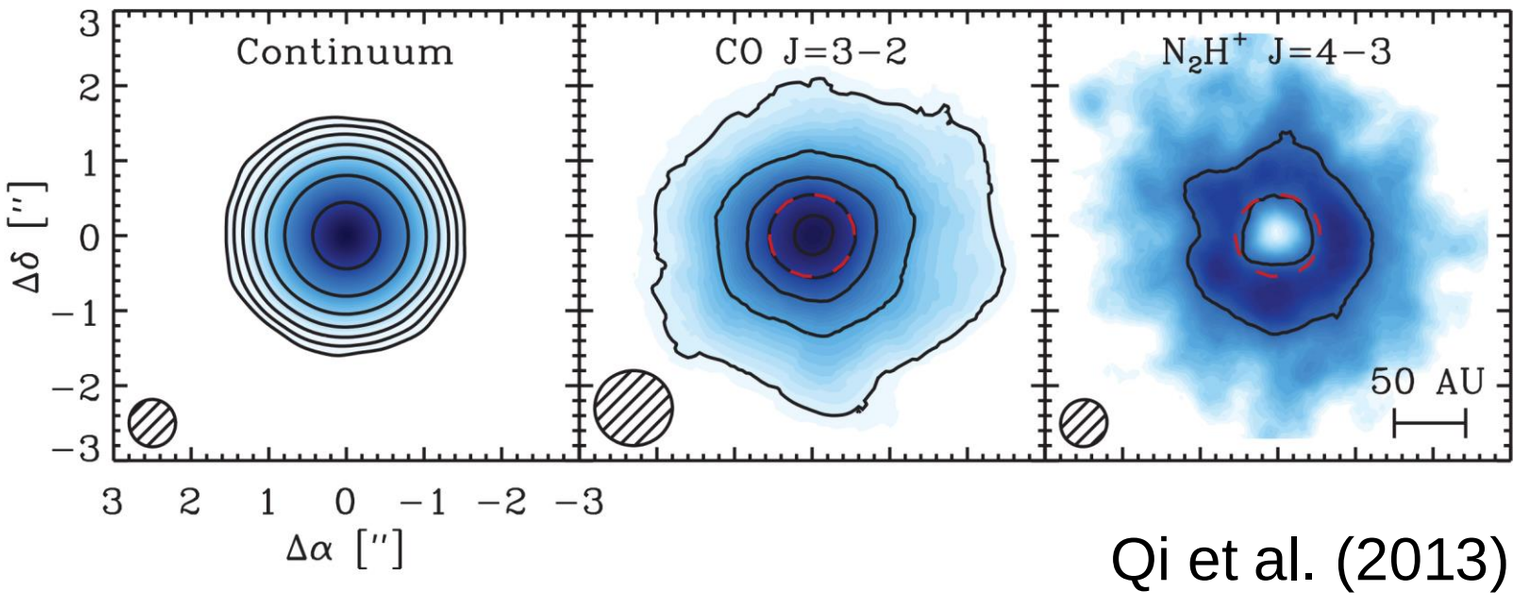
Goals of this talk

- Gain some intuition for interferometric observations
  - What is the visibility?
- Understand the imaging process
  - What is the CLEAN algorithm?
- Learn the `tclean` task in CASA
  - Hands-on session (this afternoon)

Test dataset for today

Continuum and N_2H^+ in TW Hydra at Band 7

- First Look at Imaging CASA 6.6.1 (https://casaguides.nrao.edu/index.php/First_Look_at_Imaging_CASA_6.6.1)
- First Look at Line Imaging CASA 6.6.1 (https://casaguides.nrao.edu/index.php/First_Look_at_Line_Imaging_CASA_6.6.1)



Qi et al. (2013)

tclean

Slide Credit: Kuo-Song Wang (ASIAA) + Some modification with our test data

Test dataset

```
├── MyTutorial_cont
│   ├── tclean_tutorial_cont_pre.py
│   ├── tclean_tutorial_cont.py
│   ├── twhya_calibrated.ms
│   └── twhya_calibrated.ms.tar.gz
└── MyTutorial_line
    ├── tclean_tutorial_line_pre.py
    ├── tclean_tutorial_line.py
    ├── twhya_selfcal.ms
    └── twhya_selfcal.ms.tar.gz
```

- Extract *tar.gz files in each directory → *ms files

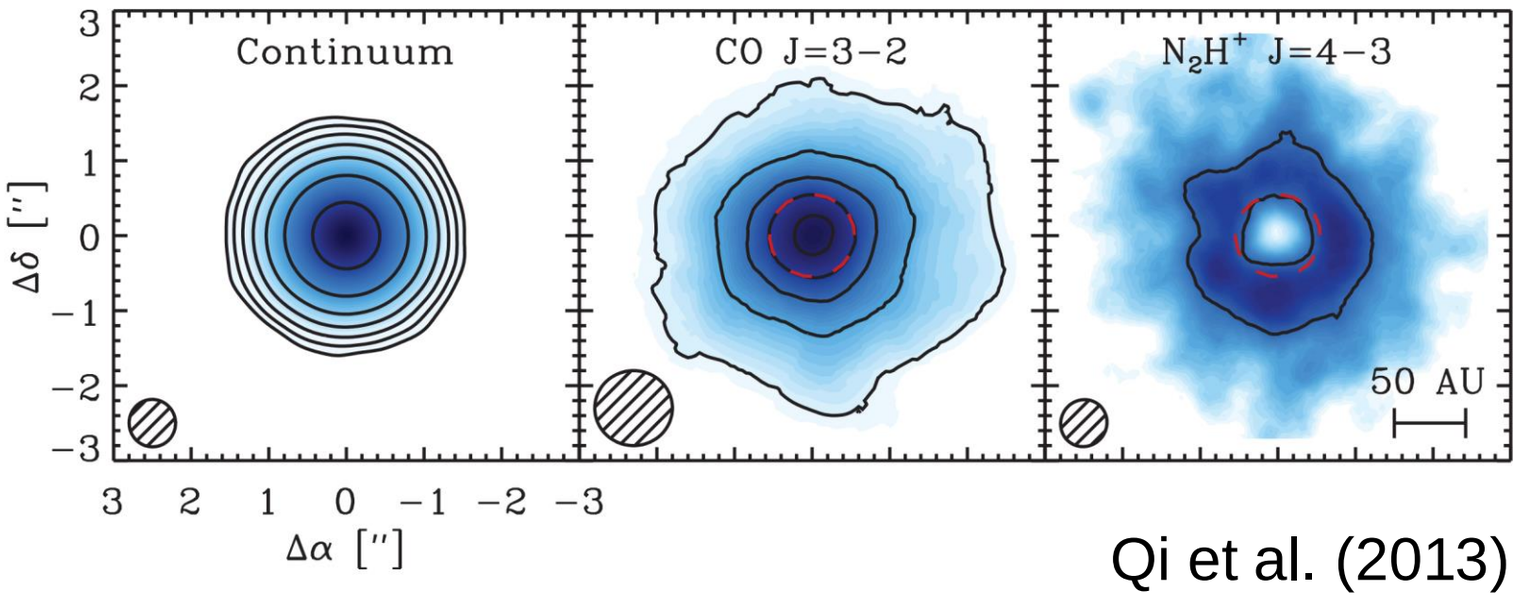
  tar –zxvf twhya_calibrated.ms.tar.gz

  tar –zxvf twhya_selfcal.ms.tar.gz

Tasks for today

use tclean to generate some images

- **A continuum image**
- A line cube



Qi et al. (2013)

CASA basics

interact with the ipython interface

- **taskhelp**: see all available CASA tasks
- **<task>?**, **help(<task>)**: see the online manual of a task (e.g. listobs?)
- **inp <task>**: to see all parameters of a task (e.g. inp listobs)
- **inp**: check what task is currently activated
- **vis='data.ms'**: to set a parameter via "="
- **par.<parameter>?**: see examples of a parameter (e.g. par.spw?)
- **go**: execute a task
- **tget <task>**: get last input values from file on disk for a specified task (tget listobs)
- **default(<task>)**: reset a task with default parameters
- **execfile('myScript.py')**: *read and execute a **Python script** from a file*

Measurement set

as multi-dimensional tables

- Get a summary of a measurement set with listobs and save the summary as a text file:

  > listobs(vis='twhya_calibrated.ms', listfile='twhya_calibrated.ms.listobs')

- If listfile is not defined, the summary will be shown in the logger.

- Open the text file with your text editor (e.g. vi, emacs, etc.)

NOTE: if you see warnings about **leap seconds**, try:
in a terminal
cd [casa directory]/data
rsync -avz rsync://casa-rsync.nrao.edu/casa-data .

Measurement set

as multi-dimensional tables

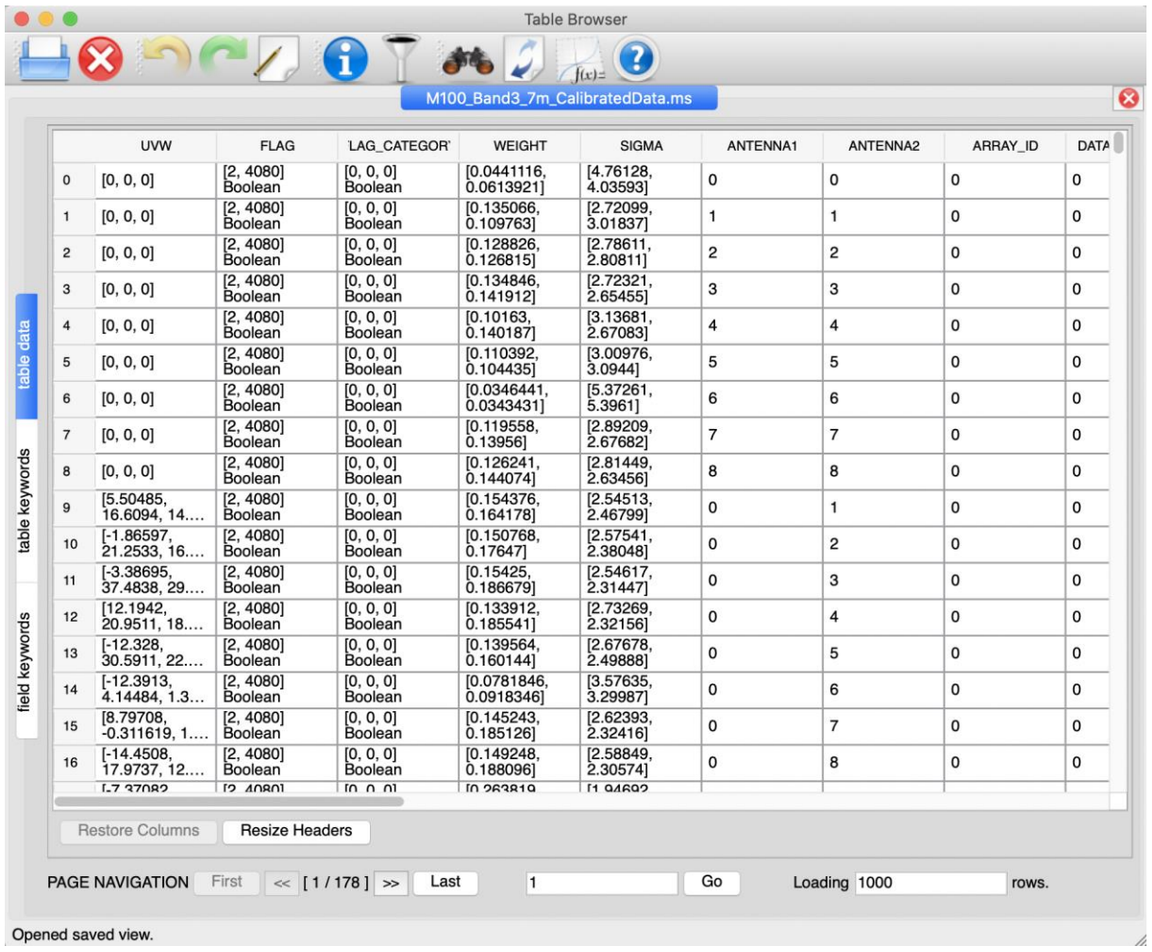
- Observation, ObservationID => time
- Fields => pointing (which parts of skies were observed)
- Spectral windows => frequency samples
- Sources => observed targets (calibrators, science targets)
- Antennas => spatial frequency samples

Visualization of a MS

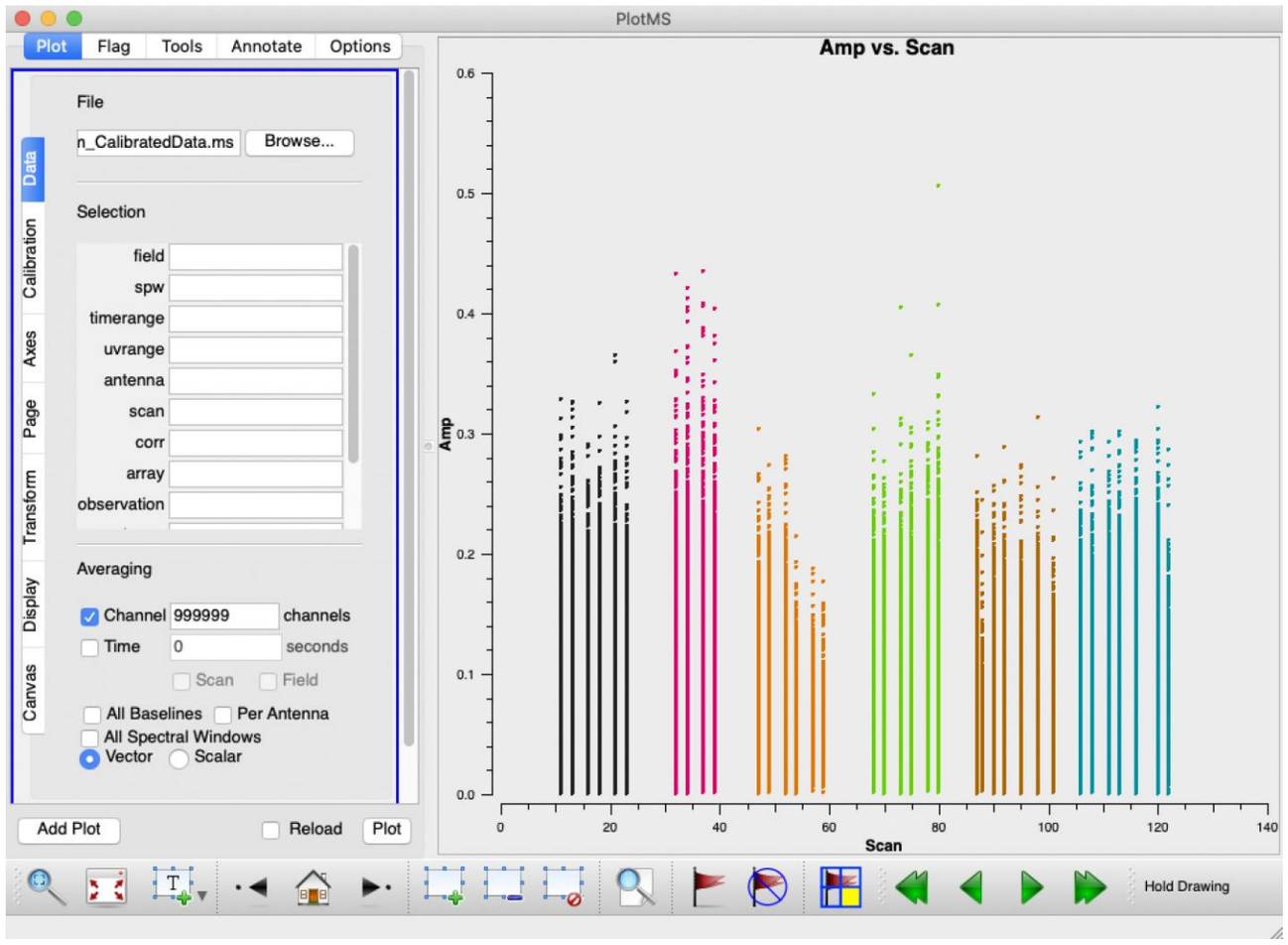
browstable and plotms

- browstable: a GUI to visualize a MS as tables
- plotms: a GUI to visualize a MS as plots

browsetable



plotms



Useful tasks to manipulate MS

split and concat

- split: create a visibility subset from an existing visibility set
- concat: concatenate several visibility data sets
- uvcontsub: continuum fitting and subtraction in the uv plane
- plotants: plot the antenna distribution in the local reference frame

- Execution block (EB): ALMA data may consist of multiple EBs (e.g. xxx.ms.split.cal; per EB < 2h)
- Each EB consists of observations of calibrators and science targets
- “split” science targets and “concat” them

Test dataset

```
├── MyTutorial_cont
│   ├── tclean_tutorial_cont_pre.py
│   ├── tclean_tutorial_cont.py
│   ├── twhya_calibrated.ms
│   └── twhya_calibrated.ms.tar.gz
```

- inside CASA: execfile(‘tclean_tutorial_cont_pre.py’)

```
listobs(vis='twhya_calibrated.ms', listfile='twhya_listobs.txt')

split(vis='twhya_calibrated.ms', field='5', width='8',
        outputvis='twhya_smoothed.ms', datacolumn='data')

listobs(vis='twhya_smoothed.ms', listfile='twhya_listobs_smoothed.txt')

plotms(vis='twhya_calibrated.ms',
       xaxis='u',
       yaxis='v',
       avgchannel='10000',
       avgspw=False,
       avgtime='1e9',
       avgscan=False,
       coloraxis='field',
       showgui=True)
```

tclean

tclean

form images from visibilities and reconstruct a sky model

- **Interactive mode**

  > inp tclean

- **Non-interactive mode**

Once you are more familiar with tclean, it is better to run tclean with a python script, for example, to construct a workflow.

- inside CASA:
  execfile('myScript.py')

- outside CASA:
  casa -c myScript.py

```
inputMS = 'sim_M51_7m.aca.cycle5.noisy.ms'

tclean(vis = inputMS,
       imagename = inputMS + '.clean',
       field = '0~6',
       spw = '0',
       specmode = 'cube',
       niter = 1000,
       threshold = '1mJy',
       deconvolver = 'hogbom',
       gridder = 'mosaic',
       imsize = [128,128],
       phasecenter = '3',
       cell = ['1.0arcsec'],
       weighting = 'natural')

myStat_residual = imstat(imagename='%s.clean.residual'%inputMS)

print myStat_residual['rms']

exportfits(imagename = '%s.clean.image'%inputMS,
           fitsimage = '%s.clean.image.fits'%inputMS,
           overwrite = True)
```

```
CASA <1>: inp tclean
--------> inp(tclean)
#  tclean :: Radio Interferometric Image Reconstruction
vis                 =         ''            #  Name of input visibility file(s)
selectdata          =       True            #  Enable data selection parameters
     field          =         ''            #  field(s) to select
     spw            =         ''            #  spw(s)/channels to select
     timerange      =         ''            #  Range of time to select from data
     uvrange        =         ''            #  Select data within uvrange
     antenna        =         ''            #  Select data based on antenna/baseline
     scan           =         ''            #  Scan number range
     observation    =         ''            #  Observation ID range
     intent         =         ''            #  Scan Intent(s)

datacolumn          = 'corrected'           #  Data column to image(data,corrected)
imagename           =         ''            #  Pre-name of output images
imsize              =      [100]            #  Number of pixels
cell                = ['1arcsec']           #  Cell size
phasecenter         =         ''            #  Phase center of the image
stokes              =        'I'            #  Stokes Planes to make
projection          =      'SIN'            #  Coordinate projection
startmodel          =         ''            #  Name of starting model image
specmode            =      'mfs'            #  Spectral definition mode (mfs,cube,cubedata, cubesource)
     reffreq        =         ''            #  Reference frequency

gridder             =  'standard'           #  Gridding options (standard, wproject, widefield, mosaic, awproject)
     vptable        =         ''            #  Name of Voltage Pattern table
     pblimit        =        0.2            #  PB gain level at which to cut off normalizations

deconvolver         =    'hogbom'           #  Minor cycle algorithm (hogbom,clark,multiscale,mtmfs,mem,clarkstokes)
restoration         =       True            #  Do restoration steps (or not)
     restoringbeam  =         []            #  Restoring beam shape to use. Default is the PSF main lobe
     pbcor          =      False            #  Apply PB correction on the output restored image

outlierfile         =         ''            #  Name of outlier-field image definitions
weighting           =   'natural'           #  Weighting scheme (natural,uniform,briggs, briggsabs[experimental])
     uvtaper        =         []            #  uv-taper on outer baselines in uv-plane

niter               =          0            #  Maximum number of iterations
usemask             =     'user'            #  Type of mask(s) for deconvolution:  user, pb, or auto-multithresh
     mask           =         ''            #  Mask (a list of image name(s) or region file(s) or region string(s) )
     pbmask         =        0.0            #  primary beam mask

fastnoise           =       True            #  True: use the faster (old) noise calculation. False: use the new improved noise calculations
restart             =       True            #  True : Re-use existing images. False : Increment imagename
savemodel           =     'none'            #  Options to save model visibilities (none, virtual, modelcolumn)
calcres             =       True            #  Calculate initial residual image
calcpsf             =       True            #  Calculate PSF
parallel            =      False            #  Run major cycles in parallel
```

```
CASA <1>: inp tclean
--------> inp(tclean)
#  tclean :: Radio Interferometric Image Reconstruction
vis                 =              ''            #  Name of input visibility file(s)
selectdata          =            True            #  Enable data selection parameters
     field          =              ''            #  field(s) to select
     spw            =              ''            #  spw(s)/channels to select
     timerange      =              ''            #  Range of time to select from data
     uvrange        =              ''            #  Select data within uvrange
     antenna        =              ''            #  Select data based on antenna/baseline
     scan           =              ''            #  Scan number range
     observation    =              ''            #  Observation ID range
     intent         =              ''            #  Scan Intent(s)

datacolumn          = 'corrected'                #  Data column to image(data,corrected)
imagename           =              ''            #  Pre-name of output images
imsize              =          [100]             #  Number of pixels
cell                = ['1arcsec']                #  Cell size
phasecenter         =              ''            #  Phase center of the image
stokes              =             'I'            #  Stokes Planes to make
projection          =           'SIN'            #  Coordinate projection
startmodel          =              ''            #  Name of starting model image
specmode            =           'mfs'            #  Spectral definition mode (mfs,cube,cubedata, cubesource)
     reffreq        =              ''            #  Reference frequency

gridder             =      'standard'            #  Gridding options (standard, wproject, widefield, mosaic, awproject)
     vptable        =              ''            #  Name of Voltage Pattern table
     pblimit        =             0.2            #  PB gain level at which to cut off normalizations

deconvolver         =       'hogbom'             #  Minor cycle algorithm (hogbom,clark,multiscale,mtmfs,mem,clarkstokes)
restoration         =            True            #  Do restoration steps (or not)
     restoringbeam  =              []            #  Restoring beam shape to use. Default is the PSF main lobe
     pbcor          =           False            #  Apply PB correction on the output restored image

outlierfile         =              ''            #  Name of outlier-field image definitions
weighting           =       'natural'            #  Weighting scheme (natural,uniform,briggs, briggsabs[experimental])
     uvtaper        =              []            #  uv-taper on outer baselines in uv-plane

niter               =               0            #  Maximum number of iterations
usemask             =          'user'            #  Type of mask(s) for deconvolution:  user, pb, or auto-multithresh
     mask           =              ''            #  Mask (a list of image name(s) or region file(s) or region string(s) )
     pbmask         =             0.0            #  primary beam mask

fastnoise           =            True            #  True: use the faster (old) noise calculation. False: use the new improved noise calculations
restart             =            True            #  True : Re-use existing images. False : Increment imagename
savemodel           =          'none'            #  Options to save model visibilities (none, virtual, modelcolumn)
calcres             =            True            #  Calculate initial residual image
calcpsf             =            True            #  Calculate PSF
parallel            =           False            #  Run major cycles in parallel
```

tclean

vis

vis: Name(s) of input visibility file(s)

default: none;

example: vis='ngc5921.ms'
vis=['ngc5921a.ms','ngc5921b.ms']; multiple MSes

tclean

selectdata: field

field: Select fields to image or mosaic. Use field id(s) or name(s).
['go listobs' to obtain the list id's or names]
default: ''= all fields

If field string is a non-negative integer, it is assumed to be a field index, otherwise, it is assumed to be a field name.

example: field = ‘0~2’; field ids 0,1,2
field = '0,4,5~7'; field ids 0,4,5,6,7
field = '3C286,3C295'; field named 3C286 and 3C295
field = '3,4C*'; field id 3, all names starting with 4C

For multiple MS input, a list of field strings can be used:
field = ['0~2','0~4']; field ids 0-2 for the first MS and 0-4 for the second
field = '0~2'; field ids 0-2 for all input MSes

tclean

selectdata: spw

spw: Select spectral window/channels

NOTE: channels de-selected here will contain all zeros if selected by the parameter mode subparameters.

default: '', all spectral windows and channels

example: spw='0~2,4'; spectral windows 0,1,2,4 (all channels)
spw='0:5~61'; spw 0, channels 5 to 61
spw='<2'; spectral windows less than 2 (i.e. 0,1)
spw='0,10,3:3~45'; spw 0,10 all channels, spw 3, channels 3 to 45.
spw='0~2:2~6'; spw 0,1,2 with channels 2 through 6 in each.

For multiple MS input, a list of spw strings can be used:
spw=['0','0~3']; spw ids 0 for the first MS and 0-3 for the second
spw='0~3' spw ids 0-3 for all input MS
spw='3:10~20;50~60' for multiple channel ranges within spw id 3
spw='3:10~20;50~60,4:0~30' for different channel ranges for spw ids 3 and 4
spw='0:0~10,1:20~30,2:1;2;3'; spw 0, channels 0-10, spw 1, channels 20-30,
and spw 2, channels, 1,2 and 3
spw='1~4;6:15~48' for channels 15 through 48 for spw ids 1,2,3,4 and 6

```
CASA <1>: inp tclean
--------> inp(tclean)
#  tclean :: Radio Interferometric Image Reconstruction
vis                 =         ''            #  Name of input visibility file(s)
selectdata          =       True            #  Enable data selection parameters
     field          =         ''            #  field(s) to select
     spw            =         ''            #  spw(s)/channels to select
     timerange      =         ''            #  Range of time to select from data
     uvrange        =         ''            #  Select data within uvrange
     antenna        =         ''            #  Select data based on antenna/baseline
     scan           =         ''            #  Scan number range
     observation    =         ''            #  Observation ID range
     intent         =         ''            #  Scan Intent(s)

datacolumn          = 'corrected'           #  Data column to image(data,corrected)
imagename           =         ''            #  Pre-name of output images
imsize              =      [100]            #  Number of pixels
cell                = ['1arcsec']           #  Cell size
phasecenter         =         ''            #  Phase center of the image
stokes              =        'I'            #  Stokes Planes to make
projection          =      'SIN'            #  Coordinate projection
startmodel          =         ''            #  Name of starting model image
specmode            =      'mfs'            #  Spectral definition mode (mfs,cube,cubedata, cubesource)
     reffreq        =         ''            #  Reference frequency

gridder             =  'standard'           #  Gridding options (standard, wproject, widefield, mosaic, awproject)
     vptable        =         ''            #  Name of Voltage Pattern table
     pblimit        =        0.2            #  PB gain level at which to cut off normalizations

deconvolver         =    'hogbom'           #  Minor cycle algorithm (hogbom,clark,multiscale,mtmfs,mem,clarkstokes)
restoration         =       True            #  Do restoration steps (or not)
     restoringbeam  =         []            #  Restoring beam shape to use. Default is the PSF main lobe
     pbcor          =      False            #  Apply PB correction on the output restored image

outlierfile         =         ''            #  Name of outlier-field image definitions
weighting           =   'natural'           #  Weighting scheme (natural,uniform,briggs, briggsabs[experimental])
     uvtaper        =         []            #  uv-taper on outer baselines in uv-plane

niter               =          0            #  Maximum number of iterations
usemask             =      'user'           #  Type of mask(s) for deconvolution:  user, pb, or auto-multithresh
     mask           =         ''            #  Mask (a list of image name(s) or region file(s) or region string(s) )
     pbmask         =        0.0            #  primary beam mask

fastnoise           =       True            #  True: use the faster (old) noise calculation. False: use the new improved noise calculations
restart             =       True            #  True : Re-use existing images. False : Increment imagename
savemodel           =      'none'           #  Options to save model visibilities (none, virtual, modelcolumn)
calcres             =       True            #  Calculate initial residual image
calcpsf             =       True            #  Calculate PSF
parallel            =      False            #  Run major cycles in parallel
```

tclean

imagename

imagename: Pre-name of output images

default: ''

example : imagename='try'

Output images will be (a subset of) :
- try.psf - Point spread function
- try.residual - Residual image
- try.image - Restored image
- try.model - Model image (contains only flux components)
- try.sumwt - Single pixel image containing sum-of-weights.
  (for natural weighting, sensitivity=1/sqrt(sumwt))
- try.pb - Primary beam model (values depend on the gridder used)

tclean

imagename

Widefield projection algorithms (gridder=mosaic,awproject) will compute the following images too.

try.weight - FT of gridded weights or the un-normalized sum of PB-square
(for all pointings)
Here, PB = sqrt(weight) normalized to a maximum of 1.0

For multi-term wideband imaging, all relevant images above will have additional .tt0,.tt1, etc suffixes to indicate Taylor terms, plus the following extra output images.

try.alpha - spectral index
try.alpha.error - estimate of error on spectral index
try.beta - spectral curvature (if nterms \> 2)

tclean

imagename

Tip : Include a directory name in 'imagename' for all output images to be
sent there instead of the current working directory : imagename='mydir/try'

Tip : Restarting an imaging run without changing 'imagename'
implies continuation from the existing model image on disk.
- If 'startmodel' was initially specified it needs to be set to ""
for the restart run (or tclean will exit with an error message).
- By default, the residual image and psf will be recomputed
but if no changes were made to relevant parameters between
the runs, set calcres=False, calcpsf=False to resume directly from
the minor cycle without the (unnecessary) first major cycle.

To automatically change 'imagename' with a numerical
increment, set restart=False (see tclean docs for 'restart').

Note : All imaging runs will by default produce restored images.
For a niter=0 run, this will be redundant and can optionally
be turned off via the 'restoration=T/F' parameter.

```
CASA <1>: inp tclean
--------> inp(tclean)
#  tclean :: Radio Interferometric Image Reconstruction
vis                 =         ''            #  Name of input visibility file(s)
selectdata          =       True            #  Enable data selection parameters
     field          =         ''            #  field(s) to select
     spw            =         ''            #  spw(s)/channels to select
     timerange      =         ''            #  Range of time to select from data
     uvrange        =         ''            #  Select data within uvrange
     antenna        =         ''            #  Select data based on antenna/baseline
     scan           =         ''            #  Scan number range
     observation    =         ''            #  Observation ID range
     intent         =         ''            #  Scan Intent(s)

datacolumn          = 'corrected'           #  Data column to image(data,corrected)
imagename           =         ''            #  Pre-name of output images
imsize              =      [100]            #  Number of pixels
cell                = ['1arcsec']           #  Cell size
phasecenter         =         ''            #  Phase center of the image
stokes              =        'I'            #  Stokes Planes to make
projection          =      'SIN'            #  Coordinate projection
startmodel          =         ''            #  Name of starting model image
specmode            =      'mfs'            #  Spectral definition mode (mfs,cube,cubedata, cubesource)
     reffreq        =         ''            #  Reference frequency

gridder             =  'standard'           #  Gridding options (standard, wproject, widefield, mosaic, awproject)
     vptable        =         ''            #  Name of Voltage Pattern table
     pblimit        =        0.2            #  PB gain level at which to cut off normalizations

deconvolver         =    'hogbom'           #  Minor cycle algorithm (hogbom,clark,multiscale,mtmfs,mem,clarkstokes)
restoration         =       True            #  Do restoration steps (or not)
     restoringbeam  =         []            #  Restoring beam shape to use. Default is the PSF main lobe
     pbcor          =      False            #  Apply PB correction on the output restored image

outlierfile         =         ''            #  Name of outlier-field image definitions
weighting           =   'natural'           #  Weighting scheme (natural,uniform,briggs, briggsabs[experimental])
     uvtaper        =         []            #  uv-taper on outer baselines in uv-plane

niter               =          0            #  Maximum number of iterations
usemask             =     'user'            #  Type of mask(s) for deconvolution:  user, pb, or auto-multithresh
     mask           =         ''            #  Mask (a list of image name(s) or region file(s) or region string(s) )
     pbmask         =        0.0            #  primary beam mask

fastnoise           =       True            #  True: use the faster (old) noise calculation. False: use the new improved noise calculations
restart             =       True            #  True : Re-use existing images. False : Increment imagename
savemodel           =     'none'            #  Options to save model visibilities (none, virtual, modelcolumn)
calcres             =       True            #  Calculate initial residual image
calcpsf             =       True            #  Calculate PSF
parallel            =      False            #  Run major cycles in parallel
```

tclean

imsize

imsize: Number of pixels

default: [100]

example: imsize = [350,250]

imsize = 500 is equivalent to [500,500]

To take proper advantage of internal optimized FFT routines, the number of pixels must be even and factorizable by 2,3,5,7 only.

tclean

cell

cell: pixel size

default: [‘1arcsec’]

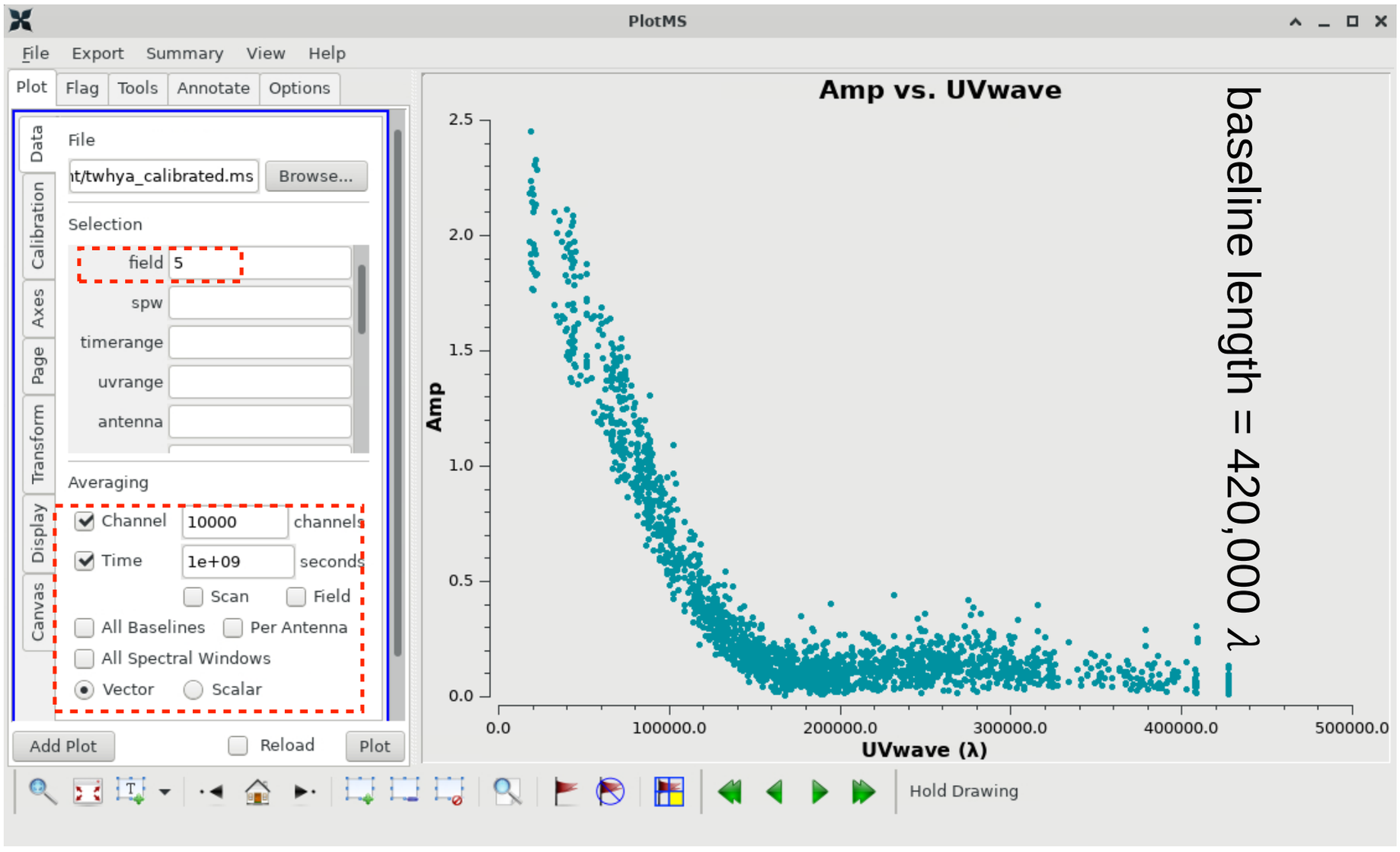
example: cell=['0.5arcsec,'0.5arcsec']
cell=['1arcmin', '1arcmin']
cell = '1arcsec' is equivalent to ['1arcsec','1arcsec']

Determine cell size and image size

simple math…

- Cell size (aka pixel size) is related to the clean beam size. Usually we use 5~7 pixels to sample the short axis of the elliptical beam.

- Estimate the clean beam size with plotms



Estimate cell size and image size

simple math…

speed of light

frequency

$$c = \lambda\nu$$

wavelength

$$\lambda = \frac{c}{\nu} \longrightarrow 0.8\ \text{mm}$$

angular resolution (aka “beam”)

$$\theta \sim \frac{\lambda}{B} \longrightarrow 1/420{,}000\ \text{rad} = 0.5'' \rightarrow \text{cell}=\text{‘0.1arcsec’}$$

baseline length = 420,000 λ

field of view (aka “primary beam”)

$$\text{FOV} \sim 1.22\frac{\lambda}{\text{D}} \longrightarrow 17'' \longrightarrow \text{imsize}=[340]$$

dish size

2x FOV

```
CASA <1>: inp tclean
--------> inp(tclean)
#  tclean :: Radio Interferometric Image Reconstruction
vis                 =         ''            #  Name of input visibility file(s)
selectdata          =       True            #  Enable data selection parameters
     field          =         ''            #  field(s) to select
     spw            =         ''            #  spw(s)/channels to select
     timerange      =         ''            #  Range of time to select from data
     uvrange        =         ''            #  Select data within uvrange
     antenna        =         ''            #  Select data based on antenna/baseline
     scan           =         ''            #  Scan number range
     observation    =         ''            #  Observation ID range
     intent         =         ''            #  Scan Intent(s)

datacolumn          = 'corrected'           #  Data column to image(data,corrected)
imagename           =         ''            #  Pre-name of output images
imsize              =      [100]            #  Number of pixels
cell                = ['1arcsec']           #  Cell size
phasecenter         =         ''            #  Phase center of the image
stokes              =        'I'            #  Stokes Planes to make
projection          =      'SIN'            #  Coordinate projection
startmodel          =         ''            #  Name of starting model image
specmode            =      'mfs'            #  Spectral definition mode (mfs,cube,cubedata, cubesource)
     reffreq        =         ''            #  Reference frequency

gridder             =  'standard'           #  Gridding options (standard, wproject, widefield, mosaic, awproject)
     vptable        =         ''            #  Name of Voltage Pattern table
     pblimit        =        0.2            #  PB gain level at which to cut off normalizations

deconvolver         =   'hogbom'            #  Minor cycle algorithm (hogbom,clark,multiscale,mtmfs,mem,clarkstokes)
restoration         =       True            #  Do restoration steps (or not)
     restoringbeam  =         []            #  Restoring beam shape to use. Default is the PSF main lobe
     pbcor          =      False            #  Apply PB correction on the output restored image

outlierfile         =         ''            #  Name of outlier-field image definitions
weighting           =  'natural'            #  Weighting scheme (natural,uniform,briggs, briggsabs[experimental])
     uvtaper        =         []            #  uv-taper on outer baselines in uv-plane

niter               =          0            #  Maximum number of iterations
usemask             =     'user'            #  Type of mask(s) for deconvolution:  user, pb, or auto-multithresh
     mask           =         ''            #  Mask (a list of image name(s) or region file(s) or region string(s) )
     pbmask         =        0.0            #  primary beam mask

fastnoise           =       True            #  True: use the faster (old) noise calculation. False: use the new improved noise calculations
restart             =       True            #  True : Re-use existing images. False : Increment imagename
savemodel           =     'none'            #  Options to save model visibilities (none, virtual, modelcolumn)
calcres             =       True            #  Calculate initial residual image
calcpsf             =       True            #  Calculate PSF
parallel            =      False            #  Run major cycles in parallel
```

tclean

specmode: mfs, cube

specmode: Spectral definition mode (mfs, cont, cube, cubedata,cubesource)

mode=‘mfs' : Continuum imaging with only one output image channel.
(mode='cont' can also be used here)

mode='cube' : Spectral line imaging with one or more channels

Parameters start, width, and nchan define the spectral coordinate system and can be specified either in terms of channel numbers, frequency or velocity in whatever spectral frame is specified in ‘outframe’. All internal and output images are made with outframe as the base spectral frame. However imaging code internally uses the fixed spectral frame, LSRK, for automatic internal software Doppler tracking so that a spectral line observed over an extended time range will line up appropriately. Therefore the output images have additional spectral frame conversion layer in LSRK on the top the base frame.

```
CASA <1>: inp tclean
--------> inp(tclean)
#  tclean :: Radio Interferometric Image Reconstruction
vis                 =         ''            #  Name of input visibility file(s)
selectdata          =       True            #  Enable data selection parameters
     field          =         ''            #  field(s) to select
     spw            =         ''            #  spw(s)/channels to select
     timerange      =         ''            #  Range of time to select from data
     uvrange        =         ''            #  Select data within uvrange
     antenna        =         ''            #  Select data based on antenna/baseline
     scan           =         ''            #  Scan number range
     observation    =         ''            #  Observation ID range
     intent         =         ''            #  Scan Intent(s)

datacolumn          = 'corrected'           #  Data column to image(data,corrected)
imagename           =         ''            #  Pre-name of output images
imsize              =      [100]            #  Number of pixels
cell                = ['1arcsec']           #  Cell size
phasecenter         =         ''            #  Phase center of the image
stokes              =        'I'            #  Stokes Planes to make
projection          =      'SIN'            #  Coordinate projection
startmodel          =         ''            #  Name of starting model image
specmode            =      'mfs'            #  Spectral definition mode (mfs,cube,cubedata, cubesource)
     reffreq        =         ''            #  Reference frequency

gridder             =  'standard'           #  Gridding options (standard, wproject, widefield, mosaic, awproject)
     vptable        =         ''            #  Name of Voltage Pattern table
     pblimit        =        0.2            #  PB gain level at which to cut off normalizations

deconvolver         =    'hogbom'           #  Minor cycle algorithm (hogbom,clark,multiscale,mtmfs,mem,clarkstokes)
restoration         =       True            #  Do restoration steps (or not)
     restoringbeam  =         []            #  Restoring beam shape to use. Default is the PSF main lobe
     pbcor          =      False            #  Apply PB correction on the output restored image

outlierfile         =         ''            #  Name of outlier-field image definitions
weighting           =  'natural'            #  Weighting scheme (natural,uniform,briggs, briggsabs[experimental])
     uvtaper        =         []            #  uv-taper on outer baselines in uv-plane

niter               =          0            #  Maximum number of iterations
usemask             =     'user'            #  Type of mask(s) for deconvolution:  user, pb, or auto-multithresh
     mask           =         ''            #  Mask (a list of image name(s) or region file(s) or region string(s) )
     pbmask         =        0.0            #  primary beam mask

fastnoise           =       True            #  True: use the faster (old) noise calculation. False: use the new improved noise calculations
restart             =       True            #  True : Re-use existing images. False : Increment imagename
savemodel           =     'none'            #  Options to save model visibilities (none, virtual, modelcolumn)
calcres             =       True            #  Calculate initial residual image
calcpsf             =       True            #  Calculate PSF
parallel            =      False            #  Run major cycles in parallel
```

tclean

deconvolver: hogbom

deconvolver: Name of minor cycle algorithm (hogbom,clark,multiscale,mtmfs,mem,clarkstokes)

Each of the following algorithms operate on residual images and psfs from the gridder and produce output model and restored images. Minor cycles stop and a major cycle is triggered when cyclethreshold or cycleniter are reached. For all methods, components are picked from the entire extent of the image or (if specified) within a mask.

hogbom : An adapted version of Hogbom Clean [Hogbom, 1974]

- Find the location of the peak residual
- Add this delta function component to the model image
- Subtract a scaled and shifted PSF of the same size as the image from regions of the residual image where the two overlap.
- Repeat

Comparison of different algorithms

a not apple-to-apple example

maximum entropy method

multi-scale CLEAN

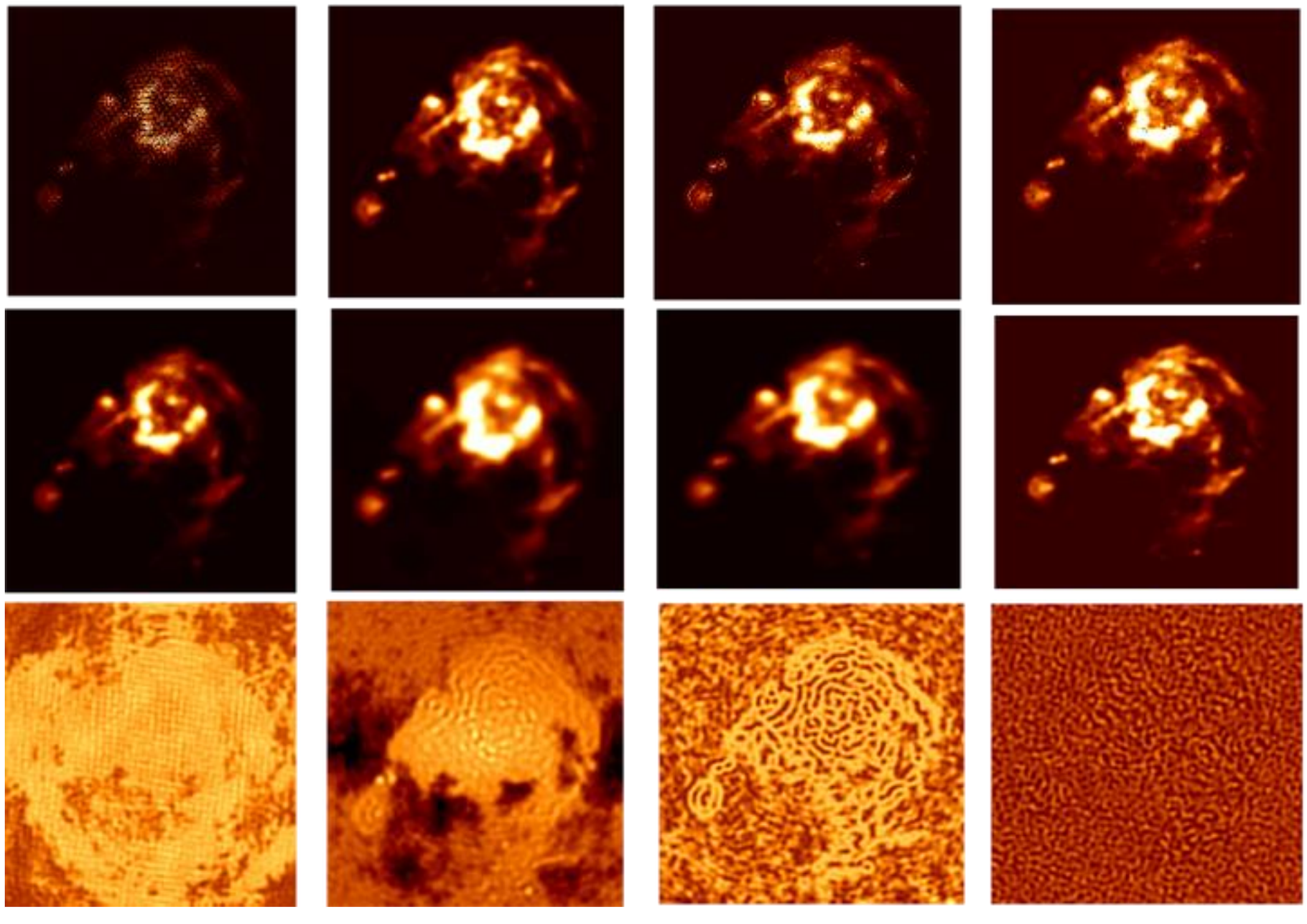
adaptive scale pixel

CLEAN

MEM

MS-CLEAN

ASP



I^m Sky model

I^{out} After clean

I^{res} Residual

```
CASA <1>: inp tclean
--------> inp(tclean)
#  tclean :: Radio Interferometric Image Reconstruction
vis                 =         ''            #  Name of input visibility file(s)
selectdata          =       True            #  Enable data selection parameters
     field          =         ''            #  field(s) to select
     spw            =         ''            #  spw(s)/channels to select
     timerange      =         ''            #  Range of time to select from data
     uvrange        =         ''            #  Select data within uvrange
     antenna        =         ''            #  Select data based on antenna/baseline
     scan           =         ''            #  Scan number range
     observation    =         ''            #  Observation ID range
     intent         =         ''            #  Scan Intent(s)

datacolumn          = 'corrected'           #  Data column to image(data,corrected)
imagename           =         ''            #  Pre-name of output images
imsize              =      [100]            #  Number of pixels
cell                = ['1arcsec']           #  Cell size
phasecenter         =         ''            #  Phase center of the image
stokes              =        'I'            #  Stokes Planes to make
projection          =      'SIN'            #  Coordinate projection
startmodel          =         ''            #  Name of starting model image
specmode            =      'mfs'            #  Spectral definition mode (mfs,cube,cubedata, cubesource)
     reffreq        =         ''            #  Reference frequency

gridder             =  'standard'           #  Gridding options (standard, wproject, widefield, mosaic, awproject)
     vptable        =         ''            #  Name of Voltage Pattern table
     pblimit        =        0.2            #  PB gain level at which to cut off normalizations

deconvolver         =    'hogbom'           #  Minor cycle algorithm (hogbom,clark,multiscale,mtmfs,mem,clarkstokes)
restoration         =       True            #  Do restoration steps (or not)
     restoringbeam  =         []            #  Restoring beam shape to use. Default is the PSF main lobe
     pbcor          =      False            #  Apply PB correction on the output restored image

outlierfile         =         ''            #  Name of outlier-field image definitions
weighting           =   'natural'           #  Weighting scheme (natural,uniform,briggs, briggsabs[experimental])
     uvtaper        =         []            #  uv-taper on outer baselines in uv-plane

niter               =          0            #  Maximum number of iterations
usemask             =     'user'            #  Type of mask(s) for deconvolution:  user, pb, or auto-multithresh
     mask           =         ''            #  Mask (a list of image name(s) or region file(s) or region string(s) )
     pbmask         =        0.0            #  primary beam mask

fastnoise           =       True            #  True: use the faster (old) noise calculation. False: use the new improved noise calculations
restart             =       True            #  True : Re-use existing images. False : Increment imagename
savemodel           =     'none'            #  Options to save model visibilities (none, virtual, modelcolumn)
calcres             =       True            #  Calculate initial residual image
calcpsf             =       True            #  Calculate PSF
parallel            =      False            #  Run major cycles in parallel
```

tclean

weighting: natural

weighting: Weighting scheme
(natural, uniform, briggs, superuniform, radial, briggsabs)

During gridding of the dirty or residual image, each visibility value is multiplied by a weight before it is accumulated on the uv-grid.
The PSF's uv-grid is generated by gridding only the weights (weightgrid).

weighting='natural' : Gridding weights are identical to the data weights
from the MS. For visibilities with similar data weights,
the weighted grid will follow the sample density
pattern on the uv-plane. This weighting scheme
provides the maximum imaging sensitivity at the
expense of a possibly fat PSF with high sidelobes.
It is most appropriate for detection experiments
where sensitivity is most important.

tclean

weighting: uniform

weighting='uniform' : Gridding weights per visibility data point are the original data weights divided by the total weight of all data points that map to the same uv grid cell : ' data_weight / total_wt_per_cell '.

The weightgrid is as close to flat as possible resulting in a PSF with a narrow main lobe and suppressed sidelobes. However, since heavily sampled areas of the uv-plane get down-weighted, the imaging sensitivity is not as high as with natural weighting. It is most appropriate for imaging experiments where a well behaved PSF can help the reconstruction.

tclean

weighting: briggs

weighting='briggs' : Gridding weights per visibility data point are given by
'data_weight / (A *total_wt_per_cell + B) ' where
A and B vary according to the 'robust' parameter.

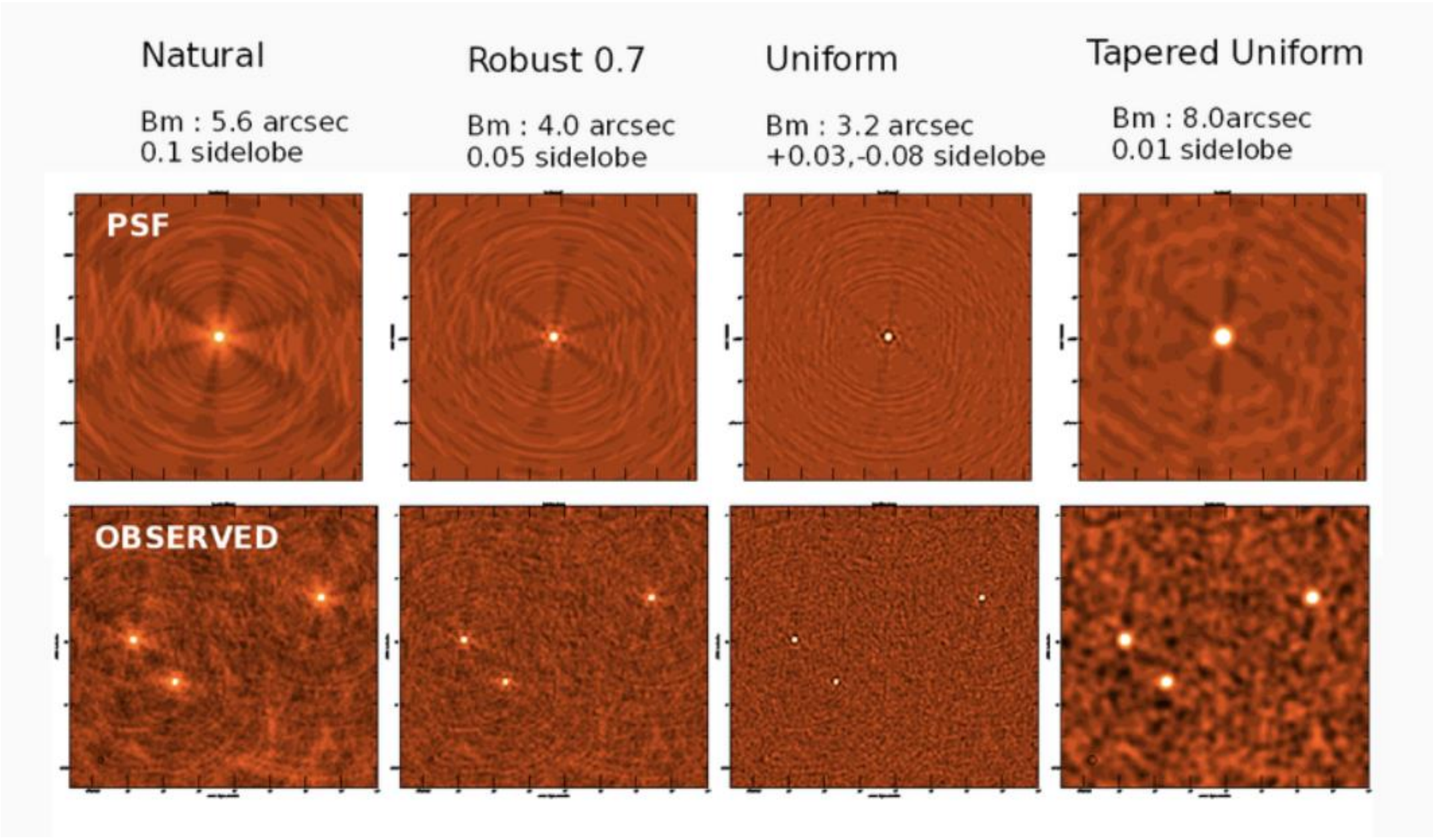
robust = -2.0 maps to A=1,B=0 or uniform weighting.
robust = +2.0 maps to natural weighting.
(robust=0.5 is equivalent to robust=0.0 in AIPS IMAGR.)

Robust/Briggs weighting generates a PSF that can
vary smoothly between 'natural' and 'uniform' and
allow customized trade-offs between PSF shape and
imaging sensitivity.

For more details on weighting please see Chapter3
of Dan Briggs' thesis (http://www.aoc.nrao.edu/dissertations/dbriggs)

Comparing different wt schemes

Beam sizes and sidelobs



tclean

uvtaper

uvtaper: uv-taper on outer baselines in uv-plane

Apply a Gaussian taper in addition to the weighting scheme specified
via the 'weighting' parameter. Higher spatial frequencies are weighted
down relative to lower spatial frequencies to suppress artifacts
arising from poorly sampled areas of the uv-plane. It is equivalent to
smoothing the PSF obtained by other weighting schemes and can be
specified either as a Gaussian in uv-space (eg. units of lambda)
or as a Gaussian in the image domain (eg. angular units like arcsec).

uvtaper = [bmaj, bmin, bpa]

NOTE: the on-sky FWHM in arcsec is roughly the uv taper/200 (klambda).

default: uvtaper=[]; no Gaussian taper applied

example: uvtaper=['5klambda'] circular taper FWHM=5 kilo-lambda
uvtaper=['5klambda','3klambda','45.0deg']
uvtaper=['10arcsec'] on-sky FWHM 10 arcseconds
uvtaper=['300.0'] default units are lambda in aperture plane

```
CASA <1>: inp tclean
--------> inp(tclean)
#  tclean :: Radio Interferometric Image Reconstruction
vis                 =         ''            #  Name of input visibility file(s)
selectdata          =       True            #  Enable data selection parameters
     field          =         ''            #  field(s) to select
     spw            =         ''            #  spw(s)/channels to select
     timerange      =         ''            #  Range of time to select from data
     uvrange        =         ''            #  Select data within uvrange
     antenna        =         ''            #  Select data based on antenna/baseline
     scan           =         ''            #  Scan number range
     observation    =         ''            #  Observation ID range
     intent         =         ''            #  Scan Intent(s)

datacolumn          = 'corrected'           #  Data column to image(data,corrected)
imagename           =         ''            #  Pre-name of output images
imsize              =      [100]            #  Number of pixels
cell                = ['1arcsec']           #  Cell size
phasecenter         =         ''            #  Phase center of the image
stokes              =        'I'            #  Stokes Planes to make
projection          =      'SIN'            #  Coordinate projection
startmodel          =         ''            #  Name of starting model image
specmode            =      'mfs'            #  Spectral definition mode (mfs,cube,cubedata, cubesource)
     reffreq        =         ''            #  Reference frequency

gridder             = 'standard'            #  Gridding options (standard, wproject, widefield, mosaic, awproject)
     vptable        =         ''            #  Name of Voltage Pattern table
     pblimit        =        0.2            #  PB gain level at which to cut off normalizations

deconvolver         =   'hogbom'            #  Minor cycle algorithm (hogbom,clark,multiscale,mtmfs,mem,clarkstokes)
restoration         =       True            #  Do restoration steps (or not)
     restoringbeam  =         []            #  Restoring beam shape to use. Default is the PSF main lobe
     pbcor          =      False            #  Apply PB correction on the output restored image

outlierfile         =         ''            #  Name of outlier-field image definitions
weighting           =  'natural'            #  Weighting scheme (natural,uniform,briggs, briggsabs[experimental])
     uvtaper        =         []            #  uv-taper on outer baselines in uv-plane

niter               =          0            #  Maximum number of iterations
usemask             =     'user'            #  Type of mask(s) for deconvolution:  user, pb, or auto-multithresh
     mask           =         ''            #  Mask (a list of image name(s) or region file(s) or region string(s) )
     pbmask         =        0.0            #  primary beam mask

fastnoise           =       True            #  True: use the faster (old) noise calculation. False: use the new improved noise calculations
restart             =       True            #  True : Re-use existing images. False : Increment imagename
savemodel           =     'none'            #  Options to save model visibilities (none, virtual, modelcolumn)
calcres             =       True            #  Calculate initial residual image
calcpsf             =       True            #  Calculate PSF
parallel            =      False            #  Run major cycles in parallel
```

If niter=0, we set restoration=False because ‘dirty map’ and ‘clean map’ are the same images.

tclean

niter

niter: Maximum number of iterations

A stopping criterion based on total iteration count.
Currently the parameter type is defined as an integer therefore the integer value larger than 2147483647 will not be set properly as it causes an overflow.

Iterations are typically defined as the selecting one flux component
and partially subtracting it out from the residual image.

niter=0 : Do only the initial major cycle (make dirty image, psf, pb, etc)

niter larger than zero : Run major and minor cycles.

tclean

niter

Note : Global stopping criteria vs major-cycle triggers

In addition to global stopping criteria, the following rules are
used to determine when to terminate a set of minor cycle iterations
and trigger major cycles [derived from Cotton-Schwab Clean, 1984]

'cycleniter' : controls the maximum number of iterations per image
plane before triggering a major cycle.
'cyclethreshold' : Automatically computed threshold related to the
max sidelobe level of the PSF and peak residual.
Divergence, detected as an increase of 10% in peak residual from the
minimum so far (during minor cycle iterations)

The first criterion to be satisfied takes precedence.

Note : Iteration counts for cubes or multi-field images :
For images with multiple planes (or image fields) on which the
deconvolver operates in sequence, iterations are counted across
all planes (or image fields). The iteration count is compared with
'niter' only after all channels/planes/fields have completed their
minor cycles and exited either due to 'cycleniter' or 'cyclethreshold'.
Therefore, the actual number of iterations reported in the logger
can sometimes be larger than the user specified value in 'niter'.
For example, with niter=100, cycleniter=20,nchan=10,threshold=0,
a total of 200 iterations will be done in the first set of minor cycles
before the total is compared with niter=100 and it exits.

Note : Additional global stopping criteria include
- no change in peak residual across two major cycles
- a 50% or more increase in peak residual across one major cycle

```
deconvolver          =    'hogbom'        #  Minor cycle algorithm (hogbom,clark,multiscale,mtmfs,mem,clarkstokes)
restoration          =       True         #  Do restoration steps (or not)
     restoringbeam   =         []         #  Restoring beam shape to use. Default is the PSF main lobe
     pbcor           =      False         #  Apply PB correction on the output restored image

outlierfile          =         ''         #  Name of outlier-field image definitions
weighting            =  'natural'         #  Weighting scheme (natural,uniform,briggs, briggsabs[experimental])
     uvtaper         =         []         #  uv-taper on outer baselines in uv-plane

niter                =          0         #  Maximum number of iterations
usemask              =     'user'         #  Type of mask(s) for deconvolution:  user, pb, or auto-multithresh
     mask            =         ''         #  Mask (a list of image name(s) or region file(s) or region string(s) )
     pbmask          =        0.0         #  primary beam mask

fastnoise            =       True         #  True: use the faster (old) noise calculation. False: use the new improved noise calculations
restart              =       True         #  True : Re-use existing images. False : Increment imagename
savemodel            =     'none'         #  Options to save model visibilities (none, virtual, modelcolumn)
calcres              =       True         #  Calculate initial residual image
calcpsf              =       True         #  Calculate PSF
parallel             =      False         #  Run major cycles in parallel
```

If niter!=0, new parameters will appear.

```
niter                =        100         #  Maximum number of iterations
     gain            =        0.1         #  Loop gain
     threshold       = '0.0006Jy'         #  Stopping threshold
     nsigma          =        0.0         #  Multiplicative factor for rms-based threshold stopping
     cycleniter      =         -1         #  Maximum number of minor-cycle iterations
     cyclefactor     =        1.0         #  Scaling on PSF sidelobe level to compute the minor-cycle stopping threshold.
     minpsffraction  =       0.05         #  PSF fraction that marks the max depth of cleaning in the minor cycle
     maxpsffraction  =        0.8         #  PSF fraction that marks the minimum depth of cleaning in the minor cycle
     interactive     =      False         #  Modify masks and parameters at runtime
```

tclean

threshold

threshold: Stopping threshold (number in units of Jy, or string)

A global stopping threshold that the peak residual (within clean mask)
across all image planes is compared to.

threshold = 0.005 : 5mJy
threshold = '5.0mJy'

Note : A 'cyclethreshold' is internally computed and used as a major cycle
trigger. It is related what fraction of the PSF can be reliably
used during minor cycle updates of the residual image. By default
the minor cycle iterations terminate once the peak residual reaches
the first sidelobe level of the brightest source.

'cyclethreshold' is computed as follows using the settings in
parameters 'cyclefactor','minpsffraction','maxpsffraction','threshold' :

psf_fraction = max_psf_sidelobe_level * 'cyclefactor'
psf_fraction = max(psf_fraction, 'minpsffraction');
psf_fraction = min(psf_fraction, 'maxpsffraction');
cyclethreshold = peak_residual * psf_fraction
cyclethreshold = max(cyclethreshold, 'threshold')

If nsigma is set (>0.0), the N-sigma threshold is calculated (see
the description under nsigma), then cyclethreshold is further modified as,

cyclethreshold = max(cyclethreshold, nsgima_threshold)

'cyclethreshold' is made visible and editable only in the
interactive GUI when tclean is run with interactive=True.

Review all parameters before ‘go’

After ‘go’

pay attention to the logger

```
Completed 2042 iterations.
---------------------------------------------------------- Run Major Cycle 5 ----------------------------
 Absolute Peak residual within mask : 0.00056293, over full image : 0.0152164
Minor Cycle controls : {'cycleniter': 2958, 'cyclethreshold': 0.00010189844761043787, 'loopgain': 0.10000000149
[phase_cal_uniform] Run Hogbom minor-cycle | CycleThreshold=0.000101898, CycleNiter=2958, Gain=0.1
[phase_cal_uniform] iters=0->2958 [2958], model=0.620792->0.619962, peakres=0.00056293->0.000337509, Reached cy
Completed 5000 iterations.
---------------------------------------------------------- Run Major Cycle 6 ----------------------------
 Absolute Peak residual within mask : 0.000337502, over full image : 0.0152439
Reached global stopping criterion : iteration limit
getSummary call: fullsummary=False
[phase_cal_uniform] : Restoring model image.
Beam for chan : 0 : 0.499474 arcsec, 0.395111 arcsec, -44.3226 deg


Completed 84 iterations.
---------------------------------------------------------- Run Major Cycle 2 ----------------------------
 Absolute Peak residual within mask : 0.0536286, over full image : 0.0536286
Minor Cycle controls : {'cycleniter': 9916, 'cyclethreshold': 0.014999999664723873, 'loopgain': 0.10000
[twhya_cont] Run Hogbom minor-cycle | CycleThreshold=0.015, CycleNiter=9916, Gain=0.1
[twhya_cont] iters=0->263 [263], model=0.74245->1.49335, peakres=0.0536286->0.0149695, Reached cyclethr
Completed 347 iterations.
---------------------------------------------------------- Run Major Cycle 3 ----------------------------
 Absolute Peak residual within mask : 0.0149695, over full image : 0.020731
Reached global stopping criterion : threshold
getSummary call: fullsummary=False
[twhya_cont] : Restoring model image.
Beam for chan : 0 : 0.560291 arcsec, 0.415713 arcsec, -55.9733 deg
Searching for images with prefix 'twhya_cont'... Found these, writing history into them: ['twhya_cont.ps
```

Generated images after tclean

check your working directory

- After tclean finished, we will see that new images are generated
  - twhya_cont.image — Cleaned and restored image (Jy/beam)
  - twhya_cont.model — Clean component (model) image (Jy/pixel)
  - twhya_cont.pb — Primary beam model
  - twhya_cont.psf — Dirty beam
  - twhya_cont.residual — Residual (Jy/beam)
  - twhya_cont.sumwt — Sum of weights

- Try to visualize them with casa viewer or CARTA

- To run CARTA
  > cd <your holder>/casaImaging
  > ./CARTA.AppImage --remote
  Copy and paste the url virgo**** to your local web browser (Chrome, Firefox, Safari)

- download CARTA at https://cartavis.github.io

Test dataset

```
├── MyTutorial_cont
│   ├── tclean_tutorial_cont_pre.py
│   ├── tclean_tutorial_cont.py
│   ├── twhya_calibrated.ms
│   └── twhya_calibrated.ms.tar.gz
```

- inside CASA: execfile('tclean_tutorial_cont.py')

Image visualization

Dirty images when niter=0

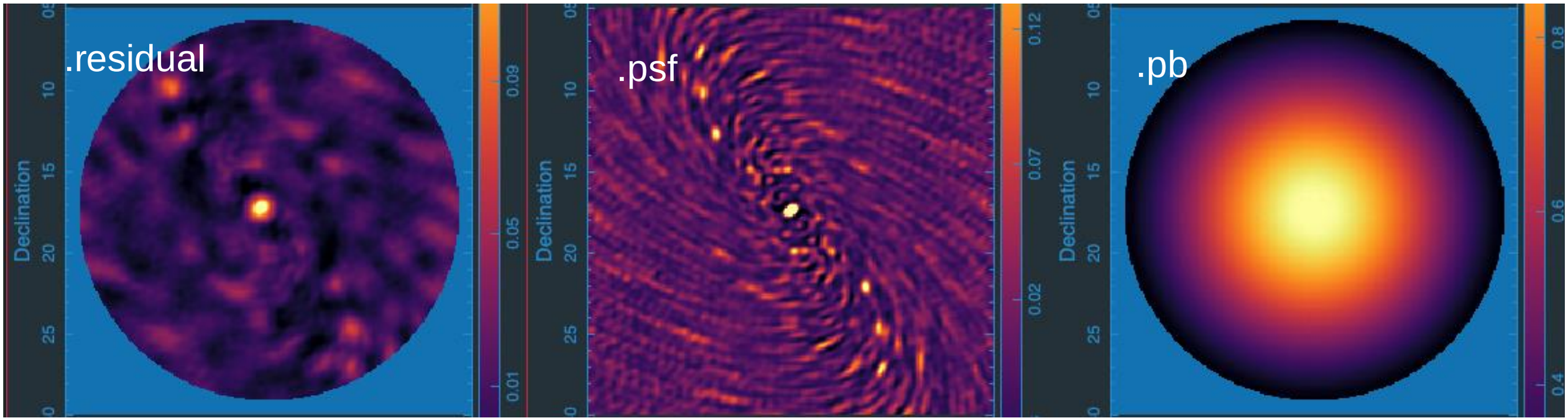


Image visualization

Clean-ed images

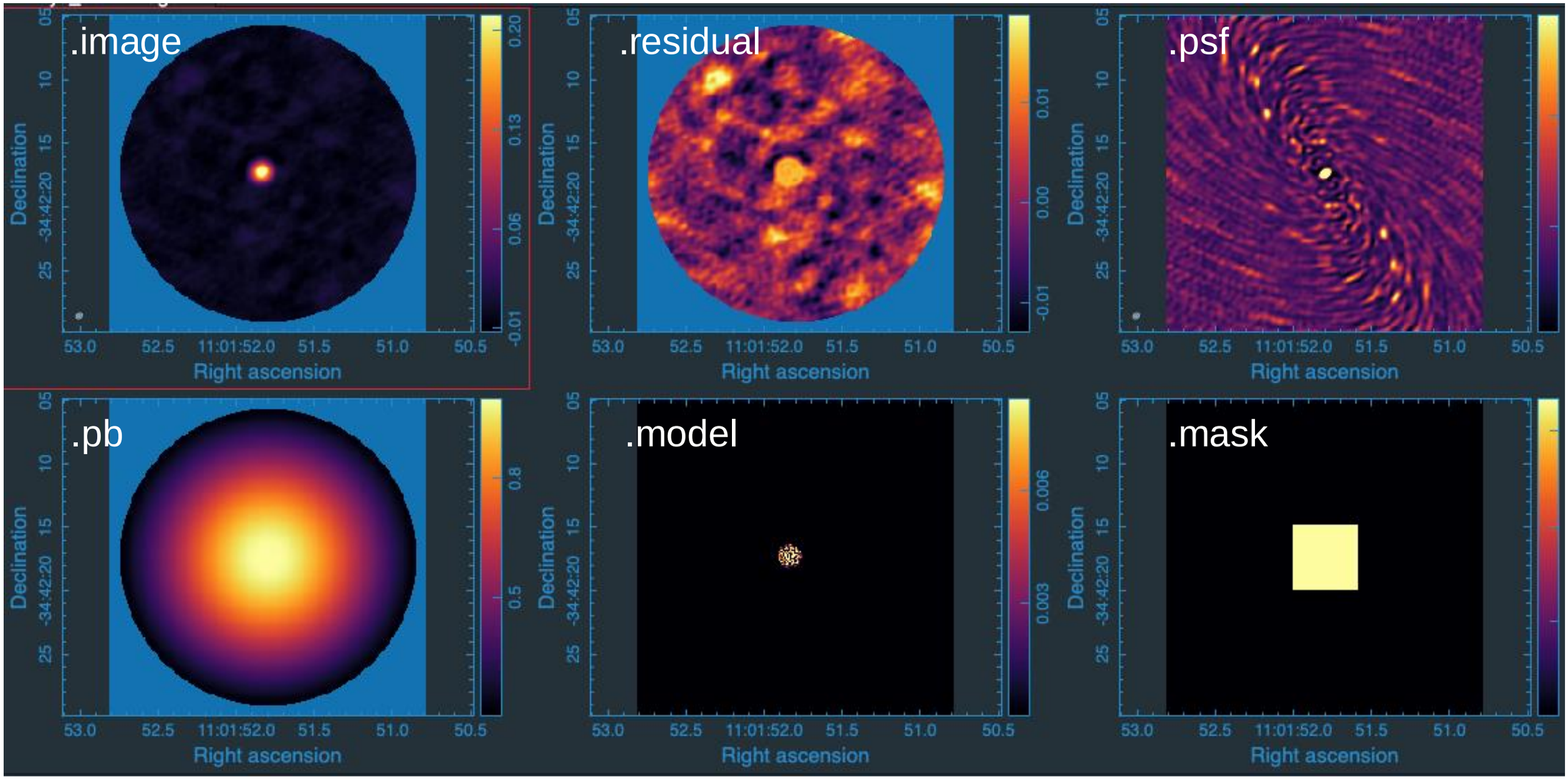
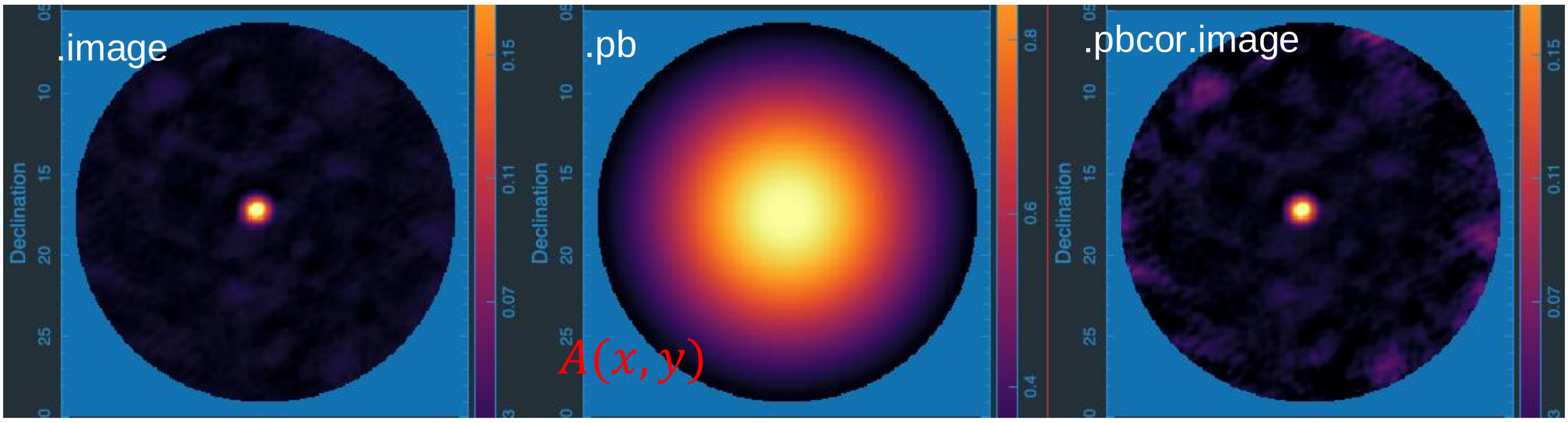


Image visualization

Primary-beam corrected image = *.image/*.pb

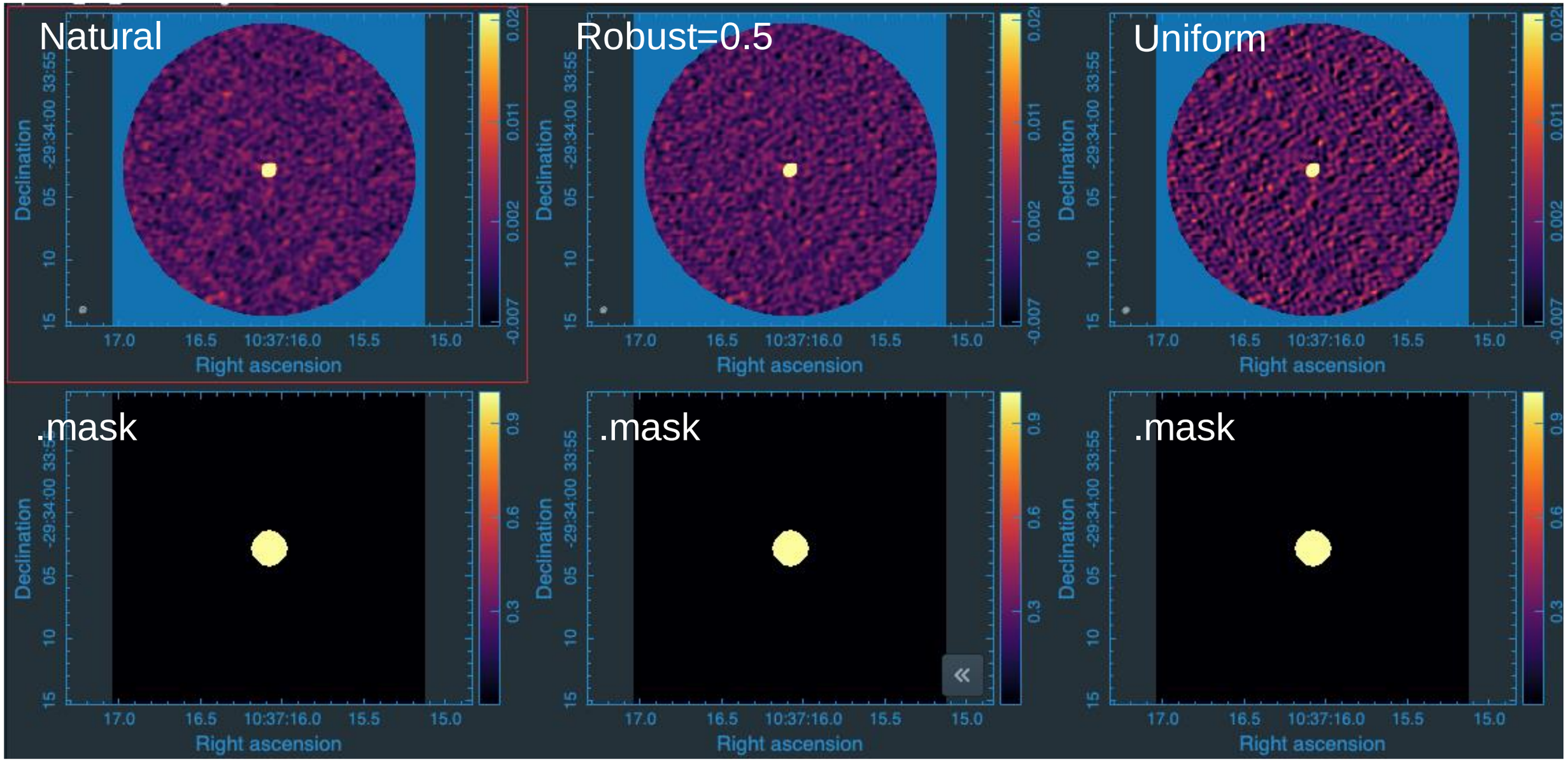


$$V_{\text{obs}}(u,v) = S(u,v) \iint I_{\text{true}}(x,y) e^{-2\pi i(ux+vy)} dxdy$$

$$\Rightarrow V_{\text{obs}}(u,v) = S(u,v) \iint A(x,y) I_{\text{true}}(x,y) e^{-2\pi i(ux+vy)} dxdy$$

Image visualization

Phase calibrator with different weighting



Tasks for today

- A continuum image
- **A line cube**

```
if 1:
    listobs(vis='twhya_selfcal.ms', listfile='twhya_selfcal_listobs.txt')

    os.system('rm -rf twhya_selfcal.ms.contsub')
    uvcontsub(vis = 'twhya_selfcal.ms',
              field = '5',
              fitspec = '0:0~239;281~383',
              fitorder = 0,
              outputvis='twhya_selfcal.ms.contsub')

if 0:
    plotms(vis='twhya_selfcal.ms',
           xaxis='channel',
           yaxis='amp',
           field='5',
           avgspw=False,
           avgtime='1e9',
           avgscan=True,
           avgbaseline=True,
           showgui = True)

if 0:
    plotms(vis='twhya_selfcal.ms.contsub',
           xaxis='channel',
           yaxis='amp',
           field='5',
           avgspw=False,
           avgtime='1e9',
           avgscan=True,
           avgbaseline=True,
           showgui = True)
```

Test dataset

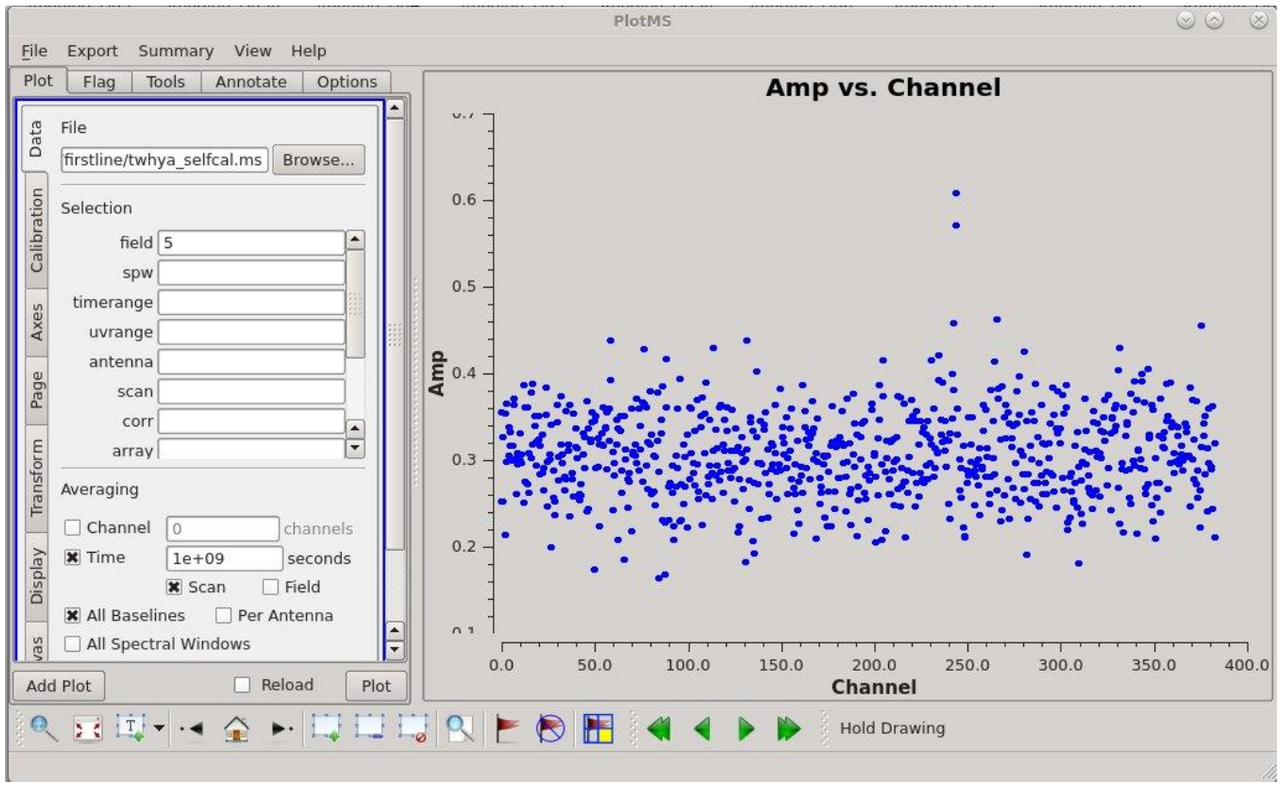
```
└── MyTutorial_line
    ├── tclean_tutorial_line_pre.py
    ├── tclean_tutorial_line.py
    ├── twhya_selfcal.ms
    └── twhya_selfcal.ms.tar.gz
```

- inside CASA: execfile(‘tclean_tutorial_line_pre.py’)

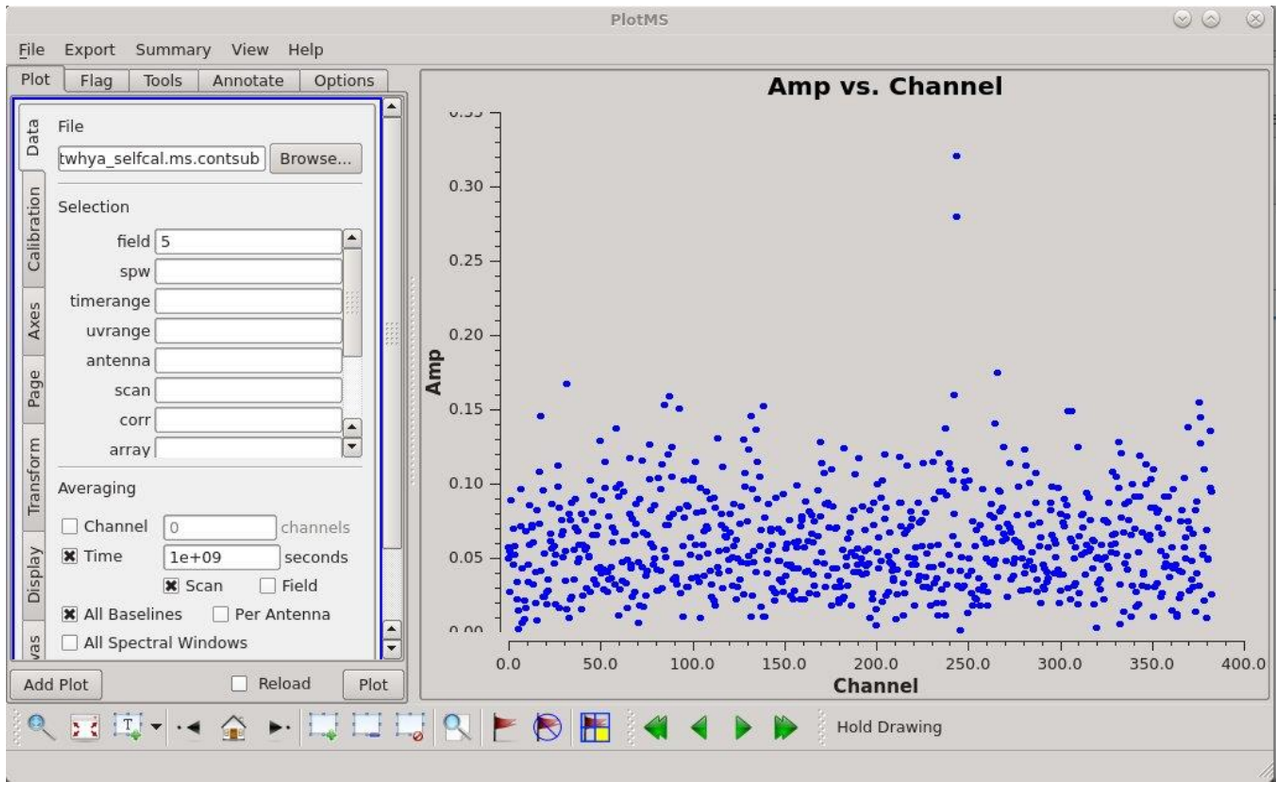
Continuum subtraction

In the uv domain

Before



After



Test dataset

```
└── MyTutorial_line
    ├── tclean_tutorial_line_pre.py
    ├── tclean_tutorial_line.py
    ├── twhya_selfcal.ms
    └── twhya_selfcal.ms.tar.gz
```

- inside CASA: execfile(‘tclean_tutorial_line.py’)

tclean

specmode: mfs, cube

specmode: Spectral definition mode (mfs, cont, cube, cubedata,cubesource)

mode=‘mfs' : Continuum imaging with only one output image channel.
(mode='cont' can also be used here)

mode='cube' : Spectral line imaging with one or more channels

Parameters start, width, and nchan define the spectral coordinate system and can be specified either in terms of channel numbers, frequency or velocity in whatever spectral frame is specified in ‘outframe’. All internal and output images are made with outframe as the base spectral frame. However imaging code internally uses the fixed spectral frame, LSRK, for automatic internal software Doppler tracking so that a spectral line observed over an extended time range will line up appropriately. Therefore the output images have additional spectral frame conversion layer in LSRK on the top the base frame.

```
specmode            =       'mfs'        #  Spectral definition mode (mfs,cube,cubedata, cubesource)
     reffreq        =            ''      #  Reference frequency
```

```
specmode            =      'cube'        #  Spectral definition mode (mfs,cube,cubedata, cubesource)
     nchan          =          -1        #  Number of channels in the output image
     start          =          ''        #  First channel (e.g. start=3,start='1.1GHz',start='15343km/s')
     width          =          ''        #  Channel width (e.g. width=2,width='0.1MHz',width='10km/s')
     outframe       =          ''        #  Spectral reference frame in which to interpret 'start' and 'width'
     veltype        =     'radio'        #  Velocity type (radio, z, ratio, beta, gamma, optical)
     restfreq       =          []        #  List of rest frequencies
     interpolation  =    'linear'        #  Spectral interpolation (nearest,linear,cubic)
     chanchunks     =           1        #  Number of channel chunks
     perchanweightdensity =       True  #  whether to calculate weight density per channel in Briggs style weighting or not
gridder             =  'standard'        #  Gridding options (standard, wproject, widefield, mosaic, awproject)
     vptable        =          ''        #  Name of Voltage Pattern table
     pblimit        =         0.2        #  PB gain level at which to cut off normalizations

deconvolver         =    'hogbom'        #  Minor cycle algorithm (hogbom,clark,multiscale,mtmfs,mem,clarkstokes)
restoration         =       True         #  Do restoration steps (or not)
     restoringbeam  =         []         #  Restoring beam shape to use. Default is the PSF main lobe
     pbcor          =      False         #  Apply PB correction on the output restored image

outlierfile         =          ''        #  Name of outlier-field image definitions
weighting           =   'natural'        #  Weighting scheme (natural,uniform,briggs, briggsabs[experimental])
     uvtaper        =         []         #  uv-taper on outer baselines in uv-plane

niter               =          0         #  Maximum number of iterations
usemask             =      'user'        #  Type of mask(s) for deconvolution:  user, pb, or auto-multithresh
     mask           =          ''        #  Mask (a list of image name(s) or region file(s) or region string(s) )
     pbmask         =         0.0        #  primary beam mask

fastnoise           =       True         #  True: use the faster (old) noise calculation. False: use the new improved noise calculations
restart             =       True         #  True : Re-use existing images. False : Increment imagename
savemodel           =      'none'        #  Options to save model visibilities (none, virtual, modelcolumn)
calcres             =       True         #  Calculate initial residual image
calcpsf             =       True         #  Calculate PSF
parallel            =      False         #  Run major cycles in parallel
```

specmode='cube'

tclean

nchan

nchan: Number of channels in the output image

For default (=-1), the number of channels will be automatically determined
based on data selected by 'spw' with 'start' and 'width'.
It is often easiest to leave nchan at the default value.

example: nchan=100

tclean

start

start: First channel (e.g. start=3,start='1.1GHz',start='15343km/s')
of output cube images specified by data channel number (integer),
velocity (string with a unit), or frequency (string with a unit).

Default:''; The first channel is automatically determined based on
the 'spw' channel selection and 'width'.
When the channel number is used along with the channel selection
in 'spw' (e.g. spw='0:6~100'),
'start' channel number is RELATIVE (zero-based) to the selected
channels in 'spw'. So for the above example,
start=1 means that the first image channel is the second selected
data channel, which is channel 7.
For specmode='cube', when velocity or frequency is used it is
interpreted with the frame defined in outframe. [The parameters of
the desired output cube can be estimated by using the 'transform'
functionality of 'plotms']

examples: start='5.0km/s'; 1st channel, 5.0km/s in outframe
start='22.3GHz'; 1st channel, 22.3GHz in outframe

tclean

width

width: Channel width (e.g. width=2,width='0.1MHz',width='10km/s') of output cube images specified by data channel number (integer), velocity (string with a unit), or
frequency (string with a unit).

Default:''; data channel width
The sign of width defines the direction of the
channels to be incremented.
For width specified in velocity or frequency with '-' in front
gives image channels in decreasing velocity or frequency, respectively.
For specmode='cube', when velocity or frequency is used it is
interpreted with the reference frame defined in outframe.

examples: width='2.0km/s'; results in channels with increasing velocity
width='-2.0km/s'; results in channels with decreasing velocity
width='40kHz'; results in channels with increasing frequency
width=-2; results in channels averaged of 2 data channels
incremented from high to low channel numbers

tclean

outframe

outframe: Spectral reference frame in which to interpret 'start' and 'width'
Options: '','LSRK','LSRD','BARY','GEO','TOPO','GALACTO','LGROUP','CMB'

example: outframe='bary' for Barycentric frame

REST -- Rest frequency
LSRD -- Local Standard of Rest (J2000)
-- as the dynamical definition (IAU, [9,12,7] km/s in galactic coordinates)
LSRK -- LSR as a kinematical (radio) definition
-- 20.0 km/s in direction ra,dec = [270,+30] deg (B1900.0)
BARY -- Barycentric (J2000)
GEO --- Geocentric
TOPO -- Topocentric
GALACTO -- Galacto centric (with rotation of 220 km/s in direction l,b = [90,0] deg.
LGROUP -- Local group velocity -- 308km/s towards l,b = [105,-7] deg (F. Ghigo)
CMB -- CMB velocity -- 369.5km/s towards l,b = [264.4, 48.4] deg (F. Ghigo)

DEFAULT = LSRK

For width specified in velocity or frequency with '-' in front gives image channels in decreasing velocity or frequency, respectively.
For specmode='cube', when velocity or frequency is used it is interpreted with the reference frame defined in outframe.

examples: width='2.0km/s'; results in channels with increasing velocity
width='-2.0km/s'; results in channels with decreasing velocity
width='40kHz'; results in channels with increasing frequency
width=-2; results in channels averaged of 2 data channels incremented from high to low channel numbers

tclean

restfreq

restfreq: List of rest frequencies or a rest frequency in a string.
Specify rest frequency to use for output image.

* Currently it uses the first rest frequency in the list for translation of
velocities. The list will be stored in the output images.

Default: []; look for the rest frequency stored in the MS, if not available,
use center frequency of the selected channels

examples: restfreq=['1.42GHz']
restfreq='1.42GHz'

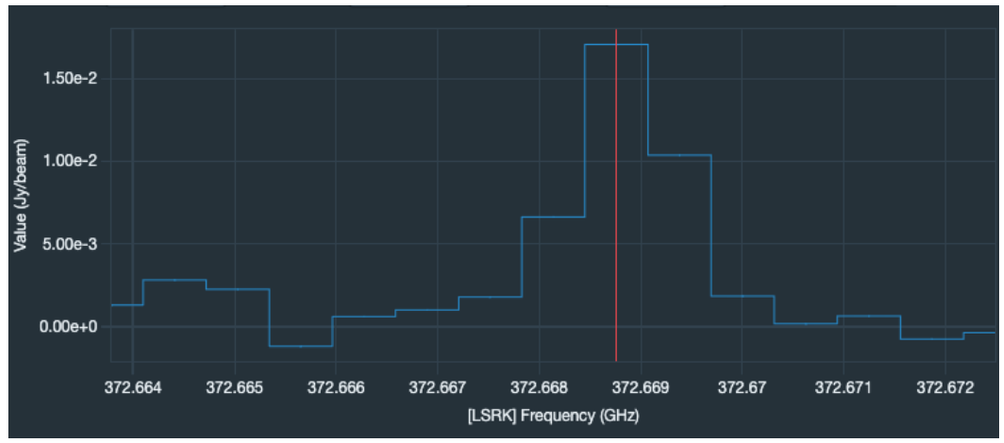
go

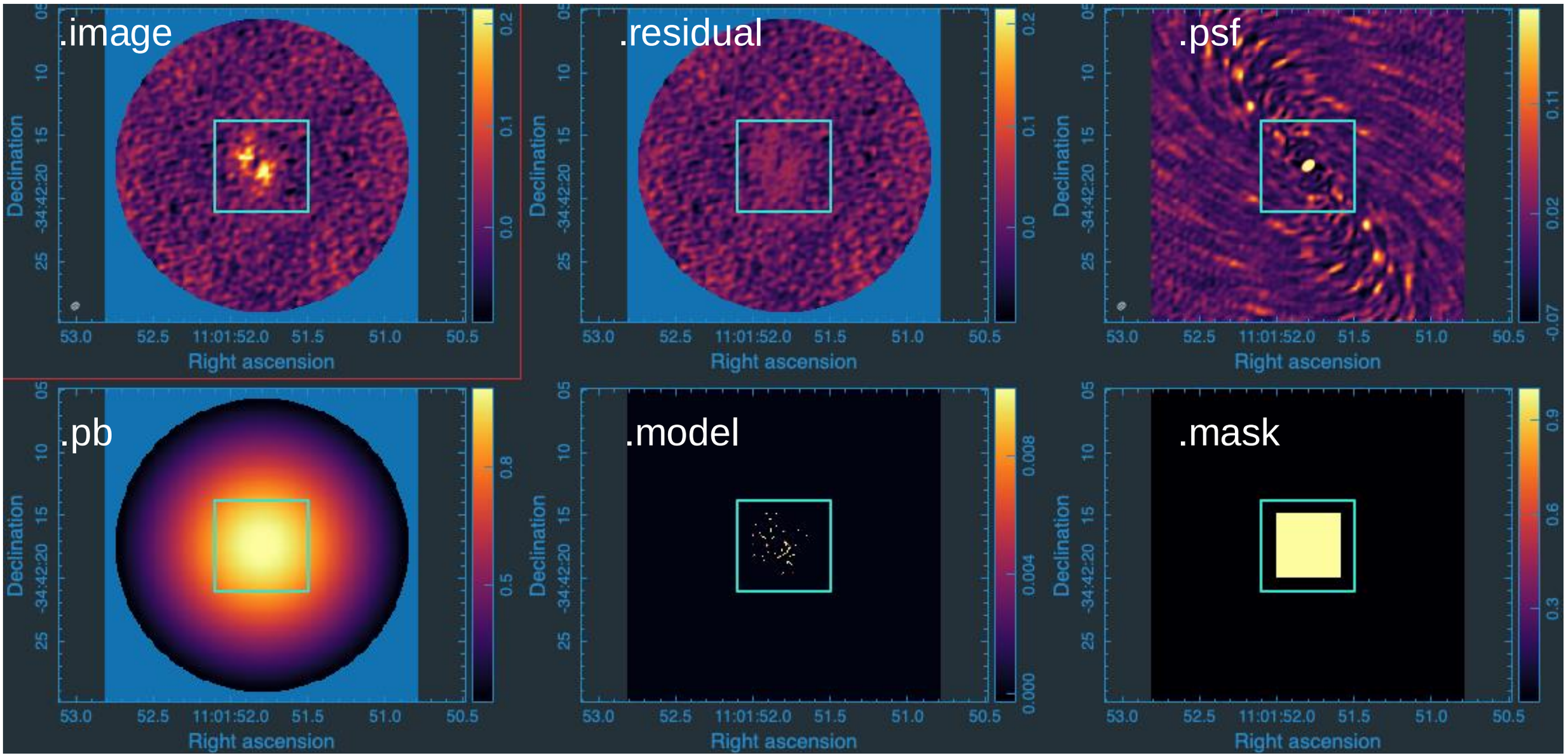
pay attention to the logger

```
[twhya_n2hp:C3] iters=0->0 [0], model=-0.182575->-0.182575, peakres=0.0497475->0.0497475, Reached cyclethre
[twhya_n2hp:C4] iters=0->0 [0], model=0.247778->0.247778, peakres=0.049796->0.049796, Reached cyclethreshol
[twhya_n2hp:C5] iters=0->1 [1], model=1.19769->1.20272, peakres=0.0503464->0.0497634, Reached cyclethreshol
[twhya_n2hp:C6] iters=1->4 [3], model=2.07364->2.07877, peakres=0.0512805->0.0498586, Reached cyclethreshol
[twhya_n2hp:C7] iters=4->7 [3], model=0.855563->0.850522, peakres=0.0511172->0.0496244, Reached cyclethresh
[twhya_n2hp:C8] iters=7->7 [0], model=0.254666->0.254666, peakres=0.0497943->0.0497943, Reached cyclethresh
[twhya_n2hp:C9] iters=7->8 [1], model=0.0620161->0.05696, peakres=0.0505608->0.04992, Reached cyclethreshol
[twhya_n2hp:C10] iters=8->8 [0], model=0.124407->0.124407, peakres=0.0493173->0.0493173, Reached cyclethres
[twhya_n2hp:C11] iters=8->8 [0], model=-0.177909->-0.177909, peakres=0.0491975->0.0491975, Reached cyclethr
[twhya_n2hp:C12] iters=8->8 [0], model=0.199665->0.199665, peakres=0.0493527->0.0493527, Reached cyclethres
[twhya_n2hp:C13] iters=8->8 [0], model=0.254298->0.254298, peakres=0.0498198->0.0498198, Reached cyclethres
[twhya_n2hp:C14] iters=8->8 [0], model=-0.0202667->-0.0202667, peakres=0.049854->0.049854, Reached cyclethr
[twhya_n2hp] Total model flux (over all planes) : 4.72494     Peak Residual (over all planes) : 0.0499917 in
Completed 2285 iterations.
---------------------------------------------------------- Run Major Cycle 7 ----------------------------
Required memory: 0.04526 GB. Available mem.: 699 GB (rc, mem. fraction: 80%, memory: -) => Subcubes: 1. Pro
MS : twhya_selfcal.ms.contsub | Selecting on fields : 5 | Selecting on spw :0 | [Opened in readonly mode]
  NRows selected : 44772
Set imaging weights : Briggs weighting: sidelobes will be suppressed over full image
Doing spectral cube Briggs weighting formula --  bwtaper
Tuning frequency data selection to match image spectral coordinates
MS : twhya_selfcal.ms.contsub | Selecting on fields : 5 | Selecting on spw :0 | [Opened in readonly mode]
  NRows selected : 44772
---------------------------------------------------------- Run Major Cycle 7 ----------------------------
 Absolute Peak residual within mask : 0.0499917, over full image : 0.145023
Reached global stopping criterion : threshold
getSummary call: fullsummary=False
[twhya_n2hp] : Restoring model image.
Getting common beam
Common Beam : 0.614894 arcsec, 0.453196 arcsec, -52.3802 deg
Searching for images with prefix 'twhya_n2hp'... Found these, writing history into them: ['twhya_n2hp.model'
```

Image visualization

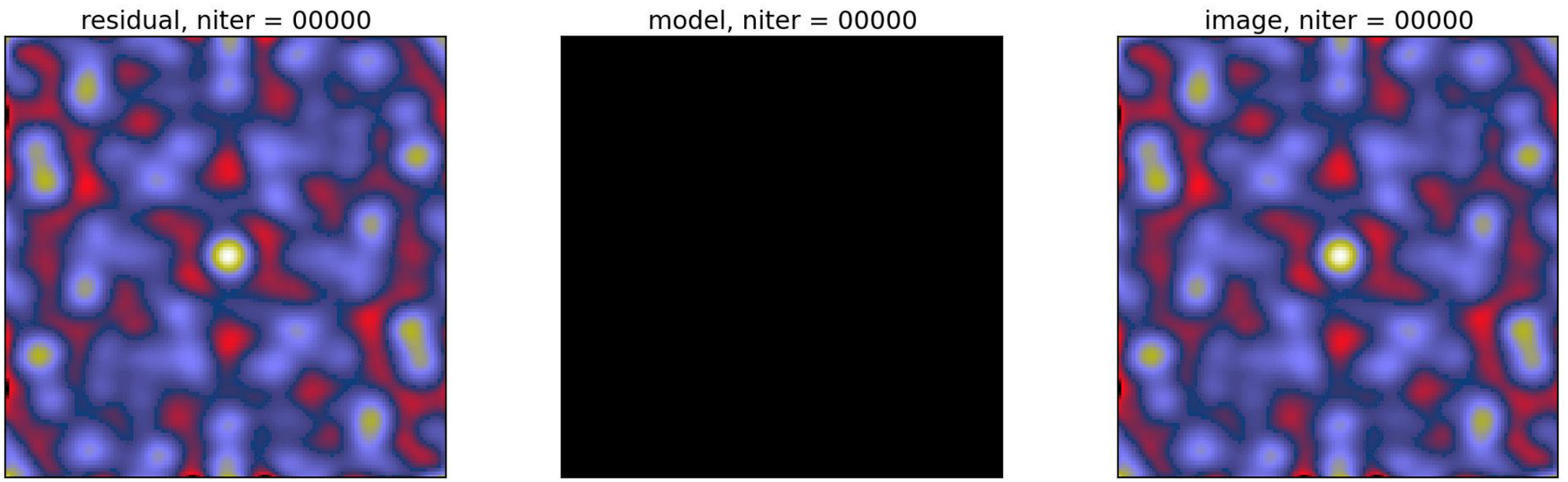
Clean-ed images

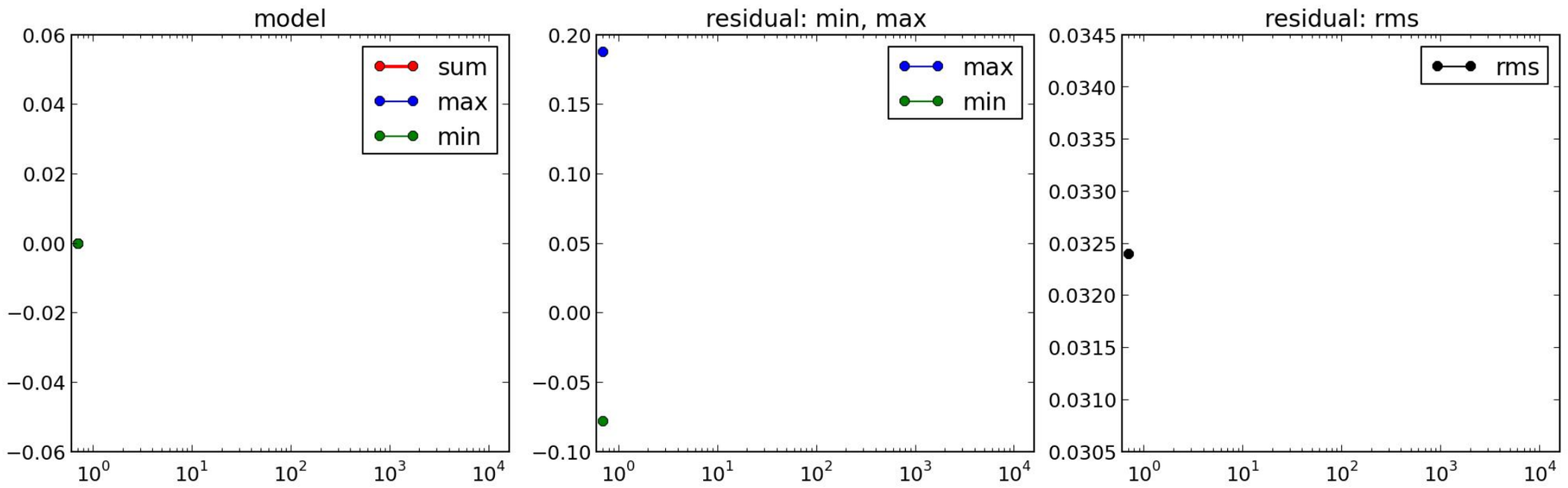




Advanced topics

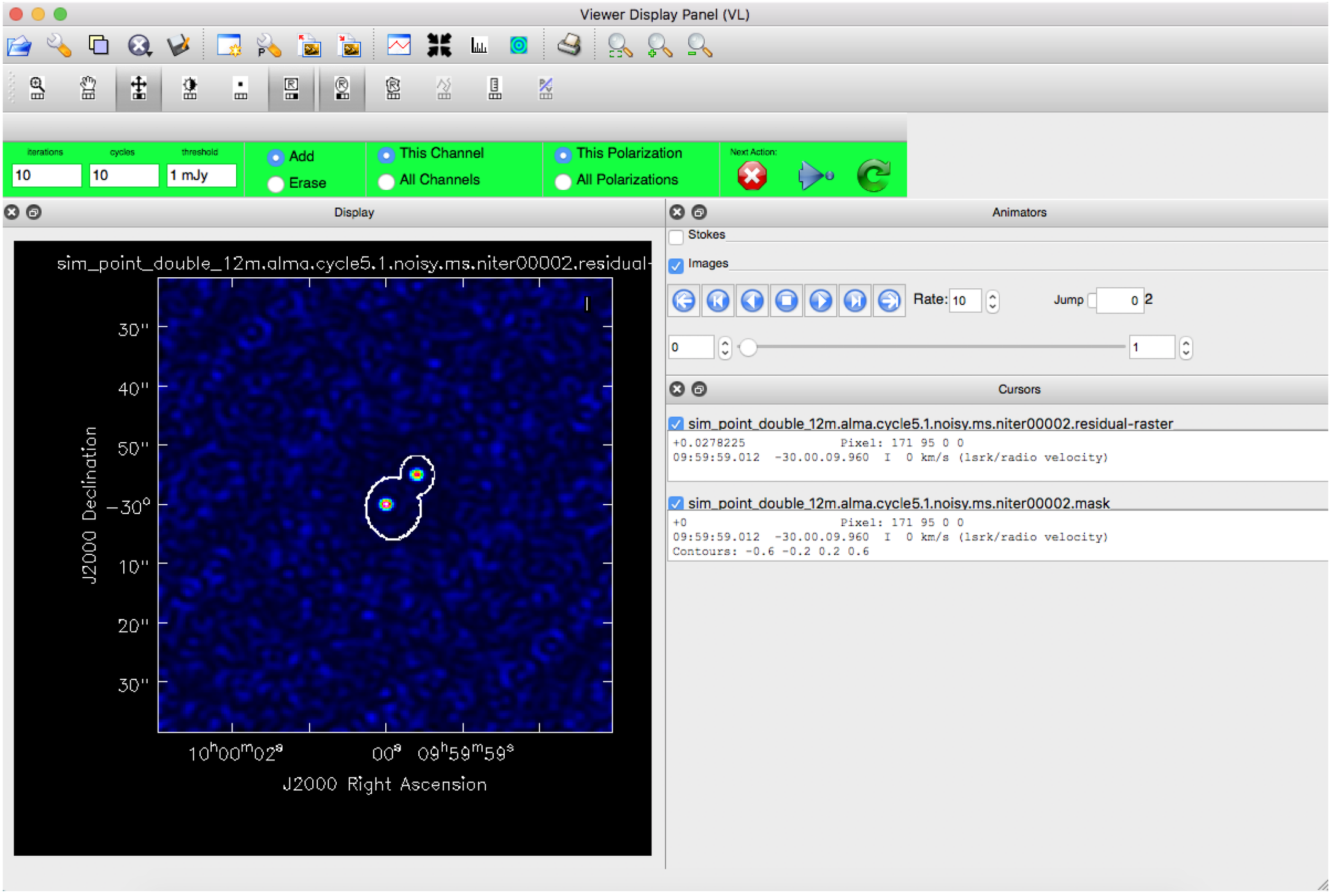
overclean





Advanced topics

interactive clean



Good references

- CASA user manual
  - https://casadocs.readthedocs.io/en/v6.6.1/
- CASA user manual: synthesis imaging
  - https://casadocs.readthedocs.io/en/v6.6.1/notebooks/synthesis_imaging.html
- CLEAN algorithm
  - Högbom (1974); http://adsabs.harvard.edu/abs/1974A%26AS...15..417H
- “Bible”
  - Interferometry and Synthesis in Radio Astronomy (https://link.springer.com/book/10.1007/978-3-319-44431-4)
  - Synthesis Imaging in Radio Astronomy II (http://www.aspbooks.org/a/volumes/table_of_contents/?book_id=292)

Try…

- Different weighting schemes
- Different deconvolver
- interactive clean