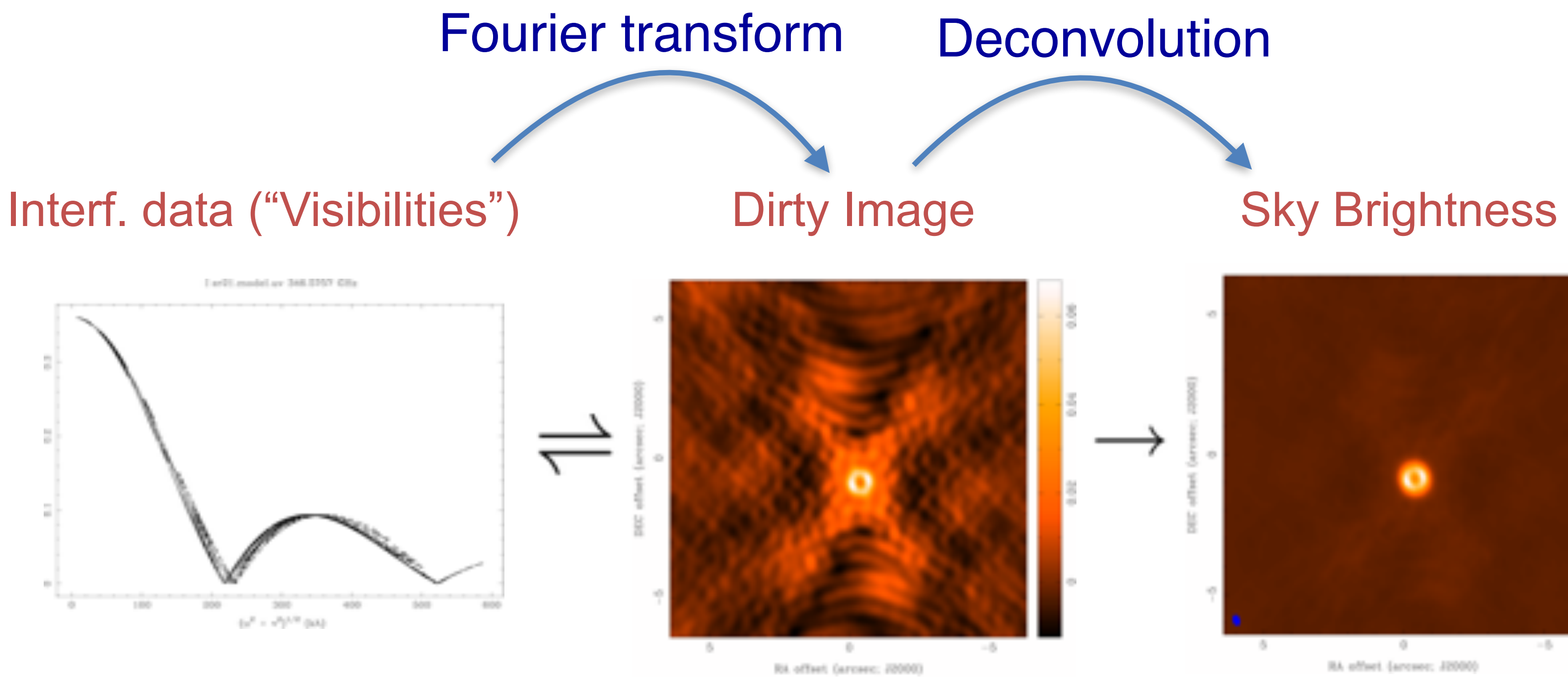
Introduction to Imaging in CASA

Luca Ricci (Rice University)

Atacama Large Millimeter/submillimeter Array
Expanded Very Large Array

Talk Outline

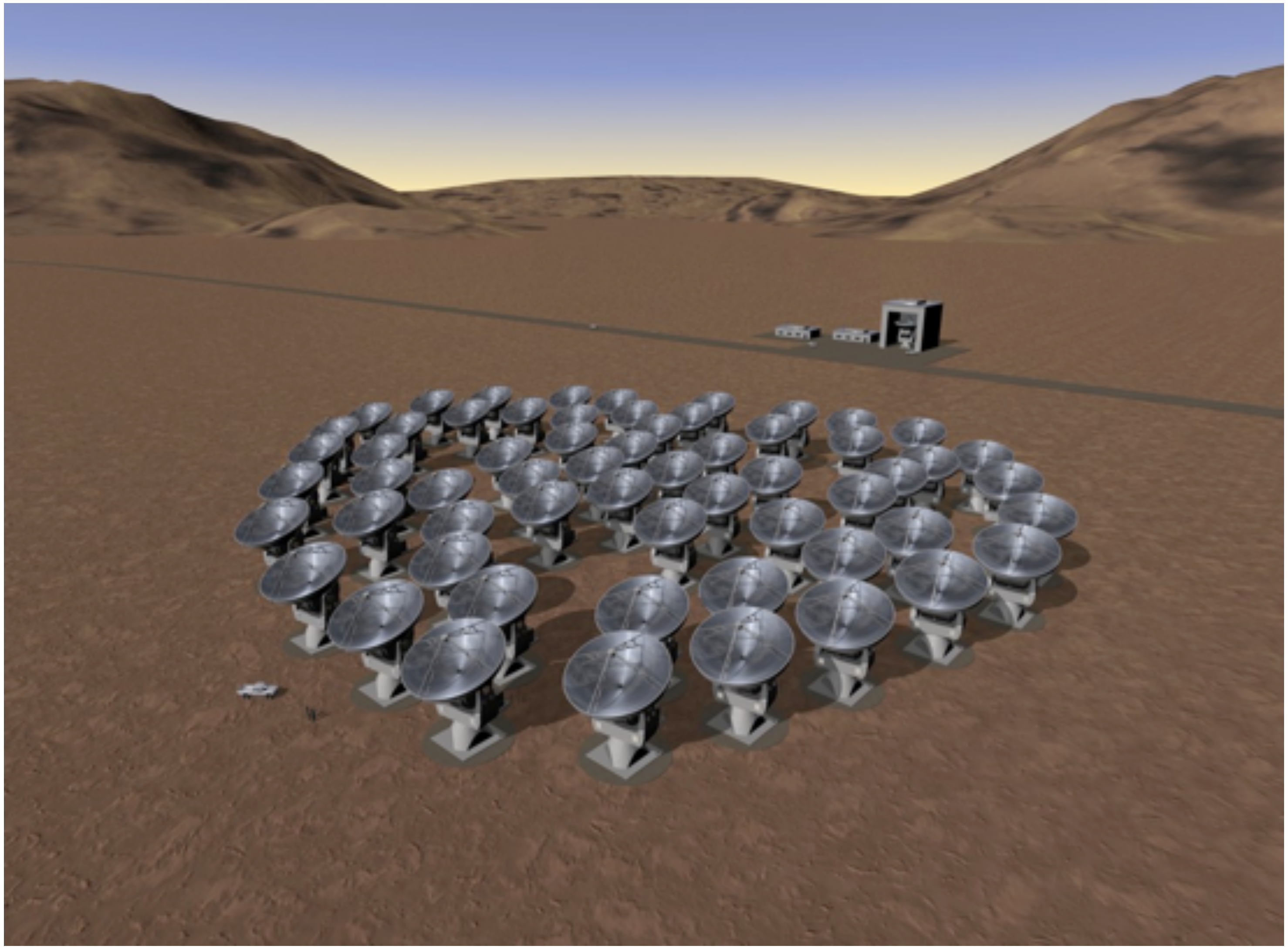


"Step 0": Calibration of Interferometric Visibilities

Interferometry Basics

Single dish: diameter is responsible for sensitivity, field of view, resolution

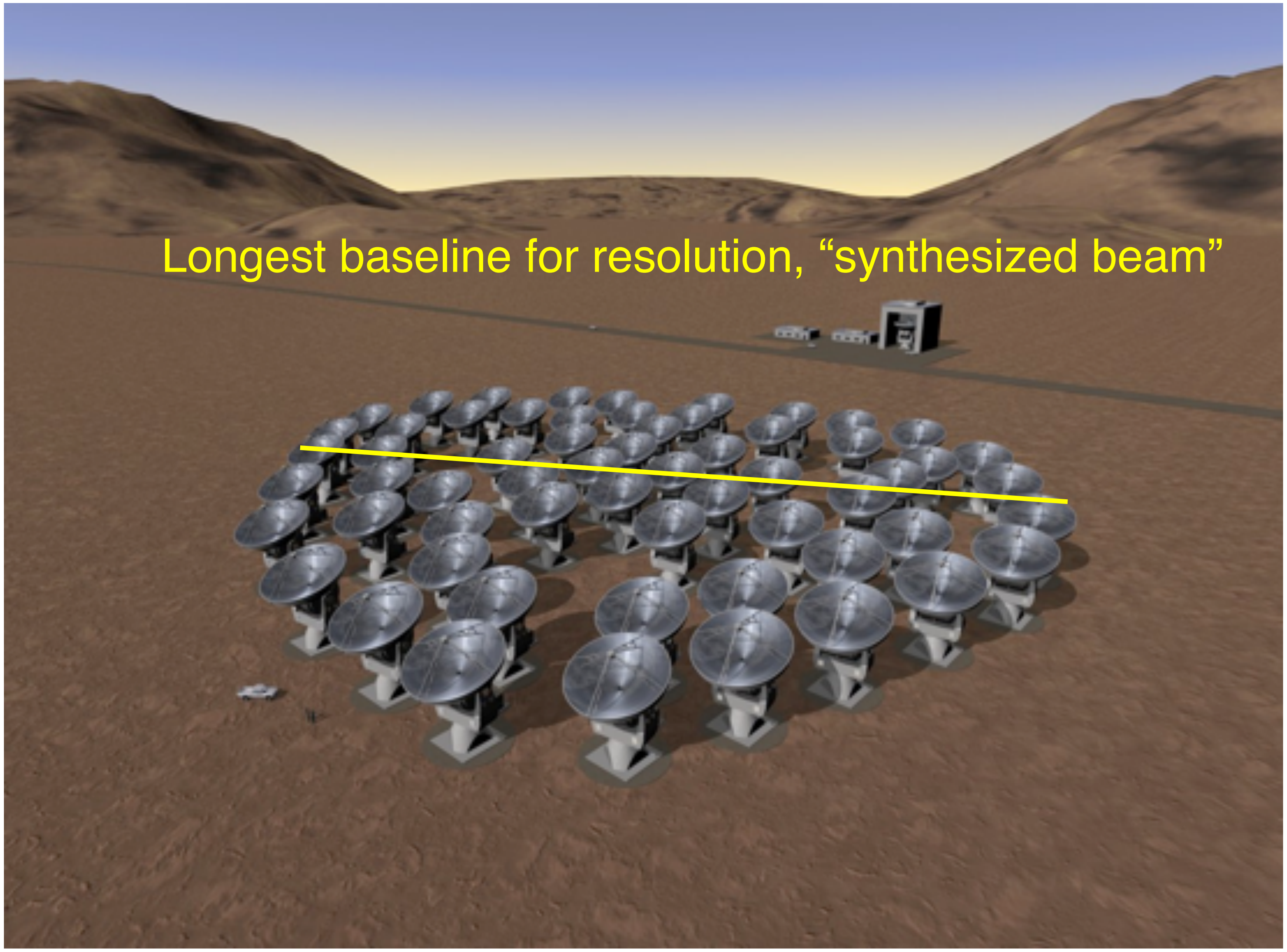
Interferometer: multiple contributing design parameters



Interferometry Basics

Single dish: diameter is responsible for sensitivity, field of view, resolution

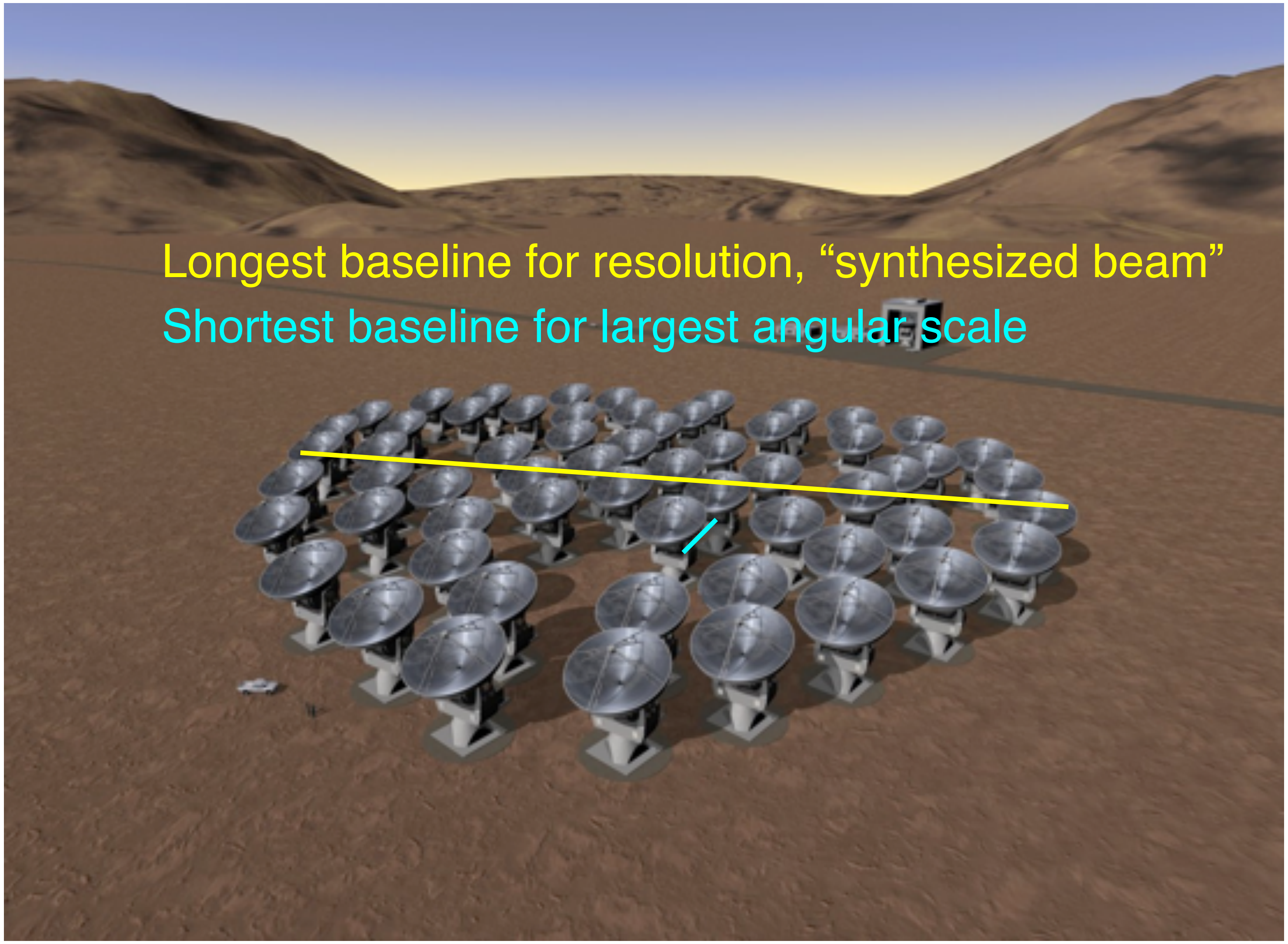
Interferometer: multiple contributing design parameters



Interferometry Basics

Single dish: diameter is responsible for sensitivity, field of view, resolution

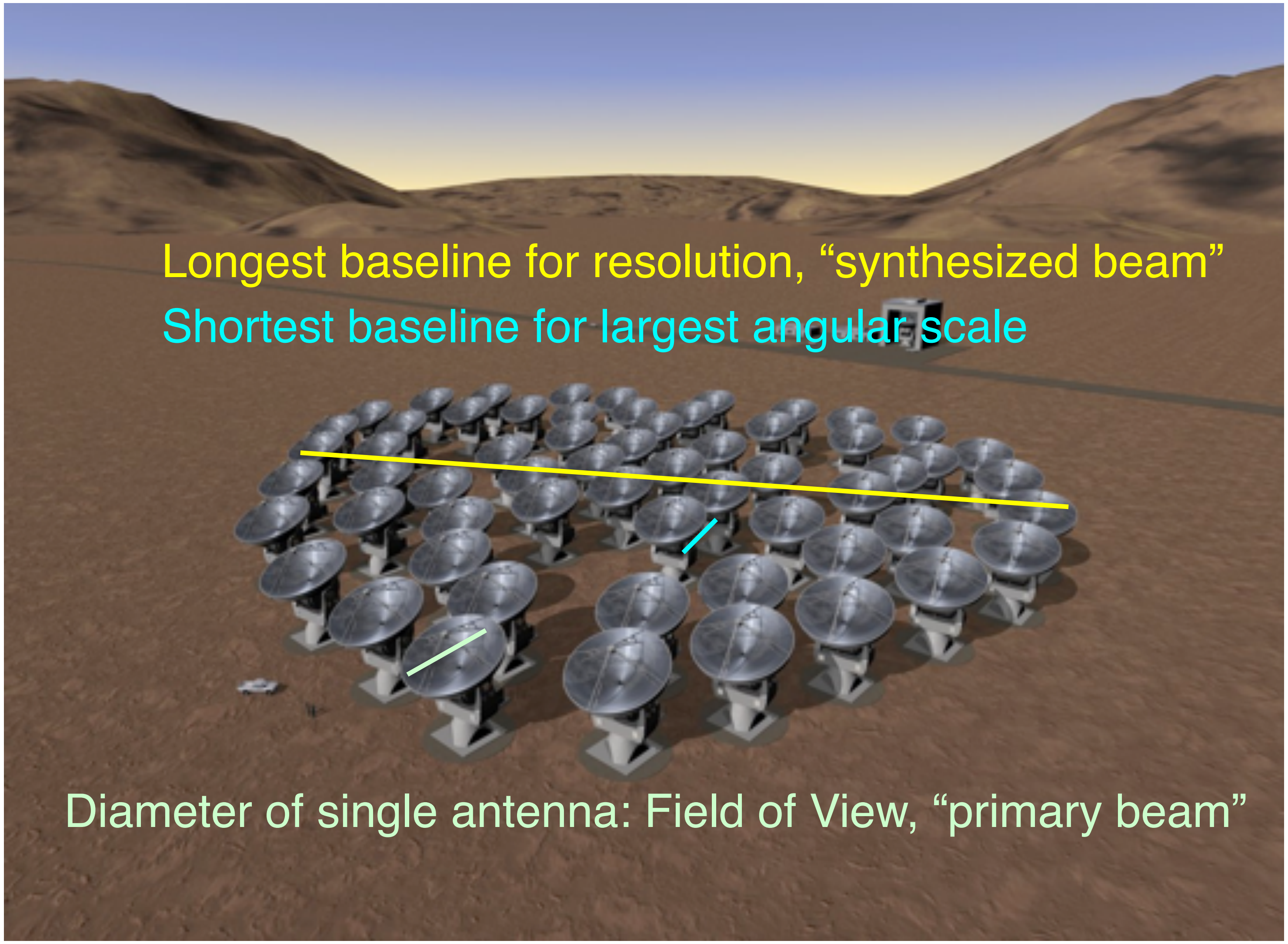
Interferometer: multiple contributing design parameters


Longest baseline for resolution, “synthesized beam”
Shortest baseline for largest angular scale

Interferometry Basics

Single dish: diameter is responsible for sensitivity, field of view, resolution

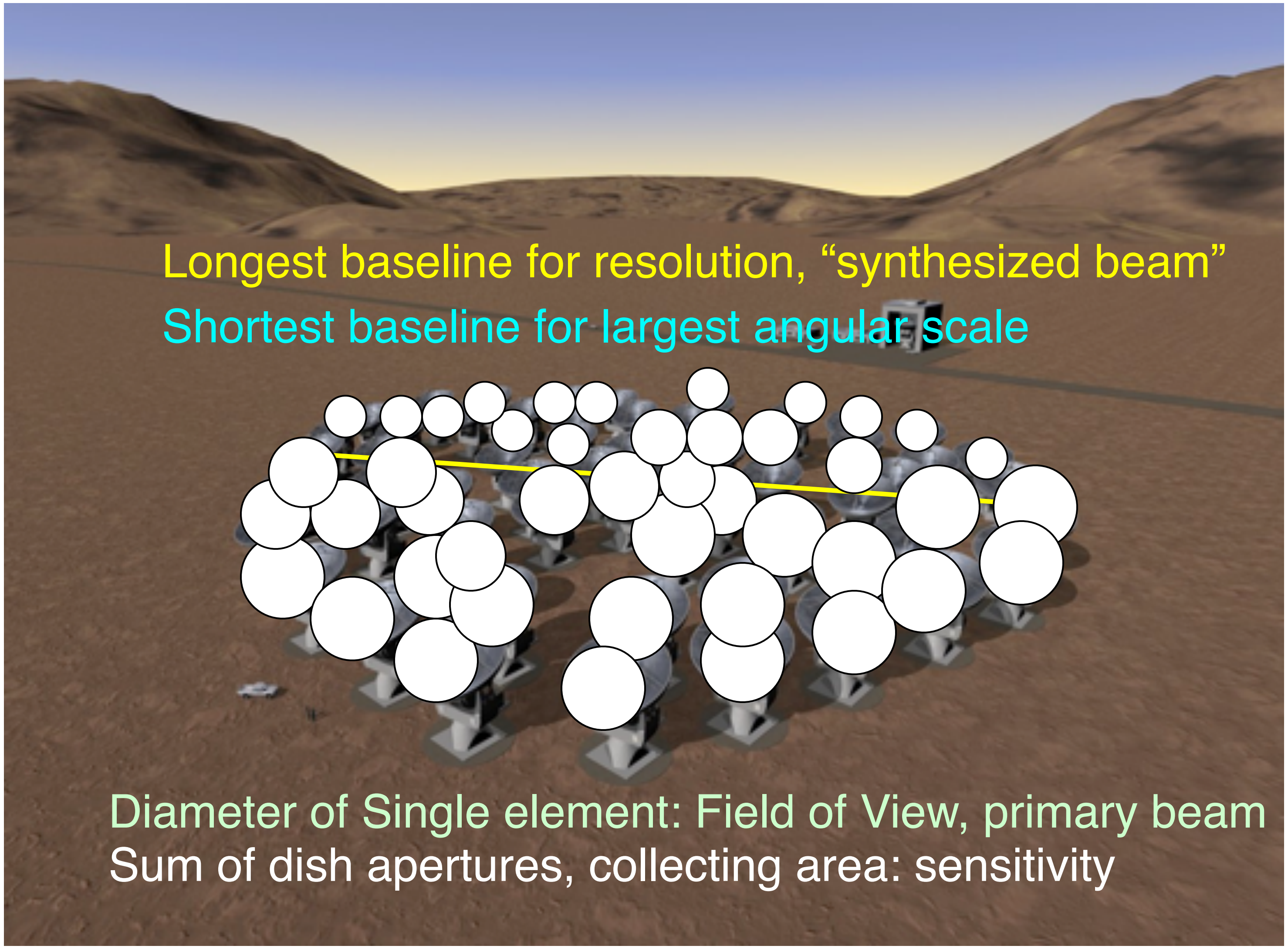
Interferometer: multiple contributing design parameters


Longest baseline for resolution, “synthesized beam”
Shortest baseline for largest angular scale
Diameter of single antenna: Field of View, “primary beam”

Interferometry Basics

Single dish: diameter is responsible for sensitivity, field of view, resolution

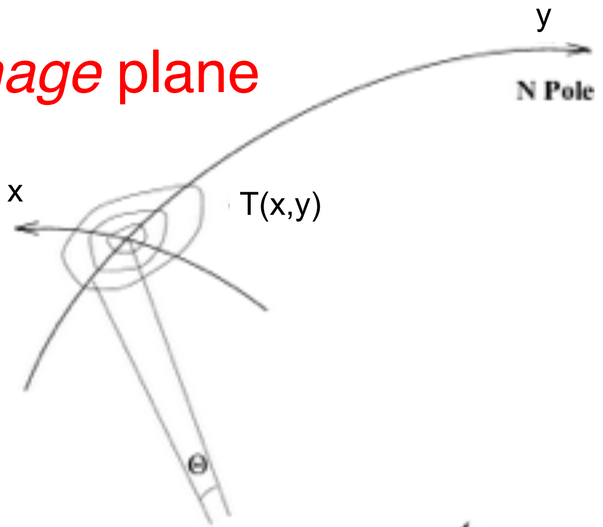
Interferometer: multiple contributing design parameters


Longest baseline for resolution, “synthesized beam”
Shortest baseline for largest angular scale
Diameter of Single element: Field of View, primary beam
Sum of dish apertures, collecting area: sensitivity

From Sky Brightness to Visibility

1. An interferometer measures the complex correlations between each pair of antennas
2. For small fields of view the complex visibility, V(u,v), is the 2D Fourier transform of the brightness of the sky, T(x,y)

image plane



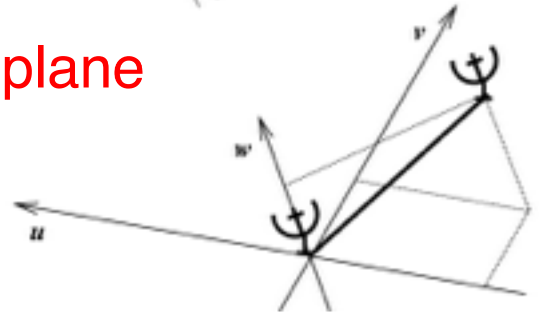
Fourier space/domain

$$V(u,v) = \int\int T(x,y) e^{2\pi i(ux+vy)} dx dy$$

$$T(x,y) = \int\int V(u,v) e^{-2\pi i(ux+vy)} du dv$$

Image space/domain

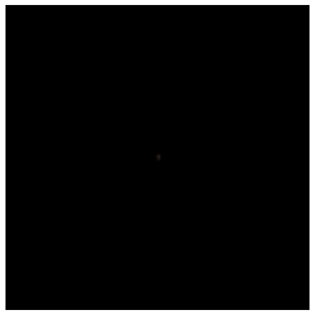
uv plane



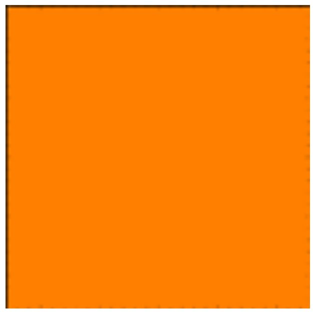
Some 2D Fourier Transform Pairs

T(x,y) Amp{V(u,v)}

δ Function

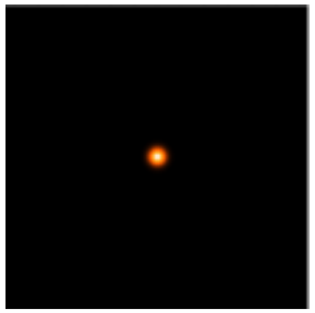


$\rightleftharpoons$

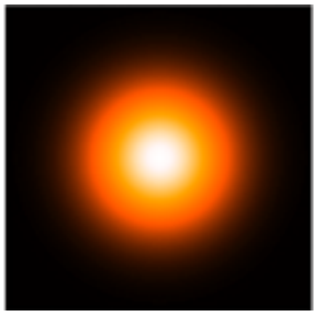


Constant

Gaussian

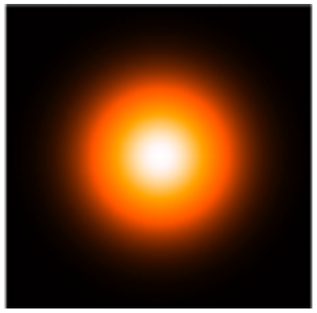


$\rightleftharpoons$

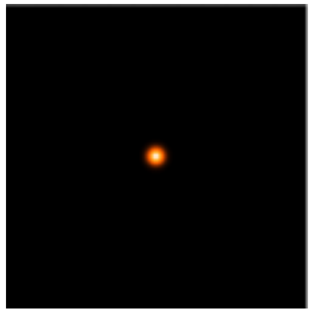


Gaussian

Gaussian



$\rightleftharpoons$



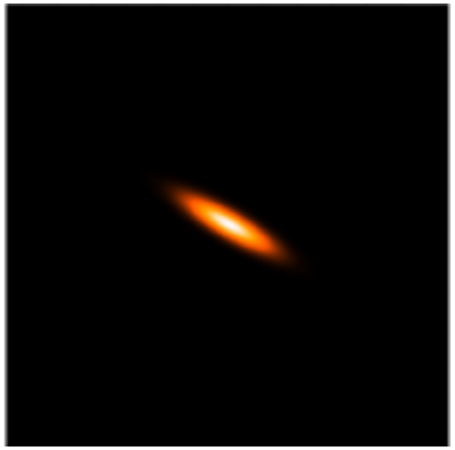
Gaussian

narrow features transform to wide features (and vice-versa)

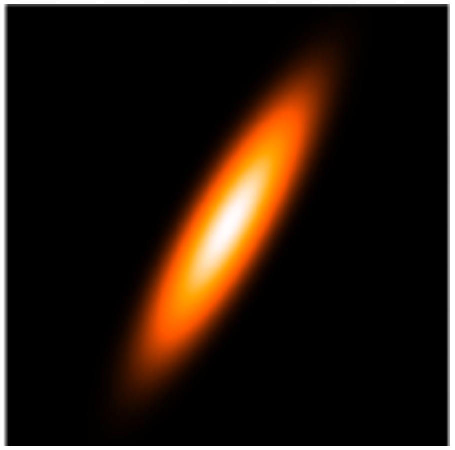
More 2D Fourier Transform Pairs

T(x,y) Amp{V(u,v)}

elliptical Gaussian

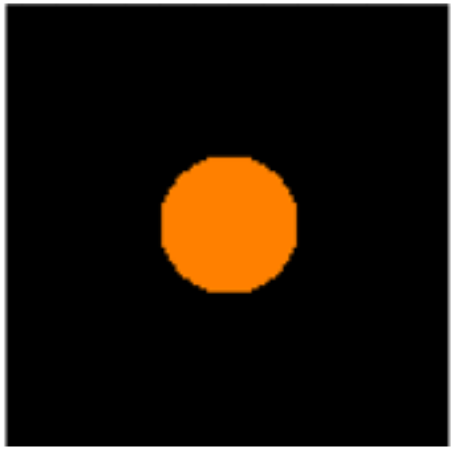


$\rightleftharpoons$

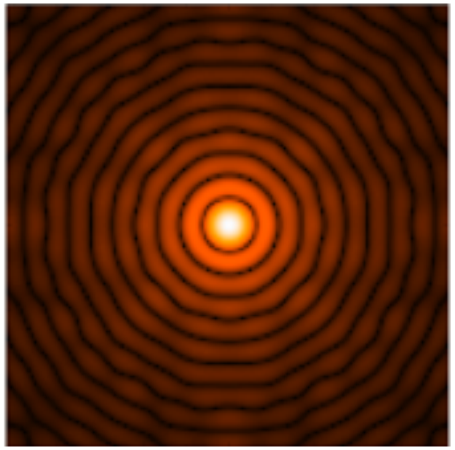


elliptical Gaussian

Disk



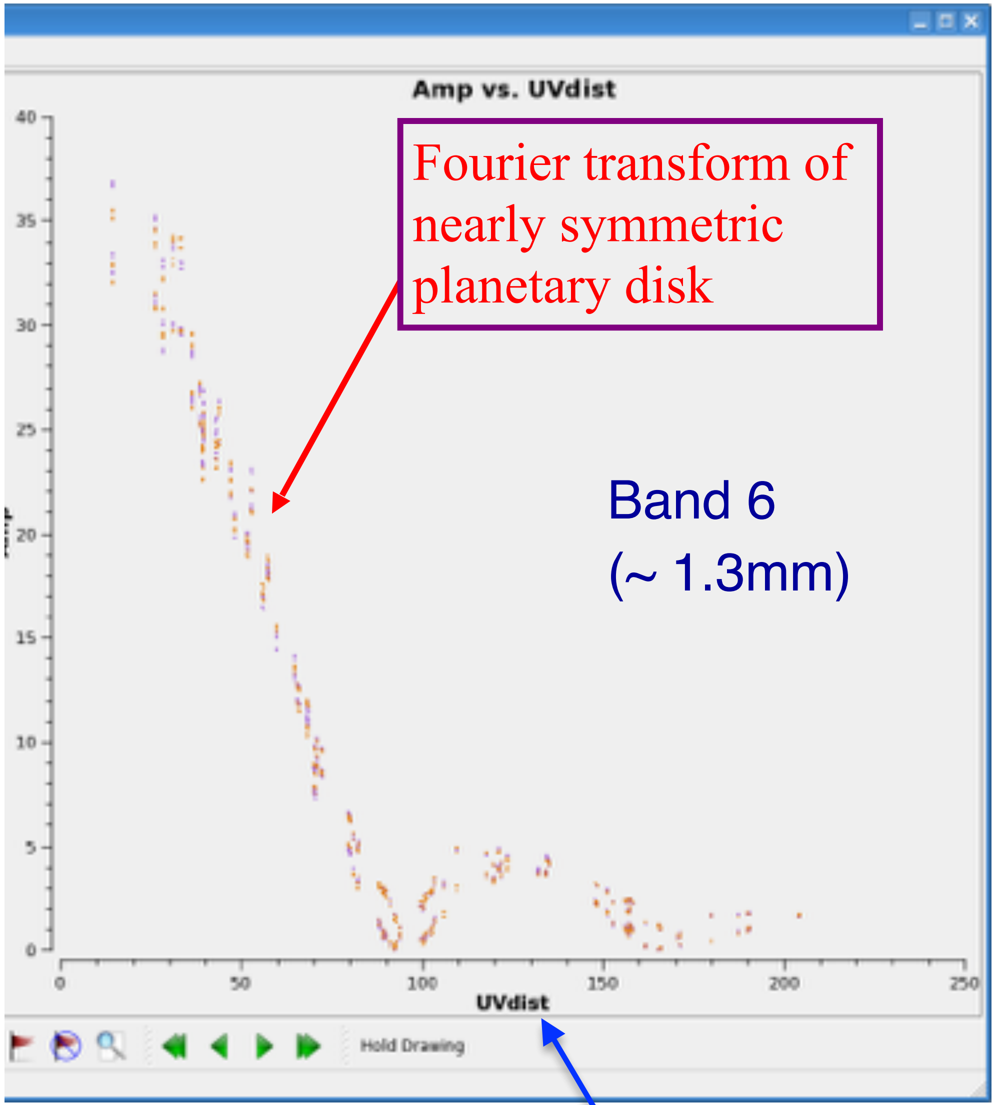
$\rightleftharpoons$



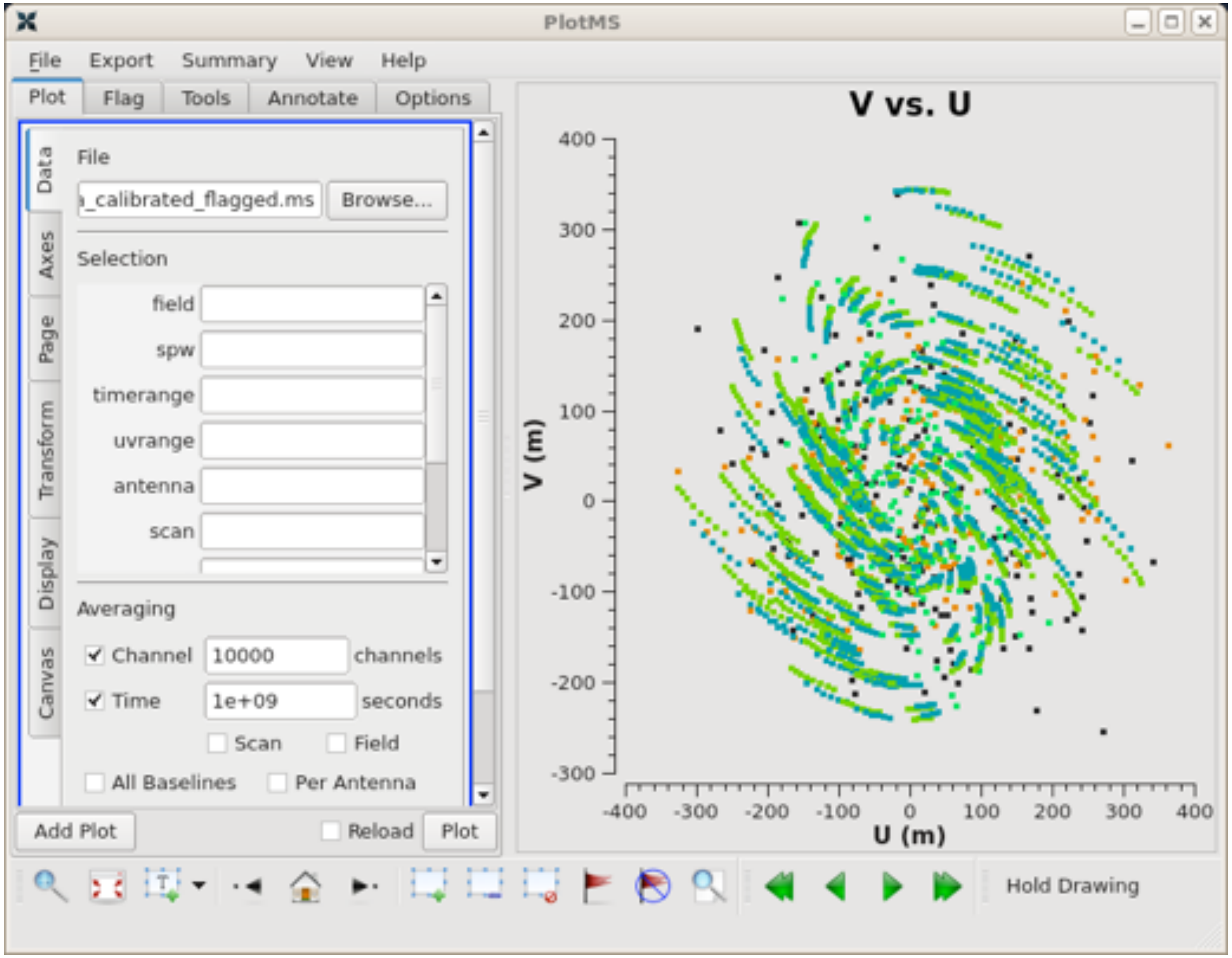
Bessel

sharp edges result in many high spatial frequencies

ALMA observes planetary disk



Plotms: Versatile examination of UV data



File: _calibrated_flagged.ms

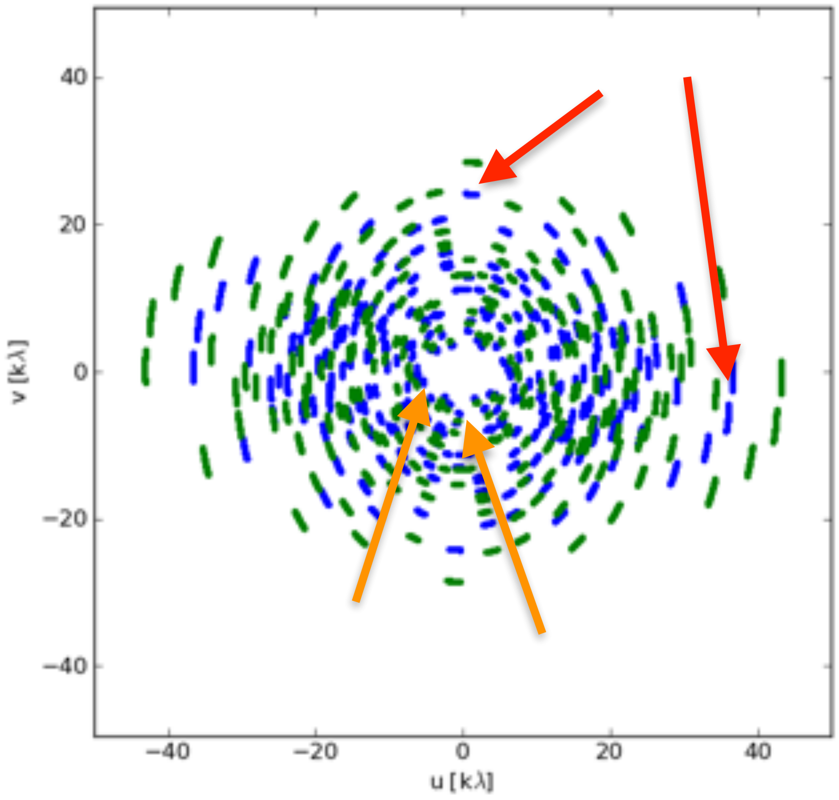
Selection
- field:
- spw:
- timerange:
- uvrange:
- antenna:
- scan:

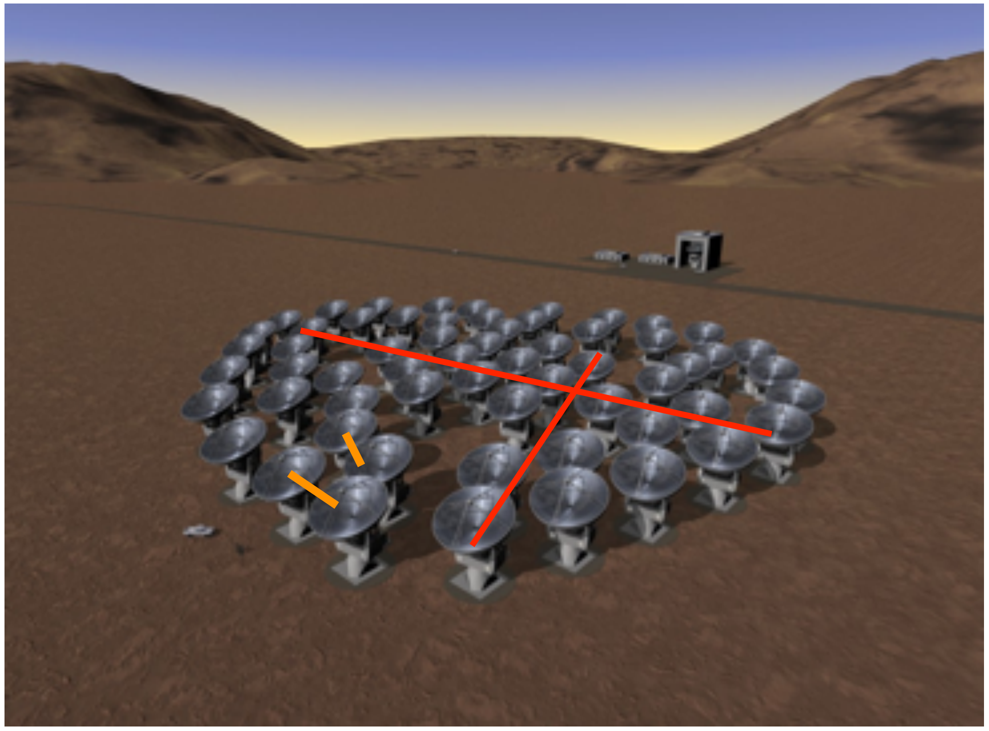
Averaging
- [✓] Channel 10000 channels
- [✓] Time 1e+09 seconds
- [ ] Scan [ ] Field
- [ ] All Baselines [ ] Per Antenna

uv-coverage

Interferometers cannot see the entire Fourier/uv domain

→ **imperfect, "dirty" image**





Short baselines: small uv-distance, low angular frequencies, i.e. extended emission

Long baselines: large uv-distance, high angular frequencies, i.e. small-scale emission

uv-coverage with many "holes" gives dirty beams with pronounced sidelobes

“Dirty Images” from a “Dirty Beam”

$$T(x,y) = \int\int V(u,v) e^{-2\pi i(ux+vy)} du dv$$

- We sample the Fourier domain at discrete points

  $B(u,v) = \sum_k (u_k, v_k)$

- The inverse Fourier Transform is

  $T^D(x,y) = FT^{-1}\{B(u,v) \times V(u,v)\}$

- The convolution theorem tells us

  $T^D(x,y) = b(x,y) \otimes T(x,y)$

- Where the point spread function is

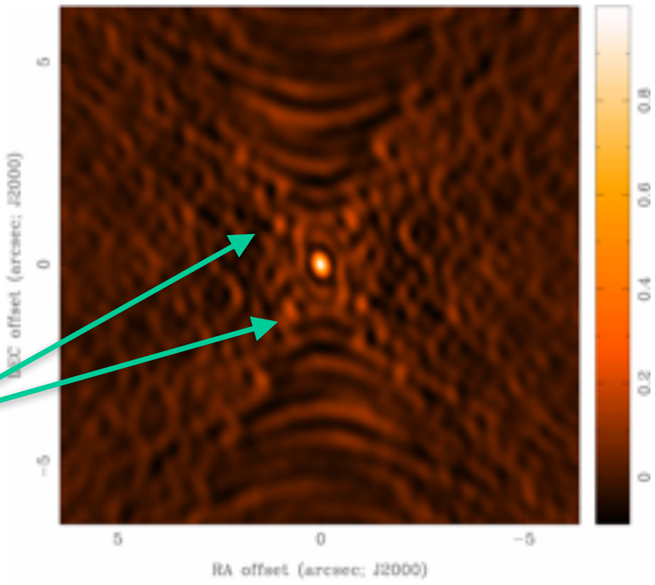
  $b(x,y) = FT^{-1}\{B(u,v)\}$ “dirty beam”

- The “dirty image” is the true image convolved with the “dirty beam”

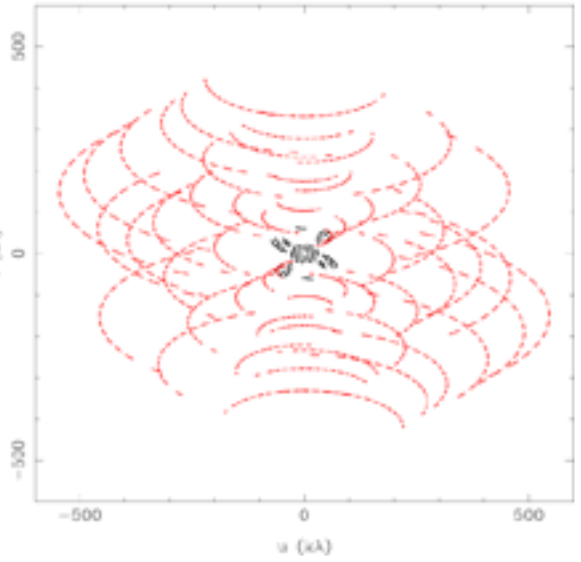
Dirty Beam and Dirty Image

b(x,y)
dirty beam

bad
sidelobes



$\rightleftharpoons$



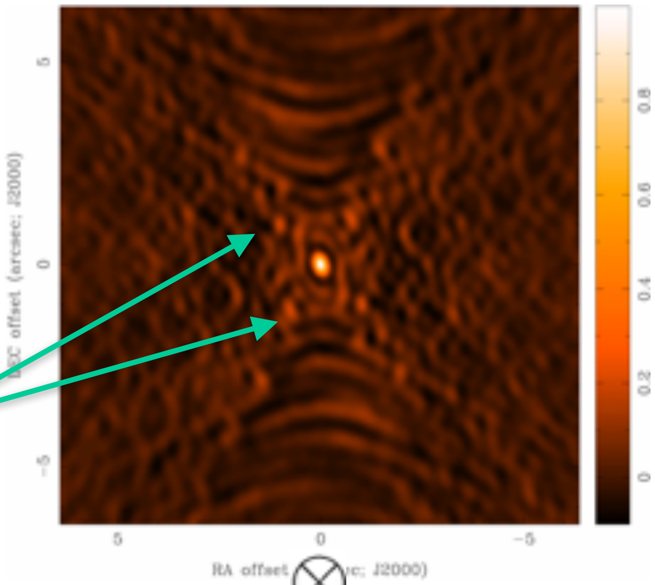
B(u,v)

Dirty Beam and Dirty Image

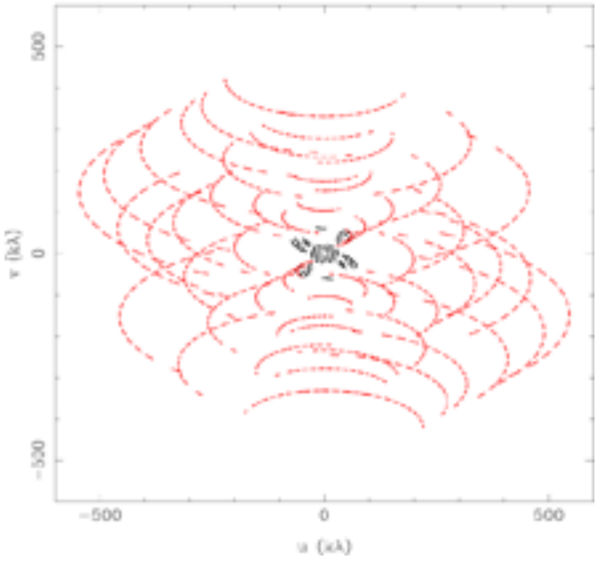
$b(x,y)$

dirty beam

bad sidelobes



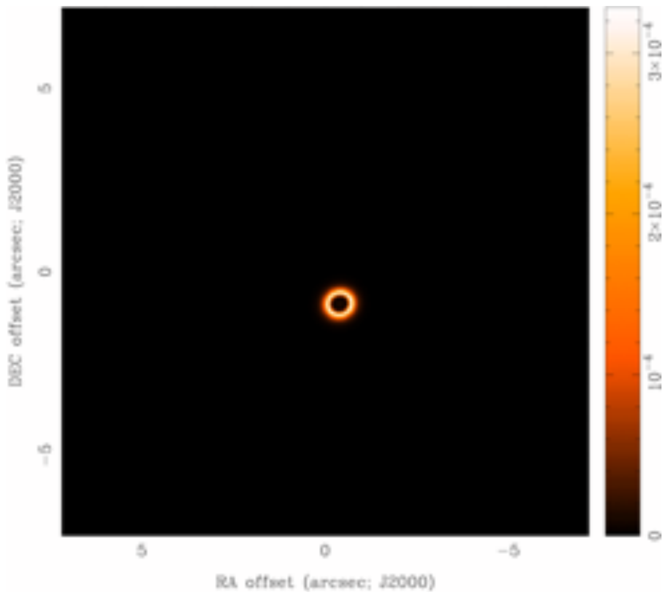
$\rightleftharpoons$



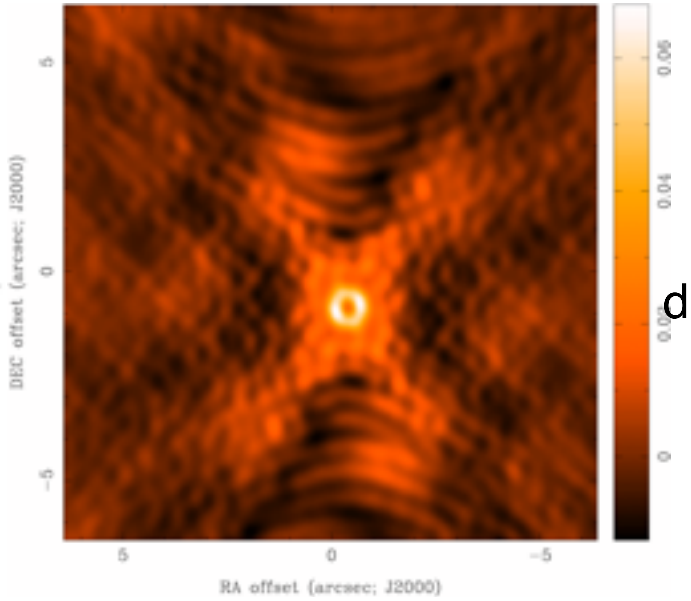
$B(u,v)$

$\otimes$

$T(x,y)$



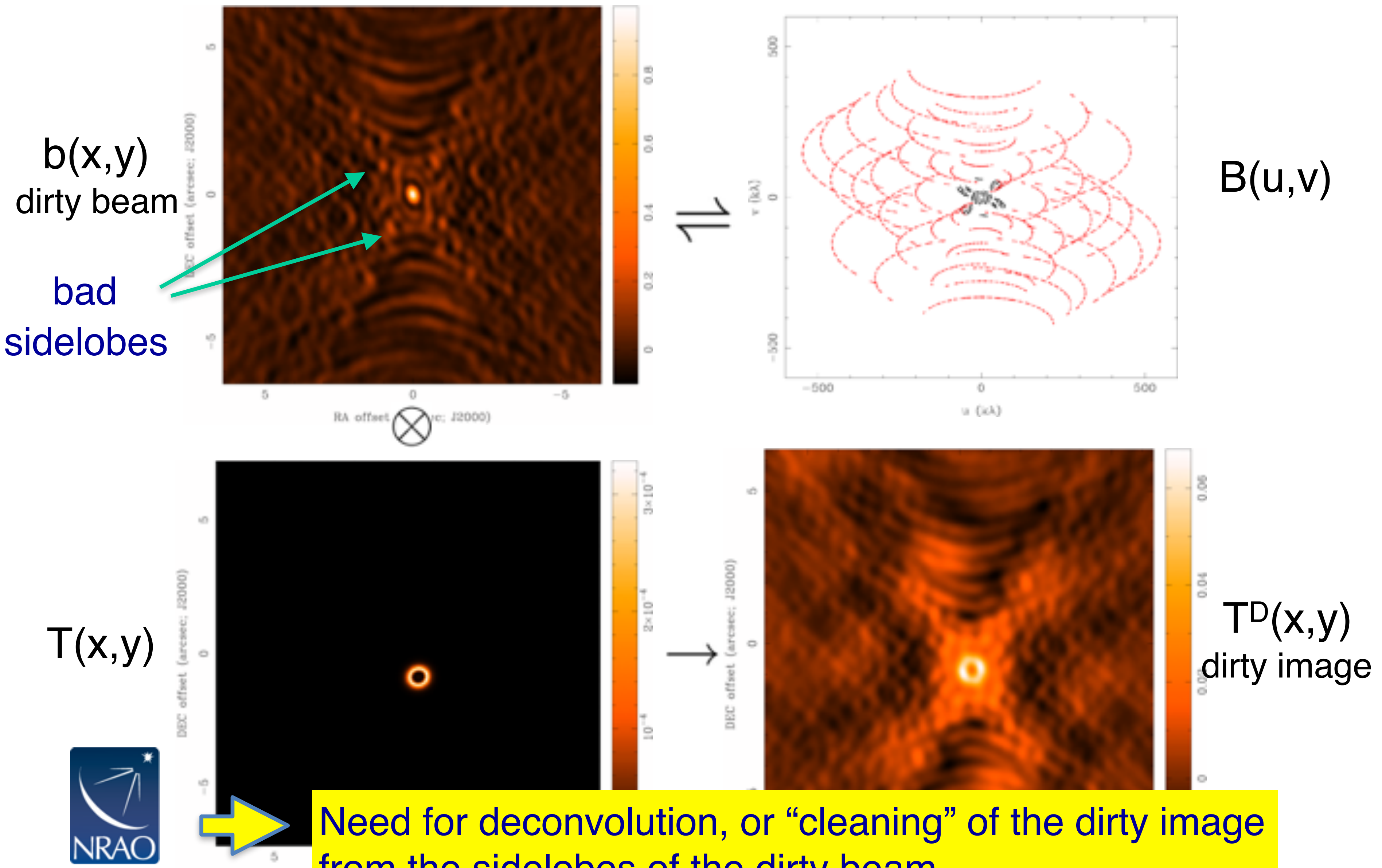
$\rightarrow$



$T^D(x,y)$

dirty image

Dirty Beam and Dirty Image

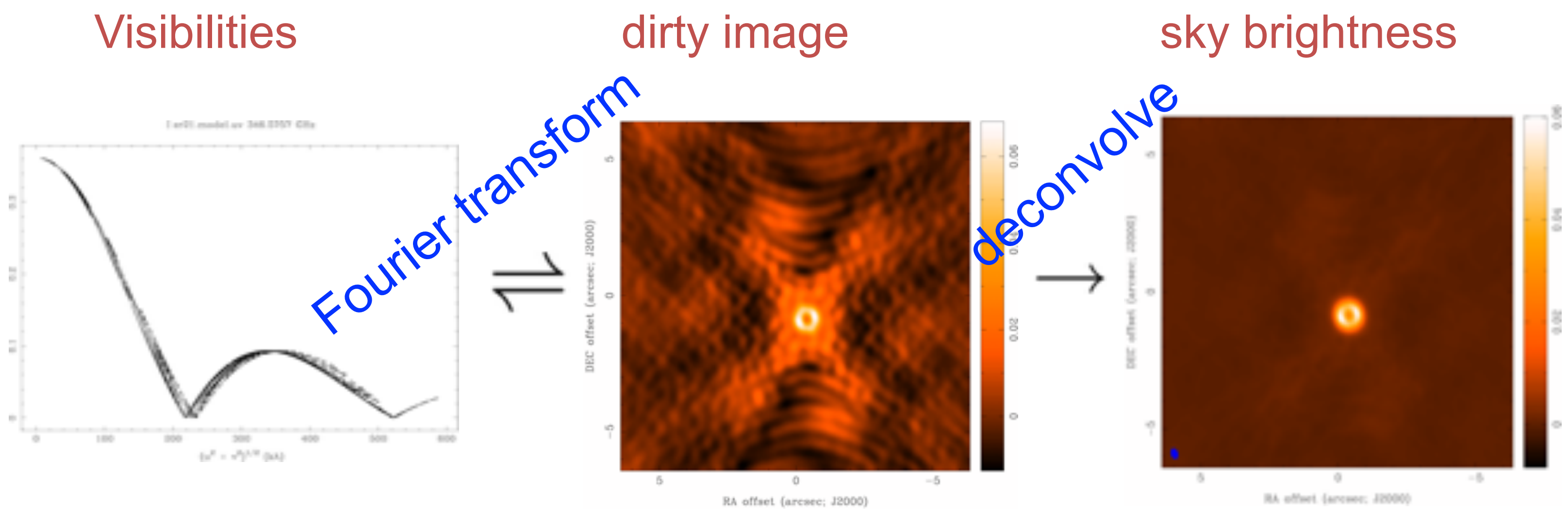


Need for deconvolution, or “cleaning” of the dirty image from the sidelobes of the dirty beam

How to analyze (imperfect) interferometer data

Image plane analysis

- dirty image $T^D(x,y)$ = Fourier transform { $V(u,v)$ x $B(u,v)$ }
- deconvolve $b(x,y)$ from $T^D(x,y)$ to determine (model of) $T(x,y)$

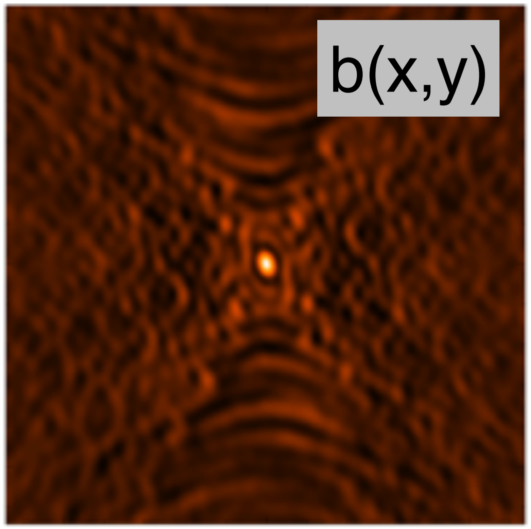


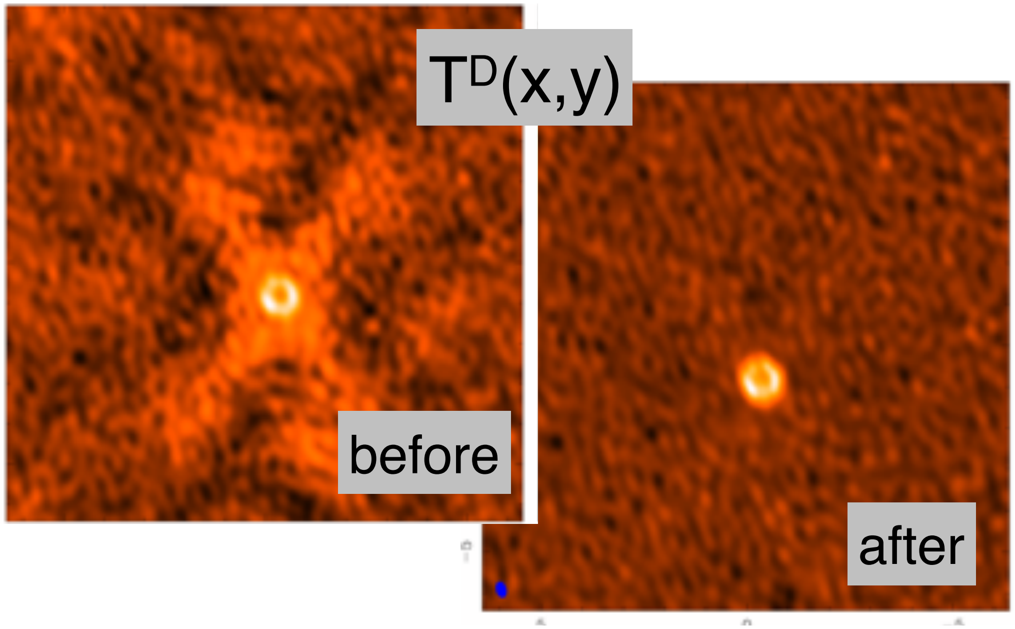
CLEAN Algorithm (deconvolution in CASA)

A. Initialize a *residual* map to the dirty image

1. Start loop
2. Identify brightest pixel in *residual* map as a point source
3. Add this point source to the clean component list
4. Convolve the point source with b(x,y) and subtract a fraction g (the loop gain) of that from *residual* map
5. If stopping criteria not reached, do next iteration

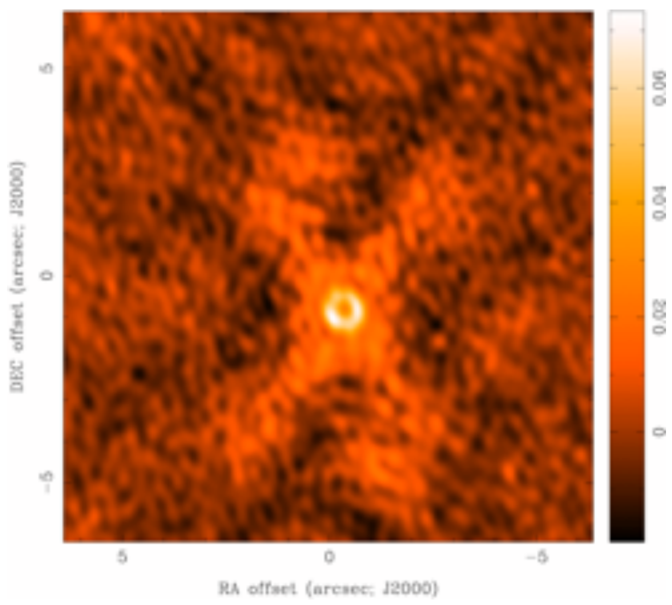
B. Convolve *Clean component* (cc) list by an estimate of the main lobe of the dirty beam (the "Clean beam") and add *residual* map to make the final "restored" image

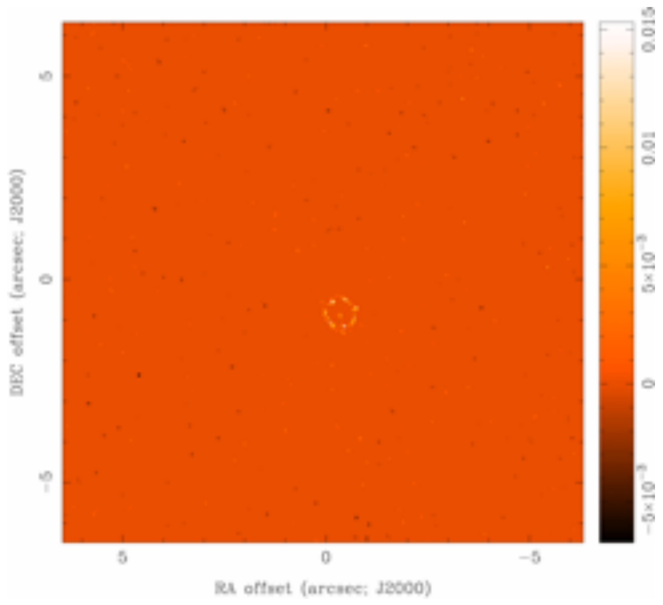




CLEAN

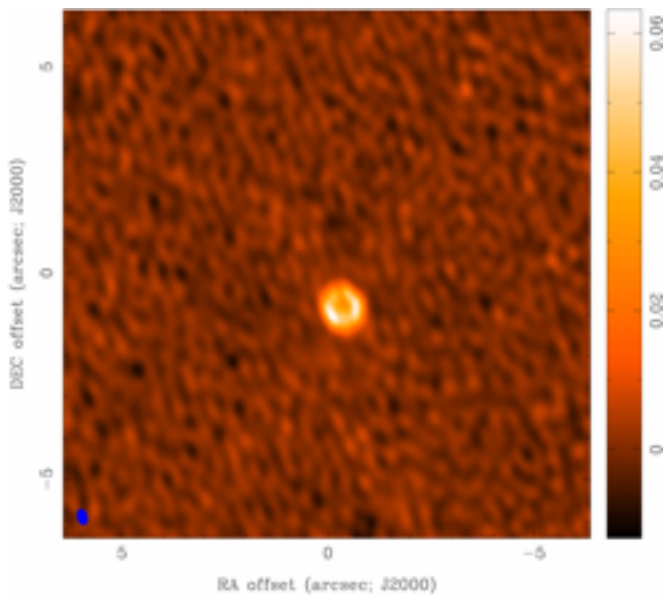
$T^D(x,y)$

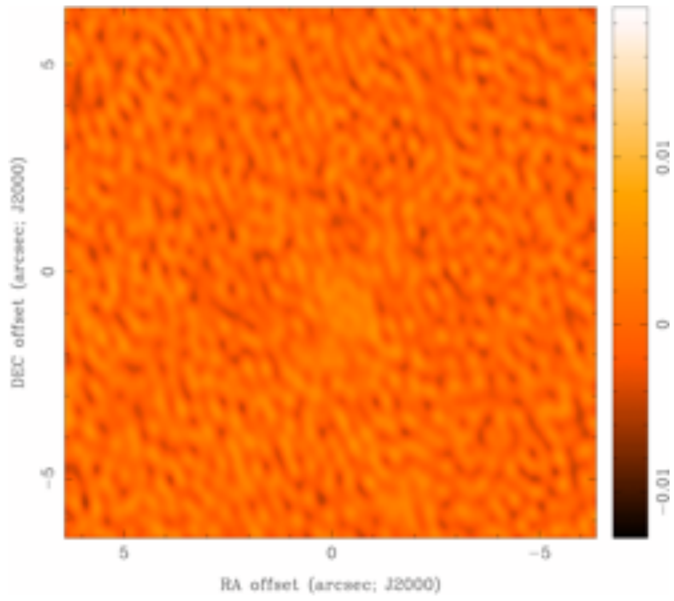




CLEAN model

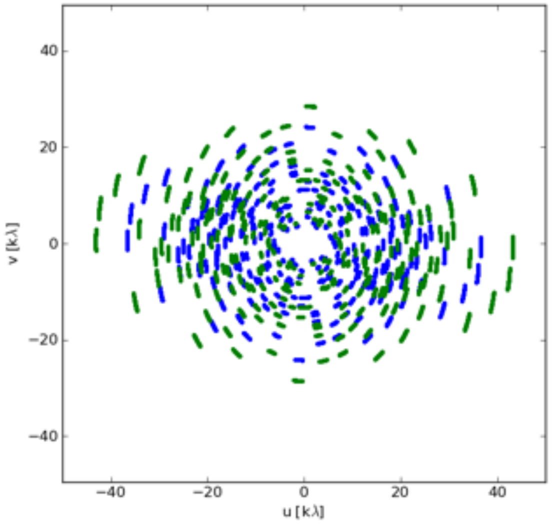
restored image





residual map

Weighting

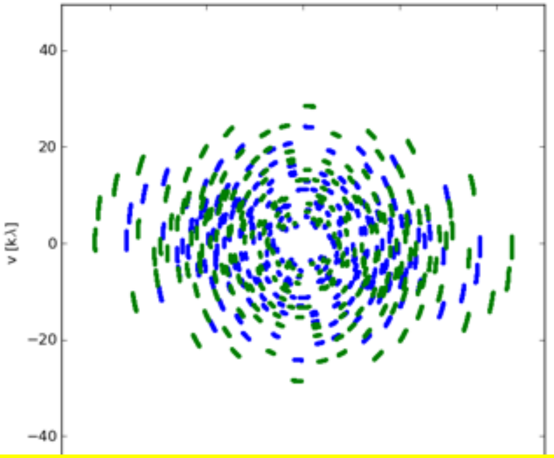


- For FT, uv-space needs to be gridded, cells sampled differently
- Each visibility point is given a weight in the imaging step
- First piece: weight given by T_{sys}, integration time, etc.
- **Natural**
  - Each sample is given the same weight
  - There are many samples at short baselines, so natural weighting will give the largest beam and the best surface brightness sensitivity (and sometimes pronounced wings in the dirty beam)
- **Uniform**
  - each visibility is given a weight inversely proportional to the sample density
  - Weighs down short baselines, long baselines are more pronounced. Best resolution; poorer noise characteristics
- **Briggs (Robust)**
  - A graduated scheme using the parameter *robust;* compromise of noise and resolution
  - In CASA, set *robust* from -2 (~ uniform) to +2 (~ natural)
  - *robust* = 0 often a good choice

- **Taper:** additional weight function to be applied (typically a Gaussian to suppress the weights of the outer visibilities – be careful, however, not to substantially reduce the collecting area)

Weighting

- For FT, uv-space needs to be gridded, cells sampled differently
- Each visibility point is given a weight in the imaging step
- First piece: weight given by T_{sys}, integration time, etc.
- **Natural**



From a <u>single</u> visibility dataset to a <u>multitude of possible images</u>!

Adjust the weighting to match your science goal:

- Detection experiment/weak extended source:
natural (maybe even with a taper).
- Finer detail of strong sources: **robust** or even **uniform**

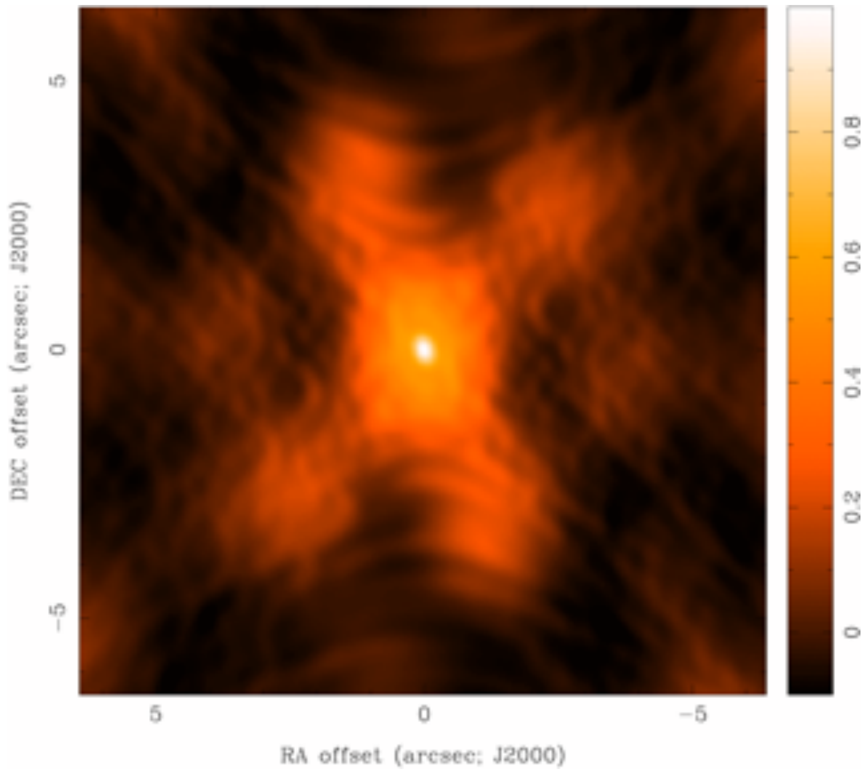
(best, when possible: model the visibilities directly)

- *robust* = 0 often a good choice
- **Taper:** additional weight function to be applied (typically a Gaussian suppress the weights of the outer visibilities – be careful, however, not to substantially reduce the collecting area)

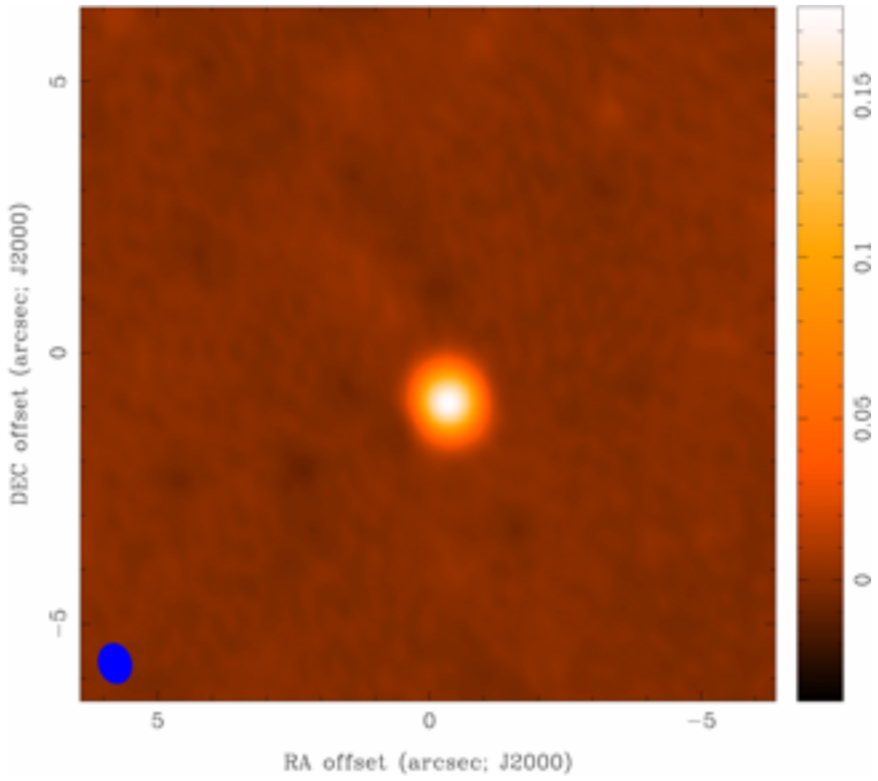
Imaging Results

(sensitivity optimized, but bad sidelobes)

Natural Weight Dirty Beam



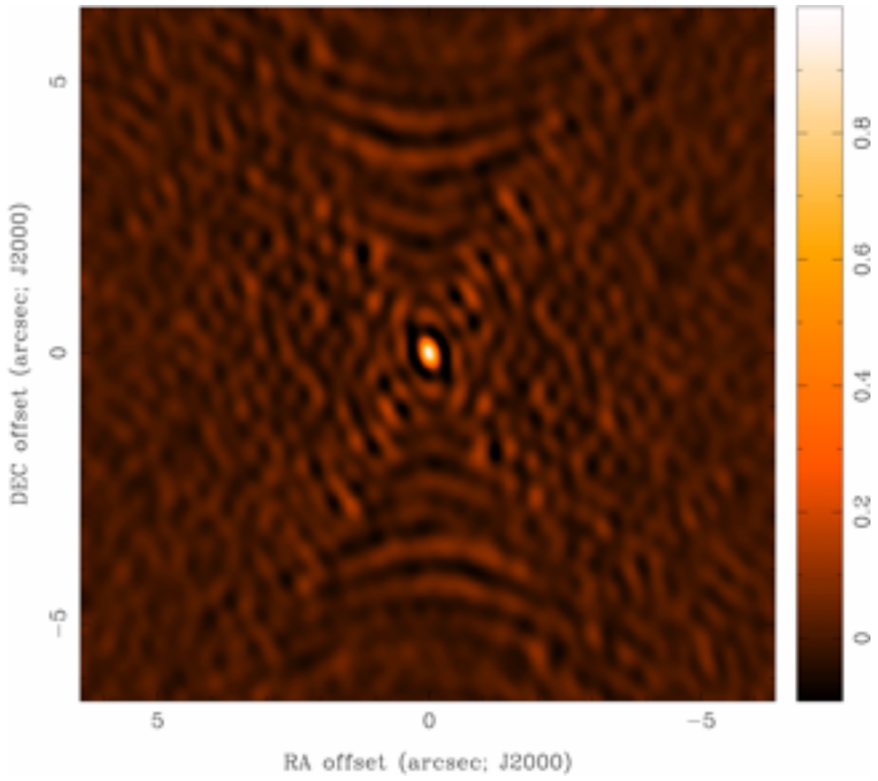
CLEAN image



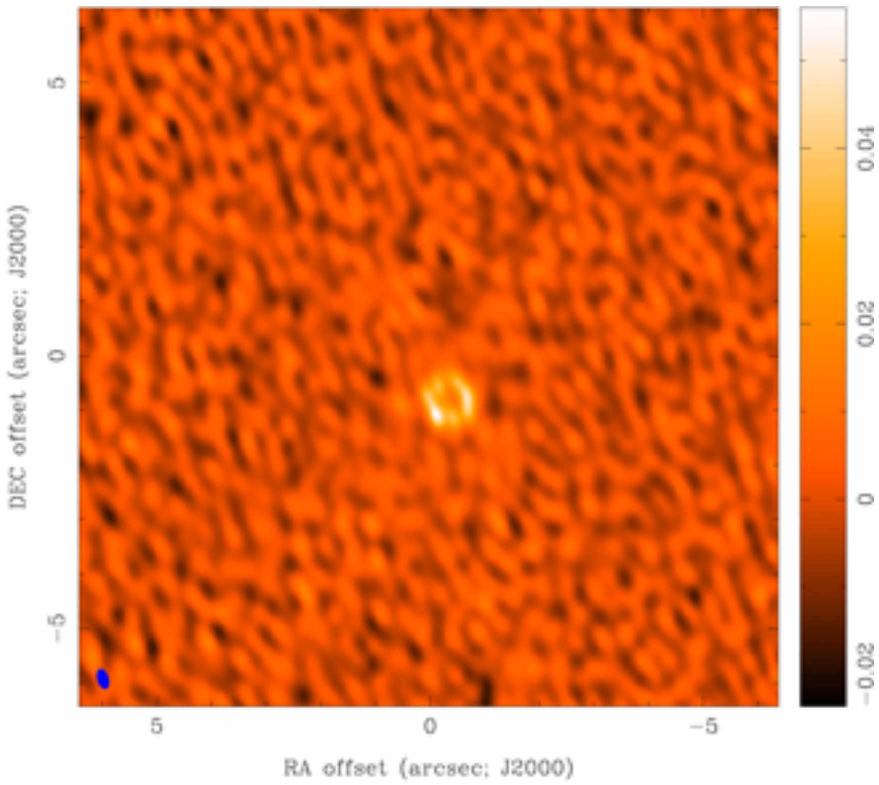
Imaging Results

(angular resolution optimized)

Uniform Weight Dirty Beam



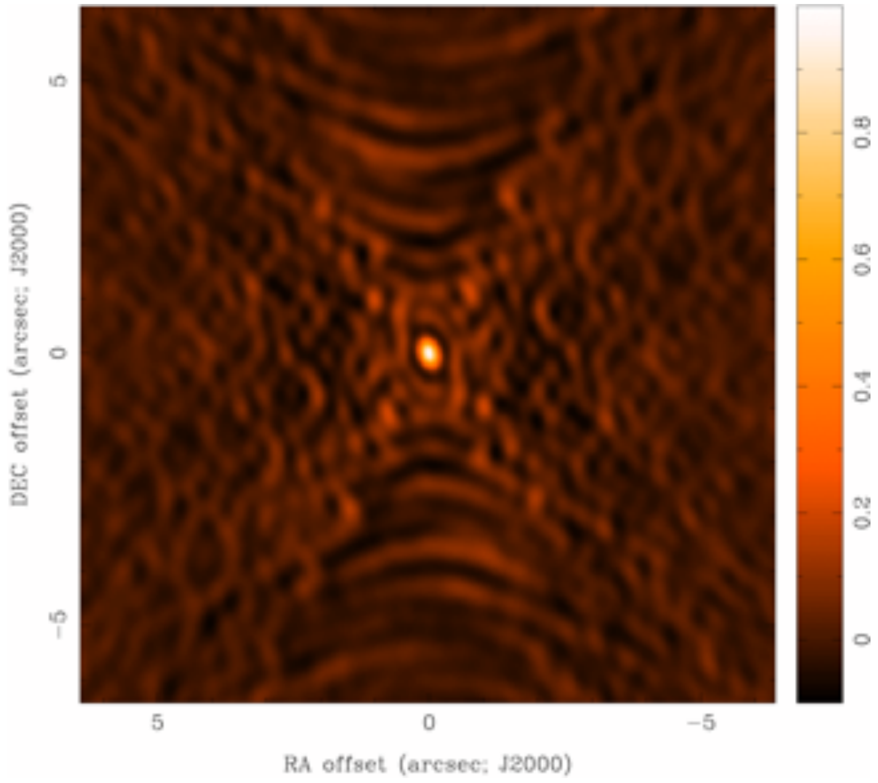
CLEAN image



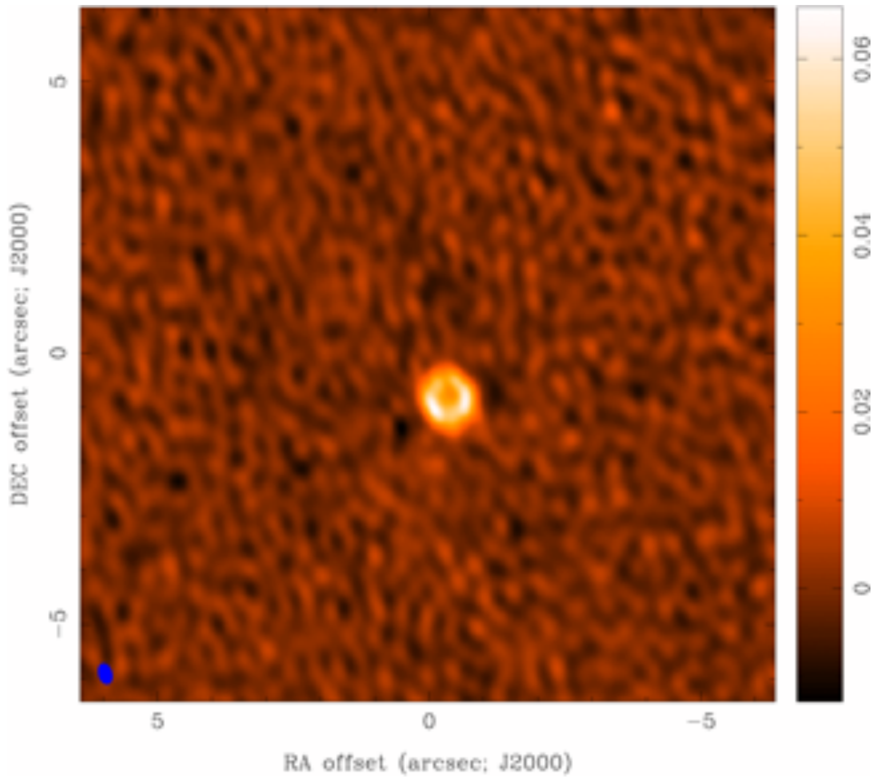
Imaging Results

(good compromise)

Robust=0 Dirty Beam



CLEAN image



CLEAN in CASA

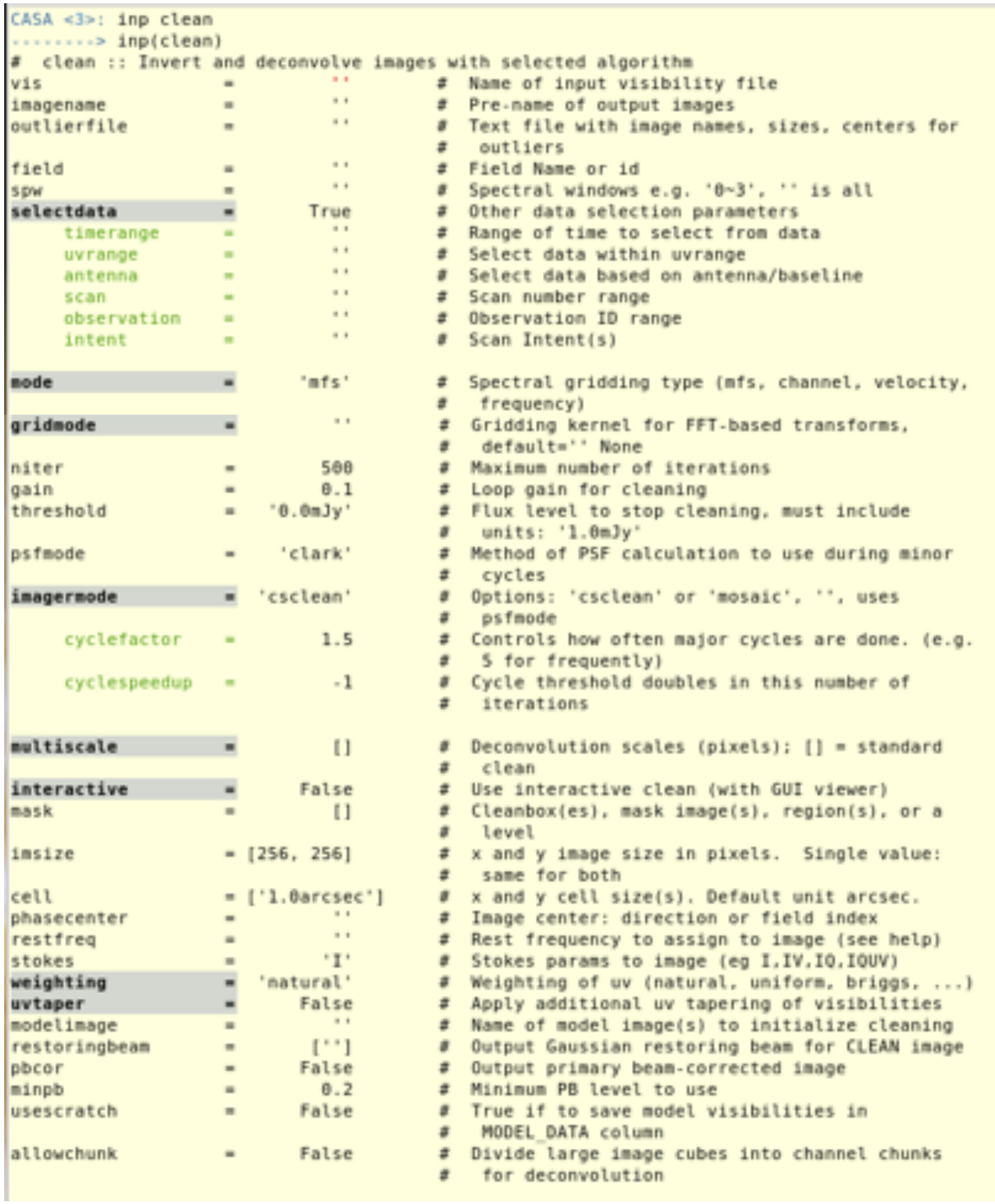
CLEAN is **the** imaging task in CASA. It:

- takes the calibrated visibilities
- grids them on the UV-plane
- performs the FFT to a dirty image
- deconvolves the image
- restores the image from clean component list and residual

Modes/Capabilities:

- continuum: incl. multi-frequency synthesis (radial extend of each visibility due to bandwidth), and Taylor term expansion (to derive spectral index and curvature
- spectral line: data cubes (many planes) grids in velocity space, takes account of Doppler shift of line
- mosaicking: combine multiple pointings to single image
- w-projection/faceting for images beyond the half-power point
- outlier fields to deconvolve strong sources in primary beam sidelobes
- multiscale cleaning
- primary beam correction

CLEAN in CASA:



```
CASA <3>: inp clean
--------> inp(clean)
#  clean :: Invert and deconvolve images with selected algorithm
vis                 =         ''        #  Name of input visibility file
imagename           =         ''        #  Pre-name of output images
outlierfile         =         ''        #  Text file with image names, sizes, centers for
                                        #   outliers
field               =         ''        #  Field Name or id
spw                 =         ''        #  Spectral windows e.g. '0~3', '' is all
selectdata          =       True        #  Other data selection parameters
     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)

mode                =      'mfs'        #  Spectral gridding type (mfs, channel, velocity,
                                        #   frequency)
gridmode            =         ''        #  Gridding kernel for FFT-based transforms,
                                        #   default='' None
niter               =        500        #  Maximum number of iterations
gain                =        0.1        #  Loop gain for cleaning
threshold           =   '0.0mJy'        #  Flux level to stop cleaning, must include
                                        #   units: '1.0mJy'
psfmode             =    'clark'        #  Method of PSF calculation to use during minor
                                        #   cycles
imagermode          =  'csclean'        #  Options: 'csclean' or 'mosaic', '', uses
                                        #   psfmode
     cyclefactor    =        1.5        #  Controls how often major cycles are done. (e.g.
                                        #   5 for frequently)
     cyclespeedup   =         -1        #  Cycle threshold doubles in this number of
                                        #   iterations

multiscale          =         []        #  Deconvolution scales (pixels); [] = standard
                                        #   clean
interactive         =      False        #  Use interactive clean (with GUI viewer)
mask                =         []        #  Cleanbox(es), mask image(s), region(s), or a
                                        #   level
imsize              = [256, 256]        #  x and y image size in pixels.  Single value:
                                        #   same for both
cell                = ['1.0arcsec']     #  x and y cell size(s). Default unit arcsec.
phasecenter         =         ''        #  Image center: direction or field index
restfreq            =         ''        #  Rest frequency to assign to image (see help)
stokes              =        'I'        #  Stokes params to image (eg I,IV,IQ,IQUV)
weighting           =  'natural'        #  Weighting of uv (natural, uniform, briggs, ...)
uvtaper             =      False        #  Apply additional uv tapering of visibilities
modelimage          =         ''        #  Name of model image(s) to initialize cleaning
restoringbeam       =       ['']        #  Output Gaussian restoring beam for CLEAN image
pbcor               =      False        #  Output primary beam-corrected image
minpb               =        0.2        #  Minimum PB level to use
usescratch          =      False        #  True if to save model visibilities in
                                        #   MODEL_DATA column
allowchunk          =      False        #  Divide large image cubes into channel chunks
                                        #   for deconvolution
```

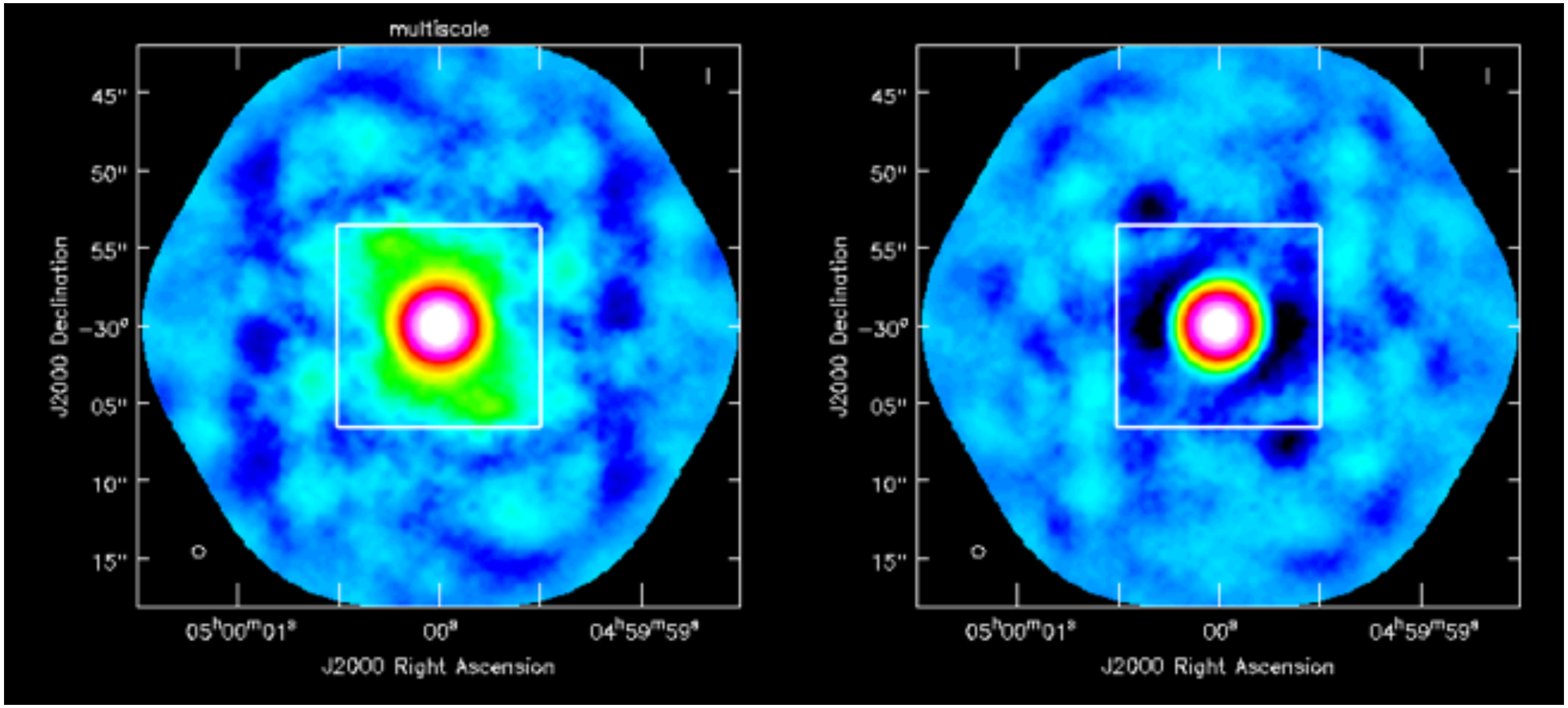
Output of CLEAN

• my_image.flux	Relative sky sensitivity
• my_image.image	Cleaned and restored image (Jy/clean beam)
• my_image.mask	Clean “boxes”
• my_image.model	Clean components (Jy/pixel)
• my_image.psf	Dirty beam
• my_image.residual	Residual (Jy/dirty beam)

Each file is a CASA image

Multi-scale CLEAN

multi-scale | “classic” scale

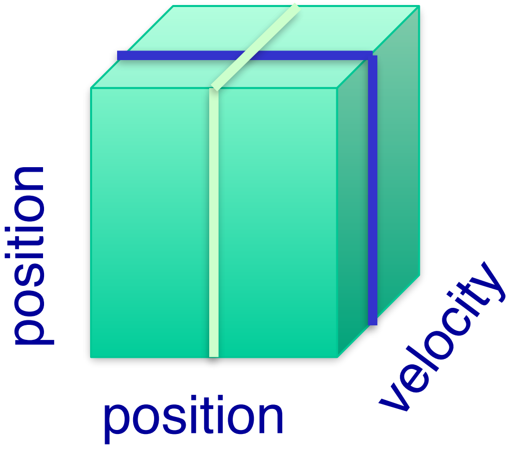


```
multiscale        = [0, 5, 12, 24, 50] #  Deconvolution scales (pixels); [] =
                                       #  standard clean
```

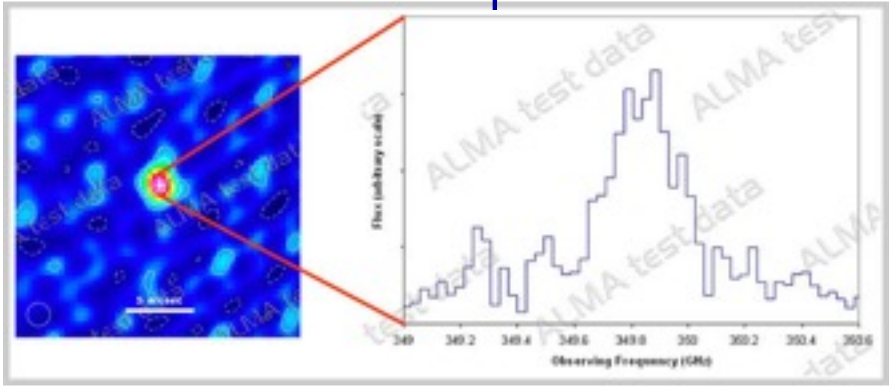
Instead of delta functions (pixels), one can use extended clean components to better match emission scales

Pick delta function, half the largest emission and a few in between

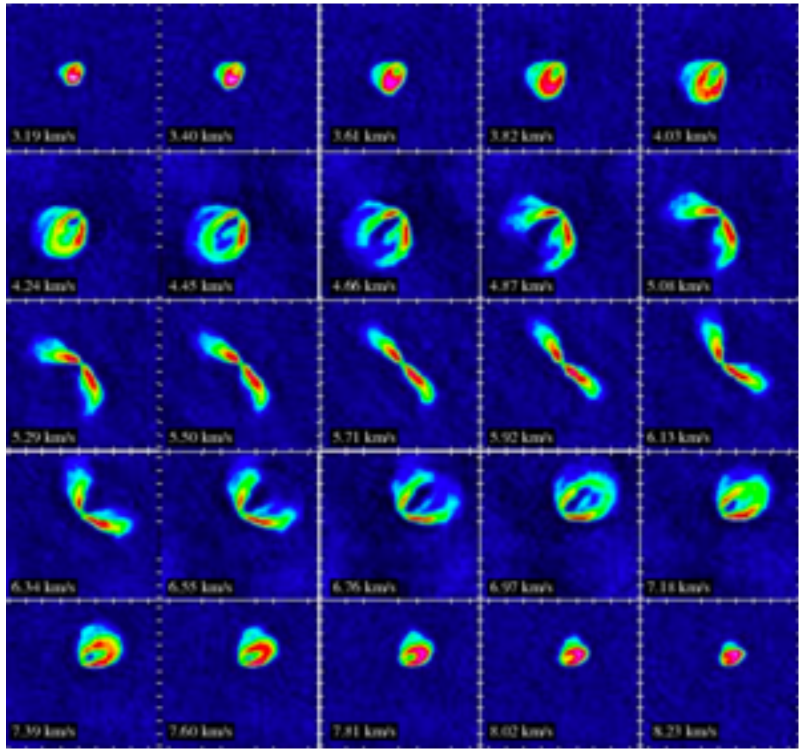
Imaging spectral lines



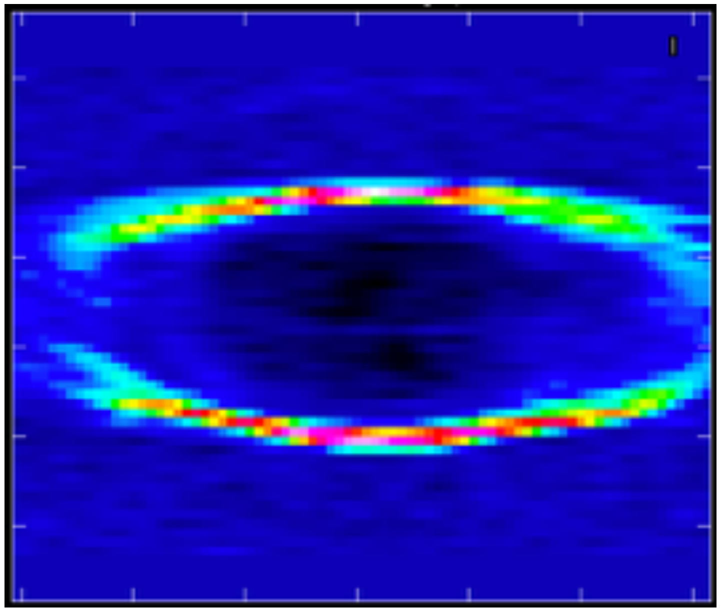
- Spectrum



- Channel map

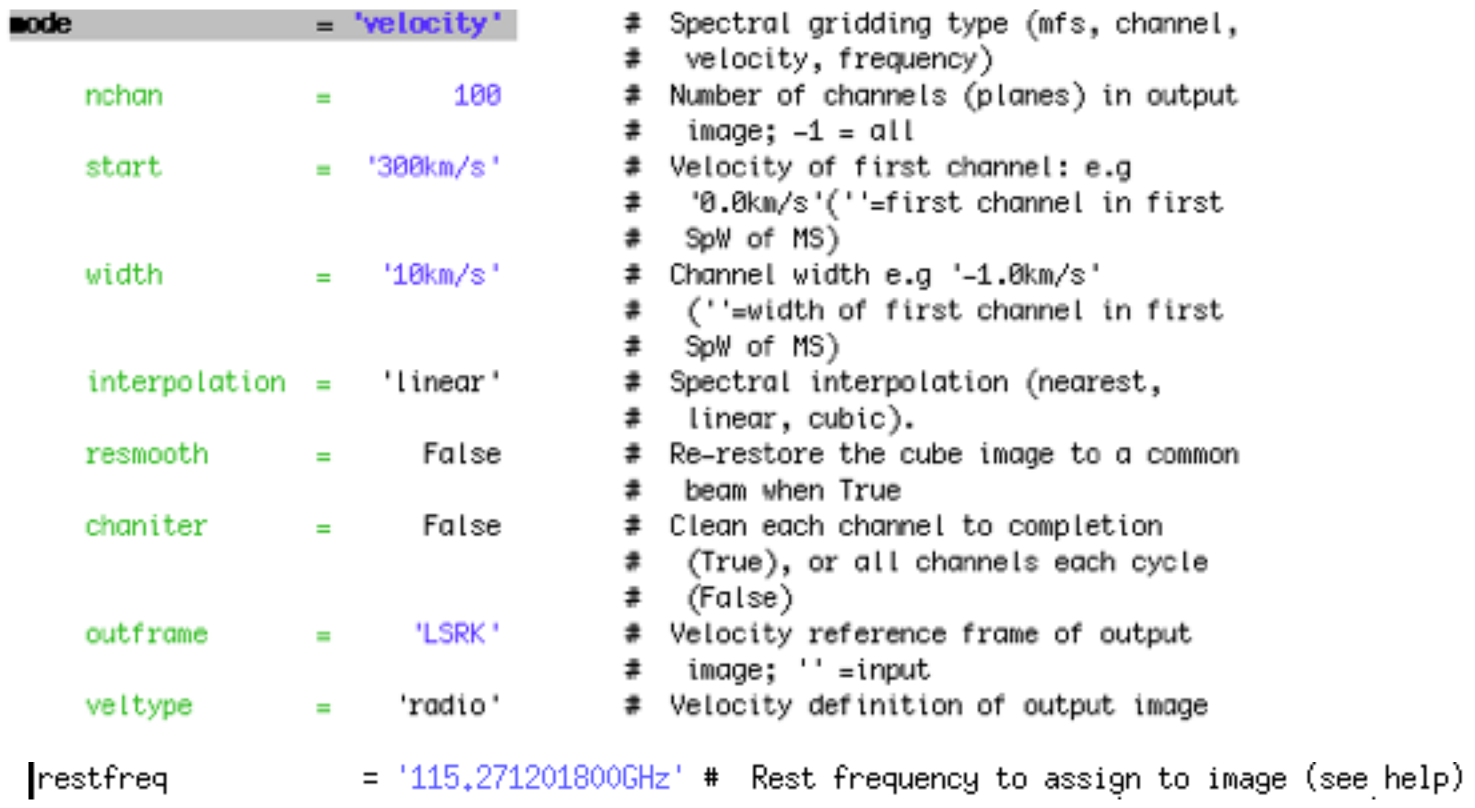


- Position-velocity map



(Along one spatial direction)

Imaging spectral lines

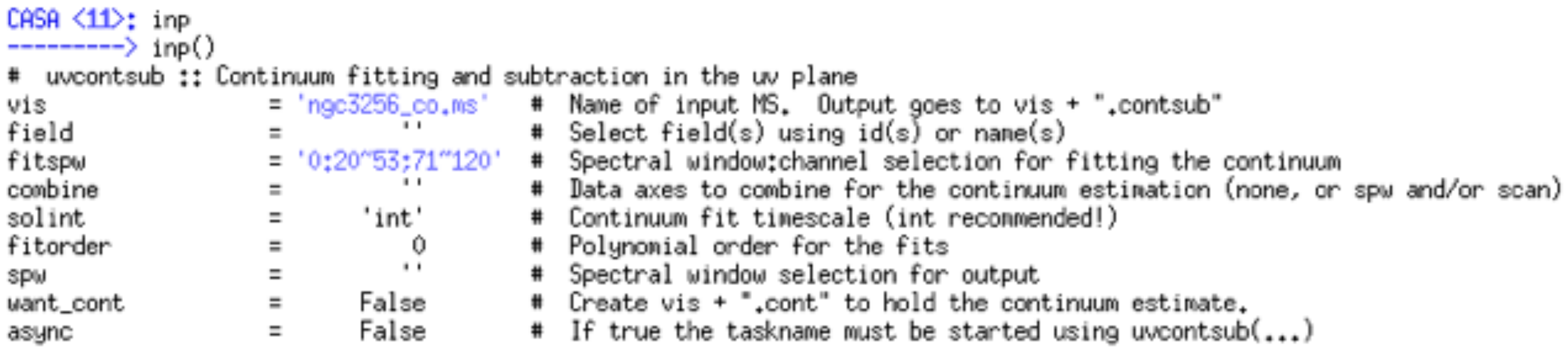


```
mode                =  'velocity'       #  Spectral gridding type (mfs, channel,
                                        #   velocity, frequency)
     nchan          =        100        #  Number of channels (planes) in output
                                        #   image; -1 = all
     start          =   '300km/s'       #  Velocity of first channel: e.g
                                        #   '0.0km/s'(''=first channel in first
                                        #   SpW of MS)
     width          =    '10km/s'       #  Channel width e.g '-1.0km/s'
                                        #   (''=width of first channel in first
                                        #   SpW of MS)
     interpolation  =    'linear'       #  Spectral interpolation (nearest,
                                        #   linear, cubic).
     resmooth       =      False        #  Re-restore the cube image to a common
                                        #   beam when True
     chaniter       =      False        #  Clean each channel to completion
                                        #   (True), or all channels each cycle
                                        #   (False)
     outframe       =      'LSRK'       #  Velocity reference frame of output
                                        #   image; '' =input
     veltype        =     'radio'       #  Velocity definition of output image

 restfreq              = '115.271201800GHz' #  Rest frequency to assign to image (see help)
```

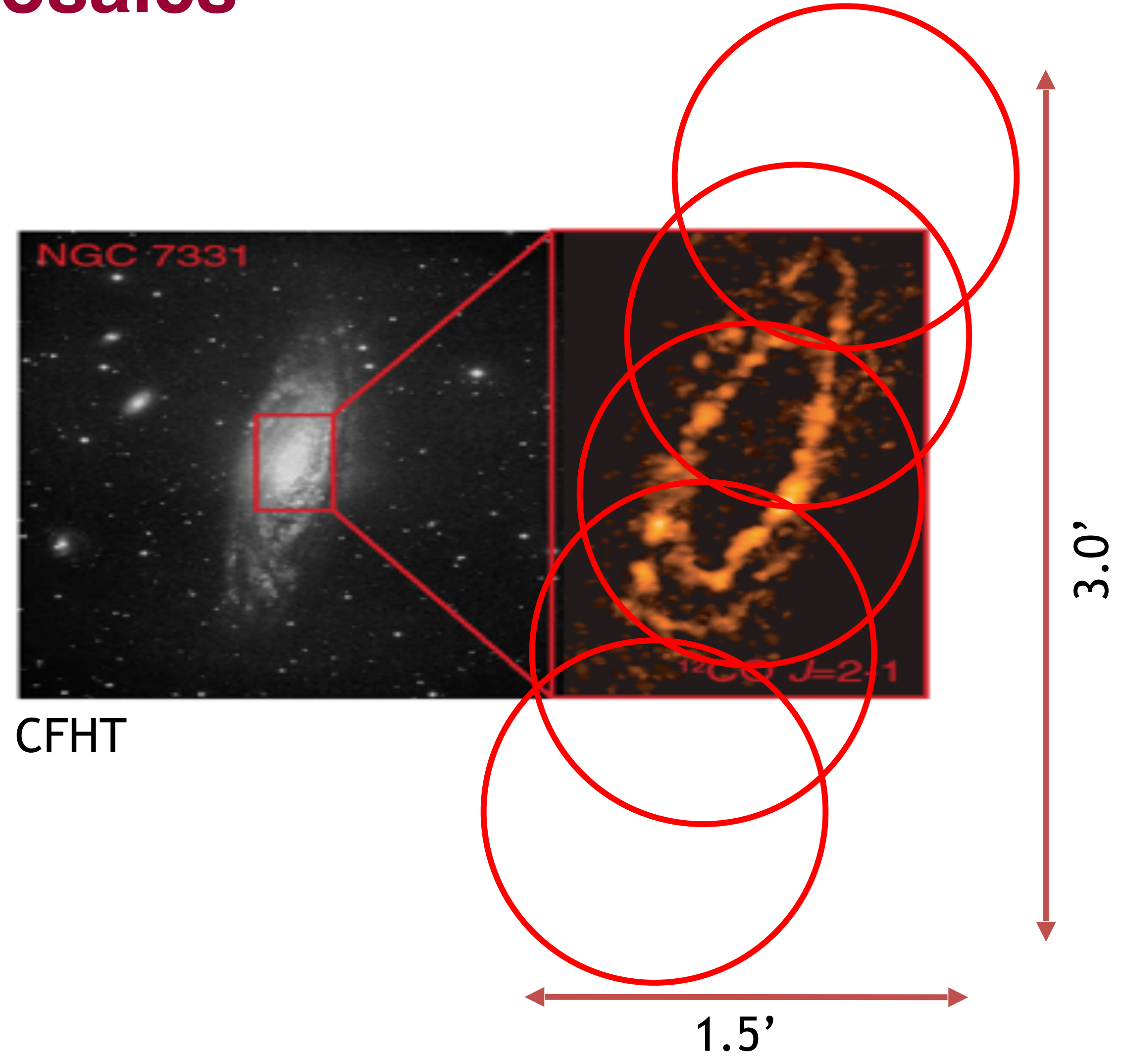
Imaging spectral lines: continuum subtraction

- Generally would like to subtract continuum emission (need to identify line-free channels first)
- Use `uvcontsub` to do the subtraction in uv plane.



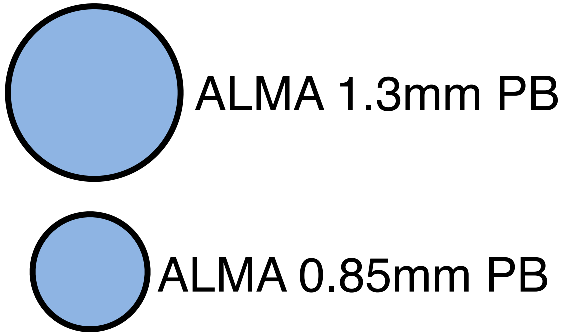
```
CASA <11>: inp
---------> inp()
#  uvcontsub :: Continuum fitting and subtraction in the uv plane
vis                 = 'ngc3256_co.ms'   #  Name of input MS.  Output goes to vis + ".contsub"
field               =        ''         #  Select field(s) using id(s) or name(s)
fitspw              = '0:20~53;71~120'  #  Spectral window:channel selection for fitting the continuum
combine             =        ''         #  Data axes to combine for the continuum estimation (none, or spw and/or scan)
solint              =      'int'        #  Continuum fit timescale (int recommended!)
fitorder            =         0         #  Polynomial order for the fits
spw                 =        ''         #  Spectral window selection for output
want_cont           =      False        #  Create vis + ".cont" to hold the continuum estimate.
async               =      False        #  If true the taskname must be started using uvcontsub(...)
```

Mosaics



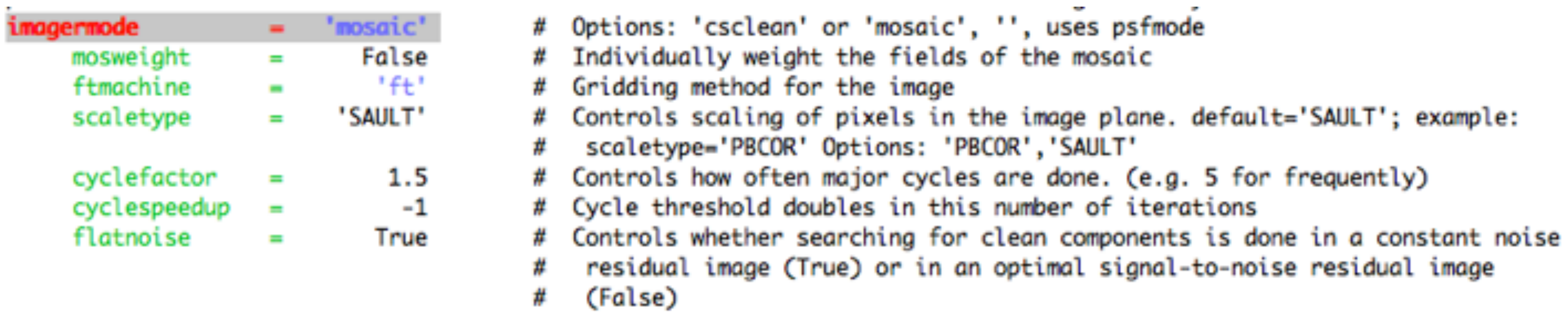
Example: SMA 1.3 mm observations: 5 pointings

- Primary beam ~1'
- Resolution ~3"



Petitpas et al.

Imaging mosaics with CLEAN



```
imagermode     =   'mosaic'   #  Options: 'csclean' or 'mosaic', '', uses psfmode
     mosweight =      False   #  Individually weight the fields of the mosaic
     ftmachine =       'ft'   #  Gridding method for the image
     scaletype =    'SAULT'   #  Controls scaling of pixels in the image plane. default='SAULT'; example:
                              #   scaletype='PBCOR' Options: 'PBCOR','SAULT'
     cyclefactor =      1.5   #  Controls how often major cycles are done. (e.g. 5 for frequently)
     cyclespeedup =      -1   #  Cycle threshold doubles in this number of iterations
     flatnoise =       True   #  Controls whether searching for clean components is done in a constant noise
                              #   residual image (True) or in an optimal signal-to-noise residual image
                              #   (False)
```

ftmachine = “mosaic” : add in uv plane and invert together,
Use *csclean* for deconvolution.

ftmachine = “ft” : shift and add in image plane

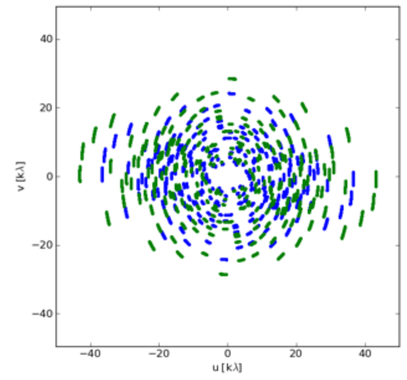
There’s a tool (“ia.linearmosaic”) to linear mosaic after cleaning each pointing and to stitch all pointings together entirely in the image domain

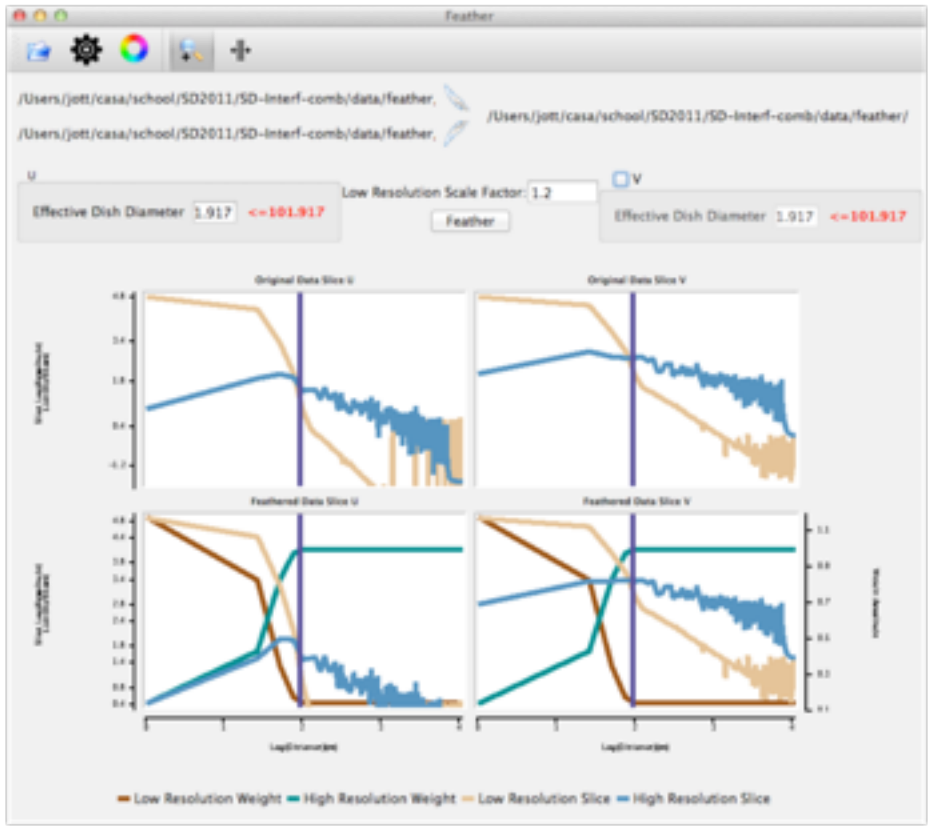
Combining with single-dish or other interferometric maps

- If you have only images:
  - feather (or “casafeather”)
- If you have an image and an MS:
  - use CLEAN with the image

    as “modelimage”
  - and/or feather
- If you have multiple MS plus an image:
  - Same as above, input to clean will

    be all the MS



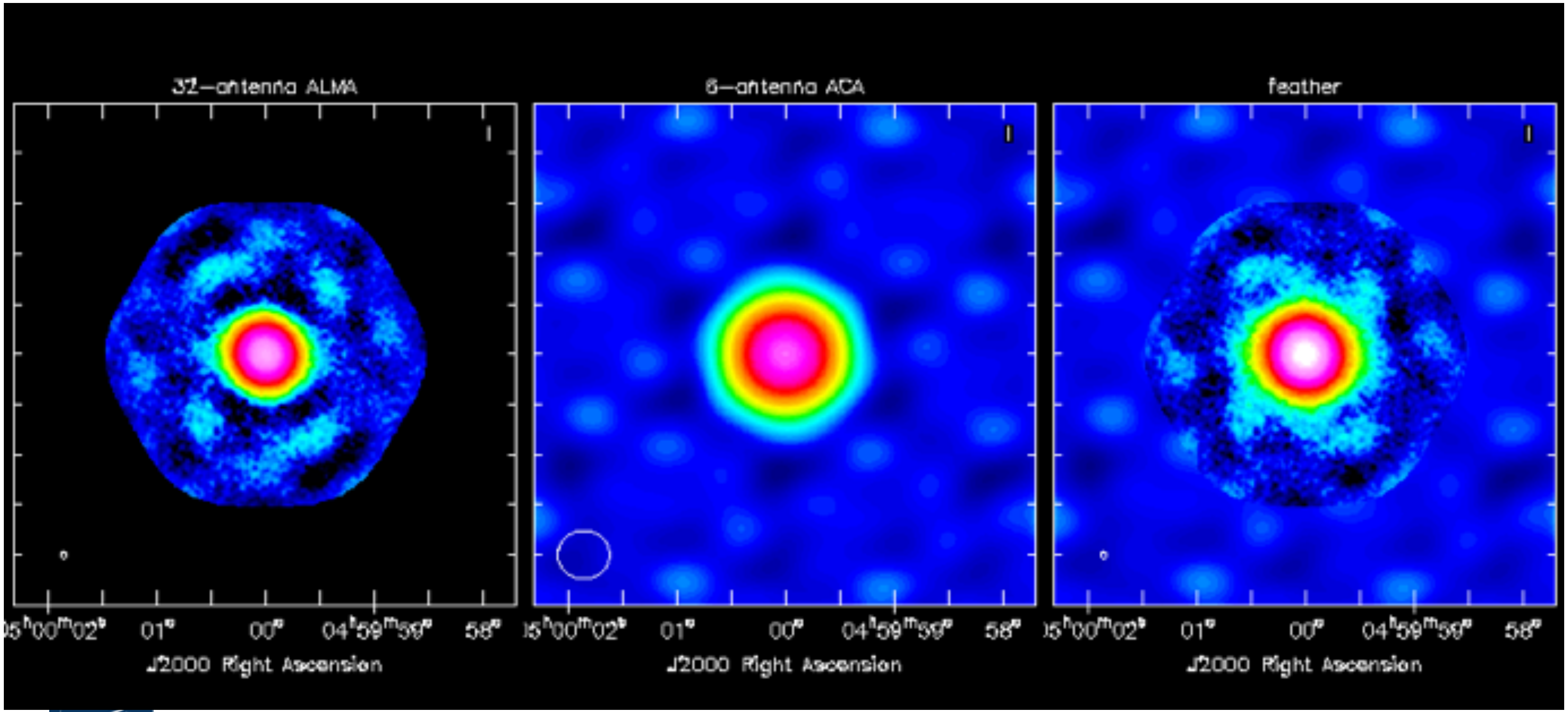


/Users/jott/casa/school/SD2011/SD-Interf-comb/data/feather,
/Users/jott/casa/school/SD2011/SD-Interf-comb/data/feather,
/Users/jott/casa/school/SD2011/SD-Interf-comb/data/feather/

U
Effective Dish Diameter 1.917 <=101.917
Low Resolution Scale Factor: 1.2
Feather
V
Effective Dish Diameter 1.917 <=101.917

Combining with other data: feather

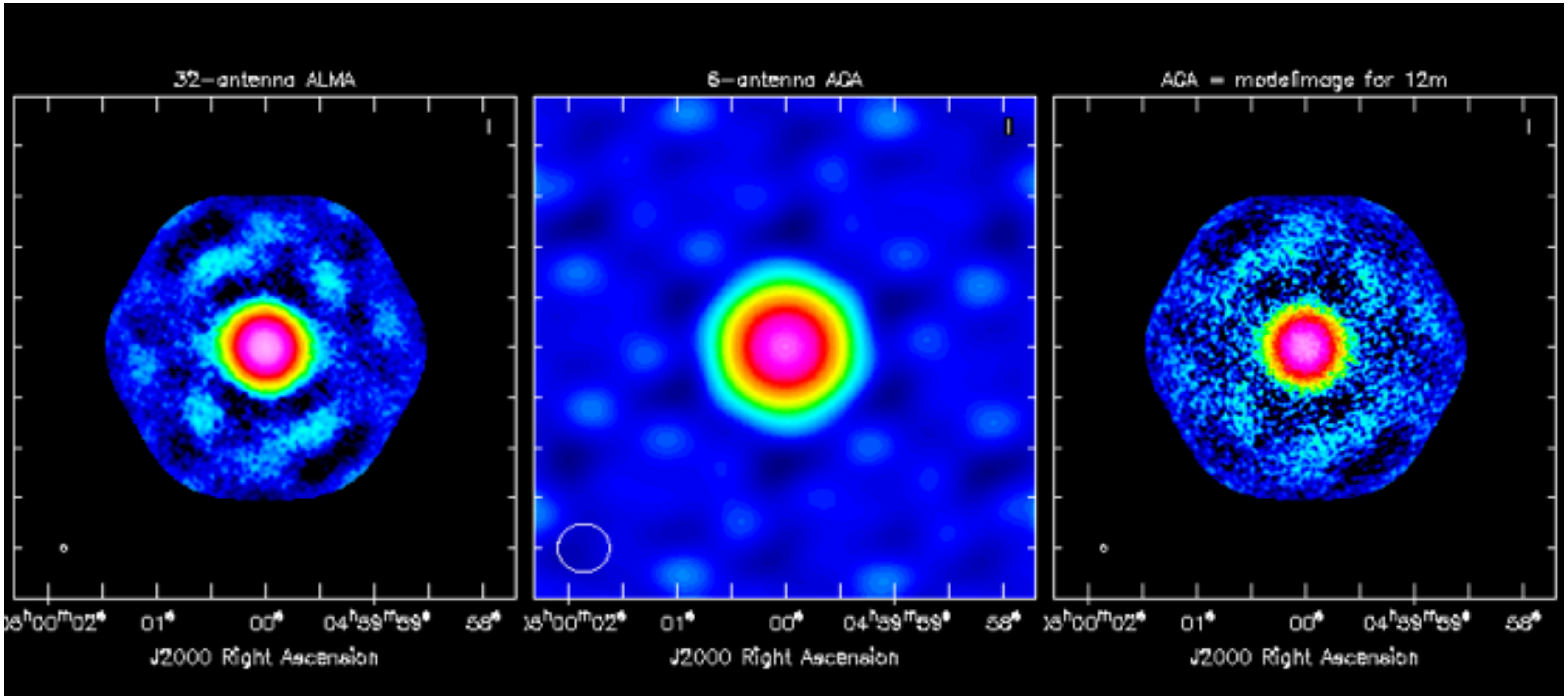
```
#  feather :: Combine two images using their Fourier transforms
imagename          =         ''         #  Name of output feathered image
highres            =         ''         #  Name of high resolution (interferometer) image
lowres             =         ''         #  Name of low resolution (single dish) image
async              =      False         #  If true the taskname must be started using feather(...)
```



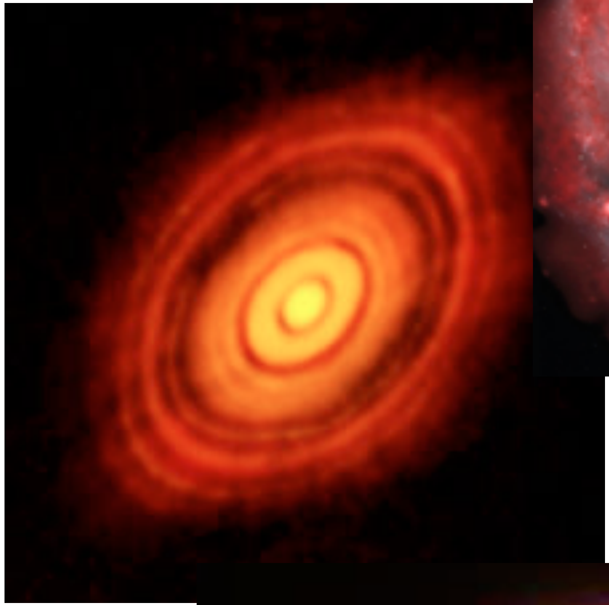
We also have a graphical tool: CASAfeather

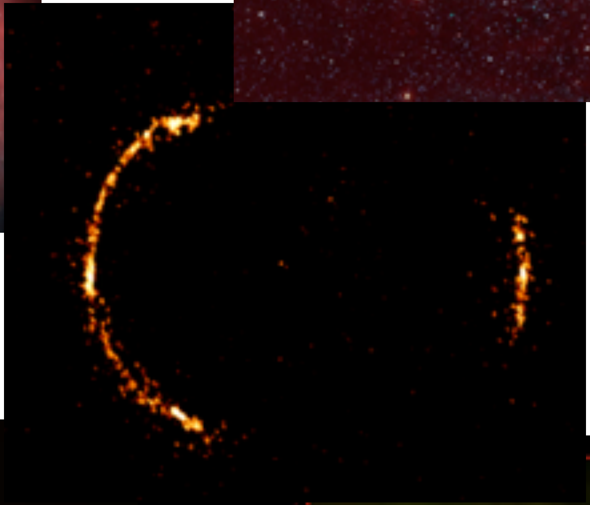
Combining with other data: modelimage

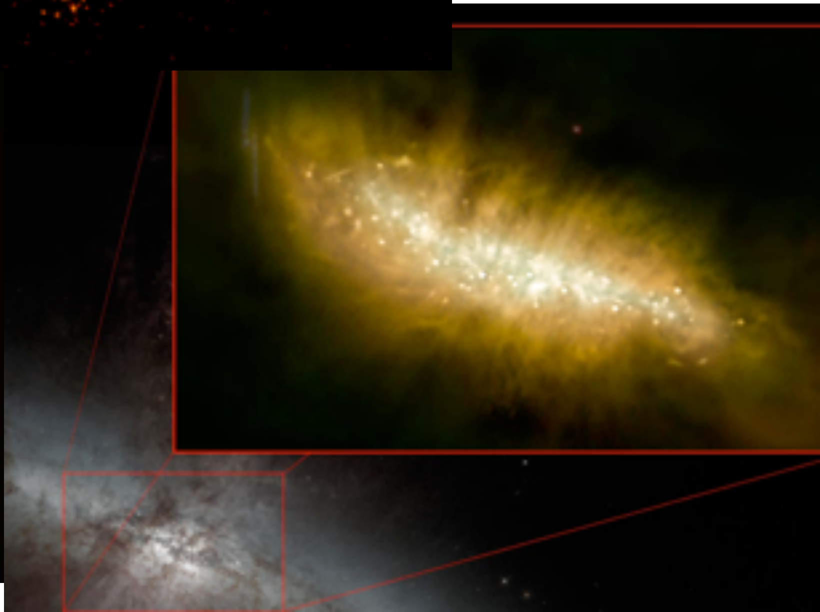
```
--------> inp(clean)
#  clean :: Invert and deconvolve images with selected algorithm
modelimage          =        ''        #  Name of model image(s) to initialize cleaning
```



… some CASA images…

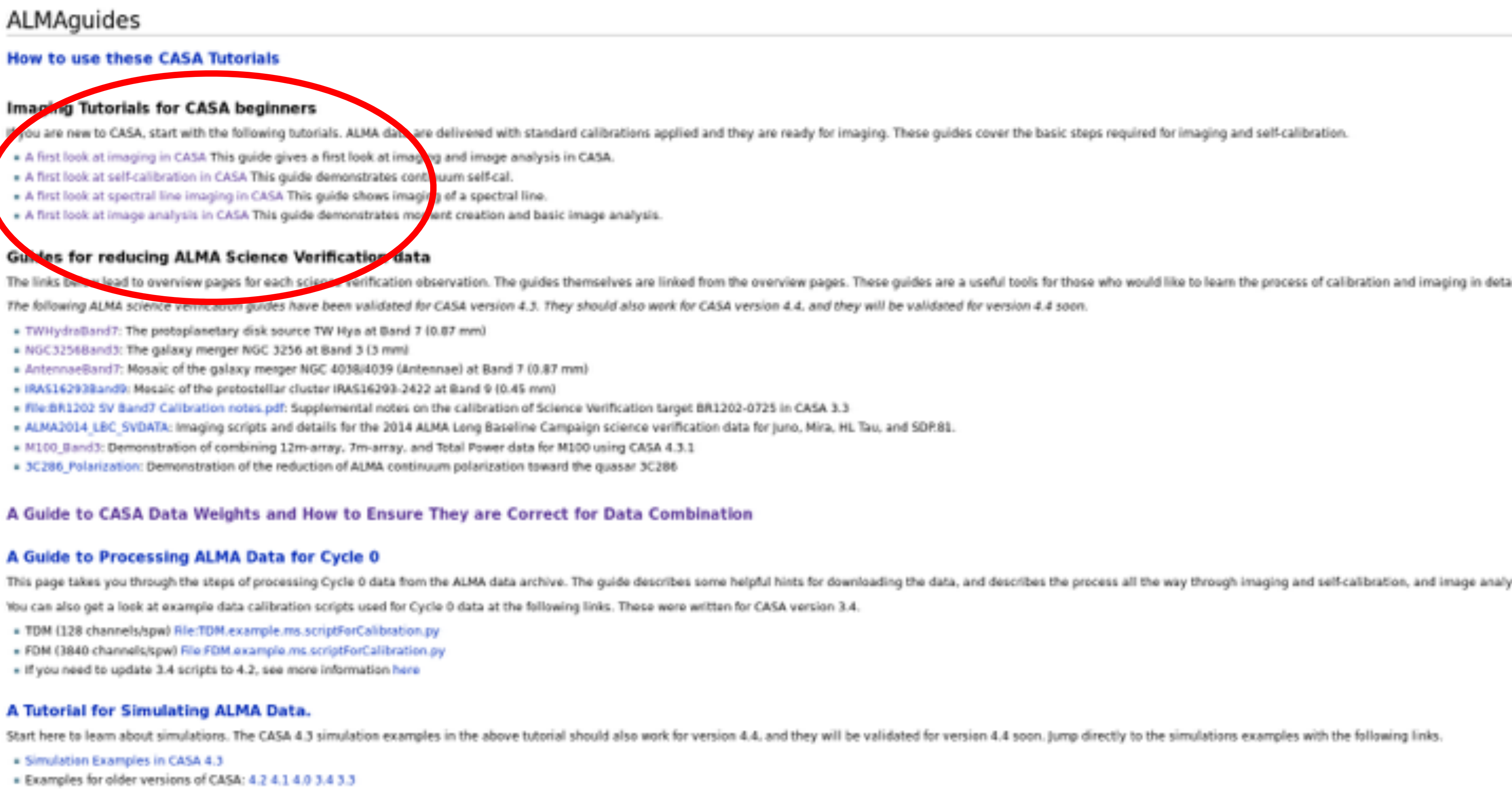






Online tutorials:

https://casaguides.nrao.edu/index.php/ALMAguides



ALMAguides

How to use these CASA Tutorials

Imaging Tutorials for CASA beginners

If you are new to CASA, start with the following tutorials. ALMA data are delivered with standard calibrations applied and they are ready for imaging. These guides cover the basic steps required for imaging and self-calibration.

- A first look at imaging in CASA This guide gives a first look at imaging and image analysis in CASA.
- A first look at self-calibration in CASA This guide demonstrates continuum self-cal.
- A first look at spectral line imaging in CASA This guide shows imaging of a spectral line.
- A first look at image analysis in CASA This guide demonstrates moment creation and basic image analysis.

Guides for reducing ALMA Science Verification data

The links below lead to overview pages for each science verification observation. The guides themselves are linked from the overview pages. These guides are a useful tools for those who would like to learn the process of calibration and imaging in deta

The following ALMA science verification guides have been validated for CASA version 4.3. They should also work for CASA version 4.4, and they will be validated for version 4.4 soon.

- TWHydraBand7: The protoplanetary disk source TW Hya at Band 7 (0.87 mm)
- NGC3256Band3: The galaxy merger NGC 3256 at Band 3 (3 mm)
- AntennaeBand7: Mosaic of the galaxy merger NGC 4038/4039 (Antennae) at Band 7 (0.87 mm)
- IRAS16293Band9: Mosaic of the protostellar cluster IRAS16293-2422 at Band 9 (0.45 mm)
- File:BR1202 SV Band7 Calibration notes.pdf: Supplemental notes on the calibration of Science Verification target BR1202-0725 in CASA 3.3
- ALMA2014_LBC_SVDATA: Imaging scripts and details for the 2014 ALMA Long Baseline Campaign science verification data for Juno, Mira, HL Tau, and SDP.81.
- M100_Band3: Demonstration of combining 12m-array, 7m-array, and Total Power data for M100 using CASA 4.3.1
- 3C286_Polarization: Demonstration of the reduction of ALMA continuum polarization toward the quasar 3C286

A Guide to CASA Data Weights and How to Ensure They are Correct for Data Combination

A Guide to Processing ALMA Data for Cycle 0

This page takes you through the steps of processing Cycle 0 data from the ALMA data archive. The guide describes some helpful hints for downloading the data, and describes the process all the way through imaging and self-calibration, and image analy

You can also get a look at example data calibration scripts used for Cycle 0 data at the following links. These were written for CASA version 3.4.

- TDM (128 channels/spw) File:TDM.example.ms.scriptForCalibration.py
- FDM (3840 channels/spw) File:FDM.example.ms.scriptForCalibration.py
- If you need to update 3.4 scripts to 4.2, see more information here

A Tutorial for Simulating ALMA Data.

Start here to learn about simulations. The CASA 4.3 simulation examples in the above tutorial should also work for version 4.4, and they will be validated for version 4.4 soon. Jump directly to the simulations examples with the following links.

- Simulation Examples in CASA 4.3
- Examples for older versions of CASA: 4.2 4.1 4.0 3.4 3.3

For more info:

http://www.almaobservatory.org

The Atacama Large Millimeter/submillimeter Array (ALMA), an international astronomy facility, is a partnership of the European Organisation for Astronomical Research in the Southern Hemisphere (ESO), the U.S. National Science Foundation (NSF) and the National Institutes of Natural Sciences (NINS) of Japan in cooperation with the Republic of Chile. ALMA is funded by ESO on behalf of its Member States, by NSF in cooperation with the National Research Council of Canada (NRC) and the National Science Council of Taiwan (NSC) and by NINS in cooperation with the Academia Sinica (AS) in Taiwan and the Korea Astronomy and Space Science Institute (KASI). ALMA construction and operations are led by ESO on behalf of its Member States; by the National Radio Astronomy Observatory (NRAO), managed by Associated Universities, Inc. (AUI), on behalf of North America; and by the National Astronomical Observatory of Japan (NAOJ) on behalf of East Asia. The Joint ALMA Observatory (JAO) provides the unified leadership and management of the construction, commissioning and operation of ALMA.