Basic Imaging

D. Petry

Slides by Martin Zwaan
(originally made for the ESO Cycle 0 PI CASA workshop, January 2012,
in turn heavily borrowed from R. Indebetouw, David Wilner,
Scott Schnee, Steve Myers, A. Richards)

Imaging Steps

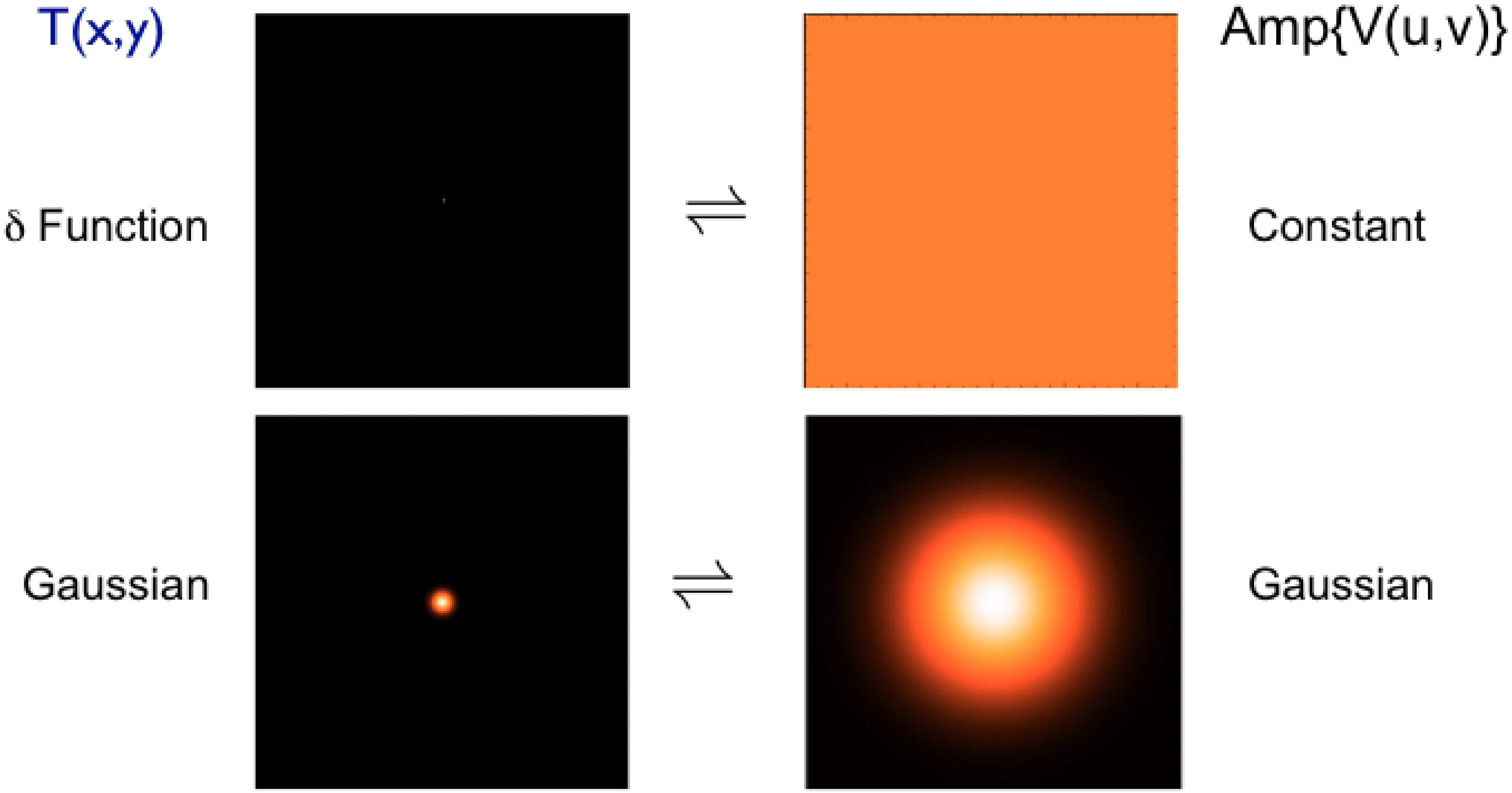
- An interferometer measures the interference pattern produced by two apertures.

- The interference pattern is directly related to the source brightness. In particular, for small fields of view the complex visibility, V(u,v), is the 2D Fourier transform of the brightness on the sky, T(x,y)

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

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

Some 2D Fourier Transform Pairs



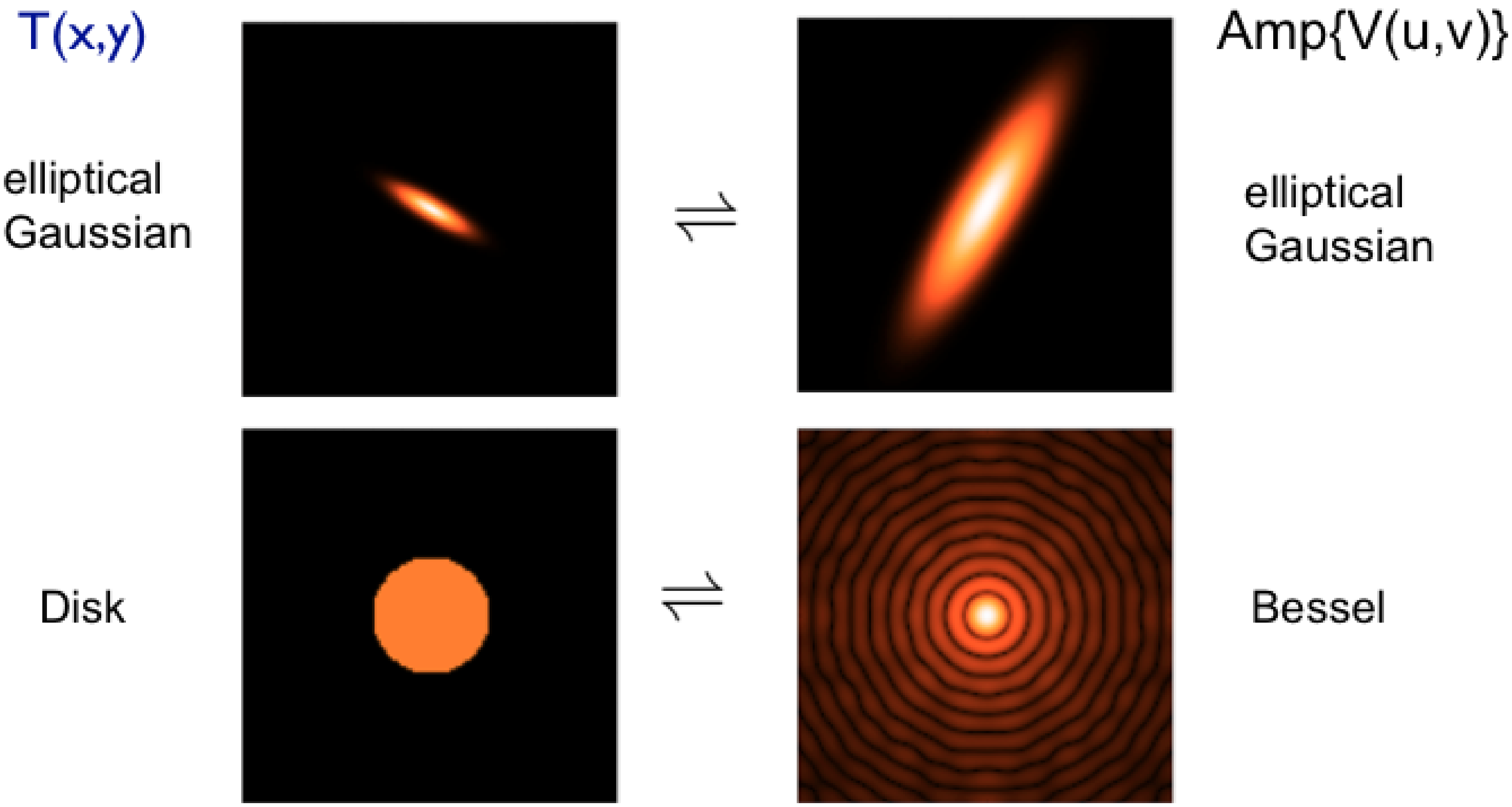
T(x,y)

Amp{V(u,v)}

δ Function ⇌ Constant

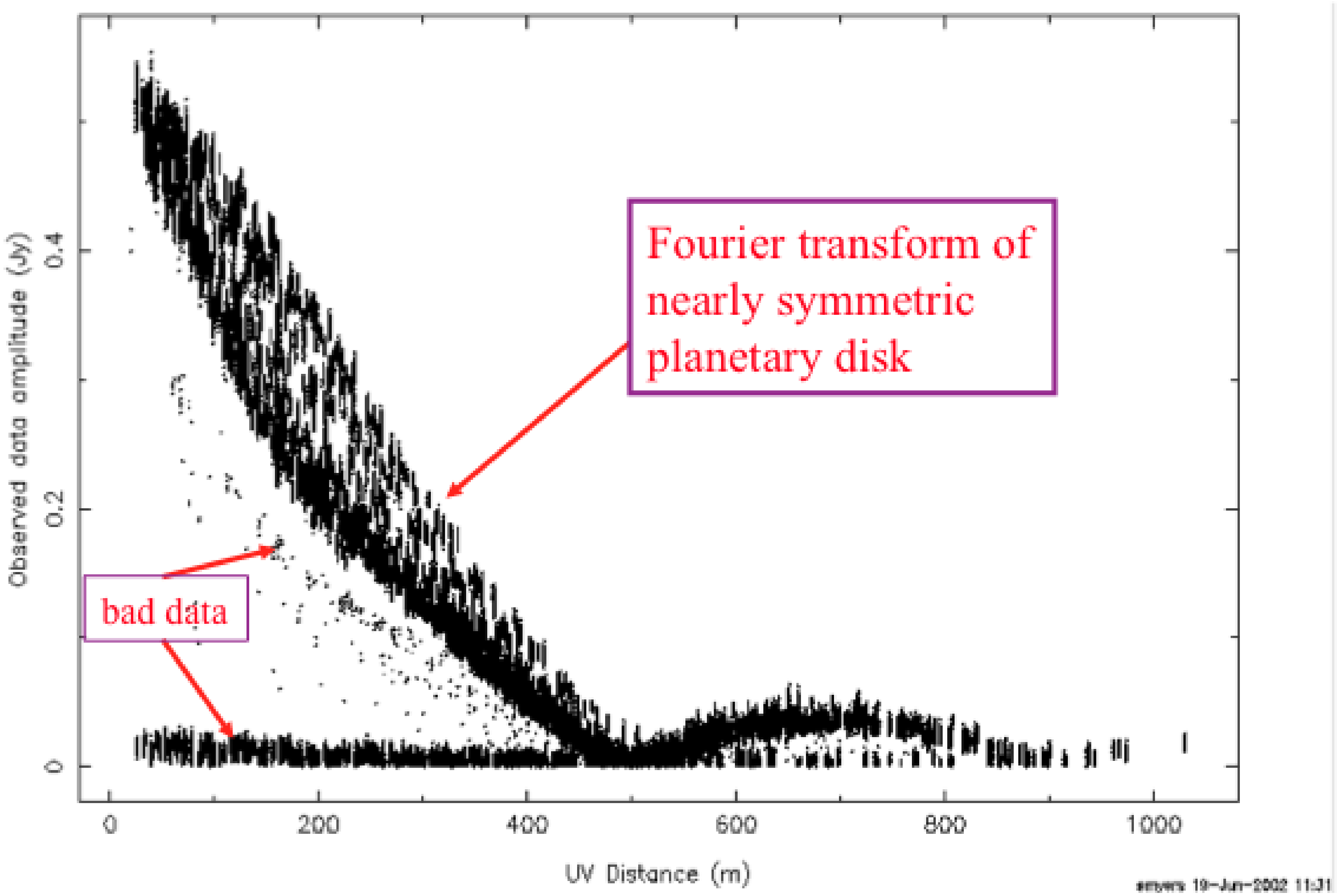
Gaussian ⇌ Gaussian

Some 2D Fourier Transform Pairs



Real example

- A 6cm VLA observation of Jupiter

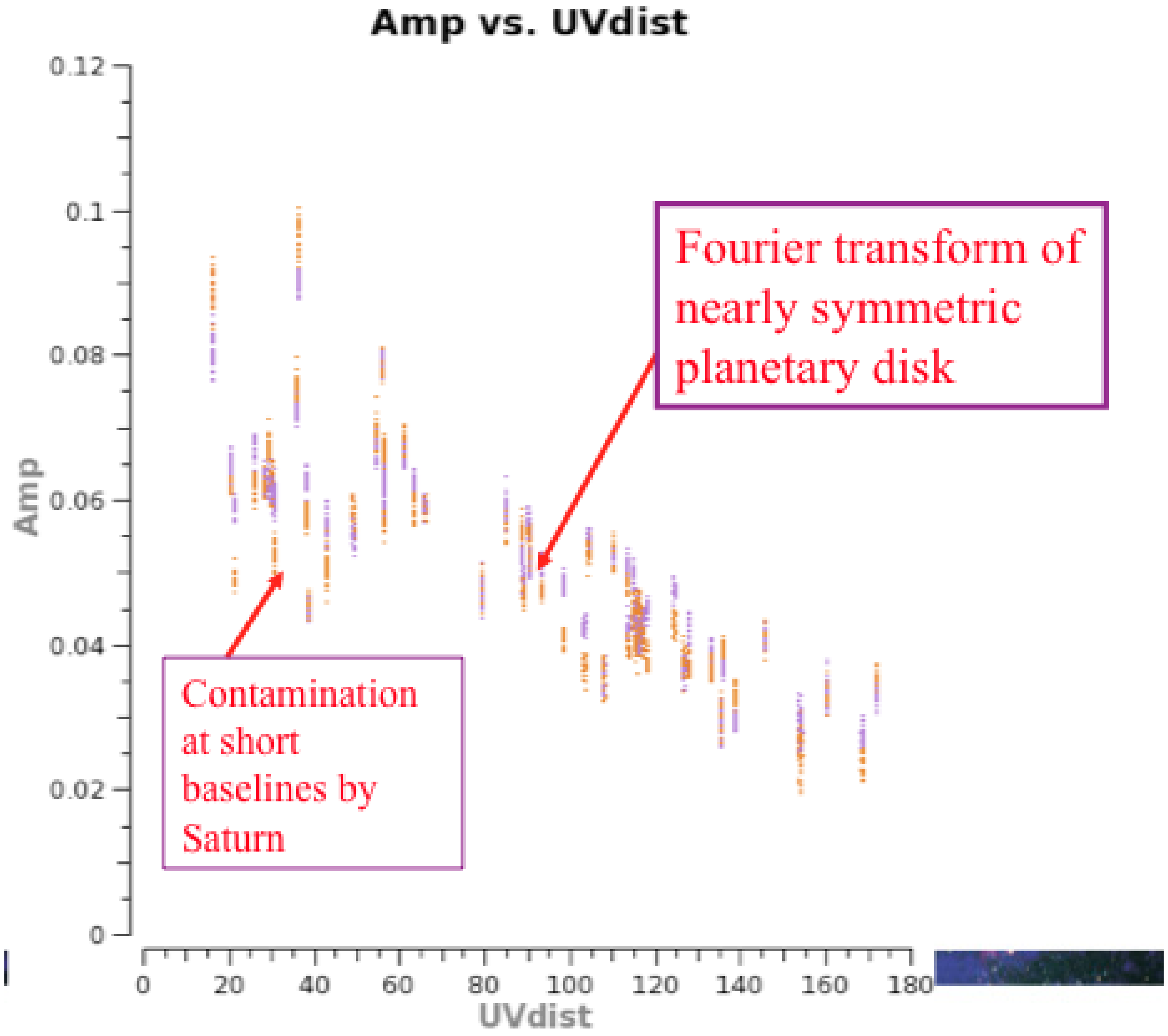


Fourier transform of nearly symmetric planetary disk

bad data

Real example

- ALMA observations of Titan



Fourier transform of nearly symmetric planetary disk

Contamination at short baselines by Saturn

But

- sample 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

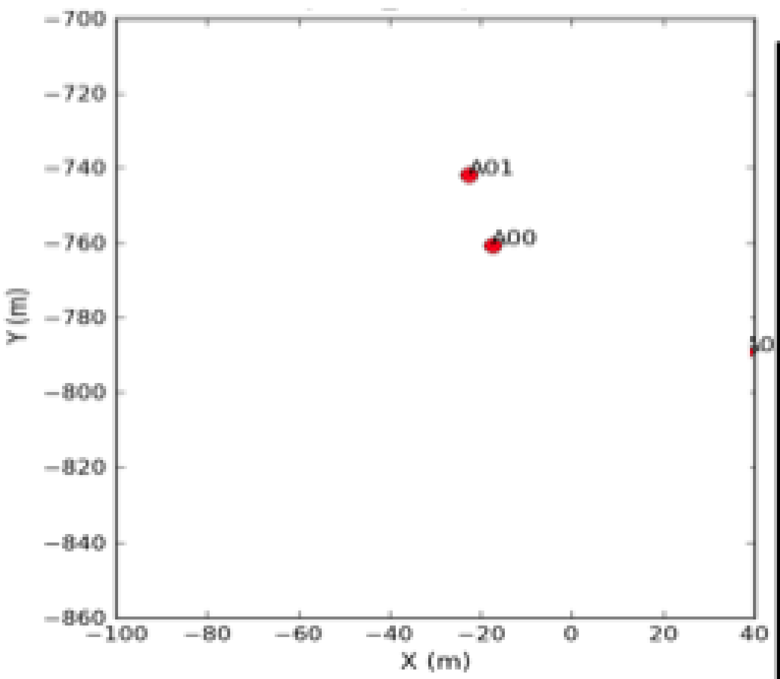
  $T^D(x,y) = b(x,y) \otimes T(x,y)$

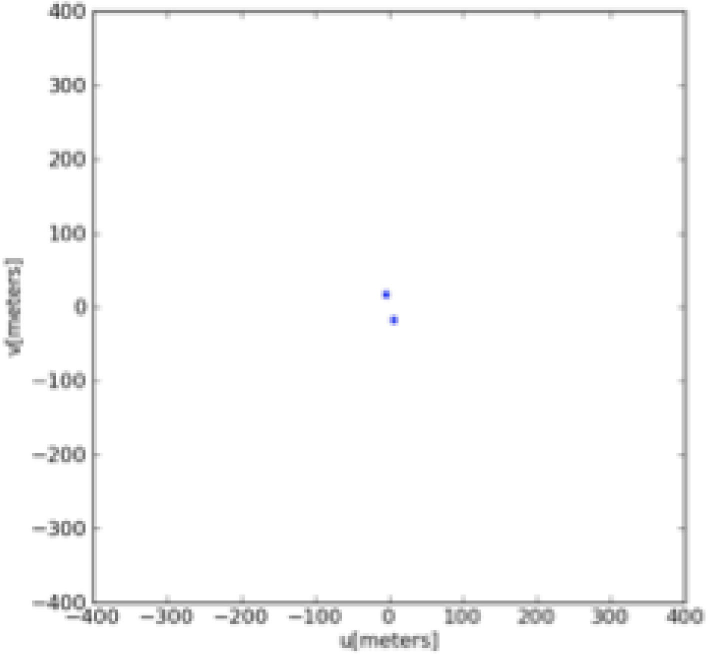
  where $b(x,y) = FT^{-1}\{B(u,v)\}$ (the point spread function)

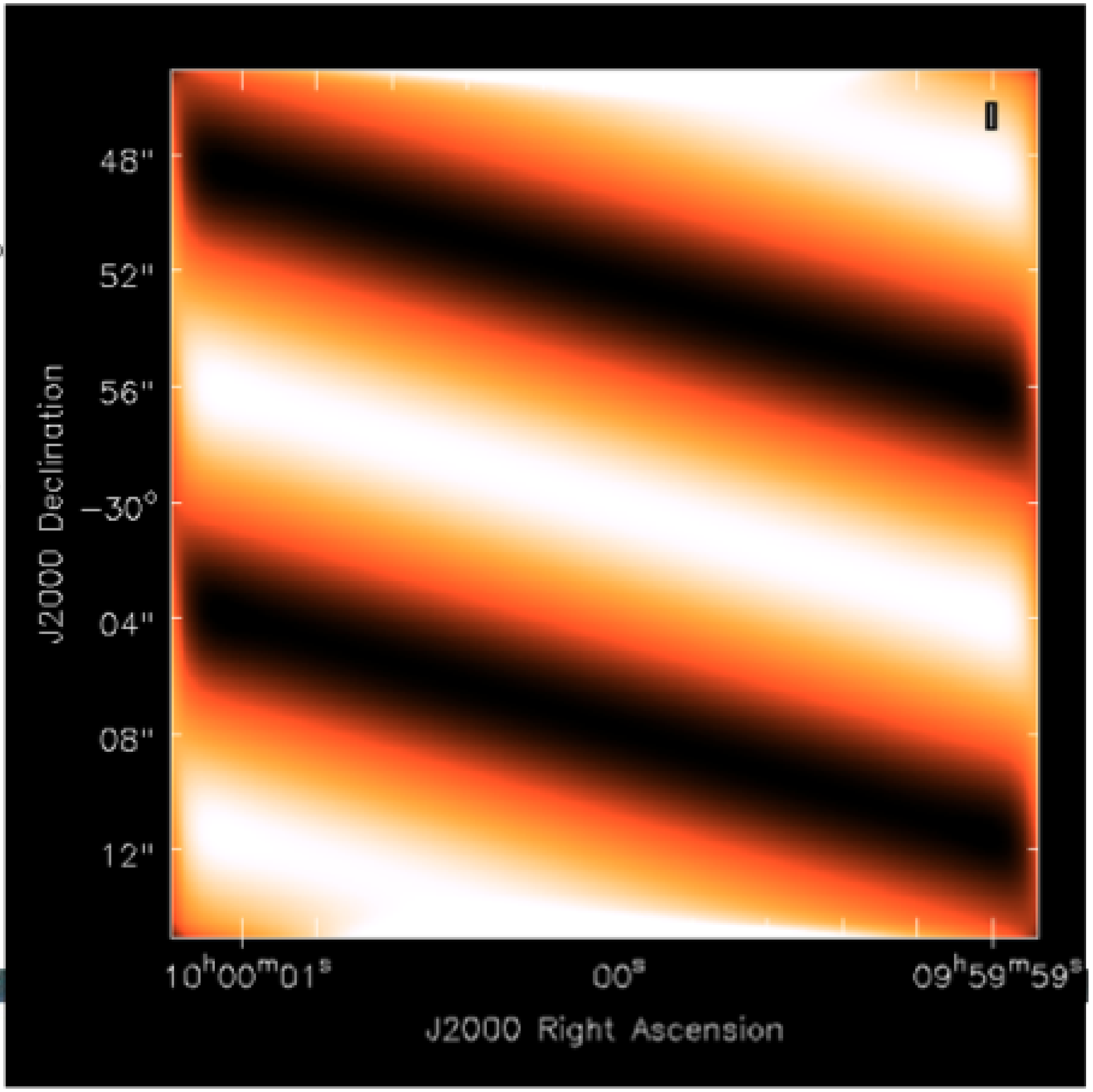
Fourier transform of sampled visibilities yields the true sky brightness convolved with the point spread function

(the "dirty image" is the true image convolved with the "dirty beam")

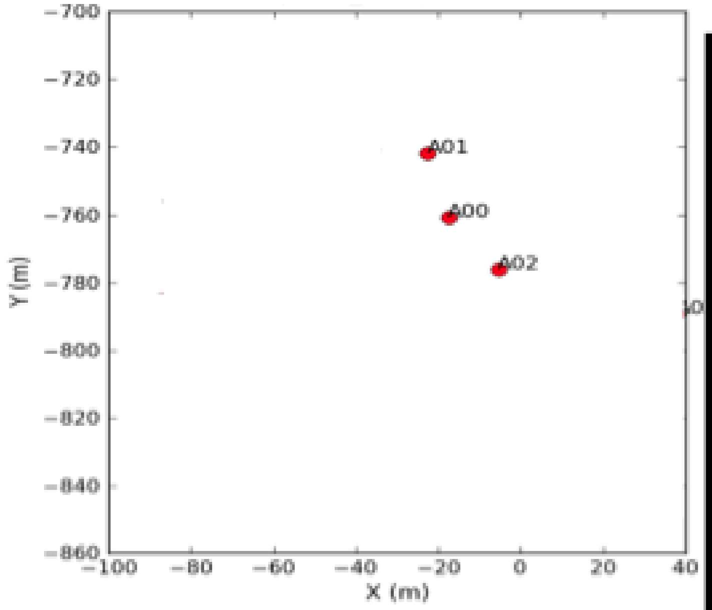
2 antennas

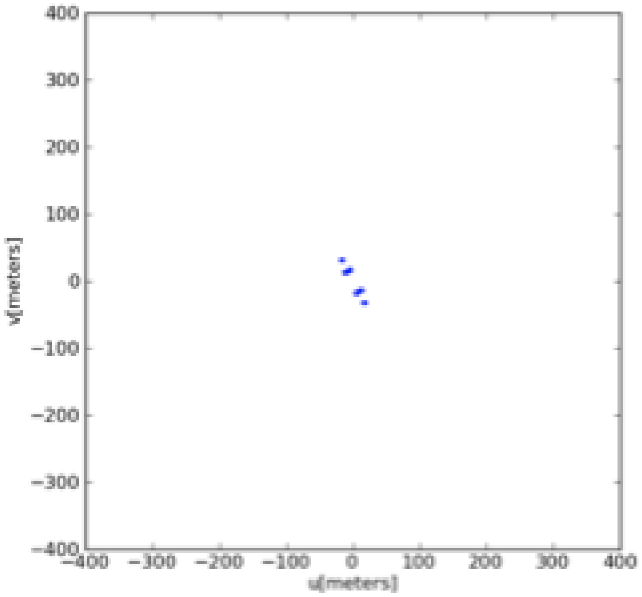

A01
A00
Y (m)
X (m)
−700
−720
−740
−760
−780
−800
−820
−840
−860
−100
−80
−60
−40
−20
0
20
40

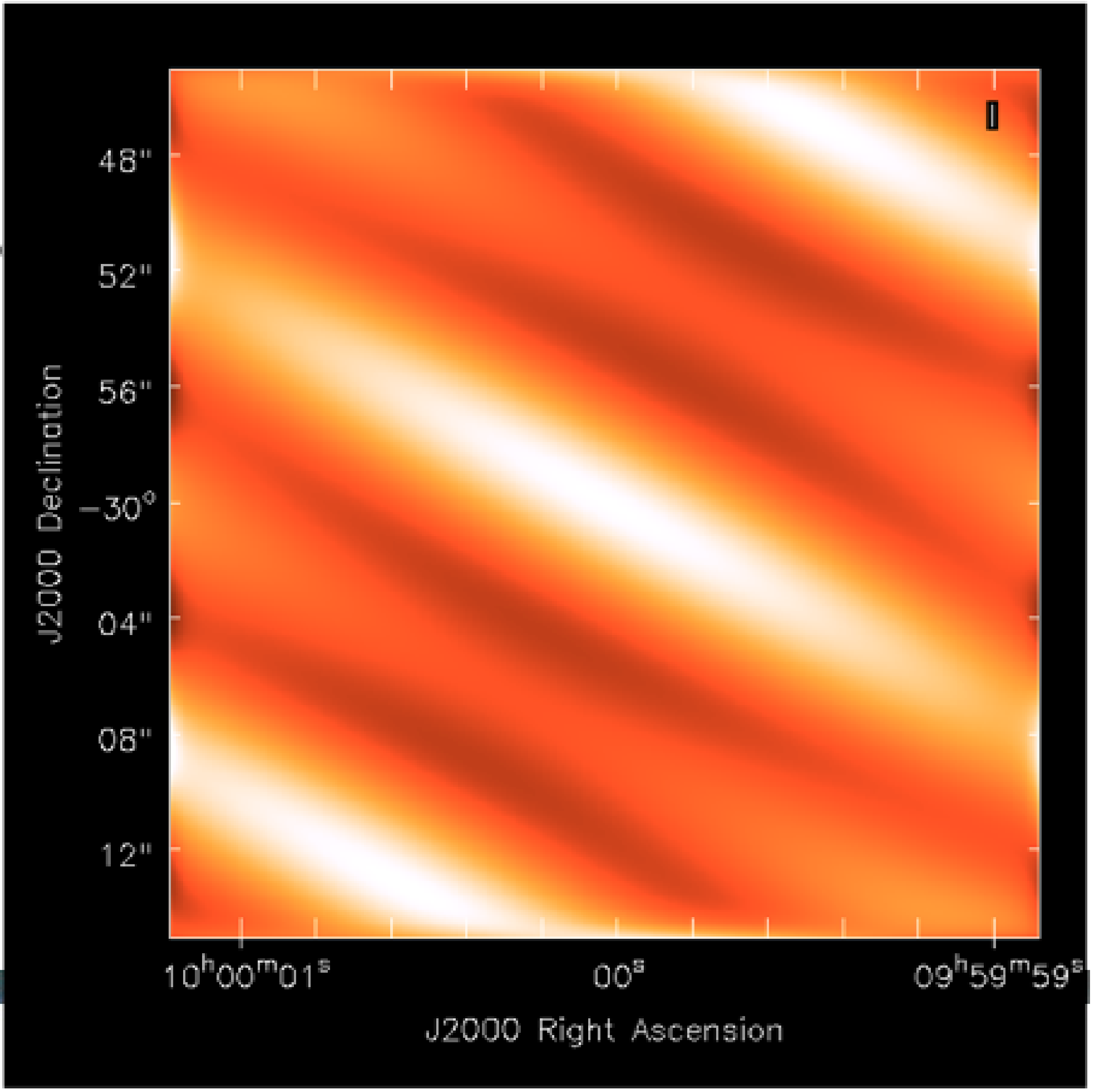

v[meters]
u[meters]
400
300
200
100
0
−100
−200
−300
−400


J2000 Declination
J2000 Right Ascension
48"
52"
56"
−30°
04"
08"
12"
$10^h00^m01^s$
00^s
$09^h59^m59^s$

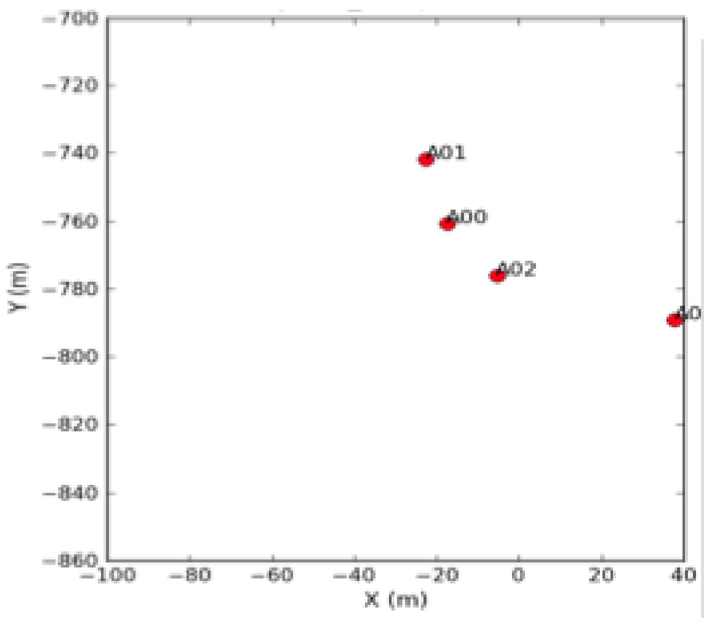
3 antennas

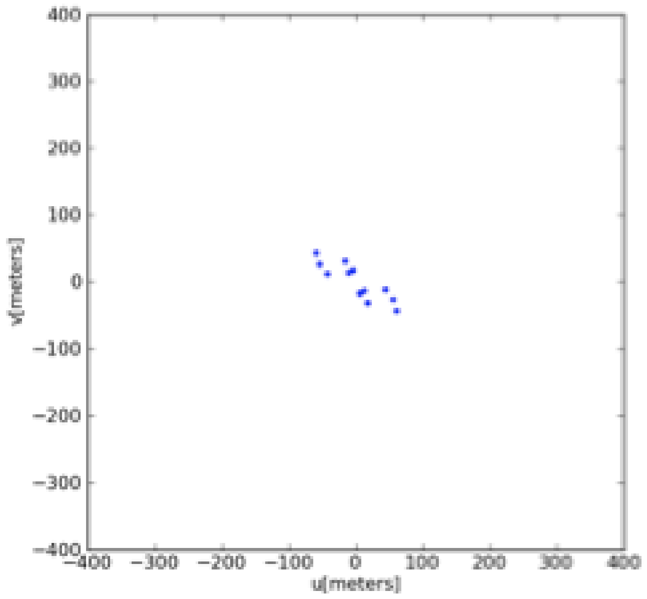

A01
A00
A02
Y (m)
X (m)
−700
−720
−740
−760
−780
−800
−820
−840
−860
−100
−80
−60
−40
−20
0
20
40

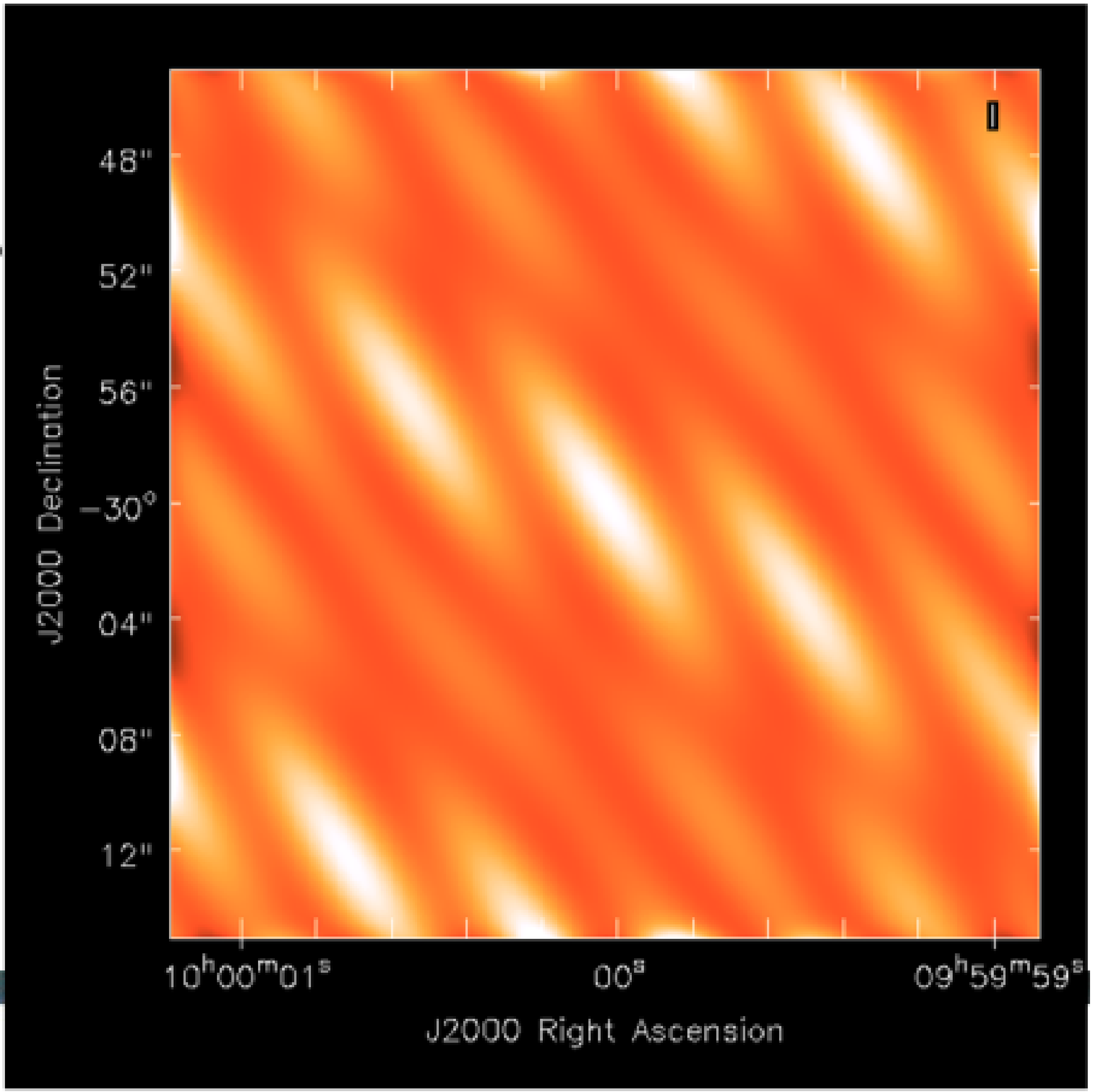

v[meters]
u[meters]
400
300
200
100
0
−100
−200
−300
−400


I
48"
52"
56"
−30°
04"
08"
12"
J2000 Declination
$10^h00^m01^s$
00^s
$09^h59^m59^s$
J2000 Right Ascension

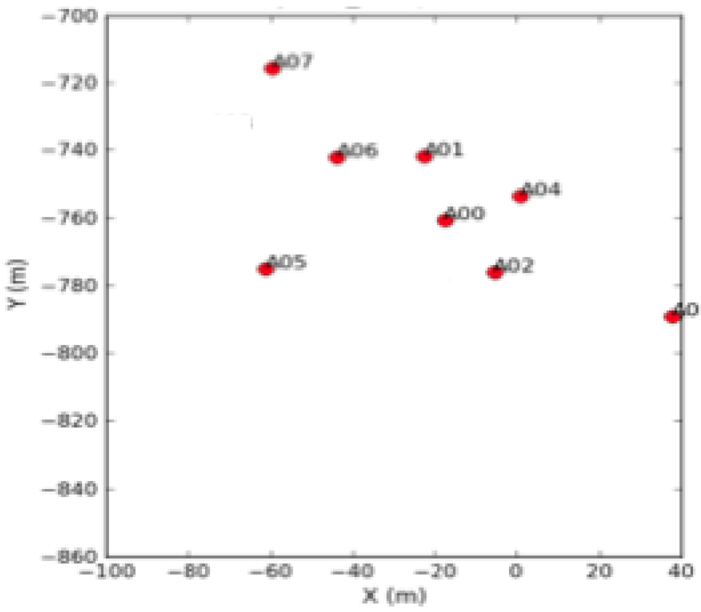
4 antennas


−700
−720
−740
−760
−780
−800
−820
−840
−860
Y (m)
A01
A00
A02
A0
−100
−80
−60
−40
−20
0
20
40
X (m)

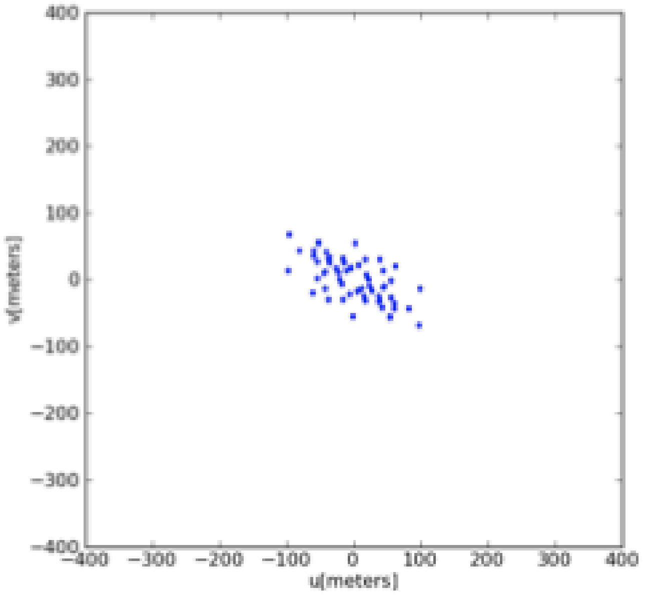

400
300
200
100
0
−100
−200
−300
−400
v[meters]
−400
−300
−200
−100
0
100
200
300
400
u[meters]


48"
52"
56"
−30°
04"
08"
12"
J2000 Declination
10h00m01s
00s
09h59m59s
J2000 Right Ascension

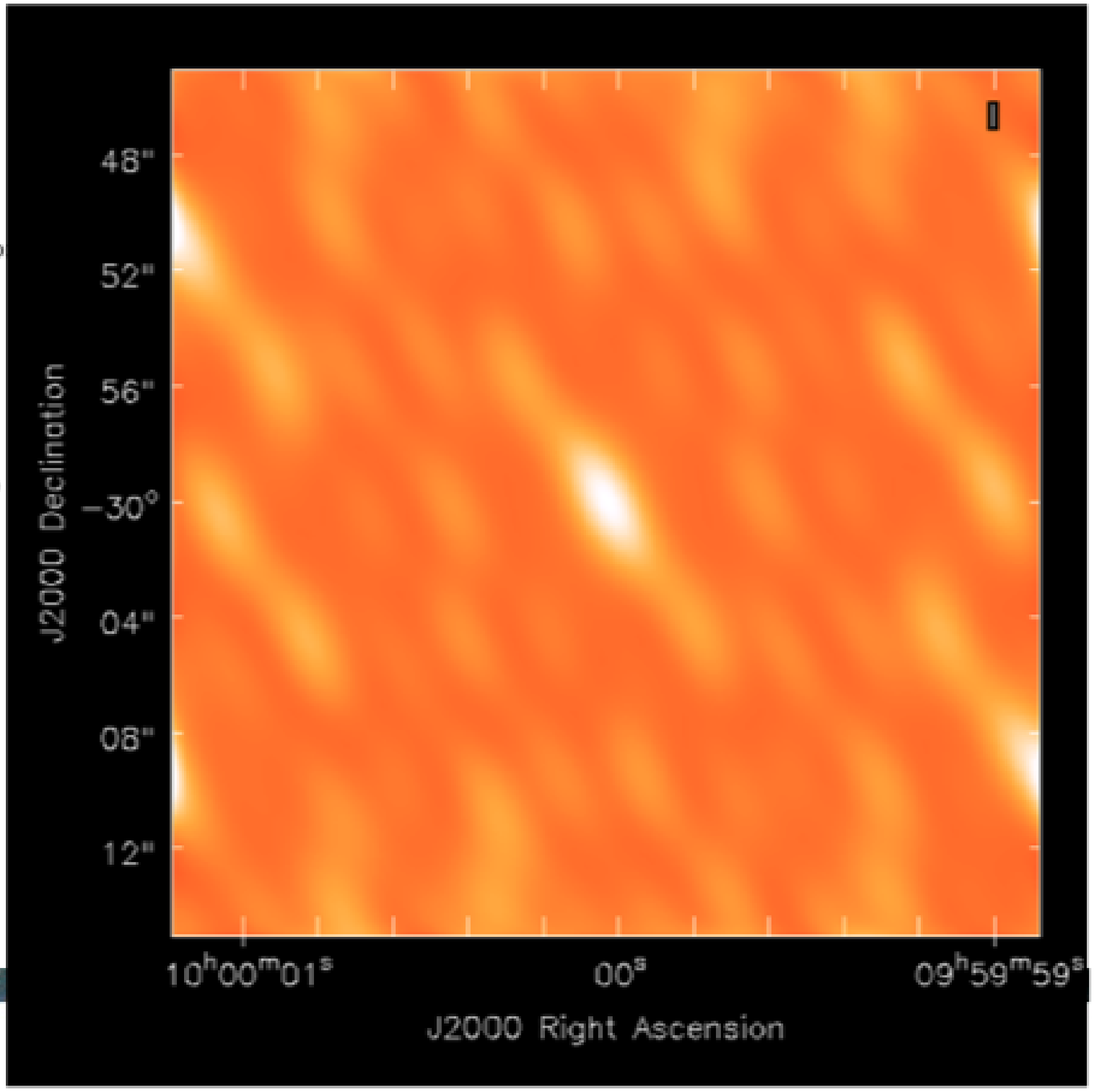
8 antennas



407
406
401
404
400
405
402
40
Y (m)
−700
−720
−740
−760
−780
−800
−820
−840
−860
−100
−80
−60
−40
−20
0
20
40
X (m)

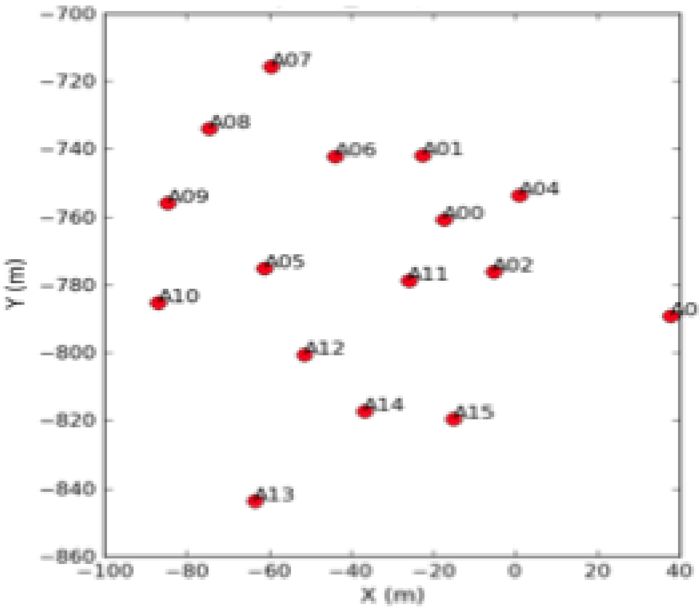


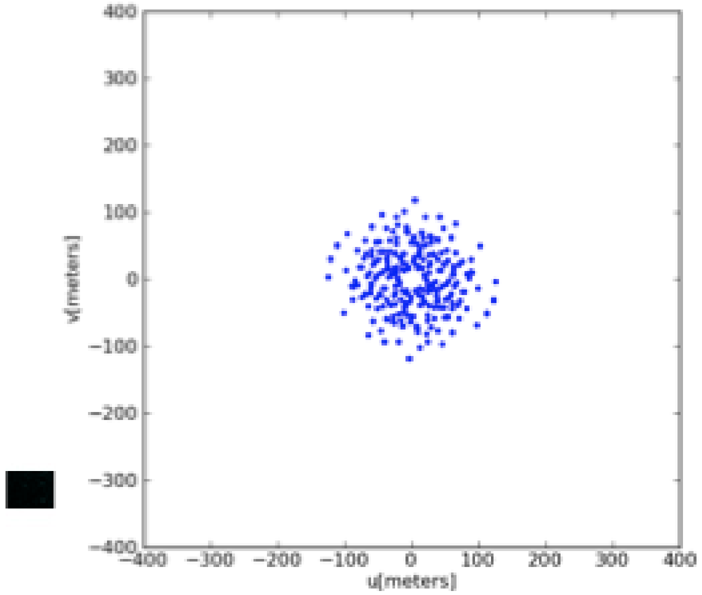
v[meters]
400
300
200
100
0
−100
−200
−300
−400
−400
−300
−200
−100
0
100
200
300
400
u[meters]

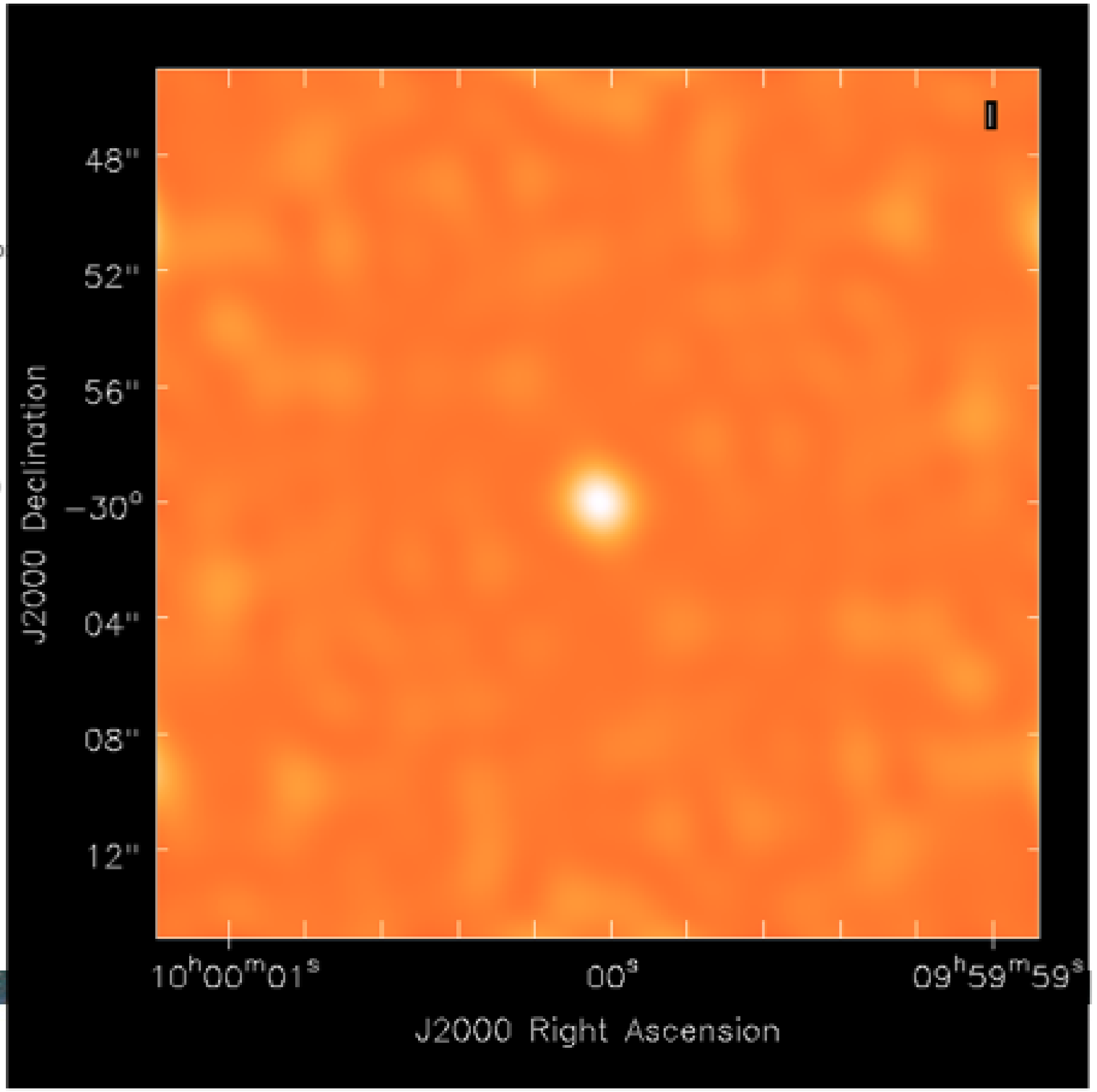


J2000 Declination
48"
52"
56"
−30°
04"
08"
12"
$10^h00^m01^s$
00^s
$09^h59^m59^s$
J2000 Right Ascension

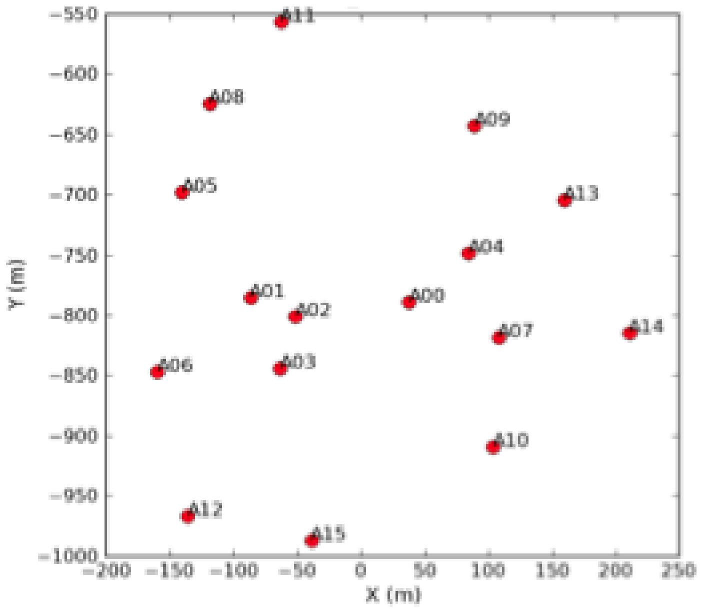
16 antennas - compact

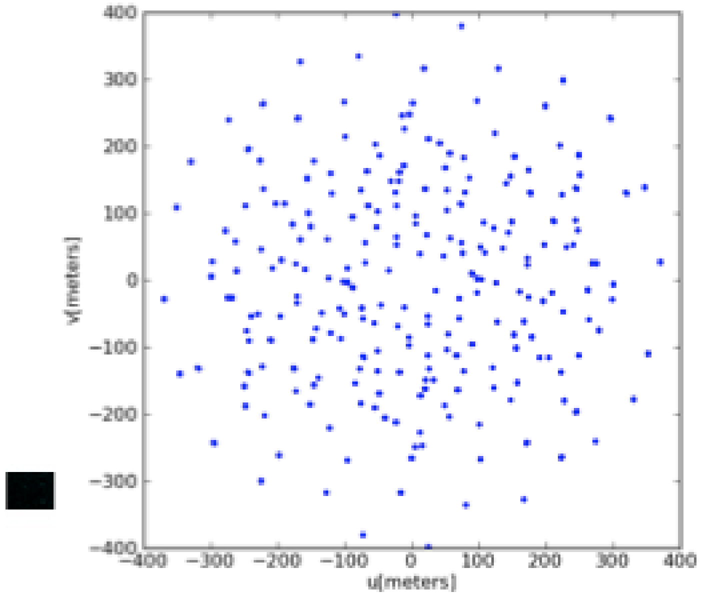


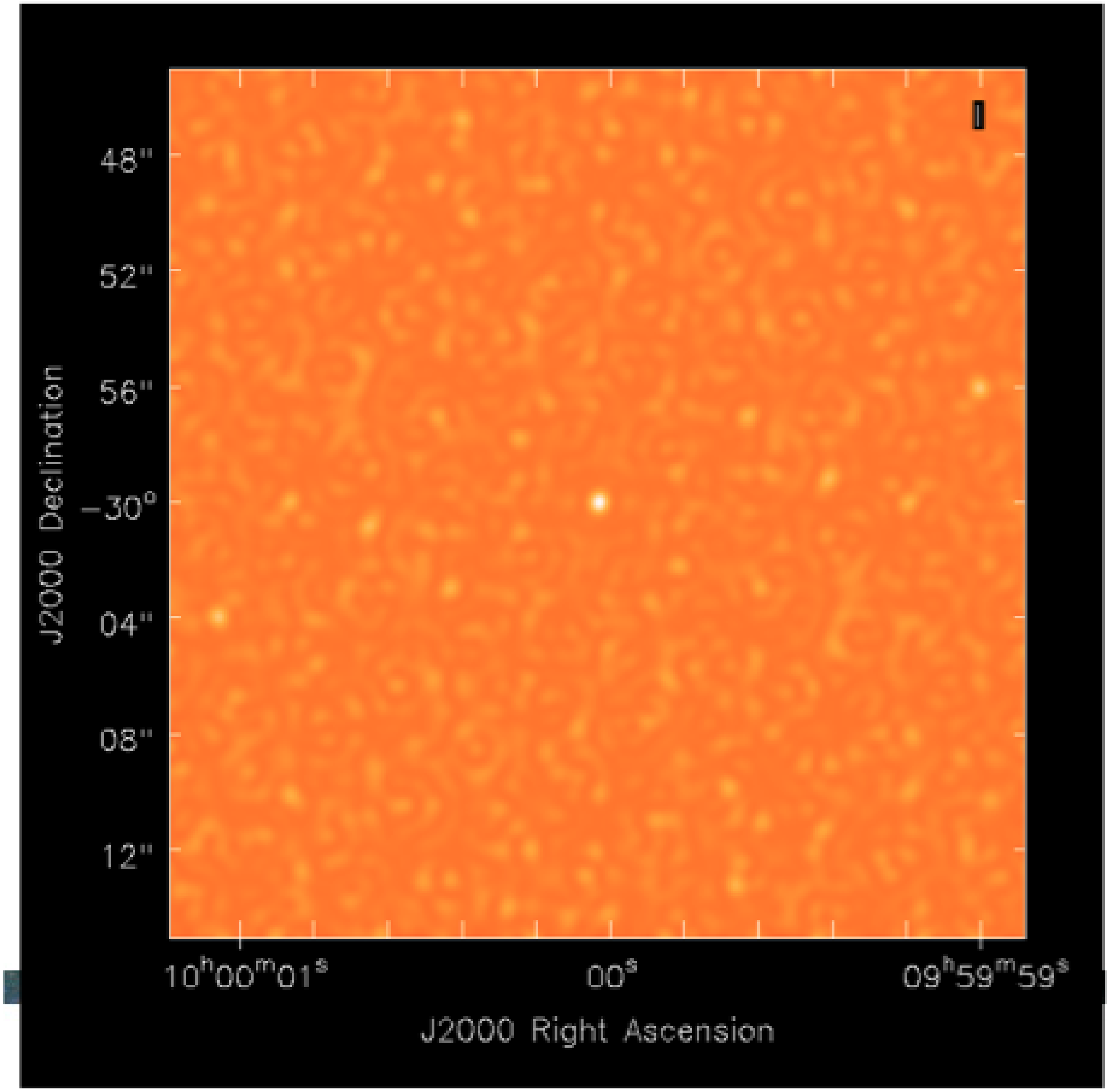




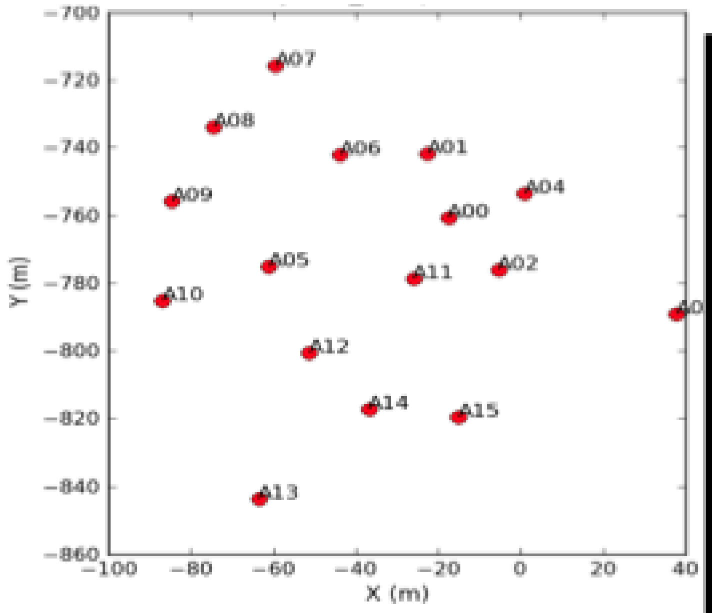
16 antennas - extended

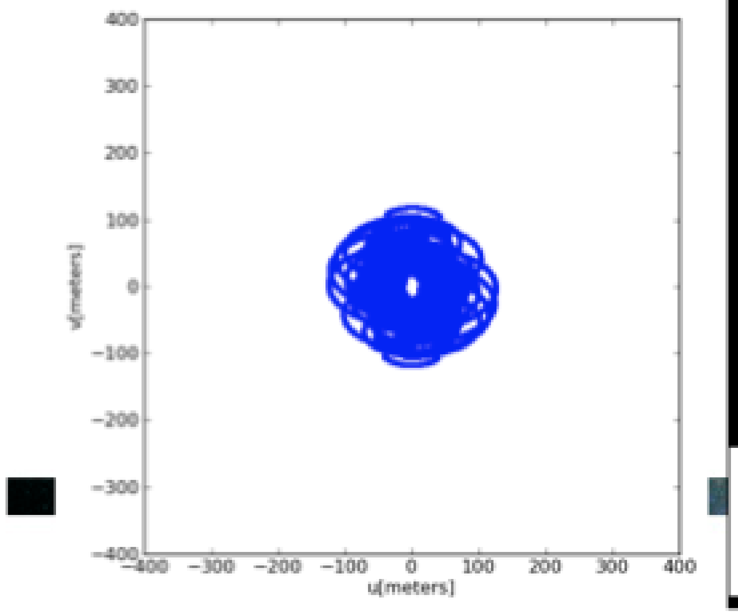


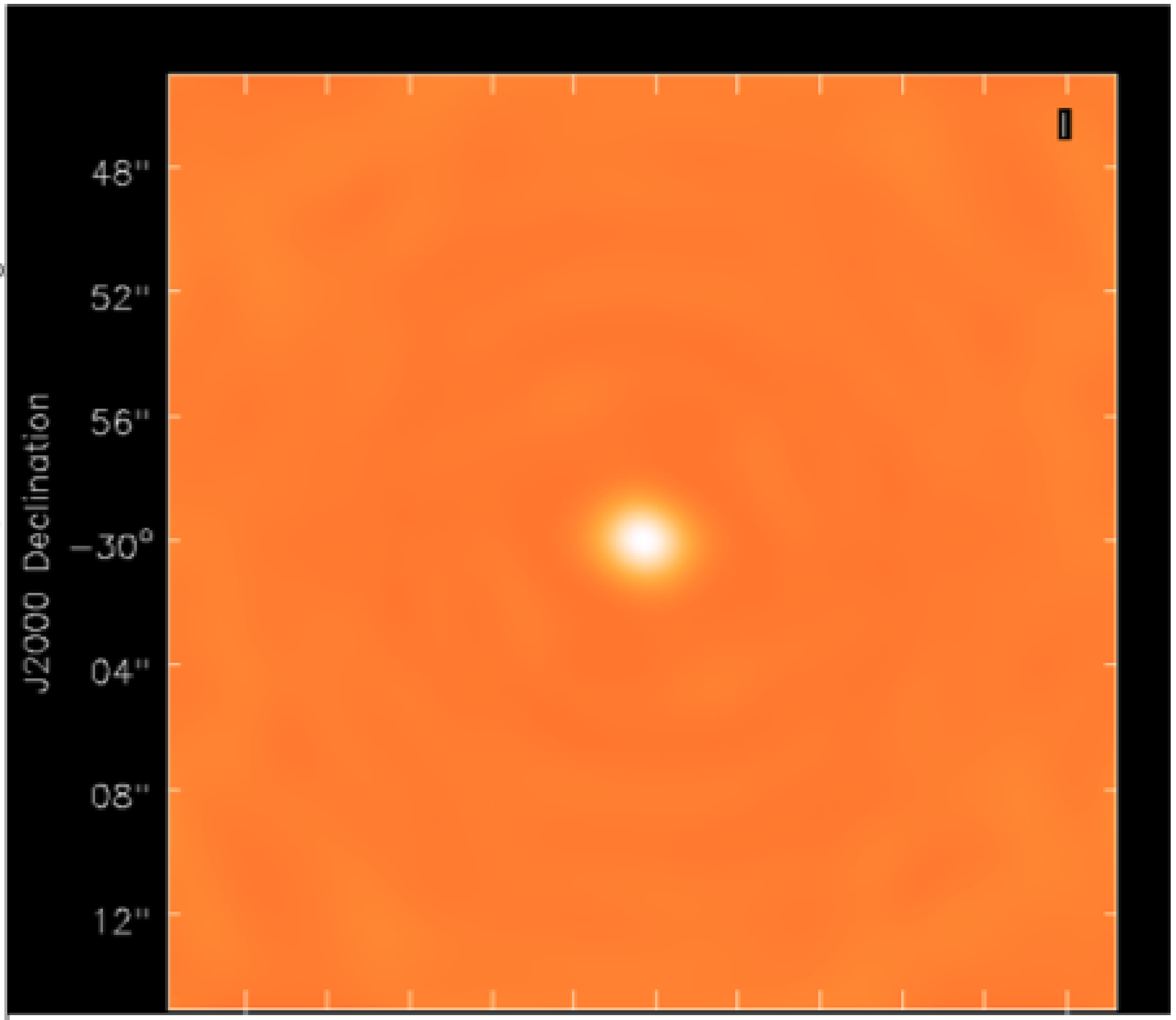




16 antennas - compact - 8 hours



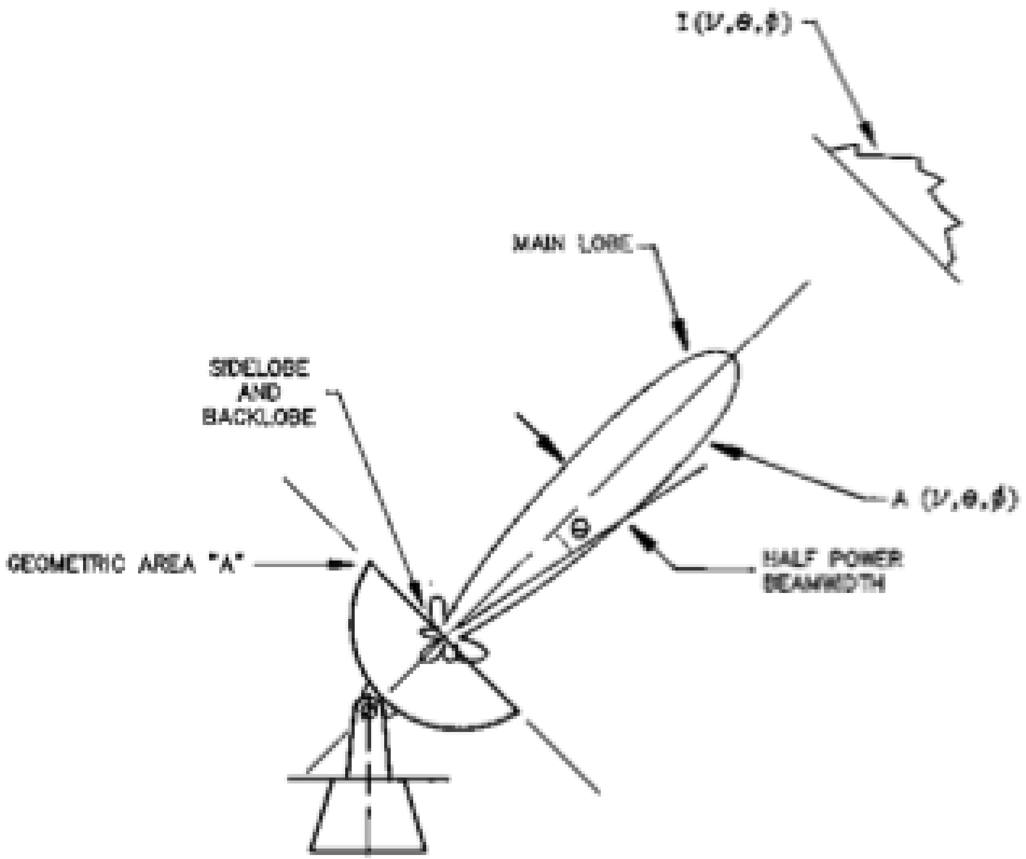




- Sky response = PSF = dirty beam = Fourier transform of uv distribution

(imperfect) reconstruction of the sky

- Incomplete sampling of uv plane: sidelobes
- non-point-like instrumental response ***A*** a.k.a finite primary beam



$$V(u,v) = \mathbf{A} \otimes FT\{\, T(x,y) \,\}$$

(imperfect) reconstruction of the sky

- Incomplete sampling of uv plane: sidelobes
- non-point-like instrumental response ***A***
- noise

$$V(u,v) = \mathbf{A} \otimes FT\{\ T(x,y)\ \} + n$$

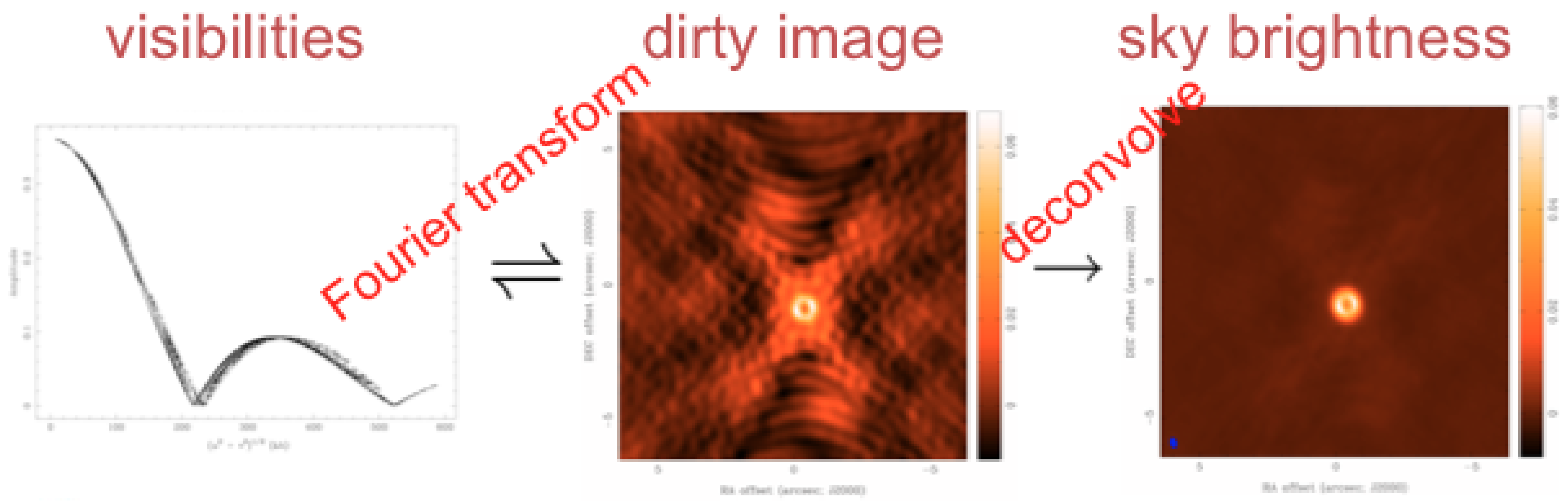
- Need to invert, but
  - FFT requires information on a regular grid in uv plane

$$V^G(u,v) = V(u,v)B(u,v) \otimes G(u,v) \rightleftharpoons T^D(x,y)g(x,y)$$

  - Ideal: gridding would invert instrument response ***A*** = FT of primary beam, but its not invertible.
  - Use spheroidal function for single-field imaging, use PB for mosaic imaging
    * spheroidal are generally more tapered, less aliasing

How to analyze (imperfect) interferometer data?

- uv plane analysis
  - best for “simple” sources, e.g. point sources, disks
- image plane analysis
  - dirty image $T^D(x,y)$ = Fourier transform { V(u,v) }
  - deconvolve b(x,y) from $T^D(x,y)$ to determine (model of) T(x,y)



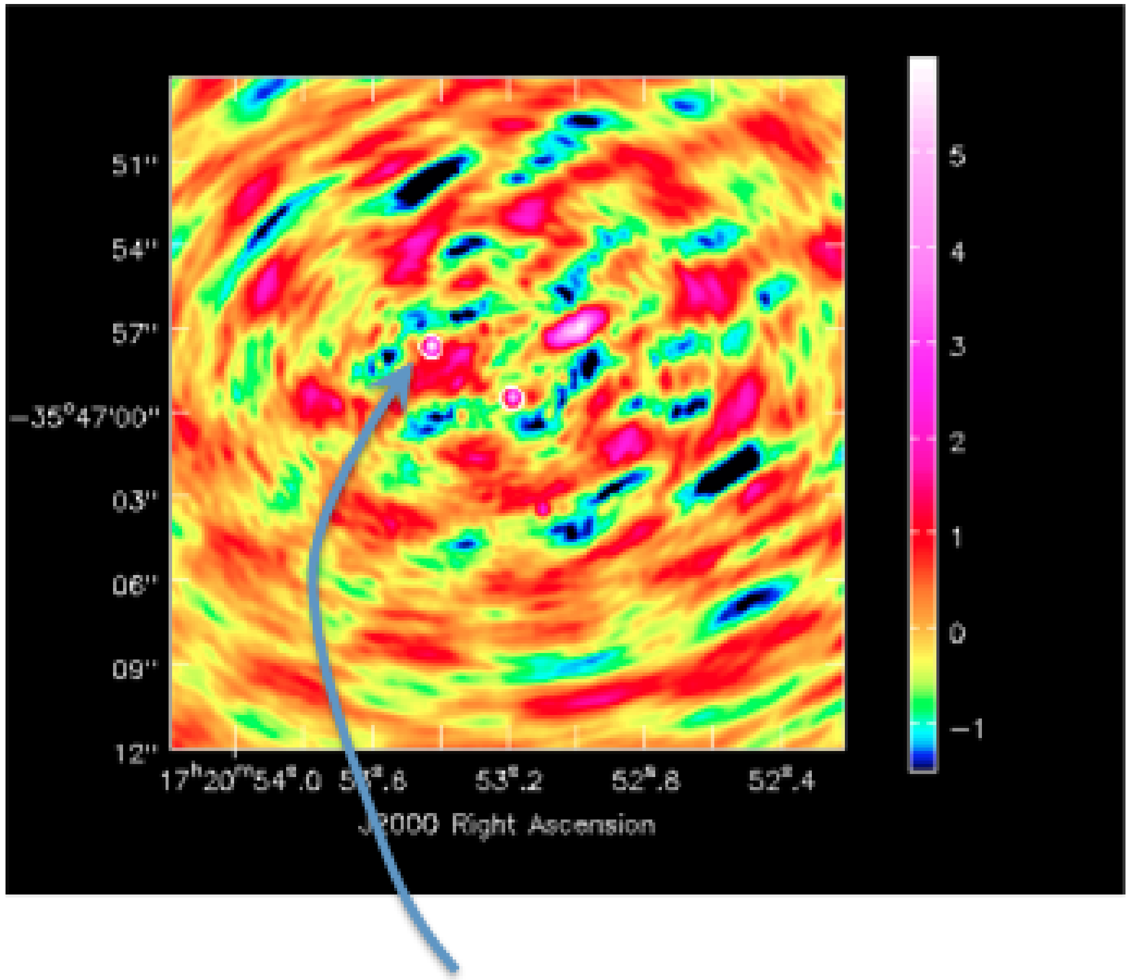
visibilities

Fourier transform ⇌

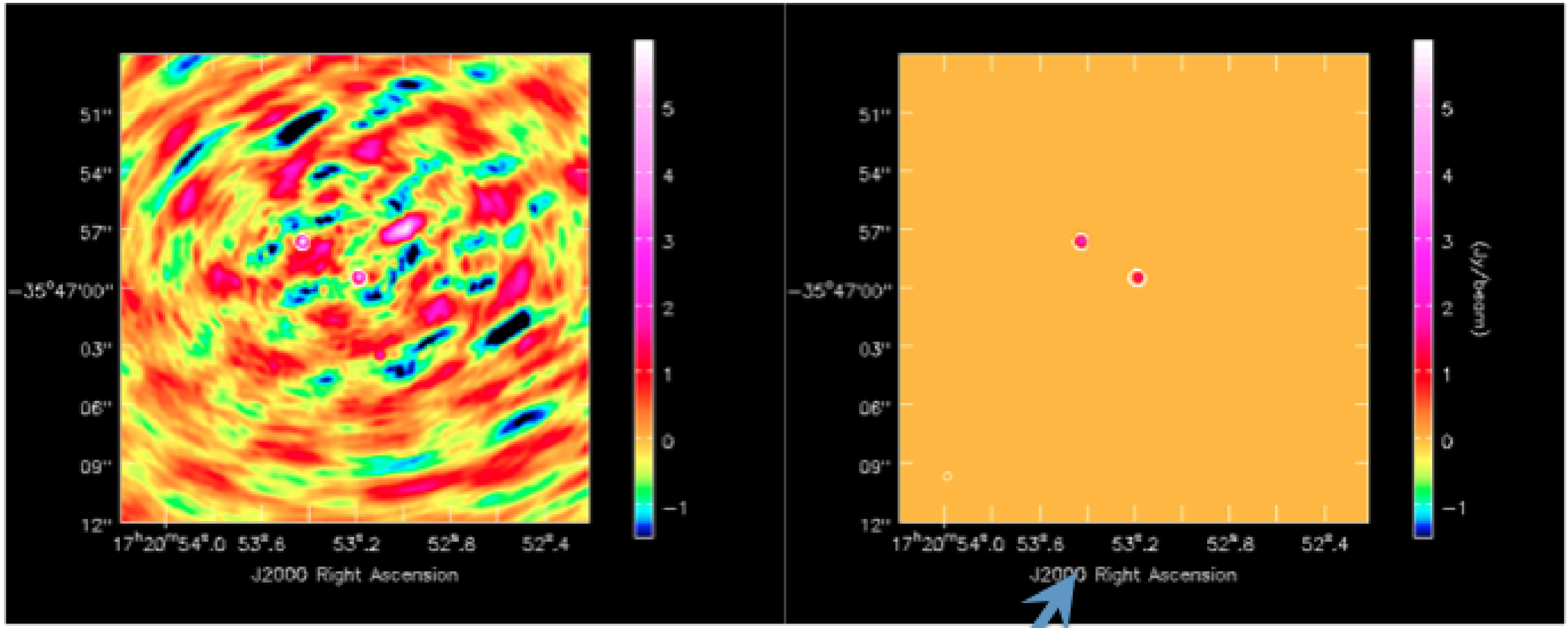
dirty image

deconvolve →

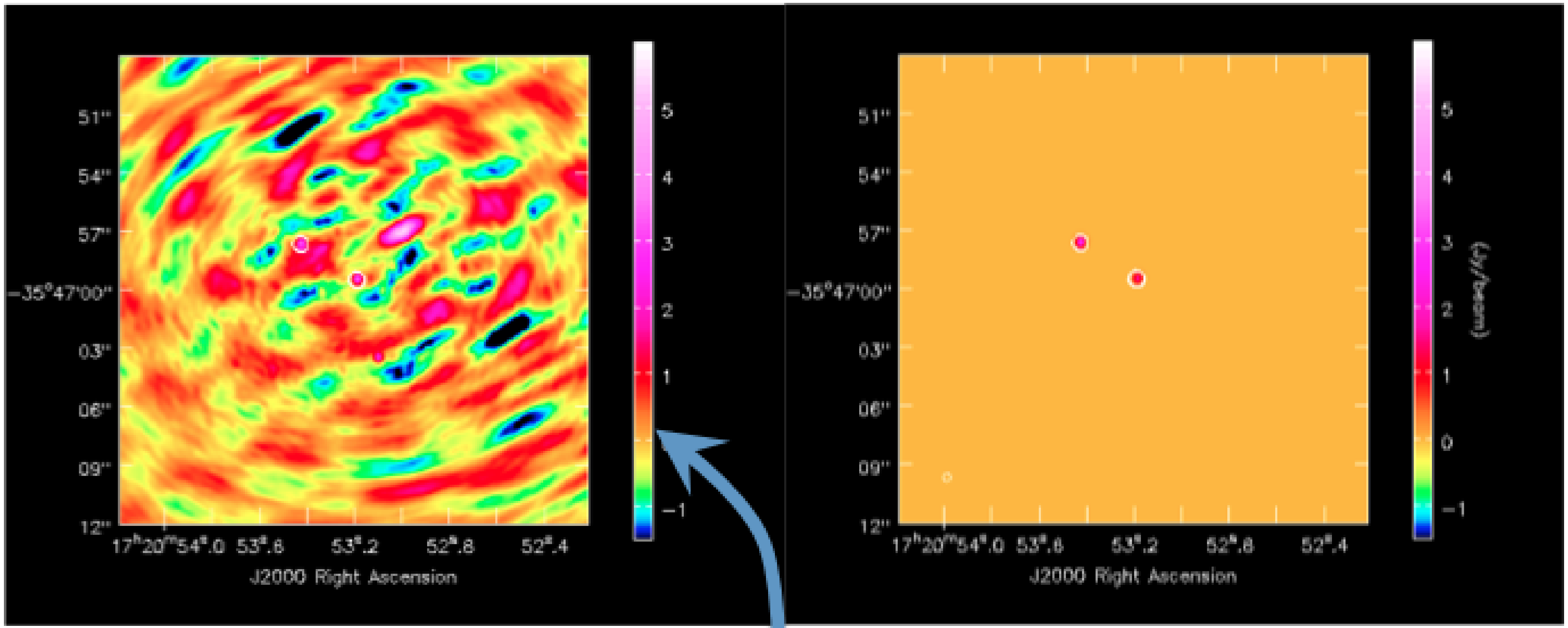
sky brightness



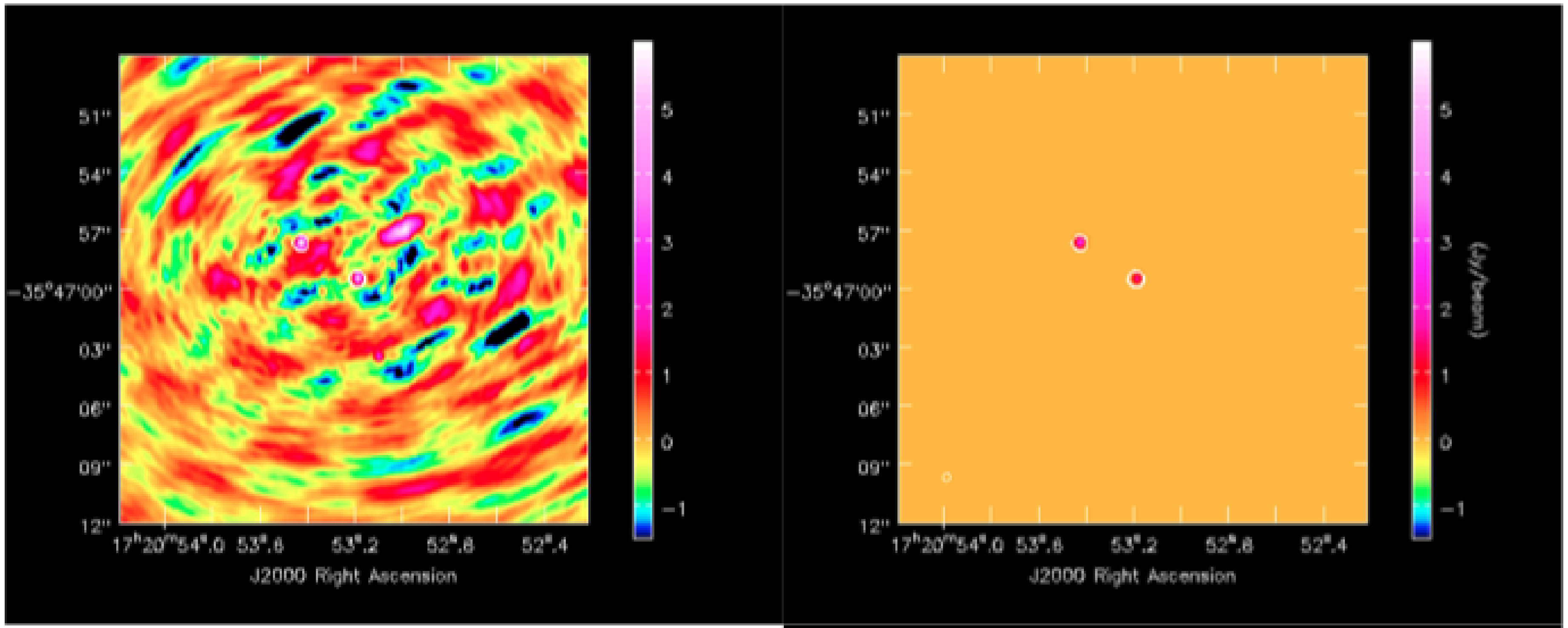
- Find brightest points in dirty image



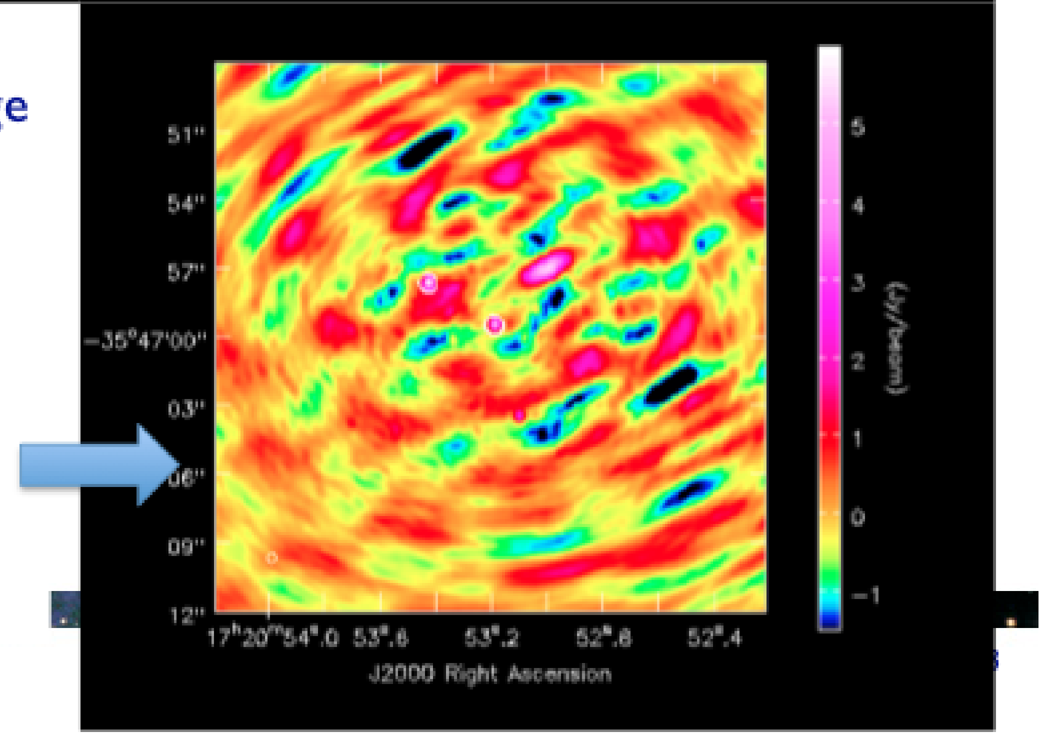
- Find brightest points in dirty image
- Create model image containing a fraction of those flux points

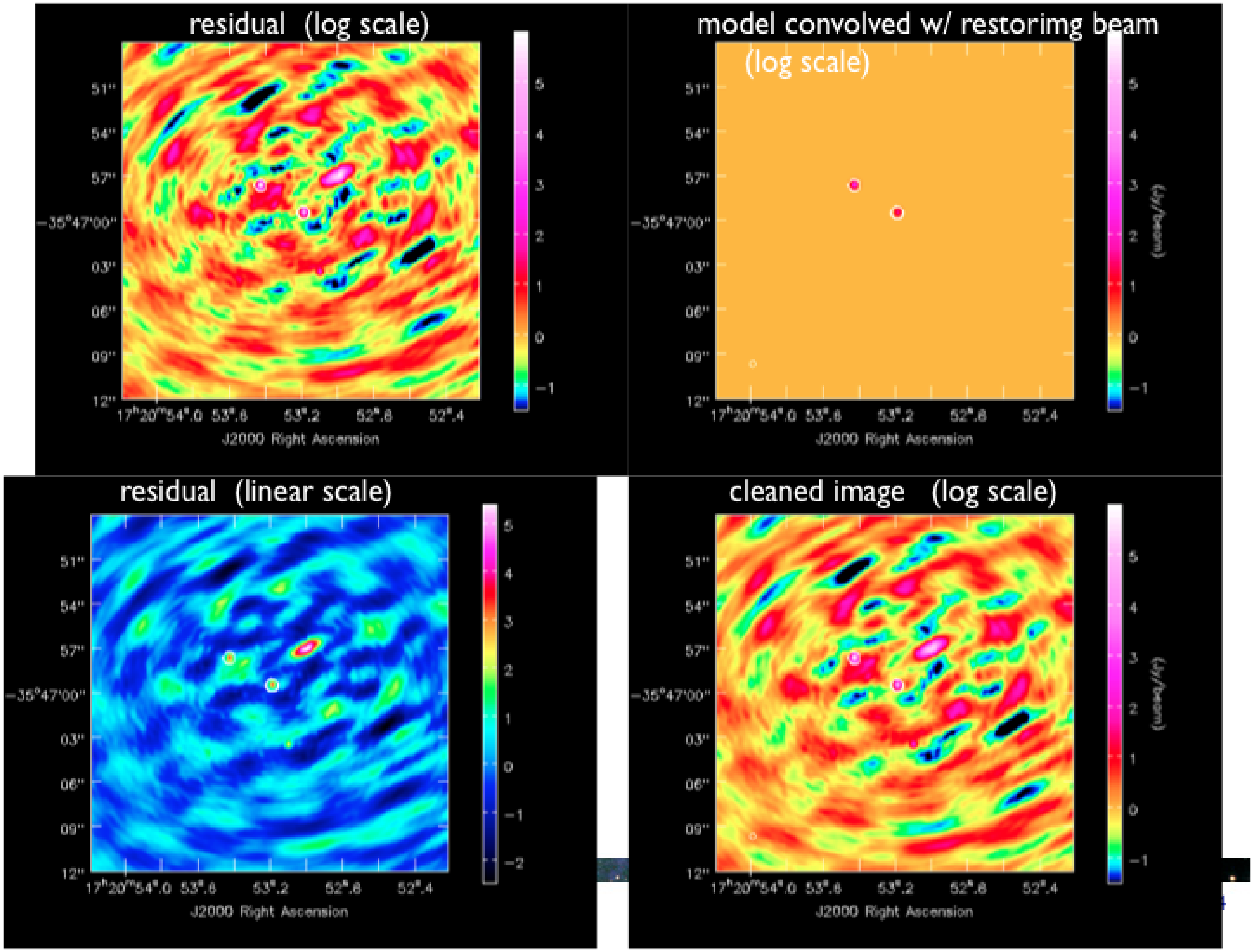


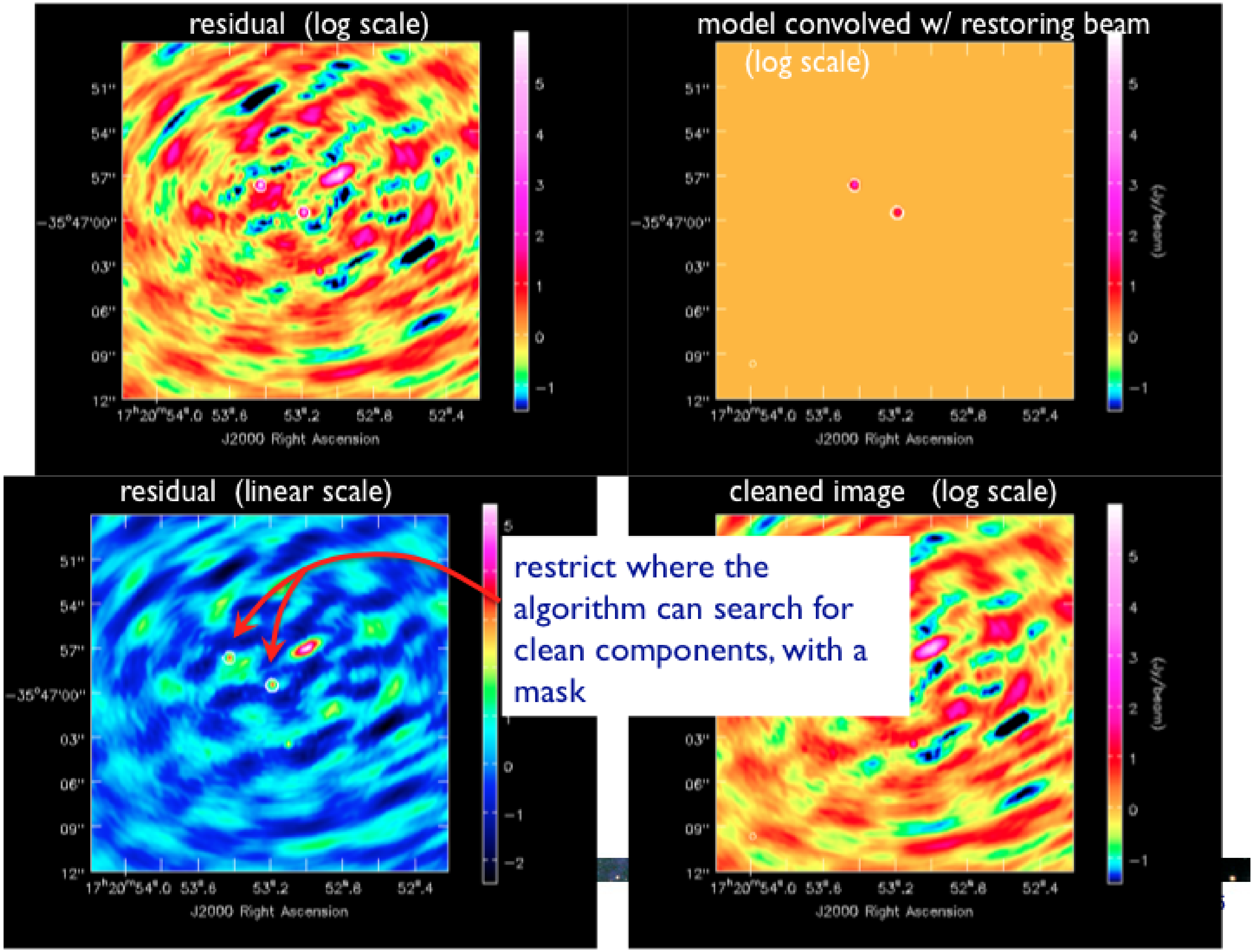
- Find brightest points in dirty image
- Create model image containing a fraction of those flux points
- Subtract model from data, leaving a residual



- Find brightest points in dirty image
- Create model image containing a fraction of those flux points
- Subtract model from data, leaving a residual
- Final product = residual + model (convolved with restoring Gaussian beam)




residual (log scale)
model convolved w/ restorimg beam
(log scale)
residual (linear scale)
cleaned image (log scale)
51"
54"
57"
−35°47'00"
03"
06"
09"
12"
17h20m54s.0 53s.6 53s.2 52s.8 52s.4
J2000 Right Ascension
(Jy/beam)
5
4
3
2
1
0
−1
−2



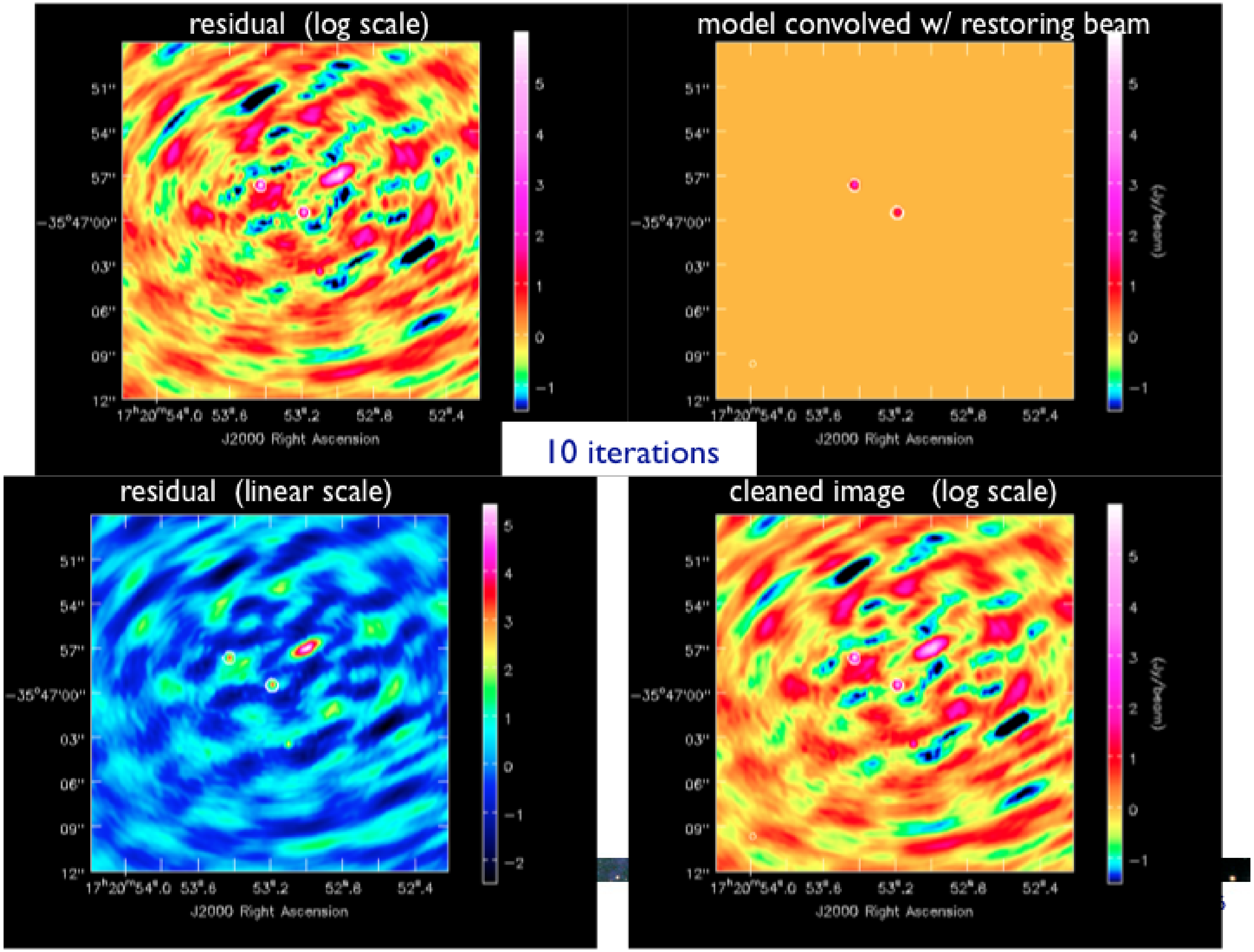
residual (log scale)

model convolved w/ restoring beam (log scale)

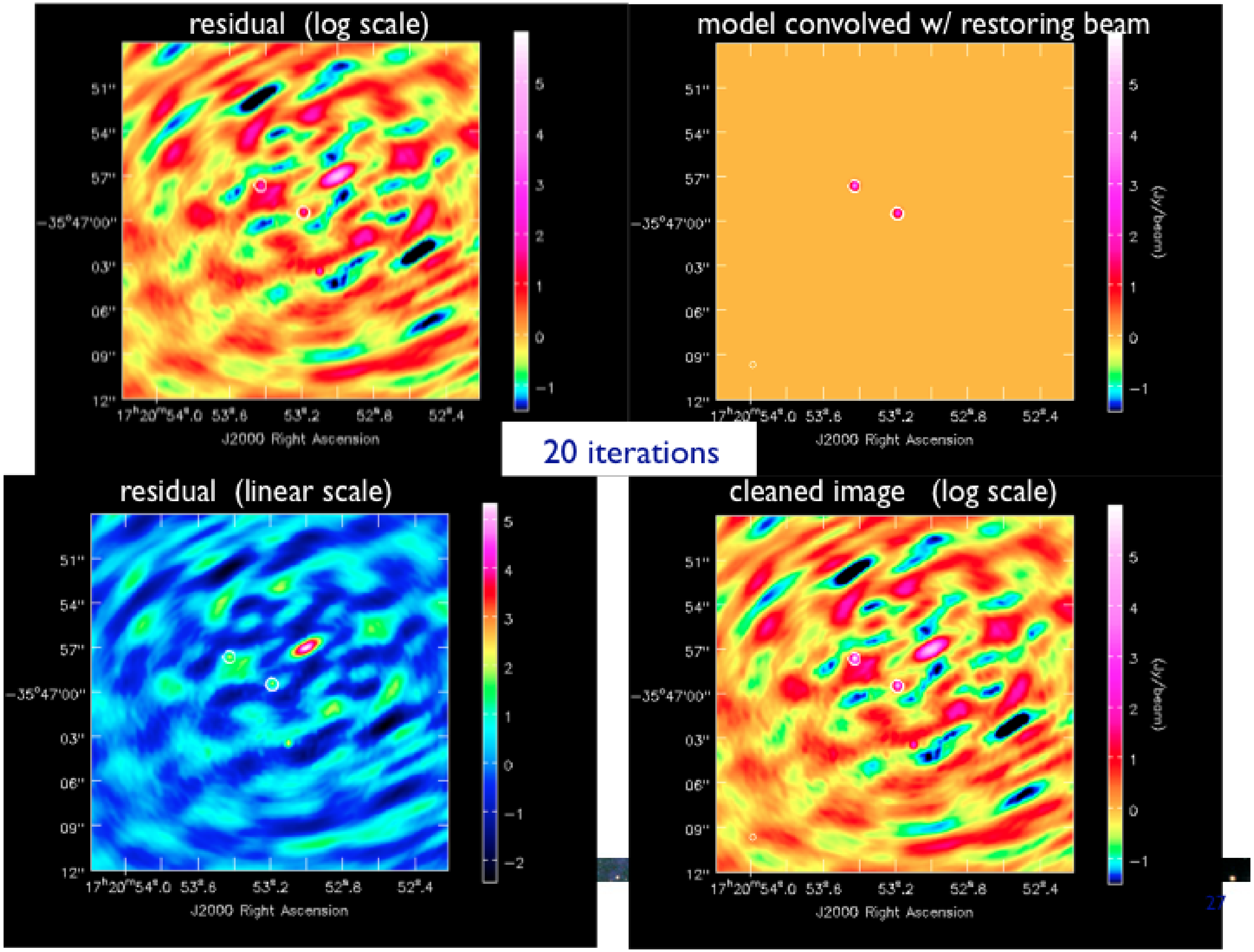
residual (linear scale)

cleaned image (log scale)

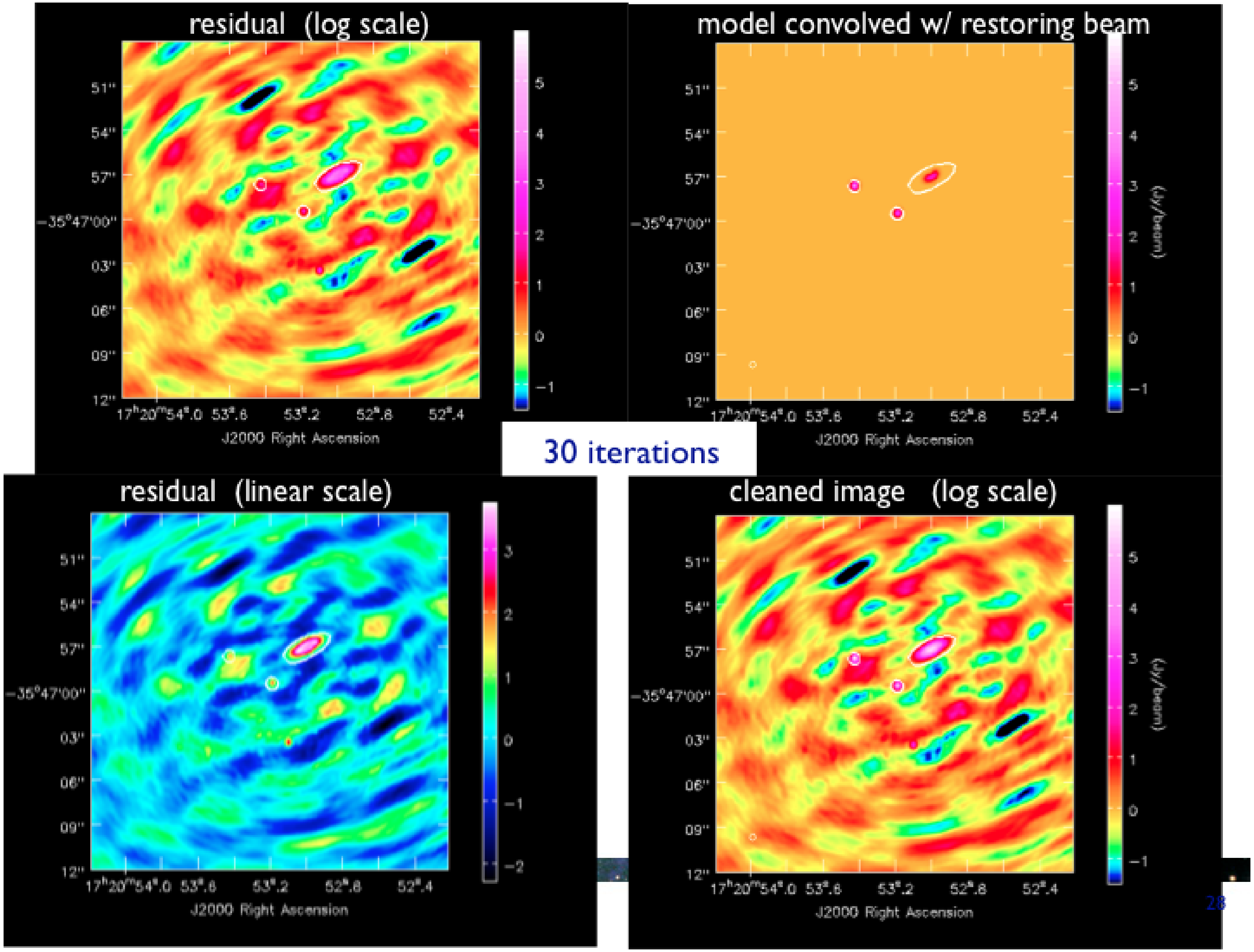
restrict where the algorithm can search for clean components, with a mask



10 iterations



20 iterations

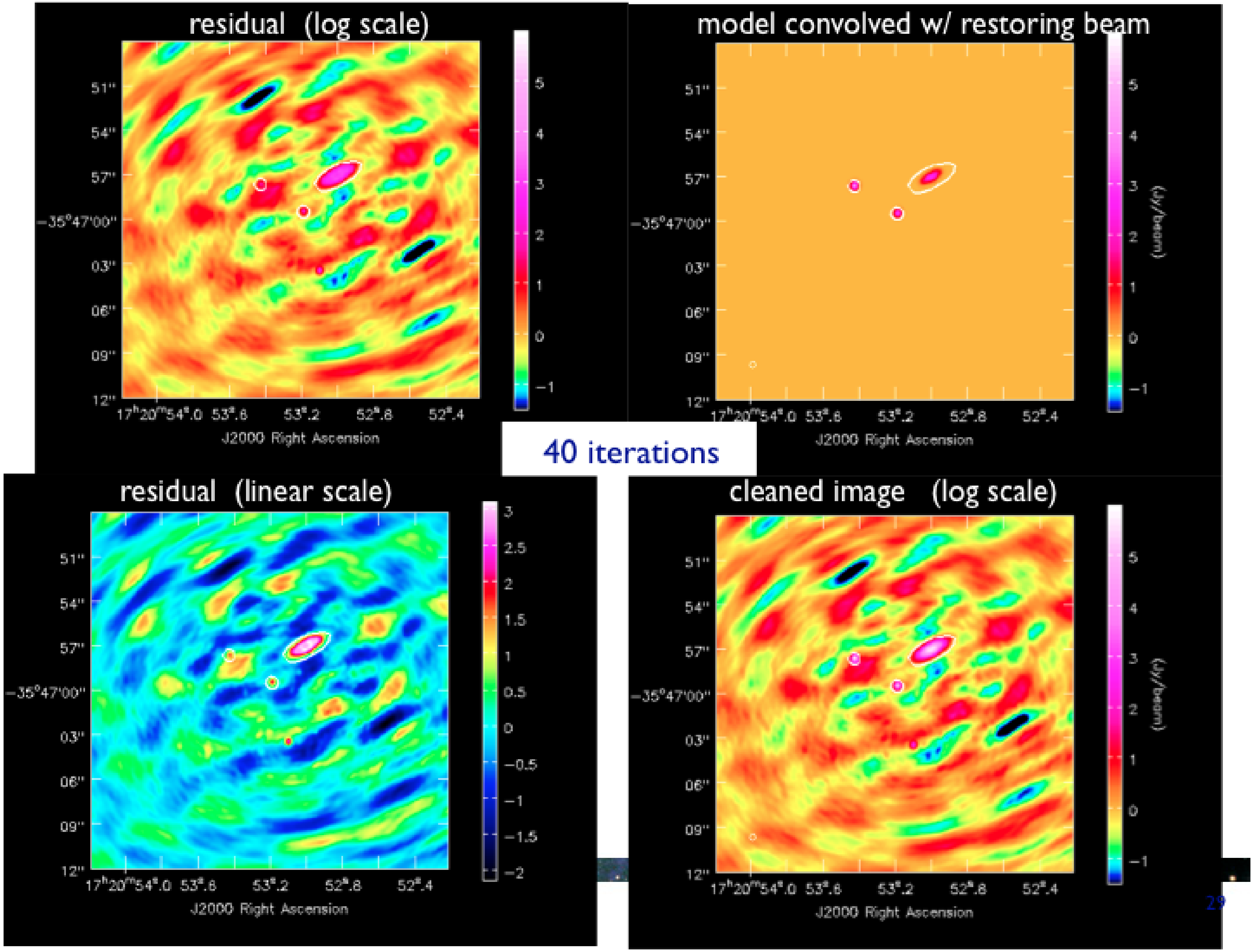

residual (log scale)

model convolved w/ restoring beam

30 iterations

residual (linear scale)

cleaned image (log scale)



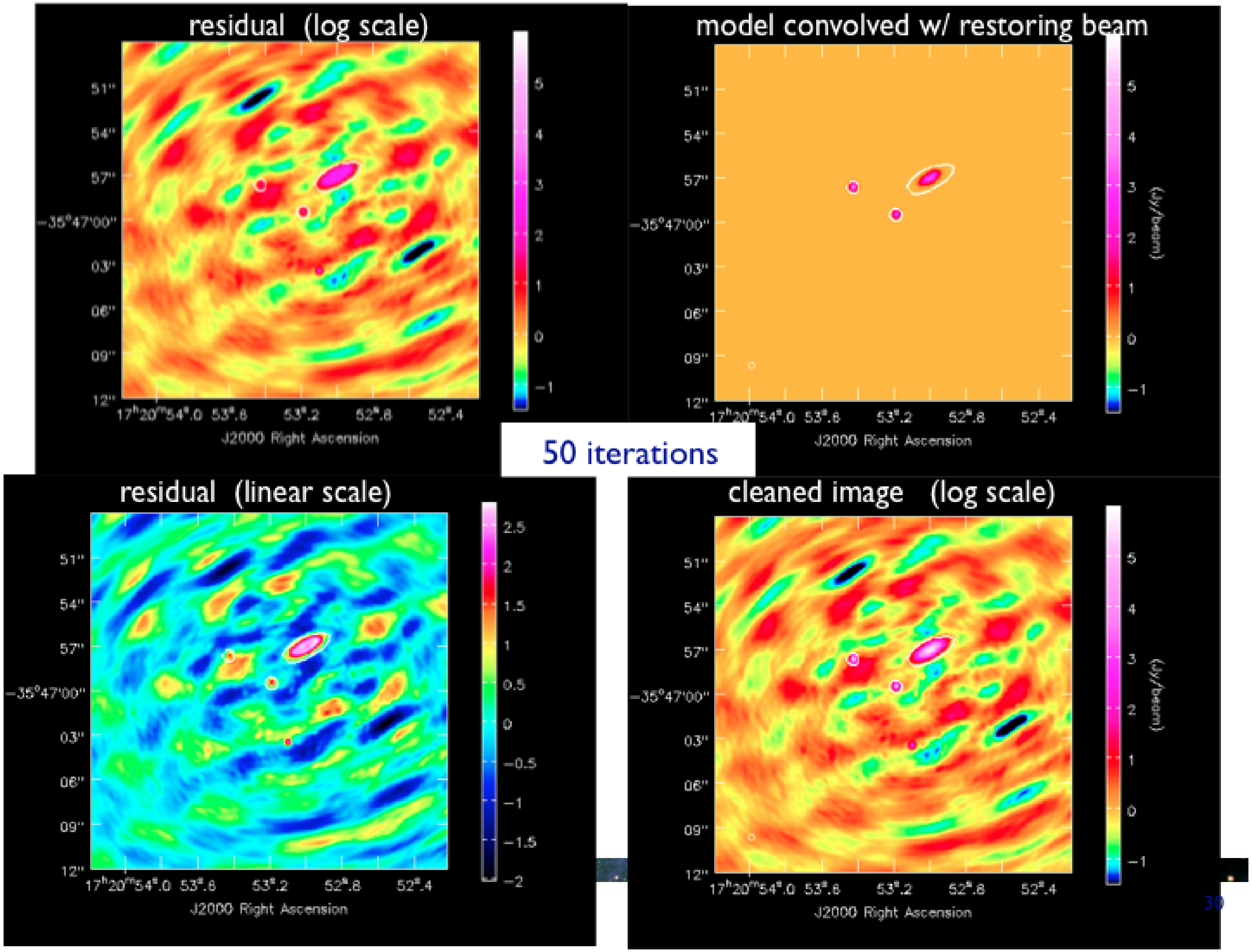
residual (log scale)

model convolved w/ restoring beam

40 iterations

residual (linear scale)

cleaned image (log scale)



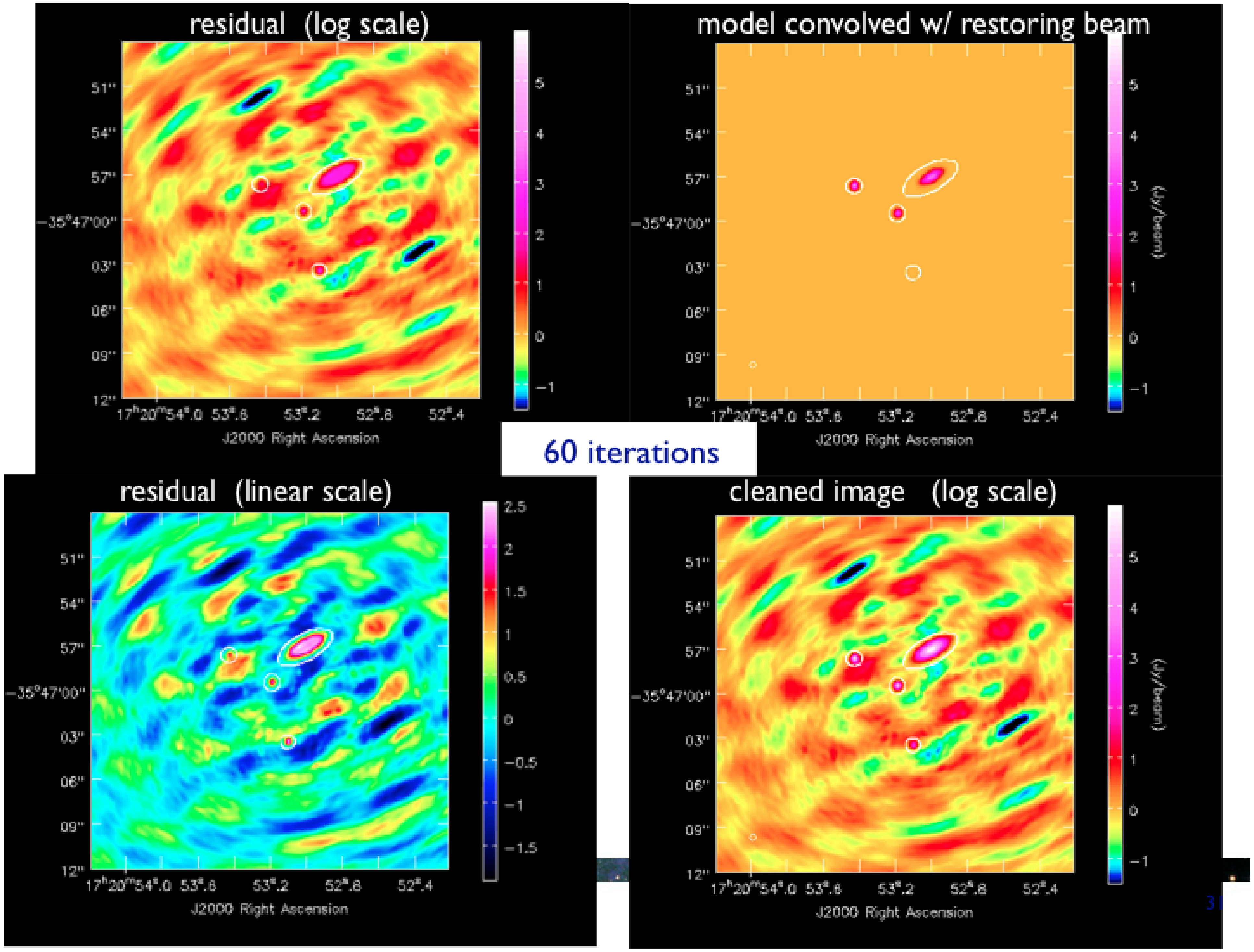
residual (log scale)

model convolved w/ restoring beam

50 iterations

residual (linear scale)

cleaned image (log scale)



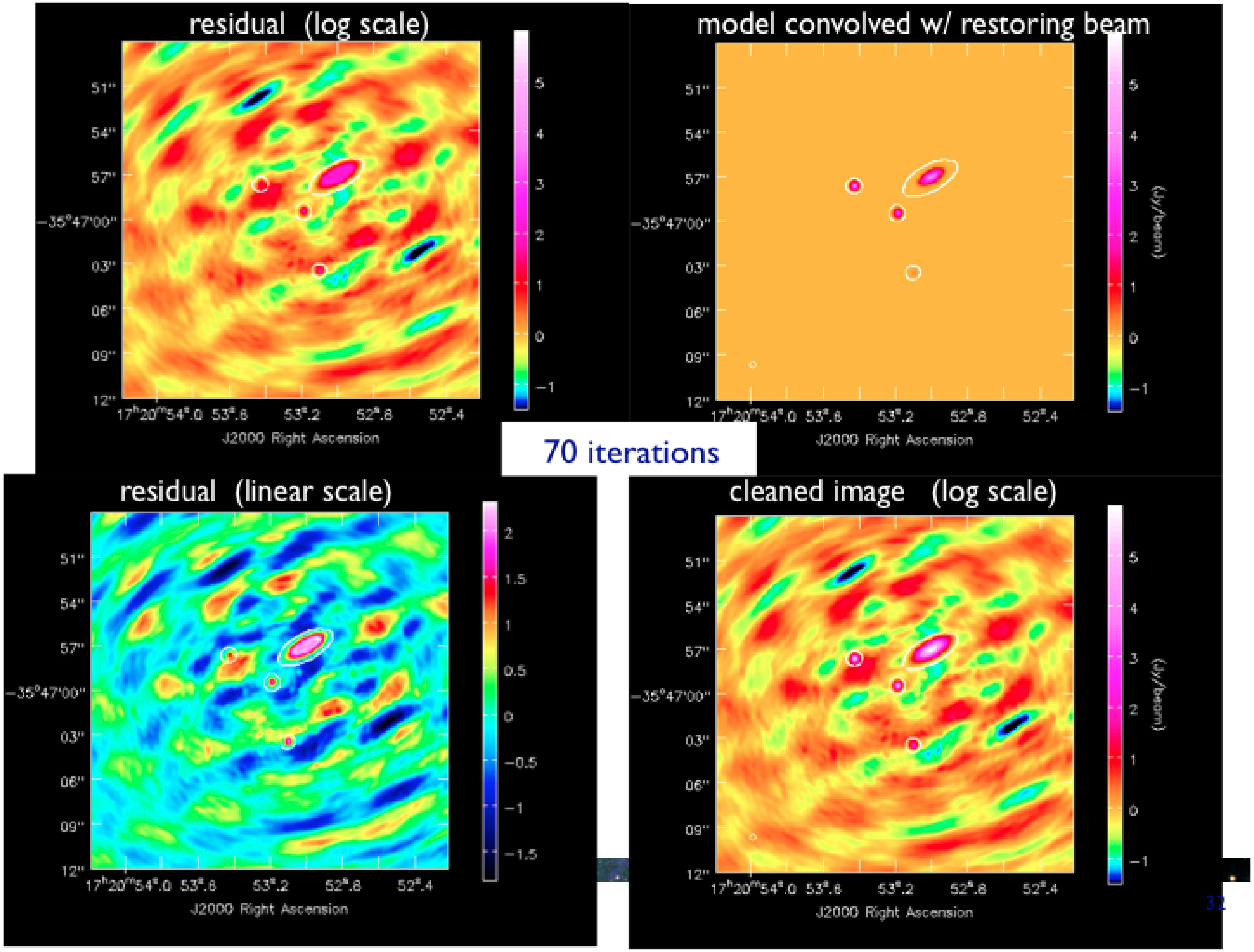
residual (log scale)

model convolved w/ restoring beam

60 iterations

residual (linear scale)

cleaned image (log scale)



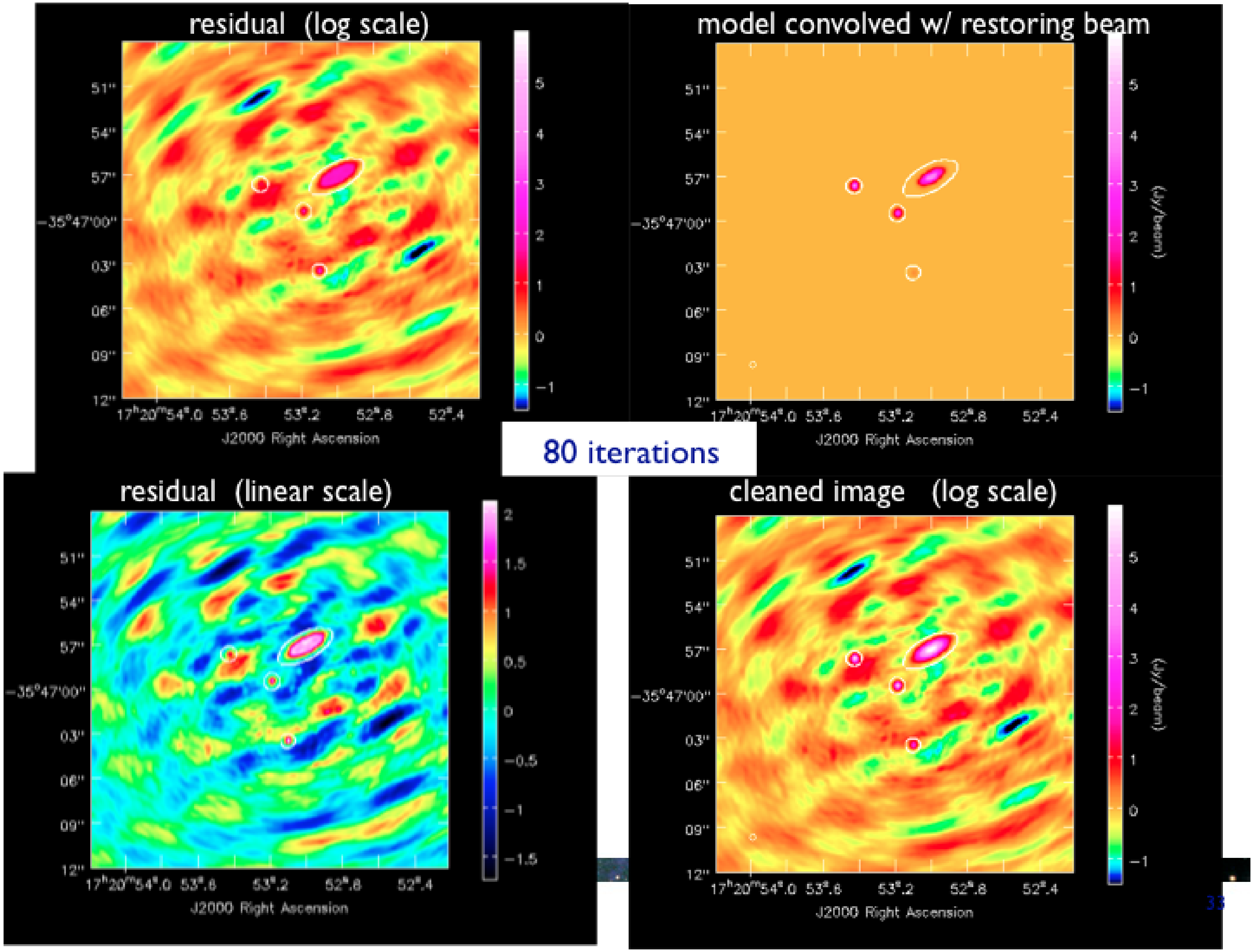
residual (log scale)

model convolved w/ restoring beam

70 iterations

residual (linear scale)

cleaned image (log scale)



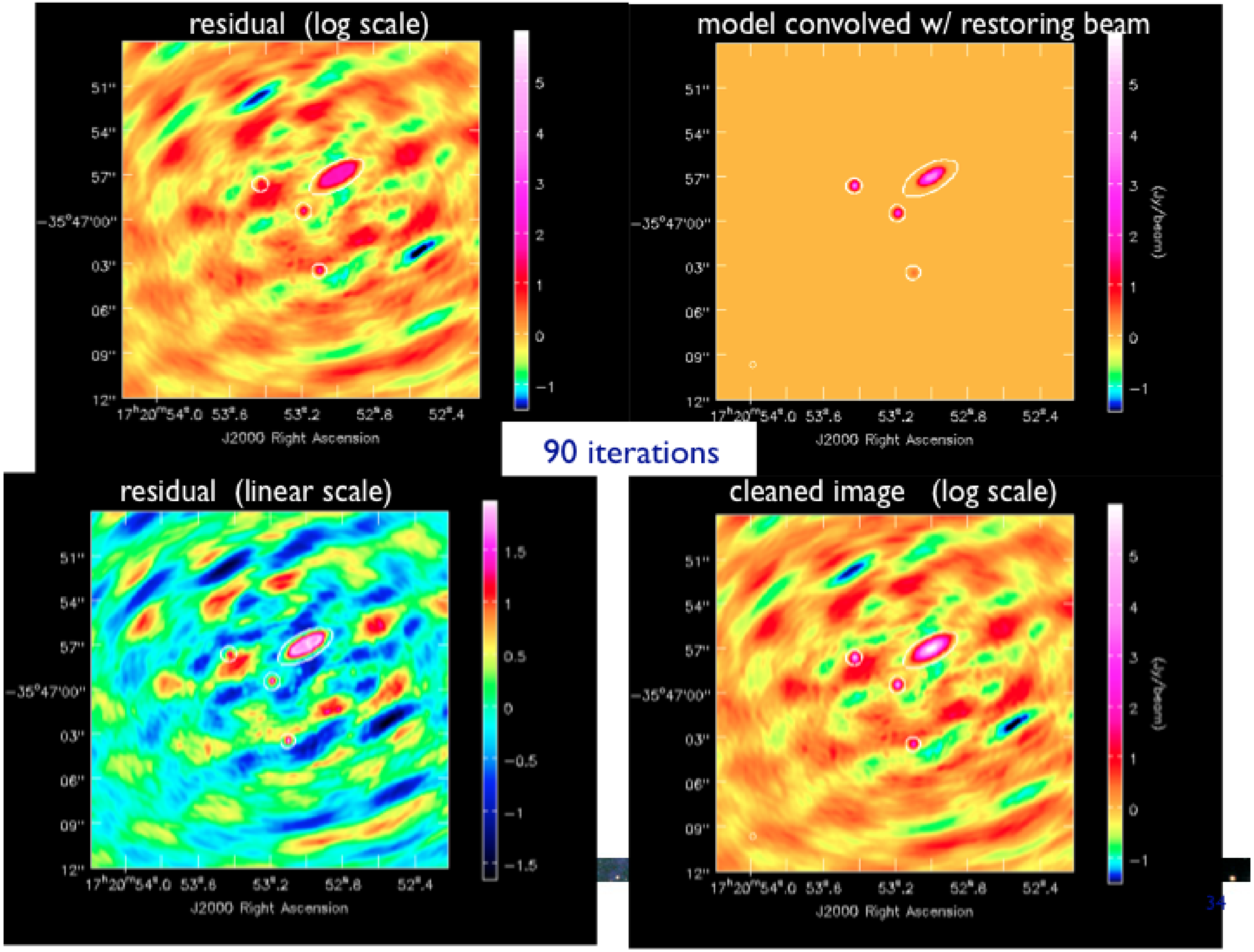
residual (log scale)

model convolved w/ restoring beam

80 iterations

residual (linear scale)

cleaned image (log scale)



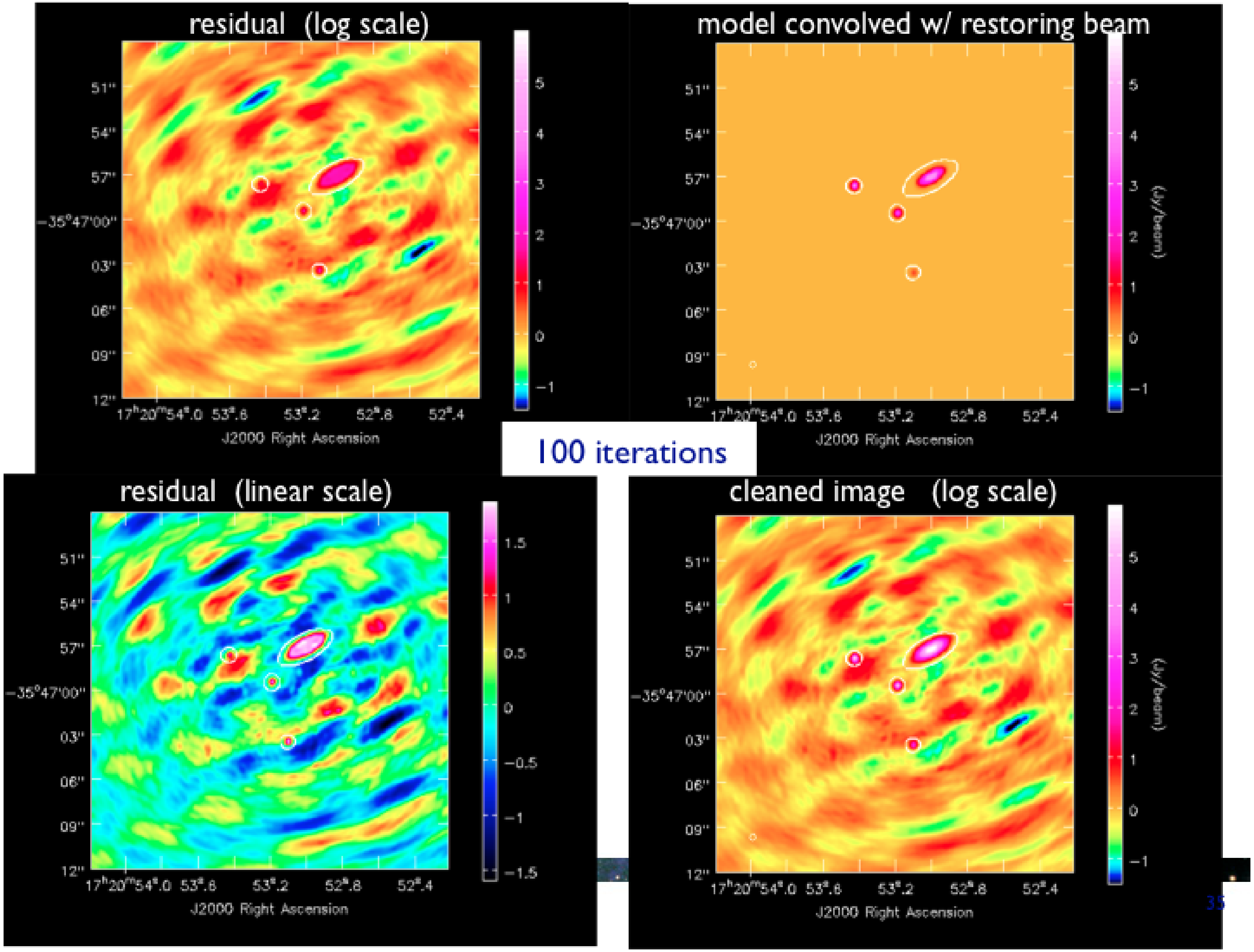
residual (log scale)

model convolved w/ restoring beam

90 iterations

residual (linear scale)

cleaned image (log scale)



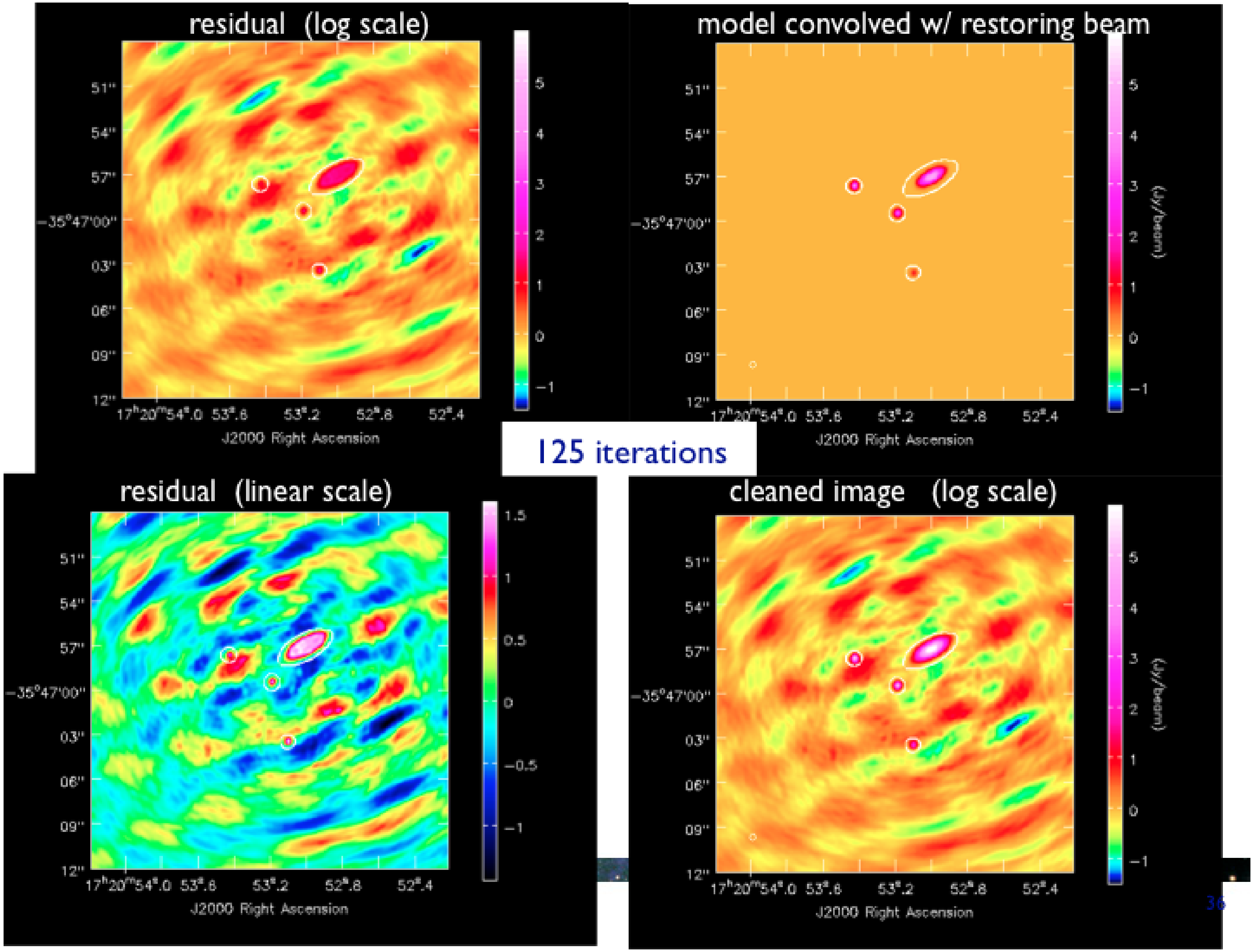
residual (log scale)

model convolved w/ restoring beam

100 iterations

residual (linear scale)

cleaned image (log scale)



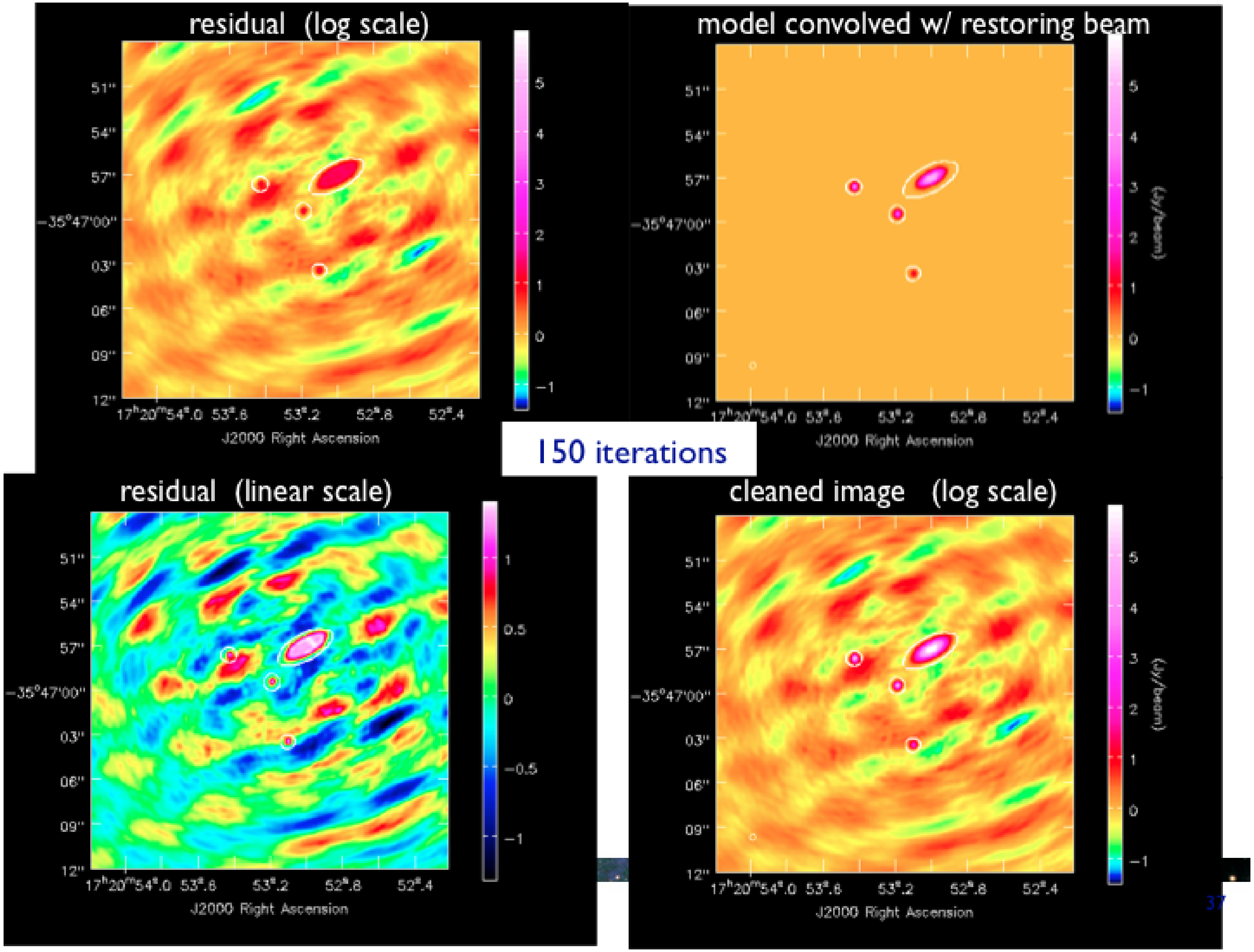
residual (log scale)

model convolved w/ restoring beam

125 iterations

residual (linear scale)

cleaned image (log scale)

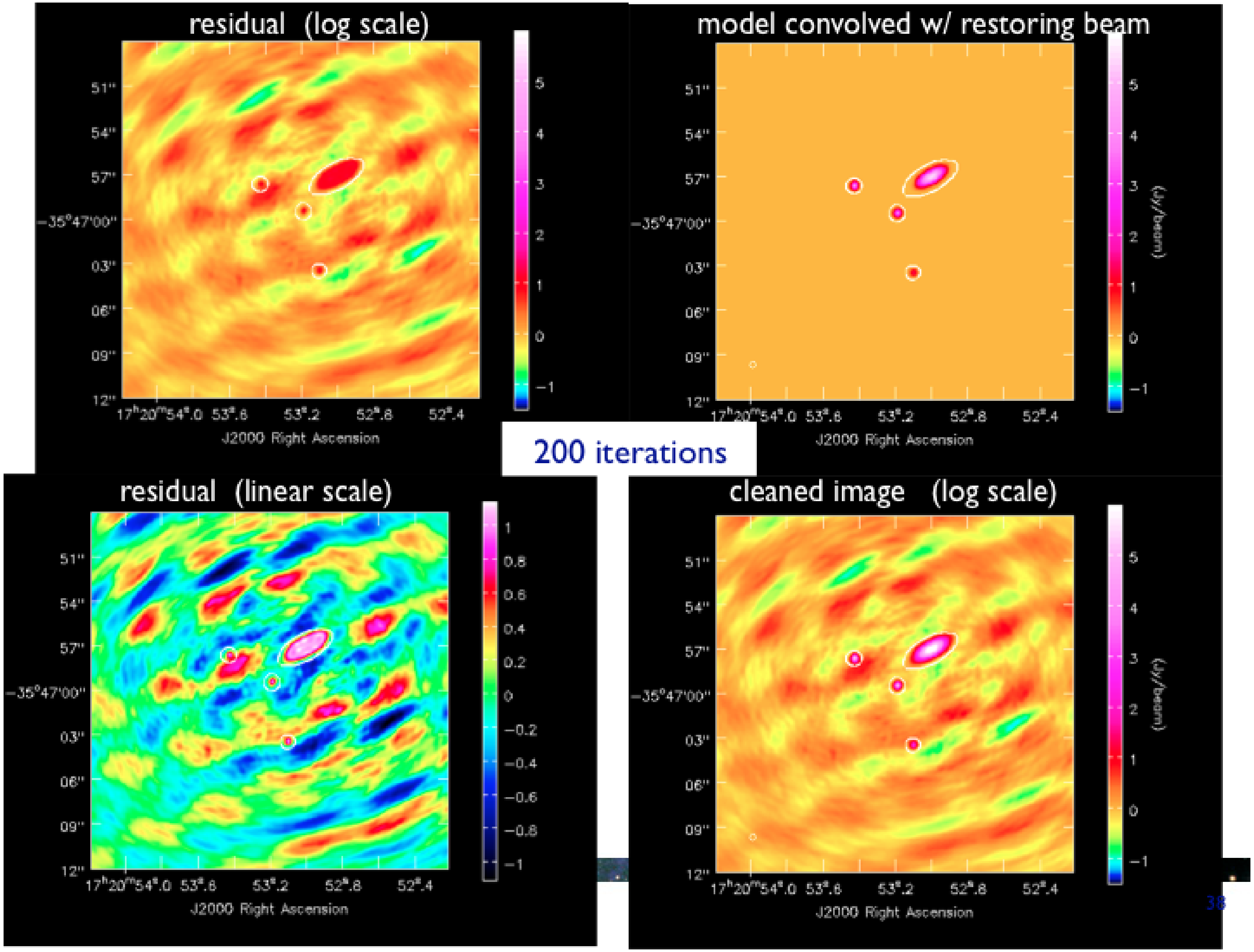

residual (log scale)

model convolved w/ restoring beam

150 iterations

residual (linear scale)

cleaned image (log scale)



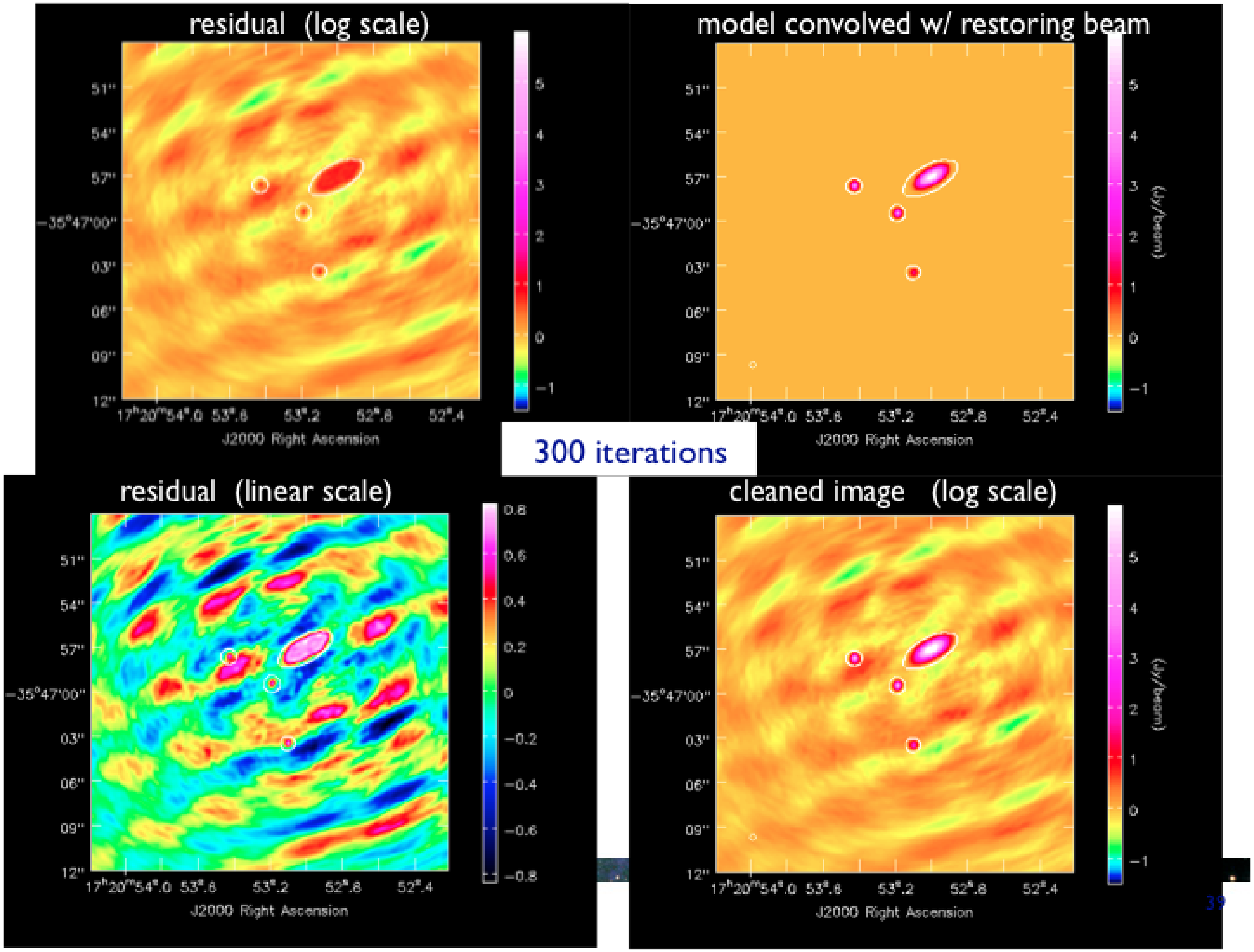
residual (log scale)

model convolved w/ restoring beam

200 iterations

residual (linear scale)

cleaned image (log scale)



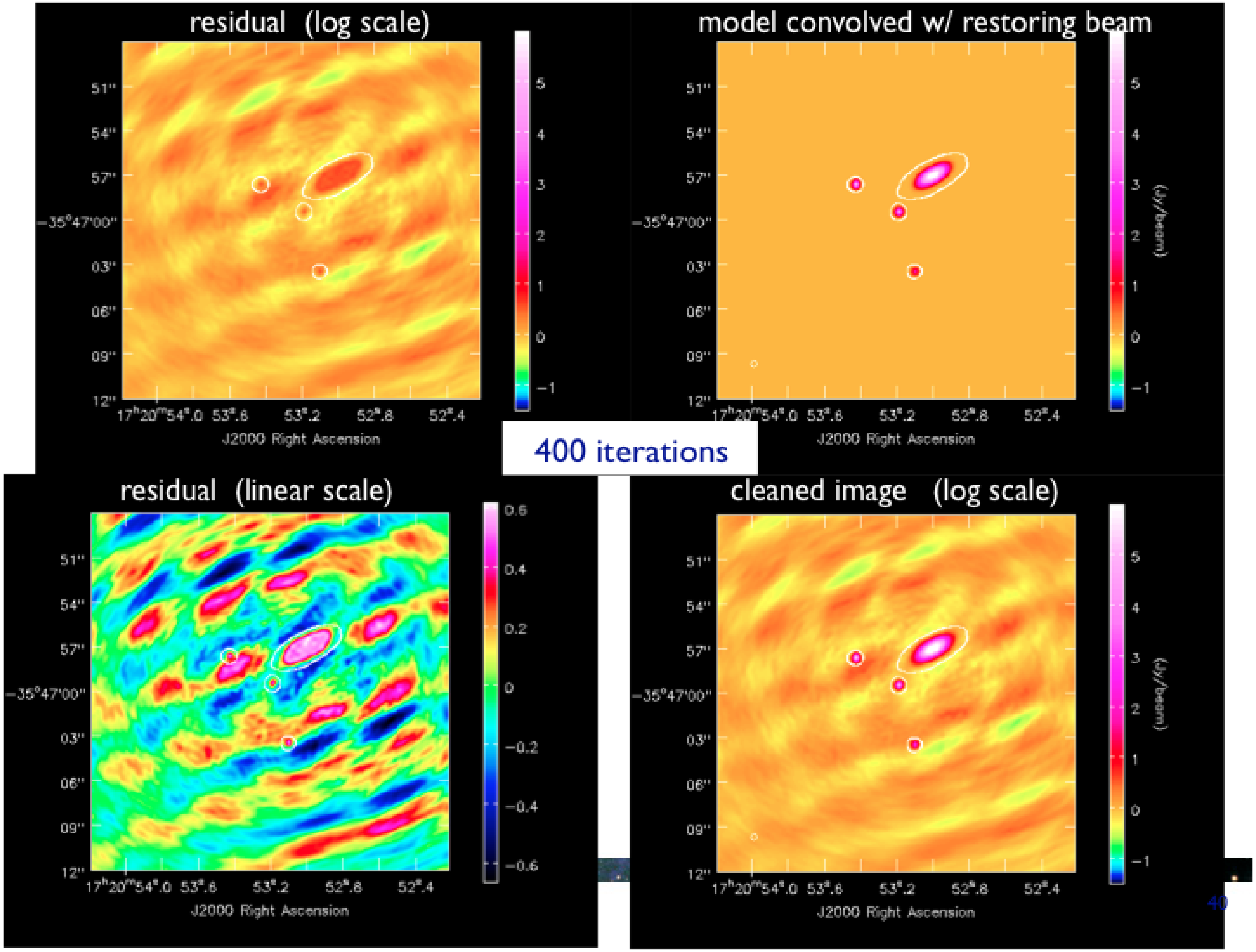
residual (log scale)

model convolved w/ restoring beam

300 iterations

residual (linear scale)

cleaned image (log scale)



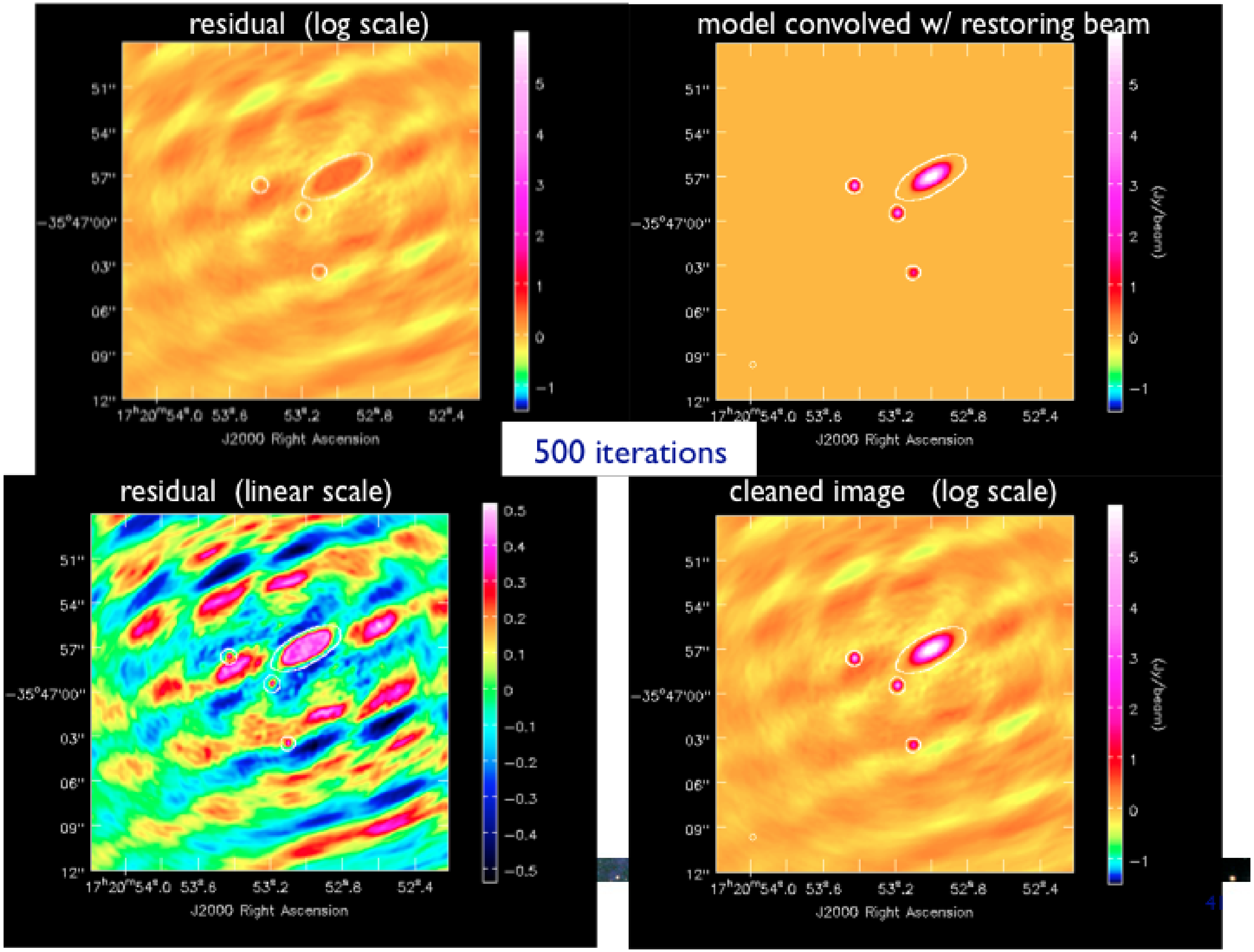
residual (log scale)

model convolved w/ restoring beam

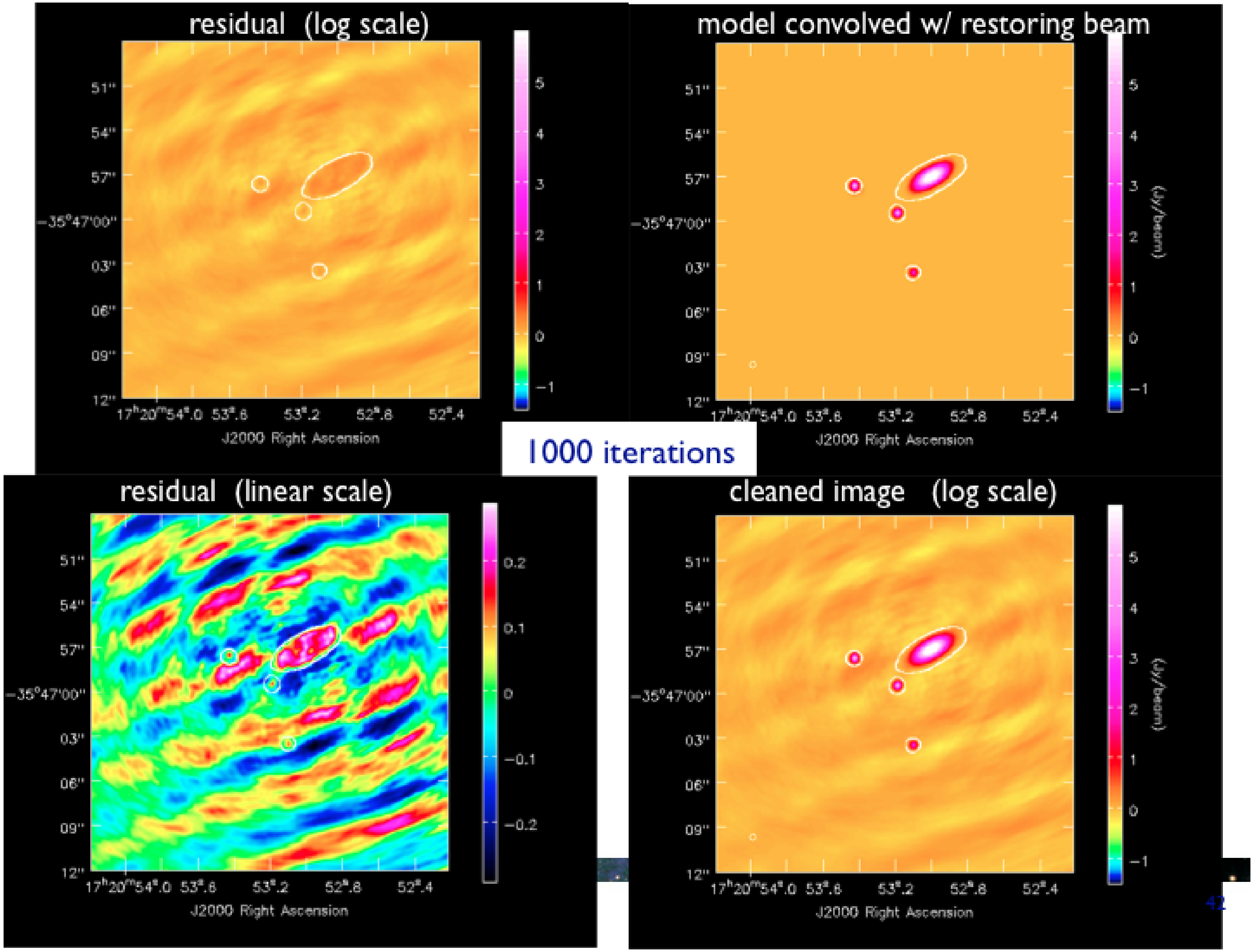
400 iterations

residual (linear scale)

cleaned image (log scale)



500 iterations



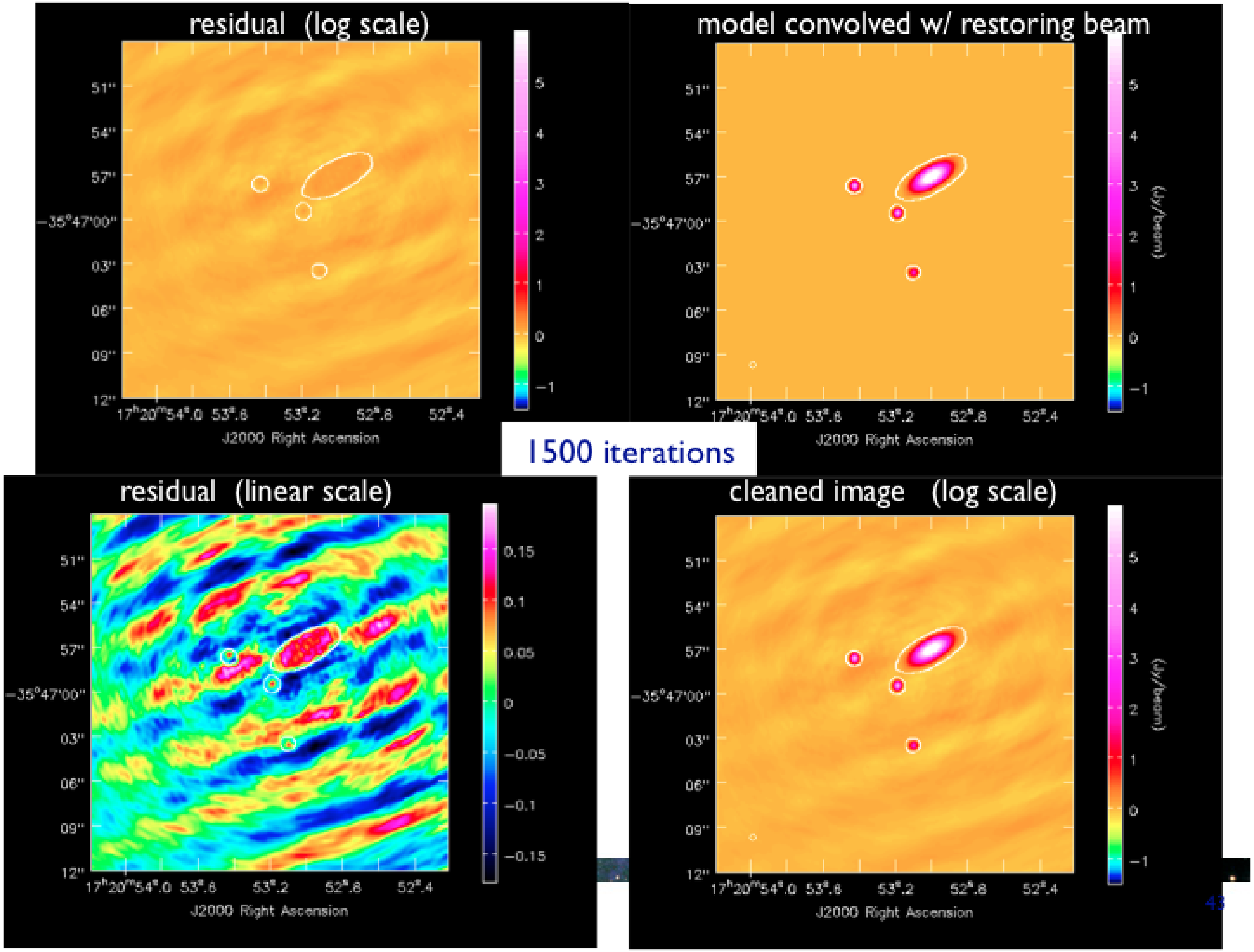
residual (log scale)

model convolved w/ restoring beam

1000 iterations

residual (linear scale)

cleaned image (log scale)



residual (log scale)

model convolved w/ restoring beam

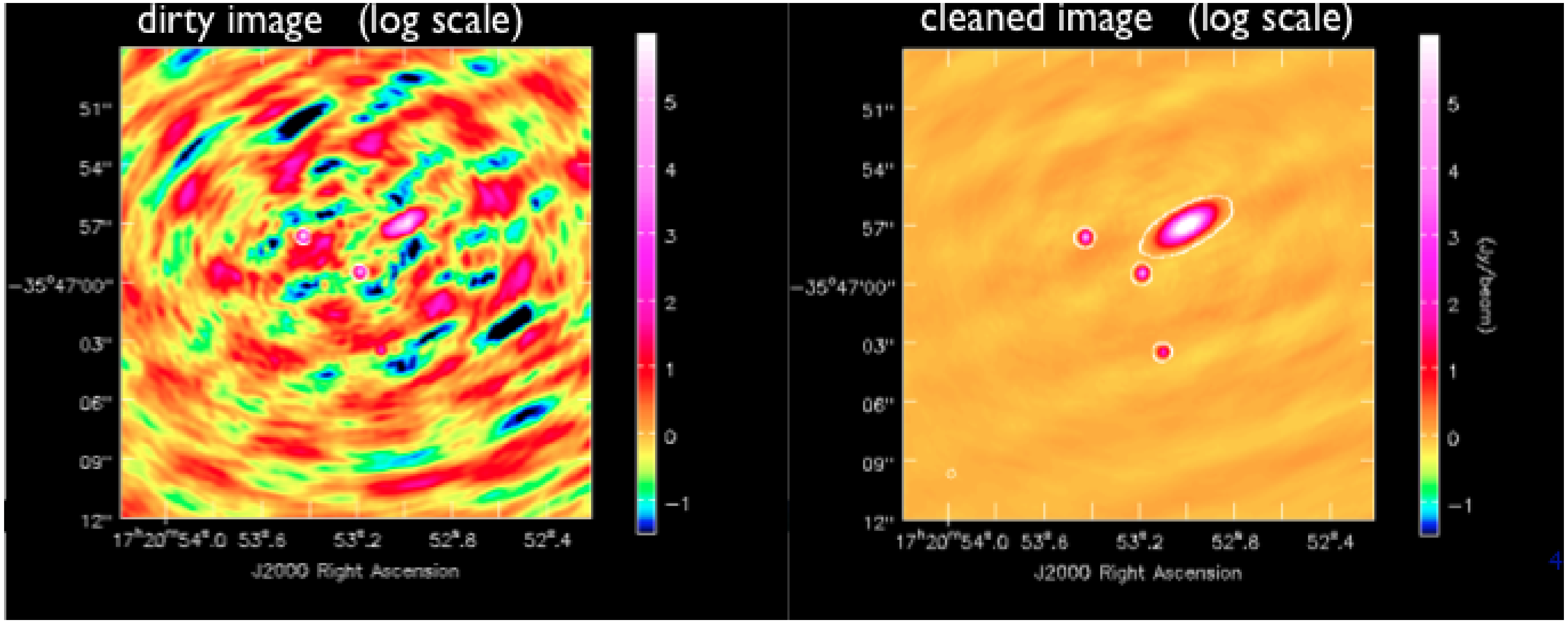
1500 iterations

residual (linear scale)

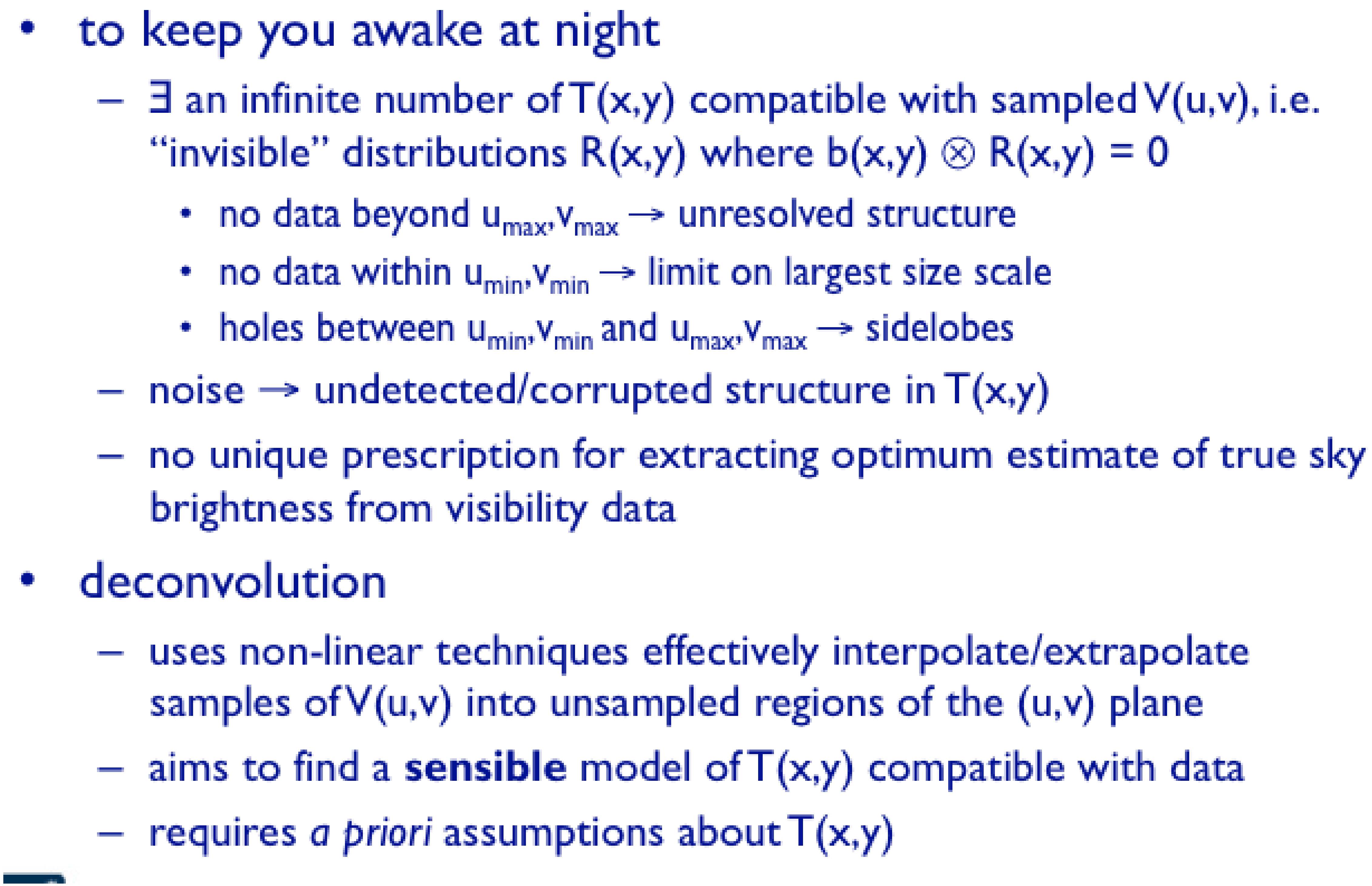
cleaned image (log scale)

Deconvolution

- Results depend on:
  - image parameters: size, cell, weighting, mosaic, cube
  - deconvolution parameters: algorithm, iterations, boxing, stopping criteria



Deconvolution



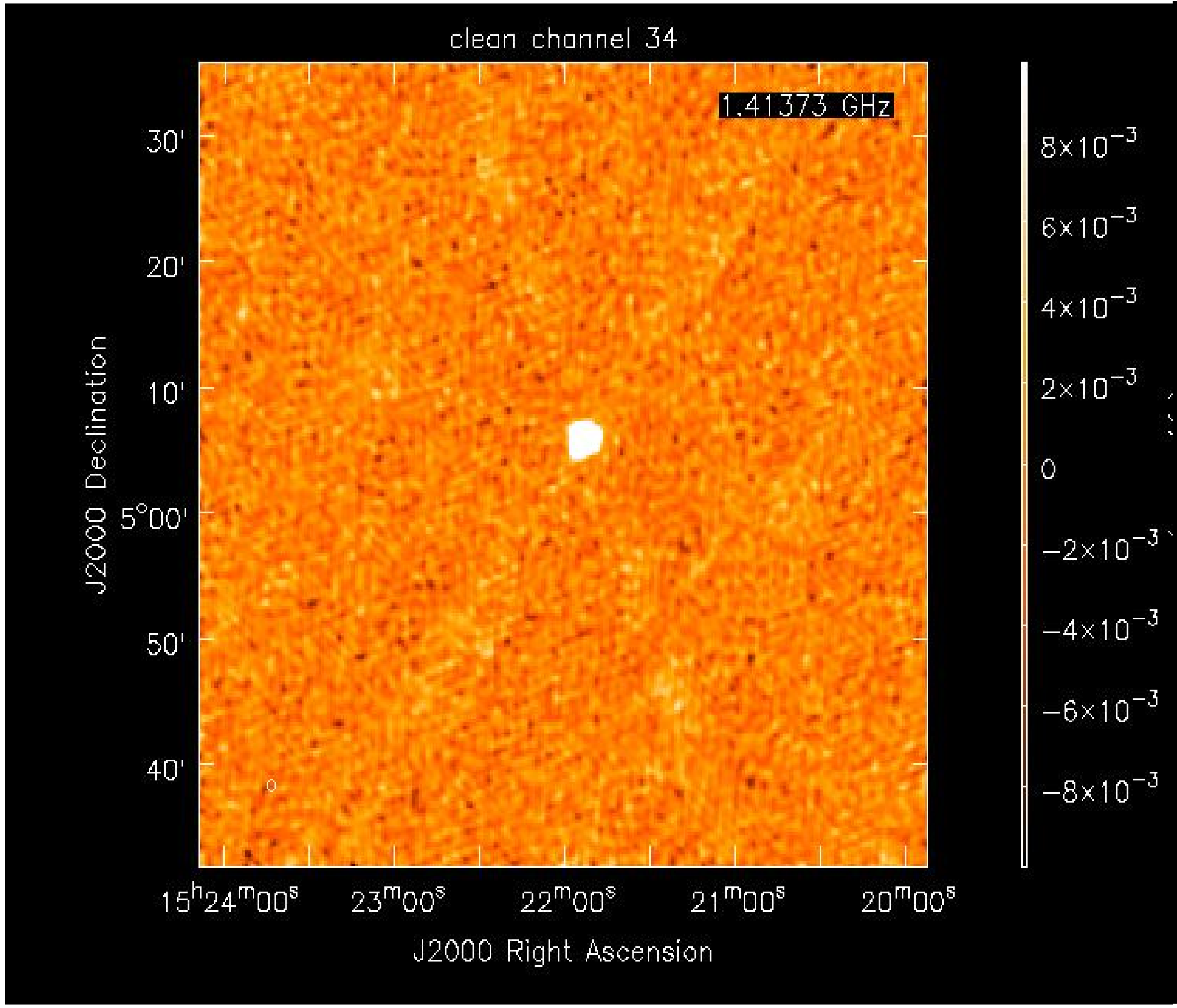
- to keep you awake at night
  - ∃ an infinite number of T(x,y) compatible with sampled V(u,v), i.e. "invisible" distributions R(x,y) where $b(x,y) \otimes R(x,y) = 0$
    - no data beyond u_{max}, v_{max} → unresolved structure
    - no data within u_{min}, v_{min} → limit on largest size scale
    - holes between u_{min}, v_{min} and u_{max}, v_{max} → sidelobes
  - noise → undetected/corrupted structure in T(x,y)
  - no unique prescription for extracting optimum estimate of true sky brightness from visibility data
- deconvolution
  - uses non-linear techniques effectively interpolate/extrapolate samples of V(u,v) into unsampled regions of the (u,v) plane
  - aims to find a **sensible** model of T(x,y) compatible with data
  - requires *a priori* assumptions about T(x,y)

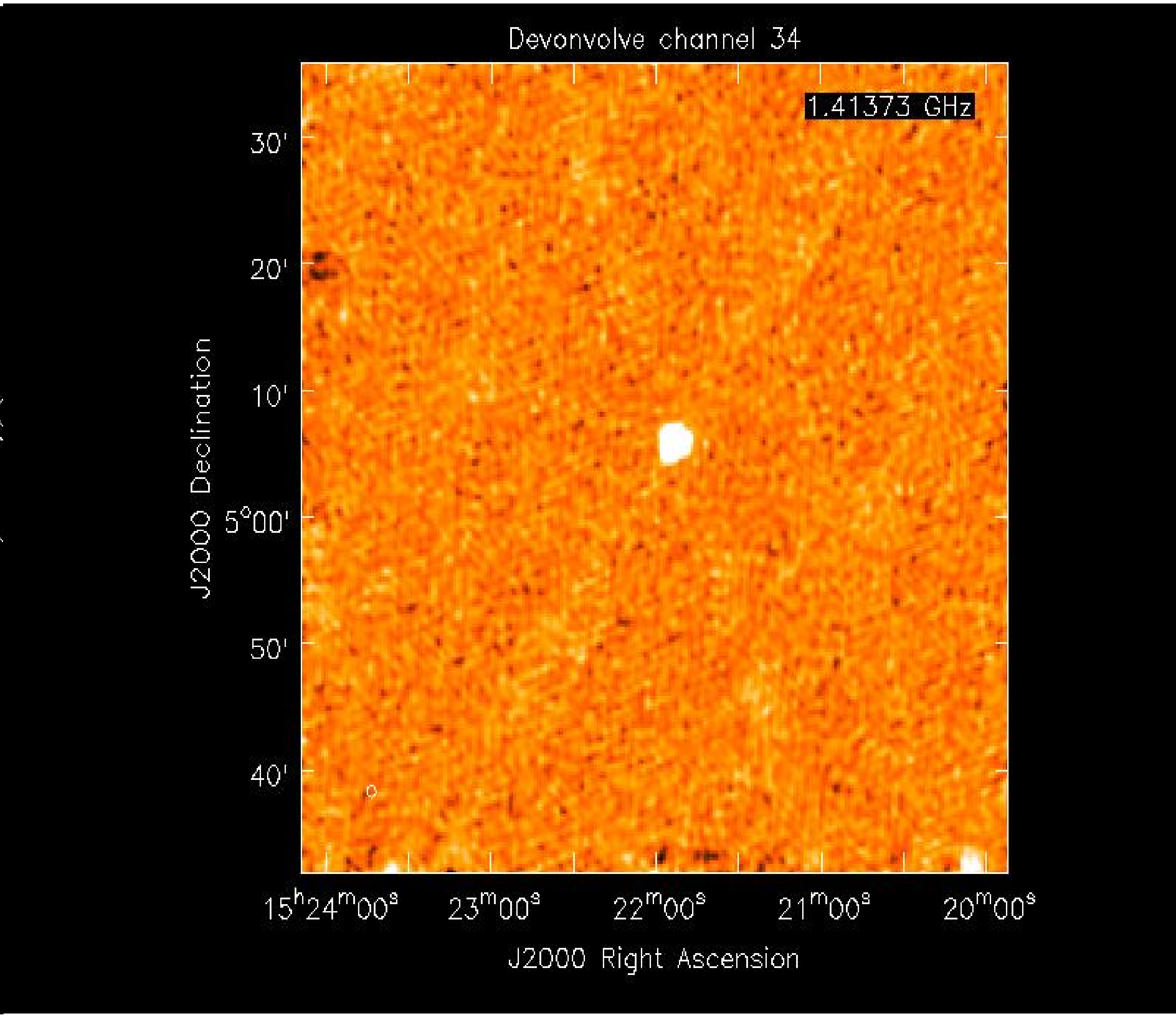
How to image in casa

- task **clean**: single field, mosaic, heterogeneous imaging, interactive and non-interactive
- task **widefield**: (low frequency) widefield imaging (w-projection and faceting)
- task **feather**: combine singe dish and interferometry

imaging

- Do Fourier transform then deconvolution, or do both at the same time (preferred): use task `clean`





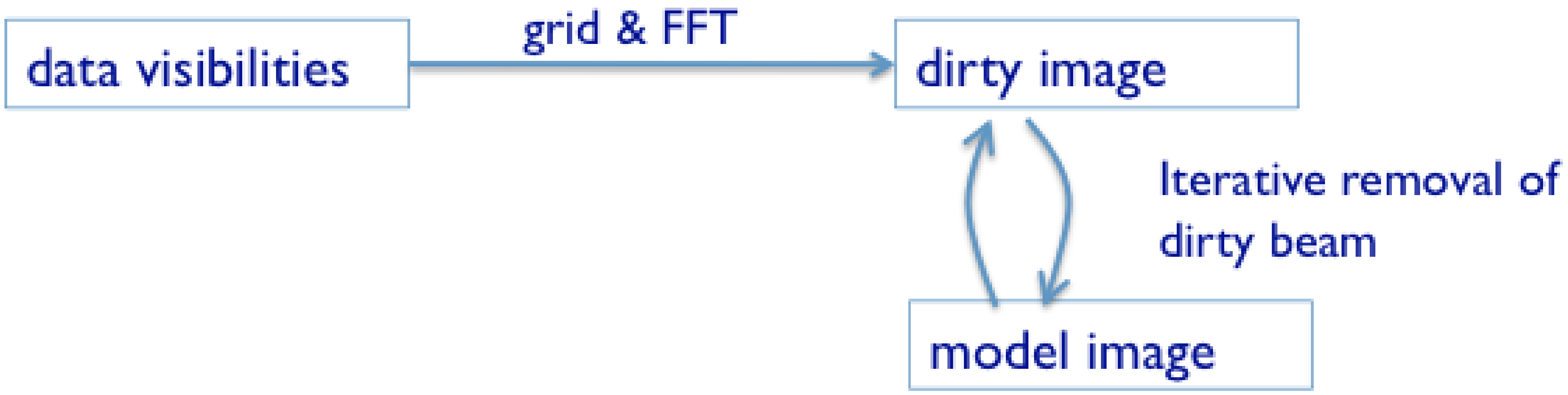
Subtracting from the ungridded visibilities

- Reduce artifacts from aliasing
- Aliasing can be removed by making a psf twice as large as the required image
- Be ready for self calibration

Imaging parameters

- **psfmode**: algorithm for calculating the psf (Clark or Hogbom)
- **imagermode**: ‘’, ‘csclean’ or ‘mosaic’
  - **hogbom** – slow and subtraction not as accurate
  - **clark** - uses minor and major cycles – in major cycles clean components are FFT’ed and subtracted from the FFT of the residual image from previous major cycle.
  - **csclean** – similar to ‘clark’ except clean components are subtracted from ungridded visibilities at each major cycle. Slower than ‘clark’ but more accurate. Should be used for high dynamic range and is always used for mosaics.

Deconvolution algorithms



data visibilities

grid & FFT

dirty image

Iterative removal of dirty beam

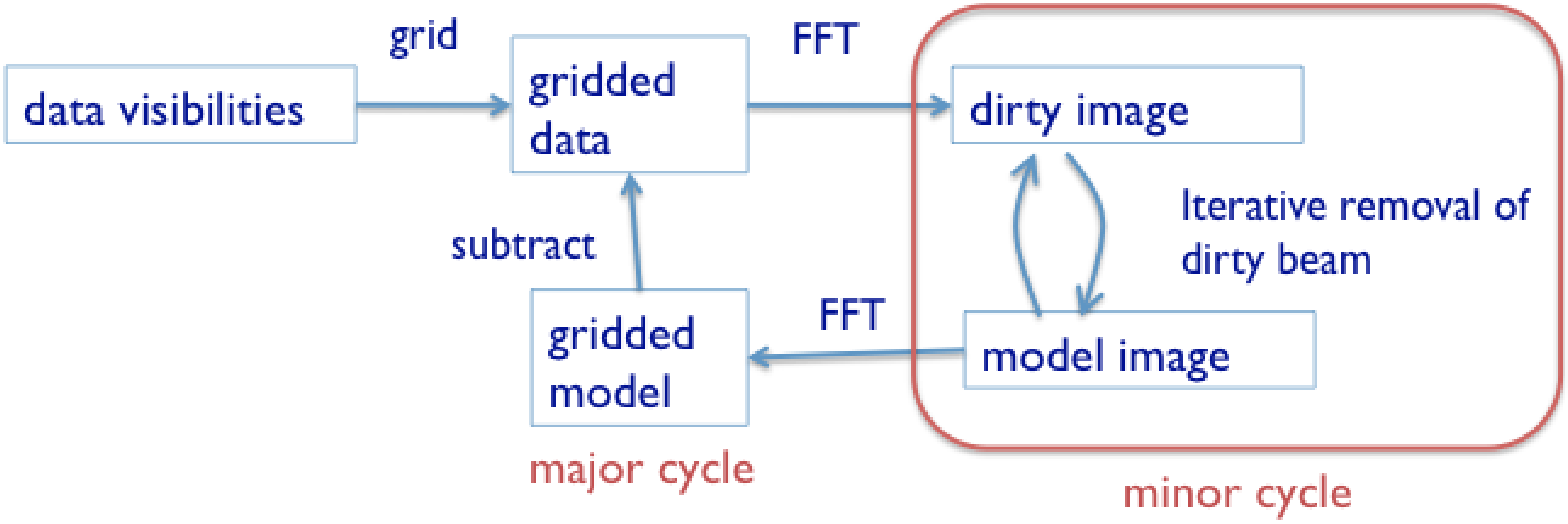
model image

Hogbom clean: (Högbom 1974)

subtracts full PSF in image domain

fast but not very accurate: errors build

Deconvolution algorithms



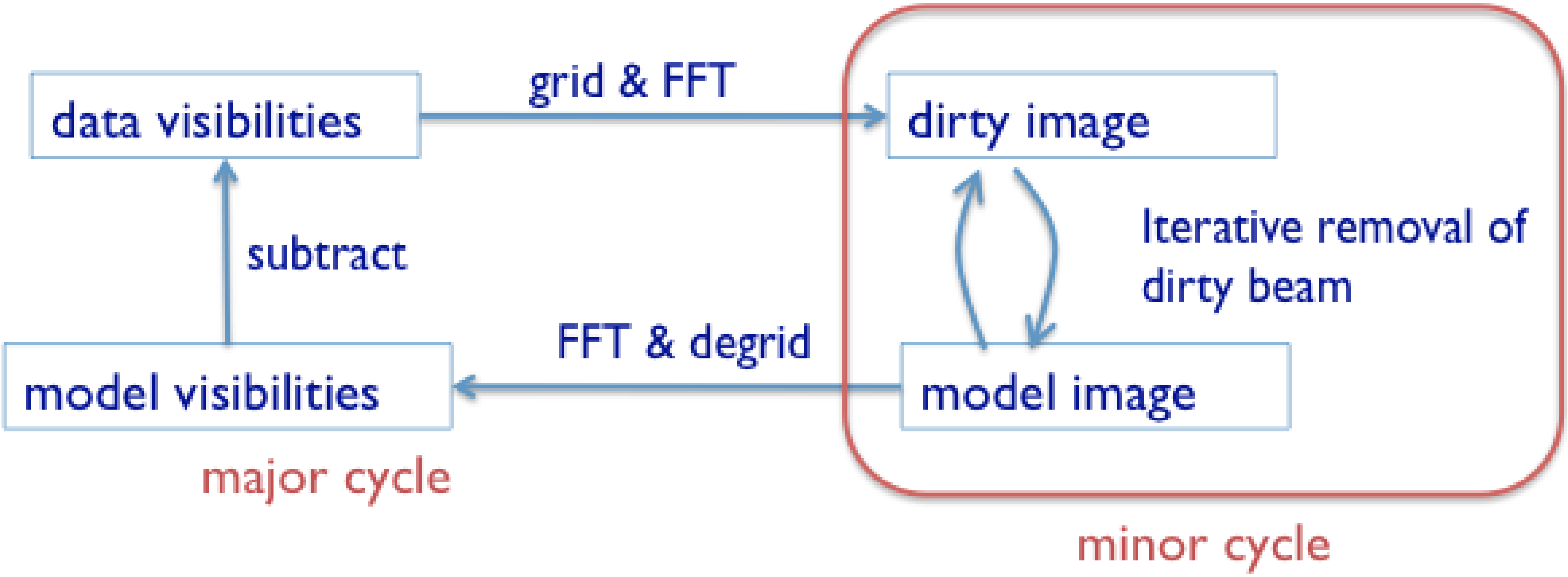
Clark :

subtracts truncated PSF in image domain

periodically subtracts from gridded data in uv domain

major/minor cycle frequency controlled by cyclefactor parameter

Deconvolution algorithms



data visibilities

grid & FFT

dirty image

subtract

Iterative removal of dirty beam

FFT & degrid

model visibilities

model image

major cycle

minor cycle

Cotton-Schwab :

subtracts truncated PSF in image domain

major cycle subtracts from full visibilities

significant I/O per major cycle

Imaging parameters

- **niter** – Number of clean iterations to do. (This can be useful when you are doing tests, but this parameter has no physical meaning)
- **threshold** – Stop cleaning when peak residual has this value, give units (i.e. mJy). (One would like to approach about 3x the theoretical rms noise)
- **interactive** - interactive clean (uses the viewer)
- **weighting** - natural, uniform, briggs
- **cell** - cellsize, 3-5 cells across beam (cell=‘0.25arcsec’)
- **imsize** - not necessarily power of 2. Helps if multiple of 2,3 and 5

Imaging parameters - channelization

- mode – ‘mfs’,‘channel’, velocity’, or ‘frequency’

- **mfs**: emulates **multi-frequency synthesis** in that each visibility-channel datum k with baseline vector B_k at wavelength k is gridded into the uv-plane at $u_k = B_k /k$

  - The result is a single image plane, regardless of how many channels are in the input dataset.

  - image plane is at the frequency given by the midpoint between the highest and lowest frequency channels in the input spw

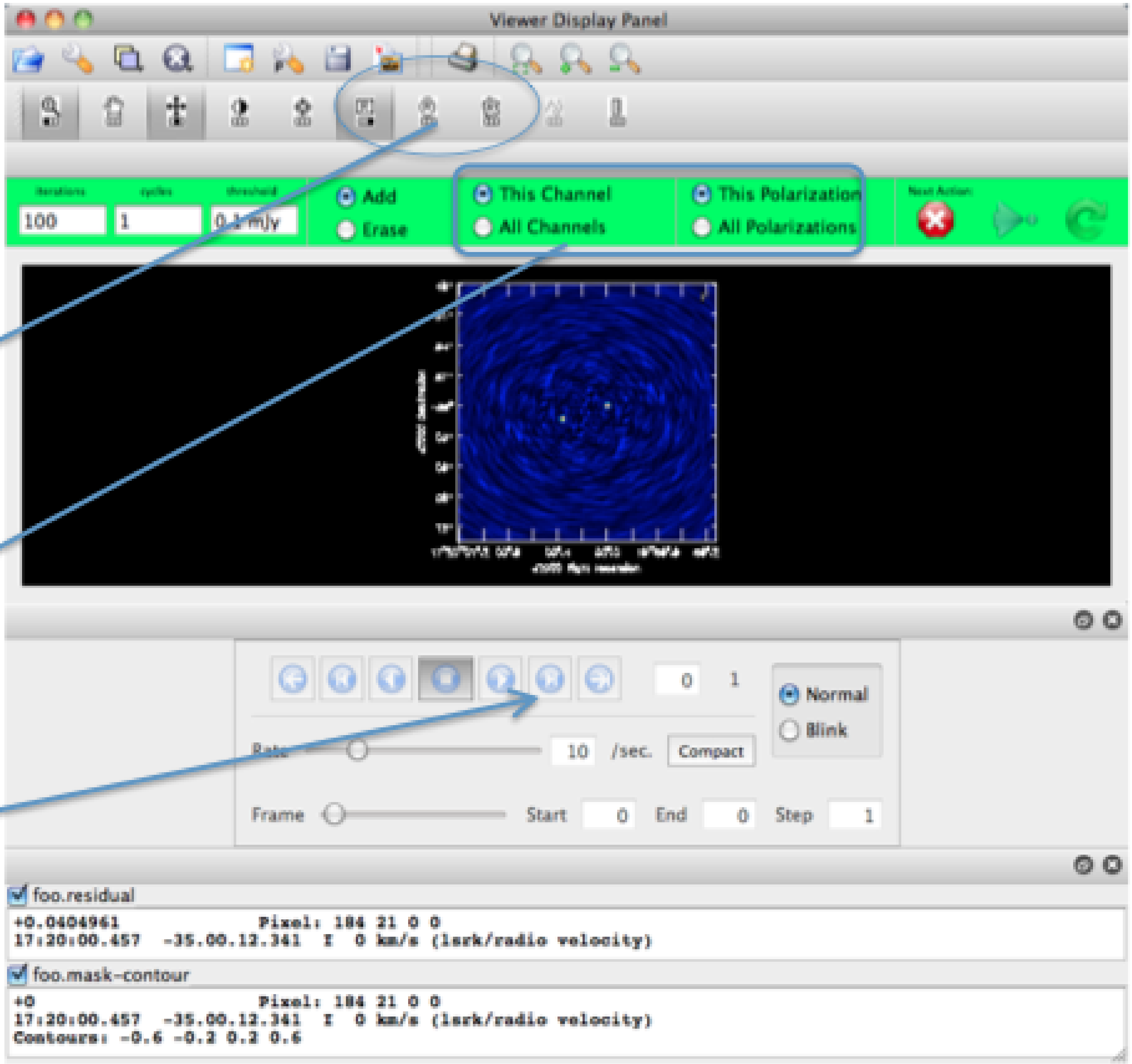
- If mode=’velocity’ is chosen, then data will be gridded into LSRK bins

- mode ‘channel’ or frequency will map every channel/freq independently

- There is currently a separate task called **hanningsmooth** that performs spectral Hanning smoothing

Interactive Clean

- residual image in viewer
- define a mask with R-click on shape type
- define the same mask for all channels
- or iterate through the channels with the tape deck and define separate masks



iterations 100, cycles 1, threshold 0.1 mJy

● Add ○ Erase

● This Channel ○ All Channels

● This Polarization ○ All Polarizations

Frame Start 0 End 0 Step 1

☑ foo.residual

```
+0.0404961               Pixel: 184 21 0 0
17:20:00.457  -35.00.12.341  I  0 km/s (lsrk/radio velocity)
```

☑ foo.mask-contour

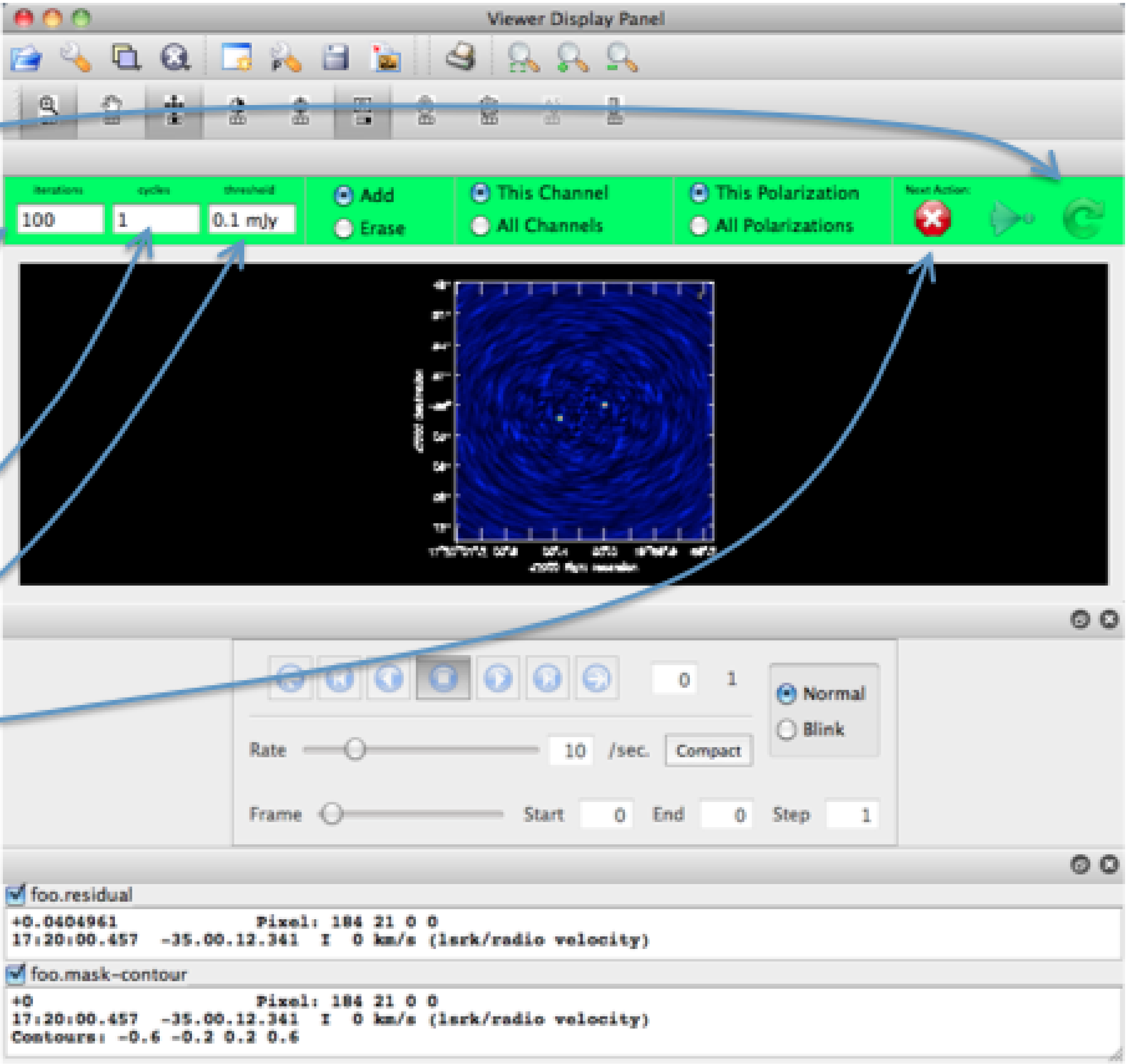
```
+0                       Pixel: 184 21 0 0
17:20:00.457  -35.00.12.341  I  0 km/s (lsrk/radio velocity)
Contours: -0.6 -0.2 0.2 0.6
```

Interactive Clean

- perform N iterations

- and return – every time the residual is displayed is a major cycle

- continue until #cycles or threshold reached, or user stop



iterations: 100
cycles: 1
threshold: 0.1 mJy
● Add ○ Erase
● This Channel ○ All Channels
● This Polarization ○ All Polarizations
● Normal ○ Blink
Rate 10 /sec.
Frame Start 0 End 0 Step 1

☑ foo.residual

```
+0.0404961            Pixel: 184 21 0 0
17:20:00.457  -35.00.12.341  I  0 km/s (lsrk/radio velocity)
```

☑ foo.mask-contour

```
+0                    Pixel: 184 21 0 0
17:20:00.457  -35.00.12.341  I  0 km/s (lsrk/radio velocity)
Contours: -0.6 -0.2 0.2 0.6
```

files created by imaging

- imagename=’imagename’
- imagename.**image** – final cleaned (or dirty if niter=0 image)
- imagename.**psf** – the point spread function of the beam
- imagename.**model** – an image of the clean components
- imagename.**residual** – the residual image after subtracting clean components
- imagename.**flux** – the primary beam response function – used to make a “flux correct image”, otherwise flux is only correct at the phase center(s)

What’s in the ms?

Data	Original visibility data
Model	FT of image made from MS
	FT of supplied model image
	FT of calibrator flux density
Corrected	Copy of visibilities with calibration tables applied (Used in imaging but not in calibration)
Weights	Imaging weights

combining single dish and interferometric data

- task **feather**
  - The single-dish and interferometer images are Fourier transformed.
  - The beam from the single-dish image is Fourier transformed (FTSDB(u,v)).
  - The Fourier transform of the interferometer image is multiplied by (1 − FTSDB(u, v)). This basically down weights the shorter spacing data from the interferometer image.
  - The Fourier transform of the single-dish image is scaled by the volume ratio of the interferometer restoring beam to the single dish beam.
  - The results from last two steps are added and Fourier transformed back to the image plane.

mosaicing

- Imagermode='mosaic'

- mosweight – mosweight=T : individual mosaic fields will receive independent weights. Use mosweight=F which will give optimal signal to noise ratio.

- ftmachine = 'mosaic' uses Fourier transform of primary beam as gridding kernel – allowing multiple pointings to be gridded onto a single uv-plane. 'mosaic' is faster and best choice in most cases.

- ftmachine = 'ft' is slower but can be more accurate for sparse arrays and/or less than Nyquist mosaic sampling. 'ft' does a weighted linear combination of each pointing separately.

Miscellaneous

- clean will restart from existing files
  - will first recompute residuals from model
  - be sure this is what you want
  - mask image in particular can be reused but be careful of imsize
- total cleaned flux not reported until end
- log messages differ between algorithms (work in progress)
- don’t do Ctrl-C while imaging – can do bad things to your MS
- calready = True required to initialize the MS model column for self-cal