“How to apply the different methods, and what is missing”

Improving Image Fidelity on Astronomical Data: Radio Interferometer and Single-Dish Data Combination

Lorentz Center, Leiden

Sandra Burkutean
Italian ALMA Regional Centre

"(How to apply the different methods), and what is missing"

already covered in great detail in previous talks

"What is missing: bits and pieces of supplementary info for data combination"

based on

slack discussions, working group initial feedback, further tools and a potpourri of other ideas

Problems with image headers and how to solve them

In yesterday's group session summaries, image header problems came up quite frequently:

*** Error *** 2019-08-13 13:59:28 SEVERE feather::imager::feather() Caught exception: (../../images/Images/ImageBeamSet.cc : 109) Failed AlwaysAssert chan >=0 && chan < Int(nchan()) && stokes >= 0 && stokes < Int(nstokes())
2019-08-13 13:59:28 SEVERE feather:::: An error occurred running task feather.

```
CASA <3>: inp
--------> inp()
#  imhead :: List, get and put image header parameters
imagename           =         ''        #  Name of the input image
mode                =  'summary'        #  Mode of operation: "add", "del",
#   "get", "history", "list", "put", or
#   "summary". Modes "add", "del", and
#   "put" will not work if the image is
#   read-only (eg a FITS image).
verbose             =      False        #  Give a full listing of beams or just
#   a short summary? Only used when the
#   image has multiple beams and
#   mode="summary".
```

get

```
hdkey               =         ''
```

list

lists all keywords in header

put

```
hdkey               =         ''
hdvalue             =         ''
```

summary

Problems with image headers and how to solve them

In yesterday's group session summaries, image header problems came up quite frequently:

The Astropy Project

```
>>> fits_image_filename = fits.util.get_testdata_filepath('test0.fits')
>>> hdul = fits.open(fits_image_filename)  # open a FITS file
>>> hdr = hdul[0].header  # the primary HDU header
```

```
>>> hdr.set('target', 'NGC1234', 'target name')
>>> # place the next new keyword before the 'TARGET' keyword
>>> hdr.set('newkey', 666, before='TARGET')  # comment is optional
>>> # place the next new keyword after the 21st keyword
>>> hdr.set('newkey2', 42.0, 'another new key', after=20)
```

yesterday's slack item

Adam Ginsburg 14:51 Uhr

```
from astropy.io import fits
fh = fits.open('skymodel.fits')
fh[0].header['CRVAL1'] = center.ra.deg
fh[0].header['CRVAL2'] = center.dec.deg
fh[0].header['CRVAL3'] = ch0freq[0][0]
fh.writeto('skymodel_at_m100.fits')
```

Problems with image headers and how to solve them

In yesterday's group session summaries, image header problems came up quite frequently:

imsmooth

A deconvolved image $\mathbf{I}$ can be smoothed to a target resolution by convolving it with a Gaussian beam $\mathbf{B}_{\rm tar}$. If the image is already convolved with another smaller beam $\mathbf{B}_{\rm cur}$ a correcting beam $\mathbf{B}_{\rm cor}$ can be calculated so that

$$\mathbf{B}_{\rm tar} * \mathbf{I} = \mathbf{B}_{\rm cor} * (\mathbf{B}_{\rm cur} * \mathbf{I}),$$

where $*$ is the convolution operator. The Fourier transform of the above equation is

$$\mathbf{B}^{\rm f}_{\rm tar}\mathbf{I}^{\rm f} = \mathbf{B}^{\rm f}_{\rm cor}(\mathbf{B}^{\rm f}_{\rm cur}\mathbf{I}^{\rm f}),$$

where the superscript f indicates the Fourier transform. The correcting beam can then be obtained by

$$\mathbf{B}_{\rm cor} = \mathcal{F}^{-1}\left(\frac{\mathbf{B}^{\rm f}_{\rm tar}}{\mathbf{B}^{\rm f}_{\rm cur}}\right).$$

Alternatively, if the input image has multiple beams, setting *kernel='**commonbeam**'* will result in the smallest area beam that encloses all beams in the image to be used as the target resolution to which to convolve all planes.

source: CASA cookbook

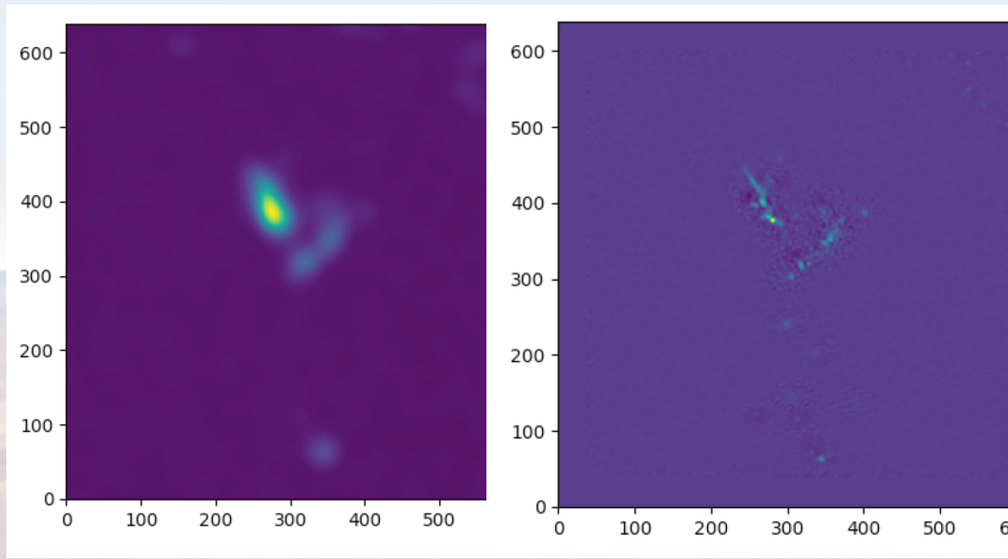
If in doubt, check combination by hand

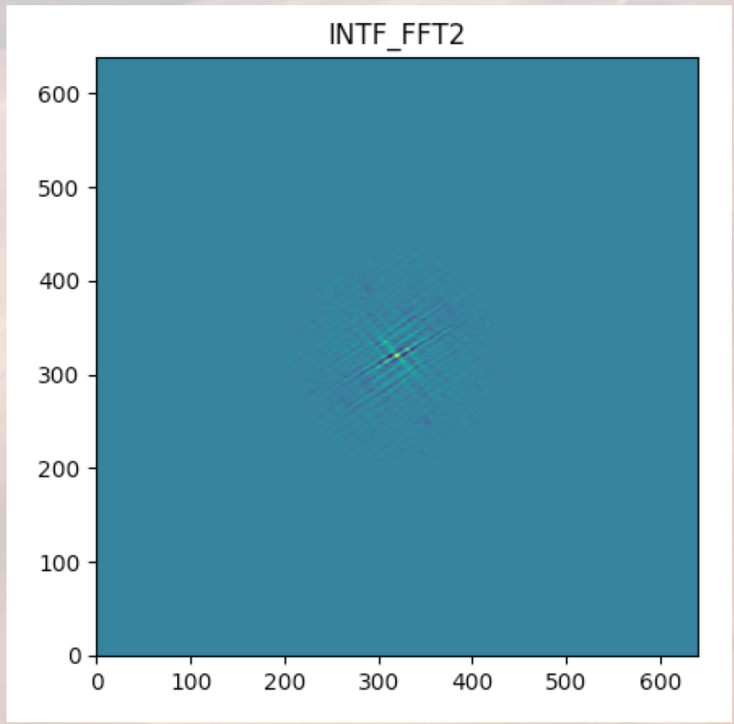
```
import numpy as np
import scipy as sc
from astropy.io import fits
import matplotlib.pyplot as plt
import math
```

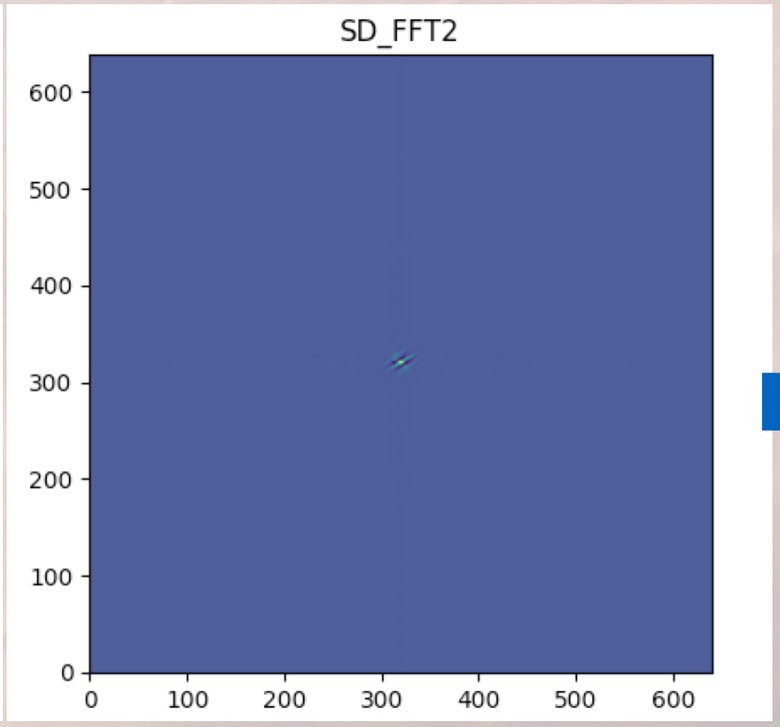
```
SD_in = fits.open('sd_regrid.fits')
INTF_in = fits.open('CHAN3_NGC_346_sci.spw22.cube.I.manual-region3by3-7m-7bins-leiden.image.pbcor.fits')


SD = SD_in[0].data[0,0,:,:]
INTF = INTF_in[0].data[0,0,:,:]

SD[np.isnan(SD)] = 0.0
INTF[np.isnan(INTF)] = 0.0
```







beam ratio scaling

apply weights

Further documentation on feather and the OBIT implementation

Publications of the Astronomical Society of the Pacific, 129:094501 (7pp), 2017 September https://doi.org/10.1088/1538-3873/aa793f

CrossMark

Fourier Plane Image Combination by Feathering

W. D. Cotton
National Radio Astronomy Observatory, 520 Edgemont Road, Charlottesville, VA, 22903 USA; bcotton@nrao.edu
Received 2017 May 18; accepted 2017 June 13; published 2017 July 20

Abstract

Astronomical objects frequently exhibit structure over a wide range of scales whereas many telescopes, especially interferometer arrays, only sample a limited range of spatial scales. To properly image these objects, images from a set of instruments covering the range of scales may be needed. These images then must be combined in a manner to recover all spatial scales. This paper describes the feathering technique for image combination in the Fourier transform plane. Implementations in several packages are discussed and example combinations of single dish and interferometric observations of both simulated and celestial radio emission are given.

Key words: techniques: image processing

Further documentation on feather and the OBIT implementation

(1) *Re-sample images*. The first step is to re-sample images with resolutions less than the maximum to the grid of the maximum resolution image with sufficient zero padding on the outside to allow an efficient FFT. The interpolation uses the Lagrangian technique in 2D to interpolate the pixels in the lower-resolution images at the locations of the highest-resolution image using a 5 × 5 pixel kernel.
(2) *Generate weighting masks.* For each resolution except the lowest, a real weighting mask is generated with a Gaussian hole in the center representing the spatial frequency range of the next lowest resolution. A sampling mask representing the spatial frequencies of each image is generated by
(a) Create image with the CLEAN beam at the center
(b) FFT
(c) Take real part
(d) Normalize to 1 at (0, 0) spatial frequency.

The weighting mask for each image i is

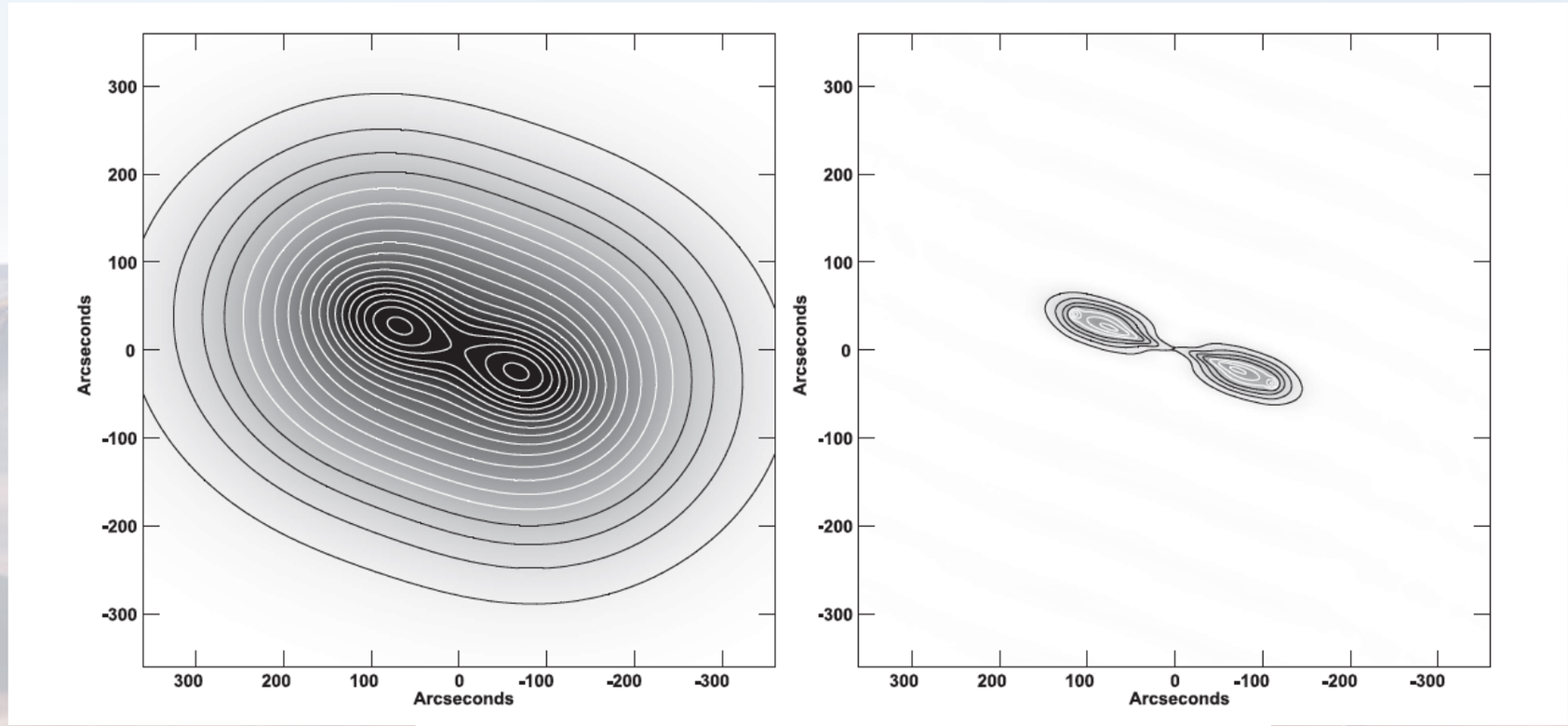
$$weight\_mask_i = 1.0 - sampling\_mask_{i+1},$$

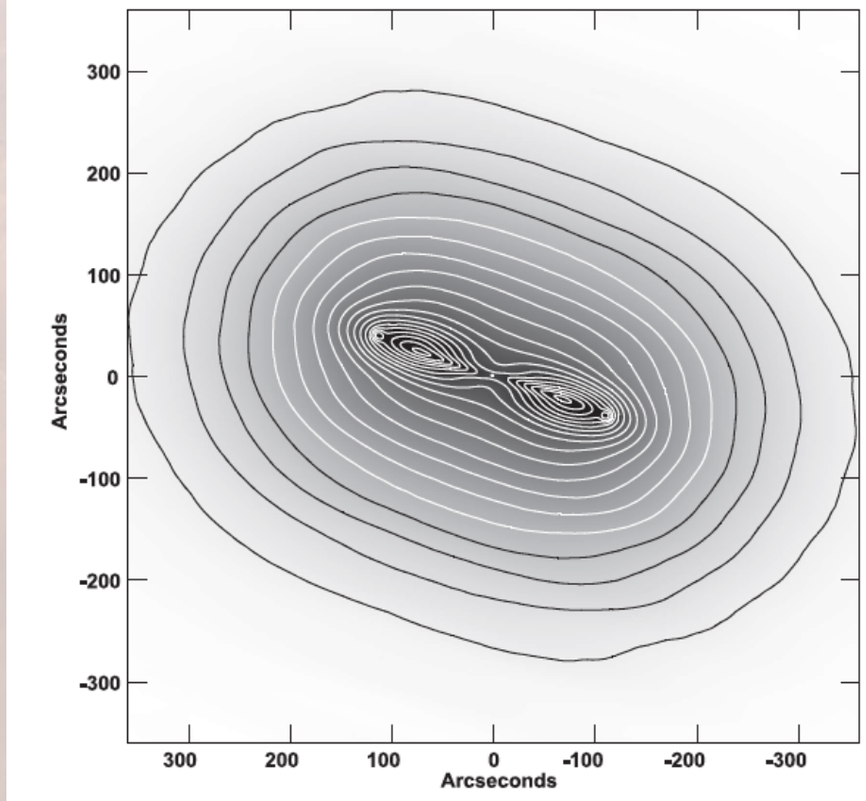
where $i + 1$ indicates the next lowest resolution. The weighting mask for the lowest resolution is 1.0 everywhere. The weighting masks are multiplied by the weights assigned to the input images.
(3) *FFT.* Each image is FFTed to the uv-plane.
(4) *Weight.* Multiply Fourier transform of image by its weighting mask.
(5) *Accumulate.* Sum the Fourier transform of the images times weight.
(6) *Inverse FFT.* Fourier transform back to the image plane.

7) *Normalize.* The normalization factor is determined by repeating the process but replacing the image with its corresponding CLEAN beam. The normalization factor is 1.0/flux density (center pixel) of the feathered beam.

Cotton et al. 2017

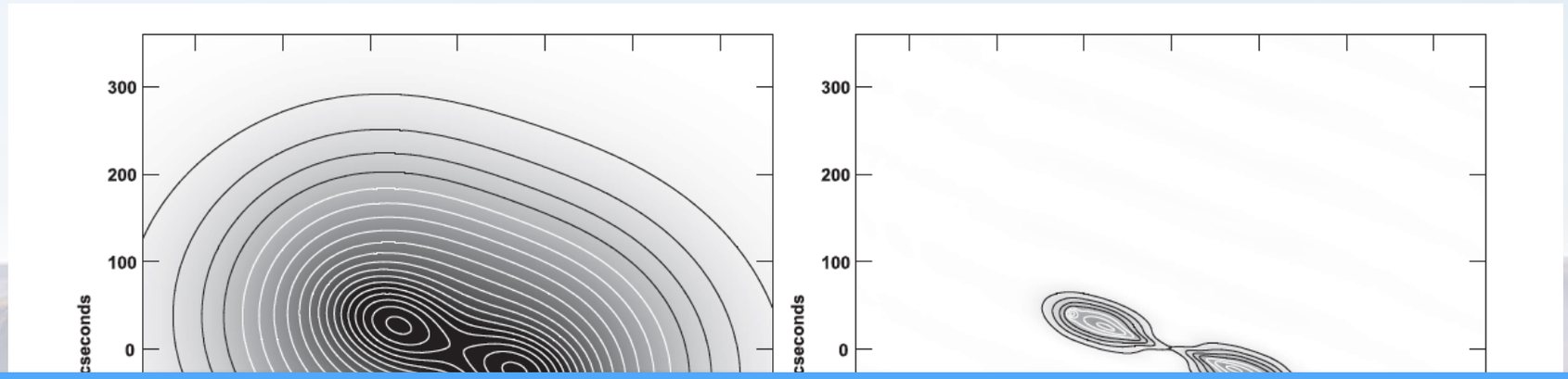
Further documentation on feather and the OBIT implementation



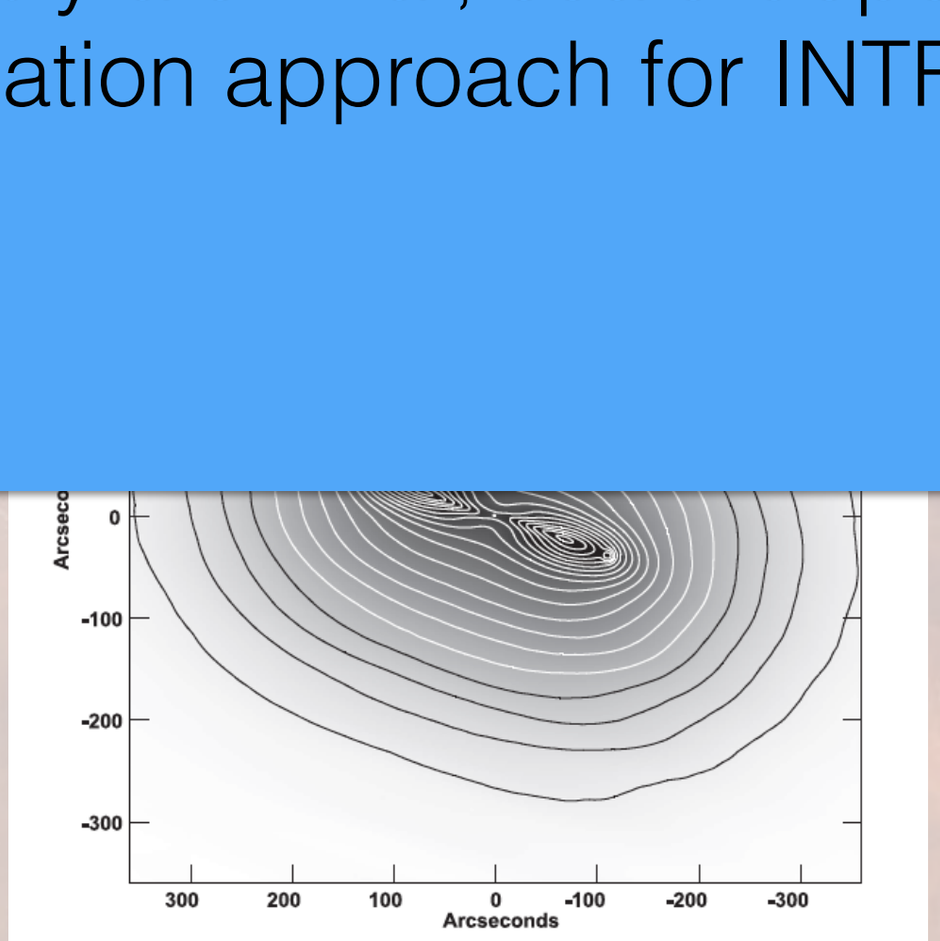


Cotton et al. 2017

Further documentation on feather and the OBIT implementation



We could try a similar, scale-dependent data combination approach for INTF+SD …



Cotton et al. 2017

Informed cleaning and subsequent feathering

tclean

startmodel

Name of starting model image
The contents of the supplied starting model image will be copied to the imagename.model before the run begins.

example : startmodel = 'singledish.im'

For deconvolver='mtmfs', one image per Taylor term must be provided. example : startmodel = ['try.model.tt0', 'try.model.tt1']
startmodel = ['try.model.tt0'] will use a starting model only for the zeroth order term. startmodel = ['','try.model.tt1'] will use a starting model only for the first order term.

This starting model can be of a different image shape and size from what is currently being imaged. If so, an image regrid is first triggered to resample the input image onto the target coordinate system.

A common usage is to set this parameter equal to a single dish image

Negative components in the model image will be included as is.

[Note : If an error occurs during image resampling/regridding, please try using task imregrid to resample the starting model image onto a CASA image with the target shape and coordinate system before supplying it via startmodel]

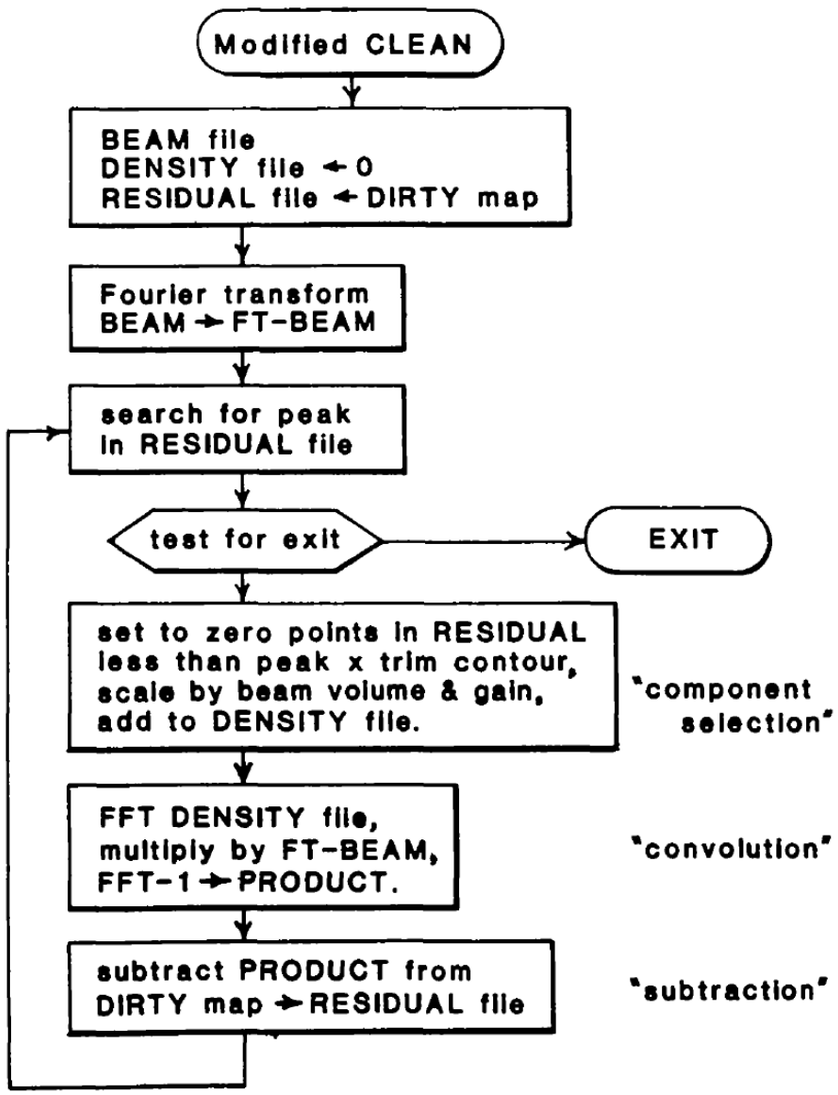
allowed: any

Default:

Steer cleaning in miriad

see Jens' talk for more detailed discussions

D. G. Steer et al.: CLEAN algorithm enhancements



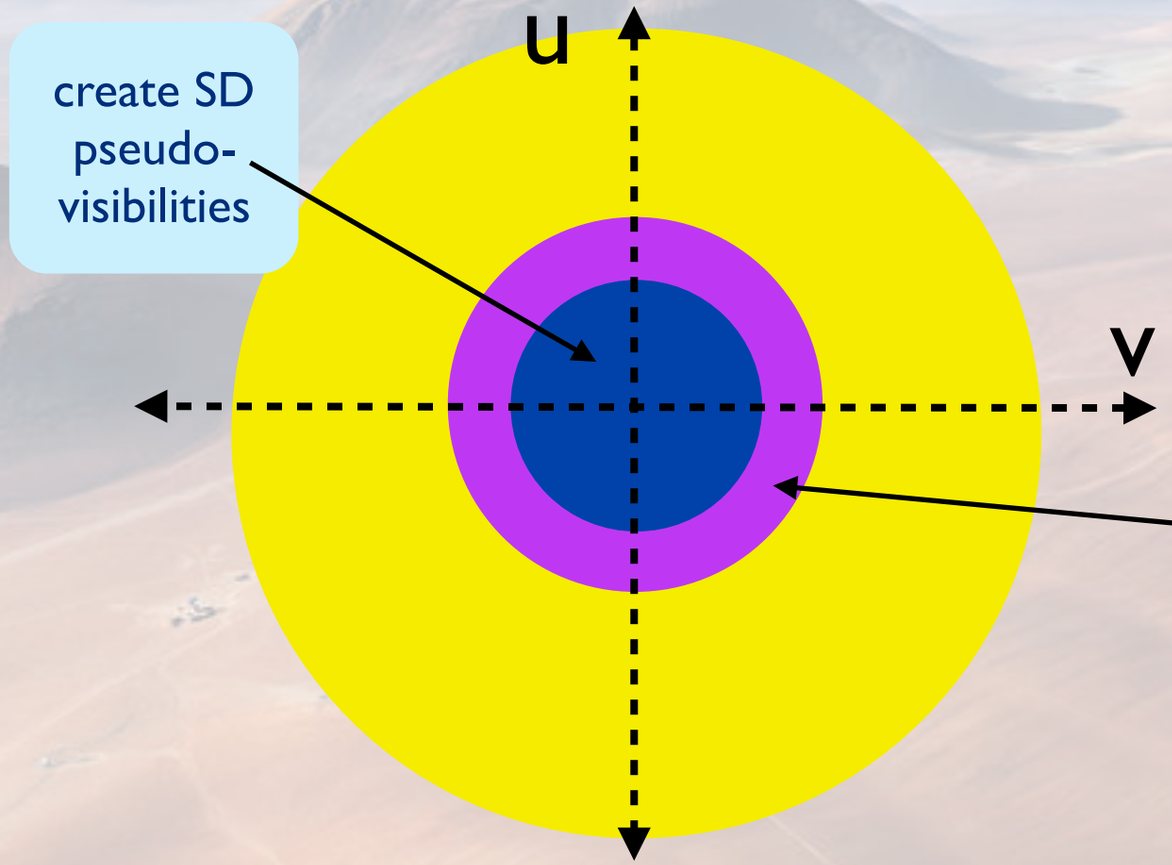
Steer et al. 1984

miriad example

```
clean% inp
  Task:    clean
  map       = "dirtymap.map"
  beam      = "dirtybeam.beam"
  model     = "anypriormodel.map"
  out       = "output_inJyperpix.map"
  gain      =
  options   =
  cutoff    =
  niters    = 100
  region    =
  phat      =
  minpatch  =
  speed     =
  mode      = "steer"
  clip      = 0.8
```

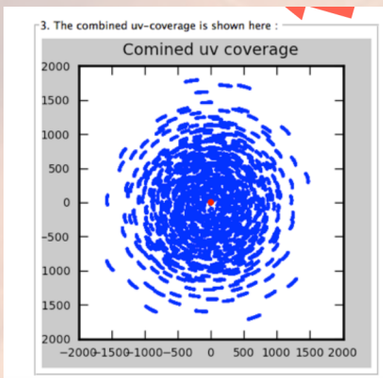
Combination before deconvolution

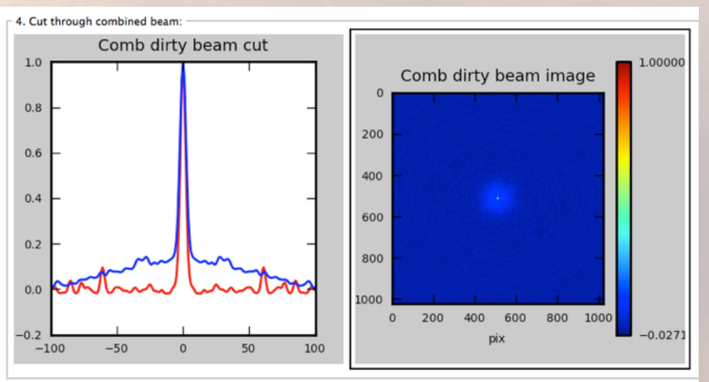
INTF SD OVERLAP



SD Pseudo-visibilities:

SD data are de-convolved and then transferred to the visibility plane to fill the inner uv-plane region. The combined Intf and SD data are then jointly de-convolved.

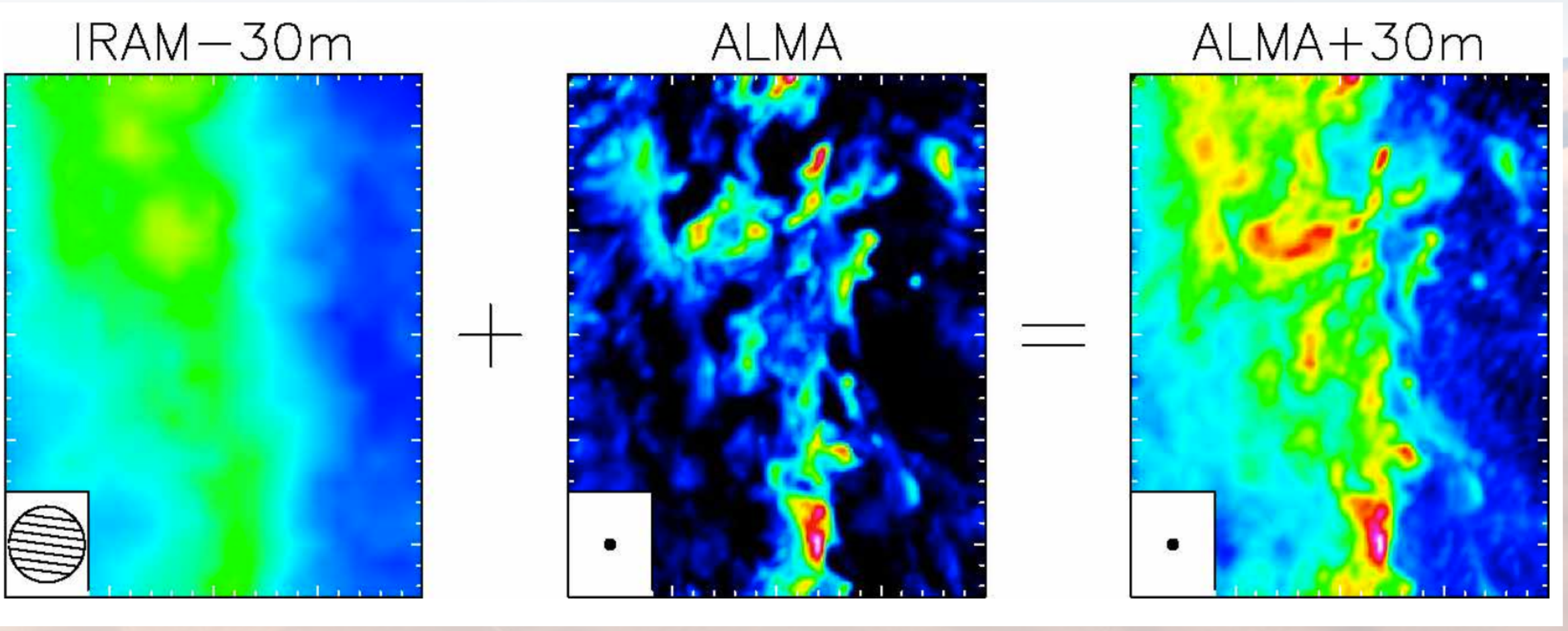




<u>for the best combination approach you need:</u>

- **a large overlap region to do the cross-calibration in**
- **matching sensitivity between the interferometer and the single-dish data in the overlap region**
- **SD map needs to be de-convolved before creating the pseudo-visibilities and INTF pb needs to be taken into account**
- **create a gaussian-like distribution of the SD pseudo-visibilities (or other weighting criteria)**

Combination in the uv-plane: the GILDAS/MAPPING approach



Goicoechea et al. 2016

image: courtesy of Edwige Chapillon

http://www.iram.fr/IRAMFR/GILDAS

Combination in the uv-plane: the GILDAS/MAPPING approach

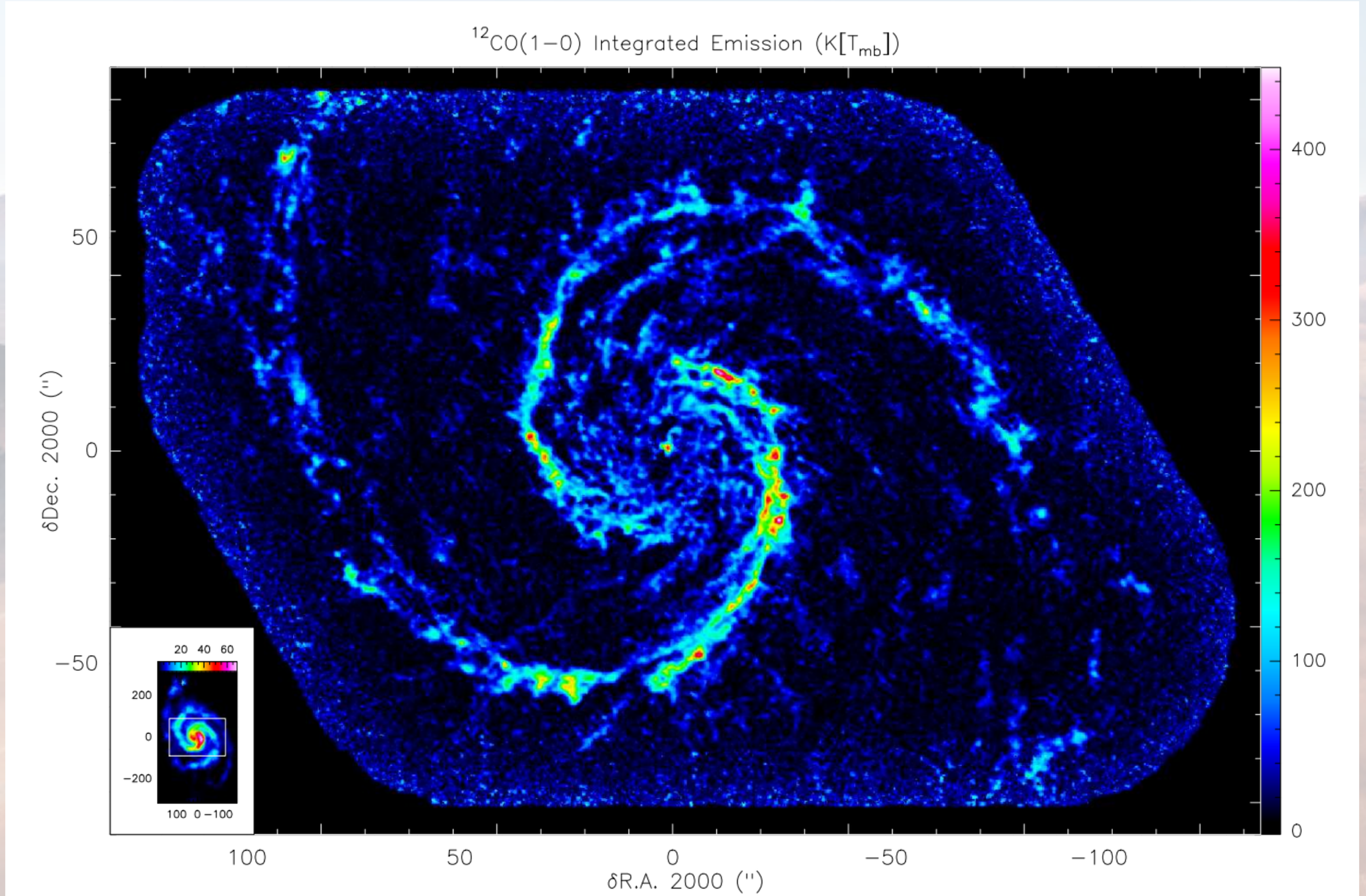
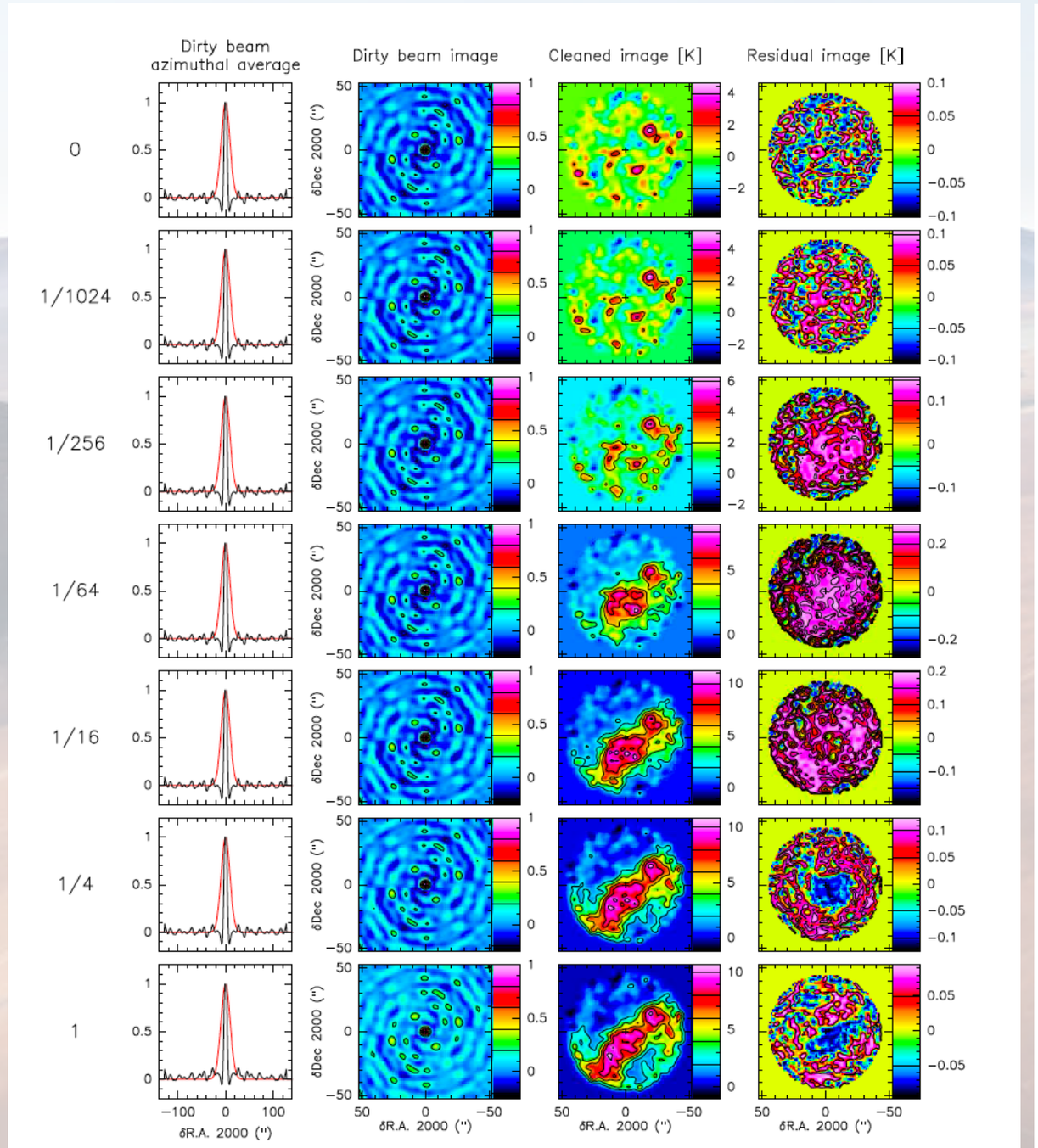
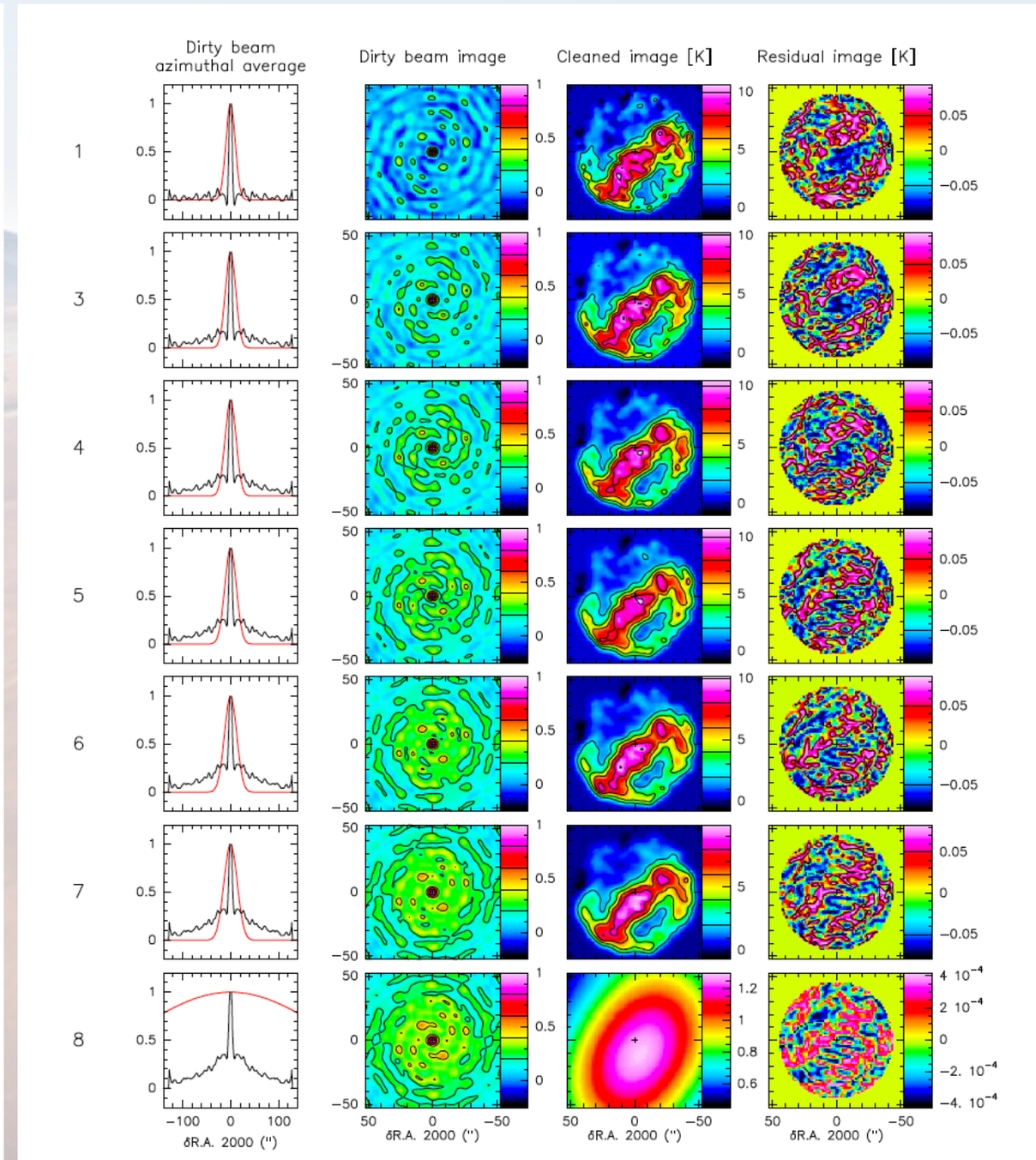


Figure 1. ^{12}CO (1–0) integrated emission of the inner $\sim$ 10 kpc $\times$ 6 kpc of the galaxy NGC 5194 (M51a). The coordinate offsets are relative to the nucleus of NGC 5194 (see Table 1). This image results from the joint deconvolution of the IRAM-30m single-dish and Plateau de Bure interferometer data sets. The image inserted at the bottom left is the ^{12}CO (1–0) integrated emission of the full M51 system (*i.e.*, NGC 5194 + NGC 5195) as observed by the IRAM-30m telescope. The white horizontal rectangle shows the PAWS field of view. The images were scaled such that the angular resolution of both data sets occupies the same size on paper. In other words, the PAWS image shows the center of the small image zoomed by a factor of 21.

Pety et al. 2013

Relative weights in uv-plane

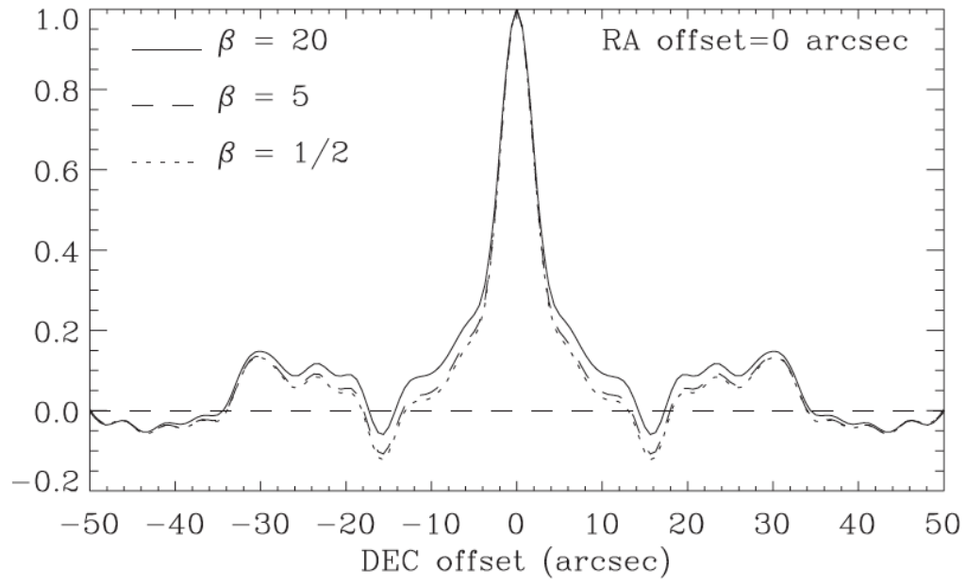


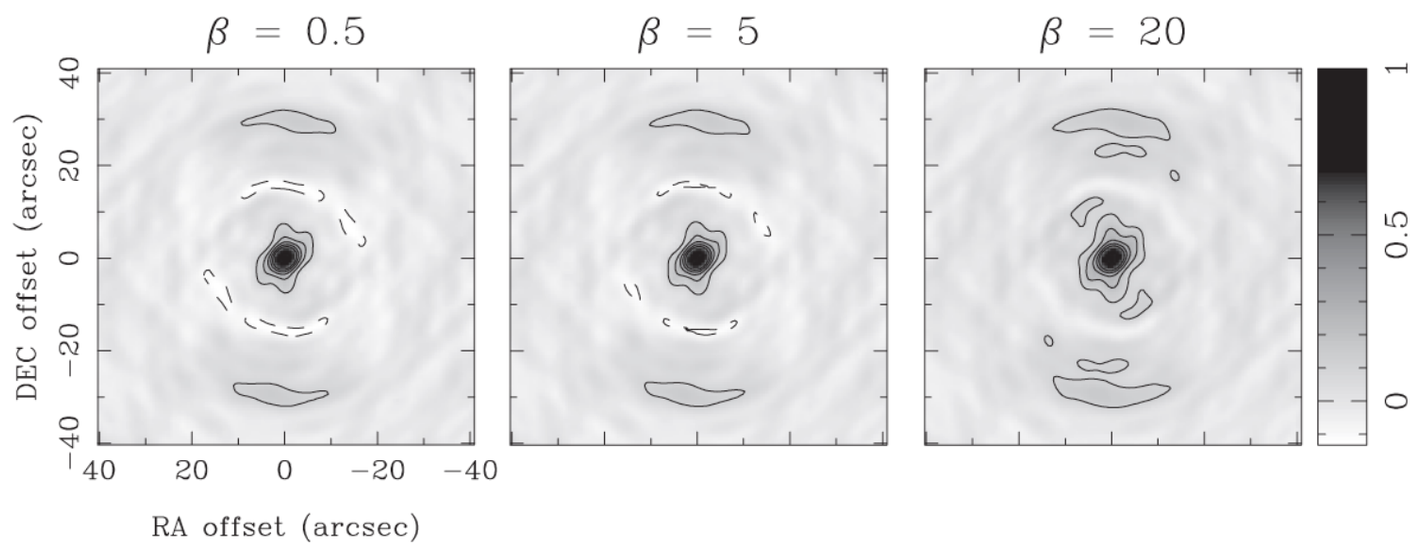


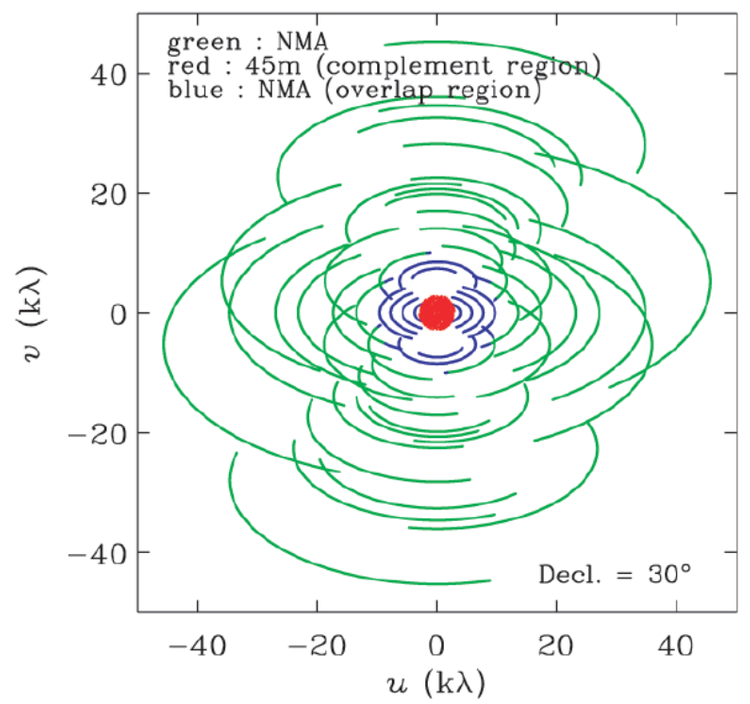
Rodriguez-Fernandez et al. 2008, IRAM Memo 2008-2

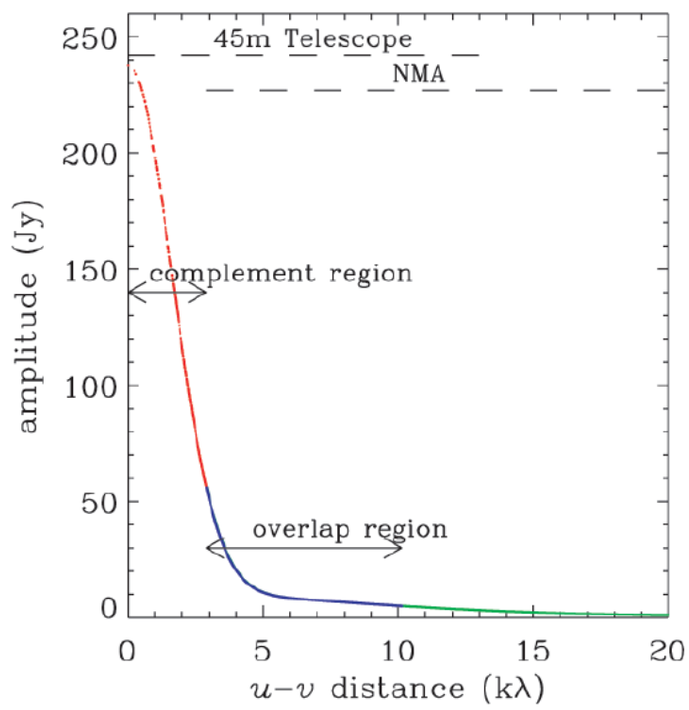
More on relative weights ...

Kurono et al. 2009









What about wavelets ?

a comment on wavelets came up in slack

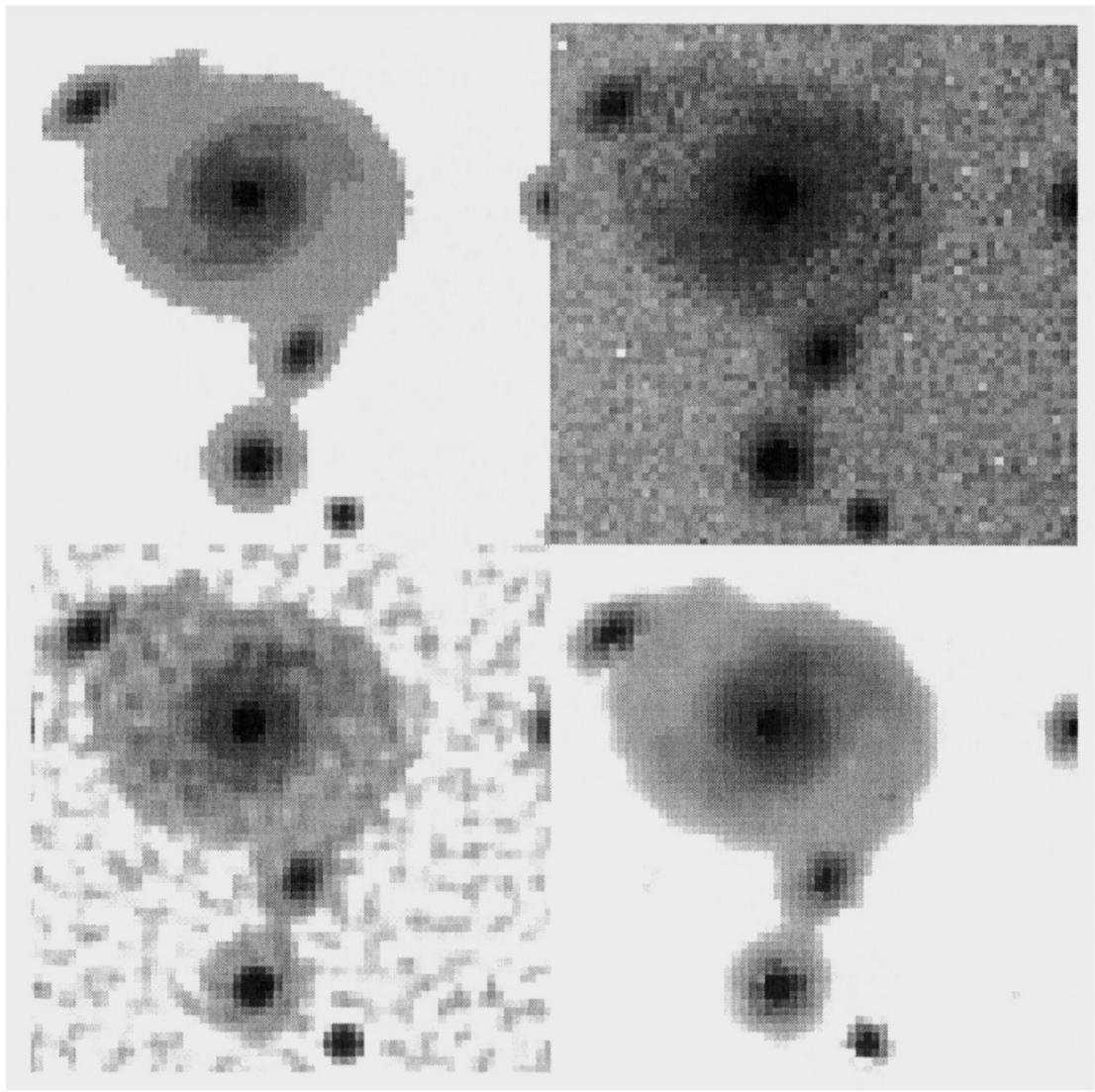


Fig. 1. Simulated object (upper left), blurred image (upper right), deconvolved image by MEM (lower left), and deconvolved image by Multiscale MEM (lower right)

Pantin & Stark 1996

More on MEM

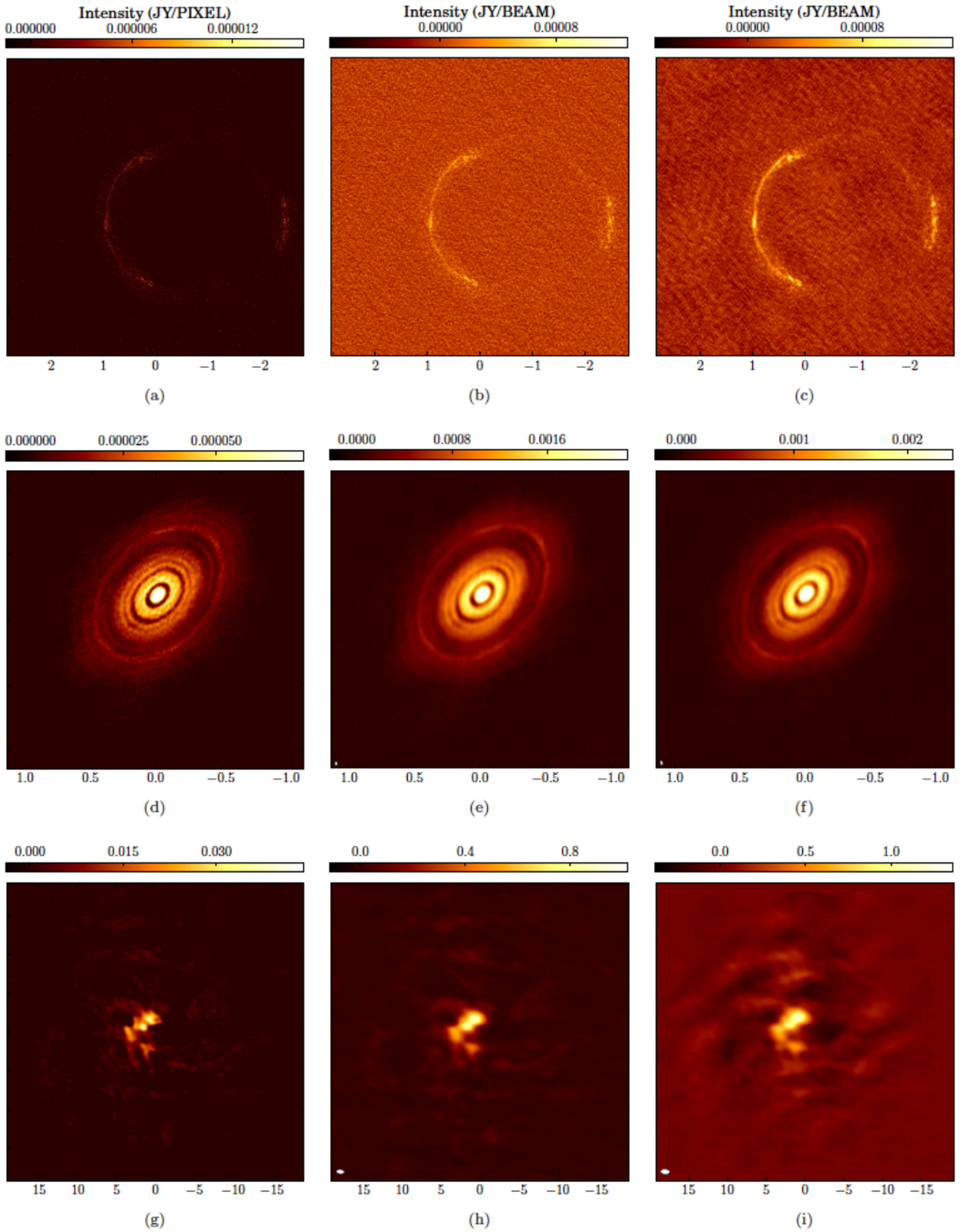


Figure 7: Image synthesis results using MEM and CLEAN. First column shows MEM model images. Second column shows MEM Restored images, and third column shows CLEAN images. (a)-(c) SDP 8.1 on Band 7, (d)-(f) HL Tau on Band 6. (g)-(i) Antennae Galaxies Northern mosaic on Band 7.

Carcamo et al. 2017

It would be worth looking
into INTF+SD MEM combination methods !

mathematically rigorous

gradient descent

sometimes hard to reach
convergence - need good rms estimate

miriad

```
Task:    mosmem
map       = "input.map"
beam      = "beam.map"
model     = "anypriormodel.map"
default   =
out       = "mosmemout.map"
niters    = 100
region    =
measure   = "gull"
tol       =
q         =
rmsfac    =
factor    =
flux      =
options   =
```

What about Bayesian approaches ?

joint Bayesian parametric modeling of the INTF and SD datasets

easiest for images with low data complexity

MEM could be a viable compromise as it
gives the most likely image, amongst many possible solutions,that is consistent with all the data

Question we should answer:
How do we quantify uncertainty in our data combination ?

The aim is not to find the “best-looking” solution but to find the most stable solution.

THANKS !