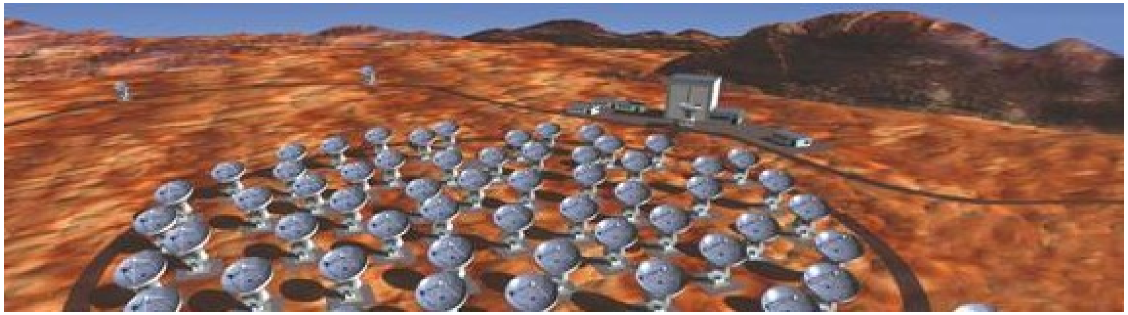
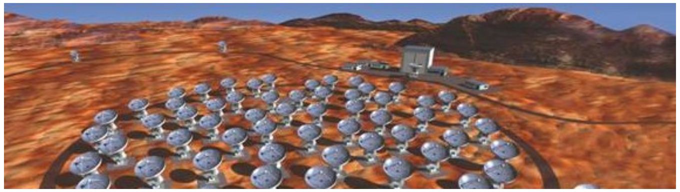


Imaging with CASA

Philippe Salomé

LERMA, Observatoire de Paris

Credits:
(Jérôme Pety, Frédéric Gueth, Crystal Brogan)



Imaging **Principles**

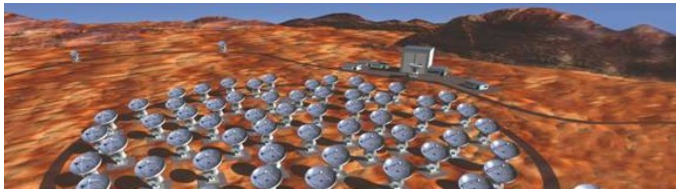
$$V_{ij}\ (b_{ij}) = 2D\ FT\{B_{primary}.I_{source}\}\ (b_{ij}) + Noise$$

- Irregular, limited sampling function:
– $S(u, v) = 1$ at (u, v) points where visibilities are measured
– $S(u, v) = 0$ elsewhere
- $B_{dirty} = 2D\ FT^{-1}\ \{S\}$
- $I_{meas} = 2D\ FT^{-1}\ \{S.V\} \rightarrow$ 1) Gridding + FFT to get I_{meas}

Fourier Transform Property #1:

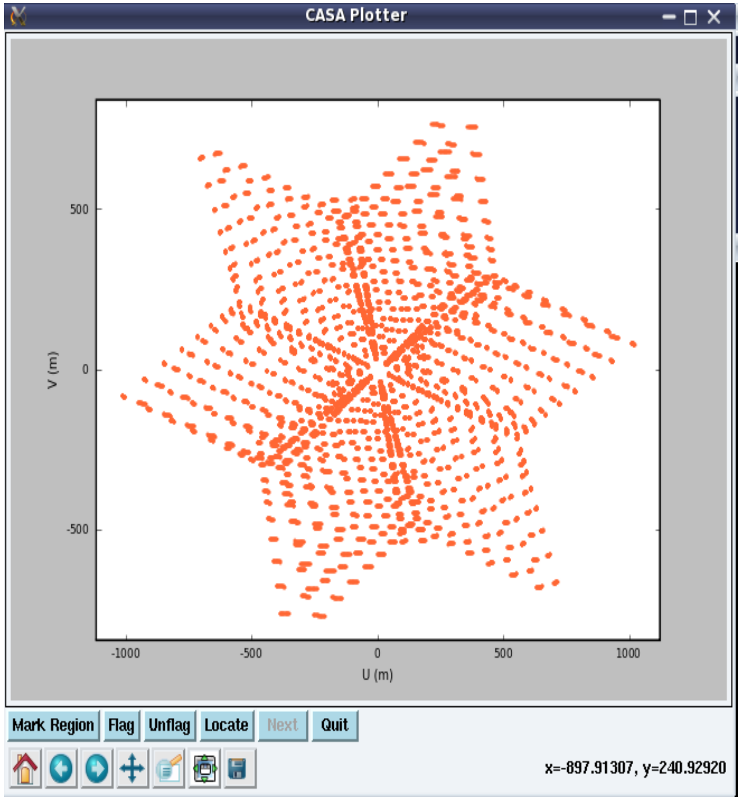
$I_{meas} = B_{dirty} * \{B_{primary}.I_{source}\} \rightarrow$ 2) Deconvolution to get I_{source}

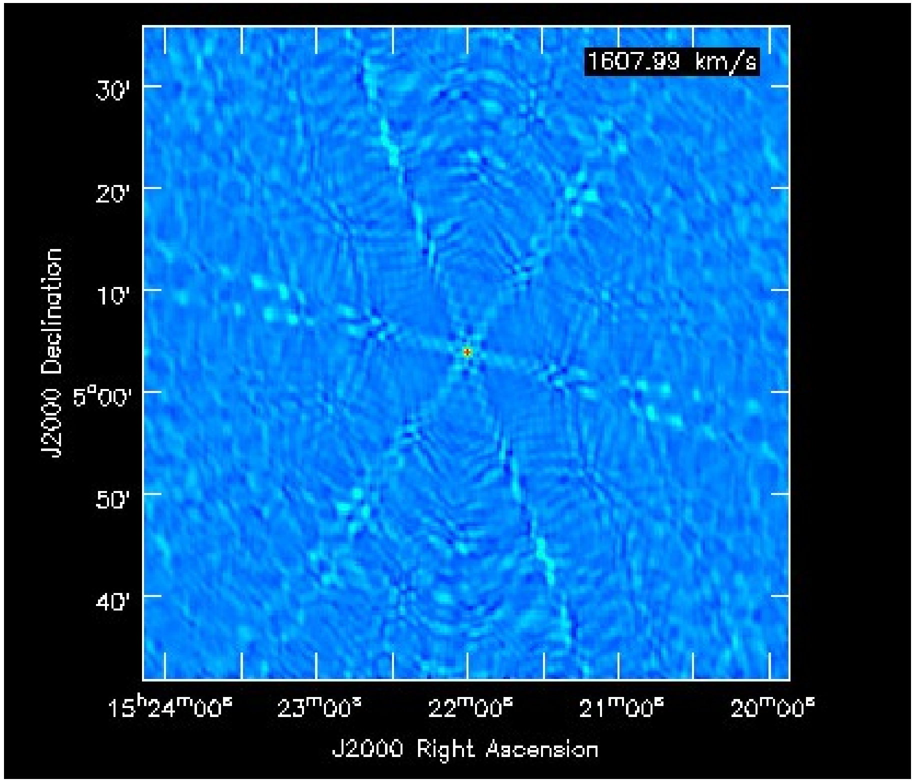
B_{dirty}: Point Spread Function (PSF) of the interferometer (i.e. if the source is a point, then $I_{meas} = I_{tot}.B_{dirty}$).

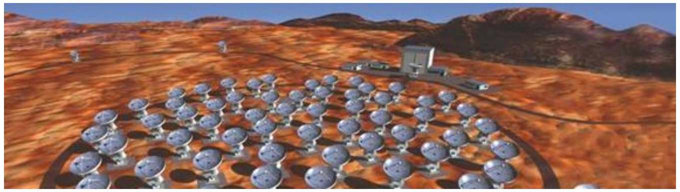


Imaging
Principles mapping

- From uv-plane to image plane → build a dirty image

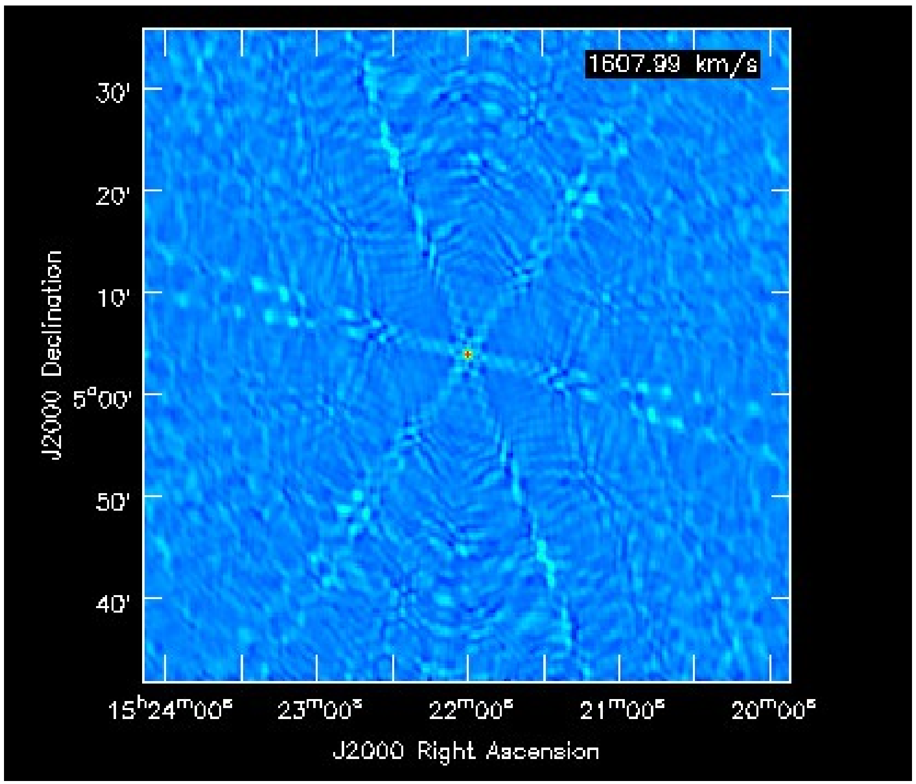


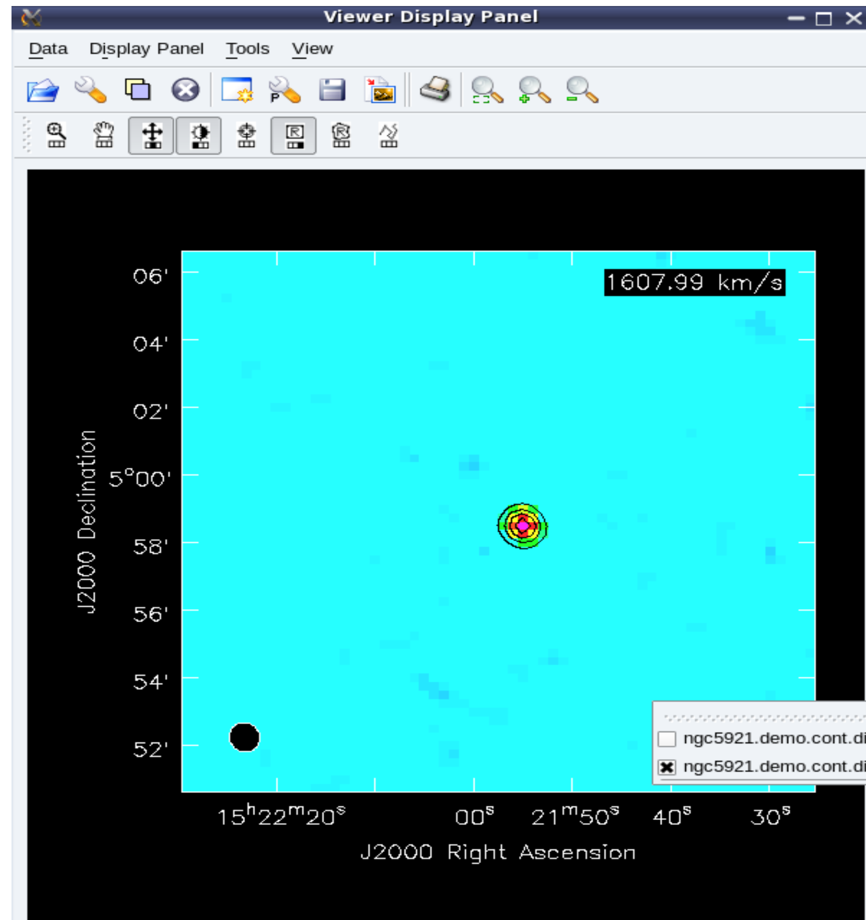


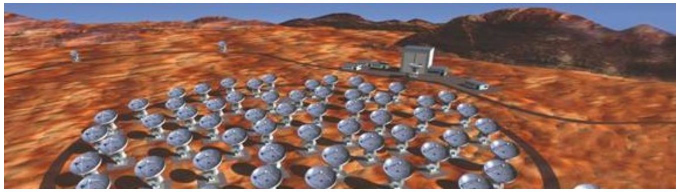


Imaging
Principles clean

- From dirty image to clean image (replace the dirty psf by a cleaned one : without sidelobs)







Imaging
CASA

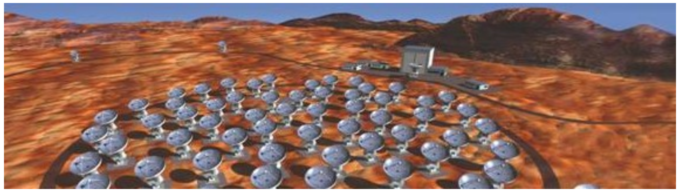
- Use the clean() task to both the image and the clean

- To build the dirty image do niter=0 (see later)

```
----------> inp(clean)
#  clean :: Invert and deconvolve images with selected algorithm
vis                 = 'ngc5921.demo.ms' #  Name of input visibility file
imagename           = 'ngc5921.demo.cleanimg' #  Pre-name of output images
outlierfile         =         ''        #  Text file with image names, sizes, centers for outliers
field               =        '0'        #  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

mode                =  'channel'        #  Spectral gridding type (mfs, channel, velocity, frequency)
     nchan          =         46        #  Number of channels (planes) in output image; -1 = all
     start          =          5        #  Begin the output cube at the frequency of this channel in the MS
     width          =          1        #  Width of output channel relative to MS channel (# to average)
     interpolation  =  'nearest'        #  Spectral interpolation (nearest, linear, cubic)
     outframe       =         ''        #  velocity frame of output image

gridmode            =         ''        #  Gridding kernel for FFT-based transforms, default='' None
niter               =       6000        #  Maximum number of iterations
gain                =        0.1        #  Loop gain for cleaning
threshold           =        8.0        #  Flux level to stop cleaning, must include units: '1.0mJy'
psfmode             =    'clark'        #  Method of PSF calculation to use during minor cycles
imagermode          =         ''        #  Options: 'csclean' or 'mosaic', '', uses psfmode
multiscale          =         []        #  Deconvolution scales (pixels); [] = standard clean
interactive         =      False        #  Use interactive clean (with GUI viewer)
mask                = [108, 108, 148, 148] #  Cleanbox(es), mask image(s), and/or mask region(s)
imsize              = [256, 256]        #  x and y image size in pixels. Single value: same for both
cell                = [15.0, 15.0]      #  x & 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, QU,IQUV)
weighting           =   'briggs'        #  Weighting of uv (natural, uniform, briggs, ...)
     robust         =        0.5        #  Briggs robustness parameter
     npixels        =          0        #  number of pixels to determine uv-cell size 0=> field of view

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
calready            =       True        #  True required for self-calibration
async               =      False        #  If true the taskname must be started using clean(...)
```

Input / output files

```
-----------> inp(clean)
#  clean :: Invert and deconvolve images with selected algorithm
vis                 = 'ngc5921.demo.ms' #  Name of input visibility file
imagename           = 'ngc5921.demo.cleanimg' #  Pre-name of output images
outlierfile         =         ''        #  Text file with image names, sizes, centers for outliers
field               =        '0'        #  Field Name or id
spw                 =         ''        #  Spectral windows e.g. '0~3', '' 
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

mode                =  'channel'        #  Spectral gridding type (mfs, channel, velocity, frequency)
     nchan          =         46        #  Number of channels (planes) in output image; -1 = all
     start          =          5        #  Begin the output cube at the frequency of this channel in the MS
     width          =          1        #  Width of output channel relative to MS channel (# to average)
     interpolation  =  'nearest'        #  Spectral interpolation (nearest, linear, cubic)
     outframe       =         ''        #  velocity frame of output image

gridmode            =         ''        #  Gridding kernel for FFT-based transforms, default='' None
niter               =       6000        #  Maximum number of iterations
gain                =        0.1        #  Loop gain for cleaning
threshold           =        8.0        #  Flux level to stop cleaning, must include units: '1.0mJy'
psfmode             =    'clark'        #  Method of PSF calculation to use during minor cycles
imagermode          =         ''        #  Options: 'csclean' or 'mosaic', '', uses psfmode
multiscale          =         []        #  Deconvolution scales (pixels); [] = standard clean
interactive         =      False        #  Use interactive clean (with GUI viewer)
mask                = [108, 108, 148, 148] #  Cleanbox(es), mask image(s), and/or mask region(s)
imsize              = [256, 256]        #  x and y image size in pixels. Single value: same for both
cell                = [15.0, 15.0]      #  x & 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, QU,IQUV)
weighting           =   'briggs'        #  Weighting of uv (natural, uniform, briggs, ...)
     robust         =        0.5        #  Briggs robustness parameter
     npixels        =          0        #  number of pixels to determine uv-cell size 0=> field of view

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
calready            =       True        #  True required for self-calibration
async               =      False        #  If true the taskname must be started using clean(...)
```



Imaging
Dirty map

Spatial Parameters

- Field, scan, antenna, uvrange

Spectral Parameters

- Start with selection parameters

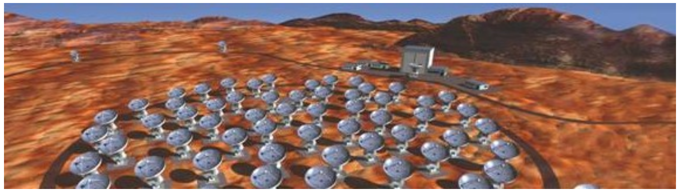
Select a spectral window and some channels with the spw parameter *spw*='0:0~10,1:20~30,2:1;2;3'

- How do I define the final spectral resolution ?

Mode: mfs (continuum), channel or velocity (emission line: nchan, start, width ...)

Spatial and Spectral selections

```
----------> inp(clean)
#  clean :: Invert and deconvolve images with selected algorithm
vis                 = 'ngc5921.demo.ms' #  Name of input
imagename           = 'ngc5921.demo.cleanimg' #  Pre-name
outlierfile         =        ''          #  Text file with
field               =       '0'          #  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

mode                = 'channel'          #  Spectral gridding type (mfs, channel, velocity, frequency)
     nchan          =        46          #  Number of channels (planes) in output image; -1 = all
     start          =         5          #  Begin the output cube at the frequency of this channel in the MS
     width          =         1          #  Width of output channel relative to MS channel (# to average)
     interpolation  = 'nearest'          #  Spectral interpolation (nearest, linear, cubic)
     outframe       =        ''          #  velocity frame of output image

gridmode            =        ''          #  Gridding kernel for FFT-based transforms, default='' None
niter               =      6000          #  Maximum number of iterations
gain                =       0.1          #  Loop gain for cleaning
threshold           =       8.0          #  Flux level to stop cleaning, must include units: '1.0mJy'
psfmode             =   'clark'          #  Method of PSF calculation to use during minor cycles
imagermode          =        ''          #  Options: 'csclean' or 'mosaic', '', uses psfmode
multiscale          =        []          #  Deconvolution scales (pixels); [] = standard clean
interactive         =     False          #  Use interactive clean (with GUI viewer)
mask                = [108, 108, 148, 148] #  Cleanbox(es), mask image(s), and/or mask region(s)
imsize              = [256, 256]         #  x and y image size in pixels. Single value: same for both
cell                = [15.0, 15.0]       #  x & 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, QU,IQUV)
weighting           =  'briggs'          #  Weighting of uv (natural, uniform, briggs, ...)
     robust         =       0.5          #  Briggs robustness parameter
     npixels        =         0          #  number of pixels to determine uv-cell size 0=> field of view

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
calready            =      True          #  True required for self-calibration
async               =     False          #  If true the taskname must be started using clean(...)
```



Imaging
Dirty map

Spatial Parameters (grid, fft)

- Start with selection parameters

Select a *field* (calibrator, target) and the *stoke* parameter (I,IV, QU,IQUV) you want to image

- What should be the cell size (sampling) ?

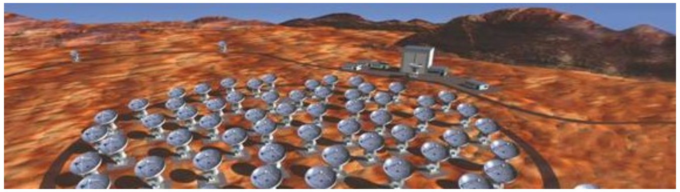
Cell : Between 1/3 and 1/5 of the synthesized beam to ease deconvolution

- What should be the map size ?

Imsize: At least twice the primary beam size or more and avoid bright sources near the edge of the image that would cause aliasing

```
----------> inp(clean)
#  clean :: Invert and deconvolve images with selected algorithm
vis                 = 'ngc5921.demo.ms' #  Name of input visibility file
imagename           = 'ngc5921.demo.cleanimg' #  Pre-name of output images
outlierfile         =         ''        #  Text file with image names, sizes, centers for outliers
field               =        '0'        #  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

mode                =  'channel'        #  Spectral gridding type (mfs, channel, velocity, frequency)
     nchan          =         46        #  Number of channels (planes) in output image; -1 = all
     start          =          5        #  Begin the output cube at the frequency of this channel in the MS
     width          =          1        #  Width of output channel relative to MS channel (# to average)
     interpolation  =  'nearest'        #  Spectral interpolation (nearest, linear, cubic)
     outframe       =         ''        #  velocity frame of output image

gridmode            =         ''        #  Gridding kernel for FFT-based transforms, default='' None
niter               =       6000        #  Maximum number of iterations
gain                =        0.1        #  Loop gain for cleaning
threshold           =        8.0        #  Flux level to stop cleaning, must include units: '1.0mJy'
psfmode             =    'clark'        #  Method of PSF calculation to use during minor cycles
imagermode          =         ''        #  Options: 'csclean' or 'mosaic', '', uses psfmode
multiscale          =         []        #  Deconvolution scales (pixels); [] =
interactive         =      False        #  Use interactive clean (with GUI vie
mask                = [108, 108, 148, 148] #  Cleanbox(es), mask image(s), and
imsize              = [256, 256]        #  x and y image size in pixels. Single value: same for both
cell                = [15.0, 15.0]      #  x & 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, QU,IQUV)
weighting           =   'briggs'        #  Weighting of uv (natural, uniform, briggs, ...)
     robust         =        0.5        #  Briggs robustness parameter
     npixels        =          0        #  number of pixels to determine uv-cell size 0=> field of view

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
calready            =       True        #  True required for self-calibration
async               =      False        #  If true the taskname must be started using clean(...)
```

Mapping parameters



Imaging
Clean Image

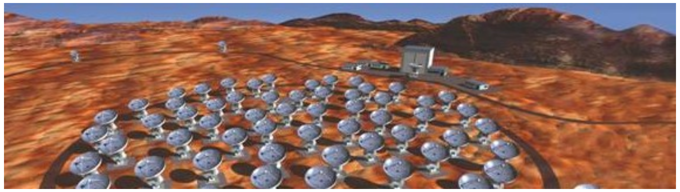
Cleaning methods

psfmode – algorithm used to calculate the point spread function (psf). Hogbom is robsut but slow. Clark is fast but unstable (when high sidelobes)

Imagermode ‘csclean’ – similar to ‘clark’ should be used for high dynamic range and is always used for mosaics. Better accuracy but slow (cf mx in gildas)

Polarization

‘hogbom’ currently only way to clean I, Q, U, & V independently. For polarization imaging ‘clark’ searches for peak in I2+V2+Q2+U2



Imaging
Clean Image

When do I stop cleaning ?

Stop cleaning when the residuals are noise like, and/or the clean has stopped converging (adding to cleaned flux)!

niter – Number of clean iterations to do. This can be useful when you are doing tests, but this parameter has NO physical meaning. Instead set to large number and let either threshold or do interactive to stop the cleaning.

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.

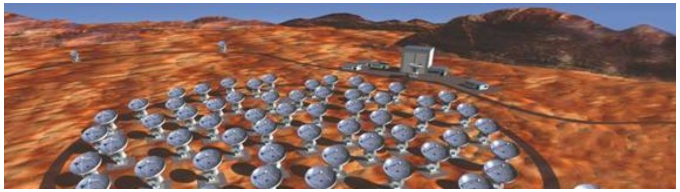
Note:

- To reach this limit the data must be well calibrated/flagged and suffer from no serious artifacts (resolved out extended structure/negative bowls, poor psf/uv-coverage, dynamic range limited etc).

- Do not set this blindly! Once you reach rms (whether close to theoretical or not), you are just picking noise up one place and putting it down in another

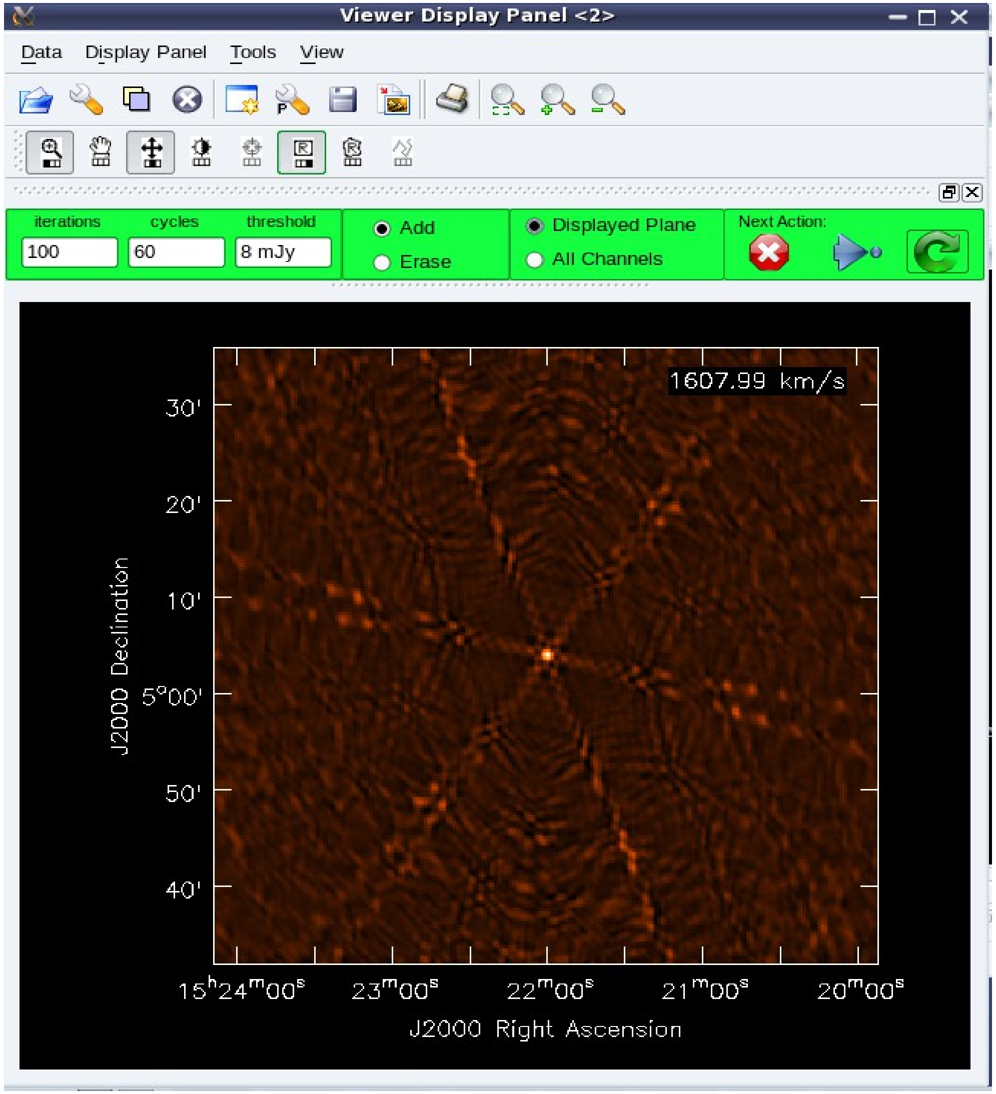
```
----------> inp(clean)
#  clean :: Invert and deconvolve images with selected algorithm
vis                 = 'ngc5921.demo.ms' #  Name of input visibility file
imagename           = 'ngc5921.demo.cleanimg' #  Pre-name of output images
outlierfile         =         ''        #  Text file with image names, sizes, centers for outliers
field               =        '0'        #  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

mode                =  'channel'        #  Spectral gridding type (mfs, channel, velocity, frequency)
     nchan          =         46        #  Number of channels (planes) in output image; -1 = all
     start          =          5        #  Begin the output cube at the frequency of this channel in the MS
     width          =          1        #  Width of output channel relative to MS channel (# to average)
     interpolation  =  'nearest'        #  Spectral interpolation (nearest, linear, cubic)
     outframe       =         ''        #  velocity frame of output image

gridmode            =         ''        #  Gridding kernel for FFT-based transforms, default = None
niter               =       6000        #  Maximum number of iterations
gain                =        0.1        #  Loop gain for cleaning
threshold           =        8.0        #  Flux level to stop cleaning, must include units: '1.0mJy'
psfmode             =    'clark'        #  Method of PSF calculation to use during minor cycles
imagermode          =         ''        #  Options: 'csclean' or 'mosaic', '', uses psfmode
multiscale          =         []        #  Deconvolution scales (pixels); [] = standard clean
interactive         =      False        #  Use interactive clean (with GUI viewer)
mask                = [108, 108, 148, 148] #  Cleanbox(es), mask image(s), and/or mask region(s)
imsize              = [256, 256]        #  x and y image size in pixels. Single value: same for both
cell                = [15.0, 15.0]      #  x & 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, QU,IQUV)
weighting           =   'briggs'        #  Weighting of uv (natural, uniform, briggs, ...)
     robust         =        0.5        #  Briggs robustness parameter
     npixels        =          0        #  number of pixels to determine uv-cell size 0=> field of view

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
calready            =       True        #  True required for self-calibration
async               =      False        #  If true the taskname must be started using clean(...)
```

Cleaning parameters



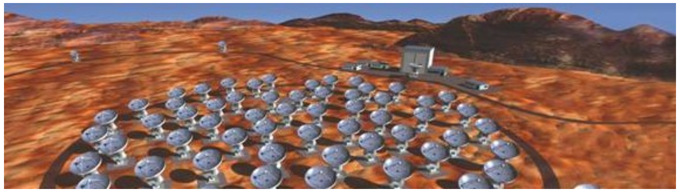
Imaging
Clean Image

Interactive Cleaning



Interactive Cleaning

```
-----------> inp(clean)
#  clean :: Invert and deconvolve images with selected algorithm
vis                 = 'ngc5921.demo.ms' #  Name of input visibility file
imagename           = 'ngc5921.demo.cleanimg' #  Pre-name of output images
outlierfile         =          ''        #  Text file with image names, sizes, centers for outliers
field               =         '0'        #  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

mode                =   'channel'        #  Spectral gridding type (mfs, channel, velocity, frequency)
     nchan          =          46        #  Number of channels (planes) in output image; -1 = all
     start          =           5        #  Begin the output cube at the frequency of this channel in the MS
     width          =           1        #  Width of output channel relative to MS channel (# to average)
     interpolation  =   'nearest'        #  Spectral interpolation (nearest, linear, cubic)
     outframe       =          ''        #  velocity frame of output image

gridmode            =          ''        #  Gridding kernel for FFT-based transforms, default='' None
niter               =        6000        #  Maximum number of iterations
gain                =         0.1        #  Loop gain for cleaning
threshold           =         8.0        #  Flux level to stop cleaning, must include units: '1.0mJy'
psfmode             =     'clark'        #  Method of PSF calculation to use du
imagermode          =          ''        #  Options: 'csclean' or 'mosaic', '',
multiscale          =          []        #  Deconvolution scales (pixels); [] = standard clean
interactive         =       False        #  Use interactive clean (with GUI viewer)
mask                = [108, 108, 148, 148] #  Cleanbox(es), mask image(s), and/or mask region(s)
imsize              = [256, 256]         #  x and y image size in pixels. Single value: same for both
cell                = [15.0, 15.0]       #  x & 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, QU,IQUV)
weighting           =    'briggs'        #  Weighting of uv (natural, uniform, briggs, ...)
     robust         =         0.5        #  Briggs robustness parameter
     npixels        =           0        #  number of pixels to determine uv-cell size 0=> field of view

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
calready            =        True        #  True required for self-calibration
async               =       False        #  If true the taskname must be started using clean(...)
```

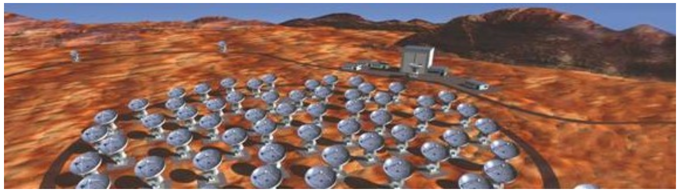


Imaging
Weighting

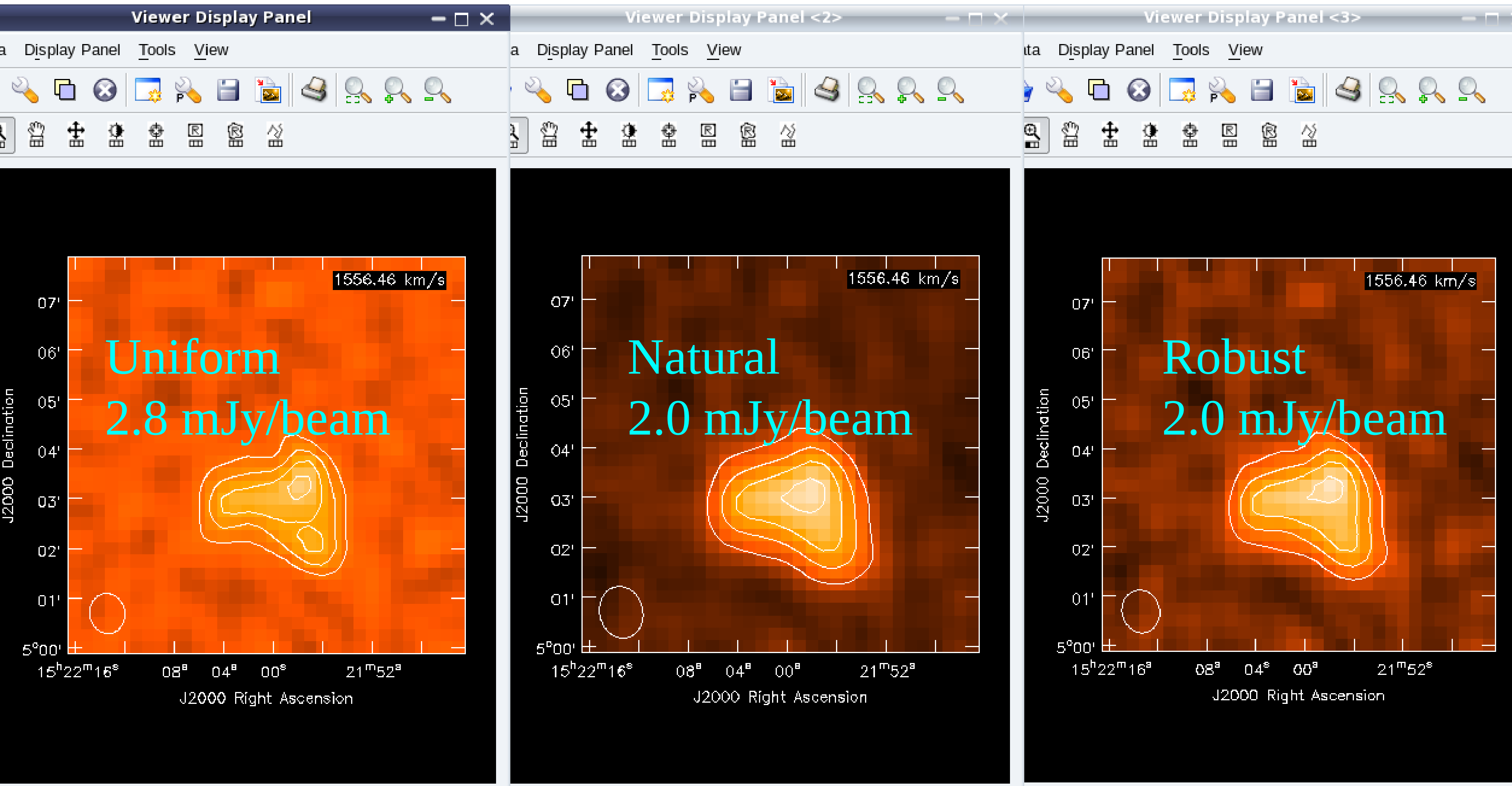
- *natura*l – lowest noise, poorest resolution, default
- *uniform* – highest noise, best resolution
- *briggs* – intermediate between natural and uniform

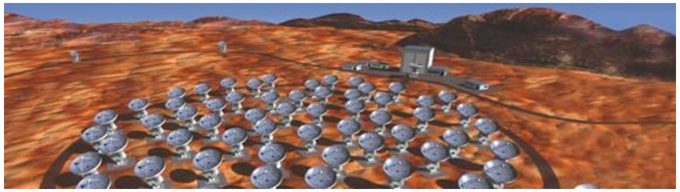
  - Default robust=0.0 is often a good choice, range -2 to 2, positive more towards natural, negative, more towards uniform

  - npixel number of pixels to determine uv-cell size, default 0 = imsize



Imaging
Weighting

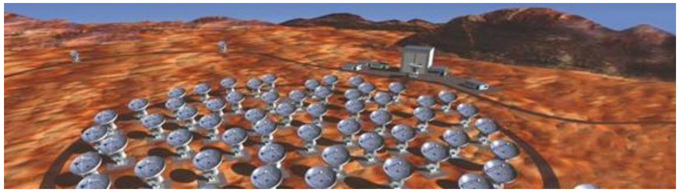




Imaging
UV-range

- An image can be dramatically changed by narrowing uv-range or applying outer uv-taper

- uvtaper=True
  - outertaper - default unit is meters
  - outertaper=['120arcsec']
  - outertaper=['5klambda','3klambda','45.0deg']
  - outertaper (klambda) / 200 = outertaper (arcsec) : 5klambda → 25 arcsec
  - Innertaper not yet implemented

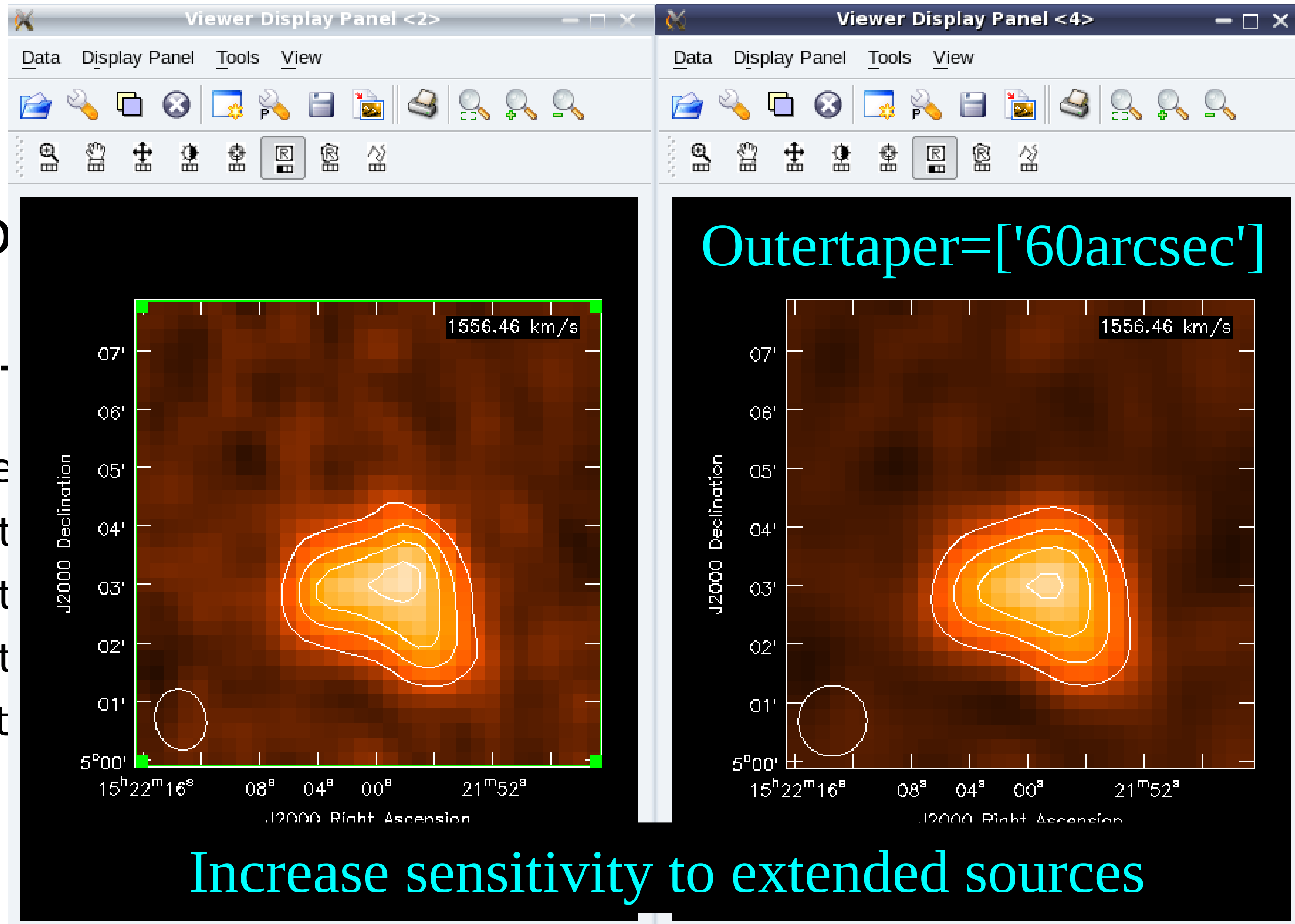


Imaging
UV-range

• An image ... uv-range or ap...

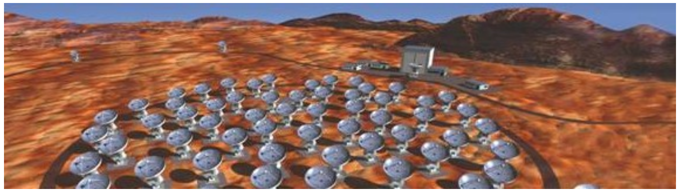
• uvtaper=...

- oute...
- outert...
- outert...
- outert... sec
- Innert...



```
----------> inp(clean)
#  clean :: Invert and deconvolve images with selected algorithm
vis                 = 'ngc5921.demo.ms' #  Name of input visibility file
imagename           = 'ngc5921.demo.cleanimg' #  Pre-name of output images
outlierfile         =         ''        #  Text file with image names, sizes, centers for outliers
field               =        '0'        #  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

mode                =  'channel'        #  Spectral gridding type (mfs, channel, velocity, frequency)
     nchan          =         46        #  Number of channels (planes) in output image; -1 = all
     start          =          5        #  Begin the output cube at the frequency of this channel in the MS
     width          =          1        #  Width of output channel relative to MS channel (# to average)
     interpolation  =  'nearest'        #  Spectral interpolation (nearest, linear, cubic)
     outframe       =         ''        #  velocity frame of output image

gridmode            =         ''        #  Gridding kernel for FFT-based transforms, default='' None
niter               =       6000        #  Maximum number of iterations
gain                =        0.1        #  Loop gain for cleaning
threshold           =        8.0        #  Flux level to stop cleaning, must include units: '1.0mJy'
psfmode             =    'clark'        #  Method of PSF calculation to use during minor cycles
imagermode          =         ''        #  Options: 'csclean' or 'mosaic', '', uses psfmode
multiscale          =         []        #  Deconvolution scales (pixels); [] = standard clean
interactive         =      False        #  Use interactive clean (with GUI viewer)
mask                = [108, 108, 148, 148] #  Cleanbox(es), mask image(s), and/or mask region(s)
imsize              = [256, 256]        #  x and y image size in pixels. Single value: same for both
cell                = [15.0, 15.0]      #  x & y cell size(s). Default unit arcsec.
phasecenter         =         ''        #  Image center: direction or field i
restfreq            =         ''        #  Rest frequency to assign to image
stokes              =        'I'        #  Stokes params to image (eg I,IV, Q
weighting           =   'briggs'        #  Weighting of uv (natural, uniform, briggs, ...)
     robust         =        0.5        #  Briggs robustness parameter
     npixels        =          0        #  number of pixels to determine uv-cell size 0=> field of view

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
calready            =       True        #  True required for self-calibration
async               =      False        #  If true the taskname must be started using clean(...)
```

Weighting parameters

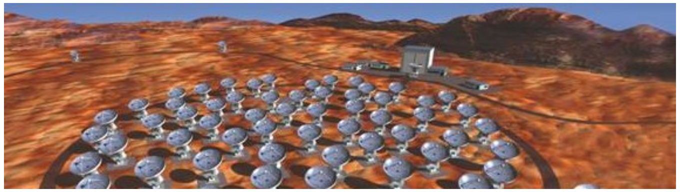


Imaging
Output files

Based on the imagename these files are created:

imagename.image – final cleaned (or dirty if niter=0 image)
imagename.psf – the point spread function of the beam, useful to check whether image artifacts are due to poor psf

imagename.model – an image of the clean components
imagename.residual – the residual image after subtracting clean components, useful to check if more cleaning is needed
imagename.flux – the primary beam response function – used to make a “flux correct image”, otherwise flux is only correct at the phase center(s). pbcor=T divides the .image by the .flux. Such images don’t look pretty because the noise at the edges are also increased, but flux densities should ONLY be calculated from pbcor’ed images.



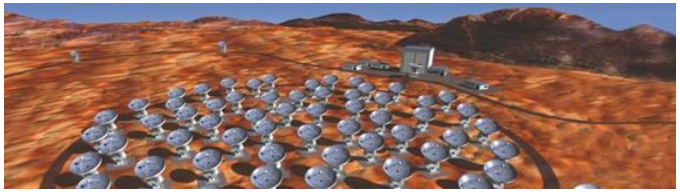
Imaging
Output files

Based on the imagename these files are created:

imagename.image – final cleaned (or dirty if niter=0 image)
imagename.psf – the point spread function of the beam, useful to check whether image artifacts are due to poor psf

Use the viewer() to see raster maps, contours...

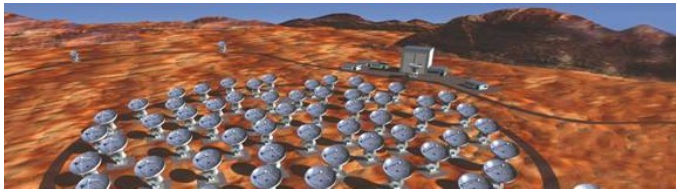
center(s). pbcor=T divides the .image by the .flux. Such images don’t look pretty because the noise at the edges are also increased, but flux densities should ONLY be calculated from pbcor’ed images.



Imaging
Analysis

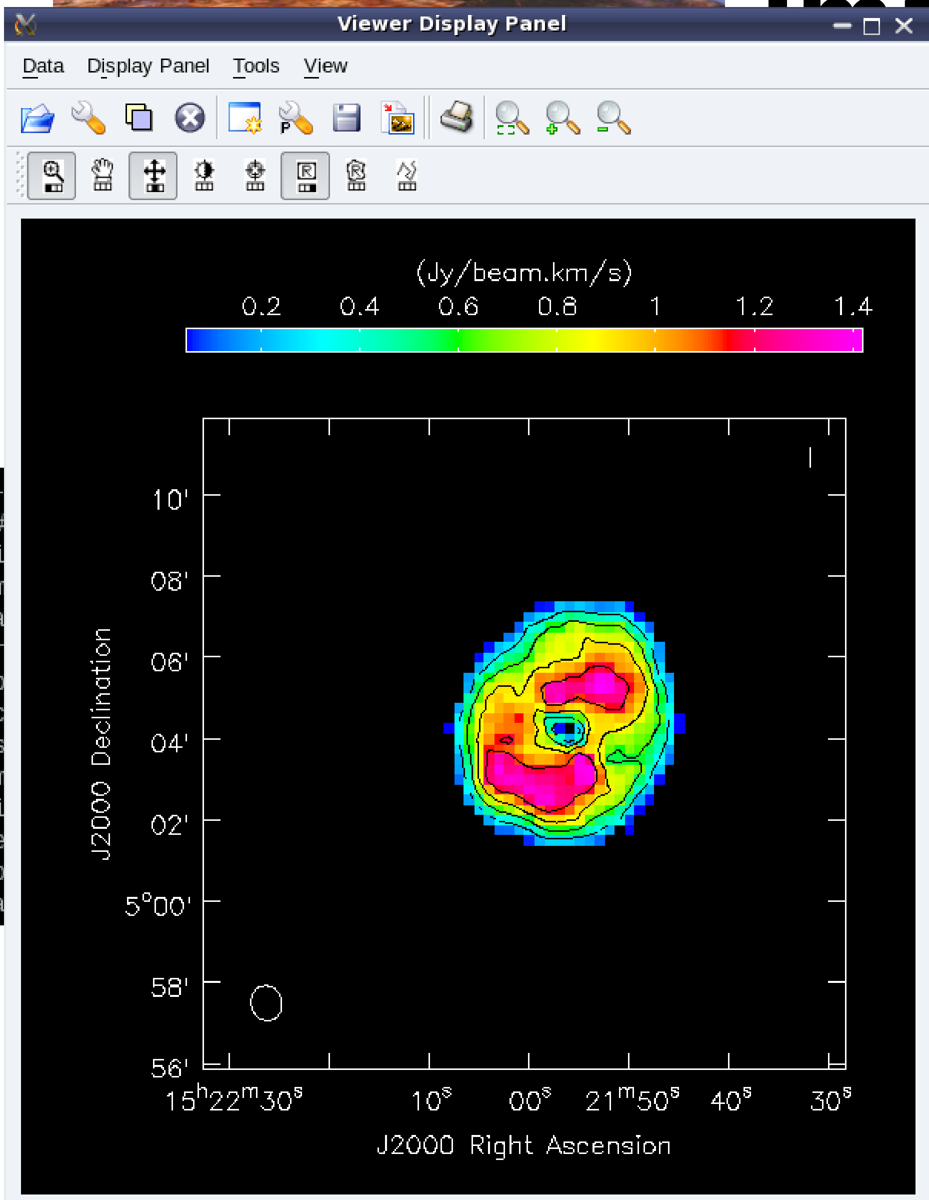
- task **imhead** - get and change image header information
- task **immoments** - computes moment images of spectral cube

- task **imstat** - return statistics on regions of image
- task **imval** - return values for pixel or region of image
- task **imfit** - fit a 2D Gaussian to the image

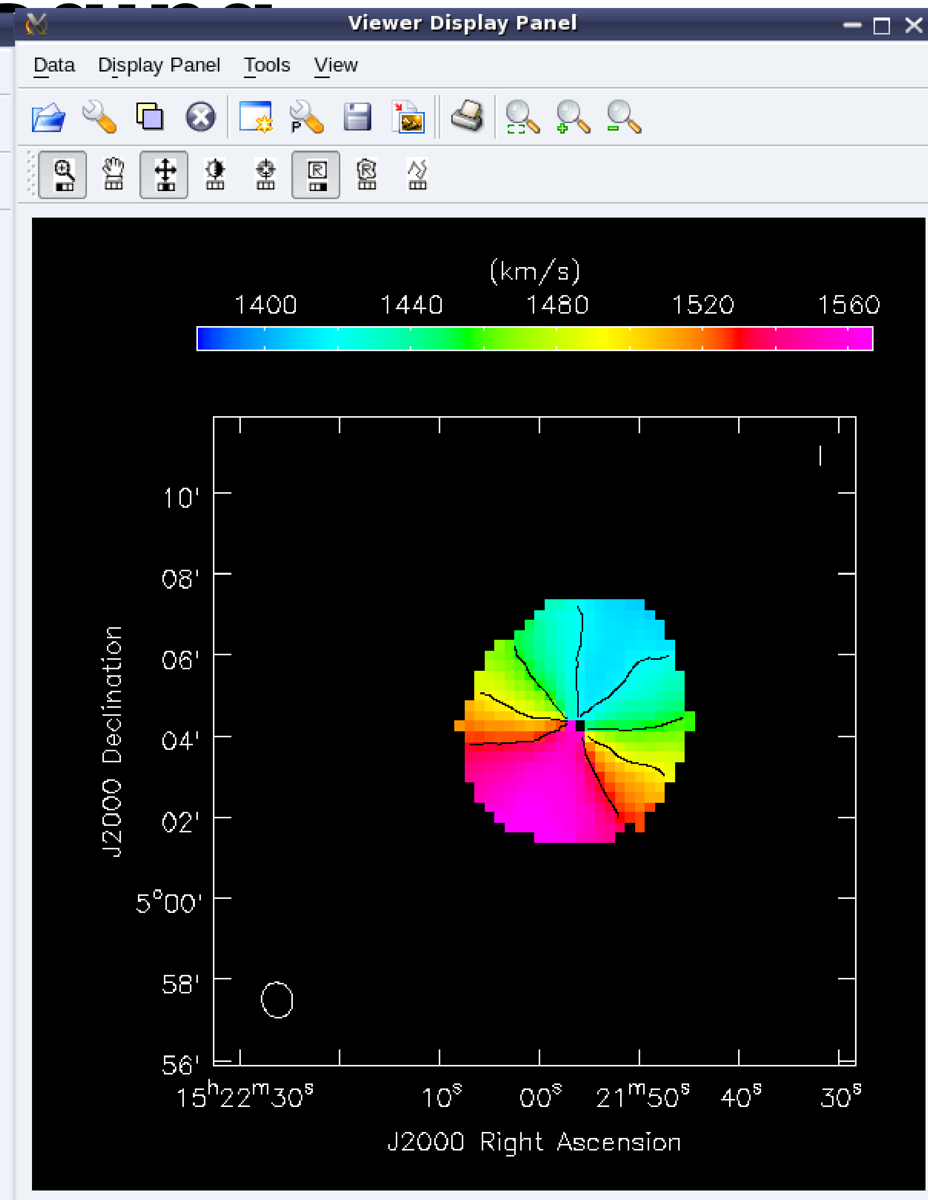


Imaging
Immoments

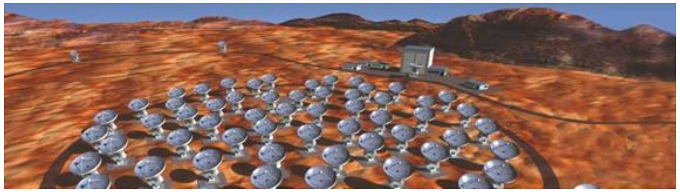
```
--ImMoments--
#  immoments :: Compute moments from an image
imagename           = 'ngc5921.demo.cleanimg.image' #  Name of the input image
moments             =     [0, 1]        #  List of moments you would like to compute
axis                = 'spectral'        #  The momement axis: ra, dec, lat, long, spectral, or stokes
region              =         ''        #  Image Region.  Use viewer
box                 =         ''        #  Select one or more box regions
chans               =         ''        #  Select the channel(spectral) range
stokes              =         ''        #  Stokes params to image (I,IV,IQU,IQUV)
mask                =         ''        #  mask used for selecting the area of the image to calculate the moments on
includepix          =         -1        #  Range of pixel values to include
excludepix          = [-100.0, 0.0089999999999999993] #  Range of pixel values to exclude
outfile             = 'ngc5921.demo.moments' #  Output image file name (or root for multiple moments)
async               =      False        #  If true the taskname must be started using immoments(...)
```



Viewer Display Panel
Data
Display Panel
Tools
View
(Jy/beam.km/s)
0.2
0.4
0.6
0.8
1
1.2
1.4
10'
08'
06'
04'
02'
5°00'
58'
56'
J2000 Declination
15h22m30s
10s
00s
21m50s
40s
30s
J2000 Right Ascension



Viewer Display Panel
Data
Display Panel
Tools
View
(km/s)
1400
1440
1480
1520
1560
10'
08'
06'
04'
02'
5°00'
58'
56'
J2000 Declination
15h22m30s
10s
00s
21m50s
40s
30s
J2000 Right Ascension



Imaging
Large scale

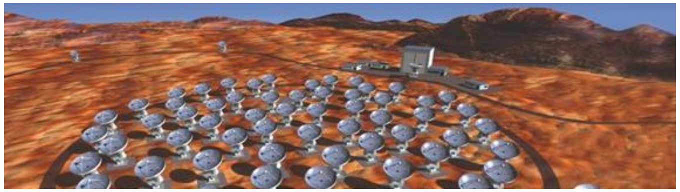
Large scale imaging (mosaic)

Widefield() or **clean()** with **imagermode='mosaic'**

Specific methods for ftt, weighting and cleaning of the different pointings observed that will make the full final image

Zero/short spacing (extended emission)

feather() – combine a single dish and interferometric image in the image-plane. If there is is good overlap in the UV-plane, and the relative calibration between the two is accurate this should work pretty well



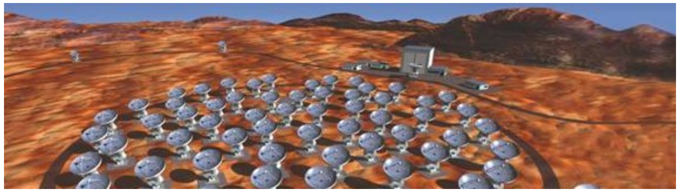
More ...

Exercices

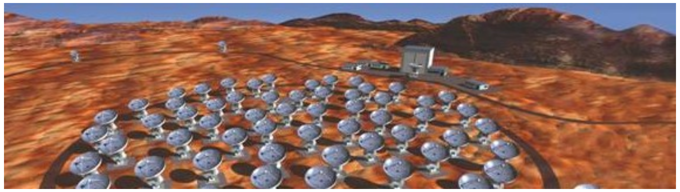
- Go back through the ngc5921_demo.py script

References

- NRAO Lectures
http://www.cv.nrao.edu/course/astr534/ERA.shtml
- IRAM schools (2010 Oct. 4th -8th , Grenoble, France)
http://www.iram.fr/IRAMFR/IS/school.htm



The end !



Imaging Principles

- From visibilities to images

$$I_{meas} = B_{drty} * (B_{prim} \times I_{source}) + N$$

I_{meas} : Dirty Map

B_{drty} : Dirty Beam

B_{prim} : Primary Beam

I_{source} : Sky brightness distribution

N : Noise