Imaging and Image Analysis

CASA
Common Astronomy Software Applications

Urvashi R.V.

National Radio Astronomy Observatory, Socorro, NM, USA

6th VLA Data Reduction Workshop (23 – 27 October, 2017)

Outline

- **Synthesis imaging as implemented in CASA** (tasks : **tclean, clean**)
  https://casa.nrao.edu/casadocs/casa-5.1.0/synthesis-imaging

  - Major and minor cycles
  - Gridding algorithms
  - Deconvolution algorithms
  - Types of Images
  - Iteration control
  - Masks for deconvolution
  - Runtime interactivity
  - Factors affecting computing cost
  - Task interface

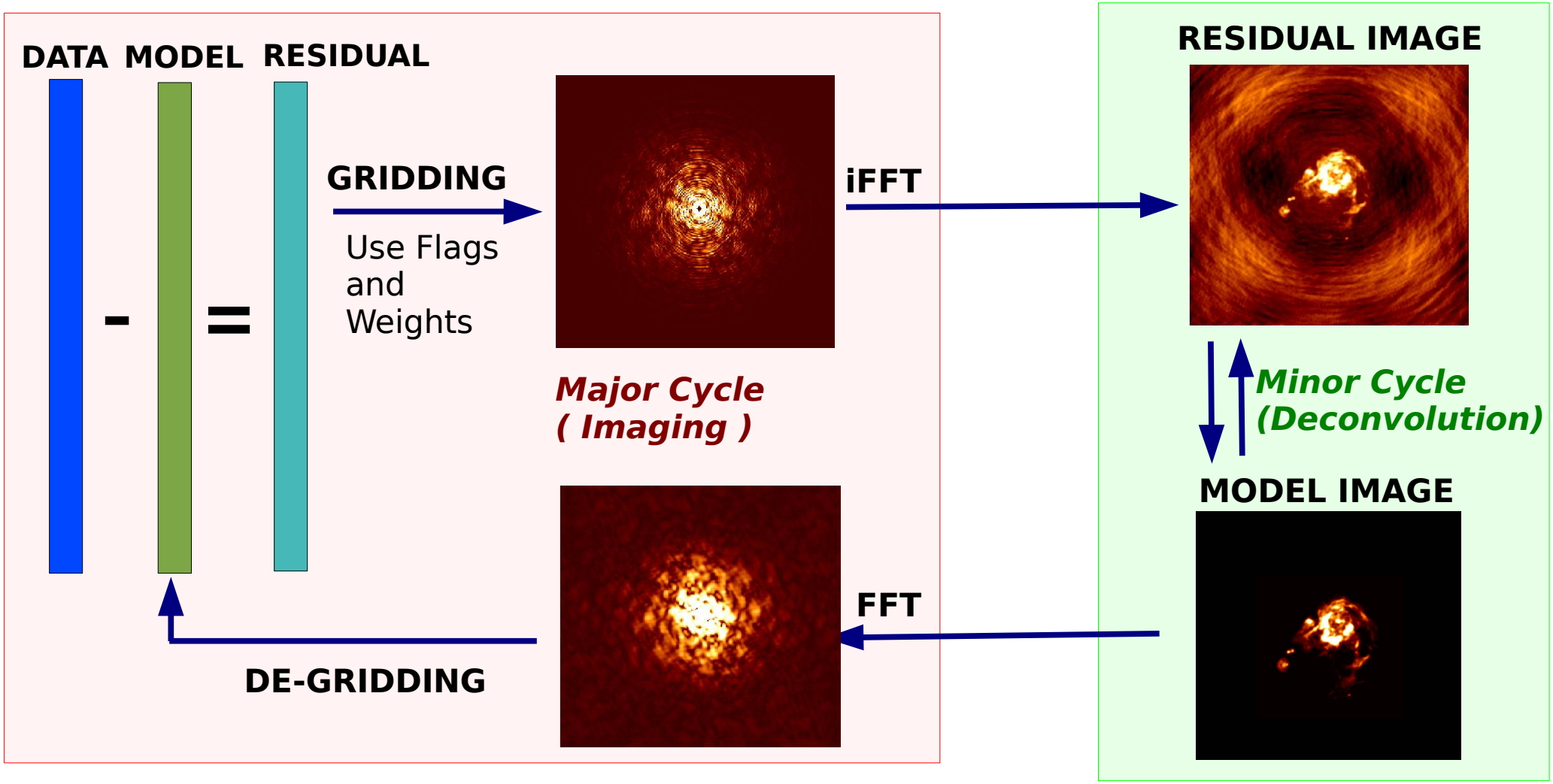
clean
- Older task
- Development frozen
- Task retained for backward compatibility of scripts
- Tools retained for various utility methods

tclean
- Newer task
- Currently developed/used
- Uses refactored imaging module
- All major features of clean plus several more
- Used by pipelines

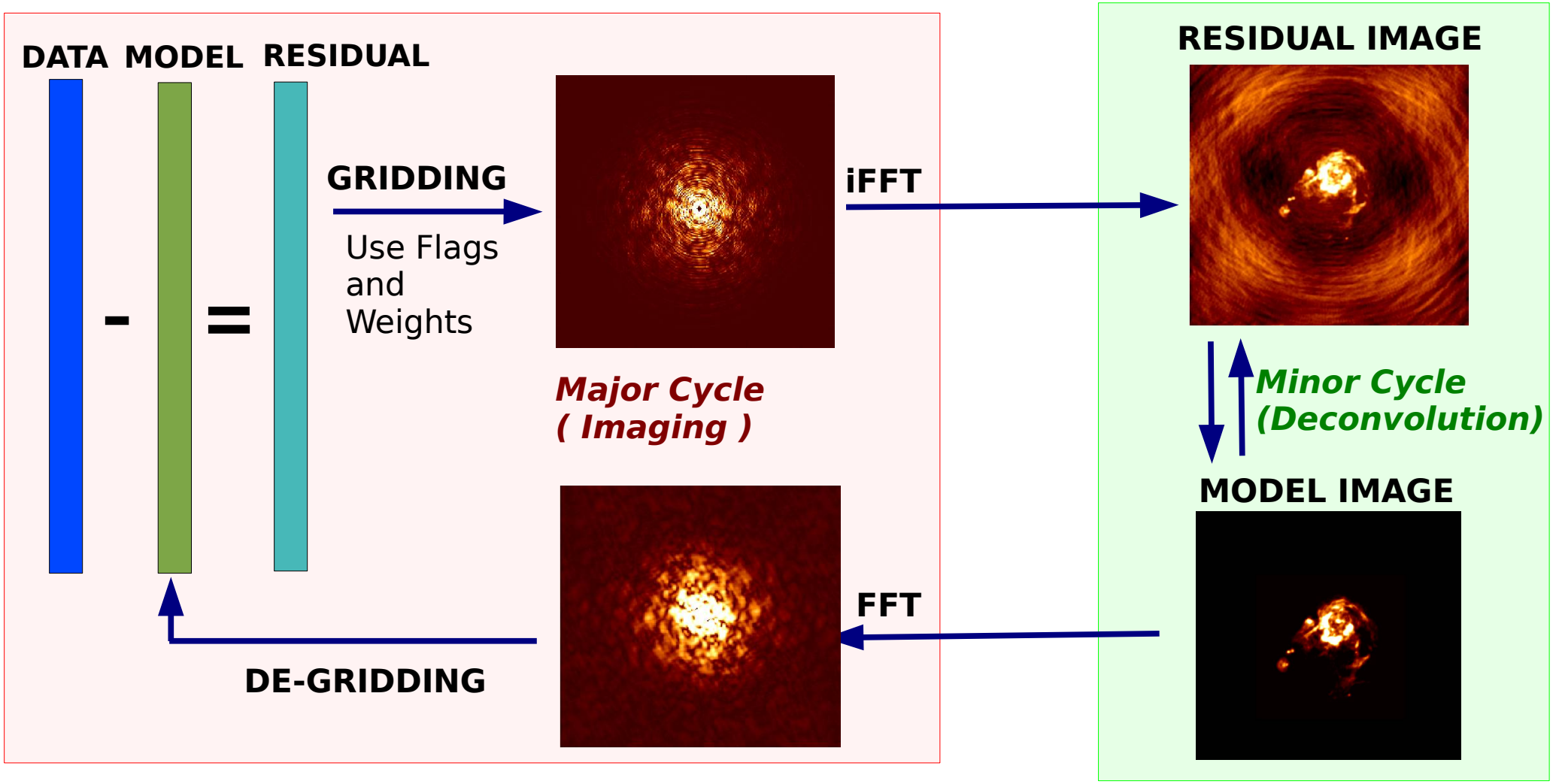
- **Image Viewing and Analysis** (task : **viewer** , 28 **image analysis** tasks)

  https://casa.nrao.edu/casadocs/casa-5.1.0/image-cube-visualization
  https://casa.nrao.edu/casadocs/casa-5.1.0/image-analysis

Iterative χ^2 minimization – Major and Minor Cycles



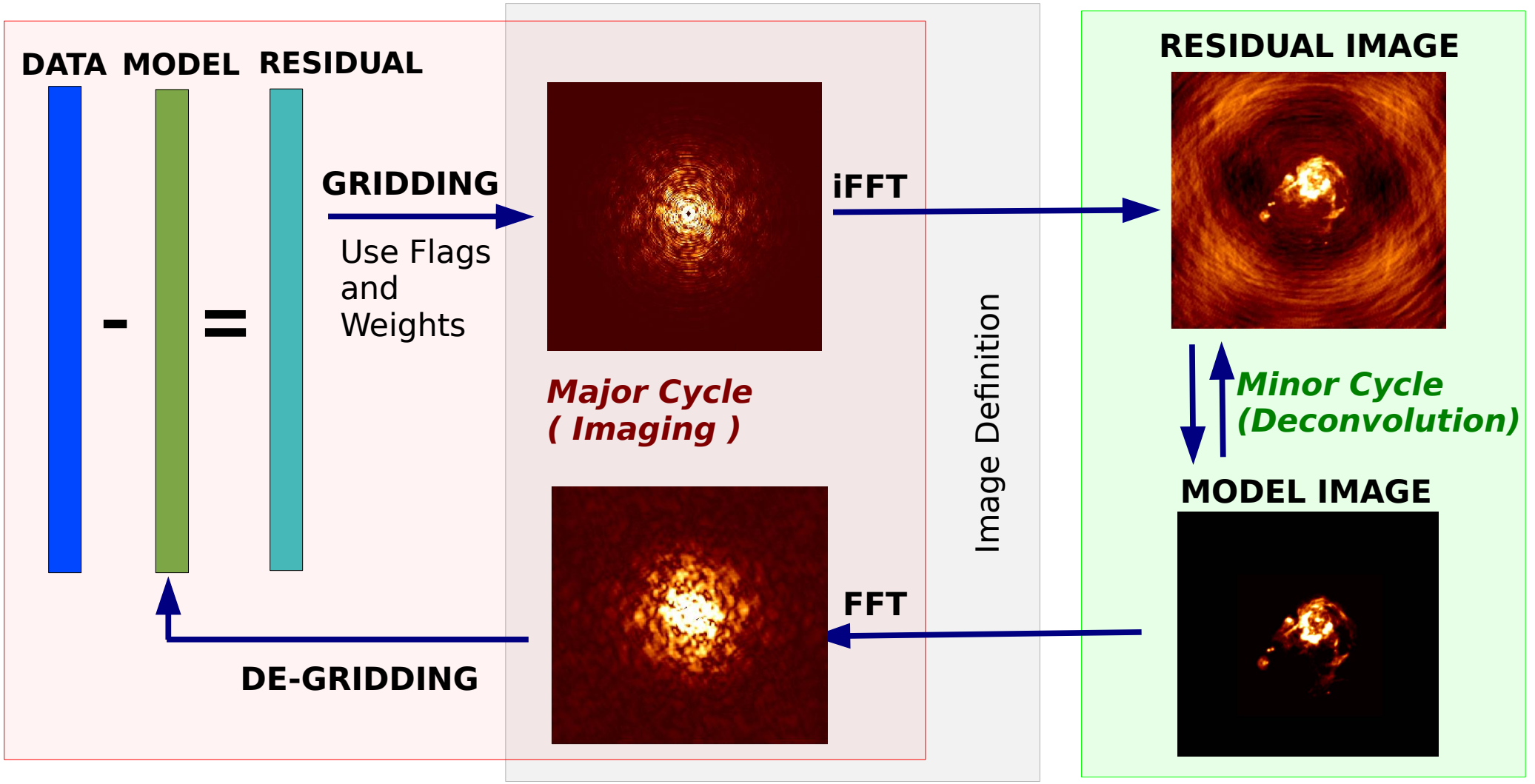
Iterative χ^2 minimization – Major and Minor Cycles



Standard gridding, W-Projection, (WB)-A-Projection, Joint Mosaics

Clean (Hogbom, Clark, MultiScale, MultiTerm, etc...)

Iterative χ^2 minimization – Major and Minor Cycles



Standard gridding, W-Projection, (WB)-A-Projection, Joint Mosaics

Cube, MFS, MT-MFS, Faceting, Stokes, Multi-Field

Clean (Hogbom, Clark, MultiScale, MultiTerm, etc...)

Imaging Algorithms (Weighting + Gridding)

Image = Weighted average of the data

$$= \frac{\text{sum of (weight x data)}}{\text{sum of (weight)}}$$

Gridding : Accumulate weighted visibilities on a regular grid : convolutional resampling

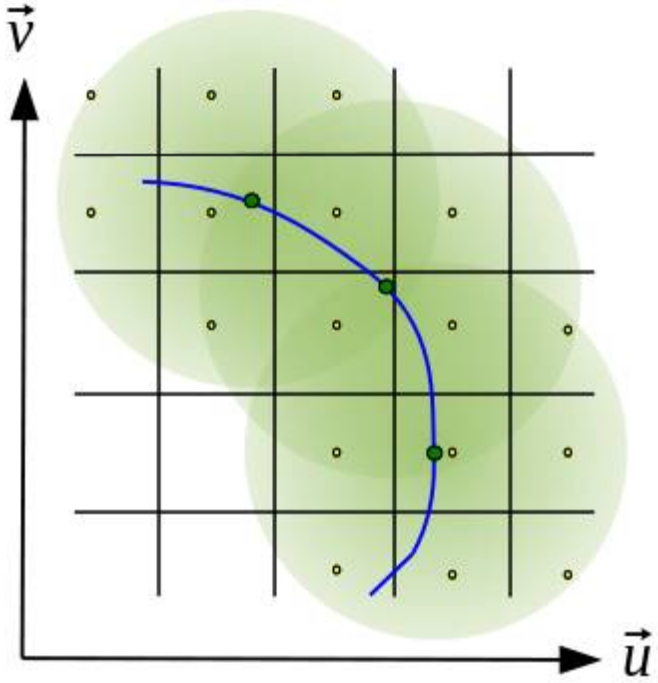
(accumulate weights too, for PSF & PB)

Choose different gridding convolution functions
=> handle wide-field effects (next slide)

iFT : Fourier Transform to image domain

Normalization : Divide by sum-of-weights
(and PB, for some gridding algorithms)

- flat noise, flat-sky, pb-square



Weights

- Data Weights
- Imaging Weights
  - Natural
  - Uniform
  - Briggs

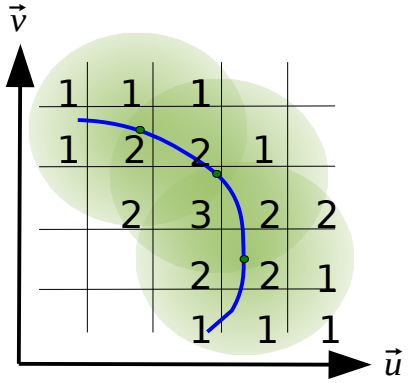
https://casa.nrao.edu/casadocs/casa-5.1.0/synthesis-imaging/data-weighting

Imaging Algorithms (Weighting + Gridding)

Weighting Schemes :

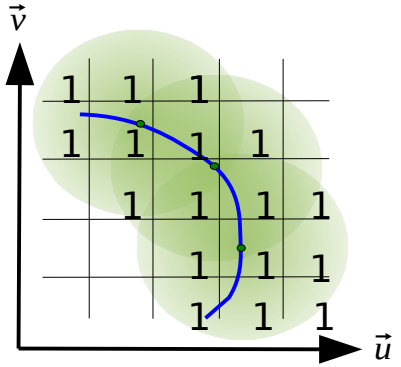
Natural

- Highest sensitivity
- PSF can be wide



Uniform

- Narrow PSF
- Suppressed sensitivity



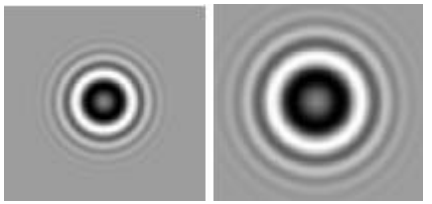
Briggs/Robust : Smoothly vary between natural and uniform
(robust 0.5 == AIPS robust 0)

UV-Taper : Emphasize large scales

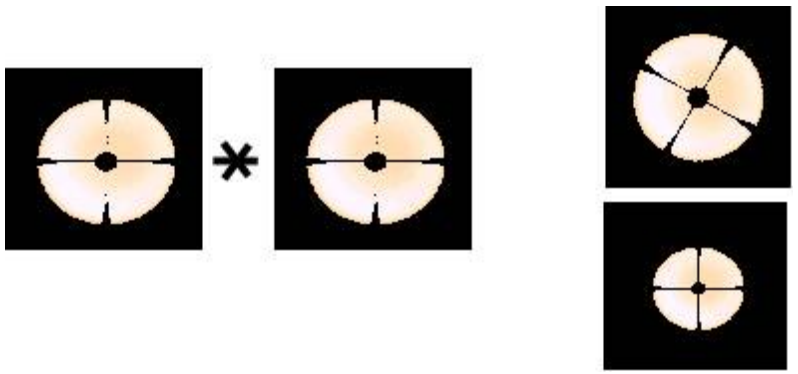
Gridding : Choice of convolution function

Standard Imaging :
Prolate Spheroidal

W-Projection :
FT of a Fresnel kernel



A-Projection :
Convolutions of Aperture Illumination Funcs + phase gradients for joint mosaics



"A", "W" kernels can change with baseline, time and/or frequency.

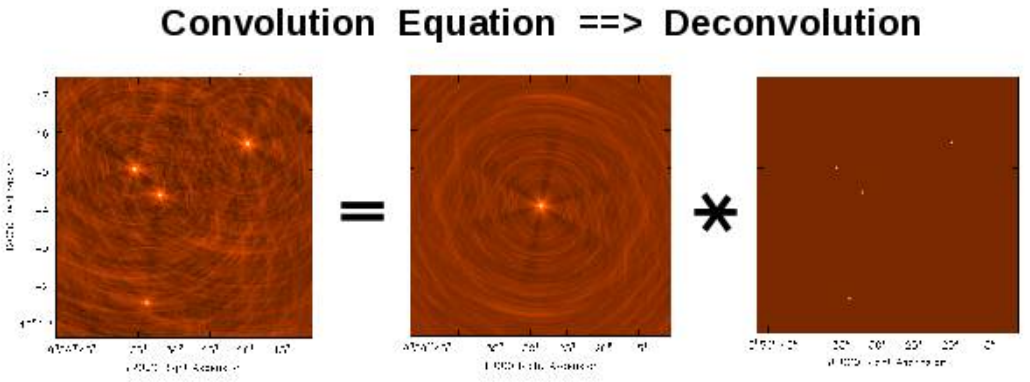
See Wideband/Widefield lecture for details

Deconvolution Algorithms

Point Source CLEAN : Sky model is a set of delta functions. Remove effect of PSF

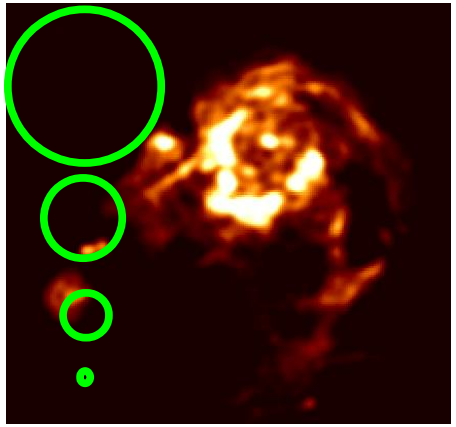
- Hogbom (use PSF of same size as image, clean until image edge)

- Clark / ClarkStokes (use patch PSF, intermediate model subtraction on the gridded uv-plane)



Multi-Term CLEAN : Joint deconvolution of a sky model described as a series of basis functions

- MS-Clean (sky model is a linear combination of 'blobs' of different sizes)



- MT-MFS (wideband sky model is a Taylor series expansion of Intensity vs Frequency)

  - Solve for intensity and spectrum together
  - Includes multi-scale

(Details : CASADocs and the Wideband/widefield lecture)

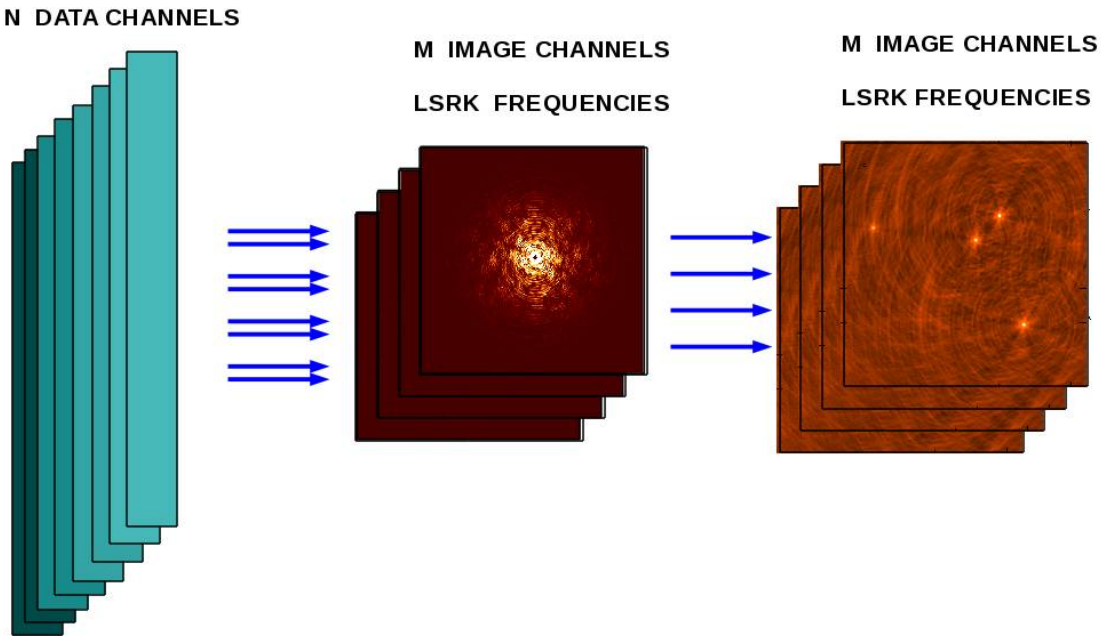
Image Definition : Spectral Cubes

specmode = ‘cube’

- N data channels are mapped to M image channels (with binning/interpolation)

- Image coordinates defined by the user : start, width, nchan, outframe (channel, frequency, velocity)

- Image coordinate system is internally stored In LSRK frame, with a conversion layer to allow relabeling to outframe for display/analysis



- All gridding/imaging is done in the LSRK frame with on-the-fly conversions to LSRK (i.e. no cvel needed)

specmode=’cubedata’ in tclean

- No internal conversion to LSRK.
- Data channels map to image channels (with only binning/interpolation)

(Soon to come : specmode = ‘cubesrc’ in tclean
to track moving sources via ephemeris tables)

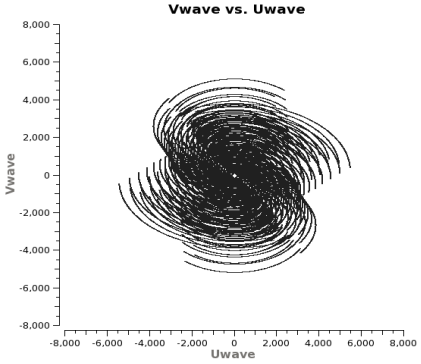
Image Definition : Continuum Imaging

specmode = ‘mfs’, ‘cont’

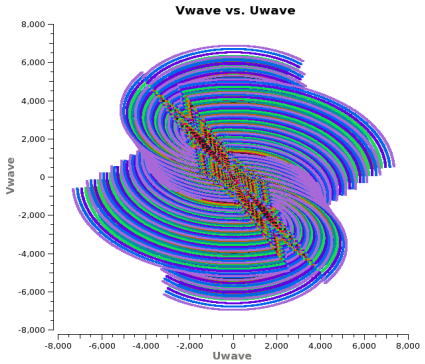
- Data from all channels are gridded onto a single uv-grid, using the appropriate u,v,w coordinates

- **Multi-Frequency-Synthesis**

1.5 GHz



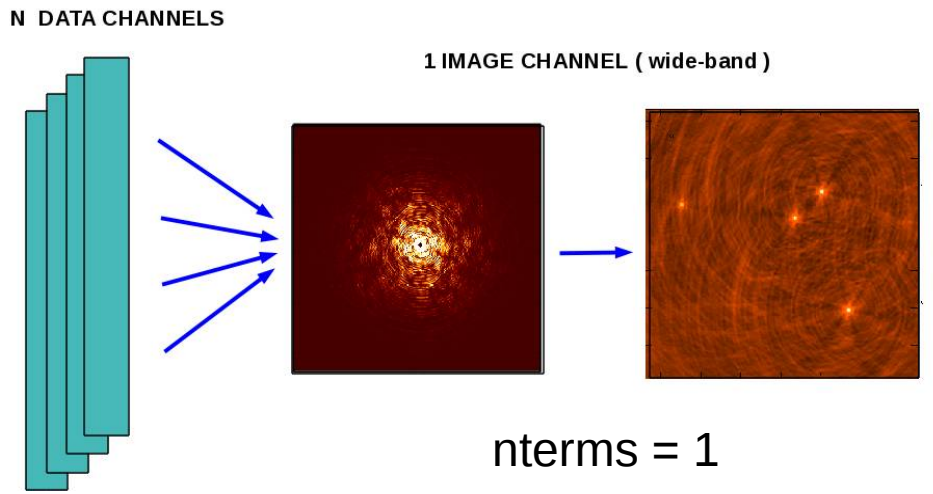
1.0-2.0 GHz



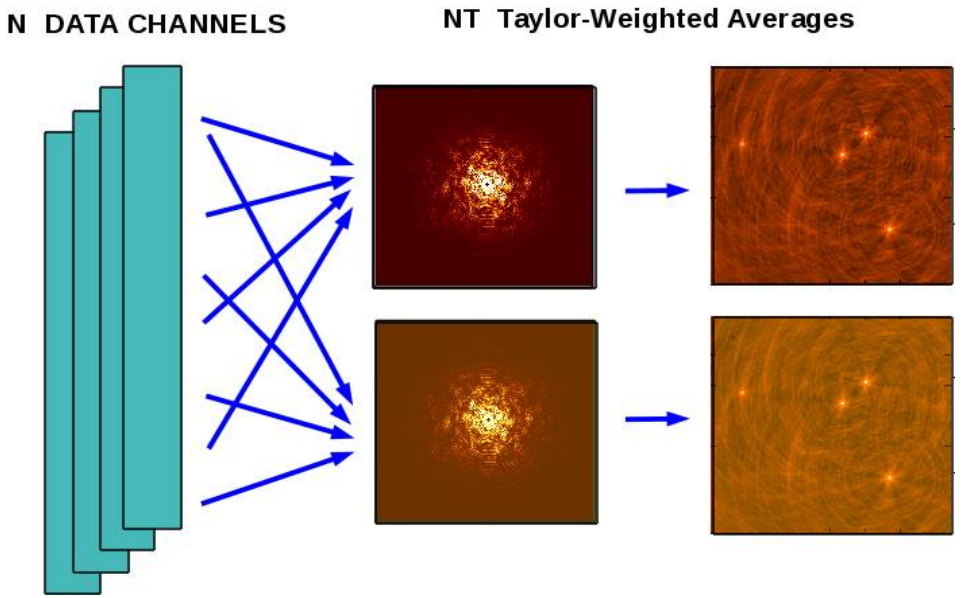
- nterms = 1 (flat spectrum assumption

- nterms > 1 (Taylor polynomial spectrum)

  - Major cycle : nterms Taylor-weighted averages of across frequency
  - Minor cycle : Solve for nterms coefficients



nterms = 1



nterms = 2

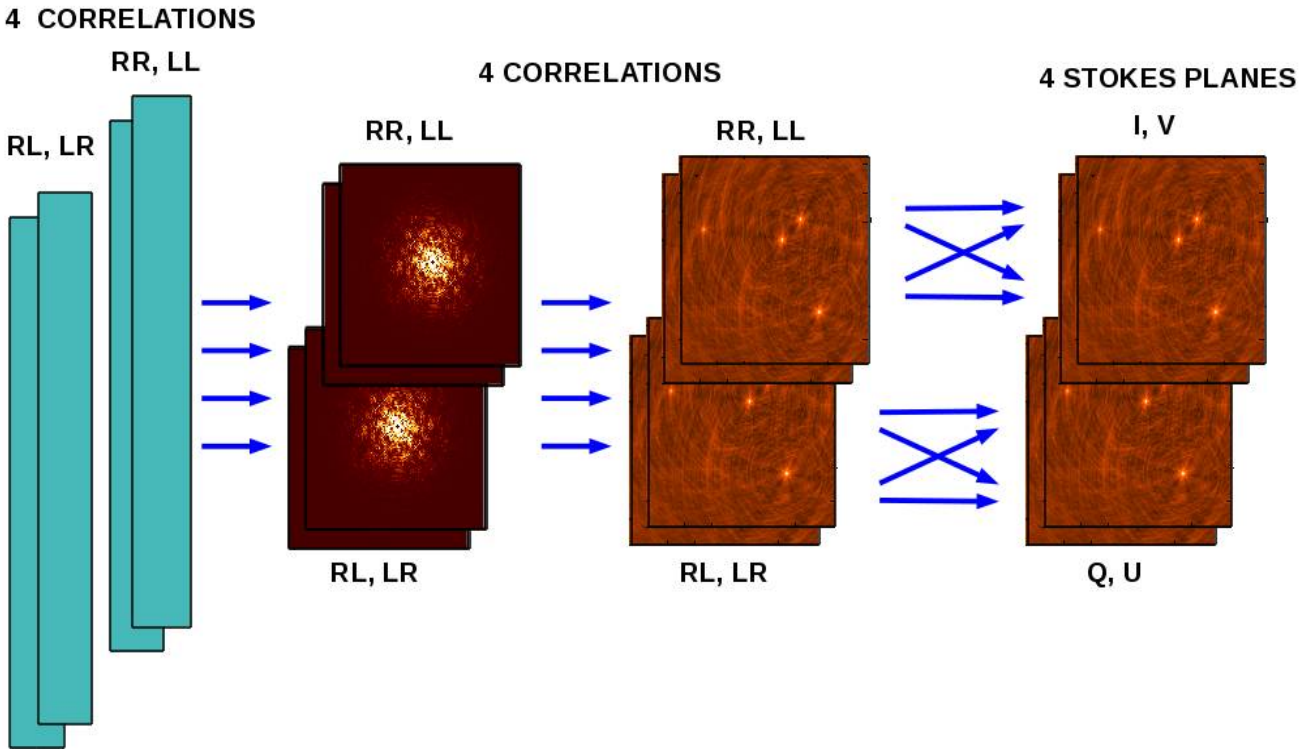
Image Definition : Correlation and Stokes planes

Polarization image planes

- Data are gridded and iFT
  In correlation basis

- Convert from correlation
  to Stokes basis

- Normalization
  (in Stokes basis)



Users can choose to make images of

R/L => I, Q, U, V, IV, QU, IQUV, R, LL, LR, RL, RRLL, RLLR, 'all'

X/Y => I, Q, U, V, IQ, UV, IQUV, XX, YY, XY, YX, XXYY, XYYX, 'all'

(Soon to come :
  For Stokes I, use data even if some correlation pairs are flagged – “pseudo-I”)

Image Definition : Multi-field and Facets

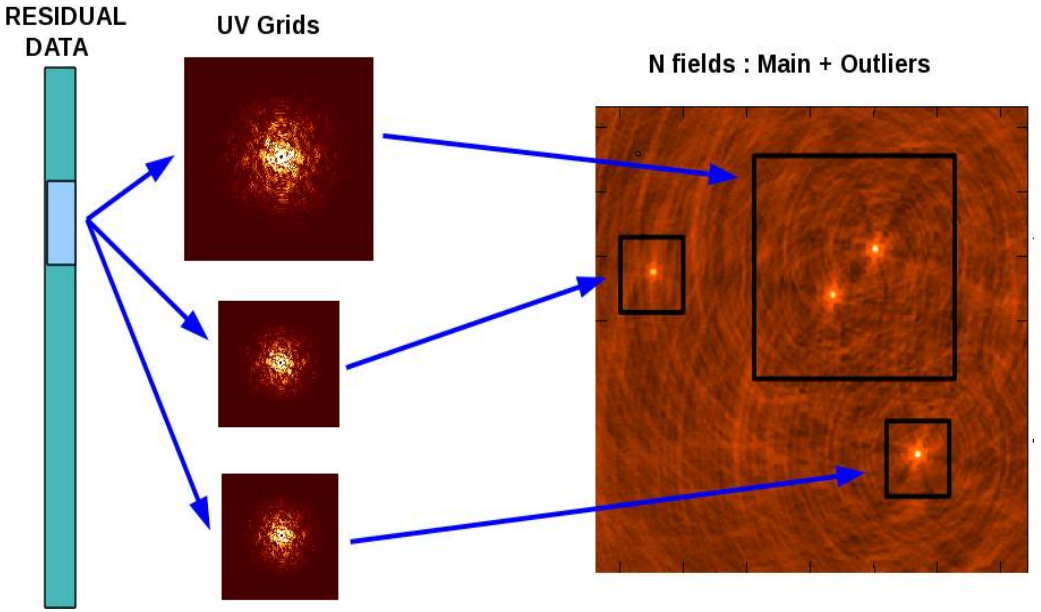
Image partitioning :
Same data, multiple images

Multiple fields (outlier fields)

- Usually, one large main field and several smaller outlier fields

- To avoid extremely large images

- Outlier file :
List of image definition parameters



Multiple Facets :

- Work with smaller field-of-view images
To get around the w-term problem
(non-coplanarity and sky curvature)

- Grid each facet separately onto a subimage, but do a single joint deconvolution (use PSF from first facet)

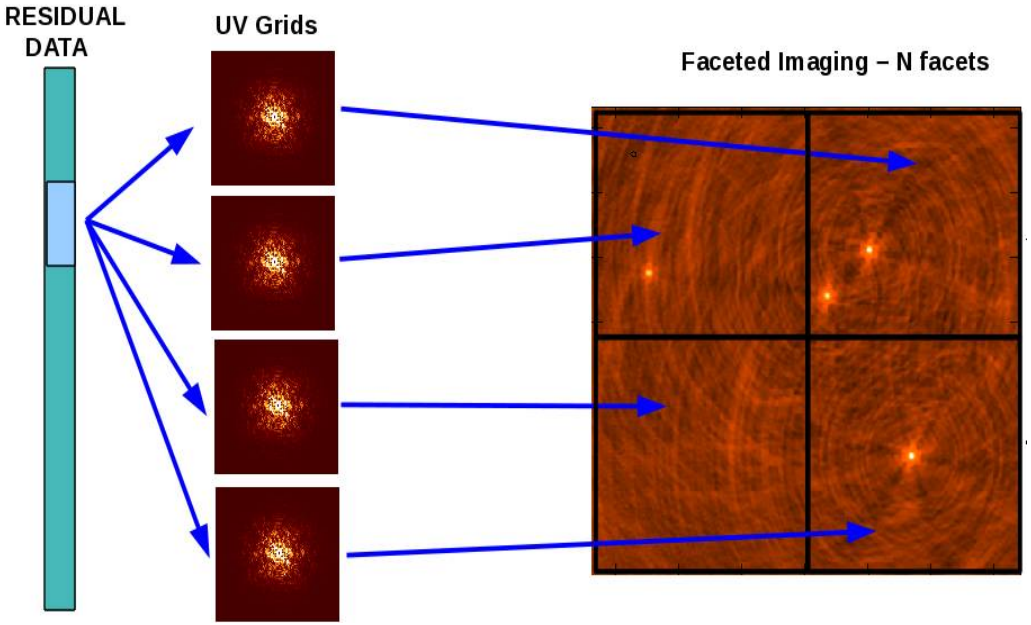


Image Definition : Stitched and Joint Mosaics

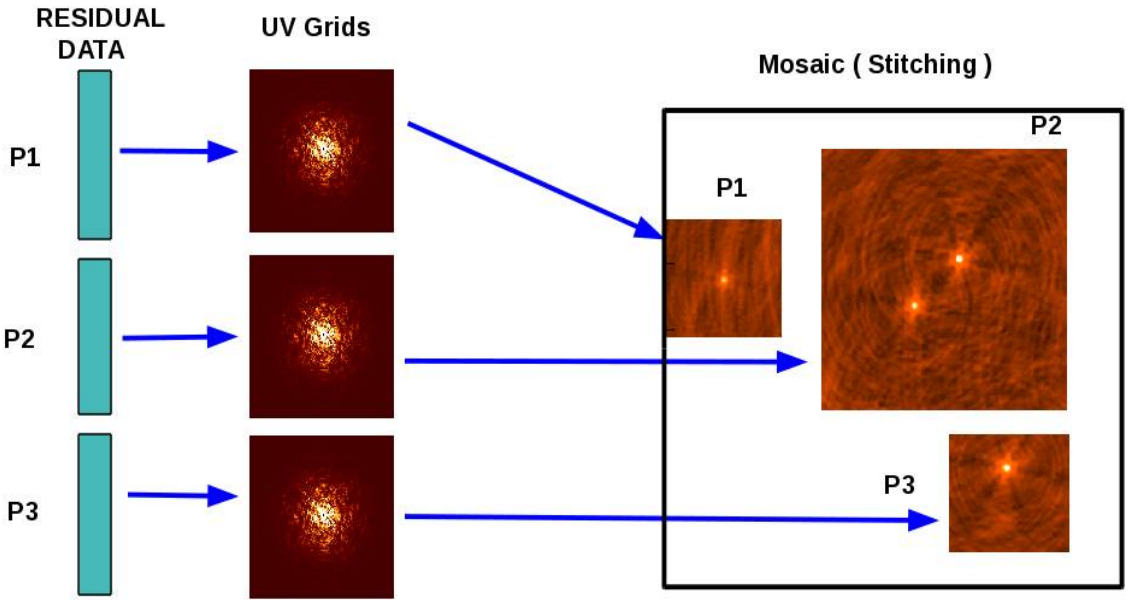
Mosaic :
Combine data from multiple pointings

Stitched Mosaic

- Image each pointing separately
- Combine them later

(lm : linearmosaic tool)

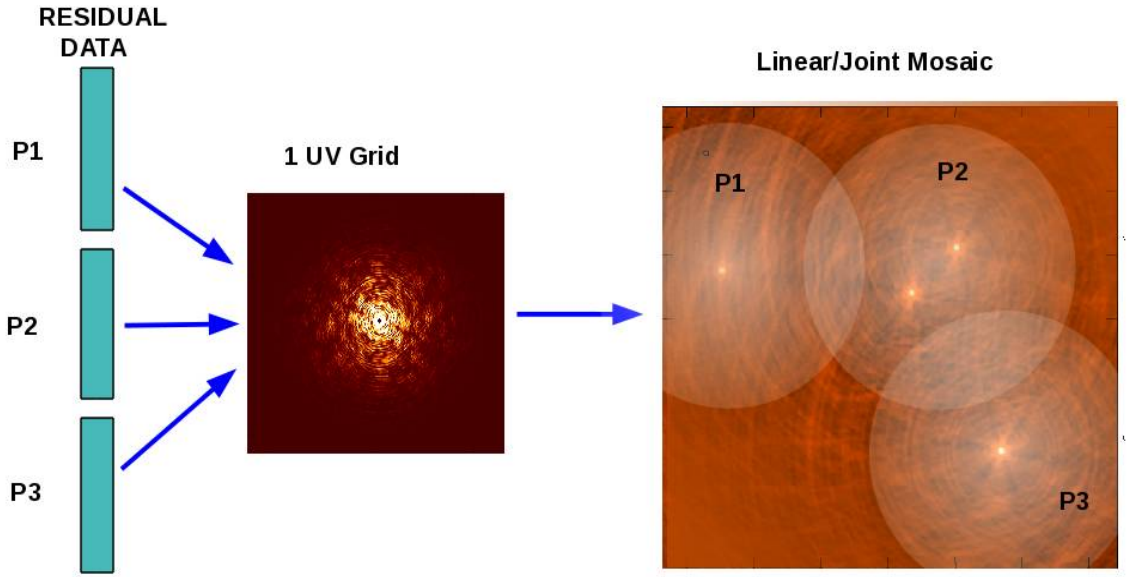
(Soon to come :
'image mosaic', grid separately but combine before minor cycle)



Joint mosaic

- Grid data from all pointings onto single uv grid, with appropriate phase gradients per pointing

- Joint deconvolution (assumes spatially invariant PSF)



Iteration Control

Loop gain (gain) : Controls how much of a source is modeled in each step.
Choose according to algorithm being used (point source / extended)

Exit criteria (niter, threshold, cycleniter, cyclefactor, etc...) :

(1) Minor cycle stopping criteria (per chan/pol plane)

- Zero mask (i.e. all pixels = False)
- Iteration limit (cycleniter)
- Peak residual in mask < cyclethreshold (computed from relative PSF sidelobe level)
- No iterations done
- Detects divergence (10% increase of peak residual across every ~2K iterations)

(2) Global stopping criteria (over all channels/pols, checked after each major cycle)

- Peal residual in mask < threshold
- Total number of iterations >= niter
- Blank mask in all planes
- No change in peak residual from previous major cycle
- Divergence (5-times increase in peak residual from previous or from minimum)

(Soon to come : N-sigma based thresholds and iteration counts (per chan/pol plane))

Making and using deconvolution masks

Mask for deconvolution : CASA image filled with 1's and 0's
(Different from internal T/F masks in CASA images)

- Supply an image name as input via 'mask' parameter
- Supply a region string as input
- Automatic mask generation (tclean)
- Generate mask by drawing it interactively

Task makemask :

- copy from another image's T/F mask,
- LEL expression to compute mask from values
- evaluate a CASA region file to form a 1/0 mask
- extend/expand existing masks across planes

Automasking :

- auto-thresh (threshold based masks, using automatically computed rms)
- auto-multithresh (adaptively grow/prune masks as cleaning proceeds)
- pbmask : outer limit for mask based on PB gain

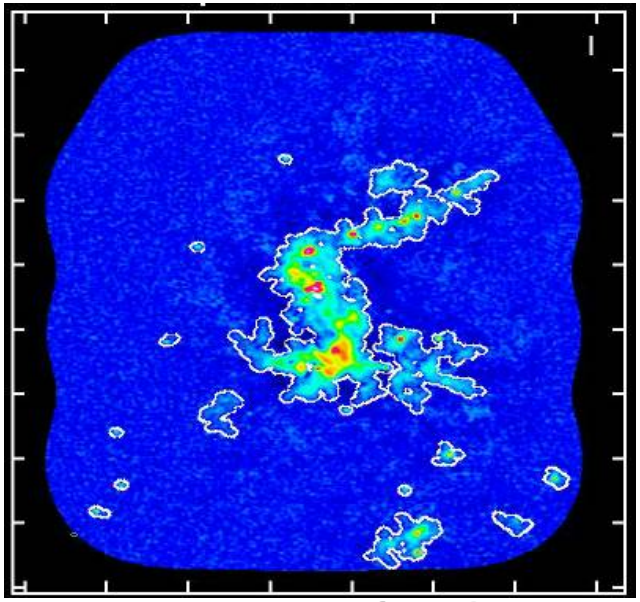


Image from C.Brogan

Interactive masking (and iteration control)

Interactive GUI

- View/edit masks in between major and minor cycles

- In tclean, the results of automasking will be visible and available for editing

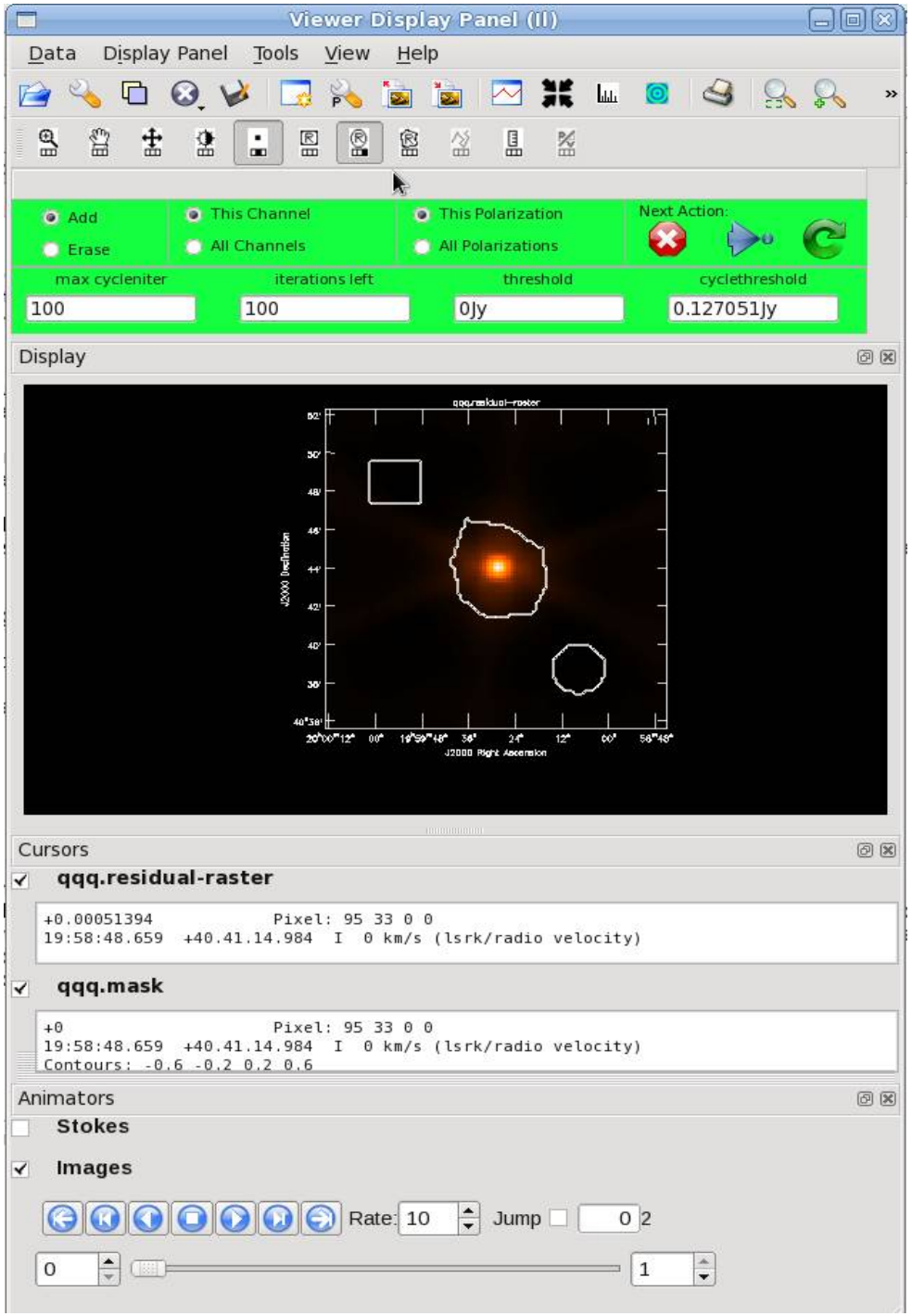
- Edit masks using fixed shapes or freely defined region drawing

  (choose from menu via mouse button mapping)

- Copy masks across channels/pols or keep them separate

- Options to Stop, Continue silently, Continue/pause after next major cycle

- Iteration control options....



Interactive masking (and iteration control)

Interactive GUI

clean

tclean

- Added ability to control major cycle triggers directly (cycleniter, cyclethreshold)

In the future : Use the new CARTA viewer for interactive masking, monitor/control

- Use CARTA's region creation/manipulation tools
- Runtime display of convergence history (peak residual, total flux...)
- Ability to interrupt at logical stopping points, and edit iteration controls

Some miscellaneous features

- Restart Options (calcpsf=F, calcres=F) : Ability to restart from minor cycle without recomputing residual image and psf (tclean)

- Use the Measurement Set in read-only mode (multiple processes imaging same MS)

- Saving a CLEAN model (savemodel) : Possible during the last major cycle
  (or) in a separate step : calcres=F, calcpsf=F, niter=0

  - modelcolumn (store model visibilities on disk in the MS)
  - virtual model* (store gridder params to compute model visibilities on-the-fly when needed)

  ---- Can sometimes use the saved model for continuum subtraction (via task uvsub)
  (Note : 'imcontsub' and 'uvcontsub' fit arbitrary smooth polynomials per pixel/baseline)

- Combining Single Dish and Interferometer data

  - Combine SD image with restored output interferometer image (task feather)
  - Use SD image as starting model for Interferometric imaging (clean, soon to come in tclean)

- Tool Interface (tclean) : Python layer (PySynthesisImager) is useful for prototyping
  https://casa.nrao.edu/casadocs/casa-5.1.0/global-task-list/task_tclean/examples

Factors affecting runtime (and memory use)

How long will my imaging run take ? Well, it depends....

- Image size : All operations scale by N^2 (for an NxN image),
  Choose a size optimized for fftw (factored by 2,3,5,7).

- Data size : Major cycle scales linearly with data volume

- Gridder : Size of gridding convolution function ranges from 3 pixels on a side to
  ~few x 100 on a side => Orders of magnitude difference in runtime

- Deconvolver : MS-Clean, MTMFS are more expensive than Hogbom/Clark

- Iterations : Number and frequency of major cycles, appropriateness of minor cycle
  algorithm to sky structure, number of minor cycle steps (and loop gain)

- Sky structure : High/low SNR, Calibration artifacts, easy/hard convergence, etc ?

- Machine specifications : RAM / Swap, Disk speed (local, nfs, ssd),nCPUs, Ncores,..

(Soon to come : A resource predictor for tclean to estimate peak memory use
  Status : Under development

Soon to come : Parallelization of entire run (cube) and for major cycles (continuum)
  Status : Exists and in a late stage of testing/validation)

Task Interface (tclean)

vis = 'msname' or ['msname1', 'msname2', ...]
- Selection options (field, spw, ...)

imagename = 'xxx'
- Image definition options (imsize, cellsize, startmodel, ...)

specmode = 'mfs' or 'cube'
- Channelization options (start, step, nchan,... in different units, reffreq)

gridder = 'standard', 'widefield', 'mosaic', 'imagemosaic', 'awproject'
- Algorithm-specific parameters (wprojplanes, cfcache, pblimit, etc....)

weighting = 'natural', 'uniform', 'briggs'
- Options for some choices (robust,...)

deconvolver = 'hogbom','clark','multiscale','mtmfs',
- Algorithm-specific options (scales, nterms, ...)

usemask = 'user','pb','auto-thresh','auto-multithresh'
- Mask creation options (pbmask, threshold, maskname, etc...)

niter = N
- Iteration control options (gain, threshold, cycleniter, cyclefactor, interactive=T/F...)

parallel = True/False (when started up with 'mpicasa')

(+ outlier files, restart option (calcres=T/F, calcpsf=T/F), savemodel, restoration, etc...

Output Images

imagename.psf	Point Spread Function
imagename.pb	Primary Beam
imagename.residual	Residual Image (or initial Dirty Image)
imagename.model	Model Image after deconvolution
imagename.image	Restored output image
imagename.image.pbcor	Primary Beam corrected image
imagename.mask	Deconvolution mask
imagename.sumwt	A single pixel image containing sum of weights per plane
imagename.weight	Image of un-normalized sum of PB-square (for mosaics and A-Projection)
imagename.XX.{tt0,tt1,tt2}	Multi-term images representing Taylor coefficients (of polynomials that model the sky spectrum)
Imagename.workdirectory	Working directory for partial products during parallel runs

Image Viewing and Analysis

Viewing Images

Task **viewer** (or **casaviewer** from the command line)

- Image cubes (2D diplays with animation controls for other dimension(s))
- Rasters, contours, vectors, markers
- Stacks of images, each of which can be a 2D, 3D or 4D ‘cube’
- Region drawing + statistics + histograms
- Spectral Profile tools : Line overlays, Spectral fitting, P/V diagrams, etc
- 2D image fitting
- Colormap editing, intensity scaling, etc on the 2D raster (per image, or global)

(scripting the viewer via task **imview**)

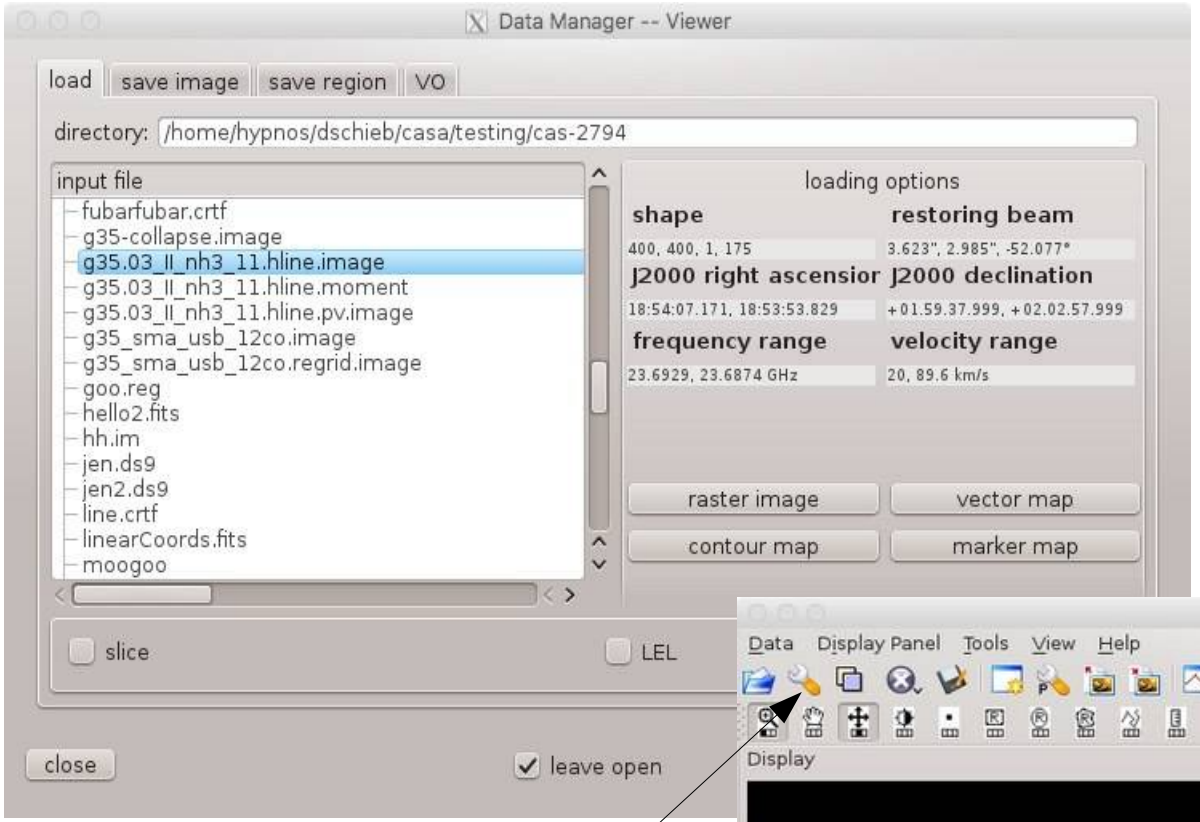
CASAdocs : https://casa.nrao.edu/casadocs/casa-5.1.0/image-cube-visualization

Image Analysis (**28 separate tasks**)

- Import/export, viewing headers, reformatting (regridding, spectral collapse, etc)
- Mathematical analysis – image plane + spectral
- Working withrRegions, masks, image sub-selection syntax

CASAdocs : https://casa.nrao.edu/casadocs/casa-5.1.0/image-analysis

Image Viewing and Analysis – Viewer task + GUI

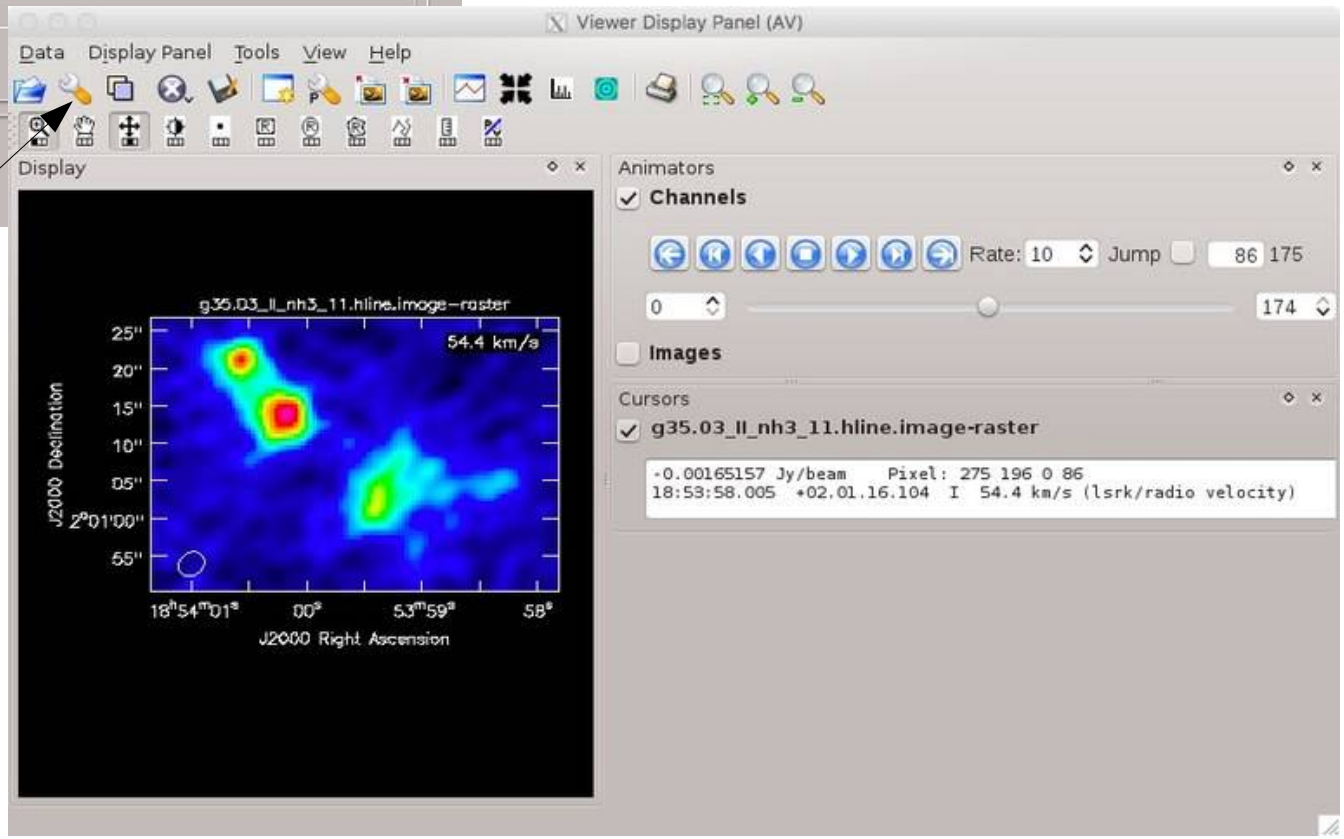


Data Manager

- Select images and display type

Viewer Display Panel

- Cursor tracking display
- Axes animation controls
- Drawing tools by mapping mouse buttons to functions

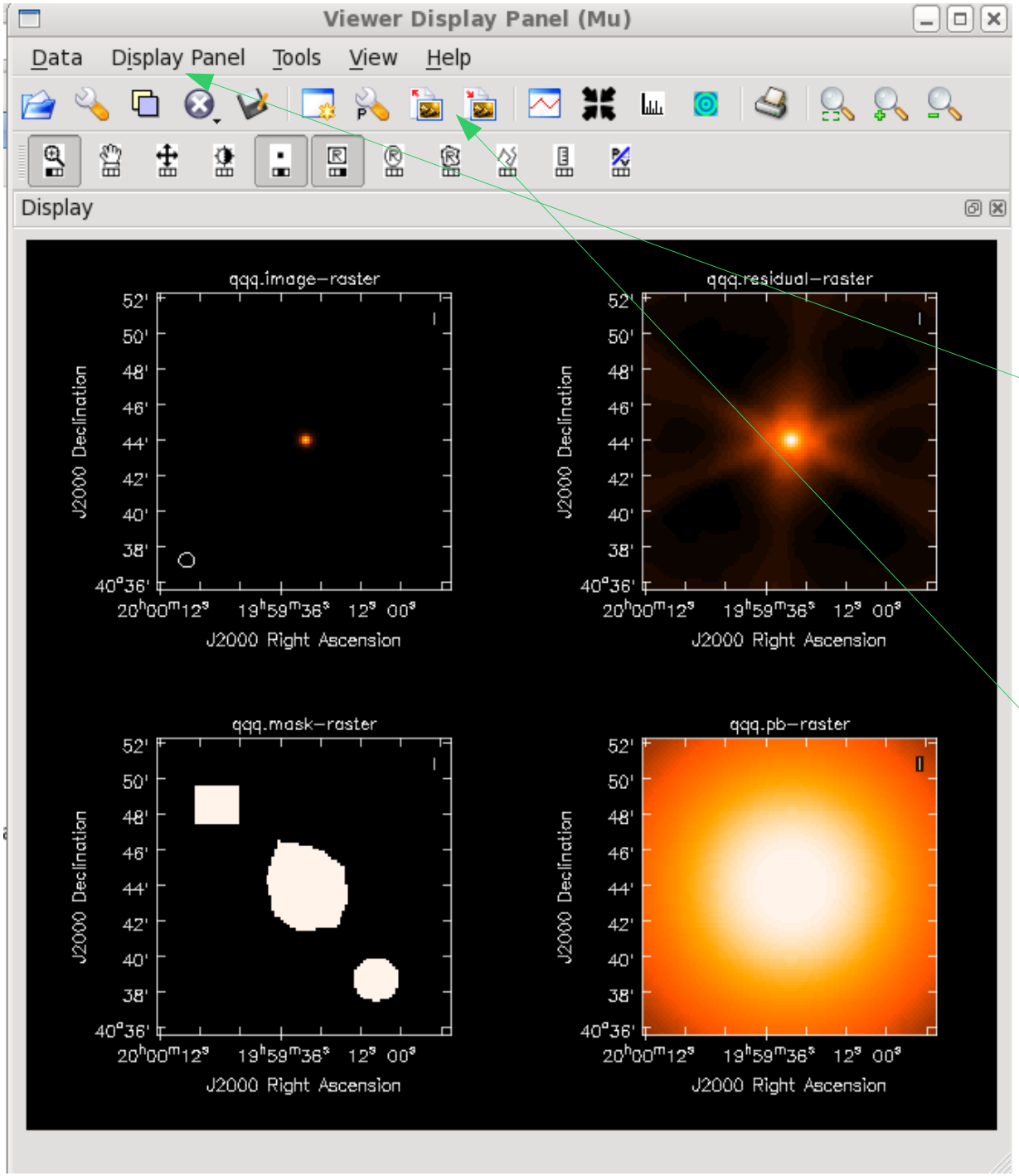


Display settings

- Adjust panel

  - Color scale, data range
  - Axis labels
  - Display/animation axes
  - Coordinate systems
  - ...etc...

Image Viewing and Analysis – Stacks and Multi-panel displays



Multi-Panel displays

- Load several images (one at a time)
- Configure Nx, Ny in the Display Panel options
- Zooming / stats / region drawing can operate on all images
- Adjust display options per image or globally

Preserving viewer State

- Save current state of the GUI (xxx.rstr file on disk)
- Restore later upon fresh restart of the viewer

Image Viewing and Analysis – Region statistics

Regions and Statistics

- Draw regions of Different shapes

- Save/import region files

- Display statistics

- Display histograms

- Animation axis for Stacks of images

- Region stats also printed to terminal

(Task **imstat** also does Image statistics)

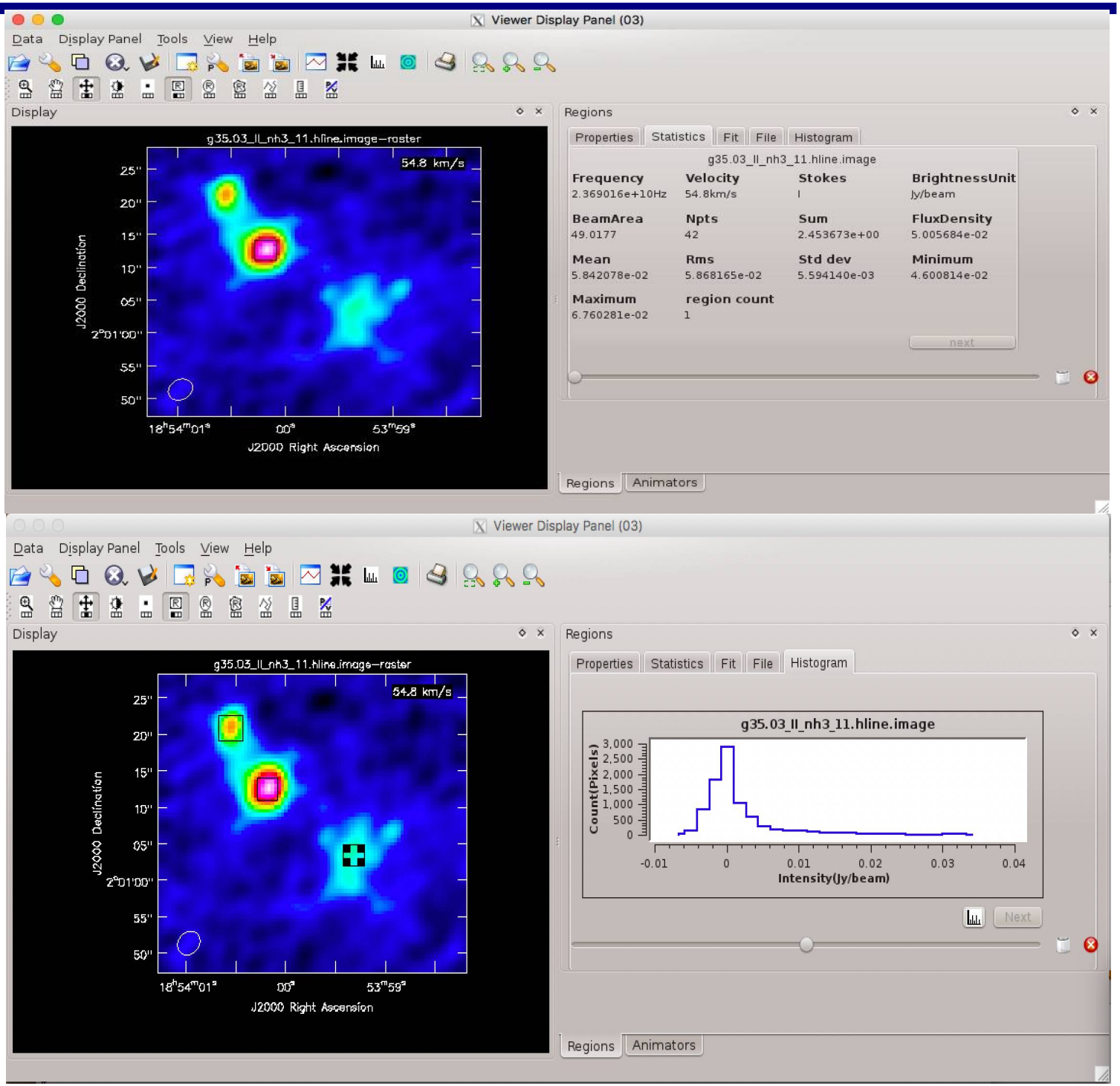
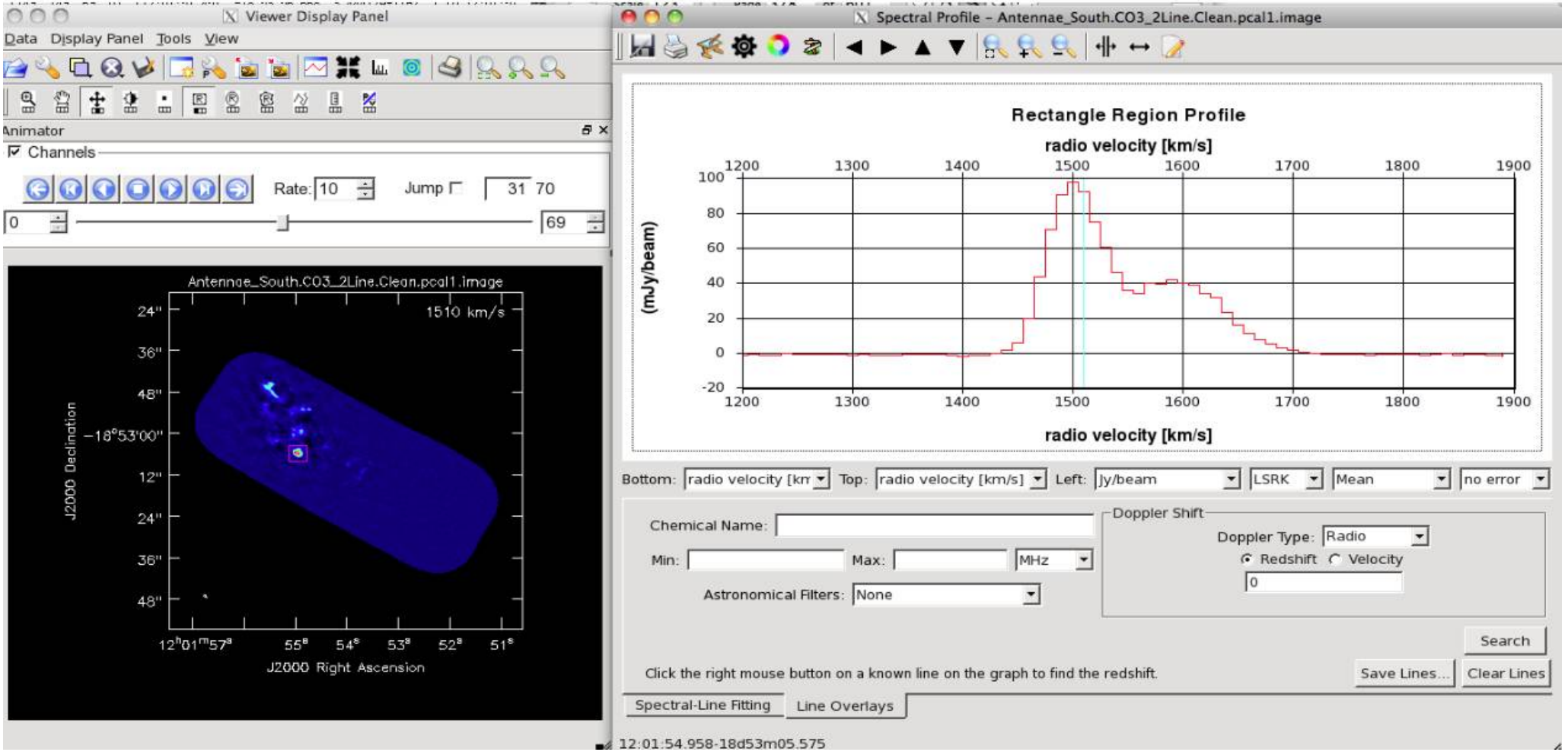


Image Viewing and Analysis – Viewer's Spectral Profile tool

View spectra of flux within selected regions



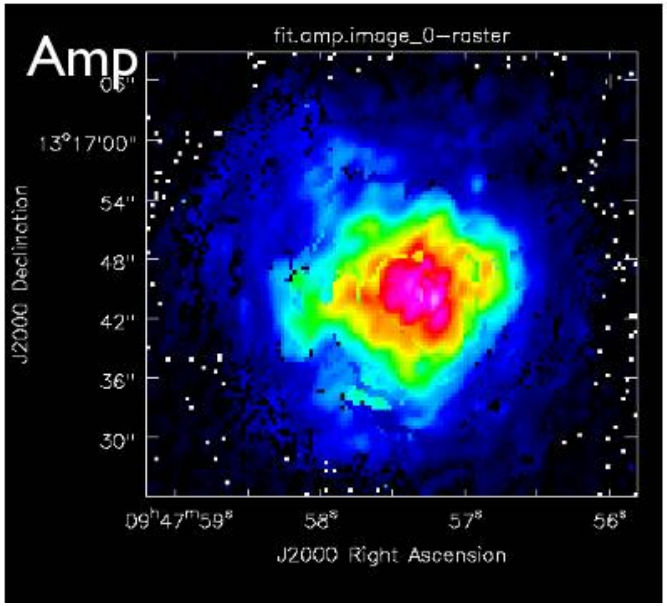
- Spectral Line fitting tool (also done by task **specfit**)
- On-the-fly spectral frame conversions
- Line overlays (via queries of Splatalogue spectral line database)

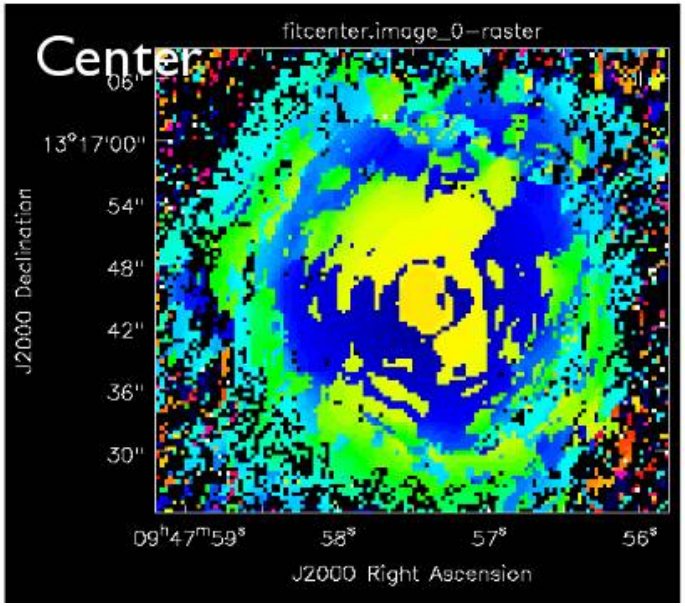
Image Viewing and Analysis – specfit task

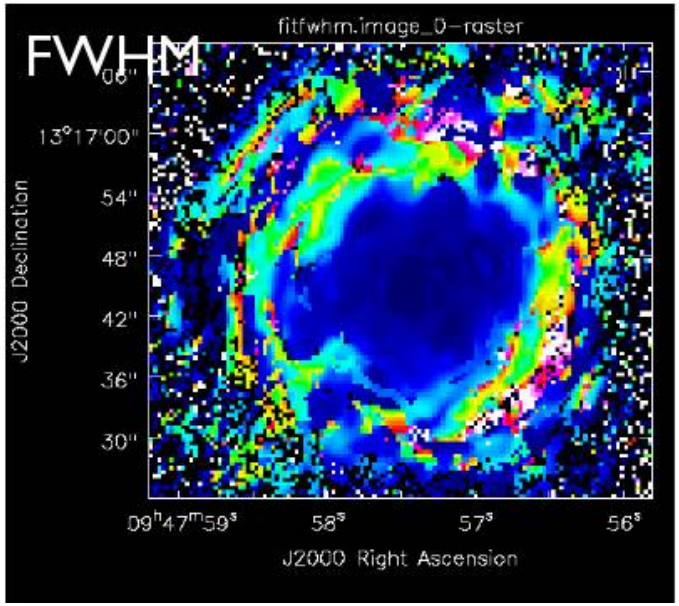
Specfit task

- Fits 2D Gaussians and/or polynomial models to an image/region, typically along the Spectral axis.

  - Can fit multiple Gaussian multiplets with constraints
  - Can be per-pixel ==> Output images of Amp, FWHM,Center velocity

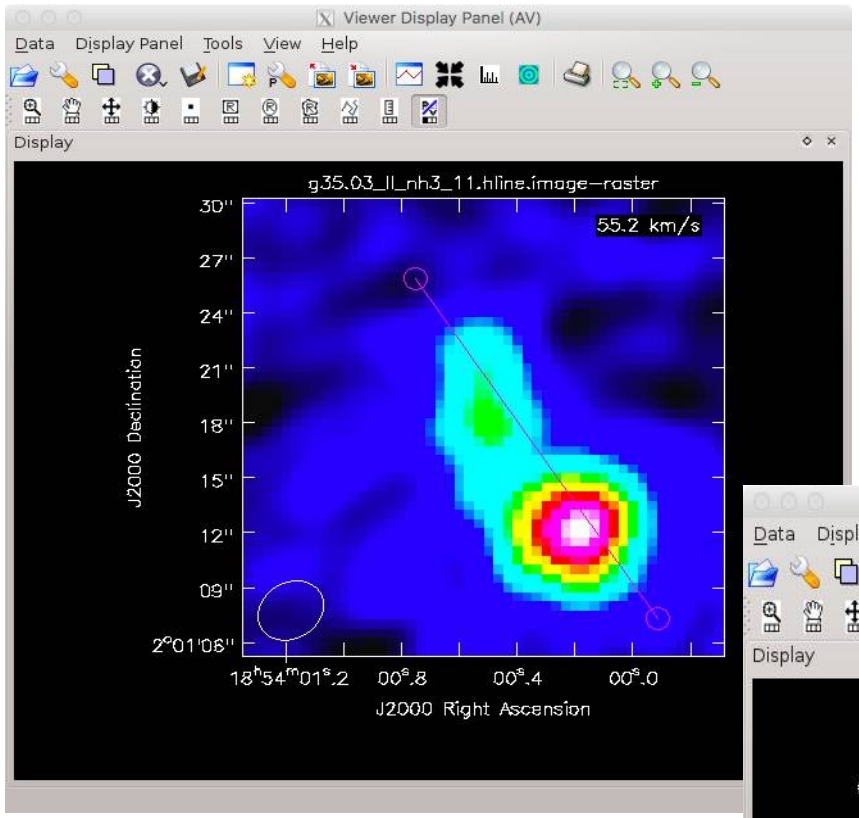




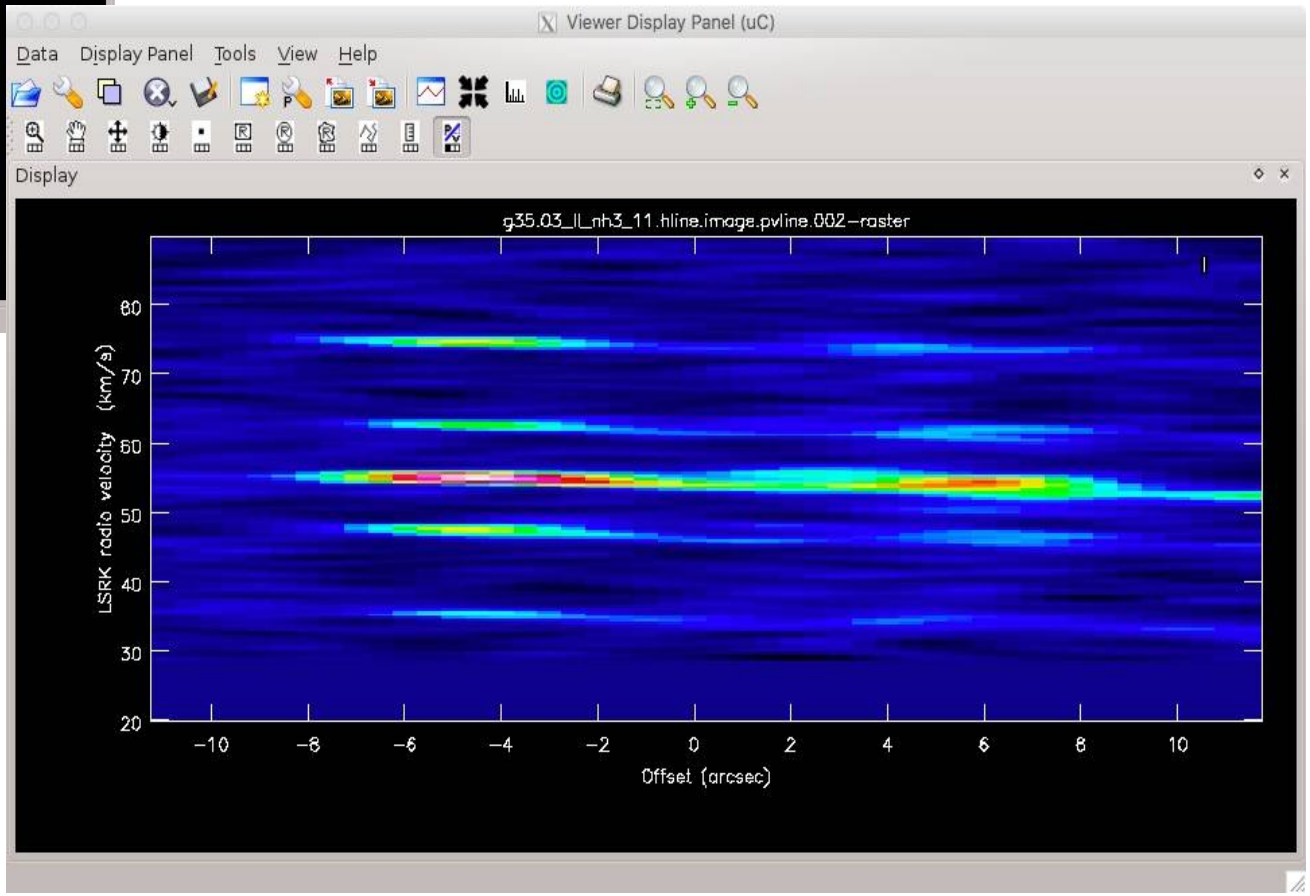


Images from J.Ott

Image Viewing and Analysis – Position/Velocity diagrams



Select a cut across the image,
and make a P/V diagram



(Task **impv** also constructs
a Position-Velocity diagram)

Image Analysis – 28 different tasks

- **Import/export** : **importfits, exportfits, imval**

- **Viewing headers** : **listfits, imhead, imhistory**

- **Reformatting** (resizing, binning, regridding to different csys, collapsing axes)

  - **imsubimage, imtrans, imregrid, imreframe, imrebin, imcollapse**

- **Spectral Analysis** : **imcontsub, immoments, impv, specsmooth, specfit, specflux, plotprofilemap, rmfit, spxfit, slsearch (splattotable)**

- **Image domain analysis** : **imfit, imsmooth, immath, imstat, imdev**

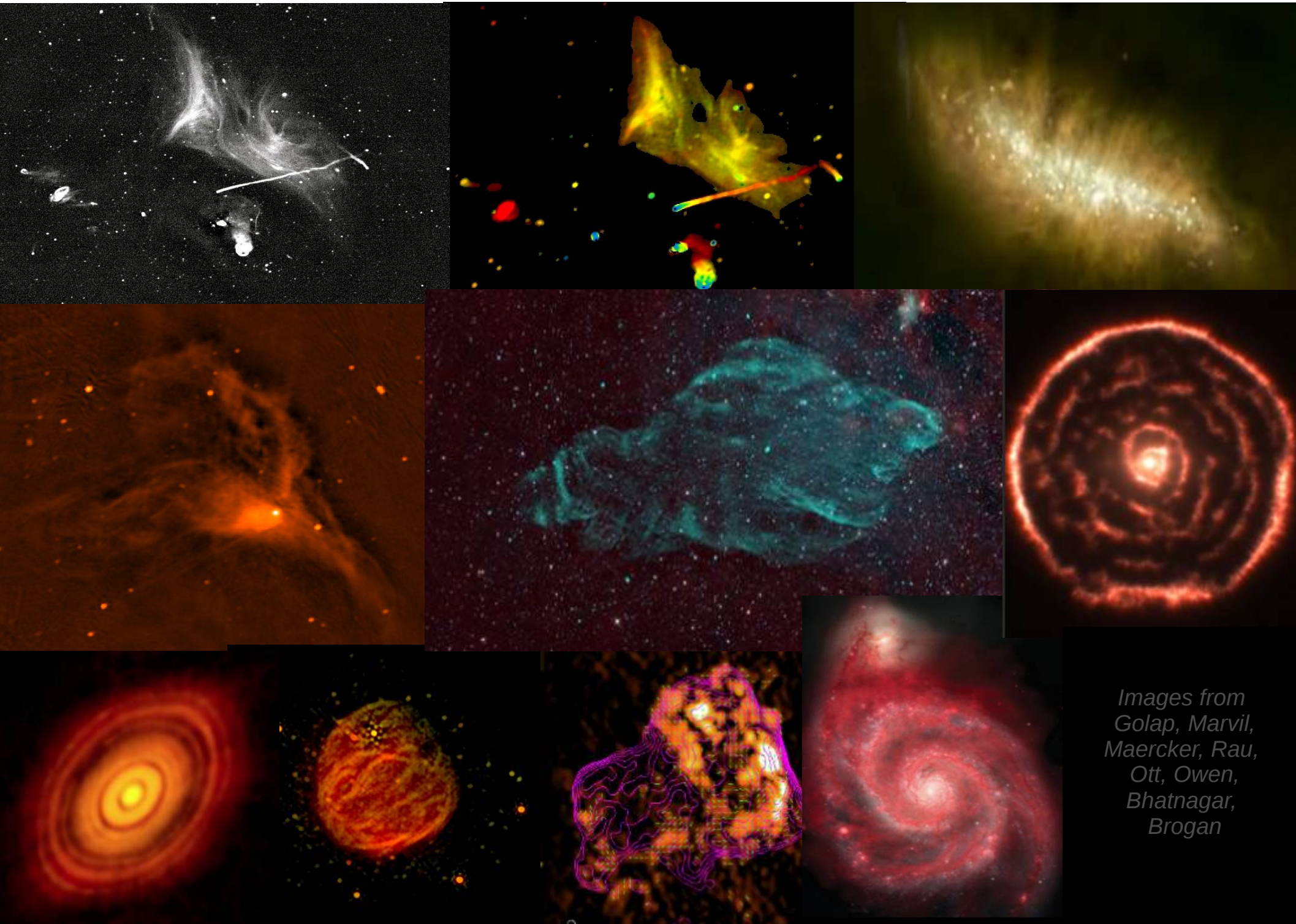
- **Additional topics** (please see CASAdocs):

  - Tool interface : ia tool
  - Working with Regions (syntax and CASA Region File Format)
  - Working with T/F masks within images (e.g. ia.maskhandler() tool)
  - Working with Lattice Expression Language (for math, and sub-selection)

  - Read pixel values into Python arrays/dictionaries for custom analysis
    E.x. data = imval('xxx.image')

https://casa.nrao.edu/casadocs/casa-5.1.0/image-analysis/dealing-with-images

Questions ?



Images from Golap, Marvil, Maercker, Rau, Ott, Owen, Bhatnagar, Brogan