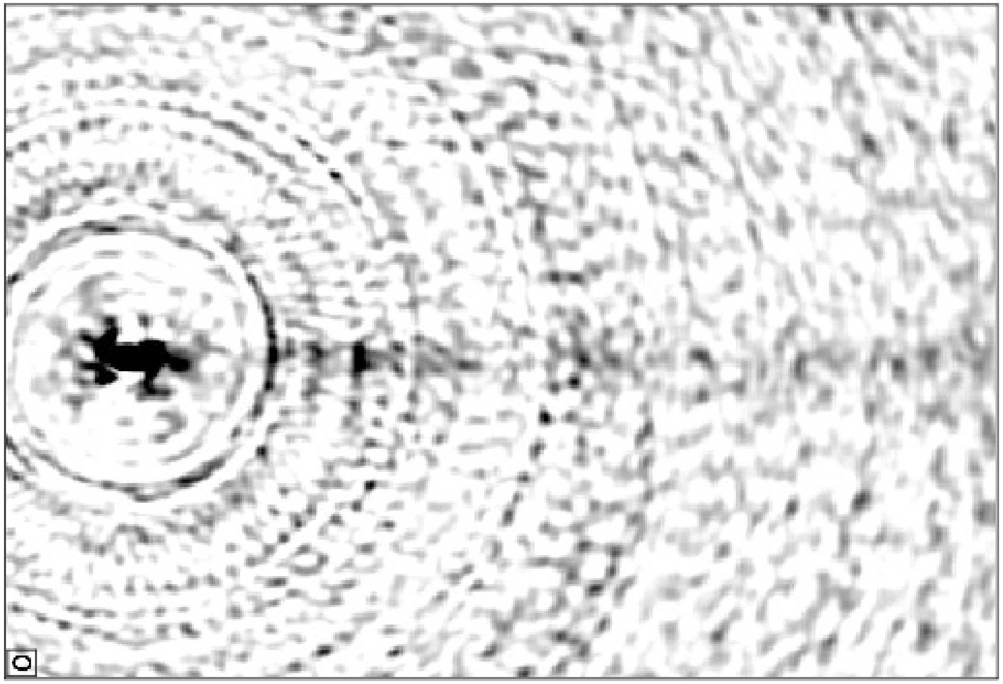
Continuum error recognition and error analysis

Robert Laing (ESO)

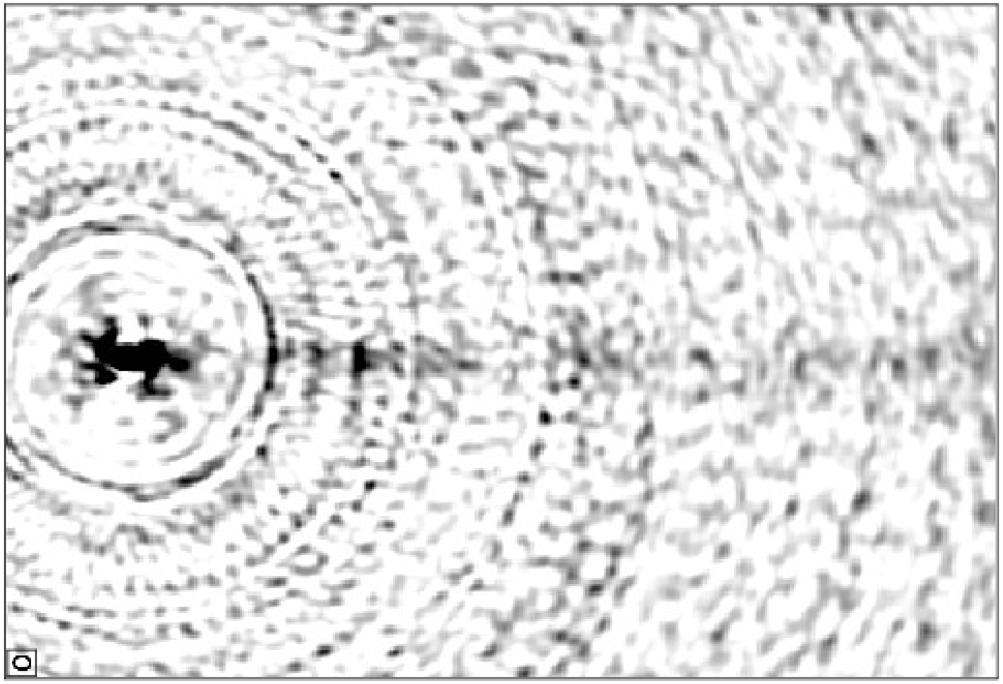


Outline

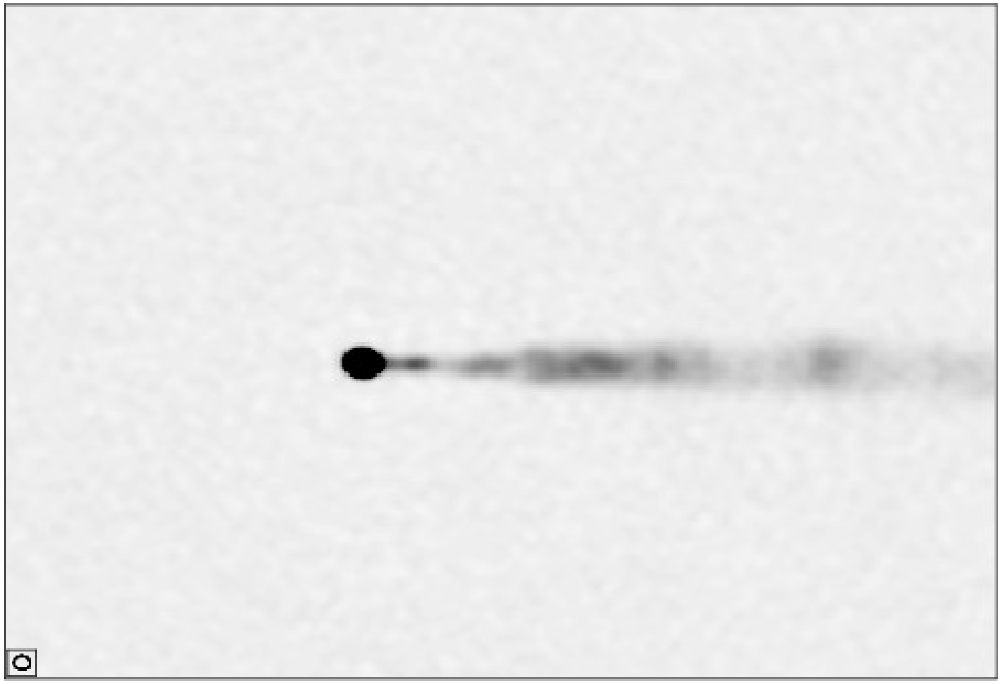
- Error recognition: how do you recognise and diagnose residual errors by looking at images?
- Image analysis: how do you extract scientifically useful numbers from images

- Unless otherwise specified, this talk is about continuum imaging in full polarization .... but many ideas also apply to spectral-line work.

Is my image good enough?



No



Yes

How can I tell (1)?

- Look at the off-source rms noise
  - Use on-line calculators (e.g. JVLA, ALMA) or formulae
  - Measure rms with (e.g.) casa viewer or imstat
  - Does the image rms increase near bright sources?
  - Is the noise random or are there ripples?
  - Compare the noise distributions for IQUV
- Are there obvious artefacts?
  - Coherent I features <-4σ
  - Rings, streaks etc.
- Properties of artefacts
  - Additive (constant over the field) or multiplicative (scales with brightness)?
  - Symmetric or antisymmetric around bright sources?

$$S_{rms} = \frac{2kT_{sys}}{A_{eff}\sqrt{N_A(N_A-1)t_{\rm int}\Delta\nu}}$$

How can I tell (2)?

- Large-scale negative structures
  - Negative “bowl' around source structure
  - Large-scale sinusoidal ripples
- Unnatural small-scale on-source structure
  - Diffuse structure looks spotty
  - Short-wavelength sinusoidal ripples

Missing short spacings

Deconvolution errors (often associated with poor u,v coverage

Possible causes: imaging problems (1)

- Is the image big enough?
  - Aliasing
  - Confusing sources outside the image
  - Make a wider-field, tapered image and look
  - Look in standard catalogues (NVSS)
- Are the pixels small enough to sample the beam?
  - Are bright point sources accurately located on pixels?
- Wide-field issues (calculate expected effects)
  - Averaging time too long? (Azimuthal smearing ∝ radius)
  - Spectral channels too wide? (Radial smearing ∝ radius)
  - w-term?
  - ionosphere?
  - direction dependence of antenna response (e.g. pointing errors at high frequencies; station beams at low frequencies)?

Possible causes: imaging problems (2)

- Missing short spacings
- Primary beam effects
- Deconvolution errors, especially with sparse u-v coverage
  - Resolution too high?
  - Poor choice of weighting?
  - Bad choice of CLEAN boxes (too small, too large, ...)
  - Insufficient CLEANing
  - single-scale CLEAN not good enough
- Source variability during the observations

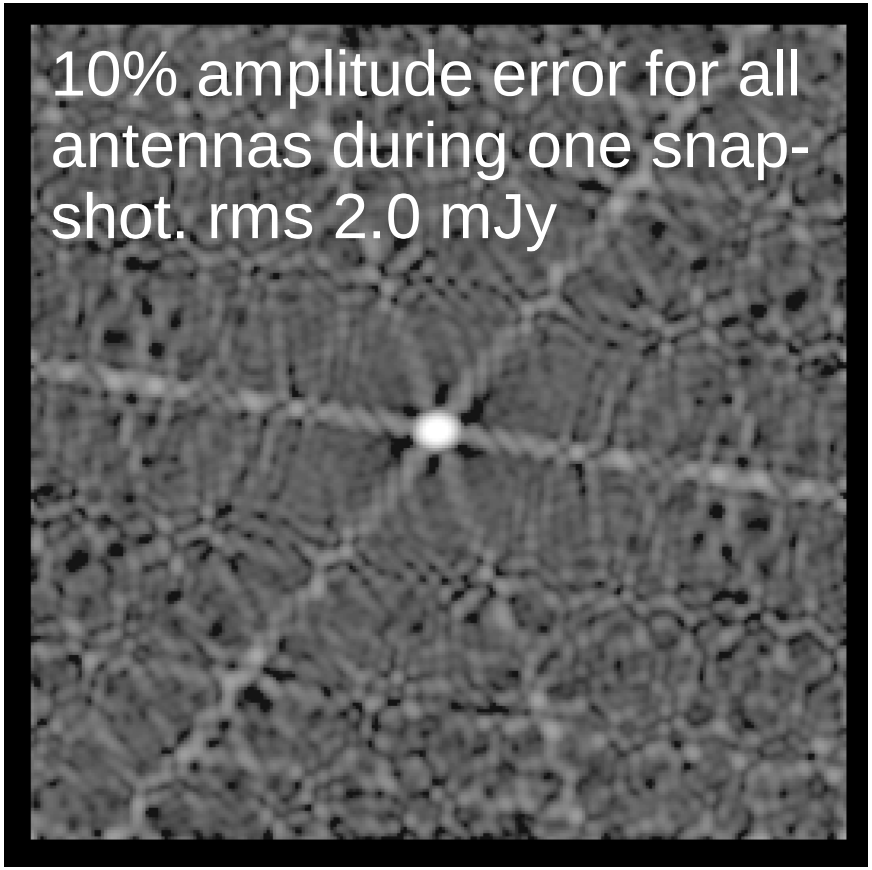
Errors in the image and (u, v) planes

- Errors obey Fourier relations between (u, v) and image planes
  - Particularly helpful in recognising additive errors
  - e.g. single very high visibility: sinusoidal fringe
- Easier to recognise narrow features
- Orientations are orthogonal
- u-v amplitude errors cause symmetric errors in the image plane
- u-v phase errors cause antisymmetric errors in the image plane

(u, v) or image plane?

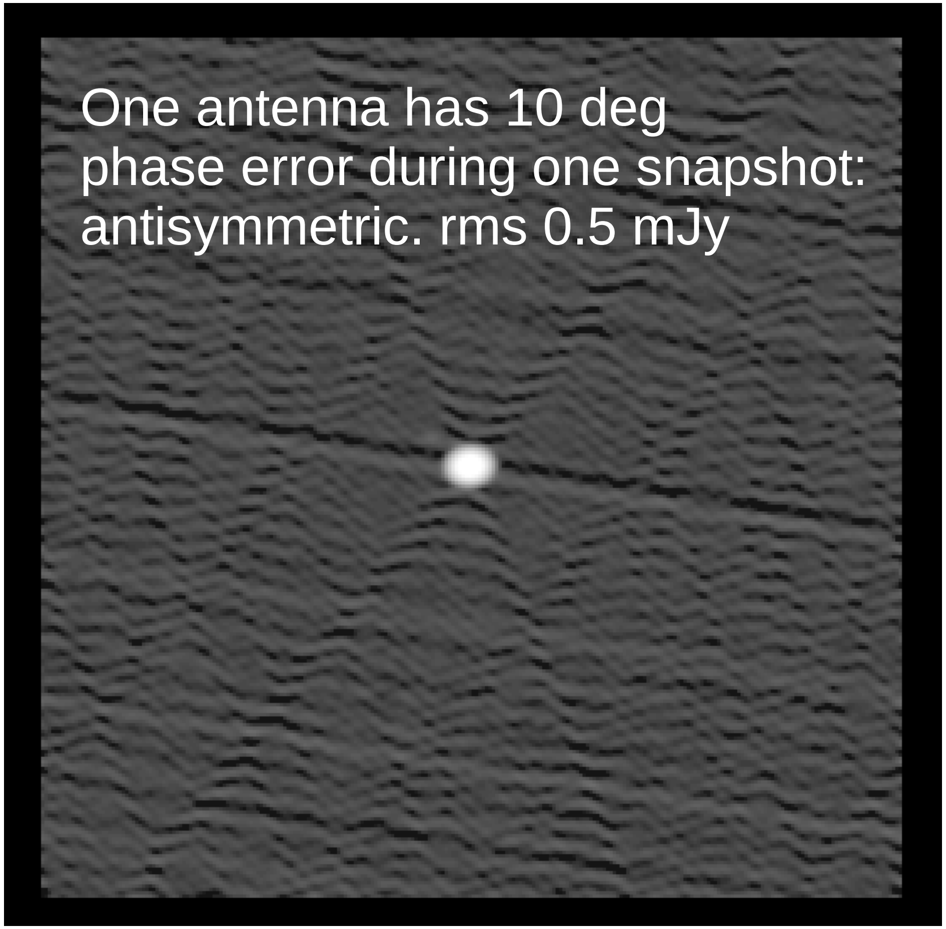
- Find the outliers in the u-v plane first
  - Gross (MJy) points have gross effects on the image (these should have been flagged, but mistakes happen)
  - A fraction f of bad data points with reasonable amplitudes give fractional error ~f in the image
- Low-level, but persistent errors are often easier to see in the image plane
- Rule of thumb: 10 deg phase error ≡ 20% amplitude error

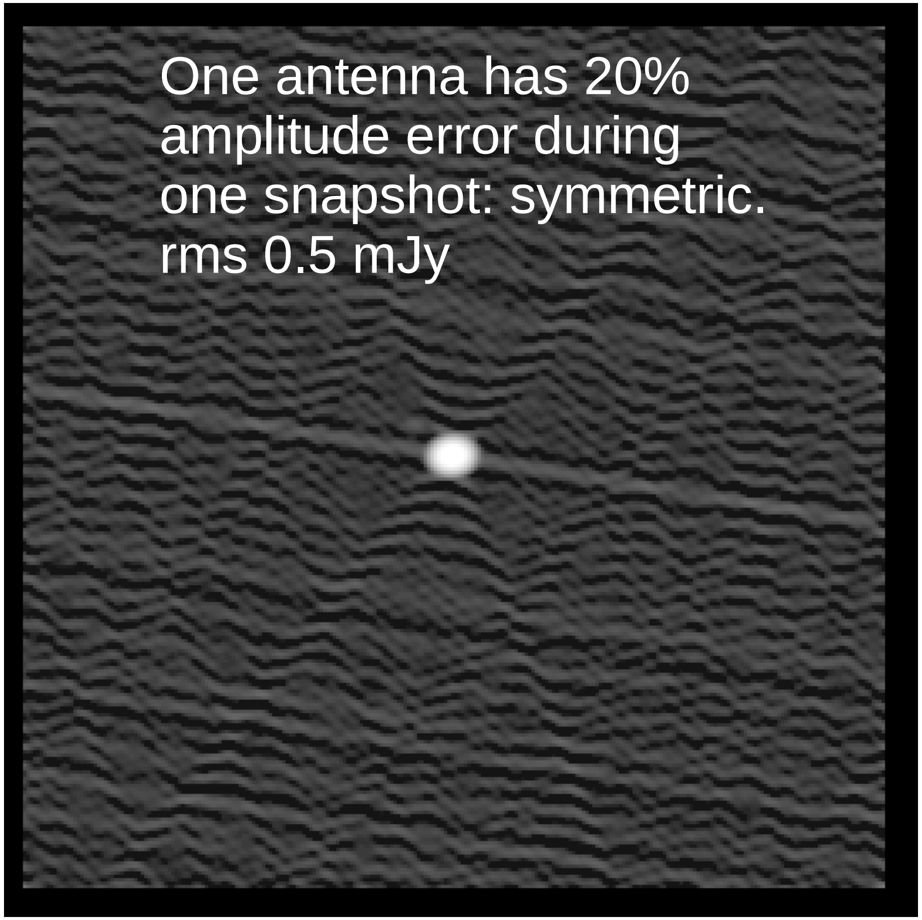
Amplitude errors: all antennas



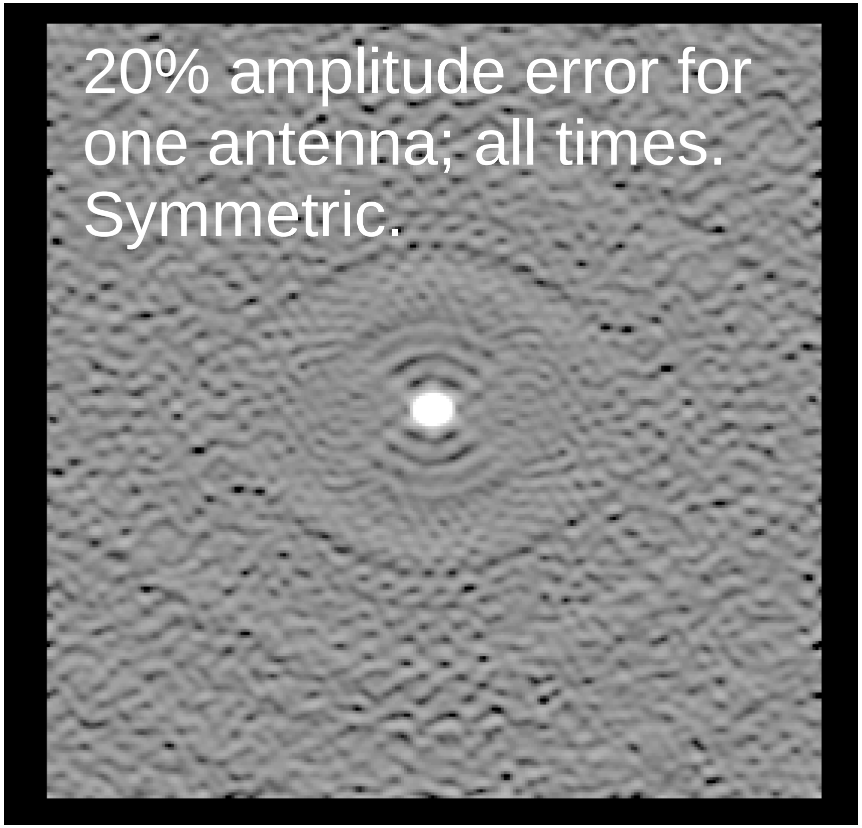
Error pattern looks like the dirty beam for a single VLA snapshot

One antenna in error at one time





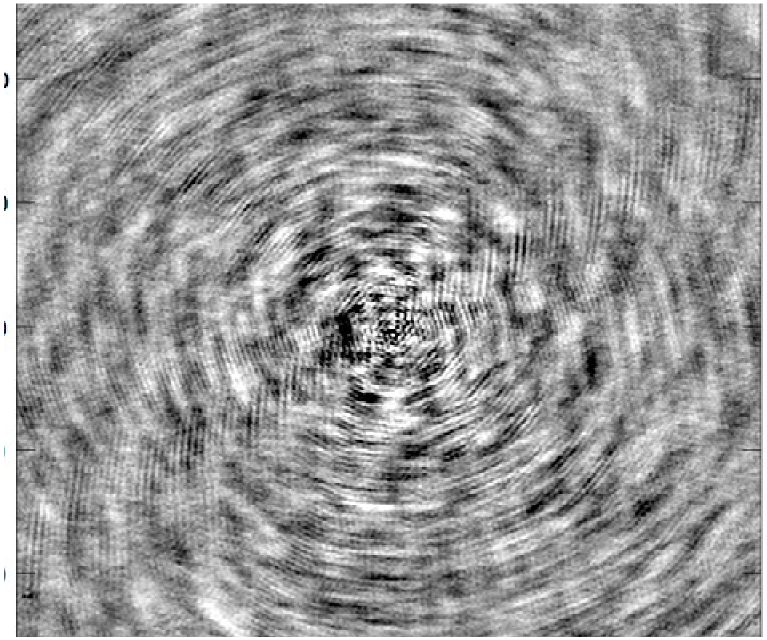
One antenna in error: all times

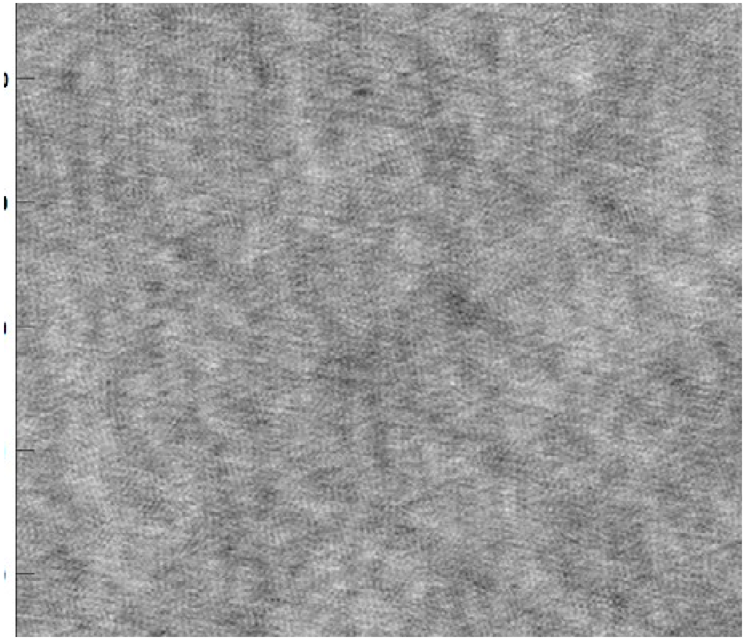


Multiplicative

Can diagnose by dropping one antenna in turn and re-imaging

Non-closing errors

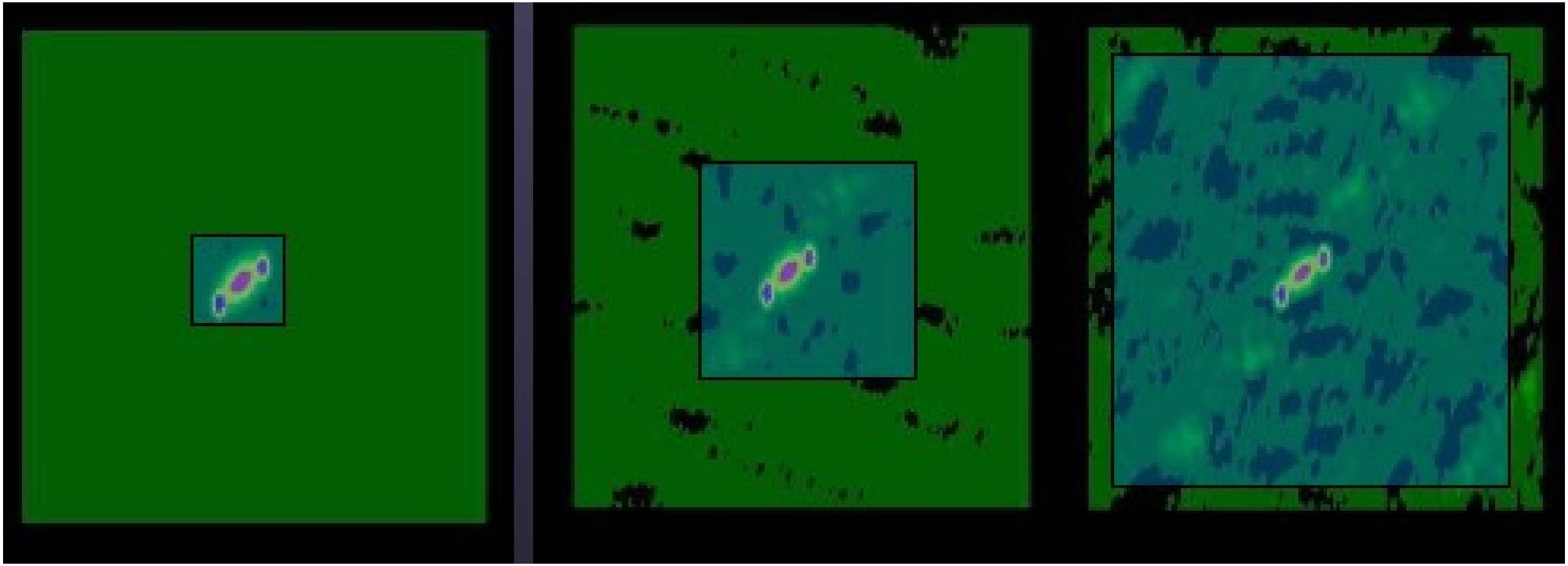




With errors

Without errors

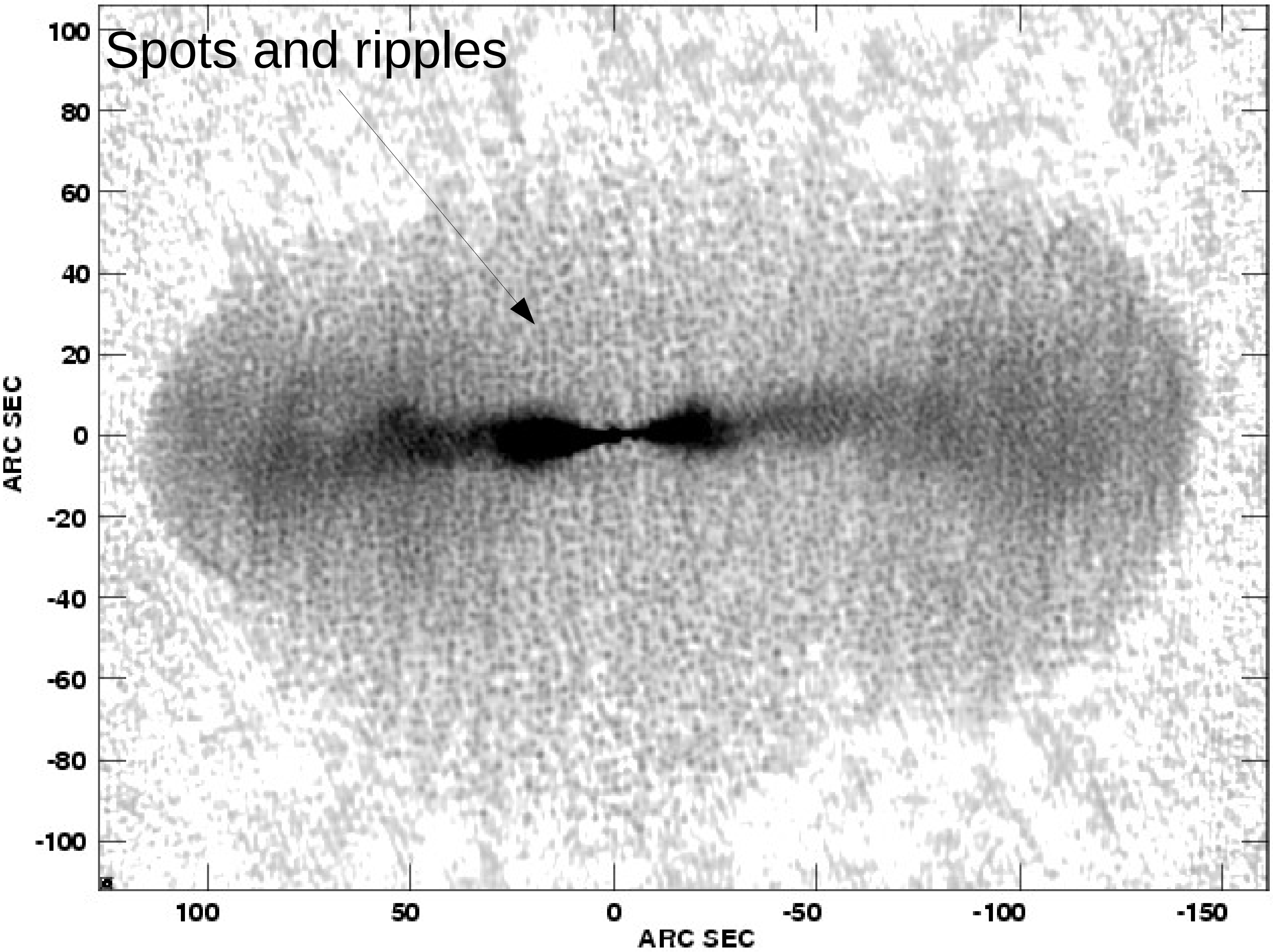
CLEAN boxes too small



Correct Too big Far too big

CLEAN functions best when the area in which it finds components is restricted ("compact support")

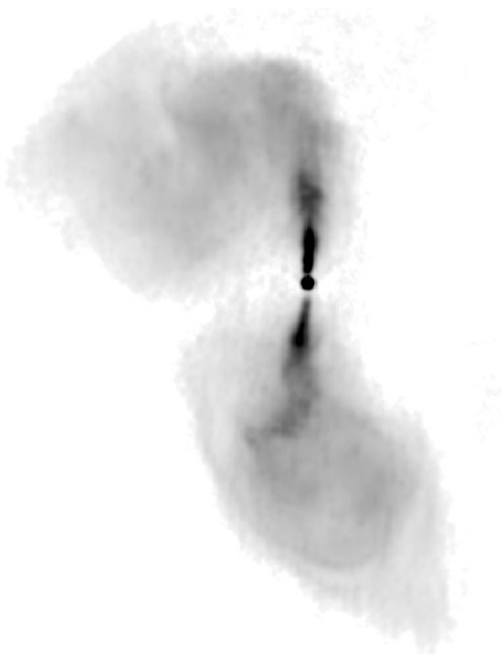
Deconvolution Errors



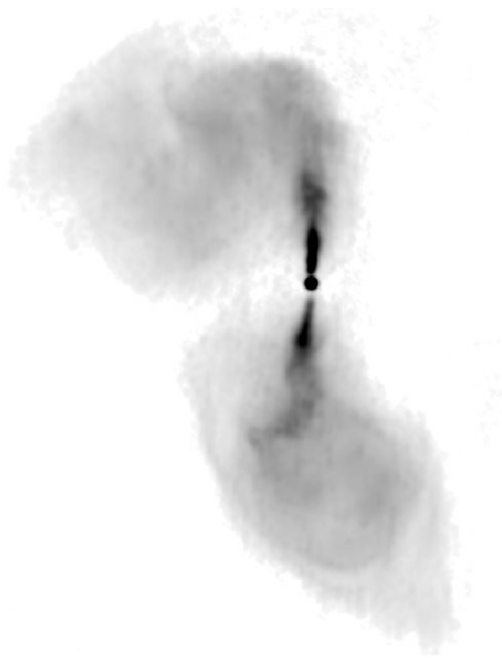
VLA A+B+C configurations. Short spacings OK, but with poor A-configuration coverage

Conventional CLEAN fails: try multi-resolution CLEAN or MEM or reduce the resolution

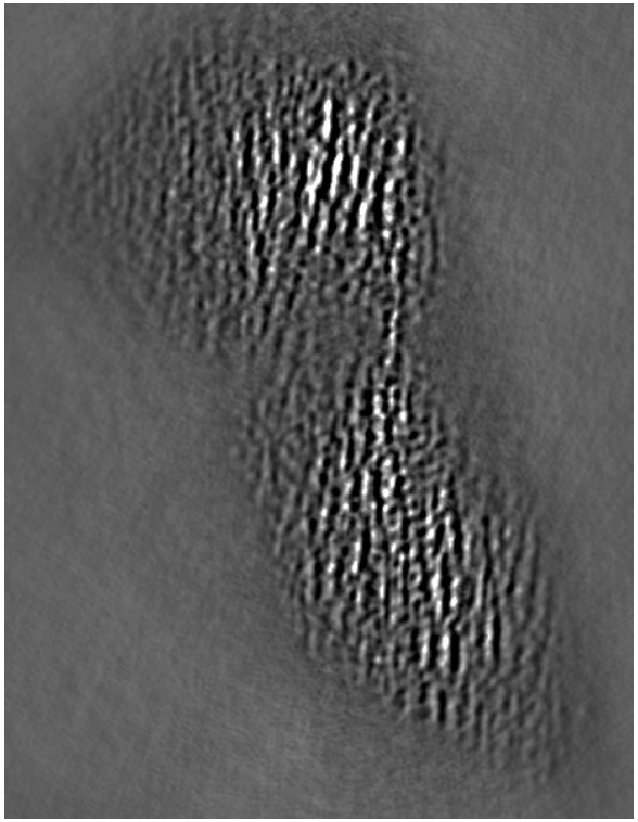
Multi-scale CLEAN helps



1-scale



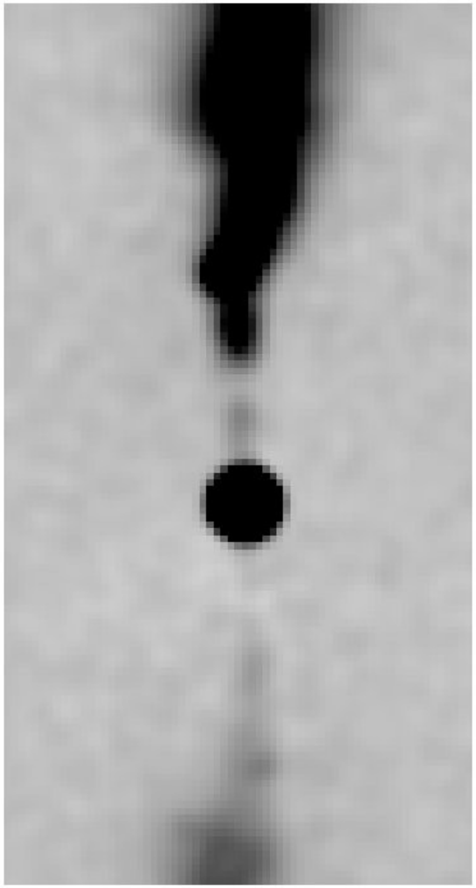
3-scale



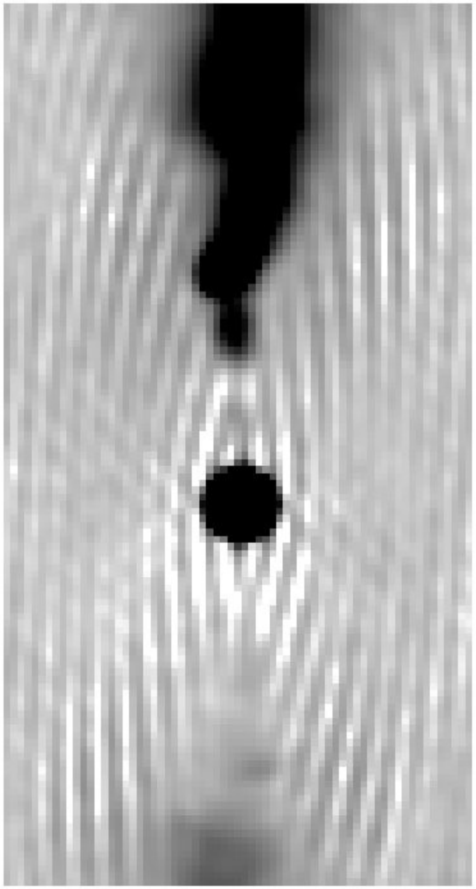
1-scale – 3-scale

Multi-scale CLEAN has removed a high-frequency ripple

Point source not on a pixel



Compact source
on a pixel

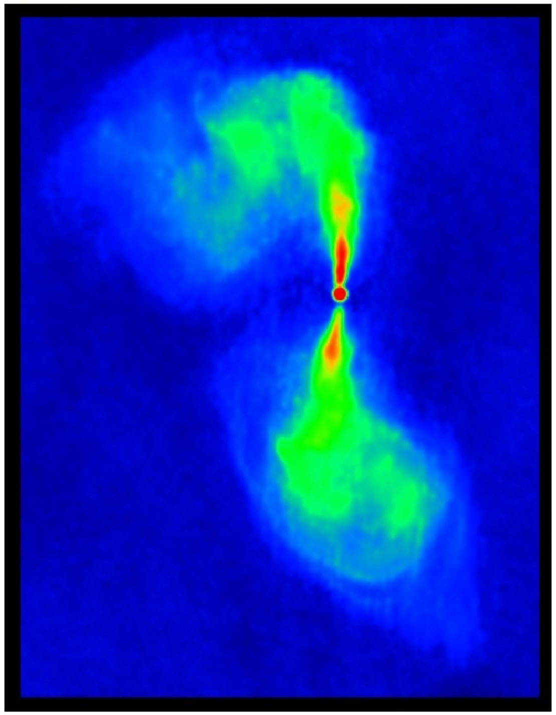


Mis-centred
by 0.5 pixel

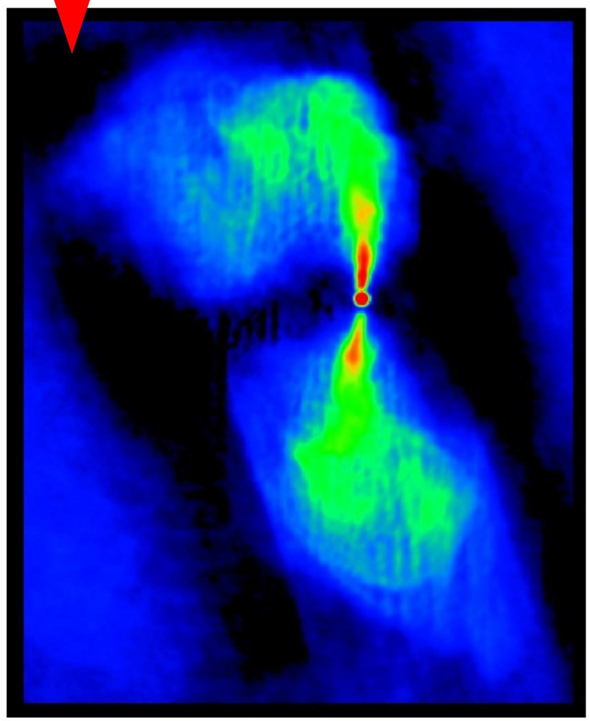
Missing short spacings

“Bowl”

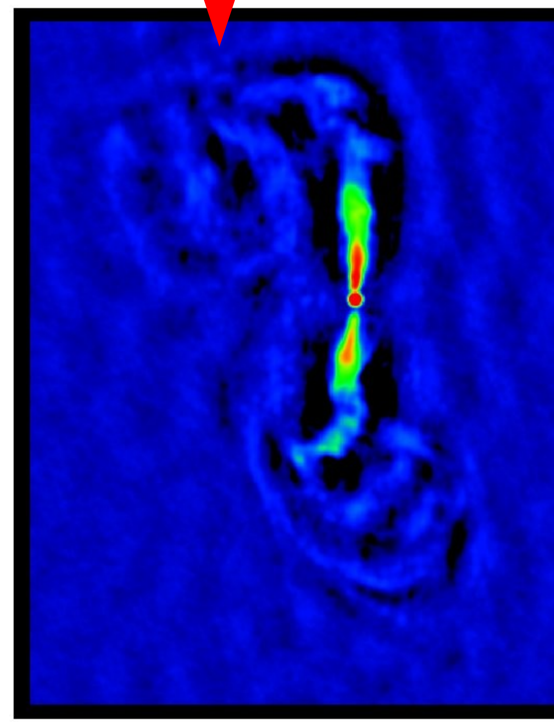
Messed up diffuse emission



uv range < 225 kλ

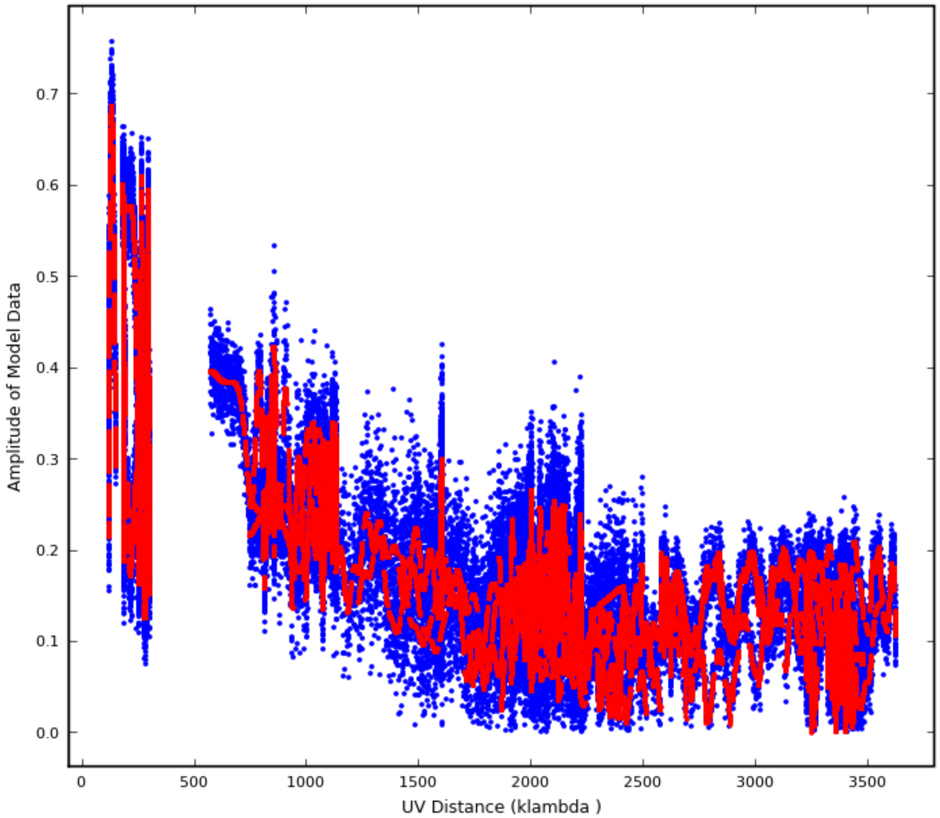


uv range 2 – 225 kλ



uv range 10 – 225 kλ

Does the model fit the data (1)?



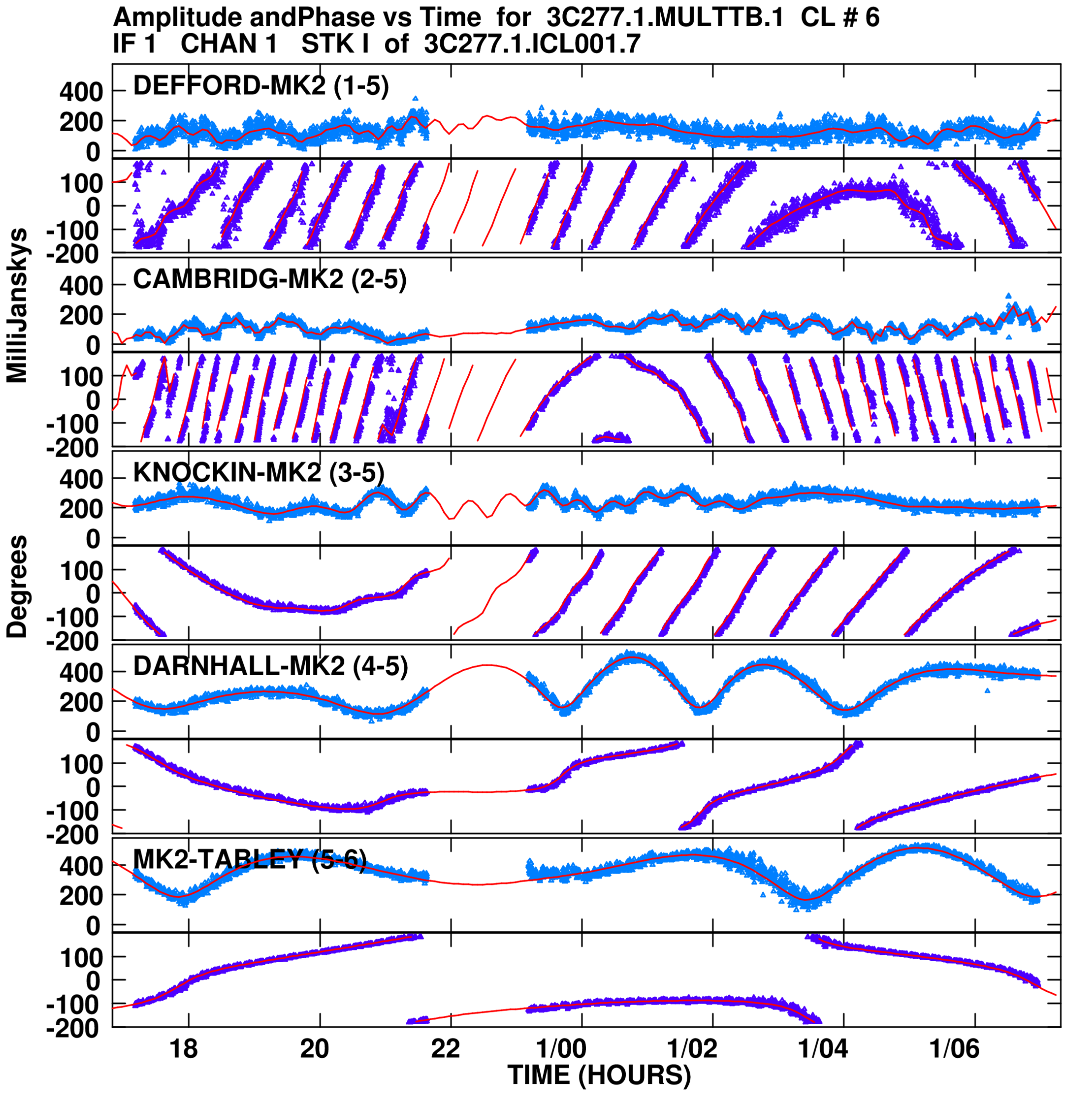
Amplitude fits

Plot amplitude against uv distance

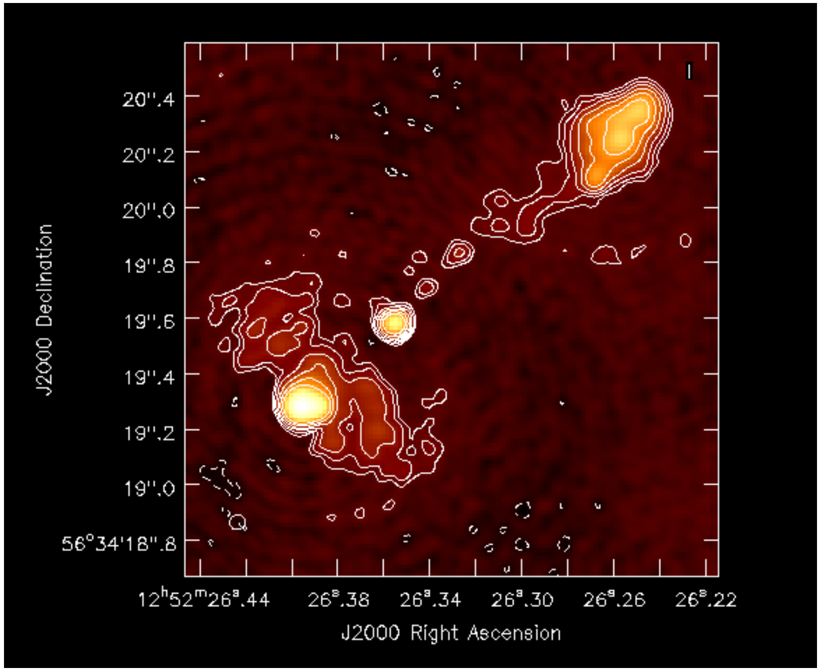
Data

Model

Does the model fits the data (2)

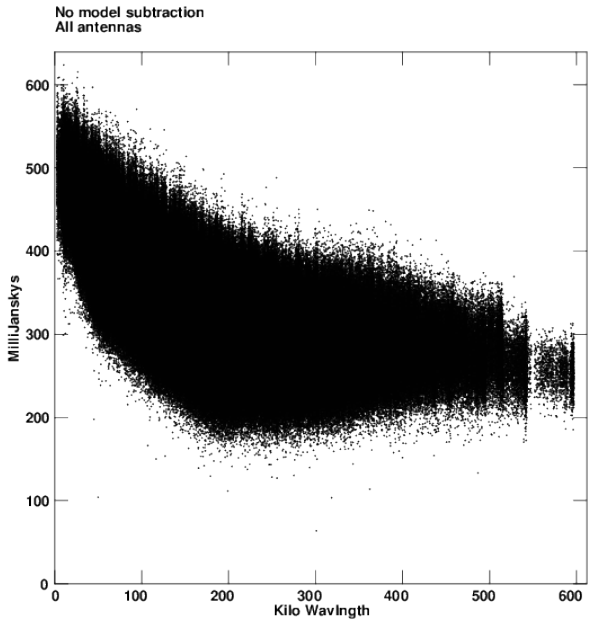


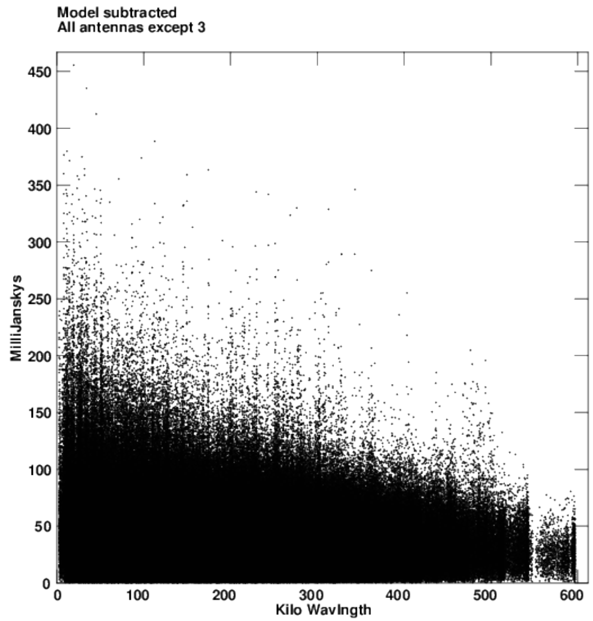
Phase fits

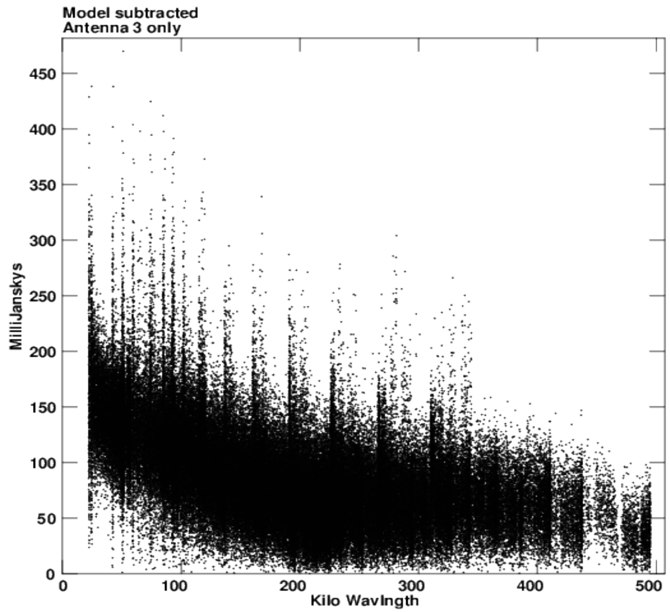


Does the model fit the data (3)

The case of a bright source with a low-level error







Error present (all antennas plotted) Nothing obvious

Model subtracted (all except antenna 3 plotted). Some discrepant data

Model subtracted (antenna 3 only). Bad data clearly visible.

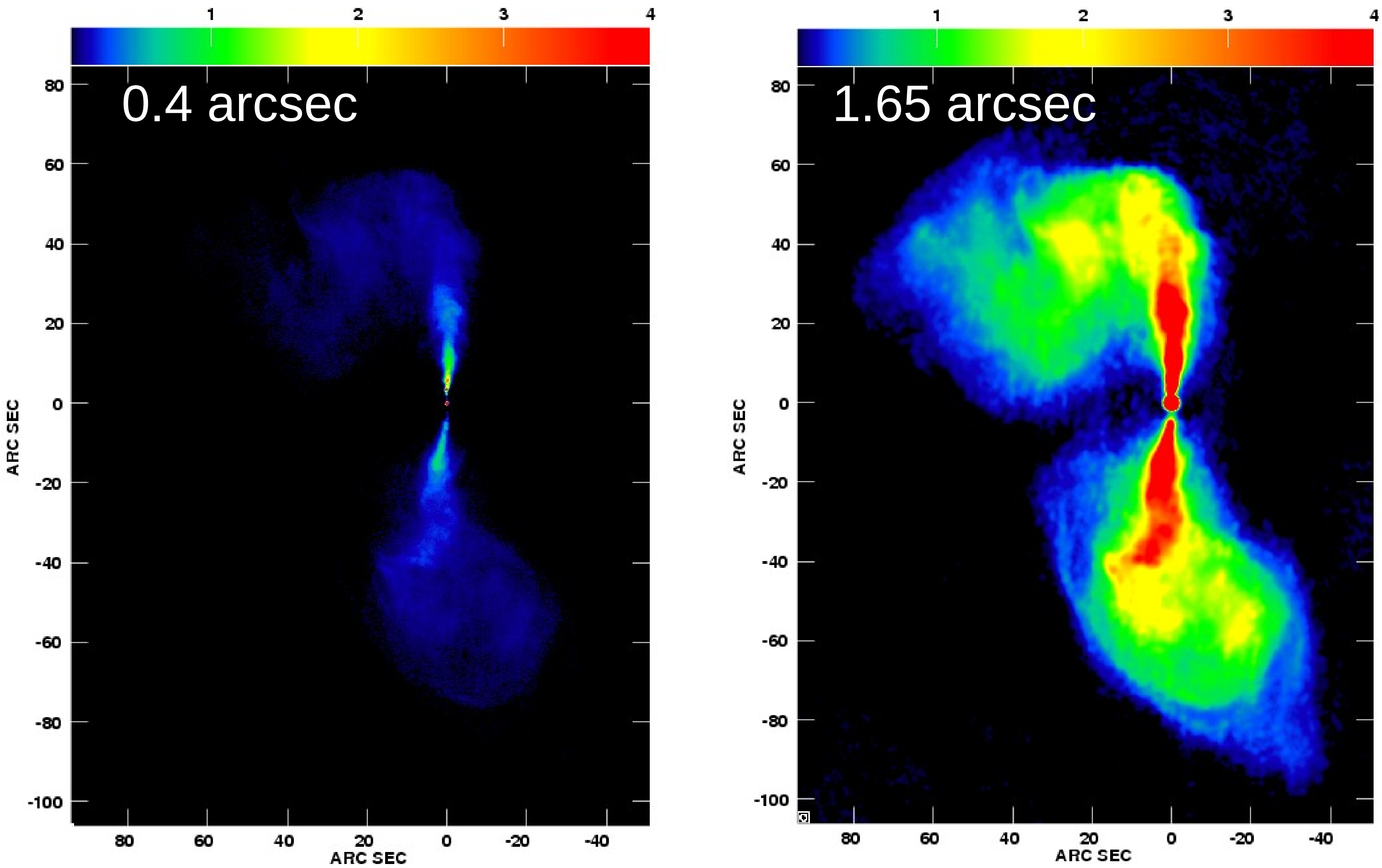
Summary of error recognition

- (u,v) plane
  - Look for outliers (high or low) – flagging tutorial
  - Subtract the best model – check residuals in amplitude and phase
- Image plane
  - Do the defects look like the dirty beam?
  - Additive or multiplicative?
  - Symmetric or antisymmetric?
  - Relate to possible data errors?
  - Missing spacings?
  - Deconvolution errors?
- If in doubt, simulate with realistic errors

Image Analysis

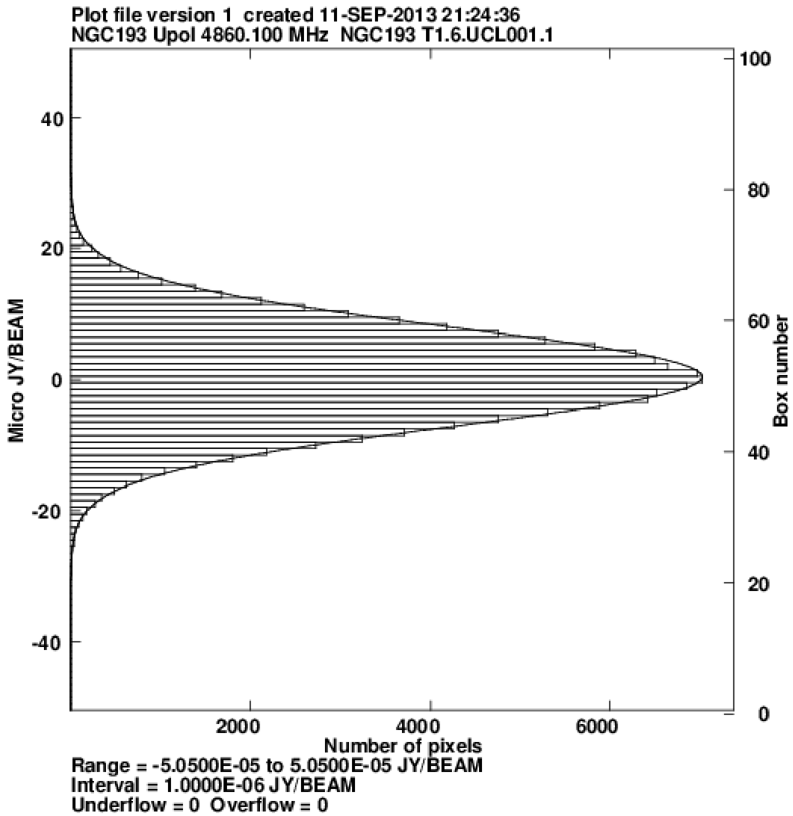
- Given: a well-calibrated dataset producing a high-quality image (or, in general, image cube)
- How can we extract scientifically useful numbers?
- This is a very open-ended problem, depending on:
  - image complexity
  - scientific goals
- Selected topics (excluding spectral line):
  - Picking the correct resolution
  - Parameter estimation
  - Comparing images: spectra, polarization etc.; registration
  - Getting images into your own code

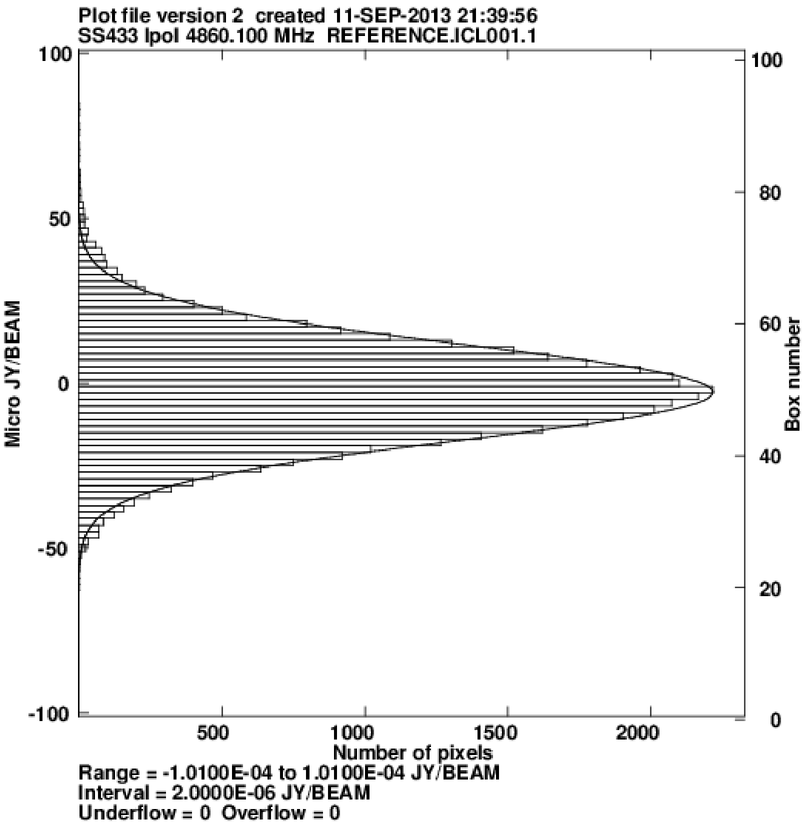
Match the resolution to the problem



Exactly the same dataset, imaged with different Gaussian tapers

Measure the off-source noise distribution





Good case: rms = 7.5μJy; Gaussian noise with zero mean

Excess noise above Gaussian tail

Estimating the flux density of an extended source

- Use a **low-resolution** image, cleaned deeply
  - The beam areas of the restored CLEAN components and residuals are not the same in general.
- Sum the flux density over some area (rectangular, polygonal, ...) – casa imstat, viewer.
- Remember that the total flux density is $\Sigma I/B$, where B is the integral over the beam. For a Gaussian,

$$B = \pi(\mathrm{FWHM/pixel})^2/4 \ln 2.$$

The reduction packages will calculate this for you.

- The reason is that the images are normalised so that a point source of flux density 1 Jy gives a **peak** response of 1 Jy/beam on the image.

Component fitting

- Image plane
  - Assume source components are ~Gaussian
  - Deep cleaning restores images with Gaussian beam
  - Size estimation quite straightforward
  - casa imfit (although AIPS JMFIT is more reliable)
- u-v plane
  - More accurate for small numbers of ~point-like sources
  - Can fit to models that are slightly more complex than point-like
  - Accounts for imperfect sampling; noise distribution may be better understood, but ...
  - ... no good for very complex brightness distributions
- Error estimates
  - ad hoc
  - From fitting routines
  - By simulation

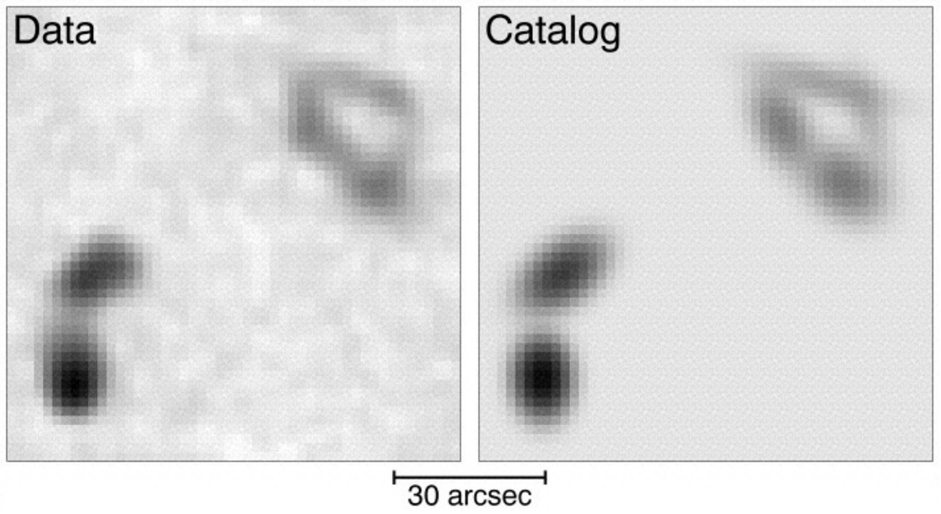
Error estimates for Gaussian fits

- Definitions
  - P = peak component flux density
  - σ = image rms noise
  - θ_B = CLEAN beam size
  - θ_{obs} = component size
  - S = P/σ = signal/noise
- rms errors
  - Error on peak flux density = σ
  - Position error = $\theta_B/2S$
  - True component size $\theta = (\theta_{obs}^2 - \theta_B^2)^{1/2}$
  - Minimum measurable component size = $\theta_B/S^{1/2}$
    - S/N >100 is needed to determine a size $<\theta_B/10$.

Assumes uncorrelated Gaussian random noise in the image plane – not always true.

Automated image fitting

- Automated routines can be used to locate and fit sources (essential for surveys). SAD in AIPS is a good example. casa is weak in this area.
- Also adapt routines used in optical astronomy (e.g. Sextractor)
  - beware incorrect noise model
- Often worthwhile to make Monte Carlo simulations to assess realistic errors in position and (especially) flux density (e.g. add model point sources).



Automated fitting of images from the FIRST survey.

Basic image arithmetic

- Standard packages allow mathematical operations on one, two or occasionally more images (casa immath allows a lot of flexibility):
  - Sum, product, quotient, ...
  - Polarized intensity and position angle from Q and U
  - Spectral index α ($S \propto \nu^{\alpha}$)
  - Faraday rotation measure
  - Optical depth
  - ....
- Can also propagate noise and blank on input values or s/n and use masks
- Other image manipulations (spatial filtering, etc.) are also possible
- Current packages are poor at fitting functions of frequency to images at more than 2 frequencies, although they are getting better for standard operations.
  - casa rmfit, spxfit

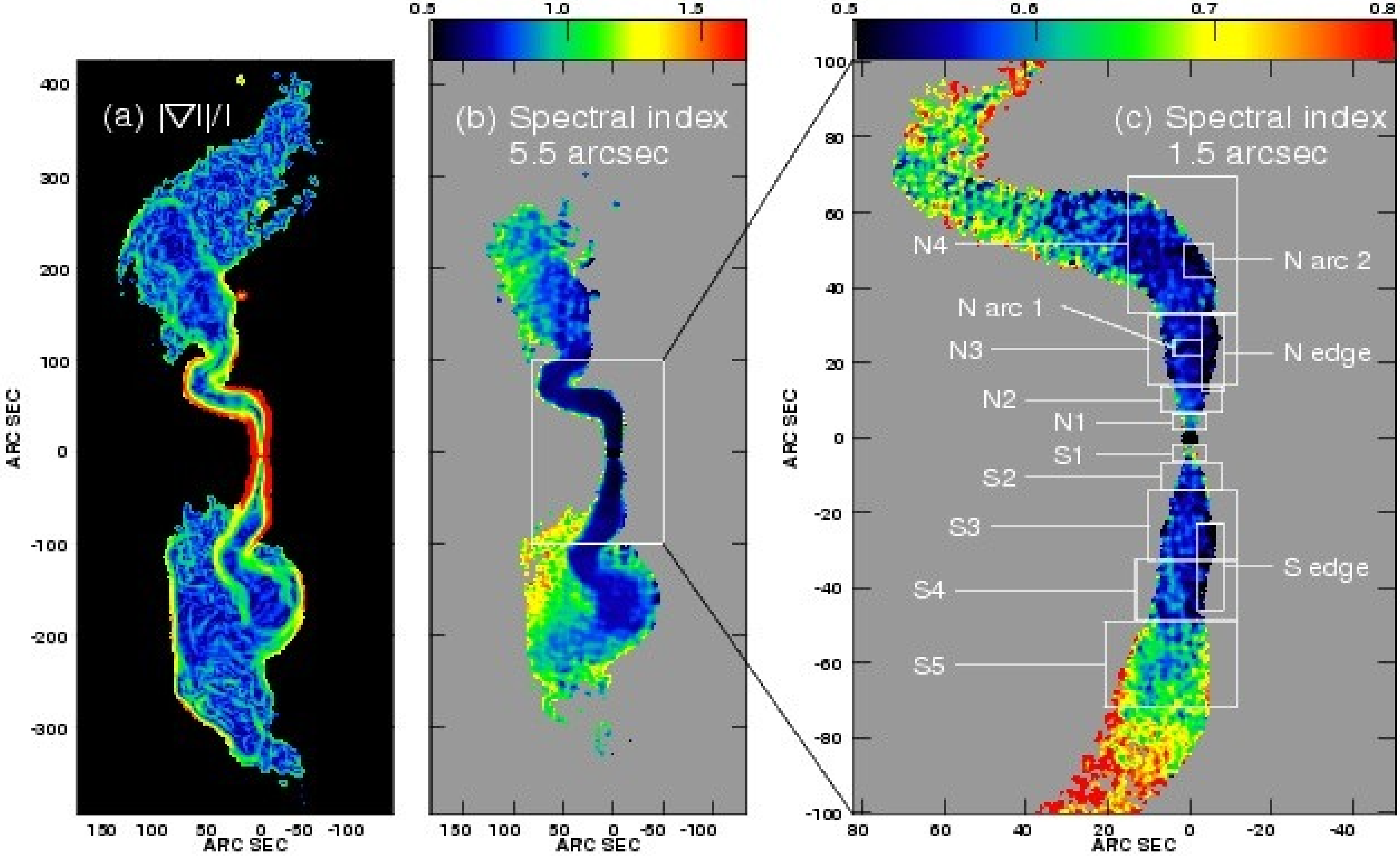
Basic image manipulation

- Often useful to make subimages
  - casa imsubimage
- Smooth images (e.g. if the restoring beam is not quite what you want)
  - Gaussian or user-supplied kernel
  - casa imsmooth
- Regridding images
  - Often needed (e.g.) to align two images with different coordinates, pixel sizes for comparison
  - casa imregrid

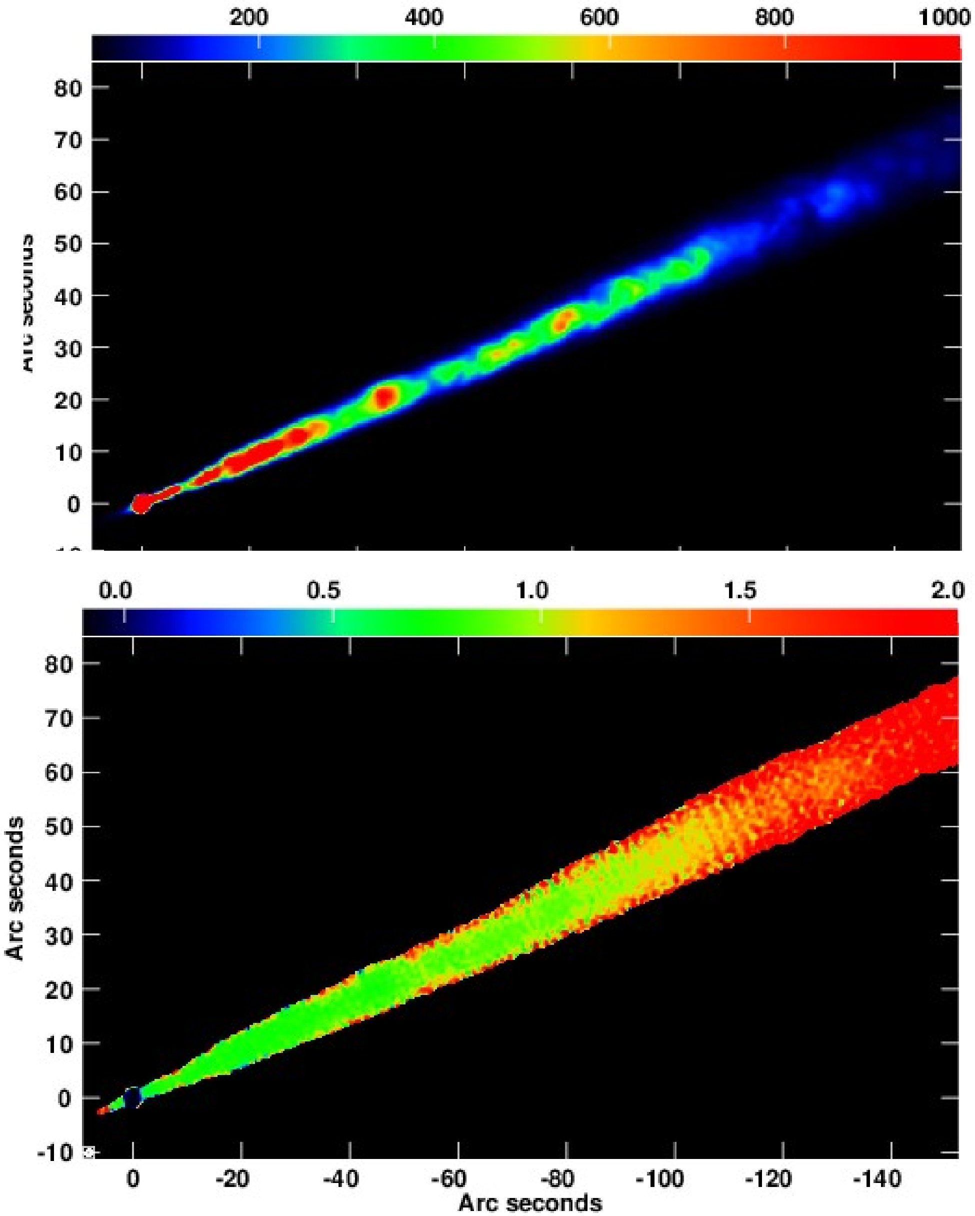
Comparing images at different frequencies

- Match the resolutions
  - Pick appropriate weighting and Gaussian taper to get approximately the same dirty beam FWHM
  - Restore with the same beam
  - Precise matching of coverage is not necessary
  - Making the coverage of one dataset worse to match the other one often leads to disaster
- Error propagation
  - Gaussian random noise in the image plane is the best case: you can only do worse
  - Be careful near edges of the source and sharp brightness gradients

Spectral index and gradient filter



In-band spectra



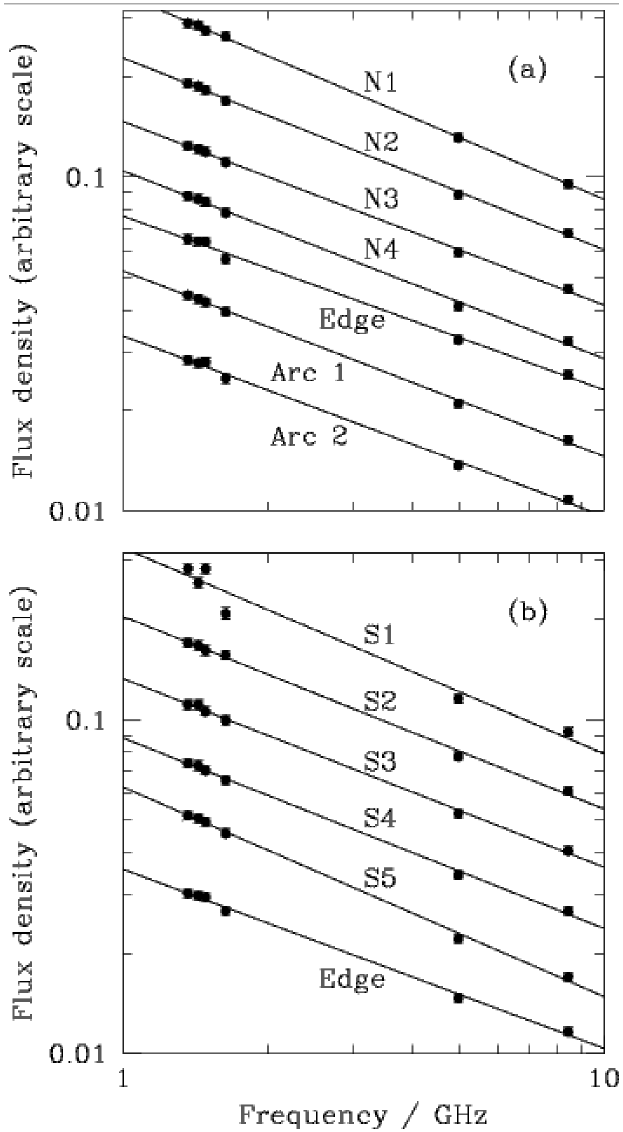
Total intensity at the reference frequency from casa clean with nterms = 2

Spectral index from clean, corrected for primary beam using widebandpbcor.

(sign convention flipped from casa $I \propto \nu^{-\alpha}$)

2 GHz bandwidth; 5 GHz centre frequency, Jansky VLA

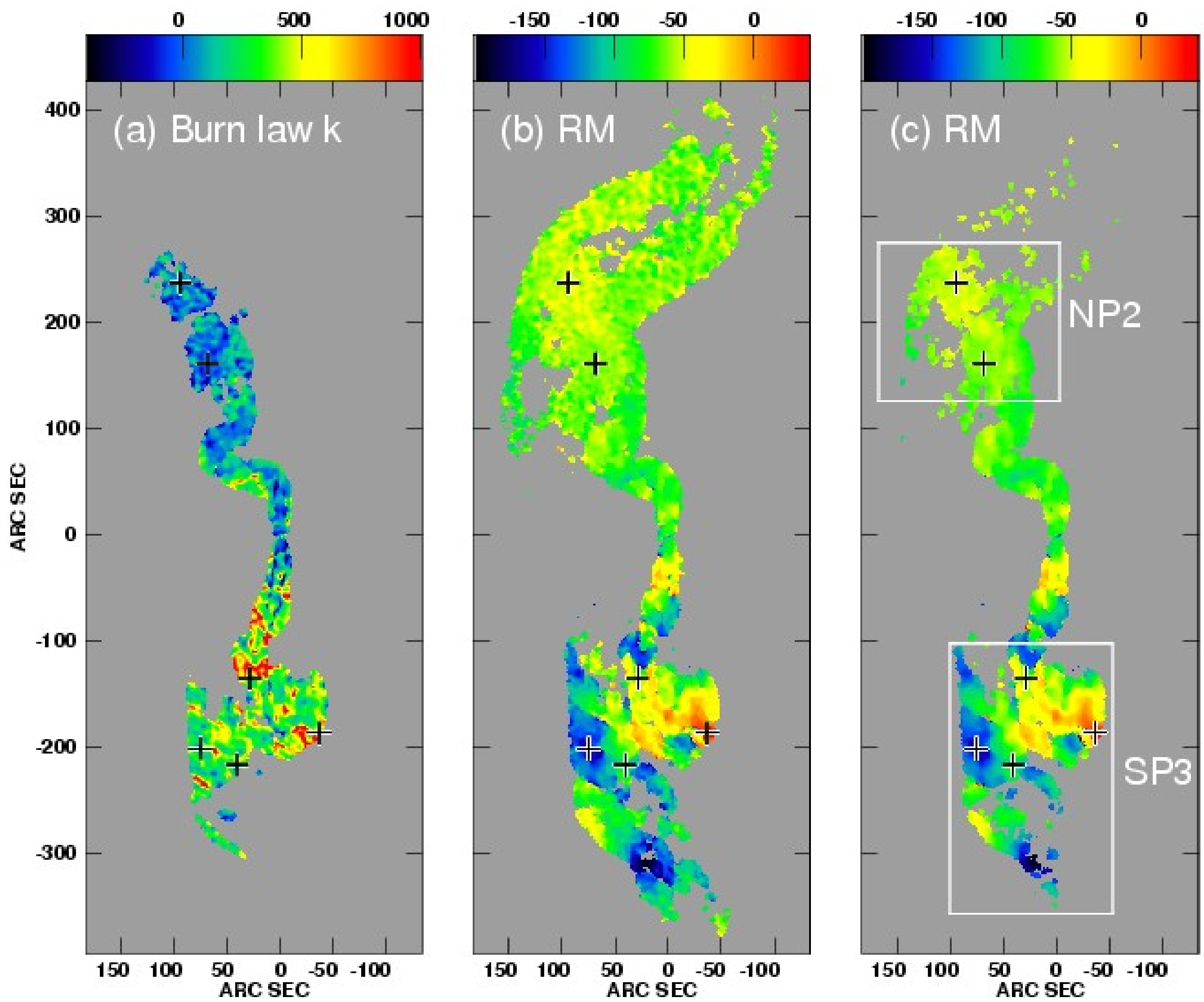
Integrated spectra



Spectra derived by integrating the flux densities over the boxes shown on the previous slide.

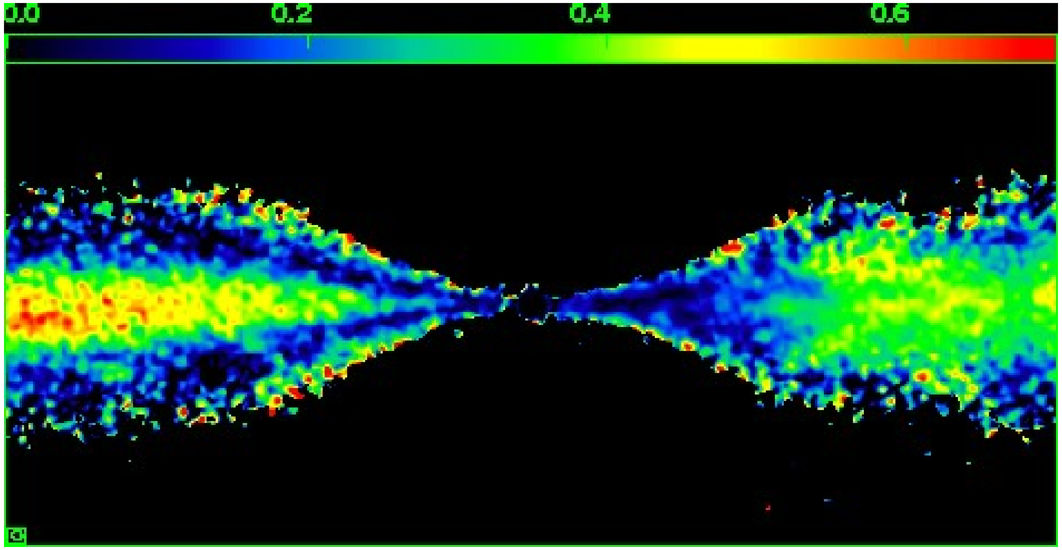
Note the slight flux scale error at 5 and 8.4 GHz.

Rotation Measure

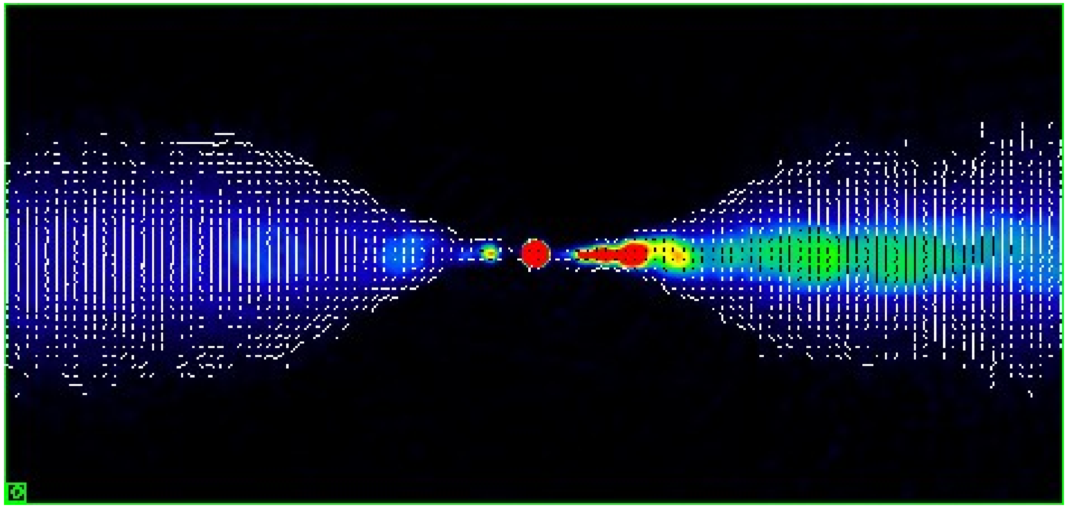


$p \propto \exp(-k\lambda^4)$ $PA = PA(0) + RM\ \lambda^2$

Displaying polarization data

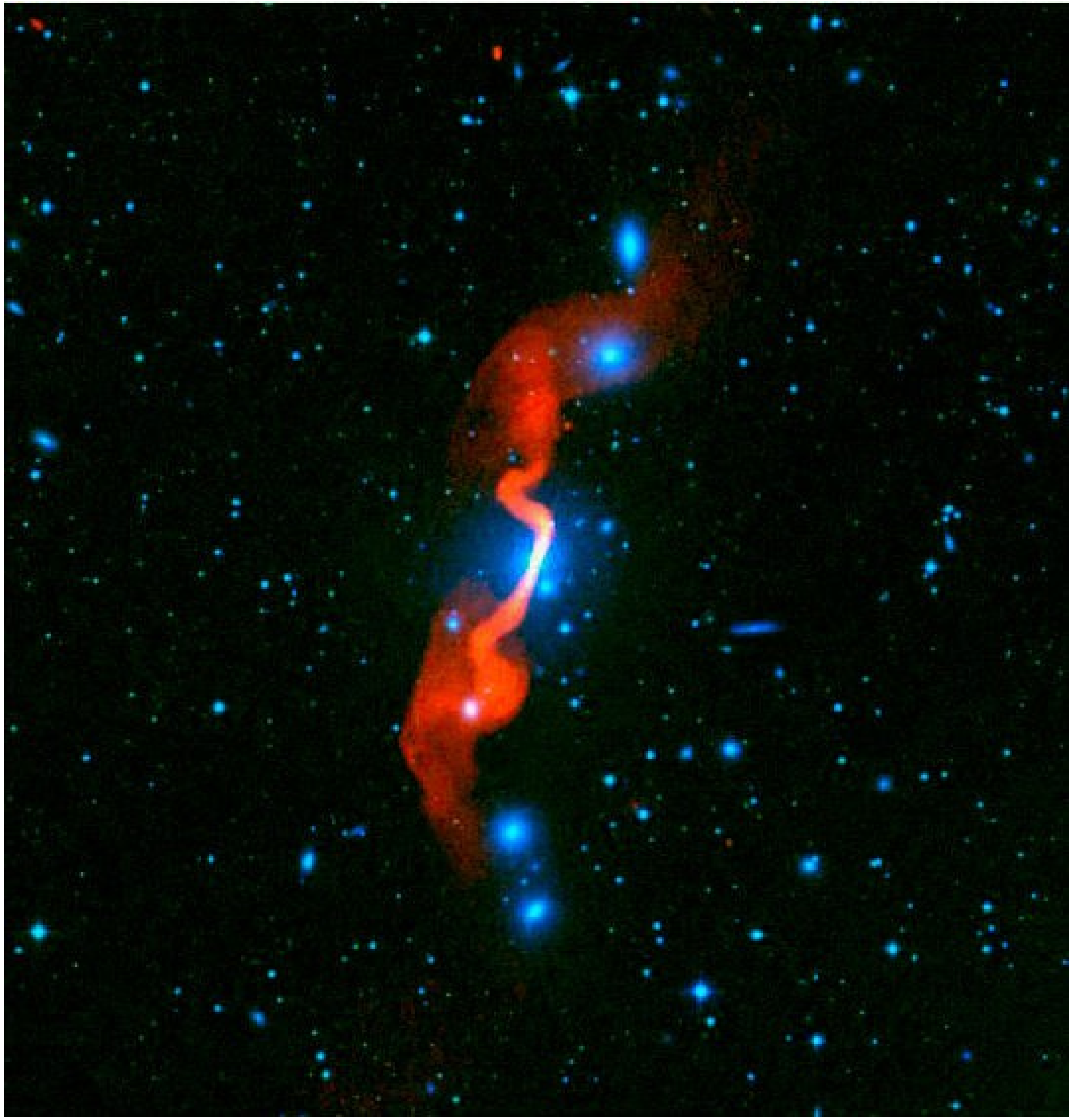


p = P/I (P corrected for Ricean bias)



Vectors; lengths ∝ p, directions along **E**-field direction + 90^0, after correction for Faraday rotation.

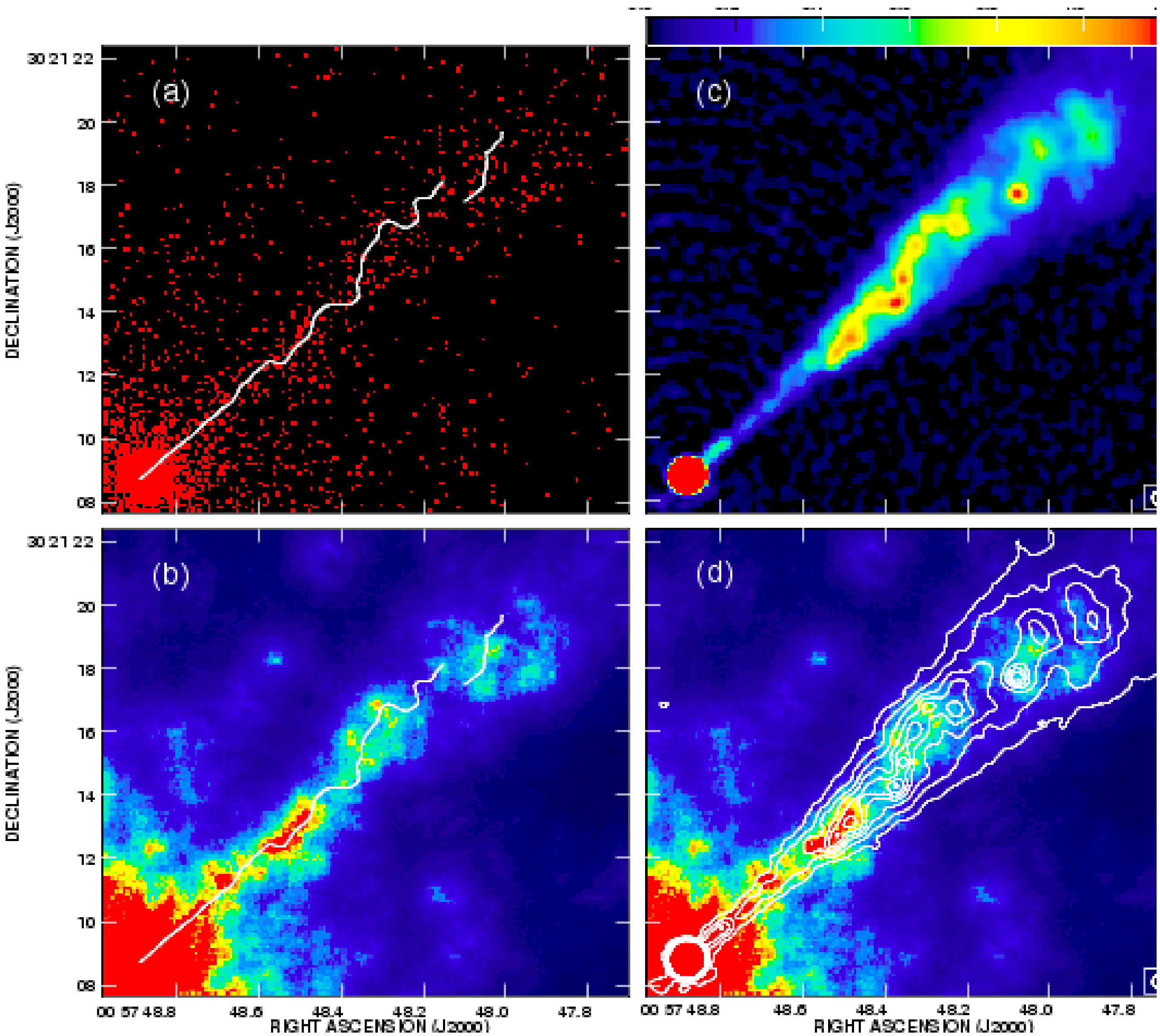
Regridding: radio – optical overlay



1.4GHz radio (VLA) in red

Optical (DSS) in blue

Regridding: radio – X-ray overlay



Radio (0.35 arcsec, 4.9 GHz, VLA)

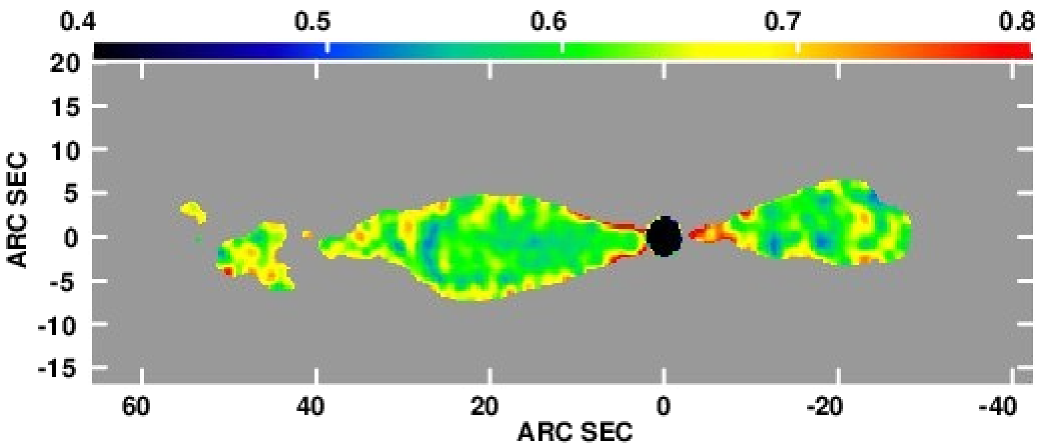
X-ray (0.6 arcsec, 0.5 – 8 keV)

Issues in image registration

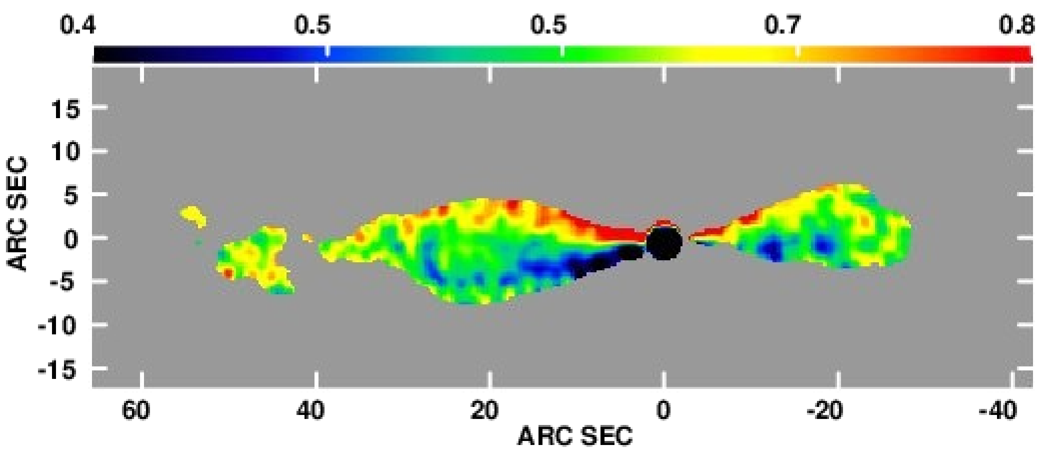
- Rationale for image combination
  - Many astrophysical applications require multiwavelength comparison
  - Proper motions may be important
- Regridding
  - Tools available (casa imregrid)
- Accuracy of registration
  - For purely radio data, ideally:
    - calibrator is close to the target
    - use the same phase calibrator for all observations
  - Watch out for errors from ionosphere, troposphere, antenna positions
  - Use internal references if possible (e.g. point sources in the field)
  - Beware changes in structure with frequency
  - N.B.: images at other wavebands may have less accurate absolute astrometry

Good astrometry is vital, but is not the subject of this lecture

Registration errors



Spectral index between 1.365 and 4.9 GHz

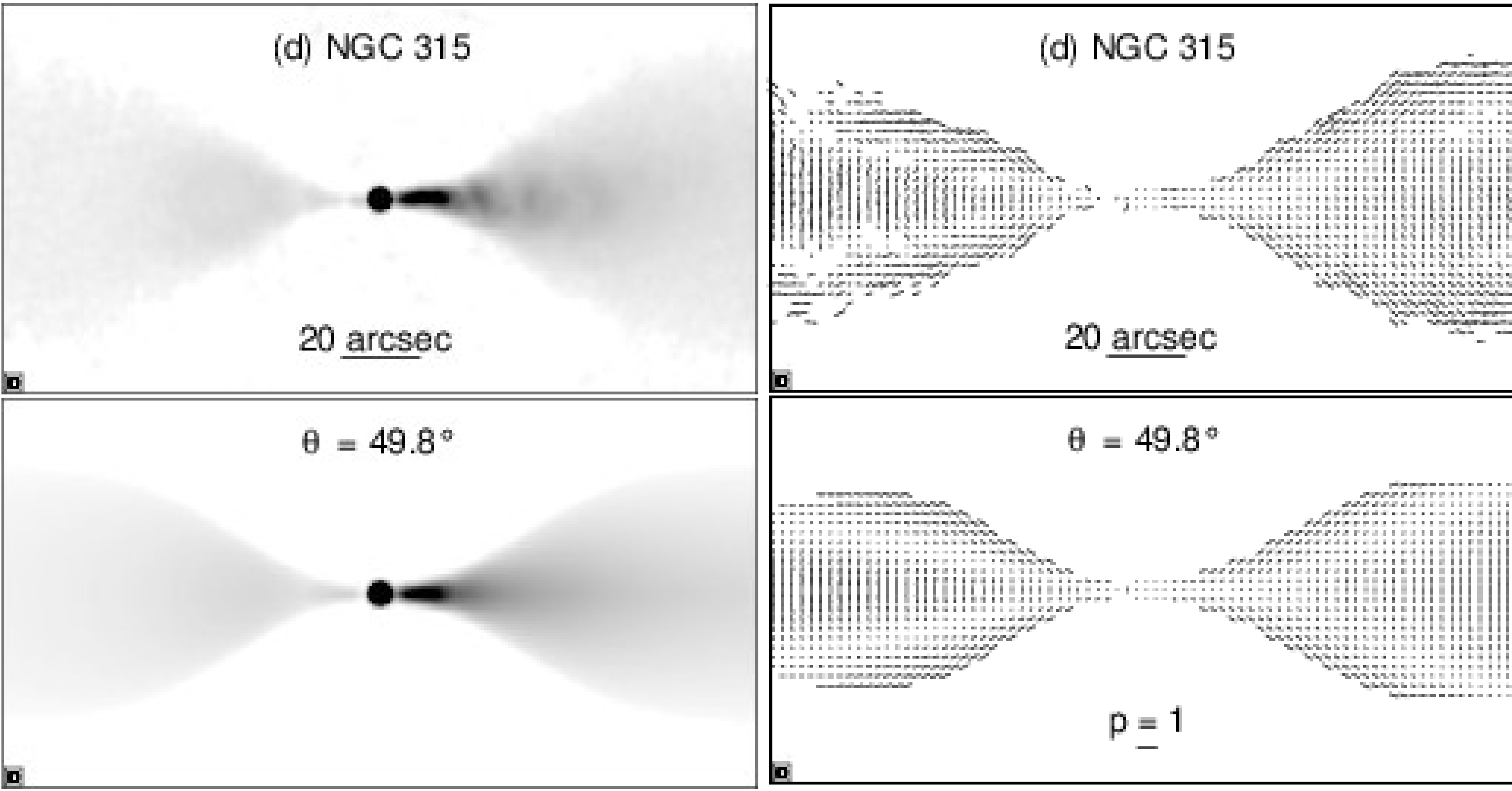


Relative shift of 0.2 FWHM

Getting data into your own code

- If the standard packages do not do what you want, do not be frightened of importing images into your own programs.
- FITS interchange standard
  - can be read and written by all radio astronomy packages
  - mostly standard for images (uv less so)
  - well documented interfaces to common programming languages (python, C, fortran, IDL, ...)
  - and even to graphics manipulation packages (gimp)
- Easy read/write from casa to python arrays

Example: jet modelling



Total intensity

Vectors p/apparent B field

Summary of continuum image analysis

- Match the resolution to the problem
- For simple images, fit component parameters and derive errors
- Image comparison
  - Simple mathematical operations are easy
  - Regridding and interpolation often required
  - Registration is an issue
  - Noise propagation
- Straightforward to read images into your own code for more sophisticated modelling