Radio Astronomy
Lecture 3:
Imaging and calibration, radio emission mechanisms

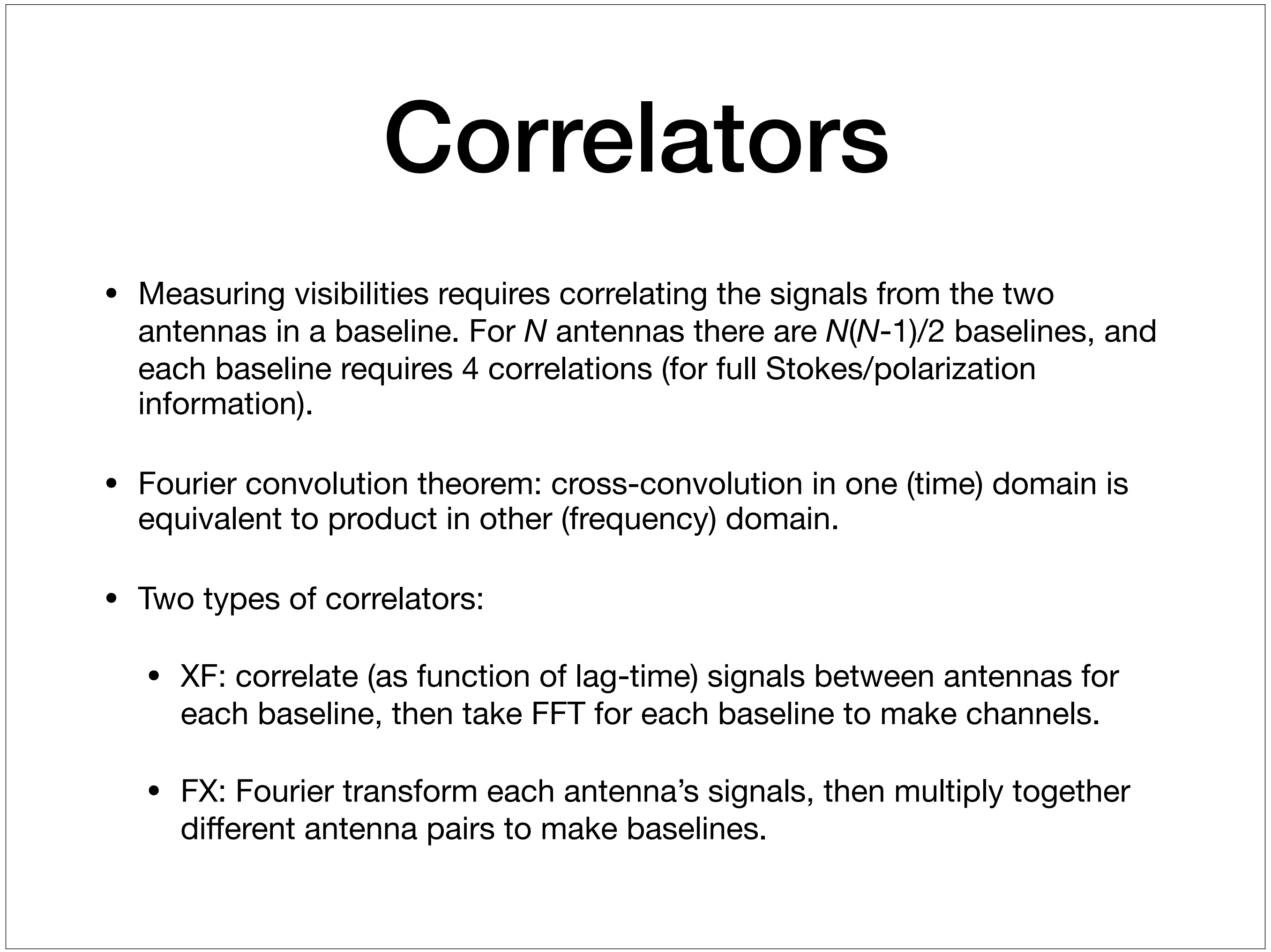
Cameron Van Eck

26 March 2020

- Correlators
- Imaging, weights, deconvolution (Clean), ~~sensitivity/noise, missing scales~~
- Calibration of interferometers
- ~~Radio emission mechanisms~~ (moved to lecture 4)

- Note: today is deadline to decide credit vs audit (drop deadline)

This took more time than expected, so lecture 2 covered up to and including imaging. Remaining topics will be moved to lecture 3 (replacing some of the applications of radio astronomy)



Correlators

- Measuring visibilities requires correlating the signals from the two antennas in a baseline. For *N* antennas there are *N*(*N*-1)/2 baselines, and each baseline requires 4 correlations (for full Stokes/polarization information).
- Fourier convolution theorem: cross-convolution in one (time) domain is equivalent to product in other (frequency) domain.
- Two types of correlators:
    - XF: correlate (as function of lag-time) signals between antennas for each baseline, then take FFT for each baseline to make channels.
    - FX: Fourier transform each antenna's signals, then multiply together different antenna pairs to make baselines.

The correlator is basically a real-time supercomputer for doing these calculations. Data flows in are huge, calculations required are enormous (because of the large number of baselines),

Imaging

- The imaging process creates an NxM pixel image from the inverse Fourier transform of the visibilities. The number of pixels, and the size of each pixel, is set by the user.

- The most efficient method is an FFT algorithm, which produces an NxM pixel image from an NxM grid of points in the *u,v* plane.

- Baselines won’t fall exactly onto this grid, so there’s some **gridding** process which puts the visibilities onto the grid. Visibilities that fall into the same grid point get average together.

Weighting

$$W(u,v) = \sum_i \sum_{j \neq i} W_{ij} \delta(u - u_{ij}) \delta(v - v_{ij})$$

- Applying weights to the visibilities can either occur directly on the individual visibilities, or on the visibility grid.
- Weights have the effect of modifying the synthesized beam, and can be chosen to optimize different aspects of the synthesized beam.

Weighting schemes

- **Natural**: weight by inverse variance on each visibility. Some noise level can be estimated for each grid cell, use inverse square of noise to weight. Minimizes noise level in output image. Tends to down-weight long baselines (lower density of visibilities), reducing resolution.

- **Uniform**: weight all grid cells (with visibilities) equally. Doesn’t account for density/noise of measurements, so higher noise level in image, but doesn’t down-weight long baselines so resolution is good.

Weighting schemes

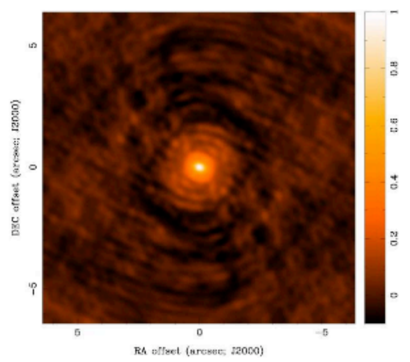
- **Robust (Brigg's):** use an equation that balances between natural and uniform, with a free parameter ('robust') to adjust the balance. Negative 'robust' is more uniform, positive is more natural, zero is balanced.

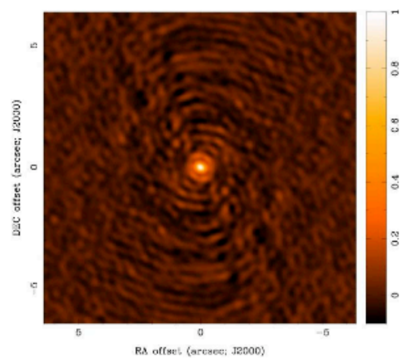
- **Gaussian taper:** down-weight long baselines by a Gaussian function of length. Reduces resolution, but reduces sidelobes in synthesized beam; may reduce noise on small scales. Applied on top of other weighting scheme.

Weighting and Tapering: Image Noise

natural
0.59x0.50

rms=1.0



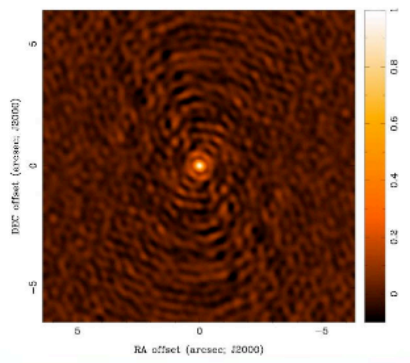


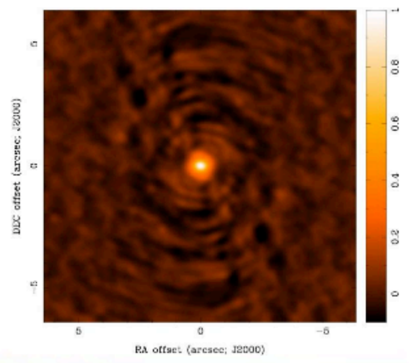
robust=0
0.40x0.34

rms=1.3

uniform
0.35x0.30

rms=2.1





robust=0
+ taper to
0.59x0.50

rms=1.2

NRAO

New Mexico CONSORTIUM

Image credit: David Wilner (CfA), 14th Synthesis Imaging workshop lectures

Multifrequency synthesis (MFS)

- Each baseline has a different length in the *u,v* plane as a function of frequency. So a broadband observation samples more of the u,v plane than a single channel, except that most emission has some frequency dependence.
- MFS assumes some model for the spectrum, and produces images of the model parameters (typically intensity and spectral index).

$$I(l,m,\nu) = I_0(l,m)\left(\frac{\nu}{\nu_0}\right)^{\alpha(l,m)}$$

$$\log(I) = \log(I_0) + \alpha\log(\nu/\nu_0) + \beta\log(\nu/\nu_0)^2 + ...$$

Can specify the number of terms to be used in MFS imaging (called Taylor terms, because it's a Taylor expansion in the logarithm). N=1 or 2 is most common.
The advantages of MFS are threefold: first, it lets you use all the information you have in producing your image, thus minimizing noise level; second, it can directly output relevant quantities (flux and spectral index) without need for extra processing; third, it gets the best resolution because it includes the baselines with the longest *uv* distance. Downside: doesn't work for emission that deviates from power-law (i.e., spectral lines and polarization).

Tangential: integrated fluxes are hard to measure

- Uncleaned image contains zero net flux density. (Can you figure out why?)

Think for a moment before going to the next slide.

Tangential: integrated fluxes are hard to measure

- Uncleaned image contains zero net flux density. (Can you figure out why?)

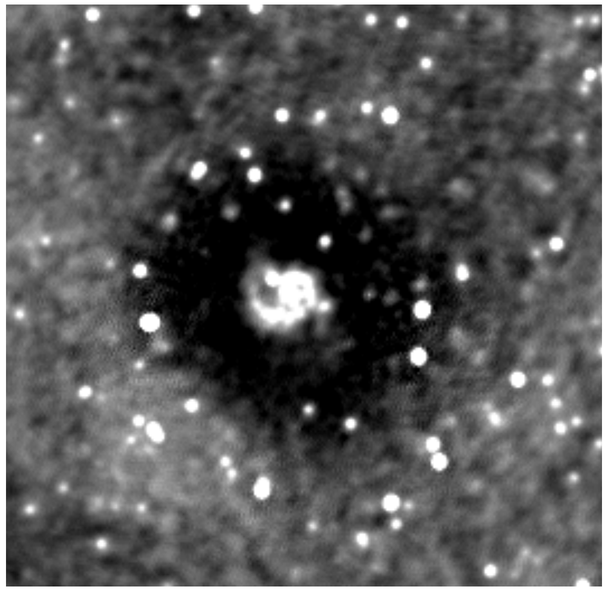
- Every sine wave used to build the image has zero mean. The relative phases lets signal add up in certain places, but it's balanced with negative intensity spread out over other places.

- The mean should come from the zero-frequency Fourier component (origin in *u,v* plane), but an interferometer doesn't measure that.

- The synthesized beam integrates to zero (over whole image). But the Gaussian clean beam doesn't!

Tangential: integrated fluxes are hard to measure

- Large extended objects will have a ‘negative bowl’ if not all the flux is cleaned. And cleaning extended intensity is hard.
- Integrating over some aperture becomes very sensitive to whether the aperture includes the negative bowl.



The only solution I know of for this is multi-scale clean, which does a better job of cleaning extended sources.

Deconvolution

- The convolution between the true sky intensity and the synthesized beam is a pain because it limits our ability to get information from an image. Especially, it limits *dynamic range*: the ability to see faint things beside much brighter things.

- We'd love to deconvolve the image: remove the beam effects and get at the true sky. But there is intrinsic information loss, because of the missing parts of the *u,v* plane. Best we can do is approximate based on some assumptions.

Deconvolution

- Deconvolution algorithms are attempts to accomplish this in different ways, with different assumptions.
- There’s probably 100 different algorithms published, but nearly none have caught on because very few people release useful software for using them (or get them built into CASA).
- The only really common algorithm is Clean and its variations.

(Högbom) Clean

- An iterative algorithm that assumes the sky is made of a collection of unresolved (point) sources, find the location of those point sources, and removes the synthesized beam from them and replaces it with a more desirable beam shape.

- Basic Clean works purely in the image plane, but there are variations that use the visibilities (more on that later).

- Many free parameters and variations: number of iterations, gain (flux removed per iteration), thresholds (how faint to clean to), clean boxes (restricting where clean can look for sources), multi-scale clean, etc.

Named after Högbom 1974, who defined the Clean algorithm.

Clean Loop

- Start with a *residual image* (initially containing the ‘dirty’/convolved image), and a blank *clean component* image. From there, iterate:
    - Find the *brightest* point in the residual image. Assume that’s the location of a (point) source.
    - Shift the synthesized (dirty) beam to that location, scale by the peak intensity times a *gain* factor (typically ~10%, but some people like higher), and subtract this from the image.
    - Add a *clean component*, a delta function at the location of the peak, with intensity equal to the peak times the gain, in the clean component image.
- Repeat until it reaches the iteration limit OR there are no pixels above some intensity *threshold*.

Clean final steps

- Create a *restored image*: take the clean component model (made of delta-functions), and convolve with a *clean beam* (typically a Gaussian with size/area equal to the convolved beam). Add the residual image to this.
- Apply primary beam correction if desired.
- Convolving back to the same size is necessary for a few reasons:
    - Preserves the intensity scale: intensity in Jy/beam requires that the beam size be properly defined for all emission.
    - Intensity below the clean threshold may be significant and needs to be included for completeness.
    - Errors in the clean components may move intensity around.

Gaussian clean-beams are used because they have no sidelobes and nicely trend towards zero.
The primary beam correction is not part of the Clean algorithm, but Clean has to be run before primary beam correction.
There is a technique called ‘super-resolution’ where you restore with a smaller Gaussian or don’t restore at all, but it has complications that most people don’t want to deal with.

clean Example

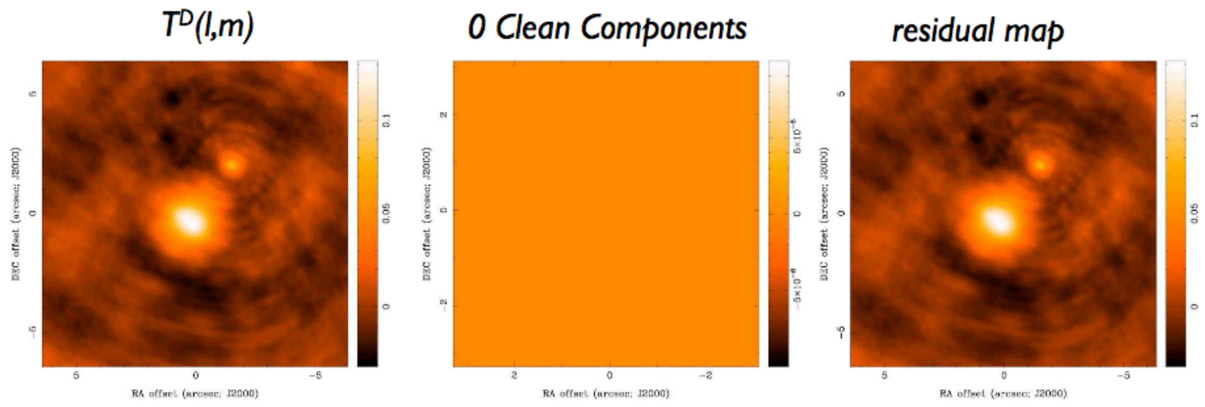


Image credit: David Wilner (CfA), 14th Synthesis Imaging workshop lectures

clean Example

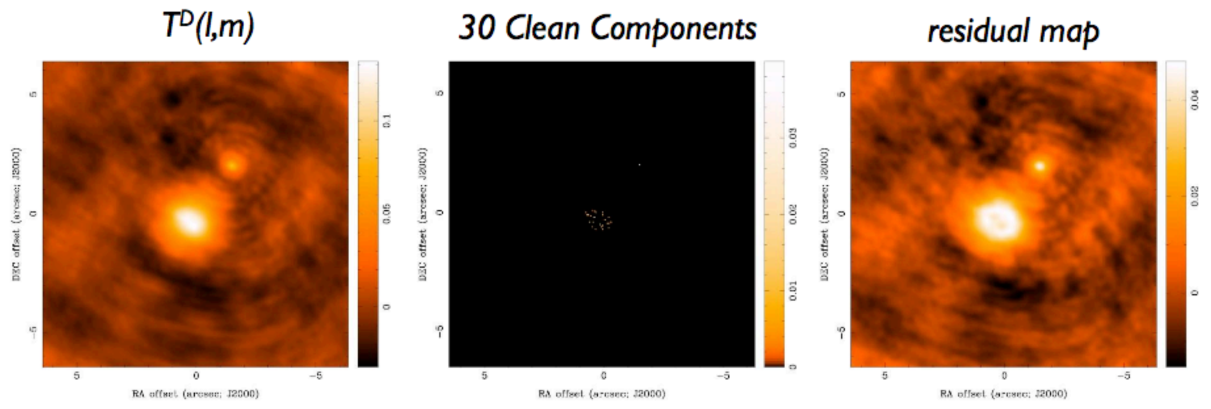


Image credit: David Wilner (CfA), 14th Synthesis Imaging workshop lectures

clean Example

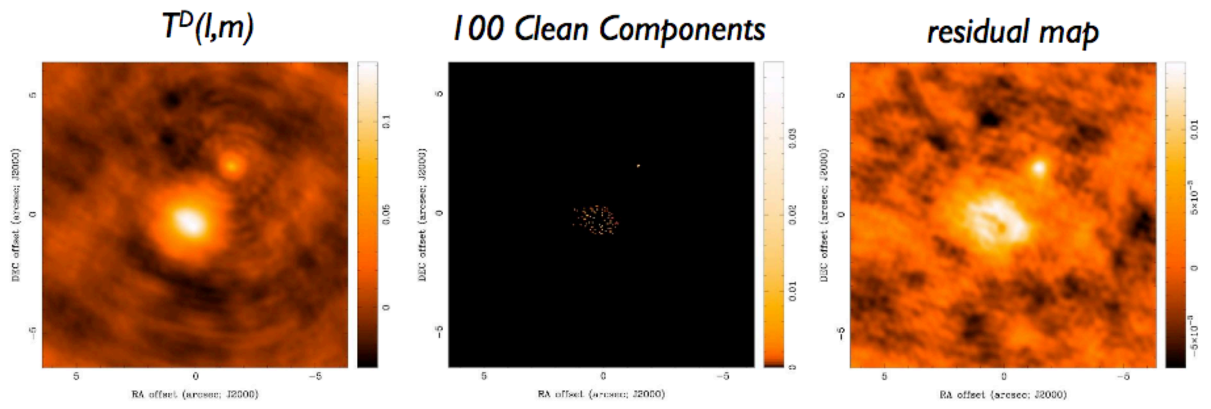


Image credit: David Wilner (CfA), 14th Synthesis Imaging workshop lectures

clean Example

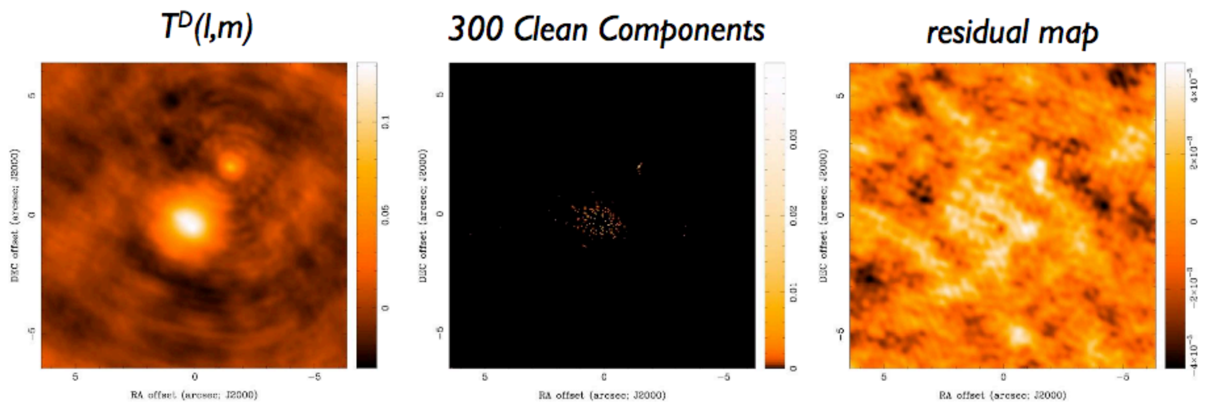


Image credit: David Wilner (CfA), 14th Synthesis Imaging workshop lectures

clean Example

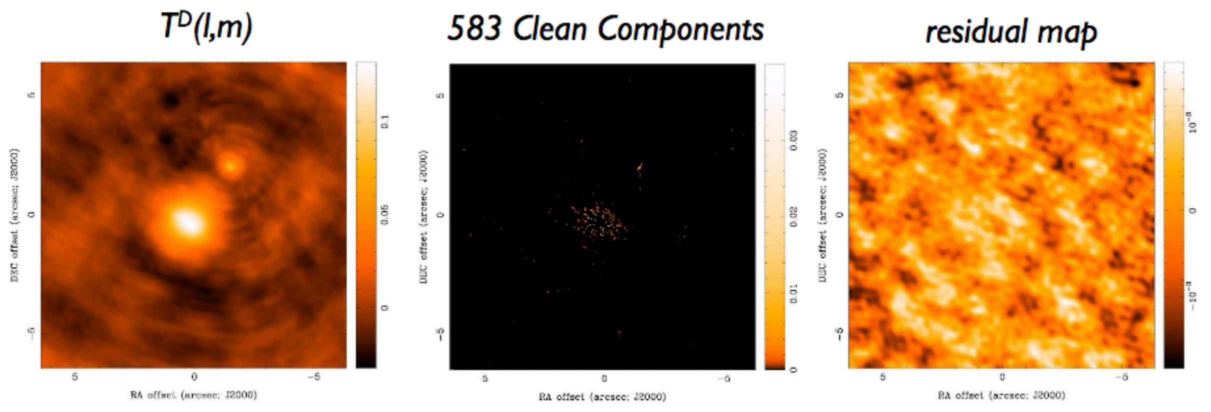
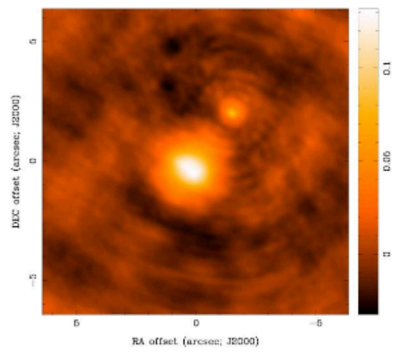


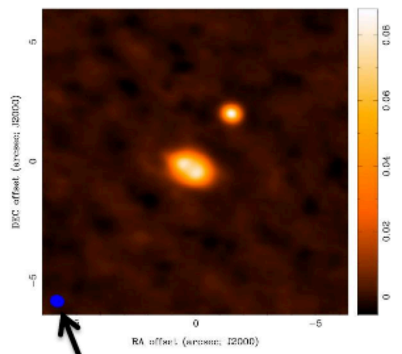
Image credit: David Wilner (CfA), 14th Synthesis Imaging workshop lectures

clean Example

$T^D(l,m)$



restored image



ellipse = clean beam fwhm

final image depends on

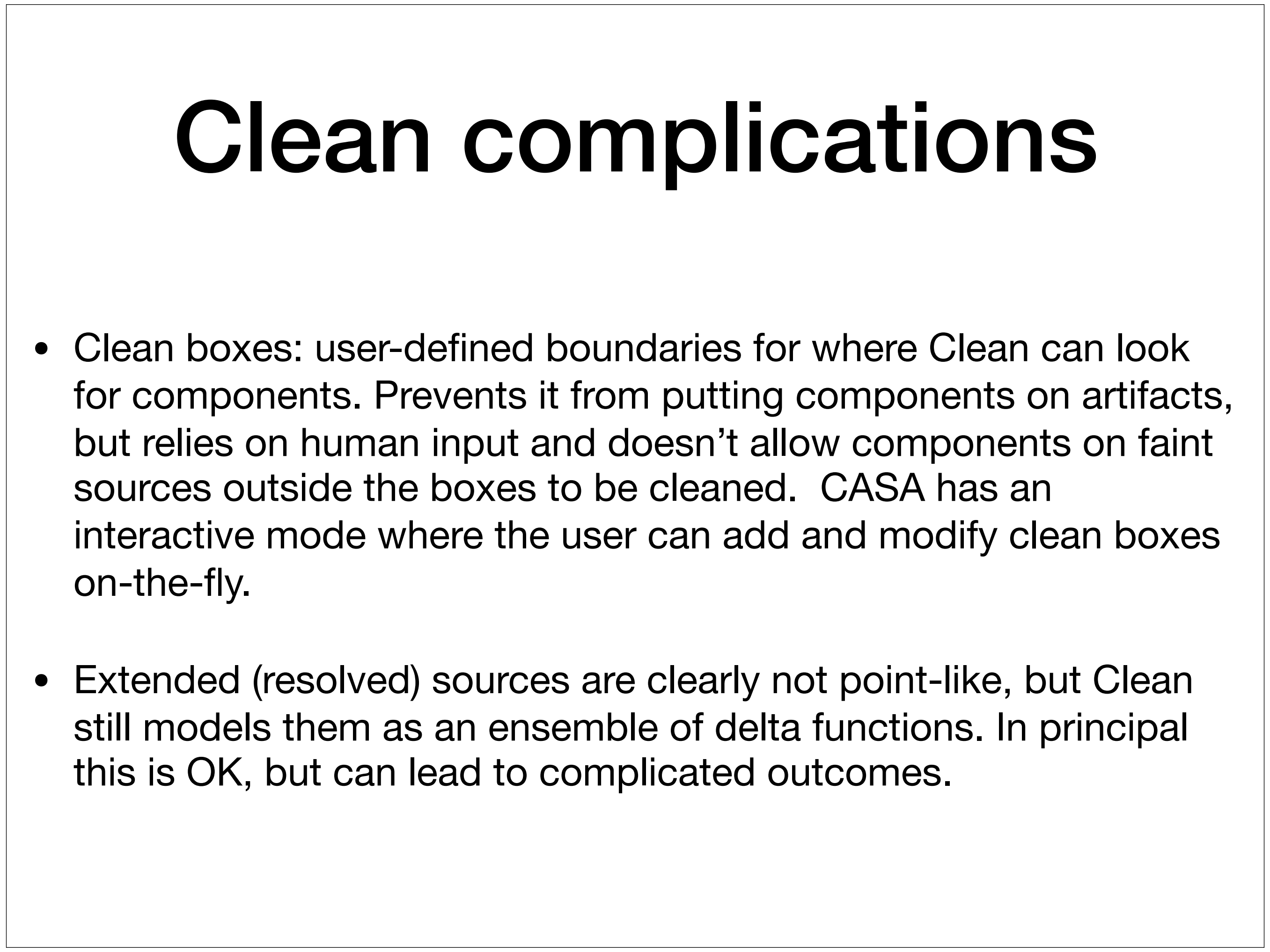
imaging parameters (pixel size, visibility weighting scheme, gridding)
and deconvolution (algorithm, iterations, masks, stopping criteria)

Image credit: David Wilner (CfA), 14th Synthesis Imaging workshop lectures

Clean complications

- Setting good Clean parameters can be hard. Number of iterations depends on the size of the image and how much signal is present. Use as a way to limit maximum possible run time.
- Clean threshold (in intensity) depends on the image: is it noise-dominated? Clean to some multiple of the noise (opinions vary how low to go). Artifact dominated (‘dynamic noise-limited’)? Some fraction of peak intensity.
- Under/over-cleaning: Under-cleaning (having the threshold too high, or not enough iterations) leaves sidelobe structure in the image, which can impact analysis. Over-cleaning tends to put a lot of clean components in places they don’t belong, like image artifacts, and waste computation. In the worst cases, over cleaning can increase the noise level (‘Clean is diverging’).

The number of iterations is mostly useful as a way of preventing Clean from going on forever. Useful to put an upper limit on run time, but not useful for astronomical purposes.
Most modern imagers have additional stopping criteria, like when the RMS starts to increase again, or when it picks out the first negative clean component (most negative pixel is greater magnitude than most positive pixel), although this doesn’t work in polarization (Stokes Q,U,V can be negative).
For clean thresholds, some people say stop at 5 sigma or 3 sigma, because you don’t really care about flux below that level; other people say clean right down into the noise, because adding noise clean components doesn’t mess things up much. Truth is probably somewhere in between.



Clean complications

- Clean boxes: user-defined boundaries for where Clean can look for components. Prevents it from putting components on artifacts, but relies on human input and doesn't allow components on faint sources outside the boxes to be cleaned. CASA has an interactive mode where the user can add and modify clean boxes on-the-fly.
- Extended (resolved) sources are clearly not point-like, but Clean still models them as an ensemble of delta functions. In principal this is OK, but can lead to complicated outcomes.

Not just extended sources, but sources that are not centred on a pixel can cause problems. Clean will model it using clean components in two adjacent pixels, but this is not the same as a source between the pixels, so this introduces errors in the residual/restored image. At some level, dynamic range is limited by these kinds of unavoidable errors.

Clean variations

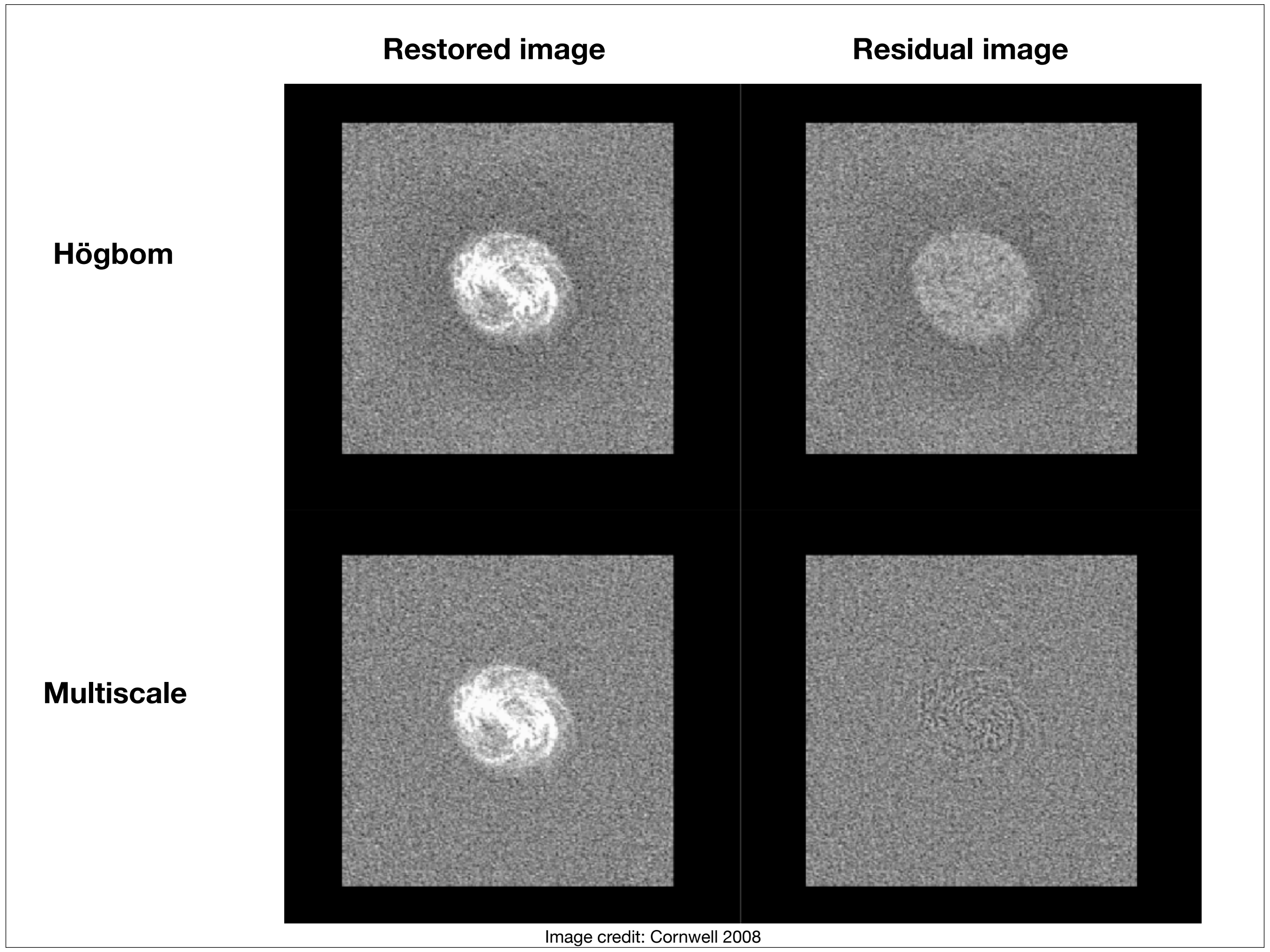
- **Clark clean** (Clark 1980): More efficient variation on Högbom Clean. Consider: subtraction steps on the full image are inefficient, as is the final convolution of the clean components with the Gaussian clean beam. Break clean iterations into levels:
    - Minor cycle (iteration): as before, but only subtract synthesized beam in a small area to save computation.
    - Major cycle (repeated after some number of iterations): Fourier transform the clean components into a visibility model, subtract from visibilities, re-image with visibilities residuals to make new residual image.
    - At the end: multiply final visibility model with Gaussian (equivalent to convolving with clean beam in image domain), add residual visibilities, and re-image to create final image.

Clean variations

- **Multiscale-clean** (Cornwell 2008): Extended sources are poorly modelled by delta-functions. Using a model that supports a width parameter, search the parameter space of (x,y,width) for the best clean component and use that.
- User must define a list of scales to be used. Each scale adds computational expense, so there's a balance of getting the scales relevant to the image while still minimizing overhead.

Standard model is some kind of truncated paraboloid, I think?

Having these extended models helps increase sensitivity to faint emission, making it easier for faint extended things to get cleaned.



The multi-scale clean successfully captures the extended diffuse flux that classic Clean misses.

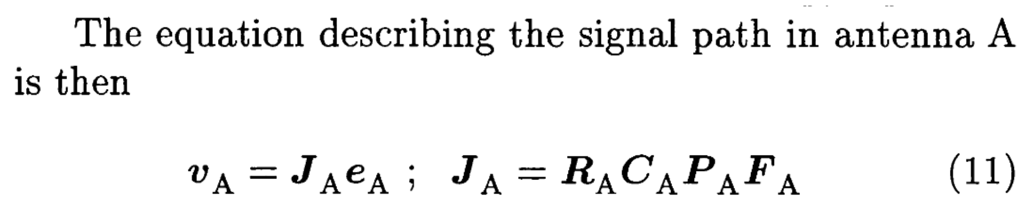
Clean variations

- Multi-frequency synthesis has it’s own Clean algorithm, although I don’t know how it works.
- Multi-scale multi-frequency synthesis (MS-MFS): doing all of this at the same time. Captures frequency behaviour and extended flux all in one algorithm.

MS-MFS is basically the gold standard for imaging at the moment, because it does ‘everything’.

Calibration

Hamaker et al. 1996:



The equation describing the signal path in antenna A is then

$$\boldsymbol{v}_{\mathrm{A}} = \boldsymbol{J}_{\mathrm{A}} \boldsymbol{e}_{\mathrm{A}} \; ; \quad \boldsymbol{J}_{\mathrm{A}} = \boldsymbol{R}_{\mathrm{A}} \boldsymbol{C}_{\mathrm{A}} \boldsymbol{P}_{\mathrm{A}} \boldsymbol{F}_{\mathrm{A}} \qquad (11)$$

- e_A is the sky, v_A is the signal that gets fed into the correlator and $\boldsymbol{J}_A$ is the Jones matrix of everything between the two. The 'A' is for antenna A, meaning this is antenna-dependent.

- $\boldsymbol{J}_A$ is a 2x2 complex matrix, with 8 parameters. Calibration is the process of finding these parameters (as functions of time, frequency, direction, antenna, etc).

- Some factors, like the beam ($\boldsymbol{C}_A$) and Faraday rotation ($\boldsymbol{F}_A$) are usually treated separately, and some like parallactic angle ($\boldsymbol{P}_A$) can be computed and accounted for. Everything else is usually lumped together into one matrix of unknowns

One thing that isn't technically accounted for in the Hamaker model, but is very important, is phase errors introduced by the ionosphere. These are very important at low frequencies, but are one of the effects lumped into the calibration matrix.

Correlation in Jones formalism

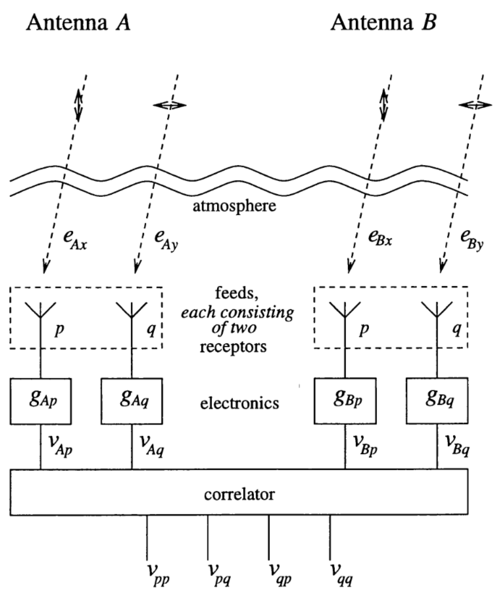


Fig. 1. Interferometer block diagram

Image credit: Hamaker+96

- Haslam defines a *visibility vector*:

$$\boldsymbol{v} = < \begin{pmatrix} v_{\mathrm{A}x}v^*_{\mathrm{B}x} \\ v_{\mathrm{A}x}v^*_{\mathrm{B}y} \\ v_{\mathrm{A}y}v^*_{\mathrm{B}x} \\ v_{\mathrm{A}y}v^*_{\mathrm{B}y} \end{pmatrix} >$$

- In Jones formalism the measured visibilities can be expressed as an outer product of the Jones vectors:

$$v_\mathrm{A} = \boldsymbol{J}_A e_{\mathrm{A},in} \; ; \; v_\mathrm{B} = \boldsymbol{J}_B e_{\mathrm{B},in}$$

$$\begin{aligned} v_\mathrm{meas} &= v_\mathrm{A} \otimes v_\mathrm{B} \\ &= (\boldsymbol{J}_\mathrm{A} \otimes \boldsymbol{J}_\mathrm{B}) < e_\mathrm{A} \otimes e_\mathrm{B} > \\ &= \boldsymbol{J}\, v_\mathrm{sky} \end{aligned}$$

where ***J*** is the outer product of the two antenna $\boldsymbol{J}_{\mathrm{A/B}}$ matrices and v_{sky} is the true sky visibilities

Haslam’s figure uses p and q for the antenna polarizations, which could be either x/y linear or R and L circular polarizations.
This isn’t super important but provides the missing link between antenna-based Jones vectors/matrices and the correlations for an interferometer.

Calibration

- Interferometer calibration essentially reduces down to solving the Jones matrices for the individual antennas (i.e., solving for antenna-dependent effects)

- In principle, there can sometimes be baseline-dependent effects, but I don't know what the cause would be. This doesn't seem to come up during normal calibration.

- Calibration requires observations for which the sky is known, so the difference between sky and measurement can be used to determine the unknown parameters.

$$v_{\mathrm{A}} = \boldsymbol{J}_{\mathrm{A}} e_{\mathrm{A}}$$

$$\begin{bmatrix} v_{\mathrm{Ax}} \\ v_{\mathrm{Ay}} \end{bmatrix} = \begin{bmatrix} G_{\mathrm{Ax}}\, e^{i\theta_{\mathrm{Ax}}} & d_{y\to x}\, e^{i\theta_{y\to x}} \\ d_{x\to y}\, e^{i\theta_{x\to y}} & G_{\mathrm{Ay}}\, e^{i\theta_{\mathrm{Ay}}} \end{bmatrix} \begin{bmatrix} e_{\mathrm{Ax}} \\ e_{\mathrm{Ay}} \end{bmatrix}$$

$$v_{\mathrm{A}} = \boldsymbol{J}_{\mathrm{A}} e_{\mathrm{A}}$$

$$\begin{bmatrix} v_{\mathrm{Ax}} \\ v_{\mathrm{Ay}} \end{bmatrix} = \begin{bmatrix} G_{\mathrm{Ax}}\, e^{i\theta_{\mathrm{Ax}}} & d_{y\to x}\, e^{i\theta_{y\to x}} \\ d_{x\to y}\, e^{i\theta_{x\to y}} & G_{\mathrm{Ay}}\, e^{i\theta_{\mathrm{Ay}}} \end{bmatrix} \begin{bmatrix} e_{\mathrm{Ax}} \\ e_{\mathrm{Ay}} \end{bmatrix}$$

- **Gains** are the amplitude calibration parameters, defining the conversion from measured output units to sky intensity (in Jy/ beam or equivalent).

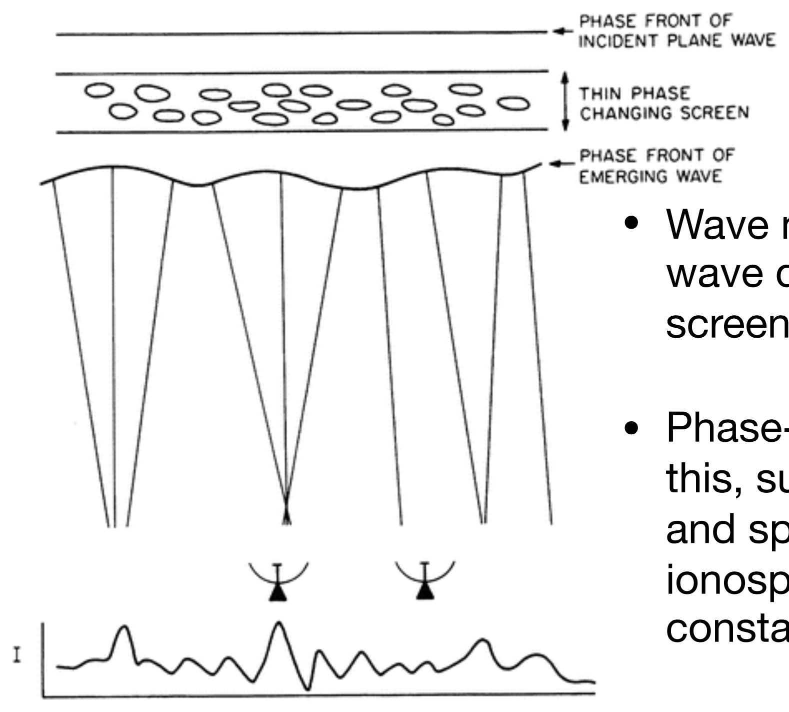
- Are a function of:
  - Frequency (bandpass filter, amplifier response)
  - Time (changes in the electronics)
  - Antenna (different signal chains)

$$v_{\mathrm{A}} = \boldsymbol{J}_{\mathrm{A}} e_{\mathrm{A}}$$

$$\begin{bmatrix} v_{\mathrm{Ax}} \\ v_{\mathrm{Ay}} \end{bmatrix} = \begin{bmatrix} G_{\mathrm{Ax}}\, e^{i\theta_{\mathrm{Ax}}} & d_{y\to x}\, e^{i\theta_{y\to x}} \\ d_{x\to y}\, e^{i\theta_{x\to y}} & G_{\mathrm{Ay}}\, e^{i\theta_{\mathrm{Ay}}} \end{bmatrix} \begin{bmatrix} e_{\mathrm{Ax}} \\ e_{\mathrm{Ay}} \end{bmatrix}$$

- **Phases** are the changes in signal phase caused by the system, which would produce the wrong visibility phase if not corrected.

- Are a function of:
  - Frequency and time (changes in the electronics)
  - Direction (phase effects from the ionosphere/atmosphere)
  - Antenna (different signal chains and different position relative ionosphere/atmosphere)


PHASE FRONT OF INCIDENT PLANE WAVE
THIN PHASE CHANGING SCREEN
PHASE FRONT OF EMERGING WAVE
I

- Wave may not arrive as plane wave due to ionospheric ‘phase-screen’ effects
- Phase-calibration can correct for this, subject to some time-scale and spatial scale over which the ionosphere can be treated as constant
- Effect is stronger at very low frequencies (ionosphere) and very high (atmosphere)

$$v_{\mathrm{A}} = \boldsymbol{J}_{\mathrm{A}} e_{\mathrm{A}}$$

$$\begin{bmatrix} v_{\mathrm{Ax}} \\ v_{\mathrm{Ay}} \end{bmatrix} = \begin{bmatrix} G_{\mathrm{Ax}}\, e^{i\theta_{\mathrm{Ax}}} & d_{y\to x}\, e^{i\theta_{y\to x}} \\ d_{x\to y}\, e^{i\theta_{x\to y}} & G_{\mathrm{Ay}}\, e^{i\theta_{\mathrm{Ay}}} \end{bmatrix} \begin{bmatrix} e_{\mathrm{Ax}} \\ e_{\mathrm{Ay}} \end{bmatrix}$$

- **Polarization leakage** (‘d-terms’) is the part of the signals that ‘leak’ from one polarization into the other. Due to imperfections in the antennas, cross-coupling between signal chains, etc.
- Are a function of:
  - Antenna (each antenna’s physical construction is a little different)
  - Time and frequency?

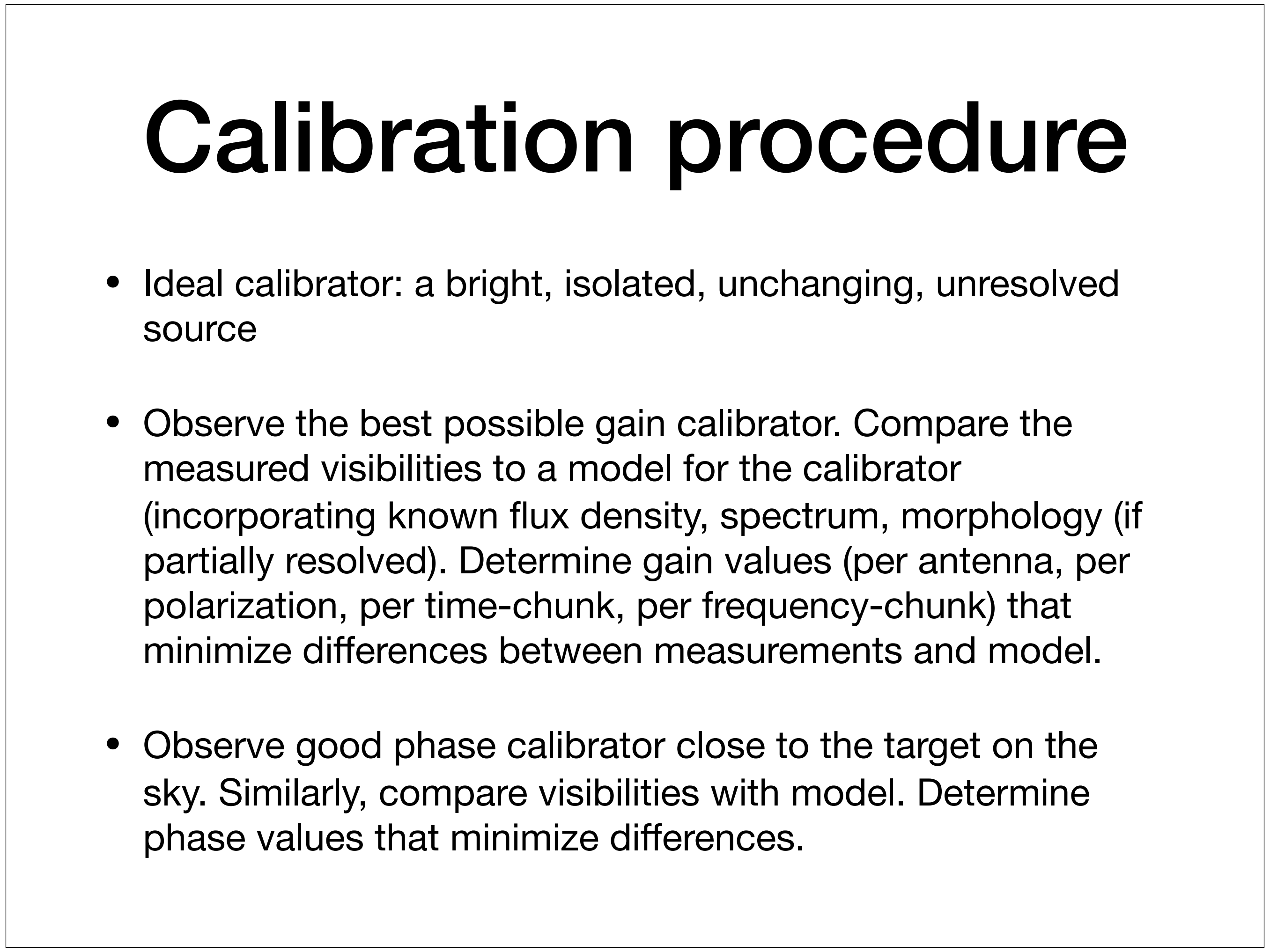
Polarization leakage calibration is typically not done if the data is not intended for polarization analysis: effect on Stokes I (total intensity) is usually minimally effected (<1%). This can become a big problem for polarization analysis, as it can cause the other Stokes parameters to be affected by some fraction of Stokes I (up to maybe 10%), which can be much stronger than the often-weak polarized signals.

Free parameters

- For normal continuum calibration (excluding d-terms):
  - 2 N_{ant} gains
  - 2 N_{ant} - 2 phases
- Basic polarization calibration: 1 more phase (linking X and Y)
- Polarization leakage calibration: 4 N_{ant} d-term parameters

- Contrast this with the number of measurements:
  N_{ant} (N_{ant} - 1)/2 baselines, x2 polarizations
  This scales much faster than the free parameters, so larger arrays are better constrained for calibration.

The reason for the -2 phases is that all that really matters for the visibilities is the phase differences between antennas, so there's no reference for absolute phase (for both X and Y). Normally the user specifies a reference antenna, and the phase for that antenna is adjusted to zero.



Calibration procedure

- Ideal calibrator: a bright, isolated, unchanging, unresolved source
- Observe the best possible gain calibrator. Compare the measured visibilities to a model for the calibrator (incorporating known flux density, spectrum, morphology (if partially resolved). Determine gain values (per antenna, per polarization, per time-chunk, per frequency-chunk) that minimize differences between measurements and model.
- Observe good phase calibrator close to the target on the sky. Similarly, compare visibilities with model. Determine phase values that minimize differences.

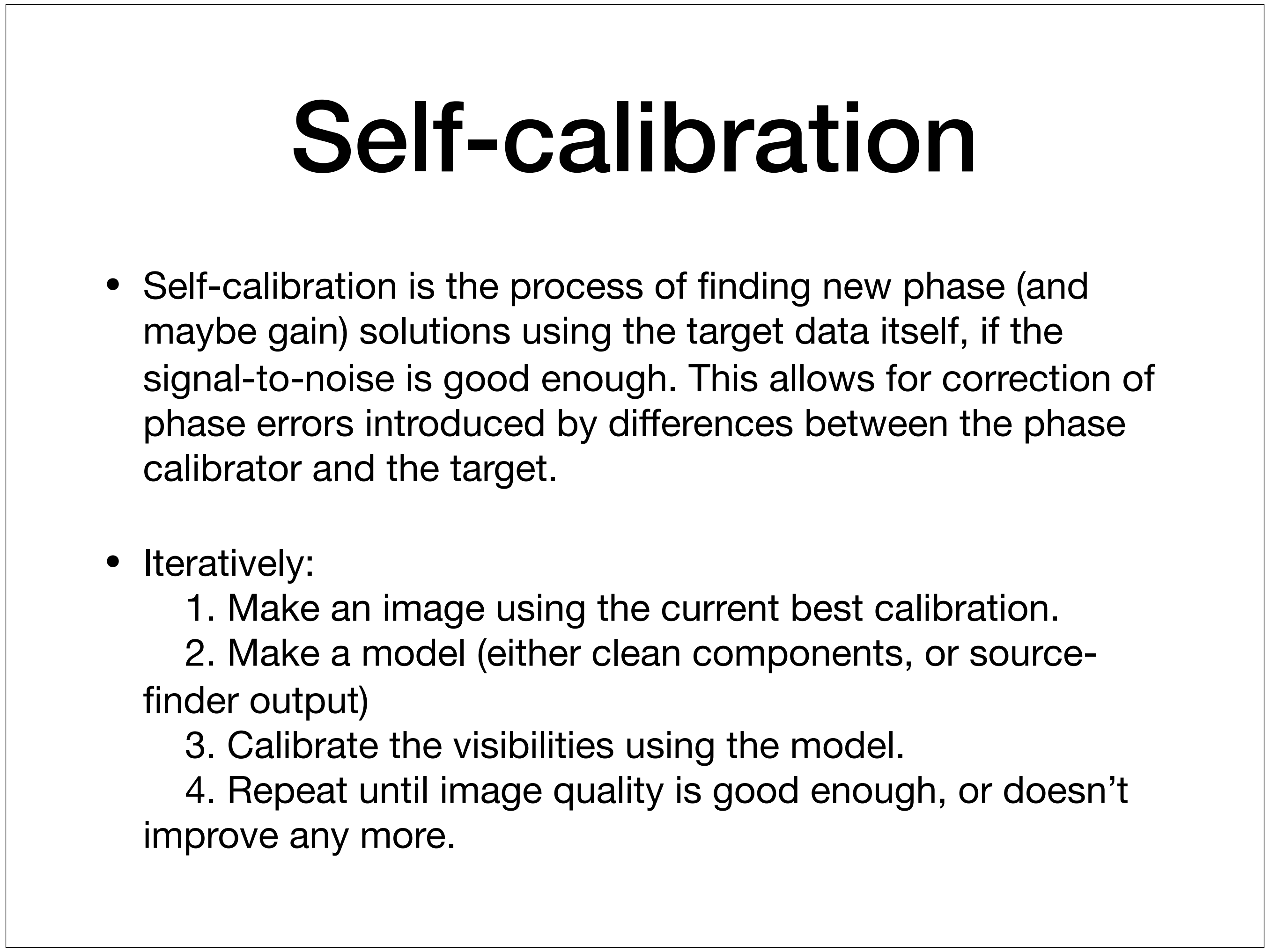
(Calibration of total intensity only. Polarization discussed separately.)
Bright: to give the best signal to noise in the calibration solutions
Isolated: to avoid contributions from other nearby sources which may not be modelled.
Unchanging: no time evolution, so we can know what the source is like at the time of observation.
Unresolved: a delta-function like source has equal brightness across the whole *u,v* plane, so it can be modelled easily and maintains high S:N even for long baselines.

The size of a time and frequency chunk are usually selected by the user. They should be short enough to capture time-variations in the gains, but long enough to maximize signal-to-noise in computing the gain solutions.

Note that sometimes the gain calibration can be split into two parts: bandpass calibration, which deals with the frequency dependence of the gains, and 'gain' calibration, which deals with time dependence

Calibration procedure

- Observe target. Apply gain and phase calibration solutions found from calibrators.
- Repeat phase and gain calibration as needed (for time-variability). Perform new phase calibration if switching to targets in different parts of the sky.
- Self-calibrate as desired.



Self-calibration

- Self-calibration is the process of finding new phase (and maybe gain) solutions using the target data itself, if the signal-to-noise is good enough. This allows for correction of phase errors introduced by differences between the phase calibrator and the target.

- Iteratively:
   1. Make an image using the current best calibration.
   2. Make a model (either clean components, or source-finder output)
   3. Calibrate the visibilities using the model.
   4. Repeat until image quality is good enough, or doesn't improve any more.

One thing that is often emphasized is that this is not 'cheating', in terms of manipulating the science quality of the data: the number of free parameters (the Jones matrix phases) is much less than the amount of information (the number of baselines), so this is a reasonable minimization problem.

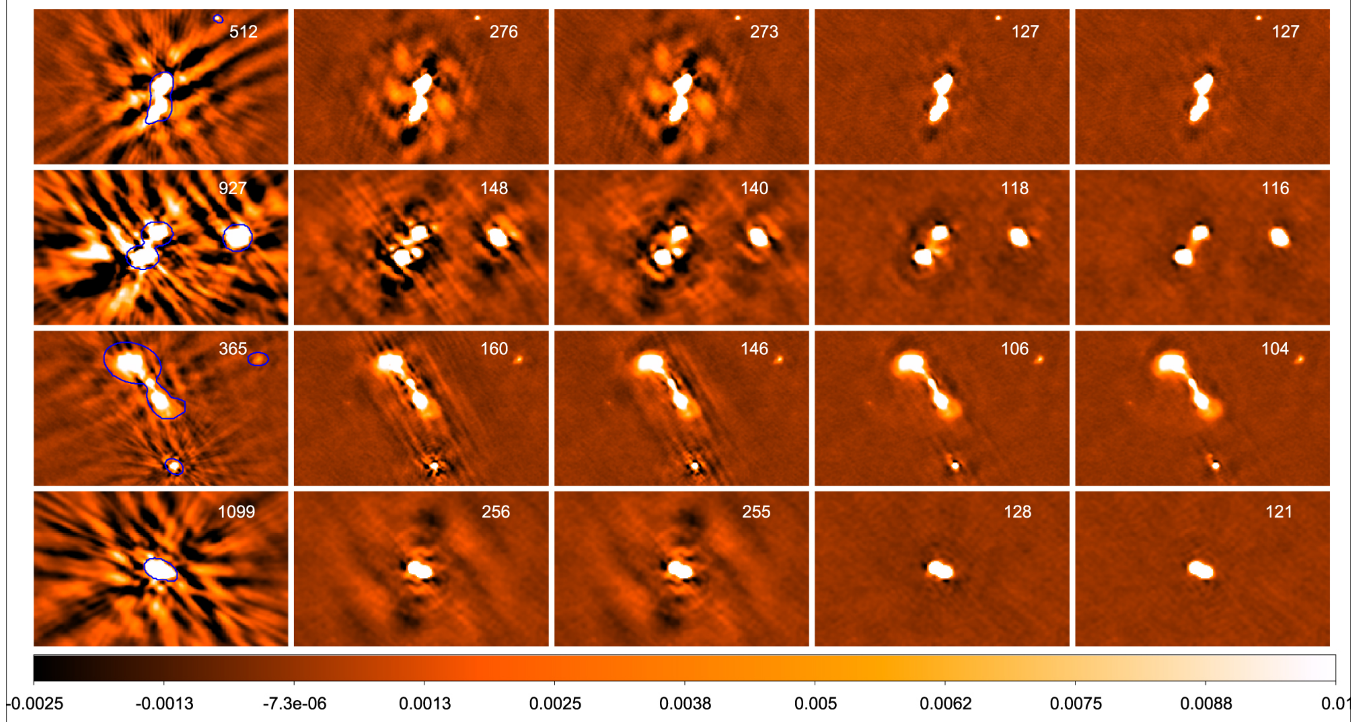
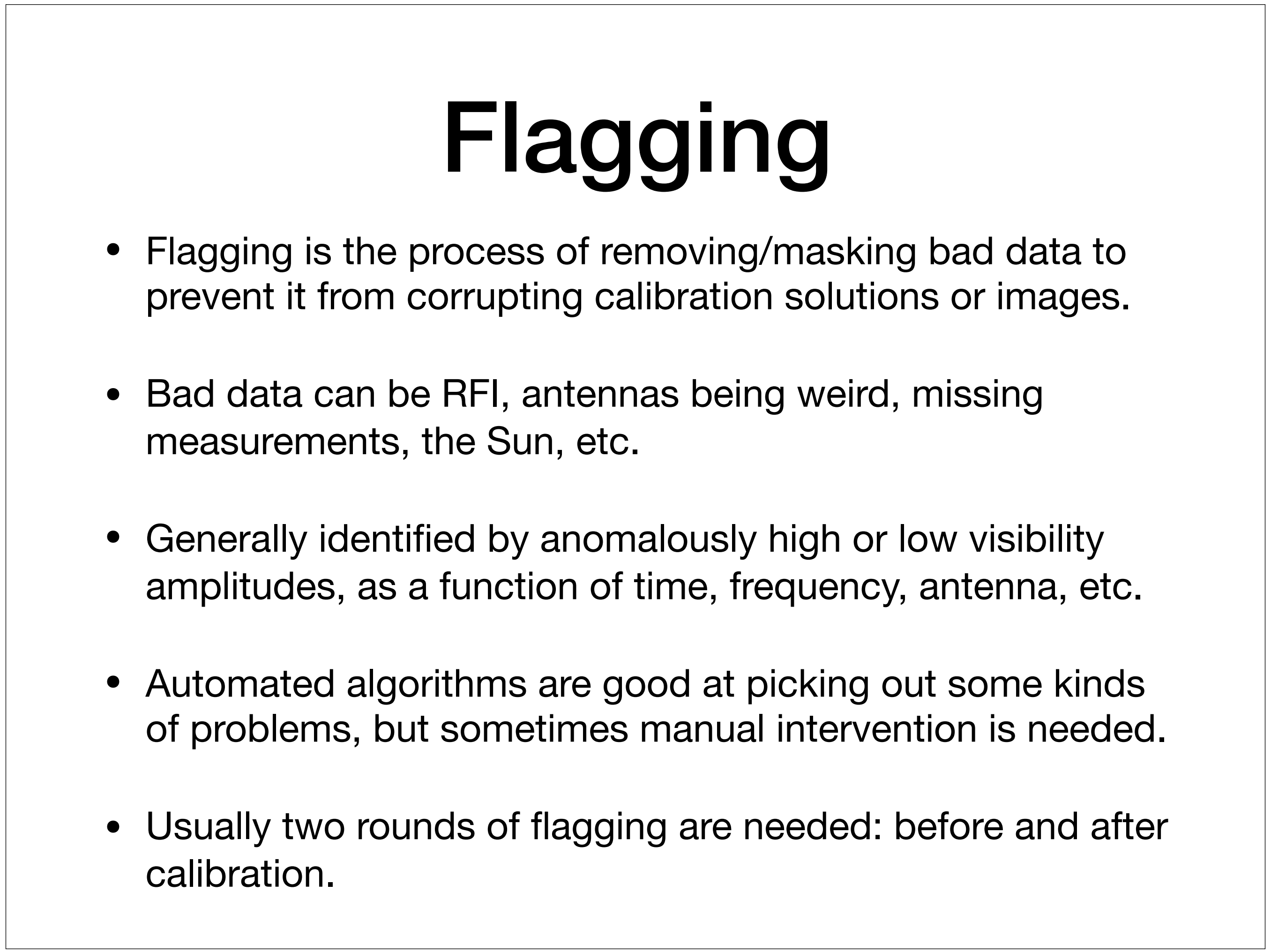


Image credit: R.J. van Weeren et al 2016

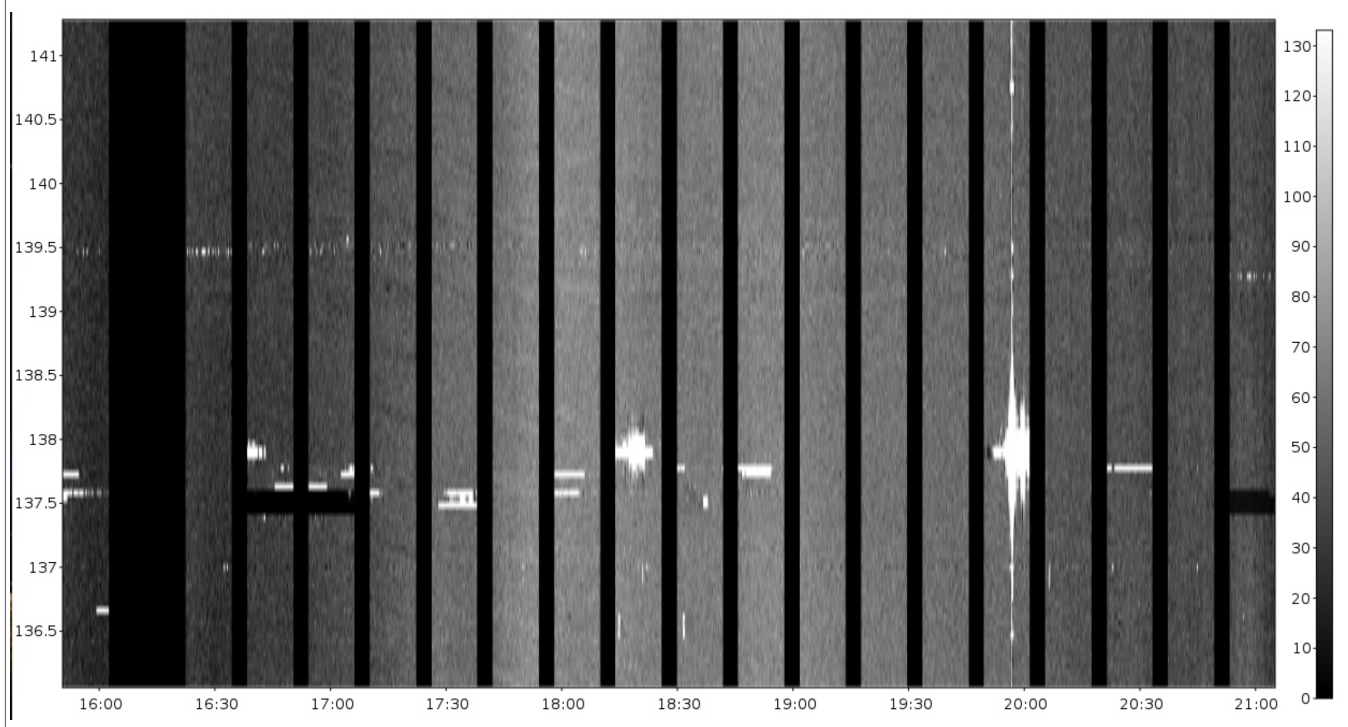
This is an example of self-calibration of many sources with LOFAR data. Each row is a different radio source, each column is an iteration of self-calibration. The numbers are the noise rms in the image, in μJy/beam.



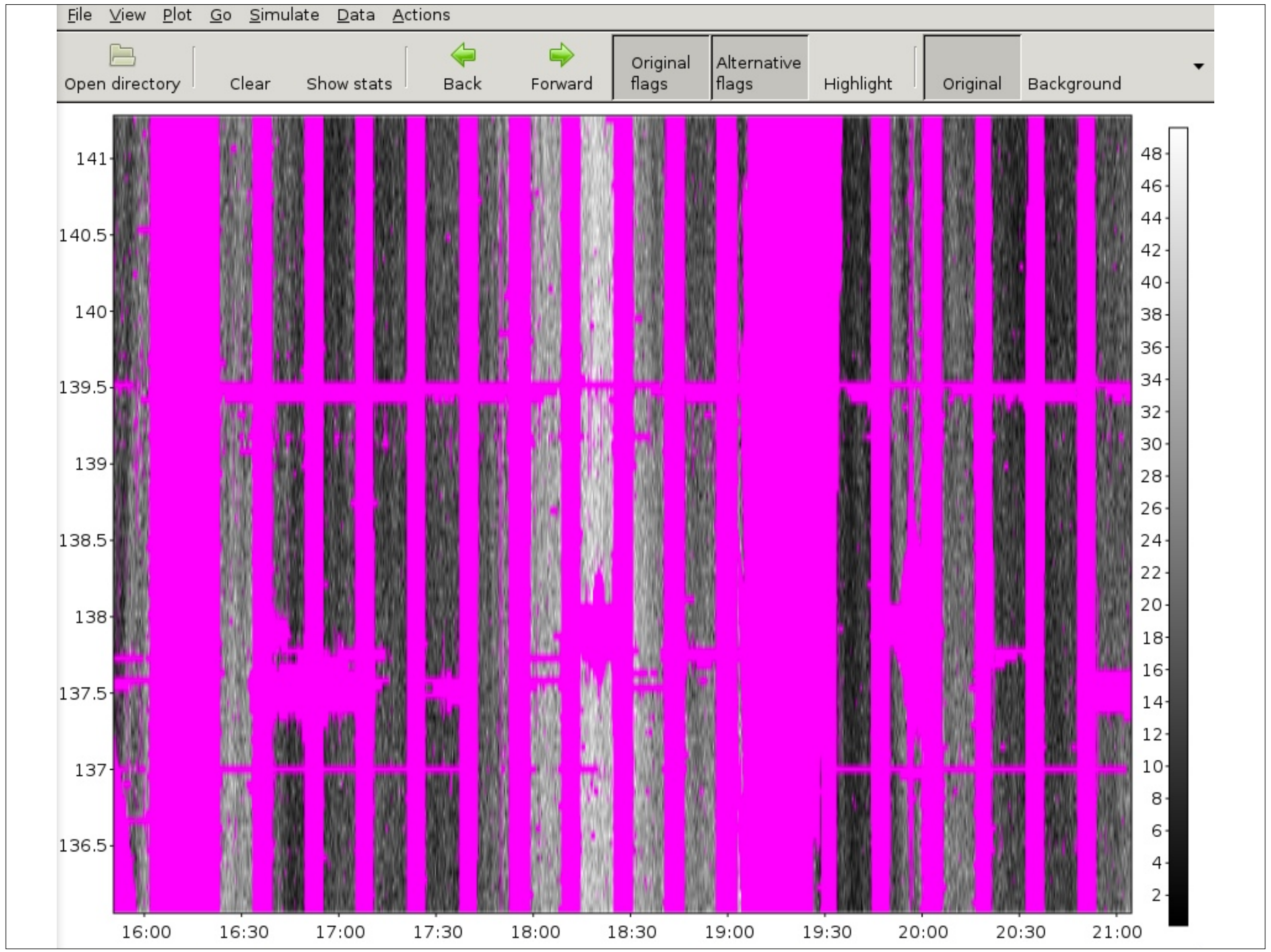
Flagging

- Flagging is the process of removing/masking bad data to prevent it from corrupting calibration solutions or images.
- Bad data can be RFI, antennas being weird, missing measurements, the Sun, etc.
- Generally identified by anomalously high or low visibility amplitudes, as a function of time, frequency, antenna, etc.
- Automated algorithms are good at picking out some kinds of problems, but sometimes manual intervention is needed.
- Usually two rounds of flagging are needed: before and after calibration.

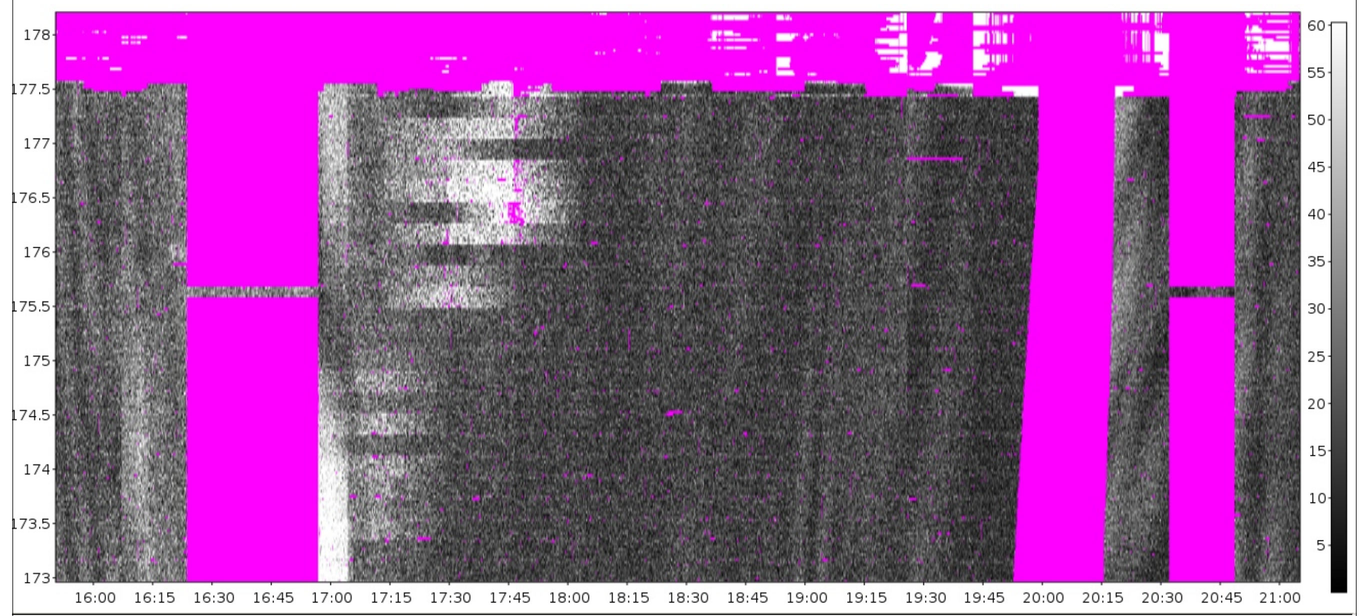
On the two rounds of flagging: sometimes bad values aren't apparent until after calibration. But it's a bit tricky: are the bad values caused by bad data, or bad calibration solutions? Human judgement is needed sometimes.
If post-calibration flagging removes a lot of data, the calibration should be redone, because that bad data was negatively affecting the first calibration.



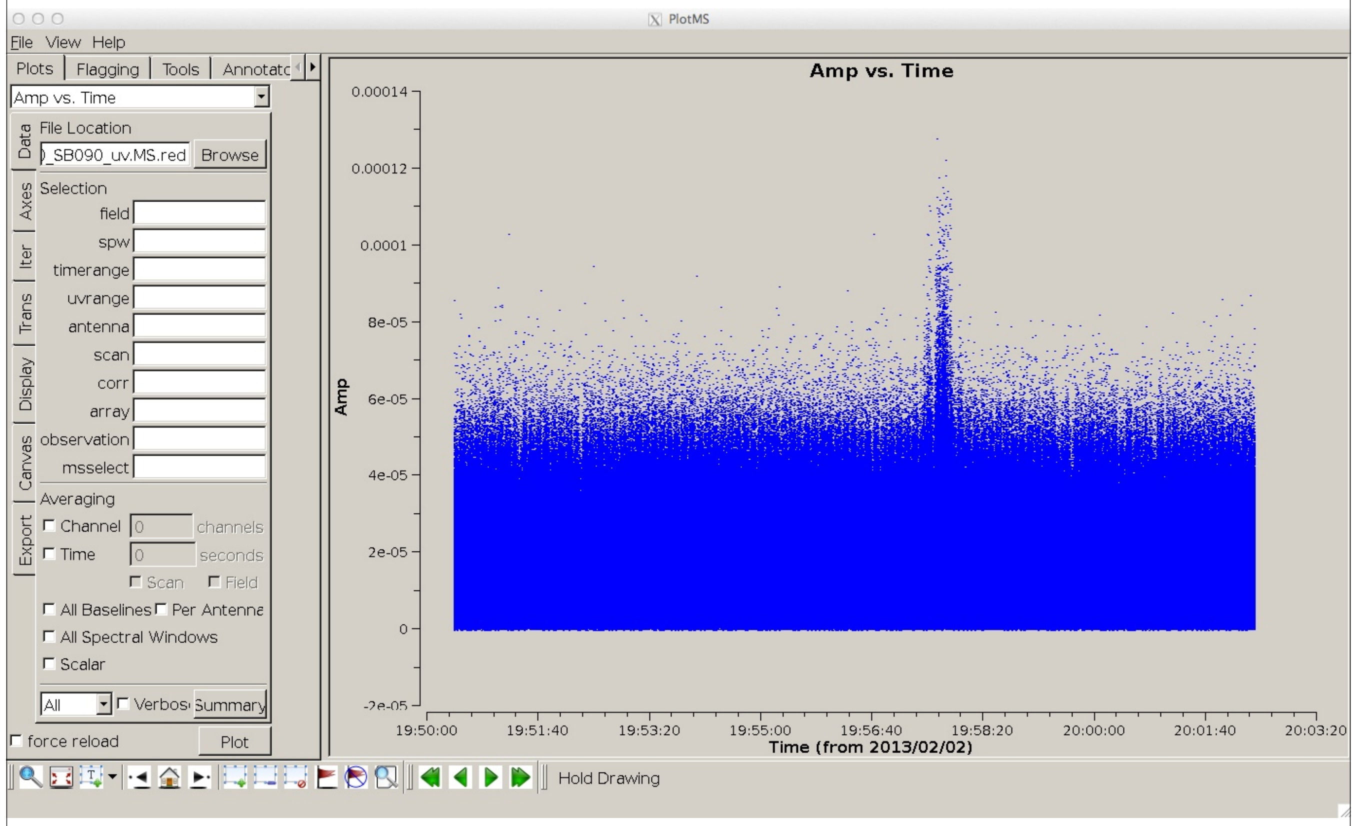
This is a ‘waterfall plot’, showing the signal over frequency and time. Many different kinds of RFI signals can be seen in there.



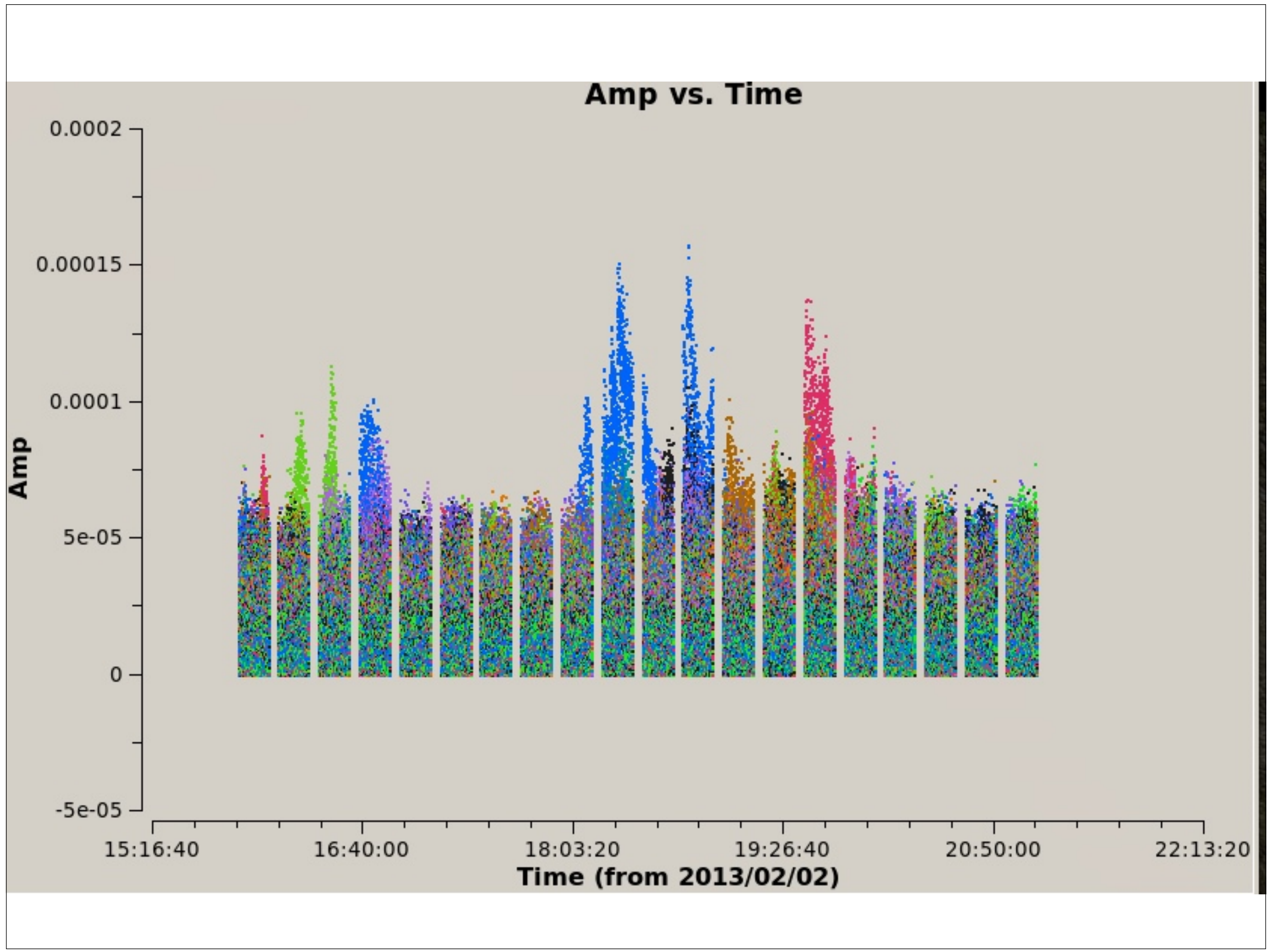
The same data, plotting what an automated flagging routine has found and removed.



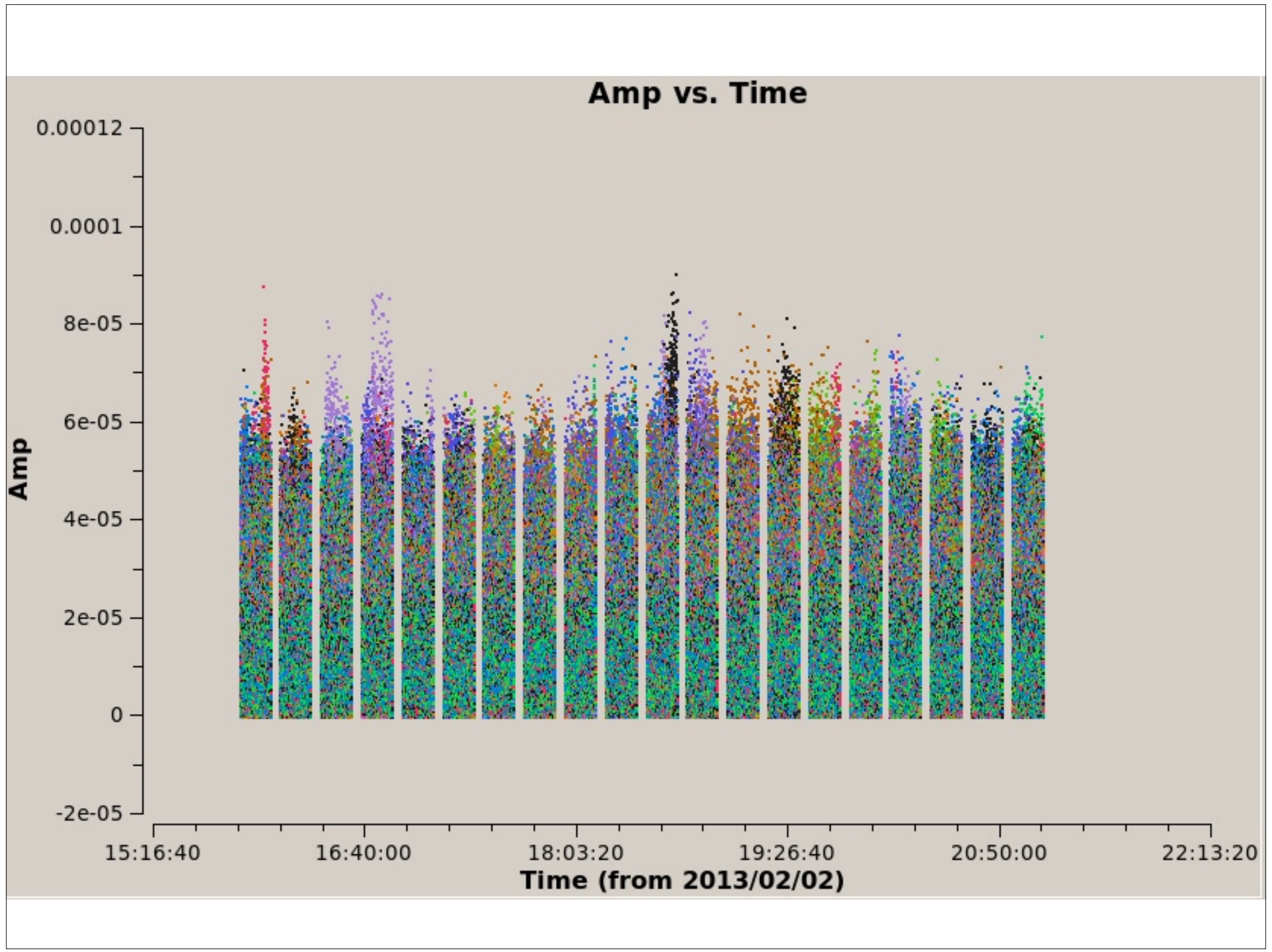
It doesn't catch everything, however…



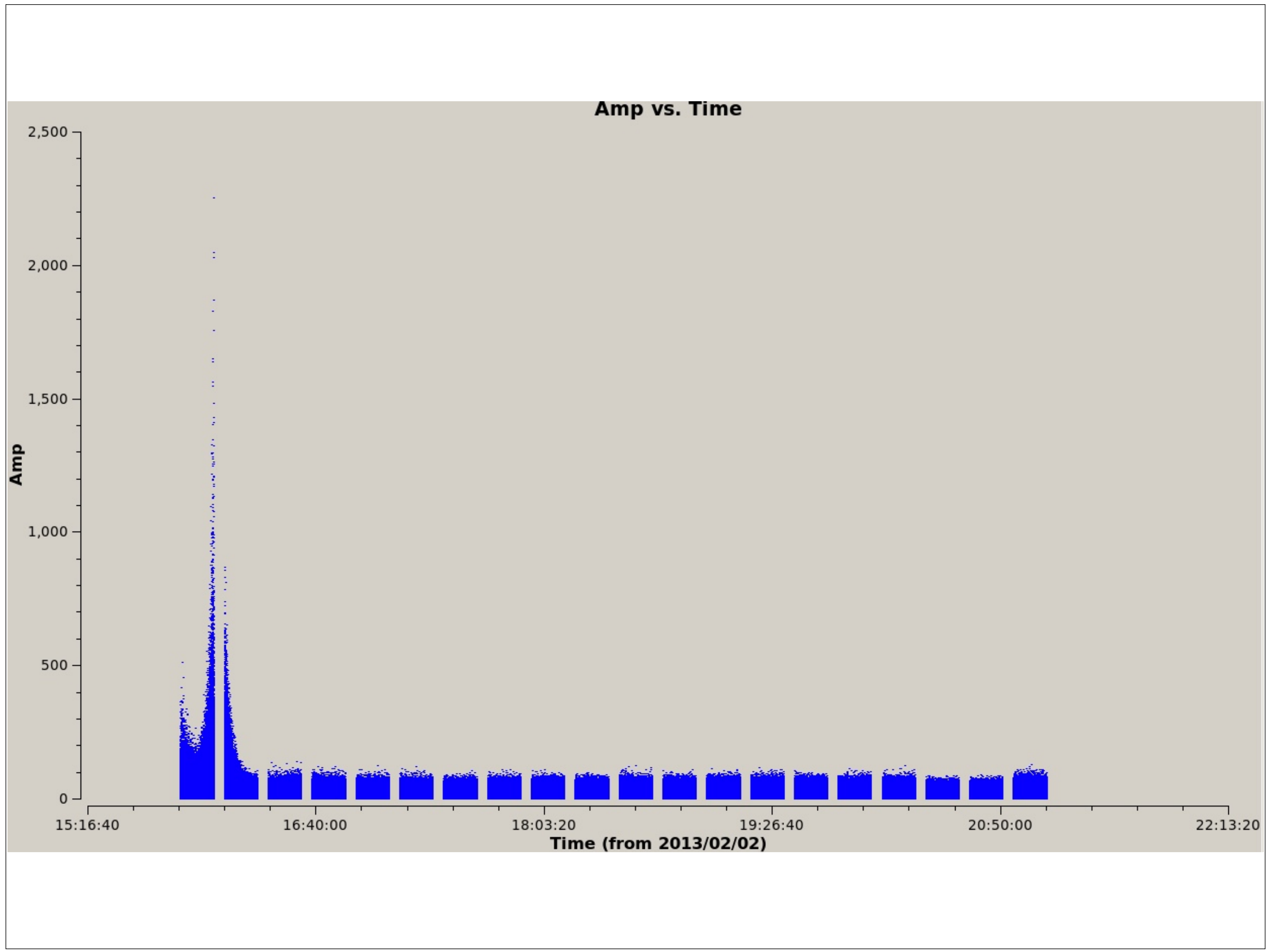
This is an example of bad data in a LOFAR target observation, AFTER automated flagging. The automated flagging didn't quite capture that short period of anomalously high amplitudes (which turned out to be antenna 27 being weird for a minute).



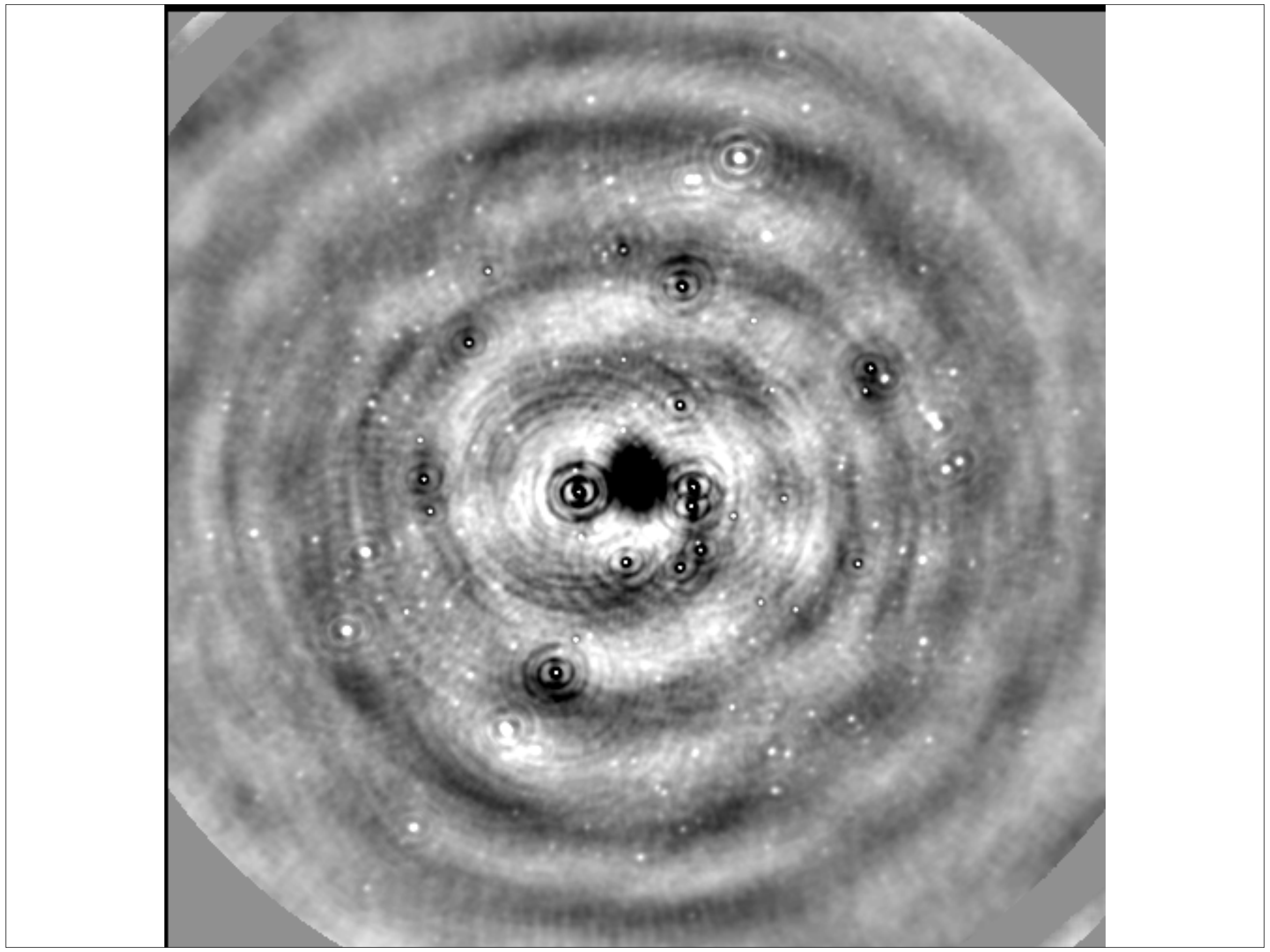
Another example, in the actual target data. Colored by antenna1, I think?



After flagging out bad data. Note that there's still weird behaviour in some places. Flagging is always about thresholds: how much does it affect the final result? Is it worth the time spent identifying it? It's surprising how little effect stuff like this can have on the final result: usually bad data has random phases, so it scatters noise throughout the image plane.



An example of bad data that didn’t really show up until after calibration.



As a distraction, here’s an image from my LOFAR data where the imager completely messed up. Something was broken with multi scale clean in the first generation of LOFAR imaging software. I have a lot of plots/images of entertainingly messed up data stashed away in my PhD notes.

Next week: CASA tutorial

- Next week will be CASA introduction/tutorial. I'll give an introduction to CASA and it's quirks, and how data reduction in CASA typically goes.
- Then we'll work through one of the VLA tutorials (pick one ahead of time; I'll post which one I'm testing/demoing). Can use screen share to share results and diagnose problems as a group.
- Install CASA ahead of time! And download a tutorial data set.
- Last lecture: polarimetry, and applications of radio astronomy.
  OR: send me questions about concepts that you don't get.

Send me suggestions of things you want to understand better. If I know about it I can make a few slides talking about it in detail.