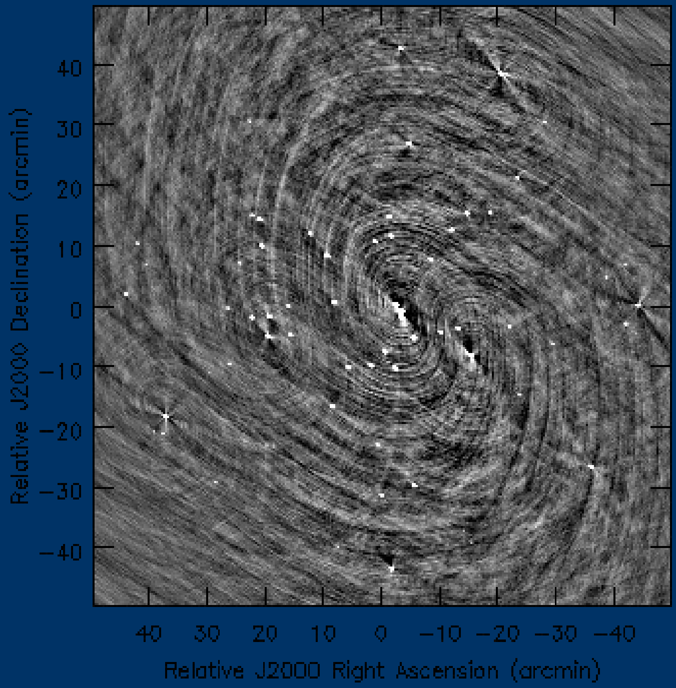
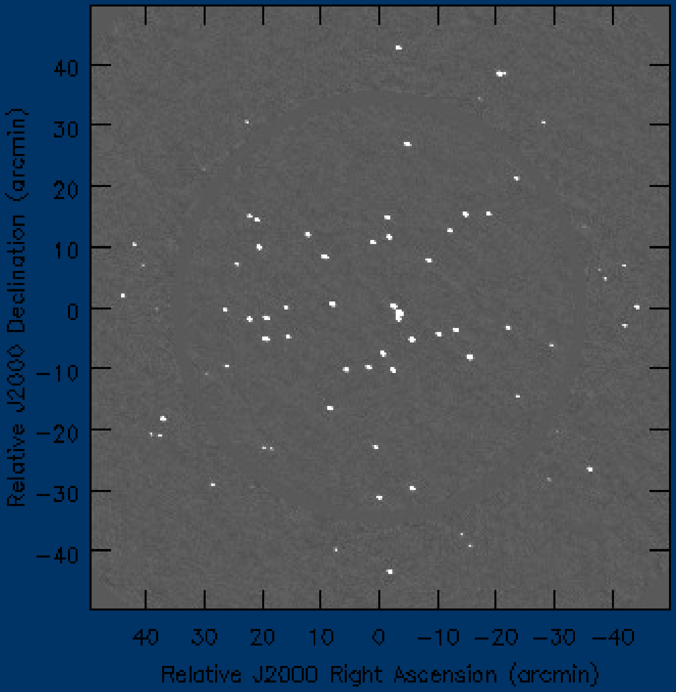
Direction-dependent effects



RMS ~15μJy/beam



RMS ~1μJy/beam

S. Bhatnagar
NRAO, Socorro

Direction dependent effects

- Instrumental
  - Primary Beam Effects
    - Time and frequency dependent
    - Polarization response
  - Pointing Errors
  - Non co-planar baselines (w-term)
  - FPA calibration/stability

- Sky
  - Stronger and more complex at low frequencies
    - Deconvolution errors, pixelation errors
  - Spectral index variations across the sky

- Ionospheric/atmospheric

Challenges (not addressed here)

- Computing and I/O loads
  - Going after TB data size, ~10^4 significant pixels
- RFI removal
  - Strong RFI: Flagging algorithms/schemes exist and work well for moderate sized database
- Weak RFI
  - Difficult to detect and remove
  - Effects high dynamic range imaging
  - Some algorithms exist
- Near field problems
  - RFI remains correlated
  - Not the same on all baselines
  - Variable in time and frequency

Measurement Equation

- Generic Measurement Equation

$$V_{ij}^{Obs}(\nu) = J_{ij}(\nu, t) W_{ij} \int J_{ij}^{S}(s, \nu, t)\ I(s, \nu)\ e^{\iota s.b_{ij}}\ ds$$

Data   Corruptions   Sky   Geometry

- Corruptions: $J_{ij} = J_i \otimes J_j^*$ Direction independent corruptions

  $J_{ij}^S = J_i^S \otimes J_j^{S*}$ Direction dependent corruptions

- Sky: Frequency dependence: $I(s,\nu) = I(s,\nu_0)\left(\frac{\nu}{\nu_0}\right)^{\alpha(s,\nu)}$
- Sky: Complex structure
  - Representation in a more appropriate basis
- Geometrical: W-term $e^{\iota s.b_{ij}} = e^{\iota[ul + vm + w(\sqrt{1-l^2-m^2}-1)]}$

- The combined LHS determines “time constant” over which averaging helps

Challenges

- Unknowns
  - J_{ij}, J^S_{ij}: Electronics, Primary Beams, antenna pointing, Ionosphere
    - Heterogeneous arrays (difference PB per baseline)
  - I^M: Extended emission, spectral index variations

- Need efficient algorithms:
  - To solve parameterized ME (Curse of Dimensionality)
  - For *known* direction dependent corrections
  - Better parameterization of the sky (I^M)
    - Including frequency dependence
  - Solver for the *unknown* DD effects (PB, ionosphere)

- Computing
  - Parallel computing & I/O
  - Software development costs

Parameterization in conventional algorithms

- $V_{ij}^{Obs}(\nu) = J_{ij}(\nu, t) W_{ij} \int J_{ij}^{S}(\mathbf{s}, \nu, t) \; I(\mathbf{s}, \nu) \; e^{\iota \mathbf{s}.\mathbf{b}_{ij}} \; d\,\mathbf{s}$

  – $J_{ij}^{S}(\mathbf{s}, \nu, t) \; = \; PB$ *Independent of time and Freq.*

    - *Post deconvolution PB-correction*
    - *Use simple PB models (mostly Gaussian fits)*

  – $I(s) = \sum_k \delta(x_k, y_k)$ *Image representation in pixel basis*

    - *Clean, MEM, and variants: Each pixel is a degree of freedom*

  – $J_{ij}(\nu, t) \; = \; J_{ij}(\nu) J_{ij}(t)$ *Direction independent gains*

    - *Single gain for the full FoV*
    - *Direction independent polarization leakage*

Algorithmic challenges

- Higher sensitivity ==> mode data + correction of more error terms
  - Imaging and calibration gets coupled
  - DD corrections can be as expensive as imaging
- *More sophisticated parametrization required for the next generation telescopes*
  - *DD correction: PB(t, Freq, Pol.), atmosphere/ionosphere*
  - *Sky: Decompose the structure in scale sensitive basis*
  - *Sky: Parametrized for frequency and poln. Dependences*
- *Physically motivated parametrization*
  - *Algorithmic performance-measure: SNR per DoF*

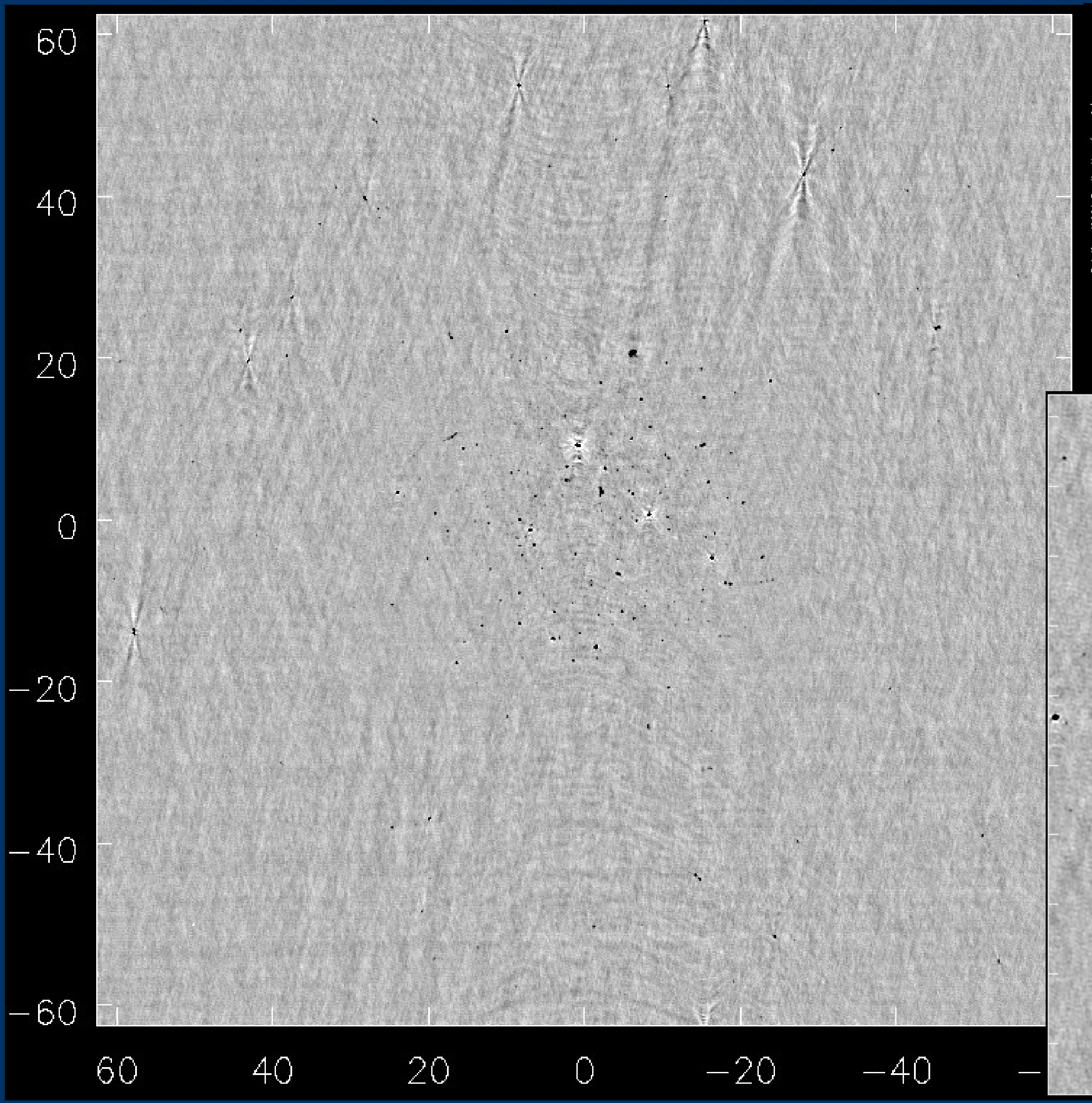
Recent advances

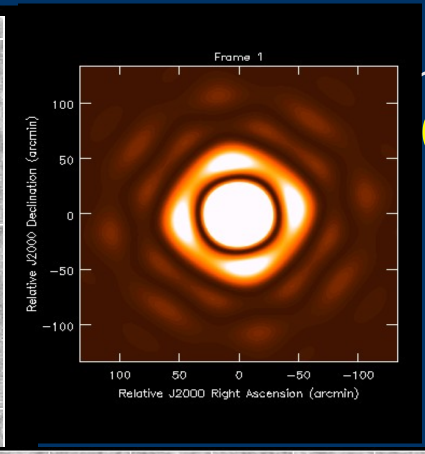
- $J_i^s(t) \neq J_j^s(t)$ (Pointing offsets, PB variations, etc.)
  - Corrections in the visibility plane
    - Scale sensitive deconvolution
      - Asp-Clean (2004), MS-Clean (2003)
    - Pointing SelfCal (2004)
    - Correction for J^s_{ij} during image deconvolution
      - W-Projection (2004)
      - AW-Projection (2005)
      - MS-MFS (2006-07)
  - Direct evaluation of the integral

$$V_{ij}^{Obs}(\nu) = J_{ij}(\nu, t) \int J_{ij}^{S}(\mathbf{s}, \nu, t) \sum_k I(x_k, y_k) \; e^{\iota \mathbf{s}.\mathbf{b}_{ij}} \; d\,\mathbf{s}$$

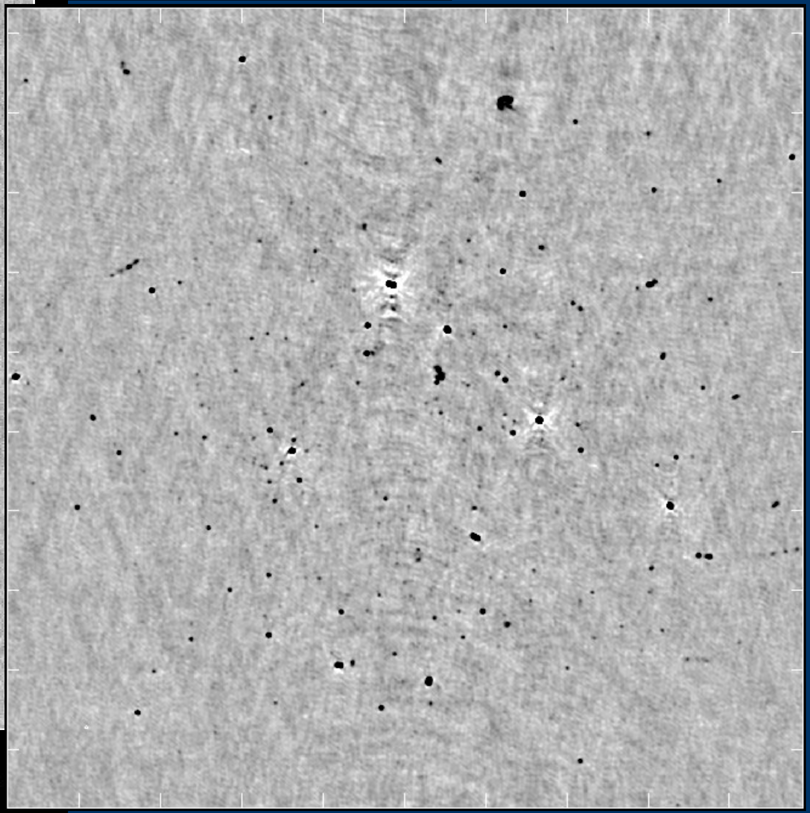
    - Peeling (since ?)/ VLA Squint correction (2008)

An example: VLA @ 1.4 GHz





~30 µJy/b @4"
(Data from:
Fomolant et al.)



Full beam imaging limits

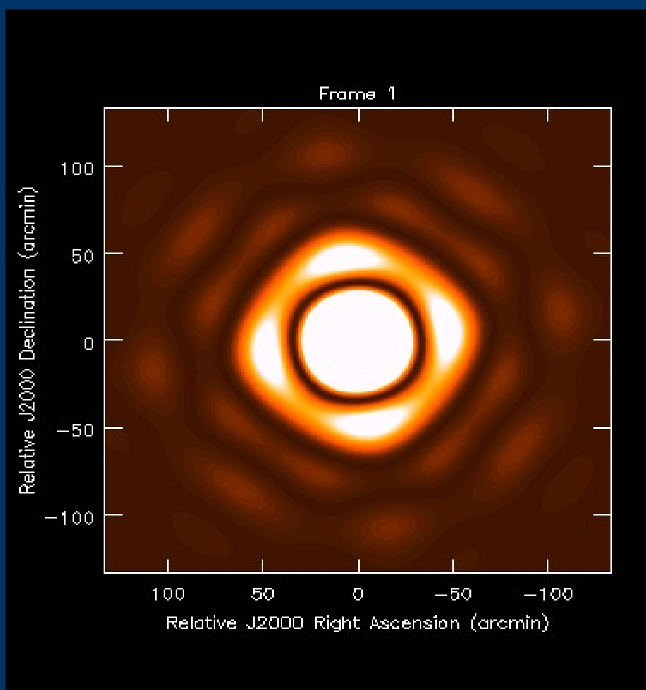
- Limits due to rotation of asymmetric PB
  - Error in PB model max. @ ~10% point
  - Max. in-beam error signal @ 50% point
  - DR of few x 10^4: 1
  - Errors higher in the first sidelobe

- Limits due to antenna pointing errors
  - In-beam max. error signal at 50% point
  - DR of a few x 10^4:1
  - Limits for mosaicking would be worse
    - Significant flux at half-power and side-lobes for many pointings

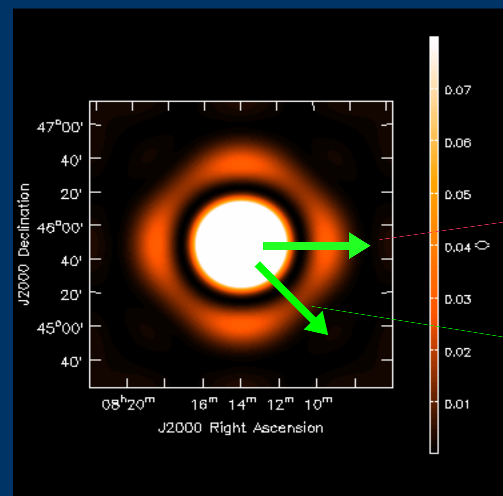
Primary beam effects

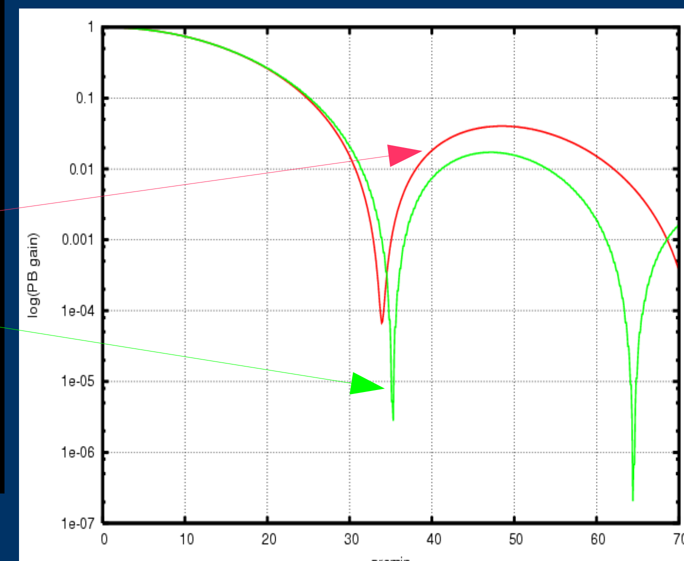
- EVLA full-beam, full-band, full-pol imaging

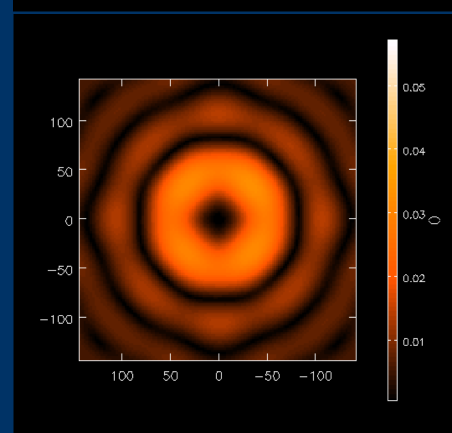
PB rotation, pointing errors



PB variation across the band
EVLA: Sources move from main-lobe to side-lobes







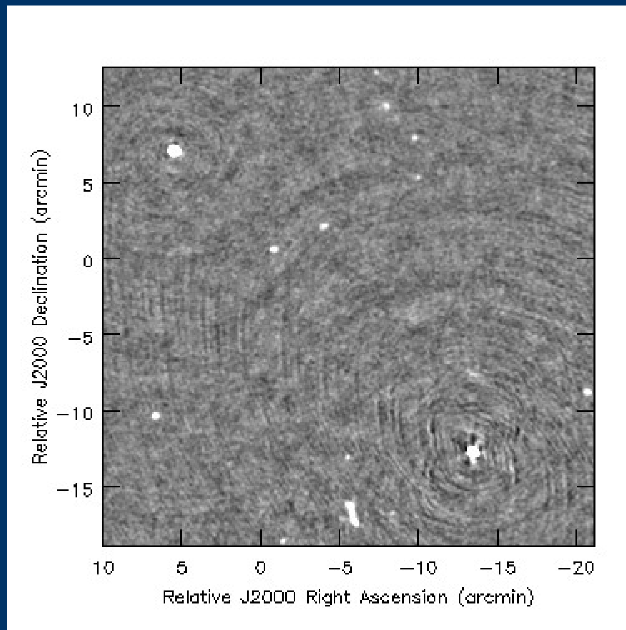
Cross hand power pattern

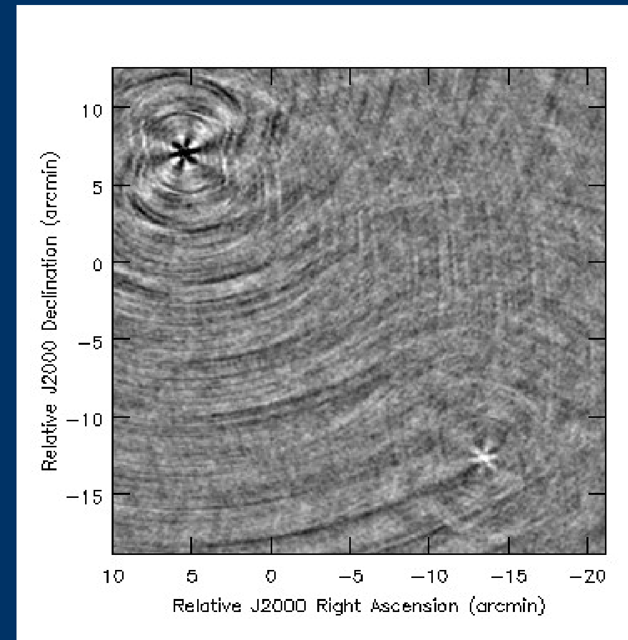
PB gain varies as a function time, frequency and direction in the sky

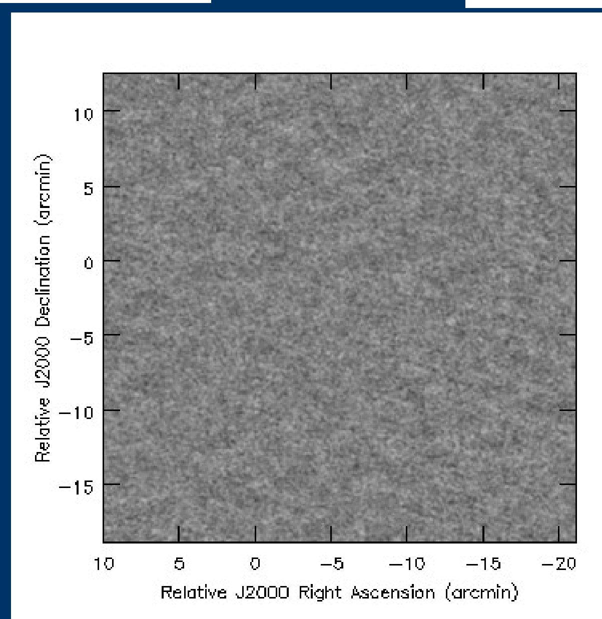
PB correction

- AW-Projection algorithm
  (Bhatnagar et al. A&A,487, 419, 2008)
  - Time and poln. Parametrization of the PB
  - No assumption about the sky emission
  - Scales well with imaging complexity
  - Straightforward to integrate with algorithms to correct for other errors (MFS, W-Projection, MS/Asp-Clean)
  - Requires a model for the Aperture Illumination

Example:VLA Stokes-I,V imaging

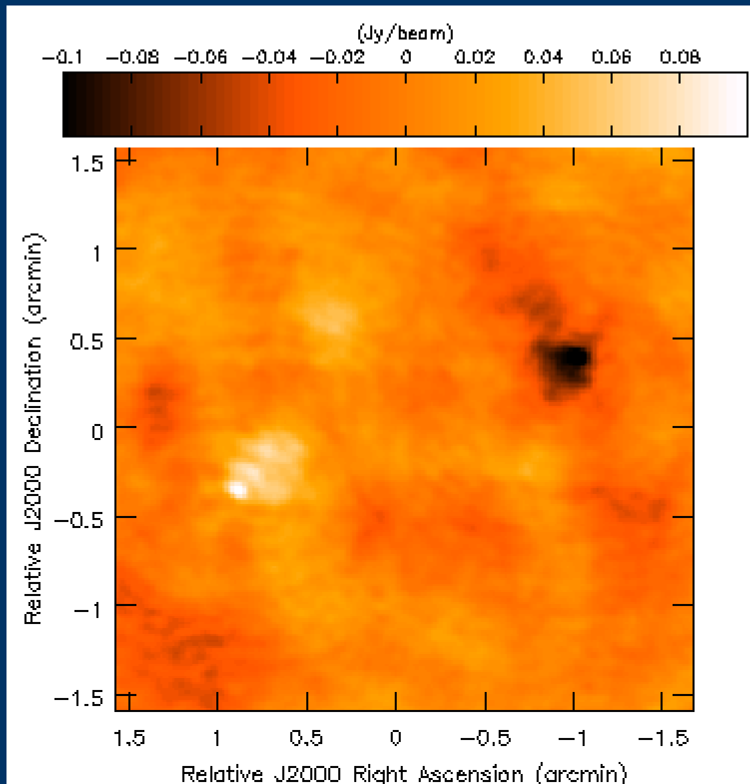


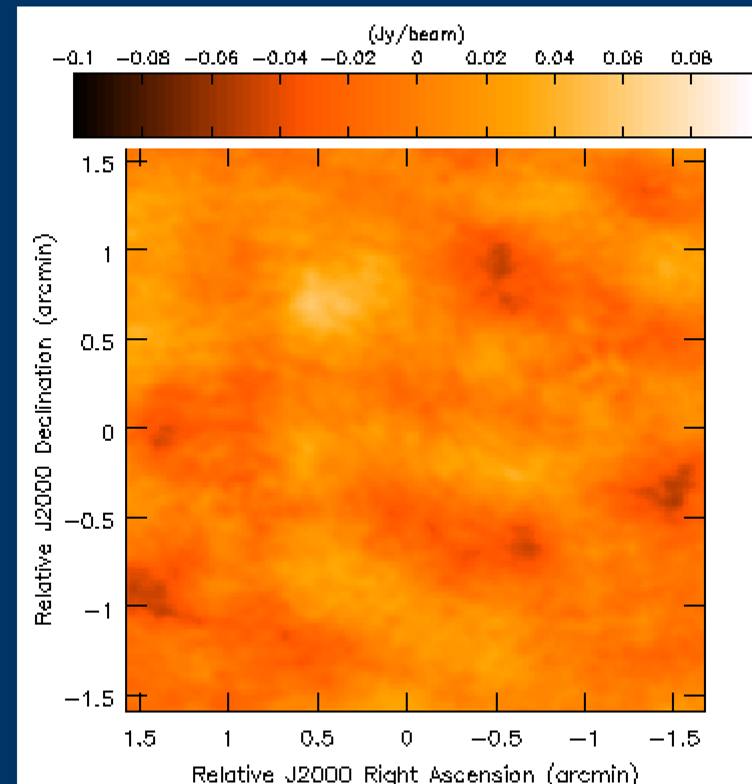




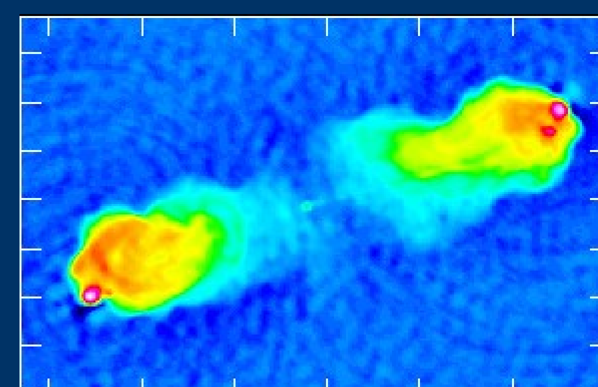
Using PB model
by W. Brisken
(EVLA Memo 58)

Example: Extended emission

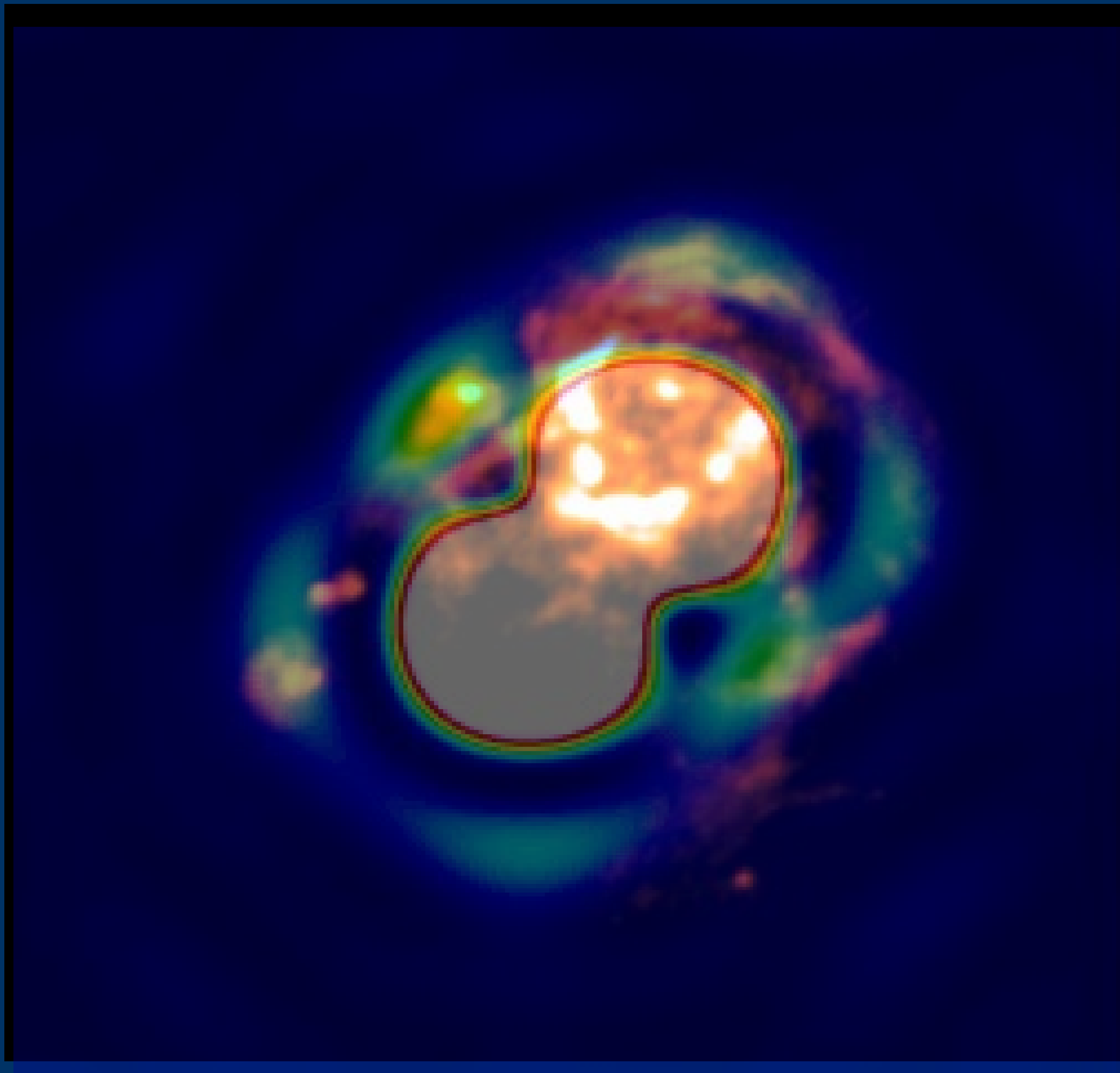


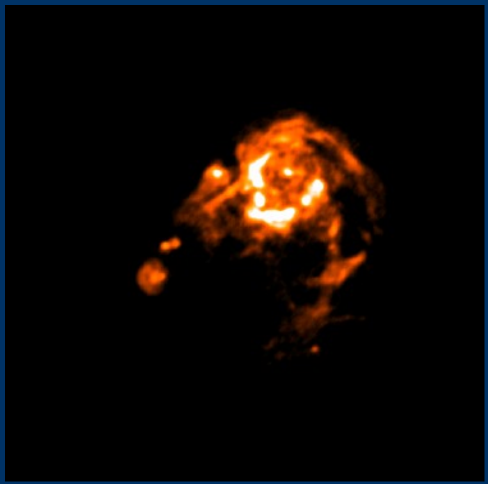


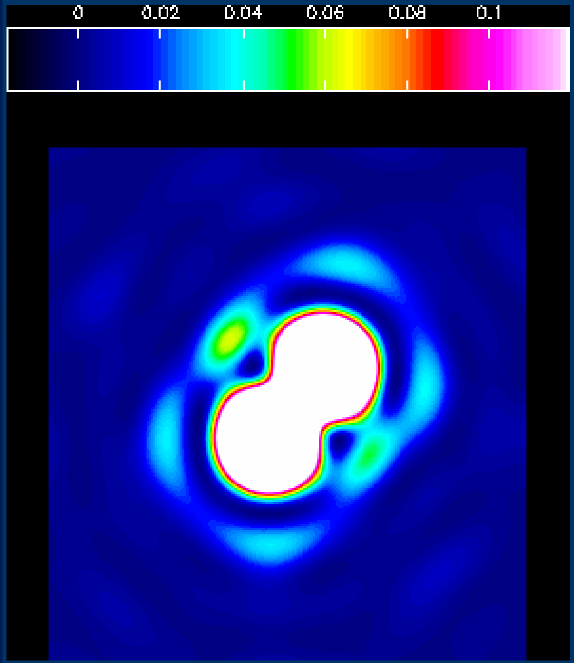
- Stokes-V imaging of extended emission
  - Algorithms designed for point sources will not work
  - Need more sophisticated modeling of the extended emission



Example: PB effects in mosaicking

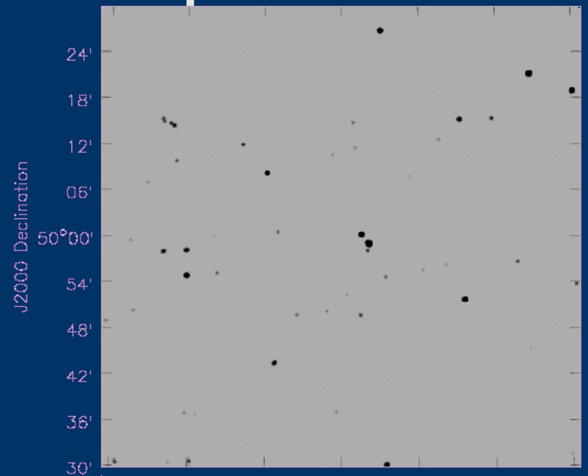




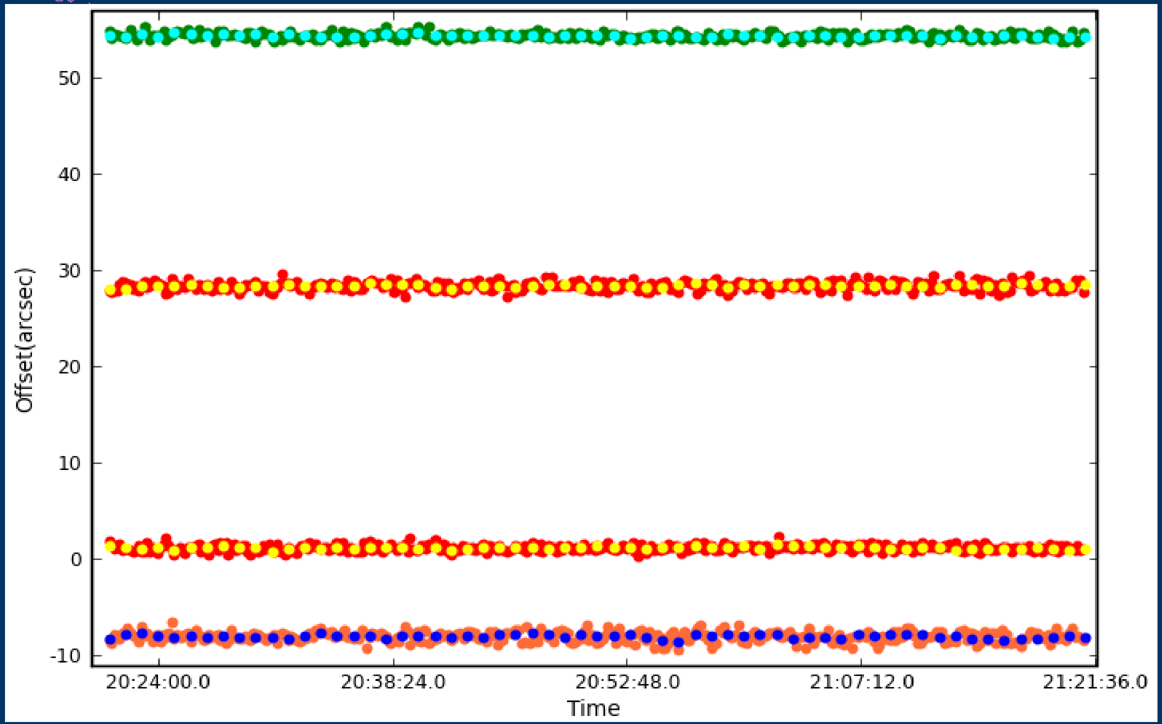


Pointing SelfCal: Solver

- PB parametrized for pointing errors



Model image: 59 sources from NVSS.
Flux range ~2-200 mJy/beam



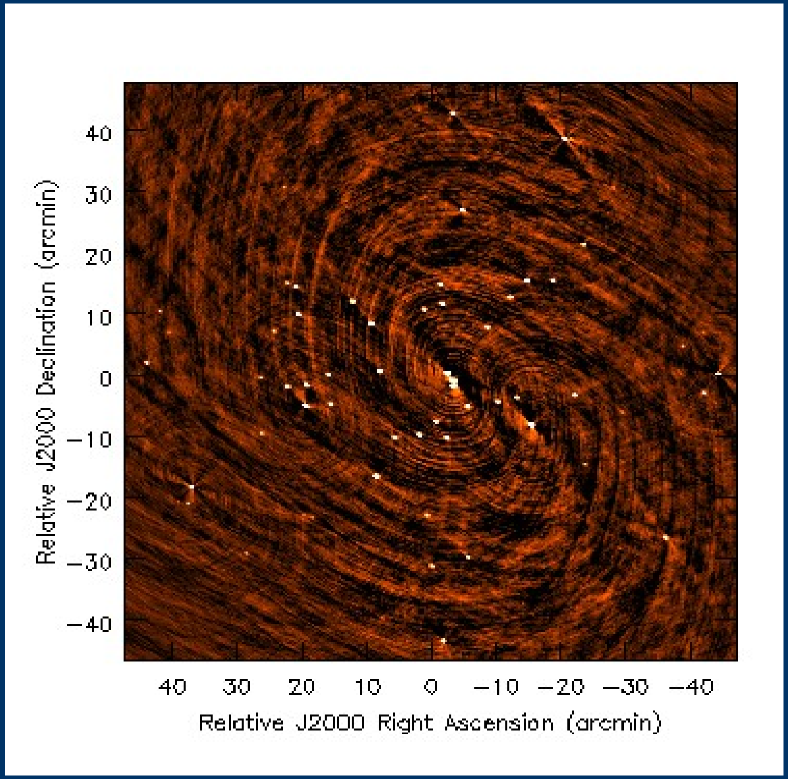
Typical antenna pointing offsets for VLA as a function of time

Over-plotted data: Solutions at longer integration time

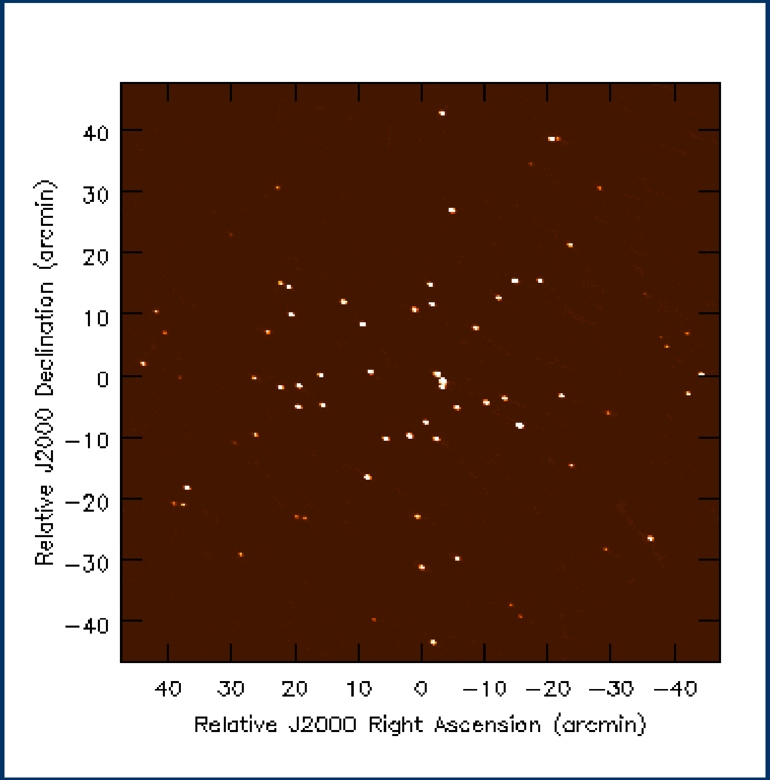
Noise per baseline as expected from EVLA

Pointing SelfCal: Correction

$$V_{ij}^{Obs}(t,\nu)=W_{ij}(\nu,t)\int PB(\mathbf{s},\nu,t)\ I(\nu)\ e^{\iota\,\mathbf{s}.\mathbf{b}_{ij}}\ d\,\mathbf{s}$$



- No pointing correction:
- RMS ~ 15μJy/b



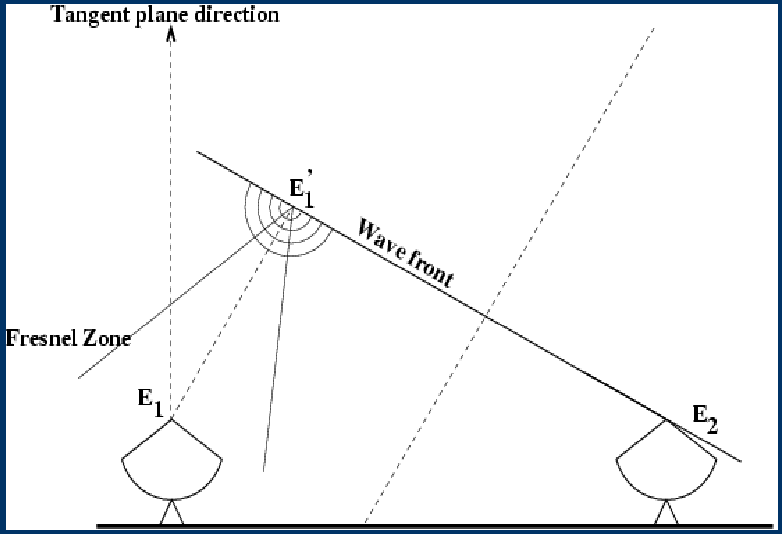
- After pointing correction:
- RMS ~ 1μJy/b

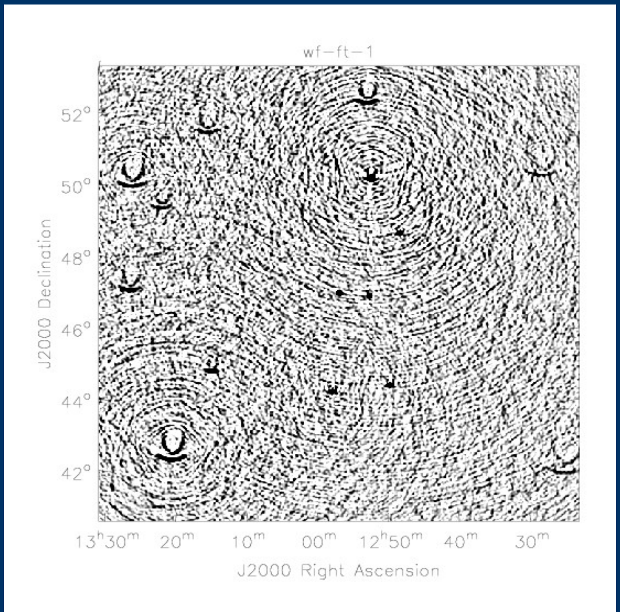
(Bhatnagar, Cornwell & Kolap, EVLA Memo #84/paper in prep.)

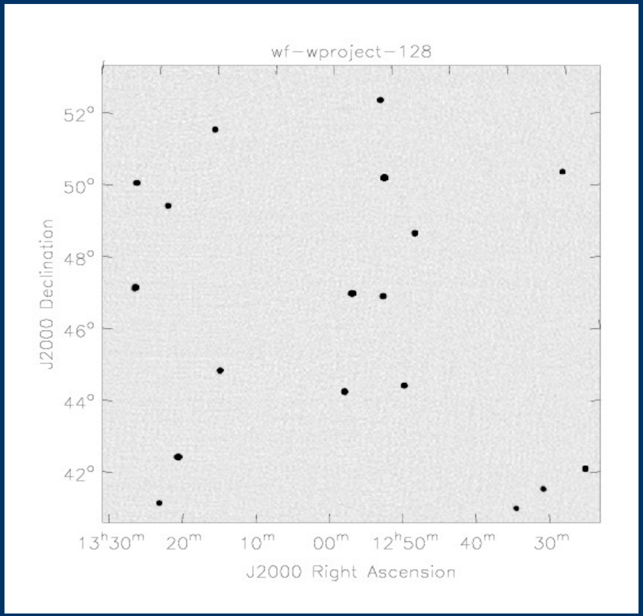
W-Projection algorithm

- $V(u,v,w)=G(u,v,w)*V(u,v,w=0)$

  $where\, \overline{G}(l,m,w)=e^{2\pi\iota\left[w\left(\sqrt{1-l^2-m^2}\right)\right]}$
- $E_1=E'_1(u,v,w)$ propagated using Fresnel diffraction
- Away from the phase center, sources are distorted

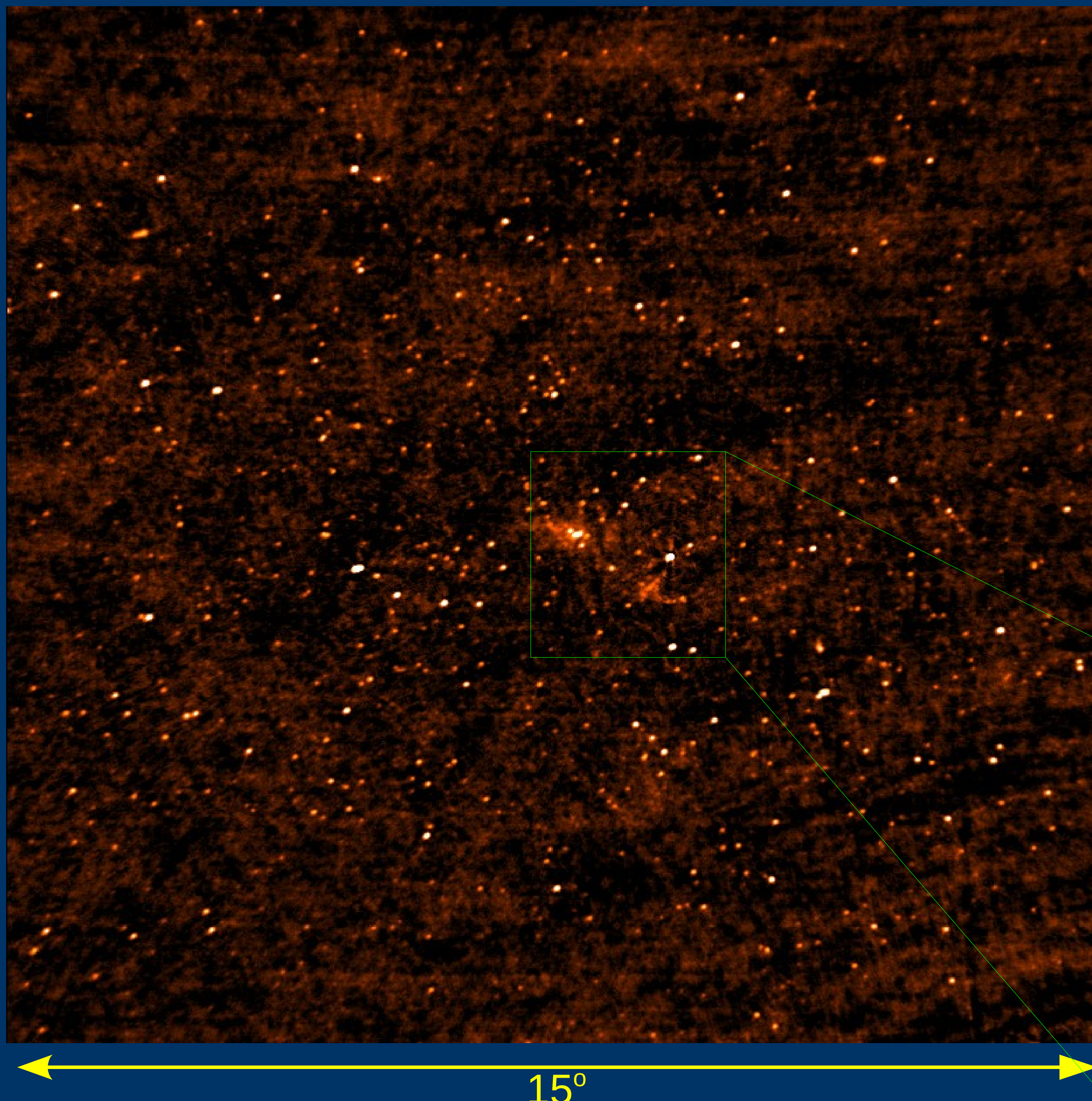


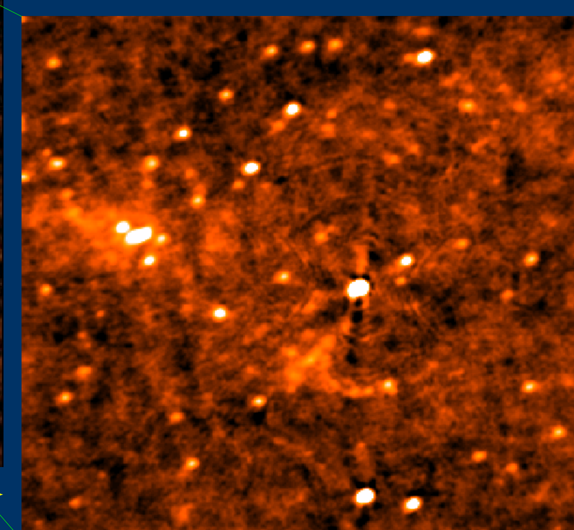




(Cornwell, Kolap & Bhatnagar, EVLA Memo (2004), IEEE Special Issue on RA, (in press))

Example: VLA @ 74 MHz

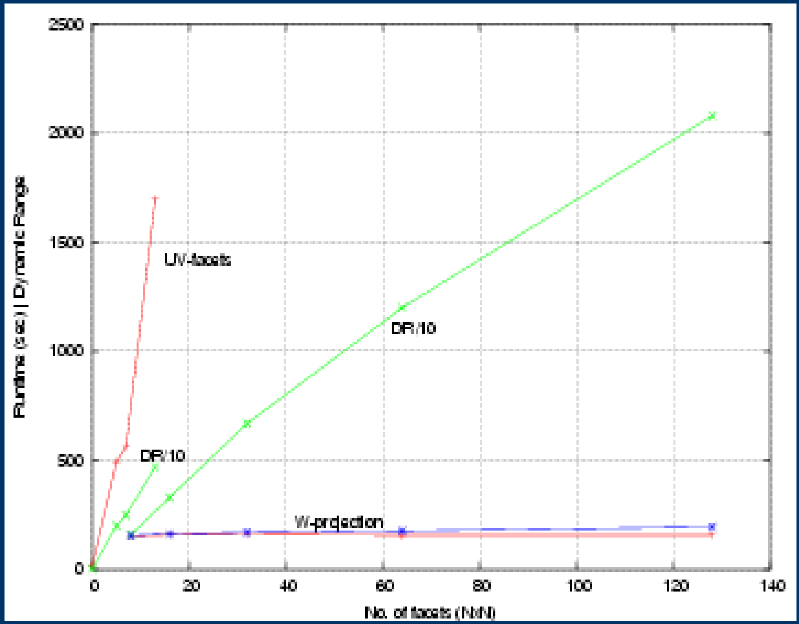




- Coma cluster at 74 Mhz/VLA
- 30 arcsec resolution, RMS ~30mJy/beam
- Imaged using the W-projection algorithm (Golap)

Computing

- Imaging scaling laws
  - Non co-planar baseline correction
    - W-Projection: $(N^2_{wproj} + N^2_{GCF})\, N_{vis}$
    - Faceting: $N^2_{facets}\, N^2_{GCF}\, N_{vis}$
  - AW-Projection: $N^2_{GCF} * N_{vis}$
  - Peeling: $N_{comp} * N_{vis} * ?$



- Scaling laws for DD solvers
  - FFT-based transforms: $N^2_{GCF} * N_{vis} * N_{iter} * N_{params}$
  - DFT-based transforms: $N_{comp} * N_{vis} * ? * N_{iter} * N_{params}$
  - N_{vis}: $10^{8\text{-}10}$, N^2_{GCF}: 50-100 , N_{comp}: $10^{4\text{-}5}$

TeraByte Initiative

- Initial tests: 512 channel, 4 Pol, T_{int} = 2s, VLA B-array, data size ~100GB
  - Standard continuum imaging: 4K x 4K x 512, Stokes-I
  - Image size on disk: 3 x 32GB
- Timing
  - Flagging (quack only) : 1h
  - Calibration solver G-Jones : 2h15m
  - Calibration solver B-Jones : 2h35m
  - Correction : 2h
  - Imaging : 20h
  - Export FITS : 2h
- Effective data I/O: ~800 GB

Near future data sizes

- Data I/O : Computing ~ 3:2 (at least)
- Expected average data rates about 10x larger
- Manual processing (data flagging, calibration and imaging) not an option
  - Need robust and efficient algorithms
  - Need robust heuristics
  - Need pipe line processing
  - Need all of this to run in a parallel computing environment
- Interoperability
  - Possible now via FITS
  - Data sizes is the problem!
  - Lower level software exchange is better
    - Sociological rather than technological problem!