

TBD

BDT:

Beams, Data compression (& Deconvolution), Tasse (Cyril)

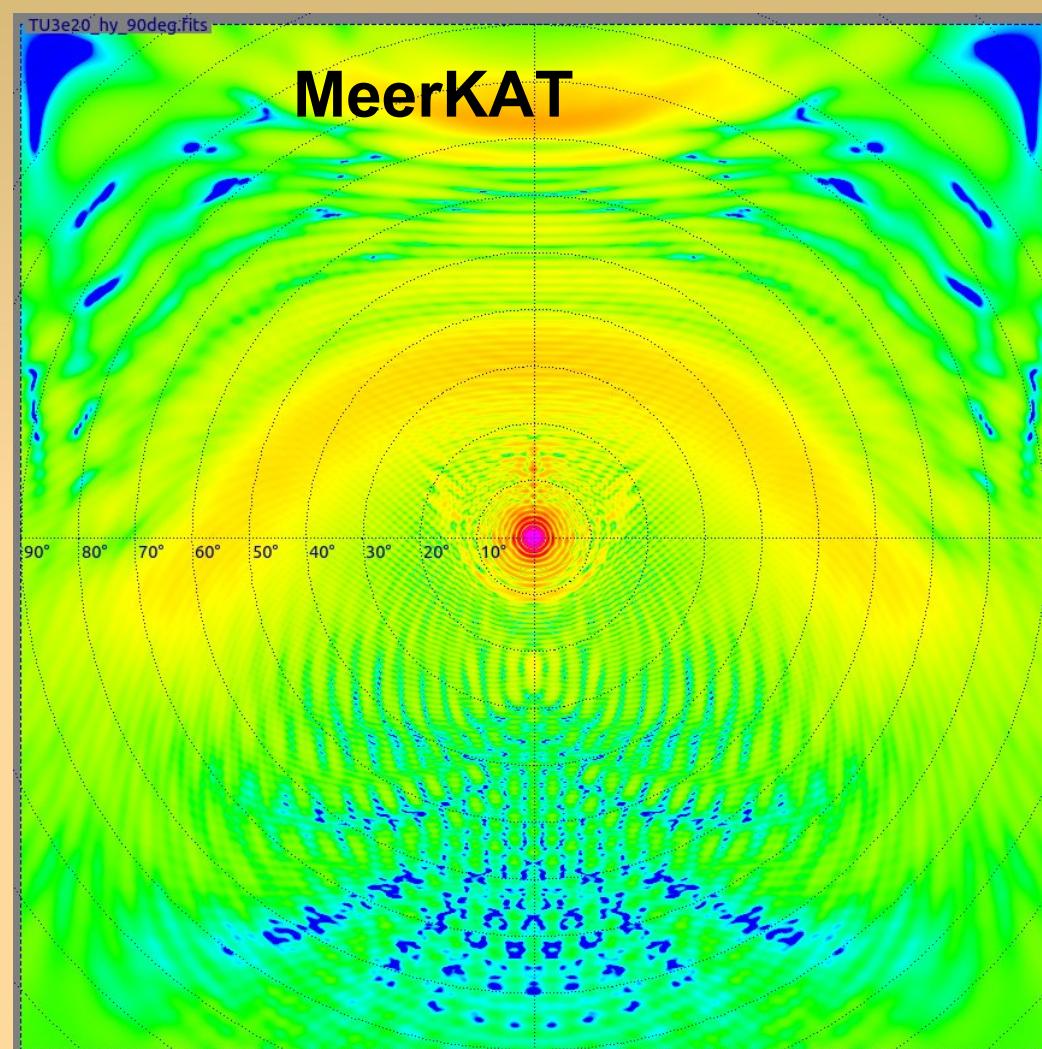
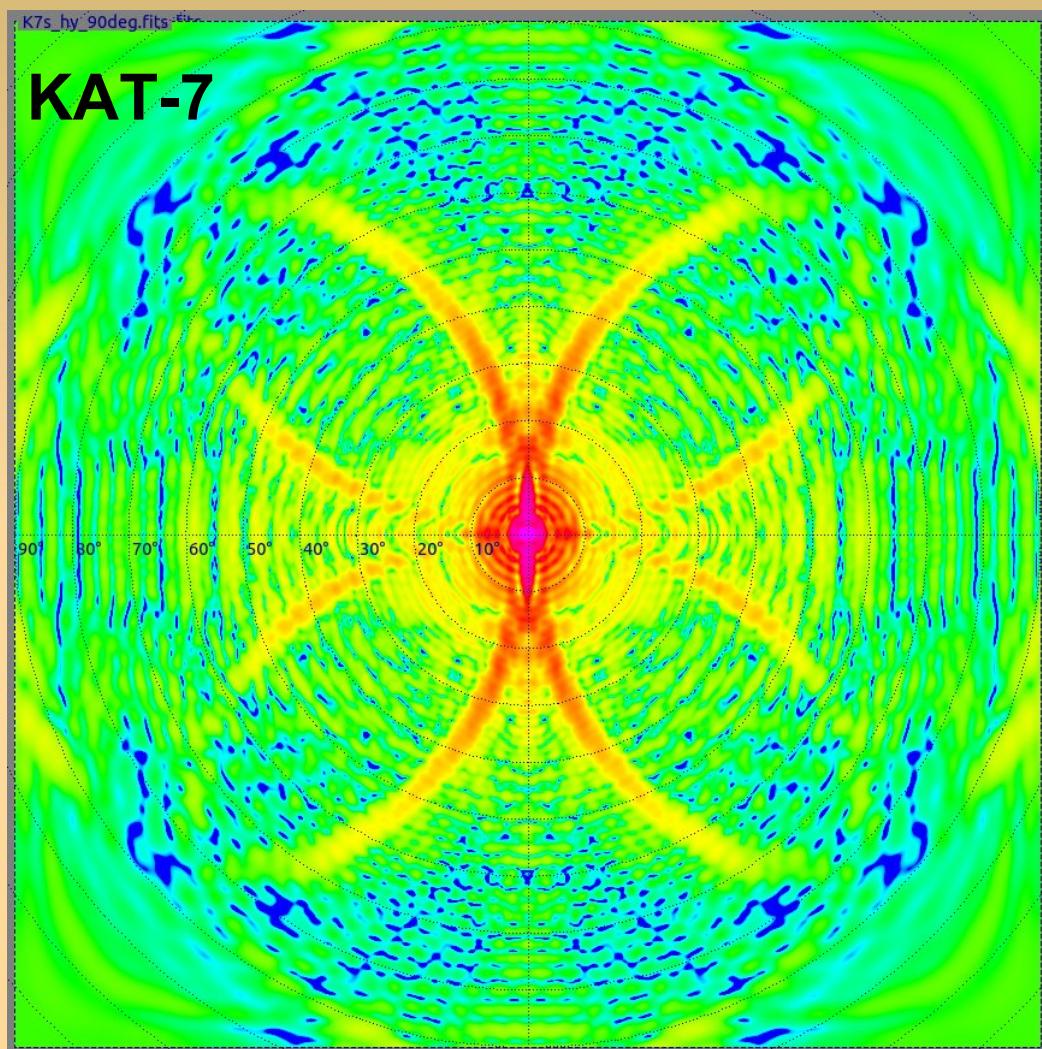
O. Smirnov

(Rhodes Centre for Radio Astronomy
Techniques & Technologies
& SKA Africa)



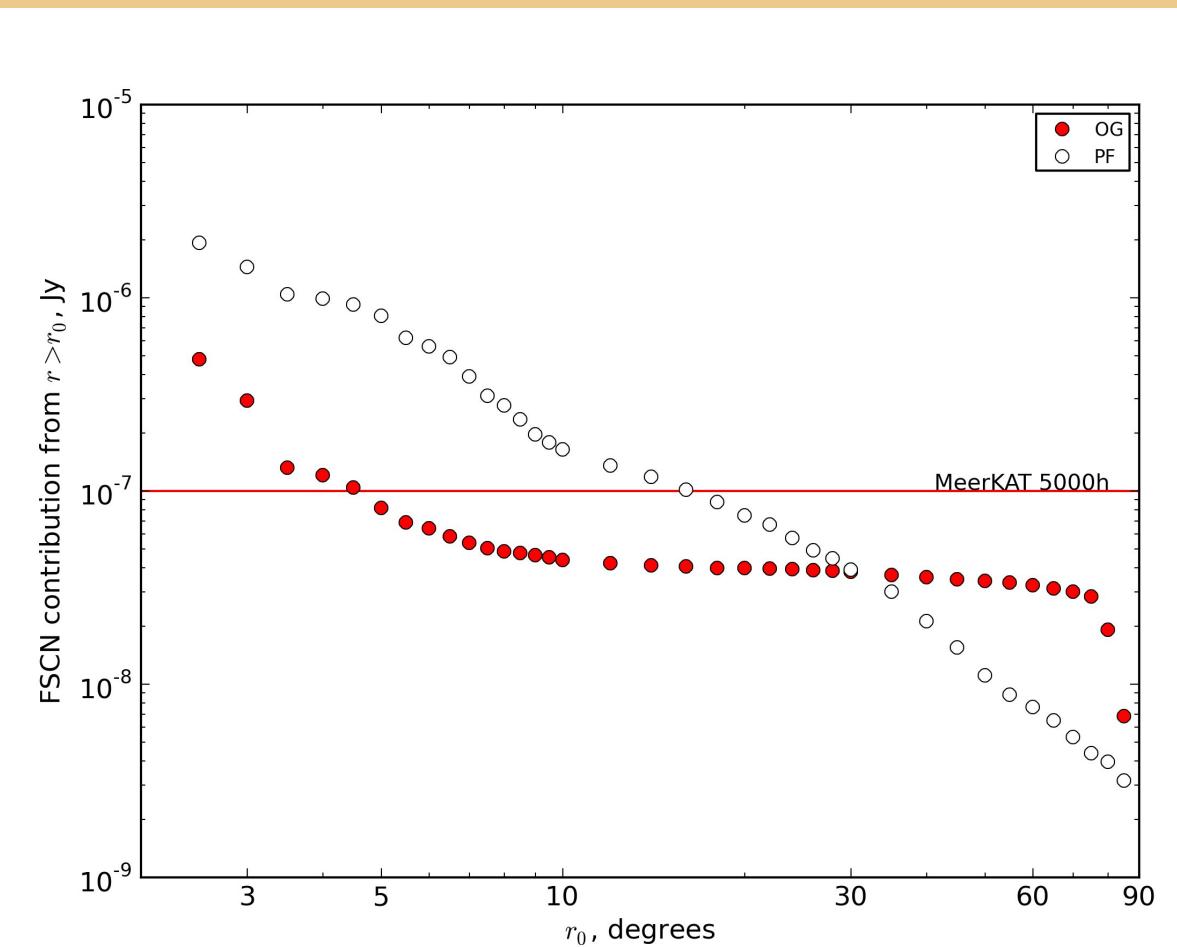
Part I: Beams & Far Sidelobe Confusion

MeerKAT/KAT-7 Sidelobes

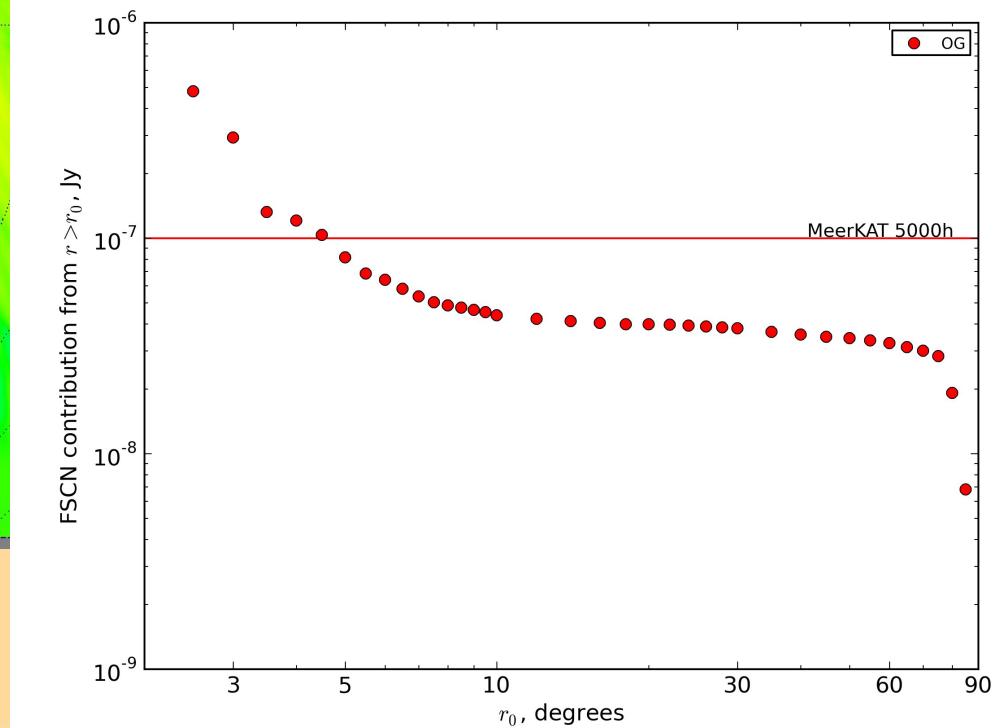
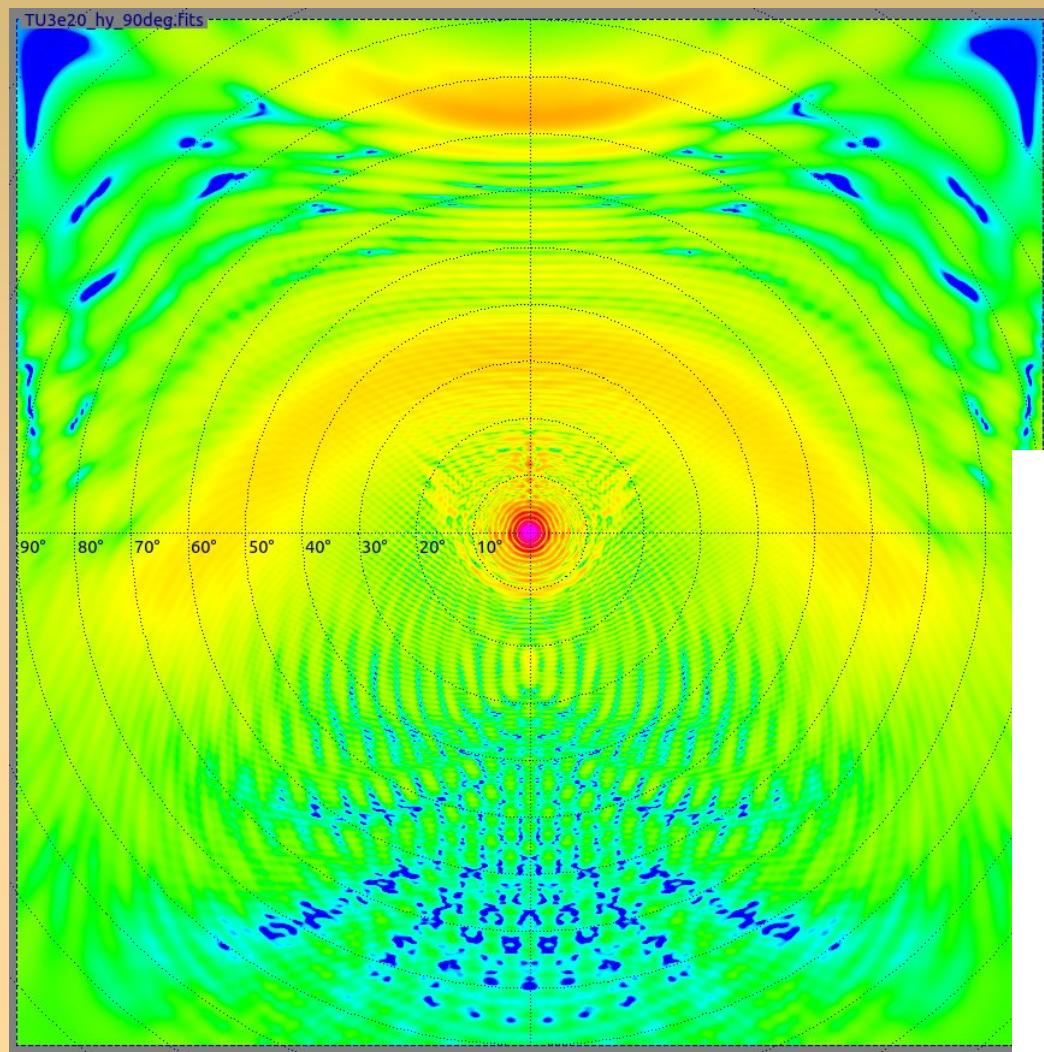


SCN Cost Curves

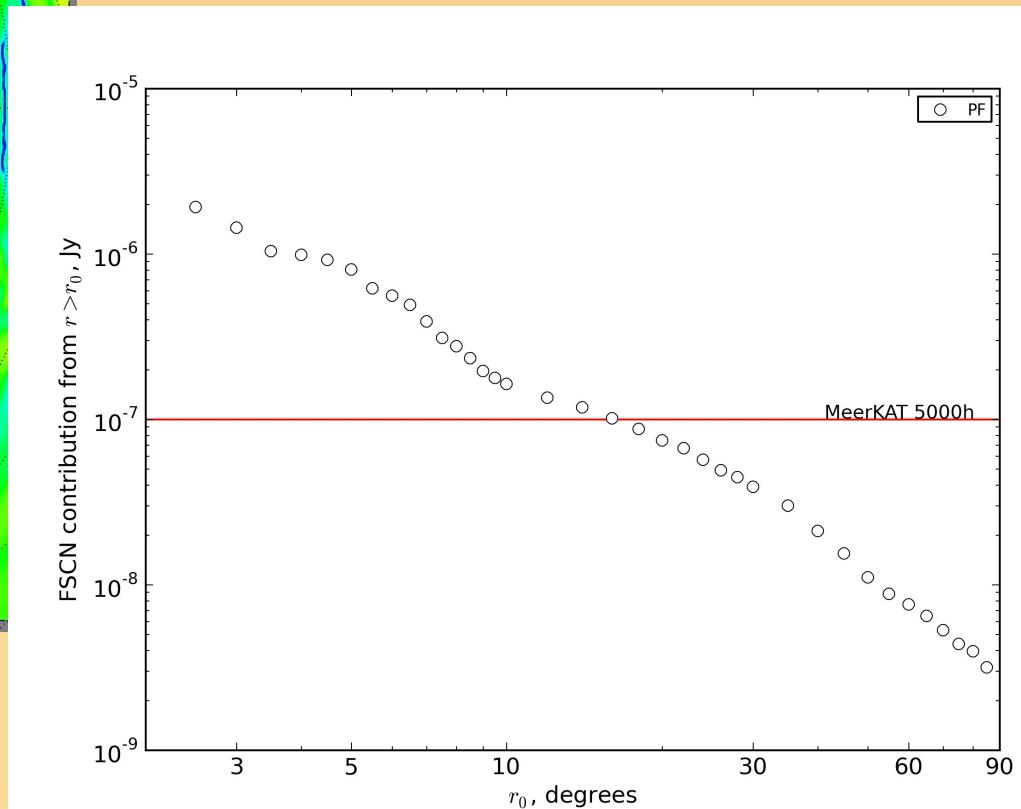
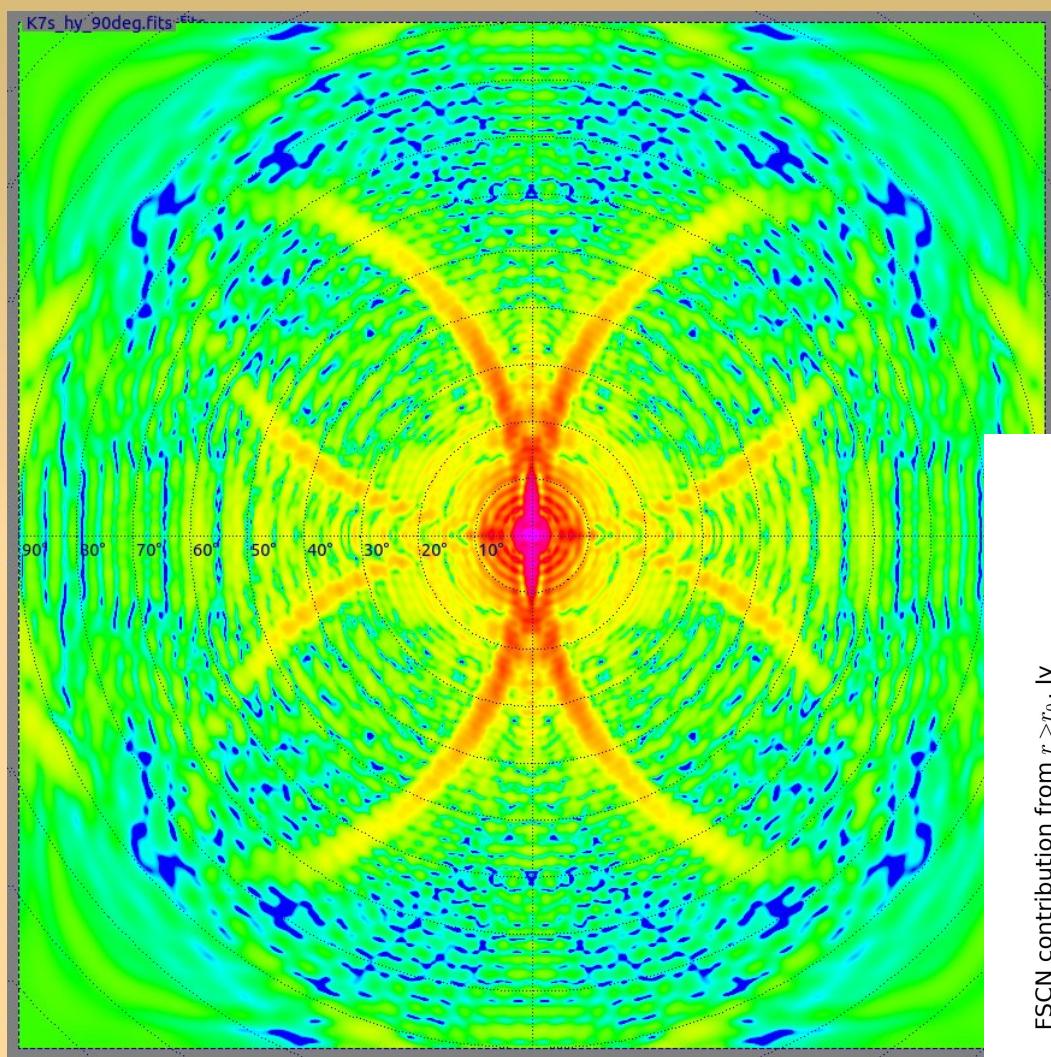
- This shows, as a function of r , the SCN contribution from sources $r \geq r_0$
- i.e. how far out do we have to image & deconvolve to drive SCN below a given level?



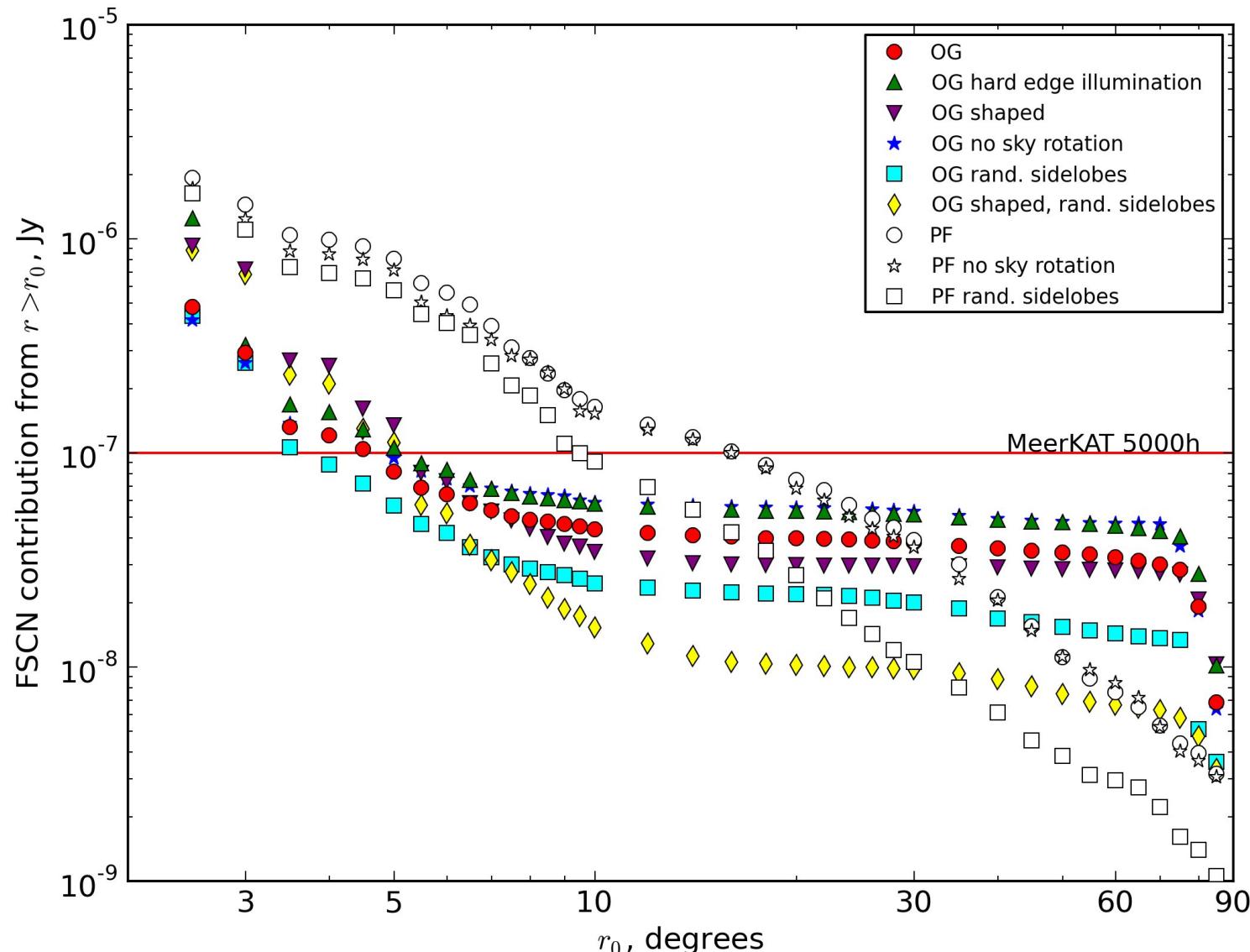
Cost Curve: Offset Gregorians



Cost Curve: Prime focus



The Picture For Many Dishes



Part II: Beam Smoothness & “Calibration Noise”

Calibration “Noise”

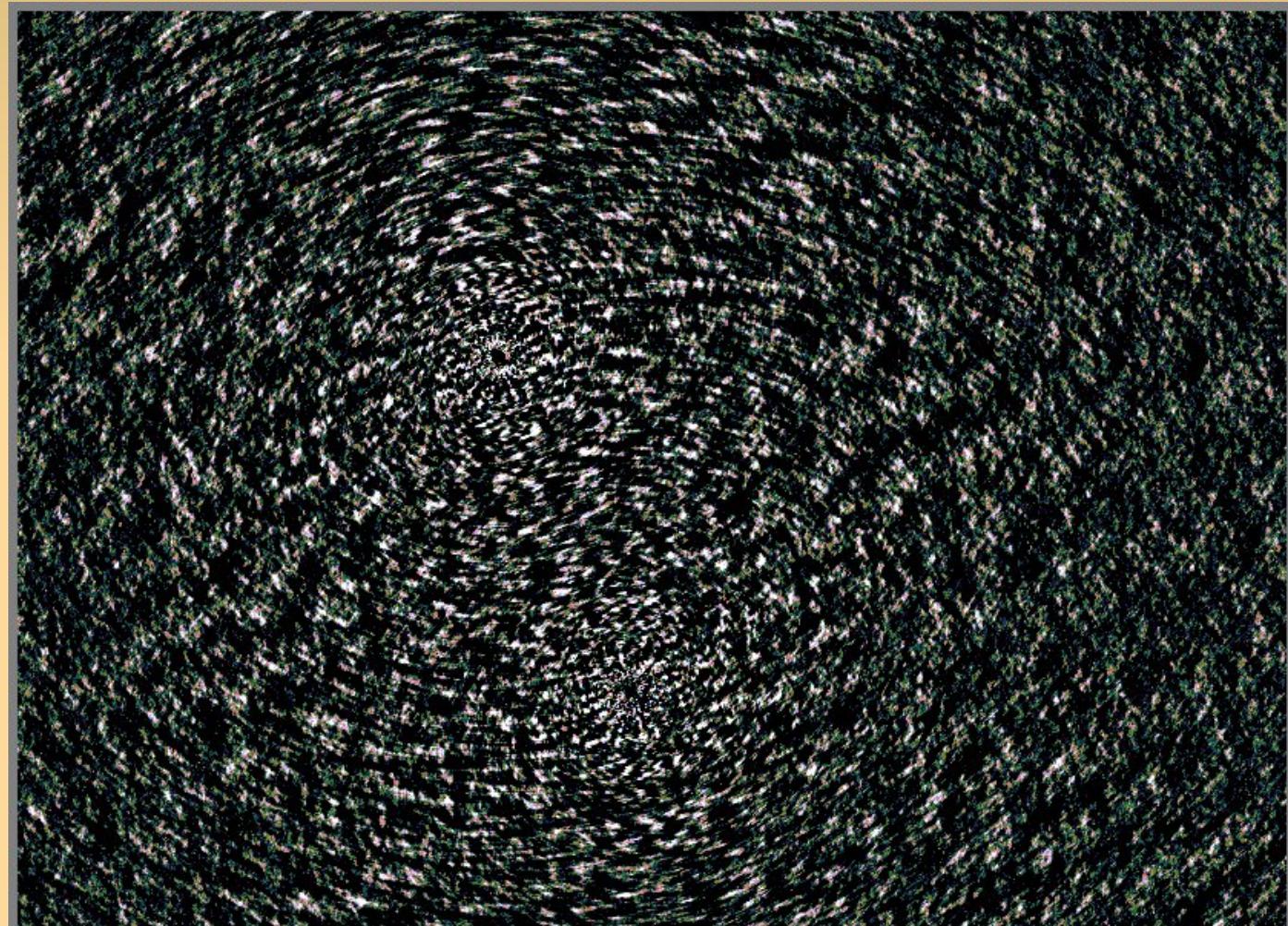
- We have been very successful at eliminating DDE-related artefacts via direction-dependent solutions
- And by “eliminating” we mean “driving below the (thermal) noise”
 - ...by which we really mean “sweeping under the carpet”
- So, how do we estimate what we have “swept”, and can it come back to haunt us?

Distilling Out The Artefacts

- Two simulations:
 - (I) Full sky + instrumental errors + noise
 - (II) Bright sources only + instrumental errors
 - ...with the same instrumental errors
 - Which include DI gains and pointing errors
- Calibrate the “full sky”
- Apply solutions to (II) and subtract model
 - Result: “distilled” artefacts that would normally be below the noise

Example: MeerKAT

- Residuals from a pair of 1 Jy sources
- DI cal only
- Here, rms
 $4.2 \mu\text{Jy}$

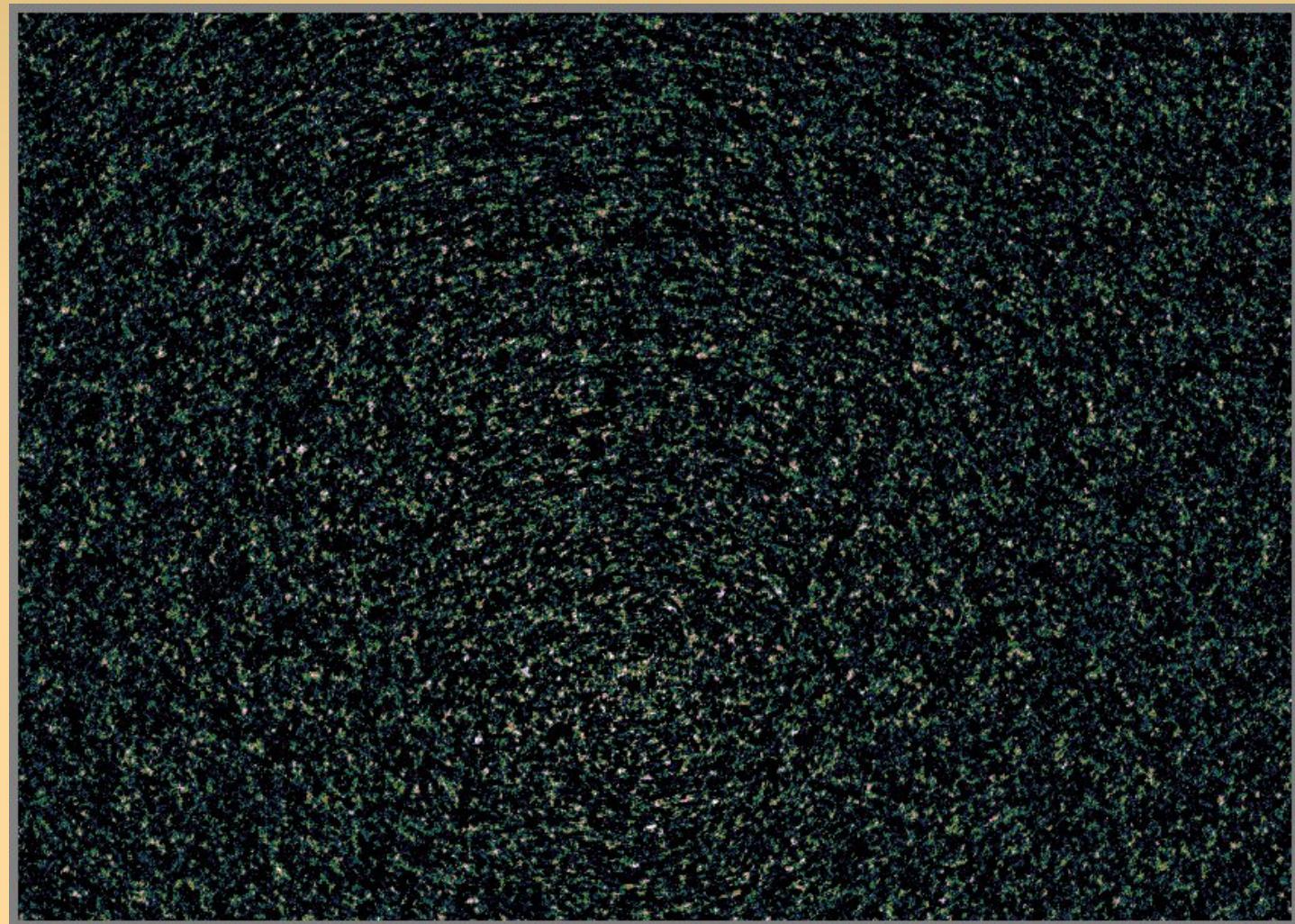


Distilling DDEs

- But nevermind, because direction-dependent solutions can take care of it, right?
- If we run a dE solution on the two contaminator sources, the resulting image (of the full residuals) becomes thermal noise limited; remaining artefacts are below the noise.
- But we can repeat the same distillation trick to see what's left anyway

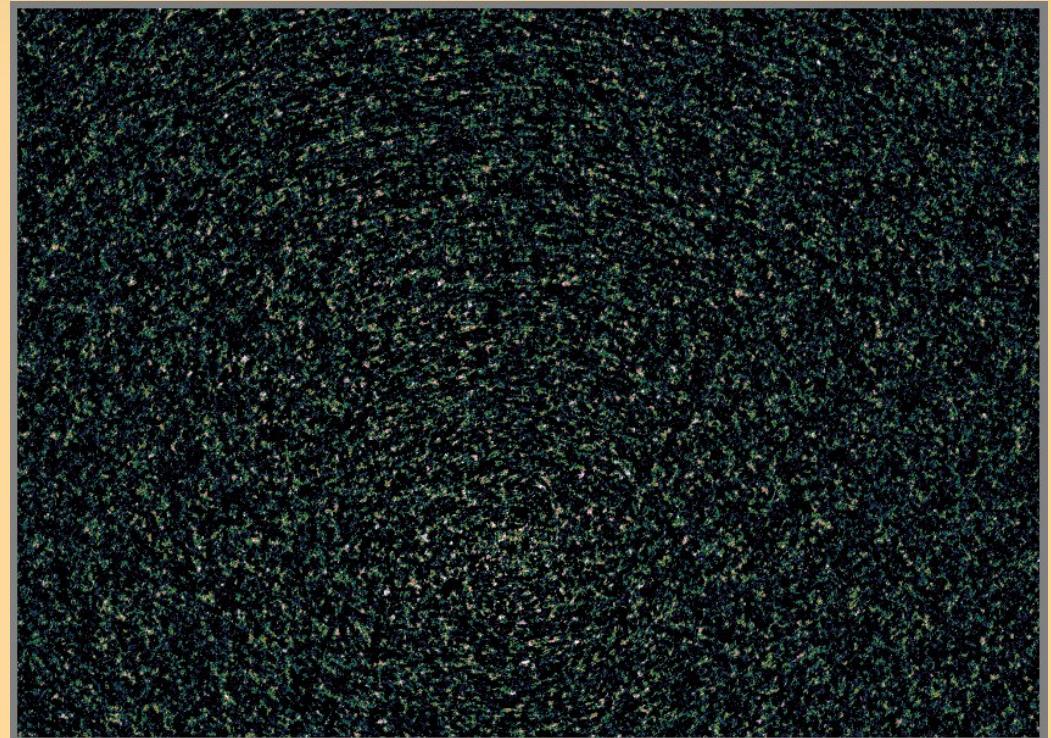
Calibration Noise, Post-dE

- Rms 2.6 μJy



Why Do We Care?

- Just an extra noise-like contribution that's below the thermal noise, so what's the big deal?
- But it can be a big deal if it doesn't average out
- Imagine repeating this independently for N pointings



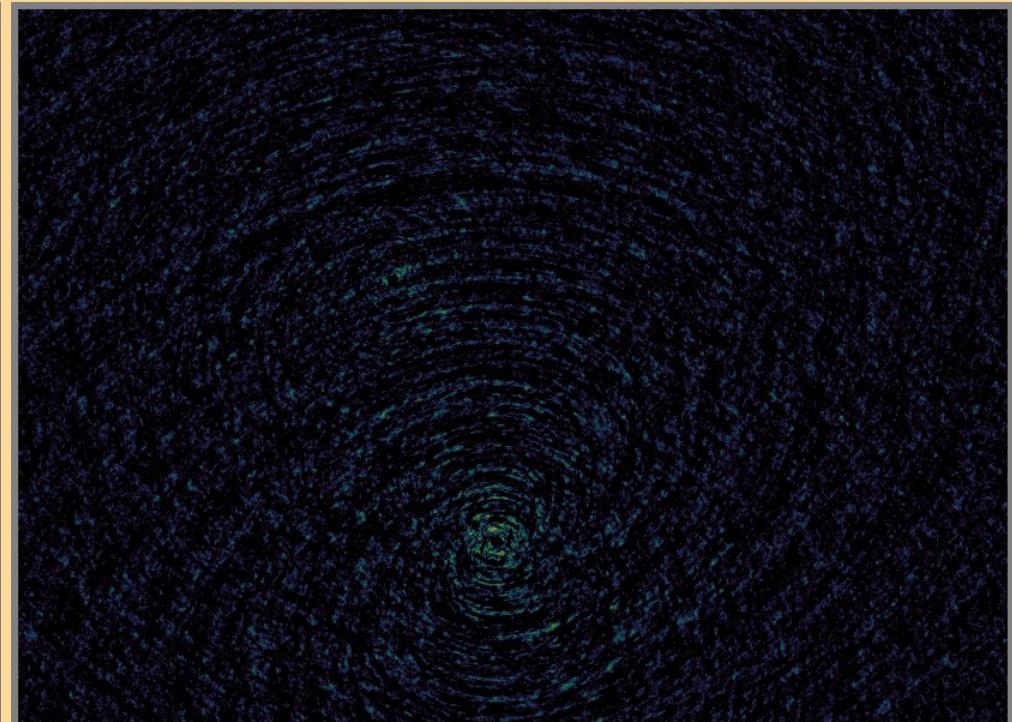
Mean Of 10 dE-Distills

- Structure shows up
- Does not average out as \sqrt{N}

1 distill, rms 2.6 μJy



10 distills, rms 1.2 μJy

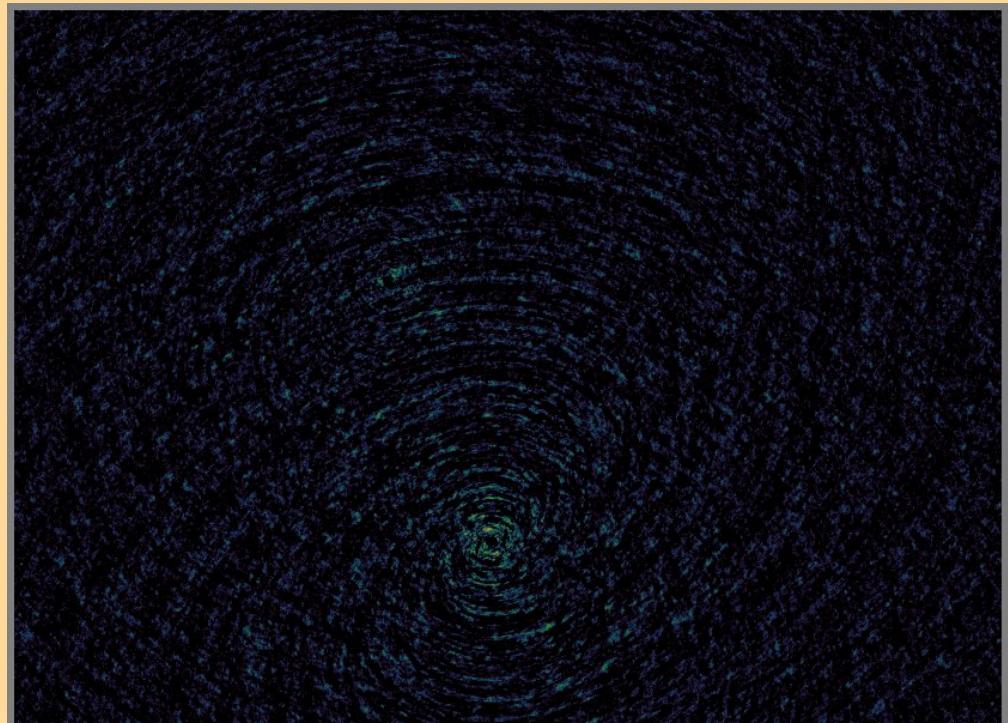
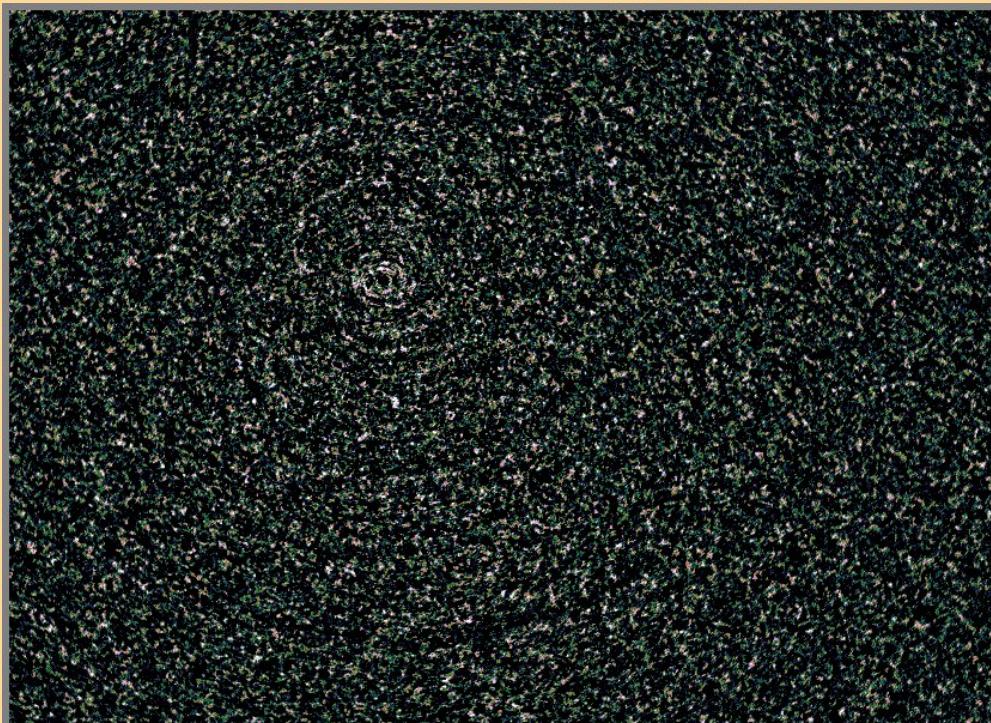


PF vs OG

- Repeat this experiment for PF and OG beam patterns
- Calibration “noise” for OG lower by a factor ~3

PF: rms 3.6 μJy

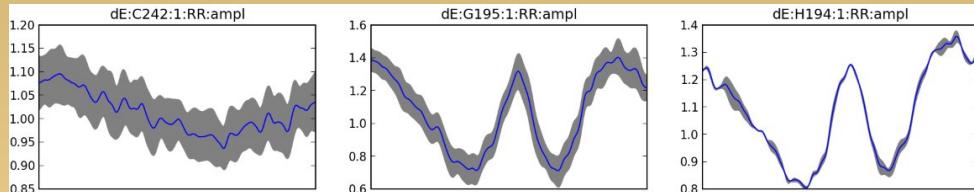
OG: rms 1.2 μJy



Why Does This Happen?

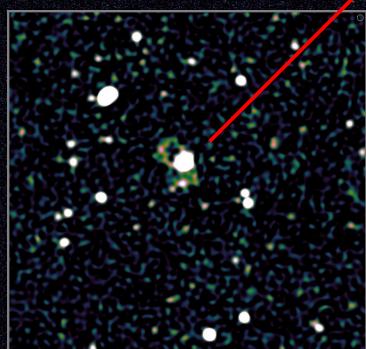
- My answer: beam smoothness
 - “Smoother” PB pattern is less sensitive to pointing error
- And now consider beam rotation...

JVLA 3.2m vs. WSRT 1.6m

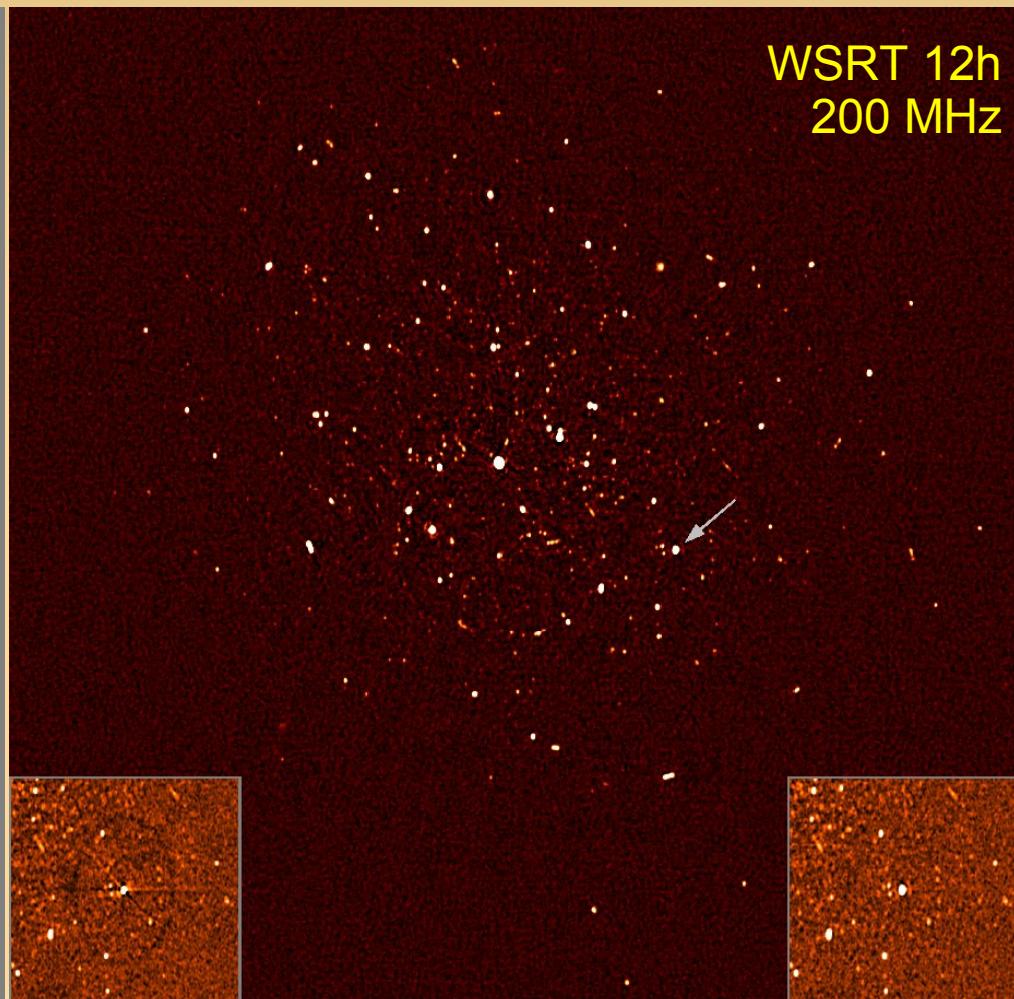


Uses stefcal for G and dE solutions

VLA C+D
~14h 192MHz

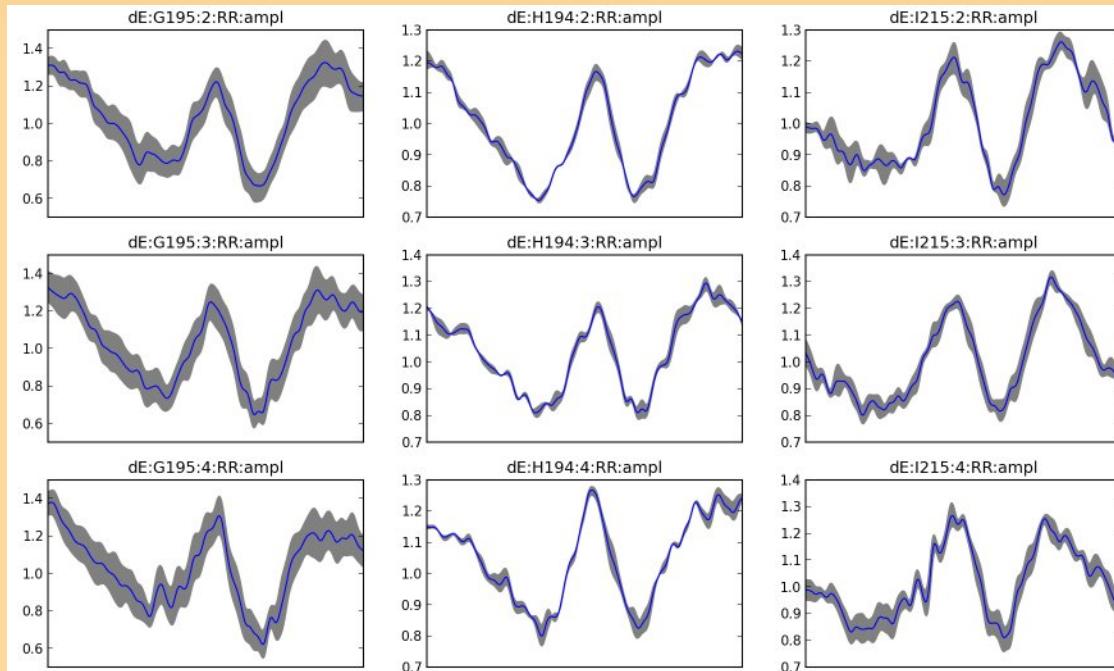


WSRT 12h
200 MHz

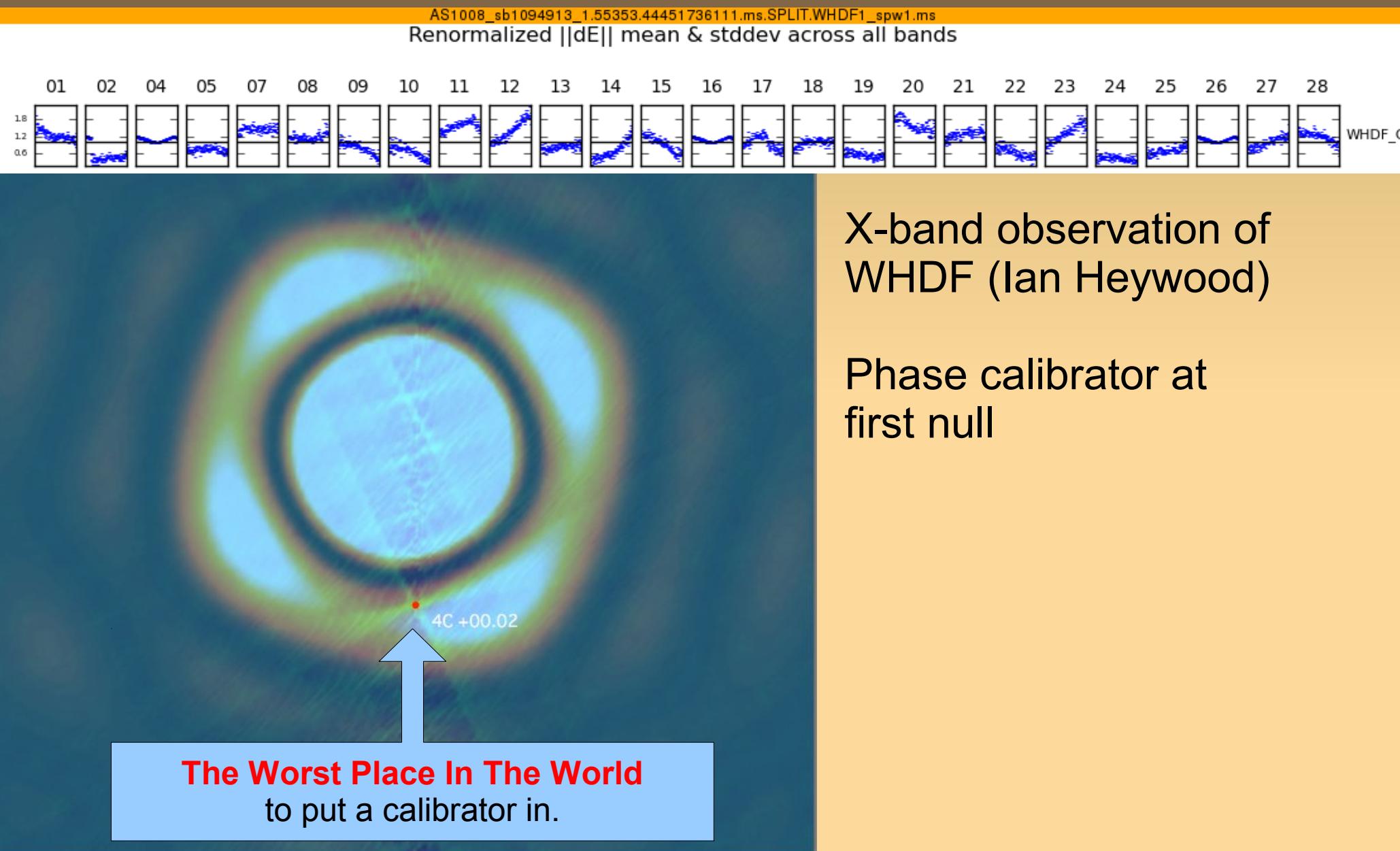


dE Solutions

- Large time variation
 - (A-Projections: higher cadence of recomputing deconvolution kernels)
- Differences antenna-to-antenna



Even More Antenna Differences



Conclusion I

- Smooth (or stationary) beams good
- Non-smooth beams bad
- OG smoother than PF

Part II: Data Compression

- Why so much data?
- Very high time/frequency resolution
 - In order to avoid smearing of long baselines
- Can we average more cleverly?

What Causes Averaging

- Averaging: $V_{\text{meas}}(u, v) = \frac{1}{\Delta t \Delta \nu} \int_{\Delta t} \int_{\Delta \nu} V(t, \nu) dt d\nu$

$$V(t, \nu) = V(\vec{u}(t)/\lambda)$$

$$V_{\text{meas}}(u, v) = \frac{1}{\text{bin}} \iint_{uv \text{ bin}} V(u, v) du dv = \iint V B du dv,$$

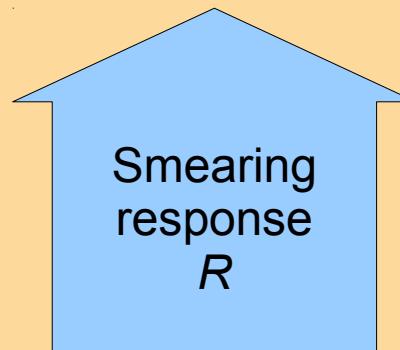
where B is a boxcar (“tophat”) function for the bin. This is a convolution:

$$V_{\text{meas}} \approx (V \circ B)S$$

Averaging \approx Convolution

$$I = \mathcal{F}^{-1}(V)$$

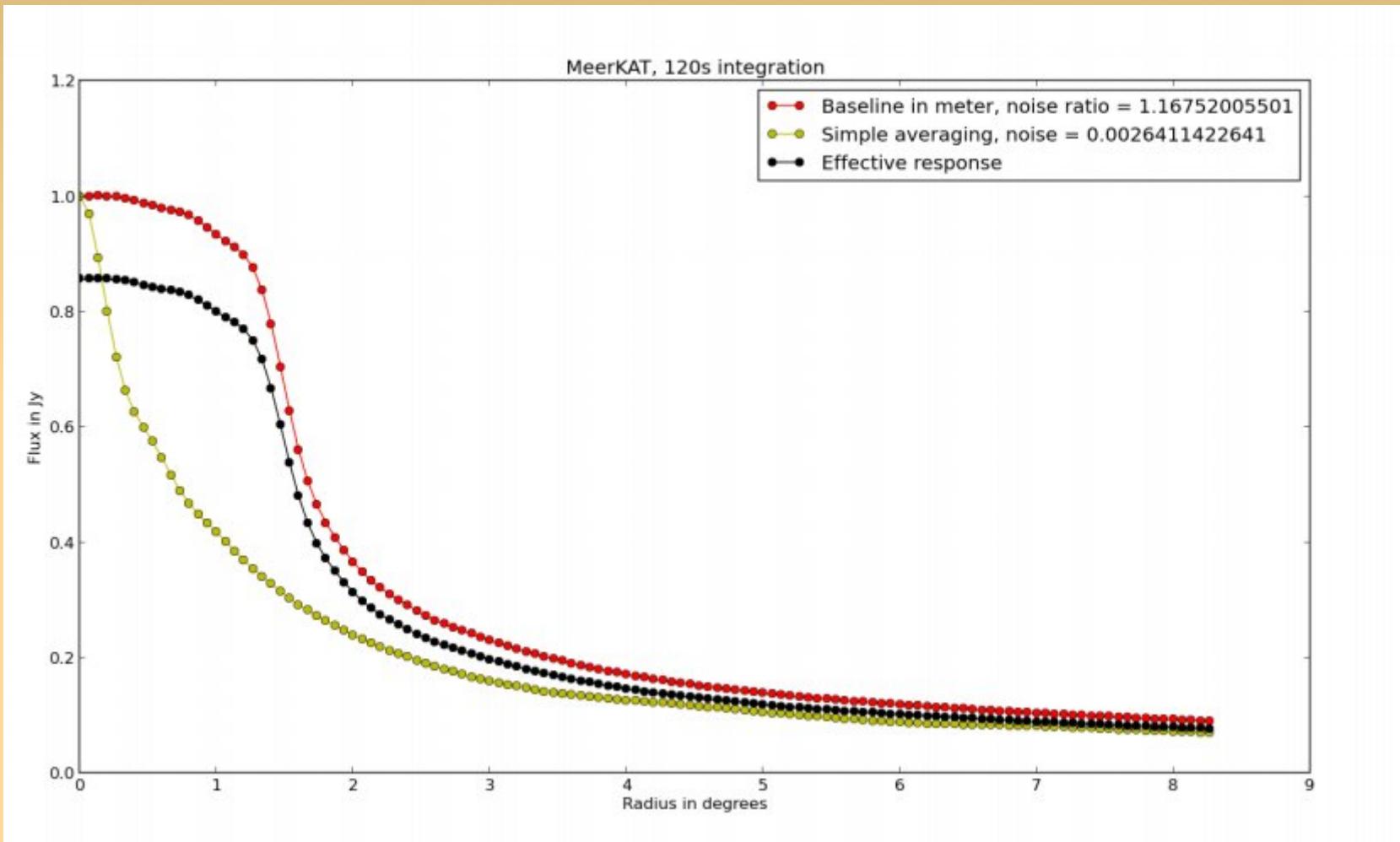
$$\mathcal{F}^{-1}(V \circ B \cdot S) = I \cdot \mathcal{F}^{-1}(B) \circ \text{PSF}$$



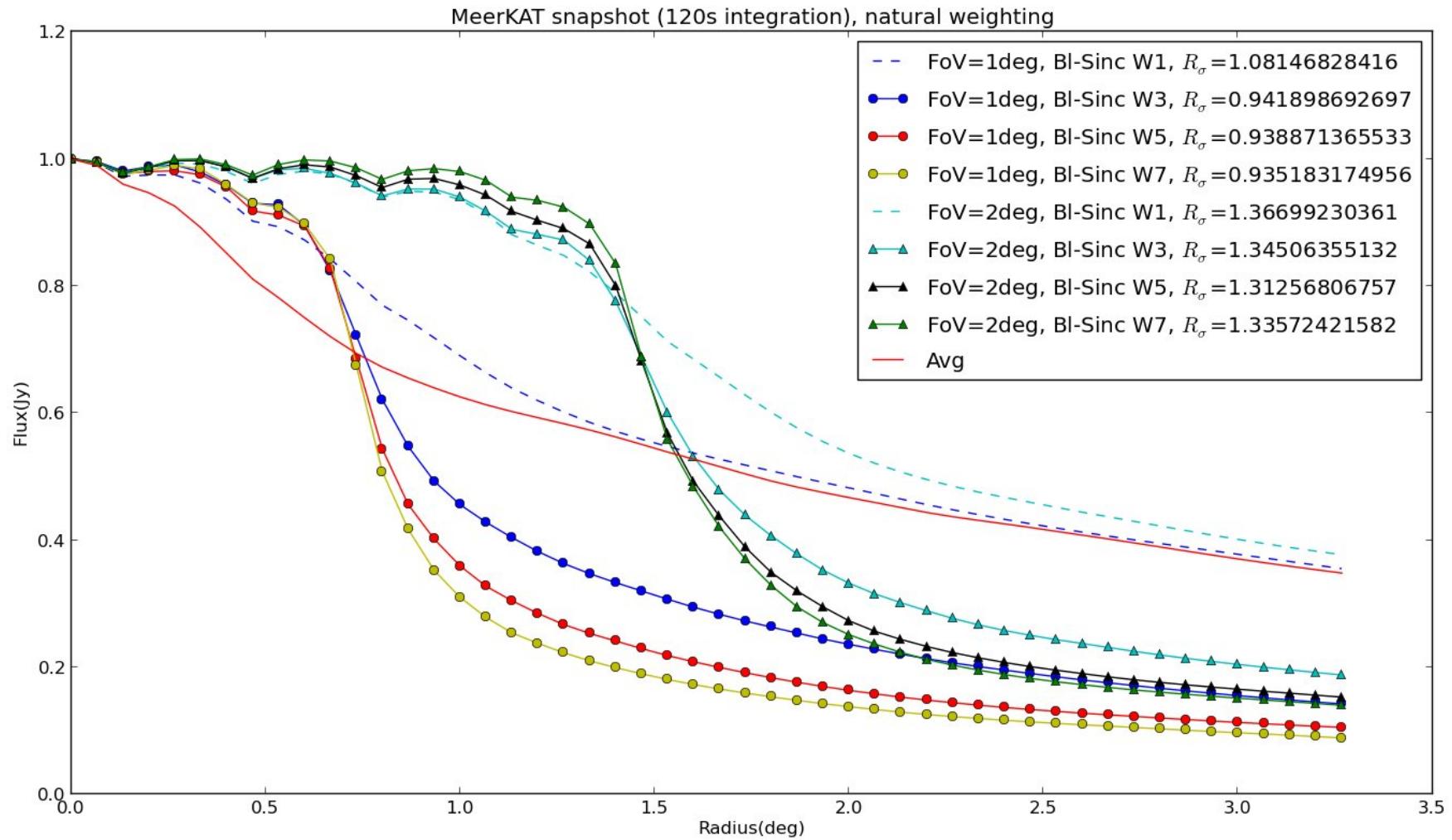
Windowing Functions

- The smearing response $R(l,m) = F^{-1}(B)$ falls off too rapidly/not rapidly enough
- So, why not play with B to optimize smearing response?
 - Replace it with some tunable windowing function W
 - Bread-and-butter filter theory
- The trade-off: increased thermal noise
 - But Stefan tells us we won't reach it anyway...
 - ...So maybe worth it across the field?

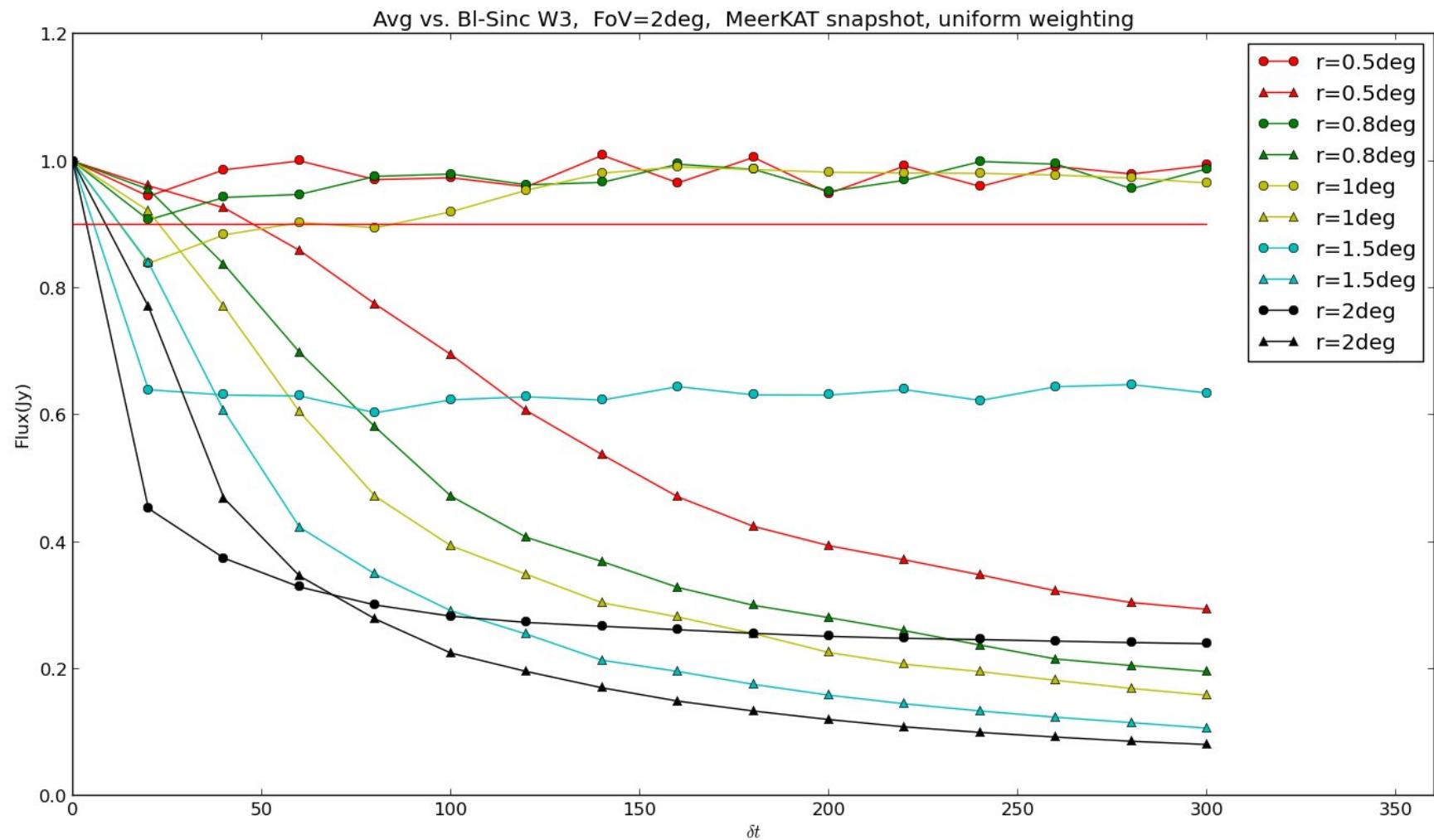
Early Results



Results: Sinc Filter

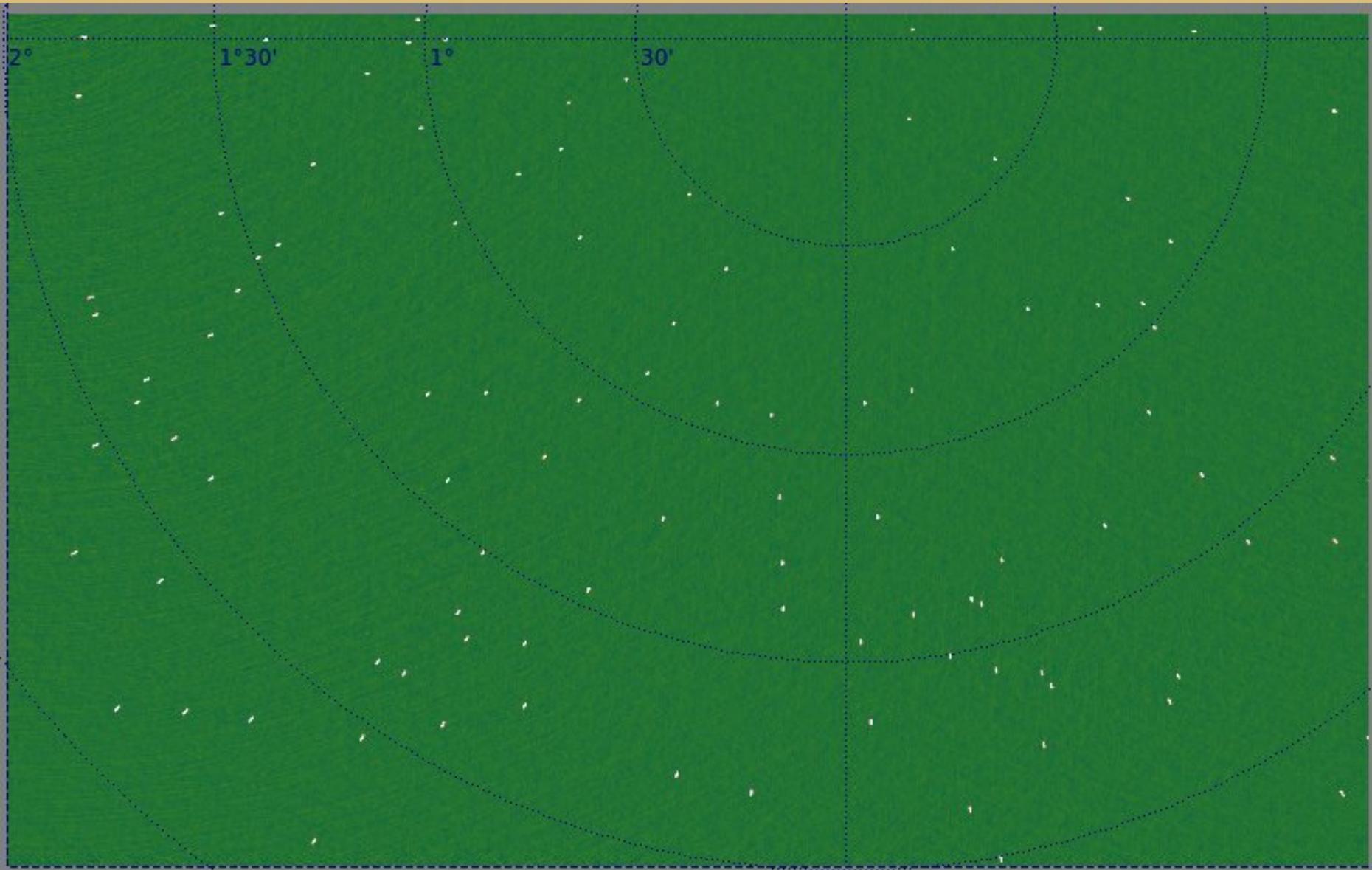


Increasing Integration Time



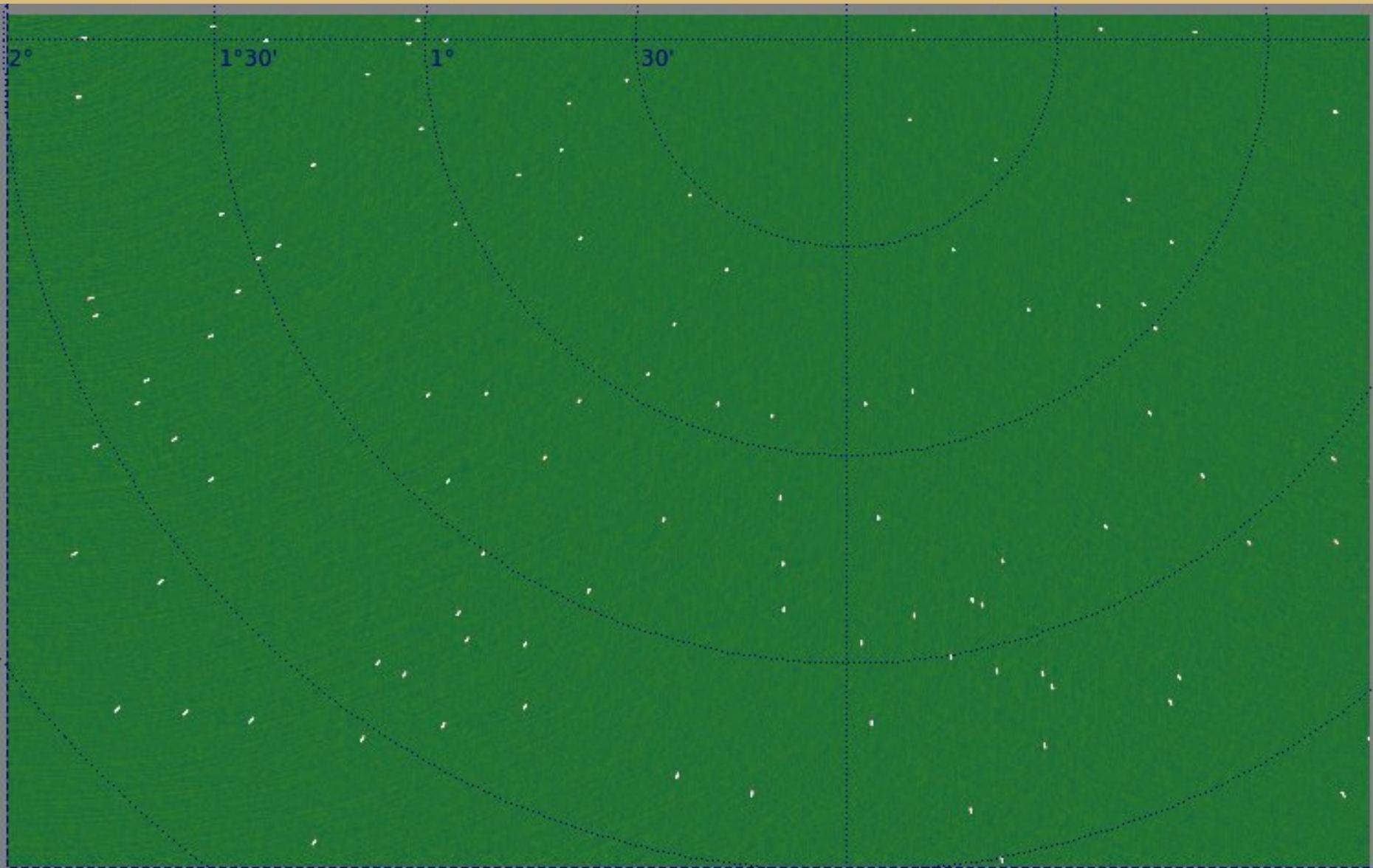
Normal Averaging

VLA-C, 5s 10MHz



Normal Averaging

VLA-C, 5s 10MHz



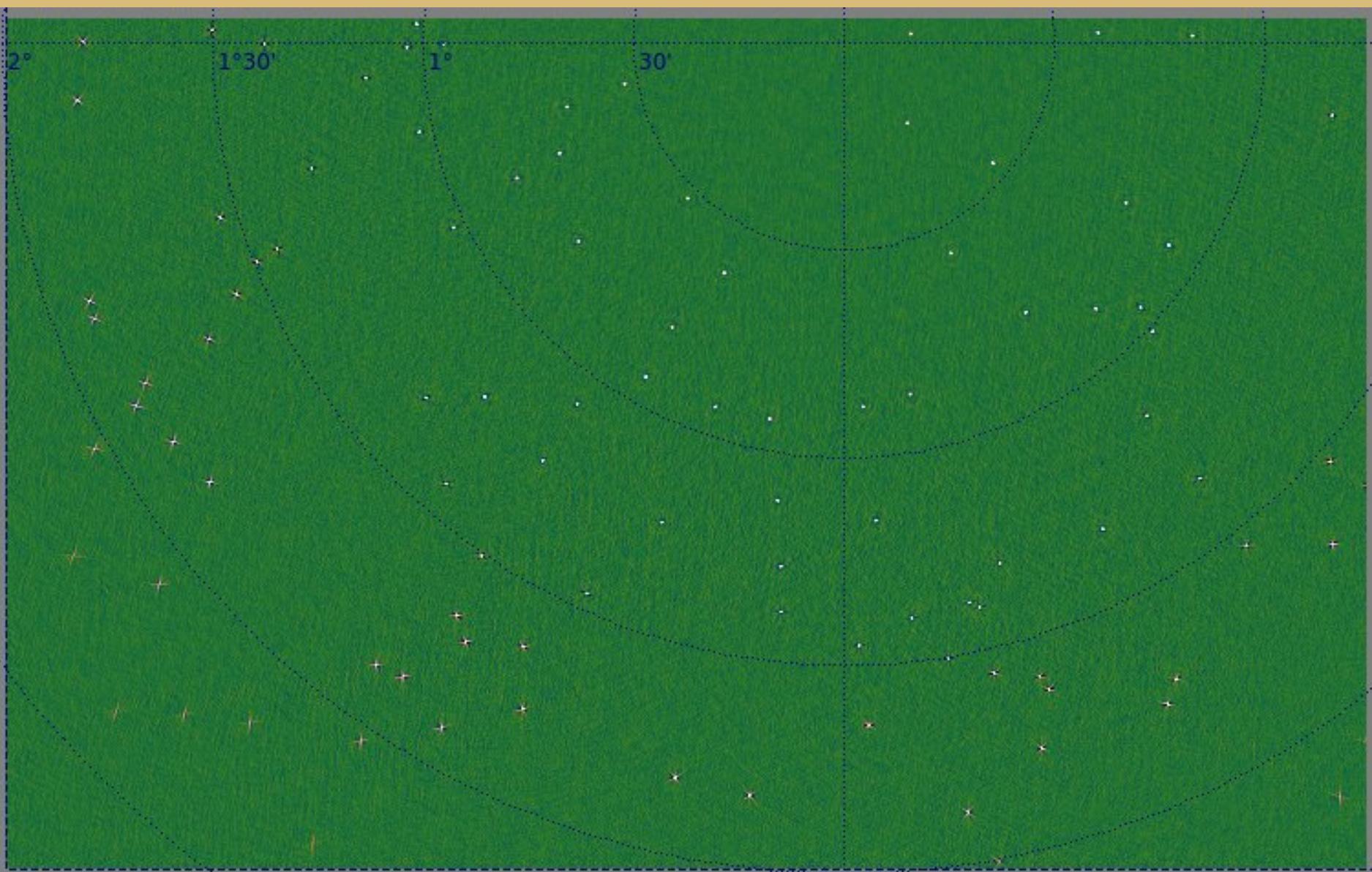
Sinc filter, FoV=2

VLA-C, 40s 50MHz (x40 compression)



Sinc filter, FoV=3

VLA-C, 40s 50MHz (x40 comporession)



Part IIa: Deconvolution

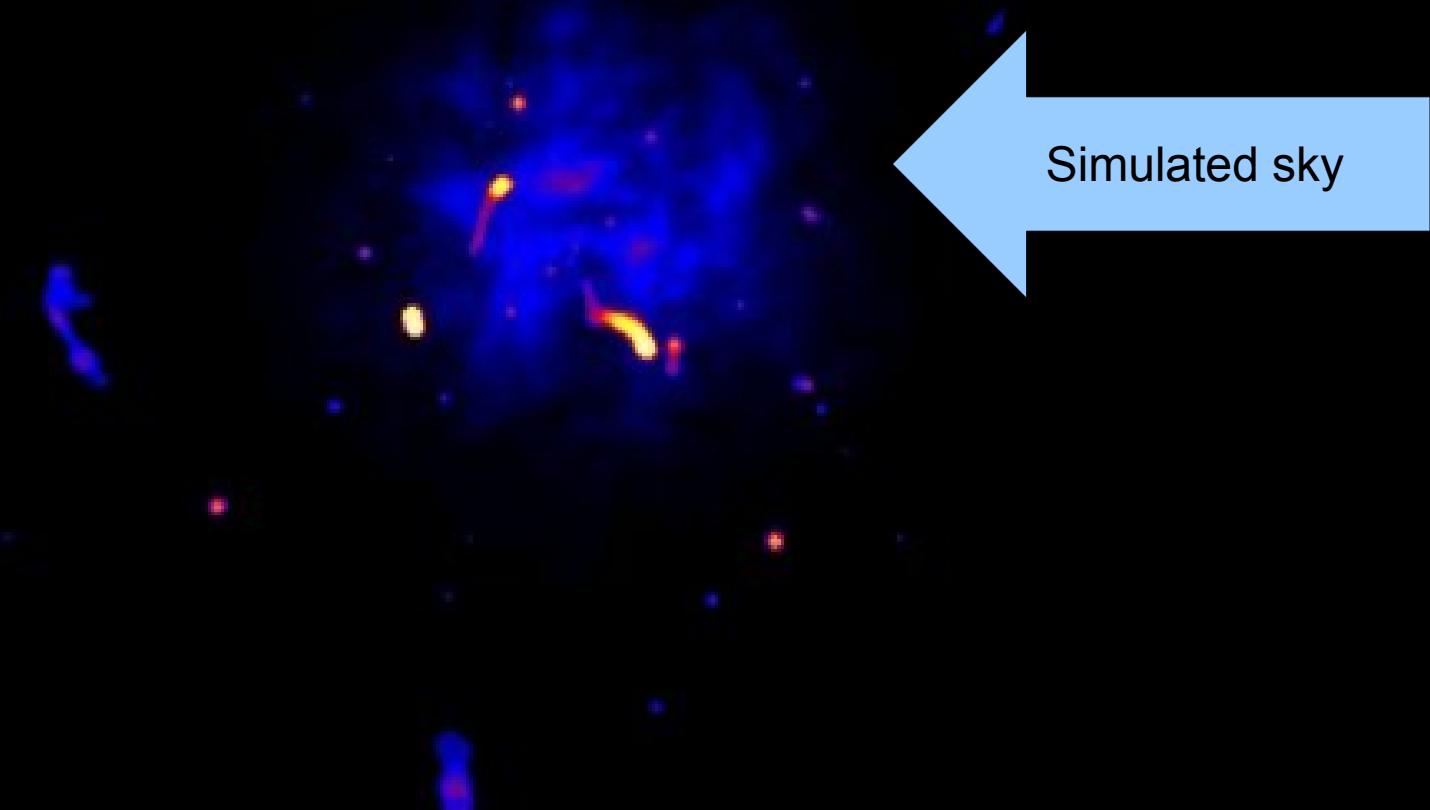
- Imaging is the ultimate data compression...

Better Deconvolution

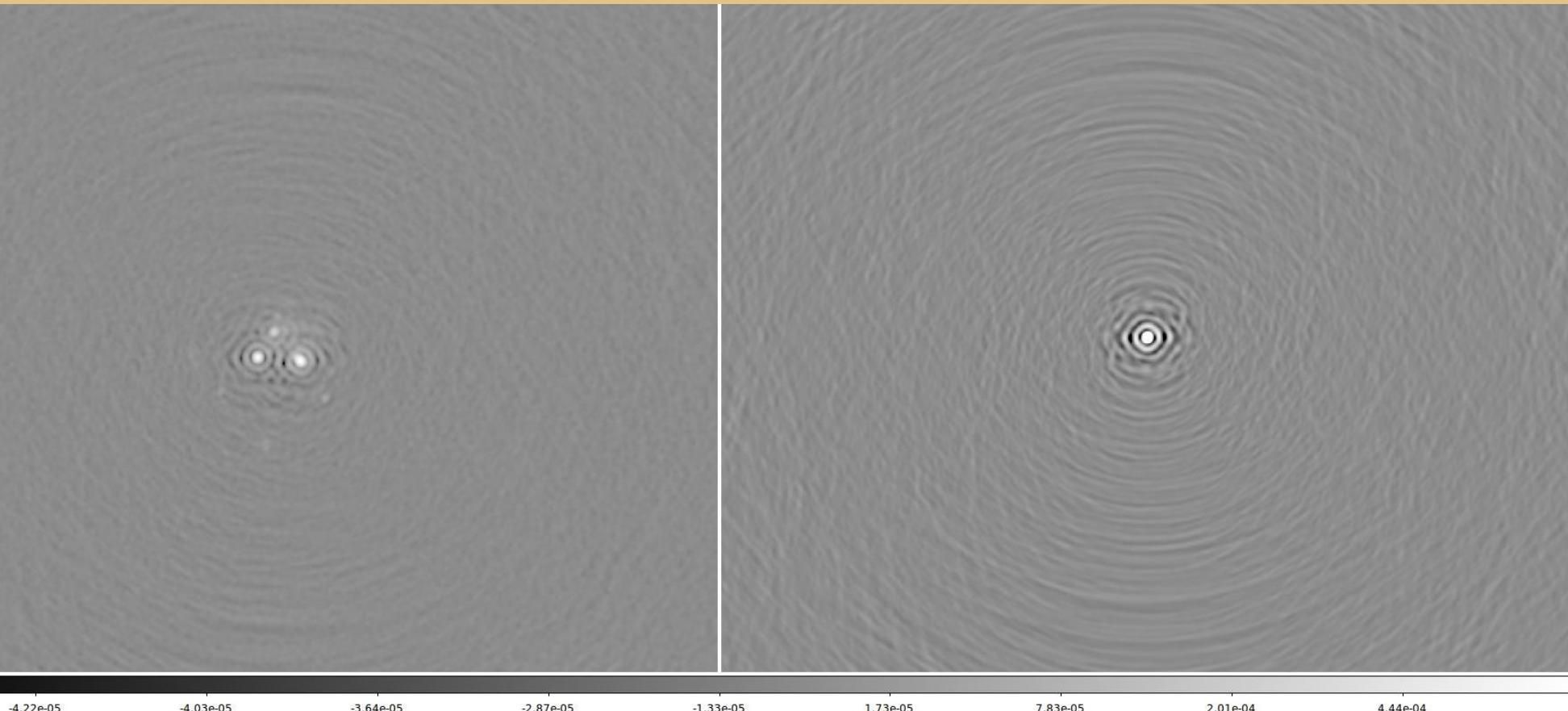
- New deconvolution algorithms coming of age
- Compressive sensing
 - SARA/PURIFY (Carrillo, Wiaux, McEwen)
 - “More Sane” (see next slide):
 - Image-plane only method, thus eliminates gridding/degridding cycle
- Bayesian image recovery
 - Sutter et al.
 - RESOLVE (Junklewitz 2013)

More Sane Deconvolution

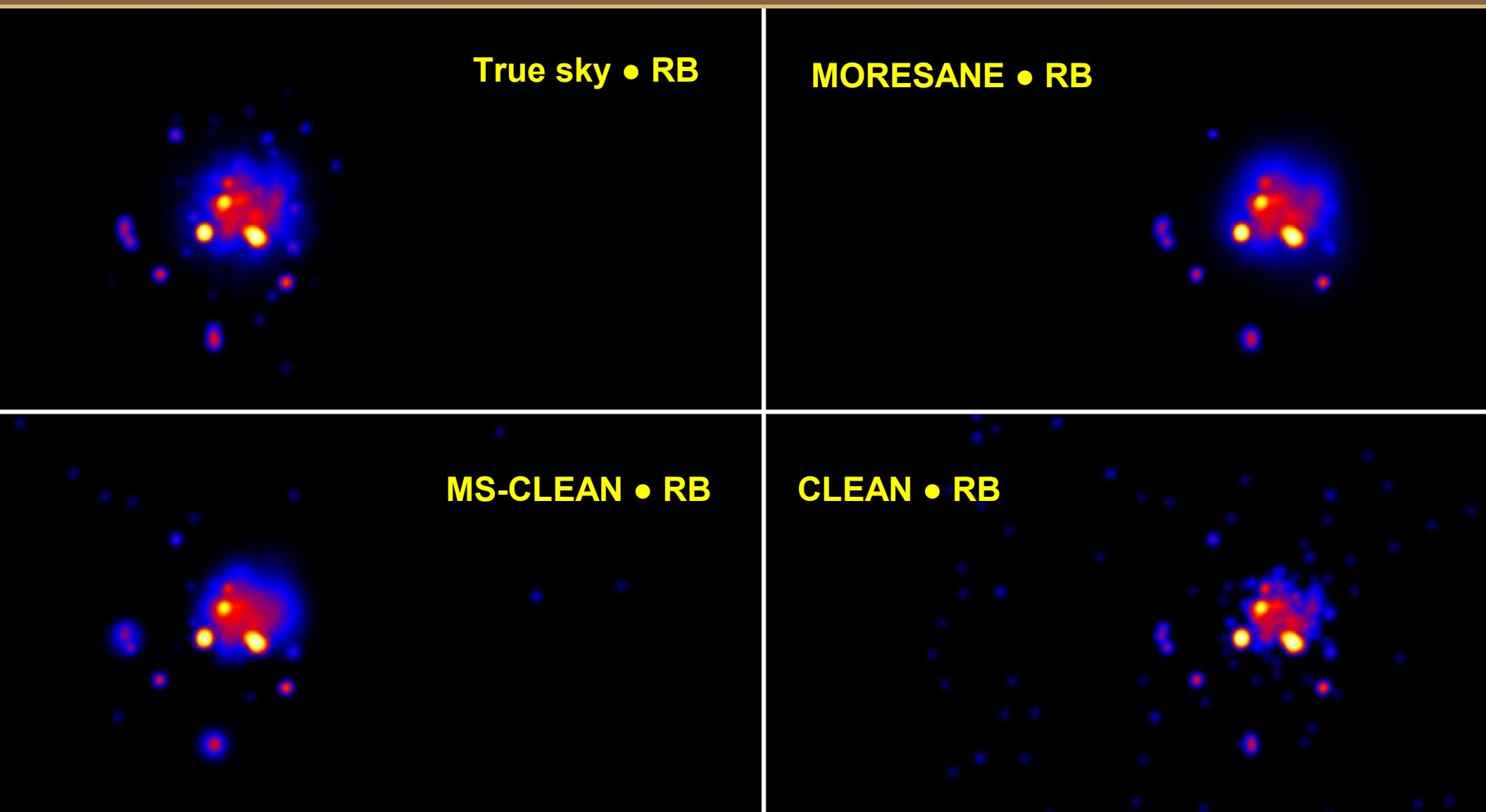
(Courtesy A. Dabbech & C. Ferrari)



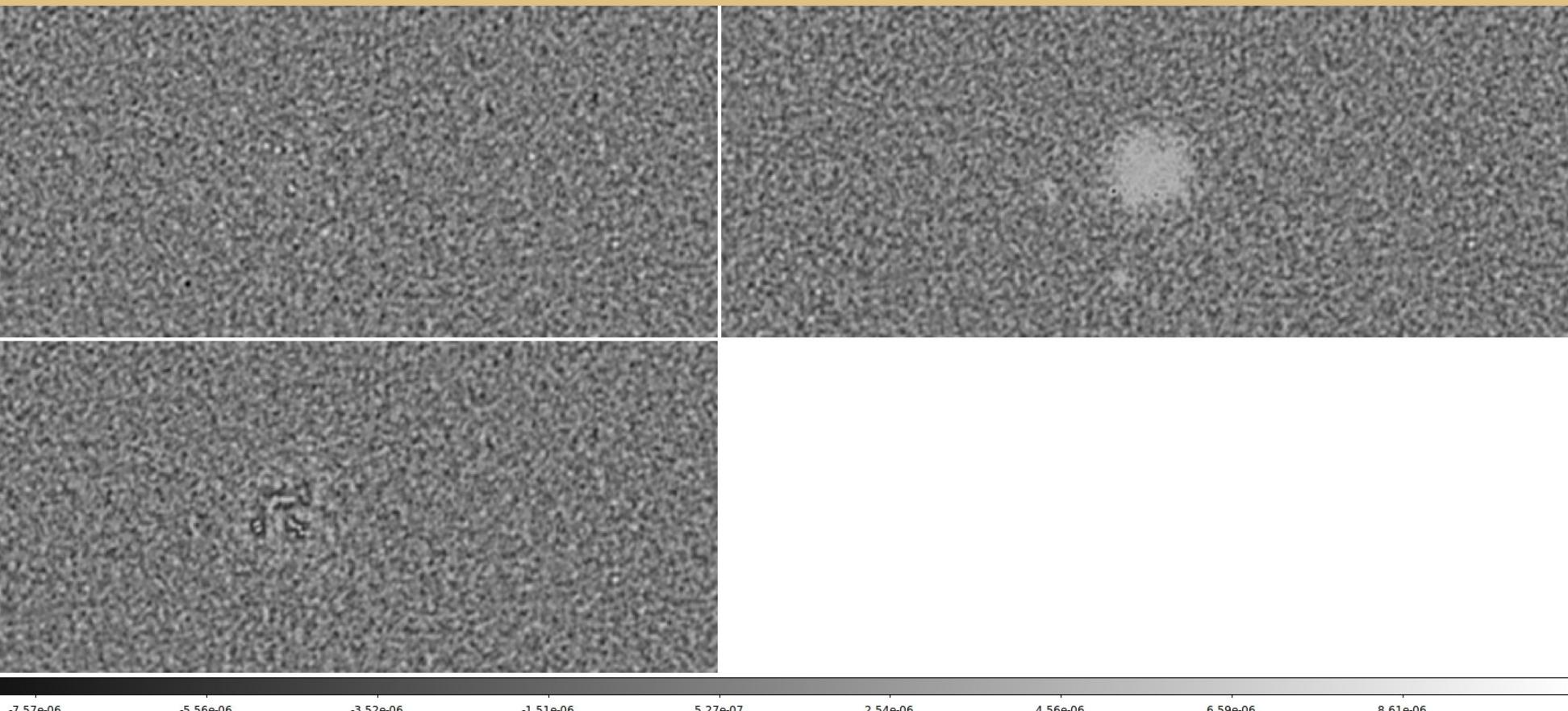
MeerKAT Dirty Image



Deconvolved Images



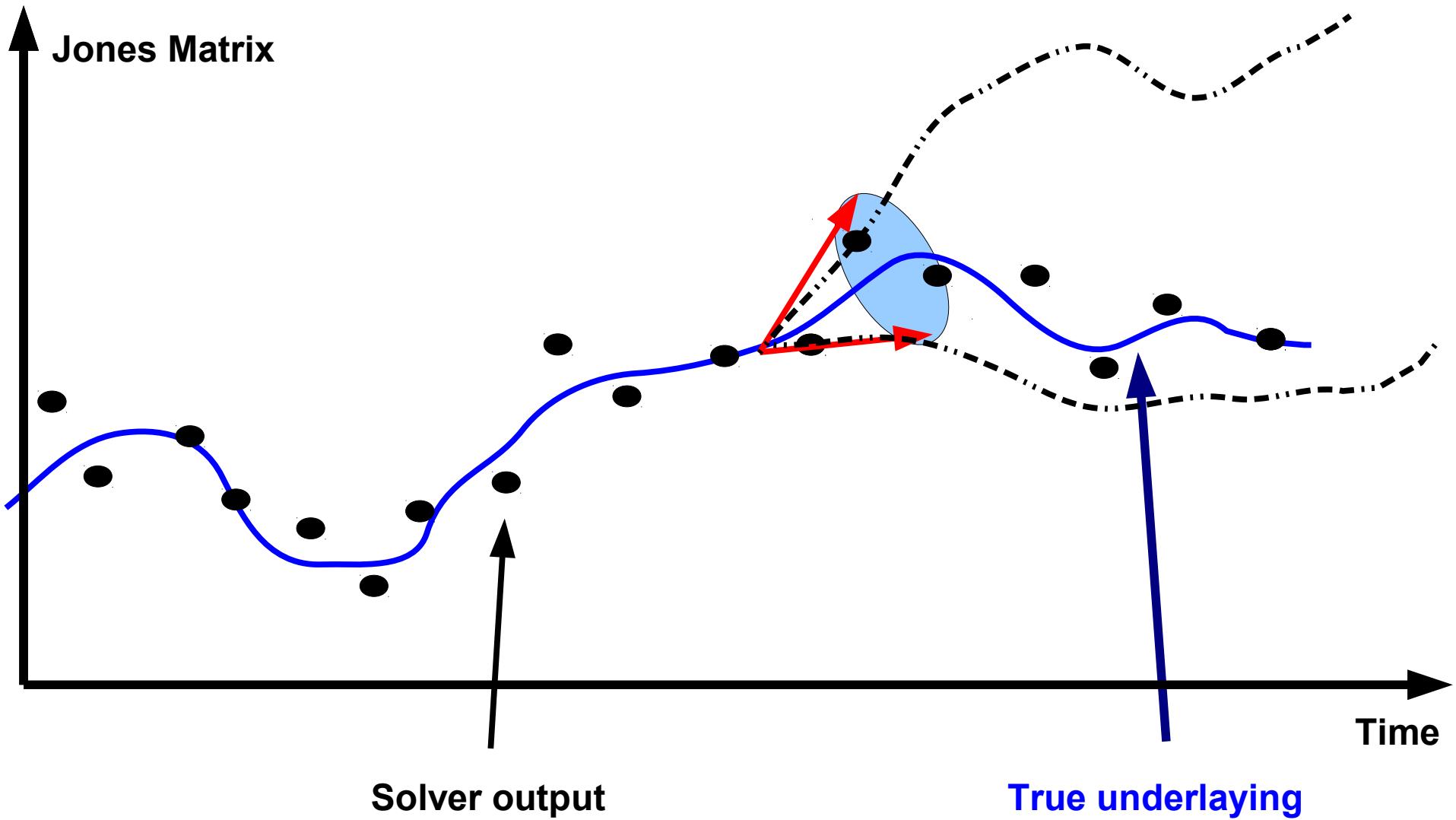
Residual Images



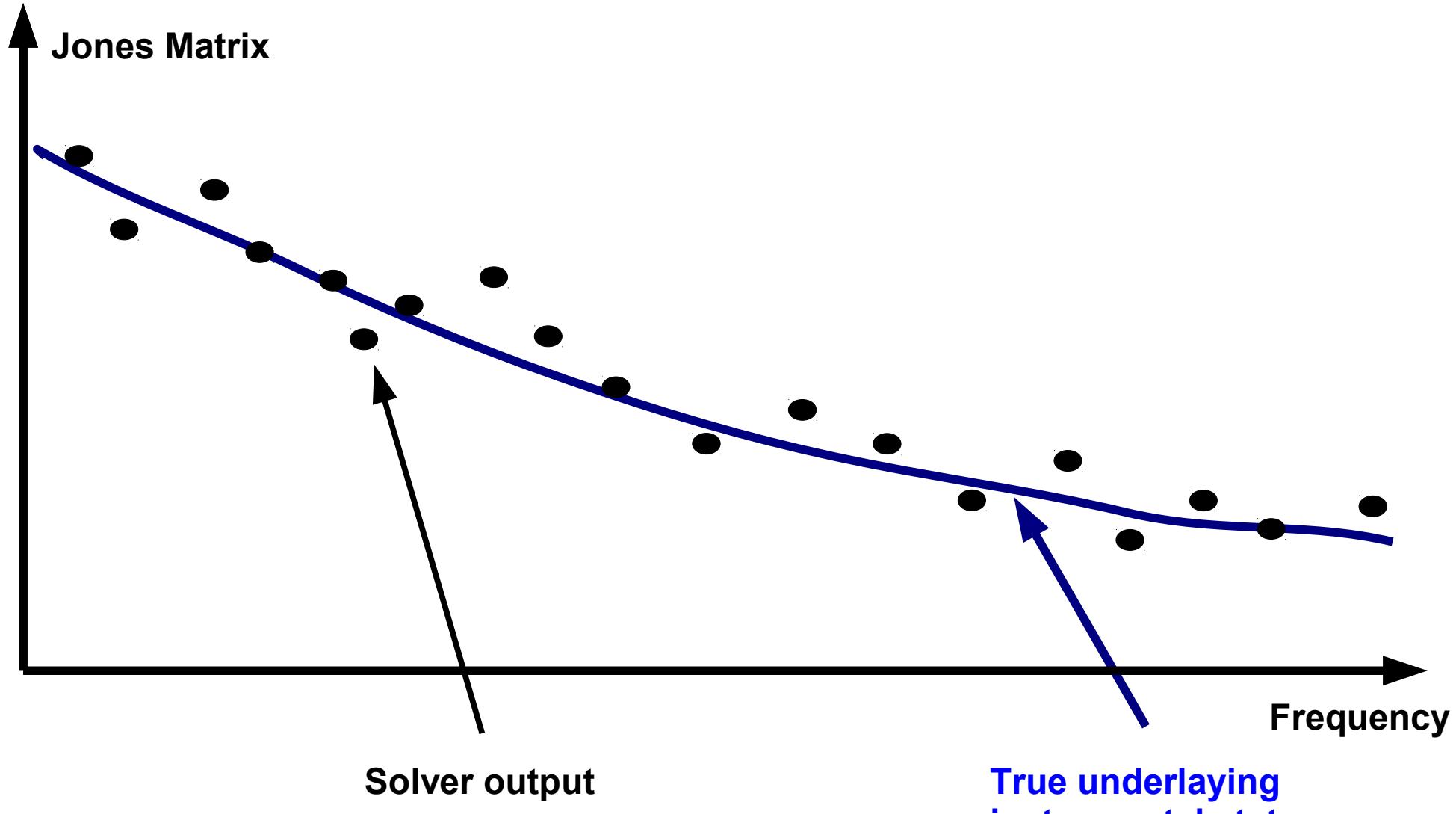
Part III: Filters

- Compression assumes calibration
- Online (streaming) calibration?
 - Stefcal fast enough
 - ...for regular gain calibration
 - Unclear how to solve for slowly-evolving parameters
- Filters vs. Solvers
 - (Courtesy of C. Tasse)

Regularity in the process



Regularity in the process



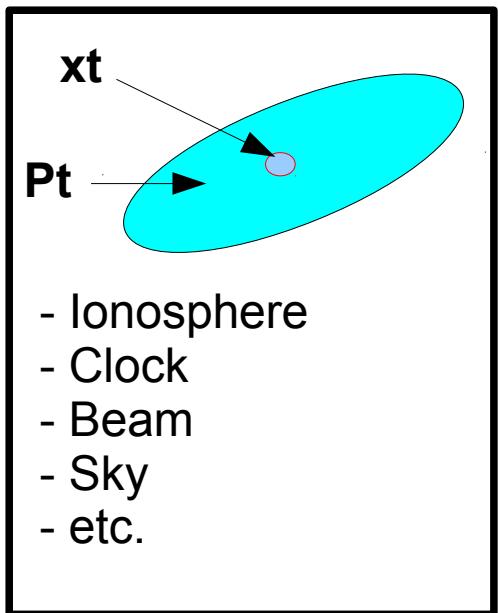
Solvers Vs. Filters

- Solver: finds the maximum likelihood solution
 - e.g. over a range of time slots
 - iterative
- Filter
 - “Solution” is process state + process variance-covariance matrix
 - ...updated at each time step using “new” data
 - Recursive (fixed cost at each step) & embarrassingly parallel
- Both approaches allow for arbitrary “physical” parameterization:
 - But the filter remains single-step

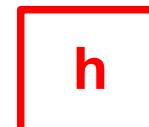
Non-linear Kalman Filters....

Process domain:

Dim=10²-10⁴

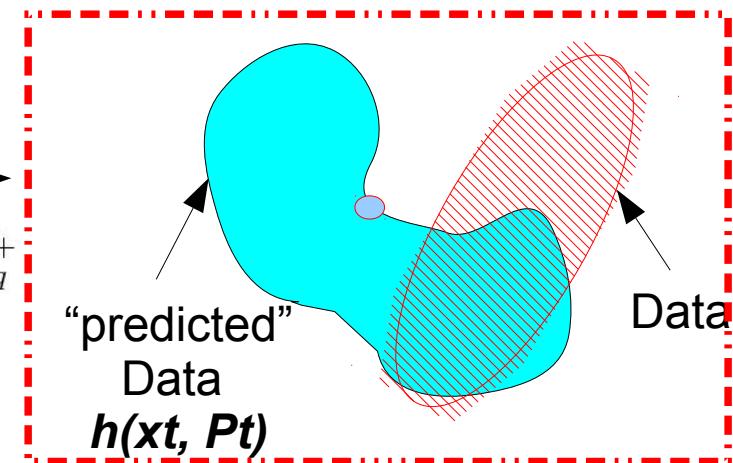


$$V_{pq} = G_p \left(\sum_{i=1}^N B_{pi} K_{pi} I_{pi} F_i \cdot F_i^+ I_{qi}^+ K_{qi}^+ B_{qi}^+ \right) G_q^+$$



Data domain:

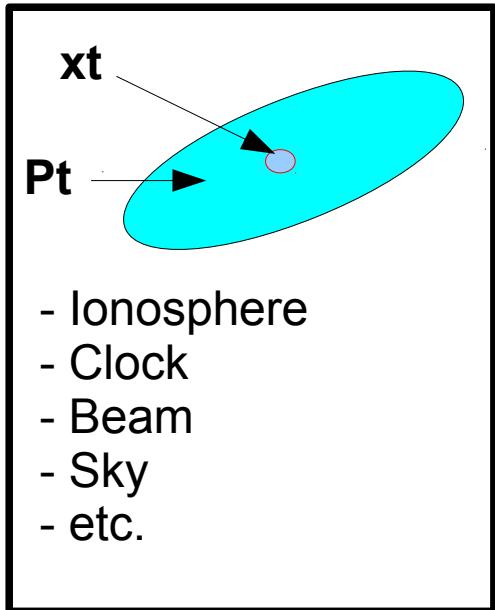
Dim=10⁴-10⁶



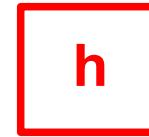
Non-linear Kalman Filters....

Process domain:

Dim=10²-10⁴

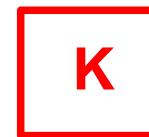
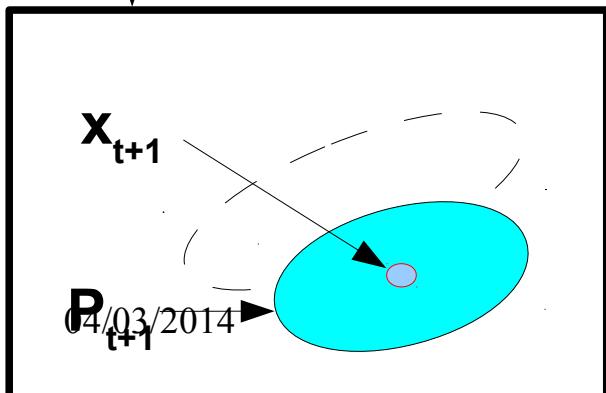
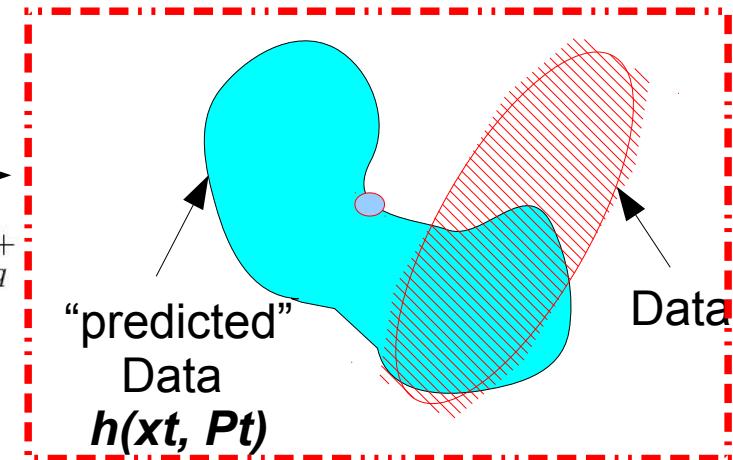


$$V_{pq} = G_p \left(\sum_{i=1}^N B_{pi} K_{pi} I_{pi} F_i \cdot F_i^+ I_{qi}^+ K_{qi}^+ B_{qi}^+ \right) G_q^+$$



Data domain:

Dim=10⁴-10⁶



$$S_k = H_k P_{k|k-1} H_k^T + R_k$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1}$$

$$\tilde{y}_k = y_k - H_k \hat{x}_{k|k-1}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

Example

- Left: phase calibration with BBS
- Right: fitting an analytic clock offset with a Kalman filter



Example 2: Ionospheric Simulation

