RFI excision using self-calibration

Tim Cornwell, Rick Perley, Kumar Golap, Sanjay Bhatnagar

Overview

- Why post-correlation?
- RFI closure
- A partitioning algorithm
- Demonstration using VLA 333MHz observations
- Implications for EVLA and SKA

Why post-correlation?

- Many advances in the performance of synthesis radio telescopes have come from post-correlation methods
  - Deconvolution and self-calibration
- Avoids disturbing system pre-correlation by *e.g.* nulling
  - Important for *e.g.* high dynamic range full field imaging

Why no reference horn?

- Not necessary…
- Array is an excellent reference horn!
- Could mimic reference horn by adding noise to one antenna!

RFI closure

- Necessary:
  - RFI from point sources enters to all antennas via sidelobes
  - Sampling in time and frequency is fine enough to avoid decorrelation

$$\Delta t < \frac{10^4}{\sqrt{SNR}} \frac{\lambda}{B \cos \delta} \quad \text{seconds}$$

- Helpful:
  - RFI occurs at known fringe rate (fixed with respect to the earth)
- Violated if:
  - *e.g.* Multiple internal birdies

VLA Case

- For the VLA, with SNR = 100, we find, in **milliseconds:**

Config.	90cm	20cm	6cm	2cm	0.7cm
E	3860	860	260	85	30
D	960	210	65	20	7.5
C	300	70	20	6.8	2.4
B	95	20	6.5	2.2	.75
A	30	6.8	2.0	.70	.25
NMA	3.0	.70	.20	.070	.025

- These are very short times, leading to very large databases.
  - For the EVLA, each channel will produce ~ 60 KB/sec at 100 msec integration. The total exceeds 1 GB/sec.
  - The red zone lies beyond the WIDAR correlation – but natural fringe winding provides 25 dB attenuation in 1 second!

Math

Measurement equation

$$V_{ij}^{\text{obs}} = g_i g_j^* V^{\text{source}} + a_i a_j^* k_i k_j^* P$$

Gain solution

$$S = \sum_{ij} w_{ij} \left| V_{ij}^{\text{obs}} - g_i g_j^* V^{\text{model}} - a_i a_j^* k_i k_j^* \right|^2$$

Gain application

$$V_{ij}^{\text{cal}} = \left(g_i g_j^*\right)^{-1} \left(V_{ij}^{\text{obs}} - a_i a_j^* k_i k_j^* P\right)$$

Antenna gain

Antenna sidelobe gain

RFI power

Propagation term

Algorithm

1. Initialize on and off axis gains $\begin{aligned} g_i &= 1 \\ a_i &= 0 \end{aligned}$

2. Calibrate using current estimates of on and off axis gains

$$V_{ij}^{\mathrm{cal}} = \left(g_i g_j^*\right)^{-1}\left(V^{\mathrm{obs}} - a_i a_j^* e^{j\omega_e t}\right)$$

3. Make Clean model from V_{ij}^{cal}

4. Stop if Clean image is satisfactory

5. Predict model visibilities V_{ij}^{model}

6. Solve for gains g_i, a_i by minimizing

$$S = \sum_{\upsilon t}\sum_{ij} w_{ij}\left|V_{ij}^{\mathrm{obs}} - g_i g_j^* V^{\mathrm{model}} - a_i a_j^* e^{j\omega_e t}\right|^2$$

7. Return to step 2

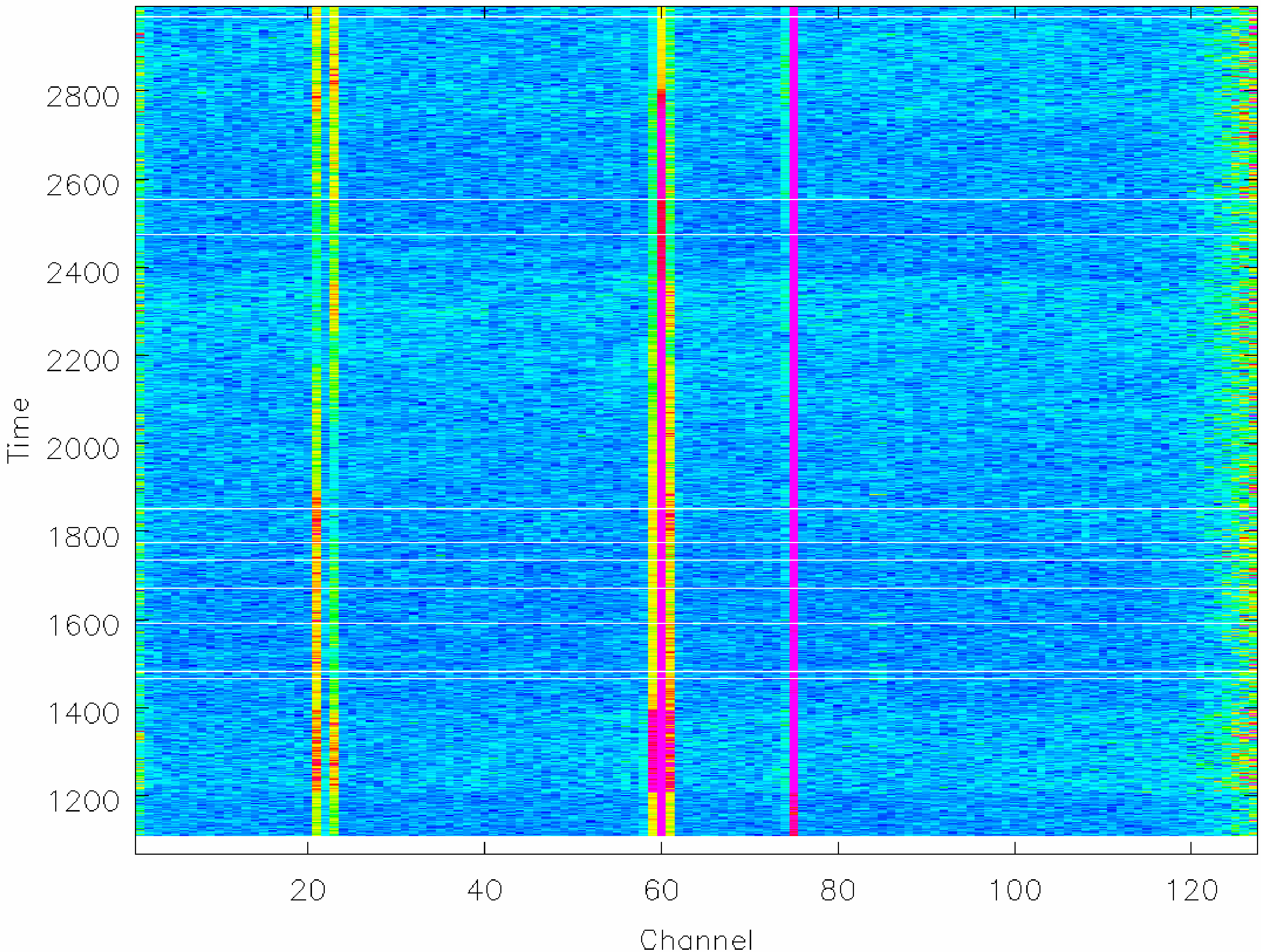
Implementation as AIPS++ script

a. Make two copies of MeasurementSet, one for the target *(Mt)* and one for the interference *(Mi).*
b. Initialize interference source model to point source at the pole.
c. Predict model visibilities for *Mt* and *Mi* :
d. Mt:
    i. Solve for off-axis gains using antenna bandpass solution, B, in calibrater.
    ii. Apply off-axis gains to model visibility (contains Fourier transform of the target) to obtain predicted observed target visibility
e. Mi:
    i. Solve for on-axis gains using antenna gain solution, G, in calibrater.
    ii. Apply on-axis gains to model visibility (contains Fourier transform of the interference) to obtain predicted observed interference
f. Cross subtract:
    i. *Mt:* Subtract predicted o bserved interference visibilities to obtain estimate of observed visibilities in absence of interference
    ii. *Mi:* Subtract predicted observed target visibilities to obtain estimate of observed interference visibilities in absence of target
g. Update estimates of on axis gains and correct *Mt*.
h. Update model of target by clean deconvolution (or similar)
i. Stop if converged, else repeat from step c onwards.

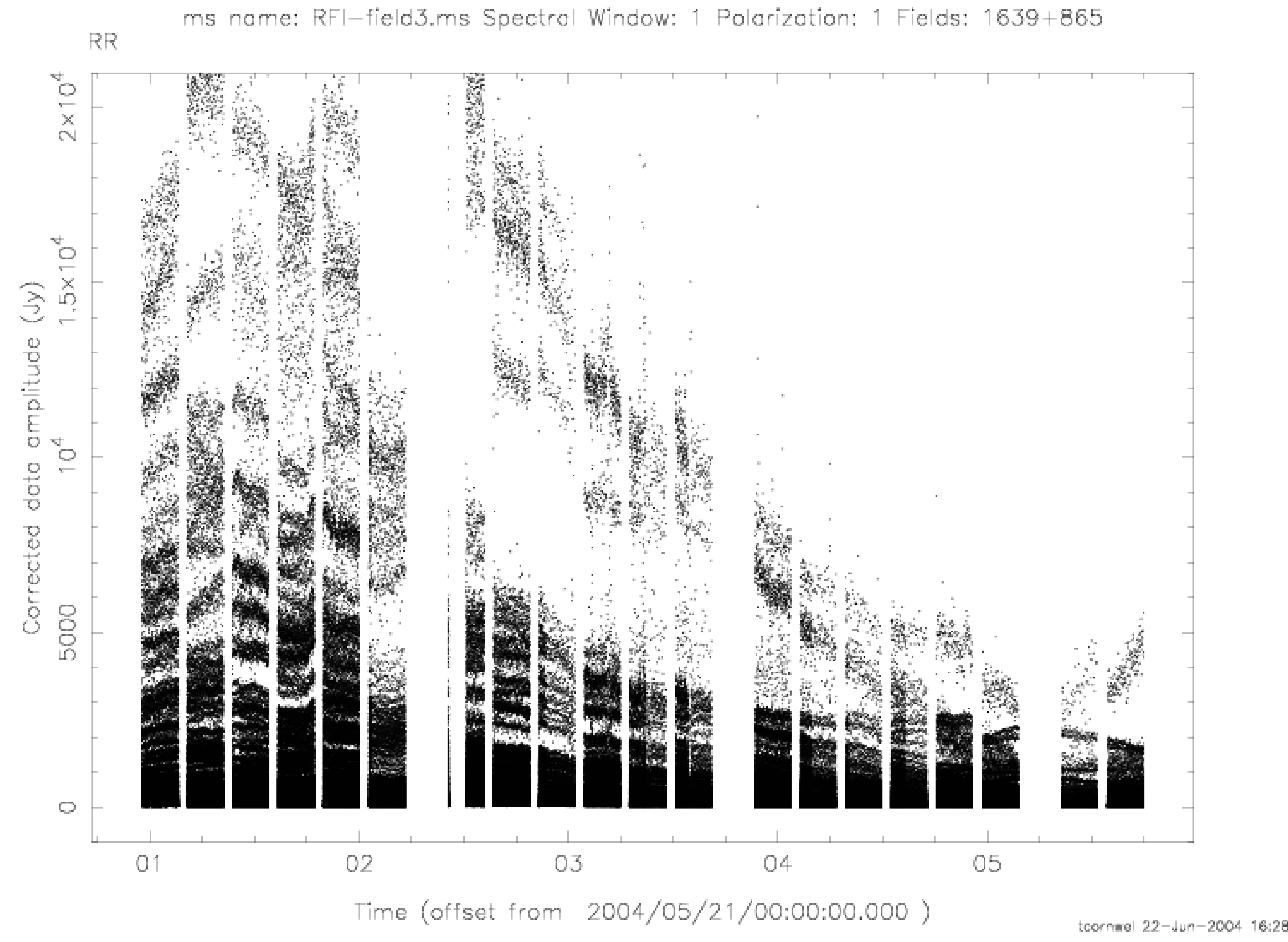
VLA 333MHz test observations

- To allow adequate sampling with current correlator, look at source close to North pole
  - Very bad case
- At 333MHz - strong line due to Albuquerque Airport radar

Configuration	D (up to 700m) with North arm in C (up to 2km)
Source	NGC6251 (declination ~ +86deg)
Observing date and time	2004May21, 00:44UT-05:46UT
Integration time	3.3s
Channelization	3.1MHz total bandwidth, 127 channels for channel width of 24.4kHz
Polarization	RR



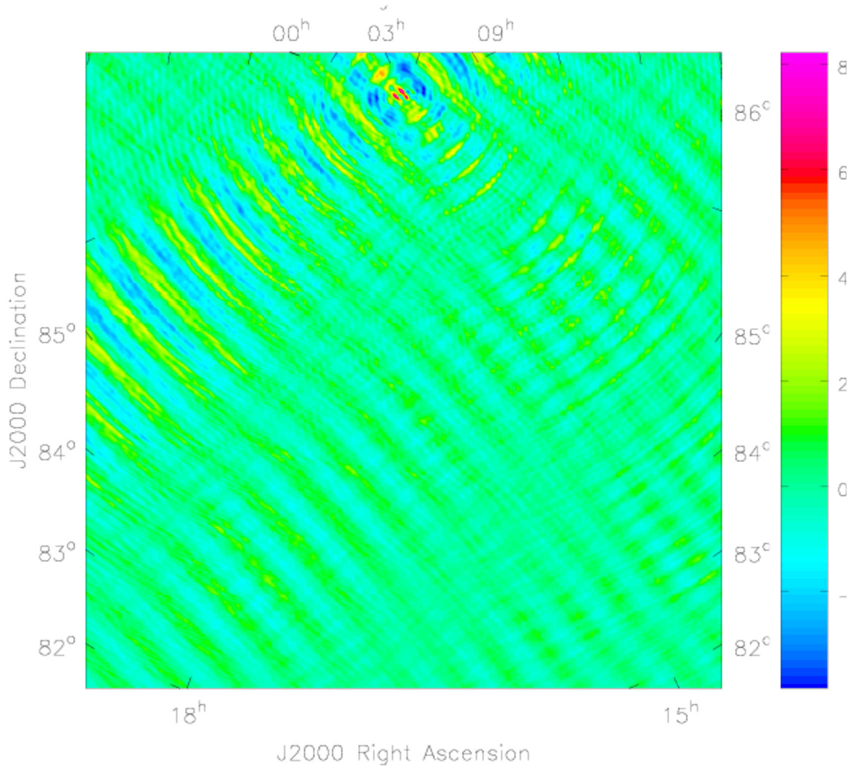
2800
2600
2400
2200
2000
1800
1600
1400
1200
Time
20
40
60
80
100
120
Channel

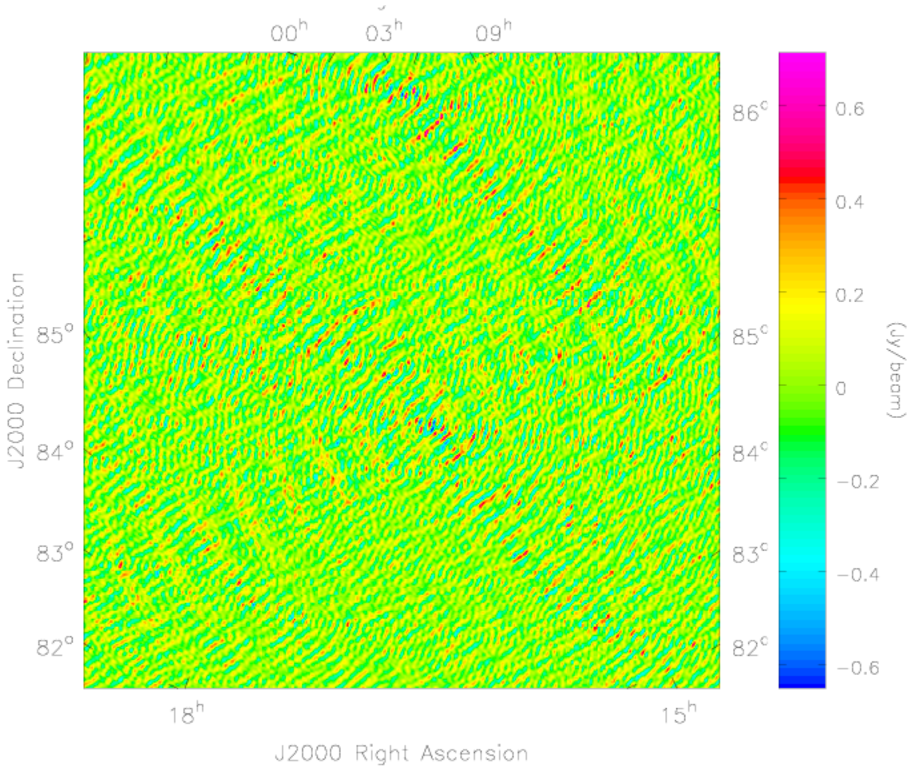


ms name: RFI-field3.ms Spectral Window: 1 Polarization: 1 Fields: 1639+865
RR
Corrected data amplitude (Jy)
0
5000
10^4
1.5x10^4
2x10^4
01
02
03
04
05
Time (offset from 2004/05/21/00:00:00.000)
tcornwel 22-Jun-2004 16:28

Channel 60 fails

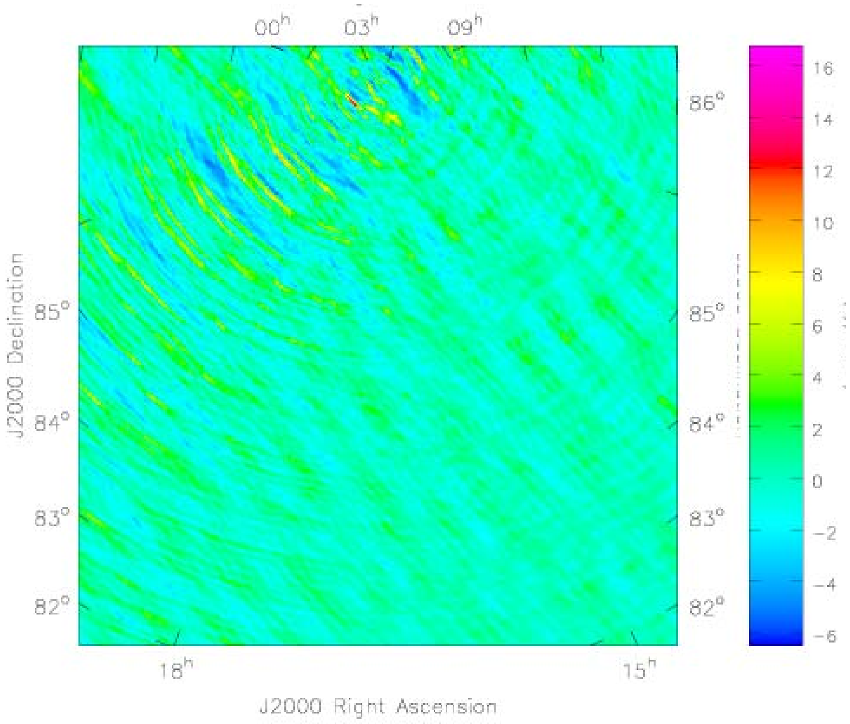
- 20dB reduction is not enough

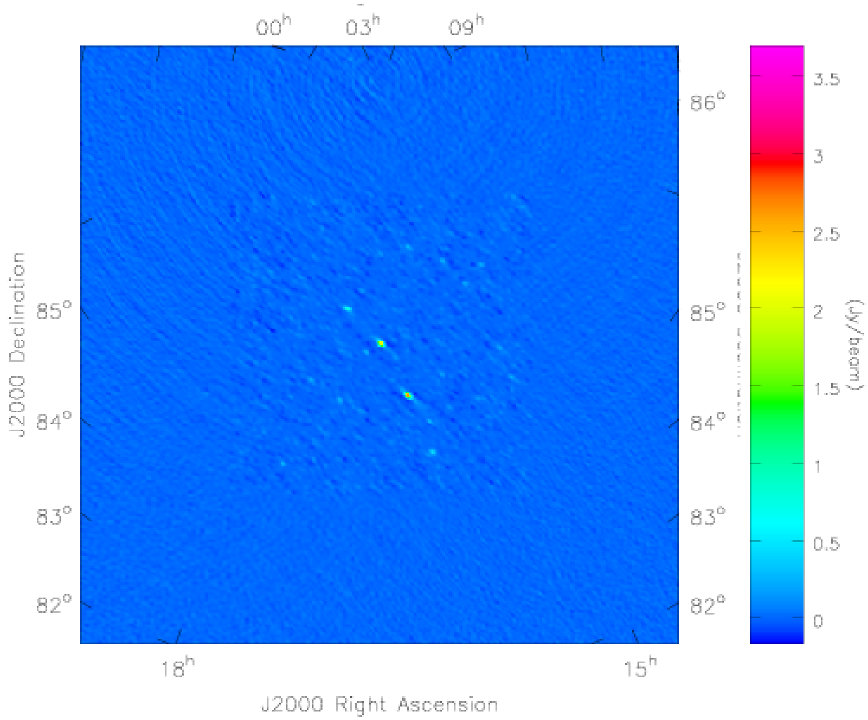




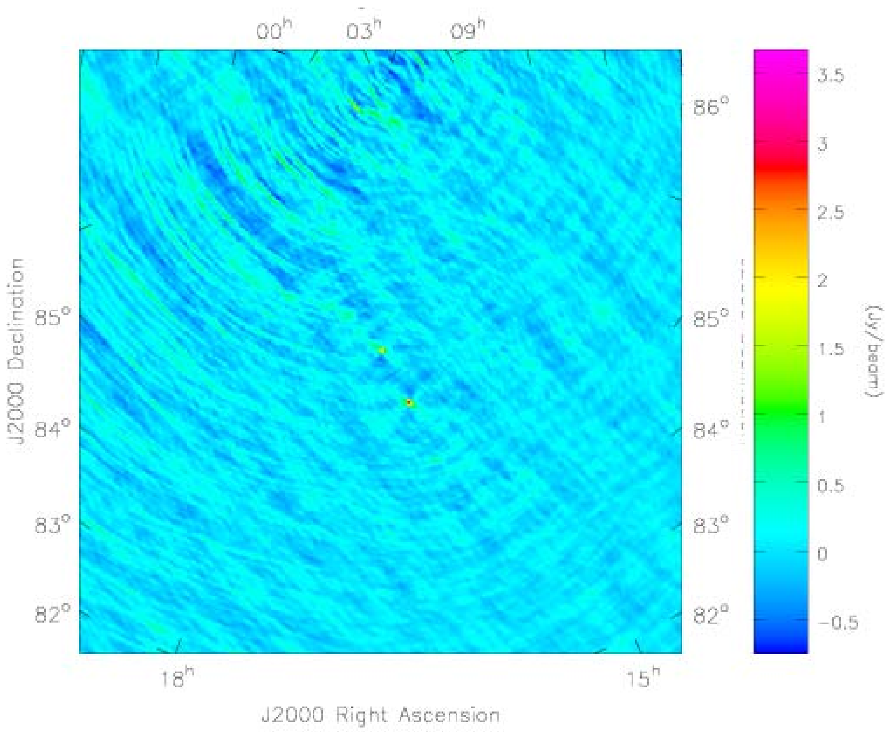
Channel 75 works

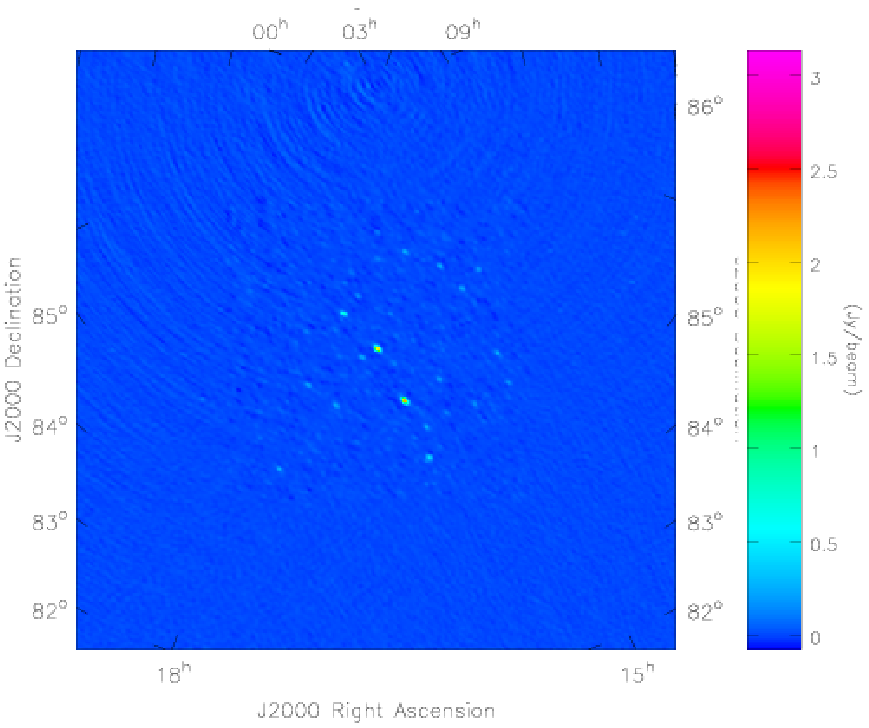
- RFI ~ 7dB lower



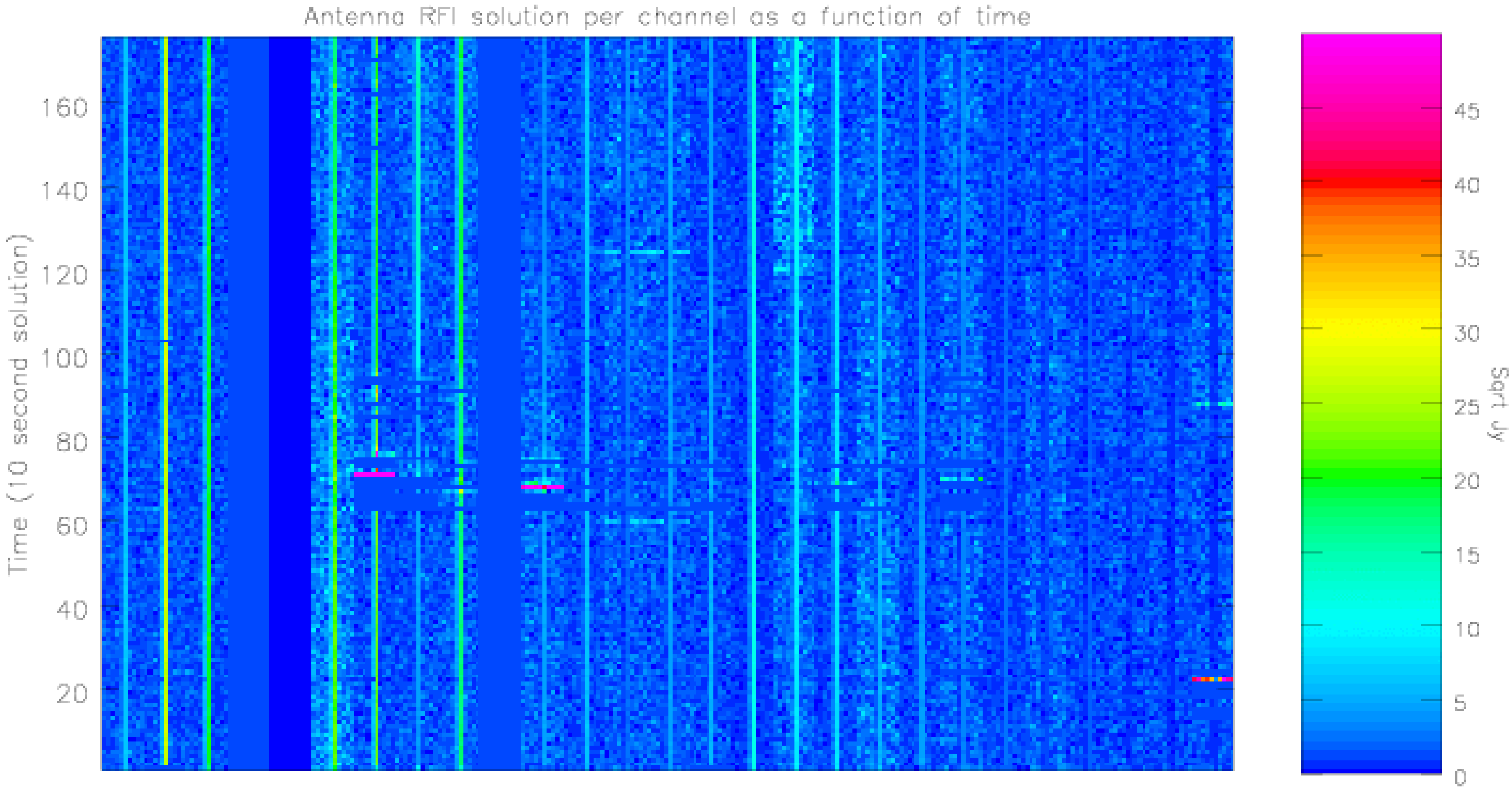


Channels 70 - 80 are even better





Solutions for RFI per antenna/channel/time



Summary of test

- RFI reduced by factor ~ 20 - 30 dB
- Works on single channel
- Better if some channels are clean

Can we afford this algorithm?

- Part of normal selfcalibration loop
- Except that we must sample to image entire horizon or sky
- Data rate expands by factor $\left(\frac{D}{\lambda}\right)^2$
- Can be done with hardware for wide-field imaging if:

$$\left(\frac{D}{\lambda}\right)^2 \leq \left(\frac{B_{\mathrm{Imaging}}}{\mathrm{B_{RFI}}}\right)^3$$

Implications

- Will work for dense, weak RFI
- Lessens need for station-beam nulling
  - Really unattractive for wide-field imaging
- Currently we design systems for small field closure
  - Now we need to design for all sky closure

Future improvements

- Threshold antenna gains to avoid processing noise
- Improved solvers
  - Eigensystem approach may be faster
- Use first order model for decorrelation due to time and frequency averaging
- Fringe search for moving objects
  - *e.g.* DME, satellites
- Excellent method to identify RFI
  - Could simply flag if number of channels small

Summary

- Post-correlation excision *without a reference horn* works well
- Have developed efficient partitioning algorithm
  - Obtain Measurement Set containing RFI only
- Processing requirements may drive SKA computing