Wirtinger Kalman Overdrive

O. Smirnov
(Rhodes & SKA SA)

C. Tasse
(Obs. Paris Meudon & Rhodes)

The 3GC Culture Wars

- Two approaches to dealing with DD effects
- The “NRAO School”:
  - Represent everything by the A-term
  - Correct during imaging (convolutional gridding)
  - Solve for pointing offsets
  - Sky models are images
- The “ASTRON School”:
  - Solve for DD gains towards (clusters of) sources
  - Make component sky models, subtract sources in *uv*-plane while accounting for DD gains

Why DD Gains

- Cons: non-physical, slow & expensive
- But, DD+MeqTrees have consistently delivered the goods with all major pathfinders
  - Early LOFAR maps and LOFAR EoR (S. Yatawatta)
  - Beautiful ASKAP/BETA maps (I. Heywood)
  - JVLA 5M+ DR (M. Mitra earlier)
- Fair bet that we'll still be using them come MeerKAT and SKA

DD Gains Are Like Whiskey

- The smoother the better
- Make everything look more attractive
- If you overindulge, you wake in in the morning wondering where your {polarized foregrounds, weak sources, science signal} have gotten to

$$V_{pq} = \overbrace{G_p}^{\text{gain \& bandpass}} \underbrace{\left(\sum_s \overbrace{\Delta E_p^{(s)}}^{\text{differential gain}} \overbrace{E_p^{(s)}}^{\text{beam}} \overbrace{X_{pq}}^{\text{source coherency}} E_q^{(s)H} \Delta E_q^{(s)H} \right)}_{\text{sum over sources}} G_q^H$$

The One True Way

The Middle Way

- DR limited by how well we can subtract the brighter source population
  - thus bigger problem for small dish/WF
- Subtract the first two-three orders of magnitude in the *uv*-plane
  - good source modelling and (deconvolution and/or Bayesian)
  - PB models, pointing solutions, **+solvable DD gains**
- Image and deconvolve the rest really well
  - A-term and/or faceting

Why Expensive

- Solving for Jones matrices is a non-linear optimization problem
- O((Nant x Ndir)^3)
- Need faster (and <u>simpler</u>) algorithms
  - GPU: often better off with many simple ops over fewer complicated ops

$$V_{pq} = \overbrace{G_p}^{\text{gain \& bandpass}} \underbrace{\left(\sum_s \overbrace{\Delta E_p^{(s)}}^{\text{differential gain}} \overbrace{E_p^{(s)}}^{\text{beam}} \overbrace{X_{pq}}^{\text{source coherency}} E_q^{(s)H} \Delta E_q^{(s)H} \right)}_{\text{sum over sources}} G_q^H$$

Non-linear Optimization

- Minimizing residuals $\{r_j\} = \{d_j - f(p_1 \ldots p_N)\}$
- Most algorithms rely on taking a derivative and heading down the gradient

Parameter update vector [Nparm]

Jacobian (gradient) $\{\partial f_j / \partial p_i\}$ Ndata x Nparm

$$\Delta \mathbf{p} = (\mathbf{J}^H \mathbf{J})^{-1} \mathbf{J}^H \mathbf{r}$$

Hessian

$$

Residual vector [Ndata]

Accelerating Things

- Major cost is inverting the Hessian
- Scales as N_p^3, so gets prohibitively expensive for many directions / many antennas
- Algorithms are iterative, so ***fast but approximate*** inversion can make a huge difference
  - Helps if the matrix is sparse (spars-ish)
  - Peeling is one such kludge
- Need insights into the structure of the matrix to come up with inversion approaches

Complex Derivatives

- Classical optimization theory deals with functions of a real variable
- Complex derivatives are funny things
  - Complex conjugate does not have a complex derivative

$$\frac{\partial \bar{z}}{\partial z}$$

- Traditional approach: take derivatives w.r.t. real and imaginary, then you have 2N real derivatives instead
  - complicates the equations

$$z = x + iy, \quad \frac{\partial z}{\partial x}, \quad \frac{\partial z}{\partial y}$$

Wirtinger Derivatives

- Wirtinger (1922): treat z and z conjugate as two independent variables, and formally define:

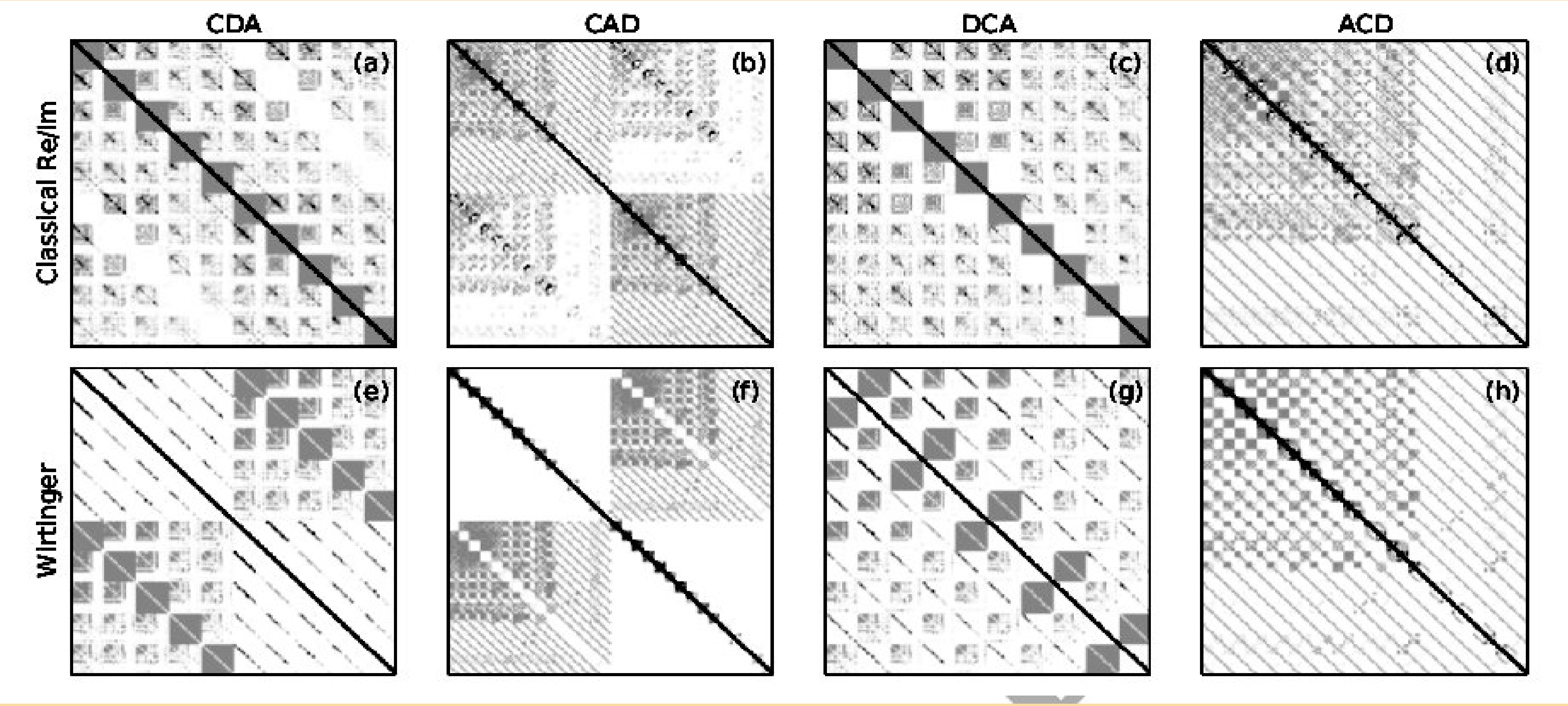
$$f(z,\bar{z}), \quad \frac{\partial f}{\partial z} = \frac{\partial f}{\partial x} - i\frac{\partial f}{\partial y}, \qquad \frac{\partial f}{\partial \bar{z}} = \frac{\partial f}{\partial x} + i\frac{\partial f}{\partial y},$$

conveniently: $\frac{\partial \bar{z}}{\partial z} = \frac{\partial f}{\partial \bar{z}} = 0$

- Purely formal definition, but allows us to define a complex gradient operator <u>that works for optimization</u>

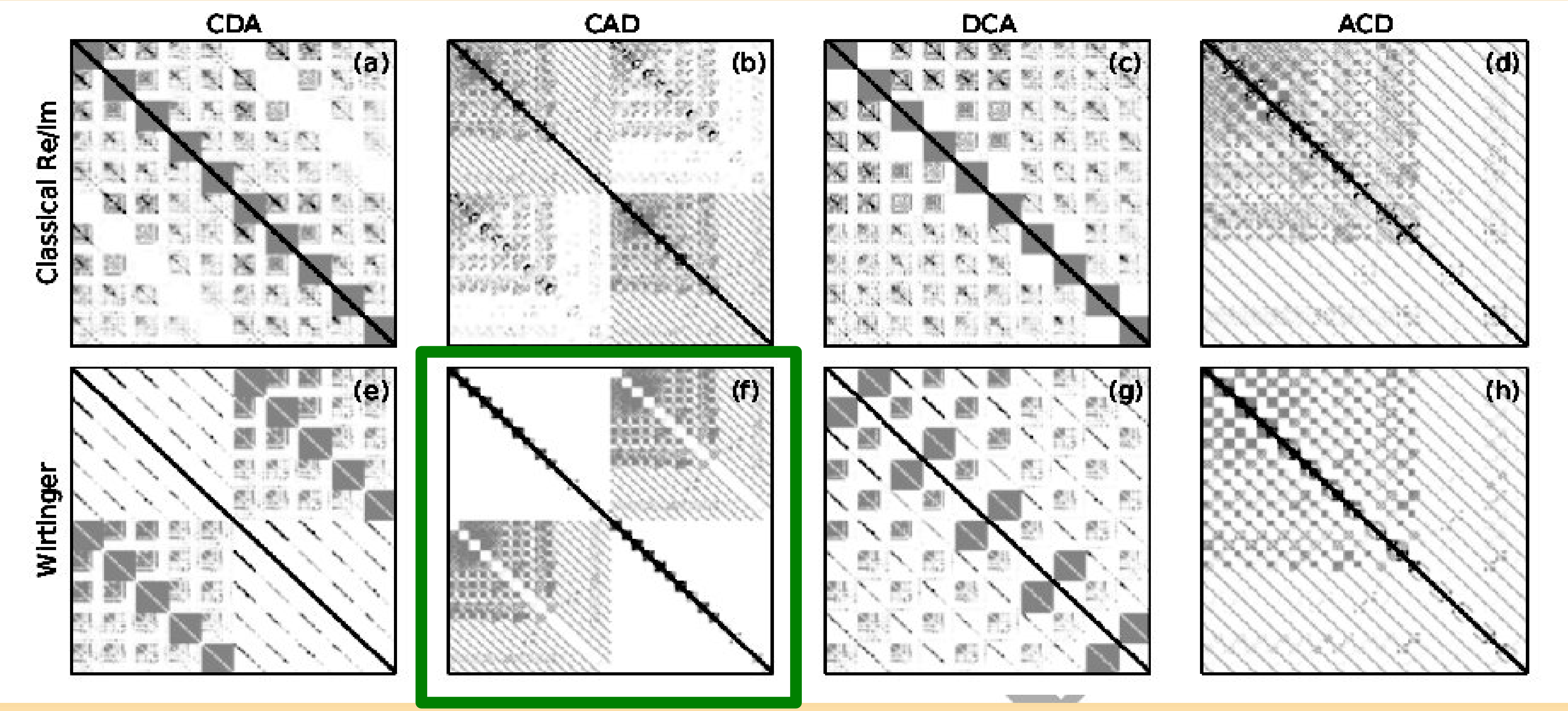
Whence Wirtinger

- Considerably simplifies the equations
- Yields new insights into the Hessian structure

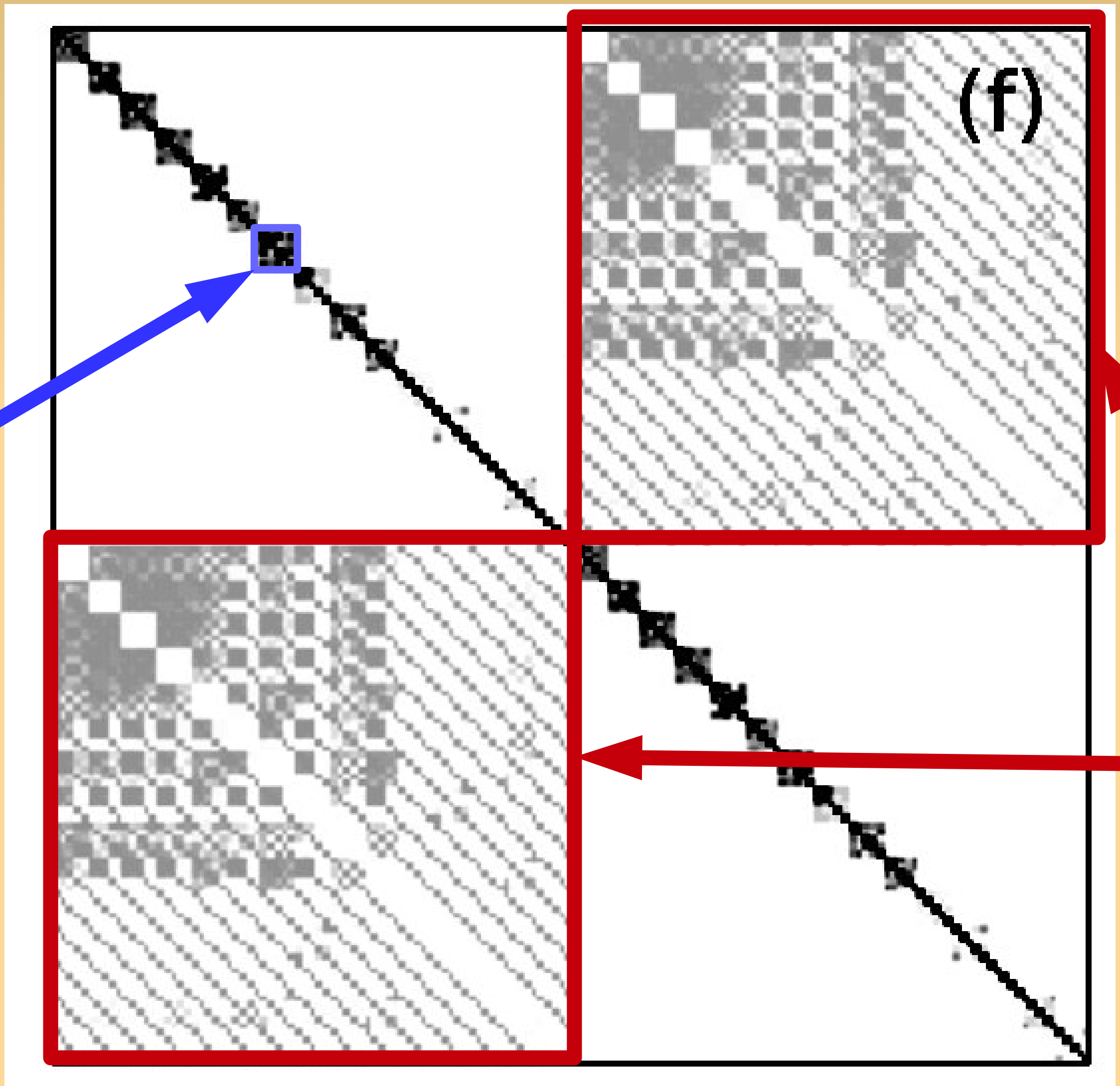


Revealing Sparsity

- Plot of amplitude of $\mathbf{J}^H\mathbf{J}$ (contrast exaggerated)
- Wirtinger style reveals sparsity



An Almost Sparse Matrix

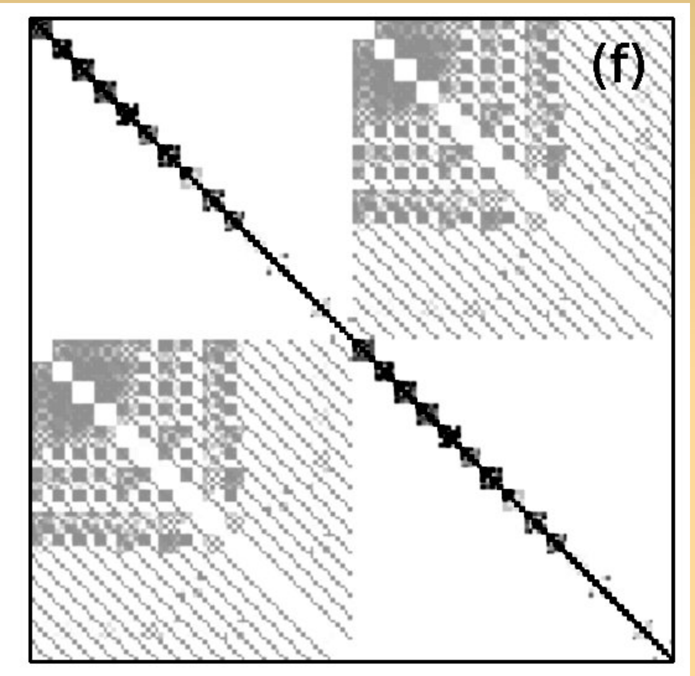


Those Blocks are (Nd x Nd)

Those blocks have a small amplitude

COHJONES

- Complex Half-Jacobian Optimization for N-directional Estimation
- Treat off-diagonal blocks as zero; diagonal blocks are block-diagonal
- Inversion scales as Ndir^3 rather than (Nant Ndir)^3
- Huge gain in performance



COHJONES = DD StefCal

- Interestingly, for Ndir=1, CohJones reduces to the StefCal algorithm
- ….which was formulated on a completely different basis:
  - Alternating direction implicit (ADI) method
- Turns bilinear equation into linear

Treat this as constant (from previous iteration)

$$\{r_{pq}\} = \{v_{pq} - g_p m_{pq} g_q^*\}$$

Treat this as solvable

A Family Of Algorithms

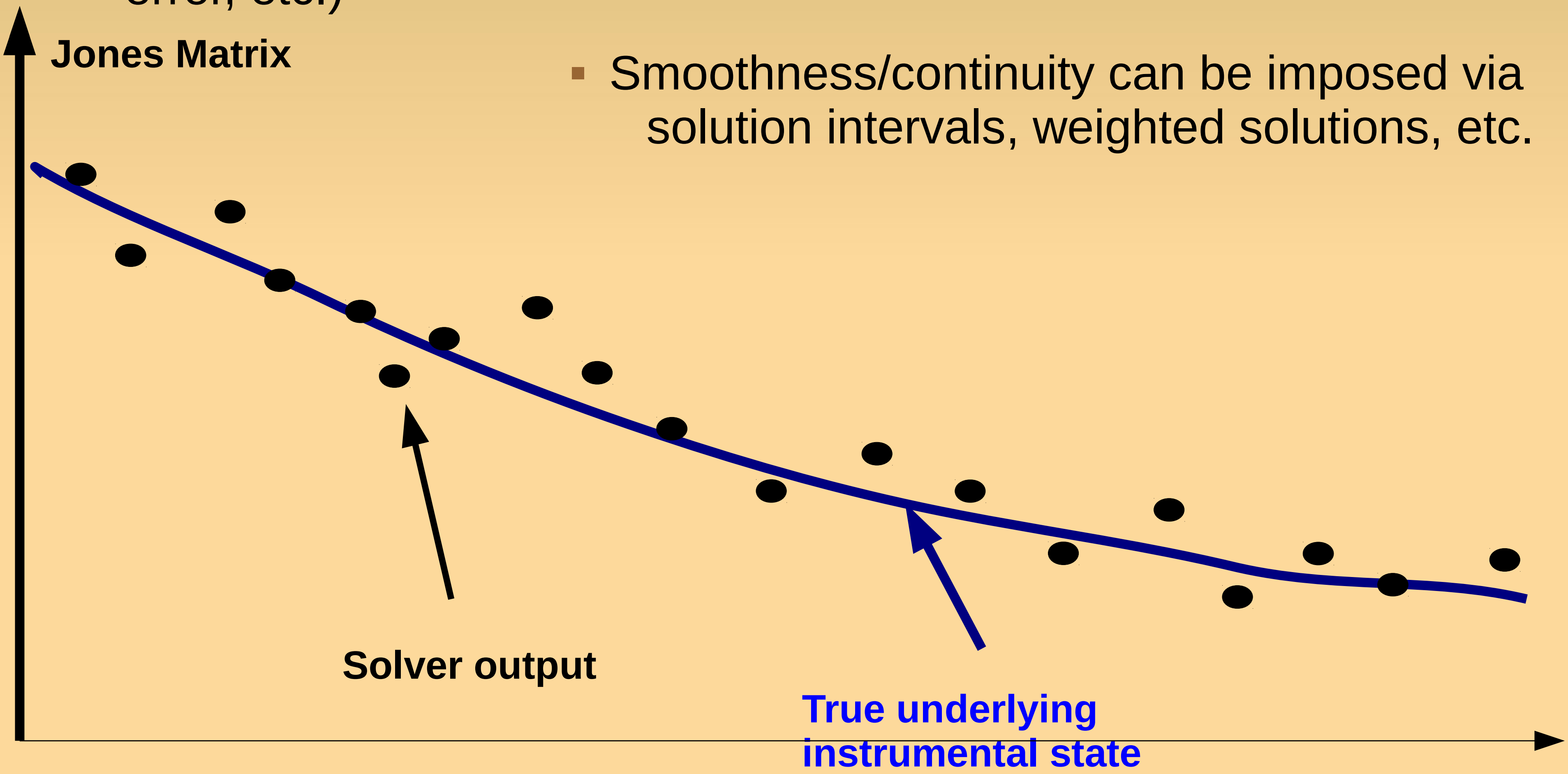
- Wirtinger calculus not limited to DD gains, can be used to simplify different calibration problems
- E.g. pointing offsets and beam shapes
- Have extended it to *Jones matrix derivatives*

$$\frac{\partial V_{pq}}{\partial G_p}, \quad \frac{\partial V_{pq}}{\partial G_q^H}$$

Filters vs. Solvers

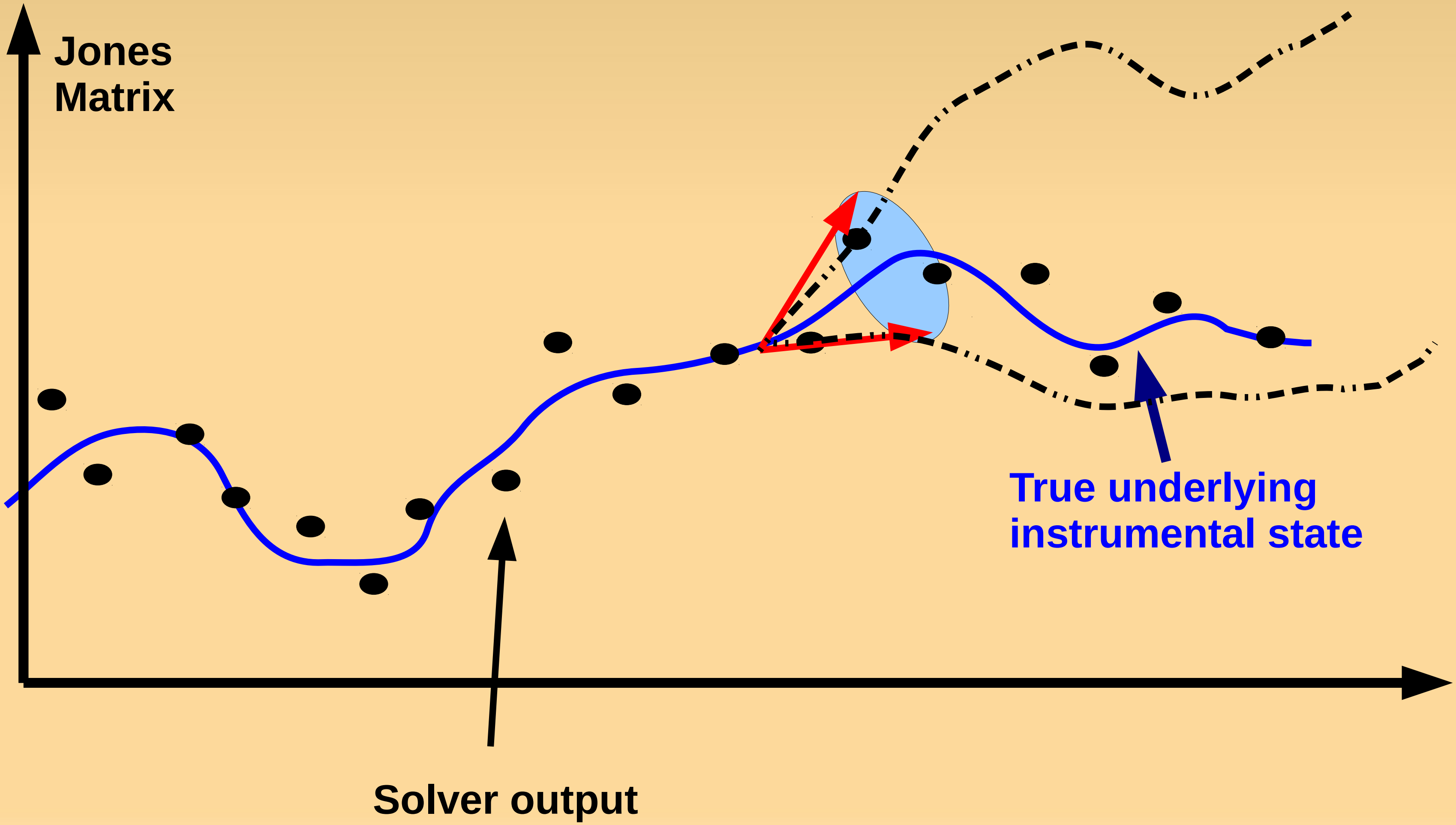
- Solvers: given the data, find the (max likelihood) underlying instrumental state (Jones matrix, ionosphere, clock delay, pointing error, etc.)

- Smoothness/continuity can be imposed via solution intervals, weighted solutions, etc.

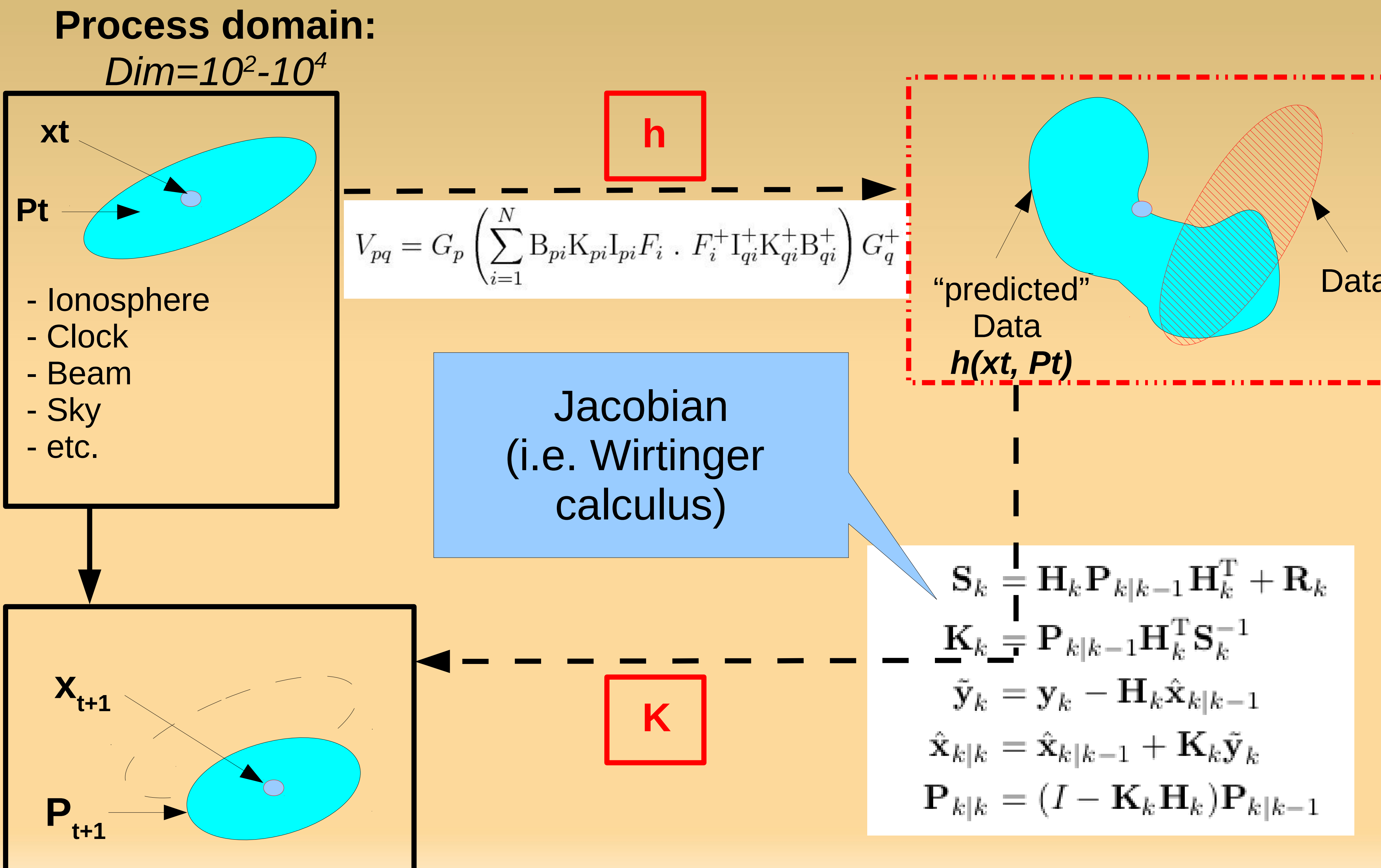


Filters vs. Solvers

- Filter: given current estimate of instrumental state, and new data, compute new instrumental state



Non-linear Kalman Filters



Iterative vs. Recursive

- Solvers are iterative
  - Start with best guess (e.g. previous state), iterate to convergence
- Filter is recursive, single step
  - New state = F(previous state, new data)
- Kalman filter is Bayesian, maximizes

Posterior is new state

Prior is previous state

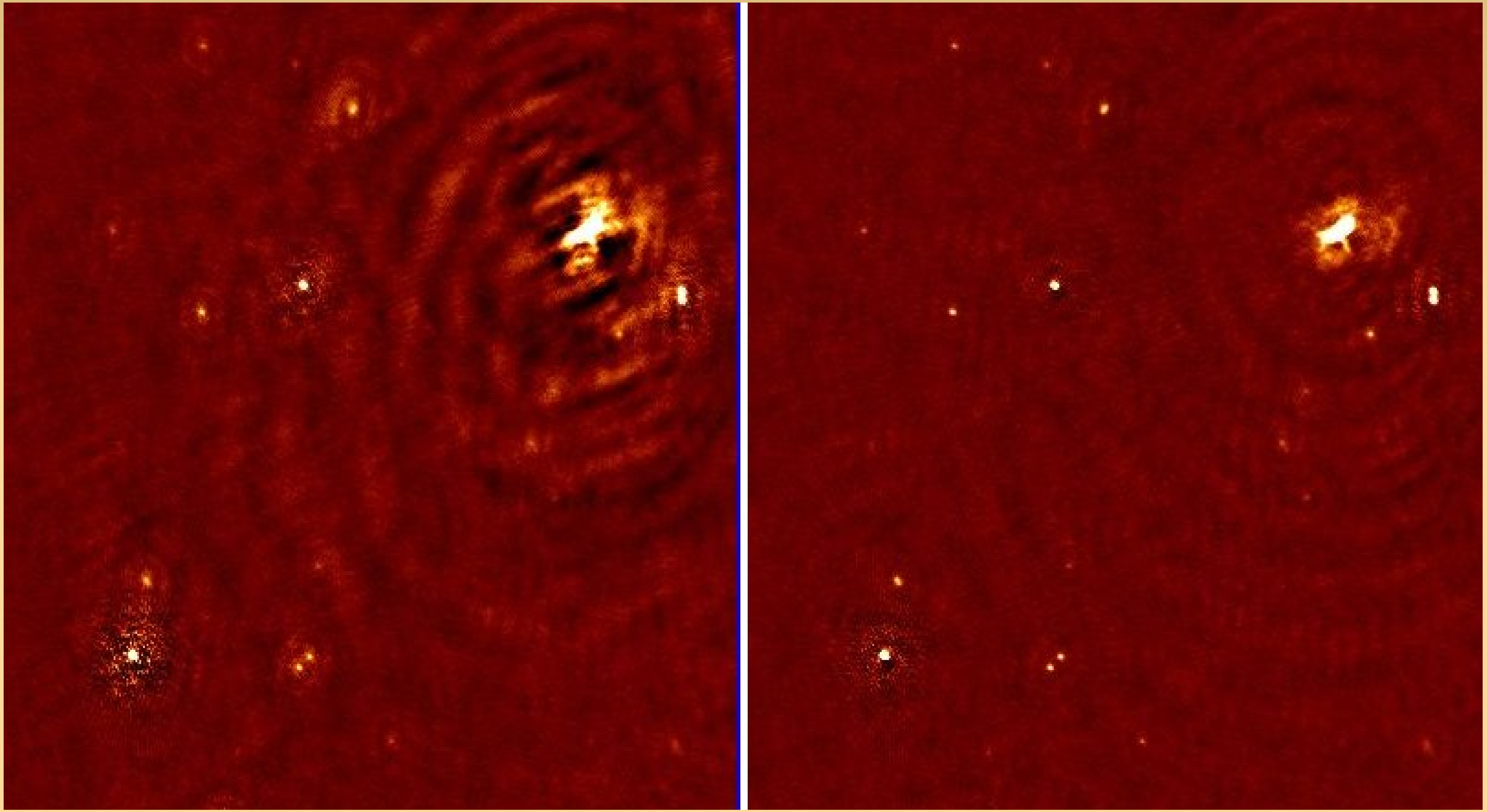
$$P(M|D) = P(D|M)P(M)$$

New data

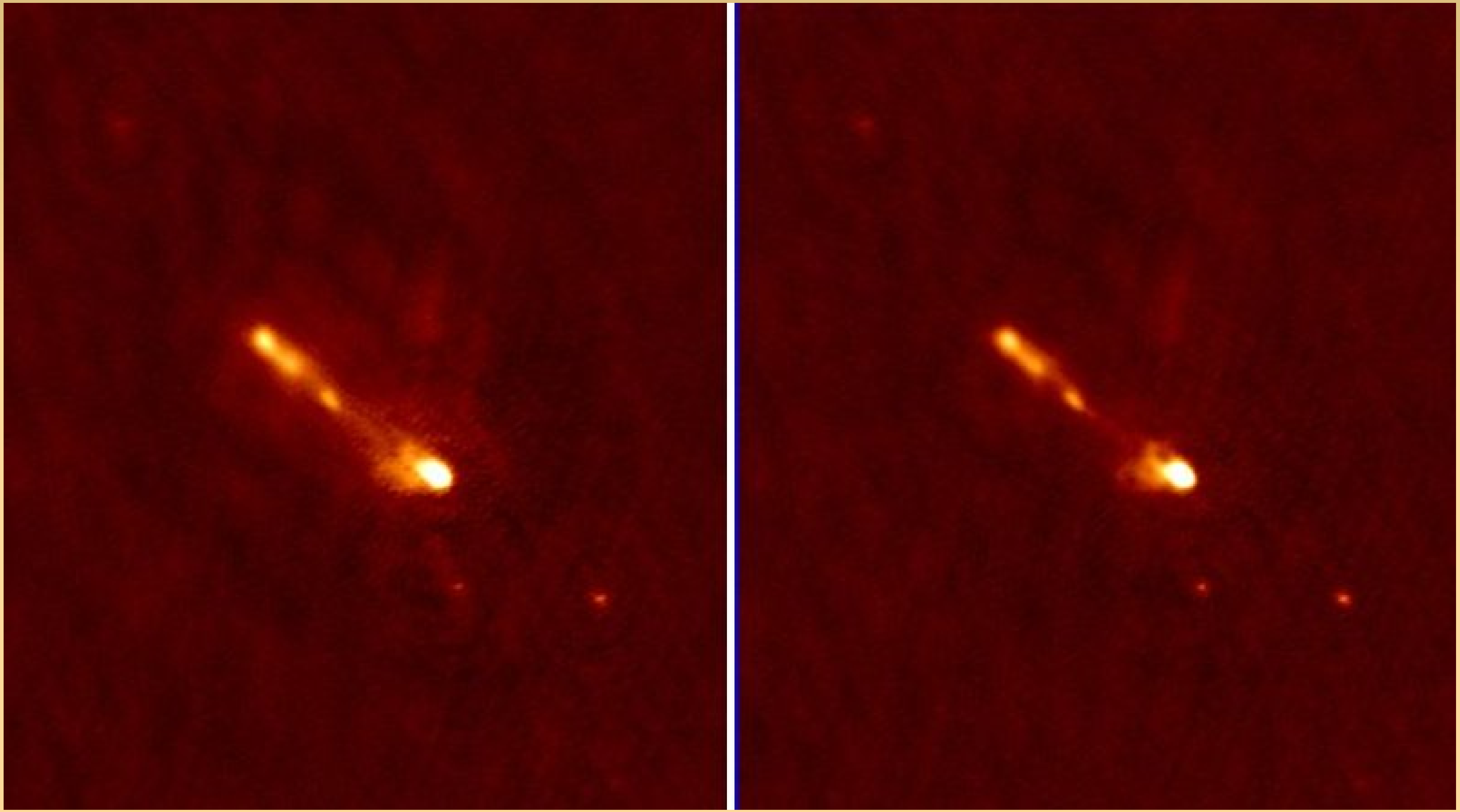
Does It Work?

- Implemented by Cyril Tasse (Obs Paris Meudon, ex SKA SA) as the KAFCA algorithm
- “Kalman Filters for Calibration”
- Can track clock offsets, TECs, DD Jones matrices
- Proven with LOFAR data

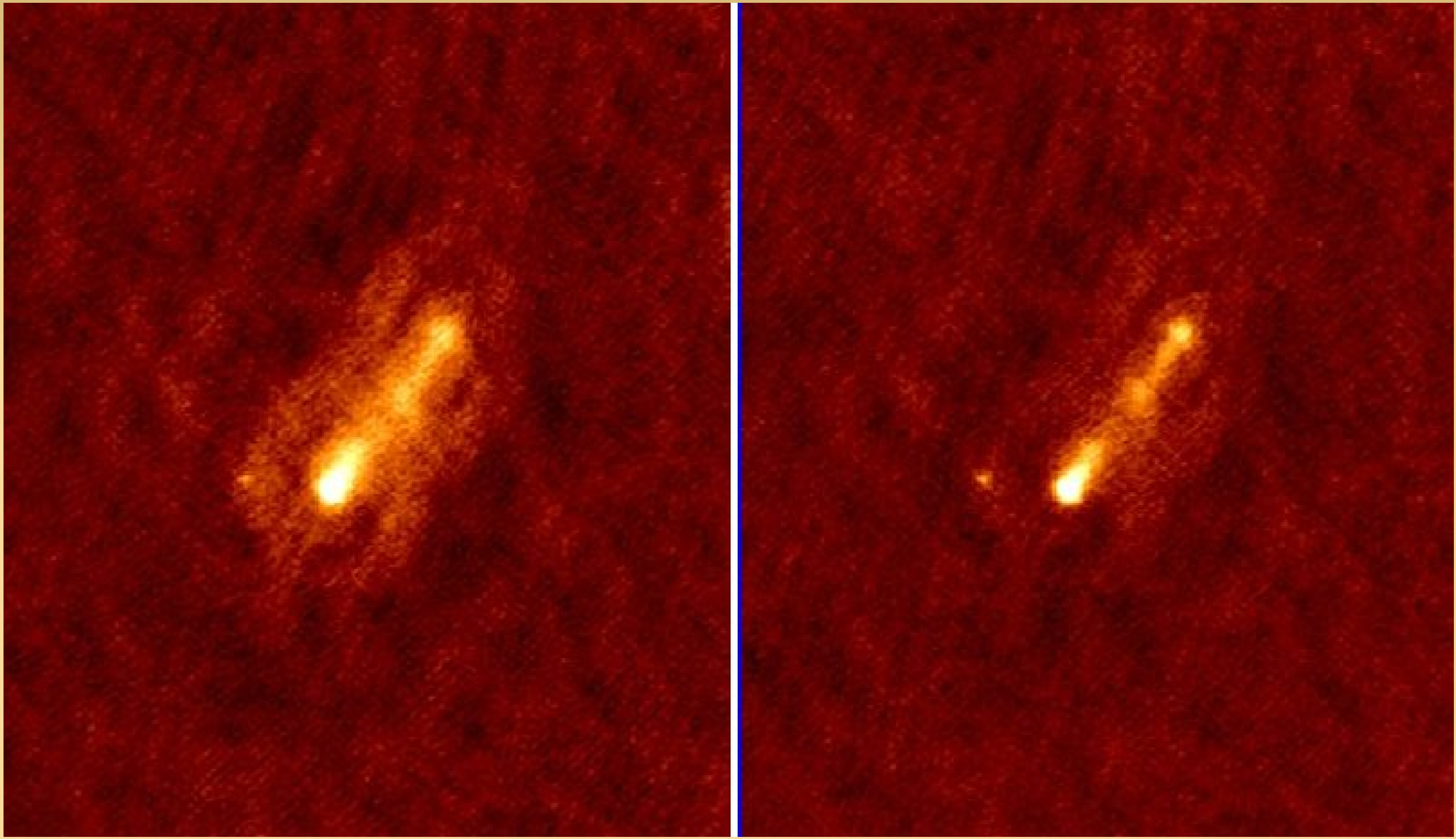
LOFAR Bootes Field



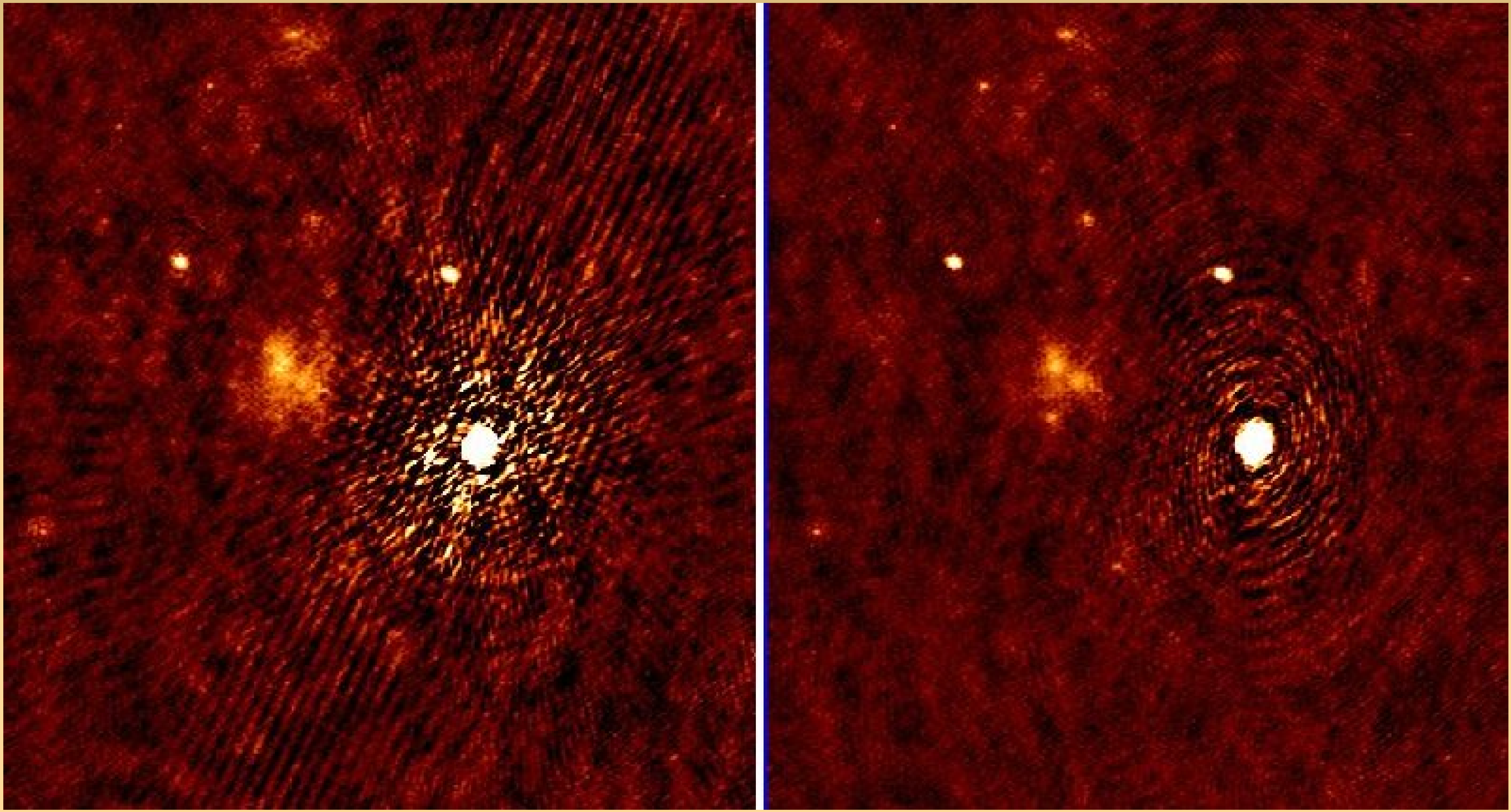
Bootes II



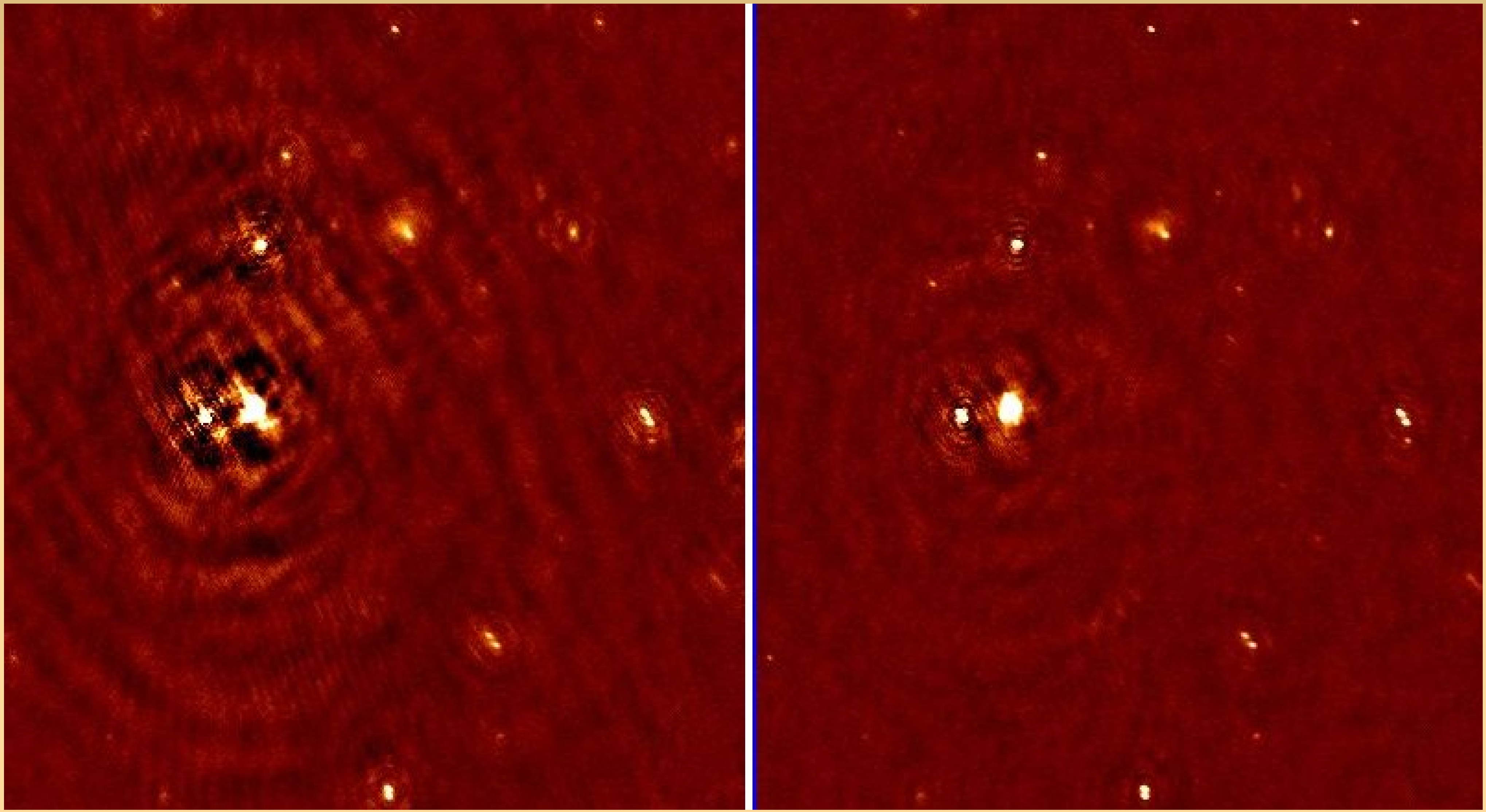
Bootes III



Bootes IV



Bootes V



Filter Advantages

- Single-pass vs. iterative
  KAFCA >> COHJONES >> Peeling!
- Stable w.r.t. bad data
- Can start thinking about streaming calibration
  - Still need a good sky model though...
- But, imagine a pipeline where you track the calibration solutions online, subtract brightest sources, and average down the data...

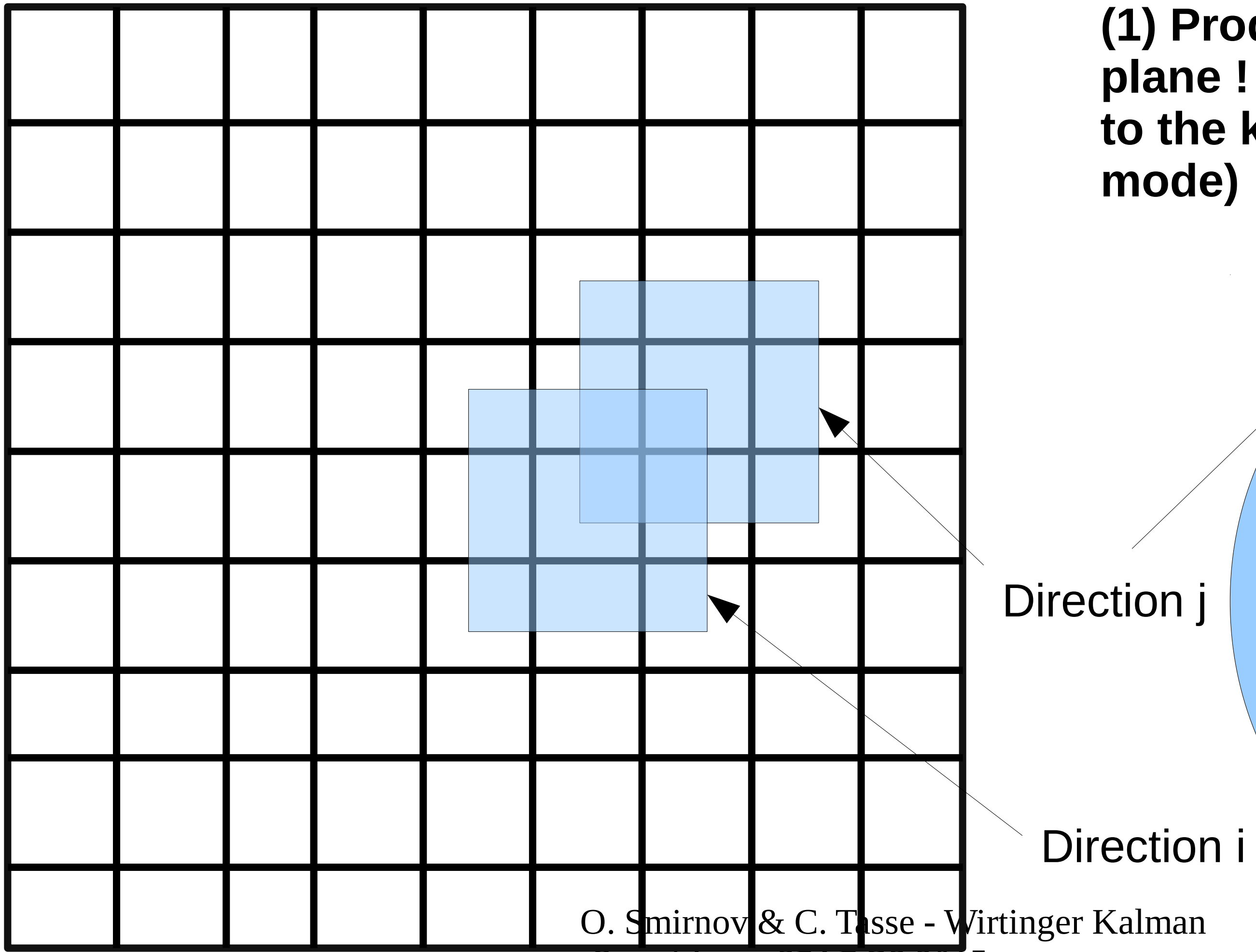
Applying DDEs

- Once the bright sources have been subtracted, how to apply solutions to the rest of the field?
- The New Way:
  - A-projection: convolutional gridding + FFT, integrated with deconvolution
- Old School: faceting
  - Image multiple facets, correct per facet
- A-projection shown to be more efficient in terms of pure FLOPS

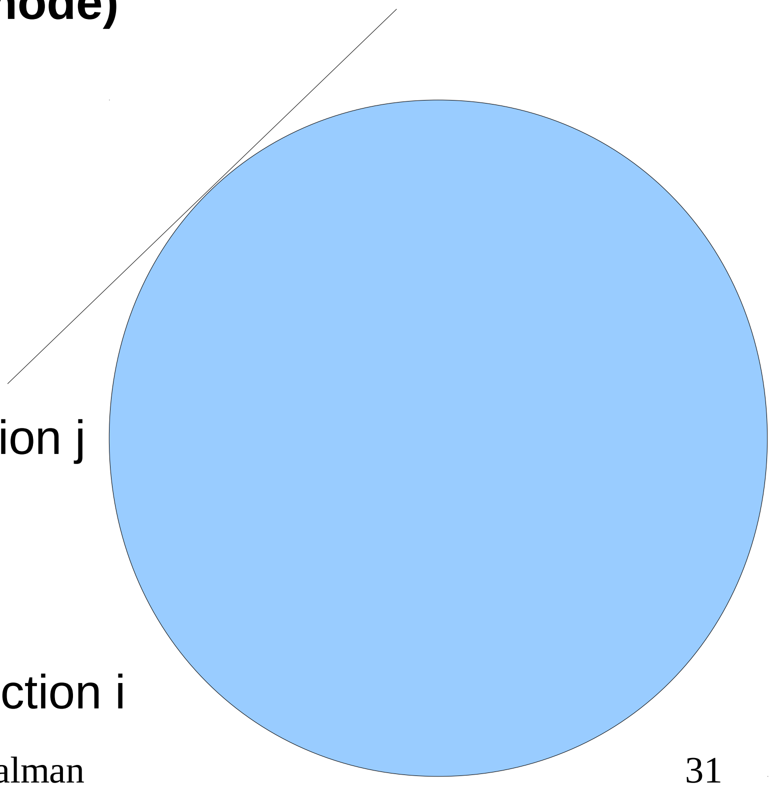
But...

- Convolutional gridding not easy to implement on GPUs (well)
  - Memory bandwidth often the bottleneck in GPU code
- Hierarchical memory (small fast vs large slow) is the current trend in HPC
  - This changes the landscape in terms of algorithmic efficiency (no longer enough to just count FLOPS)
- More computationally expensive algorithms may exhibit cheaper memory access patterns
- So, we're hedging our bets by reviving faceting

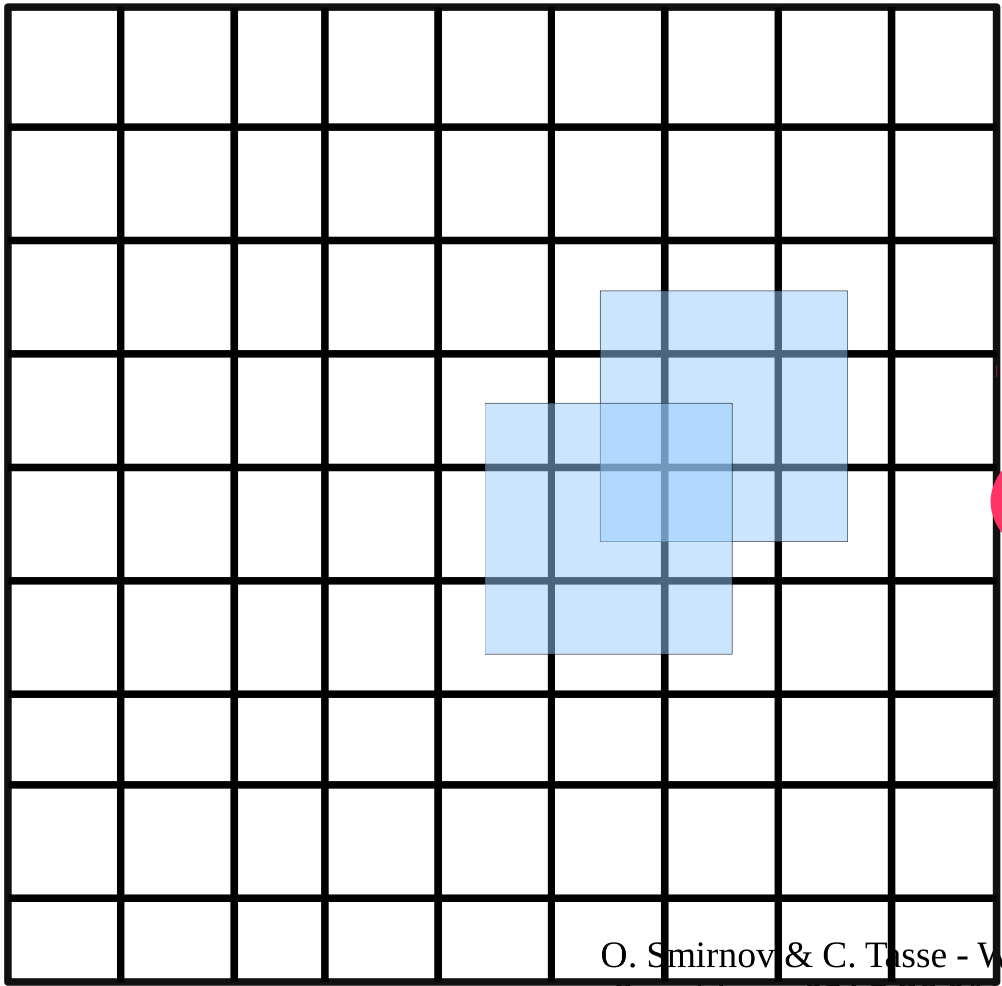
DDFacet (a baby imager)



(1) Produces a single tengential plane ! (no « noise jumps » thanks to the kalman filter, and facetting mode)



DDFacet (a baby imager)



(1) Produces a single tengential plane ! (no « noise jumps » thanks to the kalman filter, and facetting mode)

(2) Does full polarisation DDE correction

(3) Baseline Dependent Averaging
90 % of the data can be compressed

BUT
(4) Need to interpolate DDE (if drawn from Voronoi tesselation)

Baseline-Dependent Averaging

- Shorter baselines move much slower than longer ones
- And there's more of them (especially in core-heavy layouts)
- BDA (longer averaging on shorter baselines) is being explored as a means of data compression, esp. for SKA1
- Degree of (BD)A limited by field of view

BDA & Faceting

- Can average very little for wide fields
- But, a facet's FoV is tiny
  - Can average much more aggressively
  - On-the-fly, since visibilities must be phase-rotated to facet centre
- Averaging is much cheaper than gridding
- DDFacet: BDA on the fly saves >90% of gridding operations
- Impact on DR not clear (tested on wide/shallow LOFAR data for now)

Hedging Our Bets

- Benna Hugo (UCT) developing a GPU-based facet imager
- Iniyan Natarajan (UCT) developing pyImager, a generalization of A-projection to arbitrary beam patterns
- Clearly completely different computational and DR trade-offs

Conclusions

- Wirtinger calculus is easy and fun
- Kalman filters are a viable approach to (DI and DD) calibration
  - May enable (or simplify) streaming calibration
- Maybe time to remember faceting again
- Prospects are good
  - JVLA 5M+ image ~real-time processing
    - (Much much worse in human time though)
  - ...before any of the above is incorporated