Overview of the CASA Imager Module

(1) Imaging Process

(2) Required Imaging and Control Options

(3) Design of the Imaging Module

(4) Parallelization

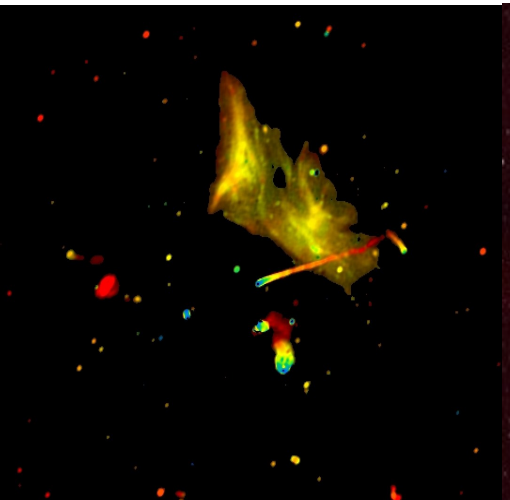
Core Imaging Team :

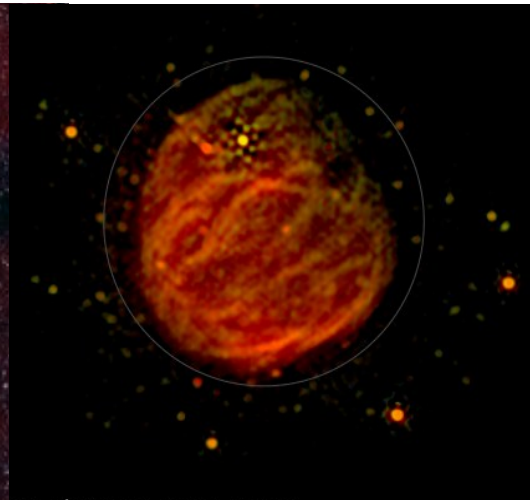
Kumar Golap

Sanjay Bhatnagar

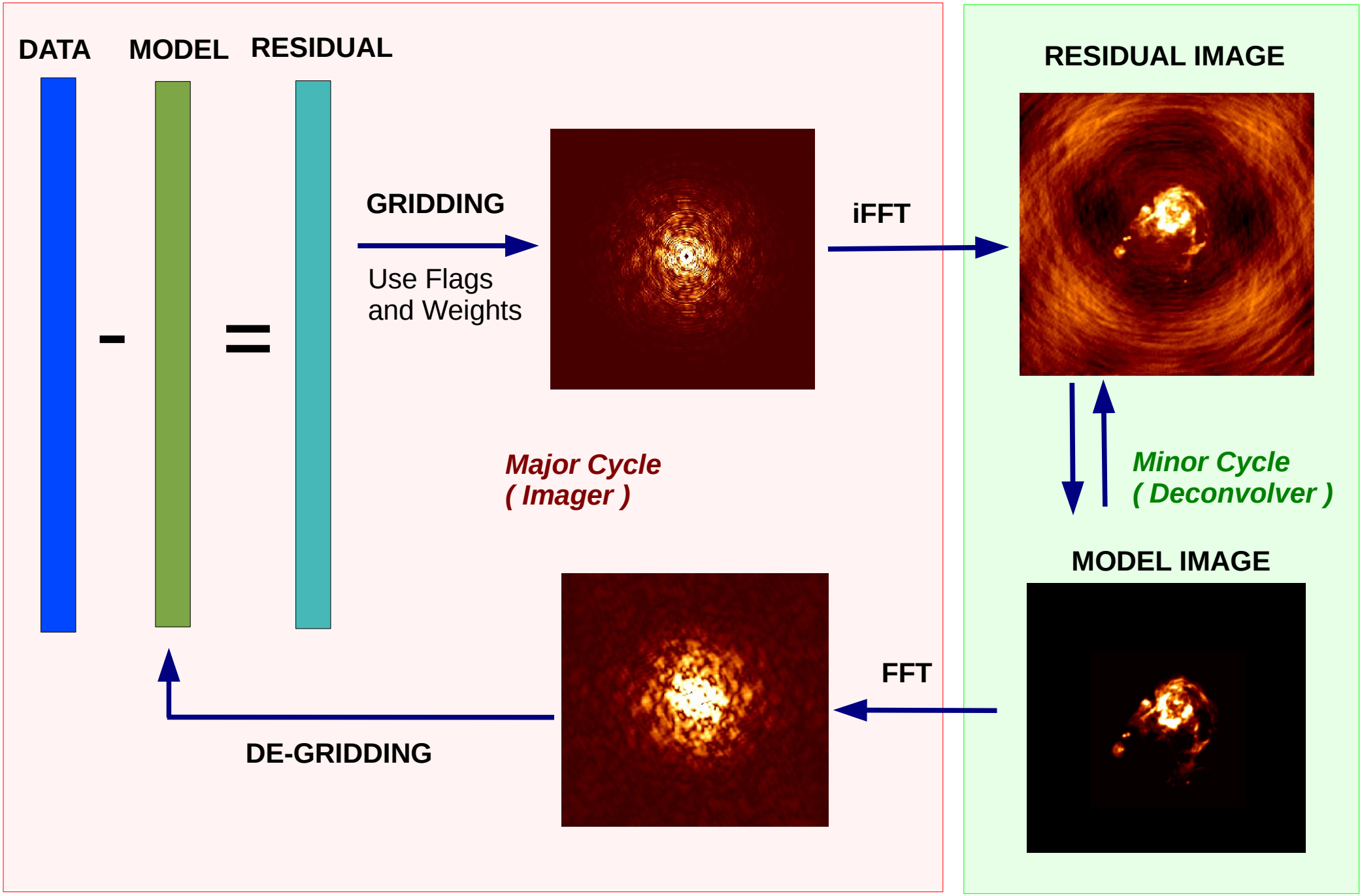
Urvashi Rau

CASA Developer Meeting, Socorro, NM 18 June 2013

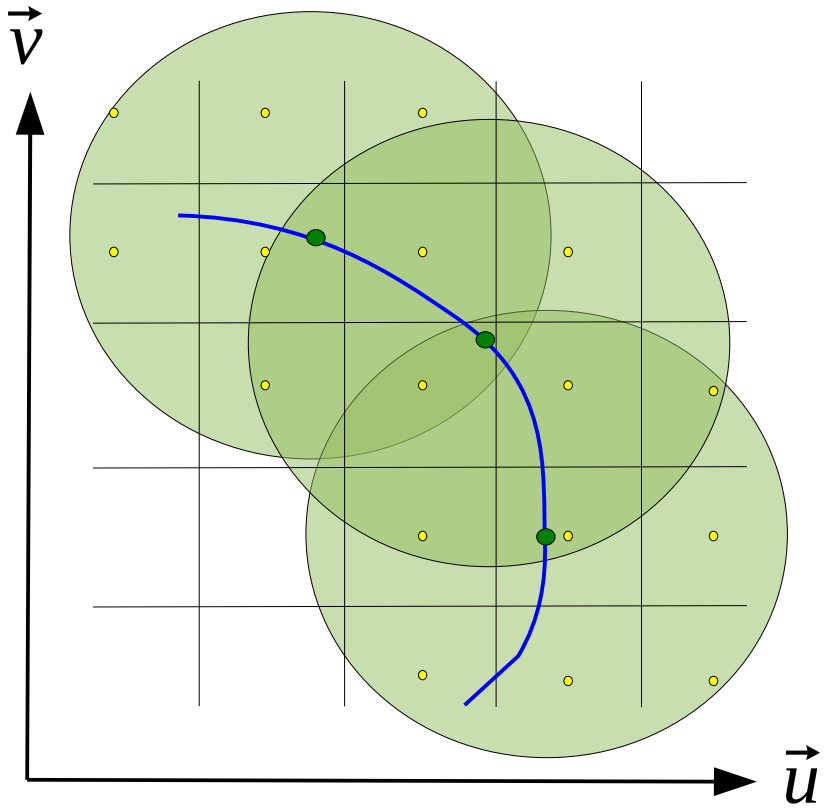




Imaging Process – Iterative model fitting via χ^2 minimization



Imaging Options – Gridding : Convolutional resampling



Gridding Convolution Function (GCF)

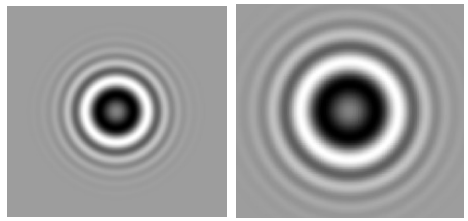
– Several GCF options (algorithms)

Size range : 3x3 to 20x20 pixels

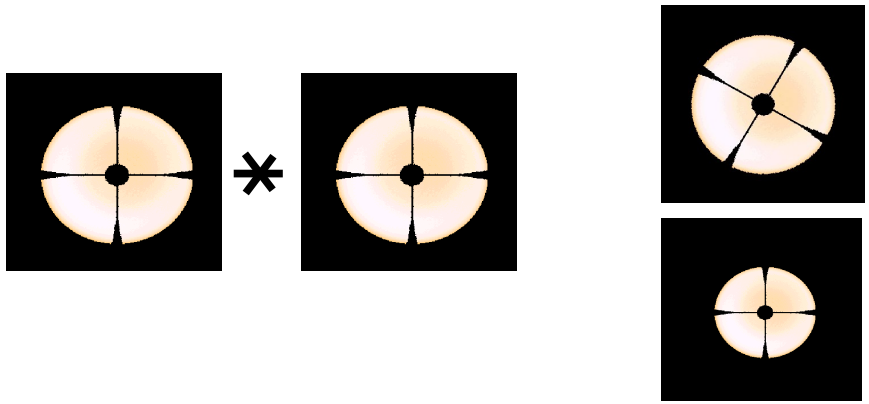
(with x100 oversampling)

Standard Imaging :
Prolate Spheroidal

W-Projection :
FT of a Fresnel kernel



A-Projection :
Convolutions of Aperture Illumination Funcs



Single Dish gridder : FT of Single-dish beam

Imaging Options – Minor Cycle Algorithms

For Point Sources :

- Hogbom Clean
- Clark Clean

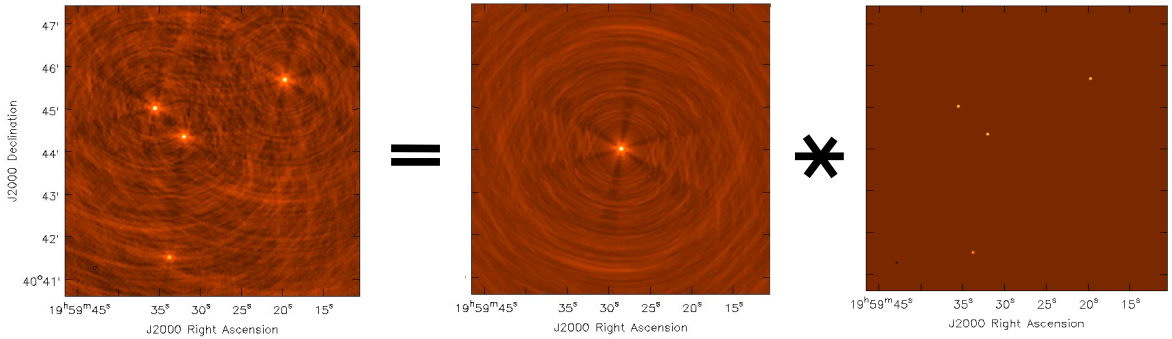
For Point/Extended Sources :

- Maximum-Entropy Method
- Adaptive-Scale Pixel Clean
- Multi-Scale-Clean

For Wide-band Images

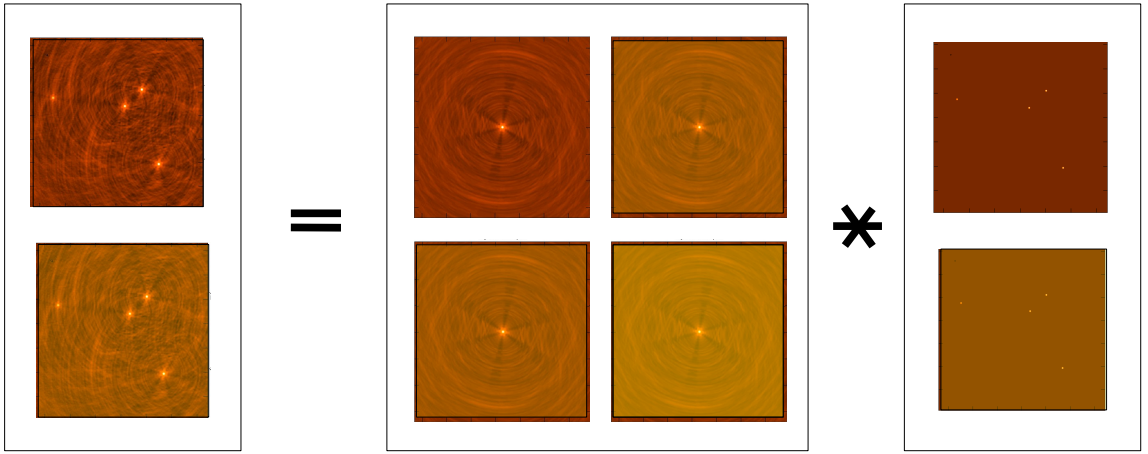
- Multi-Frequency-Clean
  (with or without Multi-Scale)

Convolution Equation ==> Deconvolution



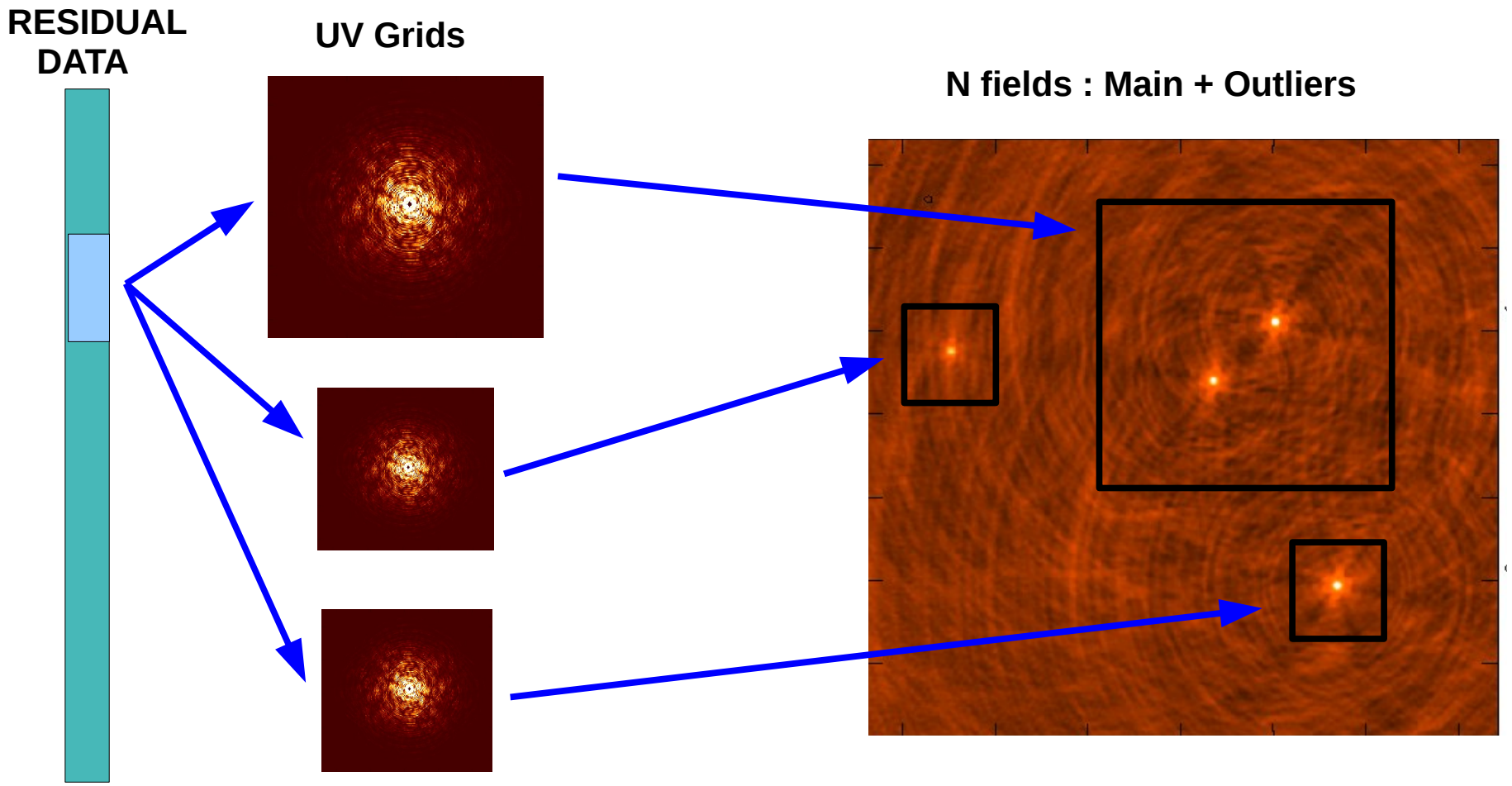
Multi-Term Convolution Equation

==> Joint Deconvolution



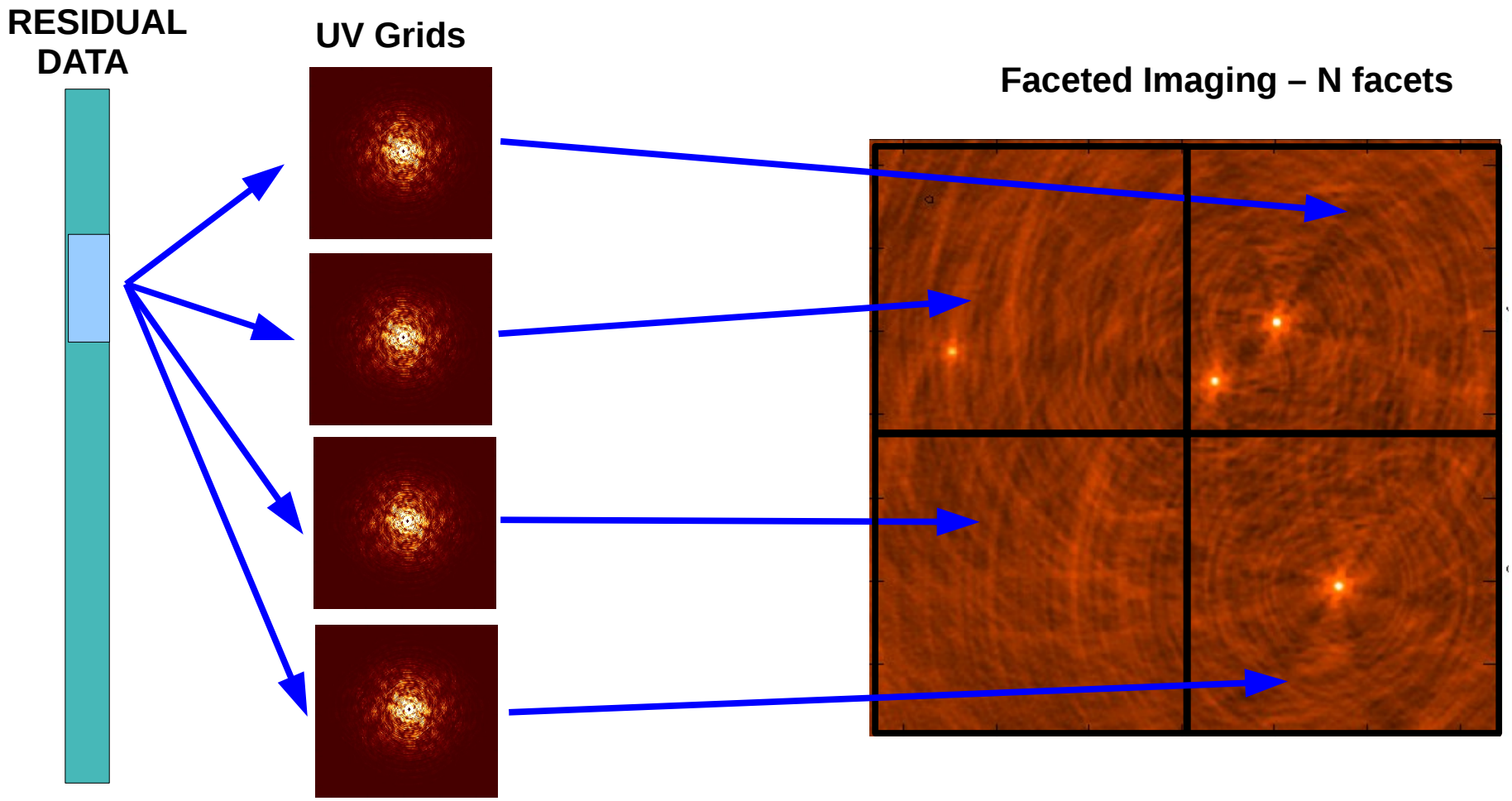
(Multi-Term Algorithms can be memory-intensive)

Imaging Options – Multiple Fields



- Work with N smaller sized images (deconvolve N images separately)
- A few outlier sources that must be reconstructed to prevent artifacts from contaminating the main field.

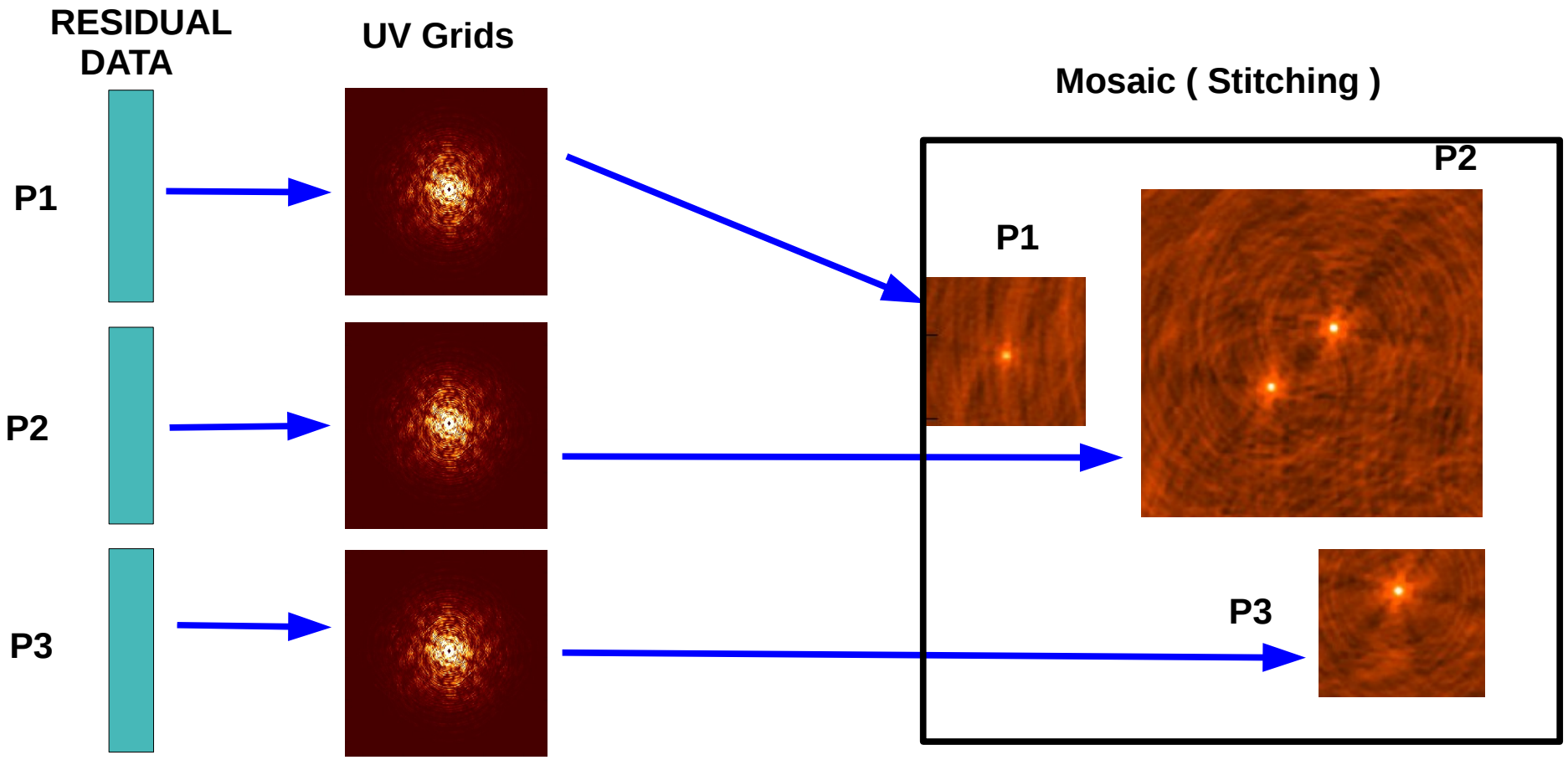
Imaging Options – Multiple Facets



– Wide-field imaging where array non-coplanarity and sky curvature produce artifacts away from the phase-center.

  – Work with smaller field-of-view images, each with its own phase-center
  – Deconvolve N facets separately (OR) as 1 single large image.

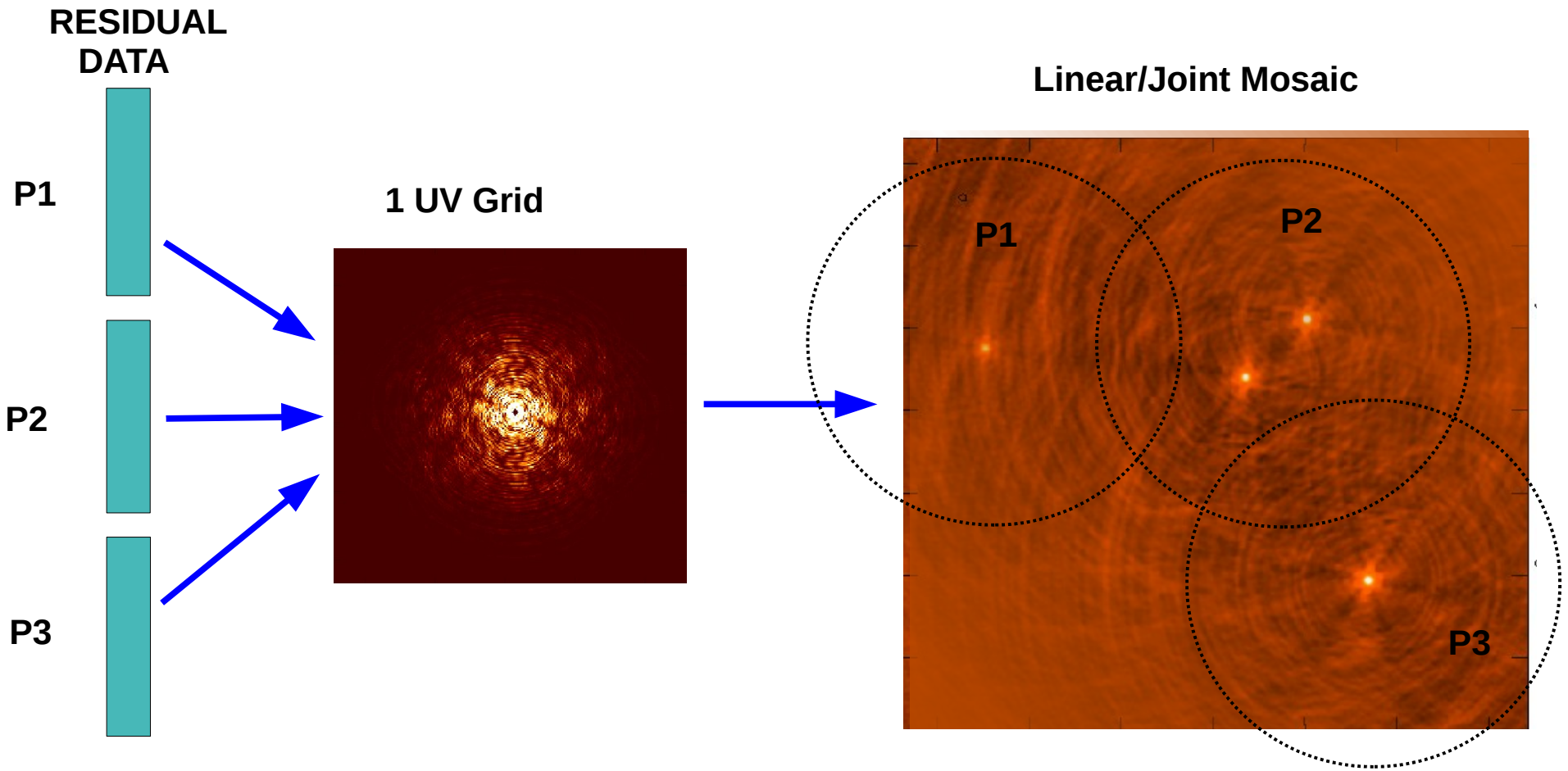
Imaging Options – Mosaics (multiple, separate pointings)



– Make a single image larger than a single-pointing field of view

– Deconvolve N images separately and stitch together final restored images.

Imaging Options – Mosaics (joint deconvolution)



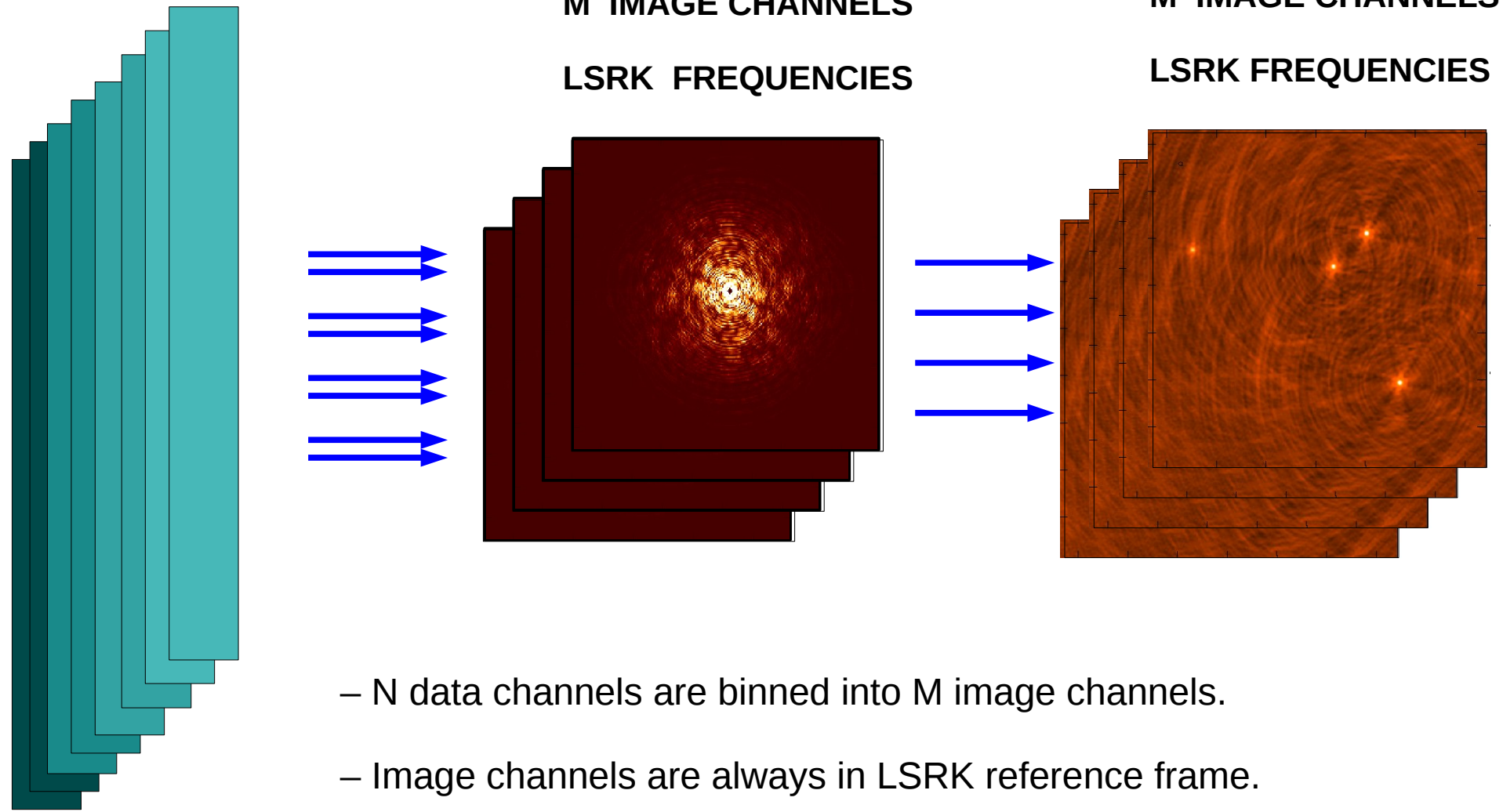
– Make a single image larger than a single-pointing field of view (Deconvolve 1 image)

– Math is very similar to “ facet ” and “ multi-field ” imaging, but using separate data.

– Uses aperture-illumination functions with phase-gradients as GCFs.

– Think of single-dish data as another 'pointing'

Imaging Options – Cube or Spectral-Line



– N data channels are binned into M image channels.

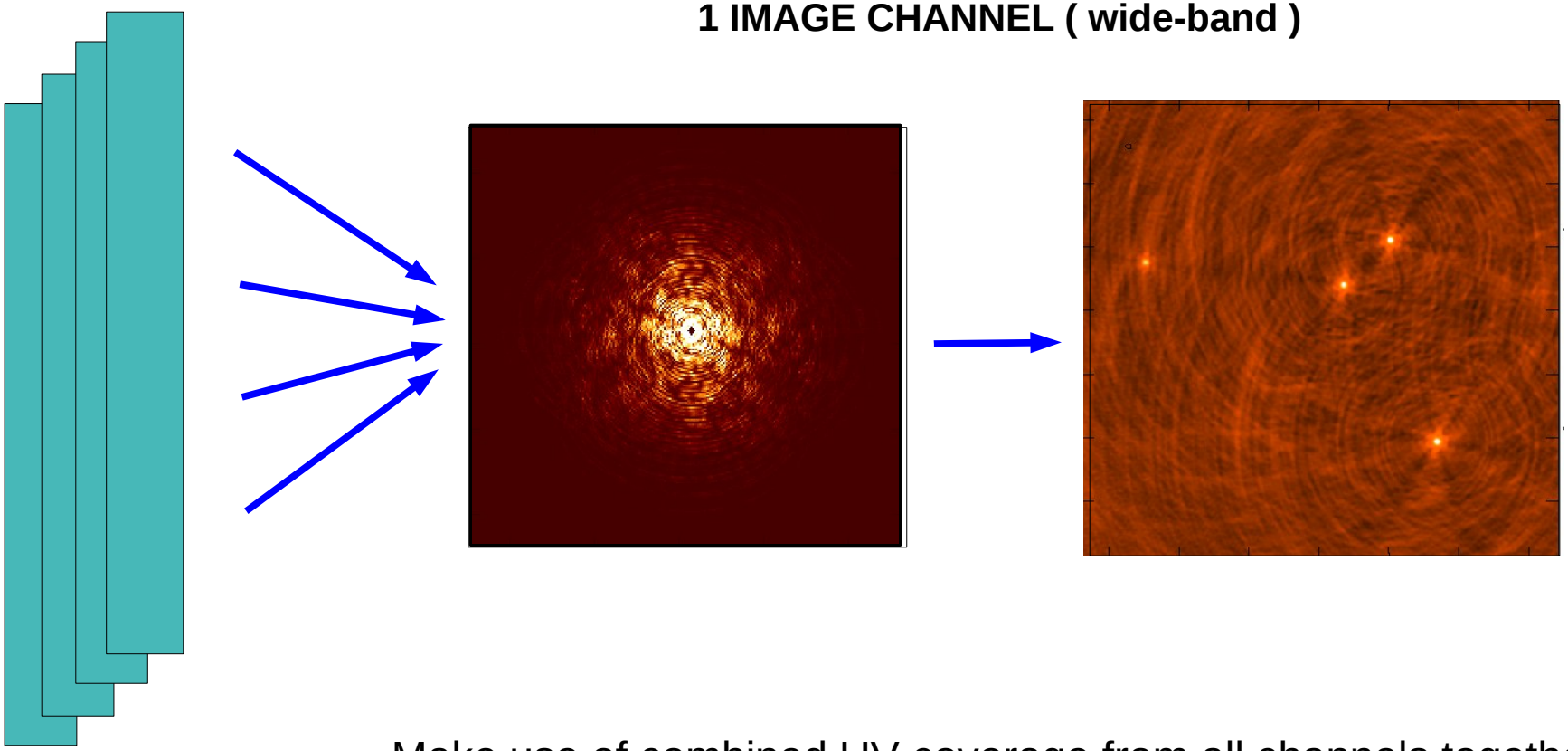
– Image channels are always in LSRK reference frame.

– Conversion to 'velocity', etc is only axis re-labeling (not regridding)

Imaging Options – Multi-Frequency Synthesis

N DATA CHANNELS

1 IMAGE CHANNEL (wide-band)

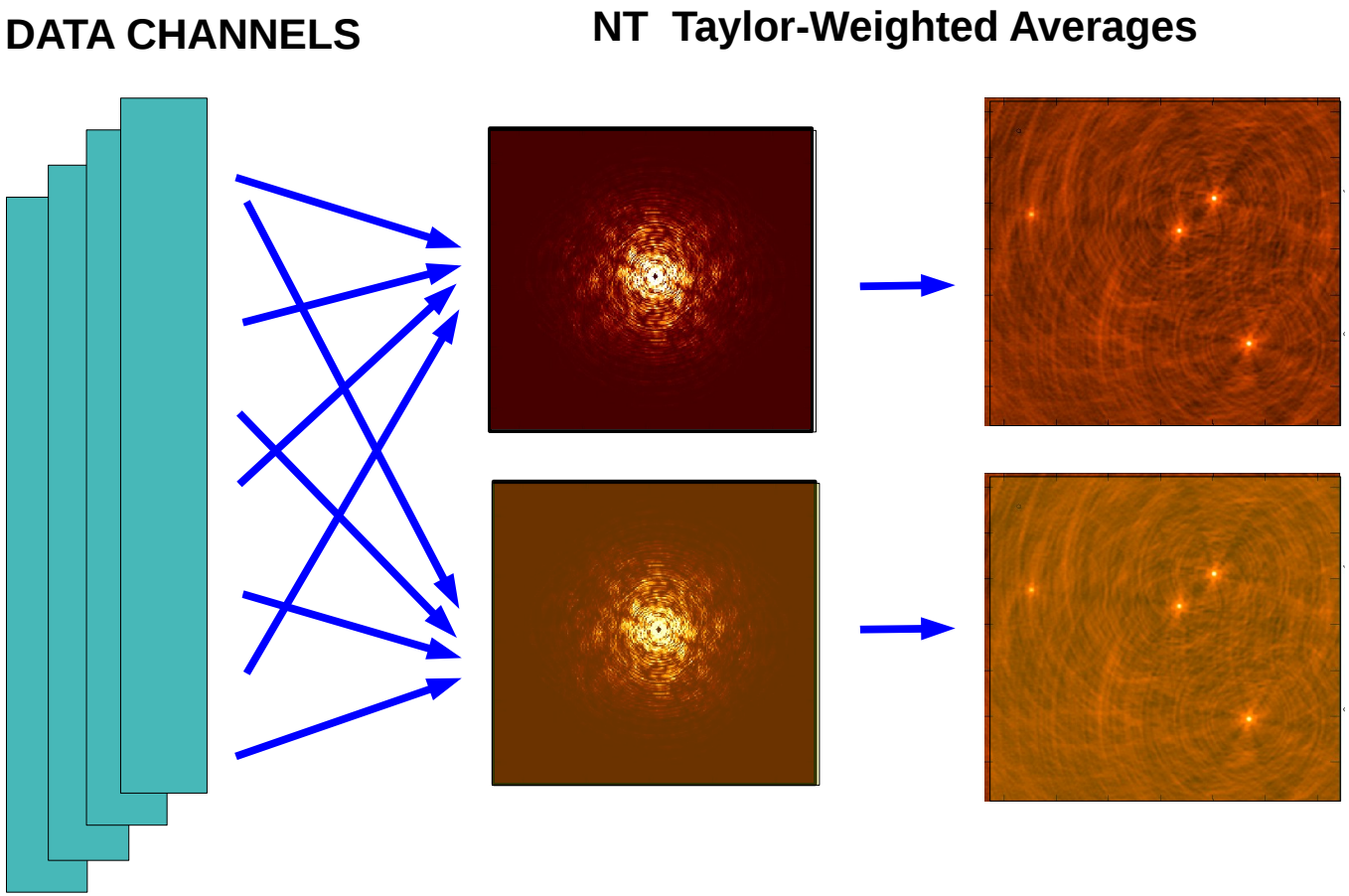


– Make use of combined UV-coverage from all channels together

– Make use of broad-band sensitivity during image reconstruction

– Deconvolve 1 image

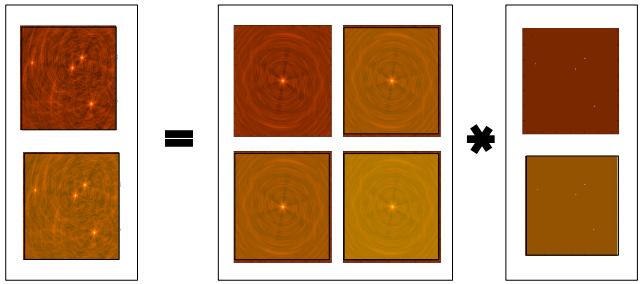
Imaging Options – Multi-Frequency Synthesis (N-Terms > 1)



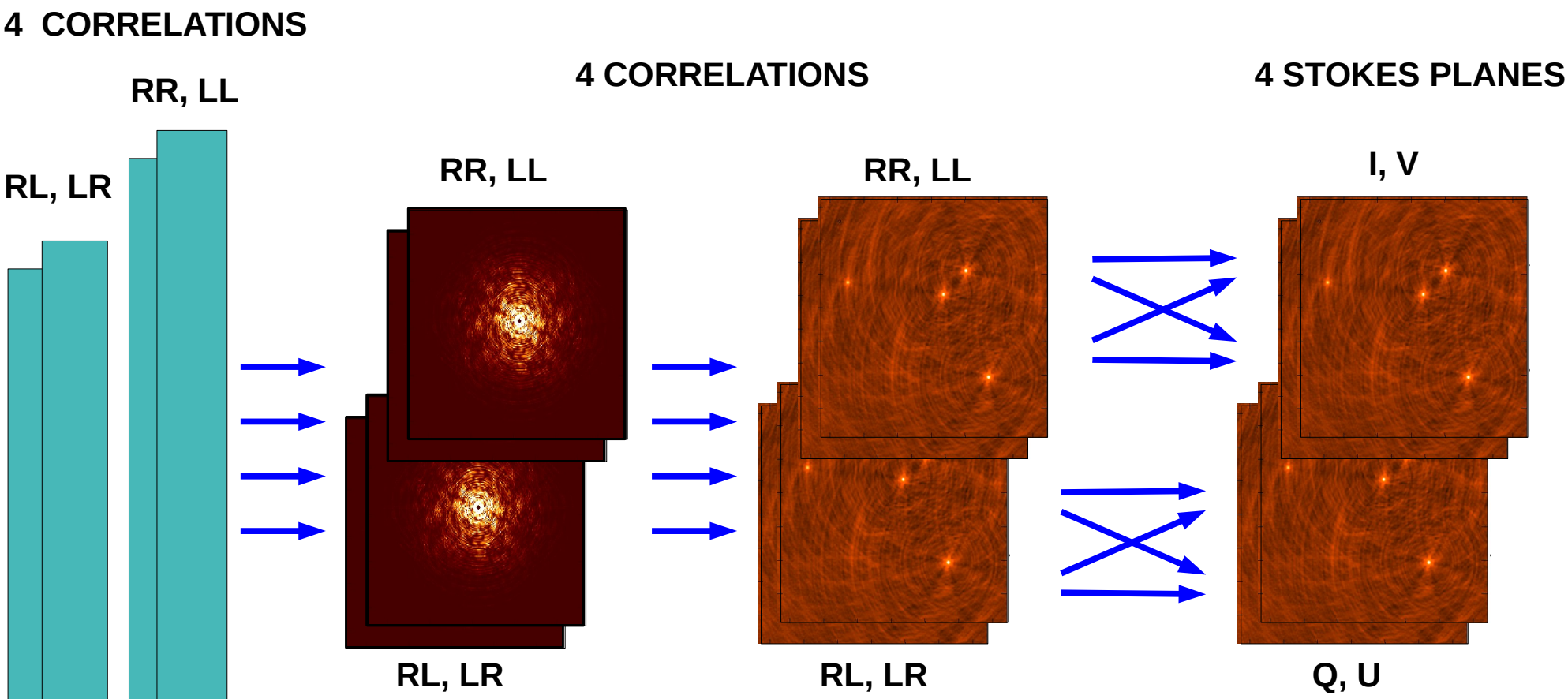
– Combined UV-coverage and broad-band sensitivity

– Solve for sky spectrum as well as intensity.

– Joint multi-term deconvolution of all Taylor coefficients



Imaging Options – Correlations and Stokes Parameters



Users can choose to make images of

R/L => I, Q, U, V, IV, QU, IQUV, RR, LL, LR, RL, RRLL, RLLR, 'all'

X/Y => I, Q, U, V, IQ, UV, IQUV, XX, YY, XY, YX, XXYY, XYYX, 'all'

Required Imaging Options – Almost all combinations !

– Gridding Convolution Functions (Standard, W-Proj, A-Proj, …)

– Deconvolution Algorithms (Clark/Hogbom Clean, MS-Clean, ASP, MEM)

– Cube / Spectral line (vs) Multi-Frequency Synthesis (Nterms = 1 or > 1)

– Stokes Parameters (I, Q, U, V, IV, QU,...., RR....., XX,...)

– Multiple Fields, Multiple Facets, Simple / Joint Mosaics

User Interaction

– Create and edit masks during the Minor Cycle
– Ability to monitor progress and change deconvolution parameters at run-time

Parallelization

– Continuum Imaging : Data Parallelization for Major Cycle
– Cube Imaging : Major and Minor Cycles are parallelized

Design of the Imager Module – C++ Lower Level

Basic Functional Unit : 1 Image field, N Frequency planes, M Stokes planes

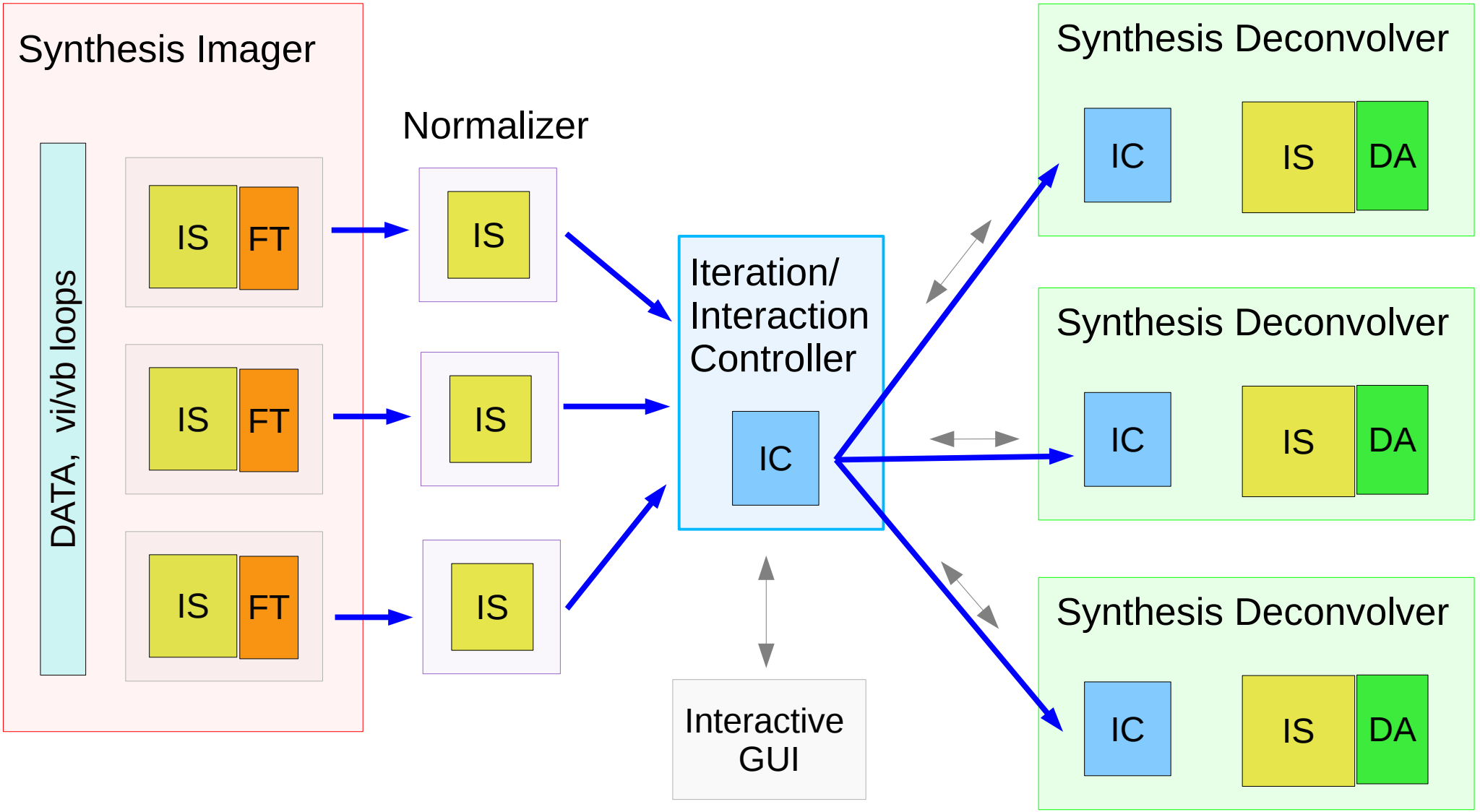
IS — Image Store : Residual, PSF, Model, Weight, Restored, Mask

FT — FTMachine : Gridding / de-Gridding + Convolution Functions

DA — Deconvolver Algorithm : Iteratively reconstruct the sky model

IC — Iteration Controller : Check stopping criterion between Major and Minor cycles + user-interaction

Design of the Imager Module – C++ Top Level, Tools



Example : 1 data set, 3 image fields
No parallelization

Design of the Imager Module – Steps at Tool level (or top C++)

Python Implementation Layer

class PySynthesisImager

Init SI , SD [n_field], IC

SI . select_Data (Data and selection parameters)

for field in range (0 , n_field) :

    SI . add_Field (Image Parameters , Gridding parameters)

    SD [field] . setup_Deconvolution (Algorithm parameters)

IC . setup_IterationControl (niter, threshold, gain...)

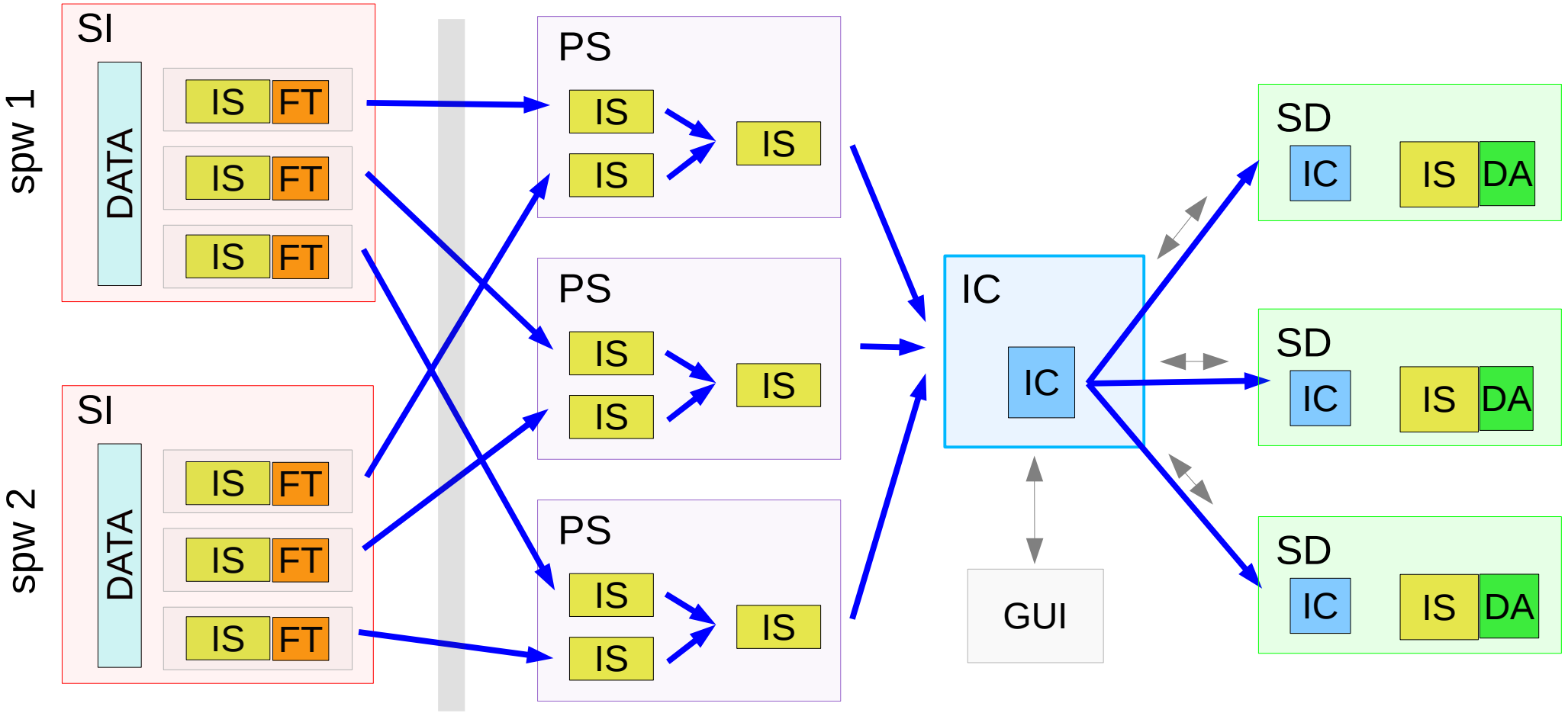
Design of the Imager Module – Steps at Tool level (or top C++)

Python Implementation Layer

class PySynthesisImager

```
peak_res = SI . run_Major_Cycle ( )

IC . update ( peak_res )

while ( ! IC . has_Converged( ) ) :

        For field in range ( 0 , n_field ) :

                rec = SD [ field ] . run_Minor_Cycle ( )

                IC . update ( rec )

         peak_res = SI . run_Major_Cycle ( )

         IC . update ( peak_res )


For field in range ( 0 , n_field ) :

         SD [ field ] . restore ( )
```

Parallelization – Continuum Imaging

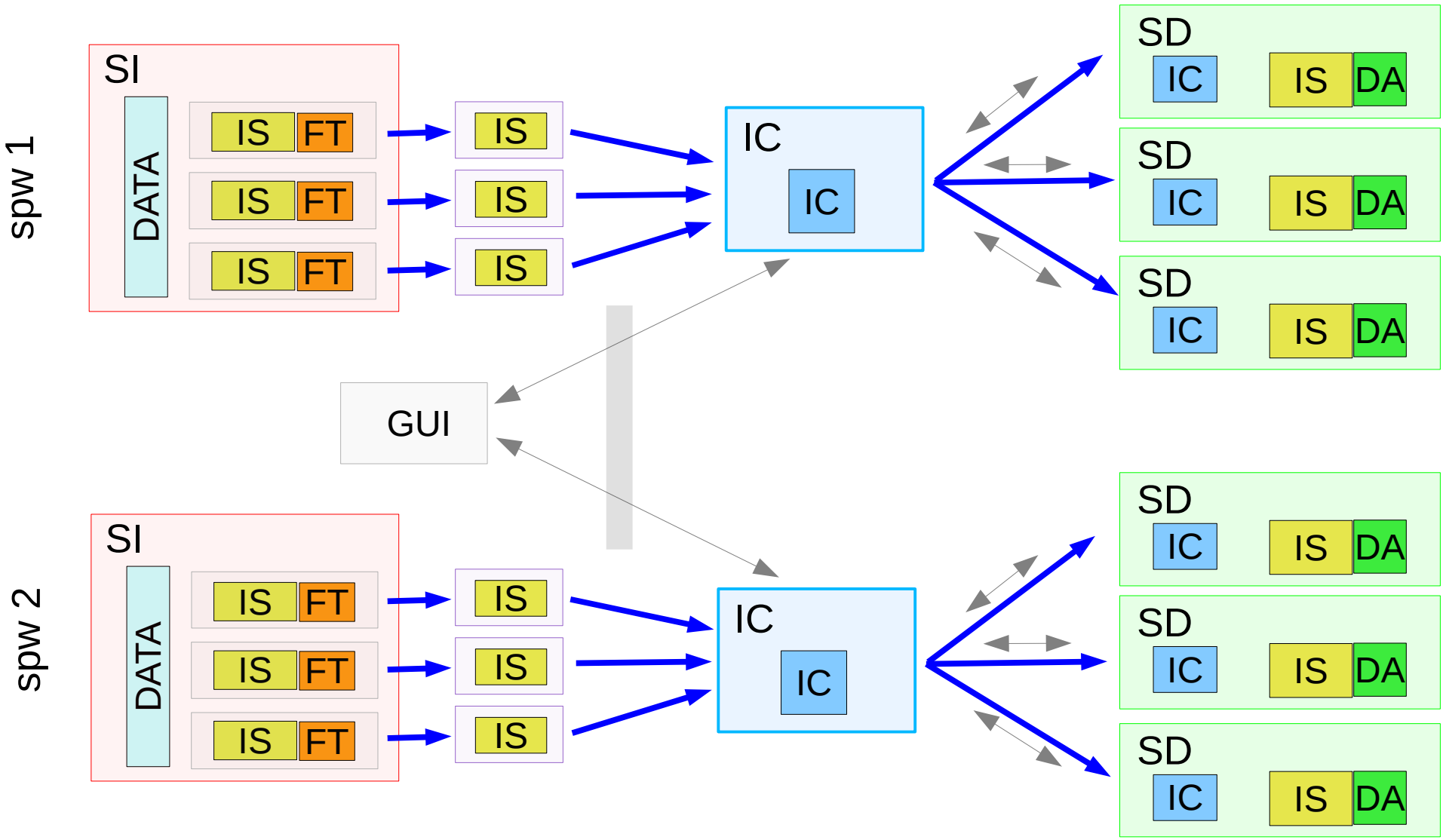


Parallel Sync (C++) : Gather_Residual , Scatter_Model , Norm_by_Weight

Example : 1 data set, 3 image fields
Data/channel parallelization on 2 nodes

class PyParallelContSI (PySI)

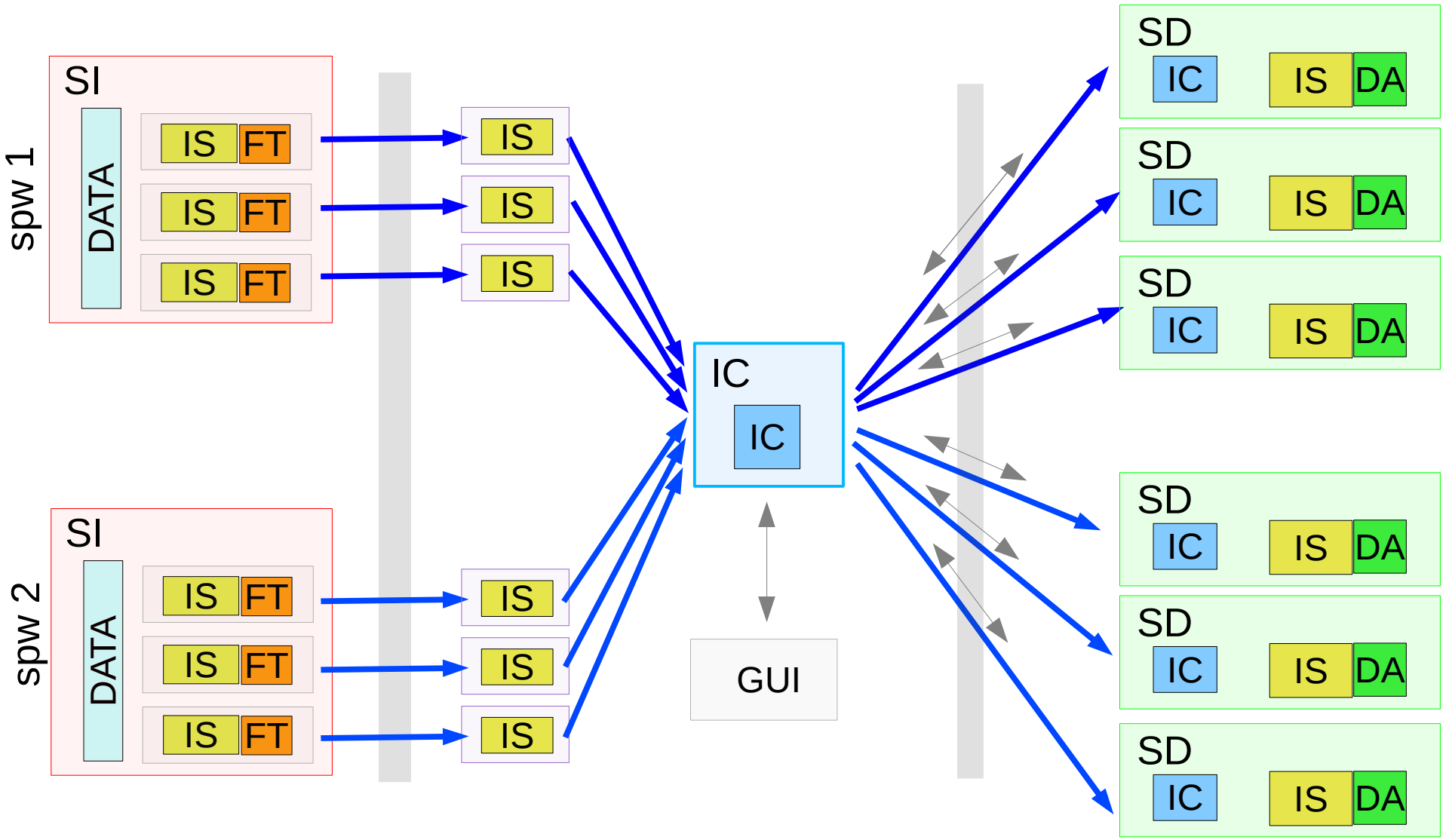
Parallelization – Cube Imaging – Independent Chunks



Example : 1 data set, 3 image fields , 2 SPWs
Parallelization on 2 nodes (cube slicing)

List of PySynthesisImagers

Parallelization – Cube Imaging – Sync after Major Cycles



Example : 1 data set, 3 image fields , 2 SPWs
Parallelization on 2 nodes (cube slicing)

Design of the Imager Module – Tasks (only user-interface)

" clean " : Runs imaging and deconvolution loops.

– Imager + ParSync + Deconvolver + IterBot

– Different interfaces showing different complexity

– Basic, Wide-field, Wide-band, Mosaic, etc
or mild / moderate / extreme (need good names here !)

" major cycle " : Makes only observed / residual images

– Imager + ParSync

" deconvolver " : Image-domain deconvolution

– Deconvolver + IterBot (+ GUI)