Imaging using GPU

V-K Veligatla, Kapteyn Institute
P. Labropoulos, ASTRON and Kapteyn Institute
L. Koopmans, Kapteyn Institute

Introduction

- What is a GPU?
- Why another Imager?
  - Large amount of data to be processed.
  - W-projection is a requirement.
- Issues to Consider
  - Wide field imaging
  - Full Stokes
  - Speed
  - Experiment with New approaches

Basic Imaging Equation

$$V(u,v,w)=\int \frac{I(l,m)B(l,m)}{\sqrt{1-l^2-m^2}} e^{-2\pi i\left(ul+vm+w\left(\sqrt{1-l^2-m^2}-1\right)\right)} dl\,dm$$

- For low frequency wide-field imaging the assumption w~0 does not hold
- If w=0, then visibilities = FFT(image)
- But this is not always the case hence W-Projection. (S. Bhatnagar and T.J.Cornwell)

W-Projection

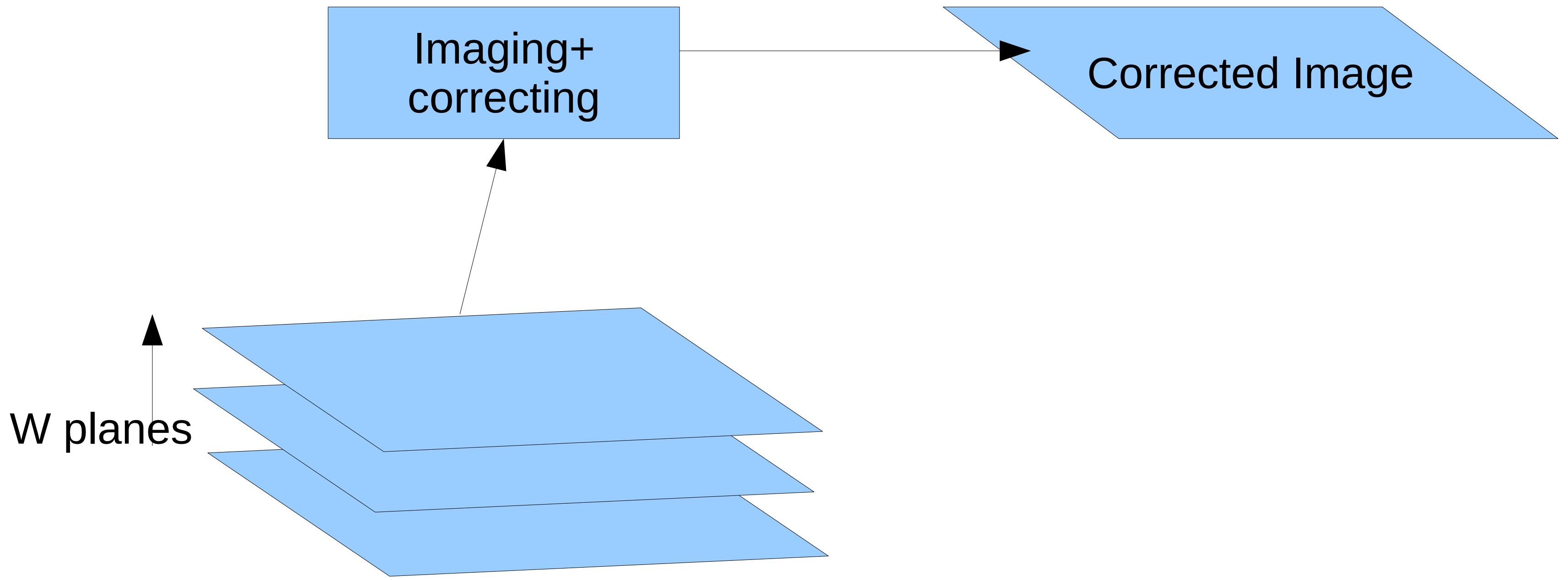
$$V(u,v,w)=\int \frac{I(l,m)B(l,m)}{\sqrt{1-l^2-m^2}} e^{-2\pi i(ul+vm+w(\sqrt{1-l^2-m^2}-1))} \, dl \, dm$$

Assuming W is constant for a given plane of UV values

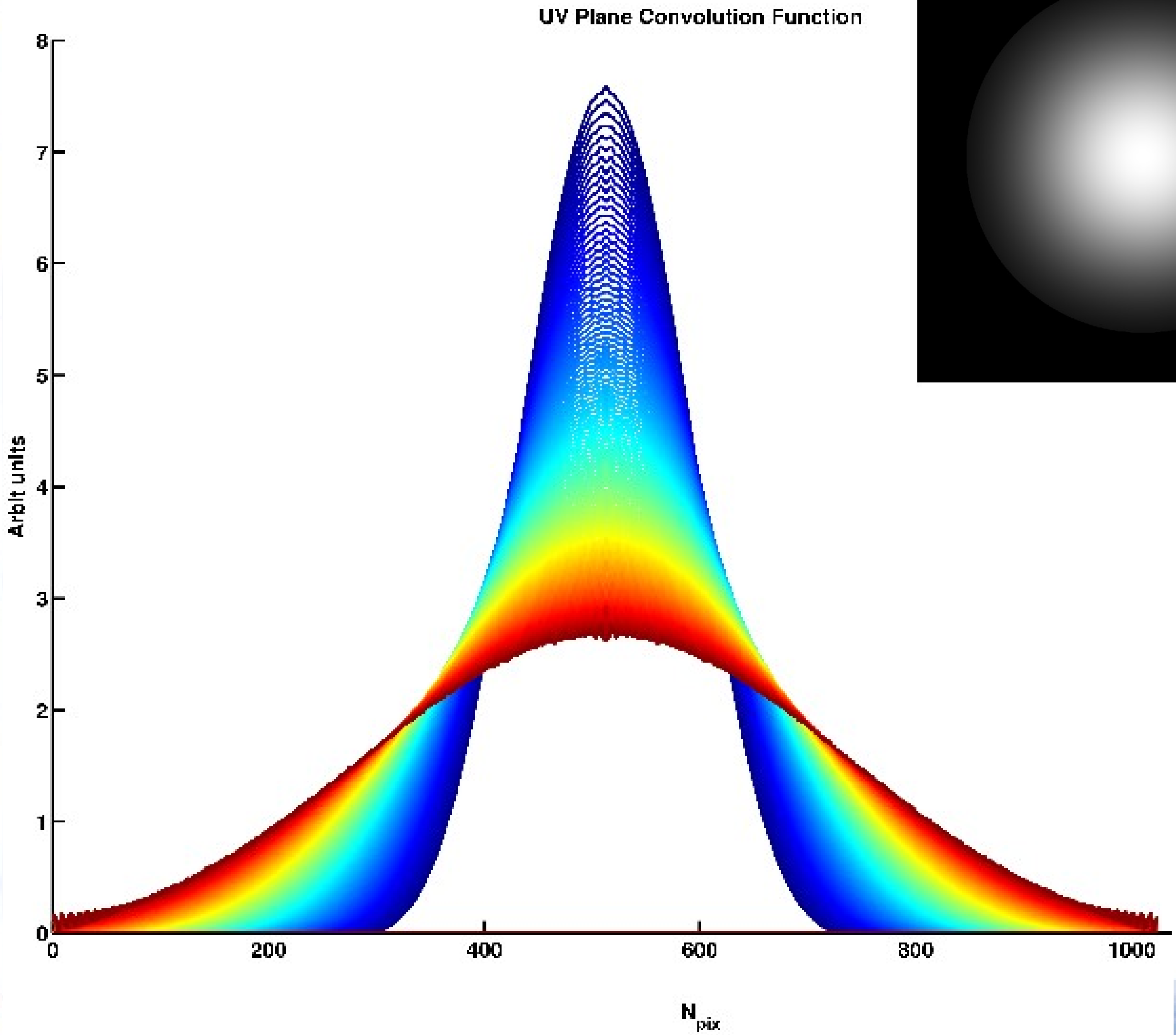
$$V(u,v,w)=\int e^{-2\pi i[w(\sqrt{1-l^2-m^2}-1)]} \frac{I(l,m)B(l,m)}{\sqrt{1-l^2-m^2}} e^{-2\pi i(ul+vm)} \, dl \, dm$$

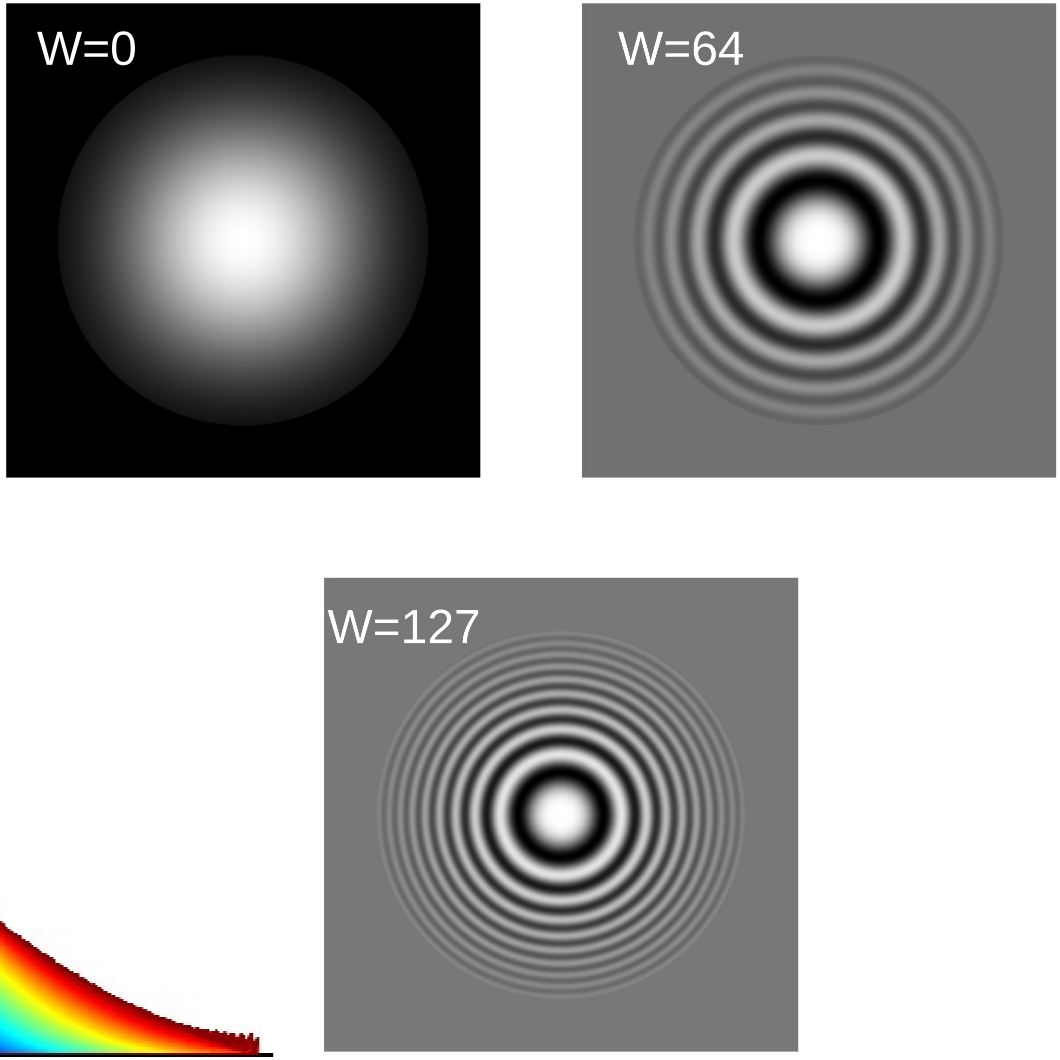
Correction Term

Block Diagram

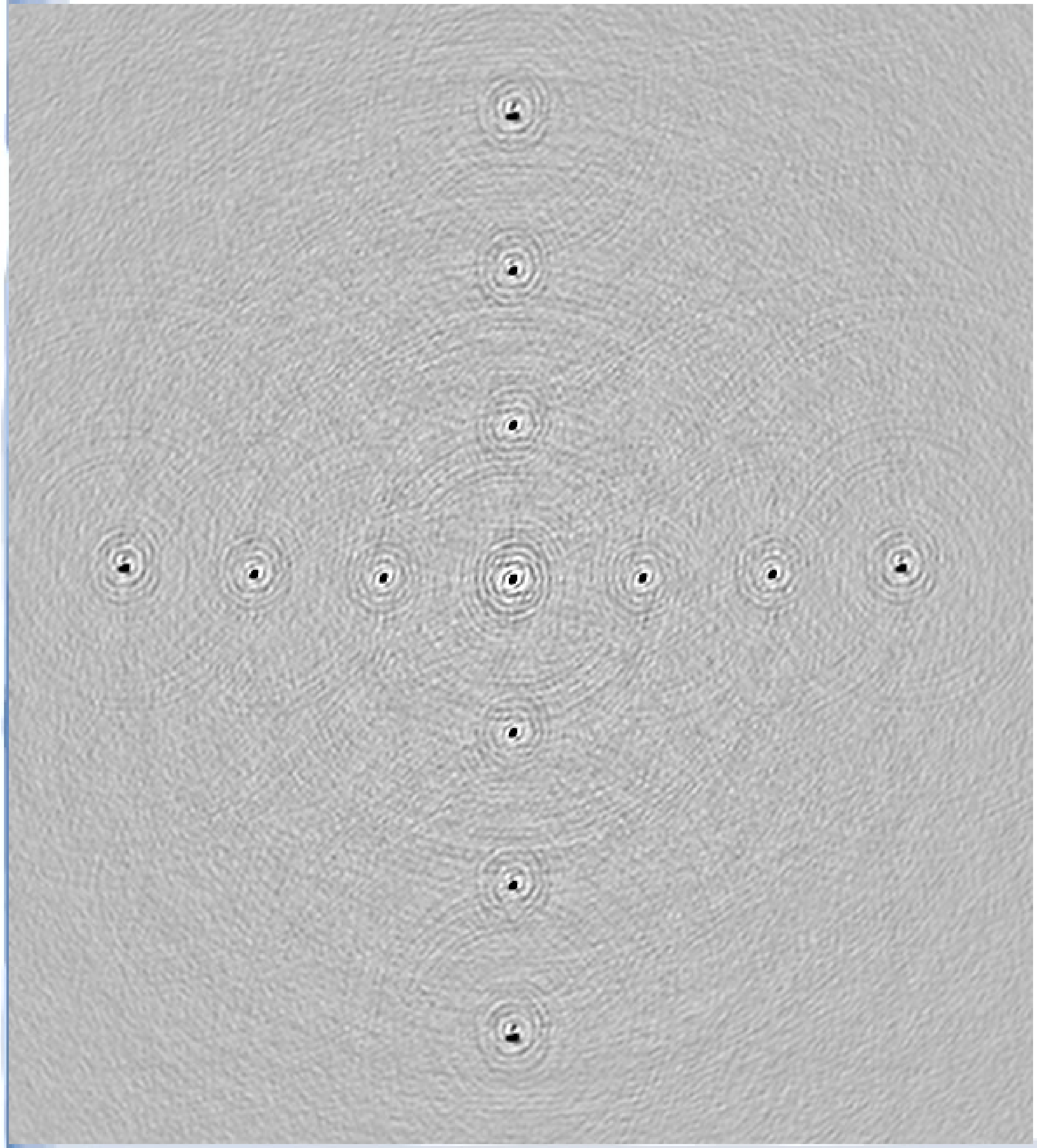


Wproj Convolution function

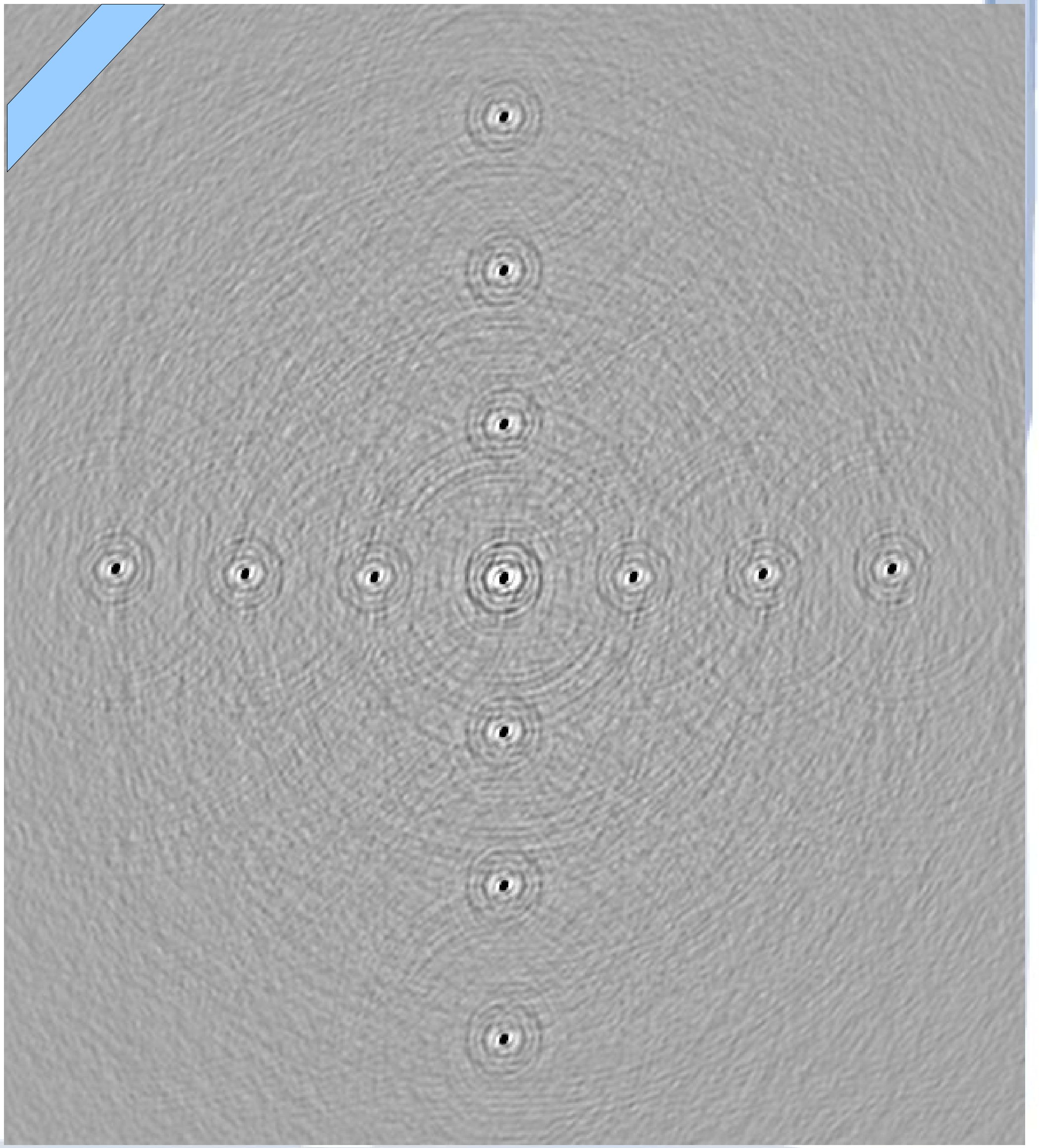




Results (Simulation)

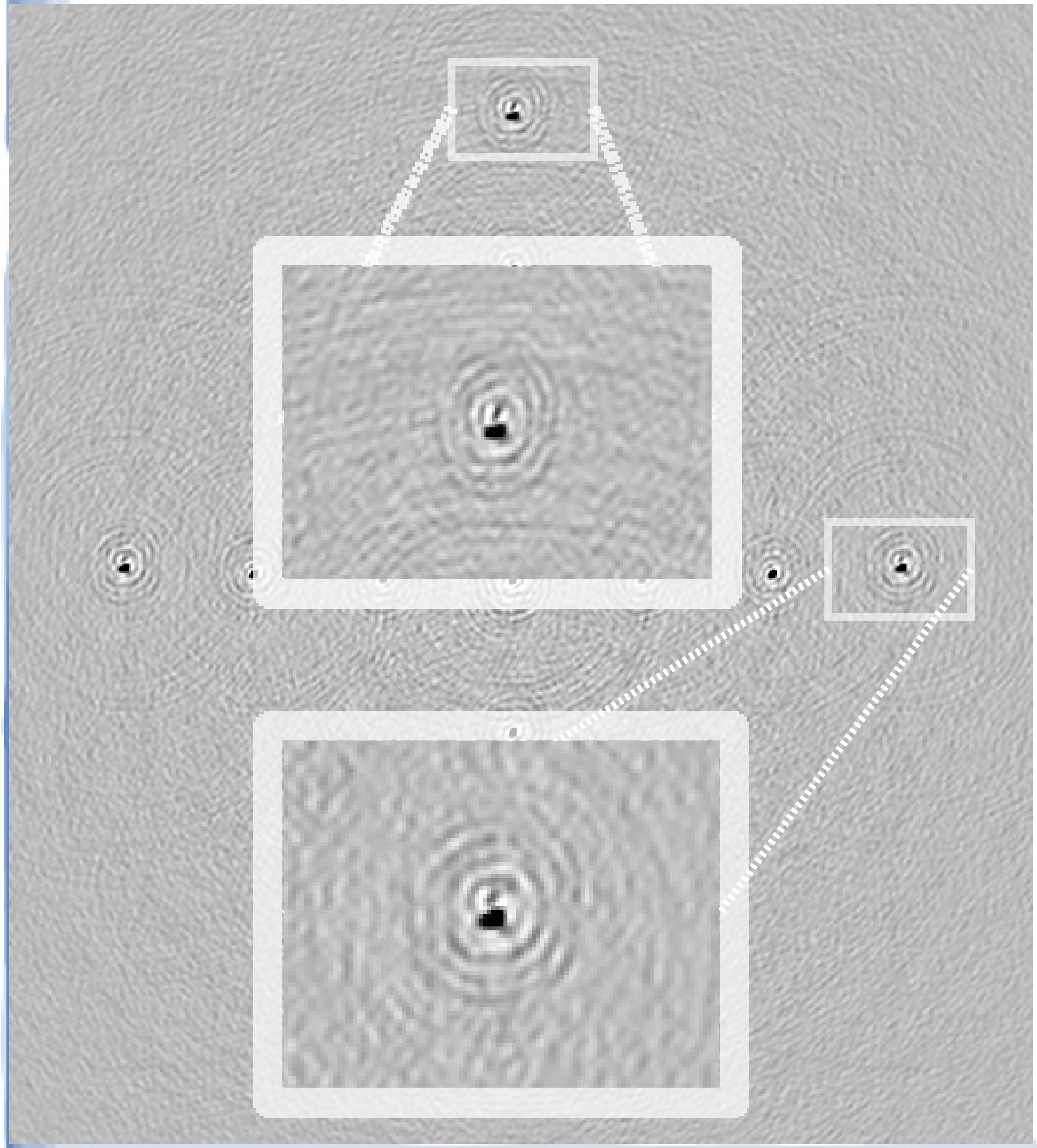


Normal Imaging

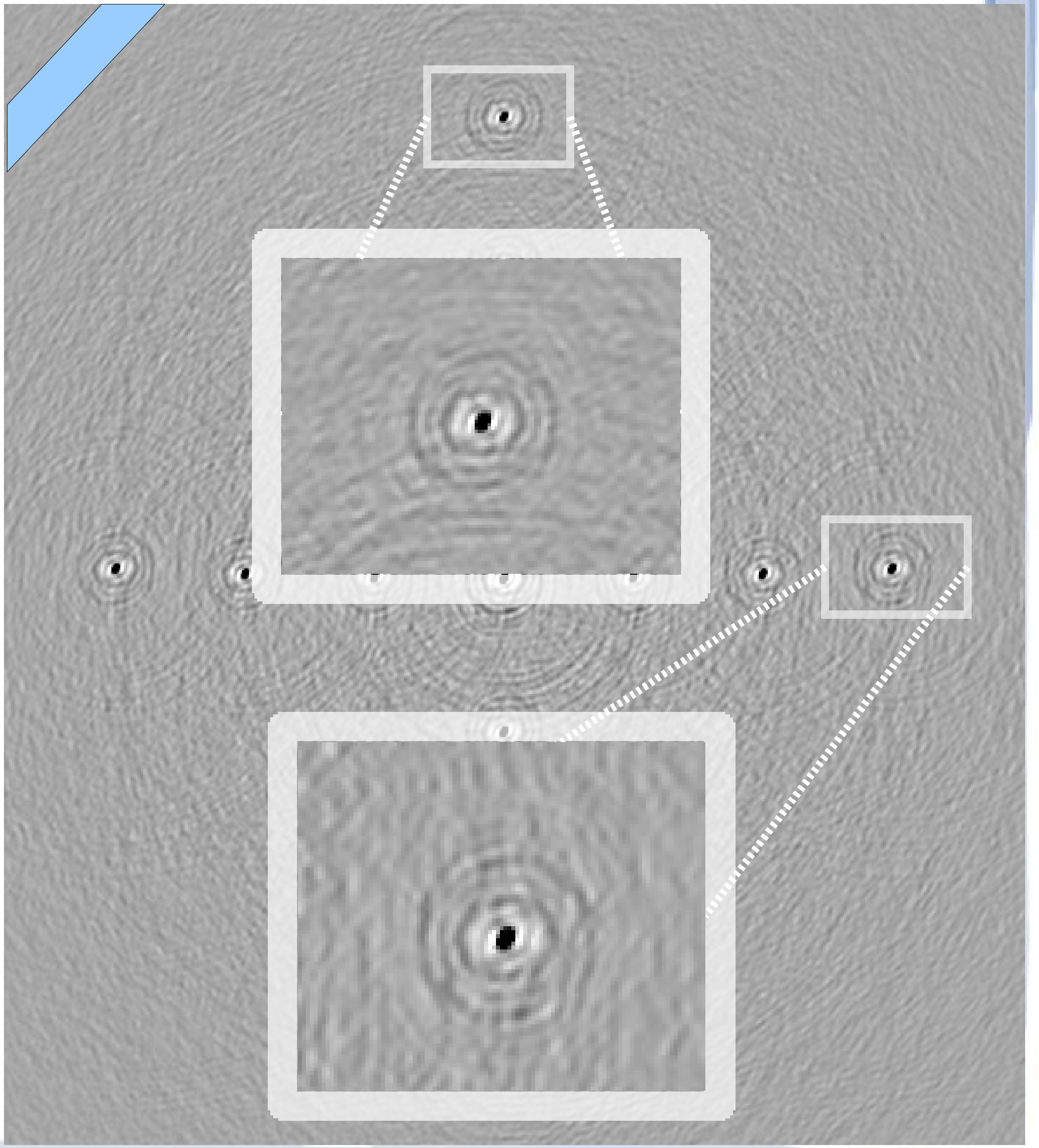


W-Projected

Results (Simulation)

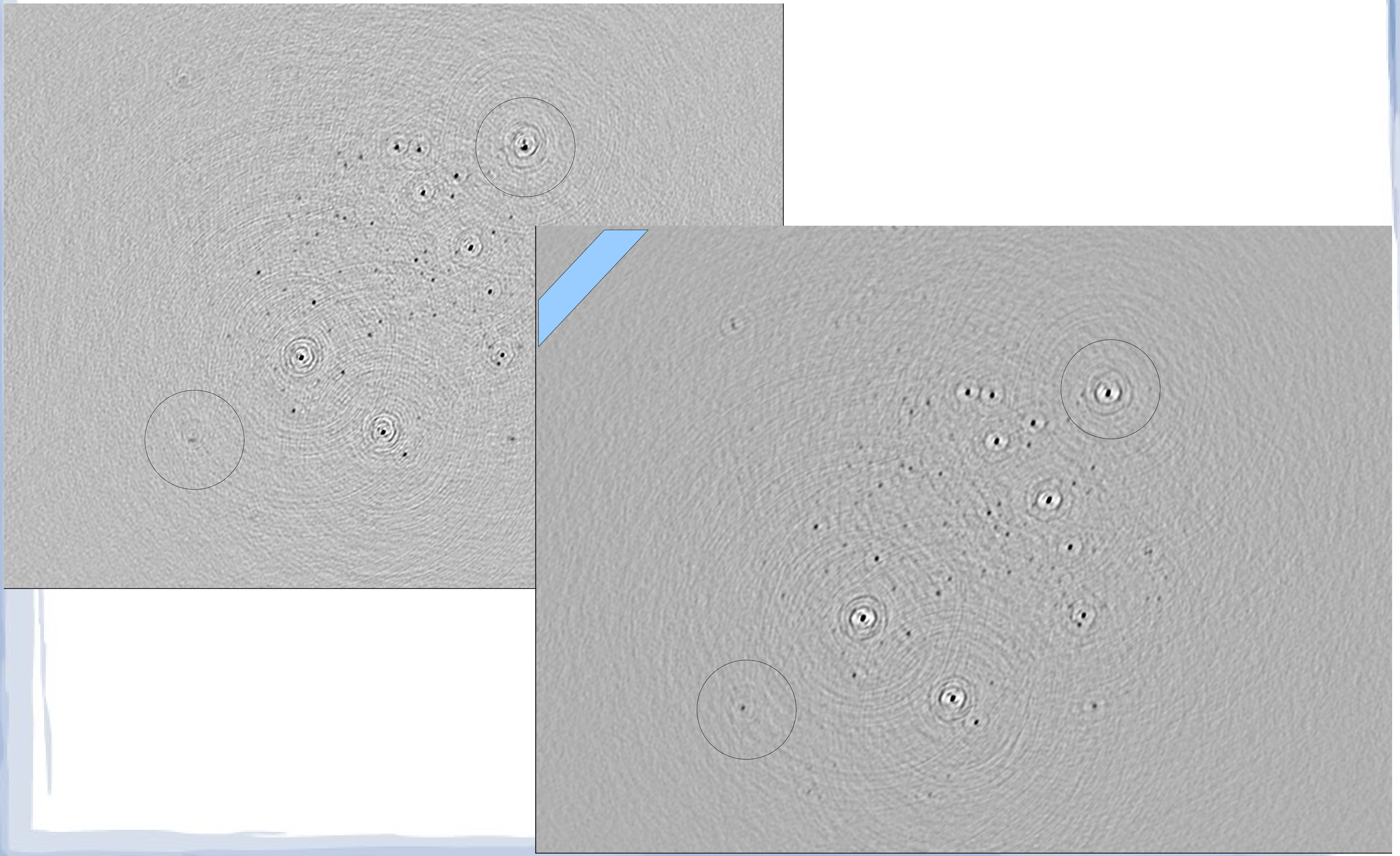


Normal Imaging

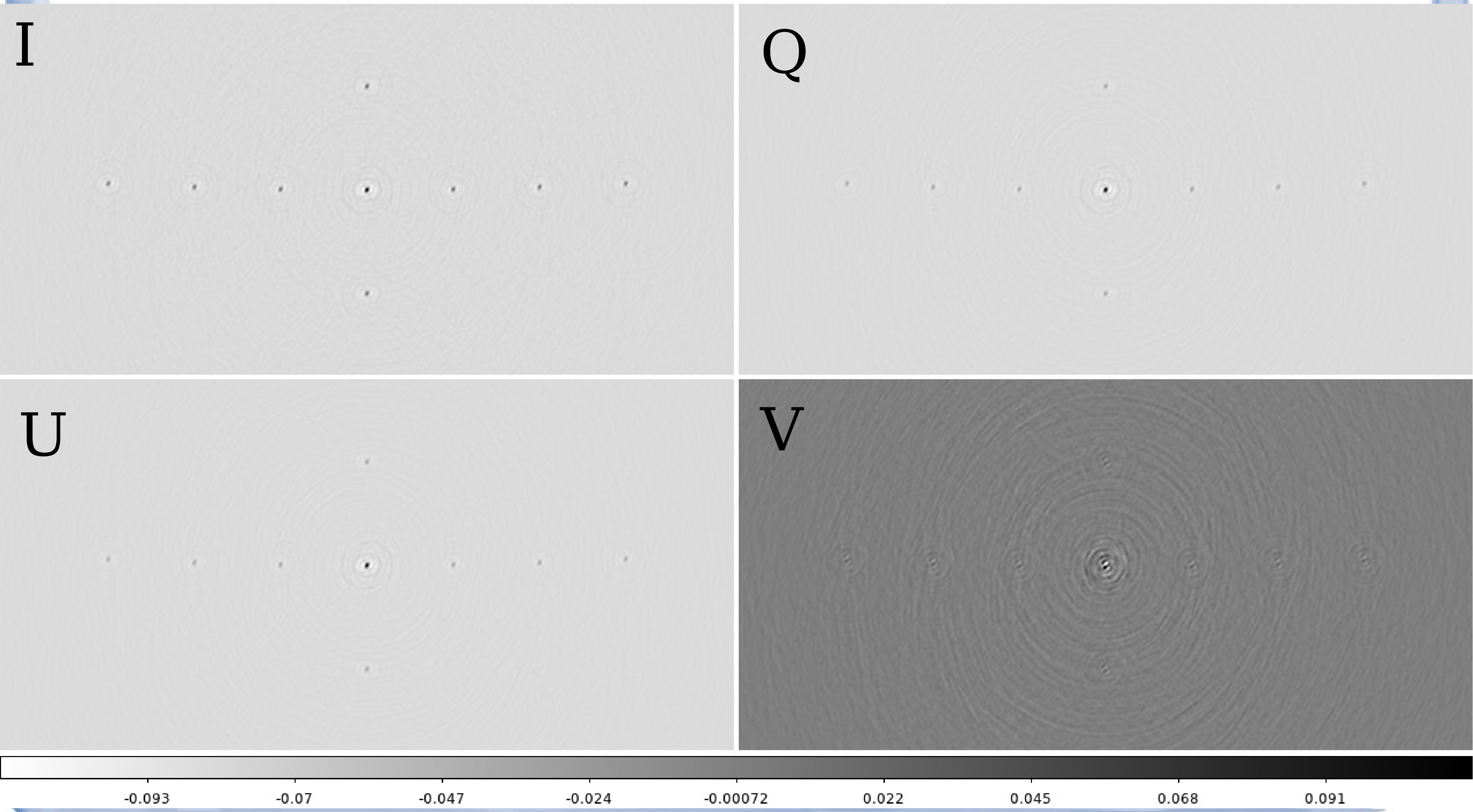


W-Projected

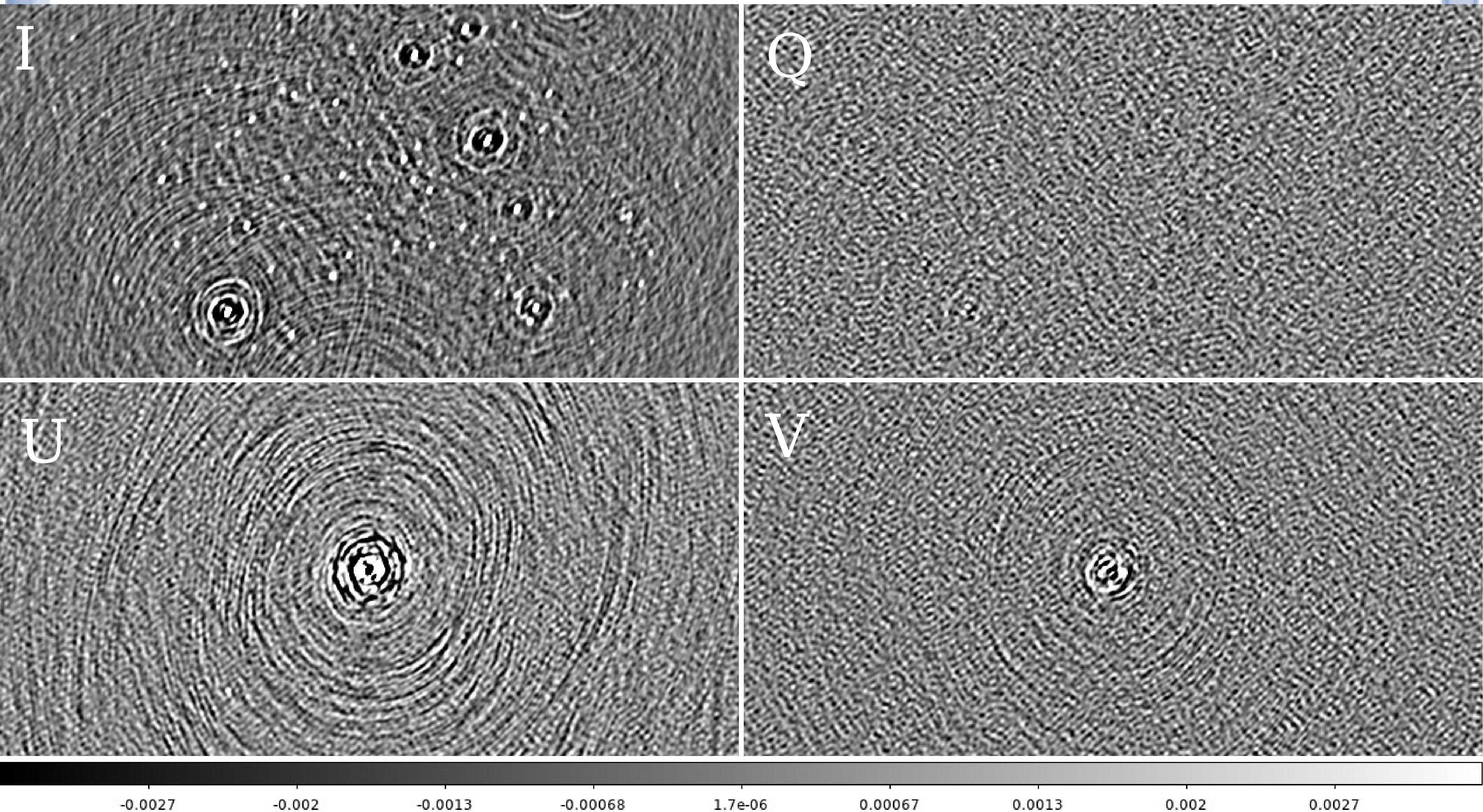
3C196 W-Corrected



Full Stokes (Sim)



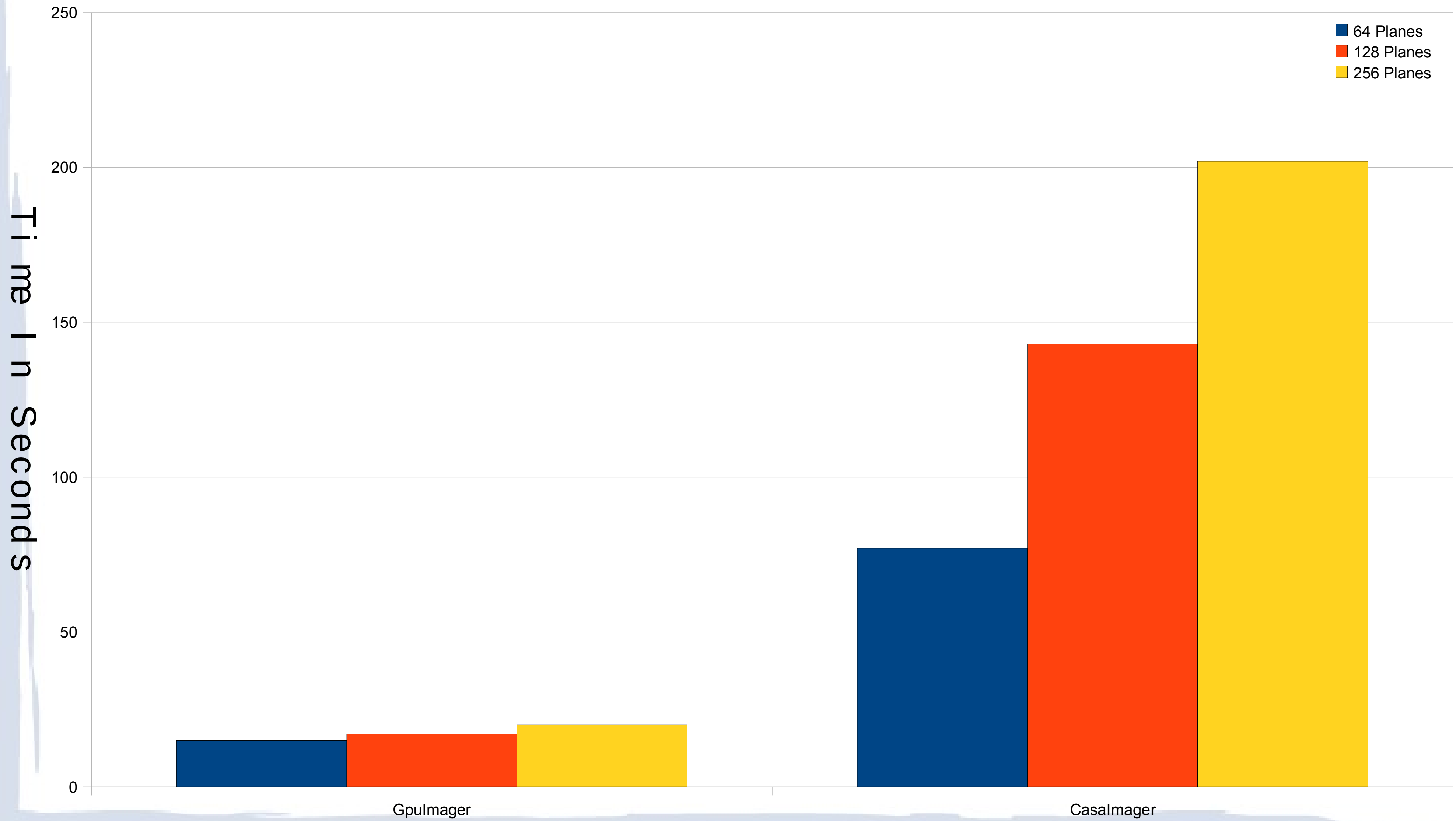
Full Stokes 3C196



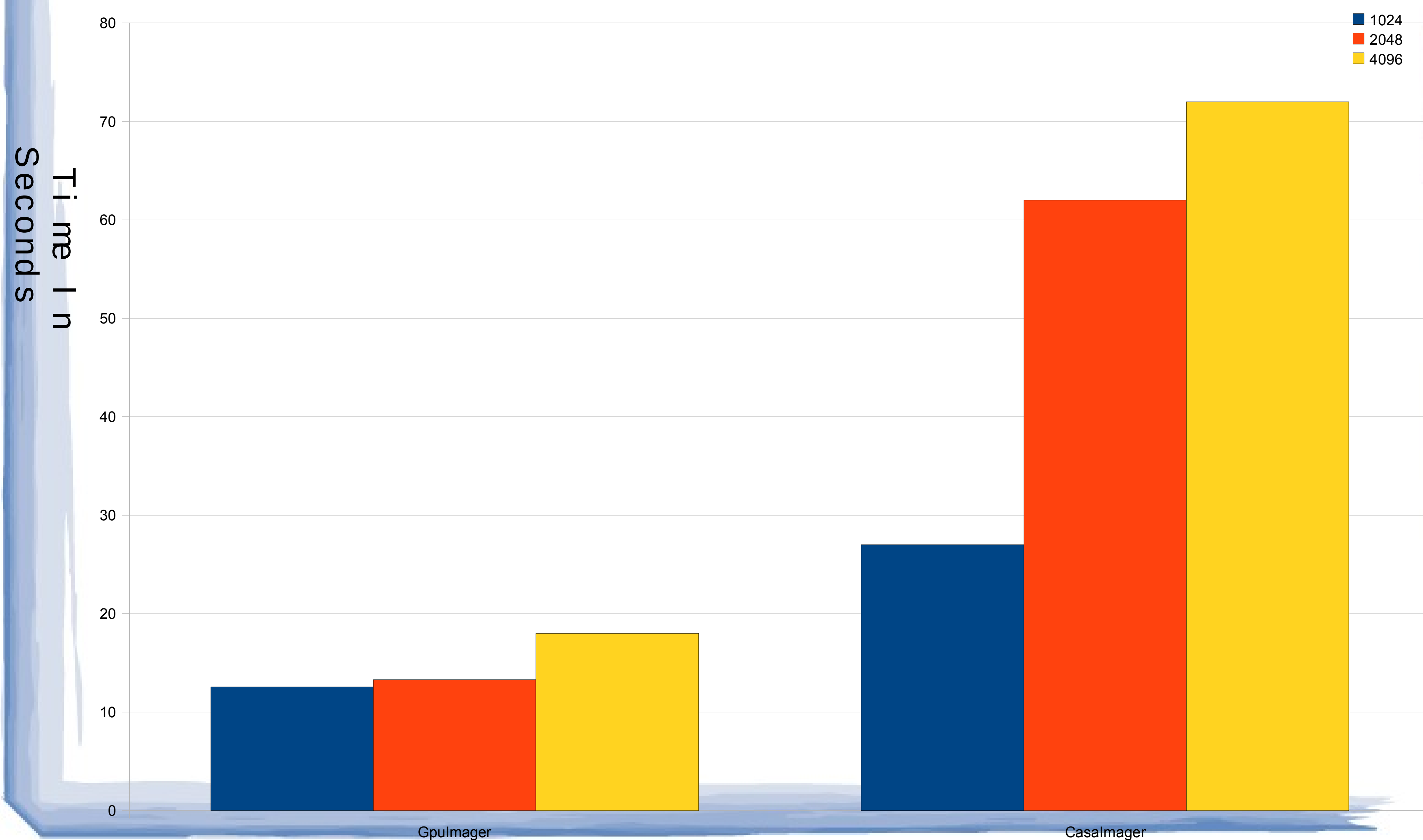
LOFAR Eor GPU Cluster

- Configuration
  - 2 x Tesla M1060 (GPU)
    - Computing capability 1.3
    - 240 Processing cores @ 602 MHz
    - 4GB Memory
  - 16 x Intel(R) Xeon(R) CPU E5520 2.27GHz
  - 12 GB Ram
  - 80 nodes

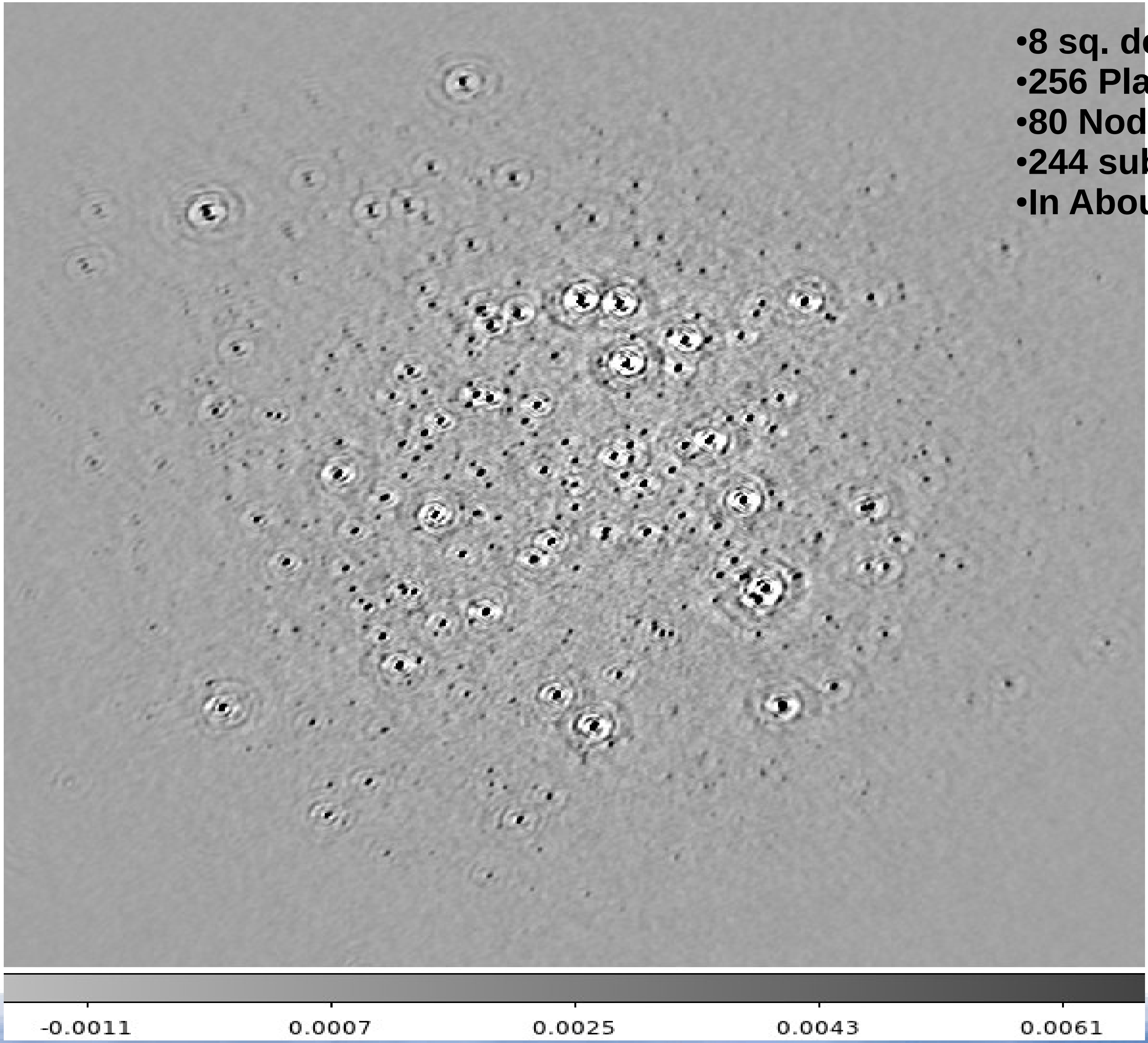
W-Projection Performance



Without W-Projection



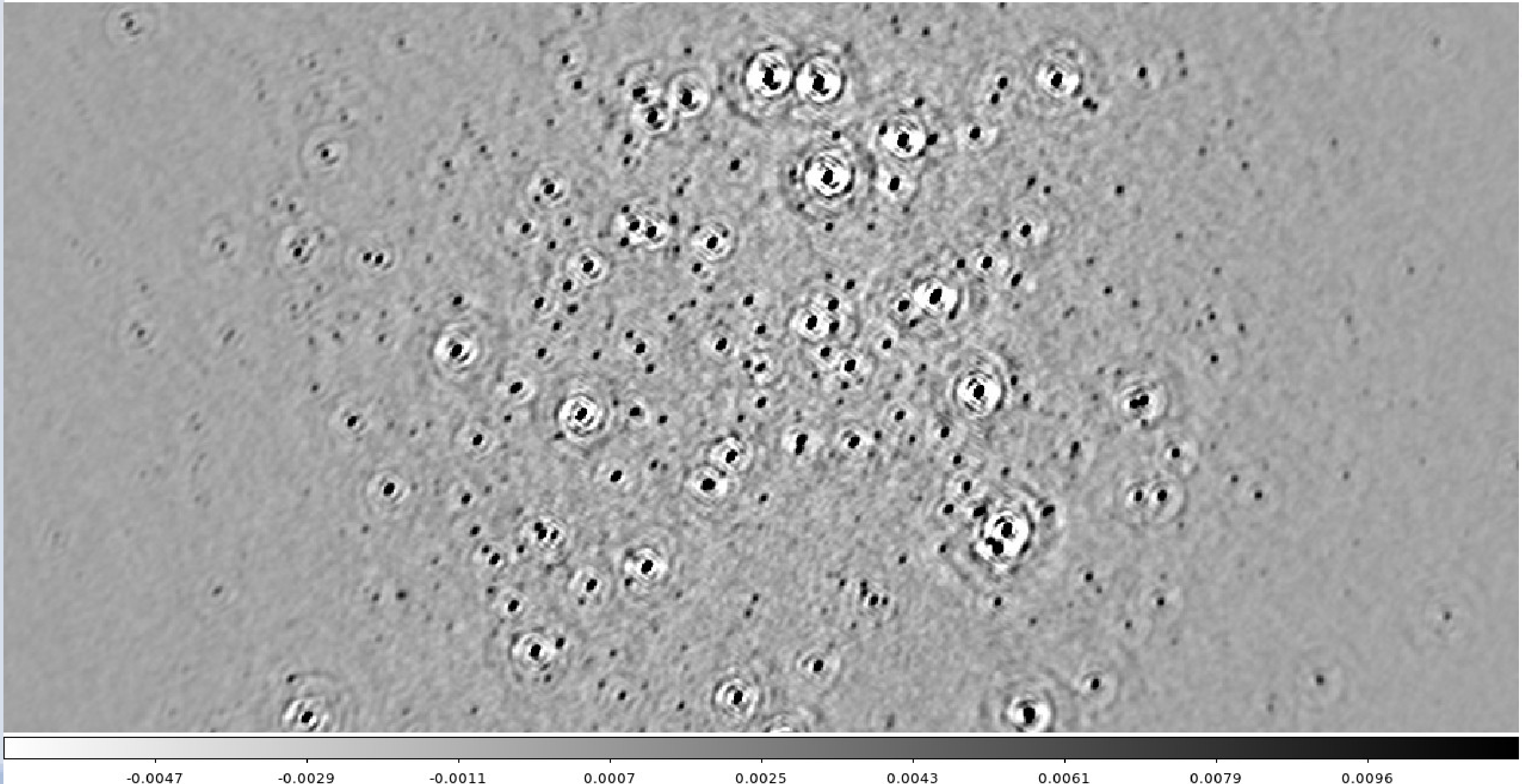
3C196



- 8 sq. degrees
- 256 Planes
- 80 Nodes
- 244 sub-bands
- In About 5 mins

3C196 (zoomed)

0.5 mJy rms noise at the edge of the image



Future Plans

- We are working on implementation of MVDR on the gpu.
- Experimenting with Weighting and Convolution schemes in the UV plane.
- Incorporating a very fast calibration scheme into the imager so both are done together

Thank You

Introduction

- What is a GPU?
- Why another Imager?
  - Large amount of data to be processed.
  - W-projection is a requirement.
- Issues to Consider
  - Wide field imaging
  - Full Stokes
  - Speed
  - Experiment with New approaches

Why use GPU

- Hundreds of Processors
- Very well suited for matrix operations.
- Well suited for parallel tasks.
- With the latest APIs (CUDA, OpenCL) easy to program.

Basic Imaging Equation

$$V(u,v,w)=\int \frac{I(l,m)B(l,m)}{\sqrt{1-l^2-m^2}} e^{-2\pi i(ul+vm+w(\sqrt{1-l^2-m^2}-1))} dl\, dm$$

- In general W~0 (U,V,W=0).
- This implies visibilities = FFT(image)
- But this is not always the case hence W-Projection. (T.J.Cornwell et all)

W-Projection

$$V(u,v,w)=\int \frac{I(l,m)B(l,m)}{\sqrt{1-l^2-m^2}} e^{-2\pi i(ul+vm+w(\sqrt{1-l^2-m^2}-1))} dl\,dm$$

Assuming W is constant for a given plane of UV values

$$V(u,v,w)=\int e^{-2\pi i[w(\sqrt{1-l^2-m^2}-1)]} \frac{I(l,m)B(l,m)}{\sqrt{1-l^2-m^2}} e^{-2\pi i(ul+vm)} dl\,dm$$

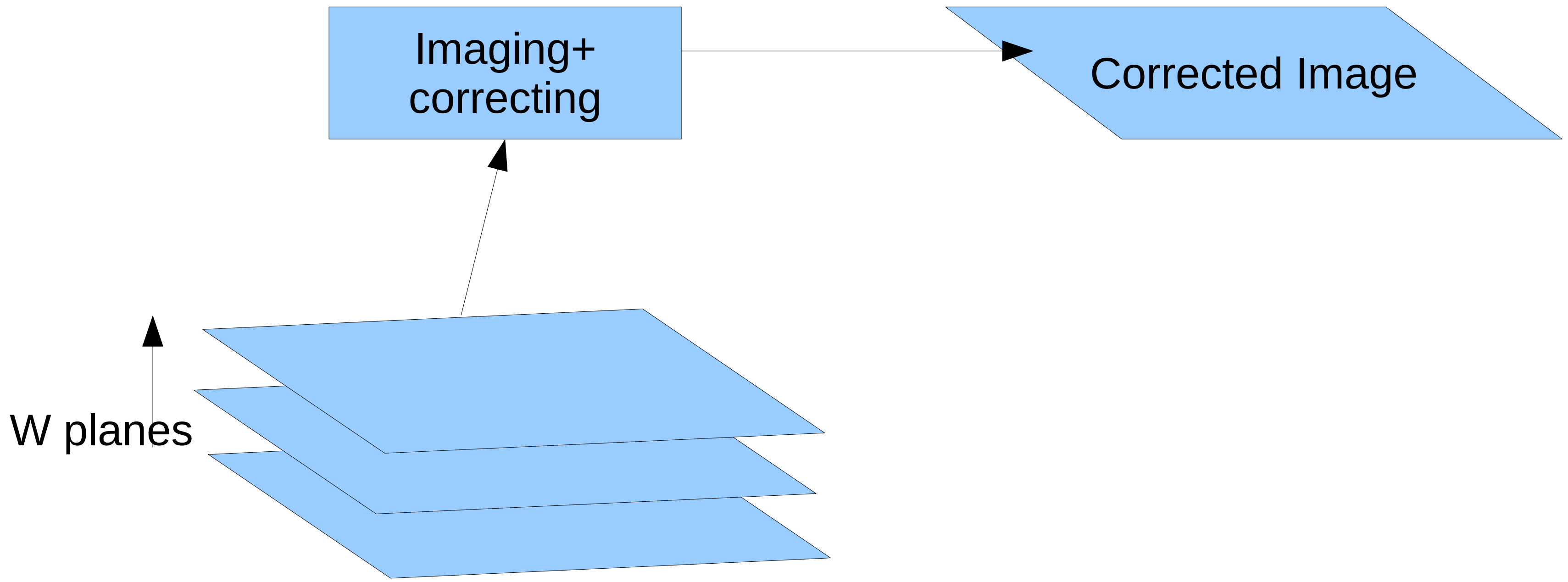
$$G(l,m,w)$$

$$V(u,v,w)=G(l,m,w)V(u,v,w=0)$$

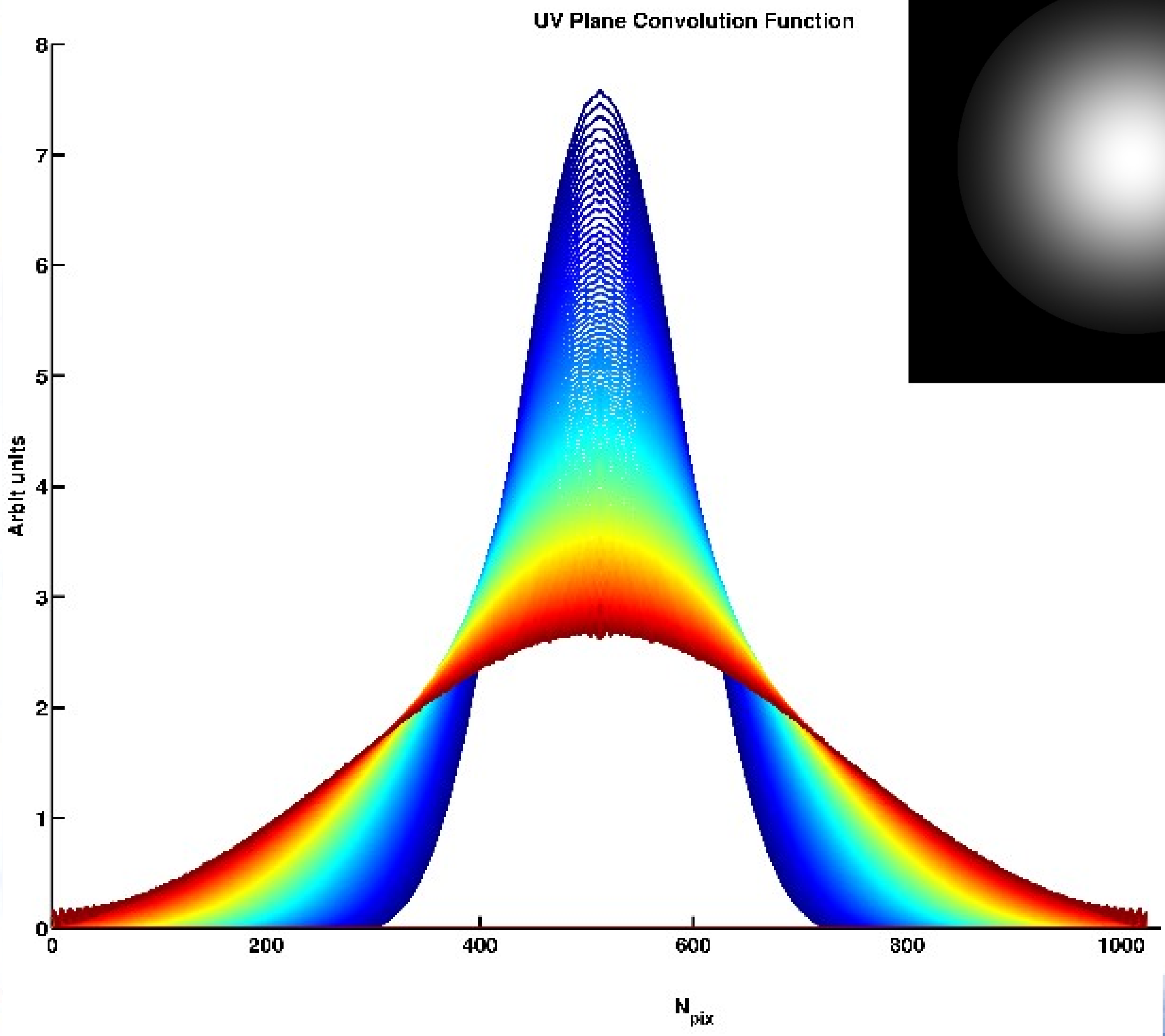
$$IFFT\,V(u,v,w)=G(l,m,w)\times IFFT\,V(u,v,w=0)$$

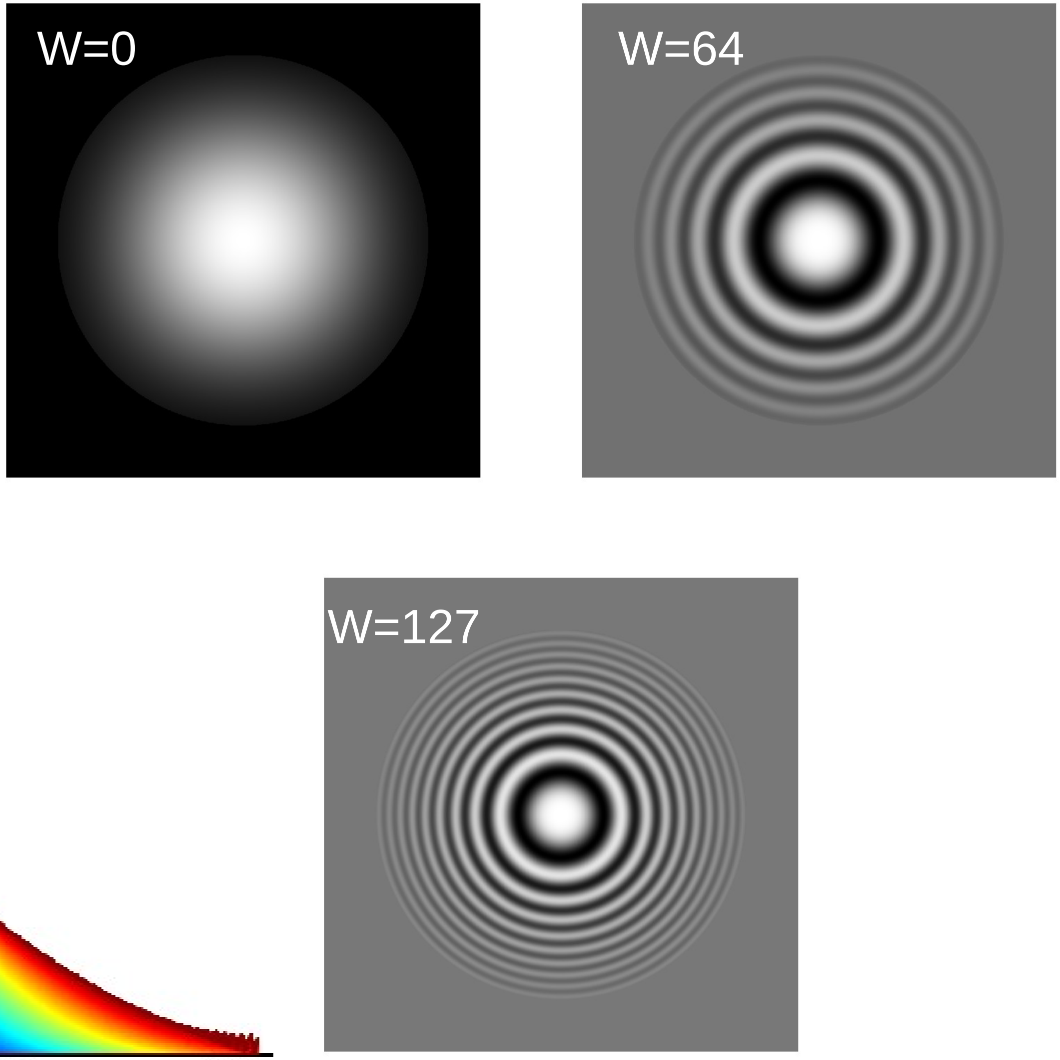
Ref:Tim Cornwell and Liu

Block Diagram

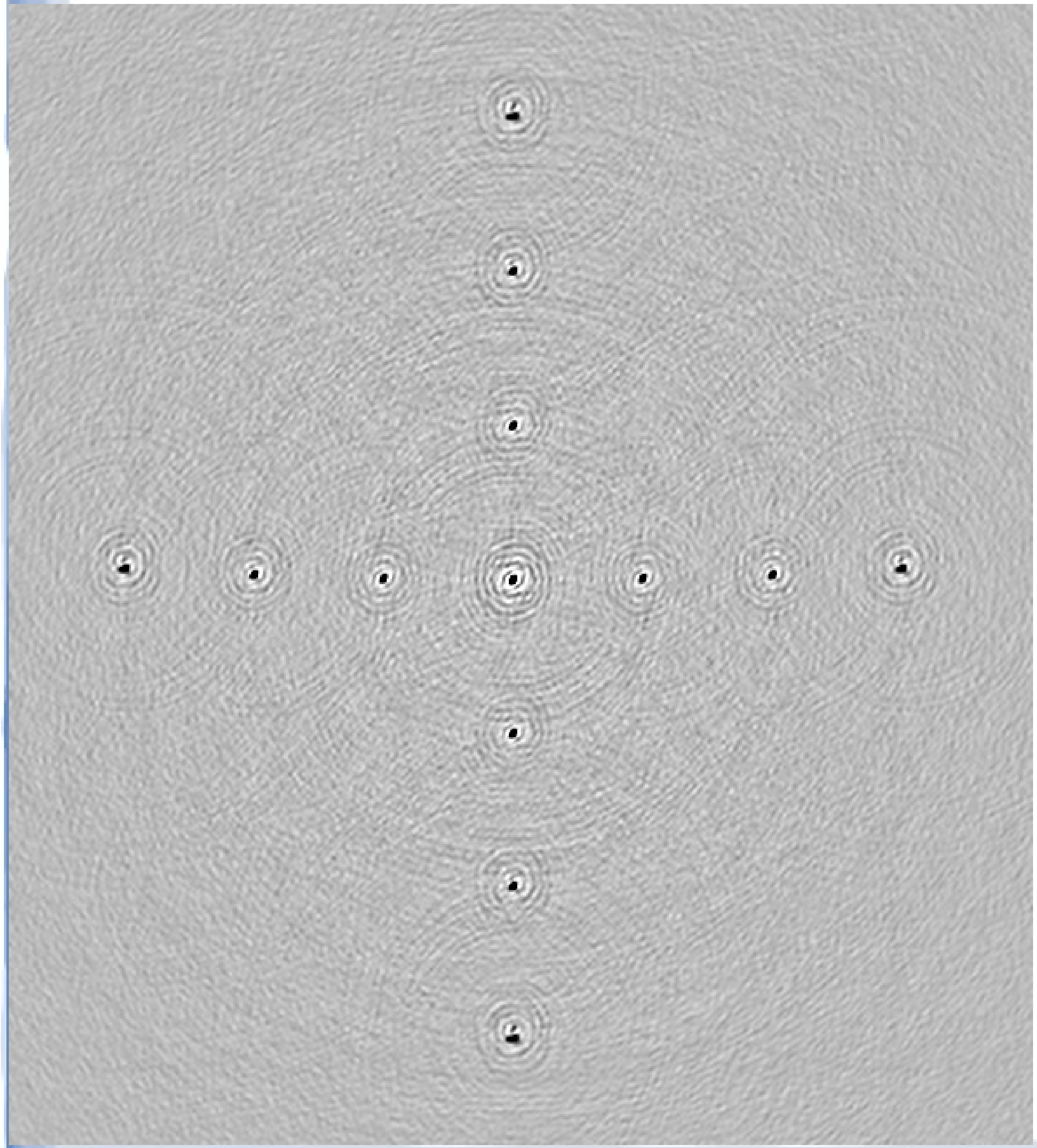


W Correction

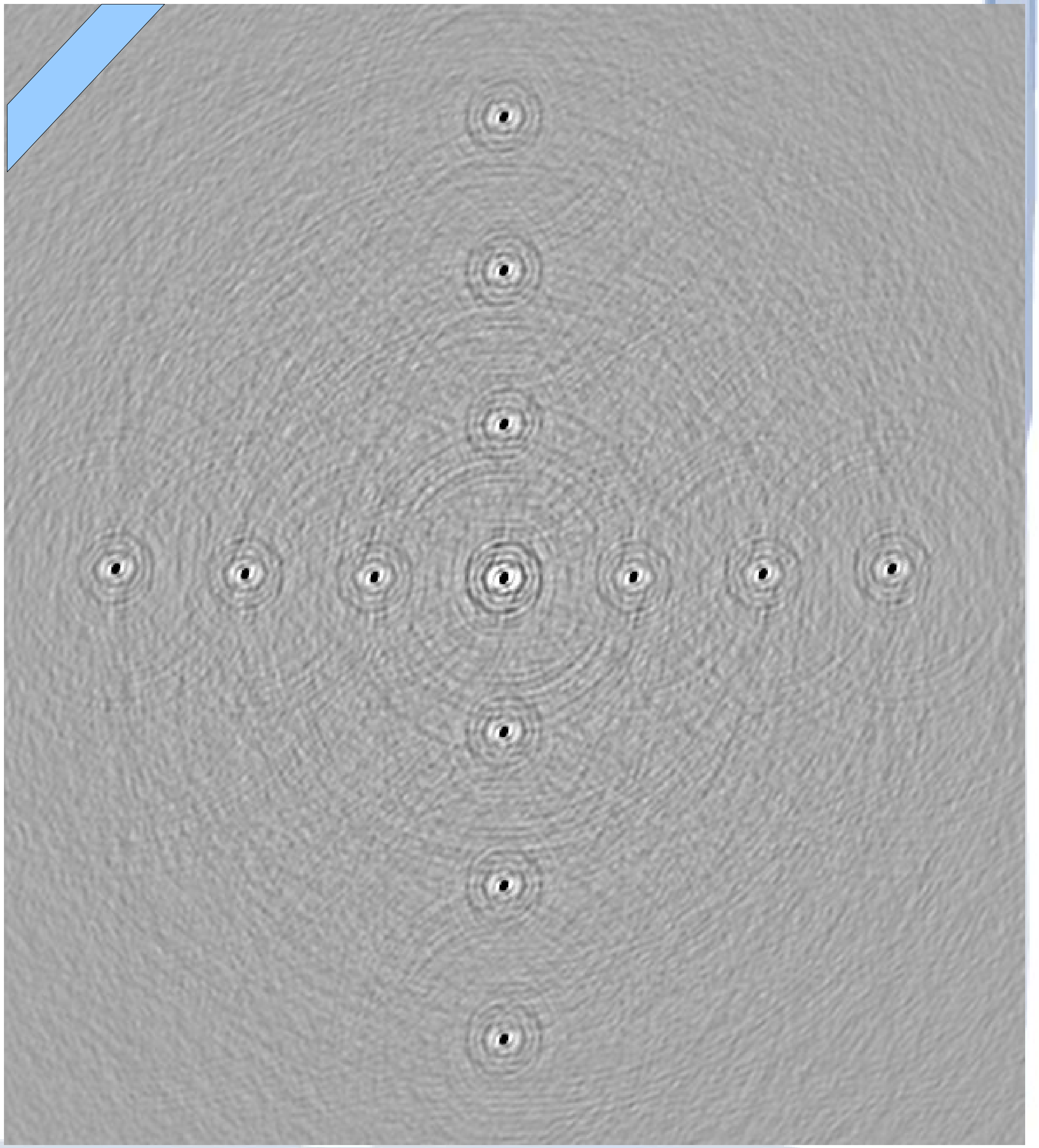




Results (Simulation)

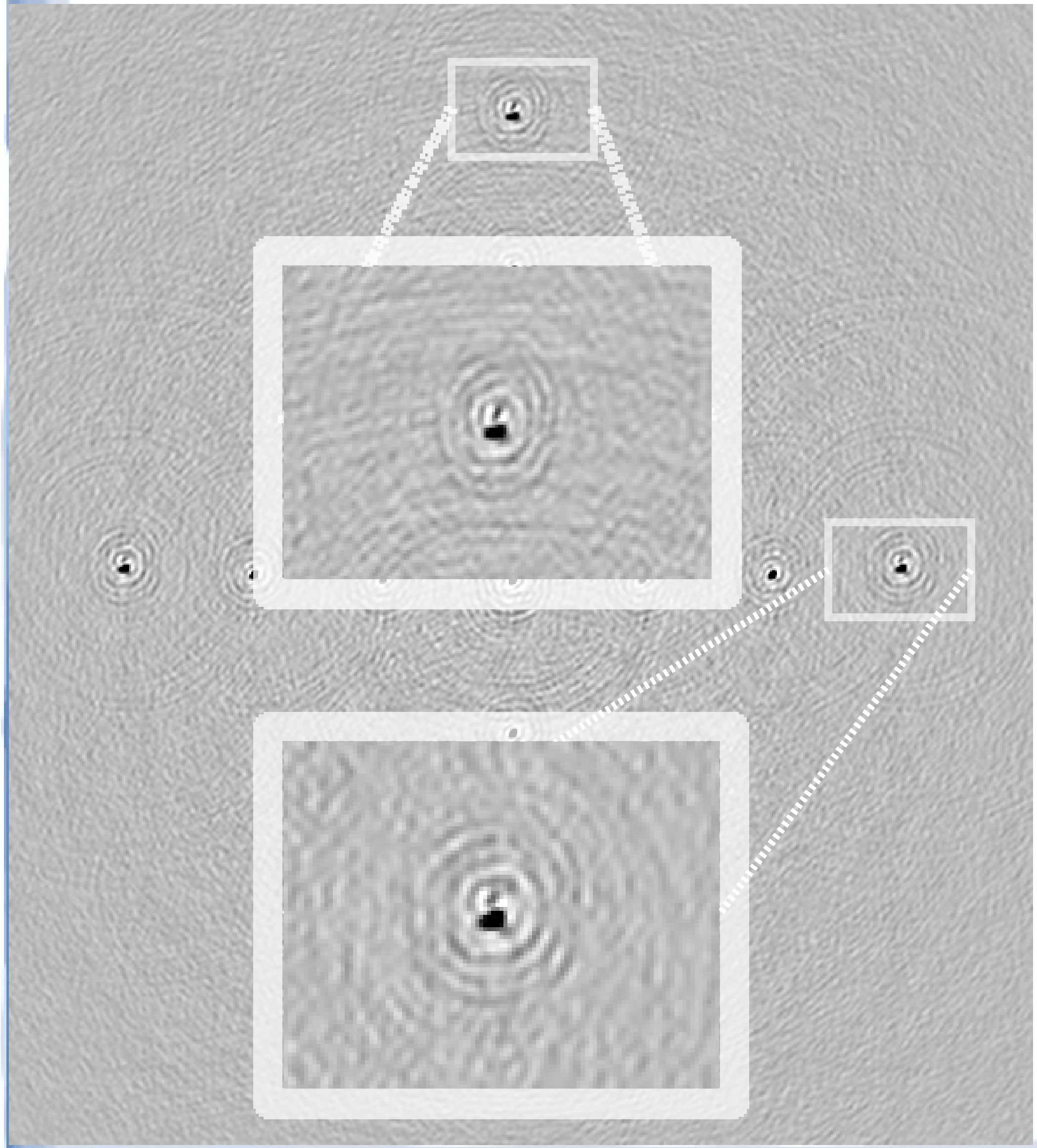


Normal Imaging

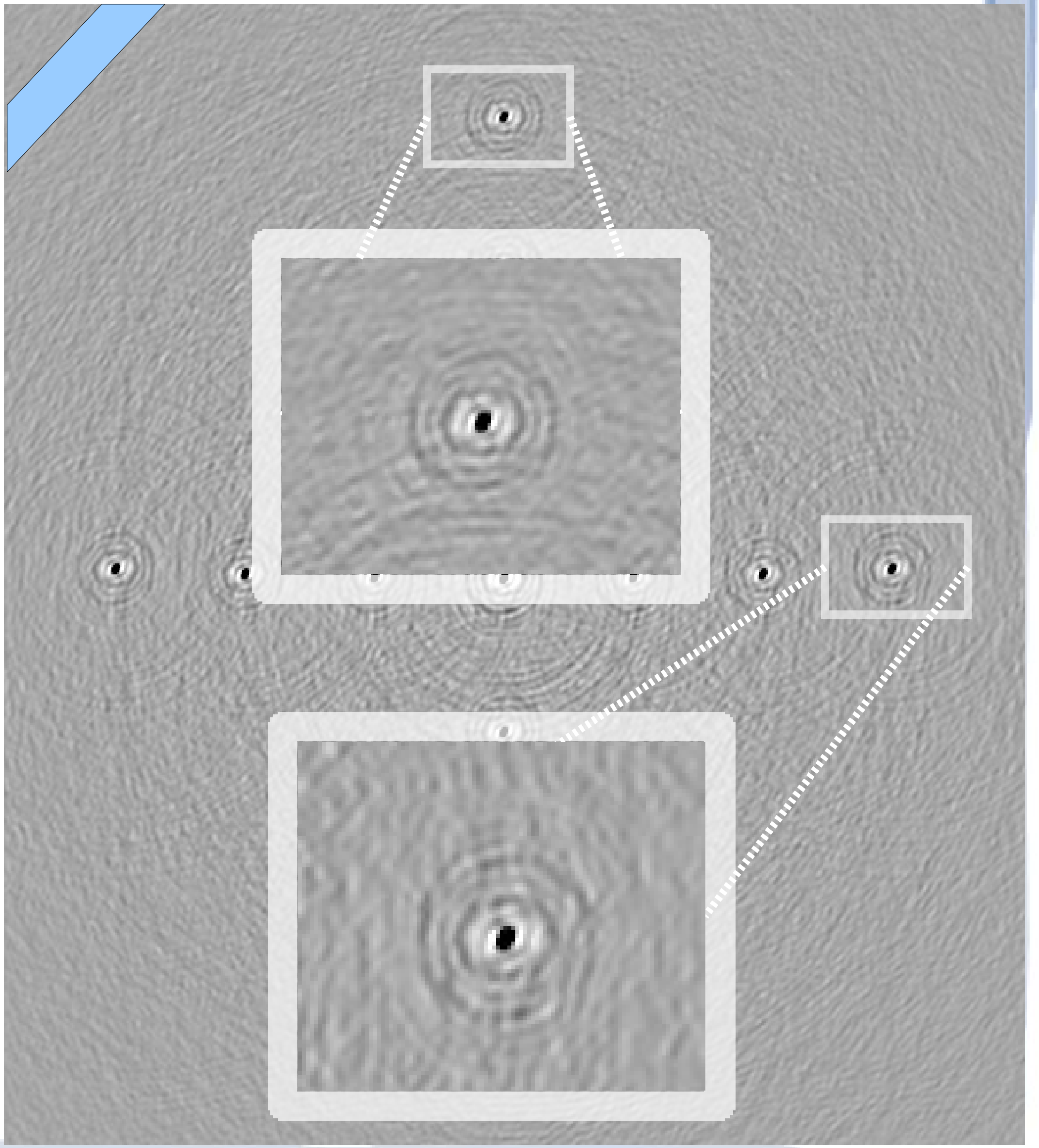


W-Projected

Results (Simulation)

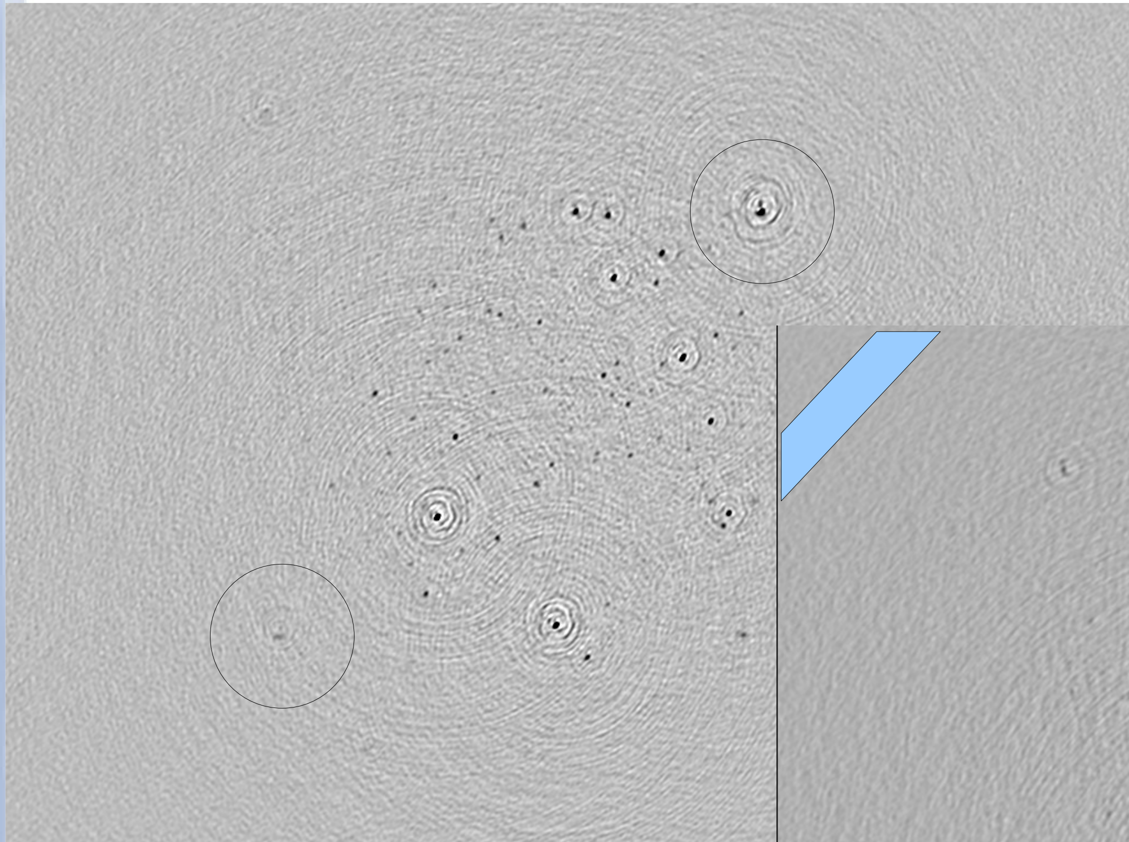


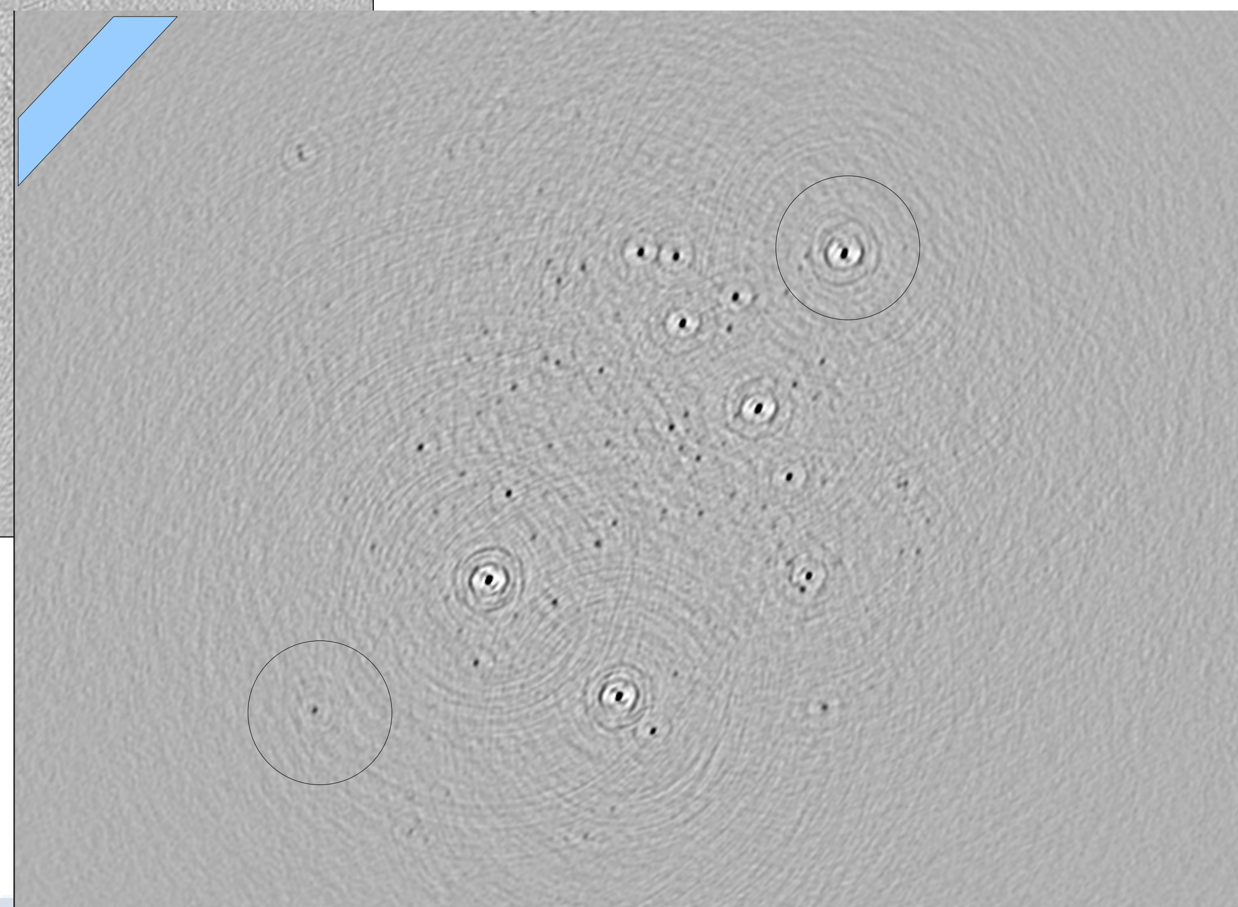
Normal Imaging



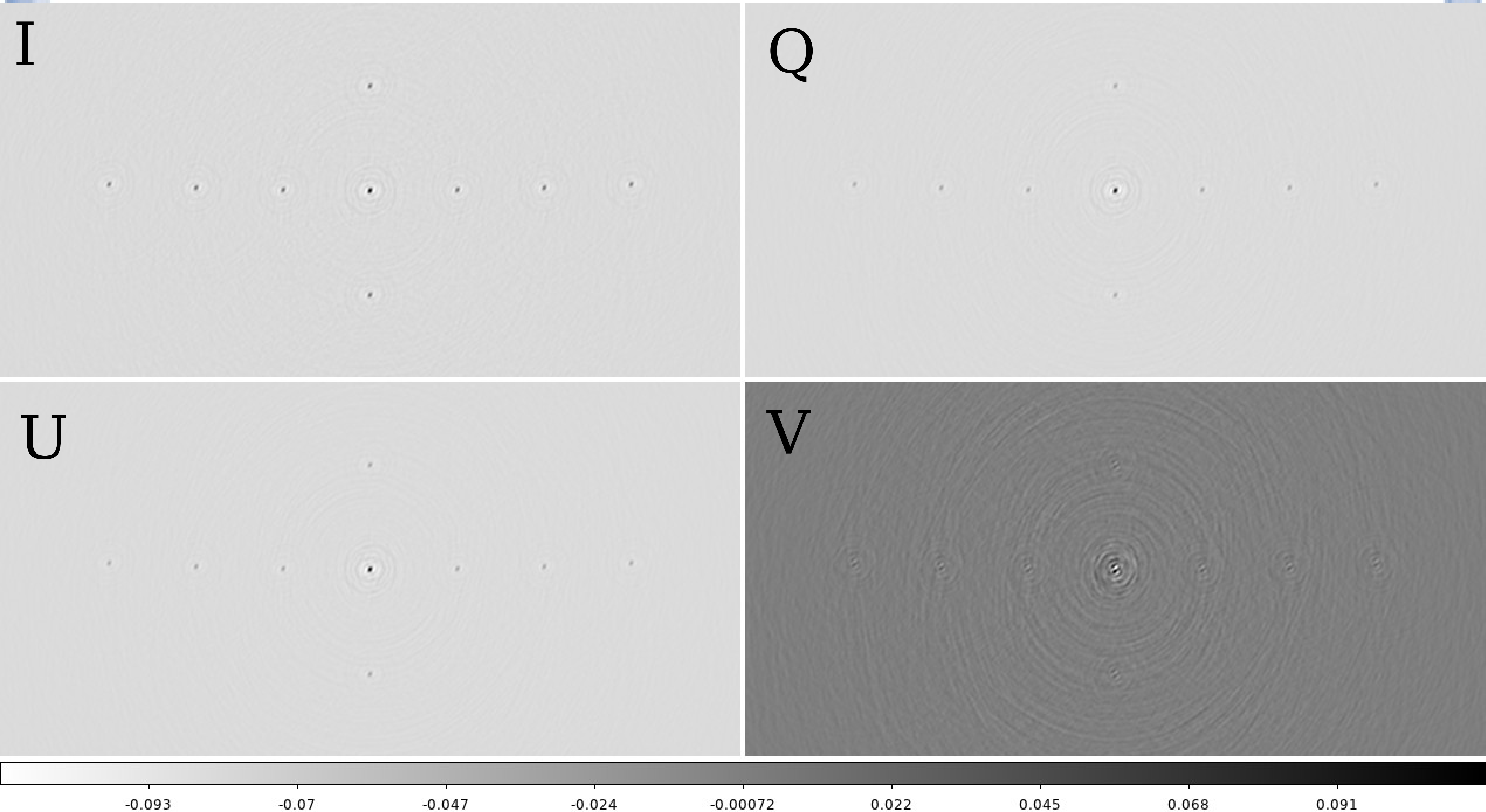
W-Projected

3C196 W-Corrected

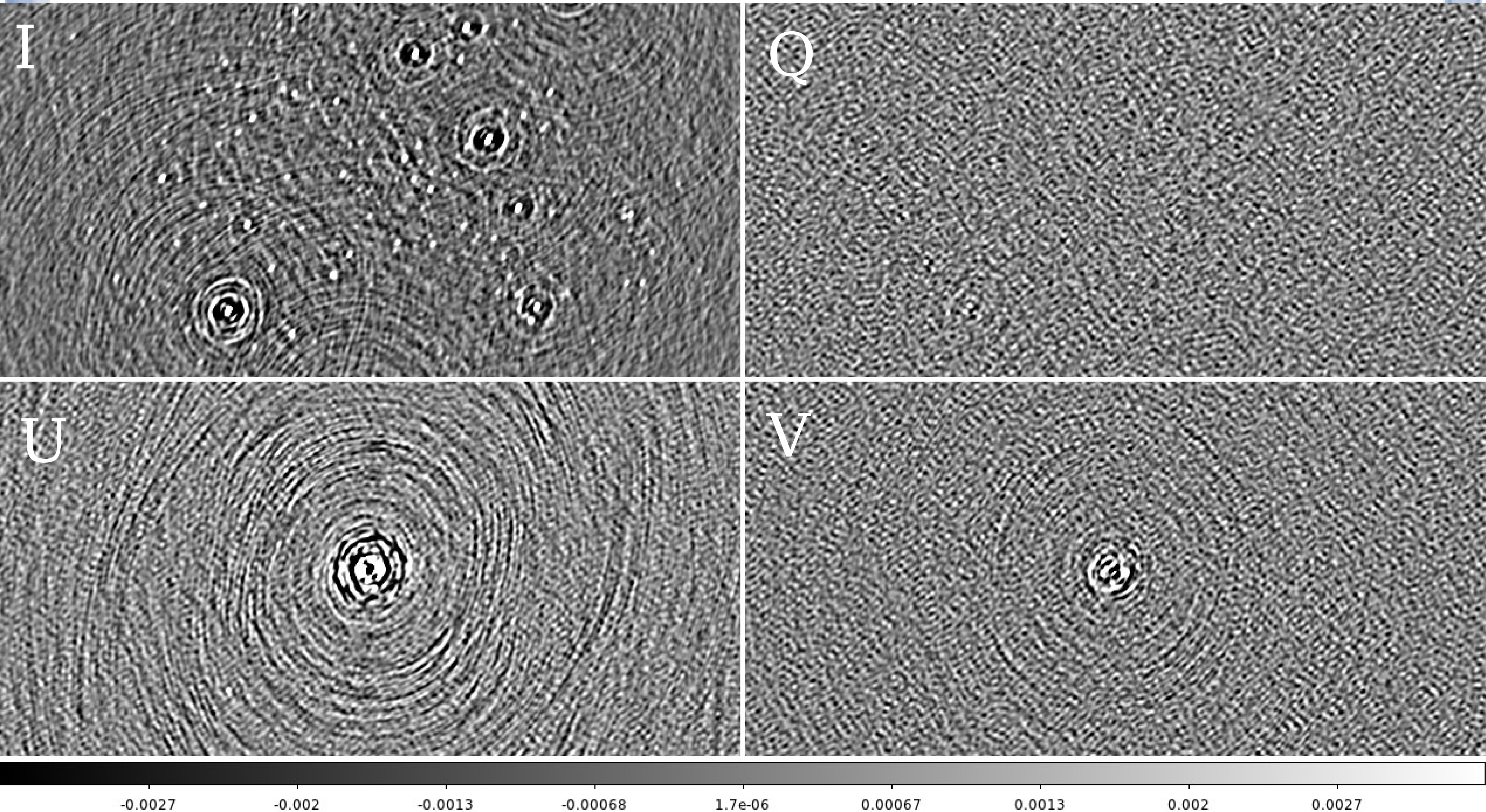




Full Stokes (Sim)



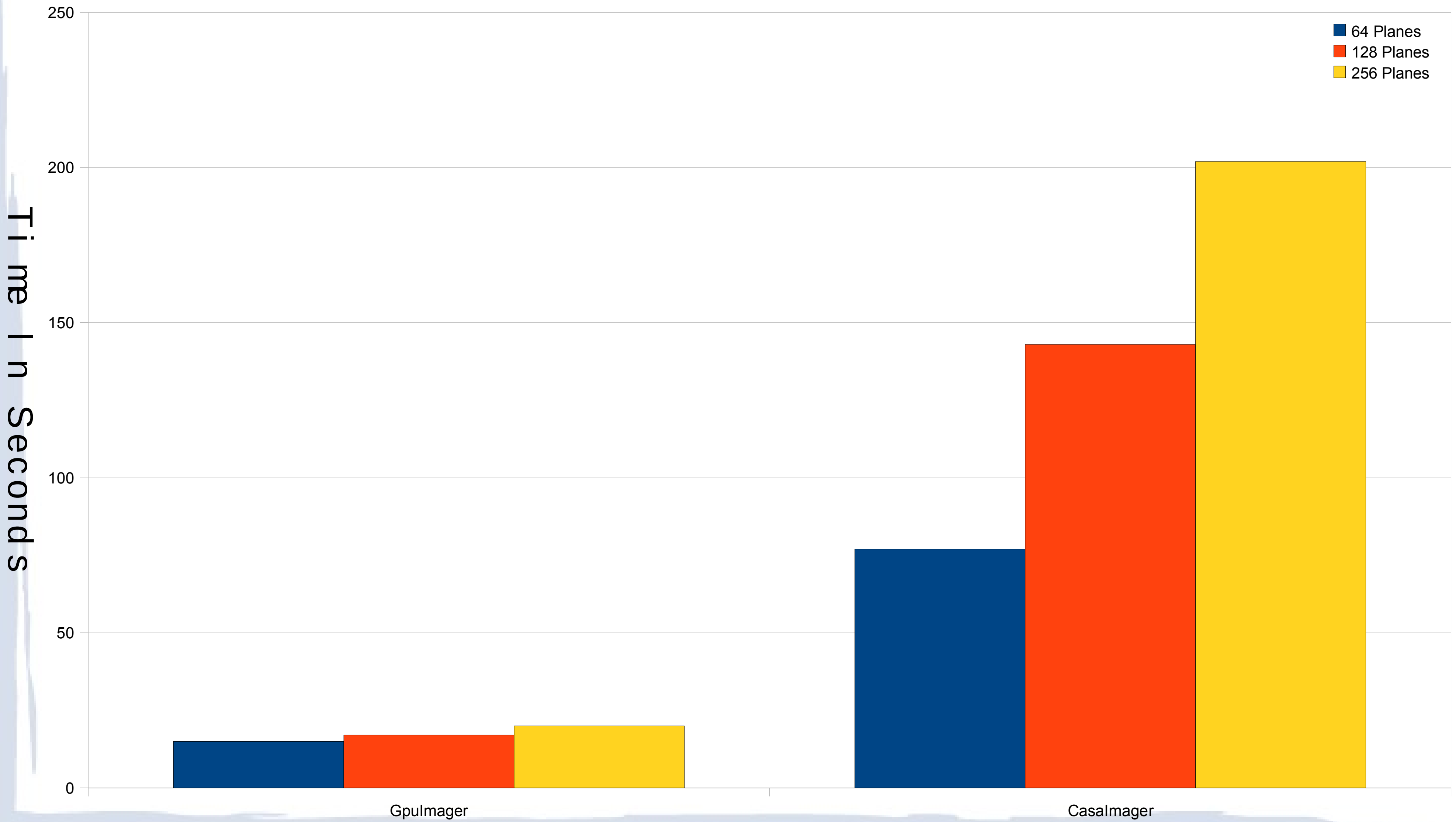
Full Stokes 3C196



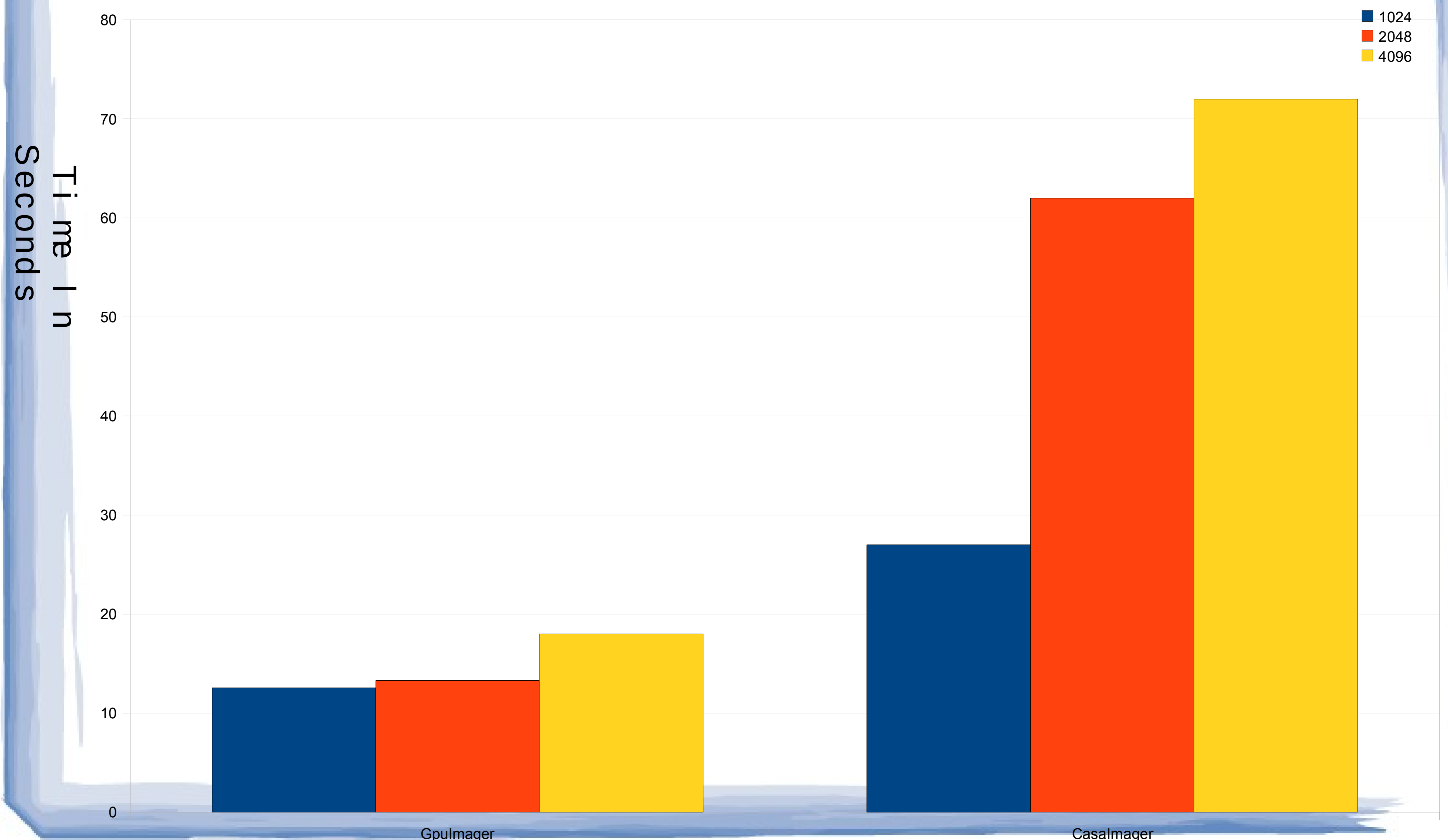
LOFAR Eor GPU Cluster

- Configuration
  - 2 x Tesla M1060 (GPU)
    - Compute 1.3 capable
    - 240 Processors @ 602 MHz
    - 4GB Memory
  - 16 x Intel(R) Xeon(R) CPU E5520 @2.27GHz
  - 12 GB Ram
  - We have 80 Such Nodes

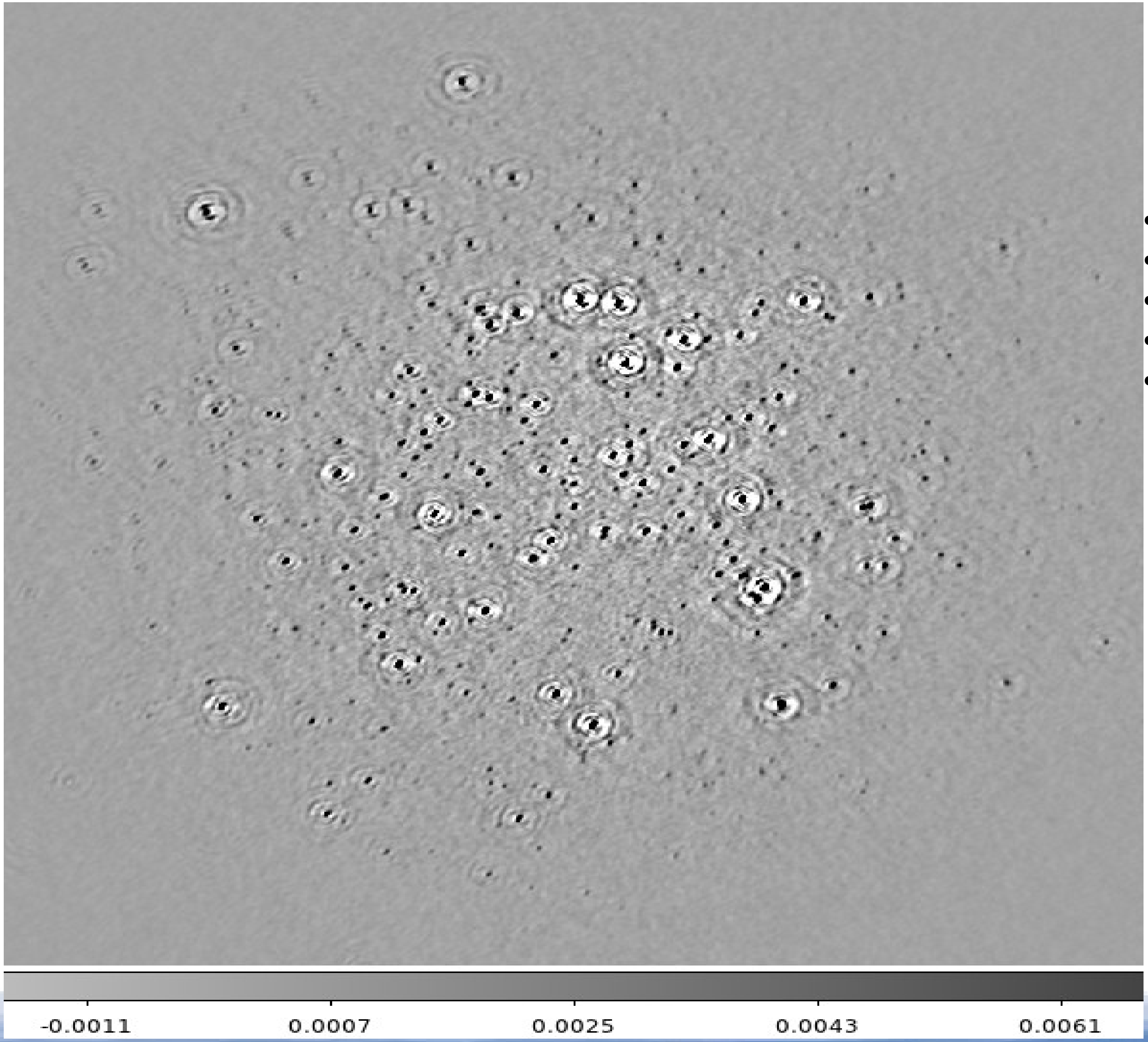
W-Projection Performance



Without W-Projection

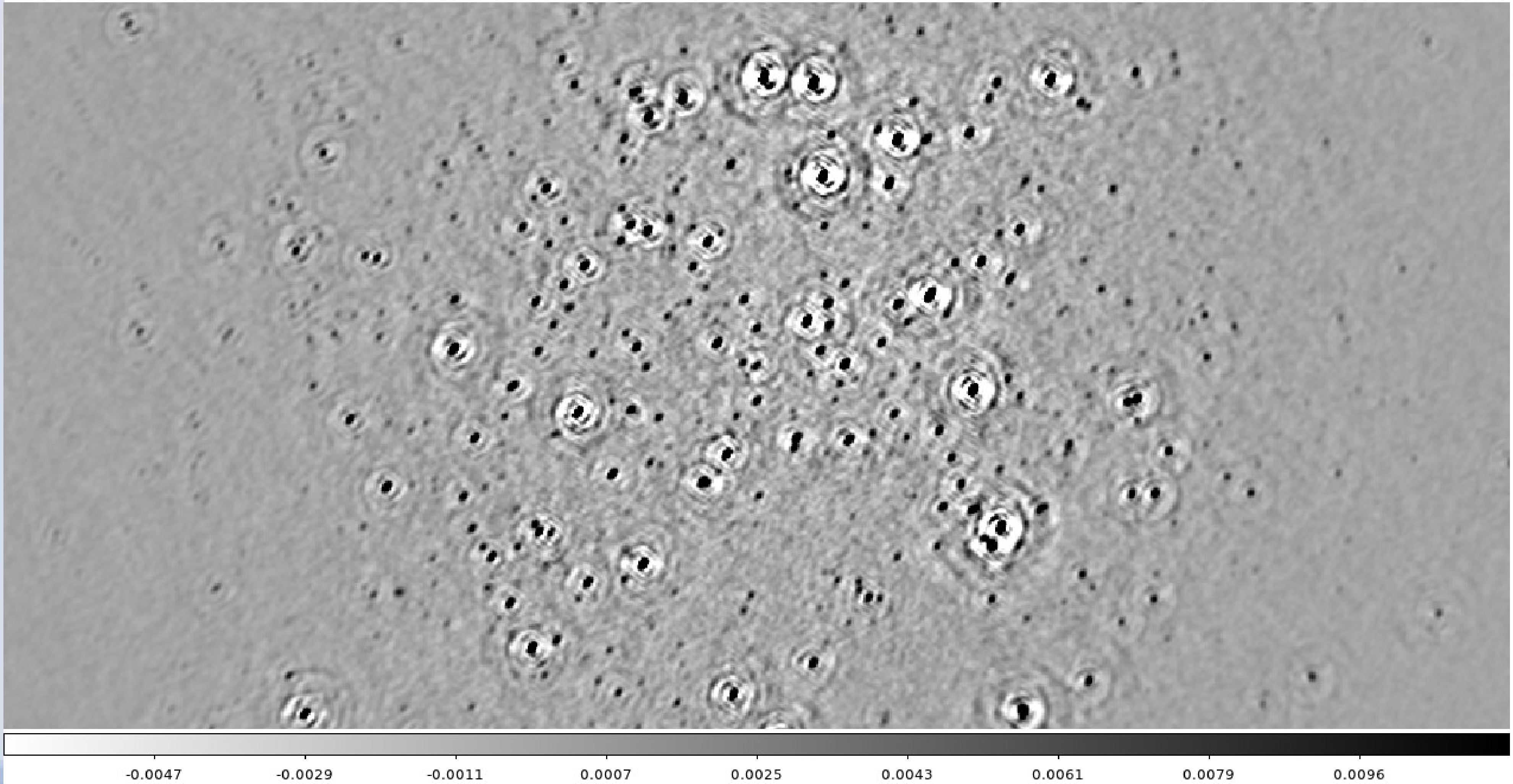


3C196



- GPU Imager
- W-Projection
- 256 Planes
- 80 Nodes
- In About 5 mins

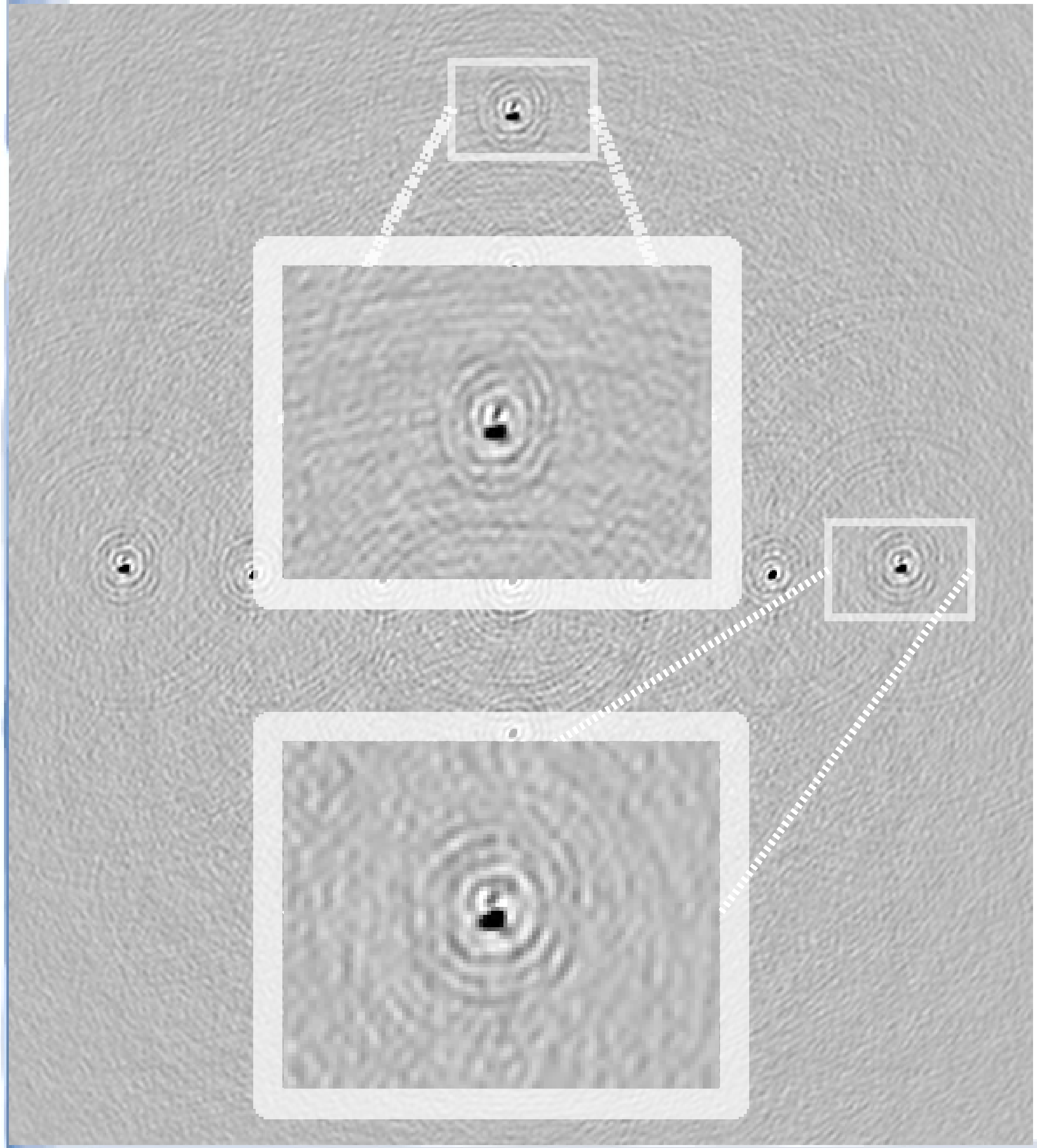
3C196 (zoomed)



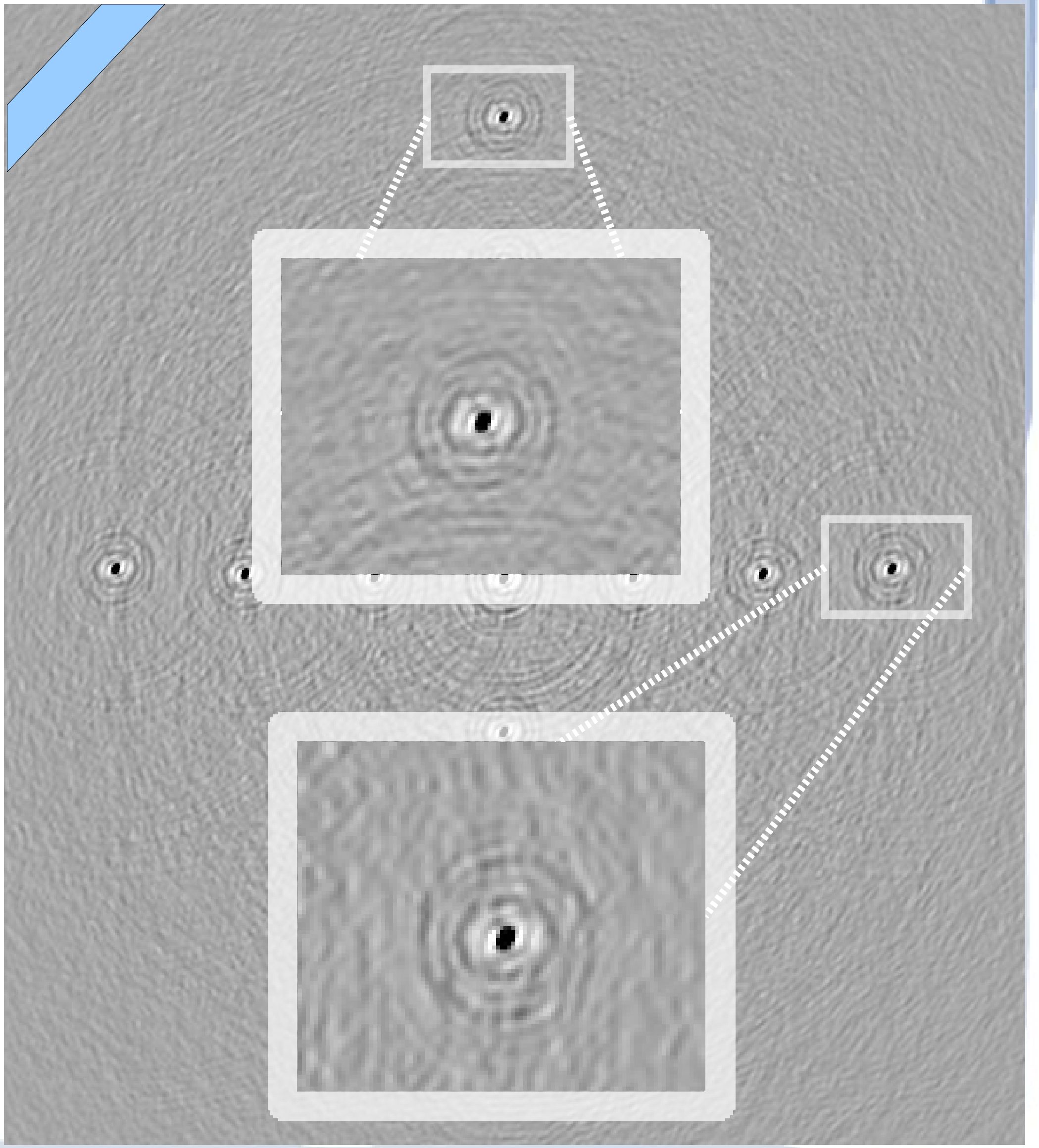
LOFAR Eor GPU Cluster

- Configuration
  - 2 x Tesla C1060 (GPU)
    - Compute 1.3 capable
    - 240 Processors @ 602 MHz
    - 4GB Memory
  - 8 x Intel(R) Xeon(R) CPU E5520 @2.27GHz
  - 12 GB Ram
  - We have 80 Such Nodes

Results (Simulation)

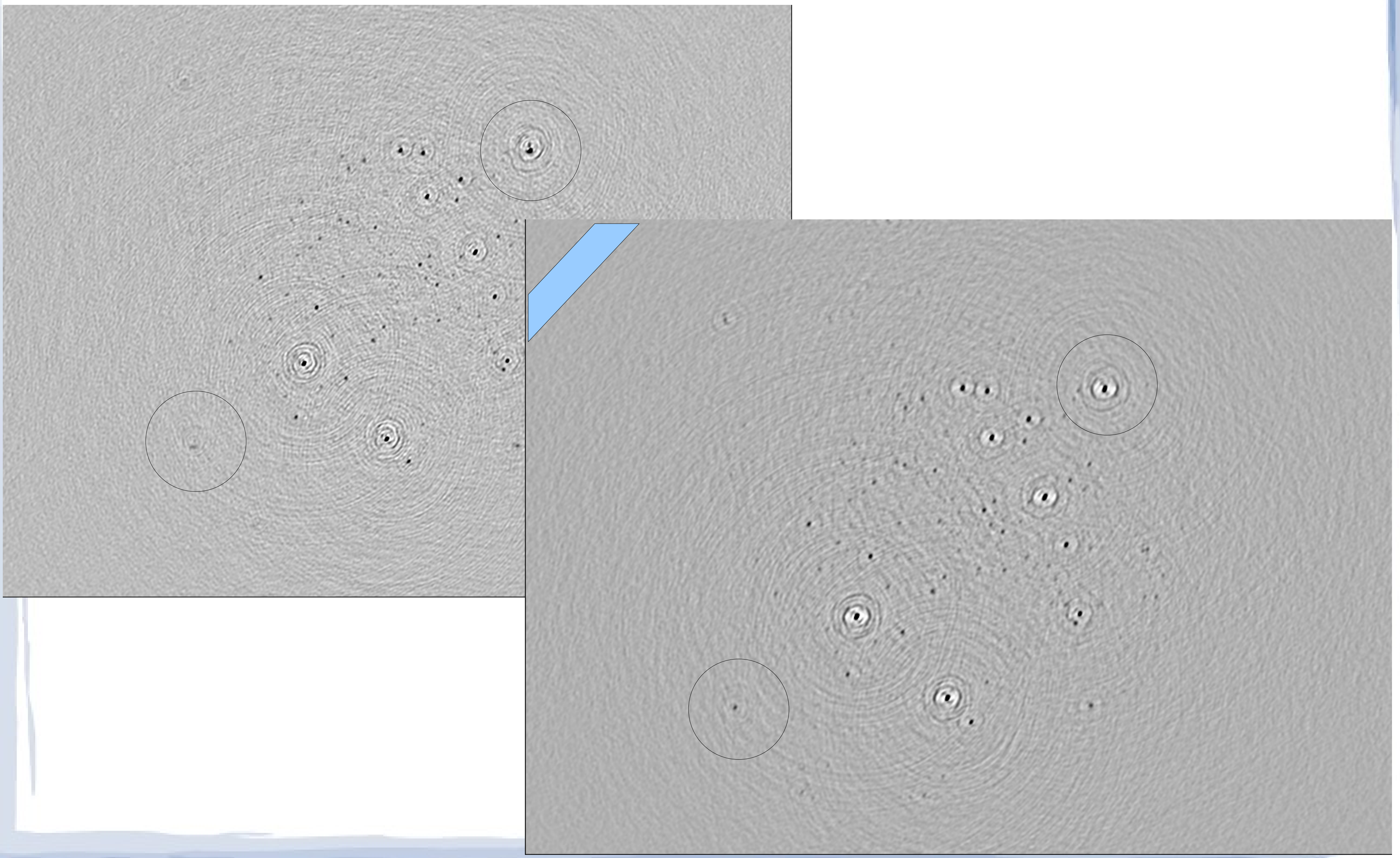


Normal Imaging

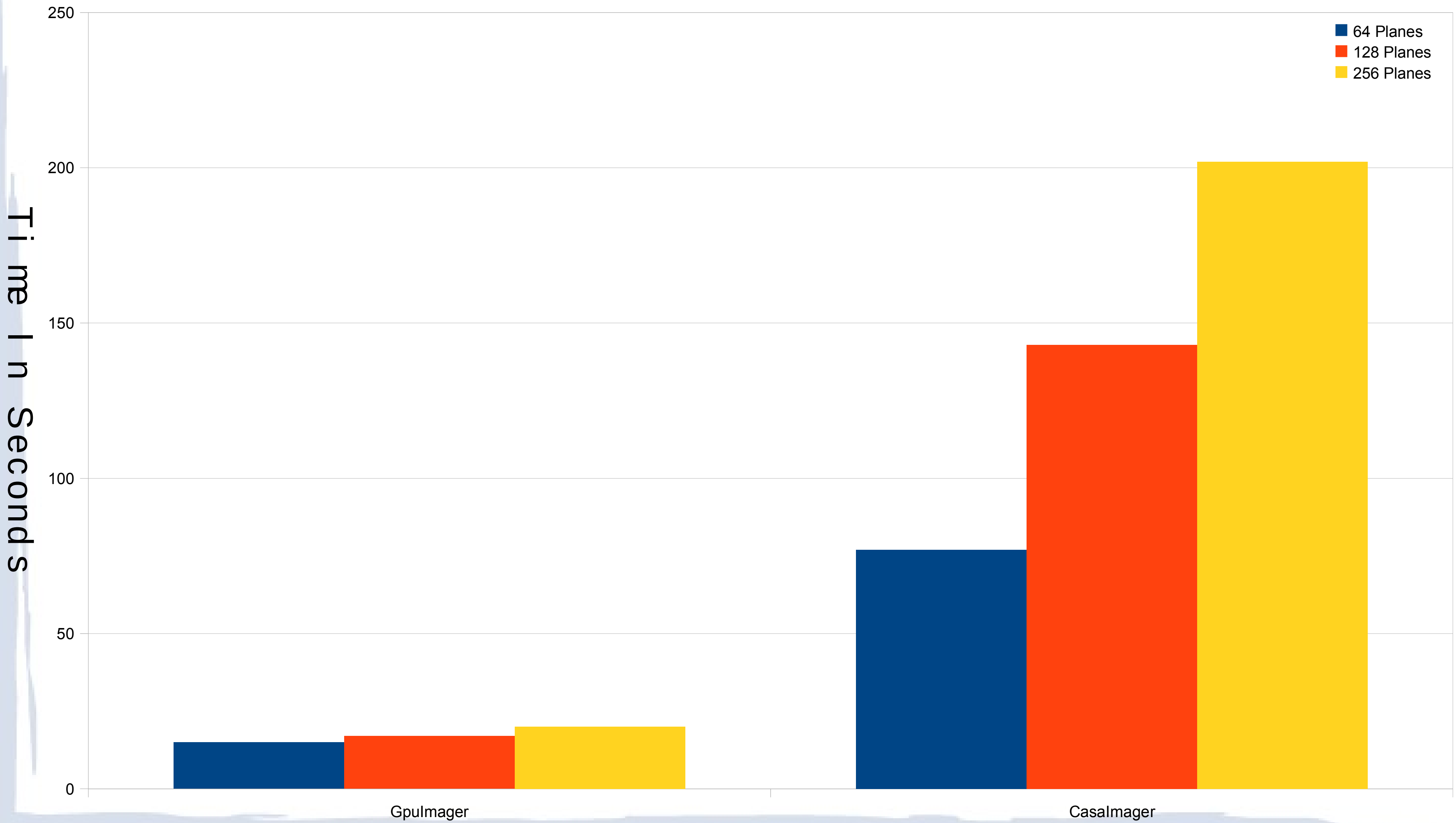


W-Projected

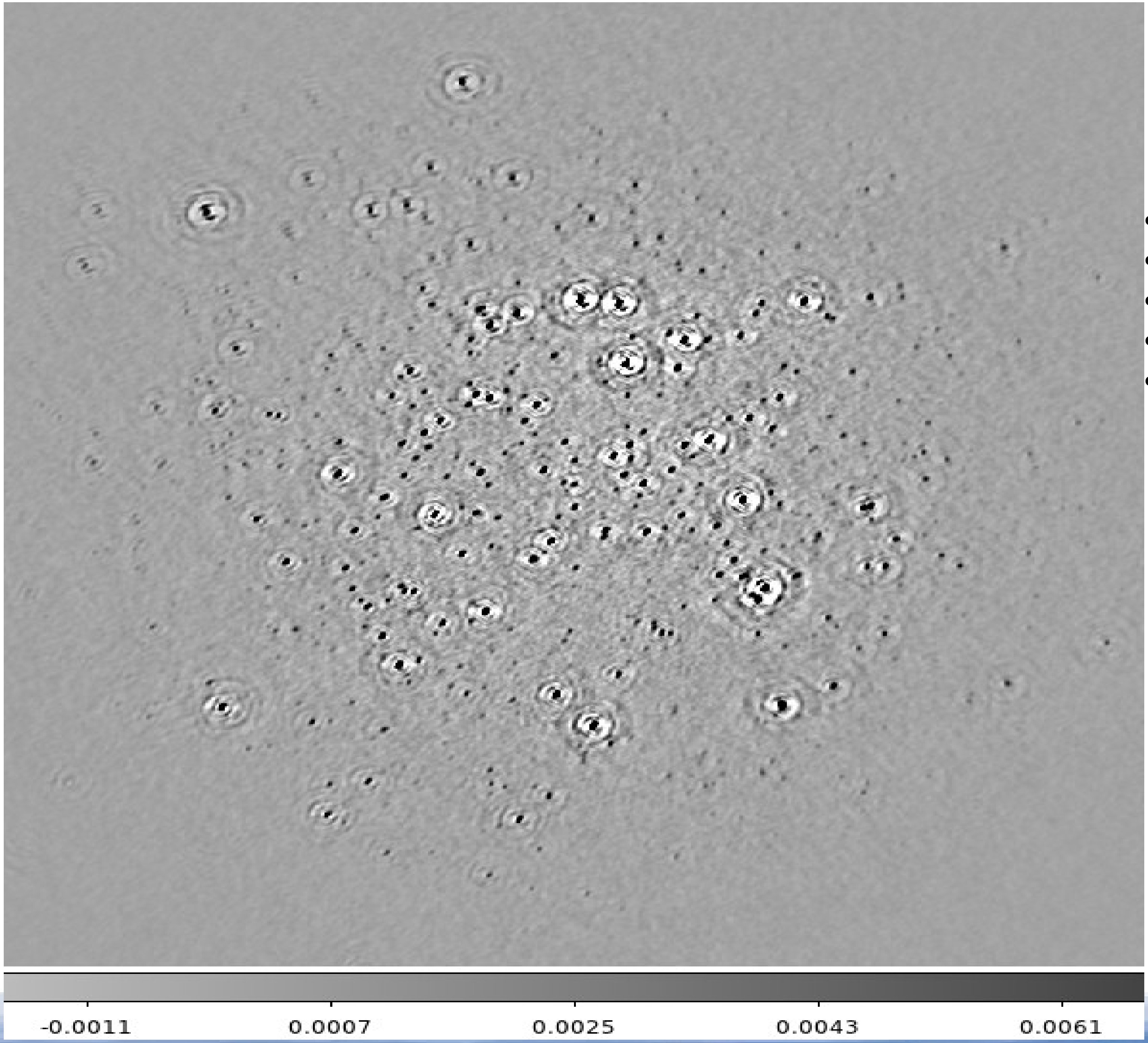
3C196 W-Corrected



W-Projection Performance



3C196



- GPU Imager
- W-Projection
- 256 Planes
- 80 Nodes
- In About 5 mins

Minimum Variance Distortionless Response

- Minimizes side lobe gains (cause beam to vary spatially)
- Computationally intensive
- Parallel computation possible.
  - Dirty Image Equation:

$$I_k(l,m)=A_k^H(l,m)\,R_k\,A_k(l,m)$$

$I_k(l,m)$	Pixel Intensity at l,m at time instance k
$A_k(l,m)$	Antenna steering vector
R_k	Array correlation matrix (ACM).

MVDR

- The weights in dirty image A(l,m) are replaced with W(l,m)
- Weight set to minimize influence of interfering sources. $W = argmin_w W_H R W$
- Under the constraint $W^H A = 1$

$$W^H = \frac{A^H R^{-1}}{A^H R^{-1} A}$$

$$I_k(l,m) = W^H R W$$

Ref: R. Levanda and Amir Leshem, Liu at al., Stoica et al.

Challenges

$$W^H = \frac{A^H R^{-1}}{A^H R^{-1} A}$$

Inverse of Visibility Matrix

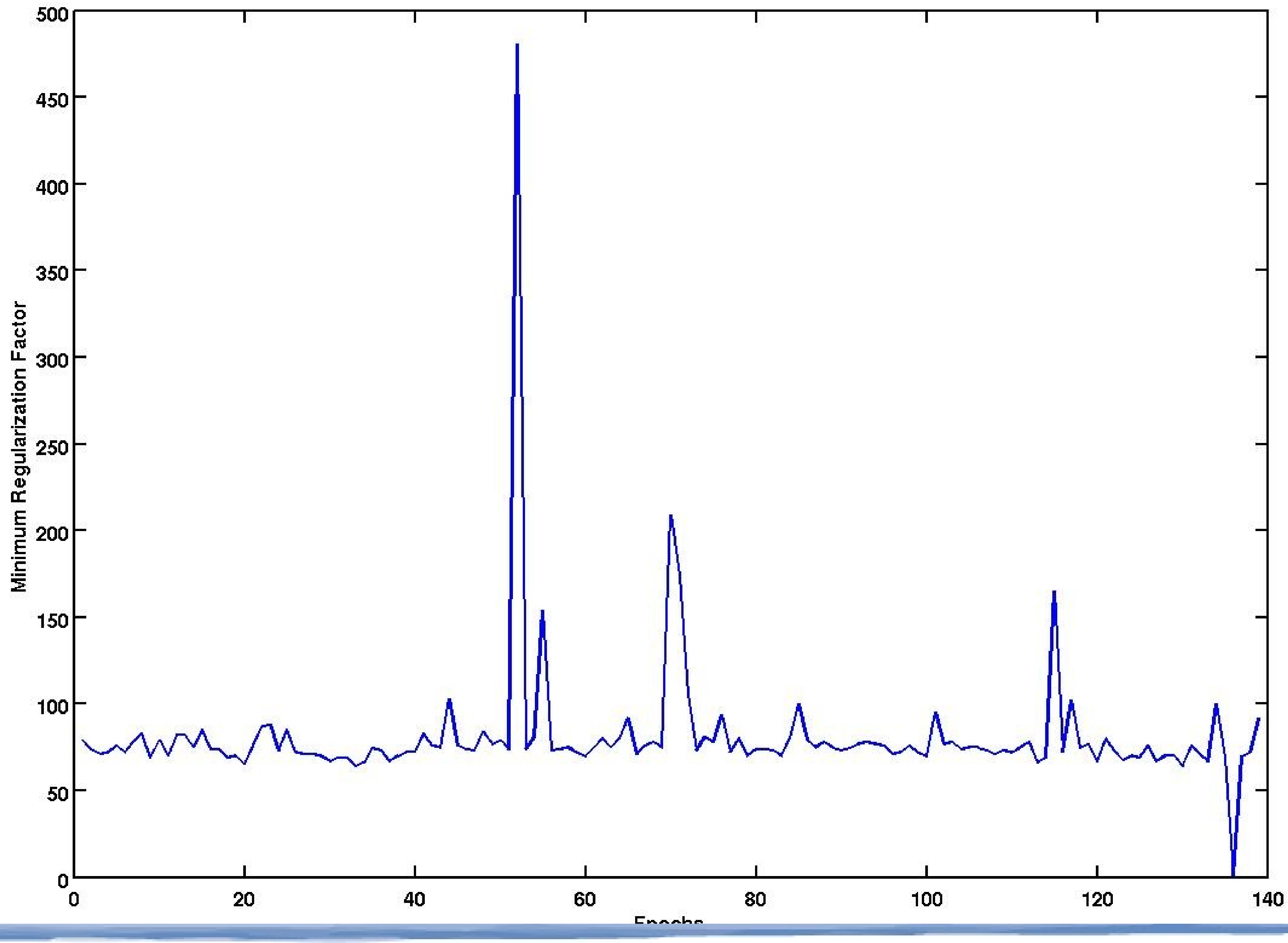
- ACM close to singular.
- Diagonal loading, (non-linear) dimensionality reduction
- L-curve type of analysis to select diagonal loading strength

Challenges: beamformer power vs time

itle:/home/vkrishna/Documents/experi/
reator:MATLAB, The MathWorks, Inc. Vers
reationDate:07/07/2011 15:17:26
anguageLevel:2

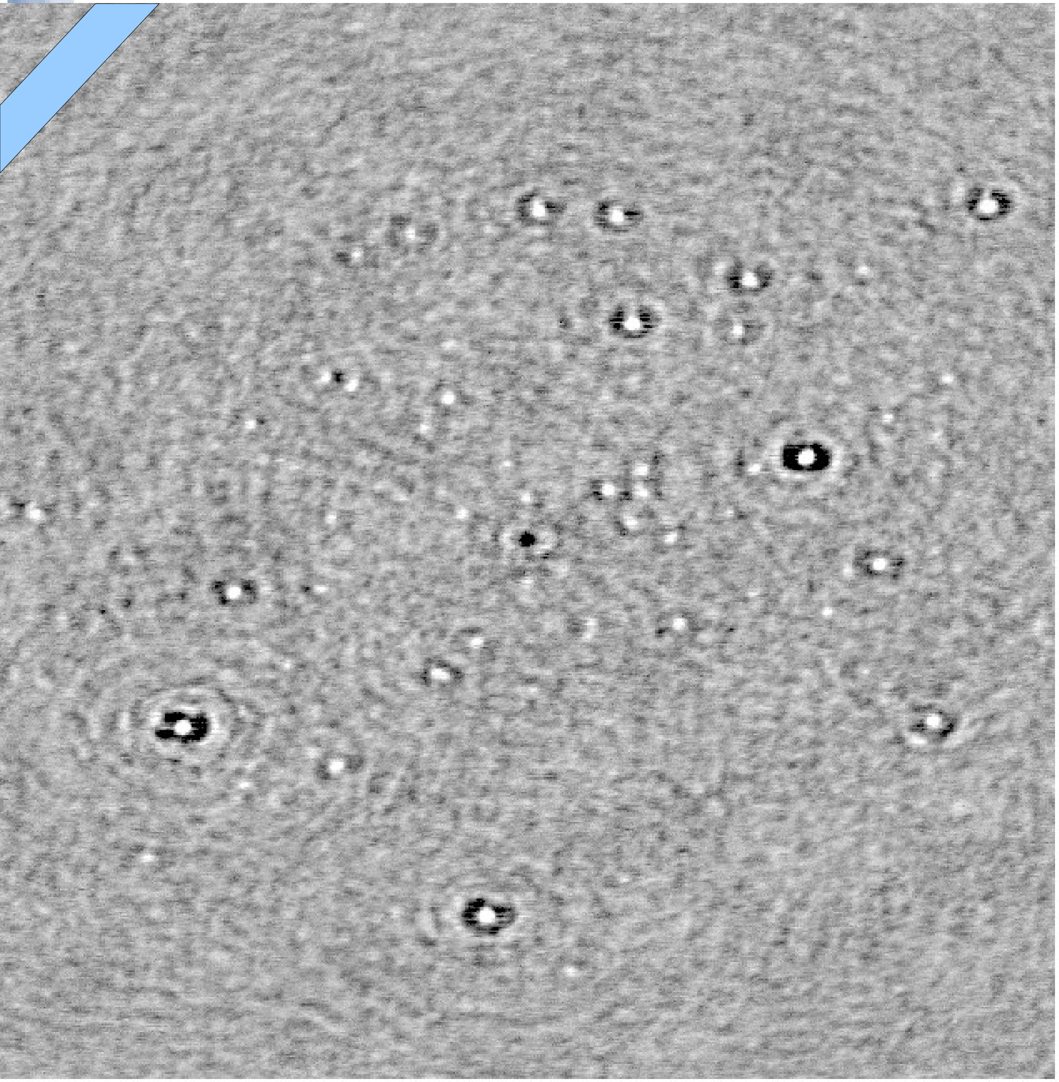
- Flagged/missing data => Higher Diagonal Loading Factor
- Higher regularization factor => higher the noise in that epoch
- Cholesky Factorization to determine level of Regularization.

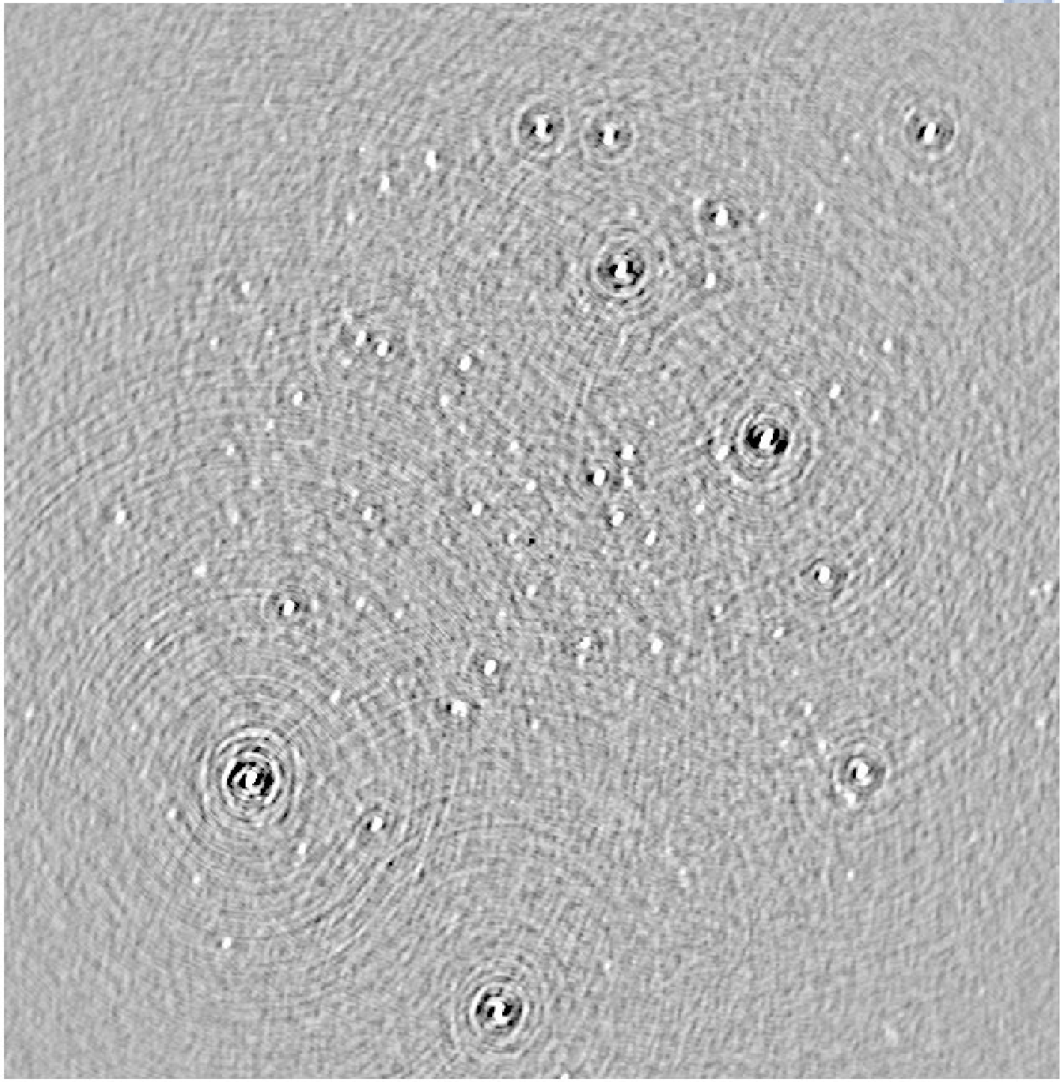
Noisy data



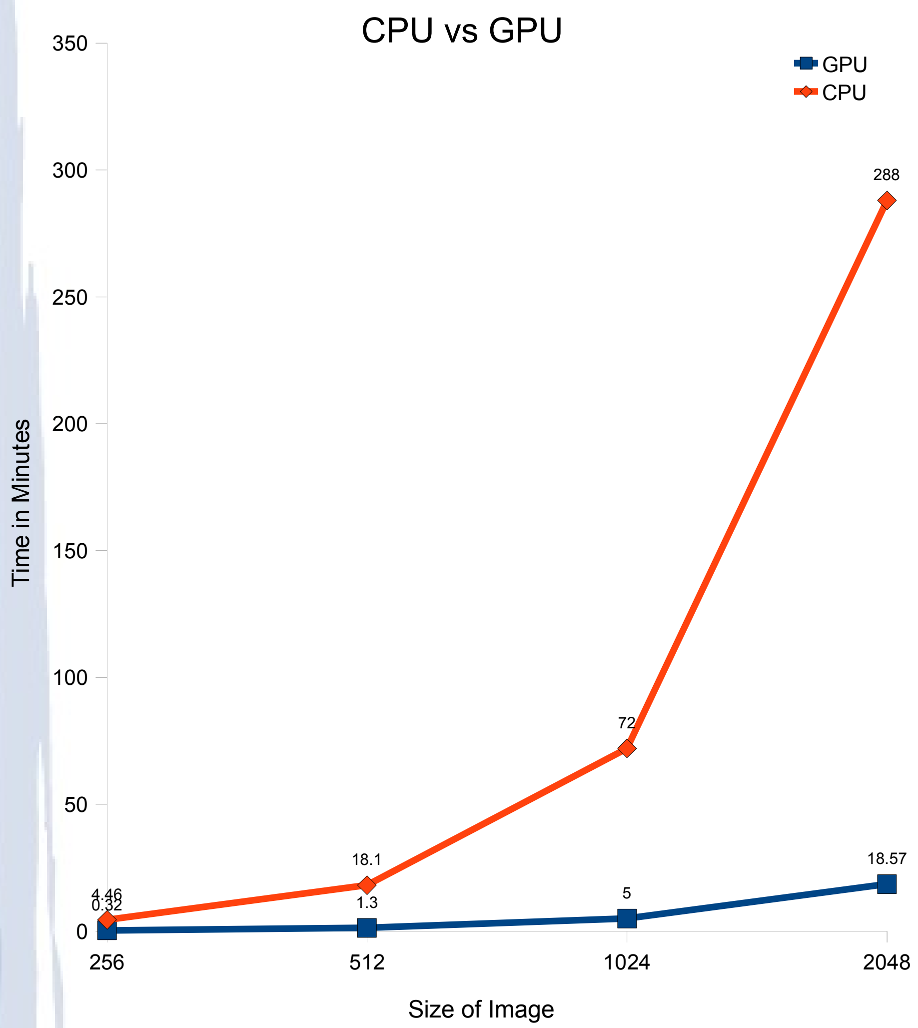
Minimum Regularization Factor
500
450
400
350
300
250
200
150
100
50
0
0
20
40
60
80
100
120
140

Results



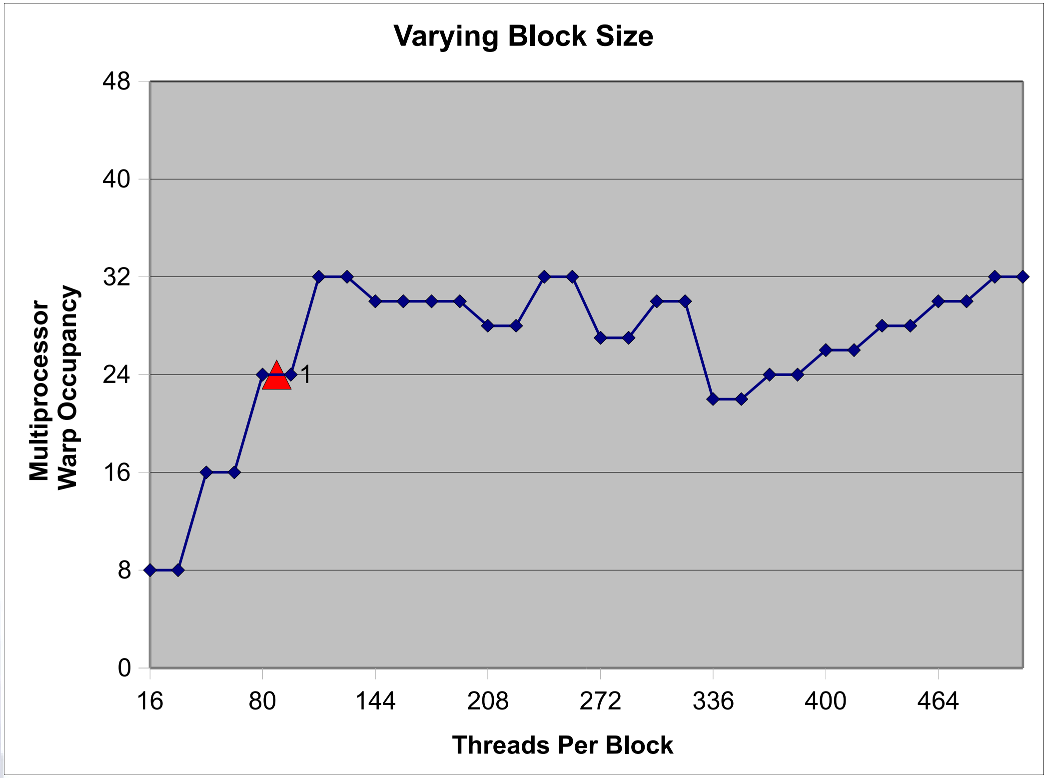


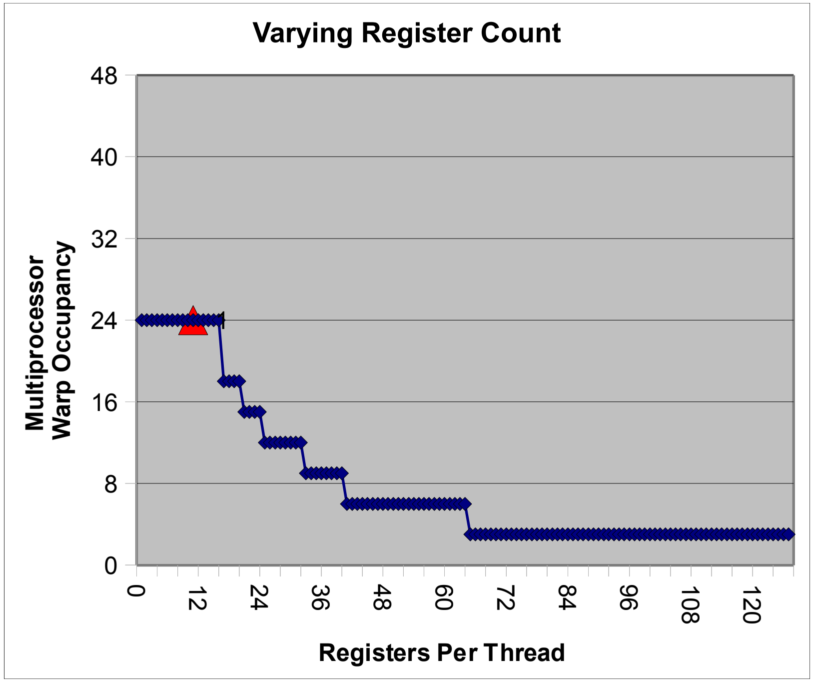
CPU vs GPU



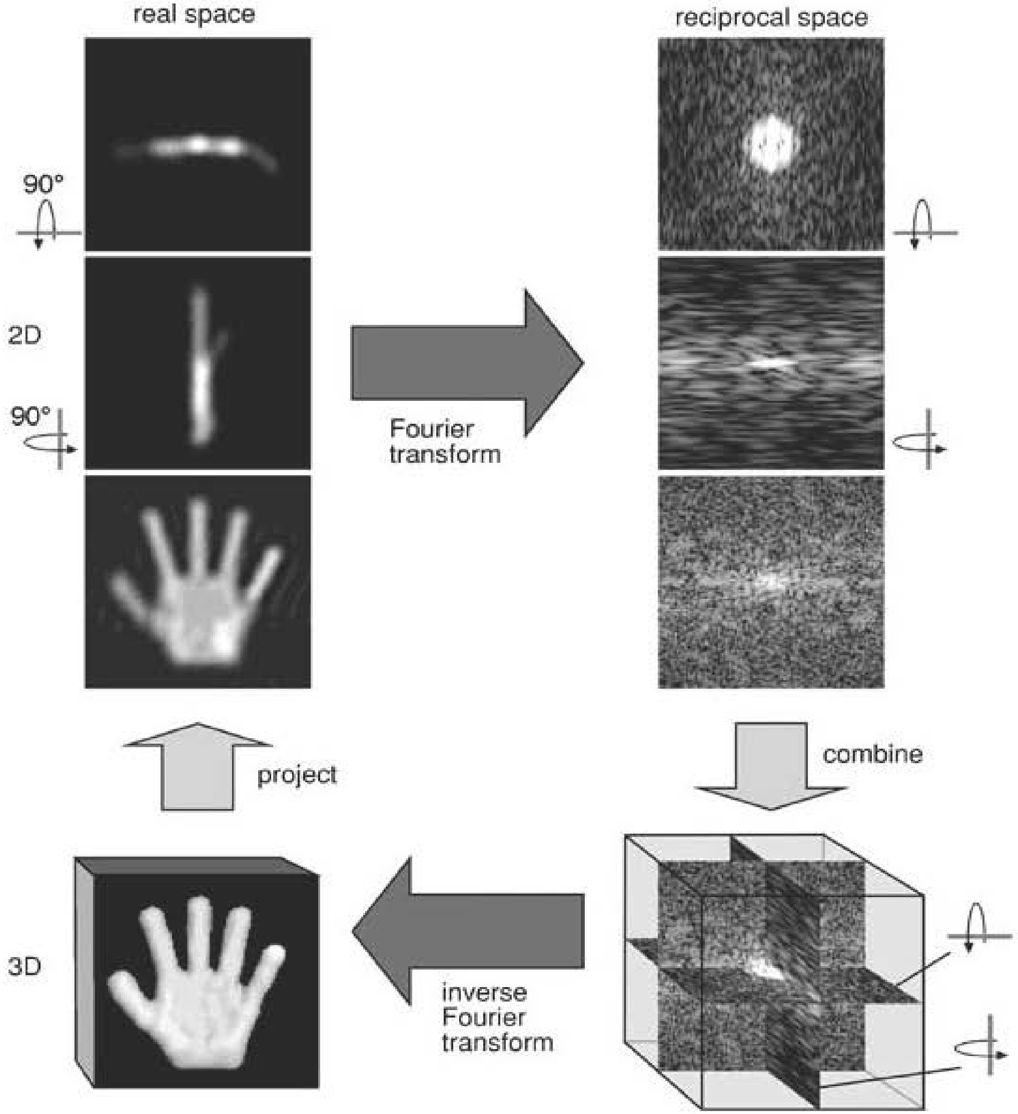
- CPU implementation:
  - Threads equivalent to #processors on system
  - Each Thread processes part of final image.
- GPU Implementation:
  - Multiple threads process a single pixel
  - Each thread calculates component of vector.

Optimization





Radon projection



Tomography 101

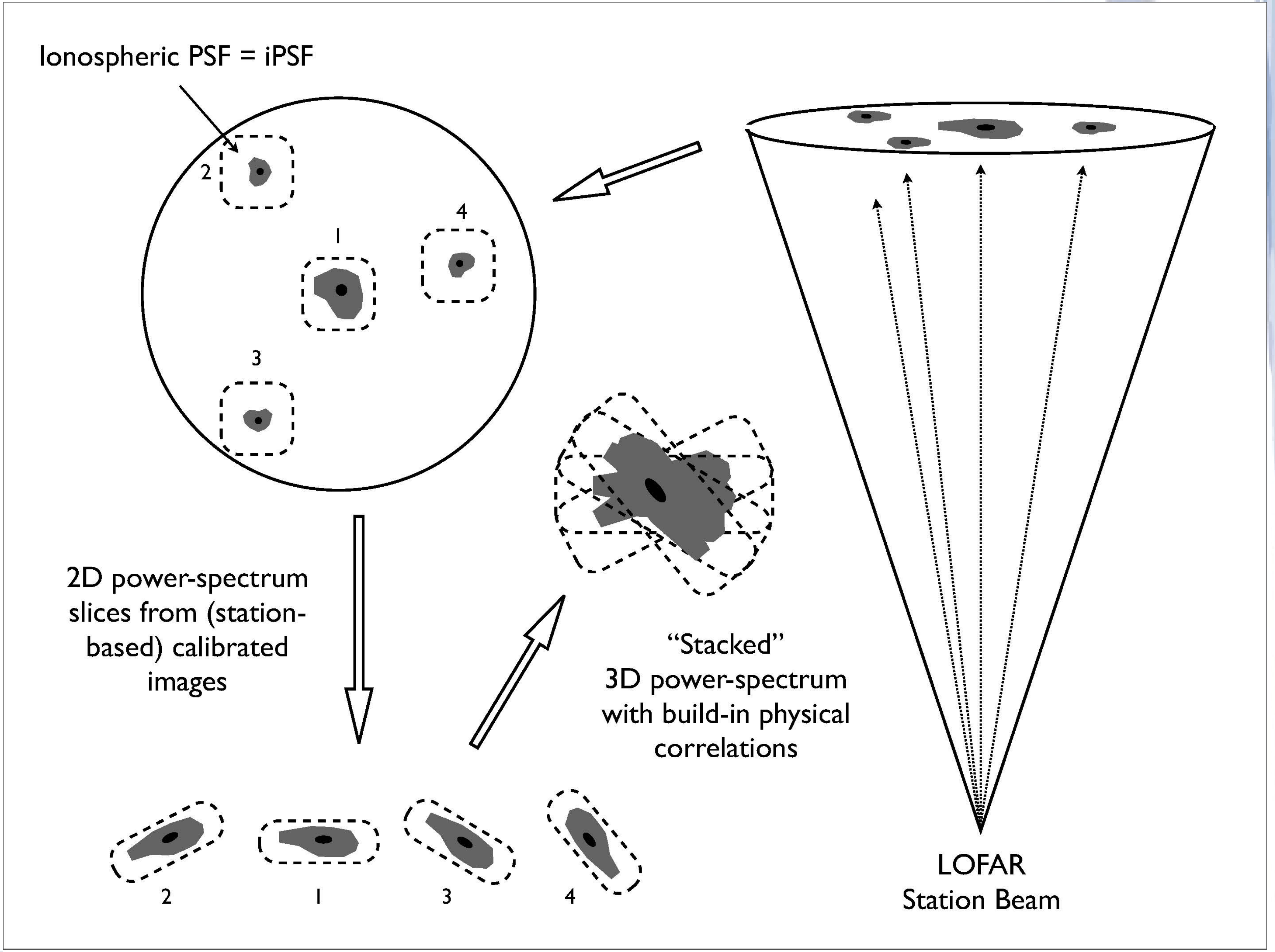
1) Take 2D images of many projections of e.g. a hand (Radon projection)

2) FT them

3) Combine 2D FTs into a single 3D FT. (Take projection angles into account)

4) 3D FT this to a 3D image of hand.

That is all there is to tomography


Ionospheric PSF = iPSF
2
1
4
3
2D power-spectrum slices from (station-based) calibrated images
"Stacked" 3D power-spectrum with build-in physical correlations
2
1
3
4
LOFAR Station Beam

Power-spectrum Tomography

Scattered Intensity

2D Sky Intensity

3D power-spectrum

$$\boxed{\delta I^{(s)}(s_u, s_v) = \frac{1}{s_w^2} \iint I^{(i)}(s_{0,u}, s_{0,v}) \, |\tilde{\Phi}(\mathbf{s} - \mathbf{s}_0)|^2 d s_{0,u} d s_{0,v}}$$

Directional Cosines

$$s_w^2 = 1 - s_u^2 + s_v^2$$

Geometric Term due to
Curve Sky and Planar Array

Rescaled Intensity

$$\boxed{\delta J^{(s)}(s_u, s_v) \equiv s_w^2 \cdot \delta I^{(s)}(s_u, s_v)}$$

Koopmans 2010

Power-spectrum Tomography

A spherical surface slice through an 3D intensity cylinder

$$\boxed{J^{(i)}_{3D}(\mathbf{s}_0) \equiv I^{(i)}_{3D}(\mathbf{s}_0) \cdot \delta(s_{0,w} - \sqrt{1 - s^2_{0,u} - s_{0,v}})}$$

Rescaled scattering as a 3D convolution

$$\boxed{\delta J^{(s)}(s_u, s_v) = \iiint J^{(i)}_{3D}(\mathbf{s}_0)\,|\tilde{\Phi}(\mathbf{s} - \mathbf{s}_0)|^2 d\mathbf{s}_0}$$

Koopmans 2010

Power-spectrum Tomography

Using the Hankel Transform

Fourier Transform of the phase-shifted w-term!

$$\mathcal{H}(u,v;w) = 2\pi \int_0^1 e^{-2\pi i w \sqrt{1-s^2}} J_0(s \cdot u_{2\mathrm{D}})\, s\, ds$$

one can show that the power-spectrum can be extracted from a model intensity and the scattered intensity

FT of sky residuals divided by FT of the sky model multiplied with a phase-shifted w-term!

$$|\tilde{\Phi}(\mathbf{s})|^2 = \mathcal{F}^{-1}\left[\frac{\delta \tilde{J}^{(s)}(u,v)}{\tilde{I}^{(i)}(u,v) \otimes \mathcal{H}(u,v;w)}\right]$$

Note that the 3D power-spectrum is build up from interference of different w-slices

Koopmans 2010

Summary

- Since we produce snapshots we can also correct them for image plane effects on the fly
- Currently implemented a station beam library
- Highly scalable
-
- New methods of Imaging can be explored

Thank You