PolyCLEAN: a Frank-Wolfe algorithm for source deconvolution with sparsity priors

Application to simulated data with RASCIL

Adrian Jarret, PhD student @EPFL/LCAV

in collaboration with Matthieu Simeoni, Julien Fageot, Martin Vetterli

Swiss SKA Days
04.10.2022

Interferometric Imaging in Radio Astronomy

The challenges ? How to go beyond CLEAN ?

1974, *Astron. Astrophys. Suppl.* **15,** 417-426.

1974

APERTURE SYNTHESIS WITH A NON-REGULAR DISTRIBUTION OF INTERFEROMETER BASELINES

J.A. HÖGBOM
Stockholm Observatory, Sweden

In high-resolution radio interferometry it is often impossible for practical reasons to arrange for the measured baselines to be regularly distributed. The standard Fourier inversion methods may then produce maps which are seriously confused by the effects of the prominent and extended sidelobe patterns of the corresponding synthesized beam. Some methods which have been proposed for avoiding these difficulties are discussed. In particular, the procedure CLEAN is described in some detail. This has been successfully applied to measurements taken with several different radio telescopes and appears to be the best method available at the time of writing.

Interferometric Imaging in Radio Astronomy

The challenges ? How to go beyond CLEAN ?

1974, *Astron. Astrophys. Suppl.* **15**, 417-426.

1974

APERTURE SYNTHESIS WITH A NON-REGULAR DISTRIBUTION
OF INTERFEROMETER BASELINES

MULTI-SCALE CLEAN

2008

Multi-Scale CLEAN deconvolution of radio
synthesis images

T.J. Cornwell

Interferometric Imaging in Radio Astronomy

The challenges ? How to go beyond CLEAN ?

1974

1974, *Astron. Astrophys. Suppl.* **15**, 417-426.

APERTURE SYNTHESIS WITH A NON-REGULAR DISTRIBUTION
OF INTERFEROMETER BASELINES

2008

MULTI-SCALE CLEAN

Multi-Scale CLEAN deconvolution of radio synthesis images

T.J. Cornwell

2011

A&A 532, A71 (2011)
DOI: 10.1051/0004-6361/201117104
© ESO 2011

Astronomy & Astrophysics

A multi-scale multi-frequency deconvolution algorithm for synthesis imaging in radio interferometry

U. Rau[1] and T. J. Cornwell[2]

Interferometric Imaging in Radio Astronomy

The challenges ? How to go beyond CLEAN ?

1974

1974, *Astron. Astrophys. Suppl.* **15**, 417-426.

APERTURE SYNTHESIS WITH A NON-REGULAR DISTRIBUTION OF INTERFEROMETER BASELINES

2008

MULTI-SCALE CLEAN

Multi-Scale CLEAN deconvolution of radio synthesis images

T.J. Cornwell

2011

A&A 532, A71 (2011)
DOI: 10.1051/0004-6361/201117104
© ESO 2011

Astronomy & Astrophysics

A multi-scale multi-frequency deconvolution algorithm for synthesis imaging in radio interferometry

U. Rau[1] and T. J. Cornwell[2]

- Some weaknesses:
  - Noise robustness
  - Large FOV: convolution model less accurate
  - Stopping criterion

Interferometric Imaging in Radio Astronomy

The challenges ? How to go beyond CLEAN ?

1974

1974, Astron. Astrophys. Suppl. 15, 417-426.

APERTURE SYNTHESIS WITH A NON-REGULAR DISTRIBUTION OF INTERFEROMETER BASELINES

2008

MULTI-SCALE CLEAN

Multi-Scale CLEAN deconvolution of radio synthesis images

T.J. Cornwell

2011

A&A 532, A71 (2011)
DOI: 10.1051/0004-6361/201117104
© ESO 2011

Astronomy & Astrophysics

A multi-scale multi-frequency deconvolution algorithm for synthesis imaging in radio interferometry

U. Rau[1] and T. J. Cornwell[2]

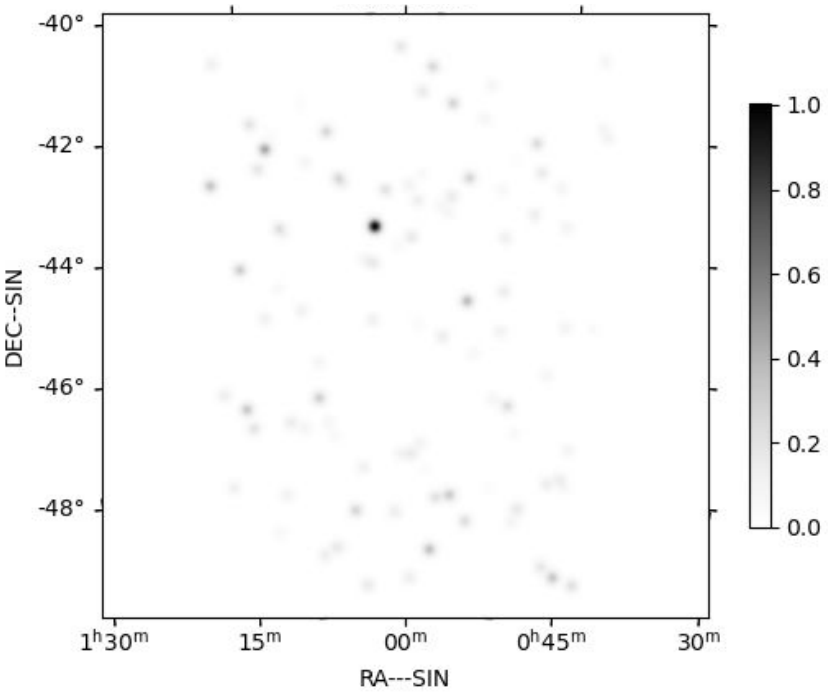
- Some weaknesses:
  - Noise robustness
  - Large FOV: convolution model less accurate
  - Stopping criterion

Our approach:

- Optimization problem with sparsity-promoting penalty

The data model

Computational imaging seen as a linear inverse problem



The data model

Computational imaging seen as a linear inverse problem

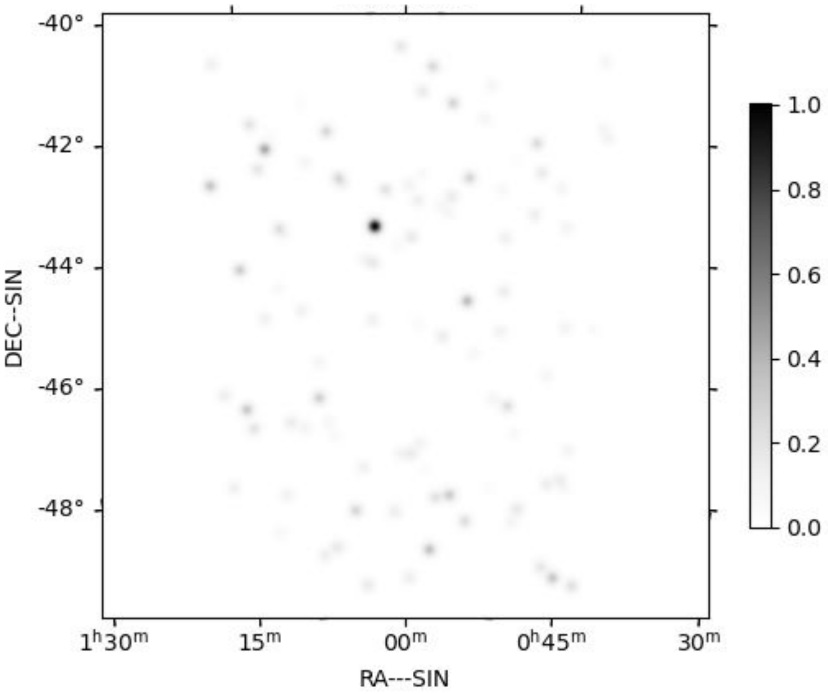
- Van Cittert - Zernike theorem:

$$\mathcal{V}(u,v) = \mathcal{F}\{I\}(u,v) = \iint I(l,m)e^{-2i\pi(ul+vm)}\mathrm{d}l\mathrm{d}m$$

Visibility function

- Partial UV coverage:

$$\mathbf{V}_{i,k} = \mathcal{V}(u_{i,k}, v_{i,k})$$



The data model

Computational imaging seen as a linear inverse problem

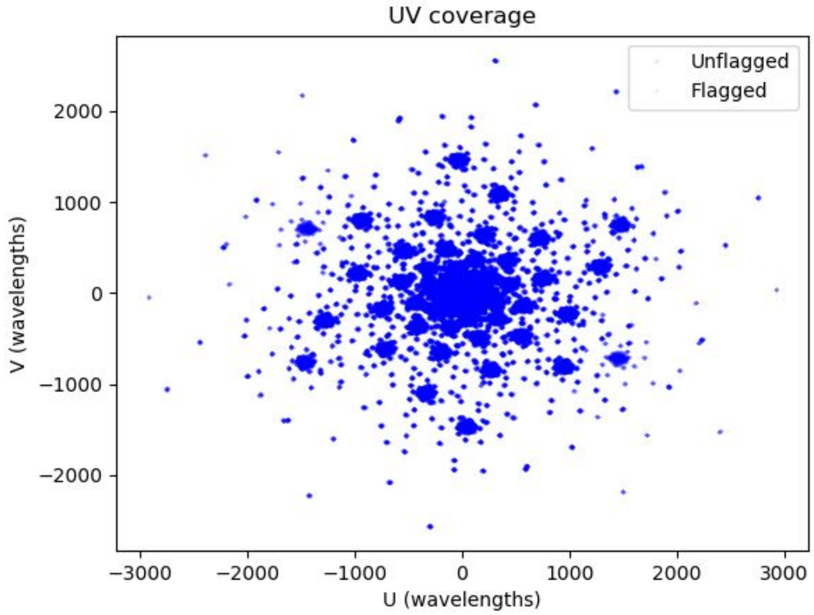
- Van Cittert - Zernike theorem:

$$\mathcal{V}(u,v) = \mathcal{F}\{I\}(u,v)$$
$$= \iint I(l,m)e^{-2i\pi(ul+vm)}\mathrm{d}l\mathrm{d}m$$

Visibility function

- Partial UV coverage:

$$\mathbf{V}_{i,k} = \mathcal{V}(u_{i,k}, v_{i,k})$$



rmax=10 km, 398 stations

The data model

Computational imaging seen as a linear inverse problem

- Van Cittert - Zernike theorem:

$$\mathcal{V}(u,v) = \mathcal{F}\{I\}(u,v) = \iint I(l,m)e^{-2i\pi(ul+vm)}\mathrm{d}l\mathrm{d}m$$

Visibility function

- Partial UV coverage:

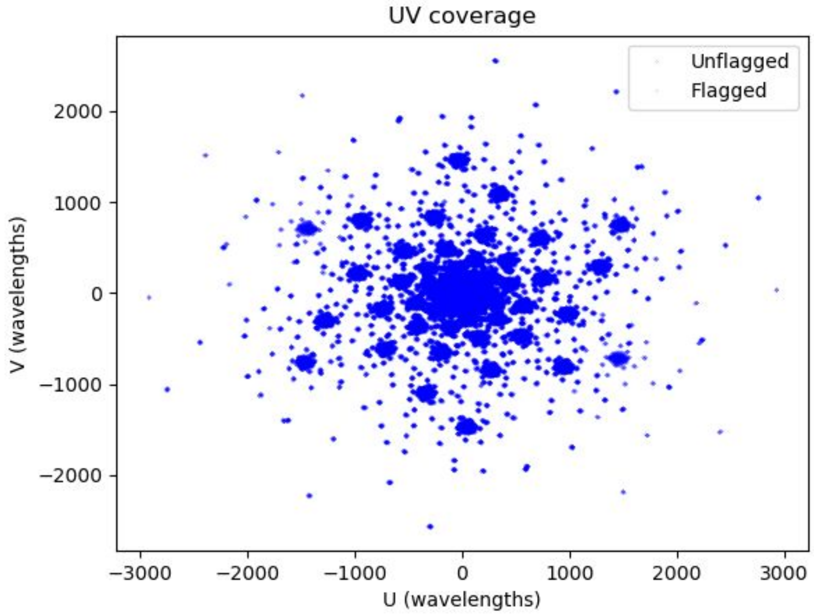
$$\mathbf{V}_{i,k} = \mathcal{V}(u_{i,k}, v_{i,k})$$

- Measurement equation:

$$\boxed{\mathbf{V} = \mathbf{\Phi}(I)} \quad \in \mathbb{C}^L$$

Visibility

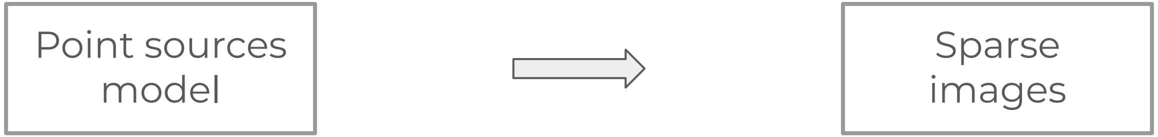
Sky Image



rmax=10 km, 398 stations

Solving with the LASSO optimization problem

Sparse recovery and robustness to noise



Solving with the LASSO optimization problem

Sparse recovery and robustness to noise

Point sources model → Sparse images

- The LASSO optimization problem:

Grid-based
=
raster image

$$\text{Minimize:} \quad ||\mathbf{V} - \mathbf{\Phi}(I)||_2^2 + \lambda ||I||_1$$

The PolyCLEAN Algorithm

A fast and scalable solver for the LASSO

Algorithm 1 Polyatomic Frank-Wolfe (PFW) for the LASSO

Let: $I_k = 0$, $\mathcal{S}_k = \emptyset$, threshold $\delta > 0$.

for $k = 1, 2, \cdots$ **do**

1. Compute the dirty residual: $\eta_k \leftarrow \frac{1}{\lambda}\mathbf{\Phi}^* \left(\mathbf{V} - \mathbf{\Phi}(I_k)\right)$
2. Update current threshold: $\delta_k \leftarrow \delta / k$
3. Search for δ_k-maxima of η_k: $\mathcal{I}_k = \{j : \eta_k[j] \in [\max \eta_k - \delta_k, \max \eta_k]\}$
4. Update the set of active components: $\mathcal{S}_{k+1} \leftarrow \mathcal{S}_k \cup \mathcal{I}_k$
5. Update the weights according tho LASSO:

$$I_{k+1} \in \underset{\mathrm{Supp}(I) \subset \mathcal{S}_{k+1}}{\arg\min} \; LASSO(I)$$

end for

[Jarret et al., 2021]

The PolyCLEAN Algorithm

A fast and scalable solver for the LASSO

Algorithm 1 Polyatomic Frank-Wolfe (PFW) for the LASSO

Let: $I_k = 0$, $\mathcal{S}_k = \emptyset$, threshold $\delta > 0$.

for $k = 1, 2, \cdots$ **do**

1. Compute the dirty residual: $\eta_k \leftarrow \frac{1}{\lambda}\mathbf{\Phi}^* \left(\mathbf{V} - \mathbf{\Phi}(I_k)\right)$
2. Update current threshold: $\delta_k \leftarrow \delta/k$
3. Search for δ_k-maxima of η_k: $\mathcal{I}_k = \{j : \eta_k[j] \in [\max \eta_k - \delta_k, \max \eta_k]\}$
4. Update the set of active components: $\mathcal{S}_{k+1} \leftarrow \mathcal{S}_k \cup \mathcal{I}_k$
5. Update the weights according tho LASSO:

$$I_{k+1} \in \underset{\mathrm{Supp}(I) \subset \mathcal{S}_{k+1}}{\arg\min} \; LASSO(I)$$

end for

Similar to CLEAN

[Jarret et al., 2021]

The PolyCLEAN Algorithm

A fast and scalable solver for the LASSO

Algorithm 1 Polyatomic Frank-Wolfe (PFW) for the LASSO

Let: $I_k = 0$, $\mathcal{S}_k = \emptyset$, threshold $\delta > 0$.

for $k = 1, 2, \cdots$ **do**

1. Compute the dirty residual: $\eta_k \leftarrow \frac{1}{\lambda}\mathbf{\Phi}^* \left(\mathbf{V} - \mathbf{\Phi}(I_k)\right)$
2. Update current threshold: $\delta_k \leftarrow \delta / k$
3. Search for δ_k-maxima of η_k: $\mathcal{I}_k = \{j : \eta_k[j] \in [\max \eta_k - \delta_k, \max \eta_k]\}$
4. Update the set of active components: $\mathcal{S}_{k+1} \leftarrow \mathcal{S}_k \cup \mathcal{I}_k$
5. Update the weights according tho LASSO:

$$I_{k+1} \in \underset{\text{Supp}(I) \subset \mathcal{S}_{k+1}}{\arg\min} \; LASSO(I)$$

end for

Identification of components

[Jarret et al., 2021]

The PolyCLEAN Algorithm

A fast and scalable solver for the LASSO

Algorithm 1 Polyatomic Frank-Wolfe (PFW) for the LASSO

Let: $I_k = 0$, $\mathcal{S}_k = \emptyset$, threshold $\delta > 0$.

for $k = 1, 2, \cdots$ **do**

1. Compute the dirty residual: $\eta_k \leftarrow \frac{1}{\lambda}\mathbf{\Phi}^* \left(\mathbf{V} - \mathbf{\Phi}(I_k)\right)$
2. Update current threshold: $\delta_k \leftarrow \delta/k$
3. Search for δ_k-maxima of η_k: $\mathcal{I}_k = \{j : \eta_k[j] \in [\max \eta_k - \delta_k, \max \eta_k]\}$
4. Update the set of active components: $\mathcal{S}_{k+1} \leftarrow \mathcal{S}_k \cup \mathcal{I}_k$
5. Update the weights according tho LASSO:

$$I_{k+1} \in \underset{\mathrm{Supp}(I) \subset \mathcal{S}_{k+1}}{\arg\min} \; LASSO(I)$$

Support constrained solution

end for

[Jarret et al., 2021]

The PolyCLEAN Algorithm

Key points to keep in mind

1. **Convergence guarantee:**
   - ➢ convergence towards optimum of the LASSO objective function, speed $\mathcal{O}(1/k)$

The PolyCLEAN Algorithm

Key points to keep in mind

1. **Convergence guarantee:**
   - ➢ convergence towards optimum of the LASSO objective function, speed $\mathcal{O}(1/k)$
2. **Much faster in practice:**
   - ➢ Similar to APGD for sparse problems, which has speed $\mathcal{O}(1/k^2)$

The PolyCLEAN Algorithm

Key points to keep in mind

1. **Convergence guarantee:**
   - ➢ convergence towards optimum of the LASSO objective function, speed $\mathcal{O}(1/k)$
2. **Much faster in practice:**
   - ➢ Similar to APGD for sparse problems, which has speed $\mathcal{O}(1/k^2)$
3. **Sparse iterates:**
   - ➢ Low memory requirements, scalability in terms of image size

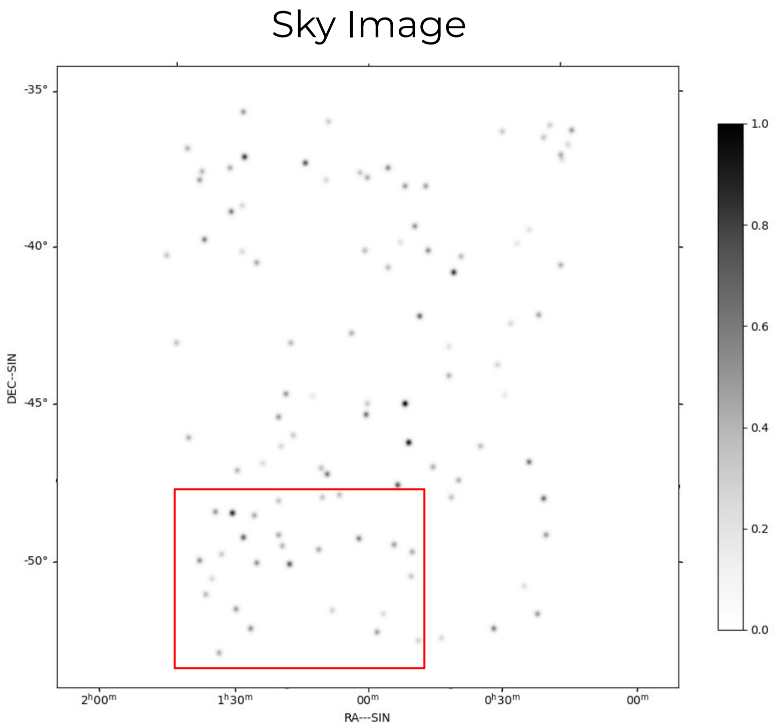
The PolyCLEAN Algorithm

Key points to keep in mind

1. **Convergence guarantee:**
   - ➢ convergence towards optimum of the LASSO objective function, speed $\mathcal{O}(1/k)$
2. **Much faster in practice:**
   - ➢ Similar to APGD for sparse problems, which has speed $\mathcal{O}(1/k^2)$
3. **Sparse iterates:**
   - ➢ Low memory requirements, scalability in terms of image size
4. **Similar to CLEAN, but with a defined objective function:**
   - ➢ Interpretation thanks to the objective (representer theorem, bayesian interpretation for noise)

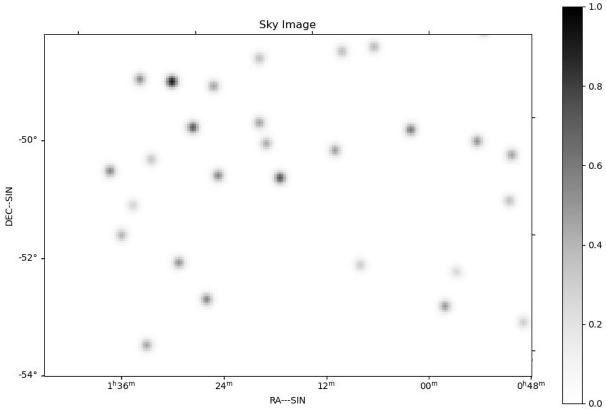
Performances in simulations

Simulated data with RASCIL



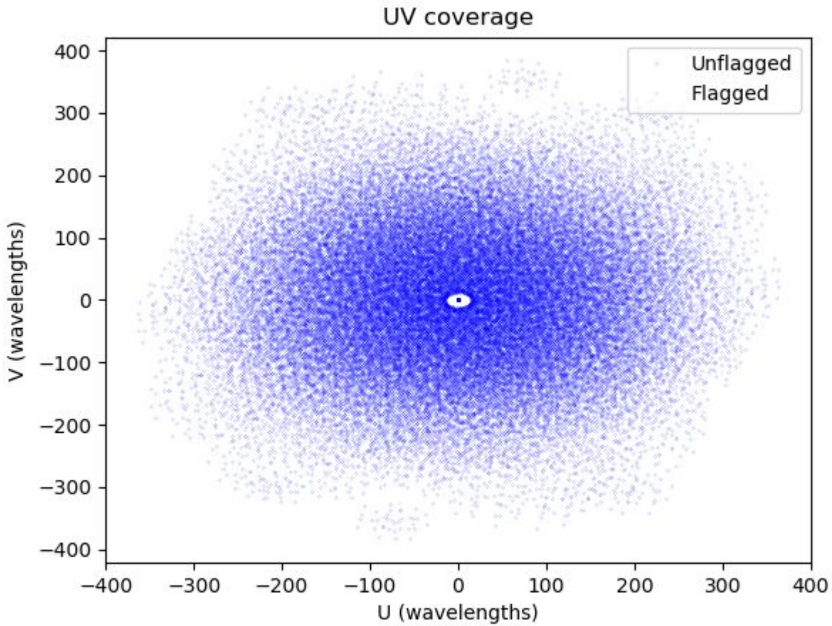
Parameters:

- 100 sources
- FOV: 20 degrees
- image size: 512*512
- Phase center: 15, -45 (deg)



Performances in simulations

Simulated data with RASCIL

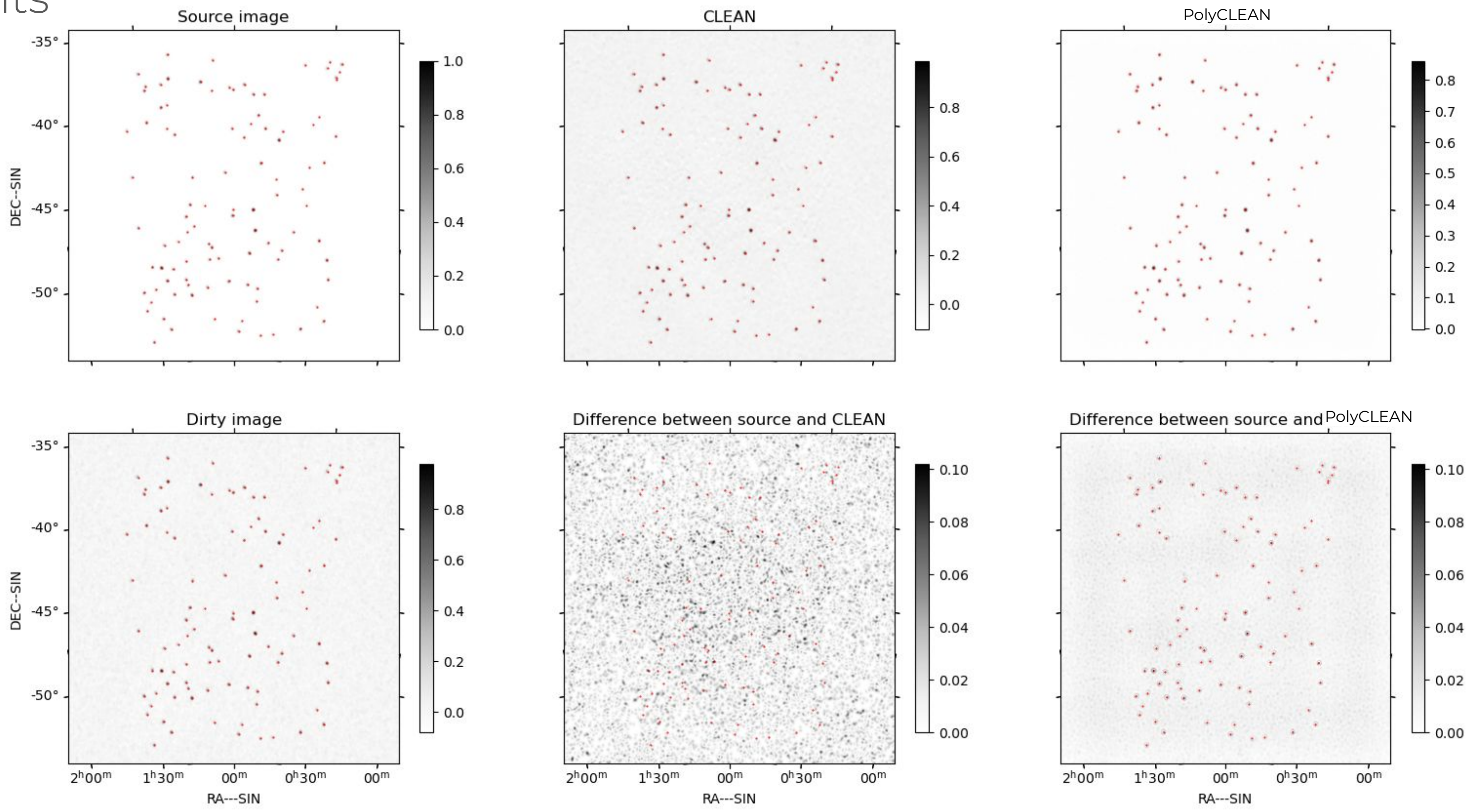



```
# Construct LOW core configuration
lowr3 = create_named_configuration("LOWBD2", rmax=750.)
```

- 236 antennas, 27730 baselines
- Frequency:
  - 10^8 Hz
- Bandwidth:
  - 10^6 Hz

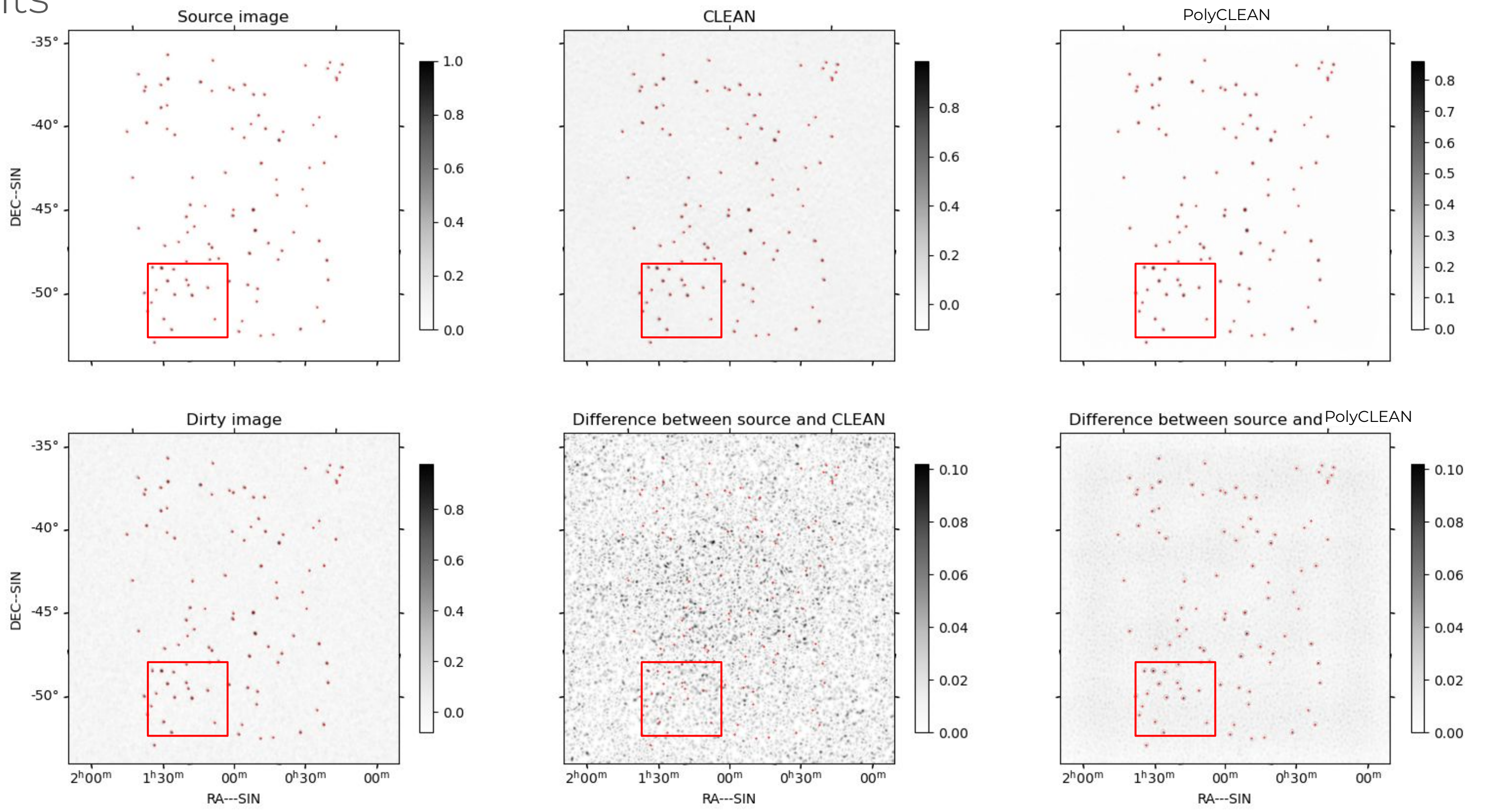
Performances in simulations

Results



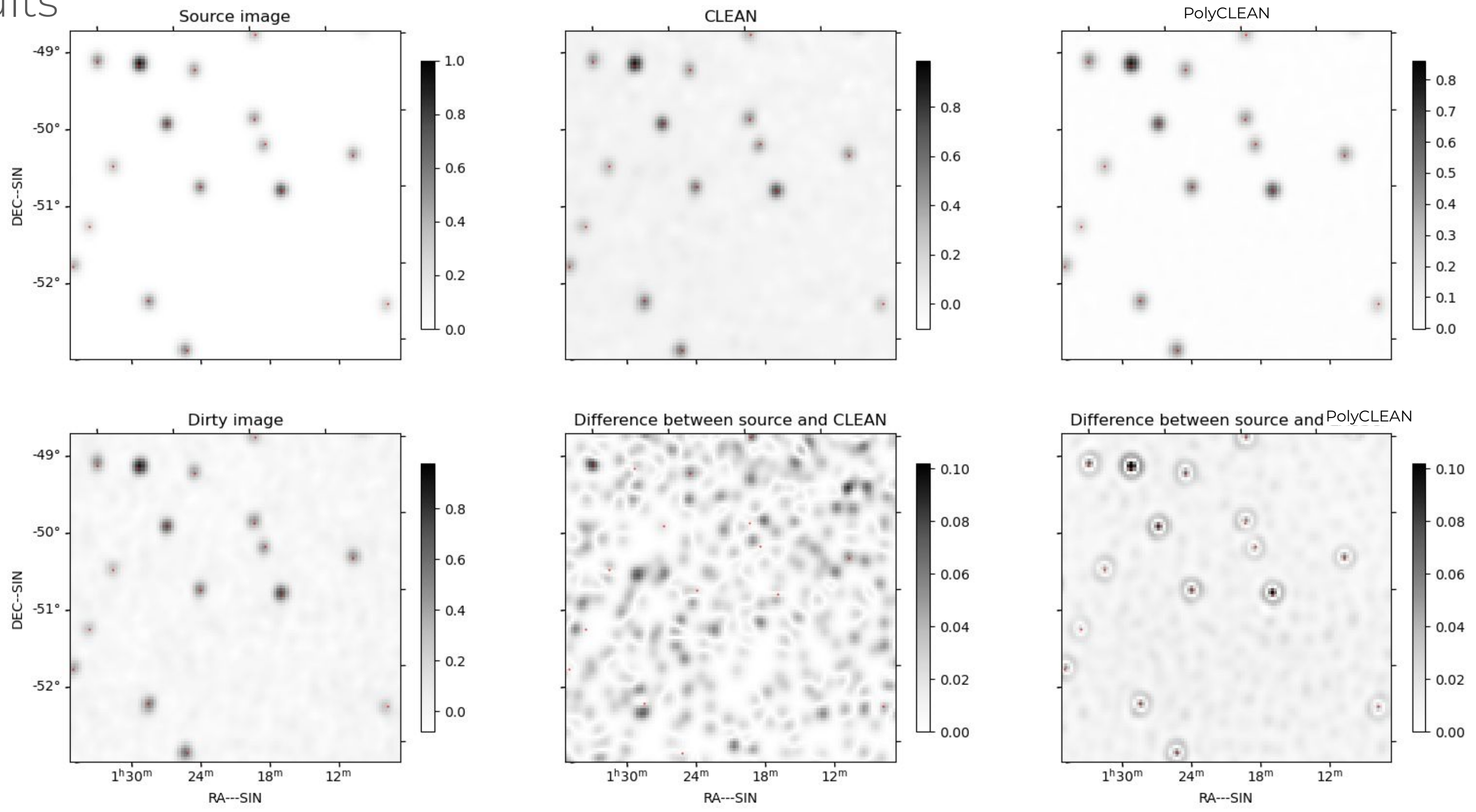
Performances in simulations

Results



Performances in simulations

Results



Performances in simulations

Results

Convolution with fitted CLEAN beam:

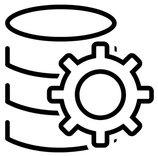
```
MSE with the source:
    Dirty : 4.641e-04
    CLEAN: 3.032e-04
    LASSO : 1.150e-04
```

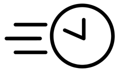
CLEAN runtime: 50s

PolyCLEAN runtime: 60s

Conclusions and future work

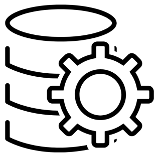
- PolyCLEAN = Polyatomic Frank-Wolfe for Radio Astronomy:
  - ✅ solves a LASSO problem
  - ✅ Scalable (by design)
  - ✅ Competitive run time

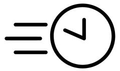




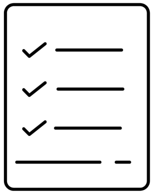
Conclusions and future work

- PolyCLEAN = Polyatomic Frank-Wolfe for Radio Astronomy:
  - ✅ solves a LASSO problem
  - ✅ Scalable (by design)
  - ✅ Competitive run time



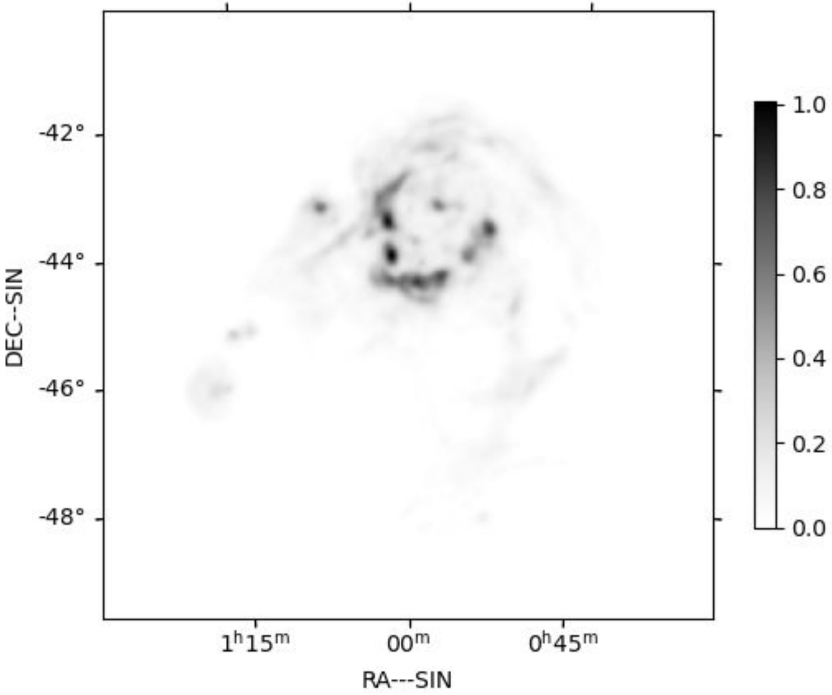


- Improvements and extensions:
  - ➕ Use NUFFT instead of Nifty-Gridder (time and precision improvements)
  - ➕ Run on real world and larger scale problems
  - ➕ Develop extended sources reconstruction



The data model

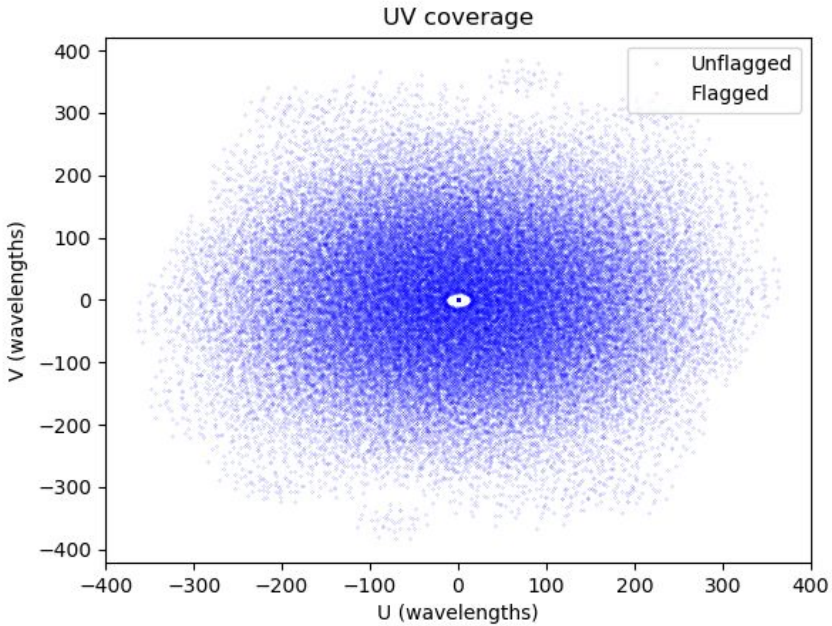
Computational imaging seen as a linear inverse problem

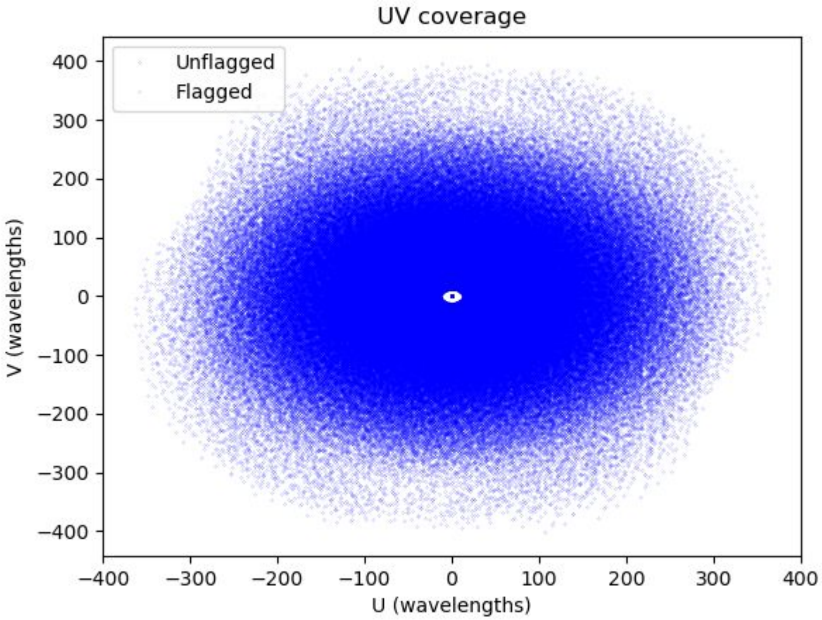


Performances in simulations

Results

5 integration times: [-π/6, +π/6]





Performances in simulations

5 integration times: [-π/6, +π/6]

Results