PolyCLEAN
A Polyatomic CLEAN-like algorithm for sparse Bayesian imaging

Adrian Jarret
PhD Student @EPFL
Swiss SKA Days 2023
01 Background
Radio Interferometry and the CLEAN realm

02 MAP estimation
Optimization problems and numerical challenges

03 PolyCLEAN
Convex optimization solved in an atomic manner

04 Demonstration
Performances and experimental reconstructions
01. Background

Radio Interferometric Imaging
Linear Inverse Problem

\[ V = \Phi I \]
Linear Inverse Problem

\[ V = \Phi I \]

Observed area of the sky

[Credits: Cyril Tasse and the LOFAR surveys team.]
Linear Inverse Problem

Visibility measurements

Spatial frequency information, Fourier-like measurements, complex valued

\[ V = \Phi I \]

Observed area of the sky

[Credits: Cyril Tasse and the LOFAR surveys team.]
Linear Inverse Problem

Visibility measurements

Spatial frequency information, Fourier-like measurements, complex valued

$$V = \Phi I$$

Interferometry operator

Depends on the location of the antennas (baselines)
And the observed wavelength

Observed area of the sky

[Credits: Cyril Tasse and the LOFAR surveys team.]
Measurement Operator

\[
\Phi(I)_k = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{I[i,j]}{n(l_i, m_j)} e^{-i2\pi(u_k l_i + v_k m_j)} W(l_i, m_j; w_k)
\]
Challenges of RI

- Noisy measurements

\[ V = \Phi I + \varepsilon \]
Challenges of RI

- Noisy measurements
- Ill-posed problem

\[ V = \Phi I + \epsilon \]

\[ \text{Null}(\Phi) \neq \{0\} \]
Challenges of RI

- Noisy measurements
- Ill-posed problem

\[
V = \Phi I + \varepsilon
\]

\[
\text{Null}(\Phi) \neq \{0\}
\]

Linear IP

Use of priors for reconstruction!
Challenges of RI

- Noisy measurements
- Ill-posed problem
- Huge volumes of data

\[ V = \Phi I + \varepsilon \]

\[ \text{Null}(\Phi) \neq \{0\} \]

Linear IP

Use of priors for reconstruction!
The CLEAN Algorithm

**The CLEAN Algorithm**

**Parametric Shape of the Solutions**

\[ I^*[n] = \sum_k \alpha_k g(n - n_k) \]

**Matching Pursuit Algorithm**

Linear IP

\[ V = \Phi I \]

**Iterative atomic updates**

\[ I^{(k+1)} \leftarrow I^{(k)} + \alpha g(\cdot - n_k) \]

**Empirical sparsity along iteration**
The CLEANS Algorithm

Parametric Shape of the Solutions

\[ I^*[n] = \sum_k \alpha_k \delta(n - n_k) \]

Point sources

Matching Pursuit Algorithm

Iterative atomic updates

\[ I^{(k+1)} \leftarrow I^{(k)} + \alpha g(\cdot - n_k) \]

Empirical sparsity along iteration

Linear IP

\[ V = \Phi I \]
The Algorithm

Algorithm 1 Högbom CLEAN Algorithm (Major cycles only)

**Parameters:** $k_{\text{max}}$ (iterations), $\alpha > 0$ (gain)

**Initialisation:** $I^{(0)} = 0$, $I_D = \Phi^* V$

for $k = 1, 2, \ldots, k_{\text{max}}$ do

1. Compute the dirty residual: $I_R^{(k)} = I_D - \Phi^* \Phi I^{(k-1)}$

2. Find the location of the next reconstructed source: $s^{(k)} = \arg \max_{(i,j)} |I_R^{(k)}[i,j]|$

3. Update the iterate: $I^{(k)} = I^{(k-1)} + \alpha (\max I_R^{(k)}) \delta_{s^{(k)}}$

end for

**Output:**

Postprocess $I^{(k)}$ (convolution with synthetic beam, add residual image)
CLEAN-Like methods

- Atomic method (scalable)
- A lot of hacks and tips to make them very fast
- Developed and maintained by the astronomers
- Long date expertise
- Calibration-compliant
CLEAN-Like methods (continued)

✔ Atomic method (scalable)
✔ A lot of hacks and tips to make them very fast
✔ Developed and maintained by the astronomers
✔ Long date expertise
✔ Calibration-compliant

❌ Only denoising = enforcing the prior model
❌ Very sensitive to stop
❌ Objective function unclear
Bayesian MAP Estimation

A principled way to introduce prior information
LASSO as a MAP estimator

\[
\arg\min I \frac{1}{2} \| V - \Phi I \|_2^2 + \lambda \| I \|_1
\]

- Convex optimization methods
- Sparse solutions \(\Rightarrow\) Well suited for **Point Sources**
Sparse Dictionary reconstruction

\[
\arg\min_\theta \frac{1}{2} \| \mathbf{V} - \Phi\Psi\theta \|_2^2 + \lambda \| \theta \|_1
\]

\[\Psi \in \mathbb{R}^{N \times M}\] Dictionary synthesis operator

\[\theta \in \mathbb{R}^M\] Dictionary coefficients
Optimization methods

✓ Denoising (with only one parameter!)
✓ Excellent reconstruction quality demonstrated
✓ Can handle very complex priors
✓ Fast principled algorithms
Optimization methods (continued)

- Denoising (with only one parameter!)
- Excellent reconstruction quality demonstrated
- Can handle very complex priors
- Fast principled algorithms

- Completely different implementation paradigm (proximal method)
- Memory scalability
- Non calibration-compliant
- Shrinkage of the reconstructed intensity

Denoising (with only one parameter!)
Excellent reconstruction quality demonstrated
Can handle very complex priors
Fast principled algorithms
03. PolyCLEAN

Convex optimization for RA enabled with an atomic method
1. Optimization method
Penalty-based prior

2. Atomic behavior
CLEAN-like algorithmic structure and minor cycles

3. Focus on Scalability
Sparsity-informed computations with Pycsou and NUFFT
1. Optimization method

Penalty-based prior

\[ \lambda \| I \|_1, I \geq 0 \]

2. Atomic behavior

CLEAN-like algorithmic structure and minor cycles

3. Focus on Scalability

Sparsity-informed computations with Pycsou and NUFFT
1. Optimization method

Penalty-based prior

\[ \lambda \| \mathbf{I} \|_1 , \mathbf{I} \geq 0 \]

2. Atomic behavior

CLEAN-like algorithmic structure and minor cycles

\[ \mathbf{I} = \sum \alpha_k \delta_{ik} \]

3. Focus on Scalability

Sparsity-informed computations with Pycsou and NUFFT
1. **Optimization method**

Penalty-based prior

\[ \lambda \| \mathbf{I} \|_1, \mathbf{I} \geq 0 \]

2. **Atomic behavior**

CLEAN-like algorithmic structure and minor cycles

\[ \mathbf{I} = \sum \alpha_k \delta_i k \]

3. **Focus on Scalability**

Sparsity-informed computations with Pycsou and NUFFT

\[ \Phi \left( \mathbf{I}^{(k)} \right) \]
The Algorithm

Algorithm 2 PolyCLEAN

Initialisation: $I^{(0)} = 0$, $S^{(0)} = \text{Supp}(I^{(0)}) = \emptyset$, $I_D = \Phi^* V$

while stopping_criterion($I^{(k)}$) not reached do

1. Compute the dirty residual: $I_R^{(k)} = I_D - \Phi^* \Phi I^{(k-1)}$

2. Place many candidate sources: $s_1^{(k)}, s_2^{(k)}, \ldots = \text{highest_level_set}(I_R^{(k)})$
   Update active set: $S^{(k)} \leftarrow S^{(k-1)} \cup \{s_1^{(k)}, s_2^{(k)}, \ldots\}$

3. Update the iterate:

$$I^{(k)} = \arg \min_{\text{Supp}(I) \subset S^{(k)}} \min_{I \geq 0} \frac{1}{2} \left\| V - I \right\|^2_2 + \lambda \left\| I \right\|_1 \quad \text{(R)}$$

end while
Focus on the sparse iterates

- Beneficial only if handled correctly
  - Low memory requirement
  - Simple model
  - Fast computation

- NU Fourier Transform: Type II $\rightarrow$ Type III

Symbiosis with HVOX
Optimization methods (continued)

- Completely different implementation paradigm (proximal method)
- Memory scalability
- Non-calibration-compliant
- Shrinkage of the reconstructed intensity

- √ Denoising (with only one parameter!)
- √ Excellent reconstruction quality demonstrated
- √ Can handle very complex priors
- √ Fast principled algorithms

Denoising (with only one parameter!) demonstrated excellent reconstruction quality and can handle very complex priors. Fast principled algorithms are also provided.
The Landscape of Methods

CLEAN

➕ L1 penalty and positivity
➕ Objective function
➕ Interpretability

Convex Optimization

➕ Atomic algorithm
➕ Scalability

PolyCLEAN
04. Numerical Results

It works, and it is fast
Performance benchmark

1. Pick an interferometer radius and image size
Performance benchmark

1. Pick an interferometer radius and image size
2. Simulate a source sky image
Performance benchmark

1. Pick an interferometer radius and image size
2. Simulate a source sky image
3. Simulate a measurement set
Performance benchmark

1. Pick an interferometer radius and image size
2. Simulate a source sky image
3. Simulate a measurement set
4. Solve with LASSO solvers:
   a. PolyCLEAN
   b. APGD
   c. MonoFW
Performance benchmark

1. Pick an interferometer radius and image size
2. Simulate a source sky image
3. Simulate a measurement set
4. Solve with LASSO solvers:
   a. PolyCLEAN
   b. APGD
   c. MonoFW
5. Solve with WS-CLEAN
Performance benchmark

1. Pick an interferometer radius and image size
2. Simulate a source sky image
3. Simulate a measurement set
4. Solve with LASSO solvers:
   a. PolyCLEAN
   b. APGD
   c. MonoFW
5. Solve with WS-CLEAN
6. Compare reconstruction time
Performance benchmark

1. Pick an interferometer radius and image size
2. Simulate a source sky image
3. Simulate a measurement set
4. Solve with LASSO solvers:
   a. PolyCLEAN
   b. APGD
   c. MonoFW
5. Solve with WS-CLEAN
6. Compare reconstruction time

Largest measured frequency
Image size (in px)
Pixel size
→ Observe scalability
Simulated Source image

200 point sources
5° x 5° FOV
Image size: 720 -> 5400 pixels
Reconstruction benchmark

720 x 720 px
~ 0.5 Mpx

5400 x 5400 px
~ 29 Mpx

Time comparison

- wsclean
- pclean
- apgd
- monofw

2360 s
650 s
296 s
203 s
140 s
Real world measurement

```python
<xarray.Visibility>
Dimensions: (time: 3595, baselines: 1953, frequency: 1,
             polarisation: 1, spatial: 3)
Coordinates:
* time  (time) float64 4.914e+09 4.914e+09 ... 4.914e+09
* baselines  (baselines) object MultiIndex
* antenna1  (baselines) int64 0 0 0 0 0 0 0 ... 58 59 59 59 60 60 61
* antenna2  (baselines) int64 0 1 2 3 4 5 6 ... 61 59 60 61 60 61 61
* frequency  (frequency) float64 1.458e+08
* polarisation  (polarisation) <U1 'I'
* spatial  (spatial) <U1 'u' 'v' 'w'
```

~ 400 MB
Selection of antennas

1/50 measurement times

24 antennas
Dirty image - Point Sources

1024 pixels = 6° FOV

\[ I_D = \Phi \ast V \]
### Reconstruction parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WS-CLEAN</strong></td>
<td></td>
</tr>
<tr>
<td>Auto-threshold parameter</td>
<td></td>
</tr>
<tr>
<td>3 $\sigma$</td>
<td></td>
</tr>
<tr>
<td>2 $\sigma$</td>
<td></td>
</tr>
<tr>
<td>1 $\sigma$</td>
<td></td>
</tr>
<tr>
<td><strong>PolyCLEAN</strong></td>
<td></td>
</tr>
<tr>
<td>Penalty parameter</td>
<td></td>
</tr>
<tr>
<td>5%</td>
<td></td>
</tr>
<tr>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>0.5%</td>
<td></td>
</tr>
</tbody>
</table>
Longest deconvolution example
Dual certificate

Definition:

\[ \mu_\lambda = \frac{1}{\lambda} \Phi^* (V - \Phi I^*) \]

Properties:

\[ \| \mu_\lambda \|_\infty \leq 1 \]

\[ \langle \mu_\lambda, I^* \rangle = \| I^* \|_1 \]

Usages:

- Convergence
- Saturation set
Dual certificate

Rate of information explained

\[ \left\| \mu_\lambda \right\|_\infty = \frac{\left\| \Phi^* (V - \Phi I^*) \right\|_\infty}{0.02 \left\| \Phi^* V \right\|_\infty} \leq 1 \]

Interpretation of the saturation set
Extended sources: Simulations

M31 image:

6.5 degrees → 256 pixels / side

Baselines:

rmax = 1000m → 31500 baselines
Extended Sources: Simulations

M31 image:
6.5 degrees $\rightarrow$ 256 pixels / side

Baselines:
$r_{\text{max}} = 1000\text{m} \rightarrow 31500$ baselines,
11 measurement times
Extended Sources: Simulations

M31 image:

6.5 degrees $\rightarrow$ 256 pixels / side

Baselines:

$r_{\text{max}} = 1000 \text{m} \rightarrow$ 31500 baselines,
11 measurement times
Reconstructions

Comparison restored sharp (components + residual)

Source

PolyCLean

CLEAN

MSE: $5.5 \times 10^{-2}$
MAD: $1.9 \times 10^{-1}$

MSE: $1.4 \times 10^{-2}$
MAD: $8.9 \times 10^{-2}$
Reconstructions (without res.)

Comparison components convolved sharp

Source

PolyCLEAN

CLEAN

MSE: $1.2 \times 10^{-2}$  
MAD: $4.6 \times 10^{-2}$

MSE: $1.0 \times 10^{-2}$  
MAD: $7.5 \times 10^{-2}$
Reconstructions (model)

Comparison components

Source

PolyCLEAN

CLEAN

MSE: $1.5 \times 10^{-4}$
MAD: $4.3 \times 10^{-3}$

MSE: $1.2 \times 10^{-3}$
MAD: $1.2 \times 10^{-2}$
Summary

1. Numerical performance
   - Scalability achievement
   - Optimization method up to speed with atomic method
   - Sparsity-based method

2. Versatility
   - Adaptable:
     - Tune the parameters
     - Fine control
   - Point and extended sources

3. Ongoing research work
   - **Dual certificate**: promising new scientific tool for RI image reconstruction
   - Room for improvement in the code as well as in the algorithm
   - Bayes estimation of the parameters
Thanks!

<table>
<thead>
<tr>
<th></th>
<th>CLEAN</th>
<th>MAP Estimation</th>
<th>PolyCLEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse iterates</td>
<td>✅</td>
<td>❌</td>
<td>✅</td>
</tr>
<tr>
<td>Flexible priors</td>
<td>~</td>
<td>✅</td>
<td>~</td>
</tr>
<tr>
<td>Fast solvers</td>
<td>✅</td>
<td>~</td>
<td>✅</td>
</tr>
<tr>
<td>Calibration compliant</td>
<td>✅</td>
<td>❌</td>
<td>✅</td>
</tr>
<tr>
<td>Interpretable obj. function</td>
<td>❌</td>
<td>✅</td>
<td>✅</td>
</tr>
</tbody>
</table>
Multi-scales dictionaries

Redundant dictionary:

\[ \Psi = [\Psi_1 \ldots \Psi_D] \quad \in \mathbb{R}^{N \times DN} \]
Multi-scales dictionaries

Redundant dictionary:

\[
\Psi = [\Psi_1 \ldots \Psi_D] \in \mathbb{R}^{N \times DN}
\]

- Wavelets:
  \[
  \Psi_d = \mathbf{W}_d \in \mathbb{R}^{N \times N}
  \]

- Gaussian kernels:
  \[
  \Psi_d \theta_d = \theta_d * g_{\sigma_d} \in \mathbb{R}^{N \times N}
  \]
A Frank-Wolfe Solver

- Convex optimization method: → Convergence guarantee
- Frank-Wolfe for atomic norm:
  → Atomic behavior
  → Sparse iterates
- Polyatomic variation:
  → Fast solver