

# P7 Compressed Sensing Image Reconstruction for CASA

Jonas Schwammberger

January 13, 2019

## **Abstract**

abstract plcaceholder

## Contents

<b>1</b>	<b>Compressed Sensing Image Reconstruction for MeerKAT</b>	<b>1</b>
1.1	The basic Measurement Equation of a radio interferometry . . . . .	1
1.2	The Major Cycle Architecture . . . . .	2
<b>2</b>	<b>Challenges for imaging MeerKAT data</b>	<b>4</b>
2.1	Wide Field of View Imaging and the third Fourier Component . . . . .	4
2.1.1	State of the art: $w$ -stacking algorithm . . . . .	5
2.2	Self-Calibration . . . . .	6
<b>3</b>	<b>Eliminating the Major Cycle</b>	<b>7</b>
3.1	Separating the Major Cycle into two Optimization Problems . . . . .	7
3.2	Reconstruction on the Sphere . . . . .	8
3.3	Explicit Fourier Transform Matrix . . . . .	8
<b>4</b>	<b>Compressed Sensing with Coordinate Descent</b>	<b>10</b>
4.1	Starlet Transform . . . . .	11
4.1.1	Active set heuristic with Starlets . . . . .	12
4.2	Coordinate Descent Implementation . . . . .	13
4.2.1	Iteration scheme . . . . .	13
4.2.2	Non-uniform FFT approximations . . . . .	14
<b>5</b>	<b>Results and scalability of Coordinate Descent</b>	<b>15</b>
5.1	Imaging on Simulated Data . . . . .	15
5.1.1	Mixed Sources . . . . .	15
5.2	Scalability Comparison . . . . .	16
5.2.1	Lower Bound Coordinate Descent . . . . .	16
5.2.2	CLEAN . . . . .	17
5.3	evaluation . . . . .	17
<b>6</b>	<b>Conclusion</b>	<b>18</b>
<b>7</b>	<b>Ehrlichkeitserklärung</b>	<b>22</b>

# 1 Compressed Sensing Image Reconstruction for MeerKAT

An instrument in the real world measures noisy data. Measurements are corrupted by noise, interference sources or the measurement instrument itself. Image reconstruction problems appear when one tries to remove the corruption from the measurements and tries to find the observed image of the instrument. This leads to an ill-posed inverse problem: A small change in the measurements may create very different image reconstructions, and many possible images match the measurements. An image reconstruction algorithm therefore has to find the observed image from a potentially large set of possible images.

In the past, image reconstructions applied simple heuristics and approximated a likely image. How close the approximation was to the observed image was in general not known. The theory of compressed sensing[3][4] introduced a new theoretical framework under which image reconstructions can be analysed. This has lead rise to new compressed sensing reconstruction algorithms, which under the right conditions are guaranteed to reconstruct the observed image. Furthermore they have shown super-resolution performance in real-world environments, creating reconstructions above the accuracy limit of the instrument.

This work applies compressed sensing image reconstruction to the field of radio astronomy. The new MeerKAT radio interferometer poses a reconstruction problem on a new scale of data volume. The raw measurements easily take up several terabytes of disk space. The focus of this work is creating a scalable compressed sensing image reconstruction algorithm.

The current state of the art reconstruction algorithm for MeerKAT is based on CLEAN[1][2]. It is a reconstruction using a simple heuristic and was developed before the theory of compressed sensing was known. In recent years, compressed sensing reconstruction algorithms were developed for radio interferometers[5][6][7]. They beat CLEAN in terms of reconstruction quality, producing super-resolved reconstructions. However, CLEAN has the upper hand in runtime complexity. Therefore, an efficient implementation of CLEAN is still the go to reconstruction algorithm for MeerKAT data.

The efficient implementation use CLEAN in the Major Cycle Architecture, which was developed with CLEAN in mind. Current compressed sensing algorithms use a similar architecture. So far, little research has gone into different architectures for compressed sensing reconstructions in radio astronomy. This work explores different architectures for compressed sensing reconstructions with the hope of reducing the computational complexity.

A new proof-of-concept reconstruction algorithm was developed with a simplified architecture. It was tested on simulated MeerKAT data and the lower-bound asymptotic complexity was evaluated. The new algorithm scales independently of the image size, but scales worse with the number of input measurements compared to CLEAN. For the MeerKAT image reconstruction, the number of input measurements tends to be the largest of all numbers in the problem. Even though the algorithm could be improved further, it is unlikely to beat CLEAN reconstructions in terms of runtime complexity on MeerKAT data.

## 1.1 The basic Measurement Equation of a radio interferometry

Real world radio interferometers have complicated measurement equations. They become even more complicated for large interferometers like MeerKAT. These problems get addressed in section 2. This section looks at the basic measurement equation of a radio interferometer (1.1) and discusses the two fundamental challenges for radio interferometry image reconstruction.

$$V(u, v) = \iint I(x, y) e^{2\pi i(ux+vy)} dx dy \quad (1.1)$$

An interferometer measures Fourier Components  $V$  (called Visibilities in Radio Astronomy) from the sky image  $I$  at position  $x$  and  $y$ . The term  $e^{2\pi i(ux+vy)}$  represents the two dimensional Fourier Transform. The task is to reconstruct the observed image  $I$  from the measured Visibilities  $V$ . In theory this task is trivial: Since the inverse Fourier Transform exists, we can reconstruct the image  $I$  by calculating the inverse Fourier Transform of  $V$ . However, two properties of the Visibilities make this task challenging in practice:

1. Non-uniform sampling pattern in Visibility space
2. Incomplete Visibility coverage.

*Property 1:* We want to reconstruct an image with uniformly spaced pixels. The instrument defines the sampling pattern in Visibility space and does not correspond to the exact pixels of the reconstructed image. This property keeps us from using the Fast Fourier Transform. The naive inverse Fourier Transform can still be calculated, but it has a quadratic runtime and does not scale to the data volume of interferometers. Current reconstruction algorithms use the non-uniform Fast Fourier Transform. The non-uniform FFT approximates the non-uniform Fourier Transform.

*Property 2:* Interferometers sample only a limited set of Visibilities. It does not have all information for reconstruction. When the inverse Fourier Transform is applied on the Visibilities, the resulting image is corrupted by the incomplete Visibilities. It contains structures which were introduced by the interferometer and were not observed. With only knowing the incomplete set of Visibilities a reconstruction algorithm has to decide which image structures were truly measured, and which are due to the instrument. This forms an ill-posed inverse problem. There are many images that fit the measurements, and a small change in the Visibilities can lead to a very different reconstruction.

CLEAN represents the instrumental effect with a Point Spread Function (PSF). After the non-uniform FFT produced the 'dirty' image, CLEAN tries to reconstruct observed image with a deconvolving the 'dirty' image with the PSF. Note however that both CLEAN nor the non-uniform FFT are approximations. In real world reconstructions, these two approximations are used in the major cycle architecture to increase the reconstruction accuracy.

## 1.2 The Major Cycle Architecture

Major cycle was created with CLEAN in mind. Compressed sensing reconstructions use essentially the same architecture with minor modifications.

A CLEAN image reconstruction for radio interferometers consists of two different steps: A non-uniform FFT, which approximates the inverse Fourier Transform efficiently and an deconvolution algorithm, which approximates the instrumental effects on the image (typically based on CLEAN). These two approximations are an error source of the reconstruction. The major cycle architecture 1 therefore tries to iteratively minimize the errors of the non-uniform FFT and the deconvolution over several iterations.

The first major cycle uses the non-uniform inverse FFT to approximate the 'dirty' image from the Visibilities. The deconvolution algorithm decides parts which parts belong to the observed image and which are due to instrumental and other noise effects. It returns the noise part of the image, which gets transformed back into residual Visibilities. The next major cycle iteration continues with the

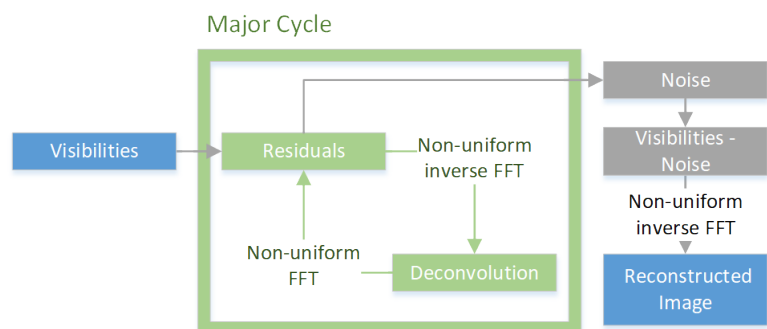


Figure 1: The Major Cycle Framework

residual Visibilities. The residual Visibilities get minimized until they contain only the instrumental effects and noise.

After several major cycles the residual on a regularly spaced image which has a small error from non-uniform samples, and a small error from incomplete measurements.

Compressed sensing reconstructions essentially use the same architecture, although with minor alterations. In general, they keep a similar scheme of forward/backward non-uniform FFT, but do not use a deconvolution in image space. Instead, they analyse constraints defined on the image (For example all pixels should be non-negative), and try to find the Visibilities that minimize the constraint violations.

For CLEAN reconstructions, the forward/backward non-uniform FFT are the most expensive operations, while the deconvolution is negligible. The compressed sensing algorithms tend to require more major cycle iterations to converge and the image constraint analysis tends to be more expensive than CLEAN deconvolutions. Current research into compressed sensing reconstructions is focussed on reducing the number of major cycles[6].

## 2 Challenges for imaging MeerKAT data

New interferometers like MeerKAT put additional challenges on the image reconstruction problem. First, the new instrument produce a new magnitude of data, forcing reconstruction algorithms to be highly scalable and distributable. Second, the more sensitive instruments with large field of view amplify effects which were negligible in older instruments, like non-coplanar baselines or the ionosphere.

Compressed sensing has to be able to work with these issues somehow. Solutions all in the context of the major cycle, a new architecture may need to handle effects differently. Hopefully in a manner that is at least as efficient.

In this work, the effects of non-coplanar baselines gets handled in more detail. The effect has a neat mathematical notation, it adds a third fourier component to the measurement equation, but breaks the two dimensional fourier relationship introduced in section 1.1.

Calibration used to be a task before image reconstruction. Calibration gets not explicitly handled here, but it is important to keep in mind. For older interferometers, the data was calibrated before image reconstruction. With the advent of self-calibration, the image reconstruction is also used to improve calibration. MeerKAT requires more calibration parameters, so an image reconstruction algorithm has to be able to calibrate, or it is not interesting.

Further issues that do not get handled here

- (Beam Pattern, A Projection)
- Full polarization
- Wide band imaging

There are several challenger for imaging meerkat data. One problem is the new amount of data.

terabytes of measurements. Large image size 32k squared are the obvious problems to solve. Distributing the problem is not part of this work.

In this work, it is focused on Wide field of view issue.

### 2.1 Wide Field of View Imaging and the third Fourier Component

The measurement equation of radio interferometers contains a third fourier component, the  $w$  component. This leads to the following measurement equation (2.1).

$$V(u, v, w) = \int \int \frac{I(x, y)}{\sqrt{1 - x^2 - y^2}} e^{2\pi i [ux + vy + w(\sqrt{1 - x^2 - y^2} - 1)]} dx dy \quad (2.1)$$

For small field of view observations, the term  $\sqrt{1 - x^2 - y^2} \approx 1$ , and we arrive at the basic measurement equation (1.1). For small field of view, the  $w$  term can be ignored, and the image can be calculated with the inverse two dimensional non-uniform FFT. For wide field of view measurements, the  $w$  term breaks the two dimensional Fourier relationship. The measurement equation (2.1) can still be inverted, but the resulting algorithm has a quadratic runtime and is not feasible in practice.

Note that the image  $I(x, y)$  of the wide field of view the measurement equation (2.1) is still two dimensional, even though the Visibilities  $V(u, v, w)$  have three. The relationship between Visibilities and image is neither the two, nor the three dimensional Fourier transform. The Visibilities represents an image on the celestial

sphere. This means the image is naturally curved, and the  $w$  component represents the distortion introduced by the curvature.

One can simply ignore the  $w$  component, ignore the curvature and use the two dimensional Fourier Transform. The image then does not lie on a sphere, but on a tangent plane. At the tangent point (where  $\sqrt{1 - x^2 - y^2} \approx 1$ , typically the image center in radio astronomy) the two reconstructions are identical. But the distortion gets more severe away from the tangent point. The image 2a shows a tangent plane reconstruction over a large field of view, while 2b shows the same reconstruction but  $w$  corrected. Close to the tangent point, the image center, the reconstructions both locate the point sources. At the image edges, the distortion gets severe enough to decorrelate the Visibilities and the point sources get 'torn' apart.



Figure 2: Celestial sphere distortion on simulated data. Source: [8]

Image reconstructions for MeerKAT need to account for the  $w$  component efficiently. In the Major Cycle framework, this is part of the non-uniform FFT. The first approach was to image facets, creating several tangent planes for the celestial sphere. A more efficient approach came in the form of the  $w$ -projection algorithm[8]. It uses convolution on the Visibilities during the non-uniform FFT. MeerKAT currently uses the  $w$ -stacking algorithm, which more efficient than  $w$ -projection on MeerKAT's data volume and is simple to distribute.

This is still an active field of research. The latest work as of time of writing is a hybrid approach[9] with both  $w$ -stacking and  $w$ -projection. As of time of writing, it was not yet adapted.

### 2.1.1 State of the art: $w$ -stacking algorithm

Here a short explanation of the  $w$ -stacking algorithm is provided. It is a surprisingly simple algorithm that seems to be here to stay. It is a modification of the nuFFT operation displayed in figure 1.

*Inverse nuFFT with  $w$ -stacking:* Visibilities get grouped into stacks by their  $w$ -term. Every visibility inside a stack has a similar  $w$ -term. The more stacks, the more accurately you can handle the  $w$  component. The figure displays the relation

The nuFFT gets called on each stack separately, and a constant  $w$  correction factor applied. Since  $w$ -term is constant, we can use the two dimensional nuFFT again. At the end, all the stacks are summed together to form the dirty image.



*nuFFT with w-stacking*: The inverse is done in a similar fashion. The sum cannot be re-separated, so the image gets copied into each stack. The constant  $w$  factor gets divided out. The nuFFT gets applied.

the runtime

(talking about more recent hybrid approach, limiting the total number of  $w$ -stacks?)

## 2.2 Self-Calibration

Complex gain term. Corrects amplitude and phase.

Traditional Calibration

A calibration source close by, with a known brightness. Phase and amplitude calibration was done before image reconstruction. For older interferometers, there was a very limited number of calibration terms to solve.

but again effects of wide field of view also increase the number of necessary calibration terms. The advent of self calibration, in which the image reconstruction was used to solve for both, the observed image and the calibration of the instrument.

Self calibration with the major cycle algorithm and a CLEAN deconvolution.

Initial phase calibration Shallow clean phase calibration deep clean phase and amplitude calibration deep clean reconstruction

### 3 Eliminating the Major Cycle

There is little research in exploring a different architecture for Compressed Sensing. The closest to this work was done by Pratley et al[10], which optimizes the non-uniform FFT in the context of compressed sensing. The Major cycle is still used. On the upside, progress in w-term accuracy and runtime.

There are ways to form a different architecture. However, the advent of self-calibration pushes a forward-backward kind of architecture on the reconstruction.

#### 3.1 Separating the Major Cycle into two Optimization Problems

The major cycle reduces two errors simultaneously. The error introduced by the non-uniform FFT approximation, and the error introduced by the deconvolution approximation. In a sense, the major cycle is a simple optimization algorithm: It iteratively calls the non-uniform FFT and deconvolution approximations with the error from the last iteration, until the error is below a certain threshold.

$$FT : \underset{I_{dirty}}{\text{minimize}} \|V - A^{-1}I_{dirty}\|_2^2 \quad (3.1)$$

$$\text{deconvolution} : \underset{x}{\text{minimize}} \|I_{dirty} - x \star PSF\|_2^2 + \lambda \|P(x)\|_1 \quad (3.2)$$

The major cycle can also be separated into two objectives (3.1) and (3.2), and solved in sequence. The first objective (3.1) tries to find the best uniformly spaced image  $I_{dirty}$  matching the observation.  $A^{-1}$  represents the non-uniform inverse FFT approximation ( $A^{-1}I_{dirty} \approx F^{-1}I_{dirty}$ ). When this is solved, the original Visibility measurements are not needed any more. Then, the second objective (3.2) searches for the deconvolved image  $x$  regularized by some prior  $P()$ .

By separating the two objectives, the hope is we can use specialized optimization algorithms that overall converge faster than several major cycles. Also, after the first objective was minimized, we can drop the Visibilities, which can be magnitudes larger than the image.

At first glance, this is a sensible idea, but the devil is in the objective (3.2). Namely in the  $PSF$  is not constant, it varies over the image. In the Major Cycle Architecture, a constant  $PSF^1$  is sufficient: It only needs to approximate the deconvolution. The error introduced by a constant  $PSF$  can be reduced in the next Major Cycle. By eliminating the Major Cycle, we also eliminated the chance to approximate the deconvolution with a constant  $PSF$ . In the worst case, we would need to estimate the  $PSF$  for each pixel in the image.

In the worst case, these many  $PSF$ s may stop us from using this approach. It is plausible that can use fewer  $PSF$ s and be close enough. No known work done in the radio astronomy. The Major Cycle is ubiquitous. But it is plausible.

However, the problem is that the imaging algorithm should also be able to calibrate. With this approach, there is a strict one way flow from measurements to image. For calibration, a backwards flow from image to measurements is necessary.

For calibrated data, this approach may be potentially faster. but since MeerKAT will be difficult to calibrate correctly, this is unlikely.

---

<sup>1</sup>CLEAN implementations in Radio Astronomy typically use a constant  $PSF$ . This is not strictly necessary, one can also implement CLEAN with a varying  $PSF$ .

### 3.2 Reconstruction on the Sphere

The signal naturally lives on the celestial sphere.

Not that much used in Radio Astronomy reconstructions. Work that uses to project on the sphere and looks at the reconstruction quality [11].

The wide field of view measurement equation (2.1) can be expanded into spherical harmonics.

[12] measurement equation with spherical harmonics. Essentially replaces the non-uniform FFT from the Major cycle with a new approach based on the fact that it has a solution to the Helmholtz equation. , see [12] for further detail. It can be used as a different way to arrive at a dirty image from wide field of view interferometers. It is useful in full sky image, because it fulfills the (property from paper) and the full sky dirty image does not have artifacts. It does not gain any complexity advantage, quite the contrary.

Even if the spherical harmonics transform would gain a speed advantage, used in the way of [12] it would still be the same major cycle architecture, just with spherical harmonics transform.

However there is the possibility to just do everything on the sphere.

Spherical haar wavelets used for simulating visibilities from a known image, essentially the opposite operation from imaging. [13]

based on the spherical harmonic transform. Doing a Spherical harmonic transform efficiently is still an active field of research [14]

transform . The wide field of view measurement equation (2.1) can be factorized into spherical harmonics. This means spherical harmonics are equivalent, calculating an image from the measurements with the three dimensional fourier transform is the same as using the spherical harmonics transform. [15]

[16]sampling theorem on a sphere

Spherical harmonics were researched in the context of Compressed Sensing image reconstructions (cite wiaux). They improved the quality of the image reconstruction, but no speed advantage.

The push on solving on the sphere directly. Using spherical haar wavelets, (cite) was able to speed up the simulation problem. Simulation tries to solve to problem of finding the measurements corresponding to a given image  $x$ . However, this has so far not been used for imaging.

### 3.3 Explicit Fourier Transform Matrix

Hardy et al[17] handled the whole inverse Fourier Transform as an explicit matrix  $F^{-1}$  of size  $M * N$ . They dropped the non-uniform FFT and  $w$ -stacking approximations. This is trivially to distribute later. The only problem is of Course that for MeerKAT data sizes  $M * N$  is too large (4gb of visibilities times 1 Billion Pixels).

However there is potential for improvement: Only a limited number of pixels in a reconstruction are not zero. We spent a lot of processing power on pixels which do not matter in the reconstruction. If we could predict which pixels are not zero, we would only need to calculate a subset of  $F^{-1}$  columns to reconstruct the image. The starlet transform[18] has a way to predict its non-zero components. We represent the image as a combination of starlets, and estimate which starlets are likely non-zero from the Visibilities directly.

To my knowledge, such an algorithm has never been tried out.

A proof-of-concept algorithm was developed that shows the prediction actually works in practice. It uses Coordinate Descent with the starlet transform, and only needs to calculate a subset of the matrix  $F^{-1}$  for a

reconstruction. The question remains how many columns of  $F^{-1}$  are needed, and is it more efficient than an algorithm using the major cycle architecture.

## 4 Compressed Sensing with Coordinate Descent

Doing compressed Sensing with Coordinate Descent and starlet regularization.

The idea behind compressed sensing is we use prior information about the image, and find the most likely image  $x$  from all possible solutions. In practice, we formulate a minimization problem with a data and regularization term. The data term forces the image to be as close to the measured Visibilities as possible, and the regularization term tells us how likely the image is to be true.

The regularization is responsible for an accurate reconstruction. In this work, two regularizations are used. The L1 pixel regularization is used to demonstrate a simple version of the coordinate descent algorithm. For the actual algorithm starlet regularization is used. Starlets have been used as regularization for the LOFAR interferometer[5], which was able to produce super-resolved reconstructions.

First, let us look at the coordinate descent algorithm in its simplest form, and discuss its properties. we want to minimize the objective (4.1), where the data term  $\|V - F^{-1}x\|_2^2$  forces the image to be close to the Visibilities  $V$ , and the regularization term  $\|x\|_1$  forces the image to have as few non-zero pixels as possible. The parameter  $\lambda$  represents our trade-off between reconstruction accuracy and regularization.

$$\underset{x}{\text{minimize}} \quad \|V - F^{-1}x\|_2^2 + \lambda \|x\|_1 \quad (4.1)$$

The objective function (4.1) has a property, which we can exploit with Coordinate Descent: The optimum for a single pixel, if we keep all others fixed, turns out to be a parabola. The optimum for a single pixel of (4.1) is simply finding the apex of a parabola, followed by a shrink operation<sup>2</sup>. Sadly, the pixels are not independent of each other, and we need multiple iterations over all pixels to converge. This leads to the following reconstruction algorithm, written in python code:

```

1 def coordinate_descent(V_residual, x, lambda, max_iter):
2     for k in range(0, max_iter):
3         for i in pixels_row:
4             for j in pixels_column:
5                 x_old = x[i, j]
6                 fourier_column = calculate_fourier_transform(i, j)
7                 fr = real(fourier_column)
8                 fi = imag(fourier_column)
9                 rr = real(V_residual)
10                ri = imag(V_residual)
11
12                #find apex
13                a = sum(fr**2 + 2*fr*fi + fi**2)
14                b = sum(fr*rr + fr*ri + fi*rr + fi*ri)
15                x_new = b / a + x_old
16
17                x_new = shrink(x_new, lambda)
18                x[i, j] = x_new
19                V_residual = V_residual - fourier_column * (x_new - x_old)

```

The simple version of coordinate descent converges to a solution which is similar to CLEAN. The convergence rate of coordinate descent is not well understood. Our image reconstruction problem falls in the class of quadratic programming, in coordinate descent converges at least linearly[19]. In practice, the convergence

---

<sup>2</sup>The shrink operation reduces the magnitude of the pixel by  $\lambda$ . For example: Pixel  $x = -12.4$ ,  $\lambda = 0.6$ . The new pixel value after shrinkage follows as  $\text{shrink}(-12.4, 0.6) = -11.9$

rate can be sped up with heuristics like active set[?] and  $\lambda$ [?]. Furthermore, coordinate descent is robust, even when the residual vector is only approximated, which is useful for distributing the algorithm.

The simple approach calculates the whole Fourier Transform Matrix  $F^{-1}$  explicitly. The upside is that we do an exact  $w$ -term correction. The downside is  $F^{-1}$  is too large to keep in memory, and too expensive to calculate on the fly for the scale of MeerKAT reconstructions.

The starlet transform is used from here on. Instead of reconstructing the image directly, we represent the image as a combination of starlets  $x = D\alpha$ . Where  $D$  is an over-complete dictionary of starlets, and  $\alpha$  is the starlet component vector. Since the dictionary  $D$  is over-complete, it is a matrix with more columns than rows, and does generally not have an inverse (more formally:  $D^{-1}x \neq \alpha$ ). However, the starlet transform does have an pseudo-inverse. In this work, the pseudo inverse is used to only look at a subset of starlet components which are likely to be non-zero, and only evaluate a subset of columns of  $F^{-1}$ .

Actual implementation

## 4.1 Starlet Transform

First how the starlet transform works, and then we discuss how we can leverage the properties to only use

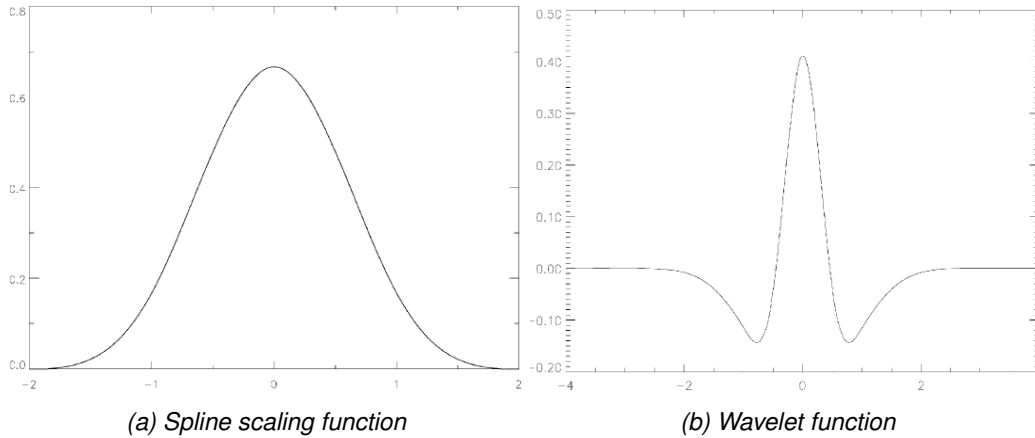


Figure 3: Scaling and wavelet function of the starlet wavelet. Source: [18]

The starlet transform is an Isotropic Undecimated Wavelet Transform (IUWT) based on the starlet wavelet shown in figure 3. The IUWT decomposes an image into multiple layers, each representing parts of the image at different scales. The lower layers contain small structures like point sources, while the higher layers contain extended emissions like hydrogen clouds. The two transforms are shown in equation (4.2) and (4.3). The transformation from starlets to image  $x$ , shown in (4.2), is a simple addition of all  $J$  layers. The transformation from image  $x$  to starlet space, shown in (4.3) is done with a multi-scale wavelet decomposition.

$$x = w_0 + w_1 + \dots + w_{J-1} + c_J \quad (4.2)$$

$$\begin{aligned} c_0 &= x \star S_0 & c_1 &= c_0 \star S_1 & \dots & c_{J-1} &= c_{J-2} \star S_{J-1} & c_J &= c_{J-1} \star S_J \\ w_0 &= x - c_0 & w_1 &= c_0 - c_1 & \dots & w_{J-1} &= c_{J-2} - c_{J-1} \end{aligned} \quad (4.3)$$

Let us look at the multi-scale wavelet decomposition in more detail.  $w_i$  represents the starlet components at layer  $i$ , where  $w_0$  contains the smallest structures in the image, and  $w_{J-1}$  contains the largest structures.

Note that the starlets  $w_i$ ,  $c_i$  and the image  $x$  all have the same dimensions.  $c_i$  contains an ever increasingly blurred version of the image  $x$ .  $c_i$  represents the image convolved with all the scaling functions  $S_0, S_1, \dots, S_i$ , where  $S_i$  is the scaling function shown in figure 3a of the layer  $i$ . The scaling function increases in size for each successive layer.

The decomposition starts iteratively, blurs the image  $x$  with the smallest scaling function  $S_0$ .

The last, heavily blurred  $c_J$  contains the structures which were too big to represent with the largest starlets, and is the last layer of the decomposition.

This is an over-complete representation. Many different components can explain the same pixel.

A trous algorithm

#### 4.1.1 Active set heuristic with Starlets

The multi-scale decomposition can be used to estimate the which starlet components  $w_i$  are likely to be non-zero. The figure 4 shows an example of the process. Each "pixel" in 4b represents a starlet at scale 0, i.e. point sources. We can apply the shrink operator on the  $w_0$  map, is essentially a parameter estimation. This property is used in the SASIR[5] reconstruction, which uses the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) for optimization.

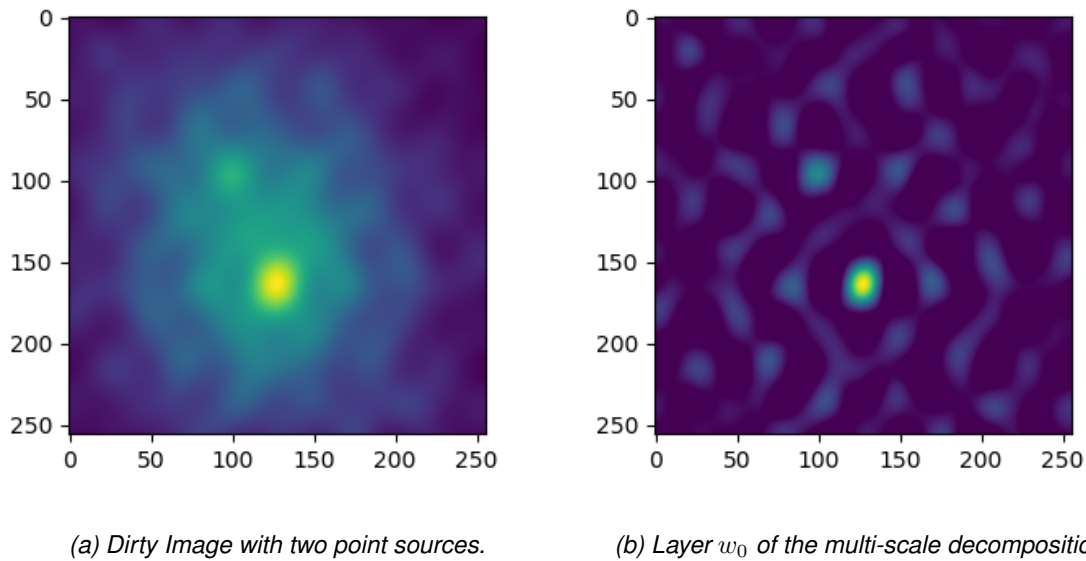


Figure 4: Dirty map and  $w_0$  starlet layer. The right image shows the probability distribution for point sources in the dirty image.

For our purposes, we use the result of the decomposition as a probability distribution. The higher the values of the decomposed  $w_i$ , the more likely it is to be non-zero.

the X highest values of the  $w_i$  decomposition are put into the active set. Only components in the active set get optimized.

Note however that the different locations are dependent on each other. The image 4b shows two areas for the two point sources. From the  $w_0$  map alone it is hard to know if the center blob is due to a single point source, or multiple point sources in a small area.

## 4.2 Coordinate Descent Implementation

Proof of concept implementation. There are many ways to improve the efficiency and the convergence speed. It is here to show the general if it is possible of the approach and if we can truly only calculate a subset of  $F^{-1}$  columns.

The objective function (4.4) minimizes the starlet components  $\alpha$ . This objective leads again to a parabola in the one dimensional case, and we can analytically find the optimum for a single component when all others are fixed.

$$\underset{\alpha}{\text{minimize}} \quad \|V - F^{-1}D\alpha\|_2^2 + \lambda \|\alpha\|_1 \quad (4.4)$$

There are still a few problems to solve, like how to handle the matrix product  $F^{-1}D$ , With the active set heuristic, we iterate over a limited number of starlet components at a time.

The question remains, how the matrix product  $F^{-1}D$  can be calculated efficiently. Since starlet transform is just made up of a couple of convolutions, we do not have to calculate the product  $F^{-1}D$  explicitly. Convolutions in image space are multiplications in Fourier space. So for any starlet layer  $w_i$  or  $c_J$ , we can pre-calculate a transform vector.  $w_2 = F^{-1}V * (\hat{S}_0\hat{S}_1 - \hat{S}_0\hat{S}_1\hat{S}_2)$

How many layers are needed. starlets rise in pixels with  $2^J$ . Meaning if we want starlets over the whole image, the number of starlets rise logarithmically to the image dimensions.

How to iterate over the image in detail.

### 4.2.1 Iteration scheme

Many ways to iterate over the image

```

1 def coordinate_descent(V_residual, x, lambda, max_iter):
2     for k in range(0, max_iter):
3         for i in pixels_row:
4             for j in pixels_column:
5                 x_old = x[i, j]
6                 fourier_column = calculate_fourier_transform(i, j)
7                 fr = real(fourier_column)
8                 fi = imag(fourier_column)
9                 rr = real(V_residual)
10                ri = imag(V_residual)
11
12                #find apex
13                a = sum(fr**2 + 2*fr*fi + fi**2)
14                b = sum(fr*rr + fr*ri + fi*rr + fi*ri)
15                x_new = b / a + x_old
16
17                x_new = shrink(x_new, lambda)
18                x[i, j] = x_new
19                V_residual = V_residual - fourier_column * (x_new - x_old)

```

---

coordinate descent with active set heuristic



#### **4.2.2 Non-uniform FFT approximations**

Non uniform FFT to calculate the convolution KERNELS!!

Stupid approach with line search. Could be done more efficiently by using the histogram of the starlet level.

## 5 Results and scalability of Coordinate Descent

Two Tasks: Show that coordinate descent works, and show its scalability. To show that it works, it was tested on two simulated MeerKAT data sets and compared to clean. The simulation was created with the Meqtrees Software package, using default settings. The simulations contain [only little gaussian noise on the Visibiliites]. The simulated data is therefore perfectly calibrated, the image reconstruction has to deal with the two fundamental issues of incomplete measurements and non-uniform sampling only.

Scalability is harder to analyse for Coordinate Descent. As mentioned in previous chapters, it does not have fast convergence guarantees. But with the heuristics but it is ripe for heuristics that work well for the special case. Therefore it was analysed what an ideal Coordinate Descent implementation would need in terms of complexity.

Two simulations, one of point sources and one of mixed sources.

### 5.1 Imaging on Simulated Data

$\lambda$  was fixed on both tests

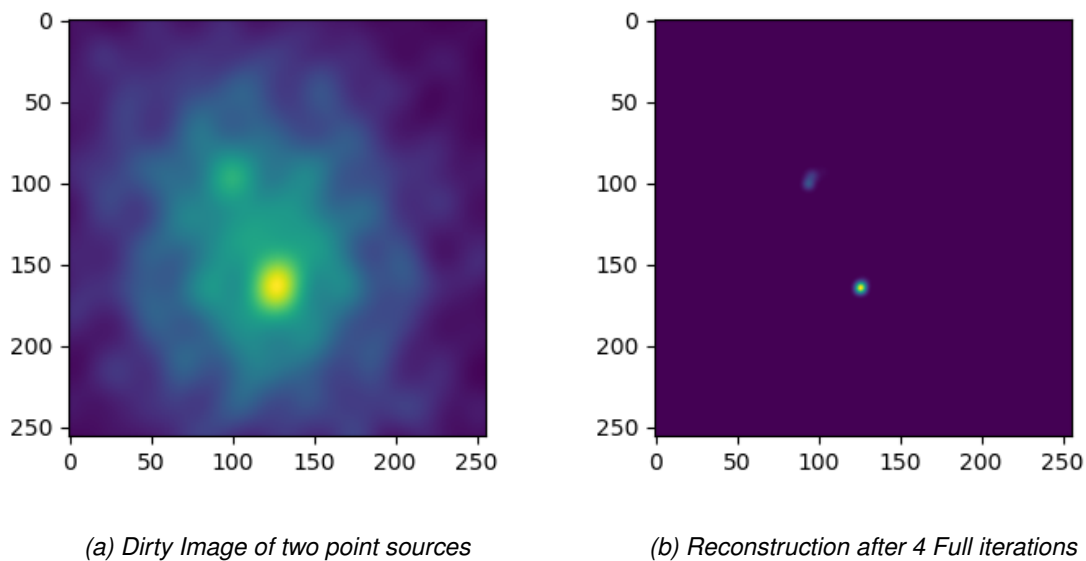


Figure 5: Image reconstruction on two point sources.

faint source is spread more widely. Not good, The paper [18] used different  $\lambda$  values for different starlet levels. Maybe with this it can be forced to do more precise results, but it was not tried in this work.

#### 5.1.1 Mixed Sources

Mixture of gaussian and point sources.

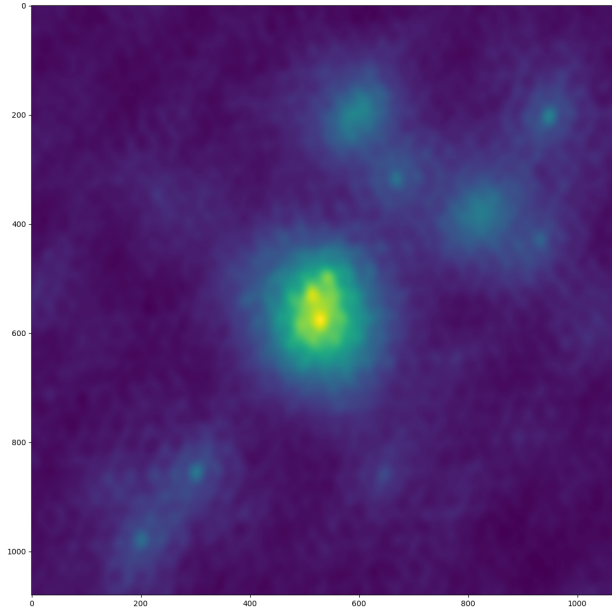


Figure 6: Dirty Image

## 5.2 Scalability Comparison

Bunch of heuristics. There are a lot of little ways to optimize this algorithm. The question is, is it worth going further and try to improve the algorithm further. So we want to estimate the lower bound of the algorithm.

Convergence is hard to determine in general. Following simplified assumptions were made: We have a heuristic with oracle performance. It returns only the locations of  $\alpha$  in constant time. Furthermore, we assume all axes are independent from each other, only one descent per non-zero axis is necessary.

### 5.2.1 Lower Bound Coordinate Descent

How many pixels need to be evaluated. Oracle performance  $S$

$$res * starlet = M \text{ descent: } genfcol = 3 * M \quad a = 3 * M \quad b = 4 * M \text{ residuals, } fcol * diff = M$$

$$total_{mitgencol} = 11M$$

$$CD(M, S, J) = S * 11M + J * 2M \quad (5.1)$$

$S$  depends on the image content directly. For example if the image contains 15 point sources and five Gaussian extended emissions, then  $S$  equals 20 non-zero components (if we assume the Gaussian sources require only one starlet for representation). Coordinate Descent therefore is independent of the image size  $N$ . It solely depends on the size of the measurements  $M$ , and the number of non-zero components in the dictionary  $S$ .

It does not use any approximation for the fourier transform.

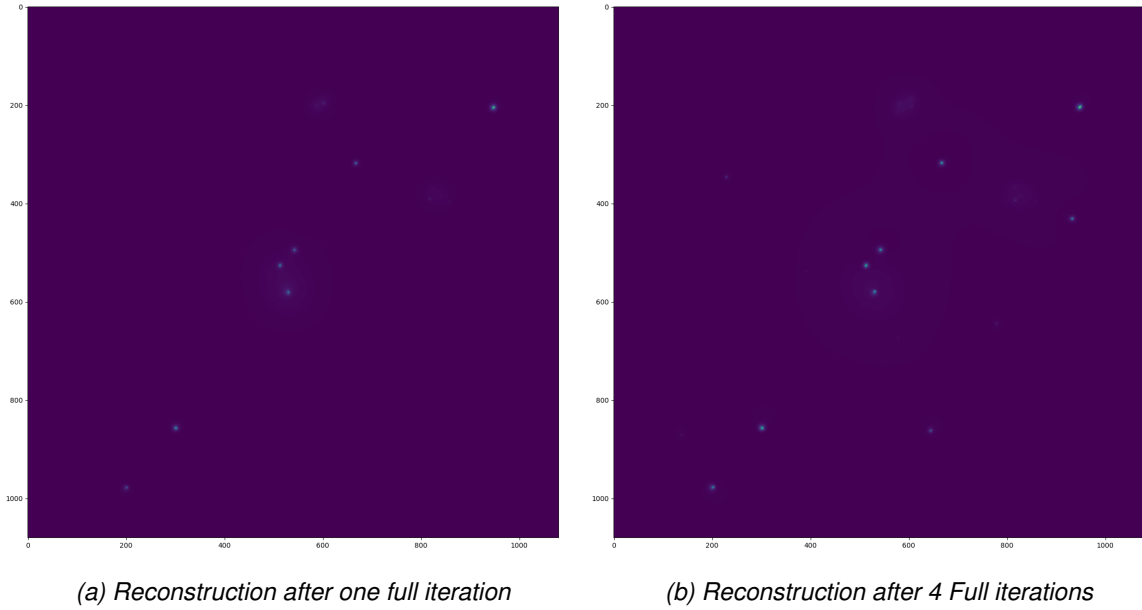


Figure 7

### 5.2.2 CLEAN

non-uniform FFT :  $M + 2N * \log(2N)$

w-stacking :  $M + W * (2N * \log(2N) + 2N) + N * \log(N)$

CLEAN deconvolution :  $D * 2N$

(Cite New W-stacking approach) Nufft:  $M + 2N \log 2N$  W-stacking =  $M + W * (2N \log 2N + 2N) + N \log N$   
Deconvolution =  $D * 2N$

$$CLEAN(M, M_{Major}, N, W, D) = M_{Major} * [M + W * (2N \log 2N + 2N) + N \log N + D * 2N] \quad (5.2)$$

Complex formula. But for MeerKAT, the number of Visibilities  $M$  is by far the largest. Simplifies to  $M_{Major} * M$   
Other compressed sensing approaches have a more expensive term than  $D * 2N$  for their complexity. The major cycle algorithm depends on more parameters. Assumptions were made in favor of Coordinate Descent.

the number of w-stacks  $W$ . If this is the case

[9] fast w-stacking w-projection

### 5.3 evaluation

## 6 Conclusion

Generally not a lot of published research into this area

The high number of visibilities compared to a low number of pixels

Calibration sorta creates the effect of needing a major cycle.

No obvious contenders for a different architecture

## References

- [1] JW Rich, WJG De Blok, TJ Cornwell, Elias Brinks, Fabian Walter, Ioannis Bagetakos, and RC Kennicutt Jr. Multi-scale clean: A comparison of its performance against classical clean on galaxies using things. The Astronomical Journal, 136(6):2897, 2008.
- [2] Urvashi Rau and Tim J Cornwell. A multi-scale multi-frequency deconvolution algorithm for synthesis imaging in radio interferometry. Astronomy & Astrophysics, 532:A71, 2011.
- [3] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on information theory, 52(2):489–509, 2006.
- [4] David L Donoho. Compressed sensing. IEEE Transactions on information theory, 52(4):1289–1306, 2006.
- [5] Julien N Girard, Hugh Garsden, Jean Luc Starck, Stéphane Corbel, Arnaud Woiselle, Cyril Tasse, John P McKean, and Jérôme Bobin. Sparse representations and convex optimization as tools for lofar radio interferometric imaging. Journal of Instrumentation, 10(08):C08013, 2015.
- [6] Arwa Dabbech, Alexandru Onose, Abdullah Abdulaziz, Richard A Perley, Oleg M Smirnov, and Yves Wiaux. Cygnus a super-resolved via convex optimization from vla data. Monthly Notices of the Royal Astronomical Society, 476(3):2853–2866, 2018.
- [7] Jasleen Birdi, Audrey Repetti, and Yves Wiaux. Sparse interferometric stokes imaging under polarization constraint (polarized sara). arXiv preprint arXiv:1801.02417, 2018.
- [8] Tim J Cornwell, Kumar Golap, and Sanjay Bhatnagar. The noncoplanar baselines effect in radio interferometry: The w-projection algorithm. IEEE Journal of Selected Topics in Signal Processing, 2(5):647–657, 2008.
- [9] Luke Pratley, Melanie Johnston-Hollitt, and Jason D McEwen. A fast and exact  $w$ -stacking and  $w$ -projection hybrid algorithm for wide-field interferometric imaging. arXiv preprint arXiv:1807.09239, 2018.
- [10] Luke Pratley, Jason D McEwen, Mayeul d’Avezac, Rafael E Carrillo, Alexandru Onose, and Yves Wiaux. Robust sparse image reconstruction of radio interferometric observations with purify. Monthly Notices of the Royal Astronomical Society, 473(1):1038–1058, 2017.
- [11] Jason D McEwen and Yves Wiaux. Compressed sensing for wide-field radio interferometric imaging. Monthly Notices of the Royal Astronomical Society, 413(2):1318–1332, 2011.
- [12] TD Carozzi. Imaging on a sphere with interferometers: the spherical wave harmonic transform. Monthly Notices of the Royal Astronomical Society: Letters, 451(1):L6–L10, 2015.
- [13] JD McEwen and AMM Scaife. Simulating full-sky interferometric observations. Monthly Notices of the Royal Astronomical Society, 389(3):1163–1178, 2008.
- [14] Nathanaël Schaeffer. Efficient spherical harmonic transforms aimed at pseudospectral numerical simulations. Geochemistry, Geophysics, Geosystems, 14(3):751–758, 2013.
- [15] J Richard Shaw, Kris Sigurdson, Ue-Li Pen, Albert Stebbins, and Michael Sitwell. All-sky interferometry with spherical harmonic transit telescopes. The Astrophysical Journal, 781(2):57, 2014.
- [16] Jason D McEwen, Gilles Puy, Jean-Philippe Thiran, Pierre Vandergheynst, Dimitri Van De Ville, and Yves Wiaux. Sparse image reconstruction on the sphere: implications of a new sampling theorem. IEEE Transactions on image processing, 22(6):2275–2285, 2013.

- [17] Stephen J Hardy. Direct deconvolution of radio synthesis images using l1 minimisation. Astronomy & Astrophysics, 557:A134, 2013.
- [18] Jean-Luc Starck, Fionn Murtagh, and Mario Bertero. Starlet transform in astronomical data processing. Handbook of Mathematical Methods in Imaging, pages 2053–2098, 2015.
- [19] Zhi-Quan Luo and Paul Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. Journal of Optimization Theory and Applications, 72(1):7–35, 1992.
- [20] Simon Felix, Roman Bolzern, and Marina Battaglia. A compressed sensing-based image reconstruction algorithm for solar flare x-ray observations. The Astrophysical Journal, 849(1):10, 2017.

## List of Figures

1	The Major Cycle Framework . . . . .	2
2	Celestial sphere distortion on simulated data. Source: [8] . . . . .	5
3	Scaling and wavelet function of the starlet wavelet. Source: [18] . . . . .	11
4	Dirty map and $w_0$ starlet layer. The right image shows the probability distribution for point sources in the dirty image. . . . .	12
5	Image reconstruction on two point sources. . . . .	15
6	Dirty Image . . . . .	16
7	. . . . .	17

## List of Tables



## 7 Ehrlichkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende schriftliche Arbeit selbstständig und nur unter Zuhilfenahme der in den Verzeichnissen oder in den Anmerkungen genannten Quellen angefertigt habe. Ich versichere zudem, diese Arbeit nicht bereits anderweitig als Leistungsnachweis verwendet zu haben. Eine Überprüfung der Arbeit auf Plagiate unter Einsatz entsprechender Software darf vorgenommen werden.

Windisch, January 13, 2019

Jonas Schwammberger