

# P9 Distributed Image Reconstruction for the new Radio Interferometers

Jonas Schwammberger

September 18, 2019

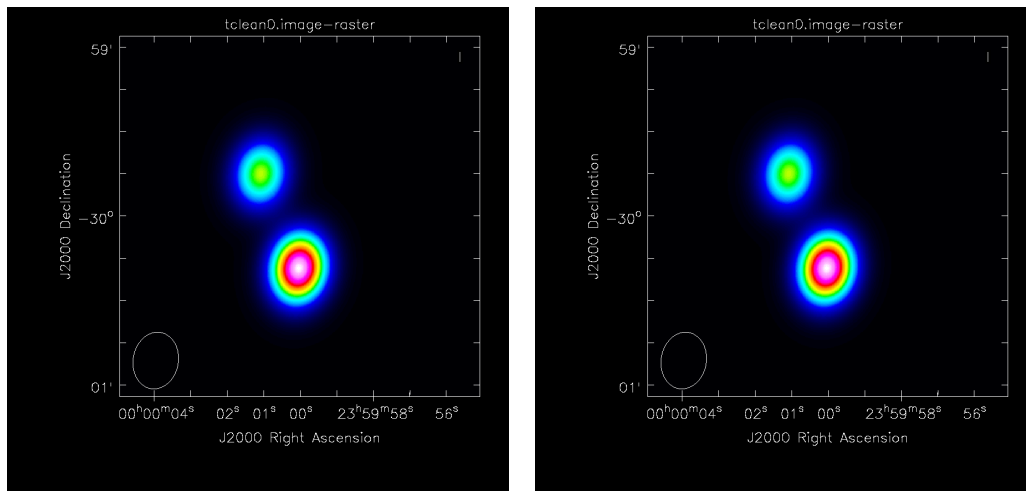
## **Abstract**

## Contents

<b>1</b>	<b>The image reconstruction problem of radio interferometers</b>	<b>1</b>
<b>2</b>	<b>Introduction to radio interferometric imaging</b>	<b>3</b>
2.1	From electromagnetic waves over visibilities to images . . . . .	4
2.1.1	The measurement equation . . . . .	5
2.2	The ill-posed image reconstruction problem . . . . .	6
2.2.1	Adding a regularization . . . . .	7
2.2.2	Compressive sampling of the sky . . . . .	8
2.2.3	Reconstruction guarantees in the real world . . . . .	8
2.3	Introduction into optimization/RI reconstruction algorithms . . . . .	9
2.3.1	The Major/Minor cycle . . . . .	9
2.3.2	Image reconstruction as deconvolution . . . . .	9
<b>3</b>	<b>State of the art image reconstruction</b>	<b>10</b>
3.1	Gridding algorithms . . . . .	10
3.1.1	$w$ -stacking . . . . .	10
3.1.2	Image Domain Gridder . . . . .	10
3.2	Deconvolution algorithms . . . . .	10
3.2.1	MS-MFS-CLEAN . . . . .	10
3.2.2	MORESANE . . . . .	11
3.3	Reconstruction algorithms which are not based on the deconvolution . . . . .	11
<b>4</b>	<b>Simple distributed image reconstruction</b>	<b>12</b>
4.1	Distributing the IDG algorithm . . . . .	13
4.2	Distributed deconvolution . . . . .	13
4.2.1	Coordinate Descent Method . . . . .	13
4.2.2	Efficient Coordinate Descent implementation . . . . .	15
4.2.3	ElasticNet Regularization . . . . .	16
4.3	Major Cycle convergence . . . . .	17
4.4	Test on MeerKAT data . . . . .	17
<b>5</b>	<b>Towards parallel coordinate descent</b>	<b>18</b>
5.1	Problems . . . . .	18
<b>6</b>	<b>Tests on the MeerKAT LMC observation</b>	<b>19</b>
6.1	Wall clock time . . . . .	19
6.2	Quality . . . . .	19
<b>7</b>	<b>Conclusion</b>	<b>20</b>
<b>8</b>	<b>attachment</b>	<b>25</b>
<b>9</b>	<b>Larger runtime costs for Compressed Sensing Reconstructions</b>	<b>26</b>
9.1	CLEAN: The Major Cycle Architecture . . . . .	27
9.2	Compressed Sensing Architecture . . . . .	27
9.3	Hypothesis for reducing costs of Compressed Sensing Algorithms . . . . .	28
9.4	State of the art: WSCLEAN Software Package . . . . .	28
9.4.1	W-Stacking Major Cycle . . . . .	28
9.4.2	Deconvolution Algorithms . . . . .	28
9.5	Distributing the Image Reconstruction . . . . .	28
9.5.1	Distributing the Non-uniform FFT . . . . .	28

9.5.2 Distributing the Deconvolution . . . . .	28
<b>10 Handling the Data Volume</b>	<b>28</b>
10.1 Fully distributed imaging algorithm . . . . .	28
<b>11 Image Reconstruction for Radio Interferometers</b>	<b>29</b>
11.1 Distributed Image Reconstruction . . . . .	30
11.2 First steps towards a distributed Algorithm . . . . .	30
<b>12 Ehrlichkeitserklärung</b>	<b>31</b>

# 1 The image reconstruction problem of radio interferometers



(a) Measurements of the interferometer in the Fourier space.

(b) Observed image of the sky, showing two stars.

Figure 1: The image reconstruction problem, the observed image has to be reconstructed from the Fourier measurements.

In Astronomy, one goal is to find ever smaller objects in the sky. For this purpose, we build instruments with higher angular resolution. The instruments angular resolution depends on two factors: On the diameter of the antenna-dish or mirror, and on the observed wavelength. With longer wavelengths we need bigger dishes/mirrors to achieve a similar angular resolution.

This is an issue for Radio Astronomy. The long radio wavelengths require huge dishes for a high angular resolution. Of course there is a practical limit on the antenna-dish diameter we can build. The famous Arecibo observatory is one of the largest single-dish radio telescopes with a diameter of 305 meters. Antennas with such a large diameter become difficult to steer accurately, let alone the construction costs. We have reached the practical limits of single-dish telescopes. If we require higher angular resolution, we need to look at another type of instrument: The radio interferometer. They use several smaller antennas together, acting like a single large dish. An interferometer can achieve angular resolutions which are comparable to dishes with a diameter of several kilometers.

But there are drawbacks: The interferometer does not measure the sky in pixels. It measures the amplitude and phase of Fourier components at a given  $u$  and  $v$  location<sup>1</sup>. The observed image has to be reconstructed from the measurements. The figure 1 shows an example of the image reconstruction problem. The figure 1a shows the measurements in the Fourier space, and the figure 1b shows the observed image of the sky, with two stars close to each other. The image reconstruction has to find the observed image 1b from the measurements 1a.

At first glance, we might believe that the image reconstruction is trivial: The interferometer measures Fourier components, and efficient algorithms for the inverse Fourier transforms are well-known. However, two properties of the measured Fourier components make the image reconstruction difficult: The measurements are both noisy and incomplete.

The atmosphere of the earth is one source that introduces noise. It adds noise to the amplitude and phase of each measured Fourier component. The atmosphere changes over time and can under the right circumstances introduce a high level of noise compared to the signal. The image reconstruction should be able to find the observed image from potentially very noisy Fourier measurements.

<sup>1</sup>  $u$  and  $v$  are the axis in Fourier space

The interferometer measures an incomplete set of Fourier components. Note that the figure 1a shows the Fourier space, which has missing components. The interferometer can only measure a limited set of Fourier components. The reconstruction algorithm has to find the observed image even though important Fourier components are missing from the measurements.

These two difficulties, the noise and the incomplete measurements, lead to the fact that there are many different candidate images that fit the measurements. From the measurements alone, we cannot decide which candidate is the truly observed image. However, we have additional knowledge that simplifies the problem: We know it is an image of the sky, which consists of stars, hydrogen clouds, etc. By including prior knowledge in the reconstruction, we can find the most likely image given the measurements.

The question remains is: How close is the most likely image to the observed one? Is exact reconstruction possible where the most likely and observed image are equal? Surprisingly the answer is yes. In theory, it is possible to even create a super-resolved reconstruction [14, 15] (where we retrieve an image below the accuracy limit of the instrument), and was shown in practice on low noise data[1, 2]. However, not all algorithms perform equally well when the noise level in the measurements is high. Also, computing resources required for each algorithm can vary significantly. In short, a reconstruction algorithm has three opposing goals:

1. Produce a reconstruction with the highest possible resolution from the measurements.
2. Robust against even heavy noise in the measurements.
3. Use as few computing resources as possible.

T

There is no reconstruction algorithm that has the optimal trade-off between all three goals. It is an active field of research.

To complicate things further, newly constructed radio interferometers pose the reconstruction problem on a big data scale. The recently finished MeerKAT radio interferometer produces roughly 80 million Fourier measurements each second. We need to distribute the image reconstruction. This is an active field of research. Research into how we can perform the image reconstruction in a

Part of data modelling. What is the prior knowledge

and Optimization algorithm, which finds the optimum trade-off

The ideal reconstruction algorithm not only achieves all three opposing goals, but is also distributable in a super-computer environment, preferably on the GPU. This is a tall order. So far, the go-to reconstruction algorithm CLEAN was performing its numerical optimization on a single machine. The new data volume raises the need for distributed and GPU-Accelerated optimization algorithms.

We explore coordinate Descent based methods

This Project focuses on the third point, distributed image reconstruction with real-world meerkat observations Received from SARAO for the purpose of algorithmic validation.

## 2 Introduction to radio interferometric imaging

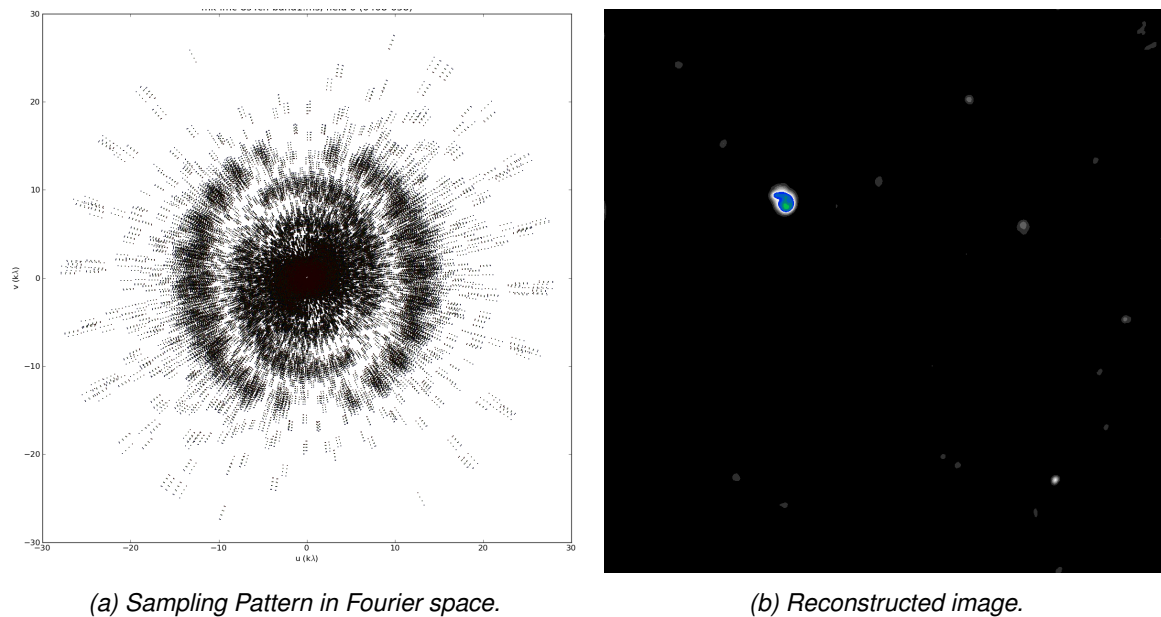


Figure 2: Example of an image reconstruction for visibility measurements of the MeerKAT radio interferometer

The figure 2 gives an example of the radio interferometric imaging. We wish to reconstruct an image, shown in 2b from the measurements of the interferometer 2a. The reconstructed image contains two object types: Point sources and extended emissions. Point sources, stars for example, are far-away emissions that are concentrated in a single pixel. Extended emissions like hydrogen clouds emit radio waves over a large area of the sky and span over several pixels in the reconstructed image.

The measurements of the radio interferometer are in the Fourier space. It does not measure how much emission was observed at a single pixel in the sky image, it measures the amplitude and phase of a Fourier component of the sky image. This means each dot in the figure 2a represents one Fourier measurement (amplitude and phase) of the observed image. The dots in the center, measurements with a low  $uv$ -value, measure the low-frequency components of the sky image. They contain the information about structures spanning over several pixels. Dots further away from the center measure the high-frequency components, they contain the information about the edges of the image and are responsible for a "sharp" image. In the radio astronomy literature, the measured Fourier components are called visibilities and for the sake of consistency, we will be calling the measured Fourier components visibilities in this work.

As we will see, we cannot retrieve the observed image by simply calculating the inverse Fourier transform. We require a reconstruction algorithm that finds the most likely observed image given the measurements. How close the most likely image is to the truly observed one depends on the reconstruction algorithm we choose. As it is often the case, how expensive an image reconstruction is in terms of runtime costs (Money spent on the necessary CPU's, time spent waiting for the results etc.) also depends on the reconstruction algorithm. We often face a trade-off between reconstruction quality and runtime costs. This work focuses on distributing the radio interferometric image reconstruction. We search for an effective way to use distributed computing hardware and test how scalable the reconstruction algorithm is with a real-world observation of the MeerKAT radio interferometer.

Introduction to how a radio interferometer works What the fundamental problems are and how we can solve them in theory and what the implications are in practice how we can This section gives a short introduction to interferometry and show the fundamental problems in the radio interferometric image reconstruction. In section 2.2 we give an introduction how we can solve the radio interferometric image reconstruction in theory,

and what the implications are in practice. Section 2.3 introduces how the common architecture for radio interferometric image reconstruction, and how the reconstruction algorithm is divided into sub-tasks in practice.

## 2.1 From electromagnetic waves over visibilities to images

We give a short introduction into how the electromagnetic wave gets measured by the interferometer, turned into visibilities and finally processed into an image. Figure 3 shows a radio source and its electro-magnetic (em) wave arriving at the antennas of the interferometer. It then shows the three processes involved to arrive at an image: Correlation, calibration and image reconstruction.

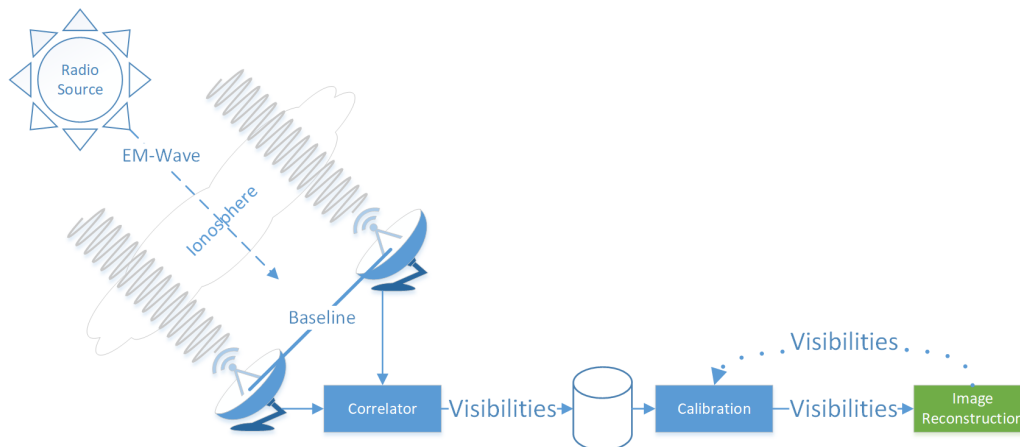


Figure 3: Radio interferometer system

First, we have a source in the sky that is emitting em-waves in the radio frequency. The waves travel to earth, through the earth's ionosphere and finally to our interferometer. Along its path, the e-m waves may get distorted from various sources. For example, it may receive a phase shift by the ionosphere.

Then, the em-wave arrives at our interferometer. We call each antenna pair a baseline. Each baseline will end up measuring a single visibility. The distance between the antennas and their orientation to the em-wave will determine where we sample the  $uv$ -plane. Short baselines measure the  $uv$ -plane close at the origin, while long baselines sample the  $uv$ -space further away from the origin. Remember that the samples away from the  $uv$ -origin contain the information about edges and other details of our image. With a longer baseline the interferometer measures more highly resolved details, regardless of the antenna dish-diameter<sup>2</sup>. The figure 3 shows the em-wave arriving at a single baseline of the interferometer. Each antenna picks up its version of the em-wave and transfers it to the correlator.

The correlator then takes the feed of each antenna and correlates the signals, which results in the amplitude and phase of the visibility component. Amplitude and phase for each visibility are measured for a short time range (i.e. fractions of a second up to several seconds). At this point, the visibilities are saved to disk for further processing. The radio interferometer produces a visibility measurement for each baseline, for each time range, for each frequency channel of the instrument. Because a single observation can take up to several hours, measured with several thousand frequency channels, radio interferometers produce an almost arbitrary large number of visibilities.

Calibration

Image Reconstruction

<sup>2</sup>Remember that this is the reason why we build radio interferometers. We do not need impossibly large dish diameters for a high angular resolution. We just need large distances between smaller antennas.



### 2.1.1 The measurement equation

As we discussed so far, the radio interferometer measures visibilities of the sky image, and we wish to find the observed image from the measurements. Put formally, we wish to invert the following system of linear equations (2.1), where  $V$  is the visibility vector<sup>3</sup>,  $F$  is the Fourier transform matrix and  $I$  is the pixel vector of the observed image.

$$V = FI \quad (2.1)$$

We wish to find the observed image  $I$ , while we only know the visibility vector  $V$  and the Fourier transform matrix  $F$ . This is what we call the measurement equation. In most context for this project, looking is an adequate view of the image reconstruction problem. We will show why we cannot find the observed image  $I$  by simply calculating the inverse Fourier transform. However, when we need to efficiently apply the Fourier transform, we need to know  $F$  in more detail. As we will see, radio interferometers have some difficulties hidden in the Fourier transform matrix, which are difficult to handle efficiently. First, let us abandon the vector notation of (2.1), and represent the measurement equation with integrals (2.2).

$$V(u, v) = \int \int I(l, m) e^{2\pi i[ul+vm]} dl dm \quad (2.2)$$

This is essentially the same problem. The main difference is that we do not represent the Fourier transform as a matrix  $F$ , but as integrals  $\int \int e^{2\pi i[ul+vm]}$ , where  $u, v$  are the coordinates in Fourier space and  $l, m$  are the angles away from the image center. A single pixel represents the intensity of the radio emission from the direction  $l, m$ . Note that the measurement equation (2.2) shows the fact that the visibilities are measured in a continuous Fourier space. If the Fourier space would also be discrete, we could replace the integrals with sums.

However, the measurement equation (2.2) is inaccurate in the sense that it ignores many effects that distort the signal. For example, it does not account for the distortion by the ionosphere, or the distortion introduced by real-world antennas. The measurement equation (2.2) shown here does not represent the real world. But depending on the instrument and the observation, these distortions may be negligible, and the measurement equation (2.2) is a good approximation.

When there is a distortion source that cannot be ignored, it has to be modelled in the measurement equation. As such there is no unified measurement equation for all radio interferometric observations, let alone radio interferometers. The equation shown in (2.2) can be seen as the basis that gets extended as necessary[3, 4, 5, 6].

For example, the measurement equation (2.2) is only accurate for small field of view observations, when  $l$  and  $m$  are both small angles. For wide field of view observations, we need to account for the fact that the visibilities have a third term  $w$ , and we arrive at the wide field of view measurement equation (2.3).

$$V(u, v, w) = \int \int \frac{I(l, m)}{c(l, m)} e^{2\pi i[ul+vm+w(c(l, m)-1)]} dl dm, \quad c(l, m) = \sqrt{1 - l^2 - m^2} \quad (2.3)$$

The third  $w$ -term has two effects on the measurement equation. It introduces a phase shift in the Fourier transform  $e^{2\pi i[...+w(c(l, m)-1)]}$ , and a normalization factor of the image  $\frac{I(l, m)}{c(l, m)}$ . Note that when the angles are small, i.e.  $l^2 + m^2 \ll 1$  then the wide field of view measurement equation (2.3) reduces to our original (2.2). This is another way of saying that for small field of views, the measurement equation (2.2) is a good approximation under the right conditions.

<sup>3</sup>We use the lower-case  $v$  to denote the axis in the Fourier space  $uvw$ , and the upper-case letter to denote the visibility vector.

In this project, we use the wide field of view measurement equation (2.3). But as we mentioned in the beginning of this section, for most contexts, it is not important whether we ignore the  $w$ -term of the visibilities or not. It is important when we design an efficient implementations for applying the wide field of view Fourier transform, because the  $w$ -term keeps us from using the Fast Fourier Transform (FFT). In every other case, we can ignore this technicality. Because even more complicated measurement equation still have a linear relationship between visibilities and image [3, 4, 5, 6]. We can view the whole reconstruction problem as a system of linear equations (2.1), where the matrix  $F$  takes care of how exactly the measurements and pixels relate in this case.

## 2.2 The ill-posed image reconstruction problem

So far, we discussed how the interferometer measures in Fourier space, and we wish to find the observed image that matches the measurements. In other words, we wish to find a solution to a system of linear equation (2.4), where  $V$  are the measurements,  $x$  is the observed image and  $F$  is the Fourier transform matrix. We also discussed that  $F$  can be complicated in practice, but is still essentially a linear operator. Meaning we know how the inverse Fourier transform matrix  $F^{-1}$ , and the question arises: Why can't we solve the equations (2.4) by calculating the inverse Fourier transform? Or, why does  $x = F^{-1}V$  not lead to the observed image?

$$\underset{x}{\text{solve}} \quad V - Fx = 0 \quad (2.4)$$

The answer is, the equations (2.4) do not have a unique solution, which makes the problem ill-posed. The problem is considered ill-posed when it has one of the following properties:

1. No solution exists.
2. There are solutions, but no unique solution exists.
3. The solution behaviour does not change continuously with the initial condition (For example: a small change in the measurements lead to a very different reconstructed image).

From linear algebra, we know that an under-determined system of linear equations, i.e. when (2.4) has more pixels than visibility measurements, then the problem is under-determined and there may be a potentially infinite number of solutions to the system. Under-determined systems arise in many similar fields, as for example in X-Ray imaging of the sun[7]. However, radio interferometers measure a large number of visibilities. We generally have more visibilities than pixels. This means the image reconstruction problem (2.4) for radio interferometers is actually over-determined.

From linear algebra, we know that an over-determined system either has one or zero solutions. At first glance it may be counter-intuitive why there are many possible solutions to (2.4) for radio interferometers. The reason why lies in two properties of the measurements: They are noisy and incomplete.

The radio interferometer measures noisy visibilities, meaning each amplitude and phase of a measurement is influenced by an unknown noise factor. Finding a reconstructed image is the same as finding the de-noised versions of the visibility measurements. This alone would make the problem ill-posed, but the visibilities actually have a second property that makes them ill-posed: incompleteness.

When we look back at figure 2a, it is clear to see that the interferometer does not sample the visibilities in a uniform way. There are regions with a high sample density. The density decreases when we move further away from the center. The higher frequency visibilities get fewer and fewer samples. This means we are missing data for crucial measurements for the reconstruction.

Also note that the question whether the measurements are incomplete essentially comes down to the image resolution of the reconstruction: Since we are missing high-frequency measurements and we can choose

the resolution of the image, we can also reduce the resolution of the reconstructed image until the missing frequencies become negligible. However, if we can solve the ill-posed inverse problem, we can reconstruct an image at a higher resolution from the same measurements. The question is how do we solve the ill-posed inverse problem?

### 2.2.1 Adding a regularization

For ill-posed inverse problems, there are two viewpoints for the same idea. From the viewpoint of optimization, we can solve the ill-posed image reconstruction problem (??) by adding a regularization. The regularization creates a system of linear equations with a unique solution. From the viewpoint of Bayesian Statistics, we include prior knowledge about the image, and therefore search the most likely image given the measurements. For this project, both terms describe the same idea and we use regularization and prior interchangeably. We know that the reconstructed image from radio interferometers contain a mixture of stars (point sources located in a single pixel) and extended emissions like hydrogen clouds. By adding this prior knowledge to the reconstruction problem, we can find the most likely image given the measurements. As we will see, under the right prior, we can create a reconstruction algorithm that is almost guaranteed to find the truly observed image in theory.

There are different ways to include regularization in the reconstruction problem. In this project, we use the following method: We cast the image reconstruction problem into an optimization objective consisting of a data fidelity term and a regularization term. A reconstruction algorithm therefore consists of an optimization objective, a prior function and an optimization algorithm.

$$\underset{x}{\text{minimize}} \quad \|V - Fx\|_2^2 + \lambda P(x) \quad (2.5)$$

The objective (2.5) has a data term  $\|V - Fx\|_2^2$ , which forces the most likely image to be as close to the measurements as possible, and the regularization term  $P(x)$ , which penalizes unlikely images according to our prior function. The parameter  $\lambda$  represents how much we penalize unlikely images and by extend, much noise we expect in the reconstruction. The parameter  $\lambda$  is either left to the user to define, can be estimated from the data [8].

The prior function  $P()$  represents our prior knowledge about the image. It assigns a high penalty for unlikely images. In in radio interferometric image reconstructions tend to use similar prior functions as for image de-noising applications. Such as: Total Variation ( $\|\nabla x\|_1$ ) [9], L2 ( $\|x\|_2$ ) [10] or the L1 norm in a wavelet space ( $\|\Psi x\|_1$ ) [11].

Finally, an optimization algorithm is necessary which can optimize the objective function (2.5) with the chosen prior function  $P()$ . Typical choices for optimization algorithms in image reconstructions are Interior Point Methods like Simplex, Matching Pursuit [12] and ADMM[13].

A reconstruction algorithm for radio astronomy consists of an optimization objective, a prior function and an optimization algorithm. It is a three dimensional design space. Not every prior is suitable for every optimization algorithm. The choice of optimization objective influences both what prior and what optimization algorithm we can use. Although there are a different choices for the optimization objective, we limit ourselves to the objective (2.5) and explore how we can distribute the reconstruction problem.

The last question that remains is, how close are the most likely image, under a given prior, to the truly observed one? Remember that the Nyquist-Shannon sampling theorem states that our uniform sampling frequency needs to be larger than twice the highest frequency in a band-limited signal<sup>4</sup>, and then the theorem guarantees exact reconstruction. For radio astronomy, we do not have uniformly sampled visibilities, and

<sup>4</sup>For example: if we want to record human voices with the highest frequency of 20 kHz, Nyquist-Shannon states our uniform sampling frequency has to be larger than 40 kHz to guarantee exact reconstruction

although we have a large number of samples, we are missing crucial parts of the Fourier space. Luckily, there is another sampling theorem that, under certain assumptions, guarantees exact reconstruction for the case of radio interferometers: The theory of compressed sensing.

### 2.2.2 Compressive sampling of the sky

For the sake of demonstration, let us assume the radio interferometer observes a patch of sky containing ten stars. It measures an incomplete set of random Fourier components of the ten stars, and we would like to reconstruct an image of size  $256^2$  pixels. The emissions from stars are concentrated into a single pixel. For compressive sampling, we need to know a space in which our reconstructed image is sparse, and we need to take measurements in a different, incoherent space.

Our reconstructed image contains only zero pixels except at the ten locations of the stars, meaning the image space for this patch of the sky is already sparse. In this case, we do not need any additional sparse space like wavelets. We can reconstruct directly in the sparse image space, and we have the first requirement met for compressive sampling.

The next requirement is that the measurement and reconstruction space (which is the image in this example), are as incoherent as possible.

Yes, incoherence.

$$\underset{x}{\text{minimize}} \quad \|V - Fx\|_2^2 + \lambda \|x\|_1 \quad (2.6)$$

Intuitively, the number of samples depend on the information content, not on the bandwidth. thi

At what point are we guaranteed? The matrix  $A$  needs to fulfill the Restricted Isometry Property (RIP) [14, 15]. Approximately orthonormal on sparse signals. (If we randomly choose ten columns of  $F$ , how much do they correlate. We want as little correlation as possible.) Calculate the RIP is NP-hard[16]. Approximations are also difficult to compute[17]. So we can only talk about how likely a given matrix fulfills the RIP. Matrix where each element is sampled from a random Gaussian distribution has the highest chance.

Random samples in the Fourier space also have a high chance to fulfill the RIP [18].

Not possible for Radio interferometers, because they sample in the Fourier space. Not random.

There are also extended emissions, clouds etc. Resulting in a lot of non-zero pixels. There may be better sparse spaces for radio astronomy.

But also RIP may be too strict. Exact reconstruction can also work under less strict conditions[19] active field of research.

### 2.2.3 Reconstruction guarantees in the real world

What does this all mean for image reconstruction? For us, we design reconstruction algorithms and cannot influence the matrix  $F$ . We do not know if it actually

The theory of compressed sensing gives us a framework. There is a data modelling task, and a task for finding efficient algorithms.

So there is a data modelling task in finding a good sparse prior. We also do not know the proper sparse space in which radio interferometric images. We know several spaces, Curvelets [20] Starlets [21], Daubechies wavelets [22]. As of the time of writing, it is currently unknown which leads to the best reconstruction. We can also learn dictionaries.

The task for efficient algorithms is finding the best way to optimize the objective with a given prior on real hardware. Fast convergence. Also difficult, because in practice the choice of prior can also influence convergence.

Compressed sensing based reconstruction algorithm. In radio astronomy.

## **2.3 Introduction into optimization/RI reconstruction algorithms**

### **2.3.1 The Major/Minor cycle**

### **2.3.2 Image reconstruction as deconvolution**

### 3 State of the art image reconstruction

This section introduces the state of the art image reconstruction algorithm in radio astronomy.

Two software packages, the Common Astronomy Software Applications (CASA)[23] and WSCLEAN[24] use the Major/Minor cycle architecture introduced in section 2.3.1. They have been the de-facto standard for decades and to our knowledge, are still the two most widely used implementations of image reconstruction algorithms. However, there has also been recent development to abandon the Major/Minor cycle architecture [1, 25]. They still use a gridding algorithm, but the reconstruction it does not contain Minor cycle deconvolution. At this point, it is not known whether the Major/Minor cycle architecture will be used in the future.

In this section, we introduce the

Major cycle architecture. Discuss the algorithms Split into two parts. We discuss the gridding first. It is responsible

#### 3.1 Gridding algorithms

Biggest part is  $w$ -term, how to handle it efficiently.

(show the problem of  $w$ -component)

Accuracy and speed.

Faceting.

$w$ -projection algorithm [26] Convolution in the Fourier domain.

The WSCLEAN [24] did a large part.

More recently, the Image domain gridding algorithm has been developed [27]. Which can put it on the gpu, along with handling more DDE's.

##### 3.1.1 $w$ -stacking

Idea of  $w$ -stacking, creating stacks of  $w$ -stack. So each visibility with similar  $w$ -component is in the same stack. Turns out we can then move

##### 3.1.2 Image Domain Gridder

Recently developed[27]. It works by partitioning the input visibilities into subgrids, and then calculates the interpolation and  $w$ -correction for each subgrid.

Called image domain, because a convolution in Fourier space is a multiplication in image space, interpolation kernels can be applied in image space. uses the idea of subgrids.

Algorithm

#### 3.2 Deconvolution algorithms

##### 3.2.1 MS-MFS-CLEAN

CLEAN basic [12].

Latest variants for multiscale and multi frequency CLEAN (MS-MFS-CLEAN)[28].

Current state-of-the-art

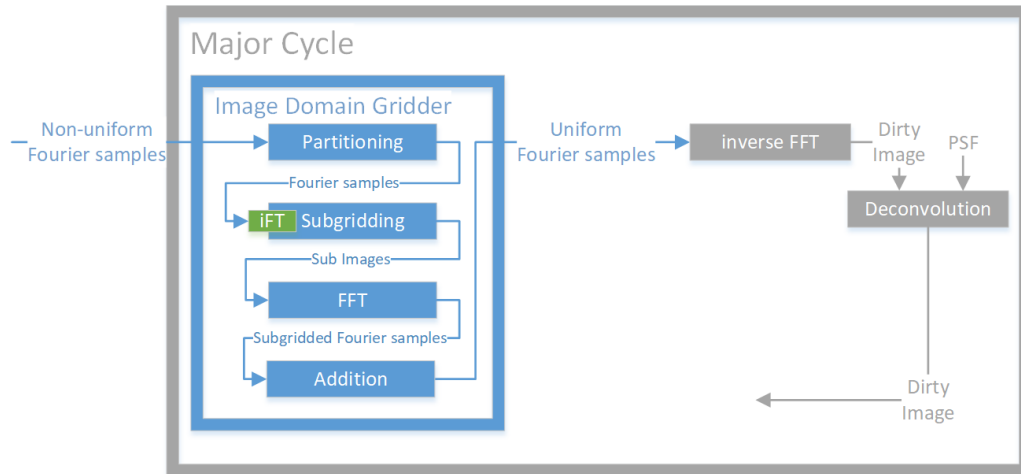


Figure 4: Image Domain Gridder in the Major Cycle Architecture

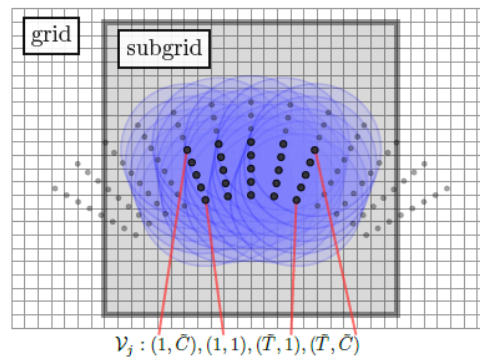


Figure 5: Subgrid

### 3.2.2 MORESANE

MORESANE [2]

### 3.3 Reconstruction algorithms which are not based on the deconvolution

SARA[1].

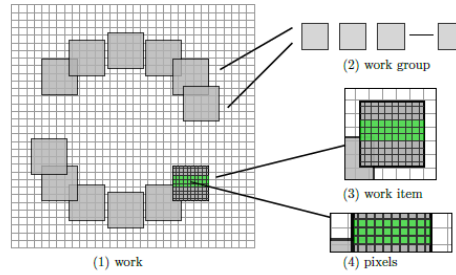


Figure 6: parallel

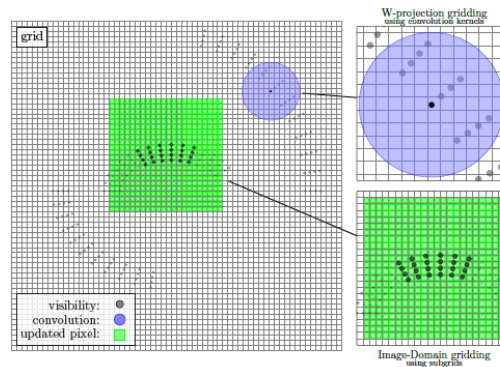


Figure 7: Image Domain Gridding in the Major Cycle Architecture

## 4 Simple distributed image reconstruction

How do we distribute the major cycle. We need to distribute every step, Gridding, FFT and Deconvolution.

Gridding, Large number of input data. This needs to be distributed We use the Image domain gridding introduces in section 3.1.2 and use it as the basis for the distributed gridding.

The FFT is generally not worth distributing, if we can keep all the data in memory. When the gridding is done, in our setup, the grid is small enough to keep in memory. (cite distributed fftw)

Deconvolution is also worth distributing. CLEAN depending on the observation is the second most time consuming step. But gridding tends to be easier to distribute, so in some observations it is the most time consuming step. Split the image into patches and deconvolve each patch. Sadly not possible, we need communication. how we communicate is important.

We use a distributed Gridding and a distributed deconvolution. Which leads us to the following architecture.

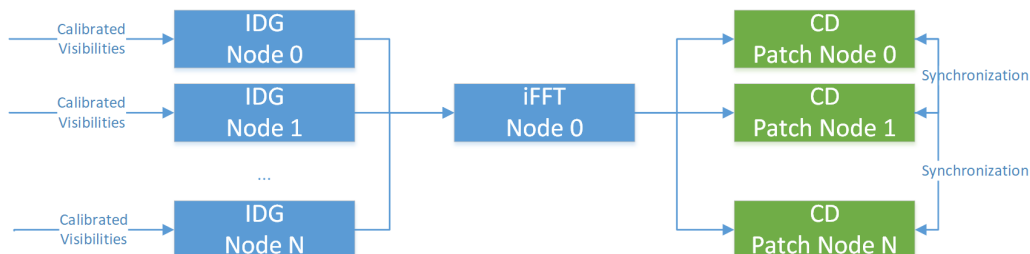


Figure 8: Distributed architecture for half a major cycle

Where each node is one computer, i.e. has its own, possibly multiple cpus and its shared memory. Split the



input visibilities onto nodes. Do the gridding locally on each node Communicate the grid inverse FFT on one node. Communicate the patches of the image. Deconvolve each patch and communicate

How we distribute the IDG algorithm, and create a distributed deconvolution algorithm. Main contribution is the distributed deconvolution algorithm

## 4.1 Distributing the IDG algorithm

## 4.2 Distributed deconvolution

Main contribution We start

What we need: An optimization objective, an optimization algorithm and a regularization.

We use (4.1) as our optimization objective with data fidelity term and regularization term. We use Coordinate Descent Methods as our optimization algorithm, which we introduce in section 4.2.1, and the ElasticNet regularization, which we discuss in section 4.2.3

$$\underset{x}{\text{minimize}} \quad \|I_{\text{dirty}} - x * PSF\|_2^2 + \lambda \text{ElasticNet}(x) \quad (4.1)$$

### 4.2.1 Coordinate Descent Method

Coordinate Descent has gained more interest, because it may exploit sparse structures in the data.

Different versions. Optimization for large data volume[29]

4 multiplications and additions only[30]

error tolerance  $O(n \sim LR / \epsilon)$ , [31]

can be more efficient on big data problems than gradient methods [32, 33]

Most problems become simple when we look at them in the one dimensional case. Finding the optimum to our objective (4.1) is hard. But when we try to find the optimum of a single pixel, independent of all other pixels, the problem becomes simple. As we will see, we can find a closed form solution to find the optimum of a single pixel. Coordinate Descent then optimizes a single pixel at a time, and we iterate over the whole image with some strategy until we converge. We may need to optimize the same pixel several times.

Monotone decreasing. Robust Many different variants.

Outperforms other algorithms like Gradient based methods, FISTA and ADMM if a single iteration (minimizing a single pixel in our case) is "cheap" to compute. How fast it is in practice depends on the regularization  $P(x)$  we use.

The basic algorithm is as follows in pseudocode: A major cycle has calculated the dirty image and the  $PSF$ . We do not care yet about computation speed.

Because it is a separable objective function, we can handle ElasticNET() separately

```

1 dirty = IFFT(Gridding(visibilities))
2 residuals = dirty
3
4 x = new Array
5 objectiveValue = SUM(residuals * residuals) + P(x)
6 oldObjectiveValue = objectiveValue
7
8 do
9 {
```

```

10  oldObjectiveValue = objectiveValue
11
12  //the core of the algorithm
13  pixelLocation = IterationStrategy(residuals)
14  optimalValue = Minimize(residuals, psf, pixelLocation)
15  optimalValue = ApplyElasticNet(optimalValue)
16
17  //housekeeping
18  x[pixelLocation] += optimalValue
19  residuals = (dirty - Convolve(x, psf))
20  objectiveValue = SUM(residuals * residuals) + lambda * ElasticNet(x)
21 } while (oldObjectiveValue - objectiveValue) < epsilon

```

The core of the algorithm consists of three functions: *IterationStrategy* which selects a pixel to optimize according to some strategy, *Minimize()* which then finds the minimum value of the pixel, and *ApplyElasticNet()*, which finds the regularized value of the pixel according to the Elastic Net regularization. First we discuss the iteration strategy and derive the *Minimize()* function is in our deconvolution task. How the function *ApplyElasticNet()* looks is discussed in section section 4.2.3.

From the viewpoint of numerical optimization, Minimize is always an "implementation" detail of the algorithm.

The iteration strategy

- Random
- Cyclic
- Greedy

Random the next point is the iteration strategy. This is where Coordinate Descent methods are robust. They are proven to converge by simply choosing randomly. Or a greedy strategy where the pixel which reduces the objective (4.1) by the largest amount. It was proven to converge for a variety of strategies. For this project, we are interested in which takes fewer computing resources to converge. Start with a greedy scheme, well known convergence guarantees at least linearly (with respect to iteration count?) [34] Cyclic scheme.

The task of the *Minimize()* function is to reduce the data term of the objective (4.1), i.e.  $\|I_{dirty} - x * PSF\|_2^2$ . There are many different ways to implement *Minimize()*. For example, we could use a line search algorithm, which tries different values for the pixel and saves the value with the lowest data objective. Luckily for our case we have a closed form solution. Meaning we have a formula that results in the optimal pixel value. It's derived from the fact that the data term reduces to a parabola when we only have one unknown pixel. The equation (4.2) derives the parabola from the data term, and the equation (4.3) shows the closed form solution, where  $\langle x, y \rangle$  is the inner product<sup>5</sup>, the element-wise multiplication followed by a sum of the two vectors.

$$\begin{aligned}
 objective(pixel) &= \|I_{res} - PSF * pixel\|_2^2 \\
 objective(pixel) &= (I_{res} - PSF * pixel)^2 \\
 objective(pixel) &= \langle I_{res}, I_{res} \rangle - 2 * \langle I_{res}, PSF \rangle * pixel + \langle PSF, PSF \rangle * pixel^2 \\
 objective(pixel) &= \langle PSF, PSF \rangle * pixel^2 - 2 * \langle I_{res}, PSF \rangle * pixel + \langle I_{res}, I_{res} \rangle
 \end{aligned} \tag{4.2}$$

<sup>5</sup>As the reader is probably aware, there is no one inner product. Rather, there are several ways to define a valid inner product. Here we use the element-wise multiplication followed by a sum.

$$\begin{aligned}
 f(x) &= a * x^2 & objective(pixel) &= \langle PSF, PSF \rangle * pixel^2 \\
 &+ b * x & &- 2 * \langle I_{res}, PSF \rangle * pixel \\
 &+ c & &+ \langle I_{res}, I_{res} \rangle
 \end{aligned} \tag{4.3}$$

$$x_{min} = \frac{-b}{2a} \qquad pixel_{min} = \frac{\langle I_{res}, PSF \rangle}{\langle PSF, PSF \rangle}$$

Note that the  $PSF$  in (4.3) is centred at the pixel we wish to minimize. Also, if the pixel is located at the edges of the image, it may be partially cut off. This fact becomes relevant as soon as we want to implement  $Minimize()$  efficiently and pre-calculate the inner-products  $\langle I_{dirty}, PSF \rangle$  and  $\langle PSF, PSF \rangle$ . From here on forward, we use the terminology  $b$ -map to denote the pre-calculated inner products  $\langle I_{dirty}, PSF \rangle$  for each pixel, and  $a$ -map for all pre-calculated  $\langle PSF, PSF \rangle$  products (from the fact that  $x_{min} = \frac{-b}{2a}$ ).

## 4.2.2 Efficient Coordinate Descent implementation

Coordinate Descent is only useful when one iteration is cheap to compute.

Pre-calculation of the two inner products Efficiently

The question of efficient implementation. In the previous section, we have seen how coordinate descent works in general. But now we have to look at the details and see how we can implement it efficiently.

Question of convolution scheme. Circular convolution is physically not possible. The  $PSF$  of the lower left corner does not wrap around the image and influence all other corners of the image. But as we will see, for an efficient coordinate descent we need to calculate a convolution to cache. The circular convolution is the output of the FFT. This is why reconstruction algorithms sometimes choose to use circular convolution [10].

The other convolution scheme, namely zero padding. We use this scheme. It results

Iteration scheme. For a greedy iteration scheme, we would need to first:

1. find the optimum pixel value for each pixel independently (i.e.  $\frac{-b}{2a}$ )
2. evaluate which optimum pixel value results in the best reduction of the objective function (4.1)

For the first step, we can cache intermediate results and drastically reduce the computation. The second step can be approximated in a way that we do not need to evaluate the objective function.

### Pre-Calculating $b$ - and $a$ -maps

Caching of intermediate results. We have seen in section 4.2.1 how we can find the optimum for a single pixel. We need to calculate  $x_{opt} = \frac{-b}{2a}$  where  $b = -2SUM(pixel * PSF * I_{dirty})$  and  $a = SUM(PSF * PSF)$ . First, we note that  $a$  depends only on the  $PSF$  which is constant, which means  $a$  is also constant. However, this is not true depending on the convolution we use.  $PSF$  at the corner gets masked off, and the  $a$  value changes. We can calculate a lookup map in linear time.

How to cache  $b$ . Use correlation of the dirty image with the  $PSF$ . Then we have a lookup map. Can be done efficiently in the Fourier domain. But the question then becomes in how to update  $b$ Map efficiently.

Unsolved problem of varying  $PSF^2$

### Approximating the objective function

$a$  is constant in  $\frac{-b}{2a}$

Putting it all together. Creating lookup maps for  $a$  and  $b$ , and do not evaluate the objective function.

```

1 dirty = IFFT(Gridding(visibilities))
2
3 //pre-calculate bMap
4 dirtyPadded = ZeroPadding(dirty , psfSize)
5 fourierDirtyPadded = FFT(dirtyPadded)
6 fourierPsf = FFT(invert(psf)) //we require the correlation, that is why we invert
  the psf
7 bMap = IFFT(fourierDirtyPadded * fourierPsf)
8
9 //pre-calculate aMap. aMap stays constant over all CD iterations
10 aMap =
11
12 x = new Array
13 do
14 {
15   oldObjectiveValue = objectiveValue
16
17   //the core of the algorithm
18   pixelLocation = IterationStrategy(residuals)
19   optimalValue = Minimize(residuals , pixelLocation)
20   optimalValue = ApplyElasticNet(optimalValue)
21
22   //housekeeping
23   x[pixelLocation] = optimalValue
24   residuals = (dirty - Convolve(x, psf))
25   objectiveValue = SUM(residuals * residuals) + ElasticNet(x)
26 } while (oldObjectiveValue - objectiveValue) < epsilon

```

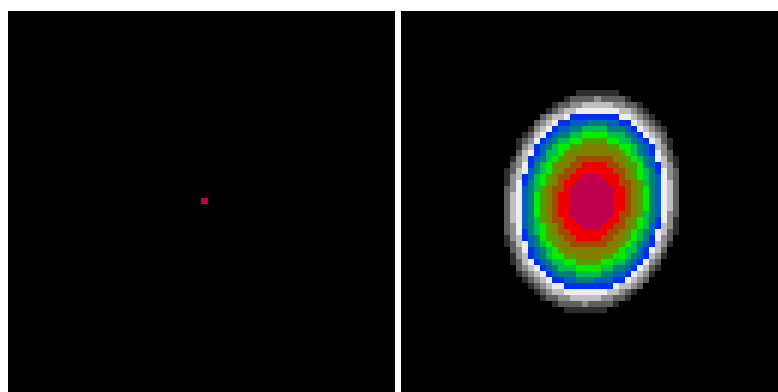
---

### 4.2.3 ElasticNet Regularization

L2 norm was used in other work. [10]

"Sparsifying L2 norm" Remember the theory of compressed sensing.

Formula



(a) Effect of the pure L1 norm ( $\lambda = 1.0$ ) on a single point source. (b) Effect of the pure L2 norm ( $\lambda = 1.0$ ) on a single point source.

Figure 9: Effect of the L1 and L2 Norm separately.

Effect

Implementation, proximal operator. Shrinkage of a division.

May even speed up convergence for correlated pixel values compared to L1 or L2[35]. But was not investigated in this project

### **4.3 Major Cycle convergence**

Putting it all together

We have the Minor Cycle, which is easy to converge.

Coordinate Descent Path optimization [35] Danger that CD takes too many pixel into a Major Cycle. Lower bound per iteration, PSF sidelobe can still be too low, danger when many psf sidelobes overlap

### **4.4 Test on MeerKAT data**

## 5 Towards parallel coordinate descent

Bells and whistles

- Random
- Block
- Parallel
- Accelerated

PCDM algorithm, and later APPROX.

### 5.1 Problems

Problem with pure random: each update is fast, but problem with useful updates.  $-\zeta$  we need blocks where we have a higher chance to update a pixel.

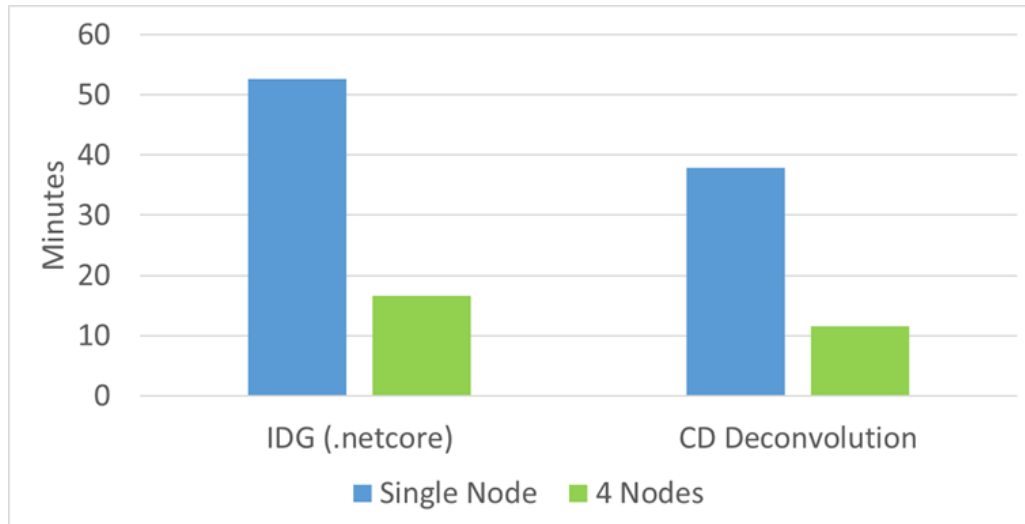
Expansion to BlockCD. Still closed form solution. But numerical problems with closed form. Solution: [36]. We use the gradient step divided by the lipschitz constant.

Towards parallel updates

## 6 Tests on the MeerKAT LMC observation

For algorithmic testing

### 6.1 Wall clock time



*Figure 10: Wall-clock time of the distributed reconstruction*

### 6.2 Quality

## 7 Conclusion



## References

- [1] 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.
- [2] Arwa Dabbech, Chiara Ferrari, David Mary, Eric Slezak, Oleg Smirnov, and Jonathan S Kenyon. More-sane: Model reconstruction by synthesis-analysis estimators-a sparse deconvolution algorithm for radio interferometric imaging. Astronomy & Astrophysics, 576:A7, 2015.
- [3] Oleg M Smirnov. Revisiting the radio interferometer measurement equation-i. a full-sky jones formalism. Astronomy & Astrophysics, 527:A106, 2011.
- [4] Oleg M Smirnov. Revisiting the radio interferometer measurement equation-ii. calibration and direction-dependent effects. Astronomy & Astrophysics, 527:A107, 2011.
- [5] Oleg M Smirnov. Revisiting the radio interferometer measurement equation-iii. addressing direction-dependent effects in 21 cm wsrt observations of 3c 147. Astronomy & Astrophysics, 527:A108, 2011.
- [6] Oleg M Smirnov. Revisiting the radio interferometer measurement equation-iv. a generalized tensor formalism. Astronomy & Astrophysics, 531:A159, 2011.
- [7] 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.
- [8] Keith Miller. Least squares methods for ill-posed problems with a prescribed bound. SIAM Journal on Mathematical Analysis, 1(1):52–74, 1970.
- [9] Yves Wiaux, Laurent Jacques, Gilles Puy, Anna MM Scaife, and Pierre Vanderghelynst. Compressed sensing imaging techniques for radio interferometry. Monthly Notices of the Royal Astronomical Society, 395(3):1733–1742, 2009.
- [10] André Ferrari, David Mary, Rémi Flamary, and Cédric Richard. Distributed image reconstruction for very large arrays in radio astronomy. In 2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM), pages 389–392. IEEE, 2014.
- [11] 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.
- [12] JA Högbom. Aperture synthesis with a non-regular distribution of interferometer baselines. Astronomy and Astrophysics Supplement Series, 15:417, 1974.
- [13] Rafael E Carrillo, Jason D McEwen, and Yves Wiaux. Purify: a new approach to radio-interferometric imaging. Monthly Notices of the Royal Astronomical Society, 439(4):3591–3604, 2014.
- [14] 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.
- [15] David L Donoho. Compressed sensing. IEEE Transactions on information theory, 52(4):1289–1306, 2006.
- [16] Andreas M Tillmann and Marc E Pfetsch. The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing. IEEE Transactions on Information Theory, 60(2):1248–1259, 2013.

- [17] Abhiram Natarajan and Yi Wu. Computational complexity of certifying restricted isometry property. arXiv preprint arXiv:1406.5791, 2014.
- [18] Ishay Haviv and Oded Regev. The restricted isometry property of subsampled fourier matrices. In Geometric Aspects of Functional Analysis, pages 163–179. Springer, 2017.
- [19] Emmanuel J Candes and Yaniv Plan. A probabilistic and riplless theory of compressed sensing. IEEE transactions on information theory, 57(11):7235–7254, 2011.
- [20] Jean-Luc Starck, David L Donoho, and Emmanuel J Candès. Astronomical image representation by the curvelet transform. Astronomy & Astrophysics, 398(2):785–800, 2003.
- [21] Jean-Luc Starck, Fionn Murtagh, and Mario Bertero. Starlet transform in astronomical data processing. Handbook of Mathematical Methods in Imaging, pages 2053–2098, 2015.
- [22] Rafael E Carrillo, Jason D McEwen, and Yves Wiaux. Sparsity averaging reweighted analysis (sara): a novel algorithm for radio-interferometric imaging. Monthly Notices of the Royal Astronomical Society, 426(2):1223–1234, 2012.
- [23] National Radio Astronomy Observations. Casa, 2019.
- [24] AR Offringa, Benjamin McKinley, Natasha Hurley-Walker, FH Briggs, RB Wayth, DL Kaplan, ME Bell, Lu Feng, AR Neben, JD Hughes, et al. Wsclean: an implementation of a fast, generic wide-field imager for radio astronomy. Monthly Notices of the Royal Astronomical Society, 444(1):606–619, 2014.
- [25] 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.
- [26] 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.
- [27] Bram Veenboer, Matthias Petschow, and John W Romein. Image-domain gridding on graphics processors. In 2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pages 545–554. IEEE, 2017.
- [28] Urvashi Rau and Tim J Cornwell. A multi-scale multi-frequency deconvolution algorithm for synthesis imaging in radio interferometry. Astronomy & Astrophysics, 532:A71, 2011.
- [29] Peter Richtárik and Martin Takáč. Distributed coordinate descent method for learning with big data. The Journal of Machine Learning Research, 17(1):2657–2681, 2016.
- [30] Peter Richtárik and Martin Takáč. Efficient serial and parallel coordinate descent methods for huge-scale truss topology design. In Operations Research Proceedings 2011, pages 27–32. Springer, 2012.
- [31] Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. Mathematical Programming, 156(1-2):433–484, 2016.
- [32] Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM Journal on Optimization, 22(2):341–362, 2012.
- [33] Yu Nesterov. Gradient methods for minimizing composite functions. Mathematical Programming, 140(1):125–161, 2013.
- [34] 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.

- [35] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. Journal of statistical software, 33(1):1, 2010.
- [36] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Mathematical Programming, 144(1-2):1–38, 2014.
- [37] 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.

## List of Figures

1	The image reconstruction problem, the observed image has to be reconstructed from the Fourier measurements. . . . .	1
2	Example of an image reconstruction for visibility measurements of the MeerKAT radio interferometer . . . . .	3
3	Radio interferometer system . . . . .	4
4	Image Domain Gridder in the Major Cycle Architecture . . . . .	11
5	Subgrid . . . . .	11
6	parallel . . . . .	12
7	Image Domain Gridder in the Major Cycle Architecture . . . . .	12
8	Distributed architecture for half a major cycle . . . . .	12
9	Effect of the L1 and L2 Norm separately. . . . .	16
10	Wall-clock time of the distributed reconstruction . . . . .	19
11	The Major Cycle Architecture . . . . .	27
12	State-of-the-art Compressed Sensing Reconstruction Architecture . . . . .	27
13	The Major Cycle Architecture of image reconstruction algorithms . . . . .	29

## List of Tables

## **8 attachment**

## 9 Larger runtime costs for Compressed Sensing Reconstructions

The MeerKAT instrument produces a new magnitude of data volume. An image with several million pixels gets reconstructed from billions of Visibility measurements. Although MeerKAT measures a large set of Visibilities, the measurements are still incomplete. We do not have all the information available to reconstruct an image. Essentially, this introduces "fake" structures in the image, which a reconstruction algorithm has to remove. Additionally, the measurements are noisy.

We require an image reconstruction algorithm which removes the "fake" structures from the image, and removes the noise from the measurements. The large data volume of MeerKAT requires the algorithm to be both scalable and distributable. Over the years, several reconstruction algorithms were developed, which can be separated into two classes: Algorithms based on CLEAN, which are cheaper to compute and algorithms based on Compressed Sensing, which create higher quality reconstructions.

CLEAN based algorithms represent the reconstruction problem as a deconvolution. First, they calculate the "dirty" image, which is corrupted by noise and fake image structures. The incomplete measurements essentially convolve the image with a Point Spread Function (*PSF*). CLEAN estimates the *PSF* and searches for a deconvolved version of the dirty image. In each CLEAN iteration, it searches for the highest pixel in the dirty image, subtracts a fraction *PSF* at the location. It adds the fraction to the same pixel location of a the "cleaned" image. After several iterations, the cleaned image contains the deconvolved version of the dirty image. CLEAN accounts for noise by stopping early. It stops when the highest pixel value is smaller than a certain threshold. This results in a light-weight and robust reconstruction algorithm. CLEAN is comparatively cheap to compute, but does not produce the best reconstructions and is difficult to distribute on a large scale.

Compressed Sensing based algorithms represent the reconstruction as an optimization problem. They search for the optimal image which is as close to the Visibility measurements as possible, but also has the smallest regularization penalty. The regularization encodes our prior knowledge about the image. Image structures which were likely measured by the instrument result in a low regularization penalty. Image structures which were likely introduced by noise or the measurement instrument itself result in high penalty. Compressed Sensing based algorithms explicitly handle noise and create higher quality reconstructions than CLEAN. State-of-the-art Compressed Sensing algorithms show potential for distributed computing. However, they currently do not scale on MeerKATs data volume. They require too many computing resources compared to CLEAN based algorithms.

This project searches for a way to reduce the runtime costs of Compressed Sensing based algorithms. One reason for the higher costs is due to the non-uniform FFT Cycle. State-of-the-art CLEAN and Compressed Sensing based algorithms both use the non-uniform FFT approximation in a cycle during reconstruction. The interferometer measures the Visibilities in a continuous space in a non-uniform pattern. The image is divided in a regularly spaced, discrete pixels. The non-uniform FFT creates an approximate, uniformly sampled image from the non-uniform measurements. Both, CLEAN and Compressed Sensing based algorithms use the non-uniform FFT to cycle between non-uniform Visibilities and uniform image. However, a Compressed Sensing algorithm requires more non-uniform FFT cycles for reconstruction.

CLEAN and Compressed Sensing based algorithms use the non-uniform FFT in a similar manner. However, there are slight differences in the architecture. This project hypothesises that The previous project searched for an alternative to the non-uniform FFT cycle. Although there are alternatives, there is currently no replacement which leads to lower runtime costs for Compressed Sensing. Current research is focused on reducing the number of non-uniform FFT cycles for Compressed Sensing algorithms.

CLEAN based algorithms use the Major Cycle Architecture for reconstruction. Compressed Sensing based algorithms use a similar architecture, but with slight modifications. Our hypothesis is that we may reduce the number of non-uniform FFT cycles for Compressed Sensing by using CLEAN's Major Cycle Architecture.

## 9.1 CLEAN: The Major Cycle Architecture

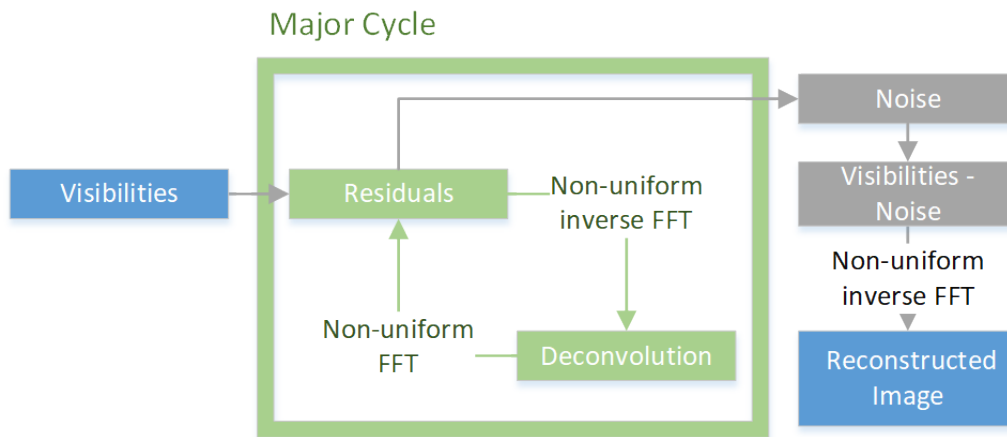


Figure 11: The Major Cycle Architecture

Figure 11 depicts the Major Cycle Architecture used by CLEAN algorithms. First, the Visibilities get transformed into an image with the non-uniform FFT. The resulting dirty image contains the corruptions of the measurement instrument and noise. A deconvolution algorithm, typically CLEAN, removes the corruption of the instrument with a deconvolution. When the deconvolution stops, it should have removed most of the observed structures from the dirty image. The rest, mostly noisy part of the dirty image gets transformed back into residual Visibilities and the cycle starts over.

In the Major Cycle Architecture, we need several deconvolution attempts before it has distinguished the noise from the measurements. Both the non-uniform FFT and the deconvolution are approximations. By using the non-uniform FFT in a cycle, it can reconstruct an image at a higher quality. For MeerKAT reconstruction with CLEAN, we need approximately 4-6 non-uniform FFT cycles for a reconstruction.

## 9.2 Compressed Sensing Architecture

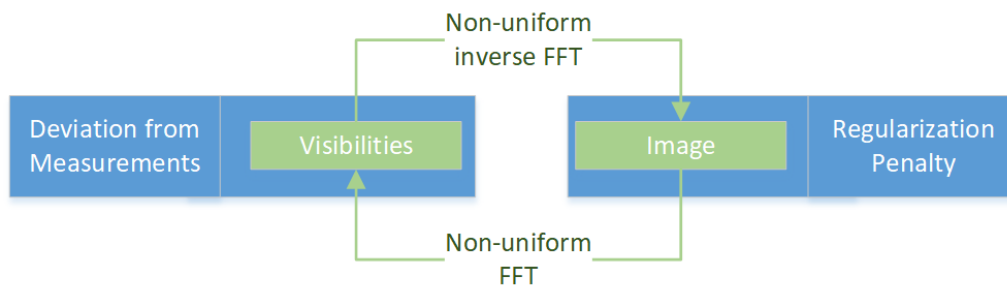


Figure 12: State-of-the-art Compressed Sensing Reconstruction Architecture

Figure 12 depicts the architecture used by Compressed Sensing reconstructions. The Visibilities get transformed into an image with the non-uniform FFT approximation. The algorithm then modifies the image so it reduces the regularization penalty. The modified image gets transformed back to Visibilities and the algorithm then minimizes the difference between measured and reconstructed Visibilities. This is repeated until the algorithm converges to an optimum.

In this architecture, state-of-the-art Compressed Sensing algorithms need approximately 10 or more non-uniform FFT cycles to converge. It is one source for the higher runtime costs. For MeerKAT reconstructions

the non-uniform FFT tends to dominate the runtime costs. A CLEAN reconstruction with the Major Cycle Architecture already spends a large part of its time in the non-uniform FFT. Compressed Sensing algorithms need even more non-uniform FFT cycle on top of the "Image Regularization" step being generally more expensive than CLEAN deconvolution. There is one upside in this architecture: State-of-the-art algorithms managed to distribute the "Image Regularization" operation.

### 9.3 Hypothesis for reducing costs of Compressed Sensing Algorithms

Compressed Sensing Algorithms are not bound to the Architecture presented in section 9.2. For example, we can design a Compressed Sensing based deconvolution algorithm and use the Major Cycle Architecture instead.

Our hypothesis is: We can create a Compressed Sensing based deconvolution algorithm which is both distributable and creates higher quality reconstructions than CLEAN. Because it also uses the Major Cycle architecture, we reckon that the Compressed Sensing deconvolution requires a comparable number of non-uniform FFT cycles to CLEAN. This would result in a Compressed Sensing based reconstruction algorithm with similar runtime costs to CLEAN, but higher reconstruction quality and higher potential for distributed computing.

### 9.4 State of the art: WSCLEAN Software Package

#### 9.4.1 W-Stacking Major Cycle

#### 9.4.2 Deconvolution Algorithms

CLEAN MORESANE

### 9.5 Distributing the Image Reconstruction

#### 9.5.1 Distributing the Non-uniform FFT

#### 9.5.2 Distributing the Deconvolution

## 10 Handling the Data Volume

The new data volume is a challenge to process for both algorithms and computing infrastructure. Push for parallel and distributed algorithms. For Radio Interferometer imaging, we require specialized algorithms. The two distinct operations, non-uniform FFT and Deconvolution, were difficult algorithms for parallel or distributed computing.

The non-uniform FFT was historically what dominated the runtime []. Performing an efficient non-uniform FFT for Radio Interferometers is an active field of research[24, 37], continually reducing the runtime costs of the operation. Recently, Veeneboer et al[27] developed a non-uniform FFT which can be fully executed on the GPU. It speeds up the most expensive operation.

In Radio Astronomy, CLEAN is the go-to deconvolution algorithm. It is light-weight and compared to the non-uniform FFT, a cheap algorithm. It is also highly iterative, which makes it difficult for effective parallel or distributed implementations. However, compressed sensing based deconvolution algorithms can be developed with distribution in mind.

### 10.1 Fully distributed imaging algorithm

Current imaging algorithms push towards parallel computing with GPU acceleration. But with Veeneboer et al's non-uniform FFT and a compressed sensing based deconvolution, we can go a step further and create a distributed imaging algorithm.



## 11 Image Reconstruction for Radio Interferometers

In Astronomy, instruments with higher angular resolution allows us to measure ever smaller structures in the sky. For Radio frequencies, the angular resolution is bound to the antenna dish diameter, which puts practical and financial limitations on the highest possible angular resolution. Radio Interferometers get around this limitation by using several smaller antennas instead. Together, they act as a single large antenna with higher angular resolution at lower financial costs compared to single dish instruments.

Each antenna pair of an Interferometer measures a single Fourier component of the observed image. We can retrieve the image by calculating the Fourier Transform of the measurements. However, since the Interferometer only measures an incomplete set of Fourier components, the resulting image is "dirty", convolved with a Point Spread Function (*PSF*). Calculating the Fourier Transform is not enough. To reconstruct the from an Interferometer image, an algorithm has to find the observed image with only the dirty image and the *PSF* as input. It has to perform a deconvolution. The difficulty lies in the fact that there are potentially many valid deconvolutions for a single measurement, and the algorithm has to decide for the most likely one. How similar the truly observed image and the reconstructed images are depends largely on the deconvolution algorithm.

State-of-the-art image reconstructions use the Major Cycle architecture (shown in Figure 13), which contains three operations: Gridding, FFT and Deconvolution.

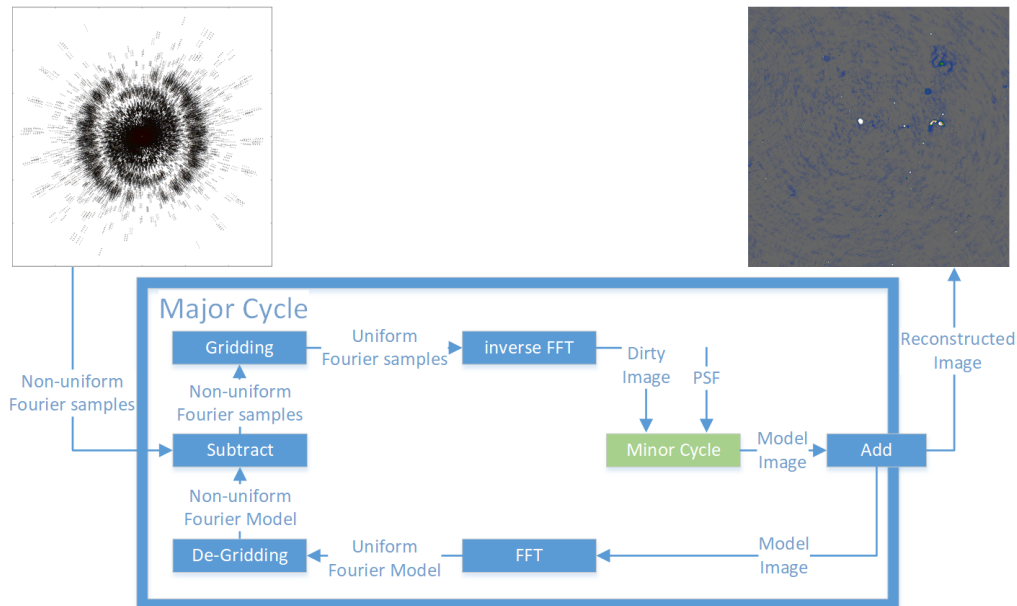


Figure 13: The Major Cycle Architecture of image reconstruction algorithms

The first operation in the Major Cycle, Gridding, takes the non-uniformly sampled Fourier measurements from the Interferometer and interpolates them on a uniformly spaced grid. The uniform grid lets us use FFT to calculate the inverse Fourier Transform and we arrive at the dirty image. A deconvolution algorithm takes the dirty image plus the *PSF* as input, producing the deconvolved "model image", and the residual image as output. At this point, the reverse operations get applied to the residual image. First the FFT and then De-gridding, arriving at the non-uniform Residuals. The next Major Cycle begins with the non-uniform Residuals as input. The cycles are necessary, because the Gridding and Deconvolution operations are only approximations. Over several cycles, we reduce the errors introduced by the approximate Gridding and Deconvolution. The final, reconstructed image is the addition of all the model images of each Major Cycle.

## 11.1 Distributed Image Reconstruction

New Interferometer produce an ever increasing number of measurements, creating ever larger reconstruction problems. A single image can contain several terabytes of Fourier measurements. Handling reconstruction problems of this size forces us to use distributed computing. However, state-of-the-art Gridding and Deconvolution algorithms only allow for limited distribution. How to scale the Gridding and Deconvolution algorithms to large problem sizes is still an open question.

Recent developments make a distributed Gridder and a distributed Deconvolution algorithm possible. Veeneboer et al[27] found an input partitioning scheme, which allowed them to perform the Gridding on the GPU. The same partitioning scheme can potentially be used to distribute the Gridding onto multiple machines. For Deconvolution, there exist parallel implementations for certain algorithms like MORESANE[2]. These can be used as a basis for a fully distributed image reconstruction.

In this project, we want to make the first steps towards an image reconstruction algorithm, which is distributed from end-to-end, from Gridding up to and including deconvolution. We create our own distributed Gridding and Deconvolution algorithms, and analyse the bottlenecks that arise.

## 11.2 First steps towards a distributed Algorithm

In this project, we make the first steps towards a distributed Major Cycle architecture (shown in figure 13) implemented C#. We port Veeneboer et al's Gridder, which is written in C++, to C# and modify it for distributed computing. We implement a simple deconvolution algorithm based on the previous project and create a first, non-optimal distributed version of it.

In the next step, we create a more sophisticated deconvolution algorithm based on the shortcomings of the first implementation. We use simulated and real-world observations of the MeerKAT Radio Interferometer and measure its speed up. We identify the bottlenecks of the current implementation and explore further steps.

From the first lessons, we continually modify the distributed algorithm and focus on decreasing the need for communication between the nodes, and increase the overall speed up compared to single-machine implementations. Possible Further steps:

- Distributed FFT
- Replacing the Major Cycle Architecture
- GPU-accelerated Deconvolution algorithm.

A state-of-the-art reconstruction algorithm has to correct large number of measurement effects arising from the Radio Interferometer. Accounting for all effects is out of the scope for this project. We make simplifying assumptions, resulting in a proof-of-concept algorithm.

## 12 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, September 18, 2019

Jonas Schwammberger