Centre for Modeling and Simulation
Savitribai Phule Pune University

Master of Technology (M.Tech.)
Programme in Modeling and Simulation

Project Report

Compressed Sensing for Radio Astronomy

Suraj Meghwani
CMS1002

Academic Year 2011-12

Centre for Modeling and Simulation
Savitribai Phule Pune University

Certificate

This is certify that this report, titled

Compressed Sensing for Radio Astronomy,

authored by

Suraj Meghwani (CMS1002),

describes the project work carried out by the author under our supervision during the period from January 2012 to June 2012. This work represents the project component of the Master of Technology (M.Tech.) Programme in Modeling and Simulation at the Center for Modeling and Simulation, Savitribai Phule Pune University.

Jayaram Chengalur, Professor
National Centre for Radio Astrophysics
(NCRA-TIFR), Pune 411007 India

Niruj Mohan Ramanujam, Scientist
National Centre for Radio Astrophysics
(NCRA-TIFR), Pune 411007 India

Mihir Arjunwadkar, Faculty
Centre for Modeling and Simulation
Savitribai Phule Pune University
Pune 411007 India

Anjali Kshirsagar, Director
Centre for Modeling and Simulation
Savitribai Phule Pune University
Pune 411007 India

Centre for Modeling and Simulation
Savitribai Phule Pune University

Author's Declaration

This document, titled

Compressed Sensing for Radio Astronomy,

authored by me, is an authentic report of the project work carried out by me as part of the Master of Technology (M.Tech.) Programme in Modeling and Simulation at the Center for Modeling and Simulation, Savitribai Phule Pune University. In writing this report, I have taken reasonable and adequate care to ensure that material borrowed from sources such as books, research papers, internet, etc., is acknowledged as per accepted academic norms and practices in this regard. I have read and understood the University's policy on plagiarism (`http://unipune.ac.in/administration_files/pdf/Plagiarism_Policy_University_14-5-12.pdf`).

Suraj Meghwani
CMS1002

Abstract

Compressed Sensing, also known as Compressive Sampling, is a novel framework that exploits the sparsity of a signal in some known basis. Recent theoretical proofs show that if a signal is sparse in a known basis then it is possible to reconstruct it via a small number of measurements. Furthermore, the signal can be accurately reconstructed using efficient algorithms. In this project we apply this technique in the Image reconstruction problem of Radio astronomy which is hampered by poor sampling and noisy Fourier measurements on aperture plane. The theory of Compressed Sensing demonstrates that such measurements may actually suffice for accurate reconstruction. Also, Compressed Sensings algorithms offers significant improvement over mainstream deconvolution algorithms used in radio interferometry such as CLEAN and MEM.

Acknowledgements

I am indebted to **Prof. Jayaram N. Chengalur, Prof. Mihir Arjunwadkar** and **Dr. Niruj Mohan Ramanujam** for their guidance on a daily basis during this project. I am thankful to Prof. Jayaram for sharing his own software for reading FITS files which made my work much easier than it would have otherwise been. Special thanks to Dr. Niruj Mohan Ramanujam for arranging several lectures on Radio Interferometry, specifically for me, despite his busy schedule. Prof. Mihir Arjunwadkar supported me a great deal on analysis and design of algorithms. I am also thankful to all of them for sharing their own wonderful reference articles with me and making me capable enough of carrying out such work independently in the future.

I would like to convey my utmost gratitude to National Centre for Radio Astrophysics, Tata Institute of Fundamental Research (NCRA-TIFR) for supporting my project and providing me with a wonderful working environment.

I hereby thank to Dr. Sheelan Chowdhury for being my internal guide and also all individuals from whom i benefited directly or indirectly.

Last but not the least i would like to thank my friends Rossi D'Souza, Sarvesh Nikumbh, Dinesh Mali and Ankit Dangi for their encouragement and help during my tenure at CMS and NCRA.

Contents

Chapter 1

Introduction

1.1 Compressed Sensing

Real world signals can be viewed as multidimensional vector and we know that image can be represented as a matrix with pixel entries as each element. It can also be viewed as multidimensional vector just by stacking the columns of the image matrix one after the other. In signal processing we measure samples of the signal in order to reconstruct the original signal. For reconstruction of such signals we consider linear measurement model

$$Ax = b \tag{1.1}$$

where, A is the known measurement matrix of order $M \times N$
M is the number of rows or number of features
N is the number of columns or number of unknowns
b is the $M \times 1$ known vector called measurement vector.
Each element of b vector is a measurement.
x is the $N \times 1$ unknown vector which we want to reconstruct.

This is our classical linear algebra problem which has three cases.

- $M = N$ then A is a square matrix and if A is non-singular then the system of equations has a unique solution.
- $M > N$ A matrix is a rectangular matrix with the number of rows (or number of equations) is greater than number of unknowns. Such systems are called Overdetermined systems and have no solution. Least Square method is one of the method for arriving at an approximate solution.
- $M < N$ then A matrix is a rectangular matrix with the number of rows smaller than the number of columns. Then, the system of equation is called as Under determined system and has a whole subspace of solution.

Suppose we have far fewer measurements than the number of features or unknowns i.e. $M << N$. Compressed Sensing can then provide a solution if we know that the signal is sparse ([10]).

1.1.1 Sparsity

With the success of image processing, where great advances in compression and estimation have come from modeling images as sparse in a wavelet domain (JPEG 2000 standard), modeling signals in a sparse basis has become a key area in signal processing. To illustrate this consider

an N dimensional vector y that can represented as Ψx, where Ψ is a $N \times N$ orthogonal matrix aka *Basis Matrix* and $x \in R^N$ has k non-zero entries. In this case we say that x is k-sparse or x is a k-sparse representation of y with respect to Ψ. The ratio $\alpha = k/N$ is called sparsity ratio of vector x.

Sparsity is the fundamental criterion for compressed sensing to work. Most signals (or images) are sparse when represented in certain sparsifying bases. An advantage in knowing the sparse representation of a signal is that the degree of freedom of signal the signal is reduced by a significant amount which leads to compression. Compression gain is higher when sparsity ratio is smaller.

Another important property of sparse signals is that it can be recovered in far less number of measurements then it would be otherwise. This technique is called as *Compressed Sensing or Compressive sampling* was developed by [7], [8], [10] .

The sampling of y is represented as a linear transformation by a matrix Φ yielding a measurement vector $b = \Phi y$.

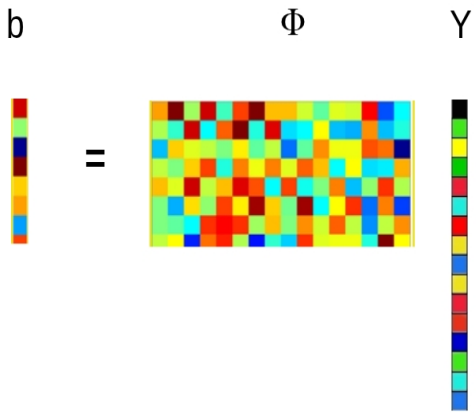


Figure 1.1: $b = \Phi y$

Φ is $M \times N$ matrix where $M << N$.Now suppose we have $y = \Psi x$ where x is a k-sparse representation of y and Ψ is the basis matrix.

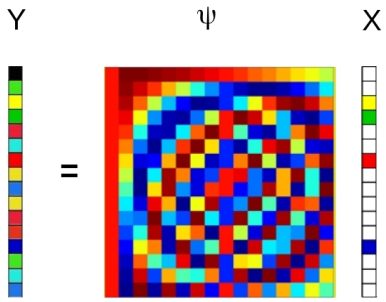
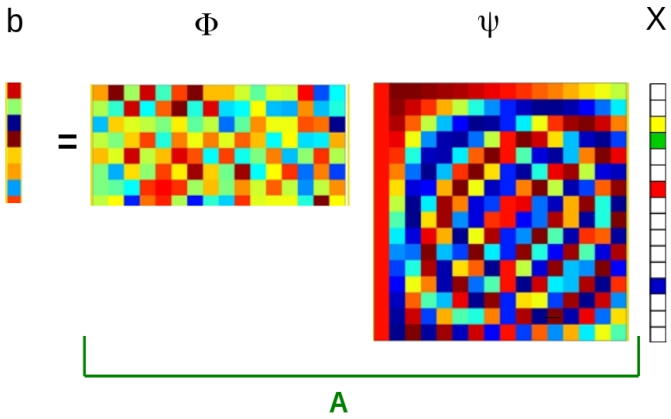


Figure 1.2: $y = \Psi x$

The complete idea of compressed sensing is shown in following figure.



In figure 1.1.1 equation $b = \Phi\Psi x$ is in $Ax = b$ form with $A = \Phi\Psi$. Since x is k-sparse, x must be in one of the $C(n, k)$ subspaces of R^N. So if we have $m > k + 1$ then exhaustive search can be done through these subspaces and we can determine which subspace y belongs to. This can be formulated as an optimization problem

$$\text{minimize} \parallel x \parallel_{\ell_0} \text{subjected to } Ax = b \tag{1.2}$$

where ℓ_0 is the quasi-norm and is defined as

$$\parallel x \parallel_{\ell_0} = \sum_{i=1}^{i=n} I(x_i \neq 0) \tag{1.3}$$

where I is an indicator function defined as

$$I(x) = \begin{cases} 1 & x \neq 0 \\ 0 & x = 0 \end{cases} \tag{1.4}$$

But performing this exhaustive search over all subspaces is computationally intractable as N increases (keeping α constant). One way of circumventing this problem is by using the ℓ_2 norm

as a proxy, which is equivalent to solving the following optimization problem.

$$\text{minimize} \parallel x \parallel_{\ell_2} \text{ subjected to } Ax = b \tag{1.5}$$

Solving the above equation leads to a solution for $Ax = b$ but it only guarantees x to be small in the ℓ_2 norm sense. In other words it does not help in reducing the number of non-zero components in the solution vector x which is our prime aim.

Hence, minimizing the ℓ_0-norm is computationally intractable and ℓ_2-norm does not lead to sparse solutions. A novel way of getting a sparse solution is by minimizing the ℓ_1-norm subjected to $Ax = b$.

$$\text{minimize} \parallel x \parallel_{\ell_1} \text{ subjected to } Ax = b \tag{1.6}$$

Solving this optimization problem leads to a sparse solution. It also guarantees the uniqueness in solution as proved in .

1.1.2 ℓ_1 vs ℓ_2

One way of visualizing why ℓ_1 leads to sparse solution and ℓ_2 does not is shown in the following figure.

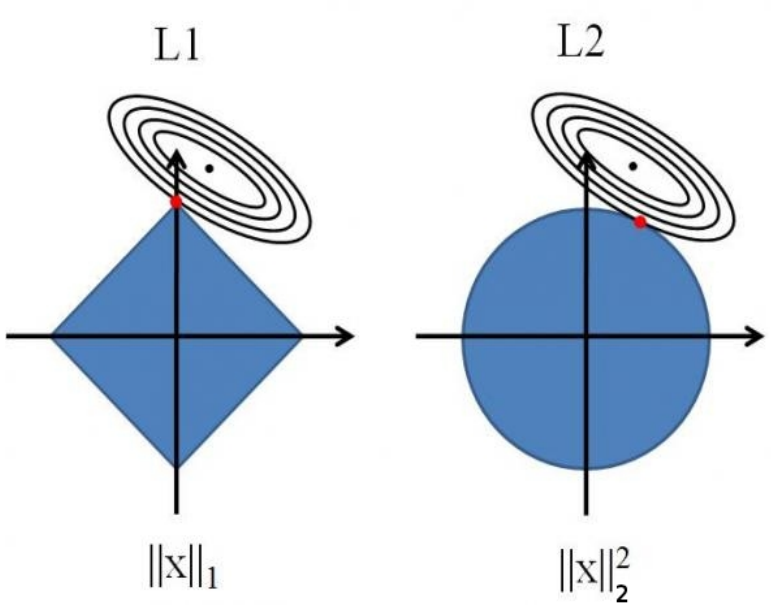


Figure 1.3: ℓ_1 vs ℓ_2 ball

Figure (1.3) shows the ℓ_1 ball and ℓ_2 ball touching the contours of $\parallel Ax - b \parallel_{\ell_2}^2$. This clearly shows that due to the sharper corners of the ℓ_1 ball, contours of $\parallel Ax - b \parallel_{\ell_2}^2$ touch the ℓ_1 ball at one of the axes, which is a sparse solution. However the ℓ_2 ball is spherical and does not leads to a sparse solution.

In general any p-norm is defined as

$$\| x \|_p = (|x_1|^p + |x_2|^p + |x_3|^p + ...|x_N|^p)^{\frac{1}{p}} \tag{1.7}$$

Further, an equivalent problem of minimizing the p-norm of x is

$$minimize(|x_1|^p + |x_2|^p + |x_3|^p + ...|x_N|^p) \tag{1.8}$$

Let us compare the minimization of the ℓ_1 norm and the ℓ_2 norm. For the ℓ_2 norm minimization, we care very little when $x_i < 1$ as compared to ℓ_1 minimization because ℓ_2 norm deteriorates the smaller components more rapidly as compared toℓ_1 norm. On the other hand during ℓ_2 minimization we have a strong dislike when if $x_i > 1$. Hence, during minimization ℓ_2 puts strong weights on larger components and a small weight (or zero weight) on smaller components of x as compared to the ℓ_1 norm. So ℓ_1 norm minimization places a larger emphasis on minimizing the smaller components which in turn makes most of the components zero or approximately zero. This simple interpretation gives a clearer insight into why a solution of equation (1.6) leads to sparse result.

1.1.3 Theorems for Compressed Sensing

Restricted Isometry Property (RIP) or Uniform Uncertainty Principle (UUP)

UUP stands for Uniform Uncertainty Principle ([6], [5]). Matrices which follow these principle are called "UUP matrices". UUP matrices can compress vectors from high-dimensional space to low-dimensional space while still being able to reconstruct these vectors.

In general, a $M \times N$ rectangular matrix A is called orthogonal only if $M \geq N$ and each column vector are orthonormal i.e. they all have unit length and are orthogonal to each other. Let $v_1, v_2...v_N$ be the 'N' columns of A matrix. Then

$$v_i^* v_j = 0 \qquad \text{where } i \neq j \text{ and } i, j \text{ is from } 1...N \tag{1.9}$$

$$\| v_i \|_{\ell_2} = 1 \qquad \text{where } i \text{ is from } 1...N \tag{1.10}$$

So this A matrix can transform an N-dimensional vector $a = (a_1, a_2...a_N)$ to corresponding M dimensional vector, which is $\sum_{j=1}^{j=N} a_j v_j$. Now if this A matrix is orthogonal, we know that norm of vectors are not changed when they are linearly transformed by an orthogonal matrix. So for above situation, we have

$$\| \sum_{j=1}^{j=N} a_j v_j \|_{\ell_2}^2 = \sum_{j=1}^{j=N} |a_j|^2 \tag{1.11}$$

This implies that $(a_1, a_2, ...a_N) \mapsto \sum_{j=1}^{j=N} a_j v_j$ is an isometry i.e. given the encoded vector $z = \sum_{j=1}^{j=N} a_j v_j$ one can uniquely recover the original coefficients $(a_1, a_2...a_N)$. Furthermore, small

changes in z will not cause large fluctuations in $(a_1, a_2...a_n)$. Indeed, one can reconstruct the original coefficients quickly and explicitly by the formula

$$a_j =< z, v_j > \tag{1.12}$$

But in the Compressed Sensing setting we cannot apply this property directly as we have $M << N$. In other words we cannot express a vector of say in 't' dimension with a basis vectors set of cardinality greater than t. So for our case

$$a_1 v_1 + a_2 v_2 + ... + a_n v_N = 0 \tag{1.13}$$

If we can try to overcome the above restriction by weakening equation (1.11) for all $(a_1, a_2, ...a_n)$ by

$$(1-\delta_1)\sum_{j=1}^{j=n}|a_j|^2 \quad \leq \quad \| \sum_{j=1}^{j=n} a_j v_j \|_{\ell_2}^2 \quad \leq \quad (1+\delta_2)\sum_{j=1}^{j=n}|a_j|^2 \tag{1.14}$$

where δ_1, δ_2 are constants such that $0 < \delta_1\delta_2 < 1$. The above condition is as good as saying columns of the matrix are locally orthogonal rather than globally perfectly orthogonal. Due to this property which is also called as *Restricted Isometry Property ([6, 5])*, it turns out that one can pack more than M vectors into M dimensional space if one localizes the almost orthogonality condition so that it only holds for sparse sets of coefficients $(a_1, a_2, ...a_N)$. Sparsity parameter k (less than M) plays an important role here. If we say that our sensing matrix A obeys the UUP with sparsity k then one has the almost orthogonality condition (1.14) for any set of coefficients $(a_1, a_2, ...a_N)$, such that at most k of the a_j are non-zero. In other words, we only assume that any k of the N vectors $v_1, v_2, ...v_N$ are almost orthogonal at one time.

Furthermore, constructing UUP matrices is an NP hard problem. In other words there is no polynomial time algorithm known which can construct these UUP matrices. The best possible methods are the one which could simply search through all matrices in a given class and test each one of them for the UUP property. But this is an exponential-time algorithm. However it has also seen that some matrices like random normalized Gaussian matrices, random normalized Bernoulli matrices are UUP matrices. But they are all probabilistic in nature; in particular, these constructions are not 100% guaranteed to actually produce a UUP matrix, although in many cases the failure rate can be proven to be exponentially small in the size of the matrix. So for many larger scale applications it is actually exponentially less prone to failure.

Definition. *UUP matrices are the generalization of (rectangular) orthogonal matrices in which columns are locally almost orthogonal rather than globally perfectly orthogonal.*

Incoherence

In Compressed Sensing we are dealing with two matrices Φ (sensing Matrix) and Ψ (Basis matrix which is used to represent y in sparse basis). Coherence between the sensing basis and Basis matrix defined as

$$\mu(\Phi, \Psi) = \sqrt{N} max_{1 \leq k,j \leq N} | < \Psi_k, \Phi_j > | \tag{1.15}$$

In simple words, coherence(μ) measures the largest correlation between two elements of Φ and Ψ. If Φ and Ψ contain correlated elements, then the coherence is large. Compressed Sensing is mainly concerned with low coherence pairs i.e. we require low values of μ. Classical

example of incoherent basis pairs are the Fourier basis and Euclidean basis. These basis pairs are maximally incoherent with incoherence $\mu = 1$. Another example is pair of wavelets bases (Ψ) and noiselets (Φ). The coherence between noiselets and Haar wavelets is $\sqrt{2}$ and between noiselets and Daubechies D4 and D8 wavelets, respectively, about 2.2 and 2.9. Noiselets are also maximally incoherent with euclidean basis and Fourier basis.

Finally, random matrices are largely incoherent with any fixed basis Ψ. We select an orthobasis Ψ uniformly at random, which can be done by orthonormalizing N vectors sampled independently and uniformly on the unit sphere. Then with high probability, the coherence between Ψ (Basis Matrix) and Φ (Sensing basis matrix) is $\simeq \sqrt{2logN}$. Random Gaussian i.i.d matrices also show very low coherence with fixed representation basis matrix Ψ.

THEOREM ([5]) 1. *Suppose $y \in R^N$ is sparse in some Ψ basis and $x \in R^N$ is k-sparse representation of y. Now we select M measurements in Φ uniformly at random. Then if*

$$M \geq C\mu^2(\Phi, \Psi)klogN \tag{1.16}$$

for some constant C the solution 1.6 is exact with overwhelming probability.

The role of coherence is now clear; the smaller the coherence, the fewer samples are needed, hence compressed sensing require low coherence pairs of basis matrices. One suffers no information loss by measuring just about any set of M coefficients which may be far less what the signal size apparently demands. If $\mu(\Phi, \Psi)$ is equal or close to one, then of the order of $klogN$ samples suffice instead of N.

1.1.4 Applications

The fact that sparse signals can be reconstructed efficiently using very few number of incoherent samples which is proportional to sparsity (k) has far reaching implications and applications. The main areas where compressed sensing had proven itself as a more advantageous and efficient technique are

- *Data compression.* In some situations, the sparse basis Ψ may be unknown at the encoder or impractical to implement for data compression. However a randomly designed Φ can be considered a universal encoding strategy, as it need not be designed with regards to the structure of Ψ. This universality may be particularly helpful for distributed source coding in multi-signal settings such as sensor networks ([1]).

- Channel coding. Compressed Sensing principles can be turned around and applied to design fast error correcting codes over the reals to protect from errors during transmission ([6]).

- *Inverse problems.* The only way to acquire x may be to use a measurement system Φ of a certain modality. However, assuming a sparse basis Ψ exists for x that is also incoherent with Φ, then efficient sensing will be possible. One such application involves MR angiography and other types of MR setups, where Φ records a subset of the Fourier transform, and the desired image x is sparse in the time or wavelet domains ([14]).

- *Data acquisition.* Finally, in some important situations the full collection of N discrete-time samples of an analog signal may be difficult to obtain (and possibly difficult to subsequently compress). Here, it could be helpful to design physical sampling devices that directly record discrete, low-rate incoherent measurements of the incident analog signal.

1.2 Radio Interferometry

1.2.1 Radio Telescopes and its Resolution

Angular resolution (θ) of a telescope depends on the wavelength of light or radio waves (λ) and the diameter (D) of the telescope being used. The relation between θ, λ and D is given by

$$\theta \sim \lambda/D \tag{1.17}$$

Here θ is in radians and λ and D in meters.

Radio telescopes are used to study radio waves emitted by astronomical objects of wavelengths roughly between about 10 meters and 1 millimeter. It is possible to observe radio waves from the ground as shown in the figure 1.4, spacecraft are needed to observe astronomical objects in gamma rays, X-rays, UV, and IR, while ground observations are possible in the visible, some parts of the near IR, and the radio.

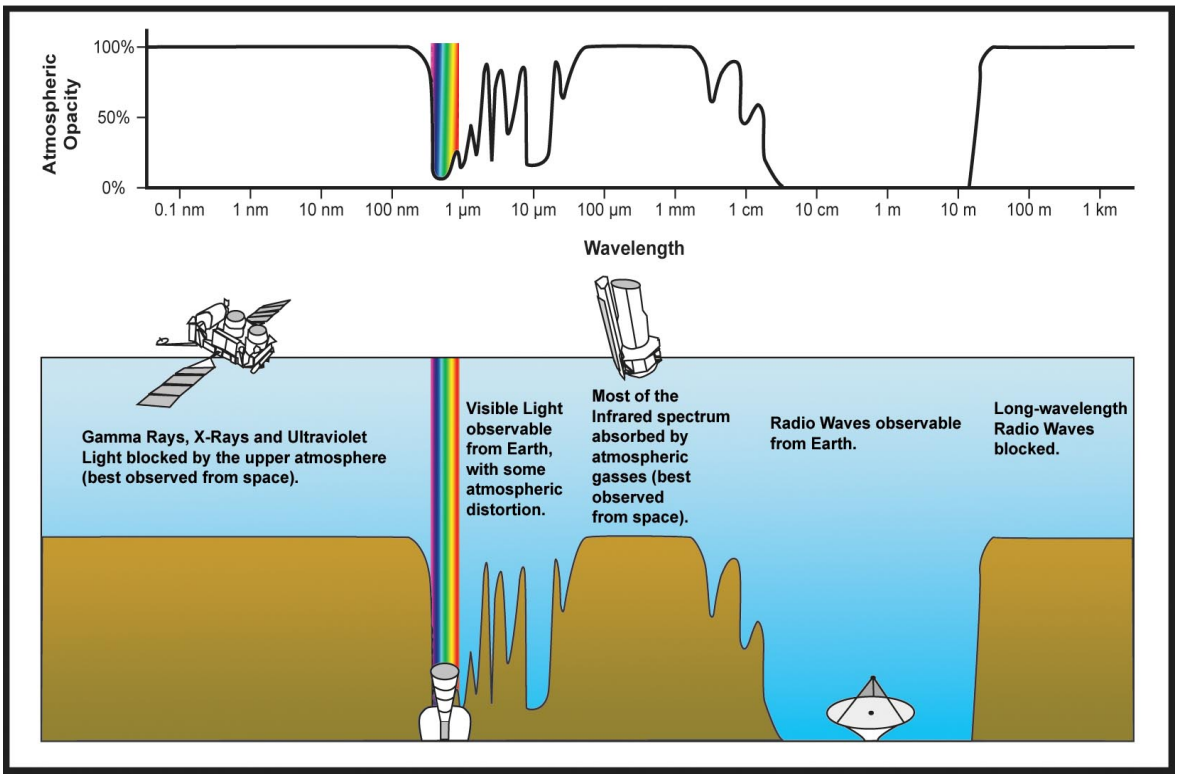


Figure 1.4: Image credit : NASA/IPAC

Due to larger operating wavelengths and primarily because of relation equation (1.17), radio telescopes have to be much larger than optical telescopes to attain good angular resolution. Angular resolution is a measure of how small detail of an area in the sky can be seen. The larger the telescope, the more detail can be observed in a given wavelength.

1.2.2 Radio Interferometry

Angular resolution of telescope given by equation (1.17) is limited by its operating wavelength and diameter of aperture. To get good resolution, one has to either decrease the operating wave-

length or increase the size of telescope. In radio astronomy, the wavelengths are so large that even though the sizes of radio telescopes are large, the angular resolution is poor as compared to optical instruments. With these larger telescopes one can achieve higher resolutions by operating wavelengths in the centimeter to millimeter range. Telescopes of hundreds of meters in diameter are needed to observe the source at metre wavelengths which would clearly impractical.

Radio astronomers achieve higher resolution at metre wavelengths using a technique called *Radio Interferometry*. The interferometers like GMRT ([17]) consist of array of antennas located on ground and resolution of such a telescope is proportional to maximum projected distance between the antennas. The separations between antennas are called as *Baseline* and are measured in units of wavelength (λ).

Van Cittert –Zernike theorem

Van Cittert –Zernike theorem is the key theorem in radio interferometric imaging. It relates the spatial coherence function at two points on the ground also called Visibility function ($V(r_1, r_2)$) with the distribution of source intensity I. It also shows that the visibility function $V(r_1, r_2)$ depends only on the relative distances between two points i.e $r_1 - r_2$

As explained in [16]. Let us consider a two element interferometer with antenna 1 and antenna 2 located on ground at $P_1(x_1, y_1, z_1)$ and $P_2(x_2, y_2, z_2)$ respectively. Consider an infinitesimal element of the source positioned at $P(x, y, z)$ in the sky shown in figure (). If electric field intensity at the point P is given by $\varepsilon(P)$, then the electric field at the observation point P_1 [3] is given by

$$E(P_1) = \int \varepsilon(P) \frac{e^{-\frac{2\pi}{\lambda} D(P_1,P)}}{D(P_1, P)} d\Omega \tag{1.18}$$

where $D(P_1, P)$ is defined as the distance between the points P and P_1. Ω is the solid angle subtend by the infinitesimal source at P.Assuming that the emission from the source is spatially incoherent, at point P_2 the electric field is given by

$$E(P_2) = \int \varepsilon(P) \frac{e^{-\frac{2\pi}{\lambda} D(P_2,P)}}{D(P_2, P)} d\Omega \tag{1.19}$$

The spatial coherence function is given by

$$< E(P_1)E^*(P_2) >= \int < \varepsilon(P)\varepsilon^*(P) > \frac{e^{-\frac{2\pi}{\lambda}(D(P_2,P)-D(P_2,P))}}{D(P_1, P)D(P_2, P)} d\Omega \tag{1.20}$$

where Intensity(I) of the source is defined as

$$I(P) =< \varepsilon(P)\varepsilon^*(P) > \tag{1.21}$$

If we imagine a celestial sphere of radius R and approximate the source lying on this sphere then we have $x = Rcos(\theta_x)$ $y = Rcos(\theta_y)$ and $z = Rcos(\theta_z)$ where $cos(\theta_x), cos(\theta_y), cos(\theta_z)$ are direction cosine l, m, n. Also $l^2 + m^2 + n^2 = 1$ and the solid angle $\Omega = \frac{dldm}{\sqrt{1-l^2-m^2}}$. Further

$$\begin{aligned} D(P_1,P) &= [(x-x_1)^2+(y-y_1)^2+(z-z_1)^2]^{\frac{1}{2}} \\ &= [(Rl-x_1)^2+(Rm-y_1)^2+(Rn-z_1)^2]^{\frac{1}{2}} \\ &= R[(l-x_1/R)^2+(m-y_1/R)^2+(n-z_1/R)^2]^{\frac{1}{2}} \\ &\simeq R[(l^2+m^2+n^2)-2/R(lx_1+my_1+nz_1)]^{\frac{1}{2}} \\ &\simeq R-(lx_1+my_1+nz_1) \end{aligned}$$

Similarly we have

$$D(P_2,P) = R-(lx_2+my_2+\sqrt{1-l^2-m^2}z_2) \tag{1.22}$$

Using above equations we can approximate $D(P_1,P)D(P_2,P)$ as R^2 as shown below

$$\begin{aligned} D(P_1,P)D(P_2,P) &= \{R-(lx_1+my_1+nz_1)\}\{R-(lx_2+my_2+nz_2)\} \\ &= R(R-(l(x_1+x_2)+m(y_1+y_2)+z(z_1+z_2)) \\ &\quad +(lx_2+my_2+nz_2)(lx_1+my_1+nz_1) \end{aligned}$$

In above equation $R-(l(x_1+x_2)+m(y_1+y_2)+z(z_1+z_2)\simeq R$ and further $R^2 >> (lx_2+my_2+nz_2)(lx_1+my_1+nz_1)$. Hence we have

$$D(P_1,P)D(P_2,P) \simeq R^2 \tag{1.23}$$

Putting the above equations (1.21),(1.23) in equation (1.20) we have

$$< E(P_1)E^*(P_2) >= \frac{1}{R^2}\int I(P)e^{-\frac{2\pi}{\lambda}(l(x_2-x_1)+m(y_2-y_1)+n(z_2-z1))}\frac{dldm}{\sqrt{1-l^2-m^2}} \tag{1.24}$$

Above equation () is the full Van Cittert-Zernike equation. It can be seen from above equation spatial coherence function or visibility function $V(r_1,r_2) =< E(P_1)E^*(P_2) >$ depends upon r_1-r_2. We define baseline co-ordinates u,v,w such that $u=(x_2-x_1)/\lambda$, $v=(y_2-y_1)/\lambda$ and $w=(z_2-z_1)/\lambda$ and also since we have relation $l^2+m^2+n^2=1$ the above equation reduced to

$$V(u,v,w) = \frac{1}{R^2}\int I(l,m)e^{-\frac{2\pi}{\lambda}(lu+mv+(\sqrt{1-l^2-m^2})w)}\frac{dldm}{\sqrt{1-l^2-m^2}} \tag{1.25}$$

Two-Dimensional Approximation

Equation (1.25) under certain assumption can be approximated as two-dimensional Fourier transform of the source intensity distribution. If we confined our observation to $u-v$ plane i.e when $w=0$ then we have

$$V(u,v) = \frac{1}{R^2}\int I(l,m)e^{-\frac{2\pi}{\lambda}(ul+vm)}\frac{dldm}{\sqrt{1-l^2-m^2}} \tag{1.26}$$

Another approximation to above equation is when we consider that the source brightness distribution is limited only to small region of sky. In this case $n=\sqrt{1-l^2-m^2}\simeq 1$. Then the equation (1.25) becomes

$$V(u,v) = \frac{1}{R^2}\int I(l,m)e^{-\frac{2\pi}{\lambda}(ul+vm)}dldm \tag{1.27}$$

Source brightness is described in lm plane while the equivalent conjugate variables in Fourier space u,v is described in uv plane.u,v is interpreted as spatial frequency and Visibility function $V(u,v)$ as spatial frequency spectrum of source brightness distribution.

1.2.3 Dirty Beam and Dirty Image

Fourier transform relation exists between source brightness I and the visibility V we can take an inverse Fourier transform which can be given by

$$I(l,m) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} V(u,v)e^{2\pi i(ul+vm)}dudv \tag{1.28}$$

Since it is impractical to cover whole uv plane, we measure visibilities only at some points. Due to which image of sky is not the true image but comes with lot of noise such an image is called Dirty Image is given by

$$I^D(l,m) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} S(u,v)V(u,v)e^{2\pi i(ul+vm)}dudv \tag{1.29}$$

where $V(u,v)$ is observed visibility and $S(u,v)$ denote sampling function which is indicator of where on $u-v$ plane visibilities are measured. It is given by

$$S(u,v) = \delta(u-u_k, v-v_k) \tag{1.30}$$

Hence practically we measure sampled Visibility function ($V^S(u,v)$) which is given by

$$V^S(u,v) = \delta(u-u_k, v-v_k)V(u_k,v_k) \tag{1.31}$$

where (u_k, v_k) are the co-ordinates of points where visibility is measured. Hence the discretized form of visibility function with discretized $l-m$ grid with reference to equation (1.27) is given by

$$V(u_j,v_j) = \sum_{k=1}^{k=N} I(l_k,m_k)e^{2\pi i(u_j l_k + v_j m_k)} \tag{1.32}$$

Note above equation denotes only one visibility at (u_j, v_j).
In short we have $V^S = SV$. If F denote Fourier operator then equation (1.29) can be written as,

$$I^D = FV^S = F(SV) \tag{1.33}$$

By Convolution theorem ([4]),

$$I^D = FS * FV \tag{1.34}$$

For a point source of unit strength at (l_0, m_0), $|V(u,v) = 1|$
$FV = \delta(l-l_0, m-m_0)$
$I^D = FS * FV = FS * \delta = FS.$
This is called the Dirty Beam which is given by

$$B = FS \tag{1.35}$$

Hence putting equation (1.35) in equation (1.34). Dirty Image is defined as the convolution of dirty beam and true image (FV).

$$I^D = B * FV \tag{1.36}$$

1.2.4 Deconvolution

The deconvolution problem in equation (1.36) corrects for the $u-v$ plane sampling effect. The lack of measurement at certain interferometric spacing means that in principle an infinite number of brightness distributions could be consistent with our visibility data. On the other

hand, we may incorporate additional information to constrain our solutions. The most widely used deconvolution method are CLEAN ([12]) and Maximum Entropy Minimization

CLEAN Algorithm

CLEAN algorithm consider sky as containing only isolated point sources and approximates an image by a collection of point sources. It then deconvolve each point source using a simple iterative approach. The final deconvolved image which is also called CLEAN image is the sum of these CLEAN components convolved with a Gaussian beam. This algorithm was published by [12] in 1974 and several variations have been proposed since then. Following are the main steps of the CLEAN algorithm.

- Find the peak in Dirty Image.
- Subtract a dirty beam (B) of approximate strength to remove its side-lobes.
- Construct CLEAN image I' using position and magnitude of subtracted components.
- Continue looping until some threshold level is met.
- Convolve the final I' with an idealized CLEAN beam.

Maximum Entropy Minimization

MEM method minimizes a smoothness function ("entropy") in an image. Maximum entropy is also called the all-poles model or auto-regressive model. For images with more than a million pixels, this algorithm is faster than CLEAN algorithm. Details of the algorithm can be found in [].

1.2.5 GMRT

National Centre for Radio Astrophysics (NCRA) has set up an international facility for radio astronomical research in the metre wavelengths range of the radio spectrum known as the Giant Metre wave Radio Telescope (GMRT). This is the world's largest array of radio telescopes at metre wavelengths and is located at a site about 80 km north of Pune. GMRT consists of 30 fully steerable gigantic parabolic dishes of 45m diameter each spread over distances of upto 25 km. There are fourteen telescopes randomly arranged in the central square 1 km by 1 km in size, with a further sixteen arranged in three arms of a nearly "Y"-shaped array each having a length of 14 km from the array centre.

Antenna feeds at five different frequency centered at 153, 233, 327, 610, and 1420 MHz, with polarization. The maximum baseline in the array gives the telescope an angular resolution (the smallest angular scale that can be distinguished) of about 1 arc-second at the frequency of 1420 MHz.

1.2.6 Compressed Sensing formulation for Radio Interferometry

Equation (1.27) shows that under certain approximation, Visibility function is the Fourier transform of source brightness. Radio Interferometers like GMRT measures visibility at very few points on $u-v$ plane. At a time GMRT measures $435(C(30,2))$ visibilities. Each pair of antenna measures one visibility. This is shown in figure[1](1.6) where each point represent one visibility.

[1]Image by A. Basu, NCRA

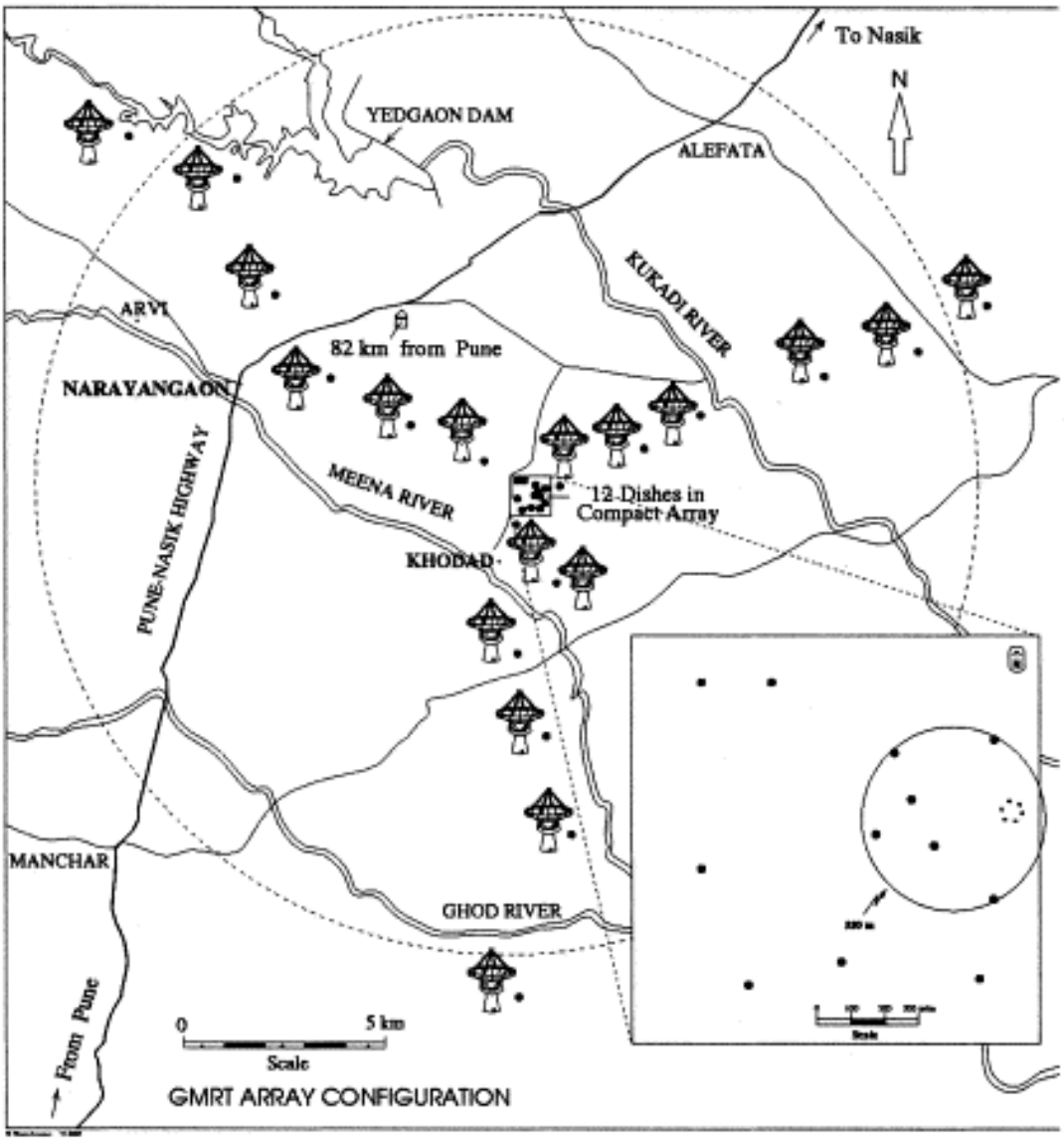


Figure 1.5: "Y" shaped Array

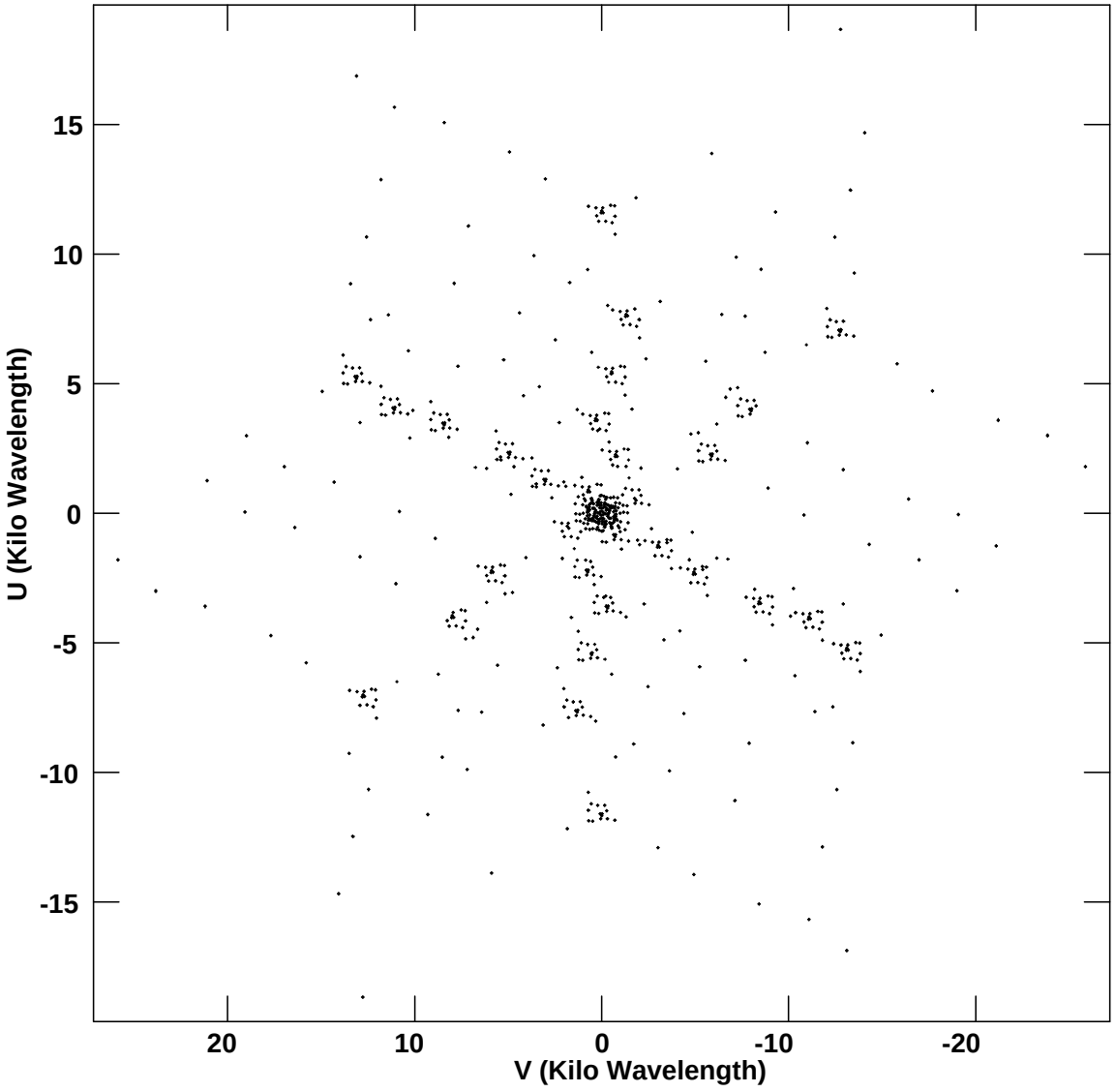


Figure 1.6: *uv* coverage at a time

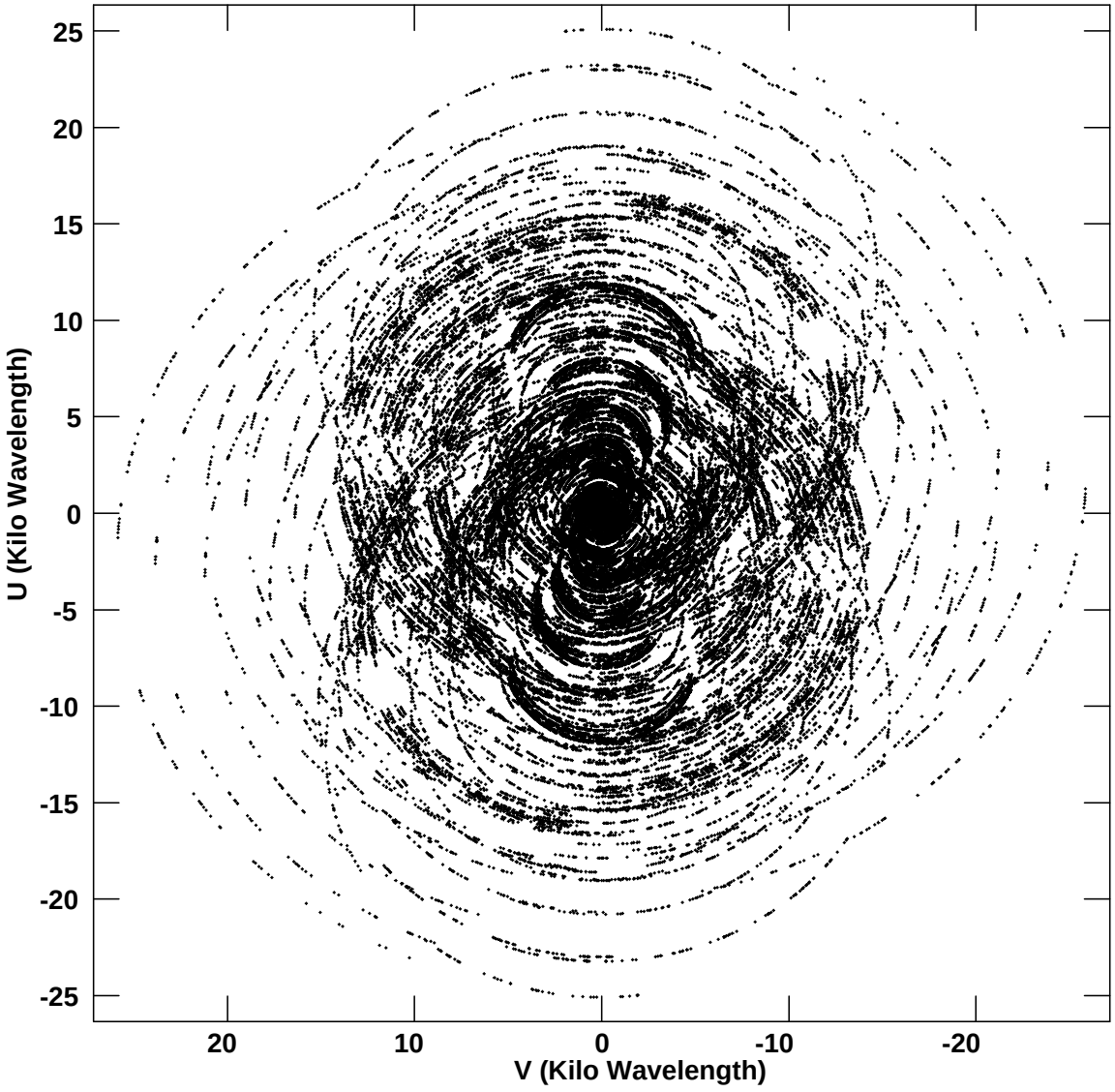


Figure 1.7: *uv* coverage for few hours

Due to rotation of earth, the projected separation between the antenna change and over certain time it covers more point on the $u - v$ plane. This is shown in figure [2](1.7)

Even after few hours of uv coverage (figure :1.7) uv coverage is not uniform. Clearly we have under-sampled visibility data and hence Compressed Sensing methods proven to be viable option. Equation (1.27) can be cast in to compressed sensing setting with visibilities as measurement vector and partial 2-D Fourier matrix as measurement matrix and sky image as the unknown. In radio astronomy, most of the sources are point source so, most of the time image is sparse in euclidean image. However there are the cases in which we have to represent image in some sparse basis which can be done with some extra effort as in case of other application of compressed sensing.

1.2.7 Why Compressed Sensing in Radio Astronomy?

In general, Compressed Sensing is advantageous when signals are sparse in some known basis, measurements are expensive and computations are cheaper at receiving end. These situations make Compressed Sensing methods a strong candidate for image deconvolution problem in radio astronomy.

Compressed Sensing methods can directly reconstruct the image from the sparse observations, without the need for an iterative deconvolution. Also it will be very efficient technique, when the 2-D approximation to Van Cittert-Zernike Theorem breaks. Further with the advent of recent efficient and robust ℓ_1 based algorithms, Compressed Sensing takeover mainstream deconvolution methods used in Radio Astronomy like CLEAN and Maximum Entropy minimization as discussed in section 2.4.

[2]Image by A. Basu, NCRA

Chapter 2

Algorithms for Compressed Sensing

2.1 Algorithms

Convex Optimization is a sub field of Optimization which can produce reliable solutions and can be solved exactly. Many signal processing problems can be formulated as convex optimization problems of form

$$minimize_{x \in R^N} \quad f_1(x) + f_2(x) + ... + f_{n-1}(x) + f_n(x) \tag{2.1}$$

where $f_1, f_2, ..., f_n$ are convex functions defined from $f : R^N \rightarrow R$ where some of the functions are non-differentiable, this which rules out our conventional smooth optimization techniques like Steepest decent method, conjugate gradient method etc. We have explored the class of algorithms which can solve equation (2.1). These methods proceed by splitting, in that the functions $f_1, ..., f_n$ are used individually so as to yield an easily implementable algorithm. They are called proximal because each non smooth function in 2.1 is involved via its proximity operator. Iterative thresholding, projected Landweber, projected gradient, alternating projections, alternating-direction method of multipliers, alternating split Bregman are special instances of proximal algorithms. Details of proximal methods are discussed in [9].

Notations

Here after, R^N is the N-dimensional euclidean space and domain of function $f : R^N \rightarrow]-\infty, +\infty]$.

Let C be the non-empty convex subset of R^N. The indicator function of C is defined as

$$i_C : x \mapsto \begin{cases} 0 & \text{if } x \in C \\ +\infty & \text{if } x \notin C \end{cases} \tag{2.2}$$

p-norm is defined as ($\| \, . \, \|_p$)

$$\| \, x \, \|_p = (|x|^p + |x|^p + |x|^p + ...|x|^p)^{\frac{1}{p}} \tag{2.3}$$

The distance form $x \in R^N$ to C is defined as

$$D_C(x) = min_{y \in C} \| \, x - y \, \| \tag{2.4}$$

If C is closed and convex, the projection of $x \in R^N$ onto C is the unique point $P_C x \in C$ such that $D_C(x) = \| x - P_C x \|_2$.

Sub differential of f is given by

$$\partial f : R^N \to 2^{R^N} : x \mapsto u \in R^N | (\forall y \in R^N)(y-x)^T u + f(x) \leq f(y)) \tag{2.5}$$

2.1.1 Proximal Gradient methods

One of the widely used convex optimization algorithm is POCS (Projection Onto Convex Sets).This algorithm is employed to recover/synthesize a signal satisfying simultaneously several convex constraints. Let f_i be the indicator function of non-empty closed convex set C_i modeling a constraint. This reduces to convex feasibility problem, which which require us to find a solution such that it lies in the intersection of all convex sets C_i. In POCS method each set C_i is incorporated by its projection operator P_{C_i}. So in each iteration x is updated as

$$x_{k+1} = P_{C_1} P_{C_2} ... P_{C_n} x_k \tag{2.6}$$

However beyond such problems Projection operators are not appropriate and more general operators are required to tackle them. Among the various generalizations of the notion of a convex projection operator that exist, proximity operators are best suited for our purposes.

Proximity operators of function f at x is defined as

Definition. *For every $x \in R^N$, the minimization problem,*

$$minimize_{y \in C} \quad f(y) + \frac{1}{2} \| x - y \|_2^2 \tag{2.7}$$

admits a unique solution which is denoted by $prox_f(x)$.

$$prox_f(x) : R^N \to R^N \tag{2.8}$$

The proximity operator of f is characterized by inclusion

$$(\forall (x,p) \in R^N \times R^N) \quad p = prox_f(x) \Leftrightarrow x - p \in \partial f(p) \tag{2.9}$$

If f is differentiable then above equation reduces to

$$(\forall (x,p) \in R^N \times R^N) \quad p = prox_f(x) \Leftrightarrow x - p \in \triangledown f(p) \tag{2.10}$$

We are interested in algorithms which can solve equation (1.6) where $n = 2$ in reference with equation (2.1). For notational convenience we use $f(x)$ in place of $f_1(x)$ and $\lambda g(x)$ in place of $f_2(x)$. Hence our optimization problem is

$$minimize_{x \in R^N} \quad f(x) + g(x) \tag{2.11}$$

collectively we say $F(x) = f(x) + g(x)$ where $f(x) = \| Ax - b \|_2^2$ and $g(x) = \lambda \| x \|_1$

2.1.2 Derivation of Soft Thresholding

In our problem $g(x) = \lambda \parallel x \parallel_1$ and $prox_g(x)$ is called soft shrinkage thresholding[1] operator which minimizes following equation.

$$prox_g x = \quad argmin_y \tfrac{1}{2} \parallel x - y \parallel_{\ell_2}^2 + \lambda \parallel y \parallel_{\ell_1} \tag{2.12}$$

Using definition of the ℓ_1-norm and the ℓ_2-norm we have

$$prox_g x = \quad argmin_y \tfrac{1}{2} \sum_{i=1}^{i=n} (x_i - y_i)^2 + \lambda \sum_{i=1}^{i=n} |y_i| \tag{2.13}$$

To find the minimizer in the above minimization problem, we have to consider three case when $y_i > 0$, $y_i < 0$, $y_i = 0$.

Case 1 : $y_i > 0$

To get the gradient of objective function in equation () we have to differentiate it with respect to each component and then equating each components to zero, the i_{th} component is given by

$$y_i = \quad x_i - \lambda \tag{2.14}$$

Case 2 : $y_i < 0$

Similarly, considering each component y_i ¡ 0, the minimizer will be given as

$$y_i = \quad x_i + \lambda \tag{2.15}$$

Case 3 : $y_i = 0$

We have to find the condition for $y_i = 0$ to be the unique minimizer. Then for

$$\begin{array}{l} \Rightarrow \frac{1}{2}(x_i^2 - 2x_i y_i + y_i^2) + \lambda |y_i| > \frac{1}{2} x_i^2 \\ \Rightarrow y_i + \frac{1}{2} y_i^2 + \lambda |y_i| > 0 \\ \Rightarrow \begin{cases} y_i + \frac{1}{2} y_i^2 + \lambda y_i > 0 & \text{for } y_i > 0 \\ y_i + \frac{1}{2} y_i^2 - \lambda y_i > 0 & \text{for } y_i < 0 \end{cases} \end{array}$$

The above inequalities will hold anyways for sufficiently large y_i. Let us therefore consider $|y_i|$ to be arbitrarily close to zero. If the inequalities above are to hold for any arbitrarily small $|x_i|$, we must have

$$|x_i| \leq \lambda \text{ for } y_i = 0 \tag{2.16}$$

Combining equation (2.14) and (2.15) we get

$$prox_g x = (|x_i| - \lambda) \times sgn(y_i) \text{ if } |y_i| > 0 \tag{2.17}$$

[1]Derived by Prof. Mihir Arjunwadkar

where,

$$sgn(y_i) = \begin{cases} 1 & \text{if } y_i > 0 \\ 0 & \text{if } y_i = 0 \\ -1 & \text{if } y_i < 0 \end{cases} \tag{2.18}$$

Further, incorporating equation (2.16), we have soft thresholding operator as

$$prox_g(x_i) = max(|x_i| - \lambda) \times sgn(y_i) \tag{2.19}$$

Note in above equation *prox* operator is applied on each of the component of x.

2.1.3 ISTA and FISTA Algorithms

The ISTA (Iterative Shrinkage-Thresholding Algorithm)and FISTA (Fast Iterative Shrinkage-Thresholding Algorithm) [2] methods minimizes functions of the form

$$F(x) = f(x) + g(x) \tag{2.20}$$

where

- $x \in R^N$;
- $f : R^N \to R$: smooth, convex continuously differentiable with Lipschitz continuous gradient $L(f)$:

  $$\|\nabla f(x) - \nabla f(y)\|_2 \leq L(f) \|x - y\|_2 \text{ for every } x, y \in R^N$$

  where $L(f) > 0$ is the Lipschitz constant for $\nabla f(x)$; and
- $g : R^N \to R$: continuous, convex, but possibly non-smooth function.

In each iteration, these algorithms calculate

$$x_i = prox_g(x_{i-1} - t_i \nabla f(x)) \tag{2.21}$$

where, t_i is the step-size parameter. From definition of proximity operator, when $g(x) = 0$ then above equation (2.21) is given as

$$x_i = x_{i-1} - t_i \nabla f(x) \tag{2.22}$$

for minimizing the function with a Lipschitz continuous gradient $L(f)$. On the other hand if $f(x) = 0$, then equation (2.21) reduces to

$$x_i = prox_g(x_{i-1}) \tag{2.23}$$

for minimizing non-differentiable function. Such schemes are called as *forward-backward proximal splitting* ([9]). with forward scheme as gradient step using $f(x)$ and backward or implicit scheme using the function $g(x)$. Further *prox* operator for FISTA and ISTA is soft thresholding operator derived in section 3.3. There are two variants of each of these algorithms — One with a fixed step size t and the other with varying step size t_i or backtracking. The pseudo codes of these algorithms are discussed in next section.

2.1.4 Pseudo Codes

ISTA (fixed step length)

Define

$$Q_L(x,y) := g(x) + f(x) + (x-y)^T \nabla f(y) + \frac{1}{2} L \left\| x-y \right\|_2^2, \tag{2.24}$$

which is a quadratic approximation to F at y, and

$$prox_g(y) := \underset{x \in R^N}{\arg\min} \left\{ Q_L(x,y) \right\} = \underset{x \in R^N}{\arg\min} \left\{ g(x) + \frac{L}{2} \left\| x - \left(y - \frac{1}{L} \nabla f\left(y\right) \right) \right\|_2^2 \right\}, \tag{2.25}$$

which is the minimizer of Q_L given f, g, L and y.

Algorithm 1 ISTA with Constant Step size

Require: $x_0 \in \mathbb{R}^N, L := L(f)$.
  $k := 0$
  repeat
    $k := k + 1$
    $x_k := prox(x_{k-1})$
  until Convergence.

The Step size is constant in this algorithm which is $1/L$ where L is Lipschitz constant of $\nabla f\left(\left(\right) x\right)$.

One can think of another variant of above algorithm with varying step size called *backtracking algorithms*. In worst case scenario, backtracking algorithms can take exponential amount of time. But in-spite of this limitation, backtracking algorithms actually work well in practice. Further, in ISTA algorithm backtracking precludes the computation of Lipschitz constant in advance, which is an added advantage. Following algorithm is the backtracking version of ISTA.

ISTA (Backtracking)

FISTA (fixed step length)

Fast Iterative-Shrinkage Thresholding Algorithm (FISTA) is the faster version of ISTA algorithm which at the preserve the simplicity of ISTA. It is theoretical and practical rate of convergence is significantly better. FISTA chooses new point y_k and easy to compute.

FISTA (Backtracking)

The stopping criterion for the all the above algorithms are given as

$$\left| \frac{f(x_k) - f(x_{k-1})}{f(x_{k-1})} \right| < \epsilon \tag{2.26}$$

Algorithm 2 ISTA with Variable Step size/Backtracking

Require: $x_0 \in \mathbb{R}^N, L > 0, \eta > 1$.
  $k := 0$
  $L := L/\eta$
  repeat
    $k := k + 1$
    repeat
      $L := \eta L$
      $z := prox_g(x_{k-1})$
    until $F(z) \leq Q_L(z, x_{k-1})$
    $x_k := z$
  until Convergence.

Algorithm 3 FISTA with Constant Step size

Require: $x_0 \in \mathbb{R}^N, L := L(f)$.
  $k := 0$
  $t := 1$
  $y := x_0$
  repeat
    $k := k + 1$
    $x_k := prox_g(y)$
    $u := t$
    $t := \frac{1+\sqrt{1+4u^2}}{2}$
    $y := x_k + \left(\frac{u-1}{t}\right)(x_k - x_{k-1})$
  until Convergence.

Algorithm 4 FISTA with Variable Step size/Backtracking

Require: $x_0 \in \mathbb{R}^N, L > 0, \eta > 1$.
  $k := 0$
  $L := L/\eta$
  $t := 1$
  $y := x_0$
  repeat
    $k := k + 1$
    repeat
      $L := \eta L$
      $z := prox_g(y)$
    until $F(z) \leq Q_L(z, y)$
    $x_k := z$
    $u := t$
    $t := \frac{1+\sqrt{1+4u^2}}{2}$
    $y := x_k + \left(\frac{u-1}{t}\right)(x_k - x_{k-1})$
  until Convergence.

where ϵ is some arbitrarily small quantity typically 10^{-7}.

2.1.5 Essential Convergence Behaviour

ISTA. The Convergence rate of these algorithms has been investigated in [2] and [11]. Error at the kth iteration with respect to the true minimum x_* is bounded as $F(x_k) - F(x_*) \leq O(1/k)$. This worst-case complexity of both ISTA variants (Algorithms 1 and 2) is established by Theorem 3.1 in [2]. The sequence of function values $F(x_k)$ generated by ISTA is non-increasing.

FISTA. Error at the kth iteration with respect to the true minimum x_* is bounded as $F(x_k) - F(x_*) \leq O\left(1/(k+1)^2\right)$ The algorithms with convergence rate $o(1/k^2)$ is discussed in classical work of [15] and Theorem 4.4 in [2]. The sequence of function values $F(x_k)$ generated by FISTA need not necessarily be non-increasing.

2.1.6 A-Matrix for Radio Interferometry

From Van CittertZernike theorem we have A matrix as

$$A = \begin{bmatrix} e^{-2\pi i(u_1 l_1 + v_1 m_1)} & e^{-2\pi i(u_1 l_1 + v_1 m_2)} & \dots & e^{-2\pi i(u_1 l_n + v_1 m_n)} \\ e^{-2\pi i(u_2 l_1 + v_2 m_1)} & e^{-2\pi i(u_2 l_1 + v_2 m_2)} & \dots & e^{-2\pi i(u_2 l_n + v_2 m_n)} \\ e^{-2\pi i(u_3 l_1 + v_3 m_1)} & e^{-2\pi i(u_3 l_1 + v_3 m_2)} & \dots & e^{-2\pi i(u_3 l_n + v_3 m_n)} \\ \vdots & \vdots & \vdots & \vdots\vdots\vdots \\ e^{-2\pi i(u_{M-1} l_1 + v_{M-1} m_1)} & e^{-2\pi i(u_{M-1} l_1 + v_{M-1} m_2)} & \dots & e^{-2\pi i(u_{M-1} l_n + v_{M-1} m_n)} \\ e^{-2\pi i(u_M l_1 + v_M m_1)} & e^{-2\pi i(u_M l_1 + v_M m_2)} & \dots & e^{-2\pi i(u_M l_n + v_M m_n)} \end{bmatrix} \vdots \tag{2.27}$$

In short element at i^{th} row and j^{th} of A matrix is given by

$$(A)_{OJ} = e^{-2\pi i(u_i l_j + v_i m_j)} \tag{2.28}$$

2.1.7 Speed of Algorithms

All the variants of ISTA algorithms which we had discussed in Section 3.5 computes the gradient of differentiable term $f(x)$ in equation (**??**) in every iteration. The gradient of $f(x) = \| Ax - b \|_{\ell_2}^2$ at k^{th} iteration is given by

$$\bigtriangledown f(x_k) = 2(A^\dagger A x_k - A^\dagger b) \tag{2.29}$$

$A^\dagger A$ and $A^\dagger b$ which is constant for all iterations, needs to be computed in advance. Particularly, computation of $A^\dagger A$ dominates speed of these algorithms. If we consider a typical sensing matrix A of order $M \times N$ with complex entries then we require huge amount of memory space (for $M = 10^4$ and $N = 10^6 \approx 16000GB$) for storing full $A^\dagger A$, which is impractical. Further computation of $A^\dagger A$ by usual matrix multiplication will require roughly MN^2 complex computation.

One can always look for the symmetry's in $A^\dagger A$ in order to reduce the computational cost of the constructing it. For example, if $A^\dagger A$ is Hermitian then this reduces space complexity and time complexity of computing it by more than half if we store only unique elements in $A^\dagger A$.

Each of element of $A^\dagger A$ is given by

$$(A^\dagger A)_{ij} = \sum_{k=1}^{k=M} e^{2\pi i[u_k l_i - l_j) + v_k(m_i - m_j)]} \tag{2.30}$$

$A^\dagger A$ is real

Since u-v plane is symmetric Visibilities exist in conjugate pairs. Equation () can be written as

$$(A^\dagger A)_{ij} = e^{2\pi i[u_1 l_i - l_j) + v_1(m_i - m_j)]} + e^{-2\pi i[u_1 l_i - l_j) + v_1(m_i - m_j)]} + ... + e^{-2\pi i[u_t l_i - l_j) + v_t(m_i - m_j)]} \tag{2.31}$$

where $t = M/2$. In above equation each consecutive pairs forms the conjugate pairs of each other whose result is real. Further using Euler's formula $e^{i\theta} = cos(\theta) + isin(\theta)$ and using identity $e^{i\theta} + e^{-i\theta} = 2cos(\theta)$, above equation () reduced to

$$(A^\dagger A)_{ij} = \sum_{k=1}^{k=t} 2cos(2\pi[u_k l_i - l_j) + v_k(m_i - m_j)]) \tag{2.32}$$

Equation (2.32) shows that each element of $A^\dagger A$ is real.

$A^\dagger A$ is Symmetric

Equation () shows that, $(A^\dagger A)_{ij}$ and $(A^\dagger A)_{ji}$ forms complex conjugate pairs (i.e. $(A^\dagger A)_{ij} = (A^\dagger A)^*_{ji}$). Further, $A^\dagger A$ is real (proved in section), which proves that $A^\dagger A$ is symmetric[2].

$A^\dagger A$ is block toeplitz with toeplitz block

Further for regular $l - m$, in general it can be seen in color map of $A^\dagger A$ (2.1) that it is block toeplitz with each block as toeplitz block. These symmetries reduces the computational complexity of computing and space complexity of storing $A^\dagger A$ by a significant amount. Toeplitz matrices (refer appendix) has degree of freedom of $2N - 1$ for $N \times N$ matrix. So if we reconstructing image of $n \times n$ where n = $\sqrt{N}$, we have only $\frac{(2\sqrt{N}-1)(2\sqrt{N}-1)}{2}$ unique elements. Hence we can construct $A^\dagger A$ with approximately $4N$ element computation of $A^\dagger A$.

2.2 Implementation

2.2.1 MATLAB Implementation

The algorithms as mentioned in Section 3.5 require computations of large matrices. In particular, they require eigenvalue computations, matrix-matrix multiplications, matrix-vector multiplications and other algebraic transformations. Considering MATLAB's capabilities to perform the required computations with programming abstractions, it is considered as an appropriate tool for initial implementations. Also, MATLAB's Toolbox for Image Processing provides additional advantage for exploratory analysis of images from Radio Interferometry data sets.

[2]proved by Dr. Niruj Mohan Ramanujam

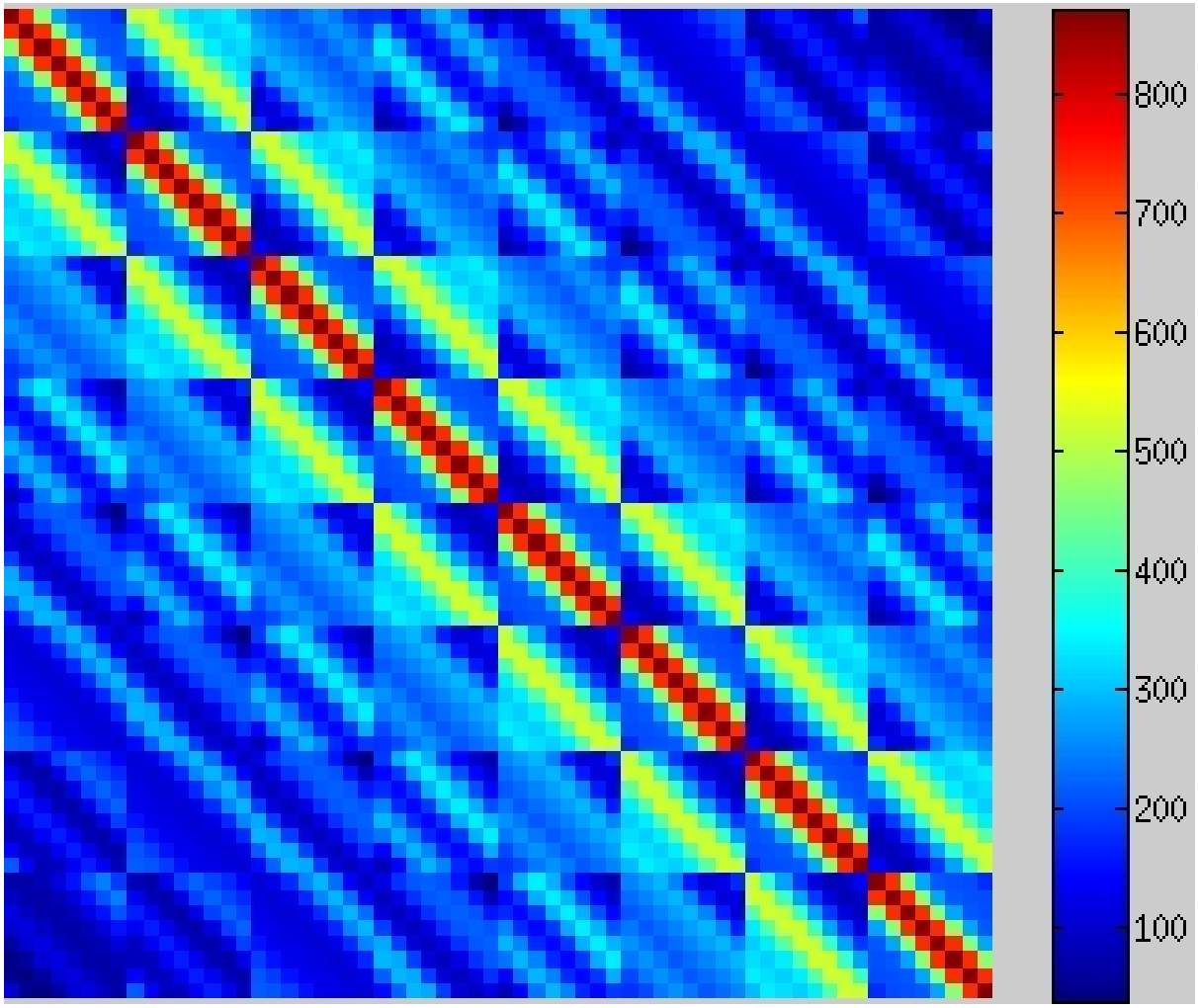


Figure 2.1: An example of $A^{\dagger}A$ for N=64

Further, algorithmic implementations were developed for all four variants of Iterative Shrinkage Thresholding Algorithms. Exploratory analysis via empirical validation were carried out by investigating the run-time complexities of the algorithms. CPU time taken were recorded, compared and evaluated. Performance benchmark trials were carried out by varying the number of iterations and the penalty parameter for determining their optimal values. Objective function values at each iteration were stored with a specified numerical accuracy. Robustness of the algorithmic implementation was validated by performing repeated trials with varying data sets each representing single point source, double point sources and grid point sources.

Input Data

Initially, data sets with characteristics similar to the original data sets of the FITS files were simulated using the AIPS tool for single point source and double point sources with varying fluxes. Each row of data set consist of a u, v, w co-ordinates, amplitude and phase of visibility measured.

Processing

Parameters required for processing the data sets are

- Lipschitz constant (L) for fixed step size variants which is maximum eigenvalue of $A^\dagger A$.
- Backtracking parameter (η). For both backtracking variants η value is taken as 1.125.
- Penalty parameter ($0 < \lambda < \infty$).
- Maximum Iteration Number (*maxiter*) which is one of the stopping criterion formats the algorithms.
- Tolerance limit explained by equation (2.26).

After parameters are decided, A matrix is computed using $u - v$ co-ordinates from data sets and regular $l - m$ grid using equation (2.27). Visibilities are computed in $a + ib$ format using phase and amplitude and were stored in b vector which is our measurement vector. Further, each variants of algorithms require $A^\dagger A$, $A^\dagger b$ which are computed as discussed in Section 3.7. Finally, following processed inputs are given to algorithms

- $A^\dagger A$
- $A^\dagger b$
- Lipschitz constant (L) (for fixed step size)
- Penalty parameter (λ)
- Backtracking Parameter (η) (for backtracking)
- Initial guess of solution which is zero vector
- Maximum iteration (*maxiter*)
- Tolerance limit (typically 10^{-7})

Output data

All four algorithms gives exactly the same solution vector with high numerical accuracy for single point source and double point source data set. By plotting the objective function values with each iterations it was clear that FISTA variants perform better than ISTA. Further, backtracking algorithms scores over fixed step size algorithms for radio interferometric data sets.

2.2.2 C Implementation

For the reasons of speed, efficiency, flexibility and to include Compressed Sensing formulation in GMRT pipeline, all algorithms are implemented in C. Radio Interferometry data sets are available in Flexible Image Transport System (FITS) formats (see appendix). We use Read-Write FITS (RWFITS) [3]. for reading the FITS file, in a specific structure which is further read for u, v, w and visibility data.

The structure of C-code is shown in Figure (2.2). The Data, after being read from FITS file are given to processing block which compute the inputs needed by algorithms as discussed in previous section. Since there is lot of symmetry in $A^{\dagger}A$ (discussed in Section 3.7) and A (due to symmetry in uv plane), there are options in the code for using these symmetries. Finally these computed matrices are given to the Compressed Sensing solvers, each algorithm gives same solution image with high precision accuracy. This reconstructed solution image is written back in to FITS file format using cfitsio (see appendix) routines.

For Basic linear algebra operations we have used intel's implementation of LAPACK and BLAS library, which performs matrix-matrix, matrix-vector, eigenvalue computations efficiency.

[3] written by Prof. Jayaram N. Chengalur

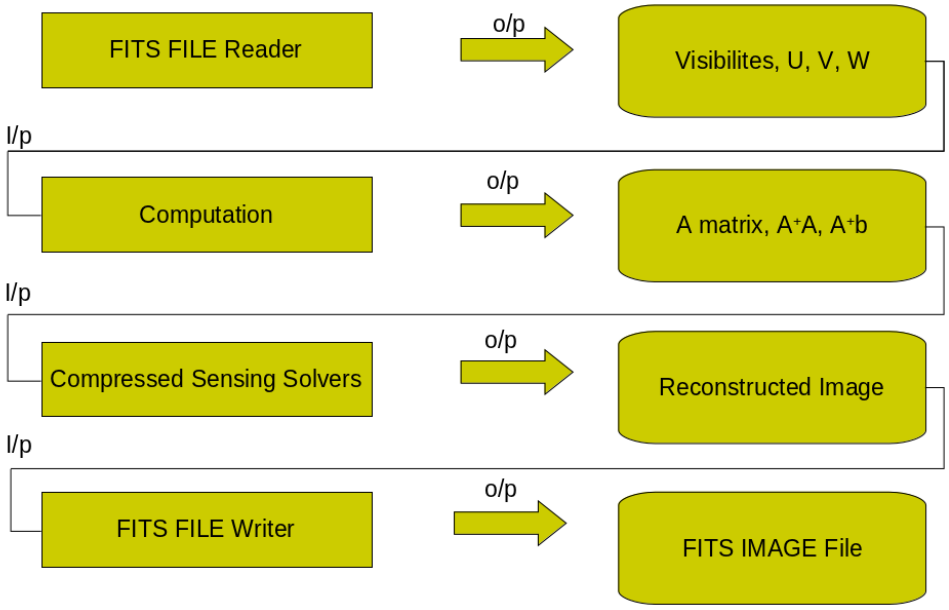


Figure 2.2: Flowchart of C Code

Chapter 3

Results and Discussion

3.1 A Toy Model

3.1.1 Formulation

To get the feel of Compressed Sensing, we applied its framework on various toy models. One such model is spike train model in which A matrix or sensing matrix of order $M \times N$ is random Gaussian matrix with normalized i.i.d entries. We constructed hypothetical example with known $M = 40$ random measurements (b vector) for a particular 40×100 A matrix and it is also known that x vector which we have to reconstruct, has four spikes. Figure (3.1 is an plot of original vector in which X-axis denotes the component number and Y axis represents values at each components. Indeed x is sparse as most of its components are either zero or approximately equal to zero. We had also added normally distributed Gaussian noise in the measurement vector as in real scenario every measurements comes with the noise.

3.1.2 Tools used

Next step was to apply compressed sensing or ℓ_1 minimization algorithms to this problem, we used MATLAB as a tool for this analysis and CVX, which is matlab program for disciplined convex programming, is used to solve 1.6 formulated for this problem. (for more information refer CVX user Guide).

3.1.3 Results and Discussion

Figure 3.1 shows plot of original vector with X-axis as component number and Y-axis as corresponding value at each components. Figure shows that plotted vector is sparse and has only four significant values or spikes.

Figure 3.2 shows plot of reconstructed vector (shown with dotted red line) and original vector (dotted blue line). With small variation reconstructed vector overlaps the original vector.

CVX internally use Interior point methods ([13]) for solving equation (1.6) for the given problem. This basic linear programming method has no performance issues while dealing with problems of such small scale but suffer for poor convergence rate otherwise. For this toy problem, interior point algorithm or CVX converges in 127 iterations. Further, CVX does not require any penalty parameter for minimization process.

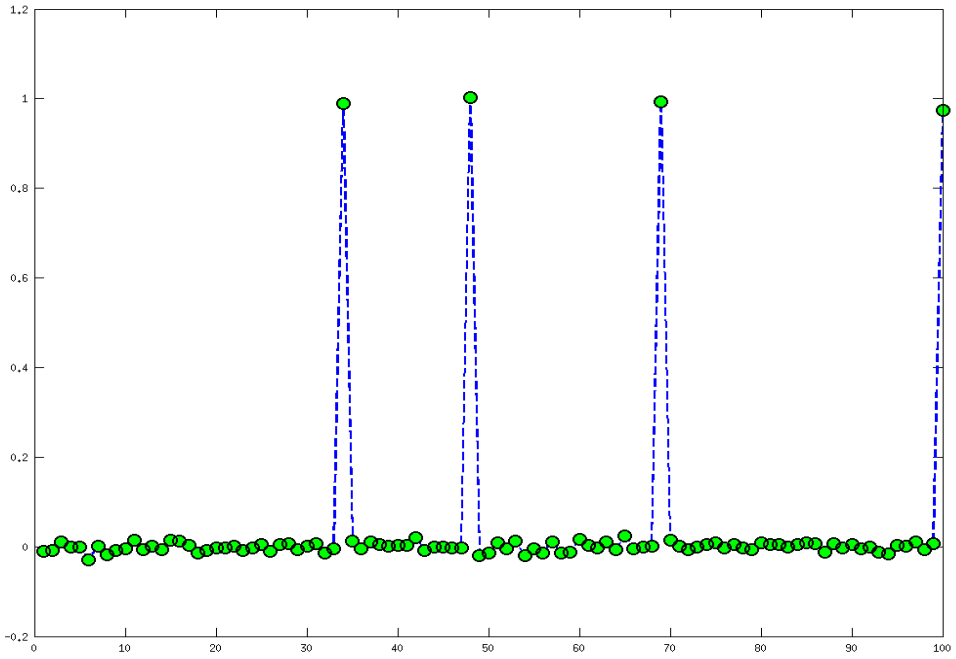


Figure 3.1: Original Vector

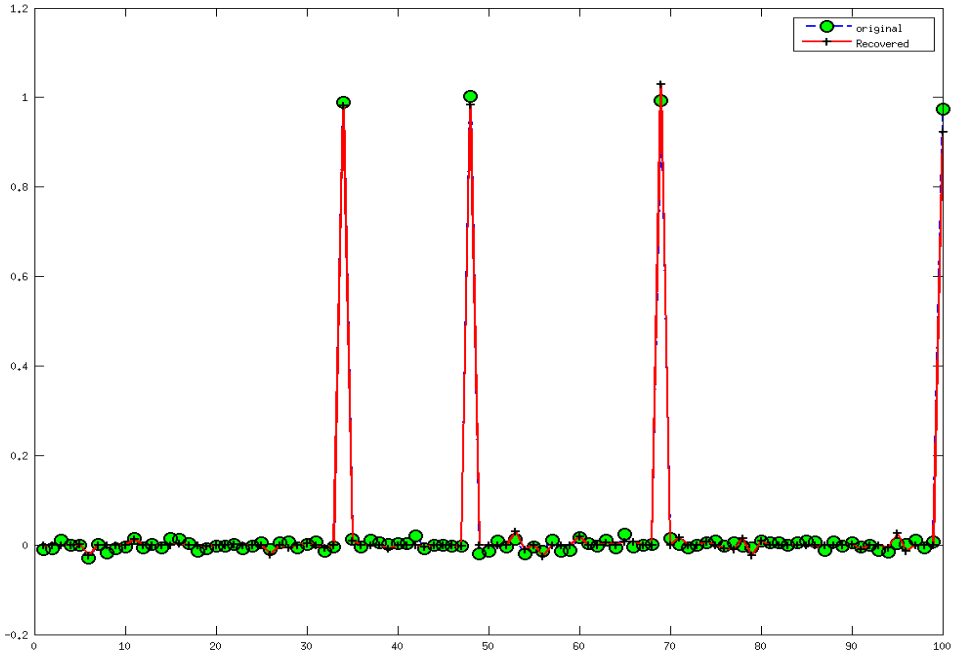


Figure 3.2: Reconstructed Vector

3.2 Reconstructing Single and Double Point Sources

3.2.1 Input

We have considered the images of single point source with illumination at center pixel of the image and double point sources with at centre and along the main diagonal. Both point sources have different fluxes and ratio of the fluxes at center to other is 0.8. We have used following input parameters for the algorithms.

3.2.2 Input Parameters

Following are the input parameters used for reconstruction.

- Lipschitz constant (L) (for fixed step size variants) = maximum eigenvalue of $A^{\dagger}A$.
- Backtracking parameter (η) = 1.125.
- Varied Penalty parameter from ($10^{-4} < \lambda < 10^{4}$).
- Maximum Iteration Number ($maxiter$) = 10000.
- Tolerance limit = 10^{-7}.

3.2.3 Output

Figure 3.3 with $\lambda = 1150$ shows reconstructed image of single point source of flux 0.4 at the center. We had produced such images for various values of λ to check the effect of λ on reconstruction.

Figure 3.4 with $\lambda = 1150$ shows reconstructed image of double point source at specific pixels. Recovered image has to ratio of fluxes (flux at center pixel : flux at other) is ~ 0.8.

Figure 3.5 shows the reconstructed image with different λ values for all solvers. $Fista_F$ denotes Fista with fixed step size and similarly others.

3.3 Discussion

3.3.1 Effect of Penalty Parameter

Equation (1.6) in which residuals are measured with ℓ_2 norm and regularization is done with ℓ_1 norm, seeks the solution that is close (in ℓ_2 norm) to the measured corrupted signal b and the one that is sparse. By varying the penalty parameter λ (aka regularization parameter) one can find out the optimal trade-off between $\| Ax - b \|_{\ell_2}$ (fidelity term)and $\| x \|_{\ell_1}$(sparsity inducing term).

Penalty parameter varies form 0 to ∞. In practical applications, for smaller value of λ in equation (1.6) put more emphasis on fidelity term, thereby neglecting sparsity in the solution. On the other hand, larger values of λ put more weight on sparsity inducing term and leads to sparsest solution which is zero vector. So finding out the optimal trade-off between these bi-criterion objectives requires optimal value of regularization parameter.

Figure 3.6 shows the reconstructed image of single point source with point source at the center. Due to very small $\lambda = 10^{-4}$, this leads to just the least solution which is not sparse.

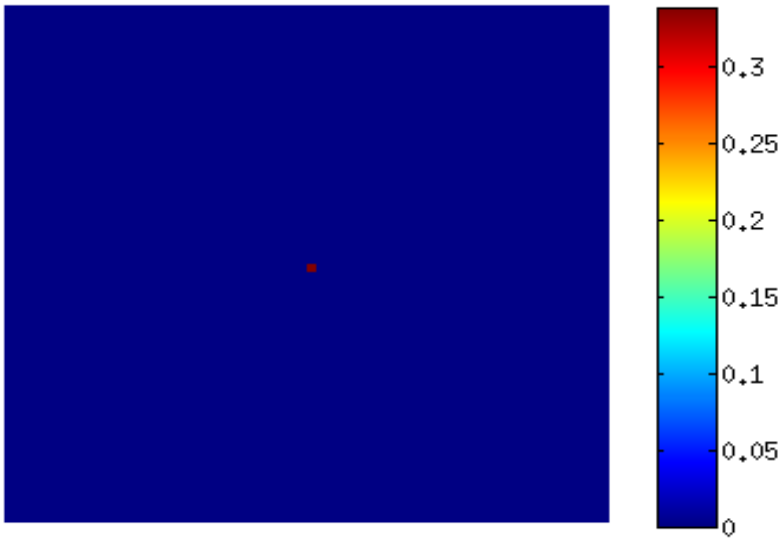


Figure 3.3: Reconstructed Image of Single point source with $\lambda = 1150$

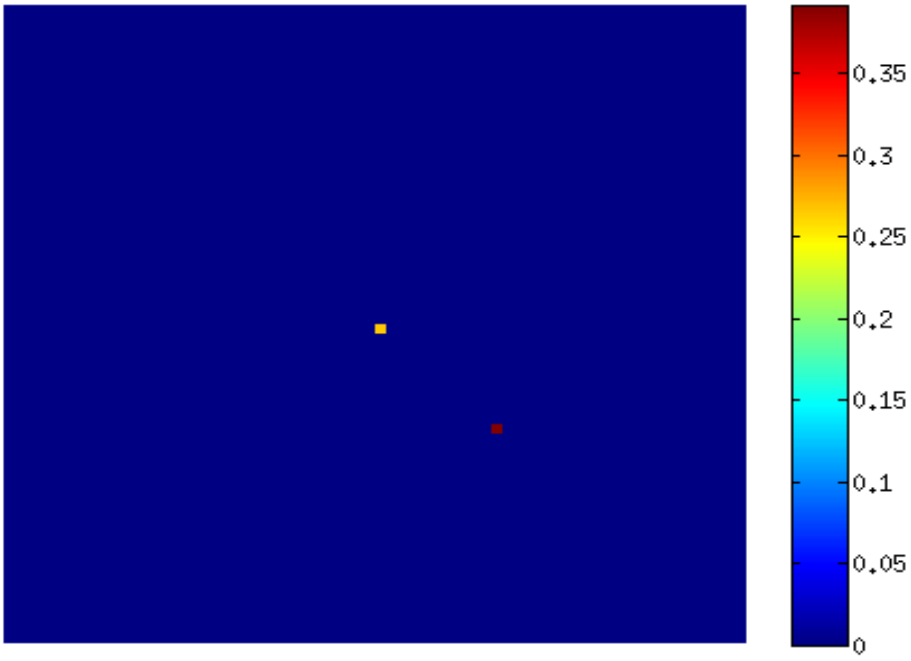


Figure 3.4: Reconstructed Image of Double point source with $\lambda = 1150$

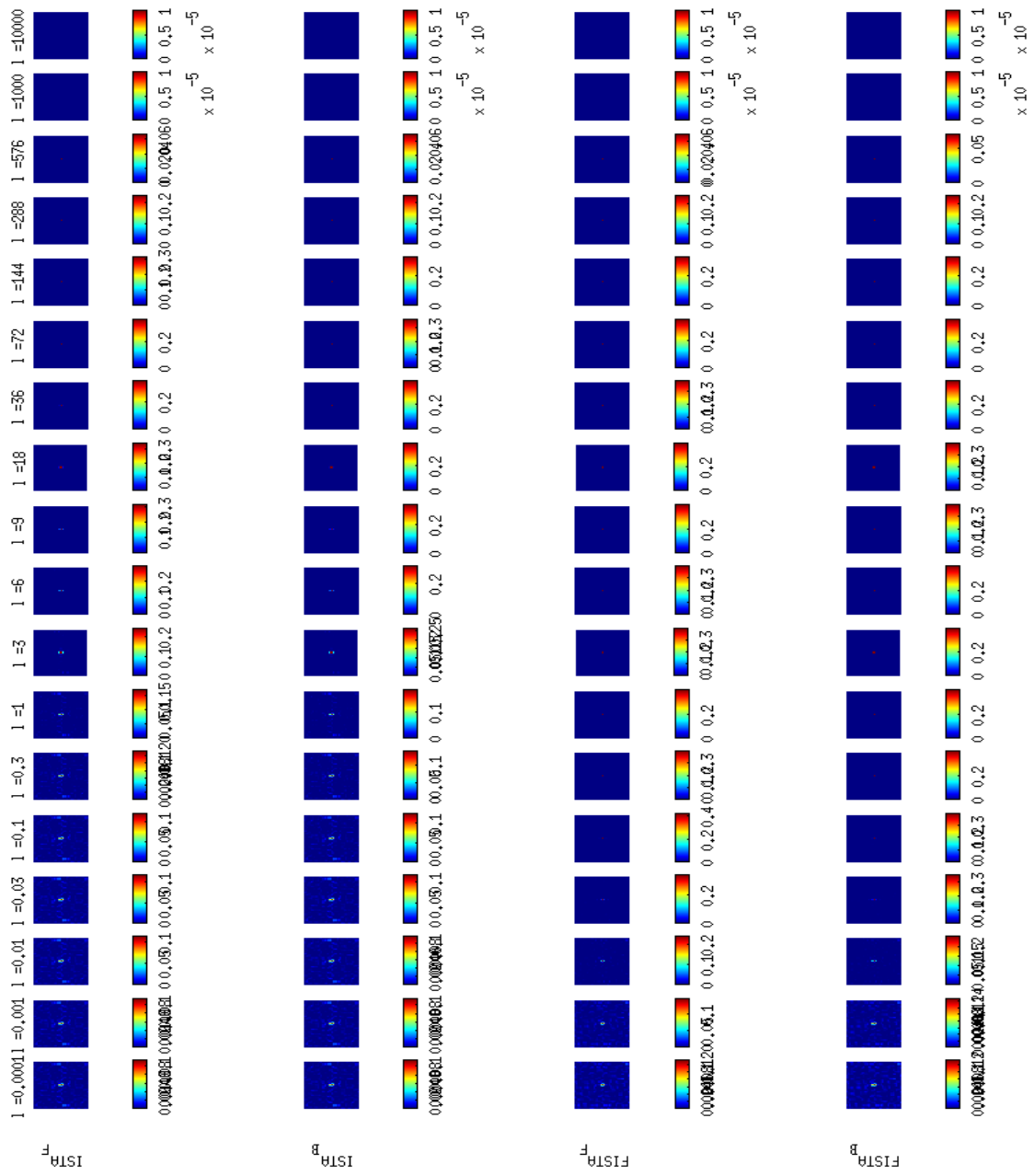


Figure 3.5: Reconstructed Image of Double point source with $\lambda = 1150$

Figure 3.7 shows the reconstructed image of same single point source with point source at the center. Large value $\lambda = 10^4$ gives zero solution as discussed.

For certain optimal $\lambda = 1150$ (found out by simulations), Figure shows reconstruction of single point source exactly.

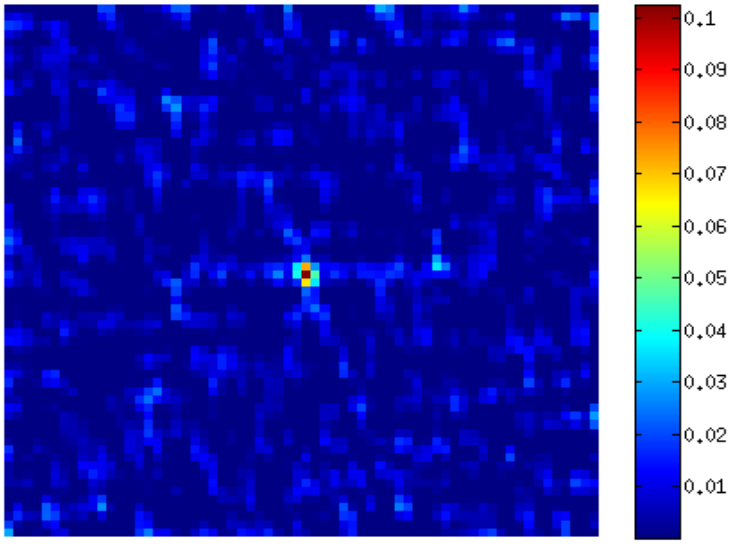


Figure 3.6: Reconstruction of single point source for $\lambda = 0.001$

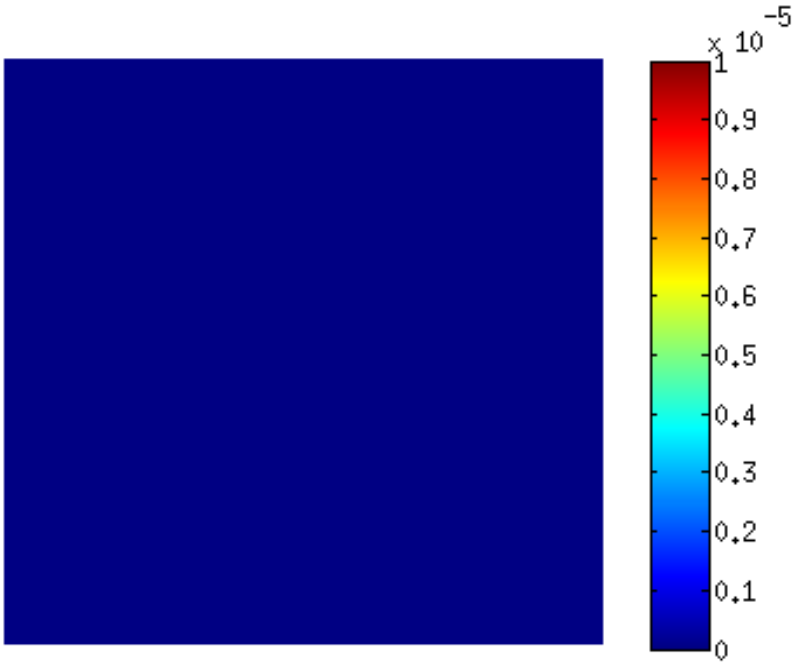


Figure 3.7: Reconstruction of single point source for $\lambda = 10000$

3.3.2 Convergence of algorithms

Figure 3.8, shows the convergence of algorithms with respect to iteration number. It is clear that FISTA shows better performance than ISTA algorithms which is also theoretically proved. Further backtracking variants of both algorithms seems to perform much better then their corresponding fixed step size variants. All four algorithms reaches the same optimal value and with same reconstructed image.

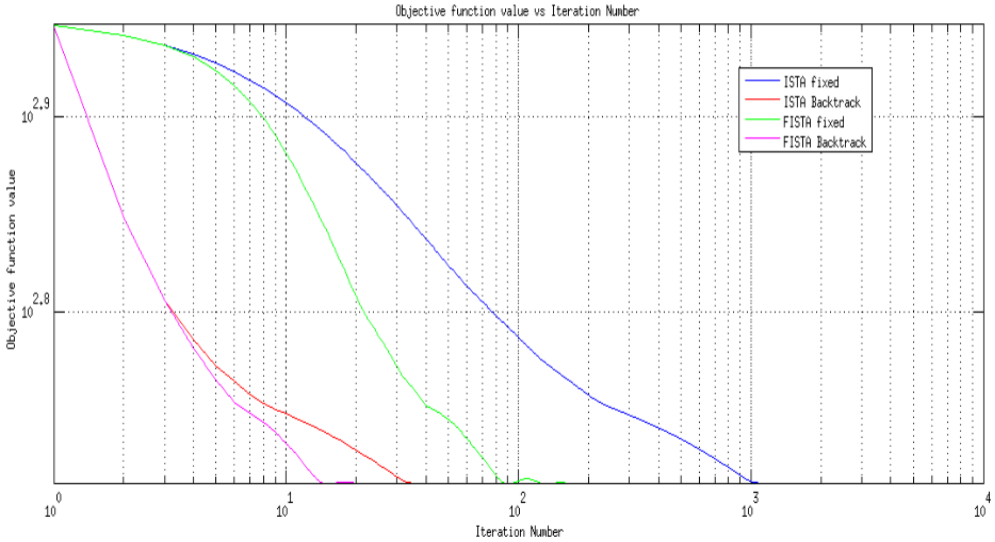


Figure 3.8: Convergence of Algorithms

3.3.3 Flux fidelity

Single point source and double point source reconstruction discussed in last chapter are done with varying fluxes of illuminated pixels of image. This is done to check flux fidelity with the penalty parameter. We have used various data set for single point source of fluxes 0.05, 0.1, 0.5, 1, 10 and 100 and double point sources we have used flux ratios as 0.1, 0.2, 0.4, 0.6 and 0.8. 3.9 shows one of the result for single point source of flux 0.4.

Figure 3.9 shows the plot peak value of the reconstructed image of single point source with flux value 0.4 (discussed in Section 6.3) with respect to penalty parameter varying from $10^{-}4$ to 10^4 (Reconstructed Images in this range of penalty parameter can be seen in Fig 6.3 in Chapter 6). It is clear from the plot that certain optimal range of penalty parameter flux of single point source recovered is 0.4.

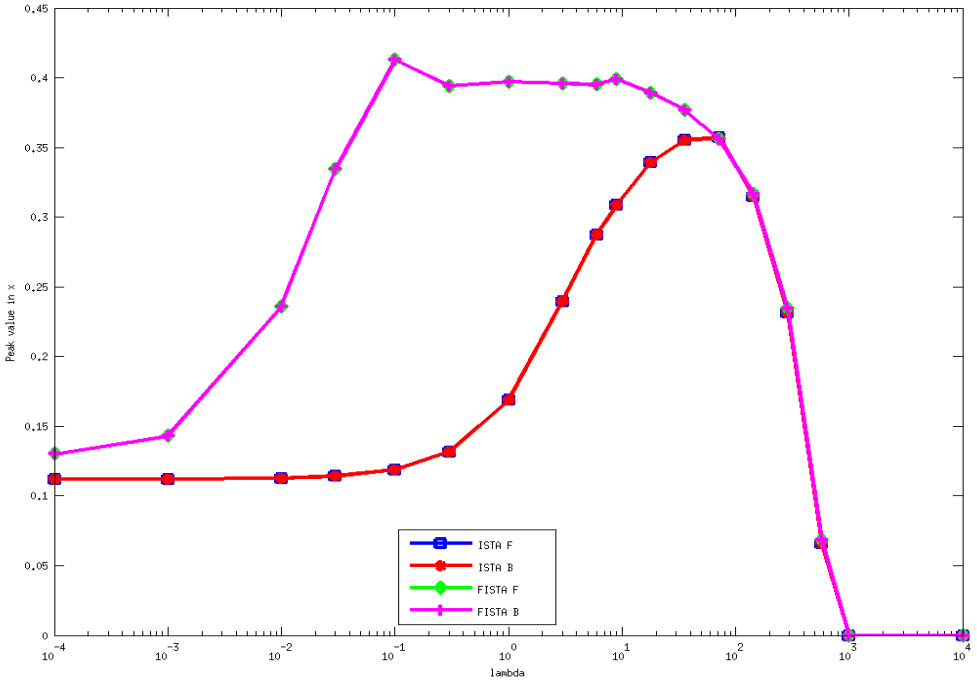


Figure 3.9: Peak Value vs Penalty Parameter

Chapter 4

Summary and Conclusion

We have evaluated a new frame work called Compressed Sensing for image reconstruction problem in Radio Interferometry. We have explored the class of algorithms called Proximal Gradient methods for efficient algorithms suitable for Radio-interferometric data. Finally we concluded that ISTA and FISTA algorithms along with their variants gives better performance and shows significant improvement in image visual quality over mainstream deconvolution algorithms like CLEAN. Further analysis, and work by other authors show that FISTA and its variants are capable of competing favourably with traditional image reconstruction methods in radio astronomy in normal scenarios, and will do much better in special cases (like 3D imaging, multi-frequency synthesis etc). We have, though a toy model, investigated and identified the core problems involved in such an exercise, and have gone on to implement a working code for applying compressed sensing to radio interferometric data from the GMRT. Along with collaborators at NCRA, we are currently involved in solving the remaining outstanding issues - namely, choosing an optimal penalty parameter, fidelity of flux reconstruction, and computational efficiency for large data sets. The code, which is a result of this project, will be further developed at NCRA into a full-fledged pipeline for the GMRT data, over the coming months. Possibilities for future work include the following:

1. To estimate optimal value of penalty parameter
2. To solve full Van CittertZernike equation.
3. To create a library of basis matrix.
4. Modify the algorithm for gridded Visibilities.
5. Modify the C code for all the above cases.

Bibliography

[1] D. Baron, M. Wakin, M. Duarte, S. Sarvotham, and R. Baraniuk. Distributed compressed sensing. *arxiv:0901.3403*, Jan 2009.

[2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Science*, 2:183–202, 2009.

[3] M. Born and E. Wolf. *Principles of Optics*. Cambridge University Press, Cambridge, UK, 1959.

[4] R. Bracewell. *Fourier Transform and Its Applications*. McGraw-Hill, New York, USA, 1999.

[5] E. Candes and J. Romberg. Sparsity and incoherence in compressive sampling. *Inverse Problems*, 23:969–985, 2007.

[6] E. Candes and T. Tao. Decoding by linear programming. *IEEE transactions on Information Theory*, 51(12):42034215, 2005.

[7] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE transactions on Information Theory*, 52(2):489–509, 2006.

[8] E. J. Candes and T. Tao. Near-optimal signal recovery from random projections: universal encoding strategies. *IEEE transactions on Information Theory*, 52(12):5406–5425, 2006.

[9] P. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, volume 49 of *Springer Optimization and Its Applications*, pages 185–212. Springer New York, 2011.

[10] D. Donoho. Compressed sensing. *IEEE transactions on Information Theory*, 52(4):1289–1306, 2006.

[11] O. Guler. On convergence of proximal point algorithm for convex optimization. *SIAM J. Control Optimization*, 10:403–419, 1991.

[12] J. Hogbom. Aperture synthesis on a non-regular distribution of interferometer baselines. *Astronomy and Astrophysics*, 15:417, 1974.

[13] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC '84, pages 302–311, New York, NY, USA, 1984. ACM.

[14] M. Lustig, D. Donoho, and J. Pauly. Rapid mr imaging with compressed sensing and randomly under-sampled 3dft trajectories. In *14th Annual Meeting ISMRMSeattle, WA*, 2006.

[15] Y. Nesterov. A method for solving convex programming problem with convergence rate. *Soviet Math. Dokl.*, 27:372–376, 1983.

[16] A. Rao. Interferometry and aperture synthesis. In J. N. Chengalur, Y. Gupta, and K. Dwarakanath, editors, *Low Frequency Radio Astronomy*, chapter 2. National Institute for Radio Astrophysics (NCRA-TIFR), Pune 411007, India, 3 edition, 2007.

[17] G. Swarup, S. Ananthkrishnan, V. Kapahi, A. Rao, C. Subrahmanya, and V. Kulkarni. Twenty-five years of radio astronomy at tifr. *Current Science*, 60:79–94, 1991.