


2017

## Signal processing with Fourier analysis, novel algorithms and applications

Alam Syed  
*University of Central Florida*

 Part of the [Electrical and Computer Engineering Commons](#)  
Find similar works at: <https://stars.library.ucf.edu/etd>  
University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of STARS. For more information, please contact [STARS@ucf.edu](mailto:STARS@ucf.edu).

---

### STARS Citation

Syed, Alam, "Signal processing with Fourier analysis, novel algorithms and applications" (2017). *Electronic Theses and Dissertations*. 5535.  
<https://stars.library.ucf.edu/etd/5535>

SIGNAL PROCESSING WITH FOURIER ANALYSIS, NOVEL ALGORITHMS AND  
APPLICATIONS

by

SYED ALAM ABBAS  
B.E. University of Mumbai, 2007

A dissertation submitted in partial fulfilment of the requirements  
for the degree of Doctor of Philosophy  
in the Department of Electrical and Computer Engineering  
in the College of Engineering and Computer Science  
at the University of Central Florida  
Orlando, Florida

Summer Term  
2017

Major Professor: Hassan Foroosh



© 2017 Syed Alam Abbas

The mind is not a vessel to be filled but a fire to be kindled.

*Plutarch, AD 46 – 120*

All models are wrong, but some are useful.

*George E. P. Box, 1919 – 2013*

## ABSTRACT

Fourier analysis is the study of the way general functions may be represented or approximated by sums of simpler trigonometric functions, also analogously known as sinusoidal modeling. The original idea of Fourier had a profound impact on mathematical analysis, physics and engineering because it diagonalizes time-invariant convolution operators. In the past signal processing was a topic that stayed almost exclusively in electrical engineering, where only the *experts* could cancel noise, compress and reconstruct signals. Nowadays it is almost ubiquitous, as everyone now deals with modern digital signals.

Medical imaging, wireless communications and power systems of the future will experience more data processing conditions and wider range of applications requirements than the systems of today. Such systems will require more powerful, efficient and flexible signal processing algorithms that are well designed to handle such needs. No matter how advanced our hardware technology becomes we will still need intelligent and efficient algorithms to address the growing demands in signal processing. In this thesis, we investigate novel techniques to solve a suite of four fundamental problems in signal processing that have a wide range of applications. The relevant equations, literature of signal processing applications, analysis and final numerical algorithms/methods to solve them using Fourier analysis are discussed for different applications in the electrical engineering / computer science. The first four chapters cover the following topics of central importance in the field of signal processing:

- Fast Phasor Estimation using Adaptive Signal Processing (Chapter 2)
- Frequency Estimation from Nonuniform Samples (Chapter 3)
- 2D Polar and 3D Spherical Polar Nonuniform Discrete Fourier Transform (Chapter 4)

- Robust 3D registration using Spherical Polar Discrete Fourier Transform and Spherical Harmonics (Chapter 5)

Even though each of these four methods discussed may seem completely disparate, the underlying motivation for more efficient processing by exploiting the Fourier domain signal structure remains the same. The main contribution of this thesis is the innovation in the analysis, synthesis, discretization of certain well known problems like phasor estimation, frequency estimation, computations of a particular non-uniform Fourier transform and signal registration on the transformed domain. We conduct propositions and evaluations of certain applications relevant algorithms such as, frequency estimation algorithm using non-uniform sampling, polar and spherical polar Fourier transform. The techniques proposed are also useful in the field of computer vision and medical imaging. From a practical perspective, the proposed algorithms are shown to improve the existing solutions in the respective fields where they are applied/evaluated. The formulation and final proposition is shown to have a variety of benefits. Future work with potentials in medical imaging, directional wavelets, volume rendering, video/3D object classifications, high dimensional registration are also discussed in the final chapter. Finally, in the spirit of reproducible research we release the implementation of these algorithms to the public using Github.

*Dedicated to my mother who loved me and took care of me  
when I was just a helpless and mostly useless piece of flesh...*

*...Blessings on the hand of women!  
Fathers, sons, and daughters cry,  
And the sacred song is mingled  
With the worship in the sky—  
Mingles where no tempest darkens,  
Rainbows evermore are hurled;  
For the hand that rocks the cradle  
Is the hand that rules the world.*

*—William Ross Wallace, 1819 – 1881*

## ACKNOWLEDGMENTS

We value virtue but do not discuss it. The honest bookkeeper, the faithful wife, the earnest scholar get little of our attention compared to the embezzler, the tramp, the cheat.

*John Steinbeck, Travels with Charley: In Search of America,*

1902 – 1968.

Writing this section is my favorite part of the entire thesis. Since this part is rewarding in its own right and not at all boring as writing the technically serious matter, as you recall and reminisce all the good, kind, generous people you met. Who helped you and made you to reach this point in your life. I wish I could include all of my teachers starting from kindergarten, if I could remember all, and even passing by strangers who did the small but kind human act of smiling at me in various places in this thesis, but alas the paper space unlike my memory space is rather limited and me missing someone who is deserving surely possible. So in the interest of time and space, I will try to acknowledge those I can.

My first thanks goes to the US taxpayer who unknowingly or knowingly paid for my extreme privilege so that as a graduate student I would hopefully do something useful for the humanity. I can't promise much, but I have tried to do good work. I was given the honor of the UCF Doctoral Fellowship for my PhD, the highest award endowed to graduate students, for that I am very grateful to the department and the university. As Columbia University professor Andrew Delbanco notes that the earliest English colleges were founded in the thirteenth century for scholars of divinity whose duties, “*included celebrating mass for the soul of the benefactor who had endowed the college and thereby spared them from menial work.*” This privilege helped me to do the kind of work that I did and along the way to meet with people whom I could have never met otherwise.

I would begin by thanking my advisors. Most PhD students are lucky if they can get one advisor but I have two co-advisors from two departments ! I would like to first and foremost thank Dr. Hassan Foroosh who has been very kind and generous, and who has been supporting me so graciously. He has been very generous and lenient towards me, that allowed me to explore and study very liberally the problems that I showed interest in. I had the pleasure and full freedom to chose my problems for this PhD and for that I am extremely grateful to him. At times when I would tend to make many mistakes in my paper drafts, he has helped by patiently working with me, even when there were some disagreements on the way, he has entertained my arguments and shown grace and kindness to me. And he has been nothing but helpful and supportive along the way.

Dr. Qiyu Sun is a *Shifu* to me. I met him about 4 years ago in his wavelet analysis class, and I have liked him since then. He was always very humble and respectful to all of his students. When I talked to him about my work in Signal Processing, he was very kind, receptive and was willing to help me in the research. Besides cleaning up my sloppy way to write mathematical equations and descriptions in the papers with his well-trained disciplined approach, he has taught me importance of writing a good scholarly work, and the meticulous care that must be taken for a high standard publication. He has shown me the art of writing a paper, now I tend to look at well-written papers very differently, almost with reverence, and he has emphasized the importance of references in me where due respect must be shown to all the previous scholars.

I will take this opportunity to thank my committee members for being kind, receptive and for their encouraging guidance, Drs. Alexzander Katsevich, Ulas Bagci, Nazanin Ravanhard and George Atia.

I would also like to thank my fellow-travelers, the members of the Computational Image Lab- Chaun Sun, Maryam Jaber, Vildan Atalay, Marjaneh Safaei, Amara Tariq, Ankit Sharma, Syed Muhamad Ali, Felix Fontan, Pooyan Balouchian, Shushant Kulkarni, Sangwoo Cho, Karan Daei-

Mojdehi, Amir Emad Marvasti, Ehsan Emad Marvasti, Sina Lotfian, Dustin Morley, Min Wang, Baoyuan Liu and the graduate students of other labs in our department - Dong Zhang, Sarfaraz Hussain, Waqas Sultani, Raj Dutta, Salman Khokkar, Andres Vargas, Ying Liu, Faraz Hussain, along with Zachary Millette, Ganesh Sundaresan and Gaby Gerritsen, without whom the journey of this long PhD would have felt more lonely.

Finally, I would also like to thank the regnant proponents, champions and providers of reproducible research such as Drs. D.L. Donoho, Emmanuel Candes, Victoria Stodden, Amir Averbuch, Micheal Elad, Daniel Potts, Peter Kostelec, Mohammad Mahoor, Hieko Bulow, etc. and thier collaborators without whom the kind of research that I have done would not have easily materialized. I dream of a scientific academic world in which not following reproducible research spirit will be frowned upon and in near future will even become unacceptable. Hence in the spirit of reproducible research, i.e. in order to be able to reproduce all the figures presented in this thesis (along with all the included papers), I have made my full source codes, supplementary materials and the tested datasets openly available to the public for academic purposes here: <https://github.com/syedalamabbas>

Lastly, I would thank my two sisters, Syed Saba Zareen and Syed Afreen Zahera, and my mother Syed Taj Fatma for the infinite care and love they gave me all of my known life.



## TABLE OF CONTENTS

LIST OF FIGURES . . . . .	xvi
LIST OF TABLES . . . . .	xxxiv
CHAPTER 1: INTRODUCTION . . . . .	1
CHAPTER 2: REAL TIME FAST PHASOR ESTIMATION . . . . .	8
2.1 Modeling Phasor Estimation as a Linear Filtering Problem . . . . .	13
2.1.1 Phasor estimation model with fundamental and harmonics . . . . .	14
2.1.2 Phasor estimation model with decaying DC component . . . . .	15
2.2 B-LMS with Conjugate Gradient Based Search Direction . . . . .	17
2.2.1 B-LMS algorithm . . . . .	17
2.2.2 Conjugate Gradient Method . . . . .	19
2.3 Proposed B-LMS with Conjugate Direction Search based Adaptive System . . . . .	21
2.3.1 The Algorithm . . . . .	22
2.3.2 Analysis of Convergence Characteristics . . . . .	23
2.4 Performance Evaluation . . . . .	27

2.4.1	Static Test . . . . .	27
2.4.2	Noise Test . . . . .	28
2.4.3	Dynamic Step and Ramp Change Test . . . . .	30
2.4.4	Transient Test . . . . .	33
2.5	Conclusions . . . . .	35
CHAPTER 3: FREQUENCY ESTIMATION FROM NONUNIFORM SAMPLES . . . . .		37
3.1	Introduction . . . . .	39
3.2	Single frequency estimation . . . . .	43
3.2.1	The proposed method . . . . .	43
3.2.2	Local Stability Analysis and Convergence . . . . .	47
3.2.3	Approximation error . . . . .	62
3.2.4	Statistical Characterization . . . . .	69
3.2.5	Extension . . . . .	73
3.3	Multiple Frequency Estimation . . . . .	77
3.3.1	Inputs with two well-separated frequencies . . . . .	83
3.3.2	Input with two very close frequencies, a pathological case . . . . .	85
3.3.3	Input with multiple harmonic frequencies . . . . .	87

3.3.4	Input with multiple frequencies . . . . .	89
3.4	Conclusions . . . . .	91
CHAPTER 4: NONUNIFORM POLAR AND SPHERICAL DISCRETE FOURIER TRANS-		
	FORM . . . . .	93
4.1	Introduction . . . . .	94
4.2	The Proposed Method- 2D . . . . .	98
4.2.1	Preliminaries: 1D Fractional Fourier Transform . . . . .	99
4.2.2	The Algorithm . . . . .	100
4.2.2.1	Exploiting Symmetry . . . . .	108
4.2.2.2	Computational Complexity . . . . .	108
4.2.3	Implementation Details . . . . .	109
4.2.4	Accuracy of the Proposed Fast PFT . . . . .	111
4.2.5	Comparison . . . . .	113
4.3	Digital Inverse Problematics . . . . .	116
4.3.1	Analysis . . . . .	117
4.3.2	Ill-Conditioning . . . . .	119
4.3.3	Fast Iterative Inversion . . . . .	122

4.3.4	Faster Inversion by Structured Matrix Computations . . . . .	127
4.4	The Proposed Method- 3D . . . . .	132
4.5	Contributions and Conclusions . . . . .	148
CHAPTER 5: ROBUST 3D REGISTRATION USING SPHERICAL POLAR FOURIER		
	TRANSFORM AND SPHERICAL HARMONICS . . . . .	150
5.1	Introduction . . . . .	151
5.2	Preliminaries: Affine Theorem in n-D . . . . .	156
5.2.1	n-dimensional Fourier Transform . . . . .	156
5.2.2	Affine Transformation in n-D Space . . . . .	157
5.3	The Proposed Method of Robust 3D Registration using SPFT and SOFT . . . . .	158
5.3.1	Class I: Isometries . . . . .	160
5.3.2	Class II: Similarity transformation . . . . .	168
5.3.3	Estimating The Translational Vector, $t$ . . . . .	169
5.3.4	Robust 3D Registration - The Algorithm . . . . .	170
5.3.4.1	Computational Complexity . . . . .	172
5.3.4.2	Higher Dimensions . . . . .	174
5.4	Experimental Validation and Performance Evaluation . . . . .	174

5.4.1	3D Rotation Estimation - Simple MRI Dataset . . . . .	175
5.4.2	Princeton Object Database, Large Set . . . . .	179
5.4.3	Robustness to Noisy Surface Points . . . . .	183
5.4.4	Missing Voxels or Holes or Partial Data . . . . .	185
5.4.5	Real World Challenging Dataset: Crashed Car Parking Disaster Scenario .	190
5.4.6	Comparison with Go-ICP Method for Partial Overlap . . . . .	197
5.5	Contributions and Conclusions . . . . .	201
CHAPTER 6: FUTURE WORK . . . . .		204
6.1	High-Dimensional Extension of Polar / Spherical DFT . . . . .	205
6.2	Fast and Exact Fourier Volume Rendering with Polar / Spherical DFT . . . . .	207
6.2.1	Algorithm I: Single Planar Projection . . . . .	210
6.2.1.1	Computational Complexity . . . . .	212
6.2.2	Algorithm II: Multiple Planar Projections . . . . .	213
6.2.3	Experiments . . . . .	215
6.3	Application in Medical Imaging of Polar / Spherical DFT - Sparse-View X-ray CT Image Reconstruction . . . . .	220
6.4	Applications in Directional Wavelets using the proposed Polar / Spherical DFT . .	222

6.5	Application in Finite Rate of Innovation . . . . .	226
6.6	Training Deep Networks on Preprocessed Fourier Domain Information . . . . .	227
CHAPTER 7: CONCLUSIONS . . . . .		229
LIST OF REFERENCES . . . . .		232

## LIST OF FIGURES

2.1	The adaptive system model for phasor estimation. The model is conventional, but the inputs and processing of weights is well designed to obtain phasor information directly using a lumped parameter model. . . . .	21
2.2	Static test results using a quarter cycle data window . . . . .	28
2.3	Dynamic response for the amplitude step . . . . .	31
2.4	Dynamic response for the phase step . . . . .	31
2.5	Dynamic response for the with amplitude step with prefiltering. . . . .	32
2.6	Dynamic response for the amplitude ramp change . . . . .	32
2.7	Simulated Current $I_A$ waveform for transient test . . . . .	34
3.1	Eigenvalue plot of the Jacobian system, $\xi = .1$ , $\gamma = .001$ in a frequency range, $\theta_0 = 2\pi f_0$ where $f_0 \in [30, 300]$ Hz in steps of 3 Hz. The eigenvalues $\lambda_3, \lambda_4$ remain strictly $< 0$ making the actual nonlinear system linearized locally at a point in the periodic orbit to be locally asymptotically stable. . . .	50
3.2	In the simulation, the signal (plotted in green) has amplitude $A = 1$ , frequency $f_0 := \theta_0/(2\pi) = 170\text{Hz}$ and phase $\phi = \pi/2$ , and the samples $z_n$ , corrupted by i.i.d. random noises $w_n$ with $\text{SNR} = 5\text{dB}$ , are taken on $T_n$ in (3.24) with $\Delta T = 1\text{ms}$ and $\tau_n = 0$ for uniform sampling (plotted in blue) and i.i.d. random $\tau_n \in [.5, 1.5]\Delta T$ for nonuniform sampling (plotted in red).	52

3.3	Flow trajectories and Poincare section of the discrete system (3.17). In the simulation, the sinusoid input $y$ in (3.5) has amplitude $A = 1$ , frequency $\theta_0 = 120\pi$ and phase $\phi = \pi/2$ , the sampling procedure (3.6) is taken uniformly at $T_n = 0.001n, 0 \leq n \leq 1000$ , seconds with additive i.i.d. random noise $w_n$ at level 5dB, the notch depth and adaption speed in (3.4) are given by $\xi = .15$ and $\gamma = .001$ , and the approximation order $m$ in (3.17) is 4. . . . .	53
3.4	Convergence of frequency estimation $\theta(T_n)/(2\pi)$ in the discrete ANF method (3.17) to the true frequencies $f_0 := \theta_0/(2\pi)$ . In the simulation, the sampling procedure and the input signal $y$ are the same as in Figure 3.2 except that $f_0 = 60\text{Hz}$ (left) and $f_0 = 170\text{Hz}$ (right). The notch depth, adaption speed and approximation order in (3.4) are the same as in Figure 3.3. . . . .	54
3.5	On the left is frequency estimation $\theta(T_n)/(2\pi)$ of the proposed ANF method for input signals $y$ with different amplitudes, while on the right is mean frequency error of the proposed ANF method for 500 independent trials for each noise level. In the simulation, the signal $y$ , the sampling procedure, and the input signal $y$ , the notch depth, adaption speed and approximation order are the same as in Fig. 3.2 and Fig. 3.3, except that $y$ has amplitudes $A = 1, 2, 4$ for the left figure, and $y$ has amplitude $A = 2$ and SNR of sampling noises $w_n$ varies from 0 to 50dB for the right figure. . . . .	55
3.6	Frequency tracking of the proposed ANF method (3.17). The sampling procedure (3.6) is taken uniformly at $T_n = .001n, 0 \leq n \leq 1000$ , seconds with additive noise $w_n$ having SNR= 20dB, the adaption speed, $\gamma = 0.01$ , and the notch depth, and the approximation order in (3.4) are the same as in Figure 3.3. . . . .	57



3.7	Comparison between the proposed ANF method (3.17) with $m = 4$ and Runge-Kutta method of order 4 on frequency estimation in uniform sampling setting. In this simulation, the input $y$ , the sampling procedure, the notch depth and adaption speed are the same as in Figures 3.2 and 3.3, except that the digital frequency $f_0 := \theta_0/(2\pi)$ of the input signal is 60Hz for the left figure and 170Hz for the right figure. . . . .	60
3.8	The sampling procedure (3.6) is taken uniformly at $T_n = .001n, 0 \leq n \leq 1200$ , seconds with additive noise $w_n$ having SNR= 18dB. The frequency changes abruptly after every .4 seconds. The parameters tuned are $\gamma = 0.005, \xi = .14$ and the ANF is initialize to frequency = 127 Hz. As it is clear that even in purely uniform settings the proposed method which is based on continuous dynamic system has similar speed and accuracy of frequency estimation as the purely discrete model based ANF proposed in [79]. . . . .	61
3.9	Frequency estimation of sinusoidal inputs $y = \sin(\theta_0 t + \pi/2)$ with frequencies $\theta_0/(2\pi)$ Hz via the algorithm (3.17) with $m = 2$ (squared magenta), $m = 3$ (circled green), $m = 4$ (diamond shaped blue) respectively. In the simulation, additive jittered sampling is same as shown in Figure 3.2. additive noise has SNR = 5dB, and the adaption speed $\gamma$ and notch depth $\xi$ are 0.001 and 0.15 respectively. . . . .	64
3.10	Block diagram for high frequency estimation . . . . .	65

3.11	Heterodyning ANF method for high-frequency estimation. In the simulation, the sinusoid input $y$ , the sampling procedure, the notch depth and adaption speed are the same as in Figure 3.3 except that the frequency $f_0 = \theta_0/(2\pi)$ of $y$ is 450Hz, modulator frequency $f_c = 490$ Hz, and frequency initialization for ANF is 55Hz. Presented on the right is a lowpass filter using the generic inbuilt MATLAB function <code>fdesign.lowpass</code> with specifications, $F_p = 50$ Hz (passband), $F_{st} = 75$ Hz (stopband), $A_p = 1$ , $A_{st} = 90$ (attenuation in dB) with specifications $F_p = 50$ Hz (passband), $F_{st} = 75$ Hz (stopband), $A_p = 1$ dB and $A_{st} = 90$ dB. . . . .	66
3.12	For jittered sampling $(n + r_n)/1000, 0 \leq n \leq 1000$ , seconds of sinusoidal inputs $y(t) = \sin(\theta_0 t + \pi/2)$ with frequencies $\theta_0/(2\pi)$ Hz, CRLB is about $6(\sigma/A)^2 a^{-2} N^{-1} = 0.6$ , where $r_n \in [-1/8, 1/8]$ are randomly selected and the signal-noise ratio $SNR := 10\log_{10}(\sigma/A)^2$ is 20. . . . .	69
3.13	Listed on the left is the mean value error between the estimated frequency $\theta/(2\pi)$ Hz in the algorithm (3.17) with $m = 4$ and the true frequency $\theta_0/(2\pi)$ Hz of the sinusoid input, while on the right is the common logarithm between the variance of the estimated frequency in the algorithm (3.17) and the Cramer-Rao low bound. In the algorithm (3.17), the jittered sampling set, SNR of the additive noise, the adaption speed and the notch depth are the same as in Figure 3.9. . . . .	70
3.14	Block diagram for discretization a dynamic system for frequency estimation. .	73

3.15	In this simulation, the input $y$ in (3.5) has amplitude $A = 1$ , frequency $\theta_0 = 120\pi$ and phase $\phi = \pi/3$ , the sampling procedure (3.6) is taken uniformly at $T_n = .001n, 0 \leq n \leq 1000$ , seconds without additive noise $w_n$ , the approximation order $m$ in (3.17) is 4 and initial frequency is about 66Hz or with 10% error. Also we set $\xi = .15$ and $\gamma = .001$ in (3.4) and (3.33), $b = .3, k_1 = k_2 = .5, \gamma = .6$ in (3.34) and $\lambda = .4, k_1 = .5, \gamma = .6$ in (3.35). . . . .	74
3.16	Frequency retrieval via discrete ANF method (3.17). In the simulation, the sampling procedure, the notch depth and adaption speed are the same as in Figures 3.2 and 3.3, two initial frequencies selected are $f_{c1} = 56\text{Hz}$ and $f_{c2} = 125\text{Hz}$ , and the input signal $y = A_1 \sin(2\pi f_1 t + \phi_1) + A_2 \sin(2\pi f_2 t + \phi_2)$ has phases $\phi_1 = \pi/2$ and $\phi_2 = \pi/7$ , frequencies $f_1 = 60\text{Hz}$ and $f_2 = 120\text{Hz}$ , and amplitudes $A_1 = 1, A_2 = 2$ (left) and $A_2 = 1/2$ (right). In this simulation, the dominated frequency of the input signals are 120 Hz for the left figure and 60 Hz for the right figure respectively. . . . .	78
3.17	Block diagram of cascade procedure for multiple frequency estimation . . . . .	80
3.18	Block diagram of discrete ANF method with prefiltering. . . . .	81

3.19	Frequency estimation via prefiltering ANF and cascade ANF methods. In this simulation, the sampling procedure, the notch depth and adaption speed, and two initial frequencies selected are the same as in Figure 3.16, and the signal $y$ in (3.37) has amplitudes $A_1 = 1$ and $A_2 = 2$ , phases $\phi_1 = \pi/2$ and $\phi_2 = \pi/7$ , and frequencies $f_1 = 60\text{Hz}$ and $f_2 = 120\text{Hz}$ . Plotted on the left is frequency estimation via cascade ANF method (in solid blue line) and prefiltering ANF method (in dashed red line), while plotted on the right is narrow band pass prefilter around two initial frequencies being used in the prefiltering ANF method. . . . .	84
3.20	Frequency retrieval of inputs with two close frequencies via the cascade ANF method. In this simulation, the input signal $y$ in (3.37) has frequencies $f_1 = 61\text{Hz}$ (dominant frequency for the right figure), $f_2 = 65\text{Hz}$ (dominant frequency for the left figure), phases $\phi_1 = \pi/2, \phi_2 = \pi/7$ , and amplitude $A_1 = A_2 = 1$ , the sampling procedure is the same as in Figure 3.19 except sampling time interval is $[0, 3]$ second, the adaption speed $\gamma = .00001$ (one hundredth of the one in Figure 3.19), and notch depth $\xi = .015$ (one tenth of the one in Figure 3.19), the first initial frequency is 63Hz (mean values of $f_1$ and $f_2$ ), and 67Hz (about 5% above the non-dominant frequency $f_2$ ). . . . .	86
3.21	Fundamental frequency tracking of an input signal with harmonic frequencies using a single ANF block. In the simulation, the sampling procedure, notch depth, adaption speed and initial frequency are the same as the ones in the Figure 3.6(b). . . . .	88

3.22	In the simulation, the sampling procedure, the notch depth and adaption speed are the same as in Figure 3.16. Presented on the left is the first four frequency estimation via the prefiltering ANF method (3.17) with the initial frequencies 70, 130, 180, 240Hz. On the right is the last three frequency estimation via discrete ANF method with heterodyning, where we use 270, 330, 450Hz as center frequencies of bandpass filters. . . . .	90
4.1	Listed on the top, bottom left and bottom right are a uniform grid, an $X$ axis scaled grid and an $XY$ axis scaled grid, respectively. In the simulation, the desired point (1.24, 2.56) for a $9 \times 9$ grid determines the exact scale factors to be used for the 1D FrFT along row or column. First is $X$ axis scaling $\alpha_1 = 1.24$ (expansion) and then along $Y$ axis, $\alpha_2 = .85333$ (shrinkage). These sequential scalings result in the desired point falling on coordinates (1, 3) of the transformed frequency grid. . . . .	101
4.2	Overlapping points of the polar grid with the rectangular scaled grid, each grid has exactly 4 radial lines that fall on it. $X$ axis scaling has horizontally oriented and $Y$ axis scaling has vertically oriented radial lines (both shown with ellipses). The four radial lines of level 1 solution are at angles $\Delta\theta$ , $180^\circ - \Delta\theta$ , $90^\circ - \Delta\theta$ , and $90^\circ + \Delta\theta$ , where $\Delta\theta = 18^\circ$ . . . . .	103
4.3	Top row (left) (a) Image with $X$ axis scaled grid, (right) (b) Image with $Y$ axis scaled grid. Bottom row (left) (c) Image with $XY$ axis scaled grid, and (right) (d) Image with $YX$ axis scaled grid, with appropriate scales respectively. The four radial lines of level 2 solution are at angles $\Delta\theta$ , $180^\circ - \Delta\theta$ , $90^\circ - \Delta\theta$ , and $90^\circ + \Delta\theta$ , where $\Delta\theta = 2 \times 18^\circ$ . This completes all the lines of the polar grid with $M = 10$ and $N = 8$ in (4.13). . . . .	107

4.4	Precision loss due to fast convolution, $E_{\max}$ vs. $N$ . As can be seen from the figure that although there is a steady loss of precision for larger array sizes, for moderately sized array the error is still of order $10^{-10}$ . This is the same precision loss with the polar grid computations using (4.6) since the fast convolution used for 1D FrFT is the building block of it. . . . .	112
4.5	In this simulation, the accuracy and computational complexity of NFFT are compared with the proposed algorithm for the polar grid. . . . .	114
4.6	Plotted are the execution times vs the input array sizes $N$ where the images are of size $(N+1)^2$ . The execution time using direct brute force computations is not plotted for $N > 150$ as it grows quickly and can take hours to complete.	116
4.7	Top row, left (i) Polar grid (pure), and right (ii) Polar grid with radial corners included. Middle row, left (iii) Polar grid 2x radial sampling, and right (iv) Polar grid 2x radial sampling with radial corners included. Bottom row, left (v) Polar grid with Cartesian corners (hybrid), and right (vi) Polar grid 2x radial sampling with Cartesian corners. Here, $N = 24$ , i.e. the spatial domain Cartesian grid is $25 \times 25$ and only the total number of angles $M = 52$ remains constant for all the grids. See Figure 4.8 for the condition number comparison of these grids. . . . .	121

4.8	Improving the ill-conditioning of the linear system in (4.21) by including the additional measurements. Without the inclusion of corners the condition number of $A_p$ even for really small sizes of grids ( $25 \times 25$ ) is very high (order of $10^{10}$ ). Inclusion of corners (see [103]) along radial direction improves the conditioning and so does the oversampling by 2x along radial lines. But the proposed hybrid solution (i.e. polar grid with Cartesian corners) greatly reduces the condition number of the system and combined with oversampling we get the best result which is closest to the ideal condition number of 1 for a perfect Cartesian grid. . . . .	122
4.9	(Top row) Corner compensated and padded reconstruction, where $M = 2(N+1)$ , and residual error is $= 8.7 \times 10^{-5}$ after just 12 iterations. We initialize the image with zeros for the first iteration, and we get a near perfect reconstruction at the image center. Observe that the sharp features have been destroyed by the use of filtered back projection method as it shows up in the error comparison. However, using the proposed method of exact discrete PFT we are able to retrieve those sharp features with the same sampling data. (Bottom row) Error comparison using the exact solution and the FBP approach by MATLAB implementation using functions radon and iradon. . . . .	126
4.10	BTTB to BCCB conversion, performing extension of real symmetric BTTB matrix using concept of embedding. . . . .	131

4.11	Spherical coordinate conventions (top), butterfly grid 2D (bottom left) and 3D (bottom right). Conventions followed here are shown. We measure the <i>azimuth</i> angle with respect to the $x$ -axis and <i>zenith</i> angle with respect to the $xy$ -plane. In 2D, the butterfly grid is a set of concentric rectangles and in 3D it is a set of concentric cuboids, both obtained by applying 1D FrFT on select axes. Only one axis is uniformly scaled with 1D FrFT, the other two axes are non-uniformly scaled. . . . .	135
4.12	3D transformations required from spatial domain rectangular coordinates of image size $(N + 1)^3$ to Fourier domain spherical coordinates grid of size $K \times M \times (N + 1)$ where the origin is included multiple times. . . . .	137
4.13	Spherical divisions required for fast and exact computation of Spherical FT. .	138
4.14	3D blocks at level $p = 1, q = 1$ (column left) and $p = 2, q = 2$ (column right) needed for fast computation of Spherical FT (from top to bottom) (a) XX-block (b) XZ-block (c) YY-block and (d) YZ-block. . . . .	140



- 4.15 A comparison for the execution times vs the input array size given by  $N$ . The brute force solution for  $N = 128$ ,  $(N + 1)^3$  volume is of complexity  $O((N + 1)^6)$ , and it takes about 4 hours on a single GeForce Titan X GPU, albeitto much longer on a CPU, but it just takes 32.35 minutes, using the proposed fast vectorized implementation on a CPU with reduced complexity. The best computational time is achieved with GPU accelerated library (CUDA) using ArrayFire [151] which is just, 2.7333 minutes on a single Titan X GPU. The proposed solution is orders of magnitude faster, while maintaining the similar level of accuracy as the brute force method. Faster solution for larger  $N$  can be achieved with additional parallelization as the proposed technique can be fully distributed and scales naturally with the computational resources. . . . . 146
- 4.16 Simple 3D reconstruction from a surface data (with given faces and vertices which is easier to visualize) transformed into volumetric data,  $N = 54$ , i.e. voxel data size is  $55 \times 55 \times 55$ , with  $M = N + 14$ , and  $K = N + 10$ . We obtain perfect reconstruction of surface data by simply thresholding the values for the reconstructed volume. This is simply due to the fact that the signal/volume is in binary form and well-padded with zeros i.e. the signal is concentrated in the center of the volume hence the corner artifacts are eliminated. . . . . 147

5.1	Given a 3D volume / matrix $f(r, c, d)$ we require transformation from spatial domain Cartesian coordinates of size $(N + 1)^3$ , $N$ being even, to the Fourier domain spherical coordinates grid $F(k, m, n)$ , of size $K \times M \times (N + 1)$ where the origin is included multiple times. In the above demonstration $N = 8$ i.e. this is volume of size $9^3$ , the spherical grid is $10 \times 10 \times 9$ points. $K = M = N + 2$ , hence the azimuth and zenith angular spacing $= 180/K = 18^\circ$ . . . . .	162
5.2	Multi-layered approach to the 3D rotation angles estimation. Unwrapping of the spherical magnitude equiangular spectrum data computed from SPFT into $2B \times 2B$ equispaced spherical images to be used for correlation using SOFT. In the above demonstration, $B = 8$ , $K = 2B$ , and $M = B$ . . . . .	164
5.3	Slices of the MRI images used to construct a full 3D volume, XY and YZ view of the volume, using the publicly available dataset [190]. . . . .	175
5.4	Slice projections and surface visualizations of the 3D volumes used for comparison with actual rotation matrix in (5.11) given by Euler angles, $(\alpha, \beta, \gamma) = (0.7069, 1.0996, 1.5080)$ . Voxel size for both objects is chosen arbitrarily as $63 \times 63 \times 63$ , and the equiangular bandwidth $B = 68$ as discussed in the Section 5.3.1. . . . .	176
5.5	Spherical images $H_n^1$ and $H_n^2$ as described in (5.14), original (top row) and rotated (bottom row) by $(\alpha, \beta, \gamma) = (0.7069, 1.0996, 1.5080)$ , at spectral radii (left to right) $n = 1, 2$ and $3$ respectively, obtained from (5.12) (SPFT) and correlation given by (5.16) (SOFT). Note the variations in the spectral signature as we go from smaller to larger radius which is equivalent to going from low to high frequencies. . . . .	177

5.6	Evolution of registration at different spectral radii, the estimated parameters at $n = 1$ are $(5.6364, 2.0443, 4.8510)$ . Although the solution as observed above is close with $\pi/2\pi$ ambiguity, it is not a perfect match within the angular resolution. The estimated parameters at $n = 2$ are $(2.4024, 1.1972, 1.7632)$ is closer to the ideal one. Finally, at radius $n = 3$ the estimated values are $(2.4024, 1.0972, 1.6632)$ , which is near perfect registration within the angular resolution defined by bandwidth $B$ with $\pi$ ambiguity for angle $\alpha$ i.e $\pi - \alpha = 0.7392$ which is $\cong$ to the actual orientation $\alpha = 0.7069$ radians. . . . .	178
5.7	Different objects with shapes and geometric complexities used in the PSB database test [191], (Clockwise from top left) (a) Beethoven (b) Bunny (c) Helicopter (d) Horse (e) Bed (e) Truck (f) Pot Plant (g) Branches. . . . .	180
5.8	The error estimates obtained for the PSB objects that are 786 in total, the known fixed 3D Euler angles are $(\alpha, \beta, \gamma) = (2.2777, 1.9018, 1.5080)$ and are used for rotation according to (5.11). The volumetric resolution for all the objects after voxelization is $63 \times 63 \times 63$ . . . . .	181
5.9	All the anomalies of the PSB test with the object IDs in $\{72, 230, 404, 410, 414, 596, 642, 725\}$ . (Top two rows) Original objects with large angular errors after registration. (Bottom two rows) Voxelized fixed and registered 3D objects (red and green respectively) after the registration using our solution. Note the full $\pi$ ambiguity for certain objects after registration and the rotationally symmetric shape of other objects. . . . .	182

5.10	Surface data transformation for rotation estimation, using <i>Dog</i> object from PSB database [191]. Voxel size for both objects is chosen as $89 \times 89 \times 89$ , and the equiangular bandwidth $B = 94$ . SNR = 10dB (measured). Actual angles used for rotation are, $(1.0505, 0.7159, 0.7226)$ . Estimated angles at spectral radius, $n = 2$ are $(\pi - 1.9050, 0.6266, \pi - 2.6737)$ . . . . .	184
5.11	Surface data transformation for rotation estimation, using <i>Beethoven</i> bust from PSB database [191]. Voxel size for both objects is chosen as $75 \times 75 \times 75$ , and the equiangular bandwidth $B = 80$ . The surface data is first voxelized then significant distortion is introduced at all levels in the form of missing surface data by setting certain values to zeros. . . . .	185
5.12	Rotation estimation using the proposed method at various spectral radii, which is fully automatic and robust. The top row represent the spherical images obtained at spectral radius 1 of the original and the noisy transformed surface data. The actual solution $\alpha, \beta$ and $\gamma$ as discussed in (5.11) is $= (2.2777, 1.8850, 1.5080)$ , and the estimated solution at radius 10 is near perfect match within the angular resolution $= (0.7854, 1.8826, 1.6632)$ again with $\pi$ ambiguity for angle $\alpha$ . . . . .	186
5.13	Rotation estimation with the same 3D volume and the transformation angles same as in the Figure 5.4. Registration at radius $= 6$ gives the estimated values $= (2.4024, 1.0972, 1.6632)$ , which is near perfect registration with $\pi$ ambiguity for angle $\alpha$ i.e $\pi - \alpha = 0.7392$ , which is approximately to the actual orientation $\alpha$ . This demonstrates that the solution is very robust to missing data/ partial overlap due to noise. . . . .	189

5.14	The Crashed Car Parking Dataset, (left to right) Front view, Close up of robot used and the close up of the site. For instance, the original volume Scan 12 (left in green) and Scan 13 (right in red) displayed in the form of surface data directly constructed from their respective volumes. . . . .	191
5.15	(Top row) Volume Scan 12 registered (left in green) and Scan 13 is fixed (right in red). Bottom close up of the combined registered scan directly using the 6DOF computation using the proposed method. . . . .	192
5.16	A direct attempt to register the two volumes using local intensity based (as seen on left) or location based registration ICP (as seen on right) fails. The local refinement registration ICP fails even after the initial estimate is computed using the global proposed method due to partial overlap. However, this demonstrates that roughly the global alignment can be computed using the proposed method and further local refinements are not useful in such scenarios.	193
5.17	Successful refinement using local registration is useful for scans with sufficient overlap e.g. Scans 13 and 14, (left) directly registered globally using proposed method, (center) further refined using image intensity based local registration and (right) refined using ICP. . . . .	194
5.18	Final integrated plot of 30 (top and bottom) the volume data and its corresponding point cloud set, as can be seen all the scans have been well aligned globally(roughly) from the (sub-sampled) volumetric data alone. Further local refinements can be done to improve the alignment by directly using the original point cloud data. . . . .	196

5.19	Failed Scans. (Top row) Scan 1,2 and 35. (Bottom row, left to right) failed registration for scan pair 1-2 (green-red), failed registration for scan pair 35-1(green-red). . . . .	197
5.20	(Top to bottom) (a) Original voxelized bunny object and its point cloud representation (b) Partial and rotated voxelized part of bunny and its corresponding point-set data (c) (viewed from a vantage point) Go-ICP with noisy and partial data, solution. The actual rotation Euler angles are $(1.2794, 1.1718, 0.7226)$ and the Go-ICP estimate $(0.2599, 1.2290, 4.0183)$ are close but not very accurate. The run time here is significant (in hours). . . . .	198
5.21	The final solution using the proposed method. The angles computed with the proposed solution are very close to the actual one $(1.2566, 1.1192, .7330)$ even when the overlap is just 38.76%. We use $B = 120$ , i.e. the two spherical grids (5.12) are of size $240 \times 120 \times 115$ , using an optimized GPU based computation on a commodity NVIDIA GTX 760 hardware. . . . .	199
5.22	(Left) Speed comparison (Go-ICP can take a lot of time-from hours to days) and, (Right) Accuracy. In the above experiments we fixed the actual rotation vector, $(\alpha, \beta, \gamma) = (1.2794, 1.1718, 0.7226)$ , and the two volumes were of size $67 \times 67 \times 67$ . . . . .	201
6.1	Direct and exact 2D slices computation of a full 3D medical volumetric data. Clockwise from top left: (a)3D Volume data such as CT scan (b) Top view or X-Y view of the 2D slice (c) 2D grids of complementary slices with respect to X-axis (d) Orthographic view of the slice which is a 2D equi-spaced normal grid. . . . .	210

6.2	Multiple projections using cylindrical Fourier transform of a full volumetric grid. . . . .	214
6.3	Single slice at a specified angle, used the complement of the image to enhance the effects of distortion. See the interpolated slice has poor reconstruction at the slice. . . . .	215
6.4	Ghosting artifacts and problem along sharp edges, while the exact reconstruction directly from the volume data using the proposed method is dense, clean and sharp. . . . .	216
6.5	Full volumetric data analysis using the proposed FVR method on <i>visible male</i> dataset, starting from top left at increasing angles of slices. . . . .	217
6.6	Full volumetric data analysis using the proposed FVR method on <i>skull</i> dataset, starting from top left at increasing angles of slices. . . . .	218
6.7	Full volumetric data analysis using the proposed FVR method on <i>Foot</i> dataset, starting from top left at increasing angles of slices. . . . .	219

6.8	(a) Original Phantom Image, (b) True polar sampling grid, (c) Reconstruction using NUFFT (Nonuniform fast Fourier transform) + ADTVM (Alternating directions total variation minimization), (d) Reconstruction using Exact Polar FFT (proposed method) + ADTVM, (e) Error in the reconstruction using NUFFT + ADTVM, and (f) Error in the reconstruction using Exact Polar FFT + ADTVM. Both the simulations run with the same input data and same number of iterations = 200. The same parameters for the ADTVM is also same for both. Error norm comparisons, $ I - \hat{I}_{NUFFT+ADTVM} _2^2 = 4.7260$ , and $ I - \hat{I}_{ExactPolar+ADTVM} _2^2 = 0.0789$ . The proposed method is superior in terms of accuracy while comparable in terms of speed (due to similar computational complexity) with the NUFFT approach. . . . .	221
-----	---	-----



## LIST OF TABLES

2.1	Convergence characteristics results . . . . .	26
2.2	Noise test results . . . . .	29
2.3	Test Results for decaying dc offset . . . . .	35
5.1	Table I: Effects of perturbation on the surface vertices. . . . .	184
5.2	Table II: Performance in case of holes/missing or partial data. . . . .	188

## CHAPTER 1: INTRODUCTION

The journey of a thousand miles begins with a single step.

*Lao Tzu, 605 – 531 BC*

Most people think it is the intellect which makes a great scientist.  
They are wrong : it is character.

*Albert Einstein, 1879 – 1955*

We are drowning in information, while starving for wisdom.  
The world henceforth will be run by synthesizers,  
people able to put together the right information at the right time,  
think critically about it, and make important choices wisely.  
*E. O. Wilson, Consilience: The Unity of Knowledge, 1998.*

The world as we see i.e., the physical and the discernible one can be reduced, at least from the perspective of an Engineer, into just signals and systems. Everything that you sense via sensory inputs is a signal - sight, sound, touch, and even thoughts - the internally generated entities can be considered as nothing but signals. After that sensation- measurements of signals, all the perception or interpretation that you do can be classified as your system. In a general sense, signals are used by the systems to transmit, receive, process and communicate between them. Sometimes it is directly generated by them, and if we can synthesize a signal, then we can readily analyze it. To understand signal analysis let us take the classical analogy of a piano. The pitch of a piano note varies linearly with the position of the key being struck; eventually, the sound one hears is the net

sum of all notes emitted. A skilled musician listening to the emitted poly-harmonic sound can hear which notes are playing and then also say which keys (frequency) were pressed and how hard (amplitude) ! The net sum of a polyphonic mixture that one hears is described as the synthesis of signals while what the accomplished musician is capable of discerning it all in his brain is called the analysis of signals. Analysis by synthesis is a common theme in at least man-made signals like audio signal processing, medical imaging and power systems especially when it comes to the direct applicability of a model of sinusoids or Fourier analysis. The sinusoidal signal once captured in its original continuous form or in discrete form can be readily analyzed if we understand and agree how to synthesize it. This is the foundation and the justification of the use of the sinusoidal model in the prevalent applications.

Could Mozart (1756 – 1791) analyze music signals down to the basic tone level in his mind? Well then he must have certainly known how to do Fourier (1768 – 1830) analysis in his brain decades before it was fully matured algorithm! Maybe the thoughts in which he would think is the exactly what the set of equations has codified into human readable form. Engineers, scientists, and mathematicians may not be that gifted, even if they were gifted no one among them would give them much credence - even if they really understood sounds at the basic atomic signal level. Arts maybe only about personal interpretations, but science is all about consensus building with justifiable and concrete evidence. However, with mathematics - the language of immortals, this effort of building accord can be resolved smoothly. Speaking in that language inherently guarantees a binary agreement or disagreement with any argument, there is no third way and nicely no room for open interpretations remains for the text of math. Engineers are trained to apply such mathematical tools to the real world problems.

We all need to understand signal processing - *sampling, transforming and filtering* [1], this statement pretty much sums up the motivation for this thesis. Digital signal processing first appeared in the 1950s was used at the service of analog signal processing [2]. It was simply designed to

simulate analog waveforms and transforms on a computer but not employed for direct processing. Nowadays the analog has taken a back seat as the digital processing/transforms have largely penetrated all forms of information processing - phones, cameras, televisions, radar, ultrasound, CT/MRI scan, etc. In this thesis, we are not concerned with the sampling part, but the transforming and the filtering / estimation aspects of the signal processing. We are also not fully restricted to the digital or discrete domain either since we show that there are some tools and accompanying advantages that we can readily borrow from such solutions that were originally developed and intended for the continuum domain. We are motivated to study these fundamental problems in signal processing since we believe that no matter how advanced our computational technology becomes and how dense our hardware technology grows, mankind will always be faced with challenging applications that are constrained by the limitation of physical resources, such as storage space, execution time and bandwidth.

The first half part of the thesis, that includes the Chapter 2 and 3 is all about studying the sinusoidal model and the estimation of its various parameters. Why is this problem relevant in the modern world? In the broadest sense where ever the sinusoidal model assumption is applicable, like power systems, audio signal processing, radar systems, ultrasounds, wireless communications etc, the work presented here can be considered relevant. The special sinusoidal signal that we call  $x_\omega$  has the complex exponential  $e^{in\omega}$  as its  $n^{th}$  sample. Its pure frequency is  $\omega$ . A more general complex exponential has a positive real amplitude  $A$  (not necessarily 1) and a real phase shift  $\phi$  (not necessarily 0):

$$x(t) = Ae^{i(\omega t + \phi)} \quad (1.1)$$

Using the famous formula,  $e^{i\theta} = \cos \theta + i \sin \theta$ , the real and imaginary part are a real cosine and

sine signals:

$$\begin{aligned}\operatorname{Re}Ae^{i(\omega t + \phi)} &= A \cos(\omega t + \phi) \\ \operatorname{Im}Ae^{i(\omega t + \phi)} &= A \sin(\omega t + \phi)\end{aligned}\tag{1.2}$$

In both chapters 2 and 3 we use real sine signal models for our study, since cosine signals are simply phase shifted versions of such models and complex signals can be easily reduced to such models by taking only their imaginary parts.

Chapter 2 is devoted to studying the transient phenomenon of power surges with an efficient and fast algorithm that estimates the amplitude and the phases changes of a power waveform. We use linear algebra techniques for solving this problem of fast phasor estimation and show how the proposed solution outperforms the existing technique in terms of speed and accuracy due to its simplicity. The synchrophasor measurement technology has exhibited great superiority in enhancing system situational awareness since it was developed and introduced into power system in the early eighties [3] - [5]. A novel approach for estimating the phasor parameters, namely magnitude and angle (not the frequency that we deal exclusively in the next chapter) in real time based on a simple linear algebra based conjugate gradient method is developed. This algorithm is capable of estimating the phasor parameters in less than a quarter cycle of an input signal which is the baseline of the state of the art algorithms. It features fast response and achieves high accuracy over a wide range of perturbations such as amplitude/phase step and ramp change, we even had some good preliminary results for chirps (non-linear change). To improve the dynamic performance when exposed to sinusoidal waveform distortions, such as modulation, an abrupt change in magnitude, etc, an adaptive approach for accurately estimating phasors while eliminating the effect of various transient disturbances on voltages and currents is proposed. Due to its fast response, reliability, high accuracy and the low computational burden it can be selected to meet certain desirable appli-

cations requirement. In addition, an approach for eliminating a decaying DC component, which has a significant impact on estimating phasors, is proposed.

Chapter 3 extends the sinusoidal analysis for estimating the frequencies of signals that are encountered in active noise and vibration control, wireless communications, audio, radar and sonar signal processing. In many such applications, it is desirable to eliminate or extract sine waves from observed data or to estimate their unknown frequencies. This problem is highly nonlinear, and sampling the signal used nonuniformly adds further complexity to it. It has been intensively studied in signal processing, instrumentation and measurements, and control theory, but most of the existing estimators are only designed to handle uniformly sampled data especially in the field of signal processing. Nonuniform sampling arises in many applications, such as frequency scanning interferometry, magnetic resonance imaging, computer tomography scans, Radon imaging and computer graphics. We propose to use a dynamic system existing in control systems literature and use it to solve a nonuniformly discrete frequency estimation problem. We show the usefulness of the proposed solution especially in the case when there is high noise and a set of signal frequencies are very closely spaced (an ill-conditioned problem). The resolvability of such closely spaced frequencies presents a major challenge, that we show to overcome using the proposed solution.

The second part of the thesis takes a different direction and studies a specialized version of Fourier transform widely used in imaging applications especially for medical imaging and then we study a particular application of 3D registration. In Chapter 4 we study the computation of a specialized nonuniform fast Fourier transform. In the new transform the domain information in spirit remains the same, however, the geometric arrangement of the Fourier coefficients are altered in such a way so as to achieve certain desirable properties of that transform. In the past, two decades and previously, tremendous amount of work has been done on this problem or a related problem of standard Radon transform connected by a sound Fourier-slice theorem. The mathematical problem underlies numerous applications such as medical imaging, radio astronomy, exploration seismol-

ogy, electron microscopy, nuclear magnetic resonance, optics, stress analysis, and non-destructive material testing and analysis, among others. As a result of this wide ranging applications, a number of scholars have taken different approaches to solve this problem. We begin with the literature survey that spans two decades, some very good scholars have been in the search for such a solution to this fundamental problem. The prevalent widely adopted solution - named as pseudo polar Fourier transform, provides an alternative to the true polar transform - a true analog of the continuous form like FFT does not exist in the previous literature. We propose a novel, fast and algebraically exact solution to the problem of computing true polar and spherical polar Fourier transform in 2D and 3D respectively. Our solution was obtained with certain insightful manipulations of the matrix transform entries and exploiting geometry and the symmetry of such regular grids to achieve a direct and efficient solution of the original problem. Further we discuss and design the practical inverse transforms and address the inherent challenges such as ill-conditioning, complexity, storage requirements. We propose an efficient algorithm based on linear algebra techniques and exploiting a special matrix structure that can be easily diagonalized using the FFT.

In Chapter 5 we investigate a real world useful application of the 3D spherical polar Fourier transform for 3D registration. Registration is a widely studied problem, it has applications in computer vision such as object pose alignment and for medical imaging of volumetric data. We focus here on a class of problems known as global frequency methods. A vast amount of literature is present for registration techniques in the Fourier transformed domain that are considered more robust. It is known that the estimation of rotation between two 3D becomes a hard problem when there are issues such as missing data, partial overlap, noisy data, and objects with intra-class similarity. We proposed a global frequency domain method for robust 3D registration. The algorithm uses multi-layered spectral data for correlation on spheres, hence exploiting 3D rotational information redundantly encoded across multiple spectral layers obtained from 3D spherical polar Fourier transform. This helps to counter the effects or noise/distortion, partial or overlap issues.

We then follow up with an important chapter describing the work that could be developed upon ours existing research in the future. Here we discuss various potential applications/ extensions of the proposed algorithms. Finally, we close with a chapter summarizing the conclusions and contributions of our research work.



## CHAPTER 2: REAL TIME FAST PHASOR ESTIMATION

Despite all the efforts of the prosecution,  
everybody could see that this man was not a “monster,”  
but it was difficult indeed not to suspect that he was a clown  
An extremely average person who relied on cliched defenses  
rather than thinking for himself and was motivated by  
professional promotion rather than ideology.

*Hannah Arendt, Eichmann in Jerusalem:*

*A Report on the Banality of Evil, 1963.*

If error is corrected whenever it is recognized as such,  
then the path of error is the path of truth.

*Hans Reichenbach, The rise of scientific philosophy, 1951.*

... Modern economic growth and the diffusion of knowledge have made it possible to avoid the Marxist apocalypse but have not modified the deep structures of capital and inequality - or in any case, not as much as one might have imagined in the optimistic decades following the World War II. When the rate of return on capital exceeds the rate of growth of output and income, as it did in the nineteenth century and seems quite likely to do again in the twenty-first, capitalism automatically generates arbitrary and unsustainable inequalities that radically undermine the meritocratic values on which democratic societies are based. There are nevertheless ways democracy can regain control over capitalism and ensure that the general interest takes precedence over private interests while preserving economic openness and avoiding protectionist and nationalist reactions. ... Indeed, the distribution of wealth is too important an issue to be left to economists, sociologists, historians, and philosophers. It is of interest to everyone, and that is a good thing. The concrete, physical reality of inequality is visible to the naked eye and naturally, inspires sharp but contradictory political judgments. Peasant and noble, worker and factory owner, waiter and banker: each has his or her own unique vantage point and sees important aspects of how other people live and what relations of power and domination exist between social groups, and these observations shape each person's judgment of what is and is not just. Hence there will always be a fundamentally subjective and psychological dimension to inequality, which inevitably gives rise to the political conflict that no purportedly scientific analysis can alleviate. Democracy will never be supplanted by a republic of experts - and that is a very good thing.

*Thomas Piketty, Capital in the Twenty-First Century, 2013.*

The phasor frequency, amplitude, and phase angle are critical variables used by many algorithms. How to rapidly and accurately estimate frequency and other phasor parameters is still a contemporary topic of research interest <sup>1</sup>. In this chapter we focus on the estimation of amplitude and phase only, the more complicated frequency estimation problem is dealt in detail in the next chapter <sup>2</sup>. Phasor magnitude and angle of various harmonic and interharmonic components of the power signal are widely used as critical variables and performance indices for power system applications such as protection, relaying and state monitoring. Here in we proposes a novel adaptive algorithm for estimating the phasor parameters in real time. It uses faster quasi-second order optimization technique to estimate amplitude and phase and does not require any matrix inversions. It features fast response, achieves high accuracy and involves lesser computational complexity than many other model ([6],[7],[8],[9]) and window based methods ([10], [11], [12]).

Power systems in many applications require real-time measurements of phasor magnitude and angle of the fundamental component and the harmonics present in the voltage and current signals of the power line. These are parameters of critical importance for the purpose of monitoring, control and protection (see [6]-[12]). Speedy and accurate estimations of such parameters are required for a proactive response under abnormal conditions in order to effectively monitor and preempt any escalation of system issues.

A variety of techniques for real-time estimation of phasors has been developed and evaluated in past two decades. They are either model based, least error squared (LES), recursive least square (RLS) [6], Kalman filtering [7] or other window based methods. They all use the stationary signal sinusoidal model. LES, RLS, and Kalman filters are more suitable for online processing since they

---

<sup>1</sup>Part of the material in this section is reprinted from “A New Fast Algorithm to Estimate Real-Time Phasors Using Adaptive Signal Processing” by Syed Alam Abbas, IEEE Trans. Power Delivery, vol. 28, no. 2, pp. 807-815, 04 January 2013 ©2013 IEEE, with permission from IEEE.

<sup>2</sup>See Chapter 3 where we discuss the frequency estimation problem and its complexity especially for nonuniformly sampled signal.

generate time trajectories of the evolved parameter but the complexity is significant and the matrix has to be fixed and accurate for the model to work. Any drift from the assumed nominal frequency will render the model highly inaccurate. Some artificial intelligence techniques such as genetic algorithms [8] and neural networks [9] have been used to achieve precise frequency and phasor estimations over a wide range with fast response. Although better performance can be achieved by these optimization techniques, the algorithmic implementation is more complex and it is intense in computations to be used for real time.

Window-based methods such as discrete Fourier transform (DFT), short time Fourier transform (STFT) and wavelets are also applied extensively for real time estimation of power system amplitude and phase parameters. DFT is desirable due to its low computational requirement and fast response. However, the implicit data window in the DFT approach requires a full cycle i.e. the fundamental period of the power signal waveform. To improve the performance of DFT-based approaches, some enhancements have been proposed. Adaptive methods based on the feedback loop by tuning the sample interval [10], adjusting the data window length [11], changing the nominal frequency used in DFT iteratively [12], correcting the gains of the orthogonal filters [13], and tuning the weighted factor [14] recursively, and compensation method to improve phasor estimation [15] are proposed. But due to the inherent limitations in such methods, at least one cycle of the analyzed signal is still required, which hardly meets the demand of high-speed response especially for protection schemes. A method using three consecutive samples of the instantaneous input signal is discussed in [16]. The noise and zero crossing issues may bring large errors to this method. STFT-based approach has limitations in its accuracy and still requires half a cycle to respond [17]. Recursive wavelet transform (RWT), which is faster and it can output phasor parameters in a quarter cycle, has been proposed recently [18]. In this method sub-optimal window-length and sampling rate may cause the weighting matrix to go singular. Also, it has the inherent limitation of having more computational requirements and higher sampling rate to achieve a reasonable ac-

curacy in short time. Hybrid algorithms have also been used to overcome individual limitations [19], but due to the trade-offs within the algorithms improved accuracy is usually achieved at the expense of increased complexity and additional delays.

Many techniques have been proposed to eliminate the impact of decaying dc components in phasor estimation. The conventional DFT algorithm achieves excellent performance when the signals contain only fundamental frequency and integer harmonic frequency components. Since, in most cases, the currents contain decaying dc components may introduce fairly large errors in the phasor estimations [20], [21]. A digital filter that features high-pass frequency response can help filter the high frequency noise. It performs well when its time constant matches the time constant of the exponentially decaying component. Theoretically, the decaying component can be completely removed from the original waveform once its parameters can be obtained. Based on this idea, [22] and [23] utilize additional samples to calculate the parameters of the decaying component. The study in [24] uses the simultaneous equations derived from the harmonics. The effect of dc components by DFT is eliminated by using the outputs of even-sample-set and odd-sample-set [25]. Reference [26] hybridizes the partial sum-based method and least-squares-based method to estimate the dc offset parameters. A new Fourier algorithm and three simplified algorithms based on Taylor expansion were proposed to eliminate the decaying component in [27]. In [28], the author estimates the parameters of the decaying component by using the phase-angle difference between voltage and current. This method requires both voltage and current inputs. As a result, it is not applicable to the current-based protection schemes. The recursive wavelet approach was introduced in protective relaying for a long time [29]- [31] . The improved model with single direction recursive equations is more suitable for the application to real-time signal processing [30]. The band energy of any center frequency can be extracted through recursive wavelet-transform (RWT) with moderately low computation burden [18].

In the proposed technique the phasor estimation is done using linear adaptive filtering techniques.

The phasor quantities to be determined are modeled as weights of the linear filter. This way the model can be easily adapted to drifts in the nominal frequency, when it is known, for each component separately, easily tracking its amplitude and phase changes. Then adaptive block least mean square algorithm (B-LMS) with optimally derived step sizes using conjugate gradient search directions is used, rather than gradient based directions, for minimizing the mean square error (MSE) or the cost function of the linear system. In simulations the performance of this new algorithm is compared with number of other popular published algorithms, both model based and window based.

This chapter is organized as follows: Section 2.1 describes the formulation of phasor estimation as a linear filtering problem. Section 2.2 gives the overview of the conjugate gradient technique and B-LMS algorithm. Section 2.3 has the proposed method along with convergence characteristics. In Section 2.4 the simulations results are presented in comparison with the other algorithms followed by conclusions in Section 2.5.

## 2.1 Modeling Phasor Estimation as a Linear Filtering Problem

*Linear Filter:* Consider a system with  $L$  inputs coming from sensors placed systematically in the environment that does weighted linear combination of inputs. Then the output of the system is given by:

$$y(n) = x_1(n)w_1(n) + x_2(n)w_2(n) + \dots + x_L(n)w_L(n). \quad (2.1)$$

Such a system is called linear filter and the weights can be adaptively estimated with some algorithm. In the above equation  $x$  denotes the signal,  $w$  as the linear weights and the output  $y$  is the linear weighted combination of inputs.

### 2.1.1 Phasor estimation model with fundamental and harmonics

Let us consider a discrete input signal (measurement)  $y$  that contains fundamental frequency with the sampling period  $\Delta T$ . Also without loss of generality we will consider a harmonic  $h$  of the fundamental frequency in the signal; where  $h$  need not be an integer.

$$\begin{aligned} y(n) &= V_{p0} \sin[2\pi f_0 n \Delta T + \theta_0] + V_{ph} \sin[2\pi f_h n \Delta T + \theta_h] \\ n &\in \mathbb{Z} \end{aligned} \quad (2.2)$$

where  $V_{p0}, f_0, \theta_0$  represents the amplitude, frequency and phase of the fundamental component respectively and denote  $V_{ph}, f_h, \theta_h$  the amplitude, frequency and phase of  $h^{th}$  harmonic or inter-harmonic if  $h$  is a real number such that  $f_h = h f_0$ , in the composite signal respectively.

Using the trigonometric expansion for (2.2), we get,

$$\begin{aligned} y(n) &= V_{p0} \sin[2\pi f_0 n \Delta T] \cos[\theta_0] + V_{p0} \cos[2\pi f_0 n \Delta T] \sin[\theta_0] \\ &\quad + V_{ph} \sin[2\pi f_h n \Delta T] \cos[\theta_h] + V_{ph} \cos[2\pi f_h n \Delta T] \sin[\theta_h] \end{aligned} \quad (2.3)$$

To simplify the equations with new parametrization,

$$\begin{aligned} w_1(n) &= V_{p0} \cos[\theta_0], w_2(n) = V_{p0} \sin[\theta_0] \\ w_3(n) &= V_{ph} \cos[\theta_h], w_4(n) = V_{ph} \sin[\theta_h] \\ x_1(n) &= \sin[2\pi f_0 n \Delta T], x_2(n) = \cos[2\pi f_0 n \Delta T] \\ x_3(n) &= \sin[2\pi f_h n \Delta T], x_4(n) = \cos[2\pi f_h n \Delta T] \end{aligned}$$

Substituting above values in (2.3), we obtain,

$$\begin{aligned} y(n) &= x_1(n)w_1(n) + x_2(n)w_2(n) \\ &\quad + x_3(n)w_3(n) + x_4(n)w_4(n) \end{aligned} \quad (2.4)$$

which is simply a linear weighted combination of inputs.

Once the weights of the filters are estimated the amplitude and phase of harmonics and interharmonics can be estimated with the following equations:

$$\begin{aligned} V_{p0} &= \sqrt{w_1(n)^2 + w_2(n)^2}, \\ \theta_0 &= \arctan \left[ \frac{w_2(n)}{w_1(n)} \right], \\ V_{ph} &= \sqrt{w_3(n)^2 + w_4(n)^2}, \\ \theta_h &= \arctan \left[ \frac{w_4(n)}{w_3(n)} \right]. \end{aligned}$$

Note that if the harmonic or the interharmonic we seek is absent in the signal, the corresponding weights describing it in the linear system will be close to zero.

### 2.1.2 Phasor estimation model with decaying DC component

Let us now consider a discrete input signal that contains only fundamental frequency with the sampling period  $\Delta T$  and a single decaying dc component.

$$\begin{aligned} y(n) &= V_{p0} \sin[2\pi f_0 n \Delta T + \theta_0] + V_{dc} e^{\frac{-n \Delta T}{\tau}} \\ n &\in \mathbb{Z} \end{aligned} \quad (2.5)$$



where  $V_{p0}, f_0, \theta_0$  represent the amplitude, frequency and phase of the fundamental component and  $V_{dc}, \tau$  are the amplitude and time constant of the decaying DC component present in the composite signal. Using Taylor's series expansion up to first order for the second quantity in (2.5), we have,

$$\begin{aligned} y(n) = & V_{p0} \sin[2\pi f_0 n \Delta T] \cos[\theta_0] + V_{p0} \cos[2\pi f_0 n \Delta T] \sin[\theta_0] \\ & + V_{dc} - \frac{V_{dc} \cdot n \Delta T}{\tau} \end{aligned} \quad (2.6)$$

We use the following notations at this point,

$$\begin{aligned} w_1(n) &= V_{p0} \cos[\theta_0], w_2(n) = V_{p0} \sin[\theta_0] \\ w_3(n) &= V_{dc}, w_4(n) = \frac{V_{dc}}{\tau} \\ x_1(n) &= \sin[2\pi f_0 n \Delta T], x_2(n) = \cos[2\pi f_0 n \Delta T] \\ x_3(n) &= 1, x_4(n) = -n \Delta T \end{aligned}$$

Substituting above values in (2.6) , we get,

$$y(n) = x_1(n)w_1(n) + x_2(n)w_2(n) + x_3(n)w_3(n) + x_4(n)w_4(n)$$

which is simply a linear weighted combination of inputs as shown in (2.1).

Once the weights of the filters are estimated the amplitude and phase of the fundamental harmonic and the amplitude and time constant of the decaying DC can be estimated with the following

equations:

$$\begin{aligned}
V_{p0} &= \sqrt{w_1(n)^2 + w_2(n)^2}, \\
\theta_0 &= \arctan \left[ \frac{w_2(n)}{w_1(n)} \right], \\
V_{dc} &= w_3(n), \\
\tau &= \frac{w_3(n)}{w_4(n)}.
\end{aligned}$$

## 2.2 B-LMS with Conjugate Gradient Based Search Direction

### 2.2.1 B-LMS algorithm

B-LMS or block least mean square error algorithm is extensively applied in numerous signal processing areas such as wireless communications, statistical, speech and biomedical signal processing [32]. The B-LMS algorithm provides a robust computational method for determining the optimum filter coefficients (i.e., weight vector  $w$ ). The algorithm is basically a recursive gradient (steepest-descent) method that finds the minimum of the MSE and thus yields the set of optimum filter coefficients. The instantaneous error signal at instant  $n$  processed in blocks of length  $L$  and weights of size  $K$  for a linear system is given by,

$$e_L(n) = y_L(n) - x_L^T(n)w(n),$$

where,

$$y_L(n) = [y(n), y(n-1) \dots y(n-L)],$$

$$x_L(n) = [X(n), X(n-1) \dots X(n-L)],$$

$$X(n) = [x_1(n), x_2(n-1) \dots x_K(n)],$$

$$w(n) = [w_1(n), w_2(n-1) \dots w_K(n)], \quad (2.7)$$

where  $y$ ,  $x$  and  $w$  are the output, input, and weight vectors, respectively. For simplicity, henceforth the subscript  $L$  will be dropped from notation. From the instantaneous error in (2.7) we can find the mean square error (MSE) signal or the cost function to minimize w.r.t to weights, which is given by,

$$E[w(n)] = \frac{1}{L} e^T(n) e(n) \quad (2.8)$$

Differentiating (2.8) w.r.t to weight vector  $w(n)$  yields,

$$\frac{\partial E[w(n)]}{\partial w(n)} = \frac{2}{L} e(n) \frac{\partial e(n)}{\partial w(n)}$$

Hence the negative gradient vector is given by,

$$-\frac{\partial E[w(n)]}{\partial w(n)} = \frac{2}{L} x^T(n) e(n) \quad (2.9)$$

Updating the filter coefficients is done proportional to negative of gradient according to the following equation:

$$\hat{w}(n+1) = \hat{w}(n) + \eta \frac{2}{L} x^T(n) e(n) \quad (2.10)$$

where  $\eta$  is the step size or learning parameter, and  $\hat{w}(n)$ , and  $\hat{w}(n+1)$  represent the initial and the new estimates of the weight vectors, respectively.

The feedback loop around the estimate of the weight vector,  $\hat{w}(n)$ , in the LMS algorithm acts like a *low pass filter*, passing the low frequency components of the error signal and attenuating its high frequency components. Also unlike other methods, the LMS algorithm doesn't require the knowledge of the statistics of the environment or prior information. Hence it is often called "stochastic gradient algorithm." In precise mathematical terms, the LMS algorithm is optimal in accordance with the  $H^\infty$  (or *minimax*) criterion [33].

### 2.2.2 Conjugate Gradient Method

We know that the instantaneous error vector is linear function of weights from (2.7) and hence the error surface (MSE) is quadratic function of weight vector from (2.8). The gradient computed from (2.9) is a first order partial differential w.r.t weights and if we replace it in (2.10) for each iteration, keeping the step size  $\eta$  constant, then the algorithm would be considered as a first order optimization steepest descent method. It may need a large number of iterations leading to slow convergence for some quadratic problems. The conjugate gradient method tries to overcome this issue by generating orthogonal search vectors of gradient. It belongs to a class of second order optimization methods collectively known as *conjugate-direction methods* [34]. Consider minimization of a quadratic function  $f(w)$  :

$$f(w) = \frac{1}{2}w^T A w - b^T w + c$$

where  $w$  is a  $L \times 1$  weight vector,  $A$  is a  $L \times L$  symmetric, positive definite matrix,  $b$  is a  $L \times 1$  vector, and  $c$  is a scalar. Minimization of the quadratic function  $f(w)$  is achieved by assigning to  $w$  the unique value,

$$w^* = A^{-1}b.$$

Therefore, minimizing  $f(w)$  and solving linear system of equations  $Aw^* = b$  are equivalent problems. Given a matrix  $A$ , the set of nonzero vectors  $s(1), s(2), \dots, s(L)$  is *A-conjugate* if:

$$s(n)^T A s(j) = 0,$$

$$\forall n, j \text{ s.t } n \neq j.$$

For a given set of vectors  $s(1), s(2), \dots, s(L)$ , the corresponding conjugate direction for the uncon-

strained minimization of the quadratic error function  $f(w)$  is defined by,

$$w(n+1) = w(n) + \eta(n)s(n), n \in \mathbb{Z}^+ \quad (2.11)$$

where  $s(n)$  is the gradient direction and  $\eta(n)$  is a scalar defined by,

$$f[w(n) + \eta(n)s(n)] = \min_{\eta} f[w(n) + \eta s(n)]$$

This is a one dimensional line search for fixed  $n$  (instant). The residual of the steepest descent direction is,  $r(n) = b - Aw(n)$ . Then to proceed to the next step we use a linear combination of  $r(n)$  and  $s(n-1)$ , as shown by the following equation,

$$s(n) = r(n) + \beta(n)s(n-1) n \in \mathbb{Z}^+ \quad (2.12)$$

where the scaling factor  $\beta(n)$  is given by *Polak-Ribiere formula* [35], a highly efficient preconditioner, that yields drastic improvements in convergence [36],

$$\beta(n) = \frac{r^T(n)[r(n) - r(n-1)]}{r^T(n-1)r(n-1)}. \quad (2.13)$$

Thus, conjugate gradient methods do not require any matrix inversions for solving linear system of equations and are faster than first order approximation methods. In combination with the B-LMS algorithm, they provide a very efficient way to solve quadratic problems.

### 2.3 Proposed B-LMS with Conjugate Direction Search based Adaptive System

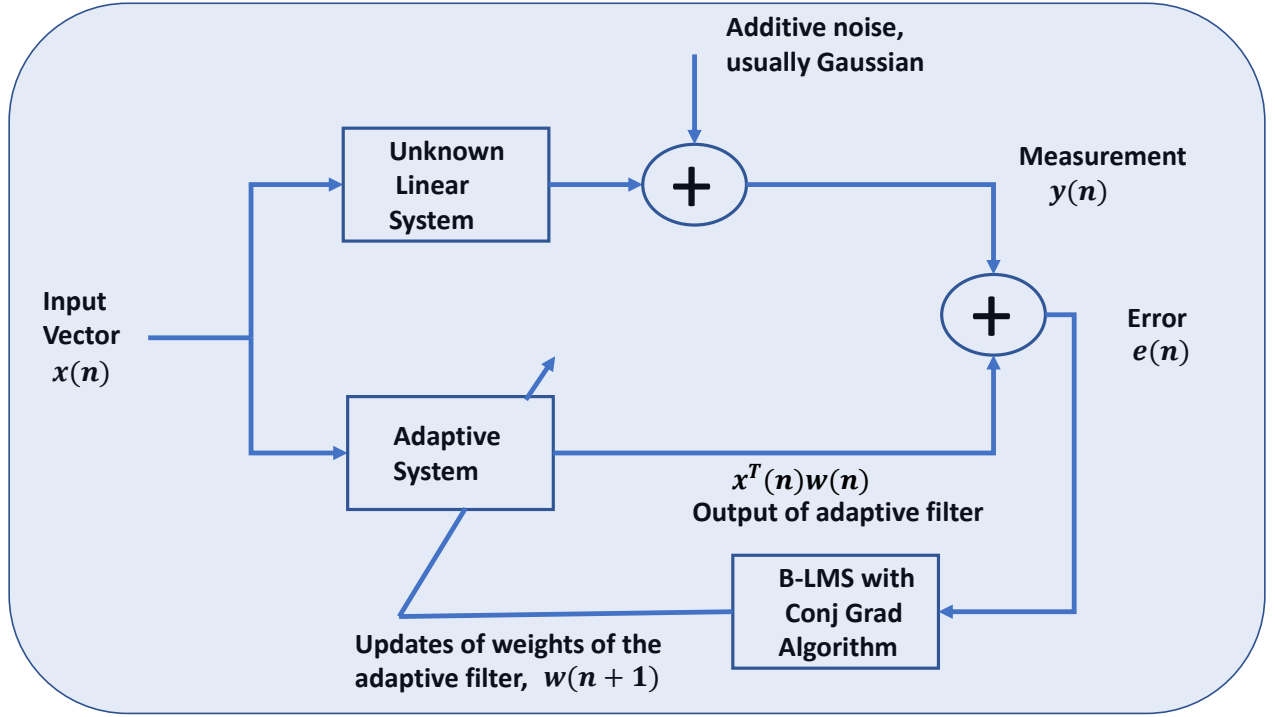


Figure 2.1: The adaptive system model for phasor estimation. The model is conventional, but the inputs and processing of weights is well designed to obtain phasor information directly using a lumped parameter model.

The block diagram for a linear adaptive filter to estimate phasors quantities is shown in Fig 2.1. The weights  $w$  are defined such that they trace the unknown parameters, amplitude and phase of the frequency components. Thus, the unknown linear system has weights reflecting the amplitude and phase of each component as shown by the formulations of phasor estimation problem in Section 2.2. The method presented here helps the adaptive system incrementally to match the output of the unknown system corrupted with inherent noise. The input vector  $x$  takes on the modeled values based on timestamp and frequencies while the desired vector  $y$  is the actual measurement vector

of the composite signal. The B-LMS with conjugate gradient based search algorithm tracks the unknown weight vector by matching the outputs of the adaptive and the unknown system through minimizing the error signal generated between the two in presence of noise. The noise need not be white Gaussian.

### 2.3.1 The Algorithm

The steps of the B-LMS algorithm with conjugate gradient based search for the linear adaptive system are presented below:

Step 1: For  $n \in \mathbb{Z}$  and for a block size of length  $L$ , and weight vector size of  $K$ , we have,

Input signal:  $x(n) = [X(n) X(n-1) \cdots X_L(n-L)]^T$

Measured signal:  $y(n) = [y(n) y(n-1) \cdots y_L(n-L)]^T$

Error signal:  $e(n) = y(n) - x^T(n)w(n)$

Initialize weights:  $\hat{w}(0) = [0 \ 0 \cdots 0]_{K \times 1}$

Gradient search direction:

$$s(0) = r(0) = -\frac{\partial E(\hat{w}(0))}{\partial \hat{w}(0)} = \frac{2}{L}e(0)x(0)$$

. Step 2: Find the optimal step size scalar parameter using the following expression (see [39] for details)<sup>3</sup>,

$$\eta(n) = \frac{e^T(n)x^T(n)s(n) + s^T(n)x(n)e(n)}{2s^T(n)x(n)x^T(n)s(n)}$$

. Step 3: Update the weight vector as follows,

$$\hat{w}(n+1) = \hat{w}(n) + \eta(n)s(n)$$

---

<sup>3</sup>It is trivial to derive the minimum  $\eta(n)$ , by simply setting the first order partial derivative of the MSE (2.8) to zero.

. Step 4: Find the new gradient direction

$$r(n+1) = -\frac{\partial E(\hat{w}(n+1))}{\partial \hat{w}(n+1)} = \frac{2}{L}e(n+1)x(n+1)$$

. Step 5: Use the *Polak-Ribiere formula* to calculate  $\beta(n+1)$ :

$$\beta(n+1) = \max\left\{\frac{r^T(n)[r(n) - r(n-1)]}{r^T(n-1)r(n-1)}, 0\right\}$$

. Step 6: Update the direction vector

$$s(n+1) = r(n+1) + \beta(n+1)s(n)$$

. Step 7: Set  $n = n + 1$ , go back to step 2.

If MSE  $E(w)$  is quadratic function of weights as shown in (2.8), the optimal value of weights ( $w^*$ ) will reach in at most  $K$  iterations where  $K$  is the size of the weight vector  $w$ .

### 2.3.2 Analysis of Convergence Characteristics

The conjugate gradient method is the most prominent iterative method for solving sparse systems of linear equations. An intuitive reason of generating vectors to be orthogonal is that it gives the minimum error. Thus conjugate vectors  $s(0), s(1) \cdots s(L-1)$  generated using (2.12) are linearly independent since they are orthogonal. They form a set of basis vectors that spans the vector space of  $w$ . So the optimum weight vector can be thought of as a linear combination of these vectors. Hence starting from an arbitrary point  $w(0)$ , the conjugate direction method is *guaranteed* to find the optimum  $w^*$  of the quadratic equation  $E(w) = 0$  in at most  $K$  iterations where  $K$  is size of weight vector  $w$ .



Since this method does not involve any matrix inversions, it is superior to those model based methods where choosing inappropriate parameters for the algorithm could render some intermediate calculation matrix singular. Also, since the search direction employed is conjugate gradient which is quasi-second order optimization technique, it is faster than first order (Jacobian) technique used in LES or RLS and comparably faster than other model based methods using non-linear curve fitting techniques such as Kalman filter with lesser computations.

Once the problem of estimation is modeled as a linear filtering problem, a combined generic model containing harmonics, interharmonics and decaying DC component could be used to compute any number of components of interest in the composite signal without any matrix inversions.

The use of such approach also provides flexibility of adapting the model to changes in frequency in real time when the drift is known. Each individual component of interest can be finely tuned rather than whole matrix changes as required in model based methods based on non-linear curve fitting like Kalman filtering or RLS.

To analyze the convergence characteristics, we define the window length as the cycle of the nominal frequency  $l_s$ , which is independent of the signal sampling frequency  $f_s$  defined as  $N$  times the nominal frequency  $f_0$  in Hertz. The variable  $l_s$  and  $f_s$  determine the number of samples  $N_s$  within a data/measurement block (i.e.,  $N_s = l_s \cdot f_s / f_0 = l_s / N$ ). There are many measures for analysis and classifications of power-quality disturbances [37]. But the standard one is the total vector error (TVE) that is used to measure the phasor accuracy [38]. Once the amplitude error  $\Delta V_h$  (in percentage of real value) and the phase error  $\Delta \theta_h$  (in degrees) are available, the expression is given by  $TVE_h = \sqrt{(\Delta V_h)^2 + (\Delta \theta_h / 0.573)^2}$ , where 0.573 is the arcsine of 1% in degrees for  $h^{th}$  frequency component.

The signal model used for convergence analysis is given below,

$$y(n) = \sum_{h=1}^5 \frac{V_0}{h} \sin[2\pi h f_0 n \Delta T + h\theta_0] + z(n)$$

$$n \in \mathbb{Z}$$

which is a discrete signal containing  $h^{th}$  order harmonics,  $\Delta T$  is the sampling period,  $n$  is the sample number.  $V_0, \theta_0, f_0$  are fixed amplitude, phase, and frequency of the fundamental harmonic.  $z(n)$  is zero mean white Gaussian noise.

In this sample test case, we let  $f_0 = 60$  Hz and harmonic components are up to 320 Hz. For high accuracy we choose a higher sampling rate, here  $N = 120$  samples per cycle (i.e., sampling frequency = 7.2 kHz). Analysis results are given in the Table 2.1, in which the window length used, convergence characteristic, response time, and TVE (maximum in 2 cycles of training) is compared for fundamental component only. The total number of components is 5, so the minimum number of samples = no of weights is  $5 \times 2 = 10$ . That is, two weights for each frequency component. The number of iterations used is equal to the number of weights for all runs. That is, the optimal number of iterations as per the B-LMS algorithm. We observe that as the data window is increased the accuracy improves with each iteration, finally at a half cycle window the response time saturates. Then only the time needed to respond is the time it takes to acquire the next sample. Also, only for a very small data window length ( $< 0.25$  cycle) the convergence is uncertain.

Table 2.1: Convergence characteristics results

Window $l_s$ =(No of samples in block)/(Total no of samples in 1 cycle)	Convergence of the algorithm	Response Time(ms) (after gathering one block of data)	TVE %
1/12	No	N/A	N/A
2/12	Yes	11.2	0.5441
3/12	Yes	9.8	0.3283
4/12	Yes	6.8	0.1660
5/12	Yes	4.61	0.1983
6/12	Yes	1.3889	0.0392
8/12	Yes	1.3889	0.0015
10/12	Yes	1.3889	0.0003
12/12 = 1	Yes	1.3889	0.0002

## 2.4 Performance Evaluation

In this section, the performance of the algorithm is fully evaluated under various test conditions covering static state, dynamic and transient state and the results are compared with conventional DFT methods and latest published techniques in [10] - [12], [14], [16], [18], [25] and [27]. All tests are performed with sampling rate  $N = 120$  samples per cycle (i.e., sampling frequency = 7.2 kHz) and a block size varies depending on number of weights to be computed and number of iteration used, if any, is equal to the number of weights. The higher sampling rate is useful for high accuracy.

### 2.4.1 Static Test

A signal model containing harmonics and 0.1% (signal-to-noise ratio  $SNR = 60$  dB) white noise is assumed, where represents the zero-mean white Gaussian noise. Let  $V_0 = 1$  and  $\theta_0 = 5^\circ$ . The total vector error of first component is plotted as a function of time. Comparing to the DFT-based methods in [10]- [12] that require a full cycle(16.67 ms), the algorithm can output phasor parameters only in about less than 4ms. Here we used a block size of 30 (i.e..25 cycle).The result is shown in the Fig 2.2.

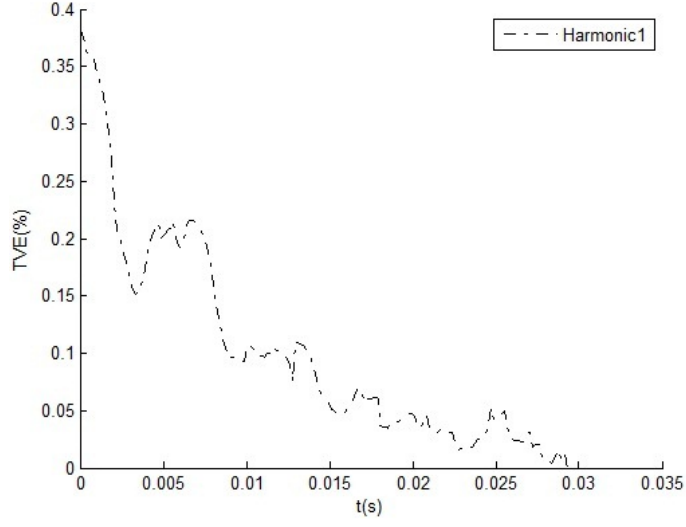


Figure 2.2: Static test results using a quarter cycle data window

#### 2.4.2 Noise Test

The inherent noise rejection capability of the algorithm is investigated by this test. The signal model for the static test is used. Each run is for 2 cycles. We observe that even under worst case condition of 30 dB signal to noise ratio, if the data window used is equal to one cycle the algorithm can give still a very low TVE. For different noise levels, the TVE for the B-LMS and RWT is evaluated and compared as shown in Table 2.2.

Table 2.2: Noise test results

Noise Level (SNR)	Data block length ( $l_s$ )	B-LMS TVE %	RWT TVE %
0.1%(60dB)	0.25	0.2295	0.36
	0.5	0.0197	0.12
	1.0	0.00007	0.036
0.32%(50dB)	0.25	0.3582	0.94
	0.5	0.1137	0.40
	1.0	0.00057	0.064
1%(40dB)	0.25	1.5025	1.60
	0.5	0.3984	1.08
	1.0	0.0023	0.26
3.2%(30dB)	0.25	8.8573	2.73
	0.5	1.3426	1.59
	1.0	0.0086	0.68

### 2.4.3 *Dynamic Step and Ramp Change Test*

To evaluate the dynamic response when exposed to an abrupt signal change, a positive step followed by a reverse step back to the starting value under various conditions is applied to the amplitude and phase angle of a sinusoidal signal, respectively. Studies indicate that under both types of steps the algorithm shows similar dynamic behavior. Here we used a block size of 12 samples since only two weights are to be computed (i.e. .1 cycle). The results of the amplitude step (10% of normal value) and phase step ( $\pi/18$  rad), are presented by Figs. 2.3- 2.5, respectively without any iterations used for blocks of data. The steps occur at 0.02 and 0.06 s. One can observe that the outputs track the changes in the inputs extremely fast. It took 1.39 ms and 1.94 ms to fully track the amplitude step change and 1.67 ms and 1.81 ms to track phase step change that occurred at two different times respectively. To investigate the effect of prefiltering on the algorithm dynamic performance, a third order Butterworth low-pass filter with a cutoff frequency of 320 Hz is used to process the input signals. Fig. 2.5 shows the result of amplitude step test. Compared to Fig. 2.3, which shows the transient behavior without signal prefiltering, one can see that the low pass just slows the response from 3 to 5 ms with no significant overshoot and undershoot and it is still less complex than the RWT-based method [18] that takes about a quarter cycle of fundamental component time period, DFT-based methods [10] -[12] and faster than instantaneous sample-based methods [14], [16] that require full cycle of fundamental component about 16.66 ms. A dynamic amplitude response for a ramp change is also done. The ramp occurs at 0.02 s and continues to steadily increase till 0.06 s then it drops back to the original value as shown in Fig 2.6. It is clear that the algorithm tracks the constant changes steadily and the drop is tracked just in 1.67 ms. The performance of the algorithm showed similar result in phase test.

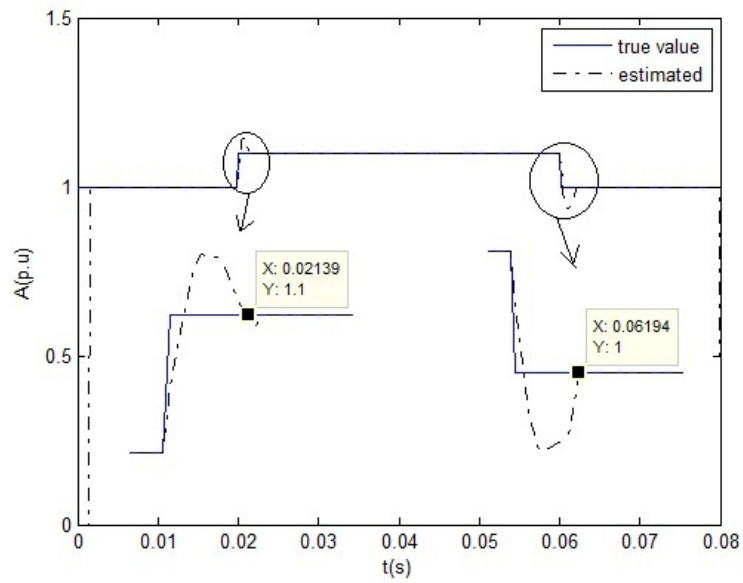


Figure 2.3: Dynamic response for the amplitude step

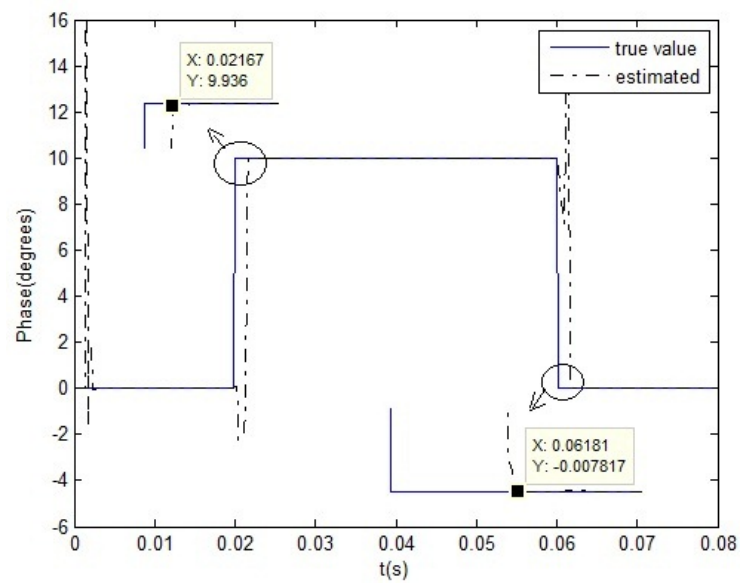


Figure 2.4: Dynamic response for the phase step



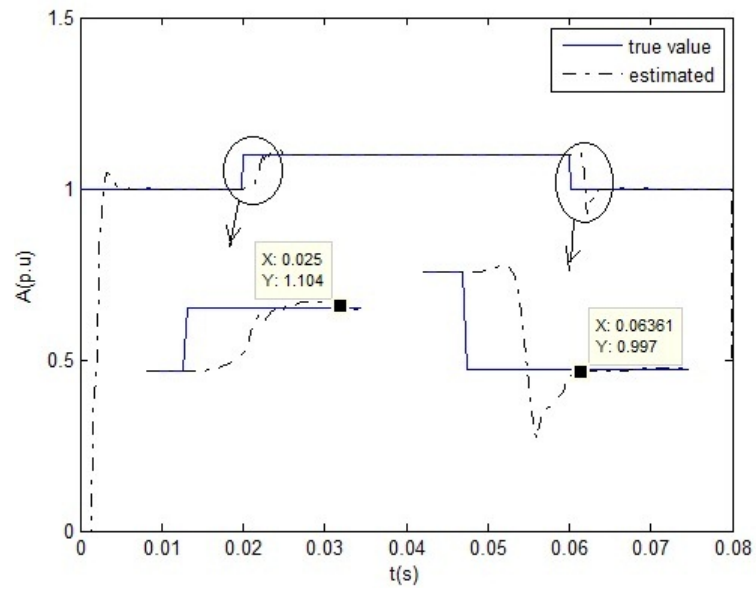


Figure 2.5: Dynamic response for the with amplitude step with prefiltering.

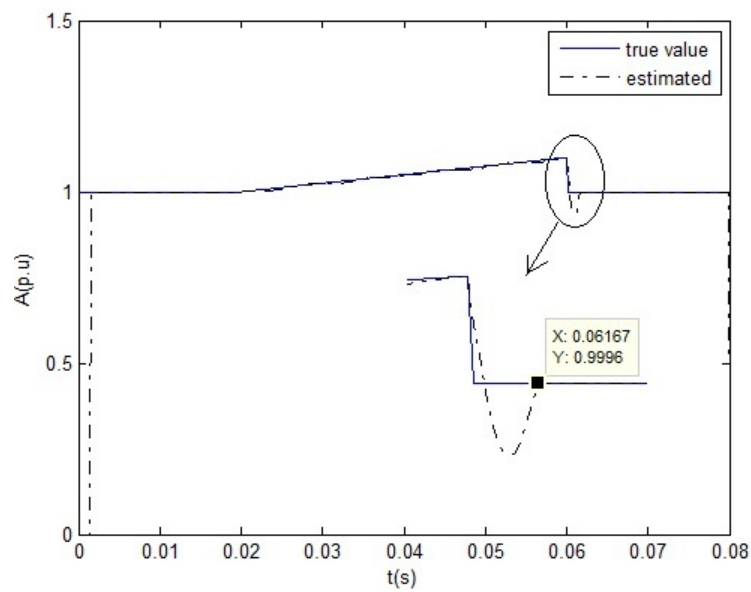


Figure 2.6: Dynamic response for the amplitude ramp change

#### 2.4.4 Transient Test

A synthetic three phase waveforms based on the model in (2.5), is used for testing the performance for eliminating the decaying dc offset. A change in the amplitude is done beginning 0.05 s, similar to a three-phase fault generated in real system. Fig. 2.7 shows the current waveform of Phase A used for simulation. One can see that the signal is contaminated with decaying dc component and high frequency noise is present at the beginning and during the fault. The third order Butterworth low pass filter with a cutoff frequency of 320 Hz is used to attenuate the high-frequency components. Parameters estimation for steady state (twenty cycles after the fault occurs) is used as a reference to measure the TVEs. As shown in the Table 2.3, the results are compared with the conventional full-cycle DFT(FCDFT), half-cycle DFT(HCDFT or STFT), least error square(LES), hybrid method [21], and simplified algorithm (SIMS) in [27]. In the Table 2.3,  $t_s$  is the time(in cycles) when the TVEs are measured. For high accuracy, the algorithm was adjusted to a full cycle window span. The results show that the accuracy is clearly comparable to those of LES. But it is superior in terms of convergence characteristics to LES, SIM3, and HM using first order optimization (Jacobian) technique.

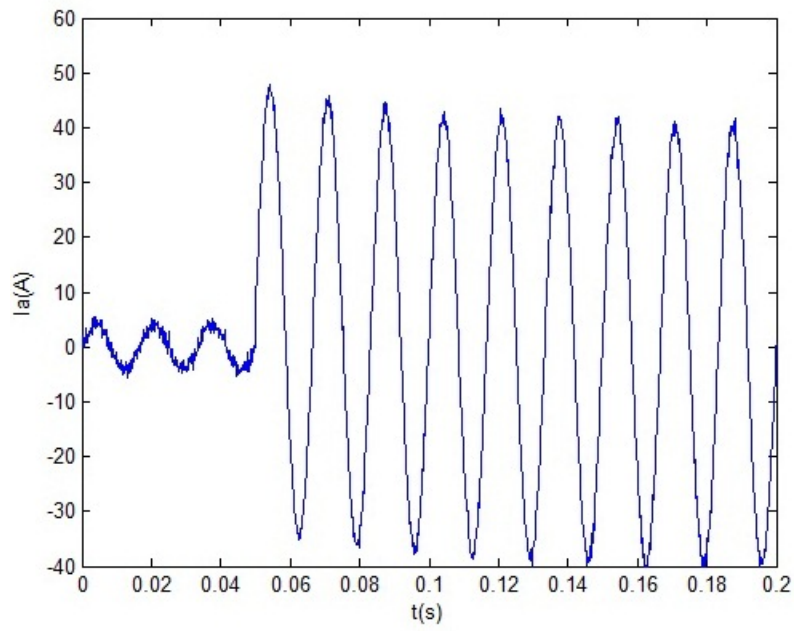


Figure 2.7: Simulated Current  $I_A$  waveform for transient test

Table 2.3: Test Results for decaying dc offset

Algorithm	ts(cycle)	IA	IB	IC
		TVE(%)	TVE(%)	TVE(%)
FCDFT	5	0.9474	0.9587	0.9559
HCDFT	12.5	1.0148	1.0196	1.0158
LES	1.0	0.1036	0.1073	0.1058
SIM3	1.0	0.1285	0.1089	0.1153
HM	1.0	0.1273	0.1168	0.1132
B-LMS	1.0	0.1023	0.1034	0.1046

## 2.5 Conclusions

The chapter introduces a new adaptive filtering approach to solve the problem of phasor estimation when the frequencies of components are known. The algorithm features very fast response and high accuracy over varied conditions in noise. It uses less than a quarter cycle of fundamental component signal to estimate amplitude and phase for a signal contaminated with harmonics or interharmonics. A high frequency resolution could be achieved using this model based technique by appropriately modeling weights. The decaying dc component can be completely removed using this technique. The performance of the algorithm is evaluated under a variety of conditions that includes static test, noise test, dynamic test and transient test. Comparison with other techniques demonstrates the advantage of using this approach. The computational burden is minimal when

compared to non-linear or second order methods or wavelet based methods; accuracy is high and response is very rapid to satisfy time-critical demand of the real time applications in power system. This model can be easily adapted to drifts in nominal frequency when it is known.

This is one of the most efficient time domain methods. So only accurate information of amplitude and phase can be extracted using this method, not the frequency information. If one method tries to extract both time and frequency domain information from a signal, for instance RWT, its complexity goes high, its response time is reduced, its accuracy is usually limited due to a trade-off. In high noise environment difficulty increases further. But a combination of two best methods, each extracting time and frequency domain information separately, would easily be better. Once this B-LMS technique is combined with a fast frequency domain method, whose computations can precede this algorithm, then it can be used for frequency distorted signals too. In power systems, as is well known, frequency is much more tightly regulated parameter than amplitude and phase of various signal components, where this technique can be productively employed. Further research is required to estimate the nominal frequencies and its drift using some efficient approach and in combination with this filtering algorithm.

## CHAPTER 3: FREQUENCY ESTIMATION FROM NONUNIFORM SAMPLES

Bone of our bone, and flesh of our flesh, and blood of our blood,  
are there half-brutish prehistoric brothers and sisters.  
Gridled about with the immense darkness of  
this mysterious universe even as we are,  
they were born and died, suffered and struggled.  
Given over to fearful crime and passion,  
plunged in the blackest ignorance,  
preyed upon by hideous and grotesque delusions,  
yet steadfastly serving the profoundest of ideals in their fixed faith  
that existence in any form is better than non-existence, they ever  
rescued triumphantly from the jaws of ever-imminent destruction  
the torch of life which, thanks to them, now lights the world for us.  
How small indeed, seem individual distinctions when  
we look back on these overwhelming numbers of human beings  
panting and straining under the pressure of that vital want !  
And how inessential in the eyes of God must be  
the small surplus of the individual's merit,  
swamped as it is in the vast ocean of the common merit of mankind,  
dumbly and undauntedly doing the fundamental duty,  
and living the heroic life !  
We grow humble and reverent  
as we contemplate the prodigious spectacle.

*William James, in Human Immortality, 1898.*

Sinusoid signals with multiple frequencies appear in various systems and their frequencies may carry some important features. Frequency estimation from their discrete samples is one of the fundamental problems and many frequency estimators have been proposed for uniform sampling setting <sup>1</sup>.

In this chapter, frequency estimators based on adaptive notch filtering are proposed for nonuniform sampling setting. Most signal processing algorithms are designed to handle uniformly sampled signals. Real world signals usually exist in its analog form, they are converted to digital form by sampling at equal intervals. If  $y_{\text{analog}}(t)$  is a *continuous time signal* (analog), its samples give a discrete/digital time signal:

$$y_{\text{digital}}(n) = y_{\text{analog}}(n\Delta T), n = 0, \pm 1, \pm 2, \dots \quad (3.1)$$

The *sampling interval* is  $\Delta T$ . Often this interval is normalized to  $\Delta T = 1$ , by a simple rescaling of the time variable.

We observe that some dynamic systems designed for continuous time signals associated with adaptive notch filters can be solved in nonuniformly sampled time steps with high accuracy. This leads us to propose a digital adaptive notch filtering method to estimate frequency of a sinusoidal signal with single frequency from its nonuniform samples. The proposed method exhibits convergent and robust frequency estimation in the presence of random sampling noises, and its variance is comparable to the Cramer-Rao lower bound in the presence of additive white noise. The above method designed for single frequency estimation could track abrupt single frequency change of an input signal, but it is not applicable directly for multiple frequency estimation. Our simulations show the proposed estimators have robust performance for sinusoidal signals with multiple distinct fre-

---

<sup>1</sup>Part of the material in this section is reprinted from “Frequency estimation of sinusoids from nonuniform samples” by Syed Alam Abbas, Qiyu Sun, and Hassan Foroosh, Signal Processing Elsevier, vol. 129, pp. 67-81, 27 May 2016 ©2016 Elsevier, with permission from Elsevier.

quencies, and they can be used to separate two very close frequencies of an input signal in a highly noisy sampling environment.

### 3.1 Introduction

Consider a mixture of sinusoidal signals, whose  $k$ -th component has amplitude  $A_k$ , frequency  $\theta_k$  and phase  $\phi_k$ ,  $1 \leq k \leq K$ ,

$$y(t) = \sum_{k=1}^K A_k \sin(\theta_k t + \phi_k). \quad (3.2)$$

Such sinusoidal signals are encountered in active noise and vibration control, wireless communications, audio, radar and sonar signal processing [40, 41, 42, 43]. In telecommunication systems, the frequencies  $\theta_k$ ,  $1 \leq k \leq K$ , contain carrier's phase information necessary for synchronization of demodulators or other components of a receiver system.

The estimation problem of frequencies  $\theta_k$ ,  $1 \leq k \leq K$ , of the signal  $y$  is a fundamental problem in systems theory with many applications. It has been intensively studied in signal processing, instrumentation and measurements, and control theory. Many frequency estimators have been proposed, including adaptive notch filtering, time frequency representation, phase locked loop, eigensubspace tracking, extended Kalman filtering, internal model method etc, see [44, 45, 46, 47] and references therein.

Most of existing estimators are derived for uniformly sampled data  $y(n\Delta T)$ ,  $n \geq 0$ , with uniform sampling frequency  $1/\Delta T$ , and often only for a single frequency, i.e.,  $K = 1$ . In this paper, we consider multiple frequency estimation problem of the signal  $y$  from its nonuniform samples,

$$z_n = y(T_n) + w_n, \quad n \geq 0, \quad (3.3)$$



corrupted by additive noises  $w_n$ , where  $T_n, n \geq 0$ , are sampling times.

Nonuniform sampling arises in many applications such as computer graphics, frequency scanning interferometry, magnetic resonance imaging, computer tomography scans and Radon imaging [48, 49, 50, 51, 52, 53]. Uniform sampling is well studied and it has been widely used in engineering applications. However in some applications, nonuniform sampling is necessary and it has better performance. For instance, in antialiasing in computer graphics [54], better results can be obtained with random sampling instead of uniform sampling. The sampling operation could be costly, and a low number of samples (but not necessarily uniform) is more desirable. For instance, in frequency scanning interferometry, the sampling effort is measured by the acquisition time at a given point, and the optimal sampling scheme is usually nonuniform [55]. Nonuniformly sampled data are harder to analyze and the related methods, even for a “simple” task of obtaining the discrete nonuniform Fourier transform, are much more difficult and often iterative [48, 56]. Several statistical frequency estimators (based on maximum likelihood estimation and filter banks) from nonuniformly sampled data have been proposed in the literature [50, 57, 58, 59, 60, 61, 62]. For nonuniform sampling problems in signal processing, the reader may refer to [63, 64, 65].

In many applications of signal processing, it is desirable to eliminate or extract sine waves from observed data or to estimate their unknown frequencies. Since the frequencies often vary with time, it is useful to apply adaptive notch filters (ANF's) that adapt their notch frequencies as a function of the observed time series, see [66], [67] and references therein. The ANF method is one of most suitable techniques to separate sinusoidal components of unknown frequencies buried in noise, and/or retrieve such periodic components ([68, 69, 70, 71, 72, 73, 74]). It is robust in the presence of sampling noise and it is capable of changing the notch frequency accordingly. Various architectures have been proposed for the construction of adaptive notch filters, see for instance [40, 75, 76, 77, 78, 79, 80].

Frequency estimation problem using ANF is modeled as a nonlinear system identification of a dynamic system either in continuous time (CT) [40] or in discrete time (DT) [68]. The CT model systems are native to the physical world, they have a built-in capability to cope with the nonuniformly sampled signal, and they offers certain advantages over purely DT model systems [84, 85]. Compared to the DT model, direct estimation of CT models is usually stable, accurate and free from undesirable sensitivity problems, particularly at high sampling rates. The frequency estimator developed in this paper is based on ANFs, which are governed by some CT dynamic systems [68, 69, 70, 71, 72, 73, 74, 81, 82, 83]. We mainly focus on a particular ANF governed by the following dynamic system,

$$\begin{cases} Dx_1 = x_2 \\ Dx_2 = -2\xi\theta x_2 - \theta^2 x_1 + \theta^2 y \\ D\theta = -\gamma(\theta^2 y - 2\xi\theta x_2)x_1, \end{cases} \quad (3.4)$$

where  $D$  represents the derivative with respect to time  $t$ ,  $x_1, x_2, \theta$  are states of the system,  $y$  is the excitation sinusoidal input with single frequency  $\theta_0$  (i.e.,  $K = 1$  in (3.2)),  $\xi$  is the notch depth, and  $\gamma$  is the adaptation speed. The above system of nonlinear ordinary differential equations (ODE) has good noise rejection capability. It was proposed by Regalia in [40] as a DT filter, it was later adapted by Bodson and Douglas [81] for a CT system, and finally a modified version was proposed by Hsu, Ortega and Damm [68]. We have chosen this dynamic system due to its superior performance compared to the other systems that can be used with a similar discretization procedure, see Section 3.2.5 for performance comparison.

The main difficulty in handling CT dynamic systems directly is the problem of evaluating derivatives of the input signal (with unknown parameters) from its nonuniform samples numerically, the numerical differentiation process in a highly noisy environment is usually unstable and impractical [86, 87, 88]. In this chapter, we propose a Taylor-like approximation of the dynamic system (3.4)

that is robust to noise and achieves high accuracy. Based on the above approximation, we introduce an ANF method (3.17) to estimate the frequency of a sinusoidal signal from its nonuniform samples. The proposed discrete ANF method reconciles the merits of CT models while restricting itself to operate directly on the DT data.

This paper is organized as follows. Section 3.2 discusses a Taylor-like approximation technique to solve the system (3.4) and a frequency estimation of the unknown input signal with single frequency. We propose the frequency estimator (3.17), perform the local stability analysis, and study its convergence, noise characteristics, statistical properties, and comparison to the conventional ODE solver for the dynamic systems associated with the ANF methods. We also perform a comparison of our method with a state of the art discrete ANF [79]. Section 3.3 describes extensions of the single frequency estimator (3.17) to multiple frequency estimations with two configurations, the cascade ANF method and the prefiltering ANF method. The two proposed multiple frequency estimators have robust performance for sinusoidal signals with multiple distinct frequencies or related harmonic frequencies. Most of the known frequency estimators have poor performance when input signal has two very close frequencies in a highly noisy environment, a pathological case where the estimation error is related to both the difference in the frequency and the noise level, see [89] and references therein. Our simulation indicates that the cascade ANF method has sound performance even in the separation of very close frequencies of the input signal in a highly noisy sampling environment. We close the paper with concluding remarks in Section 3.4.

Notation: We use Euler notation for expressing derivatives,  $D^n$  instead of  $D_t^n$  to denote the  $n^{\text{th}}$  derivative with respect to time  $t$ .

## 3.2 Single frequency estimation

Consider a sinusoidal input

$$y(t) = A \sin(\theta_0 t + \phi), \quad (3.5)$$

where  $A, \theta_0, \phi$  are its amplitude, frequency, and phase, respectively. In the first subsection, we propose a discrete ANF method to estimate frequency  $\theta_0$  of the sinusoid signal  $y$  from its nonuniform samples,

$$z_n = y(T_n) + w_n, \quad n \geq 0, \quad (3.6)$$

corrupted by additive noise  $w_n$  at sampling times  $T_n, n \geq 0$ . Then in the next four subsections, we discuss local stability, convergence, approximation error, statistical characterization, and extensions of the proposed ANF method. We also compare the performance of our approach (3.17) with some of the existing ANF methods for estimating frequency in Section 3.2.2 and Section 3.2.5.

### 3.2.1 The proposed method

The dynamical system (3.4) converges to its unique periodic orbit,

$$[x_1, x_2, \theta]^T = \left[ \frac{-A \cos(\theta_0 t + \phi)}{2\xi}, \frac{A\theta_0 \sin(\theta_0 t + \phi)}{2\xi}, \theta_0 \right]^T, \quad (3.7)$$

when the adaption speed  $\gamma$  satisfies the following condition [68]:

$$0 < \gamma < 4\xi/A^2. \quad (3.8)$$

. The dynamical system (3.4) can be rewritten as follows:

$$D\mathbf{X} = \mathbf{F}(t, \mathbf{X}), \quad (3.9)$$

where  $\mathbf{X} = [x_1, x_2, \theta]^T$  is the state of the system, and

$$\mathbf{F}(t, \mathbf{X}) = [x_2, -2\xi\theta x_2 - \theta^2 x_1 + \theta^2 y, \gamma(2\xi\theta x_2 - \theta^2 y)x_1]^T$$

is a real analytic function of  $t$  and  $\mathbf{X}$ . Therefore,  $\mathbf{X}(t)$  is real analytic by the Cauchy-Kovalevskaya theorem [90, Theorem 2 of Chapter 4], and it has the following Taylor expansion,

$$\mathbf{X}(t) = \sum_{k=0}^m \frac{D^k \mathbf{X}(T_n)}{k!} t^k + \int_{T_n}^t D^m \mathbf{F}(s, \mathbf{X}(s)) \frac{(t-s)^m}{m!} ds \quad (3.10)$$

$\forall T_n \leq t \leq T_{n+1}$  and  $m \geq 0$ .

For an input signal of sinusoidal type, we observe that the state vector  $\mathbf{X}(t)$  of the dynamical system (3.4) can be approximated by Taylor polynomials  $\sum_{k=0}^m D^k \mathbf{X}(T_n) (t - T_n)^k / k!$  of low order  $m \leq 4$ ,

$$\mathbf{X}(t) \approx \sum_{k=0}^m \frac{D^k \mathbf{X}(T_n)}{k!} (t - T_n)^k =: \mathbf{X}_{m,n}(t), T_n \leq t \leq T_{n+1}, \quad (3.11)$$

when the maximal sampling gap  $\max_n (T_{n+1} - T_n)$  is small, which in turns depends on the signal frequencies. The problem how to choose the maximal sampling gap and the choice of model order  $m$  is discussed further in section 3.2.3, cf. (3.27), where it is shown to be dependent on the signal frequency  $\theta$ .

For expressing Taylor polynomial approximation  $\mathbf{X}_{m,n}(t)$ ,  $m \leq 4$ , explicitly, we introduce few auxiliary variables  $x_3 = \theta^2$ ,  $x_4 = x_3 y$ ,  $x_5 = 2\xi\theta x_2$  and  $x_6 = x_1 x_3$ . Consider the derivatives of order up to 4:

1<sup>st</sup> order derivatives,

$$\left\{ \begin{array}{l} Dx_1 = x_2 \\ Dx_2 = x_4 - x_5 - x_6 \\ D\theta = -\gamma(x_4 - x_5)x_1 \\ Dx_3 = 2\theta D\theta \\ Dx_4 = x_3 Dy + y Dx_3 \\ Dx_5 = 2\xi(\theta Dx_2 + x_2 D\theta) \\ Dx_6 = x_1 Dx_3 + x_3 Dx_1. \end{array} \right. \quad (3.12)$$

2<sup>nd</sup> order derivatives,

$$\left\{ \begin{array}{l} D^2x_1 = Dx_2 \\ D^2x_2 = Dx_4 - Dx_5 - Dx_6 \\ D^2\theta = -\gamma\{(x_4 - x_5)Dx_1 + x_1(Dx_4 - Dx_5)\} \\ D^2x_3 = 2\{\theta D^2\theta + (D\theta)^2\} \\ D^2x_4 = -x_3^2y + 2DyDx_3 + yD^2x_3 \\ D^2x_5 = 2\xi(\theta D^2x_2 + 2Dx_2D\theta + x_2D^2\theta) \\ D^2x_6 = x_1D^2x_3 + 2Dx_1Dx_3 + x_3D^2x_1. \end{array} \right. \quad (3.13)$$

3<sup>rd</sup> order derivatives,

$$\left\{ \begin{array}{l} D^3x_1 = D^2x_2 \\ D^3x_2 = D^2x_4 - D^2x_5 - D^2x_6 \\ D^3\theta = -\gamma\{(x_4 - x_5)D^2x_1 + 2(Dx_4 - Dx_5)Dx_1 \\ \quad + x_1(D^2x_4 - D^2x_5)\} \\ D^3x_3 = 2\{\theta D^3\theta + 3D\theta D^2\theta\} \\ D^3x_4 = -x_3(\theta^2 Dy + 2\theta D\theta y) \\ \quad - 3x_4 Dx_3 + 3Dy D^2x_3 + y D^3x_3 \\ D^3x_5 = 2\xi(\theta D^3x_2 + 3D^2x_2 D\theta + 3Dx_2 D^2\theta + x_2 D^3\theta) \\ D^3x_6 = x_1 D^3x_3 + 3D^2x_1 Dx_3 + 3Dx_1 D^2x_3 + x_3 D^3x_1. \end{array} \right. \quad (3.14)$$

4<sup>th</sup> order derivatives,

$$\left\{ \begin{array}{l} D^4x_1 = D^3x_2 \\ D^4x_2 = D^3x_4 - D^3x_5 - D^3x_6 \\ D^4\theta = -\gamma\{(x_4 - x_5)D^3x_1 + 3(Dx_4 - Dx_5)D^2x_1 \\ \quad + 3(D^2x_4 - D^2x_5)Dx_1 + x_1(D^3x_4 - D^3x_5)\}. \end{array} \right. \quad (3.15)$$

In above computation of derivatives of high orders, we require knowledge of derivatives of the excitation sinusoidal signal  $y$  with unknown frequency  $\theta_0$ . As the dynamical system (3.4) converges to its unique periodic orbit (3.7), we may replace the true frequency  $\theta_0$  by the third state  $\theta$ , and the digital differentiator  $Dy$  of the sinusoidal signal  $y$ , with the product  $-2\xi\theta x_1$ . This leads to the

following approximation of  $Dy$ ,  $D^2y$ ,  $D^3y$ , and  $D^4y$ :

$$\begin{cases} Dy = A\theta_0 \cos(\theta_0 t + \phi) \approx -2\xi\theta x_1 \\ D^2y = -\theta_0^2 y \approx -\theta^2 y \\ D^3y = -\theta_0^2 Dy \approx -\theta^2 Dy \\ D^4y = \theta_0^4 y \approx \theta^4 y. \end{cases} \quad (3.16)$$

Inspired by (3.7), (3.11) and (3.16), we propose the following discretized adaptive notch filtering of order  $m$  to solve our frequency estimation problem,

$$\mathbf{X}(T_{n+1}) = \sum_{k=0}^m \frac{D^k \mathbf{X}(T_n)}{k!} (\Delta T_n)^k, \quad n \geq 0, \quad (3.17)$$

where  $2 \leq m \leq 4$ , the derivatives of  $\mathbf{X} = [x_1, x_2, \theta]^T$  are given in (3.12)–(3.15), and the ones for  $y$  are provided by (3.16). The problem of how to select the approximation order  $2 \leq m \leq 4$  of the proposed discretization procedure will be discussed in section 3.2.3.

### 3.2.2 Local Stability Analysis and Convergence

In this section, we discuss the local stability of the proposed dynamic system, demonstrate convergence of the proposed method (3.17) in general sampling setting, and compare its performance to frequency estimation using the conventional Runge-Kutta method under uniform sampling setting [93].

To study the local stability of the discrete system (3.17), we introduce the following continuous



autonomous 4<sup>th</sup> order system,

$$\begin{cases} Dx_1 = x_2 \\ Dx_2 = -2\xi\theta x_2 - \theta^2 x_1 + \theta^2 y \\ D\theta = -\gamma(\theta^2 y - 2\xi\theta x_2)x_1 \\ Dy = -2\xi\theta x_1, \end{cases} \quad (3.18)$$

where  $\mathbf{X} = [x_1, x_2, \theta, y]$  is the state vector of the system. The proposed system (3.18) is highly nonlinear and there is no explicit solution. Now we consider the behaviour of the above system at one of the equilibrium points, provided that the system converges in the periodic orbit defined in (3.7). The linearized system near the equilibrium point  $\mathbf{X}^*$  is given by

$$D\mathbf{X} = J\mathbf{X}, \quad (3.19)$$

where  $z_1 = \theta y - x_2\xi$ ,  $z_2 = \theta y - 2x_2\xi$ , and

$$J = \left[ \begin{array}{cccc} 0 & 1 & 0 & 0 \\ -\theta^2 & -2\theta\xi & 2z_2 - 2\theta_2 x_1 & \theta^2 \\ -\gamma\theta z_2 & 2\gamma\theta x_1\xi & -2\gamma x_1 z_1 & -\gamma\theta^2 x_1 \\ -2\theta\xi & 0 & -2x_1\xi & 0 \end{array} \right] \bigg|_{X^*}. \quad (3.20)$$

By Poincare-Lyapunov Theorem [92], the linearized system (3.19) at the equilibrium point  $X^*$  is stable iff all eigenvalues of  $J$  has nonpositive real part. For the pure excitation input sinusoidal signal  $y = A \sin(\theta_0 t)$  (i.e.,  $\phi = 0$  in (3.5)), the equilibrium points of the continuous system (3.18) are

$$\mathbf{X}_0 = \left[ \frac{-A \cos(\theta_0 t)}{2\xi}, \frac{A\theta_0 \sin(\theta_0 t)}{2\xi}, \theta_0, A \sin(\theta_0 t) \right]. \quad (3.21)$$

Direct computation shows that the eigenvalues of the linearized matrix  $J$  at  $\mathbf{X}_0$  with  $A = 1$  and

$t = 1$  are

$$\lambda_1 = -\theta_0 i, \lambda_2 = \theta_0 i, \lambda_3 = \frac{\theta_0(\sqrt{\Delta} + 2\gamma \sin(2\theta_0) - 16\xi^2)}{16\xi},$$

and

$$\lambda_4 = \frac{\theta_0(-\sqrt{\Delta} + 2\gamma \sin(2\theta_0) - 16\xi^2)}{16\xi},$$

where  $\Delta = 2(\gamma^2 - \gamma^2 \cos(4\theta_0) + 128\xi^4 - 64\gamma\xi \cos^2(\theta_0) - 32\gamma\xi^2 \sin(2\theta_0))$ . The first two eigenvalues  $\lambda_1, \lambda_2$  are purely imaginary. Therefore, for the local stability of the continuous system (3.18) at the equilibrium point  $\mathbf{X}_0$  in (3.21), we require that the third and fourth eigenvalues (depending on  $\gamma, \xi$  and  $\theta$  at the equilibrium point  $\mathbf{X}^*$ ) have negative real parts.

Observe that  $\lambda_3$  and  $\lambda_4$  form the conjugate roots of a quadratic equation, where  $\Delta$  is the discriminant. Thus, we can determine how to adjust the parameters  $\gamma$  and  $\xi$  for the local stability.

- For the case when  $\Delta < 0$ , the two eigenvalues  $\lambda_3, \lambda_4$  occur as a complex conjugate pair and the stability will only depend on their real parts. This implies that the system (3.18) has local stability near the equilibrium point if

$$\gamma < \frac{4\xi^2}{\sin \theta_0 \cos \theta_0}, \quad (3.22)$$

- When  $\Delta = 0$  the two eigenvalues are real and  $\lambda_3 = \lambda_4$ , the stability will still depend on the same condition (3.22).
- For the case when  $\Delta > 0$ , eigenvalues  $\lambda_3, \lambda_4$  are real and distinct. For the local stability of the system (3.18), we require that  $\lambda_3 + \lambda_4 < 0$  and  $\lambda_3 \lambda_4 > 0$ , which are equivalent to the following:

$$\gamma < \frac{4\xi^2}{\sin \theta_0 \cos \theta_0} \quad \text{and} \quad \frac{\gamma \theta_0^2 \cos^2 \theta_0}{2\xi} > 0.$$

The last condition is satisfied simply by choosing  $\gamma, \xi > 0$ , and these parameters are by definition

positive. Hence in all cases the local stability depends only on satisfying the stability condition (3.22), cf. the similar stability condition (3.8) in [68].

To study the local stability of the system (3.18) using experiments, we choose the adaption alertness parameters  $\gamma$  and noise sensitivity parameter  $\xi$  as in (3.8). For  $\xi = 0.1$  and  $\gamma = 0.001$ , numerical simulation indicates that the third and fourth eigenvalues  $\lambda_3$  and  $\lambda_4$  remain strictly negative and negatively increasing about the  $\theta_0 = 2\pi f_0$ , where  $f_0 \in [30, 300]$  Hz, see Figure 3.1.

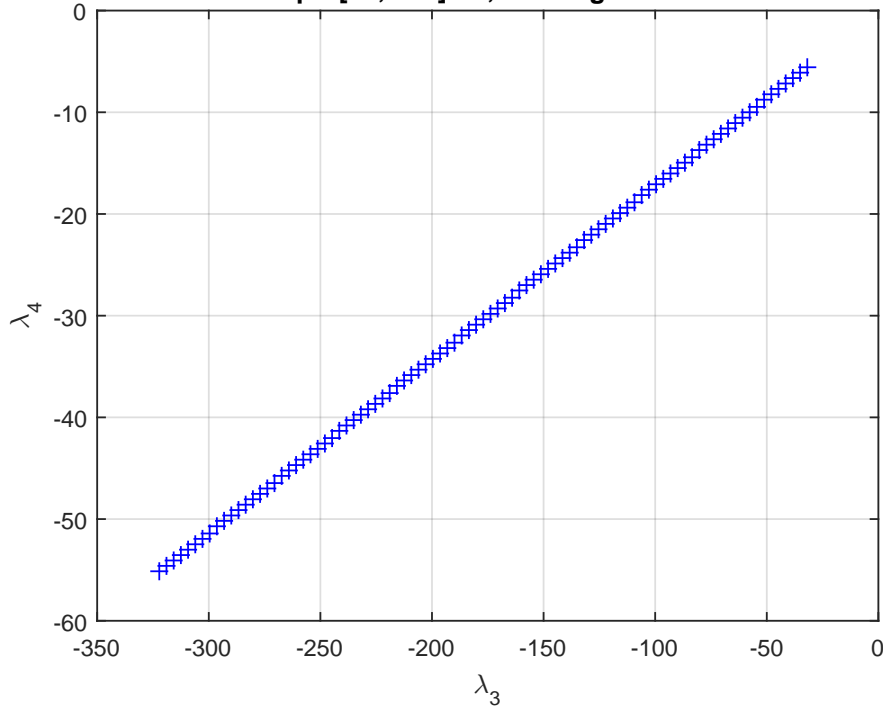


Figure 3.1: Eigenvalue plot of the Jacobian system,  $\xi = .1$ ,  $\gamma = .001$  in a frequency range,  $\theta_0 = 2\pi f_0$  where  $f_0 \in [30, 300]$  Hz in steps of 3 Hz. The eigenvalues  $\lambda_3, \lambda_4$  remain strictly  $< 0$  making the actual nonlinear system linearized locally at a point in the periodic orbit to be locally asymptotically stable.

For different values of parameters  $\xi$  and  $\gamma$ , one may find the range of frequencies so that the third and fourth eigenvalues have strictly negative real part, and hence the nonlinear dynamic

system (3.18) is locally asymptotically stable. Thus with the use of both qualitative and quantitative techniques we have established the local stability of the proposed dynamic system. There are still mathematical difficulties (such as establishing global convergence, setting parameter bounds and finding optimal and robust parameter) to study stability for the proposed dynamic system (3.18) in the discrete setting using only the nonuniform sampling data (3.17). Our numerical results show that all trajectories starting close to the periodic orbit region will stay near the region when the approximation order is chosen appropriately and nonuniform sampling is taken with small sampling gap (depending on the signal frequency).

The ANF technique governed by the dynamic system (3.4) is a globally convergent estimator with very high noise immunity [68]. The discretized version (3.17) is based on the observation that the evaluation of derivatives of the input excitation sinusoidal signal could be circumvented by using the interrelationship among the states of the dynamic system, see (3.16). The system (3.17) converges according to our analysis and simulations.

The discrete system (3.17) could be implemented for both nonuniformly or uniformly sampled data without any significant changes to its implementation. The simplest models for nonuniform sampling schemes are the jittered nonuniform sampling [61],

$$T_n = n\Delta T + \tau_n, \quad (3.23)$$

and the additive nonuniform sampling,

$$T_{n+1} = T_n + \tau_n, \quad (3.24)$$

where  $\Delta T$  is the uniform sampling rate and  $\tau_n$  is a family of identically distributed independent random variables with mean  $\Delta T$  and small  $\max_n \tau_n$  [63, 64, 65]. Presented in Fig. 3.2 is the

snapshot of the sinusoid signal  $y$  in (3.5) and its heavily-infected samples  $z_n$  in (3.6) between 20 – 40 ms via the sampling scheme (3.24).

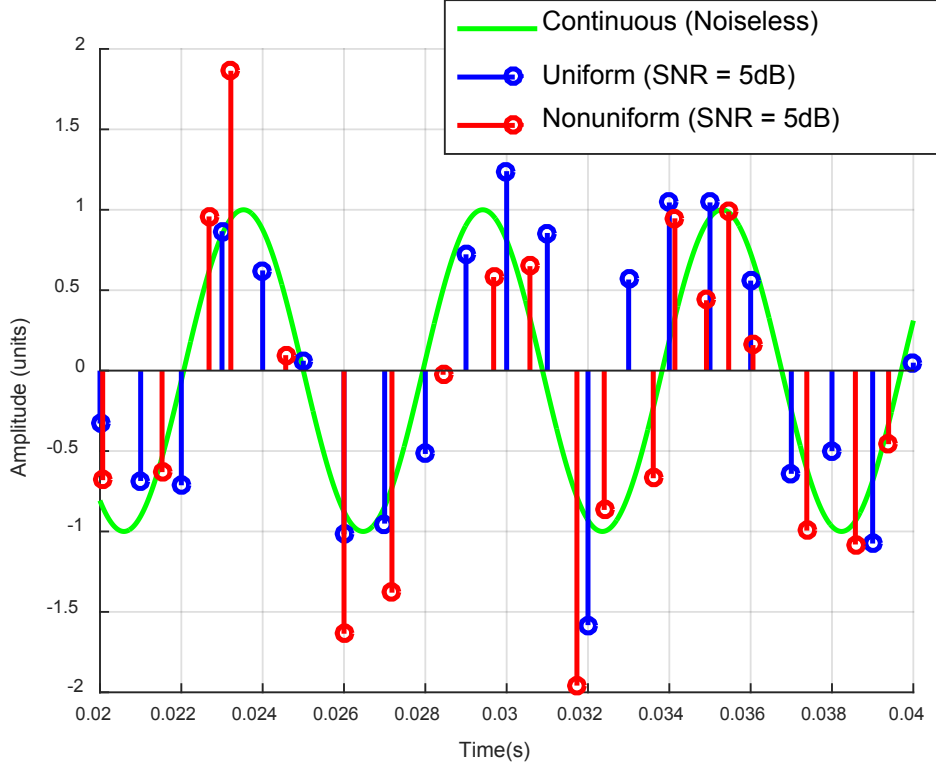


Figure 3.2: In the simulation, the signal (plotted in green) has amplitude  $A = 1$ , frequency  $f_0 := \theta_0/(2\pi) = 170\text{Hz}$  and phase  $\phi = \pi/2$ , and the samples  $z_n$ , corrupted by i.i.d. random noises  $w_n$  with  $\text{SNR} = 5\text{dB}$ , are taken on  $T_n$  in (3.24) with  $\Delta T = 1\text{ms}$  and  $\tau_n = 0$  for uniform sampling (plotted in blue) and i.i.d. random  $\tau_n \in [.5, 1.5]\Delta T$  for nonuniform sampling (plotted in red).

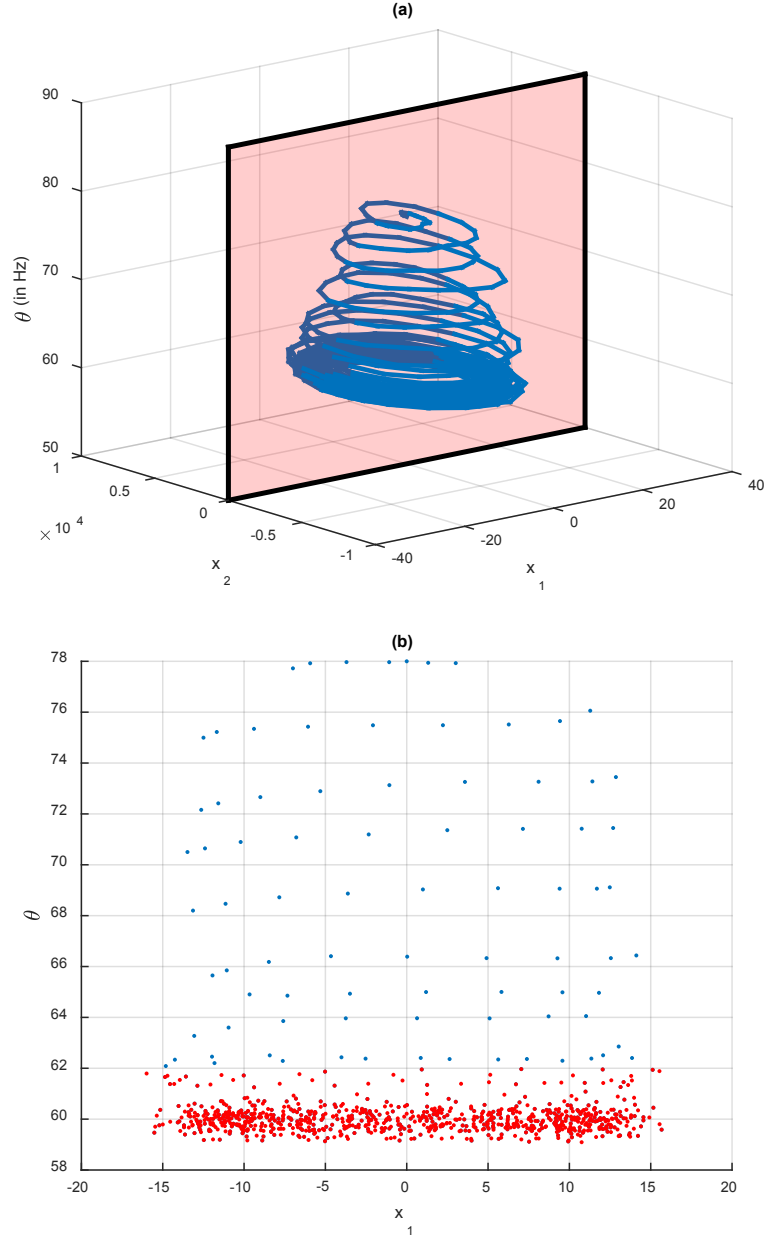


Figure 3.3: Flow trajectories and Poincare section of the discrete system (3.17). In the simulation, the sinusoid input  $y$  in (3.5) has amplitude  $A = 1$ , frequency  $\theta_0 = 120\pi$  and phase  $\phi = \pi/2$ , the sampling procedure (3.6) is taken uniformly at  $T_n = 0.001n, 0 \leq n \leq 1000$ , seconds with additive i.i.d. random noise  $w_n$  at level 5dB, the notch depth and adaption speed in (3.4) are given by  $\xi = .15$  and  $\gamma = .001$ , and the approximation order  $m$  in (3.17) is 4.

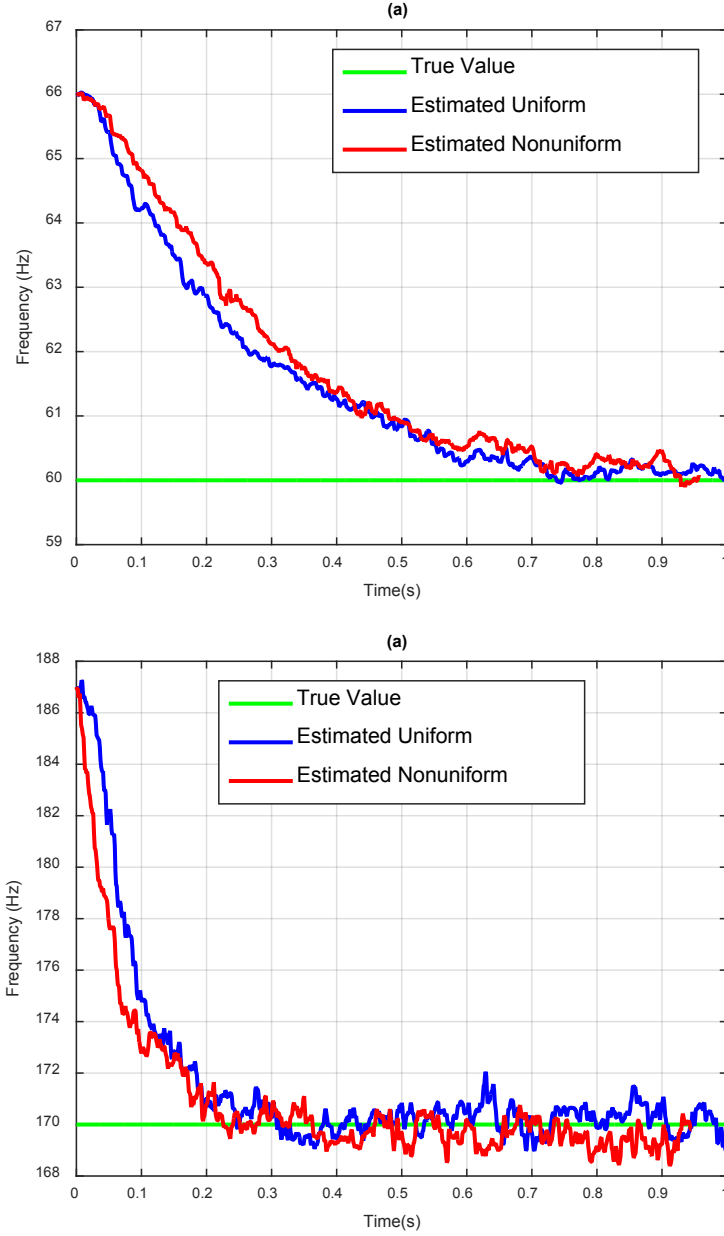


Figure 3.4: Convergence of frequency estimation  $\theta(T_n)/(2\pi)$  in the discrete ANF method (3.17) to the true frequencies  $f_0 := \theta_0/(2\pi)$ . In the simulation, the sampling procedure and the input signal  $y$  are the same as in Figure 3.2 except that  $f_0 = 60\text{Hz}$  (left) and  $f_0 = 170\text{Hz}$  (right). The notch depth, adaption speed and approximation order in (3.4) are the same as in Figure 3.3.

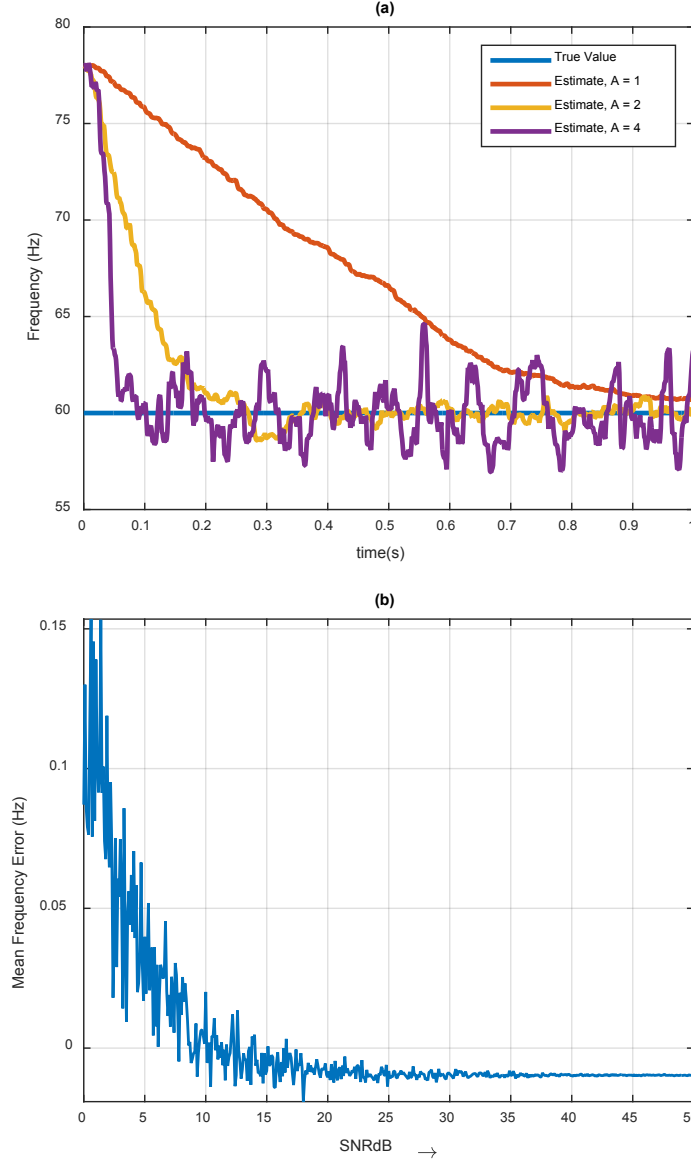


Figure 3.5: On the left is frequency estimation  $\theta(T_n)/(2\pi)$  of the proposed ANF method for input signals  $y$  with different amplitudes, while on the right is mean frequency error of the proposed ANF method for 500 independent trials for each noise level. In the simulation, the signal  $y$ , the sampling procedure, and the input signal  $y$ , the notch depth, adaption speed and approximation order are the same as in Fig. 3.2 and Fig. 3.3, except that  $y$  has amplitudes  $A = 1, 2, 4$  for the left figure, and  $y$  has amplitude  $A = 2$  and SNR of sampling noises  $w_n$  varies from 0 to 50dB for the right figure.



Fig. 3.3(a) shows the flow of trajectories in the three-dimensional state space for the system (3.17), where the frequency of a sinusoid signal could be recovered from the system as one of its state,  $\theta$ . Shown in Fig. 3.3 is a Poincare section [91] of the system (3.17), where the flow of trajectories of the system (3.4) passes through the section  $x_2 = 0$ . The attractor plotted in red reveals the periodic flow of trajectories near the true frequency of the input excitation signal in a very noisy environment, c.f. the unique periodic orbit in (3.7), provided that the initial

$$\mathbf{X}(T_0) = [y(T_0), Dy(T_0), \theta(T_0)]^T \quad (3.25)$$

in (3.17) has relative error between its third state  $\theta(T_0)$  and the true frequency being less than 10%.

The convergence rate of the discrete system (3.17) depends on the adaptation speed, the true frequency of the sinusoid input, and the relative error between the initial frequency and the true frequency of the sinusoid input. The frequency estimation in the proposed ANF method (3.17) for two different frequencies,  $f_1 = 60$  Hz and  $f_2 = 170$  Hz is shown, it is observed to have faster convergence when the true frequency of the input signal is higher, see Figures 3.4.

Our simulations also indicate that the performance of the algorithm (3.17) degrades gracefully as the SNR of additive noise decreases, see Figure 3.5.

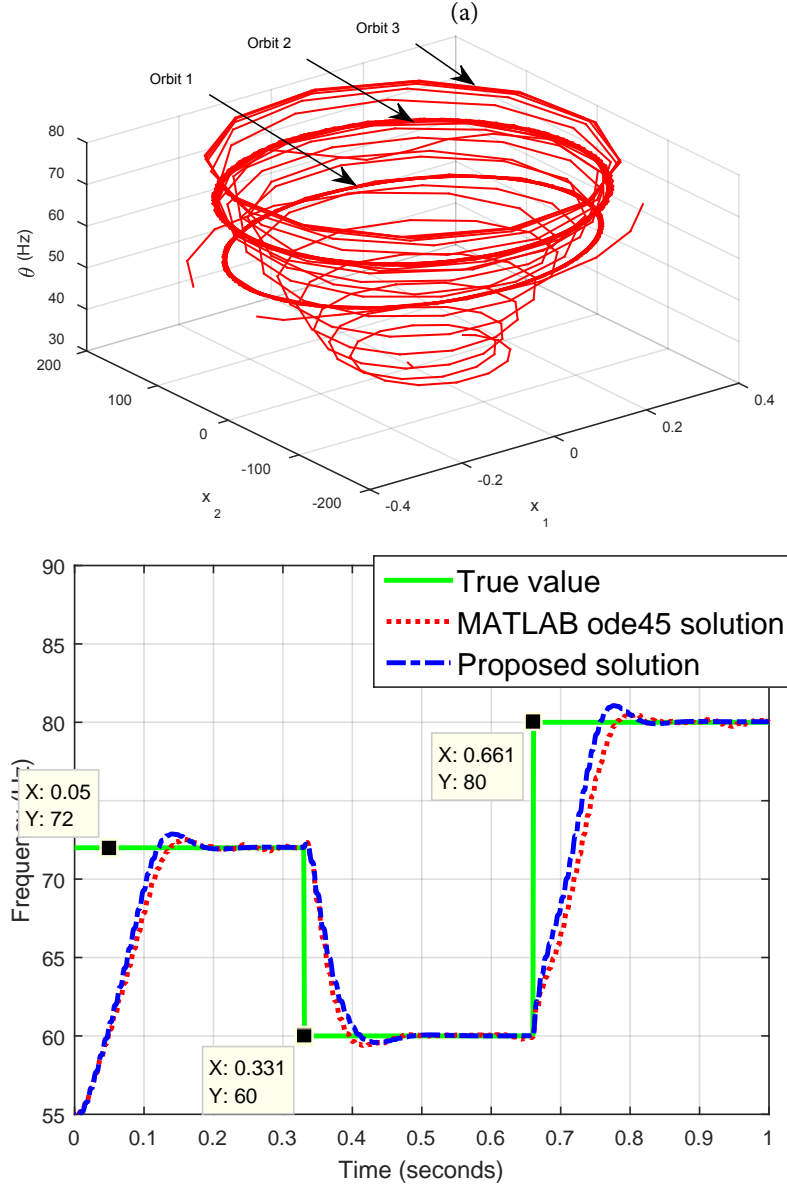


Figure 3.6: Frequency tracking of the proposed ANF method (3.17). The sampling procedure (3.6) is taken uniformly at  $T_n = .001n, 0 \leq n \leq 1000$ , seconds with additive noise  $w_n$  having SNR= 20dB, the adaption speed,  $\gamma = 0.01$ , and the notch depth, and the approximation order in (3.4) are the same as in Figure 3.3.

The proposed ANF method (3.17) is designed for a single frequency estimation of a sinusoid input.

Due to the fast convergence, it could be used to track the frequency change of the input signal,

$$y(n) = \sin[2\pi\tilde{f}n\Delta T + \pi/2] \quad (3.26)$$

where the frequency  $\tilde{f}$  of the input signal (this frequency is not the true instantaneous frequency but follows a model description for tracking ANF in [79]) changes abruptly due to various reasons, such as power surges, mechanical dysfunction or network disruption. Presented in Figure 3.6 is frequency tracking results with uniform sampling gap  $\Delta T = 1\text{ms}$ , where the frequency of the input signal changes abruptly from 72Hz to 60Hz in 1/3 second and from 60Hz to 80Hz in the next 1/3 second. It is observed that the corresponding flow trajectory has three periodic orbits, which are near the true frequencies 72Hz, 60Hz and 80Hz of the input signal.

Now, we compare our solution (3.17) of the dynamic system (3.4) with the direct ODE approximation solution. For uniform sampling with time step  $h > 0$ , the fourth order Runge-Kutta method ([93]) to solve the system (3.9) is given by

$$\mathbf{X}((n+1)h) = \mathbf{X}(nh) + \frac{h}{6}(\mathbf{Y}_{1,n} + 2\mathbf{Y}_{2,n} + 2\mathbf{Y}_{3,n} + \mathbf{Y}_{4,n}),$$

where

$$\begin{cases} \mathbf{Y}_{1,n} = \mathbf{F}(nh, \mathbf{X}(nh)) \\ \mathbf{Y}_{2,n} = \mathbf{F}((n+1/2)h, \mathbf{X}(nh) + \mathbf{Y}_{1,n}h/2) \\ \mathbf{Y}_{3,n} = \mathbf{F}((n+1/2)h, \mathbf{X}(nh) + \mathbf{Y}_{2,n}h/2) \\ \mathbf{Y}_{4,n} = \mathbf{F}((n+1)h, \mathbf{X}(nh) + \mathbf{Y}_{3,n}h), \quad n \geq 0. \end{cases}$$

The above conventional Runge-Kutta method requires the excitation signal  $y$  in the continuous form, which could be obtained from its samples (3.6) by certain interpolation. The Runge-Kutta method is not applicable for nonuniform sampling setting, while our proposed method (3.17) is designed for a generic nonuniform input data. Even for uniform sampling setting, simulations show

that the proposed ANF method (3.17) has slightly faster convergence than the Runge-Kutta method does, and it has better frequency estimation when the input signal has higher frequencies between the two frequencies shown, see Figure 3.7. This demonstrates the importance of proper discretization of the continuous dynamic system (3.4), especially when only the noisy nonuniformly sampled data is available.

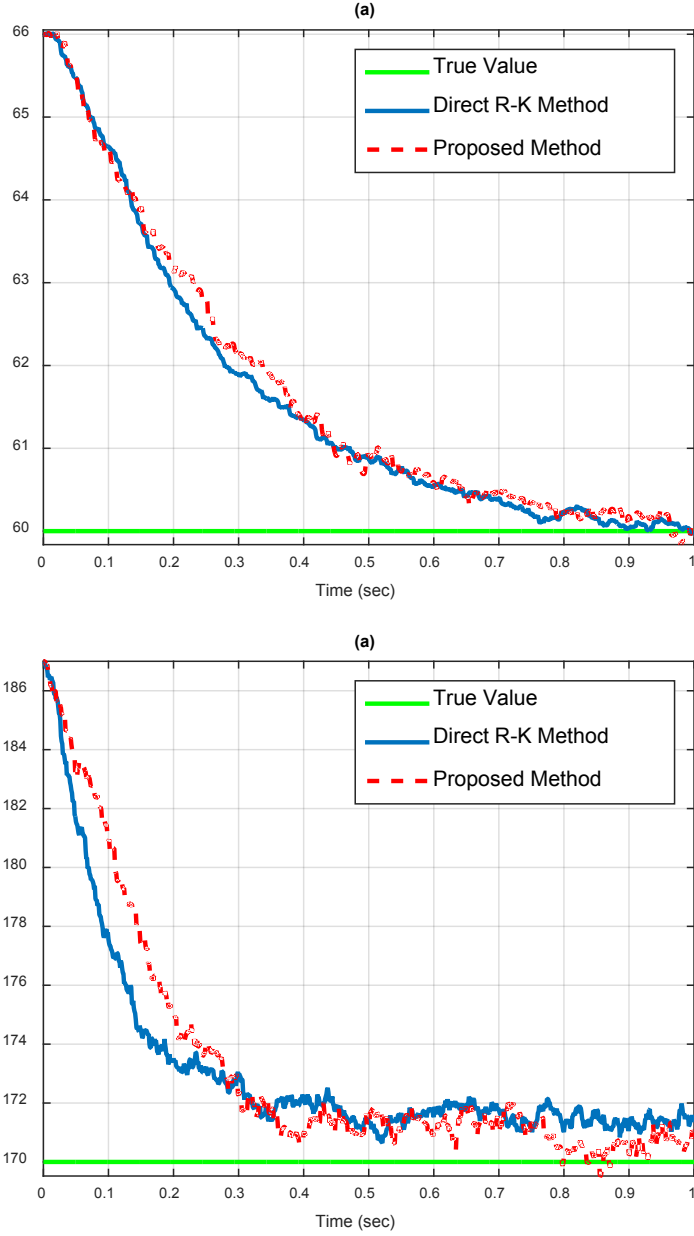


Figure 3.7: Comparison between the proposed ANF method (3.17) with  $m = 4$  and Runge-Kutta method of order 4 on frequency estimation in uniform sampling setting. In this simulation, the input  $y$ , the sampling procedure, the notch depth and adaption speed are the same as in Figures 3.2 and 3.3, except that the digital frequency  $f_0 := \theta_0/(2\pi)$  of the input signal is 60Hz for the left figure and 170Hz for the right figure.

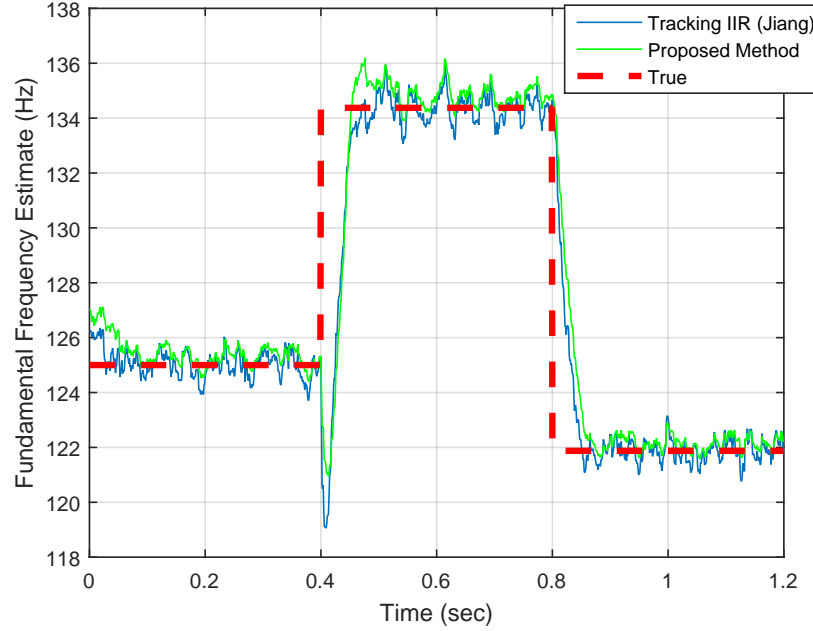


Figure 3.8: The sampling procedure (3.6) is taken uniformly at  $T_n = .001n, 0 \leq n \leq 1200$ , seconds with additive noise  $w_n$  having SNR= 18dB. The frequency changes abruptly after every .4 seconds. The parameters tuned are  $\gamma = 0.005, \xi = .14$  and the ANF is initialize to frequency = 127 Hz. As it is clear that even in purely uniform settings the proposed method which is based on continuous dynamic system has similar speed and accuracy of frequency estimation as the purely discrete model based ANF proposed in [79].

Finally we compare our technique with the ANF frequency estimation and tracking proposed in [79]. As seen from Figure 3.8, using the same signal model as described in (3.26), our method achieves comparable convergence results in purely uniform sampling settings. However, there are several fundamental differences:

- The design of the ANF in [79] is based on a discrete model of filter equations with uniform sampling, while our proposed ANF method works in both uniform and nonuniform sampling setting.
- The design of the ANF in [79] is based on gradient descent (LMS) algorithm of the error

function, while our proposed solution is derived from Lyapunov stability analysis and convergence of associated differential equation. The later approach approach is argued as a superior technique in [40] and [68].

- The ANF in [79] has only one control parameter, while there are three parameters, the frequency parameter  $\theta$ , the response tuning parameter  $\lambda$ , and the frequency rejection parameter  $\xi$  in our approach. So the ANF in [79] is restricted and it can only estimate the harmonic frequencies, while our approach works for nonharmonic frequency estimation also.
- However there is one advantage due to such simplifying assumption. The ANF in [79] converges to the global minimum, while our proposed solution is proved to have local convergence only.

### 3.2.3 Approximation error

Selection of the approximation order  $m$  in the proposed ANF method (3.17) depends on the sampling rate, the input signal frequency, the additive noise, and the sampling noise level. In this section, we discuss the problem how to choose the approximation order  $m$  for mid-range frequency estimation, and consider a combination of the proposed solution with heterodyning technique for high-range frequency estimation. These frequency ranges are specific to the performance of the proposed ANF algorithm, see section 3.2.4 for the statistical characterization.

By (3.10), the discrete ANF algorithm (3.17) has approximation error in its discretization being dominated by a multiple of  $A(\theta_0 \max_n \Delta T_n)^{m+1}$  for the sinusoidal input  $y = A \sin(\theta_0 t + \phi)$  with unknown frequency  $\theta_0$ . Thus, to achieve better accuracy in the frequency estimation (3.17), one should select the higher approximation order  $m$  in the proposed ANF method (3.17).

Observe that

$$\sin(t + \phi) \approx \begin{cases} (1 - \frac{t^2}{2!}) \sin \phi + t \cos \phi & \text{if } |t| \leq \pi/4 \\ (1 - \frac{t^2}{2!}) \sin \phi + (t - \frac{t^3}{3!}) \cos \phi & \text{if } |t| \leq \pi/3 \\ (1 - \frac{t^2}{2!} + \frac{t^4}{4!}) \sin \phi + (t - \frac{t^3}{3!}) \cos \phi & \text{if } |t| \leq \pi/2 \end{cases}$$

for all phases  $\phi$ , we may select the approximation order  $m$  in the algorithm (3.17) as follows:

$$m = \begin{cases} 2 & \text{if } 0 \leq \theta_0 \max_n \Delta T_n \leq \pi/4 \\ 3 & \text{if } 0 \leq \theta_0 \max_n \Delta T_n \leq \pi/3 \\ 4 & \text{if } 0 \leq \theta_0 \max_n \Delta T_n \leq \pi/2. \end{cases} \quad (3.27)$$

So for estimating frequency of a sinusoid input via the algorithm (3.17) of order  $m$ , we need roughly 8 samples per period for  $m = 2$ , 6 samples for  $m = 3$ , and 4 samples for  $m = 4$ , respectively. Our simulations indicate that for an excitation sinusoidal input with its frequency  $\theta_0/(2\pi)$ Hz larger than  $1/8$  of sampling rate  $1/\max_n \Delta T_n$ , it is more reasonable to choose high order approximation in the algorithm (3.17) for its convergence and robustness. For sinusoid input  $y$  with low frequency (i.e.,  $\theta_0 \max_n \Delta T_n$  is not large), a low order approximation in the algorithm (3.17) can be used, while at high frequencies (i.e.,  $\theta_0 \max_n \Delta T_n$  is large), we observe that  $D^3(\theta^2 y)$  of the rescaled input  $\theta^2 y$  used in the (3.4) could be dominated by the noise, see Figure 3.9. From our simulations, we may conclude that the discrete ANF method (3.17) is effective to estimate mid-range frequencies of the input signal which in turn depends on the sampling rate, additive noise and the sampling noise.



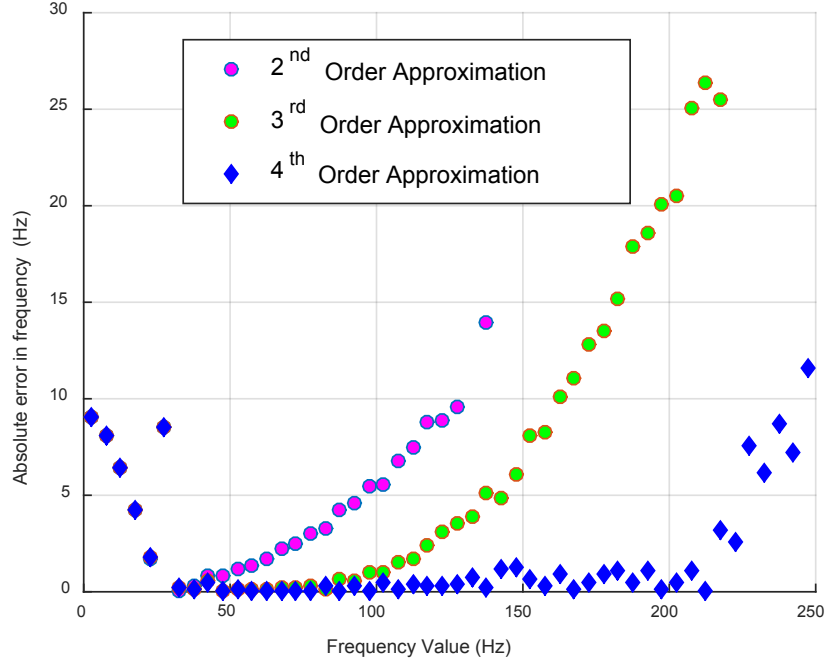


Figure 3.9: Frequency estimation of sinusoidal inputs  $y = \sin(\theta_0 t + \pi/2)$  with frequencies  $\theta_0/(2\pi)$ Hz via the algorithm (3.17) with  $m = 2$  (squared magenta),  $m = 3$  (circled green),  $m = 4$  (diamond shaped blue) respectively. In the simulation, additive jittered sampling is same as shown in Figure 3.2. additive noise has SNR = 5dB, and the adaption speed  $\gamma$  and notch depth  $\xi$  are 0.001 and 0.15 respectively.

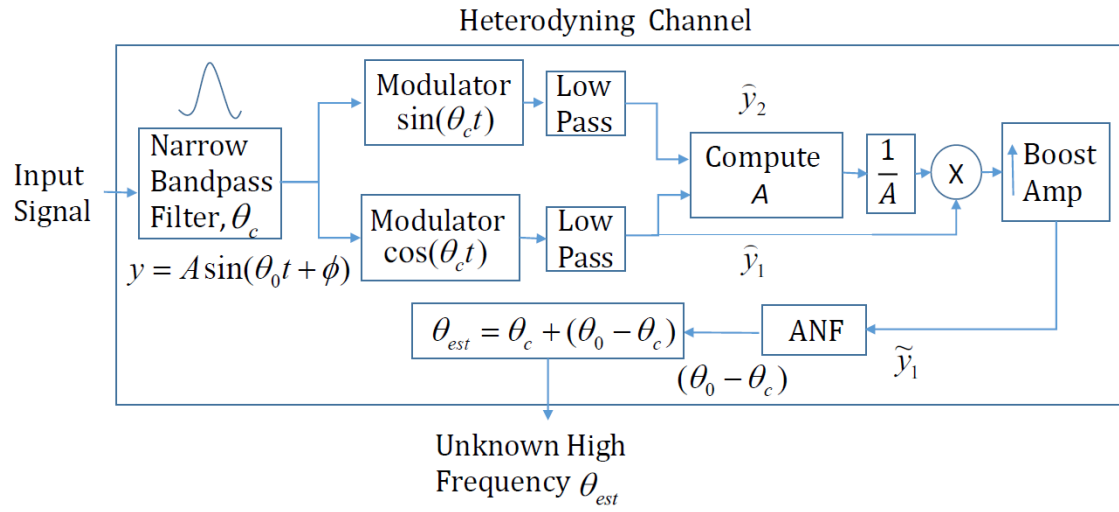


Figure 3.10: Block diagram for high frequency estimation

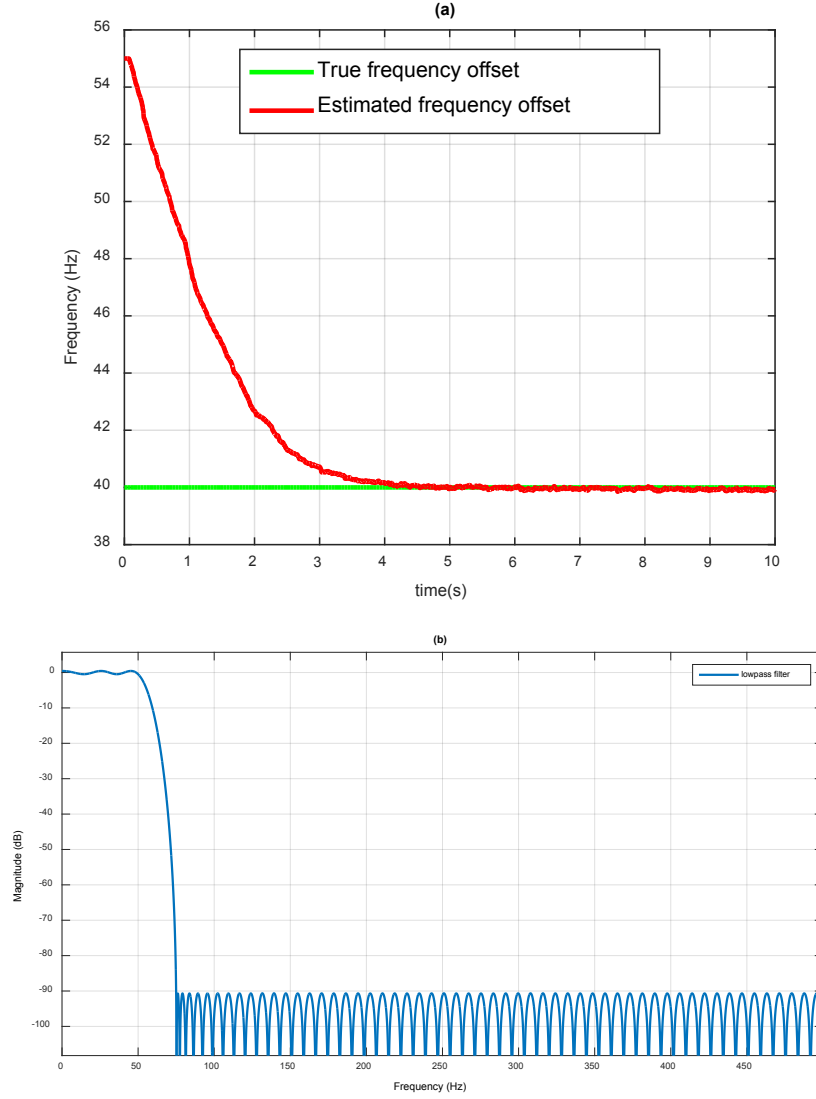


Figure 3.11: Heterodyning ANF method for high-frequency estimation. In the simulation, the sinusoid input  $y$ , the sampling procedure, the notch depth and adaption speed are the same as in Figure 3.3 except that the frequency  $f_0 = \theta_0/(2\pi)$  of  $y$  is 450Hz, modulator frequency  $f_c = 490$ Hz, and frequency initialization for ANF is 55Hz. Presented on the right is a lowpass filter using the generic inbuilt MATLAB function `fdesign.lowpass` with specifications,  $F_p = 50$ Hz (passband),  $F_{st} = 75$ Hz (stopband),  $A_p = 1$ ,  $A_{st} = 90$  (attenuation in dB) with specifications  $F_p = 50$ Hz (passband),  $F_{st} = 75$ Hz (stopband),  $A_p = 1$ dB and  $A_{st} = 90$ dB.

When the input signal has high frequency, we can increase the sampling frequency  $1/\Delta T$  and that allows us to estimate higher frequency with rapidly sampled data. However when the sampling frequency cannot be increased, we propose to combine the discrete ANF method (3.17) with the heterodyning technique ([94]). Take the input excitation signal  $y(t) = A \sin(\theta_0 t + \phi)$  with unknown high frequency  $\theta_0$ , and a modulator frequency  $\theta_c$  with  $\theta_0 - \theta_c$  in the effective range of our technique (3.17). Applying sine and cosine modulators with frequency  $\theta_c$  to the signal  $y$  gives

$$\begin{aligned} y_1(t) &= y(t) \times \sin(\theta_c t) \\ &= \frac{A}{2} \cos((\theta_0 - \theta_c)t + \phi) - \frac{A}{2} \cos((\theta_0 + \theta_c)t + \phi) \end{aligned}$$

and

$$\begin{aligned} y_2(t) &= y(t) \times \cos(\theta_c t) \\ &= \frac{A}{2} \sin((\theta_0 - \theta_c)t + \phi) + \frac{A}{2} \sin((\theta_0 + \theta_c)t + \phi). \end{aligned}$$

To eliminate high-frequency component,  $\theta_0 + \theta_c$ , of the modulated signals  $y_1$  and  $y_2$ , we apply low pass filter such that only the low frequency component  $\theta_c - \theta_0$  passes through, which generates

$$\hat{y}_1 \approx \frac{A}{2} \cos((\theta_0 - \theta_c)t + \phi)$$

and

$$\hat{y}_2 \approx \frac{A}{2} \sin((\theta_0 - \theta_c)t + \phi).$$

As

$$\sqrt{(2\hat{y}_1)^2 + (2\hat{y}_2)^2} \approx A,$$

we re-normalize  $\hat{y}_1$  by

$$\tilde{y}_1 = \frac{\hat{y}_1}{\sqrt{(2\hat{y}_1)^2 + (2\hat{y}_2)^2}}, \quad (3.28)$$

and use  $\tilde{y}_1$  to replace  $y$  as the input sinusoid signal of our algorithm (3.17), see Figure 3.10 for block diagram of the combination of heterodyning and the discrete ANF method.

In the above heterodyning ANF method for high-frequency estimation, we again need some a priori information on frequency  $\theta_0$  of the input signal so that we can select a proper modulator frequency  $\theta_c$  and construct band pass filter blocks to eliminate high-frequency components of modulated signals. This restricts the application of the heterodyning ANF method to the uniform sampling setting only. On the other hand, our signal renormalization procedure (3.28) makes our heterodyning ANF method almost independent of amplitude of the input signal, which increases the robustness of our procedure. We did simulations with a number of different frequencies in the range 200 – 450Hz and obtained similar estimation results, see Figure 3.11.

### 3.2.4 Statistical Characterization

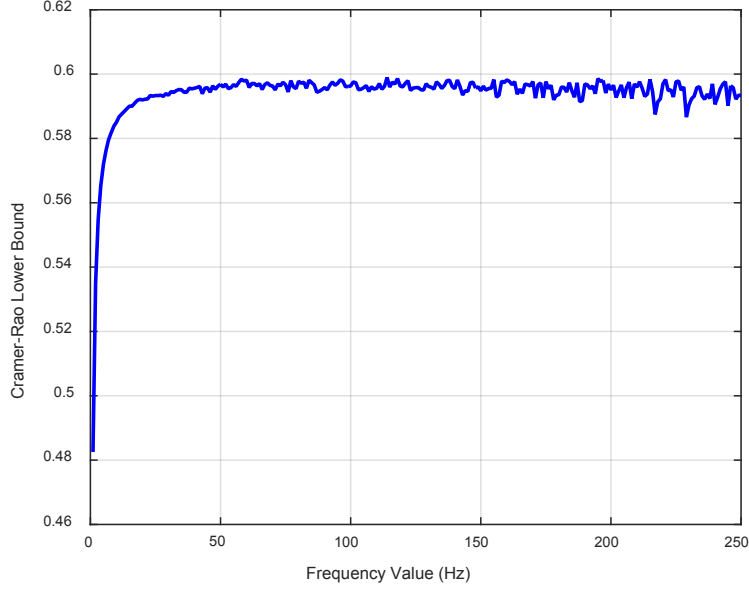


Figure 3.12: For jittered sampling  $(n + r_n)/1000, 0 \leq n \leq 1000$ , seconds of sinusoidal inputs  $y(t) = \sin(\theta_0 t + \pi/2)$  with frequencies  $\theta_0/(2\pi)$  Hz, CRLB is about  $6(\sigma/A)^2 a^{-2} N^{-1} = 0.6$ , where  $r_n \in [-1/8, 1/8]$  are randomly selected and the signal-noise ratio  $SNR := 10\log_{10}(\sigma/A)^2$  is 20.

In this subsection, we compare variance of frequency estimation in the discrete ANF method (3.17) with the Cramer-Rao lower bound (CRLB) for the jittered sampling model (3.23). The statistical property will be useful to determine the range of effective frequencies for the proposed ANF, and also the range of frequencies in which the performance of the proposed ANF degrades.

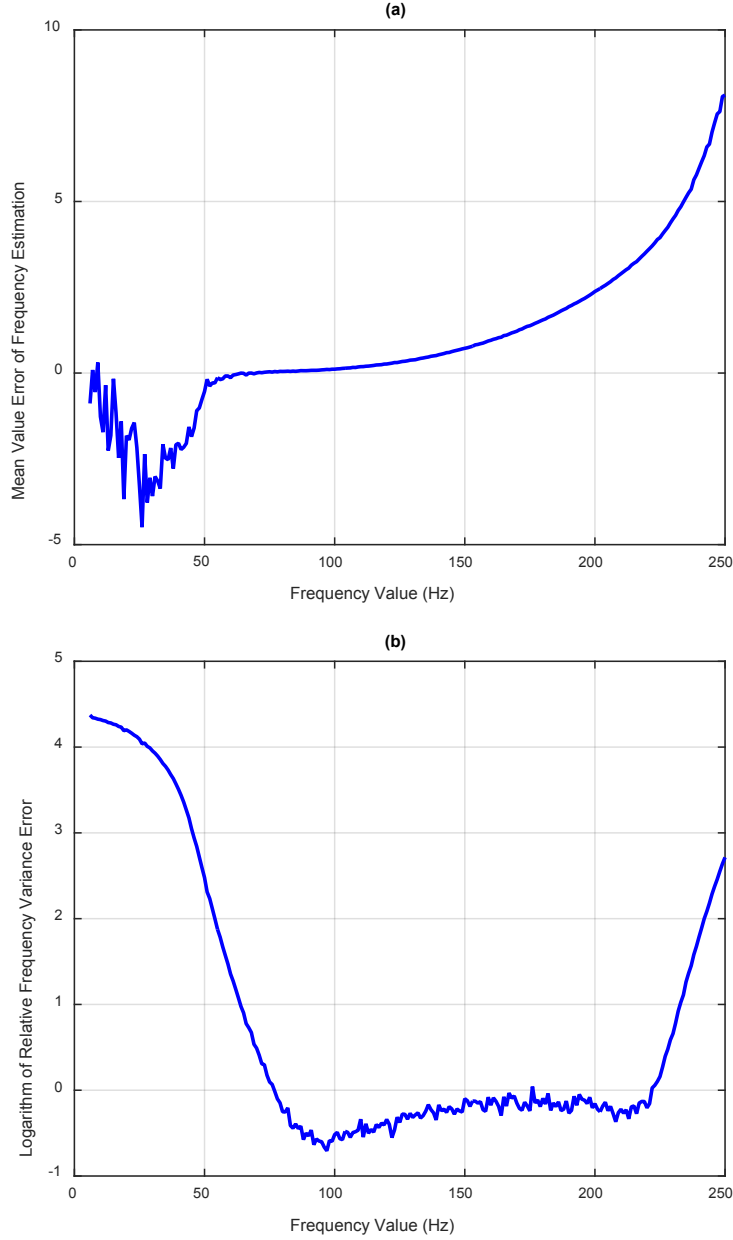


Figure 3.13: Listed on the left is the mean value error between the estimated frequency  $\theta/(2\pi)$ Hz in the algorithm (3.17) with  $m = 4$  and the true frequency  $\theta_0/(2\pi)$ Hz of the sinusoid input, while on the right is the common logarithm between the variance of the estimated frequency in the algorithm (3.17) and the Cramer-Rao low bound. In the algorithm (3.17), the jittered sampling set, SNR of the additive noise, the adaption speed and the notch depth are the same as in Figure 3.9.

For the additive white Gaussian noise (AWGN)  $w_n$  in (3.3) with variance  $\sigma^2$ , the corresponding CRLB  $\text{var}(\hat{\theta})$  for any unbiased frequency estimator  $\hat{\theta}$  depends on the frequency  $\theta_0$  of the excitation sinusoidal input, and it decreases when more sampling data are available ([95], [96]). In fact,

$$\begin{aligned}
\text{var}(\hat{\theta}) &\geq \frac{\sigma^2}{\sum_{n=0}^{N-1} \left( \frac{\partial y(T_n)}{\partial \theta} \right)^2} \\
&= \frac{(\sigma/A)^2}{\sum_{n=0}^{N-1} T_n^2 \cos^2(T_n \theta_0 + \phi)} \\
&= \frac{2(\sigma/A)^2}{A_N + B_N \cos \phi - C_N \sin \phi} \\
&\geq \frac{2(\sigma/A)^2}{A_N + \sqrt{(B_N)^2 + (C_N)^2}}, \tag{3.29}
\end{aligned}$$

where

$$A_N = \sum_{n=0}^{N-1} T_n^2, \quad B_N = \sum_{n=0}^{N-1} T_n^2 \cos(2T_n \theta_0)$$

and

$$C_N = \sum_{n=0}^{N-1} T_n^2 \sin(2T_n \theta_0).$$

From the above estimate (3.29), we see that the CRLB  $\text{var}(\hat{\theta})$  may grow without bounds when  $T_N \theta_0$  is sufficiently small, c.f. Figure 3.9, and it is proportional to the signal-noise-ratio  $\sigma^2/A^2$ .

For the uniform sampling between 0 and  $a$  seconds (i.e.,  $T_n = na/N$  seconds for  $0 \leq n \leq N-1$ ) and the sinusoidal input with positive frequency  $\theta_0$ , the CRLB  $\text{var}(\hat{\theta})$  is asymptotically proportional to reciprocal of the sampling number  $N$ ,

$$\text{var}(\hat{\theta}) \geq C_0 \frac{(\sigma/A)^2}{N} \tag{3.30}$$



for some positive constant  $C_0$ , because

$$\lim_{N \rightarrow \infty} \frac{A_N}{N} = \frac{a^2}{3},$$

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{B_N}{N} &= a^2 \int_0^1 x^2 \cos(2a\theta_0 x) dx \\ &= a^2 \rho_0^{-3} ((\rho_0^2 - 2) \sin \rho_0 + 2\rho_0 \cos \rho_0), \end{aligned}$$

and

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{C_N}{N} &= a^2 \int_0^1 x^2 \sin(2a\theta_0 x) dx \\ &= a^2 \rho_0^{-3} (-2 + (2 - \rho_0^2) \cos \rho_0 + 2\rho_0 \sin \rho_0), \end{aligned}$$

where  $\rho_0 = 2a\theta_0$ . Thus for the uniform sampling between 0 and  $a$  seconds and the sinusoidal input with large frequency  $\theta_0$ , the CRLB  $\text{var}(\hat{\theta})$  is about  $6(\sigma/A)^2 a^{-2} N^{-1}$ , see Figure 3.12.

For a nonuniform sampling between 0 and  $a$  seconds with  $T_0 = 0, T_N = a$  and

$$\max_{1 \leq n \leq N} T_n - T_{n-1} \leq \frac{c}{N}$$

for some  $c > 0$ , we have

$$\frac{A_N}{N} \geq c^{-1} \sum_{n=1}^{N-1} T_n^2 (T_{n+1} - T_n) \rightarrow c^{-1} \int_0^a t^2 dt,$$

$$\frac{B_N}{N} \geq c^{-1} \int_0^a x^2 \cos(2\theta_0 x) dx,$$

and

$$\frac{C_N}{N} \geq c^{-1} \int_0^a x^2 \sin(2\theta_0 x) dx$$

as  $N \rightarrow \infty$ . Hence the asymptotic estimate (3.29) for the CRLB bound  $\text{var}(\hat{\theta})$  holds for the above nonuniform setting, including the jittered nonuniform sampling in (3.23) and the additive nonuniform sampling in (3.24).

Due to discretization, the frequency estimate obtained by our proposed method (3.17) is biased, but the systematic error is less than 1%Hz when the input sinusoid has around mid-range frequencies from our simulations, c.f. Figure 3.5. The variance of frequency estimate is also comparable to the CRLB when the input sinusoid has mid-range frequency, see Figure 3.13.

### 3.2.5 Extension

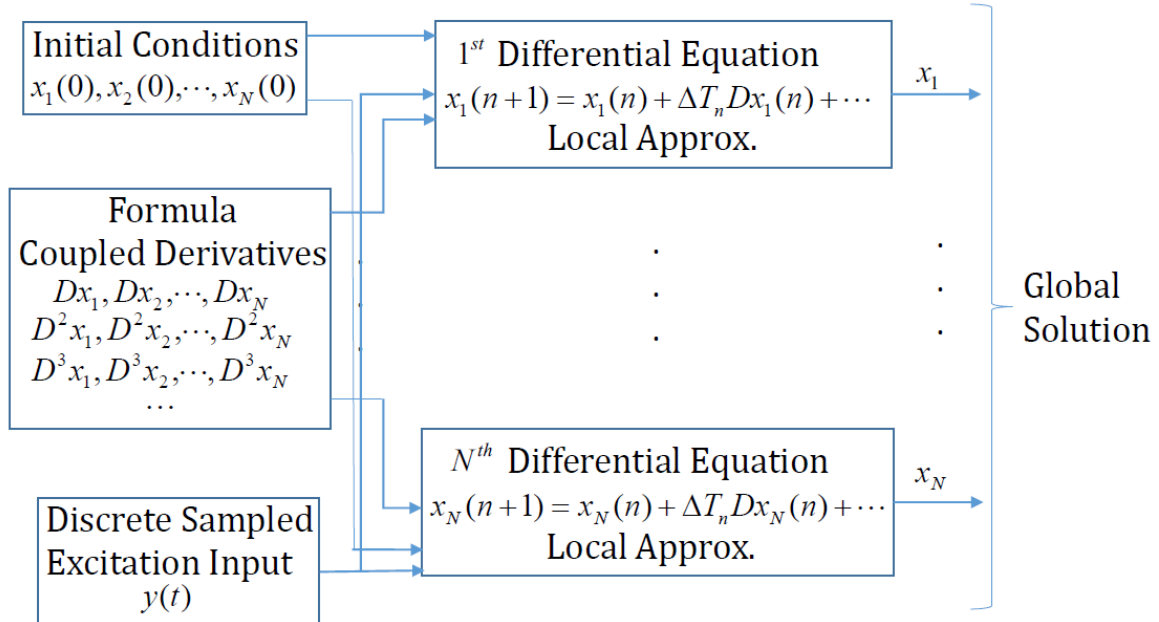


Figure 3.14: Block diagram for discretization a dynamic system for frequency estimation.

In this subsection, we extend the proposed discretization technique in section 3.2.1 to continuous dynamic systems for frequency estimation with high accuracy. We apply the extended discretization technique to adaptive notch filtering developed by Mojiri, Xia, Morino, Hou and their collaborators ([69, 70, 71, 72, 73, 74]), and compare their performance on frequency estimation of a sinusoidal input.

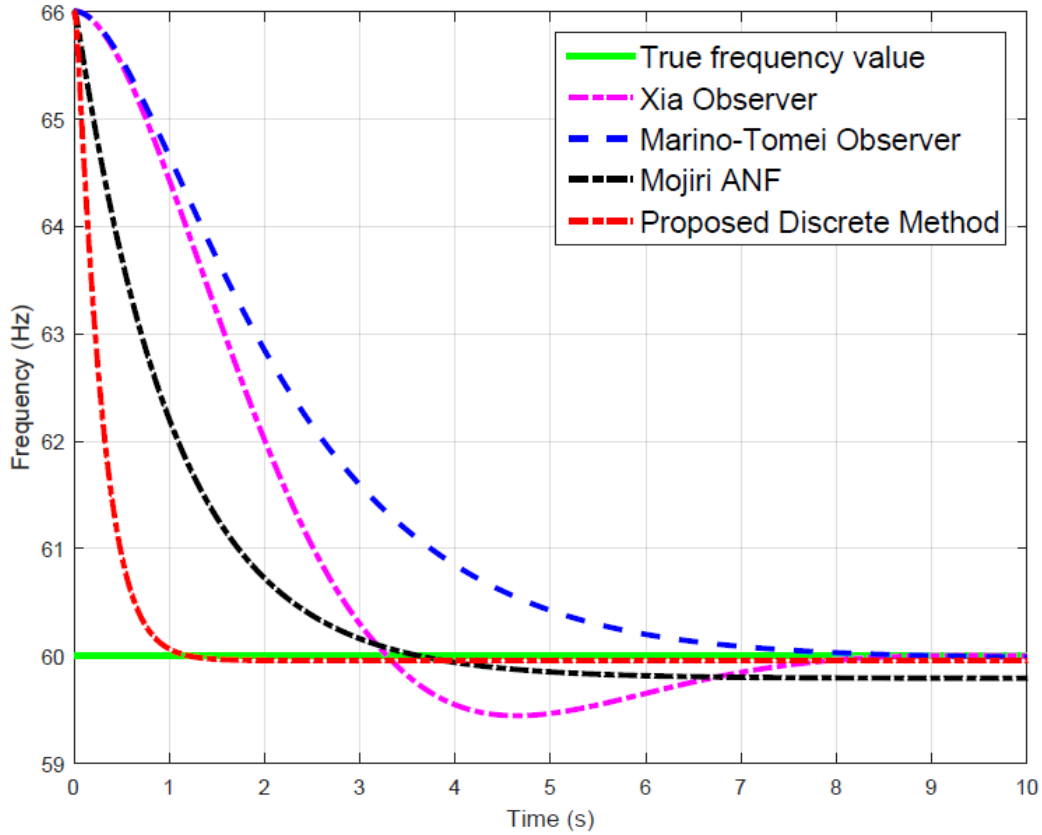


Figure 3.15: In this simulation, the input  $y$  in (3.5) has amplitude  $A = 1$ , frequency  $\theta_0 = 120\pi$  and phase  $\phi = \pi/3$ , the sampling procedure (3.6) is taken uniformly at  $T_n = .001n, 0 \leq n \leq 1000$ , seconds without additive noise  $w_n$ , the approximation order  $m$  in (3.17) is 4 and initial frequency is about 66Hz or with 10% error. Also we set  $\xi = .15$  and  $\gamma = .001$  in (3.4) and (3.33),  $b = .3, k_1 = k_2 = .5, \gamma = .6$  in (3.34) and  $\lambda = .4, k_1 = .5, \gamma = .6$  in (3.35).

Consider a dynamic system

$$D\mathbf{X} = \mathbf{F}(t, \mathbf{X}, y), \quad (3.31)$$

where  $\mathbf{X} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$  and  $\mathbf{F} : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^n$  is a real analytic function. By the *Cauchy-Kowalevskaya* theorem ([90]), the solution  $\mathbf{X}(t)$  of the dynamic system (3.31) is analytic and it can be approximated by Taylor polynomials of order  $m \geq 0$ ,

$$\mathbf{X}(t) \approx \sum_{k=0}^m \frac{D^k \mathbf{X}(T_n)}{k!} (t - T_n)^k, \quad T_n \leq t \leq T_{n+1}, \quad (3.32)$$

when maximal sampling gap  $\max_n(T_{n+1} - T_n)$  is small relative to the changing or stationary signal frequency. In the case that derivatives  $D^k y, 1 \leq k \leq m$ , of the input signal  $y$  are well approximated at sampling times  $T_n, n \geq 0$ , we can discretize the dynamic system (3.31) as we did in (3.17) for the system (3.4), see Figure 3.14 for the block diagram.

There are several variations for the ANF system (3.4). Mojiri and his collaborators decoupled the notch depth  $\xi$  and adaption speed  $\gamma$  in (3.4) ([71, 73, 74]), and introduced the following dynamic system,

$$\begin{cases} Dx_1 = x_2 \\ Dx_2 = -2\xi\theta x_2 - \theta^2 x_1 + 2\xi\theta^2 y \\ D\theta = -\gamma(\theta^2 y - \theta x_2)x_1, \end{cases} \quad (3.33)$$

which has a unique periodic orbit,

$$[x_1, x_2, \theta]^T = [-A \cos(\theta_0 t + \phi), A\theta_0 \sin(\theta_0 t + \phi), \theta_0]^T$$

for the sinusoidal input  $y = A \sin(\theta_0 t + \phi)$ . An additional constraint for stability is  $\xi > 1$ , hence the notch depth cannot be made very narrow to reject other frequencies unlike the system (3.4).

In [70], Xia observed the similarity between adaptive notch filters and adaptive identifiers, and then he introduced a frequency estimator of fourth order,

$$\begin{cases} Dx_1 = -bx_1 - y \\ Dx_2 = bx_1x_4 + k_1(y - x_3) \\ Dx_3 = x_2 + x_1x_4 + k_2(y - x_3) \\ Dx_4 = \gamma x_1(y - x_3), \end{cases} \quad (3.34)$$

where  $b, k_1, k_2, \gamma > 0$ . The above dynamic system has globally exponential convergence and its fourth state  $x_4$  converges to squared frequency  $\theta_0^2$  of the sinusoidal input  $y = A \sin(\theta_0 t + \phi)$ . Similar frequency estimator of fourth order was introduced by Marino and Tomei [69] independently,

$$\begin{cases} Dx_1 = -\lambda x_1 - y \\ Dx_2 = x_3 + \lambda y + x_1x_4 + k_1(y - x_2) \\ Dx_3 = -\lambda x_3 - \lambda^2 y \\ Dx_4 = \gamma x_1(y - x_2), \end{cases} \quad (3.35)$$

where  $\lambda, \gamma > 0$  and  $k_1 > 1/(4\lambda)$ .

For the above dynamic systems (3.33), (3.34) and (3.35) we may apply similar technique used in (3.16) to approximate  $D^k y, 1 \leq k \leq m$ , of the input signal  $y$  at sampling times  $T_n, n \geq 0$  or for the uniform sampling we can use Runge-Kutta method directly also. This leads to discretization of those dynamic systems, which is similar to the one (3.17) based on the system (3.4). Our simulations show that the solution of those dynamic systems (3.33), (3.34) and (3.35) provide the frequency estimate of a sinusoid input  $y$ , but comparing with the proposed method based on the dynamic system (3.4), they have slower convergence, see Figure 3.15.

### 3.3 Multiple Frequency Estimation

In this section, we introduce two modifications of the discrete ANF technique (3.17) and discuss their performance for multiple frequency estimation of a signal in (3.2) from its noisy nonuniform samples (3.3). We remark that once all frequencies of the signal are retrieved, its amplitudes and phases could be evaluated by finding least squares estimates [39] or directly using the states of some dynamic system by (3.7).

For multiple frequency estimation, adaptive dynamic systems that try to estimate all frequencies at the same time, have been proposed but their formulations are very complicated for increasing number of frequencies [83]. As an ideal notch filter has unit gain at all frequencies except at notch frequency where its gain is zero, we may use simple configurations of ANFs tracking only one frequency per channel. Such a configuration could be implemented by applying the discretized ANF technique (3.17) with different initial frequencies, and then approximating the input signal from its noisy samples. If the initial frequencies can be chosen in around the true frequencies, the algorithm (3.17) may lead to accurate approximation of true frequencies of the input signals, but it might only track the dominant frequency (higher amplitude) or the lower frequency of the input signal (since we have higher accuracy estimation for components with lower frequency) effectively, and hence the other frequencies could not be retrieved, see Figure 3.16.

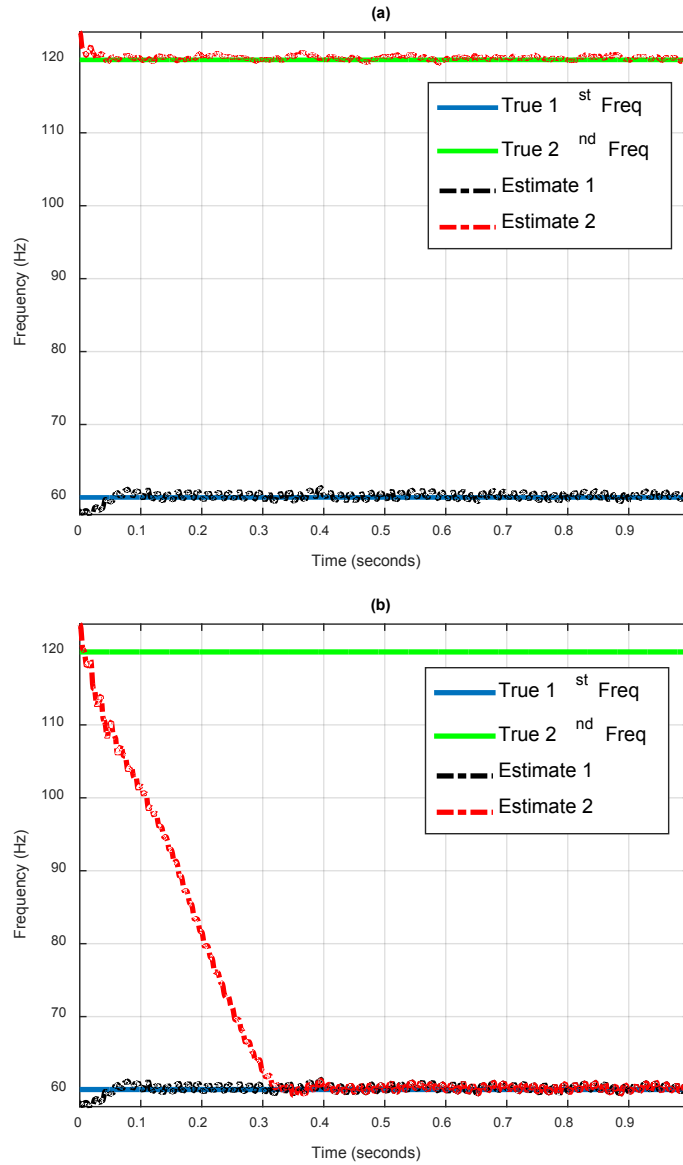


Figure 3.16: Frequency retrieval via discrete ANF method (3.17). In the simulation, the sampling procedure, the notch depth and adaption speed are the same as in Figures 3.2 and 3.3, two initial frequencies selected are  $f_{c1} = 56\text{Hz}$  and  $f_{c2} = 125\text{Hz}$ , and the input signal  $y = A_1 \sin(2\pi f_1 t + \phi_1) + A_2 \sin(2\pi f_2 t + \phi_2)$  has phases  $\phi_1 = \pi/2$  and  $\phi_2 = \pi/7$ , frequencies  $f_1 = 60\text{Hz}$  and  $f_2 = 120\text{Hz}$ , and amplitudes  $A_1 = 1$ ,  $A_2 = 2$  (left) and  $A_2 = 1/2$  (right). In this simulation, the dominated frequency of the input signals are 120 Hz for the left figure and 60 Hz for the right figure respectively.

The above observation on the frequency retrieval inspires us to introduce the following cascade ANF method:

- First find a notch frequency  $\theta_1$  (mostly the dominant one or the lower frequency) of the input signal by applying the discretized ANF method (3.17) or its combination with heterodyning.
- Then extract the sinusoid  $y_1 = A_1 \sin(\theta_1 t + \phi_1)$  with that frequency  $\theta_1$  from the signal  $y$ . The sinusoid  $y_1 \approx 2\xi x_1/\theta_1$  by (3.7), where  $x_1$  is the first state of the system (3.17).
- Next, remove that sinusoid from the signal and pass the new signal  $y - \hat{y}_1$  to the next stage.
- Finally, repeat the above procedure until the residual error is below certain threshold, or all sinusoids in the input signal are extracted.

The block diagram of the above cascade procedure is presented in Figure 3.17, c.f. [42] for similar configuration in audio signal processing.



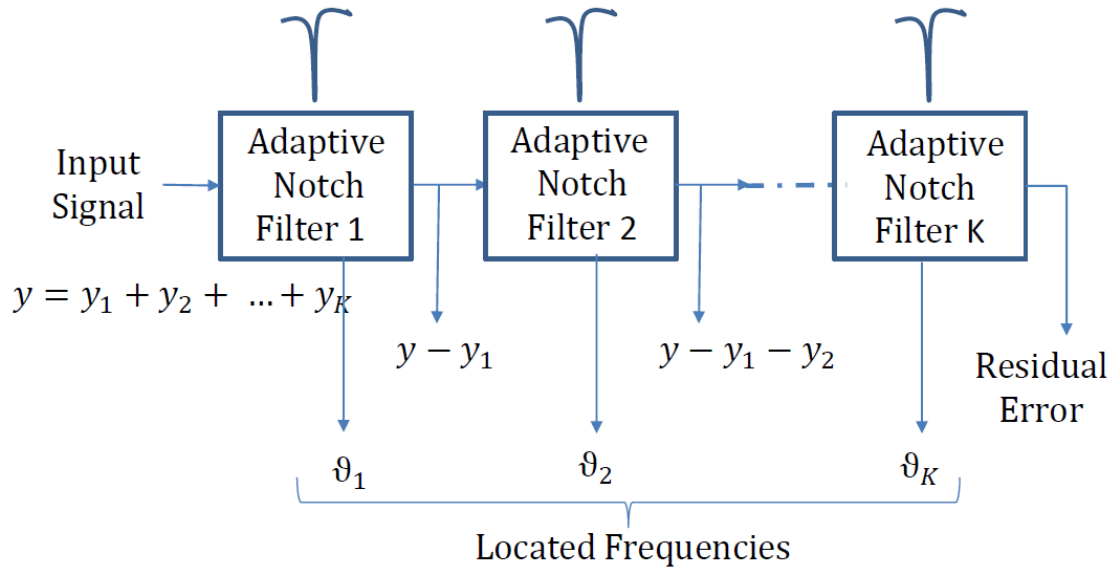


Figure 3.17: Block diagram of cascade procedure for multiple frequency estimation

The cascade procedure is attractive, as applying the discretized ANF technique (3.17) in each stage may retrieve one of the sinusoids, and the remaining stage would have no choice but to seek other sinusoids [75]. However, the disadvantages with the cascade procedure include the distortion introduced by the discretized ANF technique (3.17) at each cascade stage, and the time delay that is introduced due to processing and retrieval of each sinusoidal component. This could render online estimation impractical if the signal  $y$  is composed of many frequencies.

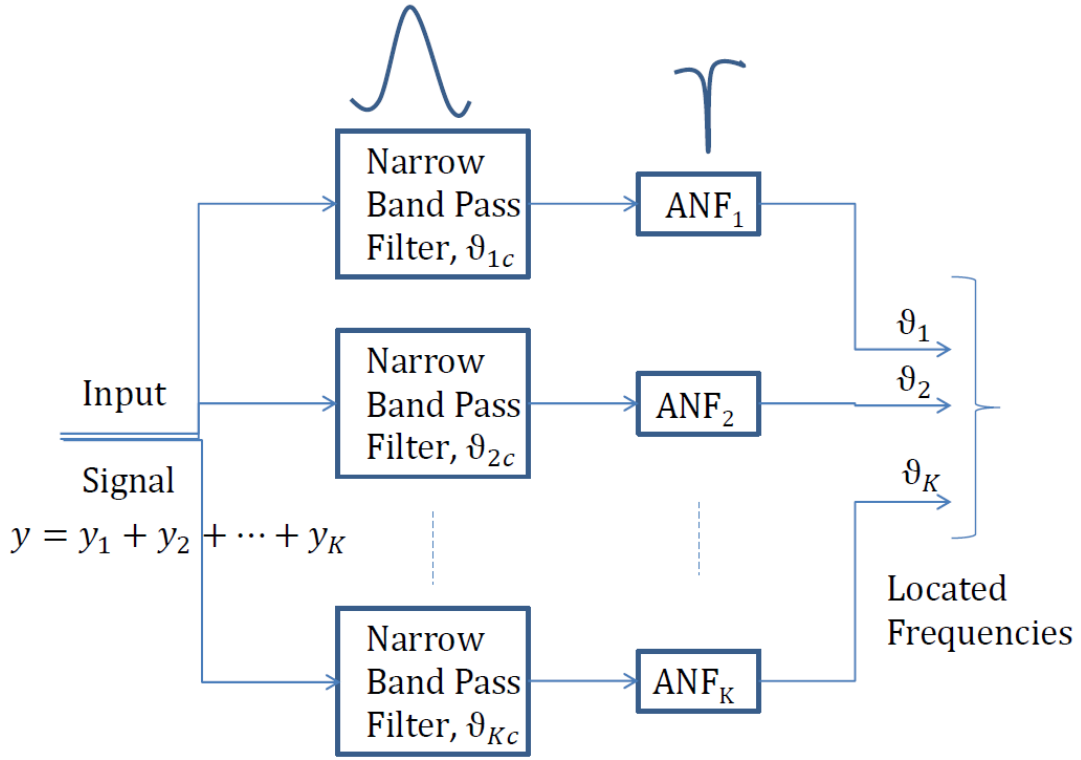


Figure 3.18: Block diagram of discrete ANF method with prefiltering.

To avoid the distortion introduced the discretized ANF technique (3.17) at each cascade stage, we propose an alternative approach to combine our discrete ANF technique (3.17) with band pass prefiltering:

- First, apply band pass filters with some a priori chosen central frequencies,  $\theta_{1c}, \theta_{2c}, \dots, \theta_{Kc}$ , to the samples of the input signal for initial spectral selection.
- Then, use the discretized ANF technique (3.17) for the filtered samples to obtain the true frequencies around the band pass frequencies independently.

The block diagram for parallel implementation of the above discrete ANF technique with prefiltering is presented in Figure 3.18, c.f. adaptive filtering in [97] for multi-frequency estimation.

By introducing narrow bandpass filters to do the initial spectral selection, we effectively eliminate distortion of the cascade procedure in each stage, while we still embrace the benefits of the parallel implementation for fast sinusoid retrieval.

For uniform sampling procedure, we may design narrow bandpass filters directly (using MATLAB `fdesign`) or design filter  $G$  from the complementary notch filters  $H$ , with their transfer functions  $G(z)$  and  $H(z)$  are related by

$$G(z) = 1 - H(z).$$

For instance,  $H$  is a notch filter of second order,

$$H(z) = \frac{1 - 2 \cos(\omega_0)z^{-1} + z^{-2}}{1 - 2r \cos(\omega_0)z^{-1} + r^2 z^{-2}} \quad (3.36)$$

with notch bandwidth parameter  $r \in (0, 1)$  and notch frequency  $\omega_0$ . The drawback of prefiltering ANF method is that narrow band prefiltering is not applicable when only nonuniform sampling of the input signal is available.

The rest of this section is devoted to demonstration of the cascade ANF procedure and the prefiltering ANF procedure to estimate multiple frequencies of the multi-sinusoid signals effectively in various settings.

### 3.3.1 Inputs with two well-separated frequencies

Consider an input signal

$$y = A_1 \sin(2\pi f_1 t + \phi_1) + A_2 \sin(2\pi f_2 t + \phi_2) \quad (3.37)$$

with its two frequencies  $f_1, f_2$  being far away. Our simulations show that both prefiltering ANF and cascade ANF provide estimations close to true frequencies of the input signal  $y$ , see Figure 3.19 and c.f. Figure 3.16.

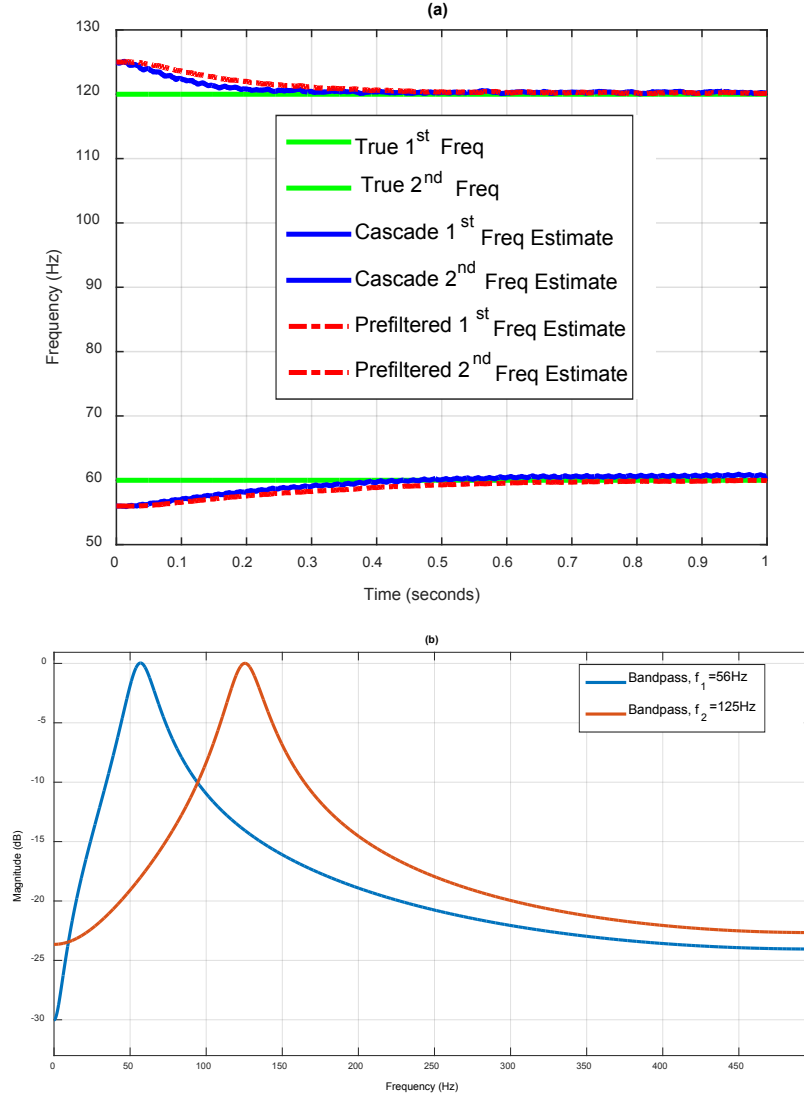


Figure 3.19: Frequency estimation via prefiltering ANF and cascade ANF methods. In this simulation, the sampling procedure, the notch depth and adaption speed, and two initial frequencies selected are the same as in Figure 3.16, and the signal  $y$  in (3.37) has amplitudes  $A_1 = 1$  and  $A_2 = 2$ , phases  $\phi_1 = \pi/2$  and  $\phi_2 = \pi/7$ , and frequencies  $f_1 = 60\text{Hz}$  and  $f_2 = 120\text{Hz}$ . Plotted on the left is frequency estimation via cascade ANF method (in solid blue line) and prefiltering ANF method (in dashed red line), while plotted on the right is narrow band pass prefilter around two initial frequencies being used in the prefiltering ANF method.

For uniform sampling setting, prefiltered configuration is a bit more suitable in terms of both speed and accuracy, but in non-uniform sampling we can effectively use cascade configuration especially when there are small number of frequencies for the input signal.

### 3.3.2 *Input with two very close frequencies, a pathological case*

Consider an input signal  $y$  of the form (3.37) with its two frequencies  $f_1$  and  $f_2$  being close. As those two frequencies cannot be separated using narrow band pre-filters especially in highly noisy sampling environment, prefiltering ANF method does not provide reasonable frequency estimations. So in this subsection, we restrict ourselves to discuss frequency estimation via the cascade ANF method in that pathological case.

In order to reject one frequency from the other, we should select small notch depth  $\xi$  in (3.4). Therefore by (3.8), the adaption speed  $\gamma$  in the dynamic system (3.4) should be small too, which implies that the discrete dynamic system (3.17) has slow convergence and there are long delay to retrieval of true frequencies. Our simulations indicates that it is better to select the first initial frequency in the middle of those two close frequencies if we do not have much information on the dominant frequency, see Figure 3.20 and c.f. Figure 3.16.

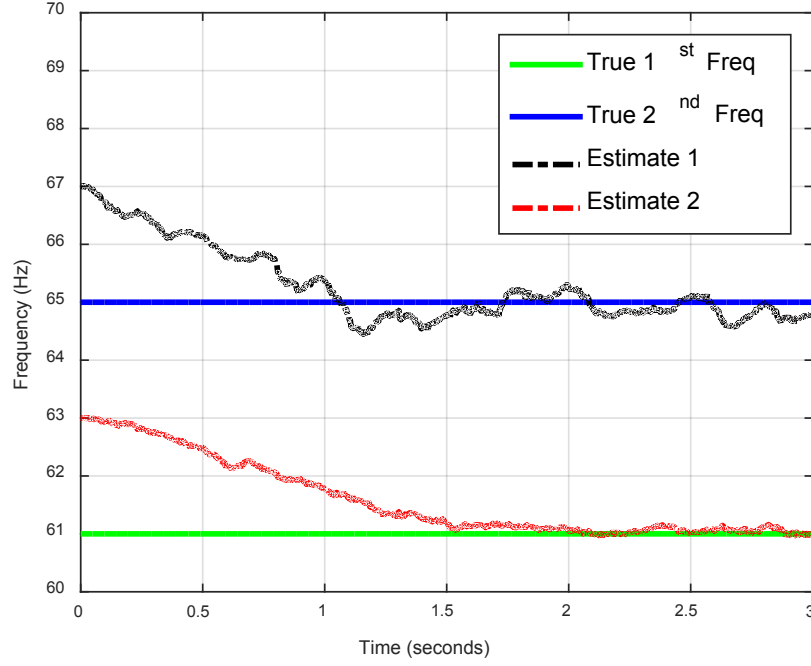


Figure 3.20: Frequency retrieval of inputs with two close frequencies via the cascade ANF method. In this simulation, the input signal  $y$  in (3.37) has frequencies  $f_1 = 61\text{Hz}$  (dominant frequency for the right figure),  $f_2 = 65\text{Hz}$  (dominant frequency for the left figure), phases  $\phi_1 = \pi/2, \phi_2 = \pi/7$ , and amplitude  $A_1 = A_2 = 1$ , the sampling procedure is the same as in Figure 3.19 except sampling time interval is  $[0, 3]$  second, the adaption speed  $\gamma = .00001$  (one hundredth of the one in Figure 3.19), and notch depth  $\xi = .015$  (one tenth of the one in Figure 3.19), the first initial frequency is 63Hz (mean values of  $f_1$  and  $f_2$ ), and 67Hz (about 5% above the non-dominant frequency  $f_2$ ).

Most of known frequency estimations has poor performance for the pathological case, see [89] and references therein. What is unexpected to select slow adaption speed and narrow notch depth is the robustness of our frequency estimation. We did the simulation for noise with SNR=40, 30, 20, 10, 0dB, and in all cases we get the same consistent and smooth estimates. Possible reasons for such phenomenon could be that the ANF has excellent noise rejection capability especially when we make the adaptation speed slow and the filter bandwidth narrow. This could make our cascade ANF method more powerful than the one in [46], where it is required that the signal power must be

greater than the noise power in their methods.

### 3.3.3 Input with multiple harmonic frequencies

Consider an input signal

$$y = \sum_{k=1}^K A_k \sin(\theta_k t + \phi_k) \quad (3.38)$$

with harmonic frequencies  $\theta_k = k\theta_1$ ,  $2 \leq k \leq K$  [39]. For a signal with harmonic frequencies, it suffices to find its fundamental frequency  $\theta_1/(2\pi)$ Hz, which could be obtained by band-pass filtering of the input signal. We observe that the fundamental frequency can be estimated consistently via a single ANF block in a highly noisy environment. Presented in Figure 3.21, c.f. Figure 3.6, is the fundamental frequency estimation of the signal  $y$  in (3.38) with amplitudes  $A_1 = 1, A_2 = A_1/2, A_3 = A_1/3, A_4 = A_1/4, A_5 = A_1/5$ , phases  $\phi_k \in [0, \pi]$  randomly selected, and fundamental frequency  $f_0$  changing abruptly from 72Hz to 60Hz at 1/3 second and from 60Hz to 80Hz at 2/3 second, c.f. signals with time-varying fundamental frequency in power systems [39]. The above observation makes our solution very suitable for time domain signal analysis similar to [98] (which is a CT system), but now we can do it with nonuniformly sampled data in presence of random sampling noise and additive noise.



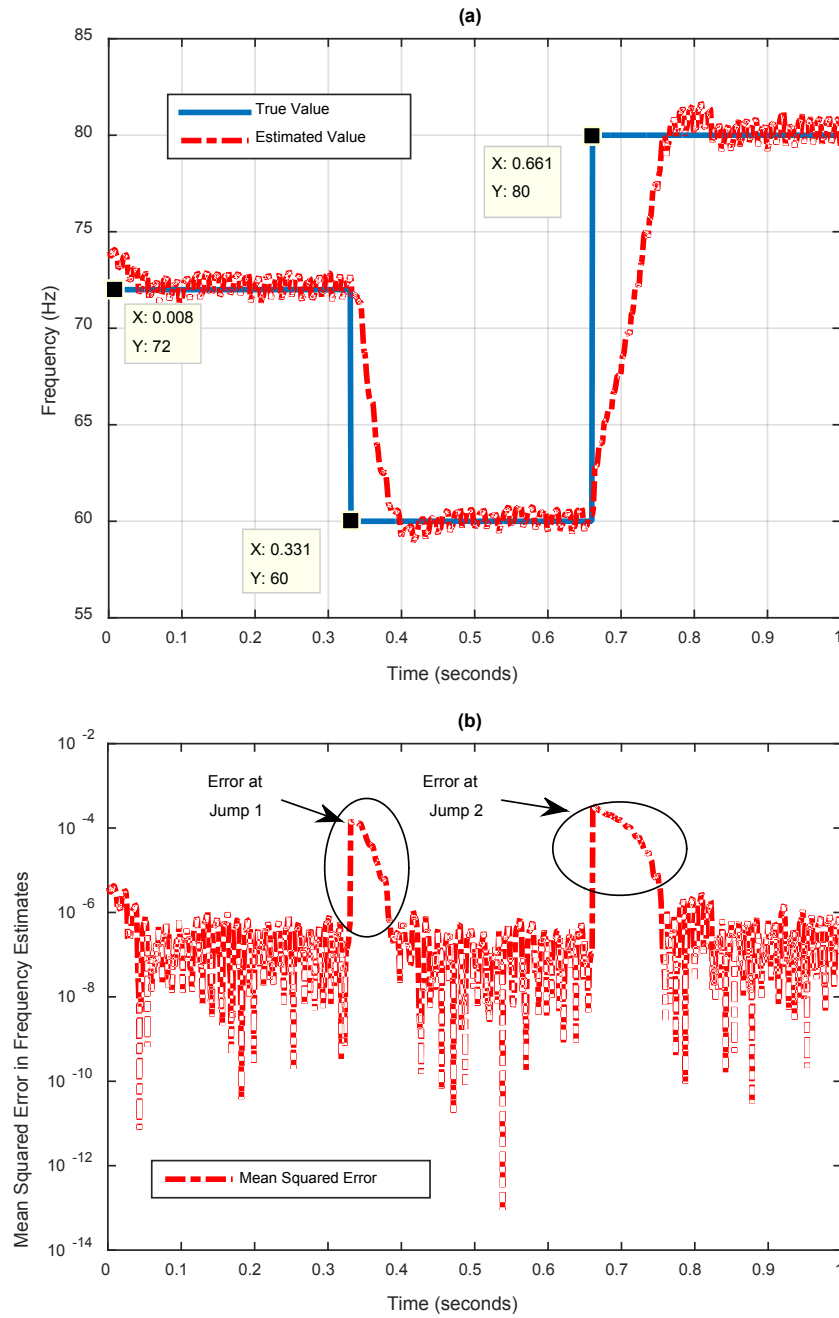


Figure 3.21: Fundamental frequency tracking of an input signal with harmonic frequencies using a single ANF block. In the simulation, the sampling procedure, notch depth, adaption speed and initial frequency are the same as the ones in the Figure 3.6(b).

### 3.3.4 *Input with multiple frequencies*

In this subsection, we consider frequency estimation when input signals (3.2) have multiple frequencies. Due to the distortion by cascade ANF in each iteration, we propose to apply prefiltering ANF method for frequency estimation. Presented in Figure 3.22 is a demonstration for our pre-filtering ANF frequency estimator, where the input signal

$$y = \sum_{k=1}^7 A_k \sin(2\pi f_k t + \phi_k)$$

has amplitudes  $A_k \in [0.5, 1.5]$  and phases  $\phi_k \in [-\pi, \pi]$ ,  $1 \leq k \leq 7$ , being randomly selected, and the actual frequencies are  $(f_1, \dots, f_7) = (78, 137, 177, 234, 288, 325, 437)\text{Hz}$ .

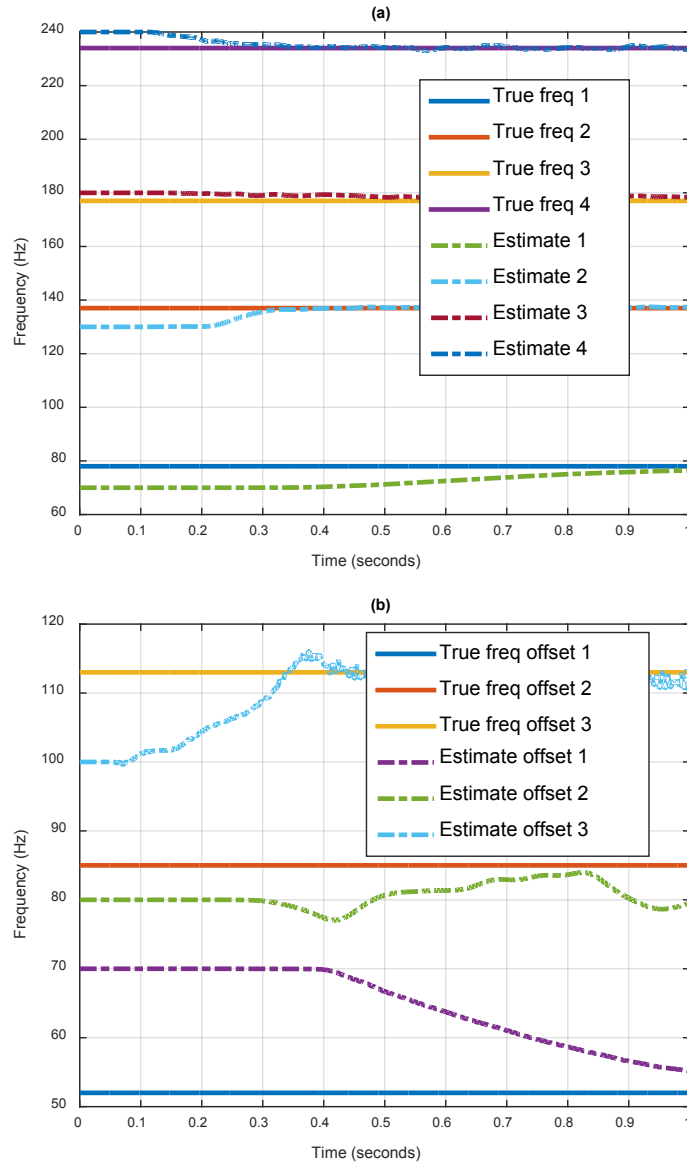


Figure 3.22: In the simulation, the sampling procedure, the notch depth and adaption speed are the same as in Figure 3.16. Presented on the left is the first four frequency estimation via the prefiltering ANF method (3.17) with the initial frequencies 70, 130, 180, 240Hz. On the right is the last three frequency estimation via discrete ANF method with heterodyning, where we use 270, 330, 450Hz as center frequencies of bandpass filters.

We have only shown the capability of the proposed ANF method as successful estimator in condi-

tions of multiple frequencies and high additive noise. For most of our simulations we have used nonuniform data, except when we simplified it to uniform case and we mention it explicitly. Once the frequencies are obtained using the proposed discrete method the problem of sinusoidal parameter estimation in (3.2) simply reduces to a linear problem of estimating amplitude and phase [39]. The amplitudes and phases in (3.2) can be computed using the states of differential equations directly, or using least squares solution of the linear model methods like [39].

### 3.4 Conclusions

Multiple frequency estimators from nonuniform sampling data are proposed in this paper, which reconcile the merits of adaptive notch filtering in continuous time while restricting to operate directly on discrete time data. Despite the mathematical difficulties, some conclusive results on the local stability analysis and convergence of the proposed solution are established. To our best knowledge, solving the frequency estimation problem directly from some dynamic systems is not presented in the literature when only nonuniform sampling data are available. We expect that this paper will stimulate further interest in this direction.

Local stability analysis, convergence and the stochastic characterization of the proposed frequency estimators are established when the input signal has a single frequency. Extensive simulations for various scenarios demonstrate robustness, accuracy and fast tracking of the proposed frequency estimators. For multiple frequencies, we have two proposed methods, the cascade procedure and the prefiltering process. While the cascade method is good for small number of frequencies in nonuniform setting, the prefiltering method is useful for faster response in uniform settings. The proposed method is robust even in the estimation of two close frequencies in high noise.

Further research on parameter tuning, which is an open problem in control research, especially

for particular applications of the proposed frequency estimators to signal processing and control, is needed. The reader may refer to [68, 69, 70, 71, 72, 73, 74] on robust parameter tuning for adaptive notch filtering in the continuous domain.

## CHAPTER 4: NONUNIFORM POLAR AND SPHERICAL DISCRETE FOURIER TRANSFORM

The nationalist not only does not disapprove of atrocities committed by his own side, but he has a remarkable capacity for not even hearing about them.

*George Orwell, 1950*

He who makes a beast of himself gets rid of the pain of being a man.

*Samuel Johnson, 1709 – 1784*

Man is neither an animal nor an angel, he shares a condition of both, but an existence of neither. ...We conceive a mind as an unending effort at comprehension, an effort that can never be concluded wholly, though we are always advancing thought with many methods of research. We have no basic concept in terms of which we can define man nor any theory that can wholly cover his actual objective existence. We must, therefore, as scientists, keep an open mind for all the empirical possibilities and guard against the temptation to reduce the human existence to one common denominator.

*Karl Jaspers, The General Psychopathology, 1910*

Numerous applied problems of 2D and 3D imaging are formulated in continuous domain.<sup>1</sup> They

---

<sup>1</sup>Part of the material in this section is reprinted from “An Exact and Fast Computation of Discrete Fourier Transform for Polar and Spherical Grid” by Syed Alam Abbas, Qiyu Sun, and Hassan Foroosh, IEEE Trans. on Signal Processing, to appear soon, ©2016 IEEE, with permission from IEEE.

place great emphasis on obtaining and manipulating the Fourier transform in polar and spherical coordinates. However, the translation of continuum ideas with the discrete sampled data on a Cartesian grid is problematic. There exists no exact and fast solution to the problem of obtaining discrete Fourier transform for polar and spherical grids in the literature. In this paper, we develop exact algorithms to the above problem for 2D and 3D, which involve only 1D equispaced FFT with no interpolation or approximation at any stage. The result of the proposed approach leads to a fast solution with very high accuracy. We describe the computational procedure to obtain the solution in both 2D and 3D, which includes fast forward and inverse transforms. We find the nested multi-level matrix structure of the inverse process, and we propose a hybrid grid and use a preconditioned conjugate gradient method that exhibits a drastic improvement in the condition number.

#### 4.1 Introduction

Given a function  $f(x, y)$  of a continuous argument  $(x, y) \in \mathbb{R}^2$ , its two-dimensional (2D) Fourier transform (FT) [99]-[101] is given by

$$\hat{F}(u, v) = \int_{\mathbb{R}^2} f(x, y) e^{-2\pi i(xu + yv)} dx dy, \quad u, v \in \mathbb{R}. \quad (4.1)$$

Transforming the frequency coordinates from Cartesian to polar coordinates, i.e.  $(u, v) = (\rho \cos \theta, \rho \sin \theta)$ , we get the continuous polar Fourier transform

$$\hat{F}(\rho, \theta) = \hat{F}(\rho \cos \theta, \rho \sin \theta) \quad (4.2)$$

with a slightly abused notation.

Numerous applications of polar FT can be found especially in imaging. The significance of this

trivial change of variables is that several fundamental procedures for manipulation of such continuous functions  $f$  (for instance rotation) can be described most elegantly in terms of polar variables. The polar FT is a powerful tool for organizing the understanding of operators and functions on the 2D continuous domain [102], [103]. Much the same can be said of higher dimensions.

The continuous domain concepts are simple and obvious, but most data in practice exists only in sampled digital form [104]. This is the case in estimating image rotation, image registration, tomography or Radon transform, synthetic aperture radar imaging and more. The prevailing belief is that there exists no such algorithm for fast and exact computation of discrete polar FT ([105], [102], [103], [113] and [114]). Several difficulties as discussed in [102] exists for constructing fast polar FT for digital data: (1) appropriate polar grid should be defined in the frequency domain; (2) a fast way is needed to evaluate or approximate the FT on these polar grid points; and (3) a fast and stable transform inversion design is required.

For modern applications in scientific computing, two existing bodies of literature contain the solution to the above formulation of polar FT using discrete data. (1) One corresponds to methods that compute the inverse Radon transform with applications in wide range of disciplines such as radar imaging, geophysical imaging, nondestructive testing, and medical imaging [106]–[117]. This approach is called the direct Fourier method. It exploits the projection-slice theorem to convert projection data to 2D Fourier data and then reconstruct by Fourier inversion. (2) The second approach treats the general problem of evaluating the FT on non-equally spaced frequency points. This problem is known as USFFT (Unequally Spaced FFT) or the NUFFT/NFFT (non-Uniform FFT) [118]–[126]. Since the frequencies are by definition unequally spaced, the problem of evaluating the polar FT can be considered as a special case of the USFFT problem in this approach.

A detailed discussion of different approaches to polar Fourier transform adopted in the literature is presented in [111], [102], [113], [114] and references therein. The method in those papers is based



on a special grid, called the pseudo polar (PP) grid, where equi-spaced angular lines are replaced by equi-sloped ones and the computation is fast. This grid has been explored extensively since the 1970s [102]. For long, it has been argued that this is the best way to obtain polar-like sampling specially adapted for the digital data, and hence it has proliferated to numerous applications as a substitute for the true polar form (i.e. the true discrete analog of its continuous form), see [113] and references therein. In comparison with true or ideal polar discrete grid, in the PP grid neither the angles are equi-spaced nor the points on the radial lines are equi-spaced with respect to one another. However, due to its data friendly nature it is considered as a well established alternative. The PP FFT is also used for compressed sensing (CS) based accelerated reconstruction [116] using a modified type of tomography called equisloped tomography [117] which was specifically designed for the PP grid as an alternative to the true polar grid. The existing state-of-the-art solution in [102] adapts the ideas from the USFFT literature, uses an oversampled PP grid, and gives an approximate solution of the true Polar FT depending on the oversampling factor.

An alternative approach to the PP grid is the well-known USFFT or NUFFT based solution [103], which requires a cut-off parameter (for window functions) and an oversampling factor. However for applications like image registration which requires an accurate computation of the true polar FT from the given digital image ([127], [128]), the method ([102]) and the related USFFT interpolation methods ([122, 103]) are slow for high accuracy, and the choice of the parameter of the oversampling rate could be difficult for an arbitrary image [129]. For the computation of polar grid [102], PP grid acts as a stepping stone. Therefore, in order to obtain high accuracy, the PP grid must be oversampled, both radially and angularly, the minimum oversampling required is 2 ([102]), see [130] for more recent development. The result is still interpolated and inaccurate. In order to reduce the interpolation errors a multi-layered FFT (MLFFT) was introduced in [129]. With the use of the MLFFT method, the registration results are improved compared to the PP grid. However, again it requires the selection of cuts (of layers), which must be chosen carefully for it

to work, but the inaccuracies still exist since it also uses an interpolation scheme. Some other variants of the PP grid such as the *Generalized* PP grid have been introduced, which are supposed to provide closer approximations to the true polar grid, but fast and accurate estimation has long been a major challenge for many image processing applications including the phase-correlation based motion estimation [128, 131]. One of the advantages of using a true polar grid instead of a PP grid, Generalized PP, MLFFT grid or log-polar grid is that we can do a straightforward computation of the phase-correlation operation in the polar Fourier domain for the registration problem, see [183] for a direct application in 3D.

To our knowledge, there exists no direct and exact computational procedure for obtaining the true *Polar* FT (2D) or *Spherical Polar* FT (3D), that is fast, devoid of any oversampling or interpolations ([102], [103]), and no design on parameters ([103], [129]). Our approach in this paper is uniquely in contrast to all the approaches discussed above, i.e. it does not require any accuracy parameter (like oversampling for PP FFT) and a cutoff parameter (like USFFT) or a design parameter (like cuts for MLFFT). Oversampling in our case for 2D and 3D refers to computing the Fourier coefficients at grid points outside the true desired grid points of polar and spherical grids, respectively. In [102], [103], [129], [131] and [132], an oversampled grid is first computed in the Fourier domain and the desired grid points (the actual points making up the polar and spherical grids) are then interpolated in it using some sort of high accuracy approximation scheme.

The remainder of this chapter is organized as follows. Section 4.2 discusses the proposed method in 2D setting. The corresponding mathematical solution is described and the implementation details are given. Section 4.3 contains the digital inverse problematics and it provides a solution based on exact forward and adjoint formulation. We show the result of inversion using a simple diagonal preconditioner and a solution to improve conditioning of the inversion matrix, and exploit the underlying special matrix structure which is a nested 1-level block Toeplitz for 2D obtained from the transform operators (forward and adjoint) of polar FT. Section 4.4 creates the formulation and

its solution in 3D, which is far more complex and has a richer geometric structure to exploit. The special underlying matrix structure is a nested 2-level deep block Toeplitz matrix and hence a very fast inversion solution can be designed. Due to the size and nature of complexity of the algorithm in 3D, a good software implementation which exploits the intricate 3D solution becomes very important. We show that the execution speed of the algorithm is greatly improved compared to the brute force (from hours to minutes) by the parallel implementation using GPU programming (CUDA) on a commodity hardware. Finally we follow by the contributions and conclusions in Section 4.5.

In the spirit of reproducible research, we will make our code available to the public for academic purposes: <https://github.com/syedalamabbas>

## 4.2 The Proposed Method- 2D

In the first part of this section, we recall the definition of 1D fractional Fourier transform (FrFT) and discuss the three-step procedure to evaluate it. In the following core subsection, we use 1D FrFT to compute frequency domain information of 2D images at an arbitrary point, and then we show how it can be adapted to the true discrete polar Fourier domain solution (4.13) based on the polar grid definition (4.6) that is fast and algebraically exact. As it only involves the use of 1D FFT, it is fully vectorizable, memory efficient, and fast. It does not involve any parameter design for the user, preprocessing, or padding of any kind, but it has several levels (4.13). The computations of different levels are completely independent, and hence it is easy to parallelize the process on a GPU, multi-CPU and grid computing (CPU/GPU) systems.

#### 4.2.1 Preliminaries: 1D Fractional Fourier Transform

Given an arbitrary scalar value  $\alpha$ , the definition of the 1D fractional Fourier Transform (FrFT), also known as the Chirp-Z transform [134], is given by

$$F^\alpha(k) = \sum_{n=-N/2}^{N/2} f(n) e^{-i \frac{2\pi k \alpha n}{N+1}}, \quad -N/2 \leq k \leq N/2, \quad (4.3)$$

where  $f(n)$  is a discrete 1D signal supported on  $-N/2 \leq n \leq N/2$  and  $N$  is an even integer. There is a fast algorithm to compute the above 1D FrFT in the order of  $(N+1) \log_2(N+1)$ , see [134] and [102] for instance. Substituting  $2kn = k^2 + n^2 - (k-n)^2$  in the exponent of (4.3), we have

$$\begin{aligned} F^\alpha(k) &= \sum_{n=-N/2}^{N/2} f(n) e^{-\frac{i\pi\alpha}{N+1}(k^2+n^2-(k-n)^2)} \\ &= e^{-\frac{i\pi\alpha}{N+1}k^2} \sum_{n=-N/2}^{N/2} f(n) e^{-\frac{i\pi\alpha}{N+1}n^2} e^{\frac{i\pi\alpha}{N+1}(k-n)^2} \\ &= E(k) \sum_{n=-N/2}^{N/2} f(n) E(n) E(k-n) \end{aligned} \quad (4.4)$$

for all  $-N/2 \leq k \leq N/2$ , where

$$E(n) = e^{-\frac{i\pi\alpha}{N+1}n^2}, \quad -N/2 \leq n \leq N/2. \quad (4.5)$$

To evaluate the FrFT (4.3), we use the following three-step procedure based on FFT and convolution in (4.4):

- (i) We first multiply  $f(n)$  by  $E(n)$  and compute its FFT.
- (ii) We then take the FFT of  $E(n)$  and multiply it to compute the convolution.

- (iii) We finally apply the IFFT to the product of the two FFTs and multiply it with  $E(k)$ , which gives us the desired FrFT.

As noted in [134] and [102], the FFT should be computed with zero padding to length  $2(N + 1)$  in order to avoid cyclic effects. A 1D FFT with  $(N + 1)$  points can be evaluated with operational cost  $5(N + 1) \log_2(N + 1)$ . In our procedure, since three 1D-FFT's are needed and the zero padded sequence length is  $2(N + 1)$ , the total operation cost of evaluating (4.3) is  $30(N + 1) \log_2(N + 1)$  floating point operations (flops).

#### 4.2.2 *The Algorithm*

We begin with the idea of using 1D FrFT to compute frequency domain information in 2D images for an arbitrary point. Consider a 2D discrete data such as image  $f(r, c)$  of size  $(N + 1) \times (N + 1)$ , where  $N$  is even, and  $r, c$  are the row and column indices, respectively. To compute the Fourier coefficient at any arbitrary location falling inside or outside the grid, we make use of separately scaled 1D FrFT in  $X$  and  $Y$  directions, see Figure 4.1.

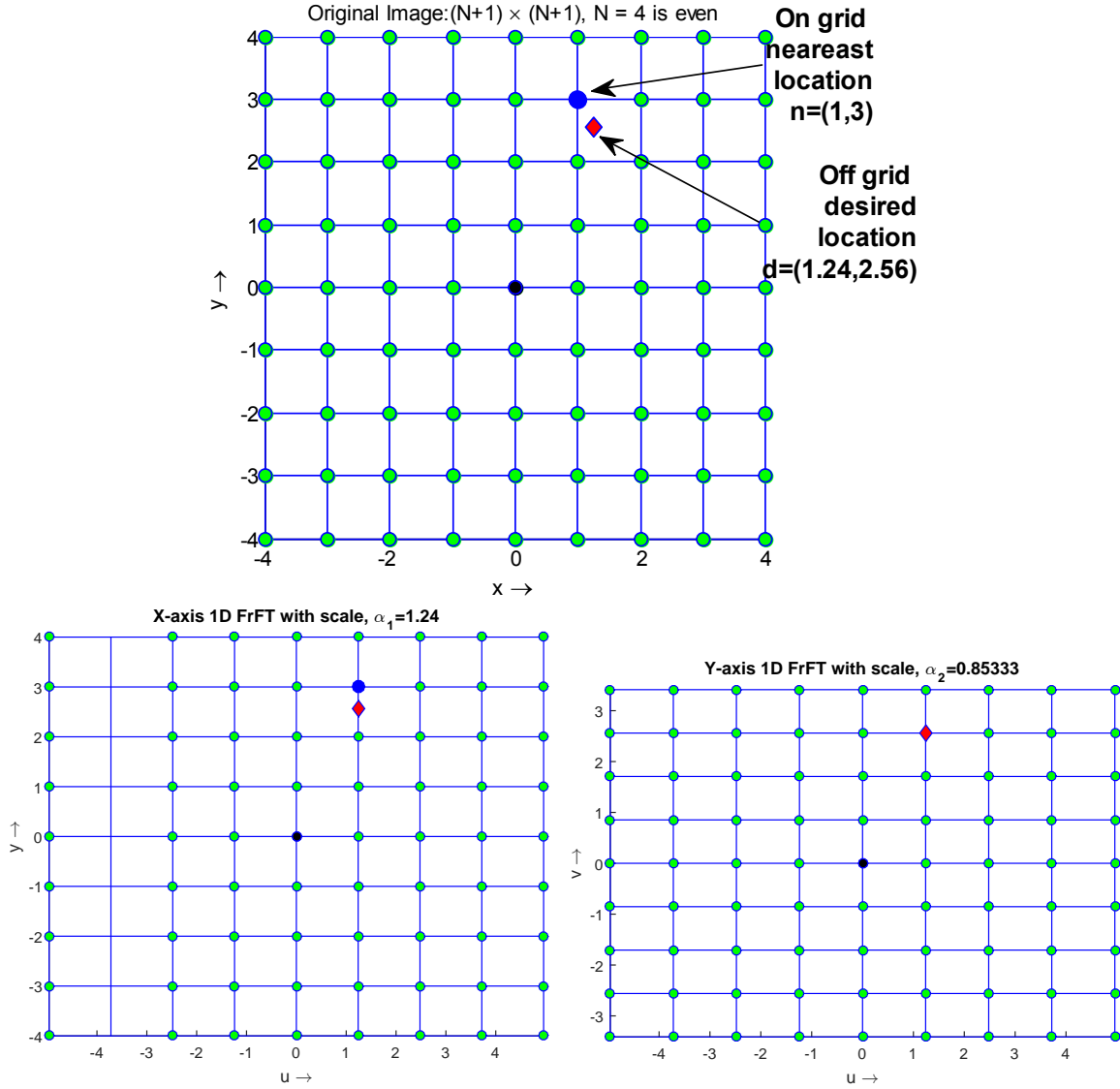


Figure 4.1: Listed on the top, bottom left and bottom right are a uniform grid, an  $X$  axis scaled grid and an  $XY$  axis scaled grid, respectively. In the simulation, the desired point  $(1.24, 2.56)$  for a  $9 \times 9$  grid determines the exact scale factors to be used for the 1D FrFT along row or column. First is  $X$  axis scaling  $\alpha_1 = 1.24$  (expansion) and then along  $Y$  axis,  $\alpha_2 = .85333$  (shrinkage). These sequential scalings result in the desired point falling on coordinates  $(1, 3)$  of the transformed frequency grid.

The total computational complexity is  $30(N+1)^2 \log_2(N+1) + (N+1)$ , where the first term in

the sum represents 1D FrFT in  $X$  or  $Y$  direction (order does not matter) for all rows or columns, respectively, and the second term represents a direct computation of 1D FrFT for a single point. Inspired by this observation we design the solution for computation of polar grid points using 1D FrFT.

As shown in Figure 4.2, we define the polar grid of frequencies  $\{u_x(m, n), v_y(m, n)\}$  in the circle inscribed in the fundamental region,  $\{u, v\} \in [N + 1, N + 1)^2$ .

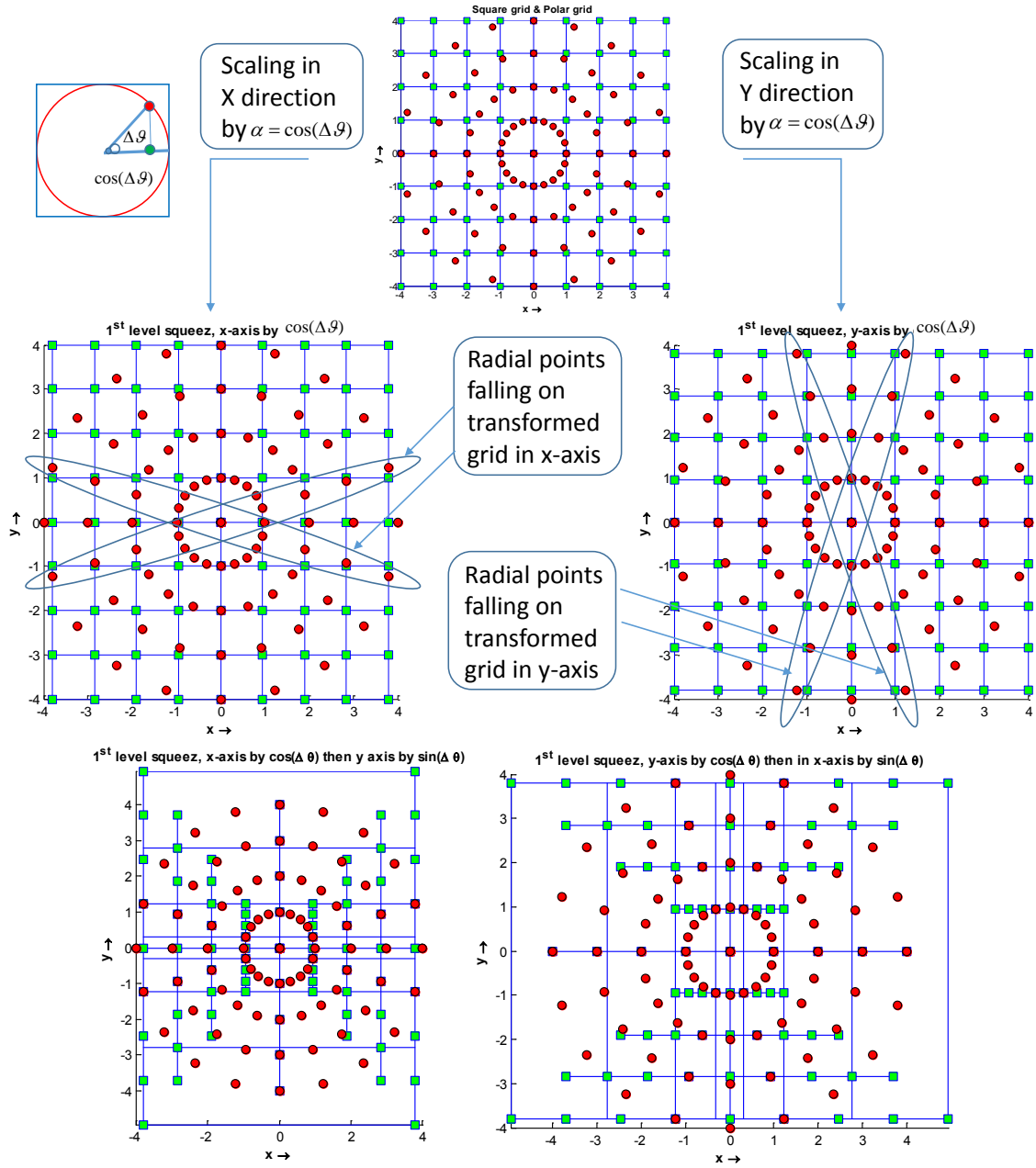


Figure 4.2: Overlapping points of the polar grid with the rectangular scaled grid, each grid has exactly 4 radial lines that fall on it.  $X$  axis scaling has horizontally oriented and  $Y$  axis scaling has vertically oriented radial lines (both shown with ellipses). The four radial lines of level 1 solution are at angles  $\Delta\theta$ ,  $180^\circ - \Delta\theta$ ,  $90^\circ - \Delta\theta$ , and  $90^\circ + \Delta\theta$ , where  $\Delta\theta = 18^\circ$ .



Given a discrete Cartesian data  $f(r, c)$ , we define the polar Fourier transform (PFT) by

$$F(m, n) = \sum_{c=-N/2}^{N/2} \sum_{r=-N/2}^{N/2} f(r, c) e^{-i \frac{2\pi n}{N+1} (r \cos \theta_m + c \sin \theta_m)}, \quad (4.6)$$

where  $-N/2 \leq n \leq N/2$  represent the points on a radial line,  $\theta_m = m\Delta\theta$ ,  $0 \leq m \leq M-1$ , are the angles,  $\Delta\theta = 180^\circ/M$  is the angular spacing, and  $M$  is the total number of radial lines.

The polar grid is of size  $M \times (N+1)$ , which is the total number of nodes with the origin included multiple times, where we choose for simplicity  $M$  as an even integer. Each radial line oriented at a specific angle  $\theta_m$  is composed of  $N+1$  points. The polar grid can be thought as a structure  $P(m, R_m)$ , where  $R_m$  is the radial line data composed of uniformly spaced  $N+1$  points oriented at an angle  $\theta_m$ . Such a definition is useful in the context of Radon transform [111].

To use 1D FrFT for the computation of Fourier coefficients on radial lines of the polar grid, we require few definitions. The 1D frequency domain scaling of a 2D image along the  $X$  axis can be computed using a 1D FrFT along each row using

$$F_x^\alpha(r, c) = \sum_{n=-N/2}^{N/2} f(r, n) e^{-i \frac{2\pi c \alpha n}{N+1}}, \quad (4.7)$$

where  $-N/2 \leq r, c \leq N/2$ . Similarly, the frequency domain scaling of a 2D image along the  $Y$  axis can be computed by using a 1D FrFT along each column using

$$F_y^\alpha(r, c) = \sum_{n=-N/2}^{N/2} f(n, c) e^{-i \frac{2\pi r \alpha n}{N+1}}, \quad (4.8)$$

where  $-N/2 \leq r, c \leq N/2$ . Figure 4.2 shows the two scaled images, that are uniformly scaled in one axis but variably scaled in its complementary axis using 1D FrFT, which we call the *Butterfly grid*. For this special grid, the different scale factors are determined entirely by the geometry of

the problem. The angular spacing  $\Delta\theta$  determines the exact scale factors that are needed.

Consider the first scaling with  $\alpha = \cos(\Delta\theta)$ . We call this the level one scaling, which is composed of two solutions obtained using (4.7) and (4.8). To compute the 2D Fourier domain radial data on the  $X$  axis scaled grid, we take another 1D FrFT but with a variable scale for each column. For the column  $c$ , we scale it by a factor  $|c|\beta$  where  $\beta = \sin(\Delta\theta)$ ,

$$F_x^{\alpha,\beta}(r, c) = \sum_{n=-N/2}^{N/2} F_x^{\alpha}(n, c) e^{-i \frac{2\pi r |c| \beta n}{N+1}}, \quad (4.9)$$

where  $-N/2 \leq r, c \leq N/2$ . Similarly, to compute the 2D FT data of the radial lines on the  $Y$  axis scaled grid, for each row  $r$ , we use the same scale factor  $|r|\beta$ ,

$$F_y^{\alpha,\beta}(r, c) = \sum_{n=-N/2}^{N/2} F_y^{\alpha}(r, n) e^{-i \frac{2\pi c |r| \beta n}{N+1}}, \quad (4.10)$$

where  $-N/2 \leq r, c \leq N/2$ .

Observe that we are interested in evaluating only 1 point for each column using (4.9), and 1 point in each row using (4.10). Thus on the 2D butterfly grid, not all points are computed, only the desired points which exactly overlap with the radial points on the polar grid lines need to be evaluated. There are 4 symmetrical radial lines, 2 at an angle  $\Delta\theta$  and  $180^\circ - \Delta\theta$  from the  $X$  axis scaled grid or basically the horizontal lines [111], and the other 2 at angles  $90^\circ \pm \Delta\theta$  from the  $Y$  axis scaled grid or the basically vertical lines, not including the orthogonal lines at  $0^\circ$  and  $90^\circ$ . These four lines in pairs of 2 overlap exactly with the scaled butterfly grids in (4.9) and (4.10), respectively. The overlaps are obtained using the 1D FrFT operations with properly selected scales  $\alpha$  and  $\beta$ , see Figure 4.2 with overlaps marked explicitly. This gives us the complete level one solution of the Fourier data on 4 radial lines:  $F(1, n)$ ,  $F(M-1, n)$ ,  $F(M/2-1, n)$  and  $F(M/2+1, n)$ , which is denoted by  $F_1$ . The radial lines at the angles  $0^\circ$  and  $90^\circ$  can be computed using data from any

levels.

For a perfect polar grid (i.e. when  $M$  is even), the different radial lines oriented in X-axis and Y-axis are given by angles  $\theta_l = l \times \Delta\theta, l \geq 0$ , and we have the corresponding scale factors  $\alpha_l = \cos \theta_l$  and  $\beta_l = \sin \theta_l$ . Consider for level one solution using (4.9), we pick points that radially cross the dc value by 1 point, as shown in Figure 4.2, to get the desired radial line. Similarly, for level two solution, we select points that radially cross the dc value by 2 points to obtain the level 2 radial lines, as displayed in Figure 4.3. Inductively, the solutions of all levels can be written as

$$F_x^{\alpha_l, \beta_l}(r, c) = \sum_{n=-N/2}^{N/2} F_x^{\alpha_l}(n, c) e^{-i \frac{2\pi r |c| \beta_l n}{l(N+1)}} \quad (4.11)$$

and

$$F_y^{\alpha_l, \beta_l}(r, c) = \sum_{n=-N/2}^{N/2} F_y^{\alpha_l}(r, n) e^{-i \frac{2\pi c |r| \beta_l n}{l(N+1)}}, \quad (4.12)$$

where  $-N/2 \leq r, c \leq N/2$  and  $1 \leq l \leq L$ . The set of scales  $S = [\alpha_l, \beta_l; 1 \leq l \leq L]$  has the cardinality of  $2L$ , and the complete DFT information on the polar grid is given by

$$F = \bigcup_{l=1}^L F_l, \quad (4.13)$$

where  $F_l, 1 \leq l \leq L$ , are the solutions for different levels  $l$ . Shown in Figures 4.2 and 4.3 are the complete solution for the polar grid on the top of Figure 4.2, where  $M = 10$  and  $L = 8/4 = 2$ .

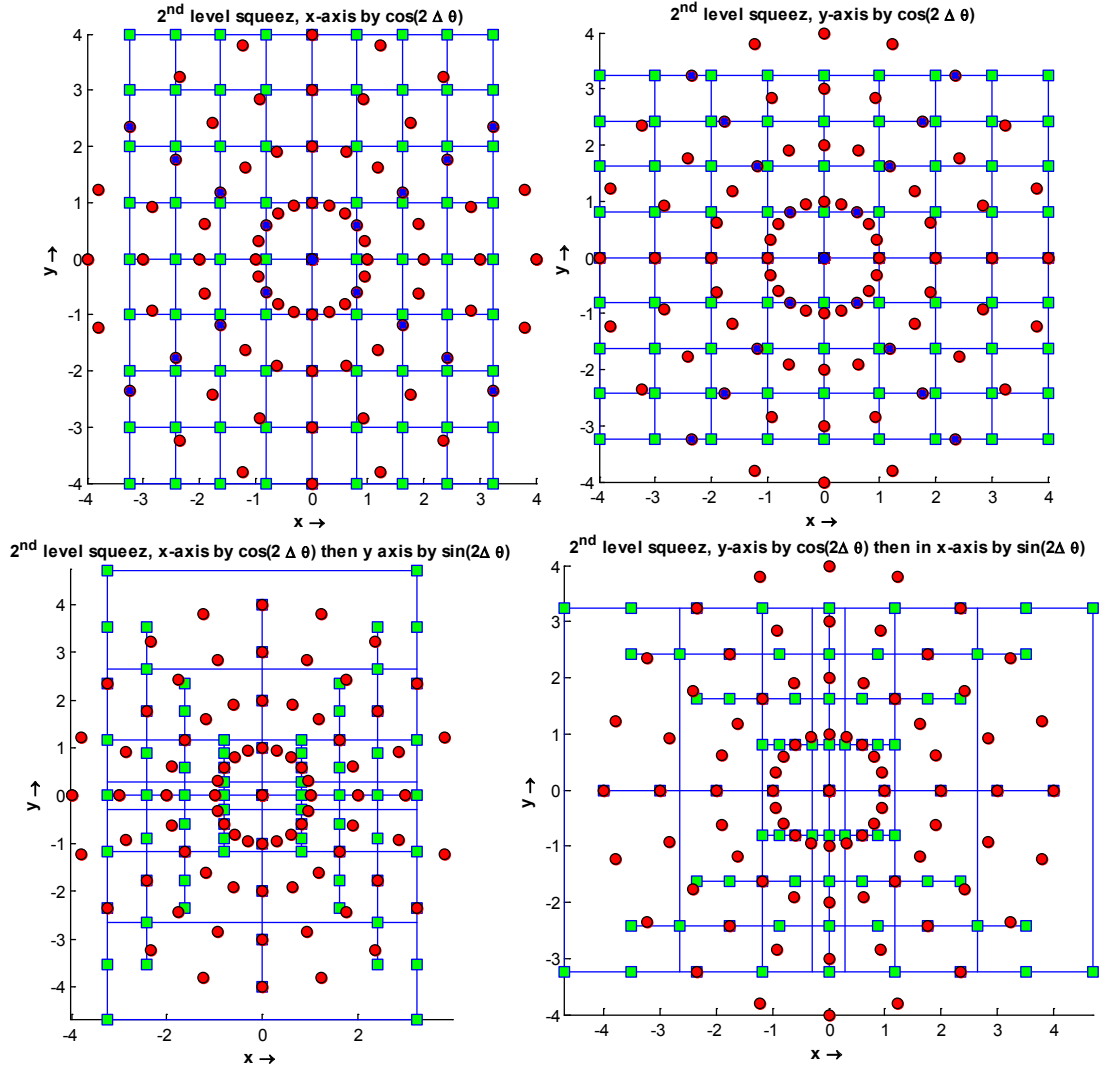


Figure 4.3: Top row (left) (a) Image with  $X$  axis scaled grid, (right) (b) Image with  $Y$  axis scaled grid. Bottom row (left) (c) Image with  $XY$  axis scaled grid, and (right) (d) Image with  $YX$  axis scaled grid, with appropriate scales respectively. The four radial lines of level 2 solution are at angles  $\Delta\theta$ ,  $180^\circ - \Delta\theta$ ,  $90^\circ - \Delta\theta$ , and  $90^\circ + \Delta\theta$ , where  $\Delta\theta = 2 \times 18^\circ$ . This completes all the lines of the polar grid with  $M = 10$  and  $N = 8$  in (4.13).

#### 4.2.2.1 Exploiting Symmetry

For the case that the scales  $\alpha_l$  and  $\beta_l$  are rational numbers, the computation of the FrFT could reduce to conventional FFT, and the computation speed could be significantly increased by exploiting the periodicity of the data sequence. However, it is meaningless to talk about periodicity when dealing with the remaining case that one of scales  $\alpha_l$  and  $\beta_l$  is irrational. So the use of same scaling within each level gives us some added advantage for that remaining case. By applying uniform scaling along  $X$  or  $Y$  axes, we obtain two symmetrical radial lines due to the conjugate symmetry in uniformly scaled grid. Due to irrational scales, there is hardly any connection across scales. However, for the same scaling  $\alpha_l$  along  $X$  and  $Y$  axes, the calculations in (4.7) and (4.8) yield

$$F_x^{\alpha_l}(r, c) = F_y^{\alpha_l}(r, c) \quad (4.14)$$

for diagonal elements with  $r = c$ . For elements satisfying  $rn_1 = cn_2$  for some integers  $n_1$  and  $n_2$ , it is not necessary to compute twice for two grids on the same level, as we can efficiently use caching,

$$f(r, c)e^{-i\frac{2\pi r\alpha n_1}{N+1}} = f(r, c)e^{-i\frac{2\pi c\alpha n_2}{N+1}}. \quad (4.15)$$

#### 4.2.2.2 Computational Complexity

The number of levels required for the polar grid is approximately  $L = M/4$ . In particular, when  $M$  is singly-even (i.e.  $M$  is divisible by 2 but not by 4), the number of levels is exactly  $(M - 2)/4$ . For the remaining case that  $M$  is doubly-even (i.e.  $M$  is divisible by 4), we have the special radial lines along the angles  $45^\circ$  and  $135^\circ$ , and hence we need to compute it only once using  $X$  or  $Y$ -axis scaled grid. Therefore the number of levels is  $M/4$ . At every level, the most expensive operation is the first uniform scaling by  $\alpha_l$  along  $X$  or  $Y$  axis using (4.7) and (4.8), respectively. The second

variable scaling to obtain the radial lines is simple since we are targeting far fewer points. Then, at each level, we have the complexity  $(N + 1)^2/2 \times \log_2(N + 1)$  for  $X$  axis scaling using real data, and similarly  $(N + 1)^2/2 \times \log_2(N + 1)$  for  $Y$  axis scaling. As we require about  $M/4$  levels, the total computational complexity of our algorithm is

$$\frac{M}{4} \times (N + 1)^2 \log_2(N + 1). \quad (4.16)$$

The above estimate of polar grid holds due to the perfect symmetry of the equispaced points both angularly and radially. In the worst scenario when all the  $M$  radial lines are non-equispaced angularly, the computational complexity will increase to its maximum  $M(N + 1)^2 \log_2(N + 1)$ . As long as the radial lines have equispaced points (even when it is angularly non-equispaced), we can use the proposed approach to solve for this type of polar grid. When even the points on the radial lines are not equispaced, unfortunately we cannot use the proposed solution based on the 1D FFT, and we should resort to approximation techniques of nonuniform FFT ([103]).

#### 4.2.3 Implementation Details

Considering (4.3), the index is in the unaliased form, but the FFT routines in most software systems are zero based or aliased form ([134]). We have to do a few modifications when we implement the computation of (4.3) using FFT for convolution. Let  $n' = n + N/2$ , then

$$\begin{aligned} F^\alpha(k) &= \sum_{n'=0}^N f\left(n' - \frac{N}{2}\right) e^{-i \frac{2\pi k \alpha (n' - \frac{N}{2})}{N+1}} \\ &= e^{i \frac{\pi \alpha N k}{N+1}} \sum_{n'=0}^N f\left(n' - \frac{N}{2}\right) e^{-i \frac{2\pi k \alpha n'}{N+1}}, \end{aligned}$$

where  $-N/2 \leq k \leq N/2$ . Similarly for  $0 \leq k' = k + N/2 \leq N$ ,

$$F^\alpha\left(k' - \frac{N}{2}\right) = e^{i\frac{\pi\alpha N(k' - \frac{N}{2})}{N+1}} \times \sum_{n'=0}^N f\left(n' - \frac{N}{2}\right) e^{-i\frac{2\pi n'\alpha(k' - \frac{N}{2})}{N+1}}.$$

Rearranging and simplifying the above expression gives

$$F^\alpha\left(k' - \frac{N}{2}\right) = C(k') \sum_{n'=0}^N \left\{ f\left(n' - \frac{N}{2}\right) B(n') \right\} e^{-i\frac{2\pi\alpha n'k'}{N+1}}, \quad (4.17)$$

where  $C(k') = e^{i\frac{\pi\alpha Nk'}{N+1}}$  and  $B(k') = e^{i\frac{\pi\alpha N(k' - \frac{N}{2})}{N+1}}$ ,  $0 \leq k' \leq N$ . Thus for the unalaised form, using the pre-multiplication factor  $B$ , and post-multiplication factor  $C$ , the expression in (4.17) reduces to zero based solution which can be computed via convolution using FFT routines as described in (4.4).

The following is the pseudo code for our proposed 2D fast PFT computation:

**Input:** Image  $f$  of size  $(N+1) \times (N+1)$  and the number  $M$  of angles, where  $M, N$  are even.

**Output:** Polar grid  $F$  of size  $M \times (N+1)$  computed according to the definition (4.6).

**Step 1:** For the level  $l$ , where  $1 \leq l \leq L$  and  $L \cong M/4$  is the total number of levels, do the following.

**Step 2:** Apply the scaling  $\alpha_l$  to evaluate 1D FrFT (4.9) for the image  $f$  row-wise.

**Step 3:** Apply the scaling  $\alpha_l$  to evaluate 1D FrFT (4.10) for the image  $f$  column-wise.

**Step 4:** Compute two horizontal lines using the scaling  $\beta_l$  according to (4.11).

**Step 5:** Compute two vertical lines using the scaling  $\beta_l$  according to (4.12).

**Step 6:** Gather four radial lines for level  $l$  in the output array.

**Step 7:** If  $l == L$ , terminate. Else  $l = l + 1$ , go to **Step 2**.

#### 4.2.4 Accuracy of the Proposed Fast PFT

The use of FFT for numerical calculation of convolutions has a fundamental challenge of aliasing error [139]. This inherent error manifests itself for larger arrays, while for computations of smaller arrays the accuracy approaches the machine precision. Figure 4.4 shows the precision loss for the computation of 1D FrFT, for a randomly generated signal  $f(n)$ ,  $-N/2 \leq n \leq N/2$  and scaling factor  $\alpha$ , where the maximum absolute error is given by

$$E_{\max} = \max_n |F_{\text{direct}}^{\alpha}(n) - F_{\text{fast}}^{\alpha}(n)| \quad (4.18)$$

is plotted against the array size  $N$ . As indicated in the figure the error steadily increases for larger array size, but the loss of precision is still graceful even for a sufficiently large array size. The error is in the order of  $10^{-9}$  for  $N = 1400$ .

The fast convolution step ([137, 138]) in the proposed fast PFT algorithm is the only source of



error, all the other steps are exact up to machine precision, since there are no interpolations, approximations or any form of oversampling. This precision loss for larger size arrays is also present in the computation of PP grid (or just about any algorithm where fast convolution is used), since the basic building block, 1D FrFT based on fast convolution, is the same for the computation of both grids.

The zero padding used for computation of the convolution (4.4) is not necessary, as there are ways to compute this without padding with additional variables, see [141] for details. The reader may refer to [137]–[140] for various methods for improving precision.

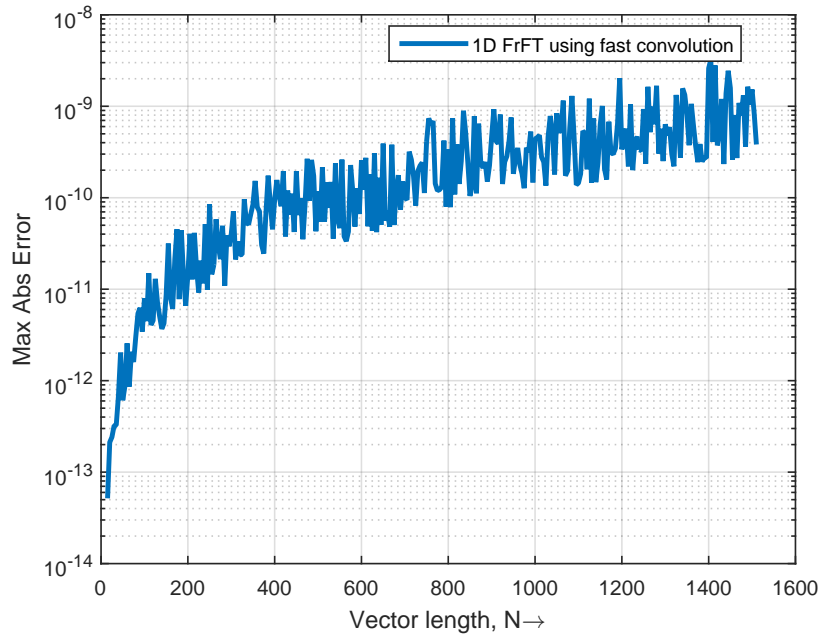


Figure 4.4: Precision loss due to fast convolution,  $E_{\max}$  vs.  $N$ . As can be seen from the figure that although there is a steady loss of precision for larger array sizes, for moderately sized array the error is still of order  $10^{-10}$ . This is the same precision loss with the polar grid computations using (4.6) since the fast convolution used for 1D FrFT is the building block of it.

#### 4.2.5 Comparison

The following numerical simulation is computed with the NFFT C-subroutine library [142]. The parameters to choose are the cutoff parameter  $\rho$  for Keiser-Bessel window functions, and the oversampling factor  $\sigma = 2$  used internally. The NFFT evaluates (4.6) at  $N \times M$  arbitrary nodes (in our case the polar grid) in

$$O((\sigma N)^2 \log(\sigma N)^2 + \rho^2 NM) \tag{4.19}$$

arithmetical operations.

Figure 4.5 shows the simulation results for an oversampled grid as well as a normally sampled grid. Here, we let  $N = 128$ ,  $M = 3N$  for an angularly oversampled grid and  $M = N$  for normally sampled one.

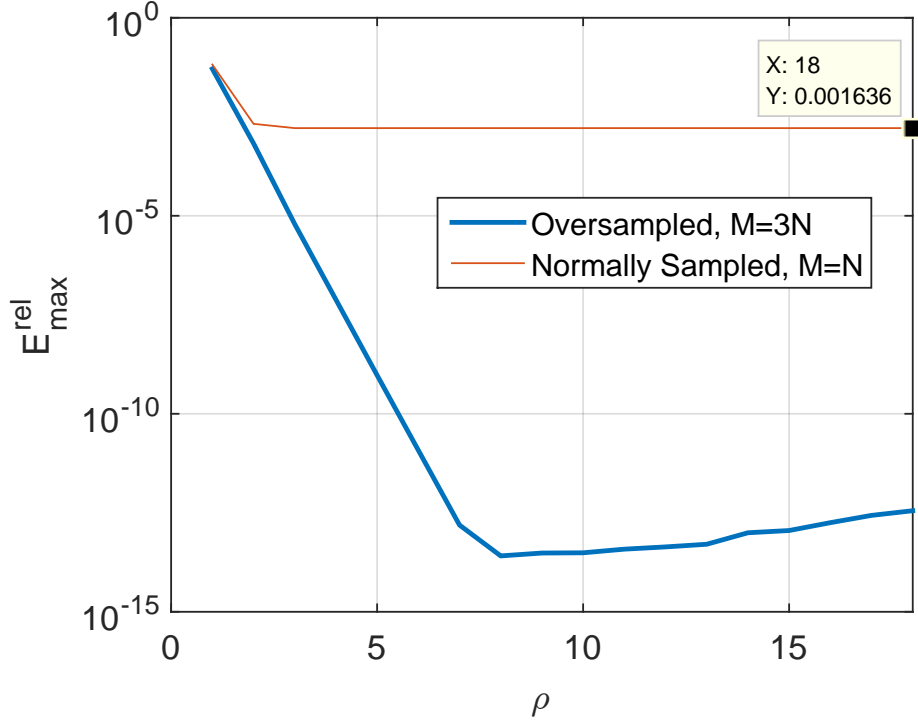


Figure 4.5: In this simulation, the accuracy and computational complexity of NFFT are compared with the proposed algorithm for the polar grid.

For the simulation, the computational complexity of our algorithm for  $M = 128$  ( $M/4 = 32$ ) is  $O(32 \times 128^2 \log_2 128) = O(224 \times 128^2)$ , and the accuracy is achieved is very high. On the other hand, the complexity of NFFT is  $O(18^2 \times 128^2 + 4 \times 128^2 \log_2 256) = O(356 \times 128^2)$ , since  $\rho = 18$ . The relative accuracy  $E_{\max}^{\text{rel}}$  achieved is of the order  $10^{-3}$ , where the relative accuracy of the NFFT is given by

$$E_{\max}^{\text{rel}} = \frac{\max_{m,n} |F_{m,n} - F_{m,n}^{\text{NFFT}}|}{\max_{m,n} |F_{m,n}|}. \quad (4.20)$$

The accuracy could improve in the oversampled case, as shown in the blue line in Figure 4.5 with  $M = 3N$ . The improvement is due to the fact that dense angular sampling reduces the interpolation

errors for NFFT. But it is not advisable to arbitrarily increase the cutoff parameter  $\rho$  as the error increases after certain threshold, for instance  $\rho > 8$  and  $M = 3N$  in Figure 4.5.

The proposed method does not depend on the angular oversampling and still achieves very high-accuracy approximation i.e. the relative error  $E_{\max}^{\text{rel}}$  is of order  $10^{-14}$  and the absolute error  $E_{\max}$  is of order  $10^{-10}$ .

The execution time of the proposed method is shown in the Figure 4.6. The direct computation of Discrete PFT is of order  $O(N^4)$ , while the proposed fast technique with far reduced complexity can handle images of sufficiently large sizes. As the execution time of the proposed solution is orders of magnitude faster, while maintaining the similar level of accuracy as brute force computations. The solution for larger  $N$  can be achieved with parallelization as the proposed technique can be fully distributed and scales naturally with the computational resources.

The proposed solution can be adapted to approximate other kinds of grid used in the literature, such as spiral grid [143], log-polar grid [129] using oversampling and approximation. It is easily extensible for limited angle tomography grid or even a completely arbitrary grid with all angles non-uniformly spaced.

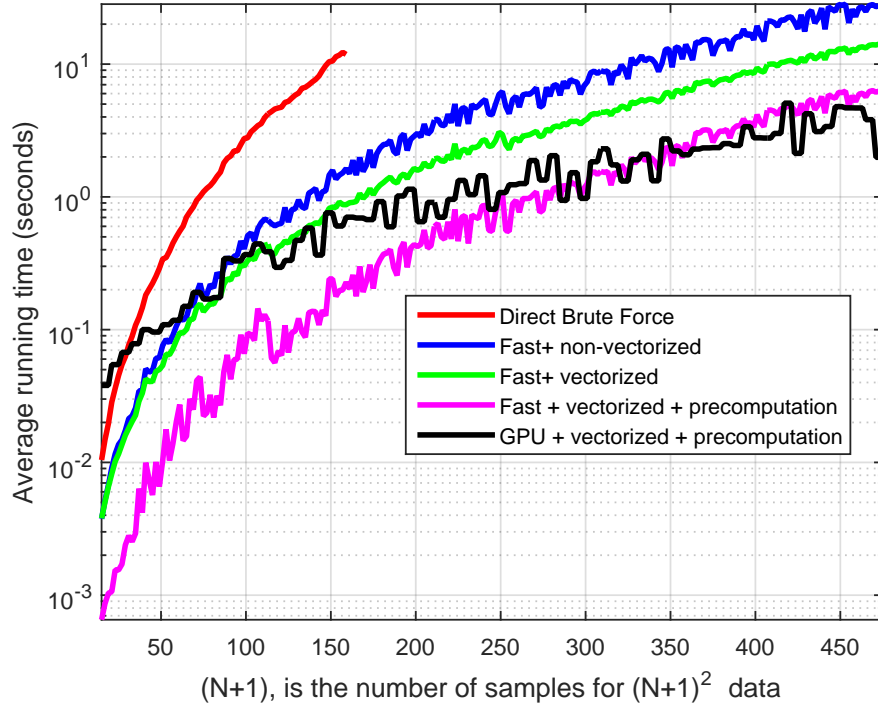


Figure 4.6: Plotted are the execution times vs the input array sizes  $N$  where the images are of size  $(N + 1)^2$ . The execution time using direct brute force computations is not plotted for  $N > 150$  as it grows quickly and can take hours to complete.

### 4.3 Digital Inverse Problematics

The accuracy of the exact forward transform computations is very high, as discussed in Section 4.2, but for inversion there is no direct, exact and fast algorithm. In practice a fast way is needed for the reconstruction given the polar Fourier data especially for large size images. In the following four subsections, we analyze and design an iterative inversion scheme using linear algebra techniques. We also discuss the ill-conditioning inherent in the polar sampling process, and give solutions that drastically improve the conditioning of the designed linear systems.

#### 4.3.1 Analysis

The proposed method of discrete PFT can be written as a linear operator  $A_p$ , which we call the discrete PFT operator. We use this matrix representation only for analysis, not for implementation. Let  $x$  be a vector formed from stacking the image  $f$  in (4.6) column-wise, and  $y$  be the vector obtained from stacking the PFT values  $F$  in (4.6) row-wise, i.e. the radial lines stacked sequentially. Then the linear transformation can be represented as

$$A_p x = y, \quad (4.21)$$

where  $A_p$  is a complex valued matrix of size  $M(N+1) \times (N+1)^2$ ,  $x$  is a  $(N+1)^2 \times 1$  real vector representing the signal/image, and  $y$  is a  $M(N+1) \times 1$  complex valued vector representing the transformed signal.

An  $(N+1)$  point FrFT in (4.3) can be expressed as an  $(N+1) \times (N+1)$  matrix-vector multiplication similar to an FFT matrix representation. The transformation matrix  $\Omega_\alpha$  with scale factor  $\alpha$  can be defined as

$$\Omega_\alpha = \left( \omega^{(N/2+1-i)(N/2+1-j)} \right)_{1 \leq i, j \leq N+1},$$

where  $\omega = e^{-i \frac{2\pi\alpha}{N+1}}$ . Let  $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots, \mathbf{r}_{N+1}$  be row vectors of the image  $f$ , then the equation (4.7) that describes the uniform scaling along  $X$  axis can be written in matrix-vector product form as

$$\mathbf{r}_n^\alpha = \Omega_\alpha \mathbf{r}_n, \quad 1 \leq n \leq N+1.$$

Stacking these new row vectors together, we get the  $(N+1) \times (N+1)$  matrix  $F_x^\alpha$  as shown in (4.7). The next scaling is variable scaling as discussed before along  $Y$  axis which requires a

different transformation matrix  $S_\beta$  of the form,

$$S_\beta = \left( \xi^{(N/2+1-j)(N/2+1-i)} \right)_{1 \leq i, j \leq N+1},$$

where  $\xi = e^{-i \frac{2\pi\beta}{N+1}}$ . Since we are only interested in just two lines which are also symmetrical in the equation (4.9), then the solution to a line (oriented at the first angle) in full matrix form becomes

$$S_\beta \circ F_x^\alpha$$

where  $\circ$  is the Hadamard elementwise product. The complementary line solution can be obtained from the matrix  $S_\beta^H \circ F_x^\alpha$  where  $S_\beta^H$  represents the Hermitian transpose of  $S_\beta$ .

Let  $J = (1, 1, \dots, 1)$  be the column vector of 1s of size  $(N+1) \times 1$ . Then for a square matrix  $A$  of size  $(N+1) \times (N+1)$ ,  $J^T A$  is the column-wise sums of  $A$ , and  $AJ$  is the row-wise sums of  $A$ , see [144]. Finally the line oriented at an angle  $\Delta\theta$  and the complementary line oriented at an angle  $180^\circ - \Delta\theta$  are given by

$$R_1 = J^T (S_\beta \circ F_x^\alpha) \text{ and } R_{M-1} = J^T (S_\beta^H \circ F_x^\alpha).$$

Similarly, all the other radial lines can be computed in the above forms. Thus the matrix  $A_p$  has elements of forms,  $S_{\beta_l} \circ W_{\alpha_l}$  and  $S_{\beta_l}^H \circ W_{\alpha_l}$ , stacked either row-wise or column-wise for all the scales  $S = [\alpha_l, \beta_l]$  in (4.11) and (4.12). Only the values of the exponents will change depending on the scales but the structure of the elements obtained would be similar. We can express  $A_p$  as a product of two matrices

$$A_p = VU, \tag{4.22}$$

where  $V, U$  are large and sparse matrices composed of values obtained using terms of  $\beta_l$  and  $\alpha_l$

respectively. The *adjoint* operator of  $A_p$  is given by

$$A_p^H = U^H V^H. \quad (4.23)$$

Consider the matrix-vector formulation of the inversion process from (4.21) (see [111], [122] for similar formulation), multiplying both sides by adjoint gives

$$A_p^H A_p x = A_p^H y.$$

Let  $A_p^+ = (A_p^H A_p)^{-1} A_p^H$  be the generalized inverse or the well known Moore Penrose pseudo inverse, a solution to (4.21) is given by

$$x = A_p^+ y. \quad (4.24)$$

#### 4.3.2 Ill-Conditioning

In contrast to the forward transform and its adjoint operation which have an exact formulation and where the considerations of the density of sampling (distribution of nodes) was of least concern, the inverse PFT and the associated reconstruction error heavily rely on the distribution of the polar sampled nodes  $y$ . The problem inherent is due to the geometry of the polar grid. There is some loss of information since the corners are not represented. From the algebraic point of view, the matrix  $A_p$  in (4.21) is ill-conditioned and the condition number grows quickly as the number  $N$  of samples in the spatial grid increases. From the reconstruction perspective, visible artifacts are present, the solution obtained is sensitive to numerical errors in the measurement process, and numerous iterations are required to achieve the desired low residual error.



One way to combat this problem of missing corners is simply to extend the sampling set for corner compensation, as suggested in [103], where a modified polar grid is proposed, the polar sampling set with more concentric circles is added and nodes not located in the square grid are excluded. If the number of polar grid nodes is  $M \times (N + 1)$ , with the origin included multiple times, the modified polar grid has slightly increased total number of nodes  $\approx \frac{4}{\pi} \log(1 + \sqrt{2})M(N + 1)$ , see Figure 4.7 (ii) and (iv). The effect of the inclusion of corners on the conditioning of the linear PFT operator  $A_p$  can be seen in Figure 4.8. The extra measurements improve the conditioning of the system, but we can further stabilize the inverse transform through a different arrangement. We propose a hybrid grid, where with the pure polar grid we include the corners from the Cartesian grid as shown in Figure 4.7 (v) and (vi).

Figure 4.7 presents several sampling grids and Figure 4.8 shows the effect on the condition number of the operator  $A_p$  with different arrangements. The spatial grid is of size  $(N + 1) \times (N + 1)$ , the number of angles for all grids remains constant (i.e.  $M = 2(N + 1)$ ) only the sampling radial density and the set geometry varies. As we can see in Figure 4.8, the best conditioning is achieved when we use the proposed hybrid grid combined with oversampling radially when the number of angles is fixed. Thus using the extra measurements in the corners the condition number will be well controlled. References [145] and [103] discuss the density of the sampling set and its influence on the condition number by using a more precise mesh-norm, which gives the maximum distance allowed between neighboring nodes. For sufficiently dense sampling sets, the reconstruction problem is well conditioned using the proposed hybrid grid, c.f. [103].

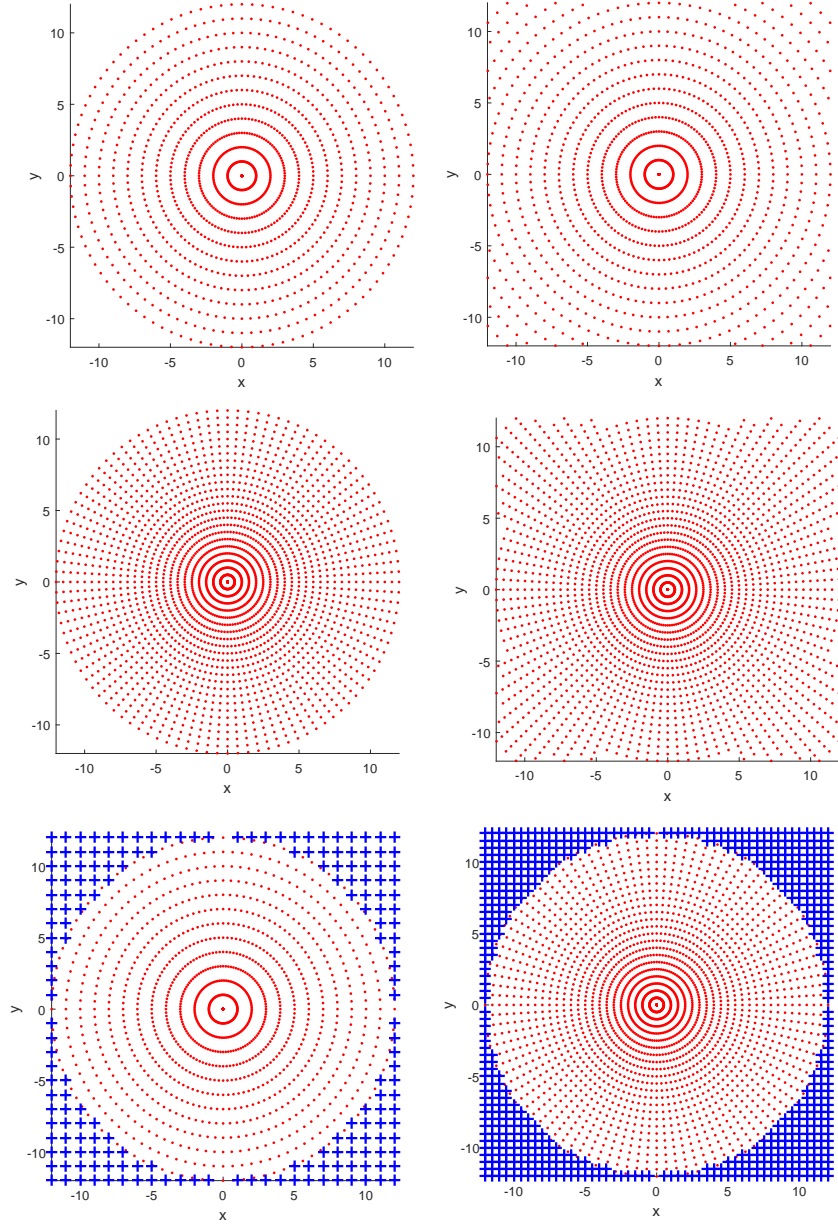


Figure 4.7: Top row, left (i) Polar grid (pure), and right (ii) Polar grid with radial corners included. Middle row, left (iii) Polar grid 2x radial sampling, and right (iv) Polar grid 2x radial sampling with radial corners included. Bottom row, left (v) Polar grid with Cartesian corners (hybrid), and right (vi) Polar grid 2x radial sampling with Cartesian corners. Here,  $N = 24$ , i.e. the spatial domain Cartesian grid is  $25 \times 25$  and only the total number of angles  $M = 52$  remains constant for all the grids. See Figure 4.8 for the condition number comparison of these grids.

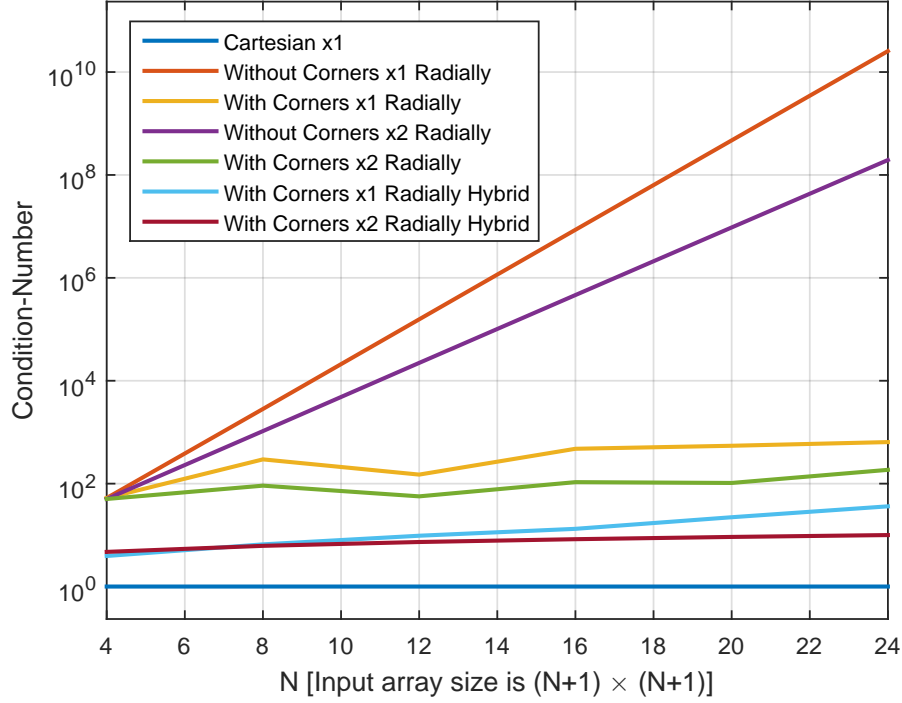


Figure 4.8: Improving the ill-conditioning of the linear system in (4.21) by including the additional measurements. Without the inclusion of corners the condition number of  $A_p$  even for really small sizes of grids ( $25 \times 25$ ) is very high (order of  $10^{10}$ ). Inclusion of corners (see [103]) along radial direction improves the conditioning and so does the oversampling by 2x along radial lines. But the proposed hybrid solution (i.e. polar grid with Cartesian corners) greatly reduces the condition number of the system and combined with oversampling we get the best result which is closest to the ideal condition number of 1 for a perfect Cartesian grid.

#### 4.3.3 Fast Iterative Inversion

The matrix based solution (4.24) is for analysis only. For reasonable sizes of input arrays, a pseudo direct inversion through the matrix  $A_p^+$  would be computationally prohibitive. It is beyond the desired low complexity and high speed of the algorithm. In this subsection we develop a fast iterative reconstruction based on linear algebra analysis and the use of conjugate gradient technique.

For iterative image reconstruction algorithms, one needs the *adjoint*  $A_p^H$  of the discrete PFT operator  $A_p$  in (4.22) to perform matrix-vector multiplications of the form  $A_p^H y = \tilde{x}$ . Since the matrix  $A_p$  is too large to store in the imaging problems of interest, and since the direct matrix-vector multiplication would be computationally inefficient, we evaluate

$$A_p^H y = U^H V^H y$$

by reversing (and not inverting) the operations performed for the forward transform in (4.22) as shown in (4.23). Similar to the above discussion the reversing of the equations (4.9), (4.10), (4.11) and (4.12) can be done with the same complexity as the forward transform.

Instead of using (4.21) and solving it directly via (4.24), we approach it as an optimization problem

$$\min_x \|A_p x - y\|_2^2.$$

Due to variations in the local sampling density of points in the polar grid and its modification, weights are introduced to compensate for its effects. We can use the formula for weights in [102], [103]. The preconditioner or weights  $W$  is used in the following way, rather than minimizing  $\|A_p x - y\|_2^2$ , we use

$$\min_x \|W \circ A_p x - W \circ y\|_2^2$$

instead and we solve it iteratively using the relation of least squares (LS) solution

$$x_{k+1} = x_k - \mu A_p^H W^2 \circ (A_p x - y) \quad (4.25)$$

where  $\mu$  is the step size and  $k$  is the iteration number. The expression  $A_p^H (A_p x - y)$  is the function's gradient and the multiplication by a positive definite matrix  $W$  guarantees descent in the LS error. The specific choice of  $W$  as discussed in [102] is such that it down-weights the points near the

origin in order to equalize the condition number.

For corner compensation as discussed in the previous section, we consider two systems. Let  $A_{cc}$  be the operator that takes into account only the Cartesian corner points, i.e. the Fourier domain coefficients falling outside the range of the polar grid. Then, the two systems with operators  $A_p$  (representing the discrete polar form) and  $A_{cc}$  (representing the Cartesian corners) can be written as

$$A_p x = y \quad (4.26)$$

$$A_{cc} x = y_c, \quad (4.27)$$

where  $y_c$  is the set of Fourier domain information at the corners. Then using the augmented system  $[A_p : A_{cc}]$  with the augmented output  $[y : y_c]$ , the new symmetric positive definite linear system becomes

$$Ax = b \quad (4.28)$$

where

$$A = A_p^H W^2 \circ A_p + A_{cc}^H A_{cc}$$

$$b = A_p^H W^2 \circ y + A_{cc}^H y_c.$$

Possible ways around the corner compensation problem is to suppress/remove the fixed pattern noise by a suitably designed filtering process (similar to the filtered back projection) or simply assume that only the center of the image matters. If the given image is padded by a factor of  $\sqrt{2}$  along both axes, a 2D FFT is applied and 2D IFFT is followed, then the resultant image is twice as big in pixel count and its frequency domain support is concentrated in the center. These steps lead to additional  $(N + 1)^2 \log_2(N + 1)$  operations to the process, see Figure 4.9 for the reconstruction

with padding. From the reconstructed image using the corner compensation and extra padding by factor of  $\sqrt{2}$ , we get almost perfect reconstruction. Similar to the approach followed in [102], we assume in the simulation that there is no corner Fourier information present and the image is concentrated at its center, i.e. the values for vector  $y_c$  are all set to zero.

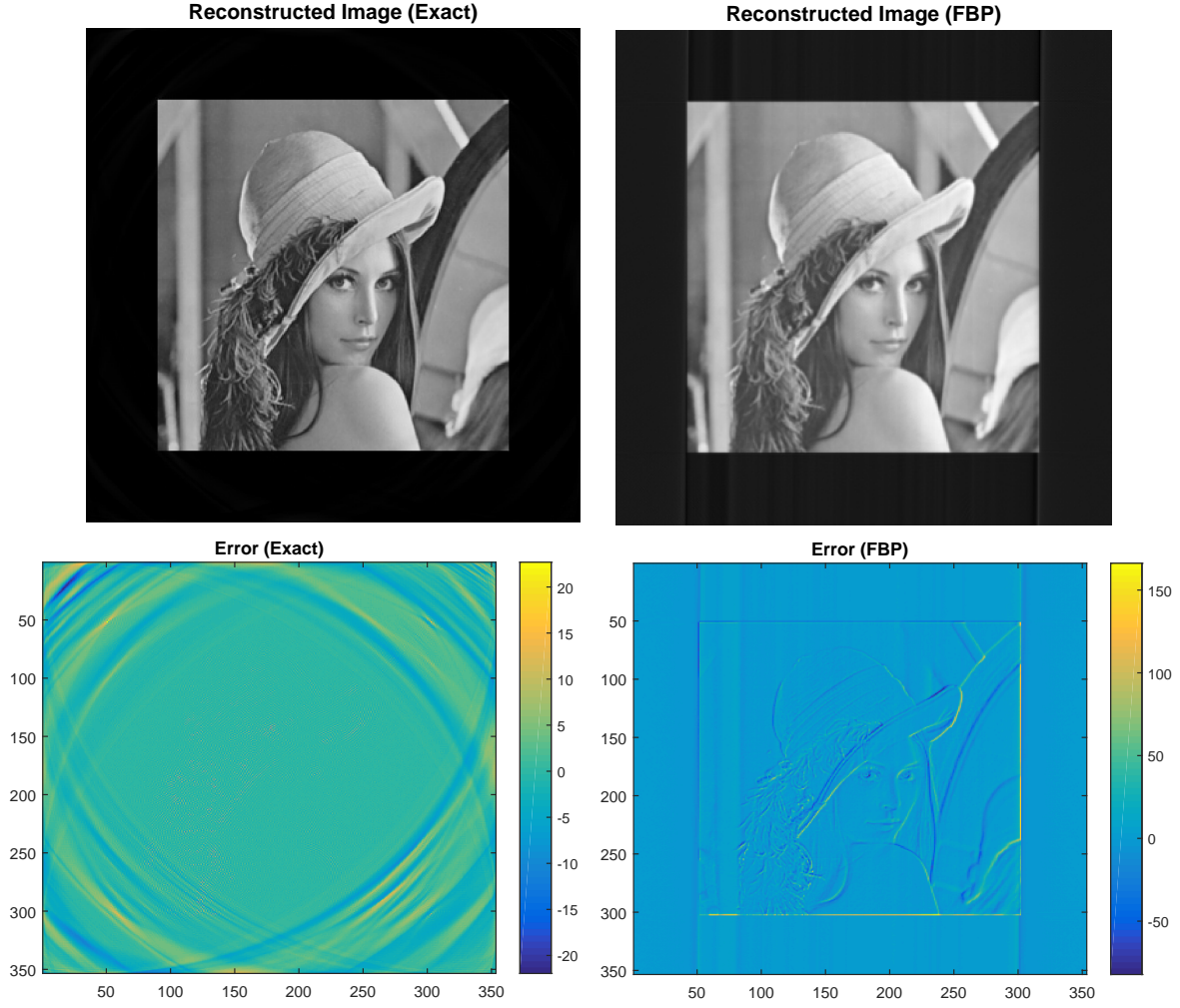


Figure 4.9: (Top row) Corner compensated and padded reconstruction, where  $M = 2(N + 1)$ , and residual error is  $= 8.7 \times 10^{-5}$  after just 12 iterations. We initialize the image with zeros for the first iteration, and we get a near perfect reconstruction at the image center. Observe that the sharp features have been destroyed by the use of filtered back projection method as it shows up in the error comparison. However, using the proposed method of exact discrete PFT we are able to retrieve those sharp features with the same sampling data. (Bottom row) Error comparison using the exact solution and the FBP approach by MATLAB implementation using functions `radon` and `iradon`.

#### 4.3.4 Faster Inversion by Structured Matrix Computations

In this subsection, we exploit the underlying special structure of the matrix  $A$  in (4.28) to get much faster solution than the one discussed in section 4.3.3.

Before we show the structured matrix computations for computing  $Ax$  for any vector  $x$ , we require a few definitions.

Definition 1:

An  $n \times n$  matrix  $T_n$  is called *Toeplitz* if it is constant along its diagonals, i.e. if

$$T_n = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{2-n} & t_{1-n} \\ t_1 & t_0 & \cdots & t_{3-n} & t_{2-n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_{n-2} & t_{n-3} & \cdots & t_0 & t_{-1} \\ t_{n-1} & t_{n-2} & \cdots & t_1 & t_0 \end{bmatrix} \quad (4.29)$$

Its entries are given by,  $T_n^{(l,m)} = t_{l-m}$ . Circulant matrices are a subclass of Toeplitz matrices, which play an essential role in general Toeplitz matrix calculations. It is well known that circulant matrices can be diagonalized efficiently using FFT.

Definition 2:

An  $n \times n$  matrix  $C_n$  is called *circulant* if it is Toeplitz and, in addition,  $c_{-k} = c_{n-k}$ .

Toeplitz matrices are very structured that arise in a variety of applications [147], for example in deblurring images, Radon transform and image reconstruction, the discretization process of partial differential equations, etc. The matrix  $A$  is a real symmetric two-level Toeplitz matrix. Block-Toeplitz-Toeplitz-block (BTTB) matrices or two-level Toeplitz matrices are the 2D analogues of Toeplitz matrices. A BTTB matrix is a block matrix with Toeplitz blocks, and it has Toeplitz struc-



ture on the block level. Block-circulant-circulant-block (BCCB) matrices are the 2D analogues of circulant matrices, which are diagonalized efficiently by the 2D FFT. With the concept of circulant embedding we can transform a real symmetric BTTB matrix into a larger real symmetric BCCB matrix. In higher dimensions such matrices have a nested multilevel block Toeplitz structure. Consider a BTTB system generated by a simple  $(N + 1) \times (N + 1)$  matrix with  $N = 2$ , since the entire real symmetric BTTB matrix can be represented by its first column,

$$Col_1 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

The BTTB system is of the form

$$Ax = b.$$

$$\left[ \begin{array}{ccc|ccc|ccc} a_{11} & a_{21} & a_{31} & a_{12} & a_{22} & a_{32} & a_{13} & a_{23} & a_{33} \\ a_{21} & a_{11} & a_{21} & a_{22} & a_{12} & a_{22} & a_{23} & a_{13} & a_{23} \\ a_{31} & a_{21} & a_{11} & a_{32} & a_{22} & a_{12} & a_{33} & a_{23} & a_{13} \\ \hline a_{12} & a_{22} & a_{32} & a_{11} & a_{21} & a_{31} & a_{12} & a_{22} & a_{32} \\ a_{22} & a_{12} & a_{22} & a_{21} & a_{11} & a_{21} & a_{22} & a_{12} & a_{22} \\ a_{32} & a_{22} & a_{12} & a_{31} & a_{21} & a_{11} & a_{32} & a_{22} & a_{12} \\ \hline a_{13} & a_{23} & a_{33} & a_{12} & a_{22} & a_{32} & a_{11} & a_{21} & a_{31} \\ a_{23} & a_{13} & a_{23} & a_{22} & a_{12} & a_{22} & a_{21} & a_{11} & a_{21} \\ a_{33} & a_{23} & a_{13} & a_{32} & a_{22} & a_{12} & a_{31} & a_{21} & a_{11} \end{array} \right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \\ b_9 \end{bmatrix}$$

The BTTB system in (4.28) can be extended to a BCCB system. If the input image stacked vector  $x$  is of size  $(N + 1)^2 \times 1$ , the BTTB matrix  $A$  is of size  $(N + 1)^2 \times (N + 1)^2$ . With the use of

circulant embedding we have the extended BCCB matrix  $\tilde{A}$  of size  $(2N)^2 \times (2N)^2$ ,

$$\tilde{A}\tilde{x} = \tilde{b}.$$

$$\begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{21} & a_{12} & a_{22} & a_{32} & a_{22} & a_{13} & a_{23} & a_{33} & a_{23} & a_{12} & a_{22} & a_{32} & a_{22} \\ a_{21} & a_{11} & a_{21} & a_{31} & a_{22} & a_{12} & a_{22} & a_{32} & a_{23} & a_{13} & a_{23} & a_{33} & a_{22} & a_{12} & a_{22} & a_{32} \\ a_{31} & a_{21} & a_{11} & a_{21} & a_{32} & a_{22} & a_{12} & a_{22} & a_{33} & a_{23} & a_{13} & a_{23} & a_{32} & a_{22} & a_{12} & a_{22} \\ a_{21} & a_{31} & a_{21} & a_{11} & a_{22} & a_{32} & a_{22} & a_{12} & a_{23} & a_{33} & a_{23} & a_{13} & a_{22} & a_{32} & a_{22} & a_{12} \\ \hline a_{12} & a_{22} & a_{32} & a_{22} & a_{11} & a_{21} & a_{31} & a_{21} & a_{12} & a_{22} & a_{32} & a_{22} & a_{13} & a_{23} & a_{33} & a_{23} \\ a_{22} & a_{12} & a_{22} & a_{32} & a_{21} & a_{11} & a_{21} & a_{31} & a_{22} & a_{12} & a_{22} & a_{32} & a_{23} & a_{13} & a_{23} & a_{33} \\ a_{32} & a_{22} & a_{12} & a_{22} & a_{31} & a_{21} & a_{11} & a_{21} & a_{32} & a_{22} & a_{12} & a_{22} & a_{33} & a_{23} & a_{13} & a_{23} \\ a_{22} & a_{32} & a_{22} & a_{12} & a_{21} & a_{31} & a_{21} & a_{11} & a_{22} & a_{32} & a_{22} & a_{12} & a_{23} & a_{33} & a_{23} & a_{13} \\ \hline a_{13} & a_{23} & a_{33} & a_{23} & a_{12} & a_{22} & a_{32} & a_{22} & a_{11} & a_{21} & a_{31} & a_{21} & a_{12} & a_{22} & a_{32} & a_{22} \\ a_{23} & a_{13} & a_{23} & a_{33} & a_{22} & a_{12} & a_{22} & a_{32} & a_{21} & a_{11} & a_{21} & a_{31} & a_{22} & a_{12} & a_{22} & a_{32} \\ a_{33} & a_{23} & a_{13} & a_{23} & a_{32} & a_{22} & a_{12} & a_{22} & a_{31} & a_{21} & a_{11} & a_{21} & a_{32} & a_{22} & a_{12} & a_{22} \\ a_{23} & a_{33} & a_{23} & a_{13} & a_{22} & a_{32} & a_{22} & a_{12} & a_{21} & a_{31} & a_{21} & a_{11} & a_{22} & a_{32} & a_{22} & a_{12} \\ \hline a_{12} & a_{22} & a_{32} & a_{22} & a_{13} & a_{23} & a_{33} & a_{23} & a_{12} & a_{22} & a_{32} & a_{22} & a_{11} & a_{21} & a_{31} & a_{21} \\ a_{22} & a_{12} & a_{22} & a_{32} & a_{23} & a_{13} & a_{23} & a_{33} & a_{22} & a_{12} & a_{22} & a_{32} & a_{21} & a_{11} & a_{21} & a_{31} \\ a_{32} & a_{22} & a_{12} & a_{22} & a_{33} & a_{23} & a_{13} & a_{23} & a_{32} & a_{22} & a_{12} & a_{22} & a_{31} & a_{21} & a_{11} & a_{21} \\ a_{22} & a_{32} & a_{22} & a_{12} & a_{23} & a_{33} & a_{23} & a_{13} & a_{22} & a_{32} & a_{22} & a_{12} & a_{21} & a_{31} & a_{21} & a_{11} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 0 \\ x_4 \\ x_5 \\ x_6 \\ 0 \\ x_7 \\ x_8 \\ x_9 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \hat{b}_4 \\ b_4 \\ b_5 \\ b_6 \\ \hat{b}_6 \\ b_7 \\ b_8 \\ b_9 \\ \hat{b}_9 \\ \hat{b}_{10} \\ \hat{b}_{11} \\ \hat{b}_{12} \\ \hat{b}_{13} \end{bmatrix}$$

Figure 4.10 shows the detailed structure of matrix  $A$  and its BCCB extension. A BCCB matrix is a normal matrix, thus a matrix  $\tilde{A}$  has a unitary spectral decomposition. It is well known that any BCCB matrix has the particular spectral decomposition

$$\tilde{A} = F^H \Sigma F$$

where  $F$  is the two dimensional unitary DFT matrix and the diagonal matrix  $\Sigma$  represents the eigenvalues of  $\tilde{A}$  [147]. The specialty of this computation lies in that neither  $F$  nor  $F^H$  need to be explicitly computed or stored. Once we know the spectral decomposition, we can efficiently perform matrix multiplication with a BCCB matrix as

$$\tilde{b} = \tilde{A}\tilde{x} = F^H \Sigma F \tilde{x},$$

i.e. the product can be computed by first taking 2D FFT of  $\tilde{x}$ , multiplying elementwise with the 2D FFT of the first column of  $\tilde{A}$  arranged in a 2D array, and finally performing 2D IFFT of the product to get the multiplication  $\tilde{A}\tilde{x}$ .

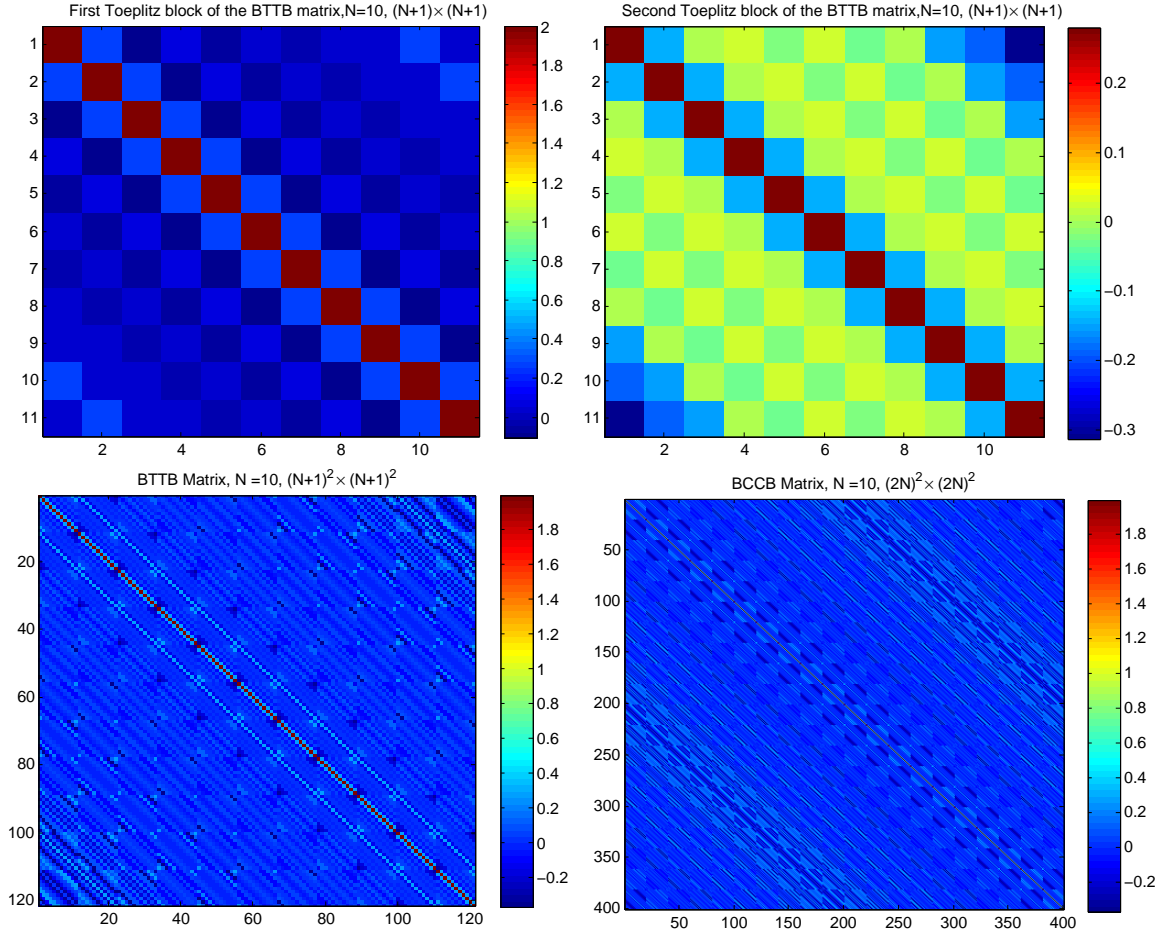


Figure 4.10: BTTB to BCCB conversion, performing extension of real symmetric BTTB matrix using concept of embedding.

For the computation of the system in (4.28) using the conjugate gradient algorithm, in each iteration we require an operation with  $A_p^H$  and an application of operator  $A_p$ , both operations require complexity of order  $O(M/4 \times (N+1)^2 \log(N+1))$ . The other operators of Cartesian corners are simply centered 2D FFT and 2D IFFT of order  $O((N+1)^2 \log(N+1))$ . Now, exploiting the BTTB structure of matrix  $A$  we get faster matrix vector multiplication  $Ax$ , such that only once the first column of  $A$  needs to be computed. The impulse response of the system  $A = A_p^H W^2 \circ A_p + A_{cc}^H A_{cc}$  gives

the first column of the BTTB matrix. Thus for each subsequent iteration of the conjugate gradient algorithm, the operation  $Ax$  for any arbitrary vector  $x$  is of complexity  $O((N + 1)^2 \log(N + 1))$ . Moreover the complexity is independent of the number of angles  $M$  but it only depends on the image size  $(N + 1)$ . Also, unlike the single level Toeplitz systems discussed in [113], [114] and [115] for the PP grid, which gives a direct inversion algorithm, higher level ( $\geq 2$ ) Toeplitz systems do not have a direct inverse solution of closed form, hence only preconditioned iterative techniques can be used. We have discussed the inverse solution without considerations to the noise, however sophisticated regularization techniques such as the T-SVD, or the Tikhonov method ([147]) can be used in the presence of noise for stable inversion.

The fast forward and the inverse transforms discussed above are based on an algebraically exact formulation with full geometric fidelity. Compared to the NFFT and the pseudo-polar grid approach, our solution provides the sparsest representation devoid of any unnecessary oversampling. The NFFT and the pseudo-polar grid both require oversampling factors larger than or equal to 2 both angularly and radially, and the available implementations ([102], [103], [111] and [112]) for discrete data are only vaguely related and not precise analogs to the continuous transforms.

#### 4.4 The Proposed Method- 3D

In this section, a similar strategy as discussed in section 4.2 is adopted for 3D setting, which has significantly more complexity and structure. Extension to 3D will require the definition of polar-like coordinates, and no exact polar sampling method for 3D is well defined [102]. Thus we must settle for polar-like grids.

It should be noted that obtaining DFT for 3D cylindrical coordinates, poses little challenge since its extension is trivial, this arrangement could be considered as exact 2D polar slices stacked along

the height of the cylinder. We define the cylindrical FT of a volume  $f$ , denoted by  $F^C$ , given by,

$$F^C(f) = F^P(F_z(f)) \quad (4.30)$$

where  $F_z$  is the centered 1D FT in the  $z$  direction and  $F^P$  is the 2D polar FT which operates on each plane that is perpendicular to the  $z$  axis. To compute the DFT for 3D cylindrical coordinates, compute 1-D centered FFT in the  $Z$  direction, then compute 2-D Polar FFT for each plane perpendicular to the  $Z$ -axis. If  $K$  is the height of cylinder then taking a 2D polar slice at each node of complexity given by (4.16), we will have total computational complexity as  $K \times M/4 \times (N+1)^2 \log_2(N+1)$ .

For DFT using spherical coordinates, we follow a simple definition of spherical transformation and show that it has a different structure compared to 2D, such that more radial lines can be computed with a single appropriately scaled grid in 3D.

Consider the transformation of 3D rectangular coordinates to 3D *spherical* or *spherical polar* coordinates, where  $(x, y, z) \rightarrow (\rho, \theta, \phi)$ ,  $x = \rho \cos \theta \cos \phi$ ,  $y = \rho \cos \theta \sin \phi$  and  $z = \rho \sin \theta$ , the radius  $\rho \in [0, \infty]$ , polar angle or *zenith* or elevation  $\theta \in [0, 180]^\circ$  and *azimuth* angle  $\phi \in [0, 360]^\circ$ . We follow the convention used in physics and also in the spherical harmonics notations (computer graphics) rather than the ones used in pure mathematics, where the roles of  $\theta$  and  $\phi$  are reversed [148]. We measure the azimuth angle with respect to the  $x$ -axis, also as it is standard in the 3D software systems (to keep the equations and the implementation consistent) we measure the zenith angles from the  $xy$ -plane rather than with respect to  $z$  axis, see Figure 4.11.

Given the Cartesian 3D matrix data  $f(r, c, d)$ , where  $r, c, d$  represent rows, columns and depths indices respectively, we define the *Spherical Polar Fourier Transform* (SPFT) to be the collection

of samples  $F(k, m, n)$ , similar to 2D form in (4.6),

$$F(k, m, n) = \sum_{d=-N/2}^{N/2} \sum_{c=-N/2}^{N/2} \sum_{r=-N/2}^{N/2} f(r, c, d) \times e^{-i \frac{2\pi n}{N+1} (r \cos \theta_k \cos \phi_m + c \cos \theta_k \sin \phi_m + d \sin \theta_k)}, \quad (4.31)$$

where  $-N/2 \leq n \leq N/2$  represents the points on a radial line,  $\theta_k = k\Delta\theta, 0 \leq k \leq K-1$  is the zenith angles,  $\Delta\theta = 180^\circ/K$  is the angular spacing in zenith angles,  $\phi_m = m\Delta\phi, 0 \leq m \leq M-1$  means the azimuth angles,  $\Delta\phi = 180^\circ/M$  is the angular spacing in azimuth angles, and  $K \times M$  is the total number of radial lines. Thus the total number of DFT points is  $K \times M \times (N+1)$ , where the origin is included multiple times. Similarly, we can define the adjoint operator of the SPFT as follows

$$\hat{f}(r, c, d) = \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} \sum_{n=-N/2}^{N/2} F(k, m, n) \times e^{+i \frac{2\pi n}{N+1} (r \cos \theta_k \cos \phi_m + c \cos \theta_k \sin \phi_m + d \sin \theta_k)}. \quad (4.32)$$

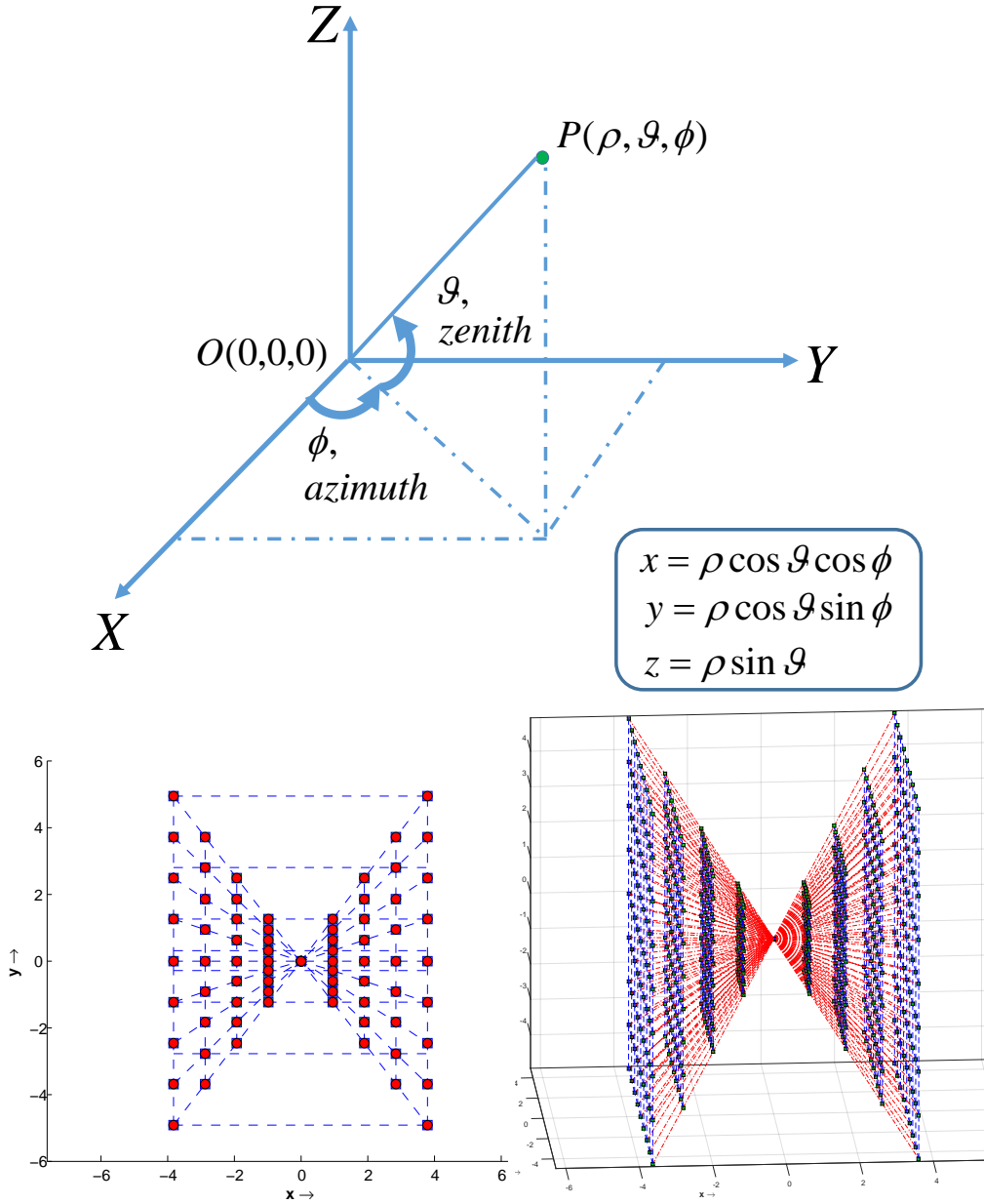


Figure 4.11: Spherical coordinate conventions (top), butterfly grid 2D (bottom left) and 3D (bottom right). Conventions followed here are shown. We measure the *azimuth* angle with respect to the  $x$ -axis and *zenith* angle with respect to the  $xy$ -plane. In 2D, the butterfly grid is a set of concentric rectangles and in 3D it is a set of concentric cuboids, both obtained by applying 1D FrFT on select axes. Only one axis is uniformly scaled with 1D FrFT, the other two axes are non-uniformly scaled.



The butterfly grid was crucial for the computation of radial lines in the 2D case as shown in Figure 4.2, because this grid has special selection of points that exactly match the positions of the desired radial points of the polar grid. We propose a 3D butterfly grid, a set of concentric points arranged in the 3D square frustum, see Figure 4.11. For the 3D butterfly grid only one axis is uniformly scaled in the frequency domain using 1D FrFT, the other two axes are non-uniformly scaled, where the scale factors for all three axes are determined by the geometrical position of the radial line. For the 3D butterfly grid, we only compute the Fourier domain information on the desired points where it overlaps with the radial lines of the spherical grid and not the entire  $(N + 1)^3$  Cartesian grid.

Similar to the partitioning in terms of levels in 2D, let  $p$  represent the level number for angle  $\theta$ ,  $q$  represent the level number for angle  $\phi$ , then  $1 \leq p \leq P$  and  $1 \leq q \leq Q$ , where  $P$  is the number of levels in angles  $\theta$  and  $Q$  is the number of levels in angles  $\phi$ , see Figure 4.12 for all the transformations of Cartesian to spherical coordinates for all angles. Inspired by the 2D polar divisions in terms of basically horizontal (X-axis) and basically vertical (Y-axis) lines,

- we first select two polar slices oriented along the X-axis with fixed angles  $q\Delta\phi$  and  $180 - q\Delta\phi$ ;
- from these two polar slices, we then select two planes oriented along the X-axis with angles  $p\Delta\theta$  and  $180 - p\Delta\theta$ ; and
- finally from these two planes we choose four lines that overlap exactly with the four unique radial lines of the spherical grid.

Since we have orientation for both angles  $\theta$  and  $\phi$  along the  $X$  axis, we call these four lines as forming a  $XX$  block. With a proper choice of scale factors we can compute a butterfly grid with this orientation that would overlap exactly with these four radial lines. Similarly, using a combination of complementary angles we can get  $XZ$ ,  $YY$  and  $YZ$  blocks and compute a butterfly

grid for each block containing four unique lines that belong to the spherical grid. This type of divisions of radial lines is crucial for determining the proper scale choices and the desired fast computations with 1D FrFT for the 3D spherical grid, see Figure 4.13 for the schematic of the spherical divisions. Note that the different scale factors chosen for different block arrangements are important, and so is the order of application of scale factors, because the first scale factor for each block is the uniform scaling factor for 1D FrFT and it is the most expensive one to calculate.

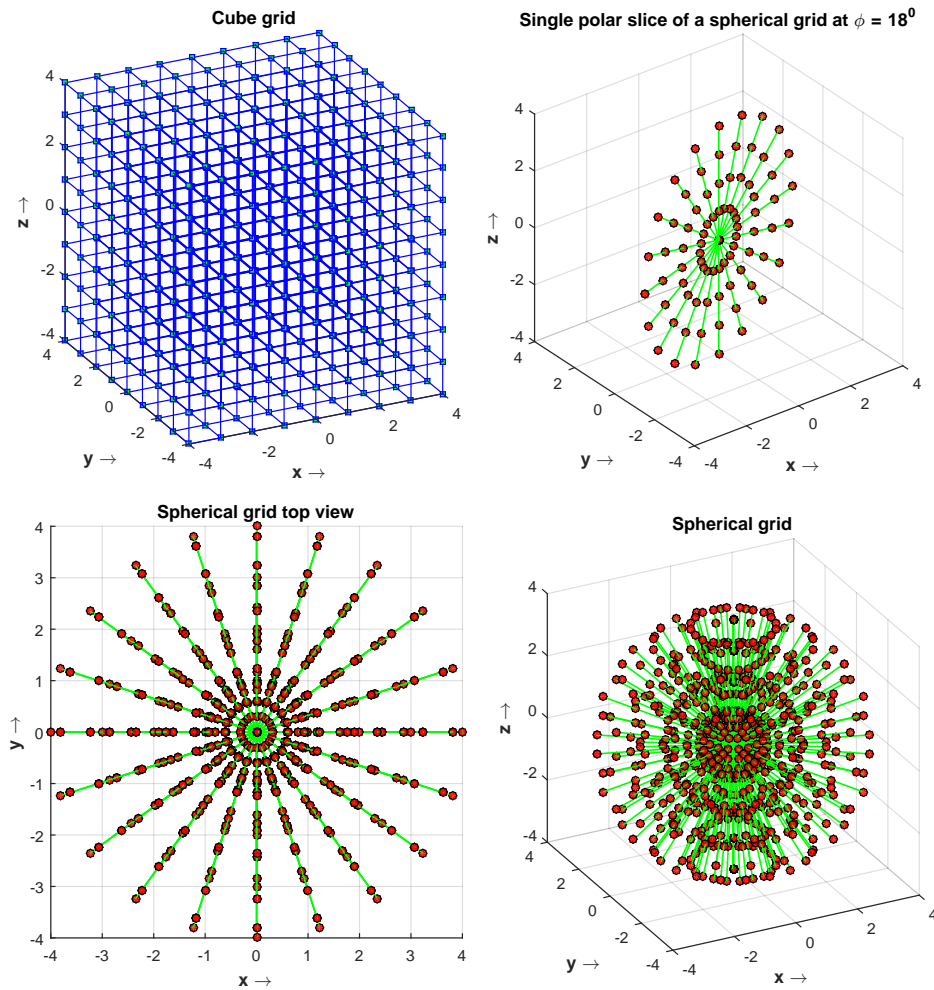


Figure 4.12: 3D transformations required from spatial domain rectangular coordinates of image size  $(N + 1)^3$  to Fourier domain spherical coordinates grid of size  $K \times M \times (N + 1)$  where the origin is included multiple times.

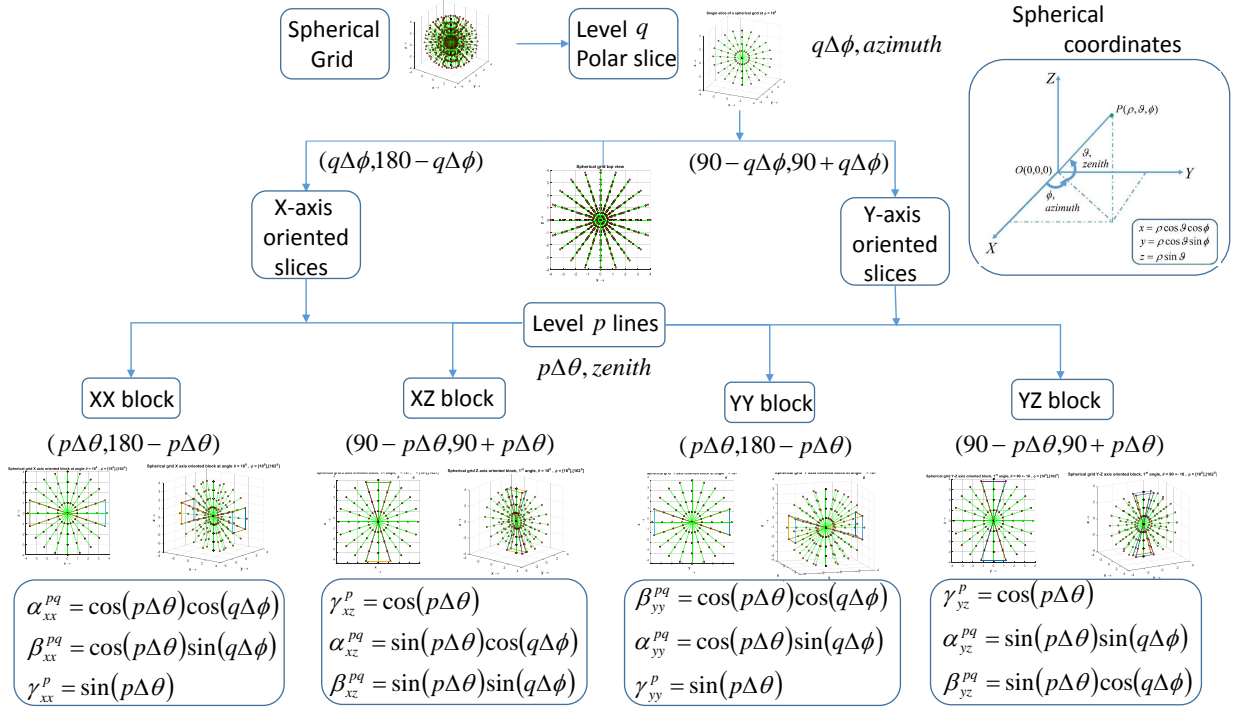


Figure 4.13: Spherical divisions required for fast and exact computation of Spherical FT.

For the  $XX$  block, the 1D frequency domain scaling of a 3D image  $f$  along the  $X$  axis oriented polar slices can be computed using a 1D FrFT along the  $X$  axis (we slightly change our notation and indexing here compared to 2D equations) such that

$$\begin{aligned} F^{\alpha_{xx}^{pq}}(r, c, d) &= \sum_{n=-N/2}^{N/2} f(n, c, d) e^{-j \frac{2\pi r \alpha_{xx}^{pq} n}{N+1}} \\ \alpha_{xx}^{pq} &= \cos(p\Delta\theta) \cos(q\Delta\phi), \end{aligned} \quad (4.33)$$

where  $-N/2 \leq r, c, d \leq N/2$ .

Next, we scale the transformed image  $F^{\alpha_{xx}^{pq}}$  in  $Y$  axis and compute 1D FrFT. Now the equation

becomes

$$\begin{aligned}
F^{\alpha_{xx}^{pq}, \beta_{xx}^{pq}}(r, c, d) &= \sum_{n=-N/2}^{N/2} F^{\alpha_{xx}^{pq}}(r, n, d) e^{-j \frac{2\pi c |c| \beta_{xx}^{pq} n}{q(N+1)}} \\
\beta_{xx}^{pq} &= \cos(p\Delta\theta) \sin(q\Delta\phi),
\end{aligned} \tag{4.34}$$

where  $-N/2 \leq r, c, d \leq N/2$ .

Finally, we scale the transformed image  $F^{\alpha_{xx}^{pq}, \beta_{xx}^{pq}}$  in  $Z$  axis,

$$\begin{aligned}
F^{\alpha_{xx}^{pq}, \beta_{xx}^{pq}, \gamma_{xx}^p}(r, c, d) &= \sum_{n=-N/2}^{N/2} F^{\alpha_{xx}^{pq}, \beta_{xx}^{pq}}(r, c, n) e^{-j \frac{2\pi d |d| \gamma_{xx}^p n}{p(N+1)}} \\
\gamma_{xx}^p &= \sin(p\Delta\theta),
\end{aligned} \tag{4.35}$$

where  $-N/2 \leq r, c, d \leq N/2$ .

Figure 4.14 demonstrates the  $XX$  blocks with  $p = 1, q = 1$  and  $p = 2, q = 2$ , respectively. We are only concerned with computing 2 complementary planes from (4.34), since the goal is to eventually compute four lines in these two planes. After this operation we get a pair of complementary 2D planes of size  $(N + 1)^2$ . Now, using the scaling in (4.35), we compute two complementary lines for each plane. Those are the desired four radial lines forming the complete  $XX$  block, which is similar to the one shown in Figure 4.14(a).

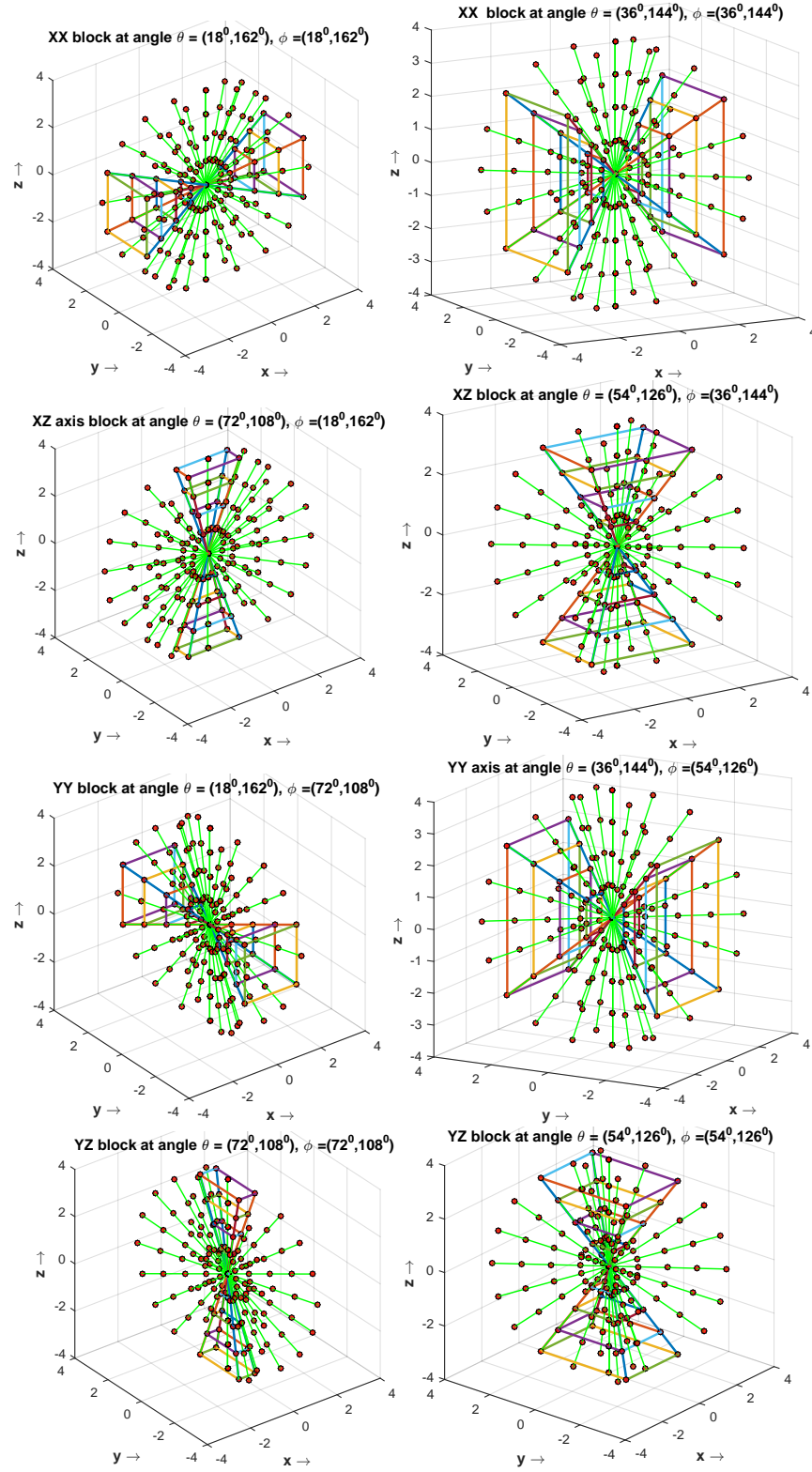


Figure 4.14: 3D blocks at level  $p = 1, q = 1$  (column left) and  $p = 2, q = 2$  (column right) needed for fast computation of Spherical FT (from top to bottom) (a) XX-block (b) XZ-block (c) YY-block and (d) YZ-block.

For  $XZ$  oriented blocks, first we uniformly scale 3-D image  $f$  along  $Z$  axis

$$\begin{aligned} F^{\gamma_{xz}^p}(r, c, d) &= \sum_{n=-N/2}^{N/2} f(r, c, n) e^{-j \frac{2\pi d \gamma_{xz}^p n}{N+1}} \\ \gamma_{xz}^p &= \cos(p\Delta\theta). \end{aligned} \quad (4.36)$$

Next we scale the transformed image  $F^{\gamma_{xz}^p}$  in  $X$  axis using variable scale

$$\begin{aligned} F^{\gamma_{xz}^p, \alpha_{xz}^{pq}}(r, c, d) &= \sum_{n=-N/2}^{N/2} F^{\gamma_{xz}^p}(n, c, d) e^{-j \frac{2\pi r |r| \alpha_{xz}^{pq} n}{p(N+1)}} \\ \alpha_{xz}^{pq} &= \sin(p\Delta\theta) \cos(q\Delta\phi). \end{aligned} \quad (4.37)$$

Finally we scale the transformed image  $F^{\gamma_{xz}^p, \alpha_{xz}^{pq}}$  in  $Y$  axis using variable scale

$$\begin{aligned} F^{\gamma_{xz}^p, \alpha_{xz}^{pq}, \beta_{xz}^{pq}}(r, c, d) &= \sum_{n=-N/2}^{N/2} F^{\gamma_{xz}^p, \alpha_{xz}^{pq}}(r, n, d) e^{-j \frac{2\pi c |c| \beta_{xz}^{pq} n}{q(N+1)}} \\ \beta_{xz}^{pq} &= \sin(p\Delta\theta) \sin(q\Delta\phi), \end{aligned} \quad (4.38)$$

where  $-N/2 \leq r, c, d \leq N/2$ . Figure 4.14(b) demonstrates the  $XZ$  blocks with  $p = 1, q = 1$  and  $p = 2, q = 2$  respectively. Similar to the  $XX$  block we can use the above equations (4.36)–(4.38) to compute for the 4 radial lines for each pair  $p, q$  with this orientation.

Similarly, for  $YY$  block we perform 1D frequency domain uniform scaling of the 3D image  $f$  along  $Y$  axis oriented polar slices such that

$$\begin{aligned} F^{\beta_{yy}^{pq}}(r, c, d) &= \sum_{n=-N/2}^{N/2} f(r, n, d) e^{-j \frac{2\pi c \beta_{yy}^{pq} n}{N+1}} \\ \beta_{xx}^{pq} &= \cos(p\Delta\theta) \cos(q\Delta\phi). \end{aligned} \quad (4.39)$$

Next we can scale the transformed image  $F^{\beta_{yy}^{pq}}$ , in  $X$  axis, we can compute 1D FrFT along each

column and the equation is given by

$$\begin{aligned}
F^{\beta_{yy}^{pq}, \alpha_{yy}^{pq}}(r, c, d) &= \sum_{n=-N/2}^{N/2} F^{\alpha_{yy}^{pq}}(n, c, d) e^{-j \frac{2\pi r |r| \alpha_{yy}^{pq} n}{q(N+1)}} \\
\alpha_{yy}^{pq} &= \cos(p\Delta\theta) \sin(q\Delta\phi).
\end{aligned} \tag{4.40}$$

Finally we scale the transformed image  $F^{\beta_{yy}^{pq}, \alpha_{yy}^{pq}}$  in  $Z$  axis

$$\begin{aligned}
F^{\beta_{yy}^{pq}, \alpha_{yy}^{pq}, \gamma_{yy}^p}(r, c, d) &= \sum_{n=-N/2}^{N/2} F^{\beta_{yy}^{pq}, \alpha_{yy}^{pq}}(r, c, n) e^{-j \frac{2\pi d |d| \gamma_{yy}^p n}{p(N+1)}} \\
\gamma_{yy}^p &= \sin(p\Delta\theta),
\end{aligned} \tag{4.41}$$

where  $-N/2 \leq r, c, d \leq N/2$ . Figure 4.14(c) demonstrates the  $YY$  blocks with  $p = 1, q = 1$  and  $p = 2, q = 2$  respectively. Similar to the  $XX$  block we can use the above equations (4.39)–(4.41) to compute for the 4 radial lines for each pair  $p, q$  with this orientation.

For  $YZ$  block first we uniformly scale 3-D image  $f$  along  $Z$  axis

$$\begin{aligned}
F^{\gamma_{yz}^p}(r, c, d) &= \sum_{n=-N/2}^{N/2} f(r, c, n) e^{-j \frac{2\pi d |d| \gamma_{yz}^p n}{N+1}} \\
\gamma_{yz}^p &= \cos(p\Delta\theta).
\end{aligned} \tag{4.42}$$

Next we scale the transformed data  $F^{\gamma_{yz}^p}$  in  $X$  axis

$$\begin{aligned}
F^{\gamma_{yz}^p, \alpha_{yz}^{pq}}(r, c, d) &= \sum_{n=-N/2}^{N/2} F^{\gamma_{yz}^p}(n, c, d) e^{-j \frac{2\pi r |r| \alpha_{yz}^{pq} n}{q(N+1)}} \\
\alpha_{yz}^{pq} &= \sin(p\Delta\theta) \sin(q\Delta\phi).
\end{aligned} \tag{4.43}$$

Finally we scale the transformed image  $F^{\gamma_{yz}^p, \alpha_{yz}^{pq}}$  in  $Y$  axis

$$F^{\gamma_{yz}^p, \alpha_{yz}^{pq}, \beta_{yz}^{pq}}(r, c, d) = \sum_{n=-N/2}^{N/2} F^{\gamma_{yz}^p, \alpha_{yz}^{pq}}(r, n, d) e^{-j \frac{2\pi c |c| \beta_{yz}^{pq} n}{p(N+1)}}$$

$$\beta_{yz}^{pq} = \sin(p\Delta\theta) \cos(q\Delta\phi), \quad (4.44)$$

where  $-N/2 \leq r, c, d \leq N/2$ . Figure 4.14(d) demonstrates the  $YZ$  blocks with  $p = 1, q = 1$  and  $p = 2, q = 2$  respectively. Finally we can use equations (4.42)–(4.44) to compute for the 4 radial lines for each pair  $p, q$  with their respective orientation.

We then repeat the above process for the  $XZ$ ,  $YY$ , and  $YZ$  blocks to obtain a complete set of scales for the computation of the spherical polar grid (see Figures 4.13, 4.14 and the above equations for detailed formulas), given by

$$S = \{\alpha_{xx}^{pq}, \beta_{xx}^{pq}, \gamma_{xx}^p, \gamma_{xz}^p, \alpha_{xz}^{pq}, \beta_{xz}^{pq}, \beta_{yy}^{pq}, \alpha_{yy}^{pq},$$

$$\gamma_{yy}^p, \gamma_{yz}^p, \alpha_{yz}^{pq}, \beta_{yz}^{pq}, 1 \leq p \leq P, 1 \leq q \leq Q\},$$

where the cardinality of the set  $S$  is  $12PQ$ . Let the collection of all the radial lines oriented by the configuration of  $XX$  block be represented by  $F_{xx}$ . By choosing the appropriate scales from the set  $S$ , we have

$$F_{xx} = \bigcup_{p=1}^P \bigcup_{q=1}^Q F^{\alpha_{xx}^{pq}, \beta_{xx}^{pq}, \gamma_{xx}^p}. \quad (4.45)$$



Similarly, we can write the equations for the other three blocks as follows:

$$\begin{aligned}
F_{xz} &= \bigcup_{p=1}^P \bigcup_{q=1}^Q F^{\gamma_{xz}^p, \alpha_{xz}^{pq}, \beta_{xz}^{pq}}, \\
F_{yy} &= \bigcup_{p=1}^P \bigcup_{q=1}^Q F^{\beta_{yy}^{pq}, \alpha_{yy}^{pq}, \gamma_{yy}^p}, \\
F_{yz} &= \bigcup_{p=1}^P \bigcup_{q=1}^Q F^{\gamma_{yz}^p, \alpha_{yz}^{pq}, \beta_{yz}^{pq}}.
\end{aligned} \tag{4.46}$$

At  $\phi = 0^\circ$ , we have a single polar 2D slice (no block arrangement for this one), which we denote as  $F_x$ , and similarly, at  $\phi = 90^\circ$  we have another slice denoted by  $F_y$ , both of these slices can be computed using the 2D polar FT in (4.6). Then the complete 3D SPFT information on the spherical grid defined in (4.31), using equations (4.33)–(4.46), is given by

$$F = F_{xx} \bigcup F_{xz} \bigcup F_{yy} \bigcup F_{yz} \bigcup F_x \bigcup F_y. \tag{4.47}$$

To find the computational complexity of the above scheme, we follow a similar approach as in (4.16). The number of levels  $P$  required are approximately  $P = (K - 2)/4$ , and the number of levels  $Q$  is approximately  $Q = (M - 2)/4$ . At every level the most expensive operation is the first uniform scaling with arbitrary scale along the  $X$ ,  $Y$  and  $Z$  axes, while the other two computations involve 2 planes and 4 radial lines, which are with far less complexity. Therefore, we can compute the 1D frequency domain uniform scaling with 1D FrFT along any axis of any scale, and the corresponding complexity is

$$(N + 1)^3/2 \times \log_2(N + 1).$$

Note that out of the 4 blocks, the first uniform scaling for both blocks  $XZ$  and  $YZ$  are the same. Therefore, we can use the same uniformly scaled grid with 1D FrFT along the  $Z$  axis for computations of the  $XZ$  and  $YZ$  blocks (the other scales still differ). Since there are now only 3 blocks

corresponding to the three axis  $X, Y$  and  $Z$  with unique uniform scaling for each level, we have the reduced complexity with this sort of spherical divisions given by

$$3 \times \frac{M}{4} \times \frac{K}{4} \times \frac{(N+1)^3}{2} \log_2(N+1). \quad (4.48)$$

Figure 4.15 shows the execution times for computations of discrete SPFT, and Figure 4.16 shows a simple 3D reconstruction example using the proposed solution. This successfully demonstrates that the application of Forward and Inverse SPFT. The inversion process is again accelerated by exploiting the nested 2-level deep block Toeplitz structure of the product of the transform operators similar to the one discussed in Section 4.3. Since this is a simplified case where the signal (volume) is binary and well-padded with zeros, the reconstruction error is zero, however in the real world problems, partial data, distortions, additive noise, sampling errors and other sources of noise will cause reconstruction errors.

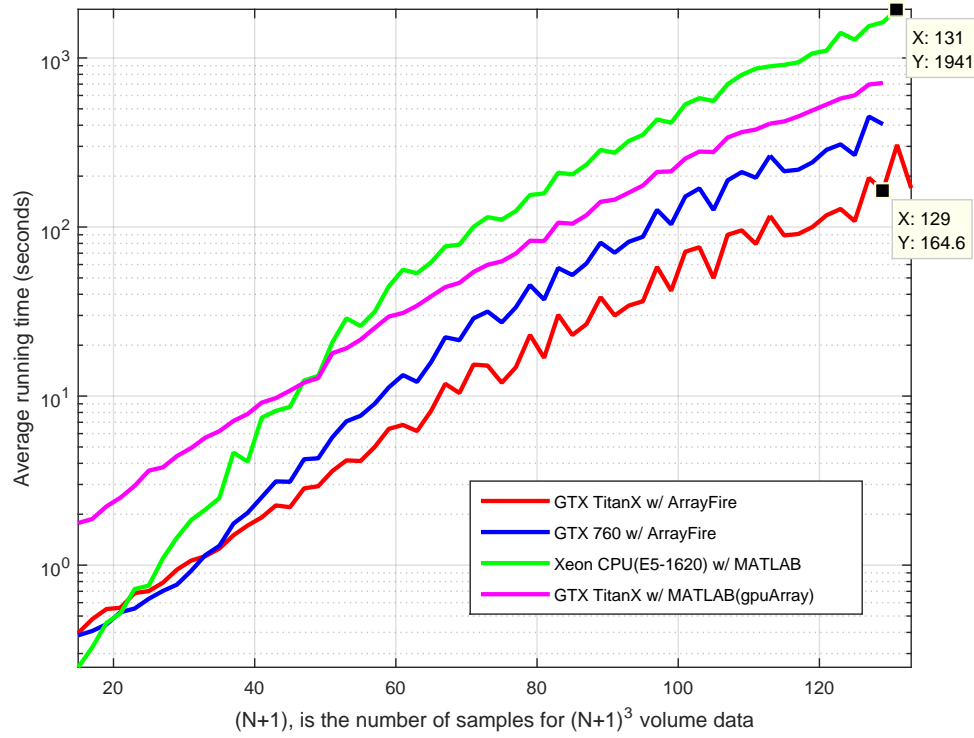


Figure 4.15: A comparison for the execution times vs the input array size given by  $N$ . The brute force solution for  $N = 128$ ,  $(N + 1)^3$  volume is of complexity  $O((N + 1)^6)$ , and it takes about 4 hours on a single GeForce Titan X GPU, albeitto much longer on a CPU, but it just takes 32.35 minutes, using the proposed fast vectorized implementation on a CPU with reduced complexity. The best computational time is achieved with GPU accelerated library (CUDA) using ArrayFire [151] which is just, 2.7333 minutes on a single Titan X GPU. The proposed solution is orders of magnitude faster, while maintaining the similar level of accuracy as the brute force method. Faster solution for larger  $N$  can be achieved with additional parallelization as the proposed technique can be fully distributed and scales naturally with the computational resources.

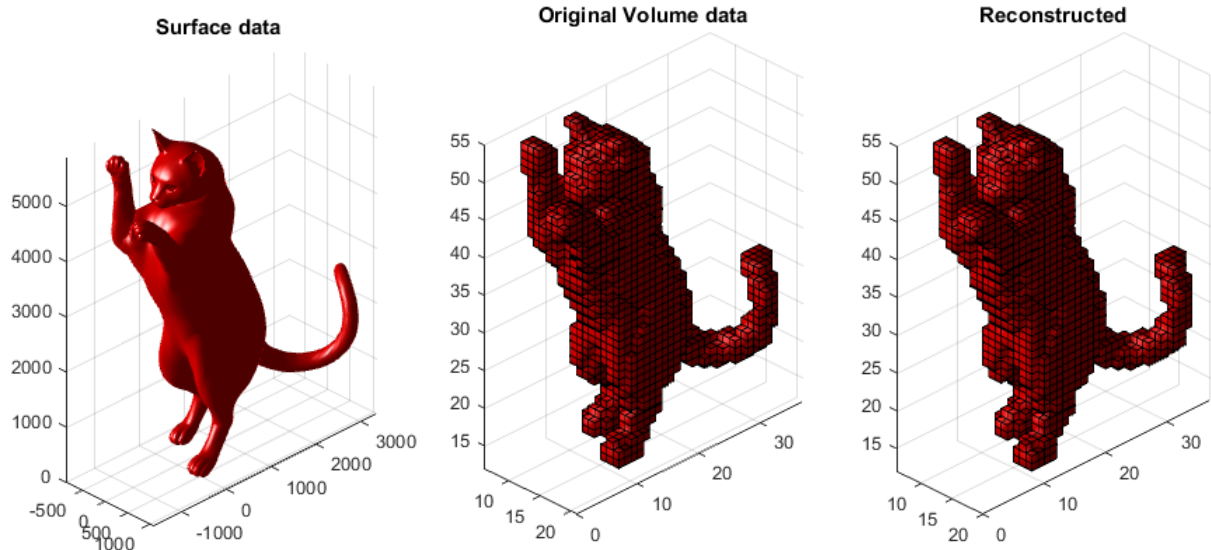


Figure 4.16: Simple 3D reconstruction from a surface data (with given faces and vertices which is easier to visualize) transformed into volumetric data,  $N = 54$ , i.e. voxel data size is  $55 \times 55 \times 55$ , with  $M = N + 14$ , and  $K = N + 10$ . We obtain perfect reconstruction of surface data by simply thresholding the values for the reconstructed volume. This is simply due to the fact that the signal/volume is in binary form and well-padded with zeros i.e. the signal is concentrated in the center of the volume hence the corner artifacts are eliminated.

Once again it should be emphasized that the computation of different levels is independent, and it can be implemented in parallel. Applications of the 3D FT is of particular importance for many numerical simulations (e.g. molecular dynamics) used in high performance computing codes. The parallelization of the 3D FFT is usually done using a 1D or 2D decomposition of the input data (see [149]), for our computations of the 3D SPFT also we have done a fully vectorized decomposition using 1D data only. So the proposed solution can be used in the areas of scientific computing where the computation of discrete, fast and exact solution for the 3D SPFT is required. The discrete definition of our algorithm can be used for Radon transform and X-ray transform both of which are established through the well-known Fourier slice theorem. The 2D PFT and 3D SPFT transforms will be useful for many applications, for instance, the rapid and exact Fourier volume

rendering used in [150] and image registration in 2D/3D [183].

#### 4.5 Contributions and Conclusions

Many algorithms in digital signal processing are based on the use of linear discrete signal transforms. Mathematically, such a transform is a matrix-vector multiplication  $y = Ax$  where  $x$  is the sampled signal, and  $A$  is the transform over some base field, which in this paper represents 2D and 3D transforms reduced in terms of 1D. Each of these algorithms has been derived through exploiting the geometric structures that arise in 2D and 3D and insightful algebraic manipulation of the transform matrix entries.

The main contributions of the paper are as follows:

- We have designed a highly specialized method to find the discrete FT on the full polar grid and the spherical grid for images in the 2D and 3D spatial domain Cartesian coordinates.
- The forward 2D and 3D algorithms are exact and fast without any oversampling radially or angularly, and they do not require interpolation in the spatial or the frequency domain, or any approximation stage.
- The fully parallel algorithms for both the polar and spherical grids can take full advantage of commodity GPGPU processing hardware. Most importantly each of the components used in our solution has a parallelizable structure and scales naturally with processing and memory resources.
- The proposed solutions of PFT and SPFT solve the problem of inversion by iteratively operating on forward and adjoint transforms. They are based on algebraically exact computations, and they are computed rapidly by using preconditioned conjugate gradient solution to

get accurate inversion.

- We drastically improve the conditioning of the inversion operation by using the proposed hybrid grid - Polar/Spherical Polar samples with Cartesian corners.
- We exploit the special matrix structure of block Toeplitz form, and we provide a solution to do a faster inversion.

Further research is needed to do a quantitative analysis taking in consideration different problem sizes, scalability, network constraints etc. More research is also needed to study these problems and transforms in the framework of modern algebra representation theory [152], which extends the relationship between signal processing and (abstract) algebra.

## **CHAPTER 5: ROBUST 3D REGISTRATION USING SPHERICAL POLAR FOURIER TRANSFORM AND SPHERICAL HARMONICS**

It is forbidden to kill; therefore all murderers are punished,  
unless they kill in large numbers and to the sound of trumpets.

*François-Marie Arouet, Voltaire, 1694 – 1778*

If an injury has to be done to a man it should be so severe that his  
vengeance need not be feared.

Niccolo Machiavelli , 1469 – 1527

The best argument against democracy is a five-minute conversa-  
tion with the average voter.

Winston Churchill, 1908 – 1965

...Of sufferings the poor have many,  
Which never can the rich annoy...

*Mary Ann Lamb, The Two Boys, 1764 – 1847*

The analysis of affine transformations in the frequency domain exploits the well-known property of separability. Whereby the translational displacement in this domain can be factored out and separately estimated from its affine matrix. This chapter presents a new and effective method for estimating up to 7 degrees of freedom (DOF), i.e. similarity transformation, for 3D volume registration using Fourier transformed data. The proposed method provides a closed-form solution

for the 3D rotation estimation problem, which is the most difficult part in a 3D similarity transformation, without any recurrence relations or searching for point correspondences between the two volumes/objects. We present the spectral registration technique using a fast and algebraically exact discrete 3D spherical polar Fourier transform, which gives us the spectral data defined on the spheres at multiple radii. Complete stacks of such layers are processed to find the best 3D rotation parameters using correlations on spheres with spherical harmonics, making it very accurate and robust to large rotations, noise, holes, missing data, and partial overlap. We compare the proposed technique with the state-of-the-art methods for rotation estimation and demonstrate the effectiveness of our solution with extensive experimentation, including a set of scanned MRI images, a crashed car parking dataset, and the Princeton shape benchmark dataset with hundreds of 3D objects.

## 5.1 Introduction

Image or data registration is the process of spatially aligning two or more datasets of an object or a scene. The goal is to estimate the most optimal geometric transformation between 3D point clouds or volumes acquired by different sensors or from different viewpoints, with the goal of enabling the fusion of information and/or the estimation of the motion of sensors.

Registration is a widely studied problem that still remains an important topic of research, with a diverse range of algorithms and methods proposed in the literature for volume/point-set registration, see [153]-[166], and references therein.

It is, for instance, crucial for object pose alignment in computer vision and used for registration of volumetric data in medical imaging. It is also used for a wide range of tasks such as object grasping, manipulation, object identification, self-localization, the interpolation of trajectory of



3D orientations, robot kinematics, flight simulation, structure from motion, 3D pose recovery of objects, and motion capture in computer vision, and robotics.

The algorithms proposed in the literature could be classified into *spatial domain* and *frequency domain* methods depending on whether the registration algorithm is operated using spatial or frequency information. While the frequency domain methods are mostly global feature-less registration algorithms, the spatial domain methods can be further classified into *local* and *global* feature based methods. In this paper, we focus only on the global frequency domain methods.

A vast amount of literature is present for registration techniques in the Fourier transformed domain. The Fourier spectrum information acts as a global, feature-less and robust descriptor [100] of a signal (e.g. 3D volume), it has the following advantages over the spatial registration algorithms:

- **Robustness:** Fourier based methods are robust and are known to be more stable to additive noises, distortions, spatial and time varying illumination disturbances, since they use all the information available rather than the content associated with only a sets of features.
- **Reliability:** Compared to feature based methods, they are less sensitive to any or specific control parameters. In a sense, the parameters in Fourier based registration techniques determine only the runtime and precision.
- **Seperability:** As it is well known, due to the nature of the FT, we can achieve full separability of affine matrix and the translation parameters of a complete affine transformation, which is not possible with spatial domain techniques, see the 2D affine theorem in [167], [168] and the *Fourier shift* theorem.
- **Speed:** They have a low computational complexity, require no initializations, and take a fixed time period for registration.

- **Versatility:** The phase correlation technique known as the phase-only matched filter (POMF) [169]-[127], is adopted in frequency based registration algorithms to deal with partially overlapped datasets [158].

For a Euclidean 3D transformation between the two volumes/objects, the problem of estimating the rotation between two 3D objects is still under active research since issues such as missing data, partial overlap, noisy data, and objects with intraclass variations and symmetries have not been completely solved [171]. Some techniques are better suited to deal with only surface data [171], while others are designed for a full volumetric data [172]. In this chapter, we focus on the volume data since surface data can be considered a subset of it. In 3D robotic mapping, the localization of the poses of the 3D scans that are to be registered are not known *a priori*, and when the robot is not on a wheeled platform operating on a smooth ground, the odometry motion estimates are either not available or are unreliable [172]. Due to the large amounts of 3D geometric points or inconsistencies between object structures, direct use and comparison of 3D points may not be suitable for rotation estimation [172]. In this context of directly using 3D points, the well known Iterated Closest Point (ICP) algorithm [173] is most popular, see also [174] and [175] for the recent development. But such spatial registration approaches have known disadvantages as discussed in the literature [171],[172], see also [161] and [162] for details in the context of 3D point sets. In general the methods that take the global appearance of the 3D scans, such as Fourier based methods, see [158],[164],[165],[171],[172],[176]-[181], are considered to be more robust especially in the presence of noise and other distortions.

The references [171],[172],[176]-[181] present frequency-based approaches that the algorithm presented in this chapter is inspired by. In the previous work of [176]-[178], an equivalent axis representation for 3D rotation is used where one angle describes a rotation around a specified axis also known as angle-axis representation. A critical point is the determination of the rotation axis in

cases where the scan data has interference and occlusions. The method followed in [171] is a robust quaternion based eigenvector decomposition solution using spherical harmonics (SH) to find the global optimal rotation. It works by incorporating a pre-processing stage in which the holes are filled based on a user parameter selection. This process requires an accumulation of all the data points for the volume onto a single unit sphere in order to evaluate the rotation angles using SH. As discussed further in the Section 5.3.1, the fine volumetric or point set structural information is largely lost in the process of the accumulation, hence most of these methods lose accuracy in the presence of noise, distortions, holes/missing data and partial overlaps. All the methods in [176]-[178], [171] use the integration/accumulation of Fourier magnitude data to determine the angle-axis rotation, which is problematic in the presence of holes/ missing data or noise. Only the technique presented in [172] is a similar multi-layered spectral approach, compared to our proposed method, with no accumulation or integration stage of the magnitude data for the 3D rotation estimation. However, it is based on interpolation to achieve resampling of Fourier domain information to spherical coordinates, which is one of the reasons why it can only estimate small rotation angles- detailed differences with our method and its advantages are discussed in the Section 5.3.1.

The principle of calculating a discrete Fourier transform (DFT) on a non-Cartesian grid was extended to 3D in [182] called the pseudo-polar (PP) grid and applied for 3D rotation estimation in [177] and [178]. Similar to this idea of evaluating the DFT on a non-Cartesian grid, which in our case is the spherical polar grid, we propose the use of spherical polar FT (SPFT) [183], which has the unique advantage of computing equiangular data points in the Fourier domain unlike the PP grid, see [183] for details. An analytical exact solution for estimating rotation angles from two equispaced spherical images was proposed in [179]. It uses SH or FFT on a rotation group i.e.  $\text{SO}(3)$  FT (SOFT) which is applicable only for an equiangular grid on a sphere  $\text{S}^2$ . Recently the authors proposed a novel algorithm to compute the discrete SPFT (equiangular) given a 3D volume data, which is fast and algebraically exact. Using SPFT, we can use the spectrum information present

not only in one spherical layer at a radius but on all radii. Given two volumes and the information that one volume is a rotated version of another, multi-layered spectrum computed from SPFTs of the two volumes can be used for correlation of the spherical images with the SOFT algorithm improving the robustness of the rotation estimation, which is precisely the technique that we propose, analyze and demonstrate in this chapter. One of the main contributions of the proposed method when compared to the state of the art method [172] (which has only limited range of estimation  $\pm 30^\circ$ ) is the ability to robustly estimate arbitrarily large rotation angles on all three axes at the same time. This is due to the combined use of SPFT and SOFT techniques. Without the use of SPFT we cannot get equiangular grid points with fast and exact computations to perform SOFT process, unlike the procedure where resampling of 3D (Cartesian) FFT space leads to interpolation and re-projection errors in frequency domain. The details of the core technique can be found in Section 5.3.

The rest of this chapter is organized as follows: Section 5.2 presents the theorem for affine transformation in  $n$ -D space where the 3D transformations can be viewed as the reduced special case, Section 5.3 first shows the robust 3D registration solution for up to 7DOF transformation using SPFT and SOFT techniques for rotation, scale and translation estimation, then we describe the pseudo-code and the computational complexity of the proposed scheme. We briefly discuss the direct extension of the rotation estimation scheme useful for higher dimensions. Section 5.4 contains extensive experimentation using volume/voxel data with hundreds of 3D objects used for registrations, we also compare the performance of the proposed method with three state of the art methods ([171],[172] and [175]), finally we follow by the contributions and conclusions in Section 5.5.

In the spirit of reproducible research, i.e. in order to be able to reproduce all the figures presented in this chapter, we will make our full source code, supplementary materials and the tested datasets openly available to the public for academic purposes here:

<https://github.com/syedalamabbas>

Notation: We use lowercase letters for the time domain functions (continuous / discrete) and uppercase letters for the frequency domain functions.

## 5.2 Preliminaries: Affine Theorem in $n$ -D

The affine transformation theorem for 2D FT case was presented in [167],[168], however it was not generalized to higher dimensions. In the following section, we generalize the theorem to higher dimensions, which shows the relationship between functions that undergo affine transformation in the frequency domain.

### 5.2.1 $n$ -dimensional Fourier Transform

The FT of  $f(x)$  is the function  $\mathbf{F}f(\xi)$  defined by [100],

$$\mathbf{F}(\xi) = \int_{\mathbf{R}^n} e^{-2\pi i \langle \mathbf{x}, \xi \rangle} f(\mathbf{x}) d\mathbf{x} \quad (5.1)$$

The inverse FT of the function  $\mathbf{F}(\xi)$  is,

$$f(\mathbf{x}) = \int_{\mathbf{R}^n} e^{2\pi i \langle \mathbf{x}, \xi \rangle} \mathbf{F}(\xi) d\xi \quad (5.2)$$

where  $\langle x, \xi \rangle$  is the usual inner product (or dot product) in  $\mathbf{R}^n$ .

### 5.2.2 Affine Transformation in $\mathbf{n}$ -D Space

**Theorem:** *Affine Theorem for  $\mathbf{n}$ -D Fourier Transform*

If  $f(x_1, x_2, \dots, x_{\mathbf{n}})$  has a  $\mathbf{n}$ -D FT  $\mathbf{F}(\xi_1, \xi_2, \dots, \xi_{\mathbf{n}})$ , then the affine transformed function of  $f$ ,

$$\begin{aligned} g(x_1, x_2, \dots, x_{\mathbf{n}}) = \\ f(a_{11}x_1 + a_{12}x_2 + \dots + a_{1\mathbf{n}}x_{\mathbf{n}} - t_{x_1}, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2\mathbf{n}}x_{\mathbf{n}} - t_{x_2}, \\ \dots, a_{\mathbf{n}1}x_1 + a_{\mathbf{n}2}x_2 + \dots + a_{\mathbf{n}\mathbf{n}}x_{\mathbf{n}} - t_{x_{\mathbf{n}}}) \end{aligned}$$

has a  $\mathbf{n}$ -D FT given by,

$$\mathbf{G}(\xi) = e^{-2\pi i \langle \xi', t \rangle} \int_{\mathbf{R}^{\mathbf{n}}} e^{-2\pi i \langle \mathbf{x}', \xi' \rangle} f(\mathbf{x}') \frac{d\mathbf{x}'}{|\det(A)|} \quad (5.3)$$

where,

$$\begin{aligned}
A &= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \\
\mathbf{x}' &= (x'_1, x'_2, \cdots, x'_n) \\
\xi' &= (\xi'_1, \xi'_2, \cdots, \xi'_n) \\
t &= (t_{x_1}, t_{x_2}, \cdots, t_{x_n}) \\
x'_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n - t_{x_1}, \\
x'_2 &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n - t_{x_2}, \\
&\vdots \\
x'_n &= a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n - t_{x_n}, \\
\begin{bmatrix} \xi'_1 & \xi'_2 & \cdots & \xi'_n \end{bmatrix} &= A^{-T} \begin{bmatrix} \xi_1 & \xi_2 & \cdots & \xi_n \end{bmatrix}.
\end{aligned}$$

It is easy to derive the above solution.

### 5.3 The Proposed Method of Robust 3D Registration using SPFT and SOFT

In this section, we describe the core technique of robust rotation estimation using SPFT, and correlations of functions on spheres using SOFT for isometry and similarity transformations, the solution for the decoupled translation, and finally we present the full algorithm to perform a 7DOF estimation for 3D registration along with its pseudo code, computational complexity and the extension to higher dimensions.

An affine transformation (or more simply an affinity) is a non-singular linear transformation fol-

lowed by a translation [185], similar to the  $n$ -D case as discussed in the Section 5.2, in 3D using homogeneous coordinates it has the matrix representation,

$$X' = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} X \quad (5.4)$$

where  $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \in \mathbf{R}^3$ , and  $t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$ .

Let  $f_1(X)$  and  $f_2(X)$ ,  $X \in \mathbf{R}^3$ , be scalar signals, associated with two volumes i.e. the luminance/intensity functions, which relates through an *affine transformation*  $H_A = \{A, t\}$ ,

$$f_2(X) = f_1(AX + t) \quad (5.5)$$

Now let us consider the corresponding relationship between the FTs of the two volumes,  $f_1(X)$  and  $f_2(X)$ , using the theorem described in Section 5.2 in the above equation we get,

$$F_2(\xi) = \det(A^{-T}) F_1(A^{-T}\xi) e^{-i2\pi\langle \xi, t \rangle} \quad (5.6)$$

where,  $\xi$  is the Fourier domain Cartesian coordinates.

An advantage offered by the consideration of affine transformation,  $H_A$  in Fourier domain is that the estimation of matrix  $A$  can be decoupled from the estimation of translational vector  $t$ . The separability of affine matrix  $A$  is achieved by using the magnitude equation,

$$|F_2(\xi)| = |\det(A^{-T})| |F_1(A^{-T}\xi)| \quad (5.7)$$



Since we are using just the magnitude first, the constraint  $\det(A) > 0$  immediately follows i.e. if there is an axis reversal it is lost. Observe also that,

$$\det(A^{-T}) \cong \frac{F_2(0, 0, 0)}{F_1(0, 0, 0)} \quad (5.8)$$

i.e. the ratio of the dc/average values of the two volumes.

### 5.3.1 Class I: Isometries

Isometries are transformation of the plane in  $\mathbf{R}^3$  that preserve Euclidean distances [185]. When the affine parameters reduces to just a rotation  $g \in \mathbf{SO}(3)$ , it is an isometry (except reflections) represented by,

$$g = A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (5.9)$$

In  $\mathbf{R}^3$ , any rotation can be expressed as the composition of three rotations about the three axes  $x$ ,  $y$ , and  $z$ . We may also express any element of the rotation group  $g \in \mathbf{SO}(3)$  in terms of the rotations in only about the  $z$  and  $y$  axes using the same conventions as used in [179]. Let,

$$\begin{aligned} R_z(\alpha) &= \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ R_y(\beta) &= \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \end{aligned} \quad (5.10)$$

where  $0 \leq \alpha < 2\pi$  and  $0 \leq \beta \leq \pi$ . Geometrically speaking,  $R_z(\alpha)$  corresponds to rotation by  $\alpha$  radians about the  $z$  axis, and  $R_y(\beta)$  to rotation by  $\beta$  radians about the  $y$  axis. Using these two matrices, any rotation  $g \in \mathbf{SO}(3)$  has an associated *Euler angle decomposition*:

$$g = g(\alpha, \beta, \gamma) = R_z(\alpha)R_y(\beta)R_z(\gamma) \quad (5.11)$$

The Euler angle decomposition provides a natural coordinate system for working with functions on  $\mathbf{SO}(3)$ , where a function  $\Lambda(g)$  for  $g \in \mathbf{SO}(3)$  is given by  $\Lambda(\alpha, \beta, \gamma)$  where  $0 \leq \alpha, \gamma < 2\pi$  and  $0 \leq \beta \leq \pi$ . Thus the affine matrix  $A$  reduces to  $g$ , i.e.  $A = g$  and  $\det(A) = \det(A^{-1}) = \det(A^T) = 1$  in transformation given by (5.6). In practice, for Euclidean transformation the rotation estimation is usually considered the most difficult and error prone. In this section we show a robust way of using the Fourier domain information to accurately determine the 3D rotation.

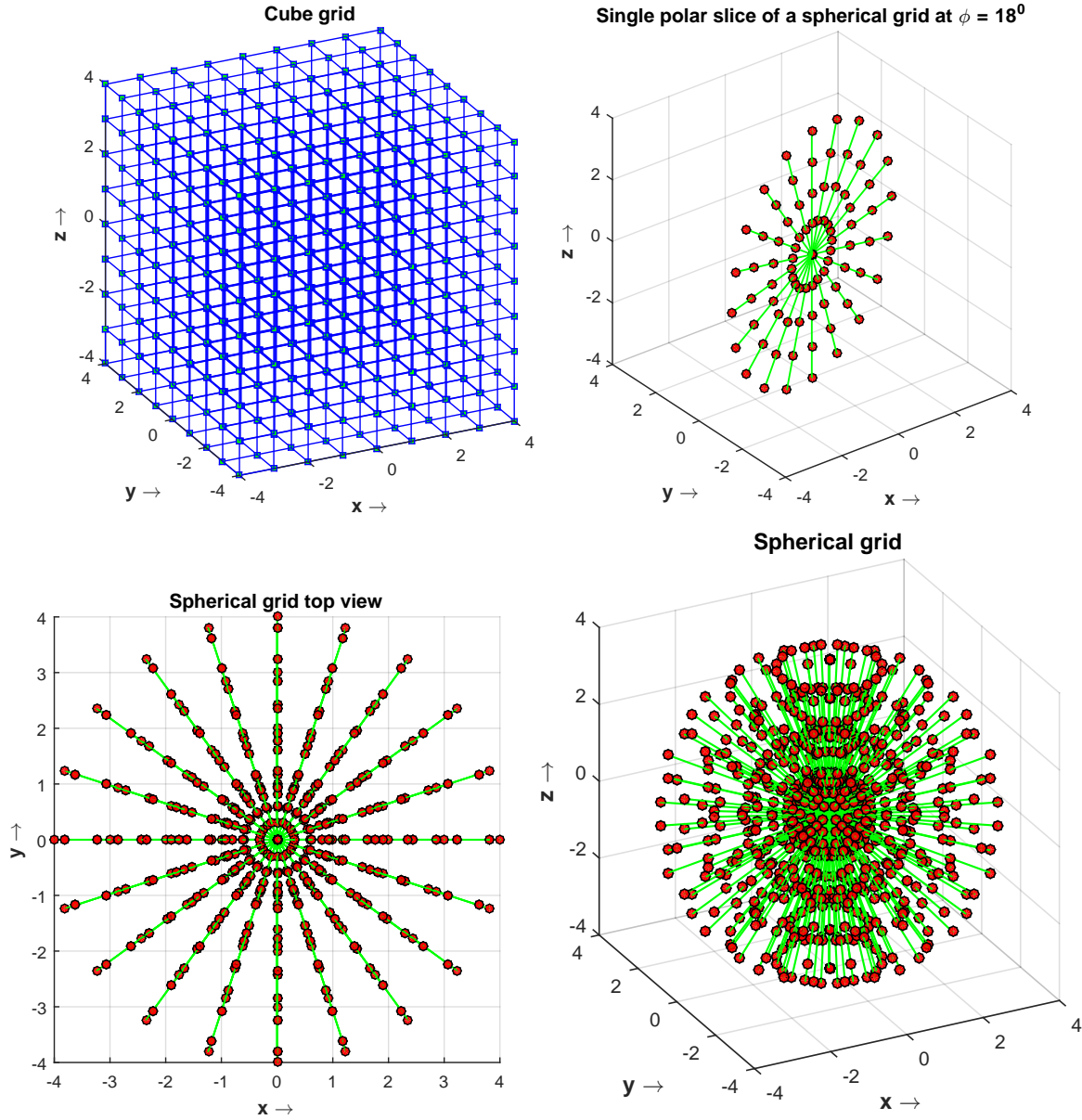


Figure 5.1: Given a 3D volume / matrix  $f(r, c, d)$  we require transformation from spatial domain Cartesian coordinates of size  $(N + 1)^3$ ,  $N$  being even, to the Fourier domain spherical coordinates grid  $F(k, m, n)$ , of size  $K \times M \times (N + 1)$  where the origin is included multiple times. In the above demonstration  $N = 8$  i.e. this is volume of size  $9^3$ , the spherical grid is  $10 \times 10 \times 9$  points.  $K = M = N + 2$ , hence the azimuth and zenith angular spacing  $= 180/K = 18^\circ$ .

Given the Cartesian 3D equispaced volume data  $f(r, c, d)$ , where  $r, c, d$  represent rows, columns and depth indices respectively, we define the *Spherical Polar* or *Spherical* FT (SPFT) to be the collection of samples  $F(k, m, n)$ , given by,

$$F(k, m, n) = \sum_{d=-N/2}^{N/2} \sum_{c=-N/2}^{N/2} \sum_{r=-N/2}^{N/2} f(r, c, d) e^{-i \frac{2\pi n}{N+1} (r \cos \theta_k \cos \phi_m + c \cos \theta_k \sin \phi_m + d \sin \theta_k)} \quad (5.12)$$

where,  $-N/2 \leq n \leq N/2$  represents the points on a radial line,  $\theta_k = k\Delta\theta, 0 \leq k \leq K - 1$  represents the *zenith* angles,  $\Delta\theta = 180^\circ/K$  is the angular spacing in zenith angles,  $\phi_m = m\Delta\phi, 0 \leq m \leq M - 1$  represents the *azimuth* angles,  $\Delta\phi = 180^\circ/M$  is the angular spacing in azimuth angles, and  $K \times M$  is the total number of radial lines. Thus the total number of DFT points is  $K \times M \times (N + 1)$ , where the origin is included multiple times. A fast and exact algorithm with complexity,

$$O\left(3 \times \frac{M}{4} \times \frac{K}{4} \times \frac{(N + 1)^3}{2} \log_2(N + 1)\right) \quad (5.13)$$

to compute (5.12) is proposed in [183], comparatively a brute force approach has a prohibitive complexity of  $O((N + 1)^6)$ , which is essential for the rotation estimation in 3D volume registration discussed below using spherical coordinates. Figure 5.1 illustrates this type of signal transformation.

For the two 3D volumes,  $f_1$  and  $f_2$ , related only by a 3D rotation, we can compute the respective discrete SPFTs using (5.12). This gives us two spherical grids in the Fourier domain,  $F_1$  and  $F_2$ , that are also related by the same 3D rotation. Consider just the spectral spherical magnitude data

of the two volumes,

$$\begin{aligned} H_n^1(k, m) &= |F_1(k, m, n)| \\ H_n^2(k, m) &= |F_2(k, m, n)| \end{aligned} \quad (5.14)$$

where  $1 \leq n \leq N/2$  represents the spectral radii, and  $k, m$  represent the zenith and azimuth angle indices respectively. Let  $L^2(S^2)$  denote the space of square integrable functions on sphere  $S^2$ . The general idea after the above computations is to use these two spherical images  $\{H_n^1(k, m), H_n^2(k, m)\} \in L^2(S^2)$  obtained from the magnitude 3D spectrum for correlation ([179]) in order to determine the 3D rotation angles. The two spherical images that are taken from different radii  $n$  from (5.14) are considered mapped on to a unit spheres. Figure 5.2 depicts the proposed multi-layered spectral approach to the rotation estimation problem. Also observe that for the determination of the rotation we only use the 3D magnitude spectrum (5.14) which is fully decoupled from the overall translation separately computed in Section 5.3.3.

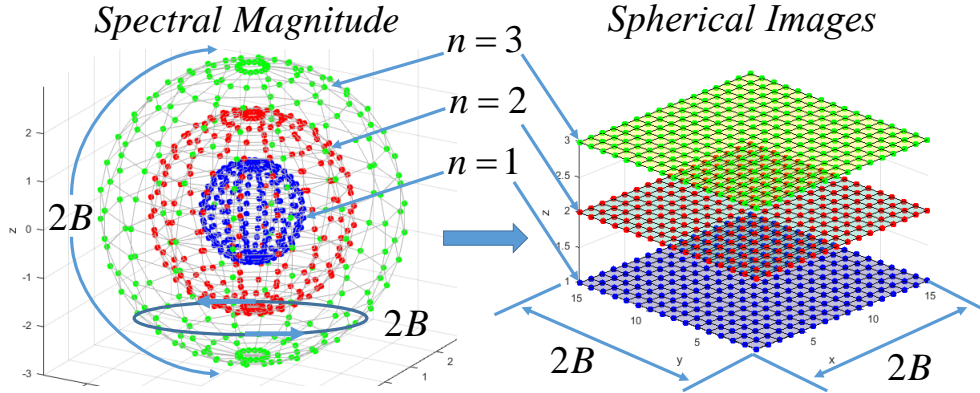


Figure 5.2: Multi-layered approach to the 3D rotation angles estimation. Unwrapping of the spherical magnitude equiangular spectrum data computed from SPFT into  $2B \times 2B$  equispaced spherical images to be used for correlation using SOFT. In the above demonstration,  $B = 8$ ,  $K = 2B$ , and  $M = B$ .

Given the two functions on the spheres,  $H_n^1$  and  $H_n^2$ , at a specific radius, i.e. where  $n$  is a constant integer, and the knowledge that  $H_n^2$  is a rotated version of  $H_n^1$ , i.e.  $H_n^2 = \Lambda(g)H_n^1$  for some  $g \in \text{SO}(\mathbf{3})$ , the goal is to find  $g$ . Here for each  $g \in \text{SO}(\mathbf{3})$ , we can associate a linear operator  $\Lambda(g)$  which acts on  $H_n^1 \in \mathbf{L}^2(\mathbf{S}^2)$ . Since  $H_n^1, H_n^2 \in \mathbf{L}^2(\mathbf{S}^2)$  are two band limited functions on the spheres, we can compute the Fourier expansions of these functions in terms of the SH functions  $Y$ ,

$$\begin{aligned} H_n^2(\omega) &= \sum_p \sum_{|q| \leq p} \mathbf{A}_{pq} Y_p^q(\omega) \\ H_n^1(\omega) &= \sum_p \sum_{|q| \leq p} \mathbf{B}_{pq} Y_p^q(\omega) \end{aligned} \quad (5.15)$$

The terms  $\mathbf{A}_{pq}$  and  $\mathbf{B}_{pq}$  represents the SH coefficients from the corresponding spherical data  $H_n^2$  and  $H_n^1$  respectively. Evaluating  $g$  or determining the maximum  $g$  (for correlation) by means of FFT on  $\text{SO}(\mathbf{3})$  can be achieved by correlating the two spherical functions as follows,

$$\mathbf{C}(g) = \int_{\mathbf{S}^2} H_n^2(\omega) \overline{\Lambda(g)H_n^1(\omega)} d\omega. \quad (5.16)$$

where  $\overline{\Lambda(g)H_n^1(\omega)}$  denotes the complex conjugate of the term  $\Lambda(g)H_n^1(\omega)$ .

Using the functions  $H_n^1$  and  $H_n^2$  in terms of SH computed from (5.15) in (5.16) and expressing the spherical harmonic of degree  $p$  in terms of *Wigner-D function* (used to express the linear combinations of the SH of the same degree), simplifying gives the result,

$$\mathbf{C}(g) = \sum_p \sum_{|q| \leq p} \sum_{|q'| \leq p} \mathbf{A}_{p-q} \overline{\mathbf{B}_{p-q'}} (-1)^{q-q'} \mathbf{D}_{qq'}^p(g). \quad (5.17)$$

The  $\mathbf{D}_{qq'}^p$  is a Wigner-D function based on Jacobi polynomials. The correlation can then be efficiently evaluated in terms of Wigner-D functions and using the inverse  $\text{SO}(\mathbf{3})$  FT. A fast and exact

algorithm using FFT can be used for correlating two equiangular spherical images, the details of computing (5.17) using SH is given in [179]. The original data mapped on the sphere is given by the bandwidth  $B$  which defines the angular spacing of the points. By choosing a square planar image of size  $2B \times 2B$  interpreted as data living on the equiangular  $2B \times 2B$  grid on the sphere as shown in the Figure 5.2, we can efficiently evaluate the correlation  $C(g)$  of  $H_n^1$  and  $H_n^2$ , at a whole slew of  $g$ 's.

Using the SPFT of the two volumes,  $F_1$  and  $F_2$ , and by choosing  $K = 2B$  and  $M = B$  in the computation of (5.12) we get an equiangular gridded spectral data on the spheres,  $H_n^1$  and  $H_n^2$ , of size  $2B \times 2B$  at different radii  $n$  to do the correlation of spherical images using (5.16). The result of the correlation is the estimate for the angles  $\alpha_n$ ,  $\beta_n$  and  $\gamma_n$  at a specific spectral radius  $n$ . An estimate from a single radius is usually sufficient in the case of full overlapping volumes as shown in our simulations, see Sections 5.4.1 and 5.4.4. However, it is known that the spectral signature at some unknown radius will become unreliable due to distortions such as missing data, noise, and partial overlap between the two volumes, see [172]. Since the spectral magnitude data is available at all radii, i.e.  $1 \leq n \leq N/2$ , we can compute the estimates of rotation  $\alpha$ ,  $\beta$  and  $\gamma$  that are supposed to be same for all radii. This built-in redundancy gives us the robustness in the parameter estimation of the rotation matrix using the two volumes as confirmed by the simulations, see Section 5.4.4.

The original spherical correlation based rotation estimation [179], and the recent development using SH [171] for 3D global rotation solution, work only for 1 layer, i.e. the points in the volume are mapped onto a sphere, and then the SOFT/SH procedure is computed for the whole 3D object just once. In 3D and for the higher dimensions in general, much of the shape information is spread across multiple layers and integrating/accumulating or remapping the whole volume only onto one sphere destroys much of the fine structural contents, which in turn leads to poor and erroneous rotational estimates especially in the presence of high noise. Reference [172] has a multilayer

spectral approach similar to the principle of our method, however there are the following major differences:

- It involves spectral resampling i.e. mapping of 3D Cartesian Fourier magnitude data only onto a set of concentric hemispheres, which is based on interpolations in the Frequency domain. The resampling process is a projection of the spherical layer, which leads inevitably to deviations from the true rotation. In the proposed method, we avoid this resampling process to compute the full 3D SPFT directly from the given volume using a fast and exact algorithm that we presented in [183], hence we are able to capture very accurately the true 3D rotation.
- It requires first the estimation of yaw angle separately with resampling of Fourier data from Cartesian to spherical coordinates, the evaluation of which is crucial for the subsequent full 3D rotation angles computation to work. Our technique is based on the SOFT method of correlation of the spherical images and hence we get all three angles in one step.
- It uses for roll/pitch rotations a perpendicular projection of the surface of the hemispheres to transform the problem into a 2D shift in matrix form of the projected structures. In our method, the whole 3D rotation angle estimation is done using 3D correlation without any further resampling/projections.
- Due to the above roll/pitch projection based solution, an assumption which holds true only for a restricted range, it is constrained to estimate only small angles i.e. the method works as long as roll and pitch both change by at most  $\pm 35^\circ$  each at the same time, where as for our technique the arbitrarily large 3D rotation angles  $\alpha_n, \beta_n, \gamma_n$  at a specific radius  $n$  can be estimated without any loss of accuracy. The accuracy only depends on the chosen bandwidth or equiangular grid resolution  $B$  which in turn also defines the angular resolution of SPFT.



However, due to all such simplifying assumptions, such as the restricted range, use of only 3D FFT, and the use of phase matching, the SRMR technique in [172] is faster than our method, due to lower complexity.

### 5.3.2 Class II: Similarity transformation

In the literature for 3D registration, mostly the estimation of 6DOF is discussed (see [171],[172], [175]), because for the 3D transformations, the 3DOF for rotation and 3DOF for translation are often considered as being sufficient. However, in this section we show that the global scale factor can be directly and efficiently estimated without any significant additional computations.

A similarity transformation (or more simply a similarity) is an isometry composed of a Euclidean transformation and an isotropic scaling. Let  $s > 0$  be the scalar representing the isotropic scaling between the two volumes, then the determinant in (5.7) becomes,  $\det(A) = s^3$ , and the scaling parameter can be directly obtained using (5.8), i.e.

$$s \approx \sqrt[3]{\frac{F_2(0, 0, 0)}{F_1(0, 0, 0)}} \quad (5.18)$$

The average or dc values  $F_2(0, 0, 0)$  and  $F_1(0, 0, 0)$  can be directly computed from the given volumes,  $f_1$  and  $f_2$ , as follows,

$$F_l(0, 0, 0) = \sum_{d=-N/2}^{N/2} \sum_{c=-N/2}^{N/2} \sum_{r=-N/2}^{N/2} f_l(r, c, d), l = 1, 2. \quad (5.19)$$

The scaling can be applied to one of the two 3D volumes, and with the new scale compensated volumes the similarity transformation reduces to a simple Euclidean isometry, where the solution

can be obtained as described in Section 5.3.1.

### 5.3.3 Estimating The Translational Vector, $t$

Consider two volumes,  $f_1$  and  $f_2$ , that differ only by a translational vector  $t = (t_x, t_y, t_z)$  i.e.

$$f_2(x, y, z) = f_1(x - t_x, y - t_y, z - t_z). \quad (5.20)$$

For the spatial domain relation in (5.20), the corresponding FTs will be related according to the *Fourier shift* theorem as follows,

$$F_2(u, v, w) = e^{-j2\pi(ut_x + vt_y + wt_z)} \times F_1(u, v, w) \quad (5.21)$$

The normalized cross power spectrum of the two images is given by,

$$Q(u, v, w) = \frac{F_2(u, v, w) \overline{F_1(u, v, w)}}{|F_2(u, v, w) \overline{F_1(u, v, w)}|} = e^{-j2\pi(ut_x + vt_y + wt_z)} \quad (5.22)$$

where  $\overline{F_1(u, v, w)}$  stands for complex conjugate. The Fourier shift theorem guarantees that the phase of the cross-power spectrum is equivalent to the phase difference between the images. It can also be seen from (5.6) that the translation information is only encoded in the phase even when the volumes are rotated and scaled. Taking the inverse Fourier transform of (5.22) we get,

$$q(x, y, z) = \mathbf{F}^{-1}\{Q(u, v, w)\} \quad (5.23)$$

Then the translation can simply be computed by,

$$(t_x, t_y, t_z) = \underset{x, y, z}{\operatorname{argmax}} q(x, y, z). \quad (5.24)$$

The above solution (5.24) is called POMF [167],[170], and it is directly applicable to the formulation in higher dimensions (5.3). There are a variety of methods to improve this phase correlation to sub-pixel/sub-voxel accuracy, such as [186], [187], [188], [189].

#### 5.3.4 Robust 3D Registration - The Algorithm

In this section we present the complete algorithm for robust 3D registration with up to 7DOF transformation, given two volumes. We use a simple metric for the quality of the estimated rotation angles, the root mean squared error (RMSE) of the two volumes after alignment. This measure is used to select the best rotational parameters from the redundant set of estimates of  $\alpha, \beta, \gamma$  computed at various spectral radii. The RMSE for the two volumes,  $f_1$  and  $f_2$  rotated by a matrix  $g(\alpha, \beta, \gamma)$  is defined using the  $L_2$  norm as follows,

$$\text{RMSE} = \frac{1}{(N+1)^3} \|gf_1 - f_2\|_2^2 \quad (5.25)$$

The following is the full pseudo code for our proposed 3D robust transformation estimation of 7DOF :

**Input:** Two volumes,  $f_1, f_2$ , of size  $(N+1)^3$ , where  $N$  is even. For SOFT procedure, the equiangular bandwidth  $B$ , where  $B$  is even, and the degree  $D$  of spherical harmonics to be used internally in the SOFT program. For SPFT algorithm, the angular resolution depends on  $B$  i.e.  $K = 2B, M = B$  for the zenith and azimuth angles respectively, where  $K, M$  are also even.

**Output:** Global scale factor,  $s$ . Rotational vector,  $(\alpha, \beta, \gamma)$ . Finally, the translational vector,  $(t_x, t_y, t_z)$ .

**Step 1:** Compute for the two volumes,  $f_1$  and  $f_2$ , the average dc values using (5.19) to obtain  $F_1(0,0,0)$  and  $F_2(0,0,0)$ .

**Step 2:** Compute the scaling  $s$  according to the solution in (5.18). This is the global scale factor,  $s$ .

**Step 3:** Compensate for scale  $s$  by scaling the first volume,  $f_1$ .

**Step 4:** Compute SPFT for the two new scale compensated volume pairs,  $f_1$  and  $f_2$ , to obtain the transformed domain spherical structures,  $F_1$  and  $F_2$ , according to the definition (5.12).

**Step 5:** For the spectral radius  $n$ , where  $1 \leq n \leq N/2$ , do the following.

**Step 6:** Gather the two spectral magnitude data,  $H_n^1$  and  $H_n^2$  as defined in (5.14).

**Step 7:** Compute the SOFT based correlation of the spherical images  $H_n^1$  and  $H_n^2$  to evaluate the corresponding,  $g_n(\alpha_n, \beta_n, \gamma_n)$ .

**Step 8:** Compute the RMSE according to the given definition in (5.25) for this spectral radius  $n$ .

**Step 9:** Gather the computed estimate at the spectral radius  $n$  of Euler angles  $\alpha_n, \beta_n, \gamma_n$  and  $\text{RMSE}_n$  in a matrix of size  $(N/2) \times 4$ .

**Step 10:** If  $n > N/2 + 1$ , terminate. Else  $n = n + 1$ , go to **Step 6**.

**Step 11:** Select minimum error  $\text{RMSE}_n$  and the corresponding angles at

$n$ , according to

$$(\alpha^*, \beta^*, \gamma^*) = \underset{\text{RMSE}_n}{\text{argmin}} (\alpha_n, \beta_n, \gamma_n, \text{RMSE}_n)$$

from the matrix in the **Step 9**. This gives us the final rotational vector.

**Step 12:** Compensate the rotation for the two volumes by rotating the first volume  $f_1$  by the transformation  $g$ .

**Step 13:** Compute for the two volumes,  $f_1$  and  $f_2$ , 3D FFT to obtain the transformed domain

Cartesian coordinates structures,  $\hat{F}_1$  and  $\hat{F}_2$ .

**Step 14:** Calculate the values for the translation vector,  $(t_x, t_y, t_z)$  using (5.24).

The above step gives us the final translation vector completing the full 7DOF transformation estimation.

#### 5.3.4.1 Computational Complexity

One of the requirements in the procedure described is the use of SPFT algorithm, where according to (5.13) (details discussed in [183]), the complexity is,

$$O\left(3 \times \frac{M}{4} \times \frac{K}{4} \times \frac{(N+1)^3}{2} \log_2(N+1)\right).$$

This is required for the two volumes. The efficient algorithm implementation for the numerical computation of FTs of the band-limited functions defined on a rotation group  $SO(3)$  ([179]), is presented in [184]. For the pair of spherical images  $H_n^1$  and  $H_n^2$  with bandwidth  $B$ , the computational complexity is  $O(B^4)$ , where the spherical harmonics is of degree at most  $B$ . There are  $N/2$  such spectral spherical magnitude data pairs, we thus have the total complexity of using the SOFT technique as,

$$O\left(2 \times \frac{N}{2} \times B^4\right).$$

There are two 3D FFTs also involved with the complexity  $2 \times (N + 1)^3 \log_2(N + 1)$ . Since we choose  $K = 2B$  and  $M = B$ , the computational complexity of SPFT algorithm dominates the entire 3D robust estimation procedure. Hence, the complexity of the entire registration algorithm is,

$$O\left(6 \times \frac{M}{4} \times \frac{K}{4} \times \frac{(N + 1)^3}{2} \log_2(N + 1)\right)$$

Due to sufficient computational complexity the proposed solution is more suitable for an off-line rotational estimation application such as sensitive medical data, or a challenging dataset where there is significant amount of distortion such as holes, partial overlaps, noise, etc., where local techniques like ICP are not easily applicable or reliable. We demonstrate the robust nature of the proposed solution in our simulations, see Section 5.4.5. However, as it can be seen the proposed method is globally robust, non-iterative (fixed complexity) and the fully parallel algorithms of SPFT and SOFT can take full advantage of commodity GPGPU processing hardware. Most importantly each of the components used in our solution has a parallelizable structure and scales naturally with processing and memory resources [183],[179]. Also, to reduce the above complexity, we can consider only a few of the spectral spherical layers for the SPFT and SOFT computations rather than many concentric spheres of spectral information. In Section 5.4, for all experiments, we have used only up to  $n = 11$  layers for robustly estimating the rotation angles, regardless of the volumetric resolution.

#### 5.3.4.2 Higher Dimensions

The solution proposed for estimation of similarity transformation can be directly applied to higher dimensions  $\mathbf{R}^n$ , i.e when  $n > 3$ . For estimation of similarity of order  $n > 3$ , we can use the transformation to  $n$ -dimensional *hyper-spherical* coordinate system using the relationship between the two *tensors* presented in the affine theorem of section 5.2. As shown in Section 5.3.1 the high dimensional rotational parameters can be computed using a fast and exact solution of *hyper-spherical* transform (similar to the one proposed in [183]) and the multilayer redundant correlations (similar to the one proposed in [179] for higher dimensions) on the *hyper-sphere*. The translation can then be simply estimated using phase correlation as discussed in Section 5.3.3.

### 5.4 Experimental Validation and Performance Evaluation

For comparison, we focus on the following three state-of-the-art methods,

- Robust Estimation using Spherical Harmonics,[171].
- Spectral Registration with Multilayer Resampling, [172].
- Go-ICP, Global Optimal Iterative Closest Point,[174], [175].

The first two methods bear some similarity to our proposed solution : they are the state-of-the-art robust methods well designed for 3D rotation estimation using Fourier domain information. The first method uses SH, and is suitable only for surface 3D data even when it has holes or missing data, while the second solution uses POMF, and can handle volume data with partial overlap, both of these techniques have some limitations when compared to our proposed method, which as discussed in the Section 5.3.1, and demonstrated and confirmed by the simulations here. The last

method is also a recent development, and provides a global solution but follows a totally different working principle than the first two methods. The Go-ICP method proposes a globally optimal solution to the 3D 6DOF registration problem defined by ICP [173]. We compare the results of the proposed method over the Go-ICP solution in case of a partial overlap and noisy registration data. We have mainly focused on demonstrating the error prone and most difficult part for any 3D transformations which is the 3D rotation estimation because the computations of scaling using (5.18) and translation vector using (5.24) are equally robust and always work.

#### 5.4.1 3D Rotation Estimation - Simple MRI Dataset

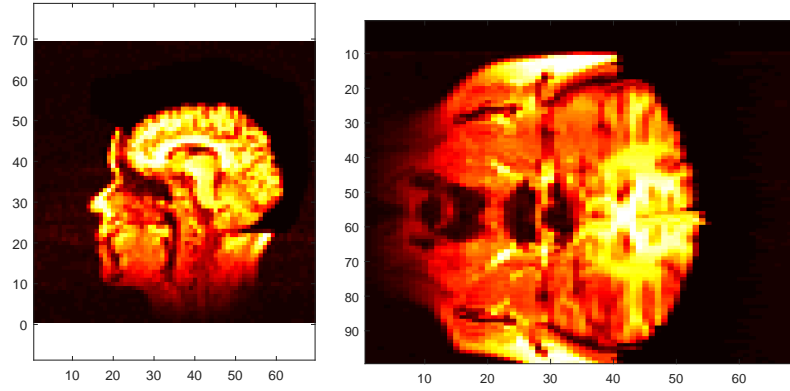


Figure 5.3: Slices of the MRI images used to construct a full 3D volume, XY and YZ view of the volume, using the publicly available dataset [190].



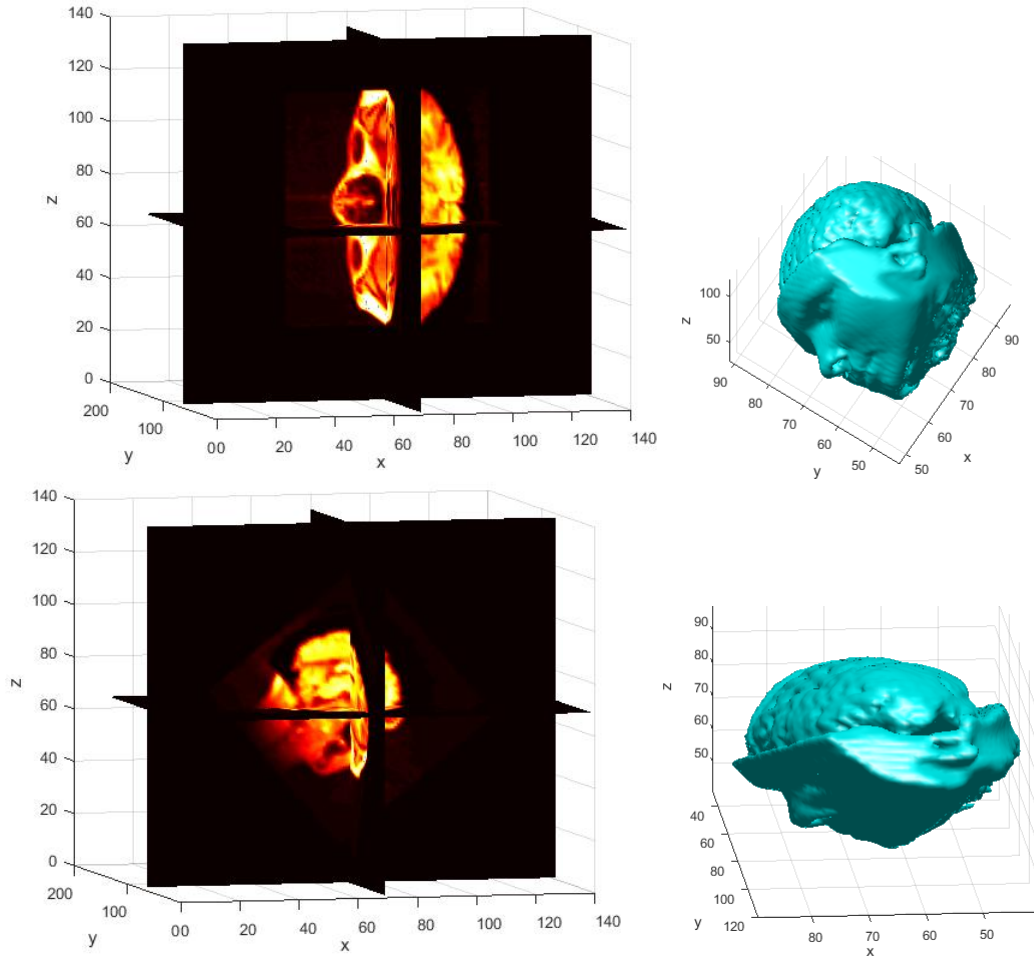


Figure 5.4: Slice projections and surface visualizations of the 3D volumes used for comparison with actual rotation matrix in (5.11) given by Euler angles,  $(\alpha, \beta, \gamma) = (0.7069, 1.0996, 1.5080)$ . Voxel size for both objects is chosen arbitrarily as  $63 \times 63 \times 63$ , and the equiangular bandwidth  $B = 68$  as discussed in the Section 5.3.1.

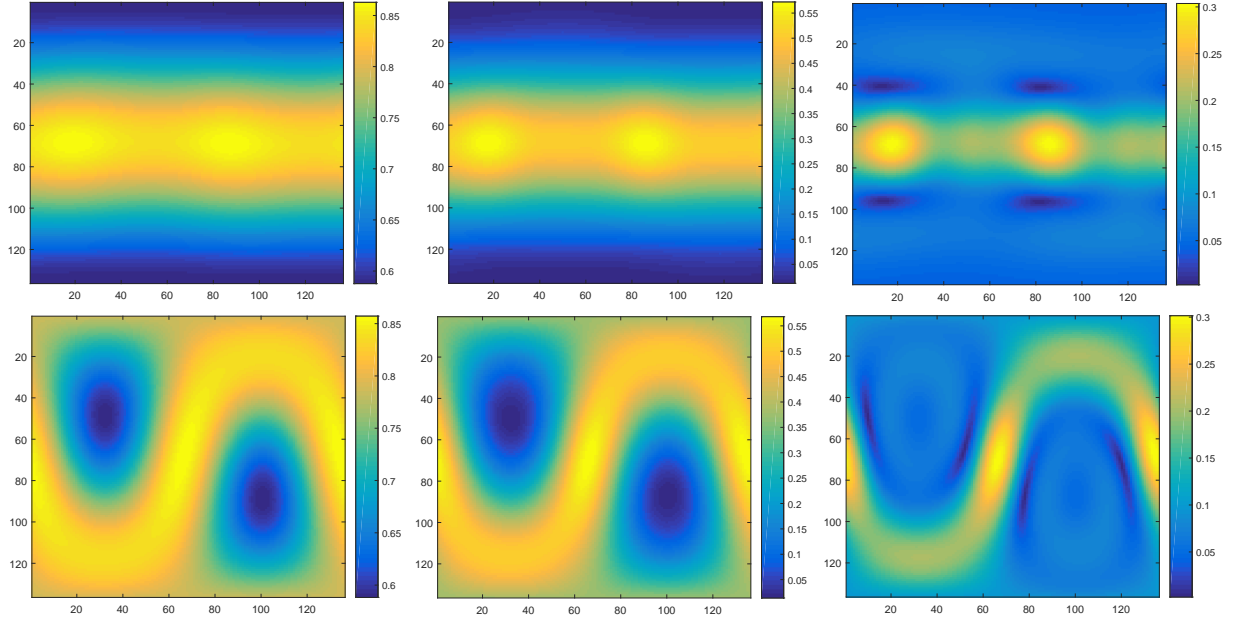


Figure 5.5: Spherical images  $H_n^1$  and  $H_n^2$  as described in (5.14), original (top row) and rotated (bottom row) by  $(\alpha, \beta, \gamma) = (0.7069, 1.0996, 1.5080)$ , at spectral radii (left to right)  $n = 1, 2$  and  $3$  respectively, obtained from (5.12) (SPFT) and correlation given by (5.16) (SOFT). Note the variations in the spectral signature as we go from smaller to larger radius which is equivalent to going from low to high frequencies.

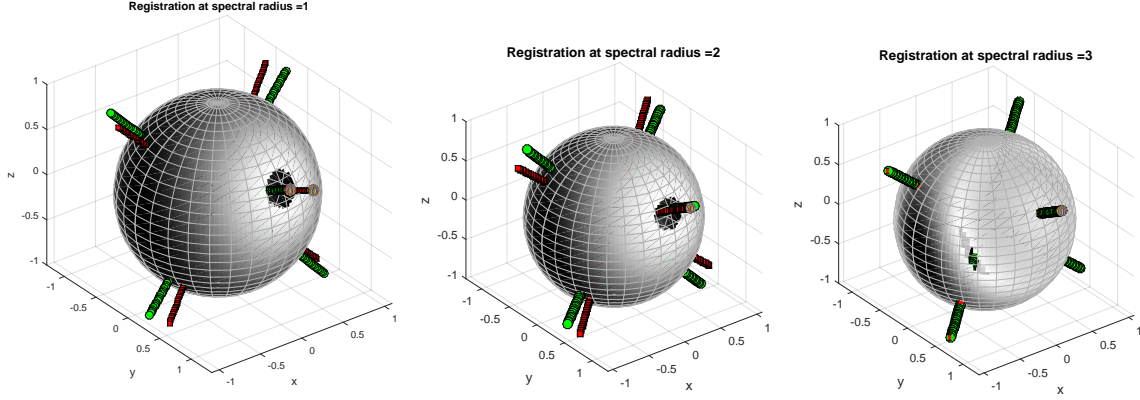


Figure 5.6: Evolution of registration at different spectral radii, the estimated parameters at  $n = 1$  are (5.6364, 2.0443, 4.8510). Although the solution as observed above is close with  $\pi/2\pi$  ambiguity, it is not a perfect match within the angular resolution. The estimated parameters at  $n = 2$  are (2.4024, 1.1972, 1.7632) is closer to the ideal one. Finally, at radius  $n = 3$  the estimated values are (2.4024, 1.0972, 1.6632), which is near perfect registration within the angular resolution defined by bandwidth  $B$  with  $\pi$  ambiguity for angle  $\alpha$  i.e  $\pi - \alpha = 0.7392$  which is  $\cong$  to the actual orientation  $\alpha = 0.7069$  radians.

We begin with a simple rotation estimation task using the MRI volume with skull removed from the publicly available data set [190]. Figures 5.3 and 5.4 show the reference/fixed volume and the rotated volume, which are visualized through images, slice projections at orthogonal angles and through the surface data. Figure 5.5 shows the images obtained using 3D SPFT [183] at radii  $n = 1, 2$  and 3, respectively. Notice the slight variations in the spectral signatures for the increasing radii, the sharpness of which will depend upon the volumetric information, especially in the presence of holes as demonstrated later. Using these images we compute the correlations on spheres with the SOFT method [179]. Figure 5.6 shows the final 3D rotation estimation results using spherical magnitude data obtained using the two volumes.

The black and white shading on the spheres reflects the wrapped intensities of the spectral spherical images. We also use special marker 3D axes to conveniently visualize the 3D rotational transformation and to visually assist in demonstrating the failed or successful estimation. The axes marked

by the stack of green circles is the known actual rotation and the axes marked by the the stack of red squares is the estimated rotation. When the two marker axes (green and red) overlap fully within the angular resolution, it signals that the proposed rotation estimation is successful, any misalignment can also be visualized well using this.

Observe that the estimates at radii  $n = 1$  and  $2$  are close to the actual rotation, while the solution at radius  $n = 3$  is a perfect match, up to the angular resolution of the equiangular grid defined by  $B$ . While both the techniques [171] and [172] perform well with full overlapping 3D models, our solution involves no approximation or interpolation steps, exploits the redundant Fourier domain information, uses only full 3D correlation, and hence it is more robust to noise/holes/partial overlaps as confirmed by the simulations in the following sections. Also note that its accuracy can be increased to an arbitrary precision by increasing the number of angles defined by the bandwidth  $B$ .

#### 5.4.2 *Princeton Object Database, Large Set*

In order to evaluate the accuracy and robustness of the proposed method for 3D rotation estimation of different 3D objects with varying geometric complexities, we conducted hundreds of experiments using the Princeton Shape Benchmark (PSB) database [191]. PSB is a large database for 3D models collected from different web domains, we used a total of 786 objects for our experimentation. These objects are composed of vertices and faces, which are then voxelized or transformed into volumes using the 3D software package [192] for the use of our method, see Figure 5.7 for few sample objects in their original form.

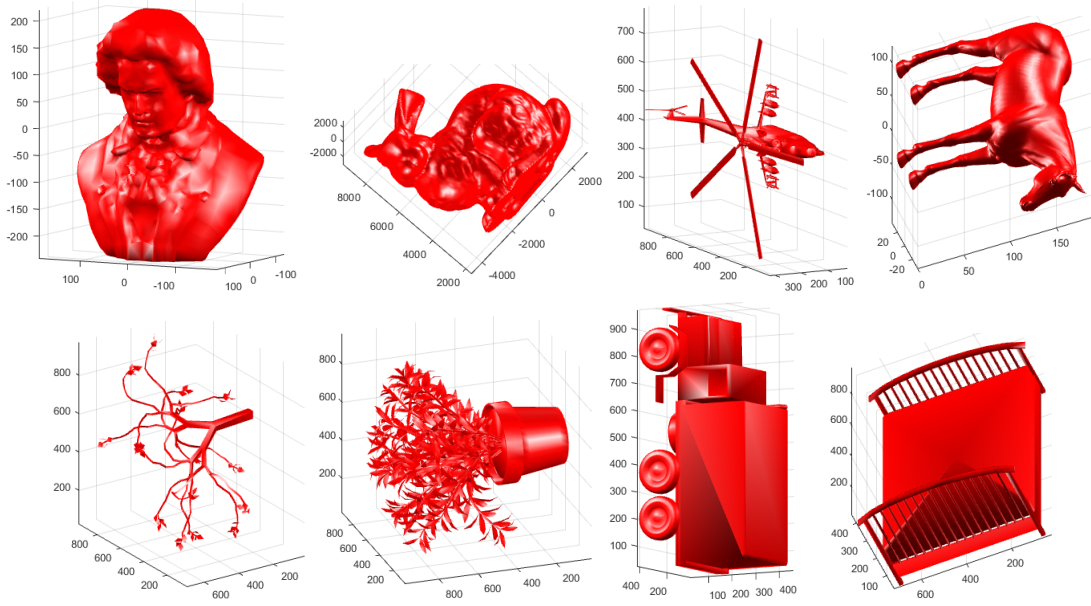


Figure 5.7: Different objects with shapes and geometric complexities used in the PSB database test [191], (Clockwise from top left) (a) Beethoven (b) Bunny (c) Helicopter (d) Horse (e) Bed (f) Truck (g) Pot Plant (g) Branches.

We compute the RMSE value between the volumes of the rotated object and reference/fixed object in the spatial domain after the rotation estimation, using (5.25) which gives us a crude global measure for the quality of estimated rotation angles in the case of fully overlapping volumes. We also compute the absolute error for each of the Euler angles  $\alpha, \beta, \gamma$ . Figure 5.8 shows the plot for all the objects used for this experiment and the error estimates.

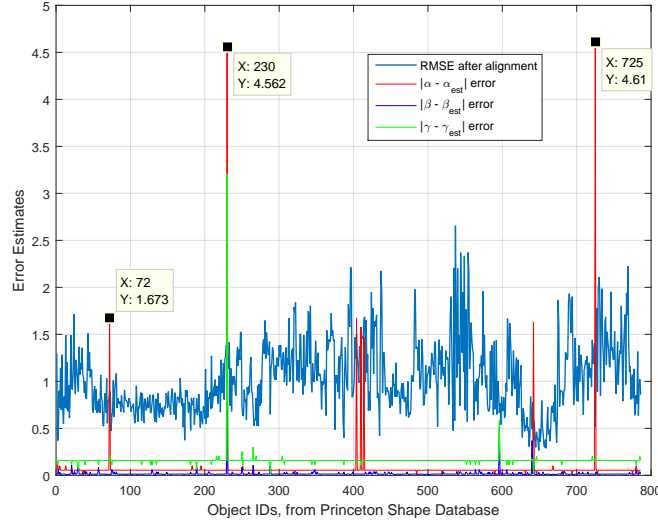


Figure 5.8: The error estimates obtained for the PSB objects that are 786 in total, the known fixed 3D Euler angles are  $(\alpha, \beta, \gamma) = (2.2777, 1.9018, 1.5080)$  and are used for rotation according to (5.11). The volumetric resolution for all the objects after voxelization is  $63 \times 63 \times 63$ .

The special objects where the angular errors are large, even though the RMSE are not as abnormal, are investigated further. The registration of these few objects is shown explicitly in the Figure 5.9. The problematic estimations occur for objects that are elongated, largely flat or rotationally symmetric about any axis. However, as it can be seen from the Figure 5.9, the estimations are largely accurate, the errors are due to the  $\pi/2\pi$  ambiguity or the rotationally symmetric problem, demonstrating that the proposed solution is effective and independent of the shape of the object to be registered.

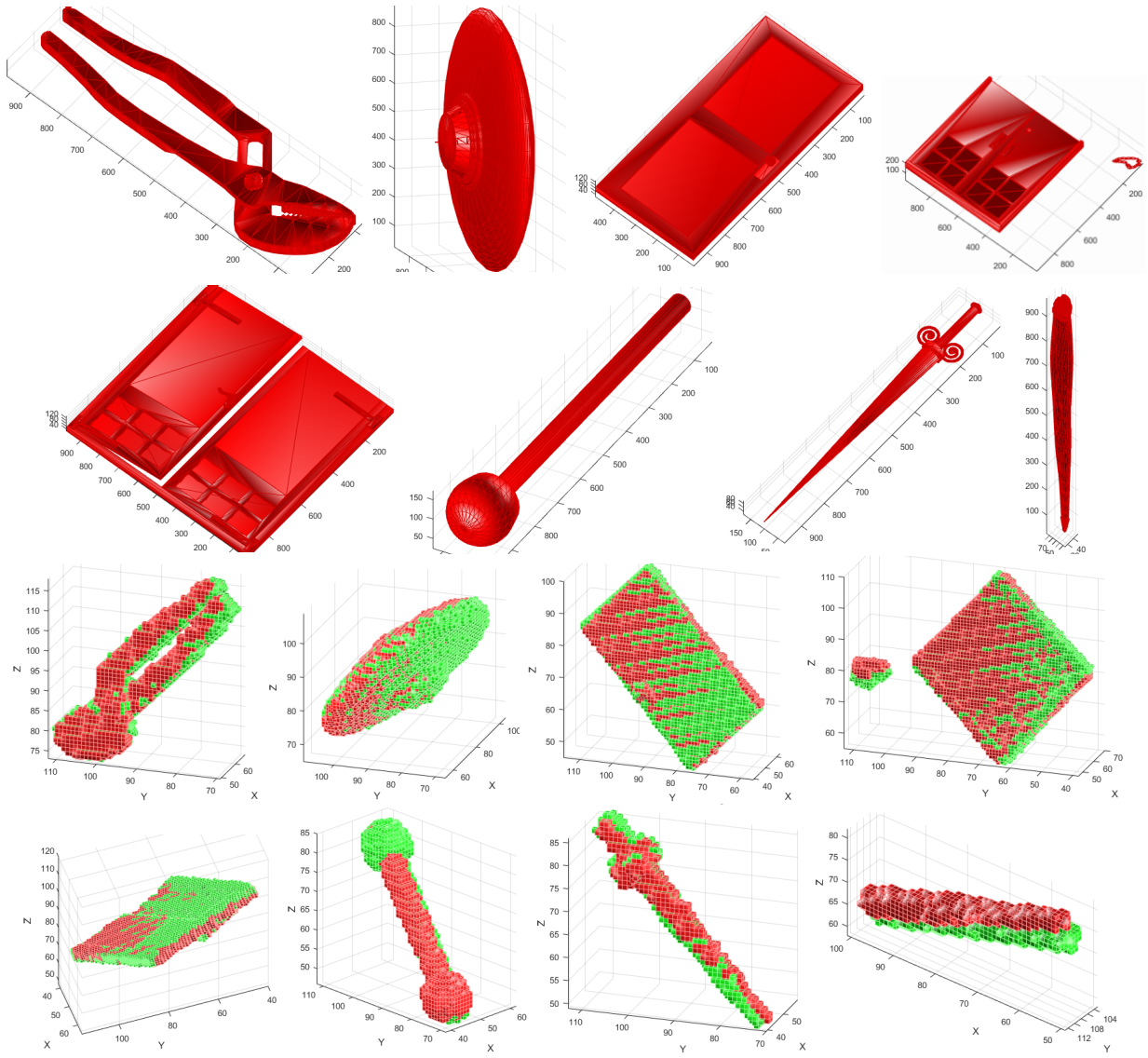


Figure 5.9: All the anomalies of the PSB test with the object IDs in  $\{72, 230, 404, 410, 414, 596, 642, 725\}$ . (Top two rows) Original objects with large angular errors after registration. (Bottom two rows) Voxelized fixed and registered 3D objects (red and green respectively) after the registration using our solution. Note the full  $\pi$  ambiguity for certain objects after registration and the rotationally symmetric shape of other objects.

### 5.4.3 Robustness to Noisy Surface Points

We evaluated the robustness of our method with additive noise in the data. The  $x, y$  and  $z$  coordinates of the vertices of the original object are perturbed by an additive white Gaussian noise, where SNR varies from 10 to 40 dB. We conducted several experiments using 20 different objects randomly chosen from the PSB database, see Figure 5.10 for the demonstration of *Dog* object and the following table where the average RMSE of the angles is shown.

We observe that the proposed solution is very robust to additive noise even when  $\text{SNR} = 10\text{dB}$  beyond which the shape of the objects becomes sufficiently distorted to not merit any reasonable comparison. The very crude spectral signatures are largely sufficient for effectively computing the rotation estimation using our proposed method. Thus, our technique can also be used with objects that have intra-class variations (e.g. slight changes in car models) as studied in [171].



Table 5.1: Table I: Effects of perturbation on the surface vertices.

SNR(dB)	$ \alpha _{\text{error}}$	$ \beta _{\text{error}}$	$ \gamma _{\text{error}}$	RMSE
10	0.1861	0.0893	0.2547	0.1893
15	0.0858	0.0558	0.1210	0.0915
20	0.0144	0.0057	0.0542	0.0325
30	0.0479	0.0057	0.0207	0.0303
35	0.0190	0.0057	0.0207	0.0165
40	0.0144	0.0057	0.0542	0.0325

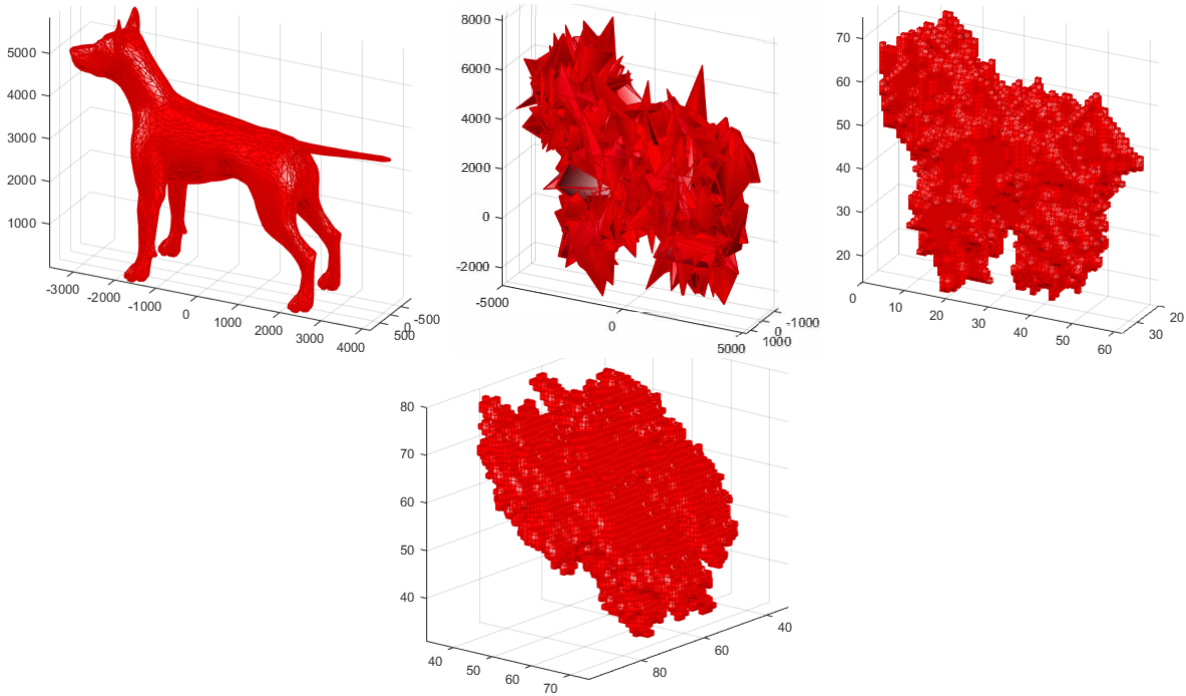


Figure 5.10: Surface data transformation for rotation estimation, using *Dog* object from PSB database [191]. Voxel size for both objects is chosen as  $89 \times 89 \times 89$ , and the equiangular bandwidth  $B = 94$ . SNR = 10dB (measured). Actual angles used for rotation are,  $(1.0505, 0.7159, 0.7226)$ . Estimated angles at spectral radius,  $n = 2$  are  $(\pi - 1.9050, 0.6266, \pi - 2.6737)$ .

#### 5.4.4 Missing Voxels or Holes or Partial Data

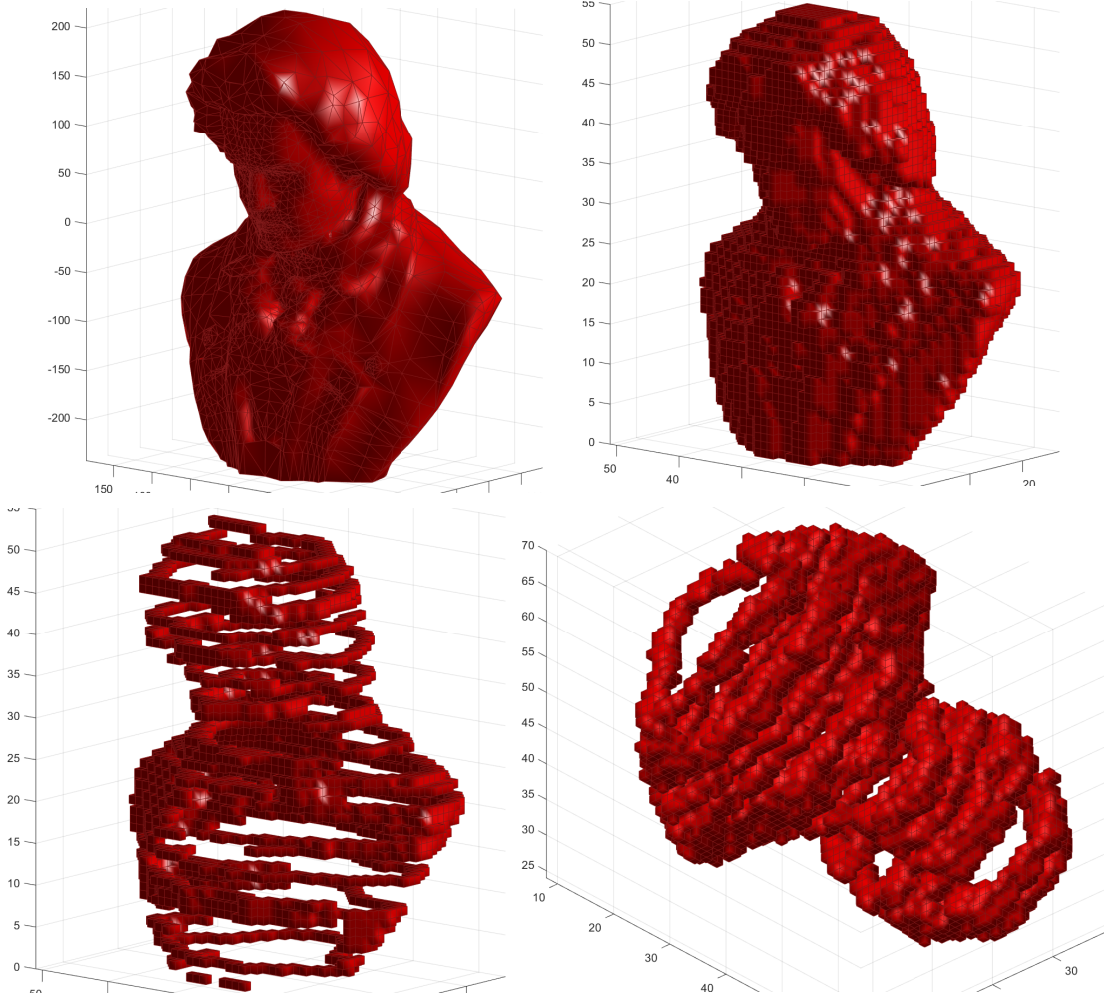


Figure 5.11: Surface data transformation for rotation estimation, using *Beethoven* bust from PSB database [191]. Voxel size for both objects is chosen as  $75 \times 75 \times 75$ , and the equiangular bandwidth  $B = 80$ . The surface data is first voxelized then significant distortion is introduced at all levels in the form of missing surface data by setting certain values to zeros.

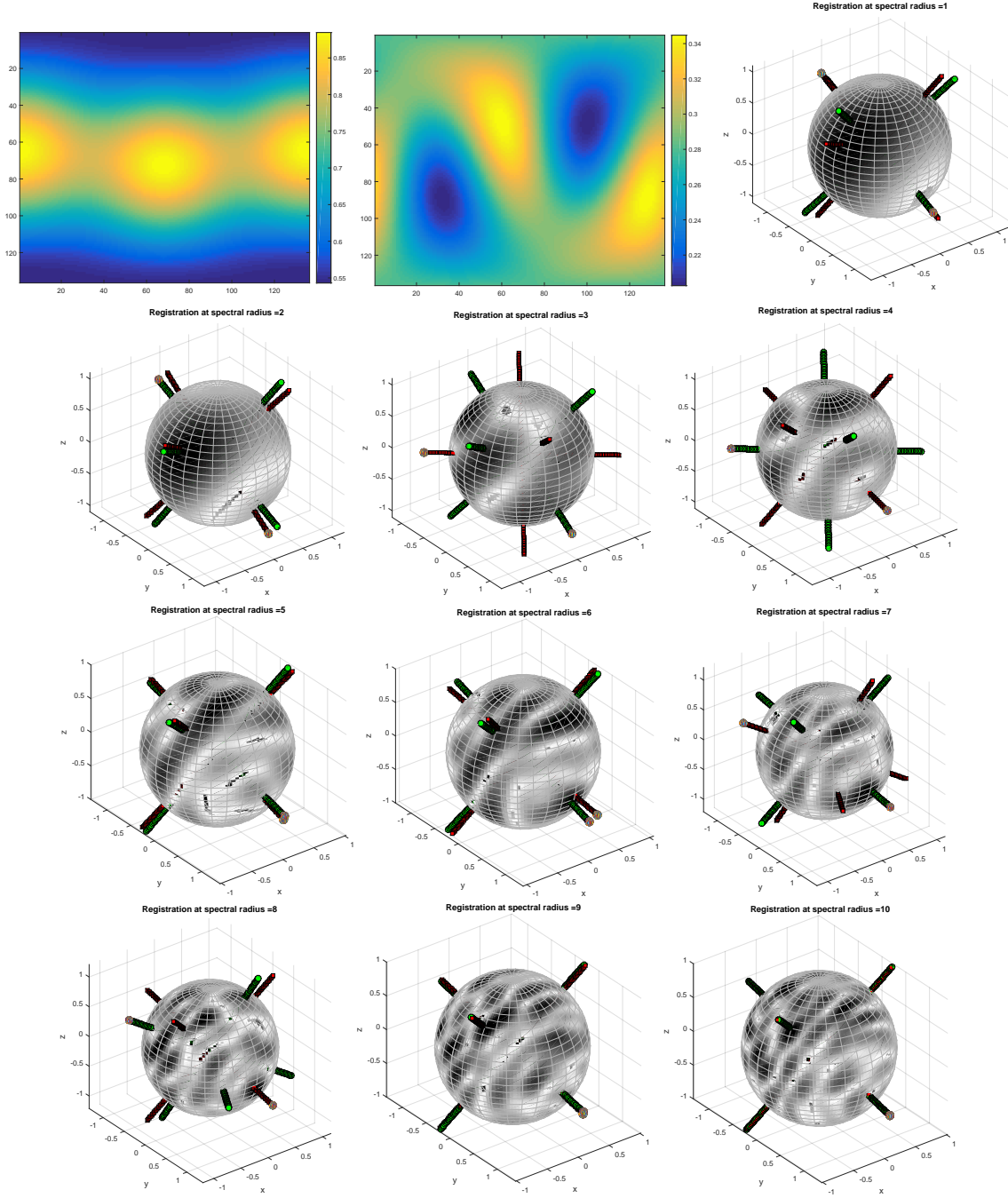


Figure 5.12: Rotation estimation using the proposed method at various spectral radii, which is fully automatic and robust. The top row represent the spherical images obtained at spectral radius 1 of the original and the noisy transformed surface data. The actual solution  $\alpha, \beta$  and  $\gamma$  as discussed in (5.11) is  $= (2.2777, 1.8850, 1.5080)$ , and the estimated solution at radius 10 is near perfect match within the angular resolution  $= (0.7854, 1.8826, 1.6632)$  again with  $\pi$  ambiguity for angle  $\alpha$ .

In this section we consider the case when one of the volumes has missing data or holes. The technique [171] (RESH) requires that the 3D surface (vertices and faces) to be connected regions with no holes on surface. In practice, this topology requirement may not be satisfied due to sensor errors resulting in surfaces with holes. In order to compute the SH coefficients for the surfaces with holes, an interpolation technique is used in RESH. The mesh surface is first converted into voxels, then filled up using a connectivity function, and finally reconverted back to the surface data, see [171] for details. This requires a pre-processing stage of holes filling defined by a user parameter  $\epsilon$  (holes larger than  $\epsilon$  will not be filled). It is difficult to choose an appropriate hole-filling parameter, since on the one hand a very conservative selection would leave the surface with holes, and on the other an aggressive hole filling would render the whole topology altered. Our solution is fully automatic, i.e. we do not require any hole filling process and still, as shown in the simulations, the performance is superior.

We perform our test of holes/missing data on 20 different 3D objects randomly chosen from PSB. Figure 5.11 shows the original surface data of the *Beethoven* bust object, which is first voxelized to obtain the noisy transformed surface data. Figure 5.12 shows the registration spherical maps for different radii. As it can be seen here, the rotation estimation is close at radii  $n = 1, 2, 5, 6$  with  $\pi/2\pi$  ambiguity, while it fails for radii  $n = 3, 4, 7, 8$  and finally it is very close to the ideal solution for radii  $n = 9$  and 10. The RESH technique is robust to very small localized holes, since the hole filling process takes care of it. However, in our case due to significant distortions and the non-localized nature of the holes, it fails. As can be seen in the simulation, due to the missing data, it affects the proposed registration process for different radii differently. The extent of distortion in a spherical layer depends on structural content that is missing in a volume.

Table II shows that our method (SPFT+) produces consistently good estimation of the rotation angles even when the overlap (5.26) reaches up to 30%. Comparatively, RESH is effective only up to 80%, below it the solution is largely random with significant differences in the angular error,

Table 5.2: Table II: Performance in case of holes/missing or partial data.

Method	> 80%	> 60%	> 40%	> 30%	> 20%
SPFT+	0.0980	0.0980	0.1091	0.1030	0.1872
RESH	0.0045	0.6832	1.4060	1.0378	1.4637

and a fine tuning of hole filling will be required depending on the objects and the extent of holes to achieve reasonable estimations.

$$\% \text{Overlap} = \frac{\# \text{Voxels Original} - \# \text{Missing Voxels}}{\# \text{Voxels Original}} \times 100 \quad (5.26)$$

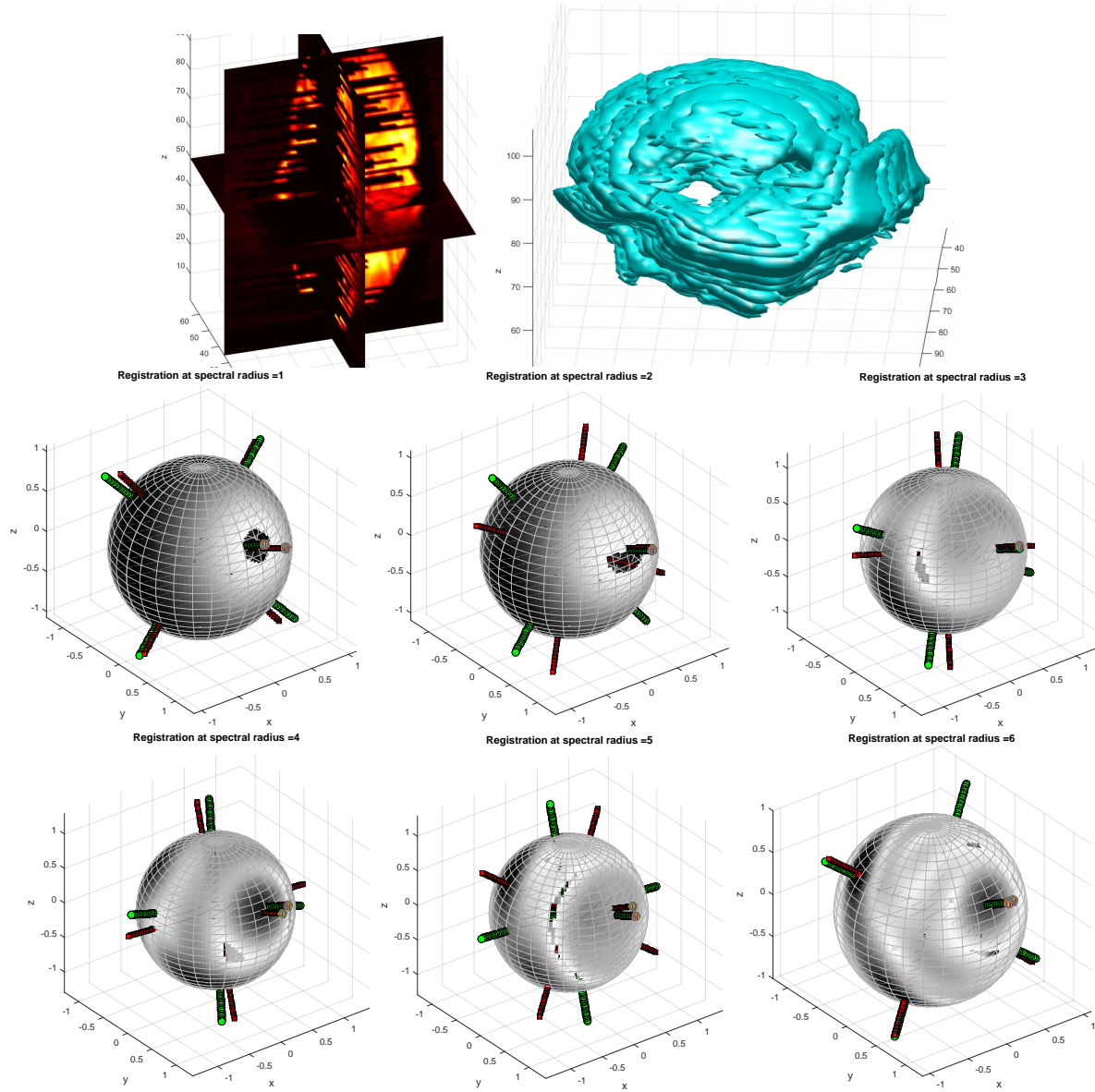


Figure 5.13: Rotation estimation with the same 3D volume and the transformation angles same as in the Figure 5.4. Registration at radius = 6 gives the estimated values =  $(2.4024, 1.0972, 1.6632)$ , which is near perfect registration with  $\pi$  ambiguity for angle  $\alpha$  i.e  $\pi - \alpha = 0.7392$ , which is approximately to the actual orientation  $\alpha$ . This demonstrates that the solution is very robust to missing data/ partial overlap due to noise.

RESH is applicable only for surface data i.e. logical volumes, while SPFT+ is well designed

to handle the full volume data which takes into account the intensity variations along with the shape information. We use the same volumes and rotation angles used previously in Figure 5.4 of Section 5.4.1, but this time we randomly introduce missing data in the images. Figure 5.13 shows the volume data with significant distortions at all layers of the volume. Observe that due to noise, registration is unsuccessful at radius = 3, in fact it fails for radius = (1, 2, 3, 4, 5), but at radius = 6 we get a perfect estimate up to the angular resolution, similar to the noiseless one in Figure 5.6.

As it can be inferred from the above simulations, the proposed method performs very well even with several broken layers in a volume, making it very powerful. Possible reasons for this robust performance are: the use of exact computations for SPFT, the global nature of Fourier transform, the multi-layered approach, and finally the use of full 3D correlations using SOFT. In the following two sections we discuss the advantages with more comparisons.

#### *5.4.5 Real World Challenging Dataset: Crashed Car Parking Disaster Scenario*

A challenging real-world dataset is used here in comparison with the method in [172] (SRMR). This dataset deals with a disaster response scenario using a robot for inspection of a collapsed car parking-lot in Disaster City, Texas, see Figure 5.14. The robot is operated under conditions with high amounts of rubble and dust on slippery surfaces and hence the scans are very noisy. There are a total of 35 scans present, and a total of 35 scan pairs were used to test the proposed method of registration, it is exactly the experimental setup used in [172]. Many of the scan pairs have only small overlaps (range 20 – 70%), and there are many interfering structures, see [172] for details.



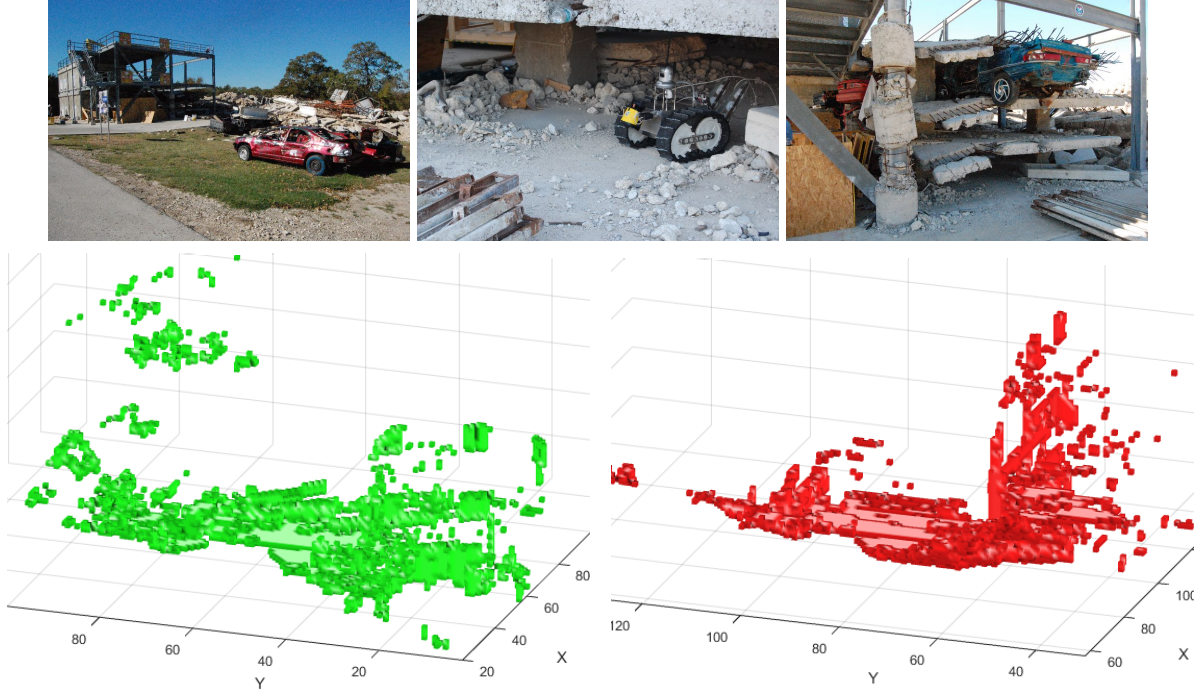


Figure 5.14: The Crashed Car Parking Dataset, (left to right) Front view, Close up of robot used and the close up of the site. For instance, the original volume Scan 12 (left in green) and Scan 13 (right in red) displayed in the form of surface data directly constructed from their respective volumes.

The captured data in the raw form is a set of point clouds, hence we require some preprocessing, i.e. we have to convert it into the volume/voxel form, for the use of our proposed method. There are different methods of triangulation (from point clouds) and voxelization, but we use the same exact set of volumes as was used in [172] for the SRMR technique to conduct our experiments. We do not use or process the original form of point clouds here, but for this kind of discrete geometry processing there are software packages with state of the art algorithms available, for instance see [192] and [193]. Although, we lose some accuracy by this process of transformation of point clouds to voxels due to sub-sampling, nevertheless the overall structure can be well captured by choosing an appropriate voxel size. For our experiments we have used all the scans of equal sizes,



$129 \times 129 \times 129$ . Once the parameters of 6DOF transformations  $(\alpha, \beta, \gamma)$  and  $(t_x, t_y, t_z)$  have been computed between the consecutive scans, it can be applied to the original raw form of point clouds obtained directly from the robot sensor, but the overall registration process can be verified just by using the voxel data as we show in our simulation. Since the computed solution is based on global estimates, it gives a good initial (rough) transformation between the consecutive scans. Further refinements can be done using this initial estimate for a more local approach such as ICP, which in combination with our method gives more accurate registration results. We also show cases when the local refinements fail despite choosing a good rough initial estimate provided by our solution. The validation of the registration result is done by visual confirmation.

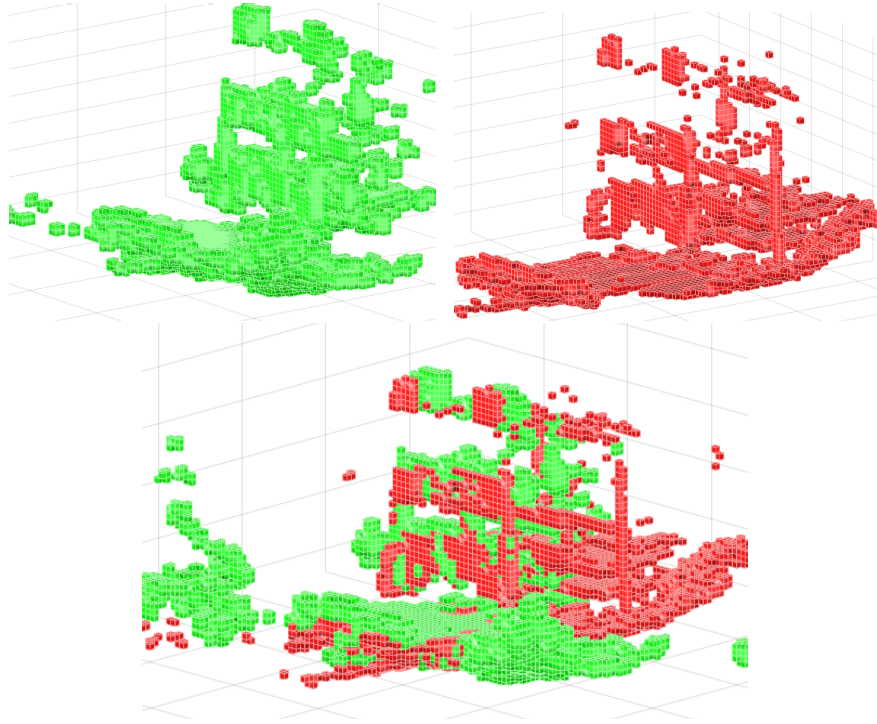


Figure 5.15: (Top row) Volume Scan 12 registered (left in green) and Scan 13 is fixed (right in red). Bottom close up of the combined registered scan directly using the 6DOF computation using the proposed method.

The problem with intensity based image/volume registration [154],[155] and ICP registration [173] (using MATLAB function *imregister* and *pcreregid*, respectively), occurs when there is only a partial overlap between the scans, see Figure 5.16. However, when the volumes have significant overlap local refinements work well to give better registration, see Figure 5.17.

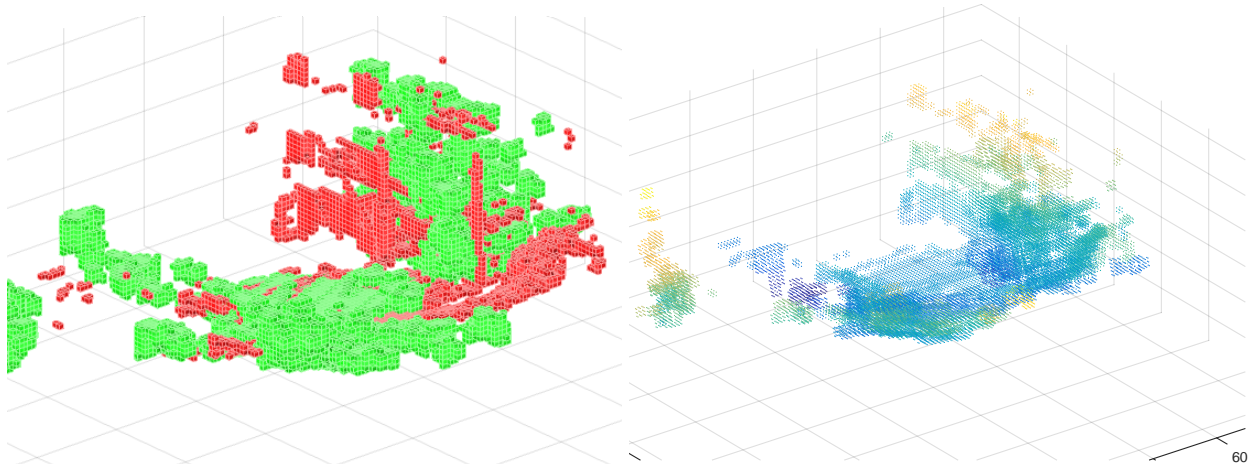


Figure 5.16: A direct attempt to register the two volumes using local intensity based (as seen on left) or location based registration ICP (as seen on right) fails. The local refinement registration ICP fails even after the initial estimate is computed using the global proposed method due to partial overlap. However, this demonstrates that roughly the global alignment can be computed using the proposed method and further local refinements are not useful in such scenarios.

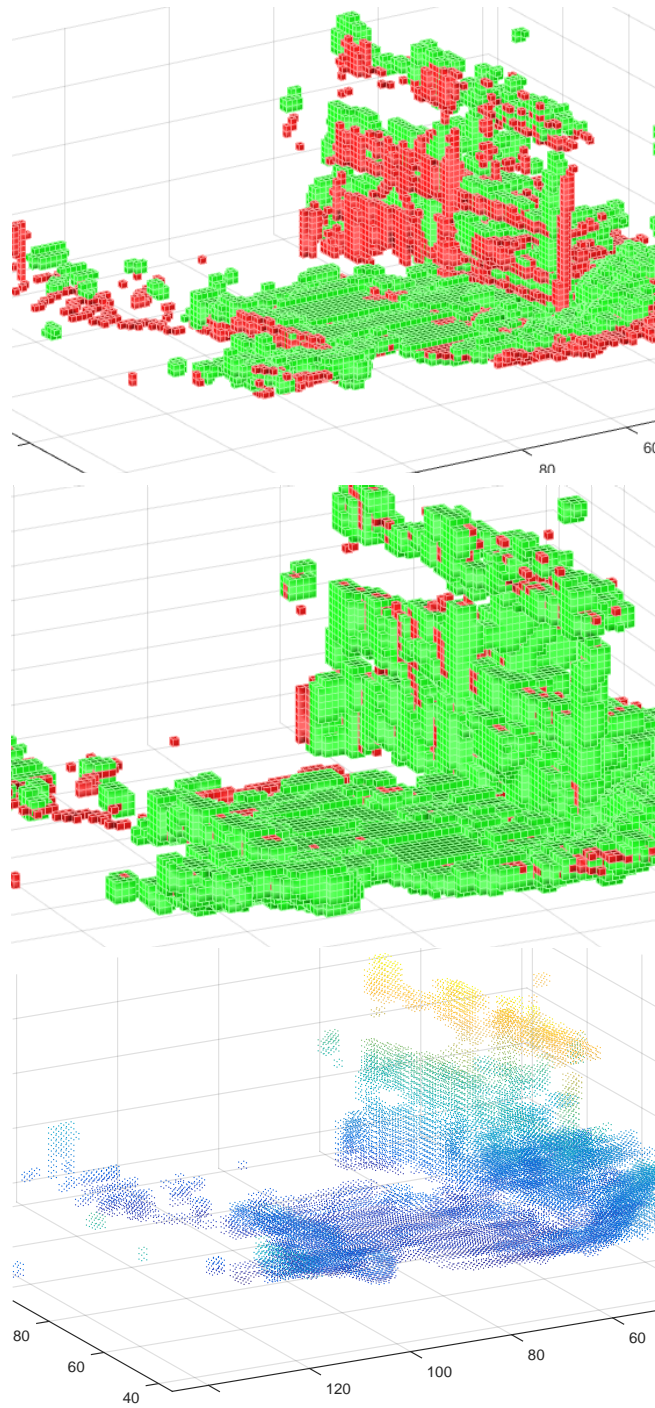


Figure 5.17: Successful refinement using local registration is useful for scans with sufficient overlap e.g. Scans 13 and 14, (left) directly registered globally using proposed method, (center) further refined using image intensity based local registration and (right) refined using ICP.

Out of the 6DOF parameters the rotational vector  $(\alpha, \beta, \gamma)$  is more difficult to determine, the translation vector solution according to (5.24) is usually more robust. The full integrated scan of a total of successful 30 scans is shown in Figure 5.18 and two of the scan pairs, for which where the registration using the proposed solution fails, are shown in Figure 5.19. The failed scans are those with largely flat structures with not much vertical (or height of the 3D structure) information, or poor overlap between the two scans. Out of the 35 scan pairs only three cannot be registered with the proposed technique i.e. 91.% success, which is competitive with SRMR ( 94% success rate or 2 failed scan pairs ) in this context with a difference of only one more failed scan. However, as discussed in Section 5.3.1 there are a number of advantages in using our method compared to the SRMR technique, which is applicable only for small rotation angles. There were many other comparisons reported in [172] using the same exact dataset used, such as ICP-40%, SOFT spectral (single layer) [179]- 83%, plane matching-71%, etc. Interestingly, even a single layer SOFT technique was comparable or second best to the SRMR method, and we have improved (compared to the single layer SOFT in [179]) its success rate and made it more robust by introducing the multi-layered approach in this chapter.

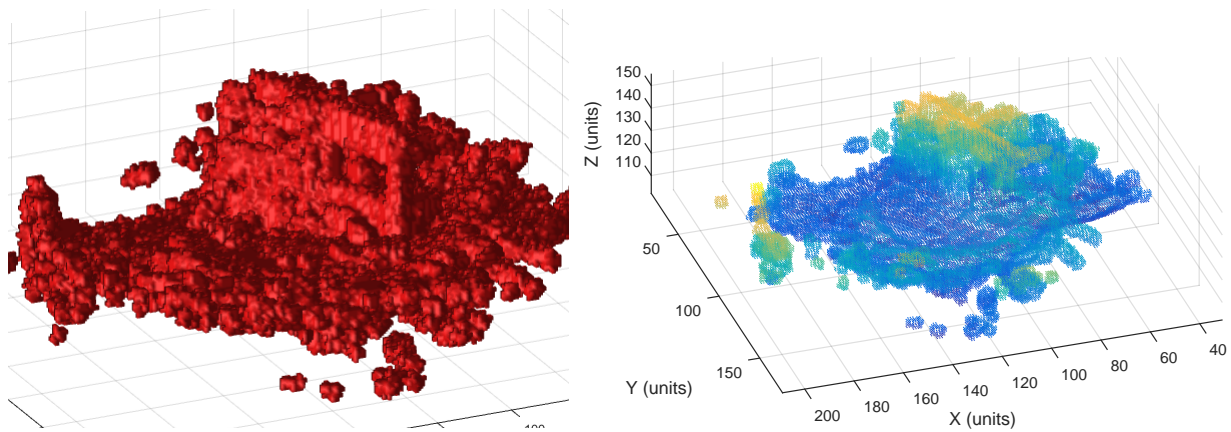


Figure 5.18: Final integrated plot of 30 (top and bottom) the volume data and its corresponding point cloud set, as can be seen all the scans have been well aligned globally(roughly) from the (sub-sampled) volumetric data alone. Further local refinements can be done to improve the alignment by directly using the original point cloud data.

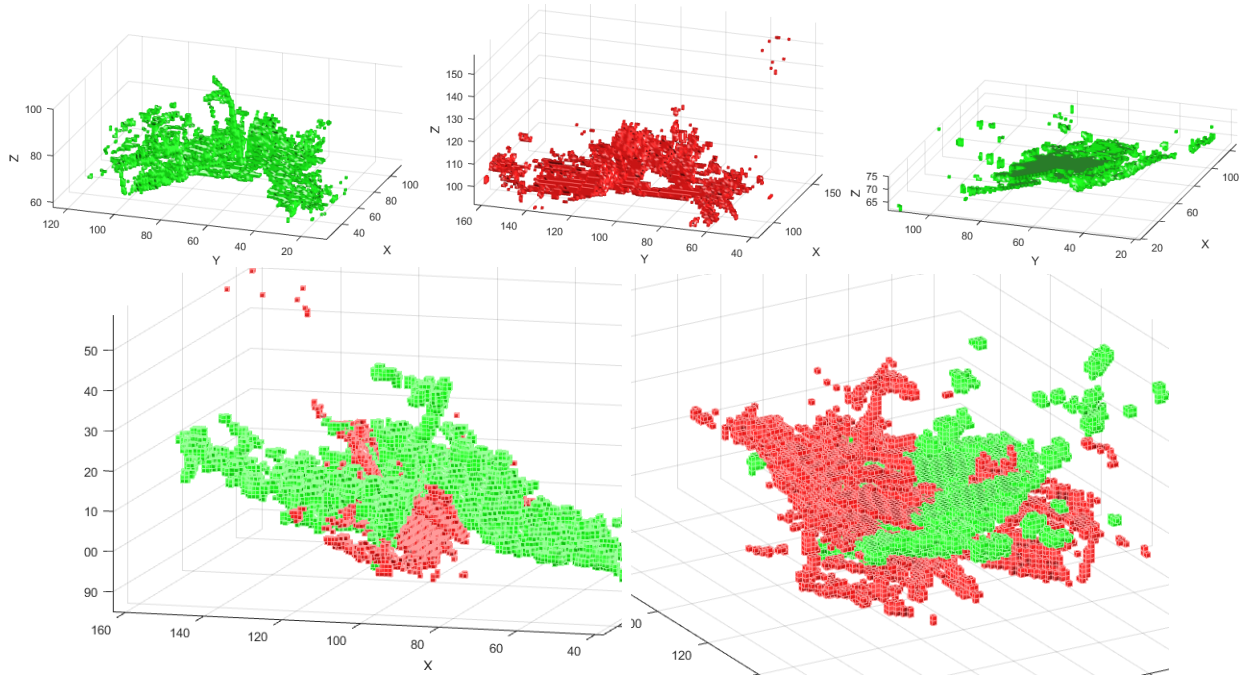


Figure 5.19: Failed Scans. (Top row) Scan 1,2 and 35. (Bottom row, left to right) failed registration for scan pair 1-2 (green-red), failed registration for scan pair 35-1(green-red).

#### 5.4.6 Comparison with Go-ICP Method for Partial Overlap

The global ICP algorithm is a recent development on the previously well-known ICP technique [173]. The ICP algorithm is the most widely used method of 3D registration with point-set data. However, it is based on local iterative optimization, and hence susceptible to converging to a local minimum. A global optimal solution to the problem of 3D registration with point-set data was recently proposed in [175] using the standard point-to-point ICP approach. The Go-ICP method is based on a branch-and-bound (BnB) scheme that searches the entire 3D motion space  $SE(3)$  or 6 parameters to obtain the best globally optimal solution.



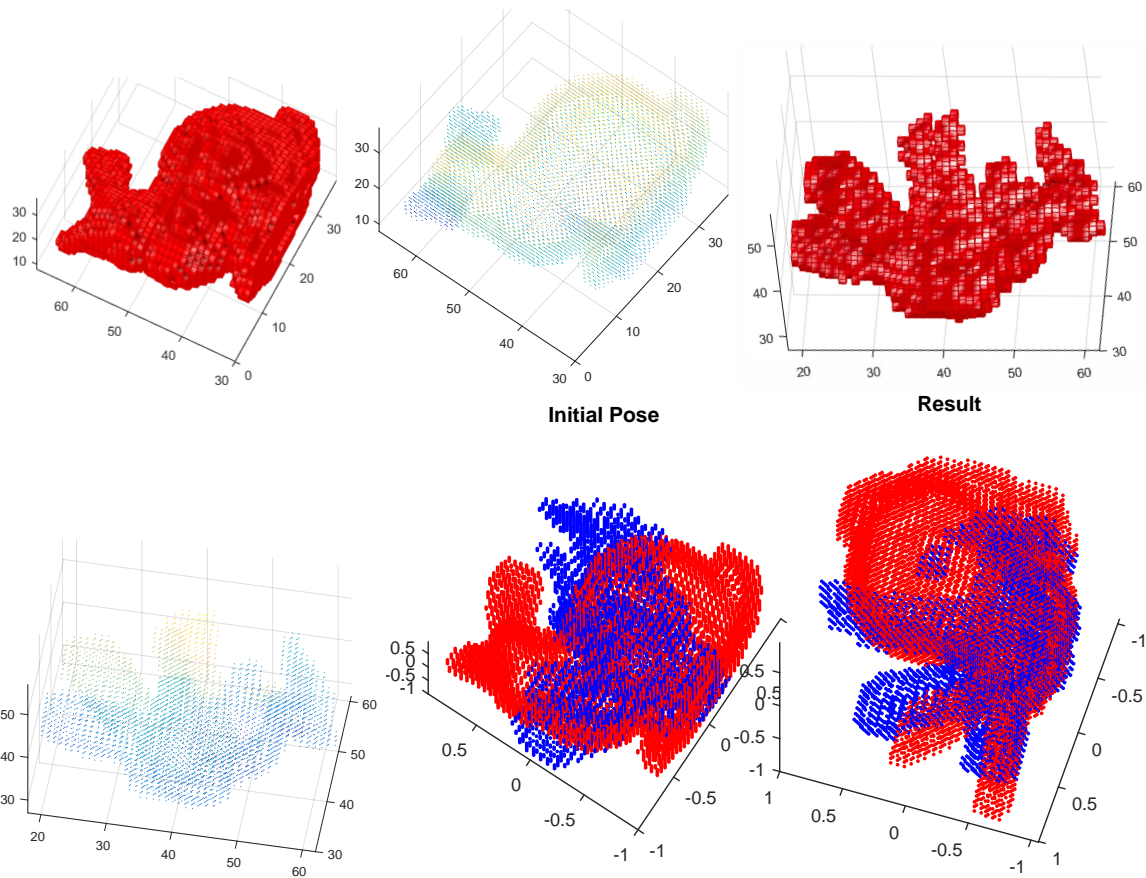


Figure 5.20: (Top to bottom) (a) Original voxelized bunny object and its point cloud representation (b) Partial and rotated voxelized part of bunny and its corresponding point-set data (c) (viewed from a vantage point) Go-ICP with noisy and partial data, solution. The actual rotation Euler angles are (1.2794, 1.1718, 0.7226) and the Go-ICP estimate (0.2599, 1.2290, 4.0183) are close but not very accurate. The run time here is significant (in hours).

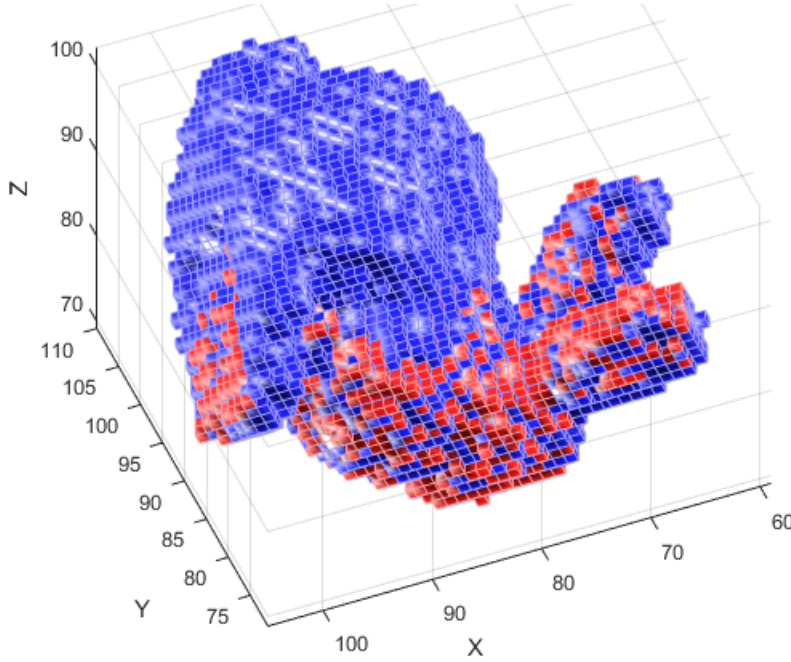


Figure 5.21: The final solution using the proposed method. The angles computed with the proposed solution are very close to the actual one (1.2566, 1.1192, .7330) even when the overlap is just 38.76%. We use  $B = 120$ , i.e. the two spherical grids (5.12) are of size  $240 \times 120 \times 115$ , using an optimized GPU based computation on a commodity NVIDIA GTX 760 hardware.

For comparisons, in order to maintain consistency between the point cloud data used for Go-ICP and the voxel data required for our method, we first voxelize the objects from the PSB database and retain the same vertices that represent the corners of the voxels. Thus the same voxels and the exact vertices of the voxels are used for the two methods for a fair comparison. Figure 5.20 shows the registration results for the Bunny object using the Go-ICP, and Figure 5.21 shows the same registration using the proposed method, where only 3D rotation transformation exists between the two volumes. For estimating the run time of the Go-ICP method, we repeated the experiment for 20 trials and the average running time estimated = 5377.9 seconds. Note that the run time here depends on the signal content and the extent of the overlap between the two objects. There is



38.76% overlap present between the fixed bunny and the rotated partial bunny 3D object. The Go-ICP estimate is not very accurate despite taking a lot of time searching a truly global solution compared to the proposed technique. The run time for the proposed method using the two volumes of size  $115^3$  are 294.17 and 294.58 seconds, which is fixed and independent of the object shapes or the overlap between them. For the execution time of the algorithm SPFT and more details, which takes up most of the time in the above procedure, the reader is referred to [183]. Note that we do not need the entire spherical polar grid for robust 3D registration, only a few spherical layers (spectral radii  $n \in [1, 11]$  with image pairs of size  $240 \times 120$ ) will suffice, so that the solution can be computed faster than reported here. For the above experiment, we used only up to 11 spherical layers to choose the most robust estimation of the angles.

A possible reason for the slow processing of Go-ICP maybe due to the exponential increase in the 6D search space with the increase in the number of points and reduction in the overlap. Even if we consider that the current code is not very optimized and hence the execution speed is low, the final estimates are still not as accurate as the proposed solution. Due to the internal use of ICP in the Go-ICP algorithm, this technique is not truly a global registration method, but a refined approach to repeatedly re-initialize the ICP method, which is strictly local within the error bounds. The Go-ICP due to its local nature depends on the signal content, and hence takes variable time to converge depending on the noise level, unlike the proposed method where the computations are fixed once the resolution and number of layers required are selected. Finally, in case of noise and outliers, the Go-ICP requires careful selection and refinement of statistical outliers before processing, while the proposed solution is obtained without any such refinements but may still benefit from some outliers rejections. Note that there is a parameter selection required by the user for the number of points (vertices) to be used between the two objects which defines the limits of the search space, e.g. from 1000 points to 2000 points the run time increases significantly (2 – 3 times higher). Thus the proposed method provides a truly global non-iterative solution to the 3D

rotation problem, it is faster than the Go-ICP method, has a fixed run time and complexity. The final solution computed using our method is also shown to be more accurate than the Go-ICP method. We conducted experiments using additional 10 randomly selected different objects from the PSB dataset, the average running time for each experiment (using both methods) with each object was computed in 20 trials. Figure 5.22 shows the final comparison results in terms of speed and accuracy, respectively. It can be seen from these experiments that the Go-ICP method running time is often unpredictable and its accuracy still is not very competitive. However, it may be more suited to point clouds of smaller sizes. The figure demonstrates the superiority of the proposed approach as it has fixed time complexity and robust accuracy.

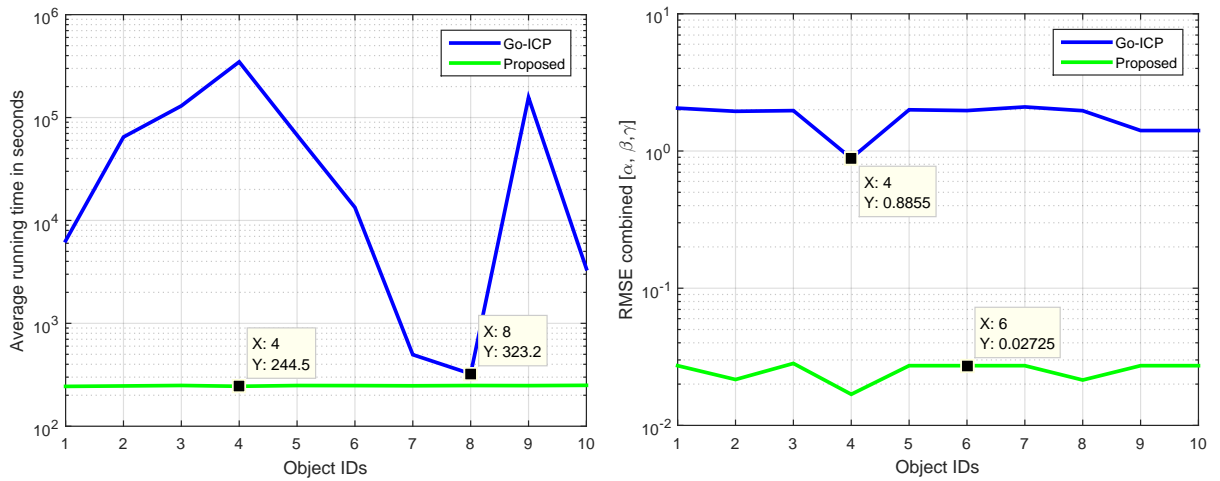


Figure 5.22: (Left) Speed comparison (Go-ICP can take a lot of time-from hours to days) and, (Right) Accuracy. In the above experiments we fixed the actual rotation vector,  $(\alpha, \beta, \gamma) = (1.2794, 1.1718, 0.7226)$ , and the two volumes were of size  $67 \times 67 \times 67$ .

## 5.5 Contributions and Conclusions

The main innovation in our proposed robust 3D registration method is the use of multi-layered spectral data for correlation on the spheres, which have the redundant 3D rotational information

encoded across all spherical layers, to perform the globally optimal 3D rotation estimation, best selected from a set of solutions present in different radii, obtained with the combination of fast and algebraically exact SPFT and SOFT algorithms.

The main contributions of this chapter are summarized as follows:

- We have designed a general method to compute the rotational angles and perform the registration given the two 3D volumes in Cartesian coordinates related by up to 7DOF transformation.
- The proposed solution is extremely robust to noise, partial overlap or missing data, and other distortions, due to the redundant nature of the computations using multi-layered spectrum.
- The algorithm presented provides a true globally optimal estimate for the rotation estimation problem, which is the most difficult part of the 3D registration process.
- The proposed method offers superior performance compared to the three state of the art methods for obtaining very accurate 3D rotational angles information.
- We do not require any interpolation or approximation stage in our method, and there is no need for user controlled hole filling/pre-processing, since our technique is very robust and fully automatic.
- The algorithms for both SPFT and SOFT are of fixed complexities and are fully parallel in principle hence a parallel implementation can take full advantage of commodity GPGPU processing hardware. Most importantly each of the components used in our solution has a parallelizable structure and scales naturally with processing and memory resources.
- Finally our solution is easily extensible to higher dimensions, again using fast and exact FT computations for *hyper-spherical* FT and SOFT, [183], [179], where the geometric structure

is rich and complex.

Further research is needed to seek more problems where the proposed registration technique can be successfully applied, and to study how it can be extended for more complicated transformations such as affine and projective. In future, it would be interesting to see an application of high dimensional extension of the proposed technique in terms of performance, complexity, parallel implementation, network constraints, etc.

## CHAPTER 6: FUTURE WORK

The Roman poet, Lucretius, who anticipated the core scientific vision of modernity, begins his work (in Martin Ferguson Smiths careful rendering) with an ardent hymn to Venus, the goddess of love, whose coming in the spring has scattered the clouds, flooded the sky with light, and filled the entire world with frenzied sexual desire:

First, goddess, the birds of the air,  
pierced to the heart with your powerful shafts,  
signal your entry. Next wild creatures and cattle bound  
over rich pastures and swim rushing rivers: so surely  
are they all captivated by your charm and eagerly follow your lead.  
Then you inject seductive love into the heart of every creature  
that lives in the seas and mountains and river torrents and bird-haunted  
thickets, implanting in it the passionate urge to reproduce its kind.

- *Prose translation of Lucretius two-thousand-year-old poem*

*“On the Nature of Things” (“De Rerum Natura”)*

*found in (Pulitzer) ‘The Swerve: How the World Became Modern’*

*by Stephen Greenblatt, Yale, 2012.*

These are the set of few incipient ideas that are being explored currently in my research, some of them also have good preliminary results. Most of these ideas can be potentially developed to a full fledged decent journal articles. One of main ideas that we are interested in exploring more is the algorithm in Polar / Spherical DFT. This was an age old problem, and we solved in an innovative and efficient way, the practical applications of this transform is immense. Another idea is based on nonuniform frequency estimation procedure that we proposed. Finally we discuss the idea of combining frequency domain techniques with machine learning and deep neural networks.

## 6.1 High-Dimensional Extension of Polar / Spherical DFT

The first logical step after proposing 2D and 3D fast nonuniform spherical FT as shown in chapter 4 is to extend this solution to  $N$ D. This is the idea that we formulate in this section. By following the methodology employed in the construction of the 2D discrete *Polar* FT and 3D discrete *Spherical Polar* FT in chapter 4, we can generalize it to have  $N$ -D discrete *Hyperspherical* FT given a  $N$ -D spatial discrete data. In the construction of the discrete *Polar* or *Spherical Polar* FT we employed four key paradigms that would allow easy extension to higher dimensions:

- We partition the Fourier domain exact computation on the basis of levels, for 2D we have just one partition denoted by  $l$  and for 3D we have nested two levels denoted by  $p$  and  $q$ .
- Each partition is then reduced to a set of radial lines 1D, and then we compute reduced form of butterfly grid for each partition.
- Butterfly grid's computation is relatively straightforward, only in one axis it is uniformly scaled using 1D FrFT and it is the most expensive to compute.
- The rest of the Fourier coefficients computation depends on the geometry of the block where we only compute the desired points in the second pass and not on the entire data.

All the steps can be used as a basis for establishing the discrete high dimensional transform definition, we formulate the problem of computing discrete *hyper-spherical* FT as follows:

Let  $N$ -dimensional Euclidean space be consisting of a radial coordinate  $\rho$  and  $N - 1$  angular coordinates  $\phi_1, \phi_2, \dots, \phi_{N-1}$  where  $\phi_{N-1} \in [0, 2\pi)$  and the other angles range over  $[0, \pi]$ . If

$x_1, x_2, \dots, x_{\mathbf{N}}$  are the Cartesian coordinates, then,

$$\begin{aligned}
x_1 &= \rho \sin \phi_1 \\
x_2 &= \rho \cos \phi_1 \sin \phi_2 \\
x_3 &= \rho \cos \phi_1 \cos \phi_2 \sin \phi_3 \\
&\vdots \\
x_{\mathbf{N}-1} &= \rho \cos \phi_1, \dots, \cos \phi_{\mathbf{N}-2} \sin \phi_{\mathbf{N}-1} \\
x_{\mathbf{N}} &= \rho \cos \phi_1, \dots, \cos \phi_{\mathbf{N}-2} \cos \phi_{\mathbf{N}-1}
\end{aligned}$$

the transformation from Cartesian coordinates spatial domain discrete function of size  $(N + 1)^n$  where  $N$  is even, to *hyper-spherical* coordinates Fourier domain discrete function of size  $K_1 \times K_2 \times \dots \times K_{n-1} \times (N + 1)$  where the origin is included multiple times and  $K_j$  is the (even) total number of angles for angular coordinate  $\phi_j$ , is given by,

$$\begin{aligned}
T : f(m_1, m_2, \dots, m_n) &\rightarrow F(k_1, k_2, \dots, k_n) \\
F(k_1, k_2, \dots, k_n) &= \sum_{m_n=-N/2}^{N/2} \dots \sum_{m_2=-N/2}^{N/2} \sum_{m_1=-N/2}^{N/2} \\
&f(m_1, m_2, \dots, m_n) e^{-i \frac{2\pi k_n}{N+1} \mathbf{Z}}
\end{aligned} \tag{6.1}$$

where,

$$\begin{aligned}
\mathbf{Z} &= m_1 \sin(k_1 \Delta\phi_1) + m_2 \cos(k_1 \Delta\phi_1) \sin(k_2 \Delta\phi_2) + \cdots \\
&+ m_n \cos(k_1 \Delta\phi_1), \cdots, \cos(k_{n-2} \Delta\phi_{n-2}) \cos(k_{n-1} \Delta\phi_{n-1}) \\
\Delta\phi_1 &= \frac{180^\circ}{K_1}, \Delta\phi_2 = \frac{180^\circ}{K_2}, \cdots, \Delta\phi_{n-1} = \frac{180^\circ}{K_{n-1}} \\
k_1, k_2, \cdots, k_n &= -\frac{N}{2}, \cdots, \frac{N}{2}.
\end{aligned}$$

More research work in terms of writing the mathematical equations, then generic formulation of partitioning technique we described above, and the finally algorithmic implementations using fast and efficient C++/CUDA libraries such as Boost, ArrayFire, etc. needs to be investigated. This could very well become our good research paper in the future for a prestigious scientific computing journal.

## 6.2 Fast and Exact Fourier Volume Rendering with Polar / Spherical DFT

In this case we do have some early preliminary results, and we show that the solution can be used for processing/slicing volumes in 3D or beyond with spherical / cylindrical coordinates.

Fourier volume rendering (FVR) is a significant visualization technique that has been used widely in digital radiography. It plays a significant role in medical imaging as it allows exploring the internal structures of volumetric data acquired by different imaging modalities. This exploration allows accurate diagnosis and consequently effective treatment. It relies on a sound *Fourier projection-slice theorem*, where using the spectral representation of a 3D spatial data and taking a central 2D slice generates attenuation-only projections that looks like *X-ray radiographs*. We propose an alge-



braically exact and fast algorithm to compute such a central slice with arbitrary orientation (angle) which has the special feature that it involves only 1D equispaced FFT's and no interpolation or approximation at any stage. The result is we get the fast solution within machine precision without any oversampling and devoid of any artifacts. We also propose an exact and fast algorithm to simultaneously compute multiple planar projections that are angularly equispaced using 3D discrete cylindrical Fourier transform.

Volume visualization or rendering is an essential tool for exploring and analyzing the anatomy of complex structures and phenomena, it is used in various scientific and engineering arenas such as medical imaging, geo-science, microscopy, mechanical engineering, and others [194]-[196]. An extensive review of the classical theory behind FVR can be found in [197],[198],[199]. Volume rendering algorithms can be classified into spatial-domain and other-domain-based techniques such as frequency domain, compression domain or the wavelet domain. In this section we discuss a new way to perform Fourier domain based volume rendering. This technique has considerable significance in the medical discipline for the following reasons:

- It produces attenuation renderings that look like the *X-ray radiographs* which the medical professionals are well trained to interpret.
- It can be used to get a sliced view of the full volume data along any direction for medical analysis.
- Due to its fast implementation using FFT's it can be suitable for visualizing of large scale data sets.

FVR uses a 3D spectral representation of the volume to compute an image that looks like *X-ray radiograph* relying on the projection-slice theorem. It works by transforming the spatial volume into frequency domain. Then it reconstructs 2D projection at any viewing angle by resampling

an extracted projection-slice along a perpendicular plane to the viewing direction followed by backtransforming this resampled slice to the spatial domain [198],[200],[201] and [202]. However practical implementation of the FVR technique is complicated by three main factors which arise when the projection slice theorem is applied to discrete sampled data, [200] and [202]. In a typical FVR algorithm a projection slice is extracted from the spectrum at certain angle specified by user. This stage is followed by an optional resampling stage to reduce the aliasing and ghosting artifacts from the final projection. The interpolated slice is then back-transformed by inverse 2D FFT operation, and the resulting projection image is wrapped-around to set its origin at the center of the image. First when the projection slice is desired along an arbitrary direction conventional FFT algorithm cannot be applied to yield a plane in the desired angle so interpolations must be used. To reduce the aliasing and ghosting artifacts in the reconstructed image, this projection slice is processed in a further step to have it resampled. The resampling stage is only mandatory if the desired projection was not orthogonal. Second the FFT algorithms yield frequency-domain output data which is not ideally structured for resampling. To mitigate this effect, it is necessary to add high performance multidimensional FFT-shift stages to the rendering pipeline to rearrange the data. Moreover, sampling the frequency domain is equivalent to the replication of the signal in the spatial domain that ultimately accounts for the appearance of ghosting artifacts in the reconstructed images. This issue is resolved by zero padding the volume in the spatial domain and using high order interpolation filters in the frequency domain. Finally there is no straightforward way for direct extraction of the complex projection slice from the 3D spectrum [203] so the entire set of operations must be performed two times one for real part and one for complex part of the spectral 3D data.

There exists no fast and exact solution to the FVR technique. In literature, several volume rendering algorithms have been presented either to improve the rendering speed of large datasets or to enhance the rendering quality of their reconstructed images. By and large, the rendering speed

and quality are traded-off and no single complete algorithm that can deliver the optimum quality associated with maximum speed exists [204]. Recently we introduced a fast and exact solution to the discrete polar and spherical polar transform [183]. We use a similar strategy by using 1D Fractional Fourier transform (FrFT) but tailored specifically for our FVR algorithm which gives us the only known solution that is both fast and algebraically exact.

### 6.2.1 Algorithm I: Single Planar Projection

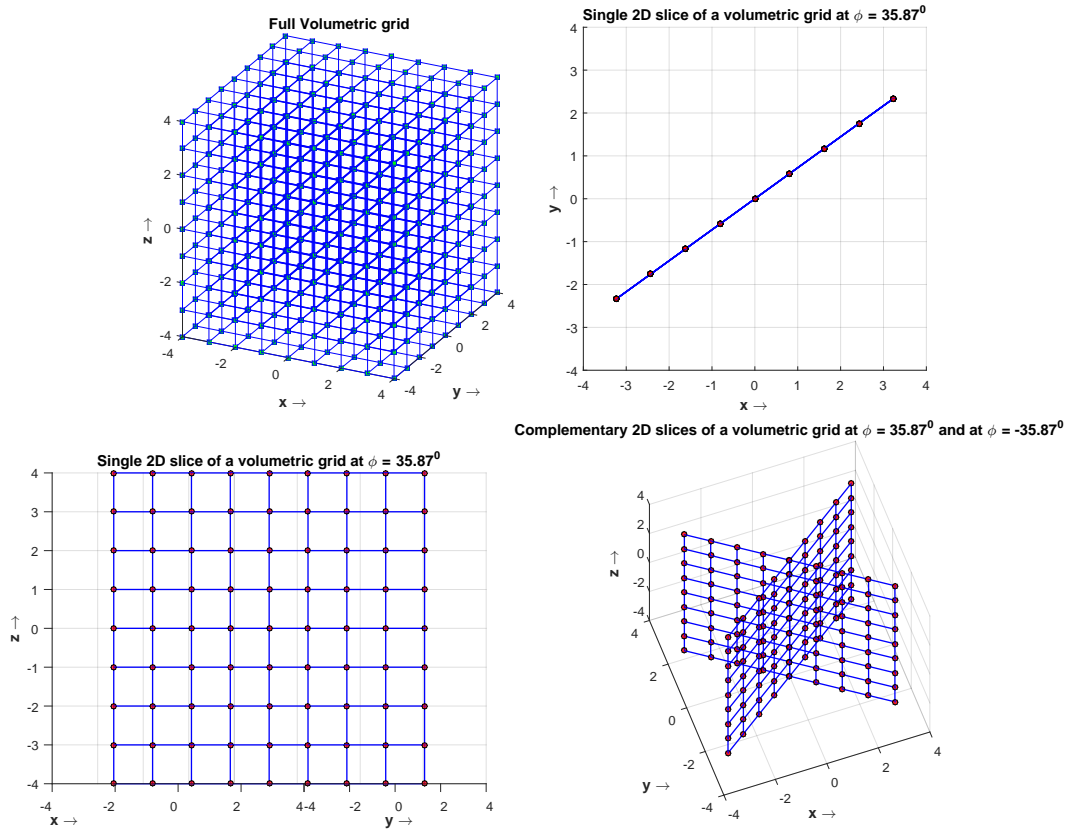


Figure 6.1: Direct and exact 2D slices computation of a full 3D medical volumetric data. Clockwise from top left: (a) 3D Volume data such as CT scan (b) Top view or X-Y view of the 2D slice (c) 2D grids of complementary slices with respect to X-axis (d) Orthographic view of the slice which is a 2D equi-spaced normal grid.

We can define a plane oriented at an arbitrary angle  $\phi$  in the X-Y plane see Fig 6.1. Then the 1D frequency domain scaling of a 3D image along X-axis can be computed using 1D FrFT along X-axis such that,

$$\begin{aligned}
F_x^\alpha(r, c, d) &= \sum_{n=-N/2}^{N/2} f(n, c, d) e^{-j \frac{2\pi r \alpha n}{N+1}} \\
\alpha &= \cos \phi, \\
r, c, d &= -\frac{N}{2}, \dots, \frac{N}{2}
\end{aligned} \tag{6.2}$$

Next we can scale the image  $f$ , in Y-axis, we can compute 1D FrFT and the equation is given by,

$$\begin{aligned}
F_y^{\alpha, \beta}(r, c, d) &= \sum_{n=-N/2}^{N/2} F_x^\alpha(r, n, d) e^{-j \frac{2\pi c |\beta| n}{(N+1)}} \\
\beta &= \sin \phi, \\
r, c, d &= -\frac{N}{2}, \dots, \frac{N}{2}
\end{aligned} \tag{6.3}$$

Finally we scale the transformed image in Z-axis,

$$\begin{aligned}
F_z^{\alpha, \beta, \gamma}(r, c, d) &= \sum_{n=-N/2}^{N/2} F_y^{\alpha, \beta}(r, c, n) e^{-j \frac{2\pi d \gamma n}{(N+1)}} \\
\gamma &= 1, \\
r, c, d &= -\frac{N}{2}, \dots, \frac{N}{2}
\end{aligned} \tag{6.4}$$

Using the (6.2) and (6.3) we gather points falling on the 2D plane which is the 2D projection-slice passing through the origin of the spectral volume at an angle  $\phi$ . This carefully extracted slice represents the 2D FFT of the desired projection and thus, the reconstructed image can be directly obtained by a 2D inverse FFT operation.

The last equation (6.4) is nothing but centered FFT along Z-axis. By simply rotating the entire

volume, and performing the operations using the above equations we can compute central slices along any direction. Thus any arbitrary plane with angle  $\phi$ , can be computed using systematic frequency domain scaling of the spatial volume grid.

This gives us the true 2D projection solution that is algebraically exact. It only involves the use of 1D FFTs and hence it is fully vectorizable, memory efficient and fast. It does not involve any design parameter choice for user, it involves no preprocessing, oversampling or padding of any kind. The computation of the 1D FrFT of the volumetric data is also completely independent, and it is easy to compute each vector in a parallel way on a GPU, multi-CPU, multi-GPU and grid computing (CPU/GPU) systems.

#### 6.2.1.1 Computational Complexity

The most expensive operation is the first uniform scaling by  $\alpha$  along  $X$  axis. We have the complexity for  $X$  axis scaling of the full 3D volumetric grid using real data,

$$(N + 1)^3/2 \times \log_2(N + 1)$$

Next scaling is in Y-axis and is differential scaling. Since we are interested in just point on the 2D plane oriented along a specified direction this operation simply reduces to a Hadamard element wise multiplication, hence we have the reduced complexity,

$$(N + 1)^3$$

Finally we compute 1D FFT on a 2D plane along Z -axis which is uniform and requires no scaling, the computational complexity of 1D-FFT is then,

$$(N + 1)^2 \times \log_2(N + 1)$$

Hence the total computational complexity is dominated by the first uniform scaling for  $N$  large, so the total computational complexity for each slice at an arbitrary angle  $\phi$  is,

$$(N + 1)^3/2 \times \log_2(N + 1) \quad (6.5)$$

Note that during this computation the complementary slice planes at angles  $180 \pm \phi$  for X-axis orientation or basically horizontal planes or  $90 \pm \phi$  for Y-axis orientation or basically vertical planes can also be computed without any overhead, which bring us on to the topic of multiple simultaneous projections.

### 6.2.2 Algorithm II: Multiple Planar Projections

Given a rotation axis  $\vec{n}$  that is parallel to one of the principle axis X,Y and Z, we can define the multiple planar projections conveniently using cylindrical coordinates representation, see figure 6.2. Consider first the Z-axis, this arrangement of cylindrical coordinates could be considered as exact 2D polar slices stacked along the height of the cylinder. We define the cylindrical FT of a volume  $f$ , denoted by  $F^C$ , given by,

$$F^C(f) = F^P(F_z(f)) \quad (6.6)$$

where  $F_z$  is the centered 1D FT in the  $z$  direction and  $F^P$  is the 2D polar FT which operates on each plane that is perpendicular to the  $z$  axis. To compute the DFT for 3D cylindrical coordinates,we

compute 1D centered FFT in the  $Z$  direction, then we compute 2D Polar FFT for each plane perpendicular to the  $Z$ -axis. In our case since its a unified volume,  $N + 1$  is the height of cylinder and let  $M$  be the even number of desired equidistant projections, then taking a 2D discrete polar Fourier transform (DPFT) for each plane perpendicular to complexity given by,

$$\frac{M}{4} \times (N + 1)^2 \log_2(N + 1) \quad (6.7)$$

See [183] for more details about fast and exact computation of 2D DPFT. Thus, we have the total computational complexity for full 3D discrete Cylindrical Fourier transform as,

$$(N + 1) \times \frac{M}{4} \times (N + 1)^2 \log_2(N + 1) \quad (6.8)$$

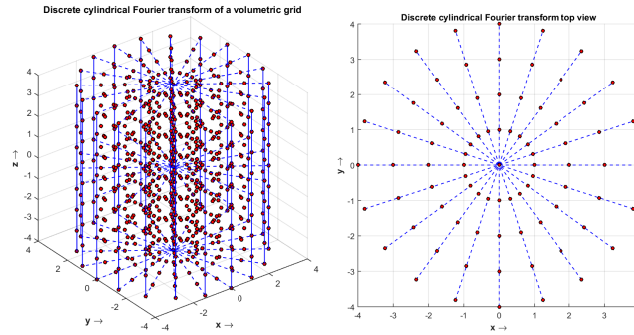


Figure 6.2: Multiple projections using cylindrical Fourier transform of a full volumetric grid.

When multiple planar projections are desired and all the projections are angularly equispaced about an axis such as  $Z$ , this method gives a one shot highly optimized solution for the problem with complexity  $\frac{M}{4} \times (N + 1)^3 \log_2(N + 1)$ . This is just one time computation which can be preprocessed for the entire volume to compute 3D discrete Cylindrical Fourier transform, and all the subsequent computations of the projections is now greatly simplified, since any projection can then be retrieved

using an inverse 2D FFT of complexity,  $(N + 1)^2 \log_2(N + 1)$ .

### 6.2.3 Experiments

We use 3 real world, publicly available, standard datasets of *visible male*, *skull* and *Foot*, to demonstrate our results. All the datasets are acquired and originally organized in regular 3D Cartesian grids [205],[206]. We compute multiple arbitrary projections of the 3D data to compute 2D radio-graphs and demonstrate the accuracy of the proposed solution. Since they are fast and exact we achieve very high accuracy but more research is needed to compare times.

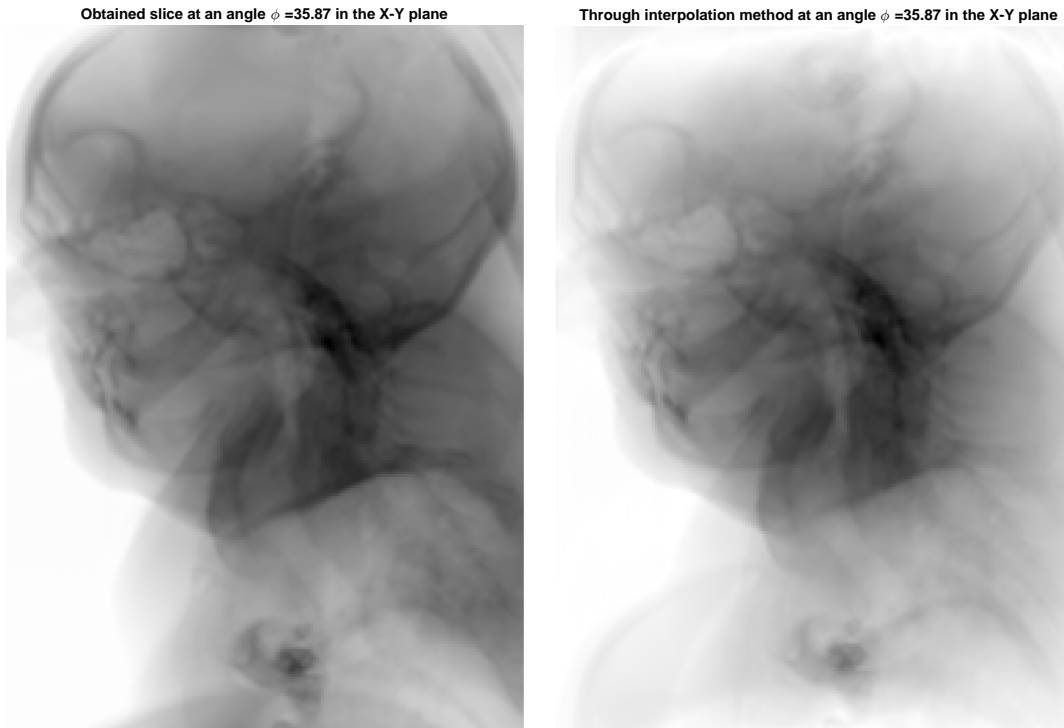


Figure 6.3: Single slice at a specified angle, used the complement of the image to enhance the effects of distortion. See the interpolated slice has poor reconstruction at the slice.



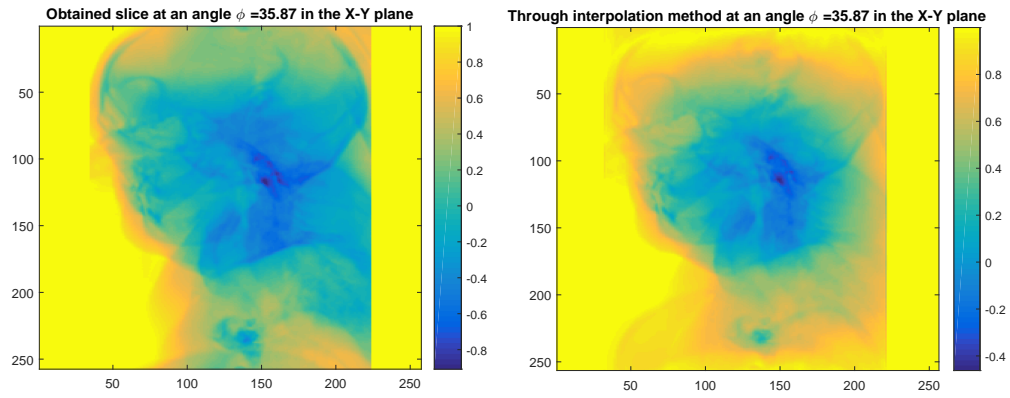


Figure 6.4: Ghosting artifacts and problem along sharp edges, while the exact reconstruction directly from the volume data using the proposed method is dense, clean and sharp.

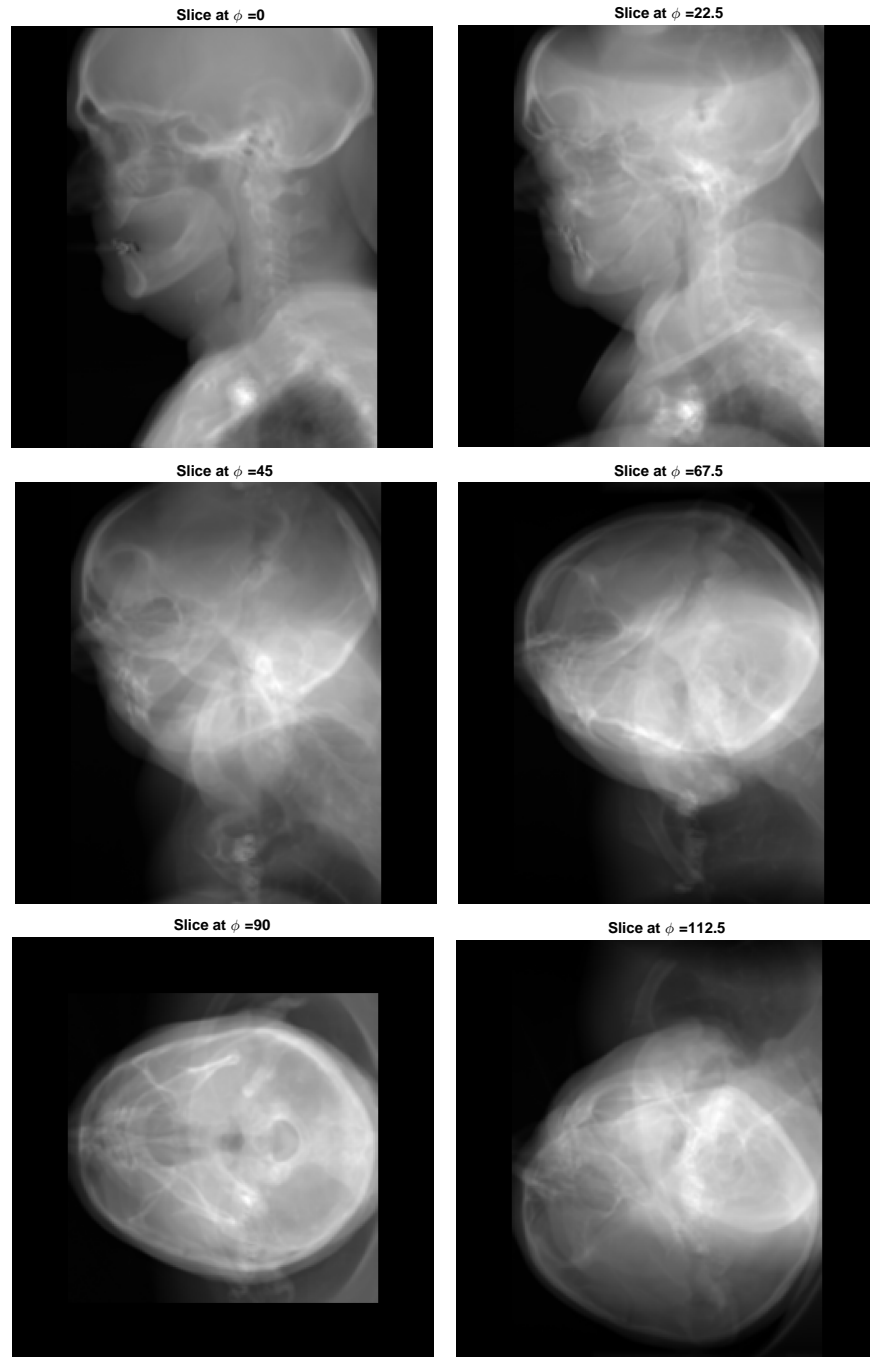


Figure 6.5: Full volumetric data analysis using the proposed FVR method on *visible male* dataset, starting from top left at increasing angles of slices.

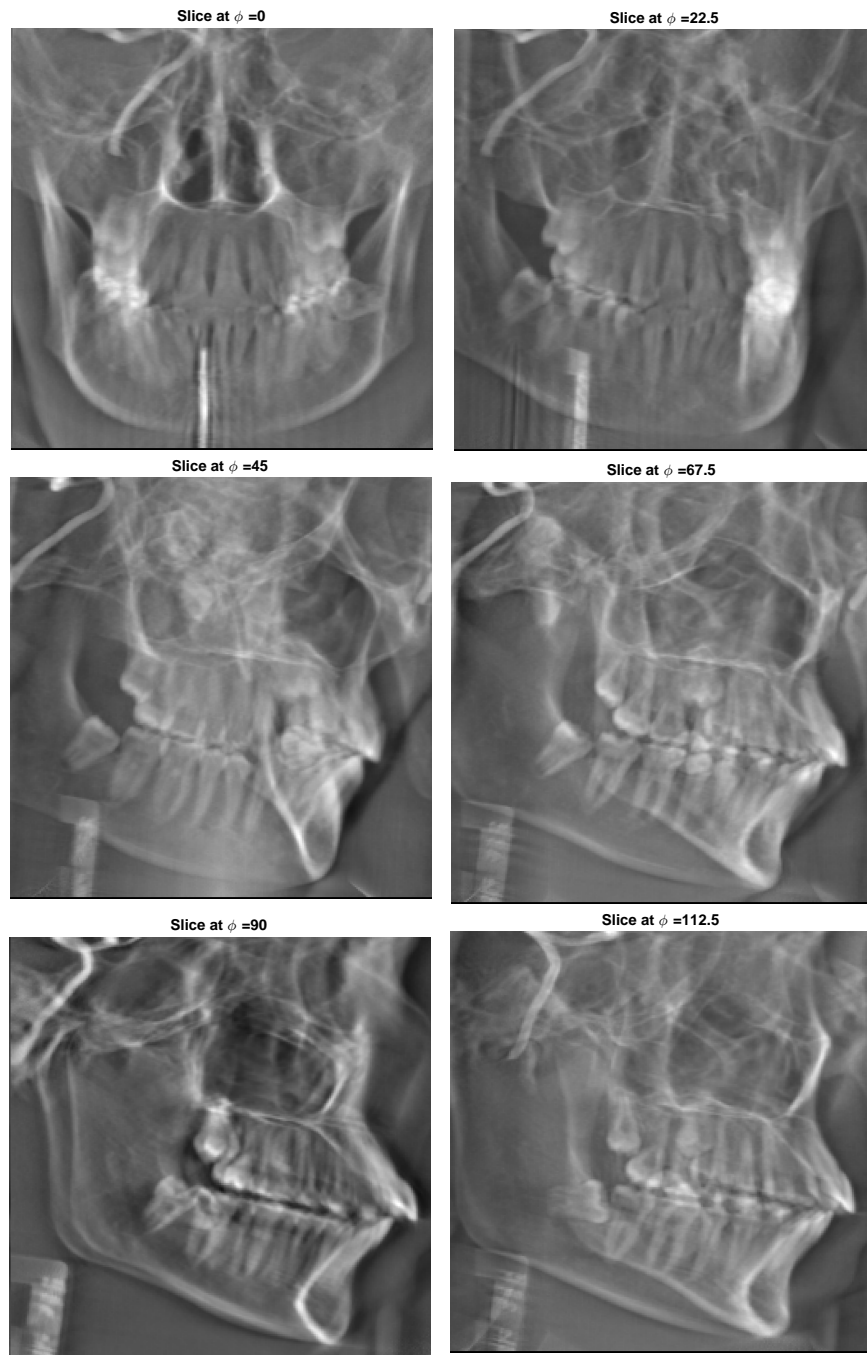


Figure 6.6: Full volumetric data analysis using the proposed FVR method on *skull* dataset, starting from top left at increasing angles of slices.

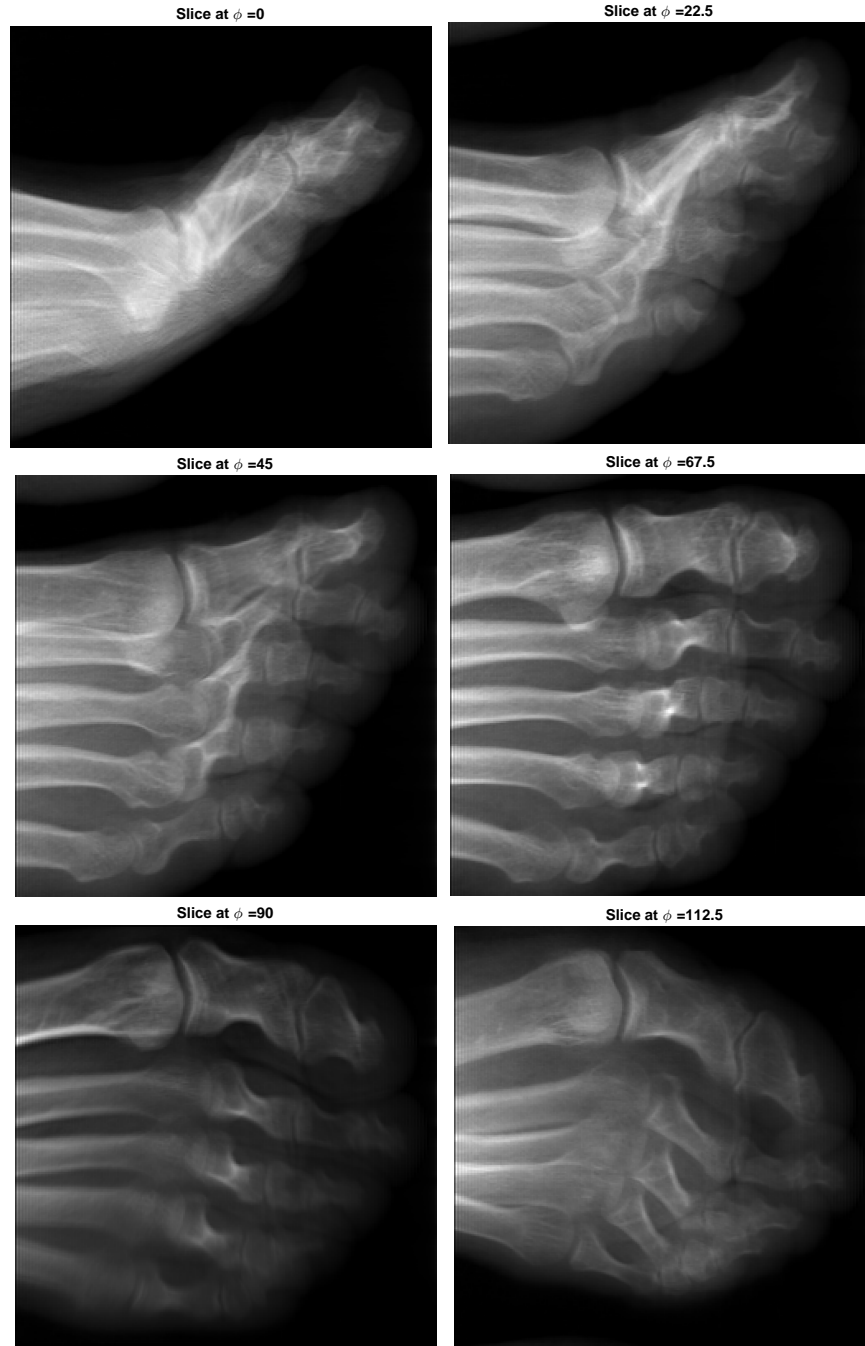


Figure 6.7: Full volumetric data analysis using the proposed FVR method on *Foot* dataset, starting from top left at increasing angles of slices.

In future we would like to explore this idea further and publish our finding in a journal paper for a biomedical issue. The proposed method relies on exact and fast computations which are capable of mitigating ghosting artifacts in FVR. The potential for applications in the medical field needs to be investigated further in terms of its clinical utility.

### 6.3 Application in Medical Imaging of Polar / Spherical DFT - Sparse-View X-ray CT Image Reconstruction

Sparse-view imaging is currently hot topic in medical imaging [207] and [208]. It is a promising scanning method which can reduce the radiation dose in X-ray computed tomography (CT). Reconstruction algorithm for sparse-view imaging system is of significant importance. The adoption of the spatial iterative algorithm for CT image reconstruction has a low operation efficiency and high computation requirement. In this section we would like to explore a novel Fourier-based iterative reconstruction technique that utilizes Polar / Spherical DFT that we proposed for sparse-view X-ray CT image reconstruction. We do have some preliminary numerical simulations and real data experiments that are performed on a parallel beam CT acquired data. The proposed method if clinically adopted appears to have extensive applications in X-ray CT imaging.

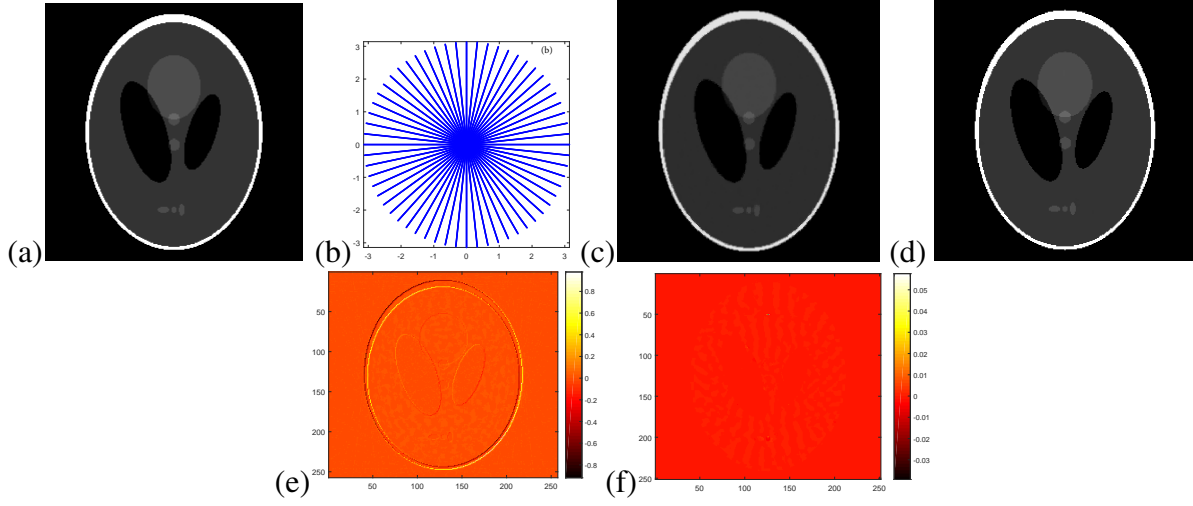


Figure 6.8: (a) Original Phantom Image, (b) True polar sampling grid, (c) Reconstruction using NUFFT (Nonuniform fast Fourier transform) + ADTVM (Alternating directions total variation minimization), (d) Reconstruction using Exact Polar FFT (proposed method) + ADTVM, (e) Error in the reconstruction using NUFFT + ADTVM, and (f) Error in the reconstruction using Exact Polar FFT + ADTVM. Both the simulations run with the same input data and same number of iterations = 200. The same parameters for the ADTVM is also same for both. Error norm comparisons,  $|I - \hat{I}_{NUFFT+ADTVM}|_2^2 = 4.7260$ , and  $|I - \hat{I}_{ExactPolar+ADTVM}|_2^2 = 0.0789$ . The proposed method is superior in terms of accuracy while comparable in terms of speed (due to similar computational complexity) with the NUFFT approach.

Fig 6.8 shows the result of a comparison with NUFFT based sparse view CT reconstruction method proposed in [208]. The idea here is to replace the NUFFT module which is an approximation of nonuniform Fourier transform with the fast and exact solution of polar FFT that we proposed and demonstrate some qualitative results. Although the initial result is promising in this direction, more work in terms of quantitative comparisons and applications needs to be investigated. We hope to do in future.

#### 6.4 Applications in Directional Wavelets using the proposed Polar / Spherical DFT

In this section we describe certain directional wavelets systems that are amenable to the direct application of the proposed Polar/Spherical DFT. This is an attempt to provides a unified treatment to all the classic systems proposed such as ridgelets, curvelets, etc. (translation of continuum ideas into digital implementation) where the computational procedure could be based on the discrete Fourier transform on Polar grid and Spherical grid with exact algorithms.

Wavelets and their geometric extensions (ridgelets, curvelets, framelets, countourlets, wave atoms,  $\dots$ ) are not only a useful mathematical tool arising in harmonic analysis and used to study function spaces, but they are also very efficient in image processing applications [209]. Despite the fact that wavelets have had a wide impact in image processing, they fail to efficiently represent objects with highly anisotropic elements such as lines or curvilinear structures (e. g. edges). The reason is that wavelets are non-geometrical and do not exploit the regularity of the edge curve [210]. The Ridgelet [211]-[216] and the Curvelet [217]-[219] transforms were developed as an answer to the weakness of the separable wavelet transform in sparsely representing what appears to be simple building atoms in an image, that is lines, curves and edges. Curvelets and ridgelets take the form of basis elements which exhibit high directional sensitivity and are highly anisotropic. These very recent geometric image representations are built upon ideas of multi-scale analysis and geometry. They have had an immense success in a wide range of image processing applications.

Ridgelet analysis may be constructed as wavelet analysis in the Radon domain. The rationale behind this is that the Radon transform translates singularities along lines into point singularities, for which the wavelet transform is known to provide a sparse representation [216], [210]. The Radon transform of a 2D object  $f(x_1, x_2)$  is the collection of line integrals indexed by  $(\theta, t) \in$

$[0, 2\pi) \times R$  given by,

$$Rf(\theta, t) = \int_{R^2} f(x_1, x_2) \delta(x_1 \cos \theta + x_2 \sin \theta - t) dx_1 dx_2 \quad (6.9)$$

where  $\delta$  is the Dirac distribution. Then the ridgelet transform is the application of a 1D wavelet transform to the slices of the Radon transform where the angular variable  $\theta$  is constant and  $t$  is varying. Thus the basic strategy for computing the continuous ridgelet transform is first to compute the Radon transform  $Rf(t, \theta)$  and second to apply a one-dimensional wavelet transform to the slices  $Rf(., \theta)$ . However transferring the ideas from continuous form to the discrete form is rather problematic since there exists no exact Radon definition for discrete data or a fast and exact algorithm for Fourier transform on true polar grid. Several digital ridgelet transforms (DRT) have been proposed [220], in broad terms, there are two viewpoints, one in the polar Frequency domain and one in the (closely related) Radon domain. One implementation is by a well known technique called gridding in tomography, a 2D-FFT or a variant called pseudo-polar (PP) FFT of a given image is computed, the result is used to evaluate the frequency values in a polar grid by means of interpolation [102]. Another implementation simply requires replacement of the true polar grid by a PP grid [102] or through the use of fast slant slack technique [111]. Another approximation scheme is through the use of non-uniform FFT (NUFFT) [103],[221]. In all cases the wavelet coefficients are directly calculated in the Fourier space. It has been shown in [212] that the DRT achieves near optimal  $M$ -term approximation- that is the non-linear approximation of  $f$  using the  $M$  highest ridgelet coefficients in magnitude- to smooth images with discontinuities along straight lines.

The Ridgelet packets library provides a large family of orthonormal bases for functions  $f(x_1, x_2)$  in  $L^2(dx_1 dx_2)$  which includes orthonormal ridgelets as well as bases deriving from tilings reminiscent from the theory of wavelets and the study of oscillatory Fourier integrals [215]. In the most



direct digital implementation of Ridgelet packets a recursive dyadic partition of the polar Fourier domain into a collection of rectangular tiles of various widths and lengths is required, however for practical considerations of digital data present in Cartesian coordinates a pseudo-polar FFT [111] was used. It was suggested that for a true geometric fidelity of digital ridgelet packets a computational procedure for true polar FFT is needed. In the companion article [216], it was pointed out that the DRT based on ideas similar to those used in this paper does not have the same qualities of coefficient decay that were available in theory for the ortho ridgelet transform of continuous functions. The slower decay of the ridgelet coefficients was attributed to the fact of an implicit ‘image mutilation’ that occurs at the heart of the Slant Stack algorithm [111] using pseudo-polar FFT. A more truthful digital representation of the continuum ideas could lead to improved coefficient decay and hence improved sparsity not only for ridgelet but for all ridgelet packet representations [215].

In image processing, edges are curved rather than straight lines and ridgelets are not able to efficiently represent such images. The curvelet transform was developed initially in the continuous domain [213] via multiscale filtering and then applying a block ridgelet transform [212] on each bandpass image partition to address the edge representation problem. Later, the authors proposed the second generation parabolic scaling curvelet transform [217] that was defined directly via frequency partitioning without using the ridgelet transform. Both curvelet constructions require a rotation operation and correspond to a 2-D frequency partition based on the polar coordinates [210]. This makes the curvelet construction simple in the continuous domain but causes the implementation for discrete images sampled on a rectangular grid to be very challenging [222]. In particular, approaching critical sampling seems difficult in such discretized constructions [222], since there exists no fast and exact computation of polar grid for discrete data.

Over the past years, various representation systems which sparsely approximate functions governed by anisotropic features such as edges in images have been proposed. Applications of curvelets include inverse problems [223], curvilinear integrals [223], curvelet based image denoising and im-

age enhancement [227] and [228], morphological component analysis and applications for texture separation and inpainting [220], astrophysics [229]. The curvelet transform is closely related to a continuous transform introduced by Hart Smith in his study of Fourier Integral Operators, [230]-[232]. A curvelet-like transform based on filterbanks called *contourlets* was proposed in [222]. Shearlets and wavelets with composite dilations that follow a curvelet-like approach within the framework of affine systems and which has more geometrically faithful discrete computational procedure is described in [233]. The digital shearlet transform also has parabolic scaling and it provides a very natural digitalization of the existing shearlet theory for continuous data [234], while the curvelets which are inherently based on operations like rotations which translate awkwardly into the digital realm. Alongside the theoretical development of contourlets, curvelets, and shearlets, algorithmic realizations of the associated transforms were provided. However, one of the most common shortcomings of these frameworks is the lack of providing a unified treatment of the continuum and digital world, i.e., allowing a digital theory to be a natural digitization of the continuum theory. In fact, as stated in [235], shearlet systems are the only systems so far which satisfy this property, yet still deliver optimally sparse approximations of cartoon-like images.

In the continuous-time curvelet definition the curvelet window smoothly extracts frequencies near the dyadic corona and the equispaced angles. Coronae and rotations are not especially adapted to Cartesian arrays. So for digital implementation these concepts are replaced by Cartesian equivalents “Cartesian coronae” based on concentric squares (instead of circles) and shears [218]. This circumvention was due to the fact that the pseudo-polar tiling of the frequency plane with trapezoids [111], was considered as a well-established data-friendly alternative to the ideal polar tiling. But with the recently proposed fast and exact polar FT solution in [183] we can use it to compute curvelet/ridgelet/ridgelet-packets or similar directional wavelets transform yielding the most faithful discretization of their respective continuous definition. So there is no need to replace the true polar grid with PP grid when it comes to digital algorithmic implementations and to use shearing

operations instead of true rotations. The potential for impact in this area by the use of true polar DFT needs to be further explored.

## 6.5 Application in Finite Rate of Innovation

Signals with finite rate of innovation (FRI) have a common feature that they possess a parametric representation with a finite number of degrees of freedom [237]. It was settled in [237] that by using an adequate sampling kernel and a sampling rate greater or equal to the rate of innovation, it is possible to reconstruct such signals uniquely. This is different from the classical sampling problem, where the signals are band-limited and hence can be reconstructed perfectly using interpolation. Instead of generic bandlimited signals, the problem here is concerned with sampling of classes of nonbandlimited signals. The main idea behind this technique is to exploit the priori known parameteric structure to reduce the sampling rate way beyond dictated by the conventional sampling theorem.

The reconstruction process for this scheme is based on the use of a locator or *annihilating filter*(AF) [238]. It is a standard high resolution spectral estimation tool widely used in spectral analysis [239] and error correction coding [240]. In particular this method is useful in signal processing context and has been applied to error detection and correction in bandlimited images [241] and [242]. In array signal processing it is used for shape reconstruction [244] and [243], or using a statistical framework with multiple sensors for retrieval of time of arrival, [245]-[247]. The potential applications for this technique is also in ultra wideband communications(UWB), electroencephalography (EEG) and optical coherence tomography (OCT) [248]. There is also a direct connection with FRI signals and compressed sensing, where these signals are seen as “sparse” in time domain. Due to the potential wide applicability of the signal with FRI property, the estimation problem has received growing attention in scientific community. A significant problem is that the

algorithms of AF are all quite complicated because they involve root finding of a polynomial, or eigenvalue decomposition, a highly nonlinear problem.

The motivation for this section is to make a connection of the FRI problem with the nonuniform frequency estimation solution that we proposed. In future it would be very interesting to analyze and compare our novel nonuniform technique based on adaptive notch filters with the annihilating filter and other conventional spectral estimation methods to identify the innovations in signals with FRI property. The problem of estimating locations of the Diracs (sparse signal) which is the basis of FRI signals using a frequency domain signal, simply becomes the problem of frequency estimation using discrete samples. We hope to explore the application of non-uniform frequency estimation proposed method in the FRI problems in future.

## 6.6 Training Deep Networks on Preprocessed Fourier Domain Information

The goal here is to exploit the signal information in such a way that it does the marriage of the world of signal processing and the power of deep learning. So we get the best of both world. For instance by a property of some transform the signal is first made rotationally invariant by means of preprocessing and then the neural networks or other learning algorithms are trained on that information of the transform domain, see [249] where radon transform is used for rotational invariance followed by support vector machine based learning. Similarly we could use the proposed 3D SPFT algorithms for achieving 3D accurate rotational invariance. Largely the deep neural networks literature has been built around directly processing the spatial domain, there are just a few handful of references that try to operate with frequency domain data or any transformed data in a different domain. It would be very interesting to explore further the implications/advantages or disadvantages of using spectral data and transforms to train deep neural network and compare it with conventionally trained networks (using directly spatial information) in terms of classification

accuracy, robustness to noise. We would like to explore this area further.

## CHAPTER 7: CONCLUSIONS

At the age of twenty three, Abraham Lincoln decided to run for the state legislature from Sangamon County, New Salem, Illinois. While it seemed next to impossible that a new settler who had just arrived in town with no family connections and little formal education could compete for the office, but it was Lincoln's drive that could attempt that. His desire to prove himself worthy, to be held in great regard, to win respect and veneration of his fellow citizens is revealed in the manner he chose to address his first speech. "I am young and unknown to many of you. I was born and have ever remained in the most humble walks of life. I have no wealthy or popular relations to recommend me. My case is thrown entirely upon independent voters of this county, and if elected they will have conferred a favor upon me, for which I shall be unremitting in my labors to compensate. But if good people in their wisdom shall see fit to keep me in the background, I have been too familiar with disappointments to be very much chagrined."

*Doris Kearns Goodwin, Team of Rivals, Pulitzer Prize, 2005.*

...Nehru devoted a whole chapter in his *Autobiography* to gently combating some of the basic aspects of Gandhian ideology. At the same time he fully appreciated the radical role that Gandhi had played and was playing in Indian society. Defending Gandhi against the left-wing critics, Nehru contented in an article written in January 1936 that ‘Gandhi has played a revolutionary role in India of the greatest importance because he knew how to make the most of the objective conditions and could reach the heart of the masses; while groups with a more advanced ideology functioned largely in the air.’ Moreover, Gandhi’s actions and teachings had ‘inevitably raised mass consciousness tremendously and made social issues vital.’ And his insistence on the raising of the masses at the cost, whenever necessary, of vested interests has given a strong orientation to the national movement in favor of the masses.

*Bipin Chandra, India’s Struggle for Independence, 1989.*

The presented dissertation deals with certain well known problems in signal processing literature such as phasor estimation, frequency estimation, computation of non-uniform Fourier transforms and robust registration using Fourier domain information. The main contribution of this thesis is the innovation in the analysis, synthesis, discretization of such well known problems. We provide novel solutions that have broad applications in audio signal processing, power systems, medical imaging and computer vision. The proposed solutions are shown to be competent with the existing state of the art methods. We summarize the contributions and conclusions of this thesis as follows:

- Chapter 2 proposes a novel adaptive signal processing algorithm. The proposed method is simple but powerful in terms of speed and accuracy when compared to the existing techniques. It is capable of dealing with inter-harmonics and harmonics unlike some existing

state of the art FFT based techniques.

- Chapter 3 advances a unique adaptive notch filter based solution for non-uniformly sampled data. The solution is shown to be very competitive and effective even in the presence of high noise and close frequencies.
- Chapter 4 contends a fast and exact solution for computing discrete Polar FT and Spherical Polar FT that has numerous applications in signal/image processing. The proposed solution is unique in the literature because it requires no user selection parameter since it does not do any interpolation. We show that the solution we proposed is fast and very accurate compared to the state of the art NFFT (non-uniform FFT) optimized algorithm.
- Chapter 5 comes up with a new and effective method of estimating up to 7 degrees of freedom (DOF), i.e. similarity transformation, for 3D volume registration in the spherical polar Fourier domain. We show that the proposed multi-radii spherical harmonics representation outperforms the existing single-radius averaged spherical harmonic representation in terms of accuracy, ability to handle large rotation angles, robustness noise, holes, missing data, and partial overlap. We compare the proposed technique with the state-of-the-art methods, and in particular show that unlike existing methods that are limited to angles smaller than 30 degrees (due to spectral averaging, interpolation and re-projection errors), we are able to accurately estimate arbitrarily large angles on all three axes simultaneously (a direct result of closed-form solution and multi-radii representation).
- Chapter 6 shows a few potential applications / extension of the work previously presented in this dissertation.



## LIST OF REFERENCES

- [1] Gilbert Strang, "Signal processing for everyone," In Computational Mathematics Driven by Industrial Problems, pp. 366-412. Springer Berlin Heidelberg, 2000.
- [2] Stephane Mallat, A wavelet tour of signal processing: the sparse way. Academic press, 2008.
- [3] P. Bonanomi, Phase angle measurements with synchronized clocks, IEEE Trans. on Power Apparatus and Systems, vol. PAS-100, no. 12, pp. 5036-5043, December 1981.
- [4] A. G. Phadke, Synchronized phasor measurements in power systems, IEEE Computer Applications in Power, vol. 6, no. 2, pp. 10-15, April 1993.
- [5] R. O. Burnett, Jr., M. M. Butts, P. S. Sterlina, Power system applications for phasor measurement Units, IEEE Computer Applications in Power, vol. 7, no. 1, pp. 8-13, Jan 1994.
- [6] I. Kamwa and R. Grondin, Fast adaptive schemes for tracking voltage phasor and local frequency in power transmission and distribution systems , in Proc. IEEE Power Eng. Soc. Transm. Distrib. Conf. , Dallas, TX, 1991, pp. 930-936.
- [7] A. A. Girgis and W. L. Peterson, "Adaptive estimation of power system frequency deviation and its rate of change for calculating sudden power system overloads," *IEEE Trans. Power Del.* ,vol.5,no 2, pp. 585-597, Apr 1990.
- [8] K. M. El-Naggar and H. K. M. Youssef, "A genetic based algorithm for frequency relaying applications," *Elect. Power Syst. Res.*, vol. 55, no. 3, pp. 173-178, 2000.
- [9] L. L. Lai and W. L. Chan, "Real time frequency and harmonic evaluation using artificial networks," *IEEE Trans. Power Del.* ,vol. 14, no 1, pp. 52-57,Jan. 1999.

- [10] G. Benmouyal, "An adaptive sampling interval generator for digital relaying," *IEEE Trans. Power Del.*, vol 4, no. 3, pp. 1602-1609, July. 1989.
- [11] D. Hart, D. Novosel, Y. Hu, B. Simth and M. Egolf, "A new frequency tracking and phasor estimation algorithm for generator protection," *IEEE Trans. Power Del.*, vol. 12, no. 3, pp. 1064-1073, Jul. 1997.
- [12] T. S. Sidhu and M. S. Sachdev, "An iterative technique for fast and accurate measurement of power system frequency," *IEEE Trans. Power Del.*, vol 13, no. 1, pp. 109-115, Jan. 1998.
- [13] P. J. Moore, J. H. Allmeling, and A. T. Johns, "Frequency relaying based on instantaneous frequency measurement," *IEEE Trans. Power Del.*, vol. 11, no. 4, pp. 1737-1742, Oct. 1996.
- [14] M. D. Kusljevic, "Simultaneous frequency and harmonic magnitude estimation using decoupled modules and multirate sampling," *IEEE Trans. Instrum Mes.*, vol. 59, no. 4, pp. 954-962, Apr. 2010.
- [15] M. Wang and Y. Sun, "A practical method to improve phasor and power measurement accuracy of DFT algorithm," *IEEE Trans. Instrum Mes.*, vol. 21, no. 3, pp. 1054-1062, Jul. 2006.
- [16] A. Lopez, J. C. Montano, M. Castilla, J. Gutierrez, M. D. Borras, and J. C. Bravo, "Power system frequency measurement under non-stationary situations," *IEEE Trans. Power Del.*, vol. 23, no. 2, pp. 562-567, Apr. 2008.
- [17] R. K. Mai, Z. Y. He, L. Fu, B. Kirby, and Z. Q. Bo, "A Dynamic Synchrophasor Estimation Algorithm for Online Application," *IEEE Trans. Instrum Mes.*, vol. 25, no. 2, pp. 570-578, April. 2010.
- [18] J. Ren and M. Kezuovic, "Real-Time Power system frequency and phasors estimation using recursive wavelet transform," *IEEE Trans. Power Del.*, vol. 26, no 3, pp. 1392-1402, Jul. 2011.

- [19] Iman Sadinezhad and Vassilios G, "Monitoring Voltage Disturbances Based on LES Algorithm, Wavelet Transform and Kaman Filter," *35th IEEE Industrial Electronics, IECON09*., pp. 1961-1966, Nov 2009.
- [20] A. G. Phadke, T. Hlibka, and M. Kezuovic, "A digital computer system for EHV substation: Analysis and field tests," *IEEE Trans. Power App. Syst.* ,vol. PAS-95, no. 1, pp. 291-301,Jan/Feb. 1976.
- [21] N. T. Stringer, "The effect of DC offset on current operated relays," *IEEE Trans. Ind. Appl.*, vol. 34, pp. 30-34, Jan/Feb. 1998.
- [22] J. C. Gu and S. L. Yu, "Removal of DC offset in current and voltage signals using a novel Fourier filter algorithm," *IEEE Trans. Power Del.* , vol. 15,no. 1,pp. 73-79,Jan. 2000.
- [23] J. Z. Yang and C. W. Liu, "Complete elimination of DC offset in current signal for relaying applications," *Proc. IEEE Power Eng. Soc. Winter Meeting* , vol. 3,no. 1,pp. 1933-1938,Jan. 2000.
- [24] T. S. Sidhu, X. Zhang, F. Albasri and M. S. Sachdev, "Discrete-Fourier-transform-based technique for removal of decaying DC offset from phasor estimates," *Proc. Inst. Elect. Eng. Gen.,Tansm. Distrib.* , vol. 150,no. 6,pp. 745-752,Nov. 2003.
- [25] S. H. Kang, D. G. Lee, S. R. Nam, P. A. Crossley and Y. C. Kang, "Fourier transform-based modified phasor estimation method immune to the effect of DC offsets," *IEEE Trans. Power Del.* ,vol. 24, no. 3, pp. 1104-1111,Jul. 2009.
- [26] S. R. Nam, J. Y. Park, S. H. Kang, and M. Kezuovic, "Phasor estimation in presence of DC offset and ct saturation," *IEEE Trans. Power Del.*,vol. 24, no 4, pp. 1842-1849,Oct. 2009.

- [27] Y. Gou and M. Kezuovic, "Simplified algorithms for removal of the effect of exponentially decaying DC-offset on the Fourier algorithms," *IEEE Trans. Power Del.*, vol. 18, no. 3, pp. 711-717, Jul. 2003.
- [28] C. S. Yu, "A discrete Fourier transform-based adaptive mimic phasor estimator for distance relaying applications," *IEEE Trans. Power Del.*, vol. 21, no. 4, pp. 1836-1846, Oct. 2006.
- [29] O. Chaari, M. Menuier and F. Brouaye, "Wavelets: A new tool for the resonant grounded power distribution system relaying applications," *IEEE Trans. Power Del.*, vol. 11, no. 3, pp. 1301-1308, Jul. 1996.
- [30] C. Zhang, Y. Huang, X. Ma, W. Lu and G. Wang, "A new approach to detect transformer inrush current by applying wavelet transform," in *Proc. POWERCON*, vol. 2, pp. 1040-1044, Aug. 1998.
- [31] X. N. Lin and H. F. Liu, "A fast recursive wavelet based boundary protection scheme," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, vol. 1, pp. 722-727, Jun. 2005.
- [32] Ying Liu, Raghuram R, Matthew T and Wasfy B. Mikhael, "Conjugate Gradient based complex Block LMS employing Time-varying optimally derived step sizes", *52nd IEEE International Midwest Symposium on Circuits and Systems*, pp. 590-593, 2009
- [33] B. Hassibi and T. Kailath, Mixed least-mean-squares/H-infinity-optimal adaptive filtering, *Proceedings of the 30th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov 1996
- [34] M. J. D. Powell, "Restart procedures for the conjugate gradient method" in *Mathematical programming*, Vol. 12, no. 1, pp. 241-254, 1977.
- [35] Grippo, Luigi, and Stefano Lucidi. "A globally convergent version of the Polak-Ribiere conjugate gradient method." *Mathematical Programming* 78.3 (1997): 375-391.

- [36] Simon Haykin, *Neural Networks, A Comprehensive Foundation*, Prentice Hall, New Jersey, 1996.
- [37] Math H.J. Bollen and Irene .Y.H. Gu, *Signal Processing of Power Quality Disturbances* , IEEE, Wiley, 2006.
- [38] *IEEE Standard for Synchrophasors for Power Systems* ,IEEE Std. C37.118-2005,Mar. 2006.
- [39] S. Alam Abbas, “A new fast algorithm to estimate real time phasors using adaptive signal processing”, *IEEE Trans. Power Del.*, vol. 28, pp. 807–815, 2013.
- [40] P. A. Regalia, “An improved lattice-based adaptive IIR notch filter,” *IEEE Trans. Signal Process.*, vol. 39, pp. 2124–2128, 1991.
- [41] C. Fuller, and A. V. Flotow, “Active control of sound and vibration,” *IEEE Contr. Syst. Mag.*, vol. 15, pp. 9–19, 1996.
- [42] A. P. Klapuri, “Multiple fundamental frequency estimation based on harmonicity and spectral smoothness,” *IEEE Trans. Speech Audio Process.*, vol. 11, pp. 804–816, 2003.
- [43] O. Simeone, Y. Bar-Ness, and U. Spagnolini, “Pilot-based channel estimation for OFDM systems by tracking the delay-subspace,” *IEEE Trans. Wireless Communications*, vol. 3, pp. 315–325, 2004.
- [44] S. F. Hsieh, K. J. R. Liu, and K. Yao, “Estimation of multiple sinusoidal frequencies using truncated least squares methods,” *IEEE Trans. Signal Process.*, vol. 41, pp. 990–994, 1993.
- [45] R. Badeau, B. David, and G. Richard, “High-resolution spectral analysis of mixtures of complex exponentials modulated by polynomials,” *IEEE Trans. Signal Process.*, vol. 54, pp. 1341–1350, 2006.

- [46] Q. Zhang, and L. J. Brown, "Noise Analysis of an Algorithm for Uncertain Frequency Identification," *IEEE Trans. Autom. Control*, vol. 51, pp. 103–110, 2006.
- [47] K. M. Ghartemani, B. T. Ooi, and A. Bakhshai, "Application of enhanced phase-locked loop system to the computation of synchrophasors," *IEEE Trans. Power Del.*, vol. 26, pp. 22–32, 2011.
- [48] A. Dutt, and V. Rokhlin, "Fast Fourier transforms for nonequispaced data," *SIAM Journal on Scientific Computing*, vol. 14, pp. 1368–1393, 1993.
- [49] P. Stoica and R. L. Moses, "*Introduction to Spectral Analysis*," Englewood Cliffs, NJ: Prentice-Hall, 1997.
- [50] P. Stoica, E. G. Larsson, and J. Li, "Adaptive filterbank approach to restoration and spectral analysis of gapped data," *Astron. J.*, vol. 120, pp. 2163–2173, 2000.
- [51] F. Marvasti, "*Nonuniform Sampling: Theory and Practice*", Vol. 1, Springer Science & Business Media, 2001.
- [52] F. Eng, "Non-Uniform Sampling in Statistical Signal Processing," PhD Thesis, 2007.
- [53] M. W. Maciejewski, H. Z. Qui, I. Rujan, M. Mobli, and J. C. Hoch, "Nonuniform sampling and spectral aliasing," *J. Magnetic Resonance*, vol. 1, pp. 88–93, 2009.
- [54] R. L. Cook, "Stochastic sampling in computer graphics," *ACM Transactions on Graphics (TOG)*, Vol, 5.1, pp. 51–72, 1986.
- [55] S. Trittler, and F. A. Hamprecht. "Near optimum sampling design and an efficient algorithm for single tone frequency estimation," *Digital Signal Processing*, vol. 19, pp. 628–639, 2009.

- [56] A. Holland, and M. Aboy, “A novel recursive Fourier transform for nonuniform sampled signals: application to heart rate variability spectrum estimation,” *Medical & Biological Engineering & Computing*, vol. 47, pp. 697–707, 2009.
- [57] J. A. Gansman, J. V. Krogmeier, and M. P. Fitz, “Single frequency estimation with non-uniform sampling,” *IEEE Asilomar Conference Signals, Systems and Computers*, vol. 1, pp. 399–403, 1996.
- [58] Y. C. Eldar, and A. V. Oppenheim, “Filterbank reconstruction of bandlimited signals from nonuniform and generalized samples,” *IEEE Trans. Signal Process.*, vol. 48, pp. 2864–2875, 2000.
- [59] H. Johansson, and P. Lowenborg, “Reconstruction of nonuniformly sampled bandlimited signals by means of digital fractional delay filters,” *IEEE Trans. Signal Process.*, vol. 50, pp. 2757–2767, 2002.
- [60] R. S. Prendergast, C. L. Bernard, and P. J. Hurst, “Reconstruction of band-limited periodic nonuniformly sampled signals through multirate filter banks,” *IEEE Trans. Circuits and Systems*, vol. 51, pp. 1612–1622, 2004.
- [61] A. Tarczynski, N. Allay, “Spectral analysis of randomly sampled signals: Suppression of aliasing and sampler jitter,” *IEEE Trans. Signal Process.*, vol. 52, no. 12, 2004.
- [62] D. Qu, and A. Tarczynski, “A novel spectral estimation method by using periodic nonuniform sampling,” *IEEE Asilomar Conference Signals, Systems and Computers*, pp. 1134–1138, 2007.
- [63] M. Unser, “Sampling-50 years after Shannon,” *Proc. IEEE*, vol. 88, pp. 569–587, 2000.
- [64] A. Aldroubi, and K. Gröchenig, “Nonuniform sampling and reconstruction in shift-invariant spaces,” *SIAM Review*, vol. 43, pp. 585–620, 2001.

- [65] Q. Sun, "Non-uniform average sampling and reconstruction of signals with finite rate of innovation," *SIAM J. Math. Anal.*, vol. 38, pp. 1389–1422, 2006.
- [66] P. Handel, and A. Nehorai, "Tracking analysis of an adaptive notch filter with constrained poles and zeros," *IEEE Transactions on Signal Processing*, vol. 42.2, pp. 281–291, 1994.
- [67] M. V. Dragosevic, and S. S. Stankovic, "An adaptive notch filter with improved tracking properties", *IEEE Transactions on Signal Processing*, vol. 43, pp. 2068–2078, 1995.
- [68] L. Hsu, R. Ortega, and G. Damm, "A globally convergent frequency estimator," *IEEE Trans. Autom. Control*, vol. 44, pp. 698–713, 1999.
- [69] R. Marino and P. Tomei, "Global Estimation of  $n$  Unknown Frequencies," *IEEE Trans. Autom. Control*, vol. 47, pp. 1324–1328, 2002.
- [70] X. Xia, "Global Frequency Estimation Using Adaptive Identifiers," *IEEE Trans. Autom. Control*, vol. 47, pp. 1188–1193, 2002.
- [71] M. Mojiri, and A. R. Bakhshai, "An adaptive notch filter for frequency estimation of a periodic signal," *IEEE Trans. Autom. Control*, vol. 49, pp. 314–318, 2004.
- [72] M. Hou, "Amplitude and frequency estimator of a sinusoid," *IEEE Trans. Autom. Control*, vol. 50, pp. 855–858, 2005.
- [73] M. Mojiri and A. R. Bakhshai, "Stability analysis of periodic orbit of an adaptive notch filter for frequency estimation of a periodic signal," *Automatica*, vol. 43, no. 3, pp. 450–455, 2007.
- [74] M. Mojiri, M. K. Ghartemani, and A. Bakhshai, "Estimation of power system frequency using an adaptive notch filter," *IEEE Trans. Instrum. Meas.*, vol. 56, pp. 2470–2477, 2007.
- [75] P. Regalia, *Adaptive IIR Filtering in Signal Processing and Control*, CRC Press, 1994.



- [76] D. Victor and S. Torres, "Multiple fully adaptive notch filter design based on allpass sections," *IEEE Trans. Signal Process.*, vol. 48, pp. 550–552, 2000.
- [77] Y. C. Lim, Y. X. Zou, and N. Zheng, "A piloted adaptive notch filter," *IEEE Trans. Signal Process.*, vol. 53, pp. 1310–1323, 2005.
- [78] J. E. Cousseau, S. Werner, and P. D. Donate, "Factorized all-pass based IIR adaptive notch filters," *IEEE Trans. Signal Process.*, vol. 55, pp. 5225–5236, 2007.
- [79] L. Tan, and J. Jiang, "Novel adaptive IIR notch filter for frequency estimation and tracking," *IEEE Signal Processing Mag.*, vol. 26, pp. 186–189, 2009.
- [80] P. A. Regalia, "A complex adaptive notch filter," *IEEE Signal Process. Lett.*, vol. 17, pp. 937–940, 2010.
- [81] M. Bodson, and S. C. Douglas, "Adaptive algorithms for the rejection of sinusoidal disturbances with unknown frequency," *Automatica* vol. 33, pp. 2213–2221, 1997.
- [82] G. Damm, L. Hsu, and R. Ortega, "The effect of noise on a global frequency estimator," in *Proc. MTNS*, Padova, Italy, 1998.
- [83] M. Hou, "Parameter Identification of Sinusoids," *IEEE Trans. Autom. Control*, vol. 50, pp. 855–858, 2013.
- [84] G. P. Rao, and H. Unbehauen, "Identification of continuous-time systems," *IEE Proceedings In Control Theory and Applications*, vol. 153, no. 2, pp. 185–220. IET, 2006.
- [85] H. Garnier and L. Wang, *Identification of continuous-time models from sampled data*, London, Springer, 2008.
- [86] C. C. Tseng, "Design of fractional order digital FIR differentiators," *IEEE Signal Processing Letters*, no. 3, pp. 77–79, 2001.

- [87] A. Papoulis, *Signal Analysis*, McGraw-Hill Companies, May 1977.
- [88] K. Cheung, and R. J. Marks, "Ill-posed sampling theorems," *IEEE Trans. on Circuits and Systems*, 32, no. 5 (1985): 481-484.
- [89] M. Shahram, and P. Milanfar, "On the resolvability of sinusoids with nearby frequencies in the presence of noise," *IEEE Trans. Signal Process.*, vol. 53, pp. 2579–2588, 2005.
- [90] L. C. Evans, *Partial Differential Equations*, American Mathematical Society, 1998.
- [91] S. Lynch, *Dynamical Systems with Applications using MATLAB*, Springer, 2004.
- [92] Khalil, Hassan K, *Nonlinear systems*, Vol. 3, Upper Saddle River, Prentice hall, 2002.
- [93] R. L. Burden and J. D. Faires, *Numerical Analysis*, 9th edition, Brooks-Cole, 2010.
- [94] H. Robert, and M. E. Jernigan, "Self-adjusting beat detection and prediction in music," *IEEE International Conference Proceedings on Acoustics, Speech, and Signal Processing (ICASSP'04)*, vol. 4, 2004.
- [95] S. M. Kay, and S. L. Marple Jr, "Spectrum analysis - a modern perspective," *IEEE Proceedings*, vol. 69, pp. 1380–1419, 1981.
- [96] S. M. Kay, "*Fundamentals of Statistical Signal Processing, Volume III: Practical Algorithm Development*," Vol. 3, Pearson Education, 2013.
- [97] E. G. Larsson, P. Stoica, and J. Li, "Spectral estimation via adaptive filterbank methods: a unified analysis and a new algorithm," *Signal processing*, vol. 82, pp. 1991–2001, 2002.
- [98] M. Mojiri, M. K. Ghartemani, and A. Bakhshai, "Time-domain signal analysis using adaptive notch filter," *IEEE Trans. Signal Processing*, vol. 55, pp. 85-93, 2007.
- [99] R. M. Gray and J. W. Goodman, *Fourier Transforms*, Kluwer, 1995.

- [100] R. N. Bracewell, *The Fourier Transform and its Applications*, McGraw Hill, 1986.
- [101] B. Osgood, *The Fourier Transform and its Applications*, Lecture Notes, Stanford University.
- [102] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Elad, and M. Israeli, “Fast and accurate polar Fourier transform,” *Applied and Computational Harmonic Analysis*, vol. 21, no. 2, pp. 145-167, 2006.
- [103] M. Fenn, S. Kunis, and D. Potts, “On the computation of the polar FFT,” *Applied and Computational Harmonic Analysis*, vol. 22, no. 2, pp. 257-263, 2007.
- [104] W. Lawton, “A new polar Fourier transform for computer-aided tomography and spotlight synthetic aperture radar,” *IEEE Trans. Acoustics Speech and Signal Processing*, vol. 36, pp. 931-933, 1988.
- [105] W. L. Briggs, *The DFT: an Owners’ Manual for the Discrete Fourier Transform*, SIAM, 1995.
- [106] F. Natterer, *The Mathematics of Computerized Tomography*, vol. 32, SIAM, 1986.
- [107] A. H. Delaney and Y. Bresler, “A fast and accurate Fourier algorithm for iterative parallel-beam tomography,” *IEEE Trans. Image Processing*, vol. 5, no. 5, pp. 740-753, 1996.
- [108] D. Gottleib, B. Gustafsson, and P. Forssén, “On the direct Fourier method for computer tomography,” *IEEE Trans. Medical Imaging*, vol. 19, no. 3, pp. 223-232, 2000.
- [109] S. Basu, and Y. Bresler, “ $O(N^2 \log_2 N)$  filtered backprojection reconstruction algorithm for tomography,” *IEEE Trans. Image Processing*, vol. 9, no. 10, pp. 1760-1773, 2000.
- [110] S. Matej, J. Fessler, and I. G. Kazantsev, “Iterative tomographic image reconstruction using Fourier-based forward and back-projectors,” *IEEE Trans. Medical Imaging*, vol. 23, no. 4, pp. 401-412, 2004.

- [111] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Israeli, and J. Walden, *Fast Slant Stack: A notion of Radon transform for data in a Cartesian grid which is rapidly computible, algebraically exact, geometrically faithful and invertible*, Department of Statistics, Stanford University, 2001.
- [112] A. Averbuch and Y. Shkolnisky, “3D Fourier based discrete Radon transform,” *Applied and Computational Harmonic Analysis*, vol. 15, no. 1, pp. 33-69, 2003.
- [113] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Israeli, and Y. Shkolnisky, “A framework for discrete integral transformations I-The pseudopolar Fourier transform,” *SIAM Journal on Scientific Computing*, vol. 30, no. 2, pp. 764-784, 2008.
- [114] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Israeli, Y. Shkolnisky, and I. Sedelnikov, “A framework for discrete integral transformations II-the 2D discrete Radon transform,” *SIAM Journal on Scientific Computing*, vol. 30, no. 2, pp. 785-803, 2008.
- [115] A. Averbuch, G. Shabat, and Y. Shkolnisky, “Direct inversion of the three-dimensional pseudo-polar Fourier transform,” *SIAM Journal on Scientific Computing*, vol. 38, no. 2, pp. 1100-1120, 2016.
- [116] S. M. Hashemi, S. Beheshti, P. R. Gill, N. S. Paul, and R. S. C. Cobbold, “Accelerated Compressed Sensing Based CT Image Reconstruction,” *Computational and Mathematical Methods in Medicine*, vol. 2015, article id-161797, pp. 16, doi:10.1155/2015/161797, 2015.
- [117] J. Miao, F. Frster, and O. Levi, “Equally sloped tomography with oversampling reconstruction,” *Physical Review B*, vol. 72, no. 5, 052103, 2005.
- [118] G. Beylkin, “On the fast Fourier transform of functions with singularities,” *Applied and Computational Harmonic Analysis*, vol. 2, no. 4, pp. 363-381, 1995.

- [119] A. Dutt, and V. Rokhlin, “Fast Fourier transforms for nonequispaced data,” *SIAM Journal on Scientific computing*, vol. 14, no. 6, pp. 1368-1393, 1993.
- [120] A. Dutt, and V. Rokhlin, “Fast Fourier transforms for nonequispaced data II,” *Applied and Computational Harmonic Analysis*, vol. 2, no. 1, pp. 85-100, 1995.
- [121] D. Potts, G. Steidl, and M. Tasche, “Fast Fourier transforms for nonequispaced data: A tutorial,” *Modern Sampling Theory*, Birkhäuser Boston, pp. 247-270, 2001.
- [122] J. Fessler, and B. P. Sutton, “Nonuniform fast Fourier transforms using min-max interpolation,” *IEEE Trans. Signal Processing*, vol. 51, no. 2, pp. 560-574, 2003.
- [123] L. Greengard, and J. Y. Lee, “Accelerating the nonuniform fast Fourier transform,” *SIAM Review*, vol. 46, no. 3, pp. 443-454, 2004.
- [124] M. Jacob, “Optimized least-square nonuniform fast Fourier transform,” *IEEE Trans. Signal Processing*, vol. 57, no. 6, pp. 2165-2177, 2009.
- [125] Q. Sun, “Non-uniform average sampling and reconstruction of signals with finite rate of innovation,” *SIAM Journal of Mathematical Analysis*, vol. 38, no. 5, pp. 1389–1422, 2006.
- [126] S. A. Abbas, Q. Sun, and H. Foroosh, “Frequency estimation of sinusoids from nonuniform samples”, *Signal Processing*, 2016.
- [127] B. S. Reddy, and B. N. Chatterji, “An FFT-based technique for translation, rotation, and scale-invariant image registration,” *IEEE Trans. Image Processing*, vol. 5, no. 8, pp. 1266-1271, 1996.
- [128] H. Foroosh, J. B. Zerubia, and M. Berthod, “Extension of phase correlation to subpixel registration,” *IEEE Trans. Image Processing*, vol. 11, no. 3, pp. 188-200, 2002.

- [129] W. Pan, K. Qin, and Y. Chen, "An adaptable-multilayer fractional Fourier transform approach for image registration." *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 400-414, 2009.
- [130] F. Nestler, "Automated parameter tuning based on RMS errors for nonequispaced FFTs." *Advances in Computational Mathematics*, DOI 10.1007/s10444-015-9446-8, 2016.
- [131] N. Chou, J. Izatt, and S. Farsiu, "Generalized pseudo-polar Fourier grids and applications in registering ophthalmic optical coherence tomography images," *IEEE Signals, Systems and Computers Record of the Forty-Third Asilomar Conference*, pp. 807-811, 2009.
- [132] D. Potts and G. Steidl, "A new linogram algorithm for computerized tomography," *IMA J. Numer. Anal.*, vol. 21, pp. 769-782, 2001.
- [133] S. A. Abbas, and H. Foroosh, "Robust 3D Registration using Spherical Polar Fourier Transform and Spherical Harmonics," *International Journal of Computer Vision*, Submitted.
- [134] D. H. Bailey, and P. N. Swarztrauber, "The fractional Fourier transform and applications," *SIAM review*, vol. 33, no. 3, pp. 389-404, 1991.
- [135] J. Jan, *Digital Signal Filtering, Analysis and Restoration*, INSPEC Telecommunications Series, 2000.
- [136] S. W. Smith, *Digital Signal Processing*, Newnes, 2003.
- [137] K. Pavel, and D. Svoboda, *Algorithms for Efficient Computation of Convolution*, INTECH Open Access Publisher, 2013.
- [138] K. Pavel, and D. Svoboda, "Convolution of large 3D images on GPU and its decomposition," *EURASIP Journal on Advances in Signal Processing*, no. 1, pp. 1-12, 2011.

- [139] P. Schaller, and G. Temnov, “Efficient and precise computation of convolutions: applying fft to heavy tailed distributions,” *Computational Methods in Applied Mathematics*, vol. 8, no. 2, pp. 187-200, 2008.
- [140] A. Jerri, *Integral and Discrete Transforms with Applications and Error Analysis*, CRC, 1992.
- [141] J. C. Bowman, and M. Roberts, “Efficient dealiased convolutions without padding,” *SIAM Journal on Scientific Computing*, vol. 33, no. 1, pp. 386-406, 2011.
- [142] S. Kunis, and D. Potts, “NFFT, Software package, C subroutine library,” 2002. Retrieved from: <https://github.com/NFFT/nfft>.
- [143] M. Lustig, J. Tsaig, J. H. Lee, and D. L. Donoho, “Fast spiral Fourier transform for iterative MR image reconstruction,” In *IEEE International Symposium Biomedical Imaging: Nano to Macro*, pp. 784-787, 2004.
- [144] A. C. Rencher, and G. B. Schaalje, *Linear Models in Statistics*, John Wiley Sons, 2008.
- [145] R. F. Bass, and K. Grchenig, “Random sampling of multivariate trigonometric polynomials,” *SIAM Journal on Mathematical Analysis*, vol. 36, no. 3, pp. 773-795, 2005.
- [146] T. Knopp, S. Kunis, and D. Potts, “A note on the iterative MRI reconstruction from nonuniform k-space data,” *International Journal of Biomedical Imaging*, vol. 2007, no. 24727, 2017.
- [147] P. C. Hansen, J. G. Nagy, and D. P. O’leary, *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM, vol. 3, 2006.
- [148] T. Dray, and C. A. Manogue, *Conventions for Spherical Coordinates*, 2002.

- [149] O. Ayala, and L. P. Wang, "Parallel implementation and scalability analysis of 3D fast Fourier transform using 2D domain decomposition," *Parallel Computing*, vol. 39, no. 1, pp. 58-77, 2013.
- [150] B. Lichtenbelt, *Fourier volume rendering*, Hewlett-Packard Laboratories, Technical Publications Department, 1995.
- [151] P. Yalamanchili, U. Arshad, Z. Mohammed, P. Garigipati, P. Entschew, B. Kloppenborg, J. Malcolm, and J. Melonakos, *ArrayFire - A high performance software library for parallel computing with an easy-to-use API*, Atlanta: AccelerEyes, 2016. Retrieved from <https://github.com/arrayfire/arrayfire>.
- [152] M. Puschel, and J. M. F. Moura, "Algebraic signal processing theory: Foundation and 1-D time," *IEEE Trans. Signal Processing*, vol. 56, no. 8, pp. 3572-3585, 2008.
- [153] Gholipour, Ali, Nasser Kehtarnavaz, Richard Briggs, Michael Devous, and Kaundinya Gopinath, "Brain functional localization: a survey of image registration techniques." *Medical Imaging, IEEE Transactions on* 26, no. 4 (2007): 427-451.
- [154] Pluim, Josien PW, JB Antoine Maintz, and Max A. Viergever. "Mutual-information-based registration of medical images: a survey." *IEEE transactions on medical imaging* 22, no. 8 (2003): 986-1004. .
- [155] Kim, Jeongtae, and Jeffrey A. Fessler. "Intensity-based image registration using robust correlation coefficients." *IEEE transactions on medical imaging* 23.11 (2004): 1430-1444.
- [156] Markelj, Primoz, et al, "A review of 3D/2D registration methods for image-guided interventions." *Medical image analysis* 16.3 (2012): 642-661.
- [157] Sotiras, Aristeidis, Christos Davatzikos, and Nikos Paragios, "Deformable medical image registration: A survey." *Medical Imaging, IEEE Transactions on* 32, no. 7 (2013): 1153-1190.



- [158] Sun, Bo, Weiwei Kong, Liwei Zhang, and Jianwei Zhang, "Fourier analysis techniques applied in data registration: A survey." In *Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*, 2014 International Conference on, pp. 1-5. IEEE, 2014.
- [159] Wyawahare, Medha V., Pradeep M. Patil, and Hemant K. Abhyankar, "Image registration techniques: an overview." *International Journal of Signal Processing, Image Processing and Pattern Recognition* 2, no. 3 (2009): 11-28.
- [160] Goshtasby, A. Ardeshir. *2-D and 3-D image registration: for medical, remote sensing, and industrial applications*. John Wiley & Sons, 2005.
- [161] Rusu, Radu Bogdan, and Steve Cousins, "3d is here: Point cloud library (pcl)." In *Robotics and Automation (ICRA)*, 2011 IEEE International Conference on, pp. 1-4. IEEE, 2011.
- [162] Dirk Holz, Alexandru-Eugen Ichim, Federico Tombari, Radu B. Rusu, and Sven Behnke, "A modular framework for aligning 3D point clouds - Registration with the Point Cloud Library", Accepted for *IEEE Robotics and Automation Magazine*, to appear in 2015.
- [163] Xu, Haili, Xingguo Zhang, Guoran Hua, and Sun An Wang, "Industrial robot pose estimation using three-dimensional registration," In *Industrial Informatics, 2009. INDIN 2009. 7th IEEE International Conference on*, pp. 674-679. IEEE, 2009.
- [164] Wang, Cailing, Xiaoyuan Jing, and Chunxia Zhao, "Local Upsampling Fourier Transform for accurate 2D/3D image registration." *Computers & Electrical Engineering* 38, no. 5 (2012): 1346-1357.
- [165] Luce, Jason, James Gray, Mark A. Hoggarth, Jeffery Lin, Elizabeth Loo, Maria I. Campana, and John C. Roeske, "Medical Image Registration Using the Fourier Transform." *International Journal of Medical Physics, Clinical Engineering and Radiation Oncology* 2014 (2014).

- [166] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. and Xiao, J., 2015. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1912-1920).
- [167] Papoulis, Athanasios. Signal analysis. Vol. 191. New York: McGraw-Hill, 1977.
- [168] Bracewell, R. N., K-Y. Chang, A. K. Jha, and Y-H. Wang. “*Affine theorem for two-dimensional fourier transform*” Electronics Letters 29, no. 3 (1993): 304.
- [169] Kuglin, C.D., Hines D.C., “The phase correlation image alignment method, IEEE 1975 Conference on Cybernetics and Society, p163-165, 1975.
- [170] Horner JL, Gianino PD. Phase-only matched filtering. Applied optics. 1984 Mar 15;23(6):812-6.
- [171] Althloothi, Salah, Mohammad H. Mahoor, and Richard M. Voyles, “A robust method for rotation estimation using spherical harmonics representation.” Image Processing, IEEE Transactions on 22, no. 6 (2013): 2306-2316.
- [172] Bulow, Henning, and Andreas Birk, “Spectral 6DOF registration of noisy 3D range data with partial overlap.” Pattern Analysis and Machine Intelligence, IEEE Transactions on 35, no. 4 (2013): 954-969.
- [173] Besl, Paul J., and Neil D. McKay, “Method for registration of 3-D shapes.” In Robotics-DL tentative, pp. 586-606. International Society for Optics and Photonics, 1992.
- [174] Yang, Jiaolong, Hongdong Li, and Yunde Jia. ”Go-icp: Solving 3d registration efficiently and globally optimally.” In Proceedings of the IEEE International Conference on Computer Vision, pp. 1457-1464. 2013.

- [175] Yang, Jiaolong, Hongdong Li, Dylan Campbell, and Yunde Jia, "Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration," *IEEE Trans. Pattern Analysis and Machine Intelligence* (2016).
- [176] Lucchese, Luca, Gianfranco Doretto, and Guido Maria Cortelazzo, "A frequency domain technique for range data registration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24, no. 11 (2002): 1468-1484.
- [177] Keller, Yosi, Amir Averbuch, and Yoel Shkolnisky, "Algebraically accurate volume registration using Euler's theorem and the 3D pseudo-polar FFT." In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 795-800. IEEE, 2005.
- [178] Keller, Yosi, Yoel Shkolnisky, and Amir Averbuch, "Volume registration using the 3-d pseudopolar Fourier transform." *Signal Processing, IEEE Transactions on* 54, no. 11 (2006): 4323-4331.
- [179] Kostelec, Peter J., and Daniel N. Rockmore, "FFTs on the rotation group." *Journal of Fourier analysis and applications* 14, no. 2 (2008): 145-179.
- [180] Keller, Yosi, Yoel Shkolnisky, and Amir Averbuch, "The angular difference function and its application to image registration." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27, no. 6 (2005): 969-976.
- [181] Bermanis, Amit, Amir Averbuch, and Yosi Keller, "3-d symmetry detection and analysis using the pseudo-polar fourier transform." *International journal of computer vision* 90, no. 2 (2010): 166-182.
- [182] Averbuch, Amir, and Yoel Shkolnisky. "3D Fourier based discrete Radon transform." *Applied and Computational Harmonic Analysis* 15, no. 1 (2003): 33-69.

- [183] S. Alam Abbas, H. Foroosh, and Q. Sun, “An exact and fast computation of Discrete Fourier Transform for Polar and Spherical grid,” *IEEE Transactions in Signal Processing*, Volume: 65, Issue: 8, April 15, 2017..
- [184] Kostelec, Peter J, “The Soft Package”, Retrieved from, <http://www.cs.dartmouth.edu/geelong/soft/>, 2015.
- [185] Hartley, Richard, and Andrew Zisserman, *Multiple view geometry in computer vision*, Cambridge university press, 2003.
- [186] Foroosh, Hassan, Josiane B. Zerubia, and Marc Berthod, “*Extension of phase correlation to subpixel registration*” *IEEE Transactions on Image Processing*, 11.3 (2002): 188-200.
- [187] Balci, M. and Foroosh, H, “Subpixel estimation of shifts directly in the Fourier domain”, *IEEE Transactions on Image Processing*, 15(7), pp.1965-1972, 2006.
- [188] Guizar-Sicairos, Manuel, Samuel T. Thurman, and James R. Fienup, “Efficient subpixel image registration algorithms.” *Optics letters* 33, no. 2 (2008): 156-158.
- [189] Tzimiropoulos, G., Argyriou, V. and Stathaki, T., 2011. Subpixel registration with gradient correlation. *IEEE Transactions on Image Processing*, 20(6), pp.1761-1767.
- [190] Chapel Hill Volume Rendering Test Data Set, Volume II, Dr. Julian Rosenman, North Carolina Memorial Hospital, University of North Carolina, retrieved from <https://graphics.stanford.edu/data/voldata/>.
- [191] Shilane, Philip, Patrick Min, Michael Kazhdan, and Thomas Funkhouser, “The princeton shape benchmark,” In *Shape modeling applications*, 2004. Proceedings, pp. 167-178. IEEE, 2004.
- [192] SPHARM-MAT, A matlab-based 3D shape modeling and analysis toolkit, Retrieved from, <http://www.iu.edu/spharm/>, 2016.

- [193] CGAL, Computational Geometry Algorithms Library, Retrieved from, <http://www.cgal.org>, 2016.
- [194] Elvins, T. Todd. "A survey of algorithms for volume visualization." *ACM Siggraph Computer Graphics* 26, no. 3 (1992): 194-201.
- [195] Ikits, Milan, Joe Kniss, Aaron Lefohn, and Charles Hansen. "Volume rendering techniques." *GPU Gems* 1 (2004).
- [196] Preim, Bernhard, and Dirk Bartz. *Visualization in medicine: theory, algorithms, and applications*. Morgan Kaufmann, 2007.
- [197] Levoy, Marc. *Volume rendering using the fourier projection-slice theorem*. Computer Systems Laboratory, Stanford University, 1992.
- [198] Malzbender, Tom, "Fourier volume rendering." *ACM Transactions on Graphics (TOG)* 12, no. 3 (1993): 233-250.
- [199] Totsuka, Takashi, and Marc Levoy, "Frequency domain volume rendering." In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pp. 271-278. ACM, 1993.
- [200] Dunne, S., S. Napel, and B. Ruth, "Interactive display of volumetric data by fast fourier projection." *Computerized medical imaging and graphics* 16, no. 4 (1992): 237-251.
- [201] Abdellah, Marwan, Ayman Eldeib, and Amr Sharawi. "Matlab-based Fourier volume rendering framework." In *Biomedical Engineering Conference (CIBEC)*, 2014 Cairo International, pp. 55-58. IEEE, 2014.
- [202] Abdellah, Marwan, Ayman Eldeib, and Amr Sharawi, "High Performance GPU-Based Fourier Volume Rendering." *International journal of biomedical imaging* 2015 (2015).

- [203] Abdellah, Marwan, Ayman Eldeib, and Amr Shaarawi, "Constructing a functional Fourier volume rendering pipeline on heterogeneous platforms." In Biomedical Engineering Conference (CIBEC), 2012 Cairo International, pp. 77-80. IEEE, 2012.
- [204] Hansen, Charles D., and Chris R. Johnson. Visualization handbook. Academic Press, 2011.
- [205] "Online library for medical MR and CT datasets, January 2013, <http://www.volvis.org/>.
- [206] The Volume Library, "Online library for Volume visualization datasets, <http://lgdv.cs.fau.de/External/vollib/>
- [207] Liu, Yan, Zhengrong Liang, Jianhua Ma, Hongbing Lu, Ke Wang, Hao Zhang, and William Moore. "Total variation-stokes strategy for sparse-view X-ray CT image reconstruction." IEEE transactions on medical imaging 33, no. 3 (2014): 749-763.
- [208] Yan, B., Jin, Z., Zhang, H., Li, L., Cai, A. (2015). NUFFT-based iterative image reconstruction via alternating direction total variation minimization for sparse-view CT. Computational and mathematical methods in medicine, 2015.
- [209] Gilles, Jrme, Giang Tran, and Stanley Osher, "2D empirical transforms. wavelets, ridgelets, and curvelets revisited," SIAM Journal on Imaging Sciences 7, no. 1 (2014): 157-186.
- [210] Fadili, Jalal, and Jean-Luc Starck. "Curvelets and ridgelets," In Computational Complexity, pp. 754-773. Springer New York, 2012.
- [211] Candes, Emmanuel Jean. "Ridgelets: theory and applications." PhD diss., Stanford University, 1998.
- [212] Candes, Emmanuel J., and David L. Donoho, "Ridgelets: A key to higher-dimensional intermittency?." Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 357, no. 1760 (1999): 2495-2509.

- [213] Candes, Emmanuel Jean, and David Leigh Donoho, Curvelets: A surprisingly effective non-adaptive representation for objects with edges. Department of Statistics, Stanford University, 1999.
- [214] Ma, Jianwei, and Markus Fenn, “Combined complex ridgelet shrinkage and total variation minimization,” *SIAM Journal on Scientific Computing* 28, no. 3 (2006): 984-1000.
- [215] Flesia, A. G., H. Hel-Or, A. Averbuch, E. J. Candes, R. R. Coifman, and D. L. Donoho, “Digital implementation of ridgelet packets.” *Studies in Computational Mathematics* 10 (2003): 31-60.
- [216] Donoho, David Leigh, and Ana Georgina Flesia, “Digital ridgelet transform based on true ridge functions,” *Studies in Computational Mathematics* 10 (2003): 1-30.
- [217] Candes, Emmanuel J., and David L. Donoho, “New tight frames of curvelets and optimal representations of objects with piecewise  $C^2$  singularities.” *Communications on pure and applied mathematics* 57, no. 2 (2004): 219-266.
- [218] Candes, Emmanuel, Laurent Demanet, David Donoho, and Lexing Ying, “Fast discrete curvelet transforms.” *Multiscale Modeling Simulation* 5, no. 3 (2006): 861-899.
- [219] Ma, Jianwei, and Gerlind Plonka, “The curvelet transform.” *Signal Processing Magazine, IEEE* 27, no. 2 (2010): 118-133.
- [220] Starck, Jean-Luc, Michael Elad, and David Donoho, “Redundant multiscale transforms and their application for morphological component separation.” *Advances in Imaging and Electron Physics* 132 (2004): 287-348.
- [221] Keiner, Jens, Stefan Kunis, and Daniel Potts. “Using NFFT 3—a software library for various nonequispaced fast Fourier transforms.” *ACM Transactions on Mathematical Software (TOMS)* 36, no. 4 (2009): 19.

- [222] Do, Minh N., and Martin Vetterli, "The contourlet transform: an efficient directional multiresolution image representation." *Image Processing, IEEE Transactions on* 14, no. 12 (2005): 2091-2106.
- [223] Candes, Emmanuel J., and David L. Donoho, "Recovering edges in ill-posed inverse problems: Optimality of curvelet frames." *Annals of statistics* (2002): 784-842.
- [224] Cands, Emmanuel J., and David L. Donoho, "Curvelets and curvilinear integrals." *Journal of Approximation Theory* 113, no. 1 (2001): 59-90.
- [225] Demanet, Laurent, "Curvelets, wave atoms, and wave equations." PhD diss., California Institute of Technology, 2006.
- [226] Hartley, Richard, and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [227] Starck, Jean-Luc, Emmanuel J. Cands, and David L. Donoho, "The curvelet transform for image denoising." *Image Processing, IEEE Transactions on* 11, no. 6 (2002): 670-684.
- [228] Chen, G. Y., and Balzs Kgl. "Image denoising with complex ridgelets." *Pattern Recognition* 40, no. 2 (2007): 578-585.
- [229] Abrial, P., Y. Moudden, J-L. Starck, B. Afeyan, J. Bobin, J. Fadili, and M. K. Nguyen, "Morphological component analysis and inpainting on the sphere: Application in physics and astrophysics." *Journal of Fourier Analysis and Applications* 13, no. 6 (2007): 729-748.
- [230] Candes, Emmanuel J., and David L. Donoho, "Continuous curvelet transform: II. Discretization and frames." *Applied and Computational Harmonic Analysis* 19, no. 2 (2005): 198-222.
- [231] Smith, Hart F, "A Hardy space for Fourier integral operators." *The Journal of Geometric Analysis* 8, no. 4 (1998): 629-653.



- [232] Smith, Hart F, “Wave equations with low regularity coefficients.” *Documenta Mathematica* (1998): 723-730.
- [233] Guo, Kanghui, and Demetrio Labate, “Optimally sparse multidimensional representation using shearlets.” *SIAM journal on mathematical analysis* 39, no. 1 (2007): 298-318.
- [234] Kutyniok, Gitta, Morteza Shahram, and David L. Donoho, “Development of a digital shearlet transform based on pseudo-polar FFT.” In *SPIE Optical Engineering+ Applications*, pp. 74460B-74460B. International Society for Optics and Photonics, 2009.
- [235] Kutyniok, Gitta, Wang-Q. Lim, and Xiaosheng Zhuang, “Digital shearlet transforms.” In *Shearlets*, pp. 239-282. Birkhuser Boston, 2012.
- [236] Grohs, Philipp, Sandra Keiper, Gitta Kutyniok, and Martin Schfer, “Parabolic Molecules: Curvelets, Shearlets, and Beyond.” In *Approximation Theory XIV: San Antonio 2013*, pp. 141-172. Springer International Publishing, 2014.
- [237] M. Vetterli, P. Marziliano, and T. Blu, “Sampling signals with nite rate of innovation”, *IEEE Trans. Signal Proc.*, 50(2002), 14171428.
- [238] Dragotti, Pier Luigi, Martin Vetterli, and Thierry Blu., “Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets StrangFix.” *Signal Processing, IEEE Transactions on* 55.5 (2007): 1741-1757.
- [239] Stoica, Petre, and Randolph L. Moses., “Introduction to spectral analysis”, Vol. 1. New Jersey:: Prentice hall, 1997.
- [240] R. E. Blahut, “Theory and Practice of Error Control Codes” Addison Wesley, MA, 1984
- [241] Ferreira, Paulo Jorge SG, and Jos MN Vieira., “Locating and correcting errors in images.” *Image Processing, 1997. Proceedings., International Conference on*. Vol. 1. IEEE, 1997.

- [242] Vieira, Jose MN, and Paulo JSG Ferreira., “Interpolation, spectrum analysis, error-control coding, and fault-tolerant computing.” *Acoustics, Speech, and Signal Processing*, 1997. ICASSP-97., 1997 IEEE International Conference on. Vol. 3. IEEE, 1997.
- [243] Elad, Michael, Peyman Milanfar, and Gene H. Golub., “Shape from moments-an estimation theory perspective.” *Signal Processing, IEEE Transactions on* 52.7 (2004): 1814-1829.
- [244] Milanfar, Peyman, et al., “Reconstructing polygons from moments with connections to array processing.” *Signal Processing, IEEE Transactions on* 43.2 (1995): 432-443.
- [245] Gu, Huaijin, and Robert Gao., “Resolution of overlapping echoes and constrained matched filter.” *Signal Processing, IEEE Transactions on* 45.7 (1997): 1854-1857.
- [246] Shan, T.J., Bruckstein, A.M. and Kailath, T., “Adaptive Resolution Of Overlapping Echoes”, *Circuits, Systems and Computers*, 1985. Nineteenth Asilomar Conference on, (1985), 431 - 435.
- [247] Weiss, Anthony J., Alan S. Willsky, and Bernard C. Levy., “Eigenstructure approach for array processing with unknown intensity coefficients.” *Acoustics, Speech and Signal Processing, IEEE Transactions on* 36.10 (1988): 1613-1617.
- [248] Blu, Thierry, et al., “Sparse sampling of signal innovations.” *Signal Processing Magazine, IEEE* 25.2 (2008): 31-40.
- [249] Chen, Yi-Chen, Challa S. Sastry, Vishal M. Patel, P. Jonathon Phillips, and Rama Chellappa, “In-plane rotation and scale invariant clustering using dictionaries,” *IEEE Transactions on Image Processing* 22, no. 6 (2013): 2166-2180.