Recent Techniques for Regularization

in Partial Differential Equations and Imaging

by

Theresa Scarnati

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved March 2018 by the
Graduate Supervisory Committee:

Anne Gelb, Co-Chair
Rodrigo Platte, Co-Chair
Douglas Cochran
Carl Gardner
Toby Sanders

ARIZONA STATE UNIVERSITY

May 2018

ABSTRACT

Inverse problems model real world phenomena from data, where the data are often noisy and models contain errors. This leads to instabilities, multiple solution vectors and thus ill-posedness. To solve ill-posed inverse problems, regularization is typically used as a penalty function to induce stability and allow for the incorporation of *a priori* information about the desired solution. In this thesis, high order regularization techniques are developed for image and function reconstruction from noisy or misleading data. Specifically the incorporation of the Polynomial Annihilation operator allows for the accurate exploitation of the sparse representation of each function in the edge domain.

This dissertation tackles three main problems through the development of novel reconstruction techniques: (i) reconstructing one and two dimensional functions from multiple measurement vectors using variance based joint sparsity when a subset of the measurements contain false and/or misleading information, (ii) approximating discontinuous solutions to hyperbolic partial differential equations by enhancing typical solvers with ℓ_1 regularization, and (iii) reducing model assumptions in synthetic aperture radar image formation, specifically for the purpose of speckle reduction and phase error correction. While the common thread tying these problems together is the use of high order regularization, the defining characteristics of each of these problems create unique challenges.

Fast and robust numerical algorithms are also developed so that these problems can be solved efficiently without requiring fine tuning of parameters. Indeed, the numerical experiments presented in this dissertation strongly suggest that the new methodology provides more accurate and robust solutions to a variety of ill-posed inverse problems.

To my family,

for their continual support and love.

ACKNOWLEDGMENTS

I would like to start by thanking my advisor, Dr. Anne Gelb for her unending support, encouragement and mentorship. I am truly grateful for all the time she has dedicated to my success and for all that she has taught me while challenging me to become a strong mathematician.

I would also like to thank my advisor Dr. Rodrigo Platte for teaching me the computational skills required to complete this dissertation and for his guidance towards interesting and important unsolved problems.

Thank you to my remaining dissertation committee members: Dr. Douglas Cochran, Dr. Carl Gardner and Dr. Toby Sanders. Dr. Cochran has been an inspiring teacher and has provided me with much insight into the field of mathematics. I also thank him for introducing me to the folks at AFRL. Thank you to Dr. Gardner for answering my numerical PDE questions. Thank you to Dr. Sanders for challenging me with tough scientific questions, for proposing the autofocusing problem and for always being down to hike a mountain with me.

Thank you to my family, Mom, Dad, sister and aunt, for supporting my decision to move across the country to pursue my Ph.D.

I would especially like to express my gratitude to my friends. First, to my best friend since seventh grade: Thank you for always being my best friend by supporting me and motivating me to never give up on this dream. Thank you to "my roommie" for being my first friend in Arizona, surviving all of our coursework together and being a great roommate. Thank you to my "adventure buddy" for your friendship and for maintaining my sanity by encouraging me go outside and be active. Thank you to all my other friends at ASU that have become family. Your emotional support throughout this experience has been so important.

Thank you to AFRL for proving financial support and great mentorship throughout my time in grad school. The two summer internships and full time position at AFRL have helped guide the research in this dissertation. Thank you to Ed Zelnio for his continued mentorship, for allowing me to ask so many questions, and for guiding my research in a relevant direction. Also thank you to Dr. Chris Paulson for being enthusiastic about my work, for our strong scientific discussions, and for your friendship.

This work has been supported in part by the grants AFOSR FA9550-15-1-0152 and NSF-DMS 1216559, the AFRL Autonomy Technology Research Center and the ASU School of Mathematical and Statistical Sciences. This document has been approved for public release, PA Approval #:[88AWB-2018-0808].

TABLE OF CONTENTS

APPENDIX

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Inverse and Ill-Posed Problems

The underlying goal of an inverse problem is to estimate an unknown function or quantity based on indirect measurements of such estimates. Typically, these measurements are noisy and contain many errors. Associated with a particular inverse problem is a forward model, which describes the relationship between the data and the unknown function. Because inverse problems can rarely be solved analytically, the forward model must be discretized to facilitate the use of computational methods. This discretization introduces further errors into the problem. With the accumulation of measurement errors and model errors, inverse problems quickly become unstable. In a general sense, an inverse problem is ill-posed if it has no solutions within a desired space, has many solutions within the space, or the solution procedure is unstable. Here, unstable means that arbitrarily small errors in the data, such as noise, may lead to large errors in the solutions. The investigations specific to ill-posedness that have proven helpful in this research include [73, 154, 64, 114, 72].

Because there are a wide variety of applications for inverse problems, encompassing physics, geophysics, medicine, ecology and economics, efficient methods for solving inverse problems have been investigated for many years. Moreover, increasingly powerful computational capabilities have spawned new numerical methods, whose approaches would have previously been deemed too costly for consideration. This has been coupled with the influx of massive data sets collected in various scientific disciplines. Some investigations along these lines that have informed this dissertation include [72, 73, 147, 166, 41, 95, 103, 111, 134].

High Order Total Variation Regularization

Researchers in the field of inverse and ill-posed problems develop and study stable methods for approximating unstable mappings, transformations or operations. *Regularization* is typically used as a penalty function to induce stability and to allow for the incorporation of a priori information about the desired solution. Specifically, regularization restricts the class of possible solutions and penalizes highly varying estimates. In this manner, it becomes feasible to find more accurate and stable solutions to ill-posed inverse problems.

Classical algorithms for image denoising and reconstruction have been primarily based on the least-squares method. Consequently, their outputs may be contaminated by the Gibbs phenomenon, and thus do not accurately approximate images containing edges or discontinuities. Total variation (TV) regularization has been used to overcome this difficulty [15, 43, 111, 119, 148, 88, 151, 158]. This technique preserves edges, however the images resulting from the application of TV in the presence of noise are often piecewise-constant and thus finer details in the original images may not be recovered satisfactorily. Specifically, the image defaults to a piecewise constant solution, often referred to as the "staircasing effect". High order total variation (HOTV) methods reduce the staircase effect while preserving jump discontinuities, [89, 134, 28, 4].

The theme of this dissertation is using HOTV regularization to reconstruct functions or images from noisy data. We first discuss how these techniques can be used to develop methods that exploit the use of multiple measurement vectors through non-parameteric, variance-based joint sparsity. We then formulate more accurate and robust estimations of discontinuous solutions to hyperbolic partial differential equations (PDEs). Finally we show that HOTV regularization is a useful technique for reducing various sources of error in synthetic aperture radar imagery.

Variance Based Weighted Joint Sparsity

Much research has recently been devoted to jointly sparse (JS) signal recovery from multiple measurement vectors (MMV) using $\ell_{2,1}$ regularization [3, 135, 120, 31, 33, 46], which is often more effective than performing separate recoveries using standard sparse recovery techniques. However, JS methods are difficult to parallelize due to their inherent coupling. In a recent paper [2] the *variance* based joint sparsity (VBJS) algorithm was introduced. VBJS is based on the observation that the pixel-wise variance across signals convey information about their shared support, motivating the use of a *weighted* ℓ_1 JS algorithm, where the weights depend on the information learned from calculated variance. Specifically, the ℓ_1 minimization should be more heavily penalized in regions where the corresponding variance is small, because it is likely there is no signal there. In this thesis, we expand on the original method, notably by introducing weights that ensure accurate, robust, and cost efficient recovery using both ℓ_1 and ℓ_2 regularization. Moreover, we show that the VBJS method can be applied in situations where some of the measurement vectors may misrepresent the unknown signals or images of interest, which is illustrated in several numerical examples.

Regularization for Hyperbolic PDEs

Regularization is commonly used in image reconstruction and signal processing when the inverse problem is ill-posed. However, the use of regularization to promote sparsity in the numerical solution of PDEs is more limited. Compressed sensing techniques were developed in [16, 17] to approximate solutions to elliptic PDEs. Hyperbolic PDEs present more challenges, especially when the solution emits shocks. Consequently, as time evolves the solution may contain a significant amount of diffusion (necessary for stability) or conversely it may be cluttered with spurious, unwanted oscillations. By diffusing the solution, the discontinuities are not resolved with a high order of accuracy. When spurious oscillations occur, errors are found throughout the entire solution domain. Those errors are amplified at each time step, and eventually solutions become unstable. Thus we see that at each time step there is an ill-posed inverse problem to be solved.

The research in this dissertation therefore takes a new approach to solving hyperbolic PDEs using HOTV regularization. Specifically, it is known that the solution will contain a sparse number of jump discontinuities at each iteration. Therefore, by incorporating the HOTV ℓ_1 regularization into the numerical method, we enforce this sparsity and restrict our solution at each time step to a class of stable and accurate solutions. The technique developed here appends a conventional numerical PDE solver with an ℓ_1 regularization term which will encourage the solution to be sparse in the jump discontinuity domain. A main advantage of this technique is that it does not require shock tracking. Further, the regularization appears to improve the time stepping restrictions normally required by the CFL condition. We provide several examples and discuss ideas on how to improve and further develop our method for other applications.

Regularization for Synthetic Aperture Radar

One area in image formation where ill-posed inverse problems arise is synthetic aperture radar (SAR) image formation. SAR is an all-weather, night and day, data and information acquisition technology commonly used for military applications, [69, 44]. To form a SAR image, a radar system transmits electromagnetic signals that hit scatterers in a target patch and reflect back to the receiver. The returned signals are then processed, and image formation algorithms are used to display a visual of the targeted scene.

To understand the signals that return to the SAR system, a continuous-time Fourier model describing the SAR data associated with the underlying reflectivity function as Fourier coefficients on a (partial) polar grid was developed, (see e.g. [69, 112, 104, 32, 44] and references therein). Typical SAR signal processing techniques can obtain high resolution images, but they do not account for the fact that, on the microscopic level, the underlying scenes are extremely rough. That is, the wavelength of the transmitted signal is much smaller than the size of the resolution cell. This leads to the constructive and destructive summation of signals that return from the same resolution cell. This phenomenon manifests as a multiplicative gamma noise that corrupts the entire imaging domain with salt and pepper like features, and is referred to as *speckle*, [57, 56, 113, 34]. Further, the Fourier model makes the unrealistic assumption that the location of the SAR imaging platform is known exactly. This erroneous assumption, along with the discretization of the continuous model, [58], leads to unwanted model errors. Thus we see that the associated inverse problem is to determine an estimate of the underlying scene given the noisy data and inaccurate forward model. These different sources of error cause the inverse problem to be extremely ill-posed, establishing the need for more accurate and robust image formation procedures.

Compressed sensing and regularization techniques have been extensively developed and explored for SAR image formation, [24, 25, 27, 166, 115, 108]. These methods use ℓ_p regularization with $p \leq 1$ to incorporate the prior knowledge that the scene contains a sparse amount of scatterers and a sparse amount of edges or variation. However, typical scenes consist of many scatterers, each with their own complex reflectivity properties and random phase. Note that none of the methods cited above use HOTV regularization, which has been shown to be effective in both low resolution environments and also when the underlying scene is highly variable, [120].

The research in this dissertation advances the work of [120] and uses the HOTV ℓ_1 regularization to form SAR imagery. The classic regularization problem is also updated to include a step for the estimation of the *phase errors* that arise from the imperfect knowledge of the location of the imaging platform. Our methods alternate between estimating the phase errors and estimating the image. A similar technique was developed in [26, 110], but again only considered TV regularization, and moreover did not account for the dependency of the phase errors on the spatial frequency values. In this thesis, phase errors are estimated using the phase synchronization technique, a common method for recovering phase in the phase retrieval community [68, 131]. The algorithm developed here is robust to noise and converges at an improved speed. Further, we propose two separate high order models for removing the multiplicative speckle noise present in the imagery. The first model is inspired by the work found in [8, 94, 9, 30], and the second model utilizes the variance based joint sparsity techniques developed in Chapter 3 of this thesis.

1.1 Contributions

This thesis makes the following novel contributions to the field of high order regularization techniques:

1. A robust, non-parametric variance based joint sparsity (VBJS) method is developed for reconstructing one and two dimensional functions from multiple measurement vectors.

2. Numerical partial differential equation solvers are enhanced using ℓ_1 regularization, gaining accuracy and stability of solutions that contain discontinuities or shocks.

3. The techniques above are then immediately used to develop two novel speckle

reduction techniques.

(a) The ℓ_1 enhanced numerical PDE solver is employed to approximate the steady state solution of an Euler-Lagrange PDE that models a denoised image.

(b) The VBJS technique is used to reduce speckle from multiple measurements of the same scene.

4. An auto-focusing method that uses an alternating minimization procedure estimating both the image and phase error from defocused SAR phase history data is developed. In particular, phase synchronization, a technique commonly utilized in the phase retrieval community, is adapted for the purpose of reducing the effects of phase error, and HOTV minimization is adapted for SAR image formation.

1.2 Outline of Thesis

The rest of this dissertation is organized as follows. Chapter 2 provides background information, including a more in depth review of ℓ_1 regularization. The HOTV operator is also thoroughly explained along with a description of the algorithm used to implement the proposed numerical method. In Chapter 3 we develop a variance-based weighted joint sparsity technique that uses the overlapping support of multiple measurement vectors to design weights that eliminate regularization parameters and ignore misleading data. Chapter 4 presents the new ℓ_1 regularization technique designed for approximating solutions to non-linear conservation laws. A detailed explanation of how standard PDE solvers are augmented with the regularization is provided, along with numerical results. We conclude the chapter with some ideas for future work in this area and provide a discussion of our results. Chapter 5 begins with a derivation of the mathematical model for speckle, and a discussion of current speckle reduction techniques, followed by development of two HOTV regularization techniques for the reduction of speckle. We then discuss phase error correction using autofocusing and explain how these phase errors should be incorporated into the forward SAR model, culminating in a joint phase error and image estimation procedure. Examples using HOTV for SAR image formation, speckle reduction and phase error estimation are displayed. To conclude, we discuss future work in SAR image formation.

Chapter 2

PRELIMINARIES

Section 2.1 consists of a thorough review of ℓ_1 regularization. We then present the *Polynomial Annihilation* (PA) operator, which is a high order total variation (HOTV) operator. Its formulation and convergence properties are given in Section 2.2, followed by a discussion of our HOTV optimization approach in Section 2.3.

2.1 ℓ_1 Regularization

In recent years, ℓ_1 regularization has received considerable attention in designing image reconstruction algorithms from under-sampled and noisy data when it is known a priori that some measurable features of the unknown image have a sparse representation in a particular domain or basis. Also, ℓ_1 regularization provides a formulation that is compatible with compressed sensing applications, specifically when an image can be reconstructed from a very small number of measurements [19, 18, 41].

Let $f \in \mathbb{R}^N$ be an unknown image or signal. We assume that some measurable features of f have sparse representation in a particular domain or basis. Define $\hat{f} \in \mathbb{R}^M$ to consist of data samples corresponding to f, and let $\mathcal{F} : \mathbb{R}^N \to \mathbb{R}^M$ be the forward model that projects f to $\hat{f}$. Typically, $\mathcal{F}$ is defined as a linear operator or invertible matrix. Define $H : \mathbb{R}^N \to \mathbb{R}$ to be the regularization operator. The objective is to solve the following optimization problem:

$$\underset{f}{\operatorname{argmin}} \;\; H(f) \quad \text{subject to} \quad \mathcal{F}f = \hat{f}. \tag{2.1}$$

Note that if the data $\hat{f}$ are under-sampled, then $\mathcal{F}$ may only contain a subset of rows of the forward model matrix. The equality constraint is the *data fidelity term*, and it measures how well the reconstructed image fits the given data for the particular forward model. The *regularization term* H enforces the known sparsity present in the underlying image by penalizing highly varying solutions and restricting the solution space to a desired class of functions. We will consider H to be the ℓ_1 norm of f (or some transformation of f) and note that any ℓ_p norm with $p \leq 1$ will enforce sparse solutions.

Typically, for measured data which is inherently noisy, the related total variation (TV) denoising problem is solved, which relaxes the equality constraint on the data fidelity term. It is formulated as

$$\operatorname*{argmin}_{f} \; H(f) \quad \text{subject to} \quad ||\mathcal{F}f - \hat{f}||_2^2 < \sigma, \tag{2.2}$$

where $\sigma \in \mathbb{R}$ is some positive regularization parameter [119, 148]. The TV denoising problem can be approximated as the following unconstrained problem by the introduction of a non-negative regularization parameter $\zeta \in \mathbb{R}$ that represents the trade-off between smoothness and fidelity to the original data [43, 95, 103, 111, 119]

$$\operatorname*{argmin}_{f} \left\{ H(f) + \frac{\zeta}{2} ||\mathcal{F}f - \hat{f}||_2^2 \right\}. \tag{2.3}$$

Ideally $H(f) = ||\mathcal{L}f||_0$, but because the ℓ_0 norm introduces non-convexity into the optimization problem, and produces an NP-hard problem [143], a convex relaxation of the ℓ_0 to the ℓ_1 problem is used with

$$H(f) := ||\mathcal{L}f||_1,$$

where $\mathcal{L} : \mathbb{R}^N \to \mathbb{R}^N$ maps f into a space where it is sparse. The regularization formulation in (2.3) can be interpreted as the numerical solution to a diffusive partial differential equation with specific initial conditions after one time step [126], or, in a Bayesian framework, the solution to a maximum a-posteriori problem with a Gaussian prior [73].

Often $\mathcal{L}$ is chosen to be the discrete total variation (TV) operator, which enforces the approximation of the first derivative of the solution to be zero at all but a small number of points. However, a well known drawback in using the TV operator is that the reconstructed image defaults to a piecewise constant approximation. While suitable for some applications, in others it is important to see more detail. This has been addressed in several ways. For example, total generalized variation (TGV), which generates piecewise (typically quadratic) polynomial approximations in smooth regions, was developed in [15]. Multi-wavelets have also been used to formulate sparsifying transforms, [127]. We will make use of the polynomial annihilation (PA) transform [4, 5, 153] as an appropriate HOTV regularization operator, which we briefly describe in the following section.

2.2 Polynomial Annihilation

Consider a function $f : [a, b] \to \mathbb{R}$. For all $y \in (a, b)$, let $f(y^-)$ and $f(y^+)$ denote the left and right hand limits of f, respectively. Then, the jump function of f at y is defined at each y by

$$[f](y) = f(y^+) - f(y^-). \tag{2.4}$$

Because $[f](y) = 0$ everywhere except in cells containing jumps, we say that $[f](y)$ has sparse representation. To approximate $[f](y)$ we use the polynomial annihilation (PA) edge detection method, originally proposed in [5], given by

$$\mathcal{L}^m f(y) = \frac{1}{q^m(y)} \sum_{x_j \in S} c_j(y) f(x_j). \tag{2.5}$$

Here m is the order of approximation to (2.4), S is a local set of $m + 1$ grid points about y from the set of given grid points, and the annihilation coefficients c_j are obtained by solving

$$\sum_{x_j} c_j(y) p_l(x_j) = p_l^{(m)}(y), \quad j = 1, ..., m+1. \tag{2.6}$$

Here p_l, $l = 0, ..., m$, is a basis for the space of polynomials of degree less than or equal to m. An explicit formula based on Newton divided differences was provided in [5]:

$$c_j(y) = \frac{m!}{\omega_j(S)}, \quad \omega_j(S) := \Pi_{i=1, i \neq j}^{m} (x_j - x_i), \tag{2.7}$$

for $j = 1, \cdots, m$. When the grid points are equally spaced, that is $x_j = j\Delta x$ with $\Delta x = (b-a)/N$ and $j = 0, \cdots, N$, (2.7) leads to

$$c_j = \frac{m!}{\prod\limits_{\substack{k=1 \\ k \neq j}}^{m+1} (j-k)\, \Delta x}. \tag{2.8}$$

The normalization factor q^m ensures convergence to the jump value at each discontinuity and is given by

$$q^m(y) = \sum_{x_j \in S^+} c_j(y), \tag{2.9}$$

where S^+ is the set of points $x_j \in S$ such that $x_j \geq y$. It was shown in [5] that (2.5) yields *mth* order convergence in smooth regions (outside the stencil containing the jump discontinuity) as long as f has m continuous derivatives in those regions. The method yields a first-order approximation to the jump value in regions where singularities are present. As m increases, oscillations develop in each jump location region. Post-processing methods can reduce the impact of these oscillations. However, because we are using (2.5) to enforce sparsity and *not* to explicitly detect edges, the oscillations occurring in jump regions do not have direct impact our results. That is, we are only interested in those locations which produce a high response when the function is transformed by the PA operator. Specifically, the PA method was used to design a *high order* ℓ_1 regularization operator in [4] as

$$H(f) := ||\mathcal{L}^m f||_1.$$

Consequently, the sparsity of edges in f will be enforced when solving (2.3).

A main advantage in using PA with $m > 1$ is that it enables distinction of jump discontinuities from steep gradients, which is critical in solving non-linear conservation laws and when reconstructing images that are not simply piecewise constant. It is also more effective than TV in low resolution environments, in particular by helping to reduce the staircase effect that occurs when using the TV operator (equivalently, PA $m = 1$) as the regularization operator.

We also note that the PA method can be used for multivariate, non-uniformly spaced data in any domain and in multiple dimensions. However, in [4] it was noted that for reconstruction on uniform grids, applying the PA transform dimension by dimension is as effective in promoting sparsity in two dimensions, and is more cost efficient. We therefore define $\mathcal{L}_x^m$ and $\mathcal{L}_y^m$ to be the respective directional PA transform operators and minimize

$$H(f) = ||\mathcal{L}_x^m f||_1 + ||\mathcal{L}_y^m f||_1, \tag{2.10}$$

to encourage sparsity in the two-dimensional jump function domain.

In this thesis investigation, (2.5) is included inside the optimization problem (2.3), and therefore only solved on the discrete set of grid points $y = x_j$, $j = 0, \cdots, N$. Hence (2.9) can also be determined explicitly and we are able to write the PA transform matrix for any integer m. For example, when considering a uniform grid spread over the domain $[a, b]$ and a periodic solution f, we have

$$\mathcal{L}^1 = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \\ -1 & & & & 1 \end{bmatrix}, \quad \mathcal{L}^3 = \frac{1}{2}\begin{bmatrix} 3 & -3 & 1 & & & -1 \\ -1 & 3 & -3 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & 3 & -3 & 1 \\ 1 & & & 1 & 3 & -3 \\ -3 & 1 & & & -1 & 3 \end{bmatrix}. \tag{2.11}$$

As noted previously, when $m = 1$, the PA transform is equivalent to using TV regularization. In this regard, the PA transform can be seen as a high-order total variation (HOTV) operator. Also note that (2.11) assumes periodicity. For non-periodic problems, there is an analogous transform that becomes one-sided as the boundaries are approached, [5]. Finally, even orders may also be used, because they still achieve sparsity in the edge domain. [1] Indeed even order transforms may be advantageous for some PDE solvers, and also in the case where there is some information known about the underlying solution. As will be demonstrated in Chapter 4, using $m = 2$ for Sod's shock tube problem yields the best results because the solution between the shock discontinuities is essentially piecewise linear. More information on the general properties of the PA transform used for recovering piecewise smooth solutions can be found in [5, 134, 4, 153].

To demonstrate the utility of the PA transform, consider a periodic function $f : [-1, 1] \to \mathbb{R}$ defined as the following saw-tooth function

$$f(x) = \begin{cases} x + 1, & -1 \le x \le 0 \\ x - 1, & 0 < x \le 1 \end{cases}, \tag{2.12}$$

[1] Even orders were not used in [5] because the post-processing techniques used for pinpointing the edges assumed that maximum (minimum) values occurred at the edge, which is true only for odd orders.

with corresponding jump function

$$[f](x) = \begin{cases} -2, & x = 0 \\ 0, & x \neq 0 \end{cases}. \tag{2.13}$$

Assume we are given f at equally spaced grid points $x_j = -1 + 2j/N$, $j = 0, \cdots, N$, where we choose $N = 64$. To determine an edge map of f, we apply (2.5) with $m = 2$. The results are shown in Figure 2.1. Observe that due to the piecewise linearity of (2.12), the PA method annihilates $[f]$ except in the jump region, where the jump is spread over two points.

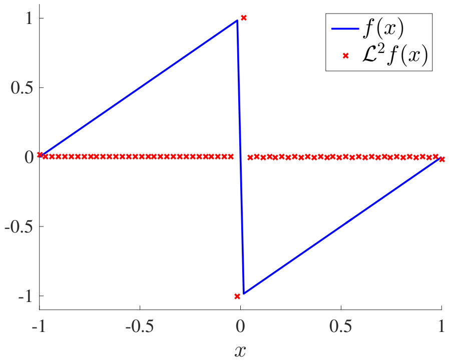

Figure 2.1: Result of applying $\mathcal{L}^m$, $m = 2$, to the saw-tooth function in equation (2.12). Observe that the edge occurs at the zero crossing between the two extrema of $\mathcal{L}^m f$.

Similarly, Figure 2.2 shows the results of applying $\mathcal{L}_x^m + \mathcal{L}_y^m$, $m = 2$, for a two dimensional image. Here, the test function used is given in [4] as

$$f(x,y) = \begin{cases} \sin(\pi\sqrt{x^2+y^2}/2), & 0 < x, y < \frac{3}{4} \\ g(x,y), & \text{otherwise}, \end{cases} \tag{2.14}$$

where

$$g(x,y) = \begin{cases} \cos(3\pi\sqrt{x^2+y^2}/2), & \sqrt{x^2+y^2} \leq \frac{1}{2} \\ \cos(\pi\sqrt{x^2+y^2}/2), & \sqrt{x^2+y^2} > \frac{1}{2}, \end{cases} \tag{2.15}$$

and $0 \leq x, y \leq 128$. Observe that the edge locations are approximated as the zero crossings of the PA response, as in the one-dimensional case.

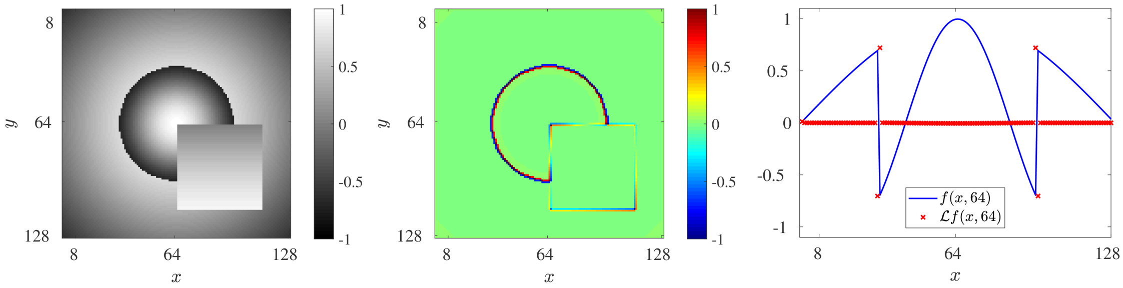

Figure 2.2: Given gray scale image (2.14) (left), the result (middle) of $\left(\mathcal{L}_x^m + \mathcal{L}_y^m\right) f$, $m = 2$ to the function defined in (2.14) and a cross section (center) of the edge response at location $y = 64$.

In summary, applying the PA transform to a piecewise smooth function f produces a vector with small, approximately zero values in the smooth regions of f and large values (in magnitude) at the jump locations. Performing minimization with $H(f) := ||\mathcal{L}^m f||_1$ encourages a solution f to (2.3) that has a sparse representation in the jump function domain [5, 134, 4, 153].

2.3 High-Order Total Variation Optimization Algorithm

It is important to solve (2.3) efficiently. In this section we discuss the alternating direction method of multipliers (ADMM), which is commonly used to solve ℓ_1 regularization problems, [88, 151, 158]. Our framework differs from the typical ADMM in two ways. First, rather than TV, we employ HOTV in the ℓ_1 regularization. Second, we also consider complex signals, both as data $\hat{f}$, as well as in the image f. We note that several comparable algorithms, for example the split Bregman algorithm, [55], have also been used to effectively split the regularization and fidelity terms.

We develop the ADMM algorithm for the equality constrained problem and note that the modifications for the unconstrained problem (2.3) simply involve eliminating the Lagrange multiplier associated with the data fidelity term. The discrete problem is formulated as

$$\underset{f}{\text{argmin}} \; ||\mathcal{L}^m f||_1 \quad \text{subject to} \quad \mathcal{F}f = \hat{f}, \tag{2.16}$$

where $f \in \mathbb{R}^N$ or $f \in \mathbb{C}^N$ is the optimal function we wish to produce. For simplicity we assume $f \in \mathbb{R}^N$ and note that only minor modifications to $\mathcal{L}^m$ are needed when the signal is complex. These modifications will be discussed in Chapter 5. $\mathcal{F} : \mathbb{R}^N \to \mathbb{R}^M$ is the forward operator, $\hat{f} \in \mathbb{R}^M$ is the given data set, and $\mathcal{L}^m : \mathbb{R}^N \to \mathbb{R}^N$ is the mth order PA operator (2.5). Application of the ADMM to numerical partial differential equations will be discussed in Chapter 4 and to SAR in Chapter 5.

The equality constrained problem in (2.16) is difficult to directly solve due to the non-differentiability present in the ℓ_1 norm and the non-seprability of f from the $\mathcal{L}^m$ operator. Instead, the solution is typically obtained by iteratively solving a sequence of unconstrained sub-problems [156, 77, 14]. To develop the algorithm, consider for example the equality constrained problem given by

$$\min_x f(x) \quad \text{subject to} \quad h(x) = 0. \tag{2.17}$$

Let $\lambda \in \mathbb{R}^N$ be a Lagrange multiplier and define the *Lagrangian function* associated with (2.17) as

$$L(x, \lambda) = f(x) - \lambda^T h(x). \tag{2.18}$$

The *first order optimality conditions* are given in the following theorem.

Theorem 2.3.1 *[156] Suppose that $\hat{x}$ is a local solution of (2.17) and that the functions f and h are continuously differentiable. Then there is a Lagrange multiplier* *vector $\hat{\lambda}$ such that the following conditions are satisfied:*

$$\nabla_x L(x, \hat{\lambda})|_{\hat{x}} = 0 \tag{2.19}$$

$$h(\hat{x}) = 0. \tag{2.20}$$

The proof of Theorem 2.3.1 can be found in any standard optimization text book such as [156]. From Theorem 2.3.1, we see that at the local minimum $\hat{x}$,

$$0 = \nabla_x L(x, \hat{\lambda})|_{\hat{x}} = \nabla_x f(x)|_{\hat{x}} - \hat{\lambda}^T \nabla_x h(x)|_{\hat{x}}. \tag{2.21}$$

Using the *quadratic penalty method* [156, 77, 14], the solution to (2.17) can be found by minimizing the quadratic penalty function

$$Q(x; \mu) = f(x) + \frac{\mu}{2} h^T(x) h(x), \tag{2.22}$$

where $\mu > 0$ is the *penalty parameter*. From (2.22) we see that driving μ towards ∞ will penalize the constraint $h(x) = 0$ with increasing severity. Thus, by considering a sequence of values $\{\mu_k\} \to \infty$ as $k \to \infty$ the quadratic penalty method seeks the minimizer x_{k+1} of $Q(x; \mu_k)$. That is,

$$x_{k+1} = \underset{x}{\operatorname{argmin}}\; Q(x; \mu_k). \tag{2.23}$$

The convergence properties of the quadratic penalty method are summarized in the following theorem.

Theorem 2.3.2 *[156] Suppose $\{\mu_k\} \to \infty$ as $k \to \infty$. If a limit point $\hat{x}$ of x_{k+1} is feasible, then $\hat{x}$ satisfies the first order optimality conditions of Theorem 2.3.1 and (2.21). Thus for $\lim_{k\to\infty} x_{k+1} = \hat{x}$,*

$$\lim_{k\to\infty} -\mu_k h(x_{k+1}) = \hat{\lambda}, \tag{2.24}$$

where $\hat{\lambda}$ is the multiplier in (2.18) that satisfies the first order optimality conditions for the equality constrained problem (2.17).

According to Theorem 2.3.2, at the final iteration of the quadratic penalty method,

$$h(x_{k+1}) \approx -\frac{\hat{\lambda}}{\mu_k}.$$

For the constraint $h(x_{k+1}) = 0$ to be satisfied, we now see that it must be that $\mu_k \to \infty$.

The *augmented Lagrangian method* (also referred to as the method of multipliers) overcomes the bottleneck of the quadratic penalty method by altering (2.23) such that we more nearly satisfy the equality constraint in (2.17) for moderate values of μ_k [156, 54]. The augmented Lagrangian function achieves this goal by including an estimate for $\hat{\lambda}$ based on (2.24). This results in a combination of the quadratic penalty function (2.22) and the Lagrangian function (2.18) associated with (2.17). The resulting augmented Lagrangian function is

$$L_A(x, \lambda; \mu) = f(x) - \lambda^T h(x) + \frac{\mu}{2} h^T(x) h(x). \tag{2.25}$$

For a fixed λ_k and μ_k, the augmented Lagrangian method seeks the minimizer x_{k+1} of $L_A(x, \lambda_k; \mu_k)$. That is,

$$x_{k+1} = \underset{x}{\operatorname{argmin}} \; L_A(x, \lambda_k; \mu_k). \tag{2.26}$$

Because x_{k+1} is the minimizer of (2.26), the optimality condition for the unconstrained problem (2.26) holds:

$$0 = \nabla_x L_A(x, \lambda_k; \mu_k)|_{x_{k+1}} = \nabla_x f(x)|_{x_{k+1}} - \left[\lambda_k^T - \mu_k h(x_{k+1})\right] \nabla_x h(x)|_{x_{k+1}}.$$

Thus, according to Theorem 2.3.1, because x_{k+1} minimizes (2.26), λ_k converges to $\hat{\lambda}$ as k increases with the following update:

$$\lambda_{k+1} = \lambda_k - \mu_k h(x_{k+1}). \tag{2.27}$$

This update ensures that λ_k is dual feasible, or equivalently $\nabla_x L(x, \lambda_{k+1})|_{x_{k+1}} = 0$. Moreover, for $\lambda_{k+1} \approx \hat{\lambda}$, we have that

$$h(x_{k+1}) = -\frac{1}{\mu_k}\left(\hat{\lambda} - \lambda_k\right), \tag{2.28}$$

and if λ_k is close to $\hat{\lambda}$, the equality constraint can be satisfied without the requirement that $\mu_k \to \infty$ as $k \to \infty$.

It can be shown that with exact knowledge of $\hat{\lambda}$, the solution $\hat{x}$ of (2.17) is a minimizer of $L_A(x, \hat{\lambda}; \mu)$ for all positive μ. Although we do not know $\hat{\lambda}$ in practice, we see from (2.28) that we can obtain a good estimate of $\hat{x}$ by minimizing (2.25) even when μ_k is not particularly large, provided that λ_k is a reasonable estimate of $\hat{\lambda}$. With the augmented Lagrangian method (2.26), x_{k+1} will be close to $\hat{x}$ if λ_k is close to $\hat{\lambda}$ *or* μ_k is large, unlike the quadratic penalty method, which only converges for large μ_k. The augmented Lagrangian method is summarized in Algorithm 1.

Algorithm 1 Augmented Lagrangian Method

```
1: Initialize: $\mu_0$, $\lambda_0$, $x_0$ and tolerance $tol$.
2: while $|| \nabla_x L_A(x, \lambda_k; \mu_k)|_{x_k} || > tol$ do
3:     Solve $x_{k+1} = \underset{x}{\operatorname{argmin}}\ L_A(x, \lambda_k; \mu_k)$.
4:     Update multiplier according to (2.27).
5:     Choose new penalty parameter $\mu_{k+1} \geq \mu_k$.
6: end while
```

We now explain how the augmented Lagrangian method is applied specifically to our problem (2.16). Observe that Step 3 in Algorithm 1 is the surrogate unconstrained minimization problem of (2.17). The particular augmented Lagrangian function associated with (2.16) is

$$\tilde{L}_A(f, \lambda; \mu) = ||\mathcal{L}^m f||_1 - \lambda^T(\mathcal{F}f - \hat{f}) + \frac{\mu}{2}||\mathcal{F}f - \hat{f}||_2^2. \tag{2.29}$$

Because $||\cdot||_1$ is not differentiable and f cannot be separated from $\mathcal{L}^m$, we introduce slack variables in (2.16)

$$w = \mathcal{L}^m f.$$

Thus, instead of solving (2.16) by minimizing (2.29), the augmented Lagrangian method is now used to solve the equivalent variant

$$\underset{w,f}{\operatorname{argmin}} ||w||_1, \quad \text{subject to} \quad \mathcal{F}f = \hat{f} \quad \text{and} \quad \mathcal{L}^m f = w. \tag{2.30}$$

Using yet another Lagrange multiplier augmentation,

$$||w||_1 - \nu^T(\mathcal{L}^m f - w) + \frac{\beta}{2}||\mathcal{L}^m f - w||_2^2, \tag{2.31}$$

the augmented Lagrangian function associated with (2.30) is

$$\begin{aligned} \tilde{L}_A(w, f, \nu, \lambda; \mu, \beta) = ||w||_1 - \nu^T(\mathcal{L}^m f - w) + \frac{\beta}{2}||\mathcal{L}^m f - w||_2^2 \\ - \lambda^T(\mathcal{F}f - \hat{f}) + \frac{\mu}{2}||\mathcal{F}f - \hat{f}||_2^2. \end{aligned} \tag{2.32}$$

Here, μ and β are parameters that do not effect the solution, but rather control the speed of convergence of the algorithm. From this we see that Step 3 in Algorithm 1 solves the specific problem

$$\underset{w,f}{\operatorname{argmin}} \; \tilde{L}_A(w, f, \nu_k, \lambda_k), \tag{2.33}$$

for fixed multipliers ν_k and λ_k, where $\tilde{L}_A(w, f, \nu, \lambda)$ is defined in (2.32), and the notational dependence on μ and β is dropped for simplicity.

As mentioned previously, the alternating direction method of multipliers (ADMM) is commonly used to solve ℓ_1 regularization problems, [88, 151, 158]. Because it is difficult to solve (2.33) jointly over both w and f, the ADMM algorithm splits the problem and alternates between solving for w with a fixed f and solving for f with a fixed w. We now apply the ADMM to solve (2.33) in Step 3 of Algorithm 1. Below we describe how this is accomplished.

In the ADMM algorithm, the problem is split into a w *sub-problem* and a f *sub-problem*. In the w sub-problem f, ν and λ are held fixed while solving for the value of w that minimizes (2.33), and in the f sub-problem w, ν and λ are held fixed while solving for the optimal f. The Lagrangian multipliers are adjusted after f and w have alternatively been updated. Thus, when performing the optimization, we only need to include terms that correspond to the sub-problem we are considering.

For example, let f_k, ν_k, λ_k and w_k be approximate minimizers of $\tilde{L}_A(w, f, \nu, \lambda)$ at the kth inner iteration. We calculate

$$w_{k+1} = \underset{w}{\operatorname{argmin}} \ \tilde{L}_A(w, f_k, \nu_k, \lambda_k)$$

by solving the w sub-problem, defined as

$$w_{k+1} = \underset{w}{\operatorname{argmin}} \left\{ ||w||_1 - \nu_k^T(\mathcal{L}^m f_k - w) + \frac{\beta}{2}||\mathcal{L}^m f_k - w||_2^2 \right\}. \tag{2.34}$$

The following lemma from [88] enables fast implementation for the w sub-problem.

Lemma 2.3.3 *Consider the following problem for $\beta > 0$ and $\nu, y \in \mathbb{R}^N$:*

$$\min_x ||x||_1 - \nu^T(y - x) + \frac{\beta}{2}||y - x||_2^2 \tag{2.35}$$

The minimizer is given by the one-dimensional shrinkage-like formula

$$\hat{x} = \max\left\{ \left| y - \frac{\nu}{\beta} \right| - \frac{1}{\beta}, 0 \right\} sign\left(y - \frac{\nu}{\beta} \right). \tag{2.36}$$

Proof 2.3.4 *Following the ideas in [88], because the objective function is convex, bounded below and coersive, there exists at least one minimizer $\hat{x}$ of (2.35) [156]. The subdifferential of $f(x) = ||x||_1$ is given element-wise as [88]*

$$(\partial_x f(x))_i = \begin{cases} sign(x_i), & x_i \neq 0 \\ \{h : |h| \leq 1, h \in \mathbb{R}\}, & otherwise, \end{cases} \tag{2.37}$$

where the origin is required to be included according to the optimality condition for convex problems. According to (2.37), each component $\hat{x}_i$ *must satisfy*

$$\begin{cases} sign(\hat{x}_i) + \beta(\hat{x}_i - y_i) + \nu_i = 0, & \hat{x}_i = 0 \\ |\nu_i - \beta y_i| \le 1, & otherwise. \end{cases} \tag{2.38}$$

If $\hat{x}_i \neq 0$*, (2.38) yields*

$$\frac{1}{\beta} sign(\hat{x}_i) + \hat{x}_i = y_i - \frac{\nu_i}{\beta}. \tag{2.39}$$

Because $1/\beta > 0$*, taking the absolute value of both sides of (2.39) gives*

$$\frac{1}{\beta} + |\hat{x}_i| = |y_i - \frac{\nu_i}{\beta}|. \tag{2.40}$$

Combining (2.39) and (2.40) shows that

$$\begin{aligned} sign(\hat{x}_i) &= \frac{sign(\hat{x}_i)|\hat{x}_i| + sign(\hat{x}_i)/\beta}{|\hat{x}_i| + 1/\beta} = \frac{\hat{x}_i + sign(\hat{x}_i)/\beta}{|\hat{x}_i| + 1/\beta} \\ &= \frac{y_i - \nu_i/\beta}{|y_i - \nu_i/\beta|} = sign\left(y_i - \frac{\nu_i}{\beta}\right). \end{aligned} \tag{2.41}$$

Thus, using (2.40) and (2.41) we have for $\hat{x}_i \neq 0$

$$\hat{x}_i = |\hat{x}_i| sign(\hat{x}_i) = \left(|y_i - \frac{\nu_i}{\beta}| - \frac{1}{\beta}\right) sign\left(y_i - \frac{\nu_i}{\beta}\right). \tag{2.42}$$

On the other hand, $\hat{x}_i = 0$ *if and only if [88]*

$$|y_i - \frac{\nu_i}{\beta}| \le \frac{1}{\beta}. \tag{2.43}$$

By coupling (2.42) with (2.43), we can conclude that

$$\hat{x}_i = \max\left\{|y_i - \frac{\nu_i}{\beta}| - \frac{1}{\beta}, 0\right\} sign\left(y_i - \frac{\nu_i}{\beta}\right).$$

In vector form, this is equivalent to (2.36). □

From Lemma 2.3.3, we are able to *exactly* solve (2.34) at every iteration as

$$w_{k+1} = \max\left\{\left|\mathcal{L}^m f_k - \frac{\nu_k}{\beta}\right| - \frac{1}{\beta}, 0\right\} \text{sign}\left(\mathcal{L}^m f_k - \frac{\nu_k}{\beta}\right). \tag{2.44}$$

Next we consider the f sub-problem. We determine f_{k+1} from (2.32), while holding w_{k+1}, ν_k and λ_k fixed. Specifically,

$$f_{k+1} = \underset{f}{\operatorname{argmin}} \quad \tilde{L}_A(w_{k+1}, f, \nu_k, \lambda_k)$$

is equivalent to solving

$$f_{k+1} = \underset{f}{\operatorname{argmin}} \ J(f), \tag{2.45}$$

where

$$J(f) := -\nu_k^T(\mathcal{L}^m f - w_{k+1}) + \frac{\beta}{2}||\mathcal{L}^m f - w_{k+1}||_2^2 - \lambda_k^T(\mathcal{F}f - \hat{f}) + \frac{\mu}{2}||\mathcal{F}f - \hat{f}||_2^2. \tag{2.46}$$

We define the combination of (2.45) and (2.46) as the f *sub-problem.* Notice $J(f)$ is a quadratic function with gradient

$$d(f) = \beta(\mathcal{L}^m)^T(\mathcal{L}^m f - w_{k+1}) - (\mathcal{L}^m)^T\nu_k + \mu\mathcal{F}^T(\mathcal{F}f - \hat{f}) - \mathcal{F}^T\lambda_k. \tag{2.47}$$

By forcing $d(f) = 0$, so that

$$\beta(\mathcal{L}^m)^T\mathcal{L}^m f + \mu\mathcal{F}^T\mathcal{F}f = \beta(\mathcal{L}^m)^T w_{k+1} + (\mathcal{L}^m)^T\nu_k + \mu\mathcal{F}^T\hat{f} + \mathcal{F}^T\lambda_k$$

we achieve the exact minimizer of $J(f)$ in (2.46) as

$$f_{k+1} = \left(\beta(\mathcal{L}^m)^T\mathcal{L}^m + \mu\mathcal{F}^T\mathcal{F}\right)^\dagger \left(\beta(\mathcal{L}^m)^T w_{k+1} + (\mathcal{L}^m)^T\nu_k + \mu\mathcal{F}^T\hat{f} + \mathcal{F}^T\lambda_k\right), \tag{2.48}$$

where $A^\dagger$ denotes the pseudo-inverse of A. As it is typically not efficient to compute the (pseudo-) inverse of a large matrix that might additionally be ill-conditioned, an iterative method is more commonly used for solving the f sub-problem. We describe the process below.

To ensure computational efficiency, we start by taking an aggressive step in the steepest descent direction starting at f_k:

$$\begin{aligned} f_{k+1} &= f_k - \alpha_k d_k \\ d_k &= d(f_k). \end{aligned} \tag{2.49}$$

The step size α_k is chosen as the Barzilai-Borwein (BB) step (see [10]),

$$\alpha_k = \frac{s_k^T s_k}{s_k^T y_k}, \tag{2.50}$$

with

$$s_k = f_k - f_{k-1}$$

$$y_k = d(f_k) - d(f_{k-1}).$$

A backtracking algorithm is performed to ensure α_k is not chosen to be too large. This requires checking what is known as the Armijo condition, [156], which guarantees that using (5.62) sufficiently reduces the magnitude of the objective function. Algorithmically, the Armijo condition is given by

$$J(f_k - \alpha_k d_k) \leq J(f_k) - \delta \alpha_k d_k^T d_k, \tag{2.51}$$

where $\delta \in (0, 1)$. If the Armijo condition (2.51) is not satisfied, we backtrack and decrease the step length according to

$$\alpha_k = \rho \alpha_k, \tag{2.52}$$

where $\rho \in (0, 1)$ is chosen as the backtracking parameter. At the kth iteration of the algorithm, after the new w and f values are found using (2.34) and (2.45), the Lagrange multipliers are updated according to

$$\begin{aligned} \nu_{k+1} &= \nu_k - \beta(\mathcal{L}^m f_{k+1} - w_{k+1}) \\ \lambda_{k+1} &= \lambda_k - \mu(\mathcal{F} f_{k+1} - \hat{f}). \end{aligned} \tag{2.53}$$

The minimization method in Algorithm 2 summarizes what is explained above and provides a step-by-step procedure of how to (alternatively) solve the w sub-problem (2.34) and f sub-problem (2.45) at each iteration, thus ultimately solving the high-order total variation (HOTV) minimization problem posed in (2.30). Typically, to initialize Algorithm 2, we choose $\rho = .4$ and $\delta = 10^{-4}$ [156, 88].

Algorithm 2 HOTV ADMM

```
Initialize $\nu_0$ and $\lambda_0$. Choose starting points $w_0$, $f_0$ and number of iteration $K$.
for $i = 0$ to $K$ do
    Set $0 < \rho, \delta < 1$ and tolerance $tol$.
    while $||f_{k+1} - f_k|| > tol$ do
        Compute $w_{k+1}$ using the shrinkage like formula (2.44).
        Set $\alpha_k$ using (5.62).
        while Armijo condition (2.51) unsatisfied do
            Backtrack $\alpha_k = \rho\alpha_k$.
        end while
        Compute $f_{k+1}$ using (2.49) and (2.47).
    end while
    Update Lagrange multipliers according to (3.40)
end for
```

This HOTV ADMM algorithm will be modified throughout this thesis to be used in variance based joint sparsity recovery (Chapter 3), the approximation of solutions to hyperbolic partial differential equations (Chapter 4), and synthetic aperture radar image formation (Chapter 5).

Chapter 3

REDUCING THE EFFECTS OF BAD DATA USING VARIANCE BASED JOINT SPARSITY RECOVERY

3.1 Introduction

Recovering sparse signals and piecewise smooth functions from under-sampled and noisy data has been a heavily investigated topic over the past decade. Typical algorithms minimize the ℓ_1 norm of an approximation of a sparse feature (e.g. wavelets, gradients, or edges) of the solution so that the reconstructed solution will preserve sparsity in its corresponding sparse domain. A weighted ℓ_1 reconstruction algorithm was introduced in [21] to reconcile the difference between the "true" sparsity ℓ_0 norm and the surrogate ℓ_1. Sparse signal recovery was accomplished by a solving a sequence of weighted ℓ_1 minimization problems, with the weights iteratively updated at each step. As was demonstrated there, updating the weights yielded successively improved estimations of the non-zero coefficient locations, and consequently relaxes standard sampling rate requirements for sparse signal recovery. An adjustment for the weight calculation was proposed in [29] resulting in an improvement to the iterative reweighting algorithm. An adaptively weighted total variation (TV) regularization algorithm, where the spatially adaptive weights were based on the difference of values between neighboring pixels, was introduced in [92]. A different weighting technique was developed in [28] to reduce the staircase effect of TV regularization. An adaptive function was used along with new parameters to balance the trade off between penalizing discontinuities and recovering sharp edges. While the method accomplishes the goal of allowing smooth transitions without reducing sharp edges, the mathematical formulation is challenging and uniqueness is not guaranteed. Further weighted ℓ_1 literature can be found at [21, 28, 92, 157, 159, 29] and references therein.

In many inverse problems, it may be possible to acquire multiple measurement vectors (MMVs) of the unknown signal or image, [52, 31, 91, 33, 46, 47, 3, 120]. MMV collection is especially useful when trying to recover solutions of an underdetermined system when the MMVs have the same, but unknown, sparsity structure. Techniques exploiting this type of commonality, referred to as *joint sparsity* (JS) methods, can be developed by extending the commonly used single measurement vector (SMV) algorithms for sparse solutions, [33]. Additional examples of this can be found in [87, 145, 142, 31, 155] and references therein. In particular, jointly sparse vectors are often recovered using the popular $\ell_{2,1}$ minimization, [37, 162, 31, 132, 164], which was thoroughly analyzed in [46, 47]. Conditions for guaranteeing improvements over SMV were determined for a class of MMV techniques in [46] and moreover, it was shown in [47] that under mild conditions the probability of *not* recovering a sparse vector with high probability (based on a chosen threshold) using $\ell_{2,1}$ regularization decays exponentially with the increase of measurements. Various algorithms are used to implement $\ell_{2,1}$ regularization, including the alternating direction method of multipliers (ADMM), split Bregman, joint-OMP, and "reduce-and-boost", [100, 144, 162]. An algorithm is typically chosen to yield the most efficiency for the particular problem at hand (for example, based on problem complexity). In this chapter we use ADMM, and note that while other methods may yield faster convergence for our chosen examples, in general $\ell_{2,1}$ regularization techniques are inherently coupled, making them difficult to parallelize.

While much work has been done on designing weighed ℓ_p (specifically ℓ_1) reconstruction methods for SMV, and in constructing joint sparsity MMV methods using the $\ell_{2,1}$ norm, there has been less work devoted to improving MMV through weighted ℓ_p minimization. Three notable investigations include: (i) [132], where the SMV weights were adapted from those in [21] to $\ell_{2,1}$ minimization for the problem of multi-channel electrocardiogram signal recovery. Although the technique enhances the sparseness of the solution and reduce the number of measurements required for accurate recovery, it requires hand tuning of parameters. (ii) [164], where a weighted $\ell_{2,1}$ minimization algorithm is used for direction of arrival estimation, high resolution radar imaging and other sparse recovery related problems using random measurement matrices. The singular value decomposition is used to exploit the relationship between the signal subspace and the noise subspace for designing the weights. (iii) [52], where a shape-adaptive jointly sparse classification method for hyperspectral imaging was developed. We note that all of these developments were problem specific, and not easily adapted for general sparse signal recovery.

In this chapter we propose using the *variance* based joint sparsity (VBJS) method for MMV, introduced in [2]. The VBJS technique exploits the idea that the variance across jointly sparse MMVs should be sparse in the sparsity domain of the underlying signal or image, an idea first proposed in [38] for the purpose of edge detection and localization. The *weights* in [2] used for the weighted ℓ_1 regularization term are essentially reciprocals of this variance (with a threshold built in to ensure no division by zero), with the idea being that the ℓ_1 term should be heavily penalized when the variance is small, but should not influence the solution as much when the variance is large. Presumably, the large variance indicates support of the image or signal in the sparse domain. One of the main advantages of VBJS is that it is easily parallelized. In particular, it was shown in [2] that VBJS is consistently more computationally efficient than $\ell_{2,1}$ regularization algorithms when using standard black box solvers. In this chapter we improve on the VBJS algorithm by designing weights that reduce the parametric dependence on the reconstruction, making it more amenable to a variety of other applications not considered in [2]. Specifically, the VBJS can now be used in situations where some measurement vectors may misrepresent the unknown function of interest. In contrast, such "rogue" data may wield undue influence on the reconstruction of piecewise smooth solutions when using the standard $\ell_{2,1}$ approach. The original VBJS approach does not adequately account for false data in the weight design, so much more parameter tuning would be needed. False data problems appear in applications including state estimation of electrical power grids, [90], large scale sensor network estimation, [160], synthetic aperture radar (SAR) automated target recognition (ATR), [75], and many others, [163, 165]. False data may be purposefully injected into these systems to decrease the performance of automated detection algorithms. In other situations, misrepresentations of data occur due to human error or environmental issues effecting the measurements. For example, in SAR ATR it is often the case that targets are obscured by their surroundings (trees) or by enemies (meshes placed over the targets). Also, additional parts may be taken off or added to targets, corrupting measurement data, [75]. As part of our reconstruction algorithm, we include a numerically efficient comparative measurement of the measurement vectors, which allow us to appropriately disregard rogue data and improve our overall reconstruction.

Our proposed VBJS technique offers several advantages: (i) Our method is (essentially) non-parametric so that regularization parameters need not be hand tuned; (ii) We take advantage of the joint sparsity information available in the MMV setup, thus improving reconstruction accuracy while decreasing sampling rates, independent of application; (iii) With some sharpness reduction, our weights allow us to use the ℓ_2 norm, which is much more cost efficient; (iv) Our method mitigates the effects of rogue data. Finally, as noted above, the VBJS algorithm is easily parallelizable, so even when using the weighted ℓ_1 norm, it is much more efficient than when the $\ell_{2,1}$

norm is used.

The rest of this chapter is organized as follows: In Section 3.2 we define joint sparsity for multi-measurement vectors and provide details for the standard $\ell_{2,1}$ regularization approach used to recover sparse signals. In Section 3.3 we describe the variance based joint sparsity (VBJS) approach, initially developed in [2], and demonstrate how weights should be constructed to reduce the impact of false information. We also propose a technique to choose the "best" solution from the set of possible solution vectors that can be recovered from the VBJS method, so that we do not have to compute each vector in the solution space. In Section 3.4 we prove that the alternating direction method of multipliers (ADMM) can be applied to the weighted ℓ_1 minimization. We also show how the VBJS method can be efficiently computed for the weighted ℓ_2 norm. Section 3.5 provides some numerical results for sparse signal recovery and one and two dimensional images. Some concluding remarks are given in Section 3.6.

3.2 Preliminaries

Consider a piecewise smooth function $f(x)$ on $[a, b]$. We seek to recover $\boldsymbol{f} : \mathbb{R}^N \to \mathbb{R}^N$, where each element of $\boldsymbol{f}$ is given as $f_i = f(x_i), i = 1, ..., N$, with

$$x_i = a + \Delta x(i-1), \tag{3.1}$$

and $\Delta x = \frac{b-a}{N}$. We note that x_i are chosen to be uniform for simplicity of numerical experiments and is not required for our algorithm.

Because the underlying function f is piecewise smooth, it is sparse in its corresponding edge domain. Formally we have:

Definition 3.2.1 *[31, 36] A vector $\boldsymbol{p} \in \mathbb{R}^N$ is s-sparse for some $1 \leq s \leq N$ if*

$$||\boldsymbol{p}||_0 = |\mathrm{supp}(\boldsymbol{p})| \leq s.$$

In our case, $\boldsymbol{p}$ corresponds to the edge vector of f at the set of grid points in (3.1).

Suppose we acquire J data vectors, $\boldsymbol{y}^j \in \mathbb{C}^M$, as

$$\boldsymbol{y}^j = \boldsymbol{A}^j(\boldsymbol{f}) + \boldsymbol{\eta}^j, \quad j = 1, ..., J. \tag{3.2}$$

Here $\boldsymbol{A}^j : \mathbb{R}^N \to \mathbb{C}^M$ is a forward operator (often defined as a square ($N = M$), orthogonal matrix for simplicity) and

$$\boldsymbol{\eta}^j \in \mathbb{C}^M, \quad j = 1, ..., J, \tag{3.3}$$

model J Gaussian noise vectors.

Due to the sparsity in the edge domain, ℓ_1 regularization provides an effective means for reconstructing $\boldsymbol{f}$ given any of the J noisy data vectors. Specifically, we compute the unconstrained optimization problem

$$\check{\boldsymbol{f}}^j = \underset{\boldsymbol{g}}{\operatorname{argmin}} \left\{ ||\boldsymbol{\mathcal{L}} \boldsymbol{g}||_1 + \frac{\mu}{2} ||\boldsymbol{A}^j \boldsymbol{g} - \boldsymbol{y}^j||_2^2 \right\}, \quad j = 1, ..., J, \tag{3.4}$$

where μ is the ℓ_1 regularization parameter. In our experiments we often sample μ from a uniform distribution for all calculations of $\check{\boldsymbol{f}}^j$ to simulate the ad-hoc proceedure for selecting typical regularization parameters. The sparsifying operator, $\mathcal{L}$, is designed so that the chosen solution is sparse in the edge domain. In this chapter we choose $\boldsymbol{\mathcal{L}}$ to be the mth order polynomial annihilation (PA), discussed in Chapter 2 and in [5, 4], and note that when $m = 1$ in (2.5) the method is equivalent to using total variation (TV). [1] To solve (3.4) we use the traditional alternating direction method of multipliers (ADMM) algorithm [88, 151, 54] discussed in Chapter 2 and described in Algorithm 2.

[1] Although there are subtle differences in the derivations and normalizations, the PA transform can be thought of as higher order total variation (HOTV). Because part of our investigation discusses parameter selection, which depends explicitly on $||\mathcal{L}f||$, we will exclusively use the PA transform as it appears in [4] so as to avoid any confusion. Explicit formulations for the PA transform matrix can be found in [4] and Chapter 2. We also note that the method can be easily adapted for other sparsifying transformations.

As shown in Figure 3.1(left), assuming that the model in (3.2) is correct, *any* of the reconstructed $\check{\boldsymbol{f}}^j$ (which we will refer to as the single measurement vector (SMV) reconstruction) should adequately approximate the underlying function f or any desired features of it. However, this may be impossible due to undersampling, noise, or bad information. Intuitively, using the redundant data from part of or all of the available data sets in (3.4) should lead to a better reconstruction algorithm. Indeed, many techniques have been developed to recover images from such multiple measurement vectors (MMV), [52, 31, 91, 33, 46, 47, 3, 120]. In our case the underlying function f is sparse in the edge domain, and so the collected set of recovered vectors is *jointly sparse* in the edge domain. The formal definition of joint sparsity is given by

Definition 3.2.2 *We say that*

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{p}^1 & \boldsymbol{p}^2 & \cdots & \boldsymbol{p}^J \end{bmatrix} \in \mathbb{R}^{N\times J}$$

is s-joint sparse if

$$||\boldsymbol{P}||_{p,0} = \left| \bigcup_{j=1}^{J} \text{supp}\left(\boldsymbol{p}^j\right) \right| \leq s,$$

where each $\boldsymbol{p}^j$ is s-sparse according to Definition 3.2.1.

For the variance based joint sparsity method in Algorithm 3, we also will assume that

$$\text{supp}(\boldsymbol{p}_1) \approx \text{supp}(\boldsymbol{p}_2) \approx \cdots \approx \text{supp}(\boldsymbol{p}_J), \tag{3.5}$$

that is, the joint sparsity of the vectors does not greatly exceed the sparsity of each individual vector.

To exploit the joint sparsity of the system, $\ell_{2,1}$ regularization is often applied, [162, 31, 135, 164]. Essentially, each vector is assumed to be sparse in its sparsity domain (e.g. edge domain), which motivates minimizing the ℓ_1 norm of each column. The "jointness" is accomplished by minimizing the ℓ_2 norm of each row (spatial elements).

The general joint sparsity technique using $\ell_{2,1}$ regularization is [135]

$$\hat{\boldsymbol{f}} = \left\{ \underset{\boldsymbol{z}\in\mathbb{R}^{N\times J}}{\operatorname{argmin}} \; ||\boldsymbol{\mathcal{L}}\boldsymbol{z}||_{2,1} \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{z} = \boldsymbol{Y} \right\}, \tag{3.6}$$

where $\mathcal{L}$ is the sparsifying transform matrix (here the PA transform (2.5) of order m), $\boldsymbol{Y} = [\boldsymbol{y}^1 \ \boldsymbol{y}^2 \ \cdots \ \boldsymbol{y}^J] \in \mathbb{R}^{M\times J}$ and $\boldsymbol{A} = \boldsymbol{A}^1 = \cdots = \boldsymbol{A}^J$. The solution $\hat{\boldsymbol{f}} = [\hat{\boldsymbol{f}}^1 \ \hat{\boldsymbol{f}}^2 \ \cdots \ \hat{\boldsymbol{f}}^J] \in \mathbb{R}^{N\times J}$ contains estimates for each measurement $\boldsymbol{y}^j$, $j = 1, ..., J$. It has been shown, both theoretically and in practice, that (3.6) yields improved approximations to *each* reconstruction in (3.4) when there is no misleading/false data, [31, 132, 164].

Note that (3.6) is typically solved using optimization techniques such as the ADMM, focal underdetermined system solvers (FOCUSS) and matching pursuit algorithms, [33]. [2] As demonstrated in Figure 3.1(middle), the joint sparsity approach using $\ell_{2,1}$ regularization is effective in cases where the data vectors are somewhat predictable, that is, when each measurement vector is determined from (3.2), and $\boldsymbol{A}^j$ is known. However, it is often the case when some of the acquired data do not have known sources. Worse, the information can be deliberately misleading, so that we assume we are acquiring $\boldsymbol{y}^j$ but in fact a completely different data set is obtained. We will refer to such a data set as a “rogue” vector. Figure 3.1(right) illustrates that in these situations, using (3.6) may be heavily influenced by the false measurements. [3] Hence we are motivated to find a technique that is able to discern “good” from “bad” information in the context of joint sparsity.

[2] We used the Matlab code provided in [37, 162] when implementing (3.6).

[3] For this simple example, the false measurements were formed by adding a false data point, with height sampled from the corresponding distribution, (binary, uniform or Gaussian) to $K = 5$ of the data vectors.

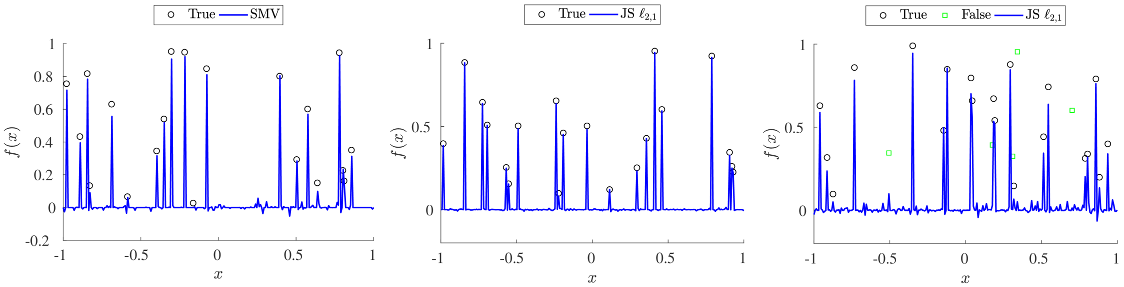

Figure 3.1: Sparse vector of uniformly distributed values on $[0, 1]$ reconstructed using (left) ℓ_1 regularization with a single measurement vector (SMV), (middle) $\ell_{2,1}$ regularization (3.6) applied on $J = 10$ true measurement vectors, and (right) $\ell_{2,1}$ regularization (3.6) applied to $J = 10$ measurement vectors, with 5 containing false data. In each case $N = 256$, $M = 100$ and $||\boldsymbol{f}||_0 = 20$ with $\boldsymbol{A}$ having i.i.d. Gaussian entries and $\mu = .25$ in (3.4). Plotted here is the average of the final 10 joint sparsity (JS) $\ell_{2,1}$ reconstructions.

3.3 Variance Based Joint Sparsity (VBJS)

Minimizing the effect of rogue measurement vectors consists of two parts. First, we must develop a technique to recognize points in the spatial domain where the measured data are inconsistent, and ensure that these regions of uncertainty do not have undue influence on the rest of the approximation. Second, we must have a way to identify the best reconstruction from the set of J solutions. With regard to the first, the *variance based weighted joint sparsity* (VBSJ) algorithm, developed in [2], can be adapted for the rogue measurement problem. The idea is described below.

We begin by gathering the (processed) measurements from (3.4) into a measurement matrix given by

$$\check{F} = \begin{bmatrix} \check{\boldsymbol{f}}^1 & \check{\boldsymbol{f}}^2 & \dots & \check{\boldsymbol{f}}^J \end{bmatrix} \in \mathbb{R}^{N \times J}. \tag{3.7}$$

We note that in most applications the initial data sets will come from (3.2), so it will be necessary to construct $\check{\boldsymbol{f}}^j$, $j = 1, \cdots, J$. Techniques other than (3.4) may be used for this purpose, however, and it might be sufficient to use a more cost efficient algorithm. Moreover, in some cases only one data vector is acquired, but is then processed in multiple (i.e. J) ways, with each processing providing different information. Indeed this was the case for one example discussed in [2], where one vector of Fourier data was collected but then several edge detection algorithms were used to construct jump function vectors (e.g. $\boldsymbol{y}^j$ in (3.4)). For ease of presentation, in this chapter we use the traditional interpretation of (3.2) followed by the computation of (3.4) for a given set of J measurement vectors to obtain (3.7), and leave these other cases to future work.

Next we define

$$\mathcal{P} = \begin{bmatrix} \mathcal{L}\check{\boldsymbol{f}}^1 & \mathcal{L}\check{\boldsymbol{f}}^2 & \cdots & \mathcal{L}\check{\boldsymbol{f}}^J \end{bmatrix} \in \mathbb{R}^{N\times J} \tag{3.8}$$

as the matrix of J vectors approximating some sparse feature of the underlying function f. For example, here $\mathcal{L}$ is the PA transform operator (2.5) so that $\mathcal{L}\check{\boldsymbol{f}}^j$ is an approximation of the edges of piecewise smooth f on the set of grid points given in (3.1). [4] Note that even if f is known explicitly, $\mathcal{L}\check{\boldsymbol{f}}^j$ will only be *approximately* zero in smooth regions, and hence is not truly sparse. However, the *behavior* of $\mathcal{L}\check{\boldsymbol{f}}^j$ should be consistent across all data sets, $j = 1, \cdots, J$, especially in smooth regions where $|\mathcal{L}\check{\boldsymbol{f}}^j|$ is small. This behavior should be confirmed in the variance vector $\check{\boldsymbol{v}} = (\check{v}_i)_{i=1}^N$, where each component is given by

$$\check{v}_i = \frac{1}{J}\sum_{j=1}^{J} \mathcal{P}_{i,j}^2 - \left(\frac{1}{J}\sum_{j=1}^{J} \mathcal{P}_{i,j}\right)^2, \quad i = 1, ..., N. \tag{3.9}$$

That is, (3.9) should yield small values in smooth regions when the data measurements are consistent. Note that $\text{supp}(\check{\boldsymbol{v}}) \approx \bigcup_{j=1}^{J} \text{supp}(\mathcal{L}\check{\boldsymbol{f}}_j)$ and in our experiments we

[4] Specifically it approximates the jump function $[f](x) = f(x^+) - f(x^-)$ on a set of N grid points.

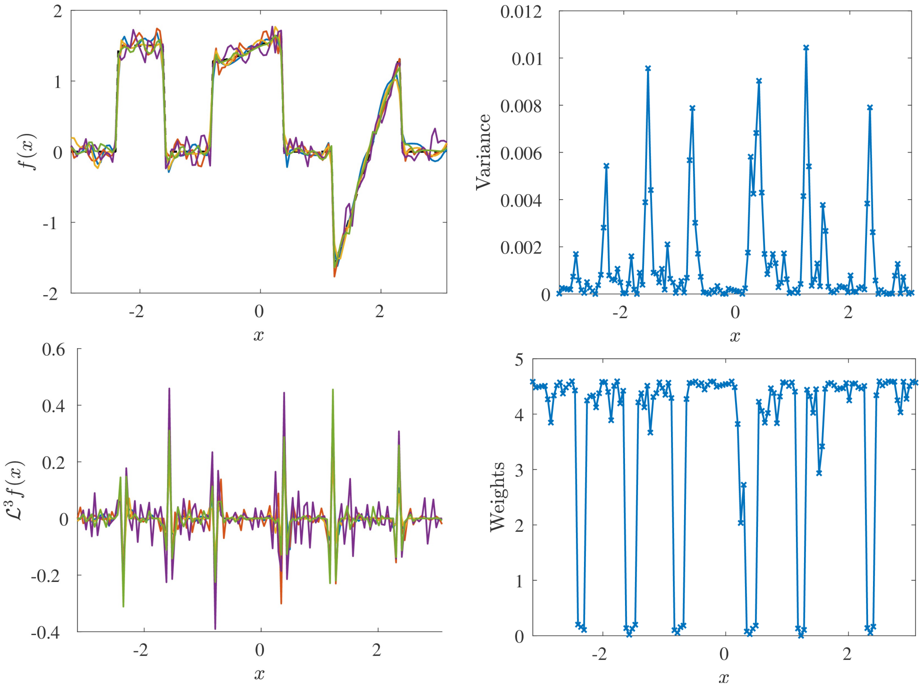

Figure 3.2: (top-left) Five measurements of the underlying function in Example 3.5.1, acquired using (3.4). (bottom-left) Corresponding five sparsity vectors (3.8) with order $m = 3$. (top-right) The variance of the sparsity vectors calculated using (3.9). (bottom-right) The corresponding weights calculated as in (3.12).

consider.

We will exploit (3.9) in determining how the joint sparsity algorithm should be regularized. Figure 3.2 demonstrates how this may be useful. Five measurement vectors of the function in Example 3.5.1, where $\boldsymbol{A}$ has i.i.d. entries sampled from a uniform distribution on $[0, 1]$ and the noise is Gaussian with mean zero and variance 0.1, is shown in the top left. The bottom left displays the corresponding sparsity vectors, $\boldsymbol{\mathcal{L}}\check{\boldsymbol{f}}^j$. Observe that the variance of the sparsity vectors, provided in the top right, is spatially variant, with the larger values occuring near the jump discontinuities as well as where more noise is apparent in the data measurements. This suggests that a spatially variant (weighted) ℓ_1 norm might work better than the uniform $\ell_{2,1}$ norm in regularizing the joint sparsity approximation. Algorithm 3 describes this process.

Algorithm 3 Variance-Based Joint Sparsity algorithm

```
1: Recover the vectors $\check{\boldsymbol{f}}^j$, $j = 1, \dots, J$, separately using (3.4) to obtain (3.7).
2: Compute the variance of $\mathcal{L}\check{\boldsymbol{f}}^j$, $j = 1, \cdots, J$, using (3.9).
3: Use the results from (3.9) to determine the weights for the weighted $\ell_p$ norm,
   $1 \leq p \leq 2$, in the joint sparsity reconstruction. In particular, $\check{v}_i$ should be large
   when the index $i$ belongs to the support of $\check{\boldsymbol{v}}$, while $\check{v}_i \approx 0$ otherwise. Hence we
   compute a vector of nonnegative weights $\boldsymbol{w} = (w_i)_{i=1}^N$, $0 \leq w_i \leq C$, $C \in \mathbb{R}$ based
   on this information. In general, $w_i \approx 0$ when $\check{v}_i$ is large and $w_i \approx C$ when $\check{v}_i \approx 0$.
   The weights we design for this purpose are provided in (3.12).
4: Determine data vector $\hat{\boldsymbol{y}} \in \{\boldsymbol{y}^j | j = 1, \cdots, J\}$, and corresponding matrix $\hat{\boldsymbol{A}}$ that
   will be used as the "best" initial vector approximation. This is done according to
   (3.15) and (3.16).
5: Solve the weighted $\ell_p$ minimization problem to get the final reconstruction of the
   vector $\boldsymbol{f}$:
   $$\hat{\boldsymbol{g}} = \underset{\boldsymbol{g} \in \mathbb{R}^N}{\operatorname{argmin}} \left\{ \frac{1}{p} ||\mathcal{L}\boldsymbol{g}||_{p,\boldsymbol{w}}^p + \frac{\mu}{2} ||\hat{\boldsymbol{A}}\boldsymbol{g} - \hat{\boldsymbol{y}}||_2^2, \right\} \tag{3.10}$$
   for $\mu > 0$ a constant parameter.
```

Remark 3.3.1 *Observe that in contrast to (3.6), any $p \in \{1, 2\}$ can be used in Step 5 of Algorithm 3. While $p = 1$ is consistent with compressive sensing techniques, a spatially variant weighting vector may relax the requirements on p while still achieving the goal of sparsity. Intuitively, using ℓ_1 effectively promotes sparsity because of the higher penalty placed on small values in the reconstruction of what is presumably sparse (e.g. edges of piecewise smooth f), as compared to the standard ℓ_2 minimization, which imposes a penalty proportional to the square of each value in the reconstructed edge vector. Employing a (spatially variant) weighted ℓ_2 minimization* *designed to more strongly enforce small values in sparse regions should yield the same desired property for promoting sparsity. Moreover, using $||\cdot||_{2,\boldsymbol{w}}$ will be much more efficient numerically, because a closed form gradient of the objective function is available. A complete characterization of ℓ_1 and weighed ℓ_2 minimizers can be found in [36].*

3.3.1 *Weight Design*

In contrast to (3.6), where each grid point in the sparsity domain is equally weighted in the regularization term, Algorithm 3 uses a spatially variant regularization, with the weights $(w_i)_{i=1}^N$ being inherently linked to (3.9). In particular, because small variance values strongly suggest joint sparsity in the sparsity domain, the associated values $|\mathcal{L}\boldsymbol{f}_i|$, where $\boldsymbol{f}_i \approx f(x_i)$ of the underlying function and $\mathcal{L}$ is the sparsifying transform operator, should be heavily penalized in the regularization term. On the other hand, large variance values may indicate that the the corresponding indices belong to the support of the function (or image) in the sparsity domain. Large variance values may also indicate unreliable information at that particular spatial grid point. Hence $|\mathcal{L}\boldsymbol{f}_i|$ should be penalized less at those indices when minimizing the regularization term. Figure 3.2 (bottom right) depicts the weights chosen by (3.12) to minimize the weighted ℓ_p norm in (3.10).

From the discussion above and illustrated in Figure 3.2, we see that the weights for the regularization term should not depend on how the measurements in (3.4) are constructed, but rather only the expectation that they be jointly sparse in the same domain, as defined in Definition 3.2.2. In our examples, we assume that this joint sparsity occurs in the edge domain. The variance calculated in (3.9) provides a means of determining the actual joint sparsity, and moreover provides us a way to reduce the effects of bad data.

To determine the specific weighting vector $\boldsymbol{w}$, we begin by defining a weighting scalar as the average ℓ_1 norm across all measurements of the normalized sparsifying transform of our measurements. This will enable us to scale the weights according to the magnitude of the values in the sparsity domain, and ultimately reduce the need for fine tuning regularization parameters in the numerical implementation. In our investigation we use the PA transformation (2.5) to approximate the edges of the underlying function or image, so we scale the weights according to the spatially variant jump height of our solutions. Thus we define

$$\tilde{\mathcal{P}} = \begin{bmatrix} \tilde{\mathcal{P}}_1 & \tilde{\mathcal{P}}_2 & \cdots & \tilde{\mathcal{P}}_J \end{bmatrix} \in \mathbb{R}^{N\times J}$$

as the normalized PA transform matrix from (3.8), where

$$\tilde{\mathcal{P}}_{i,j} = \frac{|\mathcal{P}_{i,j}|}{\max\limits_i |\mathcal{P}_{i,j}|}, \quad j = 1, ..., J.$$

We also define the weighting scalar as

$$C = \frac{1}{J}\sum_{j=1}^{J}\sum_{i=1}^{N}\tilde{\mathcal{P}}_{i,j}. \tag{3.11}$$

Finally, we construct the weight vector $\boldsymbol{w}$ element-wise as

$$w_i = \begin{cases} C\left(1 - \frac{v_i}{\max_i v_i}\right), & i \notin I \\ \frac{1}{C}\left(1 - \frac{v_i}{\max_i v_i}\right), & i \in I \end{cases} \tag{3.12}$$

where I consists of the indices i such that

$$\frac{1}{J}\sum_{j=1}^{J}\tilde{\mathcal{P}}_{i,j} > \tau. \tag{3.13}$$

Here τ is a threshold chosen so that when (3.13) is satisfied, we assume there is a corresponding edge at x_i, and that the index i is part of the support in the sparse domain of $\boldsymbol{f}$. Because the jumps are normalized, it is reasonable for $\tau = \mathcal{O}(\frac{1}{N})$, that is, τ is resolution dependent. Because there is noise in the system, we choose $\tau > \frac{1}{N}$, and in our examples $\tau = .1$, and note that if more is known apriori about the size of the noise, then τ can be chosen accordingly. In general as τ increases, more noise is assumed to be in the system, which corresponds to a more uniform weighting scheme. Choosing weights based on information about system noise and nuisance parameters will be addressed more in future investigations.

Observe that $w_i \in [0, C]$, $i = 1, \cdots, N$ and $C > 1$. The weighting scalar C defined in (3.11) allows the regularization to better account for functions that contain multiple edges with different magnitudes. Specifically, the weights in (3.12) are designed to scale the penalty of the regularization according to the size of the jump, with the largest weights being reserved for regions where the function is presumably smooth. The intuition used for determining the weights formula in (3.12) is illustrated in Figure 3.2. In this case we have $J = 5$ measurements for Example 3.5.1. We use the PA transform in (3.8) with order $m = 3$, and $\mu = .25$ in (3.4). [5]

For comparative purposes, we will also consider weights that were used in [2]

$$w_i = \frac{1}{v_i + \epsilon}, \tag{3.14}$$

where ϵ is a small parameter chosen to avoid dividing by zero. In [2] it was demonstrated that this weighting strategy was robust in sparse signal recovery (in the noiseless case) for $\epsilon = 10^{-2}$.

3.3.2 Determining the Optimal Solution Vector

The traditional $\ell_{2,1}$ method that exploits the joint sparsity of J multi-measurement vectors (MMV) in (3.6) recovers J solution vectors. This is also the case in [2], however

[5] It was observed in [21] that multiple scales in jump heights can be handled by iteratively redefining a weighed $\ell_{2,1}$ norm in the MMV case (3.6). This method proved to be computationally expensive, as the optimization problem must be resolved at each iteration, however.

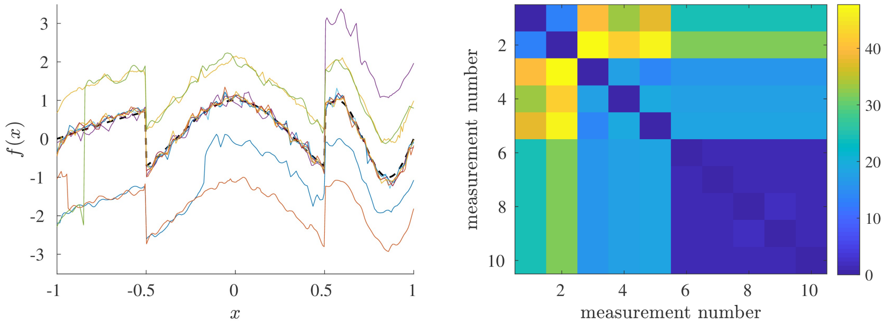

Figure 3.3: (left) Five false measurements and five true measurements of Example 3.5.2. The true underlying function is displayed as the bold dashed line. (right) The corresponding construction of the distance matrix $\mathcal{D}$ in (3.15).

we are only interested in one "best" solution, as described in Algorithm 3. Moreover, we want to avoid using any bad information or rogue vectors as the base of our solution. Therefore, we choose the final data vector $\hat{\boldsymbol{y}}$ in Step 4 of Algorithm 3 to be one whose corresponding measurements are closest to most of the other measurement vectors in the set of J vectors. Thus we define the distance matrix $\mathcal{D}$ with entries

$$\mathcal{D}_{i,j} = ||\check{\boldsymbol{f}}^i - \check{\boldsymbol{f}}^j||_2, \tag{3.15}$$

where each $\check{\boldsymbol{f}}$ is defined in (3.4). The data vector $\hat{\boldsymbol{y}} = \boldsymbol{y}^{j^*}$ and forward operator $\hat{\boldsymbol{A}} = \boldsymbol{A}^{j^*}$ correspond to the j^*th index that solves

$$j^* = \min_{j \neq i} \mathcal{D}_{i,j}, \quad i = 1, \cdots, J. \tag{3.16}$$

An example of this process is depicted in Figure 3.3. On the left we see ten measurements of Example 3.5.2 where the first five measurements are false measurements. Displayed on the right is the matrix $\mathcal{D}$ given in (3.15). Assuming that the number of true measurement vectors $J - K$ is greater than 2, it is reasonable to use (3.16) to determine the "best" data vector for the final reconstruction. It must also be true that rogue data vectors are not similar to one another. That is, for all $i, j = 1, \cdots, K$,

$||\check{\boldsymbol{f}}^i - \check{\boldsymbol{f}}^j|| > \sigma$ where $\sigma > 0$ is a chosen distance threshold. The quality of the solution is clearly dependent on the number of rogue measurements in the collection set. More analysis is needed to determine the relationship between the ratio of false and true measurements and the success of Algorithm 3, and will be the subject of future work.

3.4 Efficient Implementation of the VBJS Algorithm

After determining initial solutions, (3.4), the weighting vector, (3.12), and the most suitable vector for reconstruction given in (3.16), we can now approximate the solution $\hat{\boldsymbol{g}}$ to (3.10) in Algorithm 3. When using the weights designed in (3.12), we eliminate the need to tune the parameter μ to ensure convergence, and thus we set $\mu = 1$ in (3.10) for our experiments.

For $x \in \mathbb{R}^N$, the weighted ℓ_p norm is defined as

$$||x||_{p,\boldsymbol{w}} = \left(\sum_{i=1}^{N} w_i |x_i|^p \right)^{1/p} = ||Wx||_p, \tag{3.17}$$

where $W = \text{diag}(\boldsymbol{w}) \in \mathbb{R}^{N \times N}$. With this definition we can now solve (3.10) using stardard ℓ_p minimization techniques, see e.g. [88, 54, 151, 156].

In two dimensions ($x \in \mathbb{R}^{N \times N}$), especially as the number of data points N increase, it quickly becomes undesirable to write the weights as a diagonal matrix. That is, even though "stacking" the columns (noted by the *vec* function) holds intuitive appeal for solving (3.10), because $W = \text{diag}(vec(\boldsymbol{w})) \in \mathbb{R}^{N^2 \times N^2}$, it would be ideal to not multiply by this dense matrix.

Fortunately, however, we are able to show that the ADMM algorithm can also be applied in this case, as will be described below. For this purpose we first define the weighted ℓ_p norm as

$$||x||_{p,\boldsymbol{w}}^p = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{i,j} |x_{i,j}|^p, \tag{3.18}$$

where $w_{i,j}$ are elements of $\boldsymbol{w} \in \mathbb{R}^{N \times N}$ and $x \in \mathbb{R}^{N \times N}$.

3.4.1 The ADMM Algorithm for the ℓ_1 Case

We now demonstrate how the ADMM can be efficiently applied to solve (3.10) when $p = 1$. While the algorithm can be used for either the one- or two-dimensional case, for computational efficiency, such an approach is desirable for two-dimensional problems.

To start, we write (3.10) with $\mu = 1$ as the equivalent non-parametric weighted ℓ_1 problem

$$(\hat{\boldsymbol{g}}, \hat{\boldsymbol{z}}) = \left\{ \underset{\boldsymbol{g},\boldsymbol{z}}{\operatorname{argmin}} \ ||\boldsymbol{z}||_{1,\boldsymbol{w}} + \frac{1}{2}||\hat{\boldsymbol{A}}\boldsymbol{g} - \hat{\boldsymbol{y}}||_2^2 \quad \text{subject to} \quad \boldsymbol{\mathcal{L}}\boldsymbol{g} = \boldsymbol{z} \right\}. \tag{3.19}$$

Here we assume $\hat{\boldsymbol{A}}, \boldsymbol{g}, \boldsymbol{z}$ and $\hat{\boldsymbol{y}}$ are all in $\mathbb{R}^{N \times N}$, and because of the non-differentiability in the $\ell_{1,\boldsymbol{w}}$ norm and the non-seprability of $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{g}$ we have introduced slack variables $\boldsymbol{z} \in \mathbb{R}^{N \times N}$. To solve (3.19), we minimize its corresponding augmented Lagrangian function

$$\underset{\boldsymbol{g},\boldsymbol{z}}{\operatorname{argmin}} \left\{ ||\boldsymbol{z}||_{1,\boldsymbol{w}} - \boldsymbol{\nu}^T vec\left(\boldsymbol{\mathcal{L}}\boldsymbol{g} - \boldsymbol{z}\right) + \frac{\beta}{2}||\boldsymbol{\mathcal{L}}\boldsymbol{g} - \boldsymbol{z}||_2^2 + \frac{1}{2}||\hat{\boldsymbol{A}}\boldsymbol{g} - \hat{\boldsymbol{y}}||_2^2 \right\}, \tag{3.20}$$

where $\boldsymbol{\nu} \in \mathbb{R}^{N^2}$ is the Lagrangian multiplier.

Remark 3.4.1 *Two parameters, μ from (3.10) and β in (3.20), typically must be prescribed in ADMM. In (3.19) we observe that we can use $\mu = 1$ because the weighting of this term is considered in the construction of the weighting vector (3.12). We also note that although we have not formally analyzed the impact of using the weighted ℓ_1 norm on the overall rate of convergence, our numerical experiments demonstrate that choosing $\beta = 1$ yields reasonably fast convergence. A study of how the weighting vector affects the convergence rate for different choices of β will be the subject of future investigations. Thus we see that the ADMM method for VBJS is robust, as no fine tuning of parameters is needed at the optimization stage.*

The problem is now split into two sub-problems, known as the *z-subproblem* and the *g-subproblem.*

The z-*subproblem*

To analyze the *z-subproblem*, we assume that the values of $\boldsymbol{g}$ and $\boldsymbol{\nu}$ are known and fixed and set $\beta = 1$ in (3.20), so that

$$\hat{\boldsymbol{z}} = \underset{\boldsymbol{z}}{\operatorname{argmin}} \left\{ ||\boldsymbol{z}||_{1,\boldsymbol{w}} - \boldsymbol{\nu}^T vec\left(\boldsymbol{\mathcal{L}g} - \boldsymbol{z}\right) + \frac{1}{2}||\boldsymbol{\mathcal{L}g} - \boldsymbol{z}||_2^2 \right\}. \tag{3.21}$$

Lemma 3.4.2 demonstrates that a closed form solution exists in general for the z-subproblem for any $\beta > 0$.

Lemma 3.4.2 *For a given $\beta > 0$, $x, y \in \mathbb{R}^{N\times N}$ and $\nu \in \mathbb{R}^{N^2}$, the minimizer of*

$$\underset{x}{\operatorname{argmin}} \left\{ ||x||_{1,w} - \nu^T vec(y - x) + \frac{\beta}{2}||y - x||_2^2 \right\} \tag{3.22}$$

is given by the shrinkage-like formula

$$\hat{x} = \max\left\{ \left| y - \frac{\nu}{\beta} \right| - \frac{w}{\beta}, 0 \right\} sign\left(y - \frac{\nu}{\beta} \right). \tag{3.23}$$

Proof 3.4.3 *Following the technique described in [88] for the non-weighted, one-dimensional case, let $x \in \mathbb{R}^{N\times N}$ and $w_{i,j} \geq 0$ for all $i, j = 1, ..., N$. We drop the vec notation for simplicity.*

Define the objective function $H : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ as

$$H(x) := ||x||_{1,w} - \nu^T(y - x) + \frac{\beta}{2}||y - x||_2^2. \tag{3.24}$$

To show $H(x)$ is convex, we first observe that for $\alpha \in (0, 1)$ and $p, q \in \mathbb{R}^{N\times N}$, we *have*

$$
\begin{aligned}
&||y - \alpha p - (1-\alpha)q||_2^2 - \big(\alpha||y-p||_2^2 + (1-\alpha)||y-q||_2^2\big) \\
&= (y - \alpha p - (1-\alpha)q)^T\,(y - \alpha p - (1-\alpha)q) - \big(\alpha(y-p)^T(y-p) + (1-\alpha)(y-q)^T(y-q)\big) \\
&= \alpha(\alpha-1)\left(p^Tp - p^Tq - q^Tp + q^Tq\right) \\
&= \alpha(\alpha-1)||p-q||_2^2 \\
&\leq 0.
\end{aligned}
\tag{3.25}
$$

Applying (3.25) to H yields

$$
\begin{aligned}
&H(\alpha p + (1-\alpha)q) - (\alpha H(p) + (1-\alpha)H(q)) \\
&= ||\alpha p + (1-\alpha)q||_{1,w} - \nu^T\,(y - (\alpha p + (1-\alpha)q)) + \frac{\beta}{2}||y - (\alpha p + (1-\alpha)q)||_2^2 \\
&- \alpha||p||_{1,w} - (1-\alpha)||q||_{1,w} + \alpha\nu^T(y-p) + (1-\alpha)\nu^T(y-q) - \frac{\beta\alpha}{2}||y-p||_2^2 - \frac{\beta(1-\alpha)}{2}||y-q||_2^2 \\
&\leq \frac{\beta}{2}||y - (\alpha p + (1-\alpha)q)||_2^2 - \frac{\beta\alpha}{2}||y-p||_2^2 - \frac{\beta(1-\alpha)}{2}||y-q||_2^2 \\
&= \frac{\beta}{2}\alpha(\alpha-1)||p-q||_2^2 \\
&\leq 0.
\end{aligned}
\tag{3.26}
$$

Therefore H is convex. For $p \neq q$, H is strictly/strongly convex and thus coercive [109, 12, 11]. Hence there exists at least one solution $\hat{x}$ of (3.22), [156].

The subdifferential of $f(x) = ||x||_{1,w}$ is given element-wise as

$$
(\partial_x f(x))_{i,j} = \begin{cases} sign(x_{i,j})w_{i,j}, & x_{i,j} \neq 0 \\ \left\{h; |h| \leq w_{i,j},\ h \in \mathbb{R}\right\}, & \textit{otherwise}, \end{cases}
\tag{3.27}
$$

where the origin is required to be included according to the optimality condition for convex problems. According to (3.27), to minimize (3.24), each component $\hat{x}_{i,j}$, $i, j =$

$1, ..., N$, must satisfy

$$\begin{cases} sign(\hat{x}_{i,j})w_{i,j} + \beta(\hat{x}_{i,j} - y_{i,j}) + \nu_{i,j} = 0, & x_{i,j} \neq 0 \\ |v_{i,j} - \beta y_{i,j}| \leq w_{i,j}, & otherwise. \end{cases} \tag{3.28}$$

If $\hat{x}_{i,j} \neq 0$, (3.28) yields

$$\frac{w_{i,j}}{\beta} sign(\hat{x}_{i,j}) + \hat{x}_{i,j} = y_{i,j} - \frac{\nu_{i,j}}{\beta}. \tag{3.29}$$

Because $w_{i,j}/\beta > 0$, (3.29) implies

$$\frac{w_{i,j}}{\beta} + |\hat{x}_{i,j}| = |y_{i,j} - \frac{\nu_{i,j}}{\beta}|. \tag{3.30}$$

Combining (3.29) and (3.30) gives

$$\begin{aligned} sign(\hat{x}_{i,j}) &= \frac{sign(\hat{x}_{i,j})|\hat{x}_{i,j}| + sign(\hat{x}_{i,j})w_{i,j}/\beta}{|\hat{x}_{i,j}| + w_{i,j}/\beta} = \frac{\hat{x}_{i,j} + sign(\hat{x}_{i,j})w_{i,j}/\beta}{|\hat{x}_{i,j}| + w_{i,j}/\beta} \\ &= \frac{y_{i,j} - \nu_{i,j}/\beta}{|y_{i,j} - \nu_{i,j}/\beta|} = sign\left(y_{i,j} - \frac{\nu_{i,j}}{\beta}\right) \end{aligned} \tag{3.31}$$

Thus, for $\hat{x}_{i,j} \neq 0$, we have

$$\hat{x}_{i,j} = |\hat{x}_{i,j}| sign(\hat{x}_{i,j}) = \left(|y_{i,j} - \frac{\nu_{i,j}}{\beta}| - \frac{w_{i,j}}{\beta}\right) sign\left(y_{i,j} - \frac{\nu_{i,j}}{\beta}\right), \tag{3.32}$$

where we have used (3.30) and (3.31) in the result.

Conversely, we now show that $\hat{x}_{i,j} = 0$ if and only if

$$|y_{i,j} - \frac{\nu_{i,j}}{\beta}| \leq \frac{w_{i,j}}{\beta}. \tag{3.33}$$

First assume that $\hat{x}_{i,j} = 0$. Then (3.33) follows from (3.28) because $\beta > 0$.

Now assume (3.33) holds for some $\hat{x}_{i,j} \neq 0$. By (3.28), $\hat{x}_{i,j}$ satisfies (3.30). Hence

$$|\hat{x}_{i,j}| = |y_{i,j} - \frac{\nu_{i,j}}{\beta}| - \frac{w_{i,j}}{\beta} \leq 0$$

which only holds for $\hat{x}_{i,j} = 0$. Hence by contradiction, $\hat{x}_{i,j} = 0$. Combining (3.33) with (3.32) yields

$$\hat{x}_{i,j} = \max\left\{|y_{i,j} - \frac{\nu_{i,j}}{\beta}| - \frac{w_{i,j}}{\beta}, 0\right\} sign\left(y_{i,j} - \frac{\nu_{i,j}}{\beta}\right).$$

which is equivalent to (3.23) in matrix form.

In light of Lemma 3.4.2, the closed form solution to (3.21) is given as

$$\hat{\boldsymbol{z}} = \max\left\{|\boldsymbol{\mathcal{L}g} - \boldsymbol{\nu}| - \boldsymbol{w}, 0\right\} \operatorname{sign}\left(\boldsymbol{\mathcal{L}g} - \boldsymbol{\nu}\right). \tag{3.34}$$

The *g-subproblem*

Once the *z-subproblem* is solved, we can proceed using standard ADMM. Specifically, $\boldsymbol{z}$ and $\boldsymbol{\nu}$ are held fixed, $\beta = 1$ in (3.20), and we construct *g-subproblem* from (3.20) as

$$\hat{\boldsymbol{g}} = \underset{\boldsymbol{g}}{\operatorname{argmin}}\ J(\boldsymbol{g}) := \left\{\frac{1}{2}||\boldsymbol{\mathcal{L}g} - \boldsymbol{z}||_2^2 + \frac{1}{2}||\hat{\boldsymbol{A}}\boldsymbol{g} - \hat{\boldsymbol{y}}||_2^2 - \boldsymbol{\nu}^T vec\left(\boldsymbol{\mathcal{L}g} - \boldsymbol{z}\right)\right\}. \tag{3.35}$$

We solve (3.35) using gradient descent, [88, 151, 54],

$$\boldsymbol{g}_{k+1} = \boldsymbol{g}_k - \alpha_k \nabla_{\boldsymbol{g}} J(\boldsymbol{g}_k), \tag{3.36}$$

where

$$\nabla_{\boldsymbol{g}} J(\boldsymbol{g}_k) = -\boldsymbol{\nu}^T\boldsymbol{\mathcal{L}} + (\boldsymbol{\mathcal{L}})^T(\boldsymbol{\mathcal{L}g} - \boldsymbol{z}) + \hat{\boldsymbol{A}}^T(\hat{\boldsymbol{A}}\boldsymbol{g} - \hat{\boldsymbol{y}}). \tag{3.37}$$

Note that for ease of presentation we have again dropped the *vec* notation, although it is of course needed for implementation. The step length is chosen as the Barzilai-Borwein (BB) step (see [10]),

$$\alpha_k = \frac{\boldsymbol{s}_k^T\boldsymbol{s}_k}{\boldsymbol{s}_k^T\boldsymbol{u}_k}, \tag{3.38}$$

with

$$\boldsymbol{s}_k = \boldsymbol{g}_k - \boldsymbol{g}_{k-1}$$

$$\boldsymbol{u}_k = \nabla_{\boldsymbol{g}} J(\boldsymbol{g}_k) - \nabla_{\boldsymbol{g}} J(\boldsymbol{g}_{k-1}).$$

A backtracking algorithm is performed to ensure α_k is not chosen to be too large. This requires checking what is known as the Armijo condition, [156], which guarantees that using (3.38) sufficiently reduces the magnitude of the objective function. Algorithmically, the Armijo condition is given by

$$J(\boldsymbol{g}_k - \alpha\nabla_{\boldsymbol{g}} J(\boldsymbol{g}_k)) \leq J(\boldsymbol{g}_k) - \delta\alpha_k \nabla_{\boldsymbol{g}}^T J(\boldsymbol{g}_k)\nabla_{\boldsymbol{g}} J(\boldsymbol{g}_k), \tag{3.39}$$

where $\delta \in (0,1)$. If the Armijo condition (3.39) is not satisfied, we backtrack and decrease the step length according to

$$\alpha_k = \rho\alpha_k,$$

where $\rho \in (0,1)$ is the backtracking parameter. At the kth iteration of the algorithm, after the new $\boldsymbol{z}$ and $\boldsymbol{g}$ values are found using (3.34) and (3.35), the Lagrange multiplier is updated according to

$$\boldsymbol{\nu}_{k+1} = \boldsymbol{\nu}_k - vec(\boldsymbol{\mathcal{L}g}_{k+1} - \boldsymbol{z}_{k+1}) \tag{3.40}$$

Algorithm 4 provides the weighted version of the ADMM. The technique involves alternating solving the z-subproblem (3.21) and g-subproblem (3.35) at each iteration. Typical parameter choices are $\rho = .4$ and $\delta = 10^{-4}$, [88, 156].

Algorithm 4 Weighted ADMM

```
Initialize $\boldsymbol{\nu}_0$. Determine weights $\boldsymbol{w}$, starting points $\boldsymbol{g}_0$ and $\boldsymbol{z}_0$ and maximum number of iterations $K$.
for $i = 0$ to $K$ do
    Set $0 < \rho, \delta < 1$ and tolerance $tol$.
    while $||\boldsymbol{g}_{k+1} - \boldsymbol{g}_k|| > tol$ do
        Compute $\boldsymbol{z}_{k+1}$ using (3.34).
        Set $\alpha_k$ using (3.38).
        while Armijo condition (3.39) unsatisfied do
            Backtrack: $\alpha_k = \rho\alpha_k$.
        end while
        Compute $\boldsymbol{g}_{k+1}$ using (3.36) and (3.37).
    end while
    Update Lagrange multiplier according to (3.40).
end for
```

3.4.2 Efficient Implementation for the ℓ_2 Case

When $p = 2$ in (3.10) we solve

$$\hat{\boldsymbol{g}} = \underset{\boldsymbol{g}}{\operatorname{argmin}} \; J(\boldsymbol{g}) := \left\{ \frac{1}{2} ||\boldsymbol{\mathcal{L}} \boldsymbol{g}||_{2,\boldsymbol{w}}^2 + \frac{1}{2} ||\hat{\boldsymbol{A}} \boldsymbol{g} - \hat{\boldsymbol{y}}||_2^2 \right\} \tag{3.41}$$

using the gradient descent method defined in (3.36). However, some care must be taken to derive the gradient of the first term of (3.41). According to (3.18), for $\mathcal{L}, \boldsymbol{g}, \boldsymbol{w} \in \mathbb{R}^{N \times N}$,

$$||\boldsymbol{\mathcal{L}} \boldsymbol{g}||_{2,w}^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{i,j} \left(\sum_{k=1}^{N} \mathcal{L}_{i,k} g_{k,j} \right)^2 . \tag{3.42}$$

Taking the derivative of (3.42) with respect to an element of $\boldsymbol{g}$ yields

$$\frac{\partial}{\partial g_{k,j}} ||\boldsymbol{\mathcal{L}} \boldsymbol{g}||_{2,\boldsymbol{w}}^2 = 2 \sum_{i=1}^{N} w_{i,j} \mathcal{L}_{i,k} \left(\sum_{l=1}^{N} \mathcal{L}_{i,l} g_{l,j} \right), \quad k, j = 1, ..., N.$$

Performing this operation over all $k, j = 1, ..., N$, produces

$$\nabla_{\boldsymbol{g}} ||\boldsymbol{\mathcal{L}} \boldsymbol{g}||_{2,\boldsymbol{w}}^2 = 2 \mathcal{L}^T \left[\boldsymbol{w} \odot (\boldsymbol{\mathcal{L}} \boldsymbol{g}) \right], \tag{3.43}$$

where $\odot$ denotes the pointwise Hadamard product. Thus, the gradient of the objective function J in (3.41) is given by

$$\nabla_{\boldsymbol{g}} J(\boldsymbol{g}) = \mathcal{L}^T \left[\boldsymbol{w} \odot (\boldsymbol{\mathcal{L}} \boldsymbol{g}) \right] + \hat{\boldsymbol{A}}^T (\hat{\boldsymbol{A}} \boldsymbol{g} - \hat{\boldsymbol{y}}). \tag{3.44}$$

Using (3.44) in (3.36) with the BB step length (3.38), we can now solve (3.41) for $\hat{\boldsymbol{g}}$. The weighted ℓ_2 gradient descent process is described in Algorithm 5. Typical parameter choices again are $\rho = .4$ and $\delta = 10^{-4}$ and a starting step length of $\alpha_0 = 1$ is chosen to initiate the algorithm [156].

Algorithm 5 Weighted Gradient Descent

```
Initialize starting points $\boldsymbol{g}_0$ and $\alpha_0$, parameters $\delta, \rho \in (0,1)$ and tolerance $tol$.
Determine weights $\boldsymbol{w}$.
while $||\boldsymbol{g}_{k+1} - \boldsymbol{g}_k|| > tol$ do
    Set $\alpha_k$ using (3.38).
    while Armijo condition (3.39) unsatisfied do
        Backtrack: $\alpha_k = \rho\alpha_k$.
    end while
    Compute $\boldsymbol{g}_{k+1}$ using (3.36) and (3.44).
end while
```

3.5 Numerical Results

We test the variance based joint sparsity (VBJS) technique in three different situations and compare our method in Algorithm 4 and 5 to the typical $\ell_{2,1}$ minimization algorithm in (3.6), the SMV case, and the VBJS method with weighted given as (3.14). In our experiments we employ both ℓ_1 and ℓ_2 regularization in (3.10) with $\mu = 1$, demonstrating the accuracy and robustness of our methods in each case. As was shown in [2], the VBJS method is consistently more cost efficient than $\ell_{2,1}$ regularization. Moreover, using weighted ℓ_2 regularization is clearly less costly than using weighted ℓ_1.

First we consider recovering sparse signals. A similar experiment was performed for VBSJ in [2] on noiseless data. In our example the measurement vectors contain noise, and there are also measurements that contain false information. In this regard it is important to note that the weights in (3.12) are designed so that *no* additional parameters are needed in (3.20). That is, $\beta = 1$ in the *z-subproblem* and regularization parameters normally included in the ADMM *g-subproblem* are not needed [88].

However, this is not the case when using (3.14), where we will see that regularization parameters are needed to obtain any meaningful results. As noted previously, to obtain the first measurements in each algorithm, we use (3.4) with μ sampled from a uniform distribution for each $j = 1, \cdots, J$, thus simulating the ad-hoc procedure for selecting typical regularization parameters.

For the second experiment we consider two one-dimensional signals that exhibit sparsity in the edge domain. We apply the VBJS technique for both $p = 1$ and 2 in the weighted ℓ_p regularization, and again compare our method to techniques in [2] with (3.14). In our third test we reconstruct two-dimensional images with sparse edges.

3.5.1 Sparse Signal Recovery

We seek to recover the sparse signal $\boldsymbol{f}$ from a set of measurment vectors. This problem has been widely studied within the context of MMV, [33, 46, 47]. An adaptively weighted ℓ_1 reconstruction method was developed in [21] for the single measurement vector (SMV) case, and the VBJS method using the weights in (3.14) was developed for MMV in [2]. In this case each data vector $\{\boldsymbol{y}^j\}_{j=1}^J$ in (3.2) is acquired using a measurement matrix $\boldsymbol{A} \in \mathbb{R}^{N \times M}$ where each element of $\boldsymbol{A}$ is sampled independently from a zero mean unit variance Gaussian distribution. The corresponding noise vectors $\{\boldsymbol{\eta}^j\}_{j=1}^J$ are i.i.d. Gaussian with zero mean and unit variance. Of the J measurements, K contain false information and in some cases are complete misrepresentations of the underlying signal. To recover the sparse signal $\boldsymbol{f}$ we used (3.17) with $p = 1$ in Algorithm 4. Because the $J - K$ true measurements have overlapping support, we use the PA transform with order $m = 0$, that is $\boldsymbol{\mathcal{L}} = \boldsymbol{I}$ in the sparsity regularization term of (3.10).

Figure 3.4 compares the signal recovery results for three sparse signals using the

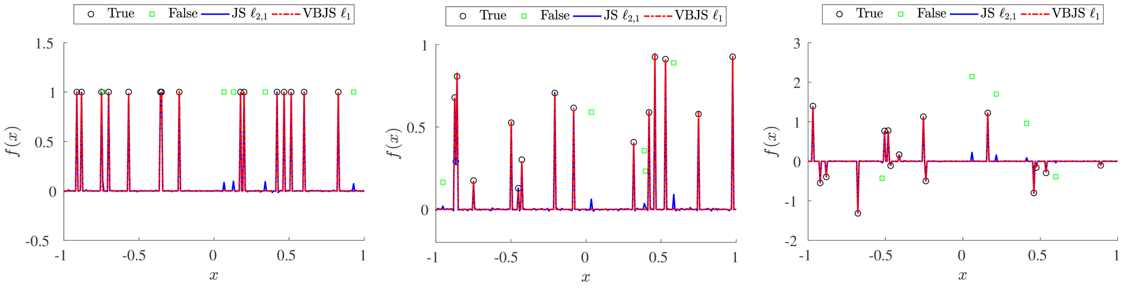

Figure 3.4: Sparse signal recovery employing the usual $\ell_{2,1}$ joint sparsity method in (3.6) and our proposed VBJS technique with $p = 1$. Here there are $J = 10$ measurements of which $K = 5$ contain false data. (left) Binary data values. (middle) Data values sampled from a uniform distribution on $[0, 1]$. (right) Data values sampled from a zero-mean unit-variance Gaussian distribution.

VBJS technique with $p = 1$ (dot-dashed) and the more classical $\ell_{2,1}$ JS regularization in (3.6), implemented using techniques in [37, 162]. In this case the final $\ell_{2,1}$ JS reconstruction is the pointwise average of the recovered vectors, $\{\hat{\boldsymbol{f}}^j\}_{j=1}^J$. In Figure 3.4(left), the signal consists of a sparse number of binary values, while the signals in Figure 3.4(middle) and (right) contain a sparse number of values sampled from a uniform distribution on $[0, 1]$ and a Gaussian distribution with zero-mean and unit-variance, respectively. In each case there are a total of $J = 10$ measurements vectors where $K = 5$ measurement vectors are corrupted with false data. Based on parameters used in other studies, [102, 2], we choose $N = 256$, $M = 100$ and sparsity $s = ||\boldsymbol{f}||_0 = 15$ for all three experiments. As is evident in Figure 3.4, the VBJS method successfully recovers each of the three sparse signals with limited influence from the false data. Conversely, the classic $\ell_{2,1}$ JS method is indeed influenced by the bad data. Similar behavior (not reported here) can be observed for different choices of N, M J, K and s.

Table 3.1: Relative reconstruction errors (3.45) for the traditional $\ell_{2,1}$ JS method and the VBJS method with $p = 1$ using the weights defined in (3.12) and (3.14).

False	Binary			Uniform			Gaussian		
Data	JS $\ell_{2,1}$	(3.12)	(3.14)	JS $\ell_{2,1}$	(3.12)	(3.14)	JS $\ell_{2,1}$	(3.12)	(3.14)
0%	.0127	.0104	.0383	.0254	.0244	.0579	.0142	.0153	.0388
20%	.1724	.0094	.0288	.1068	.0234	.0594	.1304	.0144	.0314
50%	.2961	.0108	.0499	.2943	.0243	.0621	.1249	.0168	.0262
90%	.1543	.0083	.0358	.3089	.0196	.0775	.1233	.0128	.0397

Table 4.1 displays the relative error,

$$E = \frac{||\hat{\boldsymbol{g}} - \boldsymbol{f}||_2}{||\boldsymbol{f}||_2}, \tag{3.45}$$

for the recovery vector $\hat{\boldsymbol{g}}$. In each case we use $J = 10$ measurements where K, the number of false data measurements, is based on the given percentage in the first column. For consistent comparison we use $N = 256$, $M = 100$ and sparsity $s = ||\boldsymbol{f}||_0 = 20$ in all cases. It is evident that the VBJS technique with $p = 1$ and weights in (3.12) yields small error even as the percentage of false data increases. Conversely, the traditional $\ell_{2,1}$ JS method is more susceptible to false data. For comparison we included results using the weights given in (3.14). We note that to handle the noise and different jump heights in the problem, when using the weights in (3.14), we must solve (3.10) by tuning the parameter μ to $\mu = .1$. Regardless, it is evident that the weights designed in (3.12) outperform the weights in (3.14) in all cases, and in the former case, *no* additional parameter tuning is needed.

To further demonstrate the success of our method, at varying levels of sparsity for different numbers of measurements and false data, we calculate the probability that

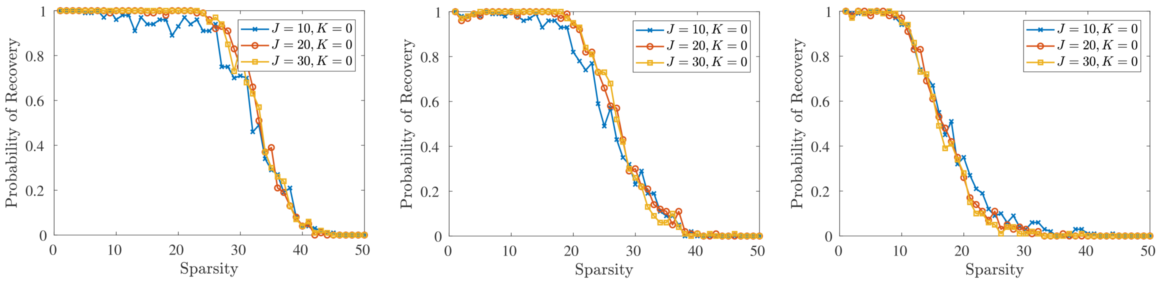

Figure 3.5: Probability of successful recovery of the sparse (left) binary signal, (middle) uniform signal, and (right) Gaussian signal with $J = 10, 20$ and 30 measurements, none of which contain false data.

the sparse signal is successfully recovered. Similar analysis was done in [91, 164, 21, 33, 46, 47]. Specifically, the probability of recovery is calculated over 100 trials at the specified configuration (J, K, and sparsity level s) with $N = 256$, $M = 100$ and no additive noise. Recovery is deemed a success if $||\hat{\boldsymbol{g}} - \boldsymbol{f}||_\infty \leq 5 \times 10^{-3}$, that is, when the VBJS method can successfully distinguish signals larger than the resolution size, $\mathcal{O}(\frac{1}{N})$.

In Figure 3.5 we see the recovery plots for each of the three signals considered with $J = 10, 20$ and 30 measurements, none of which contain false data. In this case, additional measurements do not improve the already high recovery rates. However, in Figure 3.6 we see that as the percentage of measurements that are false increases, it becomes more advantageous to have more measurements. Across top row of Figure 3.6 the percentage of false data increases to 50% while the number of measurements changes from $J = 10, 20$ to 30 for each type of sparse vector (binary, uniform, and Gaussian). Across the bottom row of Figure 3.6 the number of measurements $J = 20$ remains fixed, while the percentage of false data included increases from 20% to 50% to 90%. We see that when 50% of the measurements are false, the probability of recovery remains high for large sparsity values. When the percentage of false data

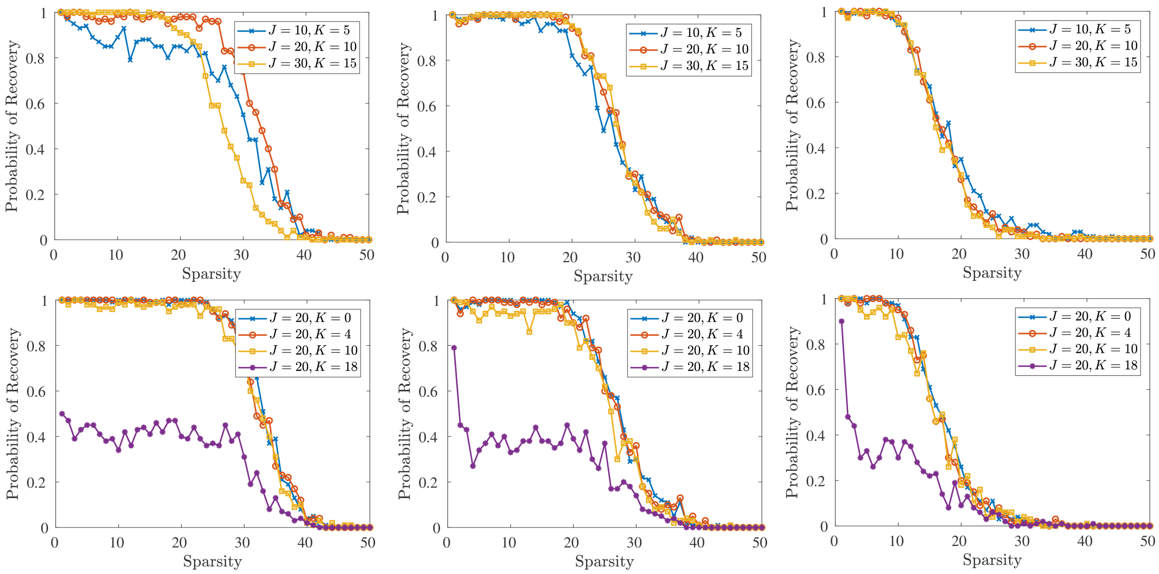

Figure 3.6: The probability of recovery of a sparse signal for various combinations of J, K and $||\boldsymbol{f}||_0$. Here $N = 256$ and $M = 100$. (left) Binary sparse vectors, (middle) uniform sparse vectors and (right) Gaussian sparse vectors.

increases to 90%, most probability of recovery values fall below .5.

3.5.2 Sparse Edges in One Dimension

We now consider the reconstruction of two piecewise smooth functions, given by

Example 3.5.1 *Define $f(x)$ on $[-\pi, \pi]$ as*

$$f(x) = \begin{cases} \frac{3}{2}, & -\frac{3\pi}{4} \leq x < -\frac{\pi}{2} \\ \frac{7}{4} - \frac{x}{2} + \sin\left(x - \frac{1}{4}\right), & -\frac{\pi}{4} \leq x < \frac{\pi}{8} \\ \frac{11}{4}x - 5, & \frac{3\pi}{8} \leq x < \frac{3\pi}{4} \\ 0, & \textit{otherwise.} \end{cases}$$

Example 3.5.2 *Define* $f(x)$ *on* $[-1,1]$ *as*

$$f(x) = \begin{cases} \cos\left(\frac{\pi}{2}x\right), & -1 \leq x < -\frac{1}{2} \\ \cos\left(\frac{3\pi}{2}x\right), & -\frac{1}{2} \leq x < \frac{1}{2} \\ \cos\left(\frac{7\pi}{2}x\right), & \frac{1}{2} \leq x \leq 1 \end{cases}$$

Each function exhibits sparsity in the jump function domain, that is $\boldsymbol{f}$ is not sparse, but $||[\boldsymbol{f}]||_0 = s$, with $s << N$, and $[\boldsymbol{f}] = \{[f](x_j)\}_{j=1}^N$ is the corresponding vector of edges. We consider the proposed weights (3.12) and the weights given by (3.14) in [2] for the weighted ℓ_p reconstructions (3.10) with $p = 1$ and 2.

For both examples we use the uniform grid in (3.1) with $N = 128$ points and $M = 128$ measurements. We acquire $J - K$ data vectors according to (3.2). The acquisition process for the K rogue vectors, as described below, considers situations where there is false information about the underlying solution as well as in the measurement matrix. In both examples we initialize the VBJS algorithm by constructing data vectors $\check{\boldsymbol{f}}^j$ for $j = 1, ..., J$ via (3.4) with μ sampled from a uniform distribution on $[0,1]$. The sparsifying transform operator $\boldsymbol{\mathcal{L}}$ is chosen to be the polynomial annihilation (PA) transform matrix of order $m = 2$ in (2.5).

In Example 3.5.1, the K false data vectors are formed by adding random shifts at random locations to the initial underlying function $\boldsymbol{f}$ in (3.2). That is, the data vectors (3.2) are modified such that

$$\boldsymbol{y}^j = \begin{cases} \boldsymbol{A}^j(\tilde{\boldsymbol{f}}^j) + \boldsymbol{\eta}^j, & j = 1, ..., K \\ \boldsymbol{A}^j(\boldsymbol{f}) + \boldsymbol{\eta}^j, & j = K+1, ..., J, \end{cases} \tag{3.46}$$

where each element $\tilde{f}_i^j$ of $\tilde{\boldsymbol{f}}^j$ is given as

$$\tilde{f}^j(x_i) = \begin{cases} f(x_i) + \alpha_j, & x \leq -1 + 2\gamma_j \\ f(x_i) + \beta_j, & x > 1 + 2\gamma_j. \end{cases}$$

Here α_j and β_j are random integers in $[-2, 2]$ for $j = 1, ..., K$ and each γ_j is i.i.d. sampled from a uniform distribution on $[0, 1]$. The forward model $\boldsymbol{A}^j$ in (3.46) is defined as a square matrix with i.i.d., zero-mean, unit-variance, Gaussian entries for all $j = 1, ..., J$, and the additive noise $\boldsymbol{\eta}^j$ is assumed to be i.i.d. Gaussian with zero-mean and variance equal to .2.

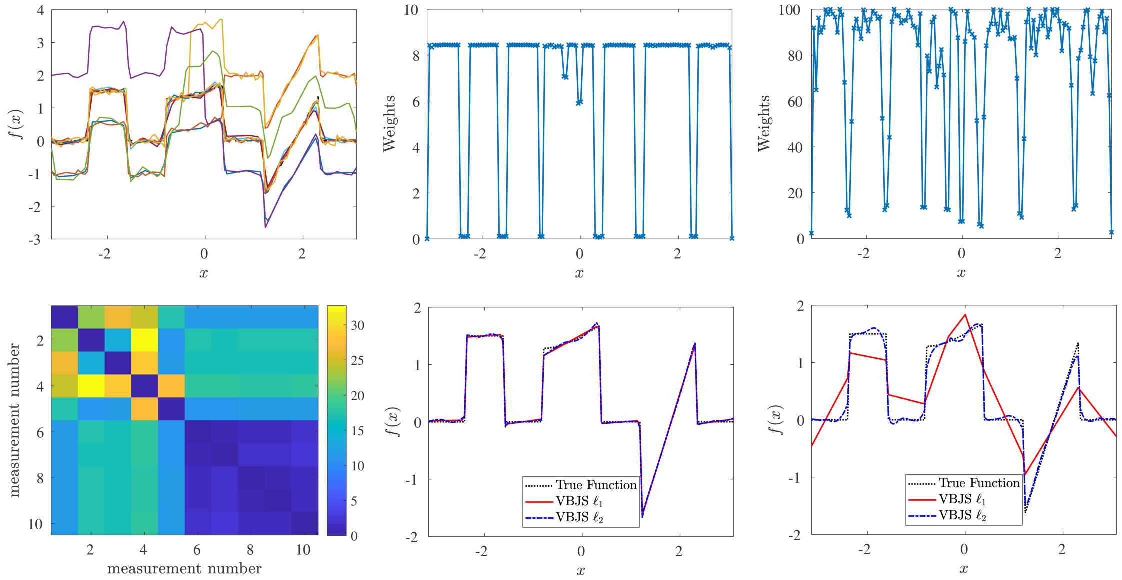

Figure 3.7: (top-left) $J = 10$ measurement vectors with $K = 5$ false data acquired using (3.4) with (3.46). Weights proposed in (top-middle) (3.12) and (top-right) (3.14) from [2]. (bottom-left) Corresponding distance matrix $\mathcal{D}$ in (3.15). VBJS reconstructions with $p = 1$ and $p = 2$ in (3.10) and weights in (bottom-middle) (3.12) and (bottom-right) (3.14).

For Example 3.5.2, we choose $\boldsymbol{A}^j$ to be a subsampled discrete Fourier transform (DFT) matrix for $j = 1, ..., K$, and the standard DFT matrix for $j = K + 1, ..., J$, so that

$$\boldsymbol{A}^j = \begin{cases} \frac{1}{\sqrt{N}} P_{\Omega^j} \boldsymbol{F}, & j = 1, ..., K \\ \frac{1}{\sqrt{N}} \boldsymbol{F}, & j = K + 1, ..., J. \end{cases} \tag{3.47}$$

Here $\boldsymbol{F} \in \mathbb{C}^{N\times N}$ is the DFT matrix and $P_{\Omega^j} \in \mathbb{R}^{N\times N}$ is a row selector matrix where each $\Omega^j \subseteq \{1, ..., N\}$ randomly selects and zeros out $N/2$ rows of $\boldsymbol{F}$. We choose to replace 75% of the selected rows with a random vector $\gamma \sin(x)$, where γ is repeatedly sampled from the normal distribution. In this way, we can simulate K false and $J-K$ true data vectors according to (3.2) where $\boldsymbol{\eta}^j$ is chosen as complex Gaussian noise with zero mean and variance equal to .75 for all $j = 1, ..., J$.

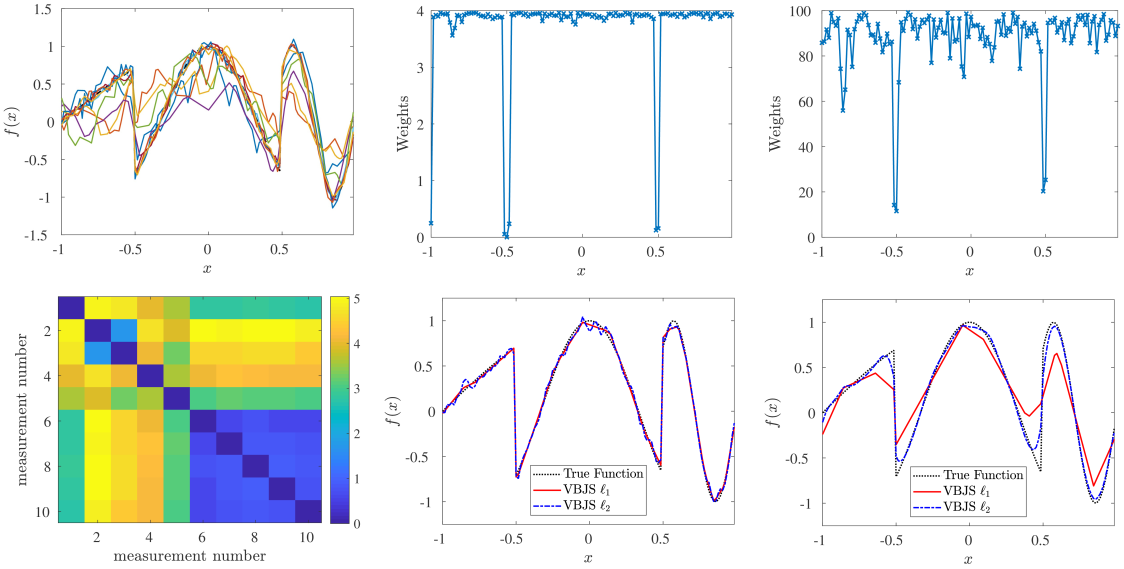

Figure 3.8: (top-left) $J = 10$ measurement vectors with $K = 5$ false data acquired using (3.4) with (3.47). Weights proposed in (top-middle) (3.12) and (top-right) (3.14) from [2]. (bottom-left) Corresponding distance matrix $\mathcal{D}$ in (3.15). VBJS reconstructions with $p = 1$ and $p = 2$ in (3.10) and weights in (bottom-middle) (3.12) and (bottom-right) (3.14).

Figures 3.7 and 3.8 display the results of reconstructing Examples 3.5.1 and 3.5.2 using VBJS with weights defined in (3.12) and (3.14) for $p = 1$ and 2 in (3.10). It is evident that using our proposed weights yields improved accuracy as well as prevents the influence of misleading/false data. We repeat these experiments, without adding

Gaussian noise to the data ($\boldsymbol{\eta}^j = 0$ for all $j = 1, ..., J$), with our proposed weights for $N = 32, 64, 128$ and 256 grid points, each time calculating the pointwise error in the reconstruction. That is, for each $\hat{\boldsymbol{g}}$ we calculate

$$\log_{10} |\hat{\boldsymbol{g}} - \boldsymbol{f}|. \tag{3.48}$$

The pointwise error plots corresponding to the reconstruction of Examples 3.5.1 and 3.5.2 are then displayed in Figure 3.9(top) and Figure 3.9(bottom), respectively, for $p = 1$ and 2 in (3.10). In Figure 3.9, the left two columns were calculated using our proposed weights (3.12) and the right two columns were calculated using the weights (3.14) given in [2]. The results shown here are consistent with those displayed in Figure 3.6. It is also evident that the weights provided by (3.12) yield better results than those given by (3.14). Finally, we see that the VBJS solutions with $p = 2$ also maintain a high level of accuracy, indicating that accurate solutions can be obtained using the less computationally intensive ℓ_2 regularization. For multi-dimensional problems with many measurement vectors, using ℓ_2 instead of ℓ_1 would provide a significant reduction in computational cost.

Table 3.2: Relative reconstruction errors (3.45) for the VBJS method (3.10) with $p = 1$ and 2 using the weights defined in (3.12) and (3.14). Here $N = 128$.

	SMV	ℓ_1 (3.12)	ℓ_2 (3.12)	ℓ_1 (3.14)	ℓ_2 (3.14)
Example 3.5.1	.0844	.0335	.0393	.4173	.1694
Example 3.5.2	.0692	.0536	.0716	.3142	.0959

For further comparison, Table 3.2 displays the relative error (3.45) for each example, while Table 3.3 measures the performance at a neighboring grid point to a jump

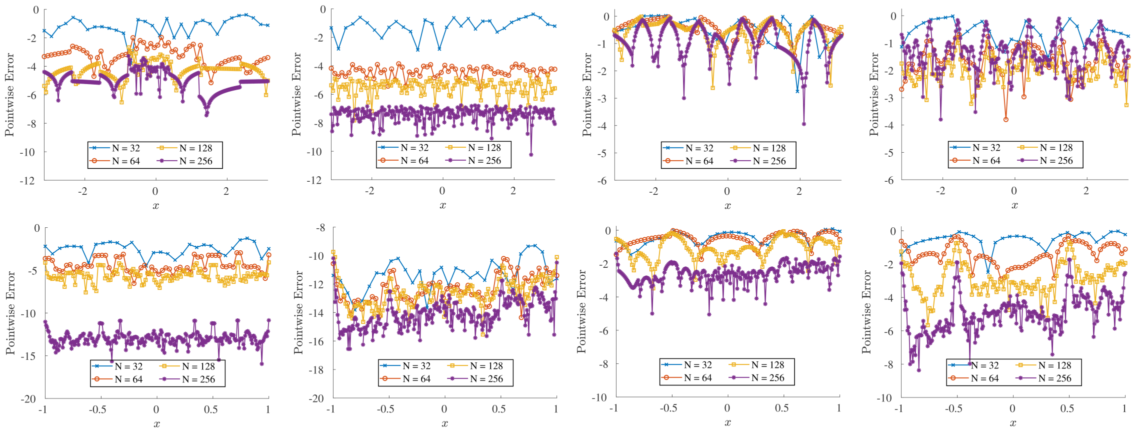

Figure 3.9: The pointwise error of the VBJS reconstructions of (top) Example 3.5.1 and (bottom) Example 3.5.2 for $N = 32, 64, 128$ and 256 with $J = 20$ measurements $K = 4$ of which are false with $p = 1$ (left,right-middle) and $p = 2$ (left-middle,right). In (left,left-middle) we use the weights given in (3.12) and in (right,right-middle) we use the weights given in (3.14).

discontinuity, given by

$$|\boldsymbol{f}(x_*) - \hat{\boldsymbol{g}}(x_*)|.$$

For the SMV approximation we choose $\hat{\boldsymbol{y}}$ using (3.16), that is, we consider the best possible solution. In each case we use $J = 10$ measurements where $K = 5$ vectors contain false infromation. Observe that using the VBJS algorithm with the weights in (3.12) with either ℓ_1 or ℓ_2 regularization yields better accuracy than the weights in (3.14), proposed in [2]. These results occur *without* any additional parameter tuning, which is required for both the SMV and VBJS using (3.14). Our method also shows general improvement over the SMV approximation, (3.4), which does not contain any false information.

Table 3.3: Absolute error near a discontinuity for the VBJS method (3.10) with $p = 1$ and 2 using the weights defined in (3.12) and (3.14). In Example 3.5.1, $x_* = 1.23$ and in Example 3.5.2, $x_* = -.55$. Here $N = 128$.

	SMV	ℓ_1 (3.12)	ℓ_2 (3.12)	ℓ_1 (3.14)	ℓ_2 (3.14)
Example 3.5.1	.1417	.0048	.0285	.3440	.4276
Example 3.5.2	.0131	.0206	.0132	.3738	.1325

3.5.3 Sparse Edges in Two Dimensions

We now consider reconstructing two dimensional images using the VBJS approach. We note that the original polynomial annihilation edge detection method constructed in [5] was, by design, multi-dimensional. However, as was discussed in [4] and in Chapter 2 in (2.10), for optimization algorithms using ℓ_1 regularization, applying the PA transform dimension by dimension was both more efficient and more accurate when on a uniform grid. Therefore, to calculate the weights (3.12) in the two dimensional case, we first calculate the two dimensional edge map for each $j = 1, \cdots, J$ as

$$\mathcal{E}^j = \mathcal{L}\check{\boldsymbol{f}}^j + \check{\boldsymbol{f}}^j\mathcal{L}^T.$$

The columns of each $\mathcal{E}^j$, $j = 1, \cdots, J$, are then stacked on top of each other to form the matrix of J vectors of approximations of some sparse feature of the underlying image, i.e. the two dimensional analogue of (3.8). Continuing as in one dimension, the weights are now calculated according to (3.12) and then reshaped into a matrix $W \in \mathbb{R}^{N\times N}$. The non-zero entries $w_{i,j}$ of W correspond to the sparse regions of the image, while the entries are approximately zero whenever an edge is assumed to be present. Observe that W is not sparse, so the implementation methods developed in Section 3.4 are critical for numerical efficiency.

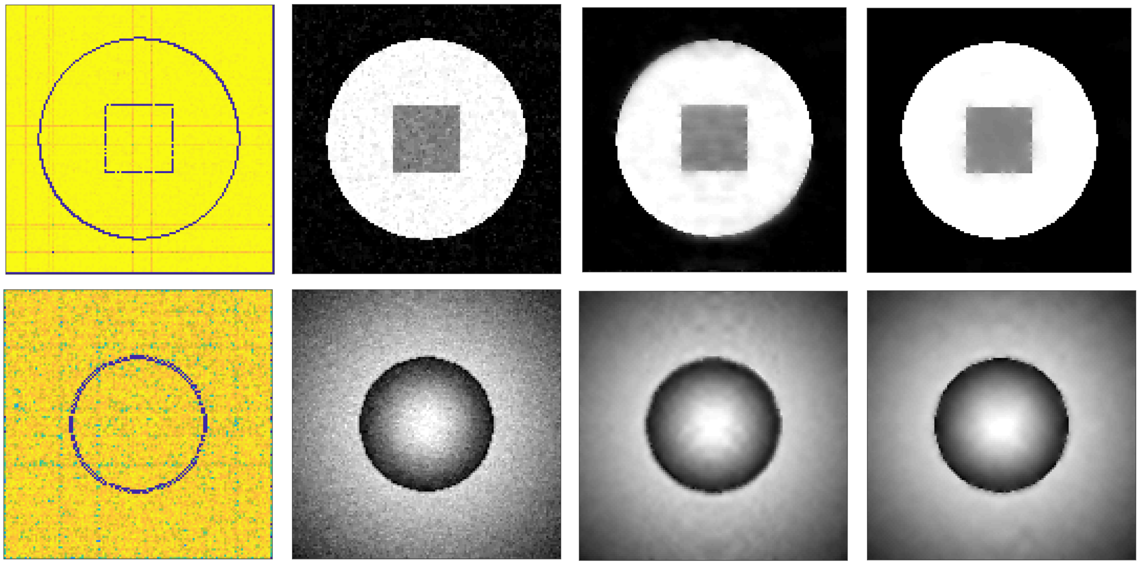

Figure 3.10: (left) Weights calculated using (3.12), where the darker shades indicate $w_{i,j} \approx 0$. (middle-left) Reconstruction of a single measurement vector using (3.15) and (3.4). (middle-right) VBJS with $p = 1$. (right) VBJS with $p = 2$. (top) Example 3.5.3 reconstruction performed with PA transform of order $m = 1$ in (2.5). (bottom) Example 3.5.4 reconstruction performed with PA transform of order $m = 2$ in (2.5).

As in the one dimensional case, we consider two examples:

Example 3.5.3 *Define* $f(x, y)$ *on* $[-1, 1]^2$ *as*

$$f(x,y) = \begin{cases} 15, & |x|, |y| \leq \frac{1}{4} \\ 20, & |x|, |y| > \frac{1}{4}, \quad \sqrt{x^2+y^2} \leq \frac{3}{4} \\ 10, & \textit{else} \end{cases}$$

Example 3.5.4 *Define* $f(x, y)$ *on* $[-1, 1]^2$ *as*

$$f(x,y) = \begin{cases} 10\cos\left(\frac{3\pi}{2}\sqrt{x^2+y^2}\right), & \sqrt{x^2+y^2} \leq \frac{1}{2} \\ 10\cos\left(\frac{\pi}{2}\sqrt{x^2+y^2}\right), & \sqrt{x^2+y^2} > \frac{1}{2} \end{cases}$$

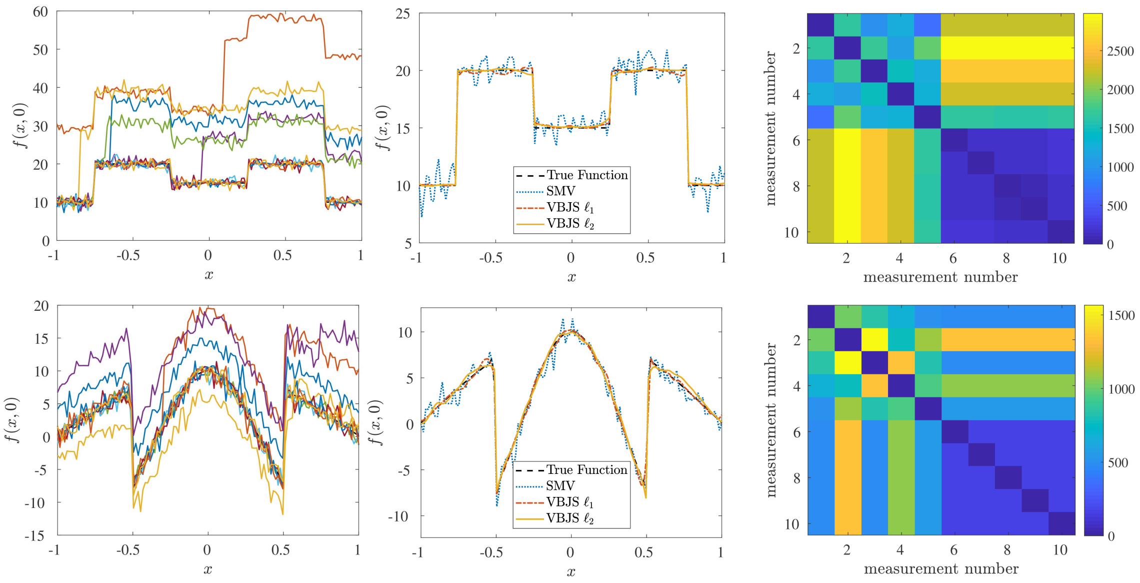

Figure 3.11: (top) Results corresponding to Example 3.5.3. (bottom) results corresponding to Example 3.5.4. (left) Cross sections ($y = 0$) of $J = 10$ measurement vectors with $K = 5$ false data representations. (middle) Cross sections ($y = 0$) of VBJS reconstructions for $p = 1$ and 2 in (3.10) compared to the SMV constructed using (3.4). (right) Data selection matrices $\mathcal{D}$ (3.15).

We sample each function $\boldsymbol{f} : \mathbb{R}^{N\times N} \rightarrow \mathbb{R}$ on a uniform grid as $f_{i,l} = f(x_i, y_l)$, where

$$x_i = -1 + \frac{2}{N}(i-1), \qquad y_l = -1 + \frac{2}{N}(l-1),$$

for each $i, l = 1, \cdots, N$. In (3.2), $\boldsymbol{A} : \mathbb{R}^{N\times N} \rightarrow \mathbb{C}^{N\times N}$ is defined to be the normalized, two dimensional discrete Fourier transform operator so that $\boldsymbol{A}^* = \boldsymbol{A}^{-1}$, and $\boldsymbol{\eta}^j$ is zero mean complex Gaussian noise with .5 variance for all $j = 1, ..., J$. As in the one dimensional case we use (3.4) to construct each $\check{\boldsymbol{f}}^j$. Because of the piecewise constant nature of Example 3.5.3 we apply the PA transform with order $m = 1$. Similarly, for Example 3.5.4 we use $m = 2$. We note that it is possible to use $m > 2$, but in this case, because of the noise, the higher order polynomial approximation leads to overfitting. For each measurement vector the regularization parameter μ in (3.4) is sampled from a uniform distribution on $[0, 10]$.

Figure 3.10 displays the result of applying VBJS with $p = 1$ (middle-right) and $p = 2$ (right) to Examples 3.5.3 and 3.5.4. For both examples we use $J = 10$ measurement vectors where $K = 5$ falsely represent the underlying function. (The corresponding measurement selection matrices (3.15) are shown in Figure 3.11(right).) Figure 3.10(middle-left) shows the the SMV results on the measurement vector selected by (3.15) calculated using (3.4). It is evident in both examples that the VBJS technique with either $p = 1$ or 2 in (3.10) leads to improved visualization over the standard SMV reconstruction, even when the standard SMV uses the "best" initialization as determined by (3.15). This result is confirmed in Figure 3.11, where we compare the corresponding one-dimensional cross sections at $y = 0$.

Figure 3.12 shows the relative error, (3.45), and pointwise error (3.48) at cross section $y = 0$, for the reconstructions of Examples 3.5.3 and 3.5.4 using the VBJS algorithm with both $p = 1$ and 2. It is evident that when there are no false measurements, for N sufficiently large, no additional measurements ($J > 10$) are needed to improve performance. It is also apparent that the results are equally accurate for $p = 1$ and $p = 2$, although using $p = 2$ is much more efficient.

3.6 Discussion and Conclusions

In this investigation we proposed a modification to the variance based joint sparsity technique (VBJS), introduced in [2], in both the weighting vector and in the choice of reconstruction vector. Our adaptation is especially critical when some data vectors contain false measurements. We additionally proved that the ADMM algorithm could be successfully used for the weighted ℓ_1 case, and moreover, that for our choice of weights in (3.12), *no* extra parameter tuning is needed to achieve high

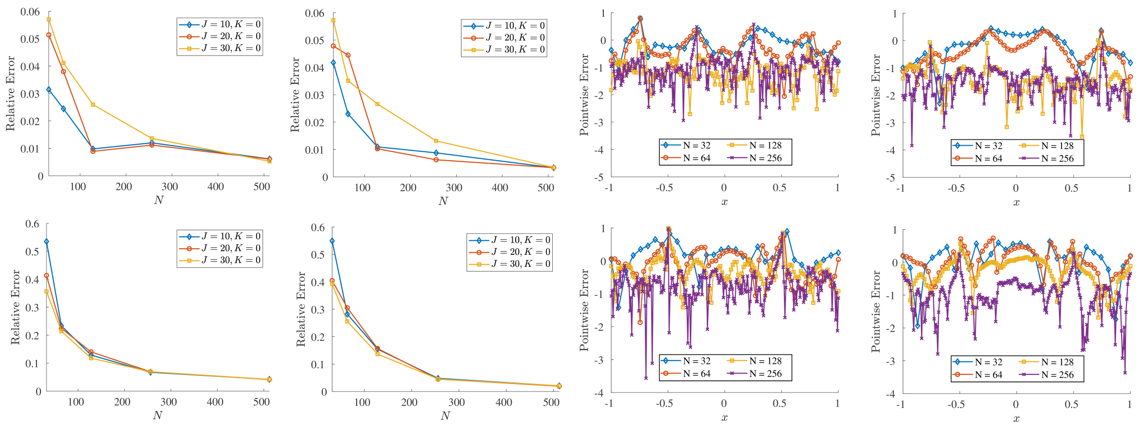

Figure 3.12: (top) Error in reconstructing Example 3.5.3. (bottom) Error in reconstructing Example 3.5.4. Relative error of the VBJS reconstructions with (left) $p = 1$ and (middle-left) $p = 2$ reconstructions for $N = 32, 64, 128, 256$ and 512. Pointwise error at cross section $y = 0$ of the VBJS reconstructions with (middle-left) $p = 1$ and (right) $p = 2$ reconstructions for $N = 32, 64, 128$ and 256.

accuracy and fast convergence. Hence our method is robust and suitable to a wide range of problems. We also presented a corresponding gradient descent method for the weighted ℓ_2 case.

Our numerical results demonstrate that the VBJS method with the weights designed in (3.12) yields improved accuracy and robustness over the single measurement vector case, the classical $\ell_{2,1}$ JS method, and the original VBJS method proposed in [2]. By including an optimal data vector selection step, we are able to obtain high accuracy and good sparse signal recovery even when a subset of the given measurement data misrepresents the underlying function. Furthermore, using the weighted ℓ_2 norm also yields good results and is much more cost effective than the weighted ℓ_1 reconstructions.

In future investigations we will conduct a thorough convergence analysis of the VBJS method, in particular to establish rigorous results for the weighted ℓ_2 case.

We will also parallelize our algorithm so that we may test it on synthetic aperture radar automatic target recognition problems, where current algorithms fail when obstructions are added to (or taken out of) imaging scenes. Because our method is non-parametric, autonomy will be maintained. This framework also lends itself to data fusion problems, where measurements of a scene are obtained through multiple imaging techniques and must be combined to yield optimal results. Finally, the VBJS algorithm can potentially be used in numerical partial differential equation solvers, in particular to develop predictor-corrector methods for equations that exhibit singularities or for which shock discontinuities evolve.

Chapter 4

USING REGULARIZATION TO IMPROVE NUMERICAL PARTIAL DIFFERENTIAL EQUATION SOLVERS

4.1 Introduction

This chapter describes work that was published in [123] in *The Journal of Scientific Computing.*

Hyperbolic systems of partial differential equations (PDEs) model a variety of phenomena in fields such as gas dynamics, acoustics, elastodynamics, optics and geophysics, [85]. Solutions to hyperbolic PDEs often contain discontinuities such as shock waves and fronts, which can develop in finite time even when the initial conditions are smooth. Although numerical algorithms for solving hyperbolic PDEs have been broadly investigated, [40, 67, 86, 85, 84, 136, 137, 138, 139], the presence of shock discontinuities still causes complications in the solutions. For example, high-order methods must include viscosity or slope limiters of some kind in order to avoid oscillations that lead to instabilities. On the flip side, too much viscosity yields a loss of resolution near shock discontinuities. Clever upwinding algorithms can reduce the amount of dissipation and still retain sharp features in the solution, but can be more difficult and computationally intensive to implement. Finally, the stability condition for non-linear PDEs is often more stringent, making long term solutions more computationally expensive, [86, 85, 84].

A hybrid, spatially-adaptive, weighted, essentially non-oscillatory (WENO) scheme was developed in [40]. With this technique, the spatial scheme is updated based on the given spatial location and the dynamics of the system at a given time. At each iteration of the algorithm, the shock is detected using various shock detection algorithms. The domain is then divided into non-smooth and smooth sub-domains, where appropriate schemes are then used to approximate the solution in each sub-domain The method is non-linear and also relies on accurate shock detection. In [139] the spectral viscosity (SV) method was adapted to include a step that locates the region containing the shock location. By doing so, less viscosity is enforced in the smooth regions of the solution. High order post-processing, which typically requires knowledge of each shock location, is required to recover spectral accuracy from the SV solution, [59, 130].

One of the main difficulties with the approaches discussed above is their reliance on detecting discontinuities. Mis-identification leads either to instability, when a shock goes undetected, or to unnecessary dissipation, when a shock is determined to exist in smooth regions. Typically, the algorithms are also computationally intensive, and often require small time steps to satisfy the CFL stability conditions.

The use of ℓ_1 regularization methods to promote sparsity is frequently encountered in imaging and signal processing applications, but they are still of limited use in solving PDEs. Sparse dynamics for hyperbolic PDEs with solutions exhibiting behaviors on multiple spatial scales was investigated in [125], where it was proposed to include the constraint that the approximate solution resides on a sparse subspace of a basis. However, solutions with singularities in the physical domain were not considered. In particular, the solutions were such that they exhibited high frequencies on a small spatial scale so that they had sparse representation when projected onto a Fourier basis. Moreover, the algorithm consists of advancing the PDE forward in time and then projecting the updated solution onto a sparse subset. This requires additional transformations between spatial and coefficient domains at each iteration, thereby adding considerable computational complexity. In [67], a method was proposed to approximate solutions to viscous conservation laws. The method utilizes sparse and low-rank decompositions for which sharp-contrast features are separated from smooth, low-energy behaviors. While higher accuracy is achieved, the technique relies on knowledge of the sharp-contrast feature locations, which is not always practical.

An ℓ_1 minimization technique was developed in [79, 80] to approximate the solutions of steady-state conservation laws in one and two dimensions, respectively. In particular, the finite volume approximation of the corresponding non-singularly perturbed problem was written as an overdetermined system and then solved by minimizing the ℓ_1 norm of the system residual. The minimization problem had a closed form explicit solution for each test problem chosen. While both linear and non-linear steady state problems were considered, the method did not include time integration, and it is not apparent how such techniques could be adapted to time dependent problems when ℓ_1 solutions are not explicitly available.

Finally, an ℓ_1 based finite element approximation method for first order partial differential equations that converge to the corresponding viscosity solutions was developed in [61]. In particular, in the case of the linear transport equation, the ℓ_1 minimizer selects the upwind solution. The method only considers stationary problems, with the justification that the solution represents a snapshot in time. However, this assumption is not valid when discontinuities are introduced. Indeed, while some test problems had initial conditions with discontinuities, shocks were not considered in any of the methods provided in [79, 80, 61].

In this thesis we offer an alternative approach that incorporates an ℓ_1 regularization term directly into a time dependent PDE solver. Our method yields some distinct advantages. First, because we account for the sparsity of the singularities in the physical domain as part of the PDE solver, we do not need to explicitly locate any shock discontinuities or subdivide the domain. Second, because we use the *polynomial annihilation* (PA) operator (discussed in Section 2.2) as our ℓ_1 regularization term, our method is high-order [4, 5, 153]. We note that using TV would recover piecewise constant solutions; that is, it yields first order approximations. Finally, our numerical results demonstrate that our method maintains stability even when the time step is larger than normally dictated by the CFL condition. Therefore, it is efficient when fast optimization algorithms are employed.

The rest of this chapter is organized as follows. In Section 4.2 we describe how a given PDE solver can be enhanced using the PA operator (2.5) in the ℓ_1 regularization term. While the PDE solver can take on a variety of forms, we use standard finite-difference schemes and pseudo-spectral methods to demonstrate our results. To ensure efficiency, we then propose an alternating direction method of multipliers (ADMM) algorithm for solving the resulting convex optimization problem in Section 4.3. We present our numerical results in Section 4.4. Examples of hyperbolic PDEs include Burgers and Euler's equations. In Chapter 5 we also test our method on a two dimensional multiplicative noise model, which is often used to reduce speckle in images. In all cases we are better able to resolve functions and images with discontinuities without explicit knowledge of the jump locations.

4.2 Using ℓ_1 Regularization in PDE Solvers

Consider the one-dimensional non-linear conservation law of the form

$$u_t(x,t) + f(u(x,t))_x = 0 \tag{4.1}$$

on a bounded domain, Ω. Here, $f : \Omega \to \mathbb{R}$ is the flux function and $u : \Omega \to \mathbb{R}$ is a conserved quantity with appropriate initial and boundary conditions. We will use (4.1) as a prototype to develop our new ℓ_1 regularization PDE solver. We seek an approximation $U_j^n \in \mathbb{R}^N$ to the solution $u_j^n := u(x_j, t_n)$ of (4.1) given by

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}\left(F_{j+1/2}^n - F_{j-1/2}^n\right), \quad j = 0, ..., N-1, \tag{4.2}$$

where $F_{j+1/2}^n$ is an approximation of the average flux along $x = x_{j+1/2}$. That is, in reference to (4.1),

$$F_{j+1/2}^n \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f\left(u\left(x_{j+1/2}, t\right)\right) dt. \tag{4.3}$$

and Δx and Δt are the appropriate grid size and time step size, respectively. We note that our technique does not require a uniform grid, as the PA operator (2.5) can be formulated for any set of data points. For our purposes, we will write (4.2) as

$$U^{n+1} = \mathcal{D}\left(F^n, U^n\right) =: b^n, \tag{4.4}$$

where U^{n+1} represents the vector of approximations U_j^{n+1} at time t_{n+1}, F^n represents the vector of approximations $F_{j+1/2}^n$ at time t_n for $j = 0, ..., N-1$, and $\mathcal{D}$ is the operator representing the chosen numerical method. Observe that b^n is simply a vector of values explicitly calculated at time t_n which will be incorporated into the data fidelity term. We consider only explicit schemes, because an implicit scheme may lead to non-convexities in the objective function. This will be explored more in future research.

As noted previously, one of the main challenges in solving (4.1) is balancing the amount of viscosity, introduced for stability purposes, with maintaining desirable high resolution properties, especially near shocks. This is particularly difficult when the shock locations are unknown. Applying the techniques described in Section 2.1, we can now adapt (4.2) to include an ℓ_1 regularization term, reflecting that the jump discontinuities in the solution of (4.1) are sparse. Specifically, our new algorithm is given by

$$U^{n+1} = \underset{V}{\operatorname{argmin}}\left\{||\mathcal{L}^m V||_1 + \frac{\lambda}{2}||V - b^n||_2^2\right\}, \tag{4.5}$$

which is analogous to the convex optimization problem given in (2.3). Observe that minimizing $||U^{n+1} - b^n||_2^2$ obtains the best solution in the least squares sense, but does not adequately capture the shocks that may appear in (4.1). However, by augmenting (4.4) with the PA transform ℓ_1 regularization term, we encourage the solution U^{n+1} to have sparse representation jump function domain. Consequently, we obtain a stable solution without introducing too much artificial viscosity, and thus are able to maintain high resolution near shock locations. Moreover, by using $m > 1$ in (2.5), we are able to see greater variation in smooth regions. The fidelity term in (4.5) contains the numerical method chosen to solve (4.1), and the ℓ_1 regularization term includes the PA transform of order m. The parameter $\lambda > 0$ determines the influence of the sparsity constraint upon the fidelity term, and its tuning is typically application-dependent. Our numerical results demonstrate that (4.5) is robust for a range of λ, but more study is needed to verify its impact on stability.

4.2.1 ℓ_1 Enhancement for Finite Difference Methods

We first demonstrate our technique using the second order Lax Wendroff (LW) scheme for (4.1) in the spatial domain $x \in [a, b]$. Assume we are given grid points

$$x_j = a + j\Delta x, \quad j = 0, ..., N-1, \quad \Delta x = \frac{b-a}{N},$$

and define

$$\begin{aligned} U_{j+\frac{1}{2}}^{n+\frac{1}{2}} &= \frac{1}{2}\left(U_j^n + U_{j+1}^n\right) - \frac{\Delta t}{2\Delta x}\left(f(U_{j+1}^n) - f(U_j^n)\right) \\ U_{j-\frac{1}{2}}^{n+\frac{1}{2}} &= \frac{1}{2}\left(U_{j-1}^n + U_j^n\right) - \frac{\Delta t}{2\Delta x}\left(f(U_j^n) - f(U_{j-1}^n)\right). \end{aligned}$$

The LW scheme is then given by

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}\underbrace{\left[f\left(U_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right) - f\left(U_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right)\right]}_{\mathcal{D}(F^{n+1/2})}. \tag{4.6}$$

It is well known that using (4.6) to solve (4.1) results in unwanted oscillations behind shock locations and may also lead to instability, [84, 85]. To apply ℓ_1 enhancement, we define

$$b^n = U^n - \frac{\Delta t}{\Delta x}\mathcal{D}(F^{n+1/2}), \tag{4.7}$$

and insert it directly into (4.5). We will refer to this particular combination as the *ℓ_1 enhanced LW method*, and note that while we only employ LW in our numerical examples, the ℓ_1 enhancement can be applied to any finite difference method that can be written in the form (4.4).

In general, (4.6) is stable for

$$\Delta t \leq \frac{\Delta x}{|\zeta|_{\max}},$$

where $|\zeta|_{\max}$ is the maximum characteristic speed in magnitude. However, the LW method yields significant dissipation, leading to a loss of shock information at each time step. Thus, as will be seen in our results, it is beneficial to consider high-order methods which are better at resolving shocks. Of course, in this case stability becomes a concern as we discuss in the next section.

4.2.2 ℓ_1 Enhancement for Spectral Methods

Spectral methods provide highly accurate approximations for sufficiently smooth functions $u : [a, b] \to \mathbb{R}$. When u contains discontinuities, the resulting Gibbs phenomenon leads to $\mathcal{O}(1)$ oscillations in neighborhoods of discontinuities and first order accuracy in smooth regions. The method becomes unstable as a result of non-linear interactions within the PDE. Filtering or additional viscosity is often introduced to mitigate this problem. However, too much dissipation is undesirable as the shocks are “smeared over” and information is lost. Methods such as (super) spectral viscosity (SV), [137, 138], were introduced to apply minimal amounts of diffusion near shocks, and in [139] some attempt was made to localize the effects of dissipation even further by determining the discontinuous regions as time evolved. The SV methods are computationally expensive, as they amount to adding high-order viscosity in the underlying conservation law. Nevertheless, they offer an alternative to standard filtering. In Section 4.4 we demonstrate that using ℓ_1 regularization improves the accuracy of standard filtering and vanishing viscosity methods. [1]

We first consider the Fourier pseudo-spectral (PS) approximation of $u : [-\pi, \pi] \to \mathbb{R}$ given by [66, 22]

$$P_N u(x,t) = \sum_{k=-N/2}^{N/2-1} \tilde{u}_k(t) e^{ikx}, \quad \tilde{u}_k(t) = \sum_{j=0}^{N-1} u(x_j,t) e^{-ikx_j}, \tag{4.8}$$

with $x_j = -\pi + j\Delta x$, and $\Delta x = \frac{2\pi}{N}$. The pseudo-spectral (PS) approximation of (4.1) takes the form

$$(u_N)_t + (P_N f(u_N))_x = 0, \tag{4.9}$$

with u_N denoting the numerical solution. As noted previously, even when given a smooth initial condition, the solution to (4.1) may develop singularities. Due to the non-linear interaction, the resulting Gibbs phenomenon will yield instabilities within finite time, [136].

Let us now define

$$b^n = \{G(U^n) - \Delta t \sum_{k=-N/2}^{N/2-1} ik\tilde{f}_k(t_n) e^{ikx_j}, \quad j = 0, \cdots, N-1\}, \tag{4.10}$$

where

$$\tilde{f}_k(t) = \sum_{j=0}^{N-1} f\left(u_N(x_j,t)\right) e^{-ikx_j} \tag{4.11}$$

and $G(U^n)$ represents a linear time-stepping scheme, e.g. Runge Kutta. All of our numerical examples (except Lax-Wendroff and denoising) use fourth order Runge

[1] We also applied ℓ_1 enhancement to the spectral viscosity method for the Fourier and Chebyshev cases. Both resulted in improved accuracy that essentially mirrored the approximations displayed in Figures 4.4 and 4.10. Hence they are not reported here.

Kutta. The ℓ_1 *enhanced pseudo-spectral method* is now constructed by substituting b^n into the fidelity term of (4.5) and using (2.5) to enforce sparsity in the jump discontinuity domain.

A small amount of viscosity is often introduced in the numerical solver to reduce oscillations and overshoots resulting from the Gibbs phenomenon, [66, 22]. The approximation of (4.1) takes the form

$$(u_N)_t + (P_N f(u_N))_x = \epsilon (u_N)_{xx}, \tag{4.12}$$

where $\epsilon > 0$ and u_N denotes the numerical solution. Following the terminology in [66, 22], we refer to (4.12) as the *vanishing viscosity* (VV) method. To incorporate ℓ_1 regularization into (4.12), we define

$$b^n = \{G(U^n) - \Delta t \sum_{k=-N/2}^{N/2-1} ik\tilde{f}_k(t_n)e^{ikx_j} + \epsilon\Delta t \sum_{k=-N/2}^{N/2-1} k^2\tilde{u}_k(t)e^{ikx}, \quad j = 0, ..., N-1\} \tag{4.13}$$

with $\tilde{f}_k(t)$ defined in (4.11) and $\tilde{u}_k(t)$ defined in (4.8). The ℓ_1 *enhanced VV method* is constructed by inserting (4.13) into the (4.5). We note that (4.9) and (4.12) could have been written in collocation form, and subsequently the corresponding ℓ_1 enhanced collocation method. However, applying the FFT is sufficiently efficient in the given form.

Filtering also helps to reduce oscillations and improve stability, [66]. In our experiments, we applied an exponential filter (EF) to the solution after each time step. Enhancement with ℓ_1 regularization is straightforward.

Because most PDEs do not admit periodic solutions, we also consider the Chebyshev collocation method. In this case we modify the interval to $[-1, 1]$ and define the Chebyshev grid points as

$$y_j = -\cos\left(\frac{\pi j}{N}\right), \quad j = 0, ..., N. \tag{4.14}$$

The Chebyshev approximation of $u(x,t)$ is [66, 22]

$$P_N u(x,t) = \sum_{k=0}^{N} \tilde{u}_k(t) T_k(x), \quad \tilde{u}_k(t) = \frac{1}{\tilde{\gamma}_k} \sum_{j=0}^{N} u(x_j) T_k(x_j) w_j, \tag{4.15}$$

where the Chebyshev polynomials are

$$T_k(x) = \cos(k \arccos(x)), \tag{4.16}$$

and the weights and normalizing factors are given respectively as

$$w_j = \begin{cases} \frac{\pi}{2N} & j = 0, N \\ \\ \frac{\pi}{N} & j = 1, ..., N-1 \end{cases}, \quad \tilde{\gamma}_k = \begin{cases} \pi & k = 0, N \\ \\ \frac{\pi}{2} & k = 1, ..., N-1 \end{cases}.$$

To improve computational efficiency, we map the Chebyshev points according to [78] [2] given by

$$x_j = \frac{2}{\alpha\pi} \arcsin\left(\beta y_j\right), \quad j = 0, ..., N, \tag{4.17}$$

where we have chosen parameters

$$\alpha = 1 + \frac{2}{N\pi} \log(10^{-5}), \quad \beta = \sin\left(\frac{\alpha\pi}{2}\right). \tag{4.18}$$

We note that no attempt was made to optimize the mapping parameters. The Chebyshev approximation of (4.1) takes the form

$$(u_N)_t + \mathcal{D}(P_N f(u_N)) = 0, \tag{4.19}$$

with u_N denoting the numerical solution and $\mathcal{D}$ denoting the Chebyshev differentiation matrix. Each entry in $\mathcal{D}$ is

$$\mathcal{D}(i,j) = \frac{1}{N c_j} \sum_{k=0}^{N} \frac{T_k(x_j) T_k'(x_i)}{c_k}, \tag{4.20}$$

[2] With a small decrease in accuracy, the mapped Chebyshev method allows the time step to increase to $\mathcal{O}(\frac{1}{N})$, [78].

where $c_j = 1$ for $1 \leq j \leq N-1$ and $c_0 = c_N = 2$. [3] In our experiments, we used the software described in [42] to construct the Chebyshev differentiation matrix (4.20). To implement the sparsity enforcing PDE solver we first define

$$b^n = \{G(U^n) - \Delta t \mathcal{D}(P_N f^n)\}, \tag{4.21}$$

where

$$P_N f^n = \{f(u_N(x_j, t_n)), \quad j = 0, \cdots, N\}$$

so that

$$\mathcal{D}\left(P_N f(u_N)\right) \approx \mathcal{D}(P_N f^n).$$

Here again $G(U^n)$ represents an appropriate time stepping method. Inserting (4.21) into (4.5) with the PA transform operator (2.5) yields the ℓ_1 *enhanced Chebyshev method.* Boundary conditions are implemented at the end of each time step. When the enhancement is not implemented, we must apply an exponential filter of order $p = 16$ to maintain stability, [66].

4.3 ADMM for Regularized PDE Solvers

To minimize (4.5) for the optimial solution to the PDE at each time step, we modify the ADMM algorithm [88, 54] developed in Chapter 2. This is accomplished by formulating an equivalent problem

$$(U^{n+1}, \hat{g}) = \left\{ \underset{V,g}{\operatorname{argmin}} \ ||g||_1 + \frac{\mu}{2}||V - b^n||_2^2 \quad \text{subject to} \quad \mathcal{L}^m V = g \right\}, \tag{4.22}$$

where we have introduced the slack variable $g \in \mathbb{R}^N$ due to the non-seprability of $\mathcal{L}^m$ and V in the non-differentiable ℓ_1 norm. To approximate (4.22), we introduce

[3] The explicit matrix entries for (4.20) for (4.14) can be found in [22, 66]. In our examples, we use the mapped Chebyshev points, [78], so the derivative matrix depends on the chosen grid points x_j.

the Lagrangian multiplier $\sigma \in \mathbb{R}^N$, assume σ is fixed, and solve the unconstrained minimization problem given by

$$\underset{V,g}{\operatorname{argmin}}\ J(V, g, \sigma) \tag{4.23}$$

where

$$J(V, g, \sigma) := ||g||_1 + \frac{\mu}{2}||V - b^n||_2^2 + \frac{\beta}{2}||\mathcal{L}^m V - g||_2^2 - \langle \mathcal{L}^m V - g, \sigma \rangle. \tag{4.24}$$

Here $\mu \in \mathbb{R}$ is a non-negative regularization parameter and $\beta \in \mathbb{R}$ is a penalty parameter that effects the convergence rate of the ADMM algorithm. In particular, if the Lagrangian multipliers are updated a sufficient number of times, then the solution to (4.23) will converge to the solution of the constrained problem (4.22). The solution is approximated by alternating between minimizations of V and g.

As in Chapter 2, given the current value of V_k and the multiplier σ_k, the optimal g can be exactly determined using the shrinkage-like formula, [88, 55]:

$$g_{k+1} = \max\left(\left|\mathcal{L}^m V_k - \frac{\sigma_k}{\beta}\right| - \frac{1}{\beta}, 0\right) \operatorname{sign}\left(\mathcal{L}^m V_k - \frac{\sigma_k}{\beta}\right). \tag{4.25}$$

This is analogous to (2.44). The gradient descent method (2.49) is used to find the minimum over V as

$$V_{k+1} = V_k - \alpha \nabla_V J(V, g_{k+1}, \sigma_k)|_{V_k}, \tag{4.26}$$

where the gradient of $J(V, g_{k+1}, \sigma_k)$ with respect to V is given by

$$\nabla_V J(V, g_{k+1}, \sigma_k) = \mu(V - b^n) + \beta \left(\mathcal{L}^m\right)^* \left(\mathcal{L}^m V - g_{k+1}\right) - \left(\mathcal{L}^m\right)^* \sigma_k, \tag{4.27}$$

and the step size α is chosen as in (5.62) to give sufficient descent in the gradient direction. The Lagrange multiplier σ is updated after a sufficient number of updates on g and V as

$$\sigma_{k+1} = \sigma_k - \beta(\mathcal{L}^m V_{k+1} - g_{k+1}). \tag{4.28}$$

At the final iteration of the ADMM algorithm we set $U^{n+1} = V_{k+1}$. Algorithm 6 describes the general ADMM procedure for the PDE solution approximation problem (4.5).

Algorithm 6 ADMM as a Numerical PDE Solver

```
Determine parameters $\mu$, $\beta$ and $tol$.
Initialize $V_0 = U^n$, $g_0$ and $\sigma_0$.
for $k = 0$ to $K$ do
    while $||V_{k+1} - V_k|| > tol$ do
        Minimize $J$ for $g_{k+1}$ according to (4.25).
        Minimize $J$ for $V_{k+1}$ according to (4.26) and (4.27).
    end while
    Update Lagrange multiplier according to (4.28).
end for
Set $U^{n+1} = V_{k+1}$.
```

4.4 Numerical Results

We are now ready to demonstrate our ℓ_1 enhanced numerical solver, (4.5), by employing Algorithm 6, for Burgers' and Euler's equations. Our results show that we are able to resolve solutions *without* explicit knowledge of shock locations. We are also able to relax the usual CFL conditions so that our method is cost efficient.

In each case we will compare our new method to standard techniques based on the log of the pointwise error

$$E_{log}(u_N(x,t)) = \log_{10} |u_N(x,t) - u(x,t)| \tag{4.29}$$

where $u_N(x,t)$ is the numerical approximation to the true solution $u(x,t)$ at the final time step. To demonstrate the stability gained when using (4.5), we define

$$\Delta t = \alpha \min |x_{j+1} - x_j|, \quad j = 0, ..., N-1, \tag{4.30}$$

so that the step size is increased with α. We then measure the error of the solution at a grid point neighboring a discontinuity. Hence we are able to determine the value of α for which each method becomes unstable. Our results show that stability is maintained for larger values of α (and correspondingly Δt) when using the ℓ_1 enhanced PDE solver as compared to traditional solvers.

As a final example, we solve an image denoising problem, demonstrating that the ℓ_1 enhanced PDE solver may be used in multiple dimensions, again without explicit knowledge of the jump discontinuities. The results of this two-dimensional example can be found in Chapter 5.

4.4.1 Burgers' Equation

Let $u : (-\pi, \pi) \to \mathbb{R}$ be the solution to

$$\begin{cases} u_t + \left(\frac{1}{2}u^2\right)_x = 0 & x \in (-\pi, \pi), \quad t > 0 \\ u_0(x) = 1 + \frac{1}{2}\sin(x) & x \in (-\pi, \pi), \quad t = 0 \\ u(-\pi, t) = u(\pi, t) & t > 0 \end{cases} \tag{4.31}$$

A shock develops in the solution to (4.31) when the wave breaks at time

$$t_b = -\frac{1}{\min_{x\in[-\pi,\pi]} u_0'(x)} = 2,$$

after which it will continue to re-form at each time step and propagate throughout the domain. In our simulations we advance our solutions to time $t = \pi$ so that the shock will be fully developed. The exact solution for this case is calculated using characteristic tracing, as is done in [139].

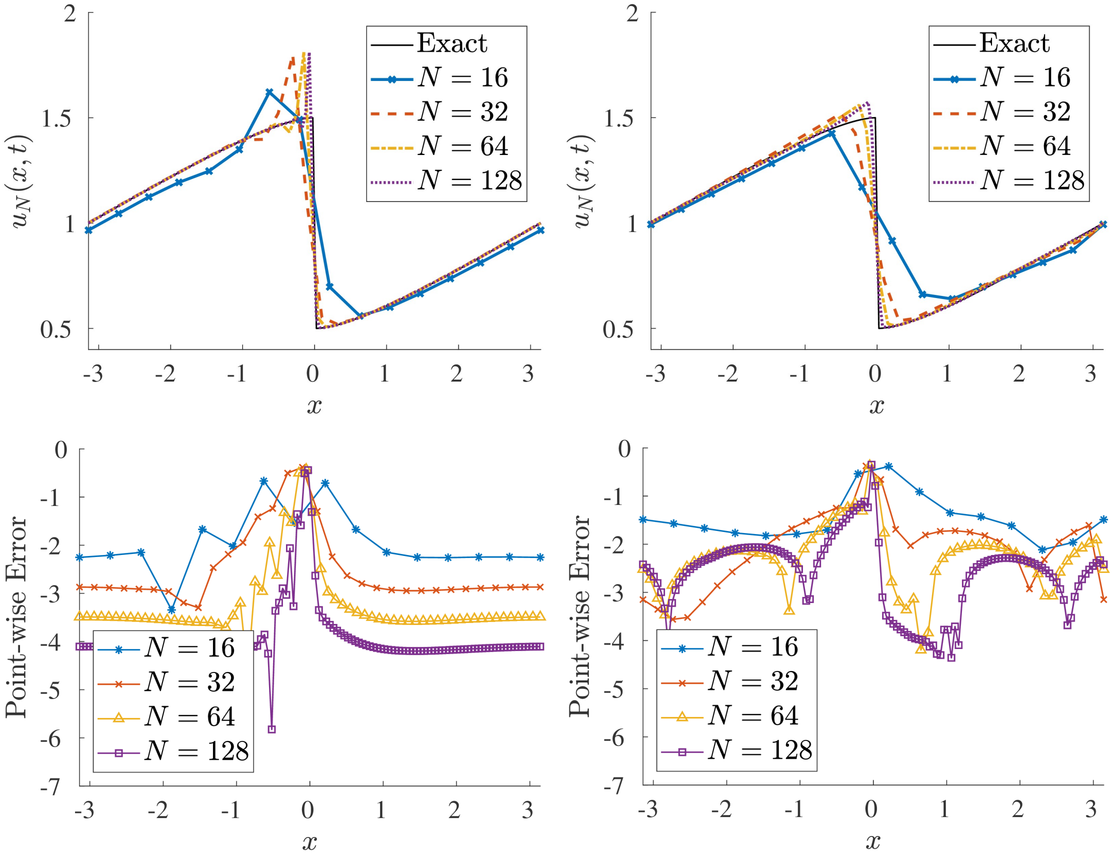

Figure 4.1: Approximation of the solution to (4.31) for $N = 16, 32, 64$ and 128 with $\Delta t = \frac{\Delta x}{2}$ for final time $t = \pi$. (top-left) LW (top-right) ℓ_1 enhanced LW method. Here we used the PA transform with $m = 2$ and $\lambda = .55$. Pointwise errors given in logarithmic scale for (bottom-left) the LW and (bottom-right) the ℓ_1 enhanced LW methods.

Figure 4.1 (top-left) and (top-right) compare the LW scheme solution to (4.31) with and without ℓ_1 regularization. Here we choose $m = 2$ for the PA transform and $\lambda = .55$ in (4.5). The pointwise errors are shown in Figure 4.1 (bottom-left) and (bottom-right). In both cases the time step is chosen as $\Delta t = \frac{\Delta x}{2}$ to ensure stability. As is apparent in Figure 4.1, augmenting the LW method with ℓ_1 regularization improves the accuracy near the shock locations. Away from the shock locations the

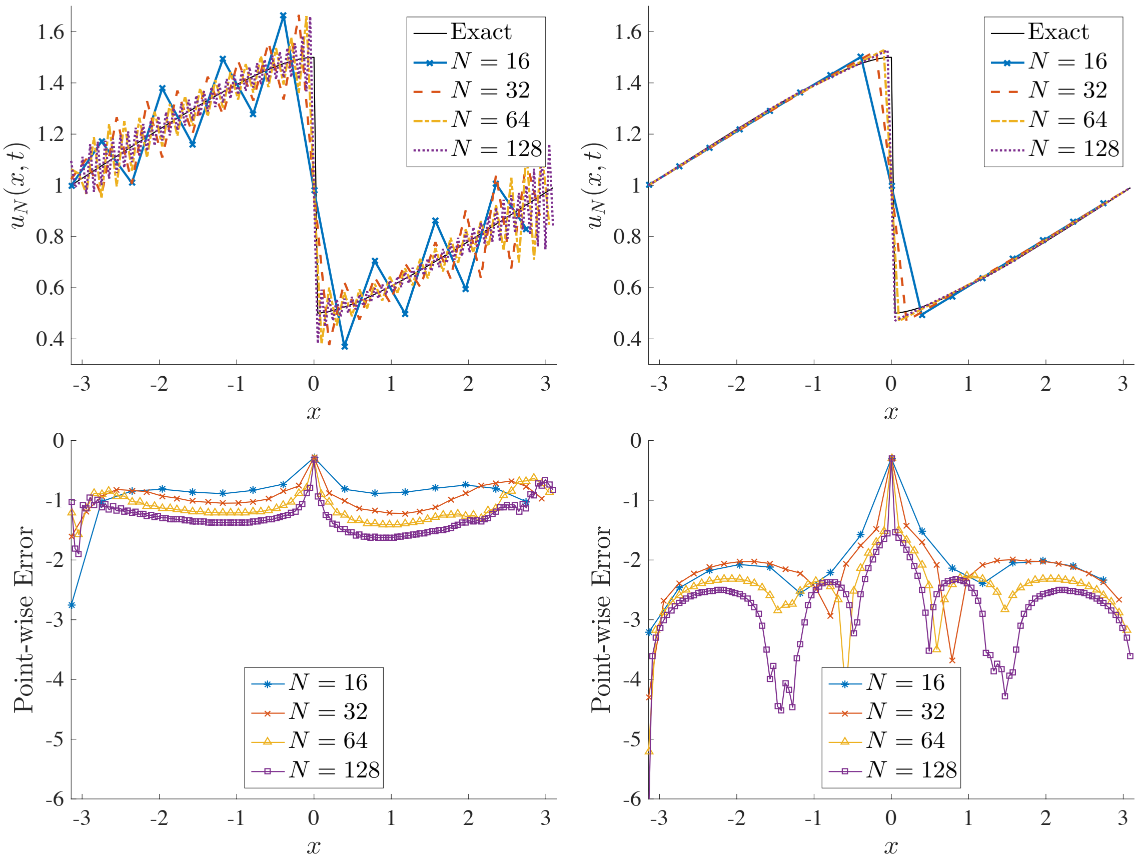

Figure 4.2: Approximation of the solution to (4.31) for $N = 16, 32, 64$ and 128 with $\Delta t = \frac{\Delta x}{2}$ for final time $t = \pi$. (top-left) PS (top-right) ℓ_1 enhanced PS method. Here we used the PA transform with $m = 2$ and $\lambda = .22$. Pointwise errors given in logarithmic scale for (bottom-left) the PS and (bottom-right) the ℓ_1 enhanced PS methods.

accuracy is dictated by the LW approximation.

Figure 4.2 demonstrates that by enhancing the Fourier pseudo-spectral (PS) method with ℓ_1 regularization, we are able to achieve accurate and stable results. As mentioned previously, the fourth order Runge-Kutta time stepping method was used for both the Fourier and Chebyshev cases.

Figure 4.3 compares the approximation to (4.31) using VV (4.12) with and without

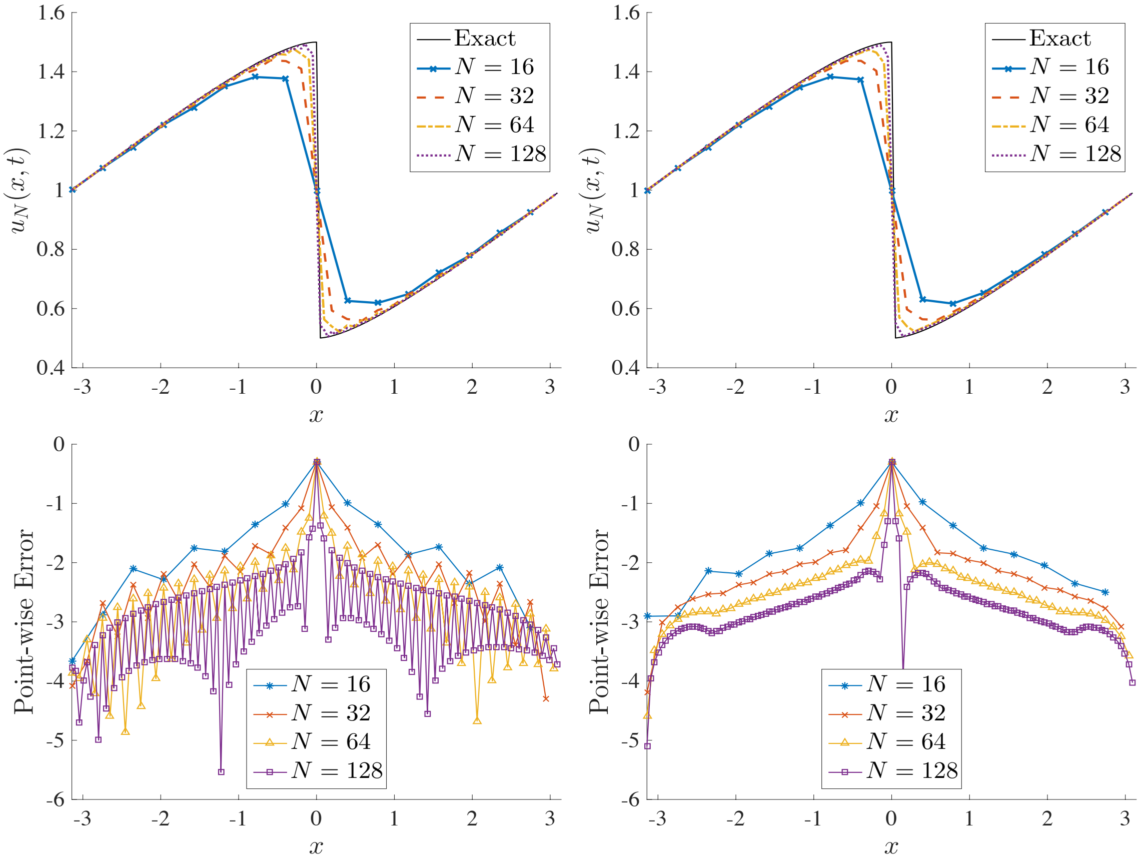

Figure 4.3: Approximation of the solution to (4.31) for $N = 16, 32, 64$ and 128 with $\Delta t = \frac{\Delta x}{2}$ and $\epsilon = \mathcal{O}(\frac{1}{N^2})$ for final time $t = \pi$. (top-left) VV (top-right) VV with ℓ_1 regularization. Here we used the PA transform with $m = 2$ and $\lambda = .12$. Pointwise errors given in logarithmic scale for (bottom-left) VV and (bottom-right) VV with ℓ_1 regularization.

ℓ_1 regularization, for which we chose PA transform order $m = 2$ and regularization parameter $\lambda = .12$. Due to the numerical dissipation present in (4.12), the ℓ_1 enhancement does not appear to significantly improve the results.

The solutions to (4.31) using the pseudo-spectral Fourier method with a tenth order exponential filter (EF), [66], with and without ℓ_1 enhancement are shown in Figure 4.4. Here we used the PA transform with $m = 2$ and $\lambda = .22$ in (4.5). As

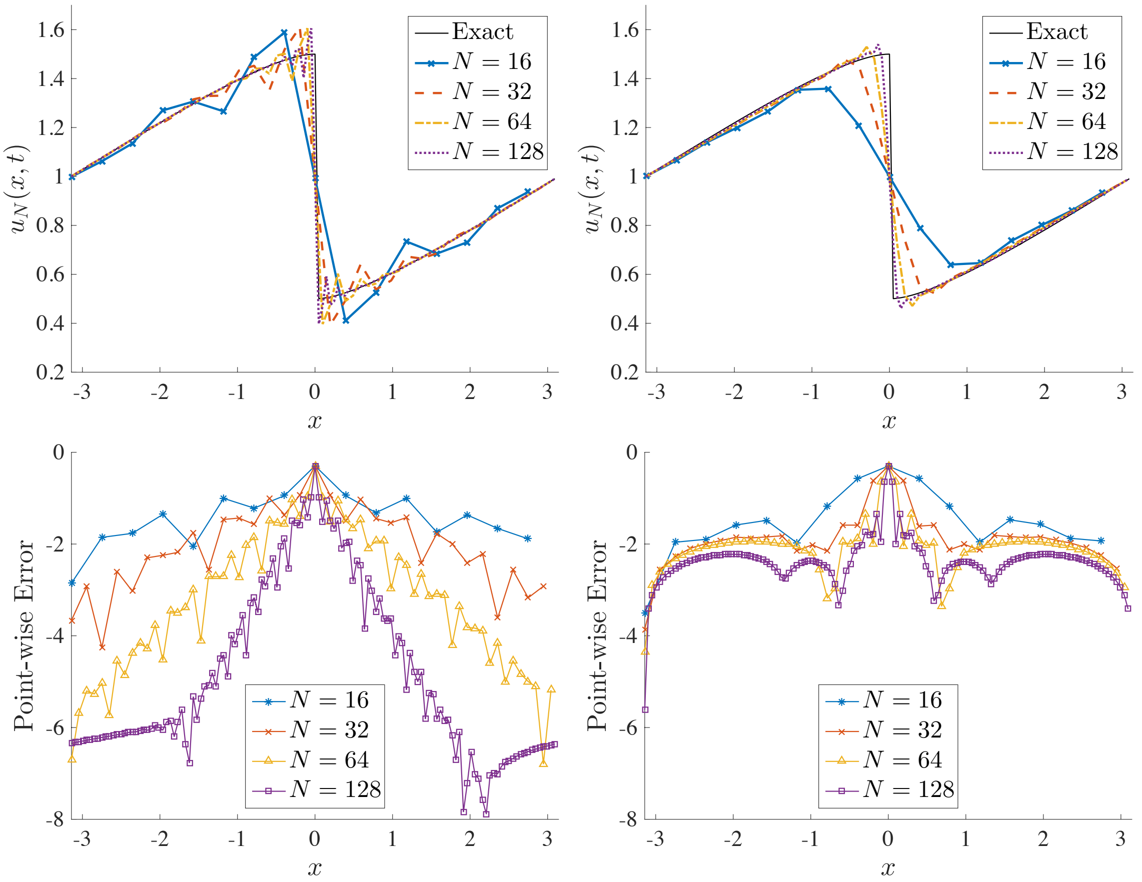

Figure 4.4: Approximation of the solution to (4.31) for $N = 16, 32, 64$ and 128 with $\Delta t = \frac{\Delta x}{2}$ for final time $t = \pi$. (top-left) EF (top-right) EF with ℓ_1 regularization. Here we used the PA transform with $m = 2$ and $\lambda = .22$. Pointwise errors given in logarithmic scale for (bottom-left) EF and (bottom-right) EF with ℓ_1 regularization.

expected, filtering improves the accuracy away from the jump discontinuities, but the ℓ_1 enhancement dramatically reduces the oscillations and overshoot.

Note that no post-processing was applied to any of the ℓ_1 enhancement approximations. It has been shown that spectral reprojection, [59], improves the accuracy in smooth regions of the spectral viscosity method, [130]. However, this requires a priori knowledge of jump discontinuity locations. While the task is manageable in one dimension, it becomes increasingly difficult in multi-dimensions. In future investigations we will use the methods proposed in [134, 4] to post-process the ℓ_1 enhanced solutions. Indeed, Figures 4.1-4.4 demonstrate that in all cases the ℓ_1 enhancement improves resolution and reduces oscillations near shocks. Although the enhancement does not have significant impact on the vanishing viscosity results, it is interesting to note that using ℓ_1 enhancement *directly on the pseudo-spectral method without filtering* yields the best approximation. This is also the most cost efficient choice, and it does not require additional derivative approximations or additional parameter inputs, such as amount of viscosity.

Discussion on accuracy

Table 4.1 provides a complete comparison of the accuracy of approximating the solution to (4.31) with each method before the shock has formed (at a final time $T = 1.5$). This accuracy comparison is also reflected in Figure 4.5.

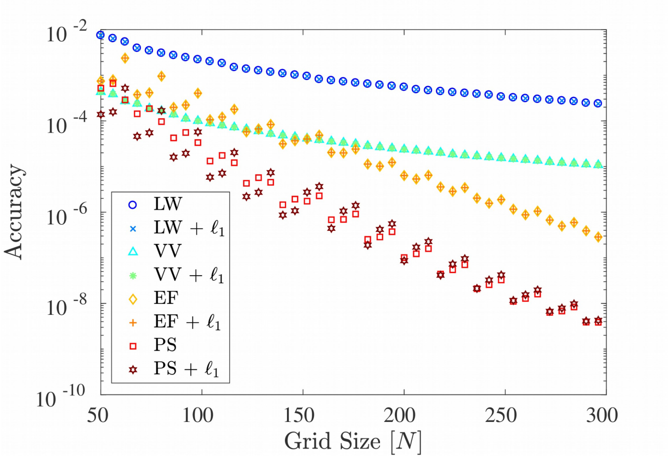

Figure 4.5: $||u(x, 1.5) - u_N(x, 1.5)||_2^2$ calculated for each method as the spatial grid is refined from $N = 50$ to $N = 300$. All other parameters remain consistent with previous experiments.

To estimate the effective order of accuracy, we consider different grid resolutions $N(k)$, $k = 0, \cdots, K$ (see Table 4.2 left-most column). Then for each $N(k)$ we define

Table 4.1: Comparison of $||u(x,1.5) - u_N(x,1.5)||_2^2$ for each method.

N	LW	LW + ℓ_1	PS	PS + ℓ_1
50	5.18E-02	5.19E-02	5.21E-04	1.38E-04
100	7.47E-03	7.47E-03	2.13E-05	8.71E-06
150	2.14E-03	2.14E-03	1.24E-06	7.38E-07
200	9.71E-04	9.71E-04	1.01E-07	8.63E-08
250	5.49E-04	5.49E-04	1.45E-08	1.50E-08
300	2.32E-04	2.32E-04	3.69E-09	3.88E-09

N	VV	VV + ℓ_1	EF	EF + ℓ_1
50	4.31E-04	4.29E-04	7.44E-04	7.46E-04
100	9.63E-05	9.63E-05	1.36E-04	1.36E-04
150	4.23E-05	4.23E-05	2.86E-05	2.86E-05
200	2.37E-05	2.37E-05	6.33E-06	6.34E-06
250	1.51E-05	1.51E-05	1.48E-06	1.48E-06
300	1.05E-05	1.05E-05	3.65E-07	3.66E-07

a corresponding $\Delta(N(k)) = \max|x_{j+1} - x_j|$, $j = 1, \cdots, N(k)$, and compute

$$p(N(k+1)) \approx \log\left(\frac{A(N(k+1))}{A(N(k))}\right) / \log\left(\frac{\Delta(N(k+1))}{\Delta(N(k))}\right), \tag{4.32}$$

where $A(N(k)) = ||u(x,T) - u_{N(k)}(x,T)||_2^2$. Table 4.2 provides these results for $T = 1.5$ and $K = 5$. From this analysis, we see that before shocks are formed, the ℓ_1 enhancement retains the accuracy properties of the non regularized solution.

Discussion on stability

We now assess the stability properties of our method, which are displayed in

Table 4.2: Order of accuracy p for each method given by (4.32).

N	LW	LW + ℓ_1	PS	PS + ℓ_1
50	-	-	-	-
100	1.7788	1.7789	4.6126	3.9895
150	1.9301	1.9301	7.0198	6.0851
200	1.9743	1.9743	8.7084	7.4635
250	2.2400	1.2400	8.7039	7.8380
300	1.9485	1.9485	7.5057	7.4120

N	VV	VV + ℓ_1	EF	EF + ℓ_1
50	-	-	-	-
100	2.1630	2.1566	2.4497	2.4515
150	2.0278	2.0278	3.8479	3.8500
200	2.0128	2.0128	5.2422	5.2430
250	2.0073	2.0073	6.5215	6.5214
300	2.0047	2.0047	7.6667	7.6651

Figure 4.6. Here we consider four different numerical solvers with and without the ℓ_1 enhancement for a grid size of $N = 128$. For each solution, for various choices of α in (4.30), we calculate the error at location $x = -0.2454$, which is four grid points behind the shock at time $t = \pi$. Observe that the ℓ_1 enhancement yields better accuracy for larger values of Δt than the standard solvers. Figure 4.6 also displays the final ℓ_1 enhanced solutions for $\alpha = 2$, so as to show that it is possible to maintain accuracy and stability for suitable time stepping constraints when using a high order method in the fidelity term of (4.5).

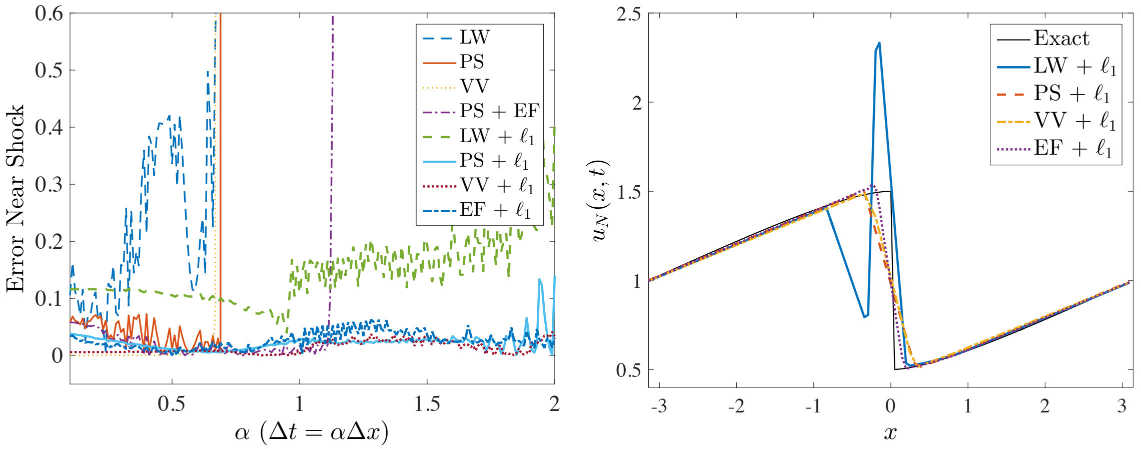

Figure 4.6: (left) Stability analysis results. (right) The ℓ_1 enhanced solutions for a value of $\alpha = 2$ at final time $t = \pi$ for $N = 128$.

Discussion on conservation

We now study the conservation properties of our method specifically to ensure convergence to a weak solution of (4.31). Recall that a method is conservative if it can be written as [84]:

$$u(x_j, t_{n+1}) = u(x_j, t_n) - \frac{\Delta t}{\Delta x}\left[f(u(x_{j+1}, t_n)) - f(u(x_j, t_n))\right]. \tag{4.33}$$

Globally, (4.33) implies that

$$\sum_{j=1}^{N} u(x_j, t_{n+1}) = \sum_{j=1}^{N} u(x_j, t_n) - \frac{\Delta t}{\Delta x}\sum_{j=1}^{N}\left[f(u(x_{j+1}, t_n)) - f(u(x_j, t_n))\right]. \tag{4.34}$$

While we cannot prove that (4.34) holds in general, Figure 4.7, which plots the residual of (4.34) over all time steps for each method considered for $0 \leq t \leq \pi$, suggests that our method is indeed conservative. In particular we see that for Burgers' equation, each method maintains the conservation throughout the entire time integration domain considered. [4]

[4] We did not separately analyze our method for Euler's equations, (4.37).

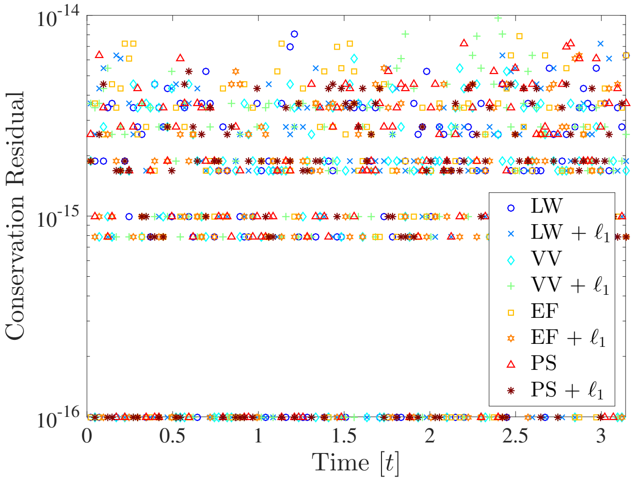

Figure 4.7: The residual of (4.34) calculated after each time step for all techniques considered in the time domain $0 \leq t \leq \pi$.

As an additional tool to measure the conservation properties of our method, we analyze energy conservation. We define energy at each time steb by ([71])

$$E_n = \frac{1}{2}||u_N(x, t_n)||_2^2, \tag{4.35}$$

with the rate of change of energy approximated as

$$\frac{dE}{dt} \approx \frac{E_{n+1} - E_n}{\Delta t}. \tag{4.36}$$

Figure 4.8 displays the result of calculating the energy and rate of change of energy for all methods considered in this chapter. We see that energy is conserved before time $t = 2$, when the shock is introduced. As expected, energy is lost after the shock forms. Observe that in all cases, including the ℓ_1 term does not cause any reduction of energy from the original scheme.

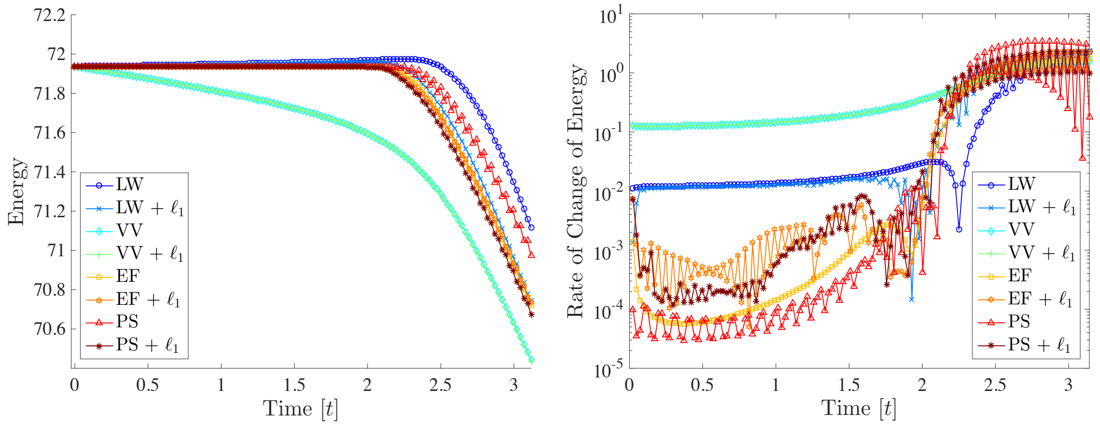

Figure 4.8: (left) Energy (4.35) and (right) rate of change of energy (4.36) calculated for each numerical method considered for $0 \leq t \leq \pi$.

4.4.2 Euler's Equations

The Riemann problem for the one dimensional Euler equations is given as

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho q\\ E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho q\\ \rho q^2 + P\\ (E+P)q\end{pmatrix} = 0, \tag{4.37}$$

where $\rho : (a,b) \to \mathbb{R}$ is density, $q : (a,b) \to \mathbb{R}$ is velocity and $E : (a,b) \to \mathbb{R}$ is the total energy. The pressure $P : (a,b) \to \mathbb{R}$ is related to the conserved quantities through the equation of state:

$$P = (\gamma - 1)\left(E + \frac{\rho q^2}{2}\right),$$

with $\gamma = 1.4$ defined as the ratio of specific heat constants. The set of initial conditions we consider describe Sod's shock tube problem, [35], and are given by

$$\begin{aligned}\rho_0(x) &= 1, \;\; q_0(x) = 0, \;\; P_0(x) = 1, \text{ when } x \in [-1,0)\\ \rho_0(x) &= 0.125, \;\; q_0(x) = 0, \;\; P_0(x) = 0.1, \text{ when } x \in [0,1]. \end{aligned} \tag{4.38}$$

We demonstrate the result of approximating the solution to (4.37) with (4.38) using the two-step Lax-Wendroff method (4.6) with and without regularization. The stability condition for this numerical scheme is

$$\max_i\{|\zeta_i|\}\frac{\Delta t}{\Delta x} \leq 1,$$

where ζ_i, $i = 1, 2, 3$ are the eigenvalues of the Jacobian matrix $\partial f/\partial u$, regarded as the propagation speeds of the corresponding characteristic waves. It can be shown [84] that for the Euler system of equations, the stability condition leads to an adaptive time step given by

$$\Delta t = \text{CFL}\frac{\Delta x}{|q| + a}, \quad \text{CFL} < 1 \tag{4.39}$$

where $a = \sqrt{\gamma P/\rho}$ is the local speed of sound. We chose CFL $= .8$ for our numerical experiments using the Lax-Wendroff method and adjust the time stepping for stability for our numerical experiments using the Chebyshev technique.

Figure 4.9 compares solutions using the LW and ℓ_1 enhanced LW methods for density ρ, velocity q and pressure P when $N = 256$. While it is evident that the ℓ_1 enhancement reduces the size of the overshoots and oscillations that occur near discontinuities, the LW method appears to be too dispersive for the ℓ_1 enhancement to be very effective.

Figure 4.10 displays the approximation results using the filtered (mapped) Chebyshev method (4.19). The approximation results using the ℓ_1 enhanced Chebyshev method (4.21) are also displayed. In the non-regularized version, a 16th order exponential filter is implemented to the solution after each time step to ensure stability. No additional filtering is needed in the ℓ_1 enhancement case. Time stepping was implemented using fourth order Runge Kutta, with the time step Δt chosen according to the stability requirements in [78]. We approximate the solution at resolutions $N = 64$, 128, 256 and 512. In each case, we use PA transform order $m = 2$ and regularization

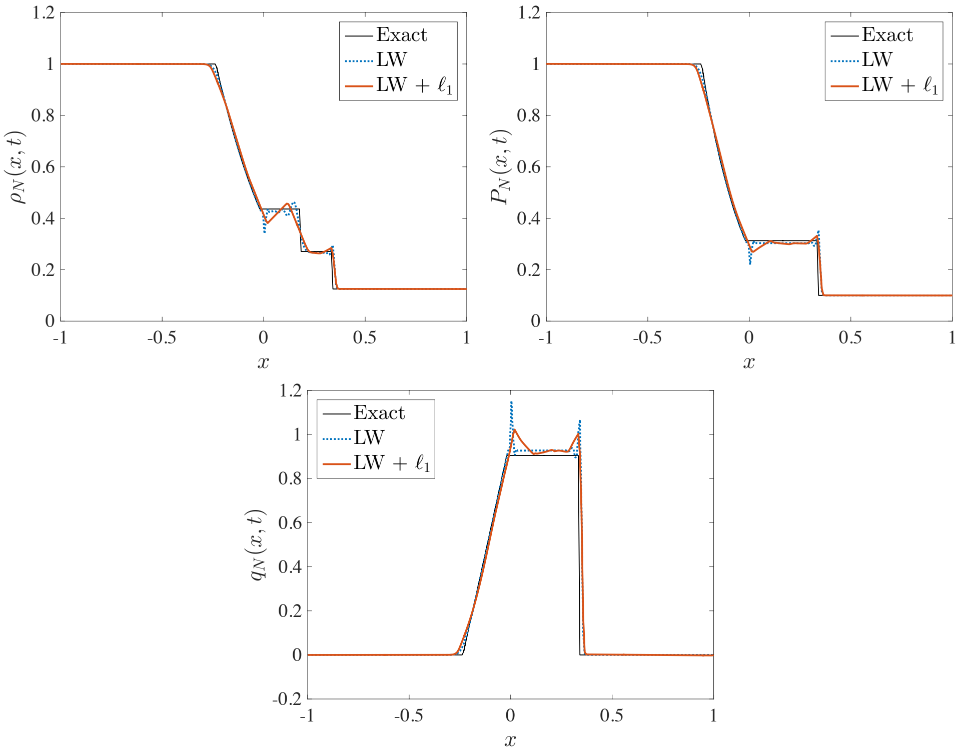

Figure 4.9: (top-left) Density (top-right) Pressure (bottom) Velocity. Here, for the ℓ_1 enhancement, we used the PA transform with $m = 2$ and $\lambda = .35$. The final time is $t = .2$ and CFL $= .8$.

parameter $\mu = 100$. The parameter β varies with N, with $\beta = .75$ for $N = 64$, 256, $\beta = 1.25$ for $N = 128$, and $\beta = .5$ for $N = 512$. Although these values for β represent the "best" results, in general the choice of β did not greatly affect the quality of the results, only the speed of algorithm convergence, as expected. Future investigations will consider parameter optimization To obtain the results in Figure 4.10, Algorithm 6 requires 5 outer and 5 inner iterations. Hence there is a maximum of 25 iterations per time step. The efficiency of Algorithm 6 ensures that no additional significant computational time is required. Figure 4.11 displays the pointwise errors associated with estimating the final density for various resolutions with and without ℓ_1 enhancement. As noted previously, due to the low order of accuracy, ℓ_1 enhancement does

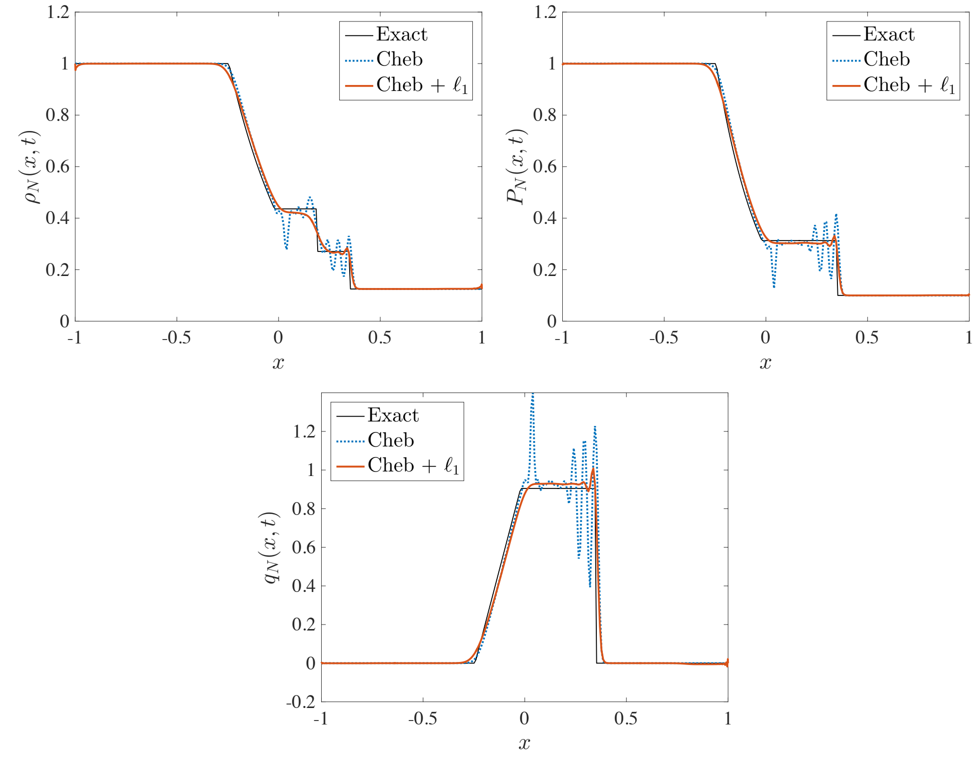

Figure 4.10: (top-left) Density (top-right) Pressure (bottom) Velocity. For the ℓ_1 enhancement we used the PA transform with $m = 2$. The final time is $t = .2$ and $\Delta t = \frac{1}{10N}$.

not significantly improve the results in the LW case. However, the ℓ_1 enhancement does dramatically improve the resolution properties near the discontinuities in the Chebyshev case. Future investigations will include studies on post-processing these solutions.

4.5 Discussion and Conclusions

In this chapter we introduced a method for solving non-linear partial differential equations using ℓ_1 regularization, and specifically, using the polynomial annihilation (PA) transform operator in the ℓ_1 term. Our results demonstrate that it is possible to efficiently implement a method that is accurate and resolves shock discontinuities.

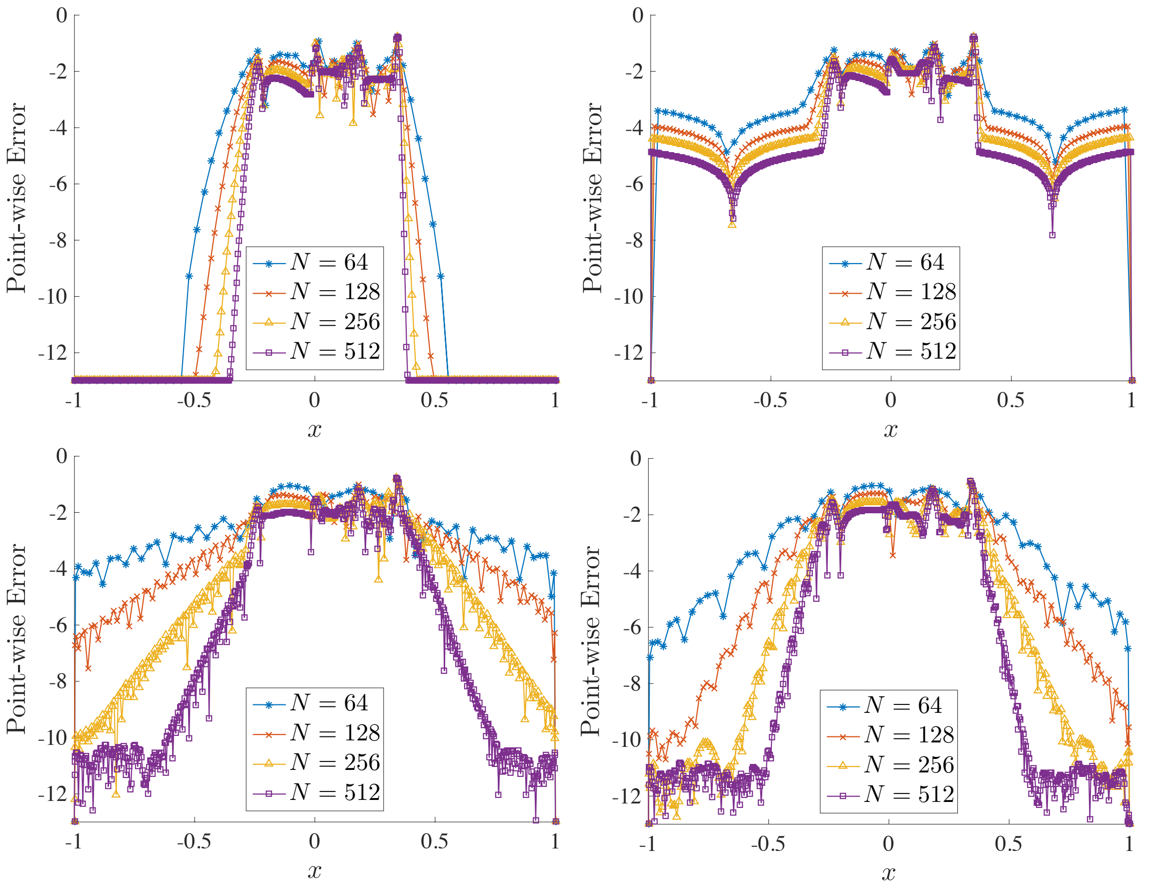

Figure 4.11: Pointwise error in approximating the solution to Sod's shock tube problem using the (top) LW method (bottom) Chebyshev method, (left) without regularization and (right) with the ℓ_1 enhancement.

The method does not require advance knowledge of the shock locations.

Our new method is particularly useful for numerically solving hyperbolic partial differential equations that develop shocks or discontinuities. The ℓ_1 regularization enhanced method exploits the knowledge that there are a sparse number of singularities in the solution. Our method is made numerically efficient by employing the alternating direction of multipliers algorithm. We used our technique to approximate the solution to Burgers' equation with a smooth initial condition and Euler's equations with initial conditions describing Sod's shock tube. A two-dimensional PDE, often employed for denoising imagery corrupted with multiplicative gamma noise, is explored in Chapter 5. In all cases, our new method showed improved accuracy near the shock locations. These results are obtained without post-processing or the need for shock tracking. However, post-processing may further improve our results. In addition, our method remains stable for larger time steps than those typically used by conventional solvers. In future investigations we will study how to optimize the parameters of our method as well as develop a more rigorous understanding of stability conditions.

Other areas to explore include using different PDE solvers, such as the discontinuous Galerkin method, in the data fidelity term. We anticipate that this will help localize the effect of smoothing to only those cells that contain discontinuities. Also beneficial could be the inclusion of time dependent regularization to enforce the knowledge that the solutions do not vary significantly from one time step to the next. Both topics will be explored in future studies.

Chapter 5

INCORPORATION OF REGULARIZATION TECHNIQUES FOR SYNTHETIC APERTURE RADAR IMAGE FORMATION

This chapter incorporates regularization techniques into traditional synthetic aperture radar (SAR) image formation procedures. A thorough discussion of SAR and current two-dimensional SAR image formation techniques is given in Appendix A. Two main sources of error corrupt the quality of SAR images and hinder the performance of target recognition and detection algorithms: (i) *speckle*, which is a salt and pepper like, granular noise and (ii) phase error, which results from not precisely knowing the imaging platform location. This error manifests as a phase error on the data.

Speckle is inherent in any coherent imaging system. It corrupts the quality of images throughout the entire imaging domain, and is typically modeled as a multiplicative noise. Current speckle reduction agorithms are discussed in Section 5.1.1, while the mathematical description of speckle, which is based on statistical arguments, is provided in Section 5.1.2. We then develop regularization methods to reduce speckle in SAR imagery using two different techniques: (i) by enhancing a currently used speckle reduction partial differential equation (PDE) model with regularization using the techniques developed Chapter 4, and (ii) by employing the variance based joint sparsity (VBJS) technique developed in Chapter 3. We first explored the techniques mentioned in (i) in [124]. Numerical experiments are provided there to demonstrate how incorporating prior knowledge about the variation in the underlying image into the speckle reduction model is beneficial. In what follows, we compare the results of the new methods in this thesis to the high-order total variation (HOTV) model for speckle reduction introduced in [120].

Correcting for the phase error caused by inexact information about the imaging platform location is called *autofocusing.* We review some current autofocusing techniques (mostly post-processing procedures) in Section 5.2.1, and discuss the need for better algorithms. In Section 5.2.2 we explain the source of this error. We then present a new phase error correction method, based on phase synchronization, which uses phase retrieval methods to jointly optimize for the underlying image while depressing the phase errors through an iterative, alternating optimization technique. We originally explored the joint optimization technique in [121], where correlations in the phase errors were exploited. The new phase synchronization technique introduced here improves upon these previous results, which is demonstrated by numerical examples in Section 5.2.5. A discussion of future work in this area is provided in Section 5.2.7.

5.1 Speckle in SAR

In observing multiple synthetic aperture radar (SAR) images of the same scene, it is apparent that the brightness distributions of the images are not smooth, but rather composed of complicated granular patterns of bright and dark spots. Further, these brightness distributions vary from image to image. This salt and pepper like feature of SAR images is called *speckle.* Even though there is only one scene being reconstructed, the acquired images are independent, so that the observed speckle is uncorrelated across them. Hence modeling speckle is difficult.

Speckle occurs in any form of coherent imaging where objects being illuminated have surface features that are rough on the microscopic scale of the illuminating wavelength. In SAR the scale is considered microscopic because, for example in Figure 5.2(left), the transmitted wavelength is approximately 0.03m while a resolution cell is approximately .3m in length (i.e. the wavelength is an order of magnitude smaller than the resolution cell). When a SAR system sends out a continuous chirp, the wave that is reflected back from the rough surface then contains contributions from many different scattering points or areas, all of which are subject to random phase delays. Therefore, at any given point in the image plane, the image will consist of a combination of many amplitude point spread functions that arise from the different scattering points on the surface of the scene. Because the phases of each path are highly varying, they may interact (sum) constructively or destructively. Thus speckle is the result of *random phasor sums* from many scattering centers within a given resolution cell. A simple comparison of constructive and destructive interference is found in Figure 5.1.

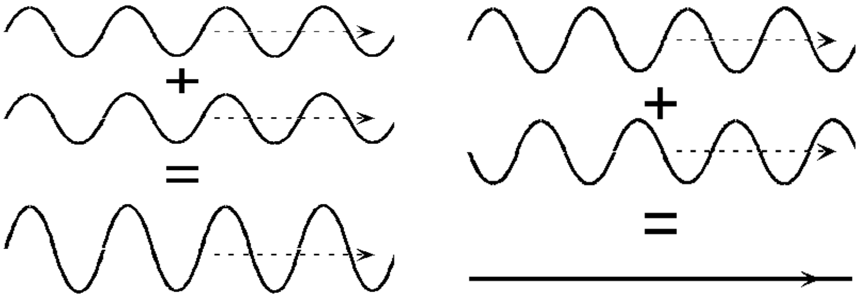

Figure 5.1: Diagram depicting the difference between constructive (left) and destructive (right) phasor sums [53].

5.1.1 Current Speckle Reduction Techniques

As is evident in Figure 5.2, speckle reduces the contrast in SAR images, which has a negative effect on texture based image analysis. It also alters the spatial statistics of the underlying scene backscatter which makes image classification difficult [97]. Consequently, in addition to diminishing the performance of both automated scene analysis and information extraction techniques, speckle may be harmful in applica-

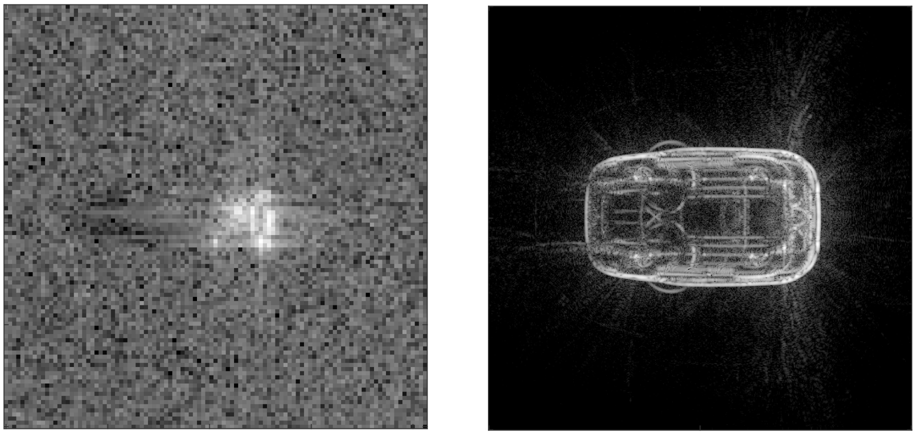

Figure 5.2: (left) Image of a T-72 tank from the measured MSTAR data set [122]. (right) Image of a Jeep 93' from the synthetic (computer generated) Civilian Vehicle Data Dome data set [116]. Notice that speckle corrupts both measured and synthetic SAR imagery.

tions requiring multiple SAR observations, such as automatic multi-temporal change detection. Finally, along with the complications that naturally arise in registering the different acquisition geometries of optical and SAR systems, speckle poses a major impediment towards the development of an effective optical-SAR fusion. [6].

Most speckle reducing methods are post-processing techniques that take advantage of the multiplicative noise model, as will be discussed in Section 5.1.2. Generally speaking, speckle noise can be reduced by multi-look processing or filtering [6, 97]. While computationally inexpensive to process, such techniques often result in over-smoothing the fine details in the image. Another disadvantage of these methods is that multiple looks of a scene are often infeasible due to data collection and mission constraints. Post-processing and image formation techniques have also used approaches based on regularization and compressive sensing [103, 6], but these methods often do not incorporate known statistical properties of speckle into the model, however. A final class of often employed denoising techniques in SAR are based on variational methods [30, 9, 94, 8, 118], where the image is often considered to be the steady state solution to an Euler-Lagrange partial differential equation. Though computationally expensive, variational methods can consider the multiplicative speckle model and the statistical properties of the noise. Currently the variational approach is only employed as a post-processing procedure. In any case, when despeckling, it is essential to avoid throwing away any useful information such as the local mean of the backscatter, point target location information, linear features and textures. More information on speckle in SAR can be found in [6, 83, 93, 97, 103, 101, 150].

5.1.2 Mathematical Modeling of Speckle Noise

Understanding speckle requires a detailed examination of the properties of electromagnetic waves after they have been reflected or scattered from rough objects [113]. Because of the difficulty in understanding such details of the small scale structures of the complex wavefronts leaving the objects, it is common to instead generate a statistical speckle model. Such models predict the statistical properties of intensity over an ensemble of different rough surfaces with the same macroscopic properties, but different in microscopic detail. Summarized below is a description of the multiplicative speckle model typically used to describe the salt and pepper phenomenon that characterizes speckle [6, 97, 30, 9, 94, 8, 118].

Important to the discussion are the definitions of *amplitude*, *intensity* and *brightness* of the image:

Definition 5.1.1 *Let $\Omega \in \mathbb{R}^2$ be some bounded rectangular domain and $f : \Omega \to \mathbb{C}$ be the complex reflectivity measured by the radar. The* amplitude *refers to the magnitude $|f|$, the* image intensity *g is given by $g = |f|^2$, and the* brightness *refers to the scaling of amplitude typically used for display purposes. In SAR applications, the brightness* *is often defined in dB scale as*

$$20\log_{10}\left(\frac{|f(x,y)|}{\max_{x,y}|f(x,y)|}\right).$$

The amplitude of the homogeneous regions of the SAR reflectivity with fully developed speckle is modeled by the Rayleigh density function. Note that all probability density functions below hold for non-negative random variables and that speckle is only considered fully developed in homogeneous regions that do not include returns from targets. Specifically, if for each pixel $i = 1, \cdots, N^2$, of the SAR reflectivity function $f \in \mathbb{C}^{N\times N}$ we define $\alpha \in \mathbb{R}^{N\times N}$ and $\beta \in \mathbb{R}^{N\times N}$ such that $\alpha_i = \mathrm{Re}(f_i)$ and $\beta_i = \mathrm{Im}(f_i)$ (so that $|f_i| = \sqrt{\alpha_i^2+\beta_i^2}$), then their joint probability density function follows a Gaussian distribution given by

$$p_{\alpha_i,\beta_i}(\alpha_i,\beta_i) = \frac{1}{\pi\sigma_i}\exp\left(-\frac{\alpha_i^2+\beta_i^2}{2\sigma_i}\right). \tag{5.1}$$

The only unknown in (5.1), σ_i, represents the radar cross section at pixel i. From (5.1) it is evident that the amplitude $|f|$ obeys the Rayleigh distribution

$$p_{|f_i|}(|f_i|) = \frac{2|f_i|}{\sigma_i}\exp\left(-\frac{|f_i|^2}{\sigma_i}\right), \tag{5.2}$$

at each pixel $i = 1, \cdots, N^2$. Moreover, each speckled image intensity value $g_i = |f_i|^2$ follows a negative exponential law

$$p_{g_i}(g_i) = \frac{1}{\sigma_i}\exp\left(-\frac{g_i}{\sigma_i}\right) \tag{5.3}$$

with mean σ_i and variance σ_i^2. If $\nu \in \mathbb{R}^{N\times N}$ defines the speckle intensity, then it also follows that ν obeys the negative exponential law given by [6]

$$p_{\nu_i}(\nu_i) = e^{-\nu_i}. \tag{5.4}$$

Finally, the speckle intensity can equivalently be written as the *multiplicative noise* model [6]

$$g = |f|^2 = \nu\sigma \approx \nu f. \tag{5.5}$$

We can confirm (5.5) since it yields (along with (5.4)) that

$$p_{g_i}(g_i) = \frac{1}{\partial g_i/\partial \nu_i} \cdot p_{\nu_i}(\nu_i = g_i/\sigma_i) = \frac{1}{\sigma_i} \exp\left(-\frac{g_i}{\sigma_i}\right),$$

i.e., we get back (5.3). Observe that if we are able to extract the unknown radar cross section σ from (5.5) then we will be able to fully characterize f using its first order statistics (5.1)-(5.3).

Multi-look data collection

Often there are L independent looks of the same scene collected and combined, yielding a mean speckled image intensity

$$g_L = \frac{1}{L}\sum_{k=1}^{L} g_k,$$

where each g_k follows the density function defined in (5.3). It can be shown that at each pixel location, the multilook intensity obeys the gamma distribution given by

$$p_{g_L}(g) = \left(\frac{L}{\sigma}\right)^L \frac{1}{\Gamma(L)} g^{L-1} \exp\left(-\frac{Lg}{\sigma}\right). \tag{5.6}$$

Note that here the dependence on pixel location i has been dropped for clarity. In this case, the speckle intensity can then be described as

$$p_\nu(\nu) = L^L \frac{1}{\Gamma(L)} \nu^{L-1} \exp\left(-L\nu\right). \tag{5.7}$$

Note that (5.6) and (5.7) reduce to (5.3) and (5.4) respectively in the single look case ($L = 1$) and that in each case $E[\nu] = 1$.

The derivations of the above probability density functions are discussed in detail in Appendix B.

5.1.3 Proposed Despeckling Techniques

In this section we propose two novel despeckling techniques. The first is an ℓ_1 enhancement of a PDE-based multiplicative noise removal technique. We utilize the methods developed in Chapter 4 to augment the solution to the variational model proposed in [9], which incorporates the statistical properties of speckle into the model. In this way, we are able to better predict the speckle behavior while enforcing sharp transitions and edges in imagery. The second method we propose is an image formation procedure adapted for SAR image formation from the variance based joint sparsity (VBJS) techniques developed in Chapter 3. We incorporate the multiplicative speckle model (5.5) into the data acquisition process and test our algorithms on both synthetic data sets and SAR phase history data sets.

PDE Based Method

In [9], maximum a-posteriori probability (MAP) estimation theory is used in generating the variational model

$$\min_{f\in\Omega}\int_{\Omega}\left\{|D(f)|+\eta\left(\log f+\frac{g}{f}\right)\right\}. \tag{5.8}$$

Here $\eta > 0$ is the inherent model parameter, f, g and ν follow the multiplicative noise model defined in (5.5), and we consider the possibility of multiple looks (i.e., (5.6)-(5.7)). We will refer to (5.8) the *AA model* and note that $\int|D(f)|$ represents the total variation of f.

As is classically done in image denoising, the solution to (5.8) is computed by embedding the integral equation into a dynamical system using the Euler-Lagrange equation, which is driven to a steady state, [119]. The Euler-Lagrange equation is a second-order PDE with solutions that are functions for which a given functional is stationary. Because a differentiable function is stationary at local minima and maxima, this is a useful tool for optimization. The time dependent Euler-Lagrange equation associated with solving the AA model (5.8) is

$$\begin{cases} \frac{\partial f}{\partial t} = \frac{1}{\eta}\text{div}\left(\frac{\nabla f}{\sqrt{\epsilon^2+|\nabla f|^2}}\right) + \frac{g-f}{f^2} & \text{in} \quad \Omega \\ \frac{\partial f}{\partial \nu} = 0 & \text{on} \quad \partial\Omega \end{cases}, \tag{5.9}$$

with given initial conditions. To remove the possible singularity when $|\nabla f| = 0$, ($f > 0$ so $f^2 \neq 0$) the model has been relaxed with $\epsilon > 0$, where we define

$$\text{div}\left(\frac{\nabla f}{\sqrt{\epsilon^2+|\nabla f|^2}}\right) := \frac{\partial}{\partial x}\left(\frac{f_x}{\sqrt{\epsilon^2+f_x^2+f_y^2}}\right) + \frac{\partial}{\partial y}\left(\frac{f_y}{\sqrt{\epsilon^2+f_x^2+f_y^2}}\right).$$

As time increases, the energy in (5.8) will decrease leading to a denoised image at steady state.

To discretize the solution to (5.9), for $0 \leq i, j \leq N$, let

$$x_i = i\Delta x, \quad y_j = j\Delta x,$$

where Δx is the pixel size. A forward Euler time stepping scheme was proposed in [9, 119] to solve (5.9) and is given by

$$\begin{aligned} \frac{f_{i,j}^{n+1} - f_{i,j}^n}{\Delta t} = \frac{f_{i,j}^n - g_{i,j}^n}{\left(f_{i,j}^n\right)^2} &+ \frac{c_1}{2\eta\Delta x^2}\left(f_{i+1,j}^n - f_{i,j}^{n+1}\right) - \frac{c_2}{2\eta\Delta x^2}\left(f_{i,j}^{n+1} - f_{i-1,j}^n\right) \\ &+ \frac{c_3}{2\eta\Delta x^2}\left(f_{i,j+1}^n - f_{i,j}^{n+1}\right) - \frac{c_4}{2\eta\Delta x^2}\left(f_{i,j}^{n+1} - f_{i,j-1}^n\right), \end{aligned} \tag{5.10}$$

where

$$
\begin{aligned}
c_1 &= \frac{1}{\sqrt{\epsilon^2 + \left(\frac{f^n_{i+1,j}-f^n_{i,j}}{\Delta x}\right)^2 + \left(\frac{f^n_{i,j+1}-f^n_{i,j}}{\Delta x}\right)^2}},\\
c_2 &= \frac{1}{\sqrt{\epsilon^2 + \left(\frac{f^n_{i,j}-f^n_{i-1,j}}{\Delta x}\right)^2 + \left(\frac{f^n_{i-1,j+1}-f^n_{i-1,j}}{\Delta x}\right)^2}},\\
c_3 &= \frac{1}{\sqrt{\epsilon^2 + \left(\frac{f^n_{i+1,j}-f^n_{i,j}}{\Delta x}\right)^2 + \left(\frac{f^n_{i,j+1}-f^n_{i,j}}{\Delta x}\right)^2}},\\
c_4 &= \frac{1}{\sqrt{\epsilon^2 + \left(\frac{f^n_{i+1,j-1}-f^n_{i,j-1}}{\Delta x}\right)^2 + \left(\frac{f^n_{i,j}-f^n_{i,j-1}}{\Delta x}\right)^2}}.
\end{aligned}
$$

for interior grid points $1 \leq i, j, \leq N-1$ and time step Δt. The boundary conditions imposed are

$$
\begin{aligned}
f_{0,j} &= f_{1,j}, \;\; f_{N,j} = f_{N-1,j}, \;\; f_{i,0} = f_{i,1}, \;\; f_{i,N} = f_{i,N-1},\\
f_{0,0} &= f_{1,1}, \;\; f_{0,N} = f_{1,N-1}, \;\; f_{N,0} = f_{N-1,1} \text{ and } f_{N,N} = f_{N-1,N-1}.
\end{aligned}
$$

The PDE is advanced in time for $n = 0, ..., N_t$. We note that using higher order temporal and spatial derivative approximations might yield higher accuracy, especially when using the ℓ_1 regularization enhancement. We will explore this idea in future investigations.

To reduce speckle and preserve the edges present in images, we enhance the Euler-Lagrange PDE associated with the AA model with ℓ_1 regularization using the PA transform (2.5). To use the technique described in Chapter 4, we first solve (5.10) for $f^{n+1}_{i,j}$ and define $B^n_{i,j}$ for $1 \leq i, j \leq N-1$ to be the right hand side of of the resulting expression

$$
\begin{aligned}
B^n_{i,j} := \frac{1}{\Lambda}\Big[f^n_{i,j} + \Delta t \frac{f^n_{i,j} - g_{i,j}}{\left(f^n_{i,j}\right)^2} &+ \frac{c_1 \Delta t}{2\eta \Delta x^2} f^n_{i,j-1} + \frac{c_2 \Delta t}{2\eta \Delta x^2} f^n_{i-1,j}\\
&+ \frac{c_3 \Delta t}{2\eta \Delta x^2} f^n_{i,j+1} + \frac{c_4 \Delta t}{2\eta \Delta x^2} f^n_{i,j-1}\Big],
\end{aligned} \tag{5.11}
$$

where

$$\Lambda = 1 + \frac{\Delta t}{2\eta \Delta x^2} \left(c_1 + c_2 + c_3 + c_4 \right).$$

Analogous to (4.4), we write

$$b^n := \begin{bmatrix} B^n_{1,1} & B^n_{1,2} & \cdots & B^n_{1,N-1} \\ B^n_{2,1} & B^n_{2,2} & \cdots & B^n_{2,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ B^n_{N-1,1} & B^n_{N-1,2} & \cdots & B^n_{N-1,N-1} \end{bmatrix}.$$

This matrix is embedded into (4.5) as

$$U^{n+1} = \operatorname*{argmin}_V \left\{ ||\mathcal{L}^m V||_1 + \frac{\lambda}{2} ||V - b^n||_2^2 \right\}, \tag{5.12}$$

to determine the solution at time t_{n+1} at the interior grid points. Again $\mathcal{L}^m$ is the mth order PA operator (2.5). After a sufficiently large number of time steps N_t, $U^{N_t} \to f$. That is, the numerical solution will approach the despeckled image f in (5.5). The ADMM algorithm for enhancing numerical PDE solvers with ℓ_1 regularization, Algorithm 6 of Chapter 4, is employed to solve (5.12).

Remark 5.1.2 *Algorithm 6 is currently used in this case as a* post-processing *method to reduce speckle from an initial approximation of the image intensity g in (5.5). In actuality, we do not have g, but rather some SAR phase history data. A better approach would be to use Algorithm 6 directly on the SAR phase history, so that information is not lost in the initial approximation of the underlying image. This will be the topic of future work.*

VBJS Weighted Regularization Approach

As an alternative to the PDE approach, we propose using the VBJS technique, developed in [2] and expanded in Chapter 3, as an effective way to reduce speckle in

SAR. As described in Chapter 3, VBJS is designed to exploit joint sparsity across different realizations of the same image while at the same time reducing the effect of bad information that is not common to all data sets. Thus the VBJS technique can be especially useful in reducing speckle, because the multiple looks of a scene have joint sparsity structure and yield highly uncorrelated speckle.

To describe how VBJS can be applied to the speckle reduction problem let us consider a piecewise smooth image $f(x,y)$ on $[a,b]^2$ in (5.5). To generate J measurement vectors corresponding to $\boldsymbol{f} : \mathbb{R}^{N\times N} \to \mathbb{C}$, the function is sampled on a uniform grid as $f_{i,l} = f(x_i, y_l)$, where

$$x_i = a + \frac{b-a}{N}(i-1), \quad y_l = a + \frac{b-a}{N}(l-1), \tag{5.13}$$

for each $i, l = 1, \cdots, N$. The J simulated phase history data vectors $\boldsymbol{y}^j \in \mathbb{C}^{N\times M}$ are corrupted by both additive Gaussian noise and multiplicative gamma noise such that

$$\boldsymbol{y}^j = \boldsymbol{A}^j(\boldsymbol{\nu}^j \odot \boldsymbol{f}) + \boldsymbol{\eta}^j, \quad j = 1, ..., J, \tag{5.14}$$

where $\odot$ represents the Hadamard product. Here, $\boldsymbol{A}^j : \mathbb{C}^{N\times N} \to \mathbb{C}^{N\times M}$ is a forward operator typically chosen as a two-dimensional discrete Fourier transform operator, [49, 60, 107], in our numerical experiments,

$$\boldsymbol{\eta}^j \in \mathbb{C}^{N\times M}, \quad j = 1, ..., J, \tag{5.15}$$

model J additive Gaussian noise vectors and

$$\boldsymbol{\nu}^j \in \mathbb{R}^{N\times N}, \quad j = 1, ..., J, \tag{5.16}$$

model J multiplicative noise vectors. To simulate the speckle noise in SAR imagery, the multiplicative noise is sampled from the gamma distribution given in (5.7). That is,

$$\boldsymbol{\nu}^j \sim \Gamma(L, 1/L), \quad j = 1, ..., J,$$

where $L > 0$ represents the number of looks made by the imaging platform.

To form each single measurement vector (SMV) of $\boldsymbol{f}$ the following ℓ_1 regularization problem is solved as

$$\check{\boldsymbol{f}}^j = \underset{\boldsymbol{z}\in\mathbb{C}^{N\times N}}{\operatorname{argmin}} \left\{ ||T\boldsymbol{z}||_1 + \frac{\zeta}{2}||\boldsymbol{A}^j\boldsymbol{z} - \boldsymbol{y}^j||_2^2 \right\}, \quad j = 1, ..., J, \tag{5.17}$$

where ζ is sampled from a uniform distribution on $[0, 10]$ to simulate the ad-hoc procedure for selecting typical regularization parameters.

The transform $T : \mathbb{C}^{N\times N} \to \mathbb{R}^{N\times N}$ maps the unknown into the space of sparse edges. Because f is the *complex* scene reflectivity, with random phase values for each pixel, the operator T must effectively extract the sparsity of $|f|$, i.e. in only the magnitude, and *not* in the phase [120]. However, because $|\cdot|$ is nondifferentiable, it can not be seamlessly integrated into (5.17). To get around this, we write $f = |f|e^{i\phi}$ where $\phi = \tan^{-1}(\text{Im}\{f\}/\text{Re}\{f\})$ [108]. We then approximate the phase angle matrix Θ, with $\Theta_{j,k} \approx \text{ang}(f_{j,k})$. Hence we obtain $\Theta^* \odot f \approx |f|$.

To obtain a more accurate estimate of $|f|$, Θ is updated each time the approximation of f is updated. Specifically, we define $T : \mathbb{C}^{N\times N} \to \mathbb{R}^{N\times N}$ such that

$$Tf = \mathcal{L}^m(\Theta^* \odot f), \tag{5.18}$$

where $\mathcal{L}^m$ is the mth order PA transform (2.5) and $\odot$ is the Hadamard product. [1] A full characterization of the PA transform and its convergence properties can be found in [5, 4].

Remark 5.1.3 *When given actual SAR data, solving (5.17) may not be necessary for obtaining the SMVs. Instead, the different phase history data sets (5.14) may be processed using techniques that exploit different information channels. For example,*

[1] In this regard, the PA transform for $m > 1$ can be viewed similarly to high order total variation (HOTV) regularization [120].

different polarization, different integration angles, different elevation heights, etc. can all be considered as the measurement vectors $\check{\boldsymbol{f}}^j$. Indeed, we demonstrate in Section 5.1.4 that the VBJS technique can be used to exploit the joint sparsity of these different information channels.

Continuing as in Section 3.3, after gathering the J measurement vectors $\check{\boldsymbol{f}}^j$ in (5.16), we form the matrix $\mathcal{P}$ similar to (3.8) as

$$\mathcal{P} = \begin{bmatrix} T\check{\boldsymbol{f}}^1 & T\check{\boldsymbol{f}}^2 & \cdots & T\check{\boldsymbol{f}}^J \end{bmatrix}, \tag{5.19}$$

where as before, $\mathcal{P}$ contains the sparse edge information for all measurements. The variance (3.9) of $\mathcal{P}$ across all measurements is then computed, revealing the joint sparsity of the measurements. From the variance we compute weights as in (3.12), so that near an edge, the weights are small, approximately zero valued, and away from the edges the magnitude of the weights is approximately equal to the average jump/edge height. In this way, we are able to exploit the true sparsity of the underlying image through the weighted ℓ_p norm.

As a final step we solve the VBJS weighted ℓ_p problem given as

$$\hat{\boldsymbol{f}} = \underset{\boldsymbol{z}\in\mathbb{C}^{N\times N}}{\operatorname{argmin}} \left\{ \frac{1}{p}||T\boldsymbol{z}||_{p,\boldsymbol{w}}^p + \frac{1}{2}||\hat{\boldsymbol{A}}\boldsymbol{z} - \hat{\boldsymbol{y}}||_2^2 \right\}, \tag{5.20}$$

where $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{y}}$ are chosen according to the optimal solution vector technique outlined in Section 3.3.2 and T is defined in (5.18). It is important to note that the incorporation of the spatially varying weighting matrix $\boldsymbol{w} \in \mathbb{R}^{N\times N}$ in (5.20) allows for a non-parametric solution and thus the need to carefully hand tune parameters is eliminated. As we will show in our numerical experiments, the inclusion of the weights enforces a more accurate speckle model. In our numerical experiments we choose $p = 1$ or 2 in (5.20) and employ the weighted ADMM algorithm (Algorithm 4) for $p = 1$ and the modified gradient descent method in Algorithm 5 for $p = 2$. These algorithms were developed in Chapter 3, Section 3.4.

Speckle Metrics

One challenging task associated with analyzing speckle is the validation and quality assessment of data processed for speckle reduction. Specifically, we want to quantitatively assess the amount of speckle reduction that our methods are producing. There are two categories of metrics that can be used: (i) with-reference indices and (ii) without-reference indices [6]. With-reference indices are those that can be used when a ground truth image is available. More often than not, however, the true reflectivity of the scene is not known. This is when without-reference indices are utilized. Without-reference indices are uniquely based on specific statistical hypotheses about the signal model as well as some simple assumptions on the degree of heterogeneity of the underlying scene. In any practical application of SAR, there is no ground truth image, so here we only discuss some possible without-reference speckle metrics.

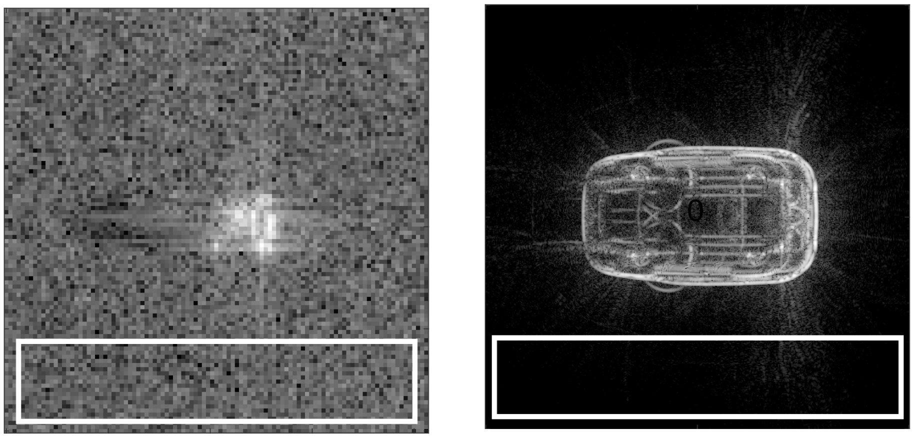

Figure 5.3: White boxes show the homogeneous regions where without-reference speckle suppression metrics are calculated for the (left) MSTAR and (right) CV Dome data sets.

We consider three different speckle metrics:

1. The *equivalent number of looks* (ENL) is a metric used for evaluating the level

of smoothing in homogeneous areas. In particular, it is desirable for the scene variation to be negligible with respect to speckle noise fluctuations in these homogeneous regions. The ENL calculates the number of multiple looks that would otherwise be required to create an image with the same level of reduced speckle. For example, an ENL of 10 implies that the speckle reduction technique is as effective as if the scene were measured 10 times, with the resulting images averaged to produce a despeckled image. Hence a good despeckling technique yields a high ENL number. We normalize all ENL calculations with respect to the single measurement vector ENL, that is, so that in the SMV case the ENL measurement is always 1. In reference to (5.5), if f is the despeckled image, then in the homogeneous areas h we have

$$\text{ENL} := \frac{E_h[f]^2}{\text{var}_h[f]}. \tag{5.21}$$

2. The *speckle standard deviation* given by

   $$\sigma_{\text{speck}} := \sqrt{\text{var}_h[f]} \tag{5.22}$$

   measures the average variation in speckle amplitude throughout homogeneous regions h. A small speckle standard deviation is indicative of desirable smoothness and speckle reduction.

3. The *reconstruction bias*, B, measures the level of bias in the estimated image. One way to calculate the bias is to measure the relative expected value of the reconstructed image to the cluttered image g, and is given by

   $$B := E\left[\frac{g-f}{f}\right]. \tag{5.23}$$

   Because we assume the fully developed speckle model (Rayleigh model), which only holds in homogeneous regions, we expect $|g-f|$ to be smallest in target areas. Hence it is desirable that reconstruction in the target area not change as a result of the speckle reduction technique, implying that (5.23) is small. Conversely, the bias will be high if these regions are greatly affected, (e.g. smoothed over). Also, the despeckled image should posses the same first order statistics as the assumed model (5.5). That is, according to (5.7), $E(\nu) = 1$, and using (5.5) we desire $E(g/f) \approx E(\nu) = 1$, or $E(g/f) - 1 \approx 0$. A bias measurement of zero represents a completely unbiased estimate, while a value of $B < .1$ indicates a low level of bias, [133].

To calculate (5.21) and (5.22), we must select a region in each image where only noise is expected to be present (no target response), [6]. For the MSTAR and CV Dome images, we chose the bottom twenty rows of the image matrix to correspond to a homogeneous region. These regions are highlighted in Figure 5.3. In the future we would like to develop more precise speckle reduction metrics that do not rely on user selection of homogeneous regions.

5.1.4 Numerical Results

To test our speckle reduction algorithms, we first consider despeckling images formed using data simulated by (5.14). The simulated images contain only homogeneous regions. Thus the fully developed speckle model developed in Section 5.1.2 will hold over the entire simulated image. We then use SAR phase history data provided by AFRL to despeckle SAR images. We compare the results of our methods to the single measurement vector (SMV), filtering, HOTV regularization and the typical AA method, without the ℓ_1 enhancement. To quantitatively compare our results, for each method we calculate the speckle reduction metrics discussed in Section 5.1.3.

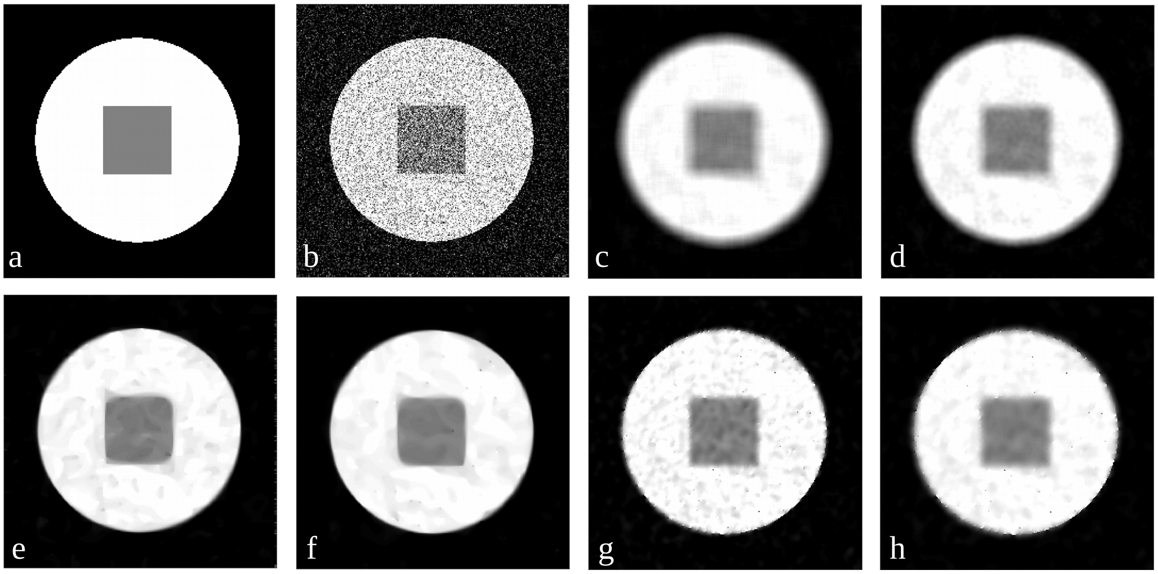

Figure 5.4: (a) True image, (b) noisy SMV image (c) filtered image (d) image reconstructed using HOTV regularization with order $m = 1$ and parameters $\mu = .06$ and $\beta = 8$, (e) AA with parameters $\eta = .2$, $\Delta t = .1$ and $N_t = 100$, (f) AA + ℓ_1 with order $m = 4$ and parameters $\eta = .2$, $\mu = 60$, $\beta = 10$, $\Delta t = .1$ and $N_t = 100$, (g) VBJS with $p = 1$, and (h) VBJS with $p = 2$.

Simulated Data

As in Chapter 3 we consider the following two examples:

Example 5.1.4 *Define* $f(x, y)$ *on* $[-1, 1]^2$ *as*

$$f(x,y) = \begin{cases} 15, & |x|, |y| \leq \frac{1}{4} \\ 20, & |x|, |y| > \frac{1}{4}, \quad \sqrt{x^2+y^2} \leq \frac{3}{4} \\ 10, & \textit{else} \end{cases}$$

Example 5.1.5 *Define* $f(x, y)$ *on* $[-1, 1]^2$ *as*

$$f(x,y) = \begin{cases} 10\left(1+\cos\left(\frac{3\pi}{2}\sqrt{x^2+y^2}\right)\right), & \sqrt{x^2+y^2} \leq \frac{1}{2} \\ 10\left(1+\cos\left(\frac{\pi}{2}\sqrt{x^2+y^2}\right)\right), & \sqrt{x^2+y^2} > \frac{1}{2} \end{cases}$$

Table 5.1: Speckle metrics for reconstructions from Example 5.1.4.

	SMV	Filter	HOTV	AA	AA + ℓ_1	VBJS ℓ_1	VBJS ℓ_2
ENL	1	81.782	121.4081	110.9548	173.0935	71.036	129.6014
σ_{speck}	1.9941	.02209	.1810	.1882	.1500	.2364	.1752
Bias	0	3.01E-5	3.52E-17	.0104	.018	1.15E-18	1.04E-18

We simulate data by sampling each function $\boldsymbol{f} : \mathbb{R}^{N\times N} \to \mathbb{R}$ on $[-1,1]^2$ on a uniform grid defined in (5.13). In (5.14), we define $\boldsymbol{A} : \mathbb{R}^{N\times N} \to \mathbb{C}^{N\times N}$ to be the normalized, two dimensional discrete Fourier transform operator so that $\boldsymbol{A}^* = \boldsymbol{A}^{-1}$ with $N = 256$. The additive noise in (5.15) is zero mean complex Gaussian noise with variance equal to 50 for Example 5.1.4 and 100 for Example 5.1.5. The multiplicative noise in (5.16) is sampled from the gamma distribution (5.6) with mean $\mu = 1$ and number of looks $L = 25$ for Example 5.1.4 and $L = 10$ for Example 5.1.5. We use (5.17) to construct each $\check{\boldsymbol{f}}^j$ with the regularization parameter ζ sampled from a uniform distribution on $[0, 10]$. We choose $J = 10$ multiple measurement vectors for each example. Because of the piecewise constant nature of Example 5.1.4 we apply the modified PA transform with order $m = 1$. Similarly, for Example 5.1.5 we use $m = 2$.

The AA method (recall Section 5.1.3) can also be used to recover the images in Examples 5.1.4 and 5.1.5. In this case we choose $\eta = .2$ in (5.9) and set $N_t = 100$ time steps of size $\Delta t = .1$. To implement (5.12), Algorithm 6 is employed using the 4th order PA transform with regularization parameters $\mu = 60$ and $\beta = 10$.

Figures 5.4 and 5.5 display the results of despeckling noisy images from Examples 5.1.4 and 5.1.5, respectively. In each case, we compare our results to the outcome of

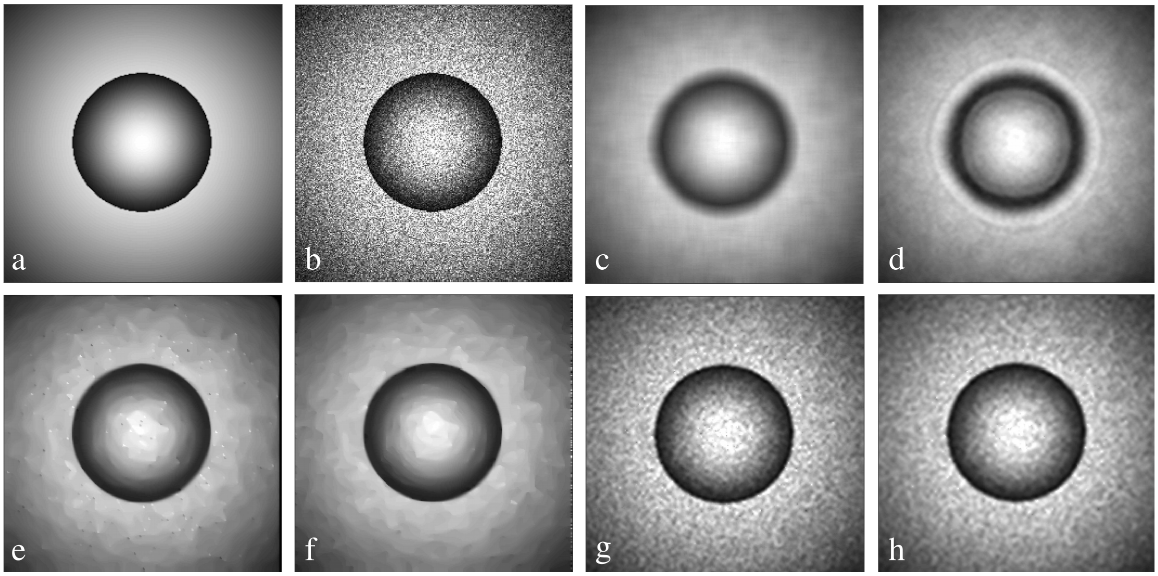

Figure 5.5: (a) True image, (b) noisy SMV image (c) filtered image (d) image reconstructed using HOTV regularization with order $m = 2$ and parameters $\mu = .005$ and $\beta = 2$, (e) AA with parameters $\eta = .2$, $\Delta t = .1$ and $N_t = 100$, (f) AA + ℓ_1 with order $m = 4$ and parameters $\eta = .2$, $\mu = 60$, $\beta = 10$, $\Delta t = .1$ and $N_t = 100$, (g) VBJS with $p = 1$, and (h) VBJS with $p = 2$.

filtering the noisy image with a Lee filter [81, 82, 83]. We also compare to the result of reconstructing the scene using the HOTV method, which requires careful tuning of parameters. The HOTV method and ADMM algorithm (Algorithm 2) were discussed in Chapter 2 and adapted for SAR in [120]. In Example 5.1.4, when using the HOTV technique, we employ the PA transform of order $m = 1$ with parameters $\mu = .06$ and $\beta = 8$, and in Example 5.1.5 we choose PA order $m = 2$ with parameters $\mu = .005$ and $\beta = 2$.

It is apparent in Figures 5.4 and 5.5 that the AA methods and VBJS techniques successfully reduce image variability due to speckle. However, it appears that the ℓ_1 enhanced AA method does not provide significant improvement over the AA method in these examples. Cross sections of the results are displayed in Figure 5.6 for the

Table 5.2: Speckle metrics for reconstructions from Example 5.1.5.

	SMV	Filter	HOTV	AA	AA + ℓ_1	VBJS ℓ_1	VBJS ℓ_2
ENL	1	6.7522	6.3041	6.6233	6.3749	4.6975	5.1115
σ_{speck}	4.1637	1.6014	1.6543	1.6057	1.6089	1.9024	1.8409
Bias	0	5.69E-4	-1.19E-17	.0239	.0328	1.46E-18	-4.33E-19

AA, ℓ_1 enhanced AA, and VBJS methods. These plots show the improvement each technique is making compared to the SMV case along with the accuracy of each method.

Tables 5.1 and 5.2 confirm these results by providing the calculated speckle metrics for each reconstruction. We see through speckle standard deviation σ_{speck} that both the VBJS ℓ_1 and ℓ_2 methods significantly reduce speckle while maintaining a low level of bias and high ENL, while eliminating the need for hand tuning parameters, as in the HOTV method. The AA methods reduce speckle and increase ENL numbers, but introduce more bias into the final image. As discussed previously, this additional bias indicates possible loss of information.

As in Chapter 3, we then calculate the relative ℓ_2 error of each reconstruction $\hat{\boldsymbol{f}}$ of $\boldsymbol{f}$ as

$$\frac{||\hat{\boldsymbol{f}} - \boldsymbol{f}||_2}{||\boldsymbol{f}||_2}. \tag{5.24}$$

Figure 5.7 displays the result of calculating this error for each technique with parameters described above and resolutions $N = 32, 62, 128, 256$ and 512. We see that the VBJS technique with $p = 1$ and the AA method both consistently maintain low relative errors. The proposed VBJS method with $p = 2$ and ℓ_1 enhanced AA methods also produce convergent results.

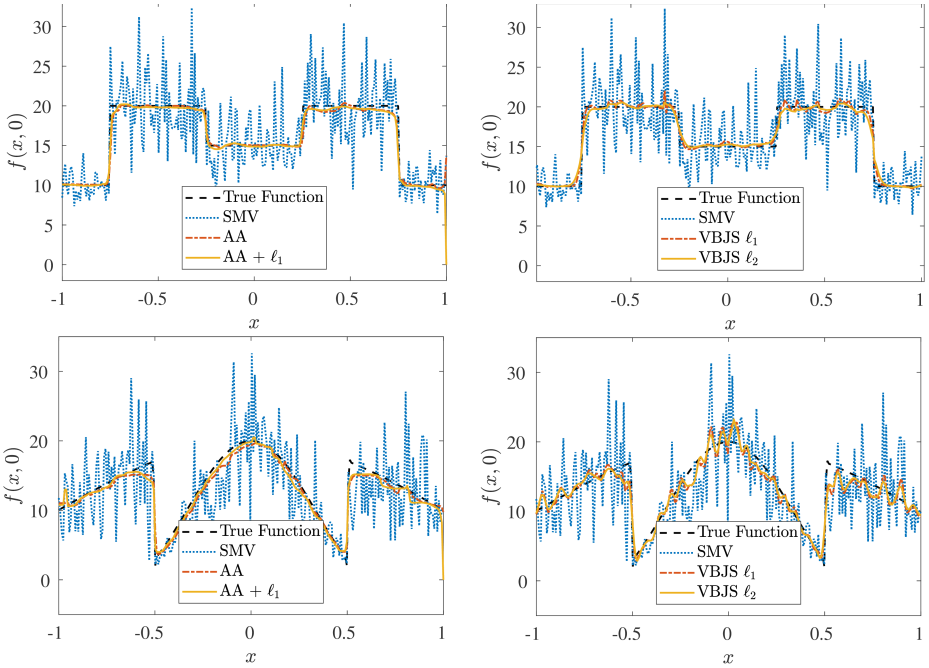

Figure 5.6: (top) Cross sections of despeckling Example 5.1.4. (bottom) Cross sections of despeckling Example 5.1.5.

SAR Phase History Data

Our results so far demonstrate that the speckle reduction techniques work well for simulated examples. We are now ready to test our methods on actual SAR phase history data. Specifically we consider the MSTAR [122] and Civilian Vehicle (CV Dome) [116] data sets provided by AFRL. Each data set that we utilize has been explained in detail in Section A of Appendix A. We also include examples of typical MSTAR and CV Dome imagery in Appendix A, Figures A.13 and A.15. We choose to explore these particular data sets to test our algorithms on both measured and computer generated SAR phase history data. The MSTAR data are measured data, as it was acquired by flying over the scene with the radar, and the CV Dome data set was generated using Xpatch [65], an electromagnetic simulation software.

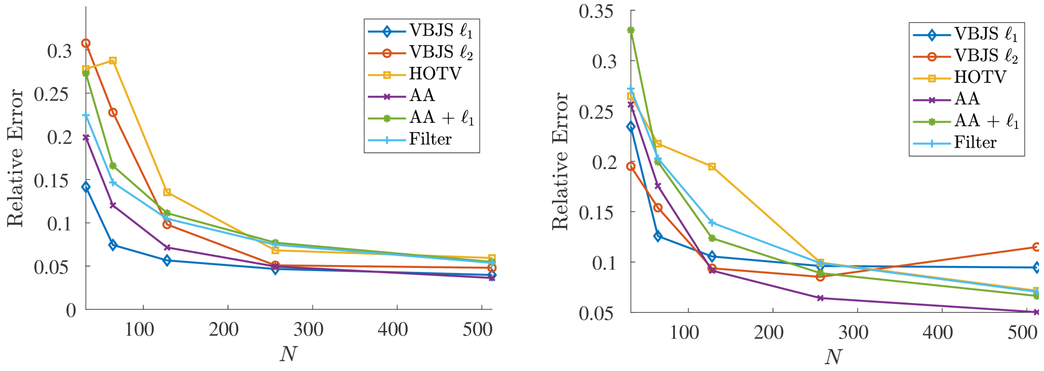

Figure 5.7: Relative error (5.24) of speckle reduction techniques for despeckling (left) Example 5.1.4 and (right) Example 5.1.5 at resolutions $N = 32, 62, 128, 256$ and 512.

Table 5.3: Speckle metrics for reconstructions from MSTAR data [122].

	SMV	Filter	HOTV	AA	AA + ℓ_1	VBJS ℓ_1
ENL	1	58.0563	93.3568	7.2904	27.3967	60.8743
σ_{speck}	5.6843	.7221	.5556	2.0971	1.0872	.6980
Bias	0	.0337	.0566	.0048	.0251	.0455

MSTAR Phase History Data

In our first experiment we use the MSTAR data set [122]. We choose to reconstruct data corresponding to a scene that contains a T-72 (SNS7) tank where the imaging platform is located 4551m above the scene center at a 15° elevation angle. For this data collect, the center frequency is set at $\omega_c = 9.6$GHz with wavelength $\lambda = c/\omega_c = .0312$m, a bandwidth of $B = 591$MHz and an integration angle of $\theta_a = 2.9361°$. The range and cross range resolutions are thus

$$\rho_x = \frac{c}{2B} = .2536\text{m} \quad \text{and} \quad \rho_y = \frac{\lambda}{2\theta_a} = .3047\text{m}.$$

We first despeckle the MSTAR imagery using the HOTV regularization technique

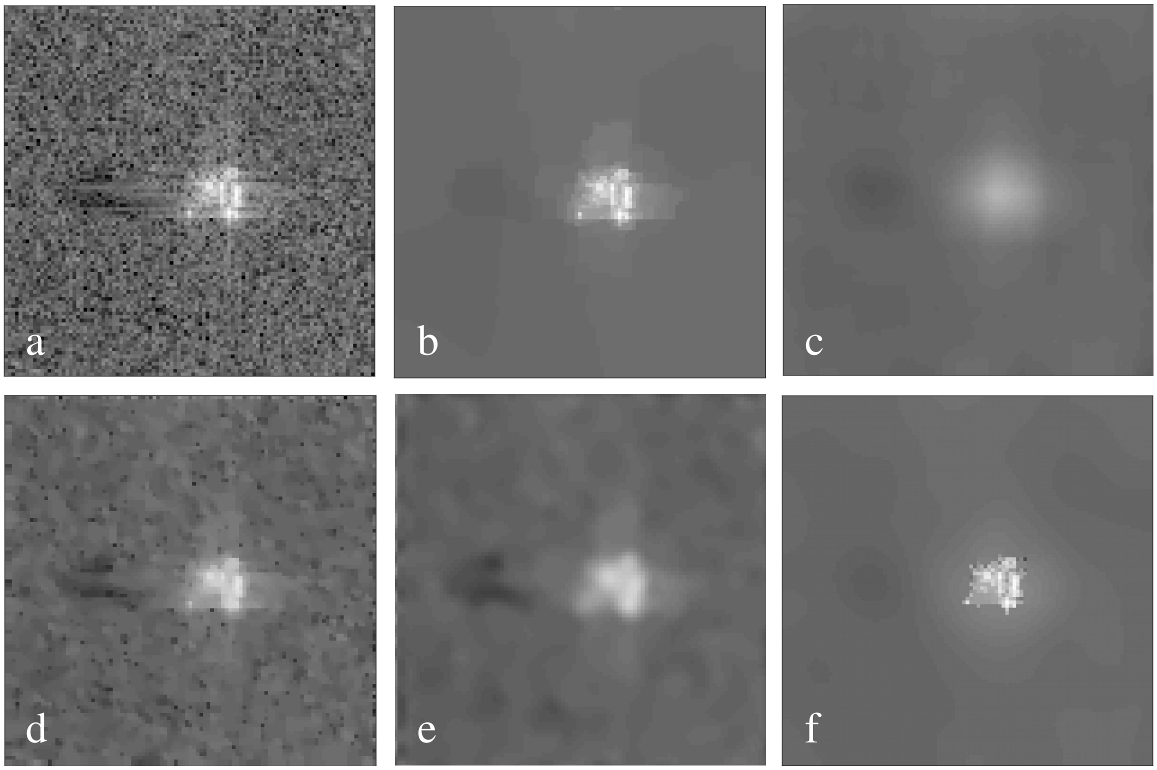

Figure 5.8: (a) Noisy SMV image (b) image reconstructed using HOTV regularization with order $m = 2$ and parameters $\mu = .25$ and $\beta = 1.5$, (c) filtered image, (d) AA with parameters $\eta = 1.5$, $\Delta t = .1$ and $N_t = 100$, (e) AA + ℓ_1 with order $m = 4$ and parameters $\eta = 1.5$, $\mu = 10$, $\beta = 5$, $\Delta t = .1$ and $N_t = 100$ and (f) VBJS with $p = 1$.

discussed in Chapter 2, for which we use PA order $m = 2$ and parameters $\mu = .25$ and $\beta = 1.5$ in Algorithm 2. The forward model is defined as the non-uniform fast Fourier transform (NUFFT) [49, 60, 107]. We also despeckle the imagery using the Lee filter [81, 82, 83], the AA method and the ℓ_1 enhanced AA method discussed in Section 5.1.3, all of which are post-processing techniques. For the AA methods we choose $\eta = 1.5$, $\Delta t = .1$, $N_t = 100$, $m = 4$, $\mu = 10$ and $\beta = 5$. As a final despeckling technique, we consider the VBJS method in Algorithm 4 where we generate measurements $\check{\boldsymbol{f}}^j$,

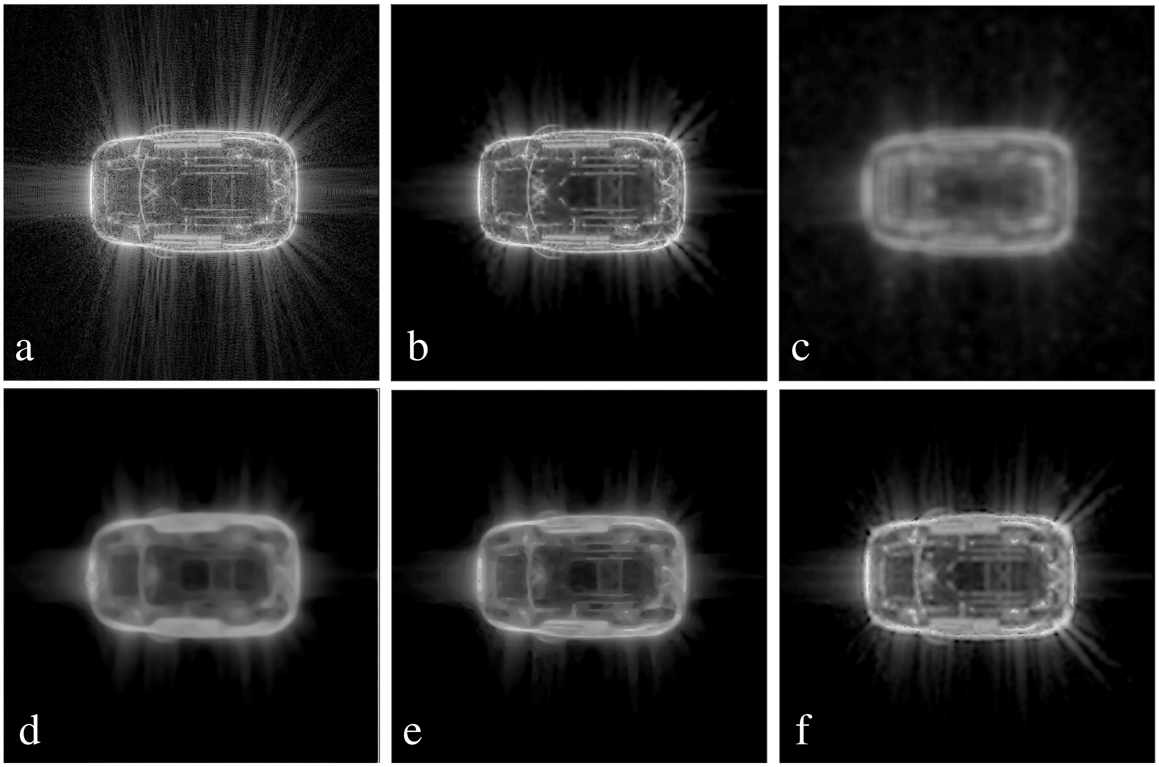

Figure 5.9: (a) Noisy SMV image (b) image reconstructed using HOTV regularization with order $m = 2$ and parameters $\mu = 60$ and $\beta = 120$, (c) filtered image, (d) AA with parameters $\eta = 2$, $\Delta t = .1$ and $N_t = 100$, (e) AA + ℓ_1 with order $m = 4$ and parameters $\eta = 2$, $\mu = 10$, $\beta = 5$, $\Delta t = .1$ and $N_t = 100$, and (f) VBJS with $p = 1$.

$j = 1, ..., 10$, by varying the regularization parameter ζ in (5.17) so that

$$\check{\boldsymbol{f}}^j = \underset{\boldsymbol{z} \in \mathbb{C}^{103 \times 103}}{\operatorname{argmin}} \left\{ ||T\boldsymbol{z}||_1 + \frac{\zeta_j}{2} ||\boldsymbol{A}\boldsymbol{z} - \boldsymbol{y}||_2^2 \right\}, \quad j = 1, ..., 10.$$

Here T is given by (5.18) with $m = 2$, $\boldsymbol{A}$ is the NUFFT, and $\boldsymbol{y}$ is the given phase history data corresponding to the MSTAR scene. We then solve (5.20) for $\hat{\boldsymbol{f}}$ with $p = 1$ and T again as the modified second order PA transform (5.18).

Figure 5.8 displays the despeckled results when using the MSTAR data, and Table 5.3 gives the associated speckle reduction metrics. Observe that the results are consistent with the previously shown simulated data case (see Figures 5.4 and 5.5 and

Tables 5.1 and 5.2). Although HOTV yields the best results for the chosen speckle metrics, careful parameter tuning is required. The VBJS technique performs comparably *without* the need for parameter tuning. Finally the ℓ_1 enhancement of the AA method now yields improvement over the standard AA method and enhances shadow information, which is important for applications such as target recognition.

Table 5.4: Speckle metrics for reconstructions from CV Dome data [116].

	SMV	Filter	HOTV	AA	AA + ℓ_1	VBJS ℓ_1
ENL	1	13.5821	2.1001	2.4384	2.6037	3.36
σ_{speck}	6.828	1.6159	4.5797	4.3180	4.2112	3.4823
Bias	0	.1517	.0334	.0032	.0061	.0787

CV Dome Phase History Data

With the CV Dome data set we choose to reconstruct the scene consisting of a 93' Jeep. The data are collected at a 30° elevation angle with a 360° azimuth sweep and center frequency of 9.6GHz. This leads to a phase history data set consisting of 512 frequencies and 5760 azimuth angles (pulses). We first despeckle the CV Dome imagery using the HOTV regularization technique discussed in Chapter 2, for which we use PA order $m = 2$ and parameters $\mu = 60$ and $\beta = 120$ in Algorithm 2. The forward model is defined again as the NUFFT. We also despeckle the imagery using the Lee filter [81, 82, 83], the AA method and the ℓ_1 enhanced AA method discussed in Section 5.1.3. For these methods we choose $\eta = 2$, $\Delta t = .1$, $N_t = 100$, $m = 4$, $\mu = 10$ and $\beta = 5$.

As a final despeckling technique, we consider the VBJS method given in (5.20) with

$p = 1$ where we generate measurements by varying the polarization of the transmitted and received signals. We consider a horizontal transmit with a horizontal receive (HH) and a vertical transmit with a vertical receive (VV). The measurements $\check{\boldsymbol{f}}^j$, $j = 1, 2$ are acquired by solving (5.17) with $\boldsymbol{y}^1$ as the HH phase history data and $\boldsymbol{y}^2$ as the VV phase history data. We find it necessary to solve (5.17) to obtain the SMVs because speckle must be reduced before enhancing the edges and finding the optimal sparsity vector (5.19). We choose $\boldsymbol{A}^1 = \boldsymbol{A}^2 = \boldsymbol{A}$ to be the NUFFT and T in (5.18) with $m = 2$. For the final VBJS reconstruction (5.20), given the measurements $\check{\boldsymbol{f}}^j$, we also choose T as in (5.18) with $m = 2$.

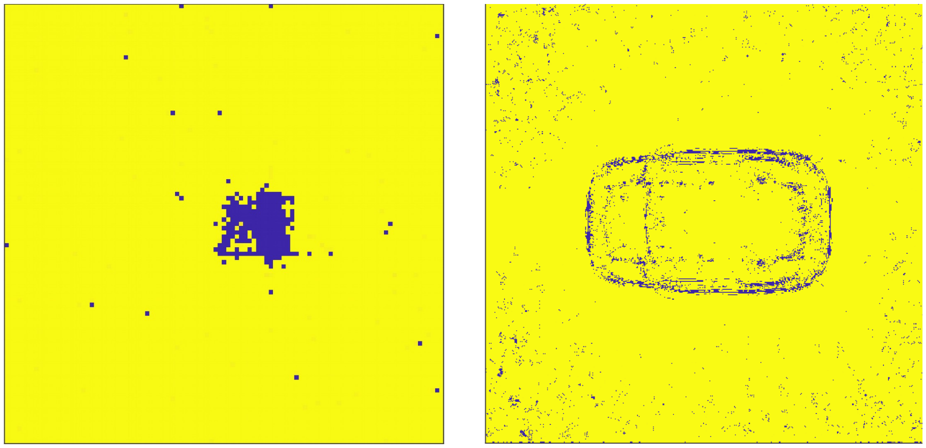

Figure 5.10: Weights calculated according to (3.12) for the VBJS technique (5.20) when despeckling (left) the MSTAR data and (right) the CV Dome data. Dark regions indicate values near zero.

Displayed in Figure 5.9 are the resulting images from the aforementioned despeckling techniques, where the VV phase history data are used in the HOTV reconstruction. All images are displayed at the same dynamic range. Calculated in Table 5.4 are the corresponding speckle metrics for each image. In this case, we see through the ENL and speckle standard deviation that filtering removes the most speckle, but introduces the most bias into the reconstruction. On the other hand, the ℓ_1 enhanced

AA method and the VBJS technique both significantly reduce speckle and remain unbiased. Furthermore, both techniques outperform the HOTV method, however the main advantage of using the VBJS method is that it does not require parameter tuning and is thus robust to different data sets.

Figure 5.10 shows the weights that were calculated by the VBJS algorithm. The dark regions indicate values near zero and the light regions indicate values significantly greater than zero. What is most evident here is that the weights are approximately zero wherever there is a strong response from the radar. Thus, by weighting the reconstruction in this way (5.20), we only regularize at locations away from the areas of interest, preserving information about targets and introducing minimal bias into the solution. This is also important because the multiplicative model for speckle does not hold in the presence of strong scatterers, and thus by only applying the model to the homogeneous regions, we obtain a more accurate model of the speckle phenomena.

5.1.5 Discussion and Conclusions

In this section we thoroughly analyzed speckle noise in SAR imagery. This led to proposing two novel techniques for speckle reduction. Our first technique, the ℓ_1 enhanced AA method, was based on ideas developed in Chapter 4 of this thesis, where we augment known PDE solvers with ℓ_1 regularization. This method incorporates the statistics of speckle intensity and enforces sharp transitions and edges within imagery. The second technique is based on methods proposed in Chapter 3 of this thesis, where we developed a variance based joint sparsity algorithm for recovering images from multiple measurements having the same sparsity profile. This method is non-parametric and exploits the extra information gained when provided with multiple measurements of the same underlying scene.

Our numerical results show that the ℓ_1 enhanced AA method provides small improvements in speckle reduction over the AA method while maintaining a low level of image bias – thus reducing speckle while preserving information, which is important when using images for object classification and recognition. However, there are many parameters to tune within the AA algorithms and they are computationally expensive as the PDE that must be solved has to be driven to steady state. Moreover, as of now, the method can only be applied as a post-processing technique, and important information is being lost in the PDE simulation. Overcoming these bottlenecks is a topic of future research.

The VBJS method has shown to be a numerically efficient, robust way to reduce speckle in SAR images. Numerical results show that the results for the VBJS method are comparable to those of HOTV regularization, but without the need to hand tune parameters. Hence the method is nearly autonomous and thus can be used in SAR automated target recognition (ATR) algorithms. Implementation of the VBJS in SAR ATR is another topic of future research, as well as implementing the VBJS method for noise reduction of imagery from other sensors.

5.2 Autofocusing

5.2.1 Introduction

Imaging via synthetic aperture radar (SAR) is a well-established technique for effective scene reconstruction under most conditions, with resolution up to a few centimeters [106, 32, 69]. SAR imaging works by acquiring data from a number of different viewpoints, or azimuth angles, and moving the transmission mechanism around a flight path about the scene. At each azimuth angle, an electromagnetic (EM) wave with microwave length frequencies is transmitted towards the scene and scattered from obstructions, or "scatterers," within the scene. The measured scattered echo response from the scene serves as the data from which to reconstruct the image.

This measurement process requires the round trip time for the EM wave to travel to the scene and return back to the sensing mechanism. Under ideal conditions, the distance from the scene center to the transmission and sensing mechanisms is accurately known. Moreover, in a vacuum it is known that the EM wave travels at precisely the speed of light. Hypothetically, the round trip time can be exactly determined. In practice however, this distance can only be approximated, and atmospheric disturbances can delay the wave propagation. Additionally, because EM waves propagate at a very high speed, minute errors in the measurement of the distance to the scene can result in significant relative errors in the round trip time estimates. These errors manifest as phase errors on the data which produce imagery that is then characterized as *defocused*. A defocused image often appears smeared or blurred, making information extraction difficult. Thus a number of *autofocusing* algorithms have been designed to alleviate this issue [110, 76, 149, 26].

The Phase Gradient Autofocus (PGA) algorithm [149, 45] is often used to correct for these errors in imagery formed from data collected by monostatic radar systems. PGA is an iterative post processing algorithm that makes a robust estimation of the gradient of the phase error with respect to azimuth angle. The estimation process exploits the redundancy of the phase error information contained in the defocused image. The PGA algorithm requires circular shifting of data to remove frequency offset due to Doppler shifting of the scatterer, windowing of the circularly shifted imagery to preserve the width of the dominant scatterers in the scene, a phase gradient estimation, and an iterative correction process to remove any estimation bias.

Within the PGA algorithm, Fourier transforms between the image and the range-compressed domains are required for each iteration. To take advantage of the redundancy in the data, this back and forth processing must be performed over many range bins, making the algorithm computationally intensive. Finally, we note that since the PGA is a post processing algorithm, it can not account for information lost in the reconstruction process.

Because of the inherent limitations of the PGA algorithm, in this investigation we propose a regularization-based autofocusing procedure that jointly estimates the underlying image and its corresponding phase errors. While similar techniques were proposed in [74, 110, 146, 161], our algorithm also exploits the redundancies and correlations present in the unknown phase errors. Furthermore, we more carefully exploit the nature of the unknown phase errors, as they vary linearly with respect to the frequencies, making them inherently two-dimensional.

Autofocusing algorithms typically correct for a one-dimensional phase error in the range compressed domain which ignores the dependency on the spatial frequencies of the transmitted EM wave. For example, a weighted least squares estimation of these one-dimensional phase errors was formulated in [161]. The algorithm makes no assumptions on the noise model or underlying scene, with the weights calculated to be inversely proportional to the variance of the phase in each range bin. The method was shown to minimize the variance of the residual phase error, and although shown to be robust with respect to various noise models, it is evident that estimation can be improved by the incorporation of prior knowledge about the scene and phase errors [63, 121, 146, 26]. We also note that while it was recognized in [98, 76, 152, 110] that the source of defocusing is a two-dimensional phase error corrupting the raw phase history data, the algorithms proposed in these investigations all perform the correction in the range compressed domain.

The effects of one-dimensional phase errors on an under-sampled SAR system were investigated in [74], where the SAR data were considered to be under-sampled in the cross-range dimension, and phase errors assumed to be constant. It was also assumed that there were a sparse number of scatterers in the underlying scene so that it was possible to jointly optimize for the phase errors and the imagery. This work was further advanced in [146] where a total variation constraint was incorporated on the reconstructed scene.

An autofocusing technique that performs one-dimensional phase error correction through the optimization of an image-domain sharpness function [2] was developed in [128] and further enhanced in [7]. In [128] a SAR sharpness function was derived as the solution to maximum-likelihood and maximum-posterior estimation for idealized SAR data. This theory was then applied to develop an autofocusing method for images formed using backprojection [7]. Such geometric interpretation of the problem allows for optimal, single-pulse phase corrections to be derived in closed form as the solution of a quartic polynomial. This was shown to reduce the computational cost of autofocusing back-projected images, but to date the work has not been extended to more numerically efficient image formation algorithms, such as those based on interpolation techniques (e.g., the non-uniform fast Fourier transform (NUFFT) or the Polar Format Algorithm.)

In [50, 51], a shear averaging technique was developed for SAR autofocus. [3] The shear averaging algorithm consists of computing the average over the sheared product of phase corrupted data from adjacent azimuth angles. The phase of the computed average is equivalent to the difference of the unknown phase from one azimuth angle to the next. Thus the unknown phase can be written as a recurrence relationship and explicitly calculated as the sum of the phases of the computed average. This preprocessing technique is only suitable for phase errors that do not depend on frequency values, i.e. that are one-dimensional. Adapting the shear averaging technique to two-

[2] In [128, 7], the sharpness function is chosen as $||\nu||_2^2$, where ν is the image intesity. In general, sharpness is a convex function of intensity.

[3] Shear averaging has some similarities to wavefront sensing by shearing interferometry, [50, 129].

dimensional phase error correction is beyond the scope of this paper.

Finally, a different approach was suggested in [110] and [63]. The autofocusing methods in those investigations also assume that the phase errors only depend on the azimuth angle, and as before the underlying scene is assumed to be dominated by a small number of strong scatterers. The phase errors and imagery are found through an alternating optimization procedure, with the scene assumption incorporated into the cost function by sparsity promoting ℓ_1 and ℓ_p norms with $p < 1$. Though the results are promising, these papers do not accurately model the phase errors.

In this section, we develop an autofocusing techniquee that exploits the correlation of the phase error on both the azumth angle and spatial frequencies while also enforcing the piecewise smooth nature of the image within the scene. Initially constructed in [121], our method estimates the phase error correction and the image through a joint optimization procedure. New to this investigation is the incorporation of a *phase synchronization technique* for the estimation of the unknown, two-dimensional phase error. This is needed because the optimization prceedure proposed in [121] was not robust to large spatial frequency values. Also, smoothness was enforced on the phase errors. This is an inaccurate assumption, however, as the phase should only be recovered as wrapped, piecewise smooth functions. As in [121], here we use high order regularization methods, yielding the additional advantage of reducing speckle in SAR images.

5.2.2 Mathematical Modeling of Phase Errors

The explanation below generally follows the work of Jakowatz et al. [69] (see Chapter 4). A comprehensive discussion of the SAR data acquisition process is given in Appendix A. Let $f : \Omega \rightarrow \mathbb{C}$ denote be the two-dimensional reflective scene of scattering objects that we want to recover, and suppose f is defined over

$$\Omega := \{(x,y) \in \mathbb{R}^2 | x^2 + y^2 \leq L^2\}.$$

At a particular position in the sensing process, indicated by an azimuth angle $\theta \in \mathbb{R}$, the transmitted linear, frequency-modulated (FM) chirp mixes with the scene in a way that depends upon θ, the angle from which the chirp is emitted. [4] In the far field case, once the transmitted signal reaches the scene it has essentially a planar wave front, and thus the points in the scene along each line perpendicular to the direction of the chirp all mix with the same values. Hence the two-dimensional setup is often simplified to a one-dimensional process by compressing the scatterers along each of these lines to a single point. This compression is commonly referred to as the projection or Radon transform of f at the angle θ, and is denoted $p : \Omega \to \mathbb{R}^2$. It can be expressed mathematically as

$$p(\theta, u) = \iint\limits_{x^2+y^2 \leq L^2} f(x,y)\delta(u - x\cos\theta - y\sin\theta)\, dx\, dy. \tag{5.25}$$

The linear FM chirp that is transmitted and mixed with the scene is described as the real part of

$$s(t) = \begin{cases} e^{i(\omega t + \alpha t^2)}, & |t| \leq \frac{T}{2} \\ 0, & \text{otherwise} \end{cases}, \tag{5.26}$$

where ω is the carrier frequency, 2α is the chirp rate, and $T > 0$ is the pulse duration. This chirp signal mixes with the scene to yield reflected signals of the form

$$r(\theta, t) = \int_{-L}^{L} \mathrm{Re}\left\{p(\theta,u)s(t - \tau_0 - \tau(u))\right\} du, \tag{5.27}$$

where $\tau_0 + \tau(u)$ is the *estimated* round trip time for the chirp to travel to scene position u. Specifically, τ_0 is the round trip time required for the chirp to travel to

[4] In practice there is also a relevant angle of elevation which is not critical to the development of our method.

the scene center and $\tau(u)$ is the additional travel time for any particular position in the scene u. If R is the distance from the transmitter/receiver to the scene center and c is the speed of light in a vacuum, we have

$$\tau_0 = 2R/c, \qquad \tau(u) = 2u/c.$$

A *deramping* process is implemented to extract approximate instantaneous frequency information (i.e. the classical Fourier transform of f) from the chirp response. In brief, this process requires the following steps:

1. Demodulation of $r(\theta, t)$ for each θ by multiplication with in-phase and quadrature signals
$$\begin{aligned} r_I(t) &= \cos\left[\omega(t-\tau_0) + \alpha(t-\tau_0)^2\right] \\ r_Q(t) &= \sin\left[\omega(t-\tau_0) + \alpha(t-\tau_0)^2\right] \end{aligned} \tag{5.28}$$
to obtain $r_d(\theta, t) = r(\theta, t)\,(r_I(t) + ir_Q(t))$. Using the appropriate trigonometric identities, one can show that
$$\begin{aligned} r_d(\theta, t) = \frac{1}{2}\int_{-L}^{L} & p(\theta, u)\mathrm{exp}\{i(\omega(2t - \tau(u) - 2\tau_0) \\ & + \alpha((t-\tau_0)^2 + (t - \tau(u) - \tau_0)^2))\}\, du \\ & + \frac{1}{2}\int_{-L}^{L} p(\theta, u)\mathrm{exp}\{i[\alpha\tau^2(u) - \tau(u)(\omega + 2\alpha(t-\tau_0))]\}\, du. \end{aligned} \tag{5.29}$$

2. Low pass filtering of (5.29) to remove the first term, yielding
$$r_d(\theta, t) \approx \frac{1}{2}\int_{-L}^{L} p(\theta, u)\exp\left\{-i\frac{2u}{c}\left[\omega + 2\alpha(t-\tau_0)\right]\right\}\exp\left\{i\alpha\frac{4u^2}{c^2}\right\} du. \tag{5.30}$$

Assuming that the chirp rate α and the scene radius L are sufficiently small, we have

$$\alpha\tau^2(u) = \frac{4\alpha u^2}{c^2} \approx 0,$$

so that $e^{i\alpha\tau^2(u)} \approx 1$. The resulting approximation of the *ideal* data is then given by

$$\hat{f}_\theta(t) := \int_{-L}^{L} p(\theta, u)e^{-iku}\, du \approx r_d(\theta, t), \tag{5.31}$$

where

$$k = k(t) := \frac{2}{c}(\omega + 2\alpha(t-\tau_0)). \tag{5.32}$$

In other words, the demodulation approximately yields the Fourier coefficients of the projection of f, and therefore by the projection slice theorem [32, 69], we have

$$\hat{f}_\theta(t) = \int_{-L}^{L} p(\theta,u)e^{-iku}\,du = \iint\limits_{x^2+y^2\le L^2} f(x,y)e^{-ik(x\cos\theta+y\sin\theta)}dxdy. \tag{5.33}$$

An error in the estimated round trip propagation time corresponds to an unknown shift in τ_0, which varies with respect to azimuth angle. To understand the effects of this error, we replace τ_0 by $\tau_0+\epsilon(\theta)$ in (5.28). The demodulation procedure now requires multiplication with shifted in-phase and quadrature signals,

$$\begin{aligned} r_I^\epsilon(t) &= \cos\left[\omega(t-\tau_0+\epsilon(\theta)) + \alpha(t-\tau_0+\epsilon(\theta))^2\right] \\ r_Q^\epsilon(t) &= \sin\left[\omega(t-\tau_0+\epsilon(\theta)) + \alpha(t-\tau_0+\epsilon(\theta))^2\right], \end{aligned} \tag{5.34}$$

resulting in the corresponding demodulation

$$r_d^\epsilon(\theta,t) = r(\theta,t)(r_I^\epsilon(t) + ir_Q^\epsilon(t)).$$

Once again using appropriate trigonometric identities, one can show that

$$\begin{aligned} r_d^\epsilon(\theta,t) = \frac{1}{2}\int_{-L}^{L} & p(\theta,u)\exp\{i(\omega(2t-2\tau_0-\tau(u)+\epsilon(\theta)) \\ & + \alpha((t-\tau_0)^2 + (t-\tau_0-\tau(u))^2 + 2\epsilon(\theta)(t-\tau_0)\}du \\ & + \frac{1}{2}\int_{-L}^{L} p(\theta,u)\exp\{-i\tau(u)(\omega+2\alpha(t-\tau_0))\}\exp\{-i\epsilon(\theta) \\ & (\omega+2\alpha(t-\tau_0))\}\exp\{i\alpha(\tau(u)^2+\epsilon(\theta)^2)\}du. \end{aligned} \tag{5.35}$$

As before, low pass filtering is used to remove the first term of (5.35) yielding

$$\begin{aligned} r_d^\epsilon(\theta,t) \approx \frac{1}{2}\int_{-L}^{L} & p(\theta,u)\exp\left\{-i\left(\tau(u)+\epsilon(\theta)\right)\left[\omega+2\alpha(t-\tau_0)\right]\right\} \\ & \exp\left\{i\alpha\left(\tau(u)^2+\epsilon(\theta)^2\right)\right\}du. \end{aligned} \tag{5.36}$$

The chirp rate α and the scene radius L are again assumed to be sufficiently small, so that $\alpha\tau^2(u) = \frac{4\alpha u^2}{c^2} \approx 0$. It is further assumed that $\epsilon(\theta)$ is small, implying

$$\alpha\epsilon(\theta)^2 \approx 0$$

and therefore

$$e^{i\alpha(\tau^2(u)+\epsilon(\theta)^2)} \approx 1.$$

This leads to the model for the *phase corrupted* data:

$$\hat{f}^{\epsilon}_{\theta}(t) := \int_{-L}^{L} p(\theta, u) e^{-iku} e^{-i\frac{\epsilon(\theta)c}{2}k} du = e^{-ik\phi(\theta)} \hat{f}_{\theta}(t) \approx r^{\epsilon}_{d}(\theta, t), \tag{5.37}$$

where the spatial frequencies k are defined in (5.32) and $\phi(\theta) = \frac{\epsilon(\theta)c}{2}$. Hence we have the following problem: Given phase corrupted data $\hat{f}^{\epsilon}$ in (5.37), how do we extract the appropriate phase correction, without explicit knowledge of $\hat{f}$, and also simultaneously estimate f? Our approach to this problem is discussed below.

Model Discretization

To discretize the problem, we let the *temporal* frequency values be given by t_j for $j = 1, ..., K$, and the azimuth angles by θ_n for $n = 1, ..., N_p$. We also denote

$$k_j = \frac{2}{c}(\omega + 2\alpha(t_j - \tau_0)), \quad j = 1, ..., K, \tag{5.38}$$

as the discretized *spatial* frequencies and $\mathcal{F} : \mathbb{C}^{N\times N} \to \mathbb{C}^{K\times N_p}$ as the discrete forward operator modeled by (5.33) that maps the reflectivity $f \in \mathbb{C}^{N\times N}$ to the data $\hat{f} \in \mathbb{C}^{K\times N_p}$. Finally, we define

$$b = \{\hat{f}_{\theta_n}(k_j)\}_{j=1,n=1}^{K,N_p} \quad \text{and} \quad b^{\epsilon} = \{\hat{f}^{\epsilon}_{\theta_n}(k_j)\}_{j=1,n=1}^{K,N_p} \tag{5.39}$$

as the vectors containing the ideal data acquired in (5.33) and the data containing phase errors in (5.37), respectively.

The classical problem is to find f satisfying $\mathcal{F}f = b$. Because of the imperfect knowledge of the round trip wave propagation time, we instead seek to solve

$$\mathcal{F}^{\epsilon} f = b^{\epsilon}, \tag{5.40}$$

where $\mathcal{F}^{\epsilon}$ is the discrete forward transform modeled by (5.37). However, due to the phase error, $\mathcal{F}^{\epsilon}$ is not known must be estimated through $\mathcal{F}$ by incorporating a phase correction into the model. The process is described below.

5.2.3 Range Compressed Data

For simplicity, define $\phi(\theta_n) = \frac{\epsilon(\theta_n)c}{2} = \phi_n$ where as before, ϵ denotes the phase error defined in (5.37). We denote $\phi = \{\phi_n\}_{n=1}^{N_p}$ as the vector of phase errors, for which we build a *correction* to the forward model as

$$E = E(\phi) = \text{diag}\{e^{-ik_j\phi_n}\}_{j,n=1}^{K,N_p}. \tag{5.41}$$

Clearly, if ϕ is known then from (5.37) we can explicitly determine the forward operator in (5.40) as $\mathcal{F}^{\epsilon} = E\mathcal{F}$. However, in practice ϕ is not known, so we must design our algorithm to attempt to recover it along with the image.

In most autofocusing algorithms, [7, 26, 51, 63, 146, 149], the phase correction is done in the range compressed domain. The range compressed data are the discrete, one-dimensional (inverse) Fourier transforms of the phase history data along the range dimension. The range compressed data corresponding to the idealized phase history data (5.33) are given by

$$c_r(\theta_n, m) = \sum_{j=1}^{K} \hat{f}_{\theta_n}(k_j)e^{ik_j m}, \quad m = 1, ...K, \tag{5.42}$$

and all autofocusing algorithms to our knowledge correct for a phase error defined as $e^{-i\phi_n}c_r(\theta_n, m)$, where the phase error, $-\phi_n$, is assumed to be one-dimensional and depend only on the azimuth angle. However, from the above derivation concluding at (5.37), it is evident that the range compressed data corresponding to the phase corrupted phase history data are actually given by

$$c_r^{\phi}(\theta_n, m) = \sum_{j=1}^{K} \hat{f}^{\epsilon}_{\theta_n}(k_j) e^{ik_j m} = \sum_{j=1}^{K} \hat{f}_{\theta_n}(k_j) e^{ik_j(m-\phi_n)} = c_r(\theta_n, m - \phi_n), \qquad (5.43)$$

for $m = 1, ..., K$. Moreover, due to the dependency of the phase error on the spatial frequencies,

$$c_r(\theta_n, m - \phi_n) \neq e^{-i\phi_n} c_r(\theta_n, m), \qquad (5.44)$$

and hence the phase correction should be done on the raw phase history data, rather than the range compressed data.

5.2.4 Proposed Method for Joint Image Formation and Phase Error Correction (Autofocusing)

Our proposed autofocusing algorithm follows the general methodology introduced in [26, 110, 63]. The idea is to alternate between the recovery of the image information f and the phase errors $\phi(\theta)$. Unique to our approach is the way the phase errors are characterized, specifically that they vary linearly with respect to the spatial frequencies, as written in (5.37). This is significant because the entire phase error,

$$\Phi_{j,n} := \text{ang}(E) = -k_j \phi_n, \quad j = 1, \cdots, K, \quad n = 1, \cdots N_p, \qquad (5.45)$$

is a two-dimensional phase error (rather than one-dimensional), which we correct for in the *raw phase history data*. Moreover, no additional information is lost to pre-processing the data by first forming the image.

In what follows we first explain how to determine the phase correction ϕ in (5.37). We then describe how the phase correction is used in our image formation procedure directly. That is, we approximate f *without* explicit knowledge of the phase error, meaning that we do not require the ideal data $\hat{f}$ in (5.33) to form the image, or equivalently, we are able to accurately estimate $\mathcal{F}^\epsilon$ in (5.40).

Phase Error Correction via Phase Synchronization

Developed in the phase retrieval community for function reconstruction from magnitude only data is an eigenvector-based *phase* (or angular) synchronization [131, 68]. In phase synchronization, one attempts to recover N phases, $e^{i\phi_1}, ..., e^{i\phi_N}$, from measurements of relative phases $e^{i(\phi_i - \phi_j)}$, $i, j = 1, ..., N$, or more simply, to recover individual phases $\phi_1, ..., \phi_N$ from phase differences $(\phi_i - \phi_j)$, $i, j = 1, ..., N$. Using this technique it is possible to construct E in (5.41) which is needed to build the forward operator $\mathcal{F}^\epsilon$ in (5.40). In what follows, we adapt the eigenvector-based phase synchronization technique for the purpose of autofocusing SAR data (5.37).

For simplicity, denote $\hat{f}^\epsilon_{n,j} = \hat{f}^\epsilon_{\theta_n}(k_j)$. For a fixed frequency k_j, define $X_j \in \mathbb{C}^{N_p \times N_p}$ for $j = 1, ..., K$ such that

$$(X_j)_{n,m} := \exp\left\{ i \left[\text{ang}\langle \hat{f}_{n,j}, \hat{f}^\epsilon_{n,j} \rangle - \text{ang}\langle \hat{f}_{m,j}, \hat{f}^\epsilon_{m,j} \rangle \right] \right\}, \tag{5.46}$$

where, for $y \in \mathbb{C}$,

$$\text{ang}(y) = \tan^{-1}\left(\frac{\textbf{Im}\{y\}}{\textbf{Re}\{y\}} \right).$$

Discretizing and substituting (5.37) into (5.46) yields

$$(X_j)_{n,m} = \exp\left\{ i k_j \left(\phi_n - \phi_m \right) \right\}.$$

By construction, when there is no noise present in the data, X_j is a rank 1, Hermitian symmetric matrix. However, noise is always present in measured data, so we define

$$Z_j := \frac{1}{2}\left(X_j + X_j^* \right) \tag{5.47}$$

to mitigate the effects of unwanted errors on the rank and symmetry of X_j. Future investigations will consider relaxing the rank 1 assumption and using a rank minimization method such as the phaselift procedure described in [20]. In the ideal case, because Z_j is Hermitian symmetric, it has an eigenvalue decomposition

$$Z_j = U\Lambda U^*,$$

where U is a unitary matrix made up of the eigenvectors $\{u_1, u_2, ..., u_{N_p}\}$ of Z_j, and $\Lambda = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_{N_p})$ is a diagonal matrix containing its corresponding eigenvalues $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_{N_p}|$. If Z_j is rank 1, Λ contains one unique, non-zero eigenvalue, λ. The corresponding eigenvector is given by

$$Y_j = e^{i\beta_{k_j}} \begin{bmatrix} e^{i(k_j\phi_1)} & e^{i(k_j\phi_2)} & \cdots & e^{i(k_j\phi_{N_p})} \end{bmatrix}^T, \tag{5.48}$$

where β_{k_j} is a constant phase term that is assumed to be dependent on frequency k_j. This is easily shown, because $Y_jY_j^* = X_j$, yielding

$$X_jY_j = Y_j(Y_j^*Y_j) = Y_j||Y_j|| = N_pY_j.$$

Hence the eigenvector Y_j (5.48) corresponds to eigenvalue $\lambda = N_p$.

As mentioned previously, X_j, and consequently Z_j, typically contain noise, in which case there is more than one non-zero eigenvalue. We therefore employ the power iteration method (see e.g. [141]) to approximate the *maximum eigenpair* (λ_1, u_1) of Z_j, where $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_{N_p}|$, and u_1 is the normalized eigenvector corresponding to λ_1. To initialize the power iteration, we set

$$v^{(0)} = [e^{i\Psi_1}, ..., e^{i\Psi_{N_p}}]^T, \quad \text{and} \quad \lambda^{(0)} = (v^{(0)})^T Z_j v^{(0)} \tag{5.49}$$

where Ψ_n is chosen to be independent and identically distributed in $[0, 2\pi]$ for all $n = 1, ..., N_p$. The power iteration algorithm is provided in Algorithm 7, and its convergence is stated in Theorem 5.2.1.

Algorithm 7 Power Iteration: approximate the maximum eigenpair (λ, v) of Z_j, $j = 1, ..., K$.

```
1: Determine tolerance $tol$, initialize vectors $v^{(0)}$ and $\lambda^{(0)}$ according to (5.49) and set $k = 1$. The matrix $Z_j$ is defined in (5.47).
2: while $||\lambda^{(k+1)} - \lambda^{(k)}|| > tol$ do
3:     Set $w = Z_j v^{(k-1)}$.
4:     Compute the normalized eigenvector $v^{(k)} = w/||w||$.
5:     Use the Rayleigh quotient to determine the corresponding eigenvalue as
           $\lambda^{(k)} = (v^{(k)})^T Z_j v^{(k)}.$
6:     Set $k = k + 1$.
7: end while
```

Theorem 5.2.1 *Let (λ_1, u_1) be the maximum eigenpair of matrix $Z_j \in \mathbb{C}^{N_p \times N_p}$. Suppose $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_{N_p}| \geq 0$ and $u_1^T v^{(0)} \neq 0$. Then the iterates of Algorithm 7 satisfy*

$$||v^{(k)} - (\pm u_1)||_2 = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right) \tag{5.50}$$

as $k \to \infty$. The $\pm$ sign means that at each step k, one or the other choice of sign is to be taken, and then the indicated bound holds.

The proof for Theorem 5.2.1 can be found in standard numerical linear algebra textbooks, see e.g. [141].

Figure 5.11 (left) demonstrates the convergence rate of Algorithm 7 given $A \in \mathbb{R}^{100\times 100}$. Here the maximum eigenpair (λ_1, u_1) is real, that is $\lambda_1 \in \mathbb{R}$ and $u_1 \in \mathbb{R}^{100}$. Figure 5.11 (right) displays the final eigenvector approximation along with the true eigenvector corresponding to λ_1. The algorithm is terminated when the relative

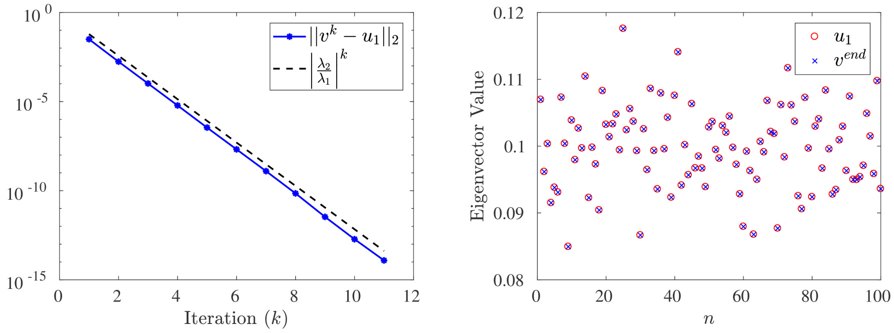

Figure 5.11: The maximum eigenpair approximation of a randomly distributed matrix $A \in \mathbb{R}^{100\times 100}$ calculated using Algorithm 7. (left) Error using Algorithm 7 compared to the theoretical bound in (5.50). (right) Comparison of the actual eigenvector, $u_1 \in \mathbb{R}^{100}$, and estimated eigenvector, v^{end}, resulting from Algorithm 7 after 11 iterations.

change is below a given tolerance. That is,

$$\frac{||v^{(k+1)} - v^{(k)}||_2^2}{||v^{(k)}||_2^2} < tol,$$

where we chose $tol = 10^{-12}$. Figure 5.12 demonstrates the results for the Algorithm when the maximum eigenpair is complex, specifically for when $\lambda_1 \in \mathbb{C}$ and $u_1 \in \mathbb{C}^{100}$. In this case the angle of the maximum eigenvector can be recovered only up to a constant phase shift. However, as will be demonstrated in what follows, this constant phase shift does not affect the overall phase correction, and thus the power iteration is still a reasonable option.

Once the eigenvector $Y_{k_j} = Y_j$ corresponding to the maximum eigenvalue for each frequency $\{k_j\}_{j=1}^K$ is determined, we can build the *phase synchronization matrix* $Y \in \mathbb{C}^{K\times N_p}$ as

$$Y := \begin{bmatrix} Y_1^T \\ Y_2^T \\ \vdots \\ Y_K^T \end{bmatrix} = \begin{bmatrix} e^{i\beta_{k_1}} e^{ik_1\phi_1} & \cdots & e^{i\beta_{k_1}} e^{ik_1\phi_{N_p}} \\ e^{i\beta_{k_2}} e^{ik_2\phi_1} & \cdots & e^{i\beta_{k_2}} e^{ik_2\phi_{N_p}} \\ \vdots & \ddots & \vdots \\ e^{i\beta_{k_K}} e^{ik_K\phi_1} & \cdots & e^{i\beta_{k_K}} e^{ik_K\phi_{N_p}} \end{bmatrix}. \tag{5.51}$$

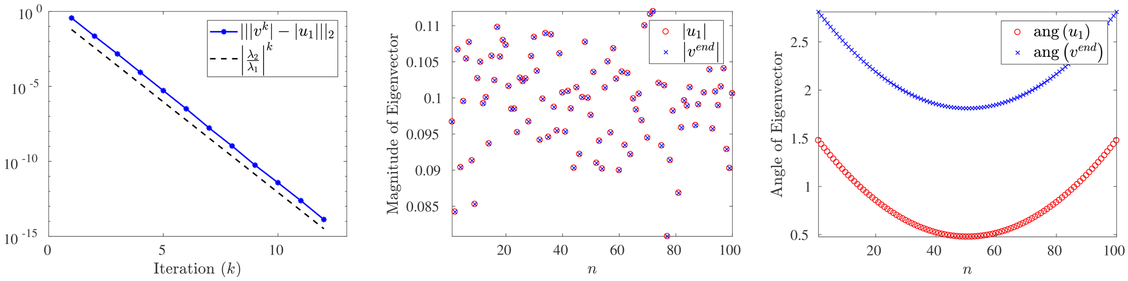

Figure 5.12: The maximum eigenpair approximation of a randomly distributed matrix $A \in \mathbb{C}^{100\times 100}$ calculated using Algorithm 7. (left) Error using Algorithm 7 compared to the theoretical bound in (5.50). Comparison of the magnitude (center) and angle (right) of the actual eigenvectors, $u_1 \in \mathbb{C}^{100}$, and estimated eigenvector, v^{end}, resulting from Algorithm 7 after 12 iterations.

To extract the phase, we sum over the columns of Y for each row $n = 1, ..., N_p$, yielding

$$\frac{\sum_{j=1}^{K} \mathrm{ang}\,(Y_j)_n\, \mathrm{sign}(k_j)}{\sum_{j=1}^{K} |k_j|} = \phi_n + \tilde{\beta}, \tag{5.52}$$

where

$$\tilde{\beta} = \frac{\sum_{j=1}^{K} \beta_{k_j} \mathrm{sign}(k_j)}{\sum_{j=1}^{K} |k_j|}. \tag{5.53}$$

The phase synchronization process separates the phase error $\phi = \{\phi_n + \tilde{\beta}\}_{n=1}^{N_p}$ from the corrupted data $\hat{f}^{\epsilon}$ using (5.52), and is further described in Algorithm 8.

As is apparent from (5.52), the recovery of ϕ includes an unknown phase shift $\tilde{\beta}$, which cannot be removed. Because this phase shift is constant with respect to the azimuth angle θ, it does not affect our phase correction (5.41). For example, Figure 5.13 shows the results using Algorithm 8 to recover from phase corrupted data an unknown phase error defined as

$$\hat{\phi}_n = \gamma \psi_n^2, \quad \psi_n = -1 + \frac{2}{N} n, \quad n = 1, ..., N, \tag{5.54}$$

for $N = 512$ and $\gamma = 17$. In this case we input both the ideal (5.33) and phase

Algorithm 8 Phase Synchronization: recover unknown phase error ϕ given phase corrupted data $\hat{f}^{\epsilon}$.

```
1: Input phase corrupted data $\hat{f}^{\epsilon}$, forward model $\mathcal{F}$, and image estimate $f$.
2: Estimate $\hat{f} = \mathcal{F}f$.
3: for $j = 1$ to $K$ do
4:     Define $X_j$ according to (5.46).
5:     Compute $Z_j$ using (5.47).
6:     Find the maximum eigenvalue $\lambda_j$ and corresponding eigenvector $Y_j$ of $Z_j$ using
       Algorithm 7.
7: end for
8: Construct phase synchronization matrix $Y$ according to (5.51).
9: for $n = 1$ to $N_p$ do
10:    Calculate phase error vector $\phi = \{\phi_n + \tilde{\beta}\}_{n=1}^{N_p}$ using (5.52).
11: end for
```

corrupted (5.37) data into Algorithm 8 so that the only discrepancy between the true phase error and estimated phase error is $\tilde{\beta}$ in (5.52). Specifically, Step 2 of Algorithm 8 is not necessary, as we input the exact $\hat{f}$. We see in Figure 5.13(left) that $\tilde{\beta}$ and ϕ_n are wrapped [5] in the same locations and that $\tilde{\beta}$ possesses a constant shift for each wrapped portion of ϕ_n. When ϕ_n is unwrapped (see Figure 5.13(right)) and compared to the true phase $\hat{\phi}_n$ for all $n = 1, ..., N$, we see that the error is only a constant shift, which does not affect the accuracy of the phase correction (5.41).

Figure 5.13 describes the phase synchronization given exact Fourier data, $\hat{f}$. However, in applications we will only be given the phase corrupted data, $\hat{f}^{\epsilon}_{\theta}(t)$ in (5.37). In this case, we expect that performing an iterative process in Step 2 of Algorithm 8

[5] A phase is considered wrapped when only its principle values (i.e. values that lie between $\pm\pi$) are considered. That is, $\Phi_n = \phi_n + 2\pi\eta_n$ and η_n is an integer function that forces $-\pi < \Phi \leq \pi$.

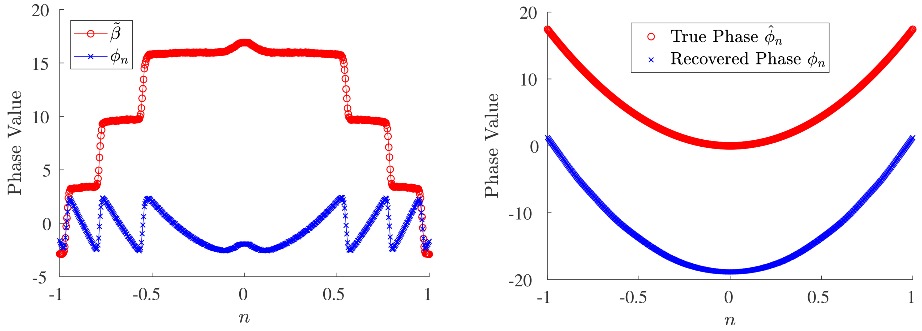

Figure 5.13: (left) The wrapped ϕ and $\tilde{\beta}$ in (5.52) and (right) the final unwrapped phase estimated using Algorithm 8 compared to the true injected phase error (5.54).

to update $\hat{f}$ such that

$$\hat{f}^{new} = \mathcal{F} f^{new}, \tag{5.55}$$

each time the image f is updated should yield a better approximation of the ideal data $\hat{f}$ which is in turn used in Step 4 to calculate (5.46), and subsequently improve the accuracy of Step 10.

Image Formation via High Order Regularization

We can now incorporate the estimated phase error ϕ, found through the phase synchronization technique described in Algorithm 8 into a high order ℓ_1 regularization procedure for estimating the image f. High order regularization has been shown to be a robust and accurate way to reconstruct images from noisy and/or undersampled data [4], and was adapted for for SAR image formation in [120]. The joint estimation of the SAR image with phase error correction is written as

$$\operatorname*{argmin}_{f,\phi} \left\{ ||Tf||_1 + \frac{\mu}{2} ||E\mathcal{F}f - b^{\epsilon}||_2^2 \right\}. \tag{5.56}$$

Here $\mu > 0$, $\mathcal{F}$ is the NUFFT operator [49, 60], and, with ϕ known exactly, $E\mathcal{F} = \mathcal{F}^{\epsilon}$ with $E = E(\phi)$ in (5.41). The phase error ϕ is determined from Algorithm 8. There are several options for choosing the sparsity transform operator T. Due to its demonstrated success in reconstructing SAR images, [120, 121], we choose T to be the (modified) mth order Polynomial Annihilation (PA) transform as given in (5.18).

Numerical Implementation

Due to its robustness and efficiency, we use the alternating direction method of multipliers (ADMM) algorithm [88, 156] to solve (5.56). Other ℓ_1 regularization solvers may be as effective, including the Split-Bregman Algorithm [55] or the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [13], but such comparisons of techniques is not the focus of this investigation so we do not consider them further. Provided below is a concise description of the ADMM as it pertains to (5.56). A more detailed analysis can be found in [120] and Chapter 2 of this dissertation.

Because f cannot be separated from T in (5.18), to implement the ADMM algorithm we must first define slack variables $g \in \mathbb{R}^{N\times N}$ such that $g = Tf$. We then introduce the Lagrange multiplier $\nu \in \mathbb{R}^{N\times N}$, yielding the augmented Lagrangian form of (5.56)

$$J_\nu(f,g,\phi) = \left\{\frac{\mu}{2}||E\mathcal{F}f - b^\epsilon||_2^2 + \frac{\beta}{2}||Tf-g||_2^2 - \langle \nu, Tf-g\rangle + ||g||_1\right\}. \tag{5.57}$$

The approximation to (5.56) is then determined as

$$\underset{f,g,\phi}{\text{argmin}}\ J_\nu(f,g,\phi). \tag{5.58}$$

for fixed multiplier ν. The problem is now split into two sub-problems, known as the g sub-problem and the f sub-problem respectively, which are solved in an alternating fashion. Specifically, at the $k+1$ iteration, for fixed f_k and ν_k, the solution to the g sub-problem is

$$g_{k+1} = \max\left\{|Tf_k - \frac{\nu_k}{\beta}| - \frac{1}{\beta}, 0\right\} \text{sign}\left(Tf_k - \frac{\nu_k}{\beta}\right). \tag{5.59}$$

Holding g_{k+1} and ν_k constant, updates over f then take the form

$$f_{k+1} = f_k - \alpha \nabla_f J_{\nu_k}(f, g_{k+1}, \phi)|_{f_k}, \tag{5.60}$$

where the gradient of J with respect to f is given by

$$\nabla_f J_{\nu_k}(f, g_{k+1}, \phi) = \mu \mathcal{F}^* \left(\mathcal{F} f - E^* b^\epsilon\right) + \beta T^* \left(T f - g_{k+1}\right) - T^* \nu_k, \tag{5.61}$$

and α is chosen as a Barzilai-Borwein step length (see [10]),

$$\alpha_k = \frac{s_k^T s_k}{s_k^T y_k}, \tag{5.62}$$

with

$$s_k = f_k - f_{k-1},$$

$$y_k = \nabla_f J_{\nu_k}(f, g_{k+1}, \phi)|_{f_k} - \nabla_f J_{\nu_k}(f, g_{k+1}, \phi)|_{f_{k-1}}.$$

If the step length does not satisfy the Armijo condition, [156], we backtrack and shorten the step length according to

$$\alpha_k = \rho \alpha_k, \tag{5.63}$$

where $\rho \in (0, 1)$ is chosen as the backtracking parameter. After a sufficient number of updates on g and f are performed, the Lagrange multiplier is updated according to

$$\nu_{k+1} = \nu_k - \beta(T f_{k+1} - g_{k+1}). \tag{5.64}$$

The joint minimization method in Algorithm 9 summarizes what is explained above and provides a step-by-step procedure of how to (alternatively) solve the g sub-problem and f sub-problem at each iteration, while updating ϕ according to the phase synchronization process. Typically, to initialize Algorithm 9, we choose $\rho = .4$ [156, 88].

Algorithm 9 Joint Minimization for SAR Autofocus

```
Determine parameters $\rho$, $\mu$, $\beta$ for the algorithm, set tolerance $tol$, maximum iterations $K$, and initialize $f_0$, $g_0$ and $\nu_0$.
Perform initial estimates of $\Theta^0 = \text{ang}(f_0)$ and $\hat{f}^0 = \mathcal{F} f_0$.
for $i = 0$ to $K$ do
    Determine phase estimate $\phi^i$ using Algorithm 8 with image estimate $f^i$ and the ideal data $\hat{f}^i$.
    while $||f_{k+1} - f_k|| > tol$ do
        Minimize $g$ according to (5.59).
        Update $f$ according to (5.60) and (5.61) with $E(\phi^i)$ in (5.41).
        Update Lagrange multiplier according to (5.64).
    end while
    Set $i = i + 1$.
    Fix $f^i = f_{k+1}$ and update the approximation of ideal data by $\hat{f}^i = \mathcal{F} f^i$.
    Update $T$ using $\Theta^i = \text{ang}(f^i)$ according to (5.18).
end for
```

5.2.5 Numerical Results

For our numerical experiments we first consider the Gotcha parking lot [23] and MSTAR [122] phase history data sets. [6] We take each phase history data set and inject phase errors into the given raw phase history data according to (5.37). In this way we are able to test our method under the more realistic assumption that the phase errors occur in the raw data, not the range compressed data, per the discussion in Section 5.2.3 culminating with (5.44). We next consider a problem where we generate our own corrupted phase history data. This is accomplished by

[6] Both data sets are provided by the Air Force Research Lab and can be downloaded at https://www.sdms.afrl.af.mil/.

adding a small error to the imaging platform location during the data acquisition procedure. Specifically, rather than simply multiplying "good" data by a phase error to simulate corrupted data (i.e. $\hat{f}^{\epsilon} = E(\phi)\hat{f}$), we instead generate data that contain phase corruption without knowing the ideal data, that is, we generate $\hat{f}^{\epsilon}$ without knowing $\hat{f}$. For this example we consider a scene containing a few isotropic point scatterers. This allows a more realistic way to model phase errors without making biased assumptions while also controlling for all other errors in the system.

For each experiment we compare our results to those acquired using high order total variation (HOTV) regularization without phase estimation. That is, we compare our reconstructions to the ones acquired by solving (5.56) with $E(\phi)$ defined as the identity matrix. We note that other regularization transforms may also be utilized for solving the general form of (5.56), and in some cases may yield more accurate results. This will be explored in future investigations.

To demonstrate the recovery of the phase error up to a constant shift, for our examples given below we will display the true two-dimensional phase error, $\{\Phi_{j,n}\}_{j=1,n=1}^{K,N_p}$ in (5.45), along with the recovered two-dimensional phase error defined as

$$\hat{\Phi}_{j,n} = \text{ang}(\hat{E}) = -\hat{\phi}_n k_j, \quad j = 1, \cdots, K, \quad n = 1, \cdots N_p, \tag{5.65}$$

where $\{\hat{\phi}_n\}_{n=1}^{N_p}$ is the final phase error recovered via Algorithm 9. We will also show the differences between the recovered and true two-dimensional phase error,

$$\tilde{\Phi}_{j,n} = \Phi_{j,n} - \hat{\Phi}_{j,n}, \quad j = 1, \cdots, K, \quad n = 1, \cdots N_p. \tag{5.66}$$

We will illustrate that the phase error remains constant for our examples by displaying (5.66) as a one-dimensional plot across spatial frequency values, $j = 1, \cdots, K$ for a subset of azimuth angles indexed by $l \subset \{1, \cdots, N_p\}$. Finally, we approximate the first derivative of $\tilde{\Phi}$ with respect to the spatial frequencies for the same subset of the azimuth angles $l \subset \{1, \cdots, N_p\}$, given by

$$\frac{\partial \tilde{\Phi}}{\partial k} \approx \frac{\tilde{\Phi}_{j,l} - \tilde{\Phi}_{j-1,l}}{\Delta k}, \quad j = 2, ..., K, \tag{5.67}$$

where $\Delta k = (k_K - k_1)/K$ and $\{k_j\}_{j=1}^K$ is defined in (5.38). If (5.67) is near zero, the phase differences (5.66) are indeed constant, as is needed for the success of our autofocusing algorithm.

5.2.6 Injection of Phase Errors

For our first two examples, we are given ideal phase history data $\hat{f}_{\theta_n}(k_j)$, for $n = 1, ..., N_p$, and $j = 1, ..., K$. We inject the phase error into the raw phase history data in accordance to (5.37) as

$$\hat{f}^{\epsilon}_{\theta_n}(k_j) = \hat{f}_{\theta_n}(k_j) e^{-ik_j\phi_n}, \quad n = 1, ..., N_p, \quad j = 1, ..., K. \tag{5.68}$$

Example 5.2.2 *We consider the Gotcha phase history data [23] with a $\theta_a = 4°$ azimuth sweep centered at $30°$ and an elevation of $45.7°$. The center frequency of this data collect is $\omega_c = 9.6GHz$ with a $622.36MHz$ bandwidth. Range resolution is $\rho_y = .2409m$ and cross range resolution is $\rho_x = .2242m$. The phase history data consists of $N_p = 586$ azimuth angles (pulses) and $K = 424$ frequencies. According to (5.68), we inject into the ideal data the following phase error*

$$\phi_n = \gamma \sin(\pi \psi_n), \tag{5.69}$$

where $\gamma = .25$, $\psi_n = -1 + n\frac{2}{N_p}$ and $n = 1, ..., N_p$. To display the one-dimensional error in the phase correction, we choose $l = \{1, 119, 237, 355, 473\}$ in (5.67) so that the difference is calculated for $\theta_l = \{27.01°, 28.01°, 29.02°, 30.03°, 31.03°\}$.

Example 5.2.3 *We utilize the MSTAR data set [122] to reconstruct data corresponding to a scene that contains a T-72 (SNS7) tank where the imaging platform is located*

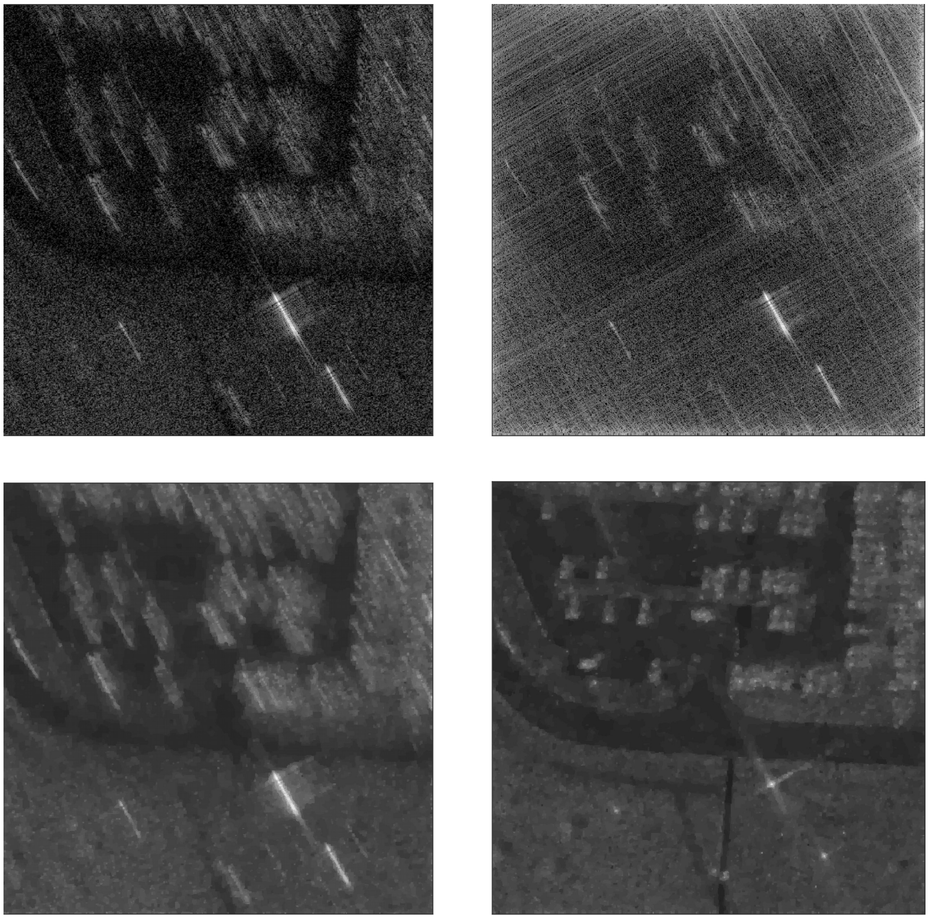

Figure 5.14: Parking lot scene reconstructed from the phase corrupted Gotcha [23] data set described in Example 5.2.2. Reconstructions are done using (top-left) the NUFFT, (top-right) least squares minimization, (bottom-left) HOTV regularization with order $m = 2$ and parameters $\mu = 128$ and $\beta = 60$ and (bottom-right) the proposed joint image formation and phase estimation (5.56) with order PA $m = 2$, $\mu = 128$ and $\beta = 60$.

4551m above the scene center at a 15° elevation angle. For this data collect, the center frequency is set at $\omega_c = 9.6GHz$ *with wavelength* $\lambda = c/\omega_c = .0312m$*, a bandwidth of* $B = 591MHz$ *and an integration angle of* $\theta_a = 2.9361°$*. The range and cross range*

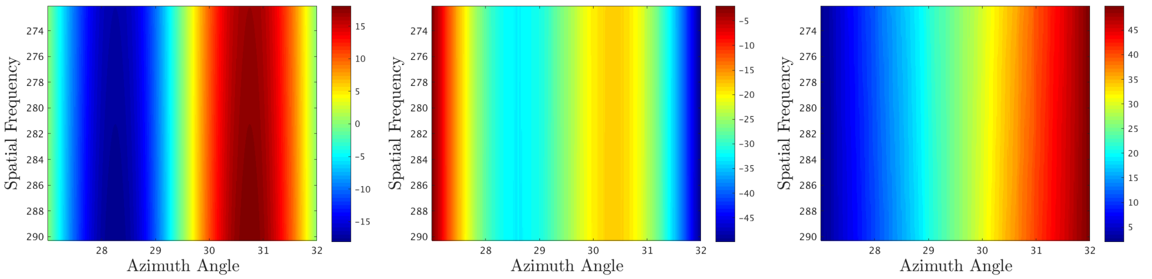

Figure 5.15: Result of estimating the phase error in the Gotcha parking lot data, Example 5.2.2. (left) True two-dimensional phase error injected into data. (middle) Two-dimensional phase correction recovered using proposed autofocusing technique. (right) The difference between the recovered and true phase, (5.66). Note the different scales in each image. This is acceptable, as we expect that for each azimuth angle there will be a constant phase shift for all frequency values.

resolutions are therefore

$$\rho_x = \frac{c}{2B} = .2536m \quad \textit{and} \quad \rho_y = \frac{\lambda}{2\theta_a} = .3047m.$$

In this case we corrupt the phase history data using

$$\phi_n = \gamma\psi_n^3 \tag{5.70}$$

in (5.68), where $\gamma = .25$, $\psi_n = -1 + n\frac{2}{N_p}$ *and* $n = 1, ..., N_p$. *To display the one-dimensional error in the phase correction, we choose* $l = \{1, 27, 53, 79, 105\}$ *in (5.67) so that the difference is calculated for* $\theta_l = \{-1.47°, -0.87°, -0.27°, 0.34°, 0.94°\}$.

In Examples 5.2.2 and 5.2.3 we use the corrupt phase history data (5.68) as the input of our joint image formation and phase correction technique described in Algorithm 9. We compare our results to using the inverse (adjoint) NUFFT operator, the least squares solution, and the HOTV solution. The reconstruction results are respectively displayed in Figures 5.14 and 5.17 for the Gotcha data and the MSTAR

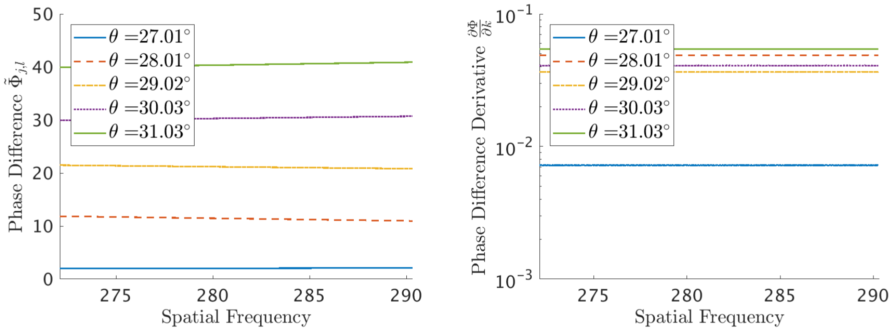

Figure 5.16: Result of estimating the phase error in the Gotcha parking lot data, Example 5.2.2. (left) Phase difference (5.66) at specific azimuth angles for all spatial frequencies. (right) Approximate derivative (5.67) of the one-dimensional phase differences at the same subset of azimuth angles.

data. In Example 5.2.2 we use $m = 2$, $\mu = 128$ and $\beta = 60$, while in Example 5.2.3 we use $m = 1$, $\mu = 70$ and $\beta = 20$. These parameters were chosen because of their performance value, but were not optimized or tested for robustness, which will be the topic of future research. The results clearly demonstrate that we are able to focus the phase corrupted data using Algorithm 9. Moreover, it is apparent that our autofocusing algorithm can be utilized both on full scenes consisting of many targets, (as in the Gotcha image), as well as on scenes where software has honed in on a specific target of interest (as in the MSTAR image).

Figures 5.15 and 5.18 show the recovered two-dimensional (5.65) phase errors outputted from Algorithm 9 compared to the true two-dimensional phase error (5.45). The discrepancy between the two errors, (5.66), is also displayed. Figures 5.16 and 5.19 demonstrate that the error in the recovery is due to the constant phase shift $\tilde{\beta}$ in (5.52) and inherent algorithm errors due to noise.

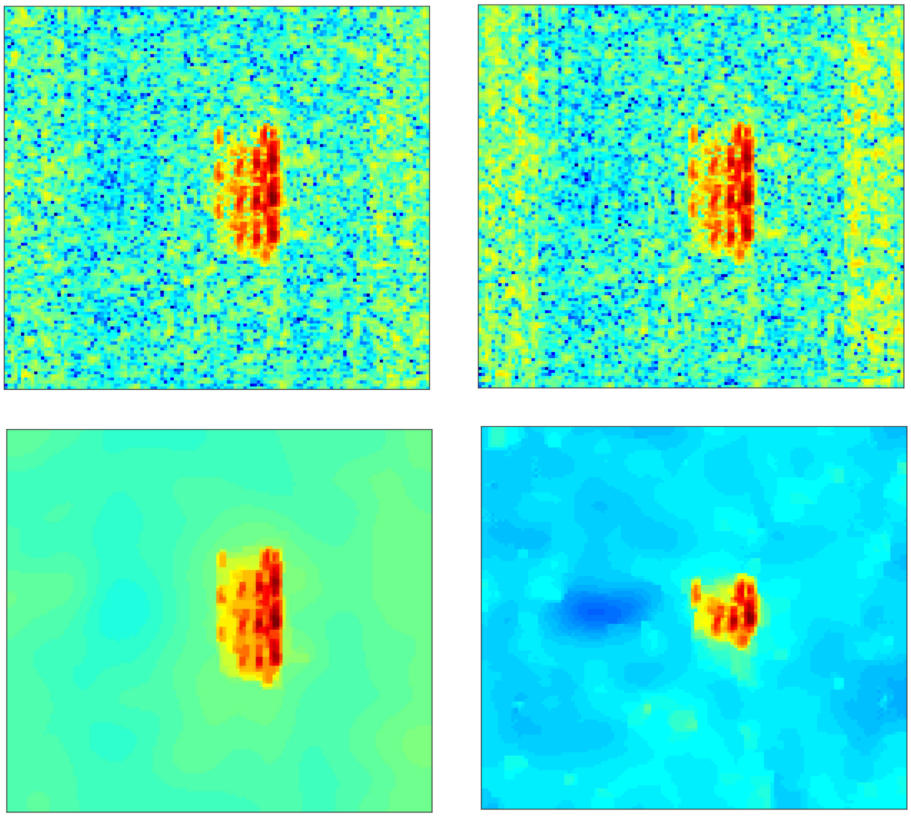

Figure 5.17: T-72 tank reconstructed from the phase corrupted MSTAR [122] data set described in Example 5.2.3. Reconstructions are done using (top-left) the NUFFT, (top-right) least squares minimization, (bottom-left) HOTV regularization with order $m = 1$ and parameters $\mu = 70$ and $\beta = 20$ and (bottom-right) the proposed joint image formation and phase estimation (5.56) with order PA $m = 1$, $\mu = 70$ and $\beta = 20$.

Generation of Phase Errors

As a final experiment, we simulate phase history data (5.33) that has been acquired without perfect knowledge of the SAR imaging platform. To do so, we discretize according to (5.37) and (5.39) to find $\hat{f}_{\theta_n}(k_j) = \hat{f}_{n,j}$ for $n = 1, ..., N_p$ and $j = 1, ..., K$. In this way, the following model, originally posed in [58], is used to acquire data from

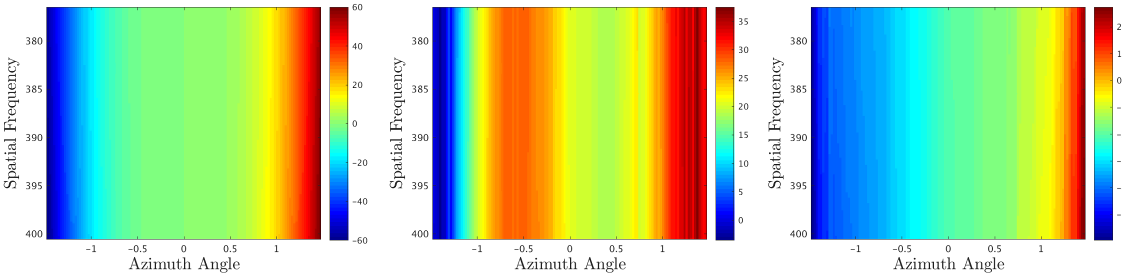

Figure 5.18: Result of estimating the phase error in the MSTAR data, Example 5.2.3. (left) True two-dimensional phase error injected into data. (middle) Two-dimensional phase correction recovered using proposed autofocusing technique. (right) The difference between the true and recovered phase, (5.66). Notice that the scales are different in each image. This is acceptable, as we expect that for each azimuth angle there will be a constant phase shift for all frequency values.

a scene containing M scatterers:

$$\hat{f}_{n,j} = \sum_{m=1}^{M} A_m \exp\left(\frac{-i4\pi \Delta R_m(\theta_n) k_j}{c} \right). \tag{5.71}$$

Here c is the speed of light, A_m is the amplitude of the mth scatterer, and $\Delta R_m(\theta_n)$ is the differential range to the mth scatterer from angle n. The differential range is defined as the distance from the scene center to a scatterer. Below we provide brief discussion on differential range and how it is affected by the imperfect knowledge of the imaging platform location. A thorough discussion of this discrete model (5.71) and the differential range in the single scatterer case is provided in Appendix A.

For a multiple scatterer scene, let $\tilde{\boldsymbol{x}}_n = [x_a(\theta_n) \quad y_a(\theta_n) \quad z_a(\theta_n)]^T \in \mathbb{R}^3$ be the location of the SAR antenna center at angle θ_n and $\boldsymbol{x}_n = [x_m(\theta_n) \quad y_m(\theta_n) \quad z_m(\theta_n)]^T \in \mathbb{R}^3$ be the location of the mth scatterer at angle θ_n. Then the distance from the antenna center to the scene center is

$$R(\tilde{\boldsymbol{x}}_n) = \sqrt{x_a(\theta_n)^2 + y_a(\theta_n)^2 + z_a(\theta_n)^2}, \tag{5.72}$$

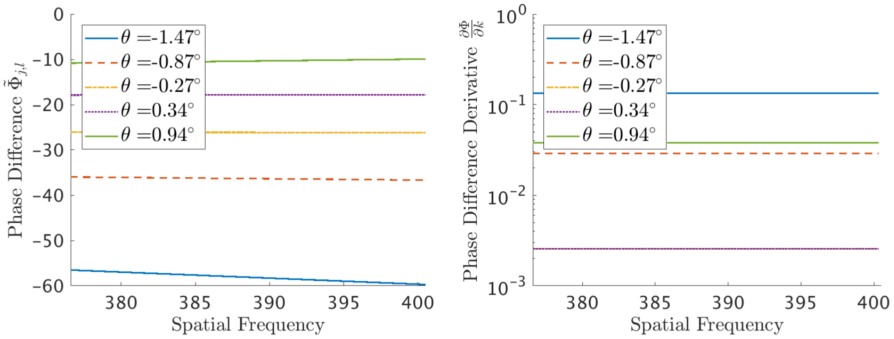

Figure 5.19: Result of estimating the phase error in the MSTAR data, Example 5.2.3. (left) Phase difference (5.66) at specific azimuth angles for all spatial frequencies. (right) Approximate derivative (5.67) of the one-dimensional phase differences at the same subset of azimuth angles.

and the distance from the antenna phase center to the mth scatterer is

$$R_m(\boldsymbol{x}_n) = \sqrt{(x_a(\theta_n) - x_m(\theta_n))^2 + (y_a(\theta_n) - y_m(\theta_n))^2 + (z_a(\theta_n) - z_m(\theta_n))^2}. \quad (5.73)$$

The differential range is then

$$\Delta R_m(\theta_n) = R_m(\boldsymbol{x}_n) - R(\tilde{\boldsymbol{x}}_n). \quad (5.74)$$

The distances described by (5.72)-(5.74) have been labeled in Figure 5.20 for clarity. Because the location of the antenna is not known exactly, we assume

$$R^{\epsilon}(\tilde{\boldsymbol{x}}_n) = R(\tilde{\boldsymbol{x}}_n) + \epsilon_n = \sqrt{x_a(\theta_n)^2 + y_a(\theta_n)^2 + z_a(\theta_n)^2} + \epsilon_n, \quad (5.75)$$

where ϵ_n represents some shift in space and $\phi_n = \epsilon_n c/2$ is the corresponding phase shift. When including the antenna location error, the differential range in (5.74) becomes

$$\Delta R^{\epsilon}_m(\theta_n) = R_m(\boldsymbol{x}_n) - R^{\epsilon}(\tilde{\boldsymbol{x}}_n). \quad (5.76)$$

We therefore approximate the phase corrupted discrete data (5.39) as

$$\begin{aligned}\hat{f}^{\epsilon}_{\theta_n}(k_j) \approx \hat{f}^{\epsilon}_{n,j} &:= \sum_{m=1}^{M} A_m \exp\left(\frac{-i4\pi\Delta R^{\epsilon}_m(\theta_n)k_j}{c}\right) \\ &= \sum_{m=1}^{M} A_m \exp\left(\frac{-i4\pi(\Delta R_m(\theta_n) - \epsilon_n)k_j}{c}\right),\end{aligned} \tag{5.77}$$

where A_m is again the amplitude of the mth scatterer and $\Delta R^{\epsilon}_m(\theta_n)$ is the *inexact* differential range to the mth scatterer at angle θ_n.

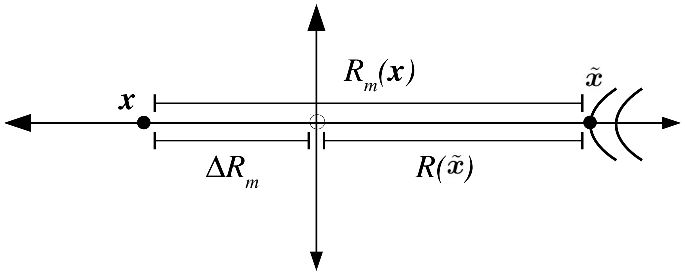

Figure 5.20: Depiction of the ranges calculated in (5.72)-(5.74) for a scene consisting of a single scatterer in one dimension (all y and z coordinates set to zero with $n = 0$).

Remark 5.2.4 *Clearly (5.71) and (5.77) do not account for all scattering phenomenology. However, by using (5.77), we are able to discern that all additional sources of error are caused by the phase.*

Example 5.2.5 *We place $M = 10$ ten scatterers randomly within our scene. Corrupted phase history data are generated according to (5.77) and typical SAR system parameters. Specifically, we use $N_p = 512$ linearly spaced azimuth angles ranging from $\theta_1 = -1.5°$ to $\theta_{N_p} = 1.5$ giving an integration angle of $\theta_a = 3°$. We consider are $K = 512$ frequencies centered at $\omega_c = 9.6GHz$ with a bandwidth of $B = 500MHz$ giving a wavelength of $\lambda = c/\omega_c = .0312m$. Our SAR platform is assumed to be at an elevation of 30° and altitude of 5km. Together this gives us a range resolution ρ_x and*

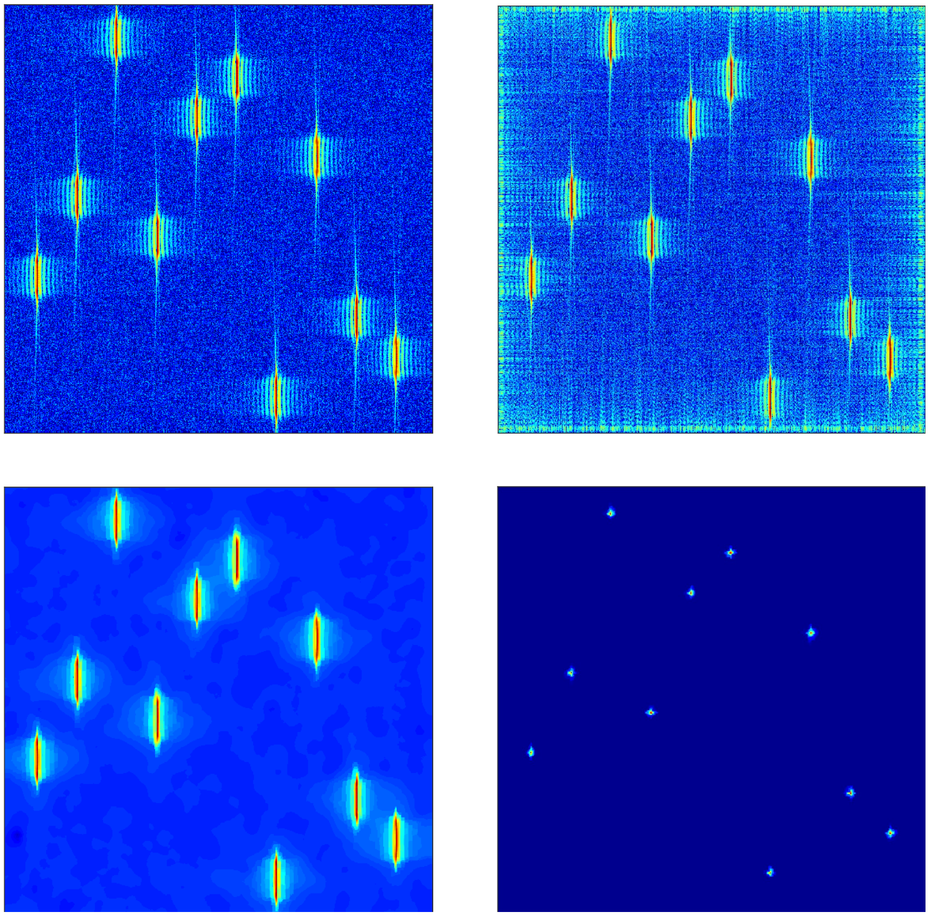

Figure 5.21: Ten target scene reconstructed from phase corrupted data described in Example 5.2.5. Reconstructions are done using (top-left) the NUFFT, (top-right) least squares minimization, (bottom-left) HOTV regularization with order $m = 2$ and parameters $\mu = 40$ and $\beta = 100$ and (bottom-right) the proposed joint image formation and phase estimation (5.56) with order PA $m = 2$, $\mu = 40$ and $\beta = 100$.

cross range resolution ρ_y of

$$\rho_x = \frac{c}{2B} = .3m, \quad \rho_y = \frac{\lambda}{2\theta_a} = .3m. \tag{5.78}$$

To generate the data using (5.77) we define the phase error as

$$\epsilon_n = \gamma\psi_n^2, \tag{5.79}$$

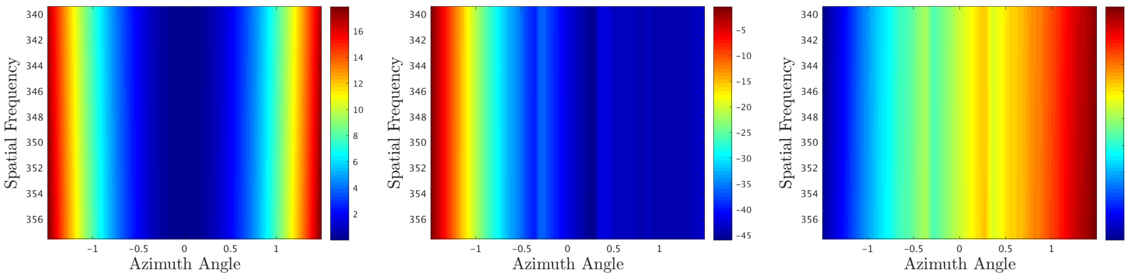

Figure 5.22: Result of estimating the phase error in the data representing the ten target scene, Example 5.2.5. (left) True two-dimensional phase error injected into data. (middle) Two-dimensional phase correction recovered using proposed autofocusing technique. (right) The difference between the recovered phase and the true phase (5.66). Notice that the scales are different in each image. This is acceptable, as we expect that for each azimuth angle there will be a constant phase shift for all frequency values.

where $\gamma = .05$*,* $\psi_n = -1 + n\frac{2}{N_p}$ *and* $n = 1, ..., N_p$*. Before the inversion process we also corrupt the data with additive complex Gaussian noise with mean zero and variance .5. Finally, we choose* $\mathcal{F}$ *in the forward model (5.56) to be the NUFFT [49, 60, 107]. To display the one-dimensional error in the phase correction, we choose* $l = \{1, 104, 207, 310, 413\}$ *in (5.67) so that the difference is calculated for* $\theta_l = \{-1.5°, -0.89°, -0.29°, 0.31°, 0.91°\}$*.*

We use Algorithm 9 to reconstruct a representation of the scene described in Example (5.2.5) from the corrupt phase history data (5.77). We chose regularization parameters $\mu = 40$, $\beta = 100$ and PA order $m = 2$. As before, the parameters were chosen for their high performance value but were not optimized or tested for robustness. Figure 5.21 compares our scene reconstruction to those using the inverse (adjoint) NUFFT operator, HOTV regularization, and least squares estimation. While HOTV regularization helps to eliminate noise in the reconstruction, it is clear that without

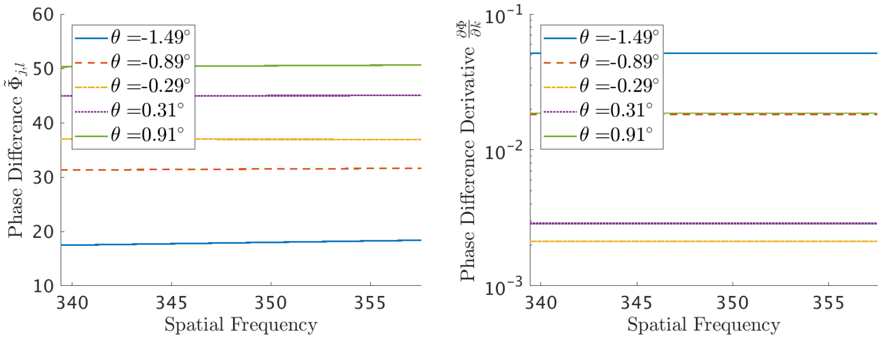

Figure 5.23: Result of estimating the phase error in the data representing the ten target scene, Example 5.2.5. (left) Phase difference (5.66) at specific azimuth angles for all spatial frequencies. (right) Approximate derivative (5.67) of the one-dimensional phase differences at the same subset of azimuth angles.

the phase estimation the targets remained blurred and their true locations cannot be discerned. Figure 5.22 displays the true two-dimensional phase error, (5.45), alongside the two-dimensional phase correction, (5.65), outputted from Algorithm 9, as well as the difference between the two. It is evident from Figure 5.23 that as in the previous examples, the phase is recovered up to a constant phase shift. This is further made apparent in Figures 5.14(right), 5.17(right), and 5.21(right), which show this constant phase shift for each azimuth angle across all spatial frequencies.

5.2.7 Discussion and Conclusions

In this section we analyzed the effects of imperfect imaging platform location measurements on SAR phase history data and showed how it is manifested as a two-dimensional phase error on the raw phase history data. Previous investigations tried to correct this error on range compressed data, but as demonstrated by the derivation of (5.43), such techniques omit the dependence of the two-dimensional phase error on the frequency values.

Instead we propose a joint phase estimation and image formation optimization procedure to autofocus SAR imagery. As the image is updated using HOTV regularization, we adjust the phase error using a phase synchronization technique, which allows us to recover the unknown phase error up to a constant shift. Additionally, the use of HOTV regularization reduces speckle and other sources of noise in the resulting image. Compared to other ℓ_1 regularization techniques, our method incurs additional computational complexity arising from calculating of K eigenvectors at each iteration to form the phase synchronization matrix. Future investigations will consider more efficient eigenvalue problem solvers. We will also investigate the sensitivity of our method to HOTV parameter fluctuations in a Bayesian framework.

Two different numerical experiments are included in this section. In the first, we inject two dimensional phase error into the Gotcha parking lot and MSTAR focused phase history data sets to create phase corrupted data. In the second, we generate phase corrupted data without explicitly knowing the ideal data. We consider sinusoidal, quadratic and cubic phase errors, each with different maximum magnitudes. We compare our results to the least squares solution, the inverse (adjoint) NUFFT solution and HOTV regularization without phase estimation. Our results show that in every case we are able to recover the appropriate phase correction and focus the imagery.

Chapter 6

CONCLUSIONS AND FUTURE WORK

The goal of this dissertation was to develop and analyze high order total variation (HOTV) techniques for robust function approximation in ill-posed inverse problems. Our major assumption throughout this dissertation is that the one and two dimensional functions we consider are sparse in the space of discontinuities/edges. We are then able to exploit this prior knowledge by using the Polynomial Annihilation transformation in an ℓ_1 regularization formulation.

After reviewing fundamental concepts in ℓ_1 regularization and optimization in Chapter 2, in Chapter 3 we developed a robust variance based joint sparsity (VBJS) technique for recovering one and two dimensional functions from multiple measurement vectors. We show that the accuracy of this VBJS technique does not decrease when a subset of the measurement vectors contain false or misleading information. Moreover, the VBJS is essentially non-parametric, allowing for autonomous implementation and further algorithm robustness.

In Chapter 4 we tackle the problem of approximating solutions to nonlinear hyperbolic partial differential equations (PDEs). Even when given smooth initial data, solutions to these types of problems develop shocks and discontinuities in finite time, making it hard to find robust numerical approximations to solutions. We propose an ℓ_1 enhanced numerical solver that augments current numerical solvers with ℓ_1 regularization. In this way, we are able to maintain stability and accuracy far beyond classical restrictions.

Chapter 5 discusses reducing model assumptions in synthetic aperture radar (SAR) image formation. First we develop two novel speckle reduction techniques. One is based on the ℓ_1 enhanced numerical PDE solver we developed in Chapter 4, and the other is an adaptation of the VBJS method described in Chapter 3. Our numerical results show that though both techniques are successful at reducing speckle, the VBJS method better maintains target intensities, which is essential for object detection and recognition. Further, VBJS performs comparably to HOTV regularization, but the need for parameter tuning is eliminated, allowing for autonomous implementation. We note, however, that the numerical method used for solving the speckle reduction PDE model was low order, so it is possible that the results may be improved by incorporating a more sophisticated solver.

The incorrect assumption that the SAR imaging platform location is known perfectly yields a phase error on the raw phase history data, resulting in defocused imagery. We develop a new autofocusing technique that, at each iteration, jointly estimates the appropriate phase correction and the resulting image. Our phase correction estimation is based on the phase (angular) synchronization technique utilized throughout the phase retrieval community, and we use HOTV regularization for our image formation algorithm. Our results show that with our algorithm, we are able to produce a focused, speckle reduced image where the true phase error is recovered up to a constant phase shift.

The results of this dissertation strongly support the idea that high order regularization methods are effective at solving a wide range of ill-posed inverse problems. In particular, they provide robust, efficient, and accurate means for solving real world problems, as was demonstrated here for reconstructing SAR images and approximating solutions to conservation laws. Possible broader application areas range from hyper-spectral and magnetic resonance imaging to threat detection.

6.1 Future Work

Much of my future work will be focused on adapting the VBJS method for Air Force applications. First, we will work to use the VBJS method for 3D SAR image reconstruction. In this scenario, we must determine the optimal way to splice the SAR data cube to obtain multiple measurement vectors that will fit into the VBJS framework. Next, we will explore the possibility of data fusion using VBJS, where the multiple measurement vectors will be data from a radio-frequency sensor (SAR) and data from an electro-optical (EO) sensor. The fusion of SAR and EO is difficult due to the different scattering phenomenologies of each sensor and different projection spaces. The sensors will have the same support as they are both obtaining reflected energy from the latent geometry of the object; however, determining the joint support in the signal and image space will challenging, as the specular nature of SAR will only coincide with the diffuse EO reflections at discrete 3D points on the objects surface. In addition, only the projection of these points will be available to the respective sensors.

Another project of interest is using the VBJS method to generate a database of SAR images that can be used in the training and testing of SAR automated target recognition (ATR) algorithms. The goal of the project is to improve the performance of the SAR ATR algorithms using the new image database. This is important because in SAR ATR, a small performance increase could result in the ATR being deployed on a real system. The proposed VBJS speckle reduction technique was shown in this thesis to remove background clutter while maintaining target fidelity and autonomy. Thus, the images in the new database will contain less noise and higher target to clutter ratios without the need to tune parameters. By generating new testing and training sets with this algorithm, SAR ATR algorithms can be retrained and then tested for performance increases or decreases. The algorithm will be tested under different operating conditions and constraints such as target obscuration and orientation.

Finally, we also will propose using joint sparsity to develop a predictor/corrector method for approximating solutions to nonlinear hyperbolic PDEs. This will require advancing the numerical solution of the PDE ahead k time steps, and then using the k solutions as multiple measurement vectors in the classic joint sparsity set up. Here we will use joint sparsity, and not VBJS, because we expect the solutions to be jointly sparse, but not jointly sparse at the same spatial locations. By performing the $\ell_{2,1}$ regularization across measurements, we can update the solution to the PDE over the k time steps and then continue to advance the solution.

REFERENCES

[1] Abramowitz, M. and I. A. Stegun, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, vol. 55 (Courier Corporation, 1964).

[2] Adcock, B., A. Gelb, G. Song and Y. Sui, "Joint sparse recovery based on variances", SIAM Journal on Scientific Computing (submitted).

[3] Ao, D., R. Wang, C. Hu and Y. Li, "A sparse SAR imaging method based on multiple measurement vectors model", Remote Sensing **9**, 3, 297 (2017).

[4] Archibald, R., A. Gelb and R. B. Platte, "Image reconstruction from undersampled Fourier data using the Polynomial Annihilation transform", Journal of Scientific Computing **67**, 2, 432–452 (2016).

[5] Archibald, R., A. Gelb and J. Yoon, "Polynomial fitting for edge detection in irregularly sampled signals and images", SIAM Journal on Numerical Analysis **43**, 1, 259–279 (2005).

[6] Argenti, F., A. Lapini, T. Bianchi and L. Alparone, "A tutorial on speckle reduction in synthetic aperture radar images", IEEE Geoscience and Remote Sensing Magazine **1**, 3, 6–35 (2013).

[7] Ash, J. N., "An autofocus method for backprojection imagery in synthetic aperture radar", IEEE Geoscience and Remote Sensing Letters **9**, 1, 104–108 (2012).

[8] Aubert, G. and J.-F. Aujol, "A nonconvex model to remove multiplicative noise", in "International Conference on Scale Space and Variational Methods in Computer Vision", pp. 68–79 (Springer, 2007).

[9] Aubert, G. and J.-F. Aujol, "A variational approach to removing multiplicative noise", SIAM Journal on Applied Mathematics **68**, 4, 925–946 (2008).

[10] Barzilai, J. and J. M. Borwein, "Two-point step size gradient methods", IMA Journal of Numerical Analysis **8**, 1, 141–148 (1988).

[11] Bauschke, H. H. and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, vol. 408 (Springer, 2011).

[12] Beck, A., *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB* (SIAM, 2014).

[13] Beck, A. and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems", SIAM Journal on Imaging Sciences **2**, 1, 183–202 (2009).

[14] Bertsekas, D. P., "Necessary and sufficient conditions for a penalty method to be exact", Mathematical Programming **9**, 1, 87–99 (1975).

[15] Bredies, K., K. Kunisch and T. Pock, "Total generalized variation", SIAM Journal on Imaging Sciences **3**, 3, 492–526 (2010).

[16] Brugiapaglia, S., S. Micheletti and S. Perotto, "COmpRessed SolvING: A numerical approximation technique for elliptic PDEs based on compressed sensing", Computers & Mathematics with Applications **70**, 6, 1306–1335 (2015).

[17] Brugiapaglia, S., F. Nobile, S. Micheletti and S. Perotto, "A theoretical study of COmpRessed SolvING for advection-diffusion-reaction problems", Mathematics of Computation **87**, 309, 1–38 (2018).

[18] Candès, E. J., J. Romberg and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information", IEEE Transactions on Information Theory **52**, 2, 489–509 (2006).

[19] Candès, E. J. and J. K. Romberg, "Signal recovery from random projections", in "Electronic Imaging 2005", pp. 76–86 (International Society for Optics and Photonics, 2005).

[20] Candès, E. J., T. Strohmer and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming", Communications on Pure and Applied Mathematics **66**, 8, 1241–1274 (2013).

[21] Candès, E. J., M. B. Wakin and S. P. Boyd, "Enhancing sparsity by reweighted l1 minimization", Journal of Fourier Analysis and Applications **14**, 5, 877–905 (2008).

[22] Canuto, C., M. Y. Hussaini, A. M. Quarteroni, *et al.*, *Spectral methods in fluid dynamics* (Springer Science & Business Media, 2012).

[23] Casteel, C. and L. Gorham, "Gotcha volumetric SAR data set overview", `https://www.sdms.afrl.af.mil/index.php?collection=gotcha`, [Online; accessed 29-June-2017] (2007).

[24] Çetin, M., *Feature-enhanced synthetic aperture radar imaging*, Ph.D. thesis, University of Salford, Manchester, UK, 1995 (2001).

[25] Çetin, M. and W. C. Karl, "Feature-enhanced synthetic aperture radar image formation based on nonquadratic regularization", IEEE Transactions on Image Processing **10**, 4, 623–631 (2001).

[26] Çetin, M., Ö. Onhon and S. Samadi, "Handling phase in sparse reconstruction for SAR: Imaging, autofocusing, and moving targets", in "9th European Conference on Synthetic Aperture Radar", pp. 207–210 (VDE, 2012).

[27] Çetin, M., I. Stojanovic, Ö. Onhon, K. Varshney, S. Samadi, W. C. Karl and A. S. Willsky, "Sparsity-driven synthetic aperture radar imaging: Reconstruction, autofocusing, moving targets, and compressed sensing", IEEE Signal Processing Magazine **31**, 4, 27–40 (2014).

[28] Chan, T., A. Marquina and P. Mulet, "High-order total variation-based image restoration", SIAM Journal on Scientific Computing **22**, 2, 503–516 (2000).

[29] Chartrand, R. and W. Yin, "Iteratively reweighted algorithms for compressive sensing", in "IEEE International Conference on Acoustics, Speech and Signal Processing", pp. 3869–3872 (IEEE, 2008).

[30] Chen, D.-Q. and L.-Z. Cheng, "Spatially adapted total variation model to remove multiplicative noise", IEEE Transactions on Image Processing **21**, 4, 1650–1662 (2012).

[31] Chen, J. and X. Huo, "Theoretical results on sparse representations of multiple-measurement vectors", IEEE Transactions on Signal Processing **54**, 12, 4634–4643 (2006).

[32] Cheney, M. and B. Borden, *Fundamentals of radar imaging* (SIAM, 2009).

[33] Cotter, S. F., B. D. Rao, K. Engan and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors", IEEE Transactions on Signal Processing **53**, 7, 2477–2488 (2005).

[34] Dainty, J. C., *Laser speckle and related phenomena*, vol. 9 (Springer Science & Business Media, 2013).

[35] Danaila, I., P. Joly, S. M. Kaber and M. Postel, *An introduction to scientific computing: Twelve computational projects solved with MATLAB* (Springer Science & Business Media, 2007).

[36] Daubechies, I., R. DeVore, M. Fornasier and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery", Communications on Pure and Applied Mathematics **63**, 1, 1–38 (2010).

[37] Deng, W., W. Yin and Y. Zhang, "Group sparse optimization by alternating direction method", Tech. rep., Rice University, Houston, TX, Department of Computational and Applied Mathematics (2012).

[38] Denker, D. and A. Gelb, "Edge detection of piecewise smooth functions from under sampled Fourier data using variance signatures", SIAM Journal on Scientific Computing **39**, 2, A559–A592 (2017).

[39] Desai, M. D. and W. K. Jenkins, "Convolution backprojection image reconstruction for spotlight mode synthetic aperture radar", IEEE Transactions on Image Processing **1**, 4, 505–517 (1992).

[40] Don, W.-S., Z. Gao, P. Li and X. Wen, "Hybrid compact-WENO finite difference scheme with conjugate Fourier shock detection algorithm for hyperbolic conservation laws", SIAM Journal on Scientific Computing **38**, 2, A691–A711 (2016).

[41] Donoho, D. L., "Compressed sensing", IEEE Transactions on Information Theory **52**, 4, 1289–1306 (2006).

[42] Driscoll, T. A., N. Hale and L. N. Trefethen, “Chebfun guide”, (2014).

[43] Durand, S. and J. Froment, “Reconstruction of wavelet coefficients using total variation minimization”, SIAM Journal on Scientific computing **24**, 5, 1754–1767 (2003).

[44] Eaves, J. and E. Reedy, *Principles of modern radar* (Springer Science & Business Media, 2012).

[45] Eichel, P., D. Ghiglia and C. Jakowatz, “Speckle processing method for synthetic aperture radar phase correction”, Optics letters **14**, 1, 1–3 (1989).

[46] Eldar, Y. C. and M. Mishali, “Robust recovery of signals from a structured union of subspaces”, IEEE Transactions on Information Theory **55**, 11, 5302–5316 (2009).

[47] Eldar, Y. C. and H. Rauhut, “Average case analysis of multichannel sparse recovery using convex relaxation”, IEEE Transactions on Information Theory **56**, 1, 505–519 (2010).

[48] Fessler, J., “Analytical tomographic image reconstruction methods”, Image Reconstruction: Algorithms and Analysis **66**, 67 (2009).

[49] Fessler, J. A. and B. P. Sutton, “Nonuniform fast Fourier transforms using min-max interpolation”, IEEE Transactions on Signal Processing **51**, 2, 560–574 (2003).

[50] Fienup, J. R., “Phase error correction by shear averaging”, Signal Recovery and Synthesis II **15**, 134–137 (1989).

[51] Fienup, J. R., “Detecting moving targets in SAR imagery by focusing”, IEEE Transactions on Aerospace and Electronic Systems **37**, 3, 794–809 (2001).

[52] Fu, W., S. Li, L. Fang, X. Kang and J. A. Benediktsson, “Hyperspectral image classification via shape-adaptive joint sparse representation”, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **9**, 2, 556–567 (2016).

[53] Gibson, G., “Constructive and destructive interference”, `http://www.phys.uconn.edu/~gibson/Notes/Section5_2/Sec5_2.htm`, [Online; accessed 24-Jan-2017] (n.d.).

[54] Glowinski, R. and P. Le Tallec, *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics* (SIAM, 1989).

[55] Goldstein, T. and S. Osher, “The split Bregman method for l1-regularized problems”, SIAM Journal on Imaging Sciences **2**, 2, 323–343 (2009).

[56] Goodman, J. W., *Speckle phenomena in optics: theory and applications* (Roberts and Company Publishers, 2007).

[57] Goodman, J. W., *Statistical optics* (John Wiley & Sons, 2015).

[58] Gorham, L. A. and L. J. Moore, “SAR image formation toolbox for MATLAB”, in “SPIE Defense, Security, and Sensing”, pp. 769906–769906 (International Society for Optics and Photonics, 2010).

[59] Gottlieb, D. and C.-W. Shu, “On the Gibbs phenomenon and its resolution”, SIAM review **39**, 4, 644–668 (1997).

[60] Greengard, L. and J.-Y. Lee, “Accelerating the nonuniform fast Fourier transform”, SIAM review **46**, 3, 443–454 (2004).

[61] Guermond, J.-L., F. Marpeau, B. Popov *et al.*, “A fast algorithm for solving first-order PDEs by l1-minimization”, Communications in mathematical Sciences **6**, 1, 199–216 (2008).

[62] Gunawardena, A. and D. Longstaff, “A matched filter based synthetic aperture radar (SAR) algorithm for stepped frequency ground penetrating radar”, in “Radar Conference, 1995., Record of the IEEE 1995 International”, pp. 239–243 (IEEE, 1995).

[63] Güngör, A., M. Çetin and H. E. Güven, “An augmented Lagrangian method for autofocused compressed SAR imaging”, in “Compressed Sensing Theory and its Applications to Radar, Sonar and Remote Sensing (CoSeRa), 2015 3rd International Workshop on”, pp. 1–5 (IEEE, 2015).

[64] Hansen, P. C., *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion* (SIAM, 1998).

[65] Hazlett, M., D. J. Andersh, S. W. Lee, H. Ling and C. Yu, “XPATCH: a high-frequency electromagnetic scattering prediction code using shooting and bouncing rays”, in “SPIE’s 1995 Symposium on OE/Aerospace Sensing and Dual Use Photonics”, pp. 266–275 (International Society for Optics and Photonics, 1995).

[66] Hesthaven, J. S., S. Gottlieb and D. Gottlieb, *Spectral methods for time-dependent problems*, vol. 21 (Cambridge University Press, 2007).

[67] Hou, T. Y., Q. Li and H. Schaeffer, “Sparse + low-energy decomposition for viscous conservation laws”, Journal of Computational Physics **288**, 150–166 (2015).

[68] Iwen, M. A., B. Preskitt, R. Saab and A. Viswanathan, “Phase retrieval from local measurements: Improved robustness via eigenvector-based angular synchronization”, arXiv preprint arXiv:1612.01182 (2016).

[69] Jakowatz, C. V., D. E. Wahl, P. H. Eichel, D. C. Ghiglia and P. A. Thompson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach* (Springer Science & Business Media, 1996).

[70] Jakowatz Jr, C. V. and N. Doren, "Comparison of polar formatting and backprojection algorithms for spotlight-mode SAR image formation", in "Defense and Security Symposium", pp. 62370H–62370H (International Society for Optics and Photonics, 2006).

[71] Jameson, A., "Energy estimates for nonlinear conservation law with applications to solutions of the Burgers' equation and one-dimensional viscous flow in a shock tube by central difference schemes", in "18th Computational Fluid Dynamics Conference by the AIAA, Miami", vol. 28 (2007).

[72] Kabanikhin, S. I., *Inverse and ill-posed problems: theory and applications*, vol. 55 (Walter De Gruyter, 2011).

[73] Kaipio, J. and E. Somersalo, *Statistical and computational inverse problems*, vol. 160 (Springer Science & Business Media, 2006).

[74] Kelly, S. I., M. Yaghoobi and M. E. Davies, "Auto-focus for under-sampled synthetic aperture radar", Institute for Digital Communications (2012).

[75] Keydel, E. R., S. W. Lee and J. T. Moore, "MSTAR extended operating conditions: A tutorial", in "Aerospace/Defense Sensing and Controls", pp. 228–242 (International Society for Optics and Photonics, 1996).

[76] Koo, V. C., T. S. Lim and H. T. Chuah, "A comparison of autofocus algorithms for SAR imagery", in "Progress in electromagnetics research symposium", vol. 1, pp. 16–19 (2005).

[77] Kort, B. and D. Bertsekas, "Multiplier methods for convex programming", in "IEEE Conference on Decision and Control including the 12th Symposium on Adaptive Processes", vol. 12, pp. 428–432 (IEEE, 1973).

[78] Kosloff, D. and H. Tal-Ezer, "A modified Chebyshev pseudospectral method with an O(N-1) time step restriction", Journal of Computational Physics **104**, 2, 457–469 (1993).

[79] Lavery, J. E., "Solution of steady-state one-dimensional conservation laws by mathematical programming", SIAM Journal on Numerical Analysis **26**, 5, 1081–1089 (1989).

[80] Lavery, J. E., "Solution of steady-state, two-dimensional conservation laws by mathematical programming", SIAM Journal on Numerical Analysis **28**, 1, 141–155 (1991).

[81] Lee, J.-S., "Digital image enhancement and noise filtering by use of local statistics", IEEE Transactions on Pattern Analysis and Machine Intelligence **2**, 165–168 (1980).

[82] Lee, J.-S., "Refined filtering of image noise using local statistics", Computer Graphics and Image Processing **15**, 4, 380–389 (1981).

[83] Lee, J.-S., “A simple speckle smoothing algorithm for synthetic aperture radar images”, IEEE Transactions on Systems, Man, and Cybernetics **1**, 85–89 (1983).

[84] LeVeque, R. J., *Numerical methods for conservation laws* (Springer Science & Business Media, 1992).

[85] LeVeque, R. J., *Finite volume methods for hyperbolic problems*, vol. 31 (Cambridge university press, 2002).

[86] LeVeque, R. J., *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems* (SIAM, 2007).

[87] Leviatan, D. and V. N. Temlyakov, “Simultaneous approximation by greedy algorithms”, Advances in Computational Mathematics **25**, 1, 73–90 (2006).

[88] Li, C., *An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing*, Ph.D. thesis, Rice University (2009).

[89] Liu, J., T.-Z. Huang, Z. Xu and X.-G. Lv, “High-order total variation-based multiplicative noise removal with spatially adapted parameter selection”, JOSA A **30**, 10, 1956–1966 (2013).

[90] Liu, L., M. Esmalifalak, Q. Ding, V. A. Emesih and Z. Han, “Detecting false data injection attacks on power grid by sparse optimization”, IEEE Transactions on Smart Grid **5**, 2, 612–621 (2014).

[91] Liu, Q.-Y., Q. Zhang, F.-F. Gu, Y.-C. Chen, L. Kang and X.-Y. Qu, “Downward-looking linear array 3D SAR imaging based on multiple measurement vectors model and continuous compressive sensing”, Journal of Sensors **2017** (2017).

[92] Liu, Y., J. Ma, Y. Fan and Z. Liang, “Adaptive-weighted total variation minimization for sparse data toward low-dose X-ray computed tomography image reconstruction”, Physics in Medicine and Biology **57**, 23, 7923 (2012).

[93] Lopes, A., E. Nezry, R. Touzi and H. Laur, “Maximum a posteriori speckle filtering and first order texture models in SAR images”, in “Geoscience and Remote Sensing Symposium, 1990. IGARSS’90.’Remote Sensing Science for the Nineties’., 10th Annual International”, pp. 2409–2412 (IEEE, 1990).

[94] Lu, J., L. Shen, C. Xu and Y. Xu, “Multiplicative noise removal in imaging: An exp-model and its fixed-point proximity algorithm”, Applied and Computational Harmonic Analysis **41**, 2, 518–539 (2016).

[95] Lustig, M., D. Donoho and J. M. Pauly, “Sparse MRI: The application of compressed sensing for rapid MR imaging”, Magnetic resonance in medicine **58**, 6, 1182–1195 (2007).

[96] Lynch, P., “The Dolph-Chebyshev window: A simple optimal filter”, Monthly weather review **125**, 4, 655–660 (1997).

[97] Mansourpour, M., M. Rajabi and J. Blais, “Effects and performance of speckle noise reduction filters on active radar and SAR images”, in “Proc. ISPRS”, vol. 36, p. W41 (2006).

[98] Mao, X., “Knowledge-aided two-dimensional autofocus for spotlight SAR polar format imagery”, arXiv preprint arXiv:1503.00207 (2015).

[99] Mersereau, R. M., “Recovering multidimensional signals from their projections”, Computer Graphics and Image Processing **2**, 2, 179–195 (1973).

[100] Mishali, M. and Y. C. Eldar, “Reduce and boost: Recovering arbitrary sets of jointly sparse vectors”, IEEE Transactions on Signal Processing **56**, 10, 4692–4702 (2008).

[101] Molina, D. E., D. Gleich and M. Datcu, “Evaluation of Bayesian despeckling and texture extraction methods based on Gauss–Markov and auto-binomial Gibbs random fields: application to TerraSAR-X data”, IEEE Transactions on Geoscience and Remote Sensing **50**, 5, 2001–2025 (2012).

[102] Monajemi, H., S. Jafarpour, M. Gavish, D. L. Donoho, S. Ambikasaran, S. Bacallado, D. Bharadia, Y. Chen, Y. Choi, M. Chowdhury *et al.*, “Deterministic matrices matching the compressed sensing phase transitions of Gaussian random matrices”, Proceedings of the National Academy of Sciences **110**, 4, 1181–1186 (2013).

[103] Moulin, P., “A wavelet regularization method for diffuse radar-target imaging and speckle-noise reduction”, in “Wavelet Theory and Application”, pp. 123–134 (Springer, 1993).

[104] Munson, D. C., J. D. O’Brien and W. K. Jenkins, “A tomographic formulation of spotlight-mode synthetic aperture radar”, Proceedings of the IEEE **71**, 8, 917–925 (1983).

[105] Natterer, F., *The mathematics of computerized tomography* (SIAM, 2001).

[106] Natterer, F., M. Cheney and B. Borden, “Resolution for radar and X-ray tomography”, Inverse Problems **19**, 6, S55 (2003).

[107] Nguyen, N. and Q. H. Liu, “The regular Fourier matrices and nonuniform fast Fourier transforms”, SIAM Journal on Scientific Computing **21**, 1, 283–293 (1999).

[108] Ni, K.-Y. and S. Rao, “SAR moving target imaging using sparse and low-rank decomposition”, in “SPIE Defense + Security”, pp. 90771D–90771D (International Society for Optics and Photonics, 2014).

[109] Niculescu, C. and L.-E. Persson, *Convex functions and their applications: a contemporary approach* (Springer Science & Business Media, 2006).

[110] Onhon, N. Ö. and M. Çetin, "A sparsity-driven approach for joint SAR imaging and phase error correction", IEEE Transactions on Image Processing **21**, 4, 2075–2088 (2012).

[111] Osher, S., M. Burger, D. Goldfarb, J. Xu and W. Yin, "An iterative regularization method for total variation-based image restoration", Multiscale Modeling & Simulation **4**, 2, 460–489 (2005).

[112] Paulson, C., *Utilizing glint phenomenology to perform classification of civilian vehicles using synthetic aperture radar*, Ph.D. thesis, University of Florida, Gainesville, FL (2013).

[113] Pinel, N. and C. Boulier, *Electromagnetic Wave Scattering from Random Rough Surfaces: Asymptotic Models* (John Wiley & Sons, 2013).

[114] Poggio, T. and V. Torre, "Ill-posed problems and regularization analysis in early vision", Tech. rep., DTIC Document (1984).

[115] Potter, L. C., E. Ertin, J. T. Parker and M. Çetin, "Sparsity and compressed sensing in radar imaging", Proceedings of the IEEE **98**, 6, 1006–1020 (2010).

[116] Potter, L. C., J. W. Nehrbass and K. E. Dungan, "Civilian vehicle data dome overview", `https://www.sdms.afrl.af.mil/index.php?collection=cv_dome`, [Online; accessed 5-June-2017] (2009).

[117] Rigling, B. D. and R. L. Moses, "Taylor expansion of the differential range for monostatic SAR", IEEE Transactions on Aerospace and Electronic systems **41**, 1, 60–64 (2005).

[118] Rudin, L., P.-L. Lions and S. Osher, "Multiplicative denoising and deblurring: theory and algorithms", in "Geometric Level Set Methods in Imaging, Vision, and Graphics", pp. 103–119 (Springer, 2003).

[119] Rudin, L. I., S. Osher and E. Fatemi, "Nonlinear total variation based noise removal algorithms", Physica D: Nonlinear Phenomena **60**, 1-4, 259–268 (1992).

[120] Sanders, T., A. Gelb and R. B. Platte, "Composite SAR imaging using sequential joint sparsity", Journal of Computational Physics **338**, 357–370 (2017).

[121] Sanders, T. and T. Scarnati, "Combination of correlated phase error correction and sparsity models for SAR", in "SPIE Commercial + Scientific Sensing and Imaging", pp. 102220E–102220E (International Society for Optics and Photonics, 2017).

[122] Sandia National Laboratory, "MSTAR overview", `https://www.sdms.afrl.af.mil/index.php?collection=mstar`, [Online; accessed 19-May-2017] (1995).

[123] Scarnati, T., A. Gelb and R. B. Platte, "Using l1 regularization to improve numerical partial differential equation solvers", Journal of Scientific Computing pp. 1–28 (2017).

[124] Scarnati, T., E. Zelnio and C. Paulson, "Exploiting the sparsity of edge information in synthetic aperture radar imagery for speckle reduction", in "Algorithms for Synthetic Aperture Radar Imagery XXIV", vol. 10201, p. 102010C (International Society for Optics and Photonics, 2017).

[125] Schaeffer, H., R. Caflisch, C. D. Hauck and S. Osher, "Sparse dynamics for partial differential equations", Proceedings of the National Academy of Sciences **110**, 17, 6634–6639 (2013).

[126] Scherzer, O. and J. Weickert, "Relations between regularization and diffusion filtering", Journal of Mathematical Imaging and Vision **12**, 1, 43–63 (2000).

[127] Schiavazzi, D., A. Doostan and G. Iaccarino, "Sparse multiresolution stochastic approximation for uncertainty quantification", Recent Advances in Scientific Computing and Applications **586**, 295 (2013).

[128] Schulz, T. J., "Optimal sharpness function for SAR autofocus", IEEE Signal Processing Letters **14**, 1, 27–30 (2007).

[129] Schwertner, M., M. Booth and T. Wilson, "Wavefront sensing based on rotated lateral shearing interferometry", Optics Communications **281**, 2, 210–216 (2008).

[130] Shu, C.-W. and P. S. Wong, "A note on the accuracy of spectral method applied to nonlinear conservation laws", Journal of Scientific Computing **10**, 3, 357–369 (1995).

[131] Singer, A., "Angular synchronization by eigenvectors and semidefinite programming", Applied and Computational Harmonic Analysis **30**, 1, 20–36 (2011).

[132] Singh, A. and S. Dandapat, "Weighted mixed-norm minimization based joint compressed sensing recovery of multi-channel electrocardiogram signals", Computers & Electrical Engineering **53**, 203–218 (2016).

[133] Solbo, S. and T. Eltoft, "A stationary wavelet-domain Wiener filter for correlated speckle", IEEE Transactions on Geoscience and Remote Sensing **46**, 4, 1219–1230 (2008).

[134] Stefan, W., R. A. Renaut and A. Gelb, "Improved total variation-type regularization using higher order edge detectors", SIAM Journal on Imaging Sciences **3**, 2, 232–251 (2010).

[135] Steffens, C., M. Pesavento and M. E. Pfetsch, "A compact formulation for the l21 mixed-norm minimization problem", in "IEEE International Conference on Acoustics, Speech and Signal Processing", pp. 4730–4734 (IEEE, 2017).

[136] Tadmor, E., "Convergence of spectral methods for nonlinear conservation laws", SIAM Journal on Numerical Analysis **26**, 1, 30–44 (1989).

[137] Tadmor, E., "Shock capturing by the spectral viscosity method", Computer Methods in Applied Mechanics and Engineering **80**, 1-3, 197–208 (1990).

[138] Tadmor, E., M. Baines and K. Morton, "Super viscosity and spectral approximations of nonlinear conservation laws", (1993).

[139] Tadmor, E. and K. Waagan, "Adaptive spectral viscosity for hyperbolic conservation laws", SIAM Journal on Scientific Computing **34**, 2, A993–A1009 (2012).

[140] Trefethen, L. N., *Spectral methods in MATLAB* (SIAM, 2000).

[141] Trefethen, L. N. and D. Bau III, *Numerical linear algebra*, vol. 50 (SIAM, 1997).

[142] Tropp, J. A., "Algorithms for simultaneous sparse approximation. Part II: Convex relaxation", Signal Processing **86**, 3, 589–602 (2006).

[143] Tropp, J. A., "Just relax: Convex programming methods for identifying sparse signals in noise", IEEE Transactions on Information Theory **52**, 3, 1030–1051 (2006).

[144] Tropp, J. A., A. C. Gilbert and M. J. Strauss, "Simultaneous sparse approximation via greedy pursuit", in "IEEE International Conference on Acoustics, Speech, and Signal Processing", vol. 5, pp. v–721 (IEEE, 2005).

[145] Tropp, J. A., A. C. Gilbert and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit", Signal Processing **86**, 3, 572–588 (2006).

[146] Ugur, S. and O. Arıkan, "SAR image reconstruction and autofocus by compressed sensing", Digital Signal Processing **22**, 6, 923–932 (2012).

[147] Vogel, C. R., *Computational methods for inverse problems* (SIAM, 2002).

[148] Vogel, C. R. and M. E. Oman, "Iterative methods for total variation denoising", SIAM Journal on Scientific Computing **17**, 1, 227–238 (1996).

[149] Wahl, D., P. Eichel, D. Ghiglia and C. Jakowatz, "Phase gradient autofocus – a robust tool for high resolution SAR phase correction", IEEE Transactions on Aerospace and Electronic Systems **30**, 3, 827–835 (1994).

[150] Walessa, M. and M. Datcu, "Model-based despeckling and information extraction from SAR images", IEEE Transactions on Geoscience and Remote Sensing **38**, 5, 2258–2269 (2000).

[151] Wang, Y., J. Yang, W. Yin and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction", SIAM Journal on Imaging Sciences **1**, 3, 248–272 (2008).

[152] Warner, D. W., D. C. Ghiglia, A. Fitzgerrell and J. Beaver, "Two-dimensional phase gradient autofocus", in "Image Reconstruction from Incomplete Data", vol. 4123, pp. 162–174 (International Society for Optics and Photonics, 2000).

[153] Wasserman, G., R. Archibald and A. Gelb, "Image reconstruction from Fourier data using sparsity of edges", Journal of Scientific Computing **65**, 2, 533–552 (2015).

[154] Weber, C. F., “Analysis and solution of the ill-posed inverse heat conduction problem”, International Journal of Heat and Mass Transfer **24**, 11, 1783–1792 (1981).

[155] Wipf, D. P. and B. D. Rao, “An empirical bayesian strategy for solving the simultaneous sparse approximation problem”, IEEE Transactions on Signal Processing **55**, 7, 3704–3716 (2007).

[156] Wright, S. and J. Nocedal, “Numerical optimization”, Springer Science **35**, 67–68 (1999).

[157] Xie, W., Y. Deng, K. Wang, X. Yang and Q. Luo, “Reweighted l1 regularization for restraining artifacts in FMT reconstruction images with limited measurements”, Optics letters **39**, 14, 4148–4151 (2014).

[158] Yang, J., Y. Zhang and W. Yin, “An efficient TVl1 algorithm for deblurring multichannel images corrupted by impulsive noise”, SIAM Journal on Scientific Computing **31**, 4, 2842–2865 (2009).

[159] Yang, Z. and L. Xie, “Enhancing sparsity and resolution via reweighted atomic norm minimization”, IEEE Transactions on Signal Processing **64**, 4, 995–1006 (2016).

[160] Ye, F., H. Luo, S. Lu and L. Zhang, “Statistical en-route filtering of injected false data in sensor networks”, IEEE Journal on Selected Areas in Communications **23**, 4, 839–850 (2005).

[161] Ye, W., T. S. Yeo and Z. Bao, “Weighted least-squares estimation of phase errors for SAR/ISAR autofocus”, IEEE Transactions on Geoscience and Remote Sensing **37**, 5, 2487–2494 (1999).

[162] Zhang, Y., “User’s guide for YALL1: Your algorithms for l1 optimization”, Technique report pp. 09–17 (2009).

[163] Zhao, B., L. Fei-Fei and E. P. Xing, “Online detection of unusual events in videos via dynamic sparse coding”, in “IEEE Conference on Computer Vision and Pattern Recognition”, pp. 3313–3320 (IEEE, 2011).

[164] Zheng, C., G. Li, Y. Liu and X. Wang, “Subspace weighted l21 minimization for sparse signal recovery”, EURASIP Journal on Advances in Signal Processing **2012**, 1, 98 (2012).

[165] Zhou, F., R. Wu, M. Xing and Z. Bao, “Approach for single channel SAR ground moving target imaging and motion parameter estimation”, IET Radar, Sonar & Navigation **1**, 1, 59–66 (2007).

[166] Zhu, S., A. Mohammad-Djafari, X. Li and J. Mao, “Inverse problems arising in different synthetic aperture radar imaging systems and a general Bayesian approach for them”, in “IS&T/SPIE Electronic Imaging”, pp. 787306–787306 (International Society for Optics and Photonics, 2011).

APPENDIX A

INTRODUCTION TO SYNTHETIC APERTURE RADAR

This Appendix discusses the synthetic aperture radar (SAR) data acquisition procedure for a scene with a single scattering target in the range direction, multiple targets in the range direction and then a two dimensional patch of scatterers spread throughout both the range and cross-range directions. We also describe the data sets made available by the Air Force Research Laboratory (AFRL), as well as some typical SAR image formation procedures. The material and notation presented throughout this appendix has been adapted from [69, 58, 24].

Radar stands for *radio detection and ranging.* The concept of echo-ranging simply states that knowing an echo signal's round trip flight time and its speed of propagation is equivalent to knowing the range from the signal source to the target. In radar systems, the echo-ranging principle is implemented by transmitting high bandwidth pulses and then using pulse compression techniques to discriminate points in a scene based on their distances from the source (in the range direction). However, it is possible for two or more points to be the same distance from a radar antenna but at slightly different angles. Hence, cross-range information is also necessary to visualize a two-dimensional image. This is the standard *aperture* problem.

If we denote the antenna width by D, the distance to the target as R, and the wavelength of the transmitted microwaves as λ, then the resolution distance is given by

$$res = \frac{\lambda R}{D}.$$

Hence given that a typical wavelength is $\lambda \approx 0.3$m, a radar placed 5km from the target would require an antenna 1500m wide to achieve 1m of resolution. As it is impossible to put an antenna of that size aboard an airplane, SAR was developed to achieve the same effect. A SAR system sends out multiple pulses from many observation points and then coherently focuses the received information to obtain a two-dimensional image. In this way, the system *synthesizes* the effect of a large antenna by using multiple observations from a small antenna. Thus the name synthetic aperture radar is appropriate.

There are three common SAR imaging modes: spotlight, stripmap, and scan. During a spotlight mode data collection, the sensor steers its antenna beam so that it continuously illuminates a single ground patch. In the stripmap mode, the antenna remains fixed to be parallel with the flight line, illuminating a ground patch that is usually orthogonal to the sensor direction. In the scan mode, the sensor steers the antenna beam to illuminate a portion of the terrain at any angle to the path of motion. Figure A.1 gives a visual depiction of the three SAR modes. Due to its common use in practice, this thesis focuses on spotlight mode SAR.

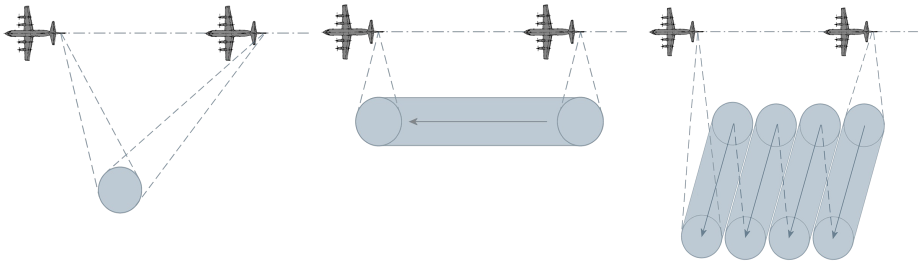

Figure A.1: Left: Spotlight mode. Middle: Stripmap mode. Right: Scan mode.

SAR Data Acquisition

Let us visualize the radar beam looking out of the side of the aircraft, pointing in a direction orthogonal to the flight path. This direction of wave propagation is referred to as the *range* direction. The direction parallel to the flight path is called the *cross-range*, or *azimuth* direction. As the aircraft moves along a flight path, it periodically transmits pulses of microwave energy that hit targets within an illumination patch and then reflect back to the radar, where a demodulation (information extraction) procedure is performed. The data collected in this way are called the *phase history data*. The phase history data are then passed onto a specific image formation processor, which produces, as an output, a reconstruction of the electromagnetic reflectivity field of the illuminated ground patch [69].

Linear Frequency Modulated Chirps and Pulse Compression

The limitations of a SAR system require us to carefully choose and process the appropriate electromagnetic signal to transmit towards a scene. *Pulse compression* allows a radar system to transmit a pulse of relatively long duration and low power to obtain the range resolution and detection performance of a short-pulse, high-power system. A short-pulsed, high-powered system is desirable because the underlying goal of a radar system is to form a high resolution image at a low cost while being undetected by enemies. High resolution images are needed for accurate object detection and classification. To avoid detection, the radar system must have a large stand off range R. Let P_t and P_r be the average power level of the transmitted and received signals, respectively. The fundamental radar equation is then

$$P_r = \frac{P_t}{R^4} C \implies R = \left(\frac{P_t}{P_r} C \right)^{1/4}, \tag{A.1}$$

where C represents a collection of constants relating to the radar system and the scene being imaged. From (A.1), we see that to have a large standoff range and a large returned power, the transmit power needs to be large.

A *high return power* also yields improved quality of the returned signal. We can see this by considering in a very general sense,

$$Energy = \int Power. \tag{A.2}$$

Let σ be the standard deviation of the noise in the returned signal, and E_r be the energy in the returned signal. The signal-to-noise ratio of the returned signal is then

$$SNR = \frac{E_r}{\sigma}. \tag{A.3}$$

Thus to have a high SNR in (A.3) and improved returned signal quality in (A.1), a large amount of energy is required returned to the radar and thus high returned power, according to (A.2).

A *short pulse* is desirable because if T_e is effective pulse length, $B = 1/T_e$ is the bandwidth of the transmitted waveform, and then range resolution is given as

$$\rho_x = \frac{c}{2B} = \frac{cT_e}{2}, \tag{A.4}$$

where $c \approx 3 \times 10^8$m/s is the speed of light. (A full discussion on resolution is provided later in this Appendix.) From (A.4) we see that short duration transmitted pulses yield improved range resolution. However, there is a trade off. The average power of a the signal is given as

$$P_e = PT_e, \tag{A.5}$$

where P is the peak power, implying that short duration pulses result in diminished transmit power and thus lower received power. Pulse compression is what allows for long duration pulses to be transmitted at a large standoff range yet achieve high resolution images. In the end, a radar system that incorporates pulse compression processing rather than a simple pulse system to achieve high range resolution provides the following advantages:

1. Improved detection performance.

2. Mutual interference reduction.

3. Increased system operational flexibility.

To perform pulse compression, a system must transmit a signal that is either frequency or phase modulated. SAR systems often transmit *linear frequency modulated*

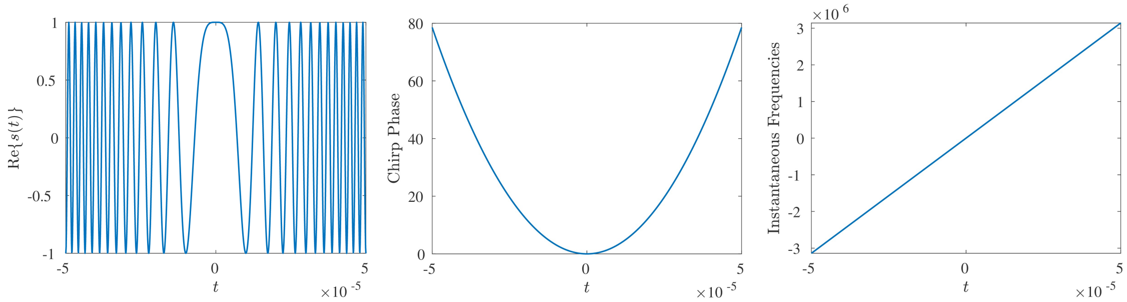

Figure A.2: (left) A linear FM chirp signal described by (A.6) with $\omega_c = 0$Hz, $T_t = 1 \times 10^{-4}$s and $\alpha = 3.14 \times 10^{10}$Hz. (middle) The corresponding chirp phase and (right) instantaneous frequencies.

(FM) chirps described by $\text{Re}\{s(t)\}$, where $s : \mathbb{R} \to \mathbb{C}$ is defined as

$$s(t) = \begin{cases} e^{i(\omega_0 t + \alpha t^2)}, & |t| \le \frac{T_t}{2} \\ 0, & \text{else} \end{cases}. \tag{A.6}$$

Here, $\omega_0 \in \mathbb{R}$ is the *carrier frequency* of the chirp, $T_t \in \mathbb{R}$ is the transmitted chirp duration, and $2\alpha \in \mathbb{R}$ is the chirp rate. The time variable $t \in \mathbb{R}$ is referred to as *fast time* because it represents the time along each pulse. During this time interval, the SAR imaging platform is assumed to be stationary, preventing the use of fast time to actually clock the motion of the SAR system. A *slow time* variable, on the other hand, allows for motion detection of the SAR platform and will be discussed later.

Frequency is interpreted as the first derivative of phase. The phase of the chirp signal is given by $\omega_0 t + \alpha t^2$, and thus the instantaneous SAR frequencies encoded by the chirp signal are

$$\omega(t) = \frac{1}{2\pi}(\omega_0 + 2\alpha t),$$

where we represent $\omega(t)$ as ω. Note that the frequencies are increasing linearly with time. A depiction of a chirp and its corresponding phase and instantaneous frequencies can be seen in Figure A.2.

The SAR central frequency is given by

$$\omega_c = \frac{\omega_0}{2\pi}, \tag{A.7}$$

with corresponding wavelength

$$\lambda = \frac{2\pi c}{\omega_0} = \frac{c}{\omega_c}, \tag{A.8}$$

where once again c is the speed of light. Because $|t| \leq T_t/2$, the chirp frequencies range from

$$\omega_0 - \alpha T_t \leq 2\pi\omega \leq \omega_0 + \alpha T_t, \tag{A.9}$$

yielding the chirp bandwidth (measured in Hertz)

$$B_\omega = \frac{|\max_t \omega - \min_t \omega|}{2\pi} = \frac{\alpha T_t}{\pi}. \tag{A.10}$$

The effective pulse length is given by

$$T_e := \frac{1}{B_\omega} = \frac{\pi}{\alpha T_t}. \tag{A.11}$$

The compression ratio CR (time-bandwidth product) gives the ratio of the average power of the transmitted compressed signal P_t to the average power of the typical waveform P_e, assuming both signals have the same peak power P. Using (A.5) and (A.11) yields a compression ratio of

$$CR = \frac{P_t}{P_e} = \frac{PT_t}{PT_e} = \frac{T_t}{T_e} = T_t B_\omega = \frac{\alpha T_t^2}{\pi}. \tag{A.12}$$

Thus the average power of the received signal is amplified by a factor of $\alpha T_t^2/\pi$ when the chirp signal is transmitted.

A depiction of a typical chirp FM signal can be seen in Figure A.3(left). For comparison, a continuous waveform (CW) simple pulse is shown in Figure A.3(right). It is given by

$$s(t) = \begin{cases} \cos(\omega_0 t), & |t| \leq \frac{T_c}{2} \\ 0, & \text{else.} \end{cases} \tag{A.13}$$

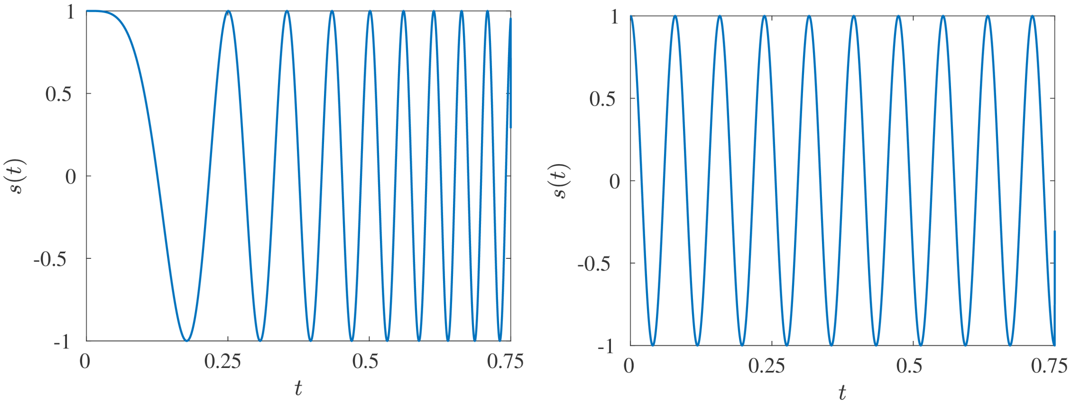

Figure A.3: (left) Depiction of a linear FM chirp signal, (A.6), and (right) continuous burst waveform, (A.13), with signal durations of $T_t = T_c = 0.75$sec, chirp rate of $\alpha = 100$Hz and center frequency $\omega_c = 500$Hz.

Figure A.4(left) compares the bandwidth (width of main lobe) of the discrete Fourier transform (in dB) given by

$$dB(|\mathcal{F}s(t)|) = 20\log_{10}\frac{|\mathcal{F}s(t)|}{\max_t |\mathcal{F}s(t)|}$$

of the chirp (A.6) and CW (A.13) waveforms each having a pulse duration $T_t = T_c =$ 0.75sec. Observe the larger bandwidth corresponding to the chirp signal, implying improved range resolution. Figure A.4(right) compares the spectrum of the same signals, only the CW is given a pulse duration of $T_c = 0.03$sec. This shows that to achieve the bandwidth of the chirp using the CW pulse, the pulse duration of the CW must be reduced by more than a factor of 10. Thus, according to (A.5), the corresponding power of the returned signal using the CW signal would also be reduced by more than a factor of 10. This is consistent with the idea that by using the frequency modulated chirp, we can increase bandwidth while decreasing power, and hence achieve the desirable resolution properties of a continuous pulse waveform without high energy costs. In Figure A.4, empirical evidence has shown that a maximum side lobe level of -35dB is ideal for approximation.

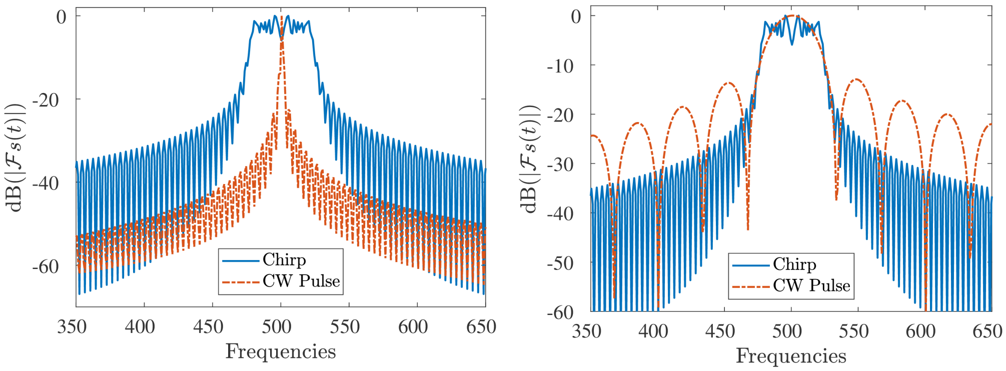

Figure A.4: dB($|\mathcal{F}s(t)|$) where $\mathcal{F}s$ is the (discrete) Fourier transform of the chirp, given in (A.6), (solid-blue), and continuous burst waveform, given in (A.13), (dashed-orange). Here $\alpha = 100Hz$ and $\omega_c = 500Hz$ with (left) both signals having pulse duration of $T_t = T_c = 0.75sec$ and (right) the chirp signal has pulse duration $T_t = 0.75sec$ and the CW signal has pulse duration $T_c = 0.03sec$.

We now consider three possible scenarios, each of which builds on the previous one, in order to develop a SAR data model that accurately describes realistic scattering phenomenology.

Model 1: Single Scatterer in Range Direction

Assume there is a single scatterer located in the range direction and 0° azimuth change, as depicted in Figure A.5. At a given *fast time* $t \in \mathbb{R}$, we assume the radar system transmits the chirp signal (A.6). The returned signal $r : \mathbb{R} \to \mathbb{C}$ is then a one-dimensional, scaled and delayed version of the transmitted signal in (A.6):

$$r(t) = A\text{Re}\left\{f(u)\exp\left[i\left(\omega_0(t-\tau_0-\tau)+\alpha(t-\tau_0-\tau)^2\right)\right]\right\}, \tag{A.14}$$

where the scene reflectivity f is only a function of the *slant range* u, and the scale factor $A \in \mathbb{R}$ accounts for propagation attenuation. In Figure A.5, the x-axis reflects

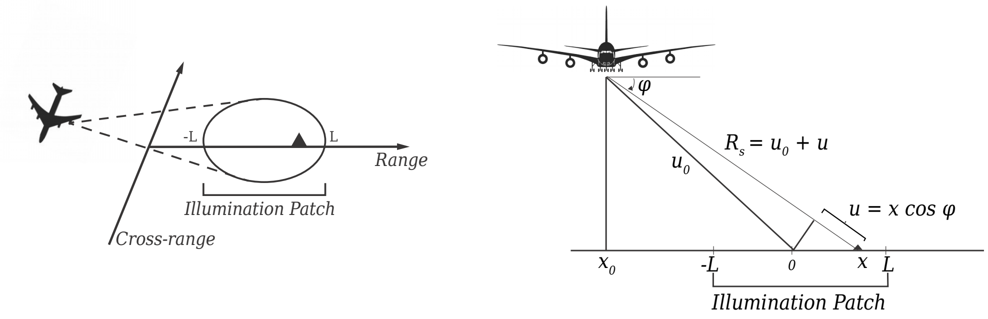

Figure A.5: Geometry of SAR imaging system with one scatterer in the scene (adapted from [69]).

the *ground range*, with the center of the illuminated patch located at $x = 0$ and offset from the plane's position by distance x_0. The distance from the plane to the illumination patch center is u_0, while the distance to the scatterer located at position x is given approximately by $R_s = u_0 + u$. Defining the *depression angle* (elevation angle) by φ, the relationship between the ground range and slant range is given by

$$u = x \cos \varphi. \tag{A.15}$$

Because the radar is assumed to be parallel to the ground, the elevation and depression angles are congruent and can be used interchangeably [112]. It is also assumed that the plane is a significant distance away from the target scatterer ($x << u_0$).

The returned signal (A.14) is dependent on the scene geometry and contains information about the microwave reflectivity function we seek to recover, $f : \mathbb{R} \times \mathbb{R} \to \mathbb{C}$, given by

$$f(x, y) = |f(x, y)|e^{i\Phi}.$$

The magnitude of reflectivity $|f(x, y)|$ determines the amount of the incident energy that is reflected back as the return signal and $\Phi \in \mathbb{R}^2$ describes the phase change of the waveform upon reception. In general, Φ (and therefore f) is determined by the electrical and physical properties of the target material at the center frequency.

The quantity $\frac{2R_s}{c} = \tau_0 - \tau(u)$ is the round trip propagation time (signal delay), where

$$\tau(u) = \frac{2u}{c} =: \tau, \quad \tau_0 = \frac{2u_0}{c}, \tag{A.16}$$

and the dependence of τ on u is dropped for simplicity. Here τ_0 is the propagation time to the center of the scene, and is assumed to be explicitly known. In reality, τ_0 can only be estimated, which causes a phase error that is observed as smearing in the reconstruction. Techniques to overcome such imperfect propagation attenuation measurements, including *autofocusing* or *phase error correction*, are discussed in Chapter 5.

Model 2: Continuum of Scatterers in Range Direction

For this model, we assume that there is continuum of scatterers located in the range direction, as shown in Figure A.6. The returned, one-dimensional signal is now given by the superposition of scaled and delayed transmitted signals that have reflected back from targets lying at slant ranges constrained to $-u_1 \leq u \leq u_1$,

$$r(t) = A \int_{-u_1}^{u_1} \mathrm{Re}\left\{ f(u) \exp\left[i \left(\omega_0 \left(t - \tau_0 - \tau\right) + \alpha \left(t - \tau_0 - \tau\right)^2 \right) \right] \right\} du, \tag{A.17}$$

where (A.15) yields $u_1 = L\cos\varphi$. Define the patch propagation time $\tau_p \in \mathbb{R}$ as the two way propagation delay between the target at the near edge and the target at the far edge of the radar illumination path, which from (A.16) is given by

$$\tau_p = 2\tau(u_1) = 2\left(\frac{2u_1}{c}\right) = 4\frac{L\cos\varphi}{c}. \tag{A.18}$$

The returned signal as defined in (A.17) therefore is supported on

$$\tau_0 + \frac{\tau_p}{2} - \frac{T_t}{2} \leq t \leq \tau_0 - \frac{\tau_p}{2} + \frac{T_t}{2}, \tag{A.19}$$

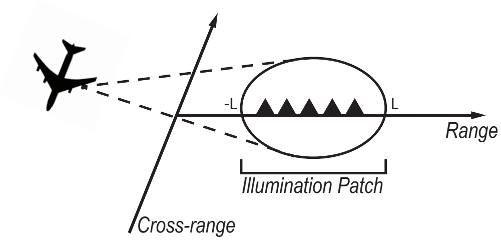

Figure A.6: Depiction of a continuum of scatterers in the range direction.

where τ_p and T_t are given in (A.18) and (A.6) respectively.

The returned signal (A.17) can be written equivalently as

$$r(t) = A \int_{-u_1}^{u_1} \mathrm{Re}\left\{ f(u) s(t - \tau - \tau_0) \right\} du,$$

so that

$$r(t + \tau_0) = A \int_{-u_1}^{u_1} \mathrm{Re}\left\{ f(u) s(t - \tau) \right\} du.$$

Finally substituting $\tau_1 = 2u_1/c$, $du = d\tau c/2$ and $A_1 = Ac/2$ yields

$$r(t + \tau_0) = A_1 \int_{-\tau_1}^{\tau_1} \mathrm{Re}\left\{ f(\tau) s(t - \tau) \right\} d\tau. \qquad \text{(A.20)}$$

It is apparent that to obtain an estimate for f we must deconvolve s from r. In SAR, this is accomplished via *quadrature demodulation.* The demodulation process, inherent in the radar system hardware, is mathematically described below.

Quadrature Demodulation in SAR

Under the assumption that $T_t >> \tau_p$ and that τ_0 is known exactly, we can *extract* approximate instantaneous frequency information (i.e. the classical Fourier transform of f) from the chirp response, (A.17), by implementing a *deramping* process. This process requires the following steps:

1. Demodulation of $r(t)$ by multiplication with *in-phase* (real) and *quadrature*

(imaginary) versions of the transmitted, complex sinusoid

$$\begin{aligned} r_I(t) &= \cos\left[\omega(t-\tau_0)+\alpha(t-\tau_0)^2\right] \\ r_Q(t) &= \sin\left[\omega(t-\tau_0)+\alpha(t-\tau_0)^2\right] \end{aligned} \tag{A.21}$$

to obtain $r_d(t) = r(t)\left(r_I(t)+ir_Q(t)\right)$. Using the appropriate trigonometric identities, one can show that

$$\begin{aligned} r_d(t) = \frac{1}{2}\int_{-u_1}^{u_1} & f(u)\mathrm{exp}\{i(\omega(2t-\tau-2\tau_0)+\alpha((t-\tau_0)^2+(t-\tau-\tau_0)^2))\}\,du \\ +\frac{1}{2}\int_{-u_1}^{u_1} & f(u)\mathrm{exp}\{i(\alpha\tau^2-\tau(\omega+2\alpha(t-\tau_0)))\}\,du. \end{aligned} \tag{A.22}$$

2. Low pass filtering of the result from step 1 to remove the first term and substituting (A.16) into the result to obtain

$$r_d(t) \approx \frac{1}{2}\int_{-u_1}^{u_1} f(u)\exp\left\{-i\frac{2u}{c}\left[\omega_0+2\alpha(t-\tau_0)\right]\right\}\exp\left\{i\alpha\frac{4u^2}{c^2}\right\}du.$$

From here, if we suppose that the chirp rate α and the scene size u_1 are sufficiently small, then $\frac{4\alpha u^2}{c^2} \approx 0$, so that by (A.16), $e^{i\alpha\tau^2} \approx 1$ and

$$\hat{f}(\Omega(t)) = \int_{-u_1}^{u_1} f(u)e^{-i\Omega(t)u}\,du \approx r_d(t), \tag{A.23}$$

where

$$\Omega(t) := \frac{2}{c}(\omega_0+2\alpha(t-\tau_0)) \tag{A.24}$$

are the spatial frequencies. In other words, the demodulation in (A.22) approximately yields the Fourier coefficients of f evaluated over a specific limited range of frequencies determined by the time-support of the return signal.

To further characterize (A.23), note that (A.19) implies that the range of $\Omega(t)$ is given by

$$\frac{2}{c}\left(\omega_0-\alpha(T_t-\tau_p)\right) \leq \Omega(t) \leq \frac{2}{c}\left(\omega_0+\alpha(T_t-\tau_p)\right).$$

Moreover, under the assumption that $T_t >> \tau_p$, we have

$$\frac{2}{c}(\omega_0 - \pi B_\omega) \leq \Omega(t) \leq \frac{2}{c}(\omega_0 + \pi B_\omega), \tag{A.25}$$

where B_ω given in (A.10). Hence the offset of spatial frequencies transduced by the chirp waveform is

$$\Delta\Omega(t) := \frac{2}{c}(\omega_0 + \pi B_\omega) - \frac{2}{c}(\omega_0 - \pi B_\omega) = \frac{4\pi B_\omega}{c}. \tag{A.26}$$

Remark A.0.1 *Observe that using the linear FM chirp and the described quadrature demodulation results in an intermediate signal that measures a portion of the Fourier spectrum of the reflectivity. In this sense, as a result of the specific transmitted waveform and corresponding deconvolution process, the range has been converted to* spatial *frequency, and the data can be thought of as spatial frequency data. This is not necessarily the case when continuous wave bursts are transmitted, however. Indeed, for continuous wave bursts, the Fourier domain data never appear as an intermediate step of the processing, but instead the output of the demodulation filter directly estimates the reflectivity.*

Range Resolution

Essential to this discussion is *range resolution*, which is defined as the minimal distance two targets must be separated in order to be discernible from one another in the range direction. The *range resolution*, ρ_x, where x represents the range dimension in Figure A.5, can be determined by conducting an impulse response analysis. More discussion can be found in [69].

To perform the impulse response analysis, assume $f(u) = \delta(u)$, the Dirac delta function arbitrarily placed at $u = 0$. We can only recover a band limited signal where the Fourier transform is zero for $|\Omega(t)| > \Delta\Omega/2$ and $\Delta\Omega$ is given in (A.26).

Specifically, we have

$$\hat{f}(\Omega(t)) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\delta(u)e^{-iu\Omega(t)}du = \begin{cases} \frac{\Delta\Omega}{2\pi}, & |\Omega(t)| \leq \Delta\Omega/2 \\ 0, & \text{else,} \end{cases} \tag{A.27}$$

yielding the band limited signal

$$\int_{-\Delta\Omega/2}^{\Delta\Omega/2}\frac{\Delta\Omega}{2\pi}e^{iu\Omega(t)}d\Omega(t) = \frac{\Delta\Omega}{2\pi}\frac{\sin(u\Delta\Omega/2)}{u\Delta\Omega/2} = \frac{\Delta\Omega}{2\pi}\text{sinc}\left(\frac{u\Delta\Omega}{2}\right) \approx f(u). \tag{A.28}$$

The first zero crossing occurs when sinc $\left(\frac{u\Delta\Omega}{2}\right) = 0$ yieding $u = \frac{2\pi}{\Delta\Omega}$. This means that the *minimal separation* for which two targets are distinguishable occurs when the *range resolution* is given by

$$\rho_x = \frac{2\pi}{\Delta\Omega} = \frac{c}{2B_\omega}. \tag{A.29}$$

where we have used (A.26). Observe that the slant range resolution ρ_u is inversely proportional to the radar bandwidth and is not a function of the center frequency.

Model 3: Continuum of Scatterers in Range and Cross Range Directions

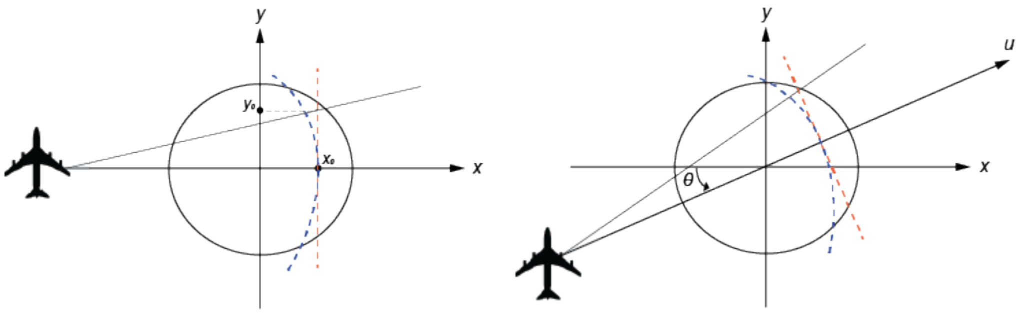

Figure A.7: Illustration of the approximation of an arc, which contains points equidistant from the radar at observation angle 0 (left) and θ (right), by a line [104].

Consider multiple scatterers located equidistant from the radar in the azimuth direction as depicted in Figures A.7 and A.9, where x and y correspondingly represent the range and cross-range directions. We now define R as the distance from the radar to the center of the scene and L as the radius of the circular ground patch. As shown in Figure A.7, the points in the ground patch equidistant from the radar lie on the curved blue arc. Because $R >> L$ for a typical radar system, we make the assumption that this arc is nearly a straight line, seen in red. This is known as the *far field assumption.* Below we describe two main sources of error caused by this far field assumption, and the conditions that guarantee that the errors due to the far field assumption are negligible with respect to the other model errors.

1. Let us define $\bar{\rho}$ as a two-dimensional (resolution) cell where the lengths in the x and y directions are respectively ρ_x and ρ_y. Here ρ_y refers to the cross-range resolution. To ensure the *range error*, that is the error in the x direction due to wavefront curvature over the target field, is negligible, we must demonstrate that it is at least as small as $\bar{\rho}$. Using Figure A.7(left), we define $E_{x,0}$ as the range error at zero azimuth angle. Because $R >> |x_0|$ and $R >> |y_0|$ at the point (x_0, y_0), we have

$$\begin{aligned} E_{x,0} &= \left[(R+x_0)^2 + y_0^2\right]^{1/2} - (R+x_0) = (R+x_0)\left[\left(1+\frac{y_0^2}{(R+x_0)^2}\right)^{1/2} - 1\right] \\ &\approx (R+x_0)\left[1+\frac{y_0^2}{2(R+x_0)^2} - 1\right] = (R+x_0)\left[\frac{y_0^2}{2(R+x_0)^2}\right] \approx \frac{Ry_0^2}{2R^2} = \frac{y_0^2}{2R}. \end{aligned}$$

where we have employed the first-order Taylor expansion $\sqrt{1+z} = 1+\frac{z}{2}$ around $z = 0$ in the approximation. Because the maximum of y_0 within the target patch occurs at $y_0 = L$, we have

$$E_{x,0} \approx \frac{L^2}{2R},$$

leading to our first condition

$$\boxed{\frac{L^2}{2R} < \rho_x.}$$

2. The other main issue stems from how the SAR system coherently combines projections from many different angles. To preserve coherency (i.e. the constant phase change), we require that the range error due to wavefront curvature at a particular point must vary by no more than a small fraction of a wavelength, (A.8), through the full range of look angles. Figure A.8 shows the wavefront curvature at the point (x_0, y_0) due to look angles 0 and θ. Observe that

$$\cos\theta = \frac{x_0}{\ell_1}, \qquad \sin\theta = \frac{\ell_2}{\ell_1}, \qquad \sin\theta = \frac{\ell_4}{\ell_3}, \qquad \cos\theta = \frac{\ell_5}{\ell_3}.$$

Therefore,

$$\begin{aligned}
\ell_1 &= \frac{x_0}{\cos\theta} \\
\ell_2 &= \ell_1 \sin\theta = x_0 \tan\theta \\
\ell_3 &= y_0 - \ell_2 = y_0 - \ell_1 \sin\theta = y_0 - x_0 \tan\theta \\
\ell_4 &= \ell_3 \sin\theta = y_0 \sin\theta - x_0 \frac{\sin^2\theta}{\cos\theta} \\
\ell_5 &= \ell_3 \cos\theta = y_0 \cos\theta - x_0 \sin\theta.
\end{aligned}$$

Analogous to the derivation for $E_{x,0}$, the range error at point (x_0, y_0) for a projection at angle θ is given by

$$\begin{aligned}
E_{x,\theta} &= \left[(R + \ell_1 + \ell_4)^2 + \ell_5^2\right]^{1/2} - (R + \ell_1 + \ell_4) \\
&= (R + \ell_1 + \ell_4)\left[\left(1 + \frac{\ell_5^2}{(R + \ell_1 + \ell_4)^2}\right)^{1/2} - 1\right] \\
&\approx (R + \ell_1 + \ell_4)\left[1 + \frac{\ell_5^2}{2(R + \ell_1 + \ell_4)^2} - 1\right] = \frac{\ell_5^2}{2(R + \ell_1 + \ell_4)} \approx \frac{\ell_5^2}{2R},
\end{aligned}$$

where we have used the fact that $R >> \ell_1 + \ell_4$.

The above two conditions allow us to calculate the difference D in wavefront

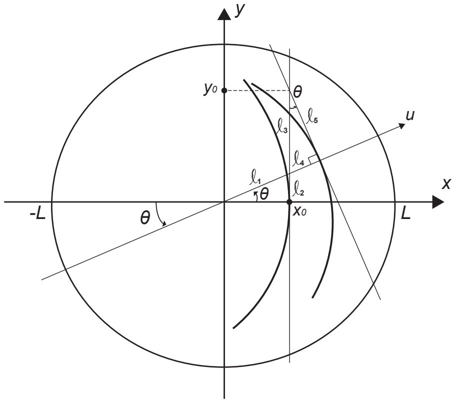

Figure A.8: Wavefront curvature for projection angles 0° and θ° (adapted from [104]).

curvature from projection angle 0° to projection angle θ° as

$$\begin{aligned} D &= E_{x,0} - E_{x,\theta} \\ &= \frac{1}{2R}\left[y_0^2 - \ell_5^2\right] \\ &= \frac{1}{2R}\left[y_0^2 - (y_0\cos\theta + x_0\sin\theta)^2\right] \\ &= \frac{1}{2R}\left[y_0^2 - y_0^2\cos^2\theta + 2x_0y_0\sin\theta\cos\theta - x_0^2\sin^2\theta\right] \\ &= \frac{1}{2R}\left[y_0^2(1-\cos^2\theta) + 2x_0y_0\sin\theta\cos\theta - x_0^2\sin^2\theta\right] \\ &= \frac{1}{2R}\left[y_0^2\sin^2\theta + 2x_0y_0\sin\theta\cos\theta - x_0^2\sin^2\theta\right] \\ &= \frac{1}{2R}\left[(y_0^2 - x_0^2)\sin^2\theta + 2x_0y_0\sin\theta\cos\theta\right]. \end{aligned}$$

WLOG, given the constraints $\theta \leq \theta_{\text{max}} \leq \frac{\pi}{4}$ and $x_0^2 + y_0^2 \leq L^2$, two cases which maximize deviations in D are

- **Case 1.** Consider $x_0 = 0$ and $y_0 = L$. Then
$$D = \frac{1}{2R}[L^2\sin^2\theta] \leq \frac{L^2\sin^2\theta_{\text{max}}}{2R} =: D_1^{\text{max}}.$$
- **Case 2.** Consider $x_0 = y_0 = L/\sqrt{2}$. Then
$$D = \frac{1}{2R}[L^2\sin\theta\cos\theta] = \frac{L^2\sin 2\theta}{4R} \leq \frac{L^2\sin 2\theta_{\text{max}}}{4R} =: D_2^{\text{max}}.$$

Because $\theta_{\max} \leq \pi/4$, we have $D_1^{\max} \leq D_2^{\max}$. Therefore to preserve coherency, $D_2^{\max}$ must be smaller than a fraction of the wavelength, say $\lambda/8$. Our second condition is therefore

$$\boxed{\frac{L^2 \sin 2\theta_{\max}}{4R} << \frac{\lambda}{8} \implies \frac{L^2 \sin 2\theta_{\max}}{R} << \frac{\lambda}{2}.}$$

In practice, these two conditions are assumed to always hold. The analysis above helps to provide insight into the errors that are committed when making this far field approximation.

When a continuum of scatterers located equidistant from the radar in the azimuth direction, the return from such scatterers will be received by the radar at precisely the same time, and thus they will not be distinguishable from one another. However, if we send pulses from multiple azimuth angles, we will be able to gather enough information about the two-dimensional reflectivity function $f : \Gamma \to \mathbb{C}$ of the scene to distinguish the scatterers from one another and reconstruct the two-dimensional image. Here, f is defined over the *illumination patch* defined as

$$\Gamma := \left\{(x, y) \in \mathbb{R}^2 | x^2 + y^2 \leq L^2\right\}.$$

The combined return of a set of scatterers equidistant from the radar is again the superposition of the returns that would be received from each individual scatterer. This leads us to tomographic formulation of SAR phase history data. Similar techniques have been utilized in medical imaging and electron microscopy [105]. Let $p : \mathbb{R} \times \mathbb{R} \to \mathbb{C}$ be a line integral, or projection, defined by $p(\theta, u)$ at distance $R + u$ from the radar at observation angle $\theta \in \mathbb{R}$, which is considered to be a counterclockwise rotation from the x axis (illustrated in Figure A.9). We now define a new coordinate system (u, v) given by

$$x = u \cos\theta - v \sin\theta, \quad y = u \sin\theta + v \cos\theta. \tag{A.30}$$

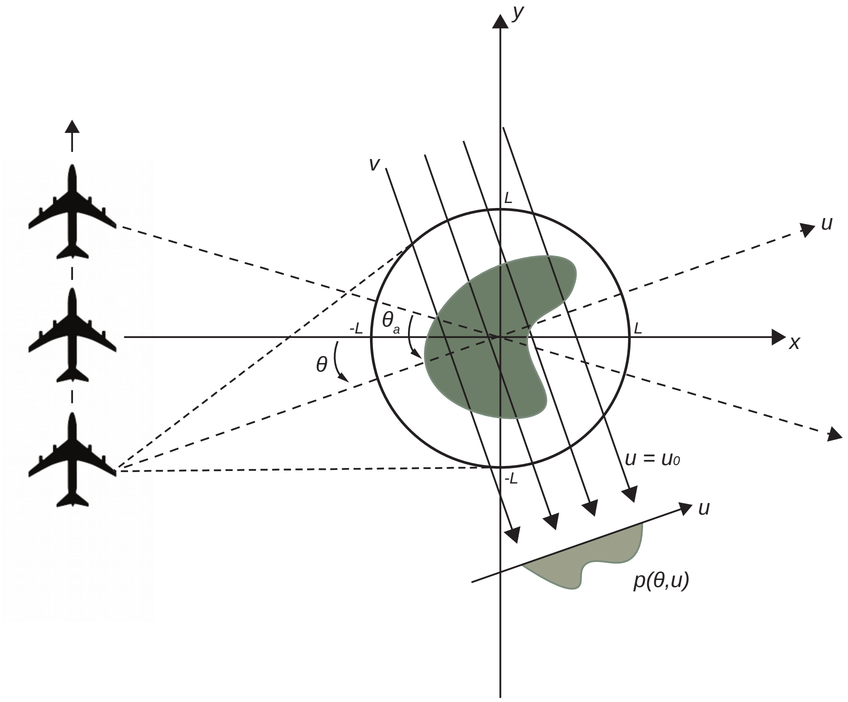

Figure A.9: Ground-plane geometry for a data collection in two-dimensional spotlight-mode SAR.

The inverse transformation is then

$$u = x\cos\theta + y\sin\theta, \qquad v = -x\sin\theta + y\cos\theta. \tag{A.31}$$

Note that u and v are the new range and cross-range directions at viewing angle θ, respectively. The relationship between the projection $p(\theta, u)$ and the reflectivity field $f(x, y)$ is given by the one-dimensional line integral through the illumination patch at angle θ:

$$\begin{aligned} p(\theta, u) &= \int_{-L}^{L} f(u\cos\theta - v\sin\theta, u\sin\theta + v\cos\theta)dv \\ &= \iint\limits_{\Gamma} \delta(u - x\cos\theta - y\sin\theta) f(x, y) dx dy, \end{aligned} \tag{A.32}$$

which is known as the Radon transform of f at angle θ, [48].

The two-dimensional return $r : \mathbb{R} \times \mathbb{R} \to \mathbb{C}$ from all targets that lie along a constant range line $u = u_0$ as in Figure A.9 is given by

$$r(\theta, t) = \mathrm{Re}\left\{p(\theta, u_0)s\left(t - \frac{2(R + u_0)}{c}\right)\right\}.$$

Thus, analogous to (A.17), the entire returned signal is the superposition of returns from all constant lines,

$$r(\theta, t) = \int_{-L}^{L} \mathrm{Re}\left\{p(\theta, u)s\left(t - \frac{2(R + u)}{c}\right)\right\} du, \tag{A.33}$$

on the time interval of the transmitted chirp given in (A.6),

$$-\frac{T_t}{2} + \frac{2(R + L)}{c} \leq t \leq \frac{T_t}{2} + \frac{2(R + L)}{c}, \tag{A.34}$$

Substituting the transmitted signal (A.6) into (A.33), and performing the same quadrature demodulation proceedure in (A.22) yields

$$\begin{aligned} r_d(\theta, t) = \frac{1}{2}\int_{-L}^{L} & p(\theta, u)\mathrm{exp}\{i(\omega(2t - \tau(u) - 2\tau_0) + \alpha((t - \tau_0)^2 + (t - \tau(u) - \tau_0)^2))\}\, du \\ + \frac{1}{2}\int_{-L}^{L} & p(\theta, u)\mathrm{exp}\{i(\alpha\tau^2(u) - \tau(u)(\omega + 2\alpha(t - \tau_0)))\}\, du, \end{aligned} \tag{A.35}$$

where τ and τ_0 are the return propagation times given in (A.16). By low pass filtering (A.35), and again assuming the chirp rate α and scene radius L are sufficiently small, we arrive at

$$\begin{aligned} \hat{f}_\theta(t) &= \int_{-L}^{L} p(\theta, u)\exp\left[-i\frac{2u}{c}(\omega_0 + 2\alpha(t - \tau_0))\right] du \\ &= \int_{-L}^{L} p(\theta, u)e^{-iu\Omega(t)}du \approx r_d(\theta, t), \end{aligned} \tag{A.36}$$

where, as in (A.24), the spatial frequencies are defined as

$$\Omega(t) = \frac{2}{c}\left[\omega_0 + 2\alpha(t - \tau_0)\right] \tag{A.37}$$

on the interval

$$\frac{2}{c}(\omega_0 - \alpha T_t) \leq \Omega(t) \leq \frac{2}{c}(\omega_0 + \alpha T_t). \tag{A.38}$$

From (A.36) we see that the radon transform $p(\theta, u)$ can equivalently be given as

$$p(\theta, u) = \int_{-\infty}^{\infty} \hat{f}_\theta(t) e^{iu\Omega(t)} dt. \tag{A.39}$$

Substituting (A.32) into (A.36) yields the relationship between the returned signal and the reflectivity field

$$\begin{aligned} \hat{f}_\theta(t) &= \int\limits_{|u| \leq L} \iint\limits_{\Gamma} \delta(u - x\cos\theta - y\sin\theta) f(x, y) e^{-iu\Omega(t)} dx dy du \\ &= \iint\limits_{\Gamma} f(x, y) e^{-i\Omega(t)(x\cos\theta + y\sin\theta)} dx dy. \end{aligned} \tag{A.40}$$

This is in fact the *Fourier (Projection) Slice Theorem* [99], which states that if the Fourier transform of the reflectivity function to be recovered exists, then the one-dimensional Fourier transform of the projection at angle θ is a slice of the two-dimensional Fourier transform of the reflectivity field taken at the same angle θ. The data $\hat{f}_\theta(t)$ collected from all observation angles are called the *phase history data*.

Range and Cross-Range Resolution for Two-Dimensional SAR

We now discuss the range and cross-range resolutions of the two-dimensional SAR system to determine the minimal distance two targets must be separated in order to be distinguished from one another. As seen in Figure A.10(left), processing returns across a range of azimuth angles, θ_a, provides samples on a polar grid in an annulus segment away from the origin. For comparison, the standard Cartesian grid is shown in Figure A.10(right). Note that the FFT does not directly apply to polar coordinates, yielding additional complexity in processing the data. Indeed several algorithms [24, 120, 69, 117, 49] have been developed to ensure the efficiency and accuracy for processing the polar coordinate Fourier data. This will be discussed further in Section A.

We can also relate the achievable resolution of SAR images to the dimensions of this annulus. Assume that the dimensions of the annulus can be approximated by a rectangle of width $\Delta\Omega_y$ and height $\Delta\Omega_x$. We then perform another impulse response function analysis, analogous to (A.27), by considering a point reflector in the scene and computing its Fourier transform limited to this rectangular region. As in the single scatterer case in (A.28), the inverse Fourier transform yields a (two-dimensional) sinc function. The wider the support of the rectangular region, the narrower the main lobe of this sinc function, resulting in better resolution. Once again, the *minimal separation* for which two targets are distinguishable occurs at the first zero crossing away from the origin, which is determined as

$$\bar{\rho} = \left(\frac{2\pi}{\Delta\Omega_x}, \frac{2\pi}{\Delta\Omega_y} \right).$$

Thus, two point scatterers are distinguishable only if they are separated by more than

$$\rho_x = \frac{2\pi}{\Delta\Omega_x}, \quad \rho_y = \frac{2\pi}{\Delta\Omega_y} \tag{A.41}$$

in the range direction and cross-range directions, respectively.

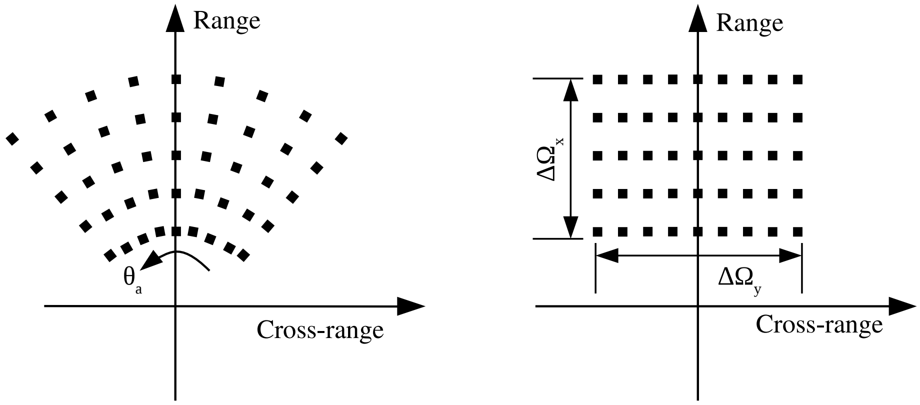

Figure A.10: Graphical representation (left) of the annulus segment containing known samples of the Fourier transform of the reflectivity density and (right) an approximation of these samples on a rectangular grid.

Assuming the width of the rectangle is equal to the radial width of the annulus, which is essentially the spatial frequency bandwidth of each return, the maximum width can be determined by substituting the maximum temporal interval (A.34) into the definition of $\Omega(t)$ given in (A.37), thereby yielding the limits of the radial extent. To this end, let us recall (A.16) that in this two-dimensional setup $\tau_0 = \frac{2R}{c}$, and so by (A.34), we have

$$-\frac{T_t}{2} + \frac{2L}{c} \leq t - \tau_0 \leq \frac{T_t}{2} - \frac{2L}{c}.$$

where T_t is the duration of the transmitted chirp (A.6). Because $T_t >> 4L/c$ for typical spotlight mode SAR, (A.38) yields

$$\max_t \Omega(t) = \frac{2}{c}\left(\omega_0 + 2\alpha\left(\frac{T_t}{2} - \frac{2L}{c}\right)\right) = \frac{2}{c}\left(\omega_0 + \alpha T_t - \frac{4L\alpha}{c}\right),$$

and

$$\min_t \Omega(t) = \frac{2}{c}\left(\omega_0 + 2\alpha\left(-\frac{T_t}{2} + \frac{2L}{c}\right)\right) = \frac{2}{c}\left(\omega_0 - \alpha T_t + \frac{4L\alpha}{c}\right).$$

Thus,

$$\Delta\Omega_x = \left|\max_t \Omega(t) - \min_t \Omega(t)\right| = \frac{4\alpha T_t}{c} = \frac{4\pi B_\omega}{c}.$$

As before, B_ω is the chirp bandwidth defined as $B_\omega = \alpha T_t/\pi$ in (A.10).

We now discuss the cross-range resolution, determined by $\Delta\Omega_y$. Referencing Figure A.10, let θ_a be the angle sweep (i.e. the *integration angle*) of the annulus segment. The horizontal (cross-range) and vertical (range) axis are given by Ω_y and Ω_x respectively. Following (A.37), the radius of the polar data in Figure A.10(left) (in polar coordinates) has length $2\omega_0/c = \Omega(\tau_0) := \Omega_0$. (Recall that the right most ray of the data occurs at $t = \tau_0$.) Now, using the (upper) right triangle created by the annulus of data in the first quadrant, we see that the hypotenuse has length $2\omega_0/c = \Omega(\tau_0) := \Omega_0$. The length of the side opposite of the angle $\theta_a/2$ has approximate length $\Delta\Omega_y/2$. Thus,

$$\sin\left(\frac{\theta_a}{2}\right) \approx \frac{\Delta\Omega_y/2}{\Omega_0} = \frac{\Delta\Omega_y c}{4\omega_0}.$$

Recall the wavelength of the transmitted pulse given by equation (A.8) is $\lambda = 2\pi c/\omega_0 = c/\omega_c$. Assuming θ_a is small enough such that $\sin\left(\frac{\theta_a}{2}\right) \approx \frac{\theta_a}{2}$ (which is typically the case in narrow angle SAR [69]), we have

$$\frac{\theta_a}{2} = \frac{\Delta\Omega_y c}{4\omega_0} \implies \Delta\Omega_y = \frac{2\theta_a\omega_0}{c} = \frac{4\pi\theta_a}{\lambda}.$$

The range and cross-range resolutions for the system are respectively then given by (A.41) as

$$\rho_x \approx \frac{c}{2B_\omega} \tag{A.42}$$

$$\rho_y \approx \frac{\pi c}{\omega_0\theta_a} = \frac{c}{2\omega_c\theta_a} = \frac{\lambda}{2\theta_a}. \tag{A.43}$$

Hence it is apparent that the resolution in the range direction depends on the bandwidth of the pulse used for transmission while the resolution in the cross-range direction depends on the angular diversity of observations and the central frequency of the transmitted chirp.

To demonstrate the importance of cross-range resolution, Figure A.11 illustrates two point scatterers located two feet apart at locations $p_1 = (0, -1)$ and $p_2 = (0, 1)$. According to (A.43), to get a two-foot (approximately .61m) cross-range resolution, we require

$$\theta_a = \frac{c}{2\omega_c\rho_y} \approx .0256\,\text{rad} \approx 1.4676^\circ,$$

where we have set the central frequency to $\omega_c = 9.6$GHz with a bandwidth of $B_\omega =$ 600MHz (A.10) for this example. Recall c is the speed of light given approximately as $3 \times 10^8 m/s$. That is, to achieve a 2ft range resolution roughly 1.5° of information is needed. As you see in Figure A.11(bottom-left), 2° of azimuth information places the scatterers in different *cross-range bins*, and thus they are distinguishable.

To demonstrate the effects of bandwidth on the range resolution in (A.42), consider two point scatterers located at $p_1 = (0, 0)$ and $p_2 = (0.5, 0)$, as seen in Figure A.12. In this example the central frequency is again set to $\omega_c = 9.6$GHz. The integration angle is $\theta_a = 20^\circ$, and the elevation angle is set to zero. When $B_\omega = 600$MHz, according to

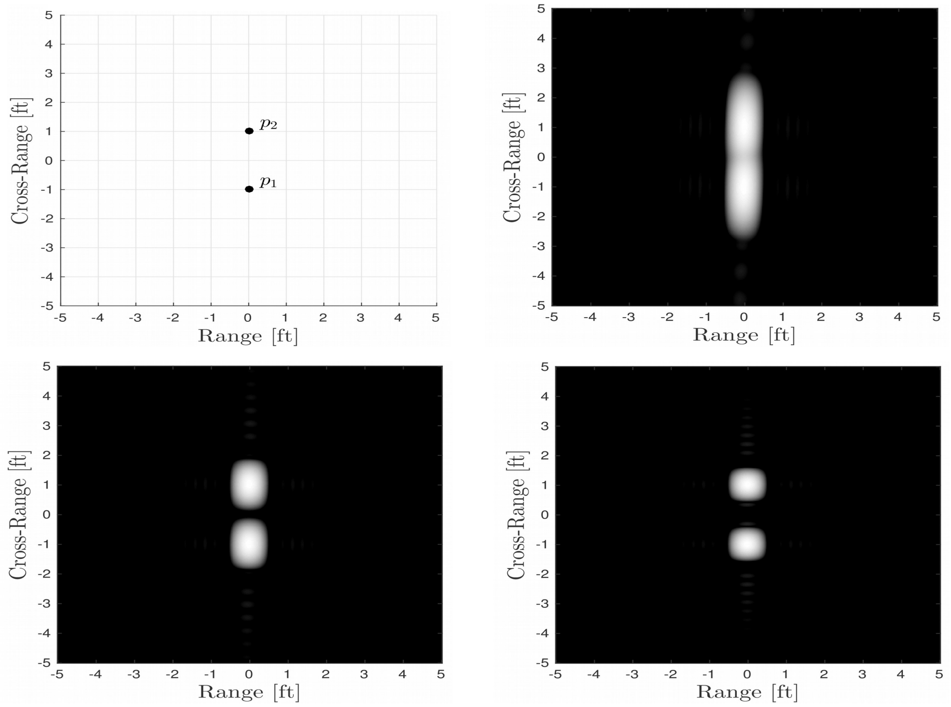

Figure A.11: Demonstration of the importance of cross-range resolution. (top-left) scatterer locations (top-right) $\theta_a = 1°$ (bottom-left) $\theta_a = 2°$ (bottom-right) $\theta_a = 3°$.

(A.42), the range resolution is

$$\rho_x = \frac{c}{2B_\omega} \approx 0.25\text{m}.$$

Because the separation is approximately .152m apart, the targets in Figure A.12 fall into the same *range bin* when $B_\omega = 600$MHz, and are therefore indistinguishable. The targets can be discerned from one another when the bandwidth is increased to 6GHz, that is, when the range resolution becomes $\rho_x \approx 0.025$m.

Discrete Data Model

So far we have described the physics underlying SAR, which is understood in a continuous-time model framework. However, data are acquired discretely, and hence we now discuss the corresponding discretized SAR model. Let $\hat{f} \in \mathbb{C}^{K \times N_p}$ be the

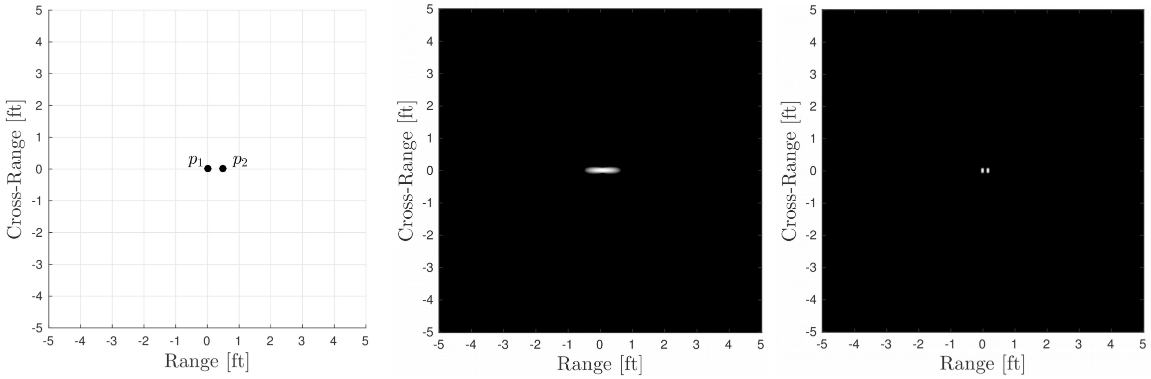

Figure A.12: Demonstration of the importance of range resolution. (Left) Two scatterers located at the same cross-range location. (Middle) Targets indistinguishable due to small bandwidth (600MHz). (Right) Targets distinguishable when bandwidth is increased (6GHz).

discrete data points, with K denoting the number of frequencies considered and N_p the number of pulses or azimuth angles considered [58, 117]. Assume that the reflectivity field $f \in \mathbb{C}^{N_x \times N_y}$ is centered at the origin of a 3D coordinate system, and that the SAR sensor travels along an arbitrary flight path such that the antenna phase center is explicitly known and located at $\boldsymbol{x} := [x_a(\tau) \;\; y_a(\tau) \;\; z_a(\tau)]^T \in \mathbb{R}^3$, where τ now denotes one instance in slow time. If we write the error in measuring the path of the antenna platform as $\tilde{\boldsymbol{x}} := [\tilde{x}_a(\tau) \;\; \tilde{y}_a(\tau) \;\; \tilde{z}_a(\tau)]^T \in \mathbb{R}^3$, then the distance from the antenna center to the scene center is

$$R(\tilde{\boldsymbol{x}}) = \sqrt{(x_a(\tau) + \tilde{x}_a(\tau))^2 + (y_a(\tau) + \tilde{y}_a(\tau))^2 + (z_a(\tau) + \tilde{z}_a(\tau))^2}.$$

Assume that for the same slow time instance τ the scatterer is at pixel location $\boldsymbol{x} = [x(\tau) \;\; y(\tau) \;\; z(\tau)]^T \in \mathbb{R}^3$, then the distance from the antenna phase center to the scatterer is

$$R_s(\boldsymbol{x}) = \sqrt{(x_a(\tau) - x(\tau))^2 + (y_a(\tau) - y(\tau))^2 + (z_a(\tau) - z(\tau))^2}. \tag{A.44}$$

The differential range to this scatterer is defined as the distance between the scene center to the scatterer and is given by

$$\Delta R(\tau) = R_s(\boldsymbol{x}) - R(\tilde{\boldsymbol{x}}). \tag{A.45}$$

To estimate (A.45), we use the Taylor expansion of $R_s(\boldsymbol{x})$ in (A.44) with respect to the scatterer location about the point $\boldsymbol{x} = (0,0,0)$ for a fixed time τ (see [117] for more in depth analysis). Note that in what follows we have dropped the dependence on τ for simplicity:

$$\begin{aligned} R_s(\boldsymbol{x}) &\approx R_s(\boldsymbol{x})\mid_{\boldsymbol{x}=\boldsymbol{0}} + x\partial_x R_s(\boldsymbol{x})\mid_{\boldsymbol{x}=\boldsymbol{0}} + y\partial_y R_s(\boldsymbol{x})\mid_{\boldsymbol{x}=\boldsymbol{0}} + z\partial_z R_s(\boldsymbol{x})\mid_{\boldsymbol{x}=\boldsymbol{0}} \\ &= \sqrt{x_a^2+y_a^2+z_a^2} + \frac{-x_a}{\sqrt{x_a^2+y_a^2+z_a^2}}x + \frac{-y_a}{\sqrt{x_a^2+y_a^2+z_a^2}}y + \frac{-z_a}{\sqrt{x_a^2+y_a^2+z_a^2}}z. \end{aligned}$$

Similarly for $R(\tilde{\boldsymbol{x}})$ with respect to the measurement error about the point $\tilde{\boldsymbol{x}} = (0,0,0)$ for the same fixed time τ, we obtain

$$\begin{aligned} R(\tilde{\boldsymbol{x}}) &\approx R(\tilde{\boldsymbol{x}})\mid_{\tilde{\boldsymbol{x}}=\boldsymbol{0}} + \tilde{x}_a\partial_{\tilde{x}_a} R(\tilde{\boldsymbol{x}})\mid_{\tilde{\boldsymbol{x}}=\boldsymbol{0}} + \tilde{y}_a\partial_{\tilde{y}_a} R(\tilde{\boldsymbol{x}})\mid_{\tilde{\boldsymbol{x}}=\boldsymbol{0}} + \tilde{z}_a\partial_{\tilde{z}_a} R(\tilde{\boldsymbol{x}})\mid_{\tilde{\boldsymbol{x}}=\boldsymbol{0}} \\ &= \sqrt{x_a^2+y_a^2+z_a^2} + \frac{x_a}{\sqrt{x_a^2+y_a^2+z_a^2}}\tilde{x}_a + \frac{y_a}{\sqrt{x_a^2+y_a^2+z_a^2}}\tilde{y}_a + \frac{z_a}{\sqrt{x_a^2+y_a^2+z_a^2}}\tilde{z}_a. \end{aligned}$$

Therefore (A.45) yields

$$\begin{aligned} \Delta R \approx & \left[\sqrt{x_a^2+y_a^2+z_a^2} + \frac{-x_a}{\sqrt{x_a^2+y_a^2+z_a^2}}x + \frac{-y_a}{\sqrt{x_a^2+y_a^2+z_a^2}}y + \frac{-z_a}{\sqrt{x_a^2+y_a^2+z_a^2}}z\right] \\ & - \left[\sqrt{x_a^2+y_a^2+z_a^2} + \frac{x_a}{\sqrt{x_a^2+y_a^2+z_a^2}}\tilde{x}_a + \frac{y_a}{\sqrt{x_a^2+y_a^2+z_a^2}}\tilde{y}_a + \frac{z_a}{\sqrt{x_a^2+y_a^2+z_a^2}}\tilde{z}_a\right] \\ = & \frac{-x_a(x+\tilde{x}_a) - y_a(y+\tilde{y}_a) - z_a(z+\tilde{z}_a)}{\sqrt{x_a^2+y_a^2+z_a^2}}. \end{aligned}$$

We now use the fact that far field assumption implies the emission of plane waves, so that we can rewrite our coordinate system as

$$x_a = L\cos\theta\cos\varphi, \quad y_a = L\sin\theta\cos\varphi, \quad z_a = L\sin\varphi, \quad \text{and } L = \sqrt{x_a^2+y_a^2+z_a^2},$$

where, as in the continuous-time model, θ is the azimuth angle, φ is the elevation (or depression) angle and L is the radial scene extent (Recall Figure A.9). This yields

$$\begin{aligned}\Delta R &\approx \left[-L\cos\theta\cos\varphi(x+\tilde{x}_a) - L\sin\theta\cos\varphi(y+\tilde{y}_a) - L\sin\varphi(z+\tilde{z}_a)\right]/L \\ &= -\cos\theta\cos\varphi(x+\tilde{x}_a) - \sin\theta\cos\varphi(y+\tilde{y}_a) - \sin\varphi(z+\tilde{z}_a).\end{aligned}$$

Assuming no error in measuring the path of the antenna, we arrive at

$$\Delta R(\tau_n) = -x\cos\theta_n\cos\varphi - y\sin\theta_n\cos\varphi - z\sin\varphi, \tag{A.46}$$

where θ_n corresponds to the azimuth angle measurement at slow time τ_n for all $n = 1, ..., N_p$.

Remark A.0.2 *Making this first order approximation along with the assumption that there are no errors in antenna location measurements causes phase errors which blur and smear the resulting reconstructed image and motivate the necessity of an autofocusing algorithm. We discuss a novel technique for correcting these errors in Chapter 5.*

At periodic intervals, the radar is transmitting pulses that reflect off of scatterers in the scene. The energy is partially reflected back to the radar. In the discrete case, we assume that a total of N_p pulses are transmitted. The two way travel time to the scatterer located at point $[x(\tau_n), y(\tau_n), z(\tau_n)]$ from the nth azimuth sampling point is referred to as the *slow time* of each pulse and is denoted by the sequence $\{\tau_n\}_{n=1}^{N_p}$. There are K temporal frequencies (*fast times*) per pulse, represented by $\{t_k\}_{k=1}^{K}$. The round-trip time of flight to a scatterer is $\tau_s = \frac{2R_s}{c}$, and the total phase change from the time of emission is $\Phi = 2\pi t_k \tau_s = \frac{4\pi t_k R_s}{c}$. If a chirped signal (A.6) is transmitted, the demodulated return from the resolution cell is the superposition of the returns from each scatterer within the resolution cell

$$\hat{f}(\theta_n, t_k) = A(\tau_n, t_k)\exp\left(\frac{-i4\pi t_k \Delta R(\tau_n)}{c}\right), \tag{A.47}$$

where c is again the speed of light in a vacuum and the amplitude $A(\tau_n, t_k)$ is related to the reflectivity f of the scene. Substituting (A.46) into (A.47) yields *discrete phase history data* (PHD) of the form

$$\hat{f}(\theta_n, t_k) = A(\tau_n, t_k) \exp\left(\frac{i4\pi t_k(x\cos\theta_n\cos\varphi + y\sin\theta_n\cos\varphi + z\sin\varphi)}{c}\right). \tag{A.48}$$

Note that (A.48) is analogous to (A.40) in the continuous-time model.

Remark A.0.3 *If the scatterer is located at the scene origin, then $\Delta R(\tau_n) = 0$, meaning a scatterer at the scene origin will have zero phase for all $k = 1, ..., K$. Correcting for errors to ensure that this is true is called* motion compensation.

Accessible Phase History Data Sets

The SAR phase history data provided by the AFRL come arranged in a matrix, where the rows of the matrix correspond to each frequency emitted, and the columns correspond to each pulse sent out. If there are K frequencies and N_p pulses (azimuth angles), the SAR phase history is a $K \times N_p$ matrix that consists of complex values:

$$\begin{pmatrix} \hat{f}(\theta_1, t_1) & \hat{f}(\theta_2, t_1) & \cdots & \hat{f}(\theta_{N_p}, t_1) \\ \hat{f}(\theta_1, t_2) & \hat{f}(\theta_2, t_2) & \cdots & \hat{f}(\theta_{N_p}, t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{f}(\theta_1, t_K) & \hat{f}(\theta_2, t_K) & \cdots & \hat{f}(\theta_{N_p}, t_K) \end{pmatrix}. \tag{A.49}$$

For each elevation angle φ, a new SAR phase history is collected. Note that if the frequency step size is given by Δt, then the maximum alias free range extent of the image is

$$W_x = \frac{c}{2\Delta t}. \tag{A.50}$$

The total bandwidth of the received pulse can be written as $B = (K-1)\Delta t$ so that the range resolution (A.42) becomes

$$\rho_x = \frac{c}{2B} = \frac{c}{2(K-1)\Delta t}. \tag{A.51}$$

Given the azimuth step size of $\Delta\theta$ and minimum frequency t_1, the maximum alias free cross-range extent of the image is

$$W_y = \frac{c}{2\Delta\theta t_1}. \tag{A.52}$$

The total integration angle of the synthetic aperture is $\theta_a = (N_p - 1)\Delta\theta$ yielding a cross-range resolution (A.43) of

$$\rho_y = \frac{\lambda}{2\theta_a} = \frac{\lambda}{2(N_p - 1)\Delta\theta}, \tag{A.53}$$

where again, $\lambda = 2\pi c/\omega_0$.

The SAR data sets provided by the AFRL consist of both measured and synthetic data. The three different data sets used in this dissertation are

1. the Civilian Vehicle Data Dome [116];
2. the Gotcha Volumetric SAR Data Set, Version 1.0 [23];
3. the MSTAR data set [122].

Each data set is described in more detail below. The measured data sets are acquired through SAR systems aboard aircraft, and there are often a variety of objects in the scene exhibiting different reflectivity patterns. Man made objects, such as cars or tanks, usually result in *diffused reflection*, where part of the radar energy will be reflected back to the radar sensor and a large (bright) response will be seen in the reconstructed image. Smooth surfaces, like paved roads or runways, result in *specular reflection*. In this case, most of the incident wave is reflected away from the radar and there is not a significant amount of return to the receiver. As a consequence, reconstructed SAR images will contain large, dark and smooth regions. By contrast, synthetic, or computer generated SAR data, are typically formed using a computer aided design (CAD) model with user determined input parameters.

Civilian Vehicle Data Dome

The Civilian Vehicle Data Dome (CV Dome) [116] [1] consists of a library of simulated X-band (frequencies ranging from 8GHz to 12GHz) scattering data for ten different civilian vehicles. For the ten facet models, the XPATCH high frequency electromagnetic scattering code [65] was used to produce fully polarized horizontal-transmit horizontal-receive (HH), verticle-transmit verticle-receive (VV), horizontal-transmit verticle-receive (HV), and verticle-transmit horizontal-receive (VH) far field, mono-static scattering for a 360° azimuth sweep and elevation angles from 30 to 60 degrees. Each data dome is stored as a set of elevation files with the file format `[vehicle]_el[elevation angle].mat`. Each file contains a structural variable called `data`, where `data.azim` is a vector of the azimuths in degrees, `data.hh`, `data.vv` and `data.hv` are complex phase history arrays, `data.elev` is the elevation angle in degrees, and `data.FGHz` is a vector of frequencies in GHz.

Gotcha Parking Lot

The Gotcha Volumetric SAR Data Set, Version 1.0 [23], consists of measured SAR phase history data collected at X-band (frequencies ranging from 8GHz to 12GHz) with 640 MHz bandwidth and full (360°) azimuth coverage at eight different elevation angles and full polarization (HH, VV, HV, VH). The scene imaged during this data collect consists of numerous civilian vehicles and calibration targets, like dihedrals and trihedrals. Thus, specular and diffuse reflection will be present in the data. The data are stored as Matlab `.mat` files. With eight passes, four polarizations per pass, and 360 degree azimuth sweeps, the data include $360 \times 8 \times 4 = 11,520$ `.mat` files. Each file contains the data for all pulses for a 1° azimuth sweep. Loading a file gives a

[1] One data dome refers to the collection of all data corresponding to a single vehicle.

single Matlab structure containing the PHD, frequencies, (x, y, z) antenna coordinate locations, range to scene center, azimuth angles and elevation angles for each pulse. In addition, a simple autofocus solution is provided for the HH and VV polarizations.

MSTAR

The Moving and Stationary Target and Recognition (MSTAR) program, 1995-2001, was charged with a mission to “design, construct and demonstrate in the laboratory an accurate and robust automatic target recognition system capable of locating and recognizing time-critical targets in air-to-ground SAR imagery,” [122]. MSTAR targets that have been imaged include the T-72 tank, the BMP2 infantry fighting vehicle and the BTR-70 armored personnel carrier. Each image consists of one vehicle on a homogeneous background, and thus there is distinct separation between the specular and diffuse reflection regions.

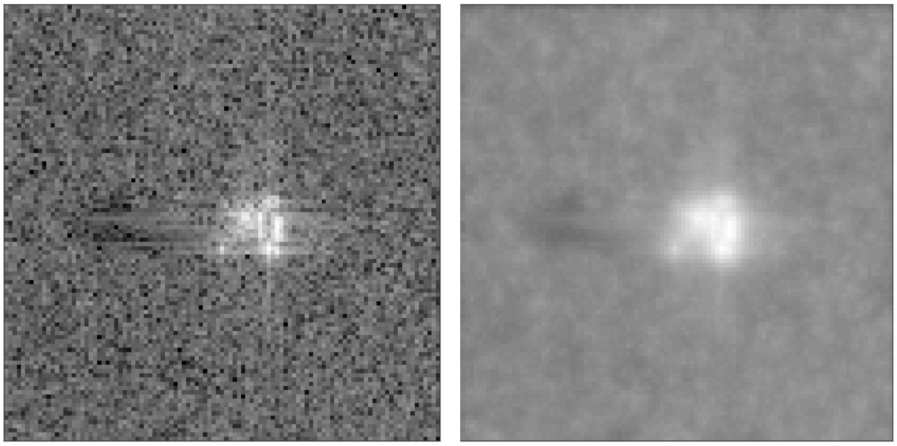

Figure A.13: (left) The image of a T-72 tank provided as a data file in the MSTAR data set. (right) The MSTAR T-72 image processed using the point and region based enhancement technique [25] with $p = .8$, $\lambda_1 = 1.7$ and $\lambda_2 = 2.5$ in (A.75).

Each MSTAR data file is constructed with a Phoenix formatted (ASCII) header that contains detailed ground truth and sensor information for the specific configuration. Following the Phoenix header is the data block, which is written in Sun floating point format and is divided into two blocks, a magnitude block followed by a phase block. The data set that is read into a readable file consists of 128 × 128 pixel image data for a variety of azimuth angles. An example of the data given in each file is displayed in Figure A.13(left). To obtain the phase history data, the image formation procedure, which is inherent in the data retrieval code, must be undone. This procedure involves taking the filtered two-dimensional inverse Fourier transform of zero-padded, phase history data on a rectangular grid. The filter used is a Taylor window, which approximates the Dolph-Chebyshev function, and is considered to be an optimal filter in the sense that it produces a frequency response corresponding to a minimum mainlobe width for a specifed maximimum side lobe level [96, 69]. The Taylor window is defined in [69] as a finite series of the form

$$w(\xi) = 1 + \sum_{m=1}^{\bar{n}-1} F_m \cos(2\pi m \xi), \quad |\xi| \le \frac{1}{2}, \tag{A.54}$$

where a discrete set of weighting coefficients is obtained by evaluating (A.54) at N equally spaced points spanning the interval $[-1/2, 1/2]$:

$$w_n = w(\xi_n),$$

where

$$\xi_n = \frac{n+1/2}{N} - \frac{1}{2}, \quad n = 0, 1, ..., N-1.$$

The coefficients of the series in (A.54) are given by

$$F_m = \frac{(-1)^{m+1} \prod_{n=1}^{\bar{n}-1} \left[1 - \frac{m^2/\sigma_p^2}{A^2 + (n+1/2)^2}\right]}{\prod_{\substack{n=1 \\ n \ne m}}^{\bar{n}-1} \left[1 - \frac{m^2}{n^2}\right]},$$

where the parameter A is determined by the specified sidelobe attenuation S_L (in dB) according to

$$A = \frac{1}{\pi} \cosh^{-1}\left(10^{S_L/20}\right),$$

and σ_p is the ratio of mainlobe width at 3 dB to the width of the ideal Dolph-Chebyshev response at 3 dB

$$\sigma_p = \frac{\bar{n}}{\sqrt{A^2 + (\bar{n} + 1/2)^2}} > 1.$$

The maximum amount of mainlobe broadening relative to Dolph-Chebshev is

$$\sigma_{pm} = \frac{\sqrt{4A^2 + 1}}{2A}$$

which occurs when

$$\bar{n} \approx 2A^2 + 1/2.$$

As $\bar{n}$ is made large, we have

$$\lim_{\bar{n} \to \infty} \sigma_p = 1,$$

and thus the Taylor window approaches the ideal Dolph-Chebyshev characteristics of minimum mainlobe width for a specified sidelobe level.

Each header file contains information about the transmitted signal bandwidth and the dynamic range of the Taylor window used to form the image. With the bandwidth information, the size of the new phase history data is determined, and the data are trimmed. The $S_L = 35$dB and $\bar{n} = 5$ Taylor window is then divided out, resulting in the untapered, trimmed phase history data. The images in Figure A.14 show the difference in the trimmed, untapered (right) versus the untrimmed, Taylor-windowed (left) MSTAR phase history data for a T-72 tank with $\theta_a = 3°$ and a $\varphi = 15°$ elevation angle.

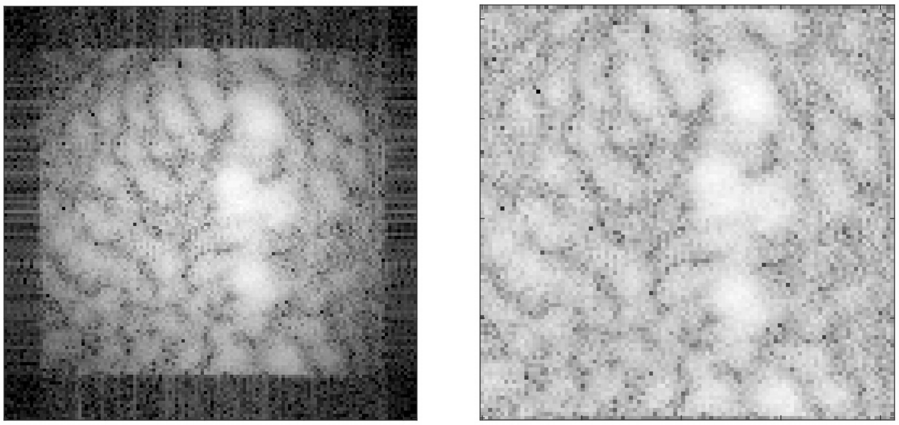

Figure A.14: Comparison of (left) the Taylor-windowed, untrimmed MSTAR PHD and (right) the trimmed, untapered MSTAR PHD for the T-72 Tank with $\theta_a = 3°$ and $\varphi = 15°$.

Current Two-Dimensional SAR Image Formation Techniques

AFRL usually employs one of three methods to form a two-dimensional SAR image: (1) the matched filter; (2) the back projection algorithm; and (3) the non-uniform fast Fourier transform (NUFFT) or the Polar Format Algorithm (PFA) [58, 60, 107]. Various types of regularization techniques have also been recently utilized in the SAR image formation process. Each method has its advantages and disadvantages, which we discuss in the following subsections for two-dimensional images. Three-dimensional image reconstruction will be left for future work. Figure A.15 displays the result of forming the image of a 1993 Jeep from the CV dome data set and the Gotcha parking lot scene using the four image formation algorithms considered.

The Matched Filter

The simplest, most straightforward SAR image formation tool is the matched filter [69, 62, 58]. It is a linear filter designed to optimize the signal-to-noise ratio

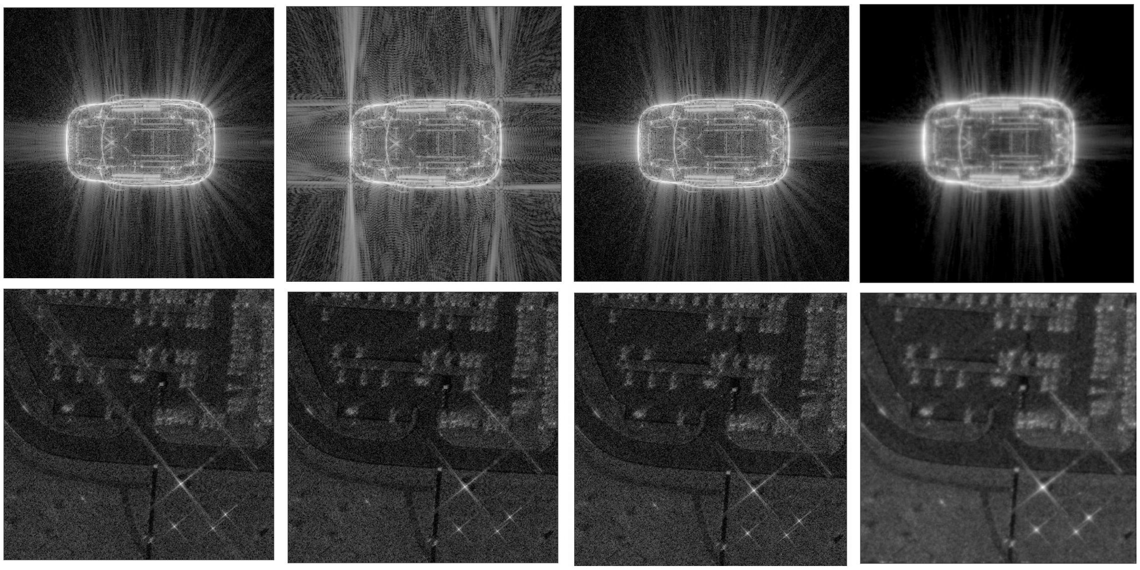

Figure A.15: (top) The 1993 Jeep image from the CV Dome data set and (bottom) the Gotcha parking lot scene reconstructed using (column 1) the matched filter, (column 2) the back projection algorithm, (column 3) the NUFFT and (column 4) a point and region based enhancement algorithm developed using ℓ_p regularization in [25]. Here the parameters are chosen so that $p = .8$, $\lambda_1 = 1.7$ and $\lambda_2 = 2.5$ in (A.75).

by convolving the returned signal with a version of the transmitted signal in the frequency domain. Using the SAR phase history data, at a fixed slow time τ_n, the matched filter response at pixel (x_l, y_m) for $l = 1, ..., N_x$ and $m = 1, ..., N_y$ is given by

$$s(x_l, y_m, \tau_n) = \frac{1}{K}\sum_{k=1}^{K} \hat{f}(\theta_n, t_k) \exp\left(\frac{i4\pi t_k \Delta R(\tau_n)}{c}\right), \quad l = 1, ..., N_x, \quad m = 1, ..., N_y, \tag{A.55}$$

where $\hat{f}(\theta_n, t_k)$ is given in (A.48). Note that according to (A.45) $\Delta R(\tau_n)$ is in fact a function of pixel location (x_l, y_m) for all $l = 1, ..., N_x$ and $m = 1, ..., N_y$, but the notational dependence is dropped for simplicity. Ideally this will then return $s(x_l, y_m, \tau_n) = \frac{1}{K}\sum_{k=1}^{K} A(\tau_n, t_k)$ so that the discrete reflectivity field $f \in \mathbb{C}^{N_x \times N_y}$ can be estimated via the matched filter as the average response over all pulses

$$f(x_l, y_m) = \frac{1}{N_p} \sum_{n=1}^{N_p} s(x_l, y_m, \tau_n), \quad l = 1, ..., N_x, \quad m = 1, ..., N_y. \tag{A.56}$$

To form an image using this method, this procedure is applied to every pixel in the image. The matched filter in this case is then the sum of phase shifted pulses. This requires calculation of the differential, ΔR, for every pixel for every pulse. Thus this method is very computationally expensive. In fact, the algorithm requires $\mathcal{O}(N^4)$ operations to construct an $(N \times N)$ two-dimensional SAR image, which is impractical for most applications. Figure (A.15)(column 1) shows an example of the Gotcha parking lot scene and a vehicle from the CV Dome data reconstructed using this matched filter algorithm. Due to its high computational costs, the matched filter is rarely used in modern SAR image formation.

Back Projection

The back projection algorithm provides an alternative method to more efficiently calculate (A.56) by using a tomographic interpretation of the SAR phase history data in (A.36), [58, 70, 39]. Back projection algorithms, such as the one described below, are commonly employed in tomography when data are collected as Fourier coefficients of projections of the underlying scene, [105]. While still computationally expensive, ($\mathcal{O}(N^3)$ operations for a $(N \times N)$ SAR image), implementing the back projection algorithm is reasonable for many SAR applications and can be parallelized. One advantage of the back projection algorithm is that SAR images can be formed as phase history data are collected, pulse-by-pulse, so that newly obtained information can be integrated into the SAR image as it becomes available.

Referencing Figure A.9, if given the projections $p(\theta, u)$, one can envision the back projection algorithm as "smearing" these projections back over the illumination patch (circle of radius L) to obtain an estimation of the reflectivity function f. Referencing (A.36) and (A.39), the filtered back projection algorithm calculates the reflectivity f such that

$$f(x, y) = \int_0^\pi \tilde{p}(\theta, x\cos\theta + y\sin\theta)d\theta \tag{A.57}$$

where the filtered radon transform (projection) at angle θ along the line u is given by

$$\tilde{p}(\theta, u) = \int_{-\infty}^{\infty} \hat{f}_\theta(t)|t|e^{iu\Omega(t)}dt. \tag{A.58}$$

Here, the multiplication by the ramp filter $|t|$ is incorporated into (A.58) to remove the blurring by $1/|t|$ that results otherwise [48]. Substituting (A.58) into (A.57) gives the back projected reflectivity $f : \Gamma \to \mathbb{C}$

$$f(x, y) = \int_0^\pi \int_{-\infty}^{\infty} \hat{f}_\theta(t)|t|e^{i\Omega(t)(x\cos\theta + y\sin\theta)}dtd\theta. \tag{A.59}$$

To estimate $f \in \mathbb{C}^{N_x \times N_y}$, the discretized back projection algorithm first range compresses the phase history data (A.48). [2] Numerical integration over the azimuth dimension is then performed by summing the interpolated samples across this set of range compressed data, [70]. Essentially, each point in the reconstructed image is obtained by numerically integrating the values back-projected from the filtered projection functions. The algorithm makes use of the fact that convolution in the spatial domain is equivalent to multiplication in the Fourier domain.

When signals are received by the radar, the returns are sorted into a set of *range bins* (frequency bins) according to their time of arrival, relative to the transmit pulse. As discussed before, the information in each range bin is not separable due to the system's range resolution. However, because the SAR system transmits long duration chirps, the returns are mapped across many bins, making the bin size small, thus

[2] Range compression is the term used for taking the one-dimensional Fourier transform of the data in the range direction.

improving resolution. The description of back projection provided below relies heavily on the one provided in Section 4 of [58].

Given the SAR phase history data, $\hat{f}(\theta_n, t_k)$ in (A.49), for $n = 1, ..., N_p$ and $k = 1, ..., K$, the *range profile* at a particular range bin b, given a received pulse at slow time τ_n, is the superposition of the range compressed data collected at the corresponding pulse

$$s(b, \tau_n) = \sum_{k=1}^{K} \hat{f}(\theta_n, t_k) \exp\left(\frac{i4\pi t_k \Delta R_b(\tau_n)}{c}\right), \quad n = 1, ..., N_p, \tag{A.60}$$

where ΔR_b is the range to every bin in the range profile. Note that $\Delta R_b \neq \Delta R$ in (A.45), but rather is a coarse sampling of ΔR. We derive ΔR_b below (see (A.65)) to obtain the maximum alias free range extent. Note also that $s(b, \tau_n)$ is the *discretized* version of the unfiltered radon transform given in (A.39).

The temporal frequency values can be written as

$$t_k = (k-1)\Delta t + t_1, \quad k = 1, ..., K, \tag{A.61}$$

where Δt denotes the distance between two consecutive frequency values. Although the frequencies are assumed to be equally spaced in this derivation, this is not in general required. Substituting (A.61) into the range profile definition, (A.60), we obtain

$$\begin{aligned} s(b, \tau_n) &= \sum_{k=1}^{K} \hat{f}(\theta_n, t_k) \exp\left[\frac{i4\pi(k-1)\Delta t \Delta R_b(\tau_n)}{c} + \frac{i4\pi t_1 \Delta R_b(\tau_n)}{c}\right] \\ &= \sum_{k=1}^{K} \hat{f}(\theta_n, t_k) \exp\left[\Phi(\Delta R_b(\tau_n))(k-1)\right] \exp\left(\frac{i4\pi t_1 \Delta R_b(\tau_n)}{c}\right), \end{aligned} \tag{A.62}$$

where

$$\Phi(\Delta R_b(\tau_n)) = \frac{i4\pi \Delta t \Delta R_b(\tau_n)}{c} \tag{A.63}$$

is the phase function.

In the discrete setting, Matlab's `fft` and `ifft` functions are used to compute (A.62) for all range bins $b = 1, .., B$. Matlab's built in functions that provide discrete Fourier transforms between $X(k)$ and $x(b)$ are defined as [140]

$$X(k) = \texttt{fft}(x(b)) = \sum_{b=1}^{B} x(b) \exp\left(\frac{-i2\pi(b-1)(k-1)}{K}\right)$$

and

$$x(b) = \texttt{ifft}(X(k)) = \frac{1}{K}\sum_{k=1}^{K} X(k) \exp\left(\frac{i2\pi(b-1)(k-1)}{K}\right) = \frac{1}{K}\sum_{k=1}^{K} X(k) \exp\left[\Theta(b)(k-1)\right],$$

where

$$\Theta(b) = \frac{i2\pi(b-1)}{K}. \tag{A.64}$$

In order to use Matlab's `fft`, we require Φ in (A.63) to be equivalent to Θ, yielding

$$\frac{i4\pi\Delta t\Delta R_b(\tau_n)}{c} = \frac{i2\pi(b-1)}{K},$$

so that

$$\Delta R_b(\tau_n) = \frac{c}{2\Delta t}\frac{(b-1)}{K} = \frac{(b-1)}{K}W_x. \tag{A.65}$$

Here W_x, given in (A.50), is the maximum alias free range extent of the image. Substituting (A.65) into (A.62) gives the range profiles as

$$\begin{aligned} s(b,\tau_n) &= \sum_{k=1}^{K} \hat{f}(\theta_n,t_k) \exp\left(\frac{i4\pi\Delta t(k-1)}{c}\Delta R_b(\tau_n)\right) \exp\left(\frac{i4\pi t_1 \Delta R_b(\tau_n)}{c}\right) \\ &= \sum_{k=1}^{K} \hat{f}(\theta_n,t_k) \exp\left(\frac{i4\pi\Delta t(k-1)}{c}\frac{(b-1)}{K}\frac{c}{2\Delta t}\right) \exp\left(\frac{i4\pi t_1}{c}\frac{(b-1)}{K}\frac{c}{2\Delta t}\right) \\ &= \sum_{k=1}^{K} \hat{f}(\theta_n,t_k) \exp\left(\frac{i2\pi(k-1)(b-1)}{K}\right) \exp\left(\frac{i2\pi t_1(b-1)}{K\Delta t}\right) \\ &= \sum_{k=1}^{K} \hat{f}(\theta_n,t_k) \exp\left(\Theta(b)(k-1)\right) \exp\left(\frac{i2\pi t_1(b-1)}{K\Delta t}\right) \\ &= K \cdot \texttt{ifft}\left[\hat{f}(\theta_n,t_k)\right] \exp\left(\frac{i2\pi t_1(b-1)}{K\Delta t}\right). \end{aligned} \tag{A.66}$$

We note that the Matlab `ifft` function by default computes (A.66) from the bin values $1 \leq b \leq B$, where $b = 1$ corresponds to the zero frequency bin and B is the maximum bin number, chosen as $B = K$ to obtain the maximum alias free range extent. However, recall from Remark A.0.3 that each pulse is *motion compensated.* That is, a scatterer at the scene origin is corrected to appear in the center frequency bin, with an equal number of bins surrounding it. Specifically, the bins should range from

$$-\frac{B}{2} + 1 \leq b \leq \frac{B}{2}. \tag{A.67}$$

Hence we use the `fftshift` function to shift the bin values to obtain the desired range in (A.67), and determine the range profiles as

$$\begin{aligned} s(b,\tau_n) &= K \cdot \texttt{fftshift}\{\texttt{ifft}(\hat{f}(\theta_n,t_k))\} \exp\left(\frac{i2\pi t_1(b-1)}{K\Delta t}\right) \\ &= K \cdot \texttt{fftshift}\{\texttt{ifft}(\hat{f}(\theta_n,t_k))\}\Phi_b(\tau_n) \end{aligned} \tag{A.68}$$

where the phase correction term is given as

$$\Phi_b(\tau_n) = \exp\left(\frac{i4\pi t_1 \Delta R_b(\tau_n)}{c}\right),$$

and the constant K in front is needed for the proper normalization of Matlab's `fft`.

Armed with the range profiles in (A.68), we are now ready to form a two-dimensional SAR image via back projection. We first make the following observations:

1. The range compressed data $s(b,\tau_n)$ are actually calculated on $\Delta R_b(\tau_n)$, but in order to use (A.55) we will need to interpolate to obtain the values corresponding to $\Delta R(\tau_n)$ in (A.45). To do this, we linearly interpolate the computed values of $s(b,\tau_n)$ to obtain $s_{int}(x_l, y_m, \tau_n)$ via the Matlab `interp1` function.

2. The true range profile $s(b,\tau_n)$ should be *band limited.* However because (A.68) is a finite Fourier approximation, it is highly oscillatory. To filter out the oscillations and more accurately recover the true range profile, we use zero-padded

sinc interpolation to obtain

$$\tilde{s}(b,\tau_n) = N \cdot \texttt{fftshift}\{\texttt{ifft}(\hat{f}(\theta_n,t_k))\}\Phi_b(\tau_n), \tag{A.69}$$

where $\hat{f}(\theta_n,t_k) = 0$ for all $k > K$. The zero-padding prevents aliasing and, due to the periodic convolutional properties of the FFT [48], approximates the filtered radon transform (A.58) on the discrete grid. Following [58], we choose the length of the `ifft` as $N = 2^q \approx 10K$, where $q > 0$ is an integer. (Note also that in order to optimize efficiency of the `fft`, $K = 2^p$, $p > 0$ an integer.)

We now use (A.69) as the values from which to interpolate, which we denote as $s_{int}(x_l,y_m,\tau_n)$. The final image response is then found simply by summing these values over each pulse τ_n as

$$f(x_l,y_m) = \sum_{n=1}^{N_p} s_{int}(x_l,y_m,\tau_n), \quad l = 1,...,N_x, \quad m = 1,...,N_y. \tag{A.70}$$

Note that this result is analogous to (A.59) in the continuous data case.

Figure A.15(column 2) displays the results of the Gotcha parking lot image and a 1993 Jeep being reconstructed using the back projection algorithm. The "smearing" inherent in the back projection is readily apparent in the CV Dome image. This is because the car being imaged only returns a large portion of the transmitted signal at the cardinal angles (0°, 90°, 180° and 270°).

Non-Uniform Fast Fourier Transform (NUFFT)

Fast Fourier transforms (FFT) are commonly used in data analysis and inversion when data are sampled on uniform grids. The computational cost of computing Fourier modes via a FFT is $\mathcal{O}(N \log N)$ for a function/image sampled at N data points, rather than the typical $\mathcal{O}(N^2)$ operations required by comparable methods. However, when data are irregularly sampled in either the physical or frequency domain, as is the case in SAR, the FFT does not apply. Over the past twenty years, a number of algorithms have been developed to overcome this limitation [60, 107, 49]. These algorithms are commonly referred to as non-uniform fast Fourier transforms (NUFFT).

The goal of the NUFFT is to efficiently compute forward and inverse Fourier transforms when samples are non-uniformly distributed. The inverse NUFFT involves taking the non-uniformly spaced frequency domain data, interpolating it onto a uniform grid and then performing an inverse FFT. This results in uniformly spaced data in the spatial/time domain. In SAR, the forward NUFFT takes uniform time data, performs an FFT, and interpolates onto a non-uniform grid to obtain the non-uniform frequency data. Figure A.15(column 3) provides an example of using the inverse NUFFT algorithm devloped in [60, 107] to reconstruct the Gotcha parking lot scene and an image from the CV dome data set. The inverse NUFFT is besot with a number of difficulties, mainly coming from interpolation errors in the frequency domain. However, as is shown in Chapter 5, the NUFFT algorithms are useful in the ℓ_1 regularization setup.

Regularization

Compressed sensing and, more generally, regularization, allows the use of a discrete forward model and facilitates incorporation of known properties of the SAR sensor and measurement parameters into the SAR image formation procedure. Such an approach allows more effective handling of the limitations of the model due to data quantity, which arise due to restrictions on angular diversity, resolution issues and missing observations. Regularization techniques also enable the use of statistical processing methods to reduce the impact of noisy data. In particular, regularization-based techniques can alleviate problems that arise in cases of incomplete data or sparse apertures and can produce images with improved resolution, suppressed side lobes, and reduced speckle noise [24, 25, 27, 110, 115, 26]. Below, we formulate the estimation of a SAR image as the solution to a regularization problem originally posed in [24, 25].

For SAR, the forward model is easily derived from the Fourier relationship inherent in the SAR phase history data [24]. We note that in using the Fourier projection algorithm, either the tomographic or classical interpretation of the data can be utilized with a corresponding change in the forward model. Recall the relationship between the observed data $\hat{f}_\theta(t)$ and underlying reflectivity function $f(x, y)$ given in (A.40) and the discretized phase history data given in (A.48). Let N_p be the number of pulses observed, K be the number of spatial frequencies, N_x be the number of pixels in the range direction and N_y be the number of pixels in the cross-range direction. Stacking the rows of the phase history data on top of one another yields $\hat{f} \in \mathbb{C}^{N_p K}$ with

$$\hat{f}_{n,k} = \left\{ \hat{f}_{\theta_n}(t_k) \right\}_{n=1,k=1}^{N_p,K},$$

where t_k are the discrete frequencies and θ_n are the discrete azimuth locations considered. Let $\mathcal{F}$ be the observation kernel $\mathcal{F} \in \mathbb{C}^{N_p K} \to \mathbb{C}^{N_x N_y}$ given at elevation angle φ as

$$\mathcal{F} = \left\{ \exp\left(\frac{i4\pi t_k (x_l \cos\theta_n + y_m \sin\theta_n) \cos\varphi}{c} \right) \right\}_{p=1,k=1}^{N_p,K}, \tag{A.71}$$

for $l = 1, ..., N_x$, $m = 1, ..., N_y$ and $f \in \mathbb{C}^{N_x N_y}$ be the unknown scene reflectivity discretized as

$$f_{l,m} = \{ f(x_l, y_m) \}_{l=1,m=1}^{N_x,N_y}.$$

The regularization model is then formulated as [24, 25, 27, 110, 115, 26]:

$$f = \underset{f}{\operatorname{argmin}}\ J(f), \tag{A.72}$$

where

$$J(f) := \left|\left| \mathcal{F}f - \hat{f} \right|\right|_2^2 + \lambda_1 \left|\left| f \right|\right|_p^p + \lambda_2 \left|\left| \nabla |f| \right|\right|_p^p. \tag{A.73}$$

Here, $||\cdot||_p$ denotes the ℓ_p-norm (here $p \leq 1$), ∇ is a discrete approximation to the two-dimensional derivative operator, $|f|$ denotes the vector of magnitudes of the complex-valued vector f, and λ_1 and λ_2 are scalar, real, nonnegative parameters. The first term in the objective function incorporates the SAR observation model and thus information about the observation geometry. The second term serves to enhance point-based features in the image. A smaller p will favor a reflectivity function with a smaller number of dominant scatterers and better preservation of the scatterers and their magnitudes. The third term is designed to enhance region-based features. By including this term, variability in homogeneous regions is reduced while discontinuities at region boundaries are preserved. This sort of behavior has been obtained in real-valued image restoration and reconstruction problems by using the constraint $||\nabla f||_p^p$, where $p \approx 1$. The direct application of such a constraint to the complex-valued SAR reflectivity function is problematic, because it separately imposes smoothness on the real and imaginary parts of the complex field f. Hence the constraint is instead only enforced on the magnitude of the field $|f|$.

In [24, 25, 27, 110, 115, 26], a quasi-Newton based algorithm is proposed to solve (A.72). In order to avoid non-differentiability of the ℓ_p-norm around the origin when $p \leq 1$, the following smooth approximation to the ℓ_p-norm is made:

$$||z||_p^p \approx \sum_{j=1}^{Z} \left(|z_j|^2 + \epsilon\right)^{p/2}, \tag{A.74}$$

where $\epsilon \geq 0$ is a small constant and Z is the length of the complex vector z. For numerical purposes, therefore, the cost function is modified as

$$J_\epsilon(f) := \left|\left|\mathcal{F}f - \hat{f}\right|\right|_2^2 + \lambda_1 \sum_{j=1}^{N_x N_y} \left(|f_j|^2 + \epsilon\right)^{p/2} + \lambda_2 \sum_{j=1}^{N_x N_y} \left(|(D|f|)_j|^2 + \epsilon\right)^{p/2}, \tag{A.75}$$

where $D|f|$ represents the discrete first derivative approximation of $|f|$, and it is noted that $J_\epsilon(f) \to J(f)$ as $\epsilon \to 0$. A simple, closed-form solution for the minimizer of $J_\epsilon(f)$

does not exist in general, so a quasi-Newton-based algorithm is used to determine an optimal estimation of f [24, 156]. Figure A.15(column 4) demonstrates the results of using this technique to form estimations of the reflectivity fields associated with the Gotcha parking lot and the CV dome. In Figure A.13(right) the same technique is applied to the MSTAR data set. In each case, the algorithm uses the optimal parameters given in [25]. In this thesis we refer to this method as the point and region based enhancement (PRBE) method.

APPENDIX B

STATISTICAL MODEL OF SPECKLE NOISE

Here we present a thorough description of the statistics of speckle noise. Pieces of this model have been presented in various sources, [57, 56, 34, 69], and are now compiled for the readers' convenience. In particular we describe how to determine the probability density function of the intensity of a speckled image.

First, let us assume that the amplitude $\alpha_k/\sqrt{N}$ and phase ϕ_k returned to the radar from the kth scatterer are statistically independent realizations of the random variables Φ and A, respectively. In each resolution cell, there exist N scatterers, thus the return from one resolution cell can be written as the random phasor sum

$$\hat{f}(\alpha, \phi) = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} \alpha_k e^{i\phi_k}. \tag{B.1}$$

This process can be illustrated as a Gaussian random walk, as seen in Figure (B.1).

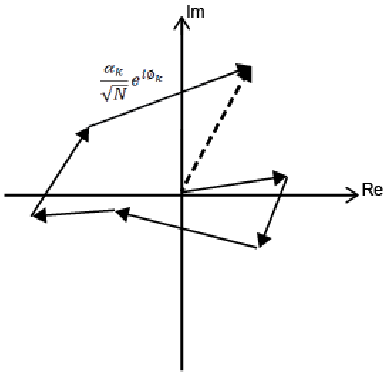

Figure B.1: Example of a Gaussian random walk which describes the summation of the N complex phasors in (B.1).

If the amplitudes are further assumed to be identically distributed for all k we have

$$E\left[\alpha_k\right] = \overline{\alpha}, \quad E\left[\alpha_k^2\right] = \overline{\alpha^2}.$$

Moreover if the phases ϕ_k are uniformly distributed on $(-\pi, \pi)$, the probability density function for each ϕ_k is given by

$$p_\Phi(\phi) = \begin{cases} \frac{1}{2\pi}, & \phi \in (-\pi, \pi) \\ 0, & \text{else.} \end{cases} \tag{B.2}$$

Note that this assumption can easily be relaxed so that the ϕ_k are uniformly distributed over another interval. Observe that (B.2) implies

$$E[\cos\phi_k] = E[\sin\phi_k] = 0, \qquad k = 1, ..., N.$$

Now let

$$R = \text{Re}\{\hat{f}(\alpha, \phi)\} = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} \alpha_k \cos\phi_k \tag{B.3}$$

$$I = \text{Im}\{\hat{f}(\alpha, \phi)\} = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} \alpha_k \sin\phi_k. \tag{B.4}$$

Note that R and I are real, independent random variables because they are sums of real, independent random variables. Let r and i be realizations of the random variables R and I. We seek the joint probability density function of these two variables $p_{R,I}(r,i)$. To do so, we first calculate the mean, variance and correlation coefficient for R and I. Using the independence of α_k and ϕ_k along with the identically distributed assumptions we have

$$E[R] = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} E\left[\alpha_k \cos\phi_k\right] = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} E\left[\alpha_k\right] E\left[\cos\phi_k\right] = \sqrt{N} E[\alpha] E[\cos\phi] = 0$$

Similarly,

$$E[I] = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} E\left[\alpha_k \sin\phi_k\right] = \sqrt{N} E[\alpha] E[\sin\phi] = 0.$$

Thus

$$Var(R) = E[R^2] - E[R]^2 = E[R^2] := \sigma_R^2,$$

$$Var(I) = E[I^2] - E[I]^2 = E[I^2] := \sigma_I^2,$$

where

$$E[R^2] = \frac{1}{\sqrt{N}}\frac{1}{\sqrt{N}}\sum_{k=1}^{N}\sum_{n=1}^{N} E\left[\alpha_k \cos\phi_k \alpha_n \cos\phi_n\right] = \frac{1}{N}\sum_{k=1}^{N}\sum_{n=1}^{N} E[\alpha_k\alpha_n]E[\cos\phi_k \cos\phi_n],$$

and

$$E[I^2] = \frac{1}{\sqrt{N}}\frac{1}{\sqrt{N}}\sum_{k=1}^{N}\sum_{n=1}^{N} E\left[\alpha_k \sin\phi_k \alpha_n \sin\phi_n\right] = \frac{1}{N}\sum_{k=1}^{N}\sum_{n=1}^{N} E[\alpha_k\alpha_n]E[\sin\phi_k \sin\phi_n].$$

Because

$$E[\cos\phi_k \cos\phi_n] = E[\sin\phi_k \sin\phi_n] = \begin{cases} 0, & k \neq n \\ \frac{1}{2}, & k = n, \end{cases}$$

we can further conclude that

$$E[R^2] = E[I^2] = \frac{E[\alpha^2]}{2} = \frac{\overline{\alpha^2}}{2} =: \sigma^2. \tag{B.5}$$

Next we evaluate the correlation between R and I. Using the trigonometric identity $\cos\phi \sin\phi = \frac{1}{2}\sin 2\phi$ and the independence of the phases,

$$E[\cos\phi_k \sin\phi_n] = \begin{cases} E[\cos\phi_k]E[\sin\phi_n] = 0, & k \neq n \\ \frac{1}{2}E[\sin 2\phi_k] = 0, & k = n. \end{cases}$$

The correlation coefficient is

$$\begin{aligned} E[RI] &= \frac{1}{\sqrt{N}}\frac{1}{\sqrt{N}}\sum_{k=1}^{N}\sum_{n=1}^{N} E\left[\alpha_k \cos\phi_k \alpha_n \sin\phi_n\right] \\ &= \frac{1}{N}\sum_{k=1}^{N}\sum_{n=1}^{N} E[\alpha_k\alpha_n]E[\cos\phi_k \sin\phi_n] = 0. \end{aligned}$$

Hence the real and imaginary parts of the resultant phasor are uncorrelated. The zero means, equality of variances and lack of correlation are true for any N, finite or infinite.

In the limit of very large N, the joint density function of the real and imaginary parts of the random phasor sum is asymptotically a complex Gaussian random variable. To show this, we rely on the uniform independence of the phases as well as the independence of the amplitudes from each other and all phases. The joint characteristic function is defined as the two-dimensional inverse Fourier transform of the probability density function:

$$M_{R,I}(\omega_1,\omega_2) := E\left[e^{i(\omega_1 R+\omega_2 I)}\right] = \iint\limits_{\mathbb{R}^2} p_{R,I}(r,i)e^{i(\omega_1 R+\omega_2 I)}dRdI. \tag{B.6}$$

Using (B.3) and (B.4) along with a transformation to polar coordinates

$$\omega_1 = \Omega\cos\chi, \quad \omega_2 = \Omega\sin\chi,$$

yields

$$\begin{aligned}
M_{R,I}(\omega_1,\omega_2) &= E\left[\exp\left\{i\left(\Omega\cos\chi\frac{1}{\sqrt{N}}\sum_{k=1}^{N}\alpha_k\cos\phi_k + \Omega\sin\chi\frac{1}{\sqrt{N}}\sum_{k=1}^{N}\alpha_k\sin\phi_k\right)\right\}\right] \\
&= E\left[\exp\left\{i\left(\frac{1}{\sqrt{N}}\sum_{k=1}^{N}\left(\Omega\cos\chi\cos\phi_k + \Omega\sin\chi\sin\phi_k\right)\alpha_k\right)\right\}\right] \\
&= E\left[\exp\left\{\frac{i\Omega}{\sqrt{N}}\sum_{k=1}^{N}\alpha_k\cos(\chi-\phi_k)\right\}\right] \\
&= E\left[\exp\left\{\frac{i\Omega}{\sqrt{N}}\alpha_1\cos(\chi-\phi_1)\right\}\right]E\left[\exp\left\{\frac{i\Omega}{\sqrt{N}}\alpha_2\cos(\chi-\phi_2)\right\}\right]\dots \\
&\quad E\left[\exp\left\{\frac{i\Omega}{\sqrt{N}}\alpha_N\cos(\chi-\phi_N)\right\}\right] \\
&= \prod_{k=1}^{N}E\left[\exp\left\{\frac{i\Omega}{\sqrt{N}}\alpha_k\cos(\chi-\phi_k)\right\}\right] \\
&\approx \prod_{k=1}^{N}E\left[I_0\left(\frac{2\alpha_k\Omega}{\sqrt{N}}\right)\right],
\end{aligned}$$

where I_0 is the zeroth order, modified Bessel function of the first kind [1]. Here we have utilized the generating function

$$e^{z\cos\theta} = I_0(z) + 2\sum_{k=1}^{\infty}I_k(z)\cos\theta.$$

The series representation for I_0 is

$$I_0(x) = \sum_{k=0}^{\infty}\frac{\left(\frac{1}{4}x^2\right)^k}{(k!)^2}. \tag{B.7}$$

Notice that as $N \to \infty$, $\left(\frac{\alpha_k \Omega}{\sqrt{N}}\right) \to 0$. Thus, the characteristic function can be approximated by the first two terms of the power series expansion (B.7) about the origin

$$M_{R,I}(\omega_1, \omega_2) = \prod_{k=1}^{N} E\left[1 - \left(\frac{2\alpha_k \Omega}{\sqrt{N}}\right)^2\right].$$

Averaging over the amplitudes α_k gives

$$M_{R,I}(\omega_1, \omega_2) = \left[1 - \frac{4\overline{\alpha^2}\Omega^2}{N}\right]^N.$$

As the number of terms N increases without bound, the joint characteristic function of the real and imaginary parts of the random phasor sum asymptotically approaches that of a zero mean, complex Gaussian random variable:

$$\lim_{N\to\infty} M_{R,I}(\omega_1, \omega_2) = \lim_{N\to\infty} \left[1 - \frac{\overline{\alpha^2}\Omega^2}{N}\right]^N = e^{-4\overline{\alpha^2}\Omega^2}.$$

Because this is the characteristic function for a complex normal, taking the inverse Fourier transform of the joint characteristic function yields the joint probability function:

$$p_{R,I}(r, i) = \frac{1}{4\pi\overline{\alpha^2}} \exp\left\{-\frac{r^2 + i^2}{4\overline{\alpha^2}}\right\} = \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{r^2 + i^2}{2\sigma^2}\right\}, \tag{B.8}$$

where we have used (B.5). Thus, by virtue of the central limit theorem, $\hat{f}(\alpha, \phi)$ in (B.1) is a circular, complex, Gaussian random variable. By assuming N is significantly large, we are assuming that the speckle is *fully developed.*

We now discuss the statistics of the magnitude α and phase ϕ of the resultant distribution where

$$\alpha = \sqrt{R^2 + I^2} \tag{B.9}$$

$$\phi = \tan^{-1} \frac{I}{R}. \tag{B.10}$$

The joint density function $p_{A,\Phi}(\alpha, \phi)$ is found by mapping into the polar coordinate system and transforming the joint probability density function given in equation (B.8):

$$R = \alpha \cos\phi; \quad I = \alpha \sin\phi,$$

where the corresponding Jacobian is

$$J = \begin{vmatrix} \frac{\partial R}{\partial \alpha} & \frac{\partial R}{\partial \phi} \\ \frac{\partial I}{\partial \alpha} & \frac{\partial I}{\partial \phi} \end{vmatrix} = \begin{vmatrix} \cos\phi & -\alpha\sin\phi \\ \sin\phi & \alpha\cos\phi \end{vmatrix} = \alpha. \tag{B.11}$$

Recall that ϕ is uniformly distributed on $(-\pi, \pi)$ and $\alpha > 0$ (because it is a magnitude). The joint density function is

$$\begin{aligned} p_{A,\Phi}(\alpha,\phi) &= J \cdot p_{R,I}\left(R = \alpha\cos\phi, I = \alpha\sin\phi\right) \\ &= \begin{cases} \frac{\alpha}{2\pi\sigma^2}\exp\left\{-\frac{\alpha^2}{2\sigma^2}\right\}, & -\pi < \phi \leq \pi, \quad \alpha > 0 \\ 0, & \text{else.} \end{cases} \end{aligned} \tag{B.12}$$

Then by marginalization the probability density function of the amplitude is

$$p_A(\alpha) = \int_{-\pi}^{\pi} g_{A,\Phi}(\alpha,\phi)d\phi = \int_{-\pi}^{\pi} \frac{\alpha}{2\pi\sigma^2}e^{-\frac{\alpha^2}{2\sigma^2}}d\phi = \begin{cases} \frac{\alpha}{\sigma^2}e^{-\alpha^2/2\sigma^2}, & \alpha > 0 \\ 0, & \text{else.} \end{cases} \tag{B.13}$$

This is the well known *Rayleigh* density function.

Once again marginalizing (B.12), this time integrating with respect to α and using the fact that integrating a probability density function over the its entire support results in unity, we obtain for $-\pi < \phi \leq \pi$:

$$p_\Phi(\phi) = \frac{1}{2\pi}\int_0^\infty \frac{\alpha}{\sigma^2}e^{-\alpha^2/2\sigma^2}d\alpha = \frac{1}{2\pi}. \tag{B.14}$$

Thus again we see that the phase is uniformly distributed on $(-\pi, \pi)$. Notice that

$$p_{A,\Phi}(\alpha,\phi) = p_A(\alpha)p_\Phi(\phi),$$

showing that the amplitude and phase are independent random variables.

Finally, we discuss the properties of the intensity g of the speckled return via the transformation

$$g = \alpha^2, \quad \alpha = \sqrt{g}.$$

Here, g is modeled as a realization of the random vector G. Because α obeys a Rayleigh probability density function (B.13), and the above transformation is monotonic on $(0, \infty)$, we have

$$p_G(g) = p_G(\alpha = \sqrt{g}) \left| \frac{d\alpha}{dg} \right| = \frac{\sqrt{g}}{\sigma^2} e^{-g/2\sigma^2} \frac{1}{2\sqrt{g}} = \begin{cases} \frac{1}{2\sigma^2} e^{-g/2\sigma^2}, & g \geq 0 \\ 0, & \text{else.} \end{cases}$$

The intensity obeys a negative exponential probability density function with the special property that its standard deviation is equal to its mean, both of which equal $2\sigma^2$:

$$\sigma_g = \bar{g} = 2\sigma^2.$$

In a more compact notation,

$$p_G(g) = \begin{cases} \frac{1}{\bar{g}} e^{-g/\bar{g}}, & g \geq 0 \\ 0, & \text{else.} \end{cases} \tag{B.15}$$

With this probability density function, we can now say how likely we are to observe a bright peak or dark null in the intensity of a SAR image. The probability that the intensity exceeds a given threshold g_t is

$$\begin{aligned} P(g > g_t) = 1 - P(g \leq g_t) &= 1 - \int_0^{g_t} p_G(g) d\nu \\ &= 1 - \int_0^{g_t} \frac{1}{\bar{g}} e^{-g/\bar{g}} d\nu = 1 - \left[1 - e^{g_t/\bar{g}}\right] = e^{-g_t/\bar{g}}. \end{aligned}$$

The analysis provided in this appendix supplements the speckle model provided in Chapter 5, and we note that many of the details here are interesting, but unnecessary to understand how speckle manifests in a coherent imaging system like SAR.

APPENDIX C

PERMISSION TO USE PUBLISHED PAPERS

Chapter 3 of this dissertation was submitted for publication to the Journal of Scientific Computing in December 2017 under the title "Reducing Effects of Bad Data Using Variance Based Joint Sparsity Recovery." It was co-authored with Prof. Anne Gelb.

Chapter 4 of this dissertation has been published under the citation

> Scarnati, T., Gelb, A., & Platte, R.B. "Using ℓ_1 Regularization to Improve Numerical Partial Differential Equation Solvers." (2018). *Journal of Scientific Computing.* 75(1), 225-252.

Parts of Section 5.1 of this dissertation has been published in the following conference proceedings:

> Scarnati, T., Zelnio, E., & Paulson, C. (2017). "Exploiting the sparsity of edge information in synthetic aperture radar imagery for speckle reduction." *SPIE Defense + Security.* 102010C. International Society for Optics and Photonics.
>
> Scarnati, T., & Gelb, A. (2018). "Variance Based Joint Sparsity Reconstruction of Synthetic Aperture Radar Data for Speckle Reconstruction." *SPIE Commercial + Scientific Sensing and Imaging.*

The work in Section 5.2 originated in

> Sanders, T., & Scarnati, T. (2017). "Combination of correlated phase error correction and sparsity models for SAR." *SPIE Commercial + Scientific Sensing and Imaging.* 102220E. International Society for Optics and Photonics.

Section 5.2 was submitted for publication to the Journal of Computational Physics in March 2018 under the title "Joint Image Formation and Two-Dimensional Autofocusing for Synthetic Aperture Radar Data." It was co-authored with Prof. Anne Gelb.

I certify that I have obtained permission from my co-authors to include these papers in this dissertation.

Theresa Scarnati