Developments in LFM-CW SAR for UAV Operation

Craig Stringham

A dissertation submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

Brigham Young University

December 2014

ABSTRACT

Developments in LFM-CW SAR for UAV Operation

Craig Stringham
Department of Electrical and Computer Engineering, BYU
Doctor of Philosophy

Opportunities to use synthetic aperture radar (SAR) in scientific studies and military operations are expanding with the development of small SAR systems that can be operated on small unmanned air vehicles (UAV)s. While the nimble nature of small UAVs make them an attractive platform for many reasons, small UAVs are also more prone to deviate from a linear course due autopilot errors and external forces such as turbulence and wind. Thus, motion compensation and improved processing algorithms are required to properly focus the SAR images. The work of this dissertation overcomes some of the challenges and addresses some of the opportunities of operating SAR on small UAVs.

Several contributions to SAR backprojection processing for UAV SARs are developed including: 1. The derivation of a novel SAR backprojection algorithm that accounts for motion during the pulse that is appropriate for narrow or ultra-wide-band SAR. 2. A compensation method for SAR backprojection to enable radiometrically accurate image processing. 3. The design and implementation of a real-time backprojection processor on a commercially available GPU that takes advantage of the GPU texture cache. 4. A new autofocus method that improves the image focus by estimating motion measurement errors in three dimensions, correcting for both amplitude and phase errors caused by inaccurate motion parameters. 5. A generalization of factorized backprojection, which we call the Dually Factorized Backprojection method, that factorizes the correlation integral in both slow-time and fast-time in order to efficiently account for general motion during the transmit of an LFM-CW pulse.

Much of this work was conducted in support of the Characterization of Arctic Sea Ice Experiment (CASIE), and the appendices provide substantial contributions for this project as well, including: 1. My work in designing and implementing the digital receiver and controller board for the microASAR which was used for CASIE. 2. A description of how the GPU backprojection was used to improved the CASIE imagery. 3. A description of a sample SAR data set from CASIE provided to the public to promote further SAR research.

Keywords: radar, SAR, UAV, GPU, autofocus, backprojection, fast-factorized backprojection, LFM-CW

ACKNOWLEDGMENTS

I thank Dr. David Long for advising me during my graduate and undergraduate work. This dissertation has been made possible by his support and guidance. Although he jokes that his students are all “Long-suffering”, I am very grateful for this “SARry” experience that has given me many great opportunities and helped me to grow. I thank all of the members of my graduate committee for their help and consideration of my work at BYU.

While I cannot name all people and organizations that have helped me along the way, I would like to specifically thank Evan Zaugg and Matt Edwards for tutoring me early on in my study of SAR, and I also thank Gordon Farquharson for the associations and discussions that we have had which encouraged me to pursue some of the work in this dissertation.

I am grateful for the financial support that I have received during this work, including funding for various projects arranged by Dr. Long and from fellowships from the Rocky Mountain Space Grant Consortium and the Jet Propulsion Laboratory.

I thank my wife Andrea, on whom I have greatly relied on over the years. I cannot imagine being here without her encouragement, love, and support. I am also very grateful for my family, especially my parents. They have always encouraged me to work hard and pursue worthy goals.

Most of all, I thank God for giving me the wonderful opportunities and associations that I have had at BYU.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The need for remote monitoring continues to increase in nearly all fields of science, defense, and security. The use of synthetic aperture radar (SAR) continues to expand in all of these areas. SAR can monitor an area regardless of solar illumination conditions. Compared to optical imaging systems, SAR systems operate at much longer wavelengths so that many systems can see through clouds and some systems penetrate through foliage and even some layers of the ground. Thus SAR can provide information that is unobtainable with optical systems.

Historically SAR systems have been both large and expensive, but due to advances in technology, both the size and the cost of SAR systems have been drastically reduced. The BYU microSAR developed in 2004 was revolutionary in being able to generate quality imagery in a package small enough to fit on a very small unmanned air vehicle (UAV) [1]. While the microSAR was very useful, it has many limitations, and so building on the microSAR design, BYU helped design the microASAR to overcome these challenges [2, 3]. With SAR systems small enough to fit on UAVs the opportunities to use SAR in scientific research and military missions has drastically increased because UAV carrying SARs can obtain high-quality high-resolution information over areas too difficult, dangerous, or costly to reach in manned aircraft.

While the nimble nature of small UAVs provides many new opportunities to use SAR, small UAVs are also much more susceptible to external forces, thus significant motion compensation is required in order to properly focus SAR images. Low-altitude operation further complicates motion compensation of stripmap SAR images, due to the large range of incidence angles and increased range cell migration. In many cases SAR backprojection can be used to sufficiently focus the images; however, because of the considerable computational burden there has been little development of backprojection and accompanying SAR focusing algorithms. Fortunately the recent advances in signal processing capabilities available in desktop computers, and particularly in recent graphics processing unit (GPU) architectures, give opportunities to overcome the high computational burden.

This dissertation addresses some of the challenges and opportunities of operating SAR on small UAVs. It includes novel contributions to the field with the design and implementation of hardware enabling operation on a small UAV, processing algorithms designed to compensate for platform motion even during a pulse, and an autofocus algorithm that can estimate three dimensional motion, which is necessary for low-altitude operation.

Several contributions to SAR backprojection processing for UAV SARs are developed including: 1. The derivation of a novel SAR backprojection algorithm that accounts for motion during the pulse that is appropriate for narrow or ultra-wide-band SAR. 2. A compensation method for SAR backprojection to enable radiometrically accurate image processing. 3. The design and implementation of a real-time backprojection processor on a commercially available GPU that takes advantage of the GPU texture cache. 4. A new autofocus method that improves the image focus by estimating motion measurement errors in three dimensions, correcting for both amplitude and phase errors caused by inaccurate motion parameters. 5. A generalization of factorized backprojection methods, which we call the Dually Factorized Backprojection method, that factorizes the correlation integral in both slow-time and fast-time in order to efficiently account for general motion during the transmit of an LFM-CW pulse.

1.1 Contributions

Specifically, the contributions of this work include:

1. The derivation of a novel SAR backprojection algorithm that accounts for motion during the pulse, extending the work by Ribalta [4] and Zaugg [5] to ultra wide-band (UWB) SAR.

2. A compensation method for SAR backprojection to enable radiometrically accurate processing. Traditional backprojection methods do not account for the pointing of

the aircraft or the number of samples that contribute to a pixel. To account for the unequal weighting, an additional summation of the weightings from the apodization windows and the propagation loss are calculated.

3. The design and implementation of a real-time backprojection processor on a GPU that includes the previous two contributions. This processor takes advantage of the unique architecture of an NVIDIA GPU, including the texture cache. This work also describes how any convolutional interpolation method can be accelerated using the GPU's texture cache.

4. An autofocus method designed to account for unmeasured motion of a low-altitude aircraft to improve the focus of the SAR image. This extends mapdrift autofocus methods to handle three dimensional motion, which can be estimated because of the wide range of incidence angles that are contained in a low altitude SAR image scene.

5. A generalization of factorized backprojection methods that accounts for motion during the transmit of a pulse. This method factorizes the correlation processing in both slow-time and fast-time resulting in fast and accurate image formation for ultra-wide band, ultra-wide beam SAR with general motion over the synthetic aperture and during a single pulse.

6. A spectral analysis of stripmap SAR collected on a platform with oscillatory motion that investigates the possibility of estimating scene topography by applying interferometric techniques. This analysis extends previous work for spot-light SAR to stripmap SAR. The motion requirements for applying interferometery to single channel stripmap SAR are considered, and are shown to be impractical for typical fixed wing aircraft operation.

In addition to these academic contributions, in the appendices I include the following three additional contributions to the SAR community:

1. With the development and application of some novel techniques, I was able to greatly improve the imagery from the Characterization of Arctic Sea Ice Experiment (CASIE) [6].

This required aligning an unsynchronized motion record to the SAR data, correcting GPS biases, and using the GPU backprojection method described in Chapter 3.

2. A description and example processing methods for the CASE SAR data, which is published on the BYU MERS website. This is one of the few publicly available SAR data sets, and to our knowledge the only available dataset that is from an LFM-CW SAR or collected on a UAV.

3. The design and implementation of an advanced digital receiver and controller subsystem for an LFM-CW SAR. This system is highly configurable for a variety of operations including airborne, UAV, and land based operation.

1.2 Outline

Following this introductory chapter this work is organized as follows: Chapter 2 provides an overview of the concepts needed to understand this work, including a review of the terminology and geometry for airborne SAR imaging and an introduction to LFM-CW SAR. Chapter 3 describes the design and implementation of a real-time SAR backprojection processor that utilizes unique features of an NVIDIA GPU. Chapter 4 explores the requirements for applying interferometric techniques to data collected with a single antenna from a flight with oscillatory motion. Chapter 5 presents an autofocus method for low altitude SAR when the geometry parameters are not well known. Chapter 6 presents the derivation and implementation of a novel SAR processing algorithm called the Dually Factorized Backprojection (DFBP) method. And finally, Chapter 7 provides a conclusion and suggestions for future work. Additionally three appendices are included that respectively describe improved processing of the CASIE data, a sample CASIE data set and processing code, and the design of the digital receiver and control subsystem for the microASAR.

Chapter 2

Background

This section provides the background material necessary to understand the concepts later described in this work. Specifically this chapter introduces the fundamentals of synthetic aperture radar (SAR) imaging and then describes the system design of a linear frequency modulated continuous wave (LFM-CW) SAR system. Lastly traditional time-domain correlation and backprojection algorithms are presented. For a more complete introduction to SAR see [7–13].

2.1 Notation and geometry

First it is useful to establish some notation and review the geometry of SAR. The transmit and receive signals are given by $s_t(\eta, t)$ and $s_r(\eta, t)$ respectively, where the terms η and t are slow-time and fast-time respectively, which is typical of SAR notation. Slow-time varies discretely with each pulse and fast-time repeats over the range $(0, T_p)$, where T_p is the pulse transmit interval. As the data is recorded at regular time intervals and digitized, square brackets are used to denote discrete time, i.e.

$$s[m, n] = s(mT_p, n/f_s), \tag{2.1}$$

where m is the slow-time index, n is the fast-time index and f_s is the sampling frequency.

The geometry of a typical SAR data collection is illustrated in Fig. 2.1. Because SAR data collections are from a straight path, the geometry for a SAR data collection is typically described in two dimensions: azimuth angle and slant range, which are frequently referred to as azimuth and range respectively. The arrow extending from the nose of the aircraft denotes the direction of the flight and is referred to as the azimuth direction. The azimuth angle

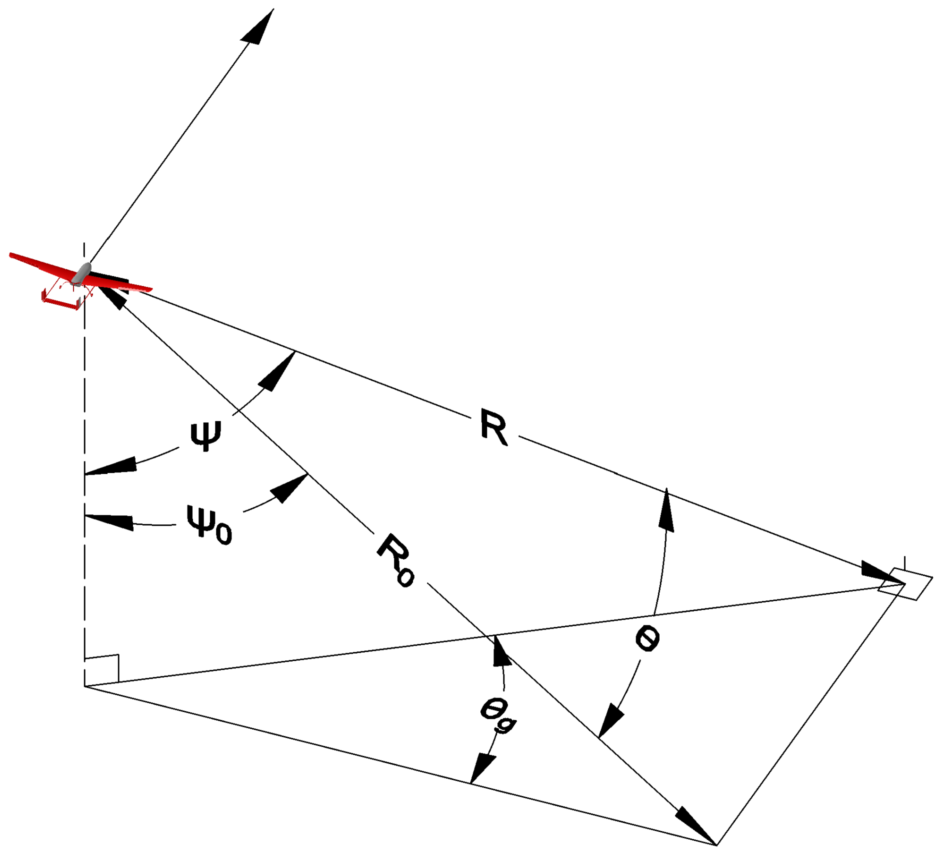


Figure 2.1: Geometry of a typical SAR data collection. The terms R, R_0, θ, θ_g, ψ, and ψ_0 respectively denote the range, slant range, azimuth angle, ground azimuth angle, incidence angle, and broadside incidence angle.

is measured with reference to plane orthogonal to the direction of flight as denoted by θ in Fig. 2.1. For a single pulse the range is given by the Euclidean distance from the antenna as denoted by R in the illustration. The slant range is given by the distance orthogonal to the flight direction, denoted R_0 in Fig. 2.1. The reduced geometry is useful because for a straight flight all objects with the same slant range are viewed over the same range of ranges during the collection. These concepts are discussed further in the following sections.

2.2 Introduction to airborne radar imaging

Synthetic aperture radar is an advanced form of airborne radar imaging. Similar to how a bat calls out and listens to the echoes in order to guide its flight, a radar transmits an electro-magnetic signal and receives the backscattered signal from objects within the antenna illumination. The received signal can be described as an integral of attenuated, time delayed copies of the transmit signal, written as

$$s_r(\eta, t) = \int_X I(\mathbf{x}) A_{\mathbf{x}}(\eta, t) s_t(\eta, t - \tau_x(\eta, t))\, dx, \tag{2.2}$$

where I is the imaging scene, $A_{\mathbf{x}}$ is an amplitude function due to the antenna pattern, incidence angle, the distance to the scatterer, $\tau_{\mathbf{x}}(\eta, t)$ is the round-trip propagation time of the radar signal, which consist of the time for the pulse to travel from the transmit antenna to the position $\mathbf{x}$ plus the time for the backscatter to radiate back to the receive antenna on the aircraft, and X is the area to be imaged. Because the motion of the aircraft is many orders of magnitude slower than the speed of light, c_0, and the transmit and receive antennae are only separated by a small distance, the propagation time is approximated as

$$\tau_{\mathbf{x}}(\eta, t) = \frac{2R}{c_0} = \frac{2\|\mathbf{p}(\eta, t) - \mathbf{x}\|}{c_0}, \tag{2.3}$$

where $\mathbf{p}$ is the position of the radar.

First consider a simple radar that transmits a signal of the form

$$s_n(\eta, t) = \exp\left\{-j\left(2\pi f_0 + \phi(\eta, t)\right)\right\} \Pi(t/T), \tag{2.4}$$

where f_0 is the carrier frequency, Π is the rect function given by

$$\Pi(t) = \begin{cases} 0 & \text{if } |t| > \frac{1}{2} \\ 1 & \text{if } |t| \leq \frac{1}{2} \end{cases}, \tag{2.5}$$

and $\phi(\eta, t)$ is a random phase term which may vary with time. The phase term can be suppressed by taking the magnitude of a single pulse. Because the phase of the signal is ignored this type of radar is called non-coherent. From a single pulse a radar can only measure the distance to objects within the antenna illumination and the strength of the cumulative backscatter from all the objects at that range within the antenna beamwidth. An image can be created using a narrow beam antenna pointed to the side of an aircraft and repeatedly transmitting pulses and stacking the amplitude of the received echoes into a two dimensional array. This type of radar is called side-looking airborne radar (SLAR) and is the predecessor of SAR.

The quality of any radar image is typically measured by the resolution and signal to noise ratio (SNR). The resolution is determined by how far apart two point targets need to be separated in order to be resolved separately. For simple SLAR system with the transmit signal given by Eq. (2.4), the azimuth resolution is limited to the beamwidth of the antenna, and the range resolution is determined by the length of the pulse. Outside of the efficiency of the antennae, the SNR is largely determined by energy in the transmit signal, which is approximately given by the product of the peak transmit power and the length of the transmitted pulse. The difficulty with SLAR is that in order to achieve high resolution and high SNR a very long antenna and very powerful transmitter are required.

SAR systems overcome the challenges of the previous system by ensuring that the system is coherent, which means that either $\phi(\eta, t)$ is known and can be accounted for or that it is stationary. To alleviate the need for a very powerful transmitter, a SAR system uses a relatively long modulated pulse. The most common modulation scheme is linear frequency modulation (LFM) described as

$$s_t(\eta, t) = \exp\left\{-j\left(2\pi f_0 t + \pi k_r t^2 + \varphi\right)\right\} \Pi(t/T_l), \tag{2.6}$$

where k_r is the chirp rate, T_l is the pulse length, and φ describes the random phase, which for this discussion is considered stationary. With a LFM signal, often called a chirp, the range resolution is determined by the bandwidth of the transmit signal and for a narrow band signal is given by

$$\Delta_r = \frac{c_0}{2B_w}, \tag{2.7}$$

where B_w is the bandwidth of the modulation given by

$$B_w = k_r T_p. \tag{2.8}$$

With a coherent receiver, a SAR system is able to accurately combine overlapping pulses to achieve very high resolution in the azimuth direction. The azimuth resolution is limited by the extent of the azimuth viewing angles, approximately written as

$$\Delta_a = \frac{c_0}{2f_c\theta_0} = \frac{\lambda_c}{2\theta_0}, \tag{2.9}$$

where λ_c is the wavelength of the center frequency and θ_0 is the azimuth angle extent over which a point in the image scene is viewed.

2.2.1 SAR operating modes

The most notable SAR operational modes are spotlight and stripmap [10, 14]. In spotlight SAR the antenna is controlled to point at the same patch of ground during the flight; whereas with stripmap SAR the side-looking antenna has fixed pointing. In spotlight mode processing, objects in the imaging scene are visible during the entire data collection (with the exception of shadowed objects obstructed by other objects in the imaging area). Whereas in stripmap mode, the imaged scene is constantly progressing. This leads to fundamental differences in the processing algorithms for the two operation modes.

Other specialized operating modes exist including interferometric SAR, ScanSAR, TOPSAR, and SweepSAR. The latter three modes are designed to image a large swath width by changing the pointing of the antennae for each azimuth look. As the roots of the word "interferometry" suggest, interferometric SAR uses multiple receive antennae to infer the angle of the source of a received signal. There are two main modes of interferometric SAR, namely along-track and cross-track interferometry. In along-track interferometery two receive antennae are separated by an along-track baseline, and likewise in cross-track interferometry the two antennae are separated in the cross-track. Along-track interferometry allows for the detection and estimation of moving targets such as vehicles or ocean waves. Cross-track interferometry allows for the estimation of topography with very high accuracy.

2.3 LFM-CW SAR

The work in this dissertation is primarily applied to linear frequency modulated continuous wave (LFM-CW) SAR. An LFM-CW SAR is designed to achieve maximum signal to noise ratio (SNR), which improves image quality for a given peak transmit power. This is accomplished by extending the pulse length T_l of Eq. (2.6) to be equal to the pulse transmit interval T_p, so that the radar is continuously transmitting. Typically the rect function is dropped and the LFM-CW transmit signal is written as

$$s_t(\eta, t) = \exp\left\{-j\left(2\pi f_0 t + \pi k_r t^2 + \varphi\right)\right\}. \tag{2.10}$$

Fig. 2.2 shows a high-level flow diagram for a typical LFM-CW SAR system. The frequency

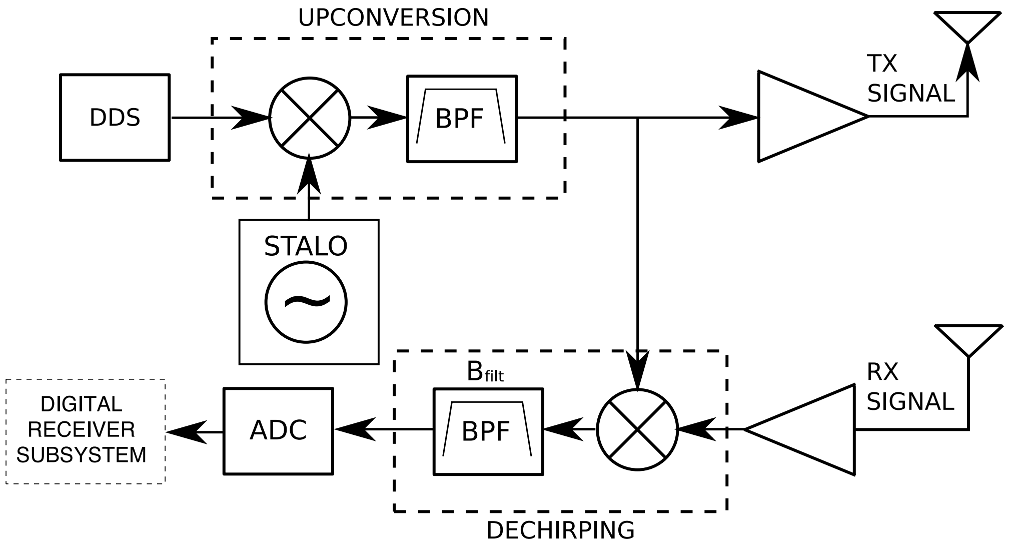


Figure 2.2: (a) A high-level flow diagram for a typical homodyne LFM-CW SAR system.

modulated chirp is generated via a direct digital synthesizer (DDS) or by a voltage controlled oscillator (VCO) and mixed up to a carrier frequency and transmitted. Unlike traditional pulsed SAR systems, an LFM-CW SAR receiver is on during transmit. Although a separate antenna is typically used for the receive channel, the non-ideal isolation of the physical

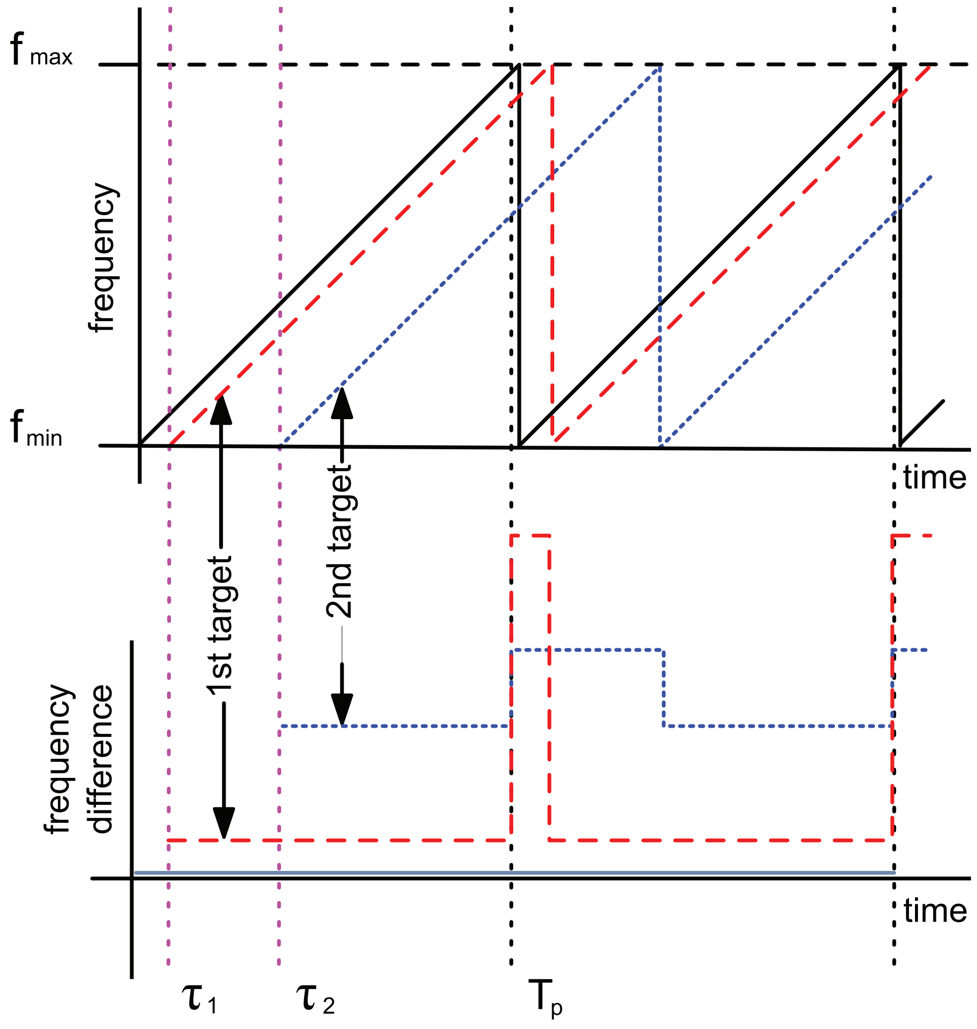


Figure 2.3: A time frequency representation of LFM-CW operation and the dechirped signal.

transmit and receive channels (in the RF system and the antennas) introduces feed-through of the transmit signal that dominates the radar echoes.

To remove the feed-through and to reduce the bandwidth of the echoes, the received signal is “dechirped”. Dechirping consists of mixing the received signal with the transmit signal. The resulting signal contains the sum and difference frequencies of the transmit and receive signals. As shown in Fig. 2.2, using a bandpass filter the sum frequencies and the feed-through are removed and the resulting signal can be represented by

$$\begin{aligned} s_{dc}(\eta,t) &= r_{\mathbf{x}}(\eta,t)s_t^*(\eta,t) \\ &= A_{\mathbf{x}}(\eta,t)\sigma_{\mathbf{x}}\exp\{j\left(2\pi k_r\tau_{\mathbf{x}}(\eta,t)t-\pi k_r\tau_{\mathbf{x}}(\eta,t)^2+2\pi f_0\tau_{\mathbf{x}}(\eta,t)\right)\}. \end{aligned} \tag{2.11}$$

Because LFM-CW SAR is typically operated from low altitude, the imaging scene is narrow enough that the dechirped signal occupies much less bandwidth than the transmit signal. This enables a significant reduction in the data storage requirements and the required speed of the instrument's analog to digital converter (ADC).

Fig. 2.3 illustrates the transmit, receive and dechirp signals in a time versus frequency plot. The transmit chirp, represented by the solid line, starts at the minimum frequency and increases linearly with a slope given by the chirp rate (k_r) up to the maximum frequency and then drops to the minimum frequency. The echoes are copies of the transmit signal occupying the same range of frequencies but are delayed in time. The process of dechirping translates the time delay into a frequency difference, as shown in the lower portion of Fig. 2.3. Because the chirp rate determines the frequency separation of targets, the selection of the PRF is directly connected to the width of the bandpass filter and the ADC.

2.4 Time-domain backprojection

Traditionally LFM-CW stripmap SAR image processing is performed with frequency domain algorithms such as the Range Doppler Algorithm (RDA), the Frequency Scaling Algorithm (FSA), and the Ω-k algorithm [9, 10, 15, 16]. Frequency domain algorithms use the assumption that the aircraft does not deviate from a straight line in order to efficiently perform convolution operations with FFTs. There are a number of modifications to these algorithms that compensate for some non-linear motion [17–19], but these methods complicate the original algorithms and their performance has limitations [13]. In contrast, the SAR backprojection algorithm described below is a time-domain algorithm that inherently compensates for non-linear flight paths and surface topography. The improved focusing capability of backprojection has been noted in a number of studies [20–22]; however, backprojection increases the computational burden of generating SAR images.

In the following, the development of a backprojection method for stripmap LFM-CW SAR is given following the development described in [4]. The received signal can be written as the integral of the dechirped signal, Eq. (2.11), from a single target over the imaging area

$$s(\eta, t) = \int_R I(\mathbf{x}) A_{\mathbf{x}}(\eta, t) e^{j(2\pi k_r t \tau_{\mathbf{x}}(\eta,t) + \tau_{\mathbf{x}}(\eta,t) 2\pi f_0 - \pi k_r \tau_{\mathbf{x}}^2(\eta,t))} \, dx, \tag{2.12}$$

where R is the imaging area, and k_r is the chirp rate. The simplest method for reconstructing the SAR image, albeit highly computationally demanding, is discrete time-domain correlation, described as

$$\hat{I}(\mathbf{x}) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} W_{\mathbf{x}}[m,n] s[m,n] s_{\mathbf{x}}^{*}[m,n], \tag{2.13}$$

where $\hat{I}(\mathbf{x})$ is the reconstructed SAR image, $s_{\mathbf{x}}$ is the reference signal for a target at pixel location $\mathbf{x}$, W is an apodization window, N is the number of samples in a pulse, and M is the number of pulses over which the backprojection is calculated. To simplify, a phase-only reference signal is used yielding

$$\hat{I}(\mathbf{x}) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} W_{\mathbf{x}}[m,n] s[m,n] \cdot \exp\{-j\left(2\pi k_r t_n \tau_{\mathbf{x}}[m,n] + 2\pi f_0 \tau_{\mathbf{x}}[m,n] - \pi k_r \tau_{\mathbf{x}}^2[m,n]\right)\}. \tag{2.14}$$

Using minimal approximations Eq. (2.14) generates the ideal reconstructed image; however it is computationally intensive, on the order of $O(MNL_aL_r)$ where L_a and L_r are respectively the number of azimuth and range pixels in the image, M is the number of pulses, and N is the number of samples in each pulse.

To reduce computation the pulses are first range-compressed. In order to perform range compression, the range from the radar to the target is approximated as stationary during a pulse via the stop-and-hop approximation, i.e.

$$\tau_{\mathbf{x}}[m,n] \approx \tau_{\mathbf{x}}[m,0]. \tag{2.15}$$

By selecting an apodization window that can be separated into range and azimuth windows W^a and W^r, Eq. (2.14) can be written as

$$\hat{I}(\mathbf{x}) \approx \sum_{m=0}^{M-1} W_{\mathbf{x}}^{a}[m] \exp\{-j\phi_\alpha[m]\} \cdot \sum_{n=0}^{N-1} W_{\mathbf{x}}^{r}[n] s[m,n] \cdot \exp\{-j\phi_r[n]\}, \tag{2.16}$$

where

$$\phi_\alpha[m] = 2\pi f_0 \tau_{\mathbf{x}}[m,0] - \pi k_r \tau_{\mathbf{x}}^2[m,0], \tag{2.17}$$

and

$$\phi_r[n] = 2\pi k_r t_n \tau_{\mathbf{x}}[m, 0]. \tag{2.18}$$

The latter summation in Eq. (2.16) describes a Fourier transform yielding

$$\hat{I}(\mathbf{x}) \approx \sum_{m=0}^{M-1} W_{\mathbf{x}}^a[m] \exp\left\{-j\left(\phi_\alpha[m]\right)\right\} \cdot S_m\left(\frac{N}{f_s}\left(k_r \tau_{\mathbf{x}}[m, 0]\right)\right), \tag{2.19}$$

where S_m is the Fourier transform of $W_r[n]s[m, n]$, which is the range compressed data. Range compression is accomplished via a Fourier transform of the dechirped data because in the dechirped data each scatterer is represented by a single frequency related to the time-delay using the stop-and-hop assumption. Using the range-compressed data instead of the dechirped data greatly reduces the computational complexity to the order of $O(ML_aL_r + MN\log_2(N))$.

2.5 Summary

This ends the review of the fundamental concepts of SAR imaging that are necessary to understand the remainder of this work. For a more thorough and detailed introduction see [7–13].

Chapter 3

GPU Backprojection for UAV Based SAR

This chapter presents the development, implementation, and analysis of a real-time GPU based SAR backprojection processor for UAV based SAR. While several implementations of backprojection on GPUs have been described in the literature, most notably [23–25] and [21], these papers focus on the simplest form of spotlight mode SAR backprojection and are not directly applicable to stripmap SAR. This chapter begins with an introduction into the NVIDIA GPU architecture. This is followed by a new derivation of backprojection that accounts for the motion of the aircraft during the pulse and for the antenna pointing variation for stripmap SAR. This new derivation enables radiometrically accurate backprojection SAR imaging. Then a discussion of the interpolation of the range compressed data is given. It is demonstrated that compared to traditional polynomial interpolation paired with zero-padding, the non-equispaced result fast Fourier transform (NERFFT) provides superior interpolation accuracy for interpolating bandlimited signals like the dechirped SAR data. Then the implementation of the backprojection processor on an NVIDIA GPU is presented. In order to achieve accurate and real-time processing, it is shown how the hardware linear interpolation provided by the GPU's texture cache can be used to accelerate the NERFFT. Finally resulting imagery processed using this processor is presented and analyzed. It is shown that processor is faster than real-time for the CASIE data. It is also shown using simulated data that the derived correction for the motion of the aircraft during the pulse accurately reconstructs the image.

3.1 GPU architecture

GPUs have anywhere from tens to thousands of processing cores, enabling the fast and efficient processing of billions of math operations per second on a single device. While

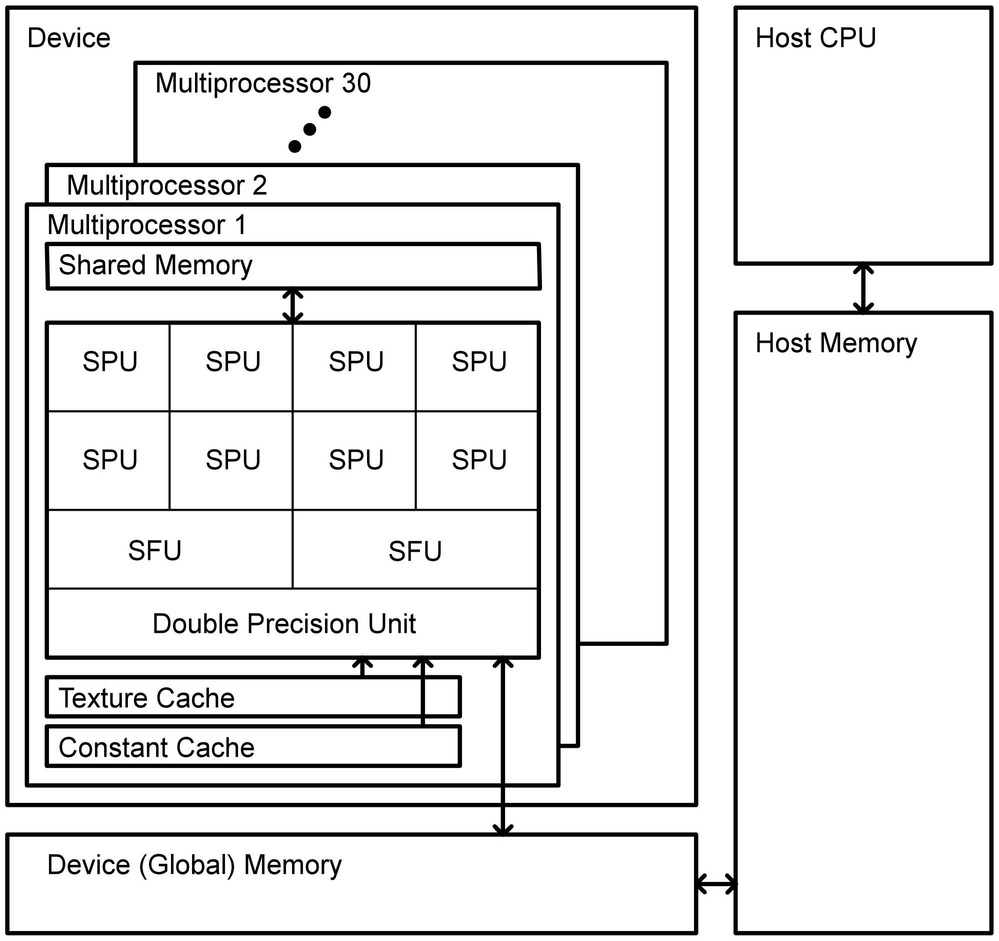


Figure 3.1: Illustration of the NVIDIA GTX 285 GPU architecture.

there are several different GPU architectures, the key aspects important to our discussion can be found in nearly all recent GPUs. As an illustration we use an NVIDIA GTX 285 GPU whose architecture is shown in Fig. 3.1. An NVIDIA GPU is broken into groups of processing cores which NVIDIA calls multiprocessors. The number of multiprocessors is dependent on the particular GPU. Multiprocessors contain three types of sub-processors including single precision units (SPU), special function units (SFU), and double precision units (DPU). Floating point operations including multiply, add, and fused multiply-add are performed by the SPUs and DPUs for 32-bit and 64-bit values respectively. The SFUs perform 32-bit transcendental operations such as inverse, inverse square root, sine, and cosine. The 64-bit transcendental operations are performed by software routines. The number of each type of processing units is also dependent on the GPU. Each multiprocessor includes an instruction unit which controls which processing units operate at a given time. All of the operating units execute the same instruction on different data, much like a single instruction multiple data (SIMD) architecture, but because the instruction unit controls which units are operating, the processing streams, or "threads", are allowed to branch independently.

In terms of memory, the GPU device has a large amount of dedicated off-chip memory termed global memory. Each multiprocessor has a large bank of 32-bit registers, a block of shared memory, and constant and texture caches. These resources are shared among the processors within a multiprocessor. It is important to understand the types of memory available because with the vast amount of compute power of the GPU, the achievable performance gains are frequently determined by the memory accesses of the threads, because memory accesses are typically serial operations and cannot benefit from pipelined architectures. Table 3.1 illustrates the latency and throughput of different GPU operations, from which it is clear that global memory accesses are drastically more expensive than other operations. The cost of global memory accesses can be minimized by performing global memory accesses infrequently by rearranging the program structure to copy data from global memory to local shared memory in large coalesced reads and vice versa for writes to global memory [26].

Table 3.1: Latency of GPU operations in number of clock cycles for the NVIDIA T10 architecture (GTX 285) [27].

Operation	Latency (cycles)	Throughput (ops/cycle)
multiply	24	11.2
add,sub,max,min,mad	24	7.9
divide	137	1.5
square-root	56	2.0
sine,cosine	48	2.0
shared memory access	38	N/A
global memory access	$\sim$ 436-443	N/A
texture cache read	261	N/A

The texture cache is a unique GPU feature. It provides read-only access to global memory, and is optimized for spatial locality for one, two, or three dimensional arrays. It has hardware built in to provide linearly-interpolated values with virtually no performance cost. The linear interpolation provided by the texture cache is performed using only 8 bits of fractional precision, but it is shown later in this chapter how using the texture cache as part of the NERFFT provides higher accuracy than cubic interpolation with full double precision arithmetic for the interpolation of the SAR signal. The method used to incorporate the texture cache as part of the NERFFT can also be applied to accelerate other interpolation methods and is explained later.

In order to utilize the compute capabilities of the GPU, NVIDIA provides the Compute Unified Device Architecture (CUDA) which augments the C programming language. CUDA adds a kernel construct which ties a groups of threads to multiprocessors, where each thread executes the same function virtually in parallel. Each thread maintains its own program counter and branches can diverge during execution. If the branches diverge the independent branches are executed serially. In this case the processor is not fully utilized. Thus it is important to avoid branch divergence to achieve optimal performance.

3.2 Backprojection processing for UAVs

In this section we develop a SAR backprojection method that accounts for platform motion during the pulse and the attitude changes of the antenna. The method is suitable for UAV based stripmap SAR.

In LFM-CW SAR, the motion of the aircraft during a pulse can be significant as noted by [4] and [28]. In this section a backprojection method is developed for stripmap LFM-CW SAR that includes an extension for ultra-wide-band (UWB) SAR. This derivation follows a similar methodology to [4]. The received signal can be written as the integral of the dechirped signal, Eq. (2.11), from a single target over the imaging area

$$s(\eta, t) = \int_R I(\mathbf{x}) A_{\mathbf{x}}(\eta, t) e^{j(2\pi k_r t \tau_{\mathbf{x}}(\eta,t) + \tau_{\mathbf{x}}(\eta,t) 2\pi f_0 - \pi k_r \tau_{\mathbf{x}}^2(\eta,t))}\, dx, \tag{3.1}$$

where R is the imaging area, k_r is the chirp rate, and I is the image scene. The simplest method for reconstructing the SAR image, albeit highly computationally demanding, is discrete time-domain correlation, described as

$$\hat{I}(\mathbf{x}) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} W_{\mathbf{x}}[m,n]s[m,n]s_{\mathbf{x}}^{*}[m,n], \tag{3.2}$$

where $\hat{I}(\mathbf{x})$ is the reconstructed SAR image, $s_{\mathbf{x}}$ is the reference signal for a target at pixel location $\mathbf{x}$, W is an apodization window, N is the number of samples in a pulse, and M is the number of pulses over which the backprojection is calculated. Square braces are used to indicate discrete samples such that

$$s[m,n] = s(\eta_m, t_n) = s(mT_p, n/f_s), \tag{3.3}$$

where T_p is the pulse length and f_s is the ADC sampling frequency. To simplify we use a phase-only reference signal yielding

$$\hat{I}(\mathbf{x}) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} W_{\mathbf{x}}[m,n]s[m,n]\cdot \exp\{-j\left(2\pi k_r t_n \tau_{\mathbf{x}}[m,n] + 2\pi f_0 \tau_{\mathbf{x}}[m,n] - \pi k_r \tau_{\mathbf{x}}^2[m,n]\right)\}. \tag{3.4}$$

Using no approximations Eq. (3.4) generates the ideal reconstructed image; however it is computationally intensive, on the order of $O(MNL_I)$ where L_I is the number of pixels in the image, M is the number of pulses, and N is the number of samples in each pulse.

To reduce computation we first range-compress the samples. In order to perform range compression, the range from the radar to the target is traditionally approximated as stationary during a pulse. However, because LFM-CW SAR uses a much longer pulse we approximate $\tau_{\mathbf{x}}$ as having linear motion during the pulse i.e.

$$\tau_{\mathbf{x}}[m,n] \approx \tau_{\mathbf{x}}[m,0] + v_{\mathbf{x}}[m]t_n, \tag{3.5}$$

where

$$\begin{aligned} v_{\mathbf{x}}[m] &= \frac{d\tau_{\mathbf{x}}(\eta_m, 0)}{dt} \\ &= \frac{2\langle \mathbf{v}, \mathbf{p}(\eta_m,0) - \mathbf{x}\rangle}{c_0\|\mathbf{p}(\eta_m,0) - \mathbf{x}\|}, \end{aligned} \tag{3.6}$$

where $\mathbf{v}$ is a vector of the velocity components. By selecting an apodization window that can be separated into range and azimuth windows W^a and W^r, Eq. (3.4) can be written as

$$\hat{I}(\mathbf{x}) \approx \sum_{m=0}^{M-1} W^a_{\mathbf{x}}[m] \exp\{-j\phi_\alpha[m]\} \cdot \sum_{n=0}^{N-1} W^r_{\mathbf{x}}[n] s[m,n] \cdot \exp\{-j\phi_r[n]\}, \tag{3.7}$$

where

$$\phi_\alpha[m] = 2\pi f_0 \tau_{\mathbf{x}}[m,0] - \pi k_r \tau^2_{\mathbf{x}}[m,0], \tag{3.8}$$

and

$$\phi_r[n] = 2\pi k_r t_n \tau_{\mathbf{x}}[m,0] + 2\pi k_r v_{\mathbf{x}}[m] t_n^2 + 2\pi f_0 v_{\mathbf{x}}[m] t_n - \pi k_r (2\tau_{\mathbf{x}}[m,0] v_{\mathbf{x}}[m] t_n + v^2_{\mathbf{x}}[m] t_n^2). \tag{3.9}$$

The t_n^2 terms in Eq. (3.9) prevent us from replacing the second summation with the range-compressed data. Ribalta's derivation accomplishes this task by noting that the linear terms $2\pi k_r t_n \tau_{\mathbf{x}}$ and $2\pi f_0 v_{\mathbf{x}} t_n$ dominate the summation in many imaging scenarios, and discards the other terms. However, to improve upon this approximation we instead use the least squares approximation

$$t_n^2 \approx T_p t_n - \frac{T_p^2}{6}. \tag{3.10}$$

The error induced by the approximation in Eq. (3.10) is minimal for low-altitude aircraft with typical speeds. Using Eq. (3.10) in Eq. (3.7) yields

$$\begin{aligned}\hat{I}(\mathbf{x}) \approx \sum_{m=0}^{M-1} & W^a_{\mathbf{x}}[m] \exp\left\{-j\left(\phi_\alpha[m] - \pi k_r v^2_{\mathbf{x}}[m]\frac{T_p^2}{6} + 2\pi k_r v_{\mathbf{x}}[m]\frac{T_p^2}{6}\right)\right\} \\ & \cdot S_m\left(\frac{N}{f_s}\left(k_r \tau_{\mathbf{x}}[m,0] + f_0 v_{\mathbf{x}}[m] + k_r T_p v_{\mathbf{x}}[m] - \frac{k_r}{2} v_{\mathbf{x}}[m](2\tau_{\mathbf{x}}[m,0] + T_p v_{\mathbf{x}}[m])\right)\right),\end{aligned} \tag{3.11}$$

where S_m is the Fourier transform of $W_r[n]s[m,n]$, which is the range compressed data. Range compression is accomplished via a Fourier transform of the dechirped data because each object is represented by a single frequency related to the time-delay using the stop-and-hop assumption in the dechirped data, Eq. (2.11). Using the range-compressed data instead of the dechirped data greatly reduces the computational complexity to the order of $O(ML_I + MN\log_2(N))$.

By analyzing a point at the far end of the range swath for a particular radar's imaging parameters, further simplifications can be made. For example the fourth term phase is only beneficial for extremely high-bandwidth and long pulses which are difficult to realize in practice, and the third phase term and the last term on the range index can be neglected because $v_x << 1$, which is a ratio of the aircraft speed and the speed of light. The third range index is only beneficial at extreme ranges that are not typical in LFM-CW imaging. With these small approximations Eq. (3.11) simplifies to

$$\hat{I}(\mathbf{x}) \approx \sum_{m=0}^{M-1} W_{\mathbf{x}}^{a}[m] \exp\left\{-j\phi_{\alpha}[m]\right\} S_m \left(\frac{N}{f_s}\left(k_r \tau_{\mathbf{x}}[m,0] + f_0 v_{\mathbf{x}}[m] + k_r T_p v_{\mathbf{x}}[m]\right)\right). \quad (3.12)$$

The first term in indexing S_m is found in traditional backprojection methods and the second term partially compensates for the motion during the pulse and was identified by Ribalta [4]. The third term, which we call the UWB correction, is not included in Ribalta's study; however,we show that in ultra-wide-bandwidth operation the third term can be significant.

3.2.1 Compensated backprojection

The phase-only backprojection described in Eq. (3.11) works well for data sets collected on platforms where the antenna attitude is fairly constant, such as satellites and large aircraft; however, a small UAV is very susceptible to external forces often resulting in undulating motion. Thus some pixels are calculated using more samples and different antenna gains than others resulting in SAR images with varying intensity. In this discussion we do not address the effects of incidence angle which are addressed in detail in [29].

In order to remove the variation of intensity due to the motion and attitude of the aircraft we examine the backprojection filter. The complex gain $A_{\mathbf{x}}$ from Eq. (2.11) is given by the radar equation and can be written as

$$A_{\mathbf{x}} = \left(B \frac{G_{\mathbf{x}}^2(\eta)}{\|\mathbf{p}(\eta) - \mathbf{x}\|^4} \sigma_{\mathbf{x}}^0\right)^{1/2}, \quad (3.13)$$

where B is a constant that accounts for all the radar parameters that are independent of flight and position such as the transmit power, receiver gain, system losses, etc., $G_{\mathbf{x}}(\eta)$ is the antenna gain (assuming mono-static operation), and $\sigma^0_{\mathbf{x}}$ is the normalized backscatter. Using Eq. (3.13) we relate the magnitude of Eq. (3.12) to the normalized backscatter as

$$\sigma^0 \approx \frac{|\hat{I}(\mathbf{x})|^2}{B(\sum_m W_a(m)\frac{G^2_{\mathbf{x}}(\eta)}{\|\mathbf{p}(\eta)-\mathbf{x}\|^4})}. \tag{3.14}$$

If all of the system parameters are precisely known this derivation produces fully-calibrated images and even when the parameters are not fully known the correction is often good within a scale factor.

3.2.2 Interpolation

Note that Eq. (3.4) uses continuous values of the range-compressed data; however, because the range-compressed data is discrete, the data must be interpolated. The selection of the interpolation method can dramatically affect both the speed and accuracy of the processor. In order to produce high-quality images, backprojection implementations use some form of interpolation; although it is not always explicitly stated. Ideal interpolation is achieved using the DTFT of the dechirped data or Dirichlet interpolation of the discrete range-compressed values. Both of these approaches are computationally demanding. Because of the simplicity of implementation, some backprojection methods use linear interpolation [23] while others significantly zero-pad the range-compression FFT to approximately apply Dirichlet interpolation at discrete points [20, 24]. Combining zero-padding and polynomial interpolation methods increases the accuracy; however, the non-equispaced result FFT (NERFFT) provides even better performance. The NERFFT is related to the non-uniform FFT (NFFT or NUFFT). The NERFFT applies a window to the dechirped data prior to using the FFT such that the DTFT results can be obtained using only a small number of range-compressed samples [25,30]. The NERFFT and an efficient implementation of it on GPUs is explored in Section 3.3.

3.2.3 Sub-Aperture processing

Another important function of a SAR processor is to provide multi-looking to reduce speckle. In multi-look processing, sub-aperture images are created using subsets of the data partitioned by the azimuth angle or Doppler frequency. The subsets can be created using azimuth filters, but, as discussed in the next section, in backprojection processing it is more effective to apply azimuth windows. The final multi-looked image is created by power summing the sub-aperture images. Each sub-aperture image is effectively created with a narrower beamwidth than the physical antenna beamwidth resulting in a lower-resolution image. With the separate sub-aperture images formed from different portions of the SAR data, the speckle statistics are independent for non-overlapping sup-apertures. Averaging the magnitude of the sub-aperture images reduces the image noise and makes it easier for users to interpret the images [13].

3.3 NERFFT

The NERFFT reconstructs a non-equispaced Fourier transform using a windowed and zero-padded discrete Fourier transform [25, 30]. The key principle of the NERFFT is that the complex exponential in the DFT summation can be rewritten as

$$e^{-ix\omega} = \frac{1}{\sqrt{2\pi}\phi(x)} \sum_{m\in\mathcal{Z}} \hat{\phi}(\omega - m)e^{-im\omega}, \tag{3.15}$$

where ϕ and $\hat{\phi}$ are a window function and its Fourier transform, and $\mathcal{Z}$ is the support of $\hat{\phi}$. Using Eq. (3.15), the interpolated range-compressed data can be described as

$$\begin{aligned} S(\omega) &= \frac{1}{\sqrt{2\pi}} \sum_{m\in\mathcal{Z}} \hat{\phi}\left(\omega - \frac{2\pi m}{cN}\right) \sum_{k=-N/2}^{N/2-1} e^{-im\frac{2\pi k}{cN}} \frac{s[k]}{\phi(\frac{2\pi k}{cN})} \\ &= \frac{1}{\sqrt{2\pi}} \sum_{m\in\mathcal{Z}} \hat{\phi}\left(\omega - \frac{2\pi m}{cN}\right) S_c\left[\lfloor\frac{\omega c N}{2\pi}\rfloor + m\right], \end{aligned} \tag{3.16}$$

where S_c is the FFT of $\frac{s[k]}{\phi(\frac{2\pi k}{cN})}$ with a zero-padding factor c. A proof of Eq. (3.16) is given in [30]. To illustrate, let ϕ be a rect window so that $\hat{\phi}$ is the Dirichlet kernel. Eq. (3.15) then yields Dirichlet interpolation. Note that the summation over k in Eq. (3.16) is centered around 0 denoting a centered FFT. This both simplifies the development and results in both ϕ and $\hat{\phi}$ being real, which reduces the number of multiplies required; however, most software routines for the FFT perform an uncentered FFT. A centered FFT can be performed by circularly shifting the input data to an uncentered FFT.

In Eq. (3.16), a Kaiser-Bessel window is used to concentrate the energy of the signal into a short summation over m. The Kaiser-Bessel window and its Fourier transform are given by

$$\phi(x) = \frac{I_0(K\sqrt{\alpha^2 - x^2})}{I_0(K\alpha)}, \tag{3.17}$$

$$\hat{\phi}(\omega) = \begin{cases} \sqrt{\frac{2}{\pi}} \frac{\sinh\left(\alpha\sqrt{K^2-\omega^2}\right)}{\sqrt{K^2-\omega^2} I_0(K\alpha)}, & \omega \leq \alpha \\ 0, & \omega > \alpha, \end{cases} \tag{3.18}$$

where

$$\alpha = \pi(2 - 1/c) - 0.01, \tag{3.19}$$

I_0 is the zeroth order modified Bessel function of the first kind, and $2K$ is the length of the support of $\hat{\phi}$. It has been shown by [25] and [30] that in the case $c = 2$, a six point window, $K = 3$, can reduce the interpolation error down to the precision of single floating point arithmetic. In Section 3.3.2 we compare several interpolation methods with a range of zero-padding factors in order to navigate the tradeoffs of memory, speed, and accuracy of the interpolation stage.

3.3.1 Accelerating interpolation using GPU texture cache

In [31], a method to accelerate cubic interpolation on GPUs is presented that casts linear interpolation as part of the cubic interpolation. This concept is extended to be used as part of any convolutional interpolation scheme. The finite fractional precision of the GPU hardware linear interpolation incurs some accuracy loss; however, when used to accelerate the NERFFT, in many cases the resulting accuracy is much better than other common interpolation methods, including nearest neighbor, linear, and cubic interpolation.

A convolutional interpolation scheme is defined by

$$S_I(x) = \sum_{m=-K+1}^{K} S[\lfloor x \rfloor + m] c_m(h), \tag{3.20}$$

where $h = x - \lfloor x \rfloor$, and the functions $c_m(h)$ are not necessarily linear. In the form of Eq. (3.20), traditional linear interpolation has the coefficient functions

$$c_0(h) = (1-h), \tag{3.21}$$

$$c_1(h) = h. \tag{3.22}$$

Cubic spline interpolation can also be described in this form. In order to take advantage of the GPU hardware linear interpolation we rewrite Eq. (3.20) with linear interpolation as an intermediate step. Consider Eq. (3.20) written as a summation in pairs

$$\begin{aligned} S_I(x) &= \sum_{m=-K+1}^{K} S[\lfloor x \rfloor + m] c_m(h) \\ &= \sum_{r=0}^{K-1} S[\lfloor x \rfloor + \nu] c_\nu(h) + S[\lfloor x \rfloor + \nu + 1] c_{\nu+1}(h), \end{aligned} \tag{3.23}$$

where $\nu = 2r - K + 1$. With some manipulation the pairs can be written in terms of a linear interpolation. First we multiply Eq. (3.23) by a unitary fraction yielding

$$\begin{aligned} S_I(x) &= \sum_{r=0}^{K-1} \frac{c_\nu(h) + c_{\nu+1}(h)}{c_\nu(h) + c_{\nu+1}(h)} \left(S\left[\lfloor x \rfloor + \nu\right] c_\nu(h) + S\left[\lfloor x \rfloor + \nu + 1\right] c_{\nu+1}(h) \right) \\ &= \sum_{r=0}^{K-1} \left(c_\nu(h) + c_{\nu+1}(h)\right) \cdot \left(\frac{c_\nu(h)}{c_\nu(h) + c_{\nu+1}(h)} S\left[\lfloor x \rfloor + \nu\right] \right. \\ &\qquad \left. + \frac{c_{\nu+1}(h)}{c_\nu(h) + c_{\nu+1}(h)} S\left[\lfloor x \rfloor + \nu + 1\right] \right). \end{aligned} \tag{3.24}$$

Equation (3.24) describes two linear interpolations multiplied by a scale factor, written as

$$\begin{aligned} S_I(x) &= \sum_{r=0}^{K-1} \left(c_\nu(h) + c_{\nu+1}(h)\right) \left((1-b)\, S\left[\lfloor x \rfloor + \nu\right] + b S\left[\lfloor x \rfloor + \nu + 1\right] \right) \\ &= \sum_{r=0}^{K-1} C(h) S_L\left(\lfloor x \rfloor + \nu + B(h)\right), \end{aligned} \tag{3.25}$$

where $B(h) = \frac{c_{\nu+1}(h)}{c_\nu(h)+c_{\nu+1}(h)}$ and $C(h) = c_\nu(h) + c_{\nu+1}(h)$. Equation (3.25) effectively applies the window weightings c_m for two memory accesses by shifting the h of the linear interpolation. Using the texture cache's linear interpolation, a pair of memory accesses is aggregated into one, effectively cutting the memory accesses in half and reducing the overall processing time.

3.3.2 Interpolation error comparison

In order to create a SAR backprojection processor that appropriately balances the tradeoff of speed and accuracy we need to select an interpolator that balances computational accuracy and complexity. Up to this point we have discussed the error of the interpolators generally. In this section we quantify the interpolation error of several interpolators for use on a GPU. In order to capture the effects of the GPU hardware interpolation and single precision arithmetic accurately, we use a Monte Carlo approach to compare the RMS error of the interpolation schemes previously discussed.

The goal of the interpolator is to provide both magnitude and phase accurate range-compressed samples at non-integer indices. The data is simulated as a band-limited signal using a Gaussian random number generator. The points at which the range-compressed data is interpolated are simulated with a uniform random number generator, which closely matches the distribution of range samples. We then pass the data and the points to each of the interpolators and calculate the RMS error compared to the DTFT of the data at the points. We average the results of multiple realizations so that the error statistics do not significantly change with more realizations.

Figure 3.2 shows the RMS interpolation error for the interpolators on a log-log plot with the RMS error in dB on the y axis and the zero-padding factor on the x axis. The error of the nearest neighbor, linear, and cubic spline interpolation methods drops log-linearly with the zero-padding factor each with slightly steeper slope, which equates to lower error. The reduction in error for these methods is due to higher zero-padded data being smoother. In contrast the NERFFT error drops quickly with small zero-padding factors and then trends to leveling out. Thus the motivation for higher zero-padding factors is greatly reduced when using the NERFFT. With wider windows (higher K), the NERFFT is considerably more accurate. Note that the number of memory accesses for $K = 1$ and $K = 2$ is equivalent to linear and cubic interpolation respectively. The $K = 1$ NERFFT has lower error than linear interpolation with zero-padding factors less than 4; however, the window applied to the raw data increases the quantization noise introduced during the FFT which limits the achievable accuracy gains for the NERFFT with higher zero padding.

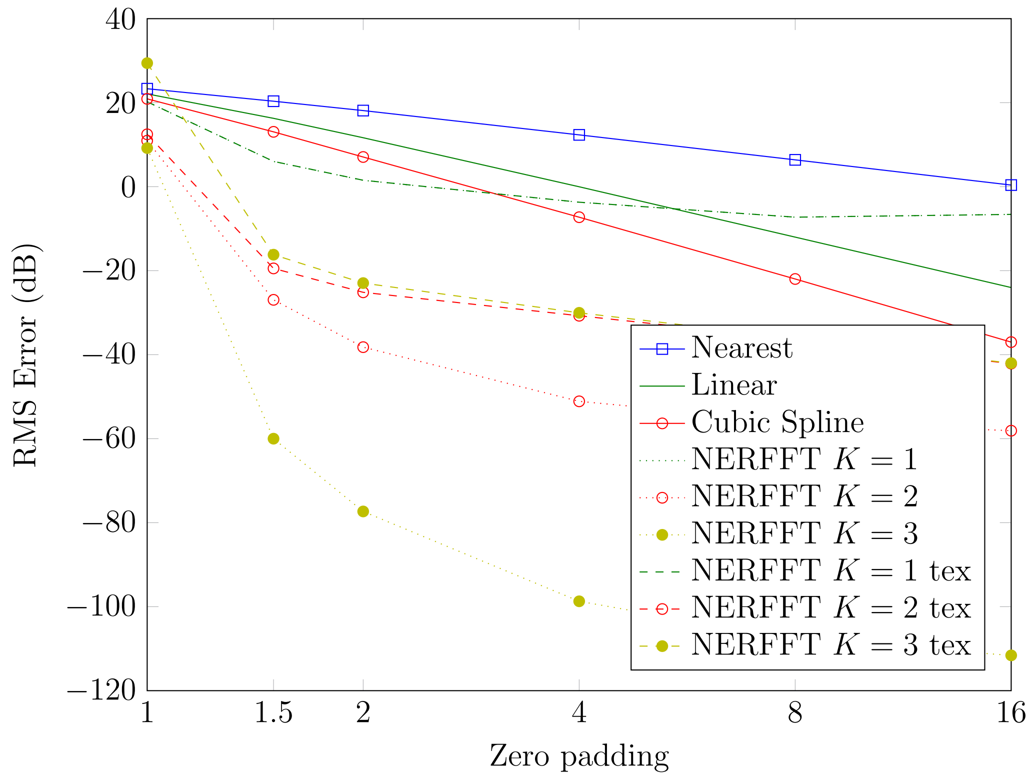


Figure 3.2: RMS error for different range-compressed data interpolation methods with several zero-padding factors. In this plot there are two versions of the NERFFT. The ones denoted "tex" are accelerated using hardware linear interpolation.

Two versions of the NERFFT are used in Fig. 3.2, the latter of them being accelerated using the hardware linear interpolation. The accelerated versions are denoted "tex" in the legend. For the $K = 1$ NERFFT, the accelerated and non-accelerated results are identical. The accelerated $K = 2$ NERFFT has slightly increased error compared to the non-accelerated version, but the accelerated $K = 3$ NERFFT has even slightly higher error than the $K = 2$ NERFFT. This demonstrates the limited accuracy available from the texture cache's linear interpolation. Note that the NERFFT $K = 2$ case is of the same computational complexity as the cubic spline and has dramatically lower error. Using the NERFFT both reduces the error and lowers the memory storage requirements.

3.4 GPU implementation

As shown in Eq. (3.12), backprojection image formation can be independently computed for individual pixels. This makes it easy to parallelize the computation; however, simple parallelization does not lead to a real-time processor in most imaging scenarios. Because the UAV is constantly moving and the antenna is not steered as in spotlight mode SAR, different areas are illuminated during each pulse. As a result, each pixel of backprojection image only needs a subset of the collected pulses in its calculation, but each pixel needs a unique subset. Therefore, if not carefully implemented, the backprojection calculations are performed for pulses that do not contribute to a pixel's final value. The ability to compute the backprojection image in real-time is largely achieved by reducing unnecessary computations.

3.4.1 Overall implementation

The structure of the backprojection processor is outlined in the pseudo-code shown in Algorithm 1. Stripmap SAR collections frequently result in very long strip images. It is neither reasonable nor desirable to process a flight's collection as a single image. Therefore the backprojection processing is broken into multiple images. To begin, we take a section of data of reasonable size resulting in an image that fits in the GPU memory. The image is oriented along the flight direction, allowing efficient use of the GPU thread structure to break up the backprojection calculation.

Each section is broken into small groups or batches of pulses. A window is applied to the dechirped data to provide the desired range apodization and the NERFFT window. Then the FFT is computed using the NVIDIA CUFFT library on the GPU. Concurrently, the CPU is used to calculate the 3-axis heading of the aircraft, simplifying the calculation of the azimuth angle that is used for the apodization window and multi-look processing. The azimuth bins that are visible during the batch are determined using the first and last antenna positions. This information is used in the backprojection kernel to reduce the number of calculations performed for pixels outside of the radar beamwidth during a batch. The size of the batch has considerable effect on the performance of the algorithm. A large batch reduces the number of global memory accesses, but a small batch reduces the number of pixels included in the backprojection calculations that receive negligible contribution from a pulse. Finding the optimal batch size for a given velocity, PRF, and beam-width is found

Algorithm 1 Pseudo-code for GPU stripmap SAR backprojection

```
Break data collection into sections
For each section
   Allocate memory for the image section
   Allocate memory for a batch of pulses
   For each batch of pulses
      Copy batch of pulses to GPU memory
      Apply apodization window
      Apply NERFFT window
      Compute FFT
      Rotate position data
      For each thread group
         Copy position and attitude to shared memory
         Call backprojection kernel
         For each thread
            For each pulse in batch
               Calculate distance and angle
               Calculate expected phase
               Interpolate range—compressed
               For each sub—aperture
                  Calculate azimuth apodization
                  Multiply sample by apodization and phase
                  Accumulate the results
                  Accumulate the apodization and
                     antenna weights
   Copy the image and apodization weights from the GPU
   Save the complex image
   Average the power of the pixels
   Divide by weightings
   Save the multi—looked image
```

with some experimentation. The backprojection kernel is then executed for each batch of pulses.

At the beginning of the backprojection kernel, the antenna positions are copied to the shared memory so that each block of threads only access the global memory for the radar positions once. Using shared memory for variables accessed by multiple threads greatly reduces the kernel runtime. The backprojection kernel calculates the distance and angle from the pixel to each antenna position. The angle is used to calculate azimuth apodization windows for the full aperture and the sub-apertures. The distance is used to calculate the expected phase of each pulse's contribution and to interpolate the range-compressed data. To further reduce the runtime, local variables are used to accumulate the pixel contributions so that writes to global memory only occur once during each batch. The following describes key portions of the backprojection algorithm, including the interpolation and sub-aperture processing steps.

3.4.2 NERFFT implementation

Using the results of the error analysis given in Fig. 3.2, we select the accelerated NERFFT $K = 2$ for the range compression because of its high accuracy, lower memory usage, and potentially lower computational cost. In order to optimally take advantage of the GPU resources described in the background, transcendental functions and memory accesses need to be reduced when possible. In the GPU implementation of the NERFFT described in [25] each NERFFT access requires $2K$ memory accesses and $2K$ evaluations of the hyperbolic sine and square root operations. We reduce the required computations and increase the speed by approximating the combined Bessel functions $C(h)$ and $B(h)$ shown in Eq. (3.25) using polynomials. These approximations effectively cut the memory access and computational burden of the NERFFT in half.

3.4.3 Sub-Aperture processing

It was suggested in [23] that sub-aperture processing can be achieved by running the backprojection kernel multiple times with different sections of the data. While such a method keeps the code simple, it is more efficient to calculate all of the sub-aperture images within the same thread. Calculating the sub-apertures within the same thread avoids duplicate memory accesses and recalculations for overlapping sub-apertures. The inclusion of the sub-apertures in the processing adds another memory layout choice. The sub-aperture image pixels can either be assigned as discrete images or one conglomerate image with the sub-apertures for each pixel occupying a contiguous slot in memory. In our experiments, we find that with a significantly large batch size and a small number of sub-apertures, either layout performs equally well because the writing of the image pixels only occurs once during each batch.

3.5 Results

This section demonstrates the effectiveness of the motion corrected backprojection and the GPU implementation using simulated data and data collected as part of the Characterization of Arctic Sea Ice Experiment (CASIE) [6].

3.5.1 Example CASIE imagery

The CASIE SAR data was collected using the microASAR, a small, LFM-CW, C-band SAR system on board the NASA SIERRA UAV [3,32]. During the CASIE mission the microASAR operated with a 160MHz bandwidth and an 11 degree beamwidth resulting in approximately 1m$\times$15cm single-look resolution. To improve the visual quality of the image we reduce the speckle using 7 sub-apertures with 50 percent overlap. The resulting images are produced with 50cm$\times$50cm pixel spacing. For a more complete description of the CASIE mission and the data processing used see Appendix A and Appendix B.

Figure 3.3 is an example of the imagery produced using the backprojection implementation described in this paper. The data was collected over sea ice in the Arctic ocean north of Svalbard Norway. The UAV flies along the top edge from left to right. The low-altitude of this image results in a very wide range of incidence angles as illustrated on the right side of the image, and the backscatter roll-off due to incidence angle can be seen in the far range.

To illustrate the effectiveness of the compensated backprojection a portion of an image created during a maneuver is shown in Fig. 3.4. In this image the aircraft is finishing a climb from an altitude of 270m to 350m. As with any maneuver, there are significant attitude changes which are typically unaccounted for in traditional SAR processing methods. In contrast to Fig. 3.4a the compensated image in Fig. 3.4c only has a slight residual variation.

The image shown in Fig. 3.3 was processed using a 2008 Mac Pro desktop equipped with 2x3.2 GHz quad-core Intel Xeon processors, 16 GB of memory, and a NVIDIA GTX 285 Video card. The image collection time is 100s. Including the time to load the data and write the image to disk, the backprojection processing takes 85s using nearest neighbor interpolation without the texture cache, 70s for the $K = 2$ NERFFT interpolation, and 60s using the $K = 1$ NERFFT interpolation. The speedup achieved by the NERFFT imple-

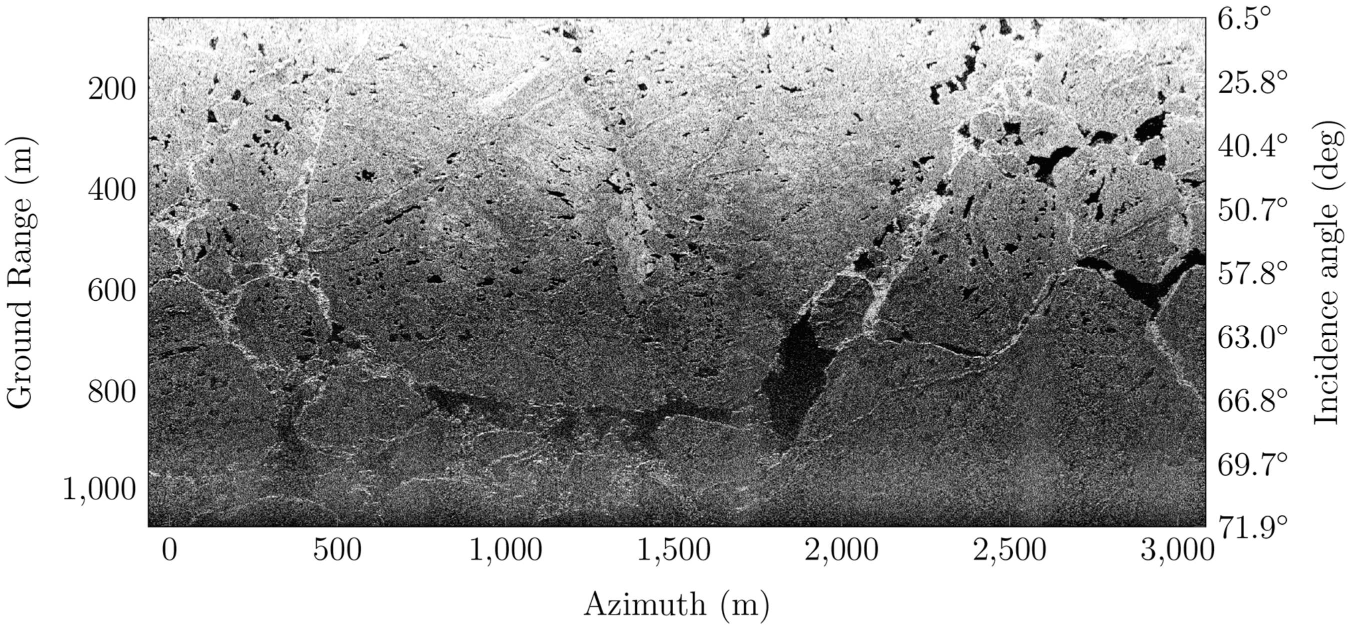
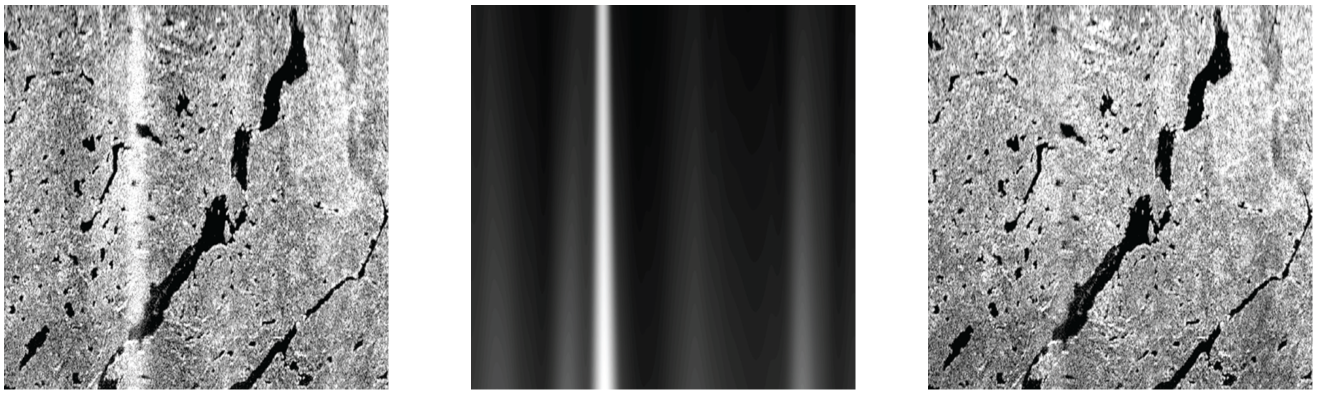


Figure 3.3: An example image of sea ice processed with the GPU SAR backprojection. The left y axis specifies the ground range in meters, and the right y axis is the incidence angle.



(a) **(b)** **(c)**

Figure 3.4: An example of the improvement made using the compensated backprojection: Fig. 3.4a shows the uncompensated image, Fig. 3.4b shows the weighting function, and Fig. 3.4c shows the compensated image.

mentations comes mainly from utilizing the GPU's texture cache and because the NERFFT implementations require less memory allowing for better locality of memory accesses. Note that because the texture cache is used as part of the NERFFT, the NERFFT interpolation effectively requires only one memory access in the $K = 1$ case and two in the $K = 2$ case.

As shown earlier in Fig. 3.2, the implementations using the NERFFT with a zero-padding factor of 2 are more accurate than the nearest interpolation with a zero padding factor of 16 and we see that the runtime for both NERFFT implementations is also better than the nearest neighbor implementation. Also note all of the GPU implementations are faster than real-time, whereas CPU implementations typically take hours to complete.

3.5.2 Simulation of compensation during a pulse

Due to the narrow band operating parameters of the microASAR during the CASIE mission, the algorithm runtime does not clearly demonstrate the improved processing resulting from the motion correction described by Eq. (3.12). Since there is significant interest in UHF UWB SAR imagery [33–35], we simulate data for an UHF radar (800MHz center frequency) with 500MHz bandwidth and aircraft velocity of 150 m/s. We note that in a practical imaging scenario, the transmit signal at this frequency would be notched to avoid conflicting bands; however the notching does not significantly alter the results of the simulation.

In Fig. 3.5 the range compressed signal received from a single point target at a 45° squint is represented by the thick line. The thin line represents the signal received as if the aircraft were stationary as assumed with stop-and-hop. The dashed vertical lines represent the three indexing calculations discussed: from right to left the lines describe the indexing term using the stop-and-hop assumption, Ribalta's correction, and the indexing term given in Eq. (3.11).

In this scenario the UWB correction described in Eq. (3.11) is needed in order to index the peak of the range compressed data and properly focus the radar image. The ratio of Ribalta's correction and additional UWB correction is the ratio of the carrier frequency and the bandwidth of the radar. So for narrow band SAR imaging, where the bandwidth is much smaller than the carrier frequency, Ribalta's correction is sufficient for focusing the SAR image.

To illustrate the image quality improvement achieved with the motion corrections we simulate a scene of with a single point target depicted using the same radar parameters

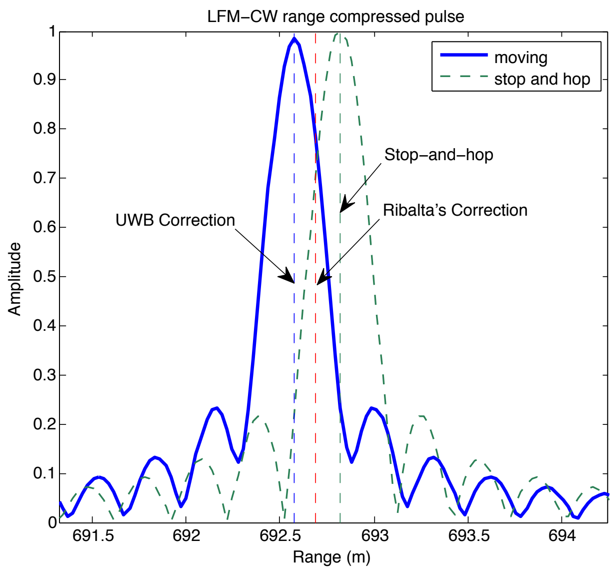


Figure 3.5: An illustration of range-compressed data for a point target observed by a moving aircraft (thick plot) and a stationary aircraft (dashed plot) and the available indexing terms. The indexing terms are represented by the vertical dashed lines. From right to left, the lines respectively represent the indexing term using the stop-and-hop assumption, Ribalta's correction, and the indexing term given in Eq. (3.12). The simulation is for an UHF radar with a 500MHz bandwidth and a target at a 45° squint.

given above. In this simulation shown in Fig. 3.6 the radar has a 60° beamwidth with fixed pointing.

As shown in Fig. 3.6a, using the stop-and-hop assumption smears and warps the impulse response. In [4], Ribalta's correction is shown to almost perfectly correct for the motion in the non-UWB imaging scenario; however, as shown in Fig. 3.6b it only slightly improves the impulse response for an UWB radar. The UWB correction derived in this work results in the improved response shown in Fig. 3.6c.

To further improve the impulse response, the motion of the aircraft can be approximated as piecewise linear during a pulse. This can be applied by breaking the dechirped data into disjoint sets in fast-time. Each subset represents a lower resolution SAR image with slightly differing center frequencies. The UWB correction can be applied to the sub-band data during backprojection and the final image is the sum of all of the sub-band images. Applying the correction in this manner results in the nearly ideal impulse response shown in Fig. 3.6d.

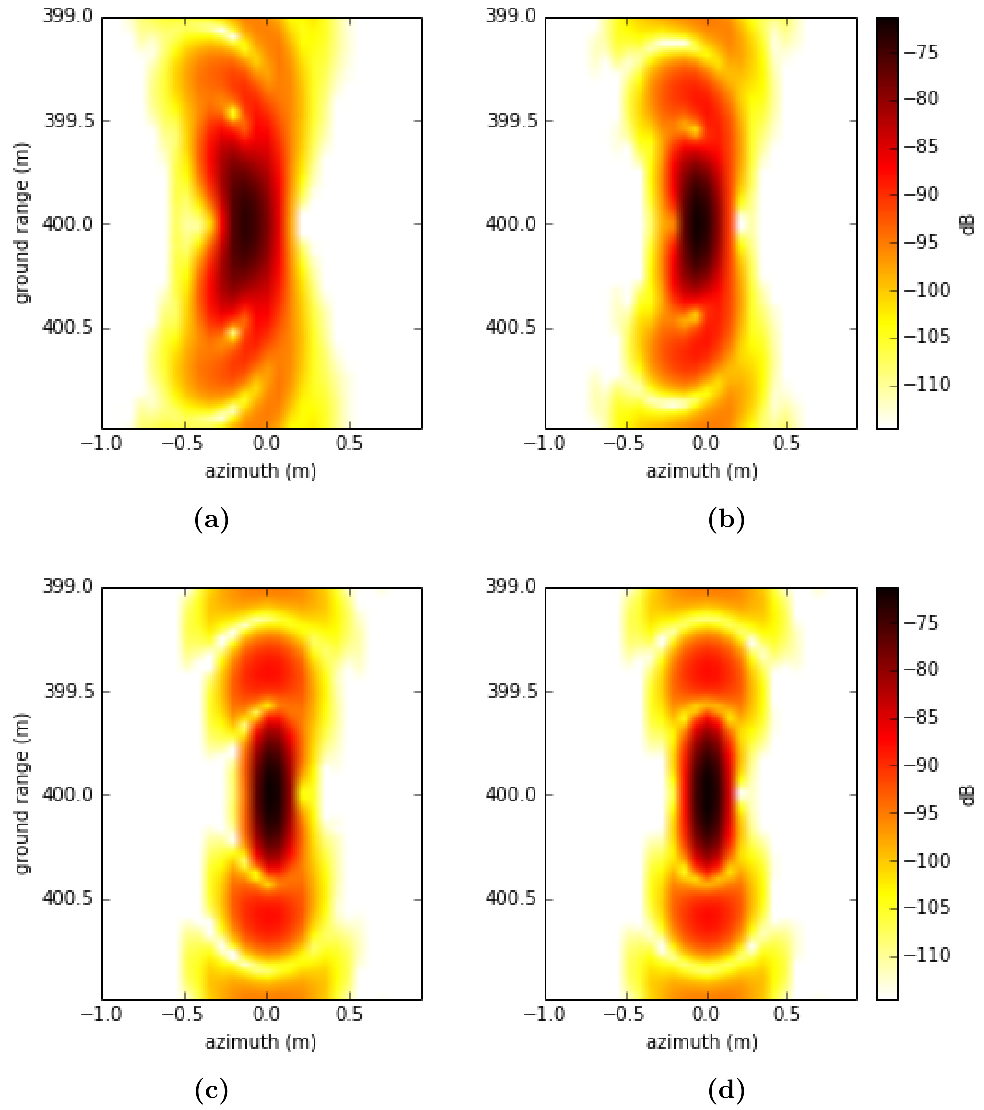


Figure 3.6: Single point target response of a UWB wide-beam SAR using (a) the stop-and-hop approximation, (b) Ribalta's correction, (c) the UWB correction, and (d) the UWB correction applied to sub-band images.

3.6 Summary

Using small LFM-CW SAR on unmanned air systems can provide unprecedented information for scientific and military missions. The low-altitude geometry and significant motion typical of small UAV operation requires significant motion compensation to generate high quality imagery. This work presents an efficient backprojection processor that provides excellent motion compensation for UAV-based stripmap SAR using consumer-grade GPUs.

We developed a stripmap backprojection algorithm from spatial coordinates accounting for the un-steered antenna and for the motion of aircraft during the long duration of an LFM-CW pulse, extending previous work by Ribalta [4] to handle UWB scenarios. The processor takes advantage of the unique processing hardware of the GPU to produce quality images faster than real-time for the CASIE mission.

Chapter 4

A Spectral Analysis of Single Antenna Interferometry

4.1 Introduction

SAR interferometry is a powerful tool that exploits the coherence of two complex images created from slightly differing aspect angles to infer further information of the scene such as topography or the detection of moving targets. Traditional SAR interferometry uses images collected using narrow beam antennae with a linear flight track from high altitude platforms. These assumptions make it easy to analyze the SAR operation in the spectral domain, and have enabled the development of computationally efficient algorithms for image compression [13, 36, 37]; however, with the recent availability of raw computational power from GPUs it has become desirable to forgo the computational efficiency of Fourier based methods and use time domain backprojection due to the simplicity of implementation and ability to handle wide-beam antennae and arbitrary flight paths. The dynamic motion of the small UAV used in CASIE, illustrated in Fig. 4.1, necessitated the motion compensation available using the time-domain backprojection in order to properly focus the images, which processing is described in Chapter 3.

It was suggested that it might be possible to use sub-aperture images from different altitudes along the aircraft flight path to perform interferometry and infer information about the topography of the sea ice. This chapter explores this proposal and extends the spectral analysis developed for traditional SAR interferometry [13, 38] to backprojection images and determines the requirements for single antenna interferometry to infer topography. Furthermore the requirements for coherence between two SAR images is clarified. It is shown that the motion of the aircraft during the CASIE mission is insufficient for the purpose.

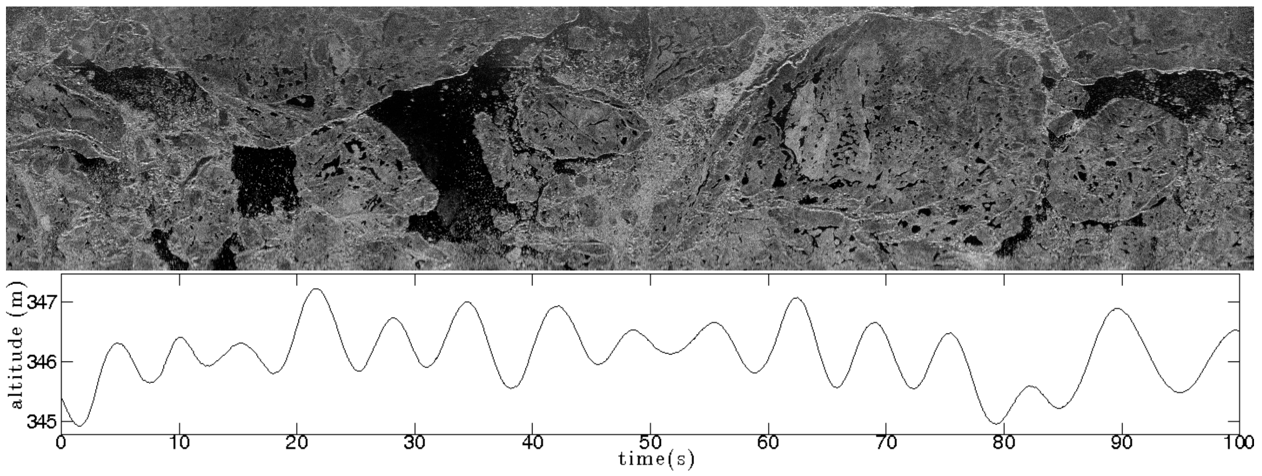


Figure 4.1: An example SAR image collected during CASIE-09, and a diagram of the altitude variation during the data collection of this image. The image resolution is 1m and the dimensions are roughly 650m x 3km.

4.2 Coherence and rough surface model

To begin, we review the definition of signal coherence and describe the rough surface model used in interferometry. Coherence is a measure of the linear dependence of two signals and is defined as

$$C_{xy} = \frac{\|G_{xy}\|}{G_{xx}G_{yy}}, \tag{4.1}$$

where G_{xy} is the cross-spectral density and G_{xx} and G_{yy} are the auto-spectral densities. The coherence of two signals is maximized when the signals are identical. In this discussion, the coherence of a signal is used to measure the overlap of the signal spectra. For an alternate yet detailed consideration of the correlation of SAR pixels from the perspective of time-domain backprojection see [39].

Interferometric estimation of topography relies on the assumption that the imaging scene $g(x, y, z)$ can be described as a rough surface

$$g(x, y, z) = r(x, y)\delta(z - h(x, y)), \tag{4.2}$$

where $h(x, y)$ is the topography of the surface and $r(x, y)$ is the surface reflectivity function, and δ is the Dirac delta function. It is also assumed that $h(x, y)$ is smooth compared to the resolution of the SAR. The surface reflectivity function, $r(x, y)$ is modeled as a wide sense stationary (WSS) random process, where each resolution cell contains multiple independently positioned point scatterers. This results in an exponential distribution of the received power, which is typical of natural surfaces [13,40]. Because $r(x, y)$ is a WSS random process its Fourier transform $R(\omega_x, \omega_y)$ is also WSS, and samples of $R(\omega_x, \omega_y)$ are statistically independent. Therefore disjoint portions of the surface spectrum used in generating the images leads to incoherence. It is apparent that in order for images to be coherent, the images must be spatially co-located and of similar shape. Image registration techniques can be applied to achieve the spatial co-location, but in order for the phase of the signal to be coherent, the pixels also need to be generated from the same portions of the surface spectrum $R(\omega_x, \omega_y)$.

In the following work, it is shown that the portion of the spectra contained in a SAR image pixel is given by the transmit frequency and bandwidth and the ranges of grazing and azimuth angles of the data collection. The discussion begins by describing the spectral sampling in two dimensions and introduces the projection slice theorem, and then extends the analysis to three dimensions. Afterwards, the principles discussed in the three dimensional spectral sampling are applied in the analysis of the single antenna interferometer.

4.3 Spectral sampling of a rough surface in two dimensions

Consider the two dimensional imaging scenario diagrammed in Fig. 4.2. The image cell in the scene is covered by multiple point scatterers as described in Section 4.2. Using the Born approximation the radar return is described as a convolution of the radar signal with the imaging scene within the beamwidth of the antenna, modelled mathematically as

$$\begin{aligned} s_r(r) = \int_0^{R_{max}} s_t(r-\tau) \int_{B_\psi} g(\tau \sin(\psi + l) + x, \tau \cos(\psi + l) + z)\, dl\, d\tau \\ s_t(r) * \int_{B_\psi} g(r \sin(\psi + l) + x, r \cos(\psi + l) + z)\, dl\,, \end{aligned} \tag{4.3}$$

where the antenna location is given by (x, z), r is the distance from the antenna, R_{max} is the maximum imaging range, ψ is the angle of the scene to the antenna, and B_ψ is the antenna elevation beamwidth. Because we are interested in pixel to pixel correlation of images, we can constrain our analysis to a single range compressed sample. Only a small portion of B_ψ

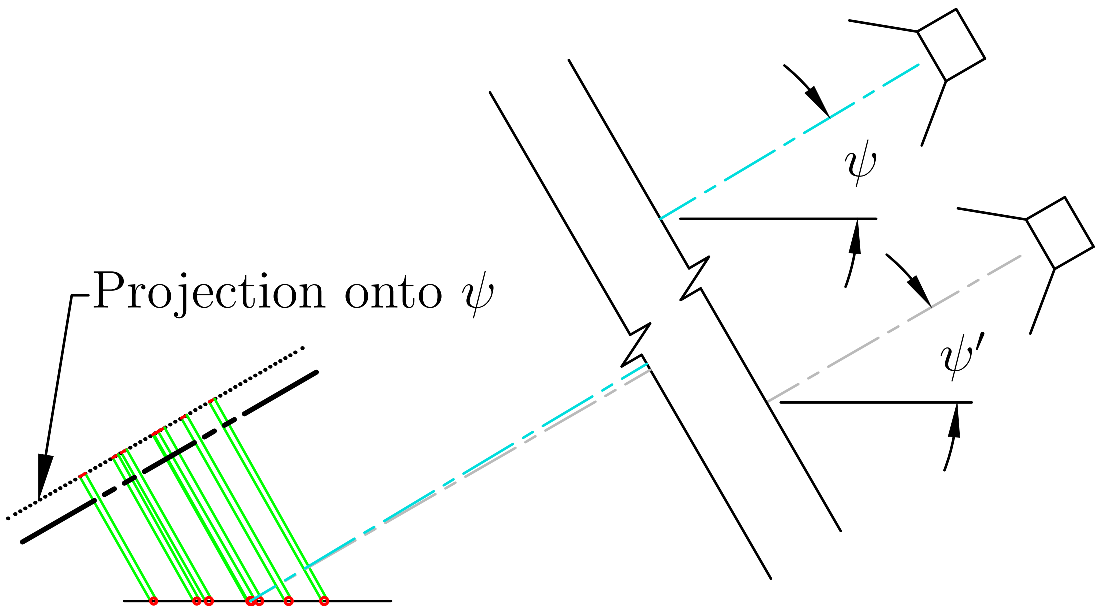


Figure 4.2: Two dimensional SAR imaging scene.

is considered, which we designate as L. Because the imaging scene is much smaller than the distance to the target we can apply the small angle approximation. Using the small angle approximation for functions of l, yields

$$\begin{aligned} s_r(r) &= s_t(r) * \int_L g\left(r\left(\sin\psi\cos l + \cos\psi\sin l\right) + x, r\left(\cos\psi\cos l - \sin\psi\sin l\right) + z\right)\, dl \\ &= s_t(r) * \int_L g\left(r\left(\sin\psi + l\cos\psi\right) + x, r\left(\cos\psi - l\sin\psi\right) + z\right)\, dl\,. \end{aligned} \tag{4.4}$$

Equation (4.4) describes an orthographic projection of g onto a line with an elevation angle ψ. The reader may note that the orthographic projection is only valid for small angle ranges, thus this analysis may not be valid for an image formation procedure for airborne SAR where the range of grazing angle is large; however, this analysis is valid in evaluating which portions of the spectrum are used in creating a pixel value.

The spectral sampling of the radar pulse given in Eq. (4.3) is defined by its Fourier transform; however, by the projection slice theorem, this is approximately equivalent to taking a slice of the two dimensional Fourier transform of the two dimensional image spectrum.

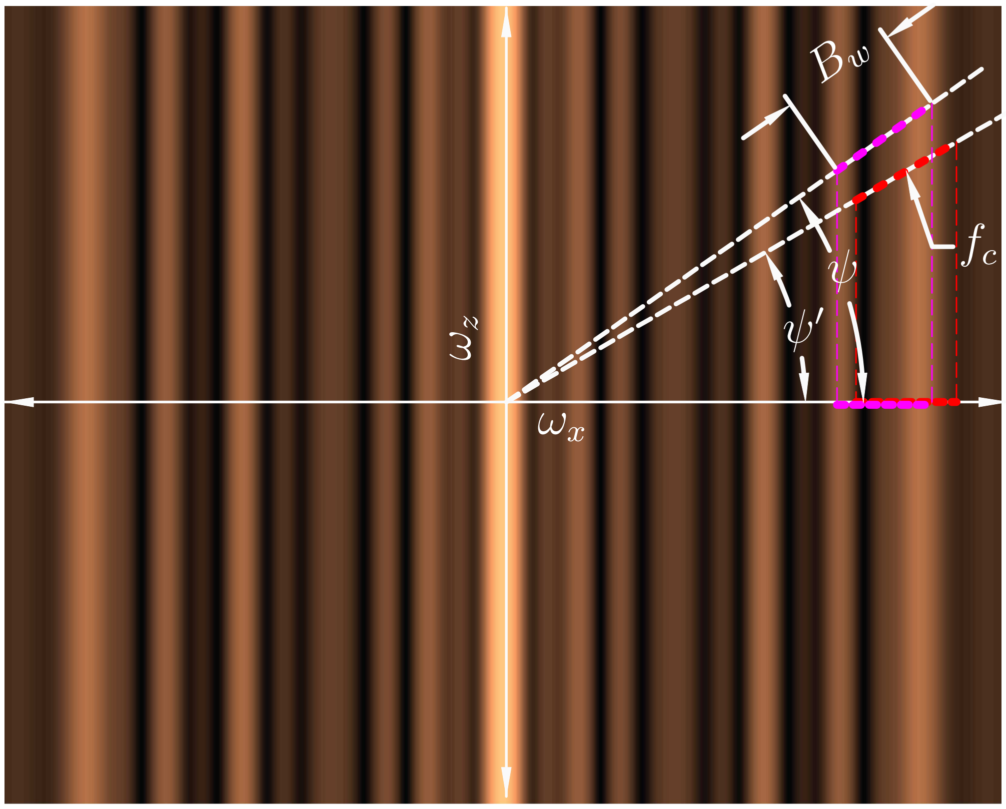


Figure 4.3: Illustration of the spectral sampling of a single pulse from two different observation angles. The background represents the spectrum of a surface contained on the x axis, thus the spectrum is constant in ω_z. The center frequency, f_c and the bandwidth, B_w, of the radar determines the sampled portion of the one dimensional slice.

This is illustrated by looking at the slice of the spectrum taken along the ω_x axis

$$\begin{aligned} G(\omega_r, 0) &= \int\int g(x,z) e^{-j2\pi x\omega_r} dx\, dz \\ &= \int \left[\int g(x,z) dz\right] e^{-j2\pi x\omega_x}\, dx \\ &= \int p_x(x) e^{-j2\pi x\omega_x} dx \\ &= \mathcal{F}\{p_x\}\,, \end{aligned} \tag{4.5}$$

where the p_x is the orthographic projection of $g(x,z)$ onto the x axis. This extends to a slice at an arbitrary angle because the Fourier transform of a rotated space is equal to the rotated Fourier transform of the non-rotated space. Note that the spectral slices always extend from the origin of the spectral space. The spectral sample is illustrated in Fig. 4.3 for two different grazing angles. The background image represents the spectrum of a surface contained on the

x axis, thus the spectrum is constant in ω_z. The observation angle of the radar determines the angle of the slice through the two dimensional spectrum. The center frequency, f_c, and the bandwidth, B_w, of the radar determine the sampled portion of the one dimensional slice. Note that the discrete samples illustrated by the colored dashes in Fig. 4.3 can be considered the raw data samples of a LFM pulse or the Fourier transform of the range compressed data. The spectrum in Fig. 4.3 is constant in the ω_z direction because the scatterers are contained on a line in the two dimensional imaging space. Likewise if the surface extended along an angle α, then the spectrum is constant along $\alpha + \frac{\pi}{2}$, or equivalently the slice is taken along $\psi + \alpha$ of the non-rotated spectrum. Thus by the spectral slice theorem, the portion of the surface spectrum sampled by the radar at angle ψ is equivalent to the portion sampled by a radar with 0 degree grazing angle and a lowered center frequency and bandwidth.

4.4 Spectral sampling in three dimensions

The previous analysis is easily extended to three dimensions because the projection slice theorem is applicable to any number of dimensions. As explained in Section 4.3, the contribution of a single resolution cell to a radar pulse can be described by the convolution of the radar transmit signal with an orthographic projection of the imaging cell onto a line proceeding from the radar to the center of the imaging cell, and by the projection slice theorem the spectral contribution contained in this pulse is approximately the one dimensional slice of the three dimensional spectrum taken at the same angles as the projection. To illustrate this refer to the geometry shown in Fig. 4.4 of two squinted SAR collections from two different altitudes. Note that the aircraft at the higher altitude effectively has a somewhat narrower beamwidth because the target is viewed from the same span of ground azimuth angles.

Figure 4.5 illustrates the spectral sampling of the SAR data collections illustrated in Fig. 4.4. The sampling lines extend at the same angles as the angles traveled by the SAR during the data collection in reference to the center of the pixel. The samples are made along one dimensional slices as specified by the radar operating frequencies. Because the scene of interest is contained on the surface of the x-y plane, the spectrum is constant in ω_z thus the two collections effectively sample the portions shown in Fig. 4.5b. Note that like the

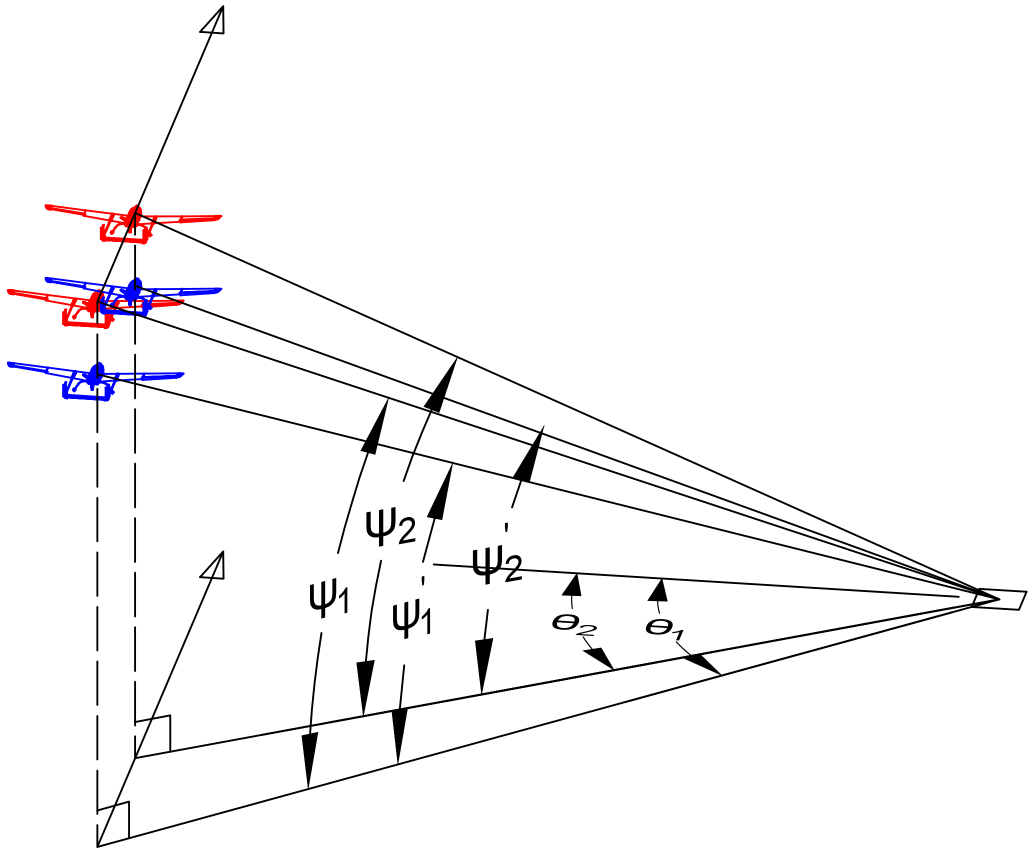
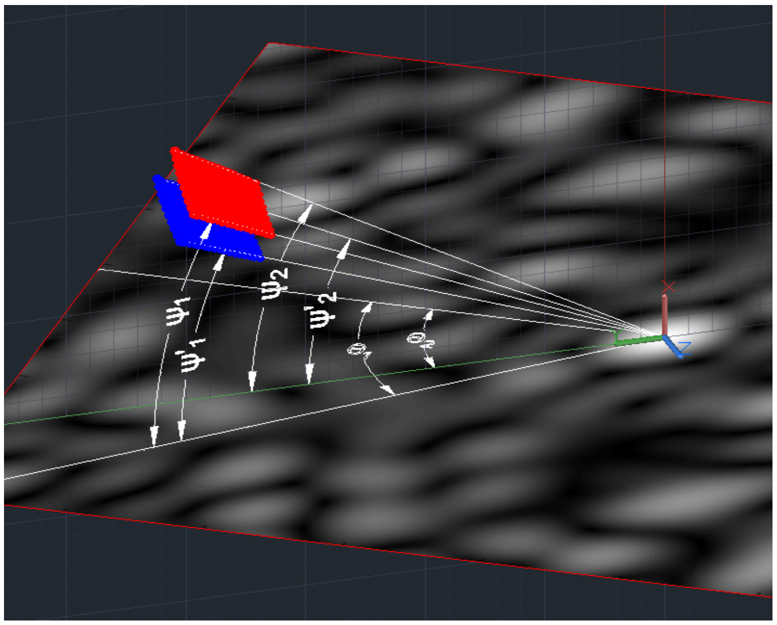
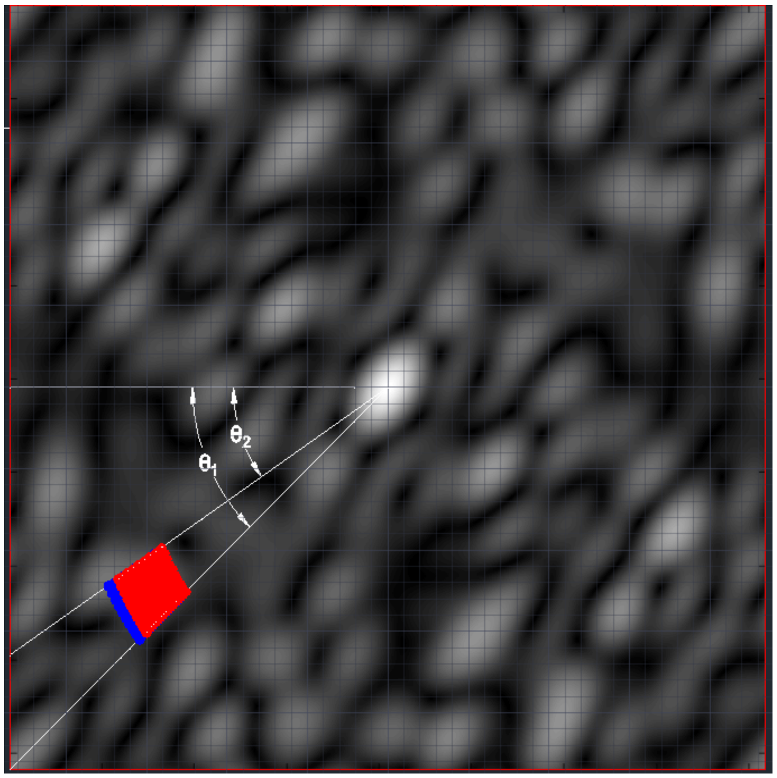


Figure 4.4: Illustration of squinted SAR geometry from two different altitudes. The blue and red planes respectively identify the ends of a SAR data collection. The azimuth and grazing angle, respectively θ and ψ, along with the radar transmit frequencies determine the portion of the surface spectrum that is sampled, as shown in Fig. 4.5. Note that the aircraft at the higher altitude effectively has a slightly narrower beamwidth because the target is viewed from only the same span of ground azimuth angles.

two dimensional imaging scenario, the difference in grazing angle shifts the sampling of the ground spectrum along the ground azimuth angle. However because the shift is only along the ground azimuth angle, it is necessary to view the scene from the same range of azimuth angles in order for the collections to have azimuth spectral overlap.



(a)



(b)

Figure 4.5: Illustration of spectral sampling in three dimensions for the flights diagramed in Fig. 4.4. For clarity an isometric (a) and plan (b) view are provided. Like the background shown in Fig. 4.3, the gray background represents the spectrum of a pixel. The surface spectrum is invariant along the z axis because the pixel surface is contained in the x-y plane. The red and blue dots note the portion of the 3D spectrum that is sampled according to the geometry of the collection and the operating frequencies.

4.5 Cross-track interferometry

Topography estimation is performed using cross-track interferometry where two radars (or alternatively, two simultaneous receivers and one radar transmitter) are placed at different elevations, resulting in different grazing angles ψ and ψ' as illustrated in Fig. 4.3 and Fig. 4.5. In this example the radars have the same center frequency and bandwidth. The radars sample slightly different portions of the surface spectrum as illustrated by the projection of the samples onto the surface. Consider the case where the image scene only consists of a single planar patch of a rough surface. Because, as illustrated in Fig. 4.3, the grazing angle of a radar shifts the portion of the spectrum that is sampled, the difference of the grazing angles of the radars can be identified by the peak of the cross-correlation of the sample spectrum. This coupled with prior knowledge of the separation of the antennae allows for the angle to the patch to be estimated, which fully describes the topography of the scene.

With a full stripmap SAR imaging scene, the spectrum of a patch cannot be directly compared because the collected data contains spectra from all of the targets in the image scene. Instead the phase of the image scenes is compared. The phase difference of the images is directly related to the frequency shift of the images. Because a shift in the frequency domain is equivalent to a modulation in the spatial domain, the pixel phase difference can be considered the DC component of the modulation plus noise. The noise of the phase difference is greatly caused by what has been referred to in the literature as geometric decorrelation, which is a result of disjoint portions of the spectrum being included in generating the pixels. The geometric decorrelation can be completely removed, increasing the coherence of the two images, by filtering off the disjoint portions of the spectrum. This can be done by applying windows in the spectral domain. For LFM-CW SAR the filtering may be done by windowing the dechirped data before range compression. Alternatively the decorrelation can be overcome by "tunable systems", which can adjust their operating frequencies as suggested in [38].

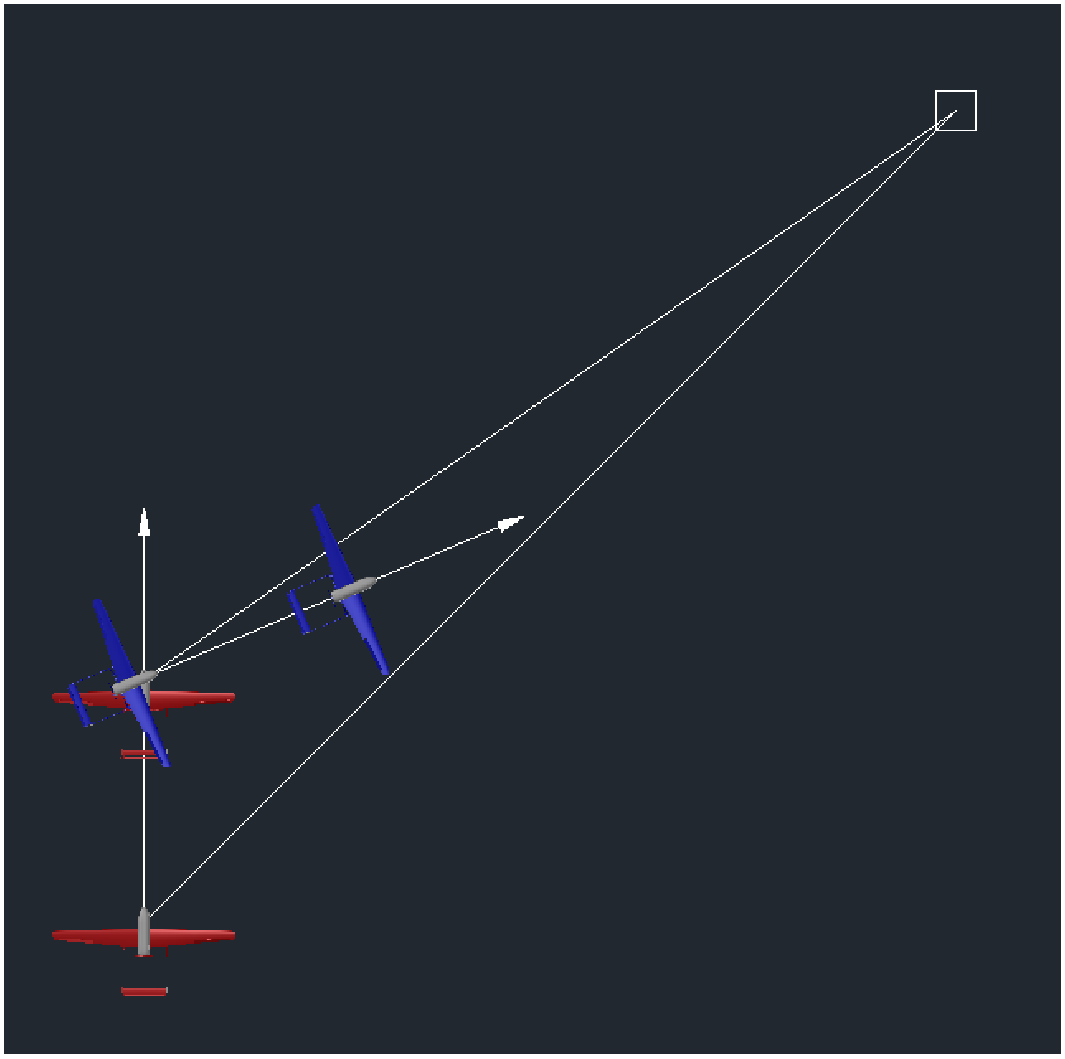


Figure 4.6: Diagram of a flight geometry that allows for azimuth spectral overlap.

4.6 Analysis of single antenna interferometry

With the principles developed in the previous sections, we can analyze the opportunity for applying interferometric techniques to data collected along an oscillatory path. First consider the case of an aircraft approaching the imaging scene where the scene of interest is to the right of the initial heading of the aircraft. We choose the reference frame origin to proceed from the scene center and to measure the ground azimuth angle between the projections of the pointing angle to the receive antenna and the angle orthogonal to the aircraft heading pointing toward the aircraft as shown in Fig. 4.6.

Consider an aircraft that has received a collection of pulses along a relatively straight flight path. Because the aircraft must traverse some of the same azimuth angles in order to achieve spectral overlap, then the aircraft must either reverse direction or the heading of the aircraft must turn significantly to the right. The turn must make the projection of the heading onto the ground plane cross to the other side of the imaging scene, as illustrated in Fig. 4.6. If the platform is capable of this turn, the length of the interferometric baseline is determined by the airplane turning radius, which in most imaging scenarios results in a significant change in the incidence angle, thus requiring a very wide range of transmit frequencies as well. I note that if the surface rises away from the aircraft the requirements may be lower.

These motion requirements are impractical for a fixed wing aircraft except at very high squint angles. However, even at a high squint angle it is very difficult to collect an appropriate data set, because in making the maneuvers required for spectral overlap, the scene of interest likely moves from the right side of the aircraft to the left. Thus this method would either require a very wide and forward pointing antenna (resulting in difficult ambiguities to resolve) or to be able to be steered from one side of the aircraft to the other.

4.7 Conclusions and summary

With this analysis it is clear that interferometric techniques cannot be used with the CASIE SAR mission to estimate topography. It is argued that in order for traditional interferometric techniques to be used with a single antenna that the heading of the aircraft must cross the pixels to be used. This work extends the spectral analysis for interferometry using narrowband narrow beam spotlight SAR in [13, 38] to wide-beam stripmap SAR. It is hoped that this analysis acts as a useful tool to help the SAR community to understand the difficult but insightful concepts resulting from a spectral analysis of SAR.

Chapter 5

Autofocus for Low Altitude SAR

Autofocus methods attempt to improve image focus caused by unknown variables, including unknown motion and atmospheric conditions. While many autofocus methods have been proposed and implemented, there continues to be a need for improved methods to address the particular defocus errors that arise in different scenarios and work with different SAR processing methods. This chapter describes autofocus from a matched filtering perspective, gives a brief review of previous autofocus methods, and develops a new autofocus method for low-altitude high-resolution SAR. By using mapdrift techniques applied to sub-images, this method accounts for defocus due to three dimensional motion error in the aircraft position. The aims of this algorithm are similar to the recently proposed Factorized Geometrical Algorithm (FGA) [41] but by employing mapdrift techniques the computation is greatly reduced.

Many traditional autofocus methods attempt to find a set of phase corrections that improve the focus of the image. The phase adjustments account for a variety of errors including motion, but correcting for phase errors corrects for only part of the defocus. This is particularly true for ultra-wide band systems where the amplitude errors caused by inaccurate motion can be as significant to defocus as the phase errors. Because geometry errors can be the major cause of defocus for low-altitude UAV SAR systems, the goal of the autofocus method is to find adjustments to the antenna positions that maximize the focus of the image. Fortunately, low-altitude SAR systems have the unique opportunity to identify three dimensional motion because they often image scenes with incidence angles ranging from $0 - 75°$. FGA uses a direct approach to autofocus by directly making adjustments to the image processing to find the motion parameters that improve the focus metric; however, the computational load of a direct method is tremendous. So tremendous in fact that only a single parameter was varied in the FGA paper [41]. This work attempts to make the goals of FGA tenable by reducing the search space using mapdrift techniques. As an alternative to directly evaluating motion parameters in FGA, in this chapter, I use mapdrift techniques to estimate the motion of the aircraft from shifts of sub-images. mapdrift methods exploit the fact that incorrect motion parameters not only defocus the image but warp the sub-aperture images relative to one another. In the past, mapdrift autofocus has only been applied to along-track velocity and acceleration errors, and this work extends it to three dimensional motion errors. In some ways this work is similar to [42], but uses both the range sphere and the Doppler cone and is designed for time-domain backprojection.

5.1 SAR defocus: a time-domain perspective

Before describing the new autofocus framework I review SAR image formation and the types and causes of image degradation. Following which I then give an overview of existing autofocus methods.

5.1.1 SAR image formation

The backprojection filter for SAR image formation is described in Chapter 2, but for convenience it is repeated here in a slightly different form

$$I(x,y) = \sum_{m} W[m](f(\mathbf{p}[m]), \psi[m], x, y) S_m\left(\mathbf{p}[m], x, y\right) e^{-j\phi(\mathbf{p}[m],x,y))}, \tag{5.1}$$

where m is the pulse number, $W[m]$ is an apodization window, S_m is the range compressed data, f is an interpolation method acting on $\mathbf{p}[m]$ which is the position of the aircraft, $\psi[m]$ is the attitude of the antennae, (x, y) is the pixel coordinates, and ϕ is the expected phase function. In some cases the indexing of S_m may factor in the aircraft velocity as discussed in Chapter 3, but for many cases it is a simple linear function of the range to the target. For simplicity, it is assumed to be a linear function of range in this discussion.

Inaccuracies in the measurements of the motion of the aircraft $\mathbf{p}[m]$ cause errors in the selection of the range compressed data and the calculation of the expected phase. Phase errors cause defocus but can be tolerated to an extent dependent on the base frequency, beamwidth, and bandwidth of the radar [9, 12, 43]. Inaccuracies in the indexing of the range compressed data are more complicated and can degrade the resulting image due the incorrect data selection. However, with only a few exceptions, autofocus algorithms only correct for phase errors, leaving significant noise introduced by the incorrect data selection. This is particularly true for ultra-wide band SAR. The autofocus algorithm presented in this chapter accounts for defocus and noise introduced by both phase inaccuracies and incorrect data selection by estimating the correct motion parameters.

5.1.2 Image degradation

For the purposes of this discussion, I classify three types of image degradation that may occur in a SAR image, including defocusing, warping, and decreased signal to noise ratio (SNR). Warping is a shifting, skewing, or rotation of part or all of the image. Defocusing is a broadening and smearing of part or all of the focused image caused by phase inaccuracies. Whereas decreased SNR of the image is characterized by a lower ratio of the scatterer peaks to the background noise, which in the SAR image formation process can be attributed to amplitude errors. The amplitude errors could be a result of inaccurate range compressed data selection or apodization window weighting due to inaccurate knowledge of the antenna pointing direction.

The following describes the major sources of image degradation and the accompanying degradations that occur. Note that the sources of image degradation usually cause multiple types of degradation. The stated goal of an autofocus method is to reduce the defocusing of the image; however, an autofocus method can be designed to improve the SNR as well. Because there is not an independent measure for image warping without a reference, image warping may still remain in focused images.

Oscillator jitter and drift

SAR systems are designed with strict coherence requirements, but despite efforts to maintain a stable clock, there always exists residual phase jitter and drift in the system oscillator. This incoherence results in defocusing of the SAR image.

Atmospheric conditions

The speed of electromagnetic waves vary depending on the transmission medium, as a result atmospheric conditions such as precipitation, clouds, or even temperature variations cause changes in the speed of propagation. For space-borne systems, this can be a significant challenge, while with low-altitude systems the distance to the ground is too small for the slight variation of medium to significantly affect the image quality.

Inaccurate geometry

The effects of inaccurate geometry parameters on the SAR image processing are complex and not fully described in the literature. A typical SAR data collection, with the exception of circular SAR, is along a nearly straight path from a high altitude observing surfaces with smooth topography. With SAR operating wavelengths ranging from centimeters to millimeters, errors in motion measurements on the order of only millimeters can cause phase errors large enough to cause defocusing. It is important to note that although the relative accuracy of the motion measurements is critical to focusing, a gradual drift results in a warping of the image rather than a defocusing as caused by errors in the relative accuracy. Later in section Section 5.3.2 the full effects of geometry errors are discussed.

It is known that if the synthetic aperture is along a straight track then all objects in the image scene properly focus in some slant range position, and thus appear in focus for any topography used in backprojection [13]. However, when the synthetic aperture deviates from a single dimension then inaccurate topography results in defocusing. The amount of defocusing may be negligible depending on the SAR system and the magnitude of the drift from a straight path. If backprojection processing is used with an accurate digital elevation model (DEM) and motion record, the entire image focuses properly.

Errors in the antenna pointing also degrade the image. The SNR is degraded if the antenna is mispointed and can be reduced to near zero if antenna pointing errors are off by more than the beamwidth of the antenna. An autofocus method can be designed to maximize the SNR by correcting antenna pointing and pattern errors. However, these errors are typically handled separately using Doppler centroid estimation techniques [9, 44].

5.2 Review of previous autofocus methods

Over the years a host of autofocus algorithms have been developed for different imaging scenarios. The algorithms can be categorized in different ways. Autofocus methods vary by the adjustments made and what metric is used in order to improve the image focus. The earliest autofocus methods were prominent point processing (PPP) and mapdrift. Later the popular phase gradient autofocus (PGA) was introduced. The most recent autofocus methods are posed as the optimization of an image metric such as minimum entropy or maximum contrast. In this section I describe several of these methods.

5.2.1 Prominent point processing

Prominent point processing (PPP), also known as inverse filtering, uses the phase of an isolated point target to estimate the phase that maximizes the entire image; however not all images contain isolated point targets, rendering this method unusable in these situations. Multiple prominent point processing (MPPP) is an extension of PPP that uses multiple points to improve the phase estimate [12, 13]. MPPP and PPP are closely related to inverse SAR (ISAR) processing methods. In theory, with enough prominent points in a non-planar orientation, MPPP can be used to estimate all of the motion parameters without the assistance of a navigation unit; however, it has been noted that this has yet to be demonstrated [45].

5.2.2 Mapdrift

Mapdrift (MD) methods compare sub-aperture images to determine a polynomial phase function that can improve the focus. The phase difference that mapdrift methods can detect are limited to smooth function whereas methods like PPP can detect finer changes. However mapdrift can incorporate all of the range data in the estimate and are thereby less susceptible to noise than PPP. Traditionally the comparison between sub-apertures is done using the magnitude of the images only. An alternative approach has been proposed called coherent mapdrift (CMD) [46]. The coherent mapdrift autofocus has been shown to have super-convergence for estimating the velocity [47].

5.2.3 Phase gradient autofocus

Phase Gradient Autofocus (PGA) is a non-parametric method that is considered by many to be the industry standard for autofocus methods [12–14]. The key steps of PGA are iteratively applied. First the data is circularly shifted such that the brightest target in each range bin is centered. Then an azimuth window is applied. In the final step, the phase common to all the range bins is estimated [13]. The PGA uses a target in each range bin and a maximum-likelihood phase estimator.

Thompson et. al describe a method for extending the phase gradient autofocus method to low-altitude stripmap SAR [48]. In order to handle the low-altitude geometry a range dependent phase estimator is proposed using weighted least squares instead of the traditional maximum likelihood estimator. Also, a stripmap to spotlight conversion is presented that handles the range dependence of low-altitude SAR [48]. A weakness of this method may be that the phase error is posed purely in the cross-range dimension; however, it has been argued by many that phase errors in azimuth (acceleration errors) are a more critical source of defocus [12, 49].

Another extension of the PGA is described in [50] that builds off of Thompson's method and accounts for both the cross-range and azimuth motion error. Their approach involves a multi step approach to compensate for motion error as well as non-range dependent errors potentially caused by atmospheric effects or clock drift and jitter.

5.2.4 Image metric optimization

Various image metric optimization autofocus methods have been proposed. The advantage of these autofocus methods is that they can use the entire received data set, they can easily be applied to stripmap SAR including backprojection SAR, and they are not limited to phase corrections. Different image metrics have been found to emphasize different image features and the selection of the image metric can have significant effect on the autofocus performance [51]. Frequently the deciding factor for the chosen metric is computation. For this reason maximum image intensity and minimum entropy methods are particularly popular, since the derivatives of the metric have well known solutions. Kragh's minimum entropy autofocus [52] is a good example of state-of-the-art autofocus algorithms. It uses a coordinate descent optimization within the optimization transfer framework which guarantees convergence. It is shown that this method is slower than PGA, but that it improves the focus.

I make note of another autofocus method, recently proposed by Ash [53], because it is one of the select few methods that can be incorporated with backprojection processing. Ash's autofocus method uses the maximum intensity metric within a coordinate descent optimization method. Using the intensity metric allows for the update step to be calculated analytically, which greatly reduces the computation. However, this method is limited to phase corrections.

5.2.5 Factorized Geometrical Algorithm

While it may be considered an image metric autofocus, the Factorized Geometrical Algorithm (FGA) [41] is by far the most complete autofocus method for accounting for motion measurement errors and deserves a separate description. It differs significantly from traditional autofocus methods [41, 45, 49] because it works within the framework of Fast Factorized Backprojection (FFBP) and attempts to estimate the unknown motion parameters by maximizing the correlation magnitude when merging sub-apertures. The motion parameters considered are:

1. the altitude of the merged sub-apertures
2. the angles between the ground plane and the vector for the merged sub-aperture
3. the angle between sub-apertures
4. the length of the two sub-apertures
5. the length of the merged aperture.

In each step of the FGA, the merging of sub-aperture images are warped according to two nonlinear transformations using a set of potential motion parameters enumerated above.

These parameters apply linear adjustments to the sub-apertures, but this requires a non-linear transformation of the image, and so the FGA effectively performs a non-linear cross-correlation at each step. This computation explains why the authors note that the slow runtime precludes it from regular usage.

On a side note, although not identified by the authors, the listed motion parameters are only sufficient for imaging near-planar surfaces that do not require the use of a digital elevation model. In a SAR imaging scenario with significant topography variation, an elevation model is required in the image generation if the synthetic aperture is non-linear. As a result, three additional parameters need to be accounted for in the model, including the x and y position of the first sub-aperture and the angle of the first sub-aperture around the vertical axis.

5.3 3D Mapdrift

In this section I present the design of a new autofocus method that, like the FGA, can estimate three dimensional motion, but requires significantly less computation. This is accomplished using mapdrift methods such that the non-linear image transformations are not needed. To begin I give an overview of the autofocus method, which is presented as the solution to an inverse problem, and then derive the forward model for the image shifts.

5.3.1 Method overview

As described earlier, mapdrift methods take advantage of the image warping that occurs due to the incorrect motion parameters in order to estimate the motion measurement errors that caused the defocus. In the past this has been done only for azimuth motion errors. I extend this concept to three dimensional motion. This is accomplished using a novel forward model for the shift of a point target, which shift is caused by inaccurate motion parameters. This forward model is then used within a minimization framework to identify potential motion parameters that can improve the focus of the image. Like many inverse problems, estimating the motion parameters is ill-posed and in many cases the true motion parameters are not found, but as shown in the results section of this chapter, the estimated motion parameters still greatly improve the image focus.

The forward model described in the following subsection calculates the shift of a single point target. Recall that we are interested in low-altitude broad-swath operation. The model is used to describe the shift of the center of SAR image sub-sections. The sub-sections are referred to as panels. Unlike traditional mapdrift techniques that measure the shift of two images using cross-correlation only in the azimuth dimension, a two dimensional cross-correlation is used, measuring the shift of the panel centers in both azimuth and range. By segmenting the panels in range, range motion errors can be differentiated from altitude motion errors because altitude errors have a greater shift near nadir than farther out in the swath, whereas range errors are consistent across the swath. Segmenting the panels in azimuth allows for the motion error to be measured over time. The size of the panels affects the magnitude and contrast of the cross-correlations and the resilience to the noise, but it also affects the number of shifts that are estimated. In order to reduce the error of the shift estimates and to have a greater number of shift estimates, the panels can be chosen to overlap in both in range and azimuth. This enables the panels to be of sufficient size to provide a good estimate of the shift and provides many reference points.

Traditional mapdrift methods use the peak of the cross-correlation in a simple model of the image shifts to estimate velocity and acceleration errors in the azimuth direction. Following a similar design to traditional mapdrift methods, the motion can be estimated as follows. After the sub-aperture shifts have been measured for each sub-aperture, an optimization method is employed to find a set of motion parameters that minimize the least-squares difference with the measured shifts. This can be done with the minimization problem

$$\arg\min_{\Delta} \sum_{l=0}^{L-1} \sum_{k=0}^{K-1} W(l,k) \left((\xi_x(\Delta)[k] - \bar{\xi_x}[k])^2 + (\xi_y(\Delta)[k] - \bar{\xi_y}[k])^2 \right), \tag{5.2}$$

where L is the number of panels, K is the number of combinations of sub-apertures, W is a weighting function, $\bar{\xi}$ are the measured shifts, and $\xi(\Delta)$ are the estimated shifts given the motion error parameters Δ. The choice of the weighting function can be a function of the magnitude of the correlation and of the relation of the compared sub-apertures. However, because this is an ill-posed problem, the solution to Eq. (5.2) is very sensitive to noise, and requires significant regularization in order to avoid overfitting the solution.

Alternatively, a method that is less sensitive to noise can be developed by using the cross-correlation of the panels as the cost function, instead of estimating the shifts from the cross-correlation peaks. Consider the cost maximization problem

$$\arg\max_{\Delta} \sum_{l=0}^{L-1} \sum_{k=0}^{K-1} W(l,k) X_{l,k} \left(\xi_x(\Delta)[k], \xi_x(\Delta)[k] \right), \tag{5.3}$$

where $X_{l,k}$ is the normalized cross-correlation for panel l and sub-aperture combination number k.

The optimization method used to maximize the objective function, Eq. (5.3), greatly determines the speed and accuracy of the autofocus method. The brute-force method used by [41] performs an exhaustive search over all of the motion parameters. The advantage of exhaustive search methods are the simplicity of implementation and the guarantee of finding the global minimum if the search spacing is appropriate. However, the accuracy of the minimization is limited by the quantization of the search used and, as has been noted in [41], an exhaustive search method is prohibitively time consuming. Thus alternative methods are used. Section 5.4 demonstrates the improved focus that can be achieved using this model, but first I describe the forward model for the image shifts.

5.3.2 Motion error forward model

To describe the three dimensional autofocus, I first present the forward model of the image warping. This novel development extends traditional mapdrift to three dimensions. The forward model for the shifts is used in an optimization framework to estimate motion parameters that improve the image focus. Mapdrift methods utilize sub-aperture processing. Because each sub-aperture image is of the same portion of ground, but is generated from a different portion of the azimuth beamwidth, the data is collected at different positions along the synthetic aperture. This is frequently termed multi-look processing. For $l = 0, 1, ..., L$ where L is the number of looks or sub-aperture images, multi-look processing can be represented in the matched filter with the inclusion of an additional apodization window,

W_l, written as

$$I_l(x,y) = \sum_m W_l[m](\mathbf{p}[m], \psi[m], x, y) W[m](\mathbf{p}[m], \psi[m], x, y) \cdot S_m\left(\mathbf{p}[m], x, y\right) e^{-j\phi(\mathbf{p}[m],x,y))}. \tag{5.4}$$

With the correct motion parameters, sub-aperture images align spatially, and accumulating of the magnitude of the sub-aperture images reduces the speckle of a SAR image. On the other hand, motion errors defocus and warp the sub-aperture images relative to one another.

Consider the backprojection processing for a point on the imaging surface, (x, y), when the incorrect motion parameters are used. This situation can be modeled by including an additional term $\vec{\Delta}$ that accounts for the incorrect motion parameters into the sub-aperture matched filter described by Eq. (5.1). This is written as

$$I_l(x,y) = \sum_{m \in \mathcal{Z}} S_m\left(g\left(\mathbf{p}[m] + \vec{\Delta}[m], x, y\right)\right) e^{-j\phi(\mathbf{p}[m]+\vec{\Delta}[m],x,y)}, \tag{5.5}$$

where $\mathbf{p}[m] + \vec{\Delta}[m]$ are the positions of the aircraft used in the matched filter, and $\mathcal{Z}$ is the support of the sub-aperture apodization window. I have ignored the magnitude of the apodization windows in order to simplify the analysis. This does not significantly alter the analysis because the windows are slowly changing amplitude-only functions. To show the warping effect of the incorrect motion parameters on a single sub-image, the motion errors are separated into two components described by

$$\vec{\Delta}[m] = \vec{\Delta}_l + \vec{\Delta}_d[m], \tag{5.6}$$

where $\vec{\Delta}_l$ is the average position error over the sub-aperture l, and $\vec{\Delta}_d$ is the difference of the actual position and the average position. Incorporating Eq. (5.6) into the phase term of Eq. (5.5) results in

$$\begin{aligned} I_l(x,y) &= \sum_{m \in \mathcal{Z}} S_m\left(g\left(\mathbf{p}[m] + \vec{\Delta}_l + \vec{\Delta}_d[m], x, y\right)\right) e^{-j\phi(\mathbf{p}[m]+\vec{\Delta}_l+\vec{\Delta}_d[m],x,y)} \\ &= \sum_{m \in \mathcal{Z}} S_m\left(g\left(\mathbf{p}[m] + \vec{\Delta}_l + \vec{\Delta}_d[m], x, y\right)\right) e^{-j\phi(\mathbf{p}[m]+\vec{\Delta}_l,x,y)} e^{-j\phi(\vec{\Delta}_d[m])}. \end{aligned} \tag{5.7}$$

The phase error due to the average position changes the phase of the result, but does not cause defocus of the image. Thus if the motion error is small, the sub-aperture image can be approximated as a warped copy of the properly focused image $\hat{I}_l$, described as

$$I_l(x,y) \approx \hat{I}_l(x+\delta_x, y+\delta_y)G(x,y), \tag{5.8}$$

where G is a blurring caused by last phase term in Eq. (5.7) and the incorrect range compressed data selection. The shifts (δ_x, δ_y) in Eq. (5.8) are a function of the aircraft positions and the imaging surface topography.

The location of a target in an image is based on the relative geometry of the synthetic aperture and the target. In modeling the effects of inaccurate motion measurements on the image, first consider the case where the aircraft flies a linear path. With a linear path, the return from a target is identical to any other target with the same range from the center of the synthetic aperture and the same cylindrical angle relative the axis of motion. Thus the target location is ambiguous. The ambiguity of the target location is described by a circle perpendicular to the axis of motion. The offset of the circle is determined by the average squint of the data collection. This has been described previously in the literature as the intersection of the range sphere and Doppler cone [12, 13, 54]. The placement of a target in a sub-aperture image is given by the intersection of the range sphere, Doppler cone, and the surface used in the SAR processing. These equations are given by

$$R^2 = \|\mathbf{p} - \mathbf{x}\|, \tag{5.9}$$

$$\cos(\psi) = \frac{\langle \mathbf{p} - \mathbf{x}, \mathbf{v} \rangle}{R\|v\|}, \tag{5.10}$$

$$z = g(x,y), \tag{5.11}$$

where the aircraft position and velocity components are given by the vectors $\mathbf{p} = [p_x, p_y, p_z]^T$ and $\mathbf{v} = [v_x, v_y, v_z]^T$, respectively, and the target position is given by $\mathbf{x} = [x, y, z]^T$.

For a single point the warping can be described by a 2D shift. This shift (δ_x, δ_y) is given by substituting $\mathbf{x}' = \mathbf{x} + [\delta_x \delta_y 0]^T$, $\mathbf{p}' = \mathbf{p} + \vec{\Delta}$, and $\mathbf{v}' = \mathbf{v} + \hat{\mathbf{v}}$, where $\hat{\mathbf{v}}$ is the average velocity error vector over the sub-aperture, into Eq. (5.7) and solving for (δ_x, δ_y). The solution to the shift can be easily found in closed form with a computer algebra system (CAS) such as Sympy [55], but is not printed here because it is very complex and is not necessary for this discussion. For clarity we describe the shift as the function

$$f(\mathbf{x}, \mathbf{p}, \mathbf{v}, \vec{\Delta}, \hat{v}) = [\delta_x, \delta_y]^T. \tag{5.12}$$

In order to measure absolute shift caused by motion errors, the sub-aperture images must be compared to a “global” reference image generated with the correct motion parameters, as was performed in [42] for bistatic SAR where one of the platforms also operated as a monostatic SAR. In the case without a global reference, only the relative shifts between two sub-aperture images can be measured. The equations for the relative shifts are no less complicated than the absolute shifts and can similarly be found in closed form with a CAS using two instances of Eq. (5.7), each instance having a different sub-aperture position and velocity respectively, $(\mathbf{p}_a, \mathbf{v}_a)$ and $(\mathbf{p}_b, \mathbf{v}_b)$. The relative shift used in Eq. (5.3) can be described in functional form as

$$\hat{\xi}(\mathbf{x}, \mathbf{p}_a, \vec{\Delta}_a, \mathbf{p}_b, \vec{\Delta}_b, \mathbf{v}_a, \hat{v_a}, \mathbf{v}_b, \hat{v_b}) = f(\mathbf{x}, \mathbf{p}_a, \mathbf{v}_a, \vec{\Delta}_a, \hat{v_a}) - f(\mathbf{x}, \mathbf{p}_b, \mathbf{v}_b, \vec{\Delta}_b, \hat{v_b}). \tag{5.13}$$

The following section demonstrates the effectiveness of this autofocus method using this forward model with an optimization method.

5.4 Implementation and example application

To describe the implementation and demonstrate the efficacy of the autofocus algorithm this section is organized as follows: first an overview of the algorithm is provided, followed by a description of the data used in the demonstration. This is followed by a description of the cross-correlation calculations. Then the implementation of the forward model and the optimization is described. Finally the improved imagery is shown.

5.4.1 Implementation overview

The autofocus algorithm is described by Eq. (5.3), and its implementation consists of using the cost function with an appropriate optimization method that iteratively evaluates potential motion parameters until the maximum is found. Evaluations of the cost function described in Eq. (5.3), begin by using the forward model to estimate the shift of panel centers that result from processing a SAR image with the initial motion parameters when the proposed parameters were correct. The normalized cross-correlation panels are then interpolated to the point specified by the shift and the resulting cost is the accumulation of the interpolated points.

So far in this discussion, the motion measurement errors have been referred to as motion parameters. This implies some model for the motion error, which greatly influences the performance of the autofocus method. The model should properly represent the motion errors that cause the defocus, but if a very high order model is used the optimization may not converge in a reasonable time. Also some motion models may be overcomplete, such that there are many parameters that cause the same shifts. This is true for linear motion error because with linear motion the location of a target in range is ambiguous around the flight path. Thus when a motion model allows both range and azimuth velocities to be varied, the image shifts depend only on the net velocity error. For this demonstration the motion parameters in all dimensions are given by low order polynomials. To avoid ambiguities resulting from by linear-only motion errors, the model can be constrained to point in the initial azimuth direction.

The selection of the optimization method is critical to the performance and accuracy of the method. While the maximization problem described by Eq. (5.3) is not guaranteed to be smooth or have a single maximum, for our purposes, we have found that the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS-B) algorithm, which is included in the SciPy library [56], provides good results in a reasonable amount of time without requiring the derivative of the cost function. The calculation of the cost function derivative could significantly speed up the convergence of the method, but since the computation time of the optimization method on a desktop computer is similar to the processing time for the GPU backprojection processing of the image, processing time is not considered a significant concern for our use.

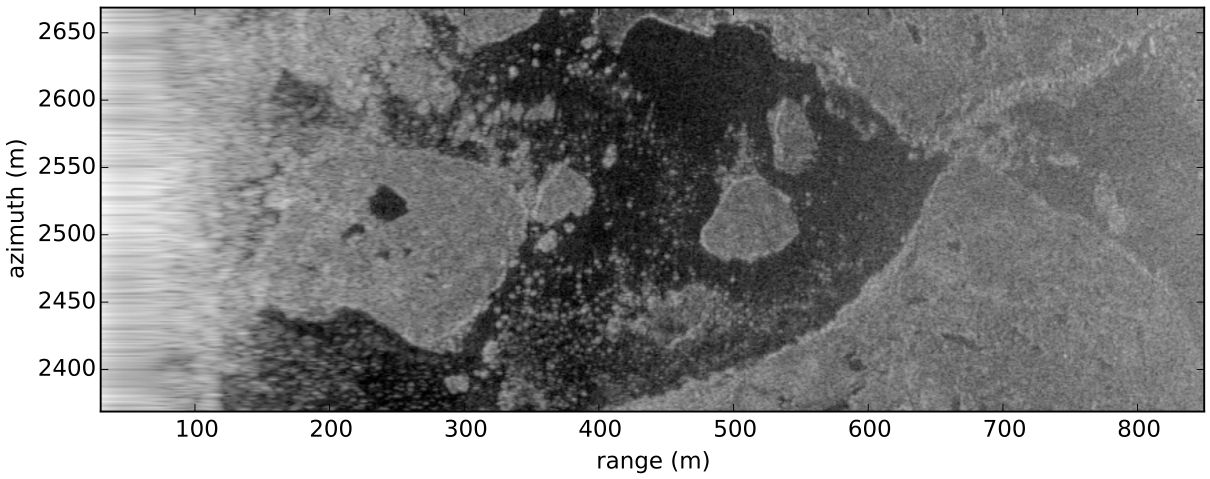


Figure 5.1: A well focused SAR image, processed using accurate motion parameters.

5.4.2 Example blurred image

In order to demonstrate the effectiveness of an implementation of this new autofocus method, a section of data collected as part of CASIE is used. In this demonstration 7 subapertures are used with 50% overlap. Figure 5.1 shows the well focused multi-looked image processed with motion estimated onboard a UAV with a high quality inertial navigation unit. During the data collection of this image, the UAV flight was fairly stable with a velocity around 30m/s. In order to clearly demonstrate the defocusing of the SAR image due to inaccurate motion parameters, the motion of the aircraft is adjusted with motion errors, resulting in the image shown in Fig. 5.2. The quadratic motion errors are chosen to highlight the ability of the algorithm to estimate three dimensional motion. The errors represent both velocity and acceleration mismatch from a non-linear flight path. The blurred image resulting from the incorrect motion parameters is shown in Fig. 5.3. While the induced motion error is a bit exaggerated from what is typically experienced by an aircraft, the simulation provides an excellent demonstration of the autofocus method's capabilities. Smaller and more linear errors can be estimated as well, but the effect of the defocus is difficult to see.

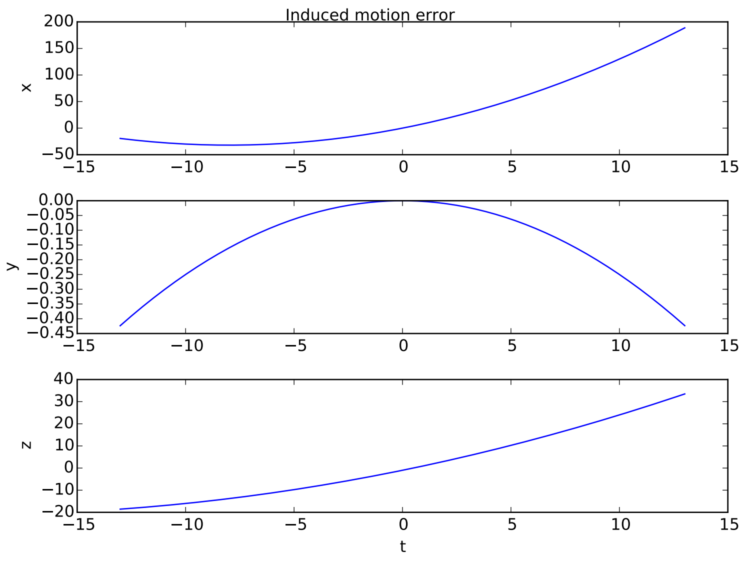


Figure 5.2: Induced motion error respectively in x, y, and z. The vertical axes are measured in meters and the horizontal axes in seconds. Over the course of this data segment, the aircraft flew a nominal 30 m/s.

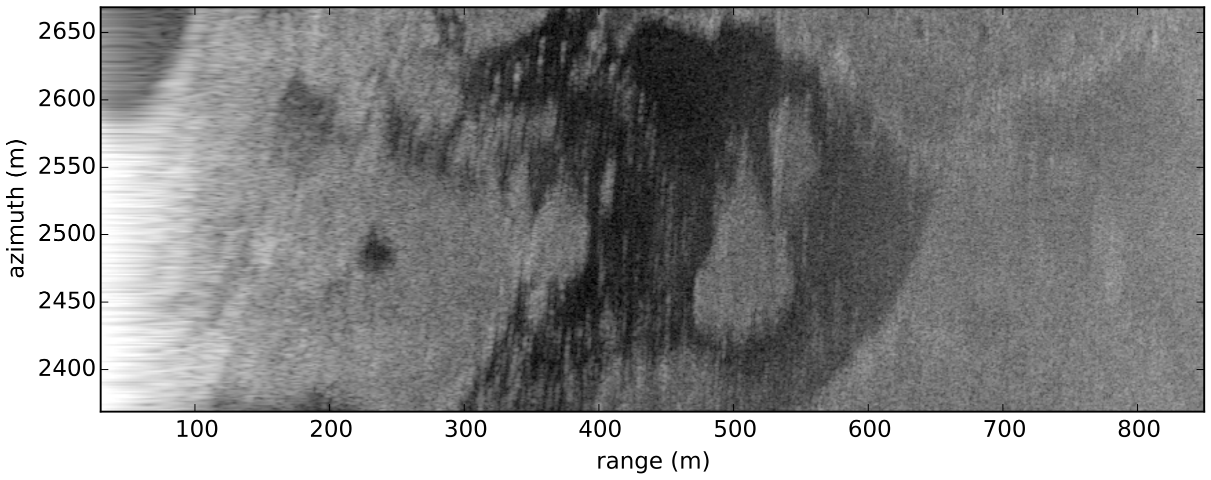


Figure 5.3: A blurred SAR image resulting from using incorrect motion parameters in the image processing.

5.4.3 Calculation of cross-correlation panels

To begin the autofocus method, the defocused image is broken into panels as shown for sub-apertures 2 and 5 in Fig. 5.4. Next, the normalized cross-correlations of the panels are

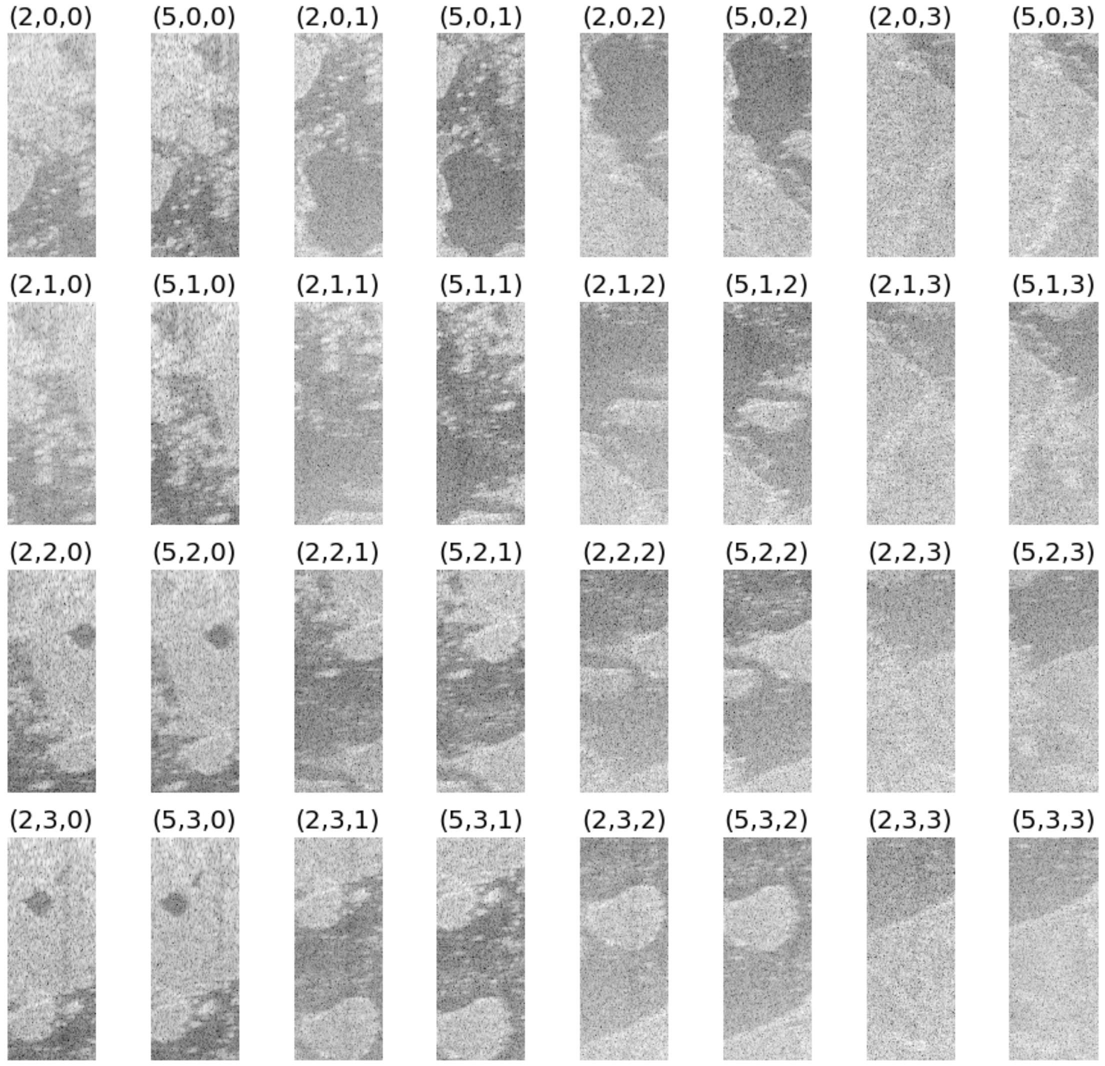


Figure 5.4: Example of the image panels for sub-apertures 2 and 5. The components of tuple above the image indicates the sub-aperture number and the range and azimuth panel indicies, respectively. Note that there is fifty percent overlap of the panels in range and azimuth.

computed. Care must be taken so that overlapping sub-apertures are not used in the cross-correlations, because the resulting cross-correlations are biased to zero by the overlapping data used to generate the panels. However, even though neighboring sub-apertures cannot be compared, a sub-aperture can be compared with sub-apertures that overlap each other. This provides finer temporal sampling than the comparison of only non-overlapping sub-apertures.

The calculation of the cross correlation begins with a zero-padded FFT of the panel magnitudes. One of the resulting spectrum panels is multiplied by the complex conjugate of the other. This is followed by an inverse DFT matrix multiply and a normalization window multiply. The DFT matrix multiply is used in order to efficiently calculate only a portion of the cross-correlation and it allows for sinc interpolation of the resulting cross-correlation. Alternatively a zero-padded inverse FFT could be used, but would require more computation because the shifts can be expected in a confined area. Figure 5.5 is an example of the normalized cross-correlation of the sub-aperture panels shown in Fig. 5.4. The x and y

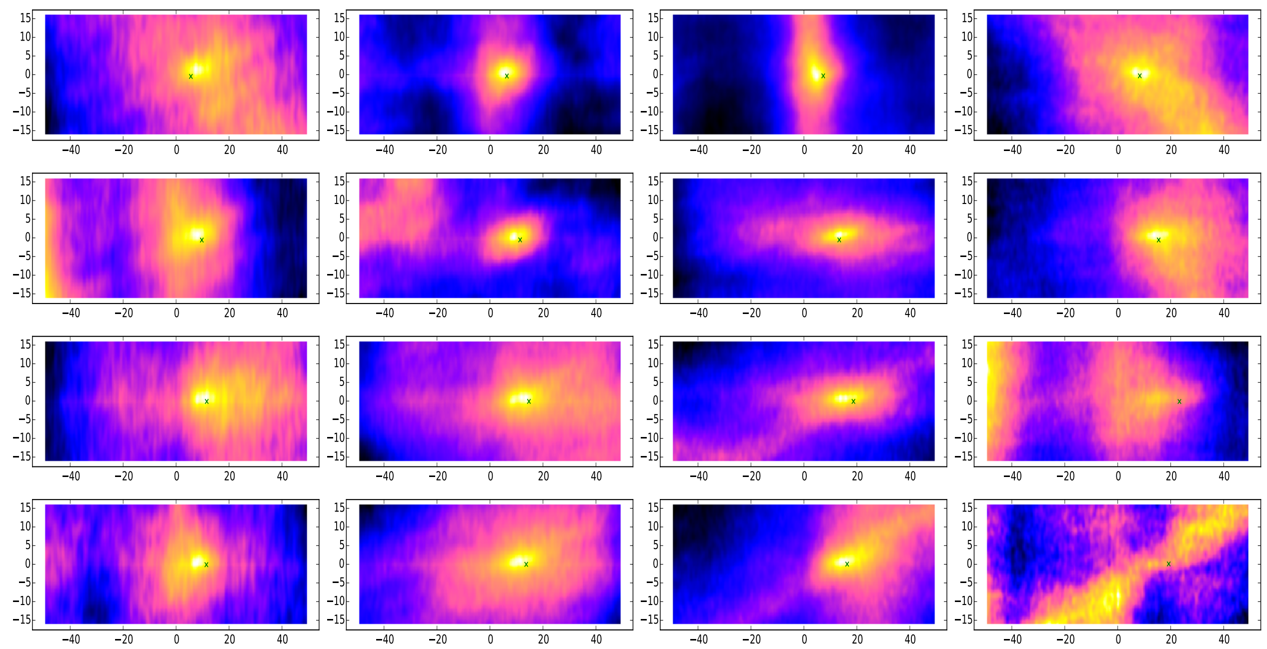


Figure 5.5: Example cross correlation panels of the panels for sub-aperture 2 and 5, which are shown in Fig. 5.4. The x and y axes, measured in meters, specify the shift in azimuth and range respectively. Note that the cross-correlations are only a portion of the full cross-correlation. The 'x' markers identify where the peak of the cross-correlation is expected as determined using the forward model on the induced motion shown in Fig. 5.2.

axes, measured in meters, specify the shift in azimuth and range respectively. Note that the cross-correlations are only a portion of the full cross-correlation. The 'x' markers identify where the peak of the cross-correlation is expected as determined using the forward model on the induced motion shown in Fig. 5.2. In most cases the marker lines up with the peak of the cross-correlation. The bottom right cross-correlation is an exception resulting from the lack of contrasting features in that panel.

5.4.4 Implementation of the forward model

As described by Eq. (5.13), the implementation of the forward model requires the position and velocity of the aircraft at the time the panel center is imaged. To provide this the backprojection kernel is modified to also calculate the effective image creation time for each pixel. This calculation is similar to the compensated backprojection, Eq. (3.14), and is given by

$$t_0(\mathbf{x}) \approx \frac{\sum_{m=0}^{M-1} W_{\mathbf{x}}^a[m] t[m]}{\sum_m W_{\mathbf{x}}^a[m]}, \tag{5.14}$$

where $W_{\mathbf{x}}^a[m]$ is the effective sub-aperture apodization window used in the backprojection processing, and $t[m]$ is the time at pulse m. The pulse time can be easily calculated within the GPU backprojection kernel with little performance penalty; however, care must be taken to avoid quantization effects from floating point calculations. With the effective imaging time for each sub-aperture image, the position and velocity parameters can be interpolated from the motion record. These are then used to calculate the expected relative shifts, described in Section 5.3.2, of each panel for all of the combinations of sub-apertures.

5.4.5 Results

Using the implementation of the forward model of the shift functions and an appropriate optimization method, motion parameters that could cause the defocus are found as shown in Fig. 5.6 and compared with the induced motion. The estimated motion parameters differ from the induced parameters, but have a similar shape. While the geometry causing the defocus is non-linear, the underlying data is collected along a nearly linear path so that the method's ability to uniquely identify the induced motion error is limited due to the ill-

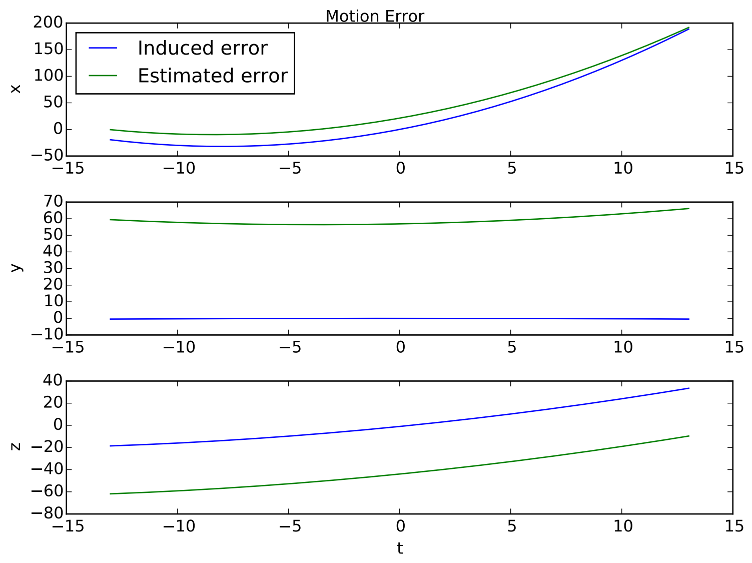


Figure 5.6: Estimated motion correction compared to the induced motion error.

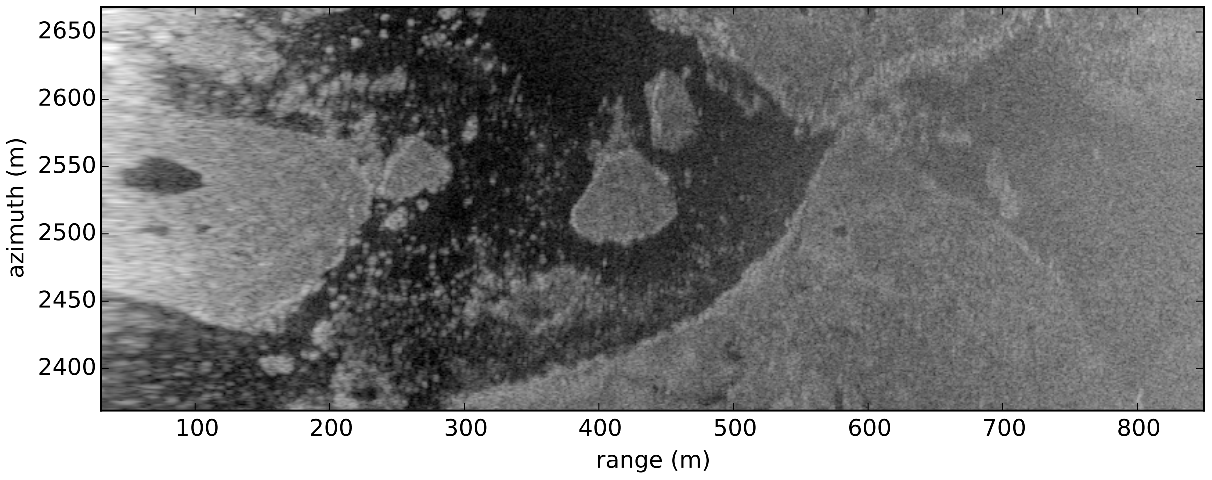


Figure 5.7: Refocused image using the estimated motion.

posedness of the problem. Although the estimated motion parameters do not exactly match the true parameters, using the estimated motion parameters to reprocess the data results in the well focused image shown in Fig. 5.7. Experimentally, we find this to be generally true. The refocused imagery demonstrates that the three dimensional mapdrift autofocus method presented in this chapter can be effective in overcoming defocus caused by motion measurement errors. Further experiments are needed to understand the bias of the estimated parameters from the true parameters.

5.5 Summary

In this chapter a novel autofocus algorithm is developed with a new forward model for the warping of an image caused by geometry errors. This method estimates motion parameters that improve the focus of the image, correcting both magnitude and phase errors. A demonstration of the effectiveness of the algorithm is provided using induced motion error on the CASIE SAR data.

Chapter 6

Dually Factorized Backprojection

For LFM-CW SAR systems with a fixed transmit and receive bandwidth, the maximum imageable range is proportional to the chirp length. Thus it can be desirable to have a long pulse; however, this violates the stop-and-hop assumption that is used in most traditional processing algorithms. Frequency domain and time domain methods have been developed to account for the motion during the pulse [4, 5, 57–59] including the work presented in Chapter 3; however these methods approximate the motion of the aircraft as linear during the pulse. This chapter presents a novel time-domain method that factorizes the correlation integral in both fast-time and slow-time, resulting in fast and accurate inversion accounting for general motion over the synthetic aperture and during a single pulse.

In the past, factorization of the SAR processing has been shown to be effective in reducing the computation of the SAR processing, but the factorization has only been done in azimuth [60]. In order to account for the motion of the aircraft during the chirp, in this work we explore factorizing in both range and azimuth. Factorizing in both range and azimuth allows for fast and accurate compensation of the motion during the chirp for LFM-CW SAR. Because this algorithm is factorized in both range and azimuth we call it the Dually Factorized Backprojection (DFBP). This work may be seen as a generalization of the Fast Factorized Backprojection (FFBP) formulation described in [60] and the FFBP is a special case of the DFBP.

6.1 Background

For convenience and clarity we begin with a slightly different presentation of the LFM-CW SAR signal and time-domain correlation processing. As described in Section 2.4

the recorded signal for an LFM-CW SAR is given by

$$s(\eta, t) = \int_{\mathcal{I}} I(x) A_x(\eta, t) \exp\{j(2\pi k_r t \tau_x(\eta, t) + \tau_x(\eta, t) 2\pi f_0 - \pi k_r \tau_x^2(\eta, t))\} \delta x. \tag{6.1}$$

The dechirped signal, Eq. (6.1), is bandpass filtered, which suppresses the transmit signal, reduces the data storage requirements, and effectively range gates the signal. In this form the visible range is a function of the bandpass filter and the chirp-rate. Thus in scenarios where the maximum transmit bandwidth is desired, the visible range is proportional to the chirp length. The time-domain correlation processor given by Eq. (2.14) correctly accounts for arbitrary motion during the pulse. For convenience, Eq. (2.14) is written in tensor form as

$$I = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} s[m, n] \cdot \exp\{-j\phi_{\mathbf{x}}[m, n]\}, \tag{6.2}$$

where

$$\phi_{\mathbf{x}}[m, n] = 2\pi k_r n / f_s \tau_{\mathbf{x}}[m, n] + 2\pi f_0 \tau_{\mathbf{x}}[m, n] - \pi k_r \tau_{\mathbf{x}}^2[m, n]. \tag{6.3}$$

Using virtually no approximations Eq. (6.2) generates the ideal reconstructed image; however it is also very computationally intensive, on the order of $O(MNL_aL_r)$ where L_a and L_r are respectively the number of azimuth and range pixels in the image, M is the number of pulses, and N is the number of samples in each pulse.

Previous work has shown how the summation over the fast-time index n can be separated using approximations to the motion of the aircraft resulting in a modified back-projection method [4, 58]. This chapter presents a method that accounts for the motion of the aircraft and greatly reduces computational burden of the SAR processing through a re-factorization of the correlation processing.

6.2 Dually Factorized Backprojection

This section describes a backprojection method that is factorized both in slow-time and fast-time, which we call Dually Factorized Backprojection (DFBP). As described in this section DFBP achieves significant computational advantages by refactoring the problem such that many computations may be combined in a similar manner to the Fast Fourier

Transform (FFT) and to the Fast Factorized Backprojection method [60], which performs a factorization in azimuth. For simplicity we represent the SAR image as I, dropping the index for the pixel location.

To describe the factorization used to accelerate the processing while maintaining the exact calculation, consider the following rearrangements of Eq. (6.2). First we define a notation to describe an image created with only a subset of the data as

$$I_{(c,d)}^{(C,D)} = \sum_{m=c\cdot C}^{((c+1)C-1)} \sum_{n=d\cdot D}^{((d+1)D-1)} s[m,n]\cdot \exp\{-j\phi_{\mathbf{x}}[m,n]\}, \tag{6.4}$$

where the superscripts C and D denote the size of the data subset and c and d are the slow-time and fast-time subset indices. In this notation the full image is described as $I_{(0,0)}^{(M,N)}$. Suppose that the subsets are selected such that the data is broken into K slow-time and L fast-time sections each of length C and D respectively, where $M = KC$ and $N = LD$, the full image can also be described as a summation of the subset images

$$\begin{aligned} I &= I_{(0,0)}^{(M,N)} \\ &= \sum_{k=0}^{K-1}\sum_{l=0}^{L-1} I_{(k,l)}^{C,D}. \end{aligned} \tag{6.5}$$

Extending this idea further, we describe the subset images recursively as summations of smaller subsets. In the case that M and N are factors of K and L we can define the recursion relation

$$I_{(c,d)}^{C,D} = \sum_{k=0}^{K-1}\sum_{l=0}^{L-1} I_{cK+k,DL+l}^{C/K,D/L}. \tag{6.6}$$

With the recursion relation the full image is generated with a maximum of the greater of $\log_K(M)$ or $\log_L(N)$ steps, where the lowest level $I_{(c,d)}^{1,1}$ represents an image created from a single sample of the SAR data. Note that at this point no approximations have been made to the correlation and Eq. (6.6) represents a refactoring of the summations; however, the refactoring of the correlation provides insights that can accelerate the processing.

6.2.1 Computational advantage

Each image described by Eq. (6.4) uses only a portion of the azimuth and modulation bandwidth such that they represent much lower resolution images than the full image. With much lower bandwidth, a fraction of the pixels contain the information necessary to reconstruct the other pixels, thus there is no need to perform the calculation of every pixel in the full image. Instead the subset images can be generated with much lower pixel density and then, because they have lower range and azimuth bandwidth, they can be interpolated to a higher pixel density with a fraction of the computation required to perform the full calculation. Furthermore with the recursion relation, the lower resolution images can be successively merged into higher resolution images until the full resolution image is created. This can lead to orders of magnitude computational advantages.

To explore the computational advantage, consider the generation of an image of dimensions $P \times P$ from radar data with M pulses each with M samples and the factorization size of K is used in both range and azimuth. At the initial stage the M^2 samples are processed using the time domain correlation described by Eq. (6.2) to generate M^2/K^2 sub-images each with H^2K^2 pixels, where H is an up-sampling factor used to provide better interpolation (like zero padding in an FFT). Thus the first stage requires $M^2K^2H^2$ evaluations of the correlation kernel. At the second stage K^2 adjacent sub-resolution images are each interpolated to a grid with K^4H^2 pixels, and because the number of pixels increases by the same factor that the number of subset images decreases, the number of computations required remains the same. This process is repeated for at most $\log_K P$ stages until all of the data has been merged or the desired image resolution is achieved. The total number of calculations is on the order of $\mathcal{O}(M^2H^2K^2 \log_K P)$, which is orders of magnitude lower than computational order of $\mathcal{O}(P^2M^2)$ required by direct correlation processing. Thus the speed-up over full correlation is lower bounded by the ratio

$$\frac{P^2M^2}{M^2H^2K^2 \log_K P} = \frac{P^2}{H^2K^2 \log_K P}, \tag{6.7}$$

which is near the same speed-up available from traditional FFBP methods. The lower bound is not achieved because, as discussed in the next section, the range interpolation is significantly more complex than the correlation processing. However, the speed-up over correlation processing can be several orders of magnitude as shown in Section 6.5.

6.3 Interpolation

Interpolation is key to the performance of DFBP both in terms of run-time and accuracy. Azimuth interpolation has been thoroughly discussed in the Fast Factorized Backprojection literature most notably [60–62], where it has been shown that traditional polynomial interpolation methods are satisfactory as long as the initial grid is generated with more azimuth pixels than necessary because the extra pixels act like zero padding. But the discussion of range interpolation is typically avoided due to the high carrier phase. In this section we derive interpolation methods based on the full time-domain correlation processing in order to accurately reconstruct the signal. We begin by discussing the spectral support of a SAR data collection in order to properly derive an accurate interpolation method for merging the sub-apertures.

Consider a pixel of a subset image in polar coordinates $I_{(c,d)}^{(C,D)}(r,\phi)$, where r and ϕ are the range and azimuth angle from the center of the effective sub-synthetic aperture given by the average of the antenna position. For this analysis we use the approximation that the aircraft speed is negligible to the speed of light. Let

$$r[m,n] = \|p[m,n] - x_{(r,\phi)}\|, \tag{6.8}$$

where $x_{(r,\phi)}$ is the location (r,ϕ) transformed to the three dimensional coordinates. Using Eq. (6.8), a single pixel in Eq. (6.4) can be written as

$$I_{(c,d)}^{(C,D)}(r,\phi) = \sum_{m=c\cdot C}^{((c+1)C-1)} \sum_{n=d\cdot D}^{((d+1)D-1)} s[m,n] \cdot e^{-j\frac{4\pi}{c_0}\left(\frac{k_r n}{f_s} r[m,n] + f_0 r[m,n] - r[m,n]^2 \frac{k_r}{c_0}\right)}. \tag{6.9}$$

Each sample in a lower resolution image contains some information about a value outside of the pixel center. This is because the radar samples a finite frequency band and so the spatial sampling is described by a sinc function whose support is infinite. Let us consider the contribution of a point target at an offset range position, r', described in a slightly different form from Eq. (6.1) as

$$s_{r'}[m,n] = I_{(c,d)}^{(C,D)}(r',\phi)A_{(r',\phi)}[m,n] \cdot \exp\left\{j\frac{4\pi}{c_0}\left(\frac{k_r n}{f_s}r'[m,n] + f_0 r'[m,n] - r'[m,n]^2\frac{k_r}{c_0}\right)\right\}, \tag{6.10}$$

where $r' = r + \delta_r$. Substituting Eq. (6.10) into Eq. (6.9) describes the contribution of a target at r' to the generated pixel centered at r described as

$$\begin{aligned} &I_{(c,d)}^{(C,D)}(r,\phi) \\ &= \sum_{m=c\cdot C}^{((c+1)C-1)} \sum_{n=d\cdot D}^{((d+1)D-1)} s_{r'}[m,n]\cdot\exp\left\{-j\frac{4\pi}{c_0}\left(\frac{k_r n}{f_s}r[m,n]+f_0 r[m,n]-r[m,n]^2\frac{k_r}{c_0}\right)\right\}. \end{aligned} \tag{6.11}$$

By limiting the slow-time extent we can use the small angle approximation

$$r'[m,n] = r[m,n] + \delta_r\,. \tag{6.12}$$

Substituting Eq. (6.12) into Eq. (6.10) yields

$$\begin{aligned} s_{r'}[m,n] &= I_{(c,d)}^{(C,D)}(r',\phi)A_{(r',\phi)}[m,n] \\ &\cdot \exp\left\{j\frac{4\pi}{c_0}\left(\frac{k_r n}{f_s}(r[m,n]+\delta_r) + f_0(r[m,n]+\delta_r) - (r[m,n]+\delta_r)^2\frac{k_r}{c_0}\right)\right\}. \end{aligned} \tag{6.13}$$

For convenience we ignore the amplitude function A and substitute Eq. (6.13) into Eq. (6.11) yielding

$$\begin{aligned} I_{(c,d)}^{(C,D)}(r,\phi) = &\sum_{m=c\cdot C}^{((c+1)C-1)} \sum_{n=d\cdot D}^{((d+1)D-1)} I_{(c,d)}^{(C,D)}(r',\phi) \\ &\cdot \exp\left\{j\frac{4\pi}{c_0}\left(\frac{k_r n}{f_s}(r[m,n]+\delta_r) + f_0(r[m,n]+\delta_r) - (r[m,n]+\delta_r)^2\frac{k_r}{c_0}\right)\right\} \\ &\cdot \exp\left\{-j\frac{4\pi}{c_0}\left(\frac{k_r n}{f_s}r[m,n] + f_0 r[m,n] - r[m,n]^2\frac{k_r}{c_0}\right)\right\} \\ = I_{(c,d)}^{(C,D)}(r',\phi) &\sum_{m=c\cdot C}^{((c+1)C-1)} \sum_{n=d\cdot D}^{((d+1)D-1)} \exp\left\{j\frac{4\pi}{c_0}n\left(\frac{k_r n}{f_s}\delta_r + f_0\delta_r \right.\right. \\ &\left.\left. - (2r[m,n]\delta_r + \delta_r^2)\frac{k_r}{c_0}\right)\right\} \\ = I_{(c,d)}^{(C,D)}(r',\phi) &\exp\left\{j\frac{4\pi}{c_0}\left(f_0\delta_r - (\delta_r^2)\frac{k_r}{c_0}\right)\right\} \\ &\cdot \sum_{m=c\cdot C}^{((c+1)C-1)} \sum_{n=d\cdot D}^{((d+1)D-1)} \exp\left\{j\frac{4\pi}{c_0}\left(\frac{k_r n}{f_s}\delta_r - (2r[m,n]\delta_r)\frac{k_r}{c_0}\right)\right\}. \end{aligned} \tag{6.14}$$

To make the next step we approximate the $r[m,n]$ term by the nominal range r from the center of the sub-aperture to the pixel and pull it out of the summation resulting in

$$\begin{aligned}
I_{(c,d)}^{(C,D)}(r,\phi) &\approx I_{(c,d)}^{(C,D)}(r',\phi) \sum_{m=c\cdot C}^{((c+1)C-1)} \exp\left\{ j\frac{4\pi}{c_0}\left(f_0\delta_r - (\delta_r^2 + 2r\delta_r)\frac{k_r}{c_0}\right)\right\} \sum_{n=d\cdot D}^{((d+1)D-1)} e^{j\frac{4\pi}{c_0}\left(\frac{k_r n}{f_s}\delta_r\right)} \\
&\approx I_{(c,d)}^{(C,D)}(r',\phi) C \exp\left\{ j\frac{4\pi}{c_0}\left(f_0\delta_r - (\delta_r^2 + 2r\delta_r)\frac{k_r}{c_0}\right)\right\} \sum_{n=d\cdot D}^{((d+1)D-1)} e^{j\frac{4\pi}{c_0}\left(\frac{k_r n}{f_s}\delta_r\right)} \\
&\approx I_{(c,d)}^{(C,D)}(r',\phi) C \exp\left\{ j\frac{4\pi}{c_0}\left(f_0\delta_r - (\delta_r^2 + 2r\delta_r)\frac{k_r}{c_0}\right)\right\} \\
&\quad \cdot \exp\left\{ j\frac{4\pi}{c_0}\left(\frac{k_r}{f_s}\left(dD + (D-1)/2\right)\delta_r\right)\right\} \sum_{k=\frac{-(D-1)}{2}}^{\frac{D-1}{2}} e^{j\frac{4\pi}{c_0}\left(\frac{k_r k}{f_s}\delta_r\right)},
\end{aligned} \tag{6.15}$$

where the final summation is over half integers and results in a Dirichlet kernel given by

$$\mathcal{D}_N(x) = \frac{\sin\left((N/2)x\right)}{\sin\left(x/2\right)}, \tag{6.16}$$

i.e.

$$I_{(c,d)}^{(C,D)}(r,\phi) \approx I_{(c,d)}^{(C,D)}(r',\phi) C \exp\left\{ j\frac{4\pi}{c_0}\left(f_{D,d}\delta_r - (\delta_r^2 + 2r\delta_r)\frac{k_r}{c_0}\right)\right\} \mathcal{D}_D\left(\frac{4\pi}{c_0}\frac{k_r}{f_s}\delta_r\right), \tag{6.17}$$

where $f_{D,d} = f_0 + \frac{k_r}{f_s}\left(dD + (D-1)/2\right)$ is the effective center frequency for the data segment. In deriving Eq. (6.17) the small angle approximation is applied multiple times; however, the effect on the final image is small because the approximation is applied locally to each pixel rather than the entire beam. Furthermore, the initial matched filtering includes all the generality of the time-domain correlation processing, and this derivation is only for the interpolation method. Using Eq. (6.17) we derive several interpolation methods as described in the following.

6.3.1 Nearest neighbor interpolation

Nearest neighbor interpolation results from solving Eq. (6.17) for $I^{(C,D)}_{(c,d)}(r', \phi)$ and ignoring the amplitude of the Dirichlet kernel is written as

$$I^{(C,D)}_{(c,d)}(r', \phi) \approx I^{(C,D)}_{(c,d)}(r, \phi) C \exp\left\{ -j\frac{4\pi}{c_0}\left(f_{D,d}\delta_r - (\delta_r^2 + 2r\delta_r)\frac{k_r}{c_0} \right) \right\}. \tag{6.18}$$

The nearest neighbor interpolation simply consists of finding the nearest neighboring value and applying a phase factor to account for the range difference.

6.3.2 Higher order interpolation

For better interpolation multiple points may be used with a least squares approach. The expression in Eq. (6.17) can be used with any r and δ_r to describe a number of points; however to ease the calculation we manipulate Eq. (6.17) as follows

$$\begin{aligned}
I^{(C,D)}_{(c,d)}(r, \phi) &\approx I^{(C,D)}_{(c,d)}(r', \phi) C \exp\left\{ j\frac{4\pi}{c_0}\left(f_{D,d}\delta_r - (\delta_r^2 + 2r\delta_r)\frac{k_r}{c_0} \right) \right\} \mathcal{D}_D\left(\frac{4\pi}{c_0}\frac{k_r}{f_s}\delta_r \right) \\
&\approx I^{(C,D)}_{(c,d)}(r', \phi) C \mathcal{D}_D\left(\frac{4\pi}{c_0}\frac{k_r}{f_s}\delta_r \right) \exp\left\{ j\frac{4\pi}{c_0}\left(f_{D,d}\delta_r - (\delta_r^2 + 2r\delta_r)\frac{k_r}{c_0} \right) \right\} \\
&\quad \cdot \exp\left\{ -j\frac{4\pi}{c_0}\left(f_{D,d}r - r^2\frac{k_r}{c_0} \right) \right\} \cdot \exp\left\{ j\frac{4\pi}{c_0}\left(f_{D,d}r - r^2\frac{k_r}{c_0} \right) \right\} \\
&\approx I^{(C,D)}_{(c,d)}(r', \phi) C \mathcal{D}_D\left(\frac{4\pi}{c_0}\frac{k_r}{f_s}\delta_r \right) \exp\left\{ j\frac{4\pi}{c_0}\left(f_{D,d}(r+\delta_r) - (r^2 + \delta_r^2 + 2r\delta_r)\frac{k_r}{c_0} \right) \right\} \\
&\quad \cdot \exp\left\{ -j\frac{4\pi}{c_0}\left(f_{D,d}r - r^2\frac{k_r}{c_0} \right) \right\} \\
&\approx I^{(C,D)}_{(c,d)}(r', \phi) C \mathcal{D}_D\left(\frac{4\pi}{c_0}\frac{k_r}{f_s}\delta_r \right) \exp\left\{ j\frac{4\pi}{c_0}\left(f_{D,d}r' - r'^2\frac{k_r}{c_0} \right) \right\} \\
&\quad \cdot \exp\left\{ -j\frac{4\pi}{c_0}\left(f_{D,d}r - r^2\frac{k_r}{c_0} \right) \right\}.
\end{aligned} \tag{6.19}$$

In this form, we can write an expression for multiple points in matrix form as

$$\begin{bmatrix} I_{(c,d)}^{(C,D)}(r_0,\phi) \\ I_{(c,d)}^{(C,D)}(r_1,\phi) \\ \vdots \\ I_{(c,d)}^{(C,D)}(r_P,\phi) \end{bmatrix} = I_{(c,d)}^{(C,D)}(r',\phi) e^{j\frac{4\pi}{c_0}\left(f_{D,d}r' - r'^2\frac{k_r}{c_0}\right)} C \cdot \begin{bmatrix} \mathcal{D}_D\left(\frac{4\pi}{c_0}\frac{k_r}{f_s}\delta_{r_0}\right) e^{-j\frac{4\pi}{c_0}\left(f_{D,d}r_0 - {r_0}^2\frac{k_r}{c_0}\right)} \\ \mathcal{D}_D\left(\frac{4\pi}{c_0}\frac{k_r}{f_s}\delta_{r_1}\right) e^{-j\frac{4\pi}{c_0}\left(f_{D,d}r_1 - {r_1}^2\frac{k_r}{c_0}\right)} \\ \vdots \\ \mathcal{D}_D\left(\frac{4\pi}{c_0}\frac{k_r}{f_s}\delta_{r_P}\right) e^{-j\frac{4\pi}{c_0}\left(f_{D,d}r_P - {r_P}^2\frac{k_r}{c_0}\right)} \end{bmatrix} + \mathcal{N}, \tag{6.20}$$

where P is the order of the interpolation schemed and $\mathcal{N}$ is a noise term that includes the approximation error and the contributions from other targets in the imaging scene. We use the system of equations given in Eq. (6.20) to generate an interpolation scheme by solving Eq. (6.20) for the least squares solution. (Alternative solutions may be used, but least squares is simple and gives a unique closed form solution). For a standard system of equations described by $Ax = b$, the least squares solution is given by the Moore-Penrose psuedoinverse for overdetermined equations and is written as

$$b = (A^H A)^{-1} A^H x, \tag{6.21}$$

where the superscript ^{H} denotes a Hermitian transpose. Using Eq. (6.21) with Eq. (6.20) a multiple point interpolation scheme can be written as

$$\begin{aligned} & I_{(c,d)}^{(C,D)}(r',\phi) \\ & \quad \approx \exp\left\{j\frac{4\pi}{c_0}\left(f_{D,d}r' - r'^2\frac{k_r}{c_0}\right)\right\} \\ & \quad\quad \cdot \frac{1}{C}\frac{\sum_k \mathcal{D}_D\left(\frac{4\pi}{c_0}\frac{k_r}{f_s}\delta_{r_P}\right)\exp\left\{-j\frac{4\pi}{c_0}\left(f_{D,d}r_P - {r_P}^2\frac{k_r}{c_0}\right)\right\}\cdot I_{(c,d)}^{(C,D)}(r+k\Delta_r,\phi)}{\sum_k \|\mathcal{D}_D\left(\frac{4\pi}{c_0}\frac{k_r}{f_s}\delta_{r_P}\right)\exp\left\{-j\frac{4\pi}{c_0}\left(f_{D,d}r_P - {r_P}^2\frac{k_r}{c_0}\right)\right\}\|^2} \\ & \quad \approx \exp\left\{j\frac{4\pi}{c_0}\left(f_{D,d}r' - r'^2\frac{k_r}{c_0}\right)\right\}\frac{1}{C}\frac{\sum_k \exp\left\{-j\frac{4\pi}{c_0}\left(f_{D,d}r_k - {r_k}^2\frac{k_r}{c_0}\right)\right\}\cdot I_{(c,d)}^{(C,D)}(r_k,\phi)}{\sum_k \|\mathcal{D}_D\left(\frac{4\pi}{c_0}\frac{k_r}{f_s}\delta_{r_k}\right)\|}. \end{aligned} \tag{6.22}$$

In this form note that the complex exponential in the numerator summation effectively shifts the image signal to base band while the complex exponential outside of the summation shifts the signal back so that the recursive sum can continue. Equation (6.22) represents a truncated Dirichlet interpolation, which gives the idea that the interpolation can be accomplished via FFTs; however, because the interpolation points are not equally spaced and the full subimage may not contain the full support of the Dirichlet kernel, a simple application of an FFT followed by a zero padded IFFT is not sufficient for the desired interpolation. Fortunately, both of these challenges have known solutions. The former challenge is addressed by the Chirp-Z transform [63] and the latter by the NERFFT [30]. It is likely that a combination of the two algorithms could overcome these challenges, but because we have used a small angle approximation in developing this scheme it may be more important to keep the order of the interpolation small in order to minimize the approximation error.

6.3.3 Modified polynomial interpolation

In the form described by Eq. (6.22) the phase correction is separate from the magnitude interpolation. By processing the first stage of the images at a higher density than required to maintain the signal integrity, we increase the number of samples within the main lobe of the Dirichlet kernel. Because the main lobe of the Dirichlet kernel is smooth the interpolation of the kernel main lobe can be performed with any traditional interpolation method, which can be much more computationally efficient than Dirichlet interpolation. A custom spline interpolation can be calculated to improve the accuracy using the main lobe of the Dirichlet kernel. However as we show later, traditional spline interpolation methods are sufficient when paired with upsampling the initial stage of the DFBP, which has the effect of zero-padding the processing.

6.3.4 Polar coordinates

The major task of factorized backprojection methods is the interpolation of lower resolution images onto higher resolution grids as shown for polar coordinates in Fig. 6.1. The interpolation can be performed on rectangular grids, but because the parameters of the Dirichlet kernel are related to the range, it can be more efficient to generate the images in polar coordinates so that interpolation to higher resolution images is easily accomplished. In DFBP we must account for the range change, as shown in the previous section. In this section we describe the polar grids and how the grids can be calculated while accurately accounting for the non-linear flight path of the platform. Because there are many levels of sub-apertures, in the following discussion we refer to the sub-aperture to be generated as the aperture and the sub-apertures that are used to generate the higher resolution aperture as sub-apertures.

The polar grids used for interpolation are shown in Fig. 6.1. A polar grid is equally

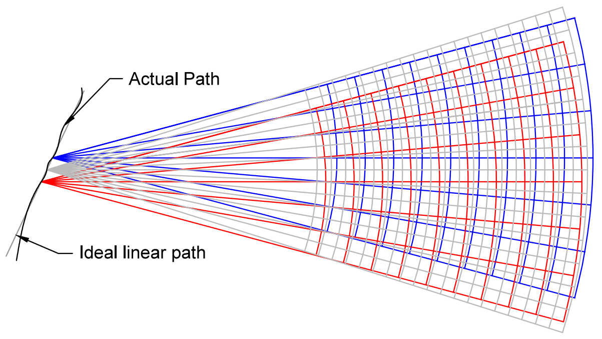


Figure 6.1: Example polar geometry for merging two sub-apertures (red and blue) to a higher resolution aperture (gray) in DFBP.

spaced in azimuth angle and range from a chosen reference point and axis. The mean of the sub-aperture positions is selected as the center of the polar grid for the aperture to be created because it is the approximate phase center for new synthetic aperture. The orientation of the grid is chosen to be parallel to the line between the first and last sub-apertures.

The generation of a higher resolution image is accomplished by interpolating each of the lower resolution images to a higher resolution grid. This involves calculating the polar coordinates of the higher resolution images in each of the sub-aperture images' reference frame. In the case of nearly linear flight where the sub-aperture centers and their respective reference axes lie on a line, the polar coordinates can be found using the law of cosines as described in [60]. The geometry for this simplified case is shown in Fig. 6.2. The range is given by

$$r_\alpha = \sqrt{\alpha[m,n]^2 + r^2 - 2\alpha[m,n]r\cos(\pi/2 - \psi)}, \tag{6.23}$$

where α is the distance between the new aperture center and the sub-aperture, r is the range from the aperture center, r_α is the range from the sub-aperture reference frame. The azimuth angle from the sub-aperture reference is given by the law of sines as

$$\sin\left(\pi/2 - \psi_\alpha\right) = r\frac{\sin\left(\frac{\pi}{2} - \psi\right)}{r_\alpha}, \tag{6.24}$$

which can be rewritten as

$$\cos\left(\psi_\alpha\right) = r\frac{\cos\left(\psi\right)}{r_\alpha}. \tag{6.25}$$

Thus the angle is given by

$$\psi_\alpha = \cos^{-1}\left(r\frac{\cos\left(\psi\right)}{r_\alpha}\right). \tag{6.26}$$

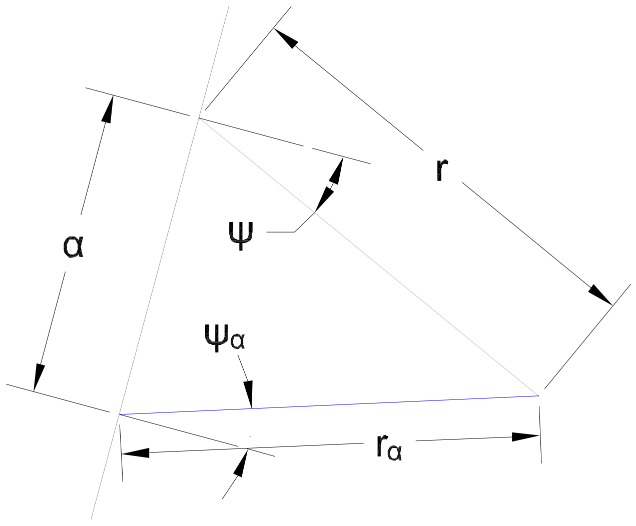


Figure 6.2: Simplified geometry for a linear aperture

For more general trajectories the topography of the scene must be accounted for in order to accurately focus the imaging scene. To account for the topography with a polar grid, the polar coordinates are transformed into Cartesian coordinates as mentioned in [64]; however, the grid spacing remains equally spaced in range and azimuth. The transformation of the two dimensional polar coordinates to three dimensional Cartesian coordinates is non-linear, and when scene has significant topography, the mapping is not unique. (The case of non-unique mapping results in what is referred to as layover in the literature). In the case of a planar imaging surface, a closed form expression can be found by the intersection of the range sphere, Doppler cone, and imaging plane. Ignoring layover areas, a greedy algorithm can be used to solve for the locations of a general surface, using a piecewise planar surface model to initialize the algorithm. Once the Cartesian coordinates for the grid have been found, the distances from each of the sub-aperture images can be done in Cartesian coordinates to the new polar grid.

As demonstrated by the FFBP methods, the use of a polar coordinate system can greatly reduce the computational burden. But as described in this section, in order to account for non-linear motion, the computational advantage of polar coordinates is greatly reduced because the image must be transformed into cartesian coordinates.

6.4 Implementation

The implementation of DFBP can be accomplished in many different ways for various criteria. The main decision is in the selection of the coordinate system and interpolation methods. In this section we describe and implementation using cartesian coordinates. The cartesian coordinate implementation has the advantage of easily accounting for the topography and non-linear motion of the aircraft, while a polar coordinate version could be more computationally efficient for nearly linear flight paths. A combination of these two implementations is possible, potentially resulting in a fast and accurate algorithm for flight paths that are nearly linear for short sections. The recursion relation set by Eq. (6.6) gives a framework for either implementations. After describing how Eq. (6.6) is implemented, we describe some key aspects of the implementation in order to achieve reasonable accuracy and run-time performance.

The implementation for the DFBP is outlined in the non-executable Python code in Algorithm 2. The DFBP process follows by taking the full data and recursively splitting the data into K azimuth data sections and L range segments until the data segments are each K by L in size. At the base case full correlation processing is performed as described by Eq. (6.2) to $H_K K$ by $H_L L$ pixels, where H_K and H_L are factors that increase the pixel spacing of the initial grid. Upon return of the recursion each sub-resolution image is interpolated to a

Algorithm 2 Non-executable Python code for Dually Factorized Backprojection.

```
def dfbp(data, pos, patch, K, L, upK=4, upL=8, m0=0, n0=0):
    M,N= data.shape
    if data.shape == (K,L):
        # base case: correllation processing
        subpatch= patch.interp(M*upK,N*upL))
        subimg, subcenter= correlationProcessing(subpatch, data,
                                                 pos, m0= m0, n0= n0 )
        return subimg, subcenter, subpatch
    #recursive case
    C= int(M/K)
    D= int(N/L)
    for m in range(K):
        for n in range(L):
            subimgs[m,n], subcenters[m,n], subpatchs[m,n]= dfbp(
                                 data[m*C:(m+1)*C, n*D:(n+1)*D],
                                 pos, patch, m0= m0+m*C, n0= n0+n*D )
    # interpolate and add the images together into a new patch
    newcenter/= K * L
    newpatch= patch.interp( subimg[0,0].shape * [K,L] )
    img= zeros( newpatch.shape[:2], dtype=complex)
    for m in range(K):
        for n in range(L):
            upsubimg =sarinterp(subimgs[m,n], n0+n*D, D,
                                  subcenters[m,n], subpatchs[m,n],
                                  newpatch)
            img+= upsubimg
    return img, newcenter, newpatch

def sarinterp(img, n0, D, center, oldpatch, newpatch):
    fd= calculateCenter( n0 , D )
    oldR= sqrt( sum( (center - oldpatch)**2 ) )
    oldphase= calculate_phase( fd, oldR )
    newR= sqrt( sum( (center - newpatch)**2 ) )
    newphase= calculate_phase( fd, newR )
    return (newphase**-1) * interpolate( oldphase * img )
```

higher resolution by the factors K and L and the interpolated images are combined. The combined image is then returned to the parent process.

6.4.1 Factorization step sizes

The factorization step sizes K and L have a significant effect on the runtime and accuracy of the processing. This is analogous to how the factorization size, or "radix", of an FFT implementation can have a dramatic effect on performance. Similar to the cursory analysis given in [65] the factorization $K = 2$ maximizes the speed-up factor given by Eq. (6.7); however, like the selection of an optimal FFT radix, there are many other details that must be considered to identify the optimal factorization [66]. For example, the initialization of the interpolation method was not considered as a computational cost in the analysis in Section 6.2.1 but it is not insignificant. Also to minimize the error due to the small angle approximation used in the interpolation stage, it is advantageous to process the range direction faster than the azimuth direction. Furthermore, the shape of the data is not typically square and so the factorization steps differ at some point in the processing. A full analysis of the factorization is beyond the scope of this work, but in practice it has been found that factorization step sizes of $K = 2$ and $L = 4$ perform well. It is expected that similar to the FFT, optimal performance is found with both compile-time and runtime optimizations.

Also note that in the analysis, the data subsets are assumed to be of equal size but that this is not a limitation of the method. In the implementation of the method, the data subsets may differ in order to accomodate non-rectangular data sets as well as cases where M and N are not factorable by small integers.

6.4.2 Image surface interpolation

At each stage of the processing, the image is progressively generated at a higher resolution. For the Cartesian coordinate implementation the grid is a representation of the imaging surface. We implement the interpolation with a rectangular bivariate spline with three knots on each side. In order to minimize the runtime of the grid interpolation the interpolation function is “memoized”, meaning the results of the interpolation are saved so that on subsequent DFBP calls for the same stage of processing, the interpolation of the grid is not recalculated.

6.4.3 SAR data interpolation

For the cartesian implementation, the SAR data is interpolated with the modified polynomial interpolation without consideration for the shape or orientation of the Dirichlet kernel. This makes the interpolation easy to implement, but the interpolation error is significant if the samples are spaced too sparsely. Thus it is critical to upsample the initial image grids. Also because only the main lobe of the Dirichlet kernel is smooth, the number of useful knots in a polynomial interpolation method is limited by the upsampling factors.

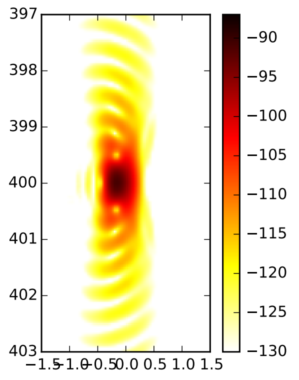
An advantage of the polar formatting is that the calculation of the range and the phase terms are identical for all of the range lines in a sub-resolution image as well as for each of the sub-resolution images when the flight path is nearly linear. Thus in the range interpolation, the square root and exponential calculations are negligible.

6.4.4 Multi-look image generation

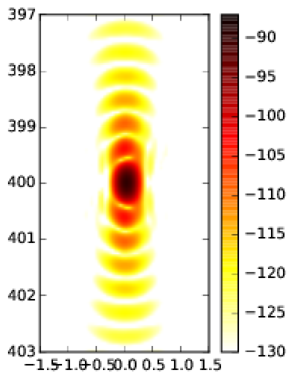
The generation of multi-look images is important for many SAR applications. Among many uses, multi-look images can be used to identify moving targets and reduce speckle. As shown in [67] sub-azimuth beam images can be performed simultaneously in traditional backprojection; however due to the structure of the DFBP this is not feasible for stripmap SAR. The sub-azimuth images can be generated by running the DFBP with several different azimuth windows. For example, to generate four independent sub-aperture images the DFBP is run with four azimuth windows equally spaced in the full SAR beamwidth, each having only a quarter of the full azimuth beamwidth. Alternatively another form of multi-looking can be applied in which the image is generated at the maximum azimuth resolution and then adjacent pixels are used similarly to the adjacent beam images.

6.5 Simulation

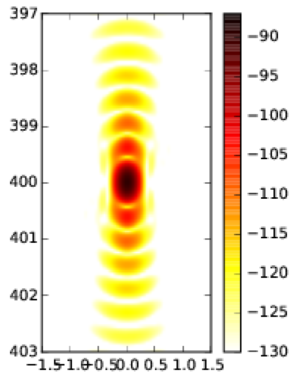
To demonstrate the effectiveness of the DFBP, we simulate a UHF radar (800MHz center frequency) with 500MHz bandwidth and aircraft velocity of 150 m/s similar to the simulation in [58]. Figure 6.3 shows the reconstruction of a simulated scene with a single point target with four time-domain algorithms. Figure 6.3d is processed with correlation processing, which provides the ideal image, but has a tremendous computational burden. As seen in Fig. 6.3c the impulse response function for the DFBP is nearly identical to the correlation processing. For comparison the image processed using traditional backprojection using the stop-and-hop assumption and using the UWB correction from [58] is shown in Fig. 6.3a and Fig. 6.3b respectively.



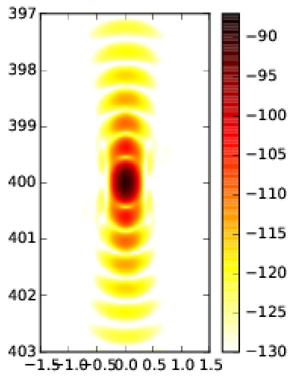
(a) Stop and hop



(b) UWB-BP



(c) DFBP



(d) Full Correlation

Figure 6.3: Single target response of an UWB wide-beam width SAR using (a) the stop-and-hop approximation, (b) the UWB correction described in [58], (c) the DFBP, and (d) full correlation processing.

6.5.1 Performance evaluation

If the interpolation and correlation processing stages are equally complex, the speed-up achievable by the DFBP formulation is given by the ratio of the operations required by correlation processing and the DFBP; however, due to the simplicity of correlation processing the speed up is only achieved for large images. In order to evaluate the speed-up achieved, a larger image area is simulated as shown in Fig. 6.4. The speed-up achieved is dependent on the product of the upsampling factors as shown in Fig. 6.5a. On the other hand as illustrated

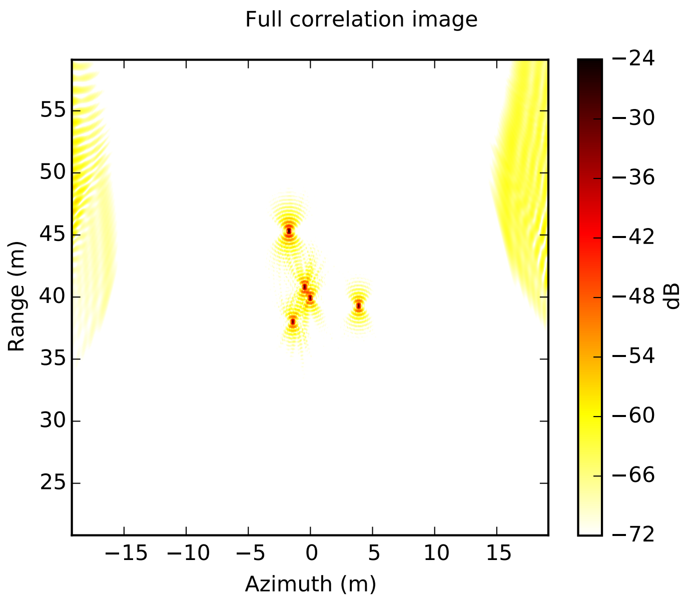
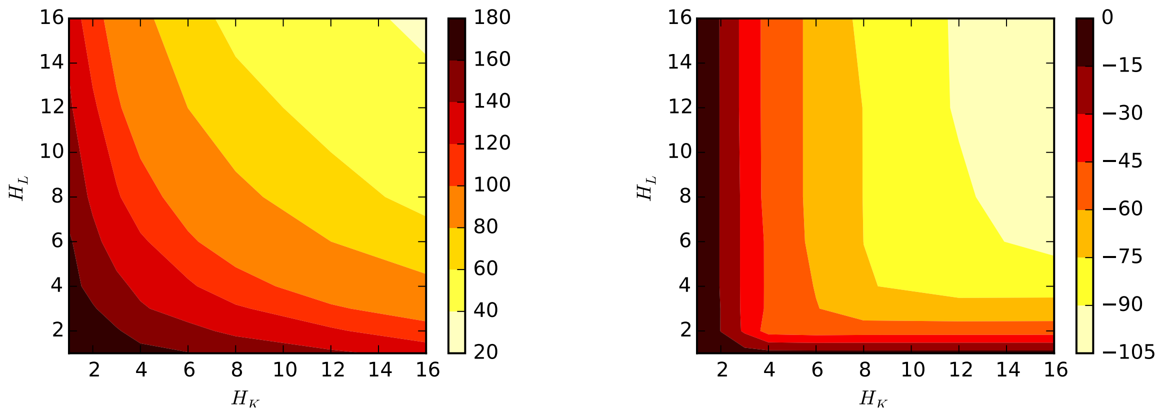


Figure 6.4: Example large simulation.



(a) Speed-up of DFBP

(b) Reconstruction Error

Figure 6.5: Speed-up and maximum error of the DFBP implemenation in cartesian coordinates over correlation processing as a function of the upsampling factors.

in Fig. 6.5b, the error of the reconstructed image lowers when the increase of the upsampling factors is balanced.

6.6 Summary

This chapter describes the DFBP, a novel SAR image formation algorithm based on factorizing correlation processing in slow-time and fast-time. The method fully accounts for the motion of the antenna during a pulse. The motion of the aircraft is not restricted to straight paths, and the algorithm can be directly applied to non-linear flights including circular SAR. Furthermore, the algorithm is suitable for ultra-wide-band ultra-wide-beam SAR systems. This work generalizes the FFBP, and like FFBP methods, theoretically, the computation order the algorithm is on nearly on par with frequency domain methods. The effectiveness of the algorithm is demonstrated via simulation of a UWB SAR system, and it is shown that the difference between the DFBP and full time-domain correlation can be made negligible with appropriate upsampling.

Chapter 7

Conclusion

The chapters of this dissertation address some of the challenges and opportunities of operating SAR on small UAVs; however this work is also very applicable to manned and satellite based SAR systems as well. Throughout this work, careful consideration has been made for ultra-wide band SAR. Chapter 1 introduces the work of this dissertation, and Chapter 2 provides the background of SAR imaging necessary to understand the following chapters. Chapter 3 presented the design and implementation of a SAR backprojection processor that takes into account all of the parameters necessary to produce high-resolution, radiometrically accurate stripmap SAR images from a UAV. Careful attention in the implementation was given to achieve real-time performance utilizing the unique architecture of a GPU. Chapter 4 explored the requirements for estimating topography from a single SAR channel, extending the spectral analysis techniques used in spotlight SAR to stripmap SAR, and it was found that the motion requirements are impractical for operation from fixed wing aircraft. In Chapter 5 a novel autofocus method was presented that estimates three dimensional motion parameters that improve the image focus. This is accomplished by extending mapdrift autofocus methods to three dimensions, using a novel forward model for the relative shifts induced by motion errors. By estimating motion parameters, the autofocus method can correct for both errors in magnitude and phase, which is particularly important for UWB SAR. Finally, Chapter 6 presented the Dually Factorized Backprojection method, which is a generalization to factorized backprojection methods and provides motion compensation during a pulse. Much of this work was conducted to support the Characterization of Arctic Sea Ice Experiment (CASIE), and the appendices provide substantial contributions for this project as well, including: 1. My work in designing and implementing the digital receiver and controller board for the microASAR which was used for CASIE. 2. A description of how the GPU backprojection was used to improved the CASIE imagery. 3. A description of a sample SAR data set from CASIE provided to the public to promote further SAR research.

7.1 Contributions

In greater detail, the contributions of this dissertation are summarized as follows:

7.1.1 Backprojection for UAV operation

Traditional frequency domain SAR processing algorithms fail to fully compensate for the motion of a small UAV. While traditional backprojection processing can fully compensate for the motion pulse to pulse; motion during the pulse is neglected resulting in image distortion. This work extends work by Ribalta [4] and Zaugg [5] to ultra wide-band SAR in order to account for linear motion of the aircraft during a pulse.

The significant motion of the aircraft experienced by a UAV causes some pixels in an image to be reconstructed from more samples and, due to gain variations, greater contributions from some pixels than other pixels. This results in varying intensity across an image. In order to remove the variation of intensity, a compensation factor is derived based on the radar equation and the backprojection kernel that results in a calibrated σ^0 image. The compensation factor is a summation of the product of the apodization window weighting, the antennae gains weightings, and the propagation loss weighting for each pulse, and can be calculated along side the backprojected image. This compensation factor allows for the generation of calibrated and radiometrically accurate images.

7.1.2 Real-time GPU implementation of backprojection

While backprojection imaging provides superior focusing, it has been avoided in the past due to its high computational cost; however, in many operations the backprojection computation can be performed in real-time with the use of modern GPUs. While other researchers have provided GPU implementations, Chapter 3 describes an implementation of real-time backprojection on an NVIDIA GPU that is appropriate for UAV stripmap SAR, and provides a novel implementation of the NERFFT utilizing the GPU texture cache, providing a fast and accurate reconstruction of the SAR image scene. The performance of the GPU backprojection processor is faster than real-time for the CASIE data.

7.1.3 Exploration of cross-track single antenna interferometry

It was suggested that it may be possible to use sub-aperture images from different altitudes along the aircraft flight path with oscillatory motion, similar to the CASIE flights, to perform interferometry and infer the topography of the sea ice. Chapter 4 explores this proposal and extends the spectral analysis developed for traditional spotlight SAR interferometry [13,38] to backprojection images and determines the requirements for single antenna interferometry to infer topography. The requirements for coherence between two SAR images is clarified, and it is shown that the requirements for single antenna interferometry are impractical for a fixed wing aircraft.

7.1.4 Three dimensional mapdrift autofocus

The accuracy of the available motion measurements for an aircraft flight limits the ability to properly focus a SAR image. This is particularly true for small UAVs where large inertial navigation units cannot be used. In order to improve the focus of the image, the autofocus method presented in Chapter 5 estimates the motion errors in three dimensions in order to improve the focus of the image. This extends traditional mapdrift methods to three dimensional motion errors. This method is particularly useful in UWB SAR imaging, where the amplitude errors caused by the motion errors are as significant as the phase errors.

7.1.5 Dually Factorized Backprojection

As the length of an LFM-CW pulse is extended, even the linear motion approximation during a pulse leads to image distortion. Chapter 6 describes a novel factorization of correlation processing in both range and azimuth. We call this method the Dually Factorized Backprojection method, because it generalizes factorized backprojection methods by factorizing in fast-time as well as slow-time. This leads to a fast and accurate SAR processing method that accounts for general motion during the pulse.

7.1.6 Characterization of Arctic Sea Ice Experiment

The previously described work represents the main contributions of this dissertation, but because much of this work was developed in connection to CASIE, the appendices provide a substantial contribution as well. Appendix A describes how the GPU backprojection was used with the CASIE data to greatly improve the imagery. Appendix B is a description of a sample CASIE SAR data set and example processing methods published on the BYU MERS website. This is one of the few publicly available SAR data sets, and to our knowledge the only available dataset that is from an LFM-CW SAR or collected on a UAV. Finally Appendix C presents my work in designing and implementing the digital receiver and controller sub-system for an LFM-CW SAR. This FPGA based system is highly configurable and has been used for a variety of operations including airborne, UAV, and land based.

7.2 Suggested future research

While working on the research described in this dissertation many other novel research avenues were identified, and the following are suggested for future work:

- This work shows that modern GPUs can be used to perform backprojection processing in real-time in many cases, but there remains a need for more efficient algorithms because the order of the backprojection computation is higher. As the number of image pixels or samples increases the number of operations required by backprojection increases significantly faster than required by Fourier domain methods. While fast factorized backprojection methods have the same big-O complexity, the Fourier domain methods still remain significantly faster. In order to design a SAR processor that is both efficient and accurate it is likely that hybrid methods can be designed that use the Fourier domain methods when appropriate and use time domain methods to combine lower level images into the final product.

- While it was shown in Chapter 4 that the topography of a scene cannot be estimated using interferometric techniques from a single fly-by, single beam SAR, because unknown topography of a scene results in defocus of the image from a non-straight flight path, it may be possible to infer the topography using autofocus techniques. In the case that the flight path can be accurately approximated as piecewise linear, topography estimation could use stereometric techniques. However, the stereometric techniques and analyses would need to be extended because stereometry has typically been performed with either parallel or perpendicular flight paths [54].

- In computer vision it is known that while two cameras with known separation allow for the three dimensional estimation of a object, with three cameras both the position of the object and the cameras can be estimated. Similarly autofocus and interferometry might be combined to estimate both the topography of the scene and the motion of the platform. Because the phase difference of properly focused SAR images is near zero [68], a combination of an along-track and multiple cross-track baselines might allow for both motion errors and topography to be estimated.

- What waveform a radar should transmit has been an ongoing question since the early beginnings of radar imaging. While the LFM-CW radars have many attractive features including high-resolution and low power and that the feed-through component can be filtered off in analog hardware, there are other signal modulation schemes that are often desirable. For example in some cases it may be desirable to use a communications link as the transmit waveform, or a more general pseudo-random waveform may be desired due to its thumbtack ambiguity function and because it would be more difficult for an outside observer to detect. A radar system designed along the same design as the recently proposed full-duplex radio [69] could allow for arbitrary waveform selection.

- Motion compensation for bistatic SAR is even more necessary for proper focusing of the SAR image than it is for monostatic SAR, and the DFBP could be modified to enable improved focusing of bistatic SAR images.

- While pseudo random waveforms have many desirable features, many of the features such as the thumbtack ambiguity function require that the image be processed with correlation processing, which is extremely computationally expensive. It is likely that the DFBP can be extended to some forms of pseudo-random waveforms.

7.3 Closing remarks

The field of SAR imaging is an exciting field of research, and is greatly expanding with developments including the contributions of this dissertation. Furthermore, as the suggested future work describe, the field is ripe for many contributions for many years to come. To help support future work, we provide a sample data set of real LFM-CW SAR data from CASIE with example processing code to the public at `http://www.mers.byu.edu/microASAR/CASIE_sample/`

Bibliography

[1] E. Zaugg, D. Hudson, and D. Long, "The BYU SAR: A small, student-built SAR for UAV operation," in *IEEE International Conference on Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006*, 2006, pp. 411–414. 1, 105, 122

[2] M. Edwards, D. Madsen, C. Stringham, A. Margulis, B. Wicks, and D. Long, "microASAR: A small, robust LFM-CW SAR for operation on UAVs and small aircraft," in *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*, vol. 5, July 2008. 1, 105, 107, 122, 126

[3] C. Stringham, D. Long, B. Wicks, and G. Ramsey, "Digital receiver design for an offset IF LFM-CW SAR," in *Radar Conference, 2011 IEEE*, May 2011. 1, 31, 109

[4] A. Ribalta, "Time-domain reconstruction algorithms for FMCW-SAR," *Geoscience and Remote Sensing Letters, IEEE*, vol. 8, no. 3, pp. 396 –400, May 2011. 2, 12, 18, 21, 34, 36, 69, 70, 89

[5] E. Zaugg, "Generalized image formation for pulsed and LFM-CW synthetic aperture radar," Ph.D. dissertation, Brigham Young University, 2010. 2, 69, 89

[6] D. G. Long, E. Zaugg, M. Edwards, and J. Maslanik, "The microASAR experiment on CASIE-09," in *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, Jul. 2010, pp. 3466 –3469. 3, 31, 100

[7] M. Cheney and B. Borden, *Fundamentals of radar imaging.* Society for Industrial Mathematics, 2009, vol. 79. 5, 14

[8] F. T. Ulaby, D. G. Long, A. K. Fung, and R. K. Moore, "Microwave remote sensing modern edition," *Artech House, Norwood, MA*, 2013. 5, 14

[9] I. Cumming and F. Wong, *Digital Processing of SAR Data.* Norwood, MA: Artech House, 2005. 5, 12, 14, 50, 51

[10] W. Melvin and J. Scheer, *Principles of Modern Radar: Advanced Techniques*, ser. Principles of Modern Radar. SciTech Pub., 2012. 5, 9, 12, 14

[11] M. A. Richards, J. Scheer, and W. A. Holm, *Principles of modern radar: basic principles.* SciTech Pub., 2010. 5, 14

[12] W. G. Carrara, R. S. Goodman, R. M. Majewski *et al.*, *Spotlight synthetic aperture radar: signal processing algorithms.* Artech House Boston, 1995. 5, 14, 50, 52, 53, 59

[13] C. V. Jakowatz, D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spotlight-mode synthetic aperture radar: a signal processing approach.* Springer, 1996. 5, 12, 14, 23, 37, 39, 47, 51, 52, 53, 59, 90

[14] J. C. Curlander and R. N. McDonough, *Synthetic aperture radar: systems and signal processing.* Wiley New York, 1991. 9, 53

[15] Y. Wu, J. Chen, and H. Zhang, "A real-time sar imaging system based on CPUGPU heterogeneous platform," in *Signal Processing (ICSP), 2012 IEEE 11th International Conference on*, vol. 1, Oct 2012, pp. 461–464. 12

[16] M. Pfitzner, F. Cholewa, P. Pirsch, and H. Blume, "FPGA based architecture for real-time SAR processing with integrated motion compensation," in *Synthetic Aperture Radar (APSAR), 2013 Asia-Pacific Conference on*, Sept 2013, pp. 521–524. 12

[17] E. Zaugg and D. Long, "Full motion compensation for LFM-CW synthetic aperture radar," in *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, Jul 2007, pp. 5198 –5201. 12

[18] A. Moreira and Y. Huang, "Airborne SAR processing of highly squinted data using a chirp scaling approach with integrated motion compensation," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 32, no. 5, pp. 1029–1040, 1994. 12

[19] D. Stevens, I. Cumming, and A. Gray, "Options for airborne interferometric SAR motion compensation," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 33, no. 2, pp. 409–420, 1995. 12

[20] C. Stringham and D. G. Long, "Improved processing of the CASIE SAR data," in *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International*, Jul. 2011. 12, 22, 102, 110

[21] L. Nguyen, M. Ressler, D. Wong, and M. Soumekh, "Enhancement of backprojection SAR imagery using digital spotlighting preprocessing," in *Radar Conference, 2004. Proceedings of the IEEE*, Apr 2004, pp. 53 – 58. 12, 15

[22] O. Frey, C. Magnard, M. Ruegg, and E. Meier, "Focusing of airborne synthetic aperture radar data from highly nonlinear flight tracks," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 47, no. 6, pp. 1844–1858, June 2009. 12

[23] A. Fasih and T. Hartley, "GPU-accelerated synthetic aperture radar backprojection in CUDA," in *Radar Conference, 2010 IEEE.* IEEE, 2010, pp. 1408–1413. 15, 22, 30

[24] T. Benson, D. Campbell, and D. Cook, "Gigapixel spotlight synthetic aperture radar backprojection using clusters of GPUs and CUDA," in *Radar Conference (RADAR), 2012 IEEE.* IEEE, 2012, pp. 0853–0858. 15, 22

[25] A. Capozzoli, C. Curcio, and A. Liseno, "Fast GPU-based interpolation for SAR backprojection," *Progress In Electromagnetics Research*, vol. 133, pp. 259–283, 2013. 15, 22, 23, 24, 30

[26] NVIDIA, “CUDA C programming guide v6.5,” *NVIDIA Corp*, Aug 2014. 17

[27] H. Wong, M. Papadopoulou, M. Sadooghi-Alvandi, and A. Moshovos, “Demystifying GPU microarchitecture through microbenchmarking,” in *Performance Analysis of Systems & Software (ISPASS), 2010 IEEE International Symposium on.* IEEE, 2010, pp. 235–246. 17

[28] A. Meta, P. Hoogeboom, and L. Ligthart, “Signal processing for FMCW SAR,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 45, no. 11, pp. 3519–3532, Nov 2007. 18

[29] O. Frey, M. Santoro, C. L. Werner, and U. Wegmuller, “DEM-based SAR pixel-area estimation for enhanced geocoding refinement and radiometric normalization,” *Geoscience and Remote Sensing Letters, IEEE*, vol. 10, no. 1, pp. 48–52, Jan 2013. 21

[30] K. Fourmont, “Non-equispaced fast Fourier transforms with applications to tomography,” *Journal of Fourier Analysis and Applications*, vol. 9, no. 5, pp. 431–450, 2003. 22, 23, 24, 78

[31] C. Sigg and M. Hadwiger, “Fast third-order texture filtering,” *GPU gems*, vol. 2, pp. 313–329, 2005. 24

[32] E. Zaugg, D. Long, M. Edwards, M. Fladeland, R. Kolyer, I. Crocker, J. Maslanik, U. Herzfeld, and B. Wallin, “Using the microASAR on the NASA SIERRA UAS in the characterization of Arctic sea ice experiment,” in *Radar Conference, 2010 IEEE*, May 2010, pp. 271 –276. 31, 100

[33] R. Baque, P. Dreuillet, O. Ruault du Plessis, H. Cantalloube, L. Ulander, G. Stenstrom, T. Jonsson, and A. Gustavsson, “LORAMbis a bistatic VHF/UHF SAR experiment for FOPEN,” in *Radar Conference, 2010 IEEE*, 2010. 33

[34] M. Soumekh, “Reconnaissance with ultra wideband UHF synthetic aperture radar,” *Signal Processing Magazine, IEEE*, vol. 12, no. 4, 1995. 33

[35] R. Thomas and J. Huang, “Ultra-wideband UHF microstrip array for GeoSAR application,” in *Antennas and Propagation Society International Symposium, 1998. IEEE*, vol. 4, 1998. 33

[36] P. Rosen, S. Hensley, I. Joughin, F. Li, S. Madsen, E. Rodriguez, and R. Goldstein, “Synthetic aperture radar interferometry,” *Proceedings of the IEEE*, vol. 88, no. 3, pp. 333–382, 2000. 37

[37] M. Richards, “A beginner’s guide to interferometric SAR concepts and signal processing,” *Aerospace and Electronic Systems Magazine, IEEE*, vol. 22, no. 9, pp. 5 –29, Sep 2007. 37

[38] F. Gatelli, A. Guamieri, F. Parizzi, P. Pasquali, C. Prati, and F. Rocca, “The wavenumber shift in SAR interferometry,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 32, no. 4, pp. 855 –865, Jul 1994. 37, 45, 47, 90

[39] M. Duersch and D. Long, "Analysis of multistatic pixel correlation in sar," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 53, no. 1, pp. 362–374, Jan 2015. 38

[40] F. Ulaby, R. Moore, and A. Fung, *Microwave remote sensing: Active and passive.* Addison-Wesley Publishing Co., 1982, vol. 2. 39

[41] J. Torgrimsson, P. Dammert, H. Hellsten, and L. Ulander, "Factorized geometrical autofocus for synthetic aperture radar processing," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 52, no. 10, pp. 6674–6687, Oct 2014. 48, 49, 54, 57

[42] H. Cantalloube and C. Nahum, "Multiscale local map-drift-driven multilateration SAR autofocus using fast polar format image synthesis," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 49, no. 10, pp. 3730–3736, Oct 2011. 49, 60

[43] P. M. Woodward, *Probability and information theory, with applications to radar.* Pergamon Press London, 1953, vol. 3. 50

[44] F. Wong and I. Cumming, "A combined SAR Doppler centroid estimation scheme based upon signal phase," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 34, no. 3, pp. 696–707, May 1996. 51

[45] H. Hellsten, P. Dammert, and A. Åhlander, "Autofocus in fast factorized backprojection for processing of SAR images when geometry parameters are unknown," in *Radar Conference, 2010 IEEE.* IEEE, 2010, pp. 603–608. 52, 54

[46] P. Samczynski and K. Kulpa, "Coherent mapdrift technique," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 48, no. 3, pp. 1505–1517, March 2010. 52

[47] P. Samczynski, "Superconvergent velocity estimator for an autofocus coherent mapdrift technique," *Geoscience and Remote Sensing Letters, IEEE*, vol. 9, no. 2, pp. 204–208, March 2012. 52

[48] D. Thompson, J. Bates, and D. Arnold, "Extending the phase gradient autofocus algorithm for low-altitude stripmap mode SAR," in *Radar Conference, 1999. The Record of the 1999 IEEE.* IEEE, 1999, pp. 36–40. 53

[49] J. Torgrimsson, "Evaluation of a new autofocus algorithm within the framework of fast factorized back-projection," Chalmers University of Technology, Tech. Rep., 2011. 53, 54

[50] L. Zhang, Z. Qiao, M. Xing, L. Yang, and Z. Bao, "A robust motion compensation approach for UAV SAR imagery," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 50, no. 8, pp. 3202–3218, Aug 2012. 53

[51] R. Morrison, M. N. Do, and D. Munson, "SAR image autofocus by sharpness optimization: A theoretical study," *Image Processing, IEEE Transactions on*, vol. 16, no. 9, pp. 2309 –2321, Sep 2007. 53

[52] T. J. Kragh, "Monotonic iterative algorithm for minimum-entropy autofocus," in *Adaptive Sensor Array Processing (ASAP) Workshop],(June 2006)*, 2006. 53

[53] J. Ash, "An autofocus method for backprojection imagery in synthetic aperture radar," *Geoscience and Remote Sensing Letters, IEEE*, vol. 9, no. 1, pp. 104 –108, Jan 2012. 54

[54] S. Hensley, "A combined methodology for SAR interferometric and stereometric error modeling," in *Radar Conference, 2009 IEEE*, May 2009, pp. 1–6. 59, 92

[55] SymPy Development Team, *SymPy: Python library for symbolic mathematics*, 2014. [Online]. Available: http://www.sympy.org 60

[56] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–, [Online; accessed 2014-11-07]. [Online]. Available: http://www.scipy.org/ 61

[57] A. Meta, P. Hoogeboom, and L. Ligthart, "Correction of the effects induced by the continuous motion in airborne FMCW SAR," in *Radar, 2006 IEEE Conference on*, 2006. 69

[58] C. Stringham and D. G. Long, "Processing for UWB LFM-CW SAR," in *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, Jul. 2014. 69, 70, 84, 85

[59] P. Prats-Iraola, R. Scheiber, M. Rodriguez-Cassola, J. Mittermayer, S. Wollstadt, F. De Zan, B. Brautigam, M. Schwerdt, A. Reigber, and A. Moreira, "On the processing of very high resolution spaceborne SAR data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, pp. 6003–6016, 2014. 69

[60] L. M. Ulander, H. Hellsten, and G. Stenstrom, "Synthetic-aperture radar processing using fast factorized back-projection," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 39, no. 3, pp. 760–776, 2003. 69, 71, 73, 79

[61] P. Frolind and L. M. H. Ulander, "Evaluation of angular interpolation kernels in fast back-projection sar processing," *Radar, Sonar and Navigation, IEE Proceedings*, vol. 153, no. 3, pp. 243–249, June 2006. 73

[62] V. Vu, T. Sjogren, and M. Pettersson, "A comparison between Fast Factorized Backprojection and frequency-domain algorithms in UWB lowfrequency SAR," in *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*, vol. 4, July 2008, pp. IV – 1284–IV – 1287. 73

[63] L. Rabiner, R. Schafer, and C. Rader, "The chirp z-transform algorithm," *Audio and Electroacoustics, IEEE Transactions on*, vol. 17, no. 2, pp. 86–92, Jun 1969. 78

[64] O. Ponce, P. Prats, M. Rodriguez-Cassola, R. Scheiber, and A. Reigber, "Processing of Circular SAR trajectories with Fast Factorized Back-Projection," in *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International*, July 2011, pp. 3692–3695. 80

[65] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965. 83

[66] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005. 83

[67] C. Stringham and D. G. Long, "GPU processing for UAS-based LFM-CW stripmap SAR," *Processing for UWB LFM-CW SAR*, December 2014. 84

[68] M. Duersch, "Backprojection for synthetic aperture radar," Ph.D. dissertation, Brigham Young University, 2013. 92

[69] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha, "Practical, real-time, full duplex wireless," in *Proceedings of the 17th annual international conference on Mobile computing and networking.* ACM, 2011, pp. 301–312. 92

[70] M. H. Hayes, *Statistical digital signal processing and modeling.* John Wiley & Sons, 1996. 101

[71] J. Maslanik, R. Crocker, K. Wegrzyn, C. Fowler, U. Herzfeld, D. Long, R. Kwok, M. Fladeland, and P. Bui, "Characterization of Fram Strait sea ice conditions using the NASA SIERRA unmanned aircraft system," in *AGU Fall Meeting Abstracts*, vol. 1, 2009, p. 06. 105

[72] M. Fladeland, R. Berthold, L. Monforton, R. Kolyer, B. Lobitz, and M. Sumich, "The NASA SIERRA UAV: A new unmanned aircraft for earth science investigations," in *AGU Fall Meeting Abstracts*, vol. 1, 2008, p. 0365. 105

[73] D. G. Long and C. Stringham. (2012, Mar.) Sample BYU microASAR data. [Online]. Available: http://www.mers.byu.edu/microASAR/CASIE_sample/ 107

[74] E. Zaugg and D. Long, "Theory and application of motion compensation for LFM-CW SAR," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 46, no. 10, pp. 2990–2998, 2008. 110

[75] B. Widrow and I. Kollar, *Quantization noise.* Cambridge University Press, 2008. 123

[76] C. Chi, D. Long, and F. Li, "Roundoff noise analysis for digital signal power processors using Welch's power spectrum estimation," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 35, no. 6, pp. 784 – 795, Jun 1987. 124

Appendix A

Improved Processing of the CASIE SAR Data

Working with real data commonly brings unforeseen challenges. This appendix presents methods used to improve the processing of the data collected as part of the Characterization of Arctic Sea Ice Experiment (CASIE). During 31 flight hours at altitudes ranging from 1000 to 1500 feet, it carried the C-band (5.4 GHz) microASAR, a small LFM-CW SAR, which collected data over most of this period. The microASAR design and its role in the CASIE09 mission have been well described in [6, 32]. Limitations in the motion measurements stored with the microASAR data during the CASIE09 mission originally precluded full motion compensation; however, motion data collected for other CASIE sensors can be employed to improve the SAR image focus and calibration.

The 160 MHz bandwidth microASAR is an LFM-CW SAR operated in a bistatic antenna configuration, where the transmit and receive lines are continuously employed. Due to the simultaneous transmit and receive, the received signal is contaminated with a strong signal (feed-through) of the transmit signal due to the proximity of the transmit antenna. To suppress the feed-through component, the received signal is "de-chirped" by mixing the received signal with the transmit signal, followed by a high pass filter. This is effective in suppressing feed-through to enable digital recording of SAR data. However, the residual feed-through component can be exploited to estimate the system time delay. In addition, the nadir return can be used to improve the height estimate and remove GPS altitude biases. Coupling these estimates with the time-aligned motion data enables better SAR image focusing and calibration. This appendix summarizes how these estimates are made and how they are used in processing the CASIE09 data to improve SAR image quality and accuracy.

A.1 CASIE focusing

The microASAR images were originally formed using the Range Doppler Algorithm (RDA), which provided good results considering the relative sparsity of the GPS position data that was included in the microASAR data stream. However, the particular flight characteristics of the UAS during CASIE09 unexpectedly caused degradation of the overall SAR image quality. Fortunately, an alternate source of higher-rate GPS and attitude data, recorded for other sensors on the CASIE09 platform, became available. To take advantage of the attitude and position data, an algorithm to temporally align it with recorded data is developed.

A.1.1 Motion measurement alignment

The original GPS data recorded with the SAR data was synchronized with a software interrupt which introduced some variability in the time alignment of raw SAR data and the GPS positions. After some experimentation, it was determined that sufficient time accuracy can be obtained by linear regression of the PRF and GPS time tags and then interpolating the GPS data to the start of each LFM chirp. In the initial regression, the average time-delay between the recorded SAR data and the GPS time tags is chosen by visually matching altitude changes in the GPS data to the nadir line in the SAR range compressed data. Fine tuning of the alignment is accomplished by minimizing the entropy of a small image section using a range of values near the visual alignment. The new motion data is then interpolated based on the interpolated GPS time of each LFM chirp. To further enhance the signal time-of-flight precision, thereby improving the image focusing, estimates of the system cable delay and the range to nadir are extracted from the SAR data.

A.1.2 Estimating the system delay

The system delay includes the time delay due to the cables in the transmit and receive lines, as well as the time delay caused by the RF components of the SAR system. The cable delay is usually estimated by the length of the cables. However, the RF component delay exhibits some variation with temperature. In this case the system delay can be estimated from the data collected during the flight using the residual feed-through.

The transmit feed-through component is dominated by the signal coupling from the transmit antenna into the receive antenna over the fixed, short distance separating the antennas. In the dechirped data, this small distance shows up as a low-frequency sinusoid. This signal component can be isolated using a low-pass filter. The sinusoid frequency is related to the measured antenna spacing and the system delay by the chirp rate. Hence, the system delay can be computed from the estimated frequency of the dechirped feed-through.

Conventional FFT-based methods for estimating the feed-through are limited by the range resolution of the SAR system (about 1m); however, the feed-through component is dominated by a single sinusoid so it can be accurately estimated with much finer resolution using the MUSIC algorithm [70].

Fig. A.1 illustrates the isolated feed-through component and the sinusoid computed from the estimated frequency. We note that the impulse response of the processing filter creates an artifact at the start of the signal. To prevent this from adversely affecting the estimate, the sinusoid fit is made using only the last 85% of the data. As illustrated in Fig. A.2, the system delay varies somewhat over a flight. Though this variation is small, estimating this delay and compensating for it in the processing improves the image focus because the slant range is more accurately determined.

A.1.3 Altitude improvement using nadir return

The quality of the SAR image is impacted by the accuracy of the platform height. Due to the limitations of the geoid used by GPS, the elevation of the aircraft in reference to the imaging surface is unknown, but because the CASIE09 flights were all conducted over open ocean and low-topography sea ice, the platform height can be estimated accurately

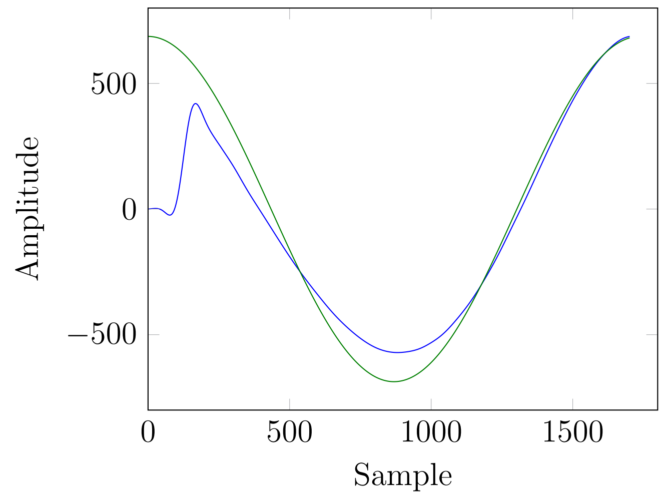


Figure A.1: Plot of the microASAR feed-through signal (line with kink at left) and the estimated sinusoid (smooth curve).

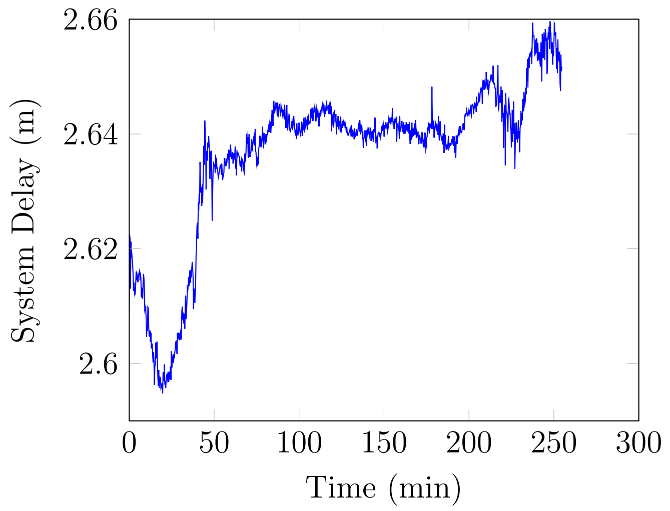
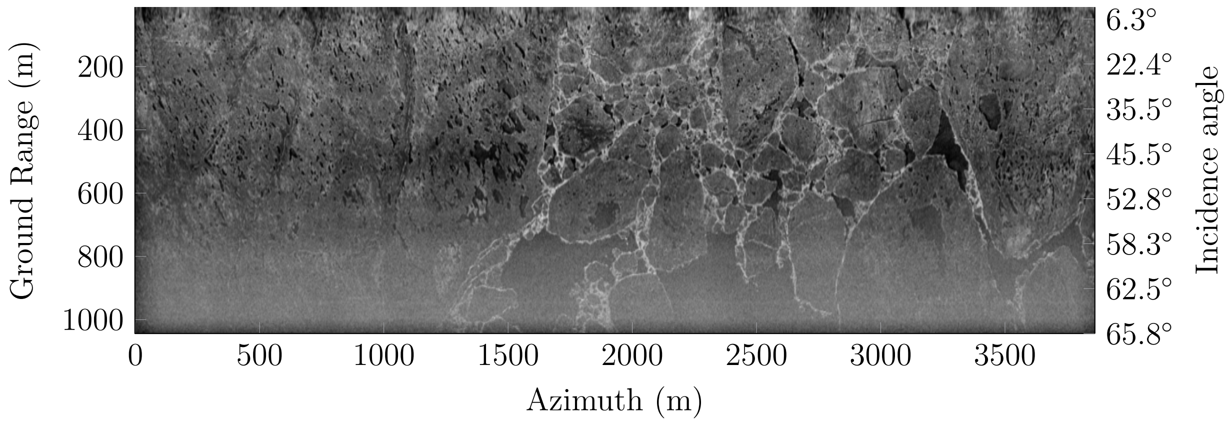


Figure A.2: Plot of the feed-through distance estimate over a 4.5 hour in-flight data collection. The time variation is attributed to the temperature dependence of the RF components.

from the SAR data [20]. Such height estimates can be used to correct for biases in the GPS altitude measurements. Linear regression is used to estimate the altitude bias error. Prior to aligning the new motion data, the nadir altitude bias is removed. To minimize noise in the altitude bias estimates, the bias estimates are first smoothed using a median filter.



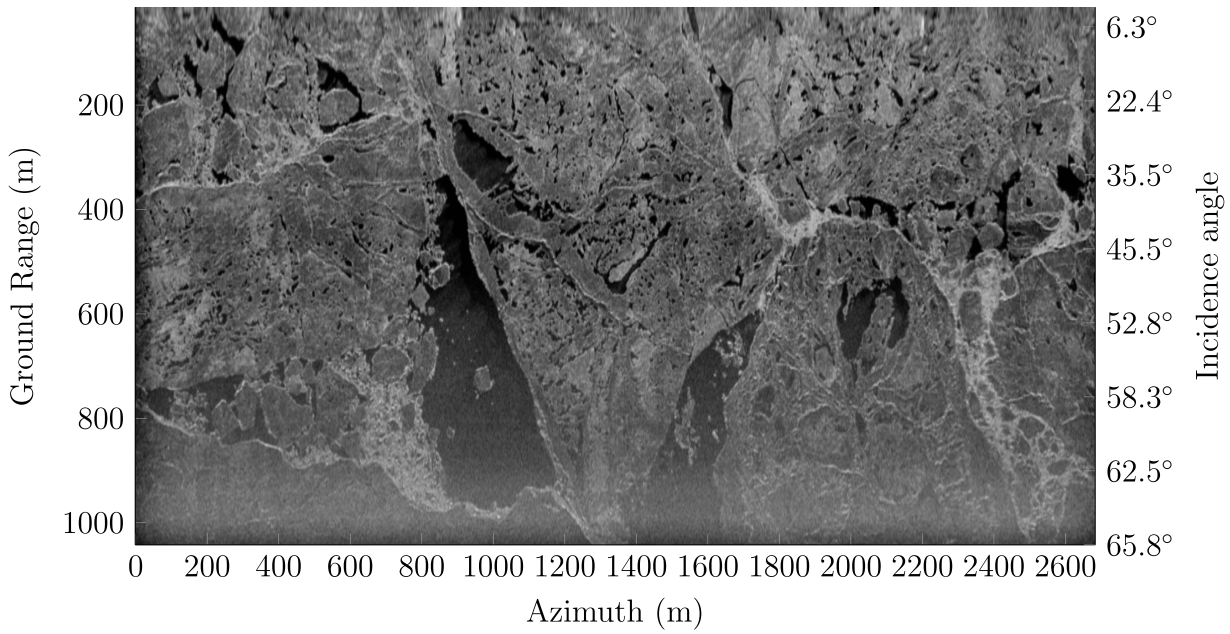


Figure A.3: Two examples of the improved imagery. The left y axis specifies the ground range in meters, and the right y axis is the incidence angle.

A.2 Results

Using the GPU based implementation described in Chapter 3, over 2000 km of SAR images were generated. Fig. A.3 shows a sample of the improved images. The flight track is at the top of the images, with the direction of travel from left to right. Both images were collected at an average altitude of approximately 450m. The new images have much better defined ice edges and even though there is considerable aircraft motion, the images are well focused. We note that the rolloff in the far range is due to lower surface backscatter due to the incidence angle increase and the reduction of antenna gain.

A.3 Summary

This appendix has presented the methodology used to improve the image processing of the CASIE-09 SAR data set. This process has required the application of several estimation techniques in order to align the GPS data with the SAR data, to account for a time variant system delay, and to correct for GPS altitude biases. The achieved improvement was made possible by the GPU backprojection implementation and the accurate alignment of the high-precision GPS data to the SAR data. The improvement thus made enhances the scientific research into the characterization of Arctic sea ice.

Appendix B

Sample BYU CASIE-09 MicroASAR Dataset

B.1 Introduction

This appendix describes a sample SAR data set collected by the BYU/Artemis microASAR system as flown as part of the Characterization of Arctic Sea Ice Experiment 2009 (CASIE-09). This data set is made available as a public service to the wider community to further interest in low-cost SAR applications. Sample Matlab scripts to process the data set into images using both the Range Doppler Algorithm (RDA) and the backprojection algorithm are also provided.

In the summer of 2009, a small, low power SAR was flown on a small, unmanned aircraft system (UAS) as part of the Characterization of Arctic Sea Ice Experiment 2009 (CASIE-09) [71] over the Arctic Ocean from Svalbard Island. The goal of the mission was to measure ice roughness in support of research monitoring ice thickness and ice age. The C-band SAR instrument, known as microASAR, collected 19.8 hours of high resolution SAR image data over 32.4 hours of UAS flight time in six UAS flights of varying length. The UAS was the NASA Sensor Integrated Environmental Remote Research Aircraft (SIERRA) [72]. As configured for operation on CASIE-09, the microASAR [2] collected data to on-board flash disk for later processing on the ground. The full dataset is considered proprietary. However, in the interest of furthering the application of SAR systems, a sample raw SAR data set from this mission is being made available for research. This document briefly describes the microASAR, the data set, and provides some examples of the processed images.

B.2 MicroASAR and the Sierra UAS

Synthetic aperture radar (SAR) can be a useful tool for sea ice observation, but SAR sensors have traditionally been large and expensive. The compact microASAR builds on the design of the BYU microSAR [1], but is a much more robust and flexible system [2]. The microASAR uses a linear frequency-modulated continuous-wave (LFM-CW) transmit signal generated by a direct digital synthesizer (DDS). The system is pseudo-monostatic, i.e. it uses separate transmit and receive antennas that that are placed closely together. This provides isolation between the transmit and receive channels and enables long transmit chirps which maximize the SNR while minimizing peak transmit power. The return signal is mixed down with a frequency-shifted copy of the transmit signal (this is known as analog dechirp), digitized, and processed with an all-digital final IF stage. Internal filtering minimizes transmit/receive signal feed-through. Raw data is stored to compact flash (CF) disk along with GPS timing and position data. Using 32 GB CF disks, over two hours of SAR data can be recorded. Table B.1 provides a summary of the microASAR hardware specifications. For the CASIE experiment, the microASAR transmit bandwidth is set to 170 MHz, yielding a maximum ideal range resolution of approximately 90 cm, although the effective resolution is reduced in processing. The transmit center frequency is 5.42876 GHz. After hardware presumming, the effective PRF is 307.292 Hz.

Table B.1: General microASAR Specifications

Physical Specifications	
Supply Power	< 35 W
Supply Voltage	+15 to +26 VDC
Dimensions	22.1×18.5×4.6 cm
Weight	2.5 kg
Radar Parameters	
Transmit Power	30 dBm
Modulation Type	LFM-CW
Operating Frequency Band	C-band
Transmit Center Frequency	5428.76 MHz
Signal Bandwidth	80-200 MHz (variable)
PRF	7-14 kHz (variable with optional hardware presumming)
Radar Operating Specifications	
Theoretical Resolution	0.75 m (@ 200 MHz BW)
Operating Velocity	10-150 m/s
Operating Altitude	500-3000 ft
Swath Width	300-2500 m (alt. dependent)
Collection Time (for 10GB)	30-60 min (PRF dependent)
Antennas (2 used)	
Type	2×8 Patch Array
Gain	15.5 dB (peak)
3dB Beamwidth	11° (Azimuth) × 42° (Elevation)
Size	$35 \times 12 \times 0.25$ cm

A key goal of CASIE-09 is to provide fine spatial resolution over difficult to access locations in the high Arctic. Satellites cannot provide the desired simultaneous combination of sensor types and resolution. Piloted aircraft typically fly too high and too fast to yield the fine-scale sampling rates and mapping patterns required by our project. UAS-based measurements can be made at low speed and low altitude in dangerous Arctic conditions without putting humans in manned aircraft at risk.

In CASIE-09 a microASAR was flown aboard the NASA SIERRA UAS. With a relatively large payload capacity, efficient mission planning software, and in-flight programmable autopilot, the SIERRA is well-suited for the long-duration missions used in the CASIE-09 experiment. Some of the other sensors on the UAS included two optical cameras, two pyrometers, an up/down-looking shortwave spectrometer, high-quality inertial measurement unit (IMU), and a laser profilometer system consisting of two down-pointing lasers, a medium-quality IMU, and differential-capable GPS receiver [2].

The UAS provided GPS position data via a serial data stream to the microSAR where it was included in the data storage. During CASIE-09 a video camera was mounted to view in the same direction as the microASAR, providing optical imagery coincident with the SAR swath (this data is not included with the sample data set). The UAS was flown at low altitude, typically between 600' and 1500'. While the low altitude limits the microSAR swath width, it also benefits the SAR measurement SNR. The low altitude operation results in an extremely wide range of incidence angles across the obseration swath.

The processed CAISE-09 SAR images show a variety of surface features from open ocean to dense pack ice. Features visible in the SAR imagery, as confirmed by the corresponding video, include ridges, rubble fields, brash ice, leads, and melt ponds. A particular microASAR image is shown in Fig. B.1.

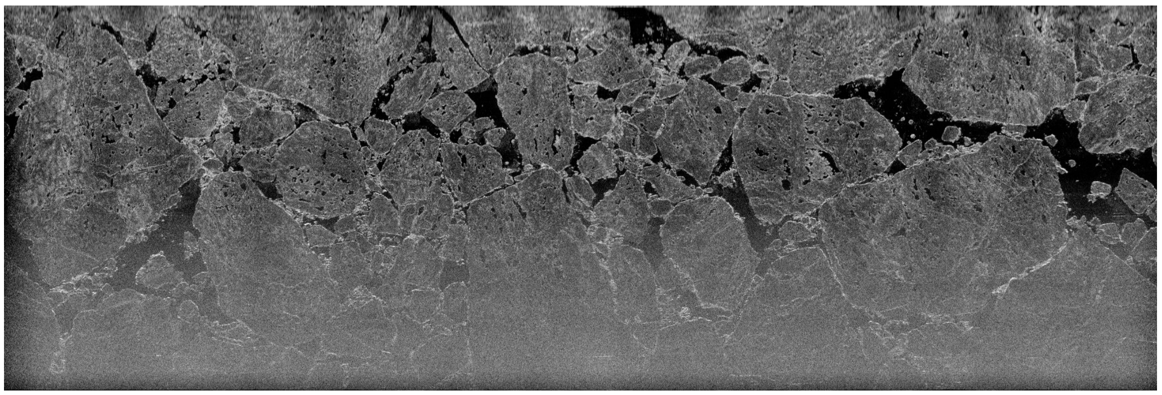


Figure B.1: Sample microASAR image from CASIE-09. Nadir is along the top. The UAS flew from left to right, viewing the surface to the right of the flight track as viewed from the aircraft (downward in the figure). The incidence angle varies from zero at the top to well over 72° at the bottom of the image. The image has been range compensated. Most of the surface is covered with ice floes with the dark areas corresponding to open water and melt ponds. The vertical banding is caused by antenna gain variations due to periodic rolling motion of the UAS. The image shows an area approximately 3.5 km long by about 1.2 km wide.

B.3 MicroASAR data

The microASAR can be programmed for operation in different modes and resolutions. For CASIE-09 raw data is divided into 1 min segments and separately processed into multi-looked image segments typically 3.5 km long in the along-track dimension by 1.2 km wide in the cross-track dimension. The sample data set made available for public distribution [73] consists of a 13 sec portion of one of these segments (segment 9, collected on 25 July 2009), which was arbitrarily selected. Figure B.2 shows a screen shot of a GoogleEarth window illustrating the full image segment shown as a georectified image. The sample data set corresponds to a portion near the center of the image. Due to the low altitude operation, the far-range is at an extreme incidence angle, and is thus not useful due to low SNR. (This area appears bright due to range compensation in this image.) The primary observation swath is defined to be from just off nadir to an incidence angle of about 72 deg. Many of the images shown later are clipped to this incidence angle range. Note that the surface is essentially flat and at sea-level for this data set. The average horizontal velocity during the sample data set collection is 30.1938 m/s at an average height of 346.5029 m.

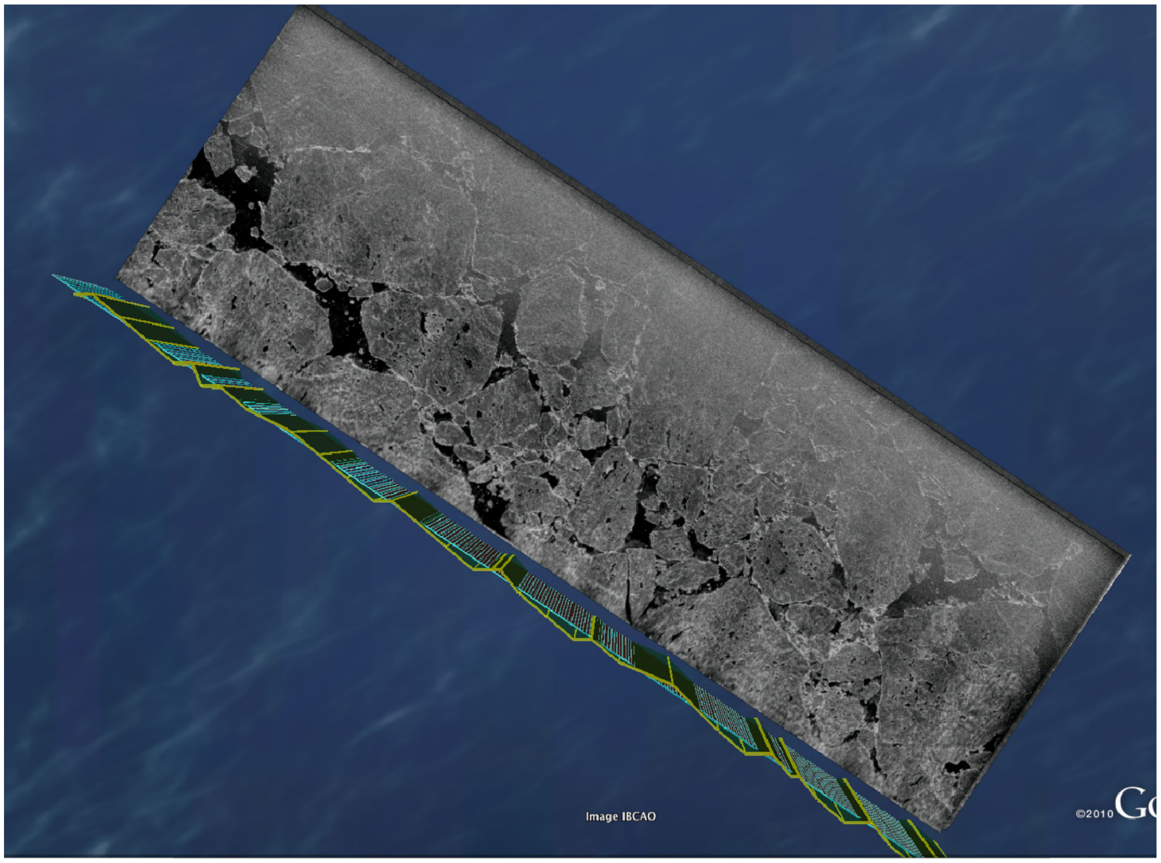


Figure B.2: Image segment 20090725_9 shown as a backprojected image in GoogleEarth. The UAS flight track is indicated with the blue path and nadir is along the bottom. MicroASAR-recorded GPS positions are in yellow. The UAS flew from lower-right to upper-left, viewing the surface to the right. The incidence angle varies from zero at nadir track near GPS track to over 72° at the top of the SAR image.

For distribution the sample CASIE-09 microASAR data is provided in a single Matlab "MAT-file", "flight9_9_sample.mat", containing arrays for the sampled raw dechirped data (named "dat") and the geometry data (named "geom"). Each of the 3885 columns of the arrays "dat" and "geom" represent the dechirped SAR data and the aircraft location interpolated for each pulse, respectively. The four rows of "geom" are the SAR pulse number, latitude in degrees, longitude in degrees, and altitude in m.

The following describes microASAR details and settings as used for the CASIE-09 experiment for the sample data set and how the data can be processed into images. Additional detail can be found in the sample Matlab processing scripts.

After digital IF processing in the microASAR hardware, the effective sample rate of the dechirped data is 24.485 MHz. To maximize the SNR for CASIE, the microASAR was set to use a high rate repeating chirp (i.e., a high PRF) with a LFM frequency ramp rate of 1.5972563681e12 Hz/s. To reduce the data rate for the CASIE-09 experiment onboard storage, the microASAR employed hardware presumming, which consists of coherently averaging several sequential pulses together, resulting in an effective PRF of 307.292 Hz (see [3] for additional detail).

Recall that in an LFM-CW radar range compression can be accomplished by computing a zero-padded FFT of each LFM chirp. (Technically, the microASAR is not strictly CW, as there are small gaps between the individual transmitted chirps, though this does not alter the processing.) In range compressing the CASIE microASAR data note that the start portion (the first 30 samples) of each LFM chirp (for convenience a single LFM chirp is referred to as a 'pulse') should be zeroed to avoid artifacts induced by transient RF signals caused by the switching of the transmitter. The dechirped pulse length is truncated to 1702 dechirped samples prior to onboard storage. Note that zeroing and truncation have the side effect of reducing the theoretical range resolution of the range compressed data to 1.35 m/sample. After zero-padding each chirp to 4096 values and computing the FFT, the redundant negative frequency half of the range compressed data can be discarded.

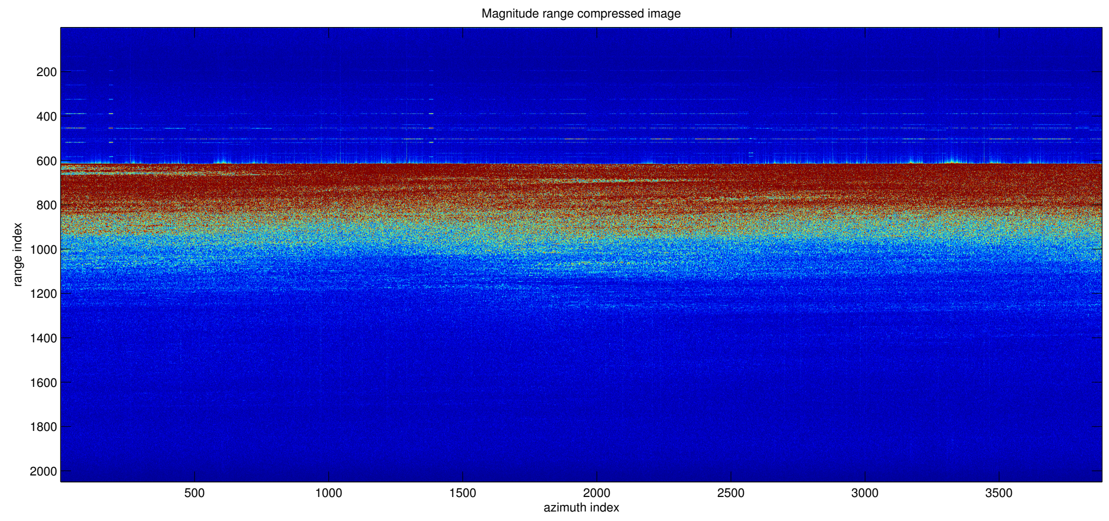


Figure B.3: Magnitude images of the range-compressed sample data set computed by zero padding and using a 4096 point FFT. Nadir is at the top of the red area and the platform moves from left to right.

Figure B.3 shows the range-compressed image created from the sample dataset. Note that zero padding changes the meters per pixel in the range-compressed image, though the effective resolution is unchanged. In this image the nadir return occurs at approximately range bin 800 in this example. Range bins prior to the nadir return (which are in the blue area at the top of the image) contain no useful information. The primary swath shows as red and light blue. The internal cabling delay corresponds to approximately 1.2 m. For this data collection, the platform height offset described in [20] has already been applied to the GPS altitude data recorded in the geom array.

Each pulse has a unique pulse counter number. Due to limitations of the data storage system there are occasional missing pulses. These can be identified by gaps in the sequential pulse counter. For range-Doppler processing such gaps need to be filled (e.g., by interpolation) to avoid introducing azimuth artifacts. See Matlab code for details.

B.4 SAR image formation

To encourage further research and development of SAR imaging techniques, we provide a Matlab script, "casie_sample_code.m" that creates focused SAR images of the sample data set using RDA and backprojection. Note that azimuth compression of an LFM-CW SAR includes an additional phase term compared to a conventional pulsed SAR system, see [74] and Chapter 3.

B.4.1 Range Doppler processing

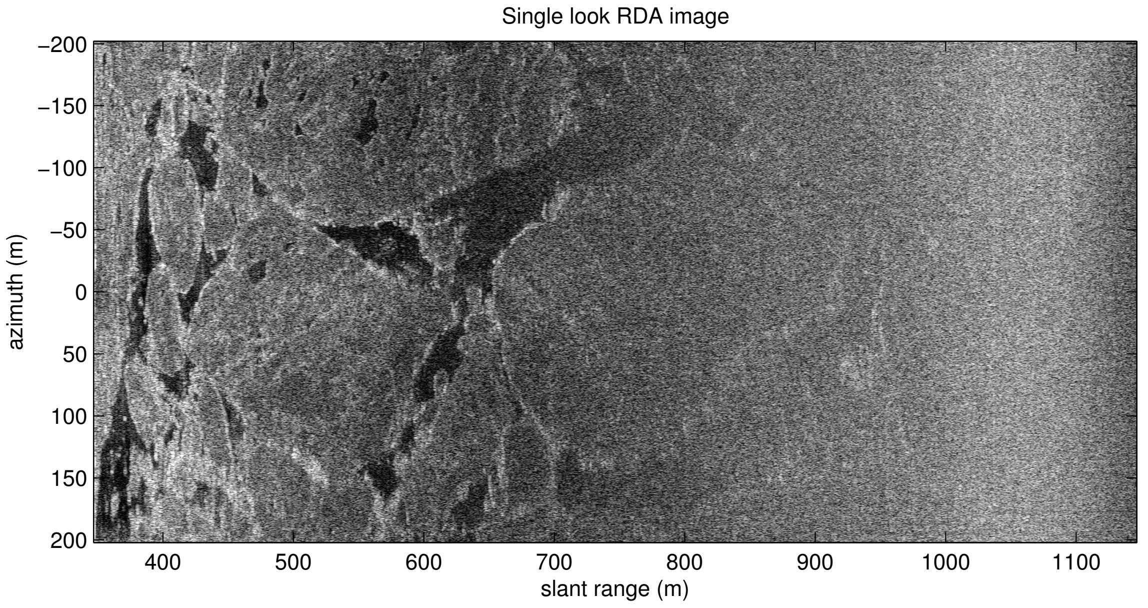
In RDA, a range-dependent azimuth filter is required to minimize aliasing and the effects of the platform's rapid attitude variation. An example of a single-look azimuth compressed image created using RDA without range migration or motion compensation is shown in Fig. B.4a. To approximately compensate for range roll-off in this image, the image pixel values are scaled by r^3. In addition the images are displayed in Decibels with an arbitrarily selected grayscale range. No compensation for the antenna gain pattern is included. Note that the backscatter response from sea ice is incidence angle dependent, but this not compensated for. In this and following images, the image is clipped to nadir at the left and an incidence angle of 72° at the right.

To optimize the focus when creating the RDA image using the low altitude CASIE data we find it advantageous to use a hyperbolic azimuth chirp rather than the traditional RDA parabolic azimuth chirp. A conventional parabolic azimuth chirp $Az(m)$ used for RDA azimuth compression at range r_a can be expressed as

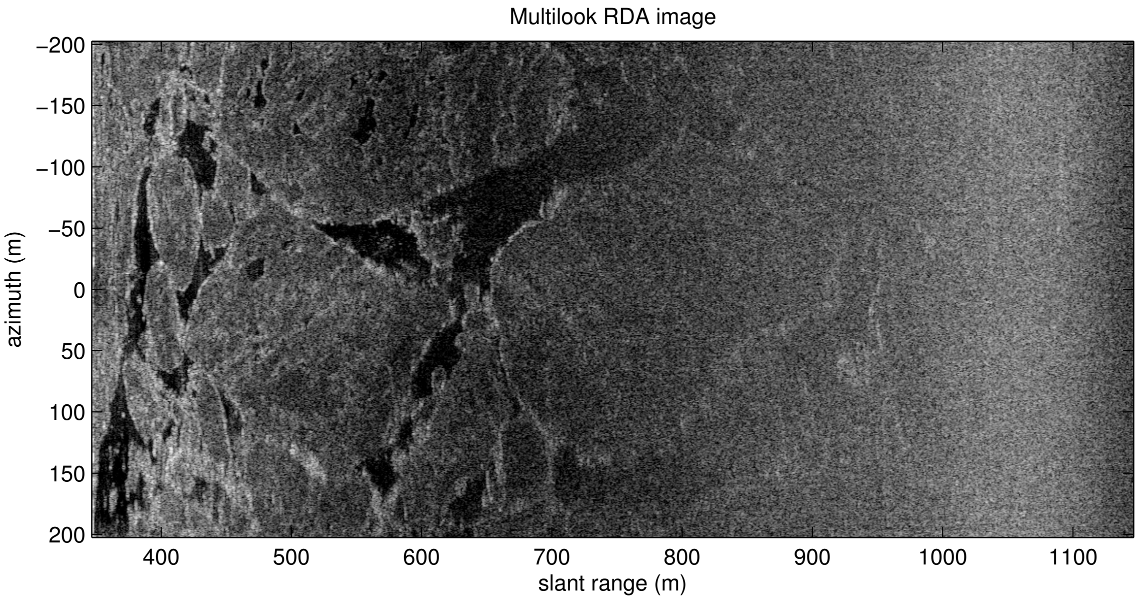
$$Az(m) = \exp\left\{-j\left(\frac{f_c m^2 v^2}{c r_a \mathrm{PRF}^2} - \frac{2 f_c v^2 m}{c^2 \mathrm{PRF}}\right)\right\}, \tag{B.1}$$

where $m = [-N_p/2..N_p/2]$ is the pulse index where N_p is the number of pulses, f_c is the carrier frequency, c is the speed of light, PRF is the pulse repetition frequency, and v is the along-track velocity. For a hyperbolic version, we define

$$\tau(m) = 2\sqrt{r_a^2 + (mv/PRF)^2}, \tag{B.2}$$



(a)



(b)

Figure B.4: Magnitude of the single look (a) and multi-looked (b) RDA images. Note that the single look image is compressed vertically to display it here. This has the effect of making the displayed image appear multi-looked and reducing the appearane of speckle.

where the azimuth displacement is mv/PRF and

$$Az(m) = \exp\left\{-j\left(2\pi f_c \tau(m) + K_r \tau^2(m)\right)\right\}, \tag{B.3}$$

where K_r is the LFM frequency ramp rate. Of course, an appropriate azimuth window should be applied.

Since the single look azimuth resolution (1/2 the antenna length) is much smaller than the range resolution, spatially square pixels can be generated by generating an image with high azimuth resolution and incoherently averaging (multi-looking) pixels in the azimuth direction to reduce the azimuth resolution and speckle noise. A multi-looked (in azimuth) image generated from the image in Fig. B.4a shown in Fig. B.4b. There is a reduction in the speckle level and the increased contrast in the multi-looked image.

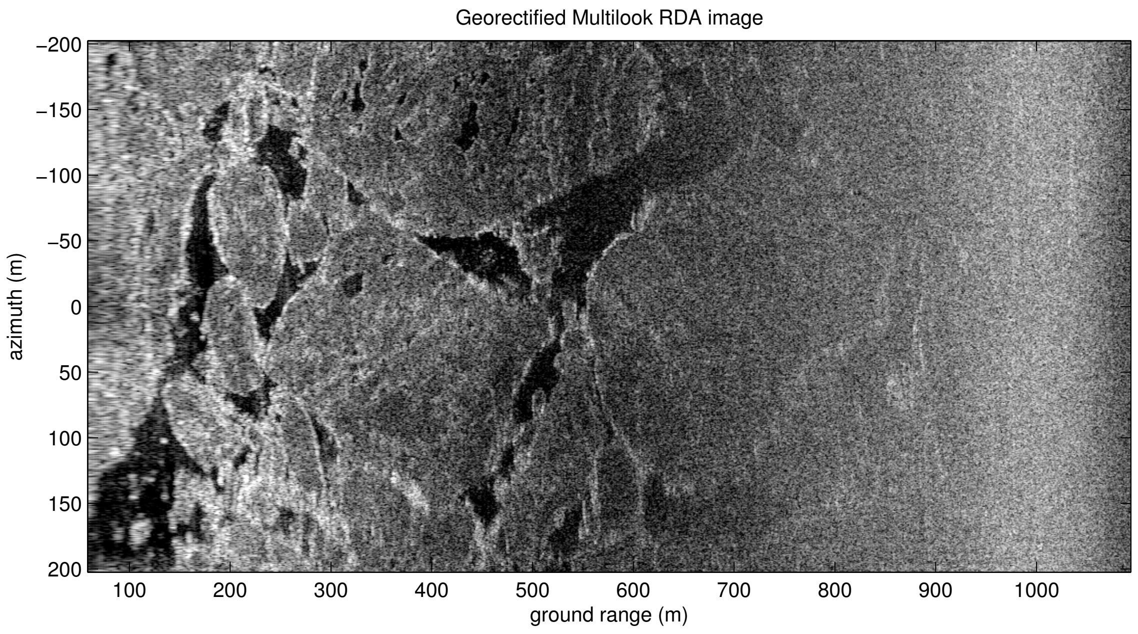


Figure B.5: Georectified RDA magnitude image. (Normally the left few image lines would be clipped from the image).

Note that Figs B.4a and B.4b are in slant range. Using the multi-looked azimuth compressed image and the height information an example of a georectified image is shown in Fig. B.5. This is in cross-track distance, or ground range, i.e. it represents a map view from above. For this image a simple nearest-neighbor range interpolation is used to convert from slant range to ground range. The low altitude emphasizes the variations in range resolution of the ground range image. Note the expected effective resolution loss in range resolution near nadir.

B.4.2 Backprojection processing

Backprojection is a deceptively simple approach for generating images from raw synthetic aperture radar (SAR) data. SAR backprojection implements the azimuth matched filter for each pixel, and accounts for both range cell migration and the motion of the aircraft.

The time domain backprojection algorithm is an exact algorithm for creating images from SAR data. The algorithm computes the radar cross-section over a grid of pixels in ground range on the surface. For a pixel located (x_0, y_0, z_0), the SAR backprojection algorithm for an LFM-CW radar can be approximately expressed as

$$A(x_0, y_0) = \sum_n S(n, d[n]) P(d[n]) \exp\{-j4\pi(d[n]/\lambda - d^2[n]K_r/c^2)\}, \tag{B.4}$$

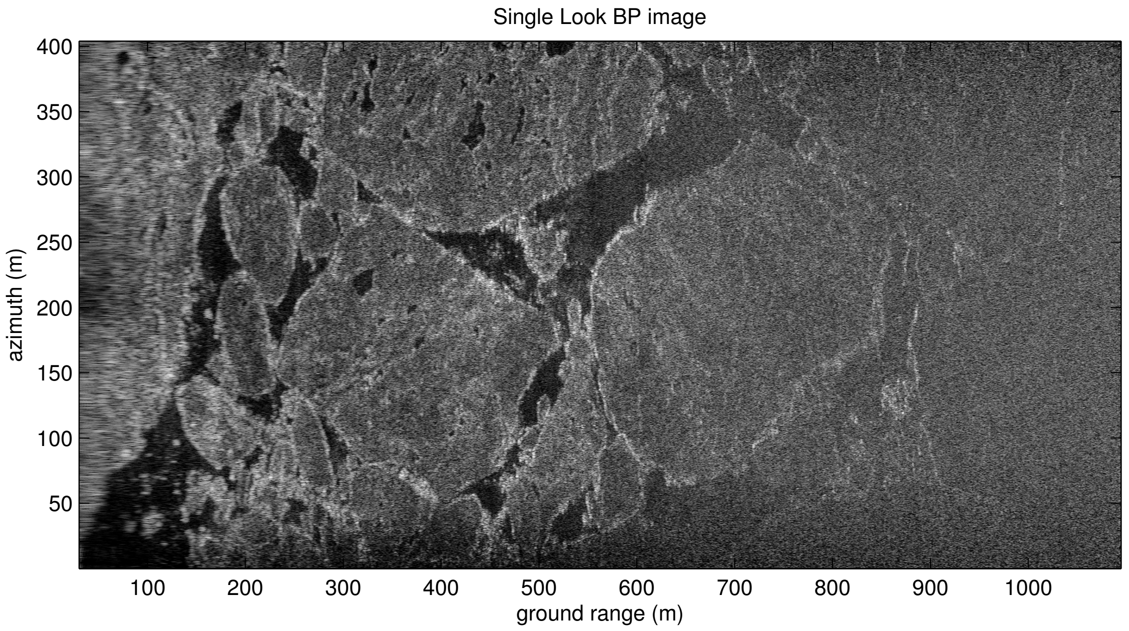
where $A(x_0, y_0)$ is the complex pixel value (the complex SAR image), λ is the wavelength of the transmit frequency at the center of the SAR band, $d[n]$ is the distance between (x_0, y_0, z_0) and the antenna phase center of the SAR antenna $(x[n], y[n], x[n])$ for pulse number n. The distance $d[n]$ is defined as

$$d[n] \quad = \quad \sqrt{(x[n] - x_0)^2 + (y[n] - y_0)^2 + (z[n] - z_0)^2}. \tag{B.5}$$

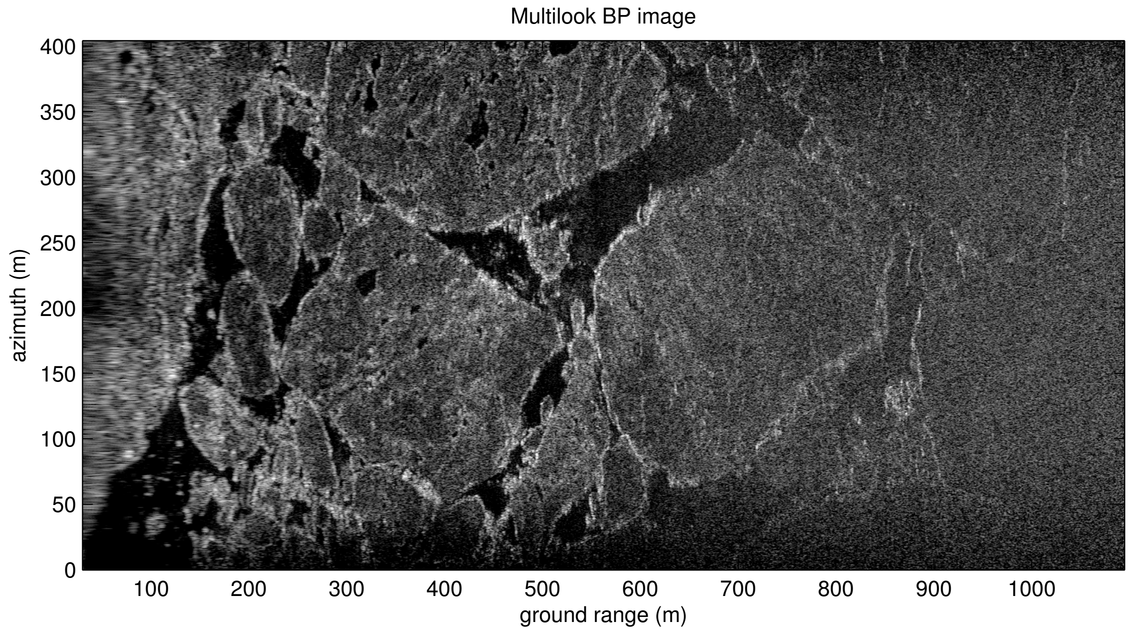
A range-dependent azimuth window $S(n, d[n])$, which is pulse index (n) dependent, is included in the processing to enable sub-aperture processing and/or reduce azimuth sidelobes. In Eq. (B.4) $P(d[n])$ is the range-compressed SAR data interpolated to slant range $d[n]$. The summation is over the pulses which for which the azimuth and elevation gain of the antenna are significant. An image is created by varying the horizontal positions x_0 and y_0 over a grid. The height z_0 of the surface at (x_0, y_0) is typically computed from a digital elevation map (DEM). Since the CASIE-09 images are at the surface $z_0 = 0$.

The challenge of time-domain backprojection is that is computationally intensive; however it can be parallelized to achieve real-time performance in many cases as described in Chapter 3. While the implementation of backprojection included in the Matlab code sample uses Matlab GPU constructs in order to speed up the processing if a GPU is available, the implementation of the backprojection algorithm provided in the sample code is written for simplicity and clarity rather than computational efficiency. The implementation includes the phase term required to compensate for motion of the platform during the transmit pulse for an LFM-CW SAR as described in Chapter 3.

Single and multi-look images generated using backprojection are shown in Fig. B.6. Note that the backprojected images are inherently in ground range, i.e. they are georectified. We note an increase in visible detail in comparison to the images generated with RDA due to the fact that backprojection accounts for range migration and platform motion.



(a)



(b)

Figure B.6: Magnitude of the single look (a) and multi-looked (b) backprojection complex images. (Note that the single look image is compressed vertically to display it here. This has the effect of making the displayed image appear multi-looked and reducing the appearane of speckle.)

B.5 Example Matlab processing code

The sample data and reference Matlab implementations of the RDA and backprojection algorithm are available at `http://www.mers.byu.edu/microASAR/CASIE_sample/`, and are printed here.

```matlab
% (c) Copyright David Long, Brigham Young University, 2011.
% All rights reserved.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This Matlab script illustrates how to process the BYU CASIE-09
% sample data set using the range-Doppler algorithm (RDA) and
% time-domain backprojection (BP) algorithm
%
% Written by Craig Stringham and David G Long, BYU 2011.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% close all, clear all;
PLOT=1;          % flag to enable plotting of images
RUNBP = 1;       % flag to enable computation of backprojection
updateTime = 5; % time between backprojection processing status updates
aresScale = 1;   % az resolution scale for BP. values>1 reduce
                 % resolution and computation time
USEGPU = 1;      % flag to enable use of GPU on capable platforms, zero
                 % value disables use of GPU
matver = sscanf(version,'%d.%d.%d');
if matver(1) <= 7 && matver(2) < 12
    warning('This version of matlab does not support use of GPU')
    USEGPU = 0;
end

if USEGPU
    cgpuArray = @(x) single(gpuArray(x));
    cgather = @(x) gather(x);
else
    cgpuArray = @(x) x;
    cgather = @(x) x;
end

%% Set up microASAR SAR and processing parameters
c0 = 299702547;          % speed of light in the atmosphere (m/s)
actual_samples=1702;     % number of samples/chirp
PRF = 307.292;           % effective pulse repetition frequency (Hz)
azBW=0.1920;             % 3dB antenna azimuth beamwidth (rad) (3dB to 3dB)
cabledelay = -1.2;       % system and cable delay (m)
f_adc = 24485000;        % adc sample rate (Hz)
bw = 170000000;          % transmit bandwidth (Hz)
fc = 5.42876e9;          % carrier frequency (Hz)
kr = 1.597249139246997e12;   % chirp rate
fftsize = 4096*4;        % size of fft to use for backprojection
                         % range compression
N = ceil(fftsize/2);
```

```
delrad = c0*f_adc/(4*kr*pi); % meters per radian
delsamp = delrad*2*pi/actual_samples; % meters per real sample
delrsamp = delsamp*(actual_samples/fftsize); % meters per
                                             % interpolated sample
max_range = c0*f_adc/(4*kr); % maximum range without aliasing
lambda=c0/fc;          % carrier wave length
ares = lambda/(2*azBW)*aresScale; % azimuth resolution
rres = c0/(2*bw);      % slant range resolution

%% load the BYU CASIE-09 sample data
load('flight9_9_sample.mat')
numpulses = size(dat,2);

%% range compress the data using a range window and zero padding
% Note that the first 30 samples are set to zero.  A Taylor window
% with NBar=4, SSL=-226 minimizes range sidelobes, but other windows
% can be used.
rc = fft(dat.*([zeros(30,1); taylorwin(actual_samples-30,4,-26)] ...
               * ones(1,numpulses)),fftsize,1);
% discard duplicated half of the spectrum
rc = [rc(1:N,:); zeros(1,size(rc,2))];

% plot the range compressed data
if PLOT
    figure(1), imagesc(abs(rc),[0 5e4]),
    title('Magnitude range compressed image')
    ylabel('range index');
    xlabel('azimuth index');
end

%% plot visuals of the flight geometry
t = (geom(1,:)-geom(1,1))/PRF; % time axis
if PLOT
    figure(2),
    th(1) = subplot(3,1,1);
    subplot(3,1,1), plot(t,geom(2,:))
    title('latitude')
    th(2) = subplot(3,1,2);
    subplot(3,1,2), plot(t,geom(3,:))
    title('longitude')
    th(3) = subplot(3,1,3);
    subplot(3,1,3), plot(t,geom(4,:))
    linkaxes(th,'x')
    title('altitude')
end

%% project lat lon onto a local tangent plane (northing easting surface)
RefLat=mean(geom(2,:));         % select a reference latitude
Conv=1852.23 * 60.0;            % latitude conversion factor
Lconv=Conv*cos(RefLat*pi/180); % longitude conversion factor
northing = geom(2,:)*Conv;
easting  = geom(3,:)*Lconv;
z = geom(4,:)-cabledelay ;      % correct height by cabledelay;

```

```
if PLOT
    figure(1),
    hold on, plot(z/delrsamp), hold off
end

%% rotate and move the gps data to a local reference
rotAngle = ...
   -atan2((easting(end)-easting(1)),(northing(end)-northing(1)))+pi/2;
cang = cos(rotAngle);
sang = sin(rotAngle);
l_northing = (northing-northing(1));
l_easting = easting-easting(1);
r = cang*l_northing-sang*l_easting;
a = sang*l_northing+cang*l_easting;
%%
if PLOT
    figure(3)
    plot(a(1),r(1),'gs',r(end),a(end),'rs',l_northing(1),l_easting(1), ...
         'gx',l_northing(end),l_easting(end),'rx'),
    figure(4)
    subplot(3,1,1), plot(a-a(end).*(t./t(end))), ...
        title('azimuth deviation'), ylabel('(m)' )
    subplot(3,1,2), plot(r), title('ground range deviation'), ylabel('(m)')
    subplot(3,1,3), plot(z-mean(z)), title('altitude deviation'), ...
        ylabel('(m)'), xlabel('time index')
end

%% Backprojection Processing
if RUNBP
    disp('Beginning Backprojection processing')
    % set up grid for backprojection
    ground_range = sqrt(max_range.^2-mean(z).^2);
    y = cgpuArray(([ares:ares:a(end)]-ares/2).');
    x = cgpuArray([rres+30:rres:ground_range]-rres/2);
    z = cgpuArray(z);
    z2 = z.^2;
    % define image grid
    [X,Y] = meshgrid(x,y);
    % pre-compute constants in exponent
    betapic=4*pi*kr/c0.^2;
    lambda4pi=4*pi/lambda;
    % define output image and phase arrays
    img = cgpuArray(single(complex(zeros(size(X)),0)));
    pphase = cgpuArray(single(complex(zeros(size(X)),0)));
    % perform backprojection
    % use clock tick to measure CPU time
    inittic = tic;
    updr = inittic;
    if PLOT
        figure(10)
    end

    %% backprojection loop
    for k = 1:numpulses
```

```
        ty = Y−a(k);
        tx = X+r(k);
        tx2 = tx.^2;
        D2 = (ty.^2+tx2+z2(k));
        D = sqrt(D2); % distance from antenna to each pixel
        % get the azimuth angle estimate
        mz = sqrt(tx2+z(k).^2).*sign(tx); % can approximate by using ...
           mean(z) to lower computation
        az = atan2(ty,mz); % azimuth angle to each pixel
        ids = round(D./delrsamp);
        ids(ids>N) = N+1;
        pphase = exp(−1i*(betapic*D2−D*lambda4pi)); % calculate the ...
           expected phase
        % multiply and accumulate image value
        img = img+pphase.*reshape(rc(ids,k),size(ids,1),size(ids,2)) ...
              .* (abs(az)<(azBW/2)).*exp(−az.^2*30);
        % plot an updated image
        tdr = toc(updr);
        if tdr > updateTime
            fprintf(1,['%d of %d complete, %fs elapsed, %fs ' ...
                       'estimated\n'], k, numpulses, toc(inittic), ...
                   toc(inittic) * (numpulses−k)/k )
            updr = tic;
            %
            if PLOT
                dimg = db(cgather(img));
                dimg(¬isfinite(dimg)) = 0;
                rangescale = mean(dimg(1:find(cgather(y)−a(k)>0,1, ...
                                             'first' ), :), 1 );
                rangescale = max(rangescale)./rangescale;
                rangescale(rangescale > 500) = 500;

                dimg = dimg.*(ones(length(y),1)*rangescale);
                imgMin = prctile(dimg(dimg>0),10);
                imgMax = prctile(dimg(dimg>0),99.99);
                set(0,'CurrentFigure',10);
                imagesc(x,y,dimg,[imgMin imgMax]), %colorbar
                set(gca,'YDir','normal'), colormap('gray')
                drawnow
                title('Single Look BP image')
            end
        end
    end

%%
if PLOT
    figure(10)
    imagesc(x,y,dimg,[imgMin imgMax]), colormap('gray')
    set(gca,'YDir','normal')
    title('Single Look BP image')
    xlabel('ground range (m)'); ylabel('azimuth (m)');
drawnow
end
%%
```

```
    img = cgather(img); % bring the gpuArray back to the cpu memory

    %% create multi-look backprojection image
    NazLook=4;
    mimg=zeros([floor(size(img,1)/NazLook) size(img,2)]);
    for k=1:NazLook:size(img,1)-NazLook
        kk=(k-1)/NazLook+1;
        mimg(kk,:)=sum(abs(img(k:k+NazLook-1,:)).^2,1)/NazLook;
    end

    %% plot a multi look image

    if PLOT
        dmimg = db(cgather(mimg));
        dmimg(¬isfinite(dmimg)) = 0;
        rangescale = mean(dmimg,1);
        rangescale = max(rangescale)./rangescale;
        rangescale(rangescale > 500) = 500;
        dmimg = dmimg.*(ones(size(dmimg,1),1)*rangescale);
        mimgMin = prctile(dmimg(dmimg>0),10);
        mimgMax = prctile(dmimg(dmimg>0),99.99);
        figure(11)
        imagesc(x,y,dmimg,[mimgMin mimgMax]), colormap('gray')
        set(gca,'YDir','normal')
        title('Multilook BP image')
        xlabel('ground range (m)'); ylabel('azimuth (m)');
    drawnow
    end
    disp('Backprojection processing completed')
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% RDA Processing
disp('Beginning RDA processing')
rdafftsize = 4096;

%% RDA processing needs evenly spaced pulses data.  Since there are gaps
% we first need to fill in the missing pulses. What follows is a simple
% but effective interpolation method for the CASIE-09 SAR data.
pulsen = geom(1,:)-geom(1,1)+1;
dat1 = zeros(actual_samples,pulsen(end));
dat1(:,pulsen) = dat;
dpulsen=diff(pulsen)-1;
skips = pulsen(dpulsen ≠0);
dpulsen = dpulsen(dpulsen ≠0);
for k = 1:length(skips)
    dat1(:,skips(k)+1:skips(k)+dpulsen(k)) = dat1(:, skips(k) - ...
                                            dpulsen(k) + 1:skips(k));
end
Naz = size(dat1,2);

%% interpolate the rest of the data
t1 =(1:pulsen(end))./PRF;
a1 = interp1(t,cgather(a),t1,'spline');
```

```
r1 = interp1(t,cgather(r),t1,'spline');
z1 = interp1(t,cgather(z),t1,'spline')+cabledelay;

%% range compress the data
% Note that the first 30 samples are set to zero.  A Taylor window
% with NBar=4, SSL=-226 minimizes range sidelobes, but other windows
% can be used.
rc1 = fft(dat1.*([zeros(30,1); taylorwin(actual_samples-30,4,-26)] ...
                    * ones(1,Naz)),rdafftsize,1);
% discard duplicated half of FFT output
rc1 = rc1(1:rdafftsize/2,:);

% compute the slant range per sample
rdadelrsamp = delsamp*(actual_samples/rdafftsize);

%% find the mean linear velocity of the aircraft
vel = a1(end)./t1(end);

ran = (0:rdafftsize/2-1)*rdadelrsamp+cabledelay;
azm = ([1:Naz]-Naz/2-1)*vel/PRF;
% meshgrid az and range vars
[AZ RA] = meshgrid(azm,ran);

disp('calculating azimuth chirp')
% conventional parabolic azimuth chirp
% M = exp(-j*(fc*2/c0*(AZ.^2./(2*RA))-2*fc*vel*AZ./(c0.^2)));
% hyperbolic azimuth chirp
tau_r=2*sqrt(RA.^2+AZ.^2)/c0;
M=exp(-1i*(2*pi*fc*tau_r+kr*tau_r.^2));

% azimuth window (rect) based on antenna beamwidth
M(abs(AZ) > 0.5*RA*azBW) = 0.;
% simple tapered azimuth window
azwin=-600.0;
M=M.*exp(azwin*(AZ./RA).^2);

% fft shift to center frequency at origin
M=fftshift(M,2);

% convert azimuth chirp to frequency domain
M=fft(M,[],2);

% Take range compressed data to the range Doppler domain
A4=fft(rc1,[],2);

% Multiply by fft of azimuth chirp to do matched filtering
A4 = A4.*conj(M);

% Finish the process using ifft
A4 = ifft(A4,[],2);

% Apply an approximate range compensation for image display
rcomp=(ran').^3;
for k=1:size(A4,2)
```

```
    A4(:,k)=A4(:,k).*rcomp;
end

%% generate a georectified RDA image
grange = sqrt(([1:rres:sqrt((max_range-2)^2-mean(z1).^2)]).^2 + ...
    mean(z1).^2);
grange = grange(find(grange>mean(z1)+5,1,'first'):end);
gA4 = interp1([1:size(A4,1)]*rdadelrsamp,A4,grange);

% generate multilook image
NazLook=8;
C=zeros([size(A4,1) floor(size(A4,2)/NazLook)]);
gC=zeros([size(gA4,1) floor(size(gA4,2)/NazLook)]);
for k=1:NazLook:size(A4,2)-NazLook
    kk=(k-1)/NazLook+1;
    C(:,kk)=sum(abs(A4(:,k:k+NazLook-1)).^2,2)/NazLook;
    gC(:,kk)=sum(abs(gA4(:,k:k+NazLook-1)).^2,2)/NazLook;
end

imrange = ceil((mean(z1)+1)/rdadelrsamp):floor((max_range-2)/rdadelrsamp);
%% plot the RDA images
if PLOT
    % display single look image
    myfigure(102)
    dA4 = db(rot90(A4(imrange,:)));
    r1low = prctile(dA4(:),5);
    r1high = prctile(dA4(:),99.9);
    azdis = [-vel*t(end)/2 vel*t(end)/2];
    imagesc(imrange*rdadelrsamp,azdis,dA4,[r1low r1high]);
    xlabel('slant range (m)'); ylabel('azimuth (m)');
    title(sprintf('Single look RDA image'))
    colormap('gray'), %colorbar
    % display multi look image
    myfigure(104)
    dC = db(rot90(C(imrange,:)));
    rmlow = prctile(dC(:),3);
    rmhigh = prctile(dC(:),99.99);
    imagesc(imrange*rdadelrsamp,azdis, dC,[rmlow rmhigh]);
    %colorbar
    xlabel('slant range (m)'); ylabel('azimuth (m)');
    colormap('gray')
    title(sprintf('Multilook RDA image'))
    %% plot georectified multilook image
    myfigure(105)
    dgC = db(rot90(gC));
    rgmlow = prctile(dgC(:),3);
    rgmhigh = prctile(dgC(:),99.9);
    imagesc(sqrt(grange.^2-mean(z1)^2),azdis,dgC, [rgmlow rgmhigh]);
    %colorbar
    xlabel('ground range (m)'); ylabel('azimuth (m)');
    colormap('gray')
    title(sprintf('Georectified Multilook RDA image'))
end
```

Appendix C

Digital Receiver Design for Offset IF LFM-CW SAR

In 2004, the Microwave Earth Remote Sensing (MERS) lab at Brigham Young University (BYU) developed the microSAR, demonstrating a small and low-cost LFM-CW SAR system [1]. Building on this experience, BYU partnered with Artemis Inc. to develop the microASAR, a more robust and capable system, that overcomes many of the limitations of the microSAR design [2]. A key feature of the microASAR design is an oversampled digital receiver. The oversampling provides three main benefits, namely: the de-chirped signal can be at an arbitrary intermediate frequency (IF), enabling better RF filtering; quantization noise is reduced via digital filtering; and the flexibility to enable the SAR to operate in both de-chirped and pulsed modes.

C.1 Offset IF LFM-CW SAR

The LFM-CW system outlined in Section 2.3 is the basis for the BYU microSAR, which worked adequately, but it was found that the high-Q filter used to suppress the feed-through causes distortion to the echo data due to the filter's long impulse response. This distortion can be avoided using an offset de-chirp. We term this system an offset IF LFM-CW SAR.

The flow diagram for an offset IF LFM-CW SAR is shown in Fig. C.1. To generate the dechirp signal, the transmit signal is first partially mixed down using ω_{IF} and filtered. This signal is then mixed with the received signal, the resulting difference components are similar to the ones in the traditional LFM-CW but are at an offset IF. With the signal of interest at a higher IF frequency, it is easier to find a high-Q filter that has linear phase, sharp cutoff frequencies, and better suppresses the feed-through.

C.2 Digital IF SAR

The design of the digital receiver for offset IF LFM-CW SAR system can follow traditional LFM-CW receiver design if another mix down stage is added to mix down the offset dechirp signal or if the sampling frequency and analog to digital converter (ADC) are carefully selected to sub-sample the offset dechirp signal; however, improved performance and flexibility can be achieved when using a high-speed ADC and an FPGA. Choosing an ADC that can sample the full bandwidth of the received chirp enables pulse mode operation as well as dechirp operation at an arbitrary IF. Incorporating an FPGA further enhances the design by providing enough I/O ports to integrate a large number of components and communication devices. The FPGA provides for various modes of operation. This section

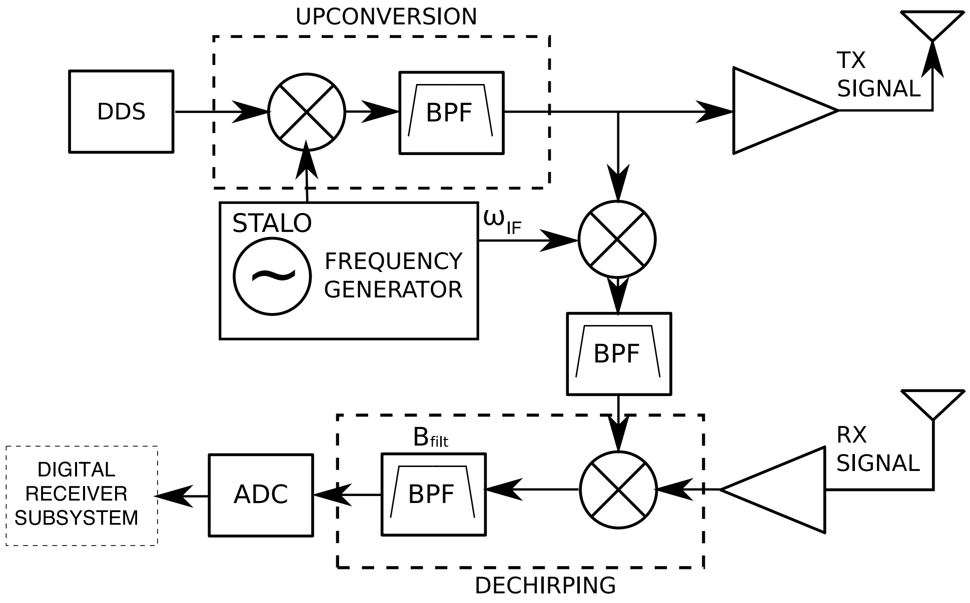


Figure C.1: A high-level flow diagram for an offset IF homodyne LFM-CW SAR. The received signal is mixed down with a frequency shifted version of the transmit signal. The resulting signal is at an offset IF. The higher IF center frequency enables the use of a better filter without distorting the signal.

describes the principles of oversampling and filtering used in implementing the microASAR. The FPGA implementation is described in the next section.

C.2.1 Oversampling

When the relative power consumption costs can be neglected and the ADC resolutions are comparable, it is best to sample the received signal at the highest rate possible to enable quantization noise reduction. Because the quantization noise is independent of sampling frequency, sampling the signal at a higher rate spreads the noise spectrum over a wider bandwidth and thereby lowers the quantization noise power over the signal bandwidth.

Introductory digital signal processing courses often neglect the effects of amplitude quantization; however, in a LFM-CW system the quantization of the incoming signal is often the major source of noise. When a sufficiently random signal is quantized at a step size q, it is equivalent to the addition of uniform white noise in the range of $\pm\frac{q}{2}$ (see [75]). By applying an appropriate filter after sampling the signal, the quantization noise to signal ratio (QNSR) is decreased by approximately 3dB for every factor of 2 the signal is oversampled.

Figure C.2 illustrates the SNR benefit available by oversampling. In Fig. C.2a a simulated LFM-CW echo is sampled at a rate just above Nyquist. The separation between the signal and the quantization noise is about 64dB. The signal in Fig. C.2b is oversampled by a factor of approximately 18 with the same number of bits as in Fig. C.2a. The separation between the signal and noise is now about 75dB. By appropriately applying a bandpass filter, oversampling enables a 11dB QNSR decrease.

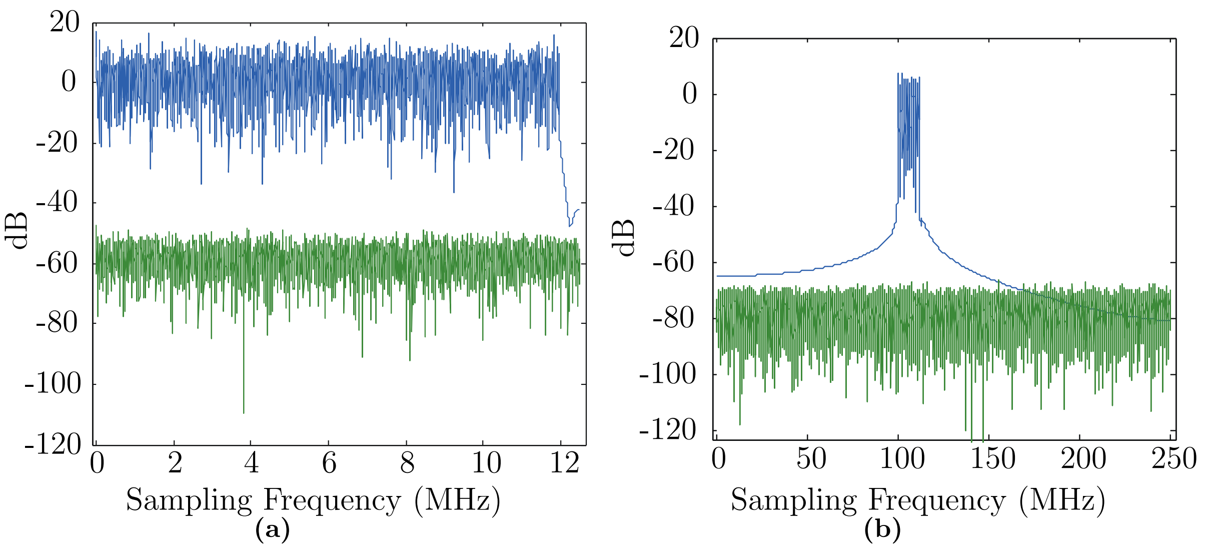


Figure C.2: Plots depicting the spectrum of the subsampled(a) and oversampled(b) signal with the respective quantization noise. Note that the oversampled signal has a larger signal to noise separation of approximately 11dB. Note that the signal spectrum has the same bandwidth in both plots but the frequency scaling of the plots is different.

C.2.2 Data rate reduction

Once the signal is sampled, the next step is to reduce the data rate (decimate) without compromising the integrity of the data. There are two steps in the process. The first is to filter the quantization noise and bring the signal to baseband such that the filtered signal sampling frequency can be reduced. All of the filters implemented are digital polyphase filters, which combine the operation of a filter and a decimator enabling a reduction in FPGA resources. The second step in reducing the data rate is presumming. Presumming consists of adding sequential echoes together and has the effect of low pass filtering the Doppler spectrum. Presumming can be used on the microASAR data because the high PRF used to separate the feed-through and the first target, as explained in Appendix C.1, is much higher than required by the Doppler bandwidth of the signal.

The order of the presumming and filtering are interchangeable from a signal processing point of view, but the order greatly affects the memory and hardware requirements of the implementation, as discussed in the following section. Also it should be noted that after every signal processing operation the bitwidth of the data path is increased to prevent overflow [76].

C.3 FPGA implementation

In order to achieve the desired system flexibility and high performance obtainable using the principles described in the previous section, the microASAR digital receiver is equipped with a 12-bit 500MHz ADC and a Xilinx Virtex-5 FX-30T FPGA. This combination enables the microASAR to sample the full 200MHz bandwidth of the transmitted signal as well as operate in various dechirp modes. This section briefly describes the general design of the FPGA implementation for dechirp operation and outlines the design strategy used.

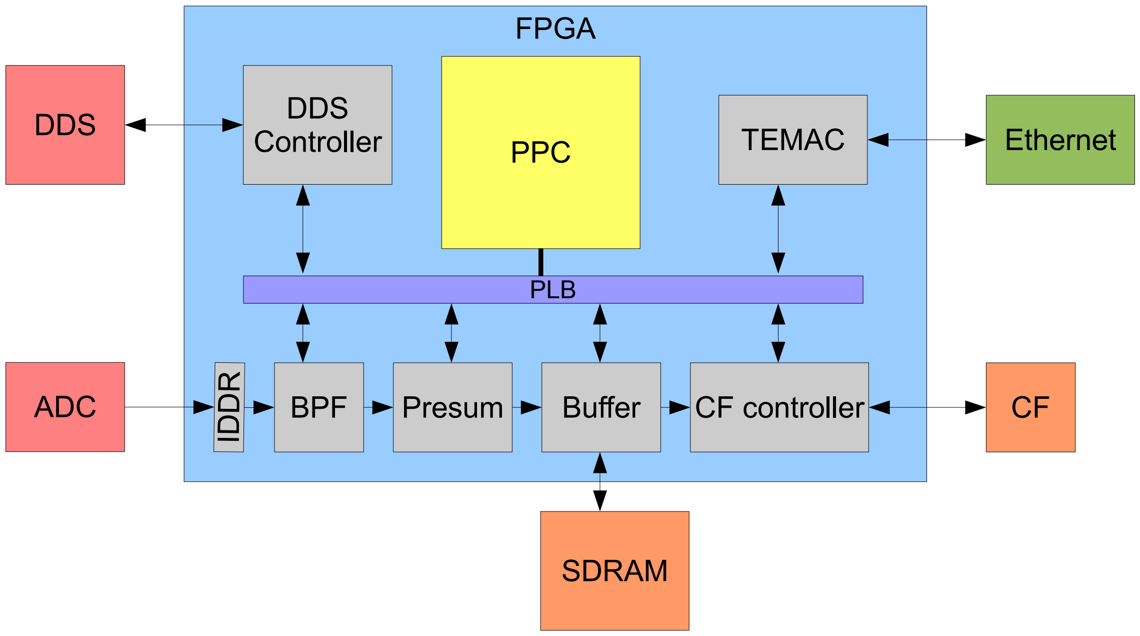


Figure C.3: Block diagram of the FPGA implementation. Note that the order of signal data path can change depending on design parameters.

The general outline of the FPGA implementation is shown in Fig. C.3. The embedded PowerPC processor on the FPGA is used to control and coordinate the operation of the complete digital receiver, and most operational parameters can be set by communicating with the powerPC via ethernet. The normal signal data path goes from the ADC to the filter subsystem and then through a buffer to the compact flash cards. Alternatively, the data path can be interrupted and streamed across the ethernet port. The data from the ADC is immediately broken into two interleaved data paths such that the filter clock rate can be reduced by a factor of two to ease timing constraints. The two data streams are 180 degrees out of phase with each other and are recombined following the presum stage.

The filter subsystem includes all the filtering, presumming, and decimation steps, and can be configured for differing operational parameters. In order to handle a large range of operations and to reduce FPGA resources, the filter subsystem consists of a polyphase filter followed by the presummer. The polyphase filter reduces quantization noise and limits the signal spectrum such that the signal can be translated to DC by decimating the digitally sampled data. For the microASAR this is done by applying a 12MHz wide BPF starting at the ω_{IF} and decimating by a factor of 20, providing approximately 3.3 bits of increased resolution. This effectively brings the effective number of bits (ENOB) of the signal equivalent to 16bit ADCs that are only available for lower sampling rates. The presumming is performed after the filter in order to reduce memory requirements so that presumming can be computed in on-chip memory.

Alternatively, performing presumming first reduces the multipliers required in filtering. In most cases though, this requires an external high-speed memory, increasing power consumption and development time. Replacing the single polyphase filter with a polyphase filter followed by a mixer and a polyphase low pass filter enables a larger bandwidth to be stored and the “empty” spectrum due to the distance from the SAR to the nearest target to be discarded. Also, instead of the mixer and low-pass filter an FFT could be used. Both of these methods require more FPGA resources and add noise to the signal due to the fixed point multiplies and sine/cosine lookup tables.

This simple setup minimizes the FPGA resources and can be operated for a variety of applications simply by varying the PRF. Decreasing the PRF decreases the chirp rate, compressing the targets in the dechirped data. The analog and digital filters effectively range gate the dechirped data. So by varying the PRF from 7-14 kHz the SAR can be operated with altitudes of 5-1000 m, a maximum swath width of 30-2500 m, and a velocity of 0-150 m/s. For a more detailed explanation see [2]. Some values of these parameters are obviously impractical for airborne operation, but the microASAR can be used for ground based systems as well.

C.4 Summary

This appendix describes my work with the design and implementation of the digital receiver and controller subsystem for the microASAR. Using a high-speed ADC and an FPGA, allows for the system to be used both for LFM-CW and pulsed SAR in various modes of operation. With the oversampling design, the dechirped signal can be be at an arbitrary IF and the quantization noise is reduced with the application of appropriate digital filters. The images generated from data collected with this system are shown in various chapters of this work, and they demonstrate that the design and implementation of this system have helped make the microASAR an effective tool for a variety of scientific and military applications.