UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

SAR IMAGE FORMATION VIA SUBAPERTURES AND 2D BACKPROJECTION

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

CALLIN SCHONE

Norman, Oklahoma

2020

SAR IMAGE FORMATION VIA SUBAPERTURES AND 2D BACKPROJECTION

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Nathan Goodman, Chair

Dr. Jay McDaniel

Dr. Justin Metcalf

Table of Contents

List of Figures

List of Tables

Abstract

Radar imaging requires the use of wide bandwidth and a long coherent processing interval, resulting in range and Doppler migration throughout the observation period. This migration must be compensated in order to properly image a scene of interest at full resolution and there are many available algorithms having various strengths and weaknesses. Here, a subaperture-based imaging algorithm is proposed, which first forms range-Doppler (RD) images from slow-time subintervals, and then coherently integrates over the resulting coarse-resolution RD maps to produce a full resolution SAR image. A two-dimensional backprojection-style approach is used to perform distortion-free integration of these RD maps. This technique benefits from many of the same benefits as traditional backprojection; however, the architecture of the algorithm is chosen such that several steps are shared with typical target detection algorithms. These steps are chosen such that no compromises need to be made to data quality, allowing for high quality imaging while also preserving data for implementation of detection algorithms. Additionally, the algorithm benefits from computational savings that make it an excellent imaging algorithm for implementation in a simultaneous SAR-GMTI architecture.

Chapter 1

Introduction

1.1 Motivation

Synthetic aperture radar (SAR), first introduced in 1951, brought forward a new realm of radar imaging that greatly enhanced surveillance potential. Whereas large antenna apertures were previously needed to achieve fine resolution in azimuth, it had now been determined that a single antenna could synthesize the same array by transmitting pulses while moving, allowing integration into airborne platforms, and achieving twice the resolution of the real aperture [1, 2]. A radar's range resolution is inversely proportional to its bandwidth, i.e. as bandwidth is increased range resolution becomes more fine. A radar's azimuth, or cross-range, resolution is inversely proportional to its aperture size, which is no longer limited by the *physical* size of the aperture. The introduction of SAR has garnered a wide range of applications: from militaristic reconnaissance and intelligence gathering to general terrain mapping of Earth and other celestial bodies.

As the demand for higher and higher image quality increases, the amount of data required grows as well. Digital processors often require compromises to be made to image quality so that the data can be processed efficiently, often limiting the size or resolution of the image. Early imaging algorithms relied heavily on

these compromises, but over time, algorithms have become more robust to allow for higher image quality without reliance on compromises. This increase in image quality comes at the cost of increased computation, but as modern computing improves, the increased computational requirements can be realized more and more efficiently. However, even with improved computing, high quality images are still formed relatively slowly. The best algorithms produce full resolution images, limited only by the operating parameters of the system, in the most efficient manner possible.

1.2 Background

Numerous imaging algorithms exist that make compromises between image quality and computational efficiency. One algorithm that can be used to produce an optimal SAR image is the matched filter algorithm. The matched filter algorithm applies a filter that is matched to a particular target's characteristics, i.e. a target's position in the scene; however, a new matched filter is required for every target hypothesis, requiring $\mathcal{O}(N^4)$ operations to form an NxN image [3]. The computational requirements of this algorithm are tremendous and prevent its use in most applications. The backprojection algorithm, mathematically equivalent to the matched filter algorithm, is a technique that can be used to produce a very high-quality image while benefiting from reduced computation [3]. This algorithm aligns and coherently sums the returns from scatterers over a coherent processing interval (CPI). The returns are summed once per pulse, allowing the backprojection algorithm to produce images as phase history is recorded. While the backprojection algorithm is significantly more efficient than the matched filter algorithm and its architecture allows for real-time implementation, it is still extremely computa-

tionally intensive, making real-time image formation impractical with modern day computing. The backprojection algorithm requires $\mathcal{O}(N^3)$ operations to form an NxN image, a factor of N smaller than the matched filter algorithm but still very large compared to other fast transform-based algorithms.

One such algorithm is the polar format algorithm (PFA), whose use of the extremely efficient fast Fourier transform (FFT) allows near real-time imaging. The reduced computational requirements of the PFA are very significant, requiring only $\mathcal{O}(N^2 logN)$ operations to form the same NxN image; however, the PFA suffers from a severe reduction in image quality [1, 4, 5]. The PFA forms an image by performing a highly efficient 2D FFT on polar-formatted data. Proper imaging by the PFA requires that the far-field approximation hold, i.e. the platform is far away when compared to the spatial extent of the scene, imposing strict limitations on scene size [4]. Images formed by the PFA exhibit geometrical warping and defocusing that become more prominent for large scene sizes. Using a second-order Taylor Series expansion, it can be shown that proper imaging of a scene is limited to a radius of 1,100 m for a SAR system operating at X-band with a 1 foot range resolution at a range of 110 km [6, 7]. This may seem like adequate area coverage, but when compared to the arbitrary scene size allowed by the matched filter algorithm and backprojection, the PFA is at a severe disadvantage.

Several techniques have been proposed that increase the properly imaged scene size allowed by the PFA, but these come at the cost of increased computation. Several methods increase area coverage by applying a spatially-variant filter that compensates for phase errors after the image has been formed [8, 9]. The spatially-variant nature of the filter requires that a new filter be created for each desired pixel within the scene of interest. Each new filter depends on the range and cross-range of each desired pixel, making pixel specific filtering highly inefficient. Instead, fil-

ters are typically approximated for a group of pixels such that the residual quadratic phase error of each pixel in the group falls below a specified level, typically between $\pi/4$ and $\pi/2$ radians [7]. After this filter is applied, an interpolation is usually performed that mitigates distortion by compensating for the remaining linear phase errors. Filter calculation can also be simplified by the use of the dual format algorithm (DFA) which, instead of forming an image directly to a Cartesian grid and filtering, forms an image on an arbitrary grid that is chosen such that the quadratic phase errors to be corrected are a function of only range [7]. Each new filter for the DFA can be computed once for an entire row of pixels within the image, reducing computational requirements further. An interpolation is then required to transform the image from the arbitrary grid to the Cartesian grid desired for image output.

Another advantageous use of the FFT is the ability to perform Doppler processing with tremendous efficiency. Implementing an FFT across slow-time will yield a range-Doppler map of the CPI; however, targets will migrate in range and Doppler over the course of the CPI. Due to the SAR collection geometry, even stationary targets and clutter will exhibit some migration for any reasonably sized CPI. This migration is extremely undesirable as it results in a range-Doppler map with degraded resolution. This migration could be mitigated by limiting the radar's bandwidth or CPI duration, but this would negatively impact image quality, as discussed further in Chapter 2. To avoid these compromises, another method must be used; linear range migration can be fully accounted for by the Keystone Transformation without any compromise to the data collection parameters [10, 11]. However, even stationary targets have a higher-order motion profile due to SAR collection geometry, resulting in the inability of the Keystone transform to fully compensate for the range migration present in a full CPI. While the Keystone transform may not be suitable for application to a full CPI, it can be applied to subintervals of the full CPI

that are chosen such that range migration is approximately linear. These sub-CPIs, or subapertures, containing only approximate linear range migration, allow the Keystone transform to fully focus target energy in each sub-CPI's range-Doppler map. Each subaperture creates an image that is fully focused with coarse resolution and poor image quality.

The inherent resolution achievable by a SAR system is defined by its operating parameters, and in order to obtain the full resolution image, all data must be used. For subaperture-based approaches, this means all subapertures must be used in the production of the full resolution image. There are various subaperture-based techniques that can be used for SAR image formation, but all such techniques must perform the same two key steps: first, low resolution images are formed for each subaperture, then these images are aggregated into a full resolution image. Subaperture-based PFA can utilize subapertures to increase area coverage over the traditional PFA. In this technique, discussed in [5], tiers of subapertures in range and azimuth are used to mitigate migration and phase errors, allowing for increased area coverage. The factorized backprojection (FBP) has been introduced as a backprojection style combination of subapertures [12]. First, the FBP forms beams from the polar formatted data of each subaperture, then combines sets of these subapertures by interpolating their beams to create the beams of the larger subapertures. The FBP continues the aggregation of adjacent subapertures and beam formation until all subapertures are used. The resulting beams now lie on a Cartesian gird, allowing a 2D FFT to produce a full resolution image.

While there exists a plethora of imaging algorithms, new algorithms are still required to increase the efficiency at which data can be processed. This thesis will serve as a focused discussion on one such algorithm, called 2D range-Doppler backprojection. This technique is a subaperture-based approach that utilizes the highly

efficient FFT to first create range-Doppler maps for each subaperture. Each subaperture is chosen such that targets exhibit only linear range migration, allowing the Keystone transform to fully focus the range-Doppler maps of each. While target energy is focused in the range-Doppler map of each sub-CPI, range and Doppler values vary across sub-CPIs as the SAR platform traverses its flight path. This variation is accounted for by a backprojection-style summation of the returns from multiple sequential subapertures. This summation accounts for phase changes from sub-CPI to sub-CPI, allowing for coherent integration of all sub-CPIs. The coherent integration of all sub-CPIs will yield a full resolution image. This technique differs from previously described subaperture-based approaches by the use of the Keystone transform to focus individual sub-CPIs and the interpolation of range-Doppler maps to form coarse resolution images from the sub-CPIs. The greatest advantage of this 2D backprojection approach over traditional backprojection is the reduced number of interpolations needed to form a full resolution image. While traditional backprojection requires a single 1D interpolation for each pulse, 2D backprojection requires only a single 2D interpolation for each sub-CPI, often composed of tens or hundreds of pulses. Performing a 2D interpolation is more computationally intensive than a 1D interpolation, but the reduced number of interpolations needed makes 2D backprojection much more efficient. The architecture of the 2D backprojection has been chosen such that several processing steps are shared with many post-Doppler GMTI techniques [13, 14, 15, 16, 17], allowing for a simultaneous SAR-GMTI pipeline that can efficiently process data to be imaged and tested for moving targets [18, 19, 20].

Several techniques have been proposed for efficient simultaneous SAR-GMTI. One popular choice for simultaneous processing is the displaced phase center antenna (DPCA) technique [21]. Here, multiple antenna elements are used to create

multiple images of the scene. The spacing between these elements is chosen such that the second pulse of the second element is transmitted and received from the same location as the first pulse by the first element. This is achieved by spacing the elements such that the platform moves by one interelement spacing between each transmitted pulse. The images formed by the first and second elements will, therefore, be identical for stationary targets, but moving targets will have a non-stationary location in the scene and will result in differences in the images. By subtracting one image from the other, the difference between the images will yield returns from moving targets only. While this technique is useful for locating moving targets, it does little to help identify target parameters. DPCA can be used in conjunction with along-track interferometric SAR (AT-InSAR) to allow for calculation of target velocity [19]. AT-InSAR takes advantage of the same displaced phase centers as DPCA and also forms SAR images for each element. However, unlike DPCA, AT-InSAR looks at the phases of the images instead of the magnitudes. Using the phases of the images allows for AT-InSAR to determine target radial velocity. Using multiple phase centers would allow 2D backprojection to be used in a SAR-GMTI architecture such as this; however, the data is formatted such that post-Doppler GMTI techniques can also be utilized with little extra computation, allowing for a wide range of possible applications.

Relevant SAR theory will be introduced in Chapter 2. Chapter 2 will then fully describe the signal model needed for development of the imaging algorithm. Chapter 3 will derive the imaging algorithm, including all relevant math and algorithm steps needed for efficient processing. Chapter 4 will provide an in-depth analysis of the results on simulated and real data. Chapter 5 will discuss the computational requirements of the algorithm, including memory requirements and relative computation times of this method compared to traditional backprojection. Finally, in

Chapter 6, conclusions will be drawn on the capabilities of the algorithm, and future work on developing GMTI processing will be discussed.

Chapter 2

SAR Theory and Signal Model

In the previous Chapter, several existing SAR imaging algorithms were introduced. These imaging algorithms all exhibit different image formation capabilities that serve a wide range of purposes. From these, 2D backprojection, a subaperture-based imaging technique, was introduced as a computationally efficient simultaneous SAR-GMTI imaging algorithm. This technique performs a backprojection-style combination of subapertures to form a high resolution image while also retaining data in a format that is useful for implementation of GMTI techniques.

In this Chapter, relevant SAR background information will be provided. Next, the general SAR signal model used for deriving the 2D backprojection algorithm will be introduced. Finally, the Keystone Transformation, used for range migration correction, will be introduced for a simplified geometry that will later be expanded for application to SAR geometry by 2D backprojection.

2.1 Introduction to SAR

SAR systems utilize platform motion to create a synthetic array of data. As the platform travels, pulses are transmitted periodically at a specified interval, known as the pulse repetition interval (PRI), given by the inverse of the pulse repetition

frequency (PRF). Returns are collected for each pulse and sampled by the ADC once converted to baseband. The returns are passed through a filter that is a copy of the time-reversed conjugated transmitted signal. This filter is known as the matched filter, and passing returns through it equates to convolving the received signal with the matched filter, yielding the autocorrelation function of the baseband waveform. Fundamentally, the autocorrelation is a measure of how similar the received signal is with the matched filter at different time delays. When a target is present, the received signal should be nearly identical to the transmitted signal, so the output of the matched filter will result in a strong peak. When no target is present, the output of the matched filter will be very weak as the received signal is primarily noise and does not match the transmitted signal.

The width of the peak of the matched filter response defines the resolution of the waveform. A simple rectangular pulse has a peak width equal to the time duration of the pulse, requiring exceptionally short pulses to achieve fine resolution. While short pulses exhibit very high bandwidth, they are undesirable as they limit the amount of energy that can be transmitted in a single pulse without also increasing peak power requirements. Long pulses are therefore required; however, lengthening the pulse decreases instantaneous bandwidth and degrades resolution. Therefore, modulation schemes must be employed that decouple the time duration and bandwidth of the waveform, allowing each to be independently specified. These modulation techniques allow the output of matched filtering to be compressed into a more narrow peak, giving rise to the name *pulse compression*. Whereas the resolution of a simple pulse is determined by the time-duration of the pulse, modulation techniques allow bandwidth, β, to be chosen independently of time-duration, allowing a resolution of $1/\beta$. This peak width has units of seconds, but can easily be

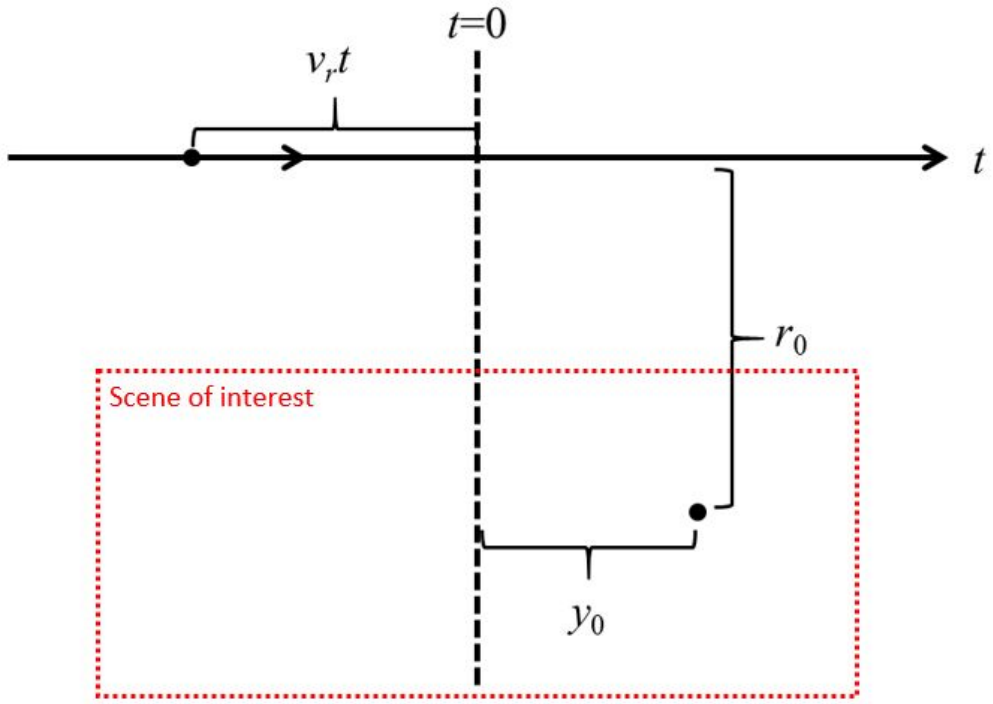


Figure 2.1: SAR platform / pixel geometry

converted into range units, giving the range resolution of the system as

$$\Delta R = \frac{c}{2\beta}, \tag{2.1}$$

where the factor of two in the denominator results from the two-way propagation of the wave as it travels to the target and back.

The cross-range resolution is now desired to find the achievable resolution of the SAR system. As the platform travels past the scene shown in Figure 2.1, a target will be observed for T_a seconds. The Doppler resolution, ΔF_d, of the system is given by $1/T_a$, requiring long observations for a fine azimuth resolution. The measured Doppler value of any target within the scene is dependent on the velocity of the platform and its location within the scene. Using the geometry defined in

Figure 2.1, we see that the radial component of the velocity, v_{rad}, is the projection of radar velocity, v_r, onto the line-of-sight vector. Equivalently, v_{rad} is the product of v_r and the cosine of the angle between the velocity vector and the line-of-sight vector. Therefore, the radial component of the radar's velocity is

$$v_{rad} = v_r cos(\theta) = v_r \frac{y_0}{R_0}, \tag{2.2}$$

where the radar is assumed to be located at $t = 0$ and R_0 is the magnitude of the line-of-sight vector and can be approximated as constant for determining resolution. Cross-range resolution, ΔCR, can now be determined by substituting $\frac{2v_{rad}}{\lambda}$ for F_d, giving

$$\Delta CR = \frac{R_0 \lambda}{2v_r T_a}, \tag{2.3}$$

where $2v_r T_a$ is equivalent to the total distance traversed by the SAR platform during the observation. It is important to note that while resolution in range is dependent only on radar operating parameters, cross-range resolution is spatially variant. This variance is accentuated for larger scenes and scenes that are near the radar. In practice, cross-range resolution can be higher than this theoretical value. If a non-linear flight path is chosen, e.g. circular SAR flight path, the angle subtended is such that the achievable resolution is much higher than the value derived here for a linear flight geometry. Therefore, to achieve high resolution in range and cross-range, SAR systems typically utilize wide bandwidth and long observation times. Long observations allow the platform to travel the distance required to create the desired synthetic aperture. The remainder of this chapter will derive the signal model needed for the derivation of 2D backprojection.

2.2 SAR Signal Model

Consider a SAR system with an idealized two-dimensional geometry and a linear flight path as shown in Figure 2.1. The system transmits a general bandpass waveform of the form

$$\tilde{x}_t(t) = a(t)cos(2\pi F_0 t + \theta(t)), \tag{2.4}$$

with complex baseband equivalent of

$$x_t(t) = a(t)exp(j\theta(t)), \tag{2.5}$$

where F_0 is the center frequency of the band, $\theta(t)$ represents a time-varying phase modulation, and $a(t)$ represents a time-varying amplitude modulation. The received signal (ignoring amplitude factors resulting from target parameters) is a delayed copy of the transmitted bandpass waveform given by

$$\tilde{x}_r(t) = a(t-\tau)cos(2\pi F_0(t-\tau) + \theta(t-\tau)), \tag{2.6}$$

with complex baseband equivalent

$$x_r(t) = a(t-\tau)exp(-j2\pi F_0\tau)exp(j\theta(t-\tau)), \tag{2.7}$$

where τ is the two-way propagation delay to a particular scatterer. The received baseband signal is passed through a matched filter, which is the time-reversed, complex conjugate of the transmitted waveform. Therefore, $h(t) = x_t^*(-t)$, and

the output of the matched filter, given by $w(t)$, is

$$\begin{aligned} w(t) = h(t) * x_r(t) &= \int_{-\infty}^{\infty} x_r(u)h(t-u)\,du \\ &= exp(-j2\pi F_0\tau) \int_{-\infty}^{\infty} a(u-\tau)exp(j\theta(u-\tau))a(t-u)exp(j\theta(t-u))\,du \\ &= exp(-j2\pi F_0\tau)R_{xx}(t-\tau) \end{aligned} \tag{2.8}$$

which is the autocorrelation function of the baseband waveform, delayed by the propagation delay of the received signal and scaled by a propagation phase term.

During a CPI, the waveform is transmitted N times at a uniform PRI, given by T_r. Adding this slow-time dimension, the baseband signal received on the n^{th} pulse, where $n \in [-(N-1)/2, (N-1)/2]$, due to a single scatterer is a delayed version of the transmitted signal given by

$$x_r(t,n) = a(t-\tau_n)exp(-j2\pi F_0\tau_n + j\theta(t-\tau_n)). \tag{2.9}$$

where τ_n is the two-way propagation delay to a particular scatterer on the n^{th} pulse. Using (2.8), the output of the matched filter due to the reflected signal on the n^{th} pulse is then

$$w(t,n) = exp(-j2\pi F_0\tau_n)R_{xx}(t-\tau_n), \tag{2.10}$$

with a Fourier transform in the fast-time dimension of

$$z(F,n) = exp(-j2\pi(F_0+F)\tau_n)S_x(F), \tag{2.11}$$

where $S_x(F)$ is the Fourier Transform of $R_{xx}(t)$ and F is the fast-time frequency

axis. The slow-time varying time delay, τ_n, introduces a phase rotation that also varies from pulse to pulse.

The range to a particular scatterer on the ground is now desired. A continuous slow-time variable, t_s, will be used to derive the range to any given scatterer on the ground, then the derived ranges will be sampled in slow-time at the PRF to obtain data in the traditional fast-time vs. slow-time format. The range to a scatter for any slow-time value is given by

$$r(t_s) = \sqrt{r_0^2 + (v_r t_s - y_0)^2}, \tag{2.12}$$

where v_r is the radar velocity, and y_0 and r_0, as shown in Figure 2.1, are the along-track and cross-track scatterer coordinates, respectively. The Taylor expansion of the range expanded around $t_s = 0$, ignoring third-order and higher terms, is

$$r(t_s) = r(0) + \frac{r'(0)}{1!}(t_s - 0) + \frac{r''(0)}{2!}(t_s - 0)^2 + \ldots, \tag{2.13}$$

where $r'(0)$ and $r''(0)$ represent the first and second time derivatives of $r(t_s)$ with respect to slow-time variable t_s, evaluated at the expansion point $t_s = 0$. We now differentiate $r(t_s)$ to find

$$r'(t_s) = v_r\big(r_0^2 + (v_r t_s - y_0)^2\big)^{-\frac{1}{2}}\big(v_r t_s - y_0\big), \tag{2.14}$$

and again to find

$$\begin{aligned} r''(t_s) = v_r^2\Bigg[&\big(r_0^2 + (v_r t_s - y_0)^2\big)^{-\frac{1}{2}} \\ &- \big(v_r t_s - y_0\big)^2\big(r_0^2 + (v_r t_s - y_0)^2\big)^{-\frac{3}{2}}\Bigg]. \end{aligned} \tag{2.15}$$

We can now define $R_0 = r(0)$ and evaluate equations (2.14) and (2.15) at $t_s = 0$, giving

$$r'(0) = -\frac{v_r y_0}{R_0}, \tag{2.16}$$

$$r''(0) = -\frac{v_r^2 y_0^2}{R_0^3} + \frac{v_r^2}{R_0}. \tag{2.17}$$

Using equations (2.16) and (2.17), the slow-time dependent range to a scatterer is

$$r(t_s) = R_0 - \frac{v_r y_0}{R_0} t_s - \frac{1}{2}\frac{v_r^2 y_0^2}{R_0^3}(t_s)^2 + \frac{1}{2}\frac{v_r^2}{R_0}(t_s)^2, \tag{2.18}$$

We can now use this slow-time varying range to calculate a range rate and convert this to Doppler frequency, given by

$$\begin{aligned} F_d(t_s) &= -\frac{2}{\lambda}\frac{dr(t_s)}{dt} \\ &= -\frac{2F_0}{c}\left(- \frac{v_r y_0}{R_0} - \frac{v_r^2 y_0^2}{R_0^3} t_s + \frac{v_r^2}{R_0} t_s \right) \end{aligned} \tag{2.19}$$

where λ is the wavelength of the carrier frequency. Using (2.18), the slow-time dependent range to the scatterer calculated on each pulse (i.e., at $t_s = nT_r$ where $t_s \in [-T/2, T/2]$) is given by

$$r(nT_r) = R_0 - \frac{v_r y_0}{R_0} nT_r - \frac{1}{2}\frac{v_r^2 y_0^2}{R_0^3}(nT_r)^2 + \frac{1}{2}\frac{v_r^2}{R_0}(nT_r)^2. \tag{2.20}$$

We can now use this range to calculate the pulse-dependent two-way propagation delay, τ_n, yielding

$$\tau_n = \frac{2}{c}\left[R_0 - \frac{v_r y_0}{R_0} nT_r - \frac{1}{2}\frac{v_r^2 y_0^2}{R_0^3}(nT_r)^2 + \frac{1}{2}\frac{v_r^2}{R_0}(nT_r)^2 \right]. \tag{2.21}$$

The pulse-varying, instantaneous Doppler shift on each pulse is, therefore,

$$\begin{aligned} F_d(nT_r) &= -\frac{2}{\lambda}\frac{dr(nT_r)}{dt} \\ &= -\frac{2F_0}{c}\left(-\frac{v_r y_0}{R_0} - \frac{v_r^2 y_0^2}{R_0^3} nT_r + \frac{v_r^2}{R_0} nT_r \right). \end{aligned} \tag{2.22}$$

The complete received signal from each pulse is the superposition of returns from all scatterers in the scene on the n^{th} pulse. An imaging algorithm must focus each pixel's contributions in order to produce a full resolution image of the scene. The received signal as sampled by the receiver, given in (2.10), is in the fast-time vs. slow-time format that is ideal for implementation of traditional backprojection, as the backprojection algorithm requires interpolation of the range-axis and summation for every pulse. However, this format is not useful for other algorithms such as the PFA. Reformatting of the data is the first step performed by any algorithm that requires a specific format for processing. As the name of the algorithm implies, the PFA requires the data to be in a polar format. For 2D backprojection, fast-time frequency vs. slow-time is the most desirable as discussed further in Chapter 3. Chapter 3 will use the signal given in (2.11) combined with the expressions for the pulse-dependent propagation delay and pulse-dependent Doppler shift, given in (2.21) and (2.22), to derive the signal model of 2D backprojection.

2.3 The Keystone Transform

The exponential term in (2.10) contains the pulse-dependent propagation delay to a target, resulting in a varying phase measured on each pulse. For simplicity of the initial derivation of the Keystone Transformation, we now assume a stationary radar. Therefore, the two-way propagation delay, τ_n, now depends only on the mo-

tion of the target. In general, the ability to perform the Keystone transform only depends on the motion of the target relative to the platform, so we could equivalently assume a stationary target with a moving platform. For a target moving radially at some constant velocity, v_{rad}, the pulse-dependent range and propagation delay are

$$r(nT_r) = R_0 - v_{rad} nT_r, \tag{2.23}$$

$$\tau_n = \frac{2}{c}\big(R_0 - v_{rad} nT_r\big), \tag{2.24}$$

where $v_{rad} nT_r$ is the pulse-to-pulse range migration. This term is typically very small, but high-velocity targets and long observation times will result in a non-trivial range migration. We can now substitute this expression for τ_n into (2.11) and have

$$\begin{aligned} z(F,n) &= exp(-j4\pi(F_0+F)(R_0 - v_{rad} nT_r)/c) S_x(F) \\ &= exp(-j4\pi(F_0+F)R_0/c) exp(j4\pi(F_0+F) v_{rad} nT_r/c) S_x(F) \end{aligned} \tag{2.25}$$

The first exponential term in this expression is the range-dependent phase term, while the second exponential contains information on the slow-time phase progression. Performing an FFT in the slow-time dimension allows us to extract Doppler information from the second exponential via the pulse-to-pulse phase progression. However, it is clear that the pulse-to-pulse phase progression is dependent on fast-time frequency, F. If range migration during the full CPI, given by $v_{rad} NT_r$, is small relative to the range resolution of the system, the effects of the coupling between n and F can be neglected, and a target will remain in a single range bin for the entire CPI. However, use of wide bandwidth and long observation times will often result in range migration that will be greater than the range resolution, resulting in a

non-trivial phase progression dependence on F. The dependence on F means that the slow-time phase progression for frequencies greater than F_0 will be faster than for frequencies less than F_0. Target range will, therefore, change throughout the CPI, resulting in target energy being spread over multiple range bins. Additionally, frequency dependent phase progression will result in a blurred Doppler response. Figure 2.2 shows a simulated target that exhibits moderate range migration and the resulting unfocused range-Doppler map. Simulation parameters for this data are given in Table 2.1. Over the course of the CPI, the target will migrate 5.12 meters away from the radar. The range resolution of this system with 400 MHz of bandwidth is .375 meters, resulting in target range migration that spans several range bins. Target energy is spread over multiple range-Doppler bins, resulting in poor

Parameter	Value
β	400 MHz
PRI	.5 ms
N	1024
R_0	5 km
v_{rad}	10 m/s

Table 2.1: Radar and target parameters for simulated dataset showcasing range migration

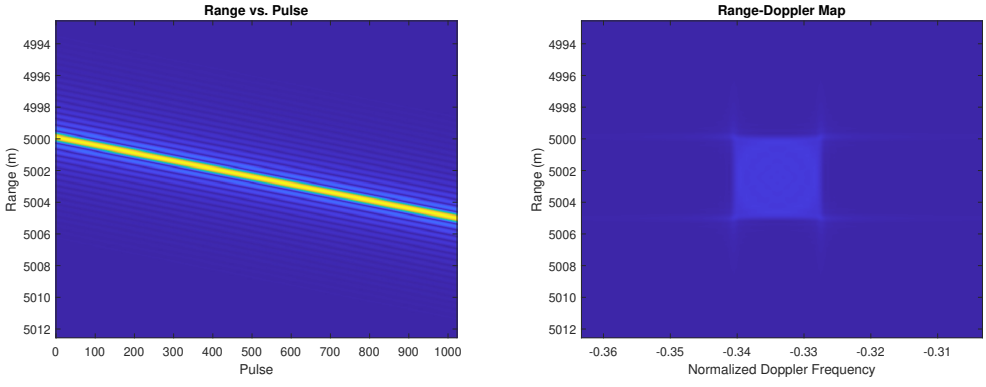


Figure 2.2: Range migration of simulated target moving away from the radar at 10 m/s for a CPI duration of 5.12 seconds

focusing of the range-Doppler map.

Obviously the response in the previous figure is highly undesirable and must be accounted for to preserve adequate image quality. The desired response in the Doppler domain is a sinc function centered at Doppler frequency, F_d, with an inverse FFT of $exp(j4\pi F_0 v_{rad} nT_r/c)$. To obtain this form, we can simply multiply (2.25) by a phase of $exp(-j4\pi F v_{rad} nT_r/c)$. While this would completely remove the dependence on F and yield the desired range-Doppler response, it will only work for a single value of v_{rad}. A more general technique is desired that will work for any arbitrary value of v_{rad}, and by clever choice of a rescaling of the slow-time axis, this can be accomplished. The received signal is sampled at an interval of nT_r, but if we perform a frequency-dependent scaling of the slow-time axis to some new slow-time variable, τ', given by

$$(nT_r)' = \big(\frac{F_0 + F}{F_0}\big) nT_r, \tag{2.26}$$

the slow-time axis is scaled such that the time interval between pulses, $(nT_r)'$, will be larger for frequencies greater than F_0 and less for frequencies smaller than F_0. This scaling results in a keystone shape of the scaled data, giving rise to the transform's name. For frequencies larger than F_0, the *sample-to-sample* phase progression is faster than desired. The rescaling of the axis doesn't change this; however, this rescaling does increase the *time between samples*, resulting in a *slow-time* phase progression that is now the same for all values of F. Substituting $(nT_r)'$ into (2.25), we have

$$z(F, (nT_r)') = exp(-j4\pi(F_0 + F)R_0/c) exp(j4\pi v_{rad} F_0 (nT_r)'/c) S_x(F). \tag{2.27}$$

It is clear that v_{rad} and F have been decoupled, resulting in the desired slow-time phase progression. When implemented, this rescaling requires a different interpolation of the slow-time axis for each frequency value. Specifically, a uniform slow-time axis is desired that samples the original data at a constant interval of T_r, requiring interpolation of the data at values of $\left(\frac{F_0}{F_0+F}\right)nT_r$. While this seems to directly conflict with the earlier choice or rescaling factor, interpolating the original data at these locations is exactly the same as sampling the Keystoned data at the original sample locations separated by T_r. Figure 2.3 shows the results of applying the Keystone transform to the same data set shown in Figure 2.2. It is clear that the Keystone transform is able to fully correct for linear range migration of targets.

When the Keystone transform is applied to SAR data, the motion profile of stationary scatterers within the scene will generally not be a simple linear equation. For a typical collection, bandwidth will be large enough and observation time long enough that significant range migration is present, and due to the collection geometry, higher order motion profiles will also be present, resulting in a scene that cannot be fully focused via the Keystone transform. As proposed in Chapter 1, the

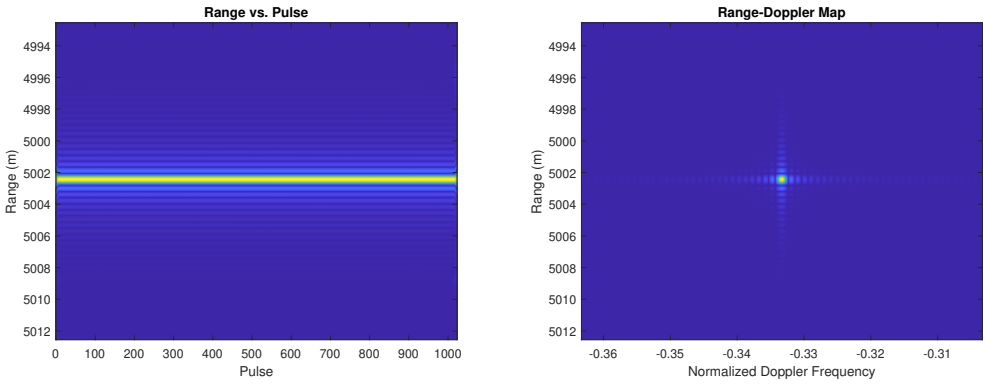


Figure 2.3: Range migration mitigation and range-Doppler focusing achieved by applying the Keystone transform to simulated data with parameters given in Table 2.1.

Keystone transform can be applied to sub-CPIs of the dataset. These sub-CPIs are chosen such that the range migration is approximately linear, allowing the Keystone transform to fully compensate for the migration. The following chapter will serve to fully describe 2D backprojection. The chapter will then go on to derive the necessary equations for implementation of this technique, including the Keystone transform which will be used to help focus individual sub-CPIs prior to aggregation over sub-CPIs into a final image.

Chapter 3

2D Range-Doppler Backprojection

Now that a general SAR signal model has been introduced, 2D backprojection can be derived. First, a general processing pipeline will be introduced for 2D backprojection. This pipeline will help to showcase the similarity to traditional backprojection, as well as help highlight the usefulness of 2D backprojection as a SAR-GMTI imaging algorithm. Next, 2D backprojection will be derived in full, including all relevant processing steps and resulting phase shifts needed to allow for coherent combination of sub-CPIs. Theses equations will serve as the foundation for the implementation of the algorithm in Chapters 4 and 5.

3.1 Processing Pipeline

The 2D backprojection will produce high resolution images of a full observation while also maintaining a processing architecture that can serve as a foundation for common GMTI techniques. The architecture of this algorithm is designed such that image formation by the 2D backprojection shares several processing steps with standard GMTI techniques, allowing for simultaneous implementation. The algorithm first divides the full CPI into several sub-CPIs, or subapertures, where range migration is approximately linear. Next, the Keystone transform is applied to ac-

count for any linear range migration within the sub-CPI. While range migration may be small within a sub-CPI, the Keystone transform will help to increase the focus of each sub-CPI. Mitigation of range migration is also essential for GMTI, as migration of moving targets will degrade detector performance. For this reason, the Keystone transform is a very useful step in a shared processing pipeline. Finally, inspired by the well-known backprojection algorithm, coherent integration across sub-CPIs will be performed to produce a full-resolution image of the desired scene.

For the side-looking linear flight geometry shown in Figure 2.1, a scatterer's range will change approximately quadratically over slow-time. While the difference in range to a scatterer between pulses may be small compared to the scene size or resolution of the system, the varying propagation phase at the carrier wavelength will introduce incoherency that inhibits proper integration of returns without proper compensation. This undesired phase must be accounted for in any coherent imaging algorithm. The traditional backprojection algorithm accounts for varying propagation phase by applying a range-dependent phase shift to the scatterer's returns on each pulse, thus allowing for coherent integration of returns over all pulses. The range-dependent phase shift applied by traditional backprojection is simply the conjugate of the undesired additional phase measured by the radar due to a scatterer's migration between pulses. To create an image of the full scene, a range interpolation and phase correction are required for every pixel within the scene. 2D backprojection aims to integrate returns in the same way as traditional backprojection but uses the scatterer's location in the range-Doppler map of each sub-CPI rather than the scatterer's range for each pulse.

In 2D Range-Doppler backprojection, a scatterer will have a range and Doppler value that changes during the course of the full CPI. The full CPI will be broken into sub-CPIs, and with the additional range migration correction from the Key-

stone transform, a scatterer will be localized in range and Doppler for each sub-CPI. While the differences from one sub-CPI to the next may be small, the range-Doppler migration results in a phase term that introduces incoherency and prevents proper integration of returns from all sub-CPIs. This incoherency is accounted for in 2D backprojection by applying both a range-dependent and Doppler-dependent phase shift to the scatterer's returns for each sub-CPI. Just as range variation requires a range interpolation and phase correction in the traditional backprojection approach, these range and Doppler variations require range and Doppler interpolations and phase corrections in order to coherently sum the returns over sequential range-Doppler maps. This procedure must be performed for every pixel within the scene to create an image.

Figure 3.1 shows the processing pipeline of traditional backprojection versus 2D backprojection. The image on the left of the figure shows the quadratically changing range of a stationary target across slow-time, consistent with the side-looking geometry used here. As described previously, the traditional backprojection algorithm interpolates the value of the range profile corresponding to a target's range for each pulse and applies a phase correction that allows for coherent integration of returns over the full CPI.

The right side of the figure shows the general procedure for SAR imaging via 2D backprojection. First, sub-CPIs are created and the Keystone transform is applied to each sub-CPI to account for linear range migration. Next, an FFT is performed across slow time to produce range-Doppler maps of each sub-CPI. Because a target's range and Doppler are varying during the full CPI, the target's location in the range-Doppler map will change from sub-CPI to sub-CPI, as shown in the figure. However, as long as a target's contributions are still localized within each range-Doppler map, their contributions can be obtained by 2D interpolation, analogous

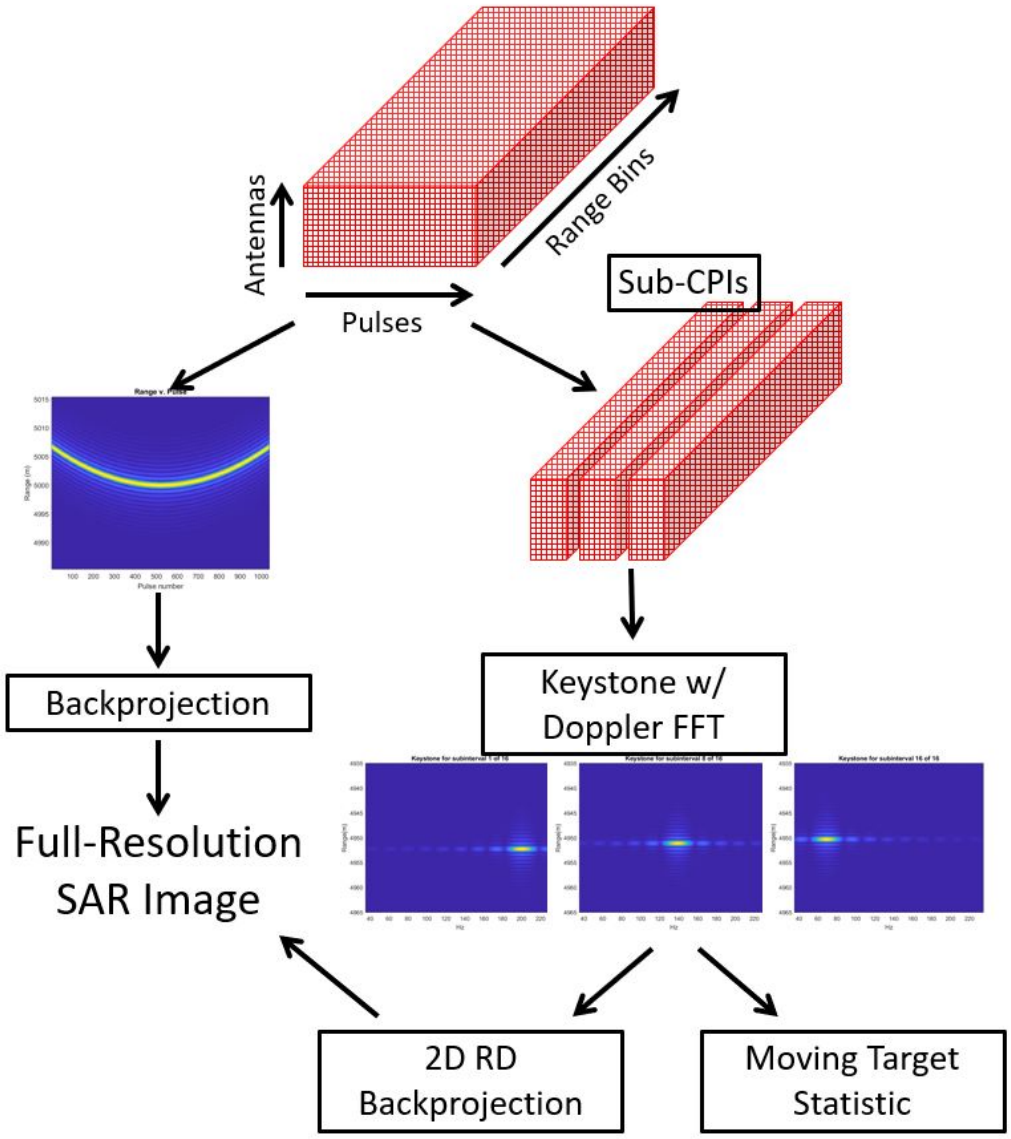


Figure 3.1: Imaging algorithm pipeline of 2D backprojection as compared to traditional backprojection.

to 1D interpolation of a range profile performed by traditional backprojection. In order to support coherent integration, we now need to apply a phase shift that fully accounts for the undesired phase shift resulting from the target's sub-CPI-dependent location in each range-Doppler map. The following section will derive the phase shift expected when the data is processed in this way; we can then apply the con-

jugate of this phase shift to the target's interpolated value, allowing for coherent integration of the target's contributions over the full CPI. Once in the Doppler domain and before imaging, we can also apply post-Doppler GMTI techniques to produce a moving target statistic, allowing for computationally efficient simultaneous imaging and target detection.

3.2 Imaging Signal Model

The received signal's Fourier transform, given in (2.11), is now split into several sub-CPIs in preparation for linear range migration compensation by performing the Keystone transform on each of the sub-CPIs. We now define the pulse index, n, in terms of a sub-CPI center pulse, m', and a local pulse index, m, giving

$$n = m' + m, \tag{3.1}$$

where $-(M-1)/2 \leq m \leq (M-1)/2$ and M is the number of pulses within a sub-CPI. Indexing the data in this way results in (2.11) now being a function of fast-time frequency, F, a sub-CPI center pulse, and a local index centered at the sub-CPI center. Performing the Keystone transform and Doppler FFT on this local pulse index allows us to perform them on each individual sub-CPI defined by center pulse m'. This indexing scheme also allows us to isolate terms that are dependent on the sub-CPI center pulse, i.e. the terms that change from one sub-CPI to the next. These terms must be accounted for to support coherent combination of sub-CPIs. To find the pulse-dependent propagation delay based on the new indexing scheme,

we will substitute (3.1) into (2.21), giving

$$\tau_m = \frac{2}{c}\Bigg[R_0 - \frac{v_r y_0}{R_0}(m'+m)T_r - \frac{1}{2}\frac{v_r^2 y_0^2}{R_0^3}((m'+m)T_r)^2 + \frac{1}{2}\frac{v_r^2}{R_0}((m'+m)T_r)^2\Bigg]. \tag{3.2}$$

Substituting (3.2) into (2.11) to find $z(F, m', m)$, we have

$$\begin{aligned} z(F,m',m) = S_x(F) exp\Bigg(- j2\pi(F_0+F)\frac{2}{c}R_0\Bigg) \times \\ exp\Bigg(j2\pi(F_0+F)\frac{2}{c}\frac{v_r y_0}{R_0}(m'+m)T_r\Bigg) \times \\ exp\Bigg(j2\pi(F_0+F)\frac{2}{c}\frac{1}{2}\frac{v_r^2 y_0^2}{R_0^3}((m'+m)T_r)^2\Bigg) \times \\ exp\Bigg(- j2\pi(F_0+F)\frac{2}{c}\frac{1}{2}\frac{v_r^2}{R_0}((m'+m)T_r)^2\Bigg). \end{aligned} \tag{3.3}$$

This equation can now be rearranged into m-independent terms and m-dependent terms. Squaring the terms as necessary within the arguments of the various exponentials, the m-independent terms include

$$\Psi_1(F) = exp\big(- j4\pi(F_0+F)\frac{R_0}{c}\big), \tag{3.4}$$

$$\Psi_2(F, m') = exp\big(j4\pi(F_0+F)\frac{v_r y_0}{R_0 c}m'T_r\big), \tag{3.5}$$

and

$$\Psi_3(F, m') = exp\Bigg(j2\pi(F_0+F)\big(\frac{v_r^2 y_0^2}{R_0^3 c} - \frac{v_r^2}{R_0 c}\big)(m')^2 T_r^2\Bigg). \tag{3.6}$$

These terms are independent of the slow-time index, m, that will be operated on by the Keystone transform. Therefore, m' will have no effect on the output of the Keystone transform. For this reason, we will now combine Ψ_1, Ψ_2, and Ψ_3 into one

exponential term, given by

$$\Psi(F,m') = exp\bigg(-j\frac{4\pi}{c}(F_0+F)\Big(R_0-\frac{v_r y_0}{R_0}m'T_r-\frac{v_r^2 y_0^2}{R_0^3}(m'T_r)^2-\frac{v_r^2}{R_0}(m'T_r)^2\Big)\bigg). \tag{3.7}$$

We can now rewrite equation (3.3) in terms of Ψ and m-dependent terms, resulting in

$$\begin{aligned} z(F,m',m) = S_x(F)\Psi(F,m')exp\bigg(j2\pi(F_0+F)\frac{2}{c}\frac{v_r y_0}{R_0}mT_r\bigg)\times \\ exp\bigg(j2\pi(F_0+F)\frac{2}{c}\frac{1}{2}\frac{v_r^2 y_0^2}{R_0^3}(2m'm+m^2)T_r^2\bigg)\times \\ exp\bigg(-j2\pi(F_0+F)\frac{2}{c}\frac{1}{2}\frac{v_r^2}{R_0}(2m'm+m^2)T_r^2\bigg). \end{aligned} \tag{3.8}$$

We notice that the sample-to-sample phase rotation is dependent on fast-time frequency, F, and v_r, exactly as shown when deriving the Keystone transform. Using the Keystone transform to uncouple these terms, the sample-to-sample phase rotation can be made to be a function of only v_r. Once decoupled, the linear range migration will be accounted for, fully focusing the range-Doppler map of the sub-CPI.

We now perform the Keystone transform by transforming the data in slow time to the new keystone-shaped grid of points. This step equates to rescaling the data to modified pulse index locations of $\hat{m} = m\frac{(F_0+F)}{F_0}$ for every value of fast-time frequency. This substitution results in

$$\begin{aligned} z(F,m',\hat{m}) = S_x(F)\Psi(F,m')exp\bigg(j2\pi(F_0+F)\frac{2}{c}\frac{v_r y_0}{R_0}(\frac{F_0}{F_0+F}\hat{m})T_r\bigg)\times \\ exp\bigg(j2\pi(F_0+F)\frac{2}{c}\frac{1}{2}\frac{v_r^2 y_0^2}{R_0^3}(2m'(\frac{F_0}{F_0+F}\hat{m})+(\frac{F_0}{F_0+F}\hat{m})^2)T_r^2\bigg)\times \\ exp\bigg(-j2\pi(F_0+F)\frac{2}{c}\frac{1}{2}\frac{v_r^2}{R_0}(2m'(\frac{F_0}{F_0+F}\hat{m})+(\frac{F_0}{F_0+F}\hat{m})^2)T_r^2\bigg). \end{aligned} \tag{3.9}$$

This equation can be rearranged to a more convenient form, given by

$$\begin{aligned} z(F, m', \hat{m}) = S_x(F)\Psi(F, m')\times \\ exp\left(j2\pi \frac{2F_0}{c}\left(\frac{v_r y_0}{R_0} + \left(\frac{v_r^2 y_0^2}{R_0^3} - \frac{v_r^2}{R_0}\right)m'T_r\right)\hat{m}T_r\right)\times \\ exp\left(j\pi\frac{2F_0}{c}\left(\frac{v_r^2 y_0^2}{R_0^3} - \frac{v_r^2}{R_0}\right)\frac{F_0}{F_0 + F}(\hat{m}T_r)^2\right). \end{aligned} \tag{3.10}$$

After rearranging, we can see that the first exponential contains the Doppler shift of the pixel, given by (2.22), evaluated at the center of the sub-CPI, i.e., the Doppler shift of the pixel at $nT_r = m'T_r$, which can be denoted $F_d(m'T_r)$. Additionally, this exponential does not depend on F, meaning that the Keystone has successfully decoupled radial velocity and F. This exponential is the ideal slow-time response desired for a well-behaved Doppler response, yielding a sinc function when converted to the Doppler domain.

The second exponential contains the time derivative of the Doppler frequency given in (2.22). The rate of change of Doppler is constant, a result of limiting the Taylor expansion to only include second-order terms. Fundamentally, this additional exponential results from the fact that Doppler frequency is not constant throughout a sub-CPI, resulting from some non-trivial acceleration. This acceleration, while small, results in a slightly nonlinear motion profile. The Keystone transform is unable to compensate for this term; however, it has been assumed that the sub-CPIs are adequately short that target migration can be approximated as linear. Additionally, this term is independent of m' and does not affect our ability to coherently sum subsequent sub-CPIs; therefore, the affects of this term can be assumed to be negligible. Later, we will discuss the implications of violating this assumption.

We now wish to convert each sub-CPI into the Doppler domain by taking the

DTFT over the new, *local* slow-time pulse index, $\hat{m}$. This step is described by

$$Z(F, m', f_d) = \sum_{\hat{m}=0}^{\hat{m}=M-1} z(F, m', \hat{m} - \frac{M-1}{2}) exp(-j2\pi f_d \hat{m}) \tag{3.11}$$

where the data index of $z(F, m', \hat{m})$ must be shifted by $-(M-1)/2$ because the DTFT is typically defined with a starting index of zero, but the data index of each sub-CPI has a starting index of $-(M-1)/2$. This shift in the index results in the traditionally-indexed DTFT multiplied by a phase shift, which can be shown via a change of variables in the DTFT expression. Converting the shifted data to a phase shift, the result is now

$$Z(F, m', f_d) = exp\bigg(- j2\pi f_d \bigg(\frac{M-1}{2} \bigg) T_r \bigg) \sum_{\hat{m}=-\frac{M-1}{2}}^{\hat{m}=\frac{M-1}{2}} z(F, m', \hat{m}) exp(-j2\pi f_d \hat{m}). \tag{3.12}$$

Plugging the expression for $z(F, m', \hat{m})$ from (3.10) into this expression, we have

$$\begin{aligned} Z(F, m', f_d) = S_x(F)\Psi(F, m') exp\bigg(- j2\pi f_d \bigg(\frac{M-1}{2} \bigg) T_r \bigg) \times \\ \sum_{\hat{m}=-\frac{M-1}{2}}^{\hat{m}=\frac{M-1}{2}} exp\bigg(j2\pi F_d (m' T_r) \hat{m} T_r \bigg) exp\bigg(- j2\pi f_d \hat{m} \bigg) \times \\ exp\bigg(j\pi \frac{2F_0}{c} \bigg(\frac{v_r^2 y_0^2}{R_0^3} - \frac{v_r^2}{R_0} \bigg) \frac{F_0}{F_0 + F} (\hat{m} T_r)^2 \bigg). \end{aligned} \tag{3.13}$$

As discussed, the last exponential in the summation contributes a small phase rotation that is independent of sub-CPI center pulse, m', so it will not affect our ability to coherently sum over all sub-CPIs; therefore, this term will be ignored for now. We now take the DTFT of the remaining exponential, which will be a "digital sinc function" in the Doppler domain centered at normalized Doppler shift, $T_r F_d(m' T_r)$,

and denoted $D_x(f_d - T_r F_d(m'T_r))$. The phase term outside of the summation contributes an extra phase rotation that depends on sub-CPI via the scatterer's Doppler shift evaluated at m'. This term is the Doppler-dependent phase shift that needs to be accounted for on each sub-CPI. Substituting $D_x(f_d - T_r F_d(m'T_r))$ into (3.13) we have

$$\begin{aligned} Z(F, m', F_d(m'T_r)T_r) = S_x(F) D_x(f_d - T_r F_d(m'T_r)) \times \\ \Psi(F, m') exp(-j2\pi F_d(m'T_r)\left(\frac{M-1}{2}\right)T_r). \end{aligned} \tag{3.14}$$

Finally, we can expand Ψ and take the inverse Fourier transform over fast-time frequency of the data, and we are left with data that is ready for interpolation and aggregation, given by

$$\begin{aligned} Z(t, m', f_d) = R_{xx}\big(t - \frac{2r(m'T_r)}{c}\big) D_x(f_d - T_r F_d(m'T_r)) \times \\ exp\left(- j\pi F_d(m'T_r)(M-1)T_r \right) exp\left(- j2\pi F_0 \frac{2r(m'T_r)}{c} \right). \end{aligned} \tag{3.15}$$

To perform 2D RD backprojection, we will calculate the range and Doppler locations of all pixels in the scene at the center time instant of each sub-CPI. Next, we interpolate the sub-CPI's range-Doppler map at those particular values. Range-Doppler interpolation gives a pixel's contribution to the range-Doppler map for each sub-CPI, which then must be aligned in phase for coherent integration over multiple sub-CPIs. Therefore, we now apply a phase correction given by the conjugate of the two exponential terms shown in (3.15), one term for the change in range from each sub-CPI to the next and one term for the change in Doppler. This is performed for each sub-CPI and then returns are integrated over sequential sub-CPIs to produce an image. Using these derived equations, Chapter 4 will present results obtained by performing 2D backprojection. These results will be compared to traditional

backprojection to explore some of the benefits and trade-offs of this algorithm.

Chapter 4

Results

The previous Chapter showed that a target with a specific location in the range-Doppler map of a sub-CPI will have a location-dependent phase shift. This location changes from one sub-CPI to the next and must be accounted for in order for range-Doppler maps to be coherently combined into a full resolution SAR image. To create an image of a distributed scene, range and Doppler values are calculated for all pixels within the scene. Range-Doppler maps are then interpolated at these locations and the corresponding phase shifts are applied to each pixel. Returns from subsequent sub-CPIs are then combined, forming a high quality image. This Chapter will show results of performing 2D backprojection on simulated and real data. Image quality will be discussed, as well as artifacts of 2D backprojection. A brief demonstration of this technique's ability to perform moving target imaging will also be shown at the end of the Chapter.

4.1 Simulated Results

4.1.1 Simulation Architecture

Data were simulated efficiently using an idealized linear flight geometry that allowed for calculation of the returns produced by the superposition of many scatterers in a scene of interest. The radar's location on each pulse is set to change in the along-track dimension by an integer number of scatterer positions, allowing for the reuse of previously computed range profiles. The recycling of ranges to scatterers in the scene provides significant computational savings and allows for efficient production of distributed scenes. The simulation also saves computation time by pre-computing the pulse compression profiles of the waveform. Typically the received signal is passed through the matched filter, but for simulation purposes, the autocorrelation of the waveform can be used as the ideal response of the system. Therefore, passing the transmitted waveform through the matched filter is equivalent to calculating the autocorrelation function of the waveform to pre-compute pulse compression profiles. These pre-computed profiles are sampled very finely to mitigate the effects of target straddling range bins, then shifted and centered at a target's location in the range-profile.

Typically, pulse compression is performed after the received waveform is simulated for a target, resulting in a measured phase-shift that depends on the two-way propagation delay of the signal. However, here the pulse compression profile is pre-computed, requiring a phase shift to be manually applied to the returns from the target. This procedure for simulation of data is very efficient and accurately replicates real data. While efficient, this approach breaks down if the waveform is not chosen such that it is Doppler tolerant. If the waveform is not Doppler tolerant, Doppler shift exhibited by a target can deform and shift the matched filter output,

an effect that is not present in the pre-computed pulse compression profiles. While a Doppler tolerant waveform is always desired, the method used here relies on the ability to assume Doppler tolerance.

4.1.2 Images of Simulated Data

A distributed scene was simulated with radar parameters given in Table 4.1 and imaging was performed using 2D RD backprojection, as well as traditional backprojection for a baseline comparison. Figure 4.1 shows the results of the two methods performed on a full scene. The two images are almost identical, and the same artifacts can be observed in both. Figure 4.2 shows a closeup view of a point scatterer like the one located at the center of the scene in Figure 4.1. This point response represents the effective imaging function of the algorithm, and again, the two are almost identical. However, the imaging function of the 2D backprojection is dependent on several factors that can improve or diminish the output of the algorithm. If the underlying operation parameters, such as bandwidth or observation time, are changed, the imaging functions of both algorithms are affected accordingly; however, because of the extra steps taken and choices made when performing 2D backprojection, its imaging function is much more reliant on parameter choices.

One of the many benefits of performing imaging via 2D backprojection is that

Table 4.1: Radar parameters used for simulating data.

Parameter	Value
f_c	10 GHz
β	200 MHz
PRI	1 ms
N	1040
R_0	5 km
v_r	125 m/s

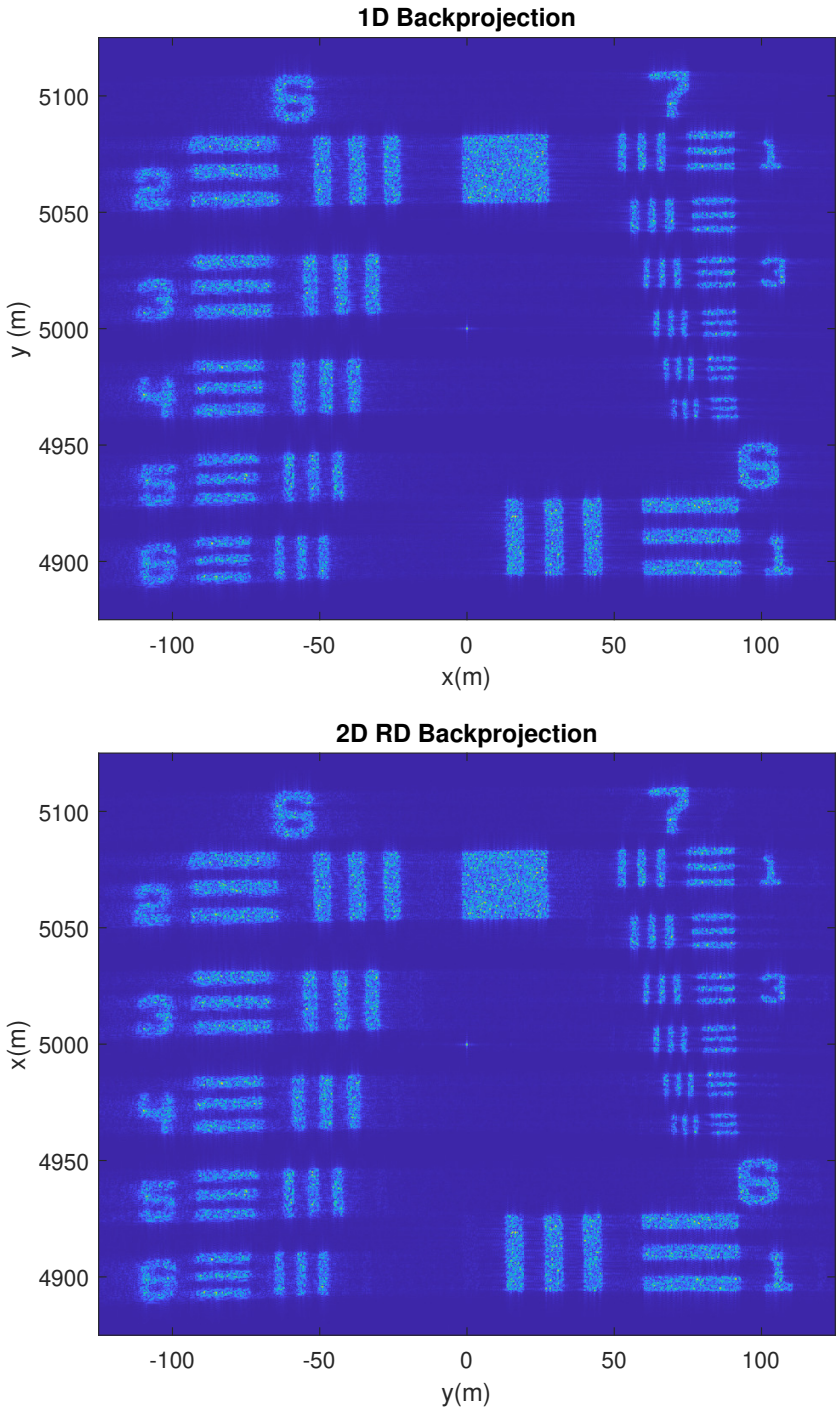


Figure 4.1: Traditional backprojection compared to 2D backprojection on the same dataset. 1,040 pulses were used for each, and 2D backprojection partitioned the CPI into 26 sub-CPIs with 40 pulses in each.

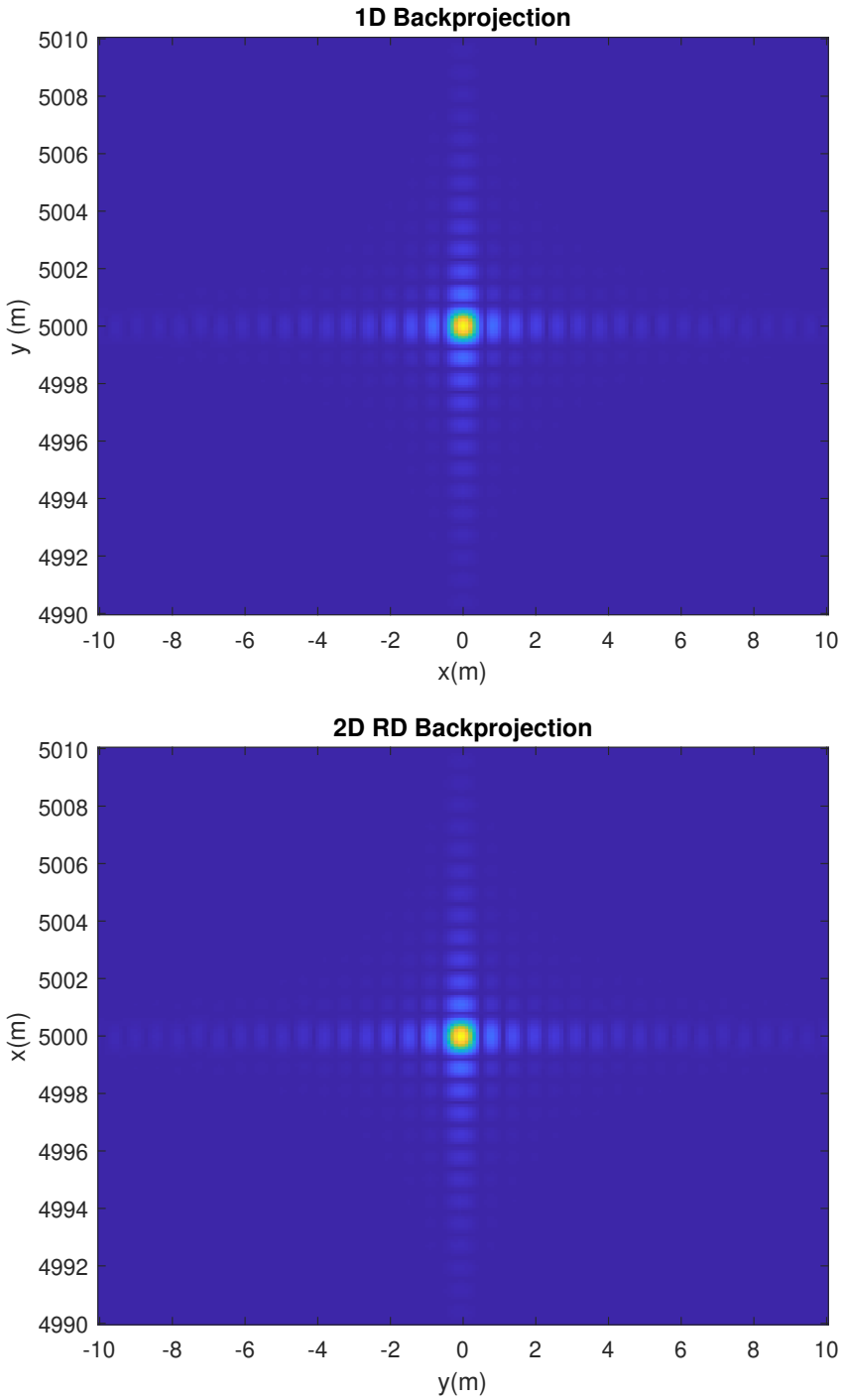
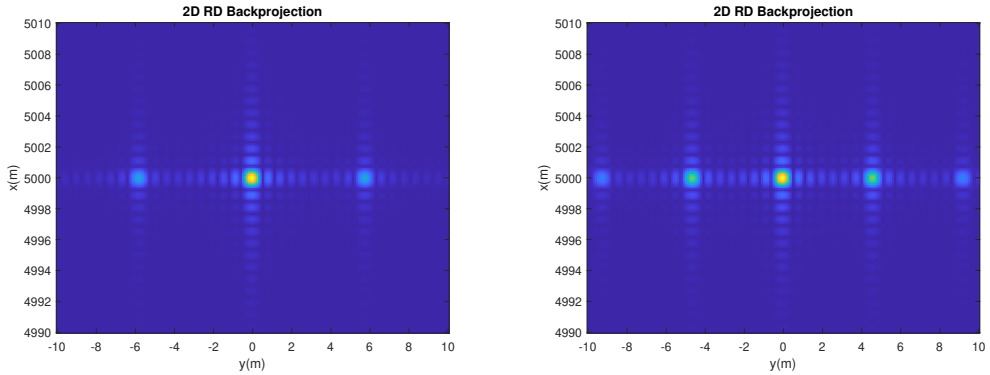


Figure 4.2: Traditional and 2D backprojection images of a point scatterer like the one at the center of Figure 4.1.

many assumptions required for other fast imaging algorithms are not required. Nevertheless, many parameters within the algorithm, such as sub-CPI length, i.e., the time duration of the sub-CPI, have a non-trivial effect on the performance of the algorithm. To reduce computation, the full CPI should be divided into as few sub-CPIs as possible, reducing the number of interpolations and phase corrections needed to form the final image. However, this results in long individual sub-CPIs, and as sub-CPI length is increased, the linear range migration approximation assumed for each sub-CPI no longer accurately models target motion. The second exponential in the summation of (3.3), once assumed to have very little effect on the focus of the sub-CPI's range-Doppler map, now can have a significant impact on image quality. This exponential is quadratic in $\hat{m}$, so it can very quickly degrade algorithm performance. The increased effect of this exponential results in poor focusing by the Keystone transform; therefore, individual range-Doppler maps are blurred, and the final image is blurred as well, implying an upper limit on sub-CPI length.

Figure 4.3 shows the imaging function for sub-CPIs that are longer than the



(a) (b)

Figure 4.3: Image formed by performing 2D backprojection on **(a)** 10 sub-CPIs of length 104 pulses and **(b)** 8 sub-CPIs of length 130 pulses.

sub-CPI length used in Figure 4.2. On the left, 10 sub-CPIs each containing 104 pulses were used to image the point target at the center of the scene. On the right, 8 sub-CPIs each containing 130 pulses were used to image the same point target. It is immediately clear that there are very high side lobes in cross-range that were not present when a shorter sub-CPI was used. These side lobes act much like grating lobes in a phased array, i.e. as the sub-CPI length is increased, the aliased copies of the target become more closely spaced and additional adjacent copies become visible as their peaks become stronger. Clearly, these aliasing artifacts are dependent on the number of pulses within the sub-CPI and, therefore, the spacing between the sub-CPI centers.

Figure 4.3 shows that as the sub-CPI length is increased, these artifacts appear closer to the true target at the center. This behavior is consistent with the behavior of grating lobes in a phased array. The best practice to prevent grating lobes in a phased array is to ensure that the element separation is smaller than $\lambda/2$. However, a $\lambda/2$ spacing is extremely small; if sub-CPIs were limited such that their subsequent centers were separated by $\lambda/2$, 2D backprojection would no longer be a useful technique for imaging. Therefore, sub-CPIs will need to be spaced by substantially more than $\lambda/2$. However, like grating lobes, these aliased copies move closer to the true target as the length of the sub-CPIs is increased. The aliased copies are weaker the further away from the true target they are; the simple solution to this problem is to limit the sub-CPI length such that aliasing is minimal, i.e. such that aliasing does not degrade image quality. This implies an upper limit to the length of sub-CPI that can be used in 2D backprojection.

Figure 4.4 shows cuts from range and cross-range of the imaging function for various sub-CPI lengths. The range cuts show that the imaging function in this dimension is invariant of sub-CPI length and identical to traditional backprojection.

Intuitively, this makes sense as the resolution in range is determined by the bandwidth whereas the the resolution in cross-range is achieved by processing of the synthetic aperture. For this reason, the 2D backprojection exhibits cross-range cuts that rely on processing technique and choice of parameters. The cross-range behavior of the imaging function, found in earlier figures, is confirmed here; as sub-CPI length is increased, aliased copies of the target act much like grating lobes by decreasing their spacing. Again, we see that the artifacts are stronger and approach the target in the center as sub-CPI length is increased. For sub-CPI lengths that show multiple sets of artifacts, we see that each subsequent artifact is weaker than the previous. Additionally, the peak of each subsequent grating lobe rises above the backprojection side lobe level (SLL) less than the previous, i.e., while the first

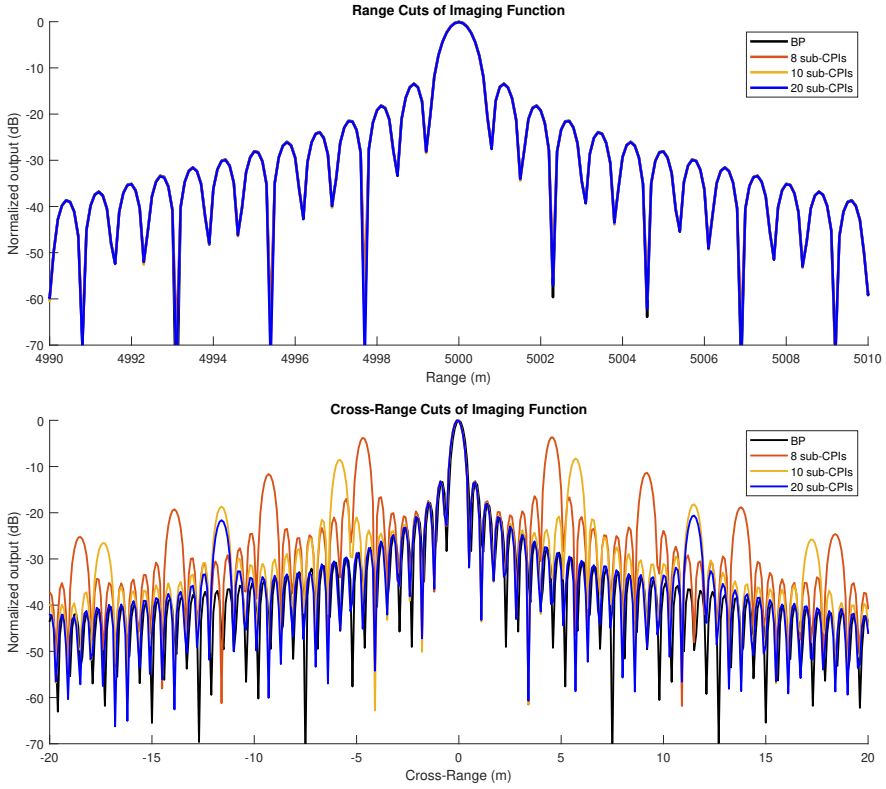


Figure 4.4: Cuts of imaging function in range and cross-range for various sub-CPI lengths.

grating lobe in the eight sub-CPI cut rises 25 dB above the backprojection SLL, the grating lobe at the leftmost extent of the figure for the same sub-CPI length rises only about 17 dB above the backprojection SLL. Far from the main lobe, this grating lobe is more than 25 dB weaker than the main lobe level; however, near the main lobe in cross-range, this grating lobe is slightly less than 4 dB weaker than the main lobe level. This response is detrimental to image quality; using long sub-CPIs in this fashion would only be suitable for extremely limited scene extent in along-track. Obviously this limitation is undesired as the along-track scene extent would be limited to less than 10 meters if the use of 8 sub-CPIs was desired. For this reason, shorter sub-CPIs are favorable; while the use of shorter sub-CPIs increases computation as more interpolations are needed to form an image, the increased imaging performance will often be much more desirable than saving computation. More details will be given in chapter 5 regarding computational requirements and savings in using 2D backprojection with various sub-CPI lengths.

Shorter sub-CPIs also exhibit weaker grating lobes than longer sub-CPIs grating lobes that fall at the same cross-range location. The first grating lobe of the 20 sub-CPI imaging function shown in Figure 4.4 coincides with the second grating lobe of the 10 sub-CPI imaging function; however, the 20 sub-CPI imaging function's grating lobe is 3 dB below that of the 10 sub-CPI imaging function. Ideally, sub-CPI length could be chosen to prevent grating lobes all together, but this may not be necessary if the grating lobe level falls below some desired level. Preventing grating lobes also requires the ability to predict where they will appear.

Figure 4.4 shows that the grating lobes appear to be well-behaved. The grating lobes of a physical aperture array containing N elements spaced by d, where N is chosen to be the same as the number of sub-CPIs and d is chosen to be twice the spacing between the center of the sub-CPIs, are perfectly matched with the locations

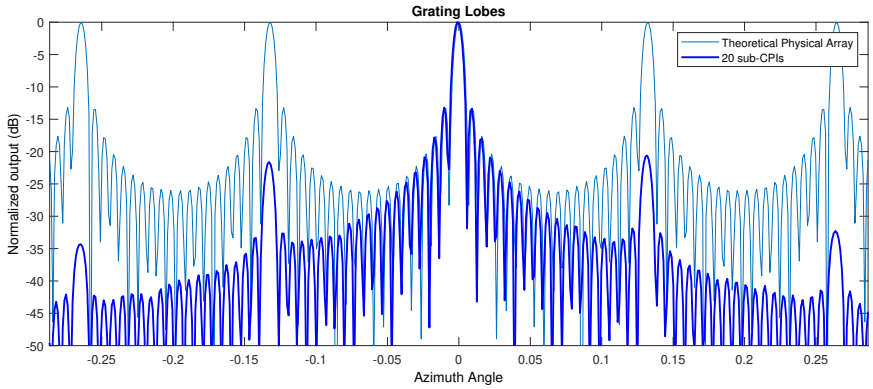


Figure 4.5: Measured grating lobes in imaging function of dataset broken into 20 sub-CPIs compared to the theoretical physical array that is emulated by the synthetic array.

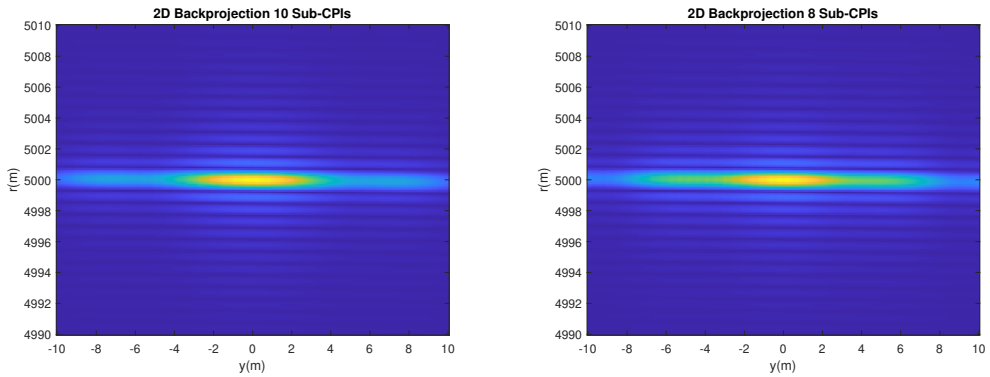
of the artifacts seen in the imaging function. The element spacing of twice the sub-CPI spacing arises from the fact that the synthetic array synthesizes a physical array twice the length of the synthetic aperture.

Figure 4.5 shows this theoretical array and how well it matches the locations of the artifacts of the 20 sub-CPI imaging function. The discontinuity between sub-CPI centers is on the order of the sub-CPI length, resulting in the artifacts coinciding with the location of grating lobes of a theoretical array. This means that these grating lobes are inherent to the 2D backprojection, but based on array processing, their behavior is predictable. Sub-CPI length can be chosen such that the first grating lobe fall outside the cross-range extent of the desired scene, or an acceptable grating lobe level can be defined to determine sub-CPI length.

Instead of observing the imaging function of all sub-CPIs integrated together, we can look at the imaging function of a single sub-CPI to isolate the effects of poor Doppler resolution We see in Figure 4.6 that the imaging functions exhibit interesting characteristics. We expect that the cross-range resolution of a single sub-CPI is worse than the full observation as the observation time is significantly shorter, and

both imaging functions in Figure 4.6 are consistent with this expectation, i.e. the cross-range resolution of the single sub-CPI is significantly worse than integrating all sub-CPIs together. We would also expect that the sub-CPI with 130 pulses would exhibit slightly better cross-range resolution than the sub-CPI with 104 pulses as a larger subaperture was used to create the image; however, the longer sub-CPI results in an imaging function with significant smearing in cross-range. This smearing is a result of interpolating the pixels from the poorly focused range-Doppler map of the long sub-CPI. In general, longer sub-CPIs will exhibit better resolution, but as sub-CPI length is increased, the approximation of linear range migration breaks down. This results in poorly focused range-Doppler maps that, once combined, decrease final image quality. While the resolution obtained from a single sub-CPI of any length may be coarse, the full CPI will be used to form the final full resolution image, allowing the freedom to choose sub-CPI length. For this reason, it is often more favorable to choose a short sub-CPI such that range-Doppler maps are focused and grating lobes are mitigated.

The following section will show results of applying 2D backprojection to real



(a) (b)

Figure 4.6: Effective imaging function of a single sub-CPI for sub-CPIs of length **(a)**104 and **(b)**130 pulses.

data. High-quality Doppler interpolation is required to ensure that aliasing artifacts are not worsened by poor interpolation quality. This oversampling of the Doppler axis helps to smooth out some of the discontinuity between sub-CPIs, allowing the use of longer sub-CPIs at the cost of increased computation. Nevertheless, the 2D backprojection is able to produce extremely high resolution images with a drastic reduction in computation.

4.2 Images of Large Scene Gotcha Dataset

Application to real data introduces several additional steps to account for the imperfections associated with collecting data in a nonideal environment. As with any SAR imaging algorithm, exact knowledge of the platform location is required for proper imaging of the scene, and any deviation from the ideal flight path results in imperfections and defocusing. One of the benefits of traditional backprojection is the ability to compensate for motion errors on a pulse-by-pulse basis. The platform location is used on every pulse to calculate the range to every pixel and correct for phase errors in traditional backprojection; however, 2D backprojection uses the platform location and velocity at the center time instant of the sub-CPI to calculate range and extract Doppler information for phase corrections.

In simulated data, the pulse at the center of the sub-CPI occurs exactly at the location corresponding to the center time instant of the sub-CPI. However in real data, errors in IMU measurements and deviation from a linear flight path can result in the center pulse of the sub-CPI being misrepresentative of the center time instant of the sub-CPI. Additionally, individual measurements can be more or less erroneous; for this reason, it is best to use an approximation technique for determining platform location and velocity. The ideal geometry also assumes the velocity and PRI are

constant. While the PRI is typically constant in real data, the velocity vector will change over the course of a single sub-CPI, further necessitating approximation.

Traditional backprojection and 2D backprojection were implemented on the Air Force Research Labratory (AFRL) Large Scene Gotcha Data for testing of the algorithms' capabilities. This dataset was collected at X-band with 600 MHz of bandwidth, yielding a range resolution of roughly 0.25 m. The platform was roughly 10 kilometers from the scene center and collected range samples for a range swath of just over 5 kilometers. Thirty thousand slow-time samples were collected, and the platform location was recorded for each pulse of data. Figure 4.7 shows a section of the full scene as imaged by traditional backprojection using all 30,000 pulses. Figure 4.8 shows the same section as imaged by 2D range-Doppler backprojection using 966 sub-CPIs each containing 31 pulses. The total number of pulses used amounts to slightly less than 30,000, a result of freely choosing sub-CPI length. By comparison we see that the images are essentially identical. With the current choice of sub-CPI length and scene size, we see that grating lobes have no effect on image quality. Previously in Figure 4.4, we saw that the SLL of traditional backprojection and 2D backprojection were essentially identical, provided grating lobes were not present. Simulated data showed that ideal geometry resulted in an imaging function that is almost identical for both algorithms, and here, real data confirms that identical image resolution is possible. Even with the inability to account for motion errors on a pulse-by-pulse basis, 2D backprojection is able to fully focus the image in the same way that traditional backprojection can.

Figure 4.9 shows a smaller section of the image for both methods. Again, we see that image quality is identical; the large scale features aligned perfectly in Figure 4.7 and 4.8, and here, the fine scale features match for both algorithms. Additionally, the value of individual pixels are almost identical for each of the methods.

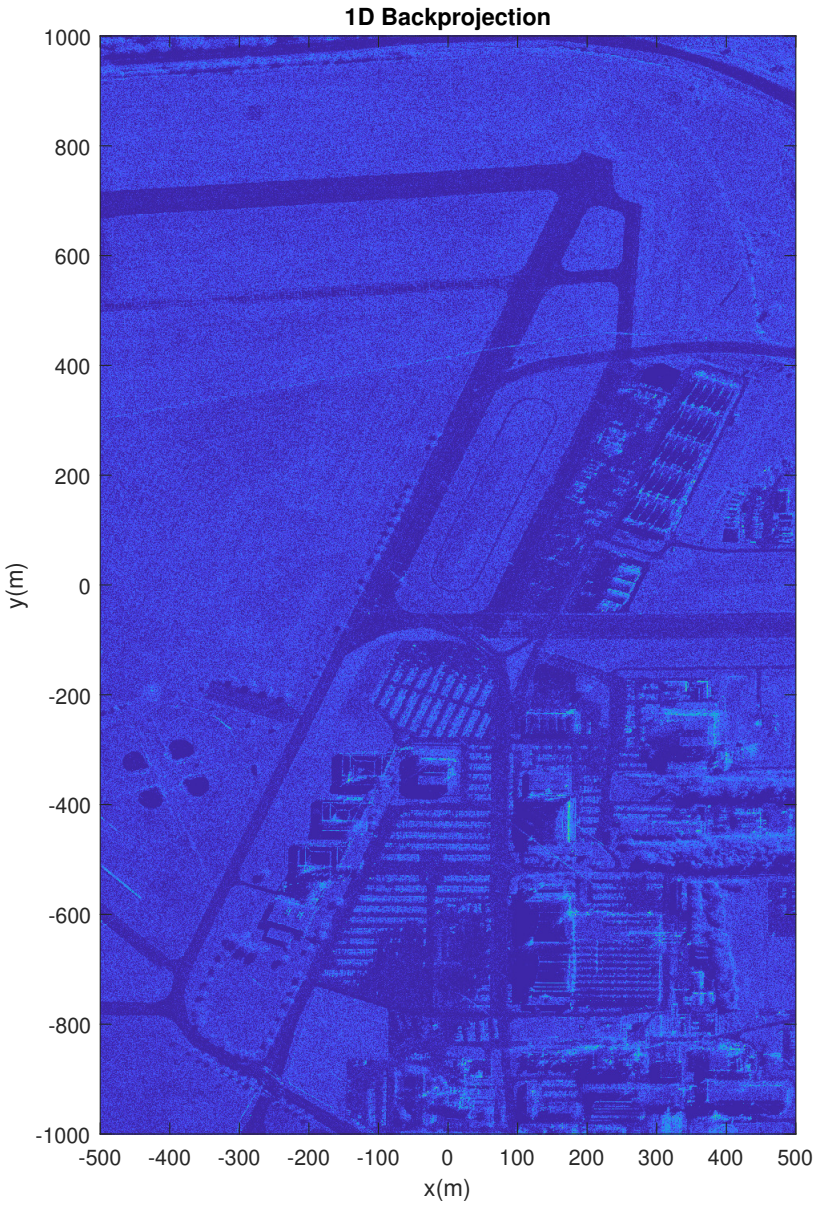


Figure 4.7: Traditional backprojection performed on SAR large scene Gotcha data. All 30,000 pulses were used to create the image. Windowing was performed across fast-time and slow-time to reduce side lobe output

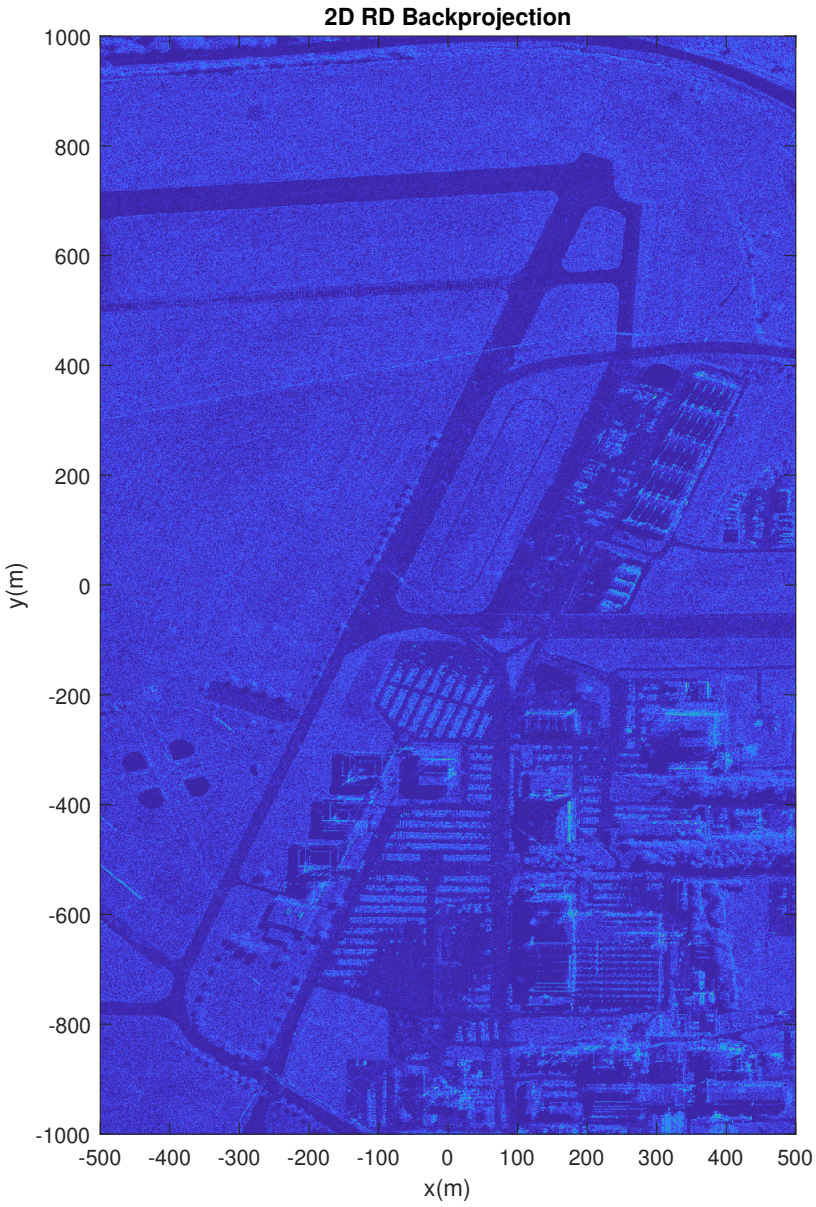


Figure 4.8: 2D backprojection performed SAR large scene Gotcha data. CPI was broken into 966 sub-CPIs containing 31 pulses each. 29,946 pulses were used in total. Windowing was performed across fast-time and sub-CPIs to reduce side lobe output.

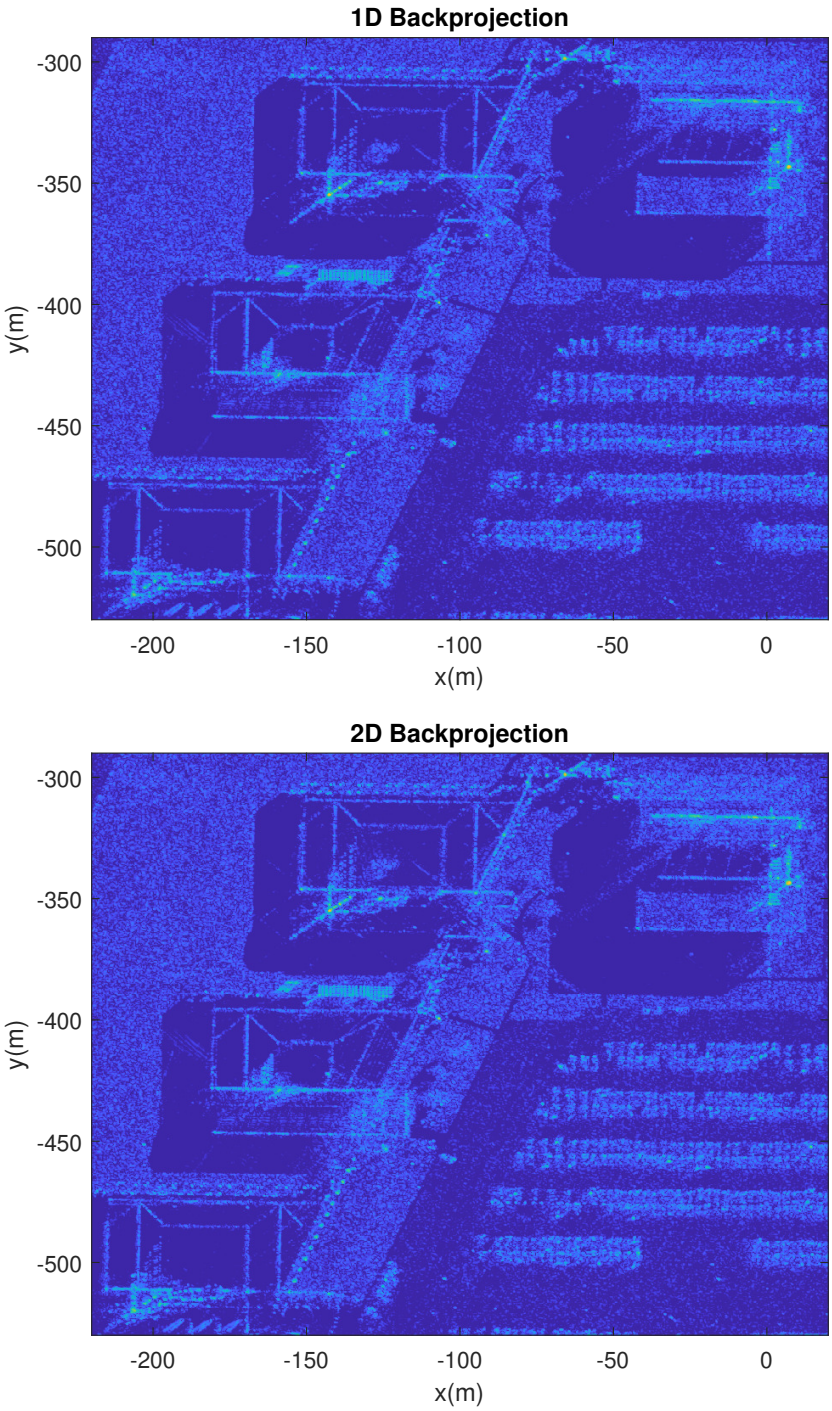


Figure 4.9: Comparison of traditional backprojection and 2D backprojection performed on the Gotcha Dataset.

The corner of the building in the upper left of both images in Figure 4.9 has a bright point that can be used to determine the difference in pixel intensity for both. The intensity of the brightest pixel at this corner differs by less than a tenth of a dB between each image. This means the processing gain achieved by both algorithms is roughly identical; intuitively, this makes sense as we are still using the full CPI to coherently sum returns, but it shows that we are not losing any coherence due to unaccounted motion errors.

Another more meaningful comparison of the two images shown in figures 4.7 and 4.8 is the formation of a difference image. By taking the difference of the two images, then converting to a percent difference, the performance of 2D backprojection can be further analyzed. Figure 4.10 shows the percent difference between the two images. It is important to note that the color axis has been limited to range from 0% to 200%. It is clear that a majority of the image lies in this range, but there are several areas within the image that are very bright and even saturate this color axis.

The general structure of the image is clear, however, the bright spots in the difference image lie in the regions of the scene that are very dark in figures 4.7 and 4.8. Since the values of pixels in these dim regions are relatively small, even small differences between the two images result in a relatively large percent difference. There appears to be a slight reduction in contrast that results from the use of 2D backprojection. Additionally, some of the artifacts that were observed in simulated data become visible for this dataset. Specifically, the dense concentration of bright pixels in the bottom right of the image is likely one of these artifacts resulting from a bright scatterer. In general, the percent differences are very small, however, large percent errors between the two images show that adding another dimension of interpolation has a nontrivial effect on the image. Nevertheless, overall image quality of 2D backprojection is still very similar to that of traditional backprojection.

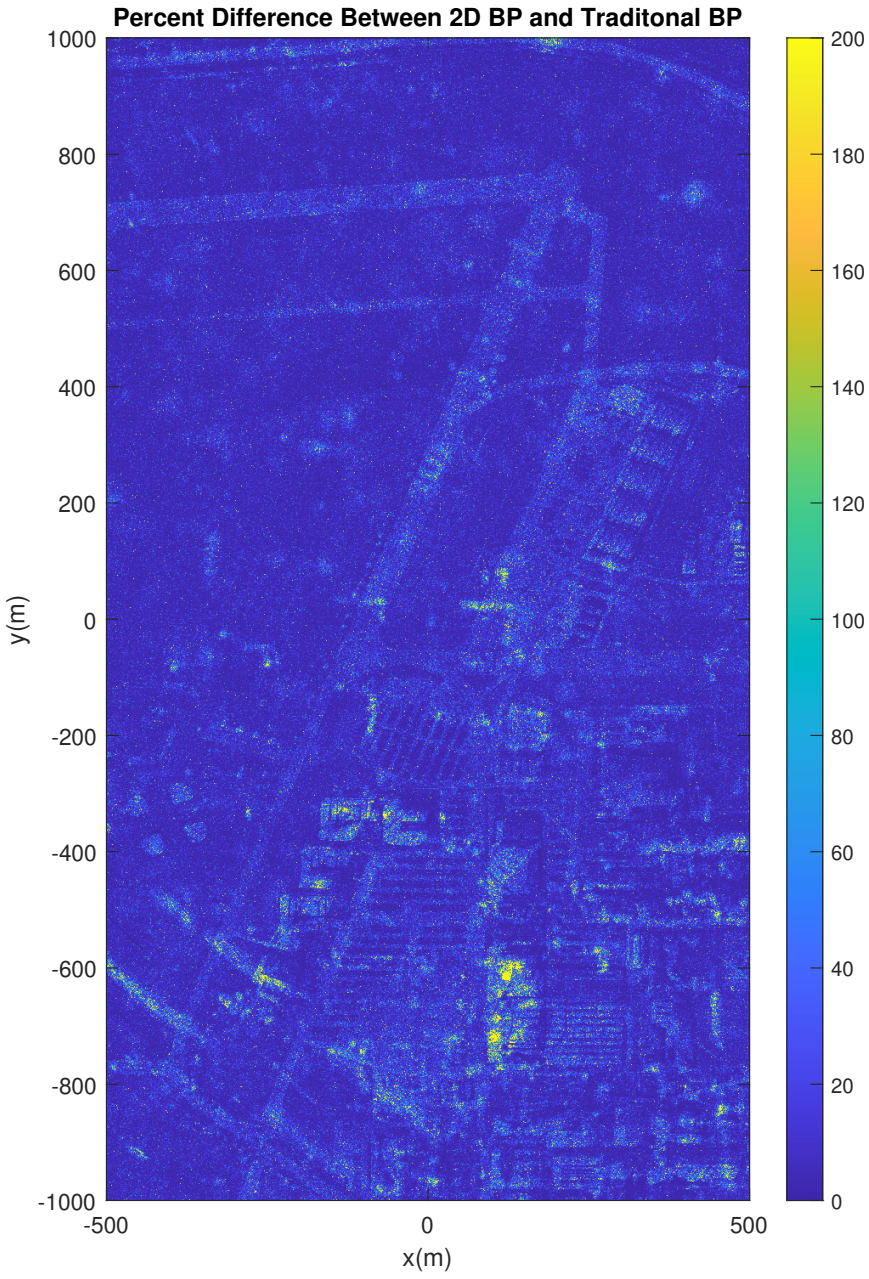


Figure 4.10: Percent difference between images formed by traditional and 2D back-projection.

To demonstrate the effects of increasing sub-CPI length on the image, images were formed for an increasing number of pulses in each sub-CPI. We expect that increasing the sub-CPI length will result in the aliasing artifacts, or grating lobes, observed in simulated data. Figure 4.11 shows three cases: 249 pulses per sub-CPI, 499 pulses per sub-CPI, and again 499 pulses per sub-CPI. The first two images are formed from sub-CPIs that have upsampled the Doppler axis during the Doppler FFT by a factor of four, whereas the last image is formed from a sub-CPI that upsampled the Doppler axis by a factor of sixteen.

We see that increasing from 31 pulses to 249 and then again to 499 results in grating lobes centered in cross-range at the brightest corner of the building seen in the previous figure. These grating lobes appear closer to this corner for a sub-CPI length of 499 pulses than 249, confirming the trend observed in simulated data. When the quality of the Doppler interpolation is poor, the effects of the grating lobes are more pronounced; therefore, the higher-quality Doppler interpolation used in the final image results in grating lobes being weaker. The copies still appear at the same location as the image using a low-quality Doppler interpolation but are only noticeable in very dark regions of the image. The cost of this high-quality Doppler interpolation is an increased memory requirement, as well as increased computation time. The 2D linear interpolation implemented on the finely upsampled range-Doppler maps is slowed for larger matrices. Additionally, the Fourier transforms needed to create the finely sampled range-Doppler maps require increased computation.

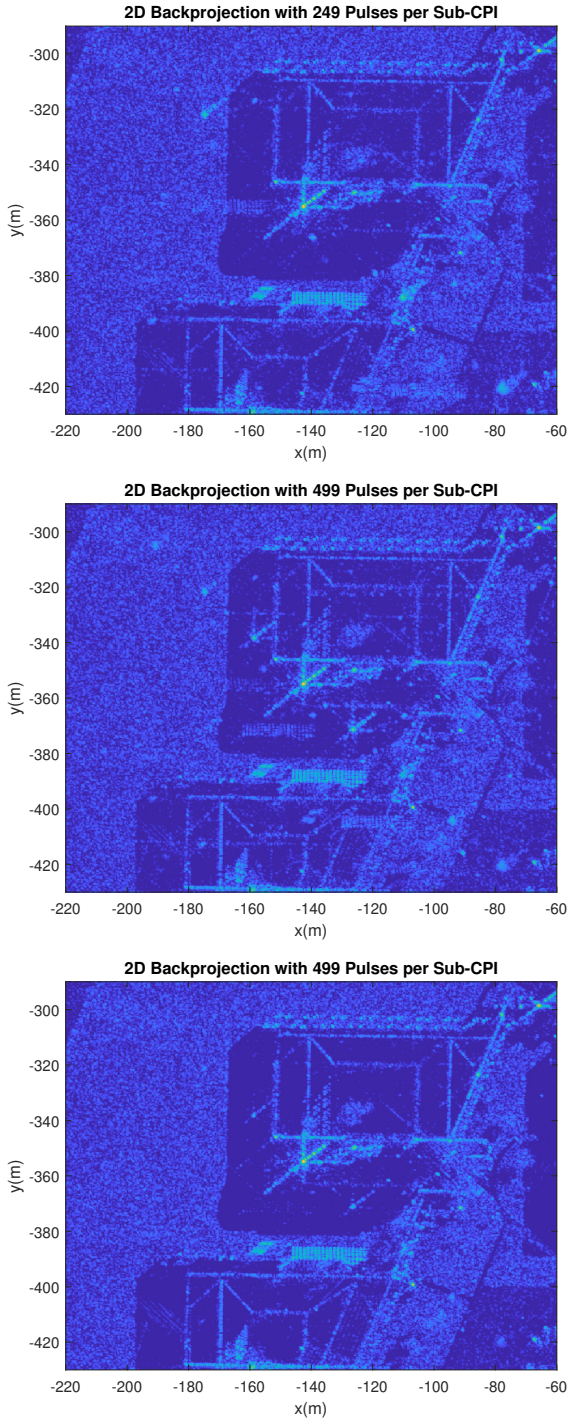


Figure 4.11: Results of increasing sub-CPI length to 249 and 499 pulses. High-quality Doppler interpolation used in the final image shows the importance of interpolation quality in preventing aliasing artifacts.

4.3 Moving Target Imaging

One of the many benefits of the 2D backprojection is the ability to perform moving target imaging for any arbitrary motion profile. This capability requires no additional assumptions beyond those required to perform imaging of a stationary scene, i.e. range migration within the sub-CPI is limited to be only linear. Just like imaging of a scene, pixel range and Doppler value must be calculated at the center of each sub-CPI. Here, each pixel will have an additional range and Doppler component depending on the motion profile of the pixel. Generally unknown, the motion profile must by hypothesized to accurately image a moving target. While difficult to predict the motion profile of a moving target, hypothesizing motion profiles of a pixel can be performed in a very similar manner to calculating motion profiles of stationary pixels in traditional imaging. Instead of using only a grid of ground-referenced pixel locations to calculate the motion profile, i.e. range and Doppler, for each sub-CPI, a grid of target velocities can also be used to calculate additional range and Doppler migration resulting from target motion. The result of this method yields a 4D image, $I(x, y, v_x, v_y)$, now also depending on v_x and v_y. While difficult to visualize in 4 dimensions, 2D cuts can be taken to image an entire scene at some given velocity or image a single pixel at all velocity hypotheses. Obviously this added capability comes at the cost of added computational complexity; every additional velocity hypothesis requires interpolation of every pixel within the scene for each new motion profile. Much too computationally intensive for imaging a large scene, an efficient implementation would be using a small set of velocity hypotheses to image only a small portion of the scene.

Figure 4.12 shows an example of an image of a simulated stationary scene containing a moving point-target. The target, moving at 5 m/s at a 45° angle from the

scene center, has not been focused and results in a smear across the stationary background. To image this moving target, we now perform 2D backprojection for a grid of pixel locations, x and y, and a grid of target velocities, v_x and v_y. The output of this approach can be interpreted as a grid of location-dependent pixel intensities, each calculated for specific motion hypotheses, given by v_x and v_y ; alternatively, this output can be interpreted as a grid of velocity-dependent pixel intensities, each calculated for a specific starting location of the target, x and y. Interpreting the output as the former, a 2D distributed image can be produced for a specific target velocity hypothesis; if this velocity hypothesis matches the true motion of the target, a well-focused image will be formed. On the left of Figure 4.13, the point target in Figure 4.12 is imaged in spatial coordinates, x and y, for a particular velocity hypothesis. It is clear that choosing the correct velocity hypothesis allows the algorithm to produce a fully focused image of the target; however, only targets moving with this exact motion profile will be imaged properly. Performing imaging for a non-zero target velocity results in the stationary scene becoming unfocused.

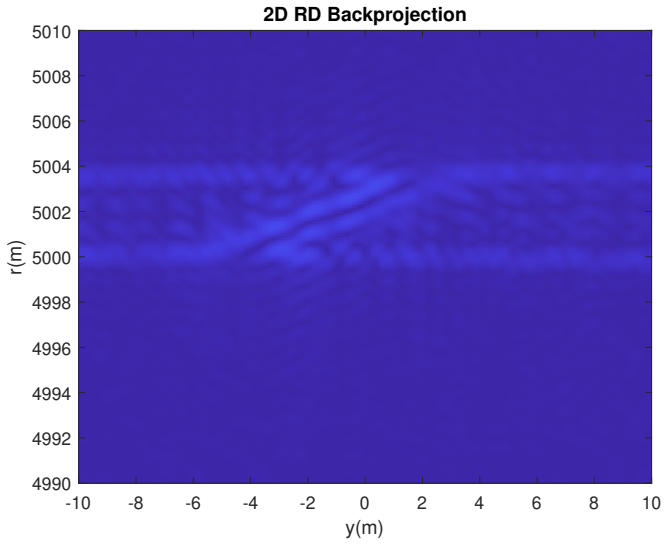
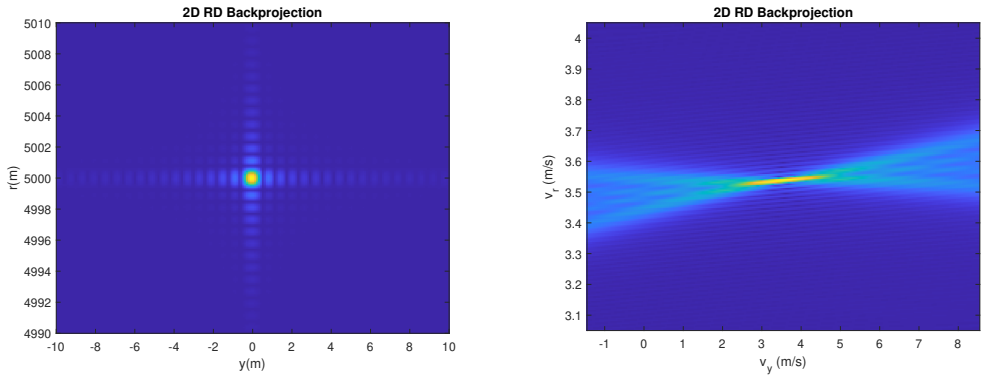


Figure 4.12: Image of a moving target with stationary target hypothesis.

On the right of Figure 4.13, the point target at the center of Figure 4.12 is imaged at a single starting location for a grid of velocities, v_x and v_y. Again, we see that the moving target is focused at the location corresponding to its velocity components in x and y; however, we notice a significant difference in resolution between radial and tangential velocity. The radar is able to make direct measurements of range and radial velocity, resulting in the high resolution observed. whereas tangential velocity is not directly measured by the radar. It is only possible to resolve tangential velocities through differences in range and Doppler, which are small for tangential velocities, hence the poor resolution in this dimension.

Imaging in this way can be very beneficial depending on the application desired. It is important to note that performing moving target imaging is not limited to a linear velocity hypothesis. Any arbitrary motion profile can be used to perform imaging; however, hypothesizing a higher order motion profile would drastically increase computational requirements. If the motion profile is known, there is no increase in computation over normal stationary imaging; the added computation

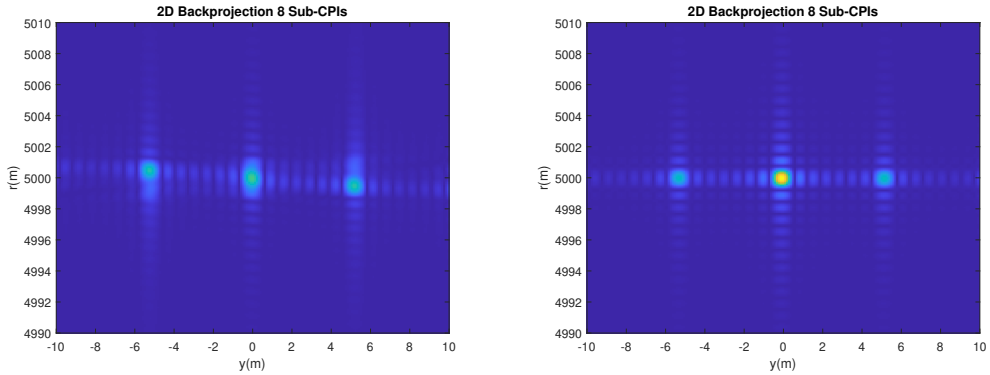


(a) (b)

Figure 4.13: Images of a moving target using **(a)** the correct motion hypothesis imaged to a grid of starting locations and **(b)** the correct starting location hypothesis imaged to a grid of velocities.

comes from the hypothesis and calculation of multiple motion profiles.

Throughout this chapter, the Keystone transform has been applied to each sub-CPI to correct for linear range migration. When short sub-CPIs are used, range migration over the course of the sub-CPI is likely small, but as sub-CPI length is increased, it will become more pronounced. This effect was shown in Figure 4.6 for sub-CPIs of two different lengths. When moving target imaging is introduced, additional range migration will result from target motion. Even if sub-CPIs are sufficiently short such that range migration due to platform motion is negligible and the Keystone transform is not required, moving targets can exhibit additional range migration that require the Keystone transform. Figure 4.14 shows two images, each imaged to the exact motion profile of a very fast moving target. The image on the left shows the effect of foregoing the Keystone transform and relying only on Doppler processing to focus a sub-CPI's range-Doppler map. Clearly, the 2D backprojection is not able to fully focus the target in the final image. The point target exhibits a very weak response and aliasing artifacts are very high, resulting from the interpolation of the poorly focused range-Doppler maps. The image on the



(a) (b)

Figure 4.14: Moving target imaging **(a)** without using the Keystone transform and **(b)** with using the Keystone transform.

right uses the Keystone transform to focus target energy in each sub-CPI's range-Doppler map, then coherently combines sub-CPIs. Application of the Keystone transform to each sub-CPI results in image quality that appears to be identical to that of imaging a stationary scene. While application of the Keystone transform adds computation, the increase in focus for moving target imaging and target detection is highly desirable and outweighs the need to increase efficiency, as focus is directly related to detector performance. Additionally, range migration can be detrimental to the ability to accurately detect targets. Therefore, the application of the Keystone transform lends itself very well to the simultaneous SAR-GMTI architecture for which 2D backprojection was designed. The added computation of performing the Keystone transform on each sub-CPI may be unnecessary for traditional imaging, but it is critical for moving target imaging and target detection.

4.4 Conclusions

In this chapter, results of imaging via 2D backprojection were presented and compared to traditional backprojection. The results showed that 2D backprojection is capable of achieving image quality that rivals that of traditional backprojection. However, aliasing artifacts that result from an increase in sub-CPI length can reduce image quality, requiring consideration when performing the algorithm. These aliasing artifacts act as if they are the grating lobes of the synthetic array, resulting from discontinuities in the method of coherently integrating returns from sub-CPIs. High-quality Doppler interpolation is required to prevent the impact of grating lobes being accentuated by poor interpolation quality; however, grating lobes will always be present unless sub-CPI length is chosen such that they fall outside the scene of interest. Additionally, the effects of these grating lobes can be ignored if they are

sufficiently weak; when applied to real data, only the brightest of scatterers within the scene show these aliasing artifacts. Even when present, they are often weak enough that overall image quality is unaffected. It is important to recall again that 2D backprojection was designed to serve as an imaging algorithm that is synergistic with GMTI processing. While image quality might be slightly degraded by choice of sub-CPI length, specific choice of sub-CPI length may be required for the GMTI technique performed on the sub-CPI. 2D backprojection also supports moving target imaging, just as traditional backprojection does. While the Keystone transform has a small effect when imaging a stationary scene, its application is required to properly image moving targets.

This chapter has focused entirely on image quality and imaging results of performing 2D backprojection in numerous ways. Throughout, the mention of computational requirements has been made several times, but only a surface level description has been supplied. Chapter 5 gives a more detailed analysis of computational differences of 2D backprojection as compared to traditional backprojection. The chapter will discuss several increases in computational requirements and will give examples of how the new requirements can be handled. In addition, the increased memory requirements of the algorithm will be discussed. The AFRL SAR Gotcha Large Scene Dataset will be used to discuss requirements of a typical application. Relative memory requirements and computation times will be given to demonstrate the requirements and capabilities of the algorithm.

Chapter 5

Computational Savings & Requirements

In Chapter 4, images were shown using 2D backprojection. These images were compared to results from traditional backprojection, and 2D backprojection was shown to exhibit comparable image quality. However, image quality was dependent on choice of sub-CPI length. Long sub-CPIs result in grating lobes that are undesirable and can become detrimental to image quality. Throughout the Chapter, mentions were made of computational requirements of 2D backprojection; here, a detailed discussion will be presented on these computational requirements. After computational saving are discussed, memory requirements of performing 2D backprojection will be discussed. The Gotcha dataset will be used to provide examples of typical memory requirements as well as computation times.

5.1 Reductions in Computation

With almost identical image resolution, it is important to note that the 2D backprojection benefits from several computational savings. While the traditional backprojection performs a 1D interpolation on each pulse, the 2D backprojection performs a 2D interpolation on each sub-CPI. A single 2D interpolation is slower than a single 1D interpolation; however, results show that the reduction in number of in-

terpolations dominates and overall computation time is decreased. Furthermore, a pixel's range and Doppler shift need only be calculated once per sub-CPI instead of computing the same pixel's range for every single pulse. Calculating the Euclidean distance to each pixel is extremely inefficient, so a sizable reduction in computations is achieved solely from the reduced number of range calculations.

While the Doppler shift of each pixel must also be calculated, relatively little computation is added by doing so. To determine what the measured Doppler shift of each pixel should be, we need the radial component of the radar's velocity. This radial velocity can be determined by first finding the angle between the radar's velocity vector and the line of sight to each individual pixel. This angle can be found by projecting one vector onto the other, described by

$$\frac{\mathbf{v_r} \cdot \mathbf{L}}{||\mathbf{v_r}||\,||\mathbf{L}||} = cos(\theta), \tag{5.1}$$

where $\mathbf{v_r}$ is the radar's velocity vector and $\mathbf{L}$ is the line of sight vector for a particular pixel. For simulated data, $\mathbf{v_r}$ is precisely known and can be chosen to have non-zero value in only one dimension, making the dot product much more simple. For real data, such as the Gotcha dataset, some approximation technique is used to find the velocity vector. The line of sight vector must be calculated for every single pixel that is to be imaged, requiring the x, y, and z coordinates of every pixel. Additionally, the magnitude of $\mathbf{L}$ is needed, requiring another expensive square root; however, this magnitude is simply the Euclidean distance from the radar to the pixel, previously calculated for use in the range interpolation and phase correction of the algorithm, so no extra computation has been added by this. The inverse cosine needed to find θ is also computationally expensive, but the cosine of the angle between the two vectors is needed when finding the radial velocity, $v_{rad} = v_r \cos(\theta)$.

Therefore, once the left side of (5.1) has been evaluated, it takes only one additional matrix multiply to find the radial component of velocity.

Another benefit of 2D backprojection is the ability to implement high quality upsampling while performing the Keystone transform. In traditional backprojection, interpolation of the range profile is performed in two steps to achieve full resolution and maintain computational efficiency. First, a fine upsampling of the range profile is performed, then a simpler interpolation method, such as linear interpolation, is used to interpolate values from the finely sampled profile. The upsampling can be performed by taking the FFT of the range profile, padding with zeros, then taking the inverse FFT to produce a range profile with a more finely-spaced range axis. For the traditional backprojection approach, this upsampling must be performed on each pulse. The 2D backprojection approach already converts the range axis into the frequency domain in order to implement the Keystone transform, so the interpolation can be directly built in by simply padding zeros before taking the inverse FFT to convert the data back into the time domain. The same strategy can also be applied when converting the sub-CPI data into the Doppler domain. Prior to performing the FFT in the pulse dimension, zeros can be padded to create very finely sampled range-Doppler maps. These range-Doppler maps will be interpolated using a simpler interpolation technique to form a full resolution image via 2D backprojection.

The Keystone transform is efficiently incorporated into the subaperture architecture of the 2D backprojection. The Keystone transform requires a fast-time frequency-dependent interpolation of slow-time data for each sub-CPI. There are several options for interpolation method, but a sinc-based interpolation is favorable for the 2D backprojection. This sinc-based interpolation requires the calculation of sinc-weights dependent on fast-time frequency and the slow-time sample loca-

tion that is desired. Calculating these weights involves calculating numerous sinc-weights needed to interpolate every single new sample location, one pulse at a time for each frequency sample in F. Typically performed in a nested for loop, interpolating in this way is much less efficient than implementing other interpolation techniques, such as a Fourier transform based interpolation. However, the structure of the sinc-based interpolation and the need to perform the Keystone transform on every sub-CPI allows for a key exploitation. The locations of the sinc-weights needed to perform the interpolation are dependent on only fast-time frequency and local slow-time sample locations.

If each sub-CPI is chosen to be structured the exact same way, i.e. each sub-CPI has the same frequency content and the same slow-time axis, sinc-weights can be computed once and recycled for use in all sub-CPIs. Additionally, the sinc-weights can be pre-computed for a know frequency axis and pulse scheme. Once sinc-weights are calculated, a single matrix multiplication performs the interpolation needed for the Keystone transform. Using this method provides a significant speed up over other interpolation methods, especially when many sub-CPIs are used. The results from the Gotcha dataset presented in the previous section used 966 sub-CPIs each containing 31 pulses. While another interpolation method may be faster on a single sub-CPI, calculation of the sinc-weights one time reduces the interpolation complexity down to a single matrix multiply per sub-CPI.

Given all the computational savings achieved by the use of sub-CPIs, it seems clear that the ideal implementation would be to use long sub-CPIs to further decrease the number of range calculation, upsamplings, and interpolations. The effects this has on image quality have been discussed extensively in the previous chapter, but now we can analyze the computational requirements of doing so. If sub-CPI length is increased, the reduced number of range calculations, upsam-

plings, and interpolations would drastically increase computational efficiency. If sub-CPI length is doubled, the number of these calculations needed is cut in half, theoretically cutting computation time in half as well. This trend would be observed until grating lobes appear and oversampling of the Doppler axis is required for mitigation. This oversampling increases computational complexity of the upsampling step, as well as increasing computational complexity of the 2D interpolation of the range-Doppler map.

5.2 Memory Requirements

It is important to point out the significant increase in memory requirement for this technique. Traditional backprojection finely upsamples a single range profile and holds it in memory for only one pulse; however, 2D backprojection must finely upsample a range-Doppler map for each sub-CPI. This range-Doppler map uses tens or hundreds of pulses to convert to Doppler and must also finely upsample the Doppler axis. Performing the same high quality upsampling of the range axis as traditional backprojection, required for adequate image quality, now results in a memory requirement that can be several hundred times more than the traditional backprojection. Luckily, modern computing can often handle this increased memory requirement, but workstation limitations can limit the ability to perform the 2D backprojection. Specifically, sub-CPI length will need to be limited to prevent the range-Doppler maps from using too much memory. Also, mitigation of aliasing artifacts discussed in the previous section requires even higher upsampling of the Doppler axis, further decreasing the number of pulses that can be used in a single sub-CPI. However, the use of more pulses in the sub-CPI, hence the sub-CPI separation, is what causes the aliasing artifacts to diminish image quality. Decreasing the

number of pulses in the sub-CPI will decrease memory requirement and increase image quality.

Using the Gotcha dataset to determine typical memory requirements and computation times for 2D backprojection, we may perform a quantitative analysis of the computational benefits of this algorithm. This dataset consists of 21,232 fast-time frequency samples and 30,000 slow-time samples. The range axis for traditional backprojection, as well as 2D backprojection, was oversampled by a factor of 10 to yield a finely-sampled range axis containing 212,320 range bins. By default, MATLAB stores arrays as double-precision floating point numbers, requiring 8 bytes of memory per array entry. By nature, radar data is complex valued, now requiring 16 bytes of memory per array element. Again, traditional backprojection need only upsample the range axis one pulse at a time, resulting in an array consuming only about 3.4 MB of memory. Obviously, modern workstations are well-suited to handle arrays of this size. However, 2D backprojection uses these same upsampled range axes, but instead uses several pulses to create a range-Doppler map.

For the image shown in Figure 4.8, 31 pulses were used per sub-CPI. The slow-time axis of each sub-CPI was then padded to length 512 and converted to the Doppler domain, resulting in a matrix consuming 1.74 GB of memory. Again, modern computing is more than capable of handling matrices of this size. However, as sub-CPI length is increased, additional upsampling must be performed. Figure 4.11 showed that oversampling is required to obtain adequate image quality when using sub-CPIs of increasing length. Here, sub-CPIs contained 499 pulses each, then padded to length 8,192 and converted to the Doppler domain. This matrix is substantially larger than the previous case, requiring 27.83 GB of memory. Additionally, this is not the only matrix that must be stored in memory for the algorithm. For large scenes containing millions of pixels, holding pixel values, as

well as all the values needed for interpolation, i.e. the range and Doppler of every pixel, is non-trivial and can require significant memory. The full Gotcha scene size is roughly 5km **x** 5km with range and cross-range resolution of about .25 m, resulting in a scene containing 400 million pixels and requiring 6.4 GB of memory just for the image itself. Range and Doppler values are each stored as real-valued doubles, requiring an additional 6.4 GB of memory. Moderate workstations will still typically possess enough RAM to handle the large memory requirements of the 2D backprojection; however, the major downfall of this increased memory usage is the inability to utilize highly efficient GPUs.

The use of GPU computing in radar image processing can provide significant speed up over general CPU computing [22]. The ease of implementation in MATLAB is aided by the extensive library of built-in functions that can be used for CPU and GPU computing. In general, GPU computing is much faster than general purpose computing on a CPU. However, the increase in efficiency is controlled by several factors, including the task being performed and the overhead of the data transfer required to send data between the CPU and GPU. Tasks that are highly parallel, such as the FFT, are much faster when processed via GPU, which is specifically designed to handle tasks of this nature. Additionally, it is extremely inefficient to send data back and forth between the CPU and GPU. It is ideal to send over all the required data, perform all processing required by the GPU, then send all the processed data back to the CPU. For 2D backprojection, this would mean holding the radar data, all the variables required for calculating range and Doppler values of every pixel, and pixel intensities. The size of the array containing the pixel locations alone is substantially large, and transferring it back and forth on each sub-CPI would vastly decrease the efficiency achieved by using the GPU. Therefore, peak efficiency is achieved when all GPU-based processing parameter are transferred once, held for

all sub-CPIs while the GPU continues to process the data, then returned to the CPU once all sub-CPIs have been processed. This imposes strict memory requirements on the GPU.

Recently, GPUs have improved substantially, but even a high end GPU possess significantly less memory than a CPU has in RAM. The NVIDIA Titan RTX used for processing this data has 24 GB of memory, an enormous amount by GPU standards, but it is clear that limitations will now be imposed on the 2D backprojection. First, the use of larger sub-CPIs is restricted to some extent; using a sub-CPI length of 499 pulses, requiring 8,192 Doppler samples for adequate image quality, is clearly not possible as the memory requirement of the data alone, nearly 28 GB, surpasses the available memory of the GPU. Shorter sub-CPIs don't violate this requirement, so implementation on a GPU is possible, but considerations are required when determining sub-CPI length desired. Additionally, the large scene desired to be imaged in this dataset introduces significant memory requirements. As discussed earlier the range and Doppler values needed for interpolation require 6.4 GB of memory, and once interpolated, pixel intensity must be stored on the GPU to allow for summation of each subsequent sub-CPI, requiring an additional 6.4 GB.

While calculating Doppler requires little extra computation, the line of sight vector must be calculated for each pixel within the scene, requiring a 3D vector for each of the 400 million pixels. This requires nearly 20 GB of memory, nearly exceeding the capabilities of the Titan RTX. Holding any other data in memory would result in fully saturating the memory of the GPU. While traditional backprojection images a scene of the same size, it does not require the calculation of Doppler; therefore, the calculation of the line of sight vector is also not required. This allows the traditional backprojection to handle the memory requirement of the full scene.

Therefore, smaller scenes will be necessary for the 2D backprojection to utilize the efficiency of GPU processing.

5.3 Computation Time of Gotcha Dataset

Now that the computational savings achieved by, as well as the memory requirements of, 2D backprojection have been discussed, we will look at the computation times measured when implementing this algorithm. We will also use computation times of traditional backprojection to provide a baseline for comparison. Table 5.1 shows the computation times of several different implementations of the algorithms. For all implementations, it was assumed the data was in the correct format and the data was already loaded into MATLAB for processing. It is clear that 2D backprojection provides a significant reduction in computation time over traditional backprojection. Additionally, we see that increasing sub-CPI length results in a slight reduction in computation time; however, this reduction is very small. While the sub-CPI length has more than doubled and the number of sub-CPIs has been cut in half, the Doppler axis contains twice as many points as compared to the shorter sub-CPI. This fact results in larger range-Doppler maps that must be interpolated for each sub-CPI.

The results shown in Table 5.1 show that the reduction in computation gained

Algorithm	**N**	**Sub-CPIs**	**M**	**Scene Size**	**Time**
BP	30,000	N/A	N/A	1km **x** 1km	0:38:10.6
2D BP USF 16	29,946	966	31	1km **x** 1km	0:8:30.6
2D BP USF 16	29,988	476	63	1km **x** 1km	0:8:14.1
2D BP USF 8	29,972	236	127	1km **x** 1km	0:4:59.5

Table 5.1: Computation times of different implementations of the 2D backprojection. Image quality of 2D backprojection was comparable to that of traditional backprojection until the upsampling factor (USF) was decreased.

by using longer sub-CPIs is about even with the increase in computation resulting from the use of larger range-Doppler maps. When sub-CPI length is again roughly doubled to 127 pulses, the number of sub-CPIs is reduced to about half. However, the Doppler axis is only upsampled by a factor of 8 here, resulting in a Doppler axis that is the same length as that of the length 63 sub-CPI. Decreasing the number of sub-CPIs and holding the length of the Doppler axis constant results in a reduction in computation that is much more significant. While the reduction in computation time is significant, the decrease in image quality is also very significant; grating lobes become much more apparent when the sub-CPI length is increased and upsampling is decreased. With any algorithm, trade-offs are required between efficiency and quality, and without sacrificing quality, 2D backprojection is still able to increase efficiency significantly. However, there is a limit to the increase in efficiency that can be achieved while still maintaining high image quality. The implementations used for calculating computation times in Table 5.1 are not optimized, but show how simple changes to sub-CPI length and Doppler oversampling can affect relative processing times.

5.4 Conclusions

In this chapter, the resulting computational requirements of 2D backprojection were discussed. The 2D backprojection technique benefits from several computational savings over the traditional backprojection. While traditional backprojection must perform range calculations and interpolations for every pixel in the scene of interest once for every pulse, the 2D backprojection requires these only once per sub-CPI. While there are far fewer interpolations needed, 2D backprojection requires a 2D interpolation that is more computationally intensive than the 1D inter-

polation of traditional backprojection. This 2D interpolation requires calculation of range and Doppler for every pixel within the scene; however, calculation of Doppler comes with little extra computation. Implementation shows that the 2D backprojection is much more efficient than the traditional backprojection, but actual computation times vary depending on several factors. This increase in computational efficiency comes at the cost of increased memory requirements. While modern day hardware can handle these requirements, limitations may be set that limit efficient calculation. Namely, smaller scene sizes may be required in comparison to traditional backprojection as the range-Doppler maps can require a significant portion of available memory. Results of applying the algorithm to the Gotcha Large Scene Dataset show that 2D backprojection is much more efficient than traditional backprojection. In addition to higher computational efficiency and comparable image quality, the algorithm's architecture is structured such that a post-Doppler GMTI technique can be performed simultaneously to efficiently produce an image and perform target detection.

Chapter 6

Conclusion

6.1 Conclusions

A 2D backprojection technique, modeled after the well-know backprojection algorithm, has been studied in great detail. 2D backprojection has been derived and implemented on real and simulated data. An analysis of image quality and artifacts of 2D backprojection has been performed. Algorithm performance in terms of both quality and efficiency was analyzed and compared to traditional backprojection. Just like traditional backprojection, 2D backprojection produces high-fidelity SAR images; however, 2D backprojection can be implemented in substantially less time. This computational efficiency is gained from the reduction in number of interpolations needed to form a SAR image; while traditional backprojection requires a 1D interpolation for each pulse, 2D backprojection requires a 2D interpolation for each sub-CPI. The number of sub-CPIs within the data collection is subject to choice, allowing for variable computation time with variable image quality. Longer sub-CPIs seem to be the most efficient choice. However, results show that maintaining acceptable image quality limits the ability to increase efficiency; adequate Doppler oversampling can result in increased memory usage and slower interpolations. The increase in computational efficiency gained from using 2D backprojection comes

at the cost of an increased memory requirement. Often, this increased memory requirement is suitable for implementation on common hardware; however, the memory requirement depends on many parameters and implementation can be limited by the abilities of the workstation being used. The ability to produce high quality images in less time than traditional backprojection makes 2D backprojection an effective imaging algorithm, but it is important to recall that the architecture of 2D backprojection has been chosen to be well-suited for simultaneous SAR-GMTI processing. Therefore, efficient imaging isn't the only capability of the algorithm; the algorithm may eventually be used in an architecture that favors computational efficiency of GMTI processing.

6.2 Future Work

The obvious next step that should be taken with this algorithm is determining the method in which GMTI processing will be implemented. The data format is such that each sub-CPI is converted to the Doppler domain, beneficial to any practical post-Doppler GMTI technique. The most rudimentary GMTI processing would be treating each sub-CPI as a single dwell, performing a post-Doppler GMTI algorithm on each sub-CPI independently. A more interesting approach is the aggregation of GMTI outputs to coherently integrate returns of all sub-CPIs. These returns could be integrated using a backprojection style approach similar to the approach taken here by the 2D backprojection technique. Just as a target has a very specific location in the range-Doppler map of a sub-CPI, a target will have a very specific location in the range-Doppler-angle cube of a sub-CPI. Each stationary target exhibits a very specific migration through the cube from sub-CPI to sub-CPI. Moving targets also have a unique migration through the cube that depends on the specific motion

profile of the target. It is theoretically possible to hypothesize a target's motion profile, then to use this motion profile to interpolate values from the datacube for that particular motion profile. If the post-Doppler GMTI technique used to create the output statistic conserves phase, it is then possible to coherently integrate returns over the full CPI. This would allow for returns from slow moving targets, typically difficult to detect as they are embedded within background clutter, to be coherently integrated. This approach would improve minimum detectable velocity and detector performance.

There is a severe increase in computational requirements that comes with this backprojection style combination; a 3D interpolation now must be performed on each sub-CPI of data to pull values from the radar datacube. The 2D backprojection already requires calculation of range and Doppler, so the only additional variable needed to perform the 3D interpolation is angle. However, the angle of each pixel has already been calculated when determining Doppler. These values would then be used to interpolate the output statistic of each sub-CPI. It is only possible to integrate the returns of moving targets if the correct motion hypothesis is used, a difficult task to accomplish without *a priori* knowledge of target motion.

The ability to improve GMTI performance through a backprojection style integration is an enticing possibility. Future work on this algorithm will be needed to determine the effects of the processing steps chosen prior to GMTI implementation. In particular, application of the Keystone transform, beneficial in preventing the undesired range migration present in a sub-CPI, may affect the ability to perform GMTI processing properly. Ideally, the Keystone works only to mitigate range migration, having no residual effect on GMTI processing. Once developed, simultaneous SAR-GMTI can be performed. The benefits of this algorithm could then be compared to other simultaneous SAR-GMTI techniques. A more detailed study of

computational requirements of 2D backprojection could also be performed in future work. Here, relative computation times were used to compare the computational requirements of 2D backprojection to that of traditional backprojection; however, this is an *ad hoc* way of measuring computational requirements. An actual analysis of computational requirements would provide a much more advanced comparison of this algorithm to other existing algorithms.

Bibliography

[1] W. Carrara, W. Carrara, R. Goodman, and R. Majewski, *Spotlight Synthetic Aperture Radar: Signal Processing Algorithms*, ser. Artech House remote sensing library. Artech House, 1995. [Online]. Available: https://books.google.com/books?id=uztiQgAACAAJ

[2] C. A. Wiley, "Synthetic aperture radars," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-21, no. 3, pp. 440–443, 1985.

[3] L. A. Gorham and L. J. Moore, "SAR image formation toolbox for MATLAB," in *Algorithms for Synthetic Aperture Radar Imagery XVII*, E. G. Zelnio and F. D. Garber, Eds., vol. 7699, International Society for Optics and Photonics. SPIE, 2010, pp. 46 – 58. [Online]. Available: https://doi.org/10.1117/12.855375

[4] R. Deming, M. Best, and S. Farrell, "Polar format algorithm for SAR imaging with Matlab," in *Algorithms for Synthetic Aperture Radar Imagery XXI*, E. Zelnio and F. D. Garber, Eds., vol. 9093, International Society for Optics and Photonics. SPIE, 2014, pp. 47 – 66. [Online]. Available: https://doi.org/10.1117/12.2050681

[5] A. W. Doery, "Synthetic aperture radar processing with tiered subapertures," Ph.D. dissertation, University of New Mexico, 1994.

[6] B. D. Rigling and R. L. Moses, "Taylor expansion of the differential range for monostatic sar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 1, pp. 60–64, 2005.

[7] L. A. Gorham and B. D. Rigling, "Dual format algorithm implementation with gotcha data," in *Algorithms for Synthetic Aperture Radar Imagery XIX*, E. G. Zelnio and F. D. Garber, Eds., vol. 8394, International Society for Optics and Photonics. SPIE, 2012, pp. 19 – 24. [Online]. Available: https://doi.org/10.1117/12.925065

[8] N. E. Doren, C. V. Jakowatz, D. E. Wahl, and P. A. Thompson, "General formulation for wavefront curvature correction in polar-formatted spotlight-mode sar images using space-variant post-filtering," in *Proceedings of International Conference on Image Processing*, vol. 1, 1997, pp. 861–864 vol.1.

[9] M. Preiss, D. Gray, and N. Stacy, "Space variant filtering of polar format spotlight sar images for wavefront curvature correction and interferometric processing," in *IEEE International Geoscience and Remote Sensing Symposium*, vol. 1, 2002, pp. 179–181 vol.1.

[10] R. P. Perry, R. C. DiPietro, and R. L. Fante, "Coherent integration with range migration using keystone formatting," in *2007 IEEE Radar Conference*, 2007, pp. 863–868.

[11] M. A. Richards, "The keystone transformation for correcting range migration in range-doppler processing," March 2014. [Online]. Available: http://users.ece.gatech.edu/mrichard/Keystone.pdf

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

[13] M. Richards, *Fundamentals Of Radar Signal Processing*. McGraw-Hill Education (India) Pvt Limited, 2005. [Online]. Available: https://books.google.com/books?id=qizdSv8MEngC

[14] J. Ward, "Space- time adaptive processing for airborne radar," Technical Report, December 1994.

[15] W. L. Melvin, "A stap overview," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 19–35, 2004.

[16] R. C. DiPietro, "Extended factored space-time processing for airborne radar systems," in *[1992] Conference Record of the Twenty-Sixth Asilomar Conference on Signals, Systems Computers*, 1992, pp. 425–430 vol.1.

[17] R. Klemm and I. of Electrical Engineers, *Space-time Adaptive Processing: Principles and Applications*, ser. Electronics & communication engineering journal. Institution of Electrical Engineers, 1998. [Online]. Available: https://books.google.com/books?id=oQZTAAAAMAAJ

[18] J. R. Fienup, "Detecting moving targets in sar imagery by focusing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 3, pp. 794–809, 2001.

[19] R. W. Deming, "Along-track interferometry for simultaneous SAR and GMTI: application to Gotcha challenge data," in *Algorithms for Synthetic Aperture Radar Imagery XVIII*, E. G. Zelnio and F. D. Garber, Eds., vol. 8051, International Society for Optics and Photonics. SPIE, 2011, pp. 201 – 218. [Online]. Available: https://doi.org/10.1117/12.883282

[20] P. K. Sanyal, D. M. Zasada, and R. P. Perry, "Detecting moving targets in sar via keystoning and multiple phase center interferometry," in *2006 IEEE Conference on Radar*, 2006, pp. 6 pp.–.

[21] C. E. Muehe and M. Labitt, "Displaced-phase-center antenna technique," *Lincoln Laboratory Journal*, vol. 12, no. 2, pp. 281–296, 2000.

[22] T. D. R. Hartley, A. R. Fasih, C. A. Berdanier, F. Ozguner, and U. V. Catalyurek, "Investigating the use of gpu-accelerated nodes for sar image formation," in *2009 IEEE International Conference on Cluster Computing and Workshops*, 2009, pp. 1–8.