ABSTRACT

FAST TARGET TRACKING TECHNIQUE FOR SYNTHETIC APERTURE RADARS

by Kyle Kauffman

Modern radar imaging requires advanced Synthetic Aperture Radar (SAR) techniques in order to compensate for the low resolution typically found on airborne radars. However, the techniques used to process the data collected during SAR operation require a great deal of computation. This paper proposes a novel efficient algorithm and approach to processing the data collected to quickly get high-resolution approximations for the locations of non-moving targets.
# Table of Contents

List of Tables .................................................. iv

List of Figures .................................................. v

Chapter 1. Introduction ......................................... 1

Chapter 2. Synthetic Aperture Radar ......................... 4
  2.1. Introduction .............................................. 4
  2.2. Problem Setup .......................................... 5
    2.2.1. Single Target, Noiseless Scenario .................. 6
    2.2.2. Multiple Target, Noiseless Scenario ............... 6
    2.2.3. Noisy Environments ................................ 7
  2.3. Radar Waveform Design .................................. 7
    2.3.1. Radar Waveform Benchmarks ......................... 7
    2.3.2. LFM .................................................. 7
    2.3.3. Gaussian Pulse ..................................... 8
    2.3.4. OFDM ............................................... 9
    2.3.5. Chosen Pulse ...................................... 10
  2.4. Previous Work ........................................... 10
    2.4.1. Range - Conventional Method ....................... 11
    2.4.2. Cross-Range - Conventional SAR Wavefront Reconstruction 11
    2.4.3. Wavefront Reconstruction Asymptotic Complexity .... 12
    2.4.4. Wavefront Reconstruction Cross-Range Resolution .... 13

Chapter 3. Search Space Algorithms ......................... 17
  3.1. Introduction ............................................. 17
  3.2. Search Methods ......................................... 17
    3.2.1. Binary Search ...................................... 17
    3.2.2. Feature Extraction ................................ 18
    3.2.3. Local Search ....................................... 18
  3.3. Methods Considered for Future Work .................... 19
    3.3.1. Genetic Algorithms ................................. 19
    3.3.2. Slow-time Window Narrowing ....................... 19
    3.3.3. Distribution Fitting ................................ 20

Chapter 4. Novel Sparse Target SAR Algorithm ............. 21
  4.1. Introduction ............................................. 21
  4.2. Algorithm Overview .................................... 23
    4.2.1. Path Acquisition .................................. 23
    4.2.2. Path Tracking ..................................... 24
  4.3. Alternate Setups ........................................ 26
    4.3.1. Real-time Constrained .............................. 26
Chapter 5. Analysis of Sparse Target SAR Algorithm .......................... 28
  5.1. Single Target Case .................................................................. 28
    5.1.1. Noiseless ........................................................................ 28
    5.1.2. Noisy ............................................................................ 29
  5.2. Multiple Target Case ............................................................... 30
    5.2.1. Range ............................................................................ 30
    5.2.2. Cross-Range Target Resolving ......................................... 31
  5.3. General Case ......................................................................... 33
  5.4. Asymptotic Complexity ........................................................... 34
  5.5. Simulations Results ................................................................. 34
    5.5.1. Cross Range Resolution Results ...................................... 34
    5.5.2. Computational Performance Results ................................. 35
    5.5.3. Missed Detection Results ............................................... 36

Chapter 6. Future Work .................................................................. 38
  6.1. Alternate Waveforms ............................................................... 38
  6.2. Simulation of Alternate Setups ................................................. 38
  6.3. Additional Scenarios .............................................................. 38

Chapter 7. Conclusion ................................................................. 39

References .................................................................................... 40
LIST OF TABLES

Table 4.1. Parameters for real-time setup. .......................................................... 26
Table 5.1. Table of parameters used for simulations. ........................................ 28
Table 5.2. Cross-range resolution as a function of target down-range. Y is the range of the two targets, and XRR is the smallest distance between the targets that the algorithm was able to distinguish them as distinct from each other. ................................................................. 35
Table 5.3. Extra computations required by novel algorithm due to tracebacks. \( T_r \) is the percent increase in computations caused by the tracebacks, compared to the novel algorithm without tracebacks enabled. It is calculated as \( T_r = 100 \left( 1 - \frac{F_T}{F_{NT}} \right) \), where \( F_T \) is the number of floating point multiplications calculated for the algorithm including tracebacks, and \( F_{NT} \) is the number calculated excluding the tracebacks. ................... 36
List of Figures

Figure 1.1. A radar tracking point targets using a large-aperture antenna (left), and with a narrow-aperture antenna (right). The enclosed area constitutes the space that the plane’s transmitted pulse will reach. The narrow-aperture antenna will receive reflections from more widely spaced targets, resulting in more ambiguities in target location.

Figure 1.2. An airplane taking 3 recordings along a fixed axis. The inner two targets would generate reflections at all 3 points, whereas the closer target would only reflect once.

Figure 2.1. A single transmission of a pulse occurs. When it reaches the point target, the pulse is reflected off in every direction in a spherical pattern.

Figure 2.2. An airplane travelling along a fixed axis. It illuminates targets in an imaging area adjacent to the direction of travel.

Figure 2.3. An LFM signal. The frequency of the sinusoid increases linearly.

Figure 2.4. A gaussian pulse with $\mu = 50$.

Figure 2.5. The spectrum of an OFDM signal.

Figure 2.6. The reference triangle wave used for simulations. Time $t=0$ marks the beginning of transmission.

Figure 2.7. Phase history of a target at $x=500$ and $y=200$. The large main-lobe is centered around the $x$ coordinate of the target. By correlating phase angles recovered against ideal phase angle of a target at a particular $x$ coordinate, we can recover the target cross-range coordinate.

Figure 2.8. (top) The phase history of two targets at $x=500$ and $x=600$. Both have range $y=200$. (middle) The phase history of the single target at $x=500$. (bottom) The phase history of the single target at $x=600$.

Figure 2.9. The cross correlation of the phase history of two targets at $x=500$ and $x=600$ with a reference phase history with $x=500$.

Figure 4.1. An airplane recording reflected signals from a target. The plane flies closer to the target until it is aligned, and then flies away from it.

Figure 4.2. An illustration of the input data vector for a single target. Each bin contains a single reflection of the reference pulse. The pulse is time delayed based on the distance.

Figure 4.3. Overview of the components of the novel cross-range algorithm. The left flowchart represents the path acquisition, and the right flowchart represents the path tracking loop.

Figure 4.4. Searching the correlation space with bounded phase shift. If we know the path in the adjacent bin can move by a max of 1, we only have to search three samples per bin.

Figure 4.5. Binary search along initial vector path.

Figure 5.1. Single, noiseless target.

Figure 5.2. Single target with noise vector added in. SNR = 5 dB.

Figure 5.3. Single target with noise vector added in. SNR = 0 dB.
Figure 5.4. Multiple targets with different range. ................................. 31
Figure 5.5. Two targets with converging paths. ................................. 31
Figure 5.6. Two targets with same range, different cross-range. .......... 32
Figure 5.7. Two targets with same range, different cross-range. The closeness of the targets causes the slopes to be very similar. ...................... 33
Figure 5.8. Random targets, with $N_t=5$ and SNR=-3 dB. ................. 33
Figure 5.9. Probability of missed detection versus cross-range resolution. 
$\delta R$ is the distance between two objects. As $\delta R$ gets small, it is more likely that the tracker will see only one object and not two, thereby not detecting the second target. ................................................. 36
Figure 5.10. Probability of missed detection versus cross-range resolution. 
$\delta R$ is the distance between two objects. We see that both $\delta R$ and SNR affect the number of missed detections. ................................. 37
Chapter 1

INTRODUCTION

The problem of detecting and locating a remote target with high precision has been considered since the invention of the early radar. In traditional echo-location radar, a large, rotating, highly directional antenna is used to illuminate a target; the recorded reflections off the target are used to determine the targets down-range distance as well as its bearing.

The resolution of this type of radar in the cross-range (the axis perpendicular to the down-range) is limited by the aperture size of the antenna [1]. Figure 1.1 illustrates echo-location with a narrow aperture and a wide aperture antenna. The narrow-aperture antenna transmits its pulse in a much wider arc, resulting in reflections from many more targets than the wide aperture antenna. By correlating we can still find range information on the point targets, however there is now more ambiguity as to where the point targets are. For example, in Figure 1.1, the two closest targets are at relatively the same range. Since they are both illuminated by the narrow-aperture radar, their reflections would be received at nearly the same time and there would be no way to distinguish them further. To increase the cross-range resolution, a bulky, larger antenna with a wide aperture is required [1]. This means that increasing target position resolution is not possible if the antenna becomes too large for the platform and application.

![Image of radar tracking point targets using a large-aperture antenna (left), and with a narrow-aperture antenna (right). The enclosed area constitutes the space that the plane’s transmitted pulse will reach. The narrow-aperture antenna will receive reflections from more widely spaced targets, resulting in more ambiguities in target location.](image)

In imaging radar, a technique called synthetic aperture radar (SAR) is used to increase the cross-range resolution beyond the aperture-imposed limit of a single antenna. This increased resolution allows SAR to be used to construct high resolution images. SAR works by simulating a larger aperture antenna using an array of smaller antennas. Since each of the SAR antennas is small, it serves as a portable...
alternative to conventional radar when high resolution is necessary but a single, bulky antenna is impractical. Each antenna in the array collects data individually, and the data is then recombined and post-processed to simulate a larger aperture antenna.

![Figure 1.2](image.png)

**Figure 1.2.** An airplane taking 3 recordings along a fixed axis. The inner two targets would generate reflections at all 3 points, whereas the closer target would only reflect once.

One downside of SAR is that it requires computationally intensive post processing of a large set of data. The resolution obtainable is often limited to the amount of computational power available. Due to this limitation, early SAR in the 1960s-1970s was based on analog processing of the antenna array data using range Doppler imaging and polar format processing [2]. As computational power grew through the 1980s, SAR processing could take advantage of more complex and computationally expensive algorithms and techniques. This allowed for more modern digital processing theory such as SAR wavefront reconstruction to be applied [3]. In modern SAR, vehicles with a large amount of computational power are capable of constructing high resolution images of targets.

SAR still remains unusable in many situations where computation of large data sets is not feasible. Since SAR is principally utilized for imaging radar, signal processing algorithms are tailored towards the construction of high resolution images, gathering large amounts of information about the environment. Instead of just extracting the location of targets, an image of the entire scene is constructed, mapping the reflectivity on an analog scale for all points in the imaged area. The large amount of data being processed during image construction causes SAR wavefront reconstruction to require a great deal of computational power. This computational requirement disallows its usage in systems with real-time requirements or in environments where processing power is limited, such as a sensor network or an embedded system with limited resources.

The goal of this thesis is the development of an efficient, fast algorithm for detection and tracking of targets using raw SAR data. Instead of generating an image, we perform local searches within the SAR data to locate and identify the location of useful information. We use the discovered features to quickly locate the positions of targets illuminated by our radar. This will require the development of a new specialized technique that examines the properties of the SAR data, evaluates a
strategy to extract target locality information, and executes a local search to retrieve the data as quickly as possible.

In Chapter 2, we examine the underlying SAR theory, and previous approaches to target detection. In Chapter 3, we evaluate existing related search algorithms for ways to efficiently search for and locate features of interest. In Chapters 4 and 5, we design a sparse target algorithm and test its performance in simulated scenarios of interest. Lastly, In Chapters 6 and 7 we will lay out a roadmap for research and usage of the SAR sparse target algorithm.
2.1 Introduction

SAR is a method of increasing cross-range resolution of targets imaged by a radar. It is used to construct high resolution images of the area illuminated by the radar transmissions. SAR is widely used in a broad range of applications, including intelligence, surveillance [4], imaging of remote space objects, and other emerging applications in fields such as navigation [5][6].

The SAR technique involves using an array of antennas to collect received data. Since each antenna is in a different location, the data collected by each antenna will vary according to the field strength at its spatial location. The SAR array captures the field strength of the reflected waves at each spatial location that an antenna is located at. By collecting a time-series of samples from each antenna, we will have recorded the field strength at each of these points and at different time indexes.

The basic SAR stripmap setup is illustrated in Figure 2.1. The radar transmits a pulse towards the target imaging area. The pulse reflects off of targets in the imaging area. The reflections from an ideal point target can be modeled as a spherical wave, such that the radiated pulse will have the same phase when measured at any two equidistant locations from the point target.

![Figure 2.1](image)

**Figure 2.1.** A single transmission of a pulse occurs. When it reaches the point target, the pulse is reflected off in every direction in a spherical pattern.

When the reflections return to the radar array, each antenna will collect a different set of samples. This is due to the spatial location of each antenna being
different, receiving the reflection at a different time. The SAR system must then
locate the target based on the reflections of the signal. Since each antenna in the
array is recording over time, the time when a single locus of the spherical wave (i.e.
a particular circle from Figure 2.1) passes through the antenna will be recorded.
By analyzing these samples and reconstructing the circle and its radius, the point
target’s location can be found. This technique is called wavefront reconstruction
[3], as it attempts to construct the wavefront shape and the relative phases by using
spatial and temporal diversity. Wavefront reconstruction is the underlying theory
behind the majority of SAR signal processing theory.

2.2 Problem Setup

Figure 2.2 shows the problem setup. The imageable area consists of a square area
of size \( n \times r \). In the stripmap configuration, the SAR aerial vehicle travels down a
single axis along the edge of the imageable area. The axis parallel to the airplane’s
movement is called the “Cross-range”, and the axis perpendicular to the airplane’s
movement is called the “Down-range”, or simply “range”.

![Figure 2.2](image)

**Figure 2.2.** An airplane travelling along a fixed axis. It illuminates targets in an
imaging area adjacent to the direction of travel.

The aerial vehicle stops at \( n \) many points, thus simulating an array of \( n \) antennas.
At each point, it collects \( m \) samples into a vector

\[
D_k = \{D_{k,1}, D_{k,2}, \ldots, D_{k,m}\}, 0 < k < n
\]

where \( k \) is the point the SAR platform took data. The notation \( D_{X,Y} \) represents
the sample taken at position \( X \) along the path of travel and at time \( c\Delta T \) after the
beginning of recording at \( X \). This model approximates the propagation velocity as
the speed of light. Each sample is separated by \( c\Delta T \), making our sampling rate

\[
S_{rate} = 1/\Delta T
\]
The radar transmits a pulse beginning at time $t = 0$, which propagates out towards the targets. The pulse can be defined as a discrete set of samples

$$\mathbf{R} = \{R_1, R_2, \ldots, R_{L_{ref}}\}, \quad 0 < k < L_{ref}$$

(2.3)

Where $L_{ref}$ is the length of the reference pulse. For convenience, we set the sample rate of $\mathbf{R}$ to be $S_{rate}$. The exact values in $\mathbf{R}$ will depend on the radar type being used, and is discussed in Section 2.3.

2.2.1 Single Target, Noiseless Scenario

For a single target with no noise, there will be only one reflection for a given pulse. When recording data at a particular location, the data will consist of a single delayed version of $\mathbf{R}$:

$$D^*_{k,l} = R_k - \phi(k) + l$$

(2.4)

Where $\phi(k)$ is a function that represents the time delay of the reference pulse in samples due to the sample being collected at position $k$. $\phi(k)$ is a function of the coordinates of the target. The time delay is equal to the distance between the target and the radar scaled by the pulse speed. For a target at point $T_x, T_y$, from Figure 2.2 the distance to the target is

$$d(x) = \sqrt{(x - T_x)^2 + T_y^2}$$

(2.5)

Where $x$ is the $x$-coordinate of the SAR platform. Thus the reference pulse delay is

$$\phi(k) = \frac{2d(k\Delta T)}{c}$$

(2.6)

2.2.2 Multiple Target, Noiseless Scenario

For multiple targets, there will be one reflection of $\mathbf{R}$ for each target in the imaging area. Thus if we are tracking $T$ targets, the recorded data $D^{**}$ will be defined as

$$D^{**}_{k,l} = \sum_{i=1}^{T} R_{k - \tilde{\phi}(i,k) + l}$$

(2.7)

Where $\tilde{\phi}(i,k)$ is the delay in samples of the reference signal due to the $i$th target, when data is collected at position $k$. We can find $\tilde{\phi}$ using the same technique as in the previous section, yielding

$$\tilde{\phi}(i,k) = \frac{2d_i(k\Delta T)}{c}$$

(2.8)

Where $d_i(x)$ is defined the same way as Eq. (2.5) but uses the $T_x$ and $T_y$ coordinates of the $i$th target.
2.2.3 Noisy Environments

The model used in noisy simulations will be identical to those in previous sections, with the addition of noise vectors at each point. Thus we will instead have $D_{k,l} + \beta_{k,l}$, where $\beta_k$ is a vector of random data sampled from the channel noise model. For simulations in this thesis, the model used is the standard normal distribution with probability density function

$$N_{\beta}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x - u)^2}{2\sigma^2}\right)$$

(2.9)

with $\mu = 0$ and $\sigma$ defined in the individual simulation.

2.3 Radar Waveform Design

The type of waveform transmitted by the radar has a great impact on the SAR imaging algorithm’s performance. It affects the range resolution, cross-range resolution, resistance to noise, jammability, inter-target interference (ITI) resilience, multipath interference, among many other benchmarks of the system’s performance. Typically, a transmit pulse is chosen based on the ease of generating the signal as well as its performance in benchmarks critical to the radar application.

In this section, we will briefly discuss the critical benchmarks necessary for our radar system, and highlight several candidate waveforms for consideration. Lastly, we will discuss the motivation behind the modulation used throughout our simulations.

2.3.1 Radar Waveform Benchmarks

For the purposes of a high-resolution radar system, we are principally interested in tests which estimate its resolution in both the cross-range and range. The range resolution provided by a radar system is directly proportional to its bandwidth. The cross range resolution is more complex as it depends on a number of factors. In conventional SAR, it depends indirectly on the sidelobe width of the transmit pulse’s autocorrelation function, the pulse’s phase history auto-correlation function, the sampling rate of the receiver, and on the center frequency the pulse’s bandwidth [3].

Beyond resolution, there are many other benchmarks which are critical for any radar’s successful operation. For example, the radar must be able to operate in noisy environments. The noise resilience of a transmit pulse is characterized by comparing the waveform’s cross correlation with the noise model used in the system’s simulation. Other benchmarks, such as jammability, ITI, and multipath are defined by more complex interactions of adversarial algorithms and environmental effects.

2.3.2 LFM

Linear Frequency Modulation (LFM), or “chirp” modulation, is standard radar pulse used in conventional radar.

The LFM signal is defined as
\[ P_{\text{LFM}}(t) = E_{\text{LFM}} \sin(t(C_{\text{LFM}} t + \omega_c)), \quad 0 < t < L_{\text{LFM}} \quad (2.10) \]

where \( E_{\text{LFM}} \) is the envelope factor of the signal, \( C_{\text{LFM}} \) is the modulation rate, and \( L_{\text{LFM}} \) is the window length. The bandwidth of the signal depends directly on \( L_{\text{LFM}} \). An example LFM signal is shown in Figure 2.3. It consists of a windowed FM signal with a linear modulation signal. LFM is easily generated, however due to its linear modulation, it is easily predicted and thus more susceptible to jamming than more sophisticated techniques such as frequency hopping. Conversely, a received reflection from an LFM signal is resilient to Doppler shift. This is due to its linearly modulated frequency, which causes a narrow correlation spike with the frequencies that match up with the dopper shifted version. It also allows for the reduction of frequency dependent loss, as only narrow portions of the signal will be affected by narrow band noise.

![Figure 2.3. An LFM signal. The frequency of the sinusoid increases linearly.](image)

### 2.3.3 Gaussian Pulse

Gaussian Pulse modulation is defined as

\[ E_{\mu} \exp \left( -\frac{x^2 - \mu^2}{2\sigma^2} \right) \quad (2.11) \]

Where \( E_{\mu} \) is the envelope factor. The Gaussian pulse is often used in radar as it is easy to generate a crude approximation to the gaussian kernel in hardware [7]. Additionally, the pulse is short ranged, giving its autocorrelation function a narrow mainlobe width. The pulse is shown in Figure 2.4. The bandwidth of the pulse is inversely proportional to \( \sigma \). This allows the radar designer to directly set the bandwidth desired for the application. One disadvantage of a simple gaussian-shaped pulse is that it is also easy to detect and retransmit. This allows for others to jam a radar using simple pulse-based transmissions.
2.3.4 OFDM

The OFDM equivalent baseband signal is defined as

\[ P_{\text{baseband}}(t) = \sum_{k=0}^{N_c} E^{(k)}_{\text{OFDM}} e^{j\omega_o k t}, 0 < t < L_{\text{OFDM}} \]  

(2.12)

Where \( E_{\text{OFDM}} \) is a vector of length \( N_c \), and \( N_c \) is the number of sub-channels defined in the OFDM pulse, and \( \omega_o \) is the spectral resolution. We can express \( \omega_o \) as \( 1/L_{\text{OFDM}} \). The transmitted signal is then

\[ P_{\text{OFDM}}(t) = \sin(\omega_c t) \Re\{P_{\text{baseband}}(t)\}. \]

(2.13)

Figure 2.5 illustrates an OFDM pulse. Each channel is a sinc function which shares bandwidth with adjacent channels. However, at the peak of a sinc function, all the other sinc functions have a value of zero (orthogonal). This means by measuring at the exact point where the sinc has a peak, you will sample only that channel. In this way you can detect energy from that channel.
OFDM allows the radar to encode the set of $E_{OFDM}$ into the pulse arbitrarily. If we let $E_{OFDM}$ be a vector of bits, we can use the OFDM pulse for data transmission. In this mode, $E_{OFDM}$ is a time-dependent vector, with multiple instances of $P_{OFDM}$ being generated and transmitted sequentially. Each instance would contain a different set of bits within $E_{OFDM}$, thereby transmitting data over time. This property is useful as it would allow an OFDM-based radar to operate as both a communications device and a radar, creating a multi-function radar. In addition, because OFDM allows us to choose which frequencies are in use (by setting $E_{OFDM}$), we can use frequency hopping techniques for anti-jamming. It also has a user definable bandwidth, limited only by the sampling rate of the waveform generator.

2.3.5 Chosen Pulse

The radar pulse used throughout the rest of this thesis is the triangle wave:

$$P_T(t) = \begin{cases} 1 - |t| & |t| < P_W \\ 0 & \text{otherwise} \end{cases}$$

(2.14)

where $P_W$ is the pulse width in seconds. The pulse used for simulations in this thesis has a width $P_W = 50\text{ns}$. It is shown in Figure 2.6. This waveform was chosen for three reasons. Firstly, it is an approximation for a Gaussian function. Secondly, after sampling $P_T(t)$ can be written as a wideband OFDM waveform by taking its Fourier components and mapping them onto the OFDM sub-carriers. Thirdly, the triangle wave has a simple spectrum which is easy to characterize and analyze. Thus we can use this pulse as an ideal approximation for an OFDM or Gaussian pulse.

![Figure 2.6. The reference triangle wave used for simulations. Time t=0 marks the beginning of transmission.](image)

2.4 Previous Work

Conventional SAR signal processing breaks the task of locating the target into two tasks: Finding the range of targets at each SAR platform location, and finding their cross-range location across all locations. This method is outlined in more detail in [3].
2.4.1 Range - Conventional Method

First we must find the range to the target in a single cross-range bin. Recall that the bin \( D_k \) is a vector of all the samples collected at point \( k \). Since we stop at \( n \) many points, we know \( 0 < k < n \). We use a matched filter between each bin and the reference pulse, yielding

\[
C_k = R \ast D_k, 0 < k < n
\]  

(2.15)

Where \( \ast \) represents the circular correlation in the time-domain. Each vector \( C_k \) will have length \( m \), making the total data set size \( m \times n \). For convenience we will define some additional notation. \( V_C \) denotes the set of all cross-correlated range vectors, as follows:

\[
V_C = \{ C_1, C_2, \ldots, C_n \}
\]  

(2.16)

We will also define the notation \( C_{k,l} \) to be the \( l \)th sample in the vector \( C_k \), as follows:

\[
C_k = \{ C_{k,1}, C_{k,2}, \ldots, C_{k,m} \}, 0 < k < n
\]  

(2.17)

In order to calculate the target range, we calculate the index of the max of the corresponding range correlation:

\[
\Gamma_k = \text{IndexOf}(\max_l C_{k,l}), 0 < k < n, 0 < l < m
\]  

(2.18)

The maximum peak of the matched filter is the best correlation between the data and the reference pulse. Since the domain of the matched filter is the phase shift between the signals, the quantity \( \Gamma_k \) corresponds to the delay in samples between the reference pulse and the echoed signal. Thus \( c \Delta T \Gamma_k \) is the range to the target from position \( k \). This result is easily extended to the multiple target case by looking for \( T \) many local maximums instead of one, for \( T \) targets.

2.4.2 Cross-Range - Conventional SAR Wavefront Reconstruction

SAR wavefront reconstruction cross-correlates the raw data collected with the known phase history signature of a single point target. The FFT is performed individually on each vector \( C_1, C_2, \ldots, C_n \). A frequency \( \omega_H \) is chosen, which is a frequency that the reference pulse has high energy at. The frequency component at \( \omega_H \) is extracted from the spectra of each vector in the raw data set. The phase angle of this component is calculated for each vector, forming the phase history \( \Phi[0], \Phi[1], \ldots, \Phi[n] \), where \( \Phi[k] \) is the phase angle of the \( \omega_H \) component of the \( k \)th bin. Thus we have

\[
\Phi[k] = \angle\{\text{FFT}(D_k)_{\omega_H}\}, 0 < k < n
\]  

(2.19)

The set of angles \( \Phi[0], \Phi[1], \ldots, \Phi[n] \) is known as the phase history of the target scene. By the Fourier time-shift property, the angle \( \Phi[k] \) corresponds to the time-delay at which we received the reference signal at position \( k \), as \( D_k \) is collected at position \( k \). Figure 2.7 shows the phase history of target at \( T_x = 500, T_y = 200 \) with a transmitted triangle wave. \( \Phi[k] \) has a wide mainlobe centered around the
cross-range position of the target. It is therefore indicative of the embedded target’s cross range location. These angles are correlated with the known phase-response $\Phi_{ref}[k, p_{ref}]$ of a simulated target at position $p_{ref}$. A high correlation peak indicates a target with the same cross-range position as the simulated target.

![Figure 2.7](image.png)

**Figure 2.7.** Phase history of a target at x=500 and y=200. The large mainlobe is centered around the x coordinate of the target. By correlating phase angles recovered against ideal phase angle of a target at a particular x coordinate, we can recover the target cross-range coordinate.

Note that the set of correlated vectors $V_C$ is used instead of the raw data in conventional cross-range SAR. This is because the correlation of the input data with the reference pulse increases the SNR of the embedded reflections in the presence of noise. This technique is called a matched filter, and is a fundamental building block of statistical signal processing.

### 2.4.3 Wavefront Reconstruction Asymptotic Complexity

Wavefront reconstruction consists of two operations: the Fourier transform of each of the $D$ vectors, and the correlation of the phase history extracted from the Fourier transforms. The correlation is usually also performed via an FFT and IFFT for performance reasons. The running time of the correlation is

$$\theta(n \log n)$$

for a single phase history. Note that the notation “$f(x)$ is $\theta(g(x))$” implies the following relation:

$$\exists C_1, C_2, x_0 \text{ s.t. } \forall x > x_0, |C_1 g(x)| < |f(x)| < |C_2 g(x)|$$

That is, after some point $x_0$, $f(x)$ is bound above and below by $g(x)$ scaled by a constant. Then the computation of each phase bin $\Phi$ requires computing the Fourier transform of all $D$ vectors; this corresponds to an FFT of $m$ points, or $\theta(m \log m)$ computation time. There are $n \Phi$ bins, one corresponding to each range bin. Calculation of all $\phi$’s takes $\theta(n \log n)$ for a total running time of

$$\theta(n \log n + n \ m \log m)$$
2.4.4 Wavefront Reconstruction Cross-Range Resolution

In this section we will derive the cross-range resolution obtained by the wavefront reconstruction method.

Radar resolution is a measure of how close targets ("peaks") can be before they become indistinguishable from each other. The local search for peaks in $\Gamma_k$ will interfere with each other if the peaks are not well isolated from each other. For cross-range resolution, we want to distinguish two targets which are separated on the cross-range axis, but share the same range.

In order to simplify our analysis, we will analyze the underlying continuous version of the problem. This analysis will still apply to the sampled version of the problem, but we remove some of the ambiguity caused by the sampling process; effectively, our sampling speed will be infinite. Thus this analysis will derive the ideal resolution, drawing an upper bound on the resolution obtainable by any wavefront reconstruction algorithm implementation.

We begin by defining continuous-variable counterparts for our discrete variables. From Section 2.3.5, the continuous-time reference pulse we will use is:

$$ r(t) = \begin{cases} 1 - |t|, & |t| < P_W \\ 0, & \text{otherwise} \end{cases} \quad (2.23) $$

Note that $R$ used in the discrete algorithm is a sampled version of $r(t)$:

$$ R_k = r \left( \frac{tk}{F_s} \right) \quad (2.24) $$

We define $v_g(k, l)$ to be the continuous counterpart of $D_{k,l}$, where $k$ is the plane’s position and $l$ represents the time domain (the time-domain data collected at $k$). We know from Eq. (2.7) that the data in a particular bin is just the sum of delayed versions of the reference pulse, and the delay is due to the position of targets. We similarly define the continuous-time data collected at position $k$ to be

$$ v_g(k, t) = \sum_{i=1}^{N_T} r(t - \tau(k, i)) \quad (2.25) $$

where $\tau(k, i)$ is the delay caused by the $i$th target when the plane is at position $k$. For cross-range resolution, we are only interested in the two target case, so we will set $N_T = 2$, and define $\tau_1(k) = \tau(k, 1)$, $\tau_2(k) = \tau(k, 2)$ to be the time delay due to the first and second target respectively. We can then write $v_g(k, t)$ as

$$ v_g(k, t) = r(t - \tau_1(k)) + r(t - \tau_2(k)) \quad (2.26) $$

We now begin our analysis of the cross-range resolution of wavefront reconstruction. The method takes the Fourier transform of a each cross-range bin, and extracts the frequency component at a chosen frequency, $\omega_H$. In the continuous case, we must take the Fourier transform of $v_g(k, t)$. Then the phase history is extracted from $V_g(k, \omega)$. The same process is done on a reference signal, extracting a reference phase history from its Fourier transform. Finally, the two phase histories are correlated and the resulting correlation is used to resolve targets in the cross-range.
We start by defining a continuous-time reference signal

\[ v_I(k, t) = r(t - \tau_I(k)) \]  

(2.27)

where \( \tau_I(k) \) is the time delay to the reference target that generates a reflection in the reference signal. Typically, the reference target is located in the middle of the imageable area, as the mid-swath target will correlate well with all targets within the imaging area [3]. Next, we need to perform the Fourier transform on \( v_I(k, t) \). First we rewrite \( r(t) \) as

\[ r(t) = \text{rect}\left(\frac{t}{P_W}\right) \ast \text{rect}\left(\frac{t}{P_W}\right) \]  

(2.28)

where

\[ \text{rect}(t) = \begin{cases} 
1, & |t| < 1 \\
0, & \text{otherwise} 
\end{cases} \]  

(2.29)

and \( \ast \) is the convolution operator. In this section we will use the uppercase convention to denote the Fourier transform (i.e. \( F\{r(t)\} = R(\omega) \)). The Fourier transform of our reference is then:

\[ R(\omega) = F\left\{ \text{rect}\left(\frac{t}{P_W}\right) \ast \text{rect}\left(\frac{t}{P_W}\right) \right\} \]  

(2.30)

By the convolution theorem and the Fourier transform of \( \text{rect} \), we have

\[ R(\omega) = |P_W| \text{sinc}^2\left(\frac{\omega P_W}{2\pi}\right) \]  

(2.31)

We can now find the Fourier transform of \( v_I(k, t) \). Using Eq.(2.27), we have

\[ V_I(k, \omega) = R(\omega) \exp(j\omega \tau_I(k)) = |P_W| \text{sinc}^2\left(\frac{\omega P_W}{2\pi}\right) \exp(j\omega \tau_I(k)) \]  

(2.32)

We extract the complex phase history from a frequency of high energy. This can be chosen arbitrarily, however we see that at a max occurs at max \( \{ \text{sinc}^2\left(\frac{\omega P_W}{2\pi}\right) \} \), or around DC. We define the chosen frequency to be \( \omega_H \). Then our complex phase history is:

\[ \Phi_I(k) = V_I(k, \omega_H) \]  

(2.33)

We wish to correlate this phase history with a phase history extracted from \( v_g(k, t) \). We compute the Fourier transform of \( v_g(k, t) \) to be

\[ V_g(k, \omega) = R(\omega) \exp(j\omega \tau_1(k)) + R(\omega) \exp(j\omega \tau_2(k)) \]  

(2.34)

with complex phase history

\[ \Phi_g(k) = V_g(k, \omega_H) \]  

(2.35)

Our phase history correlation is then
\[ H_C(k) = \Phi_g(k) * \Phi_I(k) = \int_0^{k_{\text{max}}} \Phi_g(\hat{o})\Phi_I(k-\hat{o})\delta\hat{o} \]  

where the plane moves along the interval \([0, k_{\text{max}}]\). Plugging in for the phase histories, this becomes:

\[
H_C(k) = \int_0^{k_{\text{max}}} \left[ R(\omega_H)\exp(j\omega_H\tau_1(\hat{o})) + R(\omega_H)\exp(j\omega_H\tau_2(\hat{o})) \right] \\
\left[ R(\omega_H)\exp(j\omega_H\tau_I(k-\hat{o})) \right] \delta\hat{o} 
\]

(2.37)

We see the two cross terms will cause peaks in \(H_C\) when \(\tau_I(k-\hat{o}) = \tau_1(\hat{o})\) and \(\tau_I(k-\hat{o}) = \tau_2(\hat{o})\). \(\tau_1\) and \(\tau_2\) are the delay due to the cross-range and range locations of targets one and two respectively. This means that the peaks which occur in \(H_C\) are a function of the target locations. Due to the linearity of phase history shifting [3], we know the values of \(k\) with strong peaks are also the cross-range and range locations of the target; a target shifted by an amount \(\hat{s}\) in the cross range domain will cause a shift of its phase history of \(\hat{s}\). Figure 2.8 illustrates this property. The target at \(x=600\) has the same phase history as the target at \(x=500\), with the only different being a time shift. The combined phase history of two targets still retains peaks at both of the positions of the targets. Note that the noise at the beginning of Figure 2.8 (bottom) is due to no pulses being within our imaging area for the first few sub-bins. The recorded angle in that case is the angle of the noise, which is random.

![Figure 2.8](image)

**Figure 2.8.** (top) The phase history of two targets at \(x=500\) and \(x=600\). Both have range \(y=200\). (middle) The phase history of the single target at \(x=500\). (bottom) The phase history of the single target at \(x=600\).

\(H_C\) is the correlation between the two-target phase history and a single target phase history with arbitrary shift. By evaluating \(H_C\) for a particular reference pulse, we can determine the closest distance two targets can be while still having a -3dB
drop between the peaks they generate in $H_C$. Figure 2.9 shows $H_C$ for the setup in Figure 2.8. The two target phase history is correlated against the reference phase history of a target at $x=500$. We notice two peaks, at shift $k = 0$ and $k = 100$. Since our reference target is at $x = 500$, a peak at shift 0 indicates a target at 500. The peak at shift 100 indicates a target at $500 + 100 = 600$. This is exactly where our targets exist, indicating we have found their cross range locations. The resolution obtainable can be found by moving the targets closer until the two peaks converge and there is no -3dB drop between them. The minimum distance that maintains the drop is the cross-range resolution obtainable by a perfect wavefront reconstruction algorithm with no sampling error. This analysis will be used in Section 5 to compare our novel algorithm to this previous approach.

**Figure 2.9.** The cross correlation of the phase history of two targets at $x=500$ and $x=600$ with a reference phase history with $x=500$. 
Chapter 3
SEARCH SPACE ALGORITHMS

3.1 Introduction

One principal design goal of our algorithm is to optimize the running time of our algorithm as a function of the input size, as well as a function of the resolution. During the algorithm design, consideration must be given to find methods which identify an optimal solution while evaluating the least number of candidate solutions. We specifically focus on the usage of local search methods which utilize SAR theory to intelligently search the SAR data.

In this section, we will complete a brief survey of existing search methods which inspired the algorithm development. We will also discuss other methods that were considered but not used in this thesis. Each algorithm in this section is bound by the no free lunch (NFL) principle[8]. This simply states that in terms of a generic solution space, no search algorithm performs “better” than any other, and cannot be directly compared. Instead, we must evaluate the applicability of algorithms to the specific problems encountered in SAR data processing.

3.2 Search Methods

These methods attempt to locate data by searching within a particular subset of the data set. In order to intelligently select the subset, some prior knowledge as to the structure of the data must be used. Each method will begin with an overview of its input and output requirements. This will be followed by a brief description of how the method is usually implemented.

3.2.1 Binary Search

Input:

- A 1 dimensional array \( IN_v[k] \) of length \( L_v \), where the set of \( m \in IN_v \) is well ordered.
- A value \( val \) we are searching for.

Output:

- The point \( Q \) such that \( IN_v[k] \leq val \leq IN_v[k+1] \), if \( k \) exists

A binary search allows for the quick location of an element or near-element in an ordered vector or binary search tree. The ordering requirement can be waived by conducting a sort prior to running the binary search (which incurs the associated cost). The search begins at the middle data point \( L_v/2 \). Let \( S = L_v/2 \) be the current data point, \( B = 0 \) be the beginning of the search space and \( E = L_v \) be the end. If \( IN_v[S] < val \), the search space is cut in half by setting \( B_{\text{next}} = S \) and choosing a new point \( S_{\text{next}} = (B_{\text{ext}} + E)/2 \) as the current data point. On the other hand,
if \( IN_v[S] > val \), the top half of the search space is cut off, so that \( E_{next} = S \) and \( S_{next} = (B + E_{next})/2 \). Another iteration of the program is run, using the \( S_{next}, B_{next}, \) and \( E_{next} \) values for \( S, B, \) and \( E \) as appropriate. This search space halving continues until either \( S_{next} == S \) (the item \( val \) doesn’t exist) or \( IN_v[S] == val \).

### 3.2.2 Feature Extraction

**Input:**

- An image \( IN_v[k][l] \) of raw data

**Output:**

- A data set \( Q \) with smaller dimensionality than \( IN_v[k][l] \) which contains data of interest

Feature extraction is in fact a class of algorithms that perform pattern recognition. In general, they rely on extracting a relatively small data set of useful information from a larger, redundant data set. One example is edge detection, a type of feature extraction where only high frequency components are of interest. Feature extraction can be used to generically describe many search algorithms. A ”search” implies that the total data set is redundant. In particular, by naming the local optimum point of a fitness function the “feature” of interest, all local search techniques can be considered implementation of feature extractions.

The algorithm developed in this study can be considered a form of feature extraction. The SAR raw data consists of the input image, and the output is a particular feature: the trail of reflected pulses in \( V_C \).

### 3.2.3 Local Search

**Input:**

- An input data set \( IN_v[k][l] \) of raw data

**Output:**

- A data point \( \in IN_v[k][l] \)

A local search is a heuristic method for finding solutions in large data sets. Instead of evaluating every point in the input data vector, a heuristic is used to find the supposed approximate locations of a good solution. Then the search algorithm performs a search within the local area, and returns the best solution found. This approach only works for good heuristics; a bad heuristic will potentially leave unexplored a more optimal solution. Often with this method, approximately good data points are sufficient, as it is not likely to find the optimal data point.

Local searches are used extensively in our algorithm as a main source of optimization. By eliminating SAR data points with low probability of containing a reflection, we can prune much of the search space from evaluation.
3.3 Methods Considered for Future Work

This section outlines methods which were considered for usage in the novel method developed in this thesis. In addition to a description similar to other algorithms, a reason for exclusion from the novel method is given. These methods are hoped to be evaluated in future work.

3.3.1 Genetic Algorithms

**Input:** Set of possible solutions \( S_1, S_2, \ldots, S_n \) “Goodness” function \( F(.) \), which returns the goodness of a particular solution \( \in V_1 \ldots V_n \)

**Output:** Returns a solution \( Q \in S_1, \ldots, S_n \) that on average approximates \( \max(F(val)) \)

Genetic algorithms (GAs) are inspired by the way genetic reproduction slowly improves biological life. At first, an initial set of random solutions is generated. Each of these is evaluated for their fitness by the goodness function. New solutions are generated by breeding the initial set solutions with each other. These new solutions contain partial properties of two solutions of the original set, similar to chromosomal reproduction in biology. This process continues in a generational-style, with the each new set of solutions re-evaluated for goodness, and then rebred to form another generation of solutions. Each time the breeding process occurs, more fit solutions are more likely to be chosen, in order to encourage parts of solutions which are causing good results in the goodness function. This process is similar to biological reproduction, where more fit animals are likely to produce offspring, causing the generation of animals as a whole to become more fit.

GAs rely on the assumption that there is a correlation between slices of the solution and the function’s fitness. If there is not a strong correlation, then taking a parts of two good solutions and slicing them into one solution is not likely to produce another good solution. The exact point of slicing a solution is heavily problem-specific and relies on heuristical knowledge of the underlying data. In many cases, slicing in solutions up arbitrarily will not perform effectively, as some data groups are closely related and affect the fitness as a whole. Thus it is the GA designers job to find good points of splitting solution data that maximizes the chances of the next generations fitness being high.

GAs were not used in the algorithm developed in this thesis due to the random nature of its solutions. In order to increase the reliability and dependability of the target solution, only deterministic algorithms were used in the final design.

3.3.2 Slow-time Window Narrowing

**Input:** A 2D data set \( INV[k][l] \) in row-major format. Proximity function \( F(.) \), which returns the proximity of a particular data point to the point we are looking for (with injected noise)

**Output:** Returns a point \( Q = (Q_x, Q_y) \in INV \) that approximates the point where a min occurs in \( F(.) \)
This method involves the windowing of the solution as we become more and more sure of the location of the stable point of a system. At first, two random points are passed into $F()$. Based on the slope between the points, a most likely direction of search is determined, with a wide window of error initially. Additional points are then chosen within the window of search. The likely direction vector is recalculated based on indications of correctness from the new points values (evaluate $F()$). After each iteration, the search window is narrowed, eventually leading to convergence of the vector to a single point, which is returned.

This method was evaluated for inclusion in the Tracker implementation. However, it was found that calculating a vector of travel from the past three points was sufficient to allow previous points to indicate the next point to be chosen when other indicators failed. Thus the window remained small and fixed (three points), and window narrowing was not found to be necessary.

3.3.3 Distribution Fitting

**Input:** A 1D data set $IN_V[k]$ A Distribution description $IN_{DIST}$ which is an enumeration of the type of distribution

**Output:** A confidence value in the likelihood of $IN_V[k]$ being data sampled from the distribution $IN_{DIST}$.

There are many well goodness of fit algorithms which test for a data set’s conformity to a particular distribution[10]. The general method involves looking for features of the data set that are common in all data sets polled from $IN_{DIST}$.

Distribution fitting methods were not used in this thesis due to the complexity they would add on to the calculation of resolution and running time. Instead, a more simple approach involving following the path of the line (signal) was used. This simplified both implementation and analysis. In future work, using distribution fitting may be explored.

20
Chapter 4

NOVEL SPARSE TARGET SAR ALGORITHM

4.1 Introduction

In this section we develop a new signal processing method to compute the cross-range location of targets. The designed component will be a drop-in replacement for the cross-range processing method discussed in Section 2.4.2. The main design goal of our algorithm is to utilize a local search of the raw SAR data, to find the cross-range position efficiently. The successfulness of this approach will rely on carefully choosing the area for the local search to operate in. As will be seen in Section 5.4, the asymptotic complexity (running time) of this algorithm is faster than wavefront reconstruction for many SAR scenarios.

We consider the set of raw data $D_1, D_2, \ldots, D_m$ to have been previous collected and ready for processing in an input buffer. For convenience, we will define the 2D array

$$D[k][l] = D_{k,l}, 0 < k < m, 0 < l < n$$

as our input data set. From Section 2.2, we see that the first index length will always be $m$ and the second index will be $n$. Our input data is thus $m \times n$ in size.

In order to motivate our algorithm’s design, we will consider the single target noiseless scenario. A sample setup is shown in Figure 4.1. Initially, the distance to target is large. As the plane travels, the distance becomes shorter. The shortest distance to the target will be when the y-coordinate of the plane is equal to the y-coordinate of the target. As the plane continues moving, the distance to target increases once again. These properties can also be seen by examining the distance equation in Eq. (2.5).

Figure 4.1. An airplane recording reflected signals from a target. The plane flies closer to the target until it is aligned, and then flies away from it.
From Eq. (2.4), we expect the data collected will consist of time shifted versions of the reference pulse. Figure 4.2 illustrates the set of raw data collected from this scenario with a Gaussian reference pulse. We see that each vector consists of a Gaussian, shifted in accordance with the distance function as expected. The shortest distance to the target will be the plane and the target have aligned. By finding the bin with the minimum pulse delay, we know the target’s cross-range location is the same as ours, and we can record the target’s cross-range location.

![Figure 4.2](image)

**Figure 4.2.** An illustration of the input data vector for a single target. Each bin contains a single reflection of the reference pulse. The pulse is time delayed based on the distance.

We will now define this relationship formally. From Eq. (2.6), we see that

$$\min_{k}(\phi(k))$$

occurs at

$$\min_{k}\left(\frac{2d(k\Delta T)}{c}\right)$$ (4.2)

Since $\Delta T$ is a constant, we wish to find $\min_{k}(d(k))$. From Eq. (2.5), we see that

$$\min_{x}(d(x)) = \min_{x}(\sqrt{(x - T_x)^2 + T_y^2})$$ (4.3)

occurs when $x = T_x$. Thus the minimum phase shift $\phi(k)$ occurs when the planes x-coordinate $x$ equals the targets x-coordinate $T_x$. From this analysis we conclude that by finding the location of $\min_{k}(\phi(k))$, we will have found the target’s cross-range position (equal to our own).

One primary concern during algorithm design is the affect of noisy environments on our performance. Ambient noise is a major limiting factor for radar systems[3]. In 2.4.2, we noted that conventional SAR first performs a circular correlation (matched filter) between the reference pulse and the input vectors. Instead of manipulating the raw data, the data set $V_C$ is used as an input to the signal processing algorithm.

It is a basic tenet of statistical signal processing that the matched filter is an ideal method to extract the existence and time-delay of a signal which is otherwise obscured by a white-noise environment[9]. Our algorithm must operate in noisy environments, and so we cannot rid ourselves of the computational burden of calculating and using the output of the matched filter.
The property shown in Eq. (4.3) also applies to $V_C$. For any reference signal $R$, its auto-correlation mainlobe peak occurs at time delay $\delta = 0$. This means that a signal correlated with a delayed version of itself will have its correlation mainlobe delayed proportional to the signal delay. Formally, we have

$$f(t) * f(t - \Delta) = f(t - \Delta) * f(t) \quad (4.4)$$

By finding the min phase delay within $V_C$ we can also find $\min_k(\phi(k; 0))$, and solve for the target’s position.

The calculation of $V_C$ imposes heavy computational burden. Our goal will be to choose to only evaluate and compute specific points of the $V_C$ set which are needed to find the min peak. This will allow us to reduce the amount of computation required.

4.2 Algorithm Overview

In this section we will construct a local search algorithm that locates paths of pulse reflections in $V_C$ and determines the minimum phase delay. This approach differs from the one outlined in Section 2.4.2. In that method, the entire $V_C$ space was computed, and correlated against a signature of pulses; the result of the correlation indicated the location of the minimum phase delay. In our method, we will conduct a small search of the space $V_C$ and directly locate the minimum point.

Each point in $V_C$ will be unevaluated initially. Whenever a value $C_{k,l} \in V_C$ is needed, it will be evaluated by performing the single-point correlation. From Eq. (2.15), the single-point computation is

$$C_{k,l} = \sum_{i=0}^{n} R_{i+l} D_{k,i} \quad (4.5)$$

Each point takes $n$ multiplications to compute. The goal is to locate a path of pulses within $V_C$, such that the delay of the pulse in bin $k$ is found at $C_{k,d(k\Delta t)/c}$, for all bins $0 < k < n$. Once a qualifying path is found, we will find the bin $k$ where $d(k\Delta t)/c$ is minimized. This process is repeated for multiple paths found within $V_C$.

Our solution consists of two main components: the acquisition/detection of a path of pulses, and the location of the minimum phase delay of the path. These two components are illustrated in Figure 4.3, and described in the next two sub-sections.

4.2.1 Path Acquisition

The first phase for our algorithm involves detecting how many targets exist within our input data. To find a path, we first compute all points in the first bin $C_1$. Since we are doing the full correlation, we can use the FFT to compute Eq. (2.15) for a $\theta(m \log m)$ running time. Using threshold detection, peaks within $C_1$ are isolated and the number of targets is estimated based on the number of peaks found. In particular, a -3dB threshold is used between the mainlobes of peaks in order to distinguish them as two separate paths. Let $T$ be the number of reflections discovered during in $C_1$, at locations $C_{1,x_1}, C_{1,x_2}, \ldots, C_{1,x_T}$. The set $X_1, X_2, \ldots, X_T$ denotes
Figure 4.3. Overview of the components of the novel cross-range algorithm. The left flowchart represents the path acquisition, and the right flowchart represents the path tracking loop.

the locations of the peaks in $C_1$. This set constitutes the set of peaks found. For each peak, we spawn a Tracker object initialized to the location of the peak.

4.2.2 Path Tracking

The $T$ trackers are constructed and given the pulse locations $X_1, X_2, \ldots, X_T$. Each tracker looks in the next bin, $C_2$, for a continuation of the path. By bounding the potential location of the pulse in the next bin, we can perform a local search for the path without evaluating many points of $C_2$. This is illustrated in Figure 4.2 as only searching in the shaded box area for the pulse, and leaving unevaluated the zeroed area with no pulses within.

The amount of optimization we can perform in this section depends on how much we can bound the region in which the next pulse resides. From Section 2.2, it is evident that

$$\Delta T_{mc} = \sqrt{x_2^2 + y_2^2}$$

is a necessary condition to recover reflections from the entire square imaging area. We want to bound the change in value of $\phi$ from one bin to the next. Thus we have

$$|\phi(k, 0) - \phi(k + 1, 0)| = \left| \frac{2d(k\Delta T) - 2d((k + 1)\Delta T)}{c} \right|$$

Physically, the maximum relative distance a stationary target can move from one bin to the next is the actual distance the plane moved. Since the plane moves $\Delta x$ between each bin, the distance a target moved is bounded also by $\Delta x$. Converting distance into time delay in samples we have

$$M = \max \left( \left| \frac{2d(k\Delta T) - 2d((k + 1)\Delta T)}{c} \right| \right) = \left[ \frac{\Delta x}{\Delta T c} \right]$$
Where the ceiling is due to our inability to compute partial samples. From the problem definition and Eq. (4.6) we can substitute in

\[ \Delta x = \frac{x_2}{n}, \quad \Delta T = \frac{\sqrt{x_2^2 + y_2^2}}{mc} \] (4.9)

which yields

\[ M = \left[ \frac{x_2}{n} \right] = \left[ \frac{x_2m}{n\sqrt{x_2^2 + y_2^2}} \right] \] (4.10)

giving us the bound on the pulse delay difference between adjacent bins. We can see that if \( n > m \), then the term goes to 1. This means the maximum shift up or down is one. We then have three cases: The phase shift didn’t change samples, it moved up one, or it moved down one. This process is illustrated in Figure 4.4. The shaded area constitutes the samples of \( C_v \) that have been evaluated.

This search method allows us to evaluate a constant number of samples per bin. This will greatly reduce our runtime as seen in Section 5.4. After the max peak is found in the adjacent bin, the current bin number is incremented and the next bin is searched in the same fashion. This process is repeated until the end of the data set (there are no more bins).

If no peak is found in the adjacent bin (i.e. the magnitude of all bins is below a threshold value \( T_B \)), the algorithm will assume it has lost the path, and will perform a traceback. In this situation, it will move back one bin and attempt to choose another peak. If no other peak is found in the previous bin (i.e. all the alternatives are also below \( T_B \)), it will move back another bin and repeat the process until an alternate peak is found. The tracker will then continue normal operation, setting the alternate peak as the current bin/index.

Alternatively, if all evaluated samples in a bin are high (above 1.5 * \( T_B \)), we enter saturation mode. In this mode, we assume there is a large amount of localized distortion, due to either environmental noise or colliding paths. We can no longer assume the peaks are due to our path. Instead, we look at the direction of our previous travel, averaging our last 5 movements to form vector of travel. While in saturation, we will choose peaks that allow us to continue to travel in the same direction as before entering the saturation region. This will continue until a non saturated region is entered.
If we know the path in the adjacent bin can move by a max of 1, we only have to search three samples per bin.

### 4.3 Alternate Setups

Efficient algorithms are most important in time-constrained environments as well as setups where a large amount of input data is expected. We consider the effectiveness of our algorithm in two such scenarios which extend the original SAR spotlight configuration.

#### 4.3.1 Real-time Constrained

In this section, we consider the processing of the raw data as it arrives. Instead of the assumption that the data resides in a fixed buffer ready for processing, we will assume the data arrives in a stream as the radar receives it, allowing us to begin processing before the entire data set is collected.

The main disadvantage to this scenario is that we do not have all of the information available to use. We must instead use the data available to us to the full extent we are able, until more arrives. We consider the case where using the processor actively at all times is preferential to letting it idle, as long as we are gaining from it. In some battery constrained situations such as sensor networks, this may not be the case.

The initial setup is shown in Table 4.1.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_r$</td>
<td>Processor Speed (Hz)</td>
</tr>
<tr>
<td>$M_c$</td>
<td>Multiplication Cost (Hz)</td>
</tr>
<tr>
<td>$RTT$</td>
<td>Round Trip Delay to Target (s)</td>
</tr>
<tr>
<td>$</td>
<td>P_{ref}</td>
</tr>
</tbody>
</table>

**Table 4.1. Parameters for real-time setup.**

The amount of point correlations we can do per bin is

![Figure 4.4. Searching the correlation space with bounded phase shift.](image)
$T_{comp} = (2RTT + |P_{ref}|) \frac{P_r}{M_c} \frac{1}{m}$ \hfill (4.11)

If $m > n$ is true, then our constant bound from 4.2.2 holds. We then need 3 point correlations per bin. If $T_{comp} > 2$, we have sufficient time to implement the standard algorithm defined in the previous section. If not, there is not time available in real time to analyze 3 points per bin. Instead, we must perform a sparse search, skipping bins and looking for embedded patterns that will locate min phase delay peak.

An initial vector path is created by sampling $C_1, C_2, C_3$ using the tracking algorithm, and determining the slope of the line being tracked. This slope is then followed via a binary search, looking for the lack of a peak along the line. This process is illustrated in Figure 4.5.

The binary search is susceptible to noise-induced error in the initial vector direction. Thus as time permits, extra samples will be taken near e.g. points 5&6 in the figure, to determine if we have passed the peak or our initial vector led us off of the path.

![Figure 4.5. Binary search along initial vector path.](image)
Chapter 5

Analysis of Sparse Target SAR Algorithm

In this section, we will simulate and analyze test data for benchmarking the algorithm’s performance in different environments. For the data collected in this section, the following parameters were used.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>2000</td>
</tr>
<tr>
<td>m</td>
<td>4000</td>
</tr>
<tr>
<td>y2</td>
<td>1000m</td>
</tr>
<tr>
<td>x2</td>
<td>1000m</td>
</tr>
<tr>
<td>Δx</td>
<td>0.5m</td>
</tr>
<tr>
<td>ΔT</td>
<td>1ns</td>
</tr>
<tr>
<td>$P_W$</td>
<td>50ns</td>
</tr>
<tr>
<td>Sample Rate $S_r$</td>
<td>1GHz</td>
</tr>
</tbody>
</table>

Table 5.1. Table of parameters used for simulations.

The test pulse used in these results was a ramp function generated by OFDM interpolation. The $-3dB$ width of the pulse was chosen to be 50 samples wide. This effectively simulates a wideband signal as well as gaussian-pulse shaped signals. To ensure the pulse generation was realistic, during generation it was passed through a low pass filter to remove high frequency components.

5.1 Single Target Case

5.1.1 Noiseless

This is the most simple case to analyze so it is considered first. Figure 5.1 shows $C_v$ with a simulated target at $T_x = 700m, T_y = 500m$. The horizontal axis shows the samples collected at a particular cross-range bin, and the vertical axis shows the cross-range location of the bin.

This test case is used to validate the algorithm’s performance. In all tests run, the tracker never became lost or reported a center significantly far from the actual target location. Since there is no randomness, the results for a single target are unchanging. The experimental result for the target’s location was no more than 2 samples off in all simulations. This is due to the unrealizable ideal channel and targets used in simulations in this section.
5.1.2 Noisy

Figure 5.2 demonstrates $V_C$ in white noise. The reference signal is normalized to $[0, 1]$, and the white noise is drawn from a distribution with $\mu = 0$ and $SNR = 5\text{dB}$. Since $V_C$ is the correlation with the reference pulse, we still see a strong, visible path generated by the target. Figure 5.3 shows the same target in noise with $SNR = 0\text{dB}$. We see that the correlation does not filter the noise as well.

Figure 5.2. Single target with noise vector added in. SNR = 5 dB.
5.2 Multiple Target Case

5.2.1 Range

Since the algorithm is designed to replace the SAR cross-range algorithm, in this section we are principally concerned with eliminating interference from targets in different range bins. We must be able to discern targets from each other when they have the same cross-range coordinates but are at different ranges. This case is necessary to consider, as targets in different range bins may cause trackers to behave incorrectly. When performing cross-range measurements, we will need to filter out paths at different ranges as well as possible.

Figure 5.4 shows multiple targets with different ranges. We see that the shape of the path varies based on the distance of the target from the radar platform’s line of travel. This is evident in Eq. (2.5): when $T_y$ is small, the distance equation is approximately linear. This near-effect means that the radar will not behave in the same way for close targets as near targets. Also notice that the slope of the path (change in $\Phi$) is larger for closer targets. However, in 2.6 we accounted for the max slope of all possible targets in the imaging area, including those lying on the axis. This is the origin of the bound set of $\phi$.

In the multiple target scenarios we also must consider paths which intersect. Consider Figure 5.5, which shows two targets with $T_x = 150$ and $T_x = 0$ respectively. In this case not until very near to the min of the path do the two paths differ. The naive algorithm implementation cannot distinguish between these two targets, and thus only one Tracker object will be constructed for the path.

One solution is to have the tracker object detect unusually large mainlobes of the correlation path, and split into two trackers upon request. This will require a larger number of samples to be evaluated than 3 per bin, to allow the algorithm to detect wide mainlobes. The number of samples needed will depend on the particular radar pulse used. In our case, the test pulse has $-3dB$ width of 50 samples and an
autocorrelation mainlobe width of 100 samples. Thus a pulse which is less spread in the time-domain (i.e. a UWB signal) will perform better in terms of the range resolution. This result is the same as that found in conventional SAR theory [3].

5.2.2 Cross-Range Target Resolving

This configuration considers noiseless scenarios with two or more targets at the same range but different cross range coordinates. Figure 5.6 demonstrates a scenario with two targets having cross range coordinates of 400m and 600m. We see the paths intersect at one location. The tracker will always choose its next sample
based on its original direction of travel when confronted with multiple samples all of high correlation. Thus at the intersection it will not be confused, and chooses the appropriate path. This however can change if the two targets are nearer to each other. Consider Figure 5.7, which shows targets with cross-range locations of 480 and 500. We see that at the intersection, the slope of each path slowly converges, so that the tracker cannot determine which path “out” of the intersection is the “correct” one, generated by the tracker’s target.

**Figure 5.6.** Two targets with same range, different cross-range.
5.3 General Case

The general case consists of $N_t$ targets uniformly distributed in a noisy environment. Since we cannot anticipate the structure or organization of targets in the general case, we must assume all configurations are possible. Figure 5.8 demonstrates a target scene with $N_t = 5$ and SNR=−3 dB. We see that there are many intersections and merging of paths. The test of the tracker’s ability to locate targets will be in running a large number of simulations at varying noise levels and measuring missed detections and location errors.
5.4 Asymptotic Complexity

We saw in Eq. (2.22) the running time for conventional SAR signal processing. In 4.10, we see that for \( m > n \), we only evaluate a constant number of points per bin, with \( m \) bins. In 4.5, we saw that each point evaluation requires \( n \) many multiplications to compute. Lastly, if we have \( T \) targets, the algorithm will create \( T \) many trackers. Thus our running time for the tracking loop will be

\[
\theta(T m \ast n)
\]  

(5.1)

Since our acquisition requires the full circular correlation of \( C_1 \), we will also have an order \( n \log n \) running time, for a total algorithm running time of

\[
\theta(T m \ast n + n \log n)
\]  

(5.2)

When \( T \) is small and \( m \) is large, this running time is optimal over Eq. (2.22). Thus for scenarios with many cross range bins (such as a continuous capture on a long flight) and few targets to track, our algorithm will perform more efficiently than convention SAR methods.

5.5 Simulations Results

In this section the algorithm was run in simulated environments in order to benchmark its performance versus wavefront reconstruction. The primary benchmarks are: cross-range resolution, computational requirement, and false alarms/missed detections. These three cases are examined separately.

5.5.1 Cross Range Resolution Results

The cross-range resolution of each algorithm is measured in the following manner:

- Construct scene with two targets at the same range (arbitrarily chosen range \( Y \)) but different cross ranges.
- Reduce the cross-range distance between the targets.
- Check to ensure the algorithm can still distinguish them as two separate targets
- If so, reduce the distance again, continuing until they are no longer distinguishable.
- The minimum distance that the two targets can still be distinguished is the cross range resolution.

The algorithm-dependent step is verifying if the algorithm can still distinguish targets. For our novel algorithm, simulations were done using the test setup. For wavefront reconstruction, the formulation developed in Section 2.4.4 is used in lieu of an actual algorithm. This allows us to find the optimal resolution, independent of implementation choices. Equation (2.36) is evaluated with the parameters listed in Section 5, and \( H_C \) is constructed. The range of the target used in constructing the
reference phase history is chosen to be in the middle of the imaging area \((T_y=500\, \text{m})\), as is customary in wavefront reconstruction [3].

Table 5.2 compares the resolution obtainable by the novel algorithm and wavefront reconstruction with varying target range \(Y\).

<table>
<thead>
<tr>
<th>(Y)</th>
<th>XRR of Novel Algorithm</th>
<th>XRR of Wavefront Reconstruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>37m</td>
<td>42m</td>
</tr>
<tr>
<td>200</td>
<td>36m</td>
<td>35m</td>
</tr>
<tr>
<td>300</td>
<td>35m</td>
<td>25m</td>
</tr>
<tr>
<td>400</td>
<td>35m</td>
<td>20m</td>
</tr>
<tr>
<td>500</td>
<td>34m</td>
<td>17m</td>
</tr>
</tbody>
</table>

**Table 5.2.** Cross-range resolution as a function of target down-range. \(Y\) is the range of the two targets, and XRR is the smallest distance between the targets that the algorithm was able to distinguish them as distinct from each other.

These results show that the range of the targets greatly impacts the resolution of wavefront reconstruction, but does not impact the resolution of the sparse target tracking algorithm. This is due primarily to wavefront reconstruction requiring a pre-chosen reference target range (chosen to be in the middle of the imaging area). If the actual target range is significantly different from the reference target, the phase histories will not correlate well. This is an inherent limitation of the wavefront reconstruction method. Our novel algorithm performs consistently at any range without our imaging window, although the resolution is on average less than that of wavefront reconstruction.

### 5.5.2 Computational Performance Results

The general running times have been found to be Eq. (5.2) for the novel algorithm and Eq. (2.22) for wavefront reconstruction. If the number of targets is small, and \(m > n\), the novel algorithm performs on an order of magnitude faster than wavefront reconstruction.

However there are many computational penalties that are of interest but do not directly affect the asymptotic order of the algorithm. These computations are constant to input size but can significantly increase the running performance realistic scenarios. In Section 4.2.2, we observed that sometimes the algorithm needs to perform a traceback, which causes more than a constant number of samples of \(V_C\) to be evaluated in a single cross-range bin. The tracebacks require \(mB_{num}\) multiplications to perform, where \(B_{num}\) is the number of tracebacks. Table 5.3 below characterizes the extra multiplications due to tracebacks when trying to resolve targets at different distances from each other. There are two targets separated in cross range by a distance \(\delta R\). We see that as the targets become closer, the number of extra multiplications needed grows significantly. Thus the algorithm performs better in situations where the sparse targets are separated from each other. The multiplications shown in the table are due to the setup parameters \(m = 4000\) and \(n = 2000\), giving us a data size of \(mn = 8\) million data points.
\( \delta R \) \( T_r \) of Novel Algorithm

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0.29%</td>
</tr>
<tr>
<td>300</td>
<td>0.49%</td>
</tr>
<tr>
<td>100</td>
<td>0.725%</td>
</tr>
</tbody>
</table>

Table 5.3. Extra computations required by novel algorithm due to tracebacks. \( T_r \) is the percent increase in computations caused by the tracebacks, compared to the novel algorithm without tracebacks enabled. It is calculated as \( T_r = 100 \left( 1 - \frac{F_T}{F_{NT}} \right) \), where \( F_T \) is the number of floating point multiplications calculated for the algorithm including tracebacks, and \( F_{NT} \) is the number calculated excluding the tracebacks.

5.5.3 Missed Detection Results

In this section we evaluate the performance of the algorithm in noisy environments. A missed detection occurs when the algorithm fails to successfully locate or detect a target which exists in the environment. Figure 5.9 illustrates missed detection of two nearby targets which appear as one. As the distance between the two point targets \( \delta R \) gets smaller, the chance to detect them as one increases exponentially. This is an artifact of our tracking algorithm, and is an unwanted phenomenon that does not occur in phase history methods. A 1% missed detection rate demonstrates that the method is unreliable.

Missed detections can also occur due to environmental noise. Figure 5.10 shows missed detection versus cross-range resolution in different noisy environments. We see that the number of missed detections varies as a function of the noise level.

![Figure 5.9. Probability of missed detection versus cross-range resolution. \( \delta R \) is the distance between two objects. As \( \delta R \) gets small, it is more likely that the tracker will see only one object and not two, thereby not detecting the second target.](image)
Figure 5.10. Probability of missed detection versus cross-range resolution. $\delta R$ is the distance between two objects. We see that both $\delta R$ and SNR affect the number of missed detections.
Chapter 6

Future Work

6.1 Alternate Waveforms

The results found during simulations in this study are only applicable to the particular waveform used during simulation. In future efforts, we would like to examine the effects of varying the waveforms properties, shape, bandwidth, autocorrelation function, and other factors. This should reveal what types of radar the algorithm is more applicable to.

6.2 Simulation of Alternate Setups

We would like to further analyze the effect of the targets in different range bins but the same cross-range bin. The effects of merging paths is a limiting factor in the usage of this algorithm for cross-range resolution. Further effort should be made to reduce the effects of near-range bin interference.

6.3 Additional Scenarios

We would like to consider the usage of such an algorithm for a real-world application. One particular example is the usage of a target-tracking SAR algorithm for navigation. Several objects would be located with precise location information. As the plane moves, a lock on the location of these would continue to tell the aerial vehicle the relative distance to the targets. By measuring the relative distance, we could learn the amount of distance we have traveled, assuming the target is stationary. This scenario would demonstrate the applicability of the algorithm designed in this study.
In this paper, we developed a novel efficient SAR algorithm for target tracking in sparse target environments. The method established an alternative method for locating targets within raw SAR data. The method focuses on finding the target’s cross-range position, replacing the phase history correlation method in wavefront reconstruction.

The novel algorithm exhibited cross-range resolution loss compared to wavefront reconstruction; however, this is a tradeoff for the increased speed and reduced computational complexity. The algorithm runs asymptotically faster than the conventional SAR approach, while maintaining cross-range resolution on the same order of magnitude as wavefront reconstruction. In addition, the novel algorithm can take advantage of all the energy in a wideband or ultra-wideband pulse, but tracking the strong correlation peak from these signals; in contrast, wavefront reconstruction chooses a single frequency and discards other frequency components. Thus it will not perform well with signals that spread their energy out over a large bandwidth.

The algorithm runs faster when there are few targets to track; thus it is designed to operate in sparse target environments. It also implies some constraints on the type of SAR imaging scene which can be used. In particular, the number of cross range bins must be greater than the number of samples collected at each bin.

Finally, the method was analyzed for use as an adaptive online algorithm. This modification causes the algorithm to deliver the best resolution possible when operating with extremely limited computational power. This is achieved by using a real-time adaptive modification at run time.
REFERENCES


