2018

Leveraging 3D Models for SAR-based Navigation in GPS-denied Environments

Zachary A. Reid
Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all

Part of the [Electrical and Computer Engineering Commons](#)

Repository Citation

Reid, Zachary A., "Leveraging 3D Models for SAR-based Navigation in GPS-denied Environments" (2018).
Browse all Theses and Dissertations. 2189.
https://corescholar.libraries.wright.edu/etd_all/2189

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.

Leveraging 3D Models for SAR-based Navigation in GPS-denied Environments

A Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering

by

ZACHARY A. REID
B.S.E.C.E., The Ohio State University, 2016

2018
Wright State University

Wright State University
GRADUATE SCHOOL

October 10, 2018

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Zachary A. Reid ENTITLED Leveraging 3D Models for SAR-based Navigation in GPS-denied Environments BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Electrical Engineering.

Joshua Ash, Ph.D.
Thesis Director

Brian D. Rigling, Ph.D.
Chair, Department of Electrical Engineering

Committee on
Final Examination

Joshua Ash, Ph.D.

Arnab Shaw, Ph.D.

Michael A. Saville, Ph.D., P.E.

Barry Milligan, Ph.D.
Interim Dean of the Graduate School

ABSTRACT

Reid, Zachary A. M.S.E.E., Department of Electrical Engineering, Wright State University, 2018.
Leveraging 3D Models for SAR-based Navigation in GPS-denied Environments.

This thesis considers the use of synthetic aperture radar (SAR) to provide absolute platform position information in scenarios where GPS signals may be degraded, jammed, or spoofed. Two algorithms are presented, and both leverage known 3D ground structure in an area of interest, e.g. provided by LIDAR data, to provide georeferenced position information to airborne SAR platforms. The first approach is based on the wide-aperture layover properties of elevated reflectors, while the second approach is based on correlating backprojected imagery with digital elevation imagery. Both of these approaches constitute the system we have designated: SARNAV. Building on 3D backprojection, localization solutions result from non-convex optimization problems based on image sharpness or correlation measures. Results using measured GOTCHA data demonstrate localization errors of only a few meters with initial uncertainty regions as large as 16 km^2 . Finally, the system is incorporated into a Kalman filter tracker, where periodic SARNAV updates could be used to correct drift from an inertial navigation system. With measured data, the system was able to track the true position along the route within a few meters of error.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Outline	2
2	Overview of Current Radar-based Navigation Systems	3
2.1	TERCOM	3
2.1.1	Background	3
2.1.2	System Overview	3
2.2	Odometry methods	6
3	SAR Exploitation	10
3.1	Point Spread Function	10
3.2	Backprojection	11
3.2.1	Digital Spotlighting	14
4	Proposed Localization Methods	17
4.1	Sharpness-based Localization	18
4.2	Correlation-based Localization	19
5	Optimization Techniques	28
5.1	Search Strategies	28
5.1.1	Simple Grid Search	29
5.1.2	Adaptive Grid Search	30
5.1.3	Template Matching	31
5.2	High-Performance Computing	35
5.2.1	HPC for Sharpness-based Localization	35
5.2.2	HPC for Correlation-based Localization	37
6	Location Tracking	39
7	Results	44
7.1	Synthetic Data	44
7.1.1	Synthetic Data Generation	44

7.1.2	Simulation Results	45
7.2	Measured GOTCHA Data	47
7.2.1	Localization Results	50
7.2.2	Kalman Filtering Results	54
8	Conclusions and Future Work	60
	Bibliography	61

List of Figures

2.1	Tercom elevation pings	4
2.2	Tercom elevation with IMU	5
3.1	Three-dimensional point spread function corresponding to a 90° circular aperture.	12
3.2	Scene geometry between the aimpoint r_0 , location of pixel i (r_i), and the synthetic aperture r_a . In localization, aperture point locations have an unknown common global translation.	13
3.3	Defocused layover example for a 90° aperture. (a) Imaging to the ground plane, (b) imaging plane at half building altitude, and (c) imaging plane equal to building altitude.	16
4.1	Sharpness localization system flow	20
4.2	a) Radar image where the aircraft location hypothesis is accurate b) Radar image formed where the aircraft location hypothesis is off by 100 meters in X and Y.	25
4.3	a) Misalignment from incorrect imaging plane b) Aligned images from choosing true altitude	26
4.4	Correlation system flow.	27
5.1	Heatmap produced by a simple grid search	29
5.2	Map produced by the adaptive grid search	30
5.3	SAR image with target building. This image will be searched using a template.	33
5.4	Template matching with: (a) the DEM template (b) the SAR image of the target building	33
5.5	The heatmap results from (a) template matching and (b) the simple correlation method	34
7.1	Synthetic scene (a) top view (b) side view. The red circles indicate locations of point scatterers.	46
7.2	Localization heatmaps for aperture sizes (a) 5° and (b) 15°	48
7.3	(a) Target building DEM and (b) SAR image used in correlation-based localization.	49

7.4	Comparison of localization methods. (a) sharpness method 0.75° aperture, (b) correlation method 0.75° aperture, (c) sharpness method 10.4° aperture, (d) correlation method 10.4° aperture.	51
7.5	Using 1.5° apertures, the correlation method accurately estimates 10 points along a 9 km flight path.	53
7.6	Localization performance versus aperture size for a $4\text{ km} \times 4\text{ km}$ uncertainty region. Above the threshold of 168 pulses, low localization error is achieved.	54
7.7	Kalman filter results for correlation-based localization.	55
7.8	X coordinate from small flight path experiment, showing the ground truth, the SARNAV measurement, and the state	56
7.9	Y coordinate from small flight path experiment, showing the ground truth, the SARNAV measurement, and the state	56
7.10	X coordinate from large flight path experiment, showing the evolution of the state alongside the ground truth	57
7.11	Y coordinate from large flight path experiment, showing the evolution of the state alongside the ground truth	58
7.12	Prediction error for the X coordinate in the large flight path experiment . . .	58
7.13	Prediction error for the Y coordinate in the large flight path experiment . . .	59

List of Tables

7.1	GOTCHA radar parameters	49
7.2	LIDAR parameters	49

Acknowledgment

I would like to take this opportunity thank my family and friends for supporting me in this journey. To my mom and dad, thank you for your endless support of my pursuit of education. To my brother Jefferson and my lifelong best friend Adam Bross, thank you for the encouragement and guidance you've given me my entire life. Without you guys by my side, I don't know if I would have made it this far. And a special thanks to my partner Navi Kolimi. Thank you for all of your emotional support, and for tirelessly listening to my rantings about "breakthroughs" and "phenomenal performance boosts".

I would like to take this opportunity to extend my sincere gratitude to Dr. Josh Ash. Your guidance and support along the way was invaluable, and to you I owe credit for sparking my initial interest in the signal processing domain. You provided excellent guidance as an advisor, and I found your skill as a teacher to be unparalleled in all of my educational experience. I would also like to thank the Center for Surveillance Research for both funding my research and providing access to high quality research among my peers. Another thanks to the members of my thesis committee, Dr. Michael Saville and Dr. Arnab Shaw.

Dedicated to
My parents, Christine and Jeffrey Reid

Introduction

1.1 Motivation

The global positioning system (GPS) provides tremendous value to both civilian and military applications in the form of positioning, navigation, and timing (PNT) applications. Initially, high-accuracy GPS was reserved for the U.S. military. In the 1990s, the government used Selective Availability to purposely degrade civilian GPS accuracy. However, it became clear that high-accuracy GPS could provide a huge economic benefit to the U.S. and globally. So in 2000, civilians were upgraded to military level performance, and Selective Availability was totally removed. Now, according to [15], GPS contributes an estimated \$68.7 billion to the U.S. economy each year. Its applications include markets like precision agriculture, surveying, fleet vehicle connected telematics, consumer location-based services, and air transportation.

For positioning services, the U.S. government is committed to providing GPS globally with a user range error of ≤ 7.8 meters 95% of the time, and in 2016, the actual performance was found to be $\leq .715$ meters [2]. However, user range error corresponds to the availability of the signal in space, and unlike user accuracy, it does not capture environmental effects like atmospheric attenuation or signal blockage. Because GPS signals are subject to degradation, jamming, and spoofing, user accuracy can be much worse than the user range error. An over reliance on this single technology creates a vulnerability that is lessened by considering alternative PNT solutions. A solution built around radar technol-

ogy would have several advantages. Radar works in any weather condition, as well as any time of day. Also, with GPS the end-user passively receives signals transmitted from satellites in the GPS constellation. The end-user of radar, alternatively, must actively transmit a signal and receive its response. This puts more control in the user's hands and makes it less fragile to certain types of attacks.

In this thesis, we consider the use of synthetic aperture radar (SAR) to provide absolute position information to airborne platforms. A suitable starting goal for the system would be to meet the user range error benchmarks that GPS provides. Two algorithms were designed to achieve this goal, and together they form the system that we have entitled SARNAV. It should be noted that much of this material was first published in [\[22\]](#).

1.2 Outline

This thesis has been divided into the following chapters. Chapter 2 presents an overview of the current state of radar localization techniques. This includes analysis of the TERCOM system, as well as several different advances in radar odometry. Chapter 3 discusses the foundations of SAR and its relevant properties to SARNAV localization. In Chapter 4, the two proposed localization systems are presented. In Chapter 5, different optimization techniques are discussed. This includes strategies to reduce the search space and search time, as well as performance improvements through the use of high performance computing. Chapter 6 introduces some Kalman filter background and how it is used in this thesis to create a full tracking system. In Chapter 7, localization results for both systems are reported. Both simulated and measured data results are covered. Finally, Chapter 8 presents the conclusion and suggestions for future work.

Overview of Current Radar-based Navigation Systems

In this chapter, we review prior developments in radar-based navigation. They all fall under the category of Terrain-aided navigation systems (TANS), including the popular TERCOM system.

2.1 TERCOM

2.1.1 Background

Some of the earliest work in radar-based navigation is the TERCOM [9, 23, 5, 8] system, short for terrain contour matching. This is a terrain navigation system that uses terrain correlation to achieve localization. The system was originally proposed in 1950s and has evolved further throughout the years, finding use in many different navigation arenas, such as aircrafts, drones, submarines, and cruise missiles.

2.1.2 System Overview

TERCOM systems rely on altimetry to perform localization. In order to operate, TERCOM systems require a radar altimeter, a barometric altimeter, an inertial measurement unit, and

a computer to perform calculations and correlation processing.

First, a terrain map must first be created prior to flight. Each point on the terrain map corresponds to the terrain elevation, e_T above mean sea level (MSL). To calculate e_T , the barometric altitude a_B and radar altitude a_R are also required, as shown by the following calculation:

$$e_T = a_B - a_R. \quad (2.1)$$

By subtracting the altitude of the aircraft above the terrain from the altitude above the MSL given by a_B , the resulting measurements are now advantageously independent of the flight altitude. Periodic samples are recorded while a platform in TERCOM mode is flying over a potential terrain map match. Figure 2.1 shows how an aircraft would collect measurements in TERCOM mode.

Figure 2.1: Tercom elevation pings

Cruise missiles navigation systems provide a good example of the operational aspects of TERCOM. While a cruise missile is in flight, it primarily uses an inertial navigation system (INS), consisting of accelerometers and gyroscopes. When a TERCOM system is

included, the system provides periodic updates to the INS to counteract drift and stabilize the overall accuracy. Figure 2.2 shows how a platform using TERCOM navigates from terrain map to terrain map, utilizing INS in between.

Figure 2.2: Tercom elevation with IMU

TERCOM systems can provide many strategic benefits. For one, it can operate day or night and in any weather condition. This is very important, as it provides the flexibility required for military operations. Additionally, it enables cruise missiles to fly at much lower altitudes, greatly reducing their likelihood of being detected.

There are, however, limitations to the TERCOM system. One crucial assumption is that the terrain map is absolutely unique, in that it must be able to be properly distinguished from any other patch. As [24] notes, the positioning accuracy of TERCOM systems is closely related to the selection of terrain matching areas. Such influencing selection characteristics include terrain standard deviation, terrain roughness, and terrain entropy.

2.2 Odometry methods

A second type of radar-aided navigation system uses odometry methods [11, 20, 21]. Rather than navigating by matching measured radar data to a database of known features, these systems aid in navigation by tracking the locations of scattering centers in a scene.

Reference [11] presents a solution that uses ultrawideband orthogonal frequency division multiplexed (UWB-OFDM) radar for odometry. The UWB-OFDM sensor technology used in their system grants some desirable characteristics, such as 30 cm down range and cross range resolution for precise localization, as well as high bandwidth and adaptive waveforms to counter jamming or spoofing. These benefits, however, come at the price of high computational burden. In order to unload some of the burden, the navigation system integrates the data on a pulse-by-pulse basis instead of dealing with the burden of SAR image formation.

In this system, features must first be extracted from the radar signal. The raw 1D pulse first undergoes matched filter processing. Then, the resulting range bins that exceed a specified threshold for the pulse are designated as “target observations”. The threshold must first be calculated by specifying a target constant false alarm rate (CFAR), which is done by approximating various statistics about the signal and noise power. The target observations for each pulse are then used as features.

Next, the target observations are matched to different possible tracks using a data association approach called global nearest neighbor (GNN). A cost is calculated for an observation to each of its possible tracks, using an estimate of the range of the aircraft from the observation (from the matched filter range bin) and an estimate of the range of the aircraft from the track (from the last known aircraft position). The mapped observations and set of tracks that minimize the cost function are added to a collection of tracks, but they are not yet considered confirmed. This is where the M/N algorithm is used. For a given unconfirmed track in the collection, if M out of N consecutive observation sets map to the same track, the track is confirmed and directly integrated into the guidance system as

a localization fix. An Extended Kalman Filter is used to handle position updates from the radar tracking with the INS.

Reference [21] presents a radar odometry solution designed for fixed-wing small unmanned aircraft. This system assumes that the scattering centers are stationary, and thus, the motion of the aircraft can be inferred via a system of scatterer-to-platform range equations. Optionally, a database of ground-based features can be used; however, if an existing database does not exist, it can operate in SLAM mode, or simultaneous localization and mapping. Consequentially with this approach, the system can only provide relative position tracking. All of this is finalized with an Extended Kalman Filter.

To perform tracking, a range-compressed image is generated. This shows how the returns from the different range bins evolve with time. Due to the geometry of the collection and under the assumption of a straight flight path observing stationary scatterers, a scatterer will show up as a hyperbola in the range-compressed image.

Using a range-compressed image, the system can be broken down into the following steps:

1. A range-compressed image is generated.
2. The altitude above ground level is calculated to later be used to estimate the aircraft's relative drift.
3. The range-compressed image is run through a prefilter to remove noise.
4. A Hough Transform is used on the filtered image to isolate hyperbolas, indicating strong scatterers.
5. The relative drift (Δx , Δy) is calculated using the parameterized hyperbolas.
6. The Extended Kalman Filter is updated using the relative drift, the altitude above ground level, and IMU measurements.

Niedfeldt in [20] presents a system that uses the Recursive RANdom SAmple Consensus (R-RANSAC) algorithm to track scatterers using synthetic aperture radar and estimate their motion. This work expands off of Kauffman’s GNN odometry [11] and Quist’s ground scatterer tracking by hyperbolic Hough transform [21]. They present the findings that the Quist’s Hough transform is too computationally slow, as it is a voting algorithm, and that Kauffman’s GNN and M/N detector is not robust and can diverge from true data association.

RANSAC [7] is an algorithm designed to fit a model to a data set while being robust to occasional erroneous measurements. To perform odometry, as done with the Hough transform, the goal is to estimate the parameters of hyperbolas to infer information about the motion of the aircraft. First, observation sets are randomly selected from the measurement set $S = (s, b_j[s])$, where s is the chirp pulse and $b_j[s]$ is the range bin return. A hypothesis is then generated from a set of these observations. Then, the number of ”inliers”, or measurements that fit the model within a certain error term, determine the degree of support for this hypothesis. This process is repeated over a set number of iterations, continuously revising the hypothesis by selecting for the ones with highest support. Finally, the inliers that fit the model are smoothed and the hypothesis is accepted. Once the hyperbola statistics are determined, the motion of the aircraft is inferred.

Niedfeldt’s system uses the RANSAC method to generate new hypotheses, but also extends it by implementing R-RANSAC [19] for model updating. They form two separate entities in the system. First, a measurement is taken like in the simple RANSAC case. This measurement is tested against the current set of models describing previously validated hyperbolas. If the measurement is an inlier, the model subsumes this measurement and incorporates the information. If the measurement is an outlier, a new hypothesis is formed from this measurement and all of the previous measurements within a specified window of time. R-RANSAC only maintains a set number of models, so the set of models is continuously updated, replacing old, lower support models with new ones that are more

consistent with all of the data.

SAR Exploitation

In this chapter, we review the basic elements of SAR data collection and 3D image formation. The SAR point spread function, the backprojection algorithm, and the digital spotlight algorithm will all be discussed in detail, as they will later be used in the development of novel localization algorithms.

3.1 Point Spread Function

The three-dimensional point spread function (3D-PSF) of a synthetic aperture collection is a function of spatial pixel location $\mathbf{r} = [x, y, z]^T$ and is given as [16]

$$\rho(\mathbf{r}) = \sin(\phi_0) \int_{k_1}^{k_2} k^2 \int_{\theta_1}^{\theta_2} e^{2jk\|\mathbf{r}-\mathbf{R}\|} \cdot e^{-2jk\|\mathbf{R}\|} d\theta dk, \quad (3.1)$$

where θ_1, θ_2 are the beginning and ending azimuth angles of the flight path, respectively, and k_1, k_2 are the beginning and ending wavenumbers, respectively. For propagation velocity c , the wavenumber for the i th frequency is given as $k_i = \frac{2\pi f_i}{c}$. For a circular aperture with radius R_0 and polar angle ϕ_0 , the platform position $\mathbf{R}$ is given as

$$\mathbf{R} = \begin{bmatrix} x_r(\theta) \\ y_r(\theta) \\ z_r(\theta) \end{bmatrix} = \begin{bmatrix} R_0 \cos(\theta) \sin(\phi_0) \\ R_0 \sin(\theta) \sin(\phi_0) \\ R_0 \cos(\phi_0) \end{bmatrix}. \quad (3.2)$$

Evaluating (3.1), an example of the 3D-PSF for a 90° aperture ($\theta_1 = \frac{\pi}{2}, \theta_2 = \pi$) for an X-band radar with 640 MHz bandwidth and center frequency of 10 GHz is shown in Figure 3.1. The peak of the 3D-PSF occurs at the true point location (origin). Large amplitude reflectivity also descends downward from the origin in the direction of the radar. This represents the well-known effect of layover [4] where elevated targets appear closer to the radar in ground-plane imagery. Exploring x - y slices of the 3D-PSF with descending values of elevation z , we observe that not only are high-energy areas displaced away from the origin (layover), but the spread of the PSF increases azimuthally. We refer to this effect as *defocused layover*, which is exacerbated by larger synthetic apertures and when imaging to a plane that is significantly displaced from the true elevation of a target.

For full 360° circular apertures, the 3D-PSF (3.1) may be approximated as [17]

$$\rho(\mathbf{r}) = 2\pi \sin(\phi_0) \int_{k_1}^{k_2} k^2 e^{2jkz \cos(\phi_0)} J_0(2k\sqrt{x^2 + y^2} \sin(\phi_0)) dk. \quad (3.3)$$

The locus of high-amplitude points in the 360° PSF (3.3) resembles a cone and is often referred to as a “cone of confusion” [18]. The 90° aperture considered above is a subset of a full circular aperture, and the resulting 3D-PSF in Figure 3.1 appears as a portion of a complete cone whose apex is located at the point reflector’s true location.

3.2 Backprojection

When a region to be imaged contains large elevation diversity, resulting from, e.g., terrain variance and/or tall buildings, imaging to a single elevation plane produces distorted and defocused imagery due to the 3D-PSF and defocused layover effects described above. Alternatively, backprojecting to a surface consistent with the ground elevation profile ame-

Figure 3.1: Three-dimensional point spread function corresponding to a 90° circular aperture.

liorates these effects. For a P -pulse aperture, the backprojected reflectivity for pixel i is

$$I_i = \sum_{p=1}^P Q_p(d_p(\mathbf{r}_i)) W_x(\theta_p), \quad (3.4)$$

where $Q_p(r)$ is the filtered range profile of pulse p . The frequency response, $|\omega|W_r(\omega)$, of the range filter includes the convolution backprojection frequency ramp ($|\omega|$) and an optional apodization window in the range dimension, $W_r(\omega)$. Similarly, $W_x(\theta_p)$ represents an optional cross-range apodization window. In (3.4), the p th range profile is evaluated at

$$d_p(\mathbf{r}_i) = \|\mathbf{r}_a(p) - \mathbf{r}_0\| - \|\mathbf{r}_a(p) - \mathbf{r}_i\|, \quad (3.5)$$

Figure 3.2: Scene geometry between the aimpoint $\mathbf{r}_0$, location of pixel i ($\mathbf{r}_i$), and the synthetic aperture $\mathbf{r}_a$. In localization, aperture point locations have an unknown common global translation.

the differential range to pixel i , which has physical location

$$\mathbf{r}_i = \begin{bmatrix} x_i \\ y_i \\ h(x_i, y_i) \end{bmatrix}, \quad (3.6)$$

where the function $h(x_i, y_i)$ returns the surface elevation at location (x_i, y_i) . As illustrated in Figure 3.2, $\mathbf{r}_0$ represents the 3D radar aimpoint, and $\mathbf{r}_a(p)$ denotes the 3D location of the p th point along the synthetic aperture.

A synthetic-data example of imaging elevated point targets is shown in Figure 3.3. Here, the radar and collection parameters are identical to those described in Section 3.1 with 3D-PSF illustrated in Figure 3.1. The scene in Figure 3.3 consists of a dense array of point scatterers on top of a single 100 m tall building. Three backprojection images with planer surfaces were generated: 1) Fig. 3.3(a), with the image plane $z = 0$ m set at ground level, 2) Fig. 3.3(b), with the image plane set at $z = 50$ m, and 3) with the image plane $z = 100$ m set equal to the height of the scatterers. For the ground plane image, the defocused layover effect causes the roofline to appear displaced and distorted. At the intermediate imaging plane both effects are lessened, and when the image plane coincides with the scatterers, the roofline appears sharply focused and in the correct position.

For small apertures, image plane displacement primarily results in layover with limited defocus. However, the direction of layover-induced target translation is highly dependent upon viewing angle. Further, because layover extent depends on scatterer elevation, this effect alone can cause image distortion among targets at different heights.

3.2.1 Digital Spotlighting

Digital spotlighting is a method used to improve the computational efficiency in processing SAR data. This process reduces the total number of samples in both the fast-time and slow-time dimensions while still achieving Nyquist sampling rates to avoid aliasing [10]. This can be achieved because the product of digital spotlighting is a transformed version of the original phase history data, corresponding to a much smaller spatial region from the original radar swath. Hence, digital spotlighting is ideal in situations that deal with smaller regions of interest. The algorithm is comprised of the following steps:

1. The coordinates and size of the region of interest are identified. This size should be substantially smaller than the original size of the radar swath.
2. The phase history is recentered to the new location, which can be thought of as a form of motion compensation.
3. The phase history is decimated in fast-time. This greatly reduces the total number of frequency samples per pulse.
4. An interpolation step is performed on the slow-time samples to achieve uniform spacing, since the motion compensation point was relocated to the center of the region of interest in the first two steps.
5. Finally, the slow-time samples are decimated, greatly reducing the number of slow-time pulses.

In the SARNAV algorithms described in this thesis, digital spotlighting can be used as a preprocessing step to reduce the total computation time needed to achieve localization. The patch of uncertainty is described as X and Y coordinates and bounding limits that are certain to contain the aircraft. SARNAV can use digital spotlighting directly by designating the patch of uncertainty as the region of interest.

(a)

(b)

(c)

Figure 3.3: Defocused layover example for a 90 aperture. (a) Imaging to the ground plane, (b) imaging plane at half building altitude, and (c) imaging plane equal to building altitude.

Proposed Localization Methods

We assume that the locations of the radar platform along the synthetic aperture are known up to an unknown translation common to all points, i.e.

$$\mathbf{r}_a(p) = \tilde{\mathbf{r}}_a(p) + \boldsymbol{\theta}, \quad p = 1, \dots, P \quad (4.1)$$

where

$$\boldsymbol{\theta} = \begin{bmatrix} x_0 \\ y_0 \\ 0 \end{bmatrix} \quad (4.2)$$

represents the unknown shift amount. This implies a number of assumptions about the flight path: 1) the shape is known, 2) the orientation is known, and 3) the elevation is constant and known. All of these quantities are readily available from onboard sensors, such as an IMU, compass, and altimeter. Further, as shown later, localization from very short flight paths is possible, making linear path approximations valid and lessening the impact of IMU drift. Consequently, combining an estimate of $\boldsymbol{\theta}$ with (4.1) provides us absolute position estimates for the platform along the entire aperture.

4.1 Sharpness-based Localization

As illustrated earlier, optimal image focus is obtained when the 3D backprojection surface coincides with the elevation structure of in-scene scatterers. Consequently, we can use image focus as a metric to quantify the proper relative alignment between a georegistered DEM surface and the radar flight path. Rewriting the differential range (3.5) to include aperture uncertainty (4.1), we have

$$d_p(\mathbf{r}_i, \boldsymbol{\theta}) = ||(\tilde{\mathbf{r}}_a(p) + \boldsymbol{\theta}) - (\mathbf{r}_0 + \boldsymbol{\theta})|| - ||(\tilde{\mathbf{r}}_a(p) + \boldsymbol{\theta}) - \mathbf{r}_i||. \quad (4.3)$$

Combining a georegistered DEM for pixel locations and elevations $\{\mathbf{r}_i\}$ with (4.3) for differential ranges, we use (3.4) to produce a backprojection image

$$I_i(\boldsymbol{\theta}) = \sum_{p=1}^P Q_p(d_p(\mathbf{r}_i, \boldsymbol{\theta})) W_x(\theta_p) \quad (4.4)$$

that is dependent upon the unknown aperture translation $\boldsymbol{\theta}$.

Various measures, such as entropy [14] and sharpness [6], are available to quantify the degree of focus of an image. Here, we consider image sharpness, defined as

$$f_{sharp}(\boldsymbol{\theta}) = \sum_i |I_i(\boldsymbol{\theta})|^4 \quad (4.5)$$

as a measure of the optimality of $\boldsymbol{\theta}$. The optimal position estimate is chosen as the value that maximizes the sharpness

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} f_{sharp}(\boldsymbol{\theta}). \quad (4.6)$$

The disadvantage of (4.6) is that each evaluation of the objective function f_{sharp} requires

computing a new backprojection image, which has a large computational complexity: $\mathcal{O}(PN^2)$, where N^2 denotes the total number of pixels. Additionally, to generate measurable defocus requires relatively large apertures (large P). In summary, the full localization system can be seen in Figure 4.1.

4.2 Correlation-based Localization

The correlation-based method described in this section is designed to minimize the computational and aperture requirements encountered with the sharpness-based approach. Rearranging (4.3), we have

$$d_p(\mathbf{r}_i, \boldsymbol{\theta}) = \|\tilde{\mathbf{r}}_a(p) - \mathbf{r}_0\| - \|\tilde{\mathbf{r}}_a(p) - (\mathbf{r}_i - \boldsymbol{\theta})\|, \quad (4.7)$$

which indicates that shifting the aperture by $+\boldsymbol{\theta}$ is equivalent to shifting the DEM by $-\boldsymbol{\theta}$. When backprojecting to a DEM surface $z_i = h(x_i, y_i)$, we have

$$\mathbf{r}_i - \boldsymbol{\theta} = \begin{bmatrix} x_i - x_0 \\ y_i - y_0 \\ h(x_i, y_i) \end{bmatrix}. \quad (4.8)$$

While the point $[x_i - x_0, y_i - y_0, h(x_i - x_0, y_i - y_0)]^T$ exists in a $\boldsymbol{\theta} = 0$ image, the point $\mathbf{r}_i - \boldsymbol{\theta}$ in (4.8) does not. This necessitates computing a new backprojection image for each value of $\boldsymbol{\theta}$ as required by the sharpness approach. In contrast, when backprojecting to a

Figure 4.1: Sharpness localization system flow

plane with fixed elevation $z_i = z_{const}$, we have

$$\mathbf{r}_i - \boldsymbol{\theta} = \begin{bmatrix} x_i - x_0 \\ y_i - y_0 \\ z_{const} \end{bmatrix}. \quad (4.9)$$

This expression indicates that for platform translation $\boldsymbol{\theta}$, the value of pixel i is merely shifted in the x - y image plane by (x_0, y_0) . Hence, we are motivated to develop a new localization algorithm based on a single elevation plane because all candidate imagery, for different hypothesized positions $\boldsymbol{\theta}$, may be efficiently generated by a single backprojection and subsequent translations. This effect can be seen below in Figure 4.2. Here, the image is created from backprojecting to a flat plane set to the height of the building. In (a), the correct aircraft location is hypothesized, while (b) shows the result when the hypothesized location is incorrect by 100 meters in both X and Y.

Now, to present a more rigorous explanation of this process, let $\boldsymbol{\xi}$ represent the vectorized scene reflectivity, and let

$$\mathbf{g} = \mathbf{F}_\theta \boldsymbol{\xi} + \mathbf{n} \quad (4.10)$$

denote vectorized phase history, where $\mathbf{n}$ represents additive noise, and $\mathbf{F}_\theta$ is the Fourier operator consistent with the bandwidth, flight path, and translation vector $\boldsymbol{\theta}$ of the radar platform. For complex circular white Gaussian noise with variance σ^2 , the likelihood function for $\boldsymbol{\theta}$ is

$$p(\mathbf{g}|\boldsymbol{\theta}, \boldsymbol{\xi}) \propto e^{-\frac{1}{\sigma^2}(\mathbf{g} - \mathbf{F}_\theta \boldsymbol{\xi})^H (\mathbf{g} - \mathbf{F}_\theta \boldsymbol{\xi})}, \quad (4.11)$$

and, neglecting $\boldsymbol{\theta}$ -independent terms, the log-likelihood function is

$$LL(\boldsymbol{\theta}) = 2 \operatorname{real}(\mathbf{g}^H \mathbf{F}_\theta \boldsymbol{\xi}). \quad (4.12)$$

To estimate the position vector, we consider the maximum likelihood estimate of $\boldsymbol{\theta}$

$$\hat{\boldsymbol{\theta}}_{MLE} = \arg \max_{\boldsymbol{\theta}} LL(\boldsymbol{\theta}), \quad (4.13)$$

however, because the true reflectivity image $\boldsymbol{\xi}$ is unknown, (4.13) cannot be used directly. As a surrogate for $\boldsymbol{\xi}$, we consider the digital elevation image $\mathbf{D}$, where each pixel's amplitude corresponds to its elevation. Finally, applying the adjoint property of the Fourier operator

$$\langle \mathbf{g}, \mathbf{F}_\theta \boldsymbol{\xi} \rangle = \langle \mathbf{F}_\theta^H \mathbf{g}, \boldsymbol{\xi} \rangle \quad (4.14)$$

and noting that $\mathbf{F}_\theta^H \mathbf{g} = \mathbf{I}(\boldsymbol{\theta})$ is the backprojection image from phase history $\mathbf{g}$, we obtain the following surrogate log-likelihood function

$$LL(\boldsymbol{\theta}) \approx \operatorname{real}(\mathbf{I}(\boldsymbol{\theta})^H \mathbf{D}). \quad (4.15)$$

As noted above, when imaging to a *plane*, backprojection images $\mathbf{I}(\boldsymbol{\theta})$ are merely shifts of a $\boldsymbol{\theta} = \mathbf{0}$ image. As such, the LL metric may be written as

$$f_{corr}(\boldsymbol{\theta}) = \sum_{x_i} \sum_{y_j} \operatorname{real}(\mathbf{I}(x_i - x_0, y_j - y_0)) \mathbf{D}(x_i, y_j), \quad (4.16)$$

i.e., a correlation between the backprojection image $\mathbf{I}$ and the DEM $\mathbf{D}$. The maximum

(surrogate) likelihood estimate is the peak of this correlation

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} f_{corr}(\boldsymbol{\theta}). \quad (4.17)$$

In evaluating (4.17), there are a number of implementation details that aid performance and reduce computational complexity. The steps followed are

1. Identify a region of uncertainty $\mathcal{R} \subset \mathbb{R}^3$ known to contain the collection aimpoint, $\mathbf{r}_0 \in \mathcal{R}$.
2. From the complete DEM, identify an elevated feature within $\mathcal{R}$, such as the top of a building. Denote the elevation as z_{feat} and extract a small region, $\mathbf{D}$, around the feature from the complete DEM.
3. Compute a backprojection image $\hat{\mathbf{I}}$ at fixed elevation $z = z_{feat}$ over the region $\mathcal{R}$.
4. The estimate $\hat{\boldsymbol{\theta}}$ in (4.17) is then identified as the location of the peak of the 2D cross-correlation between $\mathbf{I}$ and $\mathbf{D}$.

While the sharpness-based approach quantified the defocus of bright scatterers, the correlation-based method utilizes shape information. As such, performance is improved when the elevated target identified in Step 2 above has a unique shape within the uncertainty region $\mathcal{R}$. This aligns with the necessary uniqueness conditions of the TERCOM system presented in Chapter 2. Intuitively, correlation works well because elevated man-made objects, such as buildings, typically exhibit measurable scattering on their surface and edges that coincides with DEM geometry. By imaging to the plane $z = z_{feat}$ in Step 3, the imaged features of the elevated target appear in the proper position, irrespective of the viewing angle of the SAR platform. This ensures that $\mathbf{D}$ is maximally correlated with $\mathbf{I}$ at the proper location. It also has the benefit of distorting other portions of the scene that occur across a diversity of elevation levels. This effect can be seen in Figure 4.3. Here,

(a) and (b) contain images that were created using apertures at 0° , 90° , 180° , and 270° in the circular flight path. Figure 4.3(a) shows what happens when the altitude estimate is incorrect and (b) shows how they align when the altitude is correctly estimated. The full localization system can be seen in Figure 4.4.

(a)

(b)

Figure 4.2: a) Radar image where the aircraft location hypothesis is accurate b) Radar image formed where the aircraft location hypothesis is off by 100 meters in X and Y.

Figure 4.3: a) Misalignment from incorrect imaging plane b) Aligned images from choosing true altitude

Figure 4.4: Correlation system flow.

Optimization Techniques

Both localization methods in the previous chapter require solving complicated non-convex optimization problems based on substantial amounts of data. This necessitates careful consideration of the search strategy and application of computational resources. In this chapter, three search strategies will be presented, alongside specialized high-performance computing algorithms for each system.

5.1 Search Strategies

The localization problem is best described as an attempt to find $\hat{\theta}$, the unknown global translation of the aperture. Therefore, it can be formulated as a 2D optimization problem to find the point in 2D space that maximizes the localization system's objective function.

$$\hat{\theta} = \arg \max_{\theta} f(\theta), \quad (5.1)$$

where $f(\theta)$ represents either (4.5) in the sharpness method or (4.16) in the correlation method. This section explores techniques to solve this optimization problem, such as simple grid search, adaptive grid search, and quick localization by template matching.

5.1.1 Simple Grid Search

The most straightforward approach to solving the optimization problem is a simple grid search. This strategy performs an exhaustive search by evaluating the objective function at each point on a set grid in the 2D search space $\mathcal{R}$. This ensures that, up to the grid resolution, the true global maximum will be found. Performing a simple grid search results in a heatmap. An example heatmap can be seen in in Figure 5.1. In the figure, the x and y axes represent translations from the true aircraft location. The peak at (0,0) indicates a correct localization, and so this example produced the estimate of $\hat{\theta} = [0, 0]^T$.

Figure 5.1: Heatmap produced by a simple grid search

Since the simple grid search evaluates the objective function at unnecessary hypotheses, the simple grid search is too inefficient for real-time localization and is primarily used for analysis and validation.

5.1.2 Adaptive Grid Search

A novel optimization method was designed to circumvent evaluating each point on the grid. Originally, one idea was to use gradient descent; however, upon experimentation, the surface plot for sharpness was found to be highly irregular, and false local maxima occur in many different small pockets. So, the goal became to find the global maximum more quickly by omitting evaluations that are unlikely to be near the peak, but still evaluating at a fine enough resolution to confidently capture the true peak. An example of the adaptive grid search result can be seen in Figure 5.2. The dark blue negative space corresponds to evaluations of the objective function in the uncertainty region that were skipped.

Figure 5.2: Map produced by the adaptive grid search

To start, a tier-1 sampling rate is set for the grid in x and y . These points correspond to tier-1 evaluations and are shown by the light blue points in the figure. A brief glance at

the image shows that the tier-1 sampling rate is 10 meters. Next, an average pooling stage is performed. Each region bounded by 4 light blue points is assigned a score equal to the average of the 4 evaluations. Next, a pooling degree is set for tier-1, indicating how many regions will be chosen to advance to the next level. In the example, the pooling amount is set to 10, as indicated by the 10 regions of spaced orange dots.

At this point the process just repeats, and so the tier-2 sampling rate is set. This should be at a higher resolution than the original sampling rate, since the idea here is to iteratively increase resolution in high-probability regions. The orange dots in the example are set to a spacing of 2 meters, and each point is evaluated. A score is then assigned to each region, equal to the sum of all of the evaluations. The region with the highest score advances, and a final simple grid search is performed on it at whatever resolution required. In this example, the yellow region shows the final selected area.

This algorithm was tested in early developments for the sharpness localization system and preliminary results showed promise for reducing the total computation time by up to 50%. However, more detailed analysis was not performed because, as the next section details, progress with the correlation-based system reduced the need.

5.1.3 Template Matching

Template matching is an algorithm in image processing often used for object recognition tasks. The idea is to take a representation of an object as a template and search a larger image for any subregion that contains the template. It can be thought of as an enhanced form of matched filtering.

The use of template matching in SARNAV evolved from the correlation-based system, as it was originally observed that the speed of the correlation method could be improved by using Fast Fourier Transforms. Here, the DEM of the target building is the template and the SAR image of the uncertainty region is the search image. By computing the normalized cross-correlation (NCC) between the template and the image, the correlation method

became much quicker and much more robust. The method to compute the NCC is shown in [3], and is calculated as

$$NCC(u, v) = \frac{\sum_{i=1}^{A-1} \sum_{j=1}^{B-1} f(i, j) \cdot g(u + i, v + j)}{\sqrt{\sum_{i=1}^{A-1} \sum_{j=1}^{B-1} g^2(u + i, v + j)}} \quad (5.2)$$

where f is the template, g is the image to search, u and v are the indices in the search image, and A and B are the number of rows and columns of pixels in f . This method is more robust than simple cross-correlation, as normalization helps attenuate the impact from strong independent scatterers on the metric. With template matching through the use of the NCC, only the shape statistics of the image matter for localization.

To demonstrate the strength of the template matching approach, we used a real radar scene with a DEM of a target building as the template. The setup and results can be seen in the figures below. Figure 5.3 shows an example search image in the template matching process. This image was generated from 400 pulses of a SAR scene, with the target building encapsulated by the red circle. Figure 5.4 shows the DEM template in (a) and a zoomed up version of the SAR image of the matched building in (b). Template matching produces the heatmap result in Figure 5.5(a). The heatmap is the strongest at about $(x, y) = (510, 100)$ which aligns precisely with the location of the target building in Figure 5.3. This indicates that template matching was able to identify the pixel location of the target in the SAR image, which in turn can be used to localize the aircraft based on the principles shown previously in the correlation-based approach. Figure 5.5(b) shows the analogous heatmap produced from the same scene when simple correlation was used. A local maximum does occur at the true location, but accurate global localization is not able to be performed due to the large scene size and small aperture size.

Figure 5.3: SAR image with target building. This image will be searched using a template.

(a)

(b)

Figure 5.4: Template matching with: (a) the DEM template (b) the SAR image of the target building

(a)

(b)

Figure 5.5: The heatmap results from (a) template matching and (b) the simple correlation method

5.2 High-Performance Computing

The proposed SARNAV localization systems operate with backprojection at their core. Backprojection is computed on a pulse-by-pulse basis, which makes it prime for parallelization. Fortunately, Wright State University was able to provide access to high-performance computing for this project, using the Desch supercomputer received from WPAFB.

In 2009, Wright Patterson Air Force Base opened a new supercomputer facility, built specifically to handle data from the GOTCHA radar system [12]. This supercomputer, entitled Desch, was one of the 500 fastest computers in the world at its release, benchmarked at 23 trillion floating point operations per second, or FLOPS. Desch was built with 2,048 processors and 3 TB of RAM. Gaining access to this kind of computing power opened the door for more in depth analysis.

The more advanced optimization techniques in the previous section are, for the most part, independent of the high-performance computing in this section. Successful localization for the systems described in this thesis require the global maximum of the objective functions to be located at the ground truth location. Therefore, performing a thorough validation of the localization solution requires evaluating the objective function at each point on the grid, which can also be thought of as generating a full heatmap with no discontinuities. Further, the HPC's use case here is mostly for analysis, but it also fits the case of localization with simple grid search.

5.2.1 HPC for Sharpness-based Localization

The sharpness-based system has much to be gained from high performance computing, given that a full backprojection image must be computed for each test of the objective function. There are many factors that influence the time to perform a localization. For one, the uncertainty region must be considered. The heatmap represents the size of the

uncertainty region, of which both the x and y bounds must be considered, as well as the necessary x and y resolution. Also, the size of the backprojected image must be considered. In each backprojection for a single pulse, there must be an interpolation calculation for each pixel in the final image. Finally, the size of the synthetic aperture must be considered, as each pulse requires requires a new backprojection.

For example, when a 500 meter by 500 meter uncertainty region is to be tested at a resolution of 5 meters, there would have to be 250,000 ($500 * 500 = 250,000$) backprojections performed for each pulse, and as one can later see, the sharpness based system requires a very large number of pulses. Results from testing showed that the sharpness-based system required about 86,000 pulses for proper localization to occur. The standard imaging resolution used is 1 meter, where the extracted DEMs of the target buildings are typically about 150 meters by 150 meters, or 150 by 150 pixels.

There were two options to consider for parallelization. Since a final backprojection image is just the superposition of the images generated from each pulse, one option would be to parallelize the pulses and recombine the images at the end. Since there are so many pulses for sharpness, it seemed fitting to parallelize on this granular level. However, this option requires either a) the storage of the backprojection images for each of the 250,000 evaluations or b) more advanced parallelization techniques wherein a manager node would separate and incrementally combine the images for each evaluation of the objective function, only to be later combined into a heatmap.

A second, more advantageous option for the sharpness-based system is to parallelize each hypothesis of the objective function. All of the pulses for a hypothesis are used to generate the backprojection image, the metric is computed and stored, and the image is discarded. Each node is be assigned a subset of the total number of evaluations to be computed, where in the case above, there are 250,000. If 24 HPC nodes are used, each node will have to generate about 10,425 backprojection images. The steps proceed as follows:

1. Designate the number of HPC nodes to use.
2. Specify the size and resolution of the objective function heatmap.
3. Gather the set of evaluations (indicating a hypothesized platform position in x and y) into a list.
4. Divide the list of hypothesized platform positions equally among each of the HPC nodes.
5. Each node generates the full backprojection image for each of its hypothesized locations.
6. The sharpness metric is calculated on the final images and stored in a list.
7. Once all nodes are finished, the lists containing sharpness values are combined to form the heatmap.
8. The global maximum is found, and therefore, the localization estimate.

5.2.2 HPC for Correlation-based Localization

There are a few things to consider for parallelization of the correlation-based system. The tradeoffs with the correlation-based system are similar to those of the sharpness-based system; however, they contrast in the fact that the correlation-based system only requires one backprojection image to be generated. This prompted the decision to parallelize the system by each pulse.

1. Designate the number of HPC nodes to use.
2. Specify the size and resolution of the objective function heatmap.
3. Divide the list of pulses equally among each of the HPC nodes.

4. Each node calculates the backprojection image from their pulses.
5. Once all nodes are finished, all of the backprojection images are added together.
6. Template matching is performed on the combined backprojection image using the DEM of the target building as a template.
7. The global maximum is found from the template matching heatmap, and subsequently, the localization estimate.

Location Tracking

The earlier chapters only considered estimating a single position with no a priori knowledge other than $\mathcal{R}$. In this chapter, we extend single point estimation into a tracking system that supports periodic updates. The Kalman filter [13] is appropriate for this task, as it is a common and powerful tool used for modeling dynamic systems. It has many applications in the signal processing domain, including the filtering, smoothing and prediction of signals.

A Kalman filter is used with SARNAV to close the loop and make it a real-time tracking system that can monitor the state of the platform position using periodic updates. Experimental results with measured GOTCHA data are shown later in Section 7.2.2 to test the location tracking accuracy.

The process follows the dynamical model

$$\mathbf{s}(n) = \mathbf{A} \cdot \mathbf{s}(n - 1) + \mathbf{B} * \mathbf{U}(n), \quad (6.1)$$

where $\mathbf{A}$ is the state transition matrix, $\mathbf{s}(n)$ is the state, $\mathbf{U}(n)$ is the process noise, and $\mathbf{B}$ is a matrix to control how the noise drives the state. In the SARNAV tracker, we use the velocity to propagate the state of the aircraft. Therefore, the state of the dynamical system

must track the position and velocity. The state $\mathbf{s}(n)$ is defined as

$$\mathbf{s}(n) = \begin{bmatrix} p_x(n) \\ p_y(n) \\ v_x(n) \\ v_y(n) \end{bmatrix}, \quad (6.2)$$

where $p_x(n), p_y(n)$ represent the platform's x and y coordinates with time n , and $v_x(n), v_y(n)$ correspond to x and y velocity. The state transition matrix $\mathbf{A}$ is defined as

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (6.3)$$

where Δ is the time-step between SARNAV tracking updates. For our purposes, Δ is set to 1, as it indicates 1 time-step. The process $\mathbf{U}(n)$ is the driving noise of system, which is assumed to be a zero-mean Gaussian process with a covariance matrix $\mathbf{Q}$. Since velocity is what drives the dynamical system, the process noise matrix $\mathbf{Q}$ is defined as

$$\mathbf{Q} = \begin{bmatrix} u_x(n) & 0 \\ 0 & u_y(n) \end{bmatrix}, \quad (6.4)$$

where $u_x(n)$ and $u_y(n)$ are the process noise components corresponding to the velocity components $v_x(n)$ and $v_y(n)$. These components were experimentally tuned to achieve the best tracking performance. The control matrix $\mathbf{B}$ takes the process noise and transforms it

to properly evolve the state, selecting for the velocity components. It is defined as

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (6.5)$$

In estimating the n^{th} position, the uncertainty region $\mathcal{R}$ is centered around the previous position estimate $(p_x(n-1), p_y(n-1))$, then the correlation-based system with template matching is used to provide the noisy position observation. The observations follow the model

$$\mathbf{o}(n) = \mathbf{H} \cdot \mathbf{s}(n) + w(n), \quad (6.6)$$

where $\mathbf{o}(n)$ is the observation vector

$$\mathbf{o}(n) = \begin{bmatrix} p_x(n) \\ p_y(n) \end{bmatrix}, \quad (6.7)$$

and $\mathbf{H}$ is the observation matrix used to connect the observational components to the state, defined as

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \quad (6.8)$$

The position estimates $p_x(n)$ and $p_y(n)$ in observation vector $\mathbf{o}(n)$ have measurement noise $w(n)$, which to fit the Kalman filter model, is assumed to be Gaussian with zero-mean and covariance matrix $\mathbf{C}(n)$.

Determining an appropriate level of measurement noise is an important component in applying the Kalman filter. The noise depends on a few variables, such as the size of the uncertainty region $\mathcal{R}$, the resolution of the heatmap, and the size of the synthetic aperture. In our Kalman Filter experiments, we used an $\mathcal{R}$ of 500m by 500m, a resolution in x and y of 1m, and synthetic aperture of $.024^\circ$. In order to empirically estimate the covariance matrix $\mathbf{C}(n)$, SARNAV template matching localization estimates were computed using 30 known aperture locations. In this case, the variance of the x measurement was $.22m^2$ and the variance of the y measurement was $1.7m^2$. These likely are dependent on the aperture location of the test, but in any case, these were the values used in $\mathbf{C}(n)$, and written as

$$\mathbf{C}(n) = \begin{bmatrix} .22 & 0 \\ 0 & 1.7 \end{bmatrix}. \quad (6.9)$$

The initial state and error estimates before update are defined as $\mathbf{s}(n|n-1)$ and $\mathbf{m}(n|n-1)$. Likewise, the previous state and error estimates are defined as $\mathbf{s}(n-1|n-1)$ and $\mathbf{m}(n-1|n-1)$. Since prior state and error estimates are needed, the initial state $\mathbf{s}(-1)$ and initial error $\mathbf{m}(-1)$ need to be defined. The initial positions $p_x(-1)$ and $p_y(-1)$ in $\mathbf{s}(-1)$ are known from the GOTCHA data, and the initial velocities $v_x(-1)$ and $v_y(-1)$ are calculated from prior position differentials. Since these values are known to a high degree of accuracy, the initial error $\mathbf{m}(-1)$ should be small. An identity matrix was found to be a sufficient initial estimate for $\mathbf{m}(-1)$. Having defined all of the components of the dynamical system, the position updates follow the standard prediction-correction equations of the Kalman filter [13].

1. The first prediction step:

$$\mathbf{s}(n|n-1) = \mathbf{A} \cdot \mathbf{s}(n-1|n-1)$$

2. Calculate the prediction error:

$$\mathbf{m}(n|n-1) = \mathbf{A} \cdot \mathbf{m}(n-1|n-1)\mathbf{A}^t + \mathbf{BQB}^t$$

3. Calculate the Kalman Gain:

$$\mathbf{K}(n) = \mathbf{m}(n|n-1)\mathbf{H}^T(n) \cdot (\mathbf{C}(n) + \mathbf{H}(n) \cdot \mathbf{m}(n|n-1)\mathbf{H}^T(n))^{-1}$$

4. Correct the present state with information from the measurement:

$$\mathbf{s}(n|n) = \mathbf{s}(n|n-1) + \mathbf{K}(n) \cdot (\mathbf{o}(n) - \mathbf{H}(n)\mathbf{s}(n|n-1))$$

5. Update the MSE

$$\mathbf{m}(n|n) = (\mathbf{I} - \mathbf{K}(n)\mathbf{H}(n)) \cdot \mathbf{m}(n|n-1)$$

The SARNAV information is input into the system at Step 4. The measurement $\mathbf{o}(n)$ represents the x and y localization fix from SARNAV outputs. The measured GOTCHA data used in Chapter 7 has the platform position for each pulse, allowing us to properly test the tracking accuracy of the Kalman filter system.

Results

This chapter contains the results for testing out the two localization systems. First, results are presented for the sharpness localization system using the synthetic data generator. No synthetic data results are presented for the correlation localization system because it was conceived of after transitioning to measured GOTCHA data. Next, the localization results for both systems are presented using measured data. Finally, Kalman Filter results are shown for the correlation-based localization system.

7.1 Synthetic Data

7.1.1 Synthetic Data Generation

In an attempt to build knowledge and understanding about SAR systems, a tool was developed to generate synthetic SAR data for a scene with scatterers at arbitrary locations in 3D space. The system takes in a list of 3D points and outputs phase history data that is ready to be processed.

A point p is defined as

$$\mathbf{p}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}, \quad (7.1)$$

corresponding to the location of a scatterer. The phase history $S(\mathbf{f}_k, \tau_n)$ is generated for each $\mathbf{f}_k$ of K frequencies in the chirp, each aperture location $\mathbf{r}(\tau_n)$ of N slow-time pulses, and each $\mathbf{p}_i$ of I point scatterers. $\mathbf{r}(\tau_n)$ is relative to the motion compensation point. The phase history was generated [4] as

$$S(\mathbf{f}_k, \tau_n) = \sum_{k=0}^{K-1} A_i e^{\frac{-j4\pi f_k (||\mathbf{r}(\tau_n) - \mathbf{p}_i|| - ||\mathbf{r}(\tau_n)||)}{c}}, \quad (7.2)$$

where A_i is a noiseless constant.

Using the synthetic data generator, phase history for a sample scene was created, as seen below in Figure 7.1. The synthetic scene consists of 7 buildings with heights ranging from 5 to 20 meters in height, and each building has a total 25 scatterers on its 2 meter by 2 meter top face. The patch containing the buildings is 20 meters by 20 meters. The parameters for the radar simulation were copied from the GOTCHA radar system, and the details are listed in Table 7.1.

7.1.2 Simulation Results

Using the phase history generated from the synthetic scene, experiments with the sharpness localization system were performed using different synthetic aperture sizes. In these experiments, the sharpness localization system was tested over an uncertainty region $\mathcal{R}$ sized ± 250 m in the x and y dimensions with a resolution of 10 m, and the results are presented as a heatmap of the sharpness objective function over candidate position estimates $\hat{\boldsymbol{\theta}}$. The

(a)

(b)

Figure 7.1: Synthetic scene (a) top view (b) side view. The red circles indicate locations of point scatterers.

ground truth position lies at $\theta = [0, 0]^T$ which indicates a perfect position estimate in this experiment. Accordingly, successful localization occurs when the heatmap peaks at $(0, 0)$ in the center of the image.

Figure 7.2(a) shows the heatmap generated using a 5° synthetic aperture for the sharpness localization system. Here, one can see that the heatmap is stronger near the center, but the plateau is rather large and the true position is not the absolute global maximum. It does, however, indicate that the sharpness metric does in fact quantify some degree of localization accuracy.

The theory presented in Chapter 3 suggests that larger synthetic apertures should yield better localization performance for the sharpness method. The following experiment explores this by using an aperture that is three times larger than the previous case. Figure 7.2(b) shows the heatmap generated from a 15° synthetic aperture using the same conditions as the 5° case. As the theory predicted, the localization performance greatly increased. The global maximum was at $\hat{\theta} = [0, 0]^T$, with a far greater peak-to-sidelobe ratio than in the prior test. The peak also tightened around the ground truth position. These experiments demonstrated promise that the sharpness method could be effective for platform localization; however, it is only likely to be effective for large apertures. Additionally, we call specific attention to the non-convex nature of the utility surfaces in Figure 7.2(a) and Figure 7.2(b). This indicates that optimization techniques to find the global maximum are susceptible to false peaks if not properly handled.

7.2 Measured GOTCHA Data

Here we present localization results for the proposed methods using measured SAR data. We used phase history data from the X-band GOTCHA radar [1] for SAR measurements and a georegistered LIDAR pointcloud as the DEM. Both sensors collected wide-area data over a common region. However, the SAR data was collected two years after the LIDAR

(a)

(b)

Figure 7.2: Localization heatmaps for aperture sizes (a) 5° and (b) 15° .

data. Details of the sensors and collections are given in Tables 7.1 and 7.2.

Figure 7.3: (a) Target building DEM and (b) SAR image used in correlation-based localization.

Collection date	August 2008
Flight geometry	circular, 15 km diameter
Spot size	5 km diameter
Frequency	X-band, $F_c = 9.7$ GHz
Bandwidth	600 MHz
PRF	7.5 kHz
Range resolution	25 cm
Depression angle	47°

Table 7.1: GOTCHA radar parameters

Collection date	July 2006
Horizontal accuracy	.5 m
Vertical accuracy	.5 cm
PRF	52.3 kHz

Table 7.2: LIDAR parameters

7.2.1 Localization Results

System Comparison

As an initial test, we evaluated the sharpness-based and correlation-based systems over a small flight path consisting of a 0.75° aperture (6144 pulses) and then a larger 10.4° aperture (86,016 pulses). While the true position of the platform is known for evaluation purposes, the algorithms were given an uncertainty region $\mathcal{R}$ sized ± 100 m in the x and y dimensions, centered around the true location. Accordingly, an estimate of $\hat{\boldsymbol{\theta}} = [0, 0]^T$ indicates a perfect position estimate in this experiment. For the correlation-based method, a cluster of three buildings was identified in $\mathcal{R}$ and used as the elevated feature. DEM and SAR images of the building complex are shown in Figures 7.3(a) and 7.3(b), respectively.

Figure 7.4 illustrates surface plots evaluating the localization metrics for the sharpness method (4.5) and the correlation method (4.16). The metrics are evaluated over the $200 \text{ m} \times 200 \text{ m}$ uncertainty region $\mathcal{R}$ on a grid with 5 m spacing—hence estimates are quantized to 5 m. The small-aperture results in the top row of the figure appear noiselike with no discernable peak for the sharpness method; while the correlation method has a well-defined peak at the proper location $[0, 0]^T$. Consistent with the theory and simulation results presented above, the sharpness method does peak at larger apertures, and for the 10.4° aperture case depicted in Figure 7.4(c), we observe a global maximum at the correct location (and a local maximum displaced approximately 50 m). For the larger aperture, the correlation-based method continues to be maximized at the proper location and displays a slightly better peak-to-sidelobe ratio than for the smaller aperture case.

The experimental results indicate that relatively large apertures are required to achieve good localization performance using the sharpness method. Further, because each evaluation of the sharpness metric (4.5) requires the computation of a new backprojection image, the computational complexity for processing larger apertures becomes prohibitive. In contrast, for the correlation method, the entire cost surface may be computed from a single

Figure 7.4: Comparison of localization methods. (a) sharpness method 0.75° aperture, (b) correlation method 0.75° aperture, (c) sharpness method 10.4° aperture, (d) correlation method 10.4° aperture.

backprojection image. In the examples presented here, the sharpness-based approach took approximately two orders of magnitude more time to evaluate. Since the correlation-based method outperforms the sharpness approach in nearly all aspects, in the next section we focus our analysis on this method in a wider array of operating conditions.

Testing different aperture positions

The correlation-based examples presented in this section continue to use the three-building target shown in Figure 7.3, however similar results were obtained using other in-scene buildings as the targeted feature. In the first experiment, we localized the radar platform at ten distinct points along a 9 km flight path. The true path of the aircraft is shown by the blue line in Figure 7.5, and the estimated points are shown by the red dots. In this experiment, the uncertainty region $\mathcal{R}$ was a $300 \text{ m} \times 300 \text{ m}$ square area containing the true location, and 1.5° of aperture (2000 pulses) were used in backprojection imaging. Hypothesized positions were evaluated on a 5 m grid, making estimates quantized to this limit. Each point location was estimated independently, not aided by any prior information from, e.g., a tracking algorithm. Among the estimates, nine of the ten point locations were estimated perfectly (made possible due to the quantization grid falling on the true locations), and only one point experienced a 5 m error.

This example highlights a more real-world application where a platform is localized at various points along an extended flight path, as needed, for example, to provide periodic corrections to INS drift when GPS is unavailable. Because the look-angle of the radar changes by 67° over the 9 km flight path, this example also demonstrates robustness to variations in aperture-scene geometry. This is accomplished by the elevated backprojection plane (Step 3 above) which prevents aspect-dependent layover that would be present in ground-plane imagery.

Figure 7.5: Using 1.5° apertures, the correlation method accurately estimates 10 points along a 9 km flight path.

Cold Start Localization

As final experimental analysis, we considered a very large $4 \text{ km} \times 4 \text{ km}$ uncertainty region $\mathcal{R}$ and explored the number of pulses required to achieve good localization performance. Accommodating such a large uncertainty region is important in cold-start scenarios where reliable position priors are not available. Figure 7.6 illustrates the localization error $\|\theta - \hat{\theta}\|$ versus the number of pulses in the synthetic aperture. A distinct threshold is observed at 168 pulses. Below this threshold, spurious peaks in the correlation surface generate large localization errors. For synthetic aperture sizes greater than the threshold, the proper peak is identified and localization error is minimal, roughly equal to the size of the quantization grid. While the threshold of 168 pulses implies a very low cross-range resolution, likely unsuitable for visual interpretation of the scene, it is sufficient for localization purposes due to the processing gain of the correlation process aided by the use of template matching. The sufficiency of the small aperture in this example lends promise to a real-time implementation of the localization algorithm.

Figure 7.6: Localization performance versus aperture size for a $4 \text{ km} \times 4 \text{ km}$ uncertainty region. Above the threshold of 168 pulses, low localization error is achieved.

7.2.2 Kalman Filtering Results

As described in the previous chapter, the system was incorporated into a Kalman filter tracker to close the loop and make it a system that works with continuous updates. Here we present the results from using a Kalman filter with the correlation localization system described in Section 4.2, specifically using the template matching. In an initial experiment, we extracted a small portion of the flight path ($.7308^\circ$) from the full GOTCHA data and attempted to track the aircraft's location with a high update rate. The tracker position was updated every .0267 seconds and evaluated by comparing position estimates to the known position provided with the GOTCHA data. The uncertainty region $\mathcal{R}$ was 500 meters by 500 meters, the image resolution was 1 meter, the heatmap resolution was 1 meter, and the synthetic aperture used for each measurement was $.0244^\circ$.

Figure 7.7 shows the 2D tracking results from the experiment. The blue line represents the SARNAV localization measurements, the red line represents the smoothed state of the system, and the yellow line represents the ground truth.

Perhaps a better representation of this experiment is to observe how each coordinate evolves over time. Figure 7.8 and Figure 7.9 show the filtered localization for the x coordi-

Figure 7.7: Kalman filter results for correlation-based localization.

nate and the y coordinate over time. They show the same red, blue, and yellow lines as in Figure 7.7, but they also show the measurement error of each coordinate over time. The red bars around the state encapsulate the coordinate’s Kalman filter state, ± 1 standard deviation. This value was calculated from the mean squared error that is tracked with the state from the standard prediction-update equations in Chapter 6. In this experiment, the largest error in x was 1.01 meters, and the largest error in y was 2.50 meters. The difference here is likely because at the aperture position used in this experiment, the platform was flying mostly in the y direction.

The initial experiment tested the accuracy of the tracking system over a small aperture with a high update rate. Another experiment was conducted to measure how well the tracking system would operate over a larger aperture with less frequent updating. The same $\mathcal{R}$, image and heatmap resolutions, and synthetic aperture size were used from the first experiment, but now the measurement rate was increased from .0267 seconds to .8192 seconds, and the aperture was extended from $.7308^\circ$ to 6.735° . We let the Kalman filter drift for 6143 pulses (.8191 seconds) without providing a localization update. In practice, this could rep-

Figure 7.8: X coordinate from small flight path experiment, showing the ground truth, the SARNAV measurement, and the state

Figure 7.9: Y coordinate from small flight path experiment, showing the ground truth, the SARNAV measurement, and the state

Figure 7.10: X coordinate from large flight path experiment, showing the evolution of the state alongside the ground truth

resent time where another sensor, such as an IMU, is providing relative position update information. Figures 7.10 and 7.11 show how the x and y coordinate evolve over time, and Figures 7.12 and 7.12 show the accompanying x and y prediction error, presented using the standard deviation. At time $t = 0$, the first measurement is made, and thus the state is near the truth. Then, the state and prediction error drift until the next measurement. Each measurement corrects the state back to the ground truth, and as consecutive measurements are made, the magnitude of the state drift dampens, such that by $t = 5$ seconds, the largest drift error is 3.45 meters.

One other important result from this experiment is the amount of time needed to perform a localization. In this tracker example, each SARNNAV estimate took 10 seconds to be computed on a modern laptop computer with no parallelization, all of the way from computing the 500 x 500 pixel backprojection image to creating the template matching heatmap. We are confident that this number could be dramatically reduced since backpro-

Figure 7.11: Y coordinate from large flight path experiment, showing the evolution of the state alongside the ground truth

Figure 7.12: Prediction error for the X coordinate in the large flight path experiment

Figure 7.13: Prediction error for the Y coordinate in the large flight path experiment

jection can easily be made parallel.

These results show that high-accuracy localization tracking can be achieved using SAR-NAV with a Kalman filter. A platform's absolute global position can be accurately tracked, even when measurements are taken at lower sampling rates. Further, these experiments provide strong evidence that SARNAV has potential to bolster localization system performance on platforms equipped with the necessary equipment.

Conclusions and Future Work

In this thesis, we presented two methods for using the properties of 3D SAR imaging to provide absolute global positioning information to airborne platforms. The first method quantified DEM and SAR alignment using image sharpness, while the second approach considered correlations between a georegistered DEM and elevated-plane imagery. The correlation-based method was shown to provide an approximate maximum likelihood estimate of the platform position and had low computational complexity, requiring only a single backprojection per positioning operation. Experiments with measured GOTCHA radar data demonstrated the effectiveness of the approaches and yielded localization errors less than a few meters for initial uncertainty regions as large as 16 km^2 . Further development and experiments with a Kalman filter demonstrated SARNAV's potential as a continuous tracker.

Future work will focus on generating posterior probabilities for the position estimates in order to fuse the SAR-based solutions with other positioning sources, such as INS and EO based systems. Additionally, more analysis can be performed into the robustness of the algorithms. For one, there has been very limited analysis into using multiple scene features. There is expected to be significant improvements in precision, size of uncertainty region, and robustness to noise upon using multiple features. Future work could also consider directly incorporating IMU measurements into the Kalman filter.

Bibliography

- [1] 2008 Gotcha Radar Dataset. [Online]. Available: <https://www.sdms.afrl.af.mil>.
- [2] GPS accuracy. <https://www.gps.gov/systems/gps/performance/accuracy>. Accessed: 2018-05-25.
- [3] K. Briechle and U. D. Hanebeck. Template matching using fast normalized cross correlation. In *Proc. SPIE*, volume 4387, 2001.
- [4] P. H. Eichel D. C. Ghiglia C. V. Jakowatz, D. E. Wahl and P. A. Thompson. *Spotlight-mode synthetic aperture radar: a signal processing approach*. Springer, 1996.
- [5] M. Cannon. TERCOM performance: Analysis and simulation. Technical Report AMRL-TR-73-130, U.S. Aerospace Medical Research Laboratory, Wright Patterson AFB, Ohio, June 1974.
- [6] J.R. Fienup and J.J. Miller. Aberration correction by maximizing generalized sharpness metrics. *J. Opt. Soc. Am. A*, 20(4):609–620, 2003.
- [7] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, volume 24, 1981.

- [8] J. P. Golden. Terrain contour matching (TERCOM): A cruise missile guidance aid. In *Proc. SPIE*, volume 0238, 1980.
- [9] A. Henley. Terrain aided navigation: current status, techniques for flat terrain and reference data requirements. In *Proc. IEEE, Position Location and Navigation Symposium*, Mar. 1990.
- [10] L. J. Moore K. E. Dungan, L. A. Gorham. SAR digital spotlight implementation in MATLAB. In *Proc. SPIE*, volume 8746, 2013.
- [11] Y. Morton K. Kauffman, J. Raquet and D. Garmatyuk. Real-time UWB-OFDM radar-based navigation in unknown terrain. *IEEE Transactions on Aerospace and Electronic Systems*, 49(3), July 2013.
- [12] D. Kaufman. New supercomputer supports intelligence, surveillance, and reconnaissance research and development. <http://www.wpafb.af.mil/News/Article-Display/Article/400458/new-supercomputer-supports-intelligence-surveillance-and-reconnaissance-researc>, 2009. Accessed: 2018-04-20.
- [13] S. Kay. *Fundamentals of statistical signal processing*. Prentice-Hall, 1993.
- [14] T.J. Kragh. Monotonic iterative algorithm for minimum-entropy autofocus. In *Proc. Adaptive Sensor Array Processing Workshop*, June 2006.
- [15] I. Leveson. The economic value of GPS: Preliminary assessment. <https://www.gps.gov/governance/advisory/meetings/2015-06/leveson.pdf>. Accessed: 2018-05-20.
- [16] L. Moore and U. Majumder. An analytical expression for the three-dimensional response of a point scatterer for circular synthetic aperture radar. In *Proc. SPIE, Algorithms for Synthetic Aperture Radar Imagery XVII*, volume 7699, 2010.

- [17] L.J. Moore and L.C. Potter. Three-dimensional resolution for circular synthetic aperture radar. In *Proc. SPIE, Algorithms for Synthetic Aperture Radar Imagery XIV*, volume 6568, 2007.
- [18] L.J. Moore, L.C. Potter, and J.N. Ash. Three-dimensional position accuracy in circular synthetic aperture radar. *IEEE Aero. Elect. Sys. Mag.*, 29(1), Jan. 2014.
- [19] P. C. Niedfeldt and R. W. Beard. Recursive RANSAC: Multiple signal estimation with outliers. In *IFAC Proceedings Volumes*, 2013.
- [20] E. B. Quist P. C. Niedfeldt and R. W. Beard. Characterizing range progression of SAR point scatterers with recursive RANSAC. In *Proc. IEEE Radar Conference*, May 2014.
- [21] E. B. Quist and R. W. Beard. Radar odometry on fixed-wing small unmanned aircraft. *IEEE Transactions on Aerospace and Electronic Systems*, 52(1), February 2016.
- [22] Z. Reid and J. N. Ash. Leveraging 3D models for SAR-based navigation in GPS-denied environments. In *Proc. SPIE*, volume 10647, pages 10647 – 10647 – 11, 2018.
- [23] G. M. Siouris. *Missile Guidance and Control Systems*. Springer, 2004.
- [24] Q. Zhang and L. Zhao. Selection method of terrain matching area for tercom algorithm. In *Proc. SPIE*, volume 10458, 2017.