Multi-Baseline Interferometric Sar for Iterative Height Estimation

Adam E. Robertson
Brigham Young University - Provo

Follow this and additional works at: https://scholarsarchive.byu.edu/etd
Part of the Electrical and Computer Engineering Commons

BYU ScholarsArchive Citation
https://scholarsarchive.byu.edu/etd/4248

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
MULTI-BASELINE INTERFEROMETRIC SAR FOR ITERATIVE
HEIGHT ESTIMATION

by

Adam E. Robertson

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

Department of Electrical and Computer Engineering
Brigham Young University
December 1998
BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Adam E. Robertson

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date  

David Arnold, Chair

Date  

David Long

Date  

Brian Jeffs
BRIGHAM YOUNG UNIVERSITY

As chair of the candidate’s graduate committee, I have read the thesis of Adam E. Robertson in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

David Arnold
Chair, Graduate Committee

Accepted for the Department

Michael D. Rice
Graduate Coordinator

Accepted for the College

Douglas M. Chabries
Dean, College of Engineering and Technology
ABSTRACT

MULTI-BASELINE INTERFEROMETRIC SAR FOR ITERATIVE HEIGHT ESTIMATION

Adam E. Robertson
Department of Electrical and Computer Engineering
Master of Science

Multiple SAR interferograms with judiciously selected height sensitivities can be iteratively combined to create a high accuracy digital elevation map. An initial height estimate is refined by iteratively using larger baselines to obtain a height estimation accuracy limited by the spatial decorrelation of the antenna baseline. Spatial filtering is used to reduce the propagation of errors for accurate height estimation. Images containing regions isolated by phase discontinuities, as often found in urban environments, can be resolved by this iterative multi-baseline technique. Computationally demanding and potentially unreliable phase unwrapping is not required to determine scene elevation using SAR interferometry.
ACKNOWLEDGMENTS

Thanks to Dennis Ghiglia and Mark Pritt for the use of their data on Isolation Peak Colorado. Their book on two-dimensional phase unwrapping provided valuable insights into coding phase unwrapping algorithms.

I would like to express my appreciation to Gayle Miner for his outstanding example. His enthusiasm for electrical engineering, his kindness to students, his deep humility, his dedication to family, community, and to God will be a model for my life.

I thank David Arnold, my advisor and mentor, for his faith in my abilities. His faith in my ability to perform research exceeded my own by several orders of magnitude. David’s encouragement and insight enabled me to perform a great work largely independent of other’s help.

Finally, and most significantly, I express my love and gratitude for my wife and family. Tina has kept me fed and clothed and freely given her encouragement when I have been discouraged in my research efforts. My family has been patient in spending many lonely evenings and Saturdays without a husband or father. It is to them that I dedicate this work.
Contents

Acknowledgments vi
List of Tables ix
List of Figures xii
1 Introduction 1

2 Synthetic Aperture Radar Basics 5
  2.1 SAR Imaging Geometry .......................... 5
  2.2 Pulse Transmission and Reception ................ 7
  2.3 Range Compression ............................... 10
  2.4 Azimuth Compression ............................. 17
    2.4.1 Derivation of the Quadratic Phase Term ........ 19
    2.4.2 Azimuth Matched Filter ........................ 20
    2.4.3 PRF Requirements ............................. 23
    2.4.4 Azimuth Resolution and Verification of the Mathematical Model 24

3 Interferometry Basics 27
  3.1 Interferometry Geometry ........................... 27
  3.2 Interferogram Statistics ........................... 31
  3.3 Choosing Optimal Interferometric Imaging Parameters .... 35
    3.3.1 Height Sensitivity with Respect to System Parameters . . 37
    3.3.2 Choosing the Optimal Baseline Length and Baseline Angle .. 38
    3.3.3 Other Considerations ........................... 39
  3.4 Interferometer Design Example: YINSAR .......................... 41
4 Phase Unwrapping Techniques

4.1 Mathematical Derivation of the Least Squares Solution
    4.1.1 Least Squares Development
    4.1.2 2-D Discrete Cosine Transform Solution
    4.1.3 Unweighted Least Squares Algorithm (ULS)
    4.1.4 Examples of the Unweighted Least Squares Algorithm

4.2 Model for the Weighted Least Squares Solution
    4.2.1 Hunt’s Work
    4.2.2 Extending Hunt’s Work to the Weighted Case
    4.2.3 Efficient Algorithms
    4.2.4 Preconditioned Conjugate Gradient Algorithm
    4.2.5 Examples of the the Weighted Least Squares Algorithm
    4.2.6 Summary of the Least Squares Method

4.3 Minimum Discontinuity Method for Phase Unwrapping
    4.3.1 Introduction to Flynn’s Algorithm
    4.3.2 Minimum Discontinuity Criterion
    4.3.3 Flynn’s Method For Computing the Minimum Weighted Discontinuity Solution
    4.3.4 Computation and Memory Requirements of Flynn’s Algorithm

4.4 Patches to Speed up Flynn’s Method
    4.4.1 Sewing the Patches
    4.4.2 Examples of the Patch Modification
    4.4.3 Disadvantages and Future Work

4.5 Linking the Least Squares Solution and the Minimum Discontinuity Solution

5 Multiple Baseline Height Resolution Enhancement

5.1 Prior Work

5.2 Iterative Approach to Height Estimation
    5.2.1 Problem Definition and Setup
5.2.2 Method for Reducing the Height Ambiguity ............... 95
5.2.3 Baseline Length Selection for Iterative Height Estimation .... 98
5.2.4 Filters for Iterative Height Estimation ...................... 99
5.3 Case Study for YINSAR ......................................... 101
  5.3.1 Results from an Urban Scene ............................ 102
  5.3.2 Results from a Natural Scene ............................. 122
5.4 Other Issues .................................................. 128
  5.4.1 Design Trade Offs ......................................... 128
  5.4.2 Non-Iterative Methods ..................................... 128
  5.4.3 Partially Avoiding Phase Unwrapping ..................... 128
  5.4.4 Image Registration and Georectification ................. 130
  5.4.5 Multiple Look Angles for Complete Imaging of Urban Scenes 131
  5.4.6 Utilizing Range Dependent Height Sensitivity .......... 131

6 Conclusions ...................................................... 133
  6.1 Personal Observations from Research Experience ............ 133
  6.2 Summary of Contributions ................................... 135
  6.3 Future Research ............................................. 136

Bibliography ..................................................... 140
List of Tables

2.1 YINSAR Parameters ........................................... 25
4.1 Matrix Dimensions: Unweighted Least Squares ............... 63
4.2 Matrix Dimensions: Weighted Least Squares .................. 66
5.1 Interferometric Imaging Geometry .............................. 101
5.2 Urban Scene: Iteration Results for $P_{\text{error}} = 0.05$, Filter: mean .... 106
5.3 Urban Scene: Iteration Results for $P_{\text{error}} = 0.01$, Filter: mean .... 111
5.4 Urban Scene: Iteration Results for $P_{\text{error}} = 0.10$, Filter: mean .... 111
5.5 Urban Scene: Iteration Results for $P_{\text{error}} = 0.05$, Filter: median ... 115
5.6 Urban Scene: Iteration Results for $P_{\text{error}} = 0.01$, Filter: median ... 115
5.7 Isolation Peak: Iteration Results for $P_{\text{error}} = 0.01$, Filter: mean .... 123
5.8 Isolation Peak: Iteration Results for $P_{\text{error}} = 0.01$, Filter: median .. 125
5.9 Guided Phase Unwrapping of The Urban Scene ............... 129
List of Figures

2.1 SAR Imaging Geometry .............................................. 6
2.2 SAR Footprint ......................................................... 7
2.3 LFM Chirp .............................................................. 8
2.4 Return From a Point Target ......................................... 10
2.5 Simulated Range Compression v.s. Mathematical Model for Range Compression .............................................. 16
2.6 Range Compressed Point Target ..................................... 17
2.7 Quadratic Azimuth Chirp .............................................. 20
2.8 Azimuth Chirp Length ................................................. 22
2.9 Range and Azimuth Compressed Point Target ..................... 23
3.1 Interferometric Imaging Geometry .................................. 28
3.2 2π Wrapped Image of Isolation Peak ............................... 30
3.3 pdf(θ) for Several Values of |γ| ...................................... 32
3.4 Empirically Measured pdf(θ) for Several Values of |γ| .............. 33
3.5 Comparison of Empirically Measured and Theoretical pdf(ϕ) ....... 34
3.6 Phase Standard Deviation, σφ vs Correlation Coefficient, |γ| ....... 35
3.7 SNR vs Phase Standard Deviation .................................. 36
3.8 Sensitivity vs Baseline Tilt .......................................... 40
3.9 Sensitivity and Standard Deviation vs Baseline Length ............. 42
3.10 Critical Perpendicular Baseline Length vs Incidence Angle ....... 43
3.11 Sensitivity vs Incidence Angle for Several Baseline Tilts .......... 44
3.12 Height Standard Deviation vs Incidence Angle for Several Baseline Tilts 45
3.13 Height Sensitivity and Height Standard Deviation for Two Possible YINSAR Baseline Angles. ........................................ 46
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>ULS algorithm: Noiseless Branch cut free Example</td>
<td>55</td>
</tr>
<tr>
<td>4.2</td>
<td>ULS algorithm input: Additive Gaussian Noise</td>
<td>56</td>
</tr>
<tr>
<td>4.3</td>
<td>ULS algorithm output: Additive Gaussian Noise</td>
<td>57</td>
</tr>
<tr>
<td>4.4</td>
<td>ULS algorithm input: Patch of additive Gaussian Noise</td>
<td>57</td>
</tr>
<tr>
<td>4.5</td>
<td>ULS algorithm output: Patch of additive Gaussian Noise</td>
<td>58</td>
</tr>
<tr>
<td>4.6</td>
<td>ULS algorithm input: Patch Replaced by Uniform Noise</td>
<td>58</td>
</tr>
<tr>
<td>4.7</td>
<td>ULS algorithm output: Patch Replaced by Uniform Noise</td>
<td>58</td>
</tr>
<tr>
<td>4.8</td>
<td>ULS algorithm input: Branch Cut</td>
<td>59</td>
</tr>
<tr>
<td>4.9</td>
<td>ULS algorithm output: Branch Cut</td>
<td>59</td>
</tr>
<tr>
<td>4.10</td>
<td>WLS algorithm input: Path of additive Gaussian Noise</td>
<td>60</td>
</tr>
<tr>
<td>4.11</td>
<td>WLS algorithm output: Path of additive Gaussian Noise, 1st iteration</td>
<td>60</td>
</tr>
<tr>
<td>4.12</td>
<td>WLS algorithm output: Path of additive Gaussian Noise, 3rd iteration</td>
<td>60</td>
</tr>
<tr>
<td>4.13</td>
<td>WLS algorithm input: Path of additive Gaussian Noise</td>
<td>60</td>
</tr>
<tr>
<td>4.14</td>
<td>WLS algorithm output: Path of additive Gaussian Noise, 1st iteration</td>
<td>60</td>
</tr>
<tr>
<td>4.15</td>
<td>WLS algorithm output: Path of additive Gaussian Noise, 3rd iteration</td>
<td>60</td>
</tr>
<tr>
<td>4.16</td>
<td>WLS algorithm output: Path of additive Gaussian Noise, 10th iteration</td>
<td>60</td>
</tr>
<tr>
<td>4.17</td>
<td>Elementary Operation Example 1</td>
<td>78</td>
</tr>
<tr>
<td>4.18</td>
<td>Elementary Operation Example 2</td>
<td>79</td>
</tr>
<tr>
<td>4.19</td>
<td>Elementary Operation Example 3</td>
<td>80</td>
</tr>
<tr>
<td>4.20</td>
<td>Edges and Nodes for the Example Array φ</td>
<td>80</td>
</tr>
<tr>
<td>4.21</td>
<td>Matrix Size vs. Execution Time for Flynn’s Algorithm</td>
<td>81</td>
</tr>
<tr>
<td>4.22</td>
<td>Matrix Size vs. Execution Time, Flynn’s Algorithm vs. Patch Modification</td>
<td>82</td>
</tr>
<tr>
<td>4.23</td>
<td>Unweighted Minimum Discontinuity Solution for Isolation Peak</td>
<td>85</td>
</tr>
<tr>
<td>4.24</td>
<td>Unweighted Minimum Discontinuity Solution for Isolation Peak using Four Patches</td>
<td>86</td>
</tr>
<tr>
<td>4.25</td>
<td>Height Difference Between Flynn Algorithm and Patch Modification for Isolation Peak</td>
<td>87</td>
</tr>
<tr>
<td>4.26</td>
<td>Comparison of Flynn’s Algorithm and the Patch Algorithm for Isolation Peak</td>
<td>88</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

This thesis outlines a method for resolving the height of a scene using $N + 1$ antennas to generate $N$ interferometric SAR images. An initial estimate of the height is iteratively refined using interferograms with more accurate height measurements. Spatial filters reduce errors in the height estimate to a tolerable level. The need for phase wrapping can be minimized if not entirely eliminated and the height of a scene can be unambiguously resolved, even when the scene contains regions isolated by phase discontinuities, such as often found in urban environments.

Synthetic Aperture Radar (SAR) is an instrument that is capable of high resolution mapping. SARs are typically carried on a spacecraft or an aircraft. In the past few decades SAR imaging has matured considerably, and is even finding its way into some undergraduate curriculums. During the late 1980’s [1],[2], and more especially in the 1990’s [3],[4],[5],[6],[7],[8] SAR interferometry has gained popularity as a method for creating digital elevation maps, or determining changes in surface elevation (differential interferometry [2]). The largest obstacle in determining elevation is unwrapping the phase data which contains the height information. Low SNR, signal decorrelation, and step discontinuities in the scene are the largest sources of difficulty for two dimensional phase unwrapping of interferometric data.

Several robust methods that are capable of handling low SNR regions or areas of high signal decorrelation have been developed [9],[10],[11],[12]. The two primary methods are path following and $L_p$-norm [13]. Each method has associated advantages and disadvantages in terms of speed and accuracy [14]. None of the existing methods, however, are capable of handling any region of the image that is isolated by phase
discontinuities. Such isolated regions do not occur as often in nature, but are prevalent in urban areas where buildings form regions isolated by large step discontinuities, (rooftops for example).

In recent years several papers have approached the idea of using multiple interferograms to resolve the height and otherwise avoid phase unwrapping [15],[16],[17],[18],[19],[20]. Unlike previously proposed multi-baseline techniques, the iterative method presented in this thesis directly addressed the issue of phase noise and error propagation. Spatially filtering each new height estimation reduces the phase noise to an acceptable level. The new method has the potential to map scenes containing regions isolated by phase discontinuities, which are particularly characteristic of urban environments.

This thesis is organized in the following fashion:

- **Chapter 1** is a brief introduction to SAR Interferometry and the significance of the work presented in this thesis.
- **Chapter 2** is a rigorous derivation of range and azimuth compression for dual side band synthetic aperture radar transmitting a linear frequency modulated chirp. Range and azimuth resolution are derived, along with the minimum range and azimuth sampling rates. The results are compared with common approximations.
- **Chapter 3** provides an introduction to interferometry. The imaging geometry of interferometry is explained and the effects of perturbations of the imaging parameters on the accuracy of the height measurement are presented. The sources and the effects of decorrelation for interferometric SAR are discussed. The theoretical phase decorrelation due to thermal noise is verified with empirical measurement from the YINSAR interferometric instrument. As an example of interferometer design, the optimal imaging geometry for YINSAR is determined.
- **Chapter 4** is an outline of two common phase unwrapping techniques. The weighted least squares method is outlined as an example of a $L_p$-norm technique, and an extension of Hunt’s matrix formulation [21] to the weighted case is presented. The minimum discontinuity method is outlined as a path following technique
and a computationally more efficient modification to Flynn’s minimum discontinuity method is presented.

- **Chapter 5** introduces an iterative method for height estimation using $N$ interferograms. The height of a scene can be determined unambiguously through a careful choice of baseline lengths and an imaging geometry based on the anticipated phase decorrelation. The proposed method does not require phase unwrapping and has particular appeal for urban scenes where traditional phase unwrapping techniques fail. Examples of the application of the method to a simulated urban environment and to a natural scene are shown.

- **Chapter 6** summarizes the contributions made in this thesis. It concludes that multi-baseline techniques using $N$ interferograms for iterative refinement of the height estimation hold a promising future for digital elevation mapping of urban and non-urban areas, and suggests avenues to pursue for future research.
Chapter 2

Synthetic Aperture Radar Basics

A mathematical description of SAR data compression is essential to understanding advanced SAR topics such as range migration, range curvature, motion compensation, and interferometry. The mathematical derivation of range and azimuth matched filters for dual side band strip map mode SAR are presented in this section. The derivation is for the ideal, or uniform case. The assumptions necessary to attain a closed form solution to the range matched filter are given and justified by comparison with computer simulated matched filtering. The comparison shows that the mathematical results are as accurate as the assumptions made. Comparison of the predicted range and azimuth resolution from the mathematical derivation agree with generally accepted approximations for SAR resolution.

2.1 SAR Imaging Geometry

Figure 2.1 illustrates the geometry of a SAR. The variables defined in Fig. 2.1 will be used in the mathematical description of SAR that follows. The $x$, $y$, and $z$ direction are called the azimuth, range, and elevation respectively. The radar platform moves at a velocity, $V$ in the azimuth direction at a nominal elevation above the earth, of $H$. $R$ is the distance from the radar to a target at $(x_o, y_o)$. The minimum distance from the the radar line of flight to $(x_o, y_o)$ is $R_o$. The radar antennas are pointed in the $y$ direction and form a ‘footprint’ on the ground as indicated in Fig. 2.2. The size of the radar footprint is determined by the antenna pattern and the direction the antennas are pointing. The range beamwidth, $\theta_r$, is typically large, covering a large swath in range. The azimuth beamwidth, $\theta_a$, is typically less than
Figure 2.1: SAR imaging geometry. The $x$, $y$, and $z$ direction are called the azimuth, range, and elevation respectively. The radar platform moves at a velocity, $V$, in the azimuth direction at a nominal elevation above the earth, $H$. $R$ is the distance from the radar to a target at $(x_o, y_o)$. The minimum distance from the radar line of flight to $(x_o, y_o)$ is $R_o$.

20°. YINSAR (Brigham Young Interferometric SAR) is an interferometer built by Brigham Young University. YINSAR system parameters will be used extensively for examples in this thesis. The parameters are listed in Table 2.1 at the end of this chapter.

A SAR transmits microwave pulses at regular intervals as it travels in the azimuth direction. Each pulse reflects off of the objects in the radar footprint, and a portion of the pulse is reflected back to the radar system where it is sampled. Because objects that are farther from the radar take longer to return, the reflected pulses are sampled for a period of time sufficient to sample returns from all the points in the antenna footprint. By repeatedly transmitting and receiving pulses as the radar moves in the azimuth direction, reflections from a large rectangular region of the $x, y$
Figure 2.2: SAR footprint. The area on the ground that is illuminated by the SAR antennas is called the antenna footprint. Its size is determined by the antenna pattern and the direction the antennas are pointing. The range beamwidth, $\theta_r$, is typically large, covering a large area in range. The azimuth beamwidth, $\theta_a$, is typically less than 20°.

plane can be sampled. A detailed mathematical description of pulse transmission, reception, and image compression follows.

2.2 Pulse Transmission and Reception

The most common transmit pulse is a linear frequency modulated chirp, or LFM. For YINSAR, the LFM is generated using an arbitrary wave form generator. An LFM is graphically described in Fig. 2.3. Mathematically, an LFM chirp is,

$$LFM(t) = \cos(\beta t^2), \quad 0 < t < T.$$  (2.1)
Figure 2.3: An linear frequency modulated chirp, or LFM. $T$ is the length of the chirp in seconds, $BW$ is the frequency bandwidth of the chirp in Hertz, and $\beta$ is the slope of the line.

$\beta$ is the ‘chirp rate’ and is defined as the rate at which the frequency increases. $\beta$ is the slope of the line in Fig. 2.3. $T$ is the length of the chirp in seconds, and $BW$ is the frequency bandwidth of the chirp in Hz. From Eq. (2.1), the phase of the LFM is given by $\theta = \beta t^2$. The rate of frequency change is determined by $\theta' = 2\beta t$. The LFM chirp bandwidth, $BW$, is the maximum value of $\theta'$, which occurs at time $t = T$, to give $BW = 2\beta T$. Solving for $\beta$ in radians where $BW$ is in Hertz,

$$\beta = \frac{BW \pi}{T}. \tag{2.2}$$

The LFM chirp is mixed up to a RF carrier frequency, $w_c$, before it is transmitted. Using the trigonometric identity $\cos(\alpha) \cos(\beta) = \frac{1}{2}[\cos(\alpha - \beta) + \cos(\alpha + \beta)]$ the transmitted signal, $x_t(t)$ is,

$$x_t(t) = \cos(w_c t) \cos(\beta t^2),$$

$$= \frac{1}{2} \left[ \cos(w_c t - \beta t^2) + \cos(w_c t + \beta t^2) \right]. \tag{2.3}$$

Mixing by $w_c$ results in a down-chirp and an up-chirp, or a double side band chirp, as indicated in Eq. (2.3). Most systems use a single side band chirp, the double side band chirp is a feature unique to YINSAR. The double side band chirp is
transmitted by the radar system. A point target a distance $2R_o = t_o c$ from the radar 
will return an attenuated and delayed copy of the transmitted chirp of the form,

$$r(t) = \frac{1}{2} A \left( \cos \left[ \omega_c(t-t_0) - \beta(t-t_0)^2 \right] + \cos \left[ \omega_c(t-t_0) + \beta(t-t_0)^2 \right] \right), \quad (2.4)$$

where $A$ indicates the amount of attenuation. For simplicity, $A = 1$, or no attenuation is assumed. The received signal, $r(t)$ is mixed by $\cos(\omega_d t)$ to an intermediate frequency, $\omega_o$, and a low pass filter is used to eliminate the high frequency terms resulting from the mixing operation,

$$r(t) \cos(\omega_d t) = \frac{1}{4} \left( \cos \left[ \omega_c(t-t_0) - \beta(t-t_0)^2 - \omega_d t \right] + \cos \left[ \omega_c(t-t_0) - \beta(t-t_0)^2 + \omega_d t \right] + \cos \left[ \omega_c(t-t_0) + \beta(t-t_0)^2 - \omega_d t \right] + \cos \left[ \omega_c(t-t_0) + \beta(t-t_0)^2 + \omega_d t \right] \right), \quad \frac{\text{LPF}}{4}$$

$$= \frac{(\cos \left[ \omega_c(t-t_0) - \beta(t-t_0)^2 - \omega_d t \right] + \cos \left[ \omega_c(t-t_0) + \beta(t-t_0)^2 - \omega_d t \right])}{4}. \quad (2.5)$$

The symbol, $\frac{\text{LPF}}{4}$, indicates the low pass filter operation that eliminates the terms where $\omega_c t$ and $\omega_d t$ are summed. At this point the signal is digitally sampled. Because sampling is performed above the Nyquist rate, the mathematical derivation can and will be carried out in the continuous domain. A Hilbert Transform is applied to Eq. (2.5) to obtain the quadrature component. Letting $\omega_c - \omega_d = \omega_o$, the result is,

$$\frac{1}{4} \left( \frac{\exp \left[ j\omega_o t - j\beta(t-t_0)^2 - j\omega_o t_0 \right]}{A(t)} + \frac{\exp \left[ j\omega_o t + j\beta(t-t_0)^2 - j\omega_o t_0 \right]}{B(t)} \right) = \frac{1}{4}(A(t) + B(t)), \quad (2.6)$$

where $A$ and $B$ are defined as indicated for notational convenience. Equation (2.6) represent the uncompressed or raw data from a synthetic aperture radar. Several signal processing steps are necessary to obtain an image from the raw data. Figure 2.4 is a plot of the real portion of a return from a point target, (raw data). The point
Figure 2.4: Real portion of the return from a point target. The underlying data can not be interpreted until range and azimuth compression are performed.

The target is located at the center of the concentric circles. The linearly increasing phase can be seen extending out in all directions from the location of the point target.

The minimum range sample rate is twice the rate of frequency change. From Eq. (2.6) the phase of the mixed down raw data and its derivative are,

\[
\phi = \omega_0 t \pm \beta(t - t_0)^2 - w_e t_0, \\
\phi' = \omega_0 \pm 2\beta(t - t_0). \tag{2.7}
\]

The maximum rate of frequency change occurs at \((t - t_0) = T\). Converting from radians to Hertz and using Eq. (2.2), the Nyquist rate or the minimum analog to digital sampling rate is,

\[
\text{AtoD}_{\text{min}} = 2(f_o + BW). \tag{2.8}
\]

For YINSAR, the intermediate frequency is \(f_o = 100\) MHz and the chirp bandwidth is \(BW = 100\) MHz to give \(\text{AtoD}_{\text{min}} = 400\) MHz.
2.3 Range Compression

The first step in SAR image formation is range compression. Range compression is simply matched filtering performed in the range direction. The range matched filter is,

\[ h(t) = \exp(jw_\alpha t - \beta \ell^2)_C(t) + \exp(jw_\alpha t + \beta \ell^2)_D(t) = C(t) + D(t), \tag{2.9} \]

where \( C \) and \( D \) are defined for notational convenience. For the sake of simplicity an infinite length matched filter is assumed, a fair assumption for large \( T \). The effect of this assumption will be seen later on. Autocorrelation, or the matched filter operation, is defined as,

\[ m(t) = s(t) \otimes h^*(t), \tag{2.10} \]

\[ \mathcal{F}(m(t)) = M(w) = S(w)H(-w), \tag{2.11} \]

\[ m(t) = \int_{-\infty}^{\infty} s(t + \tau)h^*(\tau)d\tau, \tag{2.12} \]

where \( s(t) \) is the signal of interest and \( h(t) \) is the matched filter. Using the \( A,B,C,D \) notation of Eqs. (2.6) and (2.9), the range autocorrelation is,

\[ s(t) = \frac{1}{4}(A + B), \]

\[ h(t) = (C + D), \]

\[ m(t) = s(t) \otimes h^*(t), \]

\[ m(t) = \int s(t + \tau)h^*(\tau)d\tau, \]

\[ m(t) = \int \frac{1}{4}[A(t + \tau) + B(t + \tau)][C(\tau) + D(\tau)]d\tau, \]

\[ m(t) = \frac{1}{4}[A(t + \tau)C(\tau) + A(t + \tau)D(\tau) + B(t + \tau)C(\tau) + B(t + \tau)D(\tau)]d\tau, \]

\[ m(t) = \frac{1}{4} \left( \int AC + AD + BC + BD \right). \tag{2.13} \]

Observe from Eqs. (2.6) and (2.9) that \( A(\beta) = B(-\beta) \) and \( C(\beta) = D(-\beta) \). The terms differ by only \( \pm \beta \). The \( \pm \beta \) relation will be used to relate the solutions to the
\(BC\) and \(BD\) terms, simplifying the derivation. For the sake of simplicity and ease of reading, the derivation will be carried out first on the terms \(AC\) and \(AD\). Beginning with the \(AC\) term,

\[
\int AC = \\
= \int A(t + \tau)C(\tau)d\tau, \\
= \int \exp(jw_\theta(t + \tau) - j\beta(t + \tau - t_0)^2 - jw_\theta t_0) \exp(-jw_\theta \tau + \beta\tau^2)d\tau, \\
= \int \exp(jw_\theta t + jw_\theta \tau - j\beta(t^2 + \tau^2 + t_0^2 - 2tt_0 - 2\tau t_0 + 2t\tau) - \\
jw_\theta t_0 - jw_\theta \tau + \beta\tau^2)d\tau. \\
(2.14)
\]

The terms \(\pm j\beta\tau^2\) and \(\pm jw_\theta\tau\) cancel. Pulling all the terms independent of \(\tau\) out from under the integral,

\[
\int AC = \exp(jw_\theta t - jw_\theta t_0 - j\beta(t - t_0)^2) \int \exp(-j2\beta\tau(t - t_0))d\tau. \\
(2.15)
\]

For simplification let,

\[
c_1 = \frac{1}{4} \exp(-jw_\theta t_0 + jw_\theta t), \\
c_2 = \exp(-j\beta(t - t_0)^2).
\]

Applying the limits of integration and evaluating,

\[
c_1c_2 \int_{t-t_0}^{T-t+t_0} \exp(-j2\beta\tau(t - t_0))d\tau = \\
= c_1c_2 \left. \frac{1}{-j2\beta(t - t_0)} \exp(-j2\beta\tau(t - t_0)) \right|^{\tau=T-t+t_0}_{\tau=-t+t_0}, \\
= c_1c_2 \frac{1}{-j2\beta(t - t_0)} \times \\
\left( \exp[-j2\beta(T - t + t_0)(t - t_0)] - \exp[j2\beta(t - t_0)^2] \right). \\
(2.16)
\]

Plugging \(c_2\) into Eq. (2.16),

\[
\int AC = \\
= c_1 \frac{j}{2\beta(t - t_0)} \exp(-j\beta(t - t_0)^2) \left( \exp[-j2\beta T(t - t_0) + j2\beta(t - t_0)^2] - \\
\exp[j2\beta(t - t_0)^2] \right), \\
= c_1 \frac{j}{2\beta(t - t_0)} \left( \exp[-j2\beta T(t - t_0) + j\beta(t - t_0)^2] - \exp[j\beta(t - t_0)^2] \right), \\
(2.17)
\]
which completes the evaluation of the $AC$ term of Eq. (2.13). Recalling that,

$$
\int A(-\beta)C(-\beta) = \int B(\beta)D(\beta),
$$

(2.18)

the evaluation of the $BD$ term is apparent from the evaluation of $AC$ in Eq. (2.17),

$$
\int BD = c_1 \frac{-j}{2\beta(t-t_o)} \left( \exp[j2\beta T(t-t_o) - j\beta(t-t_o)^2] - \exp[-j\beta(t-t_o)^2] \right). 
$$

(2.19)

Adding the $AC$ and $BD$ terms as in Eq. (2.13),

$$
\int AC + \int BD =
$$

$$
= \frac{j c_1}{2\beta(t-t_o)} \left[ \exp[-j2\beta T(t-t_o) + j\beta(t-t_o)^2] - \exp[j2\beta(t-t_o)^2] - \exp[j2\beta T(t-t_o) - j\beta(t-t_o)^2] + \exp[-j\beta(t-t_o)^2] \right].
$$

$$
= \frac{j c_1}{2\beta(t-t_o)} \left[ -2j \sin[2\beta T(t-t_o) - \beta(t-t_o)^2] - 2j \sin[\beta(t-t_o)^2] \right].
$$

(2.20)

Using the trigonometric identity $\sin \alpha \pm \sin \beta = 2 \sin \frac{\alpha \pm \beta}{2} \cos \frac{\alpha \pm \beta}{2}$,

$$
\int AC + \int BD
$$

$$
= \frac{2c_1}{\beta(t-t_o)} \left[ \sin \left( \beta T(t-t_o) \right) \cos \left( \beta T(t-t_o) - \beta(t-t_o)^2 \right) \right],
$$

$$
= 2T \exp(-j w_\tau t_o + j w_\tau t) \text{sinc}[\beta T(t-t_o)] \cos[\beta T(t-t_o) - \beta(t-t_o)^2],
$$

(2.21)

where $\text{sinc}(x) = \frac{\sin(x)}{x}$. The $AD$ and $BD$ terms in Eq. (2.13) remain to be dealt with. Evaluating the $AD$ term,

$$
\int AD =
$$

$$
= \int A(t+\tau)D(\tau)d\tau,
$$

$$
= \int \exp[j w_\tau(t+\tau) - j\beta(t+\tau-t_o)^2] \exp[-j w_\tau t - j\beta t^2]d\tau,
$$

$$
= \int \exp[j w_\tau t + j w_\tau \tau - j\beta \tau^2 - 2j\beta(t-t_o)\tau - j\beta(t-t_o)^2 - j w_\tau t - j\beta t^2]d\tau,
$$

$$
= \exp[j w_\tau t - j\beta(t-t_o)^2] \int \exp[-2j\beta(t-t_o)\tau - 2j\beta \tau^2]d\tau.
$$

(2.22)

Because $\beta$ is large, ($O(10^{14})$), the integral in Eq. (2.22) can be evaluated asymptotically by the stationary phase method [22]. Working with the term under the
integrand and letting \( \phi(\tau) = -2(t - t_o)\tau - 2\tau^2 \), we have a stationary phase problem with \( \beta \) as the large parameter,

\[
I = \int e^{j\beta\phi(\tau)} d\tau. \tag{2.23}
\]

Taking the first and second derivative of the phase term,

\[
\begin{align*}
\phi(\tau) &= -2(t - t_o)\tau - 2\tau^2, \\
\phi'(\tau) &= -2(t - t_o) - 4\tau, \\
\phi''(\tau) &= -4.
\end{align*} \tag{2.24}
\]

The stationary phase point is where \( \phi'(\tau_o) = 0 \). Solving for \( \tau_o, \tau_o = -(t - t_o)/2 \), with the result that \( I \) is,

\[
\begin{align*}
I &= \exp[j\beta\phi(\tau_o)] \sqrt{\frac{2\pi}{\beta|\phi''(\tau_o)|}} \exp \left[ \frac{-j\pi}{4} \right], \\
&= \exp \left[ j\beta\frac{(t - t_o)^2}{2} \right] \sqrt{\frac{\pi}{\beta^2}} \exp \left[ \frac{-j\pi}{4} \right]. \tag{2.25}
\end{align*}
\]

Plugging the asymptotic result into Eq. (2.22),

\[
\int AD = \exp(jw_o t) \exp \left[ j\beta\frac{(t - t_o)^2}{2} \right] \sqrt{\frac{\pi}{\beta^2}} \exp \left[ \frac{-j\pi}{4} \right]. \tag{2.26}
\]

A similar stationary phase point analysis of the \( BC \) term gives,

\[
\int BC = \exp(-jw_o t) \exp \left[ j\beta\frac{(t - t_o)^2}{2} \right] \sqrt{\frac{\pi}{\beta^2}} \exp \left[ \frac{j\pi}{4} \right]. \tag{2.27}
\]

Now the \( AD \) and \( BC \) terms of Eq. (2.13) terms can be added,

\[
\begin{align*}
\int AD + \int BC &= \\
&= \exp(jw_o t) \sqrt{\frac{\pi}{2\beta}} \left( \exp \left[ j\beta\frac{(t - t_o)^2}{2} + j\frac{\pi}{4} \right] + \exp \left[ -j\beta\frac{(t - t_o)^2 - j\frac{\pi}{4}}{2} \right] \right), \\
&= \exp(jw_o t) \sqrt{\frac{\pi}{2\beta}} \left( 2\cos \left[ \beta\frac{(t - t_o)^2}{2} + \frac{\pi}{4} \right] \right), \\
&= \exp(jw_o t) \sqrt{\frac{2\pi}{\beta}} \left( \cos \left[ \frac{\beta(t - t_o)^2}{2} + \frac{\pi}{4} \right] \right). \tag{2.28}
\end{align*}
\]

Because \( \beta \) is large, and it appears in the denominator of Eq. (2.28), the contribution to range compression from the \( AD \) and \( BC \) terms is small and can be ignored. The
contribution to \( m(t) \) from the \( AC \) and \( BD \) terms dominate, and from Eq. (2.13), the matched filter output is,

\[
m(t) \simeq \int AC + BD
\]

\[
= 2T \exp(-j\omega_c t_o + j\omega_0 t) \text{sinc}[\beta T (t - t_o)] \cos[\beta T (t - t_o) - \beta (t - t_o)^2],
\]

(2.29)

Several interesting observations can be made from Eq. (2.29). As expected, the result of the autocorrelation is a \text{sinc} function. The longer \( T \) is, the higher the SNR of the range compressed image. \( T \) is the length of the chirp in time and should be made as long as possible, a well known result of autocorrelation. Limiting an increase in \( T \) is the hardware’s maximum range sample rate (See Eq. (2.8)).

A computer simulation of the chirp compression verifies the mathematical description of range compression. Figure 2.5 compares simulated range compression with the mathematical model of Eq. (2.29). The solid line represents the mathematical solution and the dotted line represents the computer simulation. The mathematical treatment for range compression assumed an infinite length matched filter. This assumption caused the discrepancy between the modeled and the simulated result. Noting that the 3dB point occurs at \( t \approx 1.8 \times 10^{-8} \) s, the range resolution can be calculated, \( \delta r = 2(3 \times 10^8)(1.8 \times 10^{-8}) = 1.08 \) m. Further verification is made by comparison with a common approximation for the the range resolution,

\[
\delta r = \frac{c}{2BW}.
\]

(2.30)

This approximation can be derived from Eq. (2.29) by evaluating the first null. The cosine term in Eq. (2.29) determines the location of the null. Setting the cosine term equal to zero, and letting \( (t - t_o) = x \), the first null is where \( x \) satisfies,

\[
\beta x^2 - T \beta x - \pi/2 = 0.
\]

(2.31)

Equation (2.31) quadratic in \( x \). Using Eq. (2.2),

\[
x = \frac{T \beta \pm \sqrt{T^2 \beta^2 + 2\beta \pi}}{2\beta}
\]

\[
= \frac{T}{2} \left( 1 \pm \sqrt{1 + \frac{2}{TBW}} \right).
\]

(2.32)
Figure 2.5: Simulated range compression vs the mathematical model for range compression. Due to the assumption of an infinite length matched filter the mathematical model does not agree exactly with the simulated range compression. The 3dB point occurs at $t \approx 1.8 \times 10^{-8}$ s. Multiplying by the speed of light gives the range resolution in meters, 1.08 m.

Because the product $T \cdot BW$ is large, the relation $\sqrt{1 + a} \approx (1 + a/2)$ for $a \ll 1$, (a Taylor series approximation), can be used to find $x$.

$$x = \frac{T}{2} \left( 1 \pm \left[ 1 + \frac{1}{TBW} \right] \right),$$

$$= \frac{1}{2BW}.$$  \hspace{1cm} (2.33)

Multiplying Eq. (2.33) by $c$ gives Eq. (2.30) to verify the approximation for range resolution. For YINSAR, from Eq. (2.30), $\delta r = 1$ m.
Figure 2.6 shows the result of range compression on a point target. The target is located in the center of the line. The sinc function in range is apparent in the magnitude plot.

![Range Compressed Point Target](image)

Figure 2.6: Magnitude plot of a range compressed point target. The sinc function in range is apparent in the image.

## 2.4 Azimuth Compression

The range compressed data of Eq. (2.29) now requires a matched filter in the azimuth direction. The return from a given target is sampled numerous times as the radar moves in the azimuth direction. When a target first enters the antenna footprint of Fig. 2.2 it is in front of the radar platform. The radar platform is in motion and the return has a positive Doppler shift added to it. When the radar platform is directly perpendicular to the target there is zero Doppler shift, and as the radar passes the target the return has a negative Doppler shift added to it. The Doppler shift from each target is quadratic in nature, forming an approximately linear frequency modulated chirp in the azimuth direction that can be removed by a matched filter. Azimuth matched filtering is also called azimuth compression. Starting with Eq. (2.29), when
\( t = t_o, \ \text{sinc}(0) = 1 \) and \( \cos(0) = 1 \), so that the range compressed signal is,

\[
m(t_o) = 2T \exp(-jw_c t_o + jw_o t_o).
\] (2.34)

Since \( w_c \gg w_o \), the \( w_o t_o \) term can generally be neglected. The following derivation assumes \(-w_c t_o + w_o t_o \simeq -w_c t_o\). A change of variables, \( t_o \Rightarrow t \) makes the azimuth chirp derivation more analogous to that for the range compression. Now \( t \) is the ‘slow time’, i.e. measured in terms of the azimuth sample rate or PRF (pulse repetition frequency). It is assumed that the PRF exceeds the Nyquist rate and the azimuth compression derivation will be carried out in the continuous domain. The azimuth chirp is a result of the \( w_c t_o \) term in the exponent of \( m(t) \). Keeping Eq. (2.34) in mind, a small tangent is required to understand \( w_c \), and how it puts an LFM chirp into Eq. (2.34).

If \( R(t) \) is the range to the target, let the round trip time to a target be defined as \( t = 2R(t)/c \). For a given target, \( R(t) \) is approximately quadratic in nature. The coefficients of \( R(t) \) are dependent on the location of the target. For convenience, let,

\[
\psi(t) = w_c t,
\]

\[
= \frac{2w_c R(t)}{c},
\]

\[
= \frac{4\pi R(t)}{c},
\]

\[
= \frac{4\pi R(t)}{\lambda}.
\] (2.36)

Using the approximation, \(-w_c t + w_o t \simeq -w_c t\), and \( \psi \) as defined in Eq. (2.36), from Eq. (2.34) the range compressed signal is,

\[
s(t) \simeq c_r a \exp(-jw_c t),
\]

\[
= c_r a \exp(-j\psi(t)),
\]

\[
= c_r a \exp\left(-j\frac{4\pi R(t)}{\lambda}\right),
\] (2.37)

where \( c_r a \) is a constant determined by the length of the range chirp,

\[
c_r a = 2T.
\] (2.38)

A mathematical derivation of the term \( R(t) \) based on the method outlined in [23] follows.
2.4.1 Derivation of the Quadratic Phase Term

From Fig. 2.1, the range to the target is,

\[ R(t) = \sqrt{x(t)^2 + y(t)^2 + H(t)^2}, \quad -T_a/2 < t < T_a/2 \]  \hspace{1cm} (2.39)

where \( t \) is the azimuth sample time. \( x(t) \), \( y(t) \), and \( H(t) \), are the distances to the target in the x, y, and z planes respectively. \( T_a \) is determined by the azimuth antenna beam-width, \( \theta_a \). In the uniform or motion-less case, \( y(t) \) and \( H(t) \) are constants, and \( x(t) = x_o - vt \), where \( v \) is the velocity of the aircraft, so that,

\[ R(t) = \sqrt{(x_o - vt)^2 + y_o^2 + H_o^2}. \] \hspace{1cm} (2.40)

Let \( R_o = \sqrt{x_o^2 + y_o^2 + H_o^2} \) be the nominal minimum distance to the target as in Fig. 2.1. Solving \( R_o \) for \( x_o \) and substituting into \( R(t) \),

\[ R(t) = \sqrt{R_o^2 - 2x_o vt + (vt)^2}. \] \hspace{1cm} (2.41)

Performing a Taylor series expansion about \( t = 0 \), (where \( R_o \) occurs),

\[ R(t) = R_o - \frac{x_o vt}{R_o} + \frac{v^2}{2R_o} \left(1 - \frac{x_o^2}{R_o^2}\right) t^2 + \frac{x v^3}{2R_o^3} \left(1 - \frac{x_o^2}{R_o^2}\right) t^3 + \ldots. \]  \hspace{1cm} (2.42)

Defining \( x_o = 0 \), as in Fig. 2.1, \( R(t) \) simplifies to,

\[ R(t) = R_o + \frac{v^2 t^2}{2R_o} - \frac{v^4 t^4}{8R_o^3} + \ldots. \] \hspace{1cm} (2.43)

The phase term of a point as a function of the azimuth position of the radar platform is of interest and the purpose of determining \( R(t) \). The phase term in the ideal range compressed data as given by Eq. (2.37) is,

\[ \psi(t) = 4\pi \frac{R(t)}{\lambda}, \]

\[ \approx 4\pi R_o + \frac{2\pi v^2 t^2}{\lambda R_o}. \] \hspace{1cm} (2.44)

The approximately quadratic nature of \( \psi(t) \) is show in Fig. 2.7.
Figure 2.7: Normalized phase, $\psi(t)$ of the range compressed data. $\psi(t)$ is the azimuth chirp.

### 2.4.2 Azimuth Matched Filter

After range compression, it can be seen from Eqs. (2.37) and (2.44) that the range compressed signal contains a chirp,

$$s(t) = c_r \exp \left( -j4\pi R_o \right) \exp \left( -j2\pi v^2(t - t_o)^2 \right). \quad (2.45)$$

As promised, the azimuth chirp is a linear frequency modulated chirp. The matched filter for the azimuth chirp is,

$$h(t) = \exp \left( -j2\pi v^2 t^2 \right). \quad (2.46)$$
The azimuth autocorrelation is defined as,

\[ az(t) = s(t) \otimes h^*(t), \]

\[ = \int s(t + \tau) h^*(\tau) d\tau \quad -T_a/2 < \tau + t - t_o < T_a/2, \]

\[ = c_a \exp \left( -\frac{j4\pi R_o}{\lambda} \right) \int \exp \left( -\frac{j2\pi v^2}{R_o \lambda} (t - t_o + \tau)^2 \right) \exp \left( \frac{j2\pi v^2 \tau^2}{R_o \lambda} \right) d\tau, \]

\[ = c_a \exp \left( -\frac{j4\pi R_o}{\lambda} \right) \int \exp \left( -\frac{j2\pi v^2}{R_o \lambda} [(t - t_o)^2 + 2\tau (t - t_o)] \right) d\tau. \quad (2.47) \]

Pulling out the terms independent of \( \tau \) and applying the limits of integration,

\[ az(t) = \]

\[ = c_a \exp \left( -\frac{j4\pi R_o}{\lambda} \right) \int_{-T_a/2-(t-t_o)}^{T_a/2-(t-t_o)} \exp \left( -\frac{j4\pi v^2 t - t_o}{R_o \lambda} \right) d\tau, \]

\[ = \frac{j c_a c_3}{c_4} \exp \left( -j c_4 \tau \right) \bigg|_{\tau = -T_a/2-(t-t_o)}^{\tau = T_a/2-(t-t_o)}, \quad (2.48) \]

where \( c_3 \) and \( c_4 \) are,

\[ c_3 = \exp \left( -\frac{j4\pi R_o}{\lambda} - \frac{j2\pi v^2 (t - t_o)^2}{R_o \lambda} \right), \quad (2.49) \]

\[ c_4 = \frac{4\pi v^2 (t - t_o)}{R_o \lambda}. \quad (2.50) \]

Evaluating Eq. (2.48) at the limits,

\[ az(t) = \frac{j c_a c_3}{c_4} \left[ \exp \left( -j c_4 \left( \frac{T_a}{2} - (t - t_o) \right) \right) \right. \left. - \exp \left( -j c_4 \left( \frac{-T_a}{2} - (t - t_o) \right) \right) \right], \]

\[ = \frac{j c_a c_3}{c_4} \exp(j c_4 (t - t_o)) \left[ \exp \left( -j c_4 \frac{T_a}{2} \right) \right. \left. - \exp \left( j c_4 \frac{T_a}{2} \right) \right], \]

\[ = -\frac{j c_a c_3}{c_4} \exp(j c_4 (t - t_o)) 2j \sin \left( \frac{c_4 T_a}{2} \right), \]

\[ = c_a T_o c_3 \exp(j c_4 (t - t_o)) \text{sinc} \left( \frac{c_4 T_a}{2} \right). \quad (2.51) \]

Plugging in the values for \( c_1 \) and \( c_2 \),

\[ az(t) = \]

\[ = c_a T_o \exp \left( -\frac{j4\pi R_o}{\lambda} - \frac{j2\pi v^2 (t - t_o)^2}{R_o \lambda} + \frac{j4\pi v^2 (t - t_o)^2}{R_o \lambda} \right) \text{sinc} \left( \frac{2\pi T_o v^2 (t - t_o)}{R_o \lambda} \right), \]

\[ = c_a T_o \exp \left( -\frac{j4\pi R_o}{\lambda} \right) \exp \left( \frac{j2\pi v^2 (t - t_o)^2}{R_o \lambda} \right) \text{sinc} \left( \frac{2\pi T_o v^2 (t - t_o)}{R_o \lambda} \right). \quad (2.52) \]
The azimuth compressed signal is now in terms of the the time a given target
is illuminated, \( T_a \), and the distance from the target to the flight path, \( R_o \). It would
be convenient to find a form of Eq. (2.52) such that target dependent variables are
no longer present in the equation. Although that is not entirely possible, the \textit{sinc}
function can be put in terms of the azimuth beam-width, \( \theta_a \). With velocity in the \( x \)
direction and \( L \) as the distance the radar platform travels during the time a target is
illuminated, the azimuth chirp length, \( T_a \), is defined,

\[
T_a = \frac{L}{v}.
\]  

(2.53)

Using the geometry of Fig. 2.8,

\[
\tan \left( \frac{\theta_a}{2} \right) = \frac{L/2}{R_o},
\]

\[
= \frac{T_a v}{2R_o}.
\]  

(2.54)

Solving for \( T_a \) in Eq. (2.54) and substituting into Eq. (2.52),

\[
\begin{align*}
L \\
\hline
\end{align*}
\]

\[
(x_o, y_o)
\]

\[
\theta_a
\]

\[
R_o
\]

\[
y
\]

\[
x
\]

Figure 2.8: Azimuth chirp length defined. \( \theta_a \) is the azimuth antenna beamwidth, \( R_o \)
is the nominal distance to the target, and \( L \) is the total distance the SAR travels
while illuminating a given target.
\[ T_a = \frac{2R_o}{v} \tan \left( \frac{\theta_a}{2} \right), \]  \hfill (2.55)

\[ az(t) = c_{ra} T_a \exp \left( -j4\pi R_o \right) \exp \left( \frac{j2\pi v^2(t - t_o)^2}{R_o \lambda} \right) \times \]
\[ \text{sinc} \left( \frac{4\pi v(t - t_o)}{\lambda} \tan \left( \frac{\theta_a}{2} \right) \right). \hfill (2.56) \]

The \text{sinc} function in \( az(t) \) is now expressed in terms of system parameters, and not target dependent parameters. The first two terms, however, are target dependent. The first term is a constant phase dependent only on \( R_o \). The second term is not constant, but quadratic in time. A range and azimuth compressed image of a point target is shown in Fig. 2.9. The \text{sinc} function in azimuth is apparent.

![Figure 2.9: Magnitude of a range and azimuth compressed point target. A sinc function in azimuth is apparent, a consequence of the azimuth matched filter.](image)

2.4.3 PRF Requirements

The minimum PRF (Pulse Repetition Frequency), or azimuth sample rate, is given by the maximum frequency of the azimuth chirp. The maximum frequency is
determined by the second exponential term in Eq. (2.45) when \( t - t_o = T_a \). Frequency is the rate of phase change, or the derivative of the exponent of,

\[
\exp \left( \frac{j2\pi v^2(t - t_o)^2}{R_o\lambda} \right)
\]  

(2.57)

Converting from radians to Hertz, taking the first derivative, and setting \( t - t_o = T_a \),

\[
f_{\text{max}} = \frac{v^2T_a}{R_o\lambda},
\]  

(2.58)

The minimum PRF (Hz) is double the highest frequency,

\[
PRF_{\text{min}} = \frac{2v^2T_a}{R_o\lambda},
\]

\[
= \frac{4v \tan \left( \frac{\theta_a}{2} \right)}{\lambda},
\]  

(2.59)

\[
\approx \frac{2v}{d},
\]  

(2.60)

where \( d \) is the physical size of the antenna. Small angle approximations and \( \theta_a \approx \lambda/d \) were used to get Eq. (2.60) from Eq. (2.59). Eq. (2.60) is a frequently used approximation for calculating the minimum PRF. If the PRF falls under this Nyquist sampling rate given by Eq. (2.59), azimuth aliasing will occur. Note from Eq. (2.56) that the azimuth resolution improves by increasing \( \theta_a \). As with range compression, the SNR increases with the length of time, in azimuth, the target is illuminated, \( T_a \), (See Eq. (2.56)), further motivating an increase in \( \theta_a \). However, opposing the increase in \( \theta_a \) are the hardware limitations on the maximum PRF given by Eq. (2.59).

### 2.4.4 Azimuth Resolution and Verification of the Mathematical Model

The azimuth resolution can be determined by the first null in the azimuth compressed data. From Eq. (2.56), the first null of the \( \text{sinc} \) function occurs when,

\[
\frac{4\pi v(t - t_o)}{\lambda} \tan \left( \frac{\theta_a}{2} \right) = \pi.
\]  

(2.61)

Letting \( t - t_o = x \) and solving for \( x \),

\[
x = \frac{\lambda}{4v \tan \left( \frac{\theta_a}{2} \right)},
\]

\[
= \frac{\lambda}{2v\theta_a},
\]  

(2.62)
where the small angle approximation of $\tan(\theta) \approx \theta$ was made. The antenna beamwidth can be approximated as, $\theta_a \approx \lambda/d$, where $d$ is the physical size of the antenna. Multiplying the time, $x$, by the radar velocity to obtain the azimuth resolution, $\delta a$,

$$\delta a \approx xv,$$

$$= \frac{\lambda}{2\theta},$$

$$= \frac{d}{2}. \quad (2.63)$$

Equation (2.63) is a common approximation for the azimuth resolution of synthetic aperture radar. For YINSAR, $d = 20 \text{ cm}$, so that Eq. (2.63) gives $\delta a = 10 \text{ cm}$. Plugging the YINSAR parameters in Table 2.1 into the $sinc$ term of Eq. (2.56), $\delta a = 9 \text{ cm}$.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>YINSAR value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_c$</td>
<td>transmit frequency</td>
<td>9.9 GHz</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>wavelength</td>
<td>0.0303 m</td>
</tr>
<tr>
<td>$BW$</td>
<td>chirp bandwidth</td>
<td>100 MHz</td>
</tr>
<tr>
<td>$T$</td>
<td>chirp duration</td>
<td>1.5 $\mu$s</td>
</tr>
<tr>
<td>$T \cdot BW$</td>
<td>time bandwidth product</td>
<td>225</td>
</tr>
<tr>
<td>$s_{A/D}$</td>
<td>AtoD sample frequency</td>
<td>500 MHz</td>
</tr>
<tr>
<td>$PRF_{min}$</td>
<td>Minimum Pulse Repetition Frequency (PRF)</td>
<td>600 Hz</td>
</tr>
<tr>
<td>$\theta_a$</td>
<td>azimuth 3dB beam-width</td>
<td>$10^\circ$</td>
</tr>
<tr>
<td>$\theta_r$</td>
<td>range 3dB beam-width</td>
<td>$45^\circ$</td>
</tr>
<tr>
<td>$d$</td>
<td>real aperture length</td>
<td>0.20 m</td>
</tr>
<tr>
<td>$\delta r$</td>
<td>theoretical range resolution</td>
<td>1.0 m</td>
</tr>
<tr>
<td>$\delta a$</td>
<td>theoretical azimuth resolution</td>
<td>0.10 m</td>
</tr>
<tr>
<td>$V$</td>
<td>velocity</td>
<td>60 m/s</td>
</tr>
<tr>
<td>$H$</td>
<td>nominal radar platform elevation</td>
<td>300 m</td>
</tr>
</tbody>
</table>
Chapter 3

Interferometry Basics

SAR interferometers are used to create elevation maps. Interferometry requires two receive antennas, the phase difference between the two antennas is a function of the height of the scene that is imaged and can be used to construct a digital elevation map. The imaging geometry strongly effects the instrument’s sensitivity to height, and the height measurement accuracy. A carefully designed interferometer can obtain excellent height sensitivity and minimize the effects of various parameter errors on the height measurement. In this chapter a statistical model of the interferometric phase difference and an analysis of the imaging geometry is used to determine the optimal imaging parameters. After determining the optimal parameters an example of interferometer design is given using the BYU interferometer, YINSAR.

3.1 Interferometry Geometry

The relative geometry of an interferometric SAR antenna pair and the scene under observation determine the ability of an interferometer to resolve height. A simple geometry for interferometric imaging is shown in Fig. 3.1. The figure and the following mathematical description of interferometry follow the work of Rodriguez and Martin [4], Zebker and Villasenor [24], and Zebker et al.[5]. The difference in the distance from the antennas to a target, $\delta$, is dependent on the elevation of the target, $z$. $\delta$ is a function of the parameter of interest, the phase difference between the two antennas, $\phi$,

$$\delta = \frac{\lambda \phi}{2\pi}. \quad (3.1)$$
From Fig. 3.1 and the law of cosines and noting that \( \cos\left(\frac{\pi}{2} + \zeta\right) = -\sin(\zeta) \),

\[
\cos\left(\frac{\pi}{2} - \alpha + \theta\right) = \frac{R_o^2 + B^2 - (R_o + \delta)^2}{2BR_o}
\]

(3.2)

\[
\sin(\theta - \alpha) = \frac{(R_o + \delta)^2 - R_o^2 - B^2}{2BR_o}
\]

(3.3)

The height, \( z \), can be determined by some additional geometry,

\[
\cos(\theta) = \frac{H - z}{R_o}
\]

(3.4)

Solving for \( z \) in Eqs. (3.3) and (3.4) using the trigonometric identity \( \cos(\alpha \pm \beta) = \cos(\alpha)\cos(\beta) \mp \sin(\alpha)\sin(\beta) \),

\[
z = H - R_o \cos(\theta),
\]

\[
= H - R_o \cos(\alpha) \cos(\alpha - \theta) - R_o \sin(\alpha) \sin(\alpha - \theta),
\]

\[
= H - R_o \cos(\alpha) \sqrt{1 - \sin^2(\alpha - \theta) - R_o \sin(\alpha) \sin(\alpha - \theta)}.
\]

(3.5)

Differentiation with respect to the phase shows how the height sensitivity due to a phase change varies with respect to the baseline angle, \( \alpha \), the look angle, \( \theta \), the
radar wavelength, $\lambda$, and the baseline length, $B$.

\[
\frac{dz}{d\phi} = -R_o \sin(\alpha) \frac{d}{d\phi} \left( \sin(\alpha - \theta) \right) - R_o \cos(\alpha) \frac{d}{d\phi} \left( \cos(\alpha - \theta) \right),
\]
\[
= -R_o \sin(\alpha) \frac{d}{d\phi} \left( \frac{(R_o + \delta)^2 - R_o^2 - B^2}{2R_oB} \right) - R_o \cos(\alpha) \frac{d}{d\phi} \left( \sin(\alpha - \theta) \tan(\alpha - \theta) \right),
\]
\[
= \frac{\lambda}{2\pi B} (R_o + \delta) [\sin(\alpha) - \cos(\alpha) \tan(\alpha - \theta)] \delta \theta,
\]
\[
\approx \frac{\lambda R_o}{2\pi B} [\sin(\alpha) - \cos(\alpha) \tan(\alpha - \theta)] \delta \phi, \tag{3.6}
\]

since $R_o \gg \delta$. Eq. (3.6) will prove useful both in determining the effect of phase measurement errors on the height measurement and in determining values for $B$, $\lambda$, $\theta$, $H$ and $\alpha$ as illustrated later in section 3.3. For notational convenience it is useful to define a measure for the meters in height per phase wrap, or height sensitivity. Lumping the geometrically determined constants in Eq. (3.6) together for the sake of simplicity as in [15],

\[
\lambda^* = \frac{\lambda R_o}{B} [\sin(\alpha) - \cos(\alpha) \tan(\alpha - \theta)]. \tag{3.7}
\]

With $\phi$ as the unwrapped phase of an interferogram, the height can be related to the phase,

\[
h = \frac{\lambda^*}{2\pi} \phi. \tag{3.8}
\]

The most difficult problem in calculating the height of an image from an interferogram is that the phase information is modulo $2\pi$. Recall that the wrapped phase is,

\[
\psi = \arg\{\exp(j\phi)\},
\]
\[
= \phi - 2\pi c, \tag{3.9}
\]

where $c$ is an integer. An image of the phase data wrapped in $2\pi$ is shows in Fig. 3.2. The upper image is the unwrapped phase, $\phi$, of Isolation Peak Colorado. The lower image is the wrapped version of $\phi$. The wrapped phase, $\psi$, is the only information available to calculate $\phi$, thus $\phi$ can be an integer multiple of $2\pi$ greater of less than $\psi$. This ambiguity makes it difficult to determine the height. Methods for determining
Figure 3.2: The upper image is the unwrapped phase, $\phi$ from Isolation Peak Colorado. The lower image is the wrapped version, $\psi$, of the upper image. The objective of phase unwrapping algorithms is to find $\phi$ from $\psi$. (Data from the FTP site mentioned in [13].)

$\phi$ from $\psi$ are the objective of phase unwrapping algorithms. Some algorithms find an estimate for $\phi$ where $c$ is not an integer. Solutions for $\phi$ that satisfy Eq. (3.9) are termed ‘congruent’ solutions. Methods for phase unwrapping are discussed in Chapter 4 and a proposed method for avoiding phase unwrapping is discussed in Chapter 5.
3.2 Interferogram Statistics

Just and Bamler [25] show that the probability distribution function of the interferometric phase difference, $\phi$, is,

$$
\text{pdf}(\phi) = \frac{1 - |\gamma|^2}{2\pi} \frac{1}{1 - |\gamma|^2 \cos^2(\phi - \phi_o)} \times \left\{ 1 + \frac{|\gamma| \cos(\phi - \phi_o) \cos^{-1}[-|\gamma| \cos(\phi - \phi_o)]}{[1 - |\gamma|^2 \cos^2(\phi - \phi_o)]^{1/2}} \right\},
$$

(3.10)

where $\phi$ is the phase difference of the interferometric pair,

$$
\phi = \arg\{z_1, z_2^*\}.
$$

(3.11)

$z_1$ and $z_2$ are the complex returns at antenna one and two respectively. It is assumed that the nominal flat earth induced phase difference is removed from the data and only height induced phase difference remains in $\phi$. The complex correlation coefficient, $\gamma$, is defined,

$$
\gamma = \frac{E\{z_1z_2^*\}}{\left(E\{|z_1|^2\}E\{|z_2|^2\}\right)^{1/2}},
$$

(3.12)

and $\phi_o$ is defined by,

$$
\phi_o = \arg\{\gamma\}.
$$

(3.13)

The correlation coefficient can be calculated from the data as $|\gamma|$. The pdf of Eq. (3.10) is periodic in $2\pi$. pdf($\phi$) vs $\phi$ for several values of $|\gamma|$ is plotted in Fig. 3.3. The pdf converges to a delta function as $|\gamma|$ approaches unity, and is a uniform distribution for $|\gamma| = 0$. Figure 3.4 shows the experimentally determined probability distribution function. The measurements are the results of over $10^4$ phase measurements taken by YINSAR as the SNR was varied from zero to 20 dB. The correlation coefficient was calculated using 3.12. The mean was subtracted from the data to center is at $\phi = 0$. Figure 3.5 shows the theoretical and empirically measured results of Figs. 3.3 and 3.4 together. The pdfs are in excellent agreement, verifying the accuracy of Eq. (3.10).
Figure 3.3: As the correlation coefficient approaches unity, pdf(θ) approaches a delta function. For a completely uncorrelated signal, γ = 0, the distribution is uniform. (Adapted from [25]).

Following Just and Bamlar's work and choosing a reference interval of 0 so that the pdf conveniently falls between ±π, the standard deviation can be calculated,

\[
\sigma_\phi^2 = E\{(\phi - \phi_o)^2\},
\]

\[
= \int_{\phi_o - \pi}^{\phi_o + \pi} (\phi - \phi_o)^2 pdf(\phi) d\phi,
\]

\[
= \int_{-\pi}^{\pi} \phi^2 pdf(\phi + \phi_o) d\phi,
\]

\[
= \int_{-\pi}^{\pi} \phi^2 \frac{1 - |\gamma|^2}{2\pi} \frac{1}{1 - |\gamma|^2 \cos^2(\phi)} \times \left\{1 + \frac{|\gamma| \cos(\phi) \cos^{-1}|-|\gamma| \cos(\phi)|}{[1 - |\gamma|^2 \cos^2(\phi)]^{1/2}}\right\} d\phi. \quad (3.14)
\]

The variance in the phase is dependent solely on the phase and the correlation coefficient of the data. Figure 3.6 is a plot of the standard deviation of the phase
Figure 3.4: Empirically measured probability distribution function of the phase. Each measurement of $\gamma$ used $10^4$ samples. The mean was subtracted from the data to center it at $\phi = 0$.

$\gamma$, the total decorrelation, has three components: thermal noise, spatial decorrelation, and temporal decorrelation, and can be written as,

$$\gamma = \rho_{\text{spatial}} \cdot \rho_{\text{thermal}} \cdot \rho_{\text{temporal}}. \quad (3.15)$$

Assuming that the SNR for the two returns are identical the decorrelation due to thermal effects is [4], [24],

$$\rho_{\text{thermal}} = \frac{1}{1 + \text{SNR}^{-1}}. \quad (3.16)$$
Figure 3.5: The theoretical and empirically measured results of Figs. 3.3 and 3.4 compared. The pdfs are in excellent agreement.

The phase standard deviation vs signal to noise ratio is plotted in Fig. 3.7 using Eq. (3.16) where $\rho_{\text{spatial}}$ and $\rho_{\text{temporal}}$ are assumed to have unity value. Empirical measured results from YINSAR are compared to the theoretical values and are in agreement. As expected, the phase standard deviation is highly dependent on the signal to noise ratio.

The spatial correlation coefficient is a function of the antenna geometry. When the look angle to a given target is much different between two antennas the target becomes spatially uncorrelated. The spatial decorrelation is found by an analytical integration of the complex covariance of the return at the two antennas. Neglecting surface slope and mis-registration errors the equation for $\rho_{\text{spatial}}$ as found in [4] reduces
Figure 3.6: Phase standard deviation, $\sigma_\phi$, vs. the correlation coefficient, $|\gamma|$. For completely uncorrelated data the standard deviation of the phase is 104°. Empirical measurements made with YINSAR are in agreement with the theoretical values.

to the equations found in [24] and [5],

$$\rho_{\text{spatial}} = 1 - \frac{|B_\perp R_y|}{\lambda R_o \tan(\theta)},$$

where $B_\perp$ is the baseline length perpendicular to the line of sight, $R_y$ is the slant range resolution, $R_o$ is the distance from the antenna to the target, and $\theta$ is the incidence angle.

Temporal decorrelation only applies when the scene itself has changed between radar images. A unity value for $\rho_{\text{temporal}}$ is assumed in this work.

3.3 Choosing Optimal Interferometric Imaging Parameters

From Eq. (3.7) it can be seen that the height sensitivity of an interferometer improves, ($\lambda^*$ decreases), as the baseline length increases. This would indicate that the
Figure 3.7: The phase standard deviation, $\sigma_\phi$, vs SNR. Spatial and temporal correlation are assumed unity. The phase standard deviation falls off sharply with the signal to noise ratio. Empirical measurements made with YINSAR are in agreement with the theoretical values.

The largest baseline possible should be used to achieve the best sensitivity. However, there is an upper limit on the length of the baseline. As the baseline increases the look angle to the target from the two antennas becomes increasingly different causing spatial decorrelation in the received signals, thus there exists an optimal baseline length for a given imaging geometry. Inaccuracy in the knowledge of system parameters, such as the baseline length, baseline angle, range to the target, height, etc. lead to inaccuracy in the height measurement. By analyzing the effect of errors in each parameter on the height measurement, the effect of the errors can be minimized. Guidelines for choosing interferometric imaging parameters and an example for a specific system are outlined in this section. A rigorous approach to choosing optimal interferometric parameters that includes location induced error, (correct height given for the wrong location),
can be found in [4]. The effects of phase aberrations, geometric misregistration, defocusing, and range migration can be found in [25].

3.3.1 Height Sensitivity with Respect to System Parameters

To determine the sensitivity of the height measurement to perturbations in the various system parameters, $B$, $R_o$, $\theta$, $\alpha$, $H$, and $\phi$, the derivative of $z$ with respect to each of the parameters can be taken. Recall from Eqs. (3.1), (3.3), and (3.5),

$$z = H - R_o \cos(\alpha) \sqrt{1 - \sin^2(\alpha - \theta)} - R_o \sin(\alpha) \sin(\alpha - \theta),$$

where,

$$\sin(\alpha - \theta) = \frac{(R_o + \delta)^2 - R_o^2 - B^2}{2BR_o},$$

and,

$$\delta = \frac{\lambda \phi}{2\pi}.$$

The sensitivity with respect to the baseline length, $B$, is,

$$\frac{d}{dB} \sin(\alpha - \theta) = \left( \frac{-4B^2R_o - 2R_o[(R_o + \delta)^2 - R_o^2 - B^2]}{(2BR_o)^2} \right) \delta B,$$

$$= - \left( \frac{1}{R_o} + \frac{\sin(\alpha - \theta)}{B} \right) \delta B,$$

$$\approx - \frac{\sin(\alpha - \theta)}{B} \delta B, \quad (3.18)$$

$$\frac{d}{dB} z = \frac{R_o \cos(\alpha) \sin(\alpha - \theta)}{\cos(\alpha - \theta)} \frac{d}{d\alpha} \sin(\alpha - \theta) - R_o \sin(\alpha - \theta) \frac{d}{dB} \sin(\alpha - \theta),$$

$$= \frac{R_o \tan(\alpha - \theta)}{B} [\cos(\alpha) \sin(\alpha - \theta) - \sin(\alpha) \sin(\alpha - \theta)] \delta B,$$

$$= \frac{R_o}{B} \tan(\alpha - \theta) \sin(\theta) \delta B. \quad (3.19)$$

The sensitivity with respect to $R_o$, $\theta$, $\alpha$, and $H$ are,

$$\frac{dz}{dR_o} = - \cos(\theta) \delta R_o, \quad (3.20)$$

$$\frac{dz}{d\theta} = - R_o \sin(\theta) \delta \theta, \quad (3.21)$$

$$\frac{dz}{d\alpha} = - R_o \sin(\theta) \delta \alpha, \quad (3.22)$$

$$\frac{dz}{dH} = \delta H. \quad (3.23)$$
Recall from Eq. (3.6), the sensitivity with respect the phase, $\phi$ is,

$$\frac{d\theta}{d\phi} = \frac{\lambda R_o}{2\pi B} [\sin(\alpha) - \cos(\alpha) \tan(\alpha - \theta)] \delta\phi. \quad (3.24)$$

The range to the target, $R_o$, is determined by the transmit and receive times of the system, and it’s accuracy is typically excellent. Mis-calculations in $\alpha$ and $\phi$ are particularly troublesome as they are proportional to the large value $R_o$. As mentioned in [5], for small aircraft error in the roll angle measurement can not be distinguished from a change in the height of the surface as shown by Eqs. (3.21), (3.22). Precise knowledge of the aircraft attitude is required and motion compensation must be accurately performed to attain reliable measurements of surface topography.

### 3.3.2 Choosing the Optimal Baseline Length and Baseline Angle

The critical baseline length, $B_{c\perp}$, is the baseline at which the return from a target is completely uncorrelated. Setting $\rho_{\text{spatial}} = 0$ and solving Eq. (3.17) for $B_{\perp}$,

$$B_{c\perp} = \frac{\lambda R_o \tan \theta}{R_y}. \quad (3.25)$$

From Fig. 3.1 the baseline perpendicular to the look direction can be related to the baseline length by,

$$B \sin(\alpha + \theta) = B_{\perp}. \quad (3.26)$$

The critical baseline length for a given geometry is,

$$B_c = \frac{\lambda R_o \tan \theta}{R_y \sin(\alpha + \theta)}. \quad (3.27)$$

In practice $B$ is chosen in the range of 0.5 to 0.8 of $B_c$ so that the signals remain correlated. From Eqs. (3.6) and (3.7) the standard deviation of the estimate of the height is,

$$\sigma_h = \frac{\lambda R_o}{2\pi B} [\sin(\alpha) - \cos(\alpha) \tan(\alpha - \theta)] \sigma_{\phi}$$

$$= \frac{\lambda^*}{2\pi} \sigma_{\phi}. \quad (3.28)$$
The length of the baseline must be accurately known since inaccuracy in the measurement of the baseline length affects the height measurement according to Eq. (3.19). For the general case of $\alpha \neq \theta$, the error in the height estimate induced by a mis-calculation of the baseline length is proportional to $R_\phi$. Setting $\alpha = \theta$ eliminates this error source so that Eq. (3.19) is zero. A second advantage to choosing $\alpha = \theta$ is the simplification of Eq. (3.6), which reduces to,

$$\frac{dz}{d\phi} = \frac{\lambda R_\phi}{2\pi B} \sin(\theta) \delta \theta. \quad (3.29)$$

Choosing $\alpha = \theta$ also minimizes the required baseline length for a given sensitivity and $\theta$ as shown in Fig. 3.8. The plot was made using the system parameters of YINSAR. For other systems, the shape of the curve will be the same, but the vertical scale, (height sensitivity), will be considerably different. From Eq. (3.7), when $\alpha = \theta$,

$$\lambda^* = \frac{\lambda R_\phi \sin(\theta)}{B}. \quad (3.30)$$

### 3.3.3 Other Considerations

The system bandwidth should be chosen as large as possible to increase the geometric correlation [4]. The upper limit on the bandwidth is limited by the achievable sampling rate, data rate, and the RF hardware. In some cases the bandwidth may be limited by the bandwidth of the transmit and receive antennas. The azimuth chirp should also be as large as possible. The net effect of a large azimuth beam width is to increase the time a given point is in the antenna main beam, which increases the correlation. If the azimuth antenna beam width is extremely large, the look angle from the time a target enters the azimuth antenna pattern until it exits the antenna pattern will change significantly enough to cause spatial decorrelation. In general, the sampling rate requirements for a large azimuth beam width are of a greater concern than the possibility of decorrelation.

For the best height sensitivity, $\lambda$ should be chosen as small as possible. As indicated by Eq. (3.7), the height sensitivity is directly proportional to the transmit frequency. Hardware, complexity, and budget limitations are typically the upper limits on the choice of $\lambda$. 

39
Figure 3.8: Height sensitivity, $\lambda^*$, vs baseline angle, $\alpha$. For a given incidence angle, (45° in this figure), the best height sensitivity is attained for a baseline tilt angle equal to the incidence angle, $\alpha = \theta$.

As a general note, the smaller $\lambda^*$ is, the more phase wraps will occur in a scene. The more phase wraps in an image the more difficult it is to properly unwrap the phase. Additionally, if there are any isolated step discontinuities in the image greater than $\lambda^*$ it is impossible to determine the height of any ‘island’ of data isolated by discontinuity. Islands are particularly prevalent in urban environments where buildings form jumps in phase that exceed $2\pi$, yielding it impossible to unambiguously determine the value of $c$ in Eq. (3.9) without additional information. Thus, the height sensitivity as determined by Eq. (3.7) should be determined with the prevalent scenes to be imaged in mind. If the scenes will contain large jumps it may be desirable to design for a larger value of $\lambda^*$. For example, if the interferometer will be used to create elevation maps of urban environments, $\lambda^*$ should be chosen to be at least as
large as the height of the tallest building. Because the phase data is wrapped modulo 2\( \pi \), jump discontinuities greater than \( \lambda^* \) will result in an ambiguous height.

### 3.4 Interferometer Design Example: YINSAR

As an example of the design of an interferometric radar system the BYU YINSAR will be used. A transmit frequency of 9.9 GHz was chosen in order to obtain, 1) high resolution and 2) the small antennas size required for installation on a small aircraft, such as a Cessna four passenger plane. The bandwidth of the transmitted LFM chirp is limited by the the 500 MHz sampling rate of commercially available PC based analog to digital cards. The hardware and software limitations on data rate and the stall speed of the aircraft constrain the azimuth beamwidth and therefore the bandwidth of the azimuth chirp (See Eq. (2.59)). YINSAR operates at a low elevation of \( H = 300 \text{ m} \) and utilizes a wide range of incidence angles. The useful range of incidence angles is \( 20^\circ < \theta < 75^\circ \). The remaining system parameters, baseline tilt and baseline length, must be chosen with this in mind.

The height sensitivity, \( \lambda^* \), and the standard deviation in the height, \( \sigma_h \), vs baseline length for an incidence angle of \( \theta = 45^\circ \) are plotted in Fig. 3.9. Obviously, for the best height resolution the baseline should be chosen as long as possible. Accounting for the highly incident angle dependent value of \( R_o \) for YINSAR, Fig. 3.10 shows the critical perpendicular baseline length over the useful range of incidence angles. The maximum antenna baseline that can be mounted on the aircraft, \( B = 1 \sim 2 \text{ m} \), is well under the critical baseline length for the entire range of incidence angles, (See Fig. 3.10).

With the physical constraints on the baseline length, the last parameter to be evaluated is the baseline tilt, \( \alpha \). Figures 3.11 - 3.12 illustrate the height sensitivity and the standard deviation of the height for several possible baseline tilt angles. The easiest mounting scheme is horizontal to give the sensitivity vs. incidence angle plot for a constant \( B_\perp \) as in Fig. 3.11. The sensitivity is considerably improved when the baseline tilt is nearly equal to the incidence angle as show in Fig. 3.11. Again, the incident angle dependence of \( R_o \) has been accounted for. Figure 3.12 shows the height
Figure 3.9: Sensitivity, $\lambda^*$ and height standard deviation, $\sigma_h$, vs. perpendicular baseline length, $B_\perp$ for an incidence angle of $\theta = 45^\circ$. The best (lowest) height sensitivity and standard deviation are achieved for longer baseline lengths.

standard deviation vs incidence angle for several values of baseline tilt. A baseline tilt of approximately $45^\circ$ would result in the lowest value for the standard deviation across the range of incidence angles.

Considering the aircraft frame, the two most practical mounting schemes are:
1) $\alpha = 0^\circ$ and $B = 1$ m,
2) $\alpha = 45^\circ$ and $B = 1$ m.

Fig. 3.13 illustrates these two options for YINSAR. The solid lines are for the $\alpha = 0^\circ$ case, and the dotted lines represent the $\alpha = 45^\circ$ case. The optimal mounting scheme for YINSAR is to mount the antennas as far apart as possible, with a baseline tilt angle equal to the mean incident angle of $45^\circ$. 

42
Figure 3.10: Critical perpendicular baseline length, $B_{c \perp}$, vs incidence angle, $\theta$. As the incidence angle increases the distance to the target also increases allowing a greater baseline length before total decorrelation occurs. This effect is particularly evident in YINSAR where the the distance to the target is highly dependent on the wide range of incidence angles possible.
Figure 3.11: Height sensitivity, $\lambda^*$ vs incidence angle, $\theta$ for baseline angles of $0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ$ with a perpendicular baseline length of $B_\perp = 1$ m. The height sensitivity is minimum across the range of incidence angles when the baseline angle is equal to the mean incidence angle. $\lambda^*$ for the $\pi/4$ and $3\pi/4$ baseline angles are identical as are the $\pi$ and 0 angles.
Figure 3.12: Height Standard Deviation, $\sigma_h$, vs incidence angle, $\theta$ for baseline angles of $0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ$, and $90^\circ$ with a perpendicular baseline length of $B_\perp = 1$ m. The standard deviation in the height is minimum across the range of incidence angles when the baseline angle is equal to the mean incidence angle. $\sigma_h$ for the $\pi/4$ and $3\pi/4$ baseline angles are nearly identical as are the $\pi$ and $0$ angles.
Figure 3.13: Height sensitivity, $\lambda^*$, and height standard deviation, $\sigma_h$, vs incidence angle, $\theta$ for YINSAR. The two most plausible baseline angles for YINSAR are $\alpha = 0^\circ, 45^\circ$. The solid lines are for the $\alpha = 0^\circ$ case, and the dotted lines represent the $\alpha = 45^\circ$ case. A baseline angle of $45^\circ$ is the optimal choice for maximum height sensitivity.
Chapter 4

Phase Unwrapping Techniques

Two dimensional phase unwrapping is a problem common to a number of fields. In particular, medical imaging and Interferometric Synthetic Aperture Radar, (IFSAR) are driving forces for improved 2-D phase unwrapping. Interferometers use the phase difference between two spatially offset antenna’s to calculate the height profile of the terrain as explained in Chapter 3. The central problem to this approach is that the phase difference is wrapped in $2\pi$. To correctly find the ground truth height profile the phase must be unwrapped in two dimensions. There are essentially two approaches to the problem, minimum norm methods, and path following methods, [13].

The first path following method was developed by Goldstein et al., [9]. Goldstein identifies positive and negative residues (where the phase in a closed loop does not return to zero) and attempts to connect the residues via branch cuts in an \textit{ad hoc} manner. This is followed by two dimensional integration where the integration path is not allowed to cross any branch cuts. The Goldstein method is fast but suffers from integration path dependent solutions and fails under low SNR conditions.

The minimum norm phase unwrapping methods find their source in the work of Ghiglia and Romero [10], who proposed a least squares solution to 2-D phase unwrapping. Their method, as the name suggests, is a least squares solution. The least squares approach is not path dependent and handles low SNR by effectively ‘smoothing over’ the low SNR area. A side effect of least squares solution is that it introduces some distortion in the image, which is most severe in low SNR areas and near branch cuts.
There are numerous variations on these methods and a few other methods that have not yet gained common usage. Comparisons of the most common methods [14] show that the $L_p$-norm and the Flynn [12] algorithm are the most successful phase unwrapping algorithms. The $L_p$-norm method is an iterative approach of which the least squares method is a special case. As illustrative examples of the minimum norm and path following methods, the weighted least squares and the weighted minimum discontinuity methods respectively are presented in this chapter.

The weighted least squares method example is based on Ghiglia and Romero’s work as described in the previous paragraph. A detailed description of the extension of Hunt’s matrix formulation to the weighted case is included for completeness, and efficient methods for implementing the algorithm using vector and matrix multiplies are presented. The efficient methods outlined are particularly applicable with vector and matrix based software such as MATLAB.

The minimum discontinuity algorithm proposed by Flynn [12] is the basis for the second example. Following a description of Flynn’s algorithm, a method for increasing the efficiency of Flynn’s algorithm is presented. The final section in the chapter draws a connection between the seemingly opposite 2-D phase unwrapping methods of least squares and minimum discontinuity.

4.1 Mathematical Derivation of the Least Squares Solution

This section describes the Least Squares (LS) and Weighted Least Squares (WLS) solution to phase unwrapping as originally presented by Ghiglia and Romero [10]. Ghiglia’s notation will pervade most of the mathematical model.

Assume that the unwrapped phase, $\phi$ is known. Let the wrapped phase, $\psi$, on a discrete $M \times N$ grid be defined,

$$
\psi_{m,n} = \phi_{m,n} + 2\pi c, \quad c = 0, 1, 2, 3, \ldots \quad (4.1)
$$

$$
\psi_{m,n} = \phi_{m,n}, \quad mod \ 2\pi
$$

$$
\psi_{m,n} = (\phi_{m,n})_{2\pi},
$$

$$
\psi_{m,n} = \phi_{m,n} - 2\pi \left[ \frac{\phi_{m,n}}{2\pi} \right]. \quad (4.2)
$$
In interferometry, the wrapped phase, \( \psi_{m,n} \), is found directly from the data and the unwrapped phase, \( \phi_{m,n} \), is the desired result. \( \psi \) and \( \phi \) differ by an integer number of \( 2\pi \), (\( c \) in Eq. (4.2)). The wrapped phase difference is defined in the \( x \) and \( y \) dimensions,

\[
\Delta_{m,n}^x = (\psi_{m+1,n} - \psi_{m,n})_{2\pi},
\]

\[
m = 1 \ldots M - 1, \quad n = 1 \ldots N,
\]

\[
= 0, \quad \text{otherwise}
\]

\[
\Delta_{m,n}^y = (\psi_{m,n+1} - \psi_{m,n})_{2\pi},
\]

\[
m = 1 \ldots M, \quad n = 1 \ldots N - 1,
\]

\[
= 0. \quad \text{otherwise}
\]

### 4.1.1 Least Squares Development

The least squares solution is the \( \phi \) that minimizes the least squares difference of \( \phi' \) and \( \psi' \) in the \( x \) and \( y \) dimensions simultaneously. In the \( x \) dimension,

\[
\sum_{m=1}^{M-1} \sum_{n=1}^{N} (\phi_{m+1,n} - \phi_{m,n} - \Delta_{m,n}^x)^2;
\]

and in the \( y \) dimension,

\[
\sum_{m=1}^{M} \sum_{n=1}^{N-1} (\phi_{m,n+1} - \phi_{m,n} - \Delta_{m,n}^y)^2.
\]

The key to minimization is, of course, a derivative. In this case the derivative is discrete and can be put in terms of a backward difference equation which is defined [26],

\[
\frac{df}{dx_i} \approx f(x_i) - f(x_{i-1}).
\]

Applying the backward difference equation to Eq. (4.5) and setting the result equal to zero,

\[
2(\phi_{m+1,n} - \phi_{m,n} - \phi_{m-1,n} - \Delta_{m,n}^x + \Delta_{m-1,n}^x)(\phi_{m+1,n} - \phi_{m,n} - \Delta_{m,n}^x) = 0,
\]
or,

\[ \phi_{m+1,n} - 2\phi_{m,n} + \phi_{m-1,n} = \Delta_{m,n}^x - \Delta_{m-1,n}^x. \]  

(4.8)

The same operation in \( y \) gives,

\[ 2(\phi_{m,n+1} - \phi_{m,n} - \phi_{m,n-1} - \Delta_{m,n}^x + \Delta_{m,n-m}^x)(\phi_{m,n+1} - \phi_{m,n} - \Delta_{m,n}^x) = 0, \]

or,

\[ \phi_{m,n+1} - 2\phi_{m,n} + \phi_{m,n-1} = \Delta_{m,n}^x - \Delta_{m,n-1}^x. \]  

(4.9)

Together, Eqs. (4.8) and (4.9) minimize the least squares solution. Adding the two equations together,

\[
(\phi_{m+1,n} - 2\phi_{m,n} + \phi_{m-1,n}) + (\phi_{m,n+1} - 2\phi_{m,n} + \phi_{m,n-1}) = \rho_{m,n},
\]

(4.10)

where,

\[ \rho_{m,n} = \Delta_{m,n}^x - \Delta_{m-1,n}^x + \Delta_{m,n}^y - \Delta_{m,n-1}^y. \]  

(4.11)

Equation (4.10) is the central difference approximation to the Laplace equation [26],

\[
\nabla^2 \phi = \rho, \tag{4.12}
\]

\[
\frac{\partial \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = \rho,
\]

\[ (\phi_{m+1,n} - 2\phi_{m,n} + \phi_{m-1,n}) + (\phi_{m,n+1} - 2\phi_{m,n} + \phi_{m,n-1}) = \rho_{m,n}. \]

The idea of a Poisson equation leads to some intuition on the LS solution to 2-D phase unwrapping as will be mentioned later on. To solve the Poisson equation the boundary conditions must be known. Neumann boundary conditions can be observed by noting the derivative of \( \psi \) is zero on the boundaries, a constraint imposed by Eqs. (4.4) and (4.5),

\[
\frac{\partial \psi_{m,n}}{\partial x} = 0, \quad m = 1, M, \tag{4.13}
\]

\[
\frac{\partial \psi_{m,n}}{\partial y} = 0, \quad n = 1, N. \tag{4.14}
\]
It is reasonable to ask how matching the second derivatives of $\phi$ and $\psi$ can result in a solution for $\phi$. The discontinuities in $\psi$ imply that the second derivative would result in doublets placed at every discontinuity. This indeed would be the case if the discontinuities were not removed from the first derivative of $\psi$. The removal of the discontinuity occurs in Eqs. (4.4) and (4.5). The first derivative of $\phi$ and $\psi$ are,

$$
\phi' = \phi_m - \phi_{m-1},
$$

(4.15)

$$
\psi' = (\psi_m - \psi_{m-1})_{2\pi}.
$$

(4.16)

The issue at hand is if $\phi' \equiv \psi'$. A look at the modulo operator answers the question,

$$
\psi' = (\psi_m - \psi_{m-1})_{2\pi},
$$

(4.17)

$$
= \psi_m - \psi_{m-1} - 2\pi \left[ \frac{\psi_m - \psi_{m-1}}{2\pi} \right],
$$

(4.18)

$$
= (\phi_m)_{2\pi} - (\phi_{m-1})_{2\pi} - 2\pi \left[ \frac{(\phi_m)_{2\pi} - (\phi_{m-1})_{2\pi}}{2\pi} \right].
$$

Let $\left\lfloor \frac{\phi_m}{2\pi} \right\rfloor = l$ and $\left\lfloor \frac{\phi_{m-1}}{2\pi} \right\rfloor = k$ where $l$ and $k$ are integers,

$$
\psi' = \phi_m - \phi_{m-1} - 2\pi l + 2\pi k - 2\pi \left[ \frac{\phi_m}{2\pi} - l - \frac{\phi_{m-1}}{2\pi} + k \right].
$$

(4.19)

The integers $k$ and $l$ float through the floor operator,

$$
\psi' = \phi_m - \phi_{m-1} - 2\pi \left[ \frac{\phi_m - \phi_{m-1}}{2\pi} \right].
$$

(4.20)

From Eq. (4.20) it can be seen that $\psi' = \phi'$ if and only if $|\phi_m - \phi_{m-1}| < 2\pi$. In other words, if there is less than $2\pi$ of phase difference between samples the first derivative of the unwrapped phase equals the first derivative of the unwrapped phase. When $|\phi_m - \phi_{m-1}| \geq 2\pi$ the first derivatives differ by an integer value, c. Phase unwrapping techniques assume $c = \pm 1$. There is no known method, other than guessing, to successfully unwrap the ambiguous case of a step discontinuity greater than $4\pi$.

### 4.1.2 2-D Discrete Cosine Transform Solution

Ghiglia and Romero showed that Eq. (4.10) can be solved by the fast Discrete Cosine Transform (DCT). Pritt [27] generalized the solution method to FFTs. The
two solutions are similar. The DCT solution follows.

The forward 2-D DCT is,

\[
C_{i,j} = \begin{cases} 
\sum_{m=1}^{M} \sum_{n=1}^{N} 4x_{m,n} \cos \left[ \frac{\pi}{2M} i(2m + 1) \right] \cos \left[ \frac{\pi}{2N} j(2n + 1) \right], \\
0, \quad \text{otherwise.}
\end{cases}
\]

(4.21)

Notice that the DCT imposes Neumann boundary conditions,

\[
C_{0,n} = C_{M+1,n} = 0, \\
C_{m,0} = C_{m,N+1} = 0.
\]

The inverse 2-D DCT, (IDCT), is,

\[
x_{m,n} = \begin{cases} 
\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} w_1(i)w_2(j)C_{i,j} \cos \left[ \frac{\pi}{2M} i(2m + 1) \right] \cos \left[ \frac{\pi}{2N} j(2n + 1) \right], \\
0, \quad \text{otherwise,}
\end{cases}
\]

(4.22)

\[
w_1(i) = \begin{cases} 
1/2 & i = 1, \\
1 & 2 \leq i \leq M,
\end{cases} \\
w_2(j) = \begin{cases} 
1/2 & j = 1, \\
1 & 2 \leq j \leq N.
\end{cases}
\]

The solution to \( \phi \) can be found be letting \( \phi_{m,n} = x_{m,n} \) in Eq. (4.21) so that the frequency domain representation for \( \phi \) is \( \hat{\phi} = C_{i,j} \). Eq. (4.10) can now be written in terms of the IDCT. Some tedious algebra yields the desired result, \( \hat{\phi} \) in terms of \( \hat{\rho} \), where \( \hat{\rho} \) is the DCT representation of \( \rho \). Starting with Eq. (4.10),

\[
\rho_{m,n} = (\phi_{m+1,n} - 2\phi_{m,n} + \phi_{m-1,n}) + (\phi_{m,n+1} - 2\phi_{m,n} + \phi_{m,n-1}).
\]

(4.23)
In terms of the IDCT, (Eq. (4.22)), both sides have a identical summations and several constant terms that can be removed so that,

\[
\hat{r}_{i,j} \cos \left[ \frac{\pi}{2M} i(2m + 1) \right] \cos \left[ \frac{\pi}{2N} j(2n + 1) \right] = \\
\hat{r}_{i,j} \cos \left[ \frac{\pi}{2M} i(2m + 3) \right] \cos \left[ \frac{\pi}{2N} j(2n + 1) \right] \\
- 2\hat{r}_{i,j} \cos \left[ \frac{\pi}{2M} i(2m + 1) \right] \cos \left[ \frac{\pi}{2N} j(2n + 1) \right] \\
+ \hat{r}_{i,j} \cos \left[ \frac{\pi}{2M} i(2m - 1) \right] \cos \left[ \frac{\pi}{2N} j(2n + 1) \right] \\
+ \hat{r}_{i,j} \cos \left[ \frac{\pi}{2M} i(2m + 1) \right] \cos \left[ \frac{\pi}{2N} j(2n + 3) \right] \\
- 2\hat{r}_{i,j} \cos \left[ \frac{\pi}{2M} i(2m + 1) \right] \cos \left[ \frac{\pi}{2N} j(2n + 1) \right] \\
+ \hat{r}_{i,j} \cos \left[ \frac{\pi}{2M} i(2m + 1) \right] \cos \left[ \frac{\pi}{2N} j(2n - 1) \right].
\]

Dividing both sides of the equation by \( \cos \left[ \frac{\pi}{2M} i(2m + 1) \right] \cos \left[ \frac{\pi}{2N} j(2n + 1) \right] \),

\[
\hat{r}_{i,j} = \\
- 4\hat{r}_{i,j} \\
+ \hat{r}_{i,j} \frac{\cos \left[ \frac{\pi}{2M} i(2m + 3) \right] + \cos \left[ \frac{\pi}{2M} i(2m - 1) \right]}{\cos \left[ \frac{\pi}{2M} i(2m + 1) \right]} \\
+ \hat{r}_{i,j} \frac{\cos \left[ \frac{\pi}{2N} j(2n + 3) \right] + \cos \left[ \frac{\pi}{2N} j(2n - 1) \right]}{\cos \left[ \frac{\pi}{2N} m(2j + 1) \right]}
\]

Applying the trigonometric identity \( \cos(\alpha + \beta) = \cos \alpha \cos \beta - \sin \alpha \sin \beta \) to the numerator of the second and third terms,

\[
\hat{r}_{i,j} = \\
- 4\hat{r}_{i,j} \\
+ \hat{r}_{i,j} \frac{\cos \left[ \frac{\pi}{2M} i(2m + 1) \right] \cos \left[ 2i \frac{\pi}{2M} \right] + \sin \left[ \frac{\pi}{2M} i(2m + 1) \right] \sin \left[ 2i \frac{\pi}{2M} \right]}{\cos \left[ \frac{\pi}{2M} i(2m + 1) \right]} \\
+ \hat{r}_{i,j} \frac{\cos \left[ \frac{\pi}{2N} j(2n + 1) \right] \cos \left[ 2j \frac{\pi}{2N} \right] + \sin \left[ \frac{\pi}{2N} j(2n + 1) \right] \sin \left[ 2i \frac{\pi}{2N} \right]}{\cos \left[ \frac{\pi}{2N} j(2n + 1) \right]}
\]

53
Since sine is an odd function, \( \sin(-x) = -\sin(x) \), all terms containing a sine cancel leaving,

\[
\hat{\rho}_{i,j} = -4\hat{\phi}_{i,j} + 2\hat{\phi}_{i,j} \cos \left[ i \frac{\pi}{M} \right] + 2\hat{\phi}_{i,j} \cos \left[ j \frac{\pi}{N} \right].
\]

(4.24)

Solving for \( \hat{\phi} \) in terms of \( \hat{\rho} \),

\[
\hat{\phi}_{i,j} = \frac{\hat{\rho}_{i,j}}{2 \left( \cos \left[ i \frac{\pi}{M} \right] + \cos \left[ j \frac{\pi}{N} \right] - 2 \right)}.
\]

(4.25)

### 4.1.3 Unweighted Least Squares Algorithm (ULS)

From Eq. (4.25) the unwrapped phase, \( \phi \), can be found from the wrapped phase differences, \( \Delta_{x,m,n} \) and \( \Delta_{y,m,n} \), by using \( \rho \). The algorithm as outlined in [10]:

**Unweighted Least Squares Algorithm**

1. From the data, compute the wrapped phase differences, \( \Delta_{x,m,n} \), \( \Delta_{y,m,n} \) (Eqs. (4.4), (4.5))
2. Construct \( \rho_{m,n} \) (Eq. (4.11))
3. Perform 2-D DCT on \( \rho_{m,n} \) to get \( \hat{\rho}_{i,j} \) (Eq. (4.21))
4. Construct \( \hat{\phi}_{i,j} \) (Eq. (4.25))
5. Perform 2-D IDCT on \( \hat{\phi}_{i,j} \) to get \( \phi_{i,j} \), the unwrapped phase. (Eq. (4.22))

### 4.1.4 Examples of the Unweighted Least Squares Algorithm

Fig. 4.1 is a simple simulation of the phase difference between a pair of interferometric antennas. Black corresponds to lowest point and white corresponds to the highest point in the image. The figure on the left is the figure on the right modulo \( 2\pi \). It is identical to the the input to the ULS algorithm, \( \psi \). There are no branch cuts, (i.e. jumps of more than \( 2\pi \)), from sample to sample, and zero noise. Fig. 4.1 represents the simplest possible phase unwrapping scenario, and is trivial to unwrap using any phase unwrapping algorithm. The ULS algorithm perfectly reconstructs the unwrapped phase from \( \psi \) to within a constant bias. The constant bias results
because only the relative phase is input to the algorithm leaving the absolute phase unknown. If the absolute phase at any point in the image is known, the absolute phase everywhere in the image can be found.

In Figs. 4.2 and 4.3 Gaussian noise is added across the entire image. The original phase of Fig. 4.2 was perfectly reconstructed as shown in Fig. 4.3

A more interesting example is Figs. 4.4 - 4.5 where Gaussian noise was added to a square patch of the image. The output, Fig. 4.5, shows that the ULS algorithm is sensitive to noise. The errors in the phase unwrapping due to a patch of noise propagated beyond the boundary where the patch of noise was added. Recall that the Poisson equation is the result of imposing the Least Squares minimization. From electromagnetics we know that the solution to the Poisson equation will apply smooth field lines throughout the region of interest. Because the solution to the Laplacian forced a smooth transition across the noisy patch, the error in the phase due to the patch of noise propagated beyond the border of the patch, with diminishing effects further from the patch. Similar non-localized error is seen by replacing the phase in a patch of the image with uniform noise, (completely uncorrelated with the phase of the image), as shown in Figs. 4.6 - 4.7
Figure 4.2: ULS algorithm input: Additive Gaussian Noise

As a final example of the ULS algorithm, in Figs. 4.8 - 4.9 a branch cut was placed in the middle of the image. The output of the ULS algorithm, Fig. 4.9 is far different from the desired $\phi$ shown in Fig. 4.8. The Poisson solution forced the phase to be continuous across the boundary. A ‘seaming’ effect can be seen along the length of the branch cut. The inability of the ULS to deal with branch cuts is unacceptable for most applications because real data contains branch cuts.
Figure 4.3: ULS algorithm output: Additive Gaussian Noise

Figure 4.4: ULS algorithm input: Patch of additive Gaussian Noise
Figure 4.5: ULS algorithm output: Patch of additive Gaussian Noise

Figure 4.6: ULS algorithm input: Patch Replaced by Uniform Noise
Figure 4.7: ULS algorithm output: Patch Replaced by Uniform Noise

Figure 4.8: ULS algorithm input: Branch Cut
Figure 4.9: ULS algorithm output: Branch Cut
4.2 Model for the Weighted Least Squares Solution

As the first step in making the Least Squares Algorithm robust, it is necessary to assign weights to the data. Areas of high SNR are much more reliable than those with low SNR and should be weighted accordingly. The proper means of weighting $\psi$ is most easily understood in the light and tools of linear algebra. Hunt’s work, with an extension to the weighted case, is presented here.

4.2.1 Hunt’s Work

The work of Hunt, [21] puts the ULS, (Unweighted Least Squares Algorithm), in the framework of matrix algebra. Representing Eq. (4.10) in matrix form,

$$A \phi = \psi',$$  \hspace{1cm} (4.26)

where $A$ is a $2N(N - 1) \times N^2$ matrix, $\phi$ is a $N^2 \times 1$ column vector, and $\psi'$ is a $2N(N - 1) \times 1$ column vector. Some definitions are in order, Let,

$$\phi = \begin{bmatrix} \phi_1: \\
\phi_2: \\
\vdots \\
\phi_N: \end{bmatrix},$$ \hspace{1cm} (4.27)

where $\phi_2:$ is a $N \times 1$ column vector. $\phi$ is the matrix of unwrapped phase values stretched out into a single long column vector. Similarly defining $\psi'$ as a single column vector,

$$\psi' = \begin{bmatrix} \Delta_1^x: \\
\Delta_2^x: \\
\vdots \\
\Delta_N^x: \\
\Delta_1^y: \\
\Delta_2^y: \\
\vdots \\
\Delta_{N-1}^y: \end{bmatrix}. \hspace{1cm} (4.28)$$
A phase difference vector, \( \Delta y_{m,:} \), can be described in terms of the unwrapped phase vector, \( \bar{\phi}_{n,:} \), left multiplied by a matrix \( D_1 \). The structure of \( D_1 \) is determined by Eqs. (4.4) and (4.5),

\[
\begin{bmatrix}
\Delta y_{r,1} \\
\Delta y_{r,2} \\
\Delta y_{r,3} \\
\vdots \\
\Delta y_{r,N-1}
\end{bmatrix}
= \begin{bmatrix}
1 & -1 & 0 \\
0 & 1 & -1 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 1 & -1 & 0
\end{bmatrix}
\begin{bmatrix}
\phi_{r,1} \\
\phi_{r,2} \\
\phi_{r,3} \\
\vdots \\
\phi_{r,N}
\end{bmatrix},
\tag{4.29}
\]

where \( r \) is an index and \( D_1 \) is a \((N - 1) \times N\) matrix. In more compact notation \( \Delta y_{r,:} = D_1 \bar{\phi}_{r,:) \). Similar notation can be used to describe \( \Delta x_{m,:} \). From Eq. (4.4),

\[
\Delta x_{r,:} = I \bar{\phi}_{r,:} - I \bar{\phi}_{r+1,:} \tag{4.30}
\]

where \( I \) is a \( N \times N \) identity matrix. Now that this notation is developed the equation of interest, \( A \phi = \psi' \), can be expressed as,

\[
\begin{bmatrix}
\Delta y_{1,:} \\
\Delta y_{2,:} \\
\vdots \\
\Delta y_{N,:} \\
\Delta x_{1,:} \\
\Delta x_{2,:} \\
\vdots \\
\Delta x_{N-1,:}
\end{bmatrix}
= \begin{bmatrix}
\bar{\phi}_1 \\
\bar{\phi}_2 \\
\vdots \\
\bar{\phi}_r \\
\vdots \\
\bar{\phi}_N
\end{bmatrix},
\tag{4.31}
\]

Table 4.1 summarizes the dimensions of the matrices involved in Eq. (4.31).

Note that for a typical synthetic aperture radar image, \( N \) is on the order of \( 2^{11} \). A direct implementation of Eq. (4.31) is unwieldy even for powerful workstations. The matrix notation of Eq. (4.31) and the fact that \( A \) is sparse suggests a rapid algorithm for calculating the phase difference vector, \( \psi' \). This is useful for simulation
Table 4.1: Matrix Dimensions: Unweighted Least Squares

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Rows</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta y$</td>
<td>$(N - 1)$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$\Delta x$</td>
<td>$N$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$D_1$</td>
<td>$(N - 1)$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$I$</td>
<td>$N$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$\phi_r$</td>
<td>$N$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$\psi$</td>
<td>$N^2$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$2N(N - 1)N^2$</td>
<td>$\times$</td>
<td>$N^2$</td>
</tr>
<tr>
<td>$\phi$</td>
<td>$N^2$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$\psi'$</td>
<td>$2N(N - 1)N^2$</td>
<td>$\times$</td>
</tr>
</tbody>
</table>

purposes when the unwrapped phase is known and the wrapped phase differences must be calculated for implementing the ULS algorithm. Let $r$ be the index,

**Algorithm for calculating $\Delta^x$ and $\Delta^y$**

for $r = 1 : N - 1$

\[
\Delta^y_r = D_1 \phi_r;
\]

\[
\Delta^x_r = I^\phi_r - I^\phi_{r+1};
\]

end

\[
\Delta^y_{N_r} = D_1 \phi_{N_r};
\]

Hunt showed that $\rho$ can be represented by multiplying both sides of Eq. (4.26) by $A^T$,

\[
A^T A \phi = A^T \psi',
\]

\[
A^T A \phi = \rho.
\]

$\rho$ can be found directly from $\phi$ for simulation purposes.
4.2.2 Extending Hunt’s Work to the Weighted Case

Drawing from Ghiglia’s work, the weighted least squares problem in the notation of linear algebra is,

\[
WA\phi = W\psi'.
\] (4.34)

Multiplying on the left by \(A^TW^T\),

\[
A^TW^TWA\phi = A^TW^TW\psi',
\] (4.35)

\[
Q\phi = c,
\] (4.36)

where \(Q = A^TW^TWA\) and \(c = A^TW^TW\psi'\). Ghiglia points out that \(Q\) performs the discrete Laplacian operator on the weighted wrapped phase, and similarly, \(c\) is the result of the Laplacian operator acting on the discrete unwrapped phase. The details of this extension of Hunt’s work to the weighted case generates intuition on efficient implementation of the weighted least squares solution. In particular, the conjugate gradient algorithm can only be efficiently implemented from a careful look at the framework of the weighted case. Let the weights be defined,

\[
w^x_{m,n} = \min(w_{m+1,n}, w_{m,n}),
\] (4.37)

\[m = 1 \ldots M - 1, \quad n = 1 \ldots N,\]

\[= 0, \quad \text{otherwise}\]

\[
w^y_{m,n} = \min(w_{m,n+1}, w_{m,n}),
\] (4.38)

\[m = 1 \ldots M, \quad n = 1 \ldots N - 1,\]

\[= 0, \quad \text{otherwise}\]

where \(w_{m,n}\) is the magnitude of a given pixel. The minimum of two adjacent pixels is chosen as the weight for the corresponding phase difference since the phase estimation is only as reliable as the SNR, or magnitude at that point. The dimensions of \(w^x\) and \(w^y\) are the same as \(\Delta_x\) and \(\Delta_y\), (compare Eqs. (4.37) and (4.38) to Eqs. (4.4) and
(4.5). Diagonal weighting matrices in \( x \) and \( y \) are,

\[
W_r^x = \begin{bmatrix}
  w_{r,1}^x & & \\
  & w_{r,2}^x & \\
  & & \ddots \\
  & & & w_{r,N-1}^x \\
  & & & & w_{r,N}^x
\end{bmatrix}, \quad (4.39)
\]

and

\[
W_r^y = \begin{bmatrix}
  w_{r,1}^y & & \\
  & w_{r,2}^y & \\
  & & \ddots \\
  & & & w_{r,N}^y \\
  & & & & w_{r,N-1}^y
\end{bmatrix}. \quad (4.40)
\]

Combining \( W_r^x \) and \( W_r^y \) into a \( 2N(N - 1) \times 2N(N - 1) \) diagonal weighting matrix \( W \),

\[
W = \begin{bmatrix}
  W_1^y & & \\
  & \ddots & \\
  & & W_N^y \\
  W_1^x & & \\
  & \ddots & \\
  & & W_{N-1}^x
\end{bmatrix}. \quad (4.41)
\]

From Eqs. (4.39) - (4.41) it follows that \( W^T W \) has non-zero elements only along its diagonal and those elements are an unfolded version of two full matrices composed of the squared weights, \( (w_{m,n}^x)^2 \), \( (w_{m,n}^y)^2 \). Both sides of Eq. (4.36) contain the term \( A^T W^T W \). Using the notation \( W_r^{y2} = (W_r^y)^2 \), and \( W_r^{x2} = (W_r^x)^2 \),
\[ A^T W^T W = \]
\[
\begin{bmatrix}
D_1^y y^2 W_1^2 & 0 & 0 & w_1^2 & 0 & 0 \\
0 & D_1^y y^2 W_2^2 & 0 & -w_2^2 & w_3^2 & : \\
0 & 0 & D_1^y y^2 W_{N-1}^2 & -w_{N-2}^2 & w_{N-1}^2 & : \\
0 & 0 & 0 & D_1^y y^2 W_N^2 & 0 & 0 \\
\end{bmatrix}
\]

Finally, the term \( Q = A^T W^T W A \) is given as,

\[ Q = A^T W^T W A = \]
\[
\begin{bmatrix}
D_1^2 W_1^2 D_1 + w_1^2 & -w_1^2 & 0 & 0 \\
-w_1^2 & D_1^2 W_2^2 D_1 + W_2^2 & -w_2^2 & : \\
0 & -w_2^2 & D_1^2 W_{N-1}^2 D_1 + W_{N-2}^2 & -w_{N-1}^2 & : \\
0 & 0 & 0 & D_1^2 W_N^2 D_1 + W_N^2 & 0 \\
\end{bmatrix}
\]

\( Q \) is a \( N^2 \times N^2 \) matrix. The \( N \times N \) element \( D_1^2 W_1^2 D_1 \) can be put in the form,

\[ D_1^2 W_1^2 D_1 = \]
\[
\begin{bmatrix}
w_{r,1}^2 - w_{r,2}^2 & -w_{r,2}^2 & 0 & \cdots & 0 \\
-w_{r,2}^2 & w_{r,2}^2 - w_{r,3}^2 & -w_{r,3}^2 & 0 & \cdots \\
0 & -w_{r,3}^2 & \ddots & \ddots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & w_{r,N-2}^2 - w_{r,N-3}^2 & -w_{r,N-3}^2 \\
0 & \cdots & 0 & -w_{r,N-3}^2 & w_{r,N-2}^2 - w_{r,N-2}^2 \\
\end{bmatrix}
\]

With this structure in place the weighted least squares problem \( Q \phi = c \) can be solved.

The size of the matrices introduced for the WLS algorithm are given in Table 4.2.

### 4.2.3 Efficient Algorithms

The size of the matrices involved demand that the sparse properties of the matrices be used for efficient memory usage and matrix multiplies. An efficient algorithm
Table 4.2: Matrix Dimensions: Weighted Least Squares

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Rows</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W^2_i$</td>
<td>$N$</td>
<td>$2N$</td>
</tr>
<tr>
<td>$W_i^y$</td>
<td>$N$</td>
<td>$2N$</td>
</tr>
<tr>
<td>$W$</td>
<td>$2N(N-1)$</td>
<td>$2N(N-1)$</td>
</tr>
<tr>
<td>$A^TW^TW$</td>
<td>$N^2$</td>
<td>$2N(N-1)$</td>
</tr>
<tr>
<td>$Q$</td>
<td>$N^2$</td>
<td>$N^2$</td>
</tr>
<tr>
<td>$D_1^TW_{i^2}^D_1$</td>
<td>$N$</td>
<td>$2N$</td>
</tr>
</tbody>
</table>

for the matrix multiply, $Q\phi = c$ can be seen from,

$$Q = \begin{bmatrix}
\tilde{\phi}_1; \\
\tilde{\phi}_2; \\
\vdots \\
\tilde{\phi}_N;
\end{bmatrix} = \begin{bmatrix}
\tilde{c}_1; \\
\tilde{c}_2; \\
\vdots \\
\tilde{c}_N;
\end{bmatrix}.$$  \hfill (4.45)

**Algorithm for** $Q\phi = c$

$r = 1$

$$\tilde{c}_1; = (D_1^TW_{1^2}^D_1 + W_{1^2}^2)\phi_1; - W_{1^2}^2\tilde{\phi}_2;$$

for $r = 2 : N - 1$

$$\tilde{c}_r; = -W_{r-1}^r\tilde{\phi}_{r-1}; + (D_1^TW_{r^2}^D_1 + W_{r-1}^r + W_{r^2}^r)\phi_1; - W_{r^2}^r\tilde{\phi}_{r+1};$$

end

$r = N$

$$\tilde{c}_N; = -W_{N-1}^N\tilde{\phi}_{N-1}; + (D_1^TW_{N^2}^D_1 + W_{N-1}^2)\tilde{\phi}_1;$$

Thus, a right multiply by $Q$ can be implemented efficiently, Similarly an efficient method for the implementation of a right multiply by $A^TW^T$ is seen from

$A^TW^TW\psi' = c$;
$$\begin{bmatrix}
\Delta^{y}_{1,1} \\
\Delta^{y}_{2,1} \\
\vdots \\
\Delta^{y}_{N,1} \\
\Delta^{x}_{1,1} \\
\Delta^{x}_{2,1} \\
\vdots \\
\Delta^{x}_{N^{-1},1}
\end{bmatrix}
= \begin{bmatrix}
\tilde{c}_{1,1} \\
\tilde{c}_{2,1} \\
\vdots \\
\tilde{c}_{N,1}
\end{bmatrix}
$$ (4.46)

**Algorithm for** $\mathbf{A}^{T}\mathbf{W}^{T}\mathbf{W} \psi' = \mathbf{c}$

\[
r = 1 \\
\tilde{c}_{1,1} = \mathbf{D}^{T}_{1} \mathbf{W}^{y}_{1} \mathbf{\Delta}^{y}_{1,1} + \mathbf{W}^{x}_{1} \mathbf{\Delta}^{x}_{1,1} \\
\text{for } r = 2 : N - 1 \\
\tilde{c}_{r,1} = \mathbf{D}^{T}_{r} \mathbf{W}^{y}_{r} \mathbf{\Delta}^{y}_{r,1} - \mathbf{W}^{x}_{r-1} \mathbf{\Delta}^{x}_{r-1,1} + \mathbf{W}^{x}_{r} \mathbf{\Delta}^{x}_{r,1} \\
\text{end} \\
r = N \\
\tilde{c}_{N,1} = \mathbf{D}^{T}_{N} \mathbf{W}^{y}_{N} \mathbf{\Delta}^{y}_{N,1} - \mathbf{W}^{x}_{N-1} \mathbf{\Delta}^{x}_{N-1,1}
\]

4.2.4 Preconditioned Conjugate Gradient Algorithm

The solution to the WLS algorithm is found by iterative methods. An algorithm with one of the fastest convergence rates is the preconditioned conjugate gradient method. The ULS algorithm is used for an initial guess. The algorithm for the iterative WLS algorithm (see [10] and [28]) with $k$ as the iteration counter is:

**Conjugate Gradient Method for Weighted Least Squares Algorithm**

1st Iteration:
1. Compute $\mathbf{c}$ from the wrapped data, $\psi_{m,n}$. (Eq. (4.46))
2. Solve $\mathbf{Pz}_{0} = \mathbf{c}$ using Algorithm 1 (Initial Guess)
3. $\mathbf{p}_{1} = \mathbf{z}_{0}$
perform one scalar and two vector updates
4. \( \alpha_1 = r_0^Tz_0/p_1^TP_1 \) (Eq. (4.43) )
5. \( \phi = -\alpha_1p_1 \)
6. \( r_1 = r_0 - \alpha_1Qp_1 \)

For \( k = 2 : K^{th} \) Iteration:
1. Solve \( Pz_k = r_k \) using Algorithm 1
   perform one scalar and one vector update
2. \( \beta_k = r_{k-1}^Tr_{k-1}/r_{k-2}^Tr_{k-2} \)
3. \( p_k = z_{k-1} + \beta_kp_{k-1} \)
   perform one scalar and two vector updates
4. \( \alpha_k = r_{k-1}^Tr_{k-1}/p_k^TP_k \)
5. \( \phi = \phi_{k-1} - \alpha_kp_k \)
6. \( r_k = r_{k-1} - \alpha_kQp_k \)
end

The number of iterations, \( K \), can be fixed, or iterations may continue until the error reaches an acceptable level.

4.2.5 Examples of the the Weighted Least Squares Algorithm

In Figs. 4.10 - 4.12 Gaussian noise was added to a patch of the image. The right side of Fig. 4.10 is the weighting used for the WLS algorithm. Only weights of 0 or 1 were used. The result in Fig. 4.11 shows that the zero weighting forced the solution to ignore the region of noise. As the Poisson solution would intuitively suggest, the solution in the zero weighted region is just smoothed from the boundary conditions on the edge of the zero weighted square. Further iteration results in the distortion being localized to the zero weighted region as illustrated by only 3 iterations in Fig. 4.12.

Figs. 4.13 - 4.14 illustrate the WLS algorithm with the same branch cut of Figs. 4.8 - 4.9. Gaussian noise is added to the entire image. The first iteration is only slightly better than Fig. 4.9. The weighting applied on the left in Fig. 4.13
Figure 4.10: WLS algorithm input: Path of additive Gaussian Noise

allows for the two halves of the image to unwrap independently as shown in Fig. 4.16 for 10 iterations. The final error is limited only by the number of iterations. The WLS algorithm is guaranteed to converge in \(N\) iterations, but, as can be seen from this example, is much less than that in practice.

The WLS algorithm is capable of accurate phase unwrapping. The glaring hole in its usefulness is that the location of the branch cuts must be known \textit{a priori} in order to apply weighting in the proper location. If a branch cut is not weighted it will result in global distortions that diminish farther from the unweighted branch cut as in Figs. 4.7 and 4.9.
Figure 4.11: WLS algorithm output: Path of additive Gaussian Noise, 1st iteration

Figure 4.12: WLS algorithm output: Path of additive Gaussian Noise, 3rd iteration
Figure 4.13: WLS algorithm input: Path of additive Gaussian Noise

Figure 4.14: WLS algorithm output: Path of additive Gaussian Noise, 1st iteration
Figure 4.15: WLS algorithm output: Path of additive Gaussian Noise, 3rd iteration

Figure 4.16: WLS algorithm output: Path of additive Gaussian Noise, 10th iteration
4.2.6 Summary of the Least Squares Method

The Least Squares solution to the two dimensional phase unwrapping problem can be modeled as the solution to the Poisson equation with Neumann boundary conditions. Fourier techniques are used to solve the equation. A linear algebra model provides considerable insight on efficient programming algorithms. By using a Weighted Least Squares method and iterative techniques, such as the conjugate gradient algorithm, scenes with large phase discontinuities, or branch cuts, can be successfully unwrapped. The WLS algorithm also allows for low SNR areas to be masked out so that they do not have an effect on the solution.

4.3 Minimum Discontinuity Method for Phase Unwrapping

This section describes the unweighted and weighted minimum discontinuity solutions to two dimensional phase unwrapping. The mathematical model and methodology as presented by Flynn [12] is described and examples of the minimum discontinuity solution are presented. The final note, and the highlight of this section is a proposed method for improving the computational efficiency of Flynn’s algorithm by sub-dividing the solution domain, solving for the unwrapped phase using the minimum weighted discontinuity method, and seaming the sub-domains together to obtain the unwrapped phase for the entire image.

4.3.1 Introduction to Flynn’s Algorithm

Assuming that the nominal flat earth induced phase difference has been removed from the wrapped phase difference, $\psi$, the unwrapped phase difference, $\phi$ is given,

$$\phi = (\psi + 2\pi c). \quad (4.47)$$

$c$ is an integer defining the number of wraps, so that $c$, the wrap count array, is a matrix defining the number of wrap counts for each pixel in the image. The height of the image is a function of $\phi$,

$$h = \frac{\lambda^*}{2\pi \phi}. \quad (4.48)$$
The objective of phase unwrapping is to determine the value of \( c_{m,n} \) for all \((m, n)\). If \( c \) can be determined from the wrapped phase data, \( \psi \), the height of the scene can be determined. In the absence of noise, step discontinuities, and shadowing, two dimensional phase unwrapping is trivial. When the scene of interest contains steep slopes, shadowed areas, or areas of low SNR, phase unwrapping is significantly more difficult and sometimes completely impossible.

### 4.3.2 Minimum Discontinuity Criterion

When the jump between two adjacent pixels is greater than \( \pi \) the value of \( c \) can not be unambiguously resolved. Defining magnitude difference in \( c \) from pixel to pixel as the jump count, a jump count array can be defined for both the horizontal and vertical direction. The jump count is zero whenever the jump between adjacent pixels is less than \( \pi \). The vertical and horizontal jump count arrays are,

\[
\delta_{m,n}^v = \left\lfloor \frac{\phi_{m,n} - \phi_{m-1,n} + \pi}{2\pi} \right\rfloor, \quad (m, n) \in \Delta^v, \tag{4.49}
\]

\[
\delta_{m,n}^h = \left\lfloor \frac{\phi_{m,n} - \phi_{m,n-1} + \pi}{2\pi} \right\rfloor, \quad (m, n) \in \Delta^h, \tag{4.50}
\]

where the “floor” operation, \( \lfloor x \rfloor \), rounds \( x \) to the nearest integer towards zero, \( \Delta^v = \{2 \ldots M\} \times \{1 \ldots N\} \), and \( \Delta^h = \{1 \ldots M\} \times \{1 \ldots N\} \). The jump count arrays can be expressed in terms of the wrap count array defined in Eq. (4.47) as,

\[
\delta_{m,n}^v = \left\lfloor \frac{\psi_{m,n} + 2\pi c_{m,n} - \psi_{m,n} - 2\pi c_{m-1,n} + \pi}{2\pi} \right\rfloor, \tag{4.51}
\]

\[
= c_{m,n} - c_{m-1,n} + \left\lfloor \frac{\psi_{m,n} - \psi_{m-1,n} + \pi}{2\pi} \right\rfloor, \tag{4.51}
\]

\[
\delta_{m,n}^h = c_{m,n} - c_{m,n-1} + \left\lfloor \frac{\psi_{m,n} - \psi_{m,n-1} + \pi}{2\pi} \right\rfloor. \tag{4.52}
\]

In order to obtain the absolute height, the value for \( c_{m,n} \) must be know for at least one point in the image. Given \( c_{1,1} \), the value of \( c \) can be solved for. From Eqs. (4.51), (4.52),
for $m = 2 \ldots M$,
\[
c_{m,1} = c_{m-1,1} + \delta_{m,1}^v - \frac{\left( \psi_{m,1} - \psi_{m-1,1} + \pi \right)}{2\pi}, \quad (4.53)
\]

for $n = 2 \ldots N$,
\[
c_{m,n} = c_{m,n-1} + \delta_{m,n}^h - \frac{\left( \psi_{m,n} - \psi_{m,n-1} + \pi \right)}{2\pi}. \quad (4.54)
\]

The objective of the Flynn algorithm is to minimize the total magnitude of the jump counts. The total severity of the jumps is,
\[
E_o(c, \psi) = \sum_{(m,n) \in \Delta^v} |\delta_{m,n}^v| + \sum_{(m,n) \in \Delta^h} |\delta_{m,n}^h|. \quad (4.55)
\]

The minimum discontinuity (MD) solution minimizes the sum in Eq. (4.55). The solution that minimizes $E_o$ is not necessarily unique. Obtaining the jump count arrays directly from the data, (Eqs. (4.51), (4.52) with $c = 0$), the value of $\delta_{m,n}^v$ and $\delta_{m,n}^h$ will never exceed one. In real data there are jumps larger than $\pi$ and the value of the jump count array should be greater than one for those large jumps. In order to account for lay-over and shadowing where the phase is discontinuous the MD criteria can be extended to the weighted case,
\[
E_o(c, \psi) = \sum_{(m,n) \in \Delta^v} w_{m,n}^v |\delta_{m,n}^v| + \sum_{(m,n) \in \Delta^h} w_{m,n}^h |\delta_{m,n}^h|, \quad (4.56)
\]

where $w^v$ and $w^h$ are weighting matrices. The weights can be determined from the data by using the correlation coefficient as a means for setting the value of the weights. It is desirable to choose the weights such that they do not divide the image into isolated regions. Isolated regions will be unwrapped independently by an arbitrary integer offset of $2\pi$. As true of all phase unwrapping algorithms, “Phase unwrapping relies on the presence of slowly changing, accurately measured phase connecting all parts of the image” [12].

The minimum weighted discontinuity, (MWD), solution allows for jump counts that exceed one. The means by which this is achieved is apparent in the solution
method. The implications are that the size of the weight chosen limits the maximum magnitude of the jump count from pixel to pixel. The magnitude of the weights also determine the maximum number of low weighted pixels through which the phase can be unwrapped.

4.3.3 Flynn’s Method For Computing the Minimum Weighted Discontinuity Solution

Flynn’s method for finding the $c$ that minimizes Eq. 4.56 can be described in terms of “elementary operations” or EOs. An elementary operation divides the image into two sections. The phase in one section is raised by $2\pi$, equivalent to increasing the wrap count in one section of the image by 1. By increasing the phase of one section of the image, the sum of jump counts is reduced. Elementary operations are repeated until no EOs can be formed. Flynn showed that the MWD solution is achieved when no EOs are possible [12]. No EO is possible when the MWD sum can not be reduced by changing $c$. The method of determining where to divide the image so that an EO reduces the MWD sum is the meat of Flynn’s algorithm.

EO Example

The essence of an EO is to identify phase fringes and eliminate them. The method is best demonstrated by an example. For the sake of simplicity the unweighted case is shown. Figure 4.17 is a $4 \times 4$ pixel array representing the phase difference matrix, $\phi$. The integer and fractional part of the phase, $(\phi/2\pi)$, are shown for convenience. The arrows indicate the magnitude of the jump, a single arrow for a magnitude of one, and a double arrow for a magnitude of two. Arrows pointing left or up are negative jump counts and arrows pointing right or down are positive jump counts (see Eqs. (4.49), (4.50)). The border of the gray area in Fig. 4.17 indicates the location of a loop that divides the image into two sections. By adding $2\pi$ to each pixel inside the loop the jump counts along the border of the loop are altered and reduce $E_o$ from 12 to 3 as show in Fig. 4.18. Thus an EO reduces the jump count discontinuity sum of Eq. 4.55.
Figure 4.17: Example array of φ before removal of a loop. The gray area indicates the loop boundary.

After each loop is removed the modified matrix φ is systematically searched for additional loops that can further reduce $E_o$. The gray area of Fig. 4.18 indicates another loop that that will reduce the sum of jump discontinuities. By removing the loop, $E_o$ is reduced from 3 to 2 as indicated in Fig. 4.19.

By performing two elementary operations the sum of jump discontinuities has been minimized from an initial value of 12 to a final value of 2. The c satisfying the minimum jump discontinuity sum has been found by iteratively performing elementary operations.

Node Values, Edges, and Loops

Flynn’s algorithm systematically forms paths based on the ability of a path to decrease $E_o$. Paths are extended until they form a loop, at which time an EO is performed and $E_o$ is reduced. The ‘value’ of a given path is determined by the net change in the jump discontinuity sum removal of the path would induce. Paths are extended whenever the value of the path is positive. When a path can loop back
on itself and maintain a positive value, an EO is performed to remove the loop and decrease $E_o$.

For convenience let an ‘edge’ be defined as the line separating adjacent pixels, and a node as the intersection of edges. Fig. 4.20 represents a node as $\oplus$ and an edge as a dotted line between any two adjacent nodes.

A path is a sequence of edges that may diverge but does not converge. Each edge in that path has a value determined by the effect of removing the edge on the jump count sum. Edges between boundary nodes have no effect on the jump count sum and are always zero valued. The value of a path at a given node is the sum of all the edge values in the path from the path origin to the node. If $\delta V(m_k, n_k; m_{k+1}, n_{k+1})$ is the value of the edge from node $(m_k, n_k)$ to node $(m_{k+1}, n_{k+1})$, the value of a path from its origin to node $(m, n)$ is the sum of all the edge values in the path,

$$V(m, n) = \sum_{k=0}^{l} \delta V(m_k, n_k; m_{k+1}, n_{k+1}), \quad (4.57)$$

where $k = 0 \ldots l$ describes the order of the nodes in the path. Equation (4.57) defines

```
<table>
<thead>
<tr>
<th>0.9</th>
<th>1.4</th>
<th>1.8</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4</td>
<td>1.4</td>
<td>1.9</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td>↓</td>
<td>↓</td>
<td></td>
</tr>
<tr>
<td>1.8</td>
<td>2.2</td>
<td>2.7</td>
<td>3.1</td>
</tr>
<tr>
<td>2.2</td>
<td>2.7</td>
<td>3.1</td>
<td>3.6</td>
</tr>
</tbody>
</table>
```

Figure 4.18: $\phi$ array of Fig. 4.17 after removal of the loop. The sum of jump discontinuities has been reduced from 12 to 3.
the value of the path at a given node, or the node value. The path to any given node will always be unique, no two paths lead to the same node. The value of an edge is direction dependent, as is apparent in Eqs. (4.49), (4.50). For example, the value of an edge from a node on the left to a node on the right has the opposite sign of the value of the edge from a node on the right to a node on the left.

The value of an edge for the four possible directions are,

Right edge: (Decrements the vertical jump count),

\[
\delta V(m, n; m, n + 1) = -w_{m,n}^v \left( |\delta_{m,n}^v - 1| - |\delta_{m,n}^v| \right),
\]

\[
= w_{m,n}^v \text{sgn}(\delta_{m,n}^v - 1),
\]  

(4.58)

Left edge: (Increments the vertical jump count),

\[
\delta V(m, n + 1; m, n) = -w_{m,n}^v \left( |\delta_{m,n}^v + 1| - |\delta_{m,n}^v| \right),
\]

\[
= -w_{m,n}^v \text{sgn}(\delta_{m,n}^v),
\]  

(4.59)

Figure 4.19: The minimum discontinuity solution to the \( \phi \) array of Fig. 4.17. Removing the loop of Fig. 4.18. The final sum of jump discontinuities is \( E_o = 2 \).
Figure 4.20: Edges and nodes for the example array $\phi$

Up edge: (Decrements the horizontal jump count),

$$\delta V(m + 1, n; m, n) = -w_{m,n}^h \left( |\delta_{m,n}^h - 1| - |\delta_{m,n}^v| \right),$$
$$= w_{m,n}^v \text{sgn}(\delta_{m,n}^h - 1), \quad (4.60)$$

Down edge: (Increments the horizontal jump count),

$$\delta V(m, n; m + 1, n) = -w_{m,n}^h \left( |\delta_{m,n}^h + 1| - |\delta_{m,n}^v| \right),$$
$$= -w_{m,n}^v \text{sgn}(\delta_{m,n}^h), \quad (4.61)$$

where for the left and right edges, $(m, n) \in \Delta^v$, and for the up and down edges, $(m, n) \in \Delta^h$.

**Search Method**

Flynn’s algorithm scans the input array, $\psi$, creating and extending paths. Paths are extended only when adding an edge increases the value of the path at the new node, which is equivalent to a positive change in the node value,

$$\Delta V = V(m, n) + \delta V(m, n; m', n') - V(m', n'), \quad (4.62)$$
where the edge under investigation runs from \((m, n)\) to \((m', n')\). If \(\Delta V\) is positive the path is modified to include the new edge. If \(\Delta V\) is negative the edge is not added.

Starting with an initial value of zero at all nodes and with no paths defined, the following rules for path extension from \((m, n)\) to \((m', n')\) apply when \(\Delta V\) is positive. The term ‘isolated’ refers to a node that is not a member of any path. ‘Children’ of a node, refers to all subsequent members of a path, but not prior members.

1) **Edge addition:**
When \((m, n)\) and \((m', n')\) are both isolated nodes, (not a member of any path), a new path is created and the value of \((m', n')\) is \(\Delta V\). If only \((m', n')\) is isolated the path is simply extended and the value of the new node is assigned as \(\Delta V\). If \((m', n')\) is the root of a different path, the value of all nodes in that path are increased by \(\Delta V\).

2) **Edge replacement:**
Because multiple paths are present in a given array, the case of edge replacement must be considered. When a path extension to a node that is a member of a different path is made, the edge from the old path to the node is removed. The node is assigned a value of \(\Delta V\) and children of the node that is added are increased in value by \(\Delta V\).

3) **Loop completion:**
When the extension is to a member of the same path a loop is created. The loop is removed by an elementary operation which reduces the jump discontinuity sum. Because the removal of the loop may have caused children to have negative values, which are not allowed, all children of the nodes in the loop are removed.

Flynn’s algorithm is made efficient by a carefully structured search pattern. The array is searched first for right edges, then down edges, then left edges and finally up edges. By repeatedly scanning in this fashion many loops can be rapidly removed, as the circular fashion of the search would indicate.

### 4.3.4 Computation and Memory Requirements of Flynn’s Algorithm

Flynn’s Algorithm requires a minimum of five matrices, each the size of the data set. The arrays are,
1. Phase data, weight matrix, solution array,
2. Node Values,
3. Horizontal Jump Count,
4. Vertical Jump Count,
5. Origination.

The phase data, weighting array, and solution array can share the same storage space. The other arrays are needed simultaneously. The Origination array keeps track of path membership order and a few other tricks to improve the efficiency of the algorithm ([13] pp. 153 - 163).

Pritt and Ghiglia evaluated the speed of Flynn’s algorithm, ([13] p. 163). The time required for the algorithm increases logarithmically as show in Fig. 4.21. The time increases rapidly due to the longer paths possible in larger matrices. The Flynn algorithm is one of the most reliable methods for phase unwrapping, making it very attractive in spite of its large memory requirements and high computational demand.

![Figure 4.21: Matrix Size vs. Execution Time for Flynn’s Algorithm](image_url)
4.4 Patches to Speed up Flynn’s Method

Noting that the time required to execute Flynn’s algorithm increases logarithmically with the size of the matrix, it would appear desirable to decrease the longest possible path length. By dividing the image of interest into smaller subsections the execution time required for Flynn’s algorithm can be considerably reduced. In the limit of a the minimal $2 \times 2$ pixel patch Flynn’s algorithm is a simple nearest neighbor solution incapable of dealing with discontinuities or low SNR regions. Thus, there is some value for the minimum size of a matrix that must be chosen. The lower limit on the size of the patch is determined by the largest line of discontinuity in the image. If a line of discontinuity divides a patch into two isolated regions each region will be independently unwrapped, creating an undesirable effect.

4.4.1 Sewing the Patches

The complication with using patches arises in connecting the patches together to form a single coherent image. Since the patches are unwrapped independently, the patches will be offset from one another by an integer multiple of $2\pi$. Choosing the correct integer will result in a smooth transition from patch to patch. The integer that minimizes the number of jump discontinuities between the patches is the correct solution. Let the ‘seam’ between two patches be the two vectors that form the border between the two patches. One vector is from each patch. Let one vector be denoted by a prime, so that the jump discontinuity is,

$$\delta = c - c' + \left\lfloor \frac{\psi - \psi'}{2\pi} \right\rfloor. \quad (4.63)$$

The sum of the magnitudes of the jump counts is minimized by adding an integer value to $c$ or $c'$. The phase of all the points in one patch are incremented by an integer multiple of $2\pi$ to complete the seaming process. Minimizing the jump count sum effectively ‘seams’ the patches, minimizing the jump discontinuities between them. The computational requirement for seaming the patches is negligible compared to the requirements for computing the unwrapped phase of the individual patches.

84
4.4.2 Examples of the Patch Modification

A recursive call to a function capable of performing the patch sub-divisions can make the ‘patching’ a natural and simple modification to the Flynn Algorithm. By dividing the image into patches the execution time is reduced considerably. Fig. 4.22 plots the execution time for Flynn’s algorithm along with the time required for executing Flynn’s algorithm using patches of $64 \times 64$, $32 \times 32$, and $16 \times 16$ pixels. For a fair comparison, the number of pixels per wrap count was held constant for all matrix sizes. The execution times are for an HP C200 using unoptimized code. The execution time for the patch modified algorithm does not increase as rapidly as the execution time for Flynn’s algorithm. The computational savings are particularly significant for large matrices. The patch algorithm also uses considerably less memory.

![Figure 4.22: Matrix Size vs. Execution Time, Flynn’s Algorithm vs. Patch Modification](image-url)
None of the patches are needed simultaneously and the memory requirement is simply five matrices the size of a patch, plus a few vectors for the seams.

Figure 4.23 is the unweighted minimum discontinuity solution for Isolation Peak. The true height and the wrapped phase are shown in Fig. 3.2. Figure 4.24 is the unweighted minimum discontinuity solution for Isolation Peak using four patches. Discontinuities in the image cross the patches, and result in sections of the image being unwrapped independently, as is apparent in Figs. 4.25 and 4.26. Figure 4.25 is the magnitude of the height difference between Flynn’s method and the patch method. Figure 4.26 is a magnitude plot of the height estimates. The upper image is the MD solution and the lower image is the MD solution performed on four sub-sections of the image. The figure shows that the patch algorithm could not ‘seam’ together the patches. Some of the patches were properly seemed together, but there are large discontinuities at the boundary of many of the patches.
Figure 4.24: The unweighted minimum discontinuity solution for Isolation Peak using four patches. The discontinuities in the image that span across entire patches cause errors in the height estimate. The patches were not successfully seamed together.

4.4.3 Disadvantages and Future Work

Of concern in subdividing patches are the size and location of the discontinuities in the image. An automated method for subdividing the image is desirable. The minimum size of the patch can be made region dependent, and regions of the image with small areas of discontinuities or low weights can use smaller patches than other areas.

The computational time required by Flynn’s algorithm can be considerably reduced by inputing an initial guess of the phase. The initial guess can be made using a fast algorithm, such as Goldstein’s method. The initial estimate removes most of the discontinuities, but may not be the weighted minimum discontinuity solution. With most of the discontinuities removed by the fast preliminary algorithm, the discontinuity minimization task performed by Flynn’s algorithm is simplified resulting in significantly faster execution. Likewise, an initial guess input to the patch modification of the Flynn algorithm will exhibit considerable computational savings and a significant relaxation of the memory requirements.
Figure 4.25: Height difference between the Flynn algorithm and the patch modification to the Flynn algorithm for Isolation Peak. Each patch was independently unwrapped. Discontinuities running across the patches caused the ‘seaming’ to fail.

Work needs to be done to determine the optimal methods for unwrapping a given scene. The fastest method will inherently be scene dependent.
Figure 4.26: A comparison of the magnitude of the height for Isolation Peak. The upper image is the unweighted minimum discontinuity solution and the bottom image is the same solution using four patches. From the lower image it is apparent that the patch algorithm could not ‘seam’ together the patches.
4.5 Linking the Least Squares Solution and the Minimum Discontinuity Solution

Flynn [12] showed the link between the minimum norm solution and the minimum discontinuity solution. The least squares solution, (See Eqs. (4.4) - (4.6)), is a special case of the $L_p$-norm solution, which minimizes,

$$
E = \sum |\phi_{m,n} - \phi_{m-1,n} - (\psi_{m,n} - \psi_{m-1,n})_{2\pi}|^p \\
+ \sum |\phi_{m,n} - \phi_{m,n-1} - (\psi_{m,n} - \psi_{m,n-1})_{2\pi}|^p. \quad (4.64)
$$

The least squares solution does not inherently guarantee a solution that is congruent with the input wrapped phase data. If the solution is forced to be congruent with the input unwrapped phase data, Eq. (4.47) will hold,

$$
E = \sum |\phi_{m,n} - \phi_{m-1,n} + 2\pi (c_{m,n} - c_{m-1,n}) - (\phi_{m,n} - \phi_{m-1,n})_{2\pi}|^p \\
+ \sum |\phi_{m,n} - \phi_{m,n-1} + 2\pi (c_{m,n} - c_{m+1,n}) - (\phi_{m,n} - \phi_{m,n-1})_{2\pi}|^p. \quad (4.65)
$$

From Eqs. (4.51) and (4.52), this is equivalent to minimizing,

$$
E = \sum_{(m,n)\in \Delta^v} |\delta_{m,n}^v|^p + \sum_{(m,n)\in \Delta^h} |\delta_{m,n}^h|^p, \quad (4.66)
$$

which is the minimum discontinuity solution when $p = 1$, (See Eq. (4.55)). Thus we see that the congruent minimum $L_1$ norm solution is equivalent to the minimum discontinuity solution. The $L_p$ solution uses FFT’s and the conjugate gradient solution to a matrix equation while the minimum discontinuity solution is based on path following for discontinuity removal. The methods used to find the solutions are starkly different, but solve the same problem.
Chapter 5

Multiple Baseline Height Resolution Enhancement

The applications of interferometry are limited by the restrictions imposed by the wrapped nature of the phase data. Phase unwrapping algorithms typically fail to determine the phase for region of data isolated by phase discontinuity. However, as indicated in this chapter, a judicious choice of multiple baseline lengths can reduce or eliminate the need for phase unwrapping. The height of a scene can be unambiguously resolved, even for regions of the image isolated by phase discontinuities.

5.1 Prior Work

In recent years several conference papers and a few journal articles have addressed methods for reducing or eliminating the need for phase unwrapping. Most of the proposed methods use an initial estimate of the height from a DEM or from a short baseline interferogram to resolve the ambiguity in determining $\phi$ from $\psi$. The various approaches are briefly discussed here along with the results of simulations and experiments.

Xu et al. [15] was one of the first to publish work on the use of multiple baselines and/or multiple frequencies simultaneously to resolve the height of a surface from interferograms. Xu et al. briefly investigated three possible methods for unwrapping the phase. The first uses the Chinese Remainder Theorem and is too sensitive to noise for practical use. The second is the projection method. The projection method uses $N$ estimates of the height from $N$ interferograms simultaneously to resolve the height. Assuming that the $N$ height estimates always assign the correct height to
a given pixel, the resulting height estimate will have a reduced height error standard deviation. Without the noise reducing effect of a filter, however, the projection method suffers from incorrect height estimates. The final method investigated by Xu et al. is a linear combination of the interferograms. The noise is combined as a linear combination in this method and increases with the number of interferograms used.

The work of Homer et al. [17] on multi-baseline interferometry is essentially a beam forming and direction of arrival problem. Through the use of $N$ antennas an array is formed that is capable of resolving the height to considerably accuracy. Homer’s methods may prove useful with further development.

Massonnet et al. [18] authored an excellent method for using $N$ interferograms to increase the height sensitivity in differential interferometry. Using a DEM as an initial estimate of the height, multiple passes over a given area yield interferograms with a wide range of height sensitivities. From the initial estimate of the height, predictions of the fringe pattern for interferograms with better height sensitivities can be made iteratively. Massonnet does not attempt to determine the absolute height of the scene, but only to determine the height difference due to temporal changes in the scene’s elevation. Phase unwrapping or estimation of the height is not a requirement for high resolution differential interferometry using $N$ interferograms.

Jakowatz et al. [16] used the projection method proposed by Xu et al. on a computer simulated interferogram generated from a DEM. Three antenna were used to create two simulated interferograms. Jakowatz work provides a rough analysis of the the probability of mis-calculating the height by the projection method and uses a $3 \times 3$ pixel average to reduce the effects of the errors. The filtered height estimate is visually, but not quantitatively compared to the known height of the scene. The final height estimate provided by Jakowatz et al. is not congruent with the phase data of the larger baseline, and the issues involved in using more than two interferograms are not discussed.
Consini et al. [20] also generated a computer simulation of an interferogram from a DEM model for the three antenna scenario. They show that three antennas can produce height maps that are significantly more accurate than those generated by conventional phase unwrapping, essentially by reducing the number or areas in the image with an ambiguous unwrapped phase value.

Schmitt and Wiesbeck [19] used two carrier frequencies in an anechoic chamber. Using 30 GHz and and 37 GHz, they demonstrated that the need for phase unwrapping can be eliminated. The lower frequency was used to determine the height of the scatters and the higher frequency to further resolve the height and form an image.

5.2 Iterative Approach to Height Estimation

As stated in [20], “The main unresolved problem of this processing is still the error propagation of the noise effects.” The use of multiple interferograms to determine the height may result in an increased amount of noise in the height estimate. Any linear combination of interferograms will result in an increase in the noise measurement and an unreliable estimate of the height of the scene. The following work addresses the error propagation issue and presents an application similar to projection method suggested by Xu et al. [15] to more than three antennas while minimizing the propagation of noise induced errors.

With the background presented in Chapter 3, a new iterative method for improving the height resolution using multiple baselines can be derived. Starting with an initial estimate of the height, \( \hat{h} \), with a given \( \sigma_h \), a longer baseline with a lower \( \sigma_h \) and \( \lambda^* \) can be used to obtain a more accurate estimate of the height. The new estimate of the height is then used as an initial guess to obtain an even more accurate estimate of the height, etc. The limiting factor in the height resolution attainable by this iterative method is the critical baseline length. The following derivation follows some of the ideas presented by Xu et al. [15] and Jakowitz et al. [16].
5.2.1 Problem Definition and Setup

An initial estimate of the height can come from any source, e.g. a DEM or interferogram. For the case of an interferogram, if the baseline is sufficiently small for the given imaging geometry (see Eq. (3.7)), an estimate of the height requiring minimal or no phase unwrapping can be obtained. The method presented here assumes that the initial height estimate comes from an interferogram, but can be easily adapted to an initial height estimate from other sources. Prior to explaining the method, some background and definitions are needed. Recall from Chapter 3, Eqs. (3.8), (3.9), (3.28),

\[
\begin{align*}
h_o & = \frac{\lambda^* \phi_o}{2\pi}, \\
& = \frac{\lambda^*}{2\pi} (\psi_o + 2\pi c_o), \\
\sigma_{ho} & = \frac{\lambda^*}{2\pi} \sigma_{\phi_o},
\end{align*}
\]

where the subscript ‘o’ refers to the true, or noiseless value of the parameter. Solving for \(c_o\) in Eq. (5.2),

\[
c_o = \left( \frac{h_o}{\lambda^*} - \frac{\psi_o}{2\pi} \right).
\]

Let the estimates of \(h_o, \phi_o, \text{ and } \psi_o\), for interferogram \(i\) be defined,

\[
\begin{align*}
\hat{h}_i & = h_o + \epsilon_{hi}, \\
\hat{\phi}_i & = \phi_o + \epsilon_{phi}, \\
& = (\psi_o + 2\pi c_o) + \epsilon_{phi}, \\
\hat{\psi}_i & = \psi_o + \epsilon_{\psi i}.
\end{align*}
\]

Note that \(\epsilon_{\phi} = \epsilon_{\psi}\). The relation between the error in the height estimate and the error in the phase estimate is given by Eq. (3.28),

\[
\epsilon_{hi} = \frac{\lambda^*_i}{2\pi} \epsilon_{phi}.
\]

The equations and definition given here are prerequisite to understanding the method for iteratively improving the height estimate from an initial estimate that follows.
5.2.2 Method for Reducing the Height Ambiguity

Iterative estimation of the height uses two interferograms at a time. One interferogram has a short baseline and is used to decrease the phase ambiguity in the height estimation. The other interferogram is used to obtain a high resolution height estimation. Let the interferogram with the shorter baseline be denoted by the subscript ‘s’ and the interferogram with the longer baseline be denoted by the subscript ‘l’. Let the initial estimate of the height be from the small baseline, \( B_s \), to give a height sensitivity \( \lambda_s^* \) with a standard deviation in the height of \( \sigma_{hs} \). An initial estimate of the height from the small baseline, \( \hat{h}_s \), is obtained from the phase directly or by phase unwrapping if necessary. The wrapped phase of the two interferograms are \( \hat{\psi}_s \) and \( \hat{\psi}_l \), and the length of the baseline for each of the interferograms is assumed to be no more than \( 0.8 \times B_c \), (see section 3.3.2, page 38). These constraints imply the inequalities,

\[
\lambda_s^* > \lambda_l^*, \tag{5.9}
\]

\[
\sigma_{hs} > \sigma_{hl}, \tag{5.10}
\]

\[
\sigma_{\phi s} < \sigma_{\phi l}. \tag{5.11}
\]

\[
(5.12)
\]

In order to reduce the ambiguity in the estimate of the height from the small baseline, the more accurate, \( \sigma_{hs} > \sigma_{hl} \), height information from the longer baseline must be used. This can be achieved by first estimating the jump count for the longer baseline using the height estimate from the shorter baseline. From Eqs. (5.4 - 5.7),

\[
\hat{c}_l = \text{int} \left( \frac{\hat{h}_s}{\lambda_l^*} - \frac{\hat{\psi}_l}{2\pi} \right), \tag{5.13}
\]

\[
= \left( \frac{\hat{h}_o}{\lambda_l^*} - \frac{\psi_o}{2\pi} \right) + \text{int} \left( \frac{\epsilon_{hs}}{\lambda_l^*} - \frac{\epsilon_{\phi l}}{2\pi} \right), \tag{5.14}
\]

where \( \text{int}(x) \) rounds to the nearest integer. Note that the first term will always be an integer by definition, and is in fact the correct jump count, (see Eq. (5.4)). Relating the error in height to the error in the phase through Eq. (5.8), \( \hat{c}_l \) is,

\[
\hat{c}_l = c_o + \text{int} \left( \frac{\epsilon_l}{\lambda_l^*} \right), \tag{5.15}
\]

95
where \( \epsilon_t \) is the total error,

\[
\epsilon_t = \epsilon_{hs} - \epsilon_{hi}. 
\]

(5.16)

The rounding operation in Eq. (5.14) is performed so that when \( \hat{c}_i \) is used to obtain an estimate of the height, \( \hat{h}_i \), using Eq. (5.2), the height estimate will be congruent with the phase data of the large baseline, \( \hat{\psi}_i \). Rounding to the nearest integer has several interesting implications. First of all, when the total error is less than \( \lambda_i^*/2 \) the effect of the error on the height estimate will be eliminated as shown in Eq. (5.15). When \( |\epsilon_t| > \lambda_i^*/2 \), the error is large enough that \( \hat{c}_i \) will be different from \( c_o \) by an integer value. Accordingly, the new height estimate will be off by an integer multiple of \( \lambda_i^* \). By Eqs. (5.7 - 5.8) and (5.15 - 5.16), the new height estimate, \( \hat{h}_n \), is,

\[
\begin{align*}
\hat{h}_n &= \frac{\lambda_i^*}{2\pi}(\hat{\psi}_i + 2\pi\hat{c}_i), \\
&= \frac{\lambda_i^*}{2\pi}(\hat{\psi}_i + 2\pi c_o) + \lambda_i^* \text{int} \left( \frac{\epsilon_{hs} - \epsilon_{hi}}{\lambda_i^*} \right), \\
&= \hat{h}_i + \lambda_i^* \text{int} \left( \frac{\epsilon_t}{\lambda_i^*} \right). 
\end{align*}
\]

(5.17)

The probability of an error and its effect on the height estimate must be carefully evaluated. If the effect of the integer error in the estimate \( \hat{c}_i \) can be reduced or removed it will be possible to obtain an estimate of the height, \( \hat{h}_n \), that is better than the initial estimate, \( \hat{h}_s \).

The pdf for \( \epsilon_t \) is Gaussian (see Eqs. (3.10), (3.14), and (3.28) and Fig. 3.3 ). Assuming that \( \epsilon_s \) and \( \epsilon_t \) are uncorrelated, the standard deviation of the total error is,

\[
\sigma_{ht} = \sqrt{\sigma_{hs}^2 + \sigma_{hi}^2}. 
\]

(5.18)

The probability distribution function of the error is,

\[
p(\epsilon) = \frac{1}{\sigma_{ht}\sqrt{2\pi}} \exp \left( -\frac{\epsilon^2}{2\sigma_{ht}^2} \right). 
\]

(5.19)

The probability that \( \hat{c}_i \neq c_o \), resulting in an error in the new height estimate is,

\[
P_{\text{error}} = P \left( |\epsilon_t| > \frac{\lambda_i^*}{2} \right), \\
= 2 \int_{\lambda_i^*/2}^{\infty} p(\epsilon) d\epsilon, \\
= 1 - 2\text{erf} \left( \frac{\lambda_i^*}{2\sigma_{ht}} \right),
\]

(5.20)
where the error function is defined,

\[
erf(x) = \frac{1}{\sqrt{2\pi}} \int_0^x \exp(-x^2/2) dx.
\]  

(5.21)

The probability distribution function of the new height estimate, \( \hat{h}_n \), is illustrated in Fig. 5.1. The new pdf is a discretely sampled Gaussian distribution. For example, when \( \lambda_i^*/2 < \epsilon_l < 3\lambda_i^*/2 \) an error of 1 will occur in \( \hat{c}_l \), which is equivalent to an error in the phase estimate of \( 2\pi \). By Eq. (5.17) or (3.8) this is equivalent to an error of \( \lambda_i^* \) in the height estimate, \( \epsilon_{hn} = \lambda_i^* \). The pdf, mean, and variance are,

Figure 5.1: The probability distribution function for the error in the new height estimate. An error in estimating the jump count will always be an integer value. The error in the phase estimate is an integer multiple of \( 2\pi \) and the error in the height estimate is an integer multiple of \( \lambda_i^* \).
\[
\text{pdf}(\epsilon_{\phi n}) = \delta(0) P_{\text{corr rect}} + \sum_{n=1}^{\infty} [\delta(2n\pi) + \delta(-2n\pi)] \times \frac{1}{2} P \left( \frac{(2n - 1) \lambda_i^*}{2} < \epsilon_i < \frac{(2n + 1) \lambda_i^*}{2} \right),
\]

\[
\mu_{\phi n} = E(\epsilon_{\phi n}),
\]

\[
= \int_{-\infty}^{\infty} \epsilon_{\phi n} \cdot f(\epsilon_{\phi n}) d\epsilon_{\phi n},
\]

\[
= 0,
\]

\[
\sigma_{\phi n}^2 = E[(X - \mu)^2],
\]

\[
= \int_{-\infty}^{\infty} (x - \mu)^2 f(x) dx,
\]

\[
= 8\pi^2 \sum_{n=1}^{\infty} \left[ \text{erf} \left( \frac{(2n + 1) \lambda_i^*}{2\sigma_{ht}} \right) - \text{erf} \left( \frac{(2n - 1) \lambda_i^*}{2\sigma_{ht}} \right) \right],
\]

\[
= \frac{8\pi^2}{\sqrt{2\pi}} \sum_{n=1}^{\infty} n^2 \int_{(2n-1)\lambda_i^*/\sigma_{ht}}^{(2n+1)\lambda_i^*/\sigma_{ht}} \exp(-t^2) dt.
\]

The pdf for the error in the height, pdf(\(\epsilon_{hn}\)) follows from pdf(\(\epsilon_{\phi n}\)) by Eq. (3.8). Assuming a finite bandwidth on the noise, the value of \(\sigma_{hn}\) can be determined from Eq. (5.24) using a finite summation. Noise bandwidth will determine the value of the standard deviation in the phase, and therefore, the values over which Eq. (5.24) should be summed. The larger the noise bandwidth, the larger the summation must be. For the examples in this thesis the summation was carried out to six standard deviations to insure a high degree of accuracy in the calculation of \(\sigma_{\phi n}\).

When \(\sigma_{hn} < \sigma_{hs}\) the height estimate of \(h_n\) is better than \(h_s\) and \(h_n\) can be used as a new height estimate with a standard deviation of \(\sigma_{hn}\) and a sensitivity of \(\lambda_i^*\). By iteratively proceeding with smaller and smaller \(\lambda_i^*\) to achieve better estimates of \(h_0\) a height sensitivity limited by the baseline length giving complete spatial decorrelation can be achieved.

### 5.2.3 Baseline Length Selection for Iterative Height Estimation

Of great significance is the choice of baseline lengths. The baseline length is inversely proportional to the sensitivity of the interferometer as shown by Eq. (3.7). If the value for \(\lambda_i^*\) is too large, \(\lambda_i^* \approx \lambda_s^*\), \(\sigma_{hl}\) will also be large resulting in an
increased probability of error, $\sigma_{ht} = \sqrt{\sigma_{hs} + \sigma_{hl}} \gg \sigma_{hs}$. If the value for $\lambda_i^*$ is too small compared to $\lambda_s^*$ the probability of error will be unacceptably large, resulting in a poor height estimate, (See Eq. (5.20)). Thus, there exists an optimal value for the baseline lengths.

The probability of error is used to determine the appropriate baseline length for improving the height estimate. To reduce the chance of a bad estimate of $\phi_l$ a low probability of error must be chosen. The appropriate probability of error is dependent on the type of filter that will be used as described in the following section. Given a desired $P_{error}$, the first step in finding the ideal longer baseline length is to determine the correlation coefficient from Eq. (3.15). Assuming no temporal decorrelation,

$$\gamma = \left( \frac{1}{1 + \text{SNR}^{-1}} \right) \left( 1 - \frac{|B_{\perp}|R_y}{\lambda R_a \tan(\theta)} \right).$$

(5.25)

Alternatively, for real data the complex correlation coefficient, $\gamma$ can be calculated directly from the data, (Eq. (3.12)). $\sigma_{\phi_l}$ and $\sigma_{hs}$ can then be calculated from $|\gamma|$ according to Eqs. (3.14) and (3.28). Equation (5.20) and the inverse error function are used to calculate $\lambda_i^*$,

$$\lambda_i^* = 2\sigma_{ht,\text{inverf}} \left( \frac{1 - P_{error}}{2} \right).$$

(5.26)

Because $\sigma_{hl}$ is dependent on $\sigma_{hl}$ which is a function of $\lambda_i^*$, Eq. (5.26) must be solved numerically. The final step in finding the desired longer baseline, $B_l$, is to use Eq. 3.7,

$$B_l = \frac{\lambda R_a}{\lambda_i^*} [\sin(\alpha) - \cos(\alpha) \tan(\alpha - \theta)].$$

(5.27)

When the look angle and the baseline tilt are identical $B_i\lambda_i^*$ is a constant,

$$B_i = \frac{B_s \lambda_s^*}{\lambda_i^*}.$$  

(5.28)

This method is iteratively used to find larger and larger baselines based on the desired probability of error and the accompanying standard deviation and baseline length until the upper limit of (0.5 - 0.8) $B_c$ is achieved, or the maximum number of antennas is reached.
5.2.4 Filters for Iterative Height Estimation

In order to iteratively achieve smaller and smaller $\sigma_{hi}$ values for each new $\hat{h}_i$ it is beneficial to use a filter to reduce the phase noise. The errors in $\hat{h}_n$ will be integer multiples of $\lambda_i^*$. Two common and easily implemented filter are mentioned here. More advanced filters may result in lowering the number of interferograms and/or the length of the baselines required to achieve a given quality for the height estimate.

One filter for removing the effect of the errors is a spatial average. Jakowitz [16] demonstrated that a $3 \times 3$ pixel average effectively reduced the number of errors. His result is not congruent with the input phase, $\psi_s$. By applying a congruency operator after the filtering operation the error can often be further reduced and the resulting $\hat{h}_f$ has a $\sigma_{ hf}$ significantly less than the $\sigma_{hn}$ of Eq. (5.24). A spatial filter will reduce the standard deviation of the height estimate according to,

$$\sigma_{ hf}^2 = \frac{\sigma_{hn}^2}{N}, \quad (5.29)$$

where $N$ is the number of pixels averaged and a uniform weighting is assumed. (Note: spatial averaging, or Multi-Look processing, is commonly used to decrease the noise and speckle effects in SAR images). On the negative side, the spatial averaging reduces the image resolution and the sharpness of true discontinuities in the image, as is demonstrated in section 5.3.1.

Another filter for removing the integer multiples of $\lambda_i^*$ in the height estimate is the median filter. Because the errors are large compared to the rate at which the surface topography is changing, the median value of a $3 \times 3$ pixel region results in very little smoothing or loss of sharpness and avoids averaging in the errors as in the spatial average. The median filter serves to reduce the height standard deviation to some degree, but the effect can not be analytically determined due to the non-linear nature of the filter, and must be measured empirically. Congruency with the underlying data can be forced, but will result in a standard deviation equivalent to that of the underlying data. Constraining the estimate to be congruent does not necessarily reduce the standard deviation of the height estimate.
The type of filter that is used is dependent on the desired result and the underlying data. In an mountainous environment with rapid and continual elevation changes, the median filter may not be the ideal choice and the mean filter may provide the best estimate of the height. In an urban environment a median filter is superior due to it’s edge preserving qualities. Much of the decision on the filter type and the chosen probability of error is qualitative in nature. The ability of a filter to remove error in the height estimate determines the probability of error that is chosen, and therefore, the optimal baseline lengths for a given geometry.

5.3 Case Study for YINSAR

Examples demonstrating the ability to iteratively refine the height estimate in an urban and a natural environment are presented in this section. The parameters for the baseline tilt, look angle, and instrument elevation are given for YINSAR as listed in Table 5.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td>300 m</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>0.03 m</td>
</tr>
<tr>
<td>$R_y$</td>
<td>0.6 m</td>
</tr>
<tr>
<td>$B$</td>
<td>1 m</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>45°</td>
</tr>
<tr>
<td>$\theta$</td>
<td>45°</td>
</tr>
<tr>
<td>SNR</td>
<td>10 dB</td>
</tr>
</tbody>
</table>

It is assumed that the antennas are spaced non-uniformly with identical baseline tilts as in Fig. 5.2. The length of the initial baseline length is determined by the maximum step height in the image and the physical constraints on how close the antennas may be placed together. YINSAR uses slotted wave guide antennas and may be spaced as close as 20 cm from one another. Given that the scene to be imaged
Figure 5.2: An example of the antenna configuration. Here, four antennas form three baselines for use in iterative height estimation. Each antenna pair has an identical baseline tilt, $\alpha$.

has a maximum step of 30 meters, (an eight story building for the urban scene, and a 30 m mountain for the natural scene), the required baseline length is 30 cm for a sensitivity of $\lambda_{hs} = 30.30$ m and $\sigma_{hs} = 3.40$ m. The phase will not wrap and the problem of phase unwrapping is completely avoided. The standard deviation of the initial height estimate is unacceptably large and can be reduced by iteratively using longer baselines. The maximum baseline length is constrained by the critical baseline length, for YINSAR $B_c = 21.4$ m. The examples in this section all use an initial height estimate from a baseline of length 0.3 m, and demonstrate the result of four iterations. For a fair comparison of the measured standard deviation in the examples, the final iteration was also filtered. In practice, filtering after the final iteration will result in a decrease in spatial resolution and may not be desirable.

5.3.1 Results from an Urban Scene

An urban scene is illustrated in Fig. 5.3. Georectification and the removal of the nominal flat earth induced phase difference are assumed. It is also assumed that data with an SNR value as in Table 5.1 is available at all point in the image. The georectification and SNR assumptions are addressed in sections 5.4.4 and 5.4.5. Fig. 5.3 illustrates the true height of a simulated urban scene. Each ground pixel represents $0.6 \times 0.6$ meters. The tallest building in the image is eight stories tall, assuming 3 m equals one story. The building with a pyramid shaped roof reaches a total height of 21 meters and the smaller building in the foreground is four stories.
Figure 5.3: True Height of the Simulated Urban Scene

tall with 1 story side wings and steps leading up to the building (steps are not clearly visible from the perspective shown). In the upper corner of the image a one story and two story house is simulated with 2 m high fence posts placed along the road. The road is .3 meters below the level of the buildings, the size of a large curb. A 3 m high truck and a 2 m high car are visible in the streets as well as a 6 m high 1 pixel wide bar representing a stop light support. 6 m street lights are present between the tallest building and the houses.

For a comparison of phase unwrapping of an urban scene with multi-baseline techniques, consider Fig. 5.4. For simplicity, noiseless phase measurements were input to the minimum discontinuity method to unwrap the phase shown in the figure. Phase unwrapping algorithms cannot determine the height of regions of the image isolated by phase discontinuities, and assume the step size is less than λ* in magnitude. When the phase is continuous, such as for the slanted roofs and the steps leading up to the
building with side wings, the phase is properly unwrapped.

Figure 5.4: Height estimate of an urban scene using a single interferometric baseline and Flynn’s minimum discontinuity phase unwrapping method. For simplicity, noiseless phase measurements were used. Traditional phase unwrapping algorithms can not determine the height when step discontinuities are present. Phase unwrapping assumes any step discontinuity to have a magnitude of less than \( \lambda^* \).

In order to help the reader establish some intuition about the choice of filters and probability of error, examples for the urban scene for \( P_{\text{error}} = 0.01 \) and \( P_{\text{error}} = 0.05 \) are shown for both the 3\( \times \)3 pixel mean filter and the 3\( \times \)3 pixel median filter. A cross section of the scene is particularly useful in a qualitative understanding of the results. Localized errors are identified by a magnitude plot of the difference between the initial and final height estimate. Tables identifying the standard deviations in the height and the baseline lengths are useful quantitative measurements for each
example. In the tables, $\sigma_{\text{measured}}$ is the measured value of the standard deviation of the height, and $\sigma_h$ is the calculated value of the standard deviation for a given baseline length. All lengths are in meters.

**Mean Filter on an Urban Scene**

The first example is the mean filter for $P_{\text{error}} = 0.05$. Figure 5.5 is a cross section of the scene in Fig. 5.3. The true height, initial estimate of the height and the estimate from the second and fourth iterations are shown. The rounding effects of the iterative spatial average are a noticeable undesirable side effect of the filter. The

![Filter: mean, $P_{\text{error}} = 0.05$](image)

Figure 5.5: Cross section of the urban scene showing the iterative improvement in the height estimation. The baseline lengths were chosen according to Table 5.2.

values for $\sigma_{\text{measured}}$ in Table 5.2 are from a flat area in the scene and do not account
Table 5.2: Baseline length selection for $P_{\text{error}} = 0.05$. The measured standard deviations represent the result of using a $3 \times 3$ mean filter to eliminate errors in the height estimate.

<table>
<thead>
<tr>
<th>$B$</th>
<th>$\lambda^*$</th>
<th>$\sigma_{\text{measured}}$</th>
<th>$\sigma_h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>30.3</td>
<td>3.4</td>
<td>3.4</td>
</tr>
<tr>
<td>0.593</td>
<td>15.3</td>
<td>0.951</td>
<td>1.8</td>
</tr>
<tr>
<td>1.11</td>
<td>8.19</td>
<td>0.427</td>
<td>1.03</td>
</tr>
<tr>
<td>1.86</td>
<td>4.89</td>
<td>0.318</td>
<td>0.671</td>
</tr>
<tr>
<td>2.71</td>
<td>3.35</td>
<td>0.263</td>
<td>0.497</td>
</tr>
</tbody>
</table>

for the rounding effect at each step discontinuity in the image. The results in the table are intuitively pleasing. For each iteration the new value for $\lambda^*$ is roughly half of the previous value. As the baseline length increases the spatial decorrelation increases and the percent change in $\lambda^*$ correspondingly decreases. Figure 5.6 illustrates the result of the iterations performed using the baselines in Table 5.2. As expected the light poles and fence which were only one pixel wide are completely removed by the $3 \times 3$ pixel mean filter that was applied to the data. The stop light support pole is undetectable for the same reason. The sides of the buildings all have a distinct slope due to the spatial averaging filter, and look as though they were draped. Figure 5.7 is a magnitude plot of the difference between the known height and the estimated height. The largest differences occur at the corners and edges of the buildings. The inability to properly identify the height of small objects is apparent in that the fence and the light poles were completely ignored. The final plot of Fig. 5.8 is the error in the final height estimate for the cross section shown in Fig. 5.5. The error is largest at the edges of the buildings where the mean filter has smoothed the jump discontinuities.
Figure 5.6: Final height estimate of the scene using a $3 \times 3$ mean filter. The resulting standard deviation is listed in Table 5.2.
Figure 5.7: Magnitude plot of the error in the height estimate. White corresponds to no error and black corresponds to larger errors.
Figure 5.8: The error in the final height estimate for a cross section of the urban scene using $P_{error} = 0.05$ and a mean filter. The largest errors occur where the mean filter has failed to maintain sharp edges, such as those found at the boundary of each building.
As a second example, the mean filter is again used, but with $P_{\text{error}} = 0.01$. Figures 5.9 - 5.11 show a cross section, a 3-D view of the final height estimate, and a magnitude plot of error in the final height estimate. Table 5.3 lists the choice of baseline lengths and result of each iteration in terms of the height standard deviation.

![Filter: mean, $P_{\text{error}} = 0.01$](image)

Figure 5.9: Cross section of the urban scene showing the iterative improvement in the height estimation. The baseline lengths were chosen according to Table 5.3.

Comparing the height estimate using the mean filter for the $P_{\text{error}} = 0.05$ and $P_{\text{error}} = 0.01$ examples, it is obvious that $P_{\text{error}} = 0.01$ is too low, (compare Tables 5.2 and 5.3). The final baseline length for the $P_{\text{error}} = 0.05$ example is longer than that of the $P_{\text{error}} = 0.01$ example and resulted in a lower standard deviation in the height estimate. $P_{\text{error}} = 0.01$ is too small to give an optimal value for $\sigma_{\text{measured}}$. Consider
Table 5.3: Baseline length selection for $P_{error} = 0.01$. The measured standard deviations represent the result of a using a $3 \times 3$ mean filter to eliminate errors in the height estimate.

<table>
<thead>
<tr>
<th>B</th>
<th>$\lambda^*$</th>
<th>$\sigma_{measured}$</th>
<th>$\sigma_h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>30.3</td>
<td>3.4</td>
<td>3.4</td>
</tr>
<tr>
<td>0.414</td>
<td>21.9</td>
<td>0.877</td>
<td>2.5</td>
</tr>
<tr>
<td>0.559</td>
<td>16.3</td>
<td>0.604</td>
<td>1.9</td>
</tr>
<tr>
<td>0.706</td>
<td>12.9</td>
<td>0.492</td>
<td>1.54</td>
</tr>
<tr>
<td>0.865</td>
<td>10.5</td>
<td>0.426</td>
<td>1.28</td>
</tr>
</tbody>
</table>

$P_{error} = 0.10$ as show in Table 5.4. For the final height estimate using $P_{error} = 0.10$, $\sigma_{measured} > \sigma_h$, but $\sigma_{measured}$ is less than the measured height standard deviation using $P_{error} = 0.05$. A wise choice of $P_{error}$ is critical for optimal height resolution.

Table 5.4: Baseline length selection for $P_{error} = 0.10$. The measured standard deviations represent the result of a using a $3 \times 3$ mean filter to eliminate errors in the height estimate.

<table>
<thead>
<tr>
<th>B</th>
<th>$\lambda^*$</th>
<th>$\sigma_{measured}$</th>
<th>$\sigma_h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>30.3</td>
<td>3.4</td>
<td>3.4</td>
</tr>
<tr>
<td>0.739</td>
<td>12.3</td>
<td>1.01</td>
<td>1.48</td>
</tr>
<tr>
<td>1.65</td>
<td>5.51</td>
<td>0.525</td>
<td>0.739</td>
</tr>
<tr>
<td>3.16</td>
<td>2.88</td>
<td>0.387</td>
<td>0.442</td>
</tr>
<tr>
<td>4.98</td>
<td>1.82</td>
<td>0.319</td>
<td>0.317</td>
</tr>
</tbody>
</table>
Figure 5.10: Final height estimate of the scene using a $3 \times 3$ mean filter. The resulting standard deviation is listed in Table 5.3.
Height Error. Filter: mean, $P_{error} = 0.01$

Figure 5.11: Magnitude plot of the error in the height estimate
Median Filter on an Urban Scene

Now consider the use of a median filter on the urban scene. Figures 5.12 - 5.15 are for $P_{\text{error}} = 0.05$ and Figs. 5.16 - 5.19 are for $P_{\text{error}} = 0.01$. Tables 5.5 and 5.6 show the baseline lengths used and the resulting height standard deviations, measured from each height estimate, $\sigma_{\text{measured}}$, and measured from for a given baseline, $\sigma_h$.

Figure 5.12: Cross section of the urban scene showing the iterative improvement in the height estimation. The baseline lengths were chosen according to Table 5.5.

Several observations can be made from the examples with the urban scene. Small objects, such as the fence posts and light poles were undetectable with either filter. Objects that are large compared to the filter size were properly resolved. It is apparent that the median filter is superior due to its edge preserving qualities.
Table 5.5: Baseline length selection for $P_{\text{error}} = 0.05$. The measured standard deviations represent the result of a using a $3 \times 3$ median filter to eliminate errors in the height estimate.

<table>
<thead>
<tr>
<th>B</th>
<th>$\lambda^*$</th>
<th>$\sigma_{\text{measured}}$</th>
<th>$\sigma_h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>30.3</td>
<td>3.4</td>
<td>3.4</td>
</tr>
<tr>
<td>0.593</td>
<td>15.3</td>
<td>0.748</td>
<td>1.8</td>
</tr>
<tr>
<td>1.11</td>
<td>8.19</td>
<td>0.428</td>
<td>1.03</td>
</tr>
<tr>
<td>1.86</td>
<td>4.89</td>
<td>0.293</td>
<td>0.671</td>
</tr>
<tr>
<td>2.71</td>
<td>3.35</td>
<td>0.237</td>
<td>0.497</td>
</tr>
</tbody>
</table>

Table 5.6: Baseline length selection for $P_{\text{error}} = 0.01$. The measured standard deviations represent the result of a using a $3 \times 3$ median filter to eliminate errors in the height estimate.

<table>
<thead>
<tr>
<th>B</th>
<th>$\lambda^*$</th>
<th>$\sigma_{\text{measured}}$</th>
<th>$\sigma_h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>30.3</td>
<td>3.4</td>
<td>3.4</td>
</tr>
<tr>
<td>0.414</td>
<td>21.9</td>
<td>1.04</td>
<td>2.5</td>
</tr>
<tr>
<td>0.559</td>
<td>16.3</td>
<td>0.77</td>
<td>1.9</td>
</tr>
<tr>
<td>0.706</td>
<td>12.9</td>
<td>0.559</td>
<td>1.54</td>
</tr>
<tr>
<td>0.865</td>
<td>10.5</td>
<td>0.53</td>
<td>1.28</td>
</tr>
</tbody>
</table>

As shown in Fig. 5.14, errors from using the median filter occur only at the corners of the building, while errors from using the mean filter occur at every edge, (Fig. 5.11).

Focusing on the median filter, the implications of the choice of $P_{\text{error}}$ can be seen. For $P_{\text{error}} = 0.05$ the final height standard deviation is lower than that for the $P_{\text{error}} = 0.01$ example, (Compare Tables 5.5 and 5.6). The higher probability of error allowed for the final baseline length to be more than twice that for the lower $P_{\text{error}}$ which is the reason for the the lower height standard deviation. The low height standard deviation allows the subtle feature of the curbs along the streets and a small pedestal in front of the largest building to appear in Fig. 5.13. The larger height standard deviation of the $P_{\text{error}} = 0.01$ example is larger compared to the height of the curb, and the curb can only be seen with a stretch of the imagination in Fig. 5.17.
Figure 5.13: Final height estimate of the scene using a $3 \times 3$ median filter. The resulting standard deviation is listed in Table 5.5. The median filter is edge preserving, but there are some localized errors that appear as short ‘pillars’ in the image. The final standard deviation in the height estimate is small enough that the curbs bordering the streets, and a small platform in front of the tallest building are visible.

On the other hand, localized large errors are present for the mean filter using $P_{\text{error}} = 0.05$ as shown in Figs. 5.13 and 5.14. Errors look like small ‘pillars’ at random locations in the image and are more abundant for larger $P_{\text{error}}$. Using $P_{\text{error}} = 0.01$, the possibility of localized errors is reduced if not eliminated, (See Figs. 5.17 and 5.18).

Thus, there is a trade off between the amount and type of error in the image. For a lower $P_{\text{error}}$ there are fewer localized errors, but the resulting height standard deviation for a given number of interferometric baselines is larger. For a higher $P_{\text{error}}$ there are more localized errors, but the final height standard deviation is lower for a given number of baselines.
Figure 5.14: Magnitude plot of the error in the height estimate

Figure 5.15: The error in the final height estimate for a cross section of the urban scene using $P_{\text{error}} = 0.05$ and a median filter. A large localized error is apparent.
Figure 5.16: Cross section of the urban scene showing the iterative improvement in the height estimation. The baseline lengths were chosen according to Table 5.6.
Figure 5.17: Final height estimate of the scene using a $3 \times 3$ median filter. The resulting standard deviation is listed in Table 5.6. The edges of the buildings are preserved by the median filter.
Figure 5.18: Magnitude plot of the error in the height estimate. Errors only occur in detecting objects that are small compared to the type of filter, and at the corners of the buildings.
Figure 5.19: The error in the final height estimate for a cross section of the urban scene using $P_{\text{error}} = 0.01$ and a median filter.
5.3.2 Results from a Natural Scene

Figure 5.20 is an image of a scaled DEM of Isolation Peak Colorado. (DEM data courtesy of Ghiglia and Pritt, used with permission [13]). The image is sampled in the SAR slant range. Examples of a median filter and a mean filter for a $3 \times 3$ pixel area are compared for a probability of error of 0.01. Similar to the examples for the urban scene, a cross section of the iterative height estimation, a magnitude plot of the error in the height estimation, and a cross section of the error are illustrated. The baseline lengths, height sensitivities, measured standard deviation, and estimated height standard deviation are listed in tabular form.

Mean Filter

Figures 5.21 - 5.23 and Table 5.7 illustrate iterative height estimation of Isolation Peak using a $3 \times 3$ pixel filter and a probability of error of 0.01.
Table 5.7: Baseline length selection for $P_{error} = 0.01$. The measured standard deviations represent the result of a using a $3 \times 3$ mean filter to eliminate errors in the height estimate.

<table>
<thead>
<tr>
<th>B</th>
<th>$\lambda^*$</th>
<th>$\sigma_{\text{measured}}$</th>
<th>$\sigma_h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>30.3</td>
<td>3.4</td>
<td>3.4</td>
</tr>
<tr>
<td>0.414</td>
<td>21.9</td>
<td>0.966</td>
<td>2.5</td>
</tr>
<tr>
<td>0.559</td>
<td>16.3</td>
<td>0.644</td>
<td>1.9</td>
</tr>
<tr>
<td>0.706</td>
<td>12.9</td>
<td>0.541</td>
<td>1.54</td>
</tr>
<tr>
<td>0.865</td>
<td>10.5</td>
<td>0.448</td>
<td>1.28</td>
</tr>
</tbody>
</table>

Figure 5.21: Cross section of the iterative height estimate for Isolation Peak scene using a $3 \times 3$ mean filter. The resulting standard deviation is listed in Table 5.7.
Figure 5.22: Magnitude plot of the error in the height estimate for $P_{error} = 0.01$ using a mean filter. The mean filter fails at the step discontinuity at the left side of the image.

Figure 5.23: The error in the final height estimate for a cross section of the Isolation Peak using $P_{error} = 0.01$ and a mean filter.
Median Filter

Figures 5.24 - 5.27 and Table 5.8 illustrate iterative height estimation of Isolation Peak using a $3 \times 3$ pixel median filter and a probability of error of 0.01.

Table 5.8: Baseline length selection for $P_{error} = 0.01$. The measured standard deviations represent the result of a using a $3 \times 3$ median filter to eliminate errors in the height estimate.

<table>
<thead>
<tr>
<th>B</th>
<th>$\lambda^*$</th>
<th>$\sigma_{measured}$</th>
<th>$\sigma_h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>30.3</td>
<td>3.4</td>
<td>3.4</td>
</tr>
<tr>
<td>0.414</td>
<td>21.9</td>
<td>1.12</td>
<td>2.5</td>
</tr>
<tr>
<td>0.559</td>
<td>16.3</td>
<td>0.851</td>
<td>1.9</td>
</tr>
<tr>
<td>0.706</td>
<td>12.9</td>
<td>0.692</td>
<td>1.54</td>
</tr>
<tr>
<td>0.865</td>
<td>10.5</td>
<td>0.591</td>
<td>1.28</td>
</tr>
</tbody>
</table>

Comparing the results of the mean filter and the median filter show that the median filter and the mean filter produce similar results for height estimation of the scene. The mean filter can not handle the step discontinuity in the image, but has a lower final error standard deviation for the same set of baseline lengths. The proper choice of filters and the choice of probability of error have a significant effect on the accuracy of the final height estimate.
Figure 5.24: Cross section of the iterative height estimate of the height for Isolation Peak scene using a $3 \times 3$ median filter. The resulting standard deviation is listed in Table 5.8.

Figure 5.25: Magnitude plot of the error in the height estimate for $P_{error} = 0.01$ using a median filter. The median filter has the most difficulty at the step discontinuity apparent at the left side of the image.
Figure 5.26: Final height estimate of the scene using a $3 \times 3$ median filter. The resulting standard deviation is listed in Table 5.8.

Figure 5.27: The error in the final height estimate for a cross section of the Isolation Peak using $P_{\text{error}} = 0.01$ and a median filter.
5.4 Other Issues

There are a number of additional issues concerning using multiple baselines that need addressed. The issues include combining phase unwrapping with multiple baseline techniques, image registration and orthorectification, obtaining high SNR samples for all points in the image, and range dependent height sensitivity.

5.4.1 Design Trade Offs

As is inherent in almost any design problem there are trade offs to consider when designing a multiple baseline interferometer. Choosing a low probability of error for the calculation of the baseline lengths may result in a large number of baselines to achieve a given height standard deviation. A lower probability of error relaxes the requirements on the filter and improves spatial resolution, but an increased number of baselines results in greater hardware and/or software requirements and more iterations. On the other hand, choosing a high probability of error will certainly result in a smaller number of baselines, but the filtering requirements become more stringent and there will be a loss in the spatial resolution of the resulting height estimate. A happy medium between probability of error, filter type and effectiveness, and the number of baselines must be achieved.

5.4.2 Non-Iterative Methods

It may be possible to use the information from the $N$ interferograms simultaneously to resolve the height. An optimal solution might be attainable by such a method where each interferogram is weighted according to the reliability of the data, i.e. the data with the lowest $\sigma_h$ would be weighted the highest. By using the data simultaneously there would be no need for iteration. With a little luck an optimal method may be found for multiple baseline interferometric height resolution.

5.4.3 Partially Avoiding Phase Unwrapping

When the number of receive channels is limited it may be desirable to alternate the iterative height estimation steps with actual phase unwrapping. At any point in
the process the estimate of the height can be used to guide a phase unwrapping algorithm, or phase unwrapping can be used to create an estimate of the height.

For example, such a technique could be advantageous in an urban scene where large step discontinuities can not be resolved by phase unwrapping techniques. Local averaging of an initial guess of the height from a small baseline could be used to estimate the jump count at points in the image where conventional phase unwrapping cannot resolve the true height. Consider Table 5.9 and Fig. 5.28.

Table 5.9: Phase unwrapping of the urban scene guided by an intial height estimate.

<table>
<thead>
<tr>
<th>B</th>
<th>$\lambda^*$</th>
<th>$\sigma_{measured}$</th>
<th>$\sigma_h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>30.3</td>
<td>3.4</td>
<td>3.4</td>
</tr>
<tr>
<td>0.414</td>
<td>21.9</td>
<td>0.911</td>
<td>2.5</td>
</tr>
<tr>
<td>0.559</td>
<td>16.3</td>
<td>0.584</td>
<td>1.9</td>
</tr>
<tr>
<td>0.706</td>
<td>12.9</td>
<td>0.479</td>
<td>1.54</td>
</tr>
<tr>
<td>0.865</td>
<td>10.5</td>
<td>0.961</td>
<td>1.28</td>
</tr>
</tbody>
</table>

Six antennas are used to create five baselines. The minimum discontinuity method for phase unwrapping was used to unwrap the phase for the longest baseline. An estimate of the jump count for the longer baseline based on spatially averaged phase data from a shorter baseline was used to determined the jump count for the unwrapped phase. Traditional phase unwrapping is incapable of estimating the absolute phase, the result of which is apparent by an offset of an integer multiple of $\lambda^*$ from the true height in Fig. 5.28. Traditional phase unwrapping also fails to properly resolve regions isolated by step discontinuities, the height of the building was not resolved without the guidance of the smaller baseline. The result of larger errors in the jump count estimate are evident in the guided phase unwrapping data.
Guided Phase Unwrapping, Filter: mean, $P_{\text{error}} = 0.01$

Figure 5.28: Cross section of height estimate for the urban scene using an initial height estimate. Traditional phase unwrapping is incapable of estimating the absolute phase, the result of which is apparent by an offset of an integer multiple of $\lambda^*$ from the true height. Traditional phase unwrapping also fails to properly resolve regions isolated by step discontinuities.

5.4.4 Image Registration and Georectification

Two key steps prerequisite to applying the multi-baseline technique are image registration and georectification. The images must be registered, e.g. each pixel must correspond to the same point on the ground. Re-sampling techniques can achieve sub-pixel registration accuracy [29]. Image registration and re-sampling are addressed on pp. 291-302 of [30]. Georectification, or orthorectification is a rather large assumption, and an area that needs additional research. Some of the issues of mapping a SAR image to a ground plane can be found on pp 317-329 of [30]. It has been
demonstrated that orthorectification can be achieved using the processing power of modern workstations [31] or PCs [32]. Automated methods for orthorectification have been developed for airborne interferometric SARs using differential GPS and motion compensated data [32].

5.4.5 Multiple Look Angles for Complete Imaging of Urban Scenes

Under the assumption of georectified images, complete topographical imaging of urban scenes may be possible. If the scene is imaged from multiple aspect angles, a high SNR return for each georectified point in the image can be collected. With a reliable sample for each point the iterative scheme presented in this chapter can be employed to resolve the height everywhere in the image. The numerous passes necessary for such imaging would create a large set of interferograms. The choice of which interferograms to estimate the height from can be based on the local values of $\sigma_h$, $\gamma$, and $\lambda$ in the interferograms. By choosing estimates of the height from highly correlated interferograms and iteratively choosing pairs with longer baselines and lower $\sigma_h$ the height of the urban scene could be resolved unambiguously everywhere.

5.4.6 Utilizing Range Dependent Height Sensitivity

It may be possible to use the range dependence of the height sensitivity of an instrument like YINSAR to implement the iterative height estimation technique using a single baseline length and multiple passes. The different passes would each have a different incidence angle relative to the area of interest to create interferograms different height sensitivities. An appropriate combination of the interferograms could be used for iterative height estimation. Similarly, passes with identical incidence angles, but at different ranges from the area of interest can be used to create a set of interferograms with height sensitivities that can be applied in an iterative manner to resolve the height of a scene.
Chapter 6

Conclusions

This thesis describes several methods for determining surface elevation from interferometric SAR data. Basic SAR signal processing steps are outlined as a background for understanding interferometry. The effects of interferometer imaging geometry and phase decorrelation on the instrument’s height sensitivity are evaluated. Criteria for the optimal imaging geometry to minimize errors in the height measurement and maximize height sensitivity are outlined. Examples of each of the two predominant phase unwrapping methods, the \( L_p \)-norm and minimum discontinuity methods, are presented. Finally, a new approach to surface elevation estimation using \( N \) interferograms is described. Noise induced errors are reduced in each of the \( N \) iterative height estimates by a careful choice of baseline lengths and filters. Examples demonstrate the effects of different baseline lengths and filters on the final height resolution for an urban and a natural environment.

6.1 Personal Observations from Research Experience

As with any significant work, the creator of the work learns far more from the creation process than he/she can possibly convey by known forms of communication. There are endless specific details I could share on the mathematical intricacies of SAR signal processing, the trade-offs in SAR interferometric design, or the ‘tricks’ to coding different algorithms. However, I feel that some more general comments are of greater value to those performing research. Experienced researchers are likely to find my observations rather obvious. In sharing my observations I hope that the reader may learn from my follies and be wiser for heading my suggestions:
When tackling a new topic unfamiliar to the researcher books published on the subject are a useful introduction covering the topic in a general manner. The references in books are valuable places to start delving in for a deeper understanding of the topic. Books rarely contain sufficient detail to be more than useful in obtaining a general knowledge of a topic. In my experience it is always necessary to refer to the original journal article or conference paper to find the detail necessary to understand and implement the ideas that are presented. When a publication is difficult to understand, and even when it is not, the researcher should come back to it in a week, and a month, and even year. As the researcher’s knowledge grows his/her ability to understand the material will increase, and each time the work is read understanding will deepen.

As far as understanding complex solutions, algorithms, etc., the best way to ensure the researcher understands is to try it. Chances are that the researcher will discover that his/her comprehension was much less mature than supposed. There is no better way to obtain a deep understanding of a solution method or algorithm than to try it. I have found that sitting down with pencil and paper in a quite location where I can concentrate and work it out step by step and page by page is an invaluable means of learning.

Trying new algorithms and ideas of my own follow the same rules. When I have not performed the necessary preliminary work to ensure I have a deep understanding of what I am trying to do and how I am going to do it before I sit down at a computer or a piece of equipment to test my theories, I waste a great deal of time. Learning by trial an error is highly inefficient. It is wise for the researcher to take a few hours with pencil and paper and refine and expand the new concept before implementation is attempted. Simplified cases are the most useful for initial tests. Start simple and work up the more general case.

A note on the use of computers is in order. Never trust a computer. Anyone who has had significant experience with computers will agree that a computers is a tool to verify and broaden the researcher’s understanding, and not a means of creating understanding. The tool is only as powerful as the human that is commanding it.
I have spent many hours on ‘wild goose chases’ of what I thought the results of my computer programs suggested, only to find that there was a bug in my code, or I misused a command.

6.2 Summary of Contributions

The contributions of this thesis are:

- Dual side band range and azimuth compression are analytically described.
  YINSAR is unique in its use of dual side-band chirp. To my knowledge an analytical analysis of range and azimuth compression of dual side band SAR has not been published. The analysis shows that the range and azimuth resolution are comparable with single side-band SAR systems. The minimum range and azimuth sampling rates are derived.

- The optimal interferometric imaging geometry for YINSAR is determined.
  The work of several authors on interferometric SAR design are combined and applied to the range dependent YINSAR system. The optimal imaging geometry for YINSAR is proposed.

- Hunt’s matrix formulation is extended to the weighted least squares case. In order to apply the weighted least squares phase unwrapping technique, an extension of Hunt’s matrix formulation is required. The weighted case of the matrix description of the least squares formulation has not been published.

- The efficiency of the minimum discontinuity 2-D phase unwrapping method is increased many fold by unwrapping the image in smaller sections and seaming the sections together. Flynn’s minimum discontinuity algorithm is one of the most robust phase unwrapping algorithms. However, it’s demanding memory and computational requirements make it largely unusable, especially for large matrices. By performing Flynn’s algorithm on small subsections of the image substantial savings in terms of computation time and memory requirements are achieved.

- A new iterative approach to interferometric height estimation using N antennas is developed. A careful choice of interferometric baseline lengths to create N interferograms can reduce and/or eliminate the need for phase unwrapping. For the
first time noise propagation and the probability of an error in the height estimation are directly addressed. \( N \) interferograms can be used in an iterative manner to create a high resolution digital elevation map. The method is applicable to images containing regions isolated by phase discontinuities such as are common in urban environments. Computationally demanding phase unwrapping algorithms are not required for the \( N \) baseline iterative height estimation technique proposes in this thesis.

6.3 Future Research

Work needs to be done to determine the optimal filters to achieve the most accurate height estimate with the smallest number of baselines. The ability of a filter, in particular the non-linear filters, to reduce the probability of an error in the height needs to be evaluated. In order to perform the iterative height estimation high SNR returns from every point in the image must be available. Methods for obtaining a high SNR return for all points in the image need to be examined. One method may be the combination of interferograms formed from multiple look angles in such a manner that no area is always shadowed. The means of combining the interferograms from multiple look angles into as single image need to be developed. The method for combining the interferograms will involve a weighting scheme based on the reliability of the phase measurement. Redundant measurements could be averaged for noise reduction.

Any set of interferograms that are used for multi-baseline height estimation must be registered in relation to one another and in relation to the earth. Geo-rectification is an integral part of image registration. Ortho-rectification must properly account for and eliminate regions of layover in the image.

Sufficient satellite data is available that the multi-baseline technique proposed in this thesis can be implemented using existing data sets in a fashion similar to that found in [18]. The effectiveness of the technique on actual data needs to be evaluated by comparison with existing high resolution DEMS.
Bibliography


